

Серия SYSMAC  
CS/CJ

# Программируемые контроллеры

**РУКОВОДСТВО ПО  
ПРОГРАММИРОВАНИЮ**

**OMRON**

# **Серия SYSMAC CS**

**CS1G/H-CPU□□ -EV1**

**CS1G/H-CPU□□ H**

**CS1D-CPU□□ H**

**CS1D-CPU□□ S**

# **Серия SYSMAC CJ**

**CJ1G-CPU□□**

**CJ1G/H-CPU□□ H**

**CJ1M-CPU□□**

# **Программируемые контроллеры**

## **Руководство по программированию**




*Редакция: Декабрь 2003*



## **Примечание:**

Продукты компании OMRON должны использоваться надлежащим образом, только для целей, описанных в настоящем руководстве, и только квалифицированным персоналом.

В настоящем руководстве для обозначения различных типов опасности используются следующие предупреждающие знаки. Пренебрежение этой информацией может привести к травмированию людей или нанесению материального ущерба.

-  **ОПАСНОСТЬ** Указывает на чрезвычайно опасную ситуацию, которая, если ее не избежать, приведет к смерти или серьезной травме.
-  **ВНИМАНИЕ** Указывает на потенциально опасную ситуацию, которая, если ее не избежать, может привести к смерти или серьезной травме.
-  **Предупреждение** Указывает на потенциально опасную ситуацию, которая, если ее не избежать, может привести к травме средней или легкой степени тяжести, или нанесению материального ущерба.

## **Вспомогательные обозначения**

Для выделения информации различного типа в левой колонке настоящего руководства используются следующие заголовки.

**Примечание** Особенно интересная и полезная информация о наиболее эффективных и удобных способах работы с изделиями.

- 1,2,3...** 1. Обозначение последовательности действий или любого другого списка.

© OMRON, 2001

Все права защищены. Воспроизведение, размещение в информационно-поисковой системе или передача третьему лицу какой-либо части настоящего руководства в какой-либо форме и каким-либо способом (механическим, электронным, путем ксерокопирования, записи на носитель или иным способом) не допускается без предварительного письменного разрешения компании OMRON.

Использование информации, содержащейся в настоящем руководстве, не сопряжено с какой-либо патентной ответственностью. Кроме того, поскольку компания OMRON неуклонно стремится к совершенствованию своей продукции, информация, содержащаяся в настоящем руководстве, может быть изменена без предупреждения. Подготовка настоящего руководства выполнялась с надлежащей тщательностью. Тем не менее, компания OMRON не несет ответственности за какие-либо ошибки и упущения. Компания OMRON не несет юридической ответственности за повреждения, явившиеся результатом использования информации, содержащейся в настоящем руководстве.

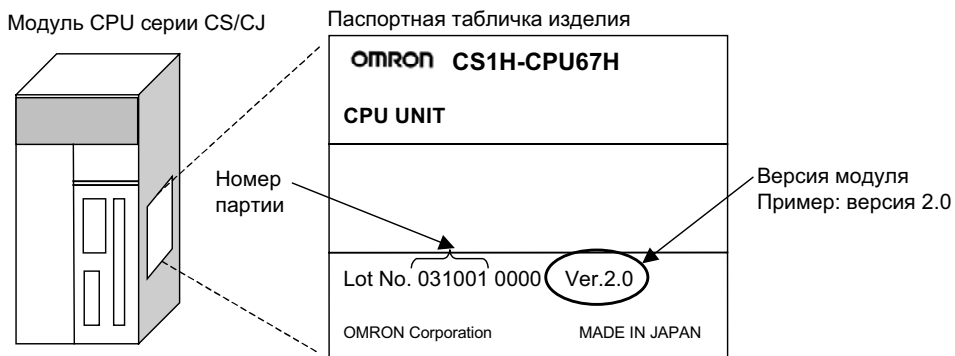
# Версии модулей CPU серии CS/CJ

## Версии модулей

Понятие "версия модуля" ("исполнение модуля") было введено для классификации модулей CPU серии CS/CJ в соответствии с различиями в их функциональности, возникающими по мере обновления модулей. Это относится к модулям CPU CS1-H, CJ1-H, CJ1M и CS1D.

### Обозначение версии модуля на изделии

Версия модуля указывается справа от номера партии на паспортной табличке изделия, для которого введена классификация версий (см. рис. ниже).



- У модулей CPU CS1-H, CJ1-H и CJ1M (за исключением моделей низшего класса), изготовленных до 4 ноября 2003г., версия не указывается (т.е., указанное выше место для версии модуля является пустым).
- Нумерация версий модулей CPU серии CS1-H, CJ1-H и CJ1M, а также модулей CPU CS1D для однопроцессорных систем начинается с версии 2.0.
- Нумерация версий модулей CPU серии CS1D для систем с дублированием CPU начинается с версии 1.1.
- Модули CPU, у которых номер версии не указан, называются модулями *Pre-Ver.* □□ (модули до версии такой-то), например, "модули CPU до версии 2.0" или "модули CPU до версии 1.1".

### Определение версии модуля с помощью программного обеспечения поддержки

Для определения версии модуля можно использовать CX-Programmer V4.0. Существует два способа.

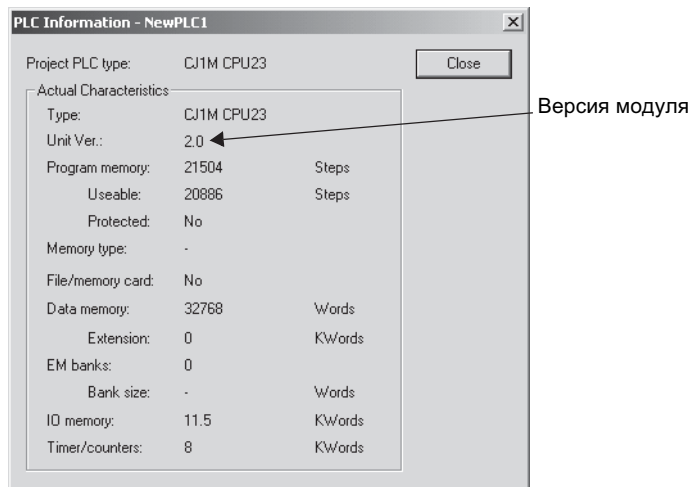
- Использование **PLC Information** (Сведения о ПЛК)
- Использование **Unit Manufacturing Information** (Сведения об изготовлении модуля) (этот способ можно использовать также для специальных модулей ввода/вывода и модулей шины CPU).

**Примечание** Для определения версий модулей нельзя использовать CX-Programmer V3.3 или более старый.

### Сведения о ПЛК

- Если вам известен тип устройства и тип CPU, выберите их в диалоговом окне **Change PLC** (Изменить ПЛК), установите связь с ПЛК (режим on-line) и выберите Меню **PLC - Edit - Information** (ПЛК - Правка - Сведения).
- Если вам не известен тип устройства и тип CPU, но ваш компьютер подключен непосредственно к модулю CPU через последовательный интерфейс, выберите **PLC - Auto Online** (ПЛК - Автоматическое установление связи), чтобы установить связь с устройством, после чего выберите в меню команду **PLC - Edit - Information** (ПЛК - Правка - Сведения).

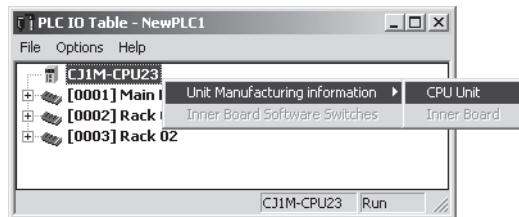
В любом случае будет отображено диалоговое окно PLC Information (Сведения о ПЛК).



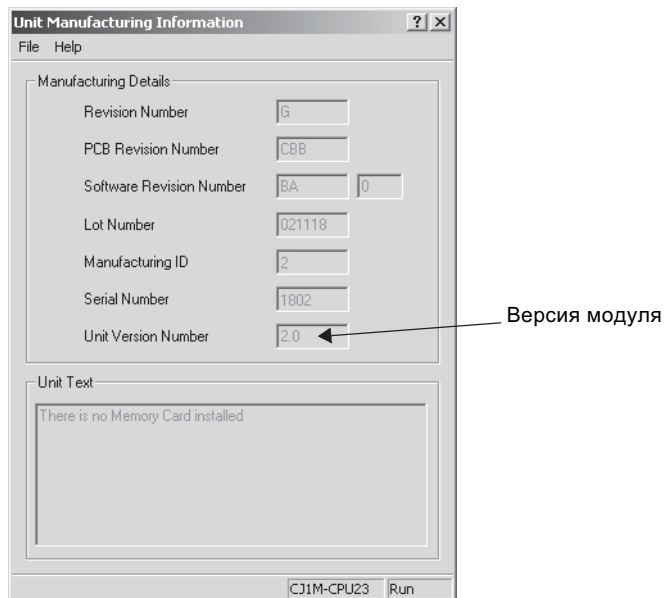
С помощью представленного выше окна определите версию модуля CPU.

### Сведения об изготовлении модуля

В окне *IO Table* (Таблица ввода/вывода) щелкните правой кнопкой мыши и выберите пункт **Unit Manufacturing information - CPU Unit** (Сведения об изготовлении модуля - Модуль CPU).



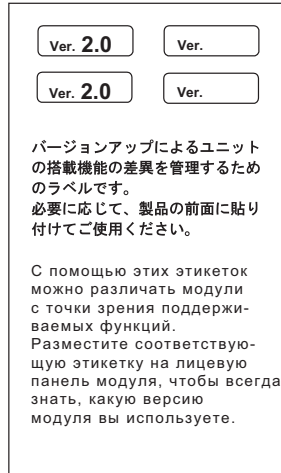
Будет отображено диалоговое окно Unit Manufacturing information (Сведения об изготовлении модуля).



С помощью представленного выше окна определите версию модуля CPU, с которым установлена связь.

**Использование этикеток с указанием версии модуля**

Вместе с модулем CPU поставляются следующие этикетки, на которых указывается версия модуля.



Эти этикетки можно прикреплять к лицевой панели модулей CPU предшествующих моделей, чтобы отличать модули CPU различных версий.

**Указание версии модуля**

В настоящем Руководстве информация о версии модуля CPU приводится в соответствии с таблицей ниже.

Паспортная табличка изделия	Модули CPU, для которых версия не указывается	Модули CPU, для которых версия указывается (Ver. □.□)
<p><b>Значение</b></p> <p>Обозначение отдельных модулей CPU (например, CS1H-CPU67H)</p> <p>Обозначение групп модулей CPU (например, модули CPU CS1-H)</p> <p>Обозначение отдельной серии модулей CPU (например, модули CPU серии CS)</p>	<p>Модули CPU CS1-H до версии 2.0</p> <p>Модули CPU CS1-H до версии 2.0</p> <p>Модули CPU CS до версии 2.0</p>	<p>Модуль CPU CS1H-CPU67H, версия □.□</p> <p>Модули CPU CS1H, версия □.□</p> <p>Модули CPU серии CS, версия □.□</p>

## Версии модулей и номера партий

Серия	Модель		Дата изготовления						
			Ранее	Сент. 2003	Окт. 2003	Нб. 2003	Дек. 2003	Позже	
Серия CS	Модули CPU CS1	CS1□-CPU□□							
		Без указания версии							
	Модули CPU CS1-V1	CS1□-CPU□□-V1							
		Без указания версии							
	Модули CPU CS1-H	CS1□-CPU□□H							
		Модули CPU до версии 2.0			Модули CPU версии 2.0 (Номер партии: 031105 и далее)				
Модули CPU CS1D	Модули CPU для систем с дублированием CPU	CS1D-CPU□□H							
		Модули CPU до версии 1.1			Модули CPU версии 1.1 (Номер партии: 031120 и далее)				
	Модули CPU для систем с одним CPU	CS1D-CPU□□S					Модули CPU версии 2.0 (№ партии: 031215 и далее)		
Серия CJ	Модули CPU CJ1	CJ1G-CPU□□							
		Модули CPU до версии 2.0							
	Модули CPU CJ1-H	CJ1□-CPU□□H							
		Модули CPU до версии 2.0			Модули CPU версии 2.0 (Номер партии: 031105 и далее)				
	Модули CPU CJ1M, за исключением моделей низшего класса	CJ1M-CPU□□							
		Модули CPU до версии 2.0			Модули CPU версии 2.0 (Номер партии: 031105 и далее)				
Модули CPU CJ1M, модели низшего класса	CJ1M-CPU11/21				Модули CPU версии 2.0 (Номер партии: 031002 и далее)				
ПО поддержки	CX-Programmer	WS02-CXPC1-EV□							
			Версия 3.2		Версия 3.3		Версия 4.0		



## Функции, поддерживаемые модулями различных версий

### Модули CPU CS1-H (CS1□-CPU□□H)

Функция		Версия модуля	
		Модули CPU до версии 2.0	Модули CPU версии 2.0
Загрузка и считывание отдельных задач		---	Поддерживается
Улучшенная защита от чтения с помощью паролей		---	Поддерживается
Защита от записи посредством команд FINS, поступающих на модули CPU через сеть		---	Поддерживается
Сетевые (Online) соединения без таблиц ввода/вывода		---	Поддерживается
Межсетевые коммуникации (макс. 8 сетевых уровней)		---	Поддерживается
Online-соединение с ПЛК через программируемые терминалы серии NS		Поддерж., начиная с партии 030201	Поддерживается
Выбор первого слова для слота		Поддерж., максимум для 8-ми групп	Поддерж., максимум для 64 групп
Автоматич. загрузка после включ. питания без файла параметров		---	Поддерживается
Автоматическое определение способа назначения входов/выходов для автоматической загрузки после включения питания		---	---
Значения времени запуска/прекращения работы		---	Поддерживается
Новые прикладные команды	MILH, MILR, MILC	---	Поддерживается
	=DT, <>DT, <DT, <=DT, >DT, >=DT	---	Поддерживается
	BCMP2	---	Поддерживается
	GR	Поддерж., начиная с партии 030201	Поддерживается
	TPO	---	Поддерживается
	DSW, TKY, НКУ, MTR, 7SEG	---	Поддерживается
	EXPLT, EGATR, ESATR, ECHRD, ECHWR	---	Поддерживается
	Чтение/запись из/в модули шины CPU с помощью IORD/IOWR	Поддерж., начиная с партии 030418	Поддерживается
	PRV2	---	---

## Модули CPU CS1D

Функция		Модули CPU CS1D для систем с дублированием CPU (CS1D-CPU□□H)		Модули CPU CS1D для однопроцес- сорных систем (CS1D-CPU□□S)
		Модули CPU до версии 1.1	Модули CPU версии 1.1	Модули CPU версии 2.0
Функции, которыми обладают только модули CPU CS1D	Модули с дублированием CPU	Поддерживается	Поддерживается	---
	Замена модулей в режиме Online	Поддерживается	Поддерживается	Поддерживается
	Модули с дублированием источника питания	Поддерживается	Поддерживается	Поддерживается
	Модули с дублированием Controller Link	Поддерживается	Поддерживается	Поддерживается
	Модули с дублированием Ethernet	---	Поддерживается	Поддерживается
Загрузка и считывание отдельных задач		---	---	Поддерживается
Улучшенная защита от чтения с помощью паролей		---	---	Поддерживается
Защита от записи посредством команд FINS, поступающих на модули CPU через сеть		---	---	Поддерживается
Сетевые Online-соединения без таблиц ввода/ вывода		---	---	Поддерживается
Межсетевые коммуникации (максимум 8 сетевых уровней)		---	---	Поддерживается
Online-соединение с ПЛК через программируемые терминалы серии NS		---	---	Поддерживается
Выбор первого слова для слота		---	---	Поддерживается макс. для 64 групп
Автоматическая загрузка после включения питания без файла параметров		---	---	Поддерживается
Автоматическое определение способа назначения входов/выходов для автоматической загрузки после включения питания		---	---	---
Значения времени запуска/прекращения работы		---	Поддерживается	Поддерживается
Новые прикладные команды	MILH, MILR, MILC	---	---	Поддерживается
	=DT, <>DT, <DT, <=DT, >DT, >=DT	---	---	Поддерживается
	BCMP2	---	---	Поддерживается
	GRY	---	---	Поддерживается
	TPO	---	---	Поддерживается
	DSW, TKY, HKY, MTR, 7SEG	---	---	Поддерживается
	EXPLT, EGATR, ESATR, ECHRД, ECHWR	---	---	Поддерживается
	Чтение/запись из/в модули шины CPU с IORD/IOWR	---	---	Поддерживается
PRV2	---	---	---	

## Модули CPU CJ1-H/CJ1M

Функция	Модули CPU CJ1-H (CJ1□-CPU□□H)		Модули CPU CJ1M, за исключением моделей нижнего класса (CJ1M-CPU□□)		Модули CPU CJ1M, модели нижнего класса (CJ1M- CPU11/21)	
	Модули CPU до версии 2.0	Модули CPU версии 2.0	Модули CPU до версии 2.0	Модули CPU версии 2.0	Модули CPU версии 2.0	
Загрузка и считывание отдельных задач	---	Поддерживается	---	Поддерживается	Поддерживается	
Улучшенная защита от чтения с помощью паролей	---	Поддерживается	---	Поддерживается	Поддерживается	
Защита от записи посредством команд FINS, поступающих на модули CPU через сеть	---	Поддерживается	---	Поддерживается	Поддерживается	
Сетевые Online-соединения без таблиц ввода/вывода	Поддерживается, но только если задано распределение таблицы ввода/вывода при включении питания	Поддерживается	Поддерживается, но только если задано распределение таблицы ввода/вывода при включении питания	Поддерживается	Поддерживается	
Межсетевые коммуникации (максимум 8 сетевых уровней)	Поддерживается макс. для 8 групп	Поддерживается макс. для 64 групп	Поддерживается макс. для 8 групп	Поддерживается макс. для 64 групп	Поддерживается макс. для 64 групп	
Online-соединение с ПЛК через программируемые терминалы серии NS	Поддерживается, начиная с номера партии 030201	Поддерживается	Поддерживается, начиная с номера партии 030201	Поддерживается	Поддерживается	
Выбор слов для 1-го слота	---	Поддерживается	---	Поддерживается	Поддерживается	
Автоматическая загрузка после включения питания без файла параметров	---	Поддерживается	---	Поддерживается	Поддерживается	
Автоматическое определение способа назначения входов/выходов для автоматической загрузки после включения питания	---	Поддерживается	---	Поддерживается	Поддерживается	
Значения времени запуска/прекращения работы	---	Поддерживается	---	Поддерживается	Поддерживается	
Новые прикладные команды	MILH, MILR, MILC	---	Поддерживается	---	Поддерживается	Поддерживается
	=DT, <>DT, <DT, <=DT, >DT, >=DT	---	Поддерживается	---	Поддерживается	Поддерживается
	BCMP2	---	Поддерживается	---	Поддерживается	Поддерживается
	GRY	Поддерживается, начиная с номера партии 030201	Поддерживается	Поддерживается, начиная с номера партии 030201	Поддерживается	Поддерживается
	TPO	---	Поддерживается	---	Поддерживается	Поддерживается
	DSW, TKY, HKY, MTR, 7SEG	---	Поддерживается	---	Поддерживается	Поддерживается
	EXPLT, EGATR, ESATR, ECHRD, ECHWR	---	Поддерживается	---	Поддерживается	Поддерживается
	Чтение/запись из/в модули шины CPU с помощью IORD/IOWR	---	Поддерживается	---	Поддерживается	Поддерживается
PRV2	---	---	---	Поддерживается, но только для моделей со встроенными входами/выходами	Поддерживается, но только для моделей со встроенными входами/выходами	

## Версии модулей и средства программирования

Использование функций, которые появились в модулях CPU версии 2.0, возможно только в случае применения CX-Programmer версии 4.0 или выше. В следующих таблицах показано, какие версии модулей с какими версиями CX-Programmer можно использовать.

### Версии модулей и средства программирования

Модуль CPU	Функции		CX-Programmer			Консоль программирования
			Версия 3.2 или более старая	Версия 3.3	Версия 4.0 или более новая	
Модули CPU CJ1M, модели низшего класса, версия 2.0	Функции, появившиеся в модулях версии 2.0	Новые функции применяются	---	---	Поддерживается	Ограничений нет
		Новые функции не применяются	---	Поддерживается	Поддерживается	
Модули CPU CS1-H, CJ1-H и CJ1M, за исключением моделей низшего класса, версия 2.0	Функции, появившиеся в модулях версии 2.0	Новые функции применяются	---	---	Поддерживается	
		Новые функции не применяются	Поддерживается	Поддерживается	Поддерживается	
Модули CPU CS1D для однопроцессорных систем, версия 2.0	Функции, появившиеся в модулях версии 2.0	Новые функции применяются	---	---	Поддерживается	
		Новые функции не применяются				
Модули CPU CS1D для систем с дублированием CPU, версия 1	Функции, появившиеся в модулях версии 1.1	Новые функции применяются	---	---	Поддерживается	
		Новые функции не применяются	Поддерживается	Поддерживается	Поддерживается	


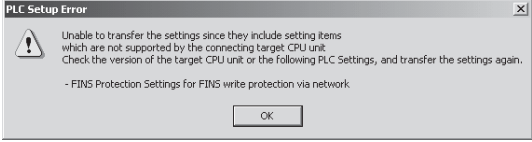
**Примечание** Как показано в таблице выше, CX-Programmer не требуется обновлять до версии 4.0, если функции, добавленные в модули версии 2.0 или модули версии 1.1, использовать не предполагается.

### Выбор типа устройства

Версия модуля не влияет на выбор типа устройства в CX-Programmer. Независимо от версии модуля CPU тип устройства следует выбрать в соответствии с таблицей ниже.

Серия	Группа модулей CPU	Модель модуля CPU	Выбираемый тип устройства в CX-Programmer версии 4.0 или выше
Серия CS	Модули CPU CS1-H	CS1G-CPU□□H	CS1G-H
		CS1H-CPU□□H	CS1H-H
	Модули CPU CS1D для систем с дублированием CPU	CS1D-CPU□□H	CS1D-H (или CS1H-H)
	Модули CPU CS1D для однопроцессорных систем	CS1D-CPU□□S	CS1D-S
Серия CJ	Модули CPU CJ1-H	CJ1G-CPU□□H	CJ1G-H
		CJ1H-CPU□□H	CJ1H-H
	Модули CPU CJ1M	CJ1M-CPU□□	CJ1M

## Поиск и устранение ошибок, связанных с версиями модулей, в CX-Programmer

Проблема	Причина	Решение
 <p>После отображения указанного сообщения в закладке <i>Compile</i> (Компиляция) окна вывода будет отображена ошибка компиляции.</p>	<p>Предпринята попытка применения CX-Programmer версии 4.0 или выше для загрузки программы, содержащей команды, которые поддерживаются только модулями CPU версии 2.0 или выше, в модули CPU до версии 2.0.</p>	<p>Проверьте программу или замените модуль CPU, в который выполняется загрузка, на модуль CPU версии 2.0 или выше.</p>
	<p>Предпринята попытка применения CX-Programmer версии 4.0 или выше для загрузки параметров ПЛК, среди которых имеются параметры, поддерживаемые только модулями CPU версии 2.0 или выше (то есть, не со своими стандартными значениями), в модули CPU версии до 2.0.</p>	<p>Проверьте параметры в настройках ПЛК или замените модуль CPU, в который выполняется загрузка, на модуль CPU версии 2.0 или более новой.</p>
<p>В программе, считанной из ПЛК CX-Programmer, отображается "????".</p>	<p>CX-Programmer версии 3.3 или ниже был применен для считывания программы, содержащей команды, которые поддерживаются только модулями CPU версии 2.0 или выше, из модуля CPU версии 2.0 или выше.</p>	<p>Новые команды не могут быть считаны с помощью CX-Programmer версии 3.3 или ниже. Используйте CX-Programmer версии 4.0 или выше.</p>

# ОГЛАВЛЕНИЕ

<b>ПРЕДВАРИТЕЛЬНЫЕ УКАЗАНИЯ</b> .....	<b>xix</b>
1 Для кого предназначено Руководство .....	xx
2 Общие предварительные указания .....	xx
3 Указания по безопасности .....	xx
4 Указания по условиям эксплуатации .....	xxii
5 Указания по применению .....	xxii
6 Соответствие Директивам ЕС .....	xxvi
<b>РАЗДЕЛ 1</b>	
<b>Функционирование модуля CPU</b> .....	<b>1</b>
1-1 Первоначальная настройка (только для модулей CPU CS1) .....	2
1-2 Применение внутренних часов (только для модулей CPU CS1) .....	5
1-3 Внутренняя структура модулей CPU .....	6
1-4 Режимы работы .....	8
1-5 Программы и задачи .....	12
1-6 Описание задач .....	14
<b>РАЗДЕЛ 2</b>	
<b>Программирование</b> .....	<b>21</b>
2-1 Основные принципы .....	22
2-2 Замечания .....	57
2-3 Проверка программ .....	66
<b>РАЗДЕЛ 3</b>	
<b>Команды и их функции</b> .....	<b>73</b>
3-1 Команды последовательного ввода .....	74
3-2 Команды последовательного вывода .....	76
3-3 Команды последовательного управления .....	79
3-4 Команды управления таймерами и счетчиками .....	83
3-5 Команды сравнения .....	87
3-6 Команды перемещения данных .....	91
3-7 Команды сдвига данных .....	94
3-8 Команды увеличения/уменьшения .....	98
3-9 Символьные математические команды .....	99
3-10 Команды преобразования .....	104
3-11 Логические команды .....	110
3-12 Специальные математические команды .....	112
3-13 Команды математических операций с плавающей запятой .....	113
3-14 Команды математических операций с плавающей запятой, двойной точности .....	117
3-15 Команды обработки табличных данных .....	121
3-16 Команды управления данными .....	125
3-17 Команды управления подпрограммами .....	129
3-18 Команды управления прерываниями .....	130
3-19 Команды управления высокоскоростными счетчиками и импульсными выходами (только CJ1M-CPU21/22/23) .....	132
3-20 Шаговые команды .....	134
3-21 Команды для базовых модулей ввода/вывода .....	134
3-22 Команды последовательного интерфейса .....	137
3-23 Сетевые команды .....	138
3-24 Команды памяти файлов .....	140
3-25 Команды дисплея .....	141
3-26 Команды управления часами .....	141

# ОГЛАВЛЕНИЕ

3-27	Команды отладки программы . . . . .	142
3-28	Команды для диагностики неисправностей . . . . .	143
3-29	Прочие команды . . . . .	144
3-30	Команды для программирования блоков . . . . .	145
3-31	Команды для обработки текстовых строк . . . . .	151
3-32	Команды управления задачами. . . . .	154
<b>РАЗДЕЛ 4</b>		
<b>Задачи . . . . .</b>		<b>155</b>
4-1	Свойства задач. . . . .	156
4-2	Применение задач . . . . .	165
4-3	Задачи обработки прерываний. . . . .	175
4-4	Использование средств программирования для задач. . . . .	187
<b>РАЗДЕЛ 5</b>		
<b>Функции памяти файлов. . . . .</b>		<b>189</b>
5-1	Память файлов. . . . .	190
5-2	Управление файлами. . . . .	208
5-3	Применение памяти файлов . . . . .	240
<b>РАЗДЕЛ 6</b>		
<b>Дополнительные функции . . . . .</b>		<b>247</b>
6-1	Длительность цикла/Скоростная обработка . . . . .	249
6-2	Регистры индексов . . . . .	266
6-3	Обмен данными через последовательные интерфейсы . . . . .	276
6-4	Изменение режима обновления текущего значения (PV) таймеров/счетчиков. . . . .	291
6-5	Применение запланированного прерывания в качестве высокоточного таймера. . . . .	299
6-6	Настройка параметров запуска и техническое обслуживание. . . . .	301
6-7	Функции диагностики. . . . .	312
6-8	Режимы работы CPU. . . . .	317
6-9	Режим приоритетного обслуживания периферии. . . . .	322
6-10	Работа без батареи. . . . .	328
6-11	Прочие функции . . . . .	330
<b>РАЗДЕЛ 7</b>		
<b>Загрузка программы, пробный запуск и отладка. . . . .</b>		<b>333</b>
7-1	Загрузка программы . . . . .	334
7-2	Пробный запуск и отладка . . . . .	334
<b>Приложения</b>		
A	Приложение А Таблицы сравнения ПЛК: ПЛК серии CJ, серии CS, C200HG/HE/HX, CQM1H, CVM1 и серии CV . . . . .	343
B	Приложение В Изменения по сравнению с предшествующими системами Host Link . . . . .	367
<b>Предметный указатель. . . . .</b>		<b>371</b>
<b>Перечень версий . . . . .</b>		<b>377</b>

## О данном Руководстве:

В настоящем Руководстве описывается программирование модулей CPU для программируемых контроллеров серии CS/CJ (ПЛК). Классификация изделий серии CS и серии CJ представлена в таблице ниже. Под таблицей перечислены Разделы, содержащиеся в настоящем Руководстве.

Модуль	Серия CS	Серия CJ
Модули CPU	Модули CPU CS1-H: CS1H-CPU□□H CS1G-CPU□□H	Модули CPU CJ1-H: CJ1H-CPU□□H CJ1G-CPU□□H
	Модули CPU CS1: CS1H-CPU□□-EV1 CS1G-CPU□□-EV1	Модули CPU CJ1: CJ1G-CPU□□-EV1 Модули CPU CJ1M: CJ1M-CPU□□
	Модули CPU CS1D: Модули CPU CS1D для систем с дублированием CPU: CS1D-CPU□□H Модули CPU CS1D для однопроцессорных систем: CS1D-CPU□□S Модули CPU CS1D для управления процессами: CS1D-CPU□□P	
Базовые модули ввода/вывода	Базовые модули ввода/вывода серии CS	Базовые модули ввода/вывода серии CJ
Специальные модули ввода/вывода	Специальные модули ввода/вывода серии CS	Специальные модули ввода/вывода серии CJ
Модули шины CPU	Модули шины CPU серии CS	Модули шины CPU серии CJ
Модули источников питания	Модули источников питания серии CS	Модули источников питания серии CJ

Пожалуйста, внимательно прочитайте настоящее Руководство, а также все связанные с ним руководства, перечисленные в таблице на следующей странице. Приступайте к монтажу или эксплуатации модулей CPU серии CS/CJ в составе ПЛК только после изучения информации, содержащейся в настоящем Руководстве.

Настоящее Руководство состоит из следующих разделов.

В **Разделе 1** описаны базовая структура и принципы работы модуля CPU.

В **Разделе 2** содержатся основные сведения, необходимые для выполнения записи, проверки и ввода программ.

В **Разделе 3** описаны команды, которые можно использовать для создания программ пользователя.

В **Разделе 4** описано применение задач.

В **Разделе 5** описаны функции, предназначенные для работы с памятью файлов.

В **Разделе 6** содержится подробное описание дополнительных функций: длительность цикла/скоростная обработка, регистры индексов, связь через последовательные интерфейсы, запуск и обслуживание, диагностика и отладка, средства программирования, а также настройка времени срабатывания входов базовых модулей ввода/вывода серии CJ.

В **Разделе 7** описаны процедуры, которые используются для загрузки программ в модуль CPU, а также функции, которые можно применять для тестирования и отладки программ.

В **Приложениях** приведено сравнение серий CS и CJ, перечислены ограничения на использование специальных модулей ввода/вывода C200H, а также изменения, произошедшие в системах Host Link.



## О данном Руководстве (продолжение)

Название	№ по каталогу	Содержание
Руководство по программированию программируемых контроллеров SYSMAC CS/CJ Series CS1G/H-CPU□□-EV1, CS1G/H-CPU□□H, CS1D-CPU□□H, CS1D-CPU□□S, CJ1G-CPU□□, CJ1G/H-CPU□□H, CJ1M-CPU□□ Programmable Controllers Programming Manual	W394	Описано программирование и способы использования функций ПЛК серии CS/CJ (это Руководство)
Руководство по эксплуатации программируемых контроллеров SYSMAC CS Series CS1G/H-CPU□□-EV1, CS1G/H-CPU□□H Programmable Controllers Operation Manual	W339	Описано проектирование, монтаж, обслуживание и другие основные процедуры для ПЛК серии CS.
Руководство по эксплуатации программируемых контроллеров SYSMAC CJ Series CJ1G-CPU□□, CJ1G/H-CPU□□H, CJ1M-CPU□□ Programmable Controllers Operation Manual	W393	Описано проектирование, монтаж, обслуживание и другие основные процедуры для ПЛК серии CJ.
Руководство по использованию функций встроенных входов/выходов SYSMAC CJ Series CJ1M-CPU21/22/23 Built-in I/O Functions Operation Manual	W395	Описаны функции встроенных входов/выходов для модулей CPU CJ1M.
Руководство по эксплуатации дублированных систем SYSMAC CS Series CS1D-CPU□□H CPU Units CS1D-CPU□□S CPU Units CS1D-DPL01 Duplex Unit CS1D-PA207R Power Supply Unit Duplex System Operation Manual	W405	Содержит описание проектирования, монтажа, обслуживания и другие основные операции для дублированных систем, создаваемых на базе модулей CPU серии CS1D.
Справочное Руководство по командам программируемых контроллеров SYSMAC CS/CJ Series CS1G/H-CPU□□-EV1, CS1G/H-CPU□□H, CS1D-CPU□□H, CS1D-CPU□□S, CJ1G-CPU□□, CJ1G/H-CPU□□H, CJ1M-CPU□□ Programmable Controllers Instructions Reference Manual	W340	Описаны команды для программирования "лестничных диаграмм", поддерживаемые ПЛК серии CS/CJ
Руководство по эксплуатации консолей программирования SYSMAC CS/CJ Series CQM1H-PRO01-E, C200H-PRO27-E, CQM1-PRO01-E Programming Consoles Operation Manual	W341	Содержатся сведения о том, как создавать программы и управлять ПЛК серии CS/CJ с помощью консоли программирования.
Справочное руководство по командам связи SYSMAC CS/CJ Series CS1G/H-CPU□□-EV1, CS1G/H-CPU□□H, CS1D-CPU□□H, CS1D-CPU□□S, CJ1G-CPU□□, CJ1G/H-CPU□□H, CJ1M-CPU□□, CS1W-SCB21-V1/41-V1, CS1W-SCU21-V1, CJ1W-SCU21/41 Communications Commands Reference Manual	W342	Описаны коммуникационные команды системы Host Link (серия C) и FINS, которые используются в ПЛК серии CS/CJ.
Руководство по эксплуатации CX-Programmer версии 3.□ SYSMAC WS02-CXP□□-E CX-Programmer Operation Manual Version 3.□	W414	Содержит сведения о том, как использовать CX-Programmer, средства программирования, которые поддерживают ПЛК серии CS/CJ, а также CX-Net, входящий в состав CX-Programmer.
Руководство по эксплуатации CX-Programmer версии 4.□ SYSMAC WS02-CXP□□-E CX-Programmer Operation Manual Version 4.□	W425	
Руководство по эксплуатации плат/модулей последовательного интерфейса SYSMAC CS/CJ Series CS1W-SCB21-V1/41-V1, CS1W-SCU21-V1, CJ1W-SCU21/41 Serial Communications Boards/Units Operation Manual	W336	Описано использование модуля и плат последовательного интерфейса для осуществления последовательных коммуникаций с внешними устройствами, включая применение стандартных системных протоколов для изделий OMRON.
Руководство по использованию CX-Protocol SYSMAC WS02-PSTC1-E CX-Protocol Operation Manual	W344	Описано использование CX-Protocol с целью создания протокольных макросов для осуществления коммуникационных последовательностей с целью обмена данными с внешними устройствами.



### **ВНИМАНИЕ**

Пренебрежение чтением и пониманием сведений, содержащихся в данном руководстве, может привести к травмированию персонала, возможно, со смертельным исходом, а также к повреждению изделия или выходу его из строя. Прочитайте, пожалуйста, каждый раздел целиком и удостоверьтесь в том, что сведения, содержащиеся в разделе, в разделах, с ним связанных, понимаются вами правильно, прежде чем приступать к любой из описанных операций или действий.

# ПРЕДВАРИТЕЛЬНЫЕ УКАЗАНИЯ

В данном разделе содержатся общие указания по использованию программируемых контроллеров (ПЛК) серии CS/CJ и связанных с ними устройств.

**Данный раздел содержит важную информацию о безотказном и безопасном применении программируемых логических контроллеров. Обязательно прочитайте этот раздел и примите к сведению всю содержащуюся в нем информацию, прежде чем приступать к настройке или использованию системы ПЛК.**

1	Для кого предназначено Руководство . . . . .	xx
2	Общие предварительные указания. . . . .	xx
3	Указания по безопасности. . . . .	xx
4	Указания по условиям эксплуатации. . . . .	xxii
5	Указания по применению . . . . .	xxii
6	Соответствие Директивам ЕС. . . . .	xxvi
6-1	Соблюдаемые Директивы . . . . .	xxvi
6-2	Содержание Директив . . . . .	xxvi
6-3	Соответствие Директивам ЕС . . . . .	xxvii
6-4	Способы подавления помех на релейных выходах . . . . .	xxvii

## 1 Для кого предназначено руководство

Данное руководство предназначено для лиц, обладающих специальными знаниями в области электрических систем (инженер-электрик и т.п.)

- Персонал, ответственный за установку систем автоматизации.
- Персонал, ответственный за разработку систем автоматизации.
- Персонал, ответственный за администрирование оборудования систем автоматизации.

## 2 Общие предварительные указания

Пользователь должен применять изделие в соответствии с эксплуатационными характеристиками, описанными в руководствах по эксплуатации.

Прежде чем использовать изделие в условиях, которые не описаны в руководстве, а также при применении изделия в системах управления на объектах атомной энергетики, в железнодорожных системах, в авиации, в транспортных средствах, в теплотехнике, в медицинском оборудовании, в игровых автоматах, в защитном оборудовании и других системах, машинах и установках, которые могут серьезно повлиять на здоровье людей и привести к повреждению имущества при условии неправильной эксплуатации, обязательно проконсультируйтесь в представительстве OMRON своего региона.

Убедитесь в том, что номинальные значения и рабочие характеристики изделия достаточны для систем, машин и оборудования, и предусматривайте в системах, машинах и оборудовании механизмы удвоенной надежности.

В данном руководстве содержатся сведения по программированию и эксплуатации модуля. Прежде чем приступить к использованию модуля, обязательно прочтите данное руководство, и держите его под рукой, чтобы использовать во время работы.

### ВНИМАНИЕ

Очень важно, чтобы ПЛК и все его модули использовались только для оговоренных целей и только в условиях, указанных в технических руководствах, особенно в тех приложениях, в которых они могут прямым или косвенным образом повлиять на здоровье человека. Прежде чем применять ПЛК системы в описанных выше приложениях, необходимо проконсультироваться в представительстве OMRON.

## 3 Указания по безопасности

### ВНИМАНИЕ




Обновление сигналов ввода/вывода выполняется модулем CPU даже тогда, когда программа в нем остановлена (т.е., даже в режиме PROGRAM). Прежде чем изменять состояние любой области памяти, отведенной для модулей ввода/вывода, специальных модулей ввода/вывода или модулей шины CPU, следует заранее обеспечить безопасные условия. Любое изменение данных, отведенных для любого модуля, может привести к работе нагрузок, подсоединенных к модулю, в непредусмотренном режиме. Ниже перечислены операции, которые могут привести к изменению состояния памяти.

- Загрузка данных памяти ввода/вывода в модуль CPU из программатора.
- Изменение текущих значений в памяти с помощью программатора.
- Принудительная установка/сброс битов с помощью программатора.
- Загрузка файлов памяти ввода/вывода из карты памяти или файловой памяти EM в модуль CPU.
- Загрузка памяти ввода/вывода из центрального компьютера или другого ПЛК сети.




### ВНИМАНИЕ

Никогда не пытайтесь разбирать модуль, когда на него подано напряжение. Это может привести к серьезному поражению током.

- ⚠ ВНИМАНИЕ** Никогда не касайтесь клемм или клеммных колодок, когда на модуль подано напряжение. Это может привести к поражению током.
- ⚠ ВНИМАНИЕ** Не пытайтесь производить разборку, ремонт или модификацию какого-либо модуля. Это может привести к выходу из строя, воспламенению или поражению током.
- ⚠ ВНИМАНИЕ** Во внешних цепях необходимо предусматривать дополнительные меры защиты, помимо предусмотренных в контроллере, чтобы обеспечить защиту системы в случае возникновения нештатной ситуации из-за неисправности в ПЛК или из-за других факторов, влияющих на работу ПЛК. Невыполнение этого требования может привести к серьезным последствиям.
- Во внешних схемах управления должны быть предусмотрены устройства аварийного останова, блокировки, ограничительные устройства и другие меры безопасности.
  - В случае обнаружения функцией самодиагностики какой-либо ошибки, а также при выполнении команды FALS (авария из-за серьезной неисправности), ПЛК произведет отключение всех выходов. На случай таких ситуаций во внешних схемах должны быть предусмотрены предохранительные устройства, обеспечивающие безопасность в системе.
  - Выходы ПЛК могут оставаться включенными или отключенными из-за металлизации или выгорания релейных выходов или выхода из строя выходных транзисторов. На случай таких неисправностей во внешних цепях должны быть предусмотрены предохранительные устройства, обеспечивающие безопасность в системе.
  - Перегрузка или короткое замыкание на выходе = 24 В (служебный источник питания ПЛК) могут привести к понижению напряжения, в результате чего выходы могут отключиться. Для защиты от такой ситуации следует предусмотреть внешние меры, обеспечивающие безопасность в системе.
- ⚠ Предупреждение** Прежде чем загружать файлы данных, хранящиеся в памяти файлов (в карте памяти или в файловой памяти EM), в область ввода/вывода (CIO) модуля CPU с помощью внешних (периферийных) средств, следует убедиться в безопасности этой операции. В противном случае устройства, подключенные к модулю вывода, могут работать с ошибками, независимо от режима работы модуля CPU.
- ⚠ Предупреждение** Пользователем должны быть предусмотрены меры защиты для обеспечения безопасности в случае возникновения сигналов недопустимого уровня, в случае пропадания сигналов из-за обрыва в сигнальных линиях или в случае кратковременного пропадания питания.
- ⚠ Предупреждение** Модули CPU CS1-H, CJ1-H, CJ1M и CS1D автоматически сохраняют во флеш-память резервную копию программы пользователя и параметров при загрузке последних в модуль CPU. Однако содержимое памяти ввода/вывода (включая области DM, EM и HR) во флеш-память не сохраняется. Содержимое областей DM, EM и HR может быть сохранено при отключенном питании при наличии батареи. Но если батарея неисправна, содержимое этих областей может оказаться недостоверным после отключения питания. Если данные областей DM, EM и HR используются для управления внешними устройствами, необходимо предусмотреть блокировку выходов в случае, когда флаг "Ошибка батареи" (A40204) = ВКЛ.
- ⚠ Предупреждение** Редактирование в режиме online (при установленной связи) можно осуществлять лишь в том случае, если увеличение времени цикла не приведет к нежелательному воздействию на систему. В противном случае входные сигналы могут оказаться нечитаемыми.
- ⚠ Предупреждение** Прежде чем загружать программу в другой узел (модуль) или изменять содержимое области памяти ввода/вывода, следует убедиться в безопасности этой операции по месту нахождения этого узла. Любая из этих операций может привести к травмированию персонала, если не будет обеспечена надлежащая безопасность.

-  **Предупреждение** При завинчивании винтов клеммной колодки модуля питания переменного тока (AC) прикладываемое усилие (вращающий момент) должно соответствовать указанному в руководстве. Если винты завинчены слабо, в процессе эксплуатации может произойти возгорание или могут возникнуть неисправности.
-  **Предупреждение** Не касайтесь модуля питания, когда он включен, или сразу после его отключения. Модуль питания нагревается при работе, и вы можете получить ожог.
-  **Предупреждение** Соблюдайте осторожность при подключении персональных компьютеров или других периферийных устройств к ПЛК, в состав которого входит модуль без гальванической развязки (CS1W-CLK12/52(-V1) или CS1W-ETN01), подсоединенный к внешнему источнику питания. Если цепь 24 В внешнего источника питания и цепь 0 В заземлены, произойдет короткое замыкание. При подключении периферийного устройства к такому ПЛК следует либо заземлять цепь 0 В внешнего источника питания, либо не заземлять внешний источник питания вообще.

## 4 Указания по условиям эксплуатации

-  **Предупреждение** Не эксплуатируйте систему управления в следующих местах:
- В местах воздействия прямого солнечного света.
  - В местах, где температура окружающей среды или влажность не соответствуют требованиям технических характеристик.
  - В местах, подверженных образованию конденсации вследствие резких перепадов температуры.
  - В местах, подверженных воздействию коррозионных или воспламеняющихся газов.
  - В местах скопления пыли (особенно, металлического порошка) или солей.
  - В местах, подверженных воздействию воды, масла или химических реактивов.
  - В местах, подверженных воздействию ударов или вибрации.
-  **Предупреждение** При монтаже систем в перечисленных ниже местах следует принимать надлежащие защитные меры:
- В местах воздействия статического электричества или любых других помех.
  - В местах воздействия интенсивного электромагнитного поля.
  - В местах воздействия радиоактивных излучений.
  - Вблизи источников электропитания или линий электропередачи.
-  **Предупреждение** Условия эксплуатации системы ПЛК могут оказать значительное влияние на срок службы и надежность системы. Не соответствующие требованиям условия эксплуатации могут привести к выходу из строя, к сбоям или другим непредвиденным проблемам в системе ПЛК. Необходимо следить за тем, чтобы условия эксплуатации соблюдались при монтаже системы, а также поддерживались в пределах установленных значений во время работы системы. Следуйте всем указаниям по монтажу и эксплуатации, приведенным в руководствах по эксплуатации.

## 5 Указания по применению

При использовании системы ПЛК соблюдайте следующие указания.

- Для написания программ с несколькими циклическими задачами следует использовать CX-Programmer (программное обеспечение, работающее под Windows). Консоль программирования позволяет создать лишь одну циклическую задачу и задачи обработки прерываний. С другой стороны, консоль программирования можно использовать для редактирования многозадачных программ, ранее созданных в CX-Programmer.

**ВНИМАНИЕ**

Всегда соблюдайте приведенные ниже указания. Несоблюдение этих указаний может привести к нанесению серьезных травм персоналу, возможно, со смертельным исходом.

- При монтаже модулей всегда должно выполняться заземление через цепь с сопротивлением менее 100 Ом. Невыполнение этого требования может привести к поражению током.
- При замыкании клемм GR и LG в модуле питания должно быть выполнено заземление через цепь с сопротивлением 100 Ом или меньше.
- Перед тем как выполнить одно из следующих действий, отключите напряжения питания ПЛК. Невыполнение этого требования может привести к выходу из строя оборудования или поражению током.
  - Монтаж или демонтаж модулей питания, модулей ввода/вывода, модулей CPU, встраиваемых плат или любых других модулей.
  - Сборка модулей.
  - Настройка DIP- или поворотных переключателей
  - Подсоединение кабелей или выполнение проводных соединений в системе.
  - Подключение или отключение соединителей/разъемов.

**Предупреждение**

Несоблюдение следующих указаний может привести к сбоям при работе ПЛК или системы, а также к выходу из строя ПЛК или его модулей. Всегда соблюдайте данные указания.

- Копия программы пользователя и данных области параметров в модулях CPU CS1-H, CS1D, CJ1-H и CJ1M сохраняется в энергонезависимую встроенную флеш-память. Когда выполняется процедура резервного копирования, на передней панели модуля CPU светится индикатор ВКУР. В этот момент нельзя выключать питание модуля CPU, иначе данные сохранены не будут.
- Если модуль CPU серии CS1 применяется в первый раз, перед загрузкой программы следует вставить в него предусмотренную в комплекте батарею CS1W-BAT1 и обнулить с помощью средства программирования все области памяти. Если используются встроенные часы, после установки батареи следует включить питание и настроить часы либо с помощью средства программирования, либо с помощью команды DATE(735). Часы не будут работать, пока не будет установлено время.
- По умолчанию (после поставки с завода) в настройках ПЛК модулей CPU указано, что после запуска модуль CPU начинает работу в режиме, выбранном на консоли программирования. Если консоль программирования не подключена, модуль CPU серии CS1 начнет работу в режиме PROGRAM, а модули CPU серии CS1-H, CS1D, CJ1, CJ1-H или CJ1M будут работать в режиме RUN, т.е., сразу же начнется выполнение программы. Ни в коем случае не начинайте работу модулей, не убедившись в безопасности этой операции.
- Создавая файл AUTOEXEC.IOM с помощью средства программирования (консоли программирования или CX-Programmer) с целью автоматической загрузки данных при запуске, следует выбрать первый адрес для записи равным D20000 и следить за тем, чтобы объем записываемых данных не превысил границы области DM. Когда файл данных считывается из карты памяти при запуске, данные записываются в модуль CPU, начиная с адреса D20000, даже если при создании файла AUTOEXEC.IOM был указан другой адрес. Кроме того, если область данных превышена (что может произойти при использовании CX-Programmer), оставшиеся данные будут записаны в область EM.

- Перед подачей питания на систему управления сначала всегда следует включать питание ПЛК. Если питание ПЛК включается после подачи питания на систему управления, могут возникать кратковременные ошибки в сигналах системы управления, поскольку в момент включения питания ПЛК на выходных клеммах модулей дискретных выходов постоянного тока и других модулей наблюдается кратковременный сигнал уровня "1".
- Должны быть предусмотрены меры защиты для обеспечения безопасности в случае, когда выходы модулей вывода остаются включенными из-за выхода из строя внутренних цепей (реле, транзисторов и других элементов).
- Должны быть предусмотрены меры защиты для обеспечения безопасности в случае возникновения сигналов недопустимого уровня, в случае пропадания сигналов из-за обрыва в сигнальных линиях или в случае кратковременного пропадания питания.
- Во внешних схемах управления должны быть предусмотрены устройства аварийного останова, блокировки, ограничительные устройства и другие меры защиты.
- Не выключайте питание ПЛК в момент передачи данных, например, при чтении или записи из/в карту памяти. Также не следует извлекать карту памяти в момент, когда светится индикатор BUSY (Занято). Чтобы извлечь карту памяти, сначала нажмите кнопку питания карты памяти, дождитесь, когда погаснет индикатор BUSY, и только после этого извлекайте карту памяти.
- Если установлен бит удержания сигналов ввода/вывода, выходные сигналы ПЛК останутся в состоянии ВКЛ и будут сохранять свои прежние состояния при переходе ПЛК из режима RUN в режим MONITOR или PROGRAM. Необходимо предусмотреть, чтобы в таких случаях не возникали аварийные ситуации, связанные с исполнительными механизмами (нагрузкой) (если работа прекращается из-за фатальной ошибки, включая ошибки, вызываемые командой FALS(007), все выходы модуля вывода будут обнулены. Будет сохранено лишь внутреннее состояние выходов.)
- Для хранения содержимого областей DM, EM и HR в модуле CPU служит батарея. Если напряжение батареи падает, эти данные могут стереться. Следует предусмотреть меры на случай падения напряжения батареи, например, производить перезапись данных по флагу 'Ошибка батареи' (A40204).
- Если для питания ПЛК серии CS используется напряжение 200...240В переменного тока, обязательно следует снять перемычку с клемм выбора напряжения питания на модуле питания (за исключением модулей питания, поддерживающих широкий диапазон напряжений). Если напряжение ~200...~240В подается при установленной перемычке, модуль выйдет из строя.
- Используйте для модулей только те напряжения питания, которые указаны в руководствах по эксплуатации. Другие напряжения могут привести к повреждению или возгоранию.
- Примите надлежащие меры по обеспечению подачи питания требуемой мощности, с требуемым номинальным напряжением и частотой, особенно, при работе с нестабильными источниками питания. Такой источник может привести к сбоям во время работы.
- Предусматривайте внешние автоматические выключатели, а также другие устройства для защиты от коротких замыканий во внешней проводке. Недостаточные меры защиты от коротких замыканий могут привести к возгоранию.
- Не подавайте на входы модулей ввода напряжение, превышающее номинальное входное напряжение. Повышенное напряжение может привести к возгоранию.
- Не следует подключать к выходам модулей вывода чрезмерную нагрузку или подавать на них напряжение, превышающее нагрузочную способность выходов модуля. Повышенное напряжение и чрезмерная нагрузка могут привести к возгоранию.

- Всегда отключайте клемму функционального заземления при выполнении испытаний на электрическую прочность. Невыполнение этого требования может привести к возгоранию.
- Монтаж модулей должен выполняться надлежащим образом, с соблюдением инструкций в руководстве по эксплуатации. Неправильный монтаж модулей может привести к появлению ошибок при работе.
- Монтажные винты модулей и объединительных шин ПЛК серии CS должны затягиваться с соблюдением крутящего момента, указанного в соответствующих руководствах. Несоблюдение этого требования может привести к сбоям или выходу оборудования из строя.
- Монтажные винты, клеммные винты и винты соединительных разъемов кабелей должны затягиваться с соблюдением крутящего момента, указанного в соответствующих руководствах. Несоблюдение этого требования может привести к сбоям или выходу оборудования из строя.
- При подключении проводов не снимайте защитную этикетку, прикрепленную к модулю. Удаление этикетки может привести к попаданию в модуль посторонних предметов и возникновению сбоев при работе.
- По завершении выполнения проводных соединений удалите этикетку, чтобы избежать перегрева модуля. Перегрев модуля может явиться причиной сбоев во время работы.
- Используйте обжимные клеммы при выполнении проводных соединений. Не вставляйте скрученные многожильные провода без обжимных клемм. Подключение проводов без зажимных клемм может привести к возгоранию.
- Не допускайте ошибок при выполнении проводных соединений.
- Прежде чем включить напряжение питания, дважды проверьте все проводные соединения и положения переключателей. Проводные соединения, выполненные с ошибками, могут послужить причиной возгорания.
- Модули должны устанавливаться только после полной проверки клемных блоков и соединителей..
- Следите за тем, чтобы клеммные колодки, модули памяти, удлинительные кабели и другие изделия, снабженные механизмами фиксации, были надежно зафиксированы на своих местах. Неадекватная фиксация может привести к сбоям во время работы.
- Перед началом работы проверьте положения переключателей, содержание области DM и другие параметры. Невыполнение этого требования может привести к работе в непредусмотренном режиме.
- Проверьте правильность выполнения программы пользователя перед тем, как запустить ее на модуле в рабочем состоянии. Невыполнение этого требования может привести к работе в непредусмотренном режиме.
- Следите за тем, чтобы выполнение одной из следующих операций не привело к нежелательным последствиям для системы. Невыполнение этого требования может привести к непредусмотренному режиму работы.
  - Изменение режима работы ПЛК.
  - Принудительная установка/сброс любого бита в памяти.
  - Изменение текущего значения или любого слова, или любого установленного значения в памяти.
- Не тяните за кабели и не изгибайте их чрезмерно. В противном случае может произойти обрыв кабеля.
- Не размещайте поверх кабелей или других проводных линий какие-либо предметы. Это может привести к обрыву кабеля.
- Не используйте обычные кабели RS-232C персональных компьютеров. Следует пользоваться только специальными кабелями, перечисленными в настоящем руководстве, или изготавливать их самостоятельно согласно техническим характеристикам, приведенным в данном руководстве. Использование стандартных кабелей может привести к повреждению подключаемых устройств или модуля CPU.
- Никогда не подсоединяйте вывод 6 (напряжение питания 5В) порта RS-232C модуля CPU к какому-либо устройству, кроме NT-AL001 или адаптера CJ1W-CIF11. Внешнее устройство или модуль CPU могут выйти из строя.
- При замене элементов следите за тем, чтобы новый элемент подходил по номинальным характеристикам, иначе могут наблюдаться сбои, а также может произойти возгорание.



- Перед тем, как взять модуль, обязательно коснитесь заземленного металлического предмета, чтобы снять электростатический заряд. Несоблюдение этого требования может привести к возникновению сбоев или выходу оборудования из строя.
- При транспортировке или хранении печатных плат их необходимо заворачивать в антистатический материал для защиты их от статического электричества. Кроме того, должна соблюдаться надлежащая температура транспортировки и хранения.
- Не трогайте печатные платы или установленные на них элементы руками. В случае неаккуратного обращения можно порезаться об острые выводы элементов.
- Батарею нельзя разбирать, заряжать, нагревать или жечь. Нельзя замыкать батарею накоротко. Не подвергайте батарею сильным ударам. Невыполнение этих требований может привести к утечке, разрушению, нагреву или возгоранию батареи. Если вы уронили батарею на пол или каким-либо иным образом подвергли её чрезмерному механическому воздействию, немедленно замените её. Батареи, подвергшиеся удару, могут потечь при эксплуатации.
- Стандартами UL предусмотрено, что замена батареи должна осуществляться только персоналом, имеющим соответствующий опыт. Неподготовленный персонал к такой работе допускаться не должен.
- ПЛК серии CJ (?): после подключения модулей питания, модулей CPU, модулей ввода/вывода, специальных модулей ввода/вывода или модулей шины CPU друг к другу их следует зафиксировать с помощью защелок сверху и снизу модулей, переведя эти защелки в положение фиксации до щелчка. Если модули плохо закреплены, при работе могут наблюдаться ошибки.
- ПЛК серии CJ: с правой стороны крайнего модуля следует обязательно устанавливать крышку. Если крышка не установлена, ПЛК не будет работать должным образом.
- Ошибки в таблице логических связей или другие неправильные параметры могут привести к работе в непредусмотренном режиме. Даже если таблица логических связей и другие параметры настроены правильно, не следует устанавливать или разрывать соединение через логическую связь, не проверив прежде безопасность этой операции.
- При загрузке таблиц маршрутизации из средства программирования в модуль CPU модули шины CPU перезапускаются. Перезапуск этих модулей необходим для чтения и вступления в силу новых таблиц маршрутизации. Прежде чем допустить перезапуск модулей шины CPU, убедитесь в безопасности этой операции для системы.

## 6 Соответствие директивам ЕС

### 6-1 Соблюдаемые Директивы

- Директивы EMC (ЭМС)
- Директива по низкому напряжению

### 6-2 Содержание Директив

#### Директивы по ЭМС

Изделия OMRON, выполняющие требования Директив ЕС, также удовлетворяют соответствующим стандартам на ЭМС, что облегчает задачу их совместного использования с другими устройствами или применение всей системы в целом. Все выпущенные изделия протестированы на соответствие стандартам ЭМС (см. примечание ниже). В то же время, соответствие изделий стандартам системы, используемой покупателем, должно проверяться самим покупателем.

Относящиеся к ЭМС характеристики изделий OMRON, соответствующих Директивам ЕС, могут изменяться в зависимости от конфигурации, схемы соединений и прочих условий, связанных с оборудованием или панелью управления, в которые установлены изделия OMRON. Поэтому покупатель должен проводить финальное тестирование на соответствие этих изделий и всей системы в целом стандартам ЭМС.

**Примечание** Применяются следующие стандарты ЭМС (электромагнитная совместимость):

EMS (Электромагнитная восприимчивость):

Серия CS: EN61131-2 и EN61000-6-2

Серия CJ: EN61000-6-2

EMI (Электромагнитные помехи)::

EN61000-6-4

(Излучения: нормативы 10-м)

**Директива по низкому напряжению**

Всегда следите за тем, чтобы устройства, работающие с напряжениями ~50...~1000 В и =75...=1500 В, удовлетворяли стандартам безопасности, применяемым для ПЛК (EN61131-2).

## 6-3 Соответствие директивам ЕС

ПЛК серии CS/CJ соответствуют Директивам ЕС. Чтобы система или устройство, в котором будет использоваться ПЛК серии CS/CJ,, удовлетворяло Директивам ЕС, монтаж ПЛК необходимо выполнять следующим образом:

**1,2,3...**

ПЛК серии CS/CJ должен устанавливаться внутри панели управления.

Для источников постоянного тока, используемых для питания устройств связи и входов/выходов, следует применять усиленную или двойную изоляцию.

ПЛК серии CS/CJ, удовлетворяющие Директивам ЕС, также соответствуют Стандарту на общие излучения (EN50081-2). Характеристики по излучениям (нормативы 10-м) могут изменяться в зависимости от конфигурации панели управления, прочих устройств, подключенных к ней, от схемы соединений и других условий. Поэтому следует обеспечивать соответствие Директивам ЕС всей системы в целом.

## 6-4 Способы подавления помех на релейных выходах

ПЛК серии CS/CJ соответствуют Стандартам на общие излучения (EN61000-6-4), оговоренным в Директивах по ЭМС. В то же время, уровень помех, генерируемых при переключении релейного выхода, может не удовлетворять этим стандартам. В этом случае в нагрузке должен быть предусмотрен фильтр, либо должны быть предусмотрены другие меры защиты, помимо предусмотренных в ПЛК.

Меры защиты, предпринимаемые с целью удовлетворения стандартам, зависят от нагрузки, схемы соединения, конфигурации системы и т.д. Ниже приводятся примеры способов подавления генерируемых помех.

### Меры защиты

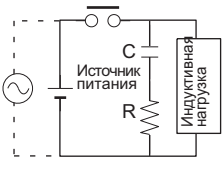
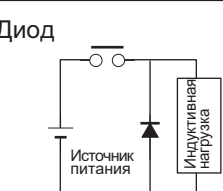
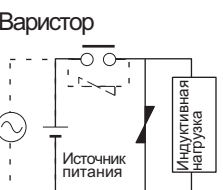
(Подробная информация приведена в EN61000-6-4.)

Применение дополнительных мер не требуется, если частота переключения нагрузки в системе ПЛК не превышает 5 раз в минуту.

Меры противодействия применяются тогда, когда частота переключения нагрузки в системе ПЛК превышает 5 раз в минуту.

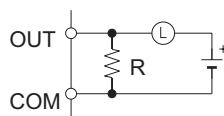
**Примеры способов подавления помех**

Если коммутируется индуктивная нагрузка, параллельно с нагрузкой или контактами следует включить демпфирующую цепочку, диоды и т.п. Схемы подключения показаны ниже.

Схема	Ток		Описание	Требования к элементу
	АС	DC		
<p>CR-цепочка</p> 	Да	Да	<p>Если в качестве нагрузки служит реле или соленоид, обесточивание нагрузки происходит некоторое время спустя после разрыва цепи питания.</p> <p>Если напряжение питания составляет 24 или 48 В, то демпфирующую цепочку следует подключать параллельно нагрузке. Если напряжение питания составляет 100... 200 В, то цепочку следует подключать между контактами.</p>	<p>Емкость конденсатора должна составлять 1...0.5 мкФ на 1 А коммутируемого тока, а сопротивление резистора должно составлять 0.5...1 Ом на 1 В напряжения на контактах. Эти значения могут, однако, меняться в зависимости от нагрузки и характеристики реле. Их можно подобрать экспериментально, учитывая, что ёмкость влияет на подавление искрового разряда в момент размыкания контактов, а сопротивление - на ограничение тока нагрузки в момент замыкания контактов.</p> <p>Диэлектрическая прочность конденсатора должна составлять 200-300 В. В случае переменного тока не следует использовать электролитические конденсаторы.</p>
<p>Диод</p> 	Нет	Да	<p>Подключенный параллельно нагрузке диод преобразует энергию, накопленную индуктивностью, в ток, который, протекая через катушку, затухает с выделением тепла вследствие омического сопротивления индуктивной нагрузки.</p> <p>Время обесточивания нагрузки после разрыва цепи питания в данном случае больше по сравнению с методом демпфирующей RC-цепочки.</p>	<p>Диэлектрическая прочность диода в обратном направлении должна, по меньшей мере, в 10 раз превышать напряжение цепи питания. Максимальный прямой ток диода должен быть равен или должен превышать ток нагрузки.</p> <p>Диэлектрическая прочность диода в обратном направлении может превышать напряжение цепи питания в 2-3 раза в случае, когда речь идет о шунтировании электронных схем с низкими напряжениями.</p>
<p>Варистор</p> 	Да	Да	<p>Подключение варистора предотвращает появление высокого напряжения между контактами за счет постоянства напряжения на варисторе. Обесточивание нагрузки после разрыва цепи питания происходит спустя некоторое время.</p> <p>При напряжении питания 24 В или 48 В варистор следует подключать параллельно нагрузке. В случае напряжения питания 100...200 В варистор подключают параллельно контактам.</p>	---

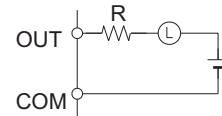
При коммутации нагрузок с высоким значением пускового тока, например, ламп накаливания, для подавления первоначального броска тока необходимо применять следующие схемы:

Способ подавления 1



В лампу поступает 1/3 номинального тока.

Способ подавления 2



Использование ограничительного резистора.

# РАЗДЕЛ 1

## Функционирование модуля CPU

В этом разделе описана основная структура и функционирование модуля CPU.

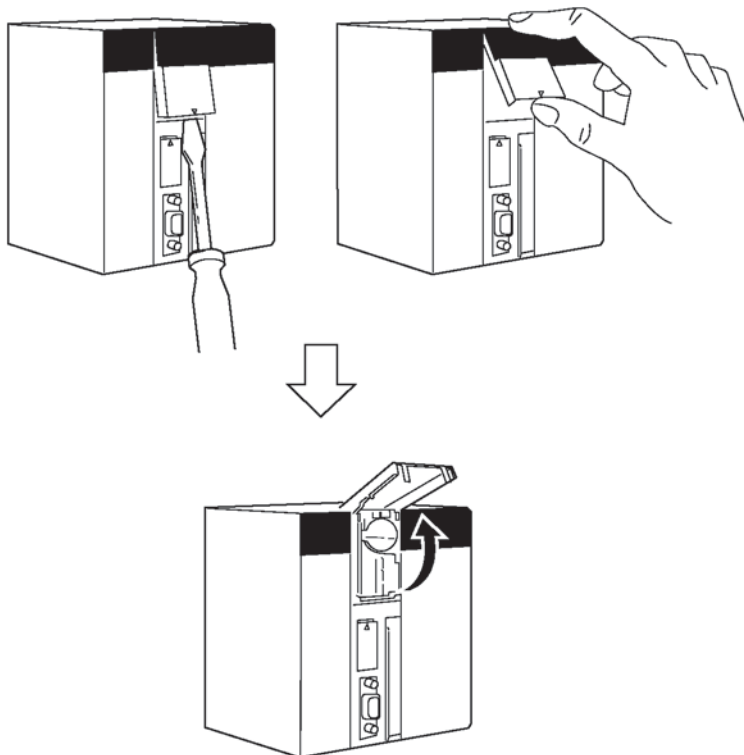
1-1	Первоначальная настройка (только для модулей CPU CS1) . . . . .	2
1-2	Применение внутренних часов (только для модулей CPU CS1) . . . . .	5
1-3	Внутренняя структура модулей CPU . . . . .	6
1-3-1	Обзор . . . . .	6
1-3-2	Структурная схема памяти модуля CPU. . . . .	7
1-4	Режимы работы . . . . .	8
1-4-1	Описание режимов работы. . . . .	8
1-4-2	Инициализация памяти ввода/вывода. . . . .	10
1-4-3	Режим запуска . . . . .	11
1-5	Программы и задачи. . . . .	12
1-6	Описание задач. . . . .	14

## 1-1 Первоначальная настройка (только модули CPU CS1)

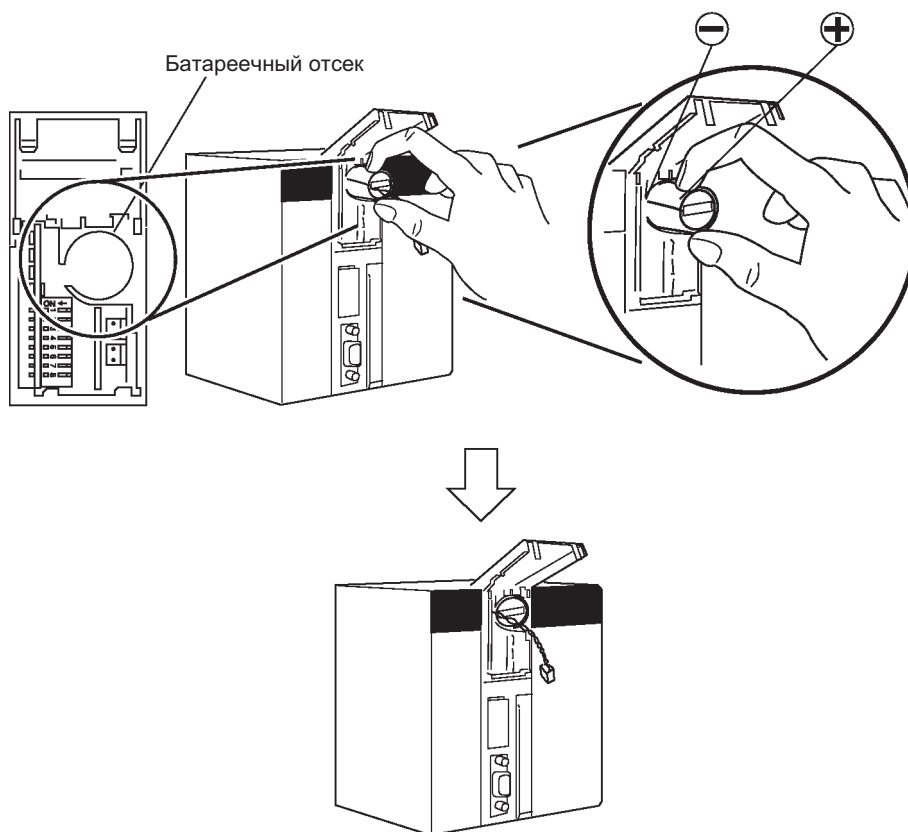
### Установка батареи

Прежде чем использовать модуль CPU CS1, в него следует установить батарею, придерживаясь следующей последовательности действий.

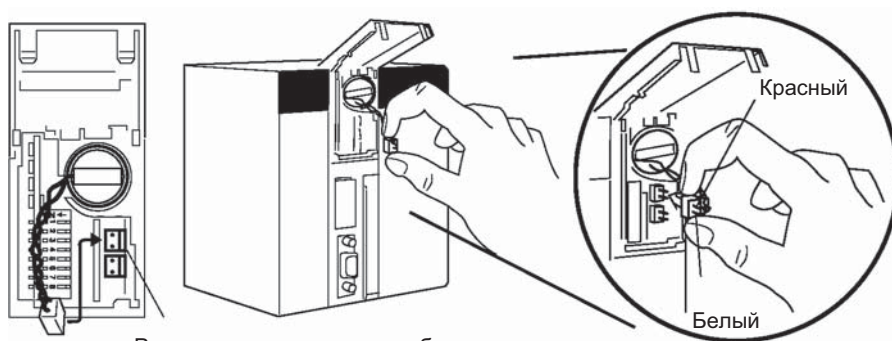
- 1,2,3... 1. Вставьте отвертку с плоским шлицем в небольшое отверстие снизу батарейного отсека, подцепите крышку снизу вверх и откройте ее.



2. Удерживая батарею кабелем наружу, вставьте ее в батарейный отсек.

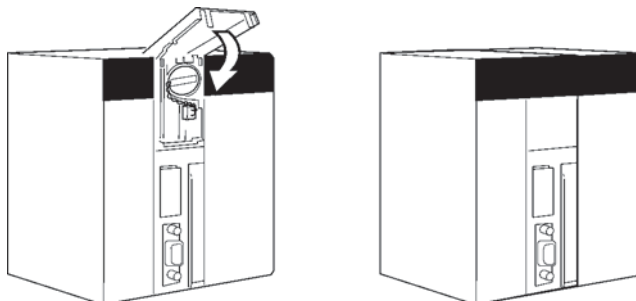


3. Вставьте разъем кабеля батареи в гнездо батарейного отсека. Красный провод должен быть сверху, а белый - снизу. В батарейном отсеке имеются два гнезда; батарею можно подключить к любому из них. Не имеет ни какого значения, какое из гнезд используется, верхнее или нижнее.



Разъемы для подключения батареи (подключите батарею в любой из разъемов).

4. Разместите кабель внутри отсека и закройте крышку отсека.



**Сброс памяти**

После установки батареи необходимо очистить содержимое памяти с целью инициализации ОЗУ внутри модуля CPU, используя специальную процедуру очистки памяти.

**Консоль программирования**

Используя консоль программирования, выполните следующие действия.



**Примечание** В случае очистки памяти с помощью консоли программирования нельзя указывать больше одной задачи, выполняемой циклически. Можно указать одну задачу, выполняемую циклически, и одну задачу, выполняемую по прерыванию; либо одну задачу, выполняемую циклически, и ни одной задачи по прерыванию. Более подробные сведения о процедуре очистки памяти содержатся в руководстве *Operation Manual*. Более подробные сведения о задачах содержатся в *Разделе 1 Функционирование модуля CPU* и в *Разделе 4 Задачи*.

**CX-Programmer**

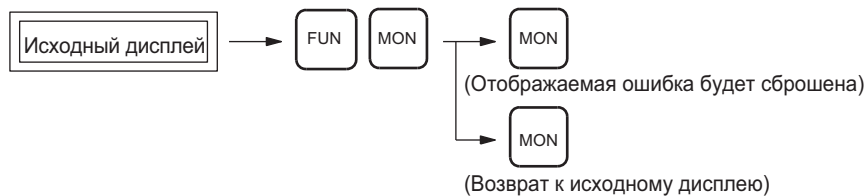
Содержимое памяти также можно очистить с помощью CX-Programmer. Описание последовательности действий содержится в руководстве *CX-Programmer Operation Manual*.

**Очистка ошибок**

Очистив содержимое памяти, следует очистить (сбросить) любые ошибки, которые присутствуют в модуле CPU, включая ошибку "низкое напряжение батареи".

**Консоль программирования**

Используя консоль программирования, выполните следующие действия.



**CX-Programmer**

Сброс ошибок также можно выполнить с помощью CX-Programmer. Последовательность действий описана в руководстве *CX-Programmer Operation Manual*.

**Примечание** Если в контроллер установлена встраиваемая плата, ошибка таблицы маршрутизации встраиваемой платы может не сброситься даже после сброса ошибок с помощью CX-Programmer (у платы последовательного интерфейса остается включенным бит A42407). В этом случае следует либо выключить/включить питание, либо перезапустить встраиваемую плату, после чего вновь сбросить ошибку.

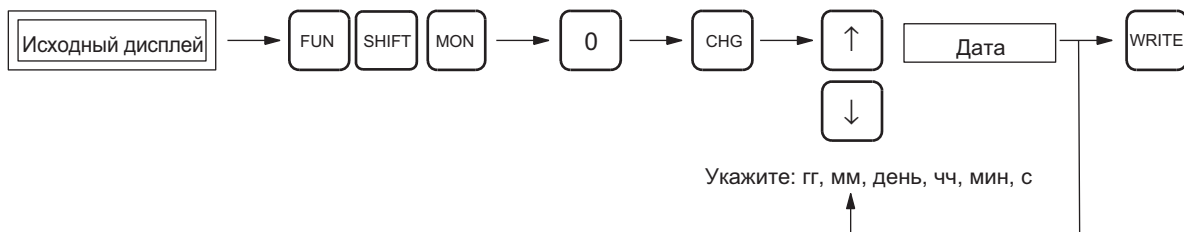
## 1-2 Применение внутренних часов (только для модулей CPU CS1)

После установки в модуль CPU серии CS батареи встроенные часы модуля CPU принимают значение “00 лет, 01 месяцев, 01 дней (00-01-01), 00 часов, 00 минут, 00 секунд (00:00:00), воскресенье (SUN)”.

Если предполагается использовать встроенные часы, после установки батареи следует включить напряжение питания и 1) с помощью устройства программирования (консоль программирования или CX Programmer) скорректировать показания часов, 2) выполнить команду CLOCK ADJUSTMENT (DATE) или 3) передать команду FINS, чтобы начать работу встроенных часов с правильным текущим временем и датой.

Последовательность действий в случае применения консоли программирования, необходимая для настройки внутренних часов, показана ниже.

### Последовательность нажатия клавиш

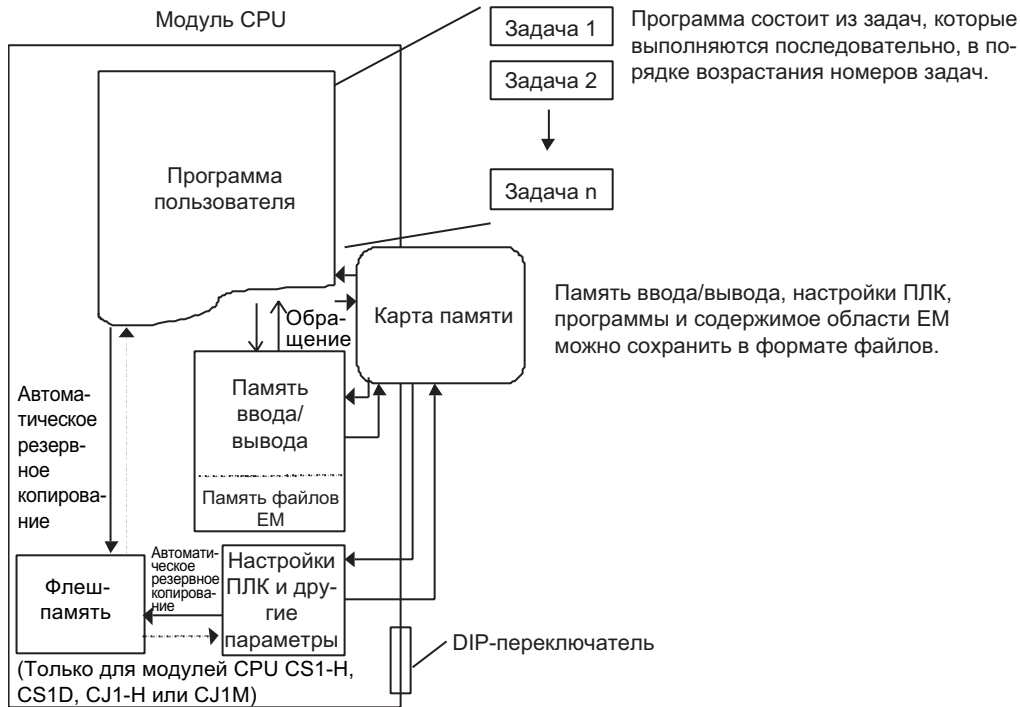




# 1-3 Внутренняя структура модулей CPU

## 1-3-1 Обзор

На следующем рисунке показана внутренняя структура модуля CPU.



### Программа пользователя

Программа пользователя создается из программных задач, которых может быть до 288, включая задачи, выполняемые по прерываниям. Задачи загружаются в модуль CPU из программного пакета CX-Programmer, предназначенного для программирования.

Имеются два типа задач. К первому типу относятся циклические задачи, которые выполняются один раз в каждом цикле (до 32-х), а ко второму типу относятся задачи, выполняемые по прерыванию, т.е., выполняемые только в том случае, когда наступают условия формирования прерывания (до 256-ти). Циклические задачи выполняются последовательно, в порядке возрастания номеров.

### Примечание

1. В модулях CPU CS1-H, CJ1-H, CJ1M или CS1D задачи, выполняемые по прерыванию, могут выполняться циклически точно так же, как и циклические задачи. Это так называемые "дополнительные циклические задачи" ("extra cyclic tasks"). Общее количество задач, выполняемых циклически, не может превышать 288.
2. В случае применения CX-Programmer следует пользоваться версией 2.1 или более новой для модуля CPU CS1-H или CJ1-H и версией 3.0 или более новой для модулей CPU CJ1M (за исключением моделей малого класса) или модулей CPU CS1D для систем с дублированием CPU. В случае применения модуля CPU CJ1M малого класса (CJ1M-CPU11/CP-U21) следует применять CX-Programmer версии 3.3 или более новой. В случае использования модулей CPU CS1-H, CJ1-H, CJ1M или CS1D версии 2.0 или выше следует применять CX-Programmer версии 4.0 или более новой версии.

Команды, содержащиеся в программе, осуществляют чтение и запись в память ввода/вывода и выполняются последовательно, начиная сверху программы. После того, как все циклические задачи выполнены, выполняется обновление входов/выходов для всех модулей, после чего цикл выполняется вновь, начиная с циклической задачи с наименьшим номером.

Сведения об обновлении входов/выходов приводятся в разделе, посвященном работе модуля CPU, в руководстве *CS/CJ Series Operation Manual*.

<b>Память ввода/вывода</b>	<p>В качестве памяти ввода/вывода используется область ОЗУ, которая применяется для операций чтения и записи, выполняемых из программы пользователя. Она состоит из одной области, которая обнуляется при выключении и включении питания, и еще одной области, в которой данные сохраняются.</p> <p>Память ввода/вывода также подразделяется на область, предназначенную для обмена данными со всеми модулями, и область, используемую исключительно для внутренних задач. Обмен данными со всеми модулями происходит один раз в каждом цикле выполнения программы, а также в случае выполнения специальных команд.</p>
<b>Настройки ПЛК</b>	<p>Область настроек ПЛК предназначена для настройки различных первичных и других параметров с помощью программных переключателей (битов).</p>
<b>DIP-переключатели</b>	<p>DIP-переключатели служат для настройки первичных и других параметров. Это аппаратные переключатели.</p>
<b>Карты памяти</b>	<p>В случае необходимости для хранения таких данных, как программы, данные памяти ввода/вывода, настройки ПЛК и комментарии к входам/выходам, созданные с помощью средств программирования, можно использовать карты памяти. Можно сконфигурировать систему таким образом, чтобы после включения питания программы и различные системные параметры считывались из карты памяти автоматически (автоматическая загрузка данных при запуске).</p>
<b>Флеш-память (только у модулей CPU CS1-H, CJ1-H, CJ1M или CS1D)</b>	<p>В модулях CPU CS1-H, CJ1-H, CJ1M или CS1D предусмотрено автоматическое резервное копирование программы пользователя и данных области параметров, например, настроек ПЛК, во встроенную флеш-память в случае каждой записи данных пользователем в модуль CPU. Это позволяет сохранять данные без батарейной подпитки и использования карты памяти. Содержимое памяти входов/выходов, включая большую часть области DM, не сохраняется без батареи.</p>

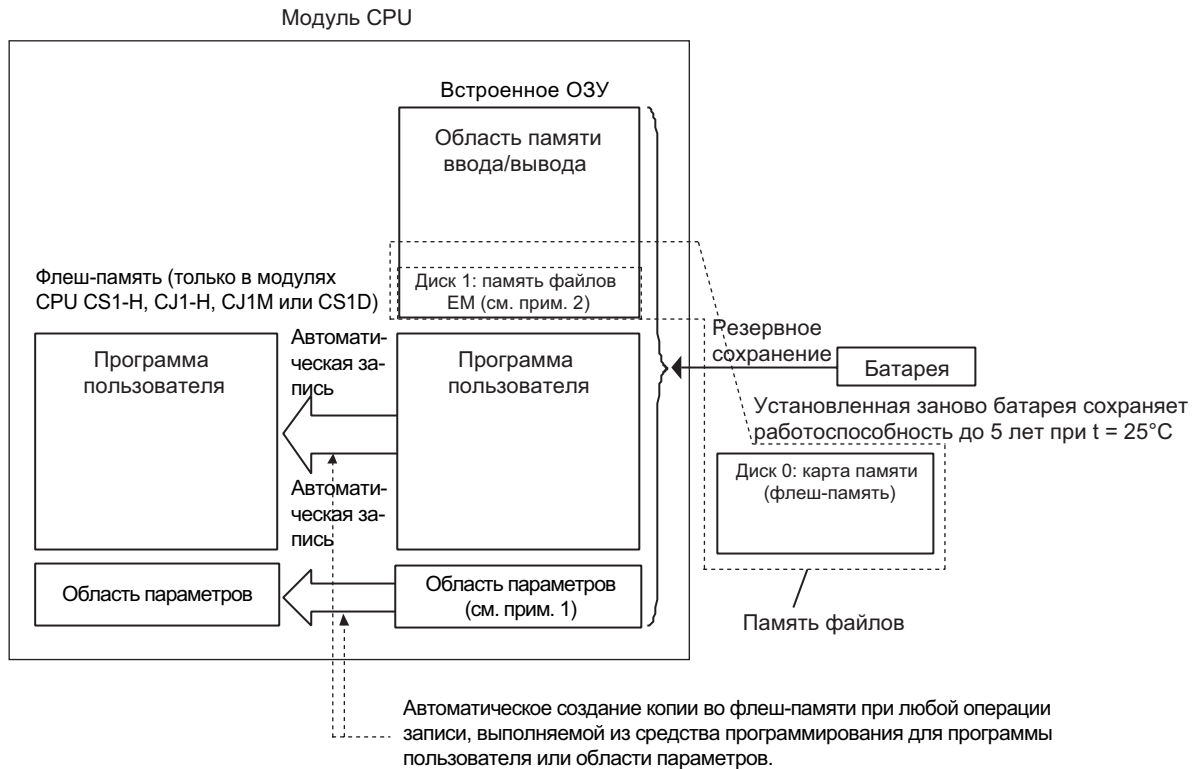
## 1-3-2 Структурная схема памяти модуля CPU

У CPU CS/CJ память модуля CPU (ОЗУ) состоит из следующих блоков:

- Область параметров (настройки ПЛК, зарегистрированная таблица входов/выходов, таблица маршрутизации и параметры модуля шины CPU)
- Области памяти ввода/вывода
- Программа пользователя

Данные в области параметров и в областях памяти ввода/вывода сохраняются с помощью батареи (серия CS: CS1W-BAT01, CJ1-H: CPM2A-BAT01) и будут утрачены в случае снижения напряжения батареи.

В модулях CPU CS1-H, CJ1-H, CJ1M или CS1D, однако, для резервного хранения данных предусмотрена встроенная флеш-память. Программа пользователя и данные области параметров автоматически сохраняются во встроенную флеш-память всякий раз, когда пользователь записывает данные в модуль CPU из устройства программирования (например, CX-Programmer или консоль программирования), в том числе при следующих операциях: загрузка данных, online-редактирование, загрузка данных из карты памяти и т.п. Это означает, что программа пользователя и данные области параметров не будут утрачены даже в случае снижения напряжения батареи.



- Примечание**
1. Область параметров и программу пользователя (т.е., память пользователя) можно защитить от записи, переведя в положение ВКЛ ключ 1 DIP-переключателя на передней панели модуля CPU.
  2. Память файлов EM является частью области EM, которая была преобразована в память файлов в настройках ПЛК. Все банки EM, начиная с указанного банка до конца области EM, можно использовать только в качестве памяти файлов для хранения данных и файлов программ.
  3. Обязательно установите поставленную в комплекте батарею (CS1W-B-AT01), прежде чем в первый раз использовать модуль CPU CS1. Установив батарею, очистите содержимое ОЗУ ПЛК с помощью средства программирования (область параметров, область памяти ввода/вывода и программа пользователя).
  4. При поставке модулей CPU CS1-H, CJ1, CJ1-H, CJ1M или CS1D с завода в них уже установлена батарея. В этом случае нет необходимости сбрасывать память или устанавливать время.
  5. Во время записи данных во флеш-память на передней панели модуля CPU светится индикатор ВКУР. Ни в коем случае не выключайте напряжение питания модуля CPU, пока процедура сохранения данных не будет завершена (т.е., пока не погаснет индикатор ВКУР). Подробные сведения смотрите в 6-6-11 *Флеш-память*.

## 1-4 Режимы работы

### 1-4-1 Описание режимов работы

Ниже описаны режимы работы, предусмотренные в модуле CPU. В этих режимах происходит управление полностью всей программой пользователя и они являются общими для всех задач.

#### Режим PROGRAM (Программирование)

В режиме PROGRAM выполнение программы прекращается, индикатор RUN (Выполнение) не светится. Этот режим предназначен для редактирования программы или выполнения других подготовительных действий, например, для следующих операций:

- Регистрация таблицы ввода/вывода.
- Изменение настроек ПЛК и других настроек.
- Загрузка и проверка программ.
- Принудительная установка и сброс битов с целью проверки цепей и размещения битов.

В этом режиме ни одна циклическая задача или задача, выполняемая по прерыванию, не выполняется (INI), т.е., работа остановлена. Более подробную информацию о задачах смотрите в разделе 1-6 *Описание задач*.

В режиме PROGRAM выполняется обновление входов/выходов. Сведения об обновлении входов/выходов содержатся в руководстве *Operation Manual*.



### Внимание

Модуль CPU обновляет входы/выходы даже в том случае, когда программа остановлена (т.е., даже в режиме PROGRAM). Прежде чем изменять какую-либо из частей памяти, зарезервированной для модулей ввода/вывода, специальных модулей ввода/вывода и модулей шины CPU, тщательно проверьте, безопасны ли эти операции. Любое изменение данных, зарезервированных для любого модуля, может привести к непредвиденному режиму работы нагрузок, подсоединенных к модулю. Любая из следующих операций может привести к изменению состояния памяти.

- Загрузка данных памяти ввода/вывода в модуль CPU из устройства программирования.
- Изменение текущих значений в памяти с помощью средства программирования.
- Принудительная установка/сброс битов с помощью средства программирования.
- Загрузка файлов памяти ввода/вывода из карты памяти или памяти файлов EM в модуль CPU.
- Передача содержимого памяти ввода/вывода из центрального компьютера или другого ПЛК в сети.

### Режим MONITOR (Мониторинг)

Когда программа выполняется в режиме MONITOR, с помощью устройств программирования можно выполнять перечисленные ниже операции. Индикатор RUN (Выполнение) в этом режиме светится. Этот режим предназначен для выполнения пробных запусков или других отладочных операций.

- Редактирование в режиме Online.
- Принудительная установка и сброс битов.
- Изменение значений в памяти ввода/вывода.

В этом режиме, по мере достижения процедурой выполнения программы соответствующих номеров задач, выполняются циклические задачи, для которых выбрано выполнение при запуске (см. прим.), а также задачи, указанные как выполняемые командой TKON(820). Если наступают условия формирования прерывания, выполняются соответствующие задачи обработки прерываний.

### Примечание

Задачи, которые должны выполняться при запуске, указываются в свойствах программы в CX-Programmer.

### Режим RUN (Выполнение)

В этом режиме происходит обычное выполнение программы. В этом режиме светится индикатор RUN (Выполнение). Некоторые операции, выполняемые с помощью устройств программирования, например, редактирование в режиме online, принудительная установка/сброс и изменение значений в памяти ввода/вывода, в этом режиме запрещены. В то же время другие операции, выполняемые с помощью средств программирования, например, контроль состояния выполнения программы (мониторинг программ и мониторинг памяти ввода/вывода), разрешены.

В этом режиме происходит обычная работа системы. Выполнение задач происходит точно так же, как и в режиме MONITOR.

Более подробное описание операций, предусмотренных в каждом из режимов работы, можно найти в разделе 10-2 *Режимы работы модуля CPU*.

## 1-4-2 Инициализация памяти ввода/вывода

В следующей таблице показано, какие области данных обнуляются, когда происходит переход из режима PROGRAM в режим RUN/MONITOR или наоборот.

Изменение режима	Несохраняемые области (прим. 1)	Сохраняемые области (прим. 2)
RUN/MONITOR → PROGRAM	Обнуляется (прим. 3)	Сохраняется
PROGRAM → RUN/MONITOR	Обнуляется (прим. 3)	Сохраняется
RUN ↔ MONITOR	Сохраняется	Сохраняется

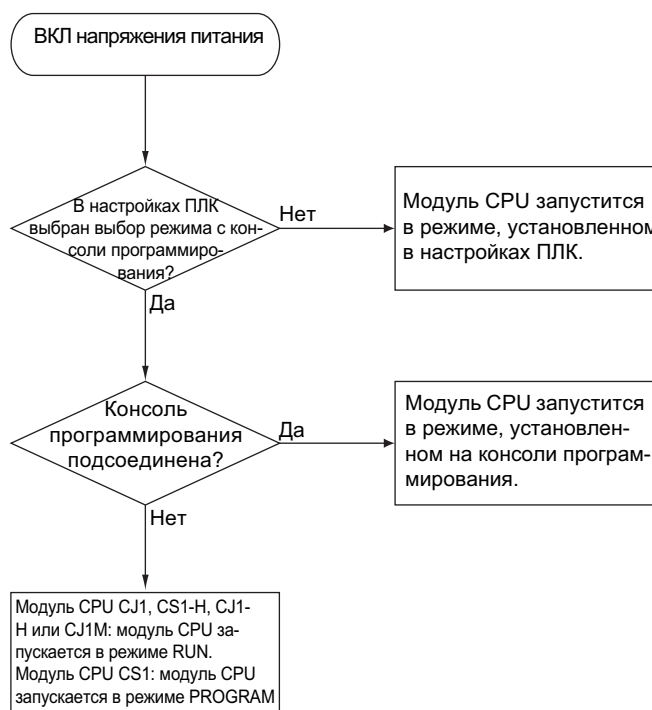
- Примечание**
1. Несохраняемые области: области CIO, рабочая область, область текущих значений (PV) таймеров, флаги завершения таймеров, регистры индексов, регистры данных, флаги задач и флаги условий (состояния некоторых ячеек в дополнительной области сохраняются, а некоторых - обнуляются).
  2. Сохраняемые области: область сохранения (удержания), область DM, область EM, текущие значения (PV) счетчиков и флаги завершения счетчиков.
  3. Данные в памяти ввода/вывода будут сохранены, если установлен бит "Удержание IOM" (A50012). Если бит "Удержание IOM" (A50012) установлен (ВКЛ) и работа прекратилась из-за фатальной ошибки (включая FALS(007)), в этом случае содержимое памяти ввода/вывода будет сохранено, но все выходы модулей вывода будут выключены (OFF).

### 1-4-3 Режим запуска

Сведения о настройке режима запуска для модуля CPU содержатся в руководстве *Operation Manual*.

**Примечание** Если не подсоединена консоль программирования, модули CPU серий CJ1, CS1-H, CJ1-H, CJ1M или CS1D запускаются в режиме RUN (Выполнение). В этом их отличие от работы модуля CPU CS1, который запускается по умолчанию в режиме PROGRAM (Программирование), если не подсоединена консоль программирования.

Условия	Модуль CPU CS1	Модуль CPU CJ1, CS1-H, CJ1-H, CJ1M или CS1D
В настройках ПЛК выбран запуск в соответствии с режимом, выбранным на консоли программирования, но консоль программирования не подсоединена.	Режим PROGRAM (Программирование)	Режим RUN (Выполнение)



## 1-5 Программы и задачи

Задачи определяют последовательность и условия формирования прерываний, по которым запускаются отдельные программы. В целом они подразделяются на следующие типы:

1,2,3...

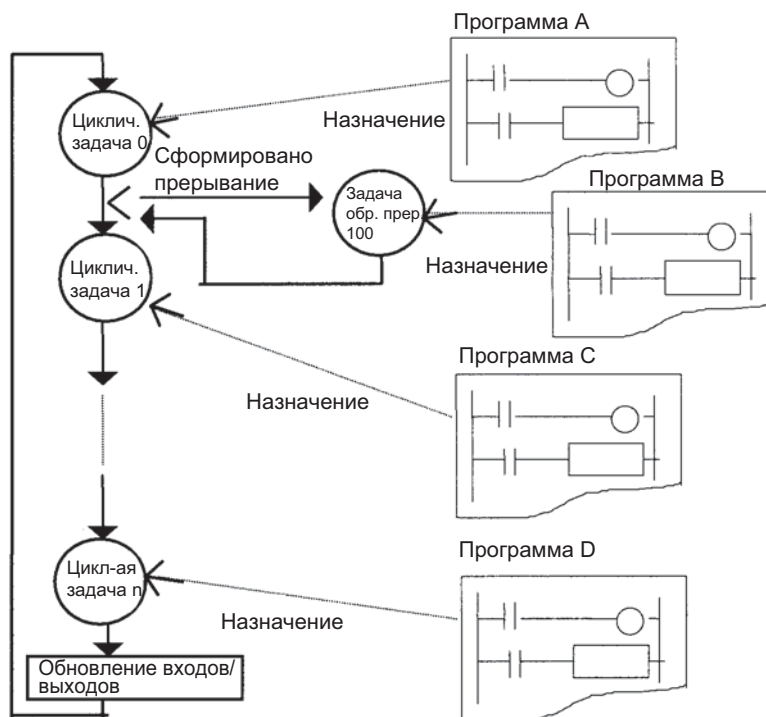
1. Задачи, которые выполняются последовательно, называются "циклическими задачами".
2. Задачи, которые выполняются по прерыванию, называются "задачами обработки прерываний".

**Примечание**

В модулях CPU CS1-H, CJ1-H, CJ1M или CS1D задачи, выполняемые по прерыванию, могут также выполняться циклически, как и обычные циклические задачи. Это так называемые "дополнительные циклические задачи".

Программы, назначенные циклическим задачам, выполняются последовательно, в порядке возрастания номеров задач, а обновление входов/выходов выполняется однократно в каждом цикле после завершения всех задач (если более точно, то после завершения задач, которые являются выполняемыми). Если во время выполнения циклических задач удовлетворяются условия формирования прерывания, выполнение циклической задачи прерывается и выполняется программа, назначенная для задачи обработки прерывания.

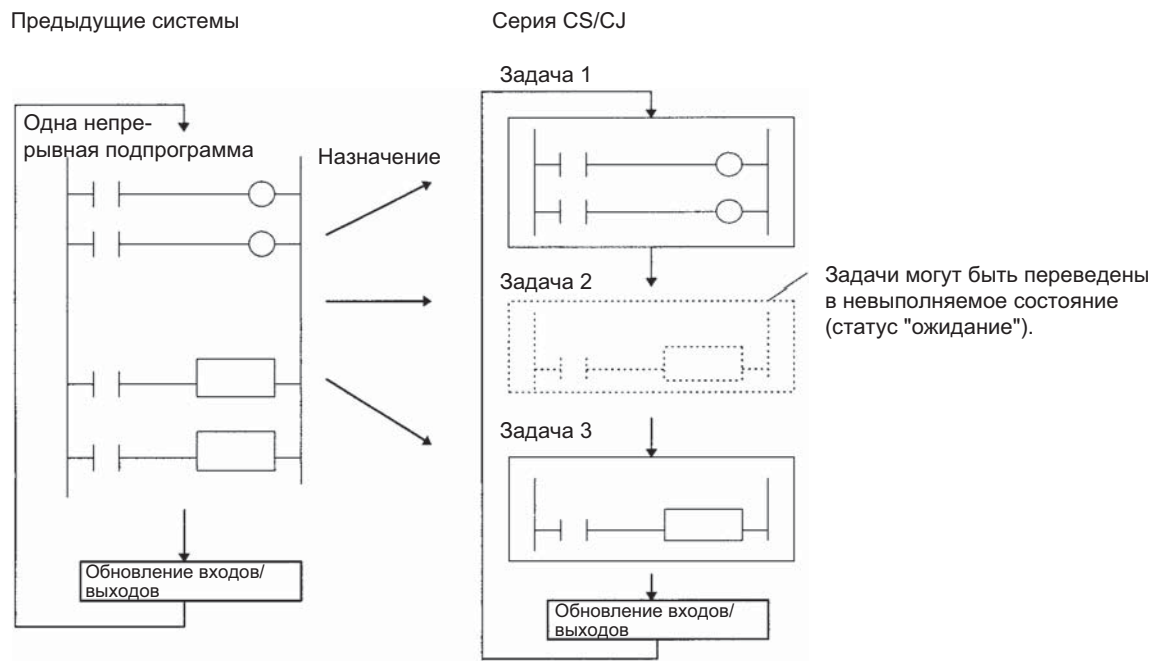
Сведения об обновлении входов/выходов содержатся в разделе, посвященном работе модуля CPU, в руководстве *CS/CJ Series Operation Manual*.



На приведенной выше диаграмме программы выполняются в следующем порядке: первая часть программы А, В, остаток программы А, С, а затем D. Предполагается, что во время выполнения программы А было сформировано прерывание, обрабатываемое задачей 100. Когда выполнение программы В завершается, выполняется остаток программы А, начиная с того места, на котором прервалось выполнение.

В более ранних версиях ПЛК OMRON из нескольких, непрерывно выполняемых частей формируется одна непрерывная программа. Программы, назначенные для каждой задачи, являются отдельными программами, которые завершаются командой END, точно так же, как и отдельные программы в предыдущих ПЛК.

Одним из свойств циклических задач является возможность их включения (статус "выполняемая задача") и отключения (статус "ожидание") с помощью команд управления задачами. Это позволяет создавать задачи, состоящие из нескольких программных компонентов, и выполнять для каждого производственного процесса только отдельные требуемые программы (задачи) (переключение шагов программы). Это позволяет существенно повысить производительность (сократить время цикла), поскольку выполняются только требуемые программы.



Выполненная задача вновь будет выполнена в последующих циклах, а невыполняемая задача сохраняет в последующих циклах статус "ожидание" до тех пор, пока она не будет выполнена вновь в другой задаче.

- Примечание** В отличие от прежнего подхода к выполнению программы, который можно сравнить со чтением непрерывной "ленты", выполнение программы в виде задач можно сравнить со чтением последовательности отдельных "карт".
- Все "карты" считываются в заданной последовательности, начиная с карты с наименьшим номером.
  - Различают активные и неактивные "карты". Неактивные "карты" пропускаются ("карты" активизируются и деактивируются с помощью команд управления задачами).



- Активизированная "карта" остается активной и считывается в последующих циклах (последовательностях). Деактивизированная "карта" остается неактивной и пропускается до тех пор, пока она не будет повторно активизирована другой "картой".

Предшествующая программа:  
похожа на "ленту"



Программа серии CS/CJ:  
похожа на набор "карт", которые можно  
включать или отключать другими "картами".



## 1-6 Описание задач

В широком смысле задачи классифицируют следующим образом:

1,2,3...

### 1. Циклические задачи (макс. 32)

Задачи, которые выполняются один раз в пределах цикла (если они являются выполняемыми). Если требуется, выполнение циклических задач может быть отменено.

### 2. Задачи, выполняемые по прерыванию

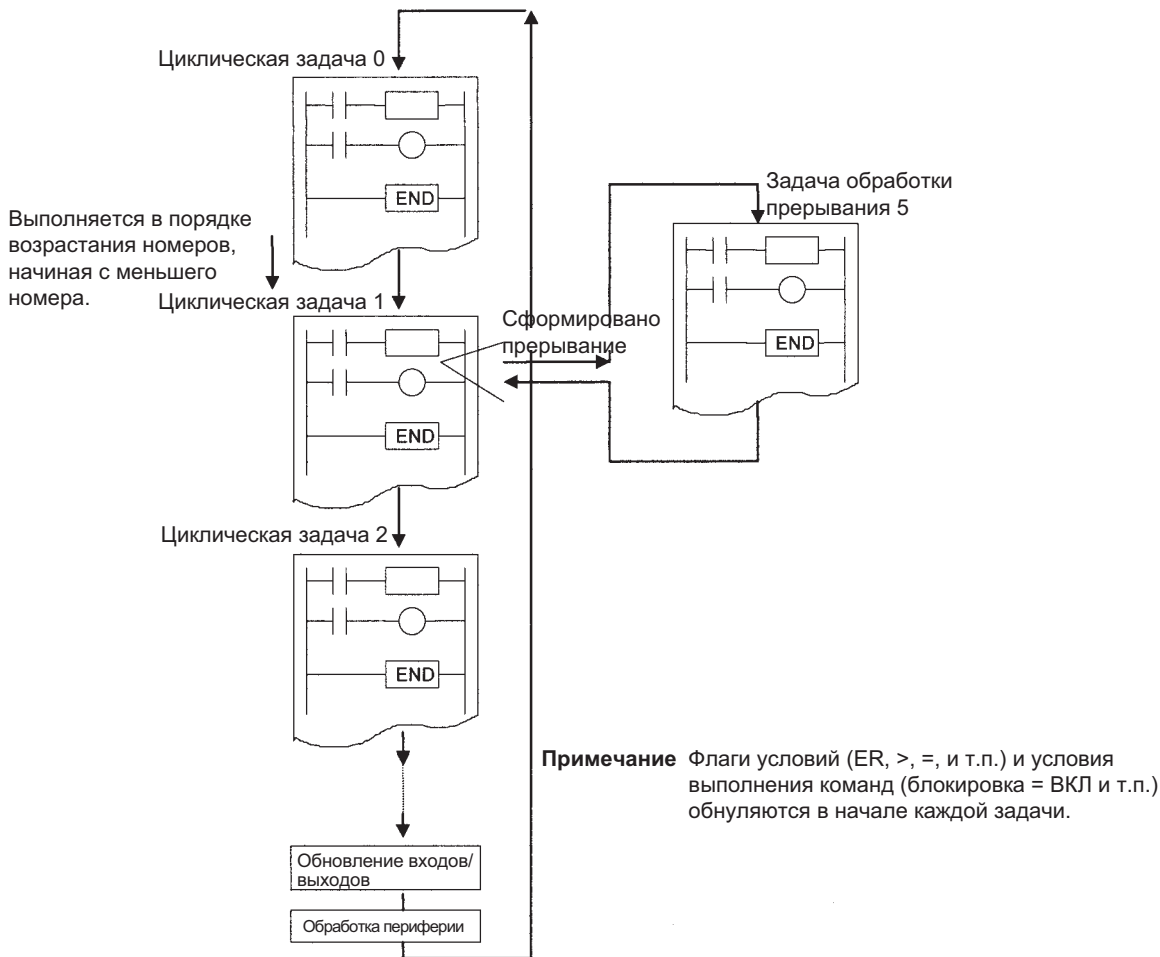
Задачи, которые выполняются, если сформировано прерывание. Задачи выполняются независимо от того, выполняется в данный момент циклическая задача, или нет. Различают следующие четыре типа задач, выполняемых по прерыванию (см. прим. 1 и 2) (пять типов, если учитывать дополнительные циклические задачи для модулей CPU CS1-H, CJ1-H, CJ1M или CS1D):

- Задача, выполняемая по прерыванию от выключения питания (не поддерживается модулями CPU CS1D для систем с дублированием CPU): Выполняется в случае пропадания питания (1 макс.).
- Задача, выполняемая по запланированному прерыванию (не поддерживается модулями CPU CS1D для систем с дублированием CPU): Выполняется с установленной периодичностью (2 макс.).
- Задача, выполняемая по прерыванию от входов/выходов (не поддерживается модулями CPU CJ1 или CS1D для систем с дублированием CPU): Выполняется, когда включается вход модуля ввода прерываний (32 макс.).
- Задача, выполняемая по внешнему прерыванию (не поддерживается модулями CPU CJ1 или CS1D для систем с дублированием CPU): Выполняется (256 макс.), когда поступает запрос от специального модуля ввода/вывода, модуля шины CPU или встраиваемой платы (только для серии CS).
- Дополнительные циклические задачи (поддерживаются только модулями CPU CS1H, CJ1-H, CJ1M и CS1D): Задачи обработки прерываний, выполняемые как циклические. Дополнительные циклические задачи выполняются один раз в каждом цикле до тех пор, пока они имеют статус "выполняемые".

С помощью CX-Programmer можно создать и управлять максимум 288 задачами с 288 программами, в том числе 32 циклические задачи и 256 задач, выполняемых по прерыванию.

- Примечание**
1. В настоящее время модули CPU CJ1 не поддерживают задачи, выполняемые по прерыванию от входов/выходов и выполняемые по внешним прерываниям. Следовательно, максимальное количество задач для модуля CPU CJ1 составляет 35, т.е., 32 циклические задачи и 3 задачи по прерыванию. Всего, таким образом, можно создать и управлять 35 программами.
  2. Модули CPU серии CS1D вообще не поддерживают задачи, выполняемые по прерыванию, однако выполняемые по прерыванию задачи можно использовать в качестве дополнительных циклических задач.

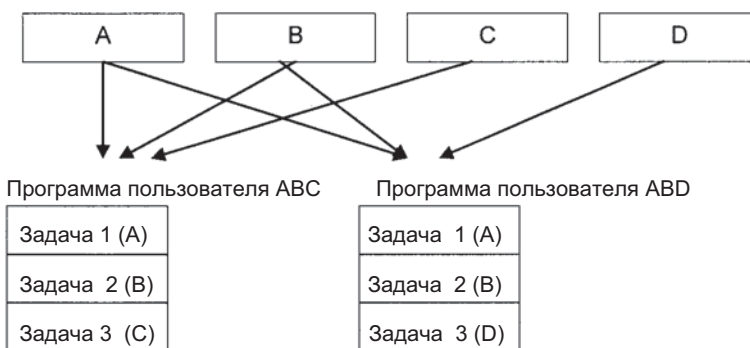
Каждая программа назначается одной определенной задаче. Для этого с помощью CX-Programmer индивидуально настраиваются свойства каждой программы.



**Структура программы**

При написании программ по мере необходимости могут создаваться стандартные подпрограммы, которые назначаются задачам. Это значит, что программы можно создавать из отдельных модулей (стандартных компонентов) и каждую задачу можно отлаживать отдельно.

Стандартные подпрограммы



При создании модульных программ в целях унификации для обозначения адресов можно использовать символы.

**Статусы "выполняемая задача" и "ожидание"**

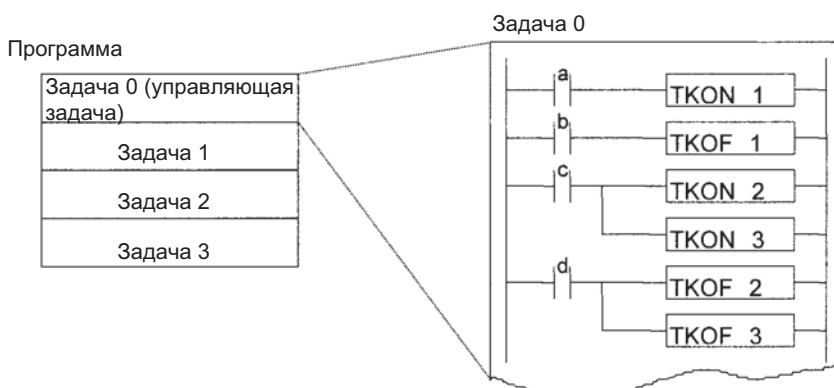
В пределах задачи допускается выполнять команды TASK ON и TASK OFF (TKON(820) и TKOF(821)), чтобы перевести другую задачу в состояние "выполняемая задача" или "ожидание".

Команды, находящиеся в задачах, имеющих статус "ожидание", не выполняются, но состояние их входов/выходов сохраняется. Когда задача возвращается в выполняемое состояние, команды начинают выполняться с учетом состояния входов/выходов, которое поддерживалось все это время.

**Пример: программа с управляющей задачей**

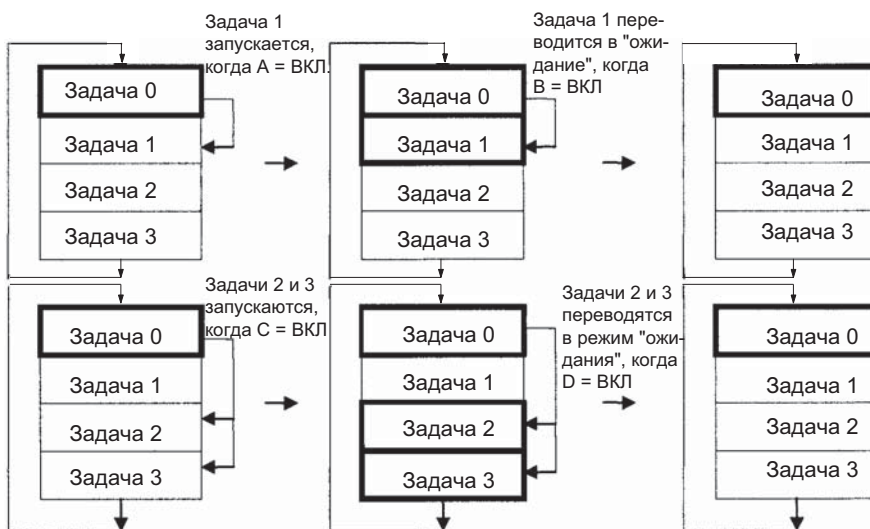
Ниже приведен пример программы, в которой задача 0 является управляющей задачей и выполняется первой в начале работы. С помощью CX-Programmer (но не с помощью консоли программирования) для других задач можно указать, должны ли они или не должны запускаться в начале работы.

Как только начато выполнение программы, выполнением задач можно управлять с помощью команд TKON(820) и TKOF(821).



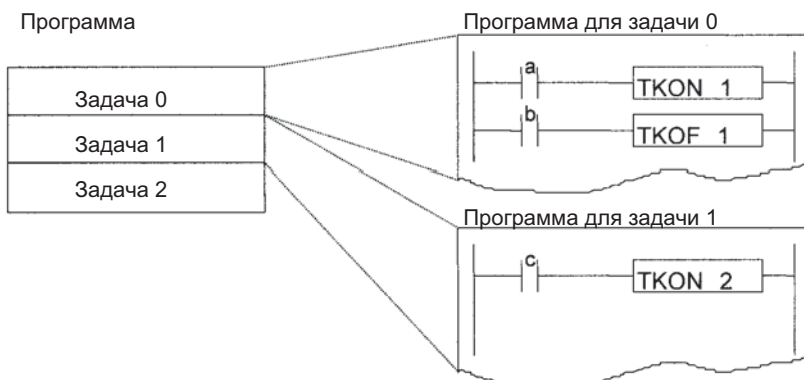
Пример: Для задачи 0 выбрано выполнение в начале работы (указано в свойствах программы в CX-Programmer).

Задача 1 становится выполняемой, когда устанавливается (ВКЛ) условие А. Задача 1 переводится в "ожидание", когда устанавливается условие В. Задачи 2 и 3 становятся выполняемыми, когда устанавливается условие С. Задачи 2 и 3 переводятся в "ожидание", когда устанавливается (ВКЛ) условие D.

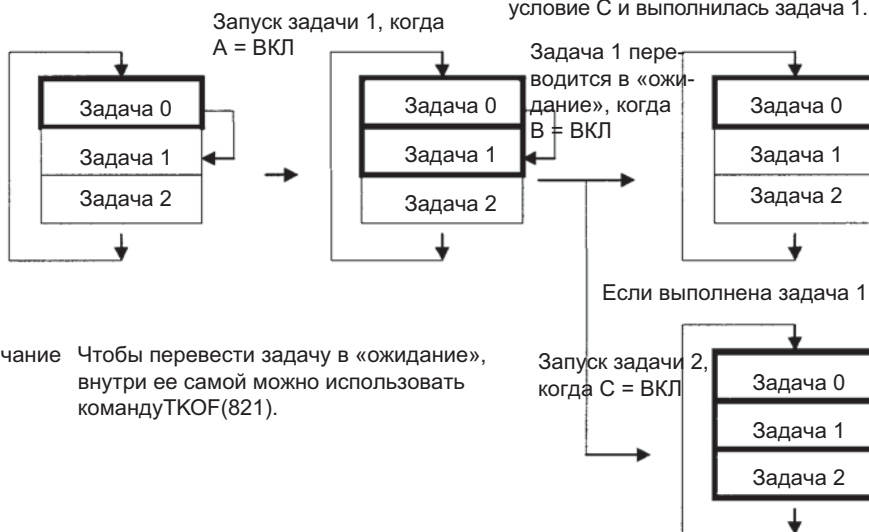


**Пример: Каждая задача управляется другой задачей**

Ниже приведен пример ситуации, когда каждая задача управляется другой задачей.



Пример: Задача 1 становится выполняемой в начале работы без каких-либо условий.  
 Задача 2 становится выполняемой, когда устанавливается (ВКЛ) условие А.  
 Задача 1 переводится в «ожидание», когда устанавливается (ВКЛ) условие В.  
 Задача 2 становится выполняемой, когда устанавливается (ВКЛ) условие С и выполнена задача 1.



Примечание Чтобы перевести задачу в «ожидание», внутри ее самой можно использовать команду TKOF(821).

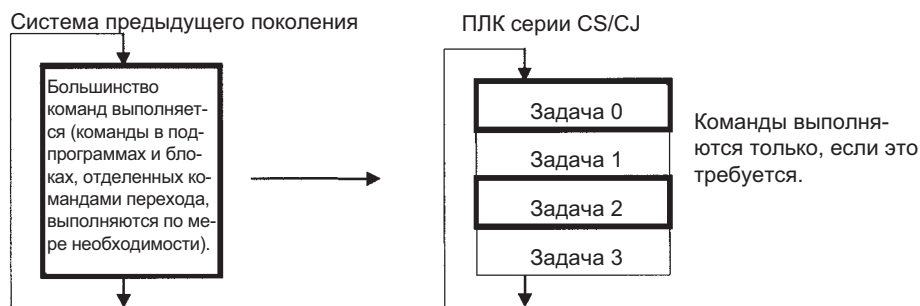
**Время выполнения задачи**

Пока задача находится в режиме ожидания, команды в этой задаче не выполняются, поэтому время их выполнения не учитывается при подсчете времени цикла.

**Примечание**

С этой точки зрения, команды в задаче, имеющей статус "ожидание", похожи на команды, расположенные в том разделе программы, который обходится с помощью команд перехода (JMP-JME).

Поскольку команды в невыполняемой задаче не увеличивают время цикла, общее быстродействие системы существенно возрастет, если вся система будет состоять из общей управляющей задачи и отдельных задач, которые выполняются по мере необходимости.





## РАЗДЕЛ 2 Программирование

В этом разделе приведена основная информация, необходимая для осуществления записи, проверки и ввода программ.

2-1	Основные принципы . . . . .	22
2-1-1	Программы и задачи . . . . .	22
2-1-2	Основные сведения о командах . . . . .	23
2-1-3	Размещение команд и условия выполнения . . . . .	25
2-1-4	Обращение к областям памяти ввода/вывода. . . . .	26
2-1-5	Указание операндов . . . . .	27
2-1-6	Форматы данных. . . . .	32
2-1-7	Варианты выполнения команд . . . . .	36
2-1-8	Условия выполнения . . . . .	36
2-1-9	Синхронизация команд ввода/вывода . . . . .	39
2-1-10	Синхронизация обновления . . . . .	41
2-1-11	Объем программы. . . . .	44
2-1-12	Основные принципы программирования «лестничных диаграмм». . . . .	44
2-1-13	Ввод мнемонических кодов . . . . .	49
2-1-14	Примеры программ. . . . .	52
2-2	Замечания . . . . .	57
2-2-1	Флаги условий . . . . .	57
2-2-2	Специальные разделы программ . . . . .	62
2-3	Проверка программ. . . . .	66
2-3-1	Ошибки при вводе программы с помощью средства программирования . . . . .	66
2-3-2	Проверка программы с помощью CX Programmer . . . . .	66
2-3-3	Проверка выполнения программы . . . . .	68
2-3-4	Проверка на наличие фатальных ошибок . . . . .	70

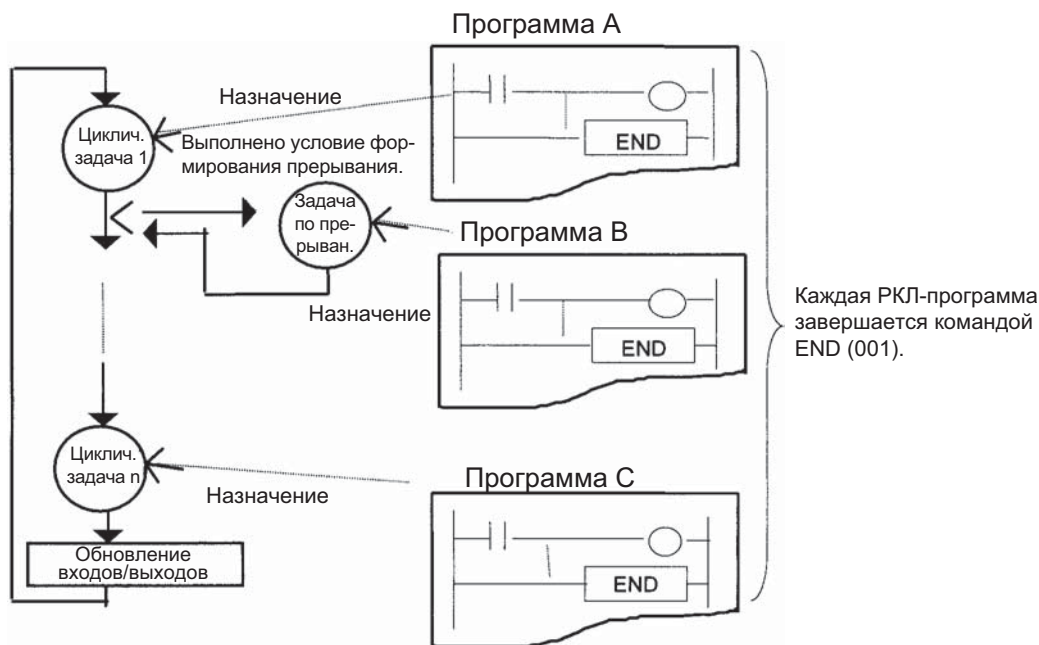


## 2-1 Основные принципы

### 2-1-1 Программы и задачи

ПЛК серии CS/CJ выполняют содержащиеся в задачах программы релейно-контактной логики (так называемые "лестничные диаграммы"). РКЛ-программа в каждой задаче завершается командой END (001), как и у обычных ПЛК.

Задачи позволяют определить порядок выполнения РКЛ-программ, а также условия для формирования прерываний.



В этом разделе описаны основные принципы, которые требуется соблюдать при написании программ для ПЛК серии CS/CJ. Более подробные сведения о задачах и их взаимосвязи с РКЛ-программами содержатся в **РАЗДЕЛЕ 4 Задачи**.

**Примечание Задачи и средства программирования**

Для управления задачами используются средства программирования, описанные ниже. Более подробно об этом можно прочитать в **РАЗДЕЛЕ 4-4 Использование средств программирования для задач**, а также в руководствах *CS/CJ-series Programming Consoles Operation Manual (W341)* и *CX-Programmer Operation Manual*.

**CX-Programmer**

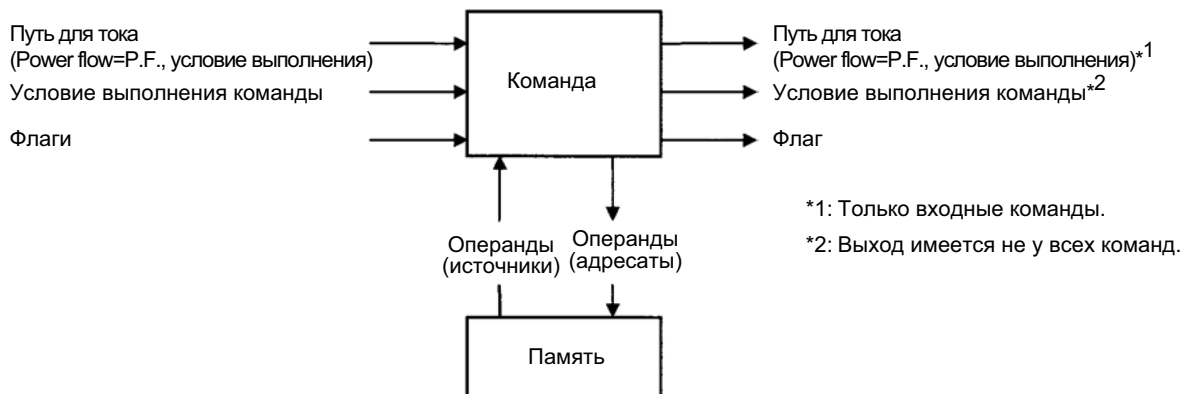
CX-Programmer служит для определения типов задач и номеров задач, как атрибутов для отдельных программ.

**Консоль программирования**

Для доступа к программам и их редактированию на консоли программирования для циклических задач указывается СТ00 ... СТ31, а для задач, выполняемых по прерыванию, указывается IT00 ... IT255. Если на консоли программирования выполняется процедура обнуления памяти, то в новую программу может быть записана только циклическая задача 0 (СТ 00). Для создания циклических задач 1 ... 31 (СТ01 ... СТ31) следует использовать CX-Programmer.

### 2-1-2 Основные сведения о командах

Программа состоит из команд. Команда содержит входы и выходы. Принцип построения команды показан на рисунке ниже.

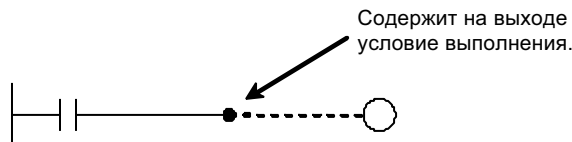


#### Путь для тока

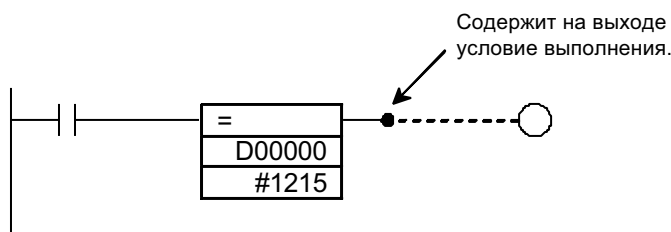
Путь для тока (P.F.) - это условие выполнения, которое служит для управления выполнением и командами, когда программы выполняются в обычном режиме. В РКЛ-программах открытый путь для тока представляет собой условие выполнения.

#### Входные команды

- Входные команды представляют логическое условие для запуска и содержат на выходе условие выполнения.

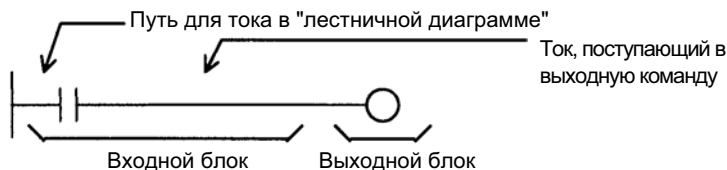


- Промежуточные команды "впускают" ток в качестве условия выполнения и "отдают" ток промежуточной или выходной команде.



#### Выходные команды

Выходные команды выполняют все функции, используя в качестве условия выполнения входной ток.



#### Условия выполнения команд

Условия выполнения команд - это специальные условия, которые относятся ко всему выполнению команды целиком и которые присутствуют на выходе команд, указанных ниже. Условия выполнения команд обладают приоритетом по отношению к пути для тока (P.F.) при принятии решения о выполнении или невыполнении команды. В зависимости от условий выполнения команды, команда может стать невыполняемой или может быть выполнена иным образом.

Условия выполнения команды сбрасываются (отменяются) в начале каждой задачи. Другими словами, они сбрасываются при смене задачи.

Для установки и отмены определенных условий выполнения команды служат пары следующих команд. Эти пары команд должны находиться в одной и той же задаче.

Условие выполнения команды	Описание	Устанавливающая команда	Отменяющая команда
Блокировка	Блокировка запрещает выполнение части программы. В качестве блокировки могут выступать такие условия, как отключение (OFF) выходных битов, сброс таймеров и счетчики с сохранением значения.	IL (002)	ILC (003)
Выполнение BREAK (514)	Завершает цикл FOR (512) - NEXT (513) во время выполнения (запрещает выполнение всех команд до появления следующей команды NEXT (513)).	BREAK (514)	NEXT (513)
	Выполняет переход от JMP0 (515) к JME0 (516).	JMP0 (515)	JME0 (515)
Выполнение прогр. блока	Выполняет программный блок между BPRG (096) и BEND (801).	BPRG (096)	(801)

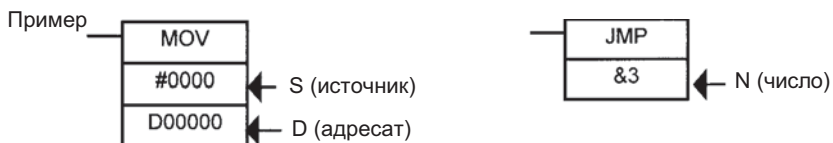
**Флаги**

В данном случае под флагом понимается бит, посредством которого отдельные команды взаимодействуют между собой.

Входные флаги	Выходные флаги
<ul style="list-style-type: none"> <li>Флаги распознавания смены состояний Флаги результатов распознавания смены состояний. Состояние этих флагов автоматически поступает на вход выходных команд с дифференцированием направления переключения (вверх/вниз), а также команд DIFU(013)/DIFD(014).</li> <li>Флаг переноса (CY) Флаг переноса служит в качестве неопределяемого операнда в командах, осуществляющих сдвиг (смещение) данных, и в командах сложения/вычитания.</li> <li>Флаги для специальных команд Сюда входят обучающие флаги для команд FPD (269) и флаги разрешения сетевых коммуникаций.</li> </ul>	<ul style="list-style-type: none"> <li>Флаги распознавания смены состояний Флаги результатов распознавания смены состояний. Состояние этих флагов автоматически поступает на выход команд с дифференцированием направления переключения (вверх/вниз), а также команд UP(521)/DOWN(522).</li> <li>Флаги условий В состав флагов условий входят флаги "всегда ВКЛ/ВЫКЛ", а также флаги, состояние которых обновляется в результате выполнения команды. В программах пользователя вместо адресов для этих флагов можно использовать символы (метки), например, ER, CY, &gt;, =, A1, A0.</li> <li>Флаги для специальных команд Сюда входят флаги для команд, предназначенных для работы с картой памяти, а также флаги завершения выполнения MSG (046).</li> </ul>

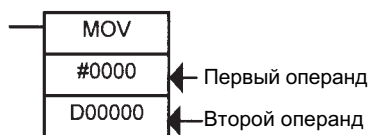
**Операнды**

С помощью операндов указываются предустановленные параметры команды (ячейки таблиц в "лестничных диаграммах"). Они позволяют указать содержимое области памяти ввода/вывода или константы. Для выполнения команды в качестве операнда можно указывать адрес или константу. В качестве операнда может быть указан источник, адресат или численное значение.



Типы операндов		Символьное обозначение	Описание	
Источник	Указывает адрес, по которому должны быть прочитаны данные, или константу.	S	Операнд-источник	Операнд-источник, не являющийся управляющими данными (С).
		C	Управляющие данные	Составные данные операнда-источника, значение которых зависит от состояния бита.
Адресат (результат)	Указывает адрес, по которому должны быть записаны данные.	D (R)	---	
Число	Указывает определенное число, которое должно использоваться в команде, например, номер для перехода или номер подпрограммы.	N	---	

**Примечание** Для операндов также используют термины "первый операнд", "второй операнд" и так далее, начиная с верхнего операнда команды.



### 2-1-3 Размещение команд и условия выполнения

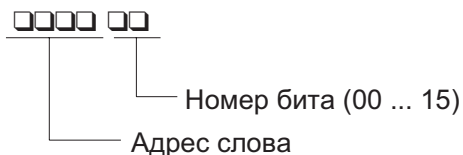
В следующей таблице показано возможное размещение команд. Команды разбиты на отдельные группы. В первую группу включены команды, для которых требуются условия разрешения выполнения, а во вторую - для которых условия разрешения выполнения не требуются. Подробное описание таких команд приведено в РАЗДЕЛЕ 3 Команды и их функции.

Тип команды		Возможное размещение	Условие выполнения	Символ в "лестничной диаграмме"	Примеры
Входные команды	Логическое условие начала выполнения (команд-"нагрузок")	Подключаются непосредственно к левой шине или размещаются в начале блока команд.	Не требуется		LD, LD TST(350), LD > (и другие команды сравнения, использующие символы)
	Промежуточные команды	Размещается между логическим условием начала выполнения и выходной командой.	Требуется		AND, OR, AND TEST(350), AND > (и другие команды сравнения, использующие символ ADD), UP(521), DOWN(522), NOT(520) и т.п.
Выходные команды		Подключается непосредственно к правой шине.	Требуется		Большинство команд, включая OUT и MOV(021).
			Не требуется		END(001), JME(005), FOR(512), ILC(003) и т.п.

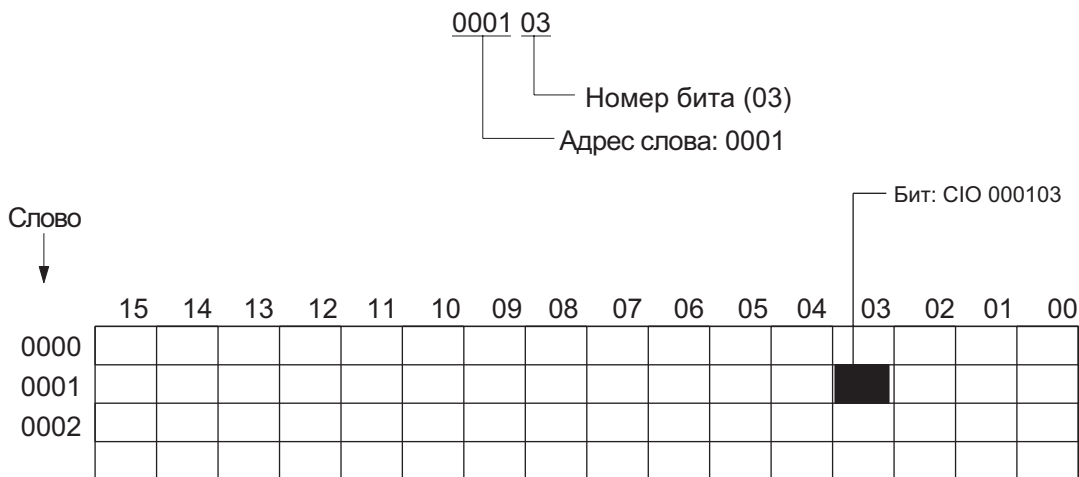
- Примечание**
1. Имеется еще одна группа команд, предназначенная для выполнения последовательности мнемонических команд, имеющих один общий вход. Это так называемые команды для программирования блоков. Подробное описание таких блочных программ содержится в руководстве *CS/CJ Series CPU Units Instruction Reference Manual*.
  2. Если команда, для которой требуется условие выполнения, подключена непосредственно к левой шине без применения команды логического запуска, при проверке программы на средстве программирования (CX-Programmer или консоль программирования) будет обнаружена ошибка программы.

## 2-1-4 Обращение к областям памяти ввода/вывода

### Адреса битов



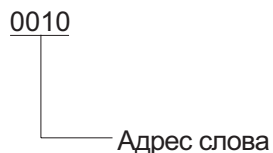
**Пример:** ниже показан адрес бита 03 в слове 0001 области СІО. В этом Руководстве такой адрес будет указываться следующим образом: "СІО 000103".



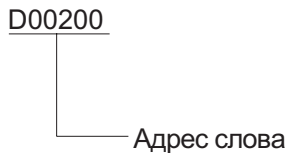
### Адреса слов



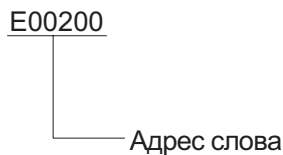
**Пример:** ниже показан адрес битов 00 ... 15 в слове 0010 области СІО. В этом Руководстве такой адрес будет обозначаться следующим образом: "СІО 0010".



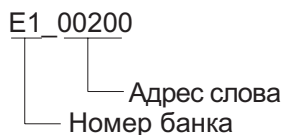
К адресам областей DM и EM добавляются префиксы "D" или "E". Ниже показан пример для адреса D00200.



**Пример:** ниже показан адрес слова 2000 в текущем банке расширенной памяти данных:



Ниже показан адрес слова 2000 в банке 1 расширенной памяти данных:



### 2-1-5 Указание операндов

Операнд	Описание	Примечание	Примеры применения
Указание адресов битов	<p>Для обращения к биту (входные биты) указываются непосредственно номер слова и номер бита.</p> <p>□□□□ □□</p> <p>Адрес слова.</p> <p>Номер бита (00 ... 15)</p> <p><b>Примечание</b> Для обращения к флагам завершения таймеров/счётчиков и для обращения к текущим значениям используются одни и те же адреса. Для обращения к флагу задачи предусмотрен только один адрес.</p>	<p>0001 02</p> <p>Номер бита (02)</p> <p>Номер слова: 0001</p>	<p>0001 02</p>
Указание адресов слов	<p>Для обращения к 16-битовому слову указывается непосредственно номер слова.</p> <p>□□□□</p> <p>Адрес слова.</p>	<p>0003</p> <p>Номер слова: 0003</p> <p>D00200</p> <p>Номер слова: 00200</p>	<p>MOV 0003 D00200</p>

Операнд	Описание	Примечание	Примеры применения
<p>Указание косвенных адресов областей DM/EM в двоичном коде</p>	<p>В данном случае для адресации указывается величина смещения от начала области. Двоичные данные (00000 ... 32767), содержащиеся по указанному адресу, указывают на адрес слова в памяти данных (DM) или в области расширенной памяти данных (EM). Чтобы указать косвенный адрес в двоичном формате, впереди адреса следует указывать символ @.</p> <p style="text-align: center;">@D□□□□□</p> <p style="text-align: center;">↓</p> <p>Двоичные данные □□□□□ 00000 ... 32767 (0000 Hex ... 7FFF Hex в двоичном формате)</p> <p style="text-align: center;">↓</p> <p>D □□□□□</p>		
	<p>1) Если в @D (□□□□□) содержится 0000 Hex ... 7FFF Hex (00000 ... 32767), это соответствует обращению к D00000 ... D32767.</p>	<p>@D00300</p> <p style="text-align: center;">□ 0 1 0 0 □ Двоичные данные Двоичный код: 256</p> <p style="text-align: center;">↓</p> <p style="text-align: center;">Адрес D00256.</p> <p style="text-align: center;">— Добавляется символ @.</p>	<p>MOV #0001 @00300</p>
	<p>2) Если в @D (□□□□□) содержится 8000 Hex ... FFFF Hex (32768 ... 65535), это соответствует обращению к E0_00000 ... E0_32767 банка 0 расширенной памяти данных (EM).</p>	<p>@D00300</p> <p style="text-align: center;">□ 8 0 0 1 □ Двоичные данные Двоичный код: 32769</p> <p style="text-align: center;">↓</p> <p style="text-align: center;">Адрес E0_00001.</p>	
	<p>3) Если в @E□ □□□□□ содержится 0000 Hex ... 7FFF Hex (00000 ... 32767), это соответствует обращению к E□_00000 ... E□_32767 в указанном банке.</p>	<p>@E1_00200</p> <p style="text-align: center;">□ 0 1 0 1 □ Двоичные данные Двоичный код: 257</p> <p style="text-align: center;">↓</p> <p style="text-align: center;">Адрес E1_00257.</p>	<p>MOV #0001 @E1_00200</p>
	<p>4) Если в @E□ □□□□□ содержится 8000 Hex ... FFFF Hex (32768 ... 65535), это соответствует обращению к E(□+1)_00000 ... E(□+1)_32767 в банке, который следует за указанным банком</p>	<p>@E1_00200</p> <p style="text-align: center;">□ 8 0 0 2 □ Двоичные данные Двоичный код: 32770</p> <p style="text-align: center;">↓</p> <p style="text-align: center;">Адрес E2_00002.</p>	
	<p><b>Примечание</b> В случае указания косвенного адреса в двоичном коде память данных (DM) и расширенная память данных (EM) (банки 0 ... C) рассматриваются как единая область с последовательно расположенными адресами. Если двоичное значение адреса с символом @ превышает 32767, считается, что указан адрес в расширенной памяти данных (EM), начинающийся с адреса 00000 в банке 0.</p> <p><b>Пример:</b> Если слово памяти данных (DM) содержит 32768, считается, что указан адрес E1_00000 в банке 0 расширенной памяти данных (EM).</p> <p><b>Примечание</b> Если для номера банка расширенной области памяти (EM) указано "n", а двоичное значение слова превышает 32767, считается, что указан адрес расширенной области памяти (EM), начинающийся с адреса 00000 в банке N+1.</p> <p><b>Пример:</b> Если банк 2 в расширенной памяти данных (EM) содержит значение 32768, считается, что указан адрес E3_00000 в банке 3 расширенной памяти данных (EM).</p>		

Операнд	Описание	Примечание	Примеры применения
Указание косвенных адресов областей DM/EM в формате BCD.	<p>В данном случае для адресации указывается величина смещения от начала области. Данные в формате BCD (0000 ... 9999), содержащиеся по указанному адресу, указывают на адрес слова в памяти данных (DM) или в области расширенной памяти данных (EM). Чтобы указать косвенный адрес в формате BCD, впереди адреса следует указывать символ (*).</p> <p>*D□□□□□□</p> <p>↓</p> <p>Значение в формате BCD 00000 ... 9999 (BCD)</p> <p>↓</p> <p>D □□□□□□</p>	<p>*D00200</p> <p>0 1 0 0</p> <p>↓</p> <p>Значение в формате BCD</p> <p>↓</p> <p>Адрес D0100</p> <p>↓</p> <p>Добавляется символ "*".</p>	MOV #0001 *D00200

Операнд	Описание		Прим.	Примеры применения
Непосредственное указание регистра	Для обращения к регистру индексов (IR) или регистру данных (DR) указывается непосредственно значение IR□ (□ : 0 ... 15) или DR□ (□ : 0 ... 15).		IR0 IR1	MOVR 000102 IR0 Содержит адрес памяти ПЛК для CIO0010 в IR0. MOVR 0010 IR1 Содержит адрес памяти ПЛК для CIO0010 в IR1.
Указание косвенного адреса с помощью регистра	Косвенный адрес (без смещения)	Обращение к биту или слову в памяти ПЛК, адрес которого содержится в IR□. Для обращения к битам и словам в операндах команд следует указывать: ,IR□.	,IR0  ,IR1	LD ,IR0 Загружает бит в памяти ПЛК с адресом, который содержится в IR0. MOV #0001 ,IR1 Записывает значение #0001 в слово в памяти ПЛК с адресом в IR1.
	Фиксированное смещение	Обращение к биту или слову в памяти ПЛК, адрес которого определяется как IR□ + или - постоянное значение. Указывается: +/- константа ,IR□. Константа может находиться в диапазоне от -2048 до +2047 (десятичный формат). Значение константы преобразуется в двоичный формат при выполнении команды.	+5,IR0  +31,IR1	LD +5 ,IR0 Загружает бит в памяти ПЛК с адресом в IR0 + 5. MOV #0001 +31 ,IR1 Записывает значение #0001 в слово в памяти ПЛК с адресом в IR1 + 31.
	Смещение в DR	Обращение к биту или слову в памяти ПЛК, адрес которого определяется как IR□ + содержимое DR□. Указывается: DR□, IR□. Содержимое регистра данных (DR) обрабатывается как двоичное число со знаком. Если двоичное число со знаком является отрицательным, сдвиг от IR□ будет сделан в обратном направлении.	DR0 ,IR0  DR0 ,IR1	LD DR0 ,IR0 Загружает бит в памяти ПЛК с адресом: IR0 + значение в DR0. MOV #0001 DR0 ,IR1 Записывает значение #0001 в слово в памяти ПЛК с адресом: IR1 + значение в DR0.
	Автоматический сдвиг вперед на 1	После обращения к значению как к адресу памяти ПЛК содержимое IR□ автоматически увеличивается на +1 или +2. +1: указывается: ,IR□+ +2: указывается: ,IR□++	,IR0 ++  ,IR1 +	LD ,IR0 ++ Увеличивает содержимое IR0 на 2 после загрузки бита в памяти ПЛК с адресом в IR0. MOV #0001 ,IR1 + Увеличивает содержимое IR1 на 1 после записи значения #0001 в слово в памяти ПЛК с адресом в IR1.
	Автоматический сдвиг назад на 1	После обращения к значению как к адресу памяти ПЛК содержимое IR□ автоматически уменьшается на -1 или -2. -1: указывается: ,-IR□ -2: указывается: ,-IR□	,-IR0  ,-IR1	LD ,-IR0 После уменьшения содержимого IR0 на 2 загружается бит памяти ПЛК с адресом в IR0. MOV #0001 ,-IR1 После уменьшения содержимого IR1 на 1 записывается значение #0001 в слово памяти ПЛК с адресом в IR1.



Данные	Операнд	Формат данных	Символ	Диапазон	Пример применения																																								
16-битовая константа	Любые двоичные данные или ограниченный диапазон двоичных данных	Двоичное без знака	##0000 ... #FFFF	---																																									
		Десятичное со знаком	±	-32768 ... +32767	---																																								
		Десятичное без знака	& (см. прим.)	&0 ... &65535	---																																								
	Любые BCD данные или ограниченный диапазон BCD данных	BCD	#	#0000 ... #9999	---																																								
32-битовая константа	Любые двоичные данные или ограниченный диапазон двоичных данных	Двоичное без знака	#	#00000000 ... #FFFFFFFF	---																																								
		Двоичное со знаком	+	-2147483648 ... +2147483647	---																																								
		Десятичное без знака	& (см. прим.)	&0 ... &429467295	---																																								
	Любые BCD данные или ограниченный диапазон BCD данных	BCD	#	#00000000 ... #99999999	---																																								
Текстовая строка	<b>Описание</b>		<b>Символ</b>	<b>Примеры</b>	---																																								
	<p>Данные текстовой строки хранятся в ASCII формате (1 символ - 1 байт, за исключением спецсимволов), располагаясь слева направо и справа (самый младший байт) налево.</p> <p>В случае нечетного количества символов младший байт (крайний правый байт) последнего слова содержит значение 00 Hex (код NUL).</p> <p>В случае четного количества символов в крайнем левом и в крайнем правом свободных байтах последнего слова +1 хранится значение 0000 Hex (2 кода NUL).</p>		---	<p>'ABCDE'</p> <table border="1"> <tr><td>'A'</td><td>'B'</td></tr> <tr><td>'C'</td><td>'D'</td></tr> <tr><td>'E'</td><td>NUL</td></tr> </table> <p>  </p> <table border="1"> <tr><td>41</td><td>42</td></tr> <tr><td>43</td><td>44</td></tr> <tr><td>45</td><td>00</td></tr> </table> <p>'ABCD'</p> <table border="1"> <tr><td>'A'</td><td>'B'</td></tr> <tr><td>'C'</td><td>'D'</td></tr> <tr><td>NUL</td><td>NUL</td></tr> </table> <p>  </p> <table border="1"> <tr><td>41</td><td>42</td></tr> <tr><td>43</td><td>44</td></tr> <tr><td>00</td><td>00</td></tr> </table> <p>MOV\$ D00100 D00200</p> <table border="1"> <tr><td>D00100</td><td>41</td><td>42</td></tr> <tr><td>D00101</td><td>43</td><td>44</td></tr> <tr><td>D00102</td><td>45</td><td>00</td></tr> </table> <p>↓</p> <table border="1"> <tr><td>D00200</td><td>41</td><td>42</td></tr> <tr><td>D00201</td><td>43</td><td>44</td></tr> <tr><td>D00202</td><td>45</td><td>00</td></tr> </table>	'A'	'B'	'C'	'D'	'E'	NUL	41	42	43	44	45	00	'A'	'B'	'C'	'D'	NUL	NUL	41	42	43	44	00	00	D00100	41	42	D00101	43	44	D00102	45	00	D00200	41	42	D00201	43	44	D00202	45
'A'	'B'																																												
'C'	'D'																																												
'E'	NUL																																												
41	42																																												
43	44																																												
45	00																																												
'A'	'B'																																												
'C'	'D'																																												
NUL	NUL																																												
41	42																																												
43	44																																												
00	00																																												
D00100	41	42																																											
D00101	43	44																																											
D00102	45	00																																											
D00200	41	42																																											
D00201	43	44																																											
D00202	45	00																																											
<p>В состав символов ASCII, которые можно использовать в текстовой строке, входят алфавитно-цифровые символы, алфавит Катакана и символы (за исключением спецсимволов). Символы представлены в следующей таблице.</p>																																													

**Примечание** Десятичное число без знака только для CX-Programmer.

ASCII символы

Биты 0 ... 3		Биты 4 ... 7															
Двоичный код	Hex	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0			Пробел	0	@	P	`	p				0	@	P		
0001	1			!	1	A	Q	a	q			!	1	A	Q		
0010	2			"	2	B	R	b	r			"	2	B	R		
0011	3			#	3	C	S	c	s			#	3	C	S		
0100	4			\$	4	D	T	d	t			\$	4	D	T		
0101	5			%	5	E	U	e	u			%	5	E	U		
0110	6			&	6	F	V	f	v			&	6	F	V		
0111	7			'	7	G	W	g	w			'	7	G	W		
1000	8			(	8	H	X	h	x			(	8	H	X		
1001	9			)	9	I	Y	i	y			)	9	I	Y		
1010	A			*	:	J	Z	j	z			*	:	J	Z		
1011	B			+	;	K	[	k	{			+	;	K	[		
1100	C			,	<	L	\	l				,	<	L	\		
1101	D			-	=	M	]	m	}			-	=	M	]		
1110	E			.	>	N	^	n	~			.	>	N	^		
1111	F			/	?	O	_	o				/	?	O	_		

### 2-1-6 Форматы данных

В следующей таблице перечислены форматы данных, с которыми могут работать ПЛК серии CS/CJ.

Тип данных	Формат данных	Десятичное значение	4-разрядное 16-ричное значение
Двоичное, без знака	<p>Двоичное: <math>2^{15} \dots 2^0</math></p> <p>Десятичное: 32768 ... 1</p> <p>Hex: <math>2^3 \dots 2^0</math></p>	0 ... 65535	0000 ... FFFF
Двоичное, со знаком	<p>Двоичное: <math>2^{15} \dots 2^0</math></p> <p>Десятичное: 32768 ... 1</p> <p>Hex: <math>2^3 \dots 2^0</math></p> <p>Бит знака: 0: положительное, 1: отрицательное</p>	-32768 ... +32767	8000 ... 7FFF
BCD (двоично-десятичный код)	<p>Двоичное: <math>2^3 \dots 2^0</math></p> <p>Десятичное: 0 ... 9</p>	0 ... 9999	0000 ... 9999

Тип данных	Формат данных	Десятичное значение	4-разрядное 16-ричное значение
Десятичное с плавающей точкой, одинарной точности	<p>Значение = <math>(-1)^{\text{Знак}} \times 1.[\text{Мантисса}] \times 2^{\text{Показатель}}</math></p> <p>Знак (бит 31) 1: отрицательный или 0: положительный</p> <p>Мантисса 23 бита (биты 00 ... 22) содержат двоичное значение мантиссы, то есть, той части, которая находится за десятичной точкой в числе 1.000 .....,</p> <p>Показатель 8 битов (биты 23 ... 30) содержат значение показателя. Показатель представляется в двоичном коде как <math>127 + n</math> в <math>2^n</math>.</p> <p><b>Примечание</b> Этот формат соответствует стандартам IEEE754 на данные, представляемые в формате с плавающей запятой, одинарной точности, и предназначен только для команд, которые осуществляют преобразования или вычисления с числами в формате с плавающей запятой. Его можно использовать для ввода или контроля значений на экране редактирования и контроля памяти ввода/вывода (Edit and Monitor Screen) в пакете CX-Programmer (консолями программирования не поддерживается). Структуру формата пользователю знать необязательно, однако ему следует учитывать, что значения в этом формате занимают два слова.</p>	---	---
Десятичное с плавающей точкой, двойной точности	<p>Значение = <math>(-1)^{\text{Знак}} \times 1.[\text{Мантисса}] \times 2^{\text{Показатель}}</math></p> <p>Знак (бит 63) 1: отрицательный или 0: положительный</p> <p>Мантисса 52 бита (биты 00 ... 51) содержат двоичное значение мантиссы, то есть, той части, которая находится за десятичной точкой в числе 1.000 .....,</p> <p>Показатель 11 битов (биты 52 ... 62) содержат значение показателя. Показатель представляется в двоичном коде как <math>1023 + n</math> в <math>2^n</math>.</p> <p><b>Примечание</b> Этот формат соответствует стандартам IEEE754 на данные, представляемые в формате с плавающей запятой, двойной точности, и предназначен только для команд, которые осуществляют преобразования или вычисления с числами в формате с плавающей запятой. Его можно использовать для ввода или контроля значений на экране редактирования и контроля памяти ввода/вывода (Edit and Monitor Screen) в пакете CX-Programmer (консолями программирования не поддерживается). Структуру формата пользователю знать необязательно, однако ему следует учитывать, что значения в этом формате занимают два слова.</p>	---	---

**Двоичные данные со знаком**

У данных в двоичном формате со знаком старший (крайний левый) бит соответствует знаку двоичного 16-битового числа. Значение представляется в формате 4-разрядного 16-ричного числа.

**Положительные значения:** значение является положительным или нулевым (0), если старший бит = 0 (ВЫКЛ). В этом случае 4-разрядное 16-ричное число находится в диапазоне 0000 ... 7FFF Hex.

**Отрицательные значения:** значение является отрицательным, если старший бит = 1 (ВКЛ). В этом случае 4-разрядное 16-ричное число находится в пределах 8000 ... FFFF Hex. Для представления абсолютного (десятичного) значения отрицательного числа используется операция дополнения до двух.

**Пример:** чтобы представить десятичное число -19 в двоичном формате со знаком, значение 0013 Hex (модуль числа -19) отнимается от FFFF Hex, после чего к результату добавляется 0001 Hex, в результате чего получается FFED Hex.

	F	F	F	F
	1111	1111	1111	1111
Действительное значение	0	0	1	3
—)	0000	0000	0001	0011
	F	F	E	C
	1111	1111	1110	1100
	0	0	0	1
+)	0000	0000	0000	0001
Дополнение до двух	F	F	E	D
	1111	1111	1110	1101

**Операция дополнения**

В общем случае под дополнением до основания x понимается операция, при которой все разряды данного числа вычитаются из значения x-1, после чего в младший разряд добавляется 1 (пример: дополнение числа 7556 до 10: 9999 - 7556 + 1 = 2444). Операция дополнения используется для представления таких операций, как вычитание, в виде операции добавления.

**Пример:** 8954 - 7556 = 1398, 8954 + (дополнение 7556 до 10) = 8954 + 2444 = 11398. Если отбросить старший бит, получаем результат вычитания 1398.

**Дополнение до двух**

Дополнение до двух - это дополнение до основания "2". В данном случае все разряды вычитаются из числа 1 (2 - 1 = 1), после чего добавляется число 1.

**Пример:** дополнение двоичного числа 1101 до двух: 1111 (F Hex) - 1101 (D Hex) + 1 (1 Hex) = 0011 (3 Hex). Ниже показано, как это значение представляется в виде 4-разрядного 16-ричного числа.

Дополнение числа a Hex до двух = FFFF Hex - a Hex + 0001 Hex = b Hex. Чтобы определить дополнение числа "a Hex" до двух и получить b Hex, следует использовать операцию: b Hex = 10000 Hex - a Hex.

**Пример:** чтобы определить дополнение числа 3039 Hex до двух, следует выполнить операцию: 10000 Hex - 3039 Hex = CFC7 Hex.

Точно так же, чтобы определить значение a Hex по числу b Hex, являющемуся дополнением числа a Hex до двух, следует использовать выражение: a Hex = 10000 Hex - b Hex.

**Пример:** чтобы определить действительное значение от числа CFC7 Hex, являющегося дополнением до двух, следует использовать выражение: 10000 Hex - CFC7 Hex = 3039 Hex.

В ПЛК серии CS/CJ предусмотрены две команды: NEG (160) (дополнение до двух) и NEGL (161) (двойное дополнение до двух). Эти команды можно использовать для определения дополнения до двух от действительного значения и определения действительного значения по значению, являющемуся результатом дополнения до двух действительного значения.

**Данные в формате BCD со знаком**

BCD-формат со знаком - это специальный формат данных, который предназначен для представления отрицательных значений в формате BCD. Хотя этот формат и находит применение, он не имеет строгого определения и зависит от конкретного случая использования. В ПЛК серии CS/CJ предусмотрены следующие команды для конвертации форматов данных: BCD со знаком в двоичный формат: BINS(470), двойной BCD-формат со знаком в двоичный формат: BISL(472),

Двоичное значение со знаком в BCD: BCDS(471) и двойное двоичное значение со знаком в BCD: BDSL(473). Более подробную информацию можно найти в руководстве *CS/CJ-series Programmable Controllers Instructions Reference Manual (W340)*.

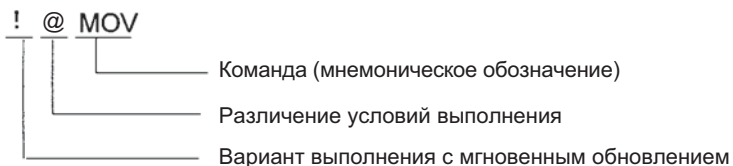
Десятичное значение	16-ричное значение	Двоичное значение	BCD	
0	0	0000		0000
1	1	0001		0001
2	2	0010		0010
3	3	0011		0011
4	4	0100		0100
5	5	0101		0101
6	6	0110		0110
7	7	0111		0111
8	8	1000		1000
9	9	1001		1001
10	A	1010	0001	0000
11	B	1011	0001	0001
12	C	1100	0001	0010
13	D	1101	0001	0011
14	E	1110	0001	0100
15	F	1111	0001	0101
16	10	10000	0001	0110

Десятичное значение	Двоичное значение без знака (4-разрядное 16-ричное число)	Двоичное значение со знаком (4-разрядное 16-ричное число)
+65535	FFFF	Не может быть представлено.
+65534	FFFE	
.	.	
.	.	
.	.	
+32769	8001	
+32768	8000	
+32767	7FFF	7FFF
+32766	7FFF	7FFF
.	.	
.	.	
.	.	
+2	0002	0002
+1	0001	0001
0	0000	0000
-1	Не может быть представлено.	FFFF
-2		FFFE
.		
.		
.		
-32767		8001
-32768	8000	

### 2-1-7 Варианты выполнения команд

В следующей таблице перечислены варианты выполнения команд, которые позволяют различать условия выполнения команд и обновлять данные при выполнении команд (мгновенное обновление).

Вариант выполнения	Символ	Описание
Определение направления переключения	ВКЛ	@ Команда выполняется по включению условия выполнения.
	ВЫКЛ	% Команда выполняется по сбросу условия выполнения.
Мгновенное обновление	!	Обновление данных в области ввода/вывода, указанной с помощью операндов, или слов специального модуля ввода/вывода происходит при выполнении команды.  (Мгновенное обновление не поддерживается модулями CPU CS1D для систем с дублированием CPU)



### 2-1-8 Условия выполнения

В ПЛК серии CS/CJ предусмотрены следующие типы основных и специальных команд.

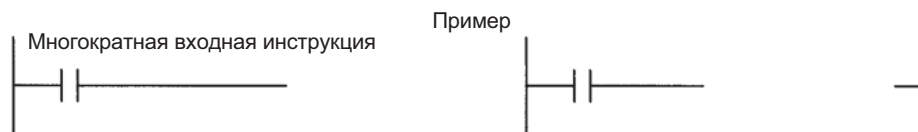
- Многократные (недифференцированные) команды выполняются в каждом цикле
- Однократные (дифференцированные) команды выполняются только один раз

#### Многократные команды

Выходные команды, для которых требуются условия выполнения, выполняются один раз в каждом цикле, когда действительно условие их выполнения (ВКЛ или ВЫКЛ).



Входные команды, которыми формируются логические условия для начала выполнения, и промежуточные команды выполняют чтение состояний битов, выполняют операции сравнения, проверяют состояние битов или выполняют другие типы операций в каждом цикле. В случае положительного результата (ВКЛ на выходе) открывается "путь для тока" (т.е., включается условие выполнения).



### Входные однократные (дифференцированные) команды

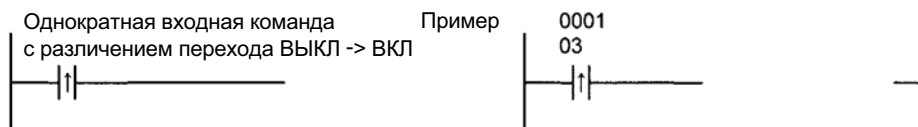
Однократные команды с различием переключения из ВЫКЛ во ВКЛ (команда, начинающаяся с @)

- **Выходные команды:** команда выполняется только в том цикле, в котором установилось (ВЫКЛ → ВКЛ) условие выполнения, и не выполняется в остальных циклах.



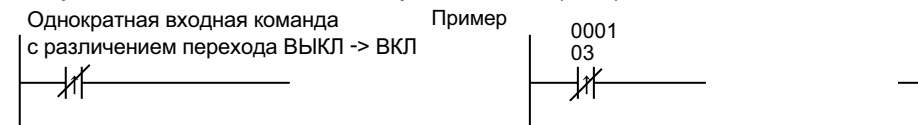
Команда MOV выполняется однократно, когда СЮ 000102 переходит из ВЫКЛ во ВКЛ.

- **Входные команды (логические условия начала выполнения и промежуточные команды):** команда выполняет чтение состояния бита, выполняет операции сравнения, проверяет состояние битов или выполняет другие операции в каждом цикле и устанавливает (ВКЛ) на выходе условие выполнения (открывает "путь для тока"), когда результат операции переходит из ВЫКЛ во ВКЛ. В следующем цикле условие выполнения вновь сбрасывается (ВЫКЛ).



Положительное (ВКЛ) условие выполнения формируется только для одного цикла, когда СЮ 000103 переходит из ВЫКЛ во ВКЛ.

- **Входные команды (логические условия начала выполнения и промежуточные команды):** команда выполняет чтение состояния бита, выполняет операции сравнения, проверяет состояния битов или выполняет другие операции в каждом цикле и сбрасывает (ВЫКЛ) с выхода условие выполнения (закрывает "путь для тока"), когда результат операции переключается из ВЫКЛ во ВКЛ. В следующем цикле условие выполнения вновь установится (ВКЛ).



Отрицательное (ВЫКЛ) условие выполнения формируется только для одного цикла, когда СЮ 000103 переходит из ВЫКЛ во ВКЛ.

Однократные команды с различием переключения из ВКЛ в ВЫКЛ (команда, начинающаяся с %)

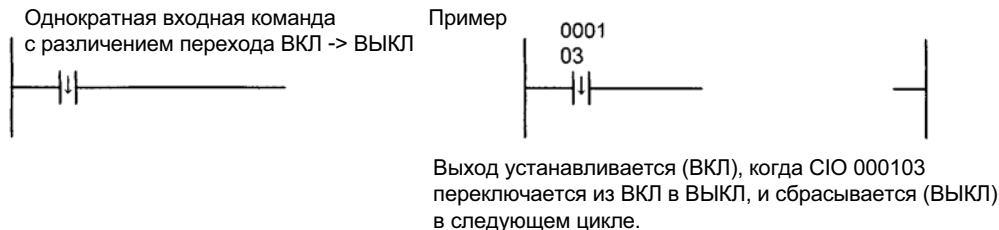
- **Выходные команды:** команда выполняется только в том цикле, в котором сбросилось (ВЫКЛ) (ВКЛ → ВЫКЛ) условие выполнения, и не выполняется в остальных циклах.



Команда SET выполняется однократно, когда СЮ 000102 переходит из ВКЛ в ВЫКЛ.

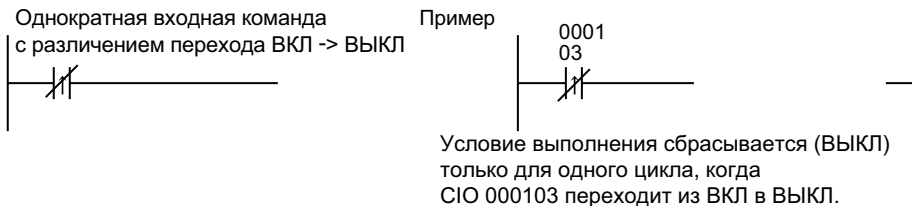


- **Входные команды (логические условия начала выполнения и промежуточные команды):** команда выполняет чтение состояния бита, выполняет операции сравнения, проверяет состояния битов или выполняет другие операции в каждом цикле и устанавливает (ВКЛ) на выходе условие выполнения (открывает "путь для тока"), когда результат операции переключается из ВКЛ в ВЫКЛ. В следующем цикле условие выполнения вновь сбрасывается (ВЫКЛ).



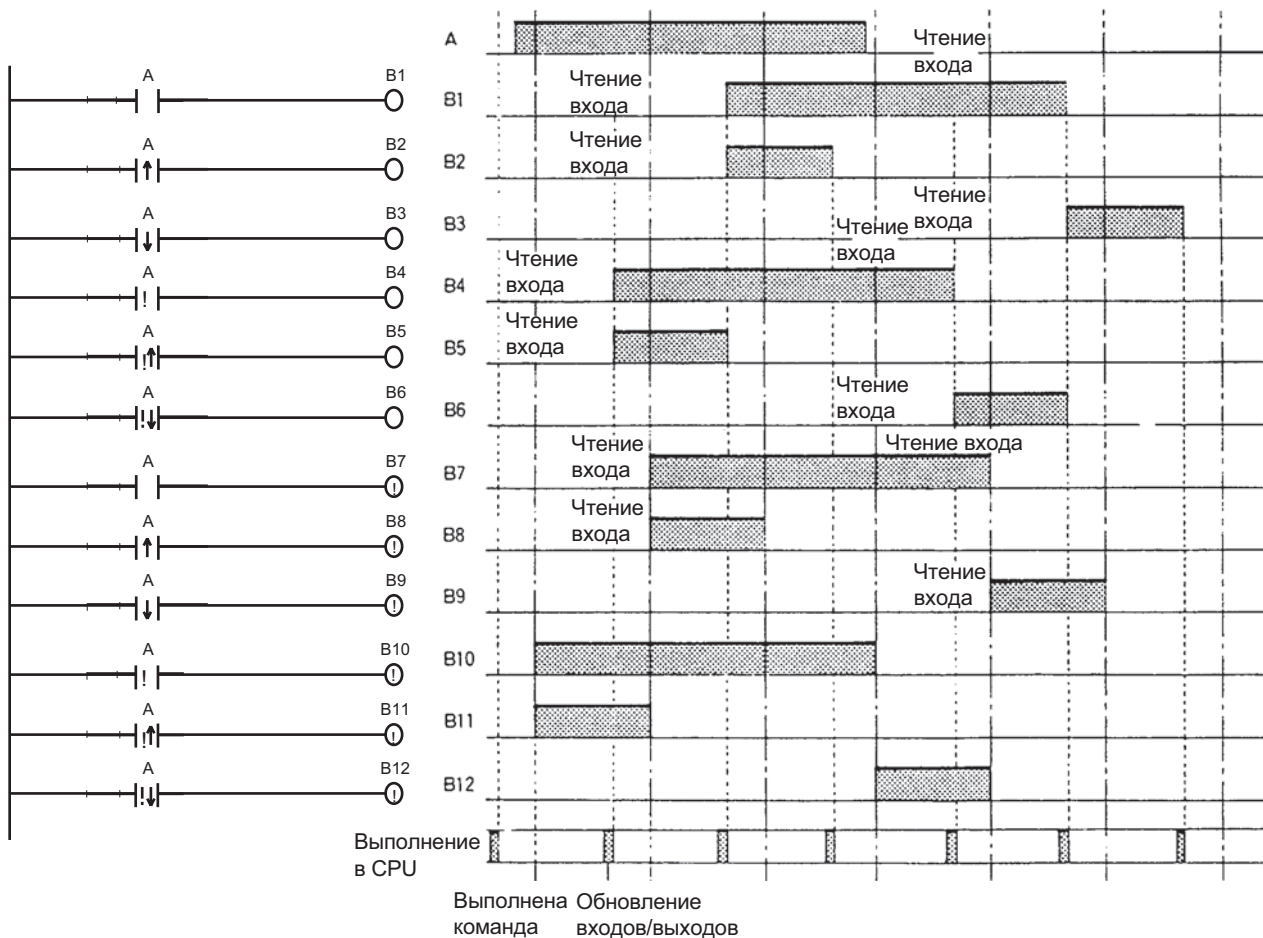
**Примечание** В отличие от однократных (дифференцированных) команд с различием переключения из ВЫКЛ во ВКЛ, вариант выполнения с различием перехода из ВКЛ в ВЫКЛ можно выбрать только для команд LD, AND, OR, SET и RSET. Чтобы реализовать вариант с выделением перехода из ВКЛ в ВЫКЛ для других команд, следует объединить эти команды с командой DIFD или DOWN. NOT можно добавлять к командам только в случае использования модулей CPU CS1-H, CJ1-H, CJ1M или CS1D.

- **Входные команды (логические условия начала выполнения и промежуточные команды):** команда выполняет чтение состояния бита, выполняет операции сравнения, проверяет состояния битов или выполняет другие операции в каждом цикле и сбрасывает (ВЫКЛ) с выхода условие выполнения (закрывает "путь для тока"), когда результат операции переключается из ВКЛ в ВЫКЛ. В следующем цикле условие выполнения вновь установится (ВКЛ).



### 2-1-9 Синхронизация команд ввода/вывода

Ниже приведена временная диаграмма, на которой наглядно показана синхронизация работы отдельных команд на примере программы, состоящей только из команд LD и OUT.

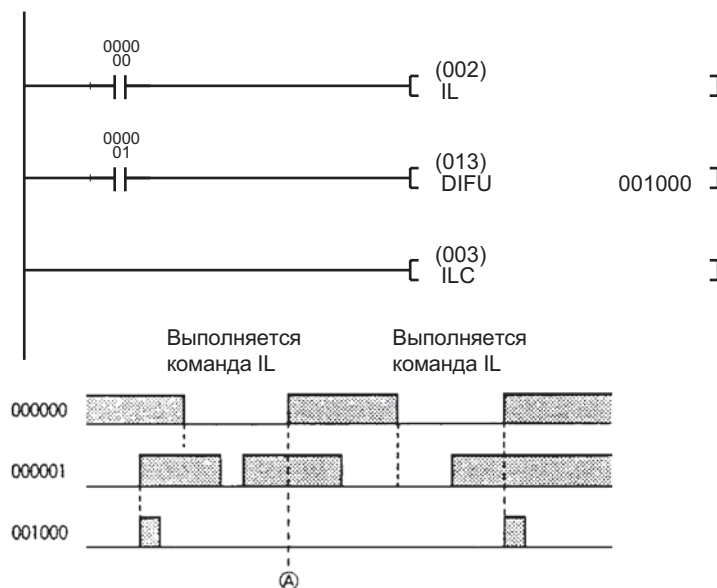


### Однократные (дифференцированные) команды

- У однократной команды имеется внутренний флаг, в котором хранится предыдущее состояние (ВКЛ или ВЫКЛ). В начале работы флаги предшествующих состояний для однократных команд, различающих переключение ВЫКЛ -> ВКЛ (DIFU и команды с префиксом @), устанавливаются в состояние ВКЛ, а флаги предшествующих значений для однократных команд, различающих переключение ВКЛ -> ВЫКЛ (DIFD и команды с префиксом %), сбрасываются в ВЫКЛ. Это позволяет предотвратить включение выходов команд в начале работы.
- На выходе однократной команды, различающей переключение ВЫКЛ -> ВКЛ (DIFU и команды с префиксом @), состояние ВКЛ устанавливается только в том случае, когда включено условие выполнения и сброшен флаг предшествующего значения.

• **Применение с командами-блокировками (команды IL - ILC)**

Ниже приведен пример программы, в которой флаг предшествующего значения для однократной (дифференцированной) команды хранит предшествующее заблокированное значение и дифференцированный выход в точке A не включается, поскольку значение не обновляется, пока действует блокировка.



• **Применение в командах-переходах (команды JMP - JME):** как и в случае блокировок, флаг предшествующего значения для однократной (дифференцированной) команды не изменяется, когда команда обходится, то есть, сохраняется предыдущее значение. На выходе однократных команд, различающих переключение ВЫКЛ -> ВКЛ и ВКЛ -> ВЫКЛ, условие выполнения устанавливается (ВКЛ) только в том случае, если на входе появляется состояние, отличающееся от состояния, которое сигнализируется флагом предшествующего значения.

- Примечание**
- a) Для однократной команды, различающей переключение ВЫКЛ -> ВКЛ, не следует использовать в качестве входного бита флаг "всегда ВКЛ" или A20011 (флаг "Первый цикл"). Команда никогда не будет выполнена.
  - b) Для однократной команды, различающей переключение ВКЛ -> ВЫКЛ, не следует использовать в качестве входного бита флаг "всегда ВЫКЛ". Команда никогда не будет выполнена.

### 2-1-10 Синхронизация обновления

Внешние входы/выходы могут обновляться в одном из следующих режимов

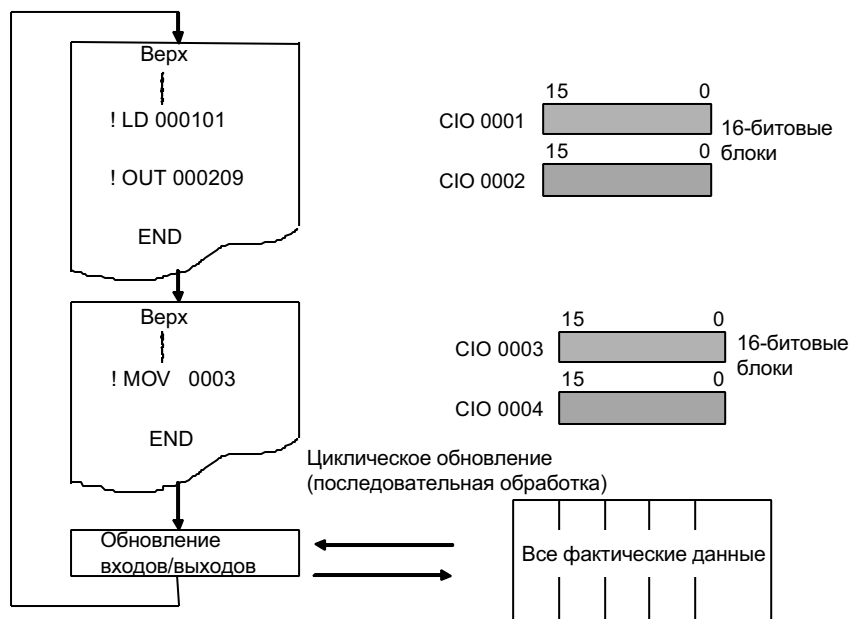
- Циклическое обновление
- Мгновенное обновление (команда с символом "!", команда IORF)

Подробное описание обновления входов/выходов содержится в разделе "Работа модуля CPU" в Руководстве CS/CJ Series Operation Manual

#### Циклическое обновление

Каждая программа, назначенная готовой циклической задаче или задаче, для которой сформировано прерывание, выполняется, начиная с начального адреса программы вплоть до команды END (001). После того, как все готовые циклические задачи или задачи, для которых сформировано прерывание, выполнены, процедура циклического обновления выполняет обновление одновременно всех точек ввода/вывода.

**Примечание** Программы могут состоять из нескольких задач. Обновление входов/выходов произойдет после того, как будет достигнута самая последняя команда END (001) в программе, принадлежащей циклической задаче с наибольшим номером (среди всех готовых циклических задач), и не происходит, когда достигаются команды END (001) в программах, назначенных другим циклическим задачам.



Если в других задачах требуется выполнить обновление входов/выходов, для всех требуемых слов, предшествующих команде END (001), следует выполнить команду IORF.

#### Мгновенное обновление

**Выполнение команд с мгновенным обновлением**

Ниже описано, как будет происходить обновление входов/выходов во время выполнения команды, если в качестве операнда указан физический (фактический) бит ввода/вывода.

Модули	Обновление данных
Базовые модули ввода/вывода C200H (только серия CS)	Будет обновлено 16 битов ввода/вывода, содержащих указанный бит.
Базовые модули ввода/вывода CJ	

- Если для команды в качестве операнда указано слово, обновление входов/выходов будет выполнено для 16 битов.
- Обновление входов или операндов-источников будет выполнено непосредственно перед выполнением команды.
- Обновление выходов или операндов-адресатов (D) будет выполнено сразу же после выполнения команды.

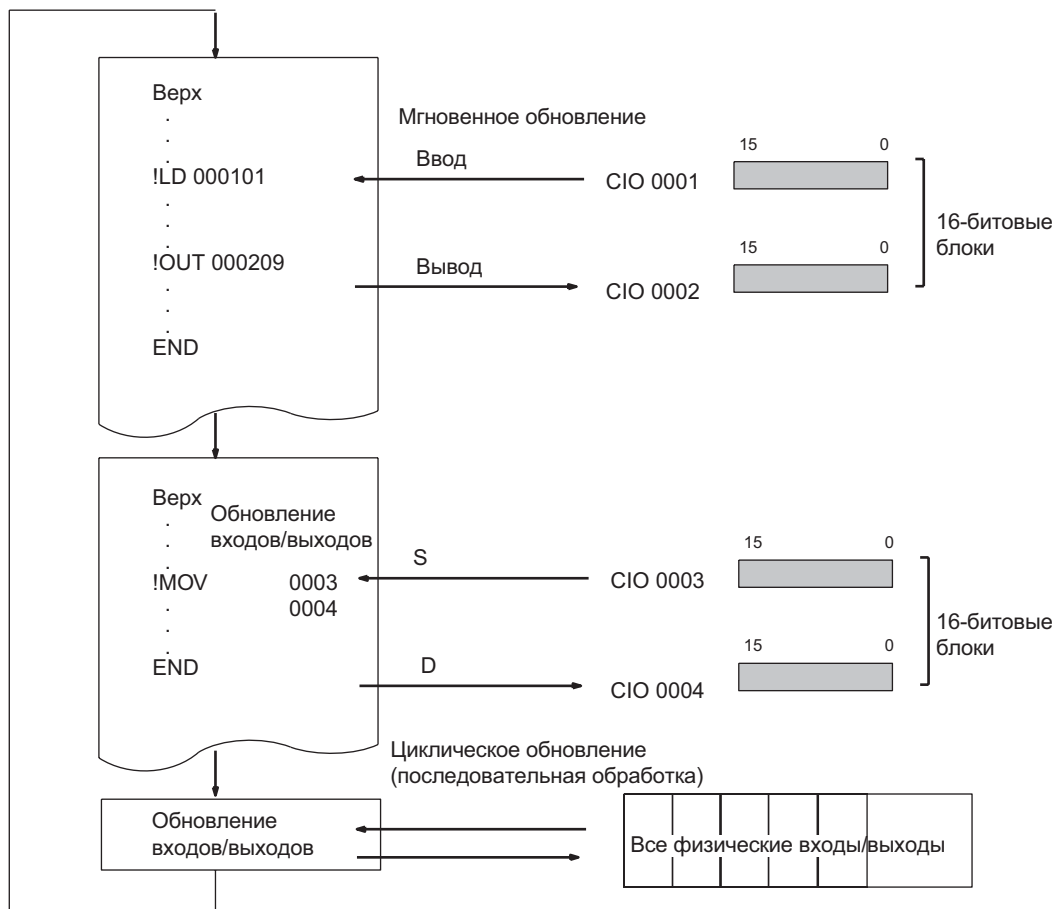
Перед командой следует добавлять восклицательный знак (!) (вариант мгновенного обновления).

**Примечание** Мгновенное обновление не поддерживается модулями CPU CS1D для систем с дублированием CPU, однако они поддерживают обновление с помощью команд IORF(097) и DLNK(226).

**Модули, поддерживающие команду I/O REFRESH (обновление входов/выходов)**

<b>Размещение</b>	CPU или стойка расширения ввода/вывода (кроме стоек ведомых устройств SYSMAC BUS)		
<b>Модули</b>	Базовые модули ввода/вывода	Базовые модули ввода/вывода серии CS/CJ	Обновляются
		Базовые модули ввода/вывода C200H (см. прим.)	Обновляются
		Модули ввода/вывода C200H, группа 2 (большая плотность точек ввода/вывода) (см. прим.)	Не обновляются
		Специальные модули ввода/вывода	Не обновляются

**Примечание** Модули ввода/вывода C200H не могут быть установлены в ПЛК серии CJ.



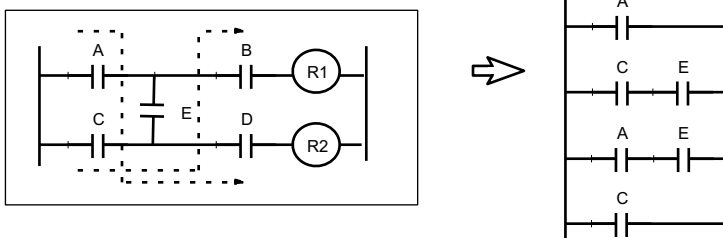
**Модули, поддерживающие команды IORF(097) или DLNK(226)**

В качестве специальной команды предусмотрена команда I/O REFRESH (IORF(097)), которая обновляет физические входы/выходы в указанном диапазоне слов. С помощью этой команды в пределах цикла можно обновить все физические входы/выходы или входы/выходы в указанном диапазоне. IORF можно также использовать для обновления слов, зарезервированных за специальными модулями ввода/вывода.

Для обновления слов, отведенных модулям шины CPU в областях CIO и DM, а также для выполнения специального обновления для модуля, например, для обновления логических связей, предусмотрена другая команда: CPU BUS UNIT REFRESH (DLNK(226)). DLNK(226) поддерживается только модулями CPU CS1-H, CJ1-H, CJ1M или CS1D.

**Модули, поддерживающие IORF(097)**

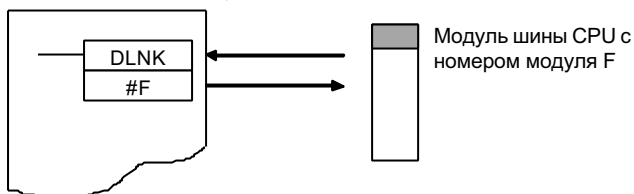
<b>Размещение</b>	Стойка CPU или стойка расширения ввода/вывода (кроме стоек ведомых устройств SYSMAC BUS)		
<b>Модули</b>	Базовые модули ввода/вывода	Базовые модули ввода/вывода серии CS/CJ	Обновляются
		Базовые модули ввода/вывода C200H	Обновляются
		Модули ввода/вывода C200H, группа 2 (большая плотность точек ввода/вывода)	Обновляются
	Специальные модули ввода/вывода	Обновляются	
Модули шины CPU	Не обновляются		



**Модули, поддерживающие DLNK(226)**

<b>Размещение</b>	Стойка CPU или стойка расширения ввода/вывода (кроме стоек ведомых устройств SYSMAC BUS)	
<b>Модули</b>	Базовые модули ввода/вывода	Не обновляются
	Специальные модули ввода/вывода	Не обновляются
	Модули шины CPU Слова, отведенные для модуля в области CIO Слова, отведенные для модуля в области DM Специальное обновление для модуля (обновление логических связей для модулей Controller Link и модулей SYSMAC Link или обновление удаленных точек ввода/вывода для модулей DeviceNet)	Обновляются

Слова, зарезервированные в области CIO и в области DM, а также любые операции специального обновления



### 2-1-11 Объем программы

В следующей таблице приведены максимальные размеры программ для различных модулей CPU серии CS/CJ (то есть, суммарная емкость всех задач). Допустимая емкость программы выражается как максимальное количество шагов. Максимальную емкость программы превышать нельзя. При попытке превысить допустимую емкость дальнейшая процедура программирования будет запрещена.

Каждая команда содержит от одного до семи шагов. Точное количество шагов, из которых состоит каждая команда, приведено в разделе 10-5 *Время выполнения команды и количество шагов* в руководстве *Operation Manual* (длина каждой команды увеличивается на один шаг, если используется операнд двойной длины).

Серия	Модуль CPU	Макс. емкость программы	Точки ввода/вывода
Серия CS	CS1H-CPU67H/CPU67-E	250К шагов	5120
	CS1D-CPU67H	250К шагов	
	CS1D-CPU67S	250К шагов	
	CS1H-CPU66H/CPU66-E	120К шагов	
	CS1H-CPU65H/CPU65-E	60К шагов	
	CS1D-CPU65H	60К шагов	
	CS1D-CPU65S	60К шагов	
	CS1H-CPU64H/CPU64-E	30К шагов	
	CS1H-CPU63H/CPU63-E	20К шагов	
	CS1G-CPU45H/CPU45-E	60К шагов	
	CS1G-CPU44H/CPU44-E	30К шагов	1280
	CS1D-CPU44S	30К шагов	
	CS1G-CPU43H/CPU43-E	20К шагов	960
	CS1G-CPU42H/CPU42-E	10К шагов	
CS1D-CPU42S	10К шагов		
Серия CJ	CJ1H-CPU66H	120К шагов	2560
	CJ1H-CPU65H	60К шагов	
	CJ1G-CPU45H/CPU45	60К шагов	1280
	CJ1G-CPU44H/CPU44	30К шагов	
	CJ1G-CPU43H	20К шагов	960
	CJ1G-CPU42H	10К шагов	
	CJ1M-CPU23/CPU13	20К шагов	640
	CJ1M-CPU22/CPU12	10К шагов	320
CJ1M-CPU11/CPU21	5к шагов	160	

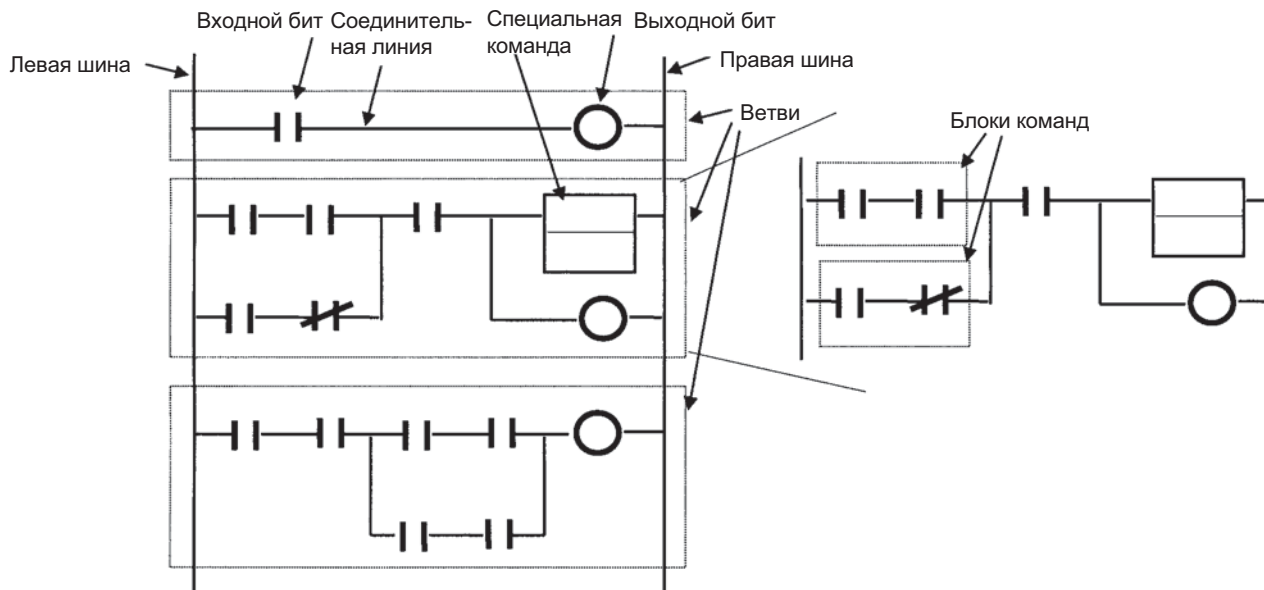
**Примечание** Емкость памяти для ПЛК серии CS/CJ измеряется в шагах, хотя емкость памяти для предыдущих ПЛК производства OMRON, таких как ПЛК серии C200HX/HG/HE и CV, измерялась в словах. Указания по конвертации емкостей программ предшествующих ПЛК производства OMRON содержатся в конце раздела 10-5 *Время выполнения команд и количество шагов* в руководстве *Operation Manual*.

### 2-1-12 Основные принципы программирования "лестничных диаграмм"

Команды выполняются в том порядке, в котором они перечислены в памяти (порядок следования мнемонических обозначений). Необходимо соблюдать основные принципы программирования, а также порядок выполнения.

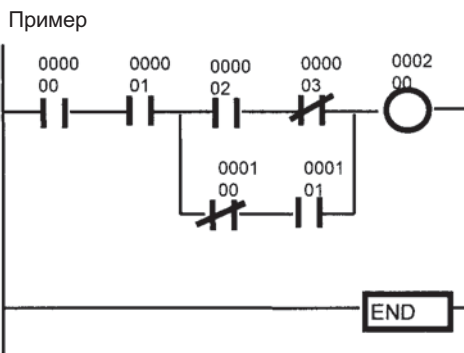
**Общая структура "лестничной диаграммы"**

Программа, составленная на языке релейно-контактной логики (РКЛ-программа или "лестничная диаграмма") состоит из левой и правой объединительных шин, соединительных линий, входных битов, выходных битов и специальных команд. Программа может содержать одну или несколько ветвей. Ветвью программы является блок, который можно отделить от других блоков горизонтальной линией. В случае представления "лестничной диаграммы" в мнемонической форме ветвь объединяет все команды, расположенные между командой LD/LD NOT и выходной командой, которая предшествует следующим командам LD/LD NOT. Ветвь программы состоит из блоков команд, которые начинаются с команды LD/LD NOT, выступающей в качестве логического условия начала выполнения.



**Мнемоническая форма**

В мнемонической форме программа имеет вид последовательности команд "лестничной диаграммы", представленных своими мнемоническими обозначениями. Программа в мнемонической форме содержит адреса, при этом один программный адрес эквивалентен одной команде. Программные адреса состоят из шести разрядов, начиная с 000000.



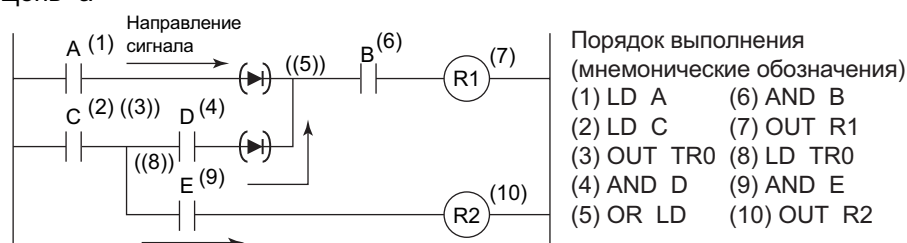
Программный адрес	Команда (мнемоническое обозначение)	Операнд
000000	LD	000000
000001	AND	000001
000002	LD	000002
000003	AND NOT	000003
000004	LD NOT	000100
000005	AND	000101
000006	OR LD	
000007	AND LD	
000008	OUT	000200
000009	END	



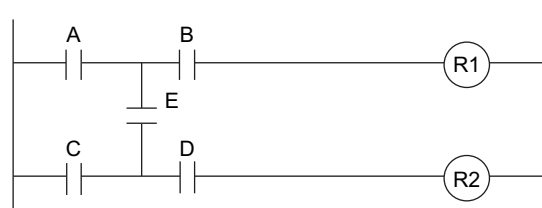
Основные принципы построения РКЛ-программ

- 1,2,3... 1. Когда "лестничная диаграмма" выполняется программируемым логическим контроллером (ПЛК), сигнал ("ток") всегда направлен слева направо. Нельзя использовать программы, в которых "ток" направлен справа налево. Таким образом, протекание сигнала в РКЛ-программах отличается от протекания тока в цепях, реализованных с помощью настоящих реле. Например, если цепь "а" реализована в виде программы ПЛК, ток протекает так, как если бы в цепь были вставлены диоды, показанные в скобках, и катушку R2 нельзя запитать через предусмотренный контакт D. Фактический порядок выполнения показан справа с помощью мнемонических обозначений. Чтобы программа работала без этих воображаемых диодов, цепь необходимо реорганизовать. Точно так же, протекание тока, предусмотренное в цепи "b", нельзя запрограммировать непосредственно, поэтому цепь необходимо реорганизовать.

Цепь "а"

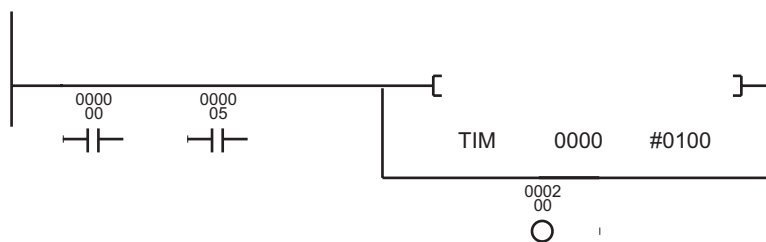


Цепь " b"

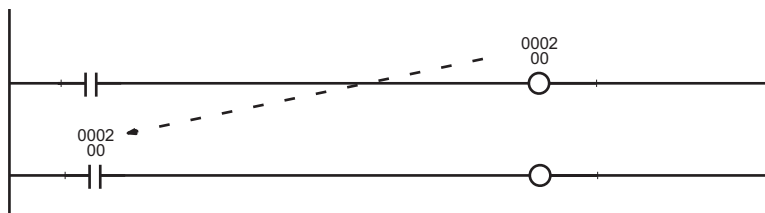


В цепи "а" катушку R2 нельзя запитать через предусмотренный контакт D. Контакт E, включенный в цепь "b", нельзя запрограммировать в "лестничной диаграмме". Программу необходимо видоизменить.

2. Количество входных/выходных битов, рабочих битов, таймеров и других входных битов не ограничено. Однако ветви должны быть как можно более простыми и понятными, даже если это предполагает применение большего количества входных битов. Благодаря этому упрощается отладка программы и последующая модернизация.
3. Количество входных битов, которое может быть включено последовательно или параллельно в последовательных или параллельных ветвях, не ограничено.
4. Два или больше выходных битов можно включить параллельно.



5. Выходные биты также можно использовать в качестве входных битов.

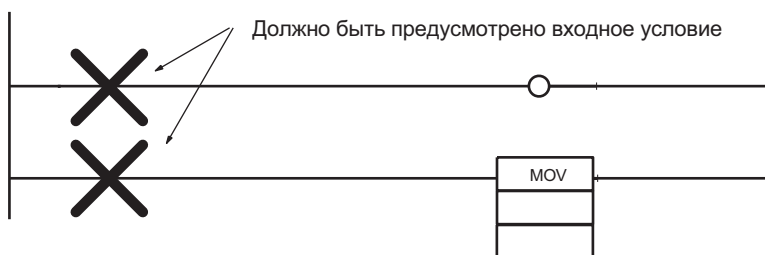


**Ограничение**

- 1,2,3... 1. Лестничная диаграмма (РКЛ-программа) должна быть замкнута таким образом, чтобы ток (сигнал) протекал от левой шины к правой шине. Если программа не замкнута, произойдет ошибка ветви (однако программа может быть выполнена).



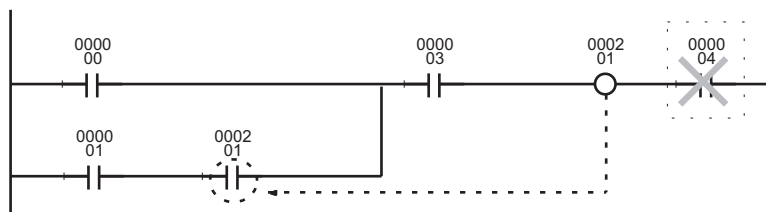
2. Выходные биты, таймеры, счетчики и другие выходные команды нельзя подсоединять непосредственно к левой шине. Если какой-либо из этих элементов подсоединен непосредственно к левой шине, во время проверки программы средством программирования будет обнаружена ошибка ветви (программа может быть выполнена, но команды OUT и MOV(021) выполняться не будут).



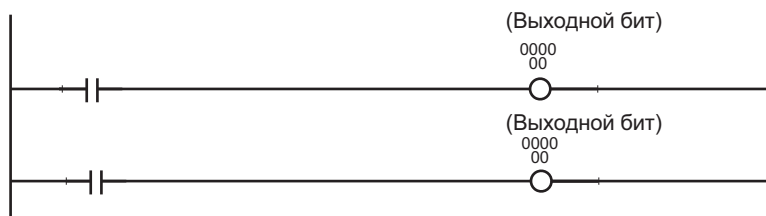
Если вход должен быть включен все время, вставьте неиспользуемый рабочий бит N.C. (нормально-замкнутый бит) или флаг "всегда ВКЛ".



3. Входной бит следует всегда размещать до, а не после выходной команды, например, до выходного бита. Если он вставлен после выходной команды, в этом случае при проверке программы на средстве программирования будет обнаружена ошибка размещения.



4. При создании программы нельзя использовать в выходной команде один и тот же выходной бит. Команды в лестничной диаграмме выполняются по порядку, начиная с верхней ветви, в пределах одного цикла, поэтому выходной бит будет содержать результат выполнения выходной команды самой нижней ветви, а результаты выполнения любых предшествующих команд, управляющих тем же битом, будут перезаписаны и на выходе будут отсутствовать.



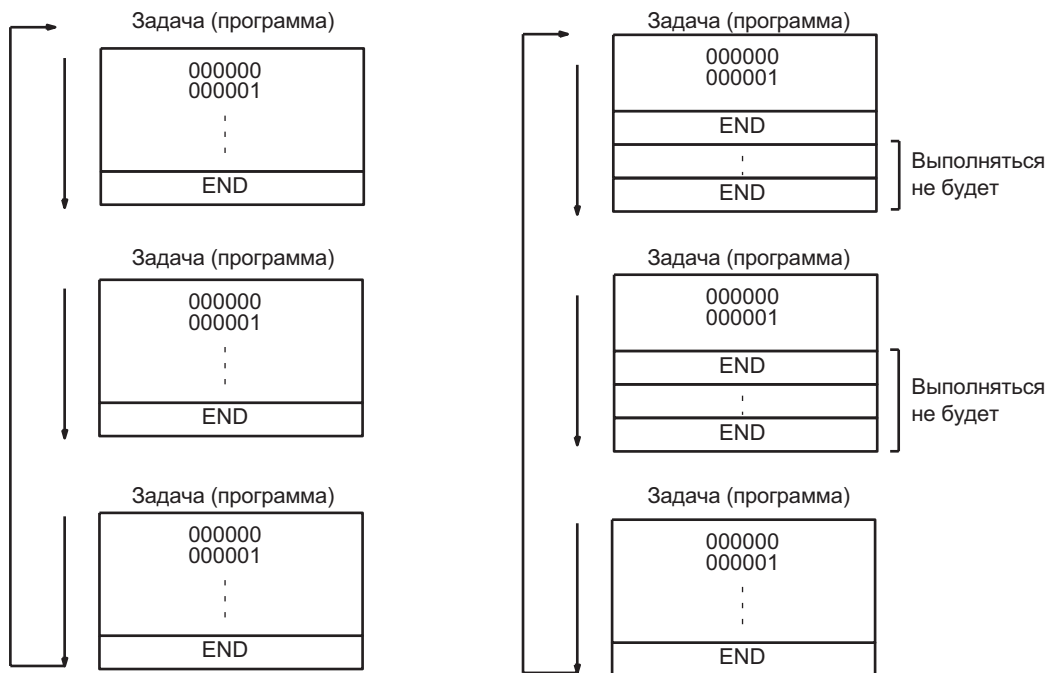
5. Входной бит нельзя использовать в команде OUTPUT (OUT).



6. В конце программы в каждой задаче следует вставлять команду END(001).

- Если начинается выполнение программы, у которой отсутствует команда END(001), произойдет ошибка "No End Instruction" ("Команда без END"), на лицевой панели модуля CPU будет светиться светодиод ERR/ALM и программа выполняться не будет.
- Если в программе имеется несколько команд END(001), программа будет выполнена до первой команды END(001).

- Отладка программы существенно упростится, если команда END(001) вставляется последовательно в различных точках программы, отделяя одни ветви программы от других. После того, как программа проверена, промежуточные команды END(001) удаляются.

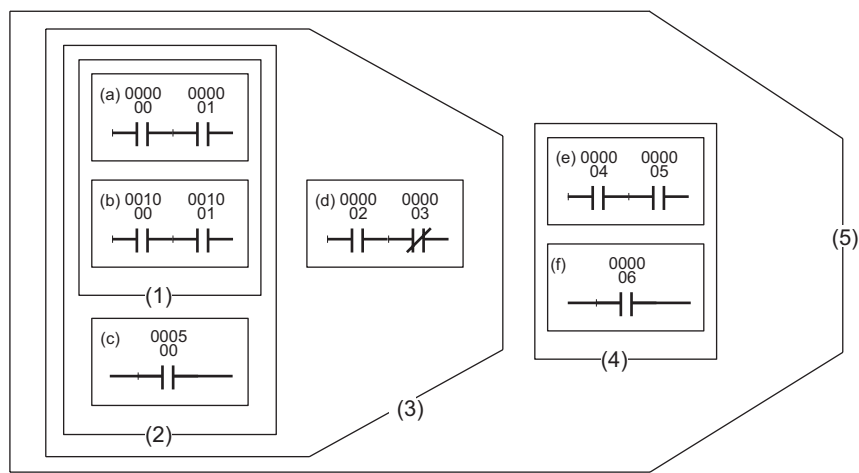
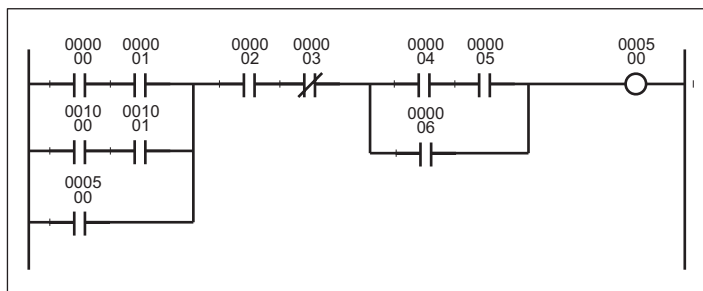


### 2-1-13 Ввод мнемонических кодов

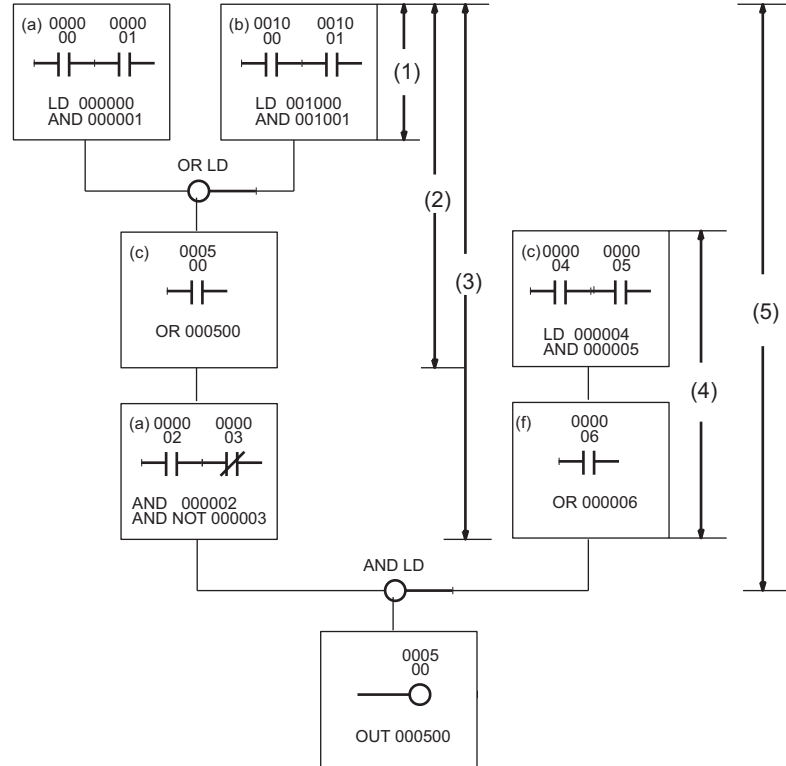
Логическое условие начала выполнения завершается командой LD/LD NOT. Область, заключенная между логическим условием начала выполнения и командой, расположенной непосредственно перед командой LD/LD NOT, считается единым блоком команд.

Для создания отдельной ветви, состоящей из двух блоков команд, используйте команду AND LD (объединение блоков в логическое И) или команду OR LD (объединение блоков в логическое ИЛИ). Процедура ввода мнемонических обозначений показана на представленном ниже примере сложной ветви.

1,2,3... 1. Сначала разбейте ветвь на небольшие блоки (а) ... (f).



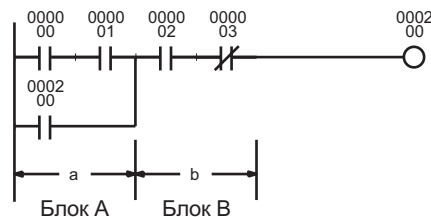
- Составьте программу из блоков, располагая их в направлении сверху вниз, слева направо.



	Адрес	Команда	Операнд
(a)	000200	LD	000000
	000201	AND	000001
(b)	000202	LD	001000
	000203	AND	001001
	000204	OR LD	---
(c)	000205	OR	000500
(d)	000206	AND	000002
	000207	AND NOT	000003
(e)	000208	LD	000004
	000209	AND	000005
(f)	000210	OR	000006
	000211	AND LD	---
	000212	OUT	000500

### 2-1-14 Примеры программ

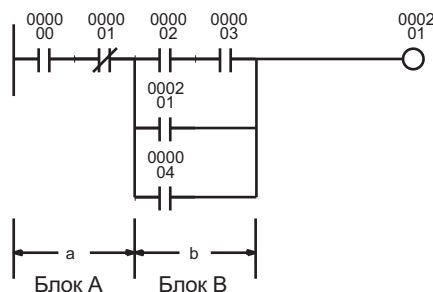
1,2,3... 1. Параллельное/последовательное включение ветвей



Команда	Операнды
LD	000000
AND	000001
OR	000200
AND	000002
AND NOT	000003
OUT	000200

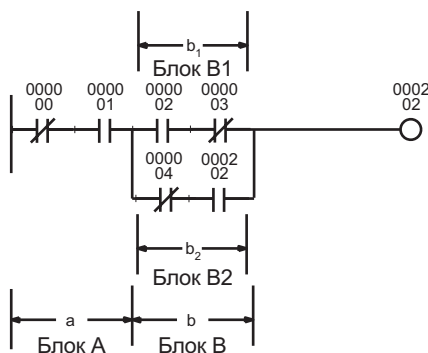
Сначала создайте блок А с параллельно включенными командами, затем создайте блок В.

2. Ветви с последовательным/параллельным включением



Команда	Операнды
LD	000000
AND NOT	000001
LD	000002
AND	000003
OR	000201
OR	000004
AND LD	---
OUT	000201

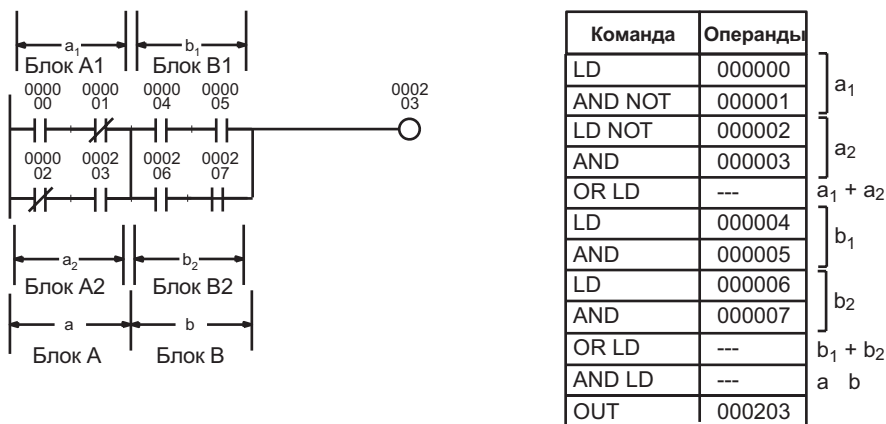
- Разбейте ветвь на блоки А и В и создайте каждый из блоков отдельно.
- Объедините блоки А и В с помощью команды END LD.
- Создайте программу блока А.



Команда	Операнды
LD NOT	000000
AND	000001
LD	000002
AND NOT	000003
LD NOT	000004
AND	000202
OR LD	---
AND LD	---
OUT	000202

- Составьте программу блока В<sub>1</sub>, а затем - программу блока В<sub>2</sub>.
- Объедините блоки В<sub>1</sub> и В<sub>2</sub> с помощью OR LD, а затем объедините блоки А и В с помощью AND LD.

3. Пример последовательного подключения в ветви

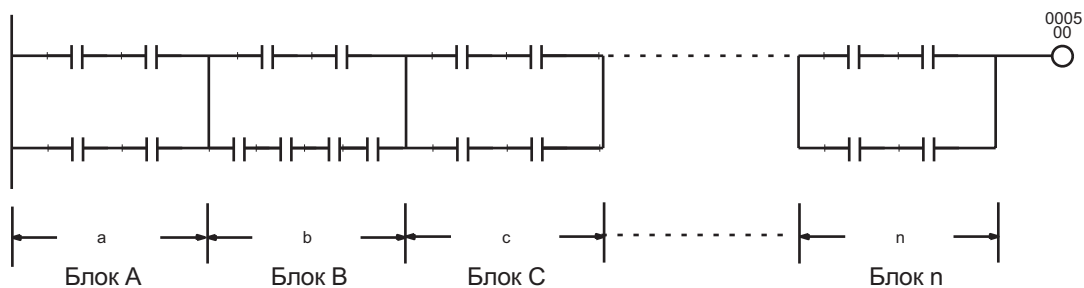


Создайте программу блока A<sub>1</sub>, программу блока A<sub>2</sub>, после чего объедините блоки A<sub>1</sub> и A<sub>2</sub> с помощью OR LD.

Аналогичным образом создайте программы для B<sub>1</sub> и B<sub>2</sub>.

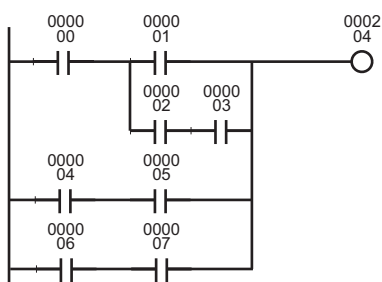
Объедините блок A и блок B с помощью AND LD.

Выполните эти действия столько раз, сколько имеется таких блоков.

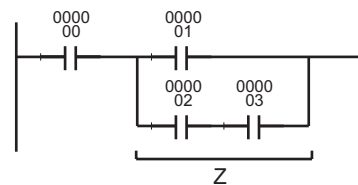




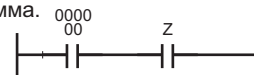
4. Сложные ветви



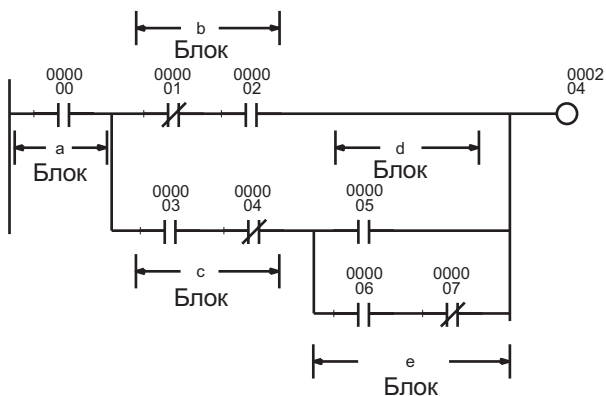
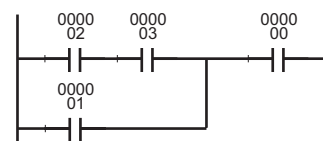
Команда	Операнд
LD	000000
LD	000001
LD	000002
AND	000003
OR LD	---
AND LD	---
LD	000004
AND	000005
OR LD	---
LD	000006
AND	000007
OR LD	---
OUT	000204



В основе верхней диаграммы лежит нижняя диаграмма.



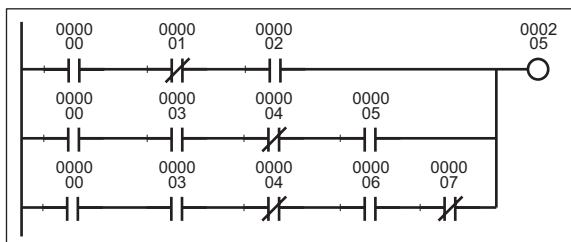
Если переписать программу, как показано ниже, можно получить более простую программу.

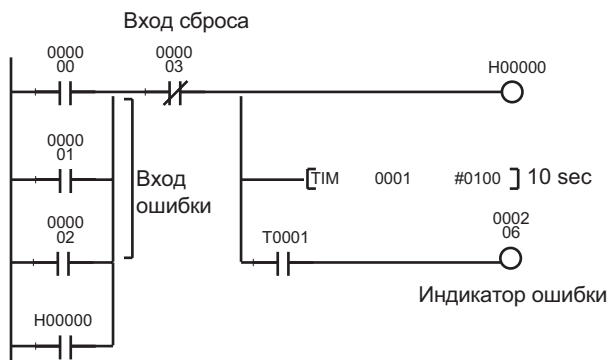


Команда	Операнд
LD	000000
LD NOT	000001
AND	000002
LD	000003
AND NOT	000004
LD	000005
LD	000006
AND NOT	000007
OR LD	---
AND LD	---
OR LD	---
AND LD	---
OUT	000205

a  
b  
c  
d  
e  
d + e  
(d + e) • c  
(d + e) • c + b  
((d + e) • c + b) • a

Эту ветвь можно переписать следующим образом:





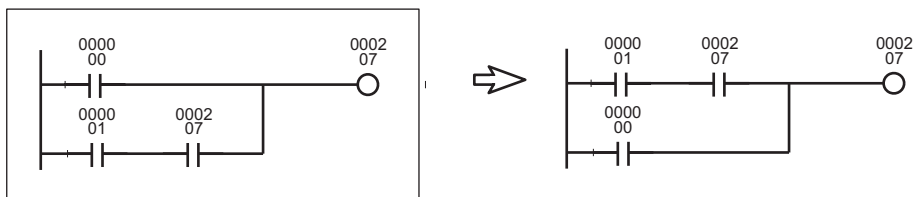
Команда	Операнд
LD	000000
OR	000001
OR	000002
OR	H00000
AND NOT	000003
OUT	H00000
TIM	0001
	0100
AND	T0001
OUT	000206

Если используется бит удержания значений, состояния ВКЛ/ВЫКЛ будет сохраняться в памяти даже после выключения питания, а сигнал ошибки будет действовать даже после повторного включения питания.

4. Ветви, требующие внимательности или перепрограммирования

**Команды OR и OL LD**

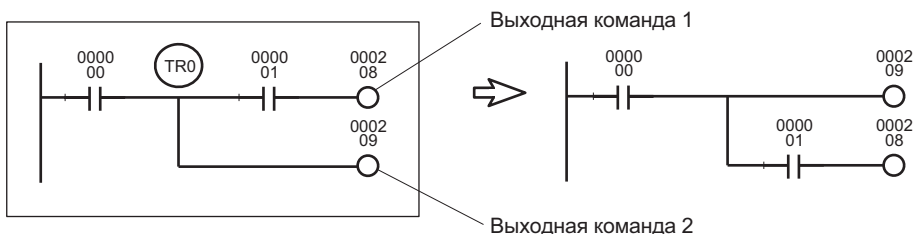
В случае команд OR или OR NOT операция логического ИЛИ берется над результатами логических операторов лестничной диаграммы, расположенных между командой LD или LD NOT и командой OR или OR NOT, поэтому ветви можно реорганизовать таким образом, чтобы необходимость в команде OR LD отпала.



Пример: если ветви составляются так, как показано выше, без каких-либо изменений, требуется применение команды OR LD. Однако, изменив структуру ветвей, как показано выше, можно избавиться от нескольких шагов.

**Ветвление выходных команд**

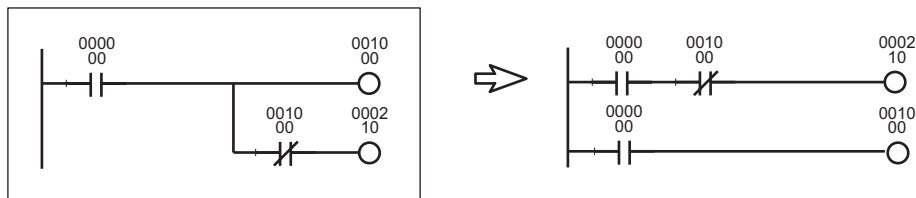
При наличии ветви перед командой AND или AND NOT требуется бит TR. Однако бит TR не потребуется, если ветвь приходит в точку, в которой она непосредственно подключается к выходным командам и к командам AND и AND NOT, или в которой выходные команды могут быть продолжены в своем первоначальном виде.



Пример: в точке ветвления необходимо наличие выходной команды с битом временного хранения TR0 и входной (нагружающей) (LD) команды, если ветви программируются без каких-либо изменений. Перестроив ветви, можно избавиться от нескольких шагов.

**Порядок выполнения мнемонических команд**

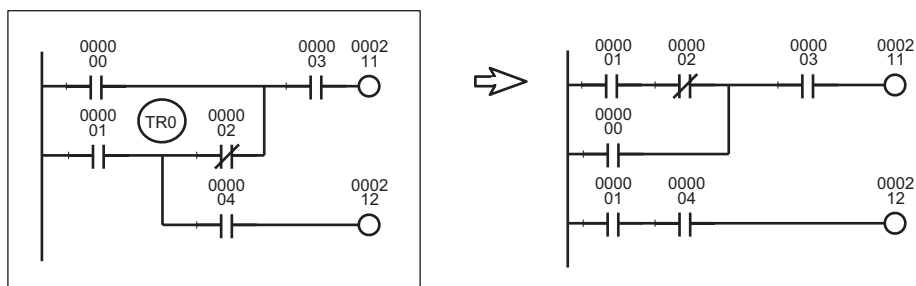
ПЛК выполняет РКЛ-программу ("лестничную диаграмму") в порядке, в котором вводились мнемонические обозначения, поэтому команды могут работать не так, как это предполагалось, в зависимости от способа записи ветвей. При создании лестничных диаграмм всегда следует учитывать порядок выполнения мнемонических обозначений.



Пример: СIO 000210 на представленной выше диаграмме не может быть передан на выход. Изменив структуру ветви, как показано выше, СIO 000210 можно включить (ВКЛ) на один цикл.

**Ветви, для которых требуется изменение структуры**

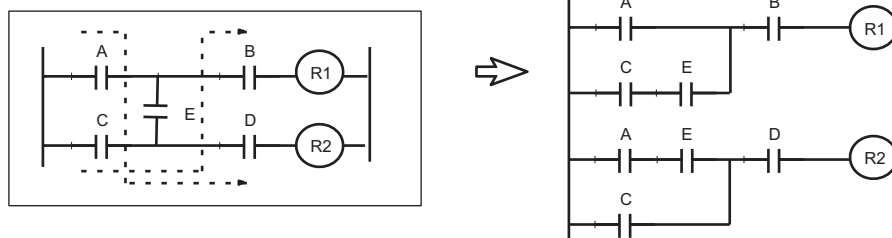
ПЛК выполняет команды в том порядке, в котором были введены мнемонические обозначения, поэтому ток (сигнал) в лестничной диаграмме направлен слева направо. Нельзя создавать программу, в которой ток протекает справа налево.



Пример: можно написать такую программу (см. верхнюю диаграмму слева), в которой в точку TR0 приходит ветвь. Однако такой же результат можно получить и с помощью ветвей, показанных на рисунке справа. Такая структура более проста и понятна. Поэтому рекомендуется изменить структуру ветвей, показанных слева, создав структуру, показанную справа.

Измените структуру ветвей, показанных на нижнем рисунке слева. В первоначальном виде такая программа выполнена быть не может.

Стрелками показано направление движения тока, когда ветви состоят из управляющих реле.



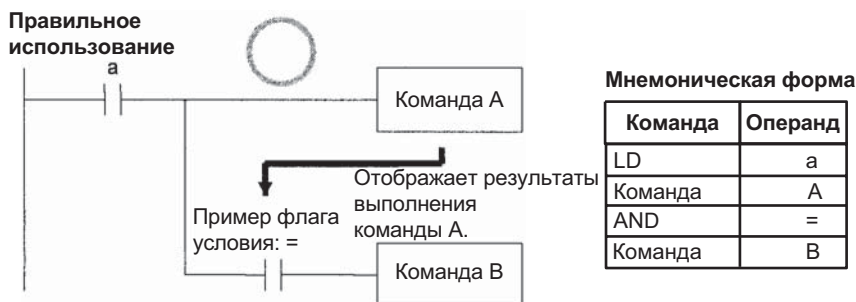
## 2-2 Замечания

### 2-2-1 Флаги условий

#### Использование флагов условий

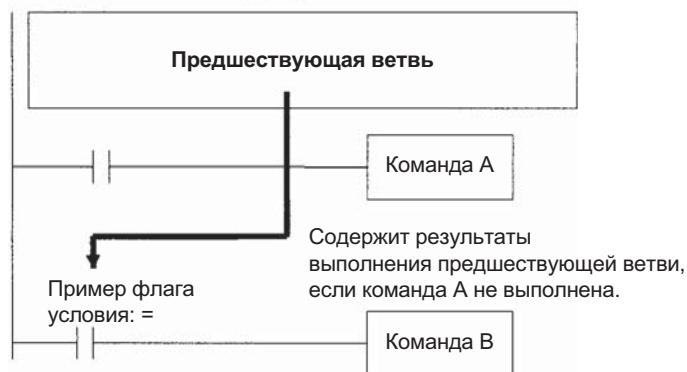
Флаги условий используются совместно со всеми командами и изменяют свое состояние в пределах цикла в зависимости от результатов выполнения отдельных команд. Следовательно, на выходе с разветвлением с одним и тем же условием выполнения флаги условий следует располагать сразу же после команды, чтобы они отображали результаты ее выполнения. Флаг условия нельзя подключать непосредственно к шине, поскольку в этом случае он будет отображать результаты выполнения других команд.

**Пример:** использование результатов выполнения команды А



Команда В выполняется в зависимости от результатов выполнения команды А, при этом для команд А и В используется одно и то же условие выполнения (а). В этом случае команда В будет выполнена в соответствии с состоянием флага условия только, если была выполнена команда А.

#### Неправильное использование

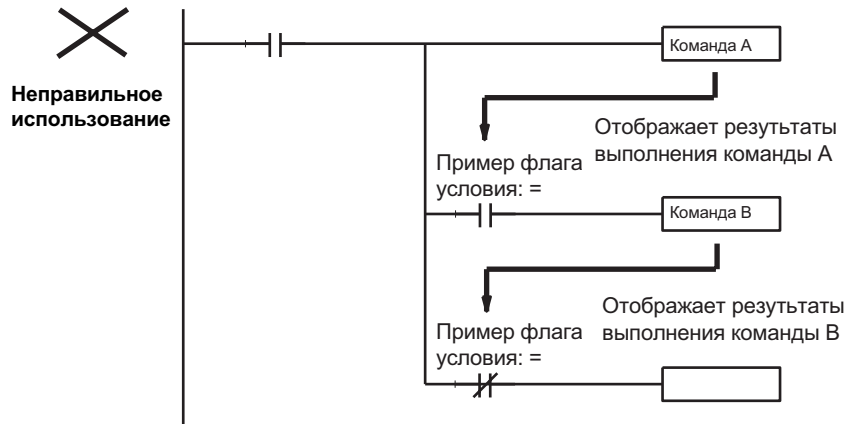


Если флаг условия подсоединен непосредственно к левой шине, команда В будет выполнена в зависимости от результатов выполнения предшествующей ветви, если не выполнена команда А.

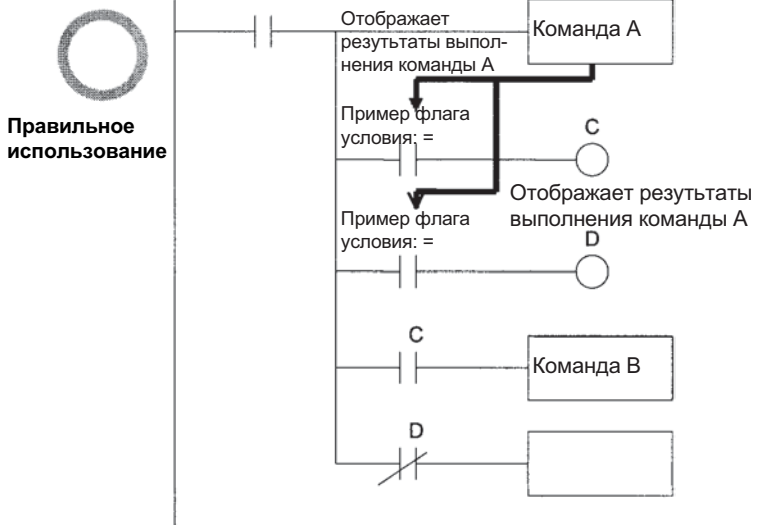
**Примечание** Флаги условий используются всеми командами в пределах одной программы (задачи), но они обнуляются при переключении задач. Следовательно, результаты выполнения, которые наблюдались в предшествующей ветви, не сохранятся в последующих задачах. Поскольку флаги условий используются совместно всеми командами, следует тщательно проверить, чтобы они не мешали работе друг друга в пределах единой "лестничной диаграммы". Пример смотрите ниже.

**Использование результатов выполнения на входах N.C. (нормально-замкнутый контакт) и N.O. (нормально-разомкнутый контакт)**

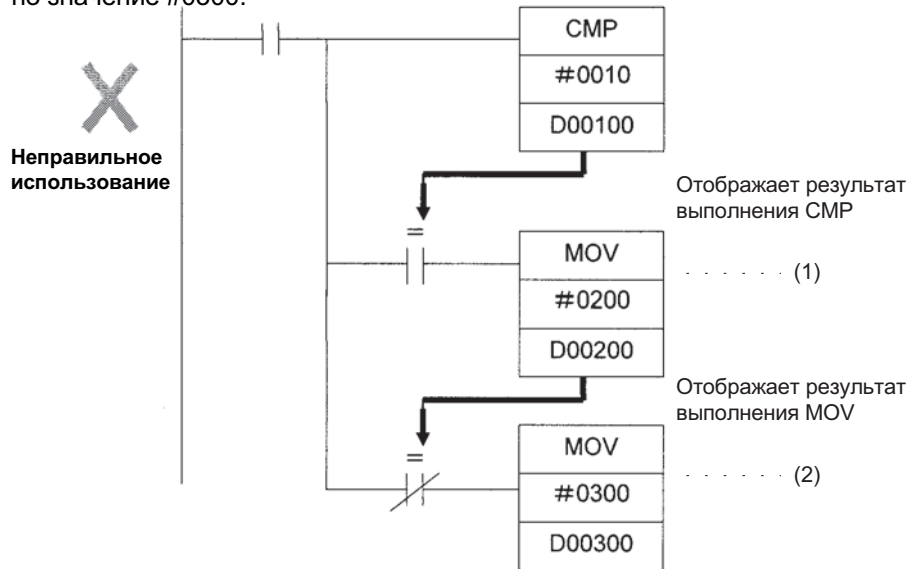
Во флаги условий будут записаны результаты выполнения команды B (см. пример ниже), несмотря на то, что входные биты N.C. и N.O. выполняются в пределах одного и того же выходного ветвления.



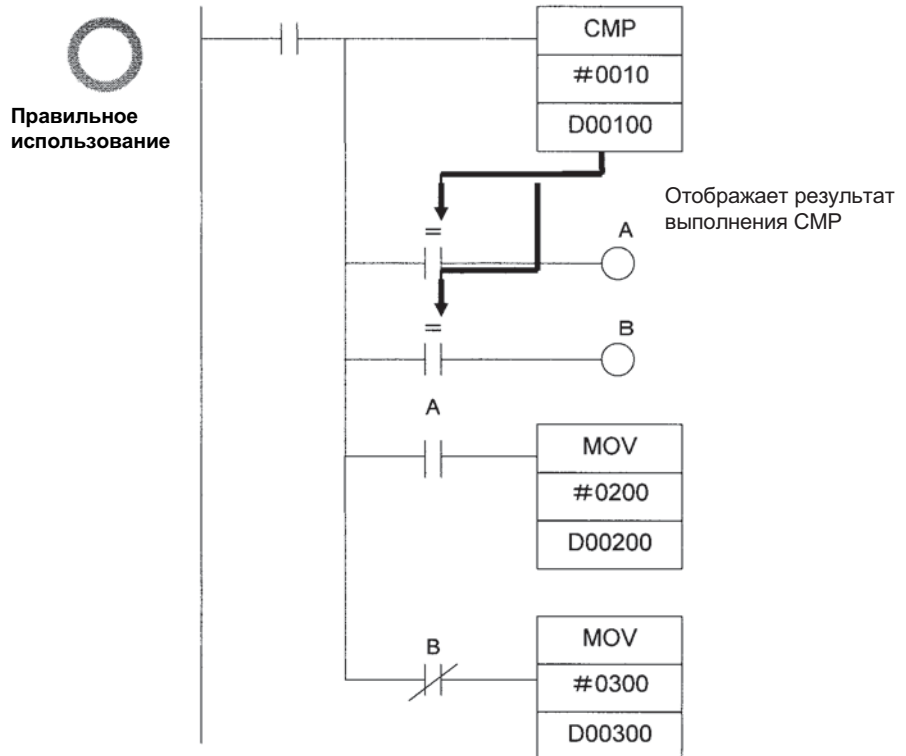
Обеспечьте, чтобы каждый из результатов считывался командой OUTPUT только один раз, чтобы не оказались считанными результаты выполнения для команды B.



**Пример:** в D00200 будет занесено значение #0200, если D00100 содержит #0010, а если D00100 не содержит значение #0010, в D00300 будет занесено значение #0300.



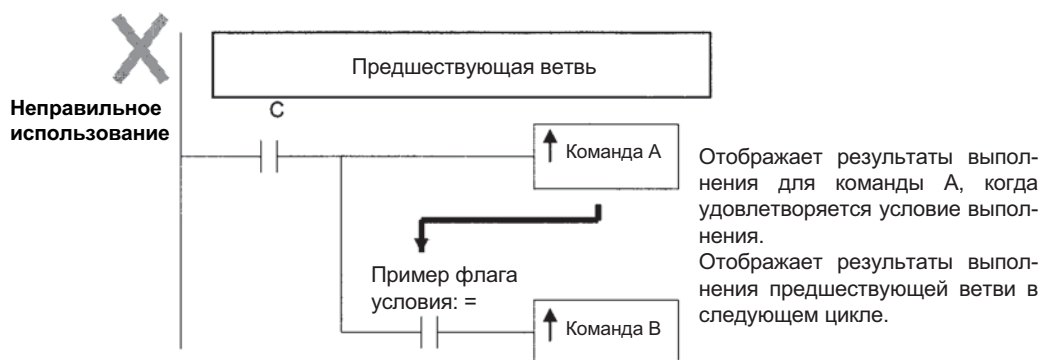
Флаг равенства включается, если D00100 в приведенной выше ветви содержит #0010. Команда (1) запишет в D00200 значение #0200, но после этого флаг равенства сбросится (ВЫКЛ), поскольку источник данных (#0200) отличается от 0000 Hex. После этого будет выполнена команда MOV (2) и в D0300 будет занесено значение #0300. Следовательно, чтобы предотвратить считывание результатов выполнения первой команды MOVE, необходимо вставить дополнительную ветвь, как показано на рисунке ниже.



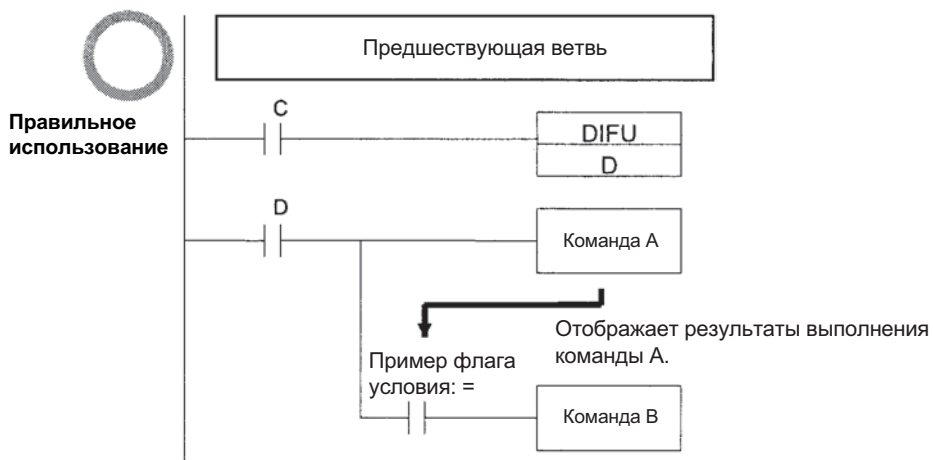
### Использование результатов выполнения однократных команд

В случае однократных (дифференцированных) команд результаты выполнения команд отображаются во флагах условий только, если удовлетворяется условие выполнения, а результаты выполнения предшествующей ветви (в отличие от результатов выполнения однократной команды) отображаются во флагах условий в следующем цикле. Поэтому, если требуется использовать результаты выполнения для однократных команд, необходимо принимать в расчет действие флагов условий в следующем цикле.

Ниже приведен пример, в котором команды А и В будут выполнены только тогда, когда выполнится условие С. В этой программе при считывании командой В результатов выполнения команды А возникает следующая проблема. Если условие выполнения С остается включенным (ВКЛ) в следующем цикле после выполнения команды А, в этом случае может непредсказуемым образом (в зависимости от условия выполнения) выполниться команда В в момент, когда флаг условия переключается из ВЫКЛ во ВКЛ, поскольку в нем отображаются результаты выполнения предшествующей ветви.



Следовательно, в данном случае команды А и В не являются однократными, и вместо них используется команда DIFU (или DIFD) (см. рисунок ниже), а обе команды А и В являются однократными, с выделением переключения ВЫКЛ -> ВКЛ (или ВКЛ -> ВЫКЛ) и выполняются только в одном цикле.



**Примечание** Модулями CPU CS1-H, CJ1-H, CJ1M или CS1D поддерживаются команды, предназначенные для сохранения и загрузки состояний флагов условий (CCS(282) и CCL(283)). Эти команды можно использовать для обращения к состояниям флагов условий, размещенных на других участках программы, в текущей задаче или в другой задаче.

### Основные условия, приводящие к включению флагов условий

#### Флаг "Ошибка"

Флаг ER устанавливается (ВКЛ) при особых обстоятельствах, например, когда для команды указан ошибочный операнд. Когда флаг ER установлен (ВКЛ), команда не выполняется.

Когда флаг ER установлен (ВКЛ), состояние других флагов условий, например, флагов "<", ">", "OF" и "UF" не изменяется, а состояние флагов "=" и "N" изменяется от команды к команде.

Перечень условий, которые приводят к включению флага ER, можно найти в описаниях отдельных команд в руководстве *CS/CJ-series Programmable Controllers Instructions Reference Manual (W340)*. Следует быть внимательным, поскольку некоторые команды сбрасывают (ВЫКЛ) флаг ER независимо от условий.

**Примечание** С помощью параметров области настройки ПЛК можно указать, должна ли прекращаться работа, когда включается флаг ER, то есть, когда происходит ошибка выполнения команды. По умолчанию работа в случае включения флага ER продолжается. Если выбрано прекращение работы после включения флага ER, работа прекращается (воспринимается как ошибка программы), а в слова A298 - A 299 записывается адрес программы, соответствующий точке, в которой прервалась работа. Одновременно включится бит A29508.

#### **Флаг "Равенство"**

Флаг "Равенство" - это временный флаг для всех команд. Он устанавливается автоматически системой и изменяется в процессе работы. Флаг "Равенство" может быть сброшен (установлен) следующей командой после того, как он был сброшен (установлен) предшествующей командой. Флаг "Равенство" установится (ВКЛ), например, когда MOV или другая команда перемещения данных перемещает в качестве исходных данных (источника) значение 0000 Hex, и будет сброшен (ВЫКЛ) во всех других случаях. Даже если команда включает флаг "Равенство", будет сразу же выполнена команда перемещения данных, и флаг "Равенство" либо установится, либо сбросится в зависимости от того, содержат ли исходные данные для команды перемещения данных значение 0000 Hex или нет.

#### **Флаг "Перенос"**

Флаг CY используется в командах сдвига данных, в командах сложения и вычитания со входом переноса, в командах сложения и вычитания со входом заема, а также в командах, предназначенных для специального модуля ввода/вывода, в командах ПИД-регулирования и в командах для FPD. Обратите внимание на следующее.

**Примечание**

1. Результаты выполнения некоторых команд могут привести к тому, что флаг CY останется включенным (выключенным) и будет использоваться другой командой (командой сложения или вычитания с переносом или командой сдвига). Обязательно обнуляйте флаг "Перенос", когда это необходимо.
2. В результате выполнения определенной команды флаг CY может включиться (выключиться) и будет выключен (включен) другой командой. Обязательно следите за тем, чтобы состояние флага "Перенос" было актуально, когда он используется.

#### **Флаги "Больше" и "Меньше"**

Флаги < и > используются в командах, выполняющих операции сравнения, а также в командах LMT, BAND, ZONE, PID и других командах.

Флаг < или > может быть сброшен (установлен) другой командой, даже если он был установлен (сброшен) в результате выполнения предшествующей команды.

#### **Флаг "Отрицательное значение"**

Флаг N сбрасывается, когда старший (крайний левый) бит слова результата выполнения определенных команд содержит "1", и сбрасывается без каких-либо условий для другой команды.

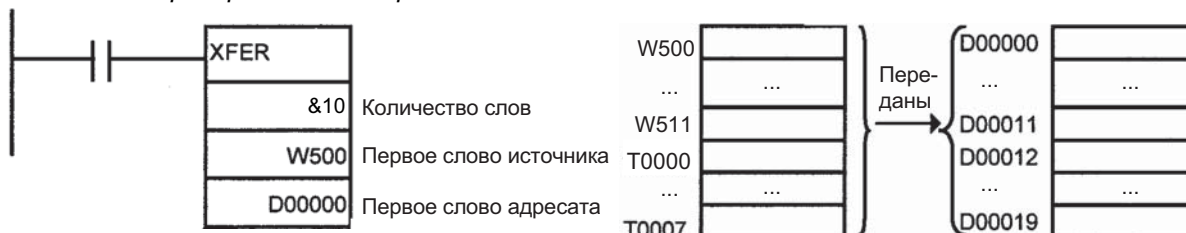
#### **Операнды, занимающие несколько слов**

В ПЛК серии CS/CJ команда будет выполнена в том виде, в котором она записана, даже если ее операнд, состоящий из нескольких слов, указан таким образом, что слова находятся в разных областях. В этом случае слова считываются в порядке следования адресов в памяти ПЛК. Флаг "Ошибка" **не** включится.



В качестве примера рассмотрим результаты выполнения операции перемещения блока с помощью XFER(070). Перемещается 20 слов, начиная с W500. В этом случае будет превышена рабочая область (Work Area), которая завершается словом W511, но команда будет выполнена без включения флага "Ошибка". В адресном пространстве ПЛК после рабочей области размещаются текущие значения таймеров, поэтому в нашем случае в ячейки D00000 ... D00011 будут переданы слова W500 ... W511, а в ячейки D00012 ... D00019 будут размещены текущие значения таймеров T0000 ... T0007.

**Примечание** Распределение адресов в памяти ПЛК можно найти в приложении *Карта памяти ПЛК и распределение адресов*



## 2-2-2 Специальные разделы программ

В программах ПЛК серии CS/CJ имеются особые разделы, предусматривающие управление условиями выполнения команд. Предусмотрены следующие специальные разделы программ.

Раздел программы	Команды	Условие выполнения команды	Состояние
Подпрограмма	Команды SBS, SBN и RET	Выполнение подпрограммы	Выполняется подпрограмма, т.е., раздел программы, размещенный между командами SBN и RET.
Раздел IL - ILC	Команды IL и ILC	Блокировка раздела	Выходные биты обнуляются, таймеры сбрасываются. Остальные команды выполняться не будут, сохраняется предшествующее состояние.
Раздел шагового выполнения лестничной диаграммы	Команды STEP S и команды STEP		
Цикл FOR-NEXT	Команды FOR и команды NEXT	Выдерживание паузы	Зацикливание
Раздел JMP0 - JME0	Команды JMP0 и команды JME0		Переход
Программный блок	Команды BPRG и команды BEND	Выполнение программного блока	Выполняется программный блок, мнемонические коды которого размещены между командами BPRG и BEND.

### Комбинации команд

В следующей таблице показано, какие специальные команды могут использоваться внутри разделов программ.

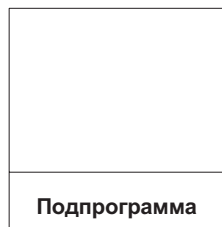
	Подпрограмма	Раздел IL - ILC	Шаговое выполнение лестничной диагр.	Цикл FOR - NEXT	Раздел JMP0 - JME0	Программный блок
Подпрограмма	Не допускается	Не допускается	Не допускается	Не допускается	Не допускается	Не допускается
Раздел IL - ILC	ОК	Не допускается	Не допускается	ОК	ОК	Не допускается
Шаговое выполнение лестничной диагр.	Не допускается	ОК	Не допускается	Не допускается	ОК	Не допускается
Цикл FOR - NEXT	ОК	ОК	Не допускается	ОК	ОК	Не допускается
Раздел JMP0 - JME0	ОК	ОК	Не допускается	Не допускается	Не допускается	Не допускается
Программный блок	ОК	ОК	ОК	Не допускается	ОК	Не допускается

**Примечание** Команды, которые служат для определения областей программ, нельзя использовать для программ в других задачах. Подробности смотрите в 4-2-2 *Ограничения на применение команд в задачах*.

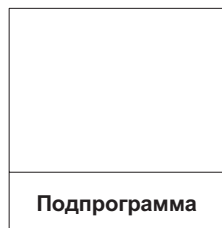
**Подпрограммы**

Размещайте все подпрограммы вместе непосредственно перед командой END(001), после основной программы (подпрограмму нельзя разместить в пошаговой лестничной диаграмме, в программном блоке, в цикле FOR - NEXT или в разделе JMP0 - JME0). Если после текста подпрограммы (после блока SBN - RET) размещена другая программа, эта программа выполняться не будет.

Программа



Программа

**Команды, которые нельзя применять в подпрограммах**

Следующие команды нельзя размещать в тексте подпрограммы

Функция	Мнемонический код	Команда
Пошаговое регулирование процесса	STEP(008)	Определяет начало раздела шаговой лестничной диаграммы
	SNXT(009)	Переход на один шаг лестничной диаграммы

**Примечание Программные блоки**

Подпрограмма может включать в себя программные блоки. Однако если программный блок находится в состоянии WAIT (ожидание), когда происходит возврат из подпрограммы к основной программе, программный блок останется в состоянии WAIT при следующем вызове.

**Команды, которые  
нельзя применять  
в разделах шаговых  
лестничных диаграмм**

Функция	Мнемонический код	Команда
Управление последовательностью	FOR(512), NEXT(513) и BREAK(514)	FOR, NEXT и BREAK LOOP
	END(001)	END
	IL(002) и ILC(003)	INTERLOCK и INTERLOCK CLEAR
	JMP(004) и JME(005)	JUMP и JUMP END
	CJP(510) и CJPN(511)	CONDITIONAL JUMP и CONDITIONAL JUMP NOT
	JMP0(515) и JME0(516)	MULTIPLE JUMP и MULTIPLE JUMP END
Подпрограммы	SBN(092) и RET(093)	SUBROUTINE ENTRY и SUBROUTINE RETURN
Программные блоки	IF(802) (NOT), ELSE(803) и IEND(804)	Команды ветвления
	BPRG(096) и BEND(801)	BLOCK PROGRAM BEGIN/END
	EXIT(806) (NOT)	CONDITIONAL BLOCK EXIT (NOT)
	LOOP(809) и LEND(810) (NOT)	Управление циклами
	WAIT(805) (NOT)	ONE CYCLE WAIT (NOT)
	TIMW(813)	TIMER WAIT
	TMHW(815)	HIGH-SPEED TIMER WAIT
	CNTW(814)	COUNTER WAIT
	BPPS(811) и BPRS(812)	BLOCK PROGRAM PAUSE и RESTART

- Примечание**
1. Раздел пошагового выполнения РКЛ-программы ("лестничной диаграммы") можно использовать внутри блокируемого раздела (между IL и ILC). Когда включается блокировка, пошаговая лестничная диаграмма полностью сбрасывается.
  2. Раздел пошаговой лестничной диаграммы можно размещать между MULTIPLE JUMP (JMP0) и MULTIPLE JUMP END (JME0).

**Команды, которые нельзя использовать внутри программных блоков**

Следующие команды нельзя размещать в программном блоке.

Функция	Мнемонический код	Команда
Управление последовательностью	FOR(512), NEXT(513) и BREAK(514)	FOR, NEXT и BREAK LOOP
	END(001)	END
	IL(002) и ILC(003)	INTERLOCK и INTERLOCK CLEAR
	JMP0(515) и JME0(516)	MULTIPLE JUMP и MULTIPLE JUMP END
Последовательный ввод	UP(521)	CONDITION ON
	DOWN(522)	CONDITION OFF
Последовательный вывод	DIFU	DIFFERENTIATE UP
	DIFD	DIFFERENTIATE DOWN
	KEEP	KEEP
	OUT	OUTPUT
	OUT	NOTOUTPUT NOT
Таймер/счетчик	TIM	TIMER
	TIM	HIGH-SPEED TIMER
	TMH(540)	ONE-MS TIMER
	TTIM(087)	ACCUMULATIVE TIMER
	TIML(542)	LONG TIMER
	MTIM(543)	TIMER
	CNT	COUNTER
	CNTR	REVERSIBLE COUNTER
Подпрограммы	SBN(092) и RET(093)	SUBROUTINE ENTRY и SUBROUTINE RETURN
Сдвиг данных	SFT	SHIFT
Управление шаговой лестничной диаграммой	STEP(008) и SNXT(009)	STEP DEFINE и STEP START
Регулирование	PID	PID CONTROL
Программный блок	BPRG(096)	BLOCK PROGRAM BEGIN
Диагностика неисправностей	FPD(269)	FAILURE POINT DETECTION

- Примечание**
1. Программный блок можно использовать внутри пошаговой лестничной диаграммы.
  2. Программный блок можно использовать внутри блокируемого раздела (между IL и ILC). Когда блокировка активна, программный блок не исполняется.
  3. Программный блок можно размещать между MULTIPLE JUMP (JMP0) и MULTIPLE JUMP END (JME0).
  4. Команду JUMP (JMP) и CONDITIONAL JUMP (CJP/CJPN) можно использовать внутри программного блока. Команды JUMP (JMP) и JUMP END (JME), а также команды CONDITIONAL JUMP (CJP/CJPN) и JUMP END (JME) нельзя включать в программный блок, если они не используются в паре. Если эти команды не использованы в паре, программа будет работать некорректно.

## 2-3 Проверка программ

Проверку программ ПЛК серии CS/CJ можно выполнить на следующих этапах.

- Проверка при вводе программы с консоли программирования
- Проверка программы с помощью CX-Programmer
- Проверка команд в процессе выполнения
- Проверка фатальных ошибок (ошибок программы) в процессе выполнения

### 2-3-1 Ошибки при вводе программы с помощью средства программирования

#### Консоль программирования

Во время ввода с консоли программирования ошибки отображаются в следующих случаях.

Индикатор ошибки	Причина
CHK MEM	Ключ 1 DIP-переключателя модуля CPU в положении ВКЛ (защита от записи).
IO No. ERR	Попытка ввода недопустимой точки ввода/вывода.

#### CX-Programmer

Программа автоматически проверяется пакетом CX-Programmer в следующих случаях.

Момент проверки	Предмет проверки
При вводе "лестничных диаграмм"	Вводимые команды, вводимые операнды, программа
При загрузке файлов	Все операнды во всех командах и все программные блоки
При считывании файлов	Модели, поддерживаемые серией CS/CJ, и все операнды во всех командах
Во время online-редактирования	Объем программы и т.п.

Результаты проверки выводятся в текстовой строке окна вывода (Output Window). Кроме того, левая шина разделов программы, в которых имеются ошибки, отображается в окне представления программы в виде "лестничной диаграммы" красным цветом.

### 2-3-2 Проверка программы с помощью CX-Programmer

В следующей таблице перечислены ошибки, которые могут быть обнаружены при проверке программы в пакете CX-Programmer.

CX-Programmer не проверяет наличие ошибок диапазонов в случае косвенной адресацией к операндам в командах. Ошибки косвенной адресации будут обнаружены во время проверки выполнения программы, для них устанавливается флаг ER (см. следующий раздел). Подробное описание приведено в руководстве *CS/CJ-series Programmable Controllers Instructions Reference Manual (W340)*.

При проверке программы с помощью CX-Programmer оператор может выбрать один из уровней проверки программы: А, В или С (в порядке возрастания "серьезности" ошибки), а также уровень проверки, конфигурируемый самим пользователем.

Область	Проверка
Недопустимые данные: "лестничная диаграмма"	Размещение команд
	Каналы ввода/вывода
	Соединения
	Полнота (завершенность) команд и операций
Поддержка команд программируемым контроллером (ПЛК)	Команды и операнды, поддерживаемые ПЛК
	Варианты выполнения команд (NOT, !, @ и %)
	Целостность объектного кода

Область	Проверка	
Диапазоны значений операндов	Диапазоны значений операндов	
	Типы данных операндов	
Размер (емкость) программы для ПЛК	Проверка типа доступа к словам, защищенным от записи	
	Проверка диапазонов значений операндов, в том числе: <ul style="list-style-type: none"> <li>• Проверка констант (#, &amp;, +, -)</li> <li>• Проверка управляющих кодов</li> <li>• Проверка границ областей для операндов, занимающих несколько слов</li> <li>• Проверка соответствия размеров для операндов, занимающих несколько слов</li> <li>• Обнаружение наложений диапазонов значений операндов</li> <li>• Проверка областей, отведенных для группы слов</li> <li>• Проверка операндов двойной длины</li> <li>• Проверка границ областей для смещений</li> </ul>	
Синтаксис	Количество шагов	
	Общий объем программы (емкость)	
	Количество задач	
Синтаксис	Проверка вызовов для парных команд <ul style="list-style-type: none"> <li>• IL-ILC</li> <li>• JMP-JME, CJP/CJPN-JME</li> <li>• SBS-SBN-RET, MCRO-SBN-RET</li> <li>• STEP-SNXT</li> <li>• BPRG-BEND</li> <li>• IF-IEND</li> <li>• LOOP-LEND</li> </ul>	
	Области, в которых запрещается размещать BPRG-BEND	
	Области, в которых запрещается размещать SBN-RET	
	Области, в которых запрещается размещать STEP-SNXT	
	Области, в которых запрещается размещать FOR-NEXT	
	Области, в которых запрещается размещать задачи обработки прерываний	
	Области, в которых запрещается размещать BPRG-BEND	
	Области, в которых запрещается размещать FOR-NEXT	
	Недопустимые многократные вложения	
	Команда END(001)	
	Согласованность нумерации	
	Структура "лестничной диаграммы"	Переполнение стека
	Дублирование выходов	Проверка дублированных выходов <ul style="list-style-type: none"> <li>• По битам</li> <li>• По словам</li> <li>• Команды для таймеров/счетчиков</li> <li>• Длинные операнды (по 2 слова и по 4 слова)</li> <li>• Резервирование нескольких слов</li> <li>• Диапазоны начала/завершения</li> <li>• Номера FAL</li> <li>• Команды с несколькими выходными операндами</li> </ul>
Задачи		Проверка задач, для которых выбран запуск в начале работы
	Назначение программ задачам	

**Примечание** Дублирование выходов проверяется только в пределах каждой отдельной задачи.

**Операнды, занимающие несколько слов**

При проверке программы проверяются границы областей памяти для операндов, занимающих несколько слов (см. таблицу ниже).

CX-Programmer	Консоли программирования
Если операнды, занимающие несколько слов, выходят за границу области памяти, CX-Programmer работает следующим образом: <ul style="list-style-type: none"> <li>• Программа не может быть записана в модуль CPU.</li> <li>• Программа не может быть прочитана из модуля CPU.</li> <li>• При проверке программы сигнализируются ошибки компиляции.</li> <li>• При offline-программировании на экране отображаются предупреждения</li> <li>• При online-редактировании в режимах PROGRAM или MONITOR на экране отображаются предупреждения.</li> </ul>	Проверка выполняется при вводе программы, то есть, операнды, которые выходят за границу области памяти, не могут быть введены.

### 2-3-3 Проверка выполнения программы

Размещение операндов и команд проверяется во время ввода программ с помощью средств программирования (включая консоли программирования), а также во время проверки программ с помощью средств программирования (за исключением консолей программирования). Но такая проверка не является окончательной.

В процессе выполнения команд проверяется наличие следующих ошибок.

Тип ошибки	Флаг, который включается при ошибке	Прекращение/продолжение работы
1. Ошибка выполнения команды	Флаг ER Флаг "Ошибка выполнения команды" (A29508) также установится (ВКЛ), если выбрано прекращение работы в случае возникновения ошибки.	В области настроек ПЛК предусмотрен параметр, с помощью которого можно указать, должна ли работа прекращаться или продолжаться в случае возникновения ошибок при выполнении команд. По умолчанию работа продолжается. Если обнаружена ошибка программы, работа прекращается только в том случае, если выбрано прекращение работы.
2. Ошибка доступа	Флаг AER Флаг "Ошибка доступа" (A29510) также установится (ВКЛ), если выбрано прекращение работы в случае возникновения ошибки.	В области настроек ПЛК предусмотрен параметр, с помощью которого можно указать, должна ли работа прекращаться или продолжаться в случае возникновения ошибок при выполнении команд. По умолчанию работа продолжается. Если обнаружена ошибка программы, работа прекращается только в том случае, если выбрано прекращение работы.
3. Недопустимая команда	Флаг ошибки "Недопустимая команда" (A29514)	Фатальная ошибка (ошибка программы)
4. Переполнение UM (память пользователя)	Флаг ошибки "Переполнение UM" (A29515)	Фатальная ошибка (ошибка программы)

#### Ошибки выполнения команд

Если для выполнения команды были предоставлены некорректные данные или была осуществлена попытка выполнения команды за пределами задачи, в этом случае возникает ошибка выполнения команды. Необходимые данные проверяются в начале выполнения команды. Если в этих данных имеется ошибка, команда не выполняется, устанавливается флаг ER ("Ошибка"), а флаги EQ и N могут остаться включенными или сбросятся в зависимости от типа команды.

Флаг ER ("Ошибка") будет сброшен, если команда (за исключением входных команд) завершается без ошибки. Условия, при которых устанавливается флаг ошибки, могут меняться в зависимости от конкретной команды. Подробное описание отдельных команд можно найти в руководстве *CS/CJ-series Programmable Controllers Programming Manual (W340)*.

Если в настройках ПЛК выбрано прекращение работы в случае возникновения ошибок выполнения команд, в этом случае работа прекращается (фатальная ошибка), устанавливается флаг "Ошибка выполнения команды" (A29508) и устанавливается флаг ER.

**Ошибки доступа**

Ошибки, связанные с некорректным обращением к данным (ошибки доступа), информируют о том, что при обращении к адресу, который был указан для операнда команды, произошло обращение к недопустимой области, а именно:

- a) Произведено чтение или запись в область параметров.
- b) Произведена запись в отсутствующую область памяти (см. примечание).
- c) Произведена запись в область EM, выбранную в качестве памяти файлов EM.
- d) Произведена запись в область, предназначенную только для чтения.
- e) Значение, указанное косвенным адресом DM/EM в режиме BCD, оказалось не в формате BCD (например, \*D000001 содержит #A000).

Когда происходит ошибка доступа, включается флаг "Ошибка доступа" (флаг AER), но выполнение команды продолжается и флаг "Ошибка" (ER) не устанавливается.

**Примечание** Ошибки доступа происходят в следующих случаях:

- Когда адрес EM, указанный для текущего банка, превышает 32767 (пример: E32768).
- Для конечного банка (пример: C) выбран косвенный адрес EM в режиме BIN, а указанное слово содержит значение 8000 ... FFFF Hex (пример: @EC\_00001 содержит #8000).
- Для текущего банка (пример: C) выбран косвенный адрес EM в режиме BIN, а указанное слово содержит значение 8000 ... FFFF Hex (пример: @EC\_00001 содержит #8000).
- Регистр IR, в котором содержится адрес бита внутренней памяти, используется для обращения к слову, либо IR, в котором содержится адрес слова внутренней памяти, используется для обращения к биту.

Если в настройках ПЛК выбрано прекращение работы в случае возникновения ошибок команд, в этом случае работа прекращается (фатальная ошибка), устанавливается флаг "Ошибка доступа" (A29510) и устанавливается флаг AER.

**Примечание** Флаг "Ошибка доступа" (флаг AER) не обнуляется после завершения задачи. Если в настройках ПЛК выбрано продолжение работы после возникновения ошибок выполнения команд, состояние этого флага сохраняется до тех пор, пока не встретится команда END(001). Это позволяет проверить, произошла ли при выполнении задачи ошибка доступа (если флаг AER контролируется с помощью консоли программирования, в этом случае контролируется состояние конечного флага AER после выполнения полностью всей программы пользователя).

**Прочие  
ошибки****Ошибка "Недопустимая команда"**

Ошибка "Недопустимая команда" возникает тогда, когда предпринимается попытка выполнения команды, не предназначенной для данной системы. Как правило, в случае создания программ с помощью средств программирования серии CS/CJ (включая консоли программирования), эта ошибка не происходит.

Если все-таки такая ошибка произошла, она воспринимается как ошибка программы, работа прекращается (фатальная ошибка) и устанавливается флаг "Недопустимая команда" (A29514).

**Ошибка "Переполнение памяти пользователя (UM)"**

Ошибки переполнения UM возникают в тех случаях, когда предпринимается попытка выполнения команды, данные которой выходят за границу последнего адреса памяти пользователя (UM), выбранной в качестве области хранения программы. Как правило, при создании программ с помощью средств программирования серии CS/CJ (включая консоли программирования), эта ошибка не происходит.



Если все-таки эта ошибка произошла, она воспринимается как ошибка программы, работа прекращается (фатальная ошибка) и устанавливается (ВКЛ) флаг "Переполнение UM" (A29515).

### 2-3-4 Проверка на наличие фатальных ошибок

Ниже перечислены фатальные ошибки программы, при возникновении которых работа модуля CPU прекращается. Когда работа прекращается из-за возникновения ошибки программы, по адресу A294 будет записан номер задачи, на которой прекратилась работа, а в A298/A299 будет записан адрес программы. Причину ошибки программы можно определить, руководствуясь следующей таблицей.

Адрес	Описание	Сохраненная информация
A294	<p>Если работа прекращается из-за ошибки программы, по этому адресу записывается тип задачи и номер задачи, на которой прекратилась работа.</p> <p>Если в данном цикле активные циклические задачи отсутствуют, сюда будет записано значение FFFF Hex.</p>	<p>Циклическая задача: 0000 ... 001F Hex (циклические задачи 0 ... 31)</p> <p>Задача обработки прерывания: 8000 ... 80FF Hex (задачи обработки прерывания 0 ... 255)</p>
A298/A299	<p>Если работа прекратилась из-за ошибки программы, по этому адресу в двоичном коде будет записан адрес программы, соответствующий точке, в которой прекратилась работа.</p> <p>Если отсутствует команда END(001) (A29511 = ВКЛ), будет записано значение адреса, по которому предполагалось наличие END(001).</p> <p>В случае ошибки выполнения задачи (A29512 = ВКЛ), в A298/A299 будет записано значение FFFFFFFF Hex.</p>	<p>A298: младшая часть адреса программы</p> <p>A299: старшая часть адреса программы</p>

**Примечание** Если устанавливается флаг "Ошибка" или флаг "Ошибка доступа", он воспринимается как ошибка программы и его можно использовать для прекращения работы CPU. Выбор режима работы CPU в случае ошибок программы осуществляется в настройках ПЛК.

Ошибка программы	Описание	Соответствующие флаги
Отсутствует команда END	В программе отсутствует команда END.	Устанавливается флаг "No END" (A29511).
Ошибка во время выполнения задачи	Ни одной готовой задачи в цикле. Задаче не назначена ни одна программа. Условие выполнения для задачи, обрабатываемой по прерыванию, соблюдено, но задача обработки прерывания с соответствующим номером отсутствует.	Включается флаг "Ошибка задачи" (29512).
Ошибка выполнения команды (флаг ER = ВКЛ) и в настройках ПЛК выбрано прекращение работы в случае ошибок выполнения команд	При попытке выполнения команды обнаружен операнд, содержащий некорректные данные.	Если в настройках ПЛК выбрано прекращение работы в случае возникновения ошибок выполнения команд, устанавливается (ВКЛ) флаг ER и флаг "Ошибка выполнения команды" (A29508).
Ошибка доступа (флаг AER = ВКЛ) и в настройках ПЛК выбрано прекращение работы в случае ошибок выполнения команд	Попытка чтения или записи в область параметров. Попытка записи в отсутствующую область памяти (см. примечание). Попытка записи в область EM, указанную в качестве памяти файлов EM. Попытка записи в область, предназначенную только для чтения. Значение, указанное косвенным адресом DM/EM в режиме BCD, оказалось не в формате BCD.	Если в настройках ПЛК выбрано прекращение работы в случае возникновения ошибок выполнения команд, устанавливаются (ВКЛ) флаг AER и флаг ошибки "Ошибка доступа" (A29510).
Ошибка BCD при косвенной адресации через DM/EM и в настройках ПЛК выбрано прекращение работы в случае ошибок выполнения команд	Значение, указанное косвенным адресом DM/EM в режиме BCD, оказалось не в формате BCD.	Если в настройках ПЛК выбрано прекращение работы в случае возникновения ошибок выполнения команд, устанавливаются (ВКЛ) флаг AER и флаг "Ошибка косвенной DM/EM-адресации в режиме BCD" (A29509).
Ошибка переполнения адреса для однократных (дифференцированных) команд	Во время online-редактирования было вставлено или удалено больше 131071 однократных (дифференцированных) команд.	Устанавливается (ВКЛ) флаг "Ошибка переполнения для однократных команд" (A29513).
Ошибка переполнения памяти пользователя (UM)	Предпринята попытка выполнения команды, данные которой выходят за границу последнего адреса памяти пользователя (UM), выделенной под хранение программы.	Устанавливается (ВКЛ) флаг "Переполнение памяти пользователя (UM)" (A29516).
Ошибка "Недопустимая команда"	Попытка выполнения невыполнимой команды.	Устанавливается (ВКЛ) флаг "Недопустимая команда" (A29514).



## РАЗДЕЛ 3

# Команды и их функции

В данном разделе описаны команды, которые можно использовать для создания программ пользователя.

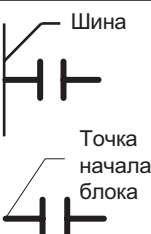





3-1 Команды последовательного ввода . . . . .	74
3-2 Команды последовательного вывода . . . . .	76
3-3 Команды последовательного управления . . . . .	79
3-4 Команды управления таймерами и счетчиками . . . . .	83
3-5 Команды сравнения . . . . .	87
3-6 Команды перемещения данных . . . . .	91
3-7 Команды сдвига данных . . . . .	94
3-8 Команды увеличения/уменьшения . . . . .	98
3-9 Символьные математические команды . . . . .	99
3-10 Команды преобразования . . . . .	104
3-11 Логические команды . . . . .	110
3-12 Специальные математические команды . . . . .	112
3-13 Команды математических операций с плавающей запятой . . . . .	113
3-14 Команды математических операций с плавающей запятой, двойной точности	117
3-15 Команды обработки табличных данных . . . . .	121
3-16 Команды управления данными . . . . .	125
3-17 Команды управления подпрограммами . . . . .	129
3-18 Команды управления прерываниями . . . . .	130
3-19 Команды управления высокоскоростными счетчиками и импульсными выходами (только SJ1M-CPU21/22/23) . . . . .	132
3-20 Шаговые команды . . . . .	134
3-21 Команды для базовых модулей ввода/вывода . . . . .	134
3-22 Команды последовательного интерфейса . . . . .	137
3-23 Сетевые команды . . . . .	138
3-24 Команды памяти файлов . . . . .	140
3-25 Команды дисплея . . . . .	141
3-26 Команды управления часами . . . . .	141
3-27 Команды отладки программы . . . . .	142
3-28 Команды для диагностики неисправностей . . . . .	143
3-29 Прочие команды . . . . .	144
3-30 Команды для программирования блоков . . . . .	145
3-31 Команды для обработки текстовых строк . . . . .	151
3-32 Команды управления задачами . . . . .	154


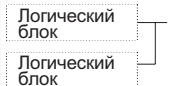
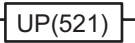
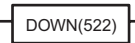
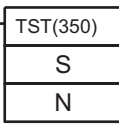
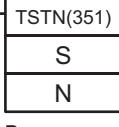
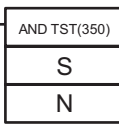
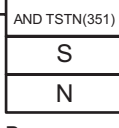
## 3-1 Команды последовательного ввода

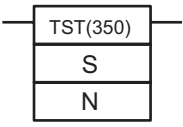
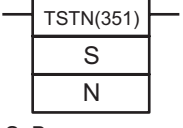
\*1: Не поддерживается модулями CPU CS1D для систем с дублированием CPU.

\*2: Поддерживается только модулями CPU CS1-H, CJ1-H, CJ1M и CS1D.

\*3: Поддерживается только модулями CPU CS1-H, CJ1-H и CJ1M.


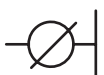
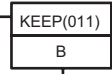

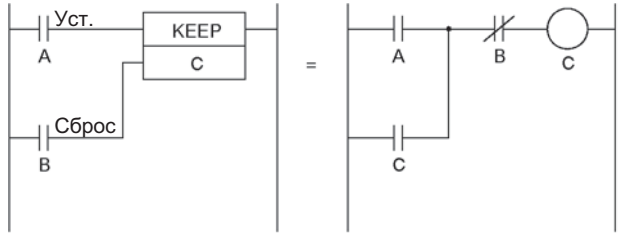
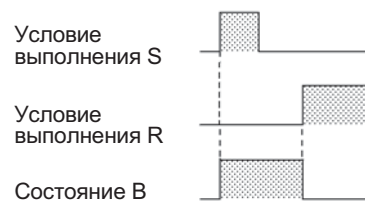
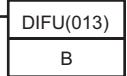
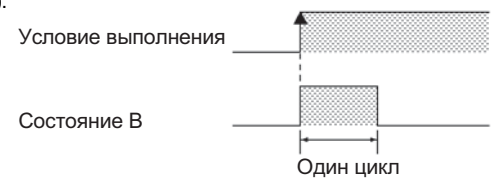
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>LOAD</b> LD @LD %LD !LD* <sup>1</sup> !@LD* <sup>1</sup> !%LD* <sup>1</sup>		Указывает начало логической ветви (ветви лестничной диаграммы) и формирует условие выполнения (ВКЛ/ВЫКЛ), которое определяется состоянием указанного бита-операнда (ВКЛ/ВЫКЛ).	Начало логической ветви Не требуется
<b>LOAD NOT</b> LD NOT @LD NOT* <sup>2</sup> %LD NOT* <sup>2</sup> !LD NOT* <sup>1</sup> !@LD NOT* <sup>3</sup> !%LD NOT* <sup>3</sup>		Указывает начало логической ветви и формирует условие выполнения (ВКЛ/ВЫКЛ), которое определяется инверсией состояния указанного бита-операнда (ВКЛ/ВЫКЛ).	Начало логической ветви Не требуется
<b>AND</b> AND @AND %AND !AND* <sup>1</sup> !@AND* <sup>1</sup> !%AND* <sup>1</sup>		Операция "логическое И" над состоянием указанного бита-операнда и текущим условием выполнения.	Внутри ветви Требуется
<b>AND NOT</b> AND NOT @AND NOT* <sup>2</sup> %AND NOT* <sup>2</sup> !AND NOT* <sup>1</sup> !@AND NOT* <sup>3</sup> !%AND NOT* <sup>3</sup>		Инверсия состояния указанного бита-операнда и операция "логическое И" над результатом инверсии и текущим условием выполнения.	Внутри ветви Требуется
<b>OR</b> OR @OR %OR !OR* <sup>1</sup> !@OR* <sup>1</sup> !%OR* <sup>1</sup>		Операция "логическое ИЛИ" над состоянием (ВКЛ/ВЫКЛ) указанного бита-операнда и текущим условием выполнения.	Внутри ветви Требуется
<b>OR NOT</b> OR NOT @OR NOT* <sup>2</sup> %OR NOT* <sup>2</sup> !OR NOT* <sup>1</sup> !@OR NOT* <sup>3</sup> !%OR NOT* <sup>3</sup>		Инверсия состояния указанного бита и операция "логическое ИЛИ" над результатом инверсии и текущим условием выполнения.	Внутри ветви Требуется

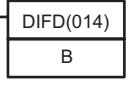

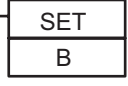
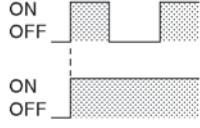


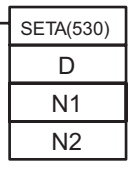
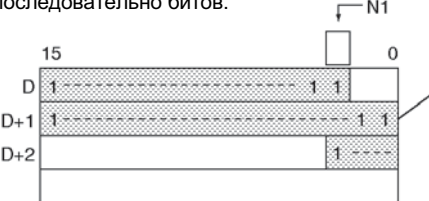
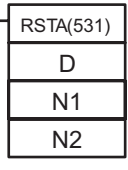
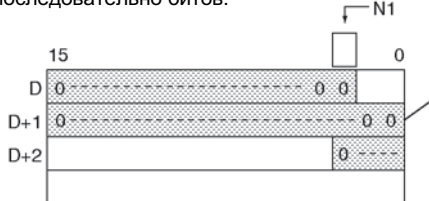
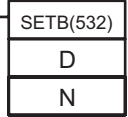
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>AND LOAD</b> AND LD		Операция "логическое И" над двумя логическими блоками.  LD } ... } Логический блок А  LD } ... } Логический блок В  AND LD ..... Последовательное включение логического блока А и логического блока В.	Внутри ветви Требуется
<b>OR LOAD</b> OR LD		Операция "логическое ИЛИ" над двумя логическими блоками.  LD } ... } Логический блок А  LD } ... } Логический блок В  OR LD ..... Параллельное соединение логического блока А и логического блока В.	Внутри ветви Требуется
<b>NOT</b> NOT 520	---	Инверсия условия выполнения.	Внутри ветви Требуется
<b>CONDITION ON</b> UP 521		Команда UP(521) включает условие выполнения на один цикл, когда условие выполнения переходит из ВЫКЛ во ВКЛ.	Внутри ветви Требуется
<b>CONDITION OFF</b> DOWN 522		Команда DOWN(522) включает условие выполнения на один цикл, когда условие выполнения переходит из ВКЛ в ВЫКЛ.	Внутри ветви Требуется
<b>BIT TEST</b> LD TST 350	 S: Входное слово N: Номер бита	Команды LD TST(350), AND TST(350) и OR TST(350) используются в программе так же, как и LD, AND и OR; условие выполнения включено, когда указанный бит в указанном слове включен, и выключено, когда этот бит выключен.	Внутри ветви Не требуется
<b>BIT TEST</b> LD TSTN 351	 S: Входное слово N: Номер бита	Команды LD TSTN(351), AND TSTN(351) и OR TSTN(351) используются в программе так же, как и LD NOT, AND NOT и OR NOT; условие выполнения выключено, когда указанный бит в указанном слове включен, и включено, когда этот бит выключен.	Внутри ветви Не требуется
<b>BIT TEST</b> AND TST 350	 S: Входное слово N: Номер бита	Команды LD TST(350), AND TST(350) и OR TST(350) используются в программе так же, как и LD, AND и OR; условие выполнения включено, когда указанный бит в указанном слове включен, и выключено, когда этот бит выключен.	Внутри ветви Требуется
<b>BIT TEST</b> AND TSTN 351	 S: Входное слово N: Номер бита	Команды LD TSTN(351), AND TSTN(351) и OR TSTN(351) используются в программе так же, как и LD NOT, AND NOT и OR NOT; условие выполнения выключено, когда указанный бит в указанном слове включен, и включено, когда этот бит выключен.	Внутри ветви Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>BIT TEST</b> OR TST 350	 S: Входное слово N: Номер бита	Команды LD TST(350), AND TST(350) и OR TST(350) используются в программе так же, как и LD, AND и OR; условие выполнения включено, когда указанный бит в указанном слове включен, и выключено, когда этот бит выключен.	Внутри ветви Требуется
<b>BIT TEST</b> OR TSTN 351	 S: Входное слово N: Номер бита	Команды LD TSTN(351), AND TSTN(351) и OR TSTN(351) используются в программе так же, как и LD NOT, AND NOT и OR NOT; условие выполнения выключено, когда указанный бит в указанном слове включен, и включено, когда этот бит выключен.	Внутри ветви Требуется

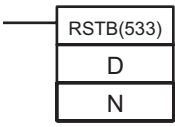
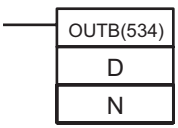
### 3-2 Команды последовательного вывода

\*1: Не поддерживается модулями CPU CS1D для систем с дублированием CPU.

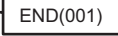
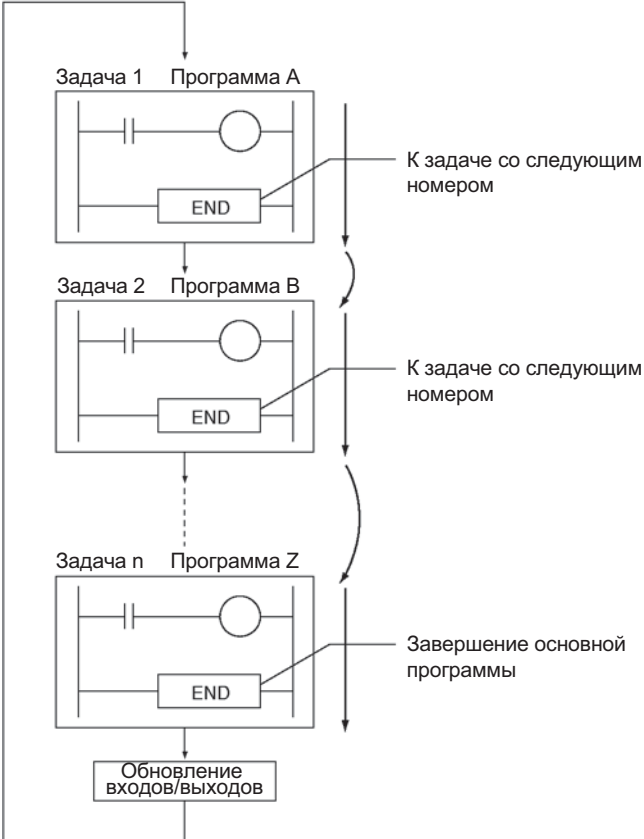

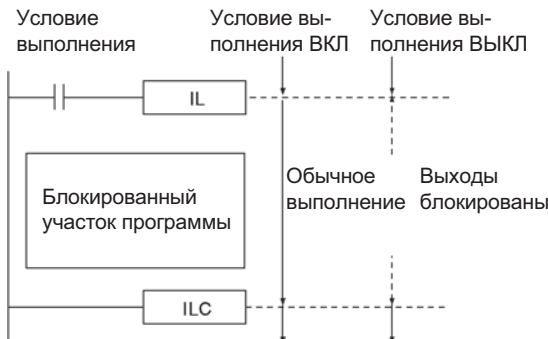
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>OUTPUT</b> OUT !OUT*1		Запись результата (условия выполнения) логической операции в указанный бит.	Выход Требуется
<b>OUTPUT NOT</b> OUT NOT !OUT NOT*1		Инверсия результата (условия выполнения) логической операции и запись результата инверсии в указанный бит.	Выход Требуется
<b>KEEP</b> KEEP !KEEP*1 011	S (Уст.) —  R (Сброс) —  B: Бит	Реле с защелкой (самоблокирующееся реле).  	Выход Требуется
<b>DIFFERENTIATE UP</b> DIFU !DIFU*1 013	 B: Бит	Команда DIFU(013) устанавливает (ВКЛ) указанный бит на один цикл, когда условие выполнения переходит из ВЫКЛ во ВКЛ (передний фронт). 	Выход Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>DIFFERENTIATE DOWN</b>  DIFD !DIFD <sup>*1</sup>  014	 B: Бит	Команда DIFD(014) включает указанный бит на один цикл, когда условие выполнения переходит из ВКЛ в ВЫКЛ (задний фронт).  Условие выполнения   Состояние B  Один цикл	Выход Требуется
<b>SET</b>  SET @SET %SET !SET <sup>*1</sup> !@SET <sup>*1</sup> !%SET <sup>*1</sup>	 B: Бит	Команда SET включает бит-операнд, когда включается условие выполнения.  Условие выполнения для SET   Состояние B	Выход Требуется
<b>RESET</b>  RSET @RSET %RSET !RSET <sup>*1</sup> !@RSET <sup>*1</sup> !%RSET <sup>*1</sup>	 B: Бит	Команда RSET сбрасывает бит-операнд, когда включается условие выполнения.  Условие выполнения для RSET   Состояние B	Выход Требуется
<b>MULTIPLE BIT SET</b>  SETA @SETA 530	 D: Первое слово N1: Первый бит N2: Количество битов	Команда SETA(530) включает указанное количество расположенных последовательно битов.   Биты N2 принимают значение 1 (ВКЛ).	Выход Требуется
<b>MULTIPLE BIT RESET</b>  RSTA @RSTA 531	 D: Первое слово N1: Первый бит N2: Количество битов	Команда RSTA(531) сбрасывает указанное количество расположенных последовательно битов.   В биты N2 записывается 0 (ВЫКЛ).	Выход Требуется
<b>SINGLE BIT SET (только для CS1-H, CJ1-H, CJ1M или CS1D)</b>  SETB @SETB !SETB <sup>*1</sup> !@SETB <sup>*1</sup>	 D: Адрес слова N: Номер бита	Команда SETB(532) включает указанный бит в указанном слове, когда включается условие выполнения.  В отличие от команды SET, команду SETB(532) можно использовать для включения бита в слове DM или EM.	Выход Требуется

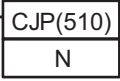
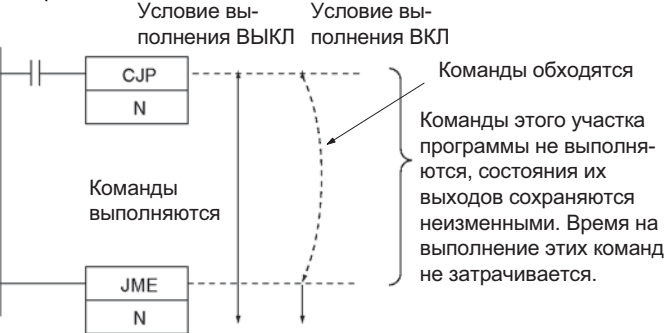
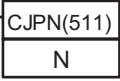
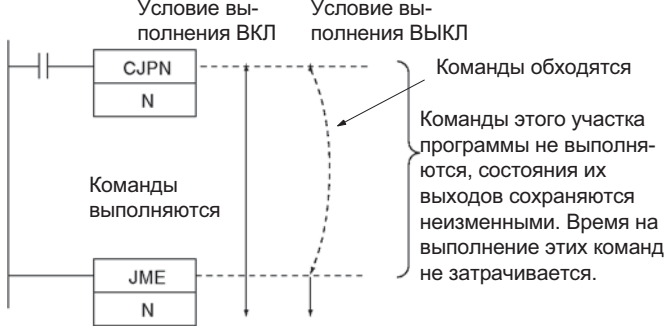
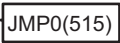
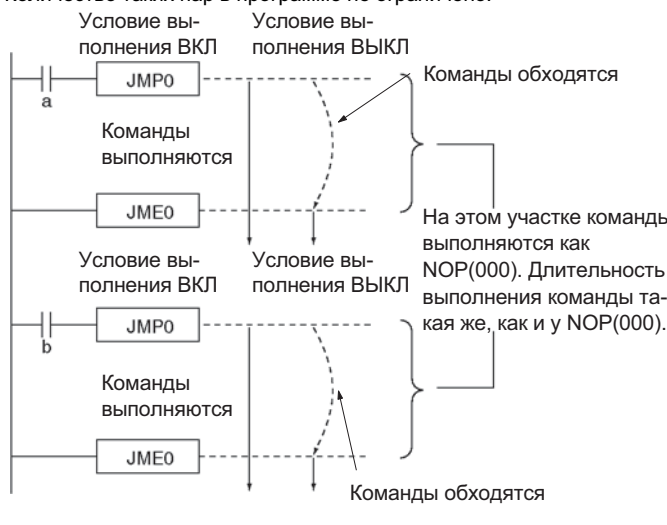
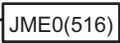


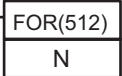
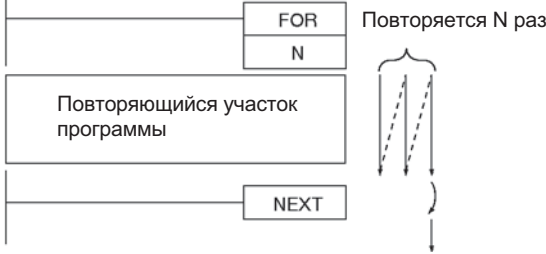
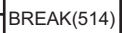
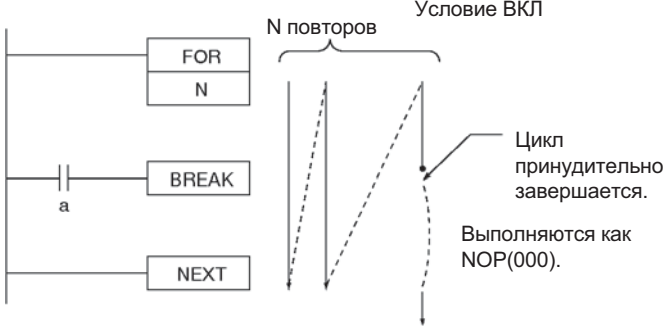

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>SINGLE BIT RESET (только для CS1-H, CJ1-H, CJ1M или CS1D)</b>  RSTB @RSTB !RSTB <sup>*1</sup> !@RSTB <sup>*1</sup>	 D: Адрес слова N: Номер бита	Команда RSTB(533) сбрасывает указанный бит в указанном слове, когда включается условие выполнения.  В отличие от команды RSET, команду RSTB(533) можно использовать для сброса бита в слове DM или EM.	Выход Требуется
<b>SINGLE BIT OUTPUT (только для CS1-H, CJ1-H, CJ1M или CS1D)</b>  OUTB @OUTB !OUTB <sup>*1</sup>	 D: Адрес слова N: Номер бита	Команда OUTB(534) записывает результат (условие выполнения) логической операции в указанный бит.  В отличие от команды OUT, команду OUTB(534) можно использовать для управления битом в слове DM или EM.	Выход Требуется

### 3-3 Команды последовательного управления

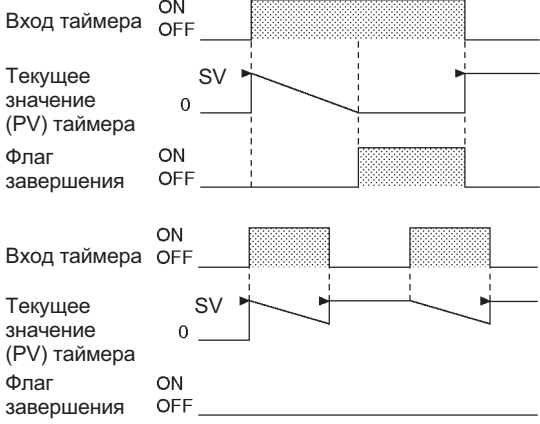
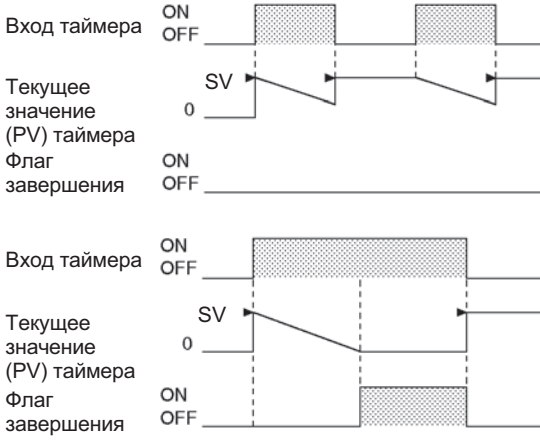
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<p><b>END</b></p> <p>END 001</p>		<p>Указывает на завершение программы. Команда END(001) завершает выполнение программы в данном цикле. Команды, размещенные после END(001), выполняться не будут. Далее начинает выполняться программа задачи со следующим номером. Если выполняется программа задачи с наивысшим номером, команда END(001) означает завершение выполнения полностью всей программы.</p> 	<p>Выход Не требуется</p>
<p><b>NO OPERATION</b></p> <p>NOP 000</p>		<p>Эта команда не выполняет никаких функций (команда NOP(000) не предполагает выполнение каких-либо операций).</p>	<p>Выход Не требуется</p>
<p><b>INTERLOCK</b></p> <p>IL 002</p>		<p>Блокировка всех выходов между IL(002) и ILC(003), когда выключено условие выполнения для IL(002). IL(002) и ILC(003), как правило, используются в паре.</p> 	<p>Выход Требуется</p>

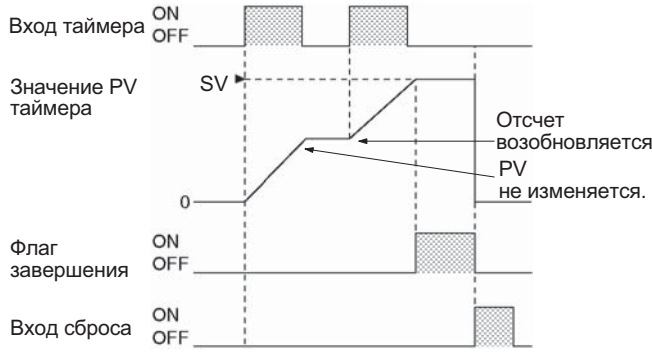
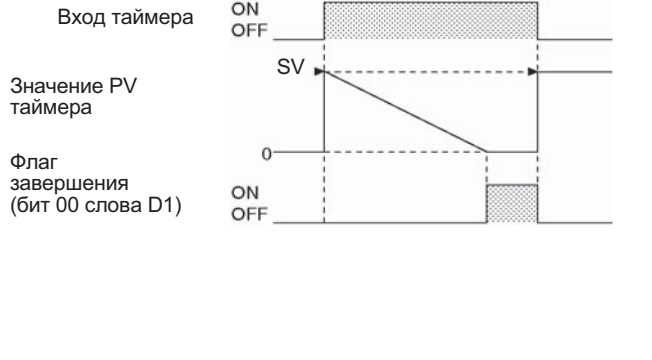
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>INTERLOCK CLEAR</b> ILC 003		Когда сбрасывается (ВЫКЛ) условие выполнения для IL(002), все выходы, расположенные между IL(002) и ILC(003), блокируются. Команды IL(002) и ILC(003) используются, как правило, в паре.	Выход Не требуется
<b>MULTI-INTER-LOCK DIFFERENTIATION HOLD</b> MILH 517 Только для модулей CPU серии CS/CJ версии 2.0 и выше	 N: Номер команды блокировки D: Бит состояния блокировки	Когда сброшено (ВЫКЛ) условие выполнения для MILH(517), выходы всех команд, расположенных между MILH(517) и следующей командой MILC(519), блокируются. MILH(517) и MILC(519) используются в паре. Допускается вложение команд блокировки MILH(517)/MILC(519) (например, MILH(517) - MILH(517) - MILC(519) - MILC(519)). Если между командой MILH(517) и соответствующей командой MILC(519) располагается команда различения состояний (DIFU, DIFD или команда с префиксом @ или %), эта команда будет выполнена после отмены блокировки, если произошло переключение состояния, распознаваемое этой командой.	Выход Требуется
<b>MULTI-INTER-LOCK DIFFERENTIATION RELEASE</b> MILR 518 Только для модулей CPU серии CS/CJ версии 2.0 и выше	 N: Номер команды блокировки D: Бит состояния блокировки	Когда условие выполнения для MILR(518) сброшено (ВЫКЛ), выходы всех команд, расположенных между MILR(518) и следующей командой MILC(519), блокируются. MILR(517) и MILC(519) используются в паре. Допускается вложение команд MILR(517)/MILC(519) (например, MILR(518) - MILR(518) - MILC(519) - MILC(519)). Если между командой MILR(518) и соответствующей командой MILC(519) располагается команда различения состояний (DIFU, DIFD или команда с префиксом @ или %), эта команда будет выполнена после отмены блокировки, если произойдет переключение состояния, распознаваемое этой командой.	Выход Требуется
<b>MULTI-INTER-LOCK CLEAR</b> MILC 519 Только для модулей CPU серии CS/CJ версии 2.0 и выше	 N: Номер команды блокировки	Отмена блокировки, установленной командой MILH(517) или MILR(518) с тем же номером. Когда условие выполнения для MILH(517)/MILR(518) сброшено (ВЫКЛ), все выходы, расположенные между командой MILH(517)/MILR(518) и соответствующей командой MILC(519) с тем же номером, блокируются.	Выход Не требуется
<b>JUMP</b> JMP 004	 N: Номер команды перехода	Когда условие выполнения для JMP(004) сброшено (ВЫКЛ), происходит переход к первой встретившейся в программе команде JME(005) с тем же номером. Команды JMP(004) и JME(005) используются в паре. 	Выход Требуется
<b>JUMP END</b> JME 005	 N: Номер команды перехода	Указывает место, куда должен быть произведен переход, начатый командой JMP(004) или CJP(510).	Выход Не требуется

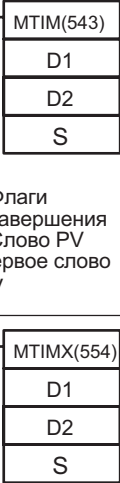
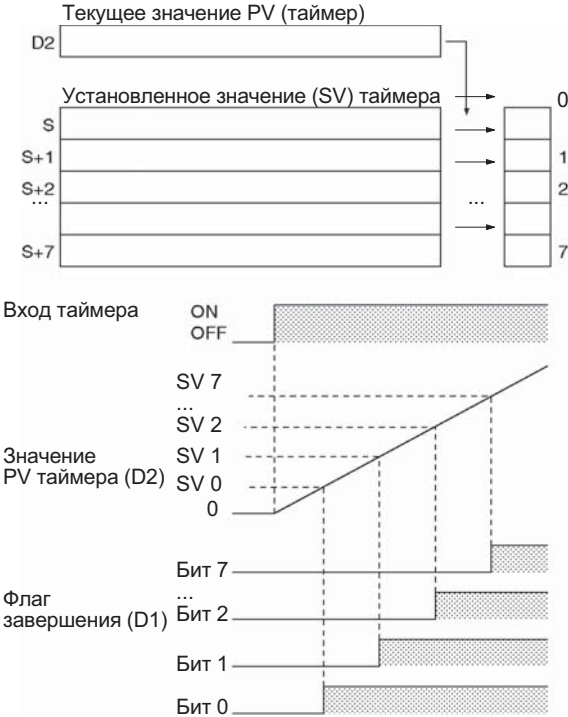
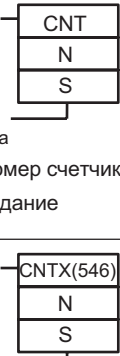
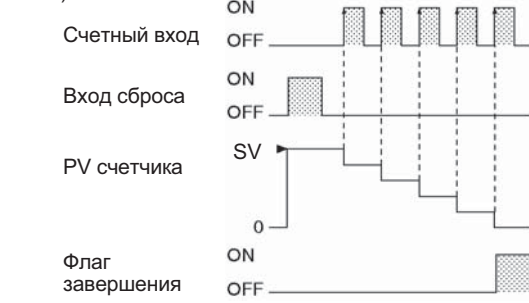
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>CONDITIONAL JUMP</b>  CJP 510	 N: Номер команды перехода	<p>Команда CJP(510) работает, в основном, противоположно команде JMP(004). Когда условие выполнения CJP(510) включено, происходит переход к первой встретившейся в программе команде JME(005) с тем же номером. Команды CJP(510) и JME(005) работают в паре.</p>  <p>Условие выполнения ВЫКЛ      Условие выполнения ВКЛ</p> <p>Команды обходятся</p> <p>Команды этого участка программы не выполняются, состояния их выходов сохраняются неизменными. Время на выполнение этих команд не затрачивается.</p> <p>Команды выполняются</p>	Выход Требуется
<b>CONDITIONAL JUMP</b>  CJPN 511	 N: Номер команды перехода	<p>Команда CJPN(511) работает точно так же, как и команда JMP(004). Когда условие выполнения CJPN(511) сброшено (ВЫКЛ), происходит переход к первой встретившейся в программе команде JME(005) с тем же номером. Команды CJPN(511) и JME(005) работают в паре.</p>  <p>Условие выполнения ВКЛ      Условие выполнения ВЫКЛ</p> <p>Команды обходятся</p> <p>Команды этого участка программы не выполняются, состояния их выходов сохраняются неизменными. Время на выполнение этих команд не затрачивается.</p> <p>Команды выполняются</p>	Выход Не требуется
<b>MULTIPLE JUMP</b>  JMP0 515		<p>Команда условие выполнения для команды JMP0(515) сброшено (ВЫКЛ), все команды, расположенные между JMP0(515) и следующей командой JME0(516), воспринимаются как команда NOP(000). Команды JMP0(515) и JME0(516) должны использоваться в паре. Количество таких пар в программе не ограничено.</p>  <p>Условие выполнения ВКЛ      Условие выполнения ВЫКЛ</p> <p>Команды обходятся</p> <p>На этом участке команды выполняются как NOP(000). Длительность выполнения команды такая же, как и у NOP(000).</p> <p>Команды выполняются</p> <p>Команды обходятся</p>	Выход Требуется
<b>MULTIPLE JUMP END</b>  JME0 516		<p>Когда условие выполнения для команды JMP0(515) сброшено (ВЫКЛ), все команды, расположенные между JMP0(515) и следующей командой JME0(516), воспринимаются как команда NOP(000). Команды JMP0(515) и JME0(516) должны использоваться в паре. Количество таких пар в программе не ограничено.</p>	Выход Не требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>FOR-NEXT LOOPS</b>  FOR 512	 <p>N: Количество циклов</p>	<p>Команды, расположенные между FOR(512) и NEXT(513), повторяются указанное количество раз. Команды FOR(512) и NEXT(513) используются в паре.</p>  <p>Повторяется N раз</p> <p>Повторяющийся участок программы</p> <p>NEXT</p>	Выход Не требуется
<b>BREAK LOOP</b> BREAK 514		<p>Размещается в цикле FOR-NEXT с целью отмены цикла в случае соблюдения условия выполнения. Оставшиеся команды цикла выполняются как команда NOP(000).</p>  <p>Условие ВКЛ</p> <p>N повторов</p> <p>Цикл принудительно завершается.</p> <p>Выполняются как NOP(000).</p> <p>а</p> <p>FOR</p> <p>N</p> <p>BREAK</p> <p>NEXT</p>	Выход Требуется
<b>FOR-NEXT LOOPS</b>  NEXT 513		<p>Команды, расположенные между FOR(512) и NEXT(513), выполняются указанное количество раз. Команды FOR(512) и NEXT(513) используются в паре.</p>	Выход Не требуется

### 3-4 Команды управления таймерами и счетчиками

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<p><b>TIMER</b></p> <p>TIM (BCD)</p> <p>TIMX (Двоичн.) (Только для CS1-H, CJ1-H, CJ1M или CS1D)</p>	<p>TIM</p> <p>N</p> <p>S</p> <p>N: Номер таймера S: Заданное значение</p> <p>TIMX(550)</p> <p>N</p> <p>S</p> <p>N: Номер таймера S: Заданное значение</p>	<p>Команда TIM/TIMX(550) запускает декрементный таймер с шагом 0.1 с. Задание таймера (SV) можно установить в пределах 0 ... 999.9 с (в формате BCD) и 0 ... 6553.5 с (в двоичном формате) (десятичное или шестнадцатеричное значение).</p>  <p>Вход таймера ON OFF</p> <p>Текущее значение (PV) таймера SV 0</p> <p>Флаг завершения ON OFF</p> <p>Вход таймера ON OFF</p> <p>Текущее значение (PV) таймера SV 0</p> <p>Флаг завершения ON OFF</p>	<p>Выход Требуется</p>
<p><b>HIGH-SPEED TIMER</b></p> <p>TIMH 015 (BCD)</p> <p>TIMHX 551 (Двоичн.) (Только для CS1-H, CJ1-H, CJ1M или CS1D)</p>	<p>TIMH(015)</p> <p>N</p> <p>S</p> <p>N: Номер таймера S: Заданное значение</p> <p>TIMHX(551)</p> <p>N</p> <p>S</p> <p>N: Номер таймера S: Заданное значение</p>	<p>Команда TIMH(015)/TIMHX(551) запускает декрементный таймер с шагом 10 мс. Задание таймера (SV) можно выбрать в пределах 0 ... 99.99 с (в формате BCD) или 0 ... 655.35 с (в двоичном формате) (десятичное или шестнадцатеричное значение).</p>  <p>Вход таймера ON OFF</p> <p>Текущее значение (PV) таймера SV 0</p> <p>Флаг завершения ON OFF</p> <p>Вход таймера ON OFF</p> <p>Текущее значение (PV) таймера SV 0</p> <p>Флаг завершения ON OFF</p>	<p>Выход Требуется</p>
<p><b>ONE-MS TIMER</b></p> <p>TMNH 540 (BCD)</p> <p>TMNHX 552 (BCD) (Только для CS1-H, CJ1-H, CJ1M или CS1D)</p>	<p>TMNH(540)</p> <p>N</p> <p>S</p> <p>N: Номер таймера S: Заданное значение</p> <p>TMNHX(552)</p> <p>N</p> <p>S</p> <p>N: Номер таймера S: Заданное значение</p>	<p>Команда TMNH(540)/TMNHX(552) запускает декрементный таймер с шагом 1 мс. Задание таймера (SV) можно выбрать равным 0 ... 9.999 с (в формате BCD) или 0 ... 65.535 с (в двоичном формате) (десятичное или шестнадцатеричное значение).</p> <p>Временная диаграмма таймера TMNH(540) и TMNHX(552) такая же, как и у таймера TIMH(015).</p>	<p>Выход Требуется</p>

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<p><b>ACCUMULATIVE TIMER</b></p> <p>TTIM 087 (BCD)</p> <p>TTIMX 555 (Двоичн.) (Только для CS1-H, CJ1-H, CJ1M или CS1D)</p>	<p>Вход таймера</p> <div style="border: 1px solid black; padding: 2px; width: fit-content;">             TTIM(087)              N              S         </div> <p>Вход сброса</p> <p>N: Номер таймера S: Заданное значение</p> <p>Вход таймера</p> <div style="border: 1px solid black; padding: 2px; width: fit-content;">             TTIMX(555)              N              S         </div> <p>Вход сброса</p> <p>N: Номер таймера S: Заданное значение</p>	<p>Команда TTIM(087)/TTIMX(555) запускает инкрементный таймер с шагом 0.1 с. Время отсчета (SV) таймера можно установить в пределах 0 ... 999.9 с (BCD) или 0 ... 6553.5 с (двоичный формат) (десятичное или шестнадцатеричное значение).</p>  <p>Вход таймера ON OFF</p> <p>Значение PV таймера</p> <p>SV</p> <p>0</p> <p>Отсчет возобновляется. PV не изменяется.</p> <p>Флаг завершения ON OFF</p> <p>Вход сброса ON OFF</p>	<p>Выход Требуется</p>
<p><b>LONG TIMER</b></p> <p>TIML 542 (BCD)</p> <p>TIMLX 553 (Двоичн.) (Только для CS1-H, CJ1-H, CJ1M или CS1D)</p>	<p>Вход таймера</p> <div style="border: 1px solid black; padding: 2px; width: fit-content;">             TIML(542)              D1              D2              S         </div> <p>D1: Флаг завершения D2: Слово PV S: Слово SV</p> <p>Вход таймера</p> <div style="border: 1px solid black; padding: 2px; width: fit-content;">             TIMLX(553)              D1              D2              S         </div> <p>D1: Флаг завершения D2: Слово PV S: Слово SV</p>	<p>Команда TIML(542)/TIMLX(553) запускает декрементный таймер с шагом 0.1 с, который может отсчитывать время на протяжении, приблизительно, 115 дней (BCD) или 49710 дней (двоичный формат) (десятичное или шестнадцатеричное значение).</p>  <p>Вход таймера ON OFF</p> <p>Значение PV таймера</p> <p>SV</p> <p>0</p> <p>Флаг завершения (бит 00 слова D1) ON OFF</p>	<p>Выход Требуется</p>

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<p><b>MULTI-OUTPUT TIMER</b></p> <p>MTIM 543 (BCD)</p> <p>MTIMX 554 (Двоичн.) (Только для CS1-H, CJ1-H, CJ1M или CS1D)</p>	<p>  </p> <p>D1: Флаги завершения D2: Слово PV S: Первое слово SV</p> <p>D1: Флаги завершения D2: Слово PV S: Первое слово SV</p>	<p>Команда MTIM(543)/MTIMX(554) запускает инкрементный таймер с шагом 0.1 с, имеющий восемь независимых заданий (значений SV) и флагов завершения. Время отсчета (SV) можно выбрать в диапазоне 0 ... 999.9 с (BCD) или 0 ... 6553.5 с (двоичный формат) (десятичное или шестнадцатеричное значение).</p> <p>Текущее значение PV (таймер) D2</p> <p>Установленное значение (SV) таймера S S+1 S+2 ... S+7</p>  <p>Вход таймера ON OFF</p> <p>Значение PV таймера (D2) SV 7 ... SV 2 SV 1 SV 0 0</p> <p>Флаг завершения (D1) Бит 7 ... Бит 2 Бит 1 Бит 0</p>	<p>Выход Требуется</p>
<p><b>COUNTER</b></p> <p>CNT (BCD)</p> <p>CNTX 546 (Двоичн.) (Только для CS1-H, CJ1-H, CJ1M или CS1D)</p>	<p>  </p> <p>Счетный вход N: Номер счетчика S: Задание</p> <p>Счетный вход N: Номер счетчика S: Задание</p>	<p>Команда CNT/CNTX(546) запускает декрементный счетчик. Задание для счетчика (SV) можно выбрать в пределах 0 ... 999.9 (BCD) или 0 ... 65535 (двоичный формат) (десятичное или шестнадцатеричное значение).</p>  <p>Счетный вход ON OFF</p> <p>Вход сброса ON OFF</p> <p>PV счетчика SV 0</p> <p>Флаг завершения ON OFF</p>	<p>Выход Требуется</p>



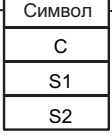
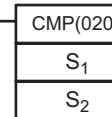

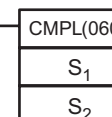
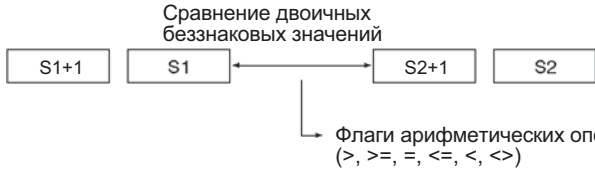


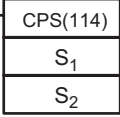
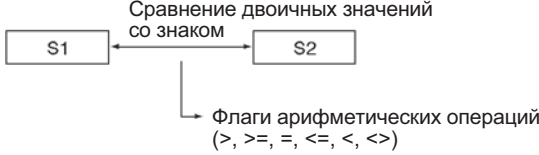
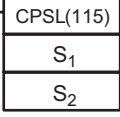

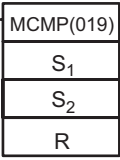
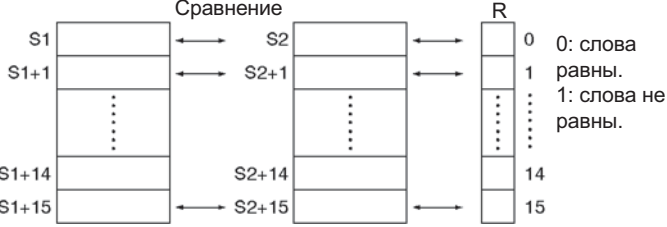
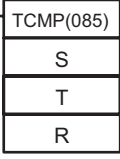
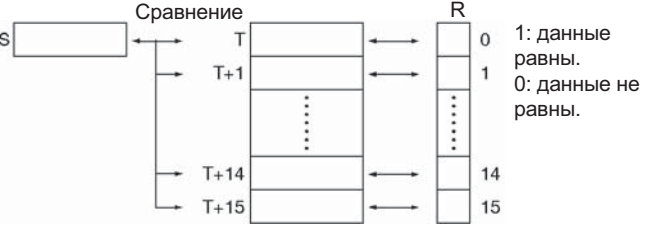
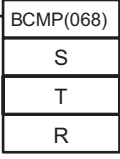

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<p><b>REVERSIBLE COUNTER</b></p> <p>CNTR 012 (BCD)</p> <p>CNTRX 548 (Двоичн.) (Только для CS1-H, CJ1-H, CJ1M или CS1D)</p>	<p>Вход инкремента — CNTR(012)</p> <p>Вход декремента — S</p> <p>Вход сброса — N</p> <p>N: Номер счетчика</p> <p>S: Задание</p> <hr/> <p>Вход инкремента — CNTRX(548)</p> <p>Вход декремента — S</p> <p>Вход сброса — N</p> <p>N: Номер счетчика</p> <p>S: Задание</p>	<p>Команда CNTR(012)/CNTRX(548) запускает реверсивный счетчик.</p>	<p>Выход Требуется</p>
<p><b>RESET TIMER/ COUNTER</b></p> <p>CNR @CNR 545 (BCD)</p> <p>CNRX @CNRX 547 (Двоичн.) (Только для CS1-H, CJ1-H, CJ1M или CS1D)</p>	<p>— CNR(545)</p> <p>N1</p> <p>N2</p> <p>N<sub>1</sub>: Первое число диапазона</p> <p>N<sub>2</sub>: Последнее число диапазона</p> <hr/> <p>— CNRX(547)</p> <p>N1</p> <p>N2</p> <p>N<sub>1</sub>: Первое число 1 диапазона</p> <p>N<sub>2</sub>: Последнее число диапазона</p>	<p>Команда CNR(545)/CNRX(547) сбрасывает таймеры или счетчики в пределах диапазона значений, выбранного для таймера или счетчика. Задание (SV) устанавливается равным максимальному значению 9999.</p>	<p>Выход Требуется</p>

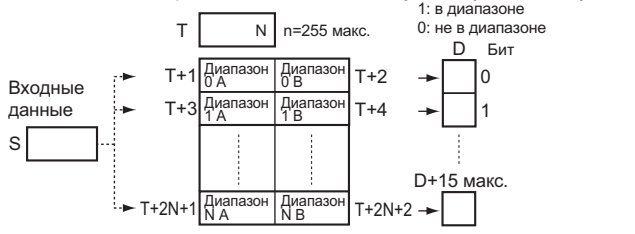
### 3-5 Команды сравнения

\*1: Не поддерживаются модулями CPU CS1D для систем с дублированием CPU.

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>Символ сравнения (без знака)</b> LD, AND, OR + =, <>, <, <=, >, >= 300 (=) 305 (<>) 310 (<) 315 (<=) 320 (>) 325 (>=)	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Символ и опции</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">S<sub>1</sub></div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">S<sub>2</sub></div> <p>S<sub>1</sub>: Сравниваемые данные 1 S<sub>2</sub>: Сравниваемые данные 2</p>	<p>Команды сравнения (для чисел без знака) выполняют сравнение двух величин (констант и/или содержимого указанных слов) в формате 16-битовых слов и устанавливают (ВКЛ) условие выполнения, если сравнение дает положительный результат. Имеется три типа команд сравнения: LD (LOAD), AND и OR.</p> <p><b>LD</b></p> <p><b>AND</b></p> <p><b>OR</b></p>	LD: не требуется AND, OR: требуется
<b>Символ сравнения (двойное слово, без знака)</b> LD, AND, OR + =, <>, <, <=, >, >= + L 301 (=) 306 (<>) 311 (<) 316 (<=) 321 (>) 326 (>=)	<p>S<sub>1</sub>: Сравниваемые данные 1 S<sub>2</sub>: Сравниваемые данные 2</p>	<p>Команды сравнения (для чисел в формате двойного слова без знака) выполняют сравнение двух величин (констант и/или содержимого указанных двойных слов), представленных в формате 32-битового двойного слова без знака, и устанавливают (ВКЛ) условие выполнения, если сравнение дает положительный результат. Имеется три типа команд сравнения: LD (LOAD), AND и OR.</p>	LD: не требуется AND, OR: требуется
<b>Символ сравнения (со знаком)</b> LD, AND, OR + =, <>, <, <=, >, >= + +S 302 (=) 307 (<>) 312 (<) 317 (<=) 322 (>) 327 (>=)	<p>S<sub>1</sub>: Сравниваемые данные 1 S<sub>2</sub>: Сравниваемые данные 2</p>	<p>Команды сравнения (для значений со знаком) выполняют сравнение двух величин (констант и/или содержимого указанных слов), представленных в формате 16-битовых двоичных слов со знаком (4-разрядные 16-ричные значения), и устанавливают (ВКЛ) условие выполнения, если сравнение дает положительный результат. Имеется три типа команд сравнения: LD (LOAD), AND и OR.</p>	LD: не требуется AND, OR: требуется

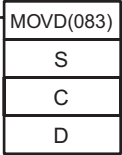
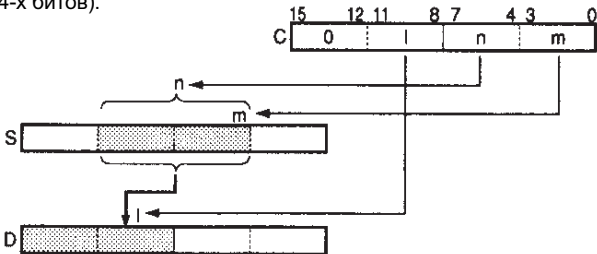
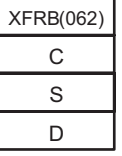
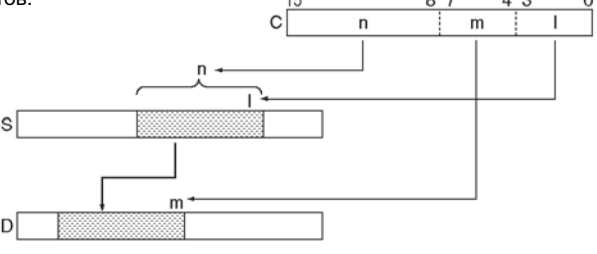
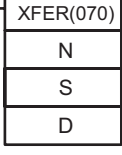
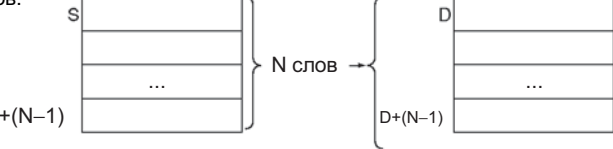
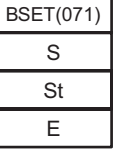

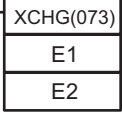
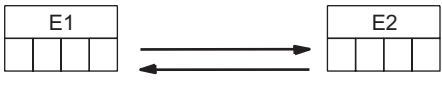
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>Символ сравнения (двойное слово со знаком)</b> LD, AND, OR + =, <, <=, >, >= +SL 303 (=) 308 (<) 313 (<=) 318 (>=) 323 (>) 328 (>=)	<b>S1:</b> Сравнимые данные 1 <b>S2:</b> Сравнимые данные 2	Команды сравнения (для значений в формате двойного слова со знаком) выполняют сравнение двух величин (констант и/или содержащих указанных двойных слов), представленных в формате 32-битовых двоичных значений со знаком (8-разрядные 16-ричные значения), и устанавливают (ВКЛ) условие выполнения, если сравнение дает положительный результат. Имеется три типа команд сравнения: LD (LOAD), AND и OR.	LD: не требуется AND, OR: требуется
<b>Сравнение времени</b> LD, AND, OR + = DT, <> DT, < DT, <= DT, > DT, >= DT 341 (= DT) 342 (<> DT) 343 (< DT) 344 (<= DT) 345 (> DT) 346 (>= DT) (Только для модулей CPU серии CS/CJ версии 2.0 или выше)	LD (LOAD):  AND:  OR:  <b>C:</b> Управляющее слово <b>S1:</b> Первое слово текущего времени <b>S2:</b> Первое слово сравниваемого времени	Команды сравнения времени выполняют сравнение двух значений времени в формате BCD и устанавливают (ВКЛ) условие выполнения, если сравнение дает положительный результат. Имеется три типа команд сравнения времени: LD (LOAD), AND и OR. Значение времени (год, месяц, день, час, минута и секунда) можно маскировать/размаскировать при сравнении, что легко позволяет создавать функции календаря.	LD: не требуется AND, OR: требуется
<b>UNSIGNED COMPARE</b> CMP !CMP*1 020	 <b>S1:</b> Сравнимые данные 1 <b>S2:</b> Сравнимые данные 2	Сравнение двух двоичных значений без знака (констант и/или содержащих указанных слов) и размещение результата во флаги арифметических операций в дополнительной области. 	Выход Требуется
<b>DOUBLE UNSIGNED COMPARE</b> CMPL 060	 <b>S1:</b> Сравнимые данные 1 <b>S2:</b> Сравнимые данные 2	Сравнение двух двойных двоичных беззнаковых значений (констант и/или содержащих указанных слов) и размещение результата во флаге арифметических операций в дополнительной области. 	Выход Требуется

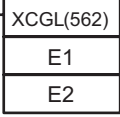
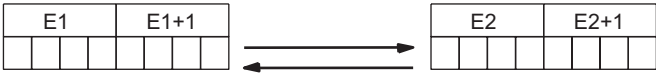
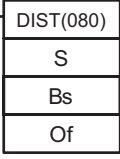
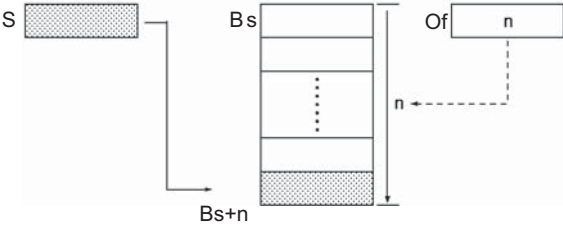
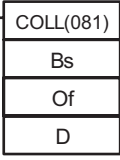
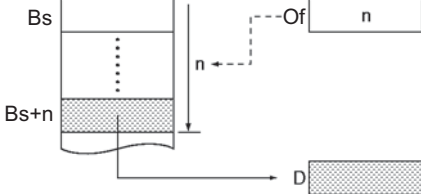
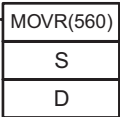
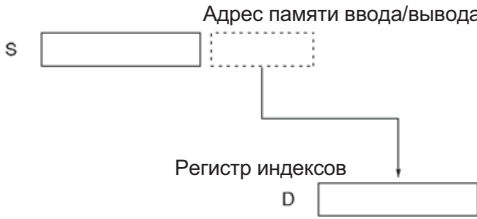
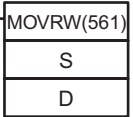
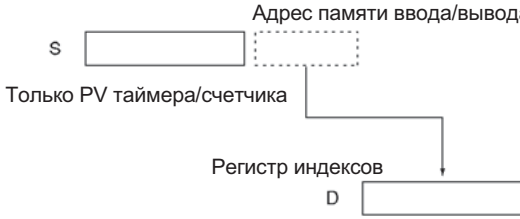
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>SIGNED BINARY COMPARE</b>  CPS !CPS*1 114	 <p>S<sub>1</sub>: Сравниваемые данные 1 S<sub>2</sub>: Сравниваемые данные 2</p>	<p>Сравнение двух двоичных значений со знаком (констант и/или содержимого указанных слов) и размещение результата во флаги арифметических операций дополнительной области.</p>  <p>Сравнение двоичных значений со знаком</p> <p>Флаги арифметических операций (&gt;, &gt;=, =, &lt;=, &lt;, &lt;&gt;)</p>	Выход Требуется
<b>DOUBLE SIGNED BINARY COMPARE</b>  CPSL 115	 <p>S<sub>1</sub>: Сравниваемые данные 1 S<sub>2</sub>: Сравниваемые данные 2</p>	<p>Сравнение двух двойных двоичных значений со знаком (констант и/или содержимого указанных слов) и размещение результата во флаги арифметических операций дополнительной области.</p>  <p>Сравнение двоичных значений со знаком</p> <p>Флаги арифметических операций (&gt;, &gt;=, =, &lt;=, &lt;, &lt;&gt;)</p>	Выход Требуется
<b>MULTIPLE COMPARE</b>  MCMP @MCMP 019	 <p>S<sub>1</sub>: Первое слово набора 1 S<sub>2</sub>: Первое слово набора 2 R: Слово результата</p>	<p>Сравнение шестнадцати последовательных слов с другими шестнадцатью последовательными словами и установка (ВКЛ) бита в слове результата, соответствующего словам, которые не равны.</p>  <p>Сравнение</p> <p>0: слова равны. 1: слова не равны.</p>	Выход Требуется
<b>TABLE COMPARE</b>  TCMP @TCMP 085	 <p>S: Входные данные T: Первое слово таблицы R: Слово результата</p>	<p>Сравнение входного значения с шестнадцатью словами и установка бита в слове результата в случае обнаружения совпадения в соответствующей позиции.</p>  <p>Сравнение</p> <p>0: данные равны. 1: данные не равны.</p>	Выход Требуется
<b>UNSIGNED BLOCK COMPARE</b>  BCMP @BCMP 068	 <p>S: Входные данные T: Первое слово таблицы R: Слово результата</p>	<p>Сравнение входного слова с шестнадцатью последовательностями, определяемыми шестнадцатью нижними и верхними границами, и установка (ВКЛ) бита в соответствующей позиции слова результата, если данные находятся внутри какого-то диапазона.</p>  <p>Диапазоны</p> <p>1: в диапазоне 0: не в диапазоне</p> <p>Нижняя граница Верхняя граница</p> <p>Входные данные</p>	Выход Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения				
<p><b>EXPANDED BLOCK COMPARE</b></p> <p>BCMP2 @BCMP2 502</p> <p>(Только для модулей CPU серии CS1-H, CJ1-H или CS1D версии 2.0 или выше) Модуль CPU CJ1M (до версии 2.0 или версии 2.0 или выше)</p>	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td style="text-align: center;">BCMP2(502)</td></tr> <tr><td style="text-align: center;">S</td></tr> <tr><td style="text-align: center;">T</td></tr> <tr><td style="text-align: center;">R</td></tr> </table> <p>S: Входные данные T: Первое слово блока R: Слово результата</p>	BCMP2(502)	S	T	R	<p>Проверка принадлежности входных данных максимум 256-ти диапазонам (которые определяются верхней и нижней границами) и установка соответствующего бита в слове результата, если входные данные принадлежат соответствующему диапазону.</p>  <p>1: в диапазоне 0: не в диапазоне</p> <p>Д Бит</p> <p>Примечание: A может быть меньше или равно B или может быть больше B.</p>	<p>Выход Требуется</p>
BCMP2(502)							
S							
T							
R							
<p><b>AREA RANGE COMPARE</b></p> <p>ZCP @ZCP 088</p> <p>(Только для CS1-H, CJ1-H, CJ1M или CS1D)</p>	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td style="text-align: center;">ZCP(088)</td></tr> <tr><td style="text-align: center;">CD</td></tr> <tr><td style="text-align: center;">LL</td></tr> <tr><td style="text-align: center;">UL</td></tr> </table> <p>CD: Сравниваемые данные (одно слово) LL: Нижняя граница диапазона UL: Верхняя граница диапазона</p>	ZCP(088)	CD	LL	UL	<p>Определение принадлежности 16-битового двоичного значения без знака (CD) (слово или константа) диапазону, определенному границами LL и UL, и запись результата во флаги арифметических операций дополнительной области.</p>	<p>Выход Требуется</p>
ZCP(088)							
CD							
LL							
UL							
<p><b>DOUBLE AREA RANGE COMPARE</b></p> <p>ZCPL @ZCPL 116</p> <p>(Только для CS1-H, CJ1-H, CJ1M или CS1D)</p>	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td style="text-align: center;">ZCPL(116)</td></tr> <tr><td style="text-align: center;">CD</td></tr> <tr><td style="text-align: center;">LL</td></tr> <tr><td style="text-align: center;">UL</td></tr> </table> <p>CD: Сравниваемые данные (два слова) LL: Нижняя граница диапазона UL: Верхняя граница диапазона</p>	ZCPL(116)	CD	LL	UL	<p>Определение принадлежности 32-битового двоичного значения без знака (регистры CD и CD+1) (слово или константа) диапазону, определенному границами LL и UL, и запись результата во флаги арифметических операций дополнительной области.</p>	<p>Выход Требуется</p>
ZCPL(116)							
CD							
LL							
UL							

### 3-6 Команды перемещения данных

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>MOVE</b> MOV @MOV !MOV !@MOV 021	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                 MOV(021)  <hr/>                 S  <hr/>                 D             </div> <p>S: Источник D: Адресат</p>	<p>Перемещение слова данных в указанное слово.</p> <p>Состояние битов не изменяется.</p>	Выход Требуется
<b>DOUBLE MOVE</b> MOVL @MOVL 498	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                 MOVL(498)  <hr/>                 S  <hr/>                 D             </div> <p>S: Первое слово источника D: Первое слово адресата</p>	<p>Передача двух слов данных в указанные слова.</p> <p>Состояние битов не изменяется.</p>	Выход Требуется
<b>MOVE NOT</b> MVN @MVN 022	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                 MVN(022)  <hr/>                 S  <hr/>                 D             </div> <p>S: Источник D: Адресат</p>	<p>Передача дополнения к слову данных в указанное слово.</p> <p>Состояние битов инвертируется.</p>	Выход Требуется
<b>DOUBLE MOVE NOT</b> MVNL @MVNL 499	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                 MVNL(499)  <hr/>                 S  <hr/>                 D             </div> <p>S: Первое слово источника D: Первое слово адресата</p>	<p>Передача дополнения к двум словам данных в указанные слова.</p> <p>Состояние битов инвертируется.</p>	Выход Требуется
<b>MOVE BIT</b> MOVB @MOVB 082	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                 MOVB(082)  <hr/>                 S  <hr/>                 C  <hr/>                 D             </div> <p>S: Слово или данные источника C: Управляющее слово D: Слово адресата</p>	<p>Передача указанного бита.</p>	Выход Требуется

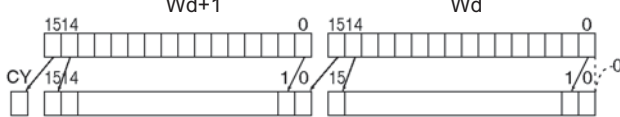
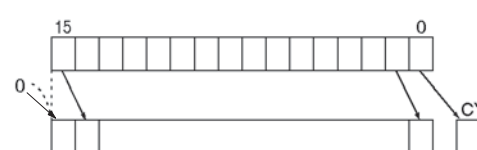
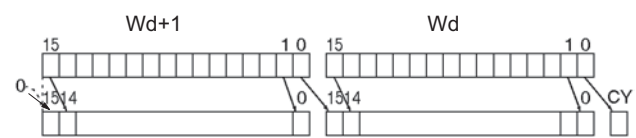


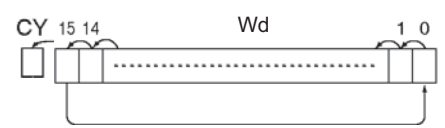
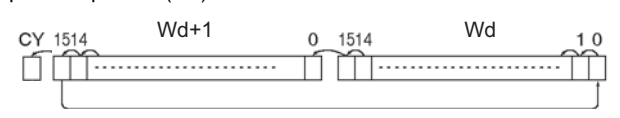
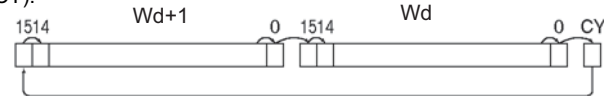
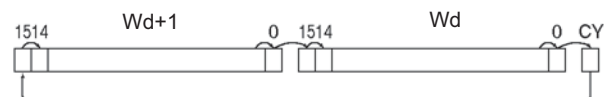
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>MOVE DIGIT</b> MOVD @MOVD 083	 <p>S: Слово или данные источника                      C: Управляющее слово                      D: Слово адресата</p>	<p>Передача указанного разряда или разрядов (каждый разряд состоит из 4-х битов).</p> 	Выход Требуется
<b>MULTIPLE BIT TRANSFER</b> XFRB @XFRB 062	 <p>C: Управляющее слово                      S: Первое слово источника                      D: Первое слово адресата</p>	<p>Передача указанного количества расположенных последовательно битов.</p> 	Выход Требуется
<b>BLOCK TRANSFER</b> XFER @XFER 070	 <p>N: Количество слов                      S: Первое слово источника                      D: Первое слово адресата</p>	<p>Передача указанного количества расположенных последовательно слов.</p> 	Выход Требуется
<b>BLOCK SET</b> BSET @BSET 071	 <p>S: Слово источника                      St: Начальное слово                      E: Конечное слово</p>	<p>Запись одного и того же значения (слова источника) в расположенные последовательно слова в определенном диапазоне.</p> <p>Слово источника      Адресуемые слова</p> 	Выход Требуется
<b>DATA EXCHANGE</b> XCHG @XCHG 073	 <p>E1: Первое слово для обмена                      E2: Второе слово для обмена</p>	<p>Обмен содержимым между двумя указанными словами.</p> 	Выход Требуется

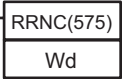
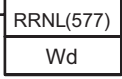
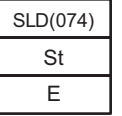
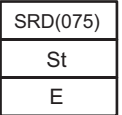
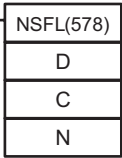
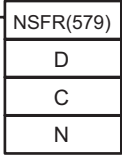
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>DOUBLE DATA EXCHANGE</b> XCGL @XCGL 562	 <p>E1: Первое слово для обмена E2: Второе слово для обмена</p>	Обмен содержимым между двумя парами последовательных слов. 	Выход Требуется
<b>SINGLE WORD DISTRIBUTE</b> DIST @DIST 080	 <p>S: Слово источника Bs: Начальное слово адресата Of: Смещение</p>	Передача содержимого слова источника в адресуемое слово, адрес которого определяется путем добавления величины смещения к начальному адресу. 	Выход Требуется
<b>DATA COLLECT</b> COLL @COLL 081	 <p>Bs: Начальное слово источника Of: Смещение D: Слово адресата</p>	Передача содержимого слова источника (адрес которого определяется путем добавления величины смещения к начальному адресу) в адресуемое слово. 	Выход Требуется
<b>MOVE TO REGISTER</b> MOVR @MOVR 560	 <p>S: Источник (требуемое слово или бит) D: Адресат (регистр индексов)</p>	Размещает в указанном регистре индексов адрес внутренней памяти ввода/вывода для указанного слова, бита или флага завершения таймера/счетчика (чтобы разместить в регистре индексов адрес внутренней памяти ввода/вывода для текущего значения (PV) таймера/счетчика, используйте команду MOVRW(561)). 	Выход Требуется
<b>MOVE TIMER/COUNTER PV TO REGISTER</b> MOVRW @MOVRW 561	 <p>S: Источник (требуемый номер TC) D: Адресат (регистр индексов)</p>	Размещает в указанном регистре индексов адрес внутренней памяти ввода/вывода для текущего значения (PV) указанного таймера или счетчика (чтобы разместить в регистре индексов адрес внутренней памяти ввода/вывода для слова, бита или флага завершения таймера/счетчика, используйте команду MOVR(560)). 	Выход Требуется



### 3-7 Команды сдвига данных

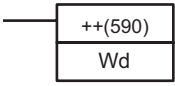

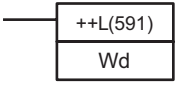



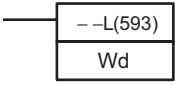
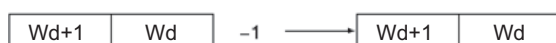
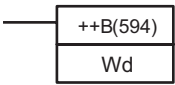



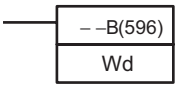



Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>SHIFT REGISTER</b> SFT 010	<p>Вход данных Вход сдвига Вход сброса</p> <p>St: Начальное слово E: Конечное слово</p>	<p>Регистр сдвига</p> <p>Теряются</p> <p>Состояние входа данных для каждого входа сдвига</p>	Выход Требуется
<b>REVERSIBLE SHIFT REGISTER</b> SFTR @SFTR 084	<p>C: Управляющее слово St: Начальное слово E: Конечное слово</p>	<p>Регистр сдвига, в котором данные сдвигаются либо вправо, либо влево.</p> <p>Ввод данных</p> <p>Направление сдвига</p>	Выход Требуется
<b>ASYNCHRONOUS SHIFT REGISTER</b> ASFT @ASFT 017	<p>C: Управляющее слово St: Начальное слово E: Конечное слово</p>	<p>Сдвиг всех ненулевых слов данных в пределах указанного диапазона слов либо в направлении ST, либо в направлении E, с замещением нулевых (0000 Hex) слов данных.</p> <p>Сдвиг</p> <p>Сдвиг</p> <p>Направление сдвига</p> <p>Разрешение сдвига</p> <p>Обнуление</p> <p>Нулевые данные</p> <p>Ненулевые данные</p>	Выход Требуется
<b>WORD SHIFT</b> WSFT @WSFT 016	<p>S: Входное слово St: Начальное слово E: Конечное слово</p>	<p>Сдвиг данных отдельно по словам в диапазоне между ST и E.</p> <p>Теряются</p>	Выход Требуется
<b>ARITHMETIC SHIFT LEFT</b> ASL @ASL 025	<p>Wd: Слово</p>	<p>Сдвиг содержимого Wd на один бит влево.</p>	Выход Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>DOUBLE SHIFT LEFT</b> ASLL @ASLL 570	ASLL(570) Wd	Сдвиг содержимого Wd и Wd+1 на один бит влево. 	Выход Требуется
<b>ARITHMETIC SHIFT RIGHT</b> ASR @ASR 026	ASR(026) Wd	Сдвиг содержимого Wd на один бит вправо. 	Выход Требуется
<b>DOUBLE SHIFT RIGHT</b> ASRL @ASRL 571	ASRL(571) Wd	Сдвиг содержимого Wd и Wd+1 на один бит вправо. 	Выход Требуется
<b>ROTATE LEFT</b> ROL @ROL 027	ROL(027) Wd	Сдвиг всех битов Wd на один бит влево, включая флаг переноса (CY). 	Выход Требуется
<b>DOUBLE ROTATE LEFT</b> ROLL @ROLL 572	ROLL(572) Wd	Сдвиг всех битов Wd и Wd+1 на один бит влево, включая флаг переноса (CY). 	Выход Требуется
<b>ROTATE LEFT WITHOUT CARRY</b> RLNC @RLNC 574	RLNC(574) Wd	Сдвиг всех битов Wd на один бит влево, за исключением флага переноса (CY). 	Выход Требуется
<b>DOUBLE ROTATE LEFT WITHOUT CARRY</b> RLNL @RLNL 576	RLNL(576) Wd	Сдвиг всех битов Wd и Wd+1 на один бит влево, за исключением флага переноса (CY). 	Выход Требуется
<b>ROTATE RIGHT</b> ROR @ROR 028	ROR(028) Wd	Сдвиг всех битов Wd на один бит вправо, включая флаг переноса (CY). 	Выход Требуется
<b>DOUBLE ROTATE RIGHT</b> RORL @RORL 573	RORL(573) Wd	Сдвиг всех битов Wd и Wd+1 на один бит вправо, включая флаг переноса (CY). 	Выход Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>ROTATE RIGHT WITHOUT CARRY</b>  RRNC @RRNC 575	 <p>Wd: Слово</p>	Сдвиг всех битов Wd на один бит вправо, за исключением флага переноса (CY). Содержимое крайнего правого бита в Wd перемещается в позицию крайнего левого бита и в позицию флага переноса (CY).	Выход Требуется
<b>DOUBLE ROTATE RIGHT WITHOUT CARRY</b>  RRNL @RRNL 577	 <p>Wd: Слово</p>	Сдвиг всех битов Wd и Wd+1 на один бит вправо, за исключением флага переноса (CY). Содержимое крайнего правого бита в Wd+1 перемещается в позицию крайнего левого бита Wd и в позицию флага переноса (CY).	Выход Требуется
<b>ONE DIGIT SHIFT LEFT</b>  SLD @SLD 074	 <p>St: Начальное слово E: Конечное слово</p>	Сдвиг данных на один разряд (4 бита) влево.	Выход Требуется
<b>ONE DIGIT SHIFT RIGHT</b>  SRD @SRD 075	 <p>St: Начальное слово E: Конечное слово</p>	Сдвиг данных на один разряд (4 бита) вправо.	Выход Требуется
<b>SHIFT N-BIT DATA LEFT</b>  NSFL @NSFL 578	 <p>D: Первое слово для сдвига C: Первый бит N: Длина сдвигаемых данных</p>	Сдвиг указанного количества битов влево.	Выход Требуется
<b>SHIFT N-BIT DATA RIGHT</b>  NSFR @NSFR 579	 <p>D: Первое слово для сдвига C: Первый бит N: Длина сдвигаемых данных</p>	Сдвиг указанного количества битов вправо.	Выход Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения			
<b>SHIFT N-BITS LEFT</b>  NASL @NASL 580	<table border="1" style="width: 100%; text-align: center;"> <tr><td>NASL(580)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>D: Сдвигаемое слово C: Управляющее слово</p>	NASL(580)	D	C	<p>Сдвиг указанных 16 битов слова данных влево на указанное количество битов.</p> <p>Значение "a" сдвинулось, записалось "0"</p> <p>Н битов</p>	Выход Требуется
NASL(580)						
D						
C						
<b>DOUBLE SHIFT N-BITS LEFT</b>  NSLL @NSLL 582	<table border="1" style="width: 100%; text-align: center;"> <tr><td>NSLL(582)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>D: Сдвигаемое слово C: Управляющее слово</p>	NSLL(582)	D	C	<p>Сдвиг указанных 32 битов слова данных влево на указанное количество битов.</p> <p>Значение "a" сдвинулось, записалось "0"</p> <p>Н битов</p>	Выход Требуется
NSLL(582)						
D						
C						
<b>SHIFT N-BITS RIGHT</b>  NASR @NASR 581	<table border="1" style="width: 100%; text-align: center;"> <tr><td>NASR(581)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>D: Сдвигаемое слово C: Управляющее слово</p>	NASR(581)	D	C	<p>Сдвиг указанных 16 битов слова данных вправо на указанное количество битов.</p> <p>Значение "a" сдвинулось, записалось "0"</p> <p>Н битов</p>	Выход Требуется
NASR(581)						
D						
C						
<b>DOUBLE SHIFT N-BITS RIGHT</b>  NSRL @NSRL 583	<table border="1" style="width: 100%; text-align: center;"> <tr><td>NSRL(583)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>D: Сдвигаемое слово C: Управляющее слово</p>	NSRL(583)	D	C	<p>Сдвиг указанных 32 битов слова данных вправо на указанное количество битов.</p> <p>Значение "a" сдвинулось, записалось "0"</p> <p>Н битов</p>	Выход Требуется
NSRL(583)						
D						
C						

## 3-8 Команды увеличения/уменьшения

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>INCREMENT BINARY</b>  ++ @++ 590	 Wd: Слово	Увеличение 4-разрядного 16-ричного значения указанного слова на 1.  	Выход Требуется
<b>DOUBLE INCREMENT BINARY</b>  ++L @++L 591	 Wd: Слово	Увеличение 8-разрядного 16-ричного значения указанного слова на 1.  	Выход Требуется
<b>DECREMENT BINARY</b>  -- @-- 592	 Wd: Слово	Уменьшение 4-разрядного 16-ричного значения указанного слова на 1.  	Выход Требуется
<b>DOUBLE DECREMENT BINARY</b>  --L @--L 593	 Wd: Первое слово	Уменьшение 8-разрядного 16-ричного значения указанного слова на 1.  	Выход Требуется
<b>INCREMENT BCD</b>  ++B @++B 594	 Wd: Word	Увеличение 4-разрядного содержимого в формате BCD указанного слова на 1.  	Выход Требуется
<b>DOUBLE INCREMENT BCD</b>  ++BL @++BL 595	 Wd: Первое слово	Увеличение 8-разрядного содержимого в формате BCD указанного слова на 1.  	Выход Требуется
<b>DECREMENT BCD</b>  --B @--B 596	 Wd: Слово	Уменьшение 4-разрядного содержимого в формате BCD указанного слова на 1.  	Выход Требуется
<b>DOUBLE DECREMENT BCD</b>  --BL @--BL 597	 Wd: Первое слово	Уменьшение 8-разрядного содержимого в формате BCD указанного слова на 1.  	Выход Требуется

## 3-9 Символьные математические команды

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>SIGNED BINARY ADD WITHOUT CARRY</b>  + @+ 400	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           +(400)            Au            Ad            R         </div> <p>Au: Слово 1-го слагаемого Ad: Слово 2-го слагаемого R: Слово результата</p>	<p>Сложение 4-разрядных (одно слово) 16-ричных данных и/или констант.</p> $\begin{array}{r} \boxed{Au} \quad (\text{Двоичное со знаком}) \\ + \quad \boxed{Ad} \quad (\text{Двоичное со знаком}) \\ \hline \boxed{CY} \quad \boxed{R} \quad (\text{Двоичное со знаком}) \end{array}$ <p>В случае переноса будет установлен флаг CY .</p>	Выход Требуется
<b>DOUBLE SIGNED BINARY ADD WITHOUT CARRY</b>  +L @+L 401	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           +L(401)            Au            Ad            R         </div> <p>Au: Первое слово 1-го слагаемого Ad: Первое слово 2-го слагаемого R: Первое слово результата</p>	<p>Сложение 8-разрядных (двойное слово) 16-ричных данных и/или констант.</p> $\begin{array}{r} \boxed{Au+1} \quad \boxed{Au} \quad (\text{Двоичное со знаком}) \\ + \quad \boxed{Ad+1} \quad \boxed{Ad} \quad (\text{Двоичное со знаком}) \\ \hline \boxed{CY} \quad \boxed{R+1} \quad \boxed{R} \quad (\text{Двоичное со знаком}) \end{array}$ <p>В случае переноса будет установлен флаг CY .</p>	Выход Требуется
<b>SIGNED BINARY ADD WITH CARRY</b>  +C @+C 402	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           +C(402)            Au            Ad            R         </div> <p>Au: Слово 1-го слагаемого Ad: Слово 2-го слагаемого R: Слово результата</p>	<p>Сложение 4-разрядных (одно слово) 16-ричных данных и/или констант с флагом переноса (CY).</p> $\begin{array}{r} \boxed{Au} \quad (\text{Двоичное со знаком}) \\ + \quad \boxed{Ad} \quad (\text{Двоичное со знаком}) \\ \quad \quad \quad \boxed{CY} \\ \hline \boxed{CY} \quad \boxed{R} \quad (\text{Двоичное со знаком}) \end{array}$ <p>В случае переноса будет установлен флаг CY .</p>	Выход Требуется
<b>DOUBLE SIGNED BINARY ADD WITH CARRY</b>  +CL @+CL 403	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           +CL(403)            Au            Ad            R         </div> <p>Au: Первое слово 1-го слагаемого Ad: Первое слово 2-го слагаемого R: Первое слово результата</p>	<p>Сложение 8-разрядных (двойное слово) 16-ричных данных и/или констант с флагом переноса (CY).</p> $\begin{array}{r} \boxed{Au+1} \quad \boxed{Au} \quad (\text{Двоичное со знаком}) \\ \quad \quad \quad \boxed{Ad+1} \quad \boxed{Ad} \quad (\text{Двоичное со знаком}) \\ \quad \quad \quad \quad \quad \quad \boxed{CY} \\ + \\ \hline \boxed{CY} \quad \boxed{R+1} \quad \boxed{R} \quad (\text{Двоичное со знаком}) \end{array}$ <p>В случае переноса будет установлен флаг CY .</p>	Выход Требуется
<b>BCD ADD WITH- OUT CARRY</b>  +B @+B 404	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           +B(404)            Au            Ad            R         </div> <p>Au: Слово 1-го слагаемого Ad: Слово 2-го слагаемого R: Слово результата</p>	<p>Сложение 4-разрядных (одно слово) данных в формате BCD и/или констант.</p> $\begin{array}{r} \boxed{Au} \quad (\text{BCD}) \\ + \quad \boxed{Ad} \quad (\text{BCD}) \\ \hline \boxed{CY} \quad \boxed{R} \quad (\text{BCD}) \end{array}$ <p>В случае переноса будет установлен флаг CY .</p>	Выход Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>DOUBLE BCD ADD WITHOUT CARRY</b>  +BL @+BL 405	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> +BL(405)  <hr/> Au  <hr/> Ad  <hr/> R </div> <p>Au: Первое слово 1-го слагаемого Ad: Первое слово 2-го слагаемого R: Первое слово результата</p>	<p>Сложение 8-разрядных (двойное слово) данных в формате BCD и/или констант.</p> $\begin{array}{r} \boxed{Au+1} \quad \boxed{Au} \quad (\text{BCD}) \\ + \quad \boxed{Ad+1} \quad \boxed{Ad} \quad (\text{BCD}) \\ \hline \boxed{CY} \quad \boxed{R+1} \quad \boxed{R} \quad (\text{BCD}) \end{array}$ <p>В случае переноса будет установлен флаг CY .</p>	Выход Требуется
<b>BCD ADD WITH CARRY</b>  +BC @+BC 406	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> +BC(406)  <hr/> Au  <hr/> Ad  <hr/> R </div> <p>Au: Слово 1-го слагаемого Ad: Слово 2-го слагаемого R: Слово рез-та</p>	<p>Сложение 4-разрядных (одно слово) данных в формате BCD и/или констант с флагом переноса (CY).</p> $\begin{array}{r} \boxed{Au} \quad (\text{BCD}) \\ + \quad \boxed{Ad} \quad (\text{BCD}) \\ \hline \boxed{CY} \quad \boxed{R} \quad (\text{BCD}) \end{array}$ <p>В случае переноса будет установлен флаг CY .</p>	Выход Требуется
<b>DOUBLE BCD ADD WITH CARRY</b>  +BCL @+BCL 407	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> +BCL(407)  <hr/> Au  <hr/> Ad  <hr/> R </div> <p>Au: Первое слово 1-го слагаемого Ad: Первое слово 2-го слагаемого R: Первое слово результата</p>	<p>Сложение 8-разрядных (двойное слово) данных в формате BCD и/или констант с флагом переноса (CY).</p> $\begin{array}{r} \boxed{Au+1} \quad \boxed{Au} \quad (\text{BCD}) \\ + \quad \boxed{Ad+1} \quad \boxed{Ad} \quad (\text{BCD}) \\ \hline \boxed{CY} \quad \boxed{R+1} \quad \boxed{R} \quad (\text{BCD}) \end{array}$ <p>В случае переноса будет установлен флаг CY .</p>	Выход Требуется
<b>SIGNED BINARY SUBTRACT WITHOUT CARRY</b>  - @- 410	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> -(410)  <hr/> Mi  <hr/> Su  <hr/> R </div> <p>Mi: Уменьшаемое слово Su: Вычитаемое слово R: Слово рез-та</p>	<p>Операция вычитания над 4-разрядными (одно слово) 16-ричными данными и/или константами.</p> $\begin{array}{r} \boxed{Mi} \quad (\text{Двоичное со знаком}) \\ - \quad \boxed{Su} \quad (\text{Двоичное со знаком}) \\ \hline \boxed{CY} \quad \boxed{R} \quad (\text{Двоичное со знаком}) \end{array}$ <p>В случае заема установится флаг CY .</p>	Выход Требуется
<b>DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY</b>  -L @-L 411	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> -L(411)  <hr/> Mi  <hr/> Su  <hr/> R </div> <p>Mi: Уменьшаемое слово Su: Вычитаемое слово R: Слово рез-та</p>	<p>Операция вычитания над 8-разрядными (двойное слово) 16-ричными данными и/или константами.</p> $\begin{array}{r} \boxed{Mi+1} \quad \boxed{Mi} \quad (\text{Двоичное со знаком}) \\ - \quad \boxed{Su+1} \quad \boxed{Su} \quad (\text{Двоичное со знаком}) \\ \hline \boxed{CY} \quad \boxed{R+1} \quad \boxed{R} \quad (\text{Двоичное со знаком}) \end{array}$ <p>В случае заема установится флаг CY .</p>	Выход Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>SIGNED BINARY SUBTRACT WITH CARRY</b> -C @-C 412	<div style="border: 1px solid black; padding: 5px; width: fit-content;">       -C(412)  <hr/>       Mi  <hr/>       Su  <hr/>       R     </div> <p>Mi: Уменьшаемое слово Su: Вычитаемое слово R: Слово рез-та</p>	<p>Операция вычитания над 4-разрядными (одно слово) 16-ричными данными и/или константами с флагом переноса (CY).</p> <div style="text-align: center;"> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Mi</div> <div>(Двоичное со знаком)</div> </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 10px;"> <div style="border: 1px solid black; padding: 2px 10px;">Su</div> <div>(Двоичное со знаком)</div> </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 10px;"> <div style="border: 1px solid black; padding: 2px 10px;">CY</div> <div style="border: 1px solid black; padding: 2px 10px;">R</div> <div>(Двоичное со знаком)</div> </div> </div> <p>В случае заема устанавливается флаг CY.</p>	Выход Требуется
<b>DOUBLE SIGNED BINARY SUBTRACT WITH CARRY</b> -CL @-CL 413	<div style="border: 1px solid black; padding: 5px; width: fit-content;">       -CL(413)  <hr/>       Mi  <hr/>       Su  <hr/>       R     </div> <p>Mi: Уменьшаемое слово Su: Вычитаемое слово R: Слово рез-та</p>	<p>Операция вычитания над 8-разрядными (двойное слово) 16-ричными данными и/или константами с флагом переноса (CY).</p> <div style="text-align: center;"> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Mi+1</div> <div style="border: 1px solid black; padding: 2px 10px;">Mi</div> <div>(Двоичное со знаком)</div> </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 10px;"> <div style="border: 1px solid black; padding: 2px 10px;">Su+1</div> <div style="border: 1px solid black; padding: 2px 10px;">Su</div> <div>(Двоичное со знаком)</div> </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 10px;"> <div style="border: 1px solid black; padding: 2px 10px;">CY</div> <div style="border: 1px solid black; padding: 2px 10px;">R+1</div> <div style="border: 1px solid black; padding: 2px 10px;">R</div> <div>(Двоичное со знаком)</div> </div> </div> <p>В случае заема устанавливается флаг CY.</p>	Выход Требуется
<b>BCD SUBTRACT WITHOUT CARRY</b> -B @-B 414	<div style="border: 1px solid black; padding: 5px; width: fit-content;">       -B(414)  <hr/>       Mi  <hr/>       Su  <hr/>       R     </div> <p>Mi: Уменьшаемое слово Su: Вычитаемое слово R: Слово рез-та</p>	<p>Операция вычитания над 4-разрядными (одно слово) данными в формате BCD и/или константами.</p> <div style="text-align: center;"> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Mi</div> <div>(BCD)</div> </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 10px;"> <div style="border: 1px solid black; padding: 2px 10px;">Su</div> <div>(BCD)</div> </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 10px;"> <div style="border: 1px solid black; padding: 2px 10px;">CY</div> <div style="border: 1px solid black; padding: 2px 10px;">R</div> <div>(BCD)</div> </div> </div> <p>В случае заема устанавливается флаг CY.</p>	Выход Требуется
<b>DOUBLE BCD SUBTRACT WITHOUT CARRY</b> -BL @-BL 415	<div style="border: 1px solid black; padding: 5px; width: fit-content;">       -BL(415)  <hr/>       Mi  <hr/>       Su  <hr/>       R     </div> <p>Mi: 1-е уменьшаемое слово Su: 1-е вычитаемое слово R: 1-е слово результата</p>	<p>Операция вычитания над 8-разрядными (двойное слово) данными в формате BCD и/или константами.</p> <div style="text-align: center;"> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Mi + 1</div> <div style="border: 1px solid black; padding: 2px 10px;">Mi</div> <div>(BCD)</div> </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 10px;"> <div style="border: 1px solid black; padding: 2px 10px;">Su+1</div> <div style="border: 1px solid black; padding: 2px 10px;">Su</div> <div>(BCD)</div> </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 10px;"> <div style="border: 1px solid black; padding: 2px 10px;">CY</div> <div style="border: 1px solid black; padding: 2px 10px;">R+1</div> <div style="border: 1px solid black; padding: 2px 10px;">R</div> <div>(BCD)</div> </div> </div> <p>В случае заема устанавливается флаг CY.</p>	Выход Требуется
<b>BCD SUBTRACT WITH CARRY</b> -BC @-BC 416	<div style="border: 1px solid black; padding: 5px; width: fit-content;">       -BC(416)  <hr/>       Mi  <hr/>       Su  <hr/>       R     </div> <p>Mi: Уменьшаемое слово Su: Вычитаемое слово R: Слово рез-та</p>	<p>Операция вычитания над 4-разрядными (одно слово) данными в формате BCD и/или константами с флагом переноса (CY).</p> <div style="text-align: center;"> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Mi</div> <div>(BCD)</div> </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 10px;"> <div style="border: 1px solid black; padding: 2px 10px;">Su</div> <div>(BCD)</div> </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 10px;"> <div style="border: 1px solid black; padding: 2px 10px;">CY</div> <div style="border: 1px solid black; padding: 2px 10px;">R</div> <div>(BCD)</div> </div> </div> <p>В случае заема устанавливается флаг CY.</p>	Выход Требуется



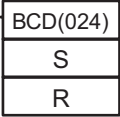

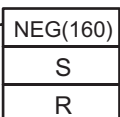
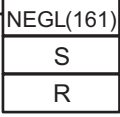
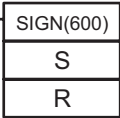
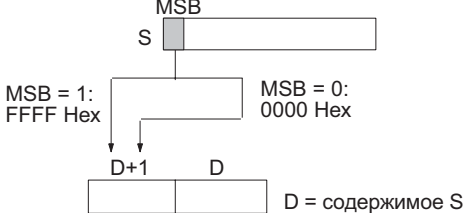
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения																				
<b>DOUBLE BCD SUBTRACT WITH CARRY</b> -BCL @-BCL 417	<table border="1"> <tr><td>-BCL(417)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi: 1-е уменьшаемое слово Su: 1-е вычитаемое слово R: 1-е слово результата</p>	-BCL(417)	Mi	Su	R	<p>Операция вычитания над 8-разрядными (двойное слово) данными в формате BCD и/или константами с флагом переноса (CY).</p> <table border="1"> <tr><td>Mi + 1</td><td>Mi</td><td>(BCD)</td></tr> <tr><td>Su + 1</td><td>Su</td><td>(BCD)</td></tr> <tr><td colspan="3" style="text-align: center;">-</td></tr> <tr><td colspan="2"></td><td>CY</td></tr> <tr><td>CY</td><td>R + 1</td><td>R (BCD)</td></tr> </table> <p>В случае заема устанавливается флаг CY.</p>	Mi + 1	Mi	(BCD)	Su + 1	Su	(BCD)	-					CY	CY	R + 1	R (BCD)	Выход Требуется	
-BCL(417)																							
Mi																							
Su																							
R																							
Mi + 1	Mi	(BCD)																					
Su + 1	Su	(BCD)																					
-																							
		CY																					
CY	R + 1	R (BCD)																					
<b>SIGNED BINARY MULTIPLY</b> * @* 420	<table border="1"> <tr><td>*(420)</td></tr> <tr><td>Md</td></tr> <tr><td>Mr</td></tr> <tr><td>R</td></tr> </table> <p>Md: Слово умножаемого Mr: Слово множителя R: Слово рез-та</p>	*(420)	Md	Mr	R	<p>Операция перемножения 4-разрядных 16-ричных чисел со знаком и/или констант.</p> <table border="1"> <tr><td>Md</td><td>(Двоичное со знаком)</td></tr> <tr><td>×</td><td>Mr</td><td>(Двоичное со знаком)</td></tr> <tr><td colspan="3" style="text-align: center;">-----</td></tr> <tr><td>R + 1</td><td>R</td><td>(Двоичное со знаком)</td></tr> </table>	Md	(Двоичное со знаком)	×	Mr	(Двоичное со знаком)	-----			R + 1	R	(Двоичное со знаком)	Выход Требуется					
*(420)																							
Md																							
Mr																							
R																							
Md	(Двоичное со знаком)																						
×	Mr	(Двоичное со знаком)																					
-----																							
R + 1	R	(Двоичное со знаком)																					
<b>DOUBLE SIGNED BINARY MULTIPLY</b> *L @*L 421	<table border="1"> <tr><td>*L(421)</td></tr> <tr><td>Md</td></tr> <tr><td>Mr</td></tr> <tr><td>R</td></tr> </table> <p>Md: 1-е слово умножаемого Mr: 1-е слово множителя R: 1-е слово результата</p>	*L(421)	Md	Mr	R	<p>Операция перемножения 8-разрядных 16-ричных чисел со знаком и/или констант.</p> <table border="1"> <tr><td>Md + 1</td><td>Md</td><td>(Двоичное со знаком)</td></tr> <tr><td>×</td><td>Mr + 1</td><td>Mr</td><td>(Двоичное со знаком)</td></tr> <tr><td colspan="4" style="text-align: center;">-----</td></tr> <tr><td>R + 3</td><td>R + 2</td><td>R + 1</td><td>R</td><td>(Двоичное со знаком)</td></tr> </table>	Md + 1	Md	(Двоичное со знаком)	×	Mr + 1	Mr	(Двоичное со знаком)	-----				R + 3	R + 2	R + 1	R	(Двоичное со знаком)	Выход Требуется
*L(421)																							
Md																							
Mr																							
R																							
Md + 1	Md	(Двоичное со знаком)																					
×	Mr + 1	Mr	(Двоичное со знаком)																				
-----																							
R + 3	R + 2	R + 1	R	(Двоичное со знаком)																			
<b>UNSIGNED BINARY MULTIPLY</b> *U @*U 422	<table border="1"> <tr><td>*U(422)</td></tr> <tr><td>Md</td></tr> <tr><td>Mr</td></tr> <tr><td>R</td></tr> </table> <p>Md: Слово умножаемого Mr: Слово множителя R: Слово рез-та</p>	*U(422)	Md	Mr	R	<p>Операция перемножения 4-разрядных 16-ричных чисел без знака и/или констант.</p> <table border="1"> <tr><td>Md</td><td>(Двоичное без знака)</td></tr> <tr><td>×</td><td>Mr</td><td>(Двоичное без знака)</td></tr> <tr><td colspan="3" style="text-align: center;">-----</td></tr> <tr><td>R + 1</td><td>R</td><td>(Двоичное без знака)</td></tr> </table>	Md	(Двоичное без знака)	×	Mr	(Двоичное без знака)	-----			R + 1	R	(Двоичное без знака)	Выход Требуется					
*U(422)																							
Md																							
Mr																							
R																							
Md	(Двоичное без знака)																						
×	Mr	(Двоичное без знака)																					
-----																							
R + 1	R	(Двоичное без знака)																					
<b>DOUBLE UNSIGNED BINARY MULTIPLY</b> *UL @*UL 423	<table border="1"> <tr><td>*UL(423)</td></tr> <tr><td>Md</td></tr> <tr><td>Mr</td></tr> <tr><td>R</td></tr> </table> <p>Md: 1-е слово умножаемого Mr: 1-е слово множителя R: 1-е слово результата</p>	*UL(423)	Md	Mr	R	<p>Операция перемножения 8-разрядных 16-ричных чисел без знака и/или констант.</p> <table border="1"> <tr><td>Md + 1</td><td>Md</td><td>(Двоичное без знака)</td></tr> <tr><td>×</td><td>Mr + 1</td><td>Mr</td><td>(Двоичное без знака)</td></tr> <tr><td colspan="4" style="text-align: center;">-----</td></tr> <tr><td>R + 3</td><td>R + 2</td><td>R + 1</td><td>R</td><td>(Двоичное без знака)</td></tr> </table>	Md + 1	Md	(Двоичное без знака)	×	Mr + 1	Mr	(Двоичное без знака)	-----				R + 3	R + 2	R + 1	R	(Двоичное без знака)	Выход Требуется
*UL(423)																							
Md																							
Mr																							
R																							
Md + 1	Md	(Двоичное без знака)																					
×	Mr + 1	Mr	(Двоичное без знака)																				
-----																							
R + 3	R + 2	R + 1	R	(Двоичное без знака)																			

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>BCD MULTIPLY</b> *B @*B 424	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           *(424) Md Mr R         </div> <p>Md: Слово умножаемого Mr: Слово множителя R: Слово рез-та</p>	<p>Операция перемножения 4-разрядных (одно слово) чисел в формате BCD и/или констант.</p> $\begin{array}{r} \boxed{\text{Md}} \text{ (BCD)} \\ \times \quad \boxed{\text{Mr}} \text{ (BCD)} \\ \hline \boxed{\text{R} + 1} \quad \boxed{\text{R}} \text{ (BCD)} \end{array}$	Выход Требуется
<b>DOUBLE BCD MULTIPLY</b> *BL @*BL 425	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           *(425) Md Mr R         </div> <p>Md: 1-е слово умножаемого Mr: 1-е слово множителя R: 1-е слово результата</p>	<p>Операция перемножения 8-разрядных (двойное слово) чисел в формате BCD и/или констант.</p> $\begin{array}{r} \boxed{\text{Md} + 1} \quad \boxed{\text{Md}} \text{ (BCD)} \\ \times \quad \boxed{\text{Mr} + 1} \quad \boxed{\text{Mr}} \text{ (BCD)} \\ \hline \boxed{\text{R} + 3} \quad \boxed{\text{R} + 2} \quad \boxed{\text{R} + 1} \quad \boxed{\text{R}} \text{ (BCD)} \end{array}$	Выход Требуется
<b>SIGNED BINARY DIVIDE</b> / @/ 430	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           /(430) Dd Dr R         </div> <p>Dd: Слово делимого Dr: Слово делителя R: Слово рез-та</p>	<p>Операция деления над 4-разрядными (одно слово) 16-ричными числами со знаком и/или константами.</p> $\begin{array}{r} \boxed{\text{Dd}} \text{ (Двоичное со знаком)} \\ \div \quad \boxed{\text{Dr}} \text{ (Двоичное со знаком)} \\ \hline \boxed{\text{R} + 1} \quad \boxed{\text{R}} \text{ (Двоичное со знаком)} \end{array}$ <p>Остаток                  Частное</p>	Выход Требуется
<b>DOUBLE SIGNED BINARY DIVIDE</b> /L @/L 431	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           /L(431) Dd Dr R         </div> <p>Dd: 1-е слово делимого Dr: 1-е слово делителя R: 1-е слово результата</p>	<p>Операция деления над 8-разрядными (двойное слово) 16-ричными числами со знаком и/или константами.</p> $\begin{array}{r} \boxed{\text{Dd} + 1} \quad \boxed{\text{Dd}} \text{ (Двоичное со знаком)} \\ \div \quad \boxed{\text{Dr} + 1} \quad \boxed{\text{Dr}} \text{ (Двоичное со знаком)} \\ \hline \boxed{\text{R} + 3} \quad \boxed{\text{R} + 2} \quad \boxed{\text{R} + 1} \quad \boxed{\text{R}} \text{ (Двоичное со знаком)} \end{array}$ <p>Остаток                  Частное</p>	Выход Требуется
<b>UNSIGNED BINARY DIVIDE</b> /U @/U 432	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           /U(432) Dd Dr R         </div> <p>Dd: Слово делимого Dr: Слово делителя R: Слово рез-та</p>	<p>Операция деления над 4-разрядными (одно слово) 16-ричными числами без знака и/или константами.</p> $\begin{array}{r} \boxed{\text{Dd}} \text{ (Двоичное без знака)} \\ \div \quad \boxed{\text{Dr}} \text{ (Двоичное без знака)} \\ \hline \boxed{\text{R} + 1} \quad \boxed{\text{R}} \text{ (Двоичное без знака)} \end{array}$ <p>Остаток                  Частное</p>	Выход Требуется

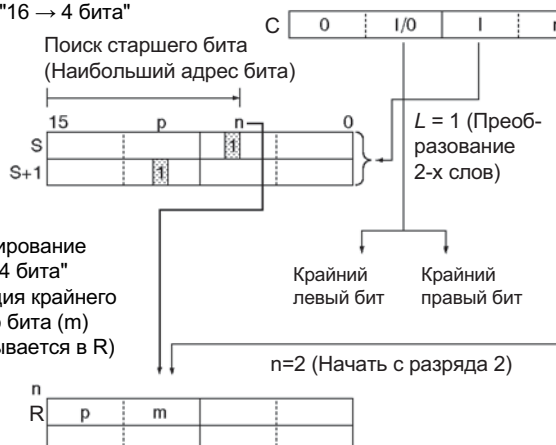


Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения				
<b>DOUBLE UNSIGNED BINARY DIVIDE</b> /UL @/UL 433	<table border="1"> <tr><td>/UL(433)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: 1-е слово делимого Dr: 1-е слово делителя R: 1-е слово результата</p>	/UL(433)	Dd	Dr	R	<p>Операция деления над 8-разрядными (двойное слово) 16-ричными числами без знака и/или константами.</p> $\begin{array}{r} \boxed{Dd + 1} \quad \boxed{Dd} \text{ (Двоичное без знака)} \\ \div \quad \boxed{Dr + 1} \quad \boxed{Dr} \text{ (Двоичное без знака)} \\ \hline \boxed{R + 3} \quad \boxed{R + 2} \quad \boxed{R + 1} \quad \boxed{R} \text{ (Двоичное без знака)} \\ \text{Остаток} \quad \text{Частное} \end{array}$	Выход Требуется
/UL(433)							
Dd							
Dr							
R							
<b>BCD DIVIDE</b> /B @/B 434	<table border="1"> <tr><td>/B(434)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: Слово делимого Dr: Слово делителя R: Слово рез-та</p>	/B(434)	Dd	Dr	R	<p>Операция деления над 4-разрядными (одно слово) числами в формате BCD и/или константами.</p> $\begin{array}{r} \boxed{Dd} \text{ (BCD)} \\ \div \quad \boxed{Dr} \text{ (BCD)} \\ \hline \boxed{R + 1} \quad \boxed{R} \text{ (BCD)} \\ \text{Остаток} \quad \text{Частное} \end{array}$	Выход Требуется
/B(434)							
Dd							
Dr							
R							
<b>DOUBLE BCD DIVIDE</b> /BL @/BL 435	<table border="1"> <tr><td>/BL(435)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: 1-е слово делимого Dr: 1-е слово делителя R: 1-е слово результата</p>	/BL(435)	Dd	Dr	R	<p>Операция деления над 8-разрядными (двойное слово) числами в формате BCD и/или константами.</p> $\begin{array}{r} \boxed{Dd + 1} \quad \boxed{Dd} \text{ (BCD)} \\ \div \quad \boxed{Dr + 1} \quad \boxed{Dr} \text{ (BCD)} \\ \hline \boxed{R + 3} \quad \boxed{R + 2} \quad \boxed{R + 1} \quad \boxed{R} \text{ (BCD)} \\ \text{Остаток} \quad \text{Частное} \end{array}$	Выход Требуется
/BL(435)							
Dd							
Dr							
R							

### 3-10 Команды преобразования

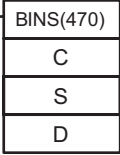

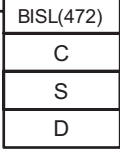
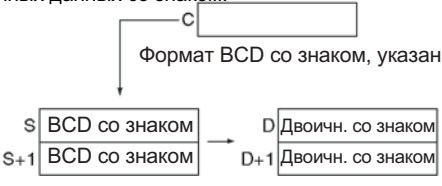
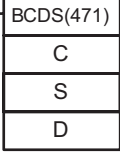
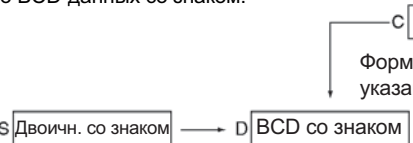
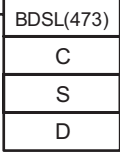
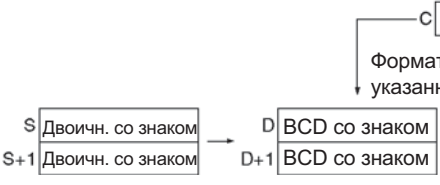
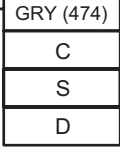
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения			
<b>BCD-TO-BINARY</b> BIN @BIN 023	<table border="1"> <tr><td>BIN(023)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: Входное слово R: Слово результата</p>	BIN(023)	S	R	<p>Преобразование значения в формате BCD в двоичное значение.</p> $s \boxed{\text{(BCD)}} \longrightarrow R \boxed{\text{(BIN)}}$	Выход Требуется
BIN(023)						
S						
R						
<b>DOUBLE BCD-TO-DOUBLE BINARY</b> BINL @BINL 058	<table border="1"> <tr><td>BINL(058)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 1-е входное слово R: 1-е слово результата</p>	BINL(058)	S	R	<p>Преобразование 8-разрядного значения в формате BCD в 8-разрядное 16-ричное (32-битовое двоичное) значение.</p> $\begin{array}{r} s \boxed{\text{(BCD)}} \\ S+1 \boxed{\text{(BCD)}} \end{array} \longrightarrow \begin{array}{r} R \boxed{\text{(BIN)}} \\ R+1 \boxed{\text{(BIN)}} \end{array}$	Выход Требуется
BINL(058)						
S						
R						

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>BINARY-TO-BCD</b> BCD @BCD 024	 <p>S: Входное слово R: Слово результата</p>	Преобразование слова двоичных данных в слово данных в формате BCD. $S \text{ (BIN)} \longrightarrow R \text{ (BCD)}$	Выход Требуется
<b>DOUBLE BINARY-TO-DOUBLE BCD</b> BCDL @BCDL 059	 <p>S: 1-е входное слово R: 1-е слово результата</p>	Преобразование 8-разрядного 16-ричного (32-битового двоичного) значения в 8-разрядное значение в формате BCD. $S \text{ (BIN)} \longrightarrow R \text{ (BCD)}$ $S+1 \text{ (BIN)} \longrightarrow R+1 \text{ (BCD)}$	Выход Требуется
<b>2'S COMPLEMENT</b> NEG @NEG 160	 <p>S: Входное слово R: Слово результата</p>	Операция "дополнение до 2-х" со словом 16-ричных данных. Дополнение до 2-х (дополнение + 1) $\overline{(S)} \longrightarrow (R)$	Выход Требуется
<b>DOUBLE 2'S COMPLEMENT</b> NEGL @NEGL 161	 <p>S: 1-е входное слово R: 1-е слово результата</p>	Операция "дополнение до 2-х" с двумя словами 16-ричных данных. Дополнение до 2-х (дополнение + 1) $\overline{(S+1, S)} \longrightarrow (R+1, R)$	Выход Требуется
<b>16-BIT TO 32-BIT SIGNED BINARY</b> SIGN @SIGN 600	 <p>S: Входное слово R: 1-е слово результата</p>	Преобразует 16-битовое двоичное значение со знаком в 32-битовое эквивалентное значение. 	Выход Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<p><b>DATA DECODER</b> MLPX @MLPX 076</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <p>MLPX(076)</p> <hr/> <p>S</p> <hr/> <p>C</p> <hr/> <p>R</p> </div> <p>S: Входное слово C: Управляющее слово R: 1-е слово результата</p>	<p>Чтение числового значения в указанном разряде (или байте) входного слова, установка соответствующего бита в слове результата (или в матрице из 16-ти слов) и сброс всех остальных битов в слове результата (или в матрице из 16-ти слов). Преобразование "4 → 16 битов"</p> <p>Преобразование "8 → 256 битов"</p> <p>Когда в L выбрано 2 байта, используется 2 матрицы из 16-ти слов.</p>	<p>Выход Требуется</p>

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения				
<b>DATA ENCODER</b> DMPX @DMPX 077	<table border="1" style="margin-left: 20px;"> <tr><td>DMPX(077)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> <tr><td>C</td></tr> </table> <p style="margin-left: 20px;">S: 1-е входное слово R: Слово результата C: Управляющее слово</p>	DMPX(077)	S	R	C	<p>Определение позиции первого или последнего включенного бита внутри входного слова (или матрицы из 16-ти слов) и запись номера позиции в указанный разряд (или байт) слова результата. Преобразование "16 → 4 бита"</p>  <p>Поиск старшего бита (Наибольший адрес бита)</p> <p>Поиск позиции крайнего левого бита (самом старшего бита) (Наивысший адрес бита)</p> <p>Декодирование "16 → 4 бита" (Позиция крайнего левого бита (m) записывается в R)</p> <p>Преобразование "256 → 8 битов"</p>  <p>L = 0 (Преобразование одной матрицы из 16-ти слов)</p> <p>Крайний левый бит</p> <p>Декодирование "256 → 8 битов" (Позиция крайнего левого бита в матрице из 16-ти слов (m) записывается в R)</p> <p>n = 1 (Начать с байта 1)</p>	<p>Выход Требуется</p>
DMPX(077)							
S							
R							
C							
<b>ASCII CONVERT</b> ASC @ASC 086	<table border="1" style="margin-left: 20px;"> <tr><td>ASC(086)</td></tr> <tr><td>S</td></tr> <tr><td>Di</td></tr> <tr><td>D</td></tr> </table> <p style="margin-left: 20px;">S: Входное слово Di: Признак разряда D: 1-е конечное слово</p>	ASC(086)	S	Di	D	<p>Преобразование 4-битовых 16-ричных разрядов входного слова в эквивалентные 8-битовые ASCII-строки.</p>  <p>1-й преобразуемый разряд</p> <p>Количество разрядов (n + 1)</p> <p>Левый (1) Правый (0)</p> <p>HEX → ASCII</p>	<p>Выход Требуется</p>
ASC(086)							
S							
Di							
D							

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения				
<b>ASCII TO HEX</b> HEX @HEX 162	<table border="1" style="width: 100%; text-align: center;"> <tr><td>HEX(162)</td></tr> <tr><td>S</td></tr> <tr><td>Di</td></tr> <tr><td>D</td></tr> </table> <p>S: 1-е входное слово                      Di: Обозначение разряда                      D: Конечное слово</p>	HEX(162)	S	Di	D	<p>Преобразование максимум 4-х байтов ASCII-данных входного слова в 16-ричные эквиваленты и запись этих разрядов в указанное конечное слово.</p>	Выход Требуется
HEX(162)							
S							
Di							
D							
<b>COLUMN TO LINE</b> LINE @LINE 063	<table border="1" style="width: 100%; text-align: center;"> <tr><td>LINE(063)</td></tr> <tr><td>S</td></tr> <tr><td>N</td></tr> <tr><td>D</td></tr> </table> <p>S: 1-е входное слово                      N: Номер бита                      D: Конечное слово</p>	LINE(063)	S	N	D	<p>Преобразование столбца битов матрицы из 16-ти слов (биты, находящиеся в одной и той же позиции в 16-ти следующих друг за другом словах) в 16 битов конечного слова.</p>	Выход Требуется
LINE(063)							
S							
N							
D							
<b>LINE TO COLUMN</b> COLM @COLM 064	<table border="1" style="width: 100%; text-align: center;"> <tr><td>COLM(064)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> <tr><td>N</td></tr> </table> <p>S: Входное слово                      D: 1-е конечное слово                      N: Номер бита</p>	COLM(064)	S	D	N	<p>Преобразование 16-ти битов входного слова в столбец битов матрицы из 16-ти конечных слов (биты, находящиеся в одной и той же позиции в 16-ти следующих друг за другом словах).</p>	Выход Требуется
COLM(064)							
S							
D							
N							

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>SIGNED BCD-TO-BINARY</b> BINS @BINS 470	 <p>C: Управляющее слово S: Входное слово D: Конечное слово</p>	Преобразование одного слова BCD-данных со знаком в одно слово двоичных данных со знаком. 	Выход Требуется
<b>DOUBLE SIGNED BCD-TO-BINARY</b> BISL @BISL 472	 <p>C: Управляющее слово S: 1-е входное слово D: 1-е конечное слово</p>	Преобразование двойного слова BCD-данных со знаком в двойное слово двоичных данных со знаком. 	Выход Требуется
<b>SIGNED BINARY-TO-BCD</b> BCDS @BCDS 471	 <p>C: Управляющее слово S: Входное слово D: Конечное слово</p>	Преобразование одного слова двоичных данных со знаком в одно слово BCD-данных со знаком. 	Выход Требуется
<b>DOUBLE SIGNED BINARY-TO-BCD</b> BDSL @BDSL 473	 <p>C: Управляющее слово S: 1-е входное слово D: 1-е конечное слово</p>	Преобразование двойного слова двоичных данных со знаком в двойное слово BCD-данных со знаком. 	Выход Требуется
<b>GRAY CODE CONVERSION</b> GRY 474  (Только модули серии CS/CJ версии 2.0 или выше, включая модули CS1-H, CJ1-H и CJ1M с номером партии 030201 и выше)	 <p>C: Управляющее слово S: Входное слово D: 1-е конечное слово</p>	Преобразование данных указанного слова, закодированных кодом Грэя, в двоичный формат, в BCD-формат или в значение угла (°) с указанным разрешением.	Выход Требуется

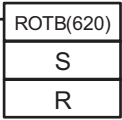
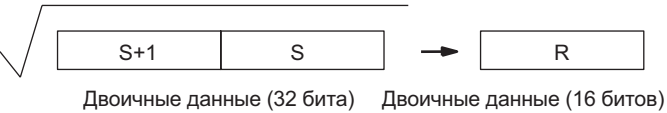
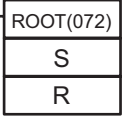
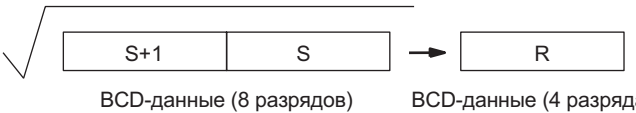
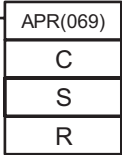
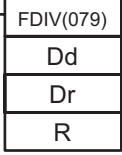
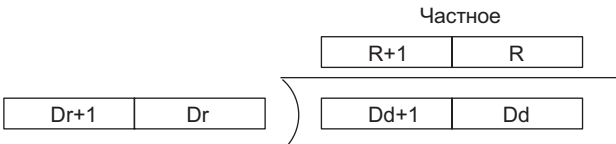
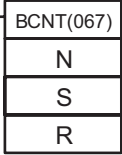
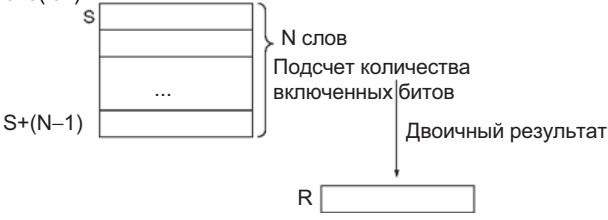


### 3-11 Логические команды

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения																			
<b>LOGICAL AND</b> ANDW @ANDW 034	<table border="1"> <tr><td>ANDW(034)</td></tr> <tr><td>I<sub>1</sub></td></tr> <tr><td>I<sub>2</sub></td></tr> <tr><td>R</td></tr> </table> <p>I<sub>1</sub>: Вход 1 I<sub>2</sub>: Вход 2 R: Слово результата</p>	ANDW(034)	I <sub>1</sub>	I <sub>2</sub>	R	<p>Операция "логическое И" над соответствующими битами одиночных слов данных и/или констант.</p> $I_1 \cdot I_2 \rightarrow R$ <table border="1"> <thead> <tr> <th>I<sub>1</sub></th> <th>I<sub>2</sub></th> <th>R</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	I <sub>1</sub>	I <sub>2</sub>	R	1	1	1	1	0	0	0	1	0	0	0	0	Выход Требуется
ANDW(034)																						
I <sub>1</sub>																						
I <sub>2</sub>																						
R																						
I <sub>1</sub>	I <sub>2</sub>	R																				
1	1	1																				
1	0	0																				
0	1	0																				
0	0	0																				
<b>DOUBLE LOGICAL AND</b> ANDL @ANDL 610	<table border="1"> <tr><td>ANDL(610)</td></tr> <tr><td>I<sub>1</sub></td></tr> <tr><td>I<sub>2</sub></td></tr> <tr><td>R</td></tr> </table> <p>I<sub>1</sub>: Вход 1 I<sub>2</sub>: Вход 2 R: Слово результата</p>	ANDL(610)	I <sub>1</sub>	I <sub>2</sub>	R	<p>Операция "логическое И" над соответствующими битами двойных слов данных и/или констант.</p> $(I_1 \cdot I_1 + 1) \cdot (I_2 \cdot I_2 + 1) \rightarrow (R, R + 1)$ <table border="1"> <thead> <tr> <th>I<sub>1</sub>·I<sub>1</sub>+1</th> <th>I<sub>2</sub>·I<sub>2</sub>+1</th> <th>R, R+1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	I <sub>1</sub> ·I <sub>1</sub> +1	I <sub>2</sub> ·I <sub>2</sub> +1	R, R+1	1	1	1	1	0	0	0	1	0	0	0	0	Выход Требуется
ANDL(610)																						
I <sub>1</sub>																						
I <sub>2</sub>																						
R																						
I <sub>1</sub> ·I <sub>1</sub> +1	I <sub>2</sub> ·I <sub>2</sub> +1	R, R+1																				
1	1	1																				
1	0	0																				
0	1	0																				
0	0	0																				
<b>LOGICAL OR</b> ORW @ORW 035	<table border="1"> <tr><td>ORW(035)</td></tr> <tr><td>I<sub>1</sub></td></tr> <tr><td>I<sub>2</sub></td></tr> <tr><td>R</td></tr> </table> <p>I<sub>1</sub>: Вход 1 I<sub>2</sub>: Вход 2 R: Слово результата</p>	ORW(035)	I <sub>1</sub>	I <sub>2</sub>	R	<p>Операция "логическое ИЛИ" над соответствующими битами одиночных слов данных и/или констант.</p> $I_1 + I_2 \rightarrow R$ <table border="1"> <thead> <tr> <th>I<sub>1</sub></th> <th>I<sub>2</sub></th> <th>R</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	I <sub>1</sub>	I <sub>2</sub>	R	1	1	1	1	0	1	0	1	1	0	0	0	Выход Требуется
ORW(035)																						
I <sub>1</sub>																						
I <sub>2</sub>																						
R																						
I <sub>1</sub>	I <sub>2</sub>	R																				
1	1	1																				
1	0	1																				
0	1	1																				
0	0	0																				
<b>DOUBLE LOGICAL OR</b> ORWL @ORWL 611	<table border="1"> <tr><td>ORWL(611)</td></tr> <tr><td>I<sub>1</sub></td></tr> <tr><td>I<sub>2</sub></td></tr> <tr><td>R</td></tr> </table> <p>I<sub>1</sub>: Вход 1 I<sub>2</sub>: Вход 2 R: Слово результата</p>	ORWL(611)	I <sub>1</sub>	I <sub>2</sub>	R	<p>Операция "логическое ИЛИ" над соответствующими битами двойных слов данных и/или констант.</p> $(I_1 \cdot I_1 + 1) + (I_2 \cdot I_2 + 1) \rightarrow (R, R + 1)$ <table border="1"> <thead> <tr> <th>I<sub>1</sub>·I<sub>1</sub>+1</th> <th>I<sub>2</sub>·I<sub>2</sub>+1</th> <th>R, R+1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	I <sub>1</sub> ·I <sub>1</sub> +1	I <sub>2</sub> ·I <sub>2</sub> +1	R, R+1	1	1	1	1	0	1	0	1	1	0	0	0	Выход Требуется
ORWL(611)																						
I <sub>1</sub>																						
I <sub>2</sub>																						
R																						
I <sub>1</sub> ·I <sub>1</sub> +1	I <sub>2</sub> ·I <sub>2</sub> +1	R, R+1																				
1	1	1																				
1	0	1																				
0	1	1																				
0	0	0																				
<b>EXCLUSIVE OR</b> XORW @XORW 036	<table border="1"> <tr><td>XORW(036)</td></tr> <tr><td>I<sub>1</sub></td></tr> <tr><td>I<sub>2</sub></td></tr> <tr><td>R</td></tr> </table> <p>I<sub>1</sub>: Вход 1 I<sub>2</sub>: Вход 2 R: Слово результата</p>	XORW(036)	I <sub>1</sub>	I <sub>2</sub>	R	<p>Операция "исключающее ИЛИ" над соответствующими битами одиночных слов данных и/или констант.</p> $I_1 \cdot \bar{I}_2 + \bar{I}_1 \cdot I_2 \rightarrow R$ <table border="1"> <thead> <tr> <th>I<sub>1</sub></th> <th>I<sub>2</sub></th> <th>R</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	I <sub>1</sub>	I <sub>2</sub>	R	1	1	0	1	0	1	0	1	1	0	0	0	Выход Требуется
XORW(036)																						
I <sub>1</sub>																						
I <sub>2</sub>																						
R																						
I <sub>1</sub>	I <sub>2</sub>	R																				
1	1	0																				
1	0	1																				
0	1	1																				
0	0	0																				


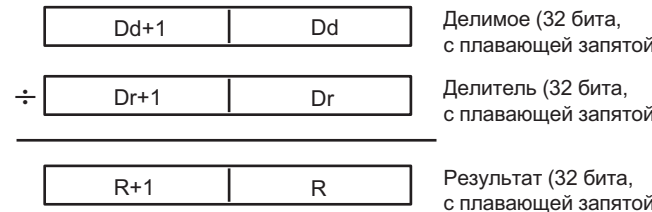
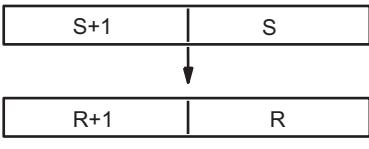
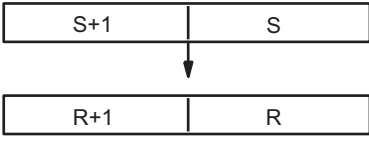
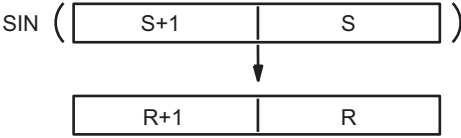
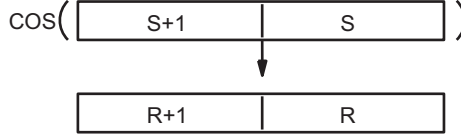
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения																			
<b>DOUBLE EXCLUSIVE OR</b> XORL @XORL 612	<table border="1"> <tr><td>XORL(612)</td></tr> <tr><td>I<sub>1</sub></td></tr> <tr><td>I<sub>2</sub></td></tr> <tr><td>R</td></tr> </table> <p>I<sub>1</sub>: Вход 1 I<sub>2</sub>: Вход 2 R: Слово результата</p>	XORL(612)	I <sub>1</sub>	I <sub>2</sub>	R	<p>Операция "исключающее ИЛИ" над соответствующими битами двойных слов данных и/или констант.</p> $(I_1 \cdot I_1 + 1) \cdot (\overline{I_2 \cdot I_2 + 1}) + (I_1 \cdot I_1 + 1) \cdot (I_2 \cdot I_2 + 1) \rightarrow (R, R + 1)$ <table border="1"> <thead> <tr> <th>I<sub>1</sub>.I<sub>1</sub>+1</th> <th>I<sub>2</sub>.I<sub>2</sub>+1</th> <th>R, R+1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	I <sub>1</sub> .I <sub>1</sub> +1	I <sub>2</sub> .I <sub>2</sub> +1	R, R+1	1	1	0	1	0	1	0	1	1	0	0	0	Выход Требуется
XORL(612)																						
I <sub>1</sub>																						
I <sub>2</sub>																						
R																						
I <sub>1</sub> .I <sub>1</sub> +1	I <sub>2</sub> .I <sub>2</sub> +1	R, R+1																				
1	1	0																				
1	0	1																				
0	1	1																				
0	0	0																				
<b>EXCLUSIVE NOR</b> XNRW @XNRW 037	<table border="1"> <tr><td>XNRW(037)</td></tr> <tr><td>I<sub>1</sub></td></tr> <tr><td>I<sub>2</sub></td></tr> <tr><td>R</td></tr> </table> <p>I<sub>1</sub>: Вход 1 I<sub>2</sub>: Вход 2 R: Слово результата</p>	XNRW(037)	I <sub>1</sub>	I <sub>2</sub>	R	<p>Операция "исключающее НЕ-ИЛИ" над соответствующими битами одиночных слов данных и/или констант.</p> $I_1 \cdot I_2 + \overline{I_1} \cdot \overline{I_2} \rightarrow R$ <table border="1"> <thead> <tr> <th>I<sub>1</sub></th> <th>I<sub>2</sub></th> <th>R</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> </tbody> </table>	I <sub>1</sub>	I <sub>2</sub>	R	1	1	1	1	0	0	0	1	0	0	0	1	Выход Требуется
XNRW(037)																						
I <sub>1</sub>																						
I <sub>2</sub>																						
R																						
I <sub>1</sub>	I <sub>2</sub>	R																				
1	1	1																				
1	0	0																				
0	1	0																				
0	0	1																				
<b>DOUBLE EXCLUSIVE NOR</b> XNRL @XNRL 613	<table border="1"> <tr><td>XNRL(613)</td></tr> <tr><td>I<sub>1</sub></td></tr> <tr><td>I<sub>2</sub></td></tr> <tr><td>R</td></tr> </table> <p>I<sub>1</sub>: Вход 1 I<sub>2</sub>: Вход 2 R: 1-е слово результата</p>	XNRL(613)	I <sub>1</sub>	I <sub>2</sub>	R	<p>Операция "исключающее НЕ-ИЛИ" над соответствующими битами двойных слов данных и/или констант.</p> $(I_1 \cdot I_1 + 1) \cdot (I_2 \cdot I_2 + 1) + (\overline{I_1 \cdot I_1 + 1}) \cdot (\overline{I_2 \cdot I_2 + 1}) \rightarrow (R, R + 1)$ <table border="1"> <thead> <tr> <th>I<sub>1</sub>.I<sub>1</sub>+1</th> <th>I<sub>2</sub>.I<sub>2</sub>+1</th> <th>R, R+1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> </tbody> </table>	I <sub>1</sub> .I <sub>1</sub> +1	I <sub>2</sub> .I <sub>2</sub> +1	R, R+1	1	1	1	1	0	0	0	1	0	0	0	1	Выход Требуется
XNRL(613)																						
I <sub>1</sub>																						
I <sub>2</sub>																						
R																						
I <sub>1</sub> .I <sub>1</sub> +1	I <sub>2</sub> .I <sub>2</sub> +1	R, R+1																				
1	1	1																				
1	0	0																				
0	1	0																				
0	0	1																				
<b>COMPLEMENT</b> COM @COM 029	<table border="1"> <tr><td>COM(029)</td></tr> <tr><td>Wd</td></tr> </table> <p>Wd: Слово</p>	COM(029)	Wd	<p>Сброс всех установленных битов и установка всех сброшенных битов в слове Wd.</p> $\overline{Wd} \rightarrow Wd: 1 \rightarrow 0 \text{ and } 0 \rightarrow 1$	Выход Требуется																	
COM(029)																						
Wd																						
<b>DOUBLE COMPLEMENT</b> COML @COML 614	<table border="1"> <tr><td>COML(614)</td></tr> <tr><td>Wd</td></tr> </table> <p>Wd: Слово</p>	COML(614)	Wd	<p>Сброс всех установленных битов и установка всех сброшенных битов в словах Wd и Wd+1.</p> $\overline{(Wd + 1, Wd)} \rightarrow (Wd + 1, Wd)$	Выход Требуется																	
COML(614)																						
Wd																						

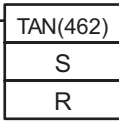
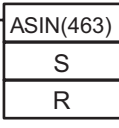
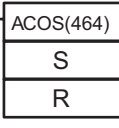
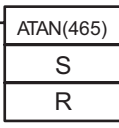
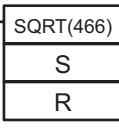
### 3-12 Специальные математические команды

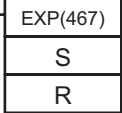
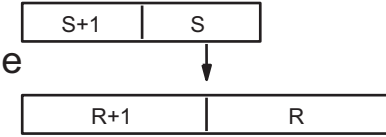
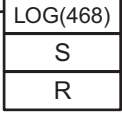
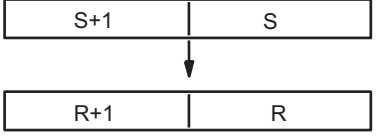
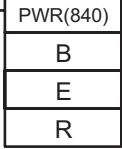
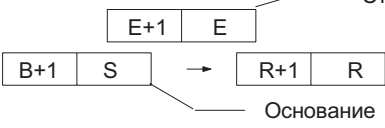

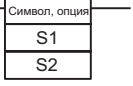

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>BINARY ROOT</b> ROTB @ROTB 620	 <p>S: 1-е входное слово R: Слово результата</p>	<p>Вычисление квадратного корня от 32-битового двоичного содержимого указанных слов и размещение целой части результата в указанное слово результата.</p> 	Выход Требуется
<b>BCD SQUARE ROOT</b> ROOT @ROOT 072	 <p>S: 1-е входное слово R: Слово результата</p>	<p>Вычисление квадратного корня от 8-разрядного BCD-числа и размещение целой части результата в указанное слово результата.</p> 	Выход Требуется
<b>ARITHMETIC PROCESS</b> APR @APR 069	 <p>C: Управляющее слово S: Входное слово R: Слово результата</p>	<p>Вычисление синуса, косинуса или линейной экстраполяции от входного значения. Функция линейной экстраполяции позволяет получить линейную аппроксимацию любой функции X (Y).</p>	Выход Требуется
<b>FLOATING POINT DIVIDE</b> FDIV @FDIV 079	 <p>Dd: 1-е слово делимого Dr: 1-е слово делителя R: 1-е слово результата</p>	<p>Деление одного 7-разрядного числа с плавающей запятой на другое. Числа с плавающей запятой представляются в экспоненциальном представлении (7-разрядная мантисса и 1 разряд показателя).</p> 	Выход Требуется
<b>BIT COUNTER</b> BCNT @BCNT 067	 <p>N: Количество слов S: 1-е входное слово R: Слово результата</p>	<p>Подсчитывается суммарное количество включенных битов в указанном слове(-ах).</p> 	Выход Требуется

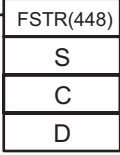
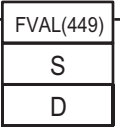
### 3-13 Команды математических операций с плавающей запятой

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>FLOATING TO 16-BIT</b>  FIX @FIX 450	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                 FIX(450)  <hr/>                 S  <hr/>                 R             </div> <p>S: 1-е входное слово R: Слово результата</p>	<p>Преобразование 32-битового значения с плавающей запятой в 16-битовое двоичное значение со знаком и запись результата в указанное слово результата.</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S+1</div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S</div> </div> <div style="text-align: center; margin-bottom: 10px;">↓</div> <div style="border: 1px solid black; padding: 5px; width: 60px; margin: 0 auto;">R</div> <p>Значение с плавающей запятой (32 бита) Двоичное число со знаком (16 битов)</p>	Выход Требуется
<b>FLOATING TO 32-BIT</b>  FIXL @FIXL 451	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                 FIXL(451)  <hr/>                 S  <hr/>                 R             </div> <p>S: 1-е входное слово R: 1-е слово результата</p>	<p>Преобразование 32-битового значения с плавающей запятой в 32-битовое двоичное значение со знаком и запись результата в указанные слова результата.</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S+1</div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S</div> </div> <div style="text-align: center; margin-bottom: 10px;">↓</div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">R+1</div> <div style="border: 1px solid black; padding: 5px;">R</div> </div> <p>Значение с плавающей запятой (32 бита) Двоичное число со знаком (32 бита)</p>	Выход Требуется
<b>16-BIT TO FLOATING</b>  FLT @FLT 452	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                 FLT(452)  <hr/>                 S  <hr/>                 R             </div> <p>S: Входное слово R: 1-е слово результата</p>	<p>Преобразование 16-битового двоичного значения со знаком в 32-битовое значение с плавающей запятой и запись результата в указанные слова результата.</p> <div style="text-align: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; width: 60px;">S</div> <div style="margin: 5px 0;">↓</div> </div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">R+1</div> <div style="border: 1px solid black; padding: 5px;">R</div> </div> <p>Двоичное число со знаком (16 битов) Значение с плавающей запятой (32 бита)</p>	Выход Требуется
<b>32-BIT TO FLOATING</b>  FLTL @FLTL 453	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                 FLTL(453)  <hr/>                 S  <hr/>                 R             </div> <p>S: 1-е входное слово R: 1-е слово результата</p>	<p>Преобразование 32-битового двоичного значения со знаком в 32-битовое значение с плавающей запятой и запись результата в указанные слова результата.</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S+1</div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">S</div> </div> <div style="text-align: center; margin-bottom: 10px;">↓</div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">R+1</div> <div style="border: 1px solid black; padding: 5px;">R</div> </div> <p>Двоичное число со знаком (32 бита) Значение с плавающей запятой (32 бита)</p>	Выход Требуется
<b>FLOATING-POINT ADD</b>  +F @+F 454	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                 +F(454)  <hr/>                 Au  <hr/>                 Ad  <hr/>                 R             </div> <p>Au: 1-е слово 1-го слагаемого AD: 1-е слово 2-го слагаемого R: 1-е слово результата</p>	<p>Сложение двух 32-битовых чисел с плавающей запятой и запись результата в указанные слова результата.</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Au+1</div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Au</div> </div> <div style="margin: 0 10px;">+</div> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Ad+1</div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Ad</div> </div> <hr style="width: 100%; border: 0.5px solid black;"/> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">R+1</div> <div style="border: 1px solid black; padding: 5px;">R</div> </div> <p>1-е слагаемое (32 бита, с плавающей запятой) 2-е слагаемое (32 бита, с плавающей запятой) Результат (32 бита, с плавающей запятой)</p>	Выход Требуется
<b>FLOATING-POINT SUBTRACT</b>  -F @-F 455	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                 -F(455)  <hr/>                 Mi  <hr/>                 Su  <hr/>                 R             </div> <p>Mi: 1-е слово уменьшаемого Su: 1-е слово вычитаемого R: 1-е слово результата</p>	<p>Вычитание одного 32-битового числа с плавающей запятой из другого и запись результата в указанные слова результата.</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Mi+1</div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Mi</div> </div> <div style="margin: 0 10px;">-</div> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Su+1</div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Su</div> </div> <hr style="width: 100%; border: 0.5px solid black;"/> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">R+1</div> <div style="border: 1px solid black; padding: 5px;">R</div> </div> <p>Уменьшаемое (32 бита, с плавающей запятой) Вычитаемое (32 бита, с плавающей запятой) Результат (32 бита, с плавающей запятой)</p>	Выход Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>FLOATING-POINT MULTIPLY</b> *F @*F 456	*F(456) Md Mr R Md: 1-е слово уменьшаемого Mr: 1-е слово множителя R: 1-е слово результата	Умножение двух 32-битовых чисел с плавающей запятой и запись результата в указанные слова результата.  Умножаемое (32 бита, с плавающей запятой) Множитель (32 бита, с плавающей запятой) Результат (32 бита, с плавающей запятой)	Выход Требуется
<b>FLOATING-POINT DIVIDE</b> /F @/F 457	/F(457) Dd Dr R Dd: 1-е слово делимого Dr: 1-е слово делителя R: 1-е слово результата	Деление одного 32-битового числа с плавающей запятой на другое и запись результата в указанные слова результата.  Делимое (32 бита, с плавающей запятой) Делитель (32 бита, с плавающей запятой) Результат (32 бита, с плавающей запятой)	Выход Требуется
<b>DEGREES TO RADIANS</b> RAD @RAD 458	RAD(458) S R S: 1-е входное слово R: 1-е слово результата	Преобразование 32-битового числа с плавающей запятой из градусов в радианы и запись результата в указанные слова результата.  Входное слово (градусы, 32 бита, с плавающей запятой) Результат (радианы, 32 бита, с плавающей запятой)	Выход Требуется
<b>RADIANS TO DEGREES</b> DEG @DEG 459	DEG(459) S R S: 1-е входное слово R: 1-е слово результата	Преобразование 32-битового числа с плавающей запятой из радианов в градусы и запись результата в указанные слова результата.  Входное слово (радианы, 32 бита, с плавающей запятой) Результат (градусы, 32 бита, с плавающей запятой)	Выход Требуется
<b>SINE</b> SIN @SIN 460	SIN(460) S R S: 1-е входное слово R: 1-е слово результата	Расчет синуса от 32-битового числа с плавающей запятой (в радианах) и запись результата в указанные слова результата.  Входное слово (32 бита, с плавающей запятой) Результат (32 бита, с плавающей запятой)	Выход Требуется
<b>COSINE</b> COS @COS 461	COS(461) S R S: 1-е входное слово R: 1-е слово результата	Расчет косинуса от 32-битового числа с плавающей запятой (в радианах) и запись результата в указанные слова результата.  Входное слово (32 бита, с плавающей запятой) Результат (32 бита, с плавающей запятой)	Выход Требуется

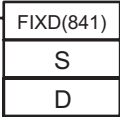
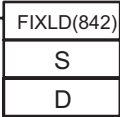
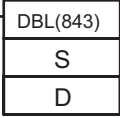
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>TANGENT</b> TAN @TAN 462	 <p>S: 1-е входное слово R: 1-е слово результата</p>	<p>Расчет тангенса от 32-битового числа с плавающей запятой (в радианах) и запись результата в указанные слова результата.</p> $\text{TAN} \left( \begin{array}{ c c } \hline S+1 & S \\ \hline \end{array} \right)$ <p style="text-align: center;">↓</p> $\begin{array}{ c c } \hline R+1 & R \\ \hline \end{array}$ <p>Входное слово (32 бита, с плавающей запятой) Результат (32 бита, с плавающей запятой)</p>	Выход Требуется
<b>ARC SINE</b> ASIN @ASIN 463	 <p>S: 1-е входное слово R: 1-е слово результата</p>	<p>Расчет арксинуса от 32-битового числа с плавающей запятой (в радианах) и запись результата в указанные слова результата (функция взятия арксинуса обратна функции синуса; она возвращает значение угла, для которого получается указанное значение синуса в пределах -1 ... 1).</p> $\text{SIN}^{-1} \left( \begin{array}{ c c } \hline S+1 & S \\ \hline \end{array} \right)$ <p style="text-align: center;">↓</p> $\begin{array}{ c c } \hline R+1 & R \\ \hline \end{array}$ <p>Входное слово (32 бита, с плавающей запятой) Результат (32 бита, с плавающей запятой)</p>	Выход Требуется
<b>ARC COSINE</b> ACOS @ACOS 464	 <p>S: 1-е входное слово R: 1-е слово результата</p>	<p>Расчет арккосинуса от 32-битового числа с плавающей запятой (в радианах) и запись результата в указанные слова результата (функция взятия арккосинуса обратна функции косинуса; она возвращает значение угла, для которого получается указанное значение косинуса в пределах -1 ... 1).</p> $\text{COS}^{-1} \left( \begin{array}{ c c } \hline S+1 & S \\ \hline \end{array} \right)$ <p style="text-align: center;">↓</p> $\begin{array}{ c c } \hline R+1 & R \\ \hline \end{array}$ <p>Входное слово (32 бита, с плавающей запятой) Результат (32 бита, с плавающей запятой)</p>	Выход Требуется
<b>ARC TANGENT</b> ATAN @ATAN 465	 <p>S: 1-е входное слово R: 1-е слово результата</p>	<p>Расчет арктангенса от 32-битового числа с плавающей запятой (в радианах) и запись результата в указанные слова результата (функция взятия арктангенса обратна функции тангенса; она возвращает значение угла, для которого получается указанное значение тангенса).</p> $\text{TAN}^{-1} \left( \begin{array}{ c c } \hline S+1 & S \\ \hline \end{array} \right)$ <p style="text-align: center;">↓</p> $\begin{array}{ c c } \hline R+1 & R \\ \hline \end{array}$ <p>Входное слово (32 бита, с плавающей запятой) Результат (32 бита, с плавающей запятой)</p>	Выход Требуется
<b>SQUARE ROOT</b> SQRT @SQRT 466	 <p>S: 1-е входное слово R: 1-е слово результата</p>	<p>Расчет квадратного корня от 32-битового числа с плавающей запятой и запись результата в указанные слова результата.</p> $\sqrt{\begin{array}{ c c } \hline S+1 & S \\ \hline \end{array}}$ <p style="text-align: center;">↓</p> $\begin{array}{ c c } \hline R+1 & R \\ \hline \end{array}$ <p>Входное слово (32 бита, с плавающей запятой) Результат (32 бита, с плавающей запятой)</p>	Выход Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>EXPONENT</b> EXP @EXP 467	 <p>S: 1-е входное слово R: 1-е слово результата</p>	<p>Расчет натурального экспоненциала (по основанию e) от 32-битового числа с плавающей запятой и запись результата в указанные слова результата.</p>  <p>Входное слово (32 бита, с плавающей запятой) Результат (32 бита, с плавающей запятой)</p>	<p>Выход Требуется</p>
<b>LOGARITHM</b> LOG @LOG 468	 <p>S: 1-е входное слово R: 1-е слово результата</p>	<p>Расчет натурального логарифма (по основанию e) от 32-битового числа с плавающей запятой и запись результата в указанные слова результата.</p>  <p>Входное слово (32 бита, с плавающей запятой) Результат (32 бита, с плавающей запятой)</p>	<p>Выход Требуется</p>
<b>EXPONENTIAL POWER</b> PWR @PWR 840	 <p>B: 1-е слово основания E: 1-е слово показателя R: 1-е слово результата</p>	<p>Возведение 32-битового числа с плавающей запятой в степень, определяемую другим 32-битовым числом с плавающей запятой.</p> 	<p>Выход Требуется</p>
<b>FLOATING SYMBOL COMPARISON</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) LD, AND или OR + =F (329), <>F (330), <F (331), <=F (332), >F (333), или >=F (334)	<p>С помощью LD:</p>  <p>С помощью AND:</p>  <p>С помощью OR:</p>  <p>S1: Данные для сравнения 1 S2: Данные для сравнения 2</p>	<p>Сравнение указанных данных одинарной точности (32 бита) или констант и установление (ВКЛ) условия выполнения в случае положительного результата сравнения.</p> <p>Для команд сравнения чисел с плавающей запятой можно использовать три типа символов: LD (нагрузка), AND и OR.</p>	<p>LD: не требуется AND или OR: требуется</p>

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>FLOATING-POINT TO ASCII</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) FSTR @FSTR 448	 <p>S: 1-е входное слово C: Управляющее слово D: Конечное слово</p>	Преобразование указанного слова с плавающей запятой одинарной точности (32-битовое число с десятичной точкой или в экспоненциальном формате) в текстовую строку (ASCII) и размещение результата в конечное слово.	Выход Требуется
<b>ASCII TO FLOATING-POINT</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) FVAL @FVAL 449	 <p>S: Входное слово D: 1-е конечное слово</p>	Преобразование текстовой строки (ASCII), представляющей число с плавающей запятой одинарной точности (с десятичной точкой или в экспоненциальном формате), в 32-битовое число с плавающей запятой одинарной точности и размещение результата в конечные слова.	Выход Требуется

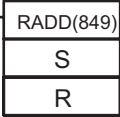
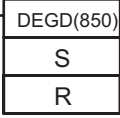
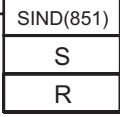
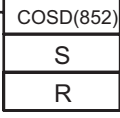
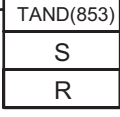
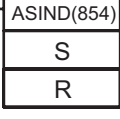
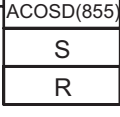
### 3-14 Команды математических операций с плавающей запятой, двойной точности

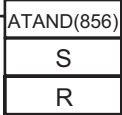
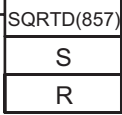
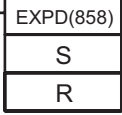
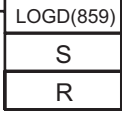
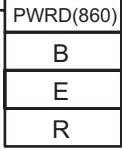



Команды для чисел с плавающей запятой двойной точности поддерживаются только модулями CPU CS1-H, CJ1-H, CJ1M или CS1D.

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>DOUBLE FLOATING TO 16-BIT BINARY</b> FIXD @FIXD 841	 <p>S: 1-е входное слово D: Конечное слово</p>	Преобразование указанного значения с плавающей запятой, двойной точности (64 бита) в 16-битовое двоичное значение со знаком и запись результата в конечное слово.	Выход Требуется
<b>DOUBLE FLOATING TO 32-BIT BINARY</b> FIXLD @FIXLD 842	 <p>S: 1-е входное слово D: 1-е конечное слово</p>	Преобразование указанного значения с плавающей запятой, двойной точности (64 бита) в 32-битовое двоичное значение со знаком и запись результата в конечные слова.	Выход Требуется
<b>16-BIT BINARY TO DOUBLE FLOATING</b> DBL @DBL 843	 <p>S: Входное слово D: 1-е конечное слово</p>	Преобразование указанного 16-битового двоичного значения со знаком в число с плавающей запятой, двойной точности (64 бита) и запись результата в конечные слова.	Выход Требуется

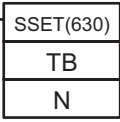
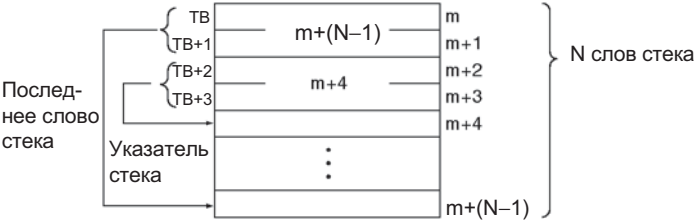
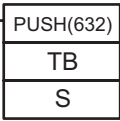
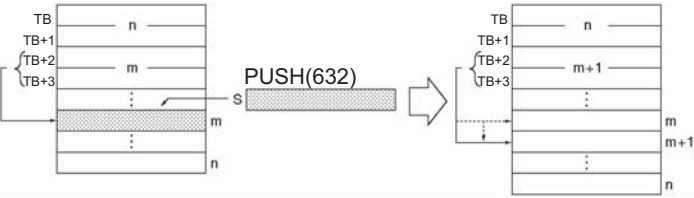
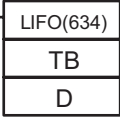
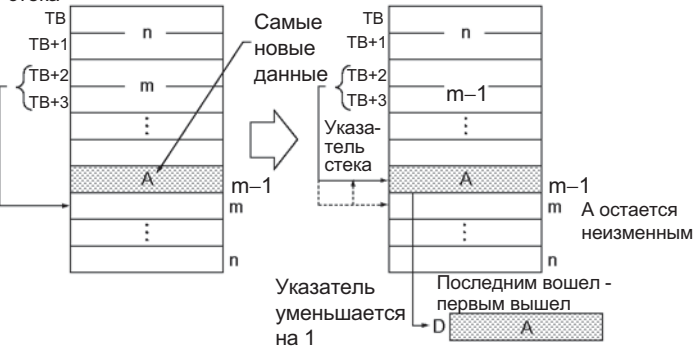
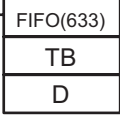
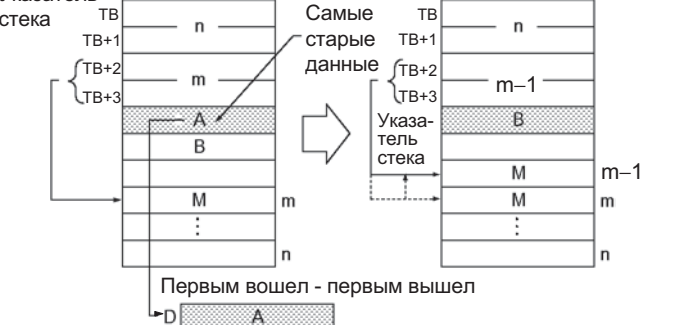


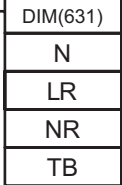
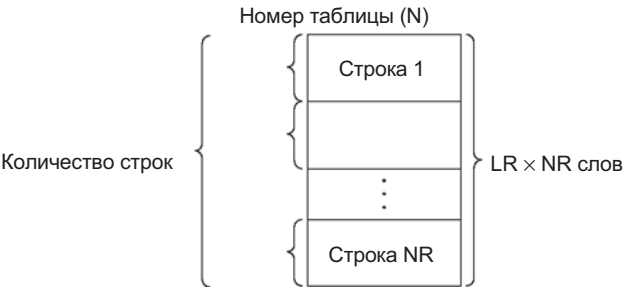
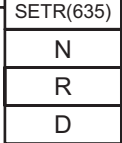
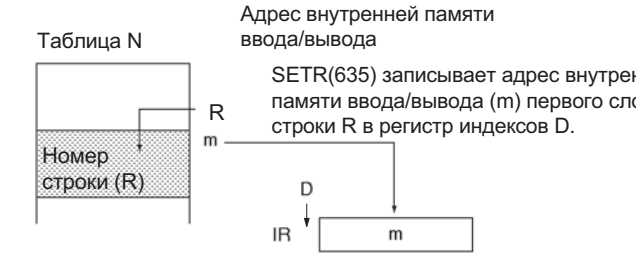
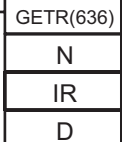
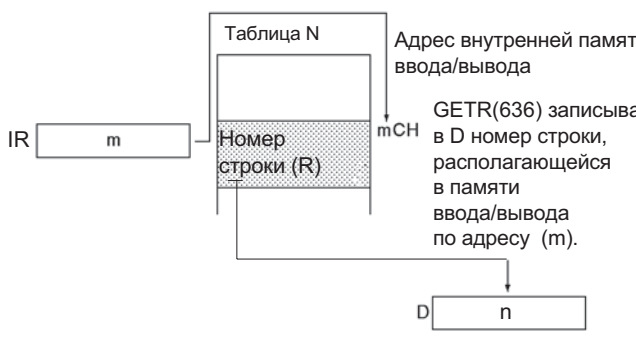
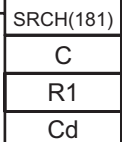
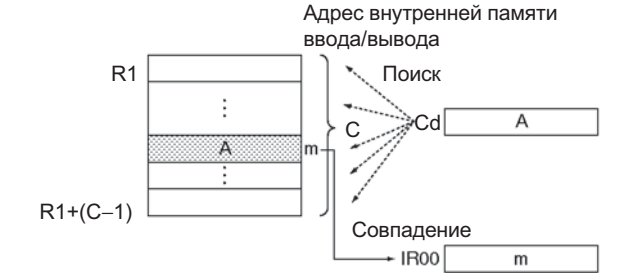
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения				
<b>32-BIT BINARY TO DOUBLE FLOATING</b> DBLL @DBLL 844	<table border="1"> <tr><td>DBLL(844)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: 1-е входное слово D: 1-е конечное слово</p>	DBLL(844)	S	D	Преобразование указанного 32-битового двоичного значения со знаком в число с плавающей запятой, двойной точности (64 бита) и запись результата в конечные слова.	Выход Требуется	
DBLL(844)							
S							
D							
<b>DOUBLE FLOATING-POINT ADD</b> +D @+D 845	<table border="1"> <tr><td>+D(845)</td></tr> <tr><td>Au</td></tr> <tr><td>Ad</td></tr> <tr><td>R</td></tr> </table> <p>Au: 1-е слово 1-го слагаемого Ad: 1-е слово 2-го слагаемого R: 1-е слово результата</p>	+D(845)	Au	Ad	R	Сложение указанных чисел с плавающей запятой, двойной точности (каждое по 64 бита) и запись результата в слова результата.	Выход Требуется
+D(845)							
Au							
Ad							
R							
<b>DOUBLE FLOATING-POINT SUBTRACT</b> -D @-D 846	<table border="1"> <tr><td>-D(846)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi: 1-е слово уменьшаемого Su: 1-е слово вычитаемого R: 1-е слово результата</p>	-D(846)	Mi	Su	R	Операция вычитания над указанными числами с плавающей запятой, двойной точности (каждое по 64 бита) и запись результата в слова результата.	Выход Требуется
-D(846)							
Mi							
Su							
R							
<b>DOUBLE FLOATING-POINT MULTIPLY</b> *D @*D 847	<table border="1"> <tr><td>*D(847)</td></tr> <tr><td>Md</td></tr> <tr><td>Mr</td></tr> <tr><td>R</td></tr> </table> <p>Md: 1-е слово умножаемого Mr: 1-е слово множителя R: 1-е слово результата</p>	*D(847)	Md	Mr	R	Умножение указанных чисел с плавающей запятой, двойной точности (каждое по 64 бита) и запись результата в слова результата.	Выход Требуется
*D(847)							
Md							
Mr							
R							
<b>DOUBLE FLOATING-POINT DIVIDE</b> /D @/D 848	<table border="1"> <tr><td>/D(848)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: 1-е слово делимого Dr: 1-е слово делителя R: 1-е слово результата</p>	/D(848)	Dd	Dr	R	Операция деления над указанными числами с плавающей запятой, двойной точности (каждое по 64 бита) и запись результата в слова результата.	Выход Требуется
/D(848)							
Dd							
Dr							
R							

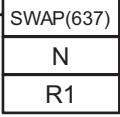
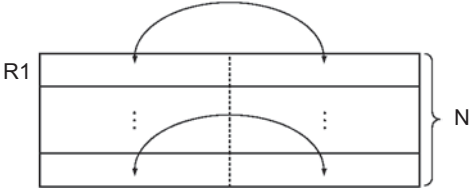
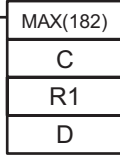
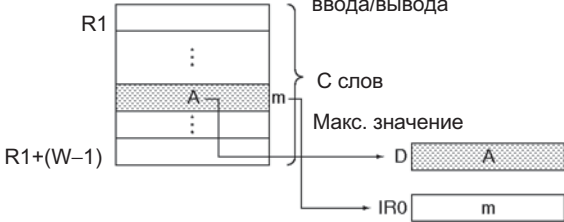
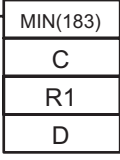
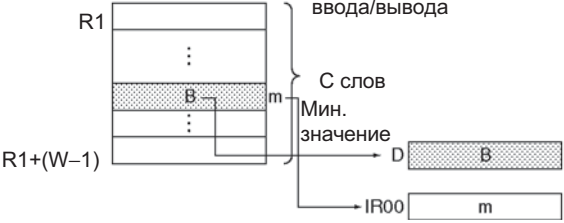
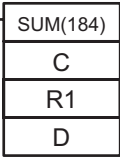
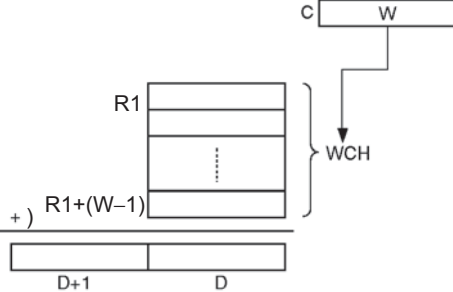
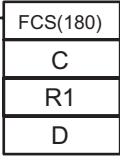
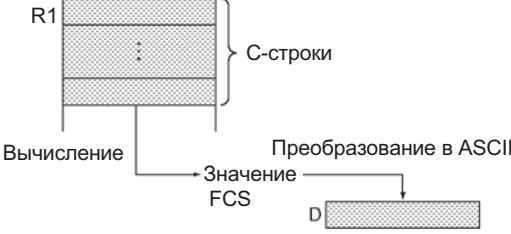
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>DOUBLE DEGREES TO RADIANS</b> RADD @RADD 849	 <p>S: 1-е входное слово R: 1-е слово результата</p>	Преобразование указанного числа с плавающей запятой, двойной точности (64 бита) из градусов в радианы и запись результата в слова результата.	Выход Требуется
<b>DOUBLE RADIANS TO DEGREES</b> DEGD @DEGD 850	 <p>S: 1-е входное слово R: 1-е слово результата</p>	Преобразование указанного числа с плавающей запятой, двойной точности (64 бита) из радианов в градусы и запись результата в слова результата.	Выход Требуется
<b>DOUBLE SINE</b> SIND @SIND 851	 <p>S: 1-е входное слово R: 1-е слово результата</p>	Вычисление синуса угла (в радианах), заданного числом с плавающей запятой, двойной точности (64 бита), и запись результата в слова результата.	Выход Требуется
<b>DOUBLE COSINE</b> COSD @COSD 852	 <p>S: 1-е входное слово R: 1-е слово результата</p>	Вычисление косинуса угла (в радианах), заданного числом с плавающей запятой, двойной точности (64 бита), и запись результата в слова результата.	Выход Требуется
<b>DOUBLE TANGENT</b> TAND @TAND 853	 <p>S: 1-е входное слово R: 1-е слово результата</p>	Вычисление тангенса угла (в радианах), заданного числом с плавающей запятой, двойной точности (64 бита), и запись результата в слова результата.	Выход Требуется
<b>DOUBLE ARC SINE</b> ASIND @ASIND 854	 <p>S: 1-е входное слово R: 1-е слово результата</p>	Вычисление угла (в радианах) по значению синуса, указанному числом с плавающей запятой, двойной точности (64 бита), и запись результата в слова результата (функция арксинуса является обратной по отношению к функции синуса; она возвращает значение угла, для которого получается указанное значение синуса в пределах $-1 \dots 1$ ).	Выход Требуется
<b>DOUBLE ARC COSINE</b> ACOSD @ACOSD 855	 <p>S: 1-е входное слово R: 1-е слово результата</p>	Вычисление угла (в радианах) по значению косинуса, указанному числом с плавающей запятой, двойной точности (64 бита), и запись результата в слова результата (функция арккосинуса является обратной по отношению к функции косинуса; она возвращает значение угла, для которого получается указанное значение косинуса в пределах $-1 \dots 1$ ).	Выход Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>DOUBLE ARC TANGENT</b> ATAND @ATAND 856	 <p>S: 1-е входное слово R: 1-е слово результата</p>	Вычисление угла (в радианах) по значению тангенса, указанному числом с плавающей запятой, двойной точности (64 бита), и запись результата в слова результата (функция арктангенса является обратной по отношению к функции тангенса; она возвращает значение угла, для которого получается указанное значение тангенса).	Выход Требуется
<b>DOUBLE SQUARE ROOT</b> SQRTD @SQRTD 857	 <p>S: 1-е входное слово R: 1-е слово результата</p>	Вычисление квадратного корня от указанного числа с плавающей запятой, двойной точности (64 бита) и запись результата в слова результата.	Выход Требуется
<b>DOUBLE EXPONENT</b> EXPD @EXPD 858	 <p>S: 1-е входное слово R: 1-е слово результата</p>	Вычисление натурального экспоненциала (основание = "e") от указанного числа с плавающей запятой, двойной точности (64 бита) и запись результата в слова результата.	Выход Требуется
<b>DOUBLE LOGARITHM</b> LOGD @LOGD 859	 <p>S: 1-е входное слово R: 1-е слово результата</p>	Вычисление натурального логарифма (основание = "e") для указанного числа с плавающей запятой, двойной точности (64 бита) и запись результата в слова результата.	Выход Требуется
<b>DOUBLE EXPONENTIAL POWER</b> PWRD @PWRD 860	 <p>B: 1-е слово основания E: 1-е слово показателя R: 1-е слово результата</p>	Возведение числа с плавающей запятой, двойной точности (64 бита) в степень, значение которой определяется другим числом с плавающей запятой двойной точности, и запись результата в слова результата.	Выход Требуется
<b>DOUBLE SYMBOL COMPARISON</b> LD, AND или OR + =D (335), <>D (336), <D (337), <=D (338), >D (339), или >=D (340)	<p>С помощью LD:</p>  <p>С помощью AND:</p>  <p>С помощью OR:</p>  <p>S1: Данные для сравнения 1 S2: Данные для сравнения 2</p>	<p>Сравнение указанных чисел двойной точности (64 бита) и установка (ВКЛ) условия выполнения в случае положительного результата сравнения.</p> <p>Для команд сравнения чисел с плавающей запятой можно использовать три типа символов: LD (нагрузка), AND и OR.</p>	LD: не требуется AND или OR: требуется

### 3-15 Команды обработки табличных данных

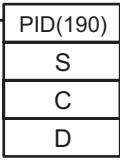
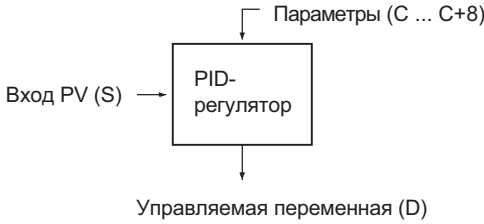
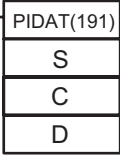
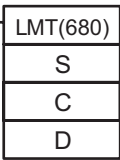
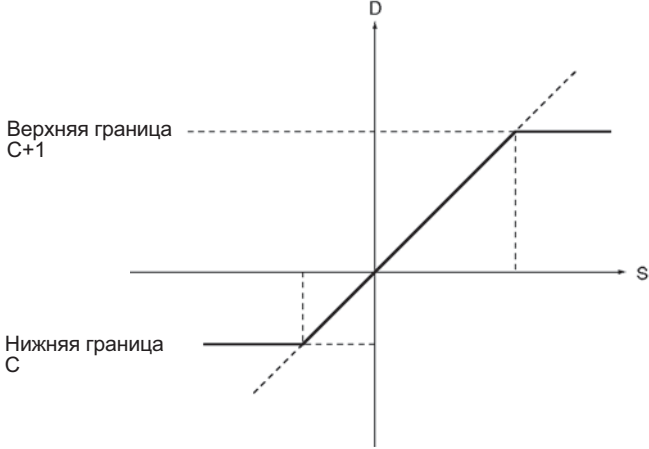
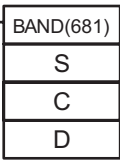
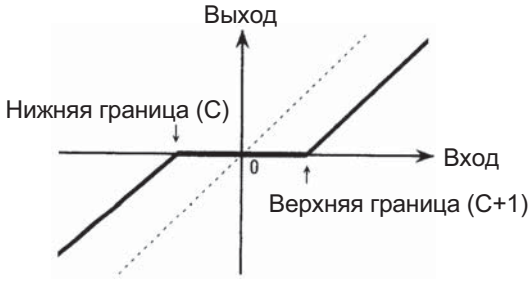
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>SET STACK</b> SSET @SSET 630	 <p>TB: Начальный адрес стека N: Количество слов</p>	<p>Создает стек указанной длины, начинающийся с указанного слова, и инициализирует слова, принадлежащие стеку, записывая в них нули.</p> <p>Адрес внутренней памяти ввода/вывода</p>  <p>Последнее слово стека</p> <p>Указатель стека</p> <p>N слов стека</p>	Выход Требуется
<b>PUSH ONTO STACK</b> PUSH @PUSH 632	 <p>TB: Начальный адрес стека S: Входное слово</p>	<p>Запись одного слова данных в указанный стек.</p> <p>Адрес внутренней памяти ввода/вывода</p> <p>Адрес внутренней памяти ввода/вывода</p>  <p>TB</p> <p>ТВ+1</p> <p>ТВ+2</p> <p>ТВ+3</p> <p>...</p> <p>n</p> <p>m</p> <p>m+1</p> <p>m+2</p> <p>m+3</p> <p>m+4</p> <p>...</p> <p>m+(N-1)</p>	Выход Требуется
<b>LAST IN FIRST OUT</b> LIFO @LIFO 634	 <p>TB: Начальный адрес стека D: Конечное слово</p>	<p>Чтение последнего слова данных, записанного в указанный стек (самые последние данные в стеке).</p> <p>Указатель стека</p> <p>Адрес внутренней памяти ввода/вывода</p> <p>Адрес внутренней памяти ввода/вывода</p>  <p>ТВ</p> <p>ТВ+1</p> <p>ТВ+2</p> <p>ТВ+3</p> <p>...</p> <p>n</p> <p>m</p> <p>m-1</p> <p>...</p> <p>m-1</p> <p>m</p> <p>n</p> <p>Самые новые данные</p> <p>Указатель стека</p> <p>Указатель стека уменьшается на 1</p> <p>Последним вошел - первым вышел</p> <p>A остается неизменным.</p>	Выход Требуется
<b>FIRST IN FIRST OUT</b> FIFO @FIFO 633	 <p>TB: Начальный адрес стека D: Конечное слово</p>	<p>Чтение первого слова данных, записанного в указанный стек (самые старые данные стека).</p> <p>Указатель стека</p> <p>Адрес внутренней памяти ввода/вывода</p> <p>Адрес внутренней памяти ввода/вывода</p>  <p>ТВ</p> <p>ТВ+1</p> <p>ТВ+2</p> <p>ТВ+3</p> <p>...</p> <p>n</p> <p>m</p> <p>m-1</p> <p>...</p> <p>m-1</p> <p>m</p> <p>n</p> <p>Самые старые данные</p> <p>Указатель стека</p> <p>Первым вошел - первым вышел</p>	Выход Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>DIMENSION RECORD TABLE</b> DIM @DIM 631	 <p>N: Номер таб-цы                      LR: Длина каждой строки                      NR: Кол-во строк                      TB: 1-е слово таблицы</p>	<p>Создает таблицу, состоящую из строк, объявляет длину каждой строки и количество строк. Может быть создано до 16-ти таблиц строк.</p> 	Выход Требуется
<b>SET RECORD LOCATION</b> SETR @SETR 635	 <p>N: Номер таб-цы                      R: Номер строки                      D: Адресуемый регистр индексов</p>	<p>Записывает адрес расположения указанной строки (адрес внутренней памяти ввода/вывода начала строки) в указанный регистр индексов.</p> 	Выход Требуется
<b>GET RECORD NUMBER</b> GETR @GETR 636	 <p>N: Номер таб-цы                      IR: Регистр индексов                      D: Адресуемое слово</p>	<p>Возвращает номер строки, расположенной по адресу внутренней памяти ввода/вывода, содержащемуся в указанном регистре индексов.</p> 	Выход Требуется
<b>DATA SEARCH</b> SRCH @SRCH 181	 <p>C: 1-е управляющее слово                      R1: 1-е слово массива                      Cd: Сравниваемые данные</p>	<p>Поиск слова данных в пределах массива слов.</p> 	Выход Требуется



Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>SWAP BYTES</b> SWAP @SWAP 637	 <p>N: Количество слов R1: 1-е слово в массиве</p>	Переставляет местами левый и правый байты во всех словах массива. Байты меняются местами. 	Выход Требуется
<b>FIND MAXIMUM</b> MAX @MAX 182	 <p>C: 1-е управляющее слово R1: 1-е слово в массиве D: Адресуемое слово</p>	Поиск максимального значения в массиве. Адрес внутренней памяти ввода/вывода 	Выход Требуется
<b>FIND MINIMUM</b> MIN @MIN 183	 <p>C: 1-е управляющее слово R1: 1-е слово в массиве D: Адресуемое слово</p>	Поиск минимального значения в массиве. Адрес внутренней памяти ввода/вывода 	Выход Требуется
<b>SUM</b> SUM @SUM 184	 <p>C: 1-е управляющее слово R1: 1-е слово в массиве D: Адресуемое слово</p>	Сложение байтов или слов массива и запись результата в два слова. 	Выход Требуется
<b>FRAME CHECK-SUM</b> FCS @FCS 180	 <p>C: 1-е управляющее слово R1: 1-е слово в массиве D: 1-е адресуемое слово</p>	Вычисление контрольной суммы кадра (FCS) в ASCII формате для указанного массива. 	Выход Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения				
<b>STACK SIZE READ</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) SNUM @SNUM 638	<table border="1"> <tr><td>SNUM(638)</td></tr> <tr><td>TB</td></tr> <tr><td>D</td></tr> </table> <p>TB: Начальный адрес стека D: Адресуемое слово</p>	SNUM(638)	TB	D	Рассчитывает объем данных в указанном стеке (количество слов).	Выход Требуется	
SNUM(638)							
TB							
D							
<b>STACK DATA READ</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) SREAD @SREAD 639	<table border="1"> <tr><td>SREAD(639)</td></tr> <tr><td>TB</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> <p>TB: Начальный адрес стека C: Значение смещения D: Адресуемое слово</p>	SREAD(639)	TB	C	D	Чтение данных из указанного элемента данных стека. Величина смещения определяет позицию требуемого элемента данных в стеке (количество элементов данных, расположенных до текущей позиции указателя).	Выход Требуется
SREAD(639)							
TB							
C							
D							
<b>STACK DATA OVERWRITE</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) SWRIT @SWRIT 640	<table border="1"> <tr><td>SWRIT(640)</td></tr> <tr><td>TB</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> </table> <p>TB: Начальный адрес стека C: Значение смещения S: Входные данные</p>	SWRIT(640)	TB	C	S	Запись входных данных в указанный элемент данных стека (перезапись существующих данных). Величина смещения определяет позицию требуемого элемента данных в стеке (количество элементов данных, расположенных до текущей позиции указателя).	Выход Требуется
SWRIT(640)							
TB							
C							
S							
<b>STACK DATA INSERT</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) SINS @SINS 641	<table border="1"> <tr><td>SINS(641)</td></tr> <tr><td>TB</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> </table> <p>TB: Начальный адрес стека C: Значение смещения S: Входные данные</p>	SINS(641)	TB	C	S	Входные данные вставляются в указанную позицию стека, остальные данные стека сдвигаются вниз. Величина смещения определяет позицию, куда должны быть вставлены данные (количество элементов данных, расположенных до текущей позиции указателя).	Выход Требуется
SINS(641)							
TB							
C							
S							
<b>STACK DATA DELETE</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) SDEL @SDEL 642	<table border="1"> <tr><td>SDEL(642)</td></tr> <tr><td>TB</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> <p>TB: Начальный адрес стека C: Значение смещения D: Адресуемое слово</p>	SDEL(642)	TB	C	D	Удаление элемента данных, расположенного в указанной позиции стека, и сдвиг остальных данных стека вверх. Величина смещения определяет позицию, откуда должны быть удалены данные (количество элементов данных, расположенных до текущей позиции указателя).	Выход Требуется
SDEL(642)							
TB							
C							
D							

### 3-16 Команды управления данными

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>PID CONTROL</b> PID 190	 <p>S: Входное слово C: 1-е слово параметра D: Выходное слово</p>	Реализация ПИД-регулятора с использованием указанных параметров. 	Выход Требуется
<b>PID CONTROL WITH AUTOTUNING</b> PIDAT 191 (только для CS1-H, CJ1-H или CJ1M)	 <p>S: Входное слово C: 1-е слово параметра D: Выходное слово</p>	Реализация ПИД-регулятора с использованием указанных параметров. Команда PIDAT(191) поддерживает автоматическую настройку ПИД-констант.	Выход Требуется
<b>LIMIT CONTROL</b> LMT @LMT 680	 <p>S: Входное слово C: 1-е слово граничного значения D: Выходное слово</p>	Ограничение выходного значения в диапазоне, определяемом указанными верхней и нижней границами. 	Выход Требуется
<b>DEAD BAND CONTROL</b> BAND @BAND 681	 <p>S: Входное слово C: 1-е слово граничного значения D: Выходное слово</p>	Подавление входного значения в пределах зоны нечувствительности ("мертвой зоны"). 	Выход Требуется

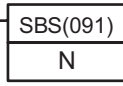
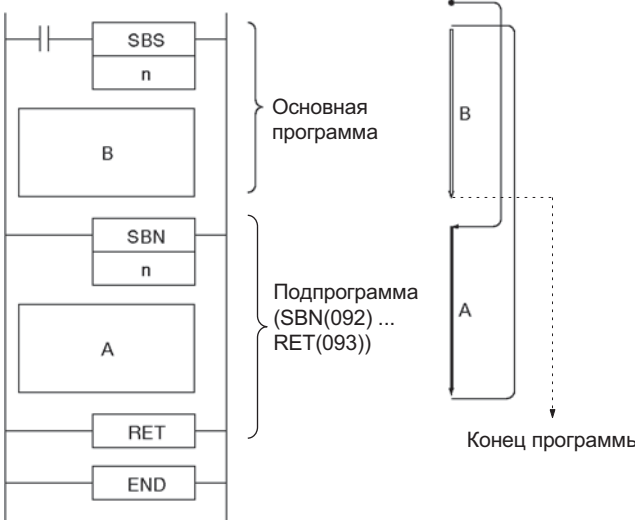
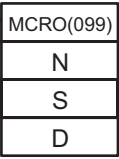
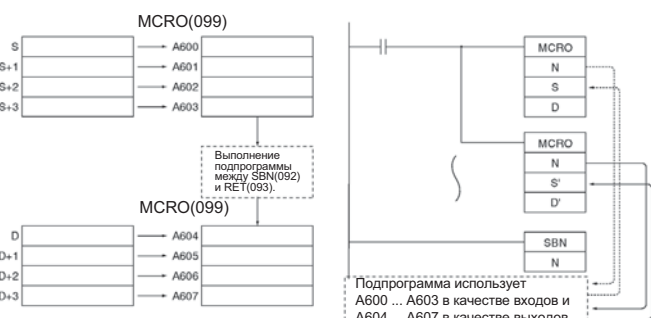
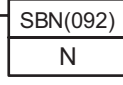
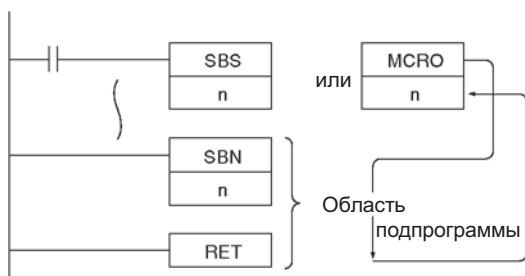
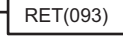


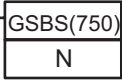
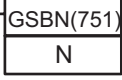

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения				
<b>DEAD ZONE CONTROL</b>  ZONE @ZONE 682	<table border="1" style="margin-left: 20px;"> <tr><td style="text-align: center;">ZONE(682)</td></tr> <tr><td style="text-align: center;">S</td></tr> <tr><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;">D</td></tr> </table> <p>S: Входное слово C: 1-е слово граничного значения D: Выходное слово</p>	ZONE(682)	S	C	D	<p>Смещает входное значение на указанную величину и размещает результат в слово результата.</p>  <p>Положительное смещение (C+1) Отрицательное смещение (C)</p>	Выход Требуется
ZONE(682)							
S							
C							
D							
<b>TIME-PROPORTIONAL OUTPUT</b>  TPO 685  (только для модуля серии CS/CJ версии 2.0 или выше)	<table border="1" style="margin-left: 20px;"> <tr><td style="text-align: center;">TPO (685)</td></tr> <tr><td style="text-align: center;">S</td></tr> <tr><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;">R</td></tr> </table> <p>S: Входное слово C: 1-е слово параметра R: Бит импульсного выхода</p>	TPO (685)	S	C	R	<p>Ввод коэффициента заполнения (скважности) или управляемой переменной из указанного слова, преобразование коэффициента заполнения (скважности) в ШИМ-выход на основе указанных параметров и размещение результата на указанный выход.</p>	Выход Требуется
TPO (685)							
S							
C							
R							
<b>SCALING</b>  SCL @SCL 194	<table border="1" style="margin-left: 20px;"> <tr><td style="text-align: center;">SCL(194)</td></tr> <tr><td style="text-align: center;">S</td></tr> <tr><td style="text-align: center;">P1</td></tr> <tr><td style="text-align: center;">R</td></tr> </table> <p>S: Входное слово P1: 1-е слово параметра R: Слово результата</p>	SCL(194)	S	P1	R	<p>Преобразование двоичного значения без знака в BCD-значение без знака в соответствии с указанной линейной зависимостью.</p>  <p>Масштабирование в соответствии с линейной зависимостью, определяемой точками A и B.</p> <p>Преобразованное значение Преобразованное значение</p>	Выход Требуется
SCL(194)							
S							
P1							
R							

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения													
<p><b>SCALING 2</b></p> <p>SCL2 @SCL2 486</p>	<table border="1" style="margin-left: 20px;"> <tr><td>SCL2(486)</td></tr> <tr><td>S</td></tr> <tr><td>P1</td></tr> <tr><td>R</td></tr> </table> <p>S: Входное слово P1: 1-е слово параметра R: Слово результата</p>	SCL2(486)	S	P1	R	<p>Преобразование двоичного значения со знаком в BCD-значение со знаком в соответствии с указанной линейной зависимостью. В заданную линейную зависимость можно ввести смещение.</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Положительное смещение</p> <p>R (BCD со знаком)</p> </div> <div style="text-align: center;"> <p>Отрицательное смещение</p> <p>R (BCD со знаком)</p> </div> </div> <div style="margin-top: 20px;"> <table border="1" style="margin-left: 20px;"> <tr><td>P1</td><td>Смещение</td><td>(Двоичное со знаком)</td></tr> <tr><td>P1 + 1</td><td>ΔY</td><td>(Двоичное со знаком)</td></tr> <tr><td>P1 + 2</td><td>ΔX</td><td>(BCD со знаком)</td></tr> </table> <div style="margin-left: 100px;"> <p>Смещение = 0000</p> </div> </div>	P1	Смещение	(Двоичное со знаком)	P1 + 1	ΔY	(Двоичное со знаком)	P1 + 2	ΔX	(BCD со знаком)	<p>Выход Требуется</p>
SCL2(486)																
S																
P1																
R																
P1	Смещение	(Двоичное со знаком)														
P1 + 1	ΔY	(Двоичное со знаком)														
P1 + 2	ΔX	(BCD со знаком)														

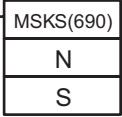
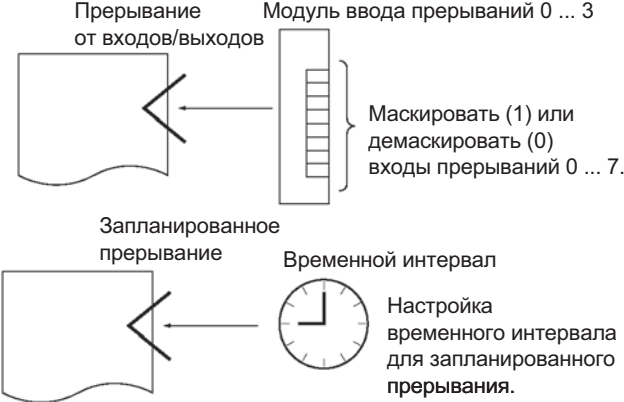
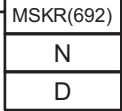
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения				
<p><b>SCALING 3</b></p> <p>SCL3 @SCL3 487</p>	<table border="1" style="margin-left: 20px;"> <tr><td>SCL3(487)</td></tr> <tr><td>S</td></tr> <tr><td>P1</td></tr> <tr><td>R</td></tr> </table> <p>S: Входное слово P1: 1-е слово параметра R: Слово результата</p>	SCL3(487)	S	P1	R	<p>Преобразование BCD-значения со знаком в двоичное значение со знаком в соответствии с заданной линейной зависимостью. В заданную линейную функцию можно ввести смещение.</p> <p>Положительное смещение                      Отрицательное смещение</p> <p>Смещение = 0000</p>	<p>Выход Требуется</p>
SCL3(487)							
S							
P1							
R							
<p><b>AVERAGE</b></p> <p>AVG 195</p>	<table border="1" style="margin-left: 20px;"> <tr><td>AVG(195)</td></tr> <tr><td>S</td></tr> <tr><td>N</td></tr> <tr><td>R</td></tr> </table> <p>S: Входное слово N: Количество циклов R: Слово результата</p>	AVG(195)	S	N	R	<p>Расчет среднего значения для входного слова за указанное количество циклов.</p> <p>S: Входное слово</p> <p>N: Количество циклов</p> <p>R</p> <p>R + 1    Указатель</p> <p>Флаг действительности среднего значения</p> <p>R + 2</p> <p>R + 3</p> <p>Среднее</p> <p>N значений</p> <p>R + N + 1</p>	<p>Выход Требуется</p>
AVG(195)							
S							
N							
R							

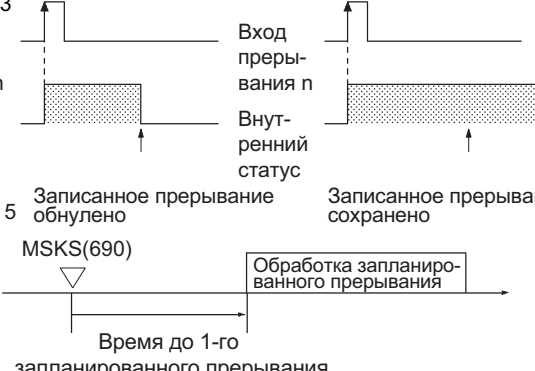
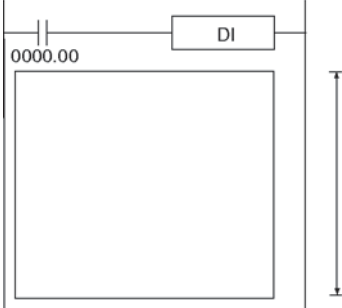
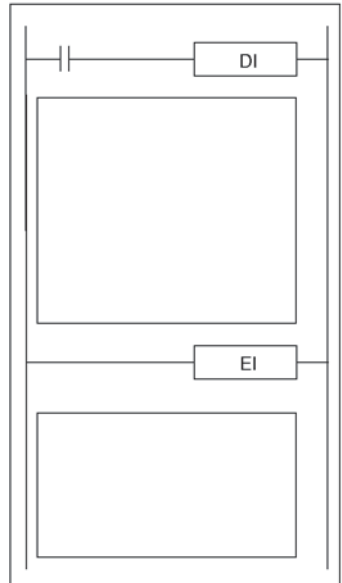
### 3-17 Команды управления подпрограммами

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>SUBROUTINE CALL</b>  SBS @SBS 091	  N: Номер подпрограммы	Вызов подпрограммы с указанным номером и ее выполнение.  Условие выполнения = ВКЛ  	Выход Требуется
<b>MACRO</b>  MCRO @MCRO 099	  N: Номер подпрограммы S: 1-е слово входного параметра D: 1-е слово выходного параметра	Вызов подпрограммы с указанным номером и ее выполнение с применением входных параметров, указанных в S ... S+3, и установкой значений выходных параметров в D ... D+3.  	Выход Требуется
<b>SUBROUTINE ENTRY</b>  SBN 092	  N: Номер подпрограммы	Указывает на начало подпрограммы с указанным номером.  	Выход Не требуется
<b>SUBROUTINE RETURN</b>  RET 093		Указывает на завершение подпрограммы.	Выход Не требуется

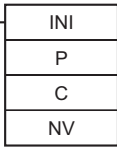
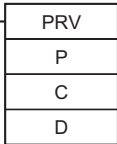
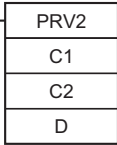

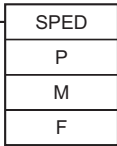
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>GLOBAL SUB-ROUTINE CALL</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) GSBS 750	 N: Номер подпрограммы	Вызов подпрограммы с указанным номером и ее выполнение.	Выход Не требуется
<b>GLOBAL SUB-ROUTINE ENTRY</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) GSBN 751	 N: Номер подпрограммы	Указывает на начало подпрограммы с указанным номером.	Выход Не требуется
<b>GLOBAL SUB-ROUTINE RETURN</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) GRET 752		Указывает на завершение подпрограммы.	Выход Не требуется

### 3-18 Команды управления прерываниями

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>SET INTERRUPT MASK</b> (Не поддерживается модулями CPU CS1D для систем с дублированием CPU) MSKS @MSKS 690	 N: Идентификатор прерывания S: Данные для прерывания	<p>Настройка обработки прерываний от входов/выходов или запланированных прерываний. При первом включении ПЛК задачи обработки прерываний от входов/выходов и задачи обработки запланированных прерываний маскированы (отключены). MSKS(690) можно использовать для демаскирования или маскирования прерываний от входов/выходов и для настройки интервалов времени для запланированных прерываний.</p> 	Выход Требуется
<b>READ INTERRUPT MASK</b> (Не поддерживается модулями CPU CS1D для систем с дублированием CPU) MSKR @MSKR 692	 N: Идентификатор прерывания D: Адресуемое слово	Чтение текущих параметров обработки прерываний, установленных с помощью MSKS(690).	Выход Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<p><b>CLEAR INTERRUPT</b> (Не поддерживается модулями CPU CS1D для систем с дублированием CPU)</p> <p>CLI @CLI 691</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                 CLI(691)  <hr/>                 N  <hr/>                 S             </div> <p>N: Идентификатор прерывания S: Данные для прерывания</p>	<p>Обнуление или сохранение записанных состояний входов прерываний для прерываний от входов/выходов и выбор времени формирования первого запланированного прерывания для запланированных прерываний.</p> <p>N = 0 ... 3 Вход прерывания n Внутренний статус</p> <p>N = 4 ... 5 Записанное прерывание обнулено Записанное прерывание сохранено</p> <p>MSKS(690)</p> <p>Обработка запланированного прерывания</p> <p>Время до 1-го запланированного прерывания</p> 	<p>Выход Требуется</p>
<p><b>DISABLE INTERRUPTS</b></p> <p>DI @DI 693</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                 DI(693)             </div>	<p>Запрет выполнения всех задач обработки прерываний, за исключением прерывания по выключению питания.</p> 	<p>Выход Требуется</p>
<p><b>ENABLE INTERRUPTS</b></p> <p>EI 694</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                 EI(694)             </div>	<p>Разрешение выполнения всех задач обработки прерываний, отключенных с помощью DI(693).</p> 	<p>Выход Не требуется</p>

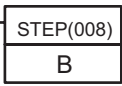
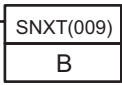
### 3-19 Команды управления высокоскоростными счетчиками и импульсными выходами (только CJ1M-CPU21/22/23)

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>MODE CONTROL</b> INI @INI 880	 <p>P: Указатель порта C: Управляющие данные NV: 1-е слово нового PV</p>	Команда IN(880) служит для запуска и прекращения сравнения с заданным значением, для изменения текущего значения (PV) скоростного счетчика, для изменения значения PV входа прерывания (в режиме счетчика), для изменения значения PV импульсного выхода или для прекращения работы импульсного выхода.	Выход Требуется
<b>HIGH-SPEED COUNTER PV READ</b> PRV @PRV 881	 <p>P: Указатель порта C: Управляющие данные D: 1-е адресуемое слово</p>	Команда PRV(881) предназначена для чтения текущего значения (PV) скоростного счетчика, импульсного выхода или входа прерывания (в режиме счетчика).	Выход Требуется
<b>COUNTER FREQUENCY CONVERT</b> PRV2 883  (только для модулей CPU CJ1M версии 2.0 или выше)	 <p>C1: Управляющие данные C2: Импульсы/обороты D: 1-е адресуемое слово</p>	Команда выполняет чтение значения со входа частоты импульсов скоростного счетчика и либо преобразует частоту в частоту вращения (число оборотов), либо преобразует значение PV счетчика в суммарное число оборотов. Результат записывается в адресуемое слово в формате 8-разрядного 16-ричного числа. Импульсы могут быть получены только от скоростного счетчика 0.	Выход Требуется
<b>COMPARISON TABLE LOAD</b> CTBL @CTBL 882	 <p>P: Указатель порта C: Управляющие данные TB: 1-е слово таблицы сравнения</p>	Команда CTBL(882) предназначена для выполнения сравнения текущего значения (PV) скоростного счетчика с заданным значением или с диапазоном значений.	Выход Требуется
<b>SPEED OUTPUT</b> SPED @SPED 885	 <p>P: Указатель порта M: Режим выхода F: 1-е слово частоты импульсов</p>	Команда SPED(885) служит для выбора частоты и вывода импульсов с этой частотой, без ускорения или торможения.	Выход Требуется

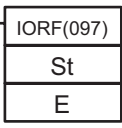
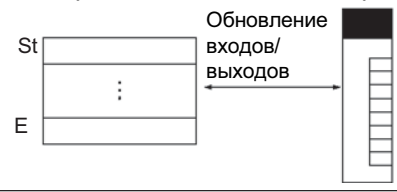
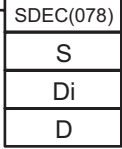

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения					
<b>SET PULSES</b> PULS @PULS 886	<table border="1"> <tr><td>PULS</td></tr> <tr><td>P</td></tr> <tr><td>T</td></tr> <tr><td>N</td></tr> </table> <p><b>P:</b> Указатель порта <b>T:</b> Тип импульсов <b>N:</b> Количество импульсов</p>	PULS	P	T	N	PULS(886) служит для выбора количества импульсов для импульсного выхода.	Выход Требуется	
PULS								
P								
T								
N								
<b>PULSE OUTPUT</b> PLS2 @PLS2 887	<table border="1"> <tr><td>PLS2</td></tr> <tr><td>P</td></tr> <tr><td>M</td></tr> <tr><td>S</td></tr> <tr><td>F</td></tr> </table> <p><b>P:</b> Указатель порта <b>M:</b> Режим выхода <b>S:</b> 1-е слово таблицы параметров <b>F:</b> 1-е слово начальной частоты</p>	PLS2	P	M	S	F	PLS2(887) позволяет установить частоту импульсов, а также скорости разгона/торможения и реализовать формирование импульсов с выбранным разгоном/торможением (то есть, с различными скоростями нарастания/убывания частоты). Можно реализовать только позиционирование.	Выход Требуется
PLS2								
P								
M								
S								
F								
<b>ACCELERATION CONTROL</b> ACC @ACC 888	<table border="1"> <tr><td>ACC</td></tr> <tr><td>P</td></tr> <tr><td>M</td></tr> <tr><td>S</td></tr> </table> <p><b>P:</b> Указатель порта <b>M:</b> Режим выхода <b>S:</b> 1-е слово таблицы параметров</p>	ACC	P	M	S	ACC(888) позволяет установить частоту импульсов и скорости разгона/торможения и реализовать вывод импульсов с разгоном/торможением (с одинаковой скоростью нарастания/убывания частоты). Возможно как позиционирование, так и регулирование скорости.	Выход Требуется	
ACC								
P								
M								
S								
<b>ORIGIN SEARCH</b> ORG @ORG 889	<table border="1"> <tr><td>ORG</td></tr> <tr><td>P</td></tr> <tr><td>C</td></tr> </table> <p><b>P:</b> Указатель порта <b>C:</b> Управляющие данные</p>	ORG	P	C	ORG(889) служит для реализации операций поиска и возврата в опорную (исходную) точку.	Выход Требуется		
ORG								
P								
C								
<b>PULSE WITH VARIABLE DUTY FACTOR</b> PWM @ 891	<table border="1"> <tr><td>PWM</td></tr> <tr><td>P</td></tr> <tr><td>F</td></tr> <tr><td>D</td></tr> </table> <p><b>P:</b> Указатель порта <b>F:</b> Частота <b>D:</b> Коэффициент заполнения</p>	PWM	P	F	D	PWM(891) предназначена для генерирования импульсов с переменным коэффициентом заполнения (скважностью).	Выход Требуется	
PWM								
P								
F								
D								

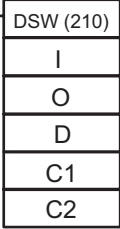
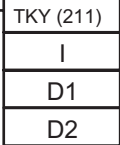
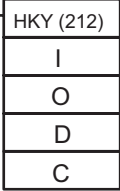
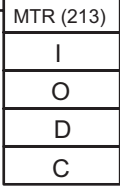


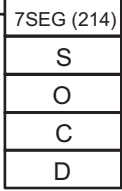
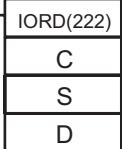

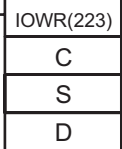
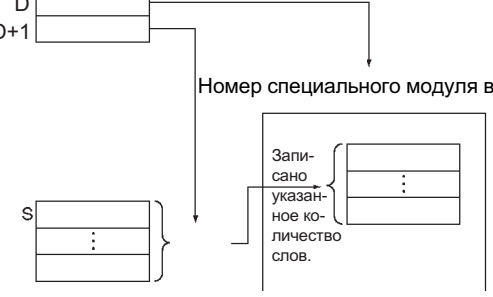
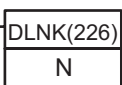
### 3-20 Шаговые команды

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>STEP DEFINE</b> STEP 008	 B: Бит	В зависимости от своего положения в программе и от состояния управляющего бита, команда STEP(008) выполняет одно из следующих действий. (1) Запуск указанного шага. (2) Завершение области пошагового выполнения программы.	Выход Требуется
<b>STEP START</b> SNXT 009	 B: Бит	Команда SNXT(009) выполняет одну из следующих функций: (1) Запуск пошагового выполнения программы. (2) Переход к следующему биту управления шагом. (3) Завершение пошагового выполнения программы.	Выход Требуется

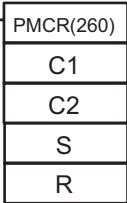
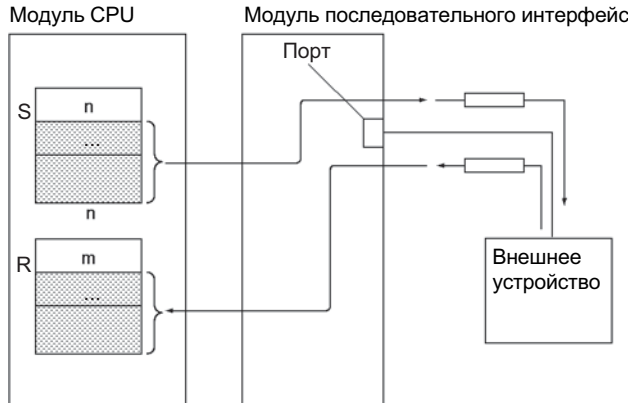
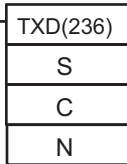
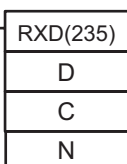
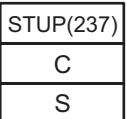
### 3-21 Команды для базовых модулей ввода/вывода

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>I/O REFRESH</b> IORF @IORF 097	 St: Начальное слово E: Конечное слово	Обновление указанных слов ввода/вывода.  Область битов ввода/вывода или область битов специального модуля ввода/вывода.      Модуль ввода/вывода или специальный модуль ввода/вывода.  	Выход Требуется
<b>7-SEGMENT DECODER</b> SDEC @SDEC 078	 S: Входное слово Di: Обозначение разряда D: 1-е конечное слово	Преобразование 16-ричного содержимого указанных разрядов в 8-битовый код 7-сегментного индикатора и размещение результата в старшие или младшие 8 битов указанных адресуемых слов.  	Выход Требуется

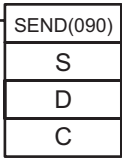
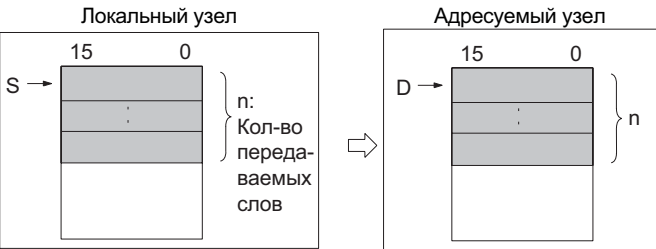
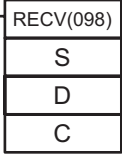
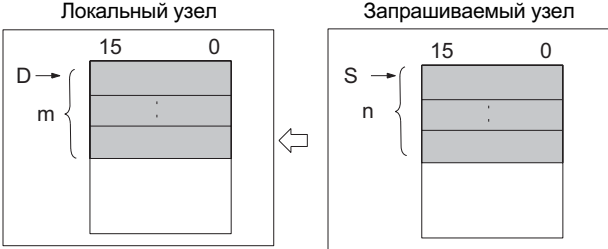
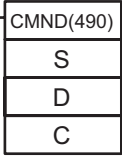
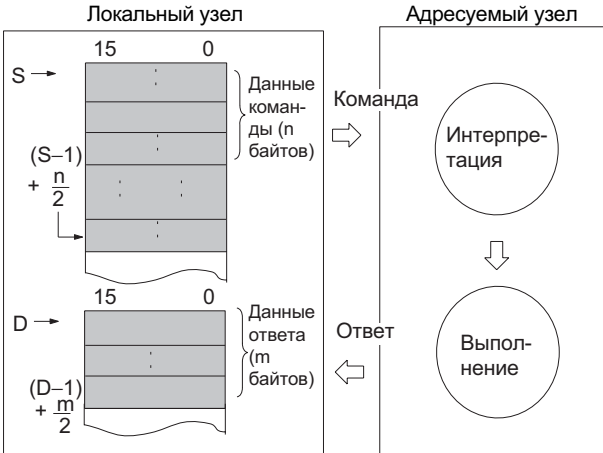
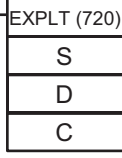
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<p><b>DIGITAL SWITCH INPUT</b></p> <p>DSW 210</p> <p>(только для модулей CPU серии CS/CJ версии 2.0 или выше)</p>	 <p>I: Входное слово данных (D0 ... D3) O: Выходное слово D: 1-е слово результата C1: Количество разрядов C2: Системное слово</p>	<p>Чтение значения, установленного с помощью внешнего цифрового переключателя (или механического вращающегося переключателя), подсоединенного к модулю ввода или модулю вывода, и запись результата в указанные слова в формате 4-разрядного или 8-разрядного BCD-числа.</p>	<p>Выход Требуется</p>
<p><b>TEN KEY INPUT</b></p> <p>TKY 211</p> <p>(только для модулей CPU серии CS/CJ версии 2.0 или выше)</p>	 <p>I: Входное слово данных D1: 1-е слово регистра D2: Слово ввода с клавиатуры</p>	<p>Чтение числового значения, введенного с десятичной клавиатуры, подсоединенной к модулю ввода, и запись до 8-разрядов BCD-числа в указанные слова.</p>	<p>Выход Требуется</p>
<p><b>HEXADECIMAL KEY INPUT</b></p> <p>HKY 212</p> <p>(только для модулей CPU серии CS/CJ версии 2.0 или выше)</p>	 <p>I: Входное слово данных O: Выходное слово D: 1-е слово регистра C: Системное слово</p>	<p>Чтение числового значения, введенного с шестнадцатеричной клавиатуры, подсоединенной к модулю ввода или модулю вывода, и запись до 8-разрядов 16-ричного числа в указанные слова.</p>	<p>Выход Требуется</p>
<p><b>MATRIX INPUT</b></p> <p>MTR 213</p> <p>(только для модулей CPU серии CS/CJ версии 2.0 или выше)</p>	 <p>I: Входное слово данных O: Выходное слово D: 1-е адресуемое слово C: Системное слово</p>	<p>Ввод до 64 сигналов матрицы 8×8, подсоединенной к модулю ввода и модулю вывода (используются 8 входов и 8 выходов), и запись 64-битового значения в четыре адресуемых слова.</p>	<p>Выход Требуется</p>

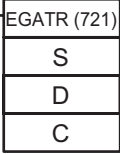
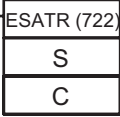
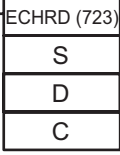
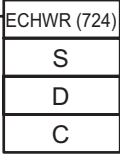
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>7-SEGMENT DISPLAY OUTPUT</b> 7SEG 214 (только для модулей CPU серии CS/CJ версии 2.0 или выше)	 <p>S: 1-е входное слово                      O: Выходное слово                      C: Управляющее слово                      D: Системное слово</p>	Преобразование входных данных (4-разрядное или 8-разрядное BCD-число) в формат 7-сегментного индикатора и размещение результата в указанное выходное слово.	Выход Требуется
<b>INTELLIGENT I/O READ</b> IORD @IORD 222	 <p>C: Управляющие данные                      S: Источник и количество слов                      D: Адресат и количество слов</p>	Чтение содержимого области памяти специального модуля ввода/вывода или модуля шины CPU (см. примечание).  <p><b>Примечание:</b> Модуль CPU серии CS/CJ версии 2.0 или выше (включая модули CPU CS1-H, CJ1-H и CJ1M, начиная с номера партии 030418) могут читать данные из модулей шины CPU.</p>	Выход Требуется
<b>INTELLIGENT I/O WRITE</b> IOWR @IOWR 223	 <p>C: Управляющие данные                      S: Источник и количество слов                      D: Передатчик и количество слов</p>	Вывод содержимого области памяти ввода/вывода модуля CPU в специальный модуль ввода/вывода или модуль шины CPU (см. примечание).  <p><b>Примечание:</b> Модуль CPU серии CS/CJ версии 2.0 или выше (включая модули CPU CS1-H, CJ1-H и CJ1M, начиная с номера партии 030418) могут записывать данные в модули шины CPU.</p>	Выход Требуется
<b>CPU BUS UNIT I/O REFRESH</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) DLNK @DLNK 226	 <p>N: Номер модуля</p>	Мгновенное обновление входов/выходов модуля шины CPU с указанным номером.	Выход Требуется

### 3-22 Команды последовательного интерфейса

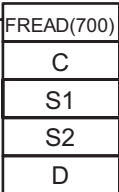
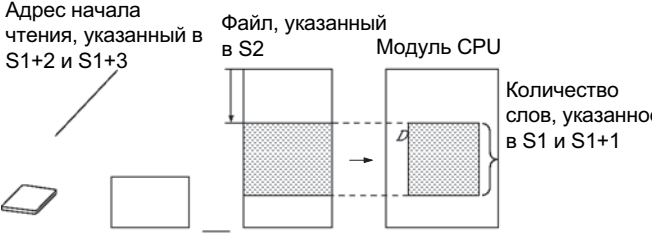
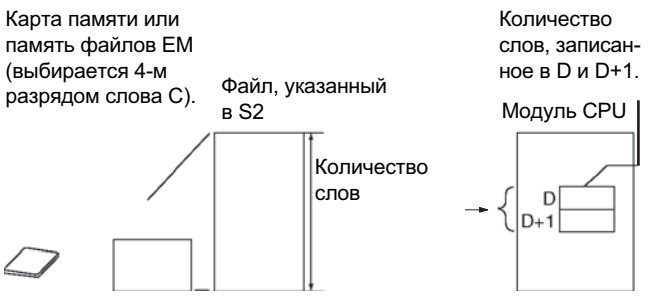
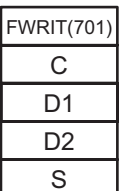



Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>PROTOCOL MACRO</b>  PMCR @PMCR 260	 <p>C1: Управляющее слово 1                      C2: Управляющее слово 2                      S: 1-е переданное слово                      R: 1-е принятое слово</p>	<p>Вызов и выполнение коммуникационной последовательности (сеанса связи), зарегистрированной в плате последовательного интерфейса (только серия CS) или модуле последовательного интерфейса.</p> 	Выход Требуется
<b>TRANSMIT</b>  TXD @TXD 236	 <p>S: 1-е слово источника                      C: Управляющее слово                      N: Кол-во байтов                      0000 ... 0100 hex                      (0 ... 256 десят.)</p>	<p>Вывод указанного количества байтов данных через порт RS-232C, встроенный в модуль CPU.</p>	Выход Требуется
<b>RECEIVE</b>  RXD @RXD 235	 <p>D: 1-е адресуемое слово                      C: Управляющее слово                      N: Кол-во записываемых байтов                      0000 ... 0100 hex                      (0 ... 256 десят.)</p>	<p>Чтение указанного количества байтов данных через порт RS-232C, встроенный в модуль CPU.</p>	Выход Требуется
<b>CHANGE SERIAL PORT SETUP</b>  STUP @STUP 237	 <p>C: Управляющее слово (порт)                      S: 1-е входное слово</p>	<p>Изменение параметров связи для последовательного порта модуля CPU, модуля последовательного интерфейса (модуль шины CPU) или платы последовательного интерфейса. Команда STUP(237), таким образом, позволяет изменять режим связи (протокол) во время работы ПЛК.</p>	Выход Требуется

## 3-23 Сетевые команды

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>NETWORK SEND</b> SEND @SEND 090	 <p><b>S:</b> 1-е входное слово <b>D:</b> 1-е адресуемое слово <b>C:</b> 1-е управляющее слово</p>	<p>Передача данных сетевому узлу.</p>  <p>Локальный узел                      Адресуемый узел</p>	Выход Требуется
<b>NETWORK RECEIVE</b> RECV @RECV 098	 <p><b>S:</b> 1-е входное слово <b>D:</b> 1-е адресуемое слово <b>C:</b> 1-е управляющее слово</p>	<p>Запрос сетевому узлу на передачу данных и прием данных.</p>  <p>Локальный узел                      Запрашиваемый узел</p>	Выход Требуется
<b>DELIVER COMMAND</b> CMND @CMND 490	 <p><b>S:</b> 1-е слово команды <b>D:</b> 1-е слово ответа <b>C:</b> 1-е управляющее слово</p>	<p>Передача команд FINS и прием ответов.</p>  <p>Локальный узел                      Адресуемый узел</p> <p>Команда</p> <p>Интерпретация</p> <p>Выполнение</p> <p>Ответ</p>	Выход Требуется
<b>EXPLICIT MESSAGE SEND</b> EXPLT 720  (только для модулей CPU серии CS/CJ версии 2.0 или выше)	 <p><b>S:</b> 1-е слово передаваемого сообщения <b>D:</b> 1-е слово принятого сообщения <b>C:</b> 1-е управляющее слово</p>	<p>Передача сообщения в явном виде с любым кодом сервиса.</p>	Выход Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>EXPLICIT GET ATTRIBUTE</b> EGATR 721  (только для модулей CPU серии CS/CJ версии 2.0 или выше)	 <p><b>S:</b> 1-е слово передаваемого сообщения <b>D:</b> 1-е слово принятого сообщения <b>C:</b> 1-е управляющее слово сообщения</p>	Чтение информации о состоянии с помощью явного сообщения (Get Attribute Single, код сервиса: 0E hex).	Выход Требуется
<b>EXPLICIT SET ATTRIBUTE</b> ESATR 722  (только для модулей CPU серии CS/CJ версии 2.0 или выше)	 <p><b>S:</b> 1-е слово передаваемого сообщения <b>C:</b> 1-е управляющее слово</p>	Запись информации о состоянии с помощью явного сообщения (Set Attribute Single, код сервиса: 0E hex).	Выход Требуется
<b>EXPLICIT WORD READ</b> ECHRD 723  (только для модулей CPU серии CS/CJ версии 2.0 или выше)	 <p><b>S:</b> 1-е слово источника удаленного модуля CPU <b>D:</b> 1-е адресуемое слово локального модуля CPU <b>C:</b> 1-е управляющее слово</p>	Чтение данных из удаленного модуля CPU и запись в локальный модуль CPU по сети (удаленный модуль CPU должен поддерживать явные сообщения).	Выход Требуется
<b>EXPLICIT WORD WRITE</b> ECHWR 724  (только для модулей CPU серии CS/CJ версии 2.0 или выше)	 <p><b>S:</b> 1-е слово источника локального модуля CPU <b>D:</b> 1-е адресуемое слово удаленного модуля CPU <b>C:</b> 1-е управляющее слово</p>	Запись данных локального модуля CPU в удаленный модуль CPU по сети (удаленный модуль CPU должен поддерживать явные сообщения).	Выход Требуется

### 3-24 Команды памяти файлов

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<p><b>READ DATA FILE</b> FREAD @FREAD 700</p>	 <p>C: Управляющее слово S1: 1-е входное слово S2: Имя файла D: 1-е адресуемое слово</p>	<p>Чтение указанных данных или области данных из указанного файла данных, расположенного в памяти файлов, и запись в указанную область данных модуля CPU.</p> <p>Адрес начала чтения, указанный в S1+2 и S1+3</p> <p>Файл, указанный в S2</p> <p>Модуль CPU</p> <p>Количество слов, указанное в S1 и S1+1</p>  <p>Количество слов, записанное в D и D+1.</p> <p>Модуль CPU</p>  <p>Карта памяти или память файлов EM (выбирается 4-м разрядом слова C).</p> <p>Файл, указанный в S2</p> <p>Количество слов</p> <p>Карта памяти или память файлов EM (указывается 4-м разрядом слова C).</p>	<p>Выход Требуется</p>
<p><b>WRITE DATA FILE</b> FWRIT @FWRIT 701</p>	 <p>C: Управляющее слово D1: 1-е адресуемое слово D2: Имя файла S: 1-е входное слово</p>	<p>Запись поверх или добавление к данным указанного файла данных, расположенного в памяти файлов, указанных данных из области данных модуля CPU. Если указанный файл не существует, создается новый файл с этим именем.</p> <p>Модуль CPU</p> <p>Начальный адрес, указанный в S</p> <p>Начальное слово, указанное в D1+2 и D1+3</p> <p>Файл, указанный в D2</p> <p>Количество слов, указанное в D1 и D1+1</p> <p>Запись поверх</p>  <p>Карта памяти или память файлов EM (указывается 4-м разрядом слова C).</p> <p>Модуль CPU</p> <p>Начальный адрес, указанный в S</p> <p>Конец файла</p> <p>Файл, указанный в D2</p> <p>Имеющиеся данные</p> <p>Количество слов, указанное в D1 и D1+1</p> <p>Добавление</p>  <p>Карта памяти или память файлов EM (указывается 4-м разрядом слова C).</p> <p>Модуль CPU</p> <p>Начальный адрес, указанный в S</p> <p>Начало файла</p> <p>Файл, указанный в D2</p> <p>Новый созданный файл</p> <p>Количество слов, указанное в D1 и D1+1</p>  <p>Карта памяти или память файлов EM (указывается 4-м разрядом слова C).</p>	<p>Выход Требуется</p>

### 3-25 Команды дисплея

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения			
<b>DISPLAY MESSAGE</b> MSG @MSG 046	<table border="1"> <tr><td>MSG(046)</td></tr> <tr><td>N</td></tr> <tr><td>M</td></tr> </table> <p>N: Номер сообщения M: 1-е слово сообщения</p>	MSG(046)	N	M	Чтение указанных 16-ти слов расширенного ASCII-формата и отображение сообщения на дисплее периферийного устройства, например, на консоли программирования.	Выход Требуется
MSG(046)						
N						
M						

### 3-26 Команды управления часами

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения																																					
<b>CALENDAR ADD</b> CADD @CADD 730	<table border="1"> <tr><td>CADD(730)</td></tr> <tr><td>C</td></tr> <tr><td>T</td></tr> <tr><td>R</td></tr> </table> <p>C: 1-е слово календаря T: 1-е слово времени R: 1-е слово результата</p>	CADD(730)	C	T	R	<p>Увеличение времени календаря на время, указанное в словах.</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>C</td><td>Минуты</td><td>Секунды</td></tr> <tr><td>C+1</td><td>День</td><td>Час</td></tr> <tr><td>C+2</td><td>Год</td><td>Месяц</td></tr> </table> <p style="text-align: center;">+</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>T</td><td>Минуты</td><td>Секунды</td></tr> <tr><td>T+1</td><td colspan="2">Часы</td></tr> </table> <p style="text-align: center;">↓</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>R</td><td>Минуты</td><td>Секунды</td></tr> <tr><td>R+1</td><td>День</td><td>Час</td></tr> <tr><td>R+2</td><td>Год</td><td>Месяц</td></tr> </table>	15	87	0	C	Минуты	Секунды	C+1	День	Час	C+2	Год	Месяц	15	87	0	T	Минуты	Секунды	T+1	Часы		15	87	0	R	Минуты	Секунды	R+1	День	Час	R+2	Год	Месяц	Выход Требуется
CADD(730)																																								
C																																								
T																																								
R																																								
15	87	0																																						
C	Минуты	Секунды																																						
C+1	День	Час																																						
C+2	Год	Месяц																																						
15	87	0																																						
T	Минуты	Секунды																																						
T+1	Часы																																							
15	87	0																																						
R	Минуты	Секунды																																						
R+1	День	Час																																						
R+2	Год	Месяц																																						
<b>CALENDAR SUBTRACT</b> CSUB @CSUB 731	<table border="1"> <tr><td>CSUB(731)</td></tr> <tr><td>C</td></tr> <tr><td>T</td></tr> <tr><td>R</td></tr> </table> <p>C: 1-е слово календаря T: 1-е слово времени R: 1-е слово результата</p>	CSUB(731)	C	T	R	<p>Уменьшение времени календаря на значение времени, указанное в словах.</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>C</td><td>Минуты</td><td>Секунды</td></tr> <tr><td>C+1</td><td>День</td><td>Час</td></tr> <tr><td>C+2</td><td>Год</td><td>Месяц</td></tr> </table> <p style="text-align: center;">-</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>T</td><td>Минуты</td><td>Секунды</td></tr> <tr><td>T+1</td><td colspan="2">Часы</td></tr> </table> <p style="text-align: center;">↓</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>R</td><td>Минуты</td><td>Секунды</td></tr> <tr><td>R+1</td><td>День</td><td>Час</td></tr> <tr><td>R+2</td><td>Год</td><td>Месяц</td></tr> </table>	15	87	0	C	Минуты	Секунды	C+1	День	Час	C+2	Год	Месяц	15	87	0	T	Минуты	Секунды	T+1	Часы		15	87	0	R	Минуты	Секунды	R+1	День	Час	R+2	Год	Месяц	Выход Требуется
CSUB(731)																																								
C																																								
T																																								
R																																								
15	87	0																																						
C	Минуты	Секунды																																						
C+1	День	Час																																						
C+2	Год	Месяц																																						
15	87	0																																						
T	Минуты	Секунды																																						
T+1	Часы																																							
15	87	0																																						
R	Минуты	Секунды																																						
R+1	День	Час																																						
R+2	Год	Месяц																																						

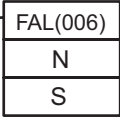
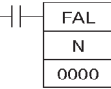
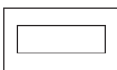
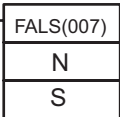
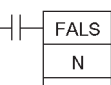
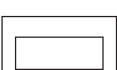
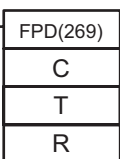
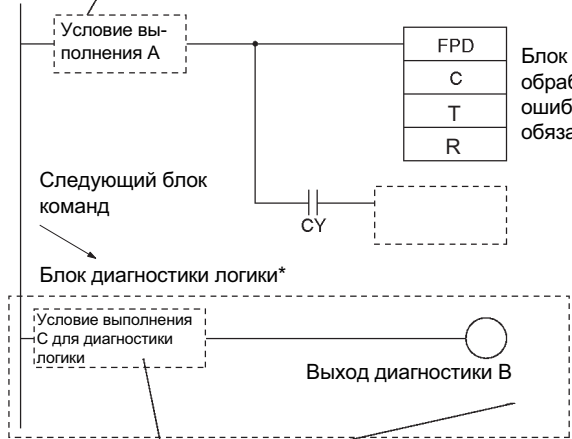


Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения			
<b>HOURS TO SECONDS</b> SEC @SEC 065	<table border="1"> <tr><td>SEC(065)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: 1-е входное слово D: 1-е адресуемое слово</p>	SEC(065)	S	D	<p>Преобразование значения времени, заданного в часах/минутах/секундах, в значение времени, выраженное в секундах.</p>	Выход Требуется
SEC(065)						
S						
D						
<b>SECONDS TO HOURS</b> HMS @HMS 066	<table border="1"> <tr><td>HMS(066)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: 1-е входное слово D: 1-е адресуемое слово</p>	HMS(066)	S	D	<p>Преобразование значения времени, выраженного в секундах, в значение, выраженное в часах/минутах/секундах.</p>	Выход Требуется
HMS(066)						
S						
D						
<b>CLOCK ADJUSTMENT</b> DATE @DATE 735	<table border="1"> <tr><td>DATE(735)</td></tr> <tr><td>S</td></tr> </table> <p>S: 1-е входное слово</p>	DATE(735)	S	<p>Запись значения времени, заданного в указанных входных словах, во внутренние часы.</p> <p>Модуль CPU</p>	Выход Требуется	
DATE(735)						
S						

### 3-27 Команды отладки программы

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения	
<b>TRACE MEMORY SAMPLING</b> TRSM 045	<table border="1"> <tr><td>TRSM(045)</td></tr> </table>	TRSM(045)	<p>Когда выполняется команда TRSM(045), считывается статус предварительно выбранного бита или слова. Статус записывается в память протоколирования (Трасе Мемори). Команду TRSM(045) можно использовать в любом месте программы любое количество раз.</p>	Выход Не требуется
TRSM(045)				

### 3-28 Команды для диагностики неисправностей

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>FAILURE ALARM</b> FAL @FAL 006	 <p>N: Номер FAL S: 1-е слово сообщения или код генерируемой ошибки</p>	<p>Формирование или сброс нефатальных ошибок пользователя. Нефатальные ошибки не приводят к остановке работы ПЛК. Команда также формирует нефатальные ошибки системы.</p>  <p>Выполнение FAL(006) формирует нефатальную ошибку с указанным номером FAL (N).</p> <ul style="list-style-type: none"> <li>Флаг "Ошибка FAL" = ВКЛ</li> <li>Соответствующий флаг "Номер вызванной ошибки FAL" = ВКЛ</li> <li>Код ошибки, записанный в A400</li> <li>Код ошибки и время, записанные в область протокола ошибок</li> <li>Индикатор ERR мигает</li> </ul>  <p>Сообщение, отображаемое на консоли программирования</p>	<p>Выход Требуется</p>
<b>SEVERE FAILURE ALARM</b> FALS 007	 <p>N: Номер FALS S: 1-е слово сообщения или код генерируемой ошибки</p>	<p>Команда служит для формирования фатальных ошибок пользователя. Фатальные ошибки приводят к прекращению работы ПЛК. Команда также формирует фатальные ошибки системы.</p>  <p>Выполнение команды FALS(007) приводит к формированию фатальной ошибки с указанным номером FALS (N).</p> <ul style="list-style-type: none"> <li>Флаг "Ошибка FALS" = ВКЛ</li> <li>Код ошибки, записанный в A400</li> <li>Код ошибки и время/дата, записанные в область протокола ошибок</li> <li>Индикатор ERR светится</li> </ul>  <p>Сообщение, отображаемое на консоли программирования</p>	<p>Выход Требуется</p>
<b>FAILURE POINT DETECTION</b> FPD 269	 <p>C: Управляющее слово T: Время контроля R: 1-е слово регистра</p>	<p>Команда служит для диагностики ошибок в блоке команд, контролируя время, которое проходит между выполнением FPD(269) и активизацией выхода диагностики, и обнаруживая вход, который препятствует включению выхода.</p> <p>Функция контроля времени: Отсчет времени начинается с момента включения условия выполнения А. Если выход В не включается в течение контрольного времени, формируется нефатальная ошибка.</p>  <p>Блок обработки ошибок (не обязателен)</p> <p>Следующий блок команд</p> <p>Блок диагностики логики*</p> <p>Условие выполнения С для диагностики логики</p> <p>Выход диагностики В</p> <p>Функция диагностики логики. Позволяет определить вход С, который препятствует включению выхода В</p>	<p>Выход Требуется</p>

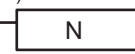
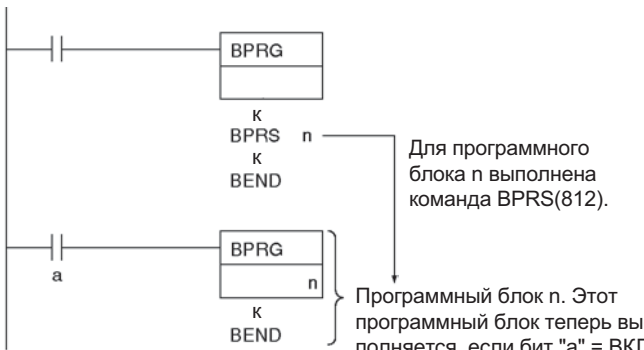
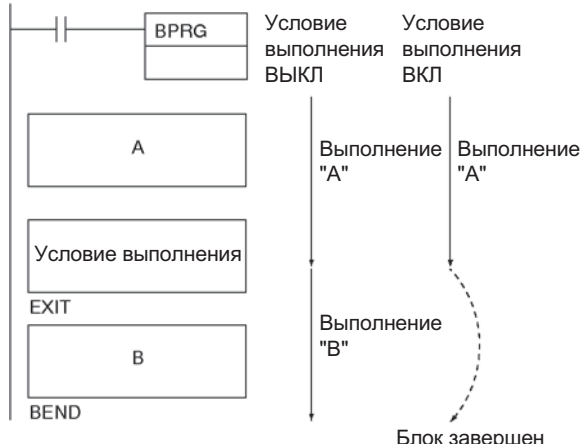
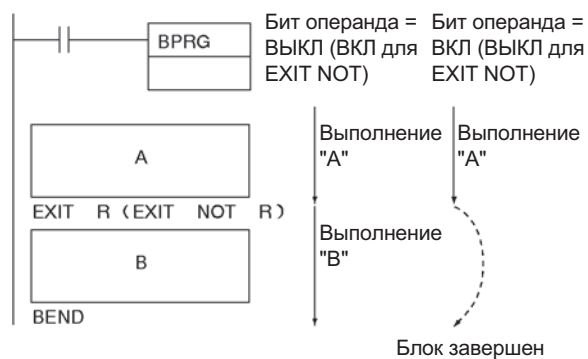
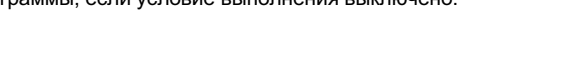
### 3-29 Прочие команды

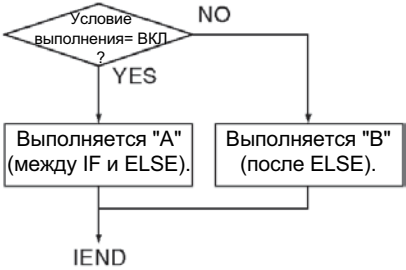

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения			
<b>SET CARRY</b> STC @STC 040	— <table border="1" style="display: inline-table;"><tr><td>STC(040)</td></tr></table>	STC(040)	Устанавливает флаг переноса (CY).	Выход Требуется		
STC(040)						
<b>CLEAR CARRY</b> CLC @CLC 041	— <table border="1" style="display: inline-table;"><tr><td>CLC(041)</td></tr></table>	CLC(041)	Сбрасывает флаг переноса (CY).	Выход Требуется		
CLC(041)						
<b>SELECT EM BANK</b> EMBC @EMBC 281	— <table border="1" style="display: inline-table;"><tr><td>EMBC(281)</td></tr><tr><td>N</td></tr></table> N: Номер банка EM	EMBC(281)	N	Изменяет текущий банк EM.	Выход Требуется	
EMBC(281)						
N						
<b>EXTEND MAXIMUM CYCLE TIME</b> WDT @WDT 094	— <table border="1" style="display: inline-table;"><tr><td>WDT(094)</td></tr><tr><td>T</td></tr></table> T: Время таймера	WDT(094)	T	Команда увеличивает максимальную длительность цикла, но только для того цикла, в котором она выполняется.	Выход Требуется	
WDT(094)						
T						
<b>SAVE CONDITION FLAGS</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) CCS @CCS 282	— <table border="1" style="display: inline-table;"><tr><td>CCS(282)</td></tr></table>	CCS(282)	Сохраняет статус флагов условий.	Выход Требуется		
CCS(282)						
<b>LOAD CONDITION FLAGS</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) CCL @CCL 283	— <table border="1" style="display: inline-table;"><tr><td>CCL(283)</td></tr></table>	CCL(283)	Чтение сохраненного статуса флагов условий.	Выход Требуется		
CCL(283)						
<b>CONVERT ADDRESS FROM</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) FRMCV @FRMCV 284	— <table border="1" style="display: inline-table;"><tr><td>FRMCV(284)</td></tr><tr><td>S</td></tr><tr><td>D</td></tr></table> S: Слово, содержащее адрес памяти серии CV D: Адресуемый регистр индексов	FRMCV(284)	S	D	Преобразует адрес памяти ПЛК серии CV в эквивалентный адрес памяти ПЛК серии CS/CJ.	Выход Требуется
FRMCV(284)						
S						
D						
<b>CONVERT ADDRESS TO CV</b> (только для CS1-H, CJ1-H, CJ1M или CS1D) TOCV @TOCV 285	— <table border="1" style="display: inline-table;"><tr><td>TOCV(285)</td></tr><tr><td>S</td></tr><tr><td>D</td></tr></table> S: Регистр индексов, содержащий адрес памяти серии CS D: Адресуемое слово	TOCV(285)	S	D	Преобразует адрес памяти ПЛК серии CS/CJ в эквивалентный адрес памяти ПЛК серии CV.	Выход Требуется
TOCV(285)						
S						
D						

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>DISABLE PERIPHERAL SERVICING</b> (только для модулей CPU CS1D для однопроцессорных систем, и для CS1-H, CJ1-H или CJ1M) IOSP @IOSP 287	— <span style="border: 1px solid black; padding: 2px;">IOSP(287)</span>	Запрещает обслуживание периферии во время выполнения программы в одном из режимов параллельного выполнения или в режиме приоритетного обслуживания периферии.	Выход Требуется
<b>ENABLE PERIPHERAL SERVICING</b> (только для модулей CPU CS1D для однопроцессорных систем, и для CS1-H, CJ1-H или CJ1M) IORS 288	— <span style="border: 1px solid black; padding: 2px;">IORS(288)</span>	Разрешает обслуживание периферии, отключенное командой IOSP(287), во время выполнения программы в одном из режимов параллельного выполнения или в режиме приоритетного обслуживания периферии.	Выход Не требуется

### 3-30 Команды для программирования блоков

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>BLOCK PROGRAM BEGIN</b>  BPRG 096	— <span style="border: 1px solid black; padding: 2px;">BPRG(096)</span>  <span style="border: 1px solid black; padding: 2px;">N</span>  N: Номер программного блока	<p>Определяет область программного блока. Для каждой команды BPRG(096) обязательно должна быть предусмотрена соответствующая команда BEND(801).</p> <p>Выполняется при включенном условии выполнения.</p>	Выход Требуется
<b>BLOCK PROGRAM END</b> BEND 801		Определяет область программного блока. Для каждой команды BPRG(096) обязательно должна быть предусмотрена соответствующая команда BEND(801).	Программный блок Требуется
<b>BLOCK PROGRAM PAUSE</b>  BPPS 811	BPPS (811) — <span style="border: 1px solid black; padding: 2px;">N</span>  N: Номер программного блока	<p>Приостанавливает и перезапускает указанный программный блок из другого программного блока.</p>	Программный блок Требуется

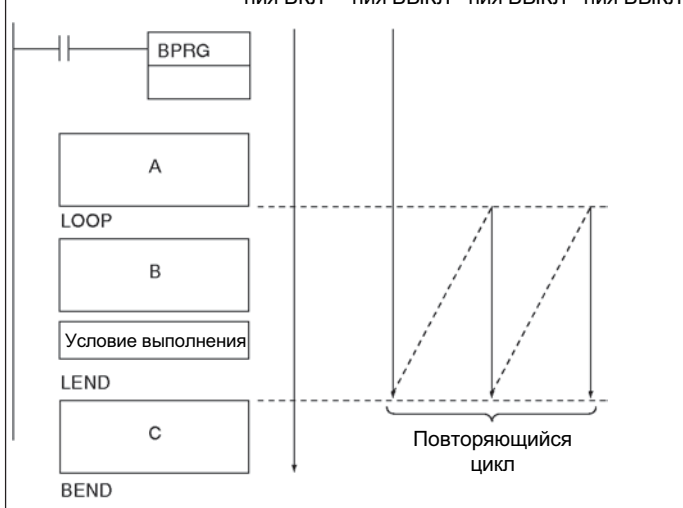
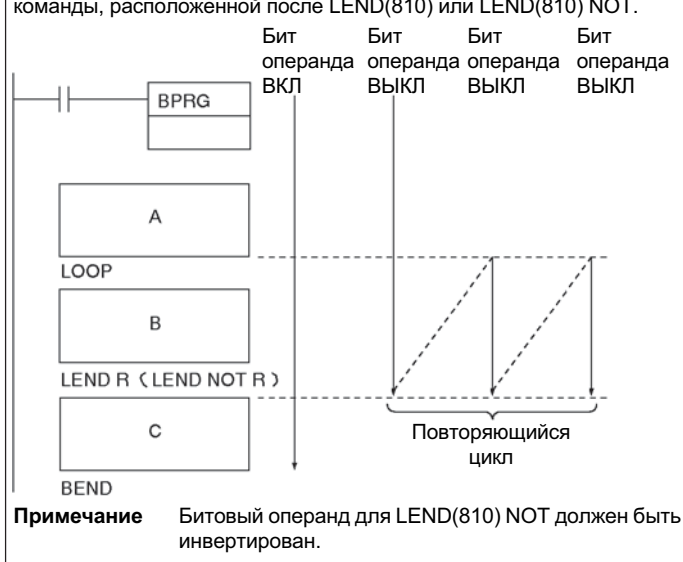
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>BLOCK PROGRAM RESTART</b>  BPRS 812	BPRS (812)   N: Номер программного блока	Приостанавливает и перезапускает указанный программный блок из другого программного блока.  	Программный блок Требуется
<b>CONDITIONAL BLOCK EXIT</b>  EXIT 806	EXIT(806) В: Битовый операнд	Команда EXIT(806) без битового операнда приводит к выходу из программы, если включено условие выполнения.  	Программный блок Требуется
<b>CONDITIONAL BLOCK EXIT</b>  EXIT 806	EXIT(806)В В: Битовый операнд	Команда EXIT(806) без битового операнда приводит к выходу из программы, если включено условие выполнения.  	Программный блок Требуется
<b>CONDITIONAL BLOCK EXIT NOT</b>  EXIT NOT 806	EXIT NOT(806) В В: Битовый операнд	Команда EXIT(806) без битового операнда приводит к выходу из программы, если условие выполнения выключено.  	Программный блок Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>CONDITIONAL BLOCK BRANCHING</b>  IF 802	IF (802)	<p>Если включено условие выполнения, выполняются команды, расположенные между IF(802) и ELSE(803), а если условие выполнения выключено, то выполняются команды между ELSE(803) и IEND(804).</p>  <pre>                     graph TD                         A[Условие выполнения = ВКЛ?] -- YES --&gt; B[Выполняется "А" (между IF и ELSE)]                         A -- NO --&gt; C[Выполняется "В" (после ELSE)]                         B --&gt; D[IEND]                         C --&gt; D                     </pre> <p>Условие выполнения</p> <p>IF</p> <p>A</p> <p>ELSE</p> <p>B</p> <p>IEND</p>	Программный блок Требуется
<b>CONDITIONAL BLOCK BRANCHING</b>  IF 802	IF (802) В В: Битовый операнд	<p>Если бит операнда установлен (ВКЛ), выполняются команды, расположенные между IF(802) и ELSE(803). Если бит операнда сброшен (ВЫКЛ), выполняются команды между ELSE(803) и IEND(804).</p>  <pre>                     graph TD                         A[Бит операнда = ВКЛ?] -- YES --&gt; B[Выполняется "А" (между IF и ELSE)]                         A -- NO --&gt; C[Выполняется "В" (после ELSE)]                         B --&gt; D[IEND]                         C --&gt; D                     </pre> <p>IF R (IF NOT R)</p> <p>A</p> <p>ELSE</p> <p>B</p> <p>IEND</p>	Программный блок Требуется
<b>CONDITIONAL BLOCK BRANCHING (NOT)</b>  IF NOT 802	IF (802) NOT В В: Битовый операнд	<p>Если бит операнда установлен (ВКЛ), будут выполнены команды, расположенные между IF(802) и ELSE(803), а если бит операнда сброшен (ВЫКЛ), то будут выполнены команды, расположенные между ELSE(803) и IEND(804).</p>	Программный блок Требуется
<b>CONDITIONAL BLOCK BRANCHING (ELSE)</b>  ELSE 803	---	<p>Если команда ELSE(803) отсутствует, а бит операнда установлен (ВКЛ), будут выполнены команды, расположенные между IF(802) и IEND(804).</p>	Программный блок Требуется
<b>CONDITIONAL BLOCK BRANCHING END</b>  IEND 804	---	<p>Если бит операнда сброшен (ВЫКЛ), будут выполнены только команды, расположенные после IEND(804).</p>	Программный блок Требуется

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>ONE CYCLE AND WAIT</b>  WAIT 805	WAIT(805)	<p>Если включено условие выполнения для команды WAIT(805), оставшиеся команды программного блока будут пропущены.</p> <p>Условие выполнения ВЫКЛ    Условие выполнения ВЫКЛ    Условие выполнения ВКЛ</p>	Программный блок Требуется
<b>ONE CYCLE AND WAIT</b>  WAIT 805	WAIT(805) В В: Битовый операнд	<p>Если бит операнда сброшен (ВЫКЛ) (ВКЛ для WAIT(805) NOT), оставшиеся команды программного блока будут пропущены. В следующем цикле ни одна из команд программного блока выполняться не будет, за исключением условия выполнения для команд WAIT(805) или WAIT(805) NOT. Когда условие выполнения перейдет во ВКЛ (ВЫКЛ для WAIT(805) NOT), будут выполнены команды, расположенные между WAIT(805) или WAIT(805) NOT и концом программы.</p>	Программный блок Требуется
<b>ONE CYCLE AND WAIT (NOT)</b>  WAIT NOT 805	WAIT(805) NOT В В: Битовый операнд	<p>Если бит операнда сброшен (ВЫКЛ) (ВКЛ для WAIT(805) NOT), оставшиеся команды программного блока будут пропущены. В следующем цикле ни одна из команд программного блока выполняться не будет, за исключением условия выполнения для команд WAIT(805) или WAIT(805) NOT. Когда условие выполнения перейдет во ВКЛ (ВЫКЛ для WAIT(805) NOT), будут выполнены команды, расположенные между WAIT(805) или WAIT(805) NOT и концом программы.</p>	Программный блок Требуется
<b>TIMER WAIT</b>  TIMW 813 (BCD)  TIMWX 816 (Двоичн.) (только CS1-H, CJ1-H, CJ1M или CS1D)	TIMW(813) N SV  N: Номер таймера SV: Время выдержки  TIMWX(816) N SV  N: Номер таймера SV: Время выдержки	<p>Задержка выполнения программного блока на указанное время. По истечении времени выдержки выполнение возобновится со следующей команды после TIMW(813)/TIMWX(816).</p> <p>SV: 0 ... 999.9 с (BCD) 0 ... 6 553.5 с (двоичн.)</p>	Программный блок Требуется

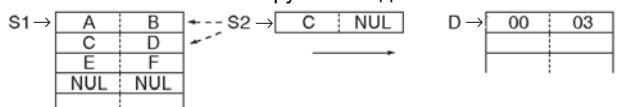
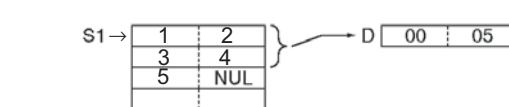
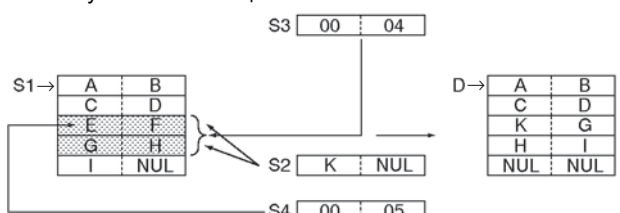
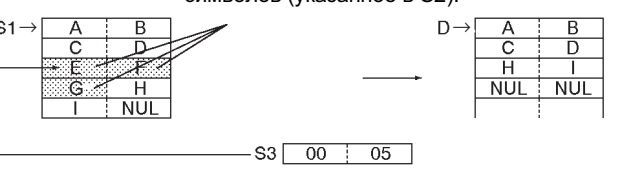
Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>COUNTER WAIT</b> CNTW 814 (BCD)  CNTWX 817 (Двоичн.) (только CS1-H, CJ1-H, CJ1M или CS1D)	CNTW(814) N SV  N: Номер счетчика SV: Задание счетчика I: Счетный вход  CNTWX(817) N SV  N: Номер счетчика SV: Задание счетчика I: Счетный вход	<p>Выполнение оставшейся части программного блока задерживается до тех пор, пока не будет достигнуто установленное значение счетчика. Выполнение будет продолжено со следующей команды, расположенной после CNTW(814)/SNTWX(817).</p> <p>SV: 0 ... 9999 (BCD)                      0 ... 65535 (двоичн.)</p>	Программный блок Требуется
<b>HIGH-SPEED TIMER WAIT</b> TMHW 815 (BCD)  TMHWX 818 (Двоичн.) (только CS1-H, CJ1-H, CJ1M или CS1D)	TMHW(815) N SV  N: Номер таймера SV: Время выдержки  TMHW(818) N SV  N: Номер таймера SV: Время выдержки	<p>Выполнение оставшейся части программного блока задерживается до тех пор, пока не истечет установленное время. По достижении заданного времени выполнение будет продолжено со следующей команды, расположенной после TMHW(815).</p> <p>SV: 0 ... 99.99 с (BCD)                      0 ... 655.35 с (двоичн.)</p>	Программный блок Требуется

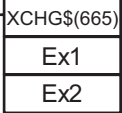
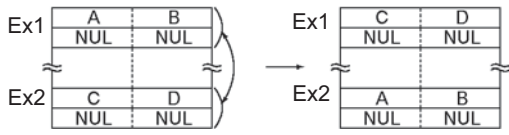
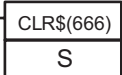

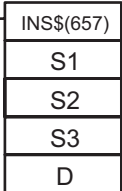
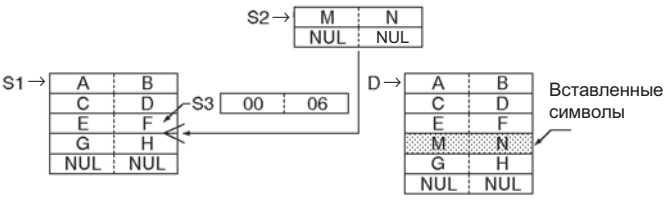
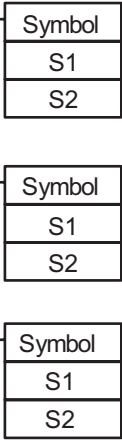


Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>LOOP</b>  LOOP 809	---	<p>Команда LOOP(809) указывает на начало программы, выполняющейся циклически.</p> <p>Условие выполнения ВКЛ    Условие выполнения ВЫКЛ    Условие выполнения ВЫКЛ    Условие выполнения ВЫКЛ</p> 	Программный блок Требуется
<b>LEND</b>  LEND 810	LEND (810)	<p>Команды LEND(810) или LEND(810) NOT указывают на завершение цикла. По достижению команды LEND(810) или LEND(810) NOT программа цикла начинает выполняться вновь, начиная с команды, расположенной после команды LEND(810), относящейся к этой циклической программе. Программа выполняется циклически до тех пор, пока для команд LEND(810) или LEND(810) NOT не будет установлен или сброшен (соответственно) бит операнда, или до тех пор, пока не будет установлено (ВКЛ) условие выполнения для LEND(810).</p>	Программный блок Требуется
<b>LEND</b>  LEND 810  В: Битовый операнд	LEND (810) В	<p>Если битовый операнд для LEND(810) сброшен (или установлен для LEND(810) NOT), программа выполняется циклически, начиная с команды, расположенной после LOOP(809). Когда устанавливается битовый операнд для LEND(810) (или сбрасывается для LEND(810) NOT), цикл завершается и выполнение продолжается со следующей команды, расположенной после LEND(810) или LEND(810) NOT.</p> <p>Бит операнда ВКЛ    Бит операнда ВЫКЛ    Бит операнда ВЫКЛ    Бит операнда ВЫКЛ</p>  <p><b>Примечание</b> Битовый операнд для LEND(810) NOT должен быть инвертирован.</p>	Программный блок Требуется
<b>LEND NOT</b>  LEND NOT 810	LEND(810) NOT В: Битовый операнд	<p>Команда LEND(810) или LEND(810) NOT указывает на завершение цикла. По достижению команды LEND(810) или LEND(810) NOT выполнение программы вновь возобновляется, начиная с команды, следующей за командой LOOP(809), принадлежащей этой циклической программе, до тех пор, пока бит операнда для LEND(810) или LEND(810) NOT не будет установлен или сброшен (соответственно), или до тех пор, пока не будет установлено условие выполнения для LEND(810).</p>	Программный блок Требуется

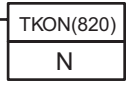
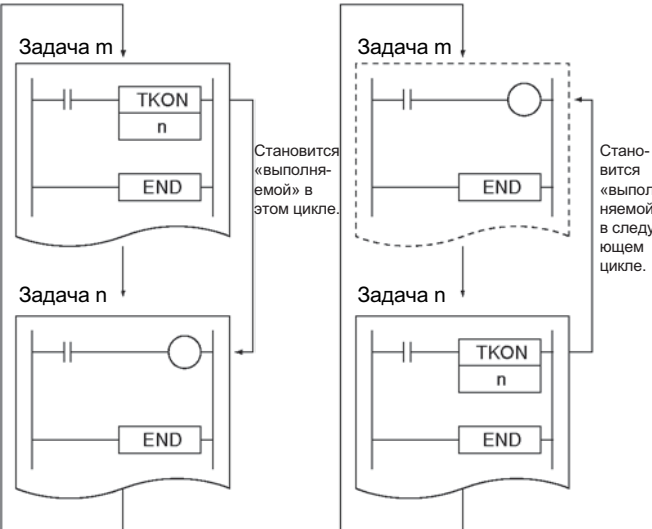
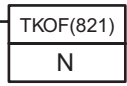
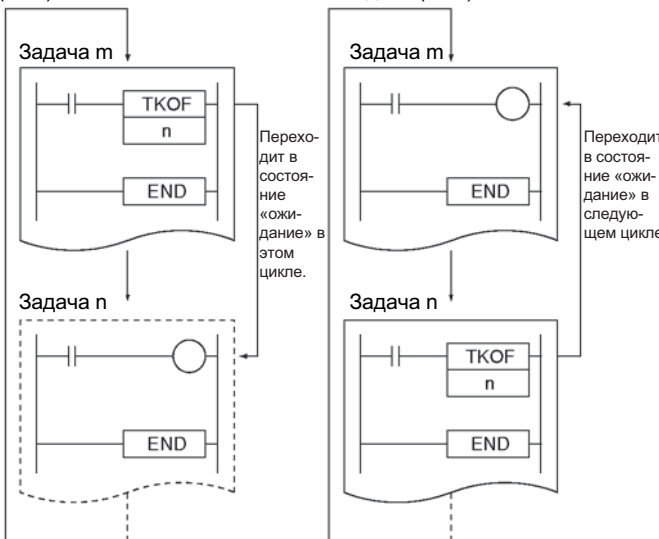
### 3-31 Команды для обработки текстовых строк

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения					
<b>MOV STRING</b> MOV\$ @MOV\$ 664	<table border="1"> <tr><td>MOV\$(664)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: 1-е входное слово D: 1-е адресуемое слово</p>	MOV\$(664)	S	D	<p>Передача текстовой строки.</p>	Выход Требуется		
MOV\$(664)								
S								
D								
<b>CONCATENATE STRING</b> +\$ @+\$ 656	<table border="1"> <tr><td>+(656)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: Текстовая строка 1 S2: Текстовая строка 2 D: 1-е адресуемое слово</p>	+(656)	S1	S2	D	<p>Объединение двух текстовых строк.</p>	Выход Требуется	
+(656)								
S1								
S2								
D								
<b>GET STRING LEFT</b> LEFT\$ @LEFT\$ 652	<table border="1"> <tr><td>LEFT\$(652)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: 1-е слово текстовой строки S2: Кол-во символов D: 1-е адресуемое слово</p>	LEFT\$(652)	S1	S2	D	<p>Считывание указанного количества символов текстовой строки, в направлении слева направо.</p>	Выход Требуется	
LEFT\$(652)								
S1								
S2								
D								
<b>GET STRING RIGHT</b> RGHT\$ @RGHT\$ 653	<table border="1"> <tr><td>RGHT\$(653)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: 1-е слово текстовой строки S2: Кол-во символов D: 1-е адресуемое слово</p>	RGHT\$(653)	S1	S2	D	<p>Считывание указанного количества символов текстовой строки, в направлении справа налево.</p>	Выход Требуется	
RGHT\$(653)								
S1								
S2								
D								
<b>GET STRING MIDDLE</b> MID\$ @MID\$ 654	<table border="1"> <tr><td>MID\$(654)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>S3</td></tr> <tr><td>D</td></tr> </table> <p>S1: 1-е слово текстовой строки S2: Кол-во символов S3: Начальная позиция D: 1-е адресуемое слово</p>	MID\$(654)	S1	S2	S3	D	<p>Чтение указанного количества символов, начиная с любой позиции в пределах текстовой строки.</p>	Выход Требуется
MID\$(654)								
S1								
S2								
S3								
D								

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения						
<b>FIND IN STRING</b> FIND @FIND\$ 660	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">FIND\$(660)</td></tr> <tr><td style="text-align: center;">S1</td></tr> <tr><td style="text-align: center;">S2</td></tr> <tr><td style="text-align: center;">D</td></tr> </table> <p>S1: 1-е слово входной текстовой строки                      S2: 1-е слово обнаруженной текстовой строки                      D: 1-е адресуемое слово</p>	FIND\$(660)	S1	S2	D	<p>Поиск указанной текстовой строки в пределах текстовой строки.</p> <p style="text-align: center;">Обнаруженные данные</p> 	Выход Требуется		
FIND\$(660)									
S1									
S2									
D									
<b>STRING LENGTH</b> LENS\$ @LENS\$ 650	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">LENS\$(650)</td></tr> <tr><td style="text-align: center;">S</td></tr> <tr><td style="text-align: center;">D</td></tr> </table> <p>S: 1-е слово текстовой строки                      D: 1-е адресуемое слово</p>	LENS\$(650)	S	D	<p>Вычисление длины текстовой строки.</p> 	Выход Требуется			
LENS\$(650)									
S									
D									
<b>REPLACE IN STRING</b> RPLC\$ @RPLC\$ 661	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">RPLC\$(664)</td></tr> <tr><td style="text-align: center;">S1</td></tr> <tr><td style="text-align: center;">S2</td></tr> <tr><td style="text-align: center;">S3</td></tr> <tr><td style="text-align: center;">S4</td></tr> <tr><td style="text-align: center;">D</td></tr> </table> <p>S1: 1-е слово текстовой строки                      S2: 1-е слово заменяющей текстовой строки                      S3: Количество символов                      S4: Начальная позиция                      D: 1-е адресуемое слово</p>	RPLC\$(664)	S1	S2	S3	S4	D	<p>Замещение текстовой строки указанной текстовой строкой, начиная с указанной позиции.</p> 	Выход Требуется
RPLC\$(664)									
S1									
S2									
S3									
S4									
D									
<b>DELETE STRING</b> DEL\$ @DEL\$ 658	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">DEL\$(658)</td></tr> <tr><td style="text-align: center;">S1</td></tr> <tr><td style="text-align: center;">S2</td></tr> <tr><td style="text-align: center;">S3</td></tr> <tr><td style="text-align: center;">D</td></tr> </table> <p>S1: 1-е слово текстовой строки                      S2: Количество символов                      S3: Начальная позиция                      D: 1-е адресуемое слово</p>	DEL\$(658)	S1	S2	S3	D	<p>Удаление указанной текстовой строки, начиная с указанной позиции в пределах текстовой строки.</p> <p style="text-align: center;">Количество удаляемых символов (указанное в S2).</p> 	Выход Требуется	
DEL\$(658)									
S1									
S2									
S3									
D									

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>EXCHANGE STRING</b> XCHG\$ @XCHG\$ 665	 <p>Ex1: 1-е слово 1-й строки                      Ex2: 1-е слово 2-й строки</p>	Замена указанной текстовой строки другой указанной текстовой строкой. 	Выход Требуется
<b>CLEAR STRING</b> CLR\$ @CLR\$ 666	 <p>S: 1-е слово текстовой строки</p>	Обнуление полностью всей текстовой строки (запись NUL = 00 hex). 	Выход Требуется
<b>INSERT INTO STRING</b> INS\$ @INS\$ 657	 <p>S1: 1-е слово основной текстовой строки                      S2: 1-е слово вставляемой текстовой строки                      S3: Начальная позиция                      D: 1-е адресуемое слово</p>	Вставка указанной текстовой строки в указанную позицию основной текстовой строки. 	Выход Требуется
<b>Сравнение строк</b> LD LD, AND, OR + =\$, <>\$, <\$, <=\$, >\$, >=\$ 670 (=\$) 671 (<>\$) 672 (<\$) 673 (<=\$) 674 (>\$) 675 (>=\$) AND OR	 <p>S1: Текстовая строка 1                      S2: Текстовая строка 2</p>	Команды сравнения строк (=\$, <>\$, <\$, <=\$, >\$, >=\$) служат для сравнения двух текстовых строк, начиная с первого символа. При этом сравниваются значения ASCII-кодов. В случае положительного результата сравнения (команды LOAD, AND или OR) включается условие выполнения.	LD: Не требуется AND, OR: Требуется

### 3-32 Команды управления задачами

Команда Мнемонический код	Символ/Операнд	Функция	Расположение и Условия выполнения
<b>TASK ON</b> TKON @TKON 820	 N: Номер задачи	<p>Переводит указанную задачу в состояние «выполняемая задача».</p> <p>Номер указанной задачи превышает номер локальной задачи (<math>m &lt; n</math>).                      Номер указанной задачи меньше номера локальной задачи (<math>m &gt; n</math>).</p> 	Выход Требуется
<b>TASK OFF</b> TKOF @TKOF 821	 N: Номер задачи	<p>Переводит указанную задачу в состояние «ожидание».</p> <p>Номер указанной задачи превышает номер локальной задачи (<math>m &lt; n</math>).                      Номер указанной задачи меньше номера локальной задачи (<math>m &gt; n</math>).</p> 	Выход Требуется

## РАЗДЕЛ 4

### Задачи

В этом разделе описаны функции задач.

4-1	Свойства задач. . . . .	156
4-1-1	Обзор. . . . .	156
4-1-2	Задачи и программы. . . . .	157
4-1-3	Основной принцип работы модуля CPU. . . . .	158
4-1-4	Типы задач. . . . .	160
4-1-5	Условия выполнения задач и настройка. . . . .	162
4-1-6	Возможные состояния циклической задачи. . . . .	163
4-1-7	Переключение состояний. . . . .	164
4-2	Применение задач. . . . .	165
4-2-1	Команды TASK ON и TASK OFF. . . . .	165
4-2-2	Ограничения на применение команд в задачах. . . . .	168
4-2-3	Специальные флаги задач. . . . .	169
4-2-4	Создание задач. . . . .	173
4-2-5	Глобальные подпрограммы. . . . .	174
4-3	Задачи обработки прерываний. . . . .	175
4-3-1	Типы задач обработки прерываний. . . . .	175
4-3-2	Приоритетность задач обработки прерываний. . . . .	182
4-3-3	Флаги и слова задач обработки прерываний. . . . .	183
4-3-4	Замечания по применению. . . . .	184
4-4	Использование средств программирования для задач. . . . .	187
4-4-1	Использование нескольких циклических задач. . . . .	187
4-4-2	Работа со средствами программирования. . . . .	187

## 4-1 Свойства задач

### 4-1-1 Обзор

Задачи управления, реализуемые с помощью ПЛК серии CS/CJ, можно классифицировать по функциям, по управляемому оборудованию, по технологическим процессам, по разработчикам или по любому другому критерию. Каждую такую задачу можно запрограммировать в виде отдельного модуля, который так и называется - "задача". Программирование в виде задач обладает следующими преимуществами:

1,2,3...

1. В создании программ могут участвовать одновременно несколько человек.

Созданные по отдельности части программы можно очень легко объединить в единую программу.

2. Можно создавать стандартные программные модули.

Ниже перечислены функции, предусмотренные для средств программирования, которые позволяют создавать программы, являющиеся автономными стандартными модулями, в отличие от узкоспециализированных программ, разработанных специально для определенных систем (машин, устройств). Это означает, что программы, разработанные по отдельности несколькими программистами, можно произвольно комбинировать, не изменяя текста программ.

- Программирование с использованием символов
- Глобальное и локальное обозначение символов
- Автоматическая привязка локальных символов к адресам

3. Повышается общее быстродействие.

Общее быстродействие повышается, поскольку теперь система состоит из общей управляющей программы и отдельных программ, причем выполняются только те программы, которые требуются в данный момент времени.

4. Облегчается проверка и отладка программ.

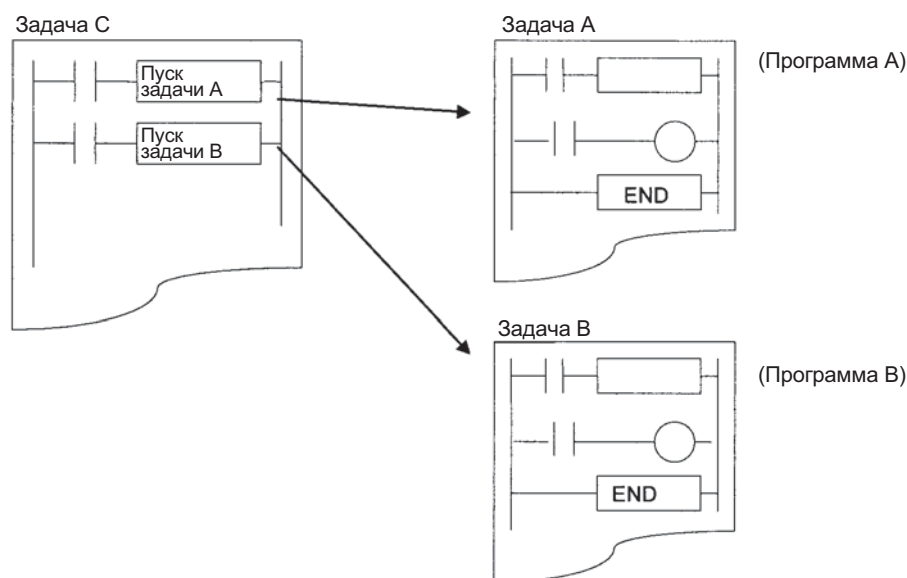
- Отладка программ становится менее трудоемкой, поскольку задачи разрабатываются отдельными программистами, и последующая проверка и отладка осуществляется отдельно для каждой задачи.
- Дальнейшая модернизация также упрощается, поскольку изменяться будет только та задача, которая требует пересмотра.
- Отладка упрощается еще и потому, что теперь легче определить, является адрес локальным или глобальным. Кроме того, адресное пространство, используемое программами, во время отладки требуется проверять только один раз, поскольку символы являются либо глобальными либо локальными, а локальные символы автоматически привязываются к адресам с помощью средств программирования.

5. Легко переключаться между программами.

Если в процессе управления технологическим процессом требуется перейти от одной задачи к другой, для переключения между задачами (программами) можно использовать команду управления задачами.

6. В программах пользователя легко разобраться.

Программы состоят из отдельных блоков, благодаря чему программы становятся более понятными и в них упрощается применение таких команд, как переходы.



## 4-1-2 Задачи и программы

Можно управлять максимум 288 программами (задачами). Одной задаче назначается одна программа. В широком смысле различают задачи следующих типов:

- Задачи, выполняемые циклически (циклические задачи)
- Задачи, выполняемые по прерыванию (задачи обработки прерываний)

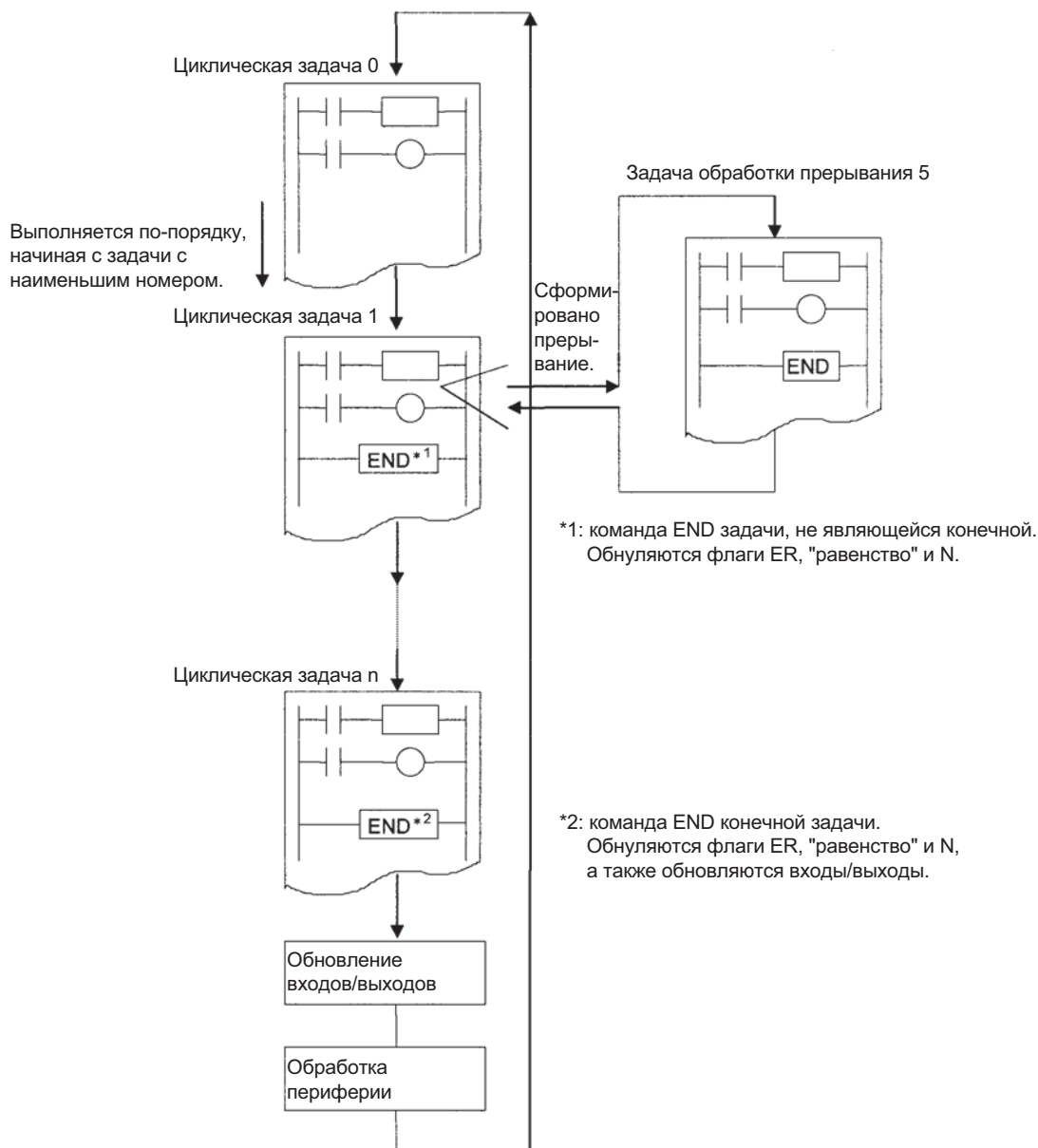
- Примечание**
1. Может быть создано до 32 циклических задач и до 256 задач, выполняемых по прерыванию, т.е., всего до 288 задач. Каждой задаче присваивается уникальный номер от 0 до 31 (для циклических задач) и от 0 до 255 (для задач, выполняемых по прерыванию).
  2. Модули CPU CS1-H, CJ1-H, CJ1M или CS1D позволяют выполнять задачи, выполняемые по прерыванию (задачи обработки прерываний 0 ... 255), как циклические задачи. Для этого эти задачи запускаются командой TKON. Такие задачи называют "дополнительными циклическими задачами". В случае применения дополнительных циклических задач суммарное количество циклических задач, которое может быть использовано, возрастает до 288.
  3. Модуль CPU CJ1 в настоящее время не поддерживает задачи, выполняемые по прерываниям от входов/выходов и по внешним прерываниям. Следовательно, максимальное количество задач для модуля CPU CJ1 составляет 35, а именно, 32 циклические задачи и 3 задачи, выполняемые по прерыванию. Таким образом, всего может быть создано 35 программ.

Каждая программа, назначенная задаче, должна завершаться командой END(001). Обновление входов/выходов происходит только после того, как выполняются все задачи текущего цикла.



### 4-1-3 Основной принцип работы модуля CPU

Модуль CPU выполняет циклические задачи (в том числе дополнительные циклические задачи, которые поддерживаются только модулями CPU CS1-H, CJ1-H, CJ1M или CS1D), начиная с задачи с наименьшим номером. Он также выполняет задачи обработки прерываний, если формируются соответствующие прерывания.



**Примечание** В начале выполнения задачи обнуляются все флаги условий (ER, CY, "равенство", AER и т.п.), а также условия выполнения команд (блокировка = ВКЛ и т.п.). Поэтому, во-первых, флаги условий не могут использоваться совместно разными задачами, а во-вторых, пары команд INTERLOCK/INTERLOCK CLEAR (IL/ILC), JUMP/JUMP END (JMP/JME) или SUBROUTINE CALL/SUBROUTINE ENTRY (SBS/SBN) нельзя разбивать, разнося их между двумя отдельными задачами.

Модули CPU CS1-H, CJ1-H, CJ1M или CS1D могут выполнять задачи обработки прерываний в качестве циклических задач, если эти задачи запускаются командой TCON. Это так называемые "дополнительные циклические задачи". Дополнительные циклические задачи (задачи, выполняемые по прерыванию, с номерами от 0 до 255) выполняются, начиная с задачи с наименьшим номером, после того, как завершается выполнение обычных циклических задач (циклических задач с номерами от 0 до 31).



## 4-1-4 Типы задач

В широком смысле задачи подразделяются на циклические задачи и задачи, выполняемые по прерыванию. Выполняемые по прерыванию задачи, в свою очередь, подразделяются на задачи, выполняемые по прерыванию от выключения питания, выполняемые по запланированным прерываниям, выполняемые по прерываниям от входов/выходов (только серия CS) и выполняемые по внешним прерываниям (только серия CS). Кроме того, задачи, выполняемые по прерыванию, могут выполняться как дополнительные циклические задачи.

**Примечание** Модули CS1-H, CJ1-H, CJ1M или CS1D могут выполнять задачи, выполняемые по прерыванию, как циклические задачи, если их выполнение начинается командой TCON. Это так называемые "дополнительные циклические задачи".

### Циклические задачи

Циклические задачи, имеющие статус READY (готовность) выполняются один раз в каждом цикле (начиная с вершины программы до команды END(001)) в порядке возрастания номеров задач, начиная с задачи с наименьшим номером. Максимальное количество циклических задач - 32 (номера циклических задач: 00 ... 31).

**Примечание** В модулях CPU CS1-H, CJ1-H, CJ1M или CS1D предусмотрена возможность выполнения задач по прерыванию (задач обработки прерываний с номерами 0 ... 255) в качестве циклических задач, точно так же, как выполняются обычные циклические задачи (задачи с номерами 0 ... 31). В случае применения дополнительных циклических задач суммарное количество циклических задач возрастает до 288.

### Задачи, выполняемые по прерыванию

Задачи обработки прерываний выполняются, если сформировано прерывание, даже если в настоящий момент выполняется циклическое задание (в том числе дополнительная циклическая задача). Задача, выполняемая по прерыванию, может быть выполнена в цикле в любое время, в том числе во время выполнения программы пользователя, во время обновления входов/выходов или обслуживания периферии, если выполнено условие формирования прерывания.

В модулях CPU CS1-H, CJ1-H, CJ1M или CS1D предусмотрена возможность выполнения задач обработки прерываний в качестве циклических задач (модули CPU CS1D для систем с дублированием CPU не поддерживают прерывания. В этих модулях задачи, выполняемые по прерыванию, можно использовать только в качестве дополнительных циклических задач).

В модуле CPU CJ1M для вызова задач обработки прерываний можно использовать встроенные входы прерываний и входы высокоскоростных счетчиков. Подробное описание смотрите в руководстве *CJ Series Built-in I/O Operation Manual*.

**Задача обработки прерывания от выключения питания** Задача обработки прерывания от выключения питания выполняется в том случае, когда выключается напряжение питания модуля CPU. Можно запрограммировать только одну задачу обработки прерывания от выключения питания (номер задачи обработки прерывания: 1).

**Примечание** Задача обработки прерывания от выключения питания должна быть завершена, прежде чем истечет указанное ниже время, либо она будет завершена принудительно:

10 мс - (время задержки обнаружения выключения питания)

**Задачи, выполняемые по запланированным прерываниям** Задача обработки запланированного прерывания выполняется через установленные интервалы времени, длительность которых определяется внутренним таймером модуля CPU. Может быть запрограммировано не более двух задач обработки запланированных прерываний (номера задач обработки прерываний: 2 и 3).

**Примечание** Прерывание для задач обработки запланированных прерываний устанавливается командой SET INTERRUPT MASK (MSKS(690)). Периодичность прерываний можно устанавливать с шагом 10 мс или 1.0 мс в настройках ПЛК.

**Задачи, выполняемые по прерываниям от входов/выходов** Задача обработки прерывания от входов/выходов выполняется, если включается вход модуля ввода прерываний. Может быть запрограммировано до 32 задач обработки прерываний от входов/выходов (номера задач обработки прерываний: 100 ... 131). В стойку CPU должен быть установлен модуль ввода прерываний. В ПЛК CJ1H такой модуль должен быть установлен как один из пяти модулей, расположенных

после модуля CPU (слоты 0 ... 4). В ПЛК серии CJ1M этот модуль должен быть установлен как один из трех модулей после модуля CPU (слоты 0 ... 2). Модули ввода/вывода прерываний, установленные в другие слоты, нельзя использовать для вызова задач обработки прерываний от входов/выходов.

Прерывания от входов/выходов не поддерживаются модулями CPU CJ1.

#### Задачи, выполняемые по внешним прерываниям

Задачи обработки внешних прерываний выполняются по запросу от пользовательской программы специального модуля ввода/вывода, модуля шины CPU или встраиваемой платы (только серия CS). При этом специальные модули ввода/вывода и модули шины CPU должны быть установлены в стойку CPU. В ПЛК CJ1H такой модуль должен быть установлен как один из пяти модулей после модуля CPU (слоты 0 ... 4). В ПЛК серии CJ1M такой модуль должен быть установлен как один из трех модулей после модуля CPU (слоты 0 ... 2). Модули, установленные в другие слоты, нельзя использовать для формирования внешних прерываний.

Может быть запрограммировано до 256 задач обработки внешних прерываний (номера задач обработки прерываний 0 ... 255). Если задача обработки внешнего прерывания имеет тот же номер, что и задача обработки прерывания от выключения питания, задача обработки запланированного прерывания или задача обработки прерываний от входов/выходов, в этом случае задача обработки прерывания будет выполнена по любому из этих прерываний (два условия объединяются в логическое ИЛИ), но в общем случае не следует дублировать номера задач обработки прерываний.

Прерывания от входов/выходов не поддерживаются модулями CPU CJ1.

#### Дополнительные циклические задачи (только для модулей CPU CS1-H, CJ1-H, CJ1M или CS1D)

Задачи обработки прерываний могут выполняться в каждом цикле, так же, как и обычные циклические задачи. Дополнительные циклические задачи (задачи обработки прерываний с номерами 0 ... 255) выполняются, начиная с задачи с наименьшим номером, после завершения выполнения обычных циклических задач (циклические задачи с номерами 0 ... 31). Максимальное количество дополнительных циклических задач составляет 256 (номера задач обработки прерываний: 0 ... 255). В то же время, циклические задачи обработки прерываний отличаются от обычных циклических задач в том, что они запускаются командой TKON (820). В пределах дополнительных циклических задач нельзя применять команды TKON (820) и TKOF, то есть, из дополнительной циклической задачи нельзя запускать обычные циклические задачи и другие дополнительные циклической задачи.

Если дополнительной циклической задаче присвоен тот же номер, что и задаче обработки прерывания от включения питания, обработки запланированного прерывания или прерывания от входов/выходов, задача обработки прерывания будет выполнена по любому из этих прерываний (два условия объединяются в логическое ИЛИ). Не следует использовать задачи обработки прерываний одновременно в качестве обычных задач обработки прерываний и в качестве дополнительных циклических задач.

#### Примечание

1. Задача обработки прерывания от выключения питания (см. п. 1 выше) обладает приоритетом и будет выполнена, когда выключается питание, даже если в данный момент выполняется другая задача обработки прерывания.
2. Если выполняется задача обработки прерывания и в этот момент происходит запланированное прерывание, прерывание от входов/выходов или внешнее прерывание, перечисленные задачи обработки прерываний не будут выполняться до тех пор, пока не будет завершена текущая задача обработки прерывания. Если несколько прерываний формируется одновременно, в этом случае задачи обработки прерываний выполняются последовательно, начиная с задачи обработки с наименьшим номером.
3. Различия между обычными циклическими задачами и дополнительными циклическими задачами пояснены в следующей таблице.

Параметр	Дополнительные циклические задачи	Обычные циклические задачи
Вызов при запуске	Настроить нельзя	Настраиваются в CX-Programmer
Использование команды TKON/TKOF	Невозможно	Возможно
Флаги задач	Не поддерживаются	Поддерживаются (циклические задачи 00 ... 31 соответствуют флагам задач TK00 ... TK31).

Параметр	Дополнительные циклические задачи	Обычные циклические задачи
Флаг "Первое выполнение задачи" (A20015) и флаг "Запуск задачи" (A20014)	Не поддерживаются.	Поддерживаются.
Значения регистров индексов (IR) и регистров данных (TR)	Не определены в начале задачи (как и у обычных задач обработки прерываний). Значения в начале каждого цикла не определены. Перед использованием регистров в них всегда следует записывать значения. Значения, установленные в предыдущем цикле, считаны быть не могут.	Не определены в начале работы. Значения, установленные в предшествующем цикле, могут быть прочитаны.

4. Модули CPU CJ1 не поддерживают задачи обработки прерываний от входов/выходов и внешних прерываний.

#### 4-1-5 Условия выполнения задач и настройка

В следующей таблице описаны условия выполнения задач, сопутствующие параметры и состояния.

Задача		Номер	Условие выполнения	Сопутствующие параметры
Циклические задачи		0 ... 31	Выполняются один раз в каждом цикле, если имеют статус READY (запускаются в начале работы или командой TKON (820)), когда получают право на выполнение.	Нет
Задачи обработки прерываний	Задачи обработки прерываний от выключения питания	Задача обработки прерываний 1	Выполняются, когда выключается питание модуля CPU.	<ul style="list-style-type: none"> <li>В настройках ПЛК разрешено прерывание от выключения питания.</li> </ul>
	Задачи обработки запланированных прерываний 0 и 1	Задачи обработки прерываний 2 и 3	Выполняются один раз через установленные интервалы времени, отсчитываемые внутренним таймером модуля CPU.	<ul style="list-style-type: none"> <li>С помощью команды SET INTERRUPT MASK (MSKS) установлено время формирования запланированного прерывания (0 ... 9999).</li> <li>В настройках ПЛК выбраны единицы измерения времени для формирования запланированного прерывания (шаг 10 мс или 1.0 мс).</li> </ul>
	Задачи обработки прерываний от входов/выходов 00 ... 31	Задачи обработки прерываний 100 ... 131	Выполняются, когда включается вход модуля ввода прерываний в стойке CPU.	<ul style="list-style-type: none"> <li>С помощью команды SET INTERRUPT MASK (MSKS) отменены маски для назначенных входов.</li> </ul>
	Задачи обработки внешних прерываний 0 ... 255	Задачи обработки прерываний 0 ... 255	Выполняются по запросу программы пользователя в специальном модуле ввода/вывода и модуле шины CPU в стойке CPU или по запросу программы пользователя во встраиваемой плате (только серия CS).	Нет (всегда разрешены)
Дополнительные циклические задачи (только для модулей CPU CS1-H, CJ1-H, CJ1M или CS1D)		Задачи обработки прерываний 0 ... 255	Выполняются один раз в каждом цикле, если имеют статус READY (запускаются командой TKON (820)), когда получают право на выполнение.	Нет (всегда разрешены)

- Примечание**
1. В стойку CPU должен быть установлен модуль ввода прерываний. В ПЛК CJ1-H модуль должен быть установлен как один из пяти модулей после модуля CPU (слоты 0 ... 4). В ПЛК CJ1M модуль должен быть установлен в качестве одного из трех модулей после модуля CPU (слоты 0 ... 2). Модули ввода/вывода прерываний, установленные в другие слоты, не могут вызывать задачи прерывания от входов/выходов.
  2. В стойку CPU должен быть установлен специальный модуль ввода/вывода или модуль шины CPU. В ПЛК CJ1-H такой модуль должен быть установлен как один из пяти модулей после модуля CPU (слоты 0 ... 4).

В ПЛК CJ1M такой модуль должен быть установлен как один из трех модулей после модуля CPU (слоты 0 ... 2). Модули, установленные в другие слоты, нельзя использовать для формирования внешних прерываний.

3. Если с помощью консоли программирования выполняется операция обнуления памяти, в этом случае количество циклических задач и задач обработки прерываний ограничено.
  - Может быть создана только циклическая задача 0.  
С помощью консоли программирования нельзя создать циклические задачи 1 ... 31, но уже созданные с помощью CX-Programmer циклические задачи редактировать можно.
  - Могут быть созданы только задачи обработки прерываний 1, 2, 3 и 100 ... 131 (только серия CS).  
Задачи обработки прерываний 0, 4 ... 99 и 132 ... 255 нельзя создать с помощью консоли программирования (за исключением задач 140 ... 143, которые можно создать для модулей CPU CJ1M), но если такие задачи были созданы с помощью CX-Programmer, их можно редактировать.

#### 4-1-6 Возможные состояния циклической задачи

В настоящем разделе описано понятие статуса (состояния) циклических задач, в том числе дополнительных циклических задач (поддерживаемых только модулями CPU CS1-H, CJ1-H, CJ1M или CS1D).

Циклические задачи всегда находятся в одном из четырех следующих состояний: задача отключена, READY (задача готова), RUN (выполняемая задача) и ожидание (WAIT).

##### Состояние "отключенная задача" (INI)

Задача в состоянии "отключенная задача" не выполняется. В режиме PROGRAM все циклические задачи имеют статус "отключенная задача". Любая циклическая задача, переведенная из этого состояния в другое, не может возвратиться к этому состоянию без перехода в режим PROGRAM.

##### Состояние READY (готовность)

С помощью атрибута задачи можно выбрать возможность управления моментом перехода задачи в состояние READY (готовность). С помощью атрибута можно выбрать активизацию задачи командой TASK ON или активизацию по запуску режима RUN.

##### **Задачи, активизируемые командами**

Для перевода циклической задачи, которая активизируется командой, из статуса "отключенная задача" или "ожидание" в статус READY (готовность) предназначена команда TASK ON (TKON(820)).

##### **Задачи, активизируемые при переходе в рабочее состояние**

Циклическая задача, запускаемая по переходу в режим работы, переходит из статуса "Отключенная задача" в статус READY (готовность), когда ПЛК переключается из режима PROGRAM в режим RUN или MONITOR. Это относится только к обычным циклическим задачам.

**Примечание** С помощью средства программирования можно выбрать одну или несколько задач, которые будут переходить в состояние READY, когда начинается работа. Это можно сделать для задач с номерами 0 ... 31. Эта возможность недоступна для дополнительных циклических задач.

##### Состояние RUN

Циклическая задача в состоянии READY переключится в состояние RUN и будет выполнена, когда она получит право на выполнение.

##### Состояние "Ожидание"

Для переключения циклической задачи из состояния "отключенная задача" в состояние "ожидание" предназначена команда TASK OFF (TKOF(821)).

**Примечание** С помощью CX-Programmer версии 4.0 или более новой версии программы, назначенные задачам для ПЛК серии CS/CJ, можно контролировать в режиме online, что позволяет следить за их состоянием (выполняются они или остановлены). Состояния индицируются в среде CX-Programmer следующим образом:

- Выполнение: задача в состоянии READY или RUN (различить эти состояния невозможно).
- Задача прекращена: задача в состоянии INI или WAIT (различить эти состояния невозможно).

### 4-1-7 Переключение состояний



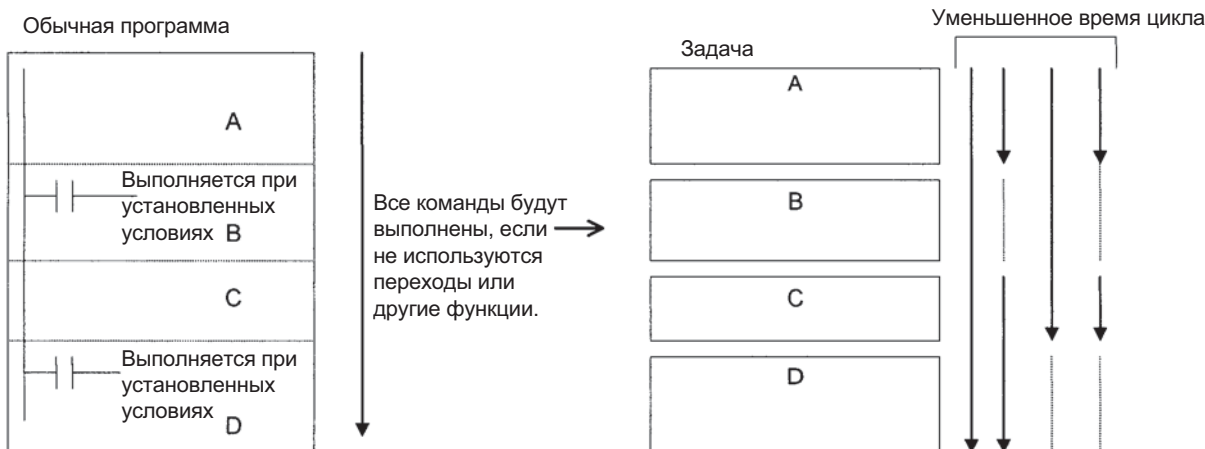
- Примечание**
1. Задача, находящаяся в состоянии RUN, будет переведена в состояние "ожидание" командой TKOF(821), даже если команда TKOF(821) выполняется в пределах этой задачи.
  2. Активизация в начале работы возможна только для обычных циклических задач. Эта возможность отсутствует для дополнительных циклических задач.

Состояние "ожидание" действует аналогично команде перехода (JMP-JME). Задача в состоянии "ожидание" сохраняет состояние своих выходов.



В состоянии "ожидание" команды не выполняются, поэтому общее время выполнения команд не возрастает. Задачи могут содержать блоки программ, которые не требуется выполнять все время.

Чтобы уменьшить длительность цикла, таким блокам можно присваивать статус "ожидание".



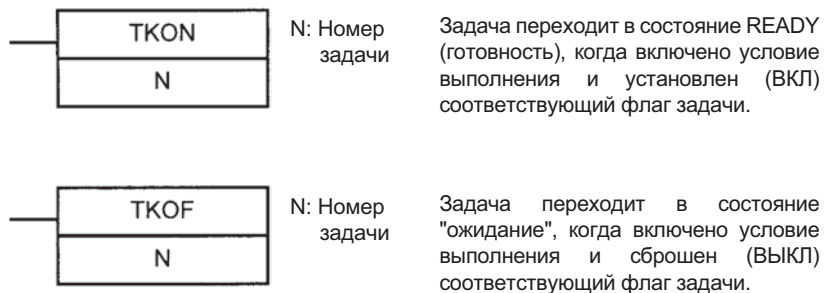
**Примечание** Статус "ожидание" означает, что задача будет пропущена при выполнении. Переход в состояние "ожидание" не означает завершение программы.

## 4-2 Применение задач

### 4-2-1 Команды TASK ON и TASK OFF

Посредством команд TASK ON (TKON(820)) и TASK OFF (TKOF(821)) циклические задачи (в том числе дополнительные циклические задачи) переключаются программно между состояниями READY (готовность) и "ожидание".

**Примечание** Дополнительные циклические задачи поддерживаются только модулями CPU CS1-N, CJ1-N, CJ1M или CS1D.



Примечание: флаги задач не работают для дополнительных циклических задач.

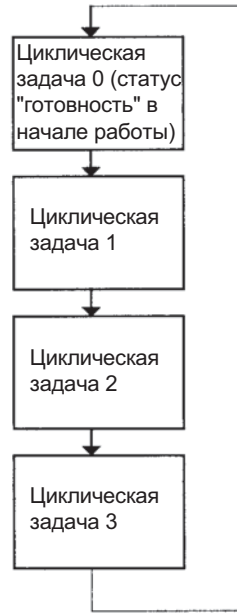
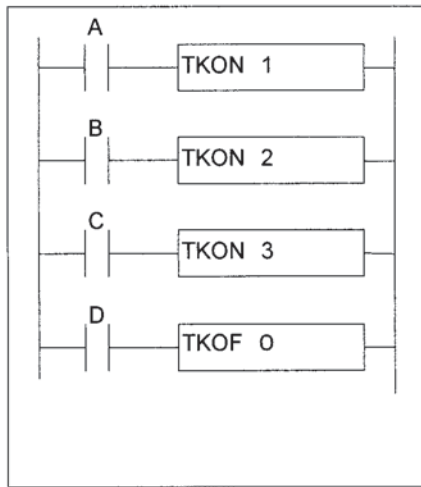
Команды TASK ON и TASK OFF можно использовать в любое время для переключения любой циклической задачи между состояниями "готовность" или "ожидание". Циклическая задача, находящаяся в состоянии "готовность", остается в последующих циклах в состоянии "готовность", а циклическая задача, находящаяся в состоянии "ожидание", остается в последующих циклах в состоянии "ожидание".

Команды TASK ON и TASK OFF можно использовать только в циклических задачах, но не в задачах обработки прерываний.

**Примечание** В каждом цикле в состоянии "готовность" должна находиться хотя бы одна циклическая задача. Если ни одна задача не находится в состоянии "готовность", в этом случае устанавливается флаг "ошибка задачи" (A29512) и модуль CPU прекращает работу.

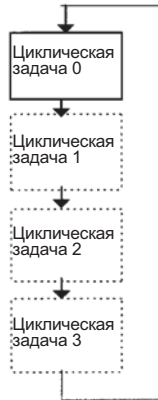


Пример: циклическая задача

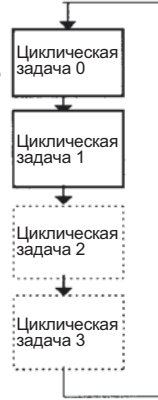


1) Задача 0 находится в состоянии "готовность" в начале работы.

Остальные задачи остаются в состоянии "отключенная задача".

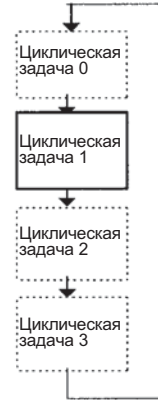




2) Задача 1 перейдет в состояние "готовность", если A=ВКЛ, а задачи 2 и 3 остаются в состоянии "отключенные задачи".



3) Задача 0 перейдет в состояние "ожидание", если D=ВКЛ.

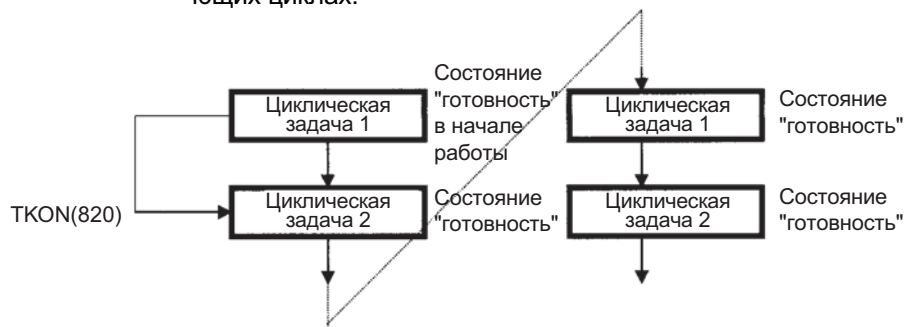
Остальные задачи остаются в своих текущих состояниях.



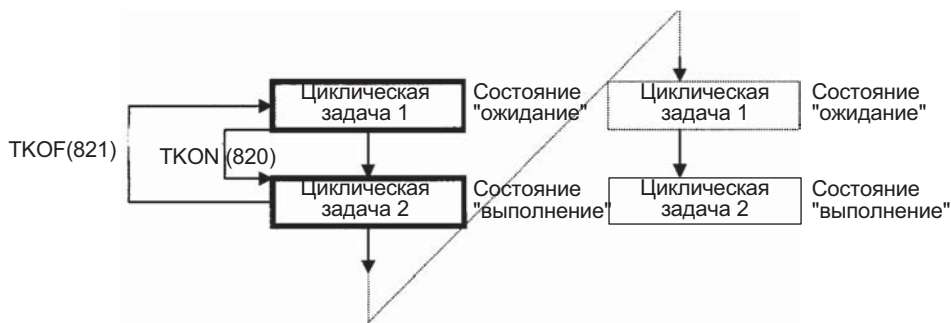
 Состояние "готовность"  
 Состояние "ожидание"/ состояние "задача отключена"

### Задачи и цикл выполнения

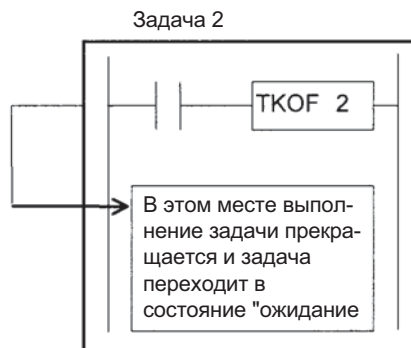
Циклическая задача (в том числе дополнительная циклическая задача), находящаяся в состоянии "готовность", сохранит свое состояние в последующих циклах.



Циклическая задача, которая находится в состоянии "ожидание", сохранит свое состояние в последующих циклах. Для перевода задачи из состояния "ожидание" в "готовность" следует использовать команду TKON(820).



Задача, внутри которой выполняется команда TKOF(821), прекращает выполняться в том месте, в котором была выполнена эта команда, и переходит в состояние "ожидание".



**Номера циклических задач и цикл выполнения (включая дополнительные циклические задачи)**

Если задача m включает задачу n и  $m > n$ , задача n переходит в состояние "готовность" в следующем цикле.

**Пример:** если задача 5 включает задачу 2, задача 2 переходит в состояние "готовность" в следующем цикле.

Если задача m включает задачу n и  $m < n$ , задача n переходит в состояние "готовность" в том же цикле.

**Пример:** если задача 2 включает задачу 5, задача 5 переходит в состояние "готовность" в том же цикле.

Если задача m переводит задачу n в состояние "ожидание" и  $m > n$ , задача перейдет в "ожидание" в следующем цикле.

**Пример:** если задача 5 переводит задачу 2 в состояние "ожидание", задача 2 переходит в состояние "ожидание" в следующем цикле.

Если задача m переводит задачу n в состояние "ожидание" и  $m < n$ , задача n переходит в "ожидание" в том же цикле.

**Пример:** если задача 2 переводит задачу 5 в состояние "ожидание", задача 5 переходит в "ожидание" в том же цикле.

**Взаимосвязь задач с памятью ввода/вывода**

Имеется два различных способа применения регистров индексов (IR) и регистров данных (DR): 1) Отдельно каждой задачей или 2) Совместно всеми задачами (поддерживается только модулями CPU CS1-H, CJ1-H, CJ1M или CS1D).

В случае независимого использования регистров, регистр IR0, используемый, например, циклической задачей 1, отличается от регистра IR0, используемого циклической задачей 2. В случае совместного использования регистров, регистр IR0, используемый циклической задачей 1, совпадает с регистром IR0, используемым циклической задачей 2.

Раздельное или совместное использование регистров выбирается с помощью CX-Programmer.

- Остальные слова и биты в памяти ввода/вывода используются совместно всеми задачами. Например, бит CIO 001000 - это один и тот же бит как для циклической задачи 1, так и для циклической задачи 2. Следовательно, всякий раз, когда при программировании используется какая-либо из областей памяти ввода/вывода, кроме областей IR и DR, следует быть очень внимательным, поскольку изменение значений в одной задаче влияет и на другие задачи.

Память ввода/вывода	Взаимосвязь с задачами
CIO, дополнительная память, память данных и другие области памяти, за исключением областей IR и DR (см. прим. 1).	Используется совместно всеми задачами.
Регистры индексов (IR) и регистры данных (DR) (см. прим. 2).	Используется отдельно каждой задачей.

- Примечание**
1. Текущий банк EM также используется совместно всеми задачами. Следовательно, если, например, циклическая задача 1 изменяет номер текущего банка EM, новый номер текущего банка EM будет действителен также и для циклической задачи 2.
  2. Значения IR и DR не определены, когда начинают выполняться задачи обработки прерываний (включая дополнительные циклические задачи). Если регистры IR и DR используются в задачах обработки прерываний, в них следует записывать значения с помощью команд MOVR/MOVRW (MOVE TO REGISTER и MOVE TIMER/COUNTER TO REGISTER) внутри задач обработки прерываний. После выполнения задачи обработки прерывания в регистры IR и DR возвращаются значения, которые находились в них до выполнения задачи обработки прерывания. Это происходит автоматически.

**Взаимосвязь задач с работой таймера** Текущие значения таймеров TIM, TIMX, TIMH, TIMHX, TMHH, TMHXX, TIMW, TIMWX, TMHW и TMHWX, запрограммированные для таймеров с номерами 0000 ... 2047, будут обновлены, даже если происходит переключение задачи или задача, содержащая таймер, переходит в состояние "ожидание" или назад в состояние "готовность".

Если задача, которая содержит TIM, переходит в состояние "ожидание", а затем возвращается в состояние "готовность", будет установлен (ВКЛ) флаг завершения, если команда была выполнена при нулевом (0) текущем значении (флаги завершения для таймера обновляются только после выполнения команд). Если команда TIM выполняется при ненулевом текущем значении, текущее значение по-прежнему будет обновляться, как если бы задача находилась в состоянии "готовность".

- Текущие значения таймеров, запрограммированные для таймеров с номерами 2048 ... 4098, сохраняют свои значения, когда задача находится в состоянии "ожидание".

**Взаимосвязь задач с флагами условий** Перед выполнением каждой задачи все флаги условий обнуляются. Следовательно, состояние флага условия, наблюдавшееся в конце задачи 1, нельзя прочитать в задаче 2. Однако у модулей CPU CS1-H, CJ1-H, CJ1M или CS1D для чтения состояний флагов условий из другого участка программы, например, из другой задачи, можно использовать команды CCS(282) и CCL(283).

- Примечание** Когда состояние флагов условий контролируется с помощью консоли программирования, консоль программирования показывает статус флага, соответствующий концу цикла, то есть, статус флага в конце последней задачи цикла.

## 4-2-2 Ограничения на применение команд в задачах

### Команды, которые должны присутствовать в той же задаче

Ниже перечислены команды, которые должны размещаться в пределах одной и той же задачи. Если такие команды оказываются в двух различных задачах, устанавливается флаг ER и команды не выполняются.

Мнемонический код	Команда
JMP/JME	JUMP/JUMP END
CJP/JME	CONDITIONAL JUMP/JUMP END
CJPN/JME	CONDITIONAL JUMP NOT/CONDITIONAL JUMP END
JMP0/JME0	MULTIPLE JUMP/JUMP END

Мнемонический код	Команда
FOR/NEXT	FOR/NEXT
IL/ILC	INTERLOCK/INTERLOCK CLEAR
SBS/SBN/RET	SUBROUTINE CALL/SUBROUTINE ENTRY/SUBROUTINE RETURN
MCRO/SBN/RET	MACRO/SUBROUTINE ENTRY/SUBROUTINE RETURN
BPRG/BEND	BLOCK PROGRAM BEGIN/BLOCK PROGRAM END
STEP S/STEP	STEP DEFINE

**Команды, которые нельзя применять в задачах обработки прерываний**

Ниже перечислены команды, которые нельзя размещать в задачах обработки прерываний. Любая попытка выполнения такой команды в задаче обработки прерывания приведет к включению флага ER, а команда выполнена не будет. Эти команды можно использовать в задаче обработки прерывания, только если она используется в качестве дополнительной задачи.

Мнемонический код	Команда
TKON(820)	TKON(820)
TKOF(821)	TASK OFF
STEP	STEP DEFINE
SNXT	STEP NEXT
STUP	CHANGE SERIAL PORT SETUP
DI	DISABLE INTERRUPT
EI	ENABLE INTERRUPT

Выполнение следующих команд непредсказуемо в случае выполнения их внутри задач обработки прерываний: TIMER: TIM и TIMX((550), HIGH-SPEED TIMER: TIMH(015) и TIMHX(551), ONE-MS TIMER: TMHH(540) и TMHHX(552), ACCUMULATIVE TIMER: TTIM(087) и TTIMX(555), MULTIPLE OUTPUT TIMER: MTIM(543) и MTIMX(554), LONG TIMER: TIML(542) и TIMLX(553), TIMER WAIT: TIMW(813) и TIMWX(816), HIGH-SPEED TIMER WAIT: TMHW(815) и TMHWX(817), PID CONT- ROL: PID(190), FAILURE POINT DETECTION: FPD(269) и CHANGE SERIAL PORT SETUP: STUP(237).

Следующие команды нельзя использовать внутри задачи обработки прерывания от выключения питания (они выполнены не будут, даже если они включены в программу, при этом флаг "ошибка" установлен **не будет**):  
 READ DATA FILE: FREAD(700), WRITE DATA FILE: FWRT(701), NETWORK SEND: SEND(090), NETWORK RECEIVE: RECV(098), DELIVER COMMAND: CMND(490), PROTOCOL MACRO: PMCR(260).

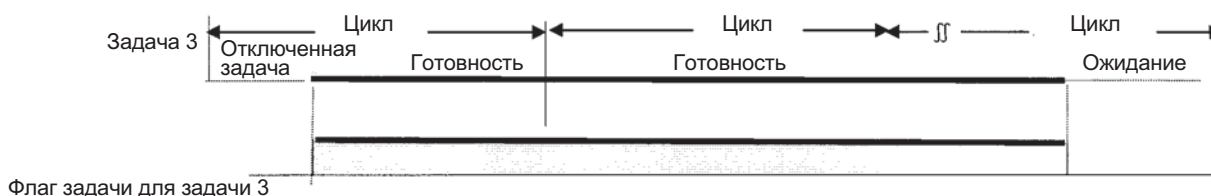
**4-2-3 Специальные флаги задач**

**Флаги, связанные с циклическими задачами**

Описанные ниже флаги работают только с обычными циклическими задачами. С дополнительными циклическими задачами они не работают.

**Флаги задач (TK00 ... TK31)**

Флаг задачи включается, когда циклическая задача находится в состоянии "готовность" (READY), и выключается, когда задача переходит в состояние "отключенная задача" (INI) или в состояние "ожидание" (WAIT). Номера задач 00 ... 31 соответствуют флагам задач TK00 ... TK31.



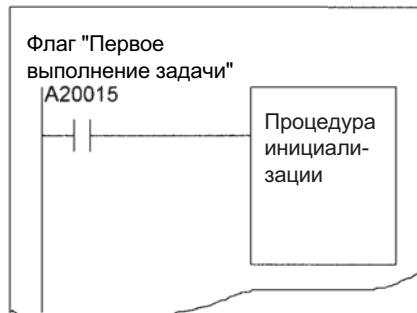
**Примечание** Флаги задач используются только с циклическими задачами и не используются с задачами обработки прерываний. Если после начала работы выполняется задача обработки прерывания, устанавливается бит A44115, а в A44100 ... A44107 (2 разряда, hex) записывается номер задачи обработки прерывания, выполнение которой заняло наибольшее время.

**Флаг "Первое выполнение задачи" (A20015)**

Флаг "Первое выполнение задачи" устанавливается, когда циклические задачи переходят из состояния "отключенная задача" (INI) в состояние "готовность", задачи получают право на выполнение и выполняются впервые. Флаг сбрасывается, когда первое выполнение задач завершено.



Флаг "Первое выполнение задачи" информирует о том, что циклические задачи выполняются впервые. Следовательно, этот флаг можно использовать для выполнения процедуры инициализации в задачах.



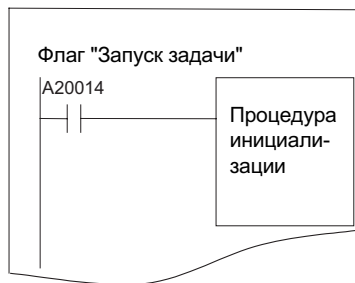
**Примечание** Даже если циклическая задача, находящаяся в режиме "ожидание", возвращается в состояние "готовность" с помощью команды TKON(820), это не считается первым выполнением и флаг "Первое выполнение задачи" (20015) не установится. Флаг "Первое выполнение задачи" (20015) также не будет установлен, если циклическая задача переводится из состояния "отключенная задача" в состояние "выполнение", или если она переводится в состояние "ожидание" другой задачей посредством команды TKOF(821), прежде чем фактически получено право на выполнение.

**Флаг "Запуск задачи" (A20014) (только для модулей CPU CS1-H, CJ1-H, CJ1M или CS1D)**

Когда запускается циклическая задача, для выполнения процедуры инициализации можно использовать флаг "Запуск задачи". Флаг "Запуск задачи" сбрасывается всякий раз, когда циклическая задача переходит из состояния "отключенная задача" (INI) или "ожидание" (WAIT) в состояние "готовность" (READY) (в отличие от флага "Первое выполнение задачи", который включается только тогда, когда состояние "отключенная задача" (INI) сменяется на "готовность" (READY)).



Флаг "Запуск задачи" можно использовать для выполнения процедуры инициализации всякий раз, когда задача переходит из состояния "ожидание" в состояние "выполнение" (RUN), то есть, когда с помощью команды TRON(820) разрешается выполнение задачи, находящейся в режиме "ожидание".



**Флаги, относящиеся ко всем задачам**

**Флаг "Ошибка задачи" (A29512)**

Флаг "Ошибка задачи" включается, если происходит одна из следующих ошибок задачи.

- В пределах цикла отсутствуют циклические задачи (включая дополнительные циклические задачи) в состоянии "готовность".
- Программа, назначенная циклической задаче (включая дополнительные циклические задачи), не существует (в случае использования CX-Programmer или консолей программирования эта ситуация не произойдет).
- Вызванной задаче обработки прерывания не назначена программа.

**Номер задачи после остановки программы (A294)**

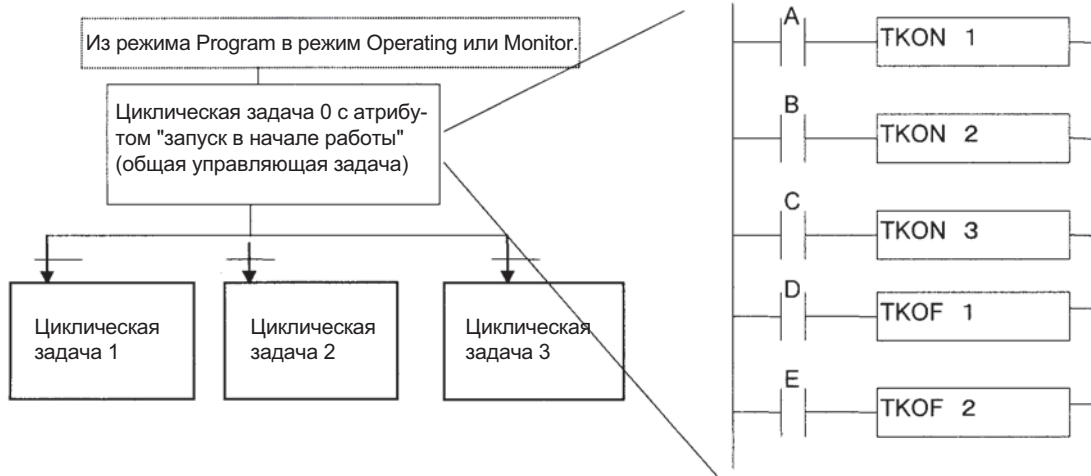
Когда выполнение задачи прекращается из-за ошибки программы, тип задачи и номер текущей задачи сохраняются в следующие слова:

Тип	A294
Циклическая задача	0000 ... 001F Hex (соответствуют номерам задач 0 ... 31)
Задача обработки прерывания	8000 ... 80FF Hex (соответствуют номерам задач обработки прерываний 0 ... 255)

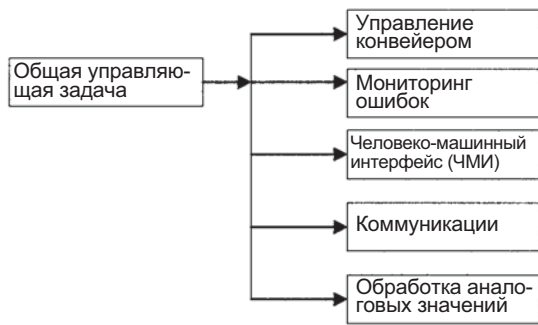
С помощью этой информации упрощается поиск места в программе, вызвавшего фатальную ошибку. Эта информация обнуляется после сброса фатальной ошибки. Адрес программы, по которому прекратилось выполнение задачи, сохраняется в A298 (младшие биты адреса программы) и в A299 (старшие биты адреса программы).

**Примеры задач**

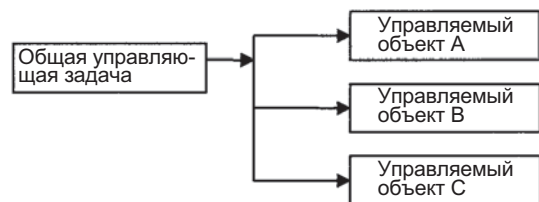
Общая управляющая задача, которая переходит в состояние "готовность" в начале работы, в общем случае служит для управления состояниями "готовность"/"ожидание" для всех остальных циклических задач (включая дополнительные циклические задачи). Разумеется, любая циклическая задача может управлять состояниями "готовность"/"ожидание" любой другой циклической задачи, если это требуется в конкретном случае.



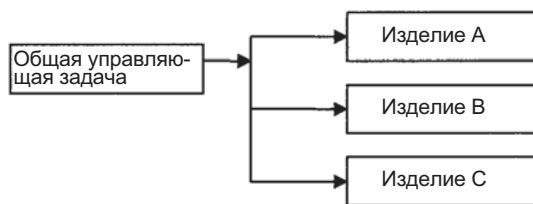
Классификация задач по функциям



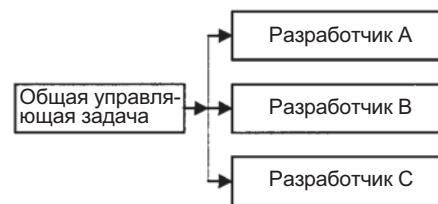
Классификация задач по объектам управления



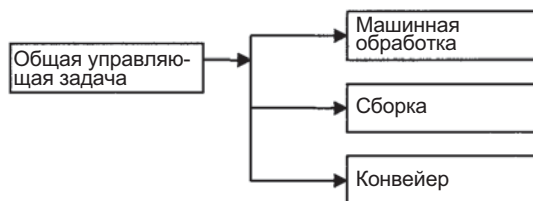
Классификация задач по изделиям



Классификация задач по разработчикам



Классификация задач по процессам



Можно также комбинировать перечисленные выше способы классификации, например, классифицировать задачи по функциям и технологическим процессам.

### 4-2-4 Создание задач

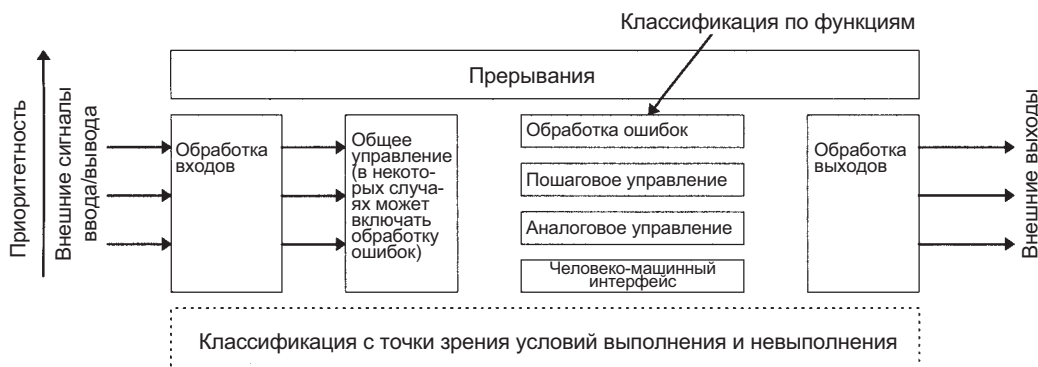
Мы можем дать следующие рекомендации разработчикам задач.

- 1,2,3... 1. При разработке отдельных задач используйте следующие стандартные подходы.
- Объединяйте отдельные условия выполнения и невыполнения задач.
  - Объединяйте условия присутствия или отсутствия сигналов на внешних входах/выходах.
  - Объединяйте функции.

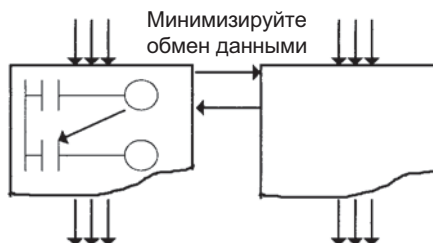
Постарайтесь свести к минимуму обмен данными между задачами пошагового управления, аналогового управления, человеко-машинного интерфейса, обработки ошибок и других процессов, сохраняя автономность задач, насколько это возможно.

- Оцените порядок выполнения задач с точки зрения их приоритетности.

Разделите задачи на циклические задачи и задачи обработки прерываний



- Старайтесь классифицировать и разрабатывать программы таким образом, чтобы обеспечивать автономность отдельных задач и сводить к минимуму обмен данными между отдельными задачами (программами).



- В общем случае для управления состояниями "готовность"/"ожидание" других задач следует использовать общую управляющую задачу.
- Назначайте наименьшие номера задачам с наибольшим приоритетом. Пример: присвойте управляющей задаче наименьший номер по сравнению с прочими задачами обработки.
- Назначайте наименьшие номера задачам обработки прерываний с более высоким приоритетом.
- Задача, находящаяся в состоянии "готовность", будет выполняться в последующих циклах, пока она не будет переведена в состояние "ожидание" другой задачей или не переведет себя сама. Если предусматривается ветвление между задачами, в других задачах обязательно следует вставлять команду TKOF(821) (TASK OFF).
- Для инициализации задач в условия выполнения команд следует вставлять флаг "Первое выполнение команды" (A20015) или флаг "Запуск команды" (A20014). Флаг "Первое выполнение команды" будет включен во время первого выполнения каждой задачи. Флаг "Запуск задачи" будет включаться каждый раз, когда задача переходит в состояние "готовность"

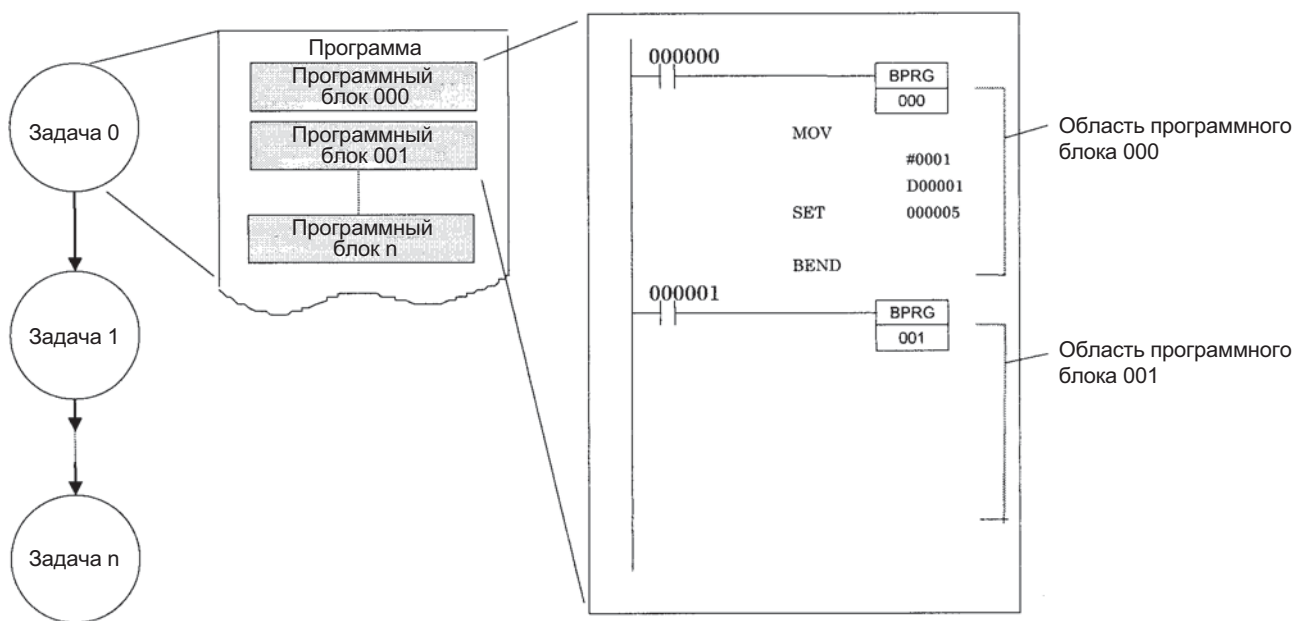


- Резервируя память ввода/вывода, выделите в ней область, используемую совместно всеми задачами, и область, которая будет использоваться отдельными задачами. Последнюю разбейте на отдельные области для каждой задачи.

**Взаимосвязь между задачами и программными блоками**

Суммарное количество программных блоков, которое может быть создано во всех задачах в целом, составляет 128. Управление выполнением каждого отдельного программного блока производится из "лестничной диаграммы", но команды внутри программного блока записываются с помощью мнемонических кодов. Другими словами, программный блок является комбинацией команд "лестничной диаграммы" и мнемонических кодов.

Применение программных блоков облегчает написание "лестничной диаграммы" в таких случаях, как ветвление по условиям или шаговое выполнение процессов, которое довольно сложно программировать с помощью обычных "лестничных диаграмм". Программные блоки размещаются в нижней части иерархии программ, а объемные программные модули, представляемые задачей, могут быть разбиты на отдельные программные блоки небольшого размера, запускаемые одним общим условием выполнения (условие ВКЛ).



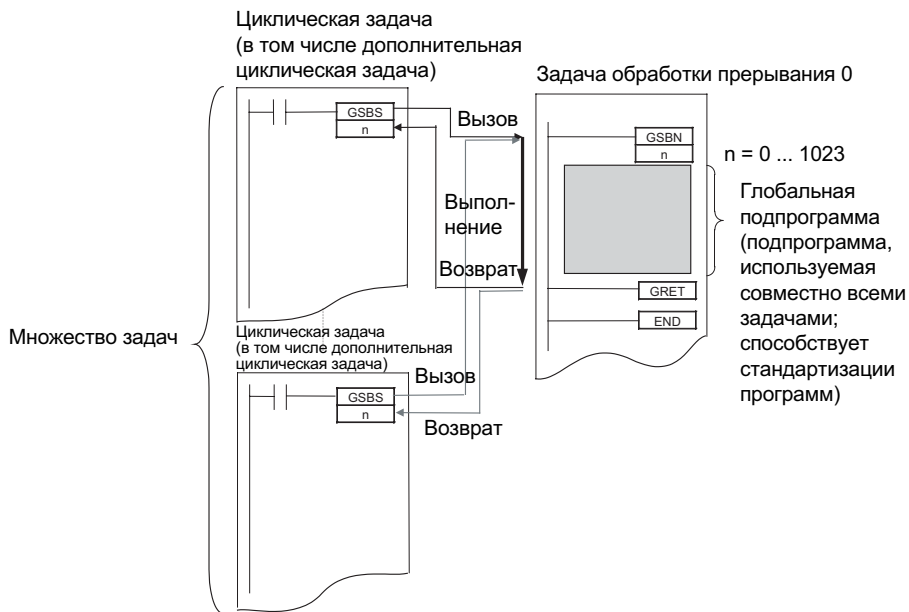
**4-2-5 Глобальные подпрограммы**

Глобальная подпрограмма может быть вызвана из нескольких задач. Глобальные подпрограммы поддерживаются только модулями CPU CS1-H, CJ1-H, CJ1M или CS1D.

В модулях CPU CS1 или CJ1 подпрограмма, принадлежащая одной задаче, не может быть вызвана из других задач. Однако в модулях CPU CS1-H, CJ1-H, CJ1M или CS1D глобальные подпрограммы можно создавать в задаче обработки прерывания 0, и такие подпрограммы можно вызывать из циклических задач (включая дополнительные циклические задачи).

Для вызова глобальной подпрограммы служит команда GSBS. Номер подпрограммы должен находиться в пределах от 0 до 1023. Глобальная подпрограмма записывается в конце задачи обработки прерывания с номером 0 (непосредственно перед END(001)) и размещается между командами GSBN и GRET.

Глобальные подпрограммы можно использовать для создания библиотеки стандартных программных блоков, которые можно вызывать по мере необходимости.



### 4-3 Задачи обработки прерываний

#### 4-3-1 Типы задач обработки прерываний

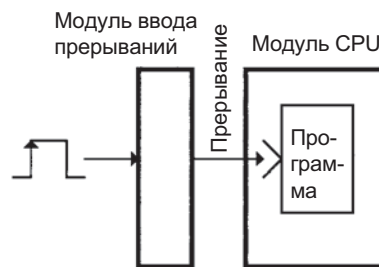
Задачи обработки прерываний могут быть выполнены в любое время в пределах цикла, если соблюдается любое из перечисленных ниже условий.

Для вызова задач обработки прерываний можно использовать встроенные входы прерываний и входы высокоскоростных счетчиков модуля CPU CJ1M. Подробное описание смотрите в руководстве *CJ Series Built-in I/O Operation Manual*.

**Примечание** Модули CPU CS1D для систем с дублированием CPU не поддерживают прерывания. В этих модулях задачи обработки прерываний можно использовать только в качестве дополнительных циклических задач.

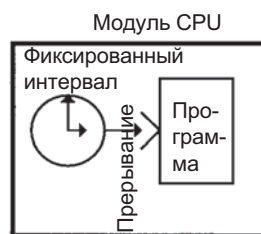
**Прерывания, формируемые входами/выходами (только серия CS)**

Задача обработки прерывания от входа/выхода выполняется, когда включается вход модуля ввода прерываний.



**Запланированные прерывания**

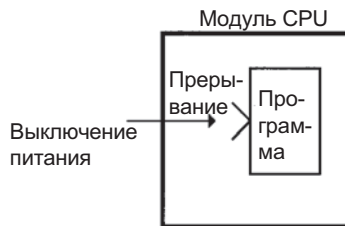
Задача обработки запланированного прерывания выполняется через установленные интервалы времени.



**Прерывание, формируемое по выключению питания**

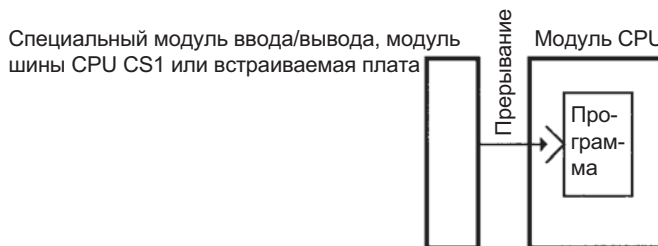
Задача обработки прерывания от выключения питания выполняется, когда происходит выключение напряжения питания.

**Примечание** Время выполнения задачи обработки прерывания по питанию не должно превышать 10 мс (время задержки обнаружения выключения питания).



**Внешние прерывания (только серия CS)**

Задача обработки внешнего прерывания выполняется, когда запрос на прерывание поступает от специального модуля ввода/вывода, от модуля шины CPU или от встраиваемой платы (только серия CS). При этом специальный модуль ввода/вывода или модуль шины CPU должен быть установлен в стойку CPU, чтобы обладать возможностью запроса на выполнение задач обработки внешних прерываний.



**Список задач обработки прерываний**

Тип	Номер задачи	Условие выполнения	Процедура настройки	Количество прерываний	Примеры применения
Прерывания от входов/выходов 00 ... 31	100 ... 131	Включение входа модуля ввода прерываний в стойке CPU (см. прим. 1).	Для назначения входов модулей ввода прерываний в стойке CPU используйте команду MSKS (SET INTERRUPT MASK).	32 точки	Организация задержки срабатывания для отдельных входов
Запланированные прерывания 0 и 1	2 и 3	С установленной периодичностью	Для выбора интервала формирования прерывания используйте команду MSKS (SET INTERRUPT MASK). Используйте параметр Шаг установки времени для формирования запланированных прерываний в настройках ПЛК.	2 точки	Контроль рабочего состояния через фиксированные интервалы времени
Прерывание по выключению питания	1	Выключение напряжения питания (по истечению принимаемого по умолчанию времени обнаружения выключения питания + время задержки обнаружения выключения питания)	Используйте параметр Задача обработки прерывания по выключению питания и Время задержки обнаружения выключения питания в настройках ПЛК.	1 точка	Выполнение неотложных действий в случае отключения питания.
Внешние прерывания 0 ... 255	0 ... 255	По запросу от специального модуля ввода/вывода или модуля шины CPU в стойке CPU или от встраиваемой платы (только серия CS) (см. прим. 2).	Нет (всегда разрешены)	256 точек	Выполнение операций, необходимых специальным модулям ввода/вывода, модулям шины CPU и встраиваемой плате.

- Примечание**
1. Модуль ввода прерываний должен быть установлен в стойку CPU. В ПЛК CJ1-N модуль должен быть установлен в качестве одного из пяти модулей после модуля CPU (слоты 0 ... 4). В ПЛК CJ1M модуль должен быть установлен в качестве одного из трех модулей после модуля CPU (слоты 0 ... 2). Модули ввода/вывода прерываний, установленные в другие слоты, нельзя использовать для формирования запросов на выполнение задач обработки внешних прерываний от входов/выходов.
  2. Специальный модуль ввода/вывода или модуль шины CPU должен быть установлен в стойку CPU. В ПЛК CJ1-N этот модуль должен быть установлен в качестве одного из пяти модулей после модуля CPU (слоты 0 ... 4).

В ПЛК CJ1M модуль должен быть установлен в качестве одного из трех модулей после модуля CPU (слоты 0 ... 2). Модули, установленные в другие слоты, не смогут формировать внешние прерывания.

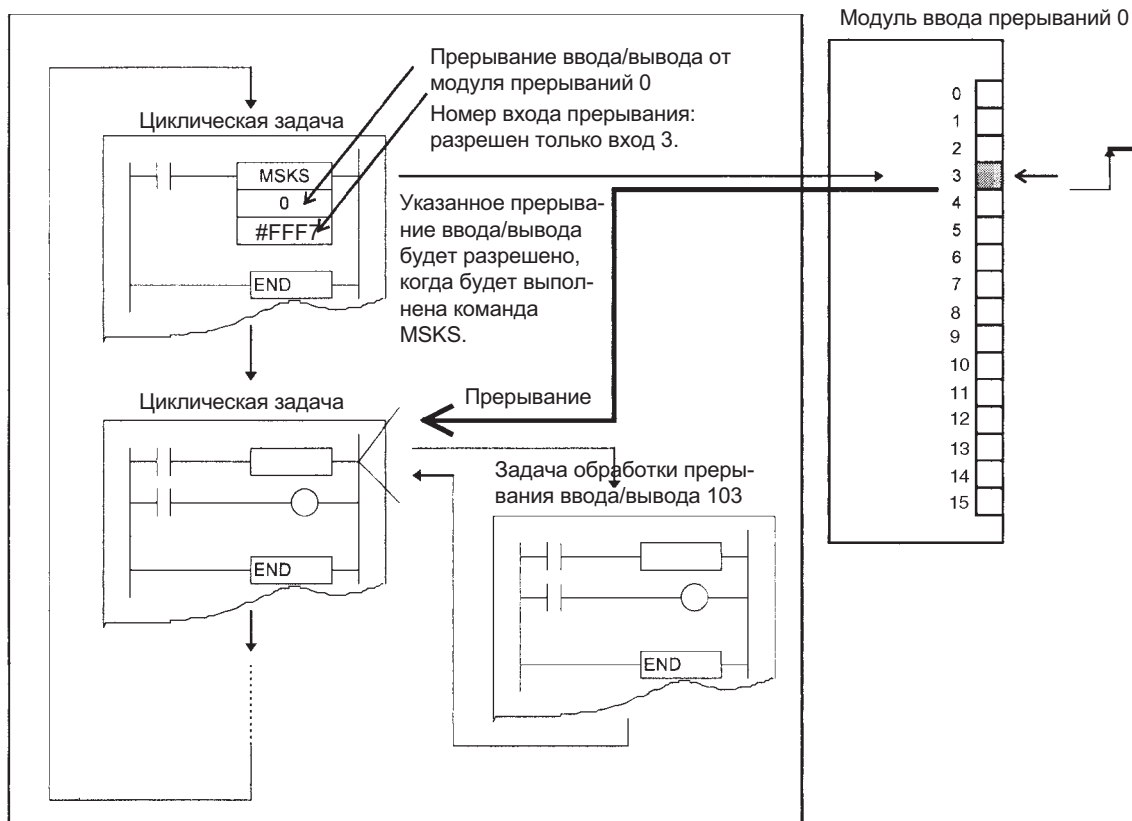
3. Модули CPU CJ1 для систем с дублированием CPU не поддерживают прерывания от входов/выходов и внешние прерывания.
4. Модули CPU CS1D для систем с дублированием CPU не поддерживают прерывания. У этих модулей задачи обработки прерываний можно использовать только в качестве дополнительных циклических задач. Другие типы задач обработки прерываний использовать нельзя.

**Задачи обработки прерываний от входов/выходов: задачи 100 ... 131**

В начале выполнения циклических задач выполнение задач обработки прерываний от входов/выходов по умолчанию отключено. Чтобы разрешить прерывания от входов/выходов, в пределах циклической задачи следует выполнить команду MSKS (SET INTERRUPT MASK) для модуля ввода прерываний, указав номер требуемого прерывания.

**Пример:** ниже приведен пример выполнения задачи обработки прерываний от входов/выходов (номер задачи 103), запускаемой по включению входа 3 модуля ввода прерываний 0 (крайний левый модуль в паре модулей 0 и 1).

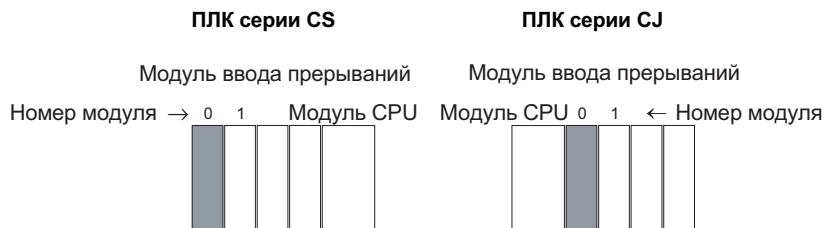
**Примечание** Не следует разрешать ненужные задачи обработки прерываний от входов/выходов. Если прерывание срабатывает из-за воздействия помех, а задача обработки прерывания с соответствующим номером отсутствует, будет сформирована фатальная ошибка (ошибка задачи), которая приведет к прекращению выполнения программы.



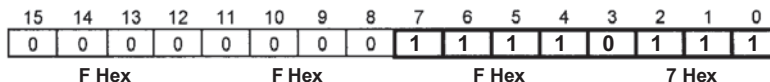
Номера модулей ввода прерываний, номера входов и номера задач обработки прерываний ввода/вывода

Номер модуля ввода прерываний (см. прим.)	Номер входа	Задачи обработки прерываний от входов/выходов
0	0 ... 15	100 ... 115
1	0 ... 15	116 ... 131

**Примечание** В ПЛК серии CS модули ввода прерываний нумеруются в порядке от 0 до 1, начиная с левой стороны стойки CPU. В ПЛК серии CJ модули ввода прерываний нумеруются по порядку от 0 до 1, начиная с модуля CPU.



**Операнд S (второй операнд) команды MSKS:** биты FFF7 Hex соответствуют входам прерываний модуля ввода прерываний. Входы прерываний 0 ... 15 соответствуют битам 0 ... 15.



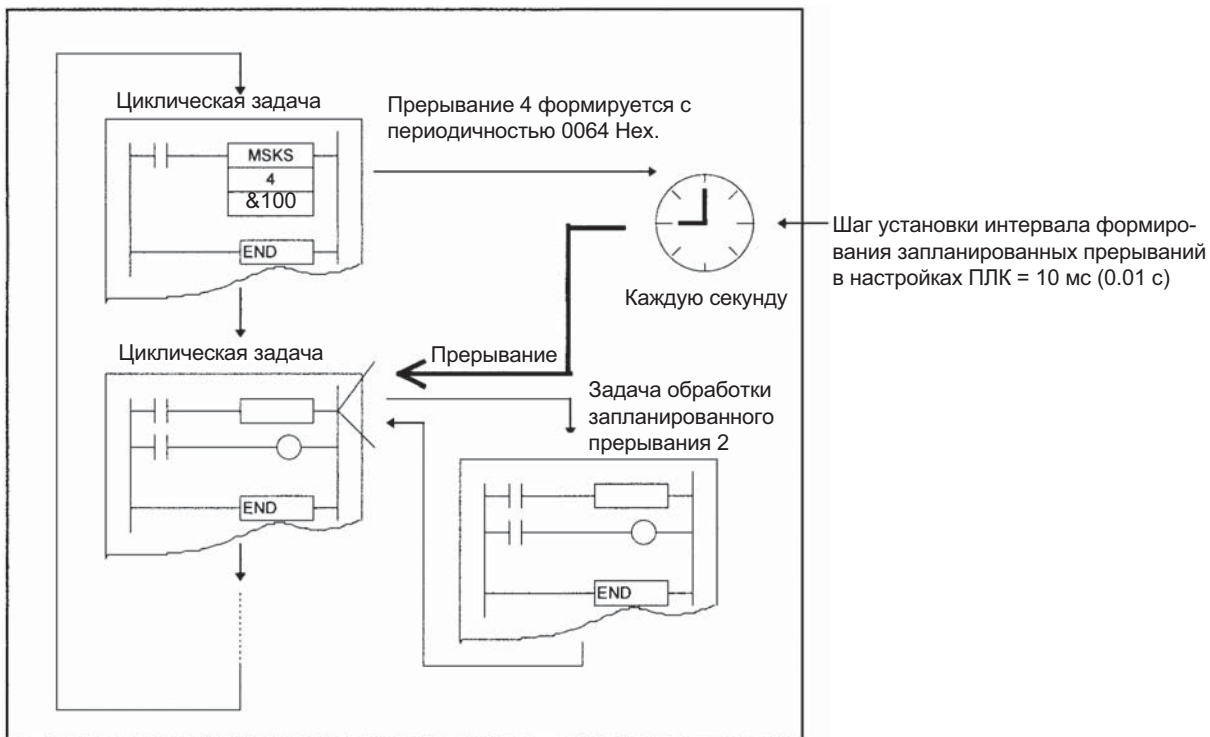
### Задачи обработки запланированных прерываний: задачи 2 и 3

В начале выполнения циклических задач выполнение задач обработки запланированных прерываний в настройках ПЛК по умолчанию отключено. Чтобы разрешить выполнение задач обработки запланированных прерываний, необходимо выполнить следующие действия.

- 1,2,3...**
1. Выполните в циклической задаче команду MSKS (SET INTERRUPT MASK) и задайте время (период) для указанного запланированного прерывания.
  2. Выберите в настройках ПЛК единицы измерения (шаг установки) для времени формирования запланированных прерываний.

**Примечание** Параметры, настроенные для времени формирования прерывания, влияют на выполнение циклических задач. Чем короче интервал формирования прерывания, тем чаще выполняются задачи и тем больше длительность цикла.

**Пример:** ниже приведены примеры выполнения задачи обработки запланированного прерывания (номер задачи 2) с периодичностью 1 раз в секунду.



**Номера прерываний и номера задач обработки запланированных прерываний**

Номер прерывания	Задача обработки запланированного прерывания
4	2
5	3

**Значения параметров в области настроек ПЛК**

Адрес	Наименование	Описание	Значения	Значение по умолчанию
Биты 0 ... 3 слова 195	Шаг установки интервала формирования прерываний	Задаёт шаг установки времени (периода) для формирования запланированных прерываний, то есть, для выполнения задач обработки прерываний через фиксированные промежутки времени.	00 Hex: 10 мс 01 Hex: 1.0 мс 02 Hex: 0.1 мс (только модуль CPU CJ1)	00 Hex

**Задача обработки прерывания по выключению питания: задача 1**

В начале выполнения циклических задач выполнение задачи обработки прерывания от выключения питания по умолчанию (настройки ПЛК) отключено.

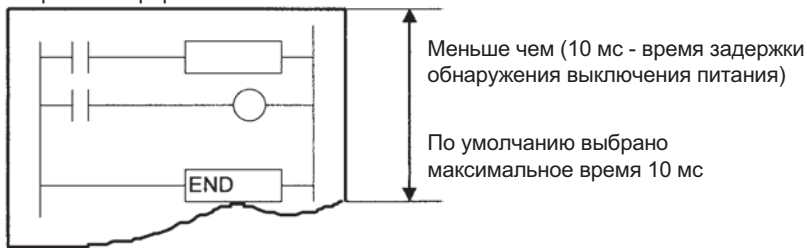
Задачу обработки прерывания от выключения питания можно разрешить в настройках ПЛК.

По умолчанию в настройках ПЛК установлено прекращение выполнения задачи обработки прерывания от выключения питания через 10 мс. Задача обработки прерывания от выключения питания должна быть выполнена менее чем за 10 мс.

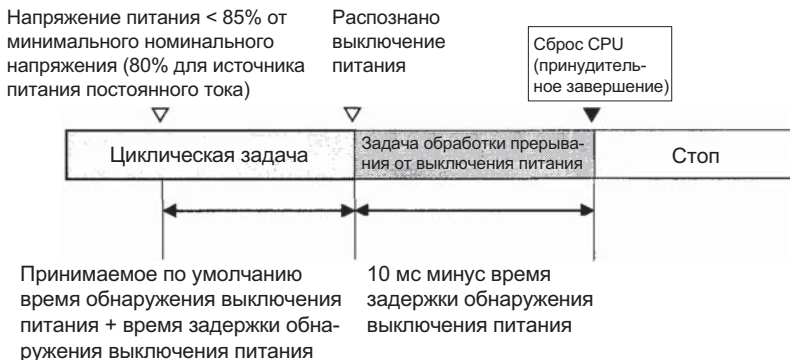
Если в настройках ПЛК установлено время задержки обнаружения выключения питания, задача обработки прерывания от выключения питания будет прекращена через 10 мс минус время задержки обнаружения выключения питания, установленное в настройках ПЛК. В этом случае задача обработки прерывания от выключения питания должна быть выполнена за время, определяемое как разница между 10 мс и временем задержки обнаружения выключения питания, заданным в настройках ПЛК.

**Пример:** Если время задержки обнаружения выключения питания в настройках ПЛК установлено равным 4 мс, в этом случае время выполнения задачи не должно превышать (10 - 4 = 6) мс.

Задача обработки прерывания от выключения питания

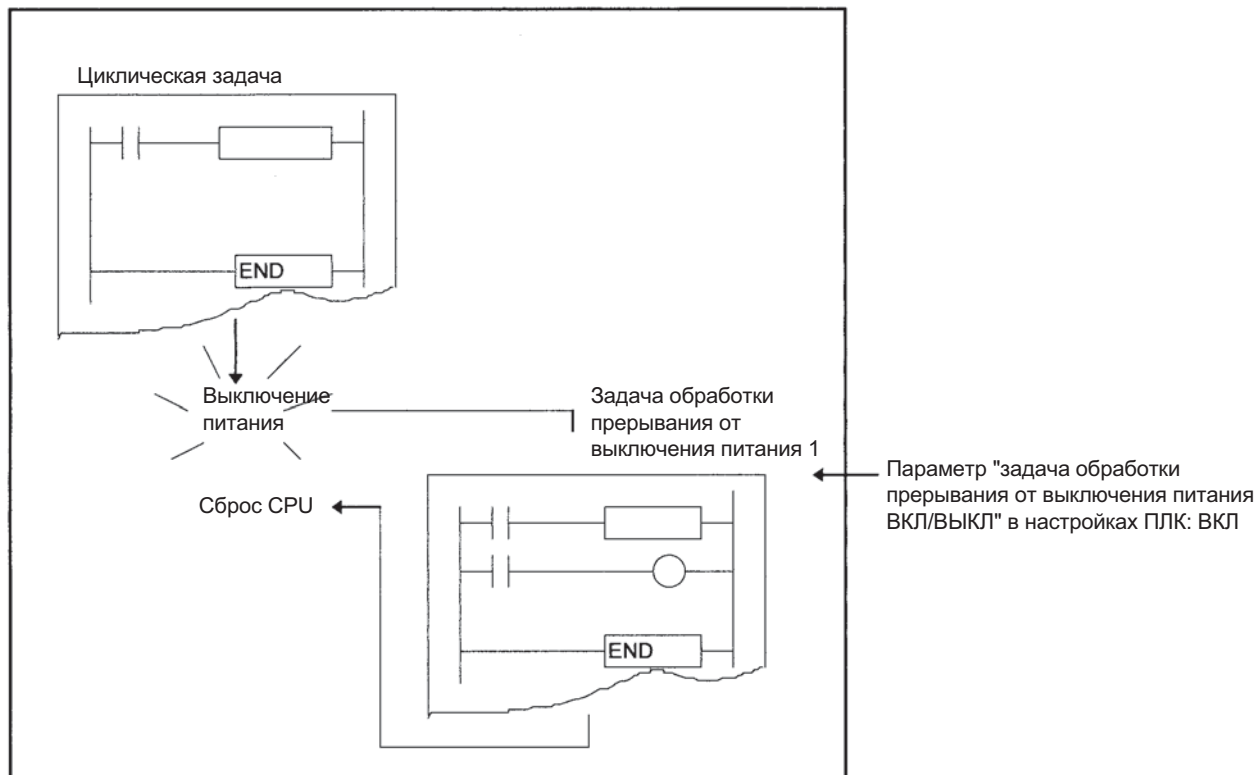


**Примечание** Выключение питания считается обнаруженным, если напряжение питания падает ниже уровня 85% от минимального номинального напряжения (80% для модулей питания постоянного тока). Время, которое проходит, прежде чем будет выполнена задача обработки прерывания от выключения питания, определяется как сумма принимаемого по умолчанию времени обнаружения выключения питания (10 ... 25 мс для источников питания переменного тока и 2 ... 5 мс для источников питания постоянного тока) и времени задержки обнаружения выключения питания, заданного в настройках ПЛК (0 ... 10 мс). В течение этого времени будут выполняться циклические задачи.



**Примечание** Обеспечьте, чтобы задача обработки прерывания от выключения питания выполнялась за время, меньшее чем 10 мс минус время задержки обнаружения выключения питания, установленное в настройках ПЛК. По истечении этого времени ни одна из оставшихся команд выполнена не будет. Задача обработки прерывания от выключения питания не будет выполнена в случае пропадания питания во время online-редактирования. Помимо команд, которые нельзя использовать ни в одной из задач обработки прерываний (см. описание в руководстве *Instructions Reference Manual*), в задаче обработки прерывания от выключения питания нельзя использовать следующие команды: READ DATA FILE: FREAD(700), WRITE DATA FILE: FWRT(701), NETWORK SEND: SEND(090), NETWORK RECEIVE: RECV(098), DELIVER COMMAND: CMND(490), TRANSMIT: TXD(236), RECEIVE: RXD(235) и PROTOCOL MACRO: PMCR(260).

Выполнение задачи обработки прерывания от выключения питания



Значения параметров в настройках ПЛК для задачи обработки прерывания от выключения питания (номер задачи: 1)

Адрес	Наименование	Описание	Значения	Значение по умолчанию
Бит 15 в + 225	Задача обработки прерывания от выключения питания	Если бит 15 в + 225 установлен (ВКЛ), при выключении питания будет выполнена задача обработки прерывания от выключения питания.	0: ВЫКЛ 1: ВКЛ	0
Биты 0 ... 7 в + 225	Время задержки обнаружения выключения питания	Выключение питания будет обнаружено по истечении этого времени + время обнаружения выключения питания, принимаемое по умолчанию (10 ... 25 мс для блоков питания переменного тока и 2 ... 5 мс для блоков питания постоянного тока).	00 ... 0A Hex: 0 ... 10 мс (с шагом 1 мс)	00 Hex

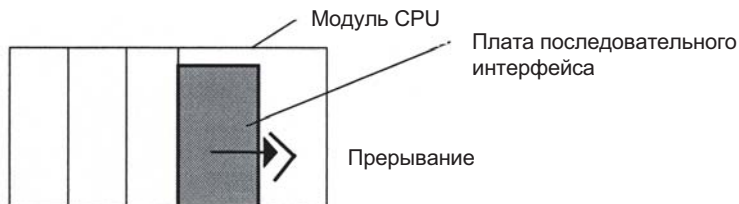
Задачи обработки внешних прерываний: задачи 0 ... 255

Задачи обработки внешних прерываний могут выполняться в любое время.

Обработка прерывания выполняется в модуле CPU в ПЛК, в состав которого входит встраиваемая плата (только серия CS), специальные модули ввода/вывода или модуль шины CPU. Никакие параметры в модуле CPU настраивать не требуется, за исключением случая, когда программа содержит задачу обработки внешнего прерывания для задачи с определенным номером.

Внешние прерывания не поддерживаются модулями CPU CJ1.

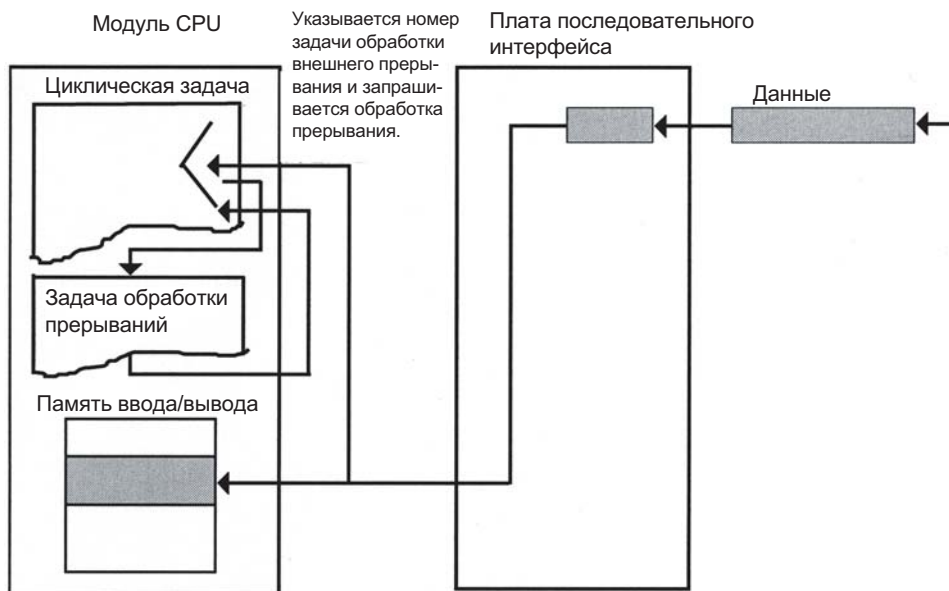
**Пример:** ниже показан пример формирования внешнего прерывания от платы последовательного интерфейса CS1W-SCB □.



Если в качестве уведомления о получении ответа от платы последовательного интерфейса выбрано прерывание (прерывание с фиксированным номером или номером конкретного случая приема), в этом случае плата выставит запрос



на выполнение задачи обработки внешнего прерывания в модуле CPU после того, как она получит данные через свой последовательный порт и запишет эти данные в память ввода/вывода модуля CPU.



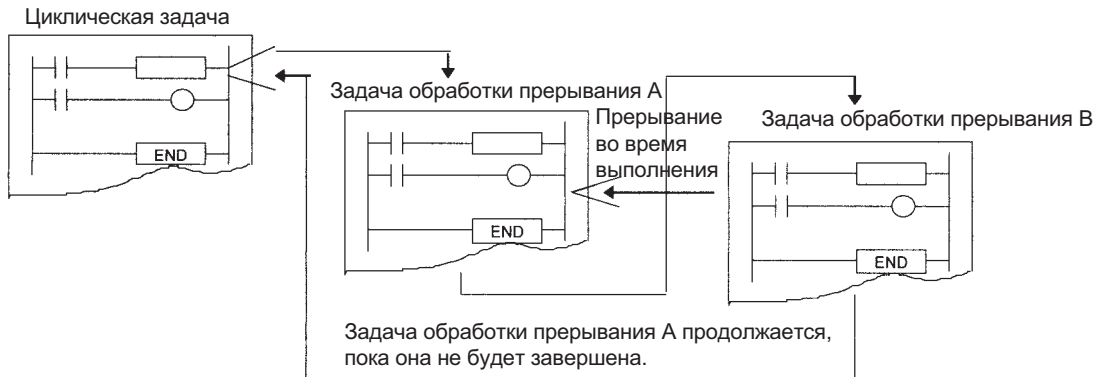
- Примечание**
1. Если в качестве способа уведомления о получении отклика выбрано формирование прерывания (с фиксированным номером), плата запрашивает выполнение задачи обработки прерывания с предустановленным номером задачи.
  2. Если в качестве способа уведомления о получении отклика выбрано формирование прерывания (с номером конкретного случая приема), в этом случае по указанной формуле рассчитывается номер задачи обработки внешнего прерывания и плата запрашивает выполнение задачи обработки прерывания с этим номером задачи.
  3. Если номер задачи обработки внешних прерываний (0 ... 255) совпадает с номером задачи обработки прерывания от выключения питания (задача 1), обработки запланированного прерывания (задача 2 или 3) или обработки прерывания от входов/выходов (100 ... 131), в этом случае задача обработки прерывания будет выполнена по возникновению любого из этих условий (внешнего прерывания или другого прерывания). В общем случае не рекомендуется дублировать номера задач.

### 4-3-2 Приоритетность задач обработки прерываний

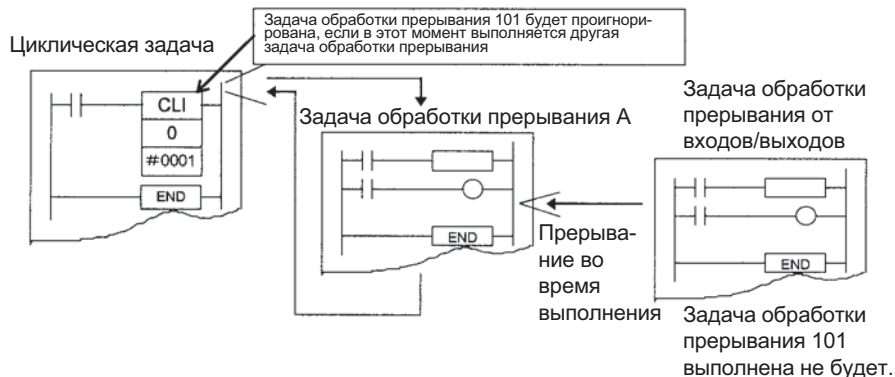
Если должна быть выполнена задача обработки прерывания от выключения питания, выполнение задачи обработки любого другого прерывания прекращается. CPU будет сброшен, прерванная задача обработки прерывания не будет выполнена по завершению выполнения задачи обработки прерывания от выключения питания.

**Возникновение прерывания во время выполнения задачи обработки прерывания**

Если прерывание поступает в момент, когда уже выполняется другая задача обработки прерывания, задача обработки для нового прерывания не начнет выполняться до тех пор, пока не завершится выполнение текущей задачи обработки прерывания.



**Примечание** Если не требуется запоминать и выполнять задачу обработки прерывания по входам/выходам с определенным номером, когда "её" прерывание поступает в момент выполнения другой задачи обработки прерывания, в модуле CPU серии CS можно выполнить команду CLI (CLEAR INTERRUPT) из другой задачи обработки прерывания, чтобы обнулить (CLEAR) сохраненный номер прерывания. Запланированные прерывания и внешние прерывания отменены быть не могут.



**Формирование нескольких прерываний одновременно**

Ниже показан порядок выполнения задач обработки прерываний (за исключением задачи обработки прерывания от выключения питания) в том случае, когда одновременно возникает несколько прерываний.

Задачи обработки прерываний от входов/выходов (только серия CS) -> задачи обработки внешних прерываний (только серия CS) -> задачи обработки запланированных прерываний.

Если формируется несколько прерываний, в пределах каждого типа задач обработки прерываний задачи выполняются по порядку, начиная с задачи с наименьшим номером.

**Примечание** Для каждой задачи обработки прерывания в память записывается факт возникновения только одного прерывания, а для прерывания, которое уже обрабатывается, факт возникновения нового прерывания не запоминается. Поскольку запланированные прерывания выполняются с наименьшим приоритетом и поскольку одновременно запоминается только одно прерывание, существует вероятность пропуска запланированного прерывания.

**4-3-3 Флаги и слова задач обработки прерываний**

**Максимальное время выполнения задачи обработки прерывания (A440)**

Максимальное время выполнения задачи обработки прерывания хранится в двоичном коде (с шагом 0.1 мс) и обнуляется в начале работы.

**Задача обработки прерывания с максимальным временем выполнения (A441)**

Номер задачи обработки прерывания, потребовавшей максимальное время выполнения, хранится в двоичном формате. В данном случае значения 8000 ... 80FF Hex соответствуют задачам с номерами 00 ... FF Hex.

После того, как будет сформировано первое прерывание после начала работы, устанавливается бит A44115. Максимальное время выполнения для последующих задач обработки прерываний сохраняется в двух старших разрядах в шестнадцатеричном формате и обнуляется в начале работы.

**Флаг "Ошибка задачи обработки прерывания" (нефатальная ошибка) (A40213)**

Если в области настроек ПЛК установлен бит "Обнаружение ошибок выполнения задач обработки прерываний", то в случае возникновения ошибок при выполнении задач обработки прерываний будет устанавливаться флаг "Ошибка выполнения задачи обработки прерывания".

**Флаг "Ошибка задачи прерывания" (A42615)/Номер задачи обработки прерывания, в которой произошла ошибка (A42600 ... 42611)**

Если включается бит A40213, в слова A42615 и A42600 ... A42611 записываются следующие данные.

A40213	Описание ошибки выполнения задачи обработки прерывания	A42615	A42600 ... 42611
"Ошибка выполнения задачи обработки прерывания" (если установлен бит "Обнаружение ошибок выполнения задач обработки прерываний" в настройках ПЛК)	Если задача обработки прерывания выполняется дольше 10 мс во время обновления удаленных входов/выходов на шине SYSMAC BUS (только серия CS) или входов/выходов специального модуля ввода/вывода C200H.	ВЫКЛ	Номер задачи обработки прерывания записывается в 12-битовое двоичное число (задачи обработки прерываний 0 ... 255: 000 ... 0FF Hex).
	При попытке обновления входов/выходов для большого количества слов с помощью команды IORF из задачи обработки прерывания во время обновления входов/выходов специального модуля ввода/вывода в режиме циклического обновления.	ВКЛ	Номер модуля для обновляемого специального модуля ввода/вывода будет записан в 12-битовое двоичное число (номера модулей 0 ... 95: 000 ... 05F Hex).

**Номер задачи, на которой прекратилось выполнение программы (A294)** В следующей таблице указаны слова, в которые будут записаны тип задачи и номер текущей задачи, в момент выполнения которой произошла остановка программы из-за возникновения ошибки.

Тип	A294
Задача обработки прерывания	8000 ... 80FF Hex (соответствуют задачам обработки прерываний 0 ... 255)
Циклические задачи	0000 to 001F Hex (соответствуют задачам 0 ... 31)

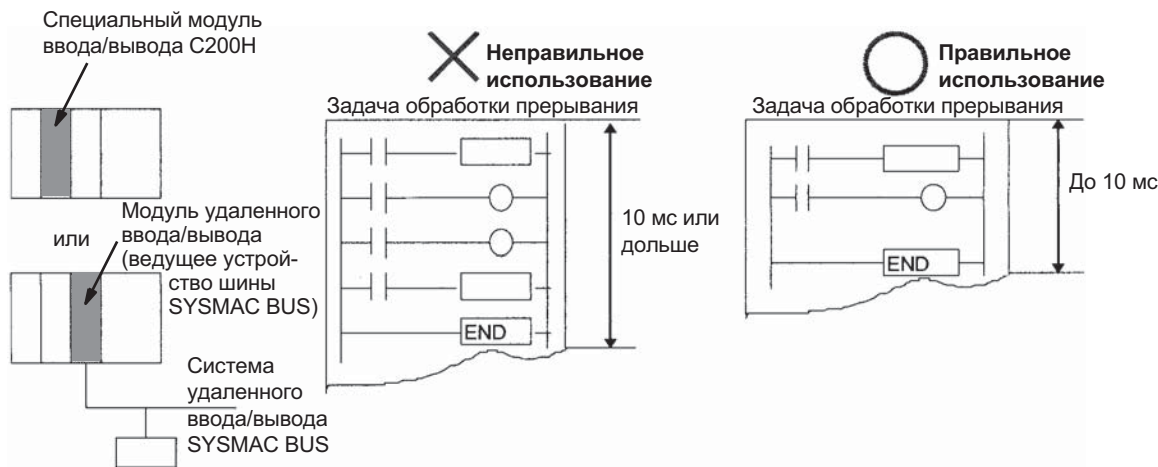
**4-3-4 Замечания по применению**

**Увеличение времени выполнения у специальных модулей ввода/вывода серии C200H или модулей шины SYSMAC BUS (только серия CS)**

Необходимо обеспечить, чтобы все задачи обработки прерываний (от входов/выходов, запланированных прерываний, прерываний от выключения питания и внешних прерываний) выполнялись в пределах 10 мс в случае использования специальных модулей ввода/вывода серии C200H или системы удаленного ввода/вывода SYSMAC BUS.

Если задача обработки прерывания выполняется дольше 10 мс в момент, когда происходит обновление входов/выходов специального модуля ввода/вывода серии C200H или системы удаленного ввода/вывода SYSMAC BUS, в этом случае произойдет ошибка выполнения задачи обработки прерывания, установится (ВКЛ) бит A40206 (флаг "Ошибка специального модуля ввода/вывода") и обновление входов/выходов в специальных модулях ввода/вывода будет прекращено. При этом модуль CPU продолжит работу.

Если в настройках ПЛК установлен бит "Обнаружение ошибок выполнения задач обработки прерываний", в этом случае при возникновении ошибок выполнения задач обработки прерываний устанавливается флаг A40213 ("Ошибка выполнения задачи обработки прерывания"), а в A426 (Ошибка задачи обработки прерывания, Номер задачи) будет записан номер задачи обработки прерывания, вызвавшей ошибку. При этом модуль CPU продолжит работу.



**Выполнение команды IORF для специального модуля ввода/вывода**

Если из задачи обработки прерывания должна быть выполнена команда IORF(097) для специального модуля ввода/вывода, следует обязательно отключить циклическое обновление для специального модуля ввода/вывода (с соответствующим номером) в настройках ПЛК.

При попытке выполнить обновление входов/выходов специального модуля ввода/вывода с помощью команды IORF(097) из задачи обработки прерывания, когда этот модуль уже обновляется в рамках циклического обновления входов/выходов или в результате выполнения команд обновления входов/выходов (IORF(097) или команд мгновенного обновления (!)), произойдет ошибка выполнения задачи обработки прерывания. Если в настройках ПЛК установлен бит "Обнаружение ошибок выполнения задач обработки прерываний", в этом случае при возникновении ошибок выполнения задач обработки прерываний устанавливается флаг A40213 ("Ошибка выполнения задачи обработки прерывания") и в слово A426 ("Ошибка задачи обработки прерывания, Номер задачи") записывается номер специального модуля ввода/вывода, с которым связано дублирование обновления входов/выходов. Модуль CPU продолжит работу.



**Примечание** С помощью старших битов слова A426 (Ошибка задачи обработки прерывания, Номер задачи) можно определить, какая из перечисленных выше ошибок выполнения задач обработки прерываний произошла (бит 15: 0 - время выполнения превысило 10 мс; 1 - дублирование обновления).

**Значения параметров в настройках ПЛК**

Адрес	Наименование	Описание	Значения	Значение по умолчанию
Бит 14 в + 128	Обнаружение ошибки выполнения задачи обработки прерывания	Указывает, следует ли обнаруживать ошибки выполнения задач обработки прерываний. Если выбрано обнаружение, используется флаг "Ошибка выполнения задачи обработки прерывания" A(40213).	0: обнаружение разрешено, 1: обнаружение запрещено	0

**Сопутствующие флаги/слова дополнительной области**

Название	Адрес	Описание
Флаг "Ошибка выполнения задачи обработки прерывания"	A40213	Устанавливается (ВКЛ), если задача обработки прерывания выполняется дольше 10 мс, когда происходит обновление входов/выходов специального модуля ввода/вывода C200H или системы удаленного ввода/вывода SYSMAC BUS, хотя модуль CPU продолжает работать. На передней панели светится светодиод ERR/ALM (только серия CS). Устанавливается при попытке выполнения обновления для специального модуля ввода/вывода с помощью команды IORF из задачи обработки прерывания, когда этот модуль уже обновляется в рамках процедуры циклического обновления входов/выходов.
Ошибка выполнения задачи обработки прерывания, Номер задачи	A426	Содержит номер задачи обработки прерывания или номер специального модуля ввода/вывода, который обновляется в данный момент. Если выполнение задачи обработки прерывания занимает 10 мс или больше, бит 15 будет сброшен (ВЫКЛ), а в случае дублирования обновления для специального модуля ввода/вывода этот бит будет установлен (ВКЛ).

**Отключение прерываний**

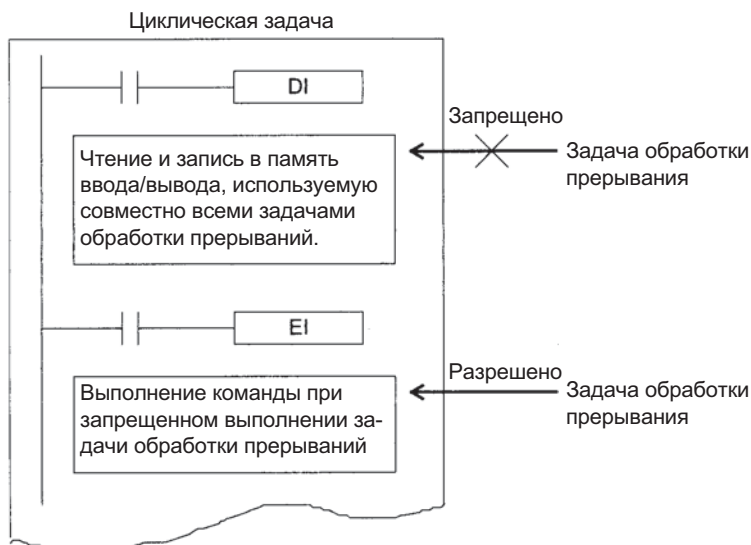
Ниже перечислены случаи, когда процедура выполнения прерывается и выполняется задача обработки прерывания.

- Во время выполнения команды
- Во время обновления входов/выходов для базового модуля ввода/вывода, модуля шины CPU, встраиваемой платы (только серия CS) или системы удаленного ввода/вывода SYSMAC BUS (только серия CS)
- Во время обслуживания HOST LINK

**Совместное использование данных циклическими задачами и задачами обработки прерываний**

Если задача обработки прерывания и циклическая задача (в том числе дополнительная циклическая задача) считывают и записывают данные по одним и тем же адресам памяти ввода/вывода, необходимо позаботиться о согласованности этих операций, то есть, о согласованном использовании данных. Чтобы отменить прерывание во время обращения к памяти из циклических задач, выполните следующие действия.

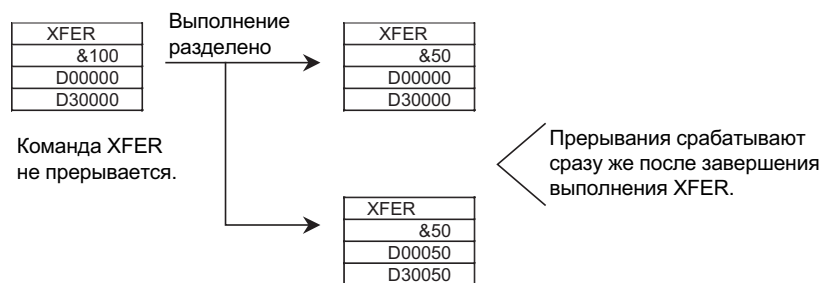
- Непосредственно перед выполнением команды в циклической задаче, осуществляющей чтение или запись, следует выполнить команду DI (DISABLE INTERRUPT), чтобы запретить выполнение задач обработки прерываний.
- Сразу же после выполнения команды разрешите выполнение задачи обработки прерываний с помощью команды EI (ENABLE INTERRUPT).



В случае команд, для которых требуется прием отклика и его обработка (например, команд для сетевого обмена данными или связи по последовательному интерфейсу), проблемы, связанные с несогласованностью данных, могут возникать даже, если для отключения задач обработки прерываний применяются команды DI(693) и EI(694).

**Примечание** У модулей CPU CS1-H, CJ1-H, CJ1M или CS1D выполнение команд BIT COUNTER (BCNT), BLOCK SET (BSET) и BLOCK TRANSFER (XFER) не будет прервано с целью

выполнения задач обработки прерывания, то есть, эти команды будут полностью выполнены, прежде чем начнет выполняться задача обработки прерывания, что увеличивает время реакции на прерывание. Чтобы предотвратить это, можно разбить обработку данных для этих команд на несколько команд, как показано ниже на примере XFER.



## 4-4 Использование средств программирования для задач

### 4-4-1 Использование нескольких циклических задач

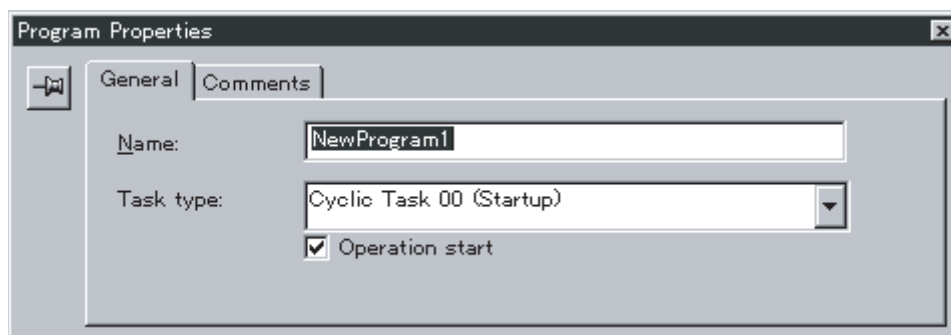
Для создания нескольких циклических задач (включая дополнительные циклические задачи) следует использовать CX-Programmer. Консоль программирования не позволяет создавать новые циклические задачи. Обязательно используйте CX-Programmer для назначения типа и номера задачи создаваемым программам.

- Несколько циклических задач, созданных с помощью CX-Programmer и загруженных в модуль CPU, можно контролировать или редактировать с помощью консоли программирования.
- Консоль программирования можно использовать для создания лишь одной циклической задачи и одной или нескольких специальных задач обработки прерываний, применяя функцию консоли программирования All Clear (Обнулить все) и указывая номера задач обработки прерываний. С помощью консоли программирования можно создать только задачи обработки прерываний 1 (прерывание от выключения питания), 2 и 3 (запланированные прерывания) и 100 ... 131 (прерывания от входов/выходов). Кроме того, для модуля CPU CJ1M также могут быть созданы задачи обработки прерываний 140 ... 143 (для встроенных входов). В начале работы ПЛК запускается циклическая задача 0.

### 4-4-2 Работа со средствами программирования

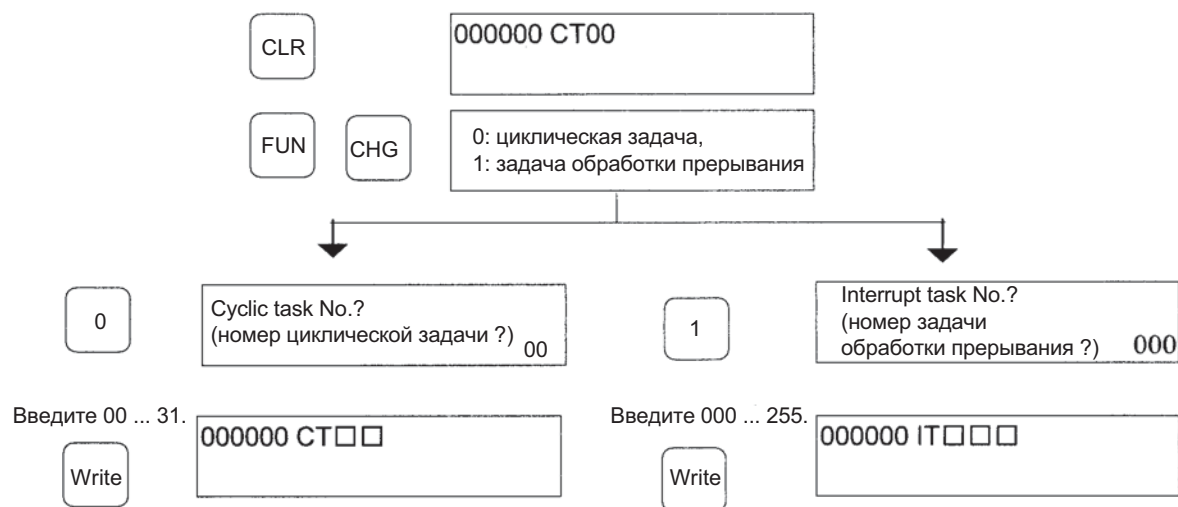
**CX-Programmer** Для каждой программы в качестве атрибутов следует указать тип задачи и номер задачи.

- 1,2,3... 1. Выберите **View/Properties** (Вид/Свойства) или щелкните правой кнопкой мыши и выберите пункт **Properties** (Свойства) из всплывающего меню, чтобы отобразить программу, которой будет назначена задача.
2. Выберите закладку **General** (Общие свойства) и выберите тип задачи (поле **Task Type**) и номер задачи (поле **Task No.**). В случае циклической задачи следует установить флажок **Operation start** (Начало работы), чтобы включить эту задачу.



**Консоль программирования**

В консоли программирования управление задачей осуществляется как управление единой программой. Для доступа и редактирования программы с помощью консоли программирования для циклической задачи следует указать СТ00 ... СТ31, а для задачи обработки прерывания - IT001 ... IT255.



- Примечание**
1. С помощью консоли программирования нельзя создавать новые циклические задачи.
  2. Модули CPU серии CJ в настоящее время не поддерживают задачи обработки прерываний от входов/выходов или внешних прерываний. Можно указать только IT001 ... IT003.

## РАЗДЕЛ 5

### Функции памяти файлов

В данном разделе описаны функции, предназначенные для управления памятью файлов.

5-1	Память файлов . . . . .	190
5-1-1	Типы памяти файлов . . . . .	190
5-1-2	Данные, сохраняемые в файлах . . . . .	192
5-1-3	Файлы. . . . .	194
5-1-4	Способы работы с файлами. . . . .	204
5-1-5	Примеры применения. . . . .	206
5-2	Управление файлами . . . . .	208
5-2-1	Средства программирования (включая консоли программирования). . . . .	208
5-2-2	Команды FINS. . . . .	211
5-2-3	FREAD(700), FWRIT(701) и CMND(490) . . . . .	212
5-2-4	Замена полностью всей программы во время работы . . . . .	217
5-2-5	Автоматическая загрузка при запуске. . . . .	222
5-2-6	Функция простого резервного сохранения данных . . . . .	228
5-3	Применение памяти файлов . . . . .	240
5-3-1	Инициализация носителя . . . . .	240
5-3-2	Рабочие процедуры. . . . .	242
5-3-3	Прерывание питания при обращении к памяти файлов. . . . .	246



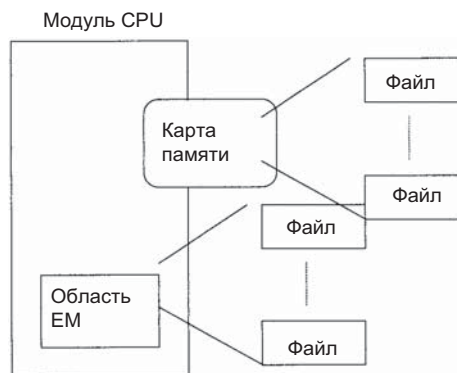
## 5-1 Память файлов

ПЛК серии CS/SJ поддерживают память файлов. В качестве памяти для хранения файлов могут служить следующие носители.



- 1,2,3...
1. Карты памяти
  2. Выбранный диапазон области EM, который называют памятью файлов EM

**Примечание** В модулях CPU SJ1M область EM отсутствует, поэтому память файлов EM использовать нельзя.

Оба типа памяти можно использовать для хранения в виде отдельных файлов полностью всей программы пользователя, данных памяти ввода/вывода и параметров



### 5-1-1 Типы памяти файлов

Категория	Тип	Емкость	Модель	Файлы, воспринимаемые модулем CPU	Допустимые операции с файлами
Карты памяти 	Флеш-память	15 Мбайт	HMC-EF172	1) Полностью вся программа пользователя. 2) Указанный диапазон памяти ввода/вывода. 3) Данные области параметров (настройки ПЛК и другие параметры).  Смотрите примечание 4.	Разрешены все операции (см. стр. 204)  Функцию автоматической загрузки при запуске нельзя использовать для загрузки данных из памяти файлов EM (описание см. на стр. 204)
		30 Мбайт	HMC-EF372		
		64 Мбайт	HMC-EF672		
Память файлов EM Область EM 	ОЗУ (RAM)	Емкость области EM модулей CPU Серия CS CS1H-CPU67H: 832 кбайт (Банки 0 ... C: E0_00000 ... EC_00000) Серия SJ SJ1H-CPU66H: 448 кбайт (Банки 0 ... 6: E0_00000 ... E6_00000)	От указанного банка области EM памяти ввода/вывода до последнего банка (указывается в настройках ПЛК)		

- Примечание**
1. Описание установки и извлечения карт памяти приведено в разделе 5-2 *Управление файлами*.
  2. Если карта памяти или память файлов EM используется впервые, ее следует инициализировать. Описание процедуры инициализации приведено в *Разделе 5-3 Применение памяти файлов*.
  3. Для установки карты памяти в слот для ПЛК-карт персонального компьютера можно использовать адаптер для карт памяти HMC-AP001. Это позволяет применять карты памяти в качестве устройства хранения данных.

4. Если используется CX-Programmer, в этом случае модуль CPU может интерпретировать таблицы символов (включая комментарии к входам и выходам) и комментарии. Адресатом загрузки является карта памяти, когда она установлена, или память файлов EM, если карта памяти не установлена.

#### **Замечания по использованию карты памяти**

Прежде чем использовать карту памяти, проверьте следующие параметры.

##### **Форматирование**

Карты памяти поставляются с завода отформатированными, поэтому их не требуется форматировать после приобретения. Если требуется выполнить их форматирование после того, как они уже побывали в эксплуатации, эту операцию следует выполнять, вставив их в модуль CPU и используя CX-Programmer или консоль программирования.

Если карта памяти форматируется, будучи вставленной непосредственно в компьютер (ноутбук), модуль CPU может не распознать карту памяти. В этом случае карту памяти нельзя будет использовать, даже если она переформатируется в модуле CPU.

##### **Количество файлов в корневом директории**

Количество файлов, которое может находиться в корневом директории карты памяти, ограничено (точно так же, как и у жесткого диска). Хотя предельное значение зависит от типа и формата карты памяти, оно всегда находится в пределах от 128 до 512 файлов. Если в ваших приложениях предусматривается запись протокольных или других файлов с определенным интервалом, эти файлы следует записывать в поддиректории, но не в корневой директорий.

Поддиректории можно создавать с помощью компьютера или с помощью команды CMND (490). Пример использования CMND (490) можно найти в разделе 3-25-5 *DELIVER COMMAND: CMND(490)*, в руководстве *CS/CJ Series Instructions Reference*.

##### **Количество операций записи**

В принципе количество операций записи во флеш-память не ограничено. Что касается карт памяти, с точки зрения гарантированного срока службы количество операций записи ограничено цифрой 100 000. Например, если запись в карту памяти осуществляется каждые 10 минут, предельное количество операций записи 100 000 будет превышено по истечении двух лет.

##### **Минимальный размер файла**

Если в карте памяти хранится очень много маленьких файлов, например, таких файлов, которые содержат лишь несколько слов данных области DM, карту памяти нельзя будет использовать в полном объеме. Например, если используется карта памяти с размером кластера 4096 байтов, в этом случае каждый файл, независимо от того, насколько он мал, будет занимать, по меньшей мере, 4096 байтов памяти. Если в карте памяти хранится 10 слов данных области DM, будет использовано 4096 байтов памяти, хотя фактически размер файла составляет всего лишь 68 байтов. Использование файлов такого небольшого размера существенно снижает эффективность использования карты памяти. С другой стороны, уменьшение размера кластера с целью повышения эффективности использования карты памяти приводит к снижению скорости доступа к карте памяти.

Размер кластера карты памяти можно проверить с помощью команды CHKDSK в командной строке DOS. Эта процедура в настоящем руководстве не описывается. Подробную информацию о назначении размеров кластеров можно найти в справочной литературе по использованию компьютеров.

##### **Указания по обращению к карте памяти.**

Когда ПЛК обращается к карте памяти, на модуле CPU светится индикатор BUSY. Необходимо соблюдать следующие указания.

- 1,2,3...**
1. Никогда не выключайте напряжение питания модуля CPU, когда светится индикатор BUSY. В противном случае карта памяти может стать непригодной для использования.

2. Никогда не извлекайте карту памяти из модуля CPU, когда светится индикатор BUSY. Прежде чем извлечь карту памяти, нажмите кнопку выключения питания карты памяти и дождитесь, пока погаснет индикатор BUSY. В противном случае карта памяти может стать непригодной для дальнейшего использования.
3. Вставляйте карту памяти, располагая ее меткой направо. Не пытайтесь вставлять карту памяти каким-нибудь иным образом. Это может привести к повреждению карты памяти или модуля CPU.
4. Модулю CPU требуется несколько секунд, чтобы распознать карту памяти после ее установки. В случае обращения к карте памяти сразу же после включения напряжения питания или установки карты памяти следует использовать NC (нормально-замкнутое) условие для флага "Карта памяти распознана" (A34315) в качестве входного условия (см. рис. ниже).



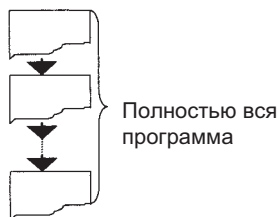
## 5-1-2 Данные, сохраняемые в файлах

Ниже перечислены файлы, которые могут быть записаны в модуль CPU из средства программирования (CX-Programmer или консоль программирования), а также с помощью команд FINS, команд лестничной диаграммы или специальных управляющих битов:

- Файлы программ
- Файлы данных
- Файлы параметров

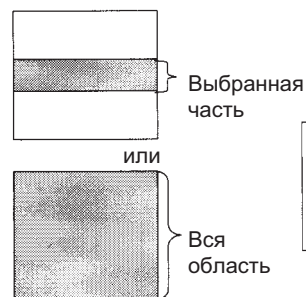
### Программа пользователя: файл программы

Полностью вся программа, включая атрибуты задач



### Выбранный диапазон памяти ввода/вывода: Файлы данных

Полностью весь диапазон или выбранная часть одной области памяти



### Данные области параметров: Файл параметров

Первоначальные настройки, используемые для модуля CPU



**Примечание** Из CX-Programmer также могут быть записаны файлы трех следующих типов

- Файлы таблиц символов
- Файлы комментариев
- Файлы индексов (указателей) программы

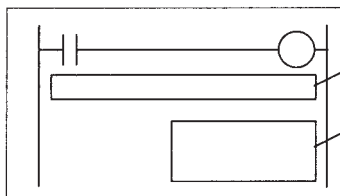
**Файлы таблиц символов**

Таблицы переменных, которые используются в CX-Programmer

Символы, адреса, типы данных, комментарии к входам/выходам

**Файлы комментариев**

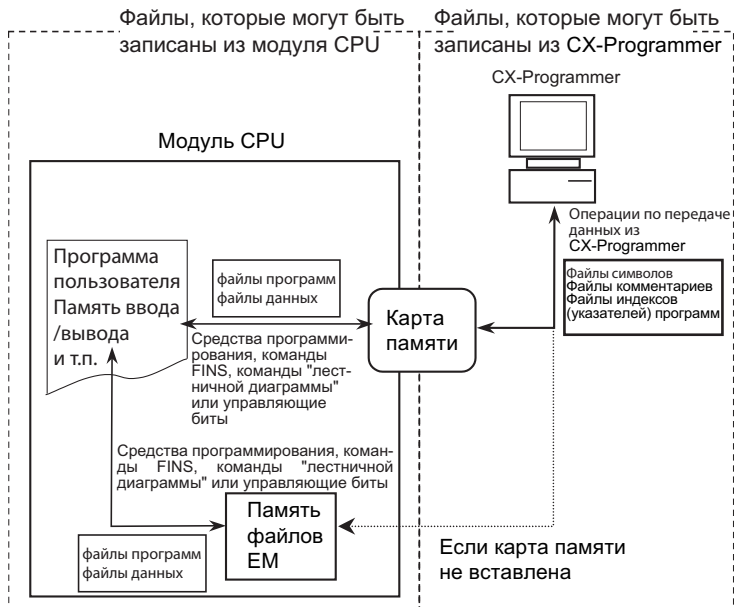
Комментарии, которые используются в CX-Programmer



**Файлы индексов программ**

Информация о разделах программы (используется в CX-Programmer)

Названия разделов, комментарии к разделам



**Примечание** Таблицы символов (символы, адреса и комментарии к входам/выходам) можно создавать в виде файлов на CX-Programmer.

Файл	Имя файла	Расширение	Содержание
Файл таблиц символов	SYMBOLS	.SYM	Глобальные и локальные символы
Файл комментариев	COMMENTS	.CMT	Комментарии к ветвям и комментарии (примечания)
Файл индексов (указателей) программ	PROGRAM	.IDX	Названия разделов и комментарии к разделам

Для проектов можно выполнять операции по передаче данных с помощью CX-Programmer с целью обмена всеми перечисленными файлами (файлами таблиц символов, файлами комментариев, файлами индексов (указателей) программ) между модулем CPU и картой памяти или памятью файлов EM (передача файлов индексов программ поддерживается, начиная с версии 2.0.). Обмен файлами таблиц символов и файлами комментариев также можно выполнять между CX-Programmer, ОЗУ (RAM) компьютера и устройством хранения данных, используя CX-Programmer версии 1.2. или выше.

### 5-1-3 Файлы

Файлы имеют формат, который используется в DOS, поэтому с ними можно работать, как с обычными файлами компьютера под управлением Windows.

Каждый файл идентифицируется с помощью имени и расширения (смотрите таблицу ниже). Имя файла может состоять из следующих символов: буквы A...Z, цифры 0...9, !, &, \$, #, ` , {, }, -, ^, (, ) и \_

В именах файлов нельзя использовать следующие символы: ;, /, \, ¥, ?, \*, ", ;, ;, <, >, =, +, "пробел" и 2-байтовые символы.

Расширение в имени файла зависит от типа сохраняемого файла. Файлы данных могут иметь расширение IOM, TXT, CSV или IOR (расширения TXT, CSV, IOR не поддерживаются модулями CPU серии CS1 до версии EV1). Файлы программ имеют расширение OBJ, а файлы параметров имеют расширение STD. Для размещения файлов памяти можно выбрать конкретный директорию; каждый директорию может иметь до пяти вложенных поддиректориев (включая корневого директорию).

#### Типы файлов, имена и расширения

Модуль CPU может управлять (читать и записывать) файлами трех типов.

- **Файлы общего назначения**

К этим файлам можно обращаться (читать или записывать) с помощью средств программирования, команд FINS, команд программы или операций над управляющими битами дополнительной области. Имена файлов могут устанавливаться пользователем произвольно.

- **Автоматическая загрузка файлов при запуске**

Эти файлы автоматически загружаются из карты памяти в модуль CPU, когда включается напряжение питания. В случае загрузки файлов параметров имя файла должно быть AUTOEXEC или ATEXEC□□. Если загружаются не файлы параметров (только для модулей серии CS/CJ версии 2.0 или выше), имя файла должно быть REPLACE.

- **Резервные файлы** (не поддерживаются модулями CPU серии CS1 до версии EV1)

Эти файлы передаются между картой памяти и модулем CPU функцией резервного сохранения. Используются фиксированные имена файлов BACKUP□□.

#### Файлы общего назначения

В следующей таблице перечислены имена и расширения файлов общего назначения.

Тип	Имя <sup>1</sup>	Расширение	Описание	Пояснение	
Файл программы	*****	.OBJ	Полностью вся программа пользователя	• Все циклические задачи и задачи обработки прерываний, а также данные для задач для одного модуля CPU.	
Файл области параметров	*****	.STD	Настройки ПЛК, зарегистрированная таблица ввода/вывода, таблица маршрутизации, параметры модуля шины CPU <sup>3</sup> и т.п.	<ul style="list-style-type: none"> <li>• Включает все первоначальные настройки для одного модуля CPU.</li> <li>• Пользователю не требуется различать типы параметров в файле.</li> </ul>	
Файл данных	*****	.IOM	Указанная область памяти ввода/вывода	<ul style="list-style-type: none"> <li>• Данные, от начального до конечного слова, в виде отдельных слов (16 битов), расположенные в одной области.</li> <li>• Поддерживаются области CIO, HR, WR, AR, DM или EM.</li> </ul>	Двоичный формат
		.TXT			Текстовый (TXT) формат <sup>2</sup> (без разделения или с разделением табуляцией)
		.CSV			CSV-формат <sup>2</sup> (с разделением с помощью запятой)

- Примечание**
1. Имена файлов, указанные в таблице выше как "\*\*\*\*\*", могут состоять максимум из 8-ми ASCII-символов.
  2. Форматы файлов TXT и CSV не поддерживаются модулями CPU серии CS1 до версии EV1.

3. В качестве примера параметров модуля шины CPU можно привести таблицы логических связей (Data Link). Описание остальных параметров можно найти в соответствующих руководствах по эксплуатации конкретных модулей.

#### Файлы, автоматически загружаемые при запуске

В колонке File (Файл) перечислены файлы, которые должны присутствовать в карте памяти, чтобы была возможна автоматическая загрузка при запуске.

Существует два способа автоматической загрузки файлов при запуске: загрузка с файлом области параметров и загрузка без файла области параметров.

#### Загрузка с файлом области параметров

Тип	Имя <sup>1</sup>	Расширение	Описание	Пояснение	Файл
Файл программы	AUTOEXEC	.OBJ	Полностью вся программа	<ul style="list-style-type: none"> <li>Файл не обязательно должен присутствовать в карте памяти, даже если выбрана автоматическая загрузка при запуске.</li> <li>Все циклические задачи и задачи обработки прерываний, а также данные задач для одного модуля CPU.</li> <li>Загрузка будет невозможна, если в карте памяти также отсутствует файл области параметров (AUTOEXEC.STD).</li> </ul>	Требуется
Файл области параметров	AUTOEXEC	.STD	Настройки ПЛК, зарегистрированная таблица ввода/вывода, таблицы маршрутизации, параметры модуля шины CPU <sup>3</sup> и т.п.	<p>Файл должен присутствовать в карте памяти, если выбрана автоматическая загрузка при запуске.</p> <p>Содержит все первоначальные настройки для одного модуля CPU.</p> <p>Пользователю не обязательно различать типы параметров в файле.</p> <p>Данные первоначальных параметров будут автоматически сохранены по указанным адресам в модуле CPU при запуске.</p> <p>Файл области параметров не будет загружен, если в карте памяти содержится файл программы с именем REPLACE.OBJ.</p>	Требуется

Тип	Имя <sup>1</sup>	Расширение	Описание	Пояснение	Файл
Файл данных	AUTOEXEC	.IOM	Данные памяти ввода/вывода (содержат указанное количество слов данных, начиная с D20000.)	<ul style="list-style-type: none"> <li>Запись данных DM, начиная с D20000, в файл с именем AUTOEXEC.IOM.</li> <li>Во время запуска все данные файла будут загружены в область DM, начиная с D20000.</li> <li>Если выбрана автоматическая загрузка при запуске, наличие этого файла в карте памяти не обязательно.</li> </ul>	---
	ATEXECDM	.IOM	Данные памяти ввода/вывода <sup>2</sup> (содержат указанное количество слов данных, начиная с D00000.)	<ul style="list-style-type: none"> <li>Запись данных DM, начиная с D00000, в файл с именем ATEXECDM.IOM.</li> <li>Во время запуска все данные файла будут загружены в область DM, начиная с D00000.</li> <li>Если выбрана автоматическая загрузка при запуске, наличие этого файла в карте памяти не обязательно.</li> </ul> <p><b>Примечание</b> Данные этого файла обладают наивысшим приоритетом, если они перекрываются с данными DM, содержащимися в AUTOEXEC.IOM.</p>	---
	ATEXECES□	.IOM	Данные области EM (банк□ <sup>2</sup> ) (содержат указанное количество слов данных, начиная с E□_00000.)	<ul style="list-style-type: none"> <li>Запись данных банка □ области EM, начиная с E□_00000, в файл с именем ATEXECES□.IOM. Максимальный номер банка зависит от модели используемого модуля CPU.</li> <li>При запуске все данные этого файла будут загружены в область EM, начиная с E□_00000.</li> <li>Если выбрана автоматическая загрузка при запуске, наличие этого файла в карте памяти не обязательно.</li> </ul>	---

## Загрузка без файла области параметров

Тип	Имя <sup>1</sup>	Расширение	Описание	Пояснение	Файл
Файл программы	REPLACE Примечание: Только для модулей CPU серии CS/CJ версии 2.0 или выше	.OBJ	Полностью вся программа пользователя	<ul style="list-style-type: none"> <li>Содержание полностью совпадает с файлом AUTOEXEC.OBJ.</li> <li>Этот файл будет загружен при запуске, даже если отсутствует файл области параметров (AUTOEXEC.STD)</li> </ul>	Требуется
Файл области параметров	Не требуется	---	---	Файл области параметров не будет загружен независимо от имени файла.	---
Файл данных	REPLACE Примечание: Только для модулей CPU серии CS/CJ версии 2.0 или выше	.IOM	Данные памяти ввода /вывода (содержат указанное количество слов данных, начиная с D20000.)	<ul style="list-style-type: none"> <li>Содержание полностью совпадает с файлом AUTOEXEC.IOM.</li> <li>Этот файл будет загружен при запуске, если в карте памяти имеется также файл программы с именем REPLACE.OBJ.</li> </ul>	---
	REPLCDM Примечание: Только для модулей CPU серии CS/CJ версии 2.0 или выше	.IOM	Данные памяти ввода /вывода (содержат указанное количество слов данных, начиная с D00000.)	<ul style="list-style-type: none"> <li>Содержание полностью совпадает с файлом ATEXECDM.IOM.</li> <li>Этот файл будет загружен при запуске, если в карте памяти имеется также файл программы с именем REPLACE.OBJ.</li> </ul>	---
	REPLCE□ Примечание: Только для модулей CPU серии CS/CJ версии 2.0 или выше	.IOM	Данные области EM (банк □) (содержат указанное количество слов данных, начиная с E□_00000.)	<ul style="list-style-type: none"> <li>Содержание полностью совпадает с файлом ATEXECE□.IOM.</li> <li>Этот файл будет загружен при запуске, если в карте памяти имеется также файл программы с именем REPLACE.OBJ.</li> </ul>	---

**Примечание**

1. Файлы, загружаемые автоматически при запуске, обязательно должны иметь имена AUTOEXEC или ATEXECE□□.
2. Файлы ATEXECDM.IOM и ATEXECE□.IOM не поддерживаются модулями CPU серии CS1 до версии EV1.
3. В качестве примера параметров модуля шины CPU можно привести таблицы логических связей. Информацию о других настраиваемых данных для конкретных модулей можно найти в соответствующих руководствах по эксплуатации.



**Резервные файлы  
(не поддерживаются  
модулями CPU серии  
CS1 до версии EV1)**

В следующей таблице перечислены файлы, которые создаются автоматически в целях создания резервной копии данных при передаче данных из/в карту памяти.

Тип	Имя <sup>1</sup>	Расширение	Описание	Пояснение
Файл данных	BACKUP	.IOM	Слова области DM, отведенные для специальных модулей ввода/вывода, модулей шины CPU и встраиваемых плат (только серия CS)	<ul style="list-style-type: none"> <li>Содержит данные DM в диапазоне D20000 ... D32767</li> <li>При чтении данных из карты памяти во время резервного сохранения этот файл должен присутствовать в карте памяти.</li> </ul>
	BACKUIO	.IOR	Области данных памяти ввода/вывода	<ul style="list-style-type: none"> <li>Содержит все данные областей данных CIO, WR, HR и AR, а также флаги завершения и текущие значения (PV) таймеров/счетчиков<sup>2</sup></li> <li>При чтении данных из карты памяти во время резервного сохранения этот файл должен присутствовать в карте памяти.</li> </ul>
	BACKUPDM	.IOM	Область DM общего назначения	<ul style="list-style-type: none"> <li>Содержит данные DM в диапазоне D00000 ... D19999.</li> <li>При чтении данных из карты памяти во время резервного сохранения этот файл должен присутствовать в карте памяти.</li> </ul>
	BACKUPE□	.IOM	Область EM общего назначения	<p>Содержит все данные EM банка □ области EM в диапазоне адресов E□_00000...E□_32767. (максимальный номер банка зависит от модели используемого модуля CPU).</p> <p>При чтении данных из карты памяти во время резервного сохранения этот файл должен присутствовать в карте памяти.</p> <ul style="list-style-type: none"> <li>При резервном сохранении данных в карту памяти все данные каждого банка EM автоматически записываются в отдельные файлы.</li> </ul>
Файл программы	BACKUP	.OBJ	Полностью вся программа пользователя	<ul style="list-style-type: none"> <li>Содержит все циклические программы и программы задач обработки прерываний, а также данные задач для одного модуля CPU</li> <li>При чтении данных из карты памяти во время резервного сохранения этот файл должен присутствовать в карте памяти.</li> </ul>
Файл параметров		.STD	Настройки ПЛК, зарегистрированная таблица ввода/вывода, таблицы маршрутизации, параметры модуля шины CPU <sup>3</sup> и т.п.	<ul style="list-style-type: none"> <li>Содержит все первичные настройки для одного модуля CPU.</li> <li>Пользователю не требуется различать данные параметров в файле по типу.</li> <li>При чтении данных из карты памяти во время резервного сохранения этот файл должен присутствовать в карте памяти.</li> </ul>
Резервные файлы модуля/платы (только для модулей CPU CS1-H, CJ1-H или CJ1M)	BACKUP□□ (где □□- адрес модуля/платы, для которого выполняется резервное сохранение данных)	.PRM	Данные отдельного модуля или платы.	<ul style="list-style-type: none"> <li>Резервное хранение управляющих данных для одного модуля или платы. Подробное описание смотрите в 5-2-6 <i>Функция простого резервного сохранения данных</i>.</li> </ul>

- Примечание**
1. Файлы, которые используются для резервного хранения данных, обязательно должны иметь имя BACKUP□□.
  2. Данные области CIO, области WR, флаги завершения и текущие значения (PV) таймеров/счетчиков, а также данные, принудительно установленные/сброшенные, которые читаются из карты памяти при запуске, будут обнулены. Эти данные можно сохранить с помощью следующих параметров в настройках ПЛК: "Состояние бита удержания IOM при запуске" и "Состояние бита сохранения принужденных состояний при запуске".
  3. Одним из примеров параметров модуля шины CPU являются таблицы логических связей. Сведения о других параметрах можно найти в руководствах по эксплуатации соответствующих модулей.

**Директории**

В ПЛК серии CS/CJ предусмотрена возможность обращения к файлам, находящимся в поддиректориях, однако консоли программирования могут обращаться только к файлам, которые расположены в корневом директории. Полное имя файла (путь к файлу) не должно превышать 65 символов. При создании поддиректориев в карте памяти с помощью таких программ, как Windows, следите за тем, чтобы не превысить максимальное количество символов.

**Размеры файлов**

В следующей таблице приведены формулы, с помощью которых можно определить размеры файлов в байтах.

Тип файла	Размер файла
Файлы данных (.IOM)	(Количество слов x 2) + 48 байт Пример: Вся область DM целиком (D00000...D32767) (32 768 слов x 2) + 48 = 65584 байт
Файлы данных (.TXT или .CSV)	Размер файла зависит от количества разделителей и символов возврата каретки ("ввод"). Код разделителя занимает 1 байт, а код "возврат каретки" - 2 байта. Пример 1: Слова без разделителей, без "возврата каретки" 123456789ABCDEF012345678 занимают 24 байта. Пример 2: Разделенные слова, "возврат каретки" через каждые 2 поля 1234,5678 ↵ 9ABC,DEF0 ↵ 1234,5678 ↵ занимают 33 байта Пример 3: Разделенные двойные слова, "возврат каретки" через каждые 2 поля 56781234,DEF01234 ↵ 56781234 ↵ занимают 29 байт
Файлы программ (.OBJ)	(Количество используемых шагов x 4) + 48 байт (см. примечание).
Файлы параметров (.STD)	16048 байт

**Примечание** Чтобы рассчитать количество шагов в файле программы, отнимите имеющиеся шаги UM из общего количества шагов UM. Эти значения можно посмотреть в "Перекрестном отчете" (Cross-Reference Report) в CX-Programmer. Подробное описание смотрите в *CX-Programmer Operation Manual*.

**Файлы данных****Файлы общего назначения**

- 1,2,3... 1. Файлы с данными общего назначения имеют расширения IOM, TXT или CSV (TXT- и CSV- файлы: не поддерживаются модулями CPU серии CS1 до версии EV1).

Расширения	Формат данных	Содержание		Слова/поле
.IOM	Двоичный	Формат данных серии CS/CJ		---
.TXT (Смотрите примечания)	Слова без разделителей	ASCII формат	Формат получается путем преобразования полей длиной в одно слово в памяти ввода/вывода (4 шестнадцатеричных разряда) в формат ASCII и объединением полей без разделителей. Отдельные записи (строки) можно разделять с помощью "возврата каретки".	1 слово
	Двойные слова без разделителей		Формат создается путем преобразования полей, состоящих из двух слов, в памяти ввода/вывода (8 шестнадцатеричных разрядов), в формат ASCII и объединением полей без разделителей. Отдельные записи (строки) можно разделять с помощью "возврата каретки".	2 слова
	Слова, разделенные табуляторами		Формат получается путем преобразования полей длиной в одно слово в памяти ввода/вывода (4 шестнадцатеричных разряда) в формат ASCII и разделением полей с помощью табуляторов. Отдельные записи (строки) можно разделять с помощью "возврата каретки".	1 слово
	Двойные слова, разделенные табуляторами		Формат создается путем преобразования полей, состоящих из двух слов, в памяти ввода/вывода (8 шестнадцатеричных разрядов) в формат ASCII и разделением полей с помощью табуляторов. Отдельные записи (строки) можно разделять с помощью "возврата каретки".	2 слова
.CSV (Смотрите примечания)	Слова, разделенные запятыми		Формат получается путем преобразования полей длиной в одно слово в памяти ввода/вывода (4 шестнадцатеричных разряда) в формат ASCII и разделением полей с помощью запятой. Отдельные записи (строки) можно разделять с помощью "возврата каретки".	1 слово
	Двойные слова, разделенные запятыми		Формат создается путем преобразования полей, состоящих из двух слов, в памяти ввода/вывода (8 шестнадцатеричных разрядов) в формат ASCII и разделением полей с помощью запятой. Отдельные записи (строки) можно разделять с помощью "возврата каретки".	2 слова

- Примечания
- Чтение и запись TXT- и CSV- файлов данных:  
TXT- и CSV- файлы данных можно записывать и читать только с помощью FREAD(700) и FWRT(701).
  - Замечание относительно символов:  
Если TXT- или CSV- файл содержит какие-либо другие символы, помимо символов с шестнадцатеричным кодом (0...9, A...F или a...f), данные будут записаны в память ввода/вывода некорректно.
  - Замечание относительно размера поля:  
Если используются слова, данные в память ввода/вывода будут записаны некорректно, если TXT- или CSV- файл содержит поля, не являющиеся 4-разрядными шестнадцатеричными значениями. Аналогично, если используются двойные слова, данные в память ввода/вывода будут записаны некорректно, если TXT- или CSV- файл содержит поля, не являющиеся 8-разрядными шестнадцатеричными значениями.
  - Порядок записи:  
Если используются слова, данные памяти ввода/вывода преобразуются в ASCII формат и записываются в виде полей длиной в одно слово в порядке возрастания адресов в памяти ввода/вывода (от самого младшего к самому старшему).  
Если используются двойные слова, данные памяти ввода/вывода преобразуются в ASCII формат и записываются в поля длиной в два слова в порядке возрастания адресов в памяти ввода/вывода (от самого младшего к самому старшему). (внутри поля, состоящего из двух слов, первым хранится слово со старшим адресом, а затем - слово с младшим адресом.)

## е) Разделители:

В отсутствии разделителей поля объединяются и записываются в виде непрерывной последовательности. В случае разделения с помощью запятых между полями размещаются запятые, после чего поля записываются. В случае разделения с помощью символа табуляции между полями вставляются символы табуляции (Табуляторы), после чего поля записываются. Если в команде FREAD(700) указано использование разделителей (запятой или знаков табуляции), данные при чтении интерпретируются как данные с разделителями длиной в одно слово (с запятыми или табуляторами).

## f) Символы "возврат каретки":

Если "возврат каретки" не используется, данные хранятся в виде непрерывной последовательности. Если "возвраты каретки" используются, код "возврат каретки" вставляется после указанного количества полей. Если внутри файла используются "возвраты каретки", в командах FREAD(700)/FWRITE(701) нельзя указать смещение относительно начала файла (начальное слово для чтения или записи).

## g) Количество полей:

Общий объем данных в файле зависит от количества полей (количества записанных элементов, которое указано в команде FWRITE(701), и от количества слов в поле. Слово состоит из одного поля в случае однословных полей, и из двух слов - в случае двухсловных полей.

2. В файлах данных не предусмотрена информация о том, какие данные в них записаны, т.е., какая область памяти сохранена. Файлам следует давать имена, отражающие их содержание (см. примеры ниже), чтобы упростить дальнейшую работу с файлами.

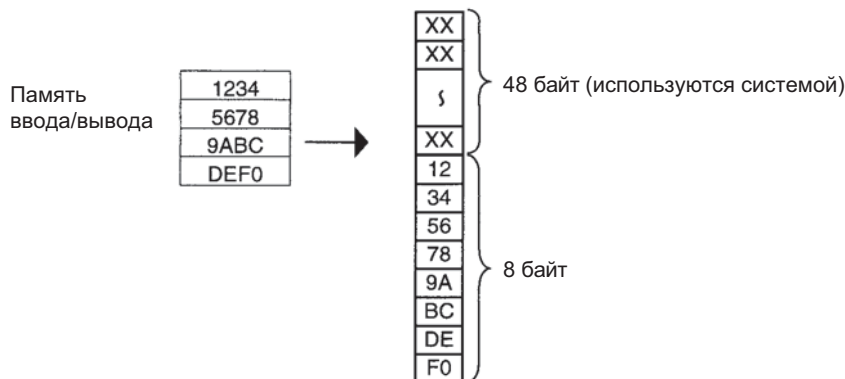
Примеры: D00100.IOM, CIO0020.IOM

Данные, расположенные в начале файла, будут записаны в память ввода/вывода, начиная с указанного адреса, даже если изначально данные, записанные в файл данных (IOM, TXT или CSV), находились не в этой области. Например, если данные области CIO файла записываются в область DM с помощью средства программирования, данные будут считаны в область DM модуля CPU без уведомления о том, что они оказались в другой области.

**Примечание** Файлы данных в формате TXT и CSV содержат шестнадцатеричные данные (0 ... 9, A ... F), благодаря чему можно передавать численные данные между памятью ввода/вывода и программами обработки табличных данных.

### Структура файла данных IOM

На следующем рисунке показана структура двоичных данных файла данных (ABC.IOM), содержащего 4 слова из памяти ввода/вывода: 1234 Hex, 5678 Hex, 9ABC Hex и DEFO Hex. Как правило, пользователю не требуется учитывать формат данных.



Содержание файла ABC.IOM

### Структура файла данных CSV/TXT (с одиночными словами)

На следующем рисунке показана структура данных файла данных CSV (ABC.CSV) с однословными полями, содержащего 4 слова из памяти ввода/вывода: 1234 Hex, 5678 Hex, 9ABC Hex и DEFO Hex. TXT-файл с однословными полями имеет точно такую же структуру.



Содержание файла ABC.CSV

### Структура файлов данных CSV/ТХТ (Двойные слова)

На следующем рисунке показана структура данных CSV-файла (ABC.CSV) с двухсловными полями, содержащего 4 слова из памяти ввода/вывода: 1234 Hex, 5678 Hex, 9ABC Hex и DEF0 Hex. TXT-файл с двухсловными полями имеет точно такую же структуру.



Содержание файла ABC.CSV

### Создание файлов данных с помощью программ для работы с электронными таблицами

Для создания файлов данных формата TXT и CSV с помощью программ для работы с электронными таблицами, например, с помощью Microsoft Excel, выполните следующие действия.

- Укажите для ячейки формат данных "числовой" или "текстовый".
- В каждой ячейке введите 4 символа, если должны использоваться однословные поля, или 8 символов, если должны использоваться двухсловные поля. Например, в случае однословных полей введите 000A, а не просто A.
- Вводите только шестнадцатеричные символы (0...9, A...F или a...f). Другие символы и коды использовать нельзя.

Если требуется сохранить шестнадцатеричные данные в память ввода /вывода, удобно выполнить преобразование десятичных чисел, введенных в ячейки, в шестнадцатеричный формат. Для преобразования десятичных чисел в шестнадцатеричный формат выполните следующие действия.

1,2,3...

1. В меню **Tools** (Сервис) выберите **Add-Ins...** (Надстройки)
2. В меню **Add-Ins** (Надстройки) выберите **Analysis ToolPak** (Инструмент "Анализ").
3. В меню **Insert** (Вставка) выберите **Function** (Функция), предварительно выбрав ту ячейку, в которой будет использоваться функция.
4. В поле **Category** (Категория) выберите Engineering (Инженерные расчеты), в поле **Function** (Функция) выберите **DEC2HEX (number, digits)**.
5. В случае преобразования в 4-разрядное шестнадцатеричное число следует ввести: IF(0<=cell location,cell location,65535+cell location)  
( IF(0<=адрес ячейки,адрес ячейки,65535+адрес ячейки) )  
В случае преобразования в 8-разрядное шестнадцатеричное число следует ввести: IF(0<=cell location,cell location,4294967296+cell location)  
( IF(0<=адрес ячейки,адрес ячейки,4294967296+адрес ячейки) )

• **Пример 1:** ввод неотрицательных десятичных значений.

Параметр	Преобразование десятичного значения без знака в 4-разрядное шестнадцатеричное число	Преобразование десятичного значения без знака в 8-разрядное шестнадцатеричное число
Функция	DEC2HEX( <i>cell_location</i> ,4)	DEC2HEX( <i>cell_location</i> ,8)
Пример	<p>Введите десятичное число 10 и преобразуйте его в 000A (4-разрядное Hex).</p>	<p>Введите десятичное число 10 и преобразуйте его в 0000000A (8-разрядное Hex).</p>

• **Пример 2:** ввод десятичных значений со знаком.

Параметр	Преобразование десятичного значения со знаком в 4-разрядное шестнадцатеричное число	Преобразование десятичного значения со знаком в 8-разрядное шестнадцатеричное число
Функция	DEC2HEX(IF(0<= <i>cell_location</i> , <i>cell_location</i> ,65536+ <i>cell_location</i> ),4)	DEC2HEX(IF(0<= <i>cell_location</i> , <i>cell_location</i> ,4294967296+ <i>cell_location</i> ),8)
Пример	<p>Введите десятичное число -10 и преобразуйте его в FFF6 (4-разрядное Hex).</p>	<p>Введите десятичное число -10 и преобразуйте его в FFFFFFF6 (8-разрядное Hex).</p>

### Файлы данных, автоматически загружаемые при запуске

Имеется три типа файлов, которые автоматически загружаются при запуске, если используется функция автоматической загрузки при запуске.

- AUTOEXEC.IOM или REPLACE.IOM: слова DM, отведенные для специальных модулей ввода/вывода и встраиваемых плат. Содержимое этих файлов загружается в область DM, начиная с D20000, когда включается питание.
- ATEXECDM.IOM или REPLCDM.IOM: слова DM общего назначения. Содержимое этих файлов загружается в область DM, начиная с D00000, когда включается питание.
- ATEXEC□.IOM или REPLCE□.IOM: слова EM общего назначения. Содержимое этих файлов загружается в область EM, начиная с E□\_00000, когда включается питание.

При создании перечисленных выше файлов данных всегда следует указывать приведенный выше первый адрес (D20000, D00000 или E□\_00000) и следить за тем, чтобы размер файла не превышал емкость указанной области данных.

Все данные каждого файла всегда загружаются, начиная с указанного первого адреса (D20000, D00000 или E□\_00000).

### Примечание

1. Создавая файлы AUTOEXEC.IOM, ATEXECDM.IOM и ATEXEC□.IOM или файлы REPLACE.IOM, REPLCDM.IOM или REPLCE□.IOM с помощью средства программирования (консоли программирования или CX-Programmer), всегда указывайте надлежащий начальный адрес (D20000, D00000 или E□\_00000), а также следите за тем, чтобы размер файла не превышал емкость области DM или указанного банка EM. Содержимое файла всегда загружается, начиная с надлежащего начального адреса (D20000, D00000 или E□\_00000), даже если указано иное начальное слово, что может привести к перезаписи содержимого части области DM или банка EM неправильными данными. Кроме того, если емкость области DM или банка EM будет превышена (что возможно при выполнении настройки в CX-Programmer), оставшиеся данные будут записаны в банк 0 области EM, если окажется превышенной область DM, или в следующий банк EM, если окажется превышенным банк EM.

2. В случае использования CX-Programmer можно указать файл данных, который превысит максимальный адрес области DM D32767 или максимальный адрес области EM E□\_32767. Если файл AUTOEXEC.IOM выйдет за границы области DM, все оставшиеся данные будут записаны в область EM, начиная с E0\_00000, и в последующие ячейки, в порядке возрастания адресов и банков памяти, вплоть до самого последнего банка. Таким образом, имеется возможность автоматической загрузки данных при запуске и в область DM, и в область EM. Аналогичным образом, если файл ATEXECE□.IOM окажется больше, чем банк EM, оставшиеся данные будут записаны в последующие банки EM.
3. Системные параметры для специальных модулей ввода/вывода, модулей шины CPU и встраиваемой платы (только серия CS) можно изменять, используя различные файлы AUTOEXEC.IOM с различными значениями параметров для области специального модуля ввода/вывода (D20000...D29599), области модуля шины CPU (D30000...D31599) и области встраиваемой платы (только серия CS, D32000...D32099). Следовательно, карты памяти можно использовать для создания библиотек системных параметров для специальных модулей ввода/вывода, модулей шины CPU и встраиваемых плат (только серия CS) для различных систем или устройств.

#### Файлы для резервного сохранения данных

Функция резервного сохранения создает 4 типа файлов данных, описанных выше.

Чтобы выполнить резервное сохранение данных, следует перевести ключ 7 DIP-переключателя модуля CPU в положение ВКЛ, а ключ 8 - в положение ВЫКЛ, вставить карту памяти, после чего нажать и удерживать нажатой кнопку питания карты памяти в течение трех секунд. В результате автоматически будут созданы и записаны в карту памяти четыре резервных файла (BACKUP.IOM, BACKUPIO.IOR, BACKUPDM.IOM и BACKUPE□.IOM).

Четыре резервных файла служат исключительно для функции резервного сохранения, хотя три из них (BACKUP.IOM, BACKUPDM.IOM и BACKUPE□.IOM) можно создать с помощью средства программирования (BACKUPIO.IOR нельзя создать с помощью средства программирования).

### 5-1-4 Способы работы с файлами

В следующей таблице обобщенно представлено 6 способов, которые можно применять для чтения и записи файлов.

Чтение: загрузка файлов из памяти файлов в модуль CPU.

Запись: сохранение файла из модуля CPU в память файлов.

Способ	Носитель	Имя файла	Описание	Полностью вся программа	Данные области данных (см. прим.3)	Данные области параметров
Средства программирования (включая консоли программирования)	Карта памяти Память файлов EM	Любое допустимое имя файла	Чтение	ОК	ОК	ОК
			Запись	ОК	ОК	ОК
			Другие операции (см. прим. 2)	ОК	ОК	ОК
Команда FINS (см. прим.1)	Карта памяти Память файлов EM	Любое допустимое имя файла	Чтение	ОК	ОК	ОК
			Запись	ОК	ОК	ОК
			Другие операции (см. прим. 2)	ОК (см. прим. 4)	ОК	ОК
Команды FREAD(700) и FWRIT(701)	Карта памяти Память файлов EM	Любое допустимое имя файла	Чтение данных из одного файла	Невозможно	ОК	Невозможно
			Запись данных в один файл	Невозможно	ОК	Невозможно

Способ	Носитель	Имя файла	Описание	Полностью вся программа	Данные области данных (см. прим.3)	Данные области параметров
С помощью бита управления (дополнительная область) во время работы заменяется полностью вся программа (не поддерживается модулями CPU серии CS1 до версии EV1)	Карта памяти	Любое допустимое имя файла	Чтение	ОК	Невозможно	Невозможно
Автоматическая загрузка при запуске	Карта памяти	AUTOEXEC, ATEXEC□□, или REPLACE (См. прим. 5)	Чтение	ОК	ОК	ОК
			Запись	Невозможно	Невозможно	Невозможно
Резервное сохранение (не поддерживается модулями CPU серии CS1 до версии EV1)	Карта памяти	BACKUP□□,	Чтение	ОК	ОК	ОК
			Запись	ОК	ОК	ОК

**Примечание**

1. Команды FINS для работы с памятью файлов можно передавать с компьютерных станций, подключенных через Host Link, с другого ПЛК, подключенного в сеть (с помощью CMND (490)), или из программы локального ПЛК (с помощью CMND (490)). (Модули CPU серии CS1 до версии EV1 не поддерживают возможность управления памятью файлов с помощью CMND (490) в том же модуле CPU, для которого выполняется управление памятью файлов.)
2. Другие операции: форматирование памяти файлов, чтение данных из файлов, запись данных в файлы, изменение имени файла, чтение данных памяти файлов, удаление файлов, копирование файлов, создание поддиректория.
3. Файлы данных с расширением TXT или CSV можно читать или записывать только с помощью команд FREAD(700) и FWRT(701). Их нельзя читать или записывать с помощью средства программирования.
4. Для передачи файлов программ (.OBJ) между ОЗУ (RAM) компьютера и средством хранения данных можно использовать CX-Programmer версии 1.2 или выше.
5. Модули CPU серии CS/CJ версии 2.0 поддерживают автоматическую загрузку файлов в модуль CPU при запуске при отсутствии файла параметров в карте памяти. Это достигается путем изменения имени файла программы на REPLACE.OBJ. Файлы данных также могут быть загружены вместе с файлом REPLACE.OBJ в случае использования следующих имен файлов: REPLACE.IOM, REPLCDM.IOM и REPLCE□.IOM.

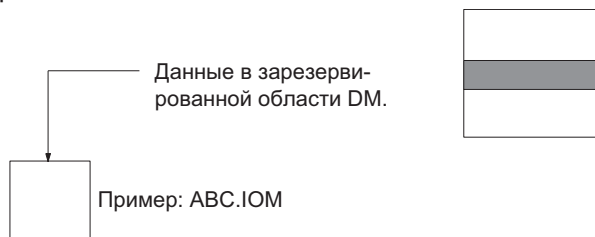


## 5-1-5 Примеры применения

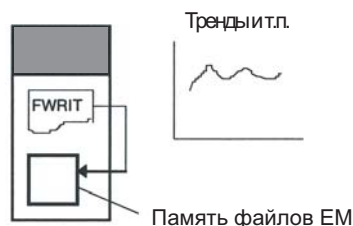
Ниже приведены примеры использования памяти файлов.

### Файлы данных

В этом примере параметры (для специальных модулей ввода/вывода, модулей шины CPU и встраиваемых плат (только серия CS)), хранящиеся в области DM, записываются в карту памяти. Если файл данных имеет имя AUTOEXEC.IOM, параметры, хранящиеся в файле, будут автоматически загружены при включении питания.



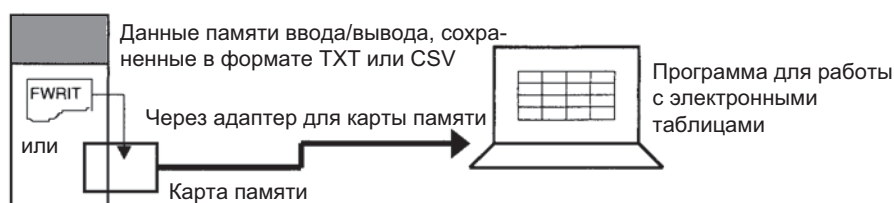
В этом примере рабочие данные (тренды, контроль качества и прочие данные), сформированные во время выполнения программы, сохраняются в память файлов EM с помощью команды WRITE DATA FILE (FWRIT(701)).



**Примечание** Данные, к которым обращение происходит часто, например, данные трендов, лучше сохранять в память файлов EM, а не в карту памяти.

### Файлы данных в формате ASCII (.TXT и .CSV)

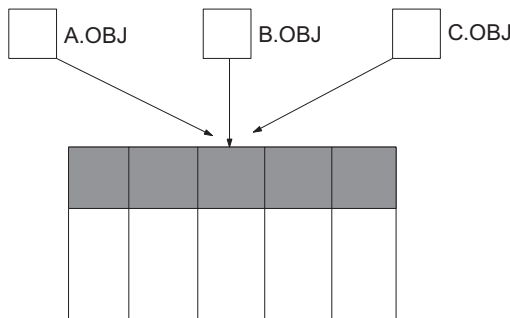
Производственные данные, сохраненные в карту памяти в формате TXT или CSV, могут быть переданы на персональный компьютер с помощью адаптера для карты памяти и отредактированы с помощью программ для работы с электронными таблицами (не поддерживается модулями CPU серии CS1 до версии EV1).



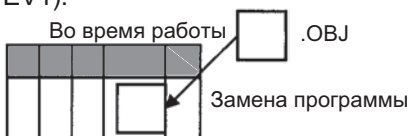
Аналогичным образом, такие данные, как параметры специального модуля ввода/вывода можно создавать с помощью программ для работы с электронными таблицами в формате TXT или CSV, сохранять их в карту памяти и загружать в модуль CPU с помощью FREAD (700) (не поддерживается модулями CPU серии CS1 до версии EV1).

**Файлы программ (.OBJ)**

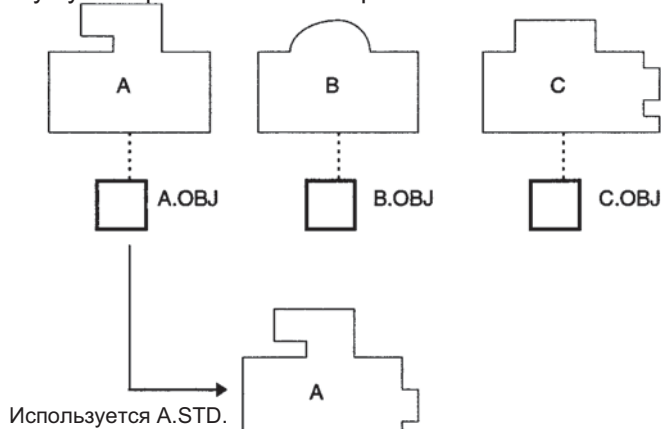
В этом примере программы, которые управляют различными процессами, сохраняются в отдельные карты памяти. Полностью всю конфигурацию ПЛК (программу, настройки ПЛК) можно заменить на другую, вставив другую карту памяти и используя функцию автоматической загрузки при запуске.



Полностью всю программу можно заменить во время работы из самой программы (без средства программирования), используя для этого бит управления дополнительной области (не поддерживается модулями CPU серии CS1 до версии EV1).

**Файлы области параметров .STD**

В этом случае в карты памяти записываются настройки ПЛК, таблицы маршрутизации, таблицы ввода/вывода и прочие данные определенных устройств или машин. Данные могут быть загружены в другое устройство или машину путем простой замены карты памяти.

**Резервные файлы**

Функцию резервного сохранения можно использовать для записи всех данных модуля CPU (полностью вся память ввода/вывода, программа и область параметров) в карту памяти без применения средства программирования. В случае возникновения проблем с данными модуля CPU сохраненные данные сразу же можно восстановить (не поддерживается модулями CPU серии CS1 до версии EV1).

**Файлы таблиц символов**

CX-Programmer можно использовать для сохранения символьных обозначений программы и комментариев к входам/выходам в файлы таблиц символов с именами SYMBOLS.SYM в карты памяти или в память файлов EM.

**Файлы комментариев**

CX-Programmer можно использовать для сохранения комментариев к ветвям программы в файлы комментариев с названием COMMENTS.CMT в карты памяти или в память EM.

## 5-2 Управление файлами

Ниже перечислены процедуры и средства, которые используются для чтения, записи и других операций с файлами.

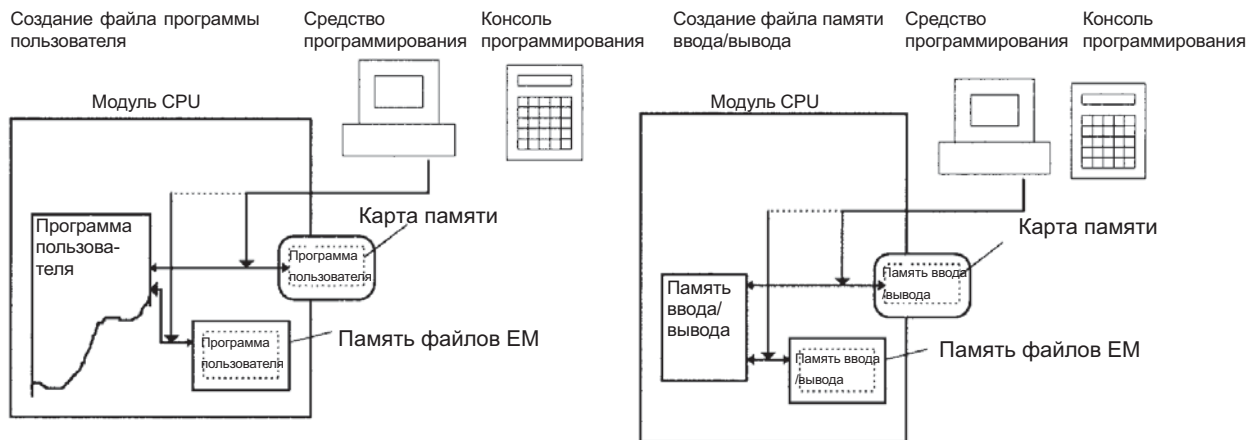
- Средства программирования
- Команды FINS
- Команды FREAD(700), FWRT(701) и CMND(490) в программе пользователя (CMND(490): не поддерживается модулями CPU серии CS1 до версии EV1).
- Замена всей программы целиком с помощью битов управления дополнительной области (не поддерживается модулями CPU серии CS1 до версии EV1).
- Автоматическая загрузка при запуске
- Функция резервного сохранения (не поддерживается модулями CPU серии CS1 до версии EV1)

### 5-2-1 Средства программирования (включая консоли программирования)

Средства программирования позволяют выполнять следующие операции.

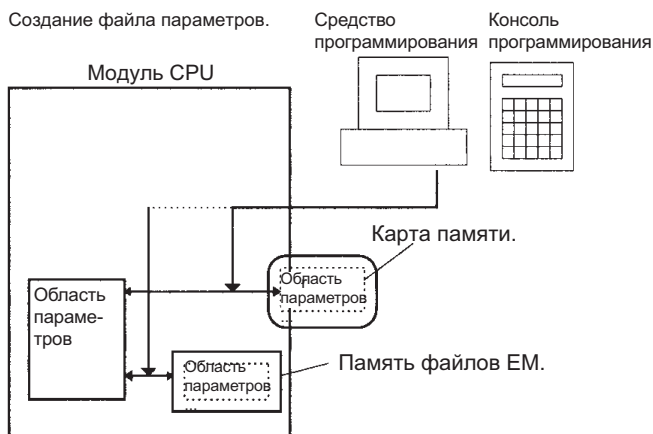
Операция	CX-Programmer	Консоль программирования
Чтение файлов (загрузка из памяти файлов в модуль CPU)	OK	OK
Запись файлов (загрузка из модуля CPU в память файлов)	OK (см. примечание)	OK (см. примечание)
Сравнение файлов (сравнение файлов модуля CPU и памяти файлов)	Невозможно	OK
Форматирование памяти файлов	Карты памяти	OK
	Файлы EM	OK
Изменение имен файлов	OK	Невозможно
Чтение данных памяти файлов	OK	Невозможно
Удаление файлов	OK	OK
Копирование файлов	OK	Невозможно
Удаление/создание субдиректориев	OK	Невозможно

**Примечание** Модуль CPU серии CS/CJ версии 2.0 или выше позволяет использовать пароль для защиты от чтения с целью предотвращения записи файла программы в память файлов (т.е., в карту памяти или память файлов EM).



**Примечание** 1. С помощью Windows Explorer можно создавать любые необходимые метки томов.

2. Для памяти файлов применяется быстрое форматирование системы Windows. В случае возникновения ошибок при форматировании карт памяти форматирование можно выполнить с помощью обычной команды *Format* (Форматирование) системы Windows.
3. Для файлов, записываемых с целью загрузки из модуля CPU в файл памяти, берутся время и дата часов модуля CPU.



**Примечание** В случае использования модуля CPU серии CS/CJ версии 2.0 или выше и CX-Programmer версии 4.0 или выше создание резервной копии файла программы (.OBJ) может быть запрещено, если для всей программы пользователя или отдельных задач устанавливается пароль. Подробное описание содержится в руководствах *CS Series PLC Operation Manual* и *CJ Series PLC Operation Manual*, раздел 1.4.2 Усиление защиты от чтения с помощью паролей, подраздел Защита программы от записи.

Карта памяти может быть вставлена в гнездо для ПЛК-карты в компьютере с помощью адаптера для карты памяти HMC-AP001 (заказывается отдельно). Установка карты памяти в компьютер позволяет другим программам (например, Windows Explorer) читать и записывать файлы этой карты.



**CX-Programmer**

При работе с памятью файлов соблюдайте следующую последовательность действий

- 1,2,3... 1. Щелкните дважды по пиктограмме Memory Card (Карта памяти) в окне Project (Проект), когда с модулем установлена связь (Online-режим). Отобразится окно Memory Card (Карта памяти).

2. Чтобы записать содержимое модуля CPU в память файлов, выберите область программы, область памяти ввода/вывода или область параметров в рабочей области проекта, выберите в меню File Memory (Память файлов) команду Transfer (Передать), после чего выберите запись в карту памяти или в память файлов EM.

**или** Чтобы загрузить содержимое памяти файлов в модуль CPU, выберите файл в памяти файлов и перетащите его мышкой на область программы, область памяти ввода/вывода или область параметров в рабочей области проекта.

**Примечание** Для создания и чтения файлов символьных таблиц (SYMBOLS.SYM) и файлов комментариев (COMMENTS.CMT) используйте команды для загрузки проекта, предусмотренные в CX-Programmer.

**Консоль программирования**

CLR	000000 CT00		
FUN	SHIFT	CONT #	0: Transfer 1: Verify
↓	0: Initialize 1: Delete		

Можно выполнять следующие операции

Параметр 1	Параметр 2	Параметр 3	Параметр 4	Параметр 5
0: Передача	0: Из ПЛК в карту памяти	Выберите OBJ, CIO, HR, WR, AR, DM, EM или STD.	Укажите начальный и конечный адреса для передачи	Тип носителя, имя файла
	1: Из карты памяти в ПЛК	Выберите OBJ, CIO, HR, WR, AR, DM, EM или STD	Укажите начальный и конечный адреса для передачи	Тип носителя, имя файла
1: Сравнение		Выберите OBJ, CIO, HR, WR, AR, DM, EM или STD	Укажите начальный и конечный адреса для сравнения	Тип носителя, имя файла
2: Инициализация		Введите 9713 (карта памяти) или 8426 (память файлов EM)	--	--
3: Удаление		Выберите OBJ, CIO, HR, WR, AR, DM, EM или STD	Тип носителя, имя файла	--

**Примечание** В следующей таблице перечислены типы файлов

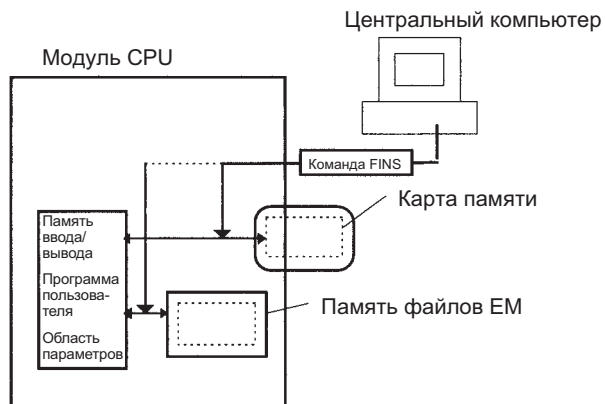
Символ	Тип файла	
OBJ	Файл программы (.OBJ)	
CIO	Файл данных (.IOM)	Область CIO
HR		Область HR
WR		Область WR
AR		Дополнительная область
DM		Область DM
EM0_		Область EM
STD	Файл параметров (.STD)	

### 5-2-2 Команды FINS

Ниже перечислены операции над памятью файлов, которые модуль CPU может выполнять, когда на него поступают соответствующие команды FINS. Они схожи с функциями средства программирования.

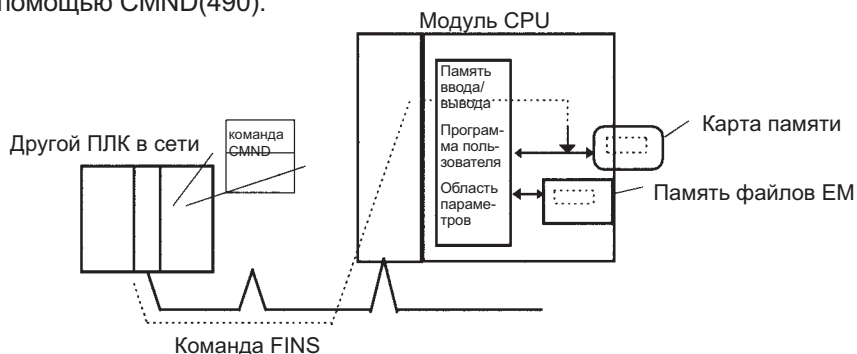
#### Команды FINS через Host Link

Компьютер, подключенный через систему Host Link, может передавать команду FINS, содержащую заголовок и признак завершения, предусмотренные в Host Link

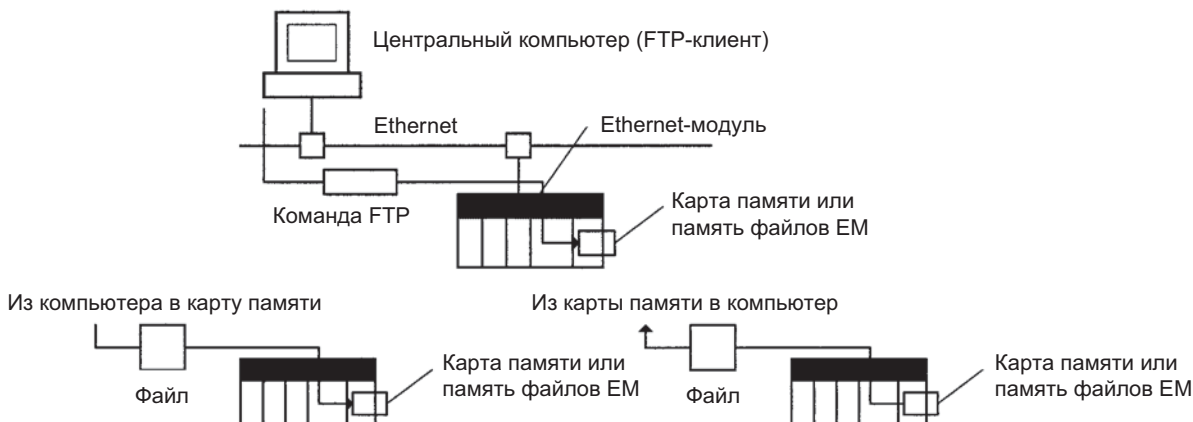


#### Команда FINS от другого ПЛК в сети

Другой ПЛК, находящийся в сети, может передать команду FINS с помощью CMND(490).



**Примечание** Компьютер в сети Ethernet может осуществлять чтение и запись из/в память файлов (в карты памяти или память файлов EM модуля CPU) через Ethernet-модуль. Данными в файлах можно обмениваться, если центральный компьютер выполняет функции FTP-клиента, а ПЛК серии CS/CJ функционирует как FTP-сервер.



В следующей таблице перечислены команды FINS, которые можно использовать для осуществления ряда функций, включая чтение и запись файлов.

Команда	Имя	Описание
2201 Hex	FILE NAME READ	Чтение данных из памяти файлов.
2202 Hex	SINGLE FILE READ	Чтение данных определенной длины, начиная с определенного места, в пределах одного файла.
2203 Hex	SINGLE FILE WRITE	Запись данных определенной длины, начиная с указанного места, в пределах одного файла.
2204 Hex	FILE MEMORY FORMAT	Форматирование (инициализация) памяти файлов.
2205 Hex	FILE DELETE	Удаление указанных файлов, хранящихся в памяти файлов.
2207 Hex	FILE COPY	Копирование файлов из одной памяти файлов в другую память файлов.
2208 Hex	FILE NAME CHANGE	Изменение имени файла.
220A Hex	MEMORY AREA FILE TRANSFER	Передача или сравнение данных в области памяти ввода/вывода и памяти файлов.
220B Hex	PARAMETER AREA FILE TRANSFER	Передача или сравнение данных в области параметров и памяти файлов.
220C Hex	PROGRAM AREA FILE TRANSFER	Передача или сравнение данных в области UM (Память пользователя) и памяти файлов.
2215 Hex	CREATE/DELETE SUBDIRECTORY	Создание и удаление поддиректориев.

**Примечание** Файлы, создаваемые в памяти файлов с помощью команд 220A, 220B, 220C и 2203, получают метку времени из внутренних часов модуля CPU.

### 5-2-3 FREAD(700), FWRT(701) и CMND(490)

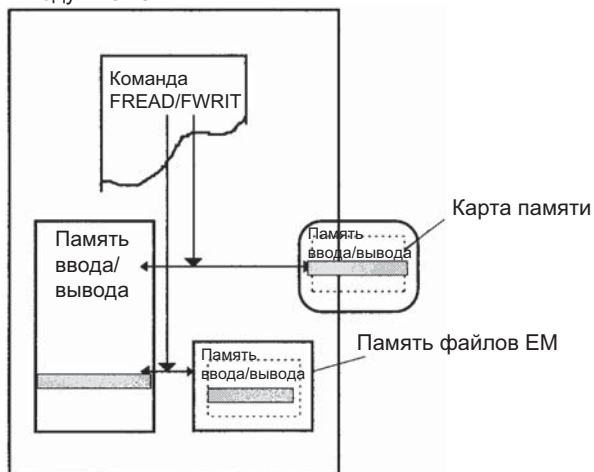
Для создания в карте памяти или в памяти файлов EM файла данных, содержащего указанные данные памяти ввода/вывода, можно использовать команду FWRT(701) (WRITE DATA FILE). Существующие файлы также можно дополнять или перезаписывать с любого места.

Команда FREAD(700) (READ DATA FILE) выполняет чтение данных памяти ввода/вывода, начиная с указанного места в файле данных, находящемся в карте памяти или в памяти файлов EM, и записывает эти данные по указанному месту в памяти ввода/вывода. Она может читать данные, начиная с любого места в указанном файле.

**Примечание** Эти команды не осуществляют передачу указанного файла, а передают указанный объем данных, начиная с указанного места в файле.

Для выполнения операций над файлами можно выполнить команду CDND(490) (DELIVER COMMAND), чтобы передать самому модулю CPU команду FINS. Над файлами в карте памяти или в памяти файлов EM можно выполнить такие операции, как форматирование, удаление, копирование и переименование файлов (не поддерживается модулями CPU серий CS1 до версии EV1).

FREAD(700)/FWRIT(701): Передача данных между памятью ввода/вывода и памятью файлов  
Модуль CPU



CMND(490): операции над памятью файлов  
(не поддерживаются модулями CPU серии CS до версии EV1)  
Модуль CPU



### Команды FREAD(700)/FWRIT(701)

FREAD(700) и FWRIT(701) служат для передачи данных между памятью ввода/вывода и памятью файлов. Все модули CPU серии CJ могут передавать двоичные данные (файлы .IOM), а модули CPU версии V1 также могут передавать файлы ASCII (файлы .TXT и .CSV).

Название	Мнемоническое обозначение	Описание
READ DATA FILE	FREAD(700)	Чтение указанных данных из файла данных или элементов данных и размещение их в память ввода/вывода
WRITE DATA FILE	FWRIT(701)	Использует указанные данные области памяти ввода/вывода для создания указанного файла данных.

**Передача файлов ASCII  
(не поддерживается  
модулями CPU серии CS1  
до версии EV1)**

Файлы в формате ASCII могут передаваться так же, как и двоичные файлы, поэтому в третьем и четвертом разрядах операнда "управляющее слово" (C) команды указывается тип передаваемого файла данных и количество полей, разделенных символом "возврат каретки".

Биты в C	Параметры	Ограничения, связанные со средством программирования
12...15	Тип данных 0: Двоичные (.IOM) 1: Слова без разделителей (.TXT) 2: Двойные слова без разделителей (.TXT) 3: Слова, разделенные запятыми (.CSV) 4: Двойные слова, разделенные запятыми (.CSV) 5: Слова, разделенные "табулятором" (.TXT) 6: Двойные слова, разделенные "табулятором" (.TXT)	Если используется CX-Programmer V1.1 или более старый, непосредственно можно указать только 0 Hex (файлы .IOM).  Если используется CX-Programmer V1.2 или более новый (или консоль программирования), в битах управляющего слова можно ввести значение 0 ... 6 Hex.
08...11	"Возвраты каретки" 0: Не используется 8: "Возврат каретки" через каждые 10 полей 9: "Возврат каретки" через каждое 1 поле A: "Возврат каретки" через каждые 2 поля B: "Возврат каретки" через каждые 4 поля C: "Возврат каретки" через каждые 5 полей D: "Возврат каретки" через каждые 16 полей	Если используется CX-Programmer V1.1 или более старый (или консоль программирования), непосредственно можно ввести только 0 Hex ("Возвраты каретки" не используются).  Если используется CX-Programmer V1.2 или более новый, в битах управляющего слова также можно ввести 0 Hex или 8 ... D Hex.



**СХ-Programmer V1.1 или более старый:****Косвенная настройка управляющего слова**

Если используется СХ-Programmer V1.1 или более старый, и в управляющем слове указана константа, определяющая тип данных и использование "возврата каретки", в этом случае с помощью команд FREAD(700) и FWRIT(701) нельзя передавать файлы в формате ASCII. Если используется константа, можно передавать только двоичные данные без "возвратов каретки".

В то же время файлы в формате ASCII можно передавать с помощью команд FREAD(700) и FWRIT(701), если управляющее слово настраивается косвенно. Следует записать требуемое значение управляющего слова в некоторое слово и указать это слово в качестве "управляющего слова" в команде FREAD(700) или FWRIT(701) (см. левую часть рисунка ниже).

СХ-Programmer версии V1.1 или старше



СХ-Programmer версии V1.2 или более новый



**Примечание** Для обозначения даты файлов, создаваемых в памяти файлов с помощью FWRIT(701), используется время внутренних часов модуля CPU.

Одновременно может выполняться только одна операция над памятью файлов, поэтому команды FREAD(700) и FWRIT(701) не должны выполняться, если над памятью файлов уже выполняется одна из следующих операций:

- 1,2,3...**
1. Выполняется FREAD(700) или FWRIT(701)
  2. Выполняется CMND(490) для передачи команды FINS модулю CPU
  3. Выполняется замена всей программы целиком с помощью операций над битом управления в дополнительной области
  4. Выполняется функция простого резервного сохранения

Чтобы предотвратить одновременное выполнение нескольких команд, предназначенных для работы с памятью файлов, т.е., не допустить выполнения операции над памятью файлов, когда уже выполняется другая операция, следует применять флаг "Операция над памятью файлов" (A34313).

Когда выполняется команда FREAD(700), установится флаг "Ошибка чтения файла" (A34310) и команда не будет выполнена, если указанный файл содержит недопустимый тип данных или данные файла повреждены. Текстовые файлы или CSV-файлы должны состоять из шестнадцатеричных кодов (соответствующих символам), через каждые 4 разряда (данные в формате слов) или через каждые 8 разрядов (данные в формате двойных слов) в них должны содержаться разделители. Данные будут прочитаны вплоть до того места, где будет обнаружен недопустимый символ.

## Сопутствующие биты/слова дополнительной области

Название	Адрес	Операция
Тип карты памяти	A34300 ... A34302	Указывает тип карты памяти, если она установлена.
Флаг "Ошибка формата памяти файлов EM"	A34306	ВКЛ, когда возникает ошибка формата первого банка EM, отведенного для памяти файлов. ВЫКЛ, когда форматирование завершается без ошибок.
Флаг "Ошибка формата карты памяти"	A34307	ВКЛ, когда карта памяти не отформатирована или произошла ошибка форматирования.
Флаг "Ошибка записи файла"	A34308	ВКЛ, когда произошла ошибка при записи в файл.
Флаг "Запись файла невозможна"	A34309	ВКЛ, когда данные не удалось записать из за того, что файл был защищен от записи или не хватило места в памяти файлов.
Флаг "Ошибка чтения файла"	A34310	ВКЛ, когда файл не удалось прочитать из за того, что данные были повреждены или файл содержал недопустимый тип данных.
Флаг "Файл отсутствует"	A34311	ВКЛ, когда файл не удалось прочитать из за отсутствия указанного файла.
Флаг "Операция над памятью файлов"	A34313	ВКЛ при одном из следующих условий: Модуль CPU выполняет команду FINS, переданную ему с помощью CMND(490). Выполняется FREAD(700) или FWRIT(701). Программа перезаписывается с помощью бита управления дополнительной области. Выполняется процедура простого резервного сохранения.
Флаг "Обращение к файлу"	A34314	ВКЛ, если происходит фактическое обращение к данным файла.
Флаг "Обнаружена карта памяти"	A34315	ВКЛ, если была обнаружена карта памяти (не поддерживается модулями CPU серии CS1 до версии EV1)
Количество передаваемых элементов	A346 ... A347	В этих словах указывается количество слов и полей, которые осталось передать (32 бита). Когда передается двоичный файл (.IOM), это количество уменьшается на один, всякий раз, когда читается слово. Когда передается текстовый или CSV файл, это число уменьшается на один, всякий раз, когда передается поле.

**CMND(490): DELIVER COMMAND**

Чтобы выполнить операции над памятью файлов, например, форматирование или удаление файлов, нужно использовать CMND(490), чтобы передать самому локальному модулю CPU команду FINS. Для передачи команды FINS, предназначенной для работы с памятью файлов, локальному ПЛК необходимо настроить управляющие слова в CMND(490) следующим образом:

- 1,2,3...**
1. Адрес сети назначения в C+2 установите равным 00 (локальная сеть).
  2. Адрес модуля назначения в C+3 установите равным 00 (модуль CPU в ПЛК), а адресуемый узел - равным 00 (локальный узел).
  3. Количество повторов в C+4 установите равным 0 (количество повторов не играет роли, поэтому его следует установить равным 0).

**Команды FINS, предназначенные для работы с памятью файлов**

Информацию о командах FINS можно найти в разделе 5-2-2 *Команды FINS*.

**Примечание** В приведенной ниже таблице перечислены не все команды FINS, которые имеют отношение к памяти файлов. Подробные сведения о командах FINS можно найти в руководстве *Communications Command Reference Manual (W342)*.

Команду CMND(490) нельзя выполнить в локальном модуле CPU, если в настоящий момент выполняется команда CMND(490) другого модуля CPU, выполняется команда FREAD(700) или FREAD(701), происходит замена программы пу-

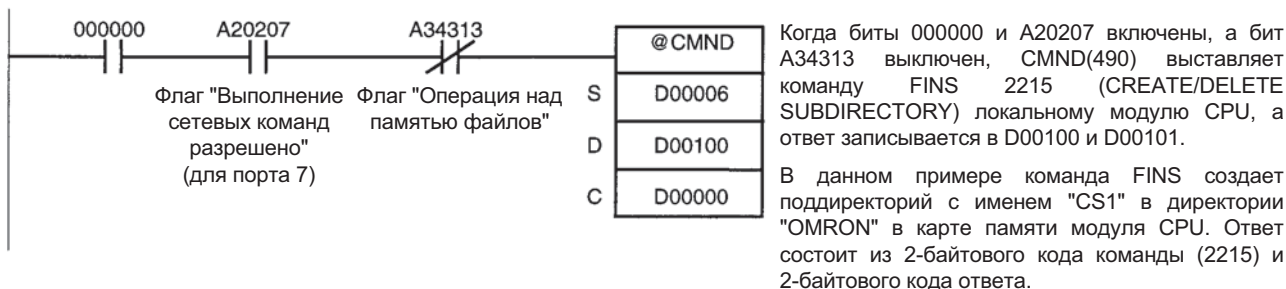
тем управления битом дополнительной области либо выполняется операция простого резервного сохранения. Чтобы предотвратить выполнение CMND(490) в момент выполнения другой операции над памятью файлов, следует применить в качестве нормально-замкнутого условия исполнения флаг "Операция над памятью файлов" (A34313).

Если CMND(490) не может быть выполнена для локального модуля CPU, включается флаг ошибки.

**Сопутствующие биты/слова дополнительной области**

Название	Адрес	Операнд
Флаг "Операция над памятью файлов"	A34313	ВКЛ, когда выполняется любое из следующих условий: <ul style="list-style-type: none"> <li>• Модуль CPU выполняет команду FINS, отправленную ему с помощью команды CMND(490).</li> <li>• Выполняется FREAD(700) или FWRT(701).</li> <li>• Программа переписывается с помощью бита управления дополнительной области.</li> <li>• Выполняется операция простого резервного сохранения.</li> </ul>
Флаг "Обнаружена карта памяти"	A34315	ВКЛ, если была обнаружена карта памяти (не поддерживается модулем CPU серии CS1 до версии EV1).

Ниже показан пример использования CMND(490) для создания поддиректория в карте памяти.



	15	8	7	0		
S:	D00006	2	2	1	5	Код команды: 2215 Hex (CREATE/DELETE SUBDIRECTORY) Номер диска: 8000 Hex (карта памяти) Параметры: 0000 Hex (создать поддиректорий)  Имя поддиректория: CS1 □□□□□□□□ (□: "пробел")  Длина директория: 0006 Hex (6 символов)  Путь: \OMRON
S+1:	D00007	8	0	0	0	
S+2:	D00008	0	0	0	0	
S+3:	D00009	4	3	5	3	
S+4:	D00010	3	1	2	0	
S+5:	D00011	2	0	2	0	
S+6:	D00012	2	0	2	0	
S+7:	D00013	2	E	2	0	
S+8:	D00014	2	0	2	0	
S+9:	D00015	0	0	0	6	
S+10:	D00016	5	C	4	F	
S+11:	D00017	4	D	5	2	
S+12:	D00018	4	F	4	E	

	15	8	7	0		
C:	D00000	0	0	1	A	Количество байтов в данных команды: 001A Hex (26 байтов) Количество байтов в данных ответа: 0004 Hex (4 байта) Адрес назначения: 0000 Hex (локальная сеть) 00 Hex (локальный узел) и 00 Hex (модуль CPU)
C+1:	D00001	0	0	0	4	
C+2:	D00002	0	0	0	0	
C+3:	D00003	0	0	0	0	
C+4:	D00004	0	7	0	0	
C+5:	D00005	0	0	0	0	Требуется ответ, порт связи 7, 0 повторов Время контроля ответа: FFFF Hex (6553.5 с)

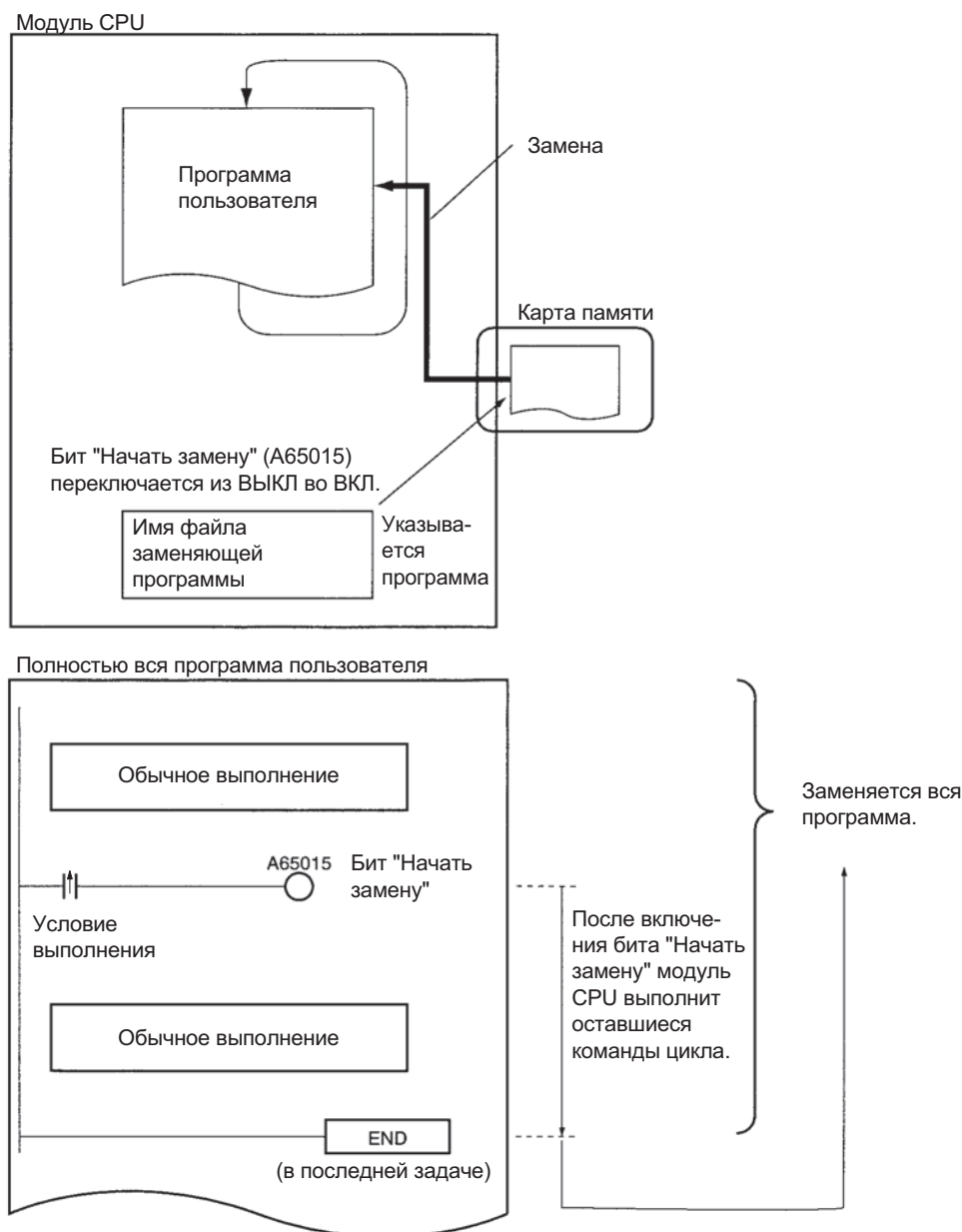
**Примечание** Помимо команд для выполнения операций над памятью файлов, перечисленных в таблице выше, также имеются другие команды FINS, которые можно передавать локальному ПЛК.

Чтобы избежать одновременного выполнения этих и других команд FINS, также следует использовать флаг "Операция над памятью файлов".

### 5-2-4 Замена полностью всей программы во время работы

(Не поддерживается модулем CPU серии CS1 до версии EV1)

Во время работы модуля CPU всю программу можно заменить целиком (в режиме RUN или MONITOR), установив (ВКЛ) бит "Начать замену" (A65015). Указанный файл будет прочитан из карты памяти и записан вместо выполняемой программы в конце текущего цикла. Параметры Program Password (Пароль программы) (A651) и Program File Name (Имя файла программы) (A654 ... A657) должны быть записаны заранее, указанный файл программы должен находиться в карте памяти. Только при этих условиях программу можно заменить во время работы.



Программу можно также заменить путем включения бита "Начать замену" со средства программирования, когда выполнение программы остановлено (режим PROGRAM) .

**Примечание** Файлы программы, предназначенные для замены, нельзя прочитать из памяти файлов EM.

Бит "Начать замену" (A65015) можно установить (ВКЛ) из любого места (адреса) в программе. После того, как бит "Начать замену" переключится из ВЫКЛ во ВКЛ, модуль CPU выполнит оставшиеся команды цикла.

Во время замены программа не выполняется. После того, как замена программы завершается, работа возобновляется вновь, как если бы модуль CPU был переключен из режима PROGRAM в режим RUN или MONITOR.

Замена программы происходит в конце цикла, в котором бит "Начать замену" переключается из ВЫКЛ во ВКЛ, то есть, после выполнения команды END(001) в последней задаче программы.

**Примечание** 1. Если требуется сохранить состояние данных памяти ввода/вывода после замены программы, следует включить бит "Удержание IOM" (A50012).

Если требуется сохранить состояние принудительно установленных и сброшенных битов после замены программы, следует включить бит "Удержание принудительных состояний" (A50013).

2. Если перед заменой программы включается бит "Удержание IOM" (A50012), состояние битов в памяти ввода/вывода сохраняется после замены программы. Необходимо обеспечить, чтобы исполнительные устройства работали надлежащим образом при сохранении состояния данных в памяти ввода/вывода.

Аналогичным образом, если перед заменой программы включается бит "Удержание принудительных состояний" (A50013), состояние принудительно установленных и сброшенных битов будет сохранено после замены программы. Необходимо обеспечить, чтобы исполнительные механизмы работали надлежащим образом при сохранении состояний принудительно установленных/сброшенных битов.

**Заменяющий файл** Файл программы с указанным именем (A654 ... A657) будет прочитан из карты памяти и заменит существующий файл в конце цикла, в котором бит "Начать замену" (A65015) переключается из ВЫКЛ во ВКЛ.

Файл	Имя и расширение файла	Указание имени заменяющего файла (*****)
Файл программы	*****.OBJ	Разместите имя файла заменяющей программы в слова A654 ... A657 перед заменой программы.

**Условие, необходимое для замены программы**

Чтобы заменить программу во время работы, необходимо соблюсти следующие условия.

- В A651 должен быть записан пароль программы (A5A5).
- В корневом директории карты памяти должен находиться файл программы с указанным именем (слова A654 ... A657).
- Модулем CPU должна быть обнаружена карта памяти (A34315 = ВКЛ).
- Не произошло фатальных ошибок.
- Не выполняются операции над памятью файлов (A34313 = ВЫКЛ).
- В область программы не записываются данные.
- Имеются права доступа (например, не производится загрузка данных из CX-Programmer в ПЛК).

**Примечание** Программу можно загрузить в любом режиме работы

**Работа CPU во время замены программы**

Во время замены программы модуль CPU работает следующим образом:

- Выполнение программы: остановлена
- Мониторинг длительности цикла: не производится

**Операции, которые продолжаются во время и после замены программы** Если включен бит "Удержание IOM" (A50012), будет сохранено состояние данных в следующих областях памяти: область CIO, рабочая область (W), флаги завершения таймеров (T), регистры индексов (IR), регистры данных (DR) и номер текущего банка EM.

**Примечание** При замене программы текущие значения таймеров будут обнулены.

Если при загрузке программы включен бит "Удержание IOM", исполнительные механизмы (нагрузки), которые были включены до замены программы, останутся включенными после замены. Следует обеспечить надлежащую работу этих исполнительных механизмов после замены программы.

Если включен бит "Удержание принудительных состояний" (A50013), состояние принудительно установленных и сброшенных битов сохраняется в процессе всей процедуры замены программы.

Прерывания маскируются.

Если выполняется протоколирование данных (взятие выборки), оно будет прекращено.

Условия выполнения команды (блокировки, точки останова и выполнение блоков программ) будут сброшены (инициализированы).

Флаги переключения состояний будут инициализированы (не зависимо от состояния бита "Удержание IOM" (ВКЛ или ВЫКЛ)).

**Работа после замены программы** Состояние циклических задач зависит от того, какой режим работы при запуске для них выбран в параметрах (они имеют тот же статус, что и при переключении ПЛК из режима PROGRAM в режим RUN/MONITOR).

В течение одного цикла после возобновления выполнения программы будет включен флаг "Первый цикл" (A20011) (такой же статус, как и при переключении ПЛК из режима PROGRAM в режим RUN/MONITOR).

**Время, необходимое для замены программы**

Размер всей программы	Время обслуживания периферии, установленное в настройках ПЛК	Приблиз. время, необходимое для замены программы
60 000 шагов	По умолчанию (4% от длительности цикла)	6 с
25 0000 шагов		25 с

#### Сопутствующие биты/слова дополнительной области

Название	Адрес	Работа
Флаг "Операция над памятью файлов"	A34313	ВКЛ при выполнении любого из следующих условий: Модуль CPU передал самому себе команду FINS с помощью CMND(490). Выполняется FREAD(700) или FWRIT(701). Программа перезаписывается с помощью управляющего бита дополнительной области (A65015). Выполняется операция простого резервного сохранения.
Флаг "Обнаружена карта памяти" (не поддерживается модулями CPU серии CS1 до версии EV1)	A34315	ВКЛ, если обнаружена карта памяти.
Бит "Удержание IOM"	A50012	Когда этот бит включен, содержимое памяти ввода/вывода сохраняется на протяжении всей процедуры замены программы.
Бит "Удержание принудительных состояний"	A50013	Когда этот бит включен, принудительно установленные и сброшенные биты сохраняют свои состояния на протяжении всей процедуры замены программы.

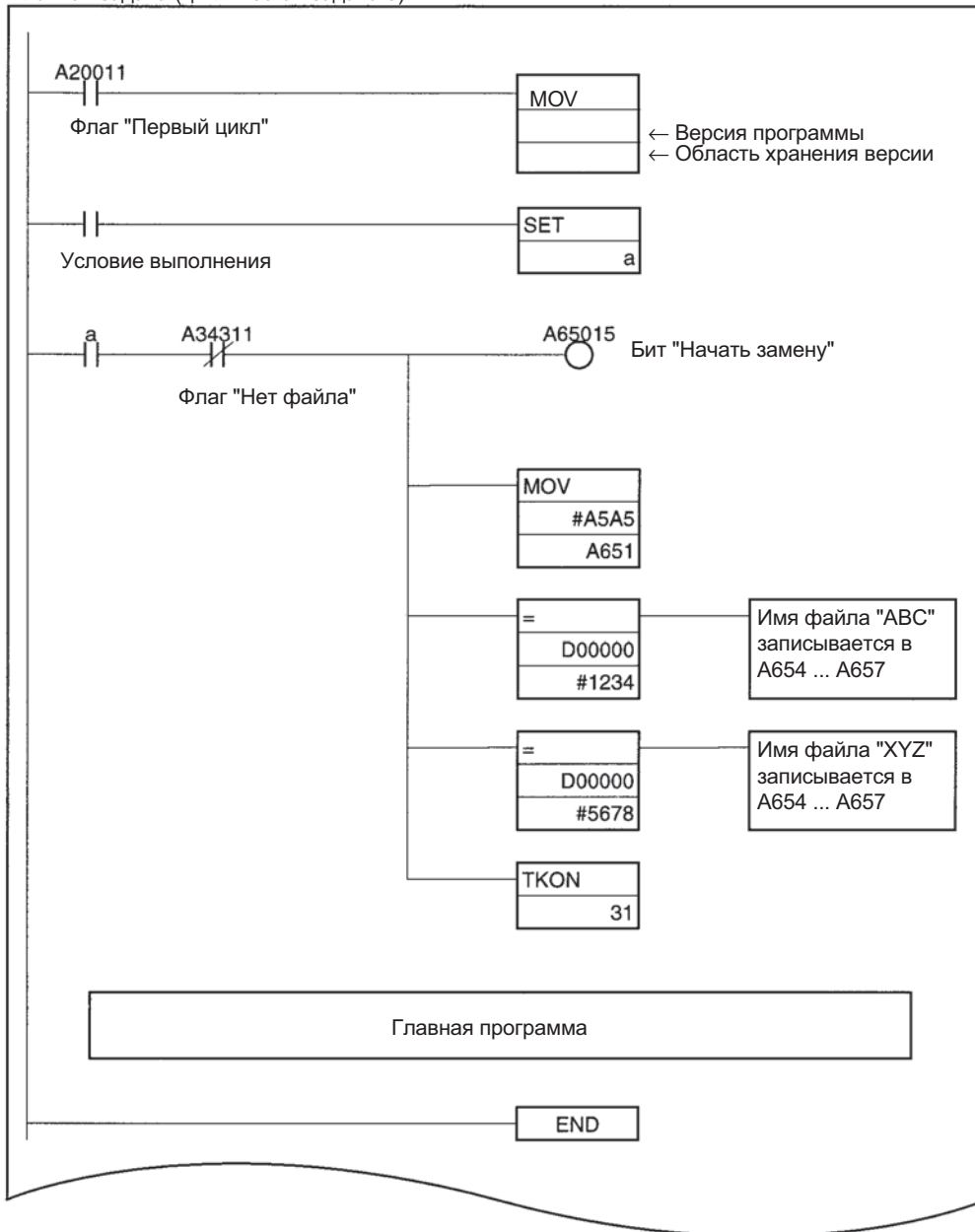
Название	Адрес	Работа															
Код "Завершение замены" (не поддерживается модулями CPU серии CS1 до версии EV1)	A65000 ... A65007	Код для завершения замены программы без ошибок (A65014 = ВЫКЛ): 01 Нех: Программа заменена файлом программы (.OBJ). Коды в случае завершения замены программы с ошибкой (A65014 = ВКЛ): 00 Нех: Произошла фатальная ошибка. 01 Нех: Произошла ошибка памяти. 11 Нех: Программа защищена от записи. 12 Нех: Пароль программы в A651 не верен. 21 Нех: Не установлена карта памяти. 22 Нех: Указанный файл не существует. 23 Нех: Указанный файл слишком велик (ошибка памяти). 31 Нех: Выполнялась одна из следующих операций: • Выполнялась операция с памятью файлов. • Записывалась программа. • Производилось изменение режима работы.															
Флаг "Ошибка замены" (не поддерживается модулями CPU серии CS1 до версии EV1)	A65014	Включается, если при попытке замены программы произошла ошибка, после того, как A65015 был переключен из ВЫКЛ во ВКЛ. Выключится в следующий раз, когда A65015 вновь перейдет из ВЫКЛ во ВКЛ.															
Бит "Начать замену" (не поддерживается модулями CPU серии CS1 до версии EV1)	A65015	Если этот бит был активизирован путем записи значения A5A5 Нех в качестве пароля программы (A651), замена программы будет начата, когда этот бит переключится из ВЫКЛ во ВКЛ. Во время замены программы этот бит нельзя вновь переключать из ВЫКЛ во ВКЛ. По завершению замены программы (с ошибкой или без) или при включении питания этот бит автоматически сбрасывается (ВЫКЛ). Состояние этого бита можно прочитать с помощью средства программирования, программируемого терминала или центрального компьютера. Это позволяет определить момент завершения программы.															
Пароль программы (не поддерживается модулями CPU серии CS1 до версии EV1)	A651	Чтобы программа могла быть заменена, в это слово должен быть записан пароль. A5A5 Нех: Активизация бита "Начать замену" (A65015). Другое значение: Запрет использования бита "Начать замену" (A65015). По завершению замены программы (с ошибкой или без) или при включении питания этот бит автоматически сбрасывается (ВЫКЛ).															
Имя файла программы (не поддерживается модулями CPU серии CS1 до версии EV1)	A654 ... A657	Прежде чем начать замену программы, в эти слова в формате ASCII следует записать имя файла заменяющей программы. Требуется записать лишь само имя файла из 8-ми символов, расширение .OBJ будет добавлено автоматически. Символы следует записывать, начиная со слова A654 (начиная с самого старшего байта). Если имя файла содержит меньше 8-ми символов, в остальные байты следует записать "пробелы" (20 Нех). Внутри самого имени файла не должно быть символов NULL или "пробелов". Ниже показан пример содержимого байтов для файла программы ABC.OBJ: <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th></th> <th>15</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>A654</td> <td>41</td> <td>42</td> </tr> <tr> <td>A655</td> <td>43</td> <td>20</td> </tr> <tr> <td>A656</td> <td>20</td> <td>20</td> </tr> <tr> <td>A657</td> <td>20</td> <td>20</td> </tr> </tbody> </table>		15	0	A654	41	42	A655	43	20	A656	20	20	A657	20	20
	15	0															
A654	41	42															
A655	43	20															
A656	20	20															
A657	20	20															

**Пример программы 1**

Сохраните файлы программ ABC.OBJ и XYZ.OBJ в карту памяти и выберите одну из этих программ (определяется значением в D00000). Запишите в D00000 значение #1234, чтобы выбрать ABC.OBJ, или значение #5678, чтобы выбрать XYZ.OBJ.

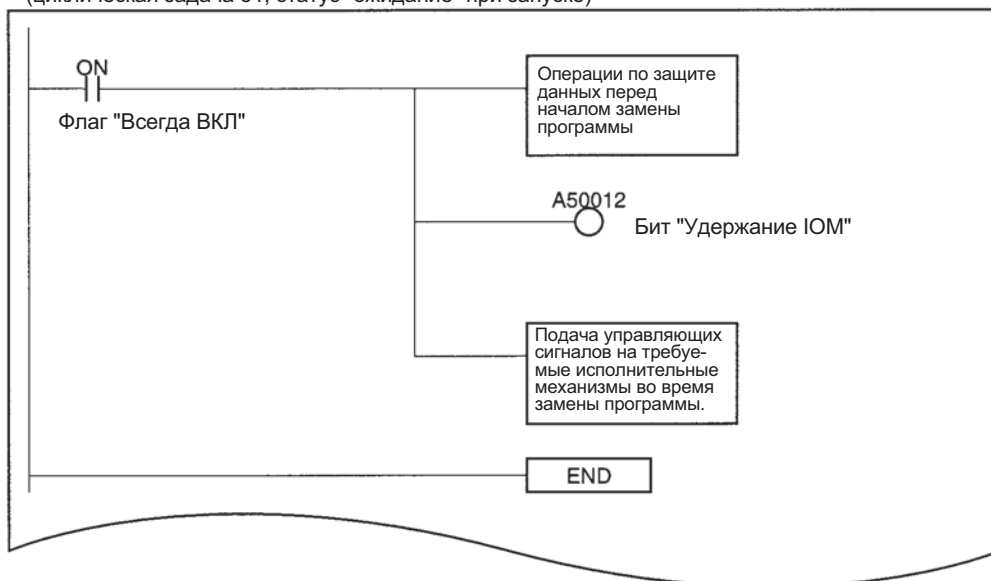
Запустите и выполните другую задачу, чтобы реализовать любые операции, необходимые перед заменой программы, или обработку бита "Удержание IOM".

Главная задача (циклическая задача 0)





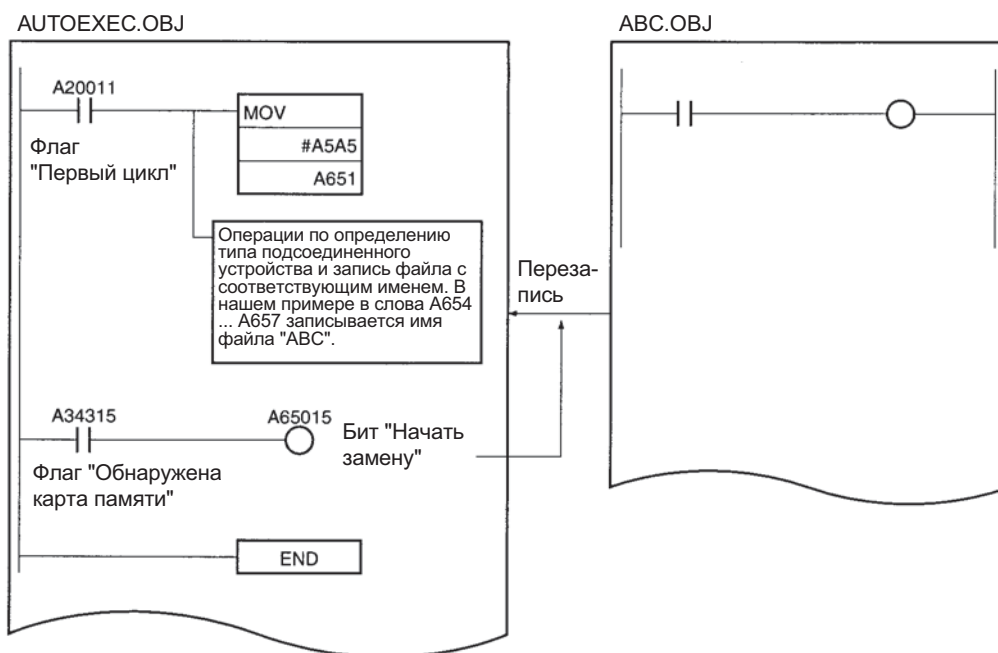
Операции по защите данных во время замены программы  
(циклическая задача 31, статус "ожидание" при запуске)



**Пример программы 2**

Запишите на карту памяти файлы программ, предназначенные для нескольких устройств, и файл программы, предназначенный для автоматической загрузки при запуске (AUTOEXEC.OBJ или REPLACE.OBJ (см. примечание)). Когда включается питание ПЛК, файл, предназначенный для автоматической загрузки при запуске, считывается из карты памяти, а позже он будет заменен файлом программы, предназначенной для другого устройства.

**Примечание** REPLACE.OBJ поддерживается только модулями CPU серии CS/CJ версии 2.0 или выше.



**5-2-5 Автоматическая загрузка при запуске**

Функция автоматической загрузки при запуске состоит в том, что программа пользователя, параметры и содержимое памяти ввода/вывода считываются из карты памяти в модуль CPU, когда включается питание. В память модуля CPU автоматически могут быть загружены следующие файлы.

**Примечание** Эту функцию нельзя использовать для чтения памяти файлов EM.

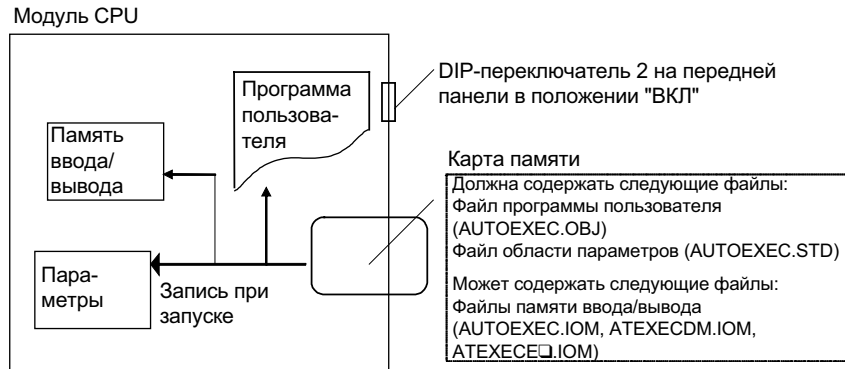
Имя файла программы зависит от того, должен ли также загружаться файл области параметров.

**Загрузка файла области параметров** Используйте следующие имена файлов.

Файл программы: AUTOEXEC.OBJ

Файл области параметров: AUTOEXEC.STD

Файлы данных: AUTOEXEC.IOM, ATEXECDM.IOM, ATEXECE□.IOM



Файл	Имя файла	При запуске	Требование для автоматической загрузки
Файл программы	AUTOEXEC.IOM	Содержимое этого файла автоматически загружается и записывается поверх всей программы пользователя, включая атрибуты задач модуля CPU.	Должен находиться в карте памяти.
Файл области параметров	AUTOEXEC.STD	Содержимое этого файла автоматически загружается и записывается поверх всех первоначальных параметров модуля CPU.	Должен находиться в карте памяти.
Файл данных	AUTOEXEC.IOM	Слова DM, отведенные для специальных модулей ввода/вывода, модулей шины CPU и встраиваемых плат (только серия CS). Содержимое этого файла автоматически загружается в область DM, начиная с D20000, когда включается питание (см. примечание 1).	Не обязательно должен находиться в карте памяти.
	AUTOEXEC.IOM	Слова DM общего назначения Содержимое этого файла автоматически загружается в область DM, начиная с D00000, когда включается питание (не поддерживается модулями CPU серии CS1 до версии EV1) (см. примечание 1).	
	AUTOEXECE□.IOM	Слова DM общего назначения Содержимое этого файла автоматически загружается в область EM, начиная с E□_00000, когда включается питание (не поддерживается модулями CPU серии CS1 до версии EV1).	

- Примечание**
1. Если данные, содержащиеся в файлах AUTOEXEC.IOM и ATEXECDM.IOM, перекрываются, данные файла ATEXECDM.IOM будут записаны поверх любых данных файла AUTOEXEC.IOM, с которыми они перекрываются, поскольку файл ATEXECDM.IOM записывается позже.
  2. Файл программы (AUTOEXEC.OBJ) и файл параметров (AUTOEXEC.STD) должны присутствовать в карте памяти. Если эти файлы отсутствуют, автоматической загрузки не произойдет, возникнет ошибка памяти и включится бит A40115 (флаг "Ошибка памяти: фатальная ошибка") (наличие файла памяти ввода/вывода (AUTOEXEC.IOM) не обязательно).
  3. С помощью средства программирования (консоли программирования или CX-Programmer) файлы AUTOEXEC.IOM, ATEXECDM.IOM и ATEXECE□.IOM можно создавать с другими начальными адресами помимо D20000, D00000 и E□\_00000. В любом случае данные будут записываться, начиная с надлежащего начального адреса, однако не следует указывать другие начальные адреса.

4. Если DIP-ключ 7 переведен во "ВКЛ", а ключ 8 - в "ВЫКЛ" (выбрана функция простого резервного сохранения), приоритетом будет обладать функция простого резервного сохранения, даже если ключ 2 также будет находиться в положении "ВКЛ". В этом случае файлы BACKUP□□ будут загружены в модуль CPU, а файлы, предназначенные для загрузки при включении, загружены не будут (не поддерживается модулями CPU серии CS1 до версии EV1).
5. Функцию автоматической загрузки при запуске можно применять совместно с функцией замены программы. Из программы, которая автоматически загружается при запуске, можно включить бит "Начать замену" (A65015), чтобы заменить эту программу другой программой.
6. Функцию автоматической загрузки при запуске можно использовать вместе с битами дополнительной области, предназначенными для полной замены программы. В этом случае программа, которая загружается автоматически в модуль CPU при запуске, может содержать команды управления битами дополнительной области, служащими для замены программы на другую.

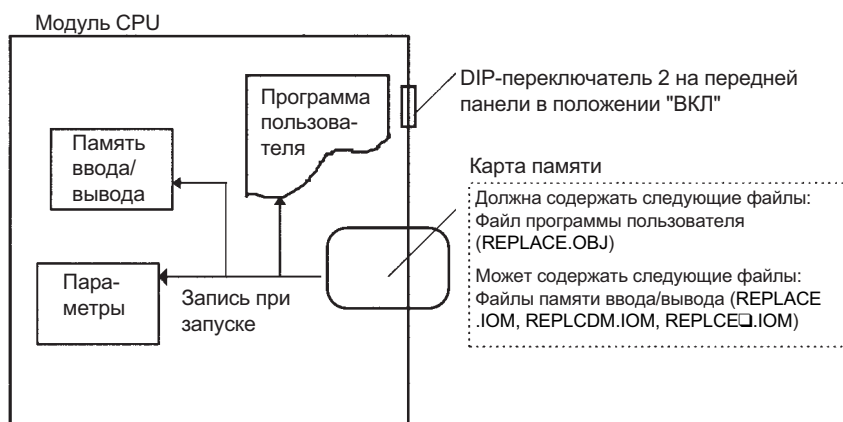
**Загрузка при отсутствии файла области параметров (только модуль CPU серии CS/CJ версии 2.0 или выше)**

Используйте следующие имена файлов.

Файл программы: REPLACE.OBJ

Файл области параметров: не требуются и не загружаются, независимо от имени файла.

Файл данных: REPLACE.IOM, REPLCDM.IOM, REPLCE□.IOM



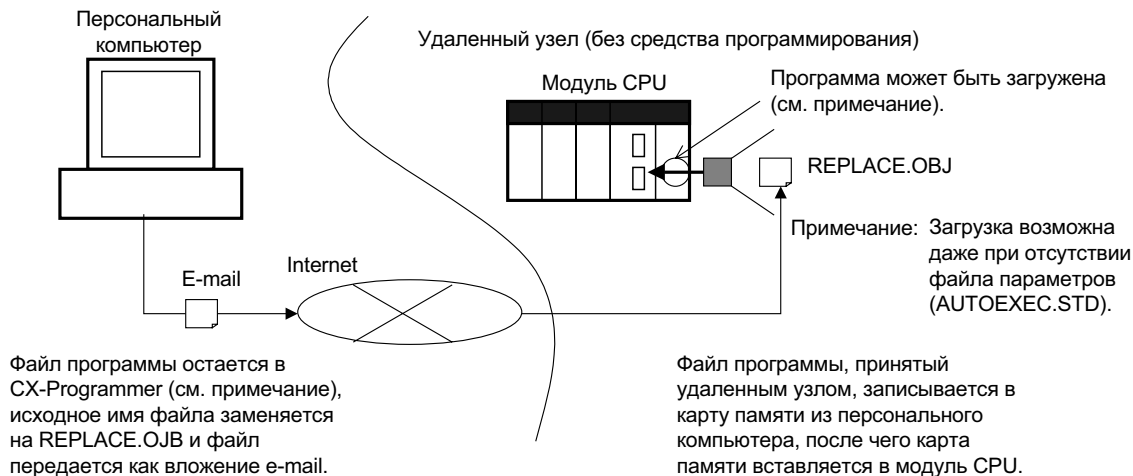
Файл	Имя файла	При запуске	Требование для автоматической загрузки
Файл программы	REPLACE.OBJ Примечание: только модуль CPU серии CS/CJ версии 2.0 или выше	Содержимое этого файла автоматически загружается и записывается поверх всей программы пользователя, включая атрибуты задач модуля CPU.	Должен находиться в карте памяти.
Файл области параметров		Не загружается, независимо от имени файла.	Не требуется.
Файл данных	REPLACE.IOM Примечание: только модуль CPU серии CS/CJ версии 2.0 или выше	Слова DM, отведенные для специальных модулей ввода/вывода, модулей шины CPU и встраиваемых плат (только серия CS). Содержимое этого файла автоматически загружается в область DM, начиная с D20000, когда включается питание.	Не обязательно должны находиться в карте памяти.
	REPLCDM.IOM Примечание: только модуль CPU серии CS/CJ версии 2.0 или выше	Слова DM общего назначения Содержимое этого файла автоматически загружается в область DM, начиная с D00000, когда включается питание.	
	REPLCE□.IOM Примечание: только модуль CPU серии CS/CJ версии 2.0 или выше	Слова DM общего назначения Содержимое этого файла автоматически загружается в область EM, начиная с E□_00000, когда включается питание. □ - соответствует номеру банка.	

- Примечание**
1. Если файл программы имеет название REPLACE.OBJ (модуль CPU серии CS/CJ версии 2.0 или выше), в этом случае файл области параметров передан не будет, даже если он имеется в карте памяти, независимо от имени файла области параметров.
  2. При создании файла REPLACE.IOM, REPLCDM.IOM или REPLACE□.IOM с помощью средства программирования (консоли программирования или CX-Programmer) всегда следует указывать надлежащий начальный адрес (D20000, D00000 или E□\_00000). Содержание файла всегда передается, начиная с надлежащего начального адреса (D20000, D00000 или E□\_00000), даже если указано другое начальное слово, что может привести к повреждению данных в этой части области DM или банка EM.
  3. Если ключ 7 DIP-переключателя находится в положении ВКЛ, а ключ 8 - в положении ВЫКЛ (выбрано использование функции простого резервного сохранения), приоритет будет отдан функции простого резервного сохранения, даже если ключ 2 также находится в положении ВКЛ. В этом случае в модуль CPU будут переданы файлы BACKUP□□, а файлы, предназначенные для автоматической загрузки программы при запуске, загружены не будут.

### Пример применения

### Автоматическая загрузка файлов без файла области параметров

Файл программы (.OBJ) можно создать, не устанавливая связь с ПЛК, после чего загрузить его по сети без файла области параметров (.STD). Файл программы можно записать в карту памяти, находящуюся на удаленном узле, не используя средство программирования. После этого карту памяти можно использовать для автоматической загрузки программы в модуль CPU при запуске.



**Возможные комбинации  
загружаемых файлов**

В следующих таблицах показано, при каких комбинациях файлов в карте памяти может быть выполнена автоматическая загрузка в модуль CPU при запуске.

**■ Файл программы: AUTOEXEC.OBJ**

Файл программы	Файл области параметров	Файлы данных	Загружается/ не загружается
AUTOEXEC.OBJ	AUTOEXEC.STD	Один или несколько следующих файлов: AUTOEXEC.IOM, ATEXECDM.IOM, ATEXECE□.IOM	Загружаются.
		Нет	
	Нет	Один или несколько следующих файлов: AUTOEXEC.IOM, ATEXECDM.IOM, ATEXECE□.IOM	Не загружаются.
		Нет	

**■ Файл программы: REPLACE.OBJ**

Файл программы	Файл области параметров	Файлы данных	Загружается/ не загружается
REPLACE.OBJ	Имеется	Один или несколько следующих файлов: REPLACE.IOM, REPLCDM.IOM, REPLCE□.IOM	Загружаются, кроме файла области параметров.
		Нет	
	Нет	Один или несколько следующих файлов: REPLACE.IOM, REPLCDM.IOM, REPLCE□.IOM	Загружаются.
		Нет	

**■ Файл программы отсутствует**

Файл программы	Файл области параметров	Файлы данных	Загружается/ не загружается
Нет	AUTOEXEC.STD	Один или несколько следующих файлов: AUTOEXEC.IOM, ATEXECDM.IOM, ATEXECE□.IOM	Не загружаются.
		Один или несколько следующих файлов: REPLACE.IOM, REPLCDM.IOM, REPLCE□.IOM	
		Нет	
	Нет	Один или несколько следующих файлов: AUTOEXEC.IOM, ATEXECDM.IOM, ATEXECE□.IOM	
Один или несколько следующих файлов: REPLACE.IOM, REPLCDM.IOM, REPLCE□.IOM			

**■ Имеются оба файла AUTOEXEC и REPLACE****Несколько файлов программы**

Файл программы	Файл области параметров	Файлы данных	Загружается/ не загружается
AUTOEXEC.OBJ	REPLACE.OBJ	Один или несколько следующих файлов: AUTOEXEC.IOM, ATEXECDM.IOM, ATEXECE□.IOM	Не загружаются.
		Один или несколько следующих файлов: REPLACE.IOM, REPLCDM.IOM, REPLCE□.IOM	
		Нет	
	Нет	Один или несколько следующих файлов: AUTOEXEC.IOM, ATEXECDM.IOM, ATEXECE□.IOM	
Один или несколько следующих файлов: REPLACE.IOM, REPLCDM.IOM, REPLCE□.IOM			

## Файлы данных нескольких типов

Файл программы	Файл области параметров	Файлы данных		Загружается/ не загружается
AUTOEXEC.OBJ	AUTOEXEC.STD	Один или несколько следующих файлов: AUTOEXEC.IOM, ATEXECMD.IOM, ATEXECE□.IOM	Один или несколько следующих файлов: REPLACE.IOM, REPLCDM.IOM, REPLCE□.IOM	Загружаются следующие файлы данных: AUTOEXEC.IOM, ATEXECMD.IOM, ATEXECE□.IOM
	Нет			Не загружаются.
REPLACE.OBJ	Игнорируется			Загружаются следующие файлы данных: REPLACE.IOM, REPLCDM.IOM, REPLCE□.IOM

## Последовательность действий

- 1,2,3...
1. Выключите питание ПЛК.
  2. Ключ 2 DIP-переключателя на лицевой панели модуля CPU переведите в положение ВКЛ. Ключи 7 и 8 должны быть в положении ВЫКЛ.  
Примечание Функция резервного сохранения обладает приоритетом над автоматической загрузкой при запуске, поэтому ключи 7 и 8 обязательно должны быть в положении ВЫКЛ.
  3. Подготовьте карту памяти следующим образом:
    - a) Загрузка вместе с файлом области параметров  
Вставьте карту памяти, содержащую файл программы пользователя (AUTOEXEC.OBJ), файл области параметров (AUTOEXEC.STD) и/или файлы памяти ввода/вывода (AUTOEXEC.IOM, ATEXECMD.IOM и ATEXECE□.IOM), созданные с помощью CX-Programmer (Файл программы и файл области параметров должны находиться в карте памяти. Файлы памяти ввода/вывода не являются обязательными).
    - b) Загрузка без файла области параметров  
Вставьте карту памяти, содержащую файл программы пользователя (REPLACE.OBJ) и/или файлы памяти ввода/вывода (REPLACE.IOM, REPLCDM.IOM и REPLCE□.IOM), созданные с помощью CX-Programmer (Файл программы должен находиться в карте памяти. Файлы памяти ввода/вывода не являются обязательными).
  4. Включите напряжение питания ПЛК.

**Примечание Сбой автоматической загрузки при запуске**

В случае сбоя автоматической загрузки при запуске возникает ошибка памяти, включается бит A40115, модуль CPU прекращает работу. Если происходит ошибка, следует выключить питание, чтобы сбросить ошибку (ошибка не будет сброшена, если питание не будет выключено).

**DIP-переключатель на передней панели модуля CPU**

Ключ(-и)	Имя	Положение
2	Автоматическая загрузка при запуске	ВКЛ: Выполнять автоматическую загрузку при запуске ВЫКЛ: Не выполнять автоматическую загрузку при запуске
7 и 8	Простое резервное сохранение	Переведите оба переключателя в положение ВЫКЛ

## Сопутствующие дополнительные биты/слова

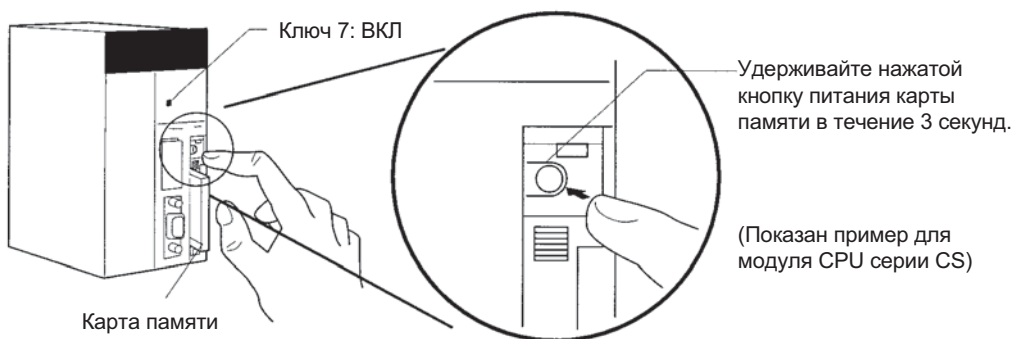
Название	Адрес	Назначение
Флаг "Ошибка памяти" (фатальная ошибка)	A40115	ВКЛ, если произошла ошибка памяти или произошла ошибка при автоматической загрузке из карты памяти после включения питания (автоматическая загрузка при запуске). Модуль CPU прекратит работу, а индикатор ERR/ALM на передней панели модуля CPU будет светиться. <b>Примечание:</b> бит A40309 включится, если во время автоматической загрузки при запуске произойдет ошибка (в этом случае ошибку сбросить нельзя).
Флаг "Ошибка загрузки из карты памяти при запуске"	A40309	ВКЛ, если была выбрана автоматическая загрузка при запуске и во время автоматической загрузки произошла ошибка (DIP-ключ 2 в положении ВКЛ). Ошибка возникнет, если произойдет ошибка загрузки, будет отсутствовать указанный файл или не будет установлена карта памяти. <b>Примечание:</b> Ошибку можно сбросить, выключив питание (ошибку нельзя сбросить, если питание не будет выключено).

## 5-2-6 Функция простого резервного сохранения данных

Эта функция не поддерживается модулями CPU серии CS1 до версии EV1.

**Резервное сохранение данных из модуля CPU в карту памяти**

Для резервного сохранения данных следует перевести ключ 7 DIP-переключателя модуля CPU в положение ВКЛ, после чего нажать и удерживать нажатой кнопку напряжения питания карты памяти в течение 3 секунд. В результате функция резервного сохранения автоматически создаст резервные файлы и запишет их в карту памяти. Резервные файлы содержат программу, данные области параметров и данные памяти ввода/вывода. Эту операцию можно выполнить в любом режиме работы.

**Восстановление данных из карты памяти в модуль CPU**

Чтобы восстановить сохраненные копии файлов и записать их в модуль CPU, следует перевести ключ 7 в положение ВКЛ и выключить и вновь включить питание ПЛК. Резервные файлы, содержащие программу, данные области параметров и данные памяти ввода/вывода, будут прочитаны из карты памяти и записаны в модуль CPU.

**Примечание:**

1. Функция резервного сохранения обладает приоритетом над функцией автоматической загрузки при запуске, поэтому после включения ПЛК в модуль CPU будут записаны резервные копии файлов, даже если ключ 2 DIP-переключателя находится в положении ВКЛ.
2. Если ключ 1 DIP-переключателя находится в положении ВКЛ (защита памяти программы от записи), данные не будут прочитаны из карты памяти в модуль CPU.
3. После того, как резервные файлы будут записаны в карту памяти с помощью функции резервного сохранения, состояние памяти ввода/вывода и принудительно установленные/сброшенные биты будут обнулены, если не были сделаны соответствующие настройки в дополнительной области и в настройках ПЛК.

Если включен бит "Удержание IOM" (A50012) и в настройках ПЛК выбрано сохранение состояния бита "Удержание IOM" при запуске, когда записываются резервные файлы, в этом случае при чтении данных из карты памяти состояние памяти ввода/вывода будет сохранено.

Если включен бит "Удержание принудительных состояний" (A50013) и в настройках ПЛК выбрано сохранение состояния бита "Удержание принудительных состояний" при запуске, когда записываются резервные файлы, в этом случае при чтении данных из карты памяти будет сохранено состояние принудительно установленных и сброшенных битов.

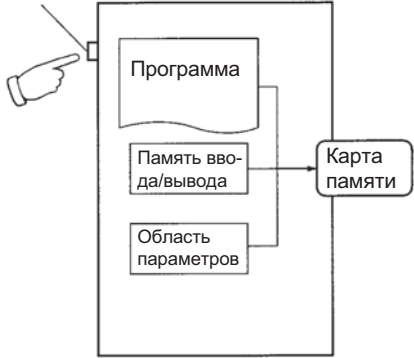
4. По завершению работы функции простого резервного сохранения модуль CPU серии CS1-H, CJ1-H, CJ1M или CS1D останется в режиме PROGRAM и не сможет перейти в режим MONITOR или RUN, пока не будет выключено и вновь включено напряжение питания. По завершению резервного сохранения следует выключить питание модуля CPU, изменить положение DIP-ключа 7, после чего вновь включить напряжение питания.
5. Процедуры резервного сохранения файлов могут занять от нескольких секунд до нескольких минут. Сведения о значениях времени выполнения операции приведены на стр. 245.


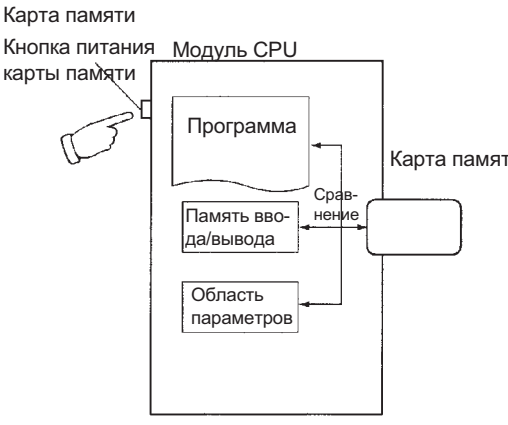
### **Сравнение данных в карте памяти и модуля CPU**

Чтобы сравнить резервные копии файлов, находящиеся в карте памяти, с данными модуля CPU, следует перевести DIP-ключ 7 модуля CPU в положение ВЫКЛ, нажать и удерживать нажатой кнопку напряжения питания карты памяти в течение 3-х секунд. В результате программа, данные области параметров и данные памяти ввода/вывода, находящиеся в карте памяти, будут сравнены с соответствующими данными модуля CPU. Эту функцию можно выполнить в любом режиме работы.



В следующей таблице приведен краткий обзор операций, связанных с функцией простого резервного сохранения.

Простое резервное сохранение	Положение ключа DIP-ключ 7	Процедура
<p>Резервное копирование данных из модуля CPU в карту памяти</p> <p>Резервное сохранение данных в карту памяти</p> <p>Карта памяти</p> <p>Кнопка включения питания карты памяти модуля CPU</p>  <p>DIP-ключ 7: ВКЛ</p>	<p>ВКЛ</p>	<p>Удерживайте нажатой кнопку напряжения питания карты памяти в течение 3-х секунд.</p>

Простое резервное сохранение	Положение ключа	Процедура
	DIP-ключ 7	
<p>Восстановление данных из карты памяти в модуль CPU</p> <p>Восстановление данных из карты памяти</p>  <p>DIP-ключ 7: ВКЛ</p>	ВКЛ	Выключите и вновь включите питание ПЛК. (см. прим. 1)
<p>Сравнение данных модуля CPU и карты памяти</p> <p>Сравнение данных с картой памяти</p>  <p>DIP-ключ 7: ВЫКЛ</p>	ВЫКЛ	Нажимайте и удерживайте нажатой кнопку питания карты памяти в течение 3-х секунд.

- Примечание**
1. Сведения о результатах операций чтения, записи и сравнения можно найти в разделе *Синхронизация операций резервного сохранения с помощью индикаторов* на стр. 234.
  2. Ориентировочные значения времени, необходимого для выполнения операций резервного сохранения, выполняемых с картой памяти, можно найти в разделе 5-3-2 *Рабочие процедуры*.

## Резервные копии файлов

## Файлы данных

Имя и расширение файла	Область данных и диапазон сохраняемых слов (адресов)		Копирование из памяти ввода/вывода в карту памяти (создание файлов)	Восстановление из карты памяти в память ввода/вывода	Сравнение карты памяти с памятью ввода/вывода		Файлы, необходимые при восстановлении данных
					CS1/CJ1	CS1-H/CJ1-H	
Модуль CPU	CS/CJ				CS1/CJ1	CS1-H/CJ1-H	
BACKUP.IOM	DM	D20000 ... D32767	Да	Да	Да	---	Должен быть в карте памяти
BACKUPIO.IOR	CIO	0000 ... 6143 Включая принужденные состояния битов	Да	---	Да	---	Должен быть в карте памяти
	WR	W000 ... W511 Включая принужденные состояния битов	Да	---	Да	---	
	HR	H000 ... H511	Да	Да	Да	---	
	AR	A000 ... A447	Да	---	---	---	
		A448 ... A959	Да	Да	Да	---	
	Таймер <sup>1</sup>	T0000 ... T4095	Да	Да			
Счетчик <sup>1</sup>	C0000 ... C4095	Да	Да	Да	---		
BACKUPDM.IOM	DM	D00000 ... D19999	Да	Да	Да	---	Должен быть в карте памяти
BACKUPE□.IOM <sup>2,3</sup>	EM	E□_00000 ... E□_32767	Да	Да	Да	---	Должен быть в карте памяти (должен совпадать с файлом в модуле CPU)

**Примечание** 1. Также сохраняются флаги завершения и текущие значения (PV).

2. □ соответствует номеру банка, который зависит от используемого модуля CPU.

Когда файлы BACKUPE□.IOM из карты памяти восстанавливаются и записываются в модуль CPU, чтение файлов происходит, начиная с банка 0, и заканчивая банком с максимальным номером, который имеется в модуле CPU. Файлы BACKUPE□.IOM с более высокими номерами не будут прочитаны, если количество резервных копий банков превышает количество банков, имеющихся в модуле CPU. И наоборот, любые банки EM с более высокими номерами в модуле CPU не будут заменены, если количество резервных копий банков меньше, чем количество банков в модуле CPU.

Если файл BACKUPE□.IOM отсутствует (например: 0, 1, 2, 4, 5, 6), будут прочитаны только файлы, расположенные до пропуска. В нашем примере данные будут записаны только в банки 0, 1 и 2.

3. Данные области EM сохраняются как двоичные данные. Банки EM, которые были преобразованы в память файлов, будут сохранены вместе с банками EM, которые преобразованы в память файлов не были.

Память файлов EM можно восстановить (записать) в область EM другого CPU только в том случае, если файлы BACKUPE□.IOM имеют последовательную нумерацию и количество резервных копий банков EM совпадает с количеством банков, имеющихся в модуле CPU. Если файлы BACKUPE□.IOM имеют пропуски, или количество банков EM не совпадает с количеством банков в модуле CPU, память файлов EM вернется в свое неформатированное состояние, файлы в памяти файлов будут некорректными (обычные банки области EM будут читаться без ошибок).

4. Как правило, содержимое области CIO, области WR, флагов завершения таймеров, текущие значения таймеров и состояния принудительно установленных/сброшенных битов будут обнулены при включении питания, после чего из карты памяти будет прочитан файл BACKUPIO.IOR.

Если включен бит "Удержание IOM" (A50012) и в настройках ПЛК выбрано сохранение состояния бита "Удержание IOM" при запуске, когда записываются резервные файлы, в этом случае данные памяти ввода/вывода будут сохранены, когда происходит чтение данных из карты памяти.

Если включен бит "Удержание принудительных состояний" (A50013) и в настройках ПЛК выбрано сохранение состояния бита "Удержание принудительных состояний" при запуске, когда записываются резервные копии файла, в этом случае при чтении данных из карты памяти будет сохранено состояние принудительно установленных и принудительно сброшенных битов.

#### Файлы программы

Имя и расширение файла	Содержание	Сохранение из памяти ввода/вывода в карту памяти (создание файлов)	Восстановление из карты памяти в память ввода/вывода	Сравнение карты памяти с памятью ввода/вывода	Файлы, необходимые при восстановлении данных
<b>Модуль CPU</b>		<b>CS/CJ</b>			
BACKUP.OBJ	Вся программа пользователя целиком	Да	Да	Да	Должен быть в карте памяти

#### Файлы параметров

Имя и расширение файла	Содержание	Сохранение из памяти ввода/вывода в карту памяти (создание файлов)	Восстановление из карты памяти в память ввода/вывода	Сравнение карты памяти с памятью ввода/вывода	Файлы, необходимые при восстановлении данных
<b>Модуль CPU</b>		<b>CS/CJ</b>			
BACKUP.STD	Настройки ПЛК Зарегистрированные таблицы ввода/вывода Таблицы маршрутизации Настройки модуля шины CPU и т.п.	Да	Да	Да	Должен быть в карте памяти

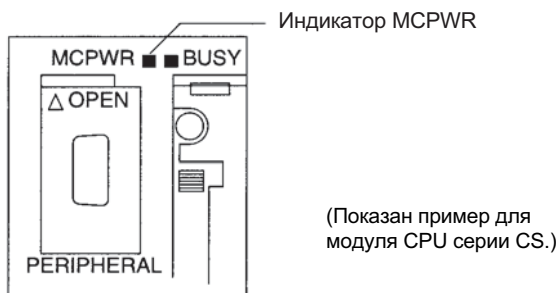
#### Резервные файлы модулей/плат (только для модуля CPU CS1-H, CS1D, CJ1-H, CJ1M или CS1D)

Имя и расширение файла	Содержание	Сохранение из памяти ввода/вывода в карту памяти (создание файлов)	Восстановление из карты памяти в память ввода/вывода	Сравнение карты памяти с памятью ввода/вывода	Файлы, необходимые при восстановлении данных
<b>Модуль CPU</b>		<b>Только для модуля CPU CS1-H, CS1D, CJ1-H, CJ1M или CS1D</b>			
BACKUP□□.PRM (где □□ - адрес модуля/платы, для которой выполняется)	Резервная копия данных из модуля или платы с указанным адресом (конкретное содержание зависит от модуля или платы).	Да	Да	Да	Должен быть в карте памяти (см. прим. 2)

- Примечание**
- Адреса модулей определяются следующим образом:  
Модули шины CPU: номер модуля + 10 Нех  
Специальные модули ввода/вывода: номер модуля + 20 Нех  
Встраиваемая плата: Е1 Нех.
  - Если при загрузке данных из карты памяти в память ввода/вывода этот файл отсутствует, ошибка в модуле CPU не произойдет, однако ошибка произойдет в модуле или плате, когда не будет произведено восстановление данных. Сведения об ошибках модулей или плат следует искать в соответствующих руководствах по эксплуатации.

**Сигнализация операций резервного сохранения с помощью индикаторов**

Индикатор состояния питания карты памяти (MCPWR) показывает, была ли завершена операция простого резервного сохранения без ошибок.



Операция резервного сохранения	Завершение без ошибок (см. прим. 1)	Произошла ошибка	
	Состояние MCPWR	Состояние MCPWR	Ошибка
Резервное сохранение данных из модуля CPU в карту памяти	Светится → Продолжает светиться при нажатой кнопке питания карты памяти → Мигает один раз → Светится при записи → Выключается после того, как данные записаны.	Светится → Продолжает светиться при нажатой кнопке питания карты памяти → Постоянно мигает → Светится, когда нажата кнопка питания карты памяти.	В случае следующих ошибок файлы созданы не будут: Недостаточно места в карте памяти (см. прим.2). Ошибка памяти в модуле CPU. Ошибка шины ввода/вывода (при записи данных в модуль или плату, только для модулей CPU CS1-H, CS1D или CJ1-H)
Восстановление данных из карты памяти в модуль CPU.	Светится после включения питания → Мигает один раз → Светится при чтении → Выключается после того, как данные прочитаны.	Светится после включения питания → Мигает пять раз → Выключается.	При наличии следующих ошибок данные прочитаны не будут: Программа в карте памяти превышает емкость памяти модуля CPU Требуемые резервные копии файлов отсутствуют в карте памяти. Программа не может быть записана из-за установленной защиты записи (DIP-ключ 1 в положении ВКЛ).
		Светится после включения питания → Мигает один раз → Светится при чтении → Мигает три раза → Выключается после того, как данные записаны.	Предостережение: при следующей ошибке данные будут прочитаны. Файл EM и банки EM модуля CPU не соответствуют друг другу (есть пропуски в нумерации банков или несоответствие макс. кол-ва банков).

Операция резервного сохранения	Завершение без ошибок (см. прим. 1)	Произошла ошибка	
	Состояние MCPWR	Состояние MCPWR	Ошибка
Сравнение данных модуля CPU и карты памяти	Светится → Продолжает светиться при нажатой кнопке питания карты памяти → Мигает один раз → Светится во время сравнения → Выключается после того, как данные сравнены.	Светится → Продолжает светиться при нажатой кнопке питания карты памяти → Постоянно мигает → Светится, когда нажата кнопка питания карты памяти.	Могут произойти следующие ошибки сравнения (см. прим. 3): Данные в карте памяти и в модуле CPU не совпадают. Требуемые резервные файлы отсутствуют в карте памяти. Файлы EM и банки EM модуля CPU не совпадают (непоследовательная нумерация банков, либо не совпадает макс. номер банка). Ошибка памяти модуля CPU Ошибка шины ввода/вывода (при сравнении данных в модуле или плате, только для модулей CPU CS1-H, CS1D или CJ1-H).
Общая индикация для всех трех операций резервного сохранения.	---	Чтение: Мигает пять раз → Выключается. Запись или сравнение: Продолжает мигать → Светится при нажатой кнопке питания карты памяти.	Ошибка доступа к карте памяти (ошибка формата или ошибка чтения/записи).

- Примечание**
1. Если операция резервного сохранения завершилась без ошибок, питание карты памяти будет отключено, когда погаснет индикатор MCPWR. Если карта памяти должна быть использована вновь, следует нажать кнопку питания карты памяти, чтобы подать на карту памяти питание, и выполнить требуемую операцию.
  2. При записи данных в карту памяти при выполнении операции простого резервного сохранения для модуля CPU CS1-H, CS1D, CJ1-H, CJ1M или CS1D слово A397 (наличие места для записи данных при простом резервном сохранении) позволяет обнаруживать ошибки отсутствия места в карте памяти. Если в A397 содержится какое-либо значение, кроме 0000 Hex, после того, как операция записи завершится, это значение будет соответствовать объему памяти (кбайт), который должен быть свободным в карте памяти.
  3. В случае модулей CPU CS1-H, CS1D, CJ1-H, CJ1M или CS1D также происходит сравнение резервных файлов для модулей и плат.

**Сопутствующие дополнительные биты/слова**

Название	Адрес	Описание
Флаг "Операция над памятью файлов"	A34313 ...	ВКЛ, когда выполняется любая из следующих операций. ВЫКЛ по завершению выполнения. <ul style="list-style-type: none"> <li>Обнаружена карта памяти</li> <li>Для локального модуля CPU выполнена команда CMND</li> <li>Команды FREAD/FWRIT</li> <li>Замена программы с помощью специальных битов управления</li> <li>Операция простого резервного сохранения</li> </ul> Когда этот флаг включен, запись данных или сравнение с содержимым карты памяти невозможно.
Первый банк в памяти файлов EM		Когда модуль CPU приступает к чтению данных из карты памяти, он опирается на это значение. Если максимальный номер банка EM для файла BACKUPЕ.ИОМ (максимальный последовательный номер банка, начиная с 0) совпадает с максимальным номером банка модуля CPU, будет выполнено форматирование области EM в соответствии со значением этого слова. Если максимальные номера банков EM не совпадают, область EM перейдет в неформатированное состояние.
Флаги "Использование команд сетевых коммуникаций разрешено" (только для модулей CS1-H, CJ1-H, CJ1M или CS1D) (см. примечание)	A20200 ... A20207	<ul style="list-style-type: none"> <li>Сбрасываются, когда начинается запись или сравнение с данными карты памяти.</li> <li>Включается, когда запись или сравнение с данными карты памяти завершаются.</li> </ul> Когда начинается операция записи в карту памяти или сравнение с данными карты памяти, операции записи или сравнения с данными модуля или платы не могут быть выполнены, если все флаги разрешения применения команд сетевых коммуникаций выключены. При попытке выполнить эти операции произойдет ошибка.
Коды завершения сетевых коммуникаций (только для модулей CS1-H, CJ1-H, CJ1M или CS1D) (см. примечание)	A203 ... A210	Содержат результаты обмена данными с модулем или платой, когда выполняются операции записи или сравнения данных для карты памяти.
Флаги "Ошибка сетевых коммуникаций" (только для модулей CS1-H, CJ1-H, CJ1M или CS1D) (см. примечание)	A21900 ... A21907	<ul style="list-style-type: none"> <li>Включается в случае возникновения ошибки связи с модулем или платой, когда выполняются операции записи или сравнения данных для карты памяти.</li> <li>Остаются выключенными (или выключаются), если ошибки связи с модулем или платой отсутствуют, когда производится запись или сравнение данных для карты памяти.</li> </ul>
Объем памяти для записи данных при простом резервном сохранении (только для модулей CS1-H, CJ1-H, CJ1M или CS1D)	A397	Содержат значение необходимого объема свободного места в карте памяти в случае сбойного завершения операции простого резервного сохранения, указывая на возникновение ошибки записи из-за отсутствия свободного места.  0001 ... FFFF Hex: ошибка записи (содержит требуемый объем свободного места в карте памяти в диапазоне 1 ... 65535 кбайт) (Сбрасывается в 0000 Hex в случае успешной записи).  0000 Hex: запись завершена без ошибок.

**Примечание** Эти флаги действительны для модулей CPU CS1-H, CJ1-H, CJ1M или CS1D, поскольку модуль CPU будет автоматически использовать доступный коммуникационный порт, когда происходит запись или сравнение данных с картой памяти.

**Резервное сохранение данных для платы и модуля**

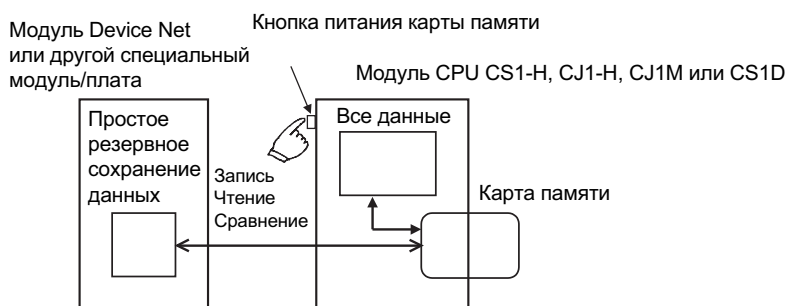
Эта функция поддерживается только модулями CPU CS1-H, CJ1-H, CJ1M или CS1D.

**Введение**

Функция простого резервного сохранения позволяет создать резервные копии для следующих данных модулей CPU CS1 и CJ1: программа пользователя, область параметров, полностью вся память ввода/вывода. Кроме того, для модулей CPU CS1-H, CJ1-H, CJ1M или CS1D также сохраняются следующие данные: данные специальных модулей и плат, установленных в ПЛК.

**Общие сведения**

Когда для модуля CPU CS1-H, CJ1-H, CJ1M или CS1D выполняется операция простого резервного сохранения данных, в карту памяти записывается резервный файл модуля/платы, который содержит данные этих модулей и плат. Данные сохраняются отдельно для каждого модуля и платы.



**Применение**

Эту функцию можно использовать для резервного сохранения данных для всего ПЛК, включая его модуль CPU, модули Device Net, модули/платы последовательного интерфейса и т.п. Ее также можно использовать для замены модуля.

**Резервные копии файлов для модуля/платы**

Для каждого модуля и платы данные сохраняются в карту памяти с использованием следующих имен файлов: BACKUP□□.PRM. Здесь "□□" - адрес модуля или платы в шестнадцатеричном формате.

**Примечание**

Адреса модулей определяются следующим образом:

Модули шины CPU: номер модуля + 10 hex

Специальные модули ввода/вывода: номер модуля + 20 hex

Встраиваемая плата: E1 Hex

Эти файлы также используются в случае чтения данных из карты памяти или сравнения данных с картой памяти.

**Поддерживаемые модули и платы**

Чтобы для модуля или платы можно было выполнить сохранение данных, функция резервного сохранения должна поддерживаться этим модулем/платой. Сведения о поддержке резервного сохранения данных следует искать в соответствующем руководстве по эксплуатации.

Модуль/плата	Номера моделей	Сохраняемые данные в случае использования модуля CPU CS1-H/CJ1-H	Объем, используемый в карте памяти для резервного сохранения данных
Модули Device Net	CJ1W-DRM21-V1 CJ1W-DRM21	Параметры устройства (все данные в EEPROM модуля)  (Хотя это те же самые данные, которые сохраняются с помощью функции резервного сохранения данных в карту памяти, поддерживаемой в программе конфигурирования модуля или Device Net (Вер. 2.0), эти файлы не совместимы).	7 кбайт
Модули последовательного	CS1W-SCU21-V1 CJ1W-SCU41	Данные Protocol macro  (Включая стандартные системные протоколы и протоколы пользователя, содержащиеся в флеш-памяти модуля или платы).	129 кбайт
Платы последовательного интерфейса	CS1W-SCB21-V1 CS1W-SCB41-V1		129 кбайт



Модуль/плата	Номера моделей	Сохраняемые данные в случае использования модуля CPU CS1-H/CJ1-H	Объем, используемый в карте памяти для резервного сохранения данных
Конфигурируемые модули счетчиков	CS1W-HIO01-V1 CS1W-HCP22-V1 CS1W-HCA22-V1 CS1W-HCA12-V1	<ul style="list-style-type: none"> <li>Программа пользователя</li> <li>Данные области DM общего назначения, предназначенные только для чтения</li> <li>Область настройки функций модуля</li> <li>Сведения о командах расширения</li> <li>РКП-программы</li> </ul>	64 кбайт
Модули управления движением	CS1W-MCH71	<ul style="list-style-type: none"> <li>Данные позиционирования</li> <li>Системные параметры</li> <li>Программы на языке G</li> </ul>	8192 кбайт
	CS1W-MC221-V1 CS1W-MC421-V1		142 кбайт
Модули позиционирования	CS1W-NC113/133/213/233/413/433 Вер. 2.0 или выше CJ1W-NC113/133/213/233/413/433 Вер. 2.0 или выше	<ul style="list-style-type: none"> <li>Параметры осей</li> <li>Параметры последовательности управления</li> <li>Параметры скорости</li> <li>Параметры времени ускорения/торможения</li> <li>Параметры сдвоенного счетчика</li> <li>Параметры зон</li> </ul>	7 кбайт

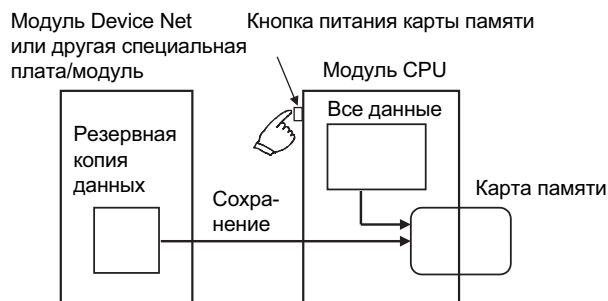
**Примечание** Перечисленные выше данные модулей и плат будут автоматически сохранены при работе функции простого резервного сохранения. Их включение или исключение не предусмотрено. Если используется консоль программирования, эти операции выполняются отдельно для области программы пользователя, области параметров и областей памяти ввода/вывода. Подробное описание приводится в руководстве *Programming Console Operation Manual (W314)*.

### Последовательность действий

Процедуры, связанные с резервным сохранением данных, не зависят, для какой именно платы или модуля они выполняются. Они всегда одни и те же (включая запись, чтение и сравнение).

#### ■ Резервное сохранение данных

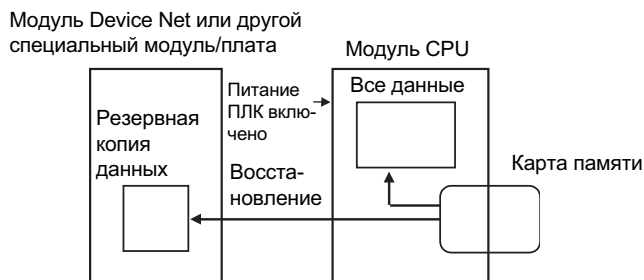
- 1,2,3...
1. Переведите ключ 7 DIP-переключателя модуля CPU в положение ВКЛ.
  2. Нажмите и удерживайте нажатой кнопку питания карты памяти в течение 3 секунд.
  3. Для модулей и плат будет создан файл с резервной копией данных, который будет записан в карту памяти вместе с остальными сохраняемыми данными.



После нажатия кнопки питания индикатор MCPWR мигнет один раз, будет светиться во время операции записи и отключится, если запись завершится без ошибок.

■ **Восстановление данных**

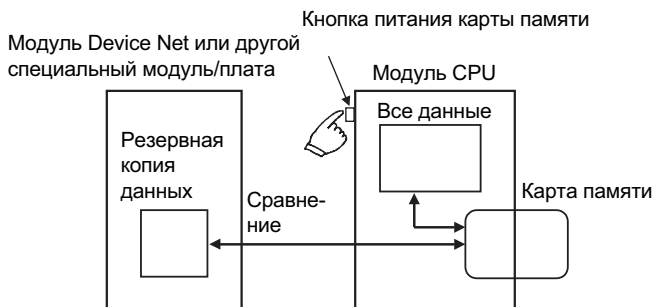
- 1,2,3... 1. Переведите ключ 7 DIP-переключателя модуля CPU в положение ВКЛ.  
 2. Включите питание ПЛК. Резервные копии файлов будут считаны и записаны в модули и платы.  
 Сохраненные данные модулей и плат будут прочитаны с карты памяти и восстановлены в модулях и платах.



После включения питания индикатор MCPWR мигнет один раз, будет светиться во время чтения и выключится, если чтение будет завершено без ошибок.

■ **Сравнение данных**

- 1,2,3... 1. Переведите ключ 7 DIP-переключателя модуля CPU в положение ВЫКЛ.  
 2. Нажмите и удерживайте нажатой кнопку питания карты памяти в течение 3 секунд.  
 Будет выполнено сравнение резервных копий данных карты памяти с данными в модулях и платах.



После нажатия кнопки питания индикатор MCPWR мигнет один раз, будет светиться во время операции сравнения и выключится, если сравнение будет завершено без ошибок и данные окажутся одинаковыми.

**Примечание** В случае применения модуля CS1W-SCU21-V1 длительность процедур резервного сохранения данных возрастает в соответствии с таблицами ниже.

**Возрастание времени для случая, когда размер файла настроек модуля шины CPU (BACKUP□□.PRM) в карте памяти составляет 60 кбайт**

Режим работы	Дополнительное время при записи в карту памяти	Дополнительное время при сравнении с картой памяти	Дополнительное время при чтении из карты памяти
PROGRAM	Приблиз. 25 с	Приблиз. 10 с	Приблиз. 4 с
RUN	Приблиз. 1 мин. 30 с	Приблиз. 30 с	Приблиз. 4 с

Возрастание времени для случая, когда размер файла настроек модуля шины CPU (BACKUP□□.PRM) в карте памяти составляет 128 кбайт

Режим работы	Дополнительное время при записи в карту памяти	Дополнительное время при сравнении с картой памяти	Дополнительное время при чтении из карты памяти
PROGRAM	Приблиз. 40 с	Приблиз. 14 с	Приблиз. 8 с
RUN	Приблиз. 2 мин. 30 с	Приблиз. 1 мин.	Приблиз. 8 с

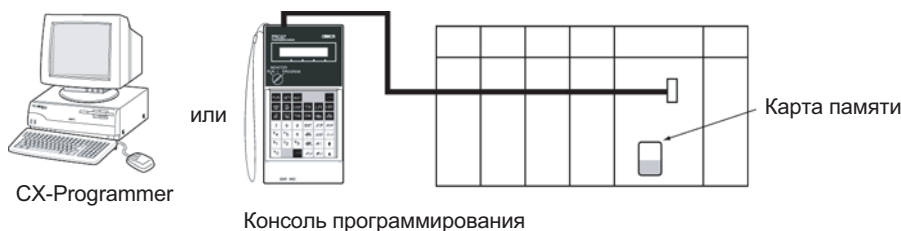
- Примечание**
- Прежде чем выполнять перечисленные выше операции, следует убедиться, что модули и платы работают надлежащим образом.
  - Прежде чем выполнять операцию простого резервного сохранения данных для определенных плат/модулей, следует проверить, работают ли модули CPU в режиме PROGRAM и не будет ли нарушена работа команд, использующих порты связи с соответствующими номерами. Когда происходит сохранение данных из определенных плат/модулей, производится поиск свободного порта связи, начиная с порта 0, и используется первый доступный порт. Если номер порта совпадает с портом, который используется командой сетевых коммуникаций, эта команда не сможет быть выполнена, пока не будет завершена операция резервного сохранения данных.

## 5-3 Применение памяти файлов

### 5-3-1 Инициализация носителя

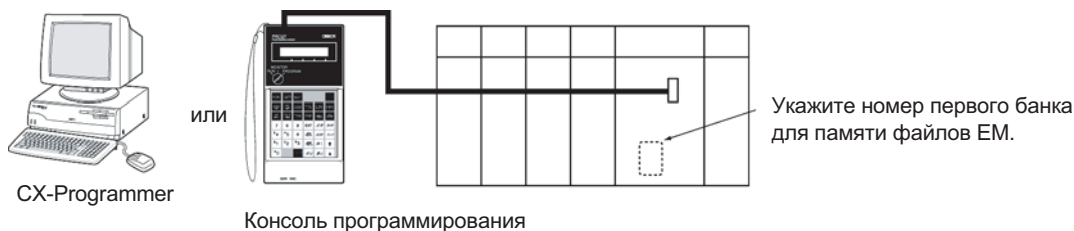
#### Карты памяти

- 1,2,3... 1. Для инициализации карт памяти используется средство программирования, например, консоль программирования.

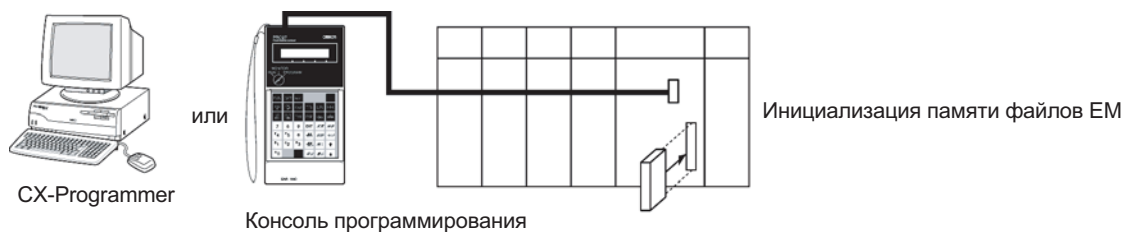


#### Память файлов EM

- 1,2,3... 1. С помощью любого средства программирования, например, консоли программирования настройте параметры памяти файлов EM в настройках ПЛК таким образом, чтобы разрешить применение памяти файлов EM, после чего выберите номер требуемого банка для памяти файлов EM в диапазоне 0 ... C Hex.



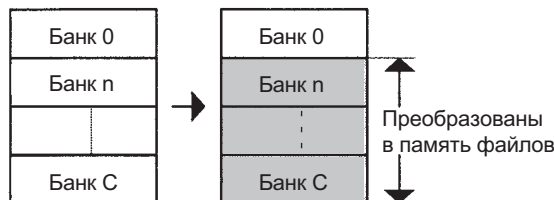
2. Чтобы инициализировать память файлов EM, используйте команду FINS или средство программирования (кроме консоли программирования).



**Инициализация отдельного банка памяти файлов EM**

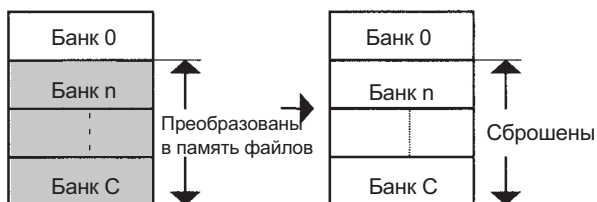
Указанный банк EM может быть преобразован из обычной памяти EM в память файлов.

**Примечание** Максимальное количество банков для модуля CPU серии CJ составляет 6.



1. Выберите n в настройках ПЛК.
2. С помощью средства программирования или команды FINS выполните форматирование, начиная с n.
3. Значение "n" хранится в A344.

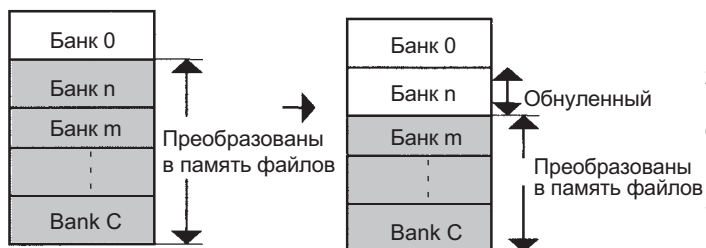
Память EM, используемая в качестве памяти файлов, вновь может быть преобразована в обычную память EM.



1. Отключите память файлов в настройках ПЛК.
2. Если для форматирования используется средство программирования или команда FINS, все ячейки памяти, начиная с n, будут обнулены (0000 Hex).
3. В A344 будет записано FFFF Hex, указывающее на отсутствие памяти файлов EM.

**Примечание:** 1. В этот момент будут удалены любые имеющиеся данные файла.  
2. Для модуля CPU серии CJ можно указать только банки 0 ... 6.

Номер первого банка для памяти файлов можно изменить.



1. Замените n на m в настройках ПЛК.
2. С помощью средства программирования или команды FINS преобразуйте банки, начиная с банка m, в память файлов.
3. Значение m будет храниться в A344.

**Примечание:** В банке n ... m-1 будет записано 0000 Hex.  
**Примечание:** 1. В этот момент будут удалены любые имеющиеся данные файла.  
2. Для модуля CPU серии CJ можно указать только банки 0 ... 6.

**Настройки ПЛК**

Адрес	Название	Описание	Начальное значение
136	Первый банк памяти файлов EM	0000 Hex: нет 0080 Hex: первый банк 0 008С Hex: первый банк С Область EM будет преобразована в память файлов, начиная с указанного номера банка. (Для модуля CPU серии CJ можно указать только банки 0 ... 6)	0000 Hex

**Сопутствующий дополнительный регистр**

Название	Адрес	Описание
Первый банк памяти файлов EM	A344	Сюда записывается номер банка, с которого в настоящий момент фактически начинается область памяти файлов EM. Файл EM будет преобразован в память файлов, начиная с банка с этим начальным номером, вплоть до последнего банка. Значение FFFF Hex будет указывать на отсутствие памяти файлов EM.

**Чтение/запись таблиц символов и комментариев с помощью CX-Programmer**

Для чтения/записи таблиц символов или комментариев, созданных с помощью CX-Programmer, из/в карту памяти или память файлов EM используйте следующую последовательность действий.

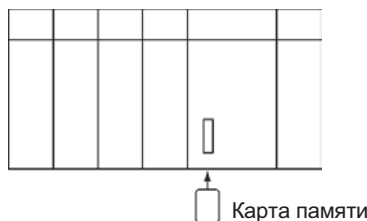
- 1,2,3...**
1. Вставьте отформатированную карту памяти в модуль CPU либо отформатируйте память файлов EM.
  2. Переведите CX-Programmer в online-режим.
  3. Выберите **Transfer** (Загрузить), а затем **To PLC** (В ПЛК) или **From PLC** (Из ПЛК) в меню PLC (ПЛК).
  4. Выберите загружаемые данные: **Symbols** (Символы) или **Comments** (Комментарии).

**5-3-2 Рабочие процедуры**

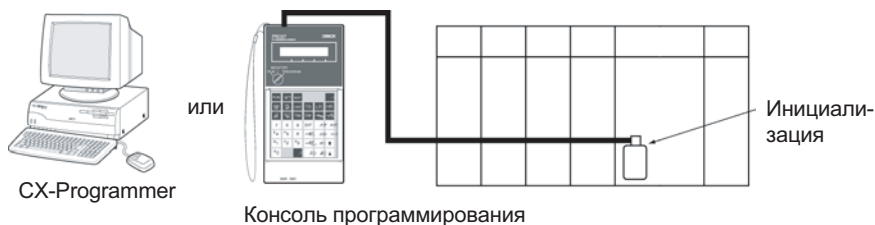
**Карты памяти**

**Использование средства программирования**

- 1,2,3...**
1. Вставьте карту памяти в модуль CPU.



2. Выполните инициализацию карты памяти с помощью средства программирования.

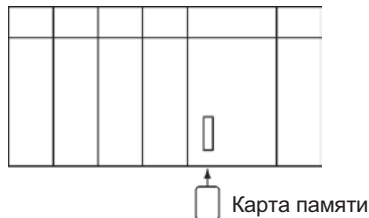


3. С помощью средства программирования выберите название данных в модуле CPU (программа пользователя, память ввода/вывода, область параметров), после чего сохраните данные в карту памяти (для чтения и записи файла карты памяти в модуле CPU используйте средство программирования).

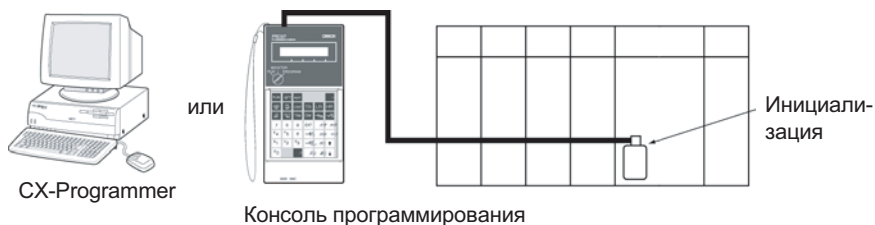
**Автоматическая загрузка файлов при запуске**

**Последовательность действий при загрузке файла области параметров**

- 1,2,3... 1. Вставьте инициализированную карту памяти в модуль CPU.

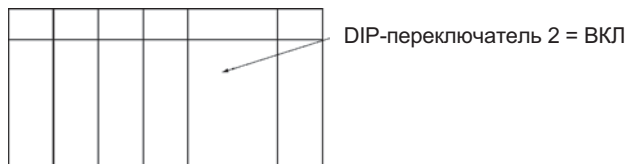


2. С помощью средства программирования запишите в карту памяти файлы, которые должны автоматически загружаться при запуске. В состав этих файлов входит файл программы (AUTOEXEC.OBJ), файл области параметров (AUTOEXEC.STD) и файлы памяти ввода/вывода (AUTOEXEC.IOM, ATEXECMD.IOM и/или ATEXECE□.IOM).



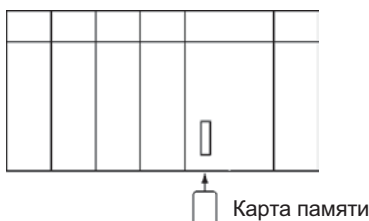
**Примечание** В карте памяти обязательно должны находиться файл программы пользователя и файл области параметров.

3. Выключите питание ПЛК.  
4. Переведите ключ 2 DIP-переключателя в положение ВКЛ (автоматическая загрузка при запуске).



**Примечание** Если DIP-ключ 7 находится в положение ВКЛ, а DIP-ключ 8 - ВЫКЛ, будет активизирована функция резервного сохранения, которая обладает приоритетом над автоматической загрузкой при запуске. Если требуется выполнить автоматическую загрузку при запуске, ключи 7 и 8 должны быть переведены в положение ВЫКЛ.

5. Вставьте карту памяти в модуль CPU.



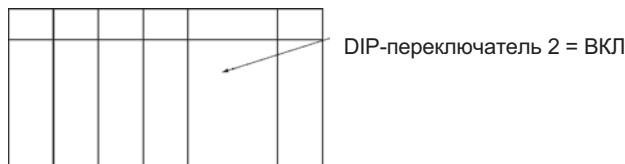
6. Подайте питание на ПЛК, чтобы выполнить чтение файлов.

**Последовательность действий для случая, когда не загружается файл области параметров**

- 1,2,3... 1. Вставьте инициализированную карту памяти в модуль CPU.  
2. С помощью средства программирования запишите в карту памяти файлы, которые должны автоматически загружаться при запуске. В состав этих файлов входит файл программы (REPLACE.OBJ) и файлы памяти ввода/вывода (REPLACE.IOM, REPLCDM.IOM и/или REPLCDE□.IOM).

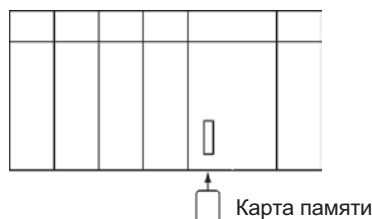
**Примечание** Область параметров не будет загружена, даже если она есть в карте памяти.

3. Выключите питание ПЛК.
4. Переведите ключ 2 DIP-переключателя в положение ВКЛ (автоматическая загрузка при запуске).



**Примечание** Если DIP-ключ 7 находится в положение ВКЛ, а DIP-ключ 8 - ВЫКЛ, будет активизирована функция резервного сохранения, которая обладает приоритетом над автоматической загрузкой при запуске. Если требуется выполнить автоматическую загрузку при запуске, ключи 7 и 8 должны быть переведены в положение ВЫКЛ.

5. Вставьте карту памяти в модуль CPU.



6. Подайте питание на ПЛК, чтобы выполнить чтение файла.

### Применение FREAD(700)/FWRIT(701)/CMND(490)

- 1,2,3... 1. Вставьте карту памяти в модуль CPU (уже инициализированную).
2. С помощью FWRIT(701) выберите имя файла в указанной области памяти ввода/вывода и сохраните файл в карту памяти.

**Примечание** Карту памяти, содержащую файлы данных в формате TXT или CSV, можно вставить в гнездо для ПЛК-карты персонального компьютера с помощью адаптера для карты памяти HMC-AP001. Это позволяет выполнить чтение файлов данных в программу обработки электронных таблиц с помощью стандартных функций Windows (не поддерживается модулями CPU серии CS1 до версии EV1).

3. С помощью FREAD(700) выполните чтение файлов из карты памяти в память ввода/вывода модуля CPU.

Операции над файлами карты памяти можно выполнять, выставя команды FINS локальному модулю CPU с помощью CMND(490) (не поддерживается модулями CPU серии CS1 до версии EV1).

### Замена программы во время работы

- 1,2,3... 1. Вставьте карту памяти в модуль CPU (уже инициализированную).
2. Запишите в слово A651 пароль программы (A5A5 Hex), а в слова A654 ... A657 запишите имя файла программы.
3. Переведите бит "Начать замену" (A65015) из состояния ВЫКЛ во ВКЛ.

### Функция простого резервного сохранения

Предусмотрено три операции, связанные с резервным сохранением данных: резервное сохранение данных в карту памяти, восстановление данных из карты памяти и сравнение данных с картой памяти.

#### Резервное сохранение данных из модуля CPU в карту памяти

- 1,2,3... 1. Вставьте карту памяти в модуль CPU (уже инициализированную).
2. Переведите ключ 7 DIP-переключателя модуля CPU в положение ВКЛ, а ключ 8 - в положение ВЫКЛ.
3. Нажмите и удерживайте нажатой кнопку питания карты памяти в течение 3-х секунд.
4. Индикатор MCPWR должен мигнуть 1 раз, а затем выключиться (другое поведение означает, что в процессе резервного сохранения данных произошла ошибка).

**Восстановление данных из карты памяти в модуль CPU**

- 1,2,3... 1. Вставьте карту памяти, содержащую резервные копии файлов, в модуль CPU.  
 2. Переведите ключ 7 DIP-переключателя модуля CPU в положение ВКЛ, а ключ 8 - в положение ВЫКЛ.  
 3. Данные будут восстановлены из резервных копий файлов после включения питания ПЛК.  
 4. Индикатор MCPWR должен мигнуть 1 раз, после чего выключиться (другое поведение означает, что в процессе восстановления данных произошла ошибка).

**Сравнение данных карты памяти и модуля CPU**

- 1,2,3... 1. Вставьте карту памяти, содержащую резервные копии файлов, в модуль CPU.  
 2. Переведите ключи 7 и 8 DIP-переключателя модуля CPU в положение ВЫКЛ.  
 3. Нажмите и удерживайте нажатой кнопку питания карты памяти в течение 3-х секунд.  
 4. Если индикатор MCPWR мигнет 1 раз, а затем выключится, значит, данные совпадают.

**Примечание** Если в процессе записи или сравнения данных произошла ошибка, индикатор MCPWR будет мигать. Мигание прекращается и индикатор MCPWR непрерывно светится, если нажимается кнопка питания карты памяти.

В следующей таблице приведены значения времени, необходимого для выполнения операций резервного сохранения, для следующего случая: объем программы 20000 шагов, длительность цикла 10 мс, режим RUN.

Режим	Резервное сохранение	Восстановление	Сравнение
PROGRAM	Приблиз. 50 с.	Приблиз. 30 с.	Приблиз. 7 с.
RUN	Приблиз. 5 мин.	Приблиз. 2 мин.	Приблиз. 7 с.

В следующей таблице приведены значения времени, необходимого для выполнения операций резервного сохранения, для следующего случая: объем программы 30000 шагов, длительность цикла 10 мс, режим RUN.

Режим	Резервное сохранение	Восстановление	Сравнение
PROGRAM	Приблиз. 50 с.	Приблиз. 30 с.	Приблиз. 7 с.
RUN	Приблиз. 5 мин. 30 с.	Приблиз. 2 мин. 40 с.	Приблиз. 7 с.

В следующей таблице приведены значения времени, необходимого для выполнения операций резервного сохранения, для следующего случая: объем программы 250000 шагов, длительность цикла 12 мс, режим RUN.

Режим	Резервное сохранение	Восстановление	Сравнение
PROGRAM	Приблиз. 1 мин. 30 с.	Приблиз. 1 мин 30 с.	Приблиз. 20 с.
RUN	Приблиз. 13 мин.	Приблиз. 7 мин. 30 с.	Приблиз. 20 с.

**Создание файлов таблиц переменных и комментариев**

Для создания файлов таблиц переменных или файлов комментариев в картах памяти или в файлах памяти EM используйте следующую последовательность действий в CX-Programmer.

- 1,2,3... 1. Вставьте отформатированную карту памяти в модуль CPU или отформатируйте память файлов EM.  
 2. Переведите CX-Programmer в online-режим.  
 3. Выберите **Transfer** (Загрузить), а затем **To PLC** (В ПЛК) или **From PLC** (Из ПЛК) в меню PLC (ПЛК).  
 4. Выберите загружаемые данные: **Symbols** (Символы) или **Comments** (Комментарии).

**Примечание** Если в область CPU вставлена карта памяти, данные могут загружаться только в карту памяти (не могут записываться в память файлов EM).



### Память файлов EM

#### Использование средства программирования

- 1,2,3...
1. В настройках ПЛК укажите первый банк EM для конвертирования в память файлов.
  2. С помощью средства программирования выполните инициализацию памяти файлов EM.
  3. С помощью средства программирования выберите данные модуля CPU (программа пользователя, память ввода/вывода, область параметров) и сохраните эти данные в память файлов EM.
  4. С помощью средств программирования выполните чтение файла из памяти файлов EM в модуль CPU.

#### Использование FREAD(700)/FWRIT(701)/CMND(490)

- 1,2,3...
1. В настройках ПЛК выберите начальный банк EM для конвертации в память файлов.
  2. С помощью средства программирования выполните инициализацию памяти файлов EM.
  3. С помощью FWRIT(701) выберите имя файла в указанной области памяти ввода/вывода и сохраните этот файл в память файлов EM.
  4. С помощью FREAD(700) выполните чтение файла из памяти файлов EM в память ввода/вывода модуля CPU.

Операции над памятью файлов EM можно выполнять, выставляя команды FINS локальному модулю CPU с помощью CMND(490).

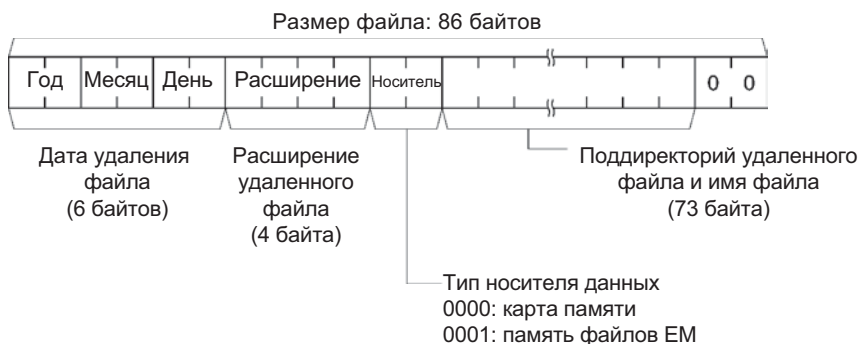
### 5-3-3 Прерывание питания при обращении к памяти файлов

Если в процессе обращения модуля CPU к памяти файлов (к карте памяти или памяти файлов EM) пропадает питание, содержимое карты памяти может быть повреждено. Обновляемый файл может быть записан некорректно, а в некоторых случаях может быть повреждена сама карта памяти.

Поврежденный файл будет удален автоматически системой при следующем включении питания. Будет установлен соответствующий флаг "Уведомление об удалении файла" (A39507 для карты памяти, A39506 для памяти файлов EM). Этот флаг будет сброшен при следующем включении питания.

Когда удаляется файл, в корневом директории карты памяти или памяти файлов EM создается файл протокола удаления (DEL\_FILE.IOM). Файл протокола удаления можно прочитать с помощью CX-Programmer или команды FREAD(700), чтобы узнать следующие сведения: дата удаления файлов, тип имеющейся памяти файлов (носителя данных), поддиректорий, имя файла и расширение. В случае необходимости удаленный файл можно создать повторно или скопировать его.

На следующем рисунке показана структура файла протокола удаления.



## РАЗДЕЛ 6

### Дополнительные функции

Настоящий раздел содержит подробную информацию о следующих дополнительных функциях:

6-1	Длительность цикла/Скоростная обработка . . . . .	249
6-1-1	Минимальная длительность цикла. . . . .	249
6-1-2	Максимальная длительность цикла (Длительность контрольного цикла) . . . . .	250
6-1-3	Контроль длительности цикла . . . . .	250
6-1-4	Скоростные входы. . . . .	251
6-1-5	Прерывания . . . . .	251
6-1-6	Способы обновления входов/выходов . . . . .	252
6-1-7	Отмена циклического обновления для специального модуля ввода/вывода. . . . .	253
6-1-8	Увеличение скорости обновления данных модуля шины CPU . . . . .	254
6-1-9	Максимальное время отклика при обновлении данных через логические связи . . . . .	256
6-1-10	Выполнение в фоновом режиме . . . . .	258
6-1-11	Совместное использование задачами регистров индексов и регистров данных . . . . .	265
6-2	Регистры индексов . . . . .	266
6-2-1	Что такое «регистры индексов»? . . . . .	266
6-2-2	Применение регистров индексов. . . . .	267
6-2-3	Операции, связанные с регистрами индексов . . . . .	270
6-3	Обмен данными через последовательные интерфейсы . . . . .	276
6-3-1	Host Link коммуникации. . . . .	278
6-3-2	Беспротокольные коммуникации . . . . .	283
6-3-3	NT Link (Режим 1:N) . . . . .	284
6-3-4	Последовательные каналы связи с ПЛК (только для модулей CPU CJ1M) . . . . .	285
6-4	Изменение режима обновления текущего значения (PV) таймеров/счетчиков. . . . .	291
6-4-1	Обзор. . . . .	291
6-4-2	Функциональные характеристики . . . . .	292
6-4-3	Выбор и подтверждение режима BCD/двоичного режима . . . . .	293
6-4-4	Мнемонические символы и данные в режиме BCD/двоичном режиме. . . . .	294
6-4-5	Ограничения . . . . .	295
6-4-6	Команды и операнды . . . . .	296
6-5	Применение запланированного прерывания в качестве высокоточного таймера (только для CJ1M). . . . .	299
6-5-1	Задание времени для запланированного прерывания с шагом 0.1 мс. . . . .	299
6-5-2	Выбор сброса таймера при запуске с помощью MSKS(690) . . . . .	300
6-5-3	Чтение текущего значения (PV) внутреннего таймера с помощью MSKR(692) . . . . .	300
6-6	Настройка параметров запуска и техническое обслуживание. . . . .	301
6-6-1	Функции «горячего» запуска/«горячего» останова. . . . .	301
6-6-2	Настройка режима запуска . . . . .	302
6-6-3	Выход RUN . . . . .	303
6-6-4	Настройка времени задержки обнаружения выключения питания . . . . .	303

6-6-5	Запрет прерываний от выключения питания. . . . .	303
6-6-6	Функции часов. . . . .	304
6-6-7	Защита программы. . . . .	305
6-6-8	Защита от записи посредством команд FINS, передаваемых модулю CPU по сети. . . . .	307
6-6-9	Дистанционное программирование и мониторинг. . . . .	308
6-6-10	Профили модулей . . . . .	308
6-6-11	Флеш-память . . . . .	309
6-6-12	Настройка условий запуска . . . . .	310
6-7	Функции диагностики . . . . .	312
6-7-1	Протокол ошибок. . . . .	312
6-7-2	Функция отключения выходов . . . . .	313
6-7-3	Функции формирования предупреждений о неисправностях. . . . .	313
6-7-4	Обнаружение местонахождения неисправности . . . . .	314
6-7-5	Имитирование системных ошибок . . . . .	316
6-7-6	Запрет записи установленных пользователем ошибок FAL в протокол ошибок. . . . .	316
6-8	Режимы работы CPU. . . . .	317
6-8-1	Режимы работы CPU. . . . .	317
6-8-2	Режим параллельного выполнения и минимальные длительности циклов. . . . .	322
6-8-3	Согласованность данных в режиме параллельного выполнения при асинхронном обращении к памяти . . . . .	322
6-9	Режим приоритетного обслуживания периферии. . . . .	322
6-9-1	Режим приоритетного обслуживания периферии . . . . .	323
6-9-2	Временное отключение режима приоритетного обслуживания периферии. . . . .	325
6-10	Работа без батареи . . . . .	328
6-11	Прочие функции . . . . .	330
6-11-1	Настройка времени срабатывания входов/выходов . . . . .	330
6-11-2	Размещение области ввода/вывода . . . . .	331

## 6-1 Длительность цикла/Скоростная обработка

В этом разделе описаны следующие функции:

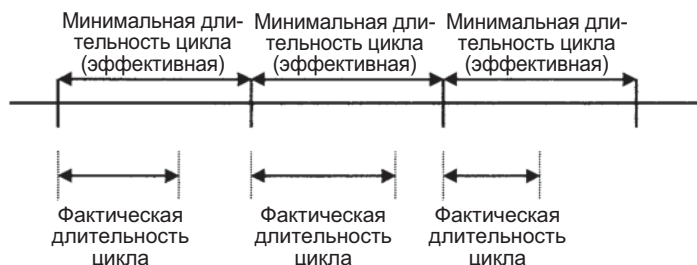
- Установка минимальной длительности цикла
- Установка максимальной длительности цикла (длит. сторожевого цикла)
- Мониторинг длительности цикла
- Быстродействующие входы
- Функции обработки прерываний
- Способы обновления входов/выходов
- Отключение циклического обновления специальных модулей ввода/вывода
- Повышение скорости обновления для логических связей и других данных в модулях шины CPU (только для CS1-H, CJ1-H, CJ1M или CS1D)
- Снижение нестабильности длительности цикла путем выполнения операций с данными в фоновом режиме (только для CS1-H, CJ1-H, CJ1M или CS1D)

### 6-1-1 Минимальная длительность цикла

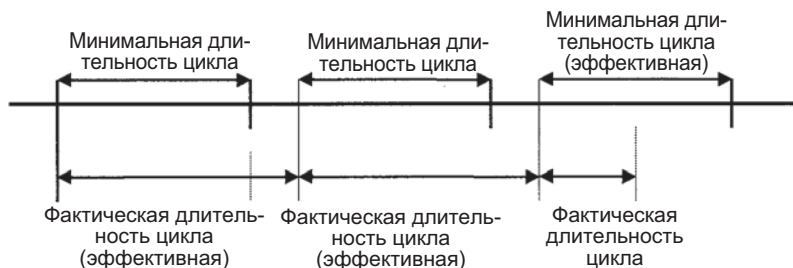
Для ПЛК серии CS/CJ можно установить минимальную (или фиксированную) длительность цикла (см. прим.). Выбрав для выполнения программы фиксированную длительность цикла, можно добиться стабильности значений времени отклика (скорости срабатывания) для входов/выходов.

Примечание Длительность цикла также можно зафиксировать для модуля CPU CS1D в однопроцессорных системах или для модулей CPU CS1-H, CJ1-H или CJ1M, а также в случае применения режима параллельного выполнения.

Минимальная длительность цикла (1...32000 мс) выбирается в настройках ПЛК с шагом 1 мс.



Если фактическая длительность цикла превышает минимальную длительность цикла, функция установки минимальной длительности цикла действовать не будет и длительность цикла будет изменяться от цикла к циклу.



#### Настройки ПЛК

Адрес	Название	Значения	По умолчанию
208 Биты: 0 ... 15	Минимальная длительность цикла	0001...7D00: 1...32000 мс (с шагом 1 мс)	0000 (мин. длит. не задана)

## 6-1-2 Максимальная длительность цикла (Длительность контрольного цикла)

Если длительность цикла (см. примечание) превышает выбранную максимальную длительность цикла, будет установлен (ВКЛ) флаг "Слишком большая длительность цикла" (A40108) и работа ПЛК будет остановлена.

**Примечание** В случае использования режима параллельного выполнения для модулей CPU CS1-H, CJ1-H или CJ1M, а также для модулей CPU CS1D в однопроцессорных системах длительность цикла определяется временем выполнения программы.

### Настройки ПЛК

Адрес	Название	Значения	По умолч.
209 Бит: 15	Активизация выбранной длительности сторожевого цикла	0: По умолчанию (1с) 1: Биты 0 ... 14	0001 (1 с)
209 Биты: 0 ... 14	Длительность сторожевого цикла (Параметр действует, если бит 15 = 1)	001 ... FA0: 10 ... 40,000 мс (с шагом 10 мс)	

### Флаги и слова дополнительной области

Название	Адрес	Описание
Флаг "Слишком большая длительность цикла"	A40108	Если длительность цикла превышает установленную длительность сторожевого цикла, будет включен бит A40108 и модуль CPU прекратит работу. В случае применения режима параллельного выполнения для модулей CPU CS1-H, CJ1-H, CJ1M или CS1D в однопроцессорных системах длительность цикла совпадает со временем выполнения программы.

**Примечание** Если длительность цикла обслуживания периферии превышает 2.0 с для модулей CPU CS1-H, CJ1-H или CJ1M, а также для модулей CPU CS1D в однопроцессорных системах в режиме параллельного выполнения, в этом случае произойдет ошибка превышения длительности цикла обслуживания периферии и модуль CPU прекратит работу. Если это произойдет, будет включен бит A40515 (флаг "Превышена длительность цикла обслуживания периферии").

## 6-1-3 Контроль длительности цикла

Значения максимальной длительности цикла и текущей длительности цикла записываются в дополнительную область в каждом цикле. Для модулей CPU серии CS1-H, CJ1-H или CS1D в однопроцессорных системах в режиме параллельного выполнения записываются значения времени выполнения программ.

### Флаги и слова дополнительной области

Название	Адрес	Описание
Максимальная длительность цикла (время выполнения программы для модулей CPU серии CS1-H, CJ1-H или CJ1M в режиме параллельного выполнения)	A262 и A263	Записывается в каждом цикле в формате 32-битового двоичного числа в следующем диапазоне: 0 ... 429 496 729.5 мс, с шагом 0.1 мс (0 ... FFFF FFFF)
Текущая длительность цикла (время выполнения программы для модулей CPU серии CS1-H, CJ1-H или CJ1M в режиме параллельного выполнения)	A264 и A265	Записывается в каждом цикле в формате 32-битового двоичного числа в следующем диапазоне: 0 ... 429 496 729.5 мс, с шагом 0.1 мс (0 ... FFFF FFFF)

Для чтения среднего за 8 циклов значения длительности цикла можно использовать средство программирования (CX-Programmer или консоль программирования).

**Уменьшение длительности цикла**

С целью уменьшения длительности цикла в ПЛК серии CS/CJ можно использовать следующие методы:

- 1,2,3...**
1. Перевод задач, которые в данный момент не исполняются, в режим ожидания (standby).
  2. Обход разделов программы, которые не должны выполняться, с помощью команд перехода JMP(004) и JME(005).

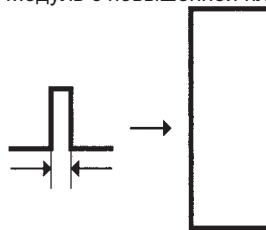
В модулях CPU серии CS1-H, CJ1-H или CS1D (для однопроцессорных систем) в режиме параллельного выполнения значение длительности обслуживания периферии записывается в слово A268 (Длительность цикла обслуживания периферии) в каждом цикле обслуживания.

**6-1-4 Скоростные входы**

Если требуется принимать импульсы, длительность которых короче длительности цикла, следует использовать модуль скоростного ввода CS1W-IDP01, либо применять скоростные входы уплотненных модулей ввода/вывода C200H-ID501/ID215 и C200H-MD501/MD115/MD215.

На скоростные входы можно подавать импульсы, ширина которых (время ВКЛ) составляет 1 мс или 4 мс (для модулей с повышенной плотностью входов C200H), либо 0.1 мс (для модулей скоростного ввода CS1W-IDP01).

Модуль скоростного ввода или  
Модуль с повышенной плотностью входов



CS1W-IDP01: 0.1 мс  
CJ1W-IDP01: 0.05 мс  
C200H-ID501/ID215/MD501/MD115/MD215: 4 мс

**6-1-5 Прерывания**

События, по которым формируются прерывания, перечислены ниже. Более подробное описание можно найти в Разделе 4-3 *Задачи обработки прерываний*.

**Примечание** Модули CPU CS1D для систем с дублированием CPU не поддерживают обработку прерываний. В модулях CPU CS1D задачи обработки прерываний можно использовать только в качестве дополнительных циклических задач.

**Прерывания от входов/выходов  
(задачи обработки прерываний 100...131)**

Задача обработки прерывания от входов/выходов выполняется, когда на соответствующий вход модуля ввода прерываний поступает сигнал (прерывание формируется по переднему фронту сигнала, а в модулях ввода прерываний серии CS/CJ - по переднему или заднему фронту сигнала).

**Запланированные прерывания (задачи обработки прерываний 2 и 3)**

Задача обработки запланированного прерывания выполняется с установленной периодичностью.

**Прерывание от выключения питания (задача обработки прерывания 1)**

Эта задача выполняется в случае пропадания питания.

**Внешние прерывания (задачи обработки прерываний 0 ... 255)**

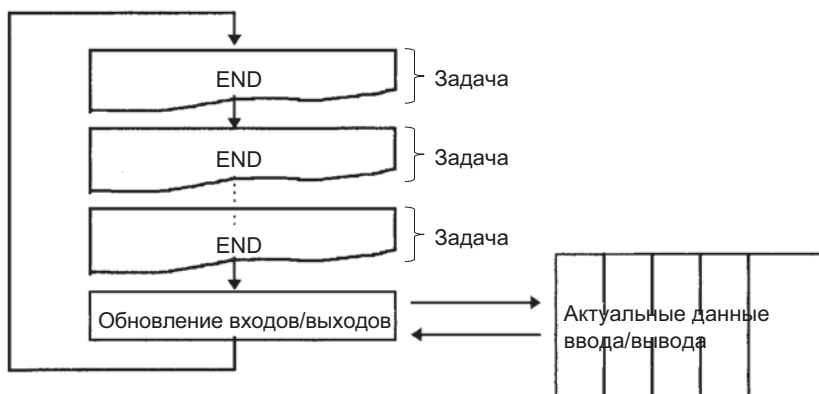
Задача обработки внешнего прерывания выполняется, когда прерывание формируется специальным модулем ввода/вывода, модулем шины CPU или встраиваемой платой.

**Примечание** Встроенные входы прерываний и входы скоростных счетчиков модулей CPU CJ1M можно использовать для вызова задач обработки прерываний. Подробные сведения содержатся в руководстве *CJ Series Built-in I/O Operation Manual*.

### 6-1-6 Способы обновления входов/выходов

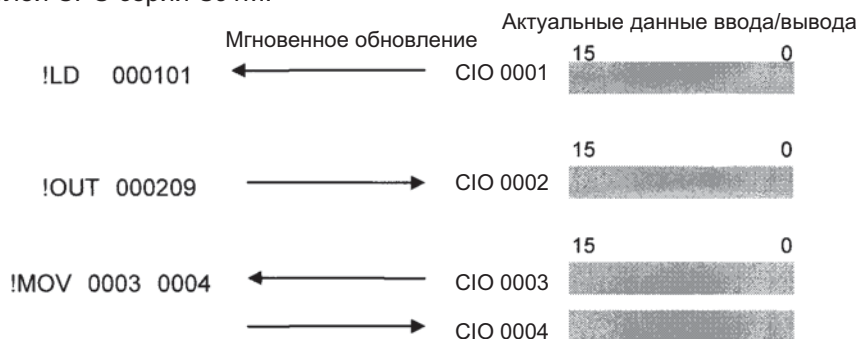
Модули CPU серии CS/CJ могут обновлять данные от базовых модулей ввода/вывода и специальных модулей ввода/вывода, используя один из трех способов: циклическое обновление, мгновенное обновление и выполнение команды IORF(097).

**1. Циклическое обновление** Обновление входов/выходов производится после того, как выполнены все команды в выполняемых задачах (в настройках ПЛК можно отключить циклическое обновление для отдельных специальных модулей ввода/вывода).



**2. Мгновенное обновление** Если в качестве операнда команды, выполняемой с мгновенным обновлением, указан адрес области ввода/вывода, данные, определяемые этим операндом, будут обновлены при выполнении команды. Команды мгновенного обновления могут обновлять данные областей, отведенных для базовых модулей ввода/вывода.

Мгновенное обновление также возможно для встроенных входов/выходов модулей CPU серии CJ1M.



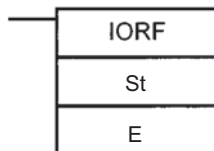
- Примечание**
1. Если команда содержит битовый операнд, будет обновлено все слово, содержащее этот бит. Если операнд имеет длину слова, будет обновлено это слово.
  2. Обновление данных ввода и входных данных производится непосредственно перед выполнением команды. Обновление данных вывода и выходных (адресуемых) данных производится непосредственно после выполнения команды.
  3. Команды с мгновенным обновлением выполняются дольше обычных команд, поэтому длительность цикла возрастает. Более подробно об этом можно прочитать в разделе 10-5 *Время выполнения команд и количество шагов* в руководстве *Operation Manual*.

4. Модули CPU CS1D для систем с дублированием CPU не поддерживают мгновенное обновление.

3. Выполнение IORF(097) и DLNK(226)

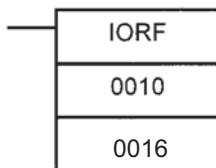
■ **IORF(097): I/O REFRESH**

Для обновления диапазона слов ввода/вывода можно использовать команду IORF(097). Обновление происходит после выполнения этой команды. Команда IORF(097) позволяет обновить данные областей, отведенных для базовых модулей ввода/вывода и специальных модулей ввода/вывода.



St: Первое слово  
E: Последнее слово  
Когда выполняется IORF(097), обновляются данные во всех словах между St и E.

Ниже показан пример применения IORF(097) для обновления восьми слов данных ввода/вывода.



Когда выполняется IORF(097), обновляются семь слов в диапазоне CIO 0010 - CIO 0016.

Если для выполнения расчетов требуется быстрое обновление входов и выходов, команду IORF(097) следует размещать непосредственно перед командами, выполняющими расчеты, и непосредственно после них.

**Примечание**

Команда IORF(097) характеризуется относительно большим временем выполнения, причем время выполнения возрастает пропорционально количеству обновляемых слов, поэтому она может существенно увеличить длительность цикла. Более подробно это описано в Разделе 10-5 *Время выполнения команд и количество шагов* в руководстве *Operation Manual*.

■ **DLNK(226): Обновление входов/выходов модуля шины CPU (только для модулей CS1-H, CJ1-H, CJ1M или CS1D)**

Команда DLNK(226) служит для обновления данных модуля шины CPU с указанным номером модуля. Обновляются следующие данные:

- Слова, отведенные для модуля в области CIO
- Слова, отведенные для модуля в области DM
- Специальные данные модуля (см. примечание).

Примечание Специальные данные модуля шины CPU включают в себя логические связи (модули Controller Link или модули SYSMAC LINK), а также удаленные входы/выходы (модули DeviceNet).



N: Номер модуля шины CPU

Пример:



Приведенная команда выполнит обновление слов, отведенных для модуля в области CIO и области DM, а также специальных данных модуля шины CPU с номером 1.

Пример применения: в случае большой длительности цикла период обновления данных логических связей модуля Controller Link может оказаться очень большим. Этот период можно сократить, выполняя для модуля Controller Link команду DLNK(226), увеличивая тем самым частоту обновления логических связей.

**6-1-7 Отмена циклического обновления для специального модуля ввода/вывода**

Для каждого специального модуля ввода/вывода в области, отведенной для специальных модулей ввода/вывода, резервируется 10 слов (CIO 2000 ... CIO 2959) в соответствии с номером модуля, установленным с помощью переключателя на передней панели модуля.



Обновление и обмен данными между этой областью и модулем CPU происходят в каждом цикле в процессе обновления входов/выходов, но это циклическое обновление можно отменить отдельно для каждого модуля в настройках ПЛК.

Как правило, существует три причины для отмены циклического обновления:

- 1,2,3... 1. Циклическое обновление для специальных модулей ввода/вывода можно отменить в том случае, когда длительность цикла слишком велика из-за большого количества установленных специальных модулей ввода/вывода.
2. Если время обновления входов/выходов слишком мало, модуль может не справиться с такой повышенной нагрузкой, будет включен флаг «Ошибка специального модуля ввода/вывода» (A40206) и специальный модуль ввода/вывода будет работать некорректно. В этом случае длительность цикла можно увеличить, выбрав соответствующее значение минимальной длительности цикла в настройках ПЛК или отменив циклическое обновление входов/выходов для специального модуля ввода/вывода.
3. Циклическое обновление для специального модуля ввода/вывода всегда следует отменять, если обновление для него будет производиться задачей обработки прерываний с помощью команды IORF(097). Если циклическое обновление и команда IORF(097) выполняются одновременно для одного и того же модуля, в этом случае произойдет ошибка задачи обработки прерывания и будет установлен флаг «Ошибка задачи обработки прерывания» (A40213).

Если циклическое обновление было запрещено, обновление данных специального модуля ввода/вывода можно осуществлять в процессе выполнения программы с помощью команды IORF(097).

#### Настройки ПЛК

Для отмены циклического обновления для специальных модулей ввода/вывода 0...95 служат 96 битов в словах 226...231.

Адрес	Название	Состояние	По умолчанию
226 бит 0	Бит отмены циклического обновления для специального модуля ввода/вывода 0	0: Разрешено 1: Отменено	0 (Разрешено)
:	:	:	:
231 бит 15	Бит отмены циклического обновления для специального модуля ввода/вывода 95	0: Разрешено 1: Отменено	0 (Разрешено)

### 6-1-8 Увеличение скорости обновления данных модуля шины CPU

Эта функция поддерживается только модулями CPU CS1-H, CJ1-H, CJ1M или CS1D.

Как правило, обновление логических связей и других специальных данных для модулей шины CPU выполняется одновременно с обновлением слов областей CIO и DM, отведенных для модулей, в рамках обновления входов/выходов, производимого по завершении цикла выполнения программы.

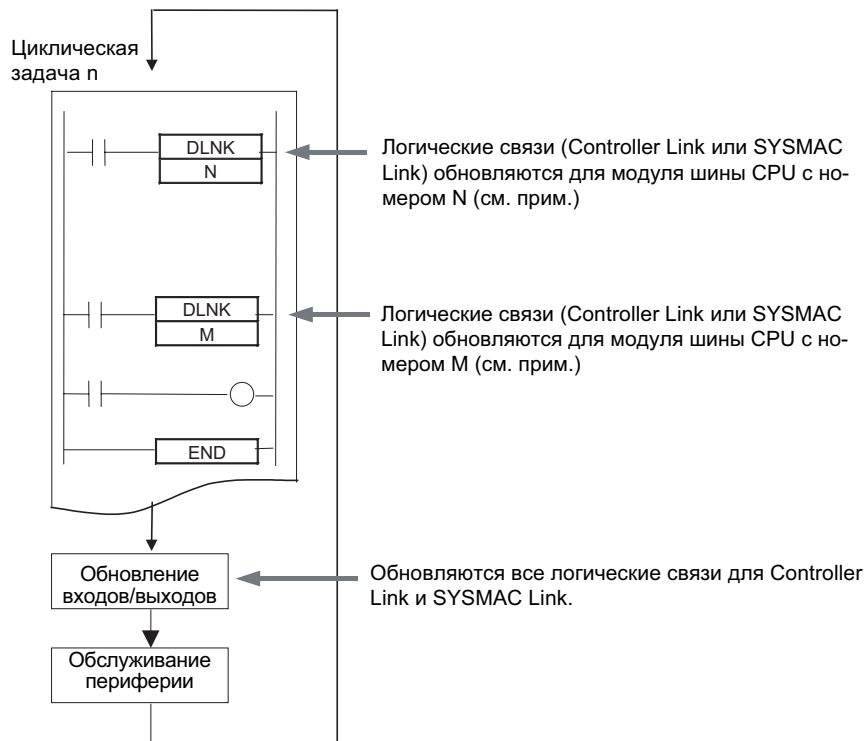
В следующей таблице приведены некоторые примеры специальных данных для модулей шины CPU.

Модули	Специальные данные
Модули Controller Link и модули SYSMAC LINK	Логические связи (data link) для модулей Controller Link и SYSMAC LINK (включая автоматические и настраиваемые пользователем логические связи)
Модули DeviceNet серии CS/CJ	Данные систем удаленного ввода/вывода DeviceNet (включая фиксированные области и области, устанавливаемые пользователем)

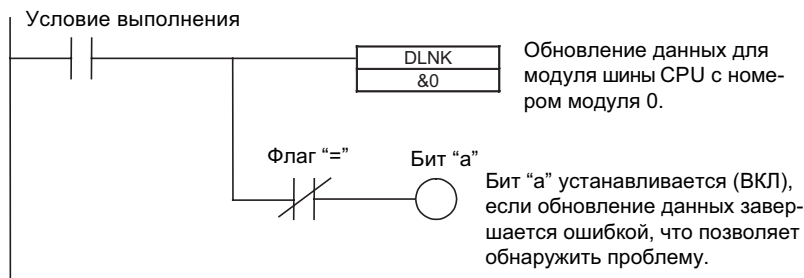
Ниже перечислены функции, которые можно использовать для повышения скорости обновления данных модулей шины CPU, применяемых с модулями CPU CS1-H, CJ1-H, CJ1M или CS1D.

- Уменьшение длительности цикла за счет применения режима параллельного выполнения или скоростных команд (режим параллельного выполнения не поддерживается модулями CPU CS1D для систем с дублированием CPU).
- Выполнение команды DLNK(226) для обновления специальных модулей шины CPU с указанными номерами (DLNK(226) можно использовать в программе несколько раз).

- Примечание** 1. Более продолжительный цикл (напр., 100 мс) приведет к увеличению периода обновления логических связей. В этом случае можно использовать команду DLNK(226) (см. пример ниже).



- Примечание** Если команда DLNK(226) выполняется для модуля шины CPU, который занят обновлением данных, в этом случае данные обновлены не будут и будет сброшен флаг "Равенство". В общем случае флаг "Равенство" следует использовать в программе, чтобы следить за ошибками при выполнении обновления (см. пример ниже).

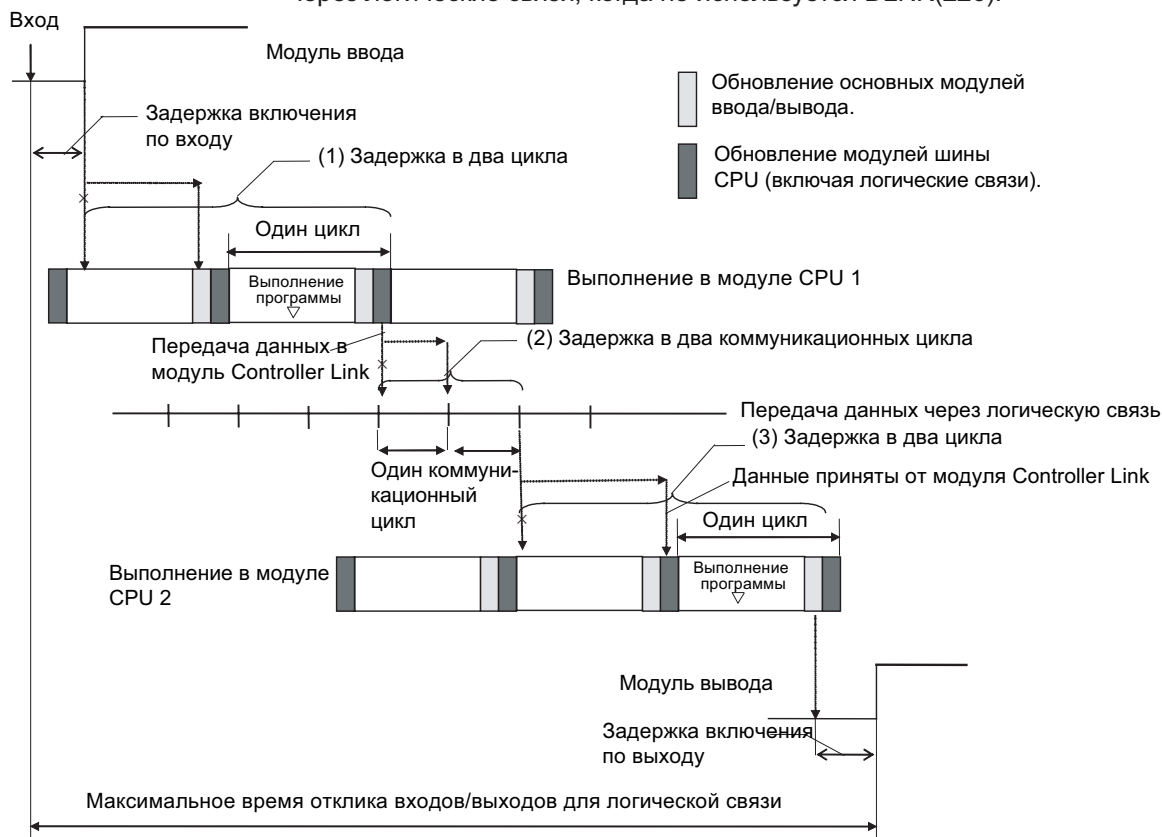


1. Команда IORF(097) используется для обновления данных для базовых модулей ввода/вывода и специальных модулей ввода/вывода. DLNK(226) предназначена для обновления модулей шины CPU (слов областей CIO и DM, отведенных для модулей, и специальных данных для модулей).

### 6-1-9 Максимальное время отклика при обновлении данных через логические связи

**Выполнение в обычном режиме**

На следующем рисунке показан поток данных, приводящий к максимальному времени отклика входов/выходов при обновлении данных через логические связи, когда не используется DLNK(226).



На этом рисунке показаны три точки, в которых происходит задержка выполнения, увеличивающая время отклика логической связи по входам/выходам.

**1,2,3...**

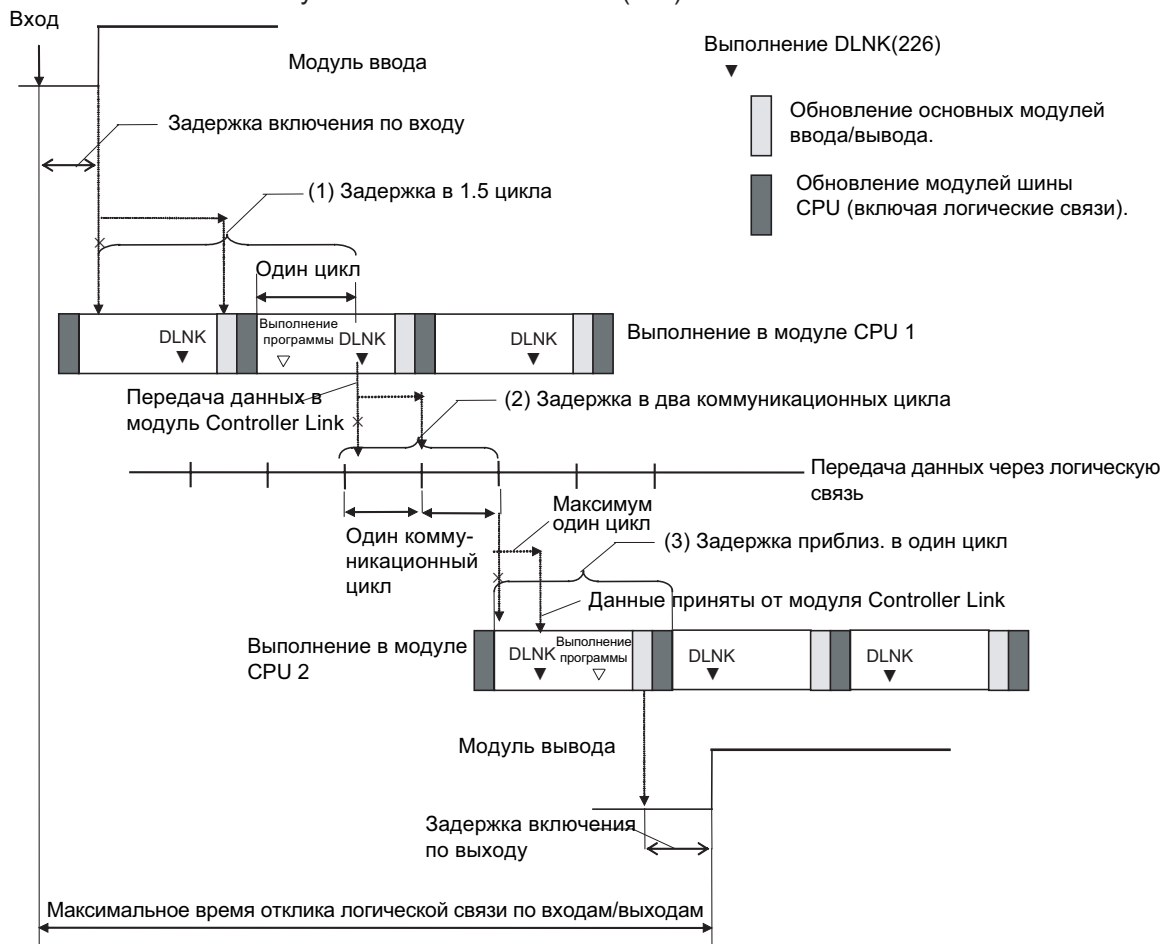
1. Входной сигнал поступает в ПЛК (модуль CPU 1) только после обновления входов/выходов, что приводит к задержке длительностью в один цикл, прежде чем состояние входа будет считано в ПЛК. Обновление модулей шины CPU происходит после выполнения программы, что в итоге приводит к задержке длительностью в два цикла.
2. Обмен данными произойдет только после того, как ПЛК передаст метку и станет, в результате, опрашивающим узлом, что приводит к задержке в один коммуникационный цикл, прежде чем данные могут быть переданы через логическую связь. Кроме того, задержка длительностью в один коммуникационный цикл происходит после приема метки, т.е., суммарная задержка достигает двух коммуникационных циклов.
3. Данные, передаваемые через логическую связь, поступают в ПЛК (модуль CPU 2) после сеанса обмена данными, поэтому данные не будут считаны в ПЛК до следующего обмена данными, что приводит к задержке длительностью в один цикл. Обновление модулей шины CPU происходит после выполнения программы, т.е., суммарная задержка составляет два цикла.

В следующей таблице приведен расчет максимальной задержки отклика логической связи по входам/выходам:

Задержка включения по входу	1.5 мс
Длительность цикла ПЛК в модуле CPU 1 x 2	25 мс x 2
Длительность коммуникационного цикла x 2	10 мс x 2
Длительность цикла ПЛК в модуле CPU 2 x 2	20 мс x 2
Задержка включения по выходу	15 мс
<b>Суммарная задержка (время отклика логической связи по входам/выходам)</b>	<b>126.5 мс</b>

Применение DLNK(226)

На следующем рисунке показан поток данных, приводящий к максимальному времени отклика логической связи по входам/выходам в случае использования DLNK(226).



На этом рисунке показаны три точки, в которых происходит задержка выполнения, увеличивающая время отклика логической связи по входам/выходам.

**Примечание** В этом примере предполагается, что в обоих модулях CPU команда DNLK(226) размещена после других команд программы.

1,2,3...

1. Входной сигнал поступает в ПЛК (модуль CPU 1) только после обновления входов/выходов, что приводит к задержке длительностью в один цикл, прежде чем состояние входа может быть считано в ПЛК. Обновление модулей шины CPU производится во время выполнения программы, что снижает суммарную задержку, приблизительно, до 1.5 циклов.
2. Обмен данными происходит только после того, как ПЛК получает метку, т.е., становится опрашиваемым узлом, что вызывает задержку длительностью до одного коммуникационного цикла, прежде чем данные могут быть переданы через логическую связь. Кроме того, задержка длительностью в один коммуникационный цикл наблюдается после приема метки, т.е., суммарная задержка достигает двух коммуникационных циклов.
3. Данные, передаваемые через логическую связь, поступают в ПЛК (модуль CPU 2) после обновления входов/выходов, но команда DLNK(226) обновляет данные, поэтому данные будут считаны в модуль ПЛК без задержки длительностью в один цикл. Обновление базовых модулей ввода/вывода происходит после выполнения программы, что приводит к суммарной задержке длительностью, приблизительно, в один цикл.

Максимальное время отклика логической связи по входам/выходам определяется следующим образом:

Задержка включения по входу	1,5 мс	---
Длительность цикла ПЛК модуля CPU 1 x 1.5	25 мс x 1.5	Задержка < на 12,5 мс (25 мс x 0.5)
Длительность коммуникационного цикла x 2	10 мс x 2	---
Длительность цикла ПЛК модуля CPU 2 x 1	20 мс x 1	Задержка < на 20 мс (20 мс x 1)
Задержка включения по входу	15 мс	---
Суммарная задержка (время отклика логической связи по входам/выходам)	94 мс	Задержка < на 32,5 мс (быстрее на 26%)

### 6-1-10 Выполнение в фоновом режиме

Чтобы уменьшить нестабильность длительности цикла, можно использовать режим выполнения в фоновом режиме. Выполнение в фоновом режиме поддерживается лишь модулями CPU CS1-H, CJ1-H или CJ1M, а также модулями CPU CS1D для однопроцессорных систем.

Операции обработки табличных данных (например, поиск данных) и обработки текстовых строк (например, поиск текстовой строки) могут существенно изменять длительность цикла, поскольку их выполнение занимает значительное время.

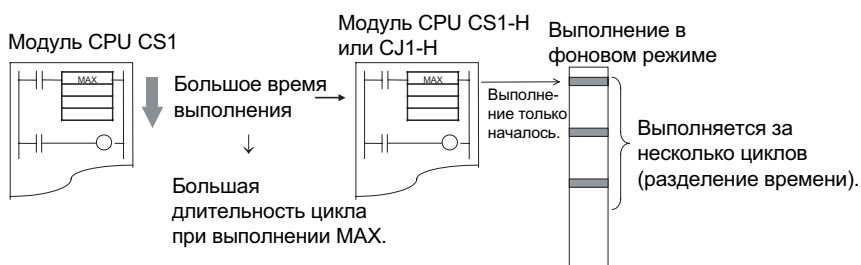
При работе с модулями CPU серии CS1-H, CJ1-H или CJ1M, либо с модулями CPU CS1D для однопроцессорных систем (см. примечание) можно, однако, использовать режим выполнения в фоновом режиме (режим разделения времени). В этом режиме команды, перечисленные ниже, выполняются в течение нескольких циклов, что позволяет снизить их влияние на длительность цикла. В настройках ПЛК можно выбрать режим фонового выполнения для команды каждого типа.

**Примечание**

Модули CPU CS1D для систем с дублированием CPU не поддерживают выполнение в фоновом режиме.

- Команды обработки табличных данных
- Команды обработки текстовых строк
- Команды сдвига данных (только ASYNCHRONOUS SHIFT REGISTER)

Выбор режима выполнения в фоновом режиме для указанных выше команд позволяет снизить эффект временного увеличения длительности цикла.



**Применение**

Выполнение в фоновом режиме можно использовать для обработки большого объема данных, например, когда компиляция или обработка данных должна выполняться только в определенные моменты времени (например, один раз в день), и уменьшение влияния на длительность цикла более важно, чем скорость обработки данных.

**Последовательность действий**

- 1,2,3...**
1. В настройках ПЛК разрешите выполнение в фоновом режиме требуемых команд.
  2. Выберите в настройках ПЛК номер порта связи (номер логического порта), который будет использоваться для выполнения в фоновом режиме. Этот номер порта будет применяться для всех команд, выполняемых в фоновом режиме.

**Примечание** Для всех команд, выполняемых в фоновом режиме, используется один порт. Следовательно, выполнение в фоновом режиме для команды не может быть начато, если в этот момент другая команда уже выполняется в фоновом режиме. Чтобы в любой момент времени в фоновом режиме выполнялась только одна команда, следует предусмотреть управление командами, выбранными для выполнения в фоновом режиме, с использованием флага "Применение порта связи разрешено".

3. Если команда, для которой выбрано выполнение в фоновом режиме, должна быть выполнена, ее выполнение будет начато только в том цикле, в котором выполнилось условие выполнения, но выполнение не будет завершено в том же цикле.
4. Когда начинается выполнение в фоновом режиме, флаг "Применение порта связи разрешено" для этого порта сбрасывается (ВЫКЛ).
5. Выполнение в фоновом режиме будет продолжаться в течение нескольких циклов.
6. Когда выполнение завершится, включится флаг "Применение порта связи разрешено" для этого порта. В результате будет разрешено выполнение другой команды в фоновом режиме.

#### Поддерживаемые команды

##### ■ Команды обработки табличных данных

Команда	Мнемоника	Код функции
DATA SEARCH	SRCH	181
SWAP BYTES	SWAP	637
FIND MAXIMUM	MAX	182
FIND MINIMUM	MIN	183
SUM	SUM	184
FRAME CHECKSUM	FCS	180

##### ■ Команды обработки текстовых строк

Команда	Мнемоника	Код функции
MOVE STRING	MOV\$	664
CONCATENATE STRING	+\$	656
GET STRING LEFT	LEFT\$	652
GET STRING RIGHT	RIGHT\$	653
GET STRING MIDDLE	MID\$	654
FIND IN STRING	FIND\$	660
STRING LENGTH	LEN\$	650
REPLACE IN STRING	RPLC\$	661
DELETE STRING	DEL\$	658
EXCHANGE STRING	XCHG\$	665
CLEAR STRING	CLR\$	666
INSERT INTO STRING	INS\$	657

##### ■ Команды сдвига данных

Команда	Мнемоника	Код функции
ASYNCHRONOUS SHIFT REGISTER	ASFT	017

#### Отличие команд, выполняемых в обычном режиме, от команд, выполняемых в фоновом режиме

Ниже описаны различия между выполнением команд в обычном режиме и выполнением в фоновом режиме.

### ■ Запись в регистры индексов (IR)

Если для записи в регистр индексов адреса слова, содержащего максимальное или минимальное значение (адреса в памяти ввода/вывода), используется команда MAX(182) или MIN(183), в этом случае адрес не будет записан в регистр индексов, а будет записан в A595 и A596. Для сохранения адреса в регистре индексов используйте команду сдвига данных (например, MOVL(498)), чтобы скопировать адрес, содержащийся в A595 и A596, в регистр индексов.

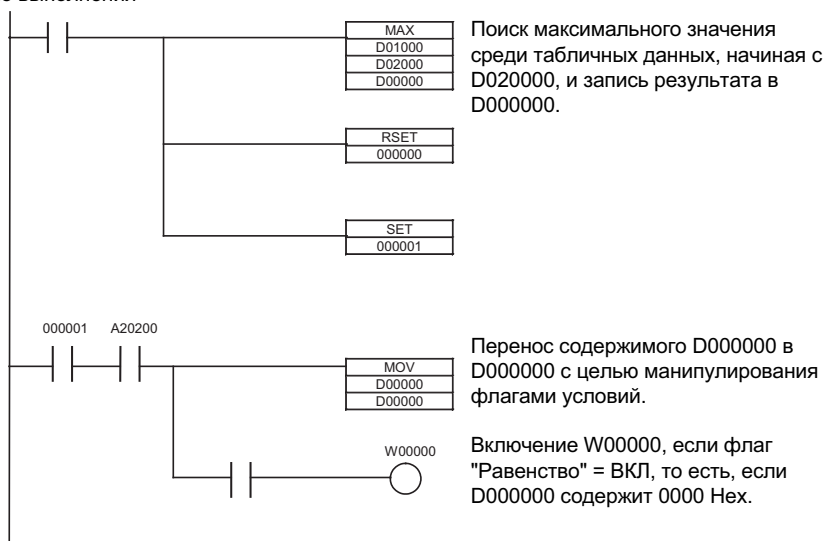
### ■ Флаги условий

Флаги условий не обновляются после выполнения команд, выполненных в фоновом режиме. Чтобы считать состояние флага условия, сначала следует выполнить команду, влияющую на флаги условий (см. пример ниже), и лишь после этого обращаться к флагам условий.

#### Пример:

Команда MOV(021) влияет на состояние флагов "Равенство" и "Отрицательное значение" точно так же, как и MAX(182). То есть, они обе включают флаг "Равенство" в случае значения 0 и устанавливают (ВКЛ) флаг "Отрицательное значение", если MSB=ВКЛ. Следовательно, команду MOV(021) можно использовать с целью копирования результатов выполнения команды MAX(182) по тому же адресу, то есть, управлять флагами условий с целью получения их состояния.

Условие выполнения



### ■ Запись в регистр индексов IR00

Если для записи в регистр индексов адреса слова, содержащего совпадшее значение (1-е слово, если слов несколько) (адрес слова в карте памяти ввода/вывода), используется команда SRCH(181), в этом случае адрес не будет записан в регистр индексов, а запишется в A595 и A596.

### ■ Запись в регистры данных (DR) для SRCH(181)

Если для записи совпадающих данных в регистр данных используется команда SRCH(181), данные не будут записаны в регистр данных, а запишутся в A597.

### ■ Совпадающие текстовые строки

Если команда SRCH(181) обнаружит совпадающие данные, она не установит флаг "Равенство", а установит бит A59801.

### ■ Ошибки выполнения команд

Если в процессе выполнения команды в фоновом режиме возникает ошибка выполнения команды или ошибка доступа к данным, в этом случае флаги ER или AER не будут установлены, а будет установлен бит A39510. Бит A39510 остается включенным до тех пор, пока команда выполняется в фоновом режиме.

### ■ Запись в регистры данных (DR) для команд MAX(1820 и MIN(183)

Если выполняются команды MAX(182) или MIN(183) и в качестве адресата для записи минимального или максимального значения выбран регистр данных, в этом случае произойдет ошибка выполнения команды и установится флаг ER.

#### Настройки ПЛК

Слово	Биты	Название	Настройка	Значение по умолчанию и момент обновления
198	15	Выполнение команды обработки табличных данных в фоновом режиме	0: Не выполняется в фоновом режиме 1: Выполняется в фоновом режиме	0: Не выполняется в фоновом режиме
	14	Выполнение команды обработки текстовой строки в фоновом режиме	0: Не выполняется в фоновом режиме 1: Выполняется в фоновом режиме	Начало работы 0 Hex: Порт 0 Начало работы
	13	Выполнение команды сдвига данных в фоновом режиме	0: Не выполняется в фоновом режиме 1: Выполняется в фоновом режиме	
	00 ... 03	Номер порта связи для выполнения в фоновом режиме	0 ... 7 Hex: Порты связи 0 ... 7 (внутренние логические порты)	0 Hex: Порт 0 Начало работы

#### Флаги и слова дополнительной области

Название	Адрес	Описание
Флаги "Применение порта связи разрешено"	A20200 ... A20207	Флаг установлен, если для порта связи с соответствующим номером может быть выполнена соответствующая команда (SEND, RECV, CMND или PMCR) либо для порта связи с соответствующим номером может быть применен режим фонового выполнения (только для модулей CPU CS1D для однопроцессорных систем или модулей CPU CS1-H, CJ1-H или CJ1M). Биты 00 ... 07 соответствуют портам связи 0 ... 7.  Если для карты памяти модуля CPU CS1-H, CJ1-H, CJ1M или CS1D выполняется операция простого резервного сохранения (запись или сравнение данных), в этом случае автоматически выделяется порт связи, и во время работы устанавливается соответствующий флаг, который сбрасывается по завершению операции.
Флаги "Ошибка порта связи"	A21900 ... A21907	Флаг устанавливается, если во время выполнения сетевой команды (SEND, RECV, CMND или PMCR) происходит ошибка. Биты 00 ... 07 соответствуют портам связи 0 ... 7.  Если для карты памяти модуля CPU CS1-H, CJ1-H, CJ1M или CS1D выполняется операция простого резервного сохранения (запись или сравнение данных), в этом случае автоматически выделяется порт связи. Соответствующий флаг устанавливается, если происходит ошибка, и сбрасывается, если функция резервного сохранения выполняется без ошибок.



Название	Адрес	Описание
Коды завершения для портов связи	A203 ... A210	Эти слова содержат коды завершения для портов с соответствующими номерами по завершению выполнения сетевых команд (SEND, RECV, CMND или PMCR). По завершению выполнения в фоновом режиме содержимое этих слов обнуляется (для модуля CPU CS1D для однопроцессорных систем или модулей CPU CS1-H, CJ1-H или CJ1M). Слова A203 ... A210 соответствуют портам связи 0 ... 7.  Если для карты памяти модуля CPU CS1-H, CJ1-H, CJ1M или CS1D выполняется простое резервное сохранение (запись или сравнение данных), в этом случае автоматически выделяется порт связи и в соответствующее слово записывается код завершения.
Флаг ER/AER для режима фонового выполнения	A39510	Флаг устанавливается, если в процессе выполнения команды в фоновом режиме возникает ошибка выполнения или происходит обращение по недопустимому адресу. Флаг сбрасывается при включении питания или при запуске.
Запись в IR00 в режиме фонового выполнения	A595 и A596	Если для размещения результата команды, выполняемой в фоновом режиме, выбран регистр индексов, результат записывается не в IR00, а в эти слова. Диапазон: 0000 0000 ... FFFF FFFF Hex 4 младших разряда: A595, 4 старших разряда: A596
Запись в DR00 в режиме фонового выполнения	A297	Если для размещения результата команды, выполняемой в фоновом режиме, выбран регистр данных, результат записывается не в DR00, а в эти слова. Диапазон: 0000 ... FFFF Hex
Флаг "Равенство" в режиме фонового выполнения	A59801	Если в процессе выполнения команды SRCH(181) в фоновом режиме будут обнаружены совпадающие данные, этот флаг включится.

**Примечание** Порты связи (внутренние логические порты) модуля CPU используются как в режиме фонового выполнения, так и для следующих команд.

- SEND(090), RECV(098) и CMND(490) (команды сетевых коммуникаций)
- PMCR(260) (PROTOCOL MACRO)

Команды, выполняемые в фоновом режиме, и команды, перечисленные выше, не могут использовать одновременно один и тот же порт. Следует обеспечить, чтобы для каждого порта одновременно выполнялась только одна команда. Используйте для этого флаги "Применение порта связи разрешено".

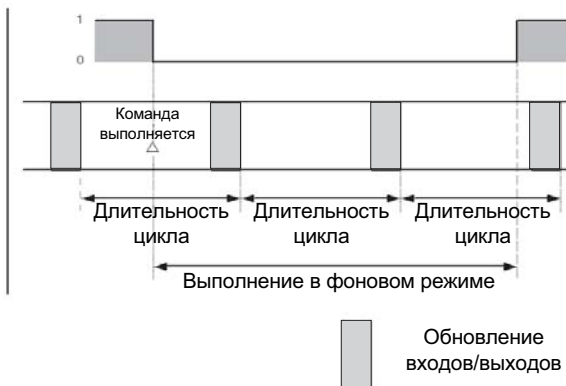
**Примечание** Если для команды, выбранной для выполнения в фоновом режиме, указан порт, для которого флаг "Применение порта связи разрешено" сброшен (ВЫКЛ), в этом случае будет установлен флаг ER и команда в фоновом режиме выполнена не будет.

#### Флаги "Применение порта связи разрешено"

Флаг "Применение порта связи разрешено" установлен (ВКЛ), когда соответствующий порт не используется, и сброшен (ВЫКЛ), если порт уже используется.

Флаг "Применение порта связи разрешено"

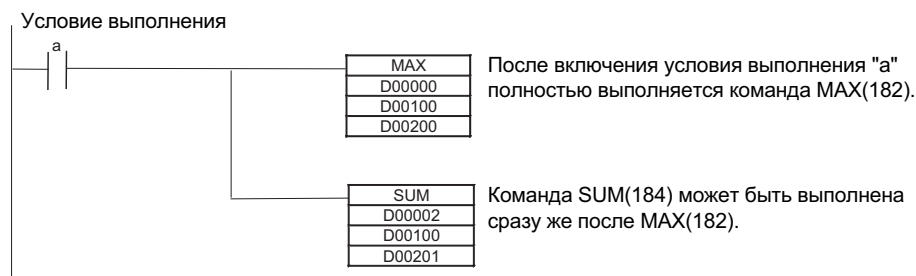
Выполнение команд программы пользователя в фоновом режиме



### Пример программы 1

#### ■ Традиционные программы без режима фонового выполнения

Как показано ниже, обработка завершается после выполнения команды.



#### ■ Программы с режимом фонового выполнения

В случае использования режима фонового выполнения программу изменяют таким образом, чтобы команда MAX(182) выполнялась только после включения указанного флага "Применение порта связи разрешено" (то есть, только тогда, когда порт не используется для выполнения в фоновом режиме или для сетевых коммуникаций). Кроме того, с помощью команд SET и RESET контролируются входные условия с целью обеспечения выполнения в требуемом порядке (в следующем примере для выполнения в фоновом режиме используется порт связи 0).



## Пример программы 2

В следующем примере показано выполнение в фоновом режиме для случая, когда для вывода результатов выбран регистр индексов, что возможно для команд MAX(182), MIN(183) и SRCH(181).

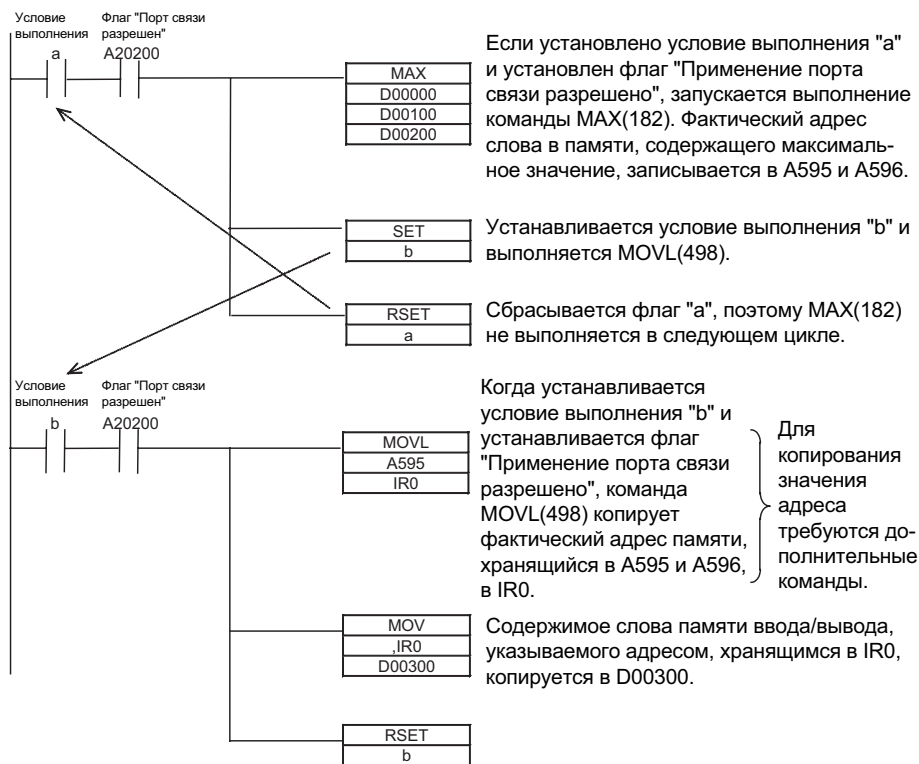
### ■ Традиционные программы без режима фонового выполнения

Как показано ниже, фактический адрес слова в памяти, содержащего максимальное значение, записывается в регистр индексов.



### ■ Программы с режимом фонового выполнения

В случае выполнения в фоновом режиме фактический адрес памяти для слова, содержащего максимальное значение, записывается в слова A595 и A596. После этого используется команда MOVL(498) для записи адреса памяти в регистр индексов.



### 6-1-11 Совместное использование задачами регистров индексов и регистров данных

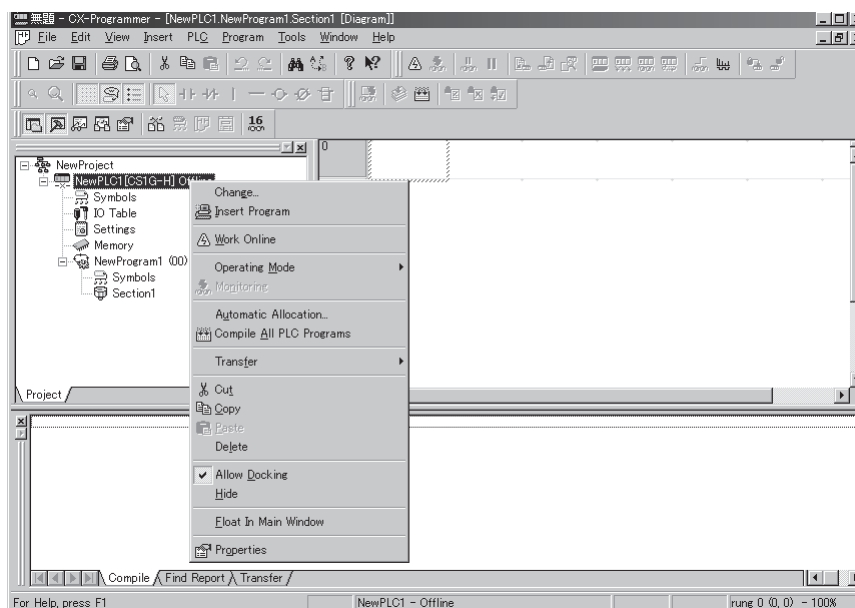
Совместное использование регистров индексов и регистров данных (IR/DR) несколькими задачами поддерживается только модулями CPU CS1-H, CJ1-H, CJ1M или CS1D. По умолчанию для каждой задачи используются отдельные регистры. Текущий режим можно определить с помощью бита A09914.

- Примечание**
1. Чтобы избежать множества операций записи и чтения регистров при обмене общими данными между задачами, можно применять регистры индексов и данных совместного пользования. Информацию о записи и чтении данных в/из регистров индексов можно найти в соответствующих разделах руководств *CS Series Operation Manual (W339)* или *CJ Series Operation Manual (W393)*.
  2. При совместном использовании регистров индексов и данных переключение между двумя задачами происходит немного быстрее. Совместное использование регистров рекомендуется применять в том случае, когда регистры не используются, либо нет особой необходимости в использовании отдельных регистров для каждой задачи.

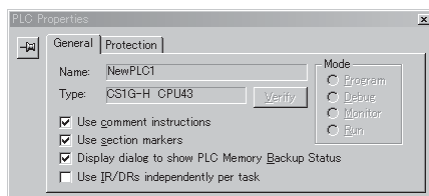
#### Настройка

Совместное использование регистров индексов и регистров данных устанавливается в CX-Programmer. Консоль программирования для этого не подходит.

- 1,2,3... 1. Выберите на дереве проекта в CX-Programmer объект PLC (ПЛК) и щелкните по нему правой кнопкой мыши.



2. Выберите **Properties** (Свойства). Откроется следующее диалоговое окно.



3. Если для каждой задачи необходимо использовать отдельные регистры индексов и данных, оставьте установленной галочку, отвечающую за независимое использование IR/DR. Снимите эту галочку, если предусматривается совместное использование регистров индексов и данных всеми задачами.

#### Флаги и слова дополнительной области

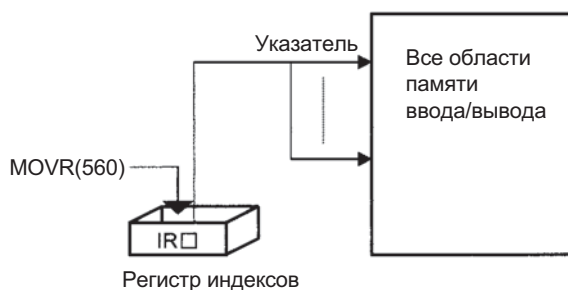
Название	Адрес	Описание
Режим использования IR/DR задачами	A09914	Флаг указывает, будут ли регистры индексов и данных использоваться совместно всеми задачами. 0: Отдельные регистры для каждой задачи (по умолчанию) 1: Регистры используются совместно всеми задачами

## 6-2 Регистры индексов

### 6-2-1 Что такое "регистры индексов"?

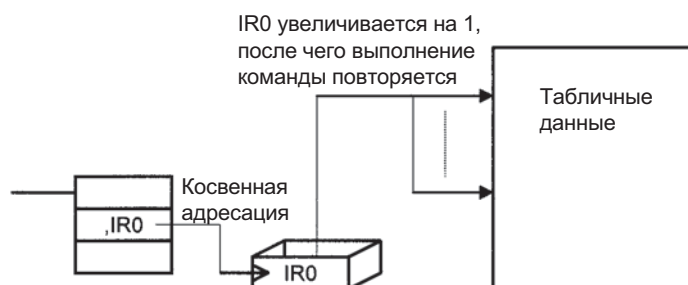
Регистры индексов действуют как указатели. Они указывают на определенные адреса памяти ПЛК (абсолютные адреса памяти ввода/вывода). Записав адрес памяти ПЛК в регистры индексов с помощью команды MOVR(560) или MOVRW(561), в дальнейшем регистр индексов можно вводить в качестве операнда для других команд с целью косвенной адресации к содержимому памяти ПЛК.

Преимущество регистров индексов состоит в том, что они могут ссылаться на любой бит или слово в памяти ввода/вывода, в том числе на текущие значения (PV) таймеров и счетчиков.



## 6-2-2 Применение регистров индексов

Регистры индексов можно очень эффективно применять в циклах FOR-NEXT. Содержимое регистров индексов очень легко инкрементируется, декрементируется и смещается. Таким образом, всего несколько команд в циклическом контуре позволяют обрабатывать таблицы данных (данных, расположенных в памяти последовательно).



### Основные операции

Ниже приведен пример типичной процедуры, в которой применяются регистры индексов:

- 1,2,3...
1. С помощью команды MOVR(560) адрес памяти ПЛК для требуемого бита или слова записывается в регистр индексов.
  2. Регистр индексов указывается в качестве операнда для большинства команд с целью косвенной адресации к требуемому биту или слову.
  3. Первоначальный адрес памяти ПЛК смещается или инкрементируется (см. ниже) для перевода указателя на другой адрес.
  4. Действия 2 и 3 повторяются, то есть, команды выполняются для любого требуемого количества адресов.

### Смещение, инкрементирование (+1) и декрементирование (-1) адресов

В следующей таблице перечислены возможности, предусмотренные для косвенной адресации.

Операция	Синтаксис
Косвенная адресация	,IR□
Косвенная адресация с постоянным смещением	Константа, IR□ (Укажите в константе знак "+" или "-")
Косвенная адресация со смещением DR	DR□, IR□
Косвенная адресация с автоинкрементом (+1)	Приращение на 1: ,IR□ + Приращение на 2: ,IR□ ++
Косвенная адресация с автодекрементом (-1)	Уменьшение на 1: ,- IR□ Уменьшение на 2: ,- - IR□

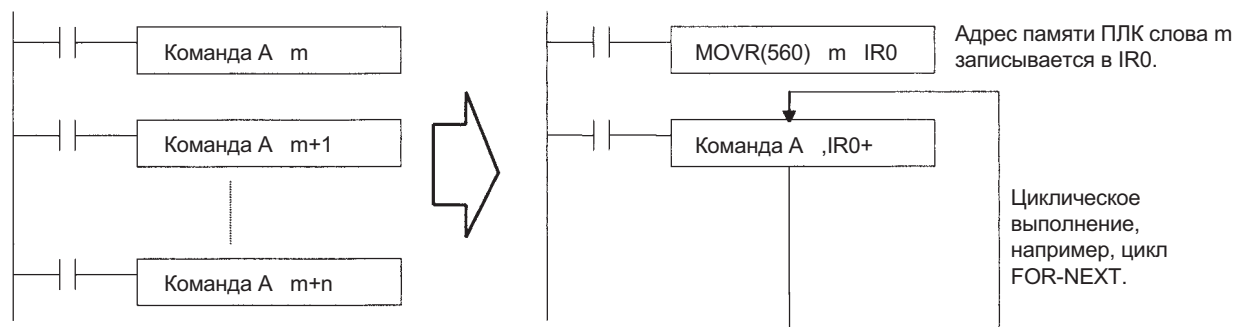
### Команды, обращающиеся к регистрам индексов на прямую

Прямой доступ к регистрам индексов имеют следующие команды.

DOUBLE SIGNED BINARY ADD WITHOUT CARRY: +L(401), DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY: -L(411), DOUBLE INCREMENT BINARY: ++L(591) и DOUBLE DECREMENT BINARY: --L(593)

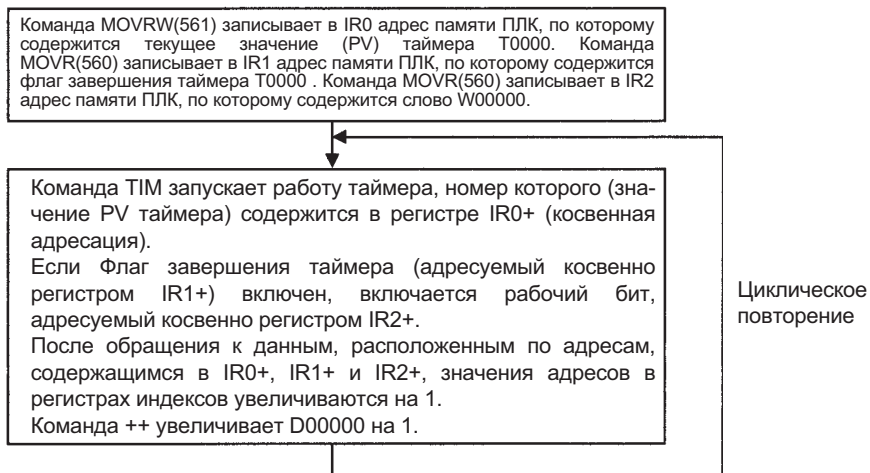
**Пример 1**

Ниже показано, как применение регистра индексов в циклической программе позволяет заменить одной командой целую последовательность команд. В этом случае команда A повторяется n+1 раз, что позволяет обработать соответствующее количество табличных значений (например, выполнить чтение и сравнение).

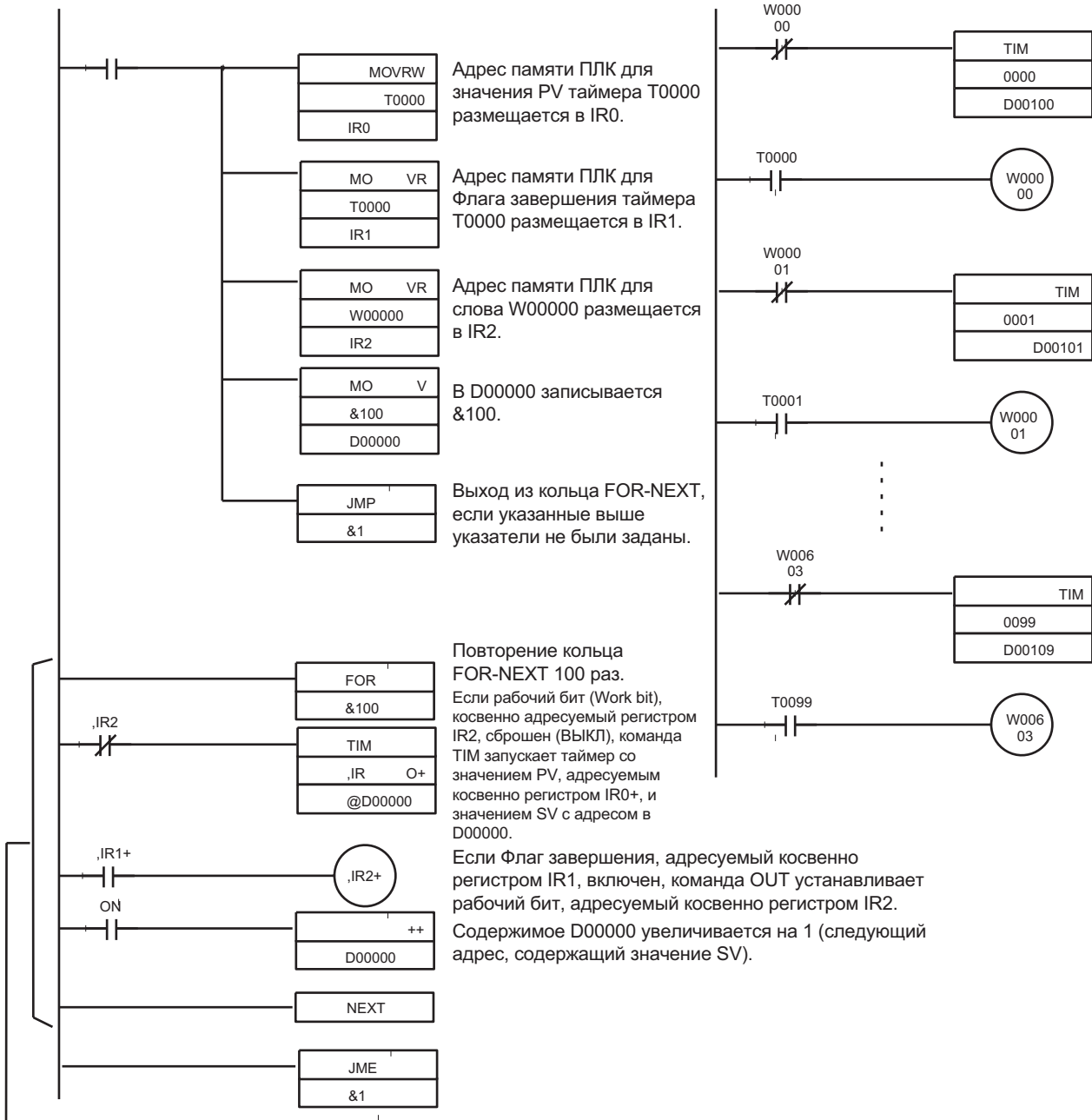


**Пример 2**

Ниже приведен пример использования регистров индексов в цикле FOR-NEXT с целью настройки и запуска 100 таймеров (T0000 ... T099), задания (SV) которых содержатся в словах D00100 ... D00109. В регистрах индексов указываются номер и флаг завершения для каждого таймера, затем цикл повторяется, причем каждый раз адреса в регистрах индексов увеличиваются на 1.



Приведенная слева подпрограмма, состоящая из 11 команд, эквивалентна подпрограмме, приведенной справа, содержащей 200 команд.



Цикл FOR-NEXT выполняется 100 раз и запускает таймеры T0000 ... T0099. При этом содержимое регистров IR0 (номер таймера/адрес PV), IR1 (адрес Флага завершения), IR2 (адрес рабочего бита) и D00000 (адрес SV) увеличивается постоянно на 1.



**Прямая адресация к регистрам индексов**

Прямая адресация к регистрам индексов возможна только для команд, перечисленных в следующей таблице.

Группа команд	Название команд	Мнемоника	Основная функция
Команды для перемещения данных	MOVE TO REGISTER	MOVR(560)	Адрес памяти ПЛК для бита или слова записывается в регистр индексов.
	MOVE TIMER/COUNTER PV TO REGISTER	MOVW(561)	
Команды для обработки табличных данных	SET RECORD LOCATION	SETR(635)	Чтение адреса памяти ПЛК, хранящегося в регистре индексов.
	GET RECORD NUMBER	GETR(636)	
Команды для перемещения данных	DOUBLE MOVE	MOVL(498)	Передача адресов между регистрами индексов. Служит для операций обмена и сравнения.
	DOUBLE DATA EXCHANGE	XCGL(562)	
Команды сравнения	DOUBLE EQUAL	=L(301)	
	DOUBLE NOT EQUAL	<>L(306)	
	DOUBLE LESS THAN	<L(311)	
	DOUBLE LESS THAN OR EQUAL	<=L(316)	
	DOUBLE GREATER THAN	>L(321)	
	DOUBLE GREATER THAN OR EQUAL	>=L(326)	
	DOUBLE COMPARE	CMPL(060)	
Команды инкремента/декремента	DOUBLE INCREMENT BINARY	++L(591)	Изменение адреса памяти ПЛК в регистре индексов путем инкрементирования (+1), декрементирования (-1) или смещения содержимого регистра индексов.
	DOUBLE DECREMENT BINARY	--L(593)	
Команды символьной математики	DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+L(401)	
	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	-L(411)	
Специальные команды	CONVERT ADDRESS FROM CV	FRMCV(284)	Преобразование физических адресов памяти ПЛК серии CV из/в адреса ПЛК серии CS/CJ. (Только для модулей CS1-H, CJ1-H, CJ1M или CS1D)
	CONVERT ADDRESS TO CV	TOCV(285)	

**Примечание** Для команд с операндами двойной длины (то есть, для команд с "L" на конце) используются регистры индексов IR0 ... IR15, поскольку каждый регистр содержит два слова.

**6-2-3 Операции, связанные с регистрами индексов**

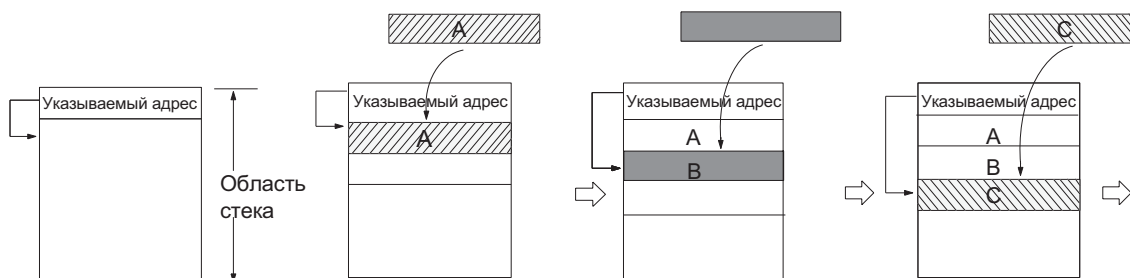
Команды модулей CPU серии CS/CJ, предназначенные для обработки табличных данных, расширяют возможности, предоставляемые регистрами индексов. Эти команды грубо можно классифицировать, как команды для обработки стека и команды для обработки таблиц.

Операция	Назначение	Команды
Обработка стека	Создание таблиц данных FIFO ("первым вошел - первым вышел") или LIFO ("последним вошел - первым вышел") и чтение, запись, вставка, удаление или подсчет элементов таблиц данных.	SSET(630), PUSH(632), FIFO(633), LIFO(634). Для модулей CPU CS1-H, CJ1-H, CJ1M или CS1D также команды SREAD(639), SWRITE(640), SINS(641), SDEL(642), SNUM(638)

Операция		Назначение	Команды
Обработка таблиц	Таблицы со строками длиной в одно слово (Команды, оперирующие диапазоном (масивом) данных)	Стандартная обработка	Определение контрольной суммы, поиск конкретного значения, максимального значения или минимального значения в определенном диапазоне значений.
		Специальная обработка	Выполнение различных операций над таблицами, например, сравнение или сортировка.
	Таблицы со строками, состоящими из нескольких слов (Команды, для обработки строк таблиц)		Обработка данных, имеющих форму строк длиной в несколько слов.
			Использование регистров индексов совместно с такими командами, как FCS(180), SRCH(181), MAX(182), MIN(183) и SUM(184)
			Использование регистров индексов совместно с такими командами, как SRCH(181), MAX(182), MIN(183) и командами сравнения.
			Использование регистров индексов совместно с такими командами, как DIM(631), SETR(635), GETR(636) и командами сравнения.

**Обработка стека** Команды обработки стека выполняют операции со специальными таблицами данных, определенными как "стеки". Для выборки данных из стека может использоваться принцип "первым вошел - первым вышел" (FIFO) или "последним вошел - первым вышел" (LIFO).

Определенная область (диапазон) памяти ввода/вывода должна быть объявлена как стек. Первые слова стека определяют длину стека и содержат указатель стека. Всякий раз, когда происходит запись данных в стек, указатель стека увеличивается на 1, указывая на следующий адрес, по которому должны быть записаны данные.

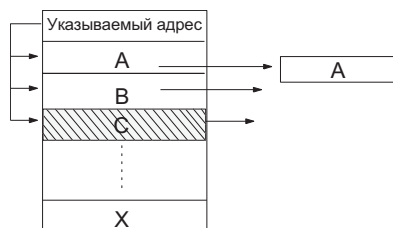


(Приводится состояние содержащего указателя стека до записи данных в стек)

**Примечание** Фактически, в первых двух словах стека содержится адрес памяти ПЛК последнего слова стека, а в следующем слове содержится указатель стека.

**FIFO ("первым вошел - первым вышел")**

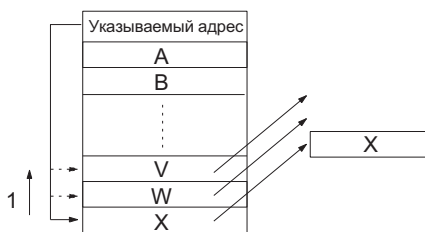
Использование стека в режиме FIFO ("первым вошел - первым вышел") продемонстрировано на следующем рисунке.



Чтение самого старого слова данных, хранящегося в стеке. Всякий раз, когда читается это слово, указатель уменьшается на 1, перемещаясь на следующий адрес.

**LIFO ("последним вошел - первым вышел")**

Использование стека в режиме LIFO ("последним вошел - первым вышел") продемонстрировано на следующем рисунке.

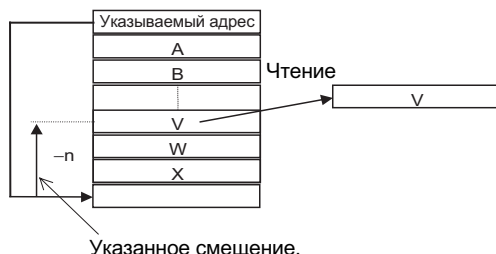


Чтение самого "свежего" слова данных, хранящегося в стеке. Всякий раз, когда читается это слово, указатель уменьшается на 1, перемещаясь на следующий адрес.

Данные прочитанной позиции остаются неизменными.

**Оперирование отдельными строками таблицы данных**

Чтение, запись, вставку или удаление можно выполнять для отдельных строк таблицы. На следующем рисунке показан пример чтения.

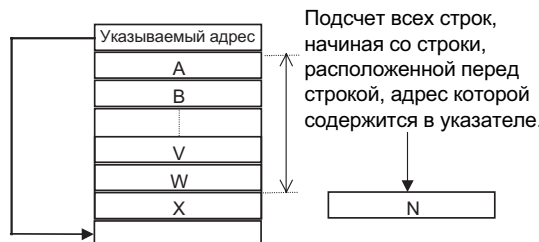


Читаются данные в позиции, определяемой смещением от адреса, содержащегося в указателе.

Оперирование отдельными строками таблицы может пригодиться, например, для поиска деталей, обрабатываемых конвейером.

**Подсчет элементов таблицы**

На следующем рисунке показан пример подсчета элементов таблицы.



Подсчет всех строк, начиная со строки, расположенной перед строкой, адрес которой содержится в указателе.

Подсчитывается количество строк в таблице данных, начиная со строки, расположенной перед строкой, на которую указывает указатель.

Это может быть полезно, например, при подсчете деталей на конвейере.

**Команды для выполнения операций со стеком**

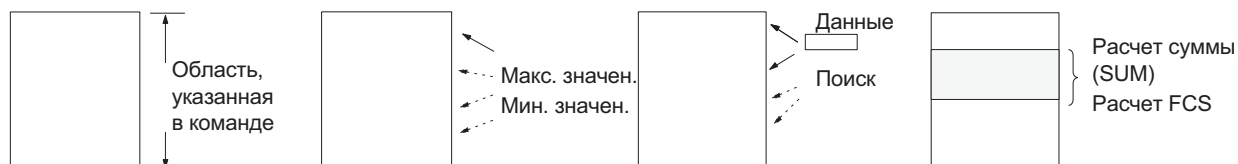
В следующей таблице перечислены команды обработки стека и выполняемые ими функции. Типичным применением стека является обработка информации о хранящихся товарах для систем автоматизации складов, обработка результатов испытаний и оперирование данными о деталях, обрабатываемых на конвейере.

Команда	Назначение
SSET(630)	Определение области стека.
PUSH(632)	Запись данных в следующее свободное слово стека.
FIFO(633)	Чтение данных из стека по принципу "первым вошел - первым вышел".
LIFO(634)	Чтение данных из стека по принципу "последним вошел - первым вышел".
SREAD(639)	Чтение определенной строки таблицы (только для модулей CPU CS1-H, CJ1-H, CJ1M или CS1D).
SWRITE(640)	Запись в определенную строку таблицы (только для модулей CPU CS1-H, CJ1-H, CJ1M или CS1D).
SINS(641)	Вставка определенной строки в таблицу (только для модулей CPU CS1-H, CJ1-H, CJ1M или CS1D).
SDEL(642)	Удаление определенной строки из таблицы (только для модулей CPU CS1-H, CJ1-H, CJ1M или CS1D).
SNUM(638)	Подсчет количества строк таблицы (только для модулей CPU CS1-H, CJ1M или CS1D).

**Обработка таблиц (команды для операций над массивами)**

Команды для операций над массивами оперируют определенным диапазоном (областью слов), который можно считать таблицей, состоящей из строк длиной в одно слово. Эти команды выполняют основные операции, например, поиск максимального или минимального значения в определенном диапазоне, поиск определенного значения в диапазоне или расчет суммы или FCS (контрольной суммы).

Адрес памяти ПЛК слова результата (слова, содержащего макс. значение, мин. значение, обнаруженное слово и т.п.) автоматически записывается в IR0. В дальнейшем регистр индексов (IR0) может использоваться в качестве операнда для таких команд, как MOV(021), если потребуется прочитать содержимое слова или выполнить другие операции над этим словом.



В следующей таблице перечислены команды для операций над массивами и их функции.

Команда	Функция	Описание
SRCH(181)	Поиск данных	Поиск определенного значения в указанной области слов и вывод адреса памяти ПЛК слова, содержащего искомое значение, в регистр IR0.
MAX(182)	Поиск макс. значения	Поиск максимального значения в указанном диапазоне слов и вывод адреса памяти ПЛК слова, содержащего это значение, в регистр IR0.
MIN(183)	Поиск мин. значения	Поиск минимального значения в указанном диапазоне слов и вывод адреса памяти ПЛК слова, содержащего это значение, в регистр IR0.
SUM(184)	Расчет суммы	Расчет суммы значений в указанном диапазоне.
FCS(180)	Расчет контрольной суммы (FCS)	Расчет контрольной суммы кадра данных в указанном диапазоне.

Регистры индексов можно использовать с другими командами (например, командами сравнения) в циклах FOR - NEXT, реализуя тем самым более сложные операции над массивами слов.

**Обработка таблиц (команды для таблиц, состоящих из строк)**

Команды обработки строковых таблиц предназначены для обработки специально созданных таблиц данных, состоящих из строк одинаковой длины. Обработка упрощается благодаря возможности обращения к строке путем указания ее номера.

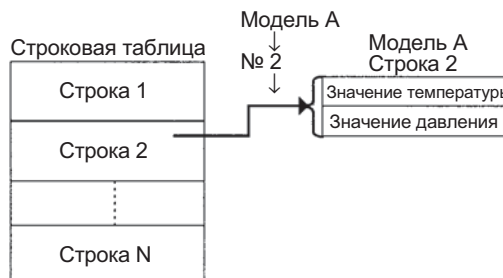
Команда	Функция	Описание
DIM(631)	Создание строковой таблицы.	Команда объявляет длину каждой строки и количество строк.
SETR(635)	Определение расположения строки.	Команда записывает информацию о расположении указанной строки (адрес памяти ПЛК, по которому располагается начало строки) в указанный регистр индексов.
GETR(636)	Чтение расположения строки.	Команда возвращает номер строки, расположенной по адресу памяти ПЛК, содержащемуся в указанном регистре индексов.

**Примечание**

Для оперирования номерами строк и адресами слов используются регистры индексов. Номер строки указывается в команде SETR(635), которая записывает в регистр индексов адрес памяти ПЛК, по которому содержится начало этой строки. Когда необходимо прочитать данные из строки, к содержимому регистра индексов добавляется требуемое смещение, что позволяет обратиться к любому слову в строке.

Команды обработки строковых таблиц используются совместно с регистрами индексов для выполнения следующих типов операций: чтение/запись данных, содержащихся в строках, поиск строк, сортировка данных, содержащихся в строках, сравнение данных, содержащихся в строках, и выполнение расчетов с данными, содержащимися в строках.

Типичным применением строковых таблиц является хранение производственной информации ("рецептов") для различных моделей продукции (например, значения температуры и давления) в виде отдельных строк и переход от одной модели к другой путем простой смены номера строки.

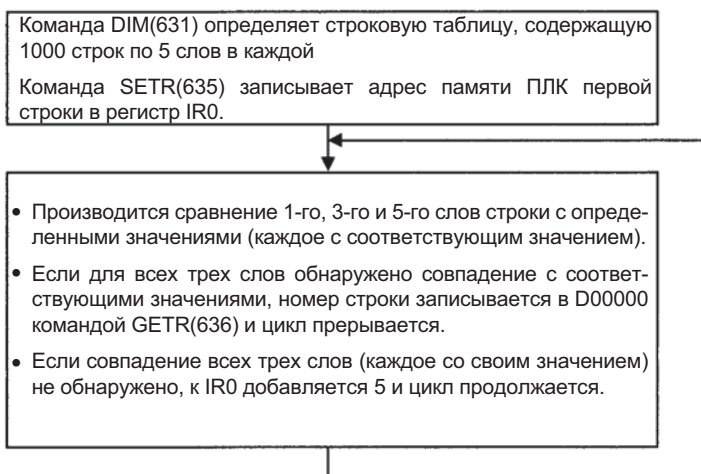


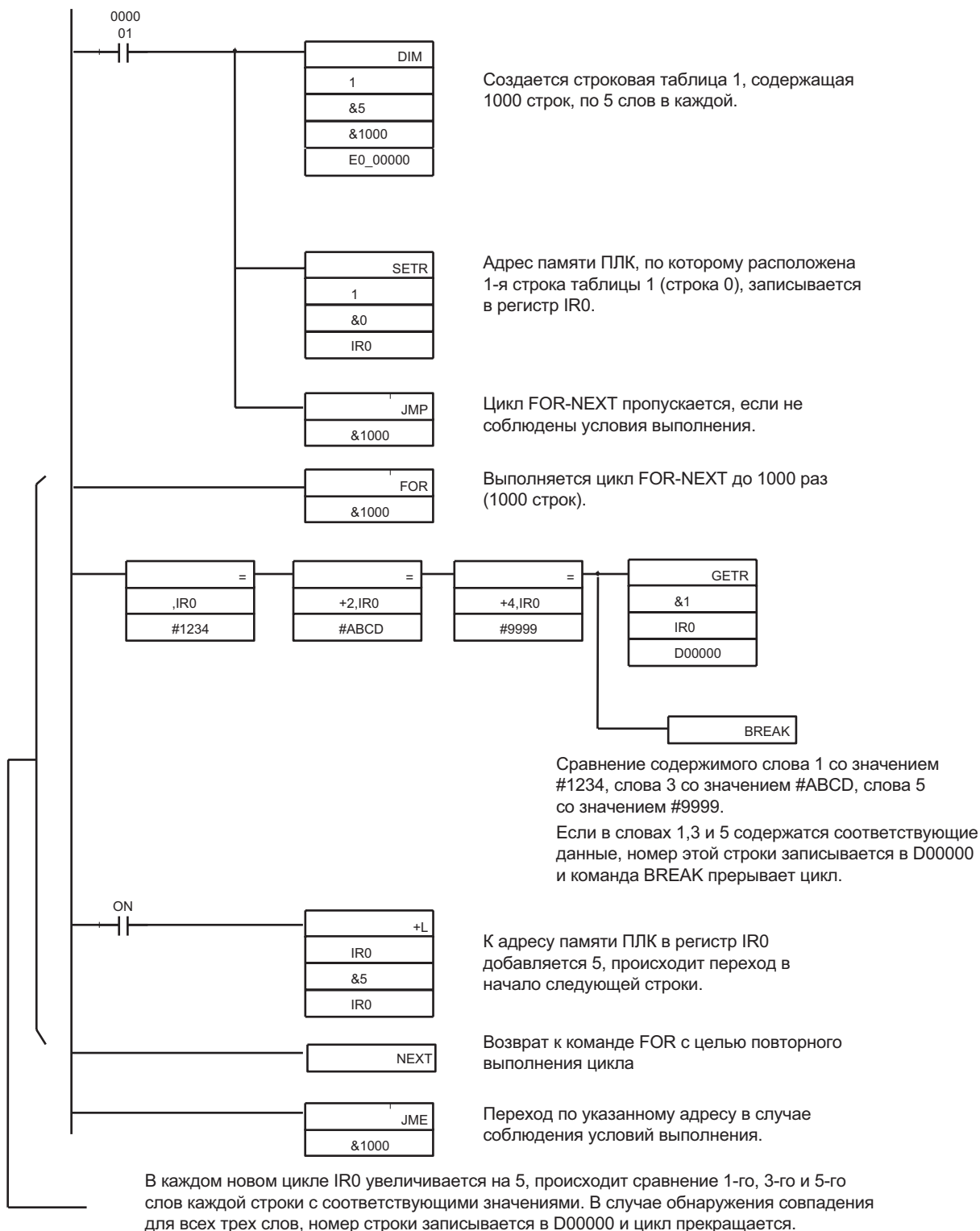
В общем случае, для работы со строковыми таблицами выполняются следующие действия:

- 1,2,3... 1. Определяется структура строковой таблицы с помощью команды DIM(631) и в регистр индексов с помощью команды SETR(635) записывается адрес памяти ПЛК, по которому располагается начало строки.
2. Для чтения или сравнения слов, содержащихся в строке, адрес памяти ПЛК, хранящийся в регистре индексов, смещается или инкрементируется.
3. Для перехода к следующей строке адрес памяти ПЛК, хранящийся в регистре индексов, смещается или инкрементируется.
4. Действия 2 и 3 повторяются требуемое количество раз.

#### Примечание

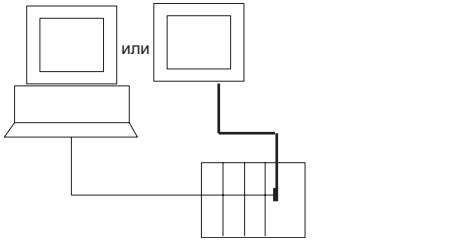
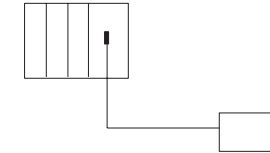
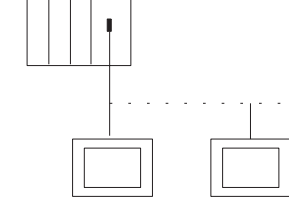
Ниже приведен пример использования регистров индексов и команд для работы со строковыми таблицами с целью сравнения трех значений, содержащихся в словах 1, 3 и 5 каждой строки. В случае обнаружения совпадения номер такой строки записывается в D00000.

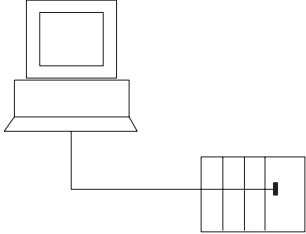




## 6-3 Обмен данными через последовательные интерфейсы

Модули CPU серии CS/CJ поддерживают перечисленные ниже функции связи по последовательному интерфейсу. В следующей части настоящего раздела будут подробно описаны функции связи Host link и беспротокольные коммуникации.

Протокол	Схема соединений	Описание	Порты	
			Периферия	RS-232C
Host link	<p>Центральная станция</p> <p>или</p> <p>OMRON PT (программируемый терминал)</p> 	<p>1) Путем формирования команд Host link или команд FINS на стороне центральной станции для модуля CPU можно выполнять различные команды управления (напр., чтение и запись из/в память ввода/вывода, изменение режима работы, принудительная установка/сброс битов).</p> <p>2) Также можно формировать команды FINS на стороне модуля CPU для центральной станции с целью передачи данных или информации.</p> <p>Host link коммуникации можно использовать для мониторинга данных, например, для слежения за рабочим состоянием, за возникновением ошибок или за признаками качества в ПЛК, либо для передачи в ПЛК таких данных, как запланированный объем производства.</p>	OK	OK
Беспротокольная связь	<p>Стандартное внешнее устройство</p> 	<p>Организация связи со стандартными устройствами, подключенными к порту RS-232C, для которых процедура обмена «команда-ответ» не специфицирована. Вместо этого, в программе выполняются команды TXD(236) и RXD(235), осуществляющие передачу данных через канал передачи или чтение данных через приемный канал. При этом можно определить заголовки кадров и коды завершения.</p>	Не допускается	OK
NT link 1:N или 1:1	<p>OMRON PT (программируемые терминалы)</p> 	<p>Обмен данными с программируемыми терминалами (PT) можно осуществлять, не программируя коммуникации в модуле CPU.</p>	OK	OK

Протокол	Схема соединений	Описание	Порты	
			Периферия	RS-232C
Периферийная шина	<p>Средства программирования (кроме консолей программирования)</p> 	<p>Обеспечение скоростного обмена данными со средствами программирования, кроме консолей программирования.</p> <p>(Удаленное программирование через модемы не поддерживается)</p>	ОК	ОК
Каналы последовательной связи с ПЛК (Serial PLC Link) (только для CJ1M)		<p>До 10-ти слов на один модуль могут использоваться совместно девятью (макс.) модулями CPU, в том числе одним опрашивающим модулем и восемью опрашиваемыми модулями.</p> <p>К порту RS-232C каждого модуля CPU можно подключить преобразователь для интерфейса RS-422A с целью реализации обмена через интерфейс RS-422A/485. Кроме того, два модуля CPU можно соединить через порты RS-232C.</p> <p>Связь по последовательному интерфейсу с ПЛК (каналы Serial PLC Link) также поддерживают в качестве опрашиваемых модулей программируемые терминалы через свои каналы NT Link (1:N) в сочетании с модулями CPU CJ1M.</p>	Не допускается	ОК

Далее будет описана связь через интерфейс Host Link и связь со свободно программируемым протоколом (беспротокольный обмен данными).

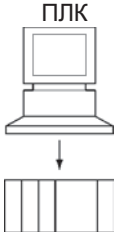
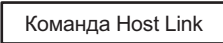
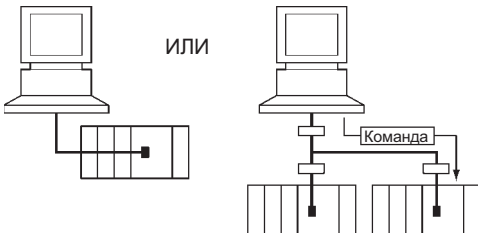

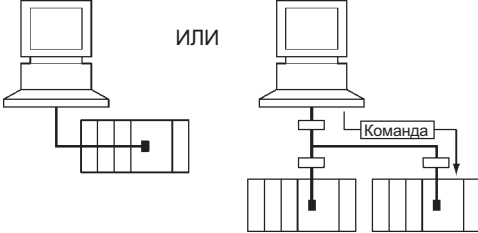
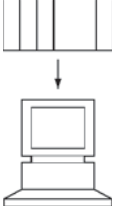

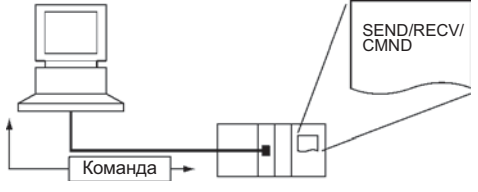
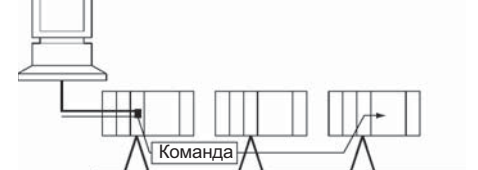
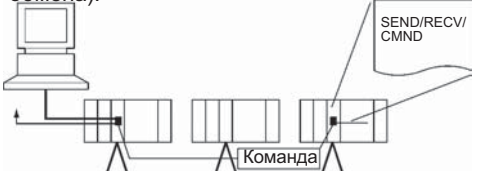
#### Примечание

Модуль CJ1W-CIF11 не имеет гальванической развязки и суммарное расстояние связи для него не превышает 50 м. Если охватываемое расстояние превышает 50 м, следует использовать NT-AL001 с гальванической развязкой и не применять CJ1W-CIF11. Если используется только NT-AL001, суммарная длина тракта связи может составлять до 500 м.



## 6-3-1 Host Link коммуникации

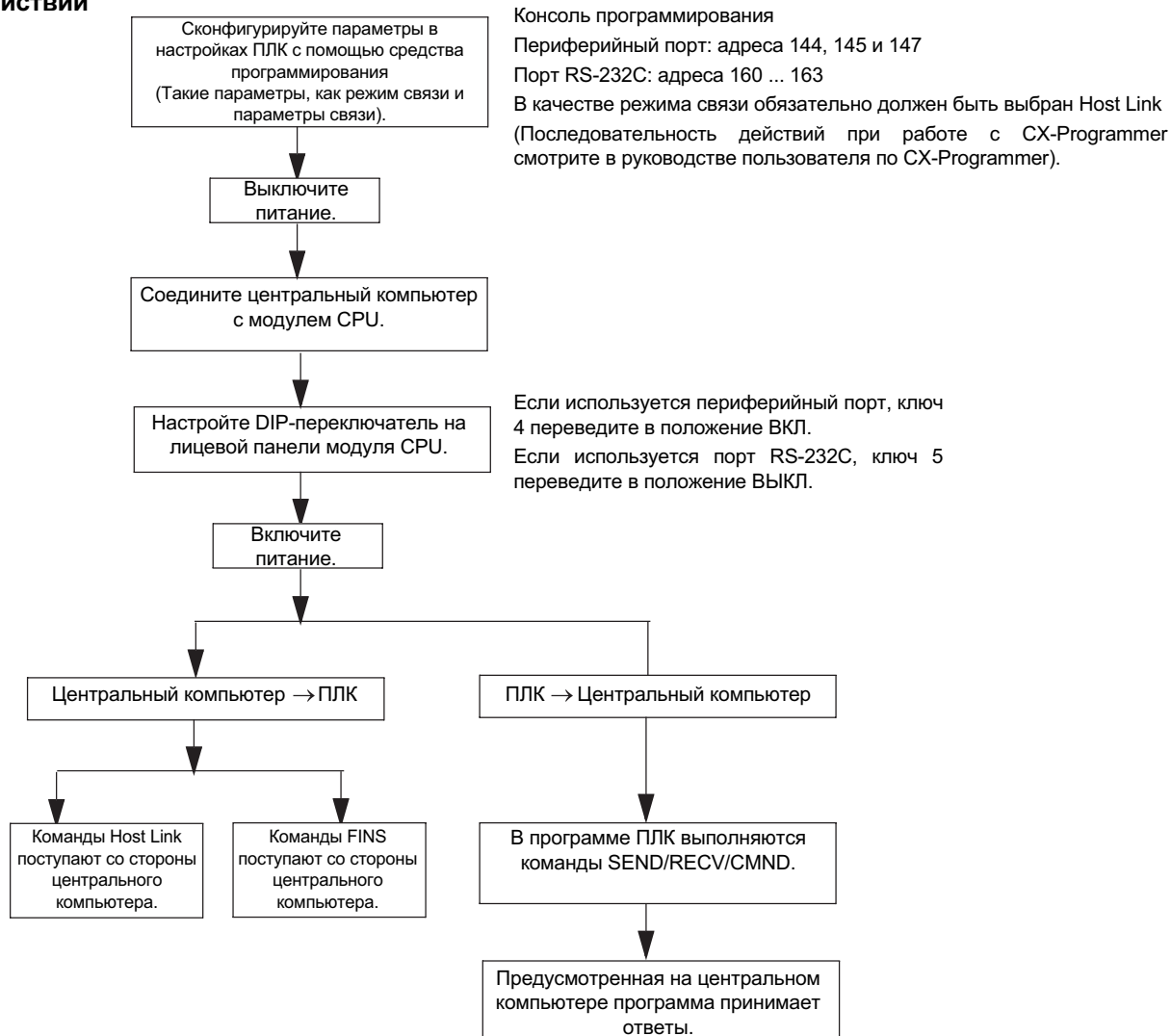
Коммуникационные функции Host Link, предусмотренные в ПЛК CS/CJ, перечислены в следующей таблице. Выберите наиболее подходящий для вас метод.

Направление команды	Тип команды	Метод организации связи	Конфигурация
Центральный компьютер 	Команда Host Link 	На центральном компьютере формируется кадр и на ПЛК отправляется команда. От ПЛК поступает ответ.  Применение: Этот способ используется в основном тогда, когда связь с ПЛК преимущественно инициируется со стороны центрального компьютера.	Центральный компьютер подключается непосредственно к ПЛК в конфигурации 1:1 или 1:N. 
	Команда FINS <sup>1</sup> (С заголовком и признаком завершения Host Link) 	На центральном компьютере формируется кадр и на ПЛК отправляется команда. От ПЛК поступает ответ.  Применение: Этот способ используется в основном тогда, когда связь с ПЛК преимущественно инициируется со стороны центрального компьютера по сети.	Центральный компьютер подключается непосредственно к ПЛК в конфигурации 1:1 или 1:N. 
ПЛК Центральный компьютер 	Команда FINS <sup>2</sup> (С заголовком и признаком завершения Host Link) 	Кадр формируется и передается с помощью команд SEND/RECV/CMND модуля CPU. От центральной станции поступает ответ.  Применение: Этот способ используется в основном тогда, когда связь с центральным компьютером инициируется со стороны ПЛК, например, с целью передачи информации о состоянии (наличии ошибок и т.п.).	Центральный компьютер подключается непосредственно к ПЛК в конфигурации 1:1. 
			Центральный компьютер обменивается данными с другими ПЛК по сети (Host Link конвертируется в протокол сетевого обмена). 
			Центральный компьютер обменивается данными с другими ПЛК по сети (Host Link конвертируется в протокол сетевого обмена). 

**Примечание** 1. Прежде чем команда FINS может быть передана центральным компьютером, к ней должны быть добавлены заголовок и признак завершения, предусмотренные в Host Link.

2. Команда FINS, передаваемая из ПЛК, содержит заголовок и признак завершения Host Link. В центральном компьютере должна быть предусмотрена программа, анализирующая команды FINS и возвращающая соответствующие ответы.

### Последовательность действий



### Команды Host Link

В следующей таблице перечислены команды Host Link. Более подробно они описаны в системном руководстве *C-series Host Link Units System Manual (W143)*.

Код заголовка	Название	Назначение
RR	CIO AREA READ	Чтение содержимого указанного количества слов области CIO, начиная с указанного слова.
RL	LINK AREA READ	Чтение содержимого указанного количества слов области связей (Link), начиная с указанного слова.
RH	HR AREA READ	Чтение содержимого указанного количества слов области удержания (Holding), начиная с указанного слова.
RC	PV READ	Чтение содержимого указанного количества текущих значений (PV) таймеров/счетчиков, начиная с указанного таймера/счетчика.
RG	T/C STATUS READ	Чтение статуса флагов завершения для указанного количества таймеров/счетчиков, начиная с указанного таймера/счетчика.
RD	DM AREA READ	Чтение содержимого указанного количества слов области DM, начиная с указанного слова.
RJ	AR AREA READ	Чтение содержимого указанного количества слов дополнительной области (Auxiliary), начиная с указанного слова.

Код заголовка	Название	Назначение
RE	EM AREA READ	Чтение содержимого указанного количества слов области EM, начиная с указанного слова.
WR	CIO AREA WRITE	Запись указанных данных (только в формате слов) в область CIO, начиная с указанного слова.
WL	LINK AREA WRITE	Запись указанных данных (только в формате слов) в область связей (Link), начиная с указанного слова.
WH	HR AREA WRITE	Запись указанных данных (только в формате слов) в область удержания (Holding), начиная с указанного слова.
WC	PV WRITE	Запись значений PV (текущих значений) для указанного количества таймеров/счетчиков, начиная с указанного таймера/счетчика.
WD	DM AREA WRITE	Запись указанных данных (только в формате слов) в область DM, начиная с указанного слова.
WJ	AR AREA WRITE	Запись указанных данных (только в формате слов) в дополнительную область (Auxiliary), начиная с указанного слова.
WE	EM AREA WRITE	Запись указанных данных (только в формате слов) в область EM, начиная с указанного слова.
R#	SV READ 1	Чтение 4-разрядной BCD-константы или адреса слова в регистре SV для указанной команды таймера/счетчика.
R\$	SV READ 2	Поиск указанной команды таймера/счетчика, начиная с указанного адреса программы, и чтение 4-разрядной константы или адреса слова в регистре SV.
R%	SV READ 3	Поиск указанной команды таймера/счетчика, начиная с указанного адреса программы, и чтение 4-разрядной BCD-константы или адреса слова в регистре SV.
W#	SV CHANGE 1	Изменение 4-разрядной BCD-константы или адреса слова в регистре SV указанной команды таймера/счетчика.
W\$	SV CHANGE 2	Поиск указанной команды таймера/счетчика, начиная с указанного адреса программы, и изменение 4-разрядной константы или адреса слова в регистре SV.
W%	SV CHANGE 3	Поиск указанной команды таймера/счетчика, начиная с указанного адреса программы, и изменение 4-разрядной константы или адреса слова в регистре SV.
MS	STATUS READ	Чтение рабочего статуса модуля CPU (режим работы, принудительно установленный/сброшенный статус, фатальная ошибка).
SC	STATUS CHANGE	Изменение режима работы модуля CPU.
MF	ERROR READ	Чтение и обнуление ошибок модуля CPU (нефатальных и фатальных).
KS	FORCE SET	Принудительная установка указанного бита.
KR	FORCE RESET	Принудительный сброс указанного бита.
FK	MULTIPLE FORCE SET/RESET	Принудительная установка, принудительный сброс или обнуление принудительного состояния для указанных битов.
KC	FORCE SET/RESET CANCEL	Отмена принудительного состояния для всех принудительно установленных и сброшенных битов.
MM	PLC MODEL READ	Чтение типа модели ПЛК.
TS	TEST	Возвращение без каких-либо изменений одного блока данных, переданного со стороны центрального компьютера.
RP	PROGRAM READ	Чтение содержимого области программы пользователя модуля CPU на языке машинных кодов (в объектном коде).
WP	PROGRAM WRITE	Запись программы на языке машинных кодов (в объектном коде), переданной со стороны центрального компьютера, в область программы пользователя модуля CPU.
MI	I/O TABLE GENERATE	Создание зарегистрированной таблицы ввода/вывода, содержащей таблицу физических входов/выходов.
QQMR	COMPOUND COMMAND	Регистрация требуемых битов и слов в таблице.
QQIR	COMPOUND READ	Чтение зарегистрированных битов и слов из памяти ввода/вывода.
XZ	ABORT (только команда)	Прерывание команды Host Link, выполняемой в настоящий момент.

Код заголовка	Название	Назначение
**	INITIALIZE (только команда)	Инициализация процедуры управления передачей для всех ПЛК, подключенных к центральной станции.
IC	Неопределенная команда (только ответ)	Этот ответ возвращается, если код заголовка команды не был распознан.

**Команды FINS**

В следующей таблице перечислены команды FINS. Более подробно они описаны в справочном руководстве *FINS Commands Reference Manual (W227)*.

Тип	Код команды		Название	Функция
I/O Memory Area Access (Обращение к области памяти ввода/вывода)	01	01	MEMORY AREA READ	Последовательное чтение данных из области памяти ввода/вывода.
	01	02	MEMORY AREA WRITE	Последовательная запись данных в область памяти ввода/вывода.
	01	03	MEMORY AREA FILL	Запись одних и тех же данных в указанный диапазон памяти ввода/вывода.
	01	04	MULTIPLE MEMORY AREA READ	Произвольное чтение данных из области памяти ввода/вывода.
	01	05	MEMORY AREA TRANSFER	Последовательное копирование и перенос данных из одной части области памяти ввода/вывода в другую.
Parameter Area Access (Обращение к области параметров)	02	01	PARAMETER AREA READ	Последовательное чтение данных из области параметров.
	02	02	PARAMETER AREA WRITE	Последовательная запись данных в область параметров.
	02	03	PARAMETER AREA FILL	Запись одних и тех же данных в указанный диапазон области параметров.
Program Area Access (Обращение к области программы)	03	06	PROGRAM AREA READ	Чтение данных из области программы пользователя.
	03	07	PROGRAM AREA WRITE	Запись данных в область программы пользователя.
	03	08	PROGRAM AREA CLEAR	Очистка указанного диапазона области программы пользователя.
Execution Control (Управление выполнением)	04	01	RUN	Переключение модуля CPU в режим RUN, MONITOR или DEBUG.
	04	02	STOP	Переключение модуля CPU в режим PROGRAM.
Configuration Read (Чтение конфигурации)	05	01	CONTROLLER DATA READ	Чтение информации о модуле CPU.
	05	02	CONNECTION DATA READ	Чтение номеров моделей указанных модулей.
Status Read (Чтение состояния)	06	01	CONTROLLER STATUS READ	Чтение информации о состоянии модуля CPU.
	06	20	CYCLE TIME READ	Чтение среднего, максимального и минимального значений длительности цикла.
Clock Access (Обращение к часам)	07	01	CLOCK READ	Чтение показаний часов.
	07	02	CLOCK WRITE	Установка показаний часов.
Message Access (Обращение к сообщениям)	09	20	MESSAGE READ/CLEAR	Чтение/очистка сообщений и сообщений FAL(S).
Access Right (Право доступа)	0C	01	ACCESS RIGHT ACQUIRE	Получение прав доступа, если они не принадлежат другому устройству.
	0C	02	ACCESS RIGHT FORCED ACQUIRE	Получение прав доступа, даже если они принадлежат другому устройству.
	0C	03	ACCESS RIGHT RELEASE	Освобождение прав доступа, независимо от того, какому устройству они принадлежат.
Error Access (Ошибка доступа)	21	01	ERROR CLEAR	Очистка ошибок и сообщений об ошибках.
	21	02	ERROR LOG READ	Чтение протокола ошибок.
	21	03	ERROR LOG CLEAR	Обнуление указателя протокола ошибок.

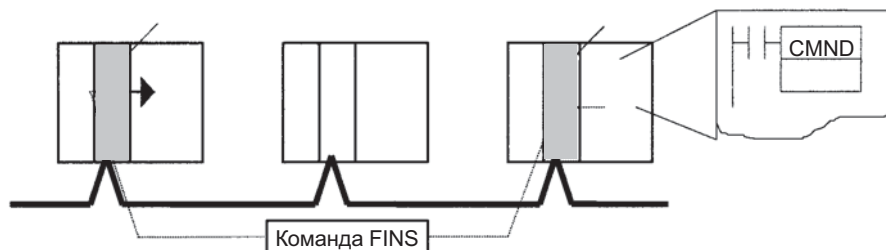
Тип	Код команды		Название	Функция
File Memory (Память файлов)	22	01	FILE NAME READ	Чтение информации о файле в памяти файлов.
	22	02	SINGLE FILE READ	Чтение указанного объема данных, начиная с указанного места в файле.
	22	03	SINGLE FILE WRITE	Запись указанного объема данных, начиная с указанного места в файле.
	22	04	FILE MEMORY FORMAT	Форматирование памяти файлов.
	22	05	FILE DELETE	Удаление указанных файлов из памяти файлов.
	22	07	FILE COPY	Копирование файлов в пределах памяти файлов или между двумя устройствами системы, используемыми в качестве памяти файлов.
	22	08	FILE NAME CHANGE	Изменение имени файла.
	22	0A	I/O MEMORY AREA FILE TRANSFER	Перенос или сравнение данных области памяти ввода/вывода и памяти файлов.
	22	0B	PARAMETER AREA FILE TRANSFER	Перенос или сравнение данных области параметров и памяти файлов.
	22	0C	PROGRAM AREA FILE TRANSFER	Перенос или сравнение данных области программы и памяти файлов.
	22	15	CREATE/DELETE DIRECTORY	Создание или удаление директория.
Forced Status (Принудительные состояния)	23	01	FORCED SET/RESET	Принудительная установка, принудительный сброс или отмена принудительных состояний для указанных битов.
	23	02	FORCED SET/RESET CANCEL	Отмена принудительного состояния для всех принудительно установленных и сброшенных битов.

### Функции протокола обмена сообщениями (FINS)

Перечисленные выше команды FINS также могут поступать на модуль CPU от других ПЛК по сети. При передаче команд FINS по сети необходимо учитывать следующие замечания.

- Если требуется передавать команды FINS, в локальный ПЛК и в адресуемый ПЛК должен быть вставлен модуль шины CPU (например, модуль Controller Link или модуль Ethernet).
- Команды FINS выставляются из программы модуля CPU командой CMND (490).
- Возможен межсетевой обмен командами FINS (до 8-ми сетей для модуля CPU серии CS/CJ версии 2.0 и до 3-х сетей - для других модулей CPU). Сети могут быть как одного типа, так и различных типов.

Модуль последовательного интерфейса    Модуль последовательного интерфейса



Более подробно функции протокола обмена сообщениями описаны в соответствующих руководствах по эксплуатации модулей шины CPU.

## 6-3-2 Беспроточольные коммуникации

В следующей таблице перечислены функции связи со свободно программируемым протоколом (беспроточольный обмен данными), поддерживаемые в ПЛК CS/CJ.

Направление передачи	Способ	Макс. объем данных	Формат кадра		Другие функции
			Код начала	Код завершения	
Передача данных (ПЛК → Внешние устройства)	Выполнение команды TXD(236) в программе*	256 байтов	Да: 00 ... FF Нет: нет	Да: 00 ... FF или CR+LF Нет: нет	Время задержки передачи (задержка между выполнением TXD и передачей данных через указанный порт): 0 ... 99,990 мс (шаг: 10 мс)
Прием данных (Внешние устройства → ПЛК)	Выполнение команды RXD(235) в программе	256 байтов			---

**Примечание** В настройках ПЛК (адрес 162) можно выбрать задержку передачи или "задержку в беспроточольном режиме". Этот параметр позволяет сформировать задержку длительностью до 30 секунд между выполнением TXD(236) и передачей данных через указанный порт.

## Последовательность действий



## Форматы кадров сообщений

При передаче данных с помощью TXD(236) между кодом начала и кодом завершения размещают передаваемые данные. С помощью RXD(235) можно принимать кадры, имеющие тот же формат. При передаче с помощью TXD(236) передаются только данные из памяти ввода/вывода, а при приеме с помощью RXD(235) только сами данные записываются в память ввода/вывода.

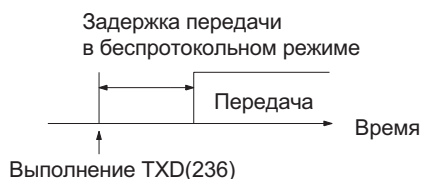
В беспrotocolном режиме может быть передано только до 256 байтов (включая коды начала и завершения).

В следующей таблице показаны форматы сообщений, которые можно выбрать для передачи и приема данных в беспrotocolном режиме. Формат определяется кодом начала (ST) и кодом завершения (ED), выбранными в настройках ПЛК.

Настройка кода начала	Настройка кода завершения		
	Нет	Да	CR+LF
Нет	Данные (данные: макс. 256 байтов)	Данные + ED (данные: макс. 255 байтов)	Данные + CR + LF (данные: макс. 254 байта)
Да	ST + данные (данные: макс. 255 байтов)	ST + данные + ED (данные: макс. 254 байта)	ST + данные + CR + LF (данные: макс. 253 байта)

- Если выбрано несколько кодов начала, используется первый код начала.
- Если выбрано несколько кодов завершения, используется первый код завершения.

- Примечание**
1. Если передаваемые данные содержат код завершения, передача данных будет прервана. В этом случае следует выбрать код завершения CR+LF.
  2. В настройках ПЛК (адрес 162: задержка в беспrotocolном режиме) можно выбрать задержку между выполнением TXD(236) и передачей данных.



Более подробно команды TXD(236) и RXD(235) описаны в справочном руководстве *CJ-series Programmable Controllers Instructions Reference Manual (W340)*.

### 6-3-3 NT Link (Режим 1:N)

ПЛК серии CS/CJ поддерживают связь с программируемыми терминалами (PT) с применением каналов NT Link (Режим 1:N).

- Примечание** Для связи нельзя использовать протокол NT Link в режиме 1:1.

Далее указаны параметры в настройках ПЛК, которые позволяют использовать скоростные каналы NT Link дополнительно к прежним стандартным каналам NT Link (не поддерживаются модулями CPU серии CS1 до версии EV1). При этом скоростные каналы NT Link поддерживаются только программируемыми терминалами NT31(C)-V2 или NT631(C)-V2.

## Настройки ПЛК

Порт связи	Адрес параметра в консоли программирования	Название	Значение параметра	Значение по умолчанию	Прочие условия
Периферийный порт	144 Биты: 8 ... 11	Режим связи по последовательному интерфейсу	02 Hex: NT Link (Режим 1:N)	00 Hex: Host Link	Ключ 4 DIP-переключателя модуля CPU =ВКЛ.
	145 Биты: 0 ... 7	Скорость передачи	00 ... 09 Hex Стандартный канал NT Link A0 Hex: Скоростной NT Link (см. прим. 1)	00 Hex: Стандартный канал NT Link	
	150 Биты: 0 ... 3	Максимальный номер модуля в режиме NT Link	0 ... 7 Hex	0 Hex (Макс. номер модуля 0)	---
Порт RS-232C	160 Биты: 8 ... 11	Режим связи по последовательному интерфейсу	02 Hex: NT Link (Режим 1:N)	00 Hex: Host Link	Ключ 4 DIP-переключателя модуля CPU = Выкл.
	161 Биты: 0 ... 7	Скорость передачи	00 ... 09 Hex Стандартный канал NT Link A0 Hex: Скоростной NT Link (см. прим. 1)	00 Hex: Стандартный канал NT Link	
	166 Биты: 0 ... 3	Максимальный номер модуля в режиме NT Link	0 ... 7 Hex	0 Hex (Макс. номер модуля 0)	---

**Примечание** При настройке параметров с помощью CX-Programmer выберите скорость передачи 115200 бит/с.

## Системное меню программируемого терминала

Выполните следующие настройки в программируемом терминале:

- 1,2,3...** 1. В системном меню программируемого терминала выберите NT Link (1:N) в качестве режима связи порта А или порта В.
2. Нажмите сенсорную кнопку SET, чтобы выбрать режим скоростной связи (Comm. Speed = High Speed).

### 6-3-4 Последовательные каналы связи с ПЛК (только для модулей CPU CJ1M)

#### Обзор

Каналы последовательной связи с ПЛК (Serial PLC Links) поддерживаются только модулями CPU CJ1M. С их помощью возможен обмен данными между модулями CPU CJ1M, снабженными встроенными портами RS-232C, без создания специальных программ. В памяти отводится специальная область слов для Serial PLC Link (CIO 3100 ... CIO 3199). Соединения между модулями CPU можно устанавливать напрямую через порты RS-232C, либо через RS-422A/485-соединения, подключая к портам RS-232C конвертеры интерфейсов RS-232C-RS-422A/485. Для перехода от RS-232C к RS-422A/485 можно использовать конвертеры интерфейсов CJ1W-CIF11 RS-422A.

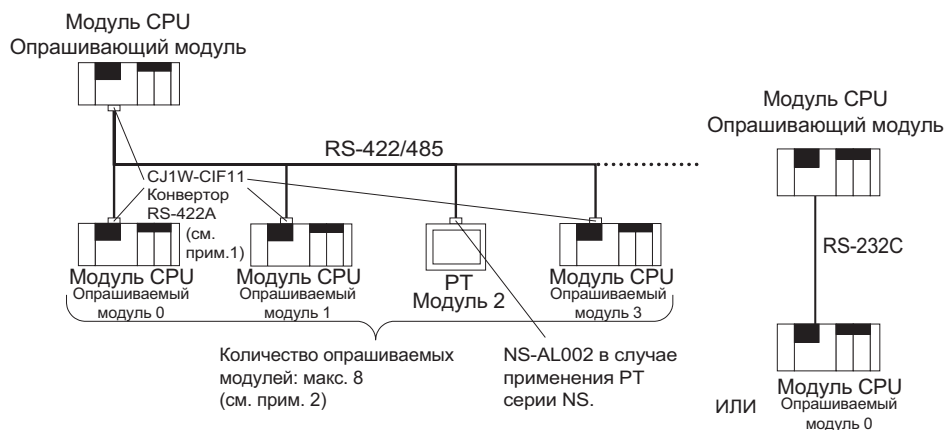
В той же сети также можно использовать программируемый терминал, для которого выбрана связь в режиме NT Link (1:N). Опрашиваемый программируемый терминал связывается с опрашивающим модулем CPU по сети в режиме NT Link (1:N). Однако в случае подключения программируемого терминала по адресам слов, отведенных для каналов Serial PLC Link, соответствующих подключенному программируемому терминалу (определяется номером модуля), содержатся неопределенные значения.



## Параметры связи

Параметр	Характеристика
Способ соединения	RS-232C или RS-422A/485-соединение через порт RS-232C модуля CPU.
Резервируемые области данных	Слова для каналов Serial PLC Link: CIO 3100 ... CIO 3199 (для каждого модуля CPU может быть отведено до 10 слов).
Количество модулей	Макс. 9 модулей, включая 1 опрашивающий модуль и 8 опрашиваемых модулей (в ту же сеть можно подключить программируемый терминал в режиме NT Link (1:N), но он считается как один из 8-ми опрашиваемых модулей).

## Конфигурация системы



- Примечание**
1. CJ1W-CIF11 не обеспечивает гальваническую развязку, поэтому максимальное расстояние составляет 50 м. Если суммарное расстояние связи превышает 50 м, вместо CJ1W-CIF11 следует использовать NT-AL001 с гальванической развязкой. Если используется только NT-AL001, в этом случае суммарное расстояние связи может достигать до 500 м.
  2. Если в сети присутствует программируемый терминал, для которого выбрана связь в режиме Serial PLC Link, в этом случае к опрашивающему модулю можно подключить до 8-ми модулей, включая сам программируемый терминал и остальные опрашиваемые модули.

## Способы обновления данных

Для обновления данных можно использовать один из двух способов.

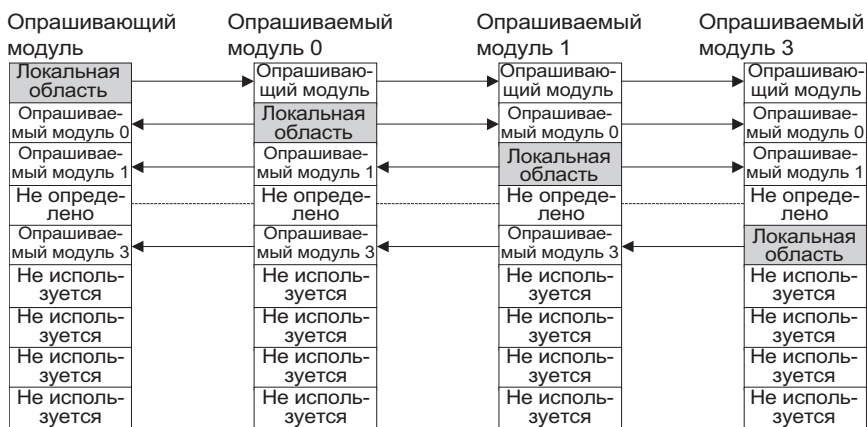
- Режим полного обновления
- Режим обновления на опрашивающем модуле

### Метод полного обновления

Данные всех узлов, участвующих в обмене данными по каналам Serial PLC Link, обновляются и на опрашивающем модуле, и на опрашиваемых модулях (исключение составляет адрес, зарезервированный за номером модуля, соответствующим подключенному в сеть программируемому терминалу, а также адреса отсутствующих в сети опрашиваемых модулей. На всех узлах содержимое этих областей данных не определено).

**Пример: метод полного обновления, наивысший номер модуля: 3**

Ниже приведена диаграмма обновления данных для случая, когда опрашиваемый модуль 2 либо является программируемым терминалом, либо отсутствует в сети, поэтому области, зарезервированные за опрашиваемым модулем 2, во всех узлах находятся в неопределенном состоянии.

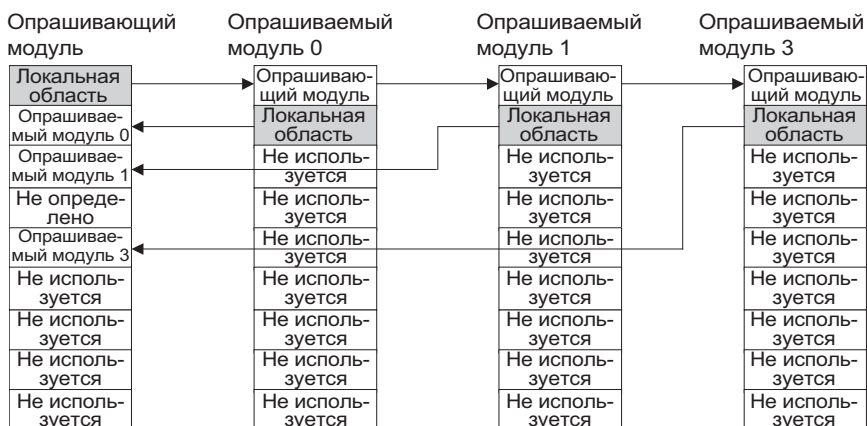


**Метод обновления на опрашивающем модуле**

Данные всех опрашиваемых модулей, подключенных через каналы Serial PLC Link, обновляются только на опрашивающем модуле, при этом данные каждого опрашиваемого модуля соответствуют данным опрашивающего модуля. Преимущество метода обновления только на опрашивающем модуле состоит в том, что адреса, зарезервированные за локальными данными опрашиваемого модуля, совпадают в каждом опрашиваемом модуле, что позволяет использовать одну и ту же программу (лестничную диаграмму) для обращения к данным любого опрашиваемого модуля. Области, зарезервированные за номерами модулей, являющихся программируемыми терминалами или отсутствующими в сети, находятся в неопределенном состоянии только на опрашивающем модуле.

**Пример: метод обновления данных на опрашивающем модуле, наивысший номер модуля: 3**

Ниже приведена диаграмма обновления данных для случая, когда опрашиваемый модуль 2 является программируемым терминалом, либо отсутствует в сети, в результате чего соответствующая область на опрашивающем модуле находится в неопределенном состоянии.



**Резервируемые слова****Метод полного обновления**

Адрес	Слова каналов	1 слово	2 слова	3 слова	...	10 слов	
CIO 3100	Слова каналов Serial PLC Link	Опрашивающий модуль	CIO 3100	CIO 3100 ... CIO 3101	CIO 3100 ... CIO 3102		CIO 3100 ... CIO 3109
		Опрашиваемый модуль 0	CIO 3101	CIO 3102 ... CIO 3103	CIO 3103 ... CIO 3105		CIO 3110 ... CIO 3119
		Опрашиваемый модуль 1	CIO 3102	CIO 3104 ... CIO 3105	CIO 3106 ... CIO 3108		CIO 3120 ... CIO 3129
		Опрашиваемый модуль 2	CIO 3103	CIO 3106 ... CIO 3107	CIO 3109 ... CIO 3111		CIO 3130 ... CIO 3139
		Опрашиваемый модуль 3	CIO 3104	CIO 3108 ... CIO 3109	CIO 3112 ... CIO 3114		CIO 3140 ... CIO 3149
		Опрашиваемый модуль 4	CIO 3105	CIO 3110 ... CIO 3111	CIO 3115 ... CIO 3117		CIO 3150 ... CIO 3159
		Опрашиваемый модуль 5	CIO 3106	CIO 3112 ... CIO 3113	CIO 3118 ... CIO 3120		CIO 3160 ... CIO 3169
		Опрашиваемый модуль 6	CIO 3107	CIO 3114 ... CIO 3115	CIO 3121 ... CIO 3123		CIO 3170 ... CIO 3179
		Опрашиваемый модуль 7	CIO 3108	CIO 3116 ... CIO 3117	CIO 3124 ... CIO 3126		CIO 3180 ... CIO 3189
CIO 3199	Не используется	CIO 3109 ... CIO 3199	CIO 3118 ... CIO 3199	CIO 3127 ... CIO 3199		CIO 3190 ... CIO 3199	

**Метод обновления на опрашивающем модуле**

Адрес	Слова каналов	1 слово	2 слова	3 слова	...	10 слов	
CIO 3100	Слова каналов Serial PLC Link	Опрашивающий модуль	CIO 3100	CIO 3100 ... CIO 3101	CIO 3100 ... CIO 3102		CIO 3100 ... CIO 3109
		Опрашиваемый модуль 0	CIO 3101	CIO 3102 ... CIO 3103	CIO 3103 ... CIO 3105		CIO 3110 ... CIO 3119
		Опрашиваемый модуль 1	CIO 3101	CIO 3102 ... CIO 3103	CIO 3103 ... CIO 3105		CIO 3110 ... CIO 3119
		Опрашиваемый модуль 2	CIO 3101	CIO 3102 ... CIO 3103	CIO 3103 ... CIO 3105		CIO 3110 ... CIO 3119
		Опрашиваемый модуль 3	CIO 3101	CIO 3102 ... CIO 3103	CIO 3103 ... CIO 3105		CIO 3110 ... CIO 3119
		Опрашиваемый модуль 4	CIO 3101	CIO 3102 ... CIO 3103	CIO 3103 ... CIO 3105		CIO 3110 ... CIO 3119
		Опрашиваемый модуль 5	CIO 3101	CIO 3102 ... CIO 3103	CIO 3103 ... CIO 3105		CIO 3110 ... CIO 3119
		Опрашиваемый модуль 6	CIO 3101	CIO 3102 ... CIO 3103	CIO 3103 ... CIO 3105		CIO 3110 ... CIO 3119
		Опрашиваемый модуль 7	CIO 3101	CIO 3102 ... CIO 3103	CIO 3103 ... CIO 3105		CIO 3110 ... CIO 3119
CIO 3199	Не используется	CIO 3102 ... CIO 3199	CIO 3104 ... CIO 3199	CIO 3106 ... CIO 3199		CIO 3120 ... CIO 3199	

**Последовательность действий**

Работа последовательных каналов Serial PLC Link определяется настройкой следующих параметров в настройках ПЛК.

**Настройка параметров опрашивающего модуля**

- 1,2,3...**
1. Выберите для порта связи RS-232C в качестве режима связи обмен через последовательные каналы Serial PLC Link (режим опрашивающего модуля).
  2. Выберите метод связи: метод полного обновления или метод обновления на опрашивающем модуле.
  3. Выберите количество слов каналов (до 10 слов для каждого модуля).
  4. Укажите максимальный номер модуля для каналов Serial PLC Link (0 ... 7).

**Настройка параметров опрашиваемых модулей**

- 1,2,3...**
1. Выберите для порта связи RS-232C в качестве режима связи обмен через последовательные каналы Serial PLC Link (режим опрашиваемого модуля).
  2. Укажите номер модуля для опрашиваемого модуля, участвующего в обмене данными через Serial PLC Link.

**Настройки в ПЛК****Настройка параметров опрашивающего модуля**

Параметр		Адрес ПЛК		Устанавливаемые значения	По умолчанию	Режим обновления
		Слово	Бит			
Параметры порта RS-232C	Режим последовательной связи	160	11 ... 08	8 hex: Каналы последовательной связи Serial PLC Link Опрашивающий модуль	0 hex	Обновление в каждом цикле
	Скорость передачи данных	161	07 ... 00	00 ... 09 hex: Стандартная 0A hex: Повышенная (см. прим. 2)	00 hex	
	Метод обновления данных	166	15	0: Полное обновление 1: Обновление на опрашивающем модуле	0	
	Количество слов на канал		07 ... 04	1 ... A hex	0 hex (см. прим. 1)	
	Максимальный номер модуля		03 ... 00	0 ... 7 hex	0 hex	

- Примечание**
1. Если используется принимаемое по умолчанию значение параметра 0 hex, автоматически резервируется 10 слов (A hex).
  2. Если выбран CX-Programmer, укажите скорость 115200 бит/с.

**Настройка параметров опрашиваемых модулей**

Параметр		Адрес ПЛК		Устанавливаемые значения	По умолчанию	Режим обновления
		Слово	Бит			
Параметры порта RS-232C	Режим последовательной связи	160	11 ... 08	7 hex: Каналы последовательной связи Serial PLC Link Опрашиваемый модуль	0 hex	Обновление в каждом цикле
	Скорость передачи данных	161	07 ... 00	00 ... 09 hex: Стандартная 0A hex: Повышенная (см. прим.)	00 hex	
	Номер опрашиваемого модуля	167	03 ... 00	0 ... 7 hex	0 hex	

- Примечание** Если выбран CX-Programmer, укажите скорость 115200 бит/с.

## Сопутствующие флаги дополнительной области

Название	Адрес	Описание	Чтение/ запись	Режим обновления
Флаг "Ошибка порта связи RS-232C"	A39204	Этот бит включается, когда происходит ошибка связи порта RS-232C. 1: Ошибка 0: Ошибок нет	Чтение	<ul style="list-style-type: none"> <li>Обнуляется по включению питания.</li> <li>Устанавливается, если происходит ошибка связи порта RS-232C.</li> <li>Сбрасывается, когда порт перезапускается.</li> <li>Не работает в режиме периферийной шины и в режиме NT Link.</li> </ul>
Флаг "Порт RS-232C обменивается данными с РТ" (см. примечание)	A39300 ... A39307	Если порт RS-232C используется в режиме NT Link, и с определенным модулем осуществляется связь, в этом слове устанавливается соответствующий бит. Номерам модулей 0 ... 7 соответствуют биты 00 ... 07. 1: Обмен данными 0: Нет обмена данными	Чтение	<ul style="list-style-type: none"> <li>Сбрасывается по включению питания.</li> <li>Устанавливается бит, соответствующий номеру модуля РТ/опрашиваемого модуля, с которым осуществляется связь через порт RS-232C в режиме NT Link или в режиме Serial PLC Link.</li> <li>Номерам модулей 0 ... 7 соответствуют биты 00 ... 07.</li> </ul>
Бит "Сброс порта RS-232C"	A52600	Чтобы перезапустить порт RS-232C, установите этот бит.	Чтение/ запись	<ul style="list-style-type: none"> <li>Сбрасывается по включению питания.</li> <li>Устанавливается в случае перезапуска порта RS-232C (за исключением связи в режиме периферийной шины).</li> </ul> <p>Примечание: в зависимости от системы бит может автоматически сбрасываться по завершению процедуры перезапуска.</p>
Флаг "Ошибка порта RS-232C"	A52800 ... A52807	В случае возникновения ошибки порта RS-232C в это слово записывается соответствующий код ошибки. Бит 00: Не используется. Бит 01: Не используется. Бит 02: Ошибка четности. Бит 03: Ошибка кадра. Бит 04: Ошибка переполнения. Бит 05: Ошибка превышения времени. Бит 06: Не используется. Бит 07: Не используется.	Чтение/ запись	<ul style="list-style-type: none"> <li>Обнуляется по включению питания.</li> <li>В случае возникновения ошибки порта RS-232C записывается соответствующий код ошибки.</li> <li>В зависимости от системы флаг может обнулиться после перезапуска порта RS-232C.</li> <li>Не работает в режиме периферийной шины.</li> <li>В режиме NT Link действует только бит 05 (Ошибка превышения времени).</li> </ul> <p>В режиме Serial PLC Link работают только следующие флаги. Ошибка опрашиваемого модуля: Бит 05: Ошибка превышения времени</p> <ul style="list-style-type: none"> <li>Ошибка CHECK опрашиваемого модуля: Бит 03: Ошибка кадра. Бит 04: Ошибка переполнения. Бит 05: Ошибка превышения времени.</li> </ul>
Флаг "Параметры порта RS-232C изменены"	A61902	Данный флаг устанавливается, если произошло изменение параметров связи порта RS-232C. 1: Параметры изменились 0: Параметры не изменились	Чтение/ запись	<ul style="list-style-type: none"> <li>Сбрасывается по включению питания.</li> <li>Находится в состоянии ВКЛ, пока производится изменение параметров связи для порта RS-232C.</li> <li>Включается, когда выполняется команда CHANGE SERIAL PORT SETUP (STUP(237)).</li> <li>Вновь сбрасывается по завершению изменения параметров.</li> </ul>

**Примечание** Точно так же, как и для существующего NT Link (1:N), состояние (связь/нет связи) программируемых терминалов, участвующих в обмене через Serial PLC Link, можно проверить на стороне опрашиваемого модуля (модуль CPU), выполнив чтение флага "Порт RS-232C обменивается данными с РТ" (A393, биты 00 ... 07 для номеров модулей 0 ... 7).

## **6-4    Изменение режима обновления текущего значения (PV) таймеров/счетчиков**

### **6-4-1    Обзор**

Ранее в модулях CPU CS1 для обновления текущих значений (PV) таймеров/счетчиков использовался только формат BCD. Поэтому все параметры таймеров/счетчиков вводились в формате BCD. В остальных модулях CPU (см. примечания 1 и 2) для обновления текущих значений таймеров и счетчиков можно использовать как BCD, так и двоичный формат (см. примечание 3).

Если используется двоичный формат, задание (время) для таймеров/счетчиков можно вводить в диапазоне 0 ... 65535, в отличие от прежнего диапазона 0 ... 9999. В качестве заданий таймеров/счетчиков также можно назначать двоичные значения, рассчитанные другими командами. Режим обновления (формат) значений PV таймеров/счетчиков также можно указать, если вместо задания таймера/счетчика указывается адрес (косвенная адресация). (Выбор режима (BCD или двоичный) определяет, в каком формате представлено содержимое адресуемого слова (BCD или двоичное значение)).

Следует заметить, что между операндами команд в режиме BCD и в двоичном режиме имеются различия, поэтому, прежде чем изменять режим обновления (формат) PV таймеров/счетчиков, необходимо разобраться и понять отличия между этими режимами.

**Примечание**

1. Модули CPU, поддерживающие двоичный формат:
  - Модули CPU CS1-H
  - Модули CPU CJ1-H
  - Модули CPU CJ1M
  - Модули CPU CS1D
2. Для модулей CPU CS1-H, CJ1-H, CJ1M или CS1D, произведенных до 31 мая 2002г. включительно, для которых выбран двоичный формат обновления текущих значений таймеров/счетчиков, мнемонические коды для двоичного формата отображаются на консоли программирования либо как мнемонические коды, либо как BCD-команды (пример: TIMX #0000 &16 отображается как TIM #0000 &16), но операции выполняются с двоичными значениями.
3. Режим обновления (формат) PV можно выбрать только с помощью CX-Programmer версии 3.0. Выбор режима (формата) не поддерживается CX-Programmer версии 2.1 или более старой версии, а также не поддерживается консолями программирования.
4. С помощью CX-Programmer версии 2.1 или более старой версии нельзя считывать программы пользователя для модулей CPU, содержащих команды двоичного формата, однако можно читать программы, в которых используются команды BCD-формата.

### 6-4-2 Функциональные характеристики

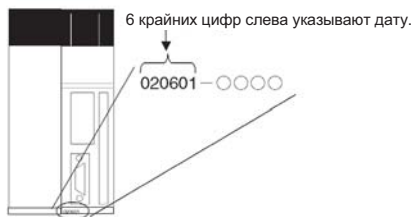
Параметр	Описание		
Способ настройки режима обновления (формата) значений PV таймеров/счетчиков	Должен настраиваться с помощью CX-Programmer версии 3.0 (не поддерживается CX-Programmer версии 2.1 или более старой). Настраивается в свойствах ПЛК в CX-Programmer версии 3.0.		
Поддерживаемые модули CPU	Модули CPU CS1-H/CJ1-H с номером партии 020601 (изготовлены 1 июня 2002г.) или более поздние (см. прим. 1), а также модули CPU CJ1M и CS1D.		
Режим	<b>Режим BCD</b>	<b>Двоичный режим</b>	
Мнемонический код	Как и у предыдущих моделей Пример: TIM	К мнемоническому коду в BCD-режиме добавляется символ X Пример: TIMX	
Код функции	Как и у предыдущих моделей	Новые коды	
Диапазон значений PV/SV	#0000 ... #9999	&0 ... &65536	#0000 ... #FFFF
Отображение PV на средстве программирования (CX-Programmer версии 3.0 или консоль программирования)	BCD Пример: #0100	Десятичный формат Пример: &100	Шестнадцатеричный формат Пример: #64

**Примечание** Для модулей CPU CS1-H/CJ1-H, произведенных до 31 мая 2002г. включительно, для которых выбран двоичный формат обновления текущих значений таймеров/счетчиков, мнемонические коды для двоичного формата отображаются на консоли программирования либо как мнемонические коды, либо как BCD-команды (пример: TIMX #0000 &16 отображается как TIM #0000 &16), но операции выполняются с двоичными значениями.

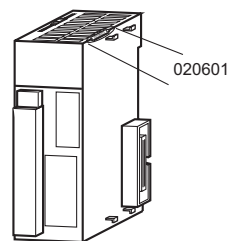
#### Определение номера партии модуля CPU

- 1,2,3...** 1. Номер партии располагается снизу на лицевой панели (серия CS) или в правом верхнем углу модуля (серия CJ) и состоит из двух последних цифр года, месяца и дня, следующих в порядке, указанном на рисунке ниже.  
Пример: 020601 (изготовлен 1 июня 2002 г.)

Модуль CPU серии CS



Модуль CPU серии CJ



2. Проверьте, какой режим выбран, установив с модулем связь в CX-Programmer, открыв окно I/O Table (Таблица входов/выходов) и выбрав **Unit Information - CPU Unit (Информация о модуле - Модуль CPU)**. Номер партии (lot No.) будет отображен в том же формате, который указан на рисунке выше, то есть, будет состоять из двух последних цифр года, месяца и дня, именно в такой последовательности.

### 6-4-3 Выбор и подтверждение режима BCD/двоичного режима

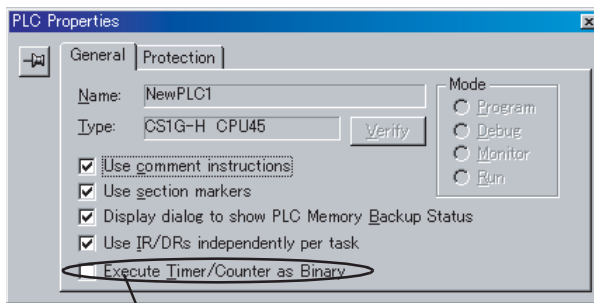
При создании новой программы режим (BCD/двоичный формат) выбирается в настройках ПЛК в CX-Programmer версии 3.0.

**Примечание** Выбор BCD/двоичного режима поддерживается только CX-Programmer версии 3.0 или выше. CX-Programmer версии 2.1 или более старой версии не поддерживает выбор режима.



#### Выбор BCD/двоичного режима

- 1,2,3... 1. Выберите название ПЛК, щелкните правой кнопкой мыши и выберите **PLC Properties** (Свойства ПЛК).



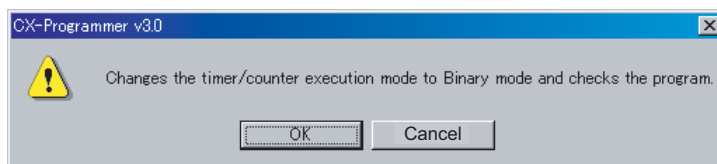
Установите флажок, чтобы задействовать параметр.

2. Щелкните по вкладке **General** (Общие свойства) и выберите **Execute Timers/Counters as Binary** (Применять для таймеров/счетчиков двоичный формат).
  - Не выбрано (по умолчанию): BCD-режим
  - Выбрано: двоичный режим

При загрузке программы пользователя из CX-Programmer в модуль CPU значение параметра, отвечающего за режим обновления PV, установленное в свойствах ПЛК, будет записано в память пользователя модуля CPU.

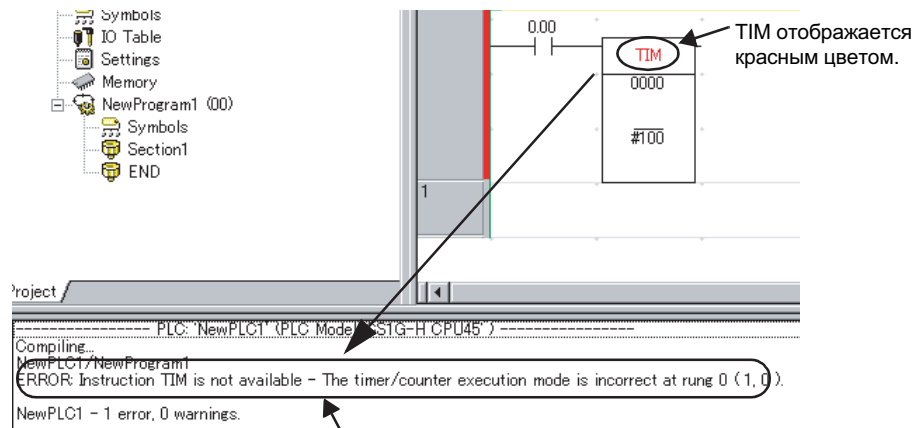


При изменении параметра автоматически отображается следующее диалоговое окно.



Чтобы выполнить проверку программы, щелкните по кнопке **ОК**. Результаты проверки программы будут отображены в окне вывода.

Пример: хотя была произведена замена на двоичный режим обновления, в программе была применена команда TIM.



Результаты проверки программы отображаются в окне вывода. Пример: другой режим работы таймера/счетчика, поэтому TIM использовать нельзя.

### **Подтверждение BCD/двоичного режима**

Чтобы проверить, в каком режиме работает модуль CPU, в BCD или в двоичном режиме, можно использовать бит A09915 в дополнительной области (флаг "Режим обновления (формат) значения PV таймеров/счетчиков").

Название	Адрес	Описание
Флаг "Режим обновления (формат) значения PV таймеров/счетчиков"	A09915	0: Режим BCD 1: Двоичный режим

### **6-4-4 Мнемонические символы и данные в режиме BCD/двоичном режиме**

#### **Мнемонические коды для BCD/двоичного режимов**

Мнемонические коды в двоичном режиме дополняются символом X (добавляется в конец BCD-мнемоники).

Пример: мнемоника для команды TIMER

BCD-режим: TIM

Двоичный режим: TIMX

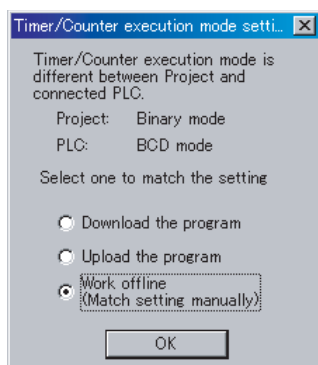
### Отображение данных в BCD/двоичном режиме

Выбранный режим (параметр ПЛК)	Значение вводимых и отображаемых символов	Диапазон значений	Пример: номер таймера: 0000, заданное значение:
BCD-режим	Значение, используемое в команде, обозначается символом # (BCD-значение, если выбран режим BCD)	#0000 ... #9999 или #00000000 ... #99999999	
Двоичный режим	Десятичное значение обозначается символом &	&0 ... &65535 или &0 ... &4294967295	
	Используемые в команде значения обозначаются символом # (16-ричное значение, если выбран режим BCD)	#0000 ... #FFFF или #0000 ... #FFFFFFFF	

**Примечание** Если используется CX-Programmer и выбран BCD-режим или двоичный режим, то в случае ввода числового значения без символа ввода/отображения # или &, обозначающего константу (например, TIM 0000 0010), заданное значение таймера/счетчика будет воспринято как адрес (например, в качестве заданного значения будет использоваться значение, содержащееся в слове 0010 области CIO).

### 6-4-5 Ограничения

- В модуле CPU нельзя одновременно применять режим BCD и двоичный режим.
- Если для создания новой программы пользователя или для обнуления памяти используется консоль программирования, в этом случае в качестве режима обновления значений PV таймеров/счетчиков всегда используется BCD-режим.
- Если связь с модулем CPU устанавливается из CX-Programmer версии 3.0, в этом случае автоматически будет использоваться значение параметра, отвечающего за режим обновления значений PV таймеров/счетчиков, хранящееся в памяти пользователя модуля CPU. Если значение параметра CPU отличается от значения, выбранного в проекте CX-Programmer, произойдет ошибка и связь установлена не будет. Будет отображено следующее сообщение.



Необходимо выбрать, какой из параметров должен быть изменен, чтобы оба параметра стали одинаковыми, параметр, хранящийся в модуле CPU, либо параметр, содержащийся в проекте CX-Programmer.

- С помощью CX-Programmer версии 2.1 или более старой нельзя считывать программы пользователя из модуля CPU, которые создавались в двоичном режиме. Программы, создававшиеся в BCD-режиме, считывать можно.

- Ниже описано чем отличается CX-Programmer от консоли программирования в случае, когда введена команда с неправильным режимом обновления значений PV таймеров/счетчиков:
  - CX-Programmer:
 

В случае ввода команды не с тем режимом обновления значений PV таймеров/счетчиков, что был выбран в настройках ПЛК, произойдет ошибка. Пример: Если для ПЛК в составе проекта выбран двоичный режим, при вводе мнемонического кода TIM произойдет ошибка. Если выбран BCD-режим, ошибка произойдет в случае ввода мнемонического кода TIMX.
  - Консоль программирования:
 

Если для команды введен код функции, соответствующий режиму обновления значений PV таймеров/счетчиков, отличающемуся от режима, который выбран для модуля CPU, в этом случае мнемонический код будет автоматически заменен на код, соответствующий режиму обновления значений PV таймеров/счетчиков, выбранному в модуле CPU.

## 6-4-6 Команды и операнды

### Команды

Тип команды	Название	Мнемонический код	
		BCD-режим	Двоичный режим
Команды для таймеров и счетчиков	TIMER (100 мс)	TIM	TIMX(550)
	HIGH-SPEED TIMER (10 мс)	TIMH(015)	TIMHX(551)
	ONE-мс TIMER (1 мс)	TMHH(540)	TMHHX(552)
	ACCUMULATIVE TIMER (100 мс)	TTIM(087)	TTIMX(555)
	LONG TIMER (100 мс)	TIML(542)	TIMLX(553)
	MULTI-OUTPUT TIMER (100 мс)	MTIM(543)	MTIMX(554)
	COUNTER	CNT	CNTX(546)
	REVERSIBLE COUNTER	CNTR(012)	CNTRX(548)
	RESET TIMER/COUNTER	CNR(545)	CNRX(547)
Команды для программных блоков	TIMER WAIT (100 мс)	TIMW(813)	TIMWX(816)
	HIGH-SPEED TIMER WAIT (10 мс)	TMHW(815)	TMHWX(817)
	COUNTER WAIT	CNTW(814)	CNTWX(818)

**Команды и операнды**

**Команды для таймеров и счетчиков**

**TIMER (100 мс)**

Название команды	BCD-режим	Двоичный режим
Мнемонический код	TIM	TIMX(550)
S (заданное значение таймера)	#0000 ... #9999 (BCD)	&0 ... &65535 (десятичный) или #0000 ... #FFFF (hex)
Задаваемое время (шаг: 0.1 с)	0 ... 999.9 с	0 ... 6553.5 с

**HIGH-SPEED TIMER (10 мс)**

Название команды	BCD-режим	Двоичный режим
Мнемонический код	TIMH(015)	TIMHX(551)
S (заданное значение таймера)	#0000 ... #9999 (BCD)	&0 ... &65535 (десятичный) или #0000 ... #FFFF (hex)
Задаваемое время (шаг: 0.01 с)	0 ... 99.99 с	0 ... 655.35 с

**ONE-MS TIMER (1 мс)**

Название команды	BCD-режим	Двоичный режим
Мнемонический код	TMH(540)	TIMHX(552)
S (заданное значение таймера)	#0000 ... #9999 (BCD)	&0 ... &65535 (десятичный) или #0000 ... #FFFF (hex)
Задаваемое время (шаг: 0.001 с)	0 ... 9.999 с	0 ... 65.535 с

**ACCUMULATIVE TIMER (100 мс)**

Название команды	BCD-режим	Двоичный режим
Мнемонический код	TTIM(087)	TTIMX(555)
S (заданное значение таймера)	#0000 ... #9999 (BCD)	&0 ... &65535 (десятичный) или #0000 ... #FFFF (hex)
Задаваемое время (шаг: 0.1 с)	0 ... 999.9 с	0 ... 6553.5 с

**LONG TIMER (100 мс)**

Название команды	BCD-режим	Двоичный режим
Мнемонический код	TIML(542)	TIMLX(553)
S, S+1 (заданные значения таймеров)	#00000000 ... #99999999 (BCD)	&0 ... &4294967295 (десятичный) или #0000 ... #FFFFFFFF (hex)
Задаваемое время (шаг: 0.1 с)	0 ... 999.9 с	0 ... 6553.5 с

**MULTI-OUTPUT TIMER (100 мс)**

Название команды	BCD-режим	Двоичный режим
Мнемонический код	MTIM(543)	MTIMX(554)
S ... S-7 (заданные значения для каждого таймера)	#0000 ... #9999 (BCD)	&0 ... &65535 (десятичный) или #0000 ... #FFFF (hex)
Задаваемое время (шаг: 0.1 с)	0 ... 999.9 с	0 ... 6553.5 с

**COUNTER**

Название команды	BCD-режим	Двоичный режим
Мнемонический код	CNT	CNTX(546)
S (заданное значение счетчика)	#0000 ... #9999 (BCD)	&0 ... &65535 (десятичный) или #0000 ... #FFFF (hex)
Задаваемые значения	0 ... 9999 раз	0 ... 65535 раз

**REVERSIBLE COUNTER**

Название команды	BCD-режим	Двоичный режим
Мнемонический код	CNTR(012)	CNTRX(548)
S (заданное значение счетчика)	#0000 ... #9999 (BCD)	&0 ... &65535 (десятичный) или #0000 ... #FFFF (hex)
Задаваемые значения	0 ... 9999 раз	0 ... 65535 раз

**RESET TIMER/COUNTER**

Название команды	BCD-режим	Двоичный режим
Мнемонический код	CNTR(545)	CNTRX(547)

Команды для программных блоков

**TIMER WAIT (100 мс)**

Название команды	BCD-режим	Двоичный режим
Мнемонический код	TIMW(813)	TIMWX(816)
S (заданное значение таймера)	#0000 ... #9999 (BCD)	&0 ... &65535 (десятичный) или #0000 ... #FFFF (hex)
Задаваемое время (шаг: 0.1 с)	0 ... 999.9 с	0 ... 6553.5 с

**HIGH-SPEED TIMER WAIT (10 мс)**

Название команды	BCD-режим	Двоичный режим
Мнемонический код	TMHW(815)	TMHWX(817)
S (заданное значение таймера) Шаг: 0.01 с	#0000 ... #9999 (BCD)	&0 ... &65535 (десятичный) или #0000 ... #FFFF (hex)
Задаваемое время (шаг: 0.01 с)	0 ... 999.9 с	0 ... 655.35 с

**COUNTER WAIT**

Название команды	BCD-режим	Двоичный режим
Мнемонический код	CNTW(814)	CNTWX(818)
S (заданное значение счетчика)	#0000 ... #9999 (BCD)	&0 ... &65535 (десятичный) или #0000 ... #FFFF (hex)
Задаваемые значения	0 ... 9999 раз	0 ... 65535 раз

## 6-5 Применение запланированного прерывания в качестве высокоточного таймера (только для CJ1M)

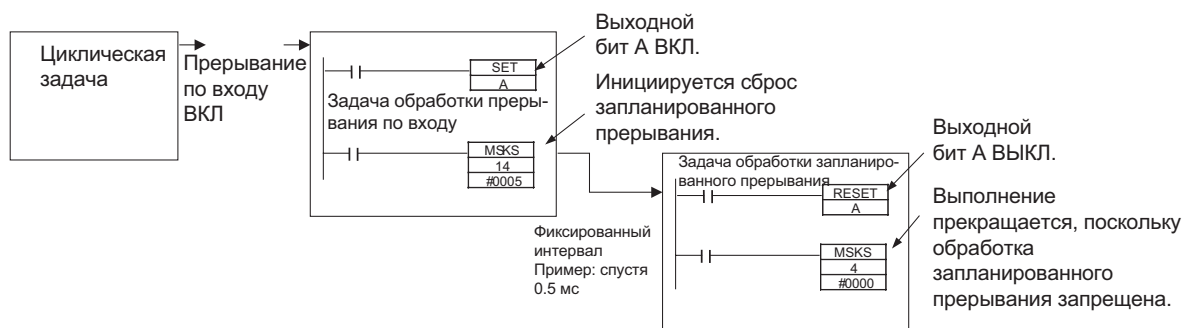
Ниже перечислены функции, позволяющие применять запланированное прерывание для реализации высокоточного таймера в случае использования модуля CPU CJ1M.

- Значение периода формирования запланированного прерывания можно вводить с шагом 0.1 мс (высокоточный таймер интервала).
- Применение команды MSKS(690) для сброса (то есть, перезапуска) (фиксированное время до первого прерывания).
- С помощью команды MSKS(692) можно считывать текущие значения (PV) внутреннего таймера (чтение текущих значений таймера интервала).

Ниже приведен пример применения перечисленных выше функций. Реализуется высокоточный таймер-одновибратор, который запускается по переходу входного бита в состояние ВКЛ. Выходной бит таймера включается на некоторое определенное время, после чего сбрасывается.

Пример:

- 1,2,3... По переходу внутреннего входного бита в состояние ВКЛ запускается задача обработки прерывания по входу.
- Задача обработки прерывания по входу включает выходной бит А, выполняется команда MSKS(690), иницирующая сброс запланированного прерывания.
- Через фиксированный промежуток времени запускается задача обработки запланированного прерывания, выходной бит А в задаче обработки запланированного прерывания сбрасывается, выполняется команда MSKS(690), запрещающая обработку запланированного прерывания.



### 6-5-1 Задание времени для запланированного прерывания с шагом 0.1 мс

Значение времени для запланированного прерывания устанавливается с помощью команды MSKS(690) с шагом, который определяется соответствующим параметром в настройках ПЛК.

В модулях CPU CJ1M время для запланированного прерывания можно задавать с шагом 0.1 мс в пределах от 0.5 мс до 999.9 мс.

#### Настройки ПЛК

Параметр	Адрес в ПЛК		Устанавливаемое значение	По умолчанию	Момент обновления
	Слово	Бит			
Шаг установки времени для запланированного прерывания	195	00 ... 03	0 hex: шаг 10 мс 1 hex: шаг 1 мс 2 hex: шаг 0.1 мс (только для модулей CPU CJ1M)	0 hex	В начале работы.

### 6-5-2 Выбор сброса таймера при запуске с помощью MSKS(690)

В случае применения модулей CPU CJ1M и использования команды MSKS(690) для запуска запланированного прерывания, внутренний таймер можно сбросить перед запуском прерывания (эта операция называется "инициирование сброса" или "сброс при запуске").

Используя эту возможность, можно указать время, которое должно пройти до формирования первого прерывания, не прибегая к команде CLI(691).

Для запуска запланированных прерываний используется команда MSKS(690), с помощью которой устанавливается время запланированного прерывания (интервал между двумя прерываниями). Однако время, которое должно пройти, прежде чем будет запущена первая задача обработки запланированного прерывания (время формирования первого прерывания), после выполнения команды MSKS(690) будет детерминированным только, если указана команда CLI(691). Таким образом, в модулях CPU CJ1M предусмотрен сброс внутреннего таймера при запуске, что позволяет устанавливать время срабатывания первого прерывания без использования команды CLI(691).

#### Операнд команды MSKS(690) (только если выбрано запланированное прерывание)

Операнд	Устанавливаемое значение
N (идентификатор прерывания)	4: Запланированное прерывание 0, обычная настройка (внутренний таймер не сбрасывается)
	5: Запланированное прерывание 1, обычная настройка (внутренний таймер не сбрасывается)
	14: Запланированное прерывание 0, сброс при запуске (только для модулей CPU CJ1M)
	15: Запланированное прерывание 1, сброс при запуске (только для модулей CPU CJ1M)

### 6-5-3 Чтение текущего значения (PV) внутреннего таймера с помощью MSKR(692)

В модулях CPU CJ1M предусмотрена возможность чтения текущего значения (PV) внутреннего таймера, производящего отсчет времени до срабатывания запланированного прерывания. Считывается время, прошедшее либо с момента запуска запланированного прерывания, либо с момента срабатывания предыдущего запланированного прерывания. Чтение значения (PV) внутреннего таймера выполняется путем выполнения команды MSKR(692). Шаг измерения времени зависит от параметра в настройках ПЛК, отвечающего за шаг установки времени для запланированного прерывания, точно так же, как и для времени запланированного прерывания.

#### Операнд команды MSKR(692) (только если выбрано запланированное прерывание)

Операнд	Устанавливаемое значение
N (идентификатор прерывания)	4: Запланированное прерывание 0, чтение времени запланированного прерывания (установленного значения)
	5: Запланированное прерывание 1, чтение времени запланированного прерывания (установленного значения)
	14: Запланированное прерывание 0, чтение значения PV внутреннего таймера (только для модулей CPU CJ1M)
	15: Запланированное прерывание 1, чтение значения PV внутреннего таймера (только для модулей CPU CJ1M)

## 6-6 Настройка параметров запуска и техническое обслуживание

В этом разделе описан ряд функций, связанных с процедурами запуска и технического обслуживания.

- "Горячий" запуск/ "горячий" останов
- Настройка режима запуска
- Настройка задержки обнаружения выключения питания
- Запрет прерываний по отключению питания
- Выход RUN
- Часы
- Защита программы
- Дистанционное программирование и контроль
- Флеш-память
- Настройка условий запуска

### 6-6-1 Функции "горячего" запуска/ "горячего" останова

#### Изменение режима работы

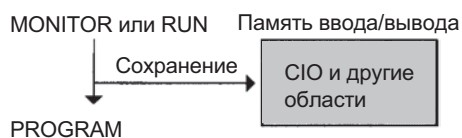
##### "Горячий" запуск

Чтобы все данные\*, содержащиеся в памяти ввода/вывода, сохранялись при переключении модуля CPU из режима PROGRAM в режим RUN/MONITOR (то есть, при запуске программы), следует установить бит удержания состояния IOM (A50012, IOM Hold).



##### "Горячий" останов

Если бит удержания состояния IOM (A50012, IOM Hold) включен, все данные\*, содержащиеся в памяти ввода/вывода, будут сохранены при переключении модуля CPU из режима RUN/MONITOR в режим PROGRAM (то есть, при остановке программы).



**Примечание** \*Если бит удержания состояния IOM (IOM Hold) включен, в этом случае при изменении режима (PROGRAM ↔ RUN/MONITOR) следующие области памяти ввода/вывода обнуляются: область CIO (область ввода/вывода, область логических связей (Data Link), область модуля шины CPU, область специальных модулей ввода/вывода, область встраиваемых плат, область SYSMAC BUS, область терминалов ввода/вывода, область DeviceNet (CompoBus/D) и области внутренних входов/выходов), рабочая область, флаги завершения таймеров и текущие значения (PV) таймеров (области встраиваемых плат, SYSMAC BUS и терминалов ввода/вывода поддерживаются только модулями CPU серии CS).

#### Флаги и слова дополнительной области

Название	Адрес	Описание
Бит удержания IOM (IOM Hold)	A50012	Если этот бит установлен (ВКЛ), при изменении режима работы (PROGRAM ↔ RUN/MONITOR) сохраняются все данные памяти ввода/вывода.

Если бит удержания IOM (IOM Hold) включен, все выходы модулей вывода сохраняют свои состояния, когда прекращается выполнение программы. При последующем запуске



ске программы выходы будут находиться в тех же состояниях, в которых они находились в момент прекращения работы программы.

(Если бит удержания IOM (IOM Hold) выключен, команды начнут выполняться после обнуления выходов).

**Включение питания ПЛК**

\*Чтобы все данные в памяти ввода/вывода сохраняли свои значения после включения ПЛК (ВЫКЛ → ВКЛ), должен быть установлен (ВКЛ) бит удержания IOM (IOM Hold), а в настройках ПЛК должна быть выбрана его защита (адрес 80, Состояние бита удержания IOM при запуске).



**Флаги и слова дополнительной области**

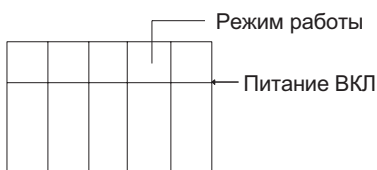
Название	Адрес	Описание
Бит удержания IOM (IOM Hold)	A50012	Когда этот бит включен, все данные памяти ввода/вывода сохраняются при изменении режима работы (PROGRAM ↔ RUN/MONITOR).

**Настройки ПЛК**

Адрес консоли программирования	Название	Значение	Значение по умолчанию
80, бит 15	Состояние бита удержания IOM при запуске	0: Бит удержания IOM обнуляется при включении питания. 1: Бит удержания IOM сохраняет свое значение при включении питания.	0 (Обнуление)

**6-6-2 Настройка режима запуска**

Первоначальный режим работы модуля CPU (сразу после включения питания) можно выбрать в настройках ПЛК.



**Настройки ПЛК**

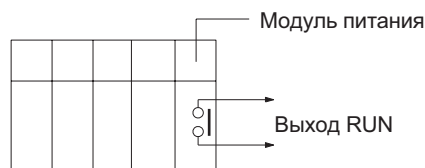
Адрес консоли программирования	Название	Назначение	Значение	Значение по умолчанию
81	Режим после запуска	Выбор режима работы, который устанавливается после запуска	PRCN: Переключатель режима на консоли программирования PRG: Режим PROGRAM MON: Режим MONITOR RUN: Режим RUN	PRCN: Переключатель режима на консоли программирования

**Примечание**

Если для выбора режима работы выбрано значение PRCN (переключатель режима работы на консоли программирования), но консоль программирования не подсоединена, в этом случае модуль CPU начнет работу в режиме RUN. Чтобы ПЛК начинал работу после включения питания в режиме MONITOR или в режиме PROGRAM, следует изменить стандартное значение в настройках ПЛК (При тех же условиях модули CPU серии CS начинают работу в режиме PROGRAM).

### 6-6-3 Выход RUN

В некоторых модулях питания (модули C200HW-PA204R, C200HW-PA209R, CJ1W-PA205R и CS1D-PA207R) предусмотрен выход RUN. Этот выход находится в состоянии ВКЛ (замкнут), когда модуль CPU работает в режиме RUN или MONITOR, и в состоянии ВЫКЛ (разомкнут), когда модуль CPU работает в режиме PROGRAM.



Этот выход RUN можно использовать для реализации дополнительных цепей защиты, например, цепи аварийного останова, которая запрещает подачу питания от внешнего модуля питания на модуль вывода, если ПЛК не включен.

**Примечание** Если применяется модуль питания без выхода RUN, аналогичный по принципу действия выход можно создать, применив в программе флаг "Всегда ВКЛ" (A1, Always ON) в качестве условия выполнения для выхода модуля вывода.

**Предупреждение**

Если напряжение питания от внешнего источника поступает на модуль вывода до включения питания ПЛК, в этом случае при первом включении ПЛК может произойти кратковременный сбой модуля вывода. Для исключения каких-либо сбоев следует предусмотреть внешнюю цепь, которая бы предотвращала подачу питания от внешнего источника на модуль вывода до того, как включается питание самого ПЛК. Создайте схему защиты, аналогичную описанной выше, чтобы питание от внешнего источника поступало только тогда, когда ПЛК находится в режиме RUN или MONITOR.

### 6-6-4 Настройка времени задержки обнаружения выключения питания

В общем случае обнаружение пропадания питания происходит спустя 10 ... 25 мс (2 ... 5 мс для источников питания постоянного тока) после падения уровня напряжения питания ниже 85% от минимального номинального значения (80% для источников питания постоянного тока). В настройках ПЛК предусмотрен параметр (адрес 225, биты 0 ... 7, Время задержки обнаружения выключения питания), с помощью которого можно увеличить это время максимум до 10 мс (максимум до 2 мс для DC-источников).

Если разрешено прерывание по выключению питания, после обнаружения пропадания питания будет выполнена задача обработки прерывания, в противном случае CPU будет сброшен сразу и работа его прекратится.

**Сопутствующие настройки**

Адрес	Название	Назначение	Значение	Значение по умолчанию
CIO 256, биты 00 ... 07	Задержка обнаружения выключения питания	Установите время задержки обнаружения прерывания питания.	00 ... 0A (Hex): 0 ... 10 мс	00 (Hex): 0 мс

### 6-6-5 Запрет прерываний от выключения питания

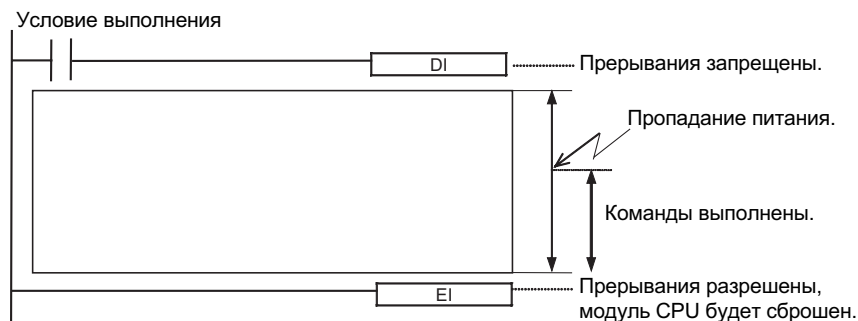
Эта функция поддерживается только модулями CPU CS1-H, CJ1-H, CJ1M или CS1D.

У модулей CPU CS1-H, CJ1-H, CJ1M или CS1D области программы можно защитить таким образом, чтобы их выполнение произошло даже в случае пропадания напряжения питания. Для этого используются команды DISABLE INTERRUPTS (DI(693)) и ENABLE INTERRUPTS (EI(694)).

Эту функцию можно использовать для набора команд, которые должны выполняться как группа, например, сделать так, чтобы выполнение не было начато с применением промежуточных сохраненных данных при следующем включении напряжения питания.

**Последовательность действий**

- 1,2,3... 1. Настройте параметр "Отключение прерываний по пропаданию питания" в словах A530 ... A5A5 Hex, разрешив запрет прерываний по выключению питания.
2. Разрешите запрет прерываний по выключению питания в настройках ПЛК (по умолчанию запрет прерываний разрешен).
3. С помощью DI(693) запретите прерывания перед входом в защищаемый раздел программы. После выхода из защищаемого раздела разрешите прерывания с помощью EI(694). Все команды, расположенные между DI(693) и EI(694), будут завершены, прежде чем будет выполнено прерывание по отключению питания, даже если пропадание питания произойдет во время выполнения команд, заключенных между DI(693) и EI(694).



**Сопутствующие параметры**

Название	Адрес	Назначение
Отключение прерываний по выключению питания	A530	С помощью DI(693) запрещается обработка прерываний по отключению питания (за исключением применения внутри задачи обработки прерывания по выключению питания) вплоть до выполнения EI(694). A5A5 Hex: С помощью DI(693) можно запрещать обработку прерываний по выключению питания Любое другое значение: запрет применения DI(693) для запрета обработки прерывания по выключению питания

**6-6-6 Функции часов**

В ПЛК серии CS/CJ предусмотрены следующие функции, использующие часы.

- Определение времени возникновения пропадания питания
- Определение времени включения ПЛК
- Определение суммарной продолжительности работы ПЛК

**Примечание** Модули CPU CS1 серии CS поставляются без штатной батареи, поэтому после установки батареи встроенные часы модуля CPU будут содержать значение 00/01/01 00:00:00 или, возможно, другое значение. Чтобы использовать функции часов, следует установить батарею, включить питание и настроить время и дату с помощью средства программирования (с помощью консоли программирования или CX-Programmer) или с помощью команды FINS (07 02, CLOCK WRITE). После того, как время и дата будут установлены, встроенные часы модуля CPU начнут работу.

**Флаги и слова дополнительной области**

Название	Адреса	Функция
Показания часов	A35100 ... A35107	Секунды: 00 ... 59 (BCD)
	A35108 ... A35115	Минуты: 00 ... 59 (BCD)
	A35200 ... A35207	Час: 00 ... 23 (BCD)
	A35208 ... A35215	День месяца: 00 ... 31 (BCD)
	A35300 ... A35307	Месяц: 00 ... 12 (BCD)
	A35308 ... A35315	Год: 00 ... 99 (BCD)
	A35400 ... A35407	День недели: 00: Воскресенье, 01: Понедельник, 02: Вторник, 03: Среда, 04: Четверг, 05: Пятница, 06: Суббота
Время запуска	A510 и A511	Содержит время включения питания.
Время прерывания питания	A512 и A513	Содержит время последнего пропадания питания.
Суммарное время включенного состояния	A523	Содержит суммарное время (двоичное значение), в течение которого ПЛК был включен, с шагом 10 часов.

**Сопутствующие команды**

Команда	Название	Функция
SEC(065)	HOURS TO SECONDS	Преобразует время в формате "часы/минуты/секунды" в эквивалентное значение времени в секундах.
HMS(066)	SECONDS TO HOURS	Преобразует время в секундах в эквивалентное время в формате "часы/минуты/секунды".
CADD(730)	CALENDAR ADD	Увеличивает значение времени на значение в указанных словах.
CSUB(731)	CALENDAR SUBTRACT	Уменьшает значение времени на значение в указанных словах.
DATE(735)	CLOCK ADJUSTMENT	Записывает во встроенные часы значение времени, содержащееся в указанных словах.

**6-6-7 Защита программы**

Программу пользователя в ПЛК серии CS/CJ можно защитить от записи или защитить полностью (защита от чтения/записи).

**Защита от записи с помощью DIP-переключателя**

Программу пользователя можно защитить от записи, переведя ключ 1 DIP-переключателя модуля CPU в положение ВКЛ. Когда переключатель находится в положении ВКЛ, изменить программу пользователя с помощью средства программирования (в том числе с помощью консоли программирования) будет невозможно. С помощью этой функции можно предотвратить случайную перезапись программы по месту ее использования.

Защищенную от записи программу можно, тем не менее, читать и отображать.

**Примечание** Для модулей CPU серии CS/CJ версий 2.0 или более поздних можно использовать указанный выше DIP-переключатель либо, если применяется CX-Programmer версии 4.0 или выше, защиту программы от чтения/записи можно установить, указав пароль либо для всей программы, либо для одной или нескольких задач.

Более подробно это описано в руководствах по эксплуатации *CS Series PLC Operation Manual* или *CJ Series PLC Operation Manual*, раздел 1-4-2 Улучшенная защита от чтения с помощью паролей, подраздел Разрешение/запрет создания файлов программ в памяти файлов.

#### Защита от чтения/записи с помощью паролей

Доступ к области программ пользователя для чтения и записи можно заблокировать из СХ-Programmer. Установка защиты позволит предотвратить несанкционированное копирование программы и кражу интеллектуальной собственности. С целью защиты программы с помощью средства программирования устанавливается пароль, в результате чего запрещается доступ полностью ко всей программе.

- Примечание**
1. Если вы забудете пароль, вы не сможете считать программу из ПЛК в компьютер. Обязательно запишите пароль и храните его в надежном месте.
  2. Если вы забудете пароль, вы не сможете загружать программы из компьютера в ПЛК. Программы могут загружаться из компьютера в ПЛК даже в том случае, когда парольная защита не была снята.

#### Защита пароля

- 1,2,3...**
1. Зарегистрируйте пароль в online- или в offline-режиме следующим образом:
    - a) Выберите **PLC** (ПЛК) и **Properties** (Свойства) в меню View (Вид).
    - b) В диалоговом окне **PLC Properties** (Свойства ПЛК) выберите **Protection** (Защита) и введите требуемый пароль.
  2. Установите парольную защиту в online-режиме следующим образом:
    - a) Выберите **PLC, Protection** (ПЛК, Защита), затем выберите **Set** (Установить). Будет отображено диалоговое окно Protection Setting (Установка защиты).
    - b) Щелкните по кнопке **OK**.

- Примечание**
- В модулях CPU серии CS/CJ версии 2.0 или выше защиту чтения можно установить не только для всей программы целиком, но также и для отдельных задач. Более подробно эта процедура описана в руководствах *SC Series PLC Operation Manual* или *CJ Series PLC Operation Manual*, раздел 1-4-2 Улучшенная защита от чтения с помощью паролей, подраздел Установка защиты от чтения для отдельных задач с помощью паролей.

#### Проверка даты создания программы пользователя

Модули CPU CS1-H, CJ1-H, CJ1M или CS1D позволяют определить дату создания программы и параметров путем чтения содержимого слов A090 ... A097.

Слова дополнительной области

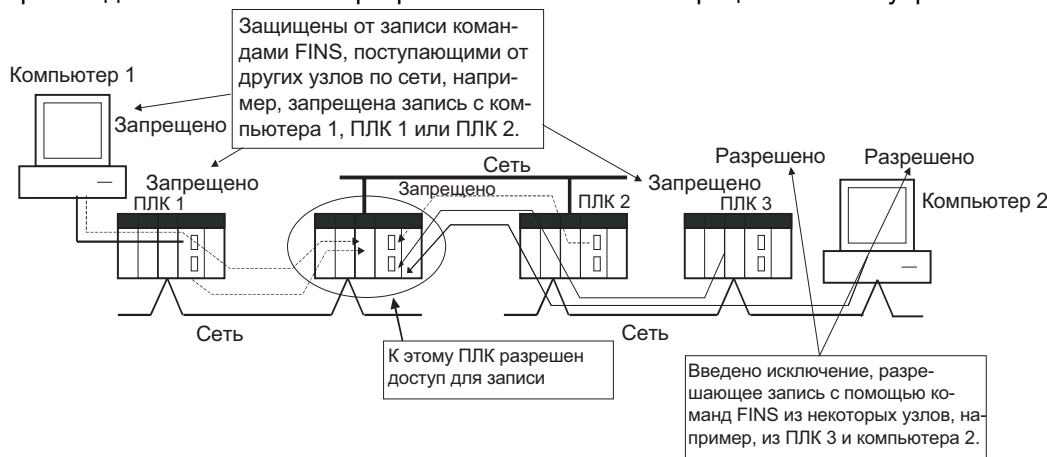
Название	Адрес	Функция
Дата создания программы пользователя	A090 ... A093	Содержит в формате BCD время и дату последней операции записи программы пользователя в память.
	A09000 ... A09007	Секунды: 00 ... 59 (BCD)
	A09008 ... A09015	Минуты: 00 ... 59 (BCD)
	A09100 ... A09107	Час: 00 ... 23 (BCD)
	A09108 ... A09115	День месяца: 00 ... 31 (BCD)
	A09200 ... A09207	Месяц: 00 ... 12 (BCD)
	A09208 ... A09215	Год: 00 ... 99 (BCD)
Дата создания параметров	A094 ... A097	Содержит в формате BCD время и дату последней операции записи параметров в память. Используется тот же формат, что и для даты программы пользователя (см. выше).
		День (00 ... 06 BCD) День недели: 00: Воскресенье, 01: Понедельник, 02: Вторник, 03: Среда, 04: Четверг, 05: Пятница, 06: Суббота

### 6-6-8 Защита от записи посредством команд FINS, передаваемых модулю CPU по сети

В модулях CPU серии CS/CJ версии 2.0 или выше можно установить защиту от записи или иного воздействия на модули CPU командами FINS, передаваемыми по сети (то есть, через другие соединения, помимо прямого соединения через последовательный интерфейс). Будут запрещены такие операции, как запись из приложений с применением FinsGateway, а также запись из CX-Programmer, CX-Protocol и CX-Process. В этом случае чтение по-прежнему возможно.

При установленной защите от записи будут запрещены следующие операции: загрузка программы пользователя, настроек ПЛК, данных памяти ввода/вывода и других данных; изменение режима работы; online-редактирование; и любые другие операции записи или управления.

Даже если защита от записи установлена, некоторые узлы могут быть исключены из запрета и для них может быть разрешено выполнение операций записи и управления.



**Примечание** Эта функция запрещает операции записи/управления только для команд FINS. Она не влияет на операции записи/управления, выполняемые любым другим способом, например, через логические связи (data link).

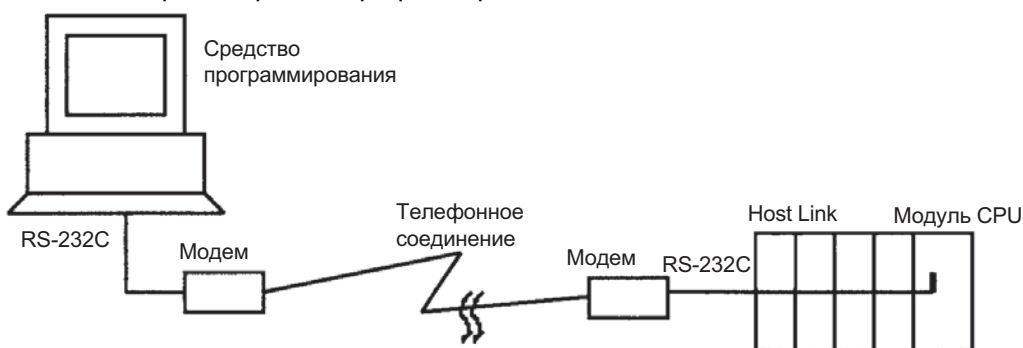
Подробно эта функция описана в руководствах *CS Series PLC Operation Manual* или *CJ Series PLC Operation Manual*, в разделе 1-4-3 *Защита от записи с помощью команд FINS, поступающих на модуль CPU по сети*.

### 6-6-9 Дистанционное программирование и мониторинг

Программирование и мониторинг ПЛК серии CS/CJ можно осуществлять дистанционно через модемное соединение или через сеть Controller Link.

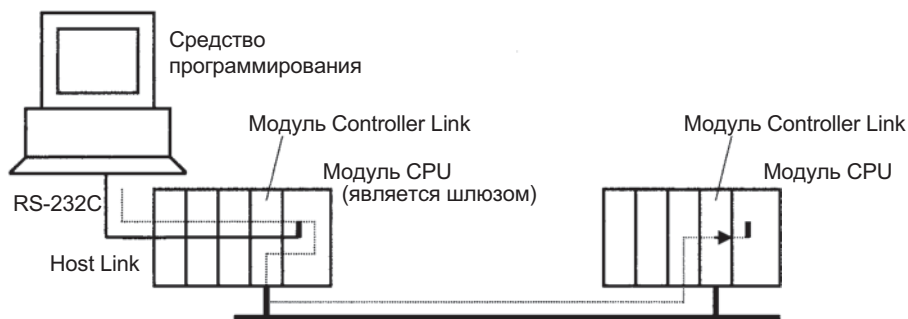
#### 1,2,3... 1. Модемные соединения

Функция host link (связь с центральным компьютером) может работать через модемное соединение, что позволяет дистанционно контролировать работу удаленного ПЛК, передавать данные и даже редактировать программу удаленного ПЛК в режиме online по телефону. Этими соединениями поддерживаются все online-операции средств программирования.



#### 2. Соединения по сети Controller Link

ПЛК, находящиеся в сети Controller Link или Ethernet, можно программировать и контролировать посредством протокола Host Link. Этими соединениями поддерживаются все online-операции средств программирования.



### 6-6-10 Профили модулей

Из пакета CX-Programmer можно прочитать следующую информацию о модулях серии CS/CJ.

- Сведения об их изготовлении (номер партии, заводской (серийный) номер и т.п.): упрощает предоставление информации в компанию OMRON в случае возникновения проблем с модулями.
- Сведения о модуле (тип, номер модели, правильное положение в стойке/установочное место): простой способ получения указаний по монтажу.
- Произвольный текст пользователя (макс. 256 символов): в карту памяти можно записывать информацию, необходимую для обслуживания (журнал проведения осмотров модуля, номера производственных линий и другие сведения).

### 6-6-11 Флеш-память

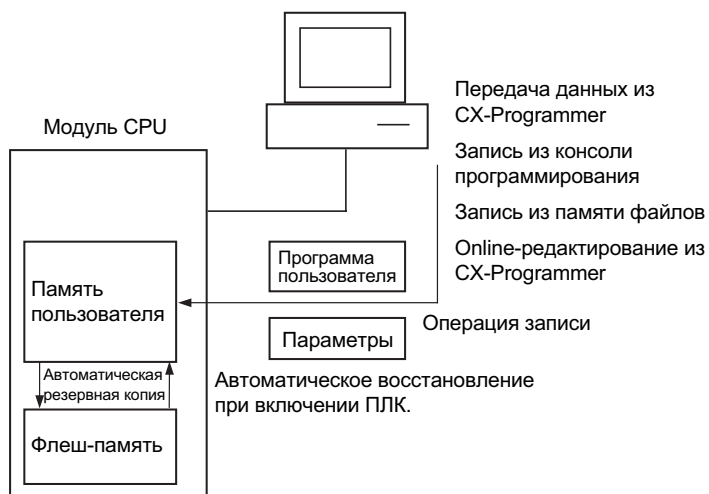
Эта функция поддерживается только модулями CPU CS1-H, CJ1-H, CJ1M или CS1D.

Всякий раз, когда программа пользователя и параметры записываются или изменяются в модуле CPU CS1-H, CJ1-H, CJ1M или CS1D, во флеш-памяти автоматически создается их резервная копия.

Резервная копия создается автоматически для следующих данных: программа пользователя, параметры (включая параметры в настройках ПЛК, зарегистрированные таблицы ввода/вывода, таблицы маршрутизации и данные модулей шины CPU, например, таблицы логических связей).

Резервная копия данных создается автоматически всякий раз, когда происходит запись программы пользователя или параметров в модуль CPU, включая такие операции, как передача данных из CX-Programmer, запись данных из консоли программирования, online-редактирование, загрузка данных из карты памяти или памяти файлов EM и т.п.

Программа пользователя и параметры, сохраненные во флеш-память, автоматически загружаются в память пользователя в модуле CPU при запуске.



**Примечание**

1. В процессе записи данных во флеш-память светится индикатор ВКУР на лицевой панели модуля CPU. Ни в коем случае не выключайте напряжение питания модуля CPU, пока не будет завершена процедура резервного сохранения, выполняемая после передачи данных из средства программирования или памяти данных, или после online-редактирования (т.е., пока не отключится индикатор ВКУР) .
2. Если в модуль CPU установлена батарея подпитки, после online-редактирования модуль CPU начнет работу в прежнем режиме (например, будет светиться индикатор ВКУР), даже если напряжение питания было выключено до завершения процедуры создания резервной копии, хотя для запуска модуля CPU может потребоваться до 1 минуты. Тем не менее даже в этом случае (даже если у модуля CPU имеется батарея подпитки) обязательно следует дожидаться завершения процедуры создания резервной копии, прежде чем отключать напряжение питания, если предполагается, что модуль CPU долгое время будет находиться в выключенном состоянии.



Время, необходимое для резервирования данных (продолжительность свечения индикатора ВКУР), зависит от объема программы пользователя, что отражено в следующей таблице.

Объем программы пользователя	Время резервного копирования		
	Режим MONITOR		Режим PROGRAM
	Длительность цикла 0.4 мс (пример)	Длительность цикла 10.0 мс (пример)	
10 000 шагов	2 с	8 с	1 с
60 000 шагов	11 с	42 с	6 с
250 000 шагов	42 с	170 с	22 с

**Примечание**

1. Индикатор ВКУР светится, когда на модуль CPU подано напряжение питания.
2. В зависимости от типа выполняемого online-редактирования для резервного копирования данных может потребоваться до 1 минуты.

 **Предупреждение**

Модули CPU CS1-H, CJ1-H, CJ1M и CS1D автоматически создают резервную копию программы пользователя и параметров во флеш-памяти, когда последние записываются в модуль CPU. В то же время, содержимое памяти ввода/вывода (включая области DM, EM и HR) во флеш-память не записывается. Состояние областей DM, EM и HR можно сохранить в случае пропадания питания с помощью батареи подпитки. Однако при возникновении проблем, связанных с батареей, содержимое этих областей может оказаться поврежденным после пропадания питания. Если содержимое областей DM, EM и HR служит для управления внешними выходными устройствами, необходимо предусмотреть блокировку выходных сигналов на случай возникновения ошибки батареи, т.е., когда установлен (ВКЛ) флаг "Ошибка батареи" (A40204).

**Примечание**

Если резервное копирование данных осуществляется из CX-Programmer для иных операций передачи данных, помимо обычных операций передачи данных (**PLC/Transfer - ПЛК/Передать**), в этом случае статус резервного копирования отображается в CX-Programmer в окне Memory Backup Status (Статус резервного копирования в память). Чтобы отобразить это окно, в свойствах ПЛК необходимо активизировать соответствующий параметр, отвечающий за отображение диалогового окна статуса резервного копирования, а в меню View (Вид) выбрать это окно. В случае обычных операций передачи данных состояние резервного копирования отображается в окне передачи следом за состоянием передачи для программы и других данных.

**Флаги дополнительной области**

Название	Адрес	Назначение
Флаг "Ошибка флеш-памяти"	A40310	Устанавливается в случае возникновения сбоя флеш-памяти.

### 6-6-12 Настройка условий запуска

Эта функция поддерживается только модулями CPU CS1-H, CJ1-H, CJ1M или CS1D.

Некоторым модулям и встраиваемым платам требуется значительное время для запуска после включения питания, что влияет на время запуска модуля CPU. Параметры в настройках ПЛК можно сконфигурировать таким образом, чтобы модуль CPU запустился без инициализации этих модулей.

Эта настройка действует для интерфейсных модулей ITNC-EIS01-CST и ITNC-EIX01-CST (Open Network Controller-CS1 Bus Interface Units) (начиная с октября 2001 и до настоящего момента встраиваемые платы, для которых действуют эти настройки, отсутствуют).

Эта функция настраивается с помощью параметра "Условия запуска" и "Ожидание встраиваемых плат", которые описаны в следующей таблице.

Условия запуска	Настройки ПЛК	
	Условия запуска (консоль программирования, адрес 83, бит 15)	Ожидание встраиваемых плат (консоль программирования, адрес 84, бит 15)
Запуск без ожидания всех модулей и плат	1: Разрешается работа без ожидания.	1: Не ожидать определенные встраиваемые платы
Запуск без ожидания всех модулей (ожидание плат)	1: Разрешается работа без ожидания.	0: Ожидать завершения инициализации всех плат перед запуском.
Ожидание всех модулей и плат перед запуском	0: Всегда ожидать завершения инициализации всех модулей/плат.	Любое состояние

**Примечание** Модуль CPU CS1 не запустится, пока не будет завершена процедура запуска всех модулей и плат.

### Настройки ПЛК

Адрес в консоли программирования		Название	Значение	Значение по умолчанию	Обновление в модуле CPU
Слово	Бит				
83	15	Условие запуска	0: Ожидать всех модулей и плат. 1: Не ожидать.	0: Ожидать	Включение питания
84	15	Настройка встраиваемой платы	0: Ожидать все платы. 1: Не ожидать определенные платы.	0: Ожидать	Включение питания

### Условия запуска

0: если имеется одна или несколько определенных плат или модулей, которые не завершили процедуру запуска, модуль CPU перейдет в состояние ожидания (standby) в режиме MONITOR или PROGRAM и будет ожидать завершения инициализации всех модулей и плат.

1: даже если имеется одна или несколько определенных плат или модулей, которые не завершили процедуру запуска, модуль CPU продолжит работу и запустится в режиме MONITOR или PROGRAM. В случае встраиваемых плат, однако, работа также зависит от следующего параметра.

### Ожидание встраиваемых плат

Этот параметр используется только тогда, когда для параметра "Условия запуска" задано значение 1, то есть, разрешен запуск без ожидания определенных модулей и плат. Этот параметр игнорируется, если параметр "Условия запуска" сброшен в 0.

0: если имеется одна или несколько отдельных плат, которые не завершили процедуру запуска, модуль CPU перейдет в состояние ожидания (standby) в режиме MONITOR или PROGRAM и будет ожидать завершения запуска всех плат.

1: даже если имеется одна или несколько определенных плат, которые не завершили процедуру запуска, модуль CPU продолжит работу и запустится в режиме MONITOR или PROGRAM.

## 6-7 Функции диагностики

В этом разделе приводится краткий обзор следующих функций диагностики и отладки.

- Протоколирование ошибок
- Функция отключения выходов
- Функции сигнализации сбоев (FAL(006) и FALS(007))
- Функция обнаружения точки сбоя (FPD(269))

### 6-7-1 Протокол ошибок

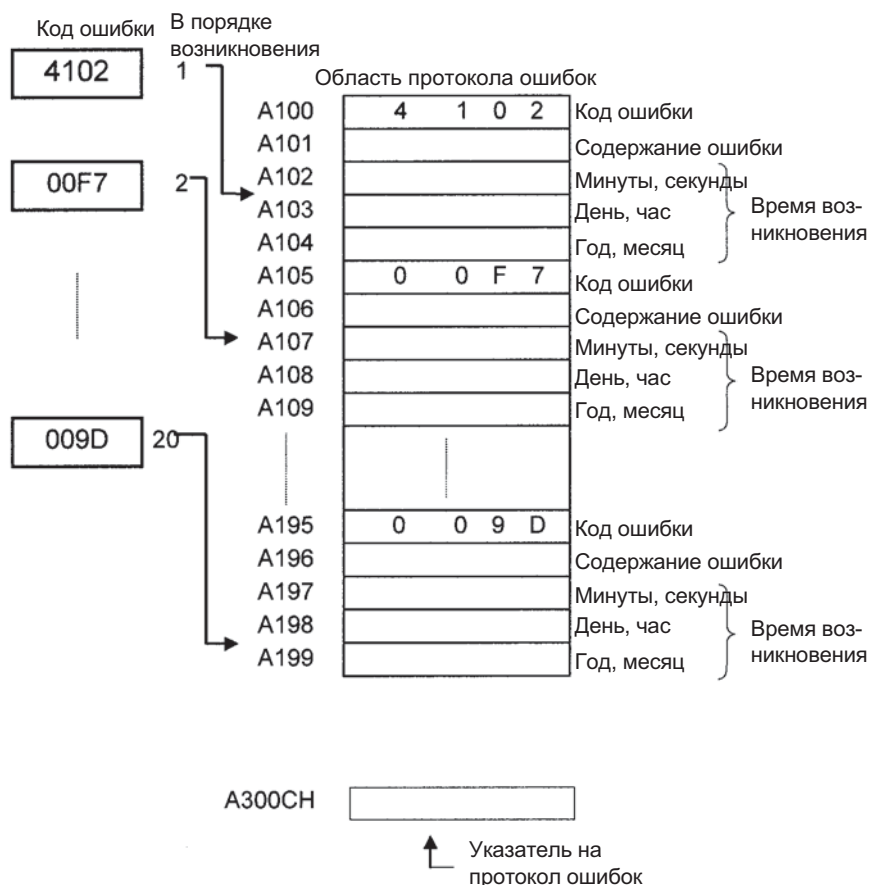
Всякий раз, когда в ПЛК серии CS/CJ происходит ошибка, модуль CPU сохраняет сведения об ошибке в область протокола ошибок. Сведения об ошибке включают в себя код ошибки (записывается в A400), содержание ошибки и время ее возникновения. В протокол ошибок может быть занесено до 20 записей.

Помимо системных ошибок ПЛК также регистрирует ошибки, сконфигурированные пользователем (FAL(006) и FALS(007)), что упрощает контроль за рабочим состоянием системы.

Подробно эти функции описаны в разделе, посвященном поиску и устранению ошибок, в руководстве *CS/CJ Series Operation Manual*.

**Примечание** Ошибка, сконфигурированная пользователем, генерируется, когда в программе выполняется FAL(006) или FALS(007). Условиями выполнения для этих команд являются условия, которые пользователь определил как условия ошибки. FAL(006) генерирует нефатальную ошибку, а FALS(007) генерирует фатальную ошибку, которая приводит к прекращению выполнения программы.

Если происходит больше 20 ошибок, наиболее старая запись об ошибке (в A100 ... A104) удаляется, оставшиеся 19 записей смещаются вниз на одну строку, и в A195 ... A199 заносится самая свежая запись.



Количество записей хранится в двоичном формате в слове указателя на протокол ошибок (A300). Если произошло больше 20 ошибок, содержание указателя в дальнейшем не наращивается.

### 6-7-2 Функция отключения выходов

В качестве меры аварийной защиты в случае возникновения ошибок все выходы модулей вывода могут быть отключены. Для этого следует установить (ВКЛ) бит "Отключить выходы" (A50015). ПЛК будет по-прежнему работать в режиме RUN или MONITOR, но все выходы будут выключены.

**Примечание** В общем случае (когда бит "Удержание IOM" = ВЫКЛ) все выходы модулей вывода выключаются при переходе из режима работы RUN/MONITOR в режим PROGRAM. Бит "Отключить выходы" можно использовать для выключения всех выходов без необходимости переключения в режим PROGRAM и прекращения выполнения программы.

**Замечание для DeviceNet** Если используется функция ведущего устройства с модулями CS1W-DRM21 или CJ1W-DRM21, выходы всех ведомых устройств будут отключены. Если используется функция ведомого устройства, все входы ведущего устройства будут отключены. В то же время, если используется C200HW-DRM21-V1, выходы ведомых устройств отключены не будут.

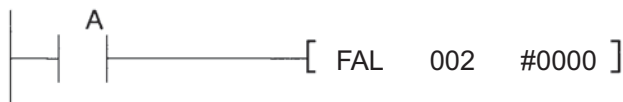
### 6-7-3 Функции формирования предупреждений о неисправностях

Команды FAL(006) и FALS(007) генерируют ошибки, сконфигурированные пользователем. FAL(006) генерирует нефатальную ошибку, а FALS(007) генерирует фатальную ошибку, которая прекращает выполнение программы.

Если выполняются условия, установленные пользователем как условия ошибки (условия выполнения для FAL(006) и FALS(007)), выполняется команда Failure Alarm (сигнализация сбоя), после чего выполняются дальнейшие операции.

- 1,2,3...
1. Устанавливается (ВКЛ) флаг "Ошибка FAL" (A40215) или флаг "Ошибка FALS" (A40106).
  2. В A400 записывается соответствующий код ошибки.
  3. В протокол ошибок записывается код ошибки и время ее возникновения.
  4. Индикатор ошибки на лицевой панели модуля CPU мигает или светится.
  5. Если выполнена команда FAL(006), модуль CPU продолжает работу. Если выполнена команда FALS(007), модуль CPU прекращает работу (выполнение программы прекращается).

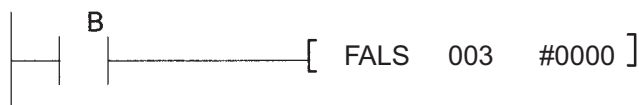
**Работа команды FAL(006)**



Когда выполняется (ВКЛ) условие выполнения A, генерируется ошибка FAL с номером 2, устанавливается флаг A40215 ("Ошибка FAL"), устанавливается флаг A36002 ("FAL № 2"). Выполнение программы продолжается.

Ошибки, сгенерированные посредством FAL(006), можно сбросить, выполнив команду FAL(006) с номером 0, либо выполнив операцию чтения/сброса ошибки с помощью средства программирования (в том числе с консоли программирования).

**Работа команды FALS(007)**



Когда выполняется условие выполнения B, генерируется ошибка FALS с номером 3, устанавливается (ВКЛ) флаг A40106 ("Ошибка FALS"). Выполнение программы прекращается.

Ошибки, сгенерированные посредством FALS(007), могут быть сброшены путем устранения причины ошибки и выполнения операции чтения/сброса ошибки с помощью средства программирования (в том числе с консоли программирования).

**6-7-4 Обнаружение местонахождения неисправности**

Команда FPD(269) позволяет контролировать время и выполнять логическую диагностику. Функция контроля времени генерирует нефатальную ошибку, если выход диагностики не включается в течение определенного времени контроля. Функция логической диагностики указывает, какой именно вход не позволяет выходу диагностики перейти в состояние ВКЛ.

**Функция контроля времени**

Команда FPD(269) начинает отсчет времени после своего выполнения и устанавливает флаг переноса, если выход диагностики не переходит в состояние ВКЛ за определенное контрольное время. Флаг переноса (Carry Flag) можно использовать в качестве условия выполнения для блока обработки ошибки. Кроме того, FPD(269) можно включить в программу в целях генерирования нефатальной ошибки FAL с требуемым номером.

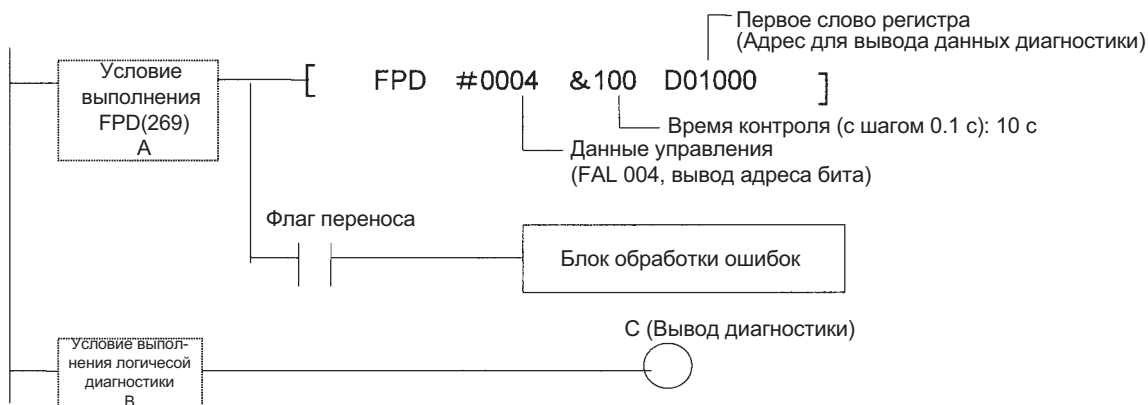
Когда генерируется ошибка FAL, регистрируется предустановленное сообщение, которое может быть отображено на средстве программирования. FPD(269) можно сконфигурировать таким образом, чтобы непосредственно перед сообщением выводились результаты логической диагностики (адрес бита, не позволяющего диагностическому выходу перейти в состояние ВКЛ).

Для автоматического определения фактического времени, требуемого выходу диагностики для перехода в состояние ВКЛ, и для настройки контрольного времени можно использовать функцию обучения.

**Функция логической диагностики**

Команда FPD(269) определяет, какой именно входной бит является причиной того, что выход диагностики остается выключенным, и выводит либо адрес этого бита (адрес памяти ПЛК) или сообщение (ASCII).

- Если выбран вывод адреса бита, в этом случае адрес памяти ПЛК бита может быть записан в регистр индексов, после чего регистр индексов можно использовать для косвенной адресации.
- Если выбран вывод сообщения, адрес бита будет записан в сообщение в формате ASCII, которое может быть отображено на средстве программирования.



**Контроль времени:**

Контролирует, перейдет ли выход С в состояние ВКЛ в течение 10 секунд после включения входа А. Если С не включится в течение 10 секунд, это будет признано за ошибку и включится флаг переноса (Carry Flag). Флаг переноса инициирует выполнение блока обработки ошибок. Кроме того, будет сгенерирована ошибка FAL (нефатальная ошибка) с номером FAL(004).

**Логическая диагностика:**

Команда FPD(269) определяет, какой именно входной бит блока В не позволяет выходу С перейти во ВКЛ. Адрес этого бита будет выведен в D01000 и в D01001.

**Флаги и слова дополнительной области**

Название	Адрес	Операция
Код ошибки	A400	Когда возникает ошибка, код ошибки записывается в A400.
Флаг "Ошибка FAL"	A40215	Установлен, когда генерируется FAL(006).
Флаг "Ошибка FALS"	A40106	Установлен, когда генерируется FALS(007).
Флаги "Номер сгенерированной FAL"	A360 ... A391	Когда генерируется ошибка FAL(006) или FALS(007), устанавливается (ВКЛ) соответствующий флаг.
Область протокола ошибок	A100 ... A199	Область протокола ошибок содержит информацию о последних 20 ошибках.
Указатель на протокол ошибок	A300	Когда возникает ошибка, указатель на протокол ошибок увеличивается на 1, указывая, куда будет записана следующая запись об ошибке. Адрес отсчитывается от начала области протокола ошибок (A100).
Бит "Сброс указателя протокола ошибок"	A50014	Чтобы сбросить указатель на протокол ошибок (A300) в 00, следует включить этот бит.
Бит "Обучение FPD"	A59800	Чтобы время контроля устанавливалось автоматически при выполнении команды FPD(269), следует включить этот бит.

### 6-7-5 Имитирование системных ошибок

Эта функция поддерживается только модулями CPU CS1-H, CJ1-H, CJ1M или CS1D. Команды FAL(006) и FALS(007) можно использовать для преднамеренного формирования фатальных и нефатальных системных ошибок. Это можно применять при системной отладке, когда требуется проверить, как отображаются сообщения на программируемых терминалах (PT), или проверить другие операторские интерфейсы. Используется следующая последовательность действий.

1,2,3...

1. Установите в A529 номер FAL или FALS, который должен использоваться для имитации (A529 используется при имитации ошибок для FAL(006) и FALS(007)).
2. Укажите номер FAL или FALS, который должен использоваться для имитации, в качестве первого операнда FAL(006) или FALS(007)).
3. Укажите код ошибки и ошибку, которая должна имитироваться, в качестве второго операнда (два слова) команды FAL(006) или FALS(007). Укажите нефатальную ошибку для FAL(006) и фатальную ошибку - для FALS(007).

Чтобы имитировать несколько системных ошибок, следует использовать несколько команд FAL(006) или FALS(007), используя процедуру, описанную выше.

#### Флаги и слова дополнительной области

Название	Адрес	Процедура
Номер FAL/FALS для имитации системных ошибок	A529	Укажите формальный номер FAL/FALS, который должен использоваться для имитации системной ошибки. 0001 ... 01FF Hex: номера FAL/FALS 1 ... 511 0000 или 0200 ... FFFF Hex: номера FAL/FALS для имитации системной ошибки не заданы.

#### Пример для ошибки батареи



**Примечание** Для сброса имитируемых системных ошибок применяются те же способы, что и для настоящих системных ошибок. Описание смотрите в руководствах *CS-series Operation Manual* или *CJ-series Operation Manual*. Все системные ошибки, которые были имитированы с помощью FAL(006) или FALS(007), можно обнулить, отключив и вновь включив напряжение питания.

### 6-7-6 Запрет записи установленных пользователем ошибок FAL в протокол ошибок

Эта функция поддерживается только модулями CS1-H, CJ1-H, CJ1M или CS1D. В настройках ПЛК имеется параметр, который позволяет отменить регистрацию в протоколе ошибок (A100 ... A199) сконфигурированных пользователем ошибок FAL, сгенерированных с помощью FAL(006), а также контроль времени для команды FPD(269). Даже если активизирован этот параметр, ошибки FAL по-прежнему генерируются и выводится следующая информация: A40215 (флаг "Ошибка FAL"), A360 ... A391 (номера сгенерированных FAL) и A400 (код ошибки).

Эту функцию можно использовать тогда, когда необходимо, чтобы в протокол ошибок заносились только системные ошибки FAL. Например, когда в программе генерируется слишком много ошибок пользователя командами FAL(006) и протокол ошибок слишком быстро переполняется.

### Настройки ПЛК

Адрес в кон-соли программирования		Название	Значение	Значение по умолчанию	Обновление в модуле CPU
Слово	Бит				
129	15	Протоколирование ошибок FAL пользователя	0: Записывать ошибки FAL пользователя в протокол ошибок 1: Не записывать ошибки FAL пользователя в протокол ошибок	0: Записывать	При каждом выполнении FAL(006) (в каждом цикле)

**Примечание** Даже если с помощью описанного выше параметра отменена запись ошибок FAL пользователя в протокол, в протокол ошибок заносятся следующие записи.

- Фатальные ошибки, сконфигурированные пользователем (FALS(007))
- Нефатальные системные ошибки
- Фатальные системные ошибки
- Нефатальные системные ошибки, имитируемые пользователем (FAL(006))
- Фатальные системные ошибки, имитируемые пользователем (FALS(007))

## 6-8 Режимы работы CPU

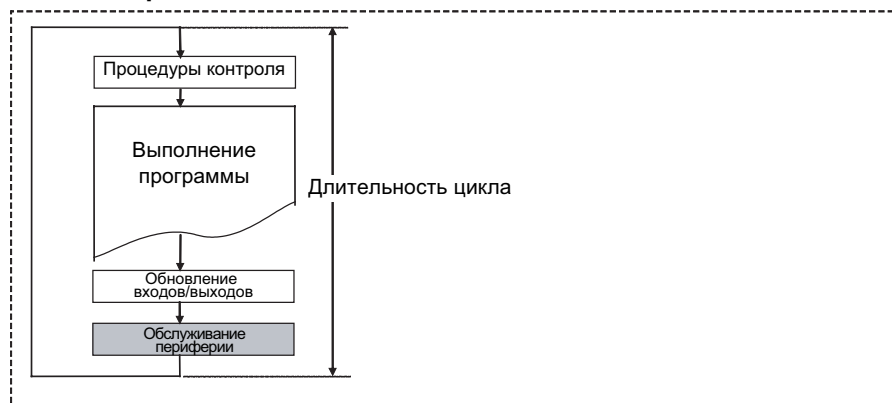
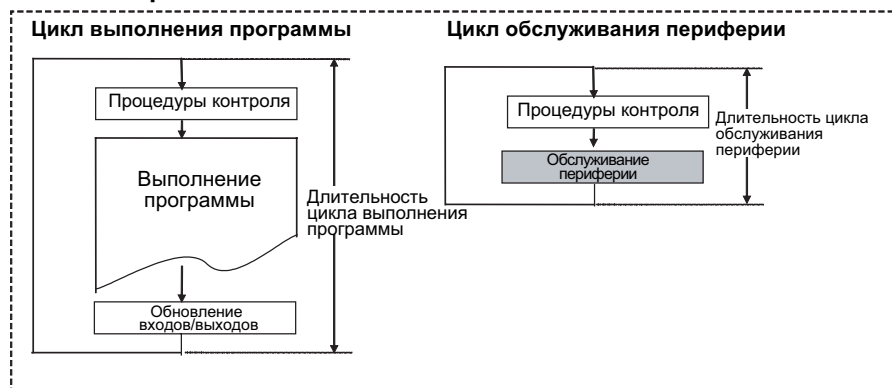
### 6-8-1 Режимы работы CPU

В общем случае, обслуживание периферии (см. прим.) выполняется однократно в конце каждого цикла (сразу за обновлением входов/выходов) и занимает время, составляющее 4% от длительности цикла, либо время, установленное пользователем для каждого сервиса. Вследствие этого, обслуживание периферии не может происходить чаще, чем это определено длительностью цикла, а длительность цикла увеличивается на время, необходимое для обслуживания периферии.

С другой стороны, модули CPU CS1D для однопроцессорных систем, а также модули CPU серии CS1-H или CJ1-H поддерживают режимы параллельного выполнения, в которых возможно одновременное выполнение программы и обслуживание периферии. В этих режимах обеспечивается более быстрое обслуживание периферийных устройств и, одновременно, сокращение длительности цикла программы, особенно, в тех случаях, когда необходимо очень интенсивное обслуживание периферийных устройств (модули CPU CS1D для двухпроцессорных систем и модули CPU CJ1M не поддерживают режимы параллельного выполнения).

**Примечание** В состав периферийных сервисов входят ациклические (незапланированные) сервисы, необходимые для обслуживания внешних устройств, например, для обслуживания событий (например, при обмене данными с помощью команд FINS) при работе со специальными модулями ввода/вывода, модулями шины CPU и встраиваемыми платами (только для серии CS), а также для обслуживания портов связи (параллельного порта и порта RS-232C) (за исключением логических связей) и обновления прочих специальных входов/выходов у модулей шины CPU).



**Обычный режим****Режимы параллельного выполнения****Режимы параллельного выполнения**

Предусмотрено два различных режима параллельного выполнения: параллельное выполнение с синхронным обращением к памяти и параллельное выполнение с асинхронным обращением к памяти.

■ **Параллельное выполнение с асинхронным обращением к памяти**

В этом режиме обращение к памяти ввода/вывода для обслуживания периферии не синхронизируется с обращением к памяти ввода/вывода для выполнения программы. Другими словами, все сервисы обслуживания периферии выполняются одновременно с выполнением программы, в том числе одновременно происходит обращение к памяти. В этом режиме выполнение происходит быстрее (по сравнению с другими режимами) как с точки зрения выполнения программ, так и с точки зрения обработки событий, что может потребоваться, когда обслуживание периферии происходит слишком интенсивно.

■ **Параллельное выполнение с синхронным обращением к памяти**

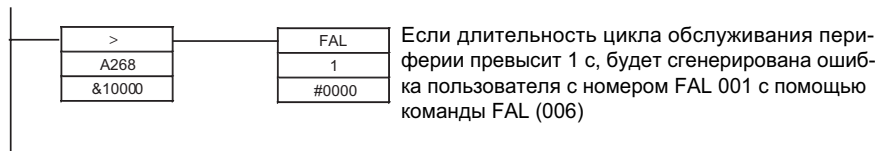
В этом режиме обращение к памяти ввода/вывода для обслуживания периферии не выполняется одновременно с выполнением программы, а происходит либо после выполнения программы, либо как и в обычном режиме выполнения, т.е., после обновления входов/выходов. Все остальные операции по обслуживанию периферии выполняются одновременно с выполнением программы.

В этом режиме обеспечивается более быстрое выполнение по сравнению с режимом обычного выполнения как с точки зрения выполнения программ, так и с точки зрения обработки событий. Длительность цикла выполнения программы больше по сравнению с параллельным выполнением с асинхронным обращением к памяти на время, необходимое для обновления входов/выходов, выполняемого для обслуживания периферии.

Ниже приведена таблица, в которой перечислены значения длительности цикла и скорости обслуживания периферии для обычного режима, режима параллельного выполнения с асинхронным обращением к памяти и режима параллельного выполнения с синхронным обращением к памяти (значения приведены для программы, состоящей из базовых команд, с длительностью цикла 10 мс, при использовании одного модуля Ethernet. Значения приведены в качестве ориентировочных и могут изменяться в зависимости от конкретной системы).

Параметр	Обычный режим	Параллельное выполнение с асинхронным обращением к памяти	Параллельное выполнение с синхронным обращением к памяти
Длит. цикла	Принуд. уст. в 1	0.9	0.9
Обслужив. периферии	Принуд. уст. в 1	0.4	1.0

- Примечание 1.** Обслуживание периферийных устройств включает в себя обслуживание событий (например, обмен данными посредством команд FINS) для специальных модулей ввода/вывода, модулей шины CPU и встраиваемых плат (только для серии CS), а также обслуживание портов связи ( периферийного порта и порта RS-232C (исключая обновление логических связей) и других специальных входов/выходов для модулей шины CPU).
- Модули CPU CS1 версии 1 или выше, а также модули CPU CS1-H или CJ1-H также поддерживают режим приоритетного обслуживания периферии, в котором обслуживание периферийных устройств осуществляется с фиксированной длительностью цикла во время выполнения программы. Благодаря этому обеспечивается более быстрое обслуживание периферии по сравнению с обычным режимом выполнения, но программа при этом выполняется медленнее. В то же время, отклик на событие не будет таким же быстрым, как и в режимах параллельного выполнения. Поэтому, если при обслуживании периферии основное внимание уделяется скорости реакции на события, обязательно следует использовать режим параллельного выполнения с асинхронным обращением к памяти.
  - В случае параллельного выполнения могут возникать описанные ниже ошибки, связанные с превышением длительности цикла обслуживания периферии в модуле CPU. В случае возникновения этой ошибки на дисплее средства программирования будет указано, что длительность цикла слишком велика, будет установлен флаг A40515 (Превышение длительности цикла обслуживания периферии) и работа будет прекращена (фатальная ошибка).
    - Если длительность цикла обслуживания периферии превышает 2.0 с, произойдет ошибка превышения длительности цикла. Продолжительность цикла обслуживания периферии можно контролировать с помощью A268, что позволяет обнаружить возможные ошибки, прежде чем они произойдут. Например, используя FAL с номером 001, можно генерировать ошибку пользователя, если длительность цикла обслуживания периферии превышает 1 с (т.е., если содержимое A268 превышает 2710 Hex (10000 десятич.)).



- Ошибка превышения длительности цикла обслуживания периферии также может возникнуть в том случае, если длительность выполнения слишком мала для цикла выполнения команды (т.е., времени выполнения команды). Это время хранится в A266 и A267 в режиме обычного выполнения. В качестве ориентира можно считать, что, если время выполнения команды составляет 2 мс или меньше, произойдет ошибка превышения длительности цикла обслуживания периферии и режим параллельного выполнения использовать нельзя. При отладке только отдельных участков программы (которые могут вызвать слишком короткое время выполнения команды) следует использовать обычный режим в целях предотвращения возникновения этой ошибки.

Когда прикладная программа пользователя работает в режиме параллельного выполнения, консоль программирования должна быть отсоединена. Для консоли программирования резервируется время обслуживания в целях обеспечения реакции на нажатие клавиш консоли программирования, что увеличивает длительность цикла обслуживания периферии и снижает эффективность параллельного выполнения.

## Настройки ПЛК

Режим выполнения выбирается в настройках ПЛК.

Адрес в консоли программирования		Название	Значение	Значение по умолчанию	Обновление в модуле CPU
Слово	Бит				
219	08 ... 15	Режим работы CPU	00 Hex: Обычный режим 01 Hex: Параллельное выполнение с синхронным обращением к памяти 02 Hex: Параллельное выполнение с асинхронным обращением к памяти 05 ... FF Hex: Временное окно в цикле выполнения программы для режима приоритетного обслуживания периферии (5...255 мс с шагом 1 мс)  Значения 03 и 04 Hex не определены и приведут к возникновению ошибок настройки ПЛК (нефатальные ошибки)	00 Hex: Обычный режим	В начале работы

## Флаги и слова дополнительной области

Название	Адрес	Описание
Превышение длительности цикла обслуживания периферии	A40515	Устанавливается, когда длительность цикла обслуживания периферии превышает 2 с. Работа прекращается.
Длительность цикла обслуживания периферии	A268	Содержит значение длительности цикла обслуживания периферии, когда используется один из режимов параллельного выполнения (с синхронным или асинхронным доступом к памяти) и ПЛК работает в режиме RUN или MONITOR. Значение времени хранится в двоичном формате в диапазоне 0.0...2000.0 (с шагом 0.1 мс).
Время выполнения команды (сумма всех временных окон для выполнения программы и всех временных окон для обслуживания периферии)	A266 и A267	В обычном режиме учитывается только время выполнения команды. Время хранится в формате 32-битового двоичного числа. 00000000 ... FFFFFFFF Hex (шаг: 0.1 мс) (0 ... 429 496 729.5 мс) A266: Младшее слово A267: Старшее слово

## Параллельное выполнение с асинхронным обращением к памяти

## Выполнение программы

Процедуры контроля		Проверка шины ввода/вывода и прочие процедуры 0.3 мс
Время выполнения команды		Суммарное время выполнения для всех команд
Расчет минимальной длительности цикла		Время выполнения для минимальной длительности цикла выполнения программы
Циклическое обслуживание	Обновление входов/выходов	Время обновления входов/выходов для каждого модуля * кол-во модулей
	Обновление специальных входов/выходов для модулей шины CPU	Время обновления специальных входов/выходов для каждого модуля * количество модулей
Обслужив. периферии	Обращение к файлам	Время обслуживания периферии, установленное в настройках ПЛК (по умолчанию: 4% от длительности цикла)

**Обслуживание периферии**

Процедуры контроля		Проверка батареи, проверка памяти программ пользователя и т.п. 0.2 мс
Обслуживание периферии	Обслуживание событий для специальных модулей ввода/вывода	Включает обслуживание событий для обращения к памяти ввода/вывода (см. прим.) Макс. 1 с на каждый сервис
	Обслуживание событий для модулей шины CPU	
	Обслуживание периферийного порта	
	Обслуживание порта RS-232C	
	Обслуживание событий для встраиваемых плат (только для серии CS)	
Обслуживание периферии для используемых портов связи (внутренних логических портов) (включая выполнение в фоновом режиме)		

**Примечание** Обслуживание событий для обращения к памяти ввода/вывода включает: 1) Обслуживание любых полученных команд FINS, обращающихся к памяти ввода/вывода (команды чтения/записи из/в память ввода/вывода с общими кодами, начиная с 01 Hex, либо команды принудительной установки/сброса с общими кодами, начиная с 23 Hex) и 2) Обслуживание любых принятых команд C-mode, обращающихся к памяти ввода/вывода (исключая каналы NT Link, использующие периферийный порт или порт RS-232C).

**Параллельное выполнение с синхронным обращением к памяти****Выполнение программы**

Процедуры контроля		Проверка шины ввода/вывода и прочие процедуры 0.3 мс
Время выполнения команды		Суммарное время выполнения для всех команд
Расчет минимальной длительности цикла		Время выполнения для минимальной длительности цикла выполнения программы
Циклическое обслуживание	Обновление входов/выходов	Время обновления входов/выходов для каждого модуля * кол-во модулей
	Обновление специальных входов/выходов для модулей шины CPU	Время обновления специальных входов/выходов для каждого модуля * количество модулей
Обслужив. периферии	Обращение к файлам	Время обслуживания периферии, установленное в настройках ПЛК (по умолчанию: 4% от длительности цикла)
	Обслуживание событий, требующее обращения к памяти ввода/вывода (см. прим.)	

**Обслуживание периферии**

Процедуры контроля		Проверка батареи, проверка памяти программ пользователя и т.п. 0.2 мс
Обслуживание периферии	Обслуживание событий для специальных модулей ввода/вывода	Включает обслуживание событий для обращения к памяти ввода/вывода (см. прим.) Макс. 1 с на каждый сервис
	Обслуживание событий для модулей шины CPU	
	Обслуживание периферийного порта	
	Обслуживание порта RS-232C	
	Обслуживание событий для встраиваемых плат (только для серии CS)	
Обслуживание событий для используемых портов связи (внутренних логических портов)(включая выполнение в фоновом режиме)		

**Примечание** Обслуживание событий для обращения к памяти ввода/вывода включает: 1) Обслуживание любых полученных команд FINS, обращающихся к памяти ввода/вывода (команды чтения/записи из/в память ввода/вывода с общими кодами, начиная с 01 Hex, либо команды принудительной установки/сброса с общими кодами, начиная с 23 Hex) и 2) Обслуживание любых принятых команд C-mode, обращающихся к памяти ввода/вывода (исключая каналы NT Link, использующие периферийный порт или порт RS-232C).

### 6-8-2 Режим параллельного выполнения и минимальные длительности циклов

Если используется режим параллельного выполнения и указана минимальная длительность цикла, после завершения программы отсчитывается время до истечения минимальной длительности цикла, но обслуживание периферии при этом продолжается.

### 6-8-3 Согласованность данных в режиме параллельного выполнения при асинхронном обращении к памяти

В случае применения параллельного выполнения с асинхронным обращением к памяти данные могут оказаться несогласованными в следующих ситуациях.

- Когда с помощью команды связи из памяти ввода/вывода считывается сразу несколько слов, данные, содержащиеся в словах, могут быть несогласованными.
- Если команда осуществляет чтение нескольких слов памяти ввода/вывода и во время выполнения команд происходит обслуживание периферии, данные, содержащиеся в словах, могут быть несогласованными.
- Если одно и то же слово памяти ввода/вывода читается одновременно несколькими командами, находящимися в различных местах программы, и между выполнением команд производится обслуживание периферии, данные, содержащиеся в слове, могут быть несогласованными.

Если необходимо, для обеспечения согласованности данных можно применить следующие методы.

1. Применяйте параллельное выполнение с синхронным обращением к памяти.
2. Используйте IOSP(287), чтобы запретить обслуживание периферии, если это требуется на соответствующем участке программы, и используйте IORS(288), чтобы вновь разрешить обслуживание периферии.

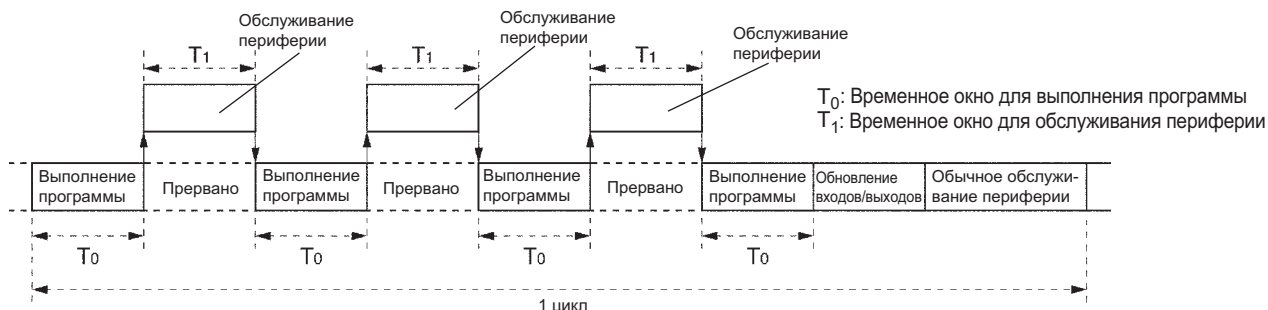
## 6-9 Режим приоритетного обслуживания периферии

Обслуживание периферии для порта RS-232C, периферийного порта, встраиваемой платы (только для серии CS), модулей шины CPU и специальных модулей ввода/вывода в общем случае производится только один раз в конце цикла после обновления входов/выходов. Для каждого сервиса отводится либо 4% длительности цикла, либо время, установленное пользователем. В то же время, предусмотрен режим, в котором разрешено периодическое обслуживание в пределах цикла. Этот режим называют режимом приоритетного обслуживания периферии, и выбирается он в настройках ПЛК.

**Примечание** Режим приоритетного обслуживания периферии можно использовать в модулях CPU серии CJ или CS, однако модуль CPU серии CS должен принадлежать к партии 001201□□□□ или к более поздней (дата изготовления: 1 Декабря 2000г. или позже). (Режим приоритетного обслуживания периферии не поддерживается модулями CPU CS1D для двухпроцессорных систем).

## 6-9-1 Режим приоритетного обслуживания периферии

Если установлен режим приоритетного обслуживания периферии, выполнение программы прерывается через установленные промежутки времени, выполняются выбранные процедуры обслуживания, после чего возобновляется выполнение программы. Этот цикл повторяется на протяжении всего выполнения программы. После выполнения входов/выходов также выполняется обычное обслуживание периферии.



Таким образом, режим приоритетного обслуживания периферии можно использовать, когда требуется периодическое обслуживание для указанных портов или модулей дополнительно к обычному обслуживанию периферии. Это удобно в тех случаях, когда обслуживание периферии является более приоритетной задачей по отношению к выполнению программы, например, в таких задачах управления процессами, в которых требуется быстрая реакция на внешние воздействия.

- Приоритетное обслуживание может быть выбрано максимум для 5 модулей или портов. Для идентификации модулей шины CPU и специальных модулей ввода/вывода CS/CJ используется номер модуля.
- В каждом временном окне обслуживания периферии обслуживается только один модуль или порт. Если обслуживание завершилось до истечения указанного времени, сразу же возобновляется выполнение программы, а следующий модуль или порт не обслуживается, пока не наступит следующее временное окно для обслуживания периферии. В то же время, возможна ситуация, когда один и тот же модуль или порт обслуживается несколько раз в одном и том же цикле.
- Модули или порты обслуживаются в том же порядке, в котором они были обнаружены модулем CPU.

**Примечание** 1. Даже если перечисленные ниже команды используют порты связи, они будут выполнены только один раз в пределах цикла выполнения, даже если используется режим приоритетного обслуживания периферии:

RXD(235) (RECEIVE)  
TXD(236) (TRANSMIT)

2. Если командой связи читается сразу несколько слов, согласованность читаемых данных не может быть гарантирована в случае применения режима приоритетного обслуживания периферии.
3. Если используется режим приоритетного обслуживания периферии, модулем CPU может быть превышена максимальная длительность цикла. Максимальная длительность цикла устанавливается в настройках ПЛК с помощью параметра «Длительность сторожевого цикла» (Watch Cycle Time). Если длительность цикла превысит значение длительности сторожевого цикла, будет установлен флаг «Слишком большая длительность цикла» (A40108) и работа ПЛК будет прервана. Если используется режим приоритетного обслуживания периферии, необходимо контролировать текущую длительность цикла (A264 и A265) и соответствующим образом скорректировать длительность сторожевого цикла (адрес: + 209). (Диапазон значений: 10 ... 40000 мс, шаг 10 мс, значение по умолчанию 1 с).

Настройки ПЛК

В настройках ПЛК необходимо сконфигурировать указанные ниже параметры в целях использования режима приоритетного обслуживания периферии.

- Временное окно для выполнения программы: 5 ... 255 мс, с шагом 1 мс
- Временное окно для обслуживания периферии: 0.1 ... 25.5 мс, с шагом 0.1 мс
- Модули и/или порты для приоритетного обслуживания:  
 модуль шины CPU (номер модуля)  
 специальный модуль ввода/вывода CS/CJ (номер модуля)  
 встраиваемая плата (только для серии CS)  
 порт RS-232C  
 периферийный порт

Адрес в консоли программирования		Значения	Значение по умолчанию	Функция	Момент вступления в силу
Слово	Бит(-ы)				
219	08 ... 15	00 05 ... FF (Hex)	00	00: Отмена режима приоритетного обслуживания 05 ... FF: Временное окно для выполнения команд (5 ... 255 мс, с шагом 1 мс)	Вступает в силу в начале работы (Нельзя изменить во время работы)
	00 ... 07	00 ... FF (Hex)	00	00: Отмена режима приоритетного обслуживания 01 ... FF: Временное окно для обслуживания периферии (0.1 ... 25.5 мс, с шагом 0.1 мс)	
220	08 ... 15	00 10 ... 1F 20 ... 2F E1 FC FD (Hex)	00	00: Отмена режима приоритетного обслуживания 10 ... 1F: Номер модуля шины CPU + 10 (Hex) 20 ... 7F: Номер специального модуля ввода/вывода серии CS/CJ + 20 (Hex)	
	00 ... 07		00	E1: Встраиваемая плата FC: Порт RS-232C FD: Периферийный порт	
221	08 ... 15		00	E1: Встраиваемая плата FC: Порт RS-232C	
	00 ... 07		00	FD: Периферийный порт	
222	08 ... 15		00		

- В следующей таблице указаны режимы работы и ошибки, которые имеют место в зависимости от значений параметров в настройках ПЛК.
- Настройка для модулей CPU CS1 или CJ1 не может быть выполнена с помощью CX-Programmer. Для модулей CPU CS1-H и CJ1-H можно выполнить настройку в CX-Programmer версии 2.1 или выше.

Условия			Режим работы модуля CPU	Ошибки настроек ПЛК	
Временное окно для обслуживания периферии	Временное окно для выполнения команд	Выбранные модули и порты			
01 ... FF: (0.1 ... 25.5 мс)	05 ... FF: (5 ... 255 мс)	Все знач. коррект. 00 и корректные значения	Режим приоритетного обслуживания периферии	Нет	
		Коррект., но излишние значения			
		Некоторые значения не корректны	Режим приоритетного обслуживания периферии для объектов с корректн. знач.	Генерируется	
		Все значения=00	Работа в обычном режиме	Генерируется	
		00 и недопустимые значения			
		Все знач. недопуст.			
00	00	---	Работа в обычном режиме	Нет	
Любые другие значения			---	Работа в обычном режиме	Генерируется

**Примечание** Если обнаружена ошибка в настройках ПЛК, включается бит A40210 и возникает нефатальная ошибка.

**Информация в дополнительной области** Если для выполнения программы и обслуживания периферии установлены временные окна, суммарная длительность всех временных окон для выполнения программы и обслуживания периферии записывается в A266 и A267. Эту информацию можно использовать для соответствующей корректировки временных окон.

Если режим приоритетного обслуживания периферии не используется, будет записано время выполнения программы. Это значение можно использовать для определения соответствующих значений для временных окон.

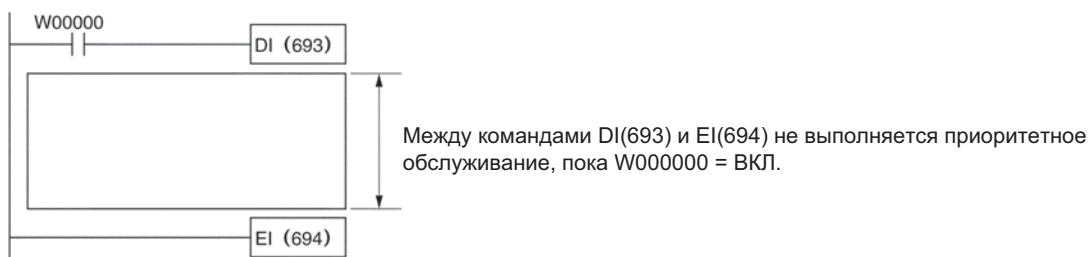
Слова	Содержание	Значение	Обновление					
A266 и A267	00000000 ... FFFFFFFF Hex (0 ... 4294967295 десятичн.)	Сумма всех временных окон для выполнения программы и всех временных окон для обслуживания периферии. 0.0 ... 429 496 729.5 мс (с шагом 0.1 мс)	Содержимое обновляется в каждом цикле и обнуляется в начале работы.					
		<table border="1"> <tr> <td>A267 (Старшие байты)</td> <td>A266 (Младшие байты)</td> <td rowspan="2">Значение записывается в формате 32-битового двоичного числа (8 разрядов Hex)</td> </tr> <tr> <td></td> <td></td> </tr> </table>	A267 (Старшие байты)	A266 (Младшие байты)	Значение записывается в формате 32-битового двоичного числа (8 разрядов Hex)			
A267 (Старшие байты)	A266 (Младшие байты)	Значение записывается в формате 32-битового двоичного числа (8 разрядов Hex)						

### 6-9-2 Временное отключение режима приоритетного обслуживания периферии

В случае использования режима приоритетного обслуживания периферии согласованность данных не может быть гарантирована в следующих ситуациях.

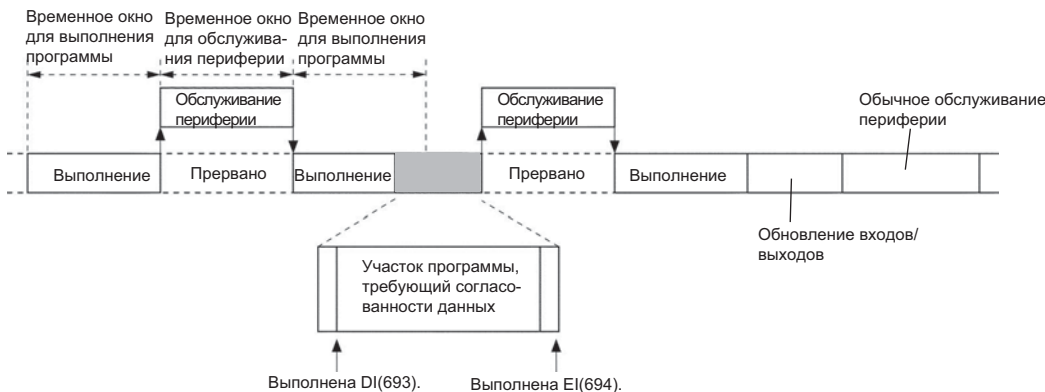
- Если с помощью команды связи из периферийного устройства читается одновременно несколько слов. Данные могут быть прочитаны за несколько различных временных окон обслуживания периферии, что может привести к их несогласованности.
- Если в программе применяются команды с большой длительностью выполнения, например, для передачи больших объемов данных памяти ввода/вывода. Операция передачи может быть прервана на время обслуживания периферии, что может привести к несогласованности данных. Это может произойти, если слова, записываемые программой, читаются из периферийного устройства до того, как была завершена их запись, либо когда слова, читаемые программой, записываются из периферийного устройства до того, как завершено их чтение.
- Если команды одновременно обращаются к одним и тем же словам в памяти. Если эти слова записываются из периферийного устройства в интервале между выполнением двух команд, две команды прочитают различные значения из памяти.

Если необходимо обеспечить согласованность данных, для модулей CPU CS1 или CJ1 можно использовать команды DISABLE INTERRUPTS и ENABLE INTERRUPTS (DI(693) и EI(694)), запрещающие приоритетное обслуживание в пределах требуемых участков программы (см. пример ниже). Для модулей CPU CS1 для однопроцессорных систем и для модулей CPU CS1-H, CJ1-H или CJ1M можно использовать команды DISABLE PERIPHERAL SERVICING и ENABLE PERIPHERAL SERVICING (IOSP(287) и IORS(288)).





Функционирование

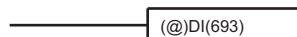


- Примечание**
1. Команды DI(693) и IOSP(287) запретят не только прерывания, предназначенные для обслуживания периферии, но также и все остальные прерывания, включая прерывания от входов/выходов, запланированные и внешние прерывания. Все сформированные прерывания будут обработаны после завершения выполнения циклической задачи (после выполнения END(001)), если перед этим не будет выполнена команда CLI(691) для сброса прерываний.
  2. Отмена прерываний с помощью DI(693) или IOSP(287) действует до выполнения EI(694) или IORS(288), пока не будет выполнена команда END(001) либо пока не будет остановлена работа ПЛК. Т.е., не могут быть выделены такие участки программы, которые выходят за границу задачи или цикла. Если необходимо запрещать прерывания в нескольких циклах или задачах, следует применять пару команд DI(693) и EI(694) или IOSP(287) и IORS(288) в каждой циклической задаче.

**Модули CPU CS1 и CJ1 DI(693)**

Выполненная команда DI(693) запрещает все прерывания (за исключением прерываний от выключения питания), в том числе прерывания для приоритетного обслуживания, прерывания от входов/выходов, запланированные прерывания и внешние прерывания. После выполнения DI(693) прерывания останутся запрещенными, если они уже были запрещены.

**Символьное обозначение**



**Поддерживаемые области программы**

Область	Применимость
Области программных блоков	Да
Области пошаговых программ	Да
Подпрограммы	Да
Задачи обработки прерываний	Нет

**Флаги условий**

Флаг	Обознач.	Описание
Флаг «Ошибка»	ER	Включается, если в пределах задачи обработки прерывания выполняется команда DI(693), и выключен во всех остальных случаях.

**EI(694)**

Выполненная команда EI(694) разрешает все прерывания (за исключением прерываний от выключения питания), в том числе прерывания для приоритетного обслуживания, прерывания от входов/выходов, запланированные прерывания и внешние прерывания. После выполнения EI(694) прерывания останутся разрешенными, если они уже были разрешены.

**Символьное обозначение**



**Поддерживаемые области программы**

Область	Применимость
Области программных блоков	Да
Области пошаговых программ	Да
Подпрограммы	Да
Задачи обработки прерываний	Нет

**Флаги условий**

Флаг	Обознач.	Описание
Флаг «Ошибка»	ER	Включается, если в пределах задачи обработки прерываний выполняется команда EI(694).

**Модули CPU CS1D для однопроцессорных систем и модули CPU CS1-H, CJ1-H и CJ1M****IOSP(287)**

Выполненная команда IOSP(287) запрещает обслуживание периферии. Обслуживание периферии останется запрещенным после выполнения IOSP(287), если оно уже было запрещено.

**Символьное обозначение**

————— IOSP

**Поддерживаемые области программы**

Область	Применимость
Области программных блоков	Да
Области пошаговых программ	Да
Подпрограммы	Да
Задачи обработки прерываний	Нет

**Флаги условий**

Флаг	Обознач.	Описание
Флаг «Ошибка»	ER	Включается, если в пределах задачи обработки прерывания выполняется команда IOSP(287), и выключен во всех остальных случаях.

**IORS(288)**

Выполненная команда IORS(288) разрешает обслуживание периферии, которое было запрещено командой IOSP(287). Обслуживание периферии останется разрешенным после выполнения команды IORS(288), если оно уже было разрешено.

**Символьное обозначение**

————— IORS

**Поддерживаемые области программы**

Область	Применимость
Области программных блоков	Да
Области пошаговых программ	Да
Подпрограммы	Да
Задачи обработки прерываний	Нет

**Флаги условий**

Флаг	Обознач.	Описание
Флаг «Ошибка»	ER	Включается, если в пределах задачи обработки прерывания выполняется команда IORS(288).

## 6-10 Работа без батареи

ПЛК серии CS и CJ могут работать без батареи (или с батареей с истекшим сроком службы). Функционирование ПЛК при работе без батареи зависит от следующих условий.

- Модуль CPU
- Сохраняется или не сохраняется содержимое памяти ввода/вывода (напр., область CIO)
- Инициализируются или не инициализируются при запуске области DM и EM
- Инициализируются или не инициализируются из программы пользователя области DM и EM

Различия в функционировании кратко описаны в следующей таблице.

Модуль CPU	Содержимое памяти ввода/вывода не сохраняется		Содержимое памяти ввода/вывода сохраняется
	Области DM и EM при запуске не инициализируются	Области DM и EM при запуске инициализируются	
		Из программы пользователя	Не из программы пользователя
CS1-H, CJ1-H, CJ1M или CS1D	Режим обычной работы (с использованием флеш-памяти или карты памяти).	Используется автоматическая загрузка из карты памяти при запуске (DIP-ключ 2 в полож. ВКЛ)	
CS1 или CJ1	Используется автоматическая загрузка из карты памяти при запуске (DIP-ключ 2 в полож. ВКЛ)		Сохранение невозможно ни одним из способов. Должна быть установлена батарея.

- Примечание**
1. В случае работы без батареи необходимо отключить функцию обнаружения пониженного напряжения батареи в настройках ПЛК (независимо от режима работы, используемого при работе без батареи).
  2. Если батарея не установлена или села, на работу модуля CPU накладываются следующие ограничения, справедливые для любого модуля CPU.
    - Состояние бита «Выходы ВЫКЛ» (A50015) не будет достоверным. Когда бит «Выходы ВЫКЛ» = ВКЛ, все выходы модуля вывода будут отключены. Чтобы все выходы модуля вывода не выключились при включении питания, в «лестничную диаграмму» необходимо вставить следующие команды.

Флаг «Первый цикл»  
(A20011)



- Содержимое памяти ввода/вывода (включая области HR, DM и EM) может быть сохранено некорректно. Поэтому параметры в настройках ПЛК следует установить таким образом, чтобы состояния флагов «Удержание памяти ввода/вывода» (A50012) и «Удержание принудительных состояний» (A50013) не сохранялись при включении питания.
- Нельзя использовать функцию часов. Показания часов в A351 ... A354 и время запуска в A510 и A511 не будут достоверными. Даты файлов, записываемых в карту памяти из модуля CPU, также не будут достоверными.
- Следующие данные будут нулевыми при запуске: Время включения питания (A523), Время выключения питания (A512 и A513) и Количество случаев отключения питания (A514).
- Протокол ошибок (A100 ... A199) не ведется.
- После запуска текущим всегда становится банк EM 0.
- Файлы в памяти файлов EM при запуске будут отсутствовать и функции, связанные с памятью файлов, использовать нельзя. В настройках ПЛК должен быть выбран сброс памяти файлов EM, перед использованием память файлов EM должна быть переформатирована.

**Модули CPU CS1-H, CJ1-H, CJ1M или CS1D**

Модули CPU CS1-H, CJ1-H, CJ1M или CS1D могут работать без батареи в обычном режиме. Программа пользователя и параметры автоматически сохраняются во флеш-память модуля CPU и автоматически восстанавливаются из флеш-памяти при запуске. В этом случае не сохраняется содержимое памяти ввода/вывода, и области DM и EM должны инициализироваться из программы пользователя.

В случае работы модулей CPU CS1-H, CJ1-H, CJ1M или CS1D без батареи также можно автоматически загружать данные из карты памяти при запуске, так же как и у модулей CPU CS1 (при наличии карты памяти также можно загружать данные областей DM и EM).

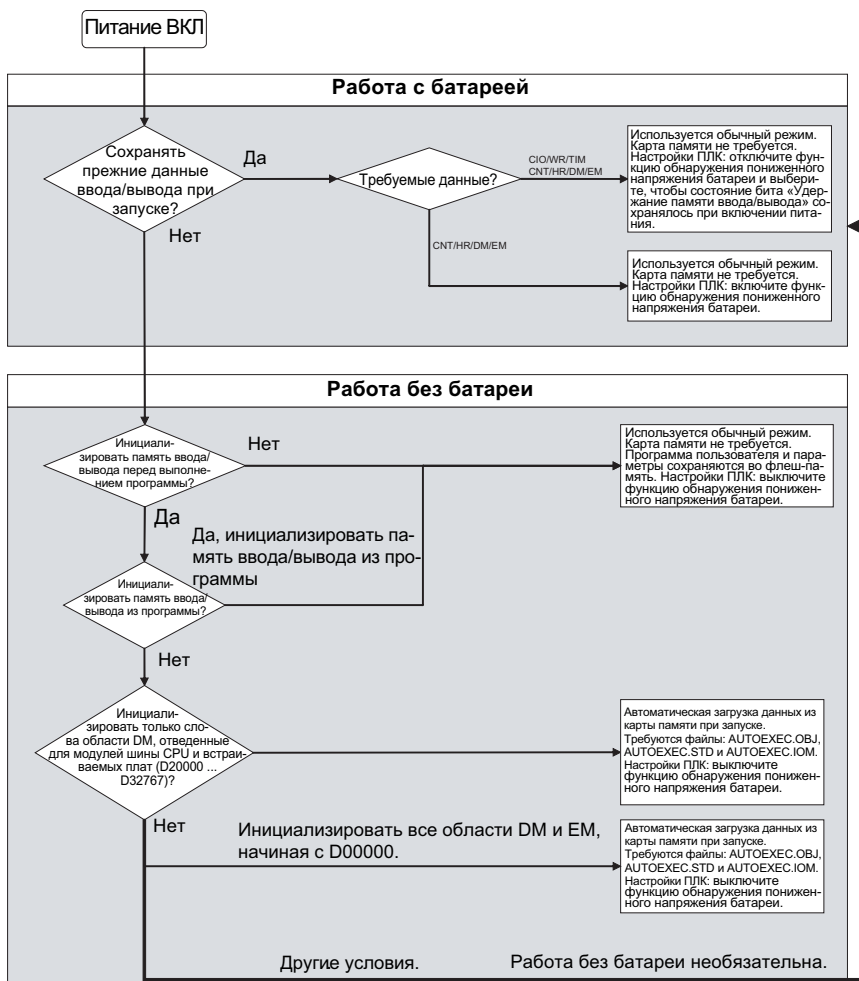
**Модули CPU CS1 и CJ1**

Модули CPU CS1 и CJ1 могут работать без батареи с применением функции автоматической загрузки данных из карты памяти при запуске. В этом случае содержимое памяти ввода/вывода не сохраняется (при наличии карты памяти можно загружать данные областей DM и EM).

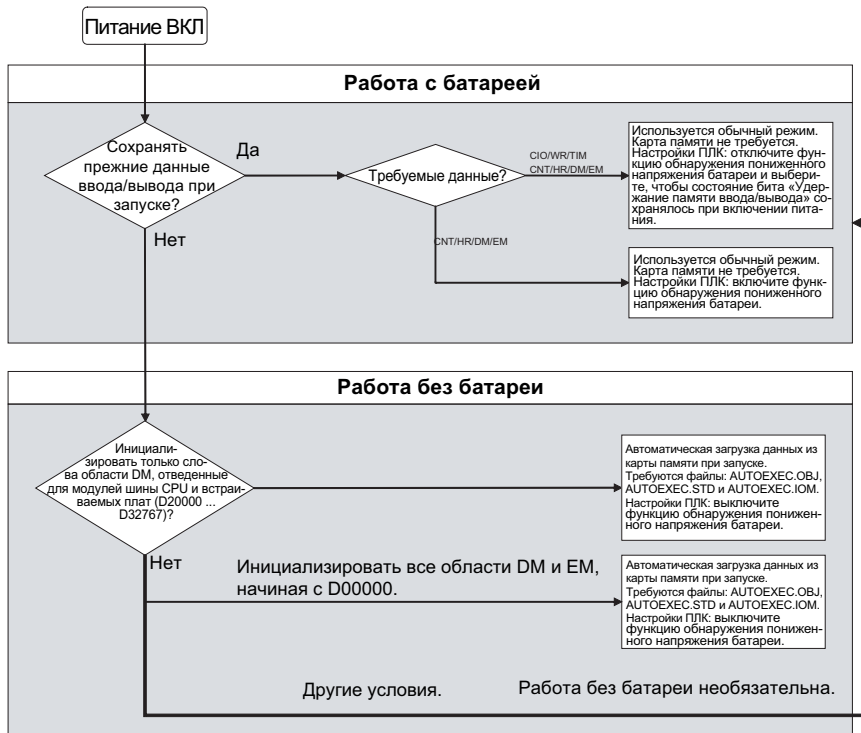
**Последовательность действий**

Ниже показаны диаграммы операций для модулей CPU двух типов.

**Модули CPU CS1-H, CJ1-H, CJ1M или CS1D**



Модули CPU CS1 и CJ1

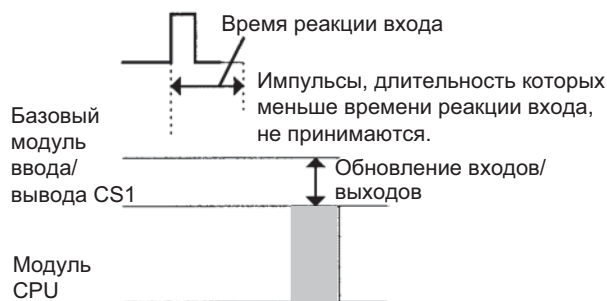
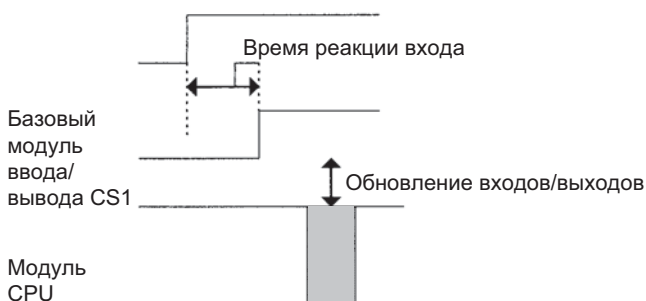


## 6-11 Прочие функции

### 6-11-1 Настройка времени срабатывания входов/выходов

Скорость реакции входов (время срабатывания) для базовых модулей ввода/вывода CS/CJ можно устанавливать, указывая номер стойки и номер слота (установочного места). Увеличение времени отклика (снижение быстродействия) позволяет уменьшить влияние дребезга и помех. Наоборот, уменьшение времени срабатывания входа (при сохранении длительности импульса большей, чем длительность цикла) можно применять для приема импульсов короткой длительности.

**Примечание** В случае модулей CPU серии CS для приема импульсов, длительность которых меньше длительности цикла, можно использовать скоростные входы, предусмотренные в некоторых модулях ввода/вывода C200H с высокой плотностью входов/выходов, или скоростной модуль ввода. Подробнее об этом написано в разделе 6-1-4 Скоростные входы.



**Настройки ПЛК**

Значения времени срабатывания для 80-ти слотов ПЛК CS/CJ (стойка 0, слот 0 ... стойка 7, слот 9) можно задать в 80-ти байтах (адреса 10 ... 49).

Адрес в консоли программирования	Название	Значения (Hex)	Значение по умолчанию (Hex)
10 Биты 0 ... 7	Время срабатывания входа базового модуля ввода/вывода CS/CJ (стойка 0, слот 0)	00: 8 мс 10: 0 мс 11: 0.5 мс 12: 1 мс 13: 2 мс 14: 4 мс 15: 8 мс 16: 16 мс 17: 32 мс	00 (8 мс)
:	:	:	:
49 Биты 8 ... 15	Время срабатывания входа базового модуля ввода/вывода CS/CJ (стойка 7, слот 9)	Те же, что и выше.	00 (8 мс)

**6-11-2 Размещение области ввода/вывода**

Резервируя область ввода/вывода для стоек расширения (стоек расширения CS/CJ и стоек расширения ввода/вывода C200H), можно выбирать расположение первого слова с помощью средства программирования. Благодаря этой функции для каждой стойки ввода/вывода можно выбирать свою собственную фиксированную область в диапазоне CIO 0000 ... CIO 0999 (первые слова привязываются к номерам стоек).



## РАЗДЕЛ 7

# Загрузка программы, пробный запуск и отладка

В данном разделе описаны операции, выполняемые для загрузки программы в модуль CPU, а также функции, которые можно использовать для тестирования и отладки программы.

7-1	Загрузка программы . . . . .	334
7-2	Пробный запуск и отладка . . . . .	334
7-2-1	Принудительная установка/сброс битов. . . . .	334
7-2-2	Мониторинг переключения состояний. . . . .	335
7-2-3	Online-редактирование. . . . .	336
7-2-4	Протоколирование данных. . . . .	339



## 7-1 Загрузка программы

Загрузка программ, настроек ПЛК, данных памяти ввода/вывода и комментариев к входам/выходам в модуль CPU, находящийся в режиме PROGRAM, осуществляется с помощью средства программирования.

### Процедура загрузки программ с помощью CX-Programmer

- 1,2,3...
1. Выберите **PLC, Transfer** (ПЛК, Загрузить), а затем **To PLC** (В ПЛК). Отобразится диалоговое окно **Download Options** (Варианты загрузки).
  2. Выберите объекты, которые должны быть загружены, из следующего списка: программы, настройки (настройки ПЛК), таблица ввода/вывода, символы, комментарии и индекс (указатель) программы.  
Примечание: таблицу ввода/вывода и комментарии можно выбрать только в том случае, если они имеются в карте памяти в модуле CPU.
  3. Щелкните по кнопке **ОК**.

Загрузку программы можно выполнить одним из следующих способов.

- Автоматическая загрузка после включения питания

Когда включается питание, из карты памяти в модуль CPU считывается файл AUTOEXEC.OBJ (ключ 2 DIP-переключателя должен быть в положении ВКЛ).

- Замена программы во время работы

Существующий файл программы можно заменить файлом программы, который указан в дополнительной области. Для этого в программе следует включить бит "Начать замену" (A65015) в дополнительной области во время работы модуля CPU. Подробное описание смотрите в *Разделе 5 Функции памяти файлов*.

**Примечание** Если используется CX-Programmer версии 4.0 или выше для модулей CPU серии CS/CJ версии 2.0 или выше, программы задач можно загружать по отдельности. Подробное описание смотрите в руководствах *CS Series PLCs Operation Manual* или *CJ Series PLCs Operation Manual, Раздел 1-4-1 Загрузка и считывание отдельных задач*.

## 7-2 Пробный запуск и отладка

### 7-2-1 Принудительная установка/сброс битов

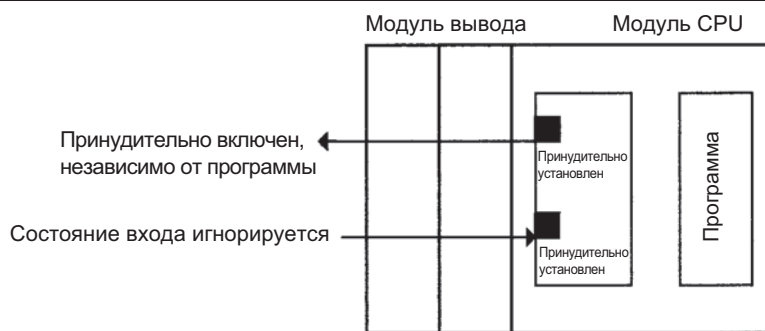
С помощью средства программирования можно принудительно устанавливать (ВКЛ) или сбрасывать (ВыКЛ) отдельные биты (область CIO, дополнительная область, область HR и флаги завершения таймеров/счетчиков). Состояние, вызванное принудительно, обладает приоритетом по отношению к состоянию, которое получается в результате выполнения программы или обновления входов/выходов. Это состояние нельзя изменить с помощью команд, оно сохраняется независимо от состояния программы или внешних входов до тех пор, пока оно не будет отменено с помощью средства программирования.

Операции принудительной установки/сброса предназначены для принудительного изменения состояний входов и выходов во время пробного запуска или для принудительного выполнения некоторых условий при отладке.

Операции принудительной установки/сброса можно выполнять в режимах MONITOR или PROGRAM, но не в режиме RUN.

**Примечание** Чтобы при переключении режима работы сохранялось состояние битов, которые были принудительно установлены или сброшены, следует одновременно включить бит "Удержание принужденных состояний" (A50013) и бит "Удержание IOM" (A50012).

Чтобы состояние битов, которые были принудительно установлены или сброшены, сохранялось после выключения и повторного включения питания, необходимо сохранять состояние бита "Удержание принужденных состояний". Для этого необходимо включить бит "Удержание принужденных состояний" (A50013) и бит "Удержание IOM" (A50012), а также настроить параметр в настройках ПЛК, отвечающий за бит "Удержание принужденных состояний" во время запуска.



Принудительно устанавливать и сбрасывать можно биты следующих областей.

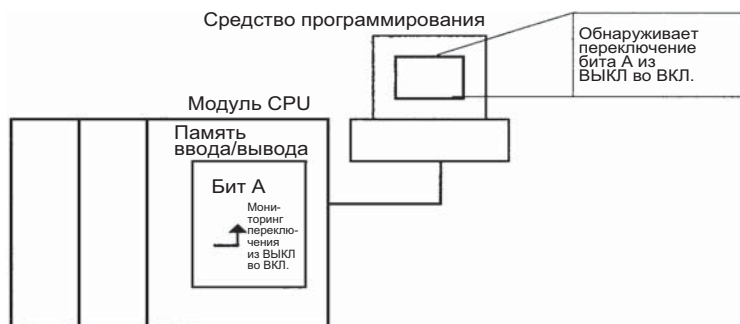
СIO (биты ввода/вывода, биты логических связей (data link), биты модулей шины CPU, биты специальных модулей ввода/вывода, биты встраиваемой платы, биты SYSMAC BUS, биты оптических модулей ввода/вывода, рабочие биты), область WR, флаги завершения таймеров, область HR, флаги завершения счетчиков (области встраиваемой платы, SYSMAC BUS и терминалов ввода/вывода поддерживаются только модулями CPU серии CS).

**Операции, выполняемые с помощью средств программирования**

- Выбор битов для принудительной установки/сброса.
- Выбор принудительной установки или принудительного сброса.
- Сброс принудительного состояния (включая одновременный сброс всех принудительных состояний).

**7-2-2 Мониторинг переключения состояний**

Когда модуль CPU определяет, что бит, установленный с помощью средства программирования, перешел из ВЫКЛ во ВКЛ или из ВКЛ в ВЫКЛ, результат этой смены состояний сигнализируется флагом "Обнаружена смена состояния" (A50809). Этот флаг включится, когда будут выполнены условия, установленные для контроля смены состояний. С помощью средства программирования можно контролировать и отображать эти результаты на экране.



**Операции, выполняемые с помощью CX-Programmer**

- 1,2,3...
1. Щелкните правой кнопкой мыши по биту, переключение состояния которого требуется контролировать.
  2. Щелкните в меню PLC (ПЛК) по пункту **Differential Monitor (Мониторинг переключения состояний)**. Будет отображено диалоговое окно Differential Monitor (Мониторинг переключения состояний). Щелкните по **Rising** (Включение) или **Falling** (Выключение).
  3. Щелкните по кнопке **Start** (Начать). В случае обнаружения выбранного переключения состояний прозвучит тональный сигнал и произойдет приращение счетчика.
  4. Щелкните по кнопке **Stop** (Завершить). Мониторинг переключения состояний будет завершен.

**Сопутствующие биты/слова дополнительной области**

Название	Адрес	Описание
Флаг "Обнаружена смена состояния"	A50809	Включается, если было выполнено условие, выбранное для обнаружения переключения состояний. Примечание: флаг обнуляется в начале мониторинга переключения состояний.

**7-2-3 Online-редактирование**

Функция online-редактирования предназначена для внесения дополнений или изменений в часть программы модуля CPU непосредственно из средства программирования, когда модуль CPU находится в режиме MONITOR или PROGRAM. Консоль программирования позволяет добавлять или изменять одновременно только одну команду, а с помощью CX-Programmer можно добавлять или изменять одновременно одну или несколько частей программы. Таким образом, эта функция предназначена для внесения незначительных изменений в программу без прекращения работы модуля CPU.

Online-редактирование можно одновременно осуществлять с нескольких компьютеров с работающим пакетом CX-Programmer и одновременно с консоли программирования, если редактируются различные задачи.



Если программа модуля CPU редактируется во время работы в режиме MONITOR, продолжительность цикла может возрасти в два и более раз.

Продолжительность цикла для модулей CPU CS1-H, CJ1-H, CJ1M и CS1D также возрастет за счет процедуры резервного сохранения данных во флеш-память после online-редактирования. В этот момент светится индикатор ВКУР. Ход продвижения процедуры резервного сохранения отображается в CX-Programmer. Степень увеличения продолжительности цикла показана в следующей таблице

Модуль CPU	Увеличение длительности цикла	
	Online-редактирование	Резервное сохранение во флеш-память
Модули CPU CS1 до EV1	макс. 90 мс	Не поддерживается.
Модули CPU CS1 EV1 или новее	макс. 12 мс	4% или длительность цикла
Модули CPU CS1-H		
Модули CPU CS1D		
Модули CPU CS1		Не поддерживается.
Модули CPU CJ1-H		4% или длительность цикла
Модули CPU CJ1M		

У модулей CPU CS1-H, CJ1-H, CJ1-M или CS1D количество исправлений, выполняемых последовательно друг за другом, ограничено. Фактически количество зависит от вносимых изменений, но в качестве ориентира можно использовать следующие данные.

CJ1M-CPU□□:	40 правок
CS1G-CPU□□H/CJ1G-CPU□□H:	160 правок
CS1H-CPU□□H/CJ1H-CPU□□H/CS1D-CPU□□H/ CS1D-CPU□□S:	400 правок

В случае превышения предельного количества исправлений на экране CX-Programmer или на консоли программирования будет отображено сообщение и дальнейшее редактирование будет невозможно до тех пор, пока модуль CPU не завершит резервное сохранение данных.

### Размер задачи и увеличение длительности цикла

Существует следующая взаимосвязь между объемом редактируемой задачи и возрастанием длительности цикла:

В случае использования модулей CPU CS1, CPU CS1-H, CPU CS1D, CPU CJ1 или CPU CJ1M (все модули версии 1 или выше) время, на которое возрастает длительность цикла из-за редактирования в режиме online, практически не зависит от объема редактируемой задачи (программы).

В случае использования модуля CPU CS1 до версии EV1 объем редактируемой задачи влияет на время, на которое приостанавливается выполнение программ для целей online-редактирования. Разбив программу на более мелкие задачи, можно сократить время, на которое возрастает продолжительность цикла из-за редактирования в online-режиме, по сравнению с ПЛК предыдущих моделей.

**Замечания** Длительность цикла будет больше по сравнению с обычной длительностью, если программа перезаписывается в процессе online-редактирования в режиме MONITOR, поэтому необходимо следить за тем, чтобы время, на которое увеличивается длительность цикла, не превышало время контроля цикла, установленное в настройках ПЛК. Если это время превышает время контроля, возникнет ошибка "Превышение длительности цикла" и модуль CPU остановится. В этом случае следует перезапустить модуль CPU, выбрав режим PROGRAMM, прежде чем перейти в режим RUN или MONITOR.

**Примечание** Если задача, редактируемая в режиме online, содержит программный блок, в этом случае данные предшествующего выполнения, например, статус "Ожидание" (WAIT) или "Пауза" будут обнулены при online-редактировании и следующее выполнение будет начато сначала.

### Online-редактирование (CX-Programmer)

- 1,2,3...**
1. Отобразите редактируемую часть программы.
  2. Выберите редактируемую команду.
  3. Выберите **Program, Online Edit** (Программа, Online-редактирование), затем **Begin** (Начать).
  4. Выполните необходимое редактирование.
  5. Выберите **Program, Online Edit**, а затем **Send Changes** (Отправить изменения). Будет выполнена проверка команд, и в случае отсутствия ошибок команды будут загружены в модуль CPU. Команды, имеющиеся в модуле CPU, будут перезаписаны, в этот момент длительность цикла возрастет.



**Предупреждение** Прежде чем приступить к Online-редактированию, обязательно убедитесь в том, что увеличение длительности цикла не повлияет негативно на работу системы. Если длительность цикла будет слишком большой, входные сигналы могут перестать считываться.

#### Временное прекращение online-редактирования

Чтобы проверить работу системы (отклик, управление оборудованием) в определенном цикле, для этого цикла можно отключить online-редактирование. Online-редактирование с помощью средства программирования будет отменено на один цикл, и все запросы online-редактирования, полученные в течение этого цикла, будут отложены до следующего цикла.

Online-редактирование отключается путем включения бита "Отключение online-редактирования" (A52709) и выбора значения 5A для слов A52700... A52707 (Действительность бита отключения online-редактирования). Когда эти настройки выполнены и получен запрос на online-редактирование, online-редактирование переводится в режим ожидания и устанавливается флаг "Online-редактирование в режиме ожидания" (A20110).

Когда бит "Отключение online-редактирования" (A52709) сбрасывается, online-редактирование становится доступно, флаг "Выполнение online-редактирования" (A20111) устанавливается, а флаг "Online-редактирование в режиме ожидания" (A20111) сбрасывается. По завершению online-редактирования флаг "Выполнение online-редактирования" (A20111) сбрасывается.

Online-редактирование также можно временно отключить, установив бит "Отключение online-редактирования" (A52709) во время online-редактирования. В этом случае также устанавливается флаг "Online-редактирование в режиме ожидания" (A20110).

Если поступает второй запрос на online-редактирование, когда уже имеется один отложенный запрос, второй запрос не запоминается и происходит ошибка.

Online-редактирование также можно отключать для предотвращения случайного внесения изменений в программу в режиме online-редактирования. Как уже было указано выше, для этого следует установить бит "Отключение online-редактирования" (A52709) и записать в слова A52700... A52707 (Достоверность бита отключения online-редактирования) значение 5A.

**Разрешение online-редактирования с помощью средства программирования**

Если online-редактирование нельзя разрешить из программы, это можно сделать из CX-Programmer.

**1,2,3...** 1. Выполнение online-редактирования с помощью консоли программирования

Если online-редактирование выполняется с помощью консоли программирования и состояние ожидания для online-редактирования сбросить не удастся, консоль программирования будет заблокирована и никаких операций с ее помощью выполнить будет нельзя.

В этом случае к другому последовательному порту следует подключить CX-Programmer, после чего сбросить бит "Отключение online-редактирования" (A52709). Online-редактирование станет доступно, и вновь можно будет выполнять операции с помощью консоли программирования.

2. Выполнение online-редактирования с помощью CX-Programmer

Если работа продолжается, когда online-редактирование находится в состоянии ожидания, связь между CX-Programmer и ПЛК может разорваться. В этом случае следует переподключить компьютер к ПЛК и сбросить бит "Отключение online-редактирования" (A52709).

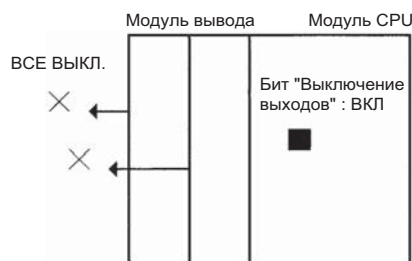
**Сопутствующие биты/слова дополнительной области**

Название	Адрес	Описание
Достоверность бита отключения online-редактирования	A52700 ... A52707	Оценивает состояние бита "Отключение online-редактирования" (A52709). не 5A: бит "Отключение online-редактирования" недействителен 5A: бит "Отключение online-редактирования" действителен
Бит "Отключение online-редактирования"	A52709	Чтобы отключить online-редактирование, следует включить этот бит и записать в слова A52700...A52707 (Достоверность бита отключения online-редактирования) значение 5A.
Флаг "Online-редактирование в режиме ожидания"	A20110	Состояние ВКЛ, когда online-редактирование находится в режиме ожидания из-за того, что оно отключено.
Флаг "Выполнение online-редактирования"	A20111	Состояние ВКЛ, когда online-редактирование доступно (выполняется).

**Выключение выходов**

Если бит "Отключение выходов" (A50015) устанавливается с помощью команды OUT или с помощью средства программирования, в этом случае все выходы всех модулей вывода будут переведены в состояние ВЫКЛ (это также относится ко встроенным выходам общего назначения или импульсным выходам модулей CPU CJ1M) и включится индикатор INH на передней панели модуля CPU.

Состояние бита "Отключение выходов" сохраняется даже после выключения и повторного включения питания.



### 7-2-4 Протоколирование данных

Функция протоколирования данных (Data Trace) создает выборку указанных данных памяти ввода/вывода, используя для синхронизации любой из перечисленных ниже методов, и сохраняет отобранные данные в памяти протокола, из которой их можно затем прочитать и проанализировать с помощью средства программирования.

- Чтение с заданным периодом (10 ... 2 550 мс с шагом 10 мс)
- Одно чтение за цикл
- По выполнению команды TRACE MEMORY SAMPLING (TRSM)

Протоколирование можно производить одновременно для максимум 31 бита и 6 слов памяти ввода/вывода. Объем памяти протокола составляет 4000 слов.

#### Основная последовательность действий

- 1,2,3...
1. Отбор значений начинается после того, как установлены параметры с помощью CX-Programmer и выполнена команда запуска протоколирования.
  2. Считанные данные (после выполнения действия 1) протоколируются, когда выполняется стартовое условие протоколирования. Данные записываются в память протокола с задержкой (см. Примечание 1).
  3. Данные в памяти протокола будут считаны и протоколирование завершится.

**Примечание** Величина задержки: определяет, как много периодов считывания следует пропустить после включения бита "Запуск протоколирования" (A50814), прежде чем выполнить запись в память протокола. Допустимые диапазоны значений перечислены в следующей таблице.

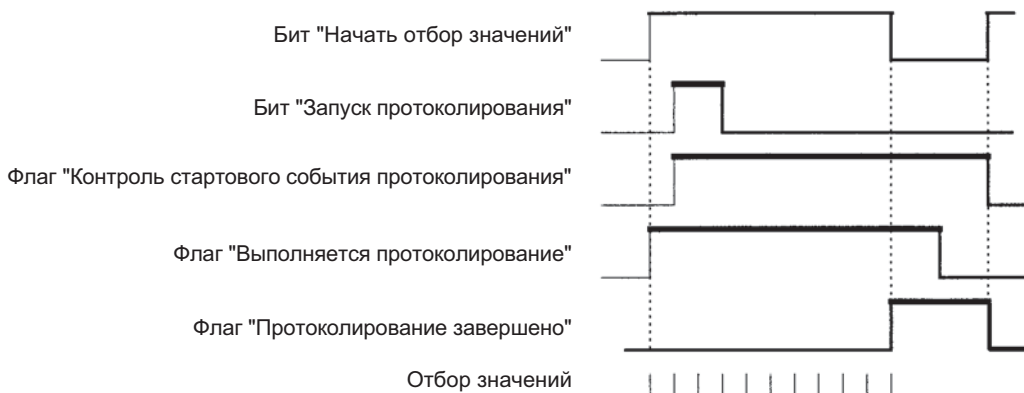
Количество протоколируемых слов	Диапазон величины задержки
0	-1999 ... 2000
1	-1332 ... 1333
2	-999 ... 1000
3	-799 ... 800
4	-665 ... 666
5	-570 ... 571
6	-499 ... 500

Положительная задержка: данные сохраняются с установленной задержкой.

Отрицательная задержка: сохраняются предшествующие данные в соответствии с установленной задержкой.

**Пример:** Задержка -30 мс в случае протоколирования с периодом 10 мс означает задержку  $-30 \times 10 = 300$  мс, т.е., регистрироваться будут данные, присутствовавшие за 300 мс до наступления события считывания.

**Примечание** Бит "Начать отбор значений" (A50815) должен включаться с помощью средства программирования. Ни в коем случае не включайте этот бит из программы пользователя.



Можно выбрать один из следующих протоколов.

#### Циклическое протоколирование данных

Циклическое протоколирование данных означает, что данные считываются через фиксированные интервалы времени. Значение интервала считывания можно выбрать в пределах от 10 до 2500 мс с шагом 10 мс. В программе пользователя не следует применять команду TRSM, а интервал должен быть выбран отличным от 0 мс.

#### Однократное протоколирование данных за цикл

Однократное протоколирование данных за цикл означает, что считываются и протоколируются результаты обновления входов/выходов после завершения всех задач полного цикла. В программе пользователя не следует применять команду TRSM, а интервал должен быть выбран отличным от 0 мс.

#### Протоколирование данных с помощью TRSM

Считывание и протоколирование выполняются однократно после выполнения команды TRASE MEMORY SAMPLING (TRSM). Если в программе использовано несколько команд TRSM, протоколирование происходит всякий раз, когда выполняется команда TRSM после наступления стартового условия протоколирования.

### Последовательность действий для протоколирования данных

Чтобы создать протокол данных, выполните следующие действия.

- 1,2,3... 1. С помощью CX-Programmer настройте параметры протокола (*PLC/Data Trace (ПЛК/Протокол данных)*), закладка *Execute/Set (Выполнить/Настроить)*: адрес протоколируемых данных, интервал отбора значений, время задержки и условия запуска.
2. С помощью CX-Programmer запустите процедуру считывания или включите бит "Начать отбор значений" (A50815).
3. По мере необходимости активизируйте стартовое условие протоколирования.
4. Завершите протоколирование.
5. С помощью CX-Programmer прочитайте данные протокола.
  - a) Выберите **Data Trace** (Протокол данных) в меню **PLC** (ПЛК).
  - b) Выберите **Select** (Выбрать) в меню **Execution** (Выполнение).
  - c) Выберите **Execute** (Выполнить) в меню **Execution** (Выполнение).
  - d) Выберите **Read** (Прочитать) в меню **Execution** (Выполнение).

## Сопутствующие биты/слова дополнительной области

Название	Адрес	Описание
Бит "Начать отбор значений"	A50815	С помощью средства программирования установите (ВКЛ) этот бит, чтобы начать отбор значений. Этот бит следует включать только с помощью средства программирования. Не включайте и не сбрасывайте этот бит из программы пользователя. Примечание: бит будет сброшен по завершению протоколирования данных.
Бит "Запуск протоколирования"	A50814	Когда установлен (ВКЛ) этот бит, контролируется условие запуска протоколирования, и считанная выборка данных записывается в память протокола, когда выполняется условие запуска. С помощью этого бита активизируются следующие протоколы: 1) Циклический протокол (протоколирование с фиксированным периодом от 10 до 2550 мс) 2) Протоколирование по команде TRSM 3) Однократное протоколирование за цикл (протоколирование после завершения всех циклических задач)
Флаг "Контроль стартового условия протоколирования"	A50811	Этот флаг устанавливается (ВКЛ), если удовлетворено условие запуска протоколирования после включения бита "Запуск протоколирования". Этот флаг сбросится (ВЫКЛ), когда протоколирование будет запущено вновь путем включения бита "Начать отбор значений".
Флаг "Выполняется протоколирование"	A50813	Этот флаг устанавливается (ВКЛ), когда протоколирование запускается битом "Начать отбор значений", и сбрасывается (ВЫКЛ), когда протоколирование завершается.
Флаг "Протоколирование завершено"	A50812	Этот флаг устанавливается (ВКЛ), если после выполнения условия запуска протоколирования в процессе протоколирования переполняется память протокола, и сбрасывается (ВЫКЛ), когда запускается следующая операция протоколирования.





# Приложение А

## Таблицы сравнения ПЛК:

### ПЛК серии CJ, серии CS, C200HG/HE/НХ, CQM1H, CVM1 и серии CV

#### Сравнение по функциям

Параметр		Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H
Базовые свойства	Емкость					
	Кол-во точек ввода/вывода	2560 точек	5120 точек	1184 точки	6144 точки	512 точек
	Емкость программы	120 000 шагов 1 шаг, как правило, эквивалентен одному слову. Описание смотрите в конце раздела 10-5 <i>Времена исполнения команд и количество шагов в руководстве Operation Manual.</i>	250 000 шагов 1 шаг, как правило, эквивалентен одному слову. Описание смотрите в конце раздела 10-5 <i>Времена исполнения команд и количество шагов в руководстве Operation Manual.</i>	2000 слов (63200 слов для - Z)	62000 слов	15200 слов
	Макс. память данных	32000 слов	32000 слов	6000 слов	24000 слов	6000 слов
	Кол-во битов ввода/вывода	160 слов (2560 битов)	320 слов (5120 битов)	40 слов (640 битов)	120 слов (2048 битов)	32 слова (512 битов)
	Кол-во рабочих битов	2644 слова (42304 бита) + WR: 512 слов (8192 бита) = 3156 слов (50496 битов)	2644 слова (42304 бита) + WR: 512 слов (8192 бита) = 3156 слов (50496 битов)	408 слов (6528 битов)	168 слов (2688 битов) + 400 слов (6400 битов)	158 слов (2528 битов)
	Кол-во битов удержания	512 слов (8192 бита)	512 слов (8192 бита)	100 слов (1600 битов)	300 слов (4800 битов) Макс.: 1400 слов (2400 битов)	100 слов (1600 битов)
	Макс. расширенная память данных	32000 слов x 7 банков	32000 слов x 13 банков	6000 слов x 3 банка (6000 слов x 16 банков для - Z)	32000 слов x 8 банков (опция)	6000 слов
	Макс. кол-во таймеров/счетчиков	По 4096 каждого	По 4096 каждого	Комбинированных таймеров/счетчиков: 512	1024 точки	Комбинированных таймеров/счетчиков: 512
	Скорость выполнения	Базовые команды (LD)	CJ1: 0.08 мкс мин. CJ1-H: 0.02 мкс мин. CJ1M: 0.1 мкс мин.	CS1: 0.04 мкс мин. CS1-H: 0.02 мкс мин.	0.104 мкс мин.	0.125 мкс мин.
Специальные команды (MOV)		CJ1: 0.25 мкс мин. CJ1-H: 0.18 мкс мин. CJ1M: 0.3 мкс мин.	CS1: 0.25 мкс мин. CS1-H: 0.18 мкс мин.	0.417 мкс мин.	4.3 мкс мин.	17.7 мкс мин.
Время на собственные нужды системы		CJ1: 0.5 мс мин. CJ1-H: 0.3 мс мин. в обычном режиме, 0.2 мс в режиме параллельного выполнения CJ1M: 0.5 мс мин.	CS1: 0.5 мс мин. CS1-H: 0.3 мс мин. в обычном режиме, 0.2 мс в режиме параллельного выполнения	0.7 мс	0.5 мс	0.7 мс
Задержка во время online-редактирования (записи)		CJ1: приближ. 12 мс CJ1-H: приближ. 11 мс для CPU4 □ и 8 мс для CPU6 CJ1M: приближ. 14 мс	CS1: приближ. 12 мс CS1-H: приближ. 11 мс для CPU4 □ и 8 мс для CPU6 CJ1M: приближ. 14 мс	80 мс (160 мс для - Z)	500 мс	Типовое значение 250 мс

Параметр		Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H
Конструкция	Винтовое крепление	Нет	Да	Да	Да	Нет
	Установка на DIN-рейку	Да	Да	Да	Нет	Да
	Задние шины	Нет	Да	Да	Да	Нет
	Размер (В x Г, мм)	90 x 65	130 x 123	130 x 118	250 x 100	110 x 107
Количество модулей/стоек	Модули ввода/вывода	40 модулей	89 модулей (включая стойки ведомых устройств)	10 или 16 модулей	64 модуля (8 стоек x 8 модулей)	16 модулей
	Модули шины CPU	16 модулей	16 модулей	Нет	16 модулей	Нет
	Стойки расширения ввода/вывода	3 стойки	7 стоек	3 стойки	7 стоек	1 стойка
Поддержка задач		Да	Да	Нет	Нет	Нет
Режим работы CPU (выполнение программы и обслуживание периферии)	Обычный режим	Да	Да	---	---	---
	Режим приоритетного обслуживания периферии	Да	Да	---	---	---
	Параллельное управление с синхронным доступом к памяти	CJ1: Нет CS1-H: Да CJ1M: Нет	CS1: Нет CS1-H: Да	Нет	Нет	Нет
	Параллельное управление с асинхронным доступом к памяти	CS1: Нет CJ1-H: Да CJ1M: Нет	CS1: Нет CS1-H: Да	Нет	Нет	Нет
Режим обновления входов/выходов	Циклическое обновление	Да	Да	Да	Да	Да
	Запланированное обновление	Нет	Нет	Нет	Да	Нет
	Обновление с переходом через 0	Нет	Нет	Нет	Да	Нет
	Мгновенное обновление	Да	Да	Нет	Да	Нет
	Мгновенное обновление с помощью команды IORF	Да	Да	Да	Да	Да
Функция часов		Да	Да	Да	Да	Да (требуется кассета памяти)
Выход RUN (выполнение)		Да (зависит от модуля питания)	Да (зависит от модуля питания)	Да (зависит от модуля питания)	Да	Нет
Режим запуска (для принимаемых по умолчанию настроек ПЛК, когда не подключена консоль программирования)		Режим RUN	CS1: режим PROGRAM CS1-H: режим RUN	Режим RUN	Режим RUN	Режим PROGRAM
Запрет обработки прерываний по питанию		CJ1: Нет CJ1-H: Да CJ1M: Да	CS1: Нет CS1-H: Да	Нет	Нет	Нет
Работа без батарей		CJ1: карта памяти CJ1-H: карта памяти или флеш-память CJ1M: карта памяти или флеш-память	CS1: карта памяти CS1-H: карта памяти или флеш-память	Карта памяти	Карта памяти	Кассета памяти
Автоматическое резервное сохранение во флеш-память		CJ1: Нет CJ1-H: Да CJ1M: Да	CS1: Нет CS1-H: Да	Нет	Нет	Нет
Продолжение работы после перезапуска		Нет	Нет	Нет	Да	Нет

Параметр		Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H	
Внешняя память	Носитель	Карта памяти (флеш-ПЗУ)	Карта памяти (флеш-ПЗУ)	Кассета памяти (EEPROM, EPROM)	Кассета памяти (ОЗУ, EEPROM, EPROM)	Кассета памяти (ОЗУ, EEPROM, EPROM)	
	Емкость	48 Мбайт	48 Мбайт	4000 ... 32000 слов (4000 ... 64000 слов для - Z)	32000 ... 512000 слов (ОЗУ: 64 ... 512 Кбайт, EEPROM: 64 ... 128 Кбайт, EPROM: 0,5 ... 1 Мбайт)	4000 ... 16000 слов	
	Содержание	Программы, память ввода/вывода, параметры	Программы, память ввода/вывода, параметры	Программы, память ввода/вывода, параметры	Программы, память ввода/вывода, параметры	Программы, область DM (только для чтения), параметры	
	Способ чтения/записи	Средство программирования, программа пользователя (команды для работы с памятью файлов) и Host Link	Средство программирования, программа пользователя (команды для работы с памятью файлов) и Host Link	Включение бита SR	Средство программирования, программа пользователя (команды для работы с памятью файлов), Host Link или прибор записи в карту памяти	Включение бита AR	
	Формат файлов	Двоичный	Двоичный	Двоичный	Двоичный	Двоичный	
	Расширенная память данных обрабатывается в качестве файлов	Да (кроме модулей CPU CJ1M)	Да	Нет	Нет	Нет	
	Программы автоматически загружаются при запуске	Да	Да	Да	Да	Да	
Встраиваемая плата		Нет	Плата последовательного интерфейса	Плата связи	Нет	Плата связи	
Встроенные последовательные порты		Да (RS-232C x 1)	Да (RS-232C x 1)	Да (RS-232C x 1)	Да (RS-232C x 1) или RS-422 x 1)	Да (RS-232C x 1)	
Последовательные интерфейсы	Периферийный порт	Периферийная шина	Да	Да	Да	Да	Да
		Host Link (SYSMAC WAY)	Да	Да	Да	Нет (возможно при подключении к периферийному интерфейсу)	Да
		Без протокола	Нет	Нет	Да	Нет	Да
		NT Link	Да	Да	Нет	Нет	Нет
	Порт RS-232C, встроенный в модуль CPU	Периферийная шина	Да	Да	Да	Нет	Нет
		Host Link (SYSMAC WAY)	Да	Да	Да	Да	Да
		Без протокола	Да	Да	Да	Нет	Да
		NT Link	Да (1:N)	Да (1:N)	Да	Нет	Да (1:1)
		Каналы последовательной связи с ПЛК	Да (только CJ1M)	Нет	Нет	Нет	Нет
	Порт RS-232C или RS-422/RS-485 платы связи	Периферийная шина	Нет	Нет	Да	Нет	Нет
		Host Link (SYSMAC WAY)	Нет	Да Команды WG, MP и CR не поддерживаются.	Да Команда CR не поддерживается.	Да Команды WG и MP не поддерживаются.	Да Команда CR не поддерживается.
		Без протокола	Нет	Нет	Да	Нет	Да
		NT Link	Нет	Да	Да	Нет	Да (1:1 и 1:N)
		Protocol macro	Нет	Да	Да	Нет	Да
		CompoWay/F Master	Нет	Да (с помощью Protocol macro)	Да (с помощью Protocol macro)	Нет	Да (с помощью Protocol macro)

Параметр		Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H
Прерывания	Прерывания от входов/выходов	Да (макс. 2 модуля ввода прерываний: 32 точки, плюс 4 точки для встроенных входов/выходов модулей CPU CJ1M) (модули CPU CJ1 не поддерживают прерывания от входов/выходов).	Да (макс. 4 или 2 модуля ввода прерываний: 32 точки)	Да (макс. 2 модуля ввода прерываний: 16 точек)	Да (макс. 4 модуля ввода прерываний: 32 точки)	Да (4 встроенных в модуль шины CPU)
	Запланированные прерывания	Да	Да	Да	Да	Да
	Одновибратор с запуском по прерыванию	Нет	Нет	Нет	Нет	Да
	Прерывания от входов в режиме счетчика	Да (только модули CPU CJ1M)	Нет	Нет	Нет	Да
	Прерывания скоростного счетчика	Да (только модули CPU CJ1M)	Нет	Нет	Нет	Да
	Внешние прерывания	Да (модули CPU CJ1 не поддерживают внешние прерывания)	Да	Нет	Нет	Нет
	От платы связи	Нет	Да	Да	Нет	Нет
	Прерывание по вкл. питания	Нет	Нет	Нет	Да	Нет
	Прерывание по выкл. питания	Да	Да	Нет	Да	Нет
	Время срабатывания прерывания	0.17 мс Встроенные входы/выходы модулей CPU CJ1M: 0.12 мс	Специальные модули ввода/вывода C200H: 1 мс Входы/выходы серии CJ: 0.1 мс	1 мс	---	Приблиз. 0.1 мс
Область настроек (параметров) ПЛК		Адреса для пользователя не предусмотрены (настройка возможна только с помощью средства программирования, включая консоль программирования)	Адреса для пользователя не предусмотрены (настройка возможна только с помощью средства программирования, включая консоль программирования)	Фиксированное размещение в области DM: DM 6600 ... DM 6655, DM 6550 ... DM 6559. Возможна настройка с консоли программирования.	Адреса для пользователя не предусмотрены (настройка возможна только с помощью средства программирования, включая консоль программирования)	Фиксированное размещение в области DM: DM 6600 ... DM 6655. Возможна настройка с консоли программирования.

Параметр		Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H	
Первоначальные настройки	Входы/ выходы	Время срабатывания входов базового модуля ввода/вывода	Задается в настройках ПЛК	Задается в настройках ПЛК	Нет	Нет	Задается в настройках ПЛК
		Начальные адреса в стойке	Задается в таблице ввода/вывода с помощью средства программирования (но порядок номеров стоек зафиксирован).	Задается в таблице ввода/вывода с помощью средства программирования (но порядок номеров стоек зафиксирован).	Нет	Задается в настройках ПЛК (можно задавать порядок номеров стоек)	Нет
		Начальный адрес оптических модулей ввода/вывода шины SYSMAC BUS, назначаемой ведущим устройством	Нет	Нет	Нет	Задается в настройках ПЛК	Нет
		Работа при ошибке проверки входов/выходов	Нет	Нет	Нет	Задается в настройках ПЛК	Нет
	Память	Защита памяти пользователя	Задается DIP-переключателем	Задается DIP-переключателем	Задается DIP-переключателем	Определяется переключателем (ключом)	Задается DIP-переключателем
		Сохраняемые области	Нет	Нет	Нет	Задается в настройках ПЛК	Нет
		Сохранение слов ввода/вывода в случае фатальных ошибок (кроме сбоя по питанию)	Нет	Нет	Нет	Задается в настройках ПЛК	Нет
		Сохранение содержимого памяти с помощью бита удержания IOM при включении питания ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК
		Сохранение памяти с помощью бита сохранения принужденного состояния при включении питания ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК
		Контроль положения DIP-переключателей	Да	Да	Да	Нет	Да
	Команды	Выбор формата (BCD или двоичный) для косвенных данных DM	Возможен непосредственный ввод	Возможен непосредственный ввод	Нет	Задается в настройках ПЛК	Нет
		Множественное использование команды JMP (0)	Множественное использование уже возможно	Множественное использование уже возможно	Нет	Задается в настройках ПЛК	Нет
		Работа в случае ошибок команд (продолжение или стоп)	Задается в настройках ПЛК	Задается в настройках ПЛК	Нет	Нет	Нет
		Выполнение в фоновом режиме	CJ1: Нет CJ1-H: Да CJ1M: Да	CS1: Нет CS1-H: Да	Нет	Нет	Нет
	Память файлов	Автоматическая загрузка при запуске	Определяется настройкой DIP- переключателя (автоматическое чтение из карты памяти)	Определяется настройкой DIP- переключателя (автоматическое чтение из карты памяти)	Определяется настройкой DIP- переключателя (автоматическое чтение из кассеты памяти)	Задается в настройках ПЛК или DIP-переключателем (автоматическое чтение из карты памяти)	Определяется настройкой DIP- переключателя (автоматическое чтение из карты памяти)
		Преобразование в файл EM	Задается в настройках ПЛК	Задается в настройках ПЛК	Нет	Нет	Нет
Прерывания	Реакция на прерывание	Нет	Нет	Задается в настройках ПЛК (C200H: скоростное срабатывание)	Нет	Нет	
	Обнаружение ошибок	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	Нет	Нет	
	Запоминание прерываний от входов/выходов во время выполнения программы обработки прерывания от входов/выходов	Нет	Нет	Нет	Задается в настройках ПЛК	Нет	
	Запрет/разрешение прерываний по выключению питания	Задается в настройках ПЛК	Задается в настройках ПЛК	Нет	Задается в настройках ПЛК	Нет	
	Настройка периода формирования запланированного прерывания	Задается в настройках ПЛК (10 мс, 1.0 мс) (также 0.1 мс только для модуля CPU CJ1M)	Задается в настройках ПЛК (10 мс, 1.0 мс)	Задается в настройках ПЛК	Задается в настройках ПЛК (10 мс, 1 мс, 0.5 мс)	Нет	

Параметр		Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H		
Первоначальные настройки (продолжение)	Напряжение питания	Удержание бита продолжения работы после перезапуска	Нет	Нет	Нет	Задается в настройках ПЛК	Нет	
		Режим запуска	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	
		Настройка условий при запуске	CJ1: Нет CJ1-H: Да CJ1M: Да	CS1: Нет CS1-H: Да	Нет	Нет	Нет	Нет
		Протоколирование запуска	Нет	Нет	Нет	Задается в настройках ПЛК	Нет	
		Обнаружение пониженного напряжения батареи	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	
		Время мгновенного пропадания питания	Нет	Нет	Нет	Задается в настройках ПЛК	Нет	
		Время задержки обнаружения отключения питания	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК (Время, в течение которого продолжается работа после обнаружения отключения питания)	Нет	Нет	
Циклы		Мгновенное прерывание питания в качестве фатальной/нефатальной ошибки	Нет	Нет	Нет	Задается в настройках ПЛК	Нет	
		Обновление входов/выходов	Нет	Нет	Задается в настройках ПЛК (только у специальных модулей ввода/вывода)	Задается в настройках ПЛК	Нет	
		Постоянная длительность цикла	Задается в настройках ПЛК (1 ... 32000 мс)	Задается в настройках ПЛК (1 ... 32000 мс)	Задается в настройках ПЛК (1 ... 9999 мс)	Задается в настройках ПЛК (1 ... 32000 мс)	Задается в настройках ПЛК (1 ... 9999 мс)	
		Контроль длительности цикла	Задается в настройках ПЛК (10 ... 40000 мс) (первоначальное значение: фикс. 1000 мс)	Задается в настройках ПЛК (10 ... 40000 мс) (первоначальное значение: фикс. 1000 мс)	Задается в настройках ПЛК (0 ... 99) Шаг: 1 с, 10 мс, 100 мс (перв. настр.: фикс. 120 мс)	Задается в настройках ПЛК (10 ... 40000 мс) (перв. настр.: фикс. 1000 мс)	Задается в настр. ПЛК (0 ... 99 мс) Шаг: 1 с, 10 мс, 100 мс (перв. настр.: фикс. 120 мс)	
		Отключение обнаружения превышения длительности цикла	Нет	Нет	Задается в настройках ПЛК	Нет	Задается в настройках ПЛК	
	Асинхронное выполнение команд и обслуживание периферии	Нет	Нет	Нет	Задается в настройках ПЛК	Нет		
Последовательные интерфейсы		Параметры порта связи RS-232C	Выбор DIP-переключателем автоматического определения или настроек ПЛК	Выбор DIP-переключателем автоматического определения или настроек ПЛК	Выбор DIP-переключателем автоматического определения или настроек ПЛК	Выбор DIP-переключателем автоматического определения или настроек ПЛК	Выбор DIP-переключателем автоматического определения или настроек ПЛК	
		Параметры периферийного порта связи	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	
		Параметры связи для платы связи	Нет	Нет	Задается в настройках ПЛК	Нет	Задается в настройках ПЛК	
Режим выполнения в CPU		Режимы параллельного выполнения	CJ1: Нет CJ1-H: Да CJ1M: Нет	CS1: Нет CS1-H: Да	Нет	Нет	Нет	
		Режим приоритетного обслуживания периферии	Да	Да	Нет	Нет	Нет	
Обслуживание прочей периферии		Время обслуживания	Задается в настройках ПЛК (фиксированное время обслуживания периферии)	Задается в настройках ПЛК (фиксированное время обслуживания периферии)	Задается в настройках ПЛК (встроенный порт RS-232C, периферийный порт, плата связи)	Нет	Задается в настройках ПЛК (встроенный порт RS-232C, периферийный порт, плата связи)	
		Измерение интервала обслуживания модуля шины CPU	Нет	Нет	Нет	Задается в настройках ПЛК	Нет	
		Прекращение циклического обновления для специального модуля ввода/вывода	Задается в настройках ПЛК	Задается в настройках ПЛК	Задается в настройках ПЛК	Нет	Нет	
		Функция CPU Bus link	Нет	Нет	Нет	Задается в настройках ПЛК	Нет	

Параметр		Серия CJ	Серия CS	C200HX/HG/HE	Серия SVM1/CV	CQM1H	
Первоначальные настройки (продолжение)	Консоль программирования	Язык консоли программирования	Задается DIP-переключателем	CS1: задается DIP-переключателем CS1-H: задается с консоли программирования	Задается DIP-переключателем	Нет	Задается DIP-переключателем
	Ошибки	Область протоколирования ошибок	Нет (Фикс.)	Нет (Фикс.)	Нет (Фикс.: DM 6001 ... 6030)	Задается в настройках ПЛК	Нет (Фикс.: DM 6569 ... 6599)
		Отмена регистрации установленных пользователем ошибок FAL в протоколе ошибок	CJ1: Нет CJ1-H: Да CJ1M: Да	CS1: Нет CS1-H: Да	Нет	Нет	Нет
	Работа	Режим ожидания CPU	Нет	Нет	Нет	Задается в настройках ПЛК	Нет
Дополнительная область	Флаги условий	Флаг ER, CY, <, >, =, всегда ВКЛ/ВЫКЛ и т.д.	Вводятся с помощью символов, например, ER	Вводятся с помощью символов, например, ER	Да	Да	Да
		Тактовые импульсы	Вводятся с помощью символов, например, 0,1 с	Вводятся с помощью символов, например, 0,1 с	Да	Да	Да
	Обслуживание	Бит отключения сервиса CPU	Нет	Нет	Нет	Да	Нет
		Коды для подсоединенных устройств	Нет	Нет	Нет	Да	Нет
		Длительности циклов обработки периферии	Нет	Нет	Нет	Да	Нет
		Интервал обслуживания модуля шины CPU	Нет	Нет	Нет	Да	Нет
		Включение/отключение периферии, подсоединенной к CPU	Нет	Нет	Нет	Да	Нет
		Бит отключения обслуживания Host Link/NT Link	Нет	Нет	Нет	Да	Нет
		Бит отключения обслуживания периферии	Нет	Нет	Нет	Да	Нет
		Бит отключения запланированного обновления	Нет	Нет	Нет	Да	Нет
		Область общего мониторинга встраиваемой платы	Нет	Да	Да	Нет	Нет
		Превышение длительности цикла	Да	Да	Да	Да	Да
	Задачи	Флаг "первая задача"	Да	Да	Нет (только флаг "первый цикл")	Нет (только флаг "первый цикл")	Нет (только флаг "первый цикл")
	Отладка	Флаг "Online-редактирование запрещено"	Да	Да	Да (AR)	Нет	Нет
		Флаг "Online-редактирование в режиме ожидания"	Да	Да	Да (AR)	Нет	Нет
		Бит "выходы ВЫКЛ"	Да	Да	Да	Да	Да
		Бит "удержание принужденного состояния"	Да	Да	Да	Да	Да
	Память файлов	Флаг "команда памяти файлов"	Да	Да	Нет	Да	Нет
		Флаг "Ошибка формата памяти файлов EM"	Да (кроме модулей CPU CJ1M)	Да	Нет	Нет	Нет
		Начальный банк для форматирования файлов EM	Да (кроме модулей CPU CJ1M)	Да	Нет	Нет	Нет
	Память	Флаги "состояние DIP-переключателя"	Да (перекл. 6)	Да (перекл. 6)	Да (AR, только перекл. 6)	Нет	Да (AR, перекл. 6)
		Бит "удержание ИОМ"	Да	Да	Да	Да	Да
	Прерывания	Макс. время выполнения подпрограммы/действия	Да	Да	Да	Нет	Нет
Флаг "Ошибка задачи обработки прерывания"		Да	Да	Да	Нет	Нет	



Параметр		Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H	
Дополнительная область, продолжение	Ошибки	Область/указатель на область хранения протокола ошибок	Да	Да	Нет	Да	Нет
		Коды ошибок	Да	Да	Да	Да	Да
	Первоначальные настройки	Инициализация настроек ПЛК	Нет	Нет	Да	Нет	Да
		Связь	Флаги рабочих уровней PLC Link	Да (бит PLC Link в дополн. области )	Да (бит PLC Link в дополн. области )	Да (AR)	Нет
	Напряжение питания	Флаг "прерывание питания"	Нет	Нет	Нет	Да	Нет
		Время прерывания питания	Нет	Нет	Нет	Да	Нет
		Время включения питания	Да	Да	Нет	Да	Нет
Время, когда произошло прерывание питания (включая отключение питания)		Да	Да	Нет	Да	Да	
Количество случаев кратковременного прерывания питания	Да (количество прерываний питания)	Да (количество прерываний питания)	Да (количество прерываний питания)	Да	Да (количество прерываний питания)		
Суммарное время включенного питания	Да	Да	Нет	Нет	Нет		
Методы резервирования	Форматирование		Резервирование в зависимости от количества слов, необходимого модулям, в соответствии с порядком их подключения.	Резервирование в зависимости от количества слов, необходимого модулям, пустые слоты пропускаются.	Фиксированное резервирование слов: каждому модулю автоматически отводится одно слово	Резервирование в зависимости от количества слов, необходимого модулям, пустые слоты пропускаются.	Резервирование в зависимости от количества слов, необходимого модулям, в соответствии с порядком их подключения.
	Резервирование для модулей с повышенной плотностью входов/ выходов группы 2		Нет	Аналогично базовому модулю ввода/вывода	Резервируется область группы 2 в области IR (положение определяется переключателем на лицевой панели)	Нет	Нет
	Способ резервирования слов		Изменение таблицы ввода/вывода с помощью CX-Programmer.	Изменение таблицы ввода/вывода с помощью CX-Programmer.	Создание таблицы ввода/вывода с пустым слотом или изменение таблицы ввода/вывода в CX-Programmer.	Фиктивный модуль ввода/вывода или изменение таблицы ввода/вывода в CX-Programmer.	Автоматическое резервирование при запуске.
	Область, резервируемая для специального модуля ввода/вывода	Область CIO	Резервируется в области для специального модуля ввода/вывода в соответствии с номером модуля, 10 слов на модуль, всего до 96 модулей.	Резервируется в области для специального модуля ввода/вывода в соответствии с номером модуля, 10 слов на модуль, всего до 96 модулей.	Резервируется в области для специального модуля ввода/вывода (в области IR) в соответствии с номером модуля, 10 слов на 1 модуль, всего до 16 модулей.	Аналогично базовым модулям ввода/вывода; 2 или 4 слова резервируются в области ввода/вывода (отдельно для каждого модуля)	Аналогично базовым модулям ввода/вывода; 1, 2 или 4 слова резервируются в области ввода/вывода (отдельно для каждого модуля)
		Область DM	Резервируется в D 20000 ... D 29599 в соответствии с номером модуля, 100 слов на модуль, всего до 96 модулей.	Резервируется в D 20000 ... D 29599 в соответствии с номером модуля, 100 слов на модуль, всего до 96 модулей.	Резервируется в DM 1000 ... DM 1999 и в DM 2000 ... DM 2599, 100 слов на модуль, всего до 16 мод.	Нет	Нет
Область, резервируемая для модуля шины CPU	Область CIO	Резервируется в области для модуля шины CPU в соответствии с номером модуля, 25 слов на 1 модуль, всего до 16 модулей.	Резервируется в области для модуля шины CPU в соответствии с номером модуля, 25 слов на 1 модуль, всего до 16 модулей.	Нет	Резервируется в области для модуля шины CPU в соответствии с номером модуля, 25 слов на 1 модуль, всего до 16 модулей.	Нет	
	Область DM	Резервируется в D 30000 ... D 31599 в соответствии с номером модуля, 100 слов на модуль, всего до 16 модулей.	Резервируется в D 30000 ... D 31599 в соответствии с номером модуля, 100 слов на модуль, всего до 16 модулей.	Нет	Резервируется в D 02000 ... D 03599 в соответствии с номером модуля, 100 слов на модуль, всего до 16 модулей.	Нет	

Параметр		Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H	
Память ввода/вывода	Область CIO	Да	Да	Да	Да	Да	
	Область WR	Да	Да	Нет	Нет	Нет	
	Временная область для реле	Да	Да	Да	Да	Да	
	Дополнительная область	Да	Да	Да	Да	Да	
	Область SR	Нет	Нет	Да	Нет	Да	
	Область связей	Да (область логических связей)	Да (область логических связей)	Да (область логических связей)	Нет	Да	
	Область для специального модуля ввода/вывода C200H	Да	Да	Да (область CIO)	Нет	Нет	
	Область для встроенных входов/выходов	Да (только для модуля CPU CJ1M со встроен. входами/выходами)	Нет	Нет	Нет	Нет	
	Область для Serial PLC Link (последовательн. соед. с ПЛК)	Да (только для модуля CPU CJ1M со встроен. входами/выходами)	Нет	Нет	Нет	Нет	
	Область DM	Да	Да	Да	Да	Да	
	Область расширенной памяти данных (EM)	Да (могут назначаться адреса, включая номер банка) (не поддерживается модулем CPU CJ1M )	Да (могут назначаться адреса, включая номер банка)	Да (для - Z могут назначаться адреса, а банки не могут)	Да (адрес, включая банк, указываться не может; банк должен изменяться. Требуется модуль EM).	Да (без банков)	
	Область таймеров/счетчиков	Да	Да	Да	Да	Да	
	Регистры индексов	Да	Да	Нет	Да	Нет	
	Регистры данных	Да	Да	Нет	Да	Нет	
	Области для принудительной установки/ сброса	Область CIO	Да	Да	Да	Да	Нет
		Область WR	Да	Да	Нет	Нет	Да
Область удержания		Да	Да	Да	Нет	Нет	
Дополнительная область		Нет	Нет	Да	Нет	Да	
Область SR		Нет	Нет	Нет	Нет	Нет	
Область связей		Нет	Нет	Да	Нет	Нет	
Таймеры/счетчики		Да (Флаг)	Да (Флаг)	Да (Флаг)	Да (Флаг)	Да (Флаг)	
Область DM		Нет	Нет	Нет	Нет	Нет	
Варианты выполнения команд/косвенная адресация	Выделение ВЫКЛ → ВКЛ (однократное)	Да (указывается символом @)	Да (указывается символом @)	Да (указывается символом @)	Да (указывается символом ↑)	Да (указывается символом @)	
	Выделение ВКЛ → ВЫКЛ (однократное)	Да (указывается символом %)	Да (указывается символом %)	Нет (вместо этого используется команда DIFD)	Да (указывается символом ↓)	Нет (применяется DIFD)	
	Мгновенное обновление	Да (указывается символом !)	Да (указывается символом !)	Нет (вместо этого используется команда IORF)	Да (указывается символом !)	Нет (используется IORF)	
	Косвенная адресация к DM/EM	Режим BCD	Да (0000 ... 9999) используется символ ""	Да (0000 ... 9999) используется символ ""	Да (0000 ... 9999)	Да (0000 ... 9999)	Да (0000 ... 9999) используется символ ""
		Двоичный режим	Да (00000 ... 32767) используется символ @. 0000 ... 7FFF Hex: 0000 ... 31767 8000 ... FFFF Hex: 00000 ... 32767 в следующем банке	Да (00000 ... 32767) используется символ @. 0000 ... 7FFF Hex: 0000 ... 31767 8000 ... FFFF Hex: 00000 ... 32767 в следующем банке	Нет	Да, но только для косвенной адресации с помощью адресов памяти ПЛК.	Нет
Способ резервирования	Выбор первого слова для стойки	Да (для всех модулей CPU)	Да (для всех модулей CPU)	Нет	Нет	Нет	
	Выбор первого слова для слота	Модули CPU до версии 2.0: см. примечание. Модули CPU версии 2.0 или новее: да	Модули CPU до версии 2.0: см. примечание. Модули CPU версии 2.0 или новее: да	Нет	Нет	Нет	

Параметр		Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H
Online-соединения через сети без создания таблиц ввода/вывода		С автоматическим назначением входов/выходов при включении питания: да (для всех модулей CPU) В ручном режиме Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Да, но только для Controller Link
Связь через несколько сетей		Модули CPU до версии 2.0: 3 уровня Модули CPU версии 2.0 или выше: 8 уровней	Модули CPU до версии 2.0: 3 уровня Модули CPU версии 2.0 или выше: 8 уровней	Нет	Да, до 3-х уровней	Нет
Online-соединения с ПЛК через программируемые терминалы серии NS		Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет
Защита модулей CPU от записи с помощью команд FINS, передаваемых по сети		Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет
Загрузка отдельных задач		Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет
Защита от чтения с помощью паролей	Полностью вся программа пользователя	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет
	Указанные задачи	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет
	Разрешение/запрет создания файлов программ в памяти файлов	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет
	Защита программы от записи	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет
Автоматическая загрузка при включении питания без файла параметров (.STD)		Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет

**Примечание** Для модулей CPU, произведенных 01.06.2002 или позже (номера партий 020601 □□□□ или выше) может быть задано до 8 адресов слотов.

## Сравнение команд (инструкций)

Параметр		Мнемонический код	Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H
Команды последовательного ввода	LOAD/AND/OR	LD/AND/OR	Да	Да	Да	Да	Да
	AND LOAD/OR LOAD	AND LD/OR LD	Да	Да	Да	Да	Да
	NOT	NOT	Да	Да	Да	Да	Нет
	CONDITION ON	UP	Да	Да	Нет	Да (*1)	Нет
	CONDITION OFF	DOWN	Да	Да	Нет	Да (*1)	Нет
	BIT TEST	TSTN	Да (положение бита указывается двоичным числом: 0000 ... 000F Hex)	Да (положение бита указывается двоичным числом: 0000 ... 000F Hex)	Да (положение бита указывается в формате BCD) (*2)	Да (положение бита указывается в формате BCD) (*1)	Нет
Команды последовательного вывода	OUTPUT	OUT	Да	Да	Да	Да	Да
	TR	TR	Да	Да	Да	Да	Да
	KEEP	KEEP	Да	Да	Да	Да	Да
	DIFFERENTIATE UP/DOWN	DIFU/DIFD	Да (LD↑, AND↑, OR↑) (LD↓, AND↓, OR↓)	Да (LD↑, AND↑, OR↑) (LD↓, AND↓, OR↓)	Да (DIFU/DIFD)	Да (LD↑, AND↑, OR↑) (LD↓, AND↓, OR↓)	Да (DIFU/DIFD)
	SET и RESET	SET/RSET	Да	Да	Да	Да	Да
	MULTIPLE BIT SET/RESET	SETA/RSTA	Да (начальный бит и количество битов указывается двоичным числом)	Да (начальный бит и количество битов указывается двоичным числом)	Нет	(*1) (Начальный бит и количество битов указывается в формате BCD)	Нет
	SINGLE BIT SET/RESET	SET/RSTB	CJ1: Нет CJ1-H: Да CJ1M: Да	CS1: Нет CS1-H: Да	Нет	Нет	Нет
	SINGLE BIT OUTPUT	OUTB	CJ1: Нет CJ1-H: Да CJ1M: Да	CS1: Нет CS1-H: Да	Нет	Нет	Нет
Команды последовательного управления	END/NO OPERATION	END/NOP	Да	Да	Да	Да	Да
	INTERLOCK/INTERLOCK CLEAR	IL/ILC	Да	Да	Да	Да	Да
	Команды для многократной блокировки (?)	MILH/MILR/MILC	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет
	JUMP/JUMP END	JMP/JME	Да (номер для перехода указывается в BCD: 0 ... 1023)	Да (номер для перехода указывается в BCD: 0 ... 1023)	Да (номер для перехода указывается в BCD: 0 ... 99)	Да (номер для перехода указывается в BCD: 0 ... 999)	Да (номер для перехода указывается в BCD: 0 ... 99)
	CONDITIONAL JUMP	CJP/CJPN	Да (номер для перехода указывается в BCD: 0 ... 1023)	Да (номер для перехода указывается в BCD: 0 ... 1023)	Нет	Да (номер для перехода указывается в BCD: 0 ... 999) (*1)	Нет
	MULTIPLE JUMP/JUMP END	JMP0/JME0	Да	Да	Нет	Нет (но в настройках ПЛК можно разрешить многократные переходы с помощью номера для перехода 0)	Нет
	FOR/NEXT LOOPS	FOR/NEXT	Да	Да	Нет	Нет	Нет
	BREAK LOOP	BREAK	Да	Да	Нет	Нет	Нет

Параметр	Мнемонический код	Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H	
Команды для таймеров и счетчиков	TIMER	TIM (BCD)	Да	Да	Да	Да	Да
		TIMX (двоичный).	Да (*4)	Да (*4)	Нет	Нет	Нет
	HIGH-SPEED TIMER	TIMH (BCD)	Да	Да	Да	Да	Да
		TIMHX (двоичный).	Да (*4)	Да (*4)	Нет	Нет	Нет
	ONE-MS TIMER	TMHN (BCD)	Да	Да	Нет	Нет	Нет
		TMHNX (двоичный).	Да (*4)	Да (*4)	Нет	Нет	Нет
	ACCUMULATIVE TIMER	TTIM (BCD)	Да	Да	Да	Да	Да
		TTIMX (двоичный).	Да (*4)	Да (*4)	Нет	Нет	Нет
	LONG TIMER	TIML (BCD)	Да	Да	Нет	Да	Нет
		TIMLX (двоичный).	Да (*4)	Да (*4)	Нет	Нет	Нет
	MULTI-OUTPUT TIMER	MTIM (BCD)	Да	Да	Нет	Да	Нет
		MTIMX (двоичный).	Да (*4)	Да (*4)	Нет	Нет	Нет
	COUNTER	CNT (BCD)	Да	Да	Да	Да	Да
		CNTX (двоичный).	Да (*4)	Да (*4)	Нет	Нет	Нет
	REVERSIBLE COUNTER	CNTR (BCD)	Да	Да	Да	Да	Да
		CNTRX (двоичный).	Да (*4)	Да (*4)	Нет	Нет	Нет
	RESET TIMER/COUNTER	CNR (BCD)	Да (сбрасывает только таймер или счетчик)	Да (сбрасывает только таймер или счетчик)	Нет	Да (также обнуляет указанный диапазон в области CIO)	Нет
		CNRX (двоичный).	Да (*4)	Да (*4)	Нет	Нет	Нет

Параметр		Мнемонический код	Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H
Команды сравнения	Символы сравнения	=, <, и т. п.	Да (поддерживаются все символы для LD, OR и AND)	Да (поддерживаются все символы для LD, OR и AND)	Да (*2) (поддерживаются только для AND)	Да (*1) (поддерживаются только для AND)	Нет
	Команды сравнения данных	=DT, <DT, и т. п.	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет
	COMPARE/ DOUBLE COMPARE	CMPL/ CMPL	Да	Да	Да	Да (*3)	Да
	SIGNED BINARY COMPARE/ DOUBLE SIGNED BINARY COM- PARE	CPS/ CPSL	Да	Да	Да	Да (*1)	Да
	BLOCK COM- PARE	BCMP	Да	Да	Да	Да	Да
	EXTENDED BLOCK COMPARE	BCMP2	Да (только модули CPU CJ1M и модули CPU CJ1-H версии 2.0)	Да (только модули CPU CJ1-H/CS1D версии 2.0)	Нет	Нет	Нет
	TABLE COMPARE	TCMP	Да	Да	Да	Да	Да
	MULTIPLE COMPARE	MCMP	Да	Да	Да	Да	Да
	EQUALS	EQU	Нет	Нет	Нет	Да	Нет
AREA RANGE COMPARE	ZCP/ ZCPL	CJ1: нет (применяются команды сравнения) CJ1-H: да CJ1M: да	CS1: нет (применяются команды сравнения) CS1-H: да	Да	Нет	Нет (применяются команды сравнения)	

Параметр	Мнемонический код	Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H	
Команды для перемещения данных	MOVE	MOV	Да	Да	Да	Да	Да
	DOUBLE MOVE	MOVL	Да	Да	Нет	Да	Нет
	MOVE NOT	MVN	Да	Да	Да	Да	Да
	DOUBLE MOVE	MVNL	Да	Да	Нет	Да	Нет
	DATA EXCHANGE	XCHG	Да	Да	Да	Да	Да
	DOUBLE DATA EXCHANGE	XCGL	Да	Да	Нет	Да	Нет
	MOVE QUICK	MOVQ	Нет	Нет	Нет	Да	Нет
	BLOCK TRANSFER	XFER	Да (передаваемое кол-во слов указывается двоичным числом: 0 ... 65535)	Да (передаваемое кол-во слов указывается двоичным числом: 0 ... 65535)	Да (передаваемое количество слов указывается в BCD: 0 ... 6144)	Да (передаваемое количество слов указывается в BCD: 0 ... 9999)	Да (передаваемое количество слов указывается в BCD: 0 ... 9999)
	BLOCK SET	BSET	Да	Да	Да	Да	Да
	MOVE BIT	MOVB	Да (положение исходного бита и адресуемого бита указывается двоичным числом)	Да (положение исходного бита и адресуемого бита указывается двоичным числом)	Да (положение исходного бита и адресуемого бита указывается в BCD)	Да (положение исходного бита и адресуемого бита указывается в BCD)	Да (положение исходного бита и адресуемого бита указывается в BCD)
	MULTIPLE BIT TRANSFER	XFRB	Да	Да	Да	Да (*1)	Да
	MOVE DIGIT	MOVD	Да	Да	Да	Да	Да
	SINGLE WORD DISTRIBUTE	DIST	Да (для другой команды поддерживается функция стека. Величина смещения указывается двоичным числом: 0 ... 65535)	Да (для другой команды поддерживается функция стека. Величина смещения указывается двоичным числом: 0 ... 65535)	Да (поддерживается функция стека. Величина смещения указывается в BCD: 0 ... 8999)	Да (для другой команды поддерживается функция стека. Величина смещения указывается в BCD: 0 ... 9999)	Да (поддерживается функция стека. Величина смещения указывается в BCD: 0 ... 8999)
	DATA COLLECT	COLL	Да (для другой команды поддерживается функция стека. Величина смещения указывается двоичным числом: 0 ... 65535)	Да (для другой команды поддерживается функция стека. Величина смещения указывается двоичным числом: 0 ... 65535)	Да (поддерживается функция стека. Величина смещения указывается в BCD: 0 ... 7999)	Да (для другой команды поддерживается функция стека. Величина смещения указывается в BCD: 0 ... 9999)	Да (поддерживается функция стека. Величина смещения указывается в BCD: 0 ... 7999)
	EM BLOCK TRANSFER BETWEEN BANKS	BXFR	Нет (функция доступна для макс. 65535 слов путем прямого обращения к области EM с помощью XERR)	Нет (функция доступна для макс. 65535 слов путем прямого обращения к области EM с помощью XERR)	Нет	Да (*1)	Нет
EM BLOCK TRANSFER	XFR2	Нет	Нет	Да	Нет	Нет	
EM BANK TRANSFER	BXF2	Нет	Нет	Да	Нет	Нет	
MOVE TO REGISTER	MOVR	Да (при косвенном обращении к DM/EM адрес не указывается)	Да (при косвенном обращении к DM/EM адрес не указывается)	Нет	Да (при косвенном обращении к DM/EM адрес не указывается)	Нет	
MOVE TIMER/COUNTER PV TO REGISTER	MOVW	Да	Да	Нет	Нет (возможно только для флагов завершения с помощью MOVW)	Нет	

Параметр	Мнемонический код	Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H	
Команды для сдвига данных	SHIFT REGISTER	SFT	Да	Да	Да	Да	Да
	REVERSIBLE SHIFT REGISTER	SFTR	Да	Да	Да	Да	Да
	ASYNCHRONOUS SHIFT REGISTER	ASFT	Да	Да	Да	Да	Да
	WORD SHIFT	WSFT	Да (как и у CV: 3 операнда)	Да (как и у CV: 3 операнда)	Да	Да	Да
	ARITHMETIC SHIFT LEFT/ARITHMETIC SHIFT RIGHT	ASL/ASR	Да	Да	Да	Да	Да
	ROTATE LEFT/ROTATE RIGHT	ROL/ROR	Да	Да	Да	Да	Да
	ONE DIGIT SHIFT LEFT/ONE DIGIT SHIFT RIGHT	SLD/SRD	Да	Да	Да	Да	Да
	SHIFT N-BIT DATA LEFT/SHIFT N-BIT DATA RIGHT	NSFR/NSFL	Да (смещение данных и начальный бит указываются в двоичном коде)	Да (смещение данных и начальный бит указываются в двоичном коде)	Нет	Да (смещение данных и начальный бит указываются в BCD) (*1)	Нет
	SHIFT N-BITS LEFT/SHIFT N-BITS RIGHT/DOUBLE SHIFT N-BITS LEFT/DOUBLE SHIFT NITS RIGHT	NASL/NASR, NSLL/NSRL	Да (количество смещаемых битов указывается в двоичном коде)	Да (количество смещаемых битов указывается в двоичном коде)	Нет	Да (количество смещаемых битов указывается в BCD) (*1)	Нет
	DOUBLE SHIFT LEFT/DOUBLE SHIFT RIGHT	ASLL/ASRL	Да	Да	Нет	Да	Нет
	DOUBLE ROTATE LEFT/DOUBLE ROTATE RIGHT	ROLL/RORL	Да	Да	Нет	Да	Нет
	ROTATE LEFT WITHOUT CARRY/ROTATE RIGHT WITHOUT CARRY/DOUBLE ROTATE LEFT WITHOUT CARRY/DOUBLE ROTATE RIGHT WITHOUT CARRY	RLNC/RRNC, RLNL/RRNL	Да	Да	Нет	Да (*1)	Нет
	Команды для пошагового увеличения и уменьшения	INCREMENT BCD/DECREMENT BCD	++B/--B (INC/DEC)	Да (++B/--B)	Да (++B/--B)	Да (INC/DEC)	Да (INC/DEC)
DOUBLE INCREMENT BCD/DOUBLE DECREMENT BCD		++BL/--BL (INCL/DECL)	Да (++BL/--BL)	Да (++BL/--BL)	Нет	Да (INCL/DECL)	Нет
INCREMENT BINARY/DECREMENT BINARY		++/-- (INCB/DECB)	Да (CY устанавливается (ВКЛ) в случае переноса или заема) (++L/--)	Да (CY устанавливается (ВКЛ) в случае переноса или заема) (++L/--)	Нет	Да	Нет
DOUBLE INCREMENT BINARY/DOUBLE DECREMENT BINARY		++L/--L (INBL/DCBL)	Да (CY устанавливается (ВКЛ) в случае переноса или заема) (++L/--)	Да (CY устанавливается (ВКЛ) в случае переноса или заема) (++L/--)	Нет	Да	Нет
Математические команды		Да	Да	Да	Да	Да	



Параметр	Мнемонический код	Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H	
Команды для преобразования данных	BCD-TO-BINARY/ DOUBLE BCD-TO- DOUBLE BINARY	BIN/ BINL	Да	Да	Да	Да	Да
	BINARY-TO-BCD/ DOUBLE BINARY- TO-DOUBLE BCD	BCD/ BCDL	Да	Да	Да	Да	Да
	2'S COMPLE- MENT/ DOUBLE 2'S COMPLE- MENT	NEG/ NEGL	Да (как и у CV, но UP не вклю- чается, когда в источнике лежит 8000 Hex)	Да (как и у CV, но UP не вклю- чается, когда в источнике лежит 8000 Hex)	Да	Да	Да
	16-BIT TO 32-BIT SIGNED BINARY	SIGN	Да	Да	Нет	Да	Нет
	DATA DECODER	MLPX	Да	Да	Да	Да	Да
	DATA ENCODER	DMPX	Да (как и у CVM1-V2: можно выбрать включе- ние для младшего бита)	Да (как и у CVM1-V2: можно выбрать включе- ние для младшего бита)	Да (для включения только старший бит)	Да (CVM1-V2: можно выбрать включение для младшего бита)	Да (для включения только старший бит)
	ASCII CONVERT	ASC	Да	Да	Да	Да	Да
	ASCII TO HEX	HEX	Да	Да	Да	Да (*1)	Да
	COLUMN TO LINE/LINE TO COLUMN	LINE/ COLM	Да (положение бита указывает- ся двоичным кодом)	Да (положение бита указывает- ся двоичным кодом)	Да (положение бита указывает- ся в формате BCD)	Да (положение бита указывает- ся в формате BCD)	Да (положение бита указывает- ся в формате BCD)
	SIGNED BCD-TO- BINARY/DOUBLE SIGNED BCD-TO- BINARY	BINS/ BISL	Да	Да	Нет	Да (*1)	Нет
	SIGNED BINARY- TO-BCD/DOUBLE SIGNED BINARY- TO-BCD	BCDS/ BDSL	Да	Да	Нет	Да (*1)	Нет
	GRAY CODE CONVERSION	GRY	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да (также поддерживается, начиная с номе- ра партии 030201 или выше)	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да (также поддерживается, начиная с номе- ра партии 030201 или выше)	Нет	Нет	Нет
Команды логических операций	LOGICAL AND/ LOGICAL OR/ EXCLUSIVE OR/ EXCLUSIVE NOR	ANDW, ORW, XORW, XNRW	Да	Да	Да	Да	Да
	DOUBLE LOGI- CAL AND/DOU- BLE LOGICAL OR/ DOUBLE EXCLU- SIVE OR/DOU- BLE EXCLUSIVE NOR	ANDL, ORWL, XORL, XNRL	Да	Да	Нет	Да	Нет
	COMPLEMENT/ DOUBLE COM- PLEMENT	COM/ COML	Да	Да	Да (только COM)	Да	Да (только COM)
Специальные математические команды	BCD SQUARE ROOT	ROOT	Да	Да	Да	Да	Да
	BINARY ROOT	ROTB	Да	Да	Нет	Да (*1)	Нет
	ARITHMETIC PROCESS	APR	Да	Да	Да	Да	Да
	FLOATING POINT DIVIDE	FDIV	Да	Да	Да	Да	Нет
	BIT COUNTER	BCNT	Да (количество слов для счета и результаты счета в двоичном коде: 0 ... FFFF Hex)	Да (количество слов для счета и результаты счета в двоич- ном коде: 0 ... FFFF Hex)	Да (количество слов для счета и результаты счета в BCD: 1 ... 6656)	Да (количество слов для счета и результаты счета в BCD: 0 ... 9999, но для 0 - ошибка)	Да (количество слов для счета и результаты счета в BCD: 1 ... 6656)

Параметр		Мнемонический код	Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H
Команды для математики с плавающей запятой	FLOATING TO 16-BIT/32-BIT BIN, 16-BIT/32-BIT BIN TO FLOATING	FIX/FIXL, FLT/FTL	Да	Да	Нет	Да (*1)	Да
	FLOATING-POINT ADD/FLOATING-POINT SUBTRACT/FLOATING-POINT MULTIPLY/FLOATING-POINT DIVIDE	+F, -F, *F, /F	Да	Да	Нет	Да (*1)	Да
	DEGREES TO RADIANS/RADIANS TO DEGREES	RAD, DEG	Да	Да	Нет	Да (*1)	Да
	SINE/COSINE/TANGENT/ARC SINE/ARC TANGENT	SIN, COS, TAN, ASIN, ACOS, ATAN	Да	Да	Нет	Да (*1)	Да
	SQUARE ROOT	SQRT	Да	Да	Нет	Да (*1)	Да
	EXPONENT	EXP	Да	Да	Нет	Да (*1)	Да
	LOGARITHM	LOG	Да	Да	Нет	Да (*1)	Да
	EXPONENTIAL POWER	PWR	Да	Да	Нет	Нет	Нет
	Сравнение десятичных чисел с плавающей запятой	Примеры: =F, < > F	CJ1: Нет CJ1-H: Да CJ1M: Да	CS1: Нет CS1-H: Да	Нет	Нет	Нет
	Преобразование десятичного числа с плавающей запятой в текстовую строку	FSTR, FVAL	CJ1: Нет CJ1-H: Да CJ1M: Да	CS1: Нет CS1-H: Да	Нет	Нет	Нет
Команды для преобразования и вычислений с плавающей запятой, с одинарной точностью (см. выше)	Пример: FIXD	CJ1: Нет CJ1-H: Да CJ1M: Да	CS1: Нет CS1-H: Да	Нет	Нет	Нет	

Параметр	Мнемонический код	Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H	
Команды обработки таблиц данных	SET STACK	SSET	Да (четыре слова данных для управления стеком. Количество слов указывается в двоичном коде: 5 ... 65535)	Да (четыре слова данных для управления стеком). Количество слов указывается в двоичном коде: 5 ... 65535)	Нет	Да (четыре слова данных для управления стеком). Количество слов указывается в BCD: 3 ... 9999)	Нет
	PUSH ONTO STACK:	PUSH	Да	Да	Нет	Да	Нет
	FIRST IN FIRST OUT	FIFO	Да	Да	Нет	Да	Нет
	LAST IN FIRST OUT	LIFO	Да	Да	Нет	Да	Нет
	FIND MAXIMUM/ FIND MINIMUM	MAX, MIN	Да (два слова в поле данных управления. Длина таблицы указывается двоичным кодом: 1 ... FFFF)	Да (два слова в поле данных управления. Длина таблицы указывается двоичным кодом: 1 ... FFFF)	Да (два слова в поле данных управления. Длина таблицы указывается в BCD: 1 ... 999)	Да (два слова в поле данных управления. Длина таблицы указывается в BCD: 1 ... 999)	Да (два слова в поле данных управления. Длина таблицы указывается в BCD: 1 ... 999)
	DATA SEARCH	SRCH	Да (длина таблицы указывается в двоичном коде: 1 ... FFFF. Адрес памяти ПЛК размещается в IR0. Количество совпадений может быть выведено в DR0)	Да (длина таблицы указывается в двоичном коде: 1 ... FFFF. Адрес памяти ПЛК размещается в IR0. Количество совпадений может быть выведено в DR0)	Да (длина таблицы указывается в BCD: 1 ... 6556. Адрес памяти ПЛК размещается в S+1. Количество совпадений не может быть выведено в DR0)	Да (длина таблицы указывается в BCD: 1 ... 9999. Адрес памяти ПЛК размещается в IR0. Количество совпадений не может быть выведено в DR0)	Да (длина таблицы указывается в BCD: 1 ... 6556. Адрес памяти ПЛК размещается в S+1. Количество совпадений не может быть выведено в DR0)
	FRAME CHECK-SUM	FCS	Да	Да	Да	Нет	Да
	SUM	SUM	Да (как и у C200HX/HG/HE: возможно суммирование байтов, а также слов)	Да (как и у C200HX/HG/HE: возможно суммирование байтов, а также слов)	Да (возможно суммирование байтов, а также слов)	Да (возможно суммирование байтов, а также слов)	Да (возможно суммирование байтов, а также слов)
	SWAP BYTES	SWAP	Да (можно использовать для обмена данными и других задач)	Да (можно использовать для обмена данными и других задач)	Нет	Нет	Нет
	DIMENSION RECORD TABLE:	DIM	Да	Да	Нет	Нет	Нет
SET RECORD LOCATION	SETR	Да	Да	Нет	Нет	Нет	
GET RECORD LOCATION	GETR	Да	Да	Нет	Нет	Нет	

Параметр	Мнемонический код	Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H	
Команды для регулирования (управления)	SCALING	SCL	Да	Да	Да	Нет	Да
	SCALING 2	SCL2	Да	Да	Нет	Нет	Да
	SCALING 3	SCL3	Да	Да	Нет	Нет	Да
	PID CONTROL	PID	Да (когда PV = SV, для выхода можно выбрать 0% или 50%. PID и период дискретизации указываются в двоичном коде)	Да (когда PV = SV, для выхода можно выбрать 0% или 50%. PID и период дискретизации указываются в двоичном коде)	Да (PID и период дискретизации указываются в BCD)	Да (PID и период дискретизации указываются в BCD) (*1)	Да (PID и период дискретизации указываются в BCD)
	PID CONTROL WITH AUTO-TUNING	PIDAT	CJ1: Нет CJ1-H: Да CJ1M: Да	CS1: Нет CS1-H: Да	Нет	Нет	Нет
	LIMIT CONTROL	LMT	Да	Да	Нет	Да (*1)	Нет
	DEAD BAND CONTROL	BAND	Да	Да	Нет	Да (*1)	Нет
	DEAD ZONE CONTROL	ZONE	Да	Да	Нет	Да (*1)	Нет
	TIME-PROPORTIONAL OUTPUT	TPO	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет
AVERAGE	AVG	Да (количество циклов указывается двоичным кодом)	Да (количество циклов указывается двоичным кодом)	Да (количество циклов указывается в BCD)	Нет	Да (количество циклов указывается в BCD)	
Команды для подпрограмм	SUBROUTINE CALL/SUBROUTINE ENTRY/SUBROUTINE RETURN	SBS, SBN, RET	Да (номер подпрограммы указывается в BCD: 0 ... 1023)	Да (номер подпрограммы указывается в BCD: 0 ... 1023)	Да (номер подпрограммы указывается в BCD: 0 ... 255)	Да (номер подпрограммы указывается в BCD: 0 ... 999)	Да (номер подпрограммы указывается в BCD: 0 ... 255)
	MACRO	MCRO	Да (номер подпрограммы указывается в BCD: 0 ... 1023)	Да (номер подпрограммы указывается в BCD: 0 ... 1023)	Да (номер подпрограммы указывается в BCD: 0 ... 255)	Да (номер подпрограммы указывается в BCD: 0 ... 999) (*1)	Да (номер подпрограммы указывается в BCD: 0 ... 255)
	Команды для глобальных подпрограмм	GSBS, GSBN, RET	CJ1: Нет CJ1-H: Да CJ1M: Да	CS1: Нет CS1-H: Да	Нет	Нет	Нет
Команды для управления прерываниями	SET INTERRUPT MASK	MSKS	Да	Да	Нет (вся обработка прерываний выполняется с помощью INT)	Да	Нет (вся обработка прерываний выполняется с помощью INT)
	CLEAR INTERRUPT	CLI	Да	Да	Нет	Да	Нет
	READ INTERRUPT MASK:	MSKR	Да	Да	Нет	Да	Нет
	DISABLE INTERRUPTS	DI	Да	Да	Нет	Нет	Нет
	ENABLE INTERRUPTS	EI	Да	Да	Нет	Нет	Нет
	ENABLE TIMER	STIM	Нет	Нет	Нет	Нет	Да

Параметр	Мнемонический код	Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H	
Команды для скоростного счетчика/импульсного выхода	MODE CONTROL	INI	Да (*5)	Нет	Нет	Нет	Да
	PRESENT VALUE READ	PRV	Да (*5)	Нет	Нет	Нет	Да
	COUNTER FREQUENCY CONVERT	PRV2	Модули CPU CJ1M версии 2.0 или выше: да (*5) CJ1-H (все модули CPU): нет	Нет	Нет	Нет	Нет
	SET COMPARISON TABLE	CTBL	Да (*5)	Нет	Нет	Нет	Да
	SET PULSES	PULS	Да (*5)	Нет	Нет	Нет	Да
	SET FREQUENCY	SPED	Да (*5)	Нет	Нет	Нет	Да
	ACCELERATION CONTROL	ACC	Да (*5)	Нет	Нет	Нет	Да
	POSITION CONTROL	PLS2	Да (*5)	Нет	Нет	Нет	Да
	ORIGIN SEARCH	ORG	Да (*5)	Нет	Нет	Нет	Нет
PWM OUTPUT	PWM	Да (*5)	Нет	Нет	Нет	Да	
Команды для шагового режима	STEP DEFINE и STEP START	STEP/SNXT	Да	Да	Да	Да	Да
Команды базового модуля ввода/вывода	I/O REFRESH	IORF	Да	Да (используется для модулей C200H группы 2 с высокой плотностью входов/выходов и специальных модулей ввода/вывода. Включает функцию GROUP-2 HIGH-DENSITY I/O REFRESH (MPRF))	Да (используется для модулей C200H группы 2 с высокой плотностью входов/выходов и специальных модулей ввода/вывода)	Да	Да
	7-SEGMENT DECODER	SDEC	Да	Да	Да	Да	Да
	DIGITAL SWITCH INPUT	DSW	Модули CPU до версии 2.0 : нет Модули CPU версии 2.0 или выше : да	Модули CPU до версии 2.0 : нет Модули CPU версии 2.0 или выше : да	Да	Нет	Да
	TEN KEY INPUT	TKY	Модули CPU до версии 2.0 : нет Модули CPU версии 2.0 или выше : да	Модули CPU до версии 2.0 : нет Модули CPU версии 2.0 или выше : да	Да	Нет	Да
	HEXADECIMAL KEY INPUT	HKY	Модули CPU до версии 2.0 : нет Модули CPU версии 2.0 или выше : да	Модули CPU до версии 2.0 : нет Модули CPU версии 2.0 или выше : да	Да	Нет	Да
	MATRIX INPUT	MTR	Модули CPU до версии 2.0 : нет Модули CPU версии 2.0 или выше : да	Модули CPU до версии 2.0 : нет Модули CPU версии 2.0 или выше : да	Да	Нет	Нет
	7-SEGMENT DISPLAY OUTPUT	7SEG	Модули CPU до версии 2.0 : нет Модули CPU версии 2.0 или выше : да	Модули CPU до версии 2.0 : нет Модули CPU версии 2.0 или выше : да	Да	Нет	Да
	GROUP-2 HIGH-DENSITY I/O REFRESH	MPRF	Нет	Нет	Да	Нет	Нет
	TEN KEY INPUT	TKY	Нет	Нет	Да	Нет	Да
	HEXADECIMAL KEY INPUT	HKY	Нет	Нет	Да	Нет	Да
	DIGITAL SWITCH INPUT	DSW	Нет	Нет	Да	Нет	Да
	MATRIX INPUT	MTR	Нет	Нет	Да	Нет	Нет
	7-SEGMENT DISPLAY OUTPUT	7SEG	Нет	Нет	Да	Нет	Да

Параметр		Мнемонический код	Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H
Команды для специального модуля ввода/вывода	SPECIAL I/O UNIT READ и SPECIAL I/O UNIT WRITE (I/O READ и I/O WRITE)	IORD/IOWR  (READ/WRIT)	IORD/IOWR (до 96 модулей. Больше не будет использоваться для передачи команд FINS)	IORD/IOWR (до 96 модулей. Больше не будет использоваться для передачи команд FINS)	IORD/IOWR	READ/WRIT	Нет
	I/O READ 2 и I/O WRITE 2	RD2/WR2	Нет	Нет	Нет	Да (*1)	Нет
Команды обработки текстовых строк	MOV STRING	MOV\$	Да	Да	Нет	Нет	Нет
	CONCATENATE STRING	+\$	Да	Да	Нет	Нет	Нет
	GET STRING LEFT	LEFT\$	Да	Да	Нет	Нет	Нет
	GET STRING RIGHT	RGHT\$	Да	Да	Нет	Нет	Нет
	GET STRING MIDDLE	MID\$	Да	Да	Нет	Нет	Нет
	FIND IN STRING	FIND\$	Да	Да	Нет	Нет	Нет
	STRING LENGTH	LEN\$	Да	Да	Нет	Нет	Нет
	REPLACE IN STRING	RPLC\$	Да	Да	Нет	Нет	Нет
	DELETE STRING	DEL\$	Да	Да	Нет	Нет	Нет
	EXCHANGE STRING	XCHG\$	Да	Да	Нет	Нет	Нет
	CLEAR STRING:	CLR\$	Да	Да	Нет	Нет	Нет
INSERT INTO STRING	INS\$	Да	Да	Нет	Нет	Нет	

Параметр	Мнемонический код	Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H	
Команды для связи по последовательному интерфейсу	RECEIVE	RXD	Да (количество записываемых байтов указывается двоичным кодом) (служит только для порта RS-232C модуля CPU. Не используется для модуля последовательной связи или периферийного порта модуля CPU)	Да (количество записываемых байтов указывается двоичным кодом) (служит только для порта RS-232C модуля CPU. Не используется для встраиваемой платы, модуля последовательной связи или периферийного порта модуля CPU)	Да (количество записываемых байтов указывается в BCD) (служит для периферийного порта, порта RS-232C или платы связи в модуле CPU)	Нет	Да (количество записываемых байтов указывается в BCD) (служит для периферийного порта, порта RS-232C или платы связи в модуле CPU)
	TRANSMIT	TXD	Да (количество записываемых байтов указывается в двоичном коде) (служит только для порта RS-232C модуля CPU. Нельзя использовать для модуля последовательного интерфейса или периферийного порта модуля CPU) (незапрашиваемый обмен данными с помощью команды Host Link EX не возможен)	Да (количество записываемых байтов указывается в двоичном коде) (служит только для порта RS-232C модуля CPU. Нельзя использовать для встраиваемой платы, модуля последовательного интерфейса или периферийного порта модуля CPU) (незапрашиваемый обмен данными с помощью команды Host Link EX не возможен)	Да (количество записываемых байтов указывается в BCD) (служит для периферийного порта, порта RS-232C или платы связи в модуле CPU) (незапрашиваемый обмен данными с помощью команды Host Link EX не возможен)	Нет	Да (количество записываемых байтов указывается в BCD) (служит для периферийного порта, порта RS-232C или платы связи в модуле CPU) (незапрашиваемый обмен данными с помощью команды Host Link EX не возможен)
	CHANGE SERIAL PORT SETUP	STUP	Да (устанавливается 10 слов) Можно использовать для модуля последовательного интерфейса	Да (устанавливается 10 слов) Можно использовать для модуля последовательного интерфейса	Да (устанавливается 5 слов)	Нет	Да (устанавливается 5 слов)
	PROTOCOL MACRO	PMCR	Да (номер последовательности указывается в двоичном коде. Четыре операнда. Можно указать адрес адресуемого модуля и номер последовательного порта)	Да (номер последовательности указывается в двоичном коде. Четыре операнда. Можно указать адрес адресуемого модуля и номер последовательного порта)	Да (номер последовательности указывается в BCD. Три операнда)	Нет	Да (номер последовательности указывается в BCD. Три операнда)
	PCMCIA CARD MACRO	CMCR	Нет	Нет	Да	Нет	Нет

Параметр	Мнемонический код	Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H	
Сетевые команды	NETWORK SEND/ NETWORK RECEIVE	SEND/ RECV	Да (можно использовать для работы с центральным компьютером через соединения Host Link. Нельзя использовать для модулей последовательного интерфейса или порта RS-232C модуля CPU)	Да (можно использовать для работы с центральным компьютером через соединения Host Link. Нельзя использовать для модулей последовательного интерфейса, порта RS-232C модуля CPU или встраиваемой платы)	Да (нельзя использовать для работы с центральным компьютером через соединения Host Link)	Да (нельзя использовать для работы с центральным компьютером через соединения Host Link)	Да (нельзя использовать для работы с центральным компьютером через соединения Host Link)
	DELIVER COMMAND	CMND	Да (используется для работы с центральным компьютером через соединения Host Link. Нельзя использовать для модулей последовательного интерфейса или порта RS-232C модуля CPU)	Да (используется для работы с центральным компьютером через соединения Host Link. Нельзя использовать для модулей последовательного интерфейса, порта RS-232C модуля CPU или встраиваемой платы)	Нет	Да (можно использовать для связи с центральным компьютером через соединения Host Link)	Да (нельзя использовать для связи с центральным компьютером через соединения Host Link)
	EXPLICIT MESSAGE SEND	EXPLT	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет
	EXPLICIT GET ATTRIBUTE	EGATR	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет
	EXPLICIT SET ATTRIBUTE	ESATR	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет
	EXPLICIT WORD READ	ECHRД	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет
	EXPLICIT WORD WRITE	ECHWR	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Модули CPU до версии 2.0: нет Модули CPU версии 2.0 или выше: да	Нет	Нет	Нет
Команды для памяти данных	READ DATA FILE/ WRITE DATA FILE	FREAD/ FWRIT	Да	Да	Нет	Да (FILR/FIWW)	Нет
	READ PROGRAM FILE	FILP	Нет	Нет	Нет	Да	Нет
	CHANGE STEP PROGRAM	FLSP	Нет	Нет	Нет	Да	Нет
Команды для отображения	DISPLAY MESSAGE	MSG	Да (сообщения завершаются кодом NUL)	Да (сообщения завершаются кодом NUL)	Да (сообщения завершаются кодом CR)	Да (сообщения завершаются кодом CR)	Да (сообщения завершаются кодом CR)
	DISPLAY LONG MESSAGE	LMSG	Нет	Нет	Да (сообщения завершаются кодом CR)	Нет	Нет
	I/O DISPLAY	IODP	Нет	Нет	Нет	Да	Нет
	TERMINAL MODE	TERM	Нет	Нет	Да	Нет	Нет
Команды для работы с часами	CALENDAR ADD	CADD	Да	Да	Нет	Да	Нет
	CALENDAR SUBTRACT	CSUB	Да	Да	Нет	Да	Нет
	HOURS TO SECONDS	SEC	Да	Да	Да	Да	Да
	SECONDS TO HOURS	HMS	Да	Да	Да	Да	Да
	CLOCK ADJUSTMENT	DATE	Да	Да	Нет	Да (*1)	Нет
Отладочные команды	TRACE MEMORY SAMPLING	TRSM	Да	Да	Да	Да	Да
	MARK TRACE	MARK	Нет	Нет	Нет	Да (номер метки указывается в BCD)	Нет



Параметр	Мнемонический код	Серия CJ	Серия CS	C200HX/HG/HE	Серия CVM1/CV	CQM1H	
Команды для диагностики неисправностей	FAILURE ALARM/ SEVERE FAILURE ALARM	FAL/ FALS	Да (сообщения завершаются кодом NUL, текстовые строки сохраняются в порядке: от старшего байта к младшему, от младшего слова к старшему. Номер FAL указывается в двоичном коде)	Да (сообщения завершаются кодом NUL, текстовые строки сохраняются в порядке: от старшего байта к младшему, от младшего слова к старшему. Номер FAL указывается в двоичном коде)	Да (сообщения завершаются кодом CR, текстовые строки сохраняются в порядке: от старшего байта к младшему, от младшего слова к старшему. Номер FAL указывается в BCD)	Да (сообщения завершаются кодом CR, текстовые строки сохраняются в порядке: от старшего байта к младшему, от младшего слова к старшему. Номер FAL указывается в BCD)	Да (сообщения завершаются кодом CR, текстовые строки сохраняются в порядке: от старшего байта к младшему, от младшего слова к старшему. Номер FAL указывается в BCD)
	FAILURE POINT DETECTION	FPD	Да (сообщения завершаются кодом NUL, текстовые строки сохраняются в порядке: от старшего байта к младшему, от младшего слова к старшему. Номер FAL указывается в двоичном коде)	Да (сообщения завершаются кодом NUL, текстовые строки сохраняются в порядке: от старшего байта к младшему, от младшего слова к старшему. Номер FAL указывается в двоичном коде)	Да (сообщения завершаются кодом CR, текстовые строки сохраняются в порядке: от старшего байта к младшему, от младшего слова к старшему. Номер FAL указывается в BCD) (*1)	Да (сообщения завершаются кодом CR, текстовые строки сохраняются в порядке: от старшего байта к младшему, от младшего слова к старшему. Номер FAL указывается в BCD) (*1)	Да (сообщения завершаются кодом CR, текстовые строки сохраняются в порядке: от старшего байта к младшему, от младшего слова к старшему. Номер FAL указывается в BCD)
Прочие команды	SET CARRY/ CLEAR CARRY	STC/ CLC	Да	Да	Да	Да	Да
	LOAD FLAGS/ SAVE FLAGS	CCL, CCS	CJ1: Нет CJ1-H: Да CJ1M: Да	CS1: Нет CS1-H: Да	Нет	Да	Нет
	EXTEND MAXIMUM CYCLE TIME	WDT	Да	Да	Да	Да (*1)	Да
	CYCLE TIME	SCAN	Нет	Нет	Да	Нет	Нет
	LOAD REGISTER/ SAVE REGISTER	REGL, REGS	Нет	Нет	Нет	Да	Нет
	SELECT EM BANK:	EMBC	Да	Да	Да	Да	Нет
	EXPANSION DM READ	XDMR	Нет	Нет	Да	Нет	Нет
	INDIRECT EM ADDRESSING	IEMS	Нет	Нет	Да	Нет	Нет
	ENABLE ACCESS/ DISABLE ACCESS	IOSP, IORS	Нет	CS1: Нет CS1-H: Да	Нет	Да	Нет
Команды для преобразования адресов CV-CS	FRMCV TOCV	CJ1: Нет CJ1-H: Да CJ1M: Да	CS1: Нет CS1-H: Да	Нет	Нет	Нет	
Команды для программирования блоков	BPRG/ BEND, IF/ ELSE/ IEND, WAIT, EXIT, LOOP/ LEND, BPPS/ BPRS, TIMW, CNTW, TMHW	Да	Да	Нет	Да (*1)	Нет	
Команды для управления задачами	TASK ON/TASK OFF	TKON/ TKOF	Да	Да	Нет	Нет	Нет

**Примечание** \*1: Поддерживается только модулем CVM1 (V2).

\*2: Поддерживается только модулями CPU □□ - Z.

\*3: Продолжение выполнения той же программы поддерживается модулем CV1M версии 2.

\*4: За исключением модулей CPU CS1 и CJ1.

\*5: Только для модулей CPU CJ1M со встроенными входами/выходами. Некоторые операнды отличаются от тех, которые используются для CQM1H.

## Приложение В

### Изменения по сравнению с предшествующими системами Host Link

Системы Host Link, создаваемые на базе плат (только серия CS) и модулей последовательного интерфейса серии CS/CJ, отличаются от систем Host Link, создаваемых на базе модулей Host Link и модулей CPU, относящихся к ПЛК других серий. Эти отличия поясняются ниже.

#### Порты RS-232C

При переходе от существующей системы Host Link к новой системе, в которой предусматривается использование порта RS-232C модуля CPU серии CS/CJ, плат последовательного интерфейса (только серия CS) или модуля последовательного интерфейса (порт RS-232C модуля CS1H/G-CPU □□, порты CS1W-SCU-21, порты CS1W-SCB21, порт 1 модуля CS1W-SCB41 или порт 2 модуля CJ1W-SCU41), необходимо учитывать следующие отличия.

Предшествующие изделия	Номер модели	Изменения, необходимые для изделий серии CS/CJ	
		Подключение цепей	Остальные изменения
Модули Host Link серии C	3G2A5-LK201-E C500-LK203 3G2A6-LK201-E	25-контактный разъем заменен на 9-контактный разъем.  Изделия серии CS/CJ не поддерживают сигналы ST1, ST2 и RT и их подключать не требуется.	<b>Для систем, синхронизируемых сигналами ST1, ST2 и RT, необходимо выполнить следующие изменения.</b>  Синхронизированная передача больше не поддерживается.  Дуплексная передача с помощью изделий серии CS/CJ возможна, но необходимо внести изменения в программу связи центрального компьютера или в аппаратные средства, или и в то, и в другое.  <b>Для систем, не синхронизируемых сигналами ST1, ST2 и RT, необходимо выполнить следующие изменения.</b>  Возможно, программы центрального компьютера получится использовать без изменений, при тех же параметрах связи (например, скорость передачи). Однако, чтобы скорректировать длины текстовых строк в кадрах или другие параметры команд CS/CJ (см. примечание), программу, возможно, потребуется изменить.
	C200H-LK201	25-контактный разъем заменен на 9-контактный разъем.	Возможно, программы центрального компьютера получится использовать без изменений, при тех же параметрах связи (например, скорость передачи). Однако, чтобы скорректировать длины текстовых строк в кадрах или другие параметры команд CS/CJ (см. примечание), программу, возможно, потребуется изменить.
Модули CPU серии C	SRM1 CPM1 CPM1A CQM1-CPU□□-E C200HS-CPU□□-E C200HX/HG/HE-CPU□□-E C200HW-COM□□-E	Никаких изменений в схеме подключения не произведено.	Возможно, программы центрального компьютера получится использовать без изменений, при тех же параметрах связи (например, скорость передачи). Однако, чтобы скорректировать другие параметры команд CS/CJ, программу, возможно, потребуется изменить.

Предшествующие изделия	Номер модели	Изменения, необходимые для изделий серии CS/CJ	
		Подключение цепей	Остальные изменения
Модули CPU серии CVM1 или CV	CVM1/CV-CPU□□-E	Никаких изменений в подключении цепей не произошло.	Возможно, программы центрального компьютера получится использовать без изменений, при тех же параметрах связи (например, скорость передачи). Однако, чтобы скорректировать параметры команд CS/CJ (см. примечание), программу, возможно, потребуется изменить.
Модуль Host Link серии CVM1 или CV	CV500-LK201	<p>Порт1: 25-контактный разъем заменен на 9-контактный разъем.</p> <p>Порт 2 выбран для RS-232C:</p> <p>Сигнал SG переведен с вывода 7 на вывод 9.</p>	<p><b>Для полудуплексной передачи данных с использованием CD необходимо выполнить следующие изменения.</b></p> <p>В случае применения SEND, RECV или CMND для инициации коммуникаций на стороне ПЛК проверьте систему на отсутствие проблем синхронизации. Проверьте отсутствие проблем с синхронизацией при передаче команд центральным компьютером. В случае необходимости, перейдите на дуплексный режим передачи.</p> <p><b>Для дуплексной передачи без использования CD необходимо выполнить следующие изменения.</b></p> <p>В случае полудуплексной передачи программы центрального компьютера, возможно, получится использовать без изменений, при тех же параметрах связи (например, скорость передачи). Однако, чтобы скорректировать параметры команд CS/CJ, программу, возможно, потребуется изменить.</p>

**Примечание** Количество слов, которое может быть прочитано и записано за один кадр (т.е., длины текстовых строк) в случае использования команд C-режима (C-mode), различается у модулей Host Link серии C и плат/модулей последовательного интерфейса серии CS/CJ. Программа центрального компьютера, ранее использовавшаяся для модулей Host Link серии C, может работать некорректно, если ее использовать для ПЛК серии CS/CJ. Прежде чем использовать программу центрального компьютера, ее необходимо проверить и скорректировать длины текстовых строк в кадрах. Подробное описание смотрите в справочном руководстве *CS/CJ-series Communications Commands Reference Manual (W342)*.

## Порты RS-422A/485

При переходе от существующей системы Host Link к системе, в которой предполагается использование порта RS-422A/485 платы последовательного интерфейса серии CS (порт 2 модуля CS1W-SCB41) или модуля последовательного интерфейса серии CJ (порт 1 модуля CJ1W-SCU41), необходимо учитывать следующие различия.

Предшествующие изделия	Номер модели	Изменения, необходимые для изделий серии CS/CJ	
		Подключение цепей	Остальные изменения
Модули Host Link серии C	3G2A5-LK201-E C200H-LK202 3G2A6-LK202-E	<p>Изменилась разводка контактов.</p> <p>SDA: вывод 9 на вывод 1</p> <p>SDB: вывод 5 на вывод 2</p> <p>RDA: вывод 6 на вывод 6</p> <p>RDB: вывод 1 на вывод 8</p> <p>SG: вывод 3 на He подключен</p> <p>FG: вывод 7 на вывод "Корпус разъема"</p>	Возможно, программы центрального компьютера получится использовать без изменений, при тех же параметрах связи (например, скорость передачи). Однако, чтобы скорректировать длины текстовых строк в кадрах или другие параметры команд CS/CJ (см. примечание), программу, возможно, потребуется изменить.
Плата связи C200HX/HG/HE	C200HW-COM□□-E	Никаких изменений в подключении цепей не произошло	Возможно, программы центрального компьютера получится использовать без изменений, при тех же параметрах связи (например, скорость передачи). Однако, чтобы скорректировать другие параметры команд CS/CJ, программу, возможно, потребуется изменить.

Предшествующие изделия	Номер модели	Изменения, необходимые для изделий серии CS/CJ	
		Подключение цепей	Остальные изменения
Модули CPU серии SVM1 или CV	CVM1/CV-CPU□□-E	Никаких изменений в подключении цепей не произошло	Возможно, программы центрального компьютера получится использовать без изменений, при тех же параметрах связи (например, скорость передачи). Однако, чтобы скорректировать другие параметры команд CS/CJ, программу, возможно, потребуется изменить.
Модуль Host Link серии SVM1 или CV	CV500-LK201		

**Примечание** Количество слов, которое может быть прочитано и записано за один кадр (т.е., длины текстовых строк) в случае использования команд С-режима (C-mode), различно для модулей Host Link серии С и плат/модулей последовательного интерфейса серии CS/CJ. Программа центрального компьютера, ранее использовавшаяся для модулей Host Link серии С, может работать некорректно, если ее использовать для ПЛК серии CS/CJ. Прежде чем использовать программу центрального компьютера, ее необходимо проверить и скорректировать длины текстовых строк в кадрах. Подробное описание смотрите в справочном руководстве *CS/CJ-series Communications Commands Reference Manual (W342)*.



## Предметный указатель

### С-О

- CX-Programmer, 22
  - память файлов, 209
- DeviceNet
  - указания, 313
- Host Link-коммуникации, 278
- Online-редактирование, 336

### А

- Аварийные сигналы (сообщения)
  - аварийные сигналы, программируемые пользователем, 313
- Автоматическая загрузка при запуске, 196, 222
- Адресация
  - адреса памяти, 26
  - косвенные адреса, 28-29
  - операнды, 27
  - регистры индексов, 266
  - см. также* регистры индексов

### Б

- Базовые модули ввода/вывода
  - время срабатывания входа, 330
  - команды базовых модулей ввода/вывода, 134
- Батарейка
  - отсек, 2
  - установка, 2
- Бит "IOM Hold", 302
- Бит "Выходы ВЫКЛ", 338
- Бит "Удержание IOM", 302
- Блокировки, 24, 40, 62

### В - Г

- Внешние прерывания
  - задачи, 161, 176–178, 181
- Время
  - настройка часов, 5
- Время срабатывания входов/выходов
  - базовые модули ввода/вывода CS/CJ, 330
- Выход RUN, 303
- Выходы
  - выключение, 313, 338
- "Горячий" запуск, 301
- "Горячий" останов, 301

### Д - Е

- Данные в формате BCD, 32
- Дата
  - настройка часов, 5
- Дата программы пользователя, 307

### Даты

- программа и параметры, 306
- Двоичное значение со знаком, 32
- Двоичные данные без знака, 32
- Десятичные значения с плавающей запятой, 33
- Диагностика, 312
- Директивы ЕС, xxvi
- Директории, 199
- Дифференцированные команды, 39
- Длительность цикла
  - время исполнения задачи, 19
  - минимальная длительность цикла, 249
  - мониторинг, 250
  - настройка, 250
- Емкость программы, 44

### З

- Загрузка программы, 334
- Задачи обработки прерываний, 157, 160, 175–186
  - предварительные указания, 184
  - приоритет выполнения, 182
  - сопутствующие флаги и слова, 183
- Задачи, 12, 155
  - введение, 160
  - взаимосвязь с программными блоками, 174
  - время выполнения, 19
  - выполнение, 166
  - задачи обработки прерываний, 157, 160, 176
  - команды управления задачами, 154
  - назначение, 173
  - номера задач, 167
  - ограничения, 168
  - описание, 14
  - преимущества, 156
  - примеры, 171
  - свойства, 156
  - см. также* задачи обработки прерываний
  - см. также* циклические задачи
  - создание задач, 187
  - состояние, 17
  - таймеры, 168
  - условия выполнения, 162
  - флаги, 169
  - функционирование флагов условий, 168
  - циклические задачи, 157, 160
- Задержка обнаружения отключения питания, 303
- Запланированные прерывания
  - задачи, 160, 175, 178–179
  - применение таймера, 299
- Запуск
  - "горячий" запуск и останов, 301
  - автоматическая загрузка данных, 196, 222
- Защита от записи, 305
- Защита от чтения/записи, 306

Значения (данные) с плавающей запятой  
команды математических операций с плавающей запятой, 113  
команды математических операций с плавающей запятой, с двойной точностью, 117

## **И**

Имена файлов, 194  
Инициализация  
карты памяти, 240  
память ввода/вывода, 10  
память файлов EM, 240

## **К**

Каналы последовательной связи с ПЛК (Serial PLC Links), 285–286  
настройки ПЛК, 289  
резервируемые слова, 288  
сопутствующие флаги, 290  
Карты памяти, 7, 190  
инициализация, 240  
предварительные указания, 191  
Команды  
варианты выполнения, 36  
входные и выходные команды, 23, 25  
дифференциация входов, 37  
дифференцированные команды, 39  
команды диагностики неисправностей, 143  
команды для базовых модулей ввода/вывода, 134  
команды для обработки табличных данных, 117, 121  
команды для обработки текстовых строк, 151  
команды для программирования блоков, 145  
команды для связи по последовательному интерфейсу, 137  
команды для создания подпрограмм, 129  
команды для управления задачами, 154  
команды логических операций, 110  
команды математических операций с плавающей запятой, 113  
команды отладки, 142  
команды перемещения данных, 91  
команды последовательного ввода, 74  
команды последовательного вывода, 76  
команды последовательного управления, 79  
команды преобразования данных, 104  
команды сдвига данных, 94  
команды сравнения, 87  
команды увеличения, 98  
команды уменьшения, 98  
команды управления данными, 125  
команды управления дисплеем, 141  
команды управления задачами, 165  
команды управления памятью файлов, 140  
команды управления прерываниями, 130

команды управления счетчиками, 83  
команды управления таймерами, 83  
команды управления часами, 141  
команды шагового управления, 134  
ограничения в задачах, 168  
однократные команды, 39  
операнды, 24  
основные команды, 23  
память файлов, 213  
программные блоки, 165  
размещение команд, 25  
регистры индексов, 270  
сетевые команды, 138  
символьные математические команды, 199  
синхронизация, 39  
специальные математические команды, 112  
условия выполнения команд, 23  
условия выполнения, 36  
циклы, 24, 62  
Команды FINS  
память файлов, 211  
список, 281  
Команды Host Link, 279  
Команды для работы с табличными записями, 273  
Команды работы с диапазонами данных, 273  
Команды с различием условий выполнения, 39  
Команды, различающие переключение ВКЛ->ВЫКЛ, 37  
Команды, различающие переключение ВЫКЛ->ВКЛ, 37  
Консоли программирования  
память файлов, 210  
Константы  
операнды, 30

## **М**

Максимальная длительность цикла, 250  
Математические операции  
команды математических операций с плавающей запятой, 113  
символьные математические команды, 99  
специальные математические команды, 112  
Мгновенное обновление, 36, 41, 252  
Метод полного обновления, 286  
Минимальная (фиксированная) длительность цикла, 249  
Мнемонические обозначения, 45  
ввод, 49  
Модули  
профили, 308  
Модули CVM1  
изменения в характеристиках связи, 368-369  
Модули Host Link  
изменения в характеристиках связи, 368  
Модули Host Link серии C  
изменения в характеристиках связи, 367  
Модули серии CV  
изменения в характеристиках связи, 368-369

Модули серии С  
изменения в характеристиках связи, 368  
Модуль CPU  
внутренняя структура, 6  
объемы программ, 44  
основной принцип работы, 158  
функционирование, 1  
Мониторинг  
направление переключения, 335  
удаленный мониторинг, 308

## **Н**

Настройка  
*см. так же* установка  
Настройка параметров  
настройка параметров запуска, 301  
*см. так же* настройка переключателей  
Настройки ПЛК, 7

## **О**

Области данных  
адресация, 26  
Область параметров, 7  
файлы, 207  
Обнаружение точки сбоя, 314  
Обновление  
IORF(097), 43, 185, 253  
мгновенное обновление, 36, 41, 252  
обновление входов/выходов, 41, 252  
циклическое обновление, 41, 252  
Обновление IORF(097), 43, 253  
задачи обработки прерываний, 185  
Обновление входов/выходов, 41  
Обновление данных, 286  
Обнуление памяти, 4  
Обработка стека, 271  
Обслуживание периферии  
приоритетность обслуживания, 322  
Операнды  
константы, 30  
описание, 24  
текстовые строки, 30  
указание, 27  
Опрашиваемые модули  
параметры, 289  
Опрашивающий модуль  
настройка, 289  
Отладка, 312, 334  
команды диагностики неисправностей, 143  
команды отладки, 142

Ошибки  
обнаружение точки сбоя, 314  
ошибка "недопустимая команда", 68  
ошибка "переполнение UM", 68  
ошибка выполнения команды, 68  
ошибка доступа, 68  
ошибка при вводе команды, 66  
ошибки программирования, 70  
ошибки, программируемые пользователем, 313  
протокол ошибок, 312  
фатальная ошибка, 70  
Ошибки программы, 70

## **П**

Память  
обнуление, 4  
*см. так же* память ввода/вывода  
*см. так же* память пользователя  
*см. так же* память файлов  
структурные схемы памяти модуля CPU, 7  
Память ввода/вывода, 7  
адресация, 26  
задачи, 167  
инициализация, 10  
Память файлов EM, 190  
инициализация, 240  
операции, 245  
*см. так же* память файлов  
Память файлов, 189  
имена файлов и типы файлов, 194  
команды управления памятью файлов, 140, 213  
обращение к директориям, 199  
применение, 206, 240  
управление файлами, 208  
файлы параметров, 207  
файлы программ, 207  
функции, 189  
Параметры, 307  
Переходы, 40, 62  
Плата связи C200H/HG/HE  
изменения в характеристиках связи, 368  
Платы связи C200H, 368  
ПЛК  
сравнение, 343  
ПЛК серии C200HX/HG/HE  
сравнение, 343  
ПЛК серии CV  
сравнение, 343  
Подпрограммы, 62  
Порты RS-232C  
изменения по сравнению с предшествующими изделиями, 367  
Порты RS-422A/485  
изменения по сравнению с предшествующими изделиями, 368



Предварительные указания, хіх  
задачи обработки прерываний, 184  
общие указания, хх  
по безопасности, хх  
по обновлению входов/выходов, 9  
по применению, ххii  
по программированию, 57  
по условиям эксплуатации, ххii  
Предшествующие изделия  
сравнение, 368  
Прерывания от входов/выходов  
задачи, 160, 175–178  
Прерывания по отключению питания  
задачи, 160, 175, 179–181  
Прерывания по питанию  
отключение, 303  
Прерывания, 251  
отключение, 186  
приоритет выполнения задач обработки  
прерываний, 182  
*см. так же* внешние прерывания  
Применение  
память файлов, 206  
предварительные указания, ххii  
Принудительная установка битов  
отладка, 334  
Принудительный сброс битов  
отладка, 334  
Пробный запуск, 334  
Программа пользователя, 6-7  
*см. так же* программирование  
Программирование, 21  
дистанционное программирование, 308  
емкость программы, 44  
загрузка программы, 334  
задачи и программы, 157  
защита программы, 305  
защита программы, 305  
мнемонические обозначения, 45  
назначение задач, 173  
ограничения, 47  
основные принципы, 44  
ошибки, 66  
примеры, 52  
проверка программ, 66  
программы и задачи, 12, 22  
программные блоки, 24, 62  
ограничения, 65  
путь для тока, 23  
размещение команд, 25  
*см. так же* программные блоки  
структура программы, 12, 15, 45  
указания, 57  
шаговое программирование, 62  
ограничения, 64

Программные блоки, 24, 62, 65  
взаимосвязь с задачами, 174  
команды для программирования блоков, 145  
Программы  
*см. так же* программирование  
Протокол Host Link, 278  
Протокол ошибок, 312  
Протоколирование данных (выборка), 339  
"Путь для тока"  
описание, 23

## **Р**

Работа  
модуль CPU, 1  
основное функционирование, 158  
отладка, 334  
пробное включение, 334  
Регистры индексов, 29, 266  
Режим MONITOR  
описание, 9  
Режим PROGRAM  
описание, 8-9  
Режим RUN  
описание, 9  
Режим запуска, 302  
Режим обновления только на опрашивающем  
модуле, 286  
Режим обновления, 291  
таймеры и счетчики, 291  
Режим приоритетного обслуживания периферии, 322  
Режимы работы  
описание, 8  
режим запуска, 11  
Резервирование областей ввода/вывода  
настройка первого слова, 331  
Резервное сохранение данных, 309

## **С**

Связь  
беспротокольный режим, 283  
команды для связи по последовательному  
интерфейсу, 137  
*см. также* связь по последовательному интерфейсу  
сообщения, 282  
Связь без применения протокола, 283  
Связь по последовательному интерфейсу  
функции, 276  
Сети  
сетевые команды, 138  
Сигнализация неисправностей, 313  
Символы ASCII, 31  
Скоростные входы, 257  
Сообщения, 282

Состояние "выполняемая задача",  
описание, 17  
Состояние "ожидание" (standby)  
описание, 17  
Сравнение  
с предшествующими изделиями, 368  
Средства программирования  
операции с задачами, 187  
память файлов, 208  
Структура программы, 45  
Счетчики  
режим обновления, 291

## **Т**

Табличные данные  
обработка, 273  
Таймеры, 291  
применение запланированных прерываний, 299  
Текстовые строки  
команды обработки текстовых слов, 151  
операнды, 30  
Типы файлов, 194

## **У**

Указания по безопасности, хх  
Условия выполнения  
варианты выполнения, 36  
задачи, 162  
Условия выполнения команд  
описание, 23  
Условия эксплуатации  
указания, ххii  
Установка  
первоначальная настройка, 2, 5  
предварительные указания, ххii

## **Ф**

Файлы данных, 206  
Файлы программ, 207  
Флаг "Больше, чем", 61  
Флаг "Меньше, чем", 61  
Флаг "Отрицательное значение", 61  
Флаг "Ошибка задачи", 171  
Флаг "Первое выполнение задачи", 170  
Флаг "Равенство", 61  
Флаг переноса, 61  
Флаги задач, 169  
Флаги условий, 57  
применение в задачах, 168  
Флаги, 24  
флаги условий, 57  
Флеш-память, 309  
Форматы данных, 32  
Функция отключения выходов, 313

## **Ц, Ч, Ш**

Цикл FOR-NEXT, 62  
Циклические задачи, 157, 160  
состояние "выполнение" (RUN), 163  
состояние "готовность" (READY), 163  
состояние "ожидание" (WAIT), 163  
состояние "отключенная задача" (INI), 163  
состояние, 163  
Циклическое обновление, 41, 252  
Циклы  
циклы FOR/NEXT, 62  
Часы, 304  
команды управления часами, 141  
настройка часов, 5  
Шаговое программирование, 62



## Перечень версий

Версия Руководства указывается в конце Номера каталога на титульной странице Руководства.

Каталог № W394-RU2-06



Обозначение версии

В таблице ниже показаны изменения, которые претерпело данное Руководство после выхода его оригинальной версии. Номера страниц относятся к предыдущим изданиям.

Обозн. версии	Дата	Пересмотренная версия
01	Апрель 2001	Оригинальная версия
02	Октябрь 2001	По всему Руководству добавлены сведения о скоростных модулях CPU серии CS и серии CJ (CS1G/H-CPU□□ и CJ1G/H-CPU□□H).
03	Июль 2002	По всему Руководству добавлены сведения о модулях CPU CJ1M. Для обозначения "программируемый контроллер" аббревиатура PC заменена на PLC (ПЛК). Также внесены следующие изменения: <b>Стр. xvi и xviii:</b> Добавлено предупреждение. <b>Стр. xix:</b> Изменен пункт 2 снизу страницы. <b>Стр. 28:</b> Изменено содержание текста. <b>Стр. 167:</b> Изменен пример программирования. <b>Стр. 168, 169, 265 и 266:</b> Добавлены сведения об источниках питания постоянного тока. <b>Стр. 179:</b> Добавлены предупреждения относительно карт памяти. <b>Стр. 229:</b> Изменена иллюстрация. <b>Стр. 262:</b> Добавлены сведения о способах обновления таймеров/счетчиков. <b>Стр. 273:</b> Добавлено указание касательно DeviceNet. <b>Стр. 301:</b> Исправлены скорости обработки для модулей. <b>Стр. 304:</b> Исправлено время срабатывания прерывания. <b>Стр. 320:</b> Изменены сведения о поддержке серий CJ1 команд IOSP/IORS.
04	Сентябрь 2002	По всему тексту добавлены сведения о модулях CPU CJ1D. Также внесены следующие изменения: <b>Стр. xv:</b> Добавлено предупреждение относительно функции резервирования. <b>Стр. xvi:</b> Заменено 1-е предупреждение и исправлены типы модулей CPU в предупреждениях относительно рабочего режима при запуске. <b>Стр. xviii:</b> В середине страницы добавлено предупреждение касательно порта RS-232C. <b>Стр. 6:</b> Добавлены сведения о версиях CX-Programmer. <b>Стр. 184:</b> В таблице исправлена информация о карте памяти. <b>Стр. 274:</b> Сверху страницы изменен диапазон максимальных номеров модуля. <b>Стр. 294:</b> Примечание заменено на предупреждение и изменено его содержание. <b>Стр. 303:</b> Изменено содержание примечания 3. <b>Стр. 304:</b> В таблицу добавлена информация.
05	Апрель 2003	<b>Стр. 44:</b> Изменен 1-й основной принцип создания лестничных диаграмм. <b>Стр. 45:</b> Изменена информация во 2-м ограничении. <b>Стр. 46:</b> Изменена информация в 4-м ограничении. 6-е ограничение удалено. <b>Стр. 54:</b> Изменена информация о ветвях, требующих внимательного отношения или изменения. <b>Стр. 157 и 158:</b> Изменена таблица различий между дополнительными и обычными циклическими задачами. <b>Стр. 220:</b> Добавлено примечание 5. <b>Стр. 226:</b> Изменена информация о поддерживаемых модулях. <b>Стр. 233:</b> Изменена информация о прерываниях по исчезновению питания во время обращения к памяти файлов. <b>Стр. 264:</b> Добавлено примечание. <b>Стр. 273:</b> Добавлено примечание. <b>Стр. 276:</b> Изменена информация в таблицах. Под 1-й и под 2-й таблицами добавлены примечания. <b>Стр. 310:</b> Изменена информация под 1-й таблицей относительно CX-Programmer. <b>Стр. 314:</b> Во 2-е примечание добавлена информация. <b>Стр. 326:</b> Добавлена информация в шаг 1 процедуры протоколирования данных.

---

## Перечень версий

---

Обозн. версии	Дата	Пересмотренная версия
06	Декабрь 2003	<p>Добавлена информация о новых функциях, поддерживаемых модулями CPU в новых исполнениях (в список добавлено несколько номеров).</p> <p><b>Стр. xvi ... xx:</b> Обновлено информация по PLP.</p> <p><b>Стр. 72:</b> Сверху таблицы добавлены примечания, а также добавлены команды AND NOT и OR NOT.</p> <p><b>Стр. 160, 201, 202, 228, 293 и 320:</b> Добавлены примечания.</p> <p><b>Стр. 189:</b> Удалена информация о CX-Programmer версии 1.0 или выше.</p> <p><b>Стр. 191:</b> Добавлена информация сверху и внутри таблицы, а также добавлена новая таблица.</p> <p><b>Стр. 197, 199, 216 ... 218, 230, 231:</b> Добавлена информация по замене без использования файла области параметров.</p> <p><b>Стр. 226:</b> Расширена таблица.</p> <p><b>Стр. 294:</b> Добавлен новый раздел 6-6-8.</p> <p><b>Стр. 337 ... 339, 342, 344, 345 и 347:</b> В таблицу добавлены строки.</p> <p><b>Стр. 339:</b> Добавлена информация о BCMP2.</p>

**OMRON CORPORATION**

FA Systems Division H.Q.  
66 Matsumoto  
Mishima-city, Shizuoka 411-8511  
Japan  
Tel: (81)559-77-9181/Fax: (81)559-77-9045

**Regional Headquarters**

**OMRON EUROPE B.V.**

Wegalaan 67-69, NL-2132 JD Hoofddorp  
The Netherlands  
Tel: (31)2356-81-300/Fax: (31)2356-81-388

**OMRON ELECTRONICS LLC**

1 East Commerce Drive, Schaumburg, IL 60173  
U.S.A.  
Tel: (1)847-843-7900/Fax: (1)847-843-8568

**OMRON ASIA PACIFIC PTE. LTD.**

83 Clemenceau Avenue,  
#11-01, UE Square,  
Singapore 239920  
Tel: (65)6835-3011/Fax: (65)6835-2711

# OMRON

Авторизованный дистрибьютор: