SIEMENS

SIMATIC S5

Программируемый контроллер AG S5-135U (S и R процессоры)

Руководство

№ заказа 6ES5 998-0UL01-1C
соответствует нем. изданию 15

Содержание		Рег
Инструкции, информации Предложения/исправления	C79000-R8500-C407	
Рук-во по эксплуатации: Центральное устройство	C79000-B8500-C260-07	**1**
S – процессор 921	C79000-B8500-C262-05	**2**
R – процессор 922	C79000-B8500-C348-04	**3**
RAM – модуль 377	C79000-B8500-C615-02	**4**
Координатор 923 A Координатор 923 C	C79000- B8500-C263-07 C79000- B8500-C349-06	**5**
Руководство по программированию: S – процессор	C79000- B8500-C264-03	**6**
Руководство по программированию: R – процессор	C79000- B8500-C364-02	**7**
Рук-во по эксплуатации: Многопроцессорный режим работы	C79000- B8500-C500-02	**8**
Рук-во по эксплуатации: Многопроцессорная коммуникация	C79000- B8500-C468-04	**9**
Указания по монтажу: Программируемые контроллеры	C79000- B8500-C452-03	**10**

СИМЕНС АО

Указания по технике безопасности

Данное руководство содержит указания, которые вы должны соблюдать для обеспечения собственной безопасности, а также защиты от повреждений продукта и связанного с ним оборудования. Эти замечания выделены в руководстве предупреждающим треугольником и помечены, как показано ниже, в соответствии с уровнем опасности:



Опасность

указывает, что если не будут приняты надлежащие меры предосторожности, то это приведет к гибели людей, тяжким телесным повреждениям или существенному имущественному ущербу.



Предупреждение

указывает, что при отсутствии надлежащих мер предосторожности это может привести к гибели людей, тяжким телесным повреждениям или к существенному имущественному ущербу.



Предостережение

указывает, что возможны легкие телесные повреждения и нанесение небольшого имущественного ущерба при непринятии надлежащих мер предосторожности.

Замечание

привлекает ваше внимание к особо важной информации о продукте, обращении с ним или к соответствующей части документации.

Квалифицированный персонал

Оборудование может вводиться в эксплуатацию и обслуживаться только **квалифицированным** персоналом. Квалифицированный персонал, в соответствии с замечаниями по безопасности, приведенными в данном руководстве, это люди, которые имеют право вводить в действие, заземлять и маркировать электрические цепи, оборудование и системы в соответствии со стандартами техники безопасности.

Надлежащее использование

Примите во внимание следующее:



Предупреждение

Это оборудование может использоваться только для применений, описанных в каталоге и технической документации, и в соединении только с теми устройствами или компонентами других производителей, которые были одобрены или рекомендованы фирмой Siemens.

Запуск не должен производиться, пока не установлено, что станок, на котором должен быть установлен данный компонент оборудования, удовлетворяет руководящим указаниям 89/392/EEC.

Безаварийная и безопасная работа данного продукта предполагает надлежащую транспортировку, надлежащее хранение, установку и монтаж, а также аккуратную эксплуатацию и обслуживание.

Товарные знаки

Зарегистрированные товарные знаки SIEMENS AG перечислены в предисловии.

Некоторые другие обозначения, использованные в этих документах, также являются зарегистрированными товарными знаками; права собственности могут быть нарушены, если они используются третьей стороной для своих собственных целей.

Copyright© Siemens AG 2000 Все права сохранены

Воспроизведение, передача или использование этого документа или его содержания не допускаются без письменного разрешения. Нарушители будут нести ответственность за нанесенный ущерб. Все права, включая права, вытекающие из предоставления патента или регистрации практической модели или конструкции, сохраняются.

Siemens AG
Автоматизация и привода
Человеко-машинный интерфейс SIMATIC
п/я 4848, D- 90327, Нюрнберг

Отказ от ответственности

Мы проверили содержание этого руководства на соответствие с описанным аппаратным и программным обеспечением. Так как отклонения не могут быть полностью исключены, то мы не можем гарантировать полного соответствия. Однако данные, приведенные в этом руководстве, регулярно пересматриваются и все необходимые исправления вносятся в последующие издания. Мы будем благодарны за предложения по улучшению содержания.

©Siemens AG 2000
Технические данные могут быть изменены.

SIEMENS

6ES5135-3KA13

6ES5135-3KA21

SIMATIC S5 6ES5135-3KA31

135U-центральное устройство 6ES5135-3KB41

Инструкция Каталоговый №ю C79000-B8500-C260-07

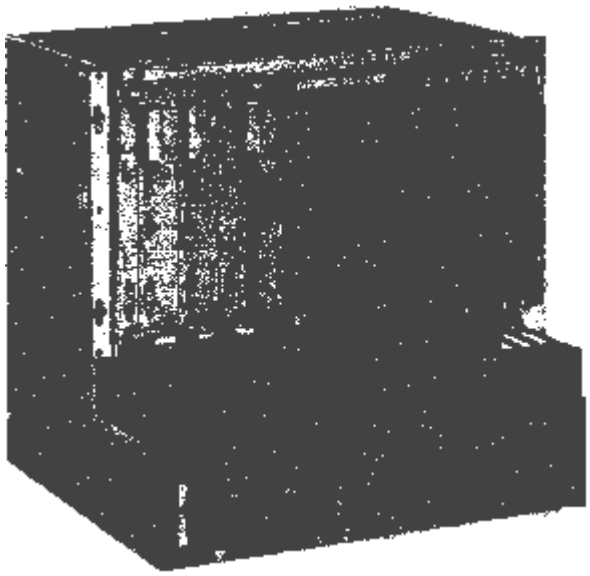


Рис.1 Центральное устройство S5-135U с S-процессором

Содержание Страница

1. Техническое описание
1.1 Область применения
1.2 Структура
1.3 Принцип действия
1.4 Технические данные

2 Монтаж
2.1 Монтаж центрального устройства
2.2 Монтаж блока питания
2.3 Монтаж дополнительного 15—вольтового модуля питания
2.4 Подключение напряжения питания
2.5 Монтаж соединительный кабелей
2.6 Указания по монтажу

3 Эксплуатация
3.1 Общие замечания
3.2 Органы управления и элементы индикации
3.3 Назначение и расположение перемычек в блоке питания
3.4 Реакция блока питания при неисправностях
3.5 Запуск и Функциональный тест
3.5.1 Запуск контроллера в однопроцессорном режиме
3.5.2 Запуск контроллера в многопроцессорном режиме

4. Обслуживание
4.1 Общие указания
4.2 Замена буферной батареи и вентилятора
4.3 Назначение контактов заднего разъема блока питания
4.4 Назначение контактов задних разъемов модулей
4.5 Расположение сигналов прерываний на внутренней шине

5 Комплектующие……..38

1 Техническое описание

1 Техническое описание

Область применения

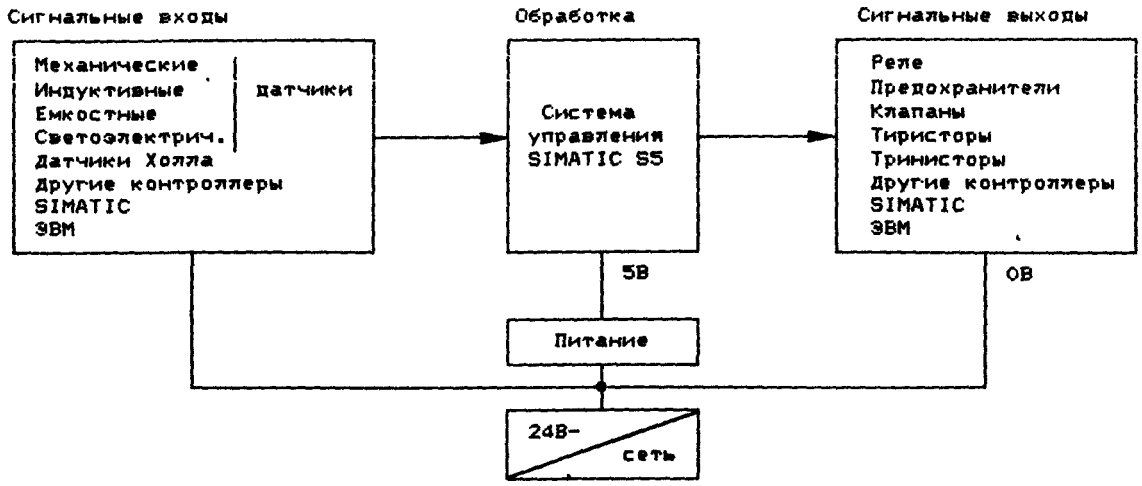


Рис.2 Область применения программируемого контроллера S5-135U

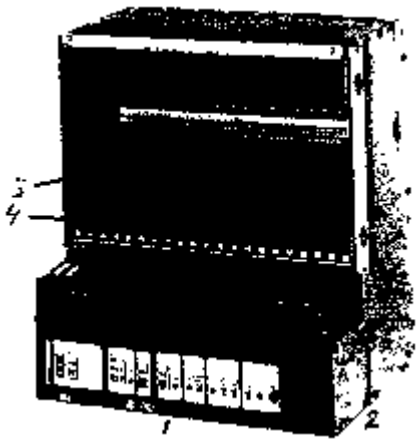
Программируемый контроллер S5-135U - представитель семейства программируемых контроллеров SIMATIC S5.

Возможность применения в контроллере различных процессоров (S-, P.-, M- процессоров и/или CPU 928) позволяет решать самые разнообразные задачи и делает AO 135U универсальным. Контроллер 6без труда может быть приспособлен к решению определенной задачи благодаря своим широким функциональным возможностям (в нем можно использовать от 1 до 4 процессоров).

.Этот контроллер может применяться в сочетании с другими типами устройств управления семейства SIMATIC S5 (.как программируемыми, так и на жесткой логике).

Для программирования, документирования и тестирования программ применяются соответствующие программирующие устройства. Программа пользователя пишется на языке STEP 5, ассемблере или языках высокого уровня.

1.2.Структура



1. Источник питания

2. Кабельный тракт
3 Разъем
4. Направляющие

Рис.3 Центральное устройство 6ES5135--3KA13

Контроллер S5-135U состоит из рамы с 21 установочным местом для плат и блока питания, который конструктивно встроен в один блок с вентиляторами. Платы вставляются непосредственно в каркас.

Кабельный тракт для укладки проводов расположен спереди над блоком питания.

Заказной No.	Блок питания[1)] с вентиляторами	Заказной No. блока питания
6ES5 135-3KA13	Вход: 220/11 OB AC Выход: 5B/18A DC 24B/0,8A DC	6ES5 955-3LC14
6ES5 135-3KA21	Вход: 220/110B AC Выход: 5B/18A DC 24B/0.8A DC	6ES5 955-3LF12
6ES5 135-3KA31	Вход: 24B DC Выход: 5B/ISA DC 24B/0.8A DC	6ES5 955-3NC13
6ES5 135-3KB41	Вход: 24B DC Выход: 5B/10A DC 24B/0.8A DC	6ES5 955-3LA12

DC - постоянный ток

AC - переменный ток

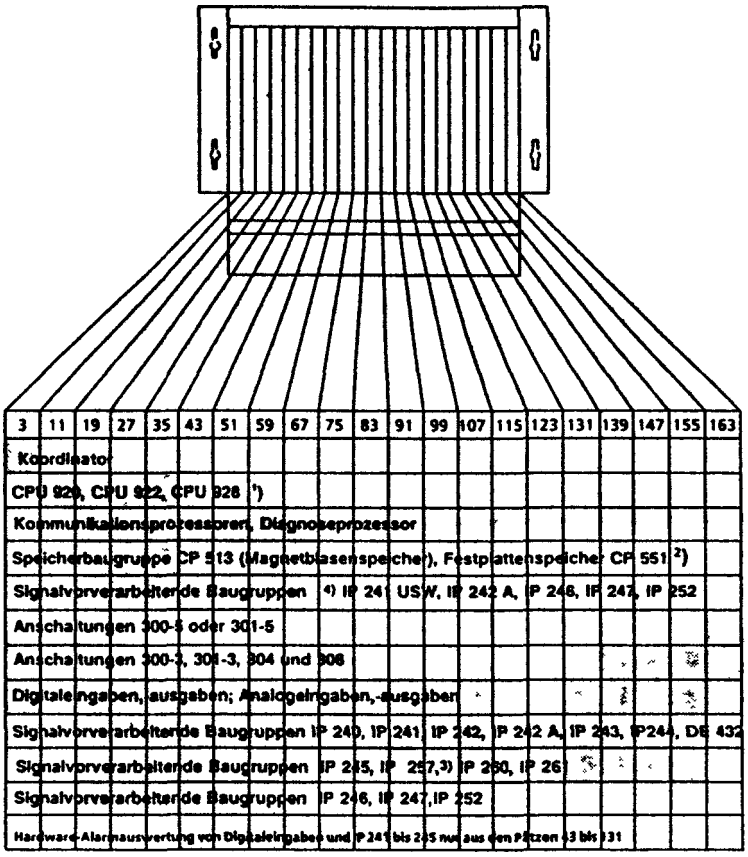
Если установочных мест в центральном устройстве не хватает для того, чтобы решить задачу, то применяется устройство расширения (см. Руководство по эксплуатации устройств расширения).

При адресации области P- или Q- периферии надо помнить следующее: Область Q можно адресовать только через модуль связи с устройством расширения (напр. AS 300, As 301), поэтому модули входов и выходов с адресами в области Q можно устанавливать только в устройства расширения

Адреса, которые используются в области Q, не могут быть использованы в центральном устройстве во избежание двойной адресации.

[1)] 15B-модуль (6ES5 956-OAA11) может быть вмонтирован в любой блок питания. Его наличие необходимо в том случае, когда в конфигурацию контроллера входит коммуникационный процессор CP535. Суммарный ток потребления источников на 24B и 15B не может превышать максимальный том 0,8 A или 2,8 A.

Схема возможного размещения плат



Координатор

CPU 920, CPU 922, CPU 928 [1]

Коммуникационные процессоры, процессоры диагностики

Модуль памяти CP 513 (память на магнитных доменах) память типа "винчестер" CP55I [2]

Модули предварительной обработки сигнал [4] IP 241 USW, IP 242A, IP 246, IP 247, IP 252

Модули связи 300-5 или 301-5

Подули связи 300-3, 301-3, 304 и 308

Цифровые входы/выходы; аналоговые входы/выходы

Модули предварительной обработки сигнала IP 240, IP 241, IP 242, IP 242A, IP 243, IP 244, DE 432

Модули предварительной обработки сигнала IP 24S, IP 257,[5] IP 260, IP 261

Модули предварительной обработки сигнала IP 246, IP 247, IP 252

Прием сигналов аппаратного прерывания с цифровых входов и IP 241 по 245

возможен только местах с 43 по 131

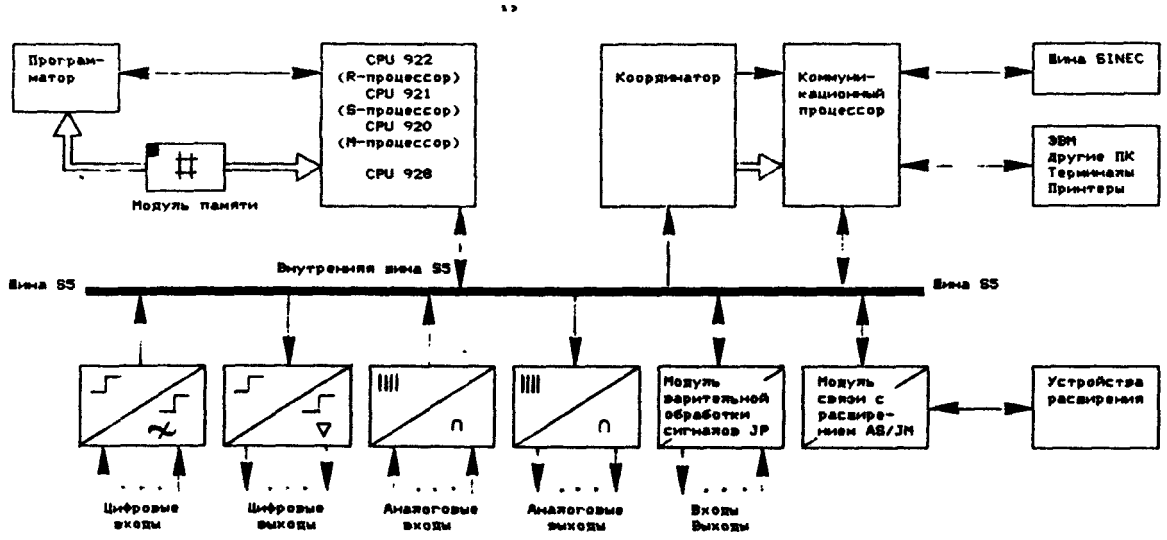
[1] Модуль двойной ширины: занимает дополнительно еще одно установочное место справа от основного.

[2] Модуль пятикратной ширины: занимает дополнительно еще четыре установочных места справа от основного.

[3]) Соответствующие модули цифровых входов/выходов 482-4 занимают до четырех мест установки справа от основного.

[5] Адресация буферов связи

1.3 Принцип действия



[1] В многопроцессорном режиме можно установить до 4 CPU

Рис.4 Структура центрального устройства S5—135U

CPU устроены так, что могут работать и в однопроцессорном и в многопроцессорном режимах.

o Однопроцессорный режим

Программа пользователя обрабатывается циклически. Доступ ко входам и выходам возможен по S5- шине в любое время.

o Многопроцессорный режим

Программируемый контроллер S5-135U - многопроцессорное устройство, в котором каждый CPU решает свою специфическую задачу.

– CPU 921: S-процессор	Предназначен, в основном для быстрой обработки двоичных сигналов (задачи управления). Может также выполнять вычислительные Функции и регулирование.
– CPU 922: R-процессор	Предназначен, в основном, для быстрой обработки слов (вычисления и регулирование.'). Возможна также обработка двоичных сигналов.
– CPU 928:	Универсальный процессор. Может вести как быструю обработку двоичных сигналов (задачи управления), так и слов (вычисления и регулирование").
– CPU 920: M-процессор	Предназначен для обработки значений измерений, решения арифметических и статистических задач. Программируется на языках BASIC, PASCAL или ASM 86/196.
– Координатор: KOR 923	Предназначен для координации работы от 2 до 4 процессоров в многопроцессорном режиме.
–Коммуникационный процессор:	Предназначен для диагностики и мониторинга процесса, связи с другими системами.

При использовании нескольких R, M или S - процессоров или CPU 928 задача автоматизации может быть наглядно разделена на составные части. Каждый процессор выполняет свою задачу независимо от других. Это повышает скорость обработки. Каждый CPU может быть запущен независимо от других. В контроллере могут работать до 4 CPU.

Обмен данными между CPU и модулями входов/выходов осуществляется по S5-шине. Принадлежность входных/выходных модулей отдельным CPU в многопроцессорном режиме устанавливается согласно спискам адресов, которые заносятся в блок данных DB 1. В однопроцессорном режиме DB1 может использоваться для оптимизации времени выполнения программы. Поэтому DB1 нельзя использовать как обычный блок данных.

Координатор в циклическом режиме предоставляет каждому процессору доступ к SS-шине. Обмен информацией между процессорами осуществляется через координатор, который имеет для этих целей область маркеров межпроцессорного обмена (923A) и память для "многопроцессорной коммуникации" (только 923C).

o Память программ

Программа пользователя хранится в субмодулях памяти. В каждый процессор можно вставить по одному такому субмодулю. Субмодули могут быть типа:

- RAM (память со свободным доступом для чтения и записи)
- RAM с буферной батареей
- EPROM (память с ультрафиолетовым стиранием).

RAM – модуль вставляется в процессор и затем может быть запрограммирован программатором по линии связи контроллер-программатор. Модуль EPROM программируется только непосредственно на программаторе через специальный разъем.

o Связь с другими устройствами

- Программатор

  Программаторы подключаются непосредственно к процессорам или координатору 923 C для программирования или запуска. Координатор 923 C нельзя использовать в качестве PG-мультиплексора в модификациях центрального устройства 6ES5 135-3KA12 и -3KB12.

- Стандартные периферийные устройства, компьютеры

  Коммуникационные процессоры CP (максимум 8 шт.) позволяют организовать независимые друг от друга каналы связи с:

  - стандартными периферийными устройствами, такими как принтер, клавиатура, терминал;
  - ЭВМ;
  - другими программируемыми контроллерами.

  Необходимые для текстов и изображений данные программируются в субмодулях RAM или EPROM для каждого коммуникационного процессора.

o Функции работы CPU (кроме M процессора)

Режим управления	Функция	Процессор 6ES5…….
Ручной новый пуск со сбросом	Все маркеры, таймеры, счетчики и область отображения процесса стираются. Программа пользователя начинает обрабатываться с начала.	Все модули
Ручной новый пуск с сохранением маркеров	Состояние маркеров сохраняется; таймеры, счетчики и область отображения процесса стираются. Программа пользователя обрабатывается с начала.	921-3UA11 921-3UA12
Автоматический новый пуск с сохранением маркеров	После сброса напряжения питания контроллер автоматически переходит в рабочий режим. В остальном этот режим напоминает ручной новый пуск с сохранением маркеров.	921-3UA11 921-3UA12
Ручной перезапуск Автоматич. перезапуск	Состояние маркеров, таймеров, счетчиков и область отображения процесса на время останова сохраняются. Обработка программы пользователя продолжается из точки прерывания.	921-3UA21 922-3UA11
Общий сброс	Вся память стирается. Контроллер переходит в начальное состояние. Если используется модуль RAM, необходимо загрузить программу, если EPROM, - то возможен новый пуск со сбросом. Общий сброс может быть выполнен с помощью переключателя на модуле CPU, программатора или прибора диагностики/OP.	Все модули

1.4 Технические данные

Соответствует правилам безопасности	VDE 0160
Класс защиты	1 I
Тип защиты	P 20
Рабочая температура	от 0 до 55 °C
Допустимая температура транспортировки и хранения	от -40 до 70 °C
Относительная влажность	макс. 95% при 25 °C без выпадения росы
Допустимая высота над уровнем моря	макс. 3500 M
Требования *к* установке и размещению	допускается встраивание в стационарные установки, которые не подвержены вибрации; допускается размещение на борту корабля или а/машины при соблюдении правил транспортировки (не допускается размещение непосредственно на двигателе)
Вес	не более 14 кг
Размеры	482,6. мм x 432 мм x 310 мм

Источник питания 6ES5 955-3LC14

Входные характеристики

Напряжение питания, Uen	230/115 В переменного тока +10% /-18.7% [1)]
Уровень срабатывания сигнализации напряжения Ue	< 187 (93) В переменного тока
Частота питающего напряжения fe, Гц	от 48 до 6.3
Входной ток с номинальной на грузкой при питающем напряжении 230 (115) В, Ien	1.25 (2.5) А
Бросок тока при включении (предельный ток включения), Ie_{max}	100 (50) А
КПД при номинальной нагрузке с вентиляторам	тип. 61%
Максимальное время отключения питающего напряжения	> 5 мс
COS Ф питающего напряжения	0.65
Входной предохранитель	быстродействующий 4 А; 250В; 6.3мм х 32мм; место установки F26

Выходные характеристики 1

Напряжение под нагрузкой Uan1	5.1В постоянного тока +/-0.5%
Диапазон выходных напряжений	(0.95...1.05) х U_{AN1}
Ток под нагрузкой I_{AN1}	18 А
Пульсации напряжения	не более 1% U_{A1}
Отклонение напряжения при резком изменении нагрузки на 50%	не более 5% U_{A1}
Время восстановления при 100% I_N	не более 5 мс
Защитное отключение при повышении напряжения U_{AI} более	6В +/-5%
Уровень срабатывания сигнализации понижения напряжения U_{AI} менее	4.75В +5%
Ограничение тока при перегрузке по току	(1.05...1.15) I_{AN1}

[1)] Выбирается переключателем сетевого напряжения

Выгодные характеристики 2

Номинальное напряжение U_{AN3}	24В пост. тока +25/-17%
Номинальный ток I_{AN3}	0.8 A
Суммарный ток нагрузки выходов 24АВ и 15В	не более 0.8 A
Пульсации напряжения	не более 0.5% U_{AZ}
Защита от перегрузки по току	1.5A быстродейств. предохр. 250В; 6.3мм *x* 32мм; место установки F90

Выгодные характеристики 3 с дополнительным модулем

Номинальное напряжение U_{AN4}	15В пост. тока +/-5%
Номинальный ток I_{AN3}	0.5 A DC[1)]
Пульсации напряжения	не более 5% U_{AN3}.
Защитное отключение при повышении напряжения	U_{A3} > 18.5В
Уровень срабатывания сигнализации понижения напряжения (светодиод на передней панели)	U_{A3} < 14В +/-3%
Защита от перегрузки по току (ограничение тока) I_{A3}	0.5...1.5 A

Выходные характеристики 4

Номинальное напряжение U_{AN4}	24В пост. тока +6./-5В
Номинальный ток I_{AN4}	0.4 A [1)]
Порог ограничения тока	0.44A
Порог срабатывания индикации понижения напряжения (светодиод на передней панели)	16В +/-20%
Емкость нагрузки	max. 100 нФ

[1]) Сумма выходным токов (I_{AZ}, I_{A3}, I_{A4}) < 0.8 A DC:

Вентиляторы	Два осевых вентилятора
Питающее напряжение	115В перем. тока
Производительность каждого	160 $м^3$ час (свободный поток)
Контроль за вентиляторами	Воздушный поток контролируется термисторами и датчиками; останов одного или обоих вентиляторов сигнализируется светодиодом, и контакты соответствующего реле отключают выходные напряжения (этот режим может быть исключен перемычкой F-R) .
Наработка на отказ	тип. 30000...40000 час при 55 °С тип. 40000... 50000 час при 30 "С
Дополнительный контроль Контроль напряжения 24В (напряжения внешнего монитора)	от 15 до 30 В
Электрическая изоляция первичной/вторичной цепи	есть
Потенциальная развязка	нет
Буферная батарея	
Емкость Напряжение Долговечность	5 Ач 3.4 В 10 лет без разрядки

Источник питания 6ES5 955—3LF12

<u>Входные характеристики</u>	
Напряжение питания U_{EN}	230/115 В AC +10/-18.7% [1])
Уровень срабатывания сигнализации пониженного напряжения Ue	< 187 В AC (93 В AC)
Частота питающего напряжения fe	48. . .63 Гц
Входной ток I_{EN} с номинальной нагрузкой при питающем напряжении U_{EN} = 230 В (115 В)	2.95 (5.9) А
Бросок тока при включении (предельное значение)	200 А (100 А)
КПД с номинальной нагрузкой	тип >70% с вентилятором
Максимальное время отключения питающего напряжения	> 5 мс
COS Ф питающего напряжения	0.73
Входной предохранитель	быстродействующий 6А^ 250В 6.3мм х 32мм; место установки F107
<u>Выходные характеристики</u>	
Номинальное напряжение под нагрузкой U_{AN1}	5.1 В DC +/-0.5%
Диапазон установки выходного напряжения	(0.95. . . 1.05) х U_{AN1}
Ток под нагрузкой I_{AN1}	40 А DC
Пульсации напряжения	не более 1% от U_{A1}
Отклонение напряжения при изменении нагрузки на 50% при максимальном токе	не более 5% от U_{A1}
Время восстановления	не более 5 мс
Защитное отключение при повышении напряжения U_{A1} более	6В +/-5%
Уровень срабатывания сигнализации понижения напряжения U_{A1} менее (светодиод на передней панели)	4.75В +5%
Ограничение тока при перегрузке по току	(1.05...1.15) х I_{AN1}

[1])Выбирается переключателем сетевого напряжения

Выходные характеристики 2

Номинальное напряжение U_{ANZ}	24В DC +25/-17%
Номинальный ток I_{ANZ}	2.8 А DC [1)]
Пульсации напряжения	< 0.5% от U_{ANZ}
Защита от перегрузки по току	4А быстродейств. Предохр.: 250В; 6.3мм x 32мм; место установки F255

Выгодные характеристики 3 с дополнительным модулем

Номинальное напряжение U_{AN3}	15В DC +/-5%
Номинальный ток I_{AN3}	2А DC [1)]
Пульсации напряжения	< 5% от U_{AN3}
Защитное отключение при повышении напряжения	U_{A3} > 18.5В
Уровень срабатывания сигнализации понижения напряжения (светодиод на передней панели)	U_{A3} < 14В +/-3%
Защита от перегрузки по току I_{A3} (ограничение тока)	2...3 А

Выходные характеристики 4: фронт 24 В.

Номинальное напряжение U_{AN4}	24В DC +6В/-5В
Номинальный ток I_{AN4} Порог ограничения тока	0.4 А [1)] 0.44 А
Уровень срабатывания сигнализации понижения напряжения (светодиод на передней панели)	U_{A4} < 16В +/-20%
Емкость нагрузки	макс 100 нф

Вентиляторы Питающее напряжение	Два осевых 230/115В пер. тока переключ.
Производительность каждого	160 $м^3$/час (свободный поток)
Контроль за вентиляторами	Воздушный поток контролируется термисторами и датчиками останов одного или обоих вентиляторов сигнализируется светодиодом, и контакты соответствующего реле отключают выходные напряжения (этот режим может быть исключен перемычкой F-R)

[1)] Сумма выходных токов (I_{A2}+I_{A3}+I_{A4}) < 0.8 А DC

Наработка на отказ	тип. 30000... 40000 час при 55 °C тип. 40000...50000 час при 30 °C
Дополнительный контроль Контроль напряжения 24В (напряжения внешнего монитора)	 > 14...20 В
Электрическая изоляция первичной/вторичной цепи	есть
Потенциальная развязка	-
Буферная батарея Емкость Напряжение Долговечность	 5 Ач 3.4В ок.10 лет без разрядки

Источник питания 6ES5 955-3NA12

Входные характеристики

Напряжение питания U_{EN}	24В DC +25/-17%
Уровень срабатывания сигнализации по-ниженного напряжения Ue	< 20В пост. тока
Входной ток I_{EN} с номинальной нагрузкой при напряжении питания 24В	-4.5А
Бросок тока при включении	100А
КПД с номинальной нагрузкой	тип. 60%
Максимальное время отключения питающего напряжения	5 мс
Предохранитель	быстродействующий 6. А; 250В; 6.3мм х 32мм; место установки F2

Выходные характеристики 1

Номинальное напряжение под нагрузкой U_{AN1}	5.1В DC +/-0.5%
Диапазон установки выходного напряжения	(0.95...1.05) х U_{AN1}
Ток под нагрузкой	10 А DC
Пульсации напряжения	< 1% от $U_{AN!}$

Отклонение напряжения при изменении нагрузки на 50% при 100% I_n	< 5% от U_{AN1}
Время восстановления	< 5 мс
Защитное отключение при повышении напряжения U_{AI} более	6В +/-5%
Уровень срабатывания сигнализации понижения напряжения U_{A1} менее (светодиод на передней панели)	4.75В +/-5%
Ограничение тока при перегрузке	(1.05. . . 1.15) x I_{AN1}

Выходные характеристики 2

Номинальное напряжение U_{AN3}	! 24В DC +25/—17% O.S A DC [1 э]
Номинальный ток I_{AN3}	< 0.8 A 24 В и 15 В
Суммарный ток нагрузки выходов	< 0.5% от DANS
Пульсации напряжения	Выстрояейств. предохр. 1.5А;
Защита от перегрузки по току	250В; 6.3мм x 32мм; место установки F490

Выгодные характеристики 3 с дополнительным модулем

Номинальное напряжение U_{AN3}	15В DC +/-5%
Номинальный ток I_{AN3}	0.5 A DC [1)]
Пульсации напряжения	< 5% от U_{AN3}
Защитное отключение при повышении напряжения	U_{A3} > 18.5 В
Уровень срабатывания сигнализации понижения напряжения (светодиод на передней панели)	U_{A3} < 14В +/-3%
Защита от перегрузки по току I_{A3} (ограничение тока)	0.5... 1.5 A

Выгодные характеристики 4: фронт 24 В

Номинальное напряжение U_{AN4}	24В DC +6В/-5В
Номинальный ток I_{AN4}	0.4 A [1)]
Порог ограничения тока	> 0.44 A
Уровень срабатывания сигнализации понижения напряжения (светодиод на передней панели)	U_{A4} < 16В +/-20%
Емкость нагрузки	макс 100 нФ

[1)] Сумма выходных токов (I_{A2} + I_{A3} + I_{A4}) < 0.8 A DC

Вентиляторы	два осевых
Питающее напряжение	24В пост. тока
Производительность каждого	160 $м^3$/час (свободный поток)
Контроль за вентиляторами	Воздушный поток контролируется термисторами и датчиками; останов одного или обоих вентиляторов сигнализируется светодиодом, и контакты соответствующего реле отключают выходные напряжения (этот режим может быть исключен перемычкой F-R).
Наработка на отказ	тип. 30000...40000 час при 55 °C тип. 40000...50000 час при 30 °C
Дополнительный контроль Контроль внешнего напряжения 24В	-
Электрическая изоляция первичной/вторичной цепи	-
Потенциальная развязка	есть
Буферная батарея емкость напряжение долговечность	 5 Ач 3.4В 10 лет без разрядки

Источник питания 6ES5 955-3NC13
Входные характеристики

Напряжение питания U_{EN}	24В +25%/-I7% DC
Уровень срабатывания сигнализации понижения напряжения Ue	< 20В DC
Входной ток I_{EN} при номинальной нагрузке и U_{EN} = 24 В	6.9 А
Бросок тока при включении	250 А
КПД с номинальной нагрузкой	тип. 67%
Максимальное время отключения питающего напряжения	➢ 5 мс
Входной предохранитель	15 А; 250 В 6.3мм x 32мм; место установки F1

Выходные характеристики 1

Номинальное напряжение под нагрузкой U_{AN1}	5.1 В DC +/-0.5%
Диапазон установки выходного U_{AN1} напряжения	(0.95. . . 1.05) x U_{AN}
Ток под нагрузкой I_{AN1}	18 A DC
Пульсации напряжения	< 1% от U_{AN1}
Отклонение напряжения при изменении нагрузки от 50% до 100% I_N	< 5%
Установочное время	< 5 мс
Защитное отключение при повышении напряжения U_{A1} более	6В +/-5%
Уровень срабатывания сигнализации понижения напряжения U_{A1} менее	4.75 В +/-5%
Ограничение тока при перегрузке	(1.05...1.15) x U_{AN1}

Выходные характеристики 2

Номинальное напряжение U_{AN2}	24В DC +25/-17%
Номинальный ток I_{AN2}	0.8 A DC [1)]
Суммарный ток нагрузки выходов 24 В и 15 В	< 0.8 A
Пульсации напряжения	< 0.5% U_{A2}
Защита от перегрузки по току	Быстродейств. предохр. 1.5A; место установки F90

Выходные характеристики 3 с дополнительным модулем

Номинальное напряжение U_{AN3}	15В DC +/-5%
Номинальный ток I_{AN3}	0.5 A DC [1)]
Пульсации напряжения	< 5% -от U_{AN3}
Защитное отключение при повышении напряжения	U_{A3} > 18.5 В
Уровень срабатывания сигнализации пониженного напряжения (светодиод на передней панели)	U_{A3} < 14В +/-3%
Защита от перегрузки по току I_{A3} (ограничение тока)	0.5...1.5 A

[1)] Сумма выходных токов (I_{A2} + I_{A3}+ U_{A4}) < 0.8 A DC

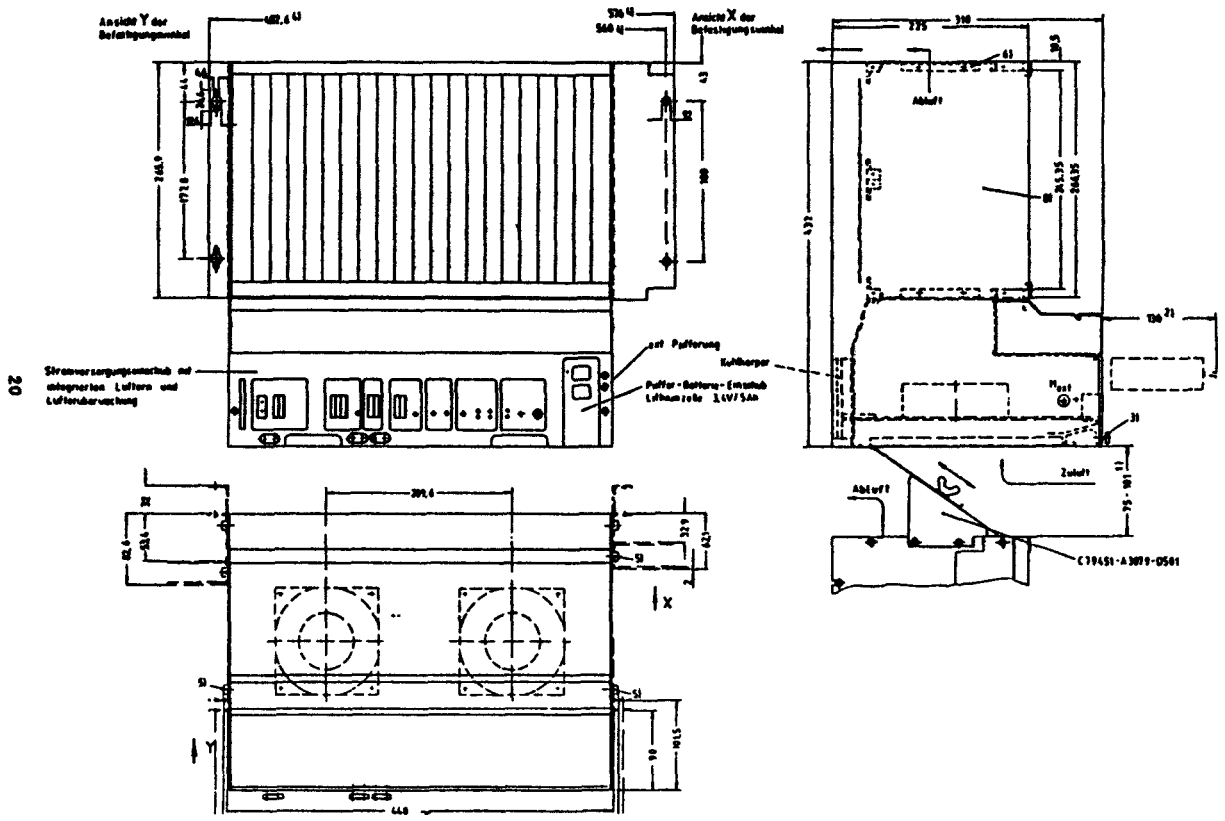
Выходные характеристики 4: фронт 24 В

Номинальное напряжение U_{AN4}	24В DC +6В/-5В
Номинальный ток I_{AN4}.	0.4 А[1)]
Порог ограничения тока	> 0.44 А
Уровень срабатывания сигнализации понижения напряжения (светодиод на передней панели)	16В +/-20%
Емкость нагрузки	макс 100 нф

Вентиляторы	Два осевых
Питающее напряжение	230/115В пер. тока переключ.
Производительность каждого	160 $м^3$/час (свободный поток)
Контроль за вентиляторами	Воздушный поток контролируется термисторами и датчиками; останов одного или обоих вентиляторов сигнализируется светодиодом и контакты соответствующего реле отключают выходные напряжения (этот режим может быть исключен перемычкой F-R).
Наработка на отказ	тип. 30000...40000 час при 55 °C тип. 40000... 50000 час при 30 °C
Дополнительный контроль Контроль напряжения 24В (напряжения внешнего монитора)	> 14...20 В
Электрическая изоляция первичной/вторичной цепи	есть
Потенциальная развязка	-
Буферная батарея Емкость Напряжение Долговечность	 5 Ач 3.4В ок.10 лет без разрядки

[1>] Сумма выходных токов (I_{A2}+ I_{A3}+ I_{A4}) < 0.8 А DC

Установочные размеры



а - вид по оси Y крепящего уголка
б - вид по оси X крепящего уголка
в - приточный воздух
г - отводимый воздух
д - выдвижной блок буферной батареи
литиевая батарея 3,4 В / 5 А ч е - радиатор
ж - внешняя буферизация
з - выдвижной блок питания со встроенными вентиля торами
и системой контроля вентиляторов
и - отводимый воздух

1) Свободное пространство для приточного воздуха при настенной установке или при монтаже в шкафах
2) Свободное пространство для замены буферной батареи
3) Шина снятия натяжения экранированных проводов
4) Ширина корпуса при произвольном положении крепящего уголка
5) Крепящий уголок привинчен к соединительным шинам
6) Крышка
7) Общая ширина корпуса с крепящими винтами (с учетом высоты головки винта)
8) Базовая несущая конструкция в соответствии с ES 902 для двойного европейского формата 233,4 x 160
Установочная ширина 28 SEP (SEP - стандартное установочное место)

2. Монтаж

2.1 Монтаж центрального устройства

S5-135U - центральный контроллер предназначен для установки в шкафы управления, открытые рамы или консоли.

Для закрепления прибора требуются винты и шайбы М6.

Сверку с боков и сзади должно быть оставлено пространство не менее 88.9 мм для циркуляции охлаждающего воздуха. Если необходимо разместить контроллеры один над другим, то рекомендуется использовать специальный воздухопровод (6ES5 981-ODA11).

Отвод тепла в самом шкафу должен обеспечиваться естественной конвекцией или вентиляцией.

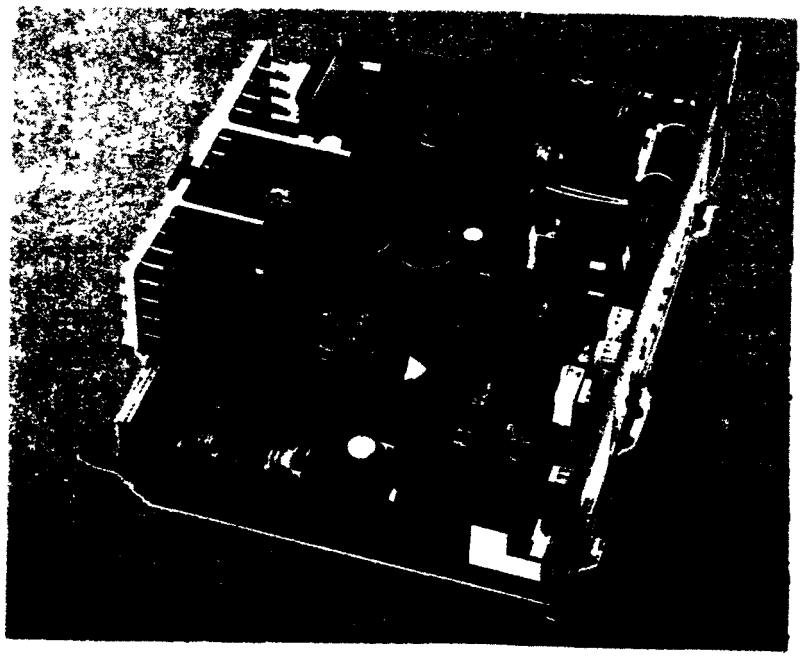
2.2 Монтаж блока питания

Блок питания можно устанавливать только при отключенном напряжении питания. Если Вы хотите снять блок питания с контроллера, — открутите два винта слева и справа передней панели, и, после этого, выдвигая на себя, можете вынуть блок питания.

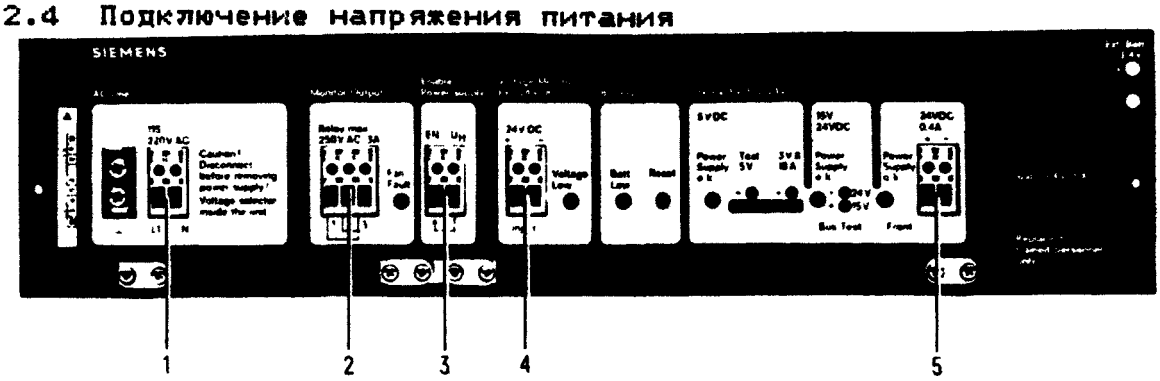
2.3 Монтаж дополнительного 15-вольтового модуля питания

Дополнительный модуль питания можно устанавливать только при отключенном напряжении.

Порядок снятия модуля описан в разделе 2.2.



2.4 Подключение напряжения питания



1. Клеммник напряжения питания:
   В зависимости от типа источника питания подключается либо 220В переменного тока, либо 24В постоянного.

2. Выход контрольного устройства:
   Если один или оба вентилятора не работают, то это сигнализируется светодиодом на передней панели, и выходные напряжения отключаются (эта функция может быть устранена посредством перемычки F-R; в этом случае задействованными остаются лишь контакты сигнального реле и светодиод) .
   1 2 3 (о подключении см. Указания по монтажу)

3. Напряжение блокировки:
   Отсутствие напряжения на EN-входе приводит к отключению источника питания. С помощью выхода U_NI можно управлять максимум семью входами EN (расположены на передней панели). (о подключении см. Указания по монтажу).

4. Контроль напряжения:
   Вход для контроля внешнего напряжения 24В; должен быть нагружен или отключен перемычкой BA-EX в источнике питания (за исключением модификации источника питания 6ES5 955-3NA11).

5. Выход 24В; 0.4А постоянного тока:
   Предназначен для запитывания деблокировочных входов U-периферии.

Рис. 7 Подключение напряжений питания

•I

При подключении необходимо руководствоваться стандартами VDE, особенно VDE 0100.

Сечение подключаемых к передним клеммам проводов не должно превышать 4 мм^2.

2.5 Монтаж соединительных кабелей

Подключение соединительных кабелей к модулям центральных процессоров, коммуникационных процессоров и интерфейсов расширения производится через передние разъемы. После подключения передние разъемы необходимо закрепить скользящим металлическим фиксатором. Обратите внимание на соответствие типа переднего разъема типу модуля, т.к. несоблюдение этого требования чревато непредсказуемыми последствиями.

2.6 Указания по монтажу

Кабели напряжения питания 220 В переменного тока внутри шкафов могут прокладываться в одних коробам с остальными соединительными кабелями только при условии экранирования сигнальных кабелей.

Заземляющий провод от внешнего источника питания к контроллеру должен быть как можно более коротким, с поперечным сечением не менее 10 мм*, хорошо закреплен и проложен отдельно от сигнальных проводников.

Если программируемый контроллер устанавливается в шкафу, то его корпус должен иметь хороший контакт с несущей консолью шкафа.

Внешний источник питания желательно устанавливать в верхней части шкафа.

Металлические части шкафа ('боковые стенки, дверь и т.д.) должны быть соединены между собой низкоомными проводами ("сечением 10—1ь мм'^{2}"). Сам шкаф необходимо соединить с защитным проводником.

Если для подключения входов/выходов используются экранированные кабели, то экран должен быть подключен с обеих сторон к экранной шине, которая с низким импедансом соединена с корпусом контроллера.

3 Эксплуатация

3.1 Общие замечания

-Недопустимо напряжение более 50В между выходными проводниками и потенциалом защитного проводника блока питания.

Защитный проводник должен быть всегда подключен.

-Если произошла перегрузка по напряжению, то источник питания блокирует выходы (выходные напряжения 5 и:24 В ограничиваются на уровне на уровне менее 0.5 В).

Блок питания можно опять запустить в работу, если выключить и включить напряжение питания, при условии, что перегрузка не была обусловлена внутренней неисправностью источника питания.

-Источник питания функционирует правильно только при токе нагрузки не менее 1А (для модификации 3LF12 - не менее 2А).

- Выходное напряжение должно достигать номинального уровня максимум за 200 мс.

- В основании корпуса источника питания можно установить воздушный фильтр.

3.2 Органы управления и элементы индикации

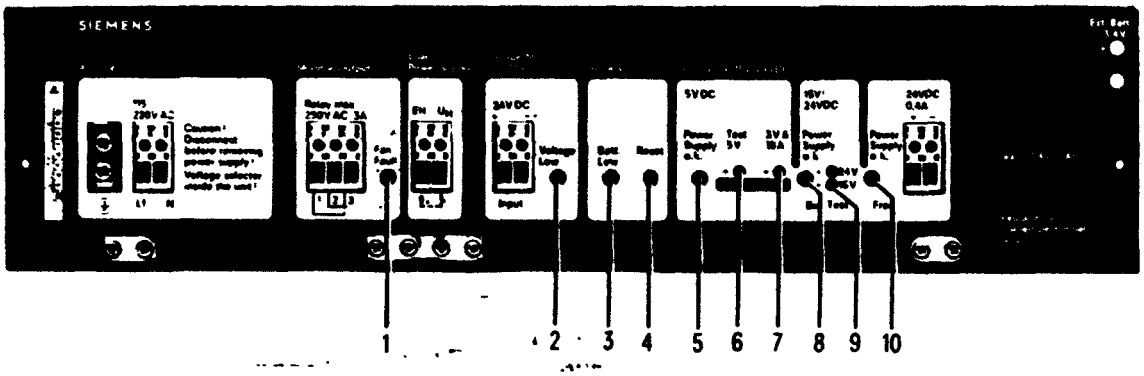


Рис. 8 Источник питания 6ES5 955-3...

1 Красный светодиод "FAN FAULT"
Этот светодиод загорается, если вентилятор неисправен. В этом случае при наличии перемычки F-R через 6-10 сек. блок питания выключается. Если по техническим причинам контроллер нельзя тотчас же отключить, то перемычку F-R необходимо снять, но помнить, что во избежание перегрева блок питания должен быть выключен не позднее, чем через 60 сек. после сообщения.

2 Красный светодиод "VOLTAGE LOW"
Этот светодиод загорается, если выходное напряжение понизилось до недопустимого уровня (это не относится к модификации источника питания -3NA12).

3 Желтый светодиод "BATT. LOW"
Желтый светодиод загорается, если напряжение буферной батареи упало до 2.7В; при выключении напряжения питания данные, буферизированные в RAM, теряются, после замены батареи светодиод "BATT. LOW" сразу гаснет.

4 Кнопка "RESET"
Если напряжение питание на контроллер не подано, то надо подать его и, если светодиод "BATT. LOW" горит, необходимо заменить батарею. После успешной замены батареи необходимо нажать кнопку квитирования, иначе контроллер при включении питания войдет в состояние "СТОП".

5 Зеленый светодиод "POWER SUPPLY O.K."
Этот светодиод горит, если источник питания обеспечивает требуемое выходное напряжение 5В.

6 Контрольные клеммы "TEST 5 V"
Предназначены для проверки выходного напряжения (стандартная величина 5.1В +/- 0.5Х).

7 Контрольные клеммы "3 V = 18 A":
Контроль выходного тока (3В при максимальном токе нагрузки соответствующего источника питания).

8 Зеленый светодиод "POWER SUPPLY O.K.";
Этот светодиод горит, если источник питания обеспечивает требуемые выходные напряжения 24В и 15В (при условии, что задействован дополнительный 15—вольтовый модуль).

9 Контрольные клеммы "15 V/24 DC";
а) для контроля выходного напряжения 24В +25/–17%;
б) для контроля выгодного напряжения 15В +/–5% (при условии, что имеется дополнительный модуль).

10 Зеленый светодиод "POWER SUPPLY O.K.":
Этот светодиод горит, если источник питания обеспечивает требуемое выходное напряжения 24В.

3.3 Назначение и расположение перемычек в блоке питания

Функция	Перемычка
Контроль батареи (/BAU) = вкл. Контроль батареи (/BAU) = выкл.	MN-MM замкн.* NN-MM разомкн.
Выключение блока питания при неисправности вентилятора При неисправности вентилятора блок питания не отключается (остается только светодиод и реле)	F-R замки. * F-R разомкн.
Работа с контролем напряжения нагрузки Работа без контроля напряжения нагрузки	BA-E X разомк. * BA-EX замкн.
Управление сигнальным реле (замкнуты контакты 2–3) - по ошибке вентилятора - при сообщении об ошибке батареи[2)] - без сообщения об ошибке батареи - при сообщении о понижении напряжения [1)] (BASPA=low) - без сообщения о понижении напряжения	не зависит от других перемычек RR-LL замкн. RR-LL разомк.* BB-AA замкн. BB-AA разомк.[1)]

* состояние поставки

[1)] При понижении напряжения на контрольном входе (U_u. < 20В < 25%) (отключается перемычкой BA-EX) или при пониженном напряжении на выходе (U_A < 4.75В) выдается сигнал BASPA низкого уровня.

[2)] Понижение напряжения батареи ($U_{бат}$ < 2.7В) ведет к сообщению об ошибке батареи (отключается перемычкой MM-NN). Кроме индикации "Batt low" и выдачи сигнала /BAU можно дополнительно опрашивать реле сообщений (при перемычке RR-LL) для приведенных ниже блоков питания.

Блок питания:

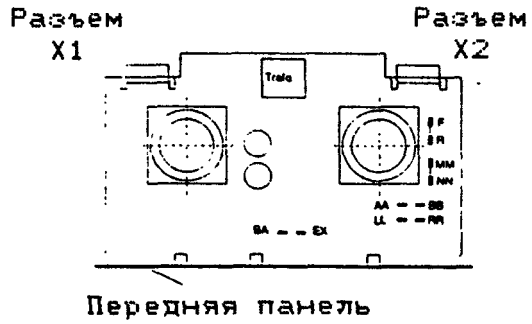
6ES5 955-3NA11 начиная с версии 9
6ES5 955-3NA11 начиная с версии 5
6ES5 955-3NA11 начиная с версии 7

При других источниках питания перемычка RR-LL не действует.

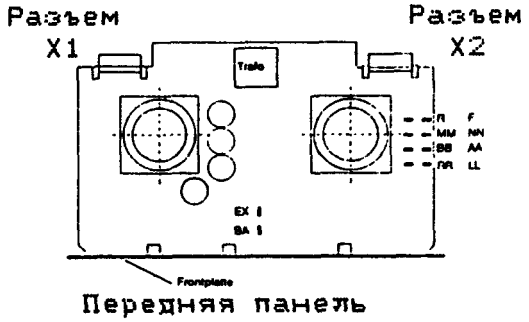
Расположение перемычек

На рисунке показано стандартное расположение перемычек (заводская установка).

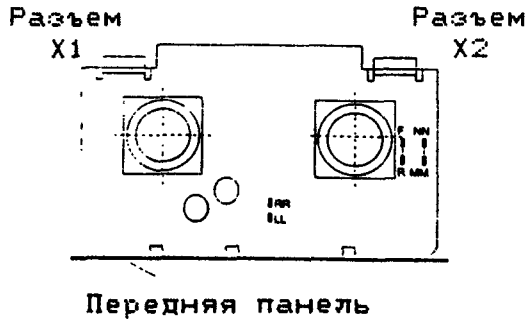
Источник питания 6.ES5 955-3LC14



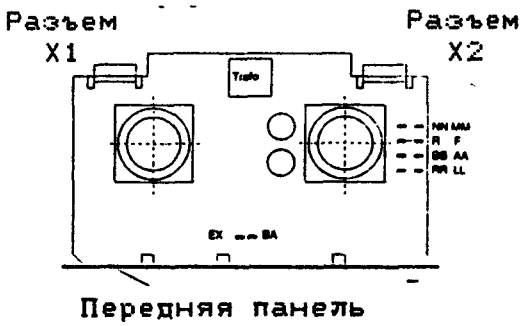
Источник питания 6ES5 955-3LF12



Источник питания 6ES5 955-3NA12



Источник питания 6.ES5 955-3NC13



3.4 Реакция блока питания при неисправностях

При выключенном блоке питания контакты реле 2-3 замкнуты, а контакты 1-2 разомкнуты.
При безошибочной работе контакты реле 1-2 замкнуты, а контакты 2-3 разомкнуты.

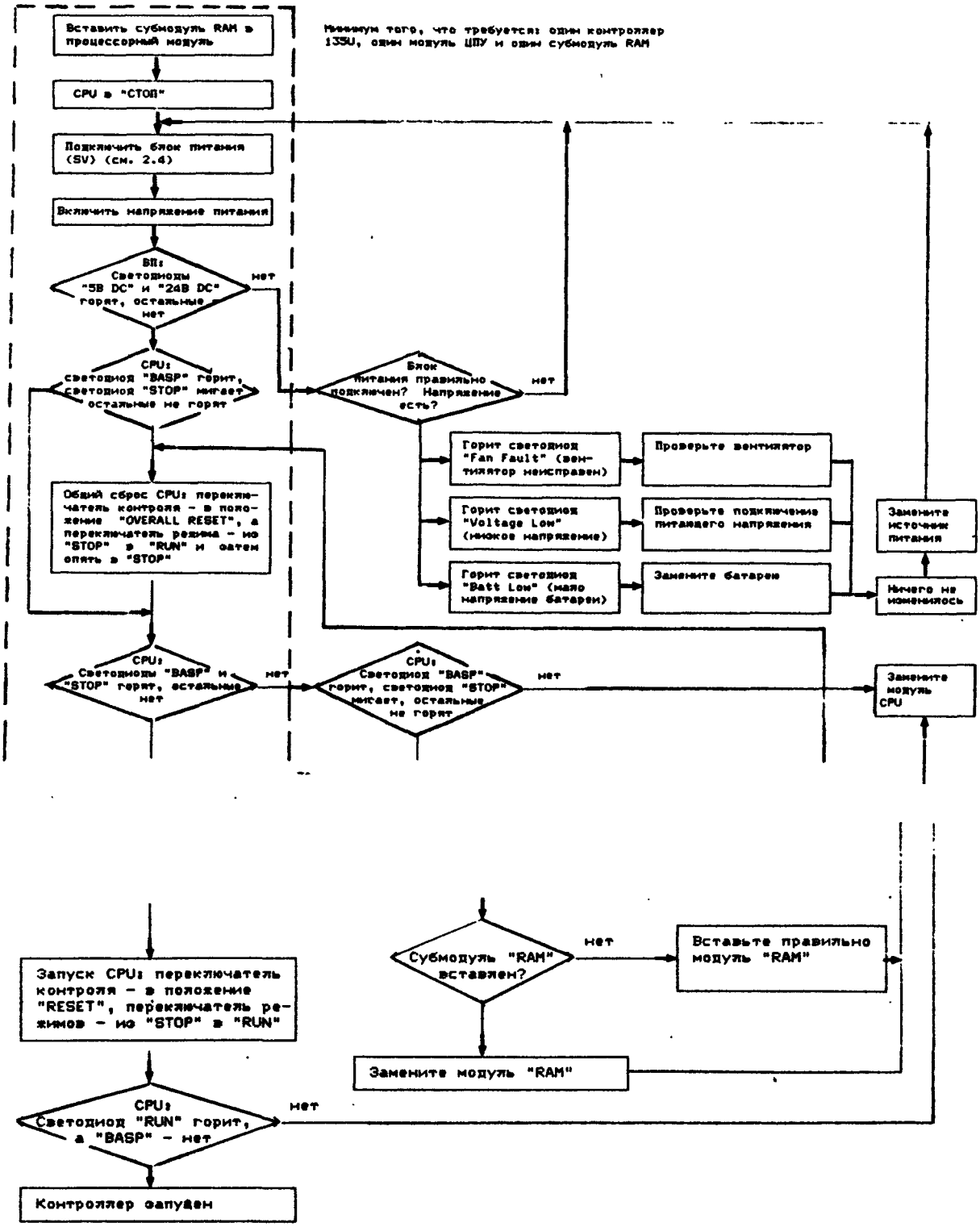
Кроме неисправностей вентилятора можно, при соответствующей установке перемычек, получать на реле сообщений информацию о других неисправностях в состоянии покоя (контакты реле 2-3 замкнуты) (смотри описание перемычек).

Следующая таблица описывает реакцию блоков питания при неисправностях (условия: перемычка MM-NN замкнута, перемычка BA-EX разомкнута).

				Сигнализация		Выход, напряжение выключено	
				светодиод	замкн. контакты реле	F-R разомкн.	F-R замкн. ”
деблокировка (EN-UH) присутствует	нет ошибки вентилятора	нет ошибки батареи и нет сообщения о пониженном напряжении		не горит	1-2	нет	нет
		ошибка батареи	RR-LL замк1)		2-3		
			RR-LL разомк.		1-2		
		Сообщения о пониж. напряжении	BB-AA замк.		2-3		
			BB-AA разомк.		1-2		
	ошибка вентилятора	горит			2-3	да	нет
деблокировка (UH-EN) отсутствует	нет ошибки вентилятора	нет ошибки батареи		не горит	1-2	да	да
		ошибка батареи	RR-LL замк. 2)		2-3		
			RR-LL разомк.		2-3		
		BB-AA замкн.			2-3		
		BB-AA разомкн.			1-2		

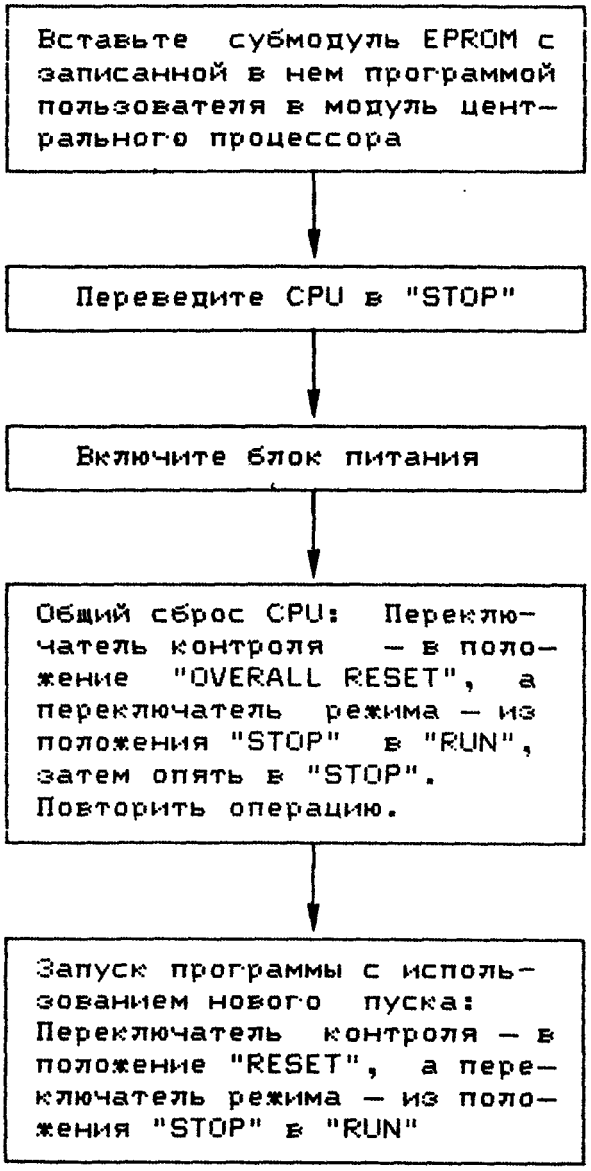
1) Смотри замечание [1)] Раздел 3.3
2) [2)] Зависит от версии. Смотри замечание [2)] Раздел 3.3

3.5 Запуск и функциональный тест

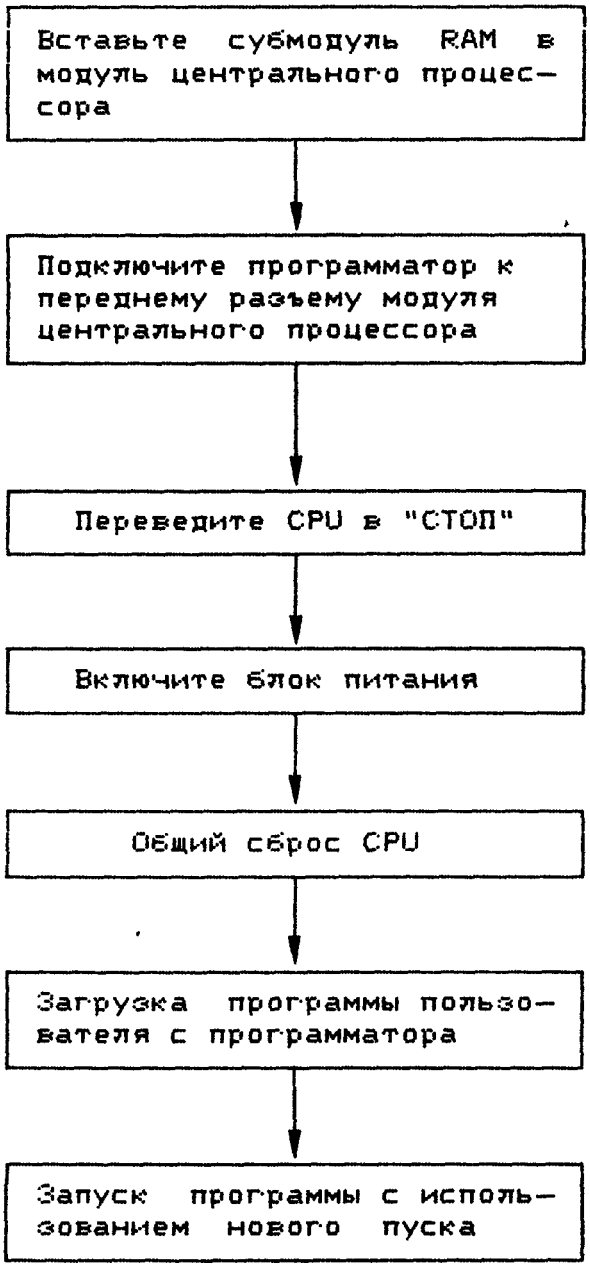


3.5.1 Запуск контролера в однопроцессорном режиме

Последовательность действий при использовании субмодуля RAM:

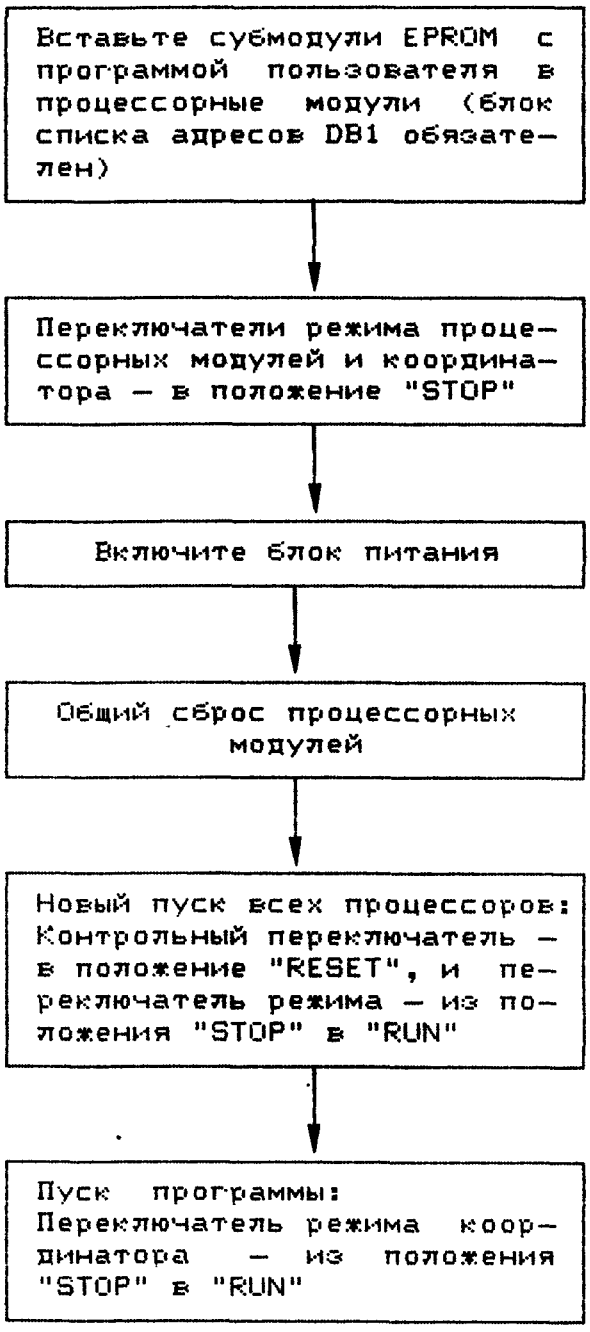


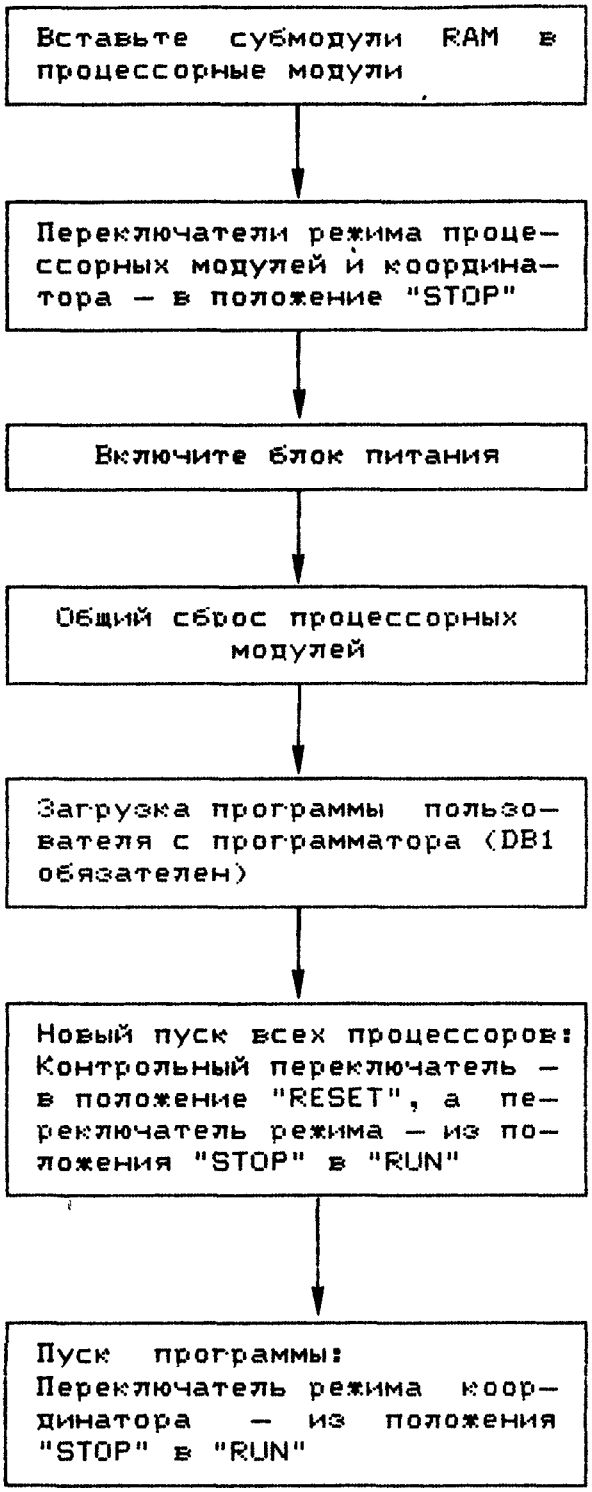
Последовательность действий при использовании субмодуля EPROM:



3.2 Запуск контроллера в многопроцессорном режиме

Последовательность действий при использовании субмодуля EPROM:	Последовательность действий при использовании субмодуля RAM:





4 Обслуживание

4.1 Общие указания

o Установка и снятие модулей

Запрещается вставлять и вынимать модули из контроллера, если он включен, т.к. это может привести к; непредсказуемым последствиям (исключение составляет U-периферия). Однако, если это по каким-либо причинам необходимо сделать, то центральный процессор при этом обязательно должен находиться в состоянии "стоп".

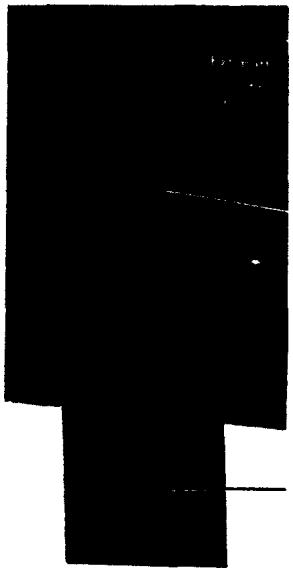
o Блок питания

Блок питания можно демонтировать только при отключенном напряжении.

При снятии блока питания надо сохранить связь буферной батареи с шиной контроллера, чтобы не потерять программу пользователя.

Профилактические работы не требуются.

4.2 Замена буферной батареи и вентилятора



Замена буферной батареи не приведет к потере информации в памяти, если контроллер в этот момент будет включен или если к клеммам "Ext. Batt." будет подведено внешнее напряжение (3.4В). В последнем случае необходимо проследить за правильностью соблюдения полярности. Буферную батарею следует менять раз в 3 года независимо от объема памяти и режима ее использования (емкость батареи 5Ачас). Чтобы заменить батарею, необходимо ;

-открыть крышку (сдвинуть ее вниз)
- вынуть батарею
- заменить батарею

Наработка вентилятора на отказ (см. Технические данные) зависит от длительности работы, температуры и условий окружающей среды. При остановке вентилятора во время работы включение контроля вентилятора (перемычка F-R замкнута) позволяет избежать нежелательного воздействия, например, на модули; блок питания отключается.

В отдельных случаях необходима профилактическая замена вентиляторов через заданный промежуток времени.

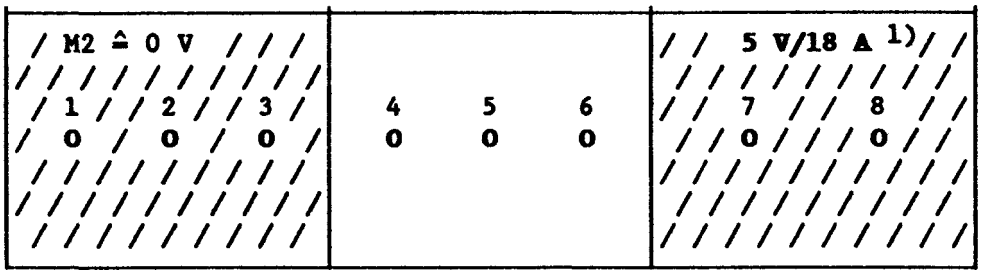
Для замены вентилятора блок питания в обесточенном состоянии снимается, крепежные винты вентилятора откручиваются, питание вентилятора (штекерные контакты) отключается. Установка вентилятора производится в обратной последовательности.

4.3 Назначение контактов заднего разъема блока питания

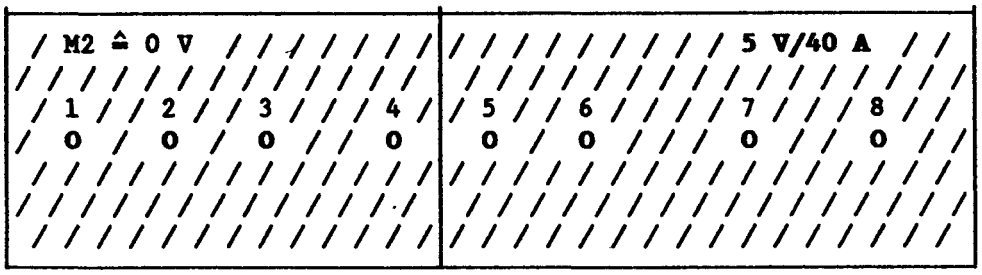
o Подключение нагрузки

Миниатюрный разъем XI, 8-полюсный, с 5/8 сильноточными контактами класса D по MIL-C24308, вид сзади

Для -3LC14, -3NA12, -31MC13:

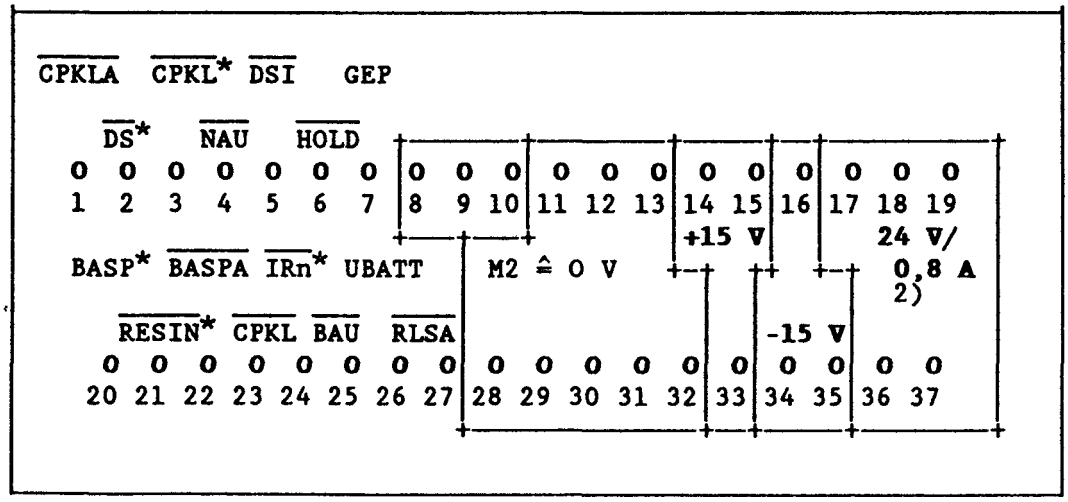


Для -3LF12:



o Подключение сигналов (включая 24В/0.8А или 24В/2.8А в случае -3LF12:)

Миниатюрный разъем X2, 37-полюсный, класса D по MIL-C24308, вид сзади:



[1)]5В/10А при блоке питания -3NA12
[2)]2.8А при блоке питания -3LF12
*Сигнал для МС-210

4.4 Назначение контактов задних разъемов модулей

Координатор (позиция 3)

	z	b	d
2	+ 5B	M 5B	
4	PL	PESPI	UBAT
6	CPKL	ADB 0	ADB 12
8	MEMR	ADB 1	ADB 13
10	MEMW	ADB 2	ADB 14
12	RDY	ADB 3	ADB 15
14	DB 0	ADB 4	PL
16	DB 1	ADB 5	PL
18	DB 2	ADB 6	PL
20	DB 3	ADB 7	PL
22	DB 4	ADB 8	PL
24	DB 5	ADB 9	PL
26	DB 6	ADB10	PL
28	DB 7	ADB11	DSI
30		BASP	PL
32	PL	M 5B	BASPA
2	+ 5B	M 5B	
4	DB 12	DB 8	
6	DB 13	DB 9	PL
8	DB 14	DB 10	PL
10	DB 15	DB 11	PL
12		PL	PL
14	NAU	PL	PL
16	BAU	PL	PL
18	CPKLA	PL	PL
20		PL	PL
22	PEU	PL	PL
24	GEP	PL	PL
26		PL	PL
28	PL	PL	PL
30	M24/15B	M24/15B	M24/15B
32	+ 24B	M 5B	+ 15B

Процессор (позиции 11,19,27,35)

	z	b	d
2	+ 5B	M 5B	
4	PL	PESPI	UBAT
6	CPKL	ADB 0	ADB 12
8	MEMR	ADB 1	ADB 13
10	MEMW	ADB 2	ADB 14
12	RDY	ADB 3	ADB 15
14	DB 0	ADB 4	IRx
16	DB 1	ADB 5	
18	DB 2	ADB 6	
20	DB 3	ADB 7	
22	DB 4	ADB 8	
24	DB 5	ADB 9	
26	DB 6	ADB 10	
28	DB 7	ADB 11	DSI
30	PL	BASP	PL
32	PL	M 5B	BASPA
2	+ 5B	M 5B	
4	DB 12	DB 8	
6	DB 13	DB 9	M 5B
8	DB 14	DB 10	
10	DB 15	DB 11	
12	M 5B		
14	NAU		
16	BAU		
18		M 5B	
20	PL	PL	PL
22	PL	PL	PL
24	GEP	M 5	
26		PL	PL
28	PL	PL	PL
30	M24/15B	M24/15B	M24/15B
32	+ 24B	M 5B	+ 15B

PL = резерв

Коммуникационный процессор
(CP 525) позиции 43,51,59,67

	z	b	d
2	+ 5B	M 5B	
4	PL	PESPI	UBAT
6	CPKL	ADB 0	ADB 12
8	MEMR	ADB 1	ADB 13
10	MEMW	ADB 2	ADB 14
12	RDY	ADB 3	ADB 15
14	DB 0	ADB 4	IRA
16	DB 1	ADB 5	IRB
18	DB 2	ADB 6	IRC
20	DB 3	ADB 7	IRD
22	DB 4	ADB 8	
24	DB 5	ADB 9	
26	DB 6	ADB10	
28	DB 7	ADB11	DSI
30		BASP	
32	PL	M 5B	BASPA

2	+ 5B	M 5B	
4	DB 12	DB 8	
6	DB 13	DB 9	
8	DB 14	DB 10	
10	DB 15	DB 11	
12			
14	NAU		
16	BAU		
18			
20		PL	PL
22		PL	PL
24	GEP		
26		PL	PL
28	PL	PL	PL
30	M24/15B	M24/15B	M24/15B
32	+ 24B	M 5B	+ 15B

Периферийные модули
позиции 75-131

	z	b	d
2	+ 5B	M 5B	
4	PL	PESPI	UBAT
6	CPKL	ADB 0	ADB 12
8	MEMR	ADB 1	ADB 13
10	MEMW	ADB 2	ADB 14
12	RDY	ADB 3	ADB 15
14	DB 0	ADB 4	IRA
16	DB 1	ADB 5	IRB
18	DB 2	ADB 6	IRC
20	DB 3	ADB 7	IRD
22	DB 4	ADB 8	
24	DB 5	ADB 9	
26	DB 6	ADB 10	
28	DB 7	ADB 11	DSI
30		BASP	PL
32	PL	M 5B	BASPA

PL = резерв

Подключение EG (позиции 139,147,155)

	z	b	d
2	+ 5B	M 5B	
4	PL	PESPI	
6	CPKL	ADB 0	ADB 12
8	MEMR	ADB 1	ADB 13
10	MEMW	ADB 2	ADB 14
12	RDY	ADB 3	ADB 15
14	DB 0	ADB 4	
16	DB 1	ADB 5	
18	DB 2	ADB 6	M 5B
20	DB 3	ADB 7	M 5B
22	DB 4	ADB 8	M 5B
24	DB 5	ADB 9	M 5B
26	DB 6	ADB10	M 5B
28	DB 7	ADB11	M 5B
30		BASP	M 5B
32		M 5B	BASPA
2	+ 5B	M 5B	
4	DB 12	DB 8	
6	DB 13	DB 9	
8	DB 14	DB 10	
10	DB 15	DB 11	
12			
14			
16			
18	CPKLA	PEU	
20			
22	M 5B	M 5B	
24	M 5B	M 5B	
26	M 5B	M 5B	
28	M 5B	M 5B	
30	M 5B	M 5B	
32	M 5B	M 5B	

Подключение EG (позиция 163)

	z	b	d
2	+ 5B	M 5B	
4	PL	PESPI	+ 5B
6	CPKL	ADB 0	ADB 12
8	MEMR	ADB 1	ADB 13
10	MEMW	ADB 2	ADB 14
12	RDY	ADB 3	ADB 15
14	DB 0	ADB 4	+ 5B
16	DB 1	ADB 5	+ 5B
18	DB 2	ADB 6	M 5B
20	DB 3	ADB 7	M 5B
22	DB 4	ADB 8	M 5B
24	DB 5	ADB 9	M 5B
26	DB 6	ADB 10	M 5B
28	DB 7	ADB 11	M 5B
30		BASP	M 5B
32		M 5B	BASPA
2	+ 5B	M 5B	
4	DB 12	DB 8	
6	DB 13	DB 9	
8	DB 14	DB 10	
10	DB 15	DB 11	
12	+ 5B	+ 5B	
14	+ 5B	+ 5B	
16	+ 5B	+ 5B	
18	CPKLA	PEU	
20			
22	M 5B	M 5B	
24	M 5B	M 5B	
26	M 5B	M 5B	
28	M 5B	M 5B	
30	M 5B	M 5B	
32	M 5BB	M 5B	

PL = резерв

4.5 Расположение сигналов прерываний на внутренней, шине

Модуль	CPU 1	CPU 2	CPU 3	CPU 4	PE/CP
Номер ячейки	11	19	27	35	ф3-131
Сигнал					
IRA	конт.14				конт.14
IRB		конт.14			конт.16
IRC			конт.14		конт.18
IRD				конт.14	конт.20

PE - модуль периферии
CP - коммуникационный процессор

5 Комплектующие

Наименование	Каталоговый номер	Ремонтопригодность [1]	Место заказа
Блок питания с вентилятором	6ES5 955-3LC13 6ES5 955-3LF11 6ES5 955-3NA11 6ES5 955-3NC12	R R R R	GWK GWK GWK GWK
Пакет вентиляторов (2 штуки) для 6ES5 955-3LC14 и 6ES5 955-3LF12	6ES5 988-3LB21	N	GWK
для 6ES5 955-3NA12 и 6ES5 955-3NC13	6ES5 988-3NB11	N	GWK
Дополнительный 15В модуль	6ES5 956-OAA11	R	GWK
Батарейный отсек	6XW79084-L1001-B5	N	GWK
Буферная батарея	W79084-L1001-B5	N	GWK
Крышка вентилятора	C79451-A3079-D501	N	GwK
Предохранители (6.3 x 32мм) 15 A A 4 A 1.5A	299461 300095 291963 287268	N	GWK
Держатель Фильтра	6ES5 981-OFA11	N	GWK
Фильтр (10 штук)	6ES5 981-OEA11	N	GWK

[1] R - ремонтопригодно

[2] N - неремонтопригодно

SIEMENS

SIMATIC S5

S-процессор 921
6ES5921-3UA11
6ES5921-3UA12

Руководство по эксплуатации

Заказной No. C79000-B8576--C262-05

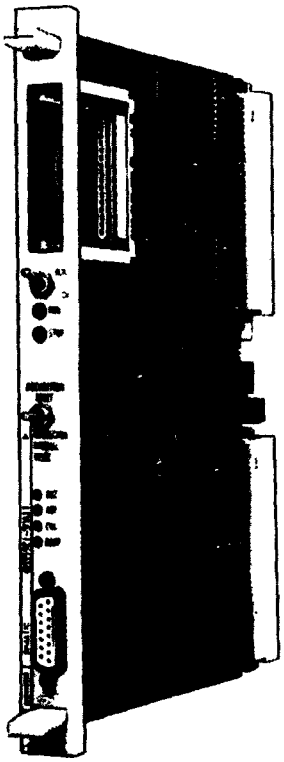


Рис.1 S-процессор 921

Оглавление Страница

1 Техническое описание

1.1 Область применения
1 .2 Структура
1.3 Принцип действия
1.3.1 Пояснения *к* блок-схеме
1.3.2 Субмодуль памяти пользователя
1.3.3 Обработка прерываний
1.4 Распределение памяти
1.5 Технические данные

2 Монтаж
2.1 Установка и снятие модулей
2.2 Установочные места центрального устройства

3 Эксплуатация
3.1 Режимы работы
3.2 Органы управления и элементы индикации
3.3 Режимы запуска
3.3.1 Ручной новый пуск без запоминания
3.3.2 Ручной новый пуск с .запоминанием
3.3.3 Автоматический новый пуск с запоминанием

4 Обслуживание
4.1 Назначение контактов задних разъемов
4.2 Назначение контактов переднего разъема

1 Техническое описание

1.1 Область применения

S-процессор разработан специально для контроллера S5-135U. Конструкция контроллера позволяет работать в многопроцессорном режиме, используя при этом до 4 CPU (центральных процессорных устройств).

S-процессор наилучшим образом подходит для быстрой обработки задач управления (обработки двоичных сигналов). Возможна также обработка слов (вычисления, управление с замкнутым циклом).

Через модуль координатора KOR 923C S-процессор -3UA12 может общаться с программатором (PG) по линии связи (в режиме "on line"). При этом, нет надобности переключать кабель программатора на каждый CPU. На прежних модификациях контроллеров S5—135U:-3KA12 и -3KB12 эта функция отсутствовала (см. управление координатором KOR 923 C.).

Язык программирования - STEP 5.

1.2 Структура

Модули представляют собой печатные платы двойного европейского формата, которые вставляются в раму контроллера. К внутренней S5-шине контроллера модули подключаются посредством двух 48-полюсных разъемов типа "2 ряда".

Передняя панель занимает по ширине 4/3 стандартной ячейки. На передней панели имеется окно с разъемом для установки в него субмодуля памяти пользователя.

Управление модулем состоит из двухпозиционного переключателя режимов работы (селектора режимов) и трехпозиционного переключателя сброса. Режим работы индицируется красным и зеленым светодиодами. Четыре маленьких красных светодиода индицируют ошибки и нерабочее состояние контроллера.

S-процессор можно соединить с программатором PG 675 или прибором диагностики DG 335 U через передний 15-полюсный разъем.

1.3 Принцип действия

1.3.1 Пояснения к блок-схеме

Словный процессор включает микропроцессор 8031 и необходимую память. Он выполняет функции операционной системы и обрабатывает словные команды программы пользователя.

Битовый процессор включает в себя микропрограммное управление и логическую часть. Он вызывает команды программы пользователя, обрабатывает битовые команды или пересылает словные команды в словный процессор. Через битовый процессор осуществляется связь с периферией.

В субмодуле программы пользователя содержатся команды пользователя в MC5-коде.

В регистре команд команда пользователя разделяется на код операции и параметр.

Память системных данных содержит данные, необходимые для работы операционной системы, а также область отображения процесса для цифровых входов/выходов, маркеров, таймеров и счетчиков.

Логическое устройство преобразовывает логическую команду в двоичную комбинацию, соответствующую исходной команде. Оно также выполняет команды сброса и установки. Максимальная глубина обработки скобковых вложений - 7 уровней.

Постоянная память отображения 1 преобразует текущий код операции во входной адрес микропрограммного управления.

Микропрограммное управление генерирует управляющие биты, необходимые для обработки отдельных команд.

Постоянная память отображения 2 содержит входные адреса для обработки команд микропроцессорам 8031.

Память системных программ содержит всю операционную систему S-процессора, интерфейсное программное обеспечение и MC5-интерпретатор словных команд.

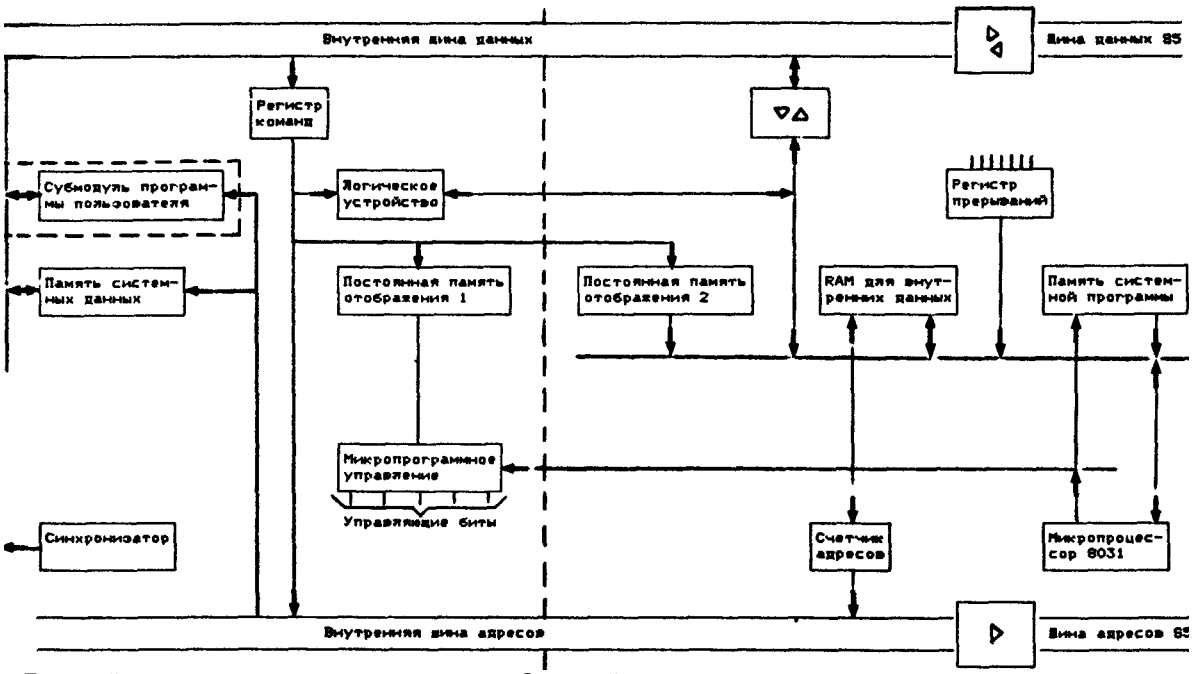
RAM для внутренних данных расширяет область памяти микропроцессора 8031.

Счетчик адресов синхронизируется микропрограммным управлением и указывает ячейку памяти, в которой находится команда.

С приходом команды перехода микропроцессор 8031 считывает и запоминает текущее состояние счетчика адресов, чтобы потом к нему вернуться, и устанавливает новое текущее значение счетчика.

Регистр прерываний хранит все необходимые данные о команде, во время выполнения которой пришел запрос на прерывание.

Синхронизатор вырабатывает опорную последовательность импульсов для синхронизации работы битового и байтового процессоров.



1.3.2 Субмодуль памяти пользователя

В S-процессоре для хранения программы пользователя можно использовать как субмодуль EPROM (6ES5 376-…), так и субмодули RAM (6ES5 377-...).

o Субмодуль EPROM

EPPOM-субмодули программируются непосредственно на программаторе PG.

Можно применять следующие типы субмодулей EPROM:

Объем памяти	Каталоговый номер
8 x 2^{10} слов	6ES5 376-0AA11
16 x 2^{10} слов	6ES5 376-0AA21
32 x 2^{10} слов	6ES5 376-0AA31 [1])

o Субмодуль RAM

Программа пользователя может быть записана в RAM-субмодуль при помощи программатора через стандартный интерфейс S-процессора в режиме "on line". Для того, чтобы не потерять данные, необходимо подпитывать RAM-субмодуль буферным напряжением (S-процессор и буферная батарея обязательно должны быть вставлены).

Можно применять следующие типы субмодулей RAM:

Объем памяти	Каталоговый номер
8 x 2^{10} слов	6ES5 377-0AA11
16 x 2^{10} слов	6ESS 377-0AB21 [2])
16 x 2^{10} слов	6ESS 377-0AA21
32 x 2^{10} слов	6ESS 377-0AA32
64 x 2^{10} слов	6ES5 377-0AA41

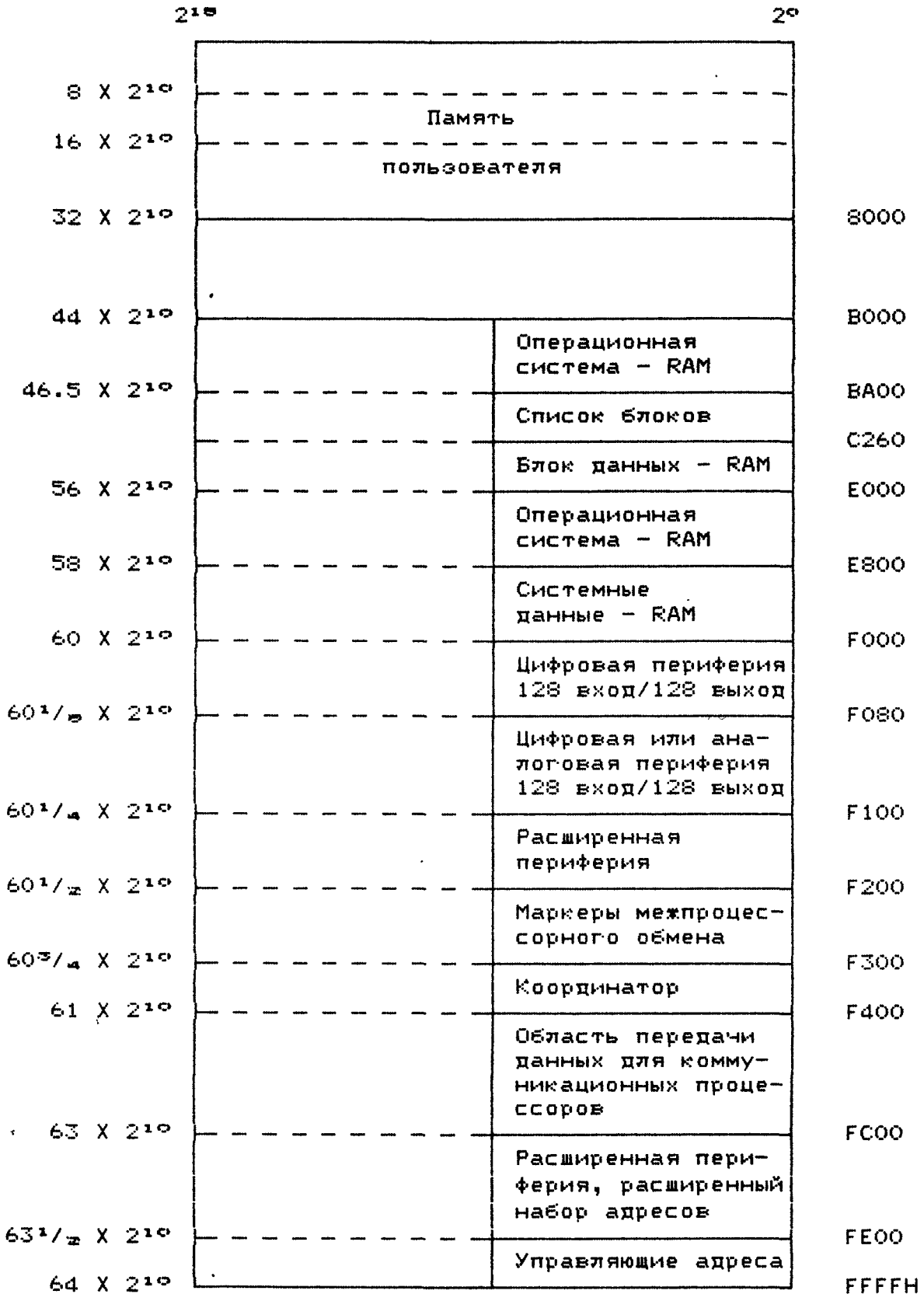
[1)]Можно программировать только с помощью MEP-адаптера 6ES5 985-2AA11

1.3.3 Обработка прерываний

В распоряжение каждого процессора AG 135U предоставляется сигнал прерывания. Он используется, когда требуется быстрая реакция на какое-то событие или цепь событий, более важных, чем остальные. В этом случае циклическая обработка программы прерывается, и начинается обработка программы прерывания, помещенной в организационный блок OB2. Для более подробного рассмотрения этого режима следует обратиться к руководству по программированию.

Режим обработки прерывания возможен при использовании цифровых модулей, подходящих для этих целей (например, 6ES5 432-...) или соответствующих модулей интеллигентной периферии.

1.4 Распределение памяти



1.5 Технические данные

Класс защиты.....IP 00
Рабочая температура……..от О до 55 "С
Температура транспортировки
и хранения……..от -40 до 70 °С
Допустимая влажность (без выпадения росы').....….95% при 25 °С
Максимальная высота над уровнем моря .....3500 м

Напряжение питания.....5 В +/-5%
Номинальный потребляемый ток…….3 А
Напряжение буферной
батареи..... 3.4 В
Ток буферной батареи (без
субмодуля RAM)…….....20 MkA
Размеры……..20.32 х 233.4 х 160 мм
Вес .....ок. 0.5 Кг

	P-область	Q-область	Всего
Цифровые входы с отображением процесса	макс.1024	макс.2048	макс.1024
Цифровые входы без отображения процесса……	макс.1024	макс. 128	макс 3072
или аналоговые входы……	макс.64		
Цифровые выходы с отображением процесса……	макс.1024	макс.2048	макс. 192
Цифровые выводы без отображения процесса……	макс.1024	макс. 128	макс.1024 макс.3072
или аналоговые выходы……	макс. 64		макс. 192
Маркеры……	2048		
Таймеры……	128		
Счетчики……	128		
Объем памяти пользователя……	макс. 32К слов (16-битовых) EPROM или RAM		
Скорость передачи данных серийного интерфейса……	9600 бит/с		
Программные блоки (PB)…….....……..	256 PB		
Шаговые блоки (SB)…….....	256 SB		
Функциональные блоки (FB)……..	256 FB		
Блоки данные (DB)…….....	256 DB		
Организационные блоки……	OB1...39		
Организационные блоки специальных функций…….	SF-OB 40...255		
Стандартные Функциональные блоки (SFB)…….....	для цифровых функций (например, деление 32-битовых чисел, извлечение корня из чисел с плавающей точкой, регистры сдвига, буферная память)		

2 Монтаж

2.1 Установка и снятие модулей

Модули вынимаются из рамы центрального устройства путем постепенного выдвижения по направляющим за ручки при помощи легкого покачивания вверх-вниз. Модули разрешается вставлять и вынимать только при отключенном контроллере.

2.2 Установочные места центрального устройства

Однопроцессорный режим: S-процессор устанавливается на 11-е установочное место центрального устройства 135U.

Многопроцессорный режим: S-, R-, и/или M- процессоры устанавливаются на 11, 19, 27 или 35 установочные места центрального устройства. Они размещаются без промежутков, начиная с 11 места (можно вставить от 2 до 4 модулей) .

3 Эксплуатация

3.1 Режимы работы

o Однопроцессорный режим

В центральное устройство устанавливается один S-процессор, при этом имеется постоянный доступ к S5-шине и непрерывный цикл.

o Многопроцессорный режим

Вставляются от 2 до 4 S-, R- и/или M- процессоров:

В этом случае доступ к S5-шине осуществляется через координатор KOR 923.

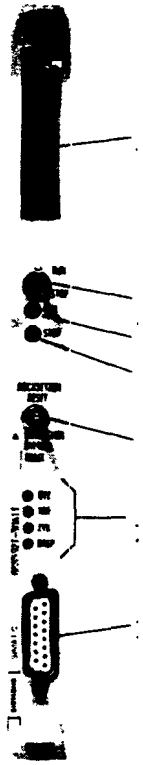
Вводные и выходные маркеры межпроцессорного обмена, а также входы и выходы модулей, должны быть определены для каждого процессора адресным списком в DB1. В отсутствие этого списка адресов контроллер сохраняет состояние "СТОП". DB1 со списком адресов заносится в субмодуль RAM процессора с помощью программатора в режиме "on line" или программируется непосредственно на программаторе в субмодуле EPROM (см. Раздел 1.4.3 Руководства по Программированию).

Работа всех процессоров синхронизируется координатором KOR 923.

Если один из процессоров переходит из циклической программы в "СТОП", то и все остальные процессоры переходят в "СТОП", при этом все цифровые выходы блокируются сигналом BASP. Для асинхронного или независимого пуска каждого процессора можно выбрать в KOR 923 тестовую функцию (см. Руководство по программированию, Раздел 5.2.3 и Руководство по эксплуатации KOR 923 A или C) .

Процессор, предназначенный для целей высокоскоростного управления (с временем реакции, например, 1мс), в то же самое время не может быть использован для обмена большими массивами данных с коммуникационными процессорами. Именно в таких случаях необходимо распределять задачи между отдельными процессорами.

3.2 Органы управления и элементы индикации



Разъем субмодуля памяти пользователя

Селектор режимов
Зеленый светодиод "RUN"
Красный светодиод "STOP"

Переключатель сброса

Светодиодный индикатор ошибок и сообщений

15—полюсный разъем интерфейса связи с PG

Рис. 3 Передняя панель S-процессора

Селектор режимов

o Положение "RUN" (= работа)
При включении в положение "RUN" S-процессор работает в циклическом режиме, и это индицируется зеленым светодиодом "RUN". Циклическая работа означает, что:

считывание области отображения входов, обработка программы пользователя в соответствии с последовательностью вызовов в OB1 (или FBO, если не загружен DB1), запись области отображения выходов, изменения в области маркеров связи (в соответствии с DB1) и переход на новый цикл обработки происходят автоматически (циклически).

Циклическая обработка программы прерывается лишь в тех случаях, когда происходит обработка таймера (запрограммированная задержка), из-за аппаратных прерываний и обработки таких прерываний или когда происходит обращение к контроллеру со стороны программатора через интерфейсный модуль.

Циклическая обработка программы прекращается, если обнаружена системная ошибка, ошибка программы или аппаратная неисправность (например, сбой напряжения питания NAU, задержка квитирования QVZ, ошибка замещения SUF).

o Положение "STOP"

При переключении из положения "RUN" в положение "STOP" (или под воздействием определенных причин в самом контроллере) процессор останавливается. При этом индицируется сигнал "BASP" на передней панели, и блокируются модули выходов. Функция контроля возможна с помощью программатора, *т.к.* в этом случае подавляется сигнал блокировки "BASP", и появляется доступ к выходным модулям.

Если селектор режимов при подаче напряжения питания стоит в положении "RUN" (в многопроцессорном режиме на всех S-, R- и/или M- процессорах, а также на координаторе KOR 923), то выполнится автоматический новый старт с запоминанием, при условии, что до выключения контроллер был в состоянии циклической обработки, и с тех пор не изменилось положение переключателей и конфигурация контроллера.

Светодиод "RUN"

Когда этот светодиод постоянно горит, то это означает, что идет циклическая обработка программы.

Светодиод "STOP"

o Светодиод горит постоянно

Если до этого программа циклически обрабатывалась, то это означает:

a) в однопроцессорном режиме селектор режимов переведен в "STOP";
б) в многопроцессорном режиме переключатель режимов любого процессора переведен в "STOP";
c) в многопроцессорном режиме с программатора активизирована Функция "AG STOP";
d) в одно- или многопроцессорном режимах имеет место аппаратная неисправность, причем не в процессорах (NAU, BAU, PEU);
e) завершено выполнение Функции программатора "контроль обработки" или выполнен общий сброс;

o Светодиод быстро мигает

a) подсказывает оператору необходимость общего сброса (см. "переключатель сброса");
б) во время инициализации была обнаружена ошибка, например: отсутствовало (даже кратковременно) буферное напряжение питания RAM-субмодуля или RAM процессора (во время замены батареи или при отключении питания), отсутствует программа пользователя в RAM- или EPROM- субмодуле или он не вставлен. Старт без устранения причин остановки невозможен. Необходим общий сброс, а затем ручной новый пуск без запоминания.

o Светодиод медленно мигает

a) процессор обнаружил ошибку процессора в одно- или многопроцессорном режиме, которая привела к остановке;
b) ошибка оператора (неправильный выбор стартовые параметры или ошибка в DB1), светодиод медленно мигает даже в том случае, если селектор режимов наводится в положении "STOP". Это означает, что контроллер не готов к циклической работе;
c) в 0B .запуска или циклической программе пользователя запрограммирована команда остановки;
d) в однопроцессорном режиме сработала команда программатора "AG STOP" (остановка контроллера).

Переключатель сброса

o Общий сброс ("OVEPALL PESET")

Вся память RAM стирается и инициализируется. Красный светодиод "STOP" должен гореть непрерывно.

a) придерживая переключатель сброса в положении "OVEPALL PESET" (общий сброс) селектор режимов переведите из положения "STOP" в "RUN", а затем опять в "STOP". Результат: Светодиод "STOP" быстро мигает, необходим общий сброс [1]).
б) придержите опять переключатель сброса в положении "OVERALL PESET" и селектор режимов переведите из положения "STOP" в "RUN", а затем опять в "STOP". Результат: Выполнен общий сброс, процессор остановлен; светодиод горит постоянно. Единственно возможным режимом является режим ручного нового пуска без сохранения.

o Сброс ("Reset")

Смотри Раздел 3.3

Светодиодный индикатор ошибок и сообщений

Ошибки и реакции на них OVZ, ADF и ZYK - отображаются тремя светодиодами.
Четвертый светодиод индицирует "BASP"- сигнал.

Светодиод "QVZ"

Горит, если программа обращается к периферийному модулю, который:

- (в однопроцессорном режиме) наводился во время нового пуска в области отображения процесса (входы от о до 127, выходы от О до 127) и который был введен в так называемую 9-ю дорожку, как реально существующий, но теперь не отвечает, или
- (в одно- или многопроцессорном режиме) был занесен в DB1 (список адресов) и был представлен, как реально существующий, но теперь не отвечает, или
- не отвечает или больше не отвечает при обращении к нему с помощью команд прямого доступа (L/T P..., L/T Q...).

Это означает следующее:

- Модуль неисправен.
- Модуль был вынут во время работы или во время останова или в отключенном состоянии без выполнения нового пуска.

Светодиод "ADF"

Во время нового пуска S-процессора создается 9 дорожка операционной системы. Все физически существующие периферийные модули отмечаются в области отображения процесса логической 1, все отсутствующие – логическим 0 на 9 дорожке. В многопроцессорном режиме, или когда программируется список адресов в DB1, 9 дорожка создается на базе DB1. Если программой пользователя опрашивается адрес периферии по которому нет модуля, то процессор прерывает циклическую обработку программы.

Светодиод "ZYK"

Зажигается, когда превышено максимально допустимое время цикла. Время цикла складывается из времен выполнения всех частей программы пользователя (циклической + управляемой по времени + управляемой по прерыванию от процесса). Сообщение об ошибке "ZYK" прерывает циклическую обработку программы.

Светодиод "BASP"

Зажигается, когда запрещена команда вывода. Цифровые выходы переводятся в безопасное состояние (высокого импеданса). Область отображения выходов не стирается. Сигнал BASP выводится, когда включается или выключается блок питания, когда напряжение блока питания ниже нормы или когда контроллер находится в состоянии "СТОП".

3.3 Режимы запуска

Все необходимые для циклической работы контроллера данные определены и установлены. Теперь контроллер может работать в циклическом режиме. См. также руководство по программированию.

Примечание [1]). Прерывание функции "общий сброс":
Переведите селектор режимов из положения "STOP" в "RUN" и затем опять в "STOP", не касаясь переключателя сброса.
Результат; общий сброс не произошел и процессор сохранил состояние "СТОП".

3.3.1 Ручной новый пуск без запоминания

Маркеры, таймеры, счетчики н область отображения процесса стерты. Обработка программы пользователя начинается с самого начала.

Контроллер должен быть остановлен. В многопроцессорном режиме селектор режимов координатора KOR 923 должен быть в положении "STOP".

Процессор сброшен перейдет к циклической обработке программы, если:

- переключатель сброса S-процессора удерживается в позиции "RESET",
- селектор режимов переводится из положения "STOP" в "RUN",
- в многопроцессорном режиме селектор режимов координатора KOR 923 переводится из положения "STOP" в "RUN",
- или задействована функция программатора "ПУСК AG'".

3.3.2 Ручной новый пуск с запоминанием

Маркеры сохраняются; таймеры, счетчики и область отображения процесса стираются. Обработка программы пользователя начинается с самого начала.

Перед остановкой контроллер должен работать в циклическом режиме. В многопроцессорном режиме переключатель координатора необходимо перевести в положение "STOP".

Процессор переходит в режим циклической обработки программы, если:

- переключатель сброса S-процессора в среднем положении,
- селектор режимов переведен из положения "STOP" в "RUN" или
- в многопроцессорном режиме селектор режимов координатора KOR 923 переведен из положения "STOP" в "RUN"
- или задействована функция программатора "START AG".

3.3.3 Автоматический новый пуск с запоминанием

Все то же самое, как при ручном новом пуске с запоминанием, однако, после включения или выключения питания никаких действий от оператора не требуется.

Автоматический новый пуск с запоминанием выполняется после включения питания, если;

- перед выключением напряжения питания контроллер работал в циклическом режиме,
- селектор режимов S-процессора (в многопроцессорном режиме и S-процессора, и координатора) установлен в положение "RUN", но его положение не менялось,
- субмодуль памяти пользователя не вынимался и
- буферная батарея функционирует нормально, что гарантирует сохранение данных в субмодуле RAM.

4.1 Назначение контактов задних разъемов

Задний разъем 1

	d	b	z
2		M 5B	+ 5B
4	UBAT	PESP	
6	ADB 12	ADB 0	CPKL
8	ADB 13	ADB 1	MEMR
10	ADB 14	ADB 2	MEMW
12	ADB 15	ADB 3	RDY
14	IR	ADB 4	DB 0
16		ADB 5	DB 1
18		ADB 6	DB 2
20		ADB 7	DB 3
22		ADB Э	DB 4
24		ADB 9	DB 5
26		ADB 10	DB 6
28	DSI	ADB 11	DB 7
30	BUSEN	BASP	QUIT T
32	BASPA	M SB	HALT

Задний разъем 2

	d	b	z
2	T1	M 5B	+ '5B
4	T2	DB 8	DB 12
6	M 5B	DB 9	DB 13
8	MA 0	DB 10	DB 14
10	MA 1	DB 11	DB 15
12	MA 2	MA 5	M 5B
14	MA 3	MA 6	NAU
16	MA 4	MA 7	BAU
18		M 5B	STAT
20		STELI	
22	TXD [1])		PEU
24	VKE	M 5B	GEP
26	ZYK	RXD *>	BE
28		PERO	
30	TE	M 24B	M 24B
32		M 5B	+ 24B

[1]) относится только к модификации -3UA12 S-процессора.

4.2 Назначение контактов переднего разъема

1	RXD	9	RXD
2		10	24В экран
3	+24В от шины	11	20 мА/источник
4		12	
5		13	20 мА/приемник
6	ТХД	14	
7	ТХД	15	

SIEMENS

SIMATIC S5
R-процессор 922
6ES5 922-3UA11

Руководства по эксплуатации Заказной No. C79000-B8576-C348-04

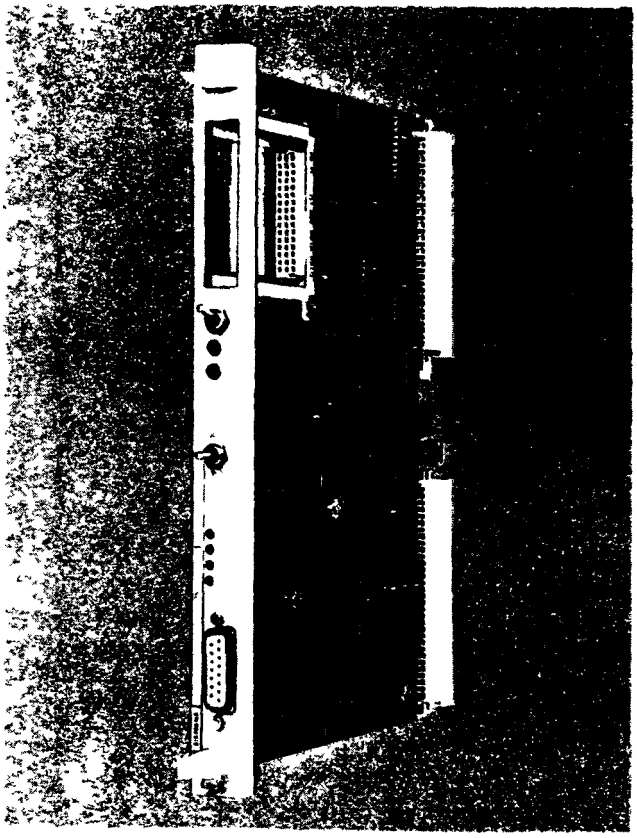


Рис.1 R-процессор 922

Содержание

1. Техническое описание
1.1 Область применение
1.2 Структура
1.3 Принцип действия
1.3.1 Пояснения к блок-схеме
1.3.2 Субмодуль памяти пользователя
1.3.3 Обработка прерываний
1.4 Распределение памяти
1.5 Технические данные

2 Монтаж
2.1 Установка н снятие модулей
2.2 Установочные места в центральном устройстве

3 Эксплуатация
3.1 Режимы работы
3.2 Органы управления и элементы индикации
3.3 Режимы запуска
3.3.1 Новый пуск
3.3.2 Ручной перезапуск
3.3.3 Автоматический перезапуск

4 Обслуживание
4.1 Назначение контактов заднего разъема
4.2 Назначение контактов переднего разъема

1 Техническое описание

1.1 Область применения

R-процессор разработан специально для программируемого контроллера (ПК) S5-135U. Конструкция контроллера позволяет работать в многопроцессорном режиме, используя при этом до 4-х CPU (центральных процессорных устройств).

R-процессор наилучшим образом подходит для быстрой обработки слов (задач арифметики, регулирования). Возможна также обработка двоичных сигналов.

Язык программирования - STEP 5.

1.2 Структура

Модули представляют собой печатные платы двойного европейского формата, которые вставляются в раму контроллера. К внутренней S5-шине контроллера модули подключаются посредством двух 48-полюсных разъемов типа "2 ряда".

Передняя панель занимает по ширине $1^{1/4}$ стандартной ячейки. На передней панели имеется окно с разъемом для установки в него субмодуля памяти пользователя.

Управление модулем состоит из двухпозиционного переключателя режимов работы (селектора режимов) и трехпозиционного переключателя сброса. Режим работы индицируется красным и зеленым светодиодами. Четыре маленьких красных светодиода индицируют ошибки и нерабочее состояние контроллера.

R-процессор можно соединить с программатором или прибором диагностики или панелью оператора через передний 15-полюсный разъем.

1.3 Принцип действия

1.3.1 Пояснения к блок-схеме

Блок-схему R-процессора можно разделить на 3 основных функциональных узла:

– Главный процессор
– Квази-двухпортовая RAM и объединитель шин
– Процессор интерфейса (AS-CPU)

o Главный процессор

Основа главного процессора – 16-,битовый процессор 80186 с прямым доступом к указанной ниже памяти:
Память системной программы содержит всю операционную систему и MC5-интерпретатор.
Память системных данных содержит данные, необходимые для работы операционной системы, а также область отображения процесса для

цифровых входов/выходов, маркеров, таймеров и счетчиков.

В субмодуле программы пользователя содержатся команды пользователя в MC5-коде.

Устройства контроля включают в себя аппаратный контроль ошибочных состояний "задержка квитирования" (QVZ), "ошибка адресации" (ADF), "превышение времени цикла" (ZYK) и ошибок синхронизации в процессоре 80186.

o Квази—двухпортовая RAM

Квази—двухпортовая RAM позволяет организовать связь микропроцессоров 80186 и 8031

o Процессор интерфейса
Основой процессора интерфейса является микропроцессор 8051.
Организация интерфейса на микропроцессоре позволяет оптимальным образом решить эту задачу в условиях ограничения пространства.
Память программного интерфейса содержит программу управления интерфейсом .

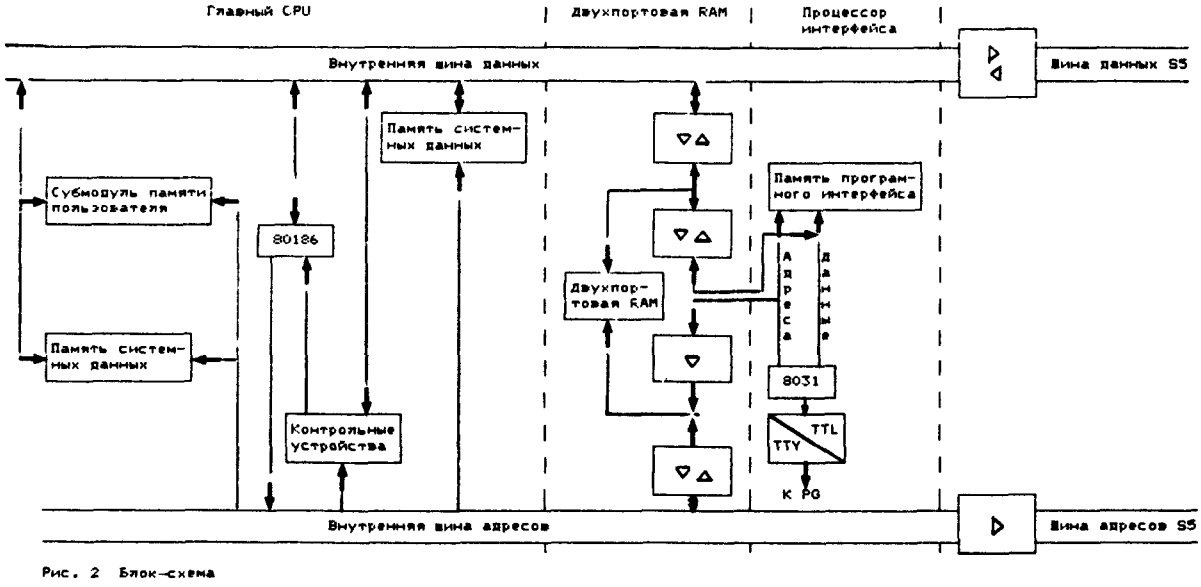


Рис. 2 Блок-схема

1.3.2 Субмодуль памяти пользователя

В R-процессоре для хранения программы пользователя можно использовать как модули EPROM (6ES5 376-....), так и модули RAM (6ES5 377-...).

o Субмодули EPROM

EPROM—субмодули программируются непосредственно на программаторе PG. Поставляются следующие типы субмодулей EPROM:

Объем памяти	Каталоговый номер
8 x 2^{10}	6ES5 376-0AA11
16 x 2^{10}	6ES5 376-0AA21
32 x 2^{10}	6ES5 376-0AA31 [1)]

o Субмодули RAM

Программа пользователя может быть записана в RAM-суёмодуль при помощи программатора через стандартный интерфейс R-процессора в режиме "on line". Для того, чтобы не потерять данные, необходимо подпитывать RAM-суёмодуль буферным напряжением (R-процессор и буферная батарея обязательно должны быть вставлены).

Можно применять следующие типы субмодулей RAM:

Объем памяти	Каталоговый номер
8 x 210 слов	6ES5 377-0AA11
16 x 210 слов	6ES5 377-0AB21
16 x 210 слов	6ES5 377-0AA31
32 x 210 слов	6ES5 377-0AA31
64 x 210 слов	6ES5 377-0AA31

1.3.3 Обработка прерываний

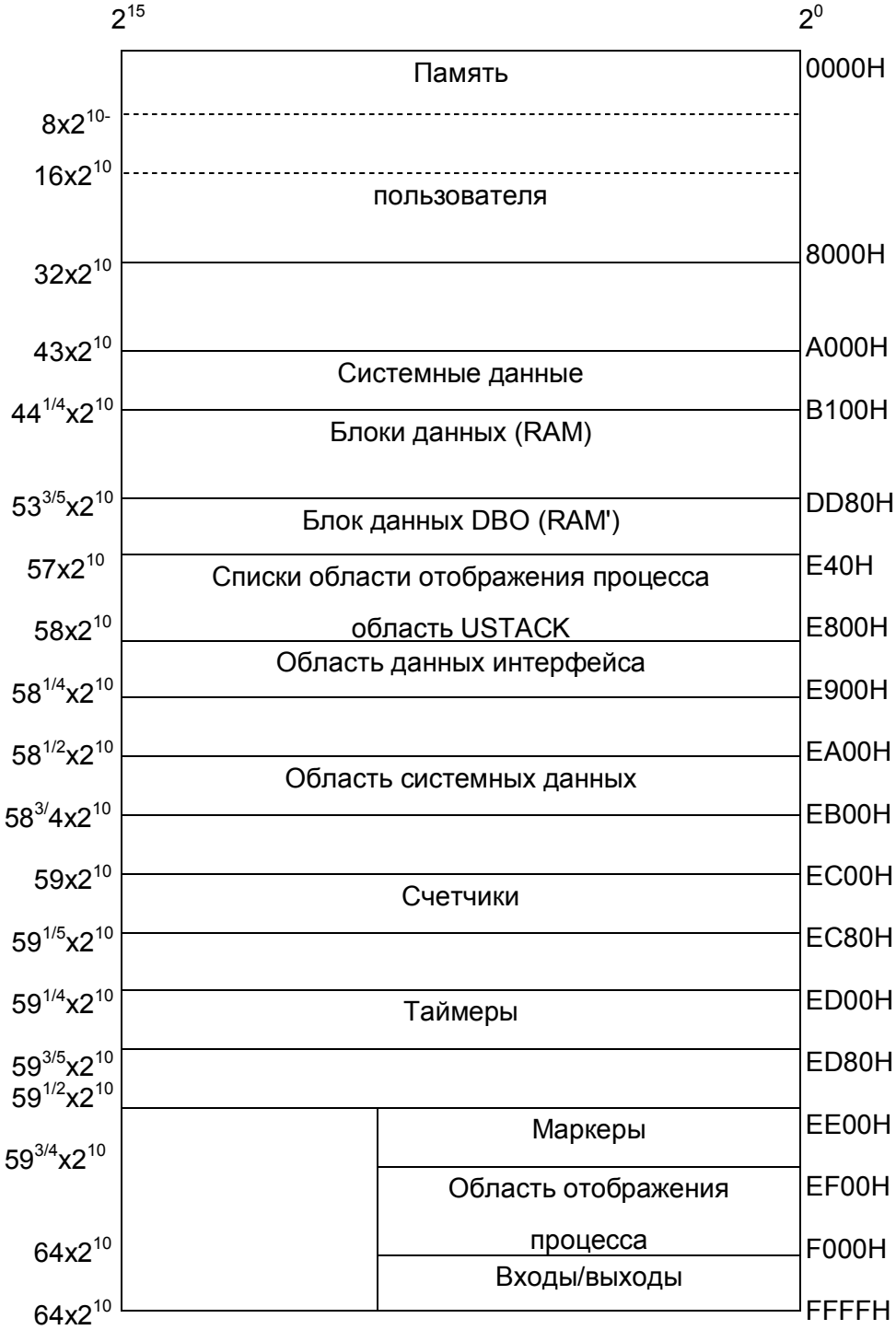
В распоряжение каждого процессора AG 135U предоставляется сигнал прерывания. Он используется, когда требуется быстрая реакция на какое-то событие или цепь событий, более важных, чем остальные. В этом случае циклическая обработка программы прерывается, и начинается обработка программы прерывания, помещенной в организационный блок OB2. Для более подробного рассмотрения этого режима следует обратиться к руководству по программированию C79000-B8576-C364.

Режим обработки прерывания возможен при использовании цифровых модулей, подходящих для .этих целей (например, 6ES5 432-...) или соответствующих модулей интеллигентной периферии.

[1]) Можно программировать только с помощью MEP-адаптера 6ES5 984-2AA11 на программаторе PG675 или PG685. Кроме того, требуется операционная система S5-DOS.

[2]) Удлиненный вариант только для пусконаладки

1.4. Распределение памяти



1.5 Технические данные

Класс зашиты …… ..... IP 00 (без защиты от попадания воды)
Рабочая температура.....…….от О до 55 °С
Температура транспортировки
и хранения.....…….. от -40 до 70 "с
Допустимая влажность (без
выпадения росы").....……...95% при 25 "С
Максимальная высота
над уровнем моря.…….....3500 м
Напряжение питания.....…… .5В +/—5%
Номинальный потребляемый ток……..2.2 А
Напряжение буферной батареи ……3.4. В
Ток буферной батареи (без RAM)……..20 mkA
Размеры…….……20.32 х 233.4 х 160 мм
Вес …….……около 0.5 кг

	Р-область	Q-область	Всего
Цифровые входы с отображением процесса..	макс.1024	-	макс.1024
Цифровые входы без отображения процесса.	макс.1024	макс.2048	макс 3072
или аналоговые входы……..	макс.64	макс. 128	макс. 192
Цифровые выходы с отображением процесса	макс.1024	-	макс.1024
Цифровые выходы без отображения процесса	макс.1024	макс.2048	макс.3072
или аналоговые выходы…….	макс.64	макс. 128	макс. 192

Маркеры……...	2048
Таймеры……...	128
Счетчики……...	128
Объем памяти пользователя…….	макс. 32К слов (16-битовых), EPROM или RAM
Скорость передачи данных серийного интерфейса…….	9600 бит/с
Программные блоки……..	256 PB
Шаговые блоки……..	256 SB
Функциональные блоки……	256 FB
Блоки данных…….	256 DB
Расширенные Функциональные блоки……	256 FX
Расширенные блоки данных……..	256 DX
Организационные блоки…….	ОВ 1…39
0В специальных функций……...	SF-OB 40...255
Стандартные функциональные блоки (SFB)…….	8 компактных регуляторов в 20 мс или макс. 64 регулятора, регистра сдвига

.

2 Монтаж

2.1 Установка и снятие модулей

Модули вынимаются из рамы центрального устройства при помощи легкого покачивания вверх-вниз и постепенного выдвижения на себя за ручки. Модули разрешается вставлять и вынимать только при отключенном контроллере (это не относится к модулям U-периферии).

2.2 Установочные места центрального устройства

Однопроцессорный режим:	R-процессор устанавливается на 11-е установочное место ПК 135U.
Многопроцессорный режим:	S-, R-, и/или M-процессоры устанавливаются на 11, 19, 27 и 35 установочные места AG 135U ,без промежутков, начиная с 11 места

3. Эксплуатация

3.1 Режимы работы

o Однопроцессорный режим

В раму контроллера вставляется один R–процессор:

При этом возможны постоянный доступ к 35-шине и непрерывный цикл.

o Многопроцессорный режим

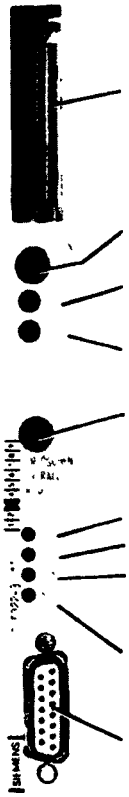
Вставляются от 2 до 4 R-, M- и/или S- процессоров: в этом случае доступ к S5-шине осуществляется через координатор KOR 923. Вводные и выходные маркеры межпроцессорного обмена, а также входы и выходы модулей, должны быть определены для каждого процессора адресным списком в DB1. В отсутствие этого списка адресов контроллер сохраняет состояние "СТОП". DB1 со списком адресов заносится в субмодуль RAM процессора с помощью программатора в режиме "on line" или программируется непосредственно на программаторе в субмодуле EPROM (см. Руководство по программированию, Раздел 1.4.3).

Работа всех процессоров синхронизируется координатором KOR 923.

Если один из процессоров переходит из циклической программы в "СТОП", то и все остальные процессоры переходят в "СТОП", при этом все цифровые выводы блокируются сигналом BASP. Для асинхронного или независимого пуска каждого процессора можно выбрать в KOR 923 тестовую функцию (см. Руководство по программированию C79000-B8500-C364, Раздел 5.2.3 и Руководство по эксплуатации KOR 923).

Процессор, предназначенный для целей высокоскоростного управления с замкнутым циклом (с временем сканирования, например, 20мс), в то же самое время не может быть использован для обмена большими массивами данных с коммуникационными процессорами. В таких случаях необходимо распределять задачи между процессорами.

3.2 Органы управления и элементы индикации



Разъем субмодуля памяти пользователя

Селектор режимов

Зеленый светодиод "RUN"

Красный светодиод "STOP"

Переключатель сброса

Светодиодный индикатор ошибок и сообщений
"QVZ"
"ADF"
"ZYK"

"BASP"

15-Полюсный разъем интерфейса связи с PG

Рис . 3 Передняя панель R-процессора

<u>селектор режимов</u>

Положение "RUN"

При включении в положение "RUN" R-процессор работает в циклическом режиме, и это индицируется зеленым светодиодом "RUN", циклическая работа означает, что считывание области отображения входов, выполнение программы пользователя в соответствии с последовательностью вызовов в OB1 или FBO, запись области отображения выходов, изменения в области маркеров связи (в соответствии с DB1) и переход на новый цикл обработки периодически повторяются, происходят автоматически (циклически).

Циклическая обработка программы прерывается лишь в тех случаях, когда происходит обработка таймера (запрограммированная задержка) , при обработке аппаратных. прерываний и эли когда происходит; обращение к контроллеру со стороны программатора через интерфейсный модуль.

Циклическая обработка программы прекращается, если обнаружена системная ошибка, ошибка программы или аппаратная неисправность (например, сбой напряжения питания NAU, задержка квитирования QVZ, ошибка замещения SUF).

Положение "STOP"

При переключении из положения "RUN" в положение "STOP" (или под воздействием определенных причин в самом контроллере) процессор останавливается. При этом индицируется сигнал BASP на передней панели, и блокируются модули выходов. Функция контроля возможна с помощью программатора, т.к. в этом случае подавляется сигнал блокировки BASP, и появляется доступ к выходным модулям.

Если селектор режимов при подаче напряжения питания стоит в положении "RUN" (в многопроцессорном режиме на всех S-, R- и/или M- процессорах, а также на координаторе KOR 923), то выполнится автоматический перезапуск, при условии, что не было ошибок кроме выключения напряжения и до выключения контроллер был в состоянии циклической обработки (см. Руководство по программированию C79000-B8500-C364).

Светодиод "RUN"

Светодиод постоянно горит:

-Это означает, что идет циклическая обработка программы.

Светодиод "STOP"

Состояние останова индицируется постоянным горением красного светодиода или его быстрым или медленным миганием.

светодиод горит постоянно:

- После включения-питания, если переключатель режима в положении "STOP", и не было обнаружено ошибок при инициализации. Возможен запуск.
- Циклическая обработка программы была прервана в многопроцессорном режиме, если прерывание было вызвано другим процессором, координатором 923 или переводом селектора режимов из "RUN" в "STOP".
- Имеет место аппаратная неисправность, причем не процессорах (NAU, BAU, PEU).
- После общего сброса.

Светодиод быстро мигает:

- Необходим общий сброс (см. "переключатель сброса"). Пуск контроллера возможен только после общего сброса и при условии устранения перед этим всех ошибок и неисправностей. Основания для общего сброса приведены в Руководстве по программированию C79000-B8500-C364.

Светодиод медленно мигает:

- Процессор обнаружил ошибку при выполнении циклической программы. Модуль переходит в состояние "СТОП", если в программе нет ошибок. После переводе селектора режимов из "RUN" в "STOP", светодиод должен гореть постоянно.
- Ошибка оператора (неправильный выбор стартовых параметров или ошибка в DB1).
- В программе пользователя запрограммирована команда остановки.

Переключатель сброса

o Общий сброс ("OVERALL RESET")

Вся память RAM стирается и инициализируется. Красный светодиод "STOP" должен гореть непрерывно.222

- Придерживая переключатель сброса в положении "OVERALL RESET" (общий сброс), селектор режимов переведите из положения "STOP" в "RUN", а затем опять в "STOP".

  Результат: Светодиод "STOP" быстро мигает, необходим общий сброс [1]).

- Придержите опять переключатель сброса в положении "OVERALL RESET", а селектор режимов переведите из положения "STOP" в "RUN", а затем опять в "STOP".
  Результат: Выполнен общий сброс, процессор остановлен; светодиод горит постоянно. Можно проводить новый пуск.

o Сброс ("RESET")
Смотри Раздел 3.3

[1]) В этот момент можно прерывать функцию "общий сброс":
Переведите переключатель режима из положения "STOP" в "RUN" и затем опять в "STOP", не касаясь переключателя сброса.

Результат: Общий сброс не произошел, и процессор сохранил состояние останова.

Светодиодный индикатор ошибок и сообщений

Светодиод "QVZ"

Горит, если программа обращается *к* периферийному модулю, который:

- (в однопроцессорном режиме) находился во время нового пуска в области отображения процесса (входы от о до 127, выходы от О до 127) и который был введен в так называемую 9-ю дорожку как реально существующий, но теперь не отвечает или
- (в одно или многопроцессорном режиме) был занесен в DB1 (список адресов) и был представлен как реально существующий, но теперь не отвечает или
- не отвечает или больше не отвечает при обращении к нему с помощью команд прямого доступа (L/T P..., L/T Q...).

Это означает следующее:
- Модуль неисправен.
- Модуль был вынут во время работы или во время останова или в отключенном состоянии без выполнения нового пуска. В ответ на появление ошибки QVZ в интерфейсных 0В можно запрограммировать определенные действия (см. руководство по программированию С79000-В8500--С364) .

Светодиод "ADF"

Во время нового пуска R-процессора операционной системой создается 9 дорожка. Все физически существующие периферийные модули отмечаются 1 в области отображения процесса, все те, которые не существуют, отмечаются О на 9 дорожке. В многопроцессорном режиме, или когда программируется список адресов в DB1, 9 дорожка создается на базе DB1. Если программой пользователя опрашивается адрес периферии по которому нет модуля, то процессор прерывает циклическую обработку программы. В ответ на появление ошибки "ADF" в интерфейсных 0В можно запрограммировать определенные действия (см. Руководство по программированию С79000-В8500-С364").

Светодиод "ZYK"

Зажигается, когда превышено максимально допустимое время цикла. Время цикла складывается из времен выполнения всех частей программы пользователя. Сообщение об ошибке "ZYK" прерывает циклическую обработку программы. Ответную реакцию на появление ошибки "ZYK" можно запрограммировать в интерфейсных 0В (см. Руководства по программированию С79000—В8500—С364).

Светодиод "BASP"

Зажигается, когда запрещена команда вывода. Цифровые выходы переводятся в безопасное состояние (высокого импеданса). Область отображения выходов не стирается. Сигнал BASP выводится, когда включается или выключается блок питания, когда напряжение блока питания ниже нормы или когда контроллер в состоянии "СТОП".

3.3 Режимы запуска

Все необходимые для циклической работы контроллера данные определены и установлены. Теперь контроллер может работать в циклическом режиме (см. Руководство по программированию). Системная программа позволяет проводить три режима пуска:

3.3.1 Новый пуск

Маркеры, таймеры, счетчики и область отображения процесса стираются. Обработка программы пользователя начинается с самого начала.

Контроллер должен быть остановлен. В многопроцессорном режиме селектор режимов координатора KOR 923 должен быть в положении "STOP".

Процессор сброшен и перейдет к циклической обработке программы. если:

- переключатель сброса R-процессора удерживается в позиции "RESET",
- селектор режимов переводится из положения "STOP" в "RUN",
- в многопроцессорном режиме селектор режимов координатора KOR 923 переводится из положения "STOP" в "RUN",

3.3.2 Ручной перезапуск

Состояние маркеров, таймеров, счетчиков и области отображения процесса сохраняется во время сброса. Обработка программы пользователя возобновляется из точки прерывания.

Перед остановкой контроллер должен работать в циклическом режиме. В многопроцессорном режиме переключатель координатора должен быть в положении "STOP".

Процессор переходит в режим циклической обработки программы, если:

- переключатель сброса R-процессора в среднем положении,
- селектор режимов переведен из положения "STOP" в "RUN" или
- в многопроцессорном режиме селектор режимов координатора KOR 923 переведен из положения "STOP" в "RUN"

3.3.3 Автоматический перезапуск

Состояние маркеров, таймеров, счетчиков и области отображения процесса сохраняется во время сброса. Обработка программы пользователя возобновляется из точки прерывания.

Автоматический перезапуск выполняется после включения питания, если:

- перед выключением напряжения питания контроллер работал в циклическом режиме,
- селектор режимов R-процсссора (в многопроцессорном режиме R-, M-, и/или S-процессоров и координатора) установлен в положение "RUN",
- субмодуль памяти пользователя не вынимался и
- буферная батарея функционирует нормально (гарантируется сохранение данных в субмодуле RAM) .

Тестовый режим: Смотри руководство по программированию C79000-B8500--C364.

4 Обслуживание

4.1 Назначение контактов задних разъемов

Задний разъем 1

	d	b	z
2		M	+ 5B
4	UBAT	PESP	
6	ADB 12	ADB 0	CPKL
8	ADB 13	ADB 1	MEMR
10	ADB 14	ADB 2	MEMW
12	ADB 15	ADB 3	RDY
14	IR	ADB 4	DB 0
16		ADB 5	DB 1
18		ADB 6	DB 2
20		ADB 7	DB 3
22		ADB 8	DB 4
24		ADB 9	DB 5
26		ADB10	DB 6
28	DSI	ADB11	DB 7
30	BUSEN	BASF	QUITT
32	BASPA	M	HALT

Задний разъем 2

	d	b	z
2		M	+5B
4			
6	M		
8			
10			
12			M
14			NAU
16			BAU
18		M	
20		STEU	
22	T x D	STOPPA	PEU
24		M	GEP
26	TEST	R x D	
28		PERO	
30		M 24 B	M 24B
32		M	+ 24B

4.2 Назначение контактов переднего разъема

1 Экран/масса/земля
2 RxD
3 +5B
4 +24B с шины
5 Земля/масса
6 TxD
7 TxD
8 Экран/земля/масса
9 RxD
10 24B земля
11 20 мА/источник
12 Земля/масса
13 20 мА/приемник
14 +5B
15 Земля/масса

SIEMENS

SIMATIC S5

RAM-модуль 377 (буферизованный) 6ES5 377-0B...

Руководство по эксплуатации Заказной No C79000-B8500-C615-02

Оглавление

1 Техническое описание

2 Режимы работ
2.1 "Нормальный" режим
2.2 Режим "Standby"
2.3 "Автономный" режим

3 Контроль батареи и неисправности батареи

4 Установка или замена буферной батареи

5 Установка буферного модуля RAM в CPU
5.1 Эксплуатация
5.2 Важные указания

6 Установка буферизированного модуля RAM в CP525
6.1 Эксплуатация
6.2 Важные указания

7 Технические данные

8 Назначение контактов разъема

9 Комплектующие

1 Техническое описание

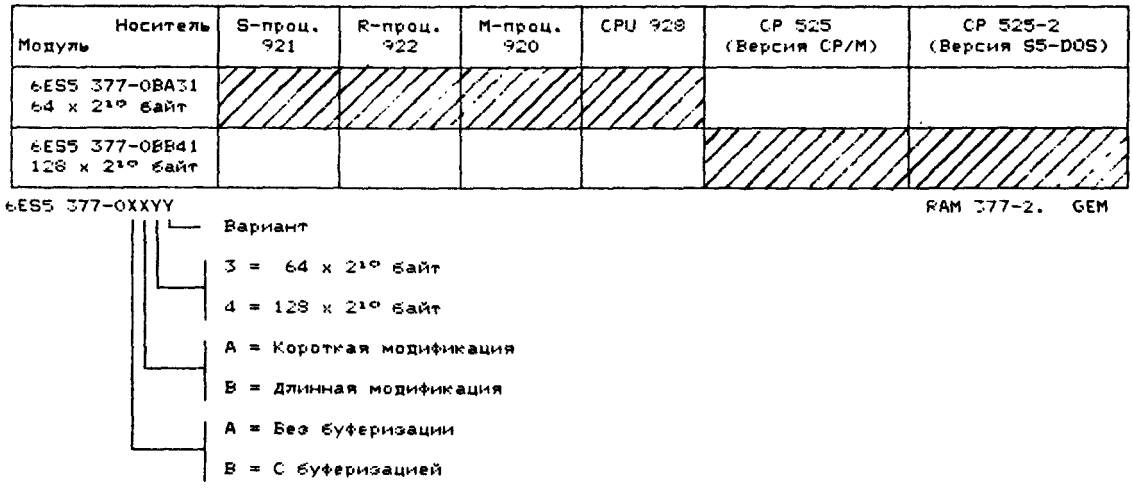
Если Вы для хранения своих программ используете модуль RAM 377 с буферной батареей, то можете этот модуль вынимать из CPU или CP без потери данных в модуле RAM: собственная батарея защищает этот модуль от потери данных и позволяет сохранить данные при повторной установке модуля.

Буферизованный модуль RAM может устанавливаться (см. Рисунок):

- в S-процессор, R-процессор, M-процессор и CPU 928 программируемого контроллера S5-135U
- в R-процессор, M-процессор и CPU 928 ПК S5-155U
- а также в коммуникационных процессорах CP 525 и CP 525-2.

Он применяется в тех случаях, когда необходимо сохранить программу пользователя вне CPU или CP.

Предлагаются следующие буферизованные модули RAM:



Важно! Буферизованный модуль RAM не является заменой модулю EPROM!

для фиксации батареи в RAM-модуле предусмотрена двухсторонняя заглушка.

Батарея в виде таблетки с петлями для подключения крепится с помощью винтов на два держателя модуля RAM (см. Раздел 2).

При правильной установке Вам будет виден светодиод неисправности батареи (см. Раздел 5).

2. Режимы работ

Существуют три режима работы буферизованного модуля RAM.

2.1 "Нормальный" режим

В этом режиме

- RAM установлен в CPU или CP.
- программируемый контроллер (AO) или устройство расширения (EG) включен,
- не задействованы ни основная батарея AG или EG, ни батарея модуля RAM.

Важно!
Снятие и установка модуля RAM в этом режиме приводит к потере или искажению данных и, поэтому, недопустимы!

2.2 Режим "Standby"

В этом режиме работы (режим Standby устройства):

- модуль RAM установлен в CPU или CP,
- AG или EG выключены,
- для буферизации модуля RAM используется основная батарея AG или EG,
- батарея модуля не задействована.

Важно!
Только в этом режиме работы снятие и установка модуля RAM не приводит к потере данных!

Если в этом режиме центральная батарея AG или EG окажется неисправной, то буферизацию данный обеспечит батарея модуля RAM. Поэтому потеря данных исключена.

2.3 "Автономный" режим

В этом режиме работы (режим Standby модуля):

- модуль RAM отсоединен от CPU или CP,
- батарея модуля берет на себя буферизацию модуля RAM; содержимое RAM сохраняется.

3. Контроль батареи и неисправности батареи

Контроль батареи

Батарея модуля RAM 377 контролируется. Если модуль RAM работает в нормальном режиме (RAM установлен в CPU или CP, напряжение питания подается от AG/EG), то при контроле батареи могут быть определены следующие ошибки:
a) Батарея модуля отсутствует.

b) Батарея модуля неисправна.

Светодиод неисправности батареи на передней панели модуля постоянно светится красным светом!

Внимание!
Временный сбой напряжения батареи модуля в "автономном" режиме (например при температуре хранения ниже 0° C или при смене батареи) контроль батареи не опознает, если после установки модуля RAM и включения AG/EG сбой напряжения будет устранен. Однако, при сбое напряжения могут быть потеряны или искажены данные!

Реакция CPU на появление ошибки данным

Если операционная система CPU опознает при запуске неверные данные в модуле RAM, то CPU переходит в состояние СТОП с быстрым миганием (требуется общий сброс). В USTACKe крестиком помечается бит "MOD-FE".

Реакция CP на появление ошибки данных

Если CP обнаруживает при запуске ошибочные данные в модуле RAM, то содержимое RAM полностью стирается или модуль переходит к подпрограмме обработки ошибки.

1 Установка или замена буферной батареи

Перед первой установкой модуля RAM необходимо подключить буферную батарею, делается это следующим образом:

o Освободите переднюю заглушку, предварительно открыв защелку.

o Установите батарею модуля и закрепите ее с помощью винтов слева и справа. Обратите при этом внимание на полярность (+/-).

o Закройте заглушку.

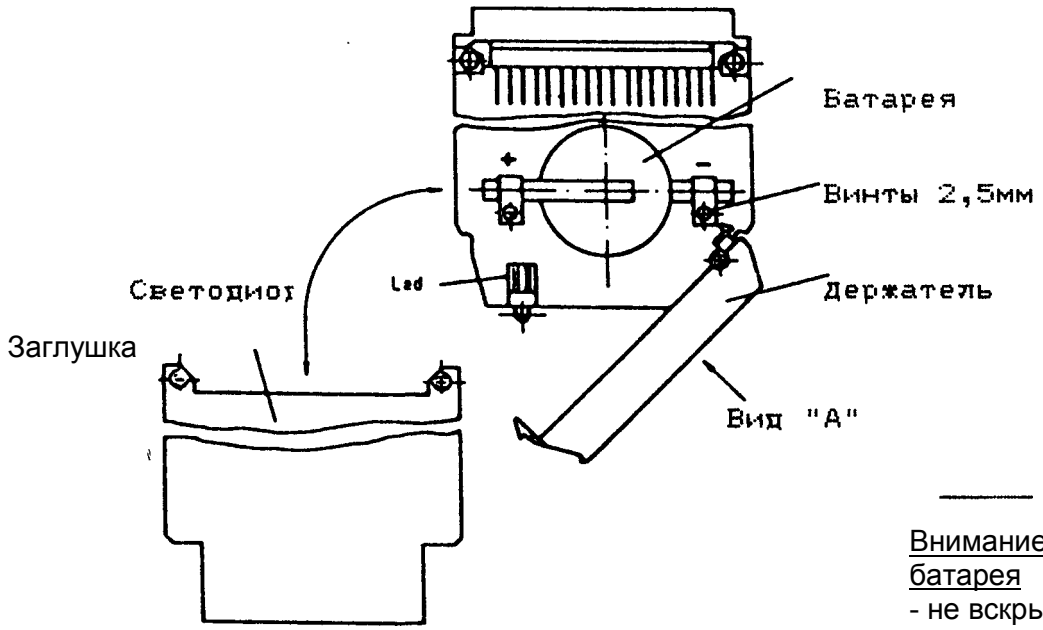
При следующим заменах батареи надо действовать следующим образом:

o Освободите переднюю заглушку, предварительно открыв защелку.

o Удалите винты слева и справа.

o Замените батарею и закрепите ее (следите за полярностью).

o Закройте заглушку



Внимание;Литиевая батарея
- не вскрывать
- не заряжать

Рис.: Схема монтажа буферизованного RAM модуля 377 (короткая и длинная модификации)

Следите, пожалуйста, за правильным хранением литиевой батареи!

5 Установка буферного модуля RAM в CPU

5.1 Эксплуатация

Исходная точка: CPU установлен в программируемом контроллере. Блок питания AG выключен. Селектор режимов CPU установлен в положение "STOP".

- Сначала установите батарею в модуль RAM (см. Раздел *2).*
- Установите модуль RAM в CPU.
- Включите блок питания программируемого контроллера.
- Произведите общий сброс.
- Подключите программатор (PG) к CPU.
- Запрограммируйте буферизированный модуль PAM в режиме on-line с помощью PG.
- После загрузки программы *в* модуль PAM произведите новый пуск CPU.

5.2 Важные указания

Если Вы установили в CPU **запрограммированный модуль RAM** и его содержимое не должно стираться, обратите внимание на следующие правила:

Перед остановкой буферизированного модуля PAM в CPU (например CPU 928), CPU должен быть полностью очищен. (Произведите общий сброс с другим модулем RAM; при каждом общем сбросе содержимое модуля RAM уничтожается!);.

Проверьте перед удалением модуля RAM исправность батареи модуля; непрерывное свечение светодиода красным светом при включенном контроллере указывает на неисправность буферной батареи! В этом случае при снятии МОДУЛЯ RAM с контроллера его содержимое будет потеряно.

Перед удалением/установкой модуля RAM в/из CPU выключите напряжение питания AG: только при этом можно быть уверенным в том, что не произойдет потеря данных RAM"

Помните, что буферизированный модуль PAM нельзя запрограммировать на программаторе, как EPPOM. а потом установить в CPU или CP!

6. Установка буферизированного модуля RAM в CP525

6.1 Эксплуатация

Исходная точка: CP установлен в программируемом контроллере или в устройстве расширения. Блок питания AO или EG выключен. Селектор режимов работы CP установлен в положение "STOP".

o Сначала установите батарею в модуль RAM (см. Раздел 2).

o Установите модуль RAM в CP.

o Включите блок питания программируемого контроллера или устройства расширения.

o Подключите программатор (PG) к CP.

o Запрограммируйте буферизированный модуль RAM в режиме on-line с помощью пакета программ PG COM 525.

o После загрузки программы пользователя с помощью пакета COM 525 в модуль RAM выполните пуск CP 525.

6.2 Важные указания

Если Вы установили в CP 525 запрограммированный модуль RAM и его содержимое не должно стираться, обратите внимание на следующие правила:

- Если буферизированный модуль RAM не был запрограммирован в CP 525, то при запуске он будет заново проинициализирован. Все данные теряются.

- Если буферизованный модуль RAM запрограммирован в CP 525 (3UA11), то при установке его в CP 525-2(-3UA11) во время запуска он будет заново проинициализирован. Все данные будут потеряны! То же самое действует и в обратном случае (программирование в CP 525-2, а установка в CP 525).
- Проверьте перед удалением модуля RAM исправность батареи модуля; непрерывное свечение светодиода красным светом при включенном контроллере указывает на неисправность буферной батареи! В этом случае при снятии модуля RAM с контроллера его содержимое будет потеряно.
- Перед удалением/установкой модуля RAM в/из CPU выключите напряжение питания AG: только при этом можно быть уверенным *в* том, что не произойдет потери данным в модуле RAM!

Помните, что буферизированный модуль RAM нельзя запрограммировать как модуль EPROM на программаторе с помощью пакета PROM 525!

7. Технические данные

Напряжение питания	+ 5В (4,75В…5,25В)
UCMOS	4,5В…5,25В в обычном режиме 2,7В…3,6В в режиме "Standby"
Потребление тока при 5В	тип. 35мА (макс.70) в байтном режиме тип. 70мА (макс.140) в словном режиме
Ток буферизации	тип. 13 мкА для 377-0ВА31 тип. 17 мкА для 377-0ВВ41
Батарея модуля	Литиевая таблетка 3В/200 мАч тип. CR 2430 (LF-1/2W) varta
Время буферизации	мин. 1 год при 25 °C
Действительное время доступа по адресу и S5	150 нС для 377-0ВА31 200 нС для 377-0ВВ41
Температура эксплуатации	0 при 55 °C
Температура хранения	0 при 60 °C
Относительная влажность	95% при 25 °C , без точки росы
Высота эксплуатации	макс. 3500 м над уровнем моря
Размеры (Н x Т x В) 377-0ВА31 377-0ВВ41	 55мм x 58мм x 14мм 55мм x94мм x 14мм
Вес	около 40г или 60г.

	c	b	a
1	SADB 12	M	+5B
2	SADB 0	SADB 1	SADB 2
3	SADB 3	SADB 4	SADB 5
4	SADB 6	SADB 7	SADB 8
5	SADB 9	SADB 10	SADB 11
fc	SADB 13	SADB 14	/R
7	/W	SDBH 0	SDBH 1
8	SDBH 2	SDBH 3	SDBH 4
9	SDBH 5	SDBH 6	SDBH 7
10	SDBL 0	SDBL 1	SDBL 2
11	SDBL 3	SDBL 4	SDBL 5
12	SDBL 6	SDBL 7	К 1
13	/CS 1	/CS 3	К 2
14	/CS 2	/STBY	К 3
15	UCMOS	PSM	К 4
16	-	M	К 5

SADB 0-14: Адресная шина 0-14 модуля памяти

SDBL 0-7: Шина данных модуля памяти Младший байт

SDBL O- 7: Шина данные модуля памяти Старший байт

/R, /W: Сигнал чтения/ записи

/CS1,/CS2,/CS3: Сигнал выбора кристалла памяти

К1-К5 : Метка модуля PSW

/STBY : Переключение на режим Standby

+5B : Подача напряжения питания

UCMOS ; Напряжение питания памяти (+5В или напряжение основной батареи)

M : Масса

9 Комплектующие

Обозначение	Заказной N9	Группа запасных частей	Место заказа
Буферная батарея	6ES5 9SO-ODA11	N	GWK

1) N = неремонтопригодно

SIEMENS

SIMATIC 55

Координатор 923A 6ES5 923—3UA11

Руководство по эксплуатации Заказной No. C79000-B8576-C263-07



Рис. 1 (Координатор 923A)

Оглавление Страница

1 Техническое описание

1.1 Область применения
1.2 Структура
1.3 Принцип действия
1.3.1 Арбитраж шин
1.3.2 Буфер обмена данными
1.4 Технические данные

2 Монтаж
2.1 Установка и снятие модулей
2.2 Установочные места в АО 1351-1

3. Эксплуатация
3.1 Органы управления
3.2 Режимы работы
3.3 Кодирование числа процессоров
3.4 Адресация буфера обмена данными
3.5 Назначение перемычек

4 Назначение контактов задних разъемов

5 Комплектующие

Техническое описание

1.1 Область применения

Координатор 923A устанавливается в программируемый контроллер (AG) 135U. При этом он выполняет две независимые Функции:

о Арбитраж шин

Координация работы CPU (центральных процессорных устройств) в многопроцессорном режиме, т.е.. при одновременной работе нескольких (от 2 до 4 S-, R- и/или M-) процессоров.

о Буфер обмена данными Это

функция обмена данными между процессорами

1.2 Структура

Координатор (KOR) 923 A представляет собой печатную плату двойного стандартного европейского формата.

На задней стороне модуля имеются два 48-полюсных разъема типа "2 ряда" для связи с внутренней S5-шиной контроллера.

Ширина передней панели — $1^1/^3$ размера стандартного установочного места.

На передней панели имеется трехпозиционный переключатель режимов работы (селектор режимов).

1.3 Принцип действия

1.3.1 Арбитраж шин

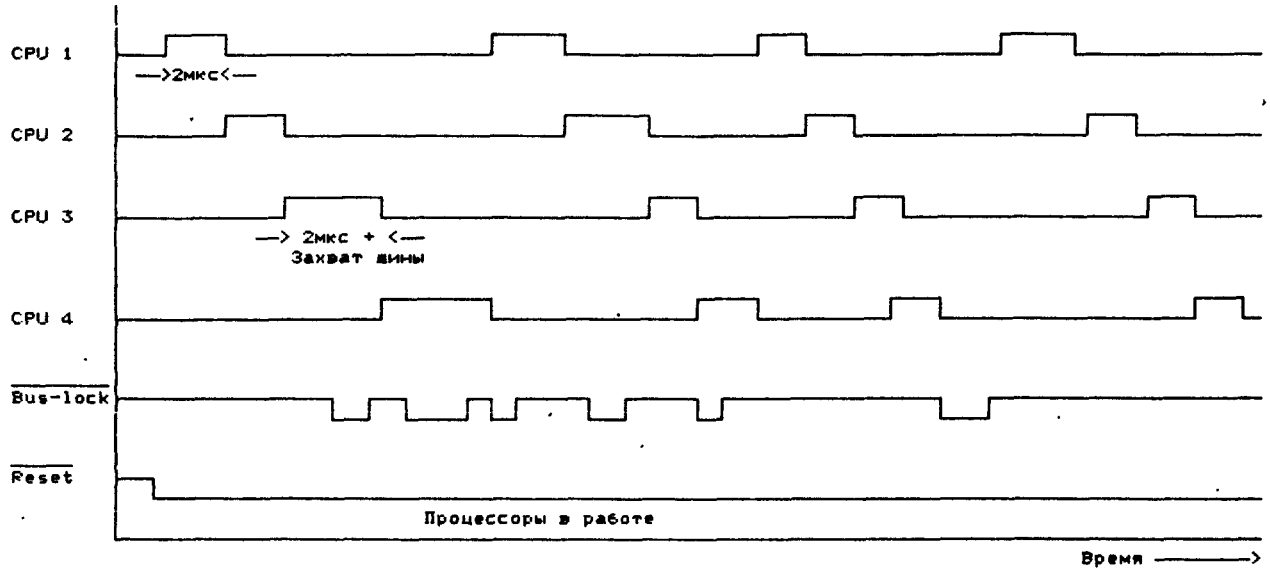
о Сигналы доступа к шине.

KOR 923A циклически разрешает доступ к шине контроллера каждому из 2/4 CPU. CPU может использовать шину только в этот момент.

Доступ к шине определяется функцией временного мультиплексирования.(переключения). Номер CPU задается перемычками на плате координатора. Время доступа к S5-шине ограничено 2мс для каждого CPU. Время доступа к шине можно увеличить с помощью захвата шины.

Сразу после сигнала сброса, инициированного источником питания шина предоставляется первому по порядку CPU и далее последовательно каждому в соответствии с присвоенными порядковыми номерами:

CPU1, CPU2, CPU3, CPU4, CPU1, CPU2, и т.д.



$\overline{\text{Bus-lock}}$ = Захват шины

$\overline{\text{Reset}}$ = Сброс

Рис. 2 Последовательность сигналов управления шиной

1.3.2 Буфер обмена данными

Буфер обмена данными KOR 923 A берет на себя функции маркеров связи (маркеров межпроцессорного обмена). Маркеры связи осуществляют циклический обмен данными между CPU (центральными процессорами) и между CPU и коммуникационными процессорами (CP) в AG 135U.

Буфер обмена данными представляет собой память RAM, буферизованную непосредственно от AG.

В руководствах по программированию процессоров объясняется, как программировать эту функцию.

1.4 Технические данные

Класс защиты	IP 00
Рабочая температура	0...55 °C
Температура транспортировки и хранения	-40...70 °C
Относительная влажность	95% при 25 °C (без конденсации)
макс. высота над уровнем моря	3500 м
Напряжение питания	5 В +/-5%
Номинальный потребляемый от источника 5В ток	0.5 А
Напряжение буферной батареи	2.7В от аккумулятора блока питания
Потребляемый от буферной батареи ток	100 нА
Размеры	20.32 х. 233.4 х 160 мм
Вес	не более 0.3 Кг

2 Монтаж

2.1 Установка и снятие модулей

Модули снимаются при помощи легкого покачивания вверх/вниз и одновременного с этим выдвижения на себя за ручки. Модули можно вставлять и вынимать, только если контроллер выключен.

2.2 Установочные места в AG 135U

KOR 923 A устанавливается на 3-е установочное место в AG 135U.

3 Эксплуатация

3.1 Органы управления

Единственным элементом управления координатора является трехпозиционный селектор выбора режима с позициями "RUN", "STOP" и "TEST", расположенный на передней панели модуля.

3.2 Режимы работы

о Состояние останова

Если селектор выбора режима после включения питания находится в положении "STOP" или если имеет место какая-нибудь другая ситуация останова, то все CPU находятся в состоянии останова.

o Новый пуск

Новый пуск происходит в том случае, если селектор режимов координатора переводится из положения "STOP" в "RUN" (обеспечивая тем самым подачу питания), при условии, что все CPU сброшены.

o Новый пуск с запоминанием, перезапуск

После перевода селектора режимов из положения "STOP" в "RUN" все CPU в принципе готовы к циклической работе, но это возможно лишь при условии, что до остановки контроллер работал в цикле, что селекторы режимов всех процессоров — тоже в положении "RUN" и что отсутствует ситуация, требующая остановки контроллера.

o Автоматический новый пуск, автоматический перезапуск

Если при включении питания селектор режимов находится в положении "RUN", то происходит новый пуск с запоминанием (S-процессор -3UA11 и -3UA12) или автоматический перезапуск (S-процессор -3UA21 и R-процессор), но это при условии, что селекторы режимов всех CPU тоже в положении "RUN", и до останова контроллер работал в циклическом режиме.

o Тест

Тест-функция используется для инициализации системы при многопроцессорной работе. Тест-функция активируется при установке перемычки 3-14 на штекере 45. При переключении селектора режимов из положения "STOP" в "TEST" контроллер входит в режим циклической работы. Сигнал "BASP" подавляется для тех CPU, которые находятся в состоянии останова благодаря своим селекторам режимов. Однако, некоторые CPU могут быть в состоянии работы без блокирования сигналом "BASP" цифровых выходных модулей.

Если в работе CPU, включенного в "RUN", обнаруживается ошибка, которая вводит его в состояние останова, то Тест-функция отключается, и выдается сигнал "BASP", блокирующий цифровые модули выходов.

После того, как старт системы полностью подготовлен. Тест должен быть отключен во избежание ошибок управления.



Рис. 3 Установка Тест-функции

3.3 Кодирование числа CPU

Каждому S-, R- и/или M-процессору в многопроцессорном режиме с помощью перемычек на плате координатора 923A присваивается порядковый номер.

Номер CPU	Перемычки штеккера 62
2 3 4	7-10, 8-9 7-10 8-9

При установке в контроллер CPU 928 с другими процессорами необходимо помнить, что CPU ,928 всегда- устанавливается справа .от остальных процессоров. Если используются два CPU 928, то для них надо освободить три установочных места, т.е. должна быть установлена перемычка 7-10.

3.4 Адресация буфера обмена данными

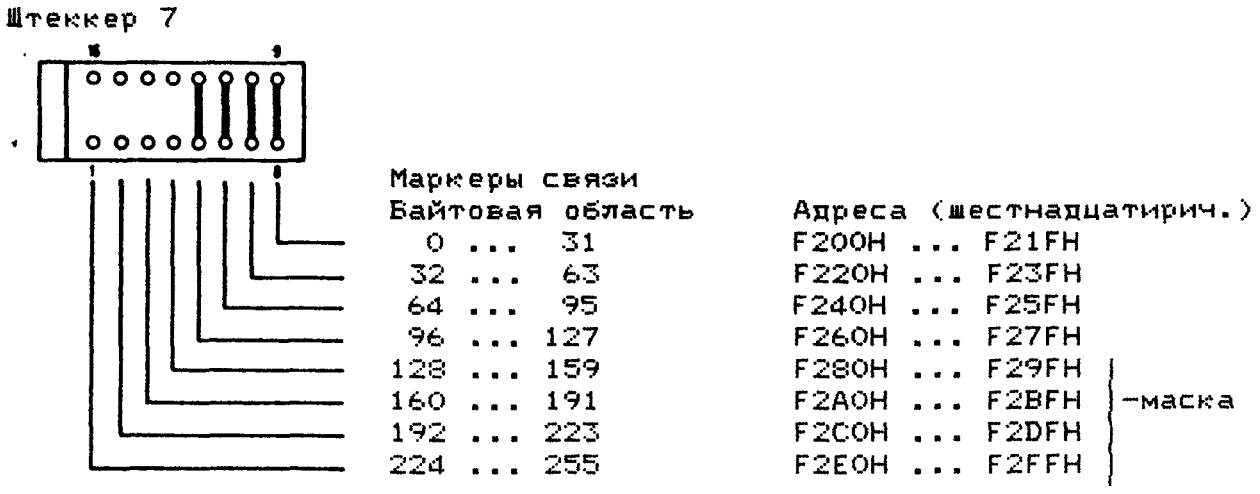
Область маркеров межпроцессорной связи занимает в памяти пространство адресов с F200H до F2FFH, всего 256 байт. Она может заполняться блоками по 32 байта. Если коммуникационные процессоры не используются, вся зта область доступна для контроллера и может использоваться нм для Функций маркеров межпроцессорной связи.

Удалением перемычек на штеккере 7 один или несколько 32—байтовых блоков могут маскироваться, т.е. защищаться от записи.

В случае использования маркеров межпроцессорной связи в CP должно применяться маскирование соответствующих областей в координаторе.

Пример:

Необходимо маскировать 4 32-байтовых блока маркеров связи со старшими адресами:



3.5 Назначение перемычек

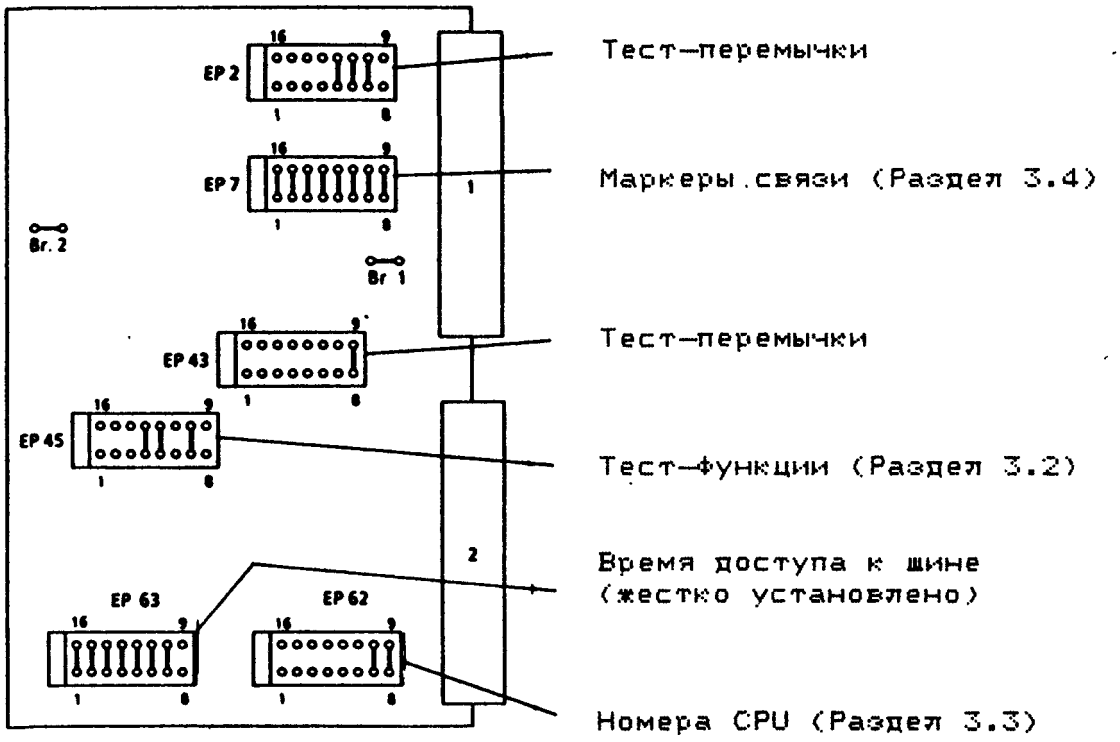


Рис. 4 Размещение перемычек и кодирующих штекеров
(показан вариант стандартной поставки)

4 Назначение контактов задних разъемов

Задний разъем 1

	d	b	z
2		M 5B	+ 5B
4	UBAT		
6		ADB 0	CPKL
8		ADB 1	MEMR
10		ADB 2	MEMW
12		ADB 3	RDY
14	BUSEN 1	ADB 4	DB 0
16	BUSEN 2	ADB 5	DB 1
18	BUSEN 3	ADB 6	DB 2
20	BUSEN 4	ADB 7	DB 3
22		ADB 8	DB 4
24		ADB 9	DB 5
26		ADB10	DB 6
28	DSI	ADB11	DB 7
30			
32		M 5B	HALT

Задний разъем 2

	d	b	z
2		M 5B	+ 5B
4			
6			
8			
10			
12			
14			NAU
16			
18			
20		STEU	
22		STOPPA	
24			
26	TEST		
28		PERO	
30			
32		M 5B	

5 Комплектующие

Кодирующий штекер C79334-A3011-B12

SIEMENS

SIMATIC S5

Координатор 923C 6ES5 923-3UC11

Руководство по эксплуатации Заказной No. C79000-B8500-C349-05

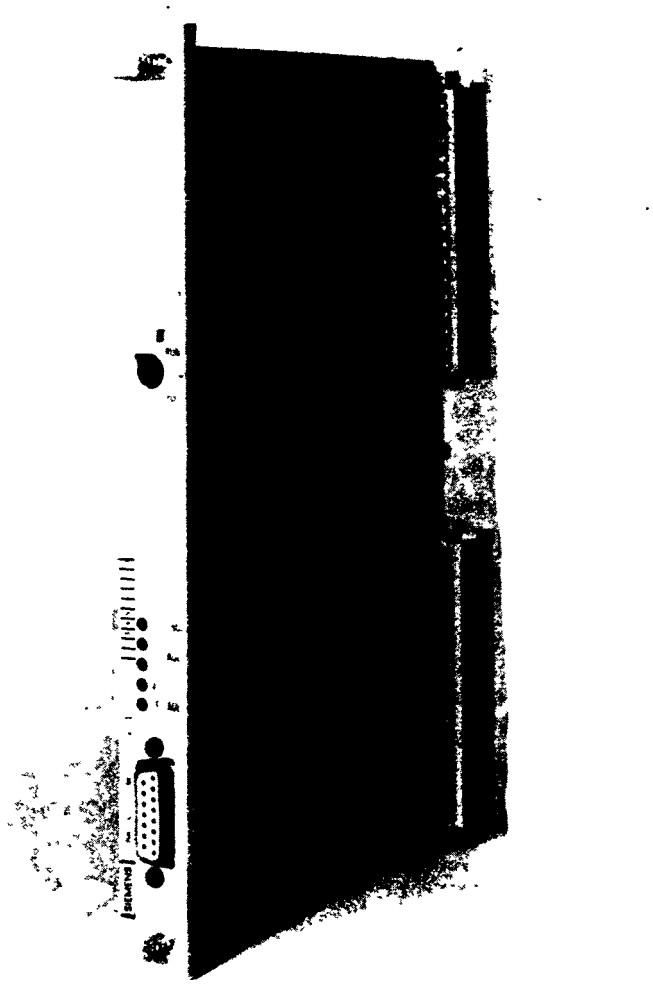


Рис. 1 координатор 923 C

Оглавление Страница

1 Техническое описание
1.1 Область применения
1.2 Структура
1.3 Принцип действия
1.3.1 Арбитраж шин
1.3.2 Контроль времени доступа *к* шине
1.3.3 Буфер обмена данными
1.3.4 PG-мультиплексор (PG-MUX)
1.4 Технические данные

2 Монтаж
2.1 Установка и снятие модулей
2.2 Установочные места в контроллере

3 Эксплуатация
3.1 Органы управления и элементы индикации
3.2 Режимы работы
3.3 Предустановка координатора
3.4 Предустановка PG-мультиплексора
3.5 Адресация буфера обмена данными
3.6 Перемычки блокировки сигналов координации
3.7 Регистр ошибок
3.8 Размещение перемычек и кодирующих штекеров

4 Обслуживание
4.1 Назначение контактов разъемов
4.2 Комплектующие

1 Техническое описание

1.1 Область применения

Координатор (KOR) 923 C устанавливается в программируемый контроллер (AG) 135U (кроме модификаций -3KA12 и -3KB12), а также в устройство расширения (EG) 185. Он обеспечивает выполнение трех независимых функций:

o Арбитраж шин

Координация работы CPU (центральных процессорных устройств) в многопроцессорном режиме, т.е. при одновременной работе нескольких (от 2 до 4) CPU (S-процессор/CPU921, R-процессор/CPU922, M-процессор/CPU920, CPU928 или CPU946/947).

o Буфер обмена данными
Это функция обмена данными между процессорами

o Мультиплексор программатора (PG—MUX)

Централизованный доступ с соответствующих программаторов (PG) по последовательному интерфейсу KOR 923 C к последовательным PG-интерфейсам максимум 8 модулей AG.

Эта Функция позволяет связать KOR 923 C через коммуникационный процессор (CP) 535 с фирменной шиной SINEC H1.

Для управления PG-мультиплексором необходима операционная система S5-DOS.

1.2 Структура

KOR 923 C представляет собой печатную плату двойного стандартного европейского формата.

На задней стороне модуля имеются два 48—полюсных разъема типа "2 ряда" для связи с внутренней S5-шиной контроллера.

Ширина передней панели — $1^{1/3}$ размера стандартного установочного места.

В верхней трети передней панели находится гнездо с крышкой. Открыв крышку, Вы получаете доступ к кодирующим штеккерам, с помощью которых можно произвести установку параметров модуля.

На передней панели имеется также трехпозиционный переключатель режима работы (селектор режимов).

Здесь же 5 небольших светодиодов, которые служат для индикации ошибок и ошибочных состояний.

Координатор 923 C через передний 15-полюсный разъем можно соединить с программатором, прибором диагностики, панелью оператора или коммуникационным процессором.

1.3. Принцип действия

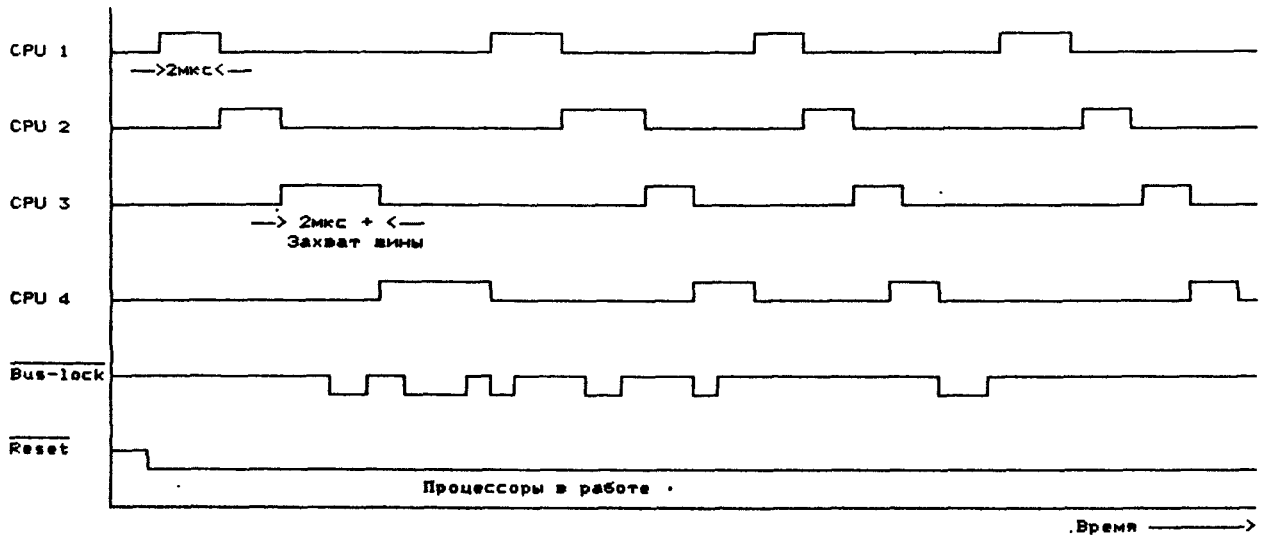
1.3.1. Арбитраж шин

– Сигналы доступа к шине

KOR 923 C циклически разрешает доступ *к* внутренней шине каждому из 2...4 CPU в AG 135U или 155U. Процессор может использовать шину только во время своего доступа к; ней.

Доступ к шине определяется функцией временного мультиплексирования (переключения). Номер процессора задается перемычками на кодирующем штекере передней панели. Время доступа к S5—шине ограничено для каждого процессора 2-мя мкс. Время доступа к шине можно увеличить с помощью захвата шины.

Сразу после сигнала сброса, инициированного источником питания, шина предоставляется первому по порядку процессору и далее последовательно каждому в соответствии с присвоенными порядковыми номерами:
CPU 1, CPU 2, CPU 3, CPU 4, CPU 1, CPU 2 и т.д.



$\overline{\text{Bus-lock}}$ = Захват шины

$\overline{\text{Reset}}$ = Сброс

Рис. 2 Последовательность сигналов управления шиной

1.3.2 Контроль времени доступа к шине

Сигнал захвата шины декодируется из 4-х старших адресным битов и равен 0, когда все четыре старших бита равны 1 (адресное пространство от 60к до 64к) . Сигнал выдается самим CPU после того, как он уже получил разрешение на доступ к шине от координатора. Время доступа CPU к шине увеличивается на время длительности сигнала захвата шины (см. Рис.2). Фирма в стандартной поставке устанавливает контроль длительности времени захвата шины на уровне 2мс. Если этот сигнал превышает 2мс, то координатор 923С останавливает все CPU.

Номер процессора, который превысил максимальное время захвата шины, помещается в регистр по адресу FEFFH, который доступен контроллеру, при этом загорается светодиод "BUS FAULT" на передней панели координатора. Этот регистр и светодиод сбрасываются, когда пропадает сигнал, вызывающий состояние останова.

1.3.3 Буфер обмена данными

Буфер обмена данными KOR 923 С берет на себя функции маркеров межпроцессорной связи. Маркеры межпроцессорной связи осуществляют циклический обмен данными между CPU в AG 135U.

Буфер обмена данными содержит четыре свободных страницы памяти (буфера связи) для обмена блоками данных между CPU.

В руководствах по программированию процессоров объясняется, как программировать эти функции.

Буфер обмена данными представляет собой память RAM, буферизованную непосредственно от AG.

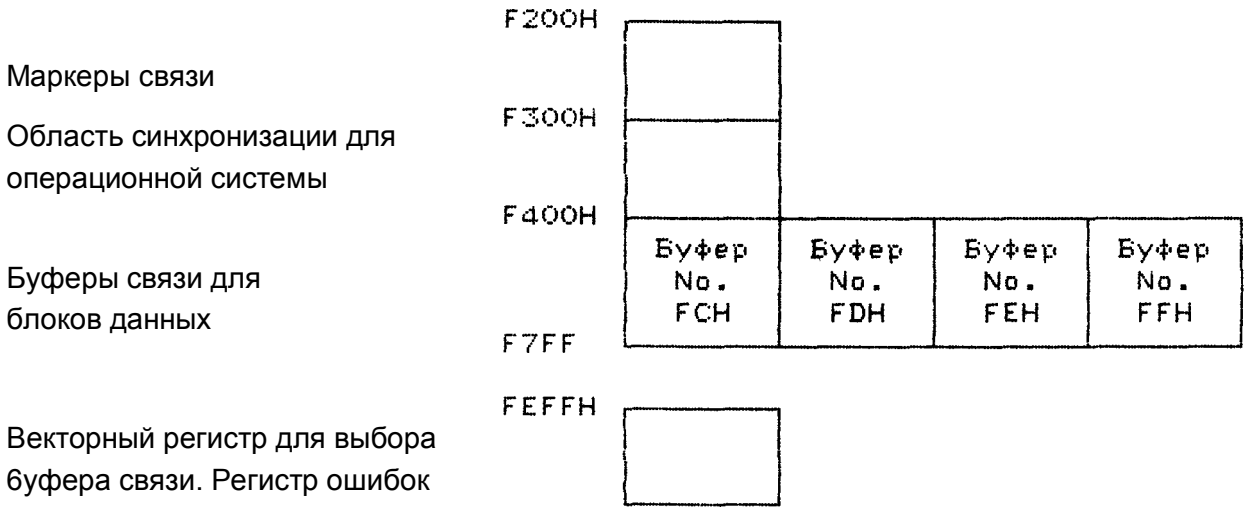


Рис. 3 Распределение памяти буфера обмена данными по S5-шине

— Процедура адресации буфера связи (векторного регистра)

Векторный регистр используется для выбора отдельным областей в (общем пространстве памяти. Это 8-битовый регистр, который может быть записан по адресу FHFFH.

Имеются четыре буфера связи, каждый объемом 1К. Каждому буферу присвоен идентификационный номер. Эти номера установлены в координаторе и не могут быть изменены.

Каждый доступ к буферу связи предваряется загрузкой соответствующего номера в векторный регистр.

Эти номера не должны использоваться в других модулях того же контроллера, чтобы избежать дублирования адресов.

В момент включения блока питания векторный регистр стирается, то есть он содержит номер ОН.

Для обмена данными с этой областью памяти используются специальные процессорные функции, которые описаны в руководстве по программированию R-процессора. S-процессор A, 6ES5 921-3UA11 не может оперировать с этими функциями.

1.3.4 PG-мультиплексор (PG-MUX)

TTY-интерФейс на передней панели координатора 923 С может подключать к 8—ми последовательным интерфейсам в контроллере. Эту Функцию выполняет микропроцессор 8031 с помощью мультиплексора.

Эти мультиплексированные интерфейсы имеют TTL—уровень и соединяются с другими модулями через задний разъем 2 и шину контроллера.

- Процедура выбора последовательного интерфейса

Все модули контроллера обслуживаются мультиплексором в соответствии с присвоенными им номерами (от 1 до 31 в десятичном счислении). Минимальный номер (базовый адрес) устанавливается в двоичном коде на кодирующем штекере S2. Этим, максимум 8 номерам, соответствуют в AG 135 U установочные места под номерами 11, 19, 27, 35, 43, 51, 59 и 67 (младший номер у 11-го установочного места).

Все 8 номеров (или установочных мест) назначаются кодирующим штекером 33: младший номер переключателем S3.1, старший — S3.8.

Если какие-то места не заняты, или предполагается с какими-то модулями работать через им передние интерфейсные разъемы, то соответствующие места необходимо маскировать при помощи штекера S3.

Передний разъем модуля, который предполагается использовать в режиме мультиплексирования, не должен быть занят.

1.5 Технические данные

Класс защиты	IP 00
Рабочая температура	O...55 °C
Температура транспортировки и хранения	-40...+70 °C
Относительная влажность (без конденсации)	95% при 25 °C
Макс. высота над уровнем моря	3500 м
Напряжение питания	5В +/-5% 24В +25/-15%
Потребляемый от источника 5В ток	тип. 1.1 А
Потребляемый от источника 24В ток	60 мА
Минимальное напряжение батареи для CMOS RAM	2.7 В
Потребляемый от буферной батареи ток	2 мкА
Квитированное время доступа к области связи через S5-шину	320 нс
Скорость передачи данным последовательного интерфейса	9600 бит/с
Кабель связи	экранированный 4-проводный кабель связи с PG, тип: 6E25 731-1
1 Максимальная дальность передачи данных	1 км при скорости 9600 бит/с
Вес	не более 0.3 кг
Размеры	20.32мм x 233.4мм x 160 мм

2 Монтаж

2.1 Установка и снятие модулей

Модули снимаются при помощи легкого покачивания вверх-вниз и одновременного с этим выдвижения на себя за ручки. Модули можно вставлять и вынимать, только если контроллер выключен.

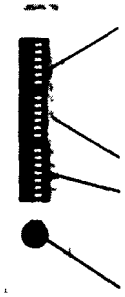
2.2 Установочные места в контроллере

Многопроцессорный режим и PG-MUX:
- в AG 135U на установочное место 3
- в AO 155U на установочное место 3

Только в режиме PG-MUX:
- в EG 135U на установочное место 11
- в EG 155U на установочное место 19

2 Эксплуатация

3.1 Органы управления и элементы индикации



Предустановка координатора:
S1.3 Тест-функция
S1.4...S1.6 номер CPU

Предустановка PG-мультиплексора:
S2.2...S2.6 базовый адрес
S3.1...S3.8 активизация ячейки

Селектор режимов для перехода в Новый пуск и Тест всех CPU при многопроцессорной работе



Светодиоды "BUS FAULT" ("ошибка шины")
Они соответствуют каждому CPU и загораются, когда превышено максимальное время доступа к шине

Светодиод "IF FAULT" (если ошибка")
Индицирует ошибку последовательного интерфейса

15-полюсный разъем интерфейса связи с программатором

Рис.4 Передняя панель координатора KOR 923C

3.2 Режимы работы

o Состояние останова

Если селектор выбора режима после включения питания наводится в положении "STOP" или если имеет место какая-нибудь другая ситуация останова, то все CPU наводятся в останове.

o Новый ПУСК

Новый пуск происходит в том случае, если селектор режимов координатора переводится из положения "STOP" в "RUN" (после подачи напряжения питания), при условии, что все CPU перед этим сброшены.

o Новый пуск с запоминанием, перезапуск

После перевода селектора режимов из положения "STOP" в "RUN" все CPU в принципе готовы к циклической работе, но это возможно лишь при условии, что до остановки контроллер работал в цикле, что селекторы режимов всех процессоров - тоже в положении "RUN" и что отсутствует ситуация, требующая остановки контроллера.

o Автоматический новый пуск, автоматический перезапуск

Если при включении питания селектор режимов находится в положении "RUN", то происходит новый пуск с запоминанием (S—процессор -3UA11 и -3UA12) или автоматический перезапуск (S-nроцессор - 3UA21 и R - процессор), но это при условии, что селекторы режимов всех CPU тоже в положении "RUN" и до останова контроллер работал в циклическом режиме.

o Тест

Тест-функция используется для инициализации системы при многопроцессорной работе. Тест-функция активируется при установке переключателя S1.3 кодирующего штекера. При переключении селектора режимов из положения "STOP" в "TEST" контроллер входит в режим циклической работы. Сигнал "BASP" подавляется для тех CPU, которые находятся в состоянии останова благодаря своим селекторам режимов. Однако, некоторые CPU могут быть в состоянии работы без блокирования сигналом "BASP" цифровых выходных модулей.

Если в работе CPU, включенного в "RUN", обнаруживается ошибка, которая вводит его в состояние останова, то Тест-функция отключается, и выдается "BASP"-сигнал, блокирующий цифровые модули выходов, работающие с этим CPU. Другие CPU при этом продолжают работать в циклическом режиме.

После того, как пуск системы полностью подготовлен. Тест должен быть отключен во избежание ошибок управления.

3.3 Предустановка координатора

Стандартная поставка;

	вкл	вык	
S1.1		X	
2		X	
3		X	Тест (см. Раздел 3.2)
4	X		количество CPU = 2 [1]
5		X	количество CPU = 3 [1]
6		X	количество CPU = 4 [1]

[1]) Может быть включен только 1 из 3 переключателей, т.к. они показывают количество процессоров, установленных в контроллере.

3.4 Предустановка PG-мультиплексора"

о Базовый адрес

Кодирующий штекер S2 предназначен для установки базового адреса, который может иметь значение между 1 и 31 (он необходим для адресации процессоров контроллера, имеющих выход на внешний разъем координатора). Мультиплексируемый модуль может иметь базовый адрес или любой из 7 следующих адресов. Базовый адрес вычислялся как сумма двоичных разрядов, выбранным кодирующими переключателями.

Стандартная поставка:

	вык	вкл	
S2.1	X		—
2	X		16
3	X		8
4	X		4
5	X		2
6		X	1 (базовый адрес = 1)

о Активизация адресов

Адреса модулей и номера установочных мест, которые управляются координатором KOR 923 C, устанавливаются кодирующим штекером S3.

Стандартная поставка:

	вык	вкл		Установочное место (в AG 135U)
S3.1	X		базовый адрес + 0	11/CPU 1
2	X		базовый адрес + 1	19/CPU 2
3	X		базовый адрес + 2	27/CPU 3
4	X		базовый адрес + 3	35/CPU 4
5	X		базовый адрес + 4	43
6	X		базовый адрес + 5	51
7	X		базовый адрес + 6	59
8	X		базовый адрес + 7	67

Пример: Необходимо установить базовый адрес 10 для модулей контроллера 135U, расположенных в ячейках 11, 35, 43 и 67, управляемых координатором COR 923C.

Установка базового адреса:

	выкл	вкл	
S2.1	X		—
2	X		16
3		X	8 ——> 8
4	X		4
5		X	2 ——> +2
6	X		1
			10 ——> базовый адрес

Активизация требуемых мест установки:

	выкл	вкл		Устан. место (в AG 135U)	Номер управляемого места	Адрес
S3.1		X	баз.ад.+ 0	11	11	10
2	X		баз.ад.+ 1	19		
3	X		баз.ад.+ 2	27		
4		X	баз.ад.+ 3	35	35	13
5		X	баз.ад.+ 4	43	43	14
6	X		баз.ад.+ 5	51		
7	X		баз.ад.+ 6	59		
8		X	баз.ад.+ 7	67	67	17

3.5 Адресация буфера обмена данными

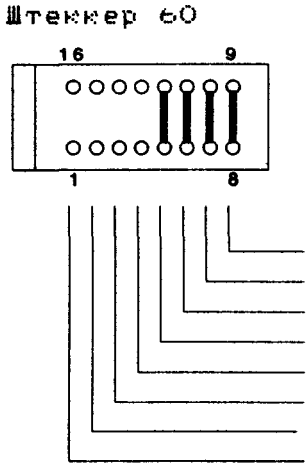
Область маркеров межпроцессорной связи (маркеров связи) занимает в памяти пространство адресов с F200H до F2FFH, всего 256 байт. Она может заполняться зонами по 32 байта. Если коммуникационные процессоры не используются, вся зта область доступна для контроллера и может использоваться им для функций маркеров связи.

Размещением перемычек на штекере 60 одна или несколько 32-байтовых зон могу маскироваться, т.е. защищаться от записи.

В случае использования коммуникационными процессорами маркеров связи должно применяться маскирование соответствующих областей в координаторе.

Пример

Необходимо маскировать 4 32-байтовых зоны маркеров связи со старшими адресами:



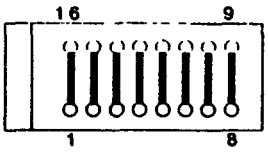
Маркеры связи

Байтовая область	Адреса (шестнадцатирич.)	
0 ... 31	F200H ... F21FH	
32 ... 63	F220H ... F23FH	
64 ... 95	F240H ... F25FH	
96 ... 127	F260H ... F27FH	
128 ... 159	F280H ... F29FH	|
160 ... 191	F2A0H ... F2BFH	|-маска
192 ... 223	F2C0H ... F2DFH	|
224 ... 255	F2E0H ... F2FFH	|

3.6 Перемычки блокировки сигналов координации

Координатор 923C при работе в качестве PG-муэтьтиплексора может устанавливаться в устройства расширения EG 1851-1 и EG 1861-1. В этом режиме координационные сигналы должны быть отключены.

Штеккер 61

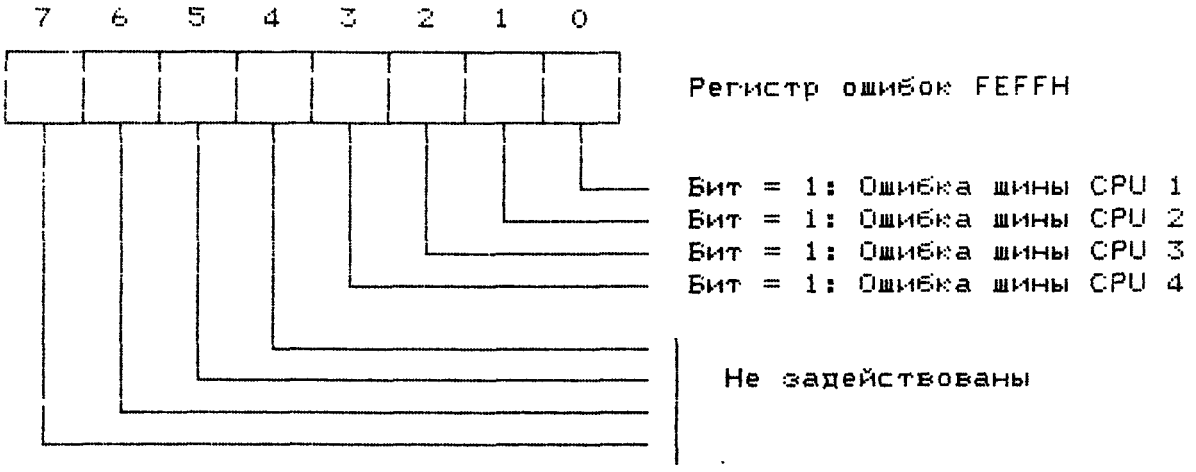


Координация	
действует	блокирована
все перемычки замкнуты	все перемычки разомкнуты

В состоянии поставки все перемычки замкнуты.

3.7 Регистр ошибок

Регистр ошибок имеет 8 разрядов и расположен по адресу FEFFH. Обращение к нему происходит, если появляется ошибка шины. Каждому CPU в регистре ошибок соответствует один бит, и он устанавливается в 1, если на шине данного CPU обнаружена ошибка. Регистр стирается каждый раз, когда сигнал останова не активирован.



Регистр ошибок может быть считан любым CPU, чтобы выполнить центральные функции.

3.8 Размещение перемычек и кодирующих штекеров

На рисунке показано стандартное положение перемычек

Предустановка координатора
(см. Раздел 3.3)

Предустановка PG-мультнплексора
(см. Раздел 3.4)

X4...X6 – тест-перемычки

Штеккер 60: Адресация буфера обмена
данными (см. Раздел 3.5)

Штеккер 61: Блокировка сигналов
координации (см. Раздел 3.6)

Штеккер 62: Контроль шины
(см. Раздел 1.3.2)

Штеккер 63: Арбитраж шины
(см. Раздел 1.3.1)

Штеккер 64: - тест-перемычки

Положение перемычек X4...X6 и штеккеров 62...64 изменять нельзя

3 Обслуживание

4.1 Назначение контактов задних разъемов

Задний разъем 1

	d	b	z
2		M 5B	+ 5B
4	UBAT		
6	ADB 12	ADB 0	CPKL
8	ADB 13	ADB 1	MEMR
10	ADB 14	ADB 2	MEMW
12	ADB 15	ADB 3	RDY
14	BUSEN 1	ADB 4	DB 0
16	BUSEN 2	ADB 5	DB 1
18	BUSEN 3	ADB 6	DB 2
20	BUSEN 4	ADB 7	DB 3
22		ADB 8	DB 4
24		ADB 9	DB 5
26		ADB10	DB 6
28	DSI	ADB11	DB 7
30		BASP	
32		M 5B	HALT

Задний разъем 2

	d	b	z
2		M 5B	+ 5B
4			
6	RXD 8		
8	TXD 8		
10	RXD 7		
12	TXD 7	RXD 6	
14	RXD 5	TXD 6	NAU
16	TXD 5	RXD 4	
18	RXD 3	TXD 4	
20	TXD 3	STEU	
22	RXD 1	STOPA	
24	TXD 1	RXD 2	
26	TEST	TXD 2	
28		PERO	
30			M 24B
32		M 5B	+ 24B

Передний разъем

Контакт	Назначение
1	Масса ($M_{экр}$)
2	Приемник TTY (–)
3	PL [1])
4	+24B
5	PL [1])
6	Передатчик TTY (+)
7	Передатчик TTY (–)
8	Масса ($M_{экр}$)
9	Приемник TTY (+)
10	24B масса (источник тока (–) 20 мА)
11	Источник тока (+) 20 мА
12	PL [1])
13	Источник тока (+) 20 мА
14	PL [1])
15	PL [1])

PL [1]) = резерв

4.2 Комплектующие

Кодирующий штеккер	C79334—A3011—B12
Передняя заглушка	C79451—A3079—C251

SIEMENS

S5-135U

ПРОГРАММИРУЕМЫЙ КОНТРОЛЛЕР S5-135U, S-процессор

Руководство по программированию Заказной No.C79000-B8500-C264-03

Оглавление **Страница**

1.	Пояснения	4
1.1	Область применения	4
1.2	Язык программирования STEP 5	4
1.3	Программирование	5
1.3.1	Структура программы	5
1.3.2	Организация программы	8
1.3.3	Хранение программы	9
1.3.4	Обработка программы	9
1.4	Программирование в многопроцессорном режиме	10
1.4.1	Маркеры межпроцессорного обмена	11
1.4.2	Структуризация программы	11
1.4.3	Распределение периферии и маркеров связи	12
1.5	Общие указания	15
1.5.1	Оптимизация программы пользователя (по времени)	15
2.	**Программные блоки**	**17**
2.1.	Программирование программным блоков	17
2.2	Вызов программных блоков	18
3	**Блоки данных**	**19**
3.1	Программирование блоков данных	19
3.2	Вызов блоков данных	20
4	**Функциональные блоки**	**21**
4.1	Общие сведения	21
4.2	Структура функциональных блоков	22
4.3	Вызов и параметрирование функциональных блоков	23
4.4	Программирование функциональных блоков	25

5 Организационные блоки 30
5.1 Общие сведения 30
5.2 Эксплуатация 32
5.2.1 Состояние останова 34
5.2.2 Общий сброс 35
5.2.3 Тестовый режим 36
5.3 Программирование параметров запуска 36
5.4 Программирование циклической обработки 39
5.5 Программирование обработки прерываний от процесса 43
5.6 Программирование обработки прерываний по времени 45
5.7 Обработка аппаратных и программных ошибок 46

6 Список команд STEP-5 с примерами программирования 53
6.1 Основные правила 53
6.2 Основной набор операций 59
6.3 Дополнительный набор операций 83

7 Спец. функции операционной системы 97
7.1 Передача блоков данных (ОВ 255) 97
7.2 PID-регулятор (ОВ 250 и 0В 251) 97
7.2.1 PID-алгоритм 100
7.2.2 Блок данных для PID регулятора 101
7.2.3 Инициализация и вызов PID регулятора 105
7.2.4 Формат входов/выходов регулятора 106
7.2.5 Общие указания 107
7.2.6 Характеристики регулятора 108
7.2.7 Двоично/десятичные дроби 109
7.2.8 Сокращения 110
7.3 Регистр сдвига 111
7.3.1 Принцип действия 111
7.3.2 Программирование регистра сдвига в программе пользователя (ОВ241-ОВ248) 114
7.3.3 Деблокировка области памяти регистра сдвига (ОБ221) 116
7.4 Включение времени сканирования (ОВ222) 116
7.5 Расширение разрядности чисел с фиксированной запятой с 16 до 32 (ОВ220.). 116
7.6 Сравнение видов запусков в многопроцессорном режиме 116
7.7 Считывание контрольной суммы операционной системы EPROM (ОВ257) 116
7.8 Передача блоков маркеров связи (ОВ224) 117
7.9 Параметрирование характеристик запуска 117
8 Специальные команды языка STEP 5 118
8.1 Создание блока данных 118
8.2 Установка/сброс семафора 118

9. Обзор операций команд языка STEP 5 119
10 Информация о неисправностях 122

Приложение 127

Эта инструкция по программированию описывает сферу применения S-процессора 6ES5 921-3UA11, версия 16 и 6ES5 921-3UA12, версия 4.

При работе с более ранними версиями некоторые Функции могут быть ограничены.

Сокращения

AG	Программируемый контроллер (ПК)
АКК 1(2)-L(H)	Аккумулятор 1(2) младший (старший) байт
AWL	Список команд
BARB	Контроль выполнения программы
BARBEND	Конец контроля выполнения программы
BCD	Двоично-десятичный код
CP	Коммуникационный процессор
CPU	Центральный процессор (ЦПУ)
DB	Блок данных
DX	Дополнительный блок данных
FB	Функциональный блок
FX	Дополнительный Функциональный блок
FUP	Функциональная схема
IP	"Интеллигентные" модули периферии
КОР	Релейно-контактная схема
KOR	Координатор
0B	Организационный блок
PA	Отображение процесса
PAA	Область отображения выходов
PAE	Область отображения входов
PG	Программатор
SB	Шаговый блок
USTACK	Стек прерываний
VKE	Результат логической операции

дополнительная литература

Следующие руководства содержат введение в программирование на языке STEP 5 и документацию по использованию стандартных функциональных блоков:

Программирование логического управления на STEP 5

Том 1, программирование основных функций
SIEMENS AG, ISBN 3-8009-1407-7
Том 2, использование стандартные функциональных блоков
SIEMENS AG, ISBN 389009-1373-9
Том 3, программирование функциональных блоков
SIEMENS AG, ISBN 389009-1366-6

1. Пояснения

1.1 Область применения

Программируемый контроллер S5-135U представляет собой высокоэффективное многопроцессорное устройство, предназначенное для автоматизации процессов (разомкнутого и замкнутого циклов управления, сигнализации, контроля, регистрации). Его можно использовать как для простейшего логического управления, так и для решения сложных задач автоматизации. Для его программирования используется язык STEP 5.
Центральное устройство контроллера (в дальнейшем центральное устройство) может выть оснащен следующими модулями;

— одним модулем центрального процессора (CPU) для однопроцессорного режима работы или
— одним координатором (KOR) и несколькими модулями CPU (в количестве до 4) для многопроцессорной работы, а также
— процессорами связи или коммуникационными процессорами (CP) (количеством до 8) для однопроцессорного режима.

Оставшиеся свободными места могут быть заполнены модулями вводов/выходов. В соответствии с конфигурацией периферии к центральному устройству может быть подключено устройство расширения.

В многопроцессорном контроллере каждый процессор обрабатывает свою часть программы пользователя, хранящуюся в его модуле памяти, независимо от других процессоров. Координатор управляет передачей данных по шине S5. Для этих целей в координаторе создается область маркеров межпроцессорного обмена данными (см. Раздел 1.4.1).

1.2 Язык программирования STEP 5

Использование языка STEP 5 делает возможным программирование целого ряда Функций — от двоичной логики до основных арифметических операций и сложных числовых расчетов.

Программа может выть .представлена тремя способами в виде релейно-контактной схемы (КОР), в виде Функциональной схемы (FUP) и в виде списка команд языка STEP-5 (AWL). Выбор способа записи программы определяется конкретной задачей. Машинные коды команд для всех трех способов одинаковы. Используя определенные правила совместимости видов представления, с помощью программатора можно перевести программу из одного вида представления в другой.

Команды расширенных операций могут быть запрограммированы только в Функциональных блоках и представлены только в виде AWL.

1.3 Программирование

1.3.1 Структура программы

Потная программа контроллера состоит из системной программы и программы пользователя. Системная программа содержит инструкции и установки для внутренних Функций (например, хранение данных в случае сбоя питания, сообщения и т.п.). Эта программа — неотъемлемая часть контроллера (содержится в EPROMe) и не может быть изменена пользователем.

Программа пользователя состоит из команд и указаний, запрограммированных пользователем для обработки и выдачи сигналов, необходимых для управления процессом. S5—135U позволяет осуществлять структурное программирование, т.е. полная программа делится на отдельные программные блоки, Такой метод дает пользователю следующие преимущества:
— упрощается программирование даже сложных программ,
— программные блоки могут быть стандартизированы,
— упрощается организации программ,
— облегчается модификация программ,
— облегчается тестирование программы,
— упрощается запуск в эксплуатацию.

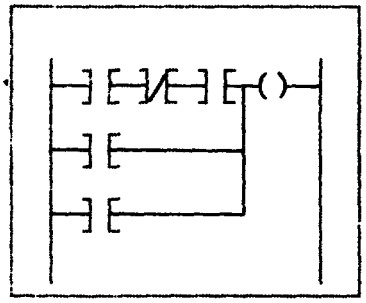
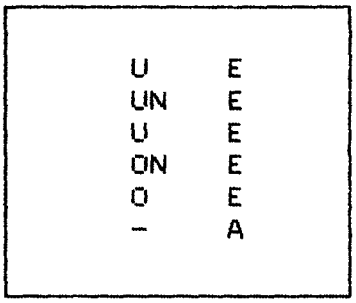
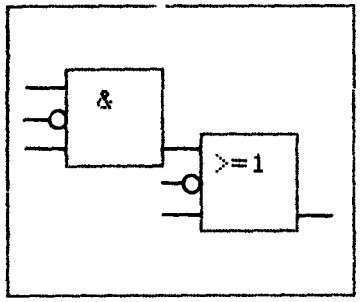
Релейно-конт. схема	Список команд	Функциональная схема
Программируется с помощью графических символов, как схема прохождения тока.	Программируется с помощью мнемонических сокращений функциональных операций.	Программируется с помощью графических символов.
Соответствует DIN 19239 (проект)	Соответствует DIN 19239 (проект)	Соответствует IEC 117-15 DIN 40700 DIN 40719 DIN 19239 (проект)
KOP	AWL	FUP
		

Рис. 1 Виды представления языка программирования STEP 5.

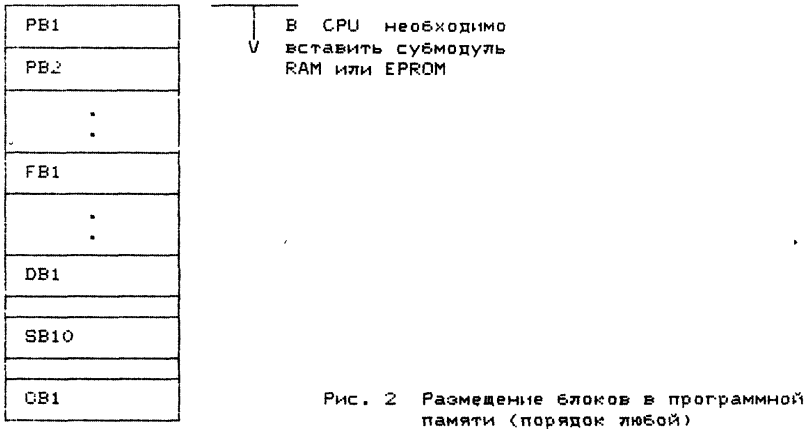


Рис. 2 Размещение блоков в программной памяти (порядок любой)

Для составления программы можно использовать различные., типы программных блоков (каждый для своего круга задач):

- Организационные блоки (OB)

Они обеспечивают связь между системной программой и программой пользователя. Есть организационные блоки, которые могут программироваться пользователем. Они программируются в некоторых ситуациях, когда должна учитываться реакция пользователя. Есть также организационные блоки, которые недоступны пользователю, они содержат системные программы специальных Функций, и их можно лишь вызывать (см. Раздел 7)

- Программные блоки (PB)

Программные блоки PB содержат части программы пользователя, отражающие отдельные технологические операции.

- Функциональные блоки (FB)

Функциональные блоки FB содержат программы часто используемых или наиболее сложных Функций (например, внутреннего контроля, сигнализации, арифметических или функций контроля замкнутого цикла). Исключение; FBO (см. Раздел 1.5.1).

- Шаговые блоки (SB)

Шаговые блоки используются для программирования каскадов или цепочек процесса.

- Блоки данных (DB)

Блоки данных используются для хранения данных или текстов. Функции блоков данных коренным образом отличаются от других блоков, т.к. не содержат программу пользователя. DBO и DB1 зарезервированы для специфических целей <см. Раздел 3).

Можно запрограммировать максимум 256 программных, Функциональных и шаговых блоков, 254 блока данных и 39 организационных блоков. Каждый блок может занимать в программной памяти максимум 4096 слов. В случае приема/передачи блоков с помощью PG необходимо принимать во внимание объем памяти PG.

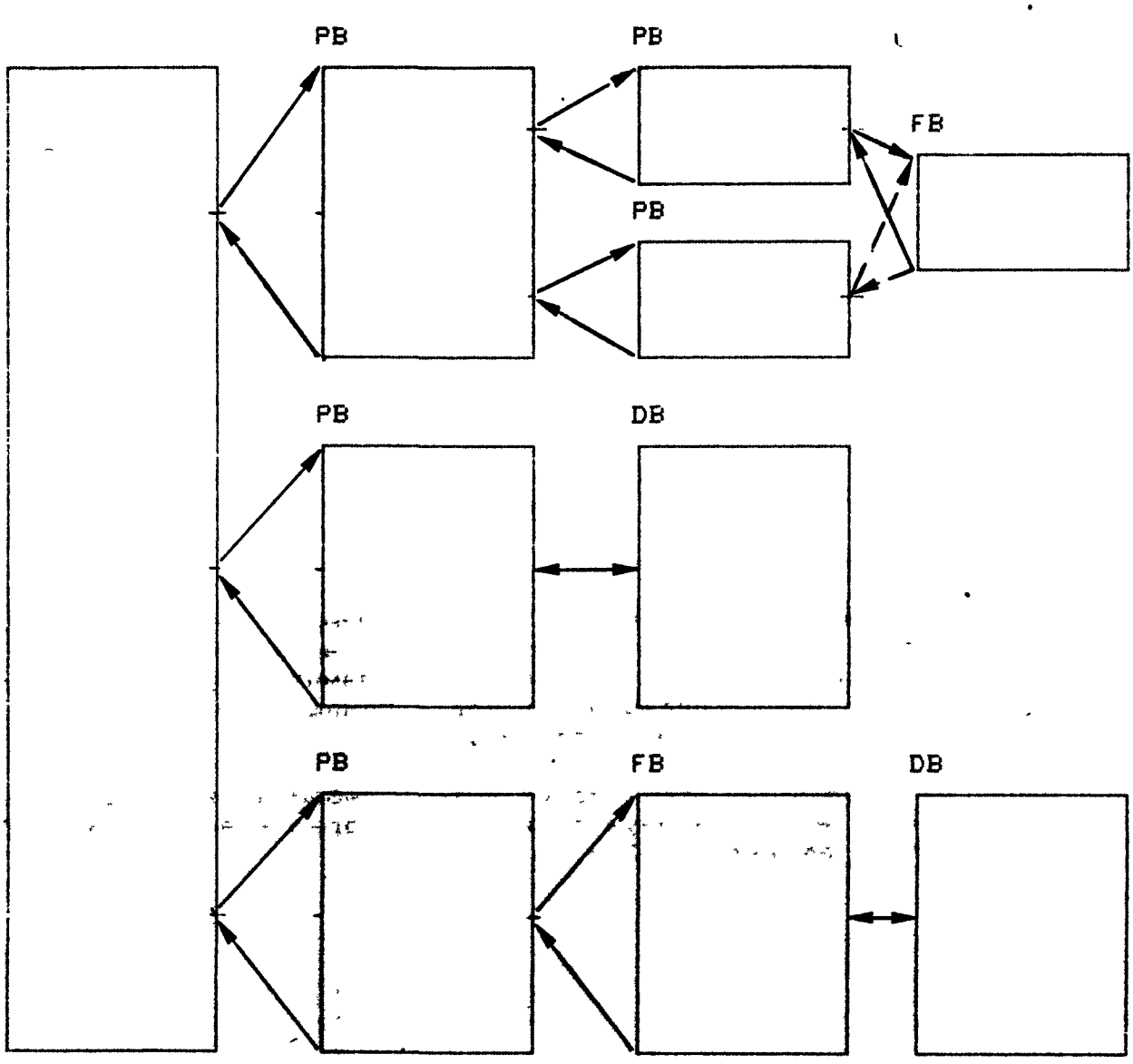
Все запрограммированные блоки могут быть в любом порядке занесены программатором в программную память, которая в виде субмодуля RAM или EPROM вставляется в CPU (Рис. 2). .

1.3.2 Организация программы

Организация программы зависит от того, как и в какой последовательности должны обрабатываться блоки программы пользователя см Рис.3). Переходы (условные и безусловные) к тем или иным блокам программируются в организационным блоках.

Дополнительные программные, Функциональные и шаговые блоки могут быть вызваны определенным сочетанием организационных, программных, Функциональных и шаговых блоков.

Максимальная глубина вложений - 24 уровня. Эту величину можно рассматривать как результирующую глубину обращения, с учетом всех возможных операций (циклических, обработки прерываний, временных функций и т.п.(см. Разделы 5.4-5.7).



OB - организационный блок
PB - программный блок
FB - Функциональный блок
DB - блок данных

Рис. 3 Организация программы STEP 5 8

1.3.3 Хранение программы

Если в CPU используется субмодуль RAM, программу пользователя можно непосредственно передать из PG в CPU. Запрограммированные блоки могут содержаться в памяти в любом порядке. Когда субмодуль RAM полон, следующие блоки данных помещаются в область памяти RAM CPU (RAM для блоков данных см. Рис.14). Объем RAM CPU - 3792 слова. Но при использовании регистров сдвига этот объем сокращается на 128 слов; по мере заполнения RAM CPU адресация смещается в сторону уменьшения.

Если используется блок EPROM, то) все запрограммированные блоки должны быть занесены в него. Блоки данных, которые содержат изменяющиеся в процессе выполнения программы данные, -должны быть скопированы с субмодуля EPROM в область, RAM CPU во время "нового пуска" (см. Раздел 7.1). Исключение "составляют блоки-данных DBO и DB1, которые задействованы в системной программе.

1.3.4 Обработка программы

Обработка программы пользователя может идти по одному из трех путей (Рис. 4):

o Циклическая работа программы (см. Раздел 5.4)

При циклической отработке программы задействован либо организационный блок OB1, либо Функциональный блок FB0:

— OB1 выполняется циклически, вызывая блоки программы пользователя
— FB0 Функционирует также, как OB1, но, в дополнение к этому, он дает возможность пользоваться некоторыми дополнительными операциями. FB0 больше всего подходит для выполнения небольших, ограниченных по времени программ, которые не требуют структурного программирования, а также не содержат обращения к блокам.

Если запрограммированы и OB1 и FB0, то работать будет лишь OB1.

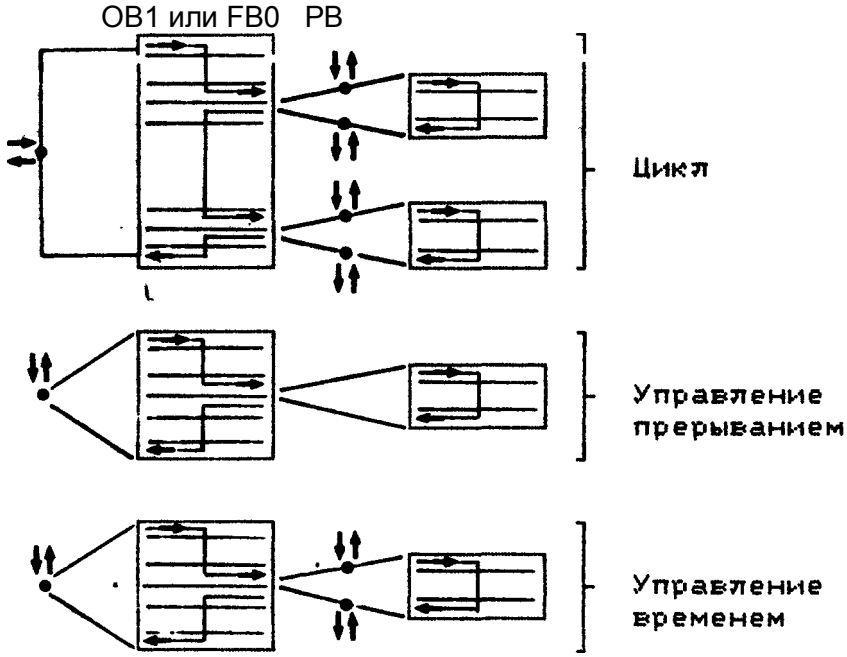
o Обработка программы прерывания (см. Раздел 5.5)

В таком режиме работы циклическая обработка программы прерывается по инициативе внешних устройств. Для вызова стандартных программ прерывания служит OB2.

o Обработка программы, управляемой по времени

В этом режиме определенные участии программы (вызываемые OB13.) автоматически включаются в циклическую программу используя временную установку (см. Рис.4).

Такой режим необходим, когда решаются задачи управления с замкнутым циклом.



Точка прерывания, в которой прекращается обработка циклической программы, предыдущей программы прерывания или программы, управляемой по времени, и начинается выполнение актуальной программы прерывания или программы, управляемой по времени. Программа, управляемая по времени, может быть прервана только программой обработки прерывания, а программа обработки прерывания - только возникшей неисправностью.

Рис. 4 Типы выполняемых программ

1.4 Программирование в многопроцессорном режиме

Программирование контроллера для многопроцессорного режима сводится к операции программирования каждого процессора, описанной в Разделе 1.3. В дополнение к этому отметим некоторые особенности многопроцессорного режима:
— Процессоры могут обмениваться данными через маркеры межпроцессорного обмена, организованные в координаторе.
— Вся программа контроллера может быть разбита на индивидуальные программы для каждого CPU в соответствии с общей задачей.
— Каждому CPU должны быть назначены свои периферийные устройства ввода/вывода.
— Доступом каждого CPU к вине контроллера управляет координатор. Количество работающих CPU устанавливается при помощи перемычек в координаторе (см. Руководство по координатору).

Если процессоров больше одного, то использование координатора обязательно. Если координатор установлен, системная программа CPU работает в многопроцессорном режиме, даже если реально работает всего один CPU. После этого надо запрограммировать в каждом CPU блок OB1.

1.4.1 Маркеры межпроцессорного обмена

Маркеры межпроцессорного обмена (маркеры связи) представляют собой байты, которые назначаются пользователем в CPU как входные и выходные. Они предназначены для побайтного циклического обмена данными между CPU.

Байт маркеров связи определяется как выходной для CPU, который передает данные в циклическом режиме посредством координатора и как входной (с тем же номером) для CPU, принимающего данные.

При работе с маркерами связи необходимо соблюдать следующие правила.

- Маркер, назначенный для одного или нескольких CPU, как входной, должен быть определен для Другого CPU, как выходной.
- Если в одном CPU, маркер назначен в качестве выходного, то он не может быть определен, как выходной для других CPU <только как входной).
- Маркерные байты, определенные, как маркеры связи в каком-либо CPU, пригодны в этом CPU лишь для передачи данных. Остальные маркеры, не определенные, как маркеры связи, могут использоваться по своему обычному назначению.

Назначение маркеров в качестве маркеров связи в блоке данных DB1 описано в Разделе 1.4.3.

Маркеры связи можно использовать также для передачи данных между CPU и CP (коммуникационными процессорами). Это возможно как в однопроцессорном, так и в многопроцессорном режиме. Область маркеров связи, имеющая максимальный объем 256 входных и выгодных байтов, может быть разбита в координаторе и/или CP на 32— байтные области (.см. Инструкцию по эксплуатации координатора и коммуникационного процессора). Все маркеры, указанные в DB1, должны быть установлены в координаторе и получить подтверждение, без -этого системная программа укажет ошибку QVZ (см. Раздел 5.7).

1.4.2 Структуризация программы

Процессоры (максимум до 4) в многопроцессорном режиме выполняют каждый свою программу независимо от других, что дает возможность пользователю разделить всю программу контроллера на индивидуальные программы для каждого процессора. Таким образом, многопроцессорный режим дает следующие преимущества:

- Разделение программы между процессорами, которые затем работают параллельно, сокращает время выполнения всей программы.

- Программы с малым временем выполнения для обработки быстрых процессов могут быть помещены каждая в своем CPU. В этом случае, время обработки программы пользователя в дальнейшем может быть сокращено за счет использования FBO вместо OB1, а также за счет точного определения длины блока таймеров (см. Раздел 1.5.1).

— Программы пользователя с большим временем выполнения для обработки процессов, которые некритичны ко времени, могут быть запрограммированы в своих "медленных" CPU отдельно от "быстрых.

— Каждый CPU может быть ориентирован на свою часть станка в зависимости от ее Функции.

1.4.3 Распределение периферии и маркеров связи

В многопроцессорном режиме пользователь должен распределить (назначить) периферийные модули входов/выходов и, побайтно, необходимые маркеры связи для каждого CPU. Для этой цели существует блок данных DB1, в который пользователь заносит распределение входов/выходов и маркеров в виде таблицы адресов, используемых в PG. При однопроцессорной работе DB1 можно запрограммировать таким образом, чтобы время выполнения программы было оптимизировано. DB1 имеет строго определенную Функцию и не может использоваться для других целей.

o Структура DB1

DB1 создается при помощи PG. Существуют 2 способа создания DB1;

а.) Слова данных 0, 1 и 2 должны быть установлены соответственно;

КН = 4D41, 534B, 3031.

После третьего слова идут списки адресов. Каждый адресный список начинается с ключевого слова. Ключевыми словами являются:

Для цифровых входов	КН = DEOO
Для цифровых выходов	КН = DAOO
Для входных маркеров	КН = CEOO
Для выходных маркеров	КН = CAOO
Для длины блока таймеров	КН = ВВСЮ

Следом за ключевым словом идут в формате данных с фиксированной запятой адреса установленных входов/выходов и маркеров, принадлежащие соответствующему списку адресов. Порядок записей в списке адресов произвольный, так же как и самих списков.
Последним в DB1 должно быть записано слово
КН=EE00,
как символ окончания

В многопроцессорном режиме DB1 Должен быть организован для каждого CPU.

Пример (DB1, ввод с предустановками):

KH = 4D41
KH = 534B Предустановки
KH = 3031

KH = DEOO
KF = +00000
KF = +00002
KF = +00007 Цифровые входы
KF = +00012
KF = +00126
KF = +00127

KH = DAOO
KF = +00001 Цифровые выходы
KF = +00003

KH = CAOO Выходные маркеры
KF = +00005

KH = CEOO Входные маркеры
KF = +00126

KH = EEOO Идентификатор окончания

Ь) Начиная с версий матобеспечения PG 675 SO A03 или SI A01 (и более поздних можно, используя функциональные клавиши F1 и F3, работать с экранной формой записи DB1. В этом случае необходимо занести в экранную форму (таблицу) адреса соответствующих байтов.

Пример (DB1, ввод экранной Формы):

НАЗНАЧЕНИЕ ПЕРИФЕРИИ:

ЦИФРОВЫЕ ВХОДЫ	,	2,	7,	12,	126,	127,	,
ЦИФРОБЫЕ ВЫХОДЫ	,	1,	3,	,	,	,	,
ВХОДНЫЕ МЕТКИ СВЯЗИ	,	126,	,	,	,	,	,
ВЫХОДНЫЕ МЕТКИ СВЯЗИ	,	5,	,	,	,	,	,
ДЛИНА БЛОКА ТАЙМЕРОВ	,	,	,	,	,	,	,

o Ввод/изменение DB1 возможно:

- Посредством PG, когда CPU в состоянии "СТОП" и если контроллер оснащен памятью пользователя RAM. DB1, который вводится или изменяется, может быть воспринят системной программой только в виде встроенного списка адресов во время холодного пуска (см. Раздел 5.3).

- Посредством программирования модуля EPROM.

o Список адресов входов/выходов для циклической работы

Во время выполнения цикла программы в памяти периодически регенерируется отображение процесса, но обновляются лишь те цифровые входы/выходы, которые есть в списке адресов. Во время каждого возобновления цикла системная программа проверяет, соответствуют эти входы/выходы входным/выходным байтам, занесенным в DB1. Если нет, то CPU выходит из цикла и выдает сообщение об ошибке DB1.

В многопроцессорном режиме непосредственный доступ к входам/выходам через область отображения процесса (команды загрузки/пересылки L PB, T PB, L PW, T PW см. в. Разделе 6.2) возможен:

- для всех цифровых входов контроллера S5-135U независимо от того, занесены они в адресный список данного CPU или нет;
- только для тех цифровых выходов, которые включены в адресный список данного CPU;
- однако, в многопроцессорном режиме, чтобы получить непосредственный доступ к входам/выходам, зачастую приходится ждать, пока шина контроллера не освободится (это происходит автоматически независимо от желания оператора).

В однопроцессорном режиме непосредственный доступ ко всем входам/выходам возможен независимо от списка адресов.

доступ к области отображения процесса посредством операций загрузки/пересылки L E.., T A.. и логических операций разрешается и в однопроцессорном, и в многопроцессорном режиме, но только для тех входов/выходов, которые указаны в DB1.

Примечание! Любой выходной байт может быть доступен только одному CPU.

o Адресный список маркеров связи

Маркерные байты, указанные в адресных списках, считываются (входные маркеры) и записываются (выходные маркеры), пока контроллер находится в циклической работе. При назначении маркеров каждому отдельному CPU должны соблюдаться правила, приведенные в Разделе 1.4.1.

1.5 Общие указания

Блоки DB0 и DB1 всегда заняты данными системной программы или установочными параметрами CPU.

1.5.1 Оптимизация программы пользователя по времени

o Структура программы

В однопроцессорном режиме, так же как и *в* многопроцессорном, время обработки программы пользователя можно уменьшить, если просто применить структурное, программирование. Поскольку каждое изменение блока требует дополнительного времени, структурного программирования следует .избегать при создании коротких, критичных по времени программ и использовать только FB0. В FB0 можно использовать весь набор команд контроллера S5—135U (см. Раздел 6).

o Время цикла

Время выполнения программы пользователя складывается из периодов обработки всех вызываемых блоков. Если программа обращается к какому-то программному блоку N раз, то время обработки этого блока N раз войдет в суммарное время обработки программы.

Сумма периодов обработки всех частей программы пользователя, включая время выполнения программ, управляемых прерываниями и время выполнения программ, управляемых временем, — это и есть время цикла. В контроллере S5—135U оно ограничено 100 мс. Время цикла контролируется системной программой, и если оно превышается, то CPU останавливается, и выдается сообщение об ошибке "ZYK" ("см. Раздел 5.7).

Вызовом специальной функции системной программы "Изменение времени цикла" (см. Раздел 7.4) пользователь может изменить время цикла еще на 100 мс, считая с момента вызова этой функции.

o Назначение (установка) входов/выходов

И в однопроцессорном, и в многопроцессорном режимах очень важно, чтобы в списке адресов входов/выходов и маркеров связи были указаны только те адреса, к которым имеет доступ программа данного CPU.

Наличие в адресном списке адресов входов/выходов и маркеров связи, которые не используются программой, но периодически опрашиваются в соответствии с DB1, неоправданно увеличивает время обработки программы пользователя.

o Длина таймерного набора

В DB1 пользователь может указать также количество таймерных позиций, входящих в таймерный набор. В этом случае время обработки таймеров, не вошедшим в таймерный набор, будет сэкономлено.

Однако, это возможно лишь в том случае, если количество таймероав, задействованных пользователем, меньше числа, указанного в длине таймерного набора. Если длина таймерного набора = О, то таймеры вообще не будут обрабатываться. Если не указывать длину таймерного набора, тогда будут доступны все таймеры. Если обрабатываемое программой количество таймеров больше указанной длины таймерного набора, то процессор останавливается и выдает сигнал об ошибке.

В однопроцессорном режиме также можно указать длину таймерного набора. Но в этом случае пользователь должен записывать в DB1 полный адресный список, т.е. указывать и адресный список входов/выходов (см. Раздел 1.4.3).

Ввод длины блока таймеров в DB1 (пример для 40 таймерных позиций с разрешенным количеством от О до 39):

-
-

KH = BBOO Идентификатор
KF = +00040 Длина таймерного набора

-
-

KH = EEOO Идентификатор окончания

2. Программные блоки

2.1 Программирование программных блоков

Следующее описание подходит для программирования организационных, программных и шаговых блоков. Эти три типа блоков в программировании почти не отличаются. Они могут программироваться в виде релейно-контактной схемы (КОР), Функциональной схемы (FUP) и в виде списка команд (AWL) языка STEP 5. Программирование начинается с присвоения номера блоку;

Программные блоки	0...255	
Шаговые блоки	0...255	
Организационные блок	1...39	(см. Раздел 5)

Затем идет непосредственно программа управления, которая завершается командой "BE". Для программирования может использоваться только набор функций языка STEP 5.

Программный блок состоит из тела блока (собственно программы на языке STEP 5) и заголовка блока. Заголовок блока автоматически записывается программатором в начало блока. Он занимает 5 слов в памяти пользователя.

Программный блок обязательно должен содержать законченную программу. Попытка выйти за границу блока, чтобы продолжить программу, воспринимается как ошибка.

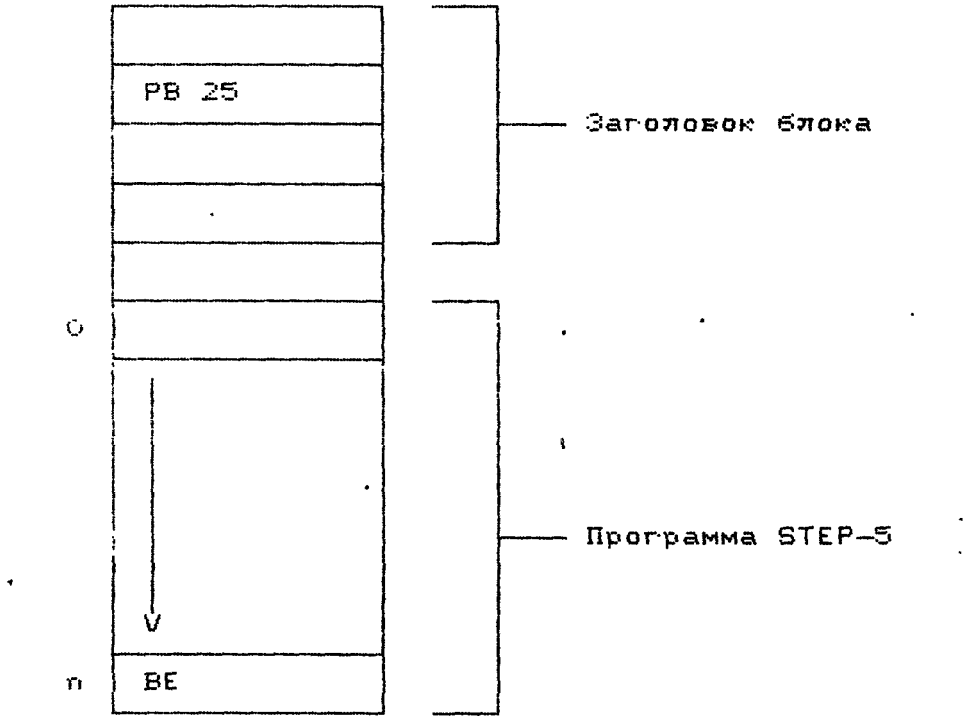


Рис. 5 Структура организационного, программного и шагового блоков

2.2 Вызов программных блоков

Программные блоки активируются с помощью операций вызова (Рис.6.). Эти вызовы можно запрограммировать в любом программном, функциональном или шаговом блоке. Они напоминают команды перехода к подпрограмме и могут быть безусловными (SPA PBX) или условными (SPB PBX).

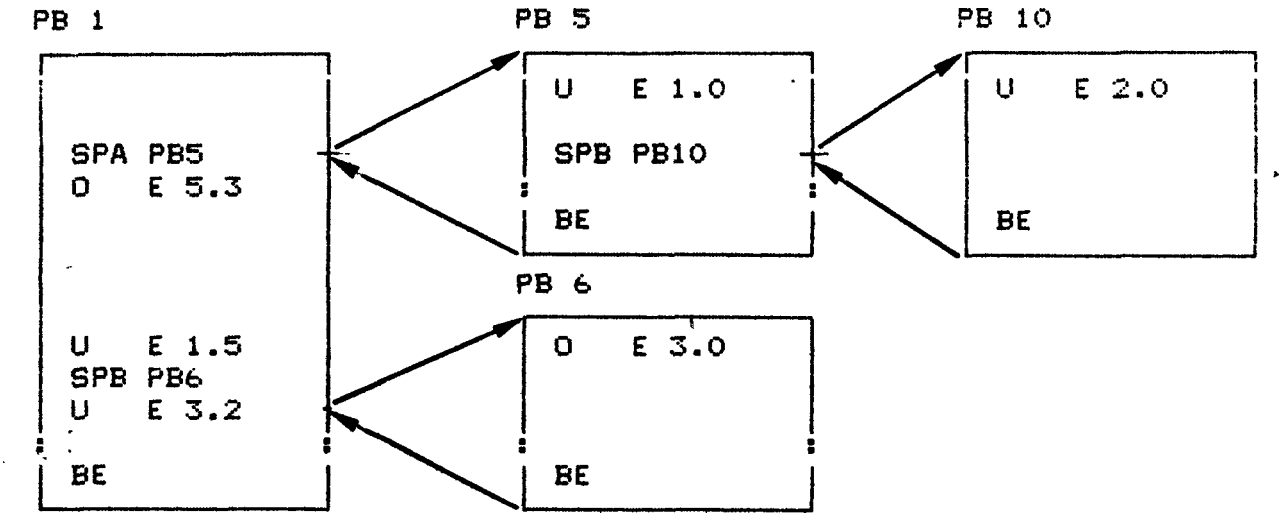


Рис.6 Вызовы блоков, которые выполняют обработку программных блоков.

После команды "BE" (конец блока) происходит возврат в блок, из которого произведен вызов, и обрабатывается команда STEP 5, следующая за командой вызова блока. При этом после команд вызова блока или "BE" невозможно использовать результат логической операции (VKE) для последующих операций, т.к. это команды, ограничивающие VKE (VKE: см. Раздел 6.1). Однако VKE тоже переходит в "новый" блок и там может быть использован.

о Безусловный вызов: SPA xx

Программный блок вызывается независимо от результата предыдущей логической операции (VKE).

о Условный вызов: SPB xx

Вызов ПРОГРАММНОГО блока зависит от результата предыдущей логической операции. Если VKE=1, команда перехода выполняется, если VKE=0 - нет. В обоих случаях команда перекопа устанавливает VKE в 1. Такая же зависимость от VKE для команды условного конца блока BEB.

3. Блоки данных

3.1 Программирование блоков данных

В блоках данных хранятся данные, необходимые для обработки программы пользователя. В блоках данных не может быть команд языка STEP-5. Данными могут быть:
- последовательность битов;
- числа (десятичные, двоичные), например, для таймеров, результаты арифметических операций;
- буквенно-цифровые значения, например, для текстовых сообщений.

Блоки данных имеют такую же структуру как и программные. Программирование начинается с указания номера блока данных от 2 до 255, например, DB25. Каждый блок данных может состоять максимум из 256 16-битовых слов. Данные должны вводиться в блок в порядке возрастания нумерации слов данных, начиная с нулевого слова DWO. Блоки данных DBO и DB1 резервируются для специальных Функций и для пользователя недоступны.

Одно слово в памяти резервируется для слова данных в памяти программ. Следующие пять слов в памяти программ занимает заголовок блока, создаваемый программатором для каждого блока данных. Блок данных может занимать максимально 4096 слов в программной памяти CPU. При передаче и приеме данных с программатором следует учитывать его объем памяти.

Внимание: С помощью команд загрузки/передачи L/T DW, можно обращаться к слову данных с номерами до 255.

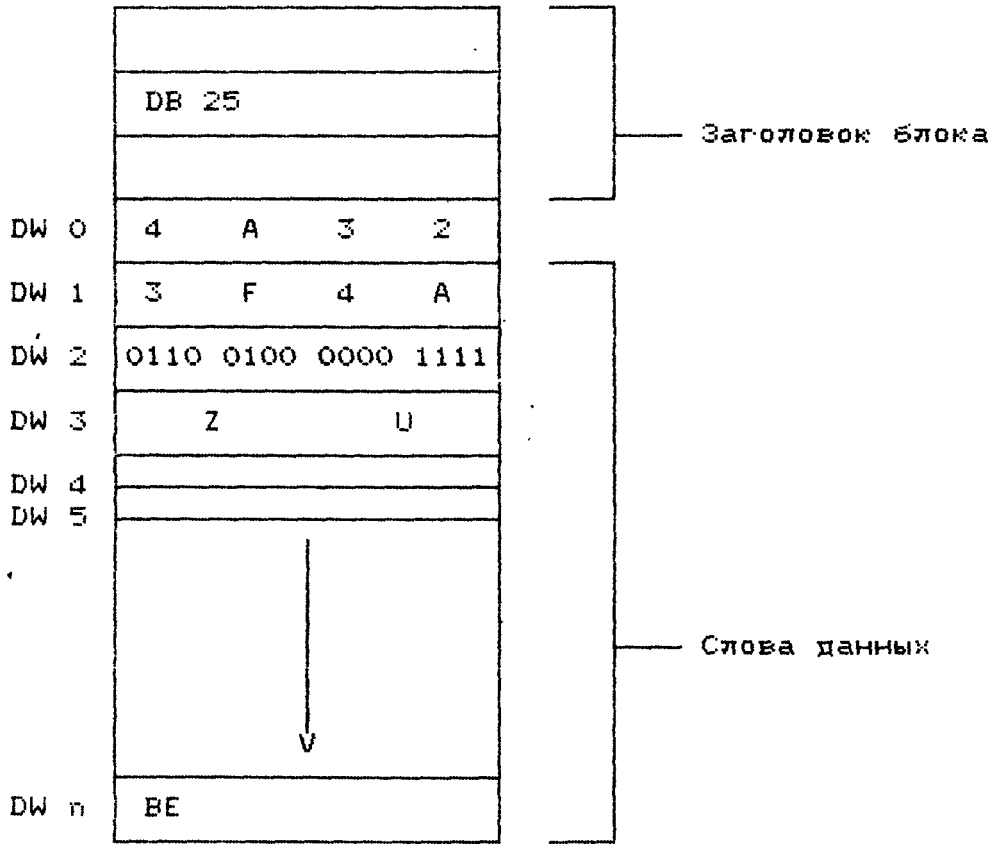


Рис.7 Структура блока данных

Если программный блок, в котором уже был адресован блок данных, вызывает другой программный блок, а в этом блоке адресован другой блок данных, то последний блок данных действителен лишь в вызванном программном блоке. После возвращения в вызвавший программный блок вновь начинает действовать предыдущий блок данных (Рис.9).

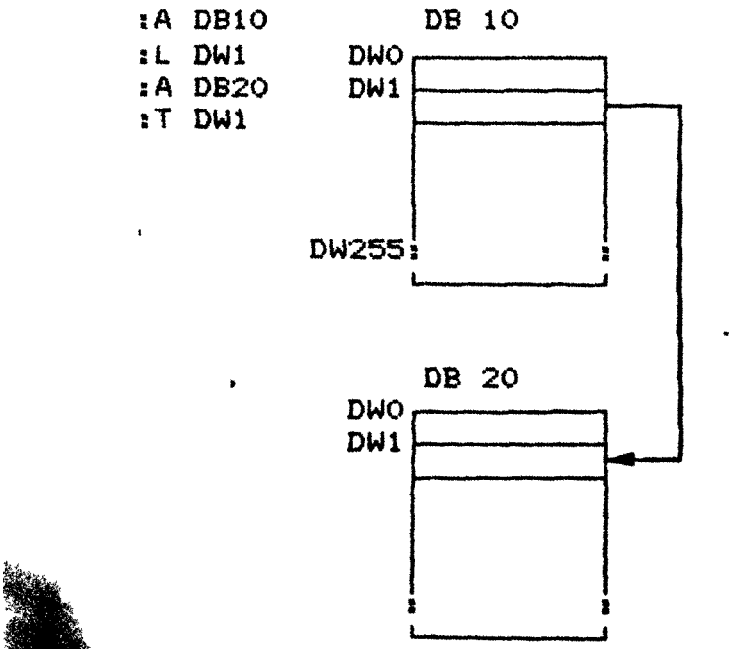


Рис. 8 Адресаций блоков данных

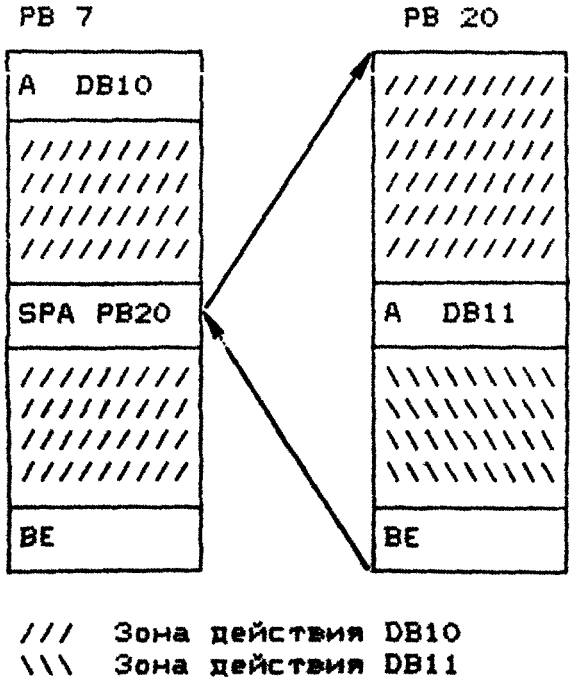


Рис 9 Зона действия выбранных блоков ванных

Пример 2 (см. Рис. 9):

В программном блоке PB7 вызван блок данных DB10. В дальнейшем обрабатываются данные этого блока.

Следующим для обработки вызывается PB20. Однако, DB10 по-прежнему является действительным. Только после вызова DB11 меняется область данных. DB11 действует до конца PB20.

После перехода назад в PB7 опять становится действительным DB10.

4. Функциональные блоки

4.1 Общие сведения

Функциональные блоки являются такими же частями программы, как например, программные блоки. Однако они имеют 4 существенных отличия:

- Функциональные блоки могут быть параметрированы, т.е. могут быть заданы фактические параметры, с которыми работает <FB>

- В противоположность организационным, программным и шаговым блокам, для программирования Функциональных блоков используется более широкий набор команд.

- Программа Функционального блока может быть представлена только в виде списка команд AWL.

- Графически Функциональный блок выглядит как "черный ящик".

Внутри программы пользователя Функциональные блоки представляют сложную, законченную Функцию. Они могут храниться в операционной системе или на гибких дисках, которые Ф.SIEMENS поставляет в качестве стандартного матобеспечения ("стандартные функциональные блоки на мини-флопи-дисках"), а также могут быть созданы самим пользователем, дополнительные операции к основным могут быть запрограммированы только в функциональных блоках.

4.2 Структура функциональных блоков



Рис. 10 Структура Функционального блока

о Заголовок блока

Заголовок блока содержит все данные, которые необходимы для программатора, чтобы представить FB и чтобы проконтролировать операнды при параметрировании FB. Перед программированием FB заголовок вводится пользователем с помощью программатора. Он хранится в памяти CPU и содержит команды перехода, которые выполняются при вызове Функционального блока, но не отображаются при его чтении (переход через список формальных операндов).

о "Тело" блока,,

"Тело" блока, содержит собственно программу FB. Выполняемые функции записаны на языке STEP-5 и хранятся в памяти. При вызове FB выполняется ...только "тело" . блока, для программирования FB может использоваться более широкий набор команд по сравнению с программными блоками (см. Раздел 6.3).

4.3 Вызов и параметрирование функциональных блоков

Функциональные блоки могут использоваться для повторяющихся или довольно сложных функций. Они загружаются в память один раз и могут вызываться многократно, причем, параметры блока могут меняться от вызова к вызову. FB записываются в память с определенным номером (от FBI до FB255), так же как программные блоки и блоки данных. Номера FB от 240 до 243 заняты встроенными FB. Вызов FB можно программировать в программном или другом FB. Вызов состоит из команды вызова и списка параметров.

*

o Команды вызова

- Безусловный вызов (SPA FBn):
  Вызванный FB обрабатывается независимо от результата предыдущей логической операции (VKE).

- Условный вызов (SPB FBn):
  Вызванный FB обрабатывается, только если VKE=1. Если VKE=0, то переход не выполняется. В обоих случаях при условном вызове VKE устанавливается в единицу.

При условном и безусловном вызовах VKE можно обрабатывать, но нельзя использовать в последующих логических операциях. VKE переходит в вызываемый блок.

o Список параметров (см. пример)

Список параметров следует непосредственно за командой вызова. В нем определены входные/выходные переменные и данные (см. "Типы параметров блоков".). Список может содержать максимум 40 параметров. Список параметров устанавливает соответствие между формальными и фактическими параметрами (операндами).

При обработке FB вместо формальных параметров используются переменные из списка. Последовательность переменных в списке параметров контролируется программатором.

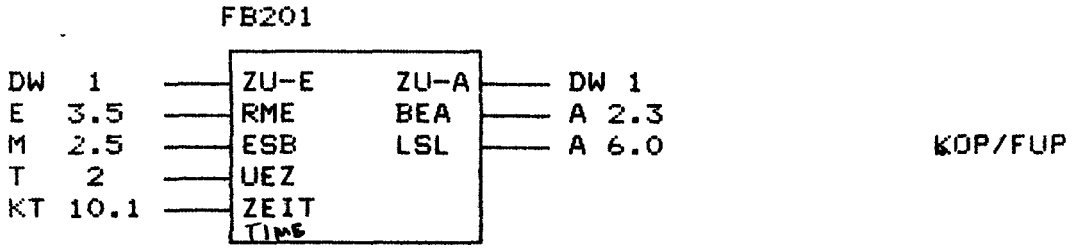
Команда перехода после вызова FB автоматически активируется программатором, но не индицируется на дисплее. Вызов FB, команда перехода и каждый параметр занимают в памяти отдельное слово (исключение - числа с плавающей запятой: они занимают два слова; .

Пример (вызов функционального блока и передача параметров в программном блоке в видах представления программы AWL и KOP/FUP")

Вид представления

```
PB25

      : SPA FB 201
ИМЯ   : E-ANTR
ZU-E:    DW  1
RME  :   E  3.5                                  AWL
ESB  :   M  2.5
UEZ  :   T   2
TIME:    KT 10.1      Список параметров
ZU-A:    DW  1
BEU  :   A  2.3
LSL  :   A  6.0
```



KOP/FUP

Идентификаторы входов/выходов функционального блока, которые появляются на экране программатора в режиме программирования, и имя FB хранятся в самом FB. Поэтому перед программированием на программаторе все необходимые функциональные блоки должны быть перенесены на рабочую дискету или непосредственно в память программатора. (Для получения более детальной информации обратитесь к Руководству по программатору).

4.4 Программирование функциональных блоков

FB создается в два приема в соответствии с его структурой;
Введение заголовка и введение "тела" блока.

Перед введением "тела" блока (на языке STEP 5) вводится заголовок. Заголовок включает;

- библиотечный номер,
- номер FB,
-.формальный операнд (имена параметров блока),
- тип параметров блока,
- тип блока данных.

o Библиотечный номер

В качестве библиотечного номера можно использовать любое число от О до 65535. FB определяется этим номером независимо, от его символьных или абсолютных параметров, достаточно один раз присвоить библиотечный номер, чтобы полностью идентифицировать FB. Стандартные FB имеют строго определенные библиотечные номера.

o Имя функционального блока

Имя FB может иметь до 8 символов. Имя идентифицирует функциональный блок символом.

o Формальный операнд (имя параметра блока)

Формальный операнд может иметь до 4 символов и должен обязательно начинаться с буквы. Каждый FB может программировать до 40 параметров .

Вид параметров блока;

В качестве параметров блока могут быть запрограммированы;

- E - вход
- A - выход
  D - данные
- B - команда вызова блока
- T - таймер
  Z - счетчик

На графическом изображении параметры E, A, D, B, T и Z расположены слева от графического изображения FB, а параметр A - справа.

Тип параметров блока

Для классов параметров E, A и D должен быть указан тип параметров

BI/BY/M/D для классов E и A
KM/KH/KY/KS/KF/KT/KC/KG - для класса D

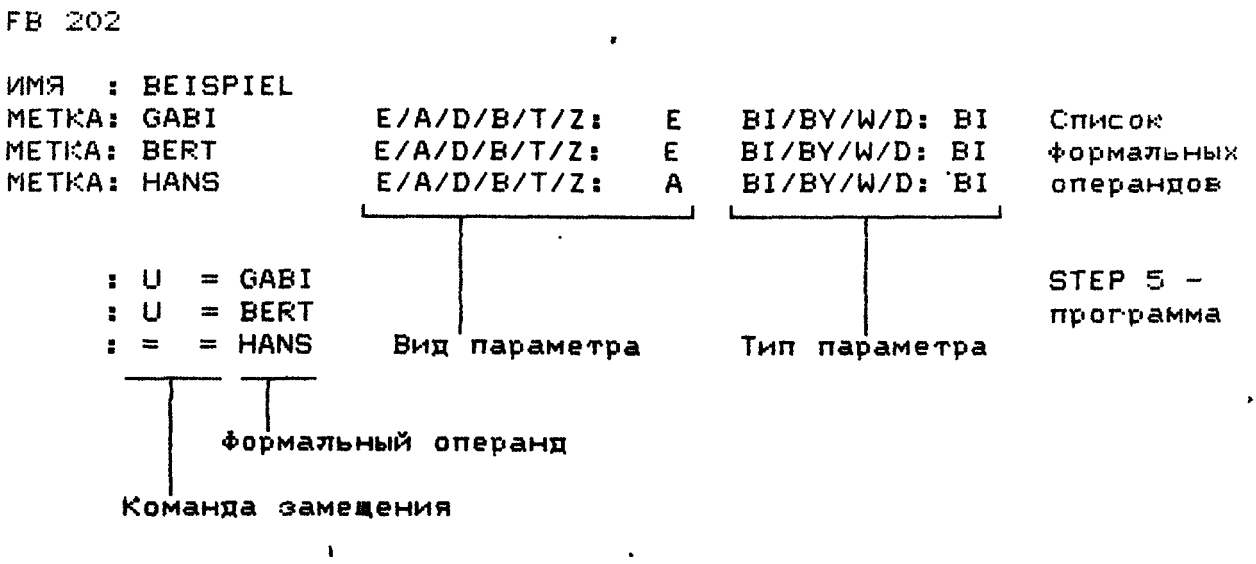
Для параметров E, A и D тип параметра определяет, какой размер -битовый, байтовый, словный или размер двойного слова - используется, и какой формат данных определен для параметра D.

Вид параметра	Тип параметра	Допустимые фактические операнды
E, A	BI - для операндов с битовым адресом	E n.m входы A n.m выходы M n.m маркеры
	BY - для операндов с байтовым адресом	EB n входные байты AB n выходные байты MB n байты маркеров DL n старший байт слова данных
	W - для операндов со словным адресом	DR n младший байт слова данных
	D - для операндов с двухсловным адресом	PB n периферийные байты EW n входные слова AW n выходные слова MW n слова маркеров DW n слова данных PW n периферийные слова ED n двойное слово входов AD n двойное слово выходов MD n двойное слово маркеров DD n двойное слово данных
D	KM - для двоичного числа (16 разрядов) KY - для двух байтовых чисел от 0 до 255 KH - для 16-ричных чисел (макс. 4 позиции) KC – для символов (макс.2 буквенно-цифровых знака	константы

Вия параметра	Тип параметра	Допустимые фактические операнды
D	KT - для времени (BCO—код) в диапазоне 1...999.3 KZ - для счетчиков (BCD) диапазон: O...999 KF - для чисел с Фиксированной запятой в диапазоне от -32766 до +32768 KG - для чисел с плавающей запятой	
D	Указание типа не требуется	DB N - блоки данных, выполняется команда ADB FB N функц. блоки (допустимы только без параметров) вызываются безусловно SPA PB N - прогр. блоки вызываются безусловно SPA SB N - шагов. блоки вызываются безусловно SPA
T	Указание типа не требуется	T0 ...127 Таймер [1)]
Z	Указание типа не требуется	Z0 ...127 Счетчик [1)]

[1)] Значение параметра таймера или счетчика должно быть указано в качестве данным или должно быть запрограммировано в качестве константы в Функциональном блоке.

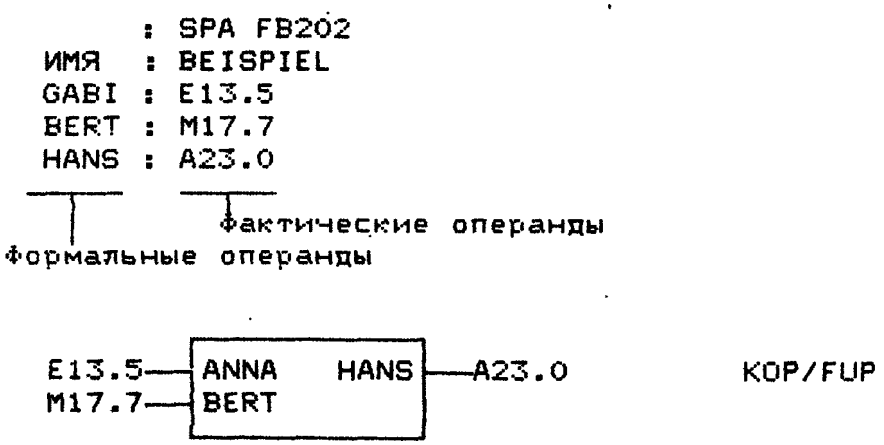
Пример (Программа в Функциональном блоке)



Пример (Вызов Функционального блока из программного блока)

Вид представления

AWL



- Выполняемая программа

:U E 13.5
:U M 17.7
:= A 23.0

Операции (команды замещения), к которым относятся параметры, должны быть запрограммированы в функциональном блоке с формальными параметрами в качестве операндов. Кроме того, к Формальным операндам можно обращаться несколько раз в различный местах функционального блока.

Во время вызова функционального блока формальные операнды замещаются фактическими параметрами из списка параметров.

Внимание! Если порядок или количество формальных операндов в заголовке функционального блока изменяется, то должны быть изменены соответствующие команды и список параметров в блоке, *в* котором происходит вызов данного функционального блок

Пример: (стандартный функциональный блок)

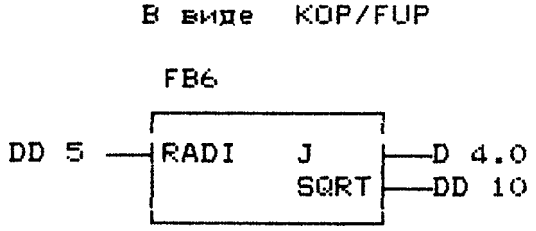
RAD:	FB для 115U
Извлечение корня из числа с плавающей запятой	FB для 135U
GP	FB19 для 150U

Функциональный блок RAD:GP извлекает квадратный корень из числа с плавающей запятой (8-битовая экспонента и 24-битовая мантисса). Результат - также число с плавающей запятой (8-битовая экспонента 24-битовая мантисса), как бы младший бит мантиссы не округлялся.
Если необходимо, функциональный блок для последующих операций устанавливает идентификатор "Корень отрицательный"
Числовые значения
:

Радикал	-0.146.9368 Exp -38	+0.1701412 Exp +39
Корень	+0.3833234 Exp -19	+0.1304384 Exp +20

Функция: Y = ±A
Y = SORT; A=RADI

Вызов функционального блока:

В виде AWL:	В виде KOP/FUP
:SPA FB 6 Имя: RAD:GP RADI: DD 5 J: D 4.0 SORT: DD 10	

В приведенном выше примере извлекается корень из числа с плавающей запятой, которое записано в двойном слове данных DD5 в виде 8-битовой экспоненты и 24-битовой мантиссы. Результат, который тоже является 32-битовым числом с плавающей запятой, записывается в DD10. Перед выполнением функции должен быть вызван соответствующий блок данных. Параметр J (вид параметра M, тип параметра BI) определяет знак результата: J=1 для отрицательного результата. Используемые слова маркеров : с MW238 по MW254.

В каталоге ST57 приведены стандартные Функциональные блоки для S5-135U, время их выполнения, необходимая память и используемые ими переменные.

Общие замечания

Если используется стандартный функциональный блок, то байты маркеров с 200 по 255 резервируются, и пользователь не должен их использовать.

Таймер TO, счетчик ZO и блоки данных DBO и DB1 также зарезервированы.

Стандартные функциональные блоки занимают номера с 1 по 199. Функциональные блоки пользователя могут, поэтому, иметь номера с 200 по 255.

Функциональный блок FBO обрабатывается системой циклически вместо организационного блока OB1, если OB1 не запрограммирован.

5. Организационные блоки

5.1 Общие сведения

Организационные блоки осуществляют связь между системной программой и программой пользователя (см. Рис. 12). Организационные блоки с OB1 по OB39 являются частью программы пользователя, также как и программные, шаговые и функциональные блоки. Организационные блоки вызываются системной программой. Пользователь имеет возможность программировать организационные блоки с OB1 по OB39 и, таким образом, косвенно иметь доступ в системную программу, для тестирования эти организационные блоки могут быть вызваны пользователем (по команде SPA/SPB OBxxx).

Соответствующим программированием организационным блоков могут быть установлены следующие режимы работы:

- Циклическое выполнение программы (OB1 или FBO)
- Обработка прерываний от процесса (OB2(
- Обработка прерываний по времени (OB13)
- Ручной новый пуск без запоминания (OB20)
- Ручной новый пуск с запоминанием (OB21),
- Автоматический новый пуск с запоминанием (OB22)

Обработка сообщений об ошибках:

- Если происходит одна из ошибок, которые описаны в Разделе 5.7, то обрабатывается OB28.

o 0B специального назначения

Помимо организационных блоков с OB1 по OB39, могут быть вызваны специальные функции системной программы с помощью организационных блоков с номерами больше 39. Эти блоки для специальных функций не могут быть запрограммированы пользователем, они могут быть только вызваны. Они не содержат программы STEP-5. Специальные функции отдельно описаны в Разделе 7.

Организационные блоки специальных функций могут вызываться из организационных блоков OB1...0B39 (при использовании версий программного обеспечения программатора PG675, начиная с SO A03 и S1 A01).

o Назначение организационных блоков

Абсолютный параметр	Назначение или причина для инициализации
OB для циклической обработки процесса:	
OB1	Начало программы (также FB0)
OB для обработки прерываний процесса:	
OB2	Прерывание процесса
OB для обработки прерываний по времени:	
OB13	База времени 0.1 сек
OB для нового пуска:	
OB20 OB21 OB22	Ручной новый пуск без запоминания Ручной новый пуск с запоминанием Автоматически новый пуск без запоминания
OB для обработки ошибок;	
OB28	Реакция на ошибку
OB для специальных функций:	
OB220 OB221 OB222 OB223 OB224 OB225 OB226 OB227 OB240 OB241 OB242 OB243 OB244 OB245 QB246 OB247 OB248 OB250 OB251 OB255	Расширение, числа с фиксир. запятой от 16. до 32 бит Стирание всех регистров сдвига из памяти пользователя Триггер времени цикла Стоп, если не все CPU работают во время многопроцессорного режима Передача меток связи в блоки при многопроцессорном режиме работы Установка параметров в начальные значения (запуск) Чтение байта из системной программы в EPROMe Чтение контрольной суммы системной программы в EPROMe Инициализация регистров сдвига Вызов регистра сдвига No.1 в циклической программе Вызов регистра сдвига No.2 в циклической программе Вызов регистра сдвига No 3 в циклической программе Вызов регистра сдвига No 4 в циклической программе Вызов регистра сдвига No.5 в циклической программе Вызов регистра сдвига No.6 в циклической программе Вызов регистра сдвига No.7 в циклической программе Вызов регистра сдвига No.8 в циклической программе Инициализация PID-регуэтятора Вызов PID-регуэтятора в циклической программе Передача блока данных из памяти программы пользователя в блок данных RAM

Описание организационных блоков для специальных функций смотри в Разделе 7.

5.2 Эксплуатация

И при однопроцессорной работе контроллера, и при многопроцессорной различают следующие рабочие состояния;

— Состояние "СТОП"
— Запуск
— Выполнение программы

Каждое рабочее состояние делится на три типа (См. Рис.11).

Стоп	Запуск	Выполнение программы
Светодиод "STOP" быстро мерцает	Ручной новый пуск без запоминания	Циклическое
Светодиод "STOP" постоянно горит	Ручной новый пуск с запоминанием	Обработка прерывания от процесса
Светодиод "STOP" медленно мерцает	Автоматический новый пуск с запоминанием	Обработка прерывания по времени

Рис.11 Рабочие состояния

Работа с субмодулями памяти RAM и EPROM описывается в Руководстве по центральному устройству. После подключения напряжения питания CPU обрабатывает программу инициализации, в которой последовательно выполняется предустановка в рабочее состояние:

— инициализация всех модулей памяти,
— создание списка адресов блоков (DBO), показывающего все блоки, которые есть в программе пользователя,
— определение однопроцессорного или многопроцессорного режима работы.

Если возникает одна из ошибок:

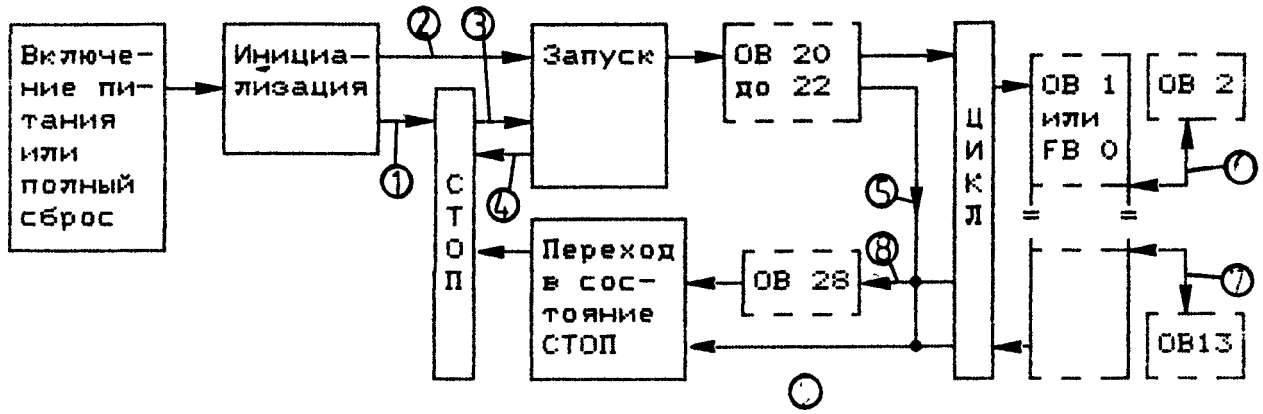
— неправильное содержимое памяти,
— отсутствие субмодуля памяти программы пользователя или если субмодуль памяти EPROM пуст,
— прерывание буферного напряжения литания памяти RAM CPU или памяти RAM программы пользователя,

обнаруженная во время инициализации, то CPU переходит в состояние "СТОП", и светодиод "STOP" начинает быстро мерцать. CPU должно быть полностью сброшено (см. Раздел 5.2.2), то же самое надо сделать и при установке модуля в центральное устройство.

Рисунок 12 показывает структуру расположения системных программ. Организационные блоки осуществляют интерфейс между системными программами и программой пользователя. Их не обязательно программировать, так как системные программы выполняют свои Функции даже без организационных блоков.

Примеры:

- Запрограммирован только OBI. CPU начинает выполнять циклическую программу используя выбранный режим запуска, даже если пользователь не запрограммировал 0В запуска.
- Запрограммирован только ОВ13. Системная программа выполняет циклическую операцию (см. Раздел 5.4) и программу обработки прерывания по времени.



1 Ошибки, которые были обнаружены во время инициализации или перед запуском было состояние "СТОП", или переключатель режима работы в положении "STOP", или автоматический пуск после общего сброса,

2 Автоматический новый пуск с запоминанием (см. Раздел 5.3)

3 Ручной новый пуск с/без запоминания (Раздел 5.3)

4 Ошибка во время запуска перед тем, как был вызван 0В запуска (например ошибка DB1, выбор неправильного режима запуска)

5 Ошибка или неисправность во время выполнения программы пользователя в 0В запуска (Раздел 5.7)

6 Выполнение программы обработки прерывания процесса (Раздел 5.5)

7 Выполнение программы обработки прерывания по времени (Раздел 5.6)

8 Вызов ОВ28, если возникла определенная ошибка (Раздел 5.7)

9 Непосредственный переход в состояние "СТОП" без вызова 0В, если возникла определенная ошибка (Раздел 5.7)

Рис.12 Структура расположения системных программ в S-процессоре

Теперь опишем различные типы рабочих состояний.

5.2.1 Состояние останова

Состояние "СТОП" процессора индицируется красным светодиодом "STOP". Существует три различных типа состояния "СТОП":

o Светодиод "STOP" горит постоянно

Состояние "СТОП" возникает после завершения выполнения программы;
a) в однопроцессорном режиме при переводе переключателя режимов работы в положение "STOP";
b) в многопроцессорном режиме как в случае a), и если другой процессор или координатор переходят в состояние "СТОП";
c) в многопроцессорном режиме при выполнении функции PO "СТОП АО"
d) в одно- или многопроцессорном режиме работы из-за аппаратной ошибки в отдельном CPU (NAU, BAU, PEU);
e) после завершения выполнения функции PG "КОНТРОЛЬ" и после выполнения последующего сброса;

o Светодиод "STOP" быстро мерцает (требуется полный сброс)i

a) во время инициализации была обнаружена ошибка, которая привела к нарушению буферного питания памяти пользователя RAM или памяти RAM CPU до того, как было подано сетевое питание;
b) субмодуль памяти пользователя EPROM либо пуст, либо не установлен. Должен быть установлен запрограммированный модуль EPROM;
c) пользователем был сделан запрос на полный сброс (Раздел 5.2.2)

В случаях a) и b) системная программа требует полный сброс. CPU должен быть сброшен (см. Раздел 5.2.2). В случае c) пользователь может выполнить сброс или избежать этого, выбрав режим запуска (Раздел 5.3).

o Светодиод "STOP" быстро мерцает (ошибка)

a) CPU является причиной ошибки (см. Раздел 5.7), в однопроцессорном или многопроцессорном режиме, которая переводит контроллер в состояние "СТОП";
b) CPU неправильно работает (неправильный режим работы, ошибка DB1*)*, даже если после этого переключатель режима работ переведен в положение "STOP". CPU еще не перешло к выполнению циклической программы (соответствует положению точки 4 на Рис. 12);
c) была запрограммирована команда STOP в 0B запуска или в циклической программе пользователя;
d) в однопроцессорном режиме выполнена функция программатора "AG СТОП";

Реакция на состояние "СТОП":

Выдается сигнал BASP (блокировка выходов) (исключение; тестовая операция), блокируя все цифровые выходы, это индицируется светодиодом "BASP". При многопроцессорной работе, если один из CPU переходит в состояние "СТОП", то остальные переходят в "СТОП" автоматически (исключение: Тестовые операции; смотри Раздел 5.2.3). При переводе контроллера в состояние "СТОП" по команде с программатора, при переключении режима работы переключателем на модуле CPU, по команде STOP, или из-за аппаратной ошибки системная программа вызывает OB28, в котором пользователем может быть запрограммирована требуемая реакция (смотри Раздел 5.7).

Возможные пути выхода из состояния СТОП

— Полный сброс, затем ручной новый пуск с сохранением маркеров
— Тестовая операция (см. Раздел 5.2.3)
— Выбор режима запуска (см. Раздел 5.3):
Ручной новый пуск без запоминания или Ручной новый пуск с запоминанием.

5.2.2 Общий сброс

Функция:

— Сброс CPU
— Стирание памяти RAM

Для инициализации полного сброса необходимо:

1. Перевести в состояние "СТОП" (светодиод "STOP" горит постоянно)
2. Перевести переключатель режимов в положение "OVERALL RESET" и удерживать в этом положении; в это время второй переключатель перевести из положения "STOP" в положение "RUN" и затем снова в "STOP".
3. Результат: светодиод "STOP" начинает быстро мерцать, показывая, что требуется "полный сброс"[1]).
4. Удерживая переключатель в положении "OVERALL RESET", перевести переключатель из положения "STOP" в положение "RUN" и затем снова в "STOP".
5. Результат: выполняется функция "полного сброса", затем процессор переходит в состояние "СТОП", и светодиод "STOP", индицируя это, горит постоянно.
6. После "полного сброса" возможен только режим ручного пуска без запоминания.

Если "полный сброс" требуется системой (светодиод "STOP" мерцает быстро, см. выше), то пункты 1 и 3 могут быть пропущены. "Полный сброс" может быть выполнен с помощью программатора (смотри .Руководство по программатору).

5.2.3 Тестовый режим

С помочью тестовой операции можно запустить отдельный процессор Б многопроцессорной системе (или любую требуемую комбинацию CPU) без CPU, находящихся в состоянии "СТОП", блокирующих весь контроллер в целом. Здесь следует отметить следующие моменты:

— Запуск отдельных процессоров не синхронизирован. В зависимости от длины вызываемых при запуске организационных блоков (OB20, OB21, OB22) цикл, запуска CPU – может иметь различное время
— Сигнал BASP не выдается. В случае ошибки цифровые выходы не блокируются (исключения смотри ниже).

— Если во время выполнения тестовой операции обнаруживается ошибка в процессоре, который находится в циклическом режиме, то в "СТОП", переходит только соответствующий процессор.

[1]) Здесь можно избежать "полного сброса", переключая "STOP"—> "RUN"—>"STOP" без переключения селектора! процессор перейдет в состояние "СТОП". После этого надо перейти в режим запуска.

Инициализация тестовых ФУНКЦИЙ

Тестовая функция должна быть деблокирована на KOR (координаторе) (смотри Руководство по координатору).

Переключатель на координаторе, должен быть переключен из положения "STOP" в положение "TEST", после этого светодиод "BASP" должен погаснуть. '

Режим запуска должен быть установлен на CPU, которые должны быть переведены в режим выполнения циклической программы (см. Раздел 5.3).

5.3 Программирование параметров запуска

Системная программа CPU имеет три различных режима запуска контроллера:

- Ручной новый пуск без запоминания (например, после программирования CPU, после "полного сброса"),
- Ручной новый пуск с запоминанием (маркеры и маркеры связи не стираются),
- Автоматический новый пуск с запоминанием (только после сбоя напряжения).

Для каждого режима запуска системная программа вызывает организационный блок, который пользователь может запрограммировать для конкретизации событий во время запуска. Если это не требуется, то данные организационные блоки можно не программировать.

Допустимость режима запуска отображается на программаторе при анализе ошибок в состоянии "СТОП" (см. Раздел 5.7, биты управления NEU-ZUL, MWA-ZUL).

Во время запуска выдается сигнал BASP. Цифровые выходы деблокируются только после начала выполнения циклической программы. Если, например, выполнение циклической программы завершается ошибкой, то ее выполнение начинается сначала после выполнения запуска.

При каждом режиме запуска все выдачи сигналов периферии и отображение процесса на выходах стираются системной программой перед выполнением пользовательских 0В запуска (ОВ20 и ОВ22), и отображение процесса на входах обновляется. Ошибки, которые возникают в программе запуска пользователя (ОВ20 и ОВ22) обнаруживаются и обрабатываются таким же образом, как и в циклической программе (см. Раздел 5.7); однако, время сканирования не контролируется. "Ручной новый пуск без запоминания" - единственный режим, позволяющий впоследствии завершить запуск.

В <u>многопроцессорном режиме работы</u> следует учитывать следующее:

- Запуск будет выполнен только в том случае, если в <u>каждом</u> CPU присутствует блок DB1.
- Остальные процессоры и координатор должны быть запущены с помощью перевода переключателя из положения "STOP" в положение "RUN". Исключения: При тестовой операции запуск координатора может не выполняться. При запуске с помощью программатора (функция "AG СТАРТ"'), координатор может быть запушен автоматически.

— Запуск отдельных процессоров синхронизируется по времени, т.е. они все переходят к циклической работе вместе, даже если время запуска каждого из них различно. Процессоры остаются *в* цикле ожидания до тех пор, пока все процессоры не закончат процедуру запуска (это не относится к тестовой операции).
— Если координатор находится в состоянии работы "RUN", переключатели режимов работы всех процессоров должны быть в положении "RUN". Режим запуска каждого из отдельных процессоров зависит от того, что оператор ввел во время состояния "СТОП". Можно для одного из процессоров выполнить ручной новый пуск с. запоминанием, а для остальных — ручной новый пуск без запоминания. Если не было ввода оператора, тогда все процессоры будут запущены ручным новым пуском с. запоминанием.
— Можно запустить АО только единственным образом, используя запуск KOR, если причиной перехода процессоров с состояние STOP явился только KOR (переключатель режима работы KOR был переведен из положения "RUN" в положение "STOP"). В этом случае при запуске KOR все процессоры выполняют ручной новый пуск с запоминанием.

о Ручной новый пуск без запоминания

Для выполнения ручного нового пуска без запоминания надо:

— удерживать переключатель в положении "RESET",
— перевести переключатель режима работы из положения "STOP" в "RUN",
— после этого при многопроцессорном режиме работы выполняется запуск отдельных процессоров: Запуск KOR
— или используется функция программатора "СТАРТ АО".
Подразумевается, что переключатели режимов работы на всех процессорах и на координаторе находятся в положении "RUN" (смотри Руководство по программатору).

Тогда системная программа:

— Сбрасывает все маркеры, таймеры и счетчики.

Вызов организационного блока ОВ20:

В организационном блоке ОВ20 пользователь может записать программу, которая выполняет отдельные операции инициализации перед запуском циклической программы, например установка маркеров, запуск таймеров, установка выходов и, если необходимо, подготовка ванных для обмена данными между контроллером и устройствами ввода/вывода. ОВ20 должен быть завершен командой BE (конец блока). После обработки ОВ20 начинается циклическое выполнение. программы в ОВ1 или FBO.

Новый пуск: с запоминанием является существенным, если DB1 был сброшен, введен или изменен, и после этого выполняется программа, которая была прервана вследствие запуска.

o Ручной новый пуск с запоминанием

Для выполнения ручного нового пуска с запоминанием надо:

— поставить переключатель в среднее положение;
— перевести переключатель режима работы из положения "STOP" в положение "RUN";
— после этого при многопроцессорном режиме работы выполняется новый пуск отдельных процессоров; запуск KOR;
— или используется функция программатора "СТАРТ AG";
Подразумевается, что переключатели режимов работы на всех процессорах и на координаторе находятся в положении "RUN".

Во время выполнения ручного нового пуска с запоминанием результат достигается до того, как контроллер переходит в состояние останова, и состояние предыдущей операции записывается в счетчик, т.е. маркеры и маркеры связи не стираются. Значения таймеров и счетчиков стираются (см. Примечание).

При выполнении ручного нового пуска с запоминанием вызывается OB21, в котором пользователь может запрограммировать предустановки до начала выполнения циклической программы.

o Автоматический новый пуск с запоминанием

Для выполнения автоматического нового пуска с запоминанием надо:
- во время выполнения циклической программы выключить и затем вновь включить питание;
- положение остальных переключателей должно оставаться в прежнем состоянии.

Если был сбой напряжения во время выполнения программы, контроллер автоматически выполняет новый пуск при появлении питания. В этом случае системная программа сначала вызывает организационный блок OB22, в котором пользователь может запрограммировать предустановки отдельных; состояний. В противном случае процедура выполнения автоматического нового пуска идентична ручному новому пуску с запоминанием. Если контроллер не должен выполнять автоматический новый пуск, то в OB22 должна быть запрограммирована команда "STOP".

```
OB22 :  STP  (STOP)
BE   :  (конец блока)
```

Примечание: Во время выполнения ручного и автоматического пуска с запоминанием пользователь имеет в своем распоряжении сохраненное состояние таймеров и счетчиков при вызове специальной Функции OB255. OB255 должен вызываться до окончания выполнения OB21 или OB22. Он становится активным во время следующего "нового" пуска с запоминанием. Следует отметить, что после каждого нового пуска без запоминания действует установка "состояние стертых таймеров и счетчиков" при вызове OB255.

5.4 Программирование циклической обработки

Выполнение циклической программы - нормальная работа программируемого контроллера (Рис. 13). После прохождения один раз через рабочий режим процессор автоматически возвращается в начало программы, и начинается новый цикл. По очереди выполняются команды языка STEP-5 до конца программы и затем снова с начала.

Организационный блок OB1 или функциональный блок FB0 являются интерфейсом между системной программой и циклически выполняемой программой пользователя. Первая команда STEP-5 в OB1 является первой командой программы пользователя. Если запрограммированы и OB1 и FB0, то системной программой обрабатывается только OB1.

В OB1 или FB0 вызываются программные, функциональные и шаговые блоки, дальше из этих блоков могут быть вызваны другие программные, функциональные и шаговые блоки, и т.д. Максимальная глубина вложений - 24 уровня. Это значение считается как сумма уровней глубины вложения всех трех возможных режимов работы (циклический, прерывание от процесса и прерывание по времени) и если выполняется прерывание (OB28, см. Раздел 5.7).

Время выполнения программы пользователя — сумма времен выполнения блоков, которые вызываются. Если блок вызывается N раз, время его выполнения должно складываться N раз (см. Раздел 1.5.1).

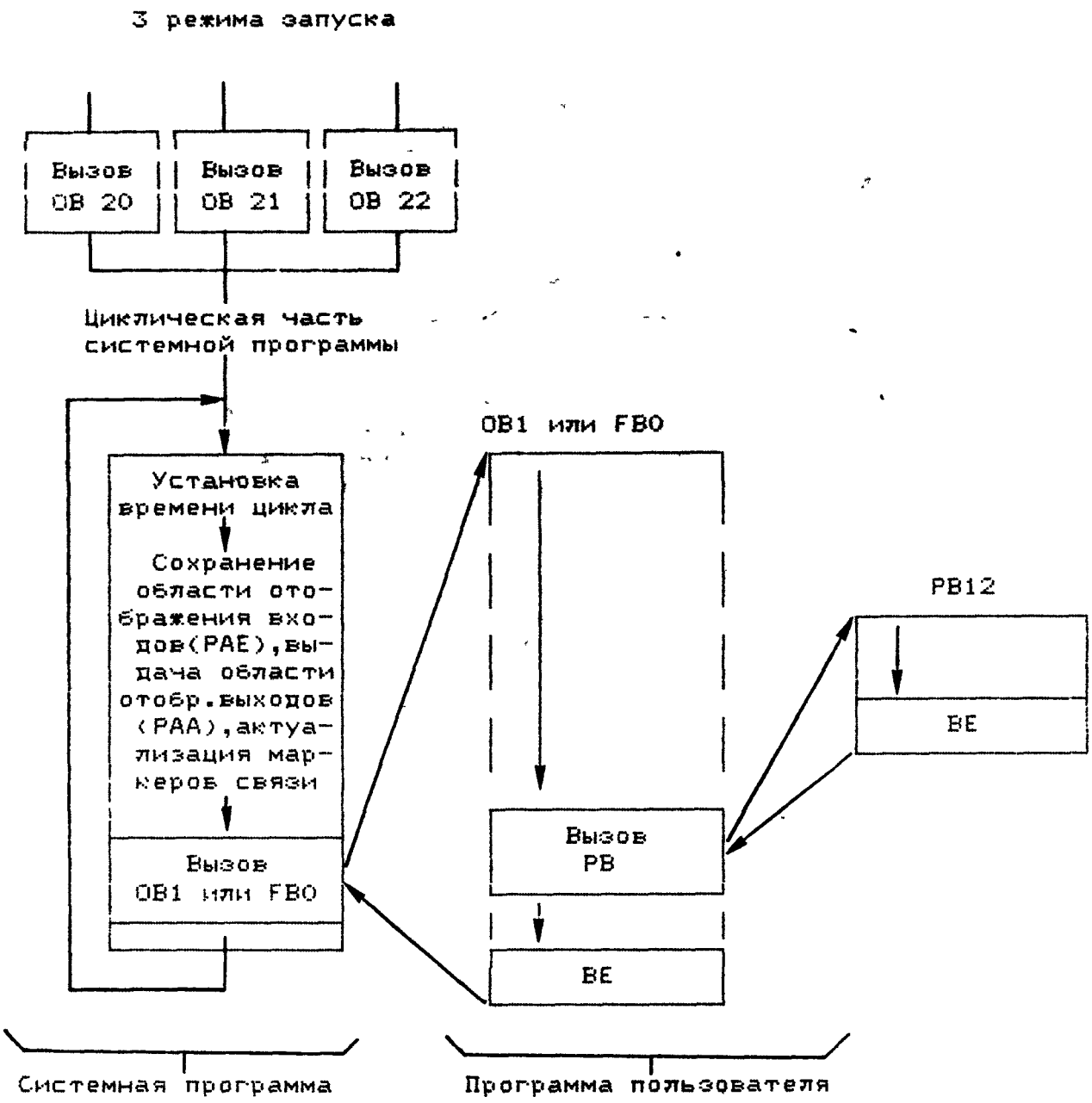


Рис.13 Выполнение циклической программы

о Общая структура программы пользователя

OB1 или FB0 содержит общую структуру программы пользователя. Документирование этого блока предназначается для показа основной структуры программы (Рис. 14) или для придания выразительности частям системы, которые соединяются в группы (Рис.15).

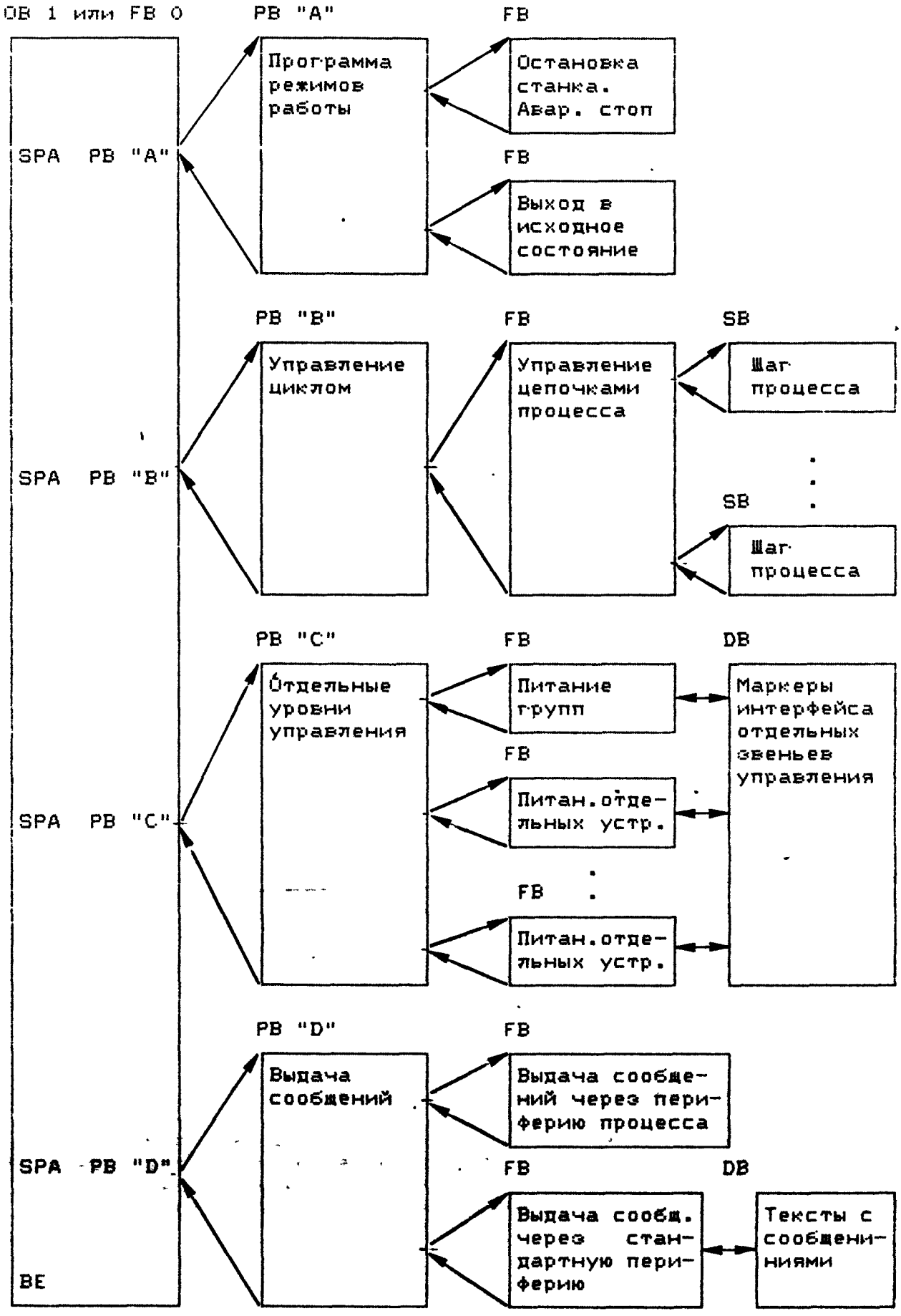


Рис.14 Базовая структура программы пользователя по отношению *к* программной структуре

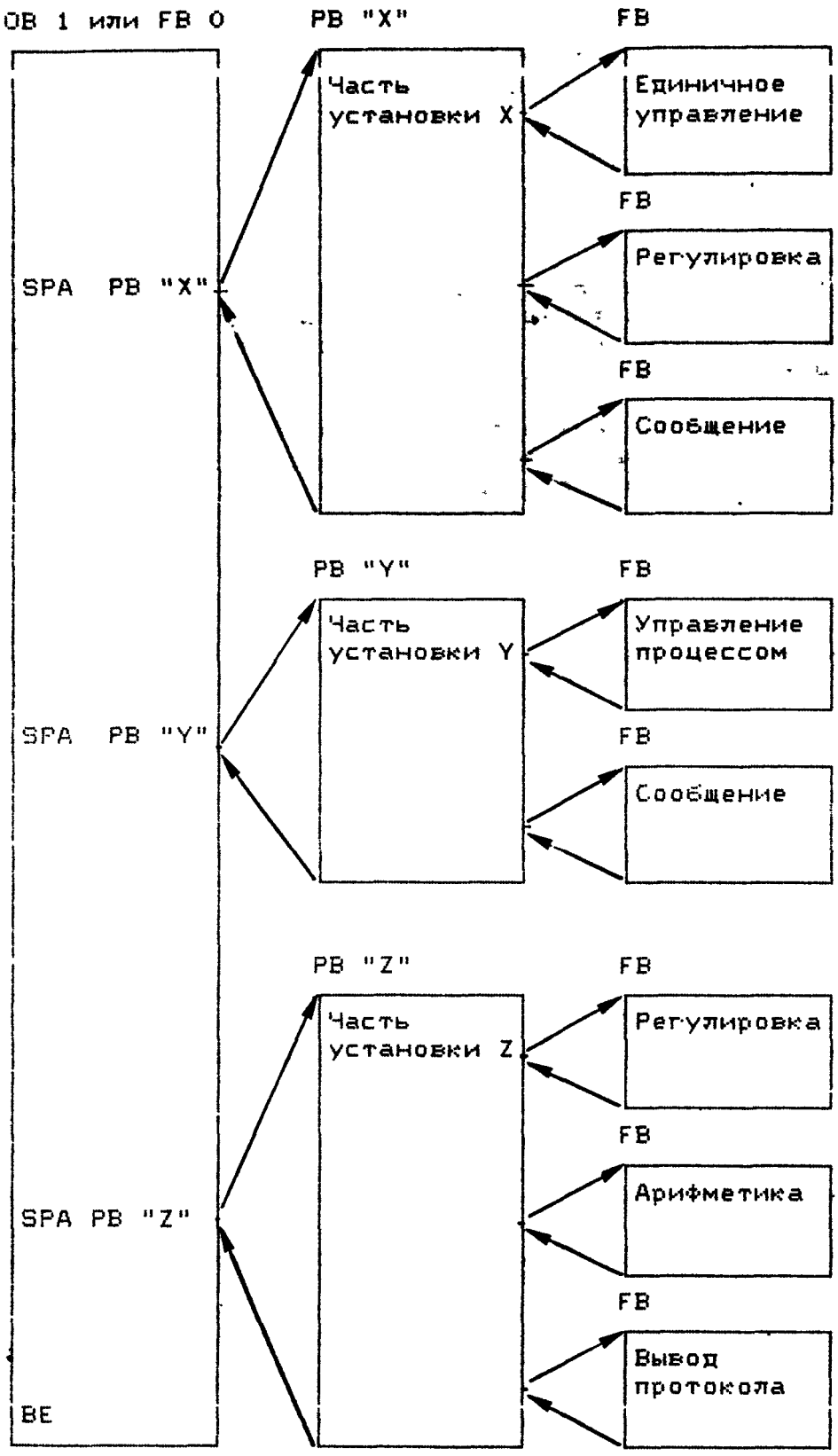


Рис 15. Базовая структура программы пользователя по отношению к системной структуре

o Обработка прерываний циклической программы Выполнение циклической

программы может быть прервано:

— выполнением программы обработки прерываний от процесса (Раздел 5.5)
— выполнением программы обработки прерываний по времени (Раздел 5.6)
— ошибками программирования и аппаратными ошибками (Раздел 5.7)
— сигналами от шины S5 (PEU, BAU, NAU)
— перевоlом переключателя режимов работы на CPU или на КОР из положения "RUN" в положение "STOP", по команде STOP в программе и функцией программатора "СТОП AG".

После прерывания или выполнения программы прерывания по времени автоматически продолжает выполняться циклическая программа с точки, в которой возникло прерывание. Все остальные типы прерываний переводят процессор в состояние "СТОП".

o Индикация

Во время безошибочной циклической работы процессора горит зеленый светодиод "RUN". Красный светодиод "STOP" и светодиоды ошибок погашены.

Замечание! Содержимое арифметических регистров, аккумуляторов с 1 по 4 и значение результата логической операции (см. Раздел 6.1) вне пределов цикла могут иметь неопределенное значение, т.е. они должны быть установлены в начале нового цикла и не должны использоваться после окончания цикла или при запуске.

5.5 Программирование обработки прерываний от процесса

Контроллер S5—135U может обрабатывать прерывания, поступающие в контроллер от процесса. В этом режиме работы выполнение циклической программы прерывается на границе блока сигналом прерывания на шине S5 (смотри Руководство по S-процессору и центральному устройству). После этого системная программа вызывает организационный блок OB2, в котором пользователь может запрограммировать требуемую реакцию на возникший сигнал прерывания. После того, как эта программа будет выполнена, процессор возвращается в точку прерывания и продолжает выполнять циклическую программу, начиная с этой точки.

Обработка прерываний от процесса означает. Что пользователь может непосредственно реагировать на сигналы от процесса.

o Точки прерываний

Прерывание циклической программы не может быть выполнено в произвольной точке по запросу прерывания от процесса. Обычно это возможно только на границе блоков. Когда происходит переход от одного блока к другому (командой вызова блока или при возврате в предыдущий блок по команде конца блока), системная программа может вызвать организационный блок для обработки реакции на сигнал прерывания от процесса.

Обработка прерывания от процесса может, быть прервана только неисправностью устройства и, не может быть прервана запросом на прерывание по времени или вновь поступившим сигналом запроса, на прерывание от процесса. Вновь возникшие запросы на прерывание будут приниматься только после завершения выполнения блока OB2. Во время следующего перехода от блока к блоку циклическая программа прерывается следующим поступившим от процесса сигналом запроса на прерывание.

o Время реакции

Во время обработки блока сигналы запроса на прерывание не обрабатываются. Если возникло прерывание, то оно будет обработано только после перехода к другому блоку, т.е. при вызове следующего блока или при его окончании. Поэтому максимальное время реакции между возникновением сигнала запроса на прерывание и реакцией на это прерывание равно времени выполнения самого длинного блока.

o Блокировка прерываний

Программа обработки прерываний вставляется в циклическую программу на границе блоков. В этой точке выполнение циклической программы прерывается. Это прерывание может иметь отрицательный эффект, если выполнение циклической программы критично по времени, например когда должно быть очень быстрое время реакции.

Если фрагмент программы не должен прерываться, возможны следующие способы достижения .этого:

- Программа не должна делиться на блоки. Поэтому нет переходов и нет возможности прерываний.

- Сама программа , представляет собой программу обработки прерываний. В этом случае тоже не может возникнуть прерывания, т.к. нет смены блоков.
- Запрет прерывания командой AS. Команда AF снова деблокирует обработку прерываний. Сегмент программы между командами AS и AF не может быть прерван сигналами запроса на прерывание.

5.6 Программирование обработки прерываний по времени

Программируемый контроллер 135 U может выполнять программы прерывания по времени. Прерывание по времени выполняется, если приходит сигнал от "внутренних часов" к процессору и вызывается организационный блок OB13, в котором пользователь может записать программу обработки прерывания по времени.

После того, как программа прерывания выполнена, процессор возвращается в ту же точку основной программы, в которой возникло прерывание и продолжает выполнять циклическую программу. Если OB13 не запрограммирован, выполнение циклической программы не прерывается.

o Интерфейс между системной программой и программой обработки прерываний по времени

OB13 является интерфейсом между системной программой и программой обработки прерываний по времени. Он может вызываться системной программой каждые 100 мс.

o Точки прерывания

Прерывание циклической программы не может быть выполнено в произвольной точке по запросу прерывания по времени. Программа обработки прерываний по времени может быть прервана, в свою очередь, сигналом запроса на прерывание от процесса на границе блоков или при возникновении аппаратной неисправности. Но вновь поступившим сигналом прерывания по времени не прерывается.

Если после 100 мс приходит новый запрос на прерывание от OB13 до завершения выполнения программы обработки первого прерывания, процессор переходит в состояние "СТОП", и выдается сообщение об ошибке (WECKFE, см. Раздел 5.7). Выполнение всей программы обработки прерывания по времени, включая программы обработки прерываний от процесса, если они могут возникнуть, должны укладываться в данный интервал времени.

o Время реакции

Во время обработки блока сигналы запроса на прерывание по времени не обрабатываются. Поэтому, если возникло прерывание, то оно будет обработано только после перехода к другому блоку, т.е. при вызове следующего блока или при окончании блока. Таким образом, максимальное, время реакции между возникновением сигнала запроса на прерывание и реакцией на его прерывание равно времени выполнения самого длинного блока. Если в это время выполняется программа прерывания от процесса, то программа прерывания по времени будет выполняться только после завершения обслуживания всех запросов на прерывание от процесса. Максимальное время реакции на сигнал прерывания по времени в этом случае увеличивается на время выполнения программы прерывания от процесса.

5.7 Обработка аппаратных и программных ошибок

Системная программа распознает следующие ошибки процессора, ошибки системной программы и ошибки программы пользователя;

— Вызов блока, который не загружен
— Задержка квитирования (подтверждения) QVZ при обращении к модулям входов/выходов в P— или Q—области или на других адресах шины S5
— Задержка квитирования во время обновления отображения процесса или во время передачи маркеров связи между процессорами
— Ошибка адресации (ADF).
— Превышение времени цикла программы (ZYK)
— Ошибка замещения.
— Ошибка кода команды
— Ошибка при обработке программы прерывания по времени
— Ошибка специальной функции
— Ошибка DB1
— Переполнение стека блоков или стека прерываний

Процессор переходит в состояние "СТОП" и происходит следующее;

— медленно мерцает светодиод "STOP" на процессоре, в котором произошла ошибка. В многопроцессорной системе остальные процессоры переходят в состояние останова, но их светодиоды "STOP" горят постоянно (исключения см. в Разделе 5.2.3).
— цифровые вы/годы блокируются сигналом BASP (исключения см. в Разделе 5.2.3).
— вызывается организационный блок OB28 (если ошибка вызвана переполнением стека блоков или ошибкой DB1, это не выполняется).

Переход в состояние "СТОП" происходит независимо от того, запрограммирован ли, и каким образом OB28.

Аппаратные ошибки, возникающие во время выполнения OB28, регистрируются системной программой таким же путем, как и при выполнении циклической программы. В этом случае процессор останавливается сразу, без вызова OB28. Организационный блок OB28 вызывается при переходе контроллера в состояние "СТОП" по команде с программатора (функция "AG СТОП"), при переключении режима работы переключателем из положения "RUN" в положение "STOP", по команде STOP или по сигналу остановки. Кроме того, на светодиодном индикаторе передней панели процессора выводятся сигналы QVZ, ADF и ZYK.

На следующих страницах показано значение сигналов стека прерываний, которые позволяют проанализировать причину перехода процессора в состояние останова (биты управления и стека прерываний). Биты управления показывают последовательность работы процессора или текущее состояние, причину ошибки, если она есть, и разрешение режима запуска. В стеке прерываний вместе с точкой прерывания показывается текущее состояние и содержимое аккумуляторов.

Кроме того, в стеке прерываний имеется информация о неисправностях и ошибках, возникающих при инициализации, запуске или выполнении программы, каждая из этих фаз более детально описывается в системных данных 3 и 4 (см. Раздел 10).

БИТЫ УПРАВЛЕНИЯ (выводится на экране программатора)

PRISTP	<======	MAFEHL	BARBEND	PGSTP	STPS	STPBEF	HALT
ANLAUF	<======	NEUST	M W A	A W A		NEU—ZUL	MWA—ZUL
			X				
ZYKLUS	<======	SIPROZ	BARB	OB1GEL	FBOGEL	OBPROZA	OBWECKA
X		X		X			X
32KRAM	16KRAM	8KRAM	EPROM	KM—AUS	KM—EIN	DIGEIN	DIGAUS
		X				X	X
URGELO	URLOIA	VERURS	ANL—ABB	UA—PG	UA—SYS		1.UA
CHS—FE	BAT—FE	AWM—FE	RAM—FE	DBO—FE	DB1—FE	DB2—FE	KOR—FE
N A U	P E U	B A U	P A A	ZYK	Q V Z	A D F	WECKFE
OPC—FE	PAR—FE	BSTNIZ	BSTNIG	SF—NIG	SF—LZF		TI—OUT

PRISTP	Процессор в состоянии останова ("СТОП")
MAFEHL	Ошибка процессора
BARBEND	Процессор в состоянии "СТОП" после выполнения функции "КОНЕЦ КОНТРОЛЯ ПРОГРАММЫ"
PGSTP	"СТОП" при выполнении функции "AG СТОП"
STPS	Переключатель режима работ в положении "STOP"
STPBEF	Выполнена команда STOP
HALT	Активна команда HALT; CPU остановлен координатором или другими CPU
ANLAUF	Запуск CPU
NEUST	Выполняется новый пуск
MWA	Выполняется ручной перезапуск
AWA	Выполняется автоматический перезапуск
NEU—ZUL	Допускается новый пуск .
MMA—ZUL	Допускается ручной перезапуск
ZYKLUS	Идет выполнение циклической программы
SIPROZ	Однопроцессорный режим работы
BARB	Выполняется функция программатора "КОНТРОЛЬ ПРОГРАММЫ"
OB1GEL	Загружен OB1. Выполняется циклическая программа, определяемая OB1.
FBOGEL	Загружен FBO. Циклическая программа определяется FBO, если не загружен OB1. Если загружены и FBO, и OB1, циклически выполняется только OB1.

OBPROZA	Загружен организационный блок обработки прерываний OB2, т.е. допускается прерывание процесса (см. Раздел 5.5).
OBWECKA	Загружен организационный блок обработки прерываний по времени OB13, т.е. допускается обработка прерываний через каждые 100 мсек (см. Раздел 5.6).
32KRAM	Память пользователя RAM с 32К слов
16KRAM	Память пользователя RAM с 16К слов
8KRAM	Память пользователя RAM с 8К слов
EPROM	Память пользователя EPROM
KM—AUS	Адресный список для выходных маркеров связи существует
KM—EIN	Адресный список для входные маркеров связи существует
DIGEIN	Адресный список для цифровых входов существует
DIGAUS	Адресный список для цифровых входов существует
URGELO	Выполняется поэтапный сброс CPU
URLOIA	Выполняется сброс
VERURS	Процессор является причиной остановки контроллера
ANLABB	Прерывание во время запуска
UA—PG	Запрос сброса с программатора
UA—SYS	Запрос на общий сбой от системной программы (должен быть выполнен общий сброс)
1.UA	Запрос на первый сброс. (= Подготовка к сбросу р помощью переключателей)
CHS—FE	Ошибка контрольной суммы во время проверки системных программ на PRQMax.
BAT—FE	Неисправность батареи (запуск невозможен)
AWM—FE	Ошибка модуля памяти пользователя (необходим сброс)
RAM-FE	Ошибка системной программы в памяти RAM. Необходим сброс.
DBO—FE	Ошибка при создании списка адресов блоков
DB1—FE	Ошибка при создании списка адресов периферии для обновления отображения процесса: DB1 в многопроцессорном режиме не программируется или входы и выходы, указанные в DB1, не дают подтверждения на соответствующих модулях.

DB2—FE	Не используется
KOR—FE	Ошибка во время обмена данными с координатором
NAU	Ошибка питания
PEU	Входы/выходы не работают
BAU	Не допускается использование буферной батареи
PAA	Задержка квитирования во время обновления отображения процесса или во время передачи маркеров межпроцессорной связи.
ZYK	Превышение допустимого времени цикла (см. Раздел 1.5.1)
WECKFE	Запрос обработки прерывания по времени (= процесс, управляемый по времени) в то время, пока предыдущее прерывание все еще обрабатывается (см. Раздел 5.6)
QVZ	Задержка квитирования
ADF	Ошибка адреса
OPC—FE	Ошибка кода команды
PAR—FE	Неправильный параметр для этой команды
BSTNIG	Вызываемый блок не загружен
BSTNIZ	Неправильный вызов блока
3F—NIG	Вызываемая специальная функция не существует
SF—LZF	Ошибка во время выполнения специальной функции
TI—OUT	Не используется

СТЕК ПРЕРЫВАНИЙ (выводится на PG)

```
TIEFE:      01

BEF-REG:   34C8    SAZ:      004F      DB-ADR:   002C    BA-ADR:   005A
BST-STP:   EAD8    OB-NR:      28      DB-NR:        5   PB-NR:        1
                   REL-ADR:0002        DBL-REG:  0000
VEK-ADR:   0000    UAMK:     0200      UALW:     0000

AKKU1:0000 6789  AKKU2:0000 6789   AKKU3:0000 0000   AKKU4:0000 0000

БИТЫ РЕЗУЛЬТАТА:    ANZ1   ANZO   OVFL   OVFLS ODER   STATUS VKE ERAB
                                                          X
ПРИЧИНА ОШИБКИ:     NAU    PEU    BAU    HALT  ZYK     QVZ    ADF STPS

                    BCF    SUF    TRAF   SFF   STUEB   STUEU ZFE
                     X

TIEFE:      02

BEF-REG:   6500    SAZ:      005A      DB-ADR:   002C    BA-ADR:   0024
BST-STP:   EADC    OB-NR:       1      DB-NR:        5   PB-NR:        1
                   REL-ADR:0004        DBL-REG:  0000
VEK-ADR:   0050    UAMK:     0000      UALW:     0000

AKKU1:0000 6789  AKKU2:0000 6789   AKKU3:0000 0000   AKKU4:0000 0000

БИТЫ РЕЗУЛЬТАТА:    ANZ1   ANZO   OVFL   OVFLS ODER   STATUS VKE ERAB

ПРИЧИНА ОШИБКИ:     NAU    PEU    BAU    HALT  ZYK     QVZ    ADF    STPS
                                                                 X
                    BCF    SUF    TRAF   SFF   STUEB   STUEU ZFE
```

TIEFE	Уровень стека прерываний, показывающий хронологический порядок возникновения прерываний
BEF—REG	Регистр команд. Содержит машинные коды (первое слово) следующей выполняемой команды или команды, которая только что была выполнена
BST—STP	Указатель стека блоков
VEK—ADR	При возникновении ошибки PAA показывается область возникновения неисправности (отсутствие квитирования), в противном случае то же самое, что и SAZ: E200H...E27FH: Цифровые входы 128...O (вход 7FH...O) E280H...E2FFH: Цифровые выходы 128...O (вход FFH...80H) E3OOH...E3FFH: Входные маркеры межпроцессорной связи 256...O (вход FFH...O); для системных программ версии A7 - по адресу EOOOH...EOFFH E400H...E4FFH: Выходные маркеры межпроцессорной связи 256...O (вход FFH...O); для системных программ версии A7 — по адресу E100H...E1FFH

SAZ	Счетчик адреса, показывающий абсолютный адрес команды, которая должна выполняться следующей, или только что выполненной команды в программной памяти
FB—NR	Тип и номер текущего блока (OB, DB, SB, PB или FB)
REL—SAZ	Счетчик относительного адреса, показывающий адрес команды, которая будет выполняться следующей, или только что выполненной команды относительно начала текущего блока (относительный адрес показывается программатором при установке ключевого переключателя в положение "блокировка ввода", смотри руководство по программатору)
UAMK	Слово маски отображения прерываний
UALW	Слово стирания отображения прерываний
DB—ADR	Абсолютный начальный адрес выбранного блока данных в программной области (адрес первого слова данных; 7FFFH, если DB не был адресован)
DBL—REG	Не используется
BA—ADR	Абсолютный адрес последней команды, которая должна была быть выполнена в последнем вызывающем блоке (команда перехода)
OB—NR	Тип и номер блока последнего вызывающего блока (OB,DB, SB, FB, PB)

БИТЫ РЕЗУЛЬТАТА: См. Раздел 6.1

AKKUl..AKKU4	Содержимое аккумуляторов
NAU	Неисправность сетевого питания
PEU	Блокировка входов/выходов (=неисправность питания в устройстве расширения)
BAU	Блокировка буферной батареи
HALT	Сигнал останова по линии HALT (CPU останавливается координатором или другим процессором)
ZYK	Превышение времени цикла
QVZ	Задержка квитирования .
ADF	Ошибка адресации
STPS	Переключатель режима работ в положении "STOP", или команда STOP, или останов по команде с программатора
BCF	Ошибка кода команды
SUF	Ошибка подстановки

TRAF	Не используется
SFF	Ошибка из группы специальных функций (соответствует управляющим битам SF—NIG и SF—LZF)
STUEB	Переполнение стека блоков - превышение глубины вложения 24 блоков
ZFE	Ошибка таймера (ошибка прерывания по времени или ошибка временных импульсов в CPU)

o Описание ошибок

QVZ	Задержка квитирования; адреса, к которым было обращение по шине 85, не дают подтверждения от соответствующих модулей, например : — нет подтверждения от модулей при прямом обращении к периферии (входам и выходам) с использованием команд L/T, PB, PW, QB, QW — задержка квитирования во время обновления отображения процесса (см. Раздел 6.2) или во время передачи маркеров межпроцессорной связи (см. Раздел 1.4.1)
ADF	Ошибка адресации; отображение процесса (см. Раздел 6.2) было адресовано (с помощью команд загрузки/передачи L/T EB, EM, ED, AB, AM, AD или с помощью двоичных логических команд или команд запоминания) к вводам или выходам, которые не подтверждаются соответствующими модулями входов/выходов при последнем запуске или к выходам, которые не были указаны при последнем новом пуске с запоминанием.
SUF	Ошибка подстановки; команда подстановки (замещения) была замещена такой командой, которая недопустима в данном случае (см. Раздел 6.3).
BCF	Ошибка кода команды; неправильный список операций или параметров, а также при вызове незагруженного блока. Пример: Использование таймера или счетчика, который был маскирован изменением длины области таймеров или счетчиков или если его номер превышает 127.
SFF	Ошибка группы специальных функций; вызываемая специальная Функция не существует, были указаны неверные параметры или неправильная обработка.
ZFE	Ошибка таймера; при выполнении прерывания по времени второй запрос приходит через 100 мсек до того, как была завершена обработка предыдущего прерывания (смотри Раздел 5.6). Случается также, если происходит сбой тактовых импульсов CPU.

6 Список команд языка STEP—5 с примерами программирования

6.1 Основные правила

Большинство операций STEP—5 используют два регистра (32 бита.) в качестве исходного операнда и в качестве назначения для результата. Это аккумуляторы АКК 1 и АКК 2.

В зависимости от метода адресации (байтовая, словная или двойными словами) команды загрузки и передачи используют содержимое аккумуляторов следующим образом:



АКК 1 всегда является источником для операции передачи и назначением для операций загрузки.

При операциях загрузки байта или слова старшие биты младшего слова в АКК 1 которые не используются, всегда заполняются нулями. Перед загрузкой в АКК 1 содержимого из адреса, по которому происходит обращение, старое содержимое младшего слова передается из АКК 1 в АКК 2.

При операциях загрузки байта или слова старшее слово в АКК 1 и 2 остаются без изменений. При операциях загрузки двойного слова полное содержимое АКК 1 передается в АКК 2 перед загрузкой.

При выполнении операций передачи содержимое АКК 1 и 2 остается без изменений. Вспомогательные регистры (АКК 3 и 4) остаются без изменений во время выполнения всех операций загрузки и передачи.

o Представление чисел

В качестве операндов для команд языка STEP—5, которые выполняют логические действия, сравнение или изменение содержимого АКК 1 и 2, могут быть использованы числа в различных видах представления. В зависимости от выполняемой операции, содержимое АКК 1 и 2 интерпретируется в одном из следующих видов:

а) Число с Фиксированной запятой; интерпретируется в качестве 16 ,битового двоичного числа в виде двоичного дополнения (расширение числа с фиксированной запятой с 16 до 32 бит, смотри Раздел 7.5). Пределы чисел : -32768....+32767. Пример (загрузка числа с фиксированной запятой): L KF -12876.

б) Число в коде BCD со знаком и 3 цифры

Значение битов в АКК 1:

Биты	15...12	11...8	7...4	3…0
	Знак	Сотни	Десятки	Единицы

Отдельные цифры представляются в виде положительных 4-битовых чисел в двоичном представлении.

Знак: 0000 - если число положительное

1111 - если число отрицательное

Предельные значения: -999...+999

с) Числа с плавающей запятой; интерпретируются как 32-битовые двоичные числа с 8-битовой экспонентой и 24-битовой мантиссой. При выполнении операций с числами с плавающей запятой +G, -G, xG, :G в S-процессоре мантисса интерпретируется как 16-битовое число; 8 младших битов устанавливаются в О.

Пример: (Ввод с программатора числа Z с плавающей точкой)

```
Z = 12.34567                    |  Z = -0.005
L KG + 1234567 + 02             |  L KG - 50000000 - 02
       ---------- ----          |         ---------- ----
       Мантисса   Экспонента    |         Мантисса   Экспонента
                  (основание 10)|                    (основание 10)
                                |
Z = +0.1234567x10**+2=12.34567  |  Z = -0.5 x 10**-2 = -0.005
```

Пределы чисел, которые могут быть представлены:

+ 0.1469368 x 10^{-35} ..... + 0.1701412 x 10^{39} и

- 0.1469368 x 10^{-35} ..... - 0.1701412 x 10^{39}

Примечание ! Внутреннее представление может не соответствовать формату, в котором числа вводятся при создании программы на программаторе. (смотри Руководство по программатору). Программатор создает представления чисел, показанные выше.

о Биты результата

Есть команды для обработки информации, состоящей из отдельных битов, и команды для обработки информации в виде байтов и слов. (8 или 16 бит*).*
В обеих группах есть команды, которые устанавливают коды условий и команды, которые интерпретируют их (смотри Раздел 8: Список инструкций, влияние на коды условий). Есть коды условий битовые и коды условий слоеные, которые соответствуют группам команд. Байт кодов условий может быть выведен с помощью программатора в виде таблицы:

	Коды условий словные				Коды условий битовые			
	ANZ1	ANZO	OV	OS	OR	STA	VKE	ERAB
Бит	7	6	5	4	3	2	1	0

Операции переходов позволяют непосредственным образом интерпретировать коды (Раздел 4.3).

Объяснение битовых кодов условий г

ERAB Первый опрос; начало логической операции. В конце цепочки логической операции (например, операций с памятью) ERAB устанавливается в состояние О. Команды, которые устанавливают ERAB в О (например, команда присвоения = Ax.x), ограничивают VKE (смотри соответствующую колонку в списке команд в Разделе 8), т.е. в дальнейшем результат логической операции может быть интерпретирован (например в командах, зависимых от VKE), но не может быть изменен. Только после первой логической команды (означающей начало цикла сканирования) VKE может быть изменен.

VKE Результат логической операции; результат битовой логической операции. Команда "истинно" при выполнении команды сравнения (смотри Приложение: Список команд, двоичные логические операции или операции сравнения).

STA Статус; в битовых командах определяется логическое состояние бита, который только что был опршен или установлен. Статус обновляется при двоичных логических операциях <(исключая U(, 0(,), 0> и операциях с памятью.

OR ИЛИ; информация для CPU, что следующая логическая операция "И" должна быть выполнена перед Логической Операцией "ИЛИ" (И перед ИЛИ).

Объяснение словных кодов условий:

OV Переполнение; определяется, произошло ли переполнение (превышение допустимых границ) при выполнении последней арифметической операции.

OS Запоминание переполнения; бит переполнения запоминается. Этот бит используется для индикации, на каком уровне выполнения нескольких арифметических операций произошло переполнение.

ANZ1 и ANZ0 - биты кодируемого результата, которые интерпретируются в соответствии со следующей таблицей.

Биты результата		Результат вычисления с фиксирован. запятой	Цифровая логическая операция	Сравнение содержимого АКК 1 и АКК 2	Сдвиг; последний сдвинутый бит
ANZ1	ANZO				
0	0	Результат = 0	*st* 0 *-^0,0*	АКК 2 = АКК 1	0
0	1	Результат < 0		АКК 2 < АКК 1	-
1	0	Результат > 0		АКК 2 < АКК 1	1

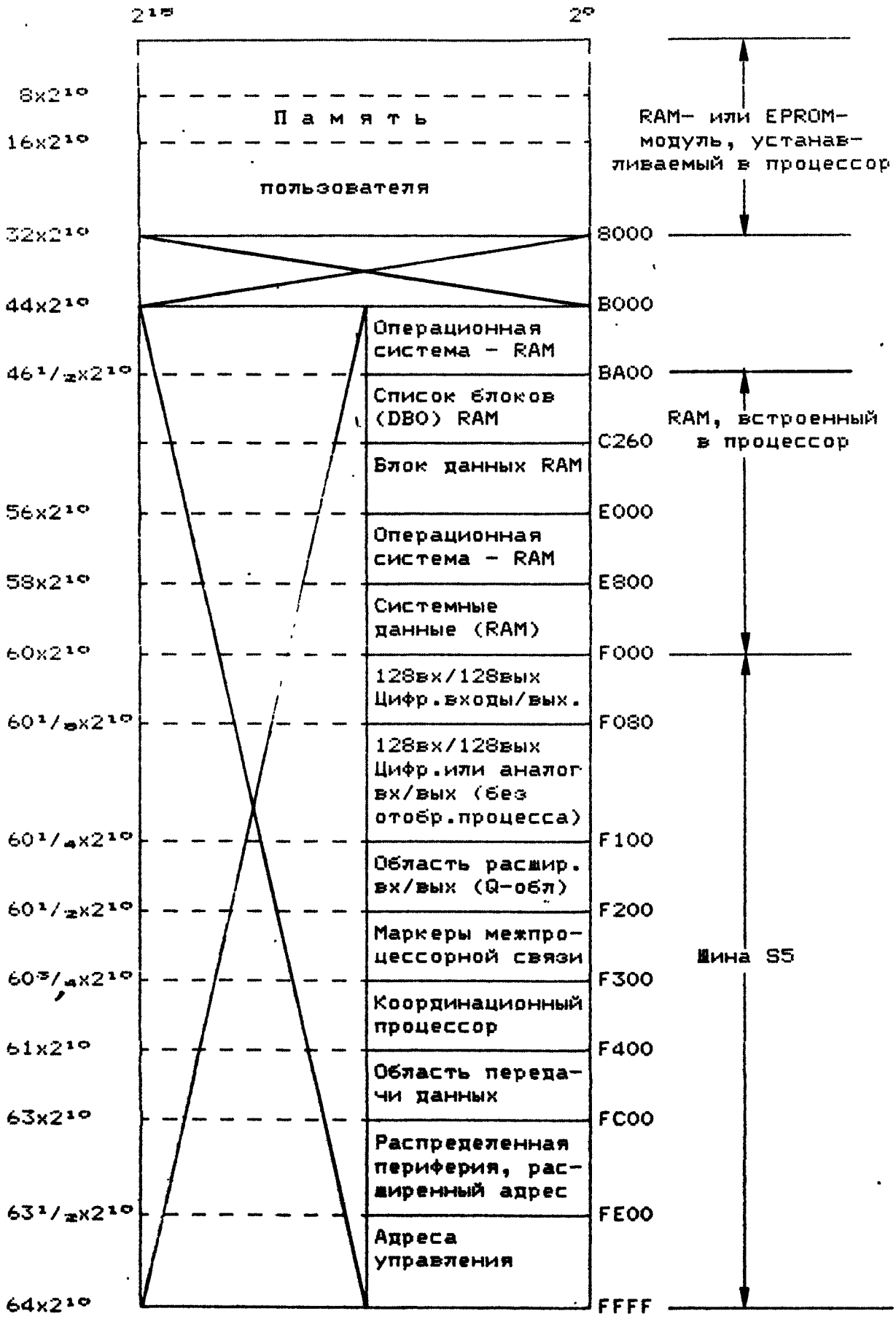


Рис 16. Распределение памяти в S-процессоре

о Адресные области для входов/выходов программ

Область (абсолютный адрес)		Адресуется с помощью	Параметр	Адресная область
EF00 EF7F	PAE (отображение входов)	L EB/T EB L EW/T EW L ED/T ED L E/UN E/O E/ON E S E/R E/=E	0...127 0...126 0...124	RAM CPU
EF80 EFFF	PAA (отображение выходов)	L AB/T AB L AW/T AW L AD/T AD L A/UN A/O A/ON A S A/R A/= A	0...127 0...126 0...124	
F000 F07F	Цифровые входы/выходы	L PB/T PB L PW/T PW	0...127 0...126	
P — периферия с отображением процесса				
F080 EF7F	Цифровые или аналоговые входы/выходы	L PB/T PB L PW/T PW	128...255 128...254	шина S5
P — периферия без отображения процесса				
F080 EF7F	Область расширенной периферии	L QB/T QB L QW/T QW	0...255 0...254	
Q — периферия				

Рис.17 Распределение адресов в контроллере 135 U

6.2 Основной набор операций

o Двоичные логические операции

Операция	Параметр	Функция
) U(0(0		Скобку закрыть Операция "И" для выражения в скобках Операция "ИЛИ" для выражения в скобках Операция "ИЛИ" над функциями "И"
U 0 E A M N E N A N M T N T Z N Z	 0.0...127.7 0.0...127.7 0.0...255.7 0.0...127.7 0.0...127.7 0.0...255.7 1...127 1...127 1...127 1...127	Операция "И" Операция "ИЛИ" с опросом входа на состояние "1" с опросом выхода на состояние "1" с опросом маркера на состояние "1" с опросом входа на состояние "0" с опросом выхода на состояние "0" с опросом маркера на состояние "0" с опросом таймера на состояние "1" с опросом таймера на состояние "0" с опросом счетчика на состояние >0 с опросом счетчика на состояние =0

Двоичные логические операции создают результат логической операции (VKE).

В начале выполнения цепочки логических операций (первый опрос) результаты первой логической операции зависят только от состояния опрашиваемых сигналов и от того, используется или нет их инверсное состояние (N- отрицание); однако, они не зависят от типа логической операции (U = UND <"И"> , O = ODER <"ИЛИ">).

Во время выполнения цепочки логических операций VKE Формируется в зависимости от типа логической операции, предыдущего состояния VKE и состояния опрашиваемого сигнала. Цепочка логических операций завершается командой, ограничивающей VKE (/ERAB = 0) (например операция установки).

VKE остается без изменений до тех пор, пока не начнется следующий "первый опрос". После этого VKE может быть интерпретирован, но не может быть установлен. Для того, чтобы интерпретировать VKE, может быть использована команда, зависимая от VKE (См. колонку в списке операций).

Программа	Статус	VKE	ERAB
:			
= A 0.0	0	0	0 — ограничение VKE, конец цепочки логической операции
U E 1.0	1	1	1 — первый опрос
U E 1.1	1	1	1
U E 1.2	0	0	1
= A 0.1	0	0	0 — ограничение VKE, конец цепочки логической операции

o Операции установки/сброса

Операция	Параметр	Функция
S		Установка
R		Сброс
=		Присвоение
E	0.0...127.7	входов в PAE
A	0.0...127.7	выходов в PAA
M	0.0...255.7	битов маркеров
D	0.0...255.15	битов слова данных

o Функции загрузки, передачи и сравнения

Операция	Параметр	Функция
L		Загрузка
T		Передача
EB	0...127	входного байта в/из PAE
EW	0....126	входного слова в/из PAE
ED	0....124	входного двойного слова в/из PAE
AB	0....127	выходного байта в/из PAA
AW	0....126	выходного слова в/из PAA
AD	0....124	выходного двойного слова в/из PAA
MB	0....255	маркерного байта
MW	0....254	маркерного слова
MD	0....252	маркерного двойного слова
DR	1....255	младшего байта слова данных из DB
DL	1...255	старшего байта слова данных из DB
DW	1....255	слова данных из DB
РБ	0....127	периферийного байта модулей цифровых входов/выходов (зависит от PAE/PAA)
	128...255	периферийного байта модулей аналоговых входов/выходов
QB	0....255	байта расширенной периферии (область Q)

o Функции загрузки, передачи и сравнения

Операция	Параметр	Функция
L T		Загрузка Передача
PW	0....126	периферийного слова модулей цифровых входов/выходов (область P)
	126...254	периферийного слова модулей аналоговых входов/выходов (область P)
QW	0.... 254	слова расширенной периферии (область Q)
L		Загрузка
KM	Любая битовая комб. (16 бит)	константы в виде битовой комбинации
KH	0...FFFF	константы в 16-ричном коде
KF	от -32768 до +32768	константы в виде числа с фиксированной запятой
KY	0...255 для кажд. байта	двухбайтовой константы
KB	0...255	однобайтовой константы
KC	2 любых алфав - цифр. символа	константы (2 символа в коде ASCII)
KT	0.0...999.3	константы времени
KZ	0... 999 [1])	константы счета
KG		константы в виде числа с плавающей запятой (32 бита)
T	0...127	значения таймера
Z	0...127	значения счетчика
LC T	0...127	BCD-кодированное значения таймера
LC Z	0...127	BCD-кодированное значения счетчика
! =		Сравнение на равенство
><		Сравнение на неравенство
>		Сравнение на больше
>=		Сравнение на больше/равно
<		Сравнение на меньше
<=		Сравнение на меньше/равно:
F		двух чисел с фиксир. запятой (16 бит)
D		двух чисел с фиксир. запятой (32 бита)
G		двух чисел с плавающей запятой (32бита)

Операции загрузки и передачи не влияют на коды условий. Команды сравнения устанавливают в качестве результата VKE и коды ANZ0 и ANZ1. Сравнивается всегда содержимое аккумуляторов АКК 1 и АКК 2 (смотри примеры программ и список команд).

Для операций загрузки. и передачи следует обратить внимание на инструкции в Разделе 6.1. Обращение к входам/выходам может быть выполнено непосредственно — с помощью операций загрузки и передачи L/T, PB, PW, QB, OW или через область отображения процесса — с помощью операций L/T EB, EW, ED, AB, AW, AD и с помощью логических операций. При использовании операций T PB 0...127 и T PW 0...126 одновременно устанавливается и PAA. (PAE/PAA -отображение процесса входов/выходов для 128 байтов периферии с адресами от 0 до 127).

Отображение процесса представляет собой область памяти, содержание которой только выдается на периферию (PAA) или считывается с периферии (PAE) один раз за цикл выполнения программы пользователя (см. Рис. 13). Это исключает частое изменение логического состояния битов в процессе .,выполнения циклической программы пользователя, которое может привести к дребезгу соответствующих выходных устройств.

Q—область может быть адресована только при использовании интерфейсных. модулей 300 или 301, так как модули входов/выходов, адресуемые в Q-области, могут быть установлены только в устройство расширения. Для всей Q— и P—области с относительными адресами байтов от 128 до 255 отображения процесса нет.

С помощью операнда, загрузки и передачи слов для адресации области, имеющей байтовую структуру (PAA, PAE, маркеры, шина S5), загружается/передается байт N и N+1; при использовании двойных слов загружаются/передаются байты с N по N+3.

Пример

L EW 5 в аккумулятор загружаются байты 5 и 6 из области PAE

L MD 10 в аккумулятор загружаются 10,11,12 и 13 байты маркеров

o Операции с таймерами и счетчиками
Перед тем, как загрузить таймер или счетчик, необходимо предварительно значение времени или счета загрузить в аккумулятор. Имеют смысл следующие операции загрузки:

Для таймеров: L KT, L EW, L AW, L MW, L DM
Для счетчиков: L KZ, L EW, L AM, L MW, L DW

Операции с таймерами и счетчиками не изменяют содержание аккумуляторов.

Операция	Параметр	Функция
SI T	1...127	Запуск таймера в качестве импульса
SV T	1...127	Запуск таймера в качестве удлиненного импульса
SE T	1...127	Запуск таймера в качестве задержки включения
SS T	1...127	Запуск таймера в качестве запоминающей задержки включения
SA T	1...127	Запуск таймера в качестве задержки выключения
R T	1...127	Сброс таймера
S Z	1...127	Установка счетчика
R Z	1...127	Сброс счетчика
ZV Z	1...127	Прямой счет Обратный счет
ZR Z	1...127	

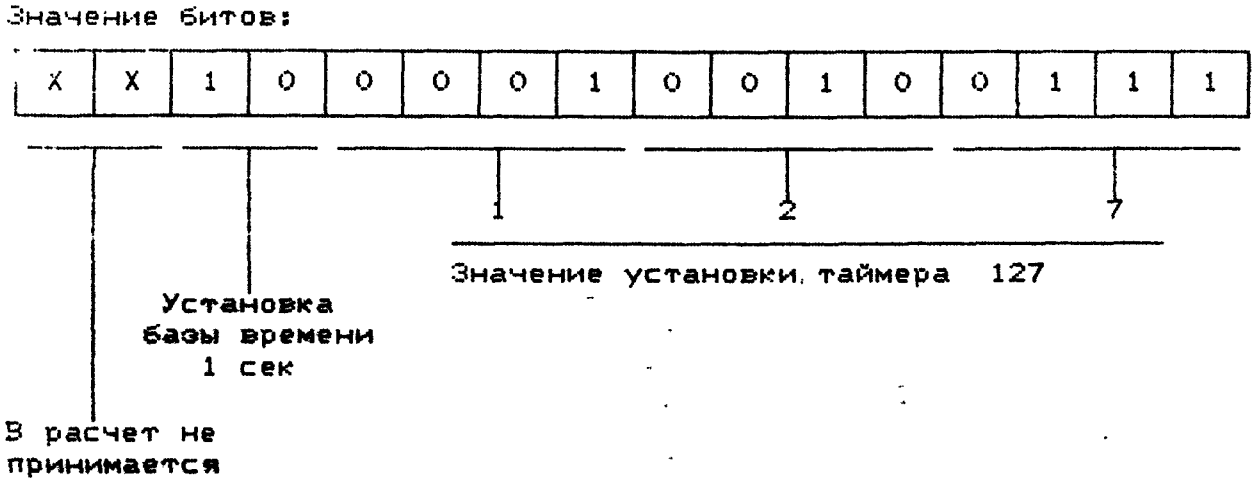
При использовании операций таймеров SI, SE, SV, SS, SA и S или счетчиков значение, записанное в аккумуляторе 1, переносится в область памяти, соответствующую данному таймеру/счетчику (команде передачи), и начинает выполняться соответствующая операция.

Если значение таймера или значение счетчика загружается при помощи EW, AW, MW или DM, то соответствующее слово должно иметь следующую структуру:

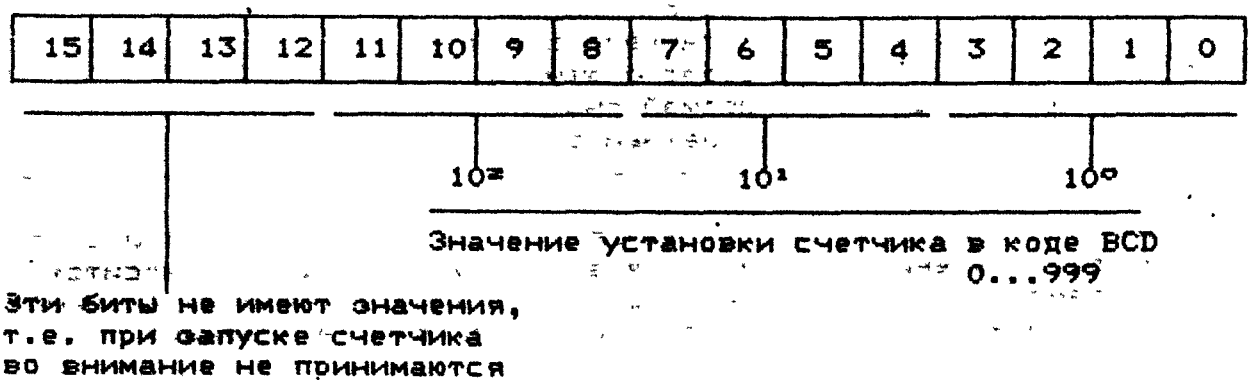
Для значения таймера



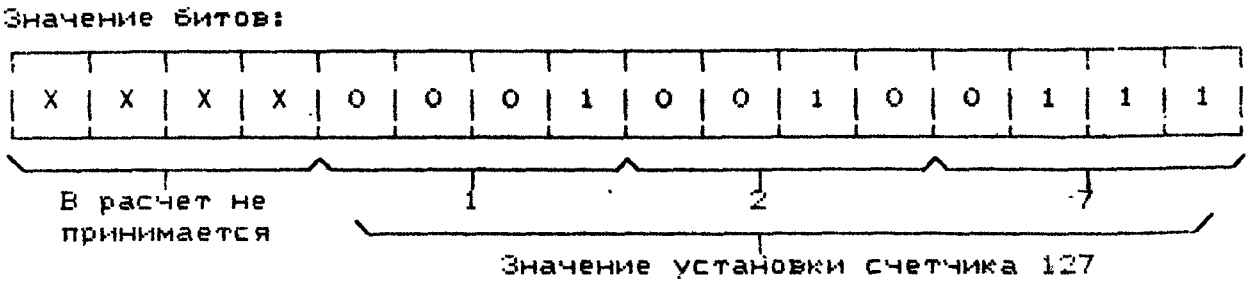
Пример : Установка времени 127 секунд



Для значения счетчика

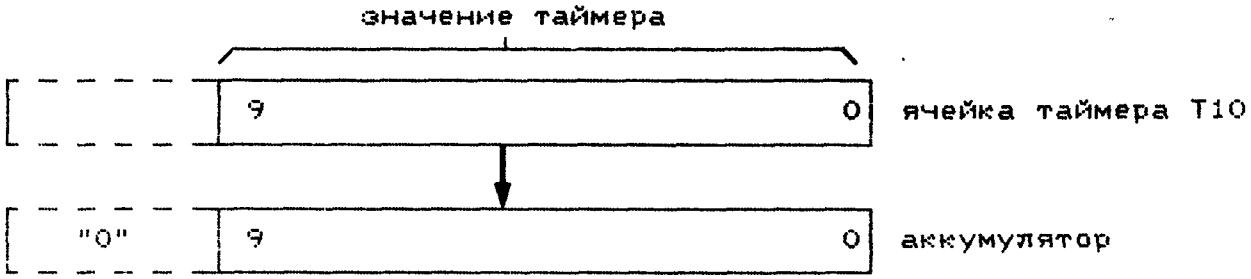


Пример : Установка значения счетчика .127



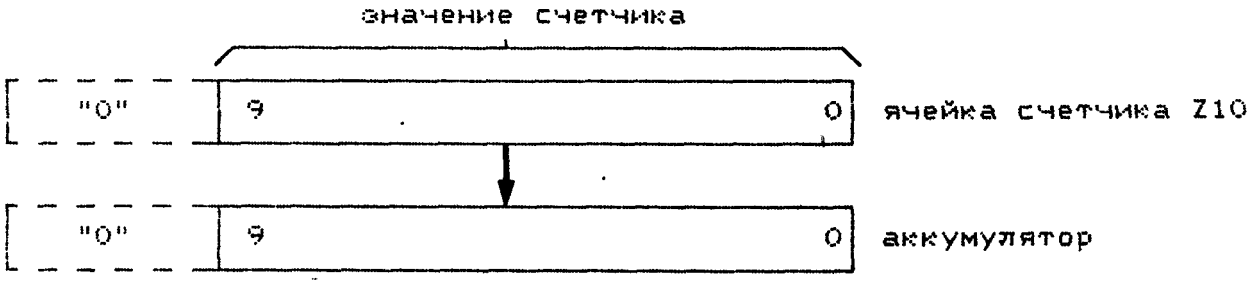
Значение таймера или счетчика хранится в области памяти, отведенной для этих целей, в виде двоичных кодов. Для того, чтобы опросить таймер - или счетчик, значение из области таймеров или счетчиков должно быть перенесено в аккумулятор непосредственно или в коде BCD.

Пример: Непосредственная загрузка значения таймера;



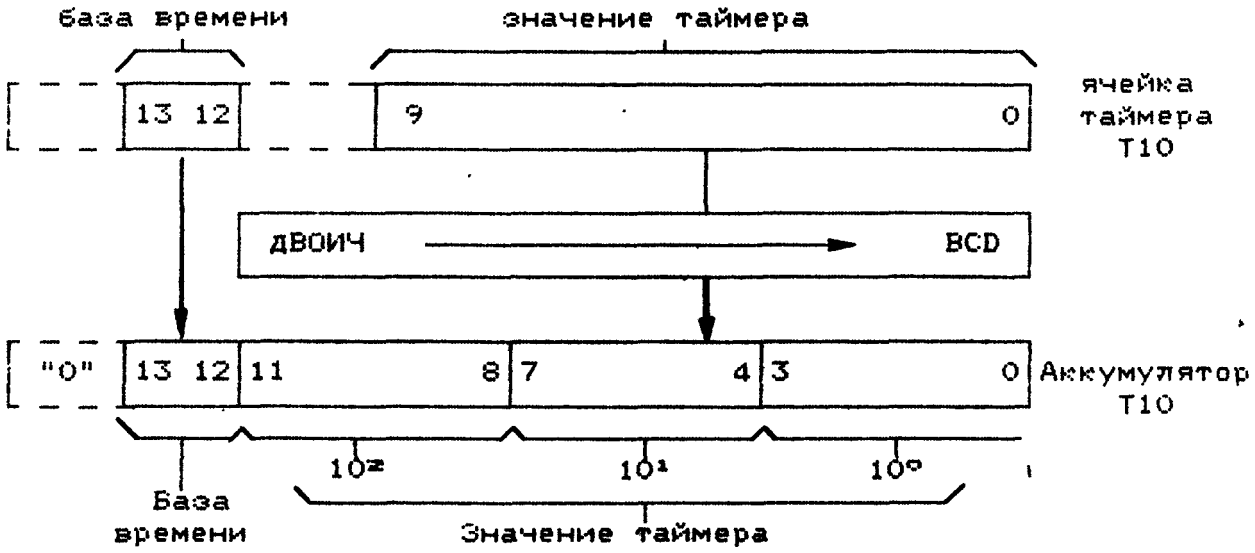
L T 10 непосредственная загрузка двоичного значения таймера T 10 в аккумулятор
Значение базы времени здесь не загружается

Непосредственная загрузка значения счетчика:



L Z 10 непосредственная загрузка двоичного значения счетчика 210 в аккумулятор

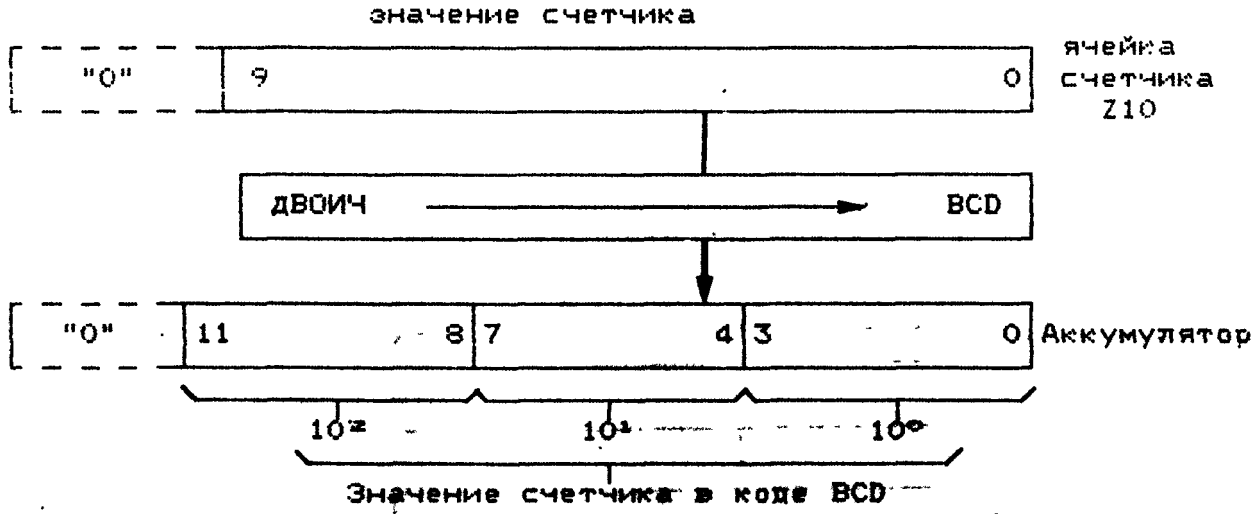
Кодированная загрузка значения таймера:



LC T 10 кодированная загрузка значения таймера T 10 и базы времени в аккумулятор

Значение базы времени здесь тоже загружается.

Кодированная загрузка значения счетчика;



LC Z 10 кодированная загрузка значения счетчика Z10 в аккумулятор

При кодированной загрузке значение битов 14 и 15 ячейки таймера и значение битов с 12 по 15 ячейки счетчика не загружается. На их место в аккумуляторе записывается "О". После этого содержимое аккумулятора может быть нужным образом обработано.

o Арифметические операции

Операция	Параметр	Функция
+ F		Сложение двух чисел с фиксированной запятой
- F		Вычитание двух чисел с фиксиров. запятой
x F		Умножение двух чисел с Фиксиров. запятой
: F		Деление двух чисел с фиксированной запятой
+ C		Сложение двух чисел с плавающей запятой
- G		Вычитание двух чисел с плавающей запятой
x G		Умножение двух чисел с плавающей запятой
: G		Деление двух чисел с плавающей запятой

Арифметические операции выполняются над содержимым аккумуляторов 1 и 2 (смотри список операций. Раздел 8). Затем результат помещается в аккумулятор 1. Арифметические регистры изменяются при арифметических операциях следующим образом:

<АКК 1> : = результат
<АКК 2> : = <АКК 3>
<АКК 3> : = <АКК 4>
<АКК 4> : = <АКК 4>

Предыдущее содержимое аккумулятора <АКК 2> теряется.

o Вызов блоков

Операция	Параметр	Функция
SPA		Переход безусловный
SPB		Переход по условию (зависит от VKE)
0B	1... 39	к организационному блоку [1])
OB	40...255	к системной программе специальной функции[1])
PB	1...255	к программному блоку
FB	1... 255	к Функциональному блоку
A DB	1... 255	Вызов блока данных
BE		Конец блока
BEB		Конец .блока условный (зависит от VKE)
BEA		Безусловный конец блока

o Пустые операции

Операция	1 Параметр	Функция
NOP, NOP BLD	0 1 0...255	Пустая операция Пустая операция Команды Формирования изображения на экране программатора (не интерпретируется как команда CPU)

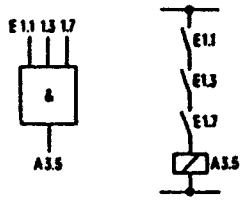
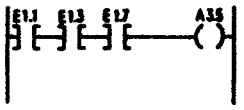
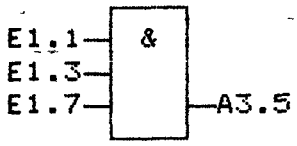
o Команда остановки

Операция	Параметр	Функция
STP		CPU переходит в состояние "СТОП"

Примеры программирования логических функций, таймеров, счетчиков и функций сравнения

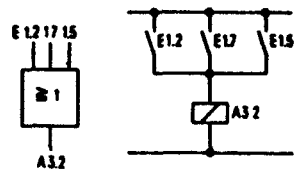
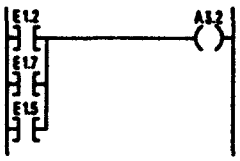
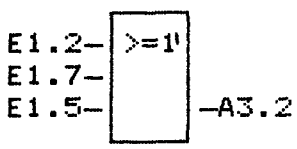
o Логические функции

Операция "И"

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E1.1 U E1.3 U E1.7 = A3.5		

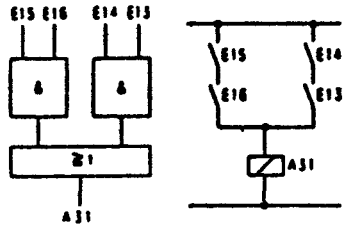
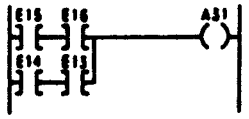
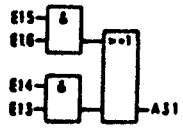
На выходе A3.5 индицируется "1", если на всех входах одновременно присутствует "1" . На выходе A3.5 "O", если хотя бы на одном из входов "O". Количество опросов и последовательность программирования произвольны

Операция "ИЛИ"

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	O E1.2 O E1.7 O E1.5 = A3.2		

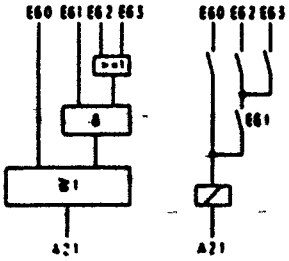
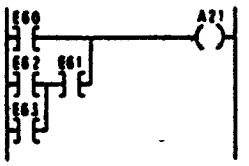
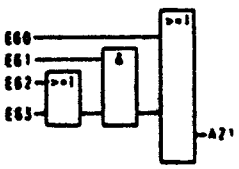
На выходе A3.2 "1", если по крайней мере на одном из входов "1". На выходе A3.2 "O", если на всех входах одновременно "O". Количество опросов и последовательность программирования произвольны.

Операция "И" перед "ИЛИ"

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E1.5 U E1.6 O U E1.4 U E1.3 = A3.1		

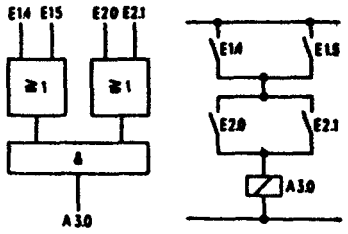
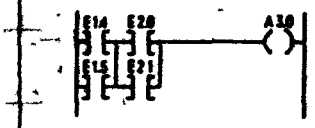
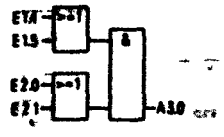
На выходе A3.1 "1", если по крайней мере одна из операций "И" выполнена (имеет на выходе "1").
На выходе A3.1 "О", если ни одна из операций "И" не выполнена.

Операция "ИЛИ-И-ИЛИ"

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	O E6.0 O U E6.1 U(O E6.2 O E6.3) = A2.1		

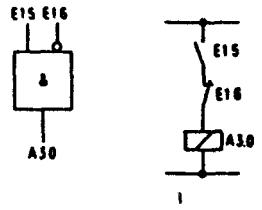
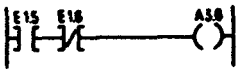
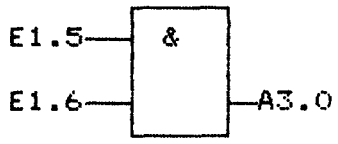
На выходе A2.1 "1", если на входе "E6.0 или на входе E6.1 и одном из входов E6.2 или E6.3 "1".
На выходе A2.1 "0" если на входе E6.0 "0" и операция "И" не выполнена.

Операция "ИЛИ" перед "И"

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U(O E1.4 O E1.5) U(O E2.0 O E2.1) = A3.0		

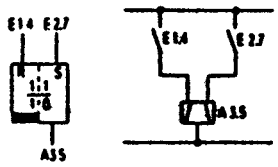
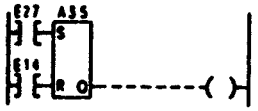
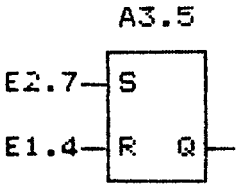
На выходе А3.0 "1", если обе операции "ИЛИ" выполнены. На выходе А3.0 "0", если не выполнена хотя бы одна операция "ИЛИ"

Опрос на 0-состояние сигнала

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E1.5 UN E1.6 = A3.0		

На выходе А3.О "1" только в том случае, если на входе Е1.5 "1" (нормально разомкнутый контакт активирован), а на входе Е1.6 "О" (нормально замкнутый контакт не активирован).

RS-триггер (элемент памяти.) для запоминания сигналов

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E2.7 S A3.5 U E1.4 R A3.5		

Состояние сигнала "1" на входе E2.7 вызывает установку элемента памяти.

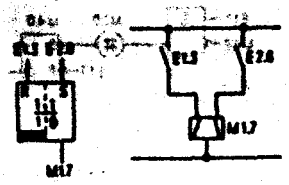
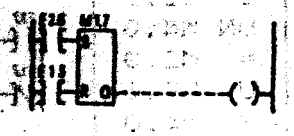
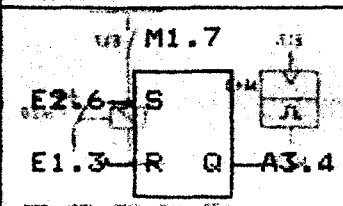
Если состояние сигнала на входе E2.7 изменится на "О", то это не повлияет на состояние RS-триггера, т.е. запоминается предыдущее состояние.

Появление "1" на входе E1.4 приводит к сбросу RS-триггера. Если состояние сигнала на входе E1.4 изменится на "О", то это не повлияет на состояние триггера, т.е. запоминается предыдущее состояние.

При одновременном поступлении "1" на оба входа — установки (E2.7) и сброса (E1.4) приоритет у последнего опроса, т.е. в данном случае у сброса (E1.4).

Команда MOP О используется в том случае, если программу потребуется представлять в КОР или FUP на PG675/PG670. При программировании RS—триггера в КОР или FUP эта операция заносится в программу автоматически.

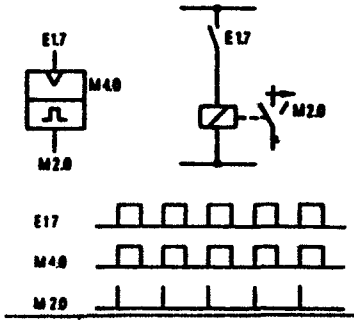
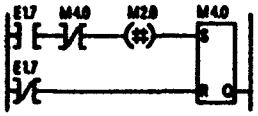
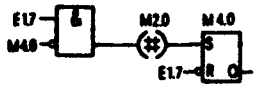
RS—элемент памяти с маркером

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E2.6 S M1.7 U E1.3 R M1.7		

Состояние сигнала "1" на входе E2.6- вызывает установку элемента памяти. Если состояние сигнала на входе E2.6 изменится на "0", то это не повлияет на состояние RS-триггера, т.е. запоминается предыдущее состояние.
Появление "1" на входе E1.3 приводит к сбросу RS—триггера. Если состояние сигнала на входе E1.3 изменится на "О", то это не повлияет на состояние триггера, т.е. запоминается предыдущее состояние.

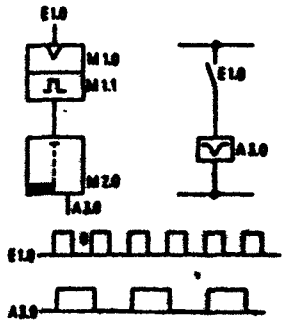
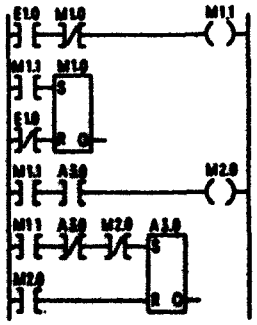
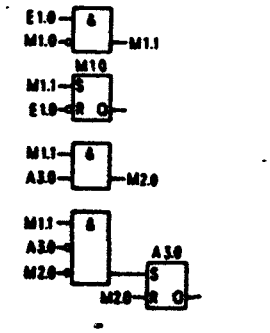
Если сигнал установки (вход E2.6) и сигнал сброса (вход E1.3) появляются одновременно, имеет приоритет операция, запрограммированная последней (в данном случае U E1.3) - сброс имеет приоритет.

Эквивалентная схема импульсного контакта
(анализ положительного фронта)

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E1.7 UN M4.0 = M2.0 U M2.0 S M4.0 UN E1.7 R M4.0		

При любом положительном фронте на входе E1.7 выполняется операция "И" (U E1.7 и UN M4.0), и результатом логической операции VKE=1 устанавливаются маркеры M4.0 (маркер фронта) и M2.0. При следующем цикле обработки программы операция "И" (U E1.7 и UN M4.0.) не выполняется, т.к. маркер M4.0 установлен. Маркер M2.0 возвращается в исходное состояние. Таким образом, маркер M2.0 только во время одного единственного цикла программы имеет состояние "1".

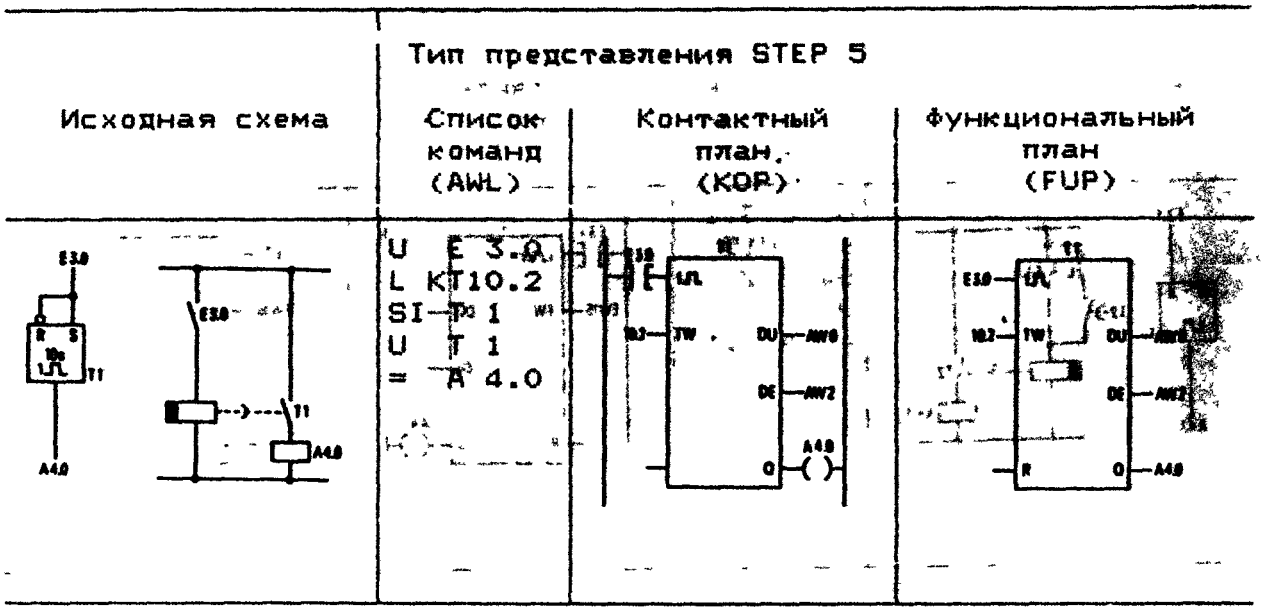
Делитель частоты .на 2 (Т-триггер)

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E1.0 UN M1.0 = M1.1 U M1.1 S M1.0 UN E1.0 R M1.0 U M1.1 U A3.0 = M2.0 U M1.1 UN A3.0 UN M2.0 S A3.0 U M2.0 R A3.0		

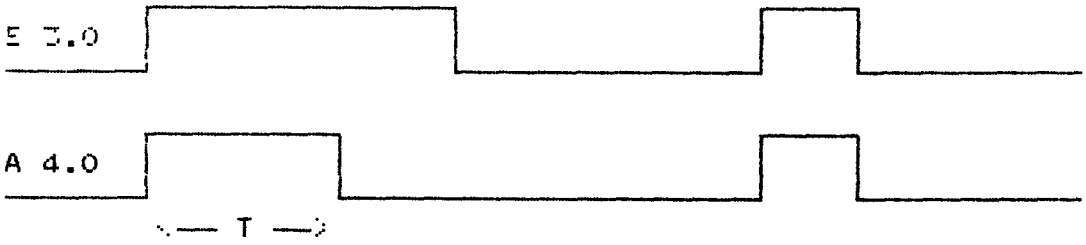
Делитель частоты на 2 (выход A3.O) меняет свое состояние при каждом положительном Фронте сигнала на входе E1.0. Поэтому на выходе элемента памяти частота сигнала в два раза меньше, чем на входе.

o Функции таймера

Импульс



Таймер запускается во время первого же цикла обработки программы, если результат логической операции (VKE) = 1. Таймер не меняет своего состояния, пока VKE=1, и не истекло запрограммированное время.
Таймер сбрасывается в "О", когда VKE становится равным "О" или
истекло время, на которое таймер запрограммирован.



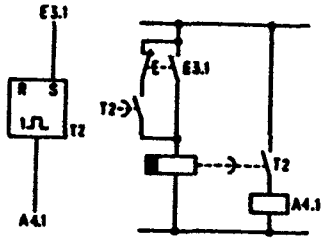
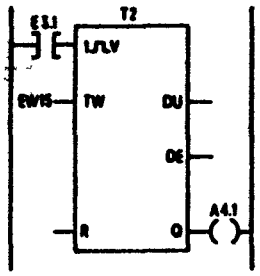
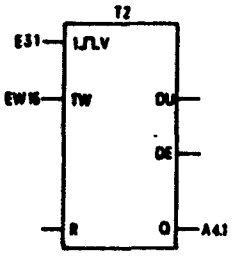
Значение KT 10.2:
Таймер, загружается заданным значением (10)

Чисто справа от точки указывает на масштаб времени:

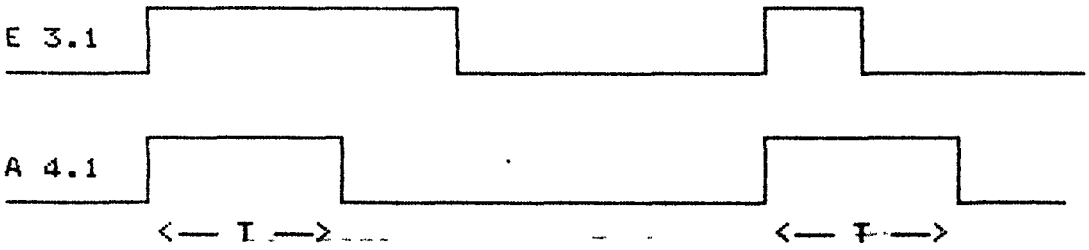
O = 0.01 сек
1 = 0.1 сек
2 = 1 сек
5 = 10 сек

Выходы DU и DE - цифровые выходы ячейки таймера. С выхода DU выдается состояние таймера в двоичном виде, а DE - в двоично-десятичном виде с базой времени.

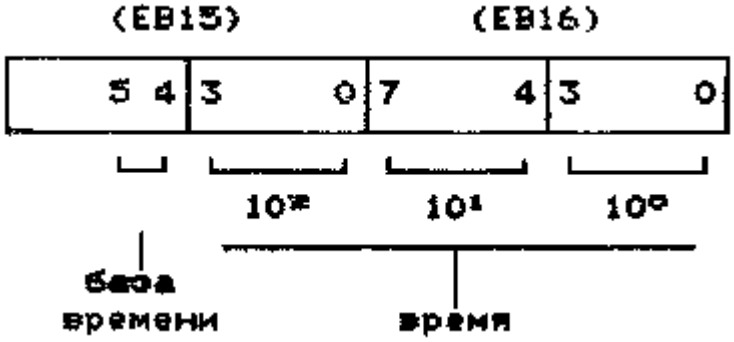
Удлиненный импульс

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E 3.1 L EW15 SV T 2 U T 2 = A 4.1		

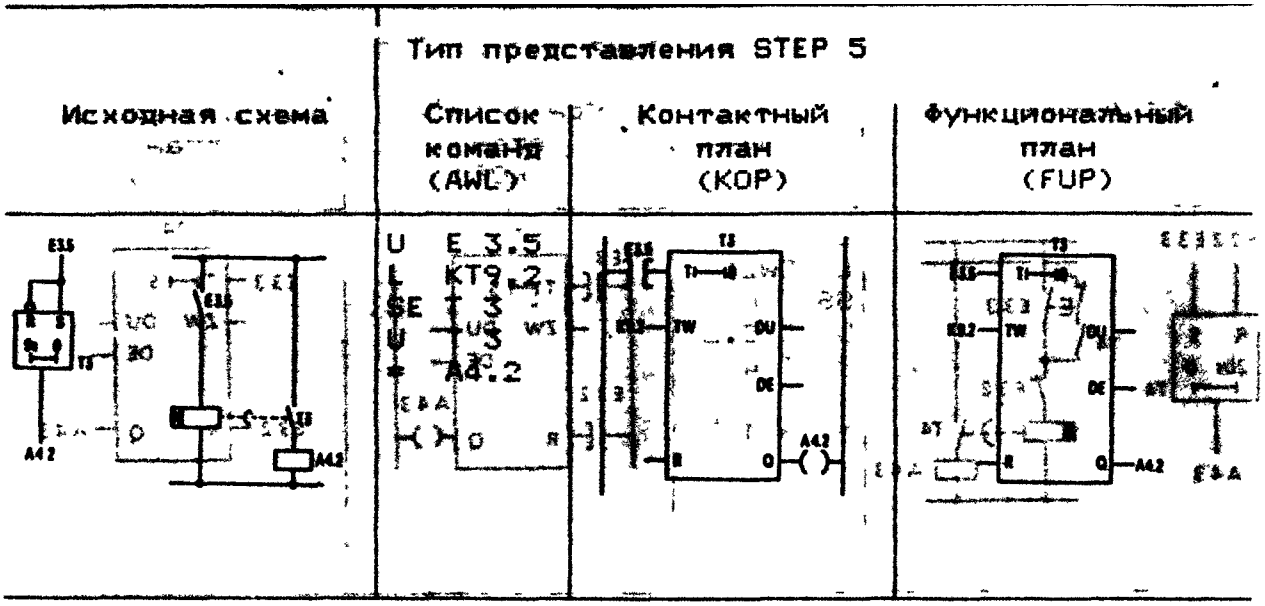
Таймер запускается во время первого же цикла обработки программы, если результат логической операции (VKE) =1. Таймер не меняет своего состояния, пока не истекло запрограммированное время, даже если VKE стал равен "О". Таймер сбрасывается в "О" по истечении времени, на которое он запрограммирован.



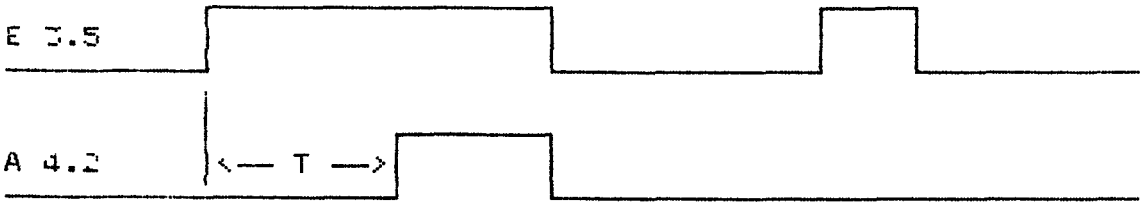
EW 15:
Установка значения величины времени с помощью заданного в двоично-десятичном коде (BCD) значения операндов E, A, M или D (в примере — слово входов 15: EW15)



Задержка включения



Таймер запускается во время первого же цикла обработки программы, если результат логической операции (VKE) =1, но он не устанавливается в "1" до тех пор, пока не истечет время, на которое запрограммирована задержка включения. Таймер сбрасывается в "О", когда VKE устанавливается в "О".

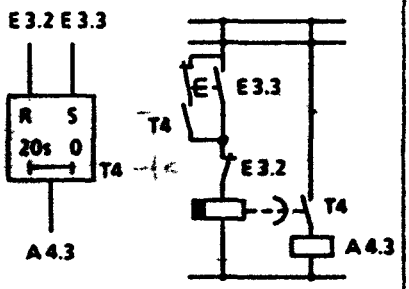
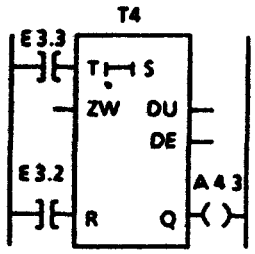
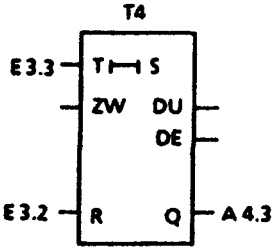


KT 9.2:

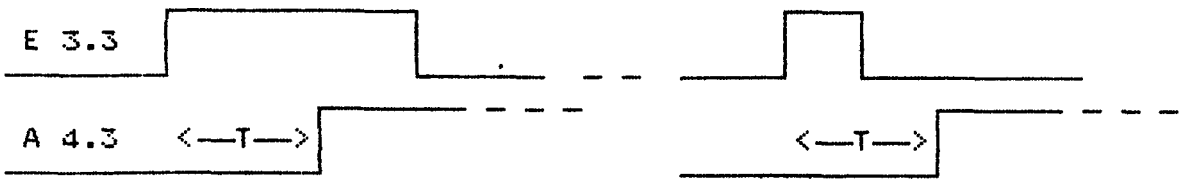
Таймер загружается заданным значением (9 сек). Число справа от точки указывает на базу времени:

0 = 0.01 сек
1 = 0.1 сек
2 = 1 сек
3 = 10 сек

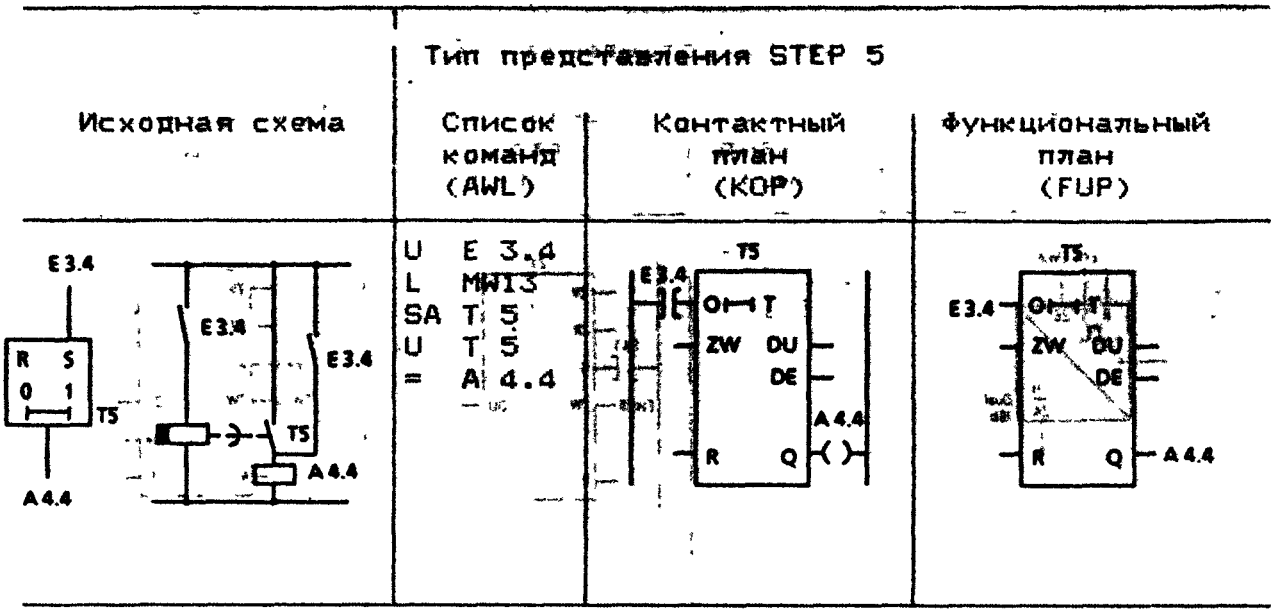
Задержка включения с запоминанием

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E 3.3 L DW21 SS T 4 U E 3.2 R T 4 U T 4 = A 4.3		

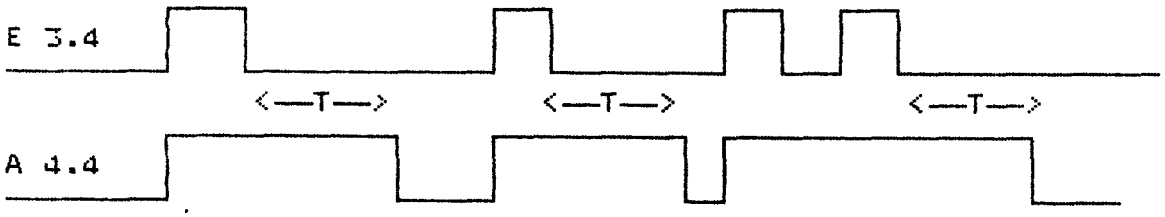
По положительному фронту сигнала на входе E3.3 начинается отсчет времени, на которое запрограммирована задержка включения. По истечении этого времени таймер устанавливается в "1". Таймер сбрасывается в "О" только функцией R T.



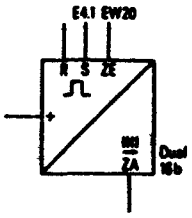
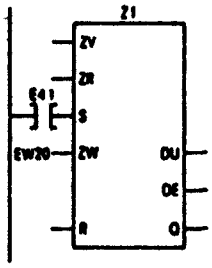
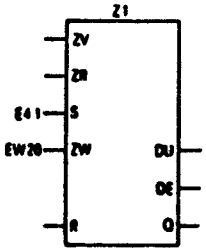
Задержка выключения



Таймер устанавливается в "1" во время первого же цикла обработки программы положительным фронтом сигнала на входе E3.4. По отрицательному фронту сигнала на входе E3.4 начинается отсчет времени, на которое запрограммирована задержка выключения. По истечении этого времени таймер сбрасывается в "О".



Установка счетчика

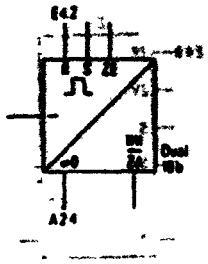
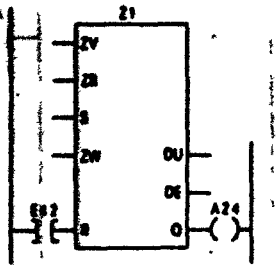
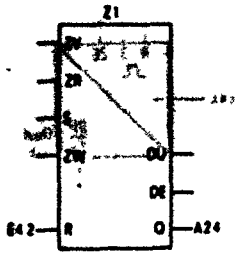
Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E 4.1 L EW20 S Z 1		

Счетчик устанавливается по входу S во время первого же цикла обработки программы и в течение этого цикла не меняет своего состояния независимо от VKE. Счетчик перезапускается в начале следующего цикла, если к его началу VKE=1.

Указание для работы с положительным фронтом содержится в слове счетчика, которым может быть слово входов EM, слово выходов AM, слово маркеров MM или слово данным DM. В слове счетчика задано также значение счета. Слово счетчика задается в BCD-коде и имеет Формат 16 бит.

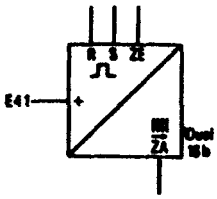
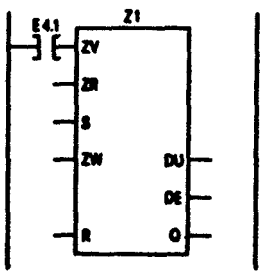
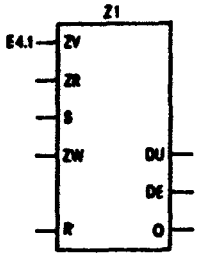
Выходы DU и DE - цифровые выводы ячейки счетчика. С выхода DU выдастся состояние счетчика в двоичном виде, а DE - в двоично-десятичном виде.

Сброс счетчика

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E 4.2 R Z 1 U Z 1 = A 2.4		

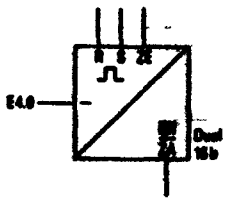
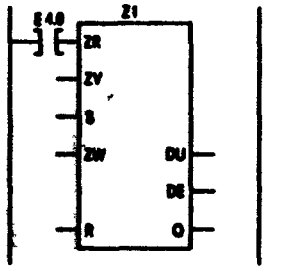
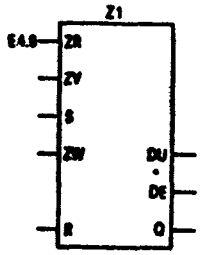
Счетчик сбрасывается по входу R, если VKE = 1
Счетчик после этого остается без изменений; даже если результат логической операции становится равным "О" на входе R.

Прямой счет

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E4.1 ZV Z 1		

Значение счетчика увеличивается на 1 при поступлении на счетный вход ZV положительного фронта сигнала от операнда, запрограммированного перед оператором ZV. Необходимые для работы с положительными Фронтами указания содержатся в слове счетчика. Используя два различных маркера для работы со счетными входами ZV и ZR, счетчик можно использовать для прямого и обратного счета.

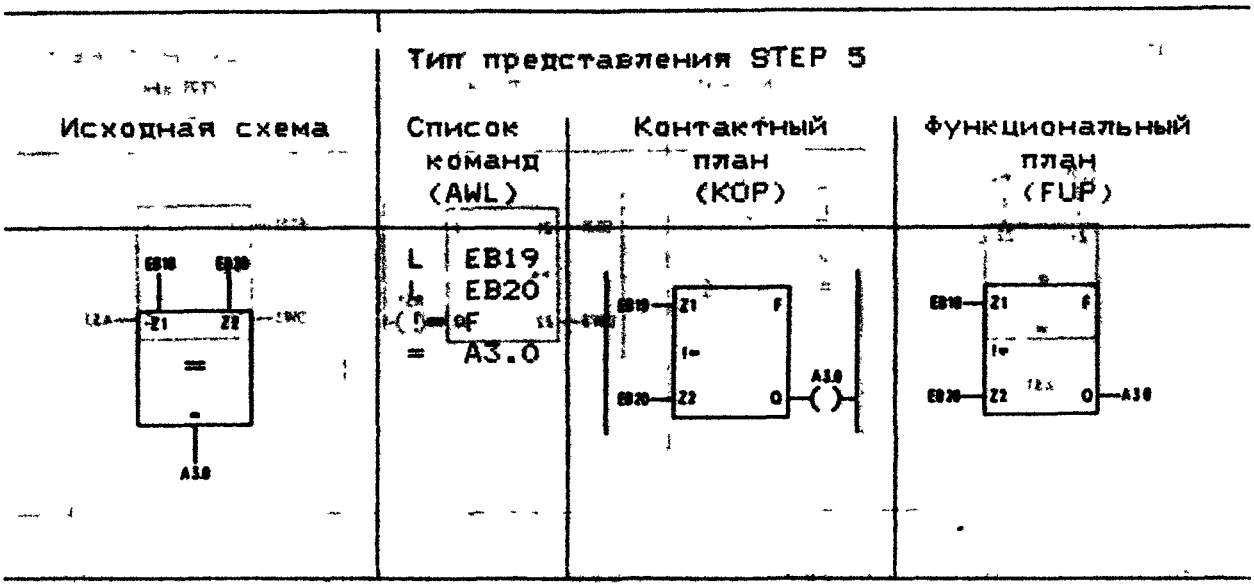
Обратный счет

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E4.0 ZR Z 1		

Значение счетчика уменьшается на 1 при поступлении на счетный вход ZR положительного фронта сигнала от операнда, запрограммированного перед оператором ZR. Необходимые для работы с положительными фронтами метки содержатся в слове счетчика. Используя два различных маркера для работы со счетными входами ZV и ZR, счетчик можно использовать для прямого и обратного счета.

o Функции сравнения

Сравнение на равенство



Операнд, заданный первым, сравнивается с последующим операндом в соответствии с функцией сравнения (Z1=Z2). По результату сравнения выставляется VKE:
VKE = 1, если операнды равны (содержимое EB19 равно содержимому EB20)

АКК 1-L = АКК 2-L

VKE = O, если операнды не равны.

АКК 1-L ≠ АКК 2-L

Коды условий ANZ1 и ANZO устанавливаются, как описано в Разделе 6.1 .

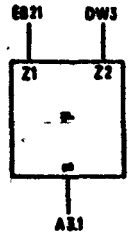
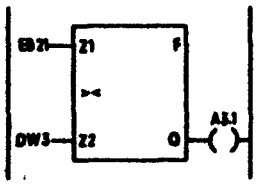
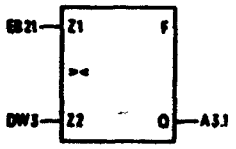
O	EB19	АКК 2-L
O	EB20	АКК 1-L

АКК 2-H и АКК 1-H остаются без изменений во время сравнения чисел с фиксированной запятой.

При сравнении чисел с фиксированной запятой (!=F) и чисел с плавающей запятой (!=G) сравнивается полное содержимое аккумуляторов АКК 1 и АКИ 2 (32 бита).

При сравнении учитывается численное представление операндов, т.е. содержимое аккумуляторов АКК 1-L и АКК 2-L интерпретируется как числа с фиксированной запятой.

Сравнение на неравенство

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	L EB21 L DW3 >< F = A3.1		

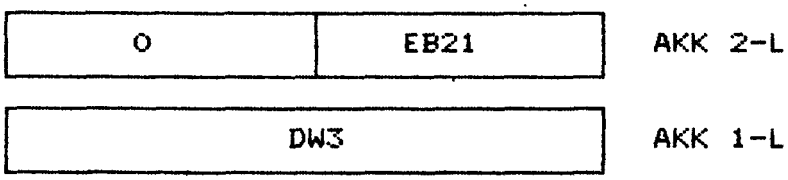
Операнд, заданный первым, сравнивается .с последующим операндом в соответствии с функцией сравнения (Z1><Z2). По результату сравнения выставляется VKE:
VKE = 1, если операнды не равны (содержимое EB21 не равно содержимому EB22)

АКК 1-L ≠ АКК 2-L

VKE = O, если операнды равны.

АКК 1-L = АКК 2-L

Коды условий ANZ1 и ANZO устанавливаются, как описано в Разделе 6.1.



АКК 2-H и АКК 1-H остаются без изменений во время 16-битового сравнения чисел с фиксированной запятой.

При сравнении 32-битовых чисел с фиксированной запятой сравнивается полное содержимое аккумуляторов АКК 1 и АКК 2 (включая АКК 1-H и АКК 2-H).

Это применимо и к сравнению на больше, больше или равно, меньше, меньше или равно (смотри список команд).

При сравнении учитывается численное представление операндов, т.е. содержимое аккумуляторов АКК 1-L и АКК 2-L интерпретируется как числа с фиксированной запятой.

6.3 Дополнительный набор операций.

В противоположность другим блокам, функциональные блоки могут программироваться с помощью расширенного. набора операций языка STEP-5. Полный набор операций для использования в функциональных блоках состоит основного и расширенного наборов операций.

Основной набор, расширенный набор операций и системные функции вместе составляют полный набор операций языка программирования STEP-5.

С помощью системных функций можно вмешиваться в работу системной программы; память может быть перезаписана в любой момент, и может быть изменено содержимое рабочих регистров центрального процессора. Поэтому, системные функции следует использовать с повышенным вниманием.

Системные функции представляются на следующих страницах

Программа функциональных блоков может быть представлена только в виде AWL. Поэтому они не могут быть запрограммированы в графическом виде представления (KOP и FUP).

Следующее описание показывает дополнительные операции и системные функции, которые могут выть использованы только в функциональных блоках. Кроме того, показана возможная комбинация операций дополнительного набора с фактическими операндами.

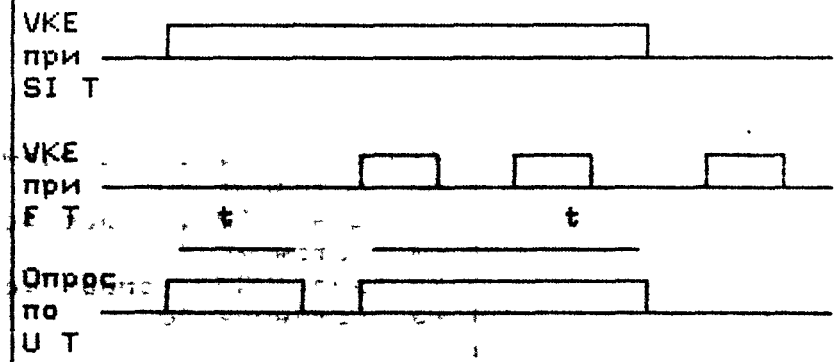
о Двоичная логика

Операция	Описание
U = ____	Функция "И", опрос формального операнда на состояние "1".
UN = ____	Функция "И", опрос формального операнда на состояние "0".
O = ____	Функция "ИЛИ", опрос формального операнда на состояние "1"
ON = ____	Функция "ИЛИ", опрос формального операнда на состояние "0"
^	Вставить формальный операнд
	В качестве актуального операнда допустимы двоично-адресованные входы/выходы и маркеры (параметр типа E,A; тип данных:BI), таймеры и счетчики (тип параметра: T, Z).

o Функции запоминания

Операция	Описание
S = ____ RB = ____ = = ____	Установка (двоичная) формального операнда. Сброс (двоичный) формального операнда. Вывод результата соединения по формальному операнду.
	Вставить формальный операнд
	В качестве фактического операнда допустимы двоично-адресованные входы,выходы,данные и маркеры (параметр типа E,A; тип данных:BI)

o Функции таймеров и счетчиков

Операция	Описание
FR T 0...127	Деблокировка таймера для запуска Операция выполняется только по переднему фронту результата логической операции VKE. Таймер перезапускается, если VKE = "1" во время операции запуска
	
FR Z 0...127	Деблокировка счетчика Операция выполняется только по переднему фронту результата логической операции VKE. Счетчик устанавливается (счет "вперед" или "назад") если результат логической операции VKE=1 в соответствующей операции
FR = ____	Деблокирование формального операнда для нового пуска (см. "Описание FR T или FR Z" в зависимости от формального операнда, тип параметра: T,Z).
RD = ----	Сброс (цифровой) формального операнда (тип параметра: T,Z).

Операция	Описание
SI = ———	Запуск таймера, заданного формальным операндом, качестве импульса с длительностью, .закодированной в АКК (тип параметра T)
SE = ———	Запуск таймера, заданного формальным операндом в качестре задержки включения с длительностью, закодированной в аккумуляторе (тип параметра: T).
SVZ = —--	Запуск таймера, заданного формальным операндом, в качестве удлиненного импульса с длительностью, закодированной в аккумуляторе или установка счетчика, заданного формальным операндом, с указанным в последствии счетным значением (тип параметра: T, Z).
SSV = ———	Запуск таймера, заданного формальным операндом в качестве запоминаемой задержки включения со значением, закодированным в аккумуляторе, или счет вперед счетчика, заданного формальным операндом (тип параметра: T, Z).
SAR = ———	Запуск таймера, заданного формальным операндом, в качестве задержки включения с длительностью, закодированной в аккумуляторе, или счет назад счетчика, заданного формальным операндом (тип параметра: T, Z).
	Вставить формальный операнд
	Таймеры и счетчики можно использовать в качестве актуальных операндов. Исключения: SI и SE (только таймеры). Значение таймера или счетчика может быть установлено как с помощью основного набора операций, так и с помощью формальных операндов следующим образом: Установка значения таймера или счетчика и виде значения в коде BCD с помощью операндов EW, AW, MW, DW, указанных в качестве Формальных операндов (класс параметра E, тип параметра W) или с помощью константы (параметр типа D, тип данных KT, KZ).

Примеры

Вызов Функционального блока	Программа в Функциональном блоке	Выполняемая программа
: SPA FB20.3 ИМЯ : BEISPIEL ANNA : E 10.3 BERT : T 17 HANS : A 18.4	: U =ANNA : L KT 010.2 : SSV =BERT : U: =BERT *:* = =HANS	U E 10.3 L KT 010.2 SS T 17 U T 17 = A 18.4
: SPA FB204 ИМЯ : BEISPIEL MAX : E 10.5 IRMA : E 10.6 EVA : E 10.7 DORA : Z 15 EMMA : M 58.3 ,	: 8 U =MAXI : SSV =DORA : U =IRMA : SAR =DORA : U =EVA : L KZ100 : SVZ =DORA : UN =DORA : = =EMMA	U E 10.5 ZV Z 15 U E 10.6 ZR Z 15 U E 10.7 L KZ100 S Z 15 UN Z 15 = M 58.3
: SPA FB205 ИМЯ : BEISPIEL KURT : E 10.4 CARL : T 18 EGON : EW20 MAUS : M 100.7	: U =KURT : L =EGON : SVZ =CARL : U =CARL : = =MAUS	U E 10.4 L EW 20 SV T 18 U T 18 = M 100.7

о Операции загрузки и пересылки

Операция	Описание
L = ____	Загрузка формального .операнда. Значение, заданное в качестве формального операнда, загружается в аккумулятор (тип параметра E , A, T, Z; тип данных BY,M,D).
LC = ____	Кодированная загрузка формального операнда значение временной или счетной ячейки в качестве формального операнда загружается в аккумулятор в BCD коде (тип параметра: T, Z)
LW = ____	Загрузка набора битов формального операнда. Формальный операнд в битовом виде загружается в аккумулятор (тип параметра: D; тип данных: KF, KH, KM, KY, KC, KT, KZ)
LD = ____	Загрузка набора битов формального операнда. Формальный операнд в битовом виде загружается в аккумулятор (тип параметра: D; тип данных: KG)
T = ____ ↑ Вставить формальный операнд	Пересылка в формальный операнд. Содержимое аккумулятора пересылается в формальный операнд (тип параметра; A, E; тип данных: BY, W, D).

В качестве фактических операндов разрешены операнды, соответствующие основному набору операций. Допустимы данные в двоичном (KM) или шестнадцатиричном (KH) представлении, двухбайтовом целочисленном (KY), со знаком (KC), с фиксированной запятой (KF), значения таймеров (KT) и счетчиков (KZ). При LD допустимы также данные в формате с плавающей запятой (KG).

Операция	Параметры	Описание
L BA	0...255	Загрузка слова в аккумулятор АКК 1 из области "интерфейса данных".
L BS [5])	0...255	Загрузка слова в аккумулятор АКК 1 из области "системных данных".
T BA [1)]	0...255	Передача слова из аккумулятора АКК 1 в область "интерфейсных данных".
T BS [5)]	0…255	Передача слова из аккумулятора АКК 1 в область "системных данных".
LIR [l)]	0...15	Загрузка регистра (косвенная): значение содержимого .слова из памяти[2)], адресованного значением; из АКК 1
TIR [1)]	0...15	Передача содержимого регистра (косвенная): в слово памяти[2)], адресованное значением из АКК 1
TNB [1)]	0...255	Передача блока по байтам: источник - в АКК 2, назначение - в АКК 1[3)].
TNM [1)]	0...255	Передача блока по словам: источник - в АКК 2, назначение - в АКК 1[4)].

[1)] Системная Функция

[2)] LIR, TIR
a) Регистры 4,7,8,13,14 не представимы -> NOR
б) Регистры 5,6,15: TIR загружает регистры, адресованные словом АКК 1-L. LIR загружает АКК 1-L в адресованный регистр. Нет передачи из/в память!
c) Если указана адресная область FFOOH...FFFFH -> NOP.
d) Обращение к 8-битовой памяти:
TIR: старший байт регистра теряется.
LIR; в старший байт регистра записывается FFH.

[3)] Команда TNB ограничена передачей следующих типов блоков:
a) Байт из области памяти BOOOH...EFFFH -> байт памяти по адресу BOOOH...EFFFH
b) Байт из области памяти BOOOH…EFFFH -> адресная область входов/выходов FOOOH...FDFFH
c) Байт из области входов/выходов FOOOH...FOFFH -> байт памяти по адресу BOOOH...EFFFH

[4)] Команда TNW ограничена передачей следующих типов блоков:
а) Память пользователя -> память пользователя
б) Память пользователя -> байт памяти, адресованный - с BOOOH по EFFFH
с) Байт памяти, адресованный с BOOOH по EFFFH-> память пользователя

[5)] Системные данные 3 и 4 содержат информацию о неисправностях, если они возникают (смотри раздел 10). Пользователем с помощью команды T BS могут быть записаны данные только в системные слова 60...63, и они не используются системной программой.

o Арифметические операции

Операция	Описание
ENT	Ввод данных в память арифметики Результат команды ENT загружается в АКК 3 и АКК 4, которые в арифметических операциях используются тоже АКК 4: = АКК 3 АКК 1: = АКК 1 АКК 3: = АКК 2 АКК 2: = АКК 2

Старое содержимое АКК 4 стирается.

Пример

должно быть вычислено выражение: (30 + 3x4)/6 =7

	АКК 1	АКК 2	АКК 3	АКК 4
Содержимое аккумулятора до выполнения арифметической операции	a	Ь	c	d
L KF 30	30	a	c	d
L KF 3	3	30	c	d
ENT	3	30	30	c
L KF 4	4	3	30	c
* F	12	30	c	c
+ F	42	c	c	c
L KF 6	6	42	c	c
/ F	7	c	c	c

Операция	Параметры	Описание
ADD BF	-127...+127	Прибавление байта константы (с фиксированной запятой) к содержимому аккумулятора АКК 1)
ADD KF	-32768...+ 32767	Прибавление слова константы (с фиксированной запятой) к содержимому аккумулятора АКК 1)
ТАК		Обмен содержимым аккумуляторов (АКК 1 и АКК 2)

[1]) Содержимое аккумуляторов 2,3 и 4 не изменяется

[2]) Содержимое аккумуляторов 3 и 4 не изменяется

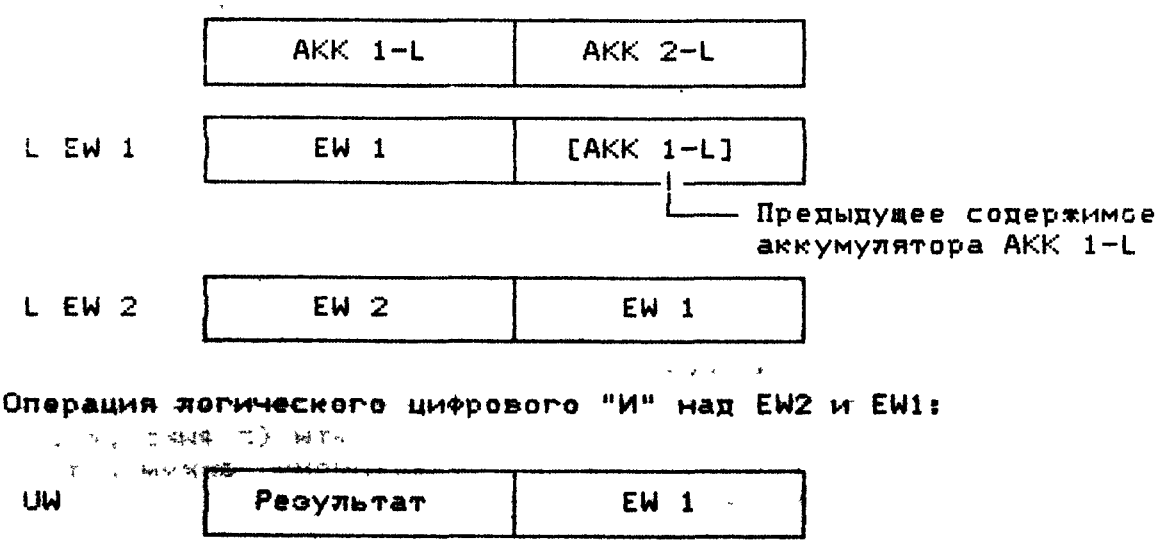
o Цифровые логические операции

Операция	Описание
UW	Операция цифрового логического "И" над содержимым аккумуляторов АКК 1 и АКК 2
OW	Операция цифрового логического "ИЛИ" над содержимым аккумуляторов АКК 1 и АКК 2
XOW	Операция цифрового логического "ИСКЛ.ИЛИ" над содержимым аккумуляторов АКК 1 и АКК 2

Аккумуляторы 3 и 4 не используются, а используются коды условий ANZ1 и ANZO (смотри Раздел 6.1).

При использовании двух операций загрузки аккумуляторы могут быть загружены соответствующими операндами. Затем содержимое обоих аккумуляторов может быть использовано для цифровых логических операций.

Пример



Организационные ФУНКЦИИ

о Функции перехода

Назначение условного или безусловного перехода указывается символически. (максимально 4 знака, начинающиеся с буквы). Символический параметр команды перехода идентичен символическому адресу команды, к которой осуществляется переход. При программировании следует принимать во внимание, что абсолютное состояние перехода не может превышать ё127 слов и что команда STEP-5 может состоять из более чем одного слова. Переход может выполняться только внутри одного блока; переход между сегментами не допустим.

Примечание Команда перехода и назначение должны быть в одном сегменте. Для сегмента в качестве назначения можно указывать только один символический адрес. Эти условия не применяются для команды перехода SPR, для которой в качестве параметра указывается абсолютный адрес расстояния перехода.

Операция ,	Описание
SPA = адр	Безусловный переход Безусловный переход выполняется при всех условиях
SPB = адр	Переход по условию Условный переход выполняется, если VKE = 1. Если VKE=0, команда не выполняется, и результат логической операции VKE устанавливается в 1.
SPZ = адр	Переход по условию: ANZ1, ANZO Переход выполняется, если только ANZ1=0 и ANZO=0. Результат логической операции не изменяется.
SPN = адр	Переход по условию: ANZ1, ANZO Переход выполняется, если ANZ1 ≠ ANZO. Результат логической операции не меняется.
SPP = адр	Переход по условию: ANZ1, ANZO Переход выполняется, если только ANZ1=1 и ANZO=0. Результат логической операции не изменяется.
SPM = адр	Переход по условию: ANZ1, ANZO Переход выполняется, если только ANZ1=0, и ANZO=1. Результат логической операции не изменяется.
SPO = адр	Переход по условию переполнения: Переход выполняется, если только условие переполнения OV=1. Если переполнения нет, (OV=0), переход не выполняется. Результат логической операции не изменяется. Если происходит превышение допустимых границ результирующего числа при выполнении арифметических операций, происходит переполнение.

Операция	Описание
SPS = адр SPR [1] -32768...+32767	Переход, если установлен код условия OS (запоминание переполнения.) (OS=1). Переход к системному матобеспечению.

адр = символический адрес (максимально 4 символа)

о Функции сдвига

Операция	Описание
SLW O...15 SRW O...15 SLD O...32 SVM O...15 SVD O...32 RLD O...32 RRD O...32 ↑ Параметр	Сдвиг влево (справа заполнение нулями) Сдвиг вправо (слева заполнение нулями) Сдвиг двойного слова влево (справа заполнение нулями) Сдвиг вправо со знаком Сдвиг двойного слова вправо со знакам (знак устанавливается слева) Циклический сдвиг влево Циклический сдвиг вправо

При операциях сдвига используется только аккумулятор АКК 1. Параметр команды определяет, на сколько позиций производится сдвиг содержимого аккумулятора. В операциях SLM, SRW, SVW используется только младшее слово аккумулятора, в операциях SLD, SVD, RLD, RRD используется полное содержимое аккумулятора (32 бита).

Операции, сдвига выполняются безусловно. Последний сдвинутый бит может быть обработан с помощью команд перехода. Происходит установка кодов ANZO и ANZ1 (смотри Раздел 6.1).

Если последний сдвинутый бит равен нулю используются команда перехода SP2. Если использовать команду перехода SPN, переход выполняется, если сдвинутый бит был равен 1.

Прим^рц

[1] Системная функция

Примеры

Программа STEP	Содержимое слова данных
: L DW52	H = 14AF
: SLW 4	
:T DW53	H = 4AF0

Программа STEP-5	Содержимое аккумулятора (16-ричн.)
: L ED0	2348 ABCD
: SLW 4	2348 BCDO
: SRW 4	2348 0BCD
: SLD 4	3480 BCD0
: SLW 4	3480 FBCD
: SVD 4	0348 0FBC
: RLD 4	3480 FBC0
: RRD 4	0348 0FBC
: BE	

o Операции преобразования

Операция	Описание
KEW	Формирование первого дополнения аккумулятора 1 (16 бит)
KZW	Формирование двоичного дополнения аккумулятора 1 (16 бит)
KZD	Формирование двоичного дополнения аккумулятора 1 (32 бит)
DEF	Преобразование числа с фиксированной запятой (16 бит) из кода BCD в двоичный вид
DUF	Преобразование числа с фиксированной запятой (16 бит) из двоичного вида в код BCD
DED	Преобразование числа с фиксированной запятой (32 бит) из кода BCD в двоичный вид
DUD	Преобразование числа с фиксированной запятой (32 бит) из двоичного вида в код BCD
FDG	Преобразование числа с фиксированной запятой (32 бит) в число с плавающей запятой
GFD	Преобразование числа с плавающей запятой (.32 бит) в число с фиксированной запятой

Примеры

Содержимое слова данных 64 инвертируется и хранится в слове данных 78.

Программа STEP-5		Содержимое слова данных
: L	DW64	BM = 0011111001011011
: KEW		
: T	DW78	BM = 1100000110100100

Содержимое слова данных 207 интерпретируется в виде числа с фиксированной запятой и записывается в слово данных 51 с противоположным знаком.

Программа STEP-5		Содержимое слова данных
: L	DW207	F: + 51
: KWZ		
: T	DW78	F: - 51

о Декремент/Инкремент

Операция	Описание
D 1…255	Декремент
I 1…255 Параметр	Инкремент

Содержимое аккумулятора АКК 1 уменьшается (декремент) или увеличивается (инкремент) на число, указанное в качестве параметра. Операция безусловная. Операция ограничивается правым байтом (без переноса).

Пример

Программа STEP-5 •	Содержимое слова данных
: L DW7	H: 1010
: I 16	
: T DW8	H: 1020
: D 33	
: T DW9	H: 10FF

o Функции обработки

Операция	Описание
B DW0 ... 255 (операция.)	Обработка слова данных Следующая указанная операция будет скомбинирована с параметром, указанным в слове данных и затем выполнена.
B MW0 ... 254 (операция)	Обработка слова маркеров Следующая указанная операция будет скомбинирована с параметром, указанным в слове , маркеров и затем выполнена.
B = формальный операнд	Обработка формального операнда (параметр вида B) Могут быть замещены только операции A DB, SPA PB, SPA FB, SPA SB
BI [1) 2)]	Обработка, использующая Формальный операнд (косвенная). Номер формального операнда находится в аккумуляторе АКК 1.
B BS [1) 2)]	Выполнение команды из системной области данных (RS).

1) Системная Функция

2) Значение, которое в системных данных или в формальном операнде в функциональном блоке интерпретируется в качестве кода команды STEP-5, которая затем выполняется. Операция допустима как с командой B DW, так и с командой B MW. Только системные данные 60 ... 63 резервируются для пользователя и не используются системной программой.

Все операции, за исключением перечисленных ниже, могут комбинироваться с командой B DM или B MW:

- A DB0, A DB1,
- все двух- или трехсловные команды,
- операции с формальными операндами,
- SPA OBxx и SPB OBxx.

Программатор не проверяет допустимость операции.

Пример: ("Обработка слова данных")

Содержимое слов данных с DW20 по DW100 устанавливается в состояние "0". Индексный регистр для параметров слов данных – DW1.

	:L KF 20	Установка индексного регистра
	:T DW 1	
M001	:L KF 0	Сброс
	:B DW 1	
	:T DW 0	Инкремент индексного регистра
	:L DW 1	
	:L KF 1	
	:+ F	
	:T DW 1	
	:L KF 100	
	:<= F	
	:SPB = M001	Переход, если индекс в пределах
	…	Продолжение программы STEP-5

o Команды блокировки/деблокировки

Операция	Описание
AS	Блокировка прерывания от процесса
AF	Деблокировка прерывания от процесса

Функция "Блокировка/деблокировка прерываний" может быть использована, например, если во время выполнения программы обработки прерываний по времени необходимо подавить обработку прерываний от процесса.

7 Спец. функции операционной системы

7.1 Передача блоков данных (OB255)

Системная программа со специальной функцией "Передача блока данных инициализируется вызовом организационного блока OB255.

OB255 предполагает, что номер блока данных, который должен передаваться, находится в аккумуляторе. Передача может быть выполнена только ив 16-битовой области памяти пользователя в 8-битовый блок данных в области памяти RAM процессора (см. Раздел 6.1).

OB255 проверяет, существует или нет блок данных в памяти пользователя и передает его (с соответствующей коррекцией адреса входа в DB0) в пустой блок данных области памяти RAM.

DB0 управляется системной программой. Он содержит начальные адреса всех блоков.

Если указанный блок не существует в области памяти пользователя, или уже существует в области памяти RAM, процессор переходит в состояние STOP и выдает сообщение об ошибке SFF (см. Раздел 5.7). То же самое происходит, если для передачи блока данных в память RAM не хватает памяти.

При возникновении ошибки любого типа частичной передачи блока данных не происходит.

Примечание! Если должен быть передан блок данных, вызванный с помощью команды A DBx, он должен быть заново создан после передачи с DBx. Если это не сделано, то все дальнейшие, обращения к словам данных, используя операции загрузки и передачи, будут относиться к старому блоку данных. Передан может быть блок длиной до 256 слов (DW0...DW255).

7.2 PID-регулятор (OB 250 и OB 251)

Пользователь может иметь в S-процессоре контроллера 35 135U один или более PID-регуляторов.

Каждый регулятор должен быть инициализирован в стартовом организационном блоке, для передачи параметров используется блок данных.

Текущий алгоритм управления встроен в операционную систему и может быть только вызван пользователем. Блок данных используется в качестве интерфейса данных между алгоритмом управления и программой пользователя.

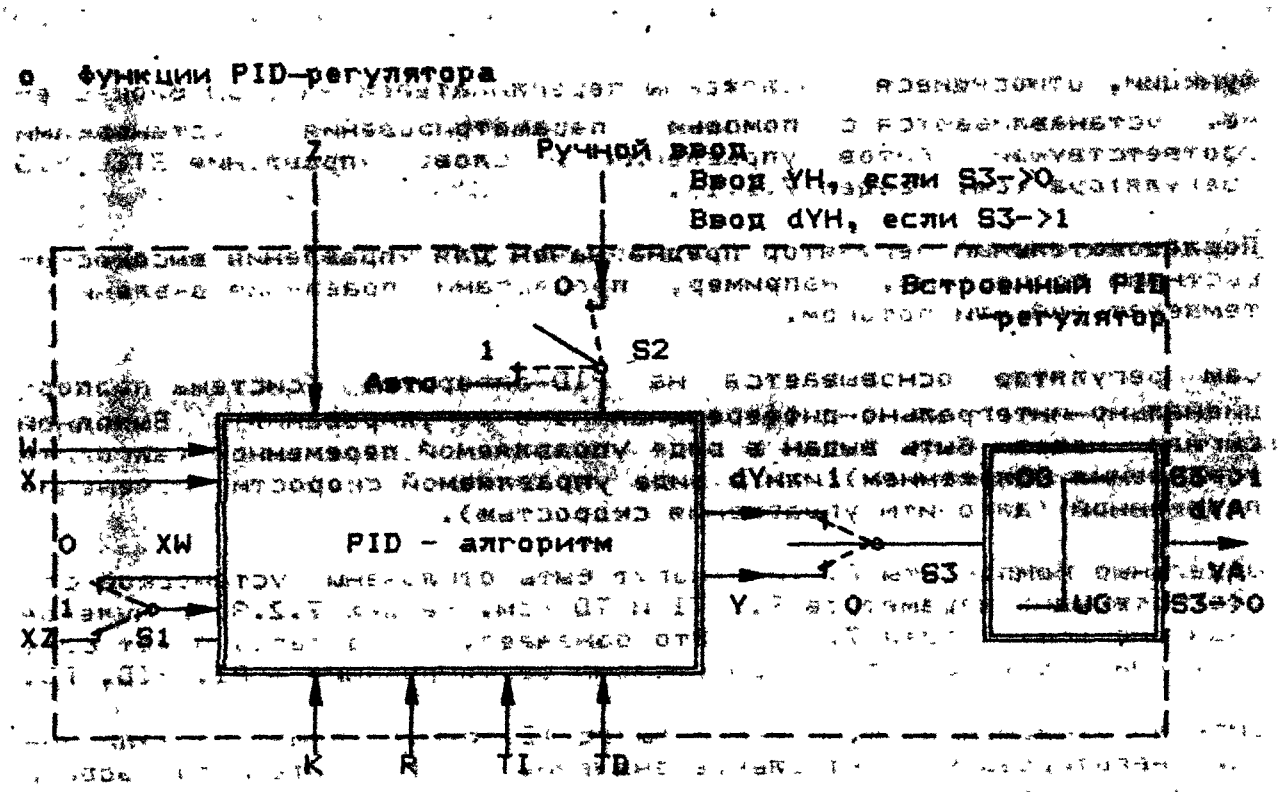


Рис.18 Блок-схема PID-регулятора.

Индекс K: K – опрос

Переключатель	Положение	Эффект
S1 STEU бит 1	0 1	Дифференциатор выполняет дифференциацию регулировки XWk Через XZ может быть подан другой сигнал
S2 STEU бит 0	0 1	Ручное управление Автоматическое управление
S3 STEU бит 3	0 1	Алгоритм управления положением Алгоритм управления скоростью
S4 STEU бит 4	0 1	С прямым управлением Без прямого управления

Функции, относящиеся *к* положению переключателей на этой блок-схеме, устанавливаются с помощью параметрирования установками соответствующих битов управления в слове управления STEU PID-регулятора (см. Раздел 7.2.1).

Последовательный регулятор предназначен для управления высокоскоростными процессами, например, процессами управления давлением, температурой или потоком.

Сам регулятор основывается на FID-алгоритме (система пропорционально-интегрально-дифференциального регулирования). Выходной сигнал может быть выдан в виде управляемой переменной (алгоритм управления положением) или в виде управляемой скорости изменения, переменной (алгоритм управления скоростью).

Отдельные компоненты P, I и D могут быть отключены установкой соответствующих параметров R, TI и TD (см. Раздел 7.2.8) в нулевое значение (см. Раздел 7.2.5). Это означает, что легко может быть получена требуемая структура системы регулирования - PI, PID, PD.

Сигнал отклонения XW, (через выход XZ) любая влияющая величина или инвертированное актуальное значение X могут быть использованы в блоке отклонений.

Для компенсации влияния изменений осуществляется управление скоростью подачи без контроля времени. Сигнал помехи Z, связанный с процессом, подается для алгоритма управления. При ручном управлении в эту точку подается предварительно устанавливаемая переменная YN (см таблицу значений управляющего слова STEU дальше в тексте).

Если требуется изменение направления, следует устанавливать отрицательное значение параметра K.

Если информация по управлению (dY или Y) на пределе, компонента I выключается автоматически для того, чтобы избежать ухудшения отклика регулирования.

Программа регулирования может содержать фиксированные значения или адаптивные (динамические) параметры (K, R, TI, TD). Они вводятся через установку памяти отдельных параметров.

7.2.1 PlD-алгоритм

PID регулятор основан на алгоритме управления скоростью, согласно которому производится вычисление соответствующего управляющего приращения dYk в отдельной точке во время t=k. TA по формуле:

$$dYk = K((XW_k - XW_{r-1})R + TA/TN \times XW_k + ½(TV/TA(XU_k - 2XU_{k-2} + XU_{k-2}) + dD_{k-1})) =$$

$$= K(dPW_k + dI_k + dD_k)$$

P-часть I-часть D-часть

$dXXX_k$: Изменение величины XXX во время t.

U может быть W или Z, в зависимости от того, XW или XZ подключено к отклоняющему входу. Выполняется следующее:

При подключении XW	При подключении XZs
$PW_k = W_k - X_k$	
$PW_k = XW_k - XW_{k-1}$	$PZ_k = XZ_k - XZ_{k-1}$
$QW_k = PW_k - PW_{k-1}$	$QZ_k = PZ_{k-1}$
$QW_k = XW_k - 2XW_{k-1} + XW_{k-2}$	$QZ_k = XZ_k - 2XZ_{k-1} + XZ_{k-2}$

$$dPW_k = (XW_k - XW_{k-1})R$$

$$dI = TI \cdot XW_k \qquad TI = \frac{TA}{TN}$$

$$dD_k = \frac{1}{2}(TD \cdot QU_k + dD_{k-1}) \qquad TD = \frac{TV}{TA}$$

Если управляемая переменная Y_k используется в качестве управляющего выхода во время t_k, то она формируется в соответствии с формулой;

$$Y_k = \sum_{m=0}^{m=1} dY_m$$

В большинстве проектируемых процессов управления можно допустить, что R=1, если требуется реакция P.

Величина R может быть использована для установки пропорциональной компоненты PID-регулятора.

7.2.2 Блоки данных для PID-регулятора

Специальные данные для регулятора вводятся с использованием блока данных (для инициализации и вызова PID-регулятора смотри Раздел 7.2.3).

Пользователь должен ввести следующие данные в передаваемый блок данных:
K, R, TI, TD, W, STEU, YH, BGOG, BGUG.
Структура передаваемого блока более детально описана ниже.

о Структура передаваемого блока данных

Номер слова	Имя	Вх/вых	Формат числа	Формат U370	Примечание
0					Зарезервировано
1	K	Вх	GP	KG	Коэффициент пропорциональности K>0 : положительное направление управления, т.е.изменение установки и изменяемой переменной в одинаковых направлениях K<0 : отрицательное направление управления; в пределах числа с плавающей запятой
3	R	вх	GP	KG	R параметр, обычно =1 для регуляторов с компонентой P; в пределах числа с плавающей запятой
5	TI	вх	GP	KG	TI = TA/TN; в пределах числа с плавающей запятой
7	TD	вх	GP	KG	TD = TV/TA; в пределах числа с плавающей запятой
9	W_k	вх	GP	KG	Здесь вход точки установки, если STEU-бит 6 = 1; в противном случае вход в слове No. 19 ($-1 < W_k < 1$)
11	STEU	вх	BM	KM	Управляющее слово (смотри страницу No)
12	YH_k	вх	GP	KG	Ручной вход здесь,если STEU-бит 6 = 1; в противном случае вход в слове No.18 ($-1 < YH_k < 1$). При алгоритме управления скоростью здесь нужно указать инкремент управляющей переменной
14	BGOG	вх	GP	KG	Значение верхнего предела $0 < BGOG < 1$ (YA_k макс.)
16	BGUG	вх	GP	KG	Значение нижнего предела $-1 < BGUG < 0$ (YA_k мин.)
18	YH_k	вх	LP	KF	Ручной ввод здесь,если STEU-бит 6 = 0 ($-1 < W_k < 1$)
19	W_k	вх	LP	KF	Здесь вход точки установки,если STEU-бит 6 = 1, ($-1 < W_k$ 1)
20	MEPI				Зарезервирован

Номер слова	Имя	Вх/вых	Формат числа	Формат PG	Примечание
21	X_k	вх	LP	KF	Ввод актуального значения для STEU-бит 7 = 0 ($-1 < X_k < 1$)
22	X_k	вх	GP	KG	Ввод актуального значения для STEU-бит 7 = 1 ($-1 < X_k < 1$)
24	Z_k	вх	LP	KF	Воздействующая величина ($-1 < Z_k < 1$)
25	Z_k	вх	G	KG	Воздействующая величина вводится здесь, если STEU-бит 7 = 1 ($-1 < Z_k < 1$)
27	Z_{k-1}		GP	KG	Предыдущее значение воздействующей величины
29	XZ_k	вх	LP	KF	Значение, прикладываемое к блоку источника через вход XZ ($-1 < XZ_k < 1$); вход здесь, если STEU бит 7 = 0
30	XZ_k	вх	GP	KG	XZ вводится здесь, если STEU-бит 7 = 1 ($-1 < XZ_k < 1$)
32	XZ_{k-1}		GP	KG	Предыдущее ограничение XZ_k
34	PZ_{k-1}		GP	KG	$XZ_{k-1} - XZ_{k-2}$
36	dD_{k-1}		GP	KG	Исходная компонента
38	XW_{k-1}		GP	KG	Предыдущее значение источника управления
40	PW_{k-1}		GP	KG	$XW_{k-1} - XW_{k-2}$
42	—	—	—	—	Зарезервировано
44	Y_{k-1}		GP	KG	Предыдущее значение вычисленной управляемой величины Y_{k-1} или dY_{k-1} до ограничения
46	YA_k	вых	GP	KG	Выход величины
48	YA_k	вых	LP	KF	Выход величины BGUG < YA < BGOG

Рекомендуемый формат (KH и KM тоже допустимы)

GP = Формат числа с плавающей запятой
LP = Двоичная или десятичная часть

вх — вход, вых — выход

о Значение слова управления STEU (слово 11 в передаваемом блоке данных)

Бит	Имя	Значение
0	AUTO	= 1 : Автоматический режим работы = 0 : Ручной режим работы
1	XZ_EIN	= 1 : дифференциальная величина, которая не должна быть XW_k, подается на блок источника через вход XZ = 0 : XW_k подается на блок источника. XZ игнорируется
2	REG_AUS	= 1 : При вызове регулятора (OB251) все значения в блоке данных регулятора (DW20...DW48) сразу стираются (кроме K, R, TI, TD, BGOG, BUGG, STEU, YH_k и W_k). Регулятор выключается. Предыдущее значение входной величины обновляется = 0 : Регулятор включается
3	GESCHW	= 1 : Алгоритм управления скоростью = 0 : Алгоритм управления положением
4[1]	HANDART	= 1 : для GESCHW = 0 (алгоритм управления положением) последняя управляемая выдаваемая переменная остается. Для GESCHW=1 (алгоритм управления скоростью) устанавливается приращение позиционирования $dY_k = 0$. = 0 : для GESCHW=0, после переключения в ручной режим, значение управляемой переменной YA экспоненциально стремится к значению, установленному вручную за 4 шага. После этого значение, устанавливаемое вручную, передается непосредственно на выход. Для GESCHW=1, значение, установленное вручную, передается непосредственно на выход регулятора. При ручном режиме работы действуют ограничители, и выполняются следующие соотношения: 1) X_k, XW_{k-1}, PW_{k-1} 2) XZ_k, XZ_{k-1}, PZ_{k-1}, если STEU бит1=1 3) Z_k и Z_{k-1}, если STEU бит5=0 Значение dD_{k-1} устанавливается в "0". Алгоритм не выполняется.
5	NO_Z	= 1 : Без прямого управления = 0 : С прямым управлением
6	PGDG	= 1 : W_k–, YH_k– вводятся в виде чисел с плавающей запятой. = 0 : Ввод в виде двоичной/десятичной части.
7	VAR_GP	= 1 : Переменные X_k, XZ_k, Z_k вводятся в виде чисел с плавающей запятой = 0 : Ввод переменной в виде двоичной/десятичной части
8–15		Не имеют значения

[1] Действительно только для ручного режима (AUTO = O).

7.2.3 Инициализация и вызов PID-регулятора

o Инициализация в организационном блоке запуска 0B 20/21/22

- Выбор передаваемого блока данных DBx (содержащего параметры)
- Вызов OB250 (программа инициализации регулятора)

Для передачи данных каждый регулятор должен использовать свои собственный блок данных DBx (x<255). Системная программа создает другой блок данных DBx+1 в памяти пользователя RAM процессора, который регулятор использует в качестве поля данных во время выполнения циклической программы, поэтому соответствующие номера блоков данных должны быть допустимы к использовании. Эти DBx+1 блоки данных являются интерфейсом данных между регулятором и пользователем или периферией.

Внимание: Если DBx+1 нельзя использовать во время инициализации, он будет использован без какой-либо индикации от операционной системы в качестве поля данных регулятора: подразумевается, что его длина такая же, как и DB регулятора (48 слов данных) и слова DW20...DW48 стираются. В противном случае контроллер переходит в состояние "СТОП".

o Вызов регулятора во время выполнения цикла

Вызов регулятора, после того как время опроса прошло:

— выбор поля данных DBx+1,
— загрузка входных данных X_k, XZ_k, Z_k, YH_k, или их комбинации,
— входные данные преобразуются в нужный формат и передаются в DBx+1,
— вызов OB251 (PID регулятор)
— загрузка выходных данных YA_k из DBx+1,
— обмен данными и передача их процессу входов/выходов.

7.2.4 Формат входов/выходов регулятора

Алгоритм PID-регулятора использует формат чисел с плавающей запятой для внутреннего представления чисел и может параметрироваться значениями с плавающей запятой. Кроме того, могут использоваться двоичные и десятичные части (смотри бит 6 и 7 в слове управления STEU). В этом случае регулятор автоматически преобразует слова в формат с плавающей запятой.

В языке STEP 5 требуется меньше времени на преобразование слов входов и выходов, если они вводятся в двоичном или десятичном виде.

о Входы

W, YH, X, Z, и XZ могут быть введены как в виде чисел с плавающей запятой, так и в двоичном/десятичном виде. Блок передачи данных для каждого значения имеет различные позиции в памяти.

Внимание! Для того, чтобы быть в пределах номинальных пределов модуля аналоговых входов, важно помнить, что значение отдельных битов будет различно в зависимости от того, какие пределы входного сигнала используются. Это следует обязательно учитывать, особенно когда устанавливается прямое управление. В противном случае может возникнуть ситуация, когда прямое управление, введенное с помощью программатора, не может быть выполнено, хотя фактическое значение намного больше, чем требуемое значение.

Если аналого-цифровой преобразователь используется для отрицательных чисел в виде значения со знаком, то из этого значения сначала должна формироваться его двоичное дополнение, перед тем как оно передается в блок данных регулятора. Двоичное значение 15 бита должно быть установлено в "1".

Если при использовании аналого-цифрового преобразователя получается число "О", то оно представляется в виде:

1000000000000000
Затем формируется двоичное дополнение. Вместо этого числа в блок данных записывается +0:

0000000000000000

о Выходы

Выход регулятора YA Формируется в DB в виде числа с фиксированной запятой, расположенной слева. Формат двоичной/десятичной части входа и выхода в зависимости от используемого модуля входов/выходов (Аналого-цифровой преобразователь, цифро-аналоговый преобразователь) должен быть преобразован до и после вызова регулятора в программе пользователя STEP-5 перед тем, как значения передаются в или из блока данных регулятора.

7.2.5 Общие указания

Параметр TI = TA/TN не может быть равен нулю, в противном случае когда происходит переключение из ручного режима работы в автоматический в устойчивом состоянии ($XWк = XW_{k-1}$, = XW_{k-2}), любое отклонение управления не может изменить управляемую переменную (формула для вычисления dYk), : смотри Раздел 7.2.1).

Блок данных регулятора DBx+1 не может быть перезагружен во время циклической работы.

Если каскадируется управление с двумя или более регуляторами, следует учитывать следующее:

— Если каскад должен быть разделен на части, то или все регуляторы должны быть переведены в ручное управление одновременно, так, чтобы ни один из регуляторов не изменял компоненту I, или, по крайней мере, регулятор внешней петли должен быть в ручном режиме, работы, так, чтобы последняя управляемая переменная, которая относится к точке установки внутренней петли, была установлена или могла бы быть установлена в безопасное значение.

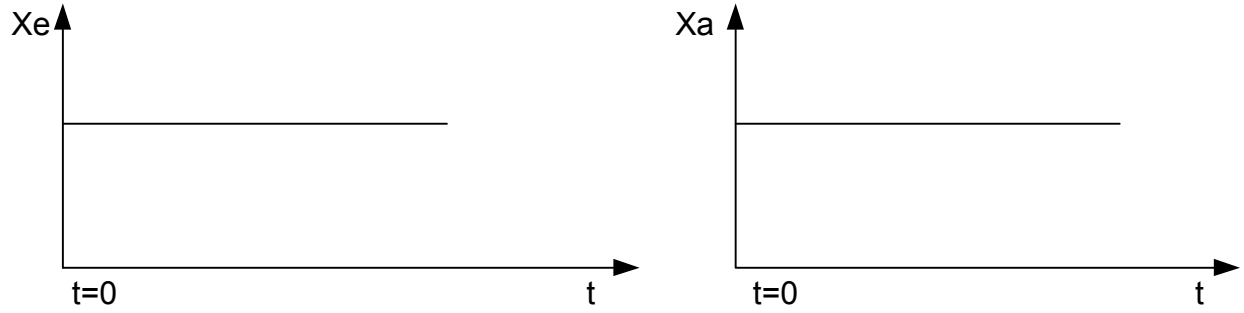
— Если каскад должен быть замкнут, то обе петли должны находиться в автоматическом режиме управления или, по крайней мере, внутренняя петля, так, чтобы управляемая переменная внешней петли могла бы быть принята за точку установки.

Если во время переключения в ручной режим работы управляемая система отделяется от регулятора и регулируется непосредственно конечный элемент управления, то результирующая управляемая переменная должна быть подключена к ручному входу регулятора. Этим обеспечивается то, что когда происходит переключение из ручного режима работы в автоматический, регулятор выдает набор управляемых переменных, соответствующий работе в ручном режиме. При использовании алгоритма управления скоростью - это управляемая переменная отклонения.

7.2.6 Характеристики регулятора

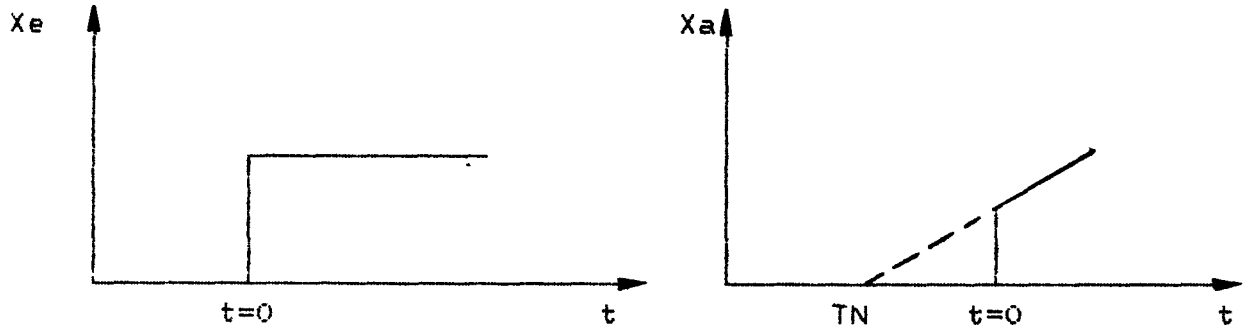
o P-регулятор

Количественная характеристика для P-регулятора – это K. Это отношение входной и выходной величин: $K = X_{out}/X_{in}$



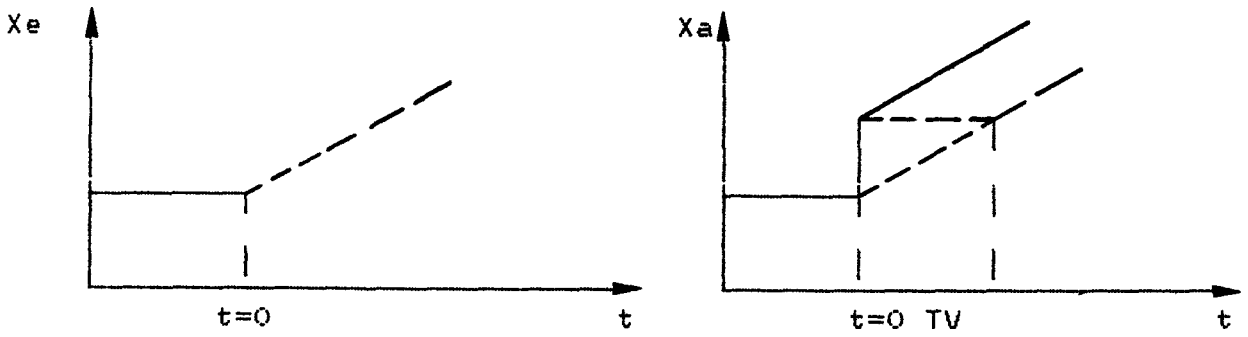
o PI-регулятор

Коэффициент пропорциональности K и интегральная константа времени TN являются количественными характеристиками для PI-регулятора. Коэффициент пропорциональности K — отношение входной и выходной величин - определяет реакцию P. Время сброса TN — это время, требуемое для реакции и достижения того же значения регулируемой величины с помощью воздействия I, как и для компоненты P.



o PD—регулятор

Коэффициент пропорциональности K (см. выше) и константа времени отклонения TV являются количественными характеристиками PD-регулятора. Время прямого управления — -это время, которое необходимо P—регулятору, чтобы произошло такое же изменение на выходе, какое в PD—регуляторе, при наличии компоненты D. Для определения времени подачи линейного изменения входное значение используется вместо шаговой функции.



o PID—регулятор

Количественные характеристики PID—регулятора — это коэффициент пропорциональности K, время сброса TN и время подачи TV. Они определяют P, I и D реакцию.

7.2.7 Двоичные/десятичные дроби

Для представления двоичных и десятичных частей в блоке данных требуется одно слово. Связь между десятичным числом, двоичным числом и представлением в виде числа в Формате с фиксированной запятой KF в программаторе показана на следующем примере.

Часть		Число с фиксированной запятой
десятичная	двоичная	
– 0.999...	1000000000000001	– 32767
– 0.75	1010000000000000	– 24576
– 0.5	1100000000000000	– 16384
– 0.25	1110000000000000	– 8192
0	0000000000000000	0
+ 0.25	0010000000000000	+ 8192
+ 0.5	0100000000000000	+ 16384
+ 0.75	0110000000000000	+ 24576
+ 0.999...	0111111111111111	+ 32767

В двоичном представлении отрицательные числа отличаются от положительных тем, что они образованы двоичным дополнением.

Двоичные/десятичные числа (LP) могут быть преобразованы в значения, отображаемые программатором в формате KF по формуле:

LP x 32767 = KF

если -1 < LP <=1 и -32767 < KF < +32767

7.2.8 Сокращения

dY_k	Вычисление приращения позиционирования
dZ_k	Приращение воздействия
GP	Представление в виде числа с плавающей запятой
k	k-тый опрос
K	Коэффициент пропорциональности
LP	Представление в виде числа с запятой слева (0.123)
UL	Верхний предел
R	R-параметр
TA	Время опроса
TD	TV/TA
TI	TA/TN
t	Точка опроса (время) = k.TA
TN	Время сброса
TV	Время подачи вперед
LL	Нижний предел
W_k	Точка установки
X_k	Актуальное значение
XW_k	Отклонение управляющего воздействия
Y_k	Вычисленная управляемая переменная
YA_k	Значение управляемой переменной (приращение положения или управляемой переменной)
Z_k	Воздействующая величина

7.3 Регистр сдвига

7.3.1 Принцип действия

Следующая диаграмма иллюстрирует принцип построения программного сдвигового регистра. Он состоит из ряда 8—битовых позиций хранения в блоке памяти RAM контроллера 135 U.

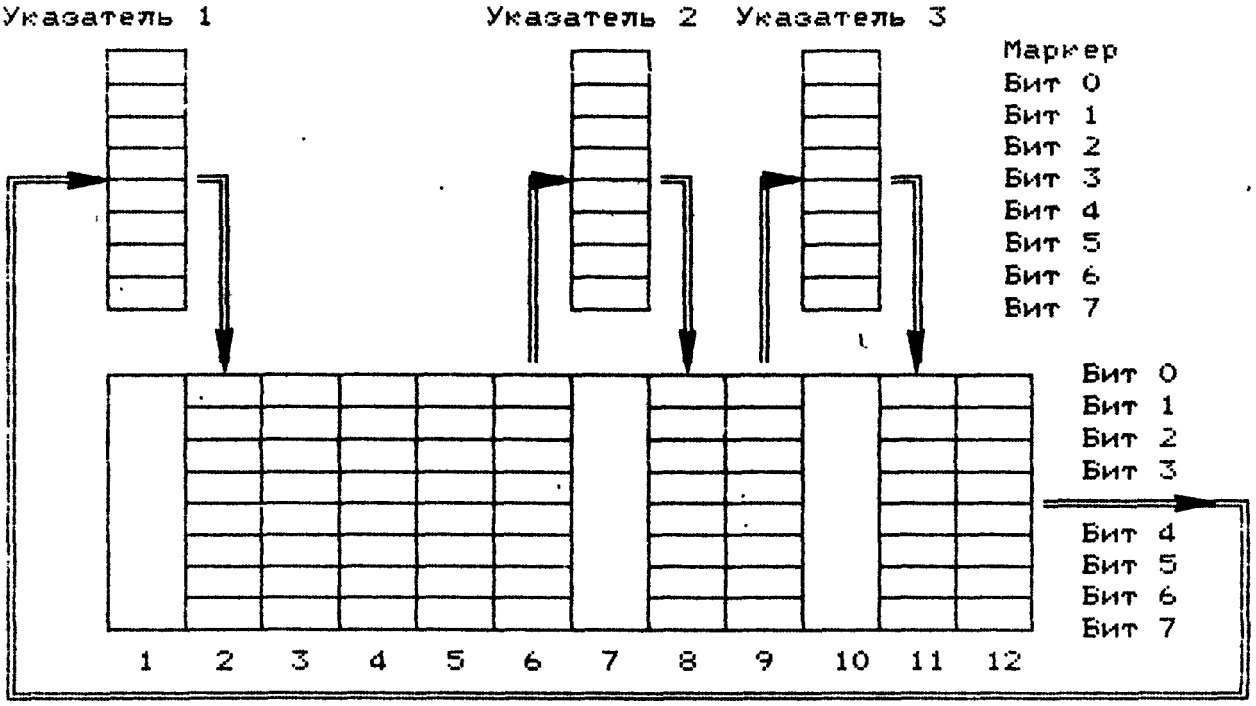


Рис.19 Схематическая диаграмма сдвигового регистра с 3 указателями и 12 ячейками хранения

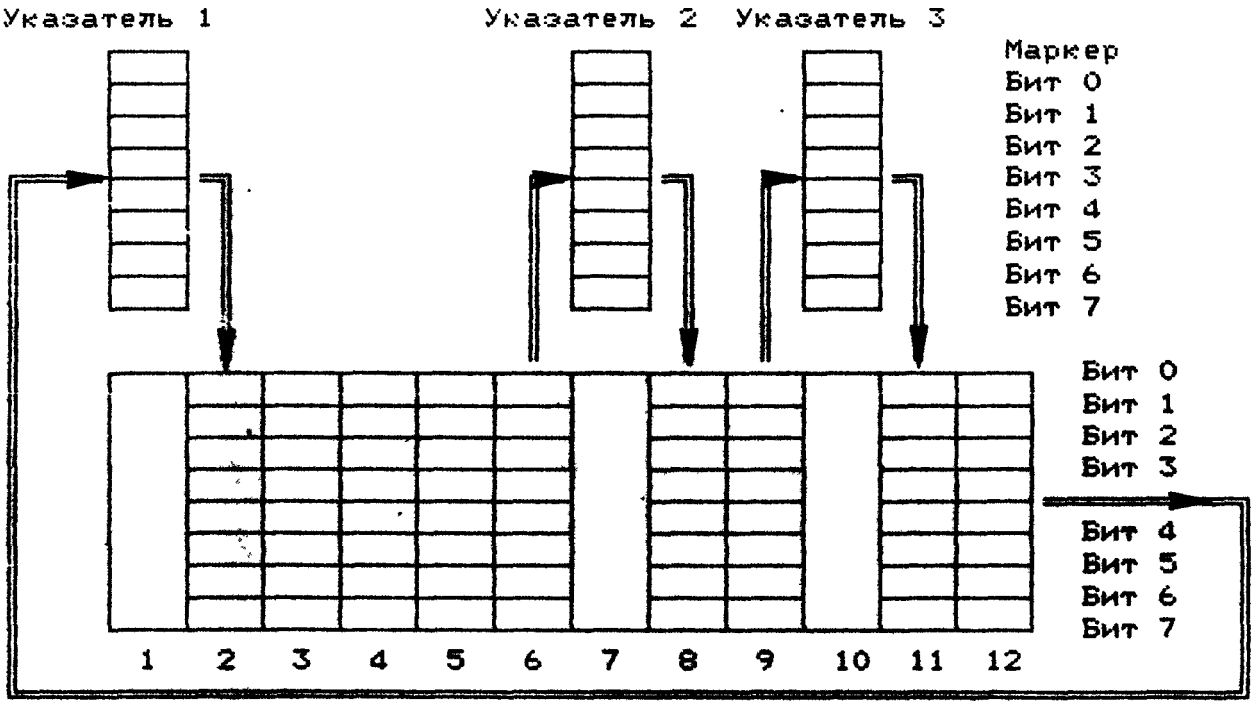


Рис.20 Схематическая диаграмма сдвигового регистра с 3 указателями и 12 ячейками хранения перед первым тактовым импульсом

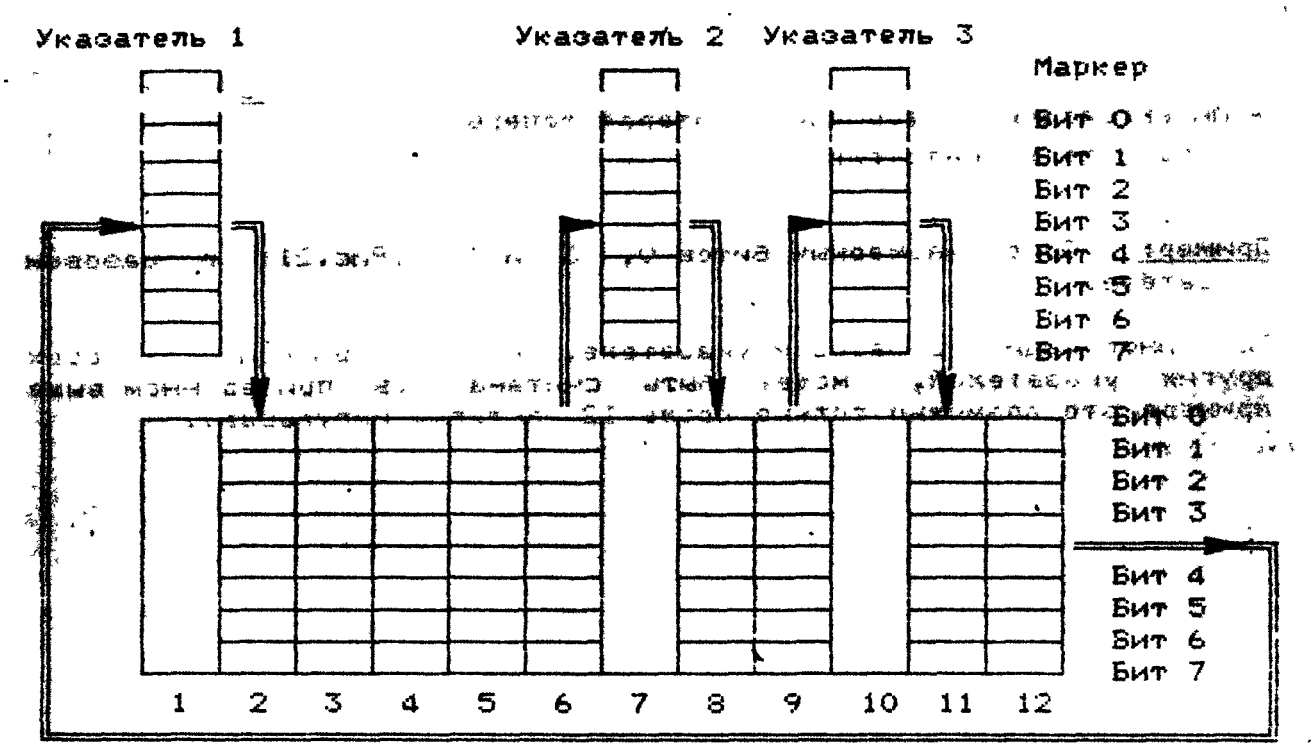


Рис.21 Схематическая диаграмма сдвигового регистра с 3 указателями и 12 ячейками хранения после первого тактового импульса

Количество ячеек хранения может быть выбрано пользователем в пределах от 2 до 256. Отдельные сдвиговые регистры находятся вне памяти RAM в байтах маркеров, которые после их назначения используются в этом качестве.

Установка первого указателя (базовый указатель) жестко располагается в первой ячейке. Все остальные указатели могут располагаться пользователем относительно базового указателя произвольно. Количество указателей определяется пользователем от одного (базовый.) до шести. Как в случае аппаратно-реализуемых сдвиговые регистров, информация сдвигается побайтно от позиции 1, как показано на рисунке, к позиции L. Отсюда информация возвращается снова в первую позицию. Каждый вызов функции сдвигового регистра заставляет всю информацию сдвигаться на 1 позицию хранения → 1 тактовый импульс.

Пользователь может вводить информацию в регистр сдвига или выбирать информацию из него. Это возможно только при использовании указателей, т.е. байты маркеров используются в качестве указателей.

— Перед вызовом функции SR: установить/сбросить биты маркеров (Рис. 20).

Пример: STEP-5

Установить бит маркера О указателя 1	S M 0.0
Установить бит маркера 3 указателя 2	S M 1.3
Установить бит маркера 2 указателя 3	S M 2.2

— Вызвать функцию сдвигового регистра SPA OB 241

Результат: Информация (8 бит) в ячейке сдвигается на 1 позицию.

— Интерпретация информации, которая теперь находится в указателе

STEP - 5
L МБ О

Пример: Опрос маркерных битов 0, 3 и 2 (Рис. 21) в базовом указателе.

Вся информация в базовом указателе, которая приходит от всех других указателей, может быть считана (в приведенном выше примере это возможно только после 12 тактовых импульсов).

7.3.2 Программирование регистра сдвига в программе пользователя (OB241...OB248)

В контроллере S5-135U могут быть запрограммированы 8 регистров сдвига. Их вызов встраивается в программу пользователя с использованием организационных блоков OB241… OB248. До того как сдвиговый регистр может быть вызван в программе пользователя, должны быть установлены его параметры в блоке данных. В следующей таблице приводятся номера организационных блоков и блоков данных, соответствующих номерам регистров сдвига, расположенные по порядку

Регистр сдвига 1	OB241	DB 241
Регистр сдвига 2	OB242	DB242
Регистр сдвига 3	OB243	DB243
Регистр сдвига 4	OB244	DB244
Регистр сдвига 5	OB245	DB245
Регистр сдвига 6	OB246	DB246
Регистр сдвига 7	OB247	DB247
Регистр сдвига 8	OB248	DB248

Каждый регистр сдвига использует в качестве указателей по шесть маркерных байтов:

Маркерный байт 0 -> Указатель 1 Маркерный байт 1 -> Указатель 2 Маркерный байт 2 -> Указатель 3 Маркерный байт 3 -> Указатель 4 Маркерный байт 4 -> Указатель 5 Маркерный байт 5 -> Указатель 6	Регистр сдвига 1
Маркерный байт 6 -> Указатель 1 Маркерный байт 7 -> Указатель 2 Маркерный байт 8 -> Указатель 3 Маркерный байт 9 -> Указатель 4 Маркерный байт 10 -> Указатель 5 Маркерный байт 11 -> Указатель 6	Регистр сдвига 2
Маркерный байт 12 -> Указатель 1 Маркерный байт 13 -> Указатель 2 Маркерный байт 14 -> Указатель 3 Маркерный байт 15 -> Указатель 4 Маркерный байт 16 -> Указатель 5 Маркерный байт 17 -> Указатель 6	Регистр сдвига 3
Маркерный байт 18 ->	
Маркерный байт 42 -> Указатель 1 Маркерный байт 43 -> Указатель 2 Маркерный байт 44 -> Указатель 3 Маркерный байт 45 -> Указатель 4 Маркерный байт 46 -> Указатель 5 Маркерный байт 47 -> Указатель 6	Регистр сдвига 8

Рис.22 Фиксированное назначение маркерных байтов регистрам сдвига с 1 по 8.

o Назначение параметров регистров сдвига в организационных блоках запуска (OB20/OB21/OB22)

В блок данных, который относится к регистру сдвига, должна быть занесена следующая информация (пример для 6 указателей):

Слово 0	0
Слово 1	Длина L
Слово 2	Расстояние n_2
Слово 3	Расстояние n_3
Слово 4	Расстояние n_4
Слово 5	Расстояние n_5
Слово 6	Расстояние n_6
Слово 7	0

L — место расположения сдвигового регистра в памяти (2 < L < 247)

n_2 - расстояние от базового указателя до второго указателя.
n_3.- расстояние от базового указателя до третьего указателя и т.д.

Первое и последнее слово блока данных параметров должны быть равны нулю. Если есть только один указатель, базовый, то следует указать только место расположения сдвигового регистра.

Слово О: О
Слово 1: L
Слово 2: О

Для установки параметров при запуске контроллера S5—135U (OB20/OB21/OB22) для каждого регистра сдвига используются;
— соответствующий блок данных и
— (затем должен быть вызван) организационный блок ОВ240.
При инициализации в блоке памяти RAM для каждого регистра сдвига резервируются 128 слов. С учетом этого конечный адрес блоков данных в области памяти RAM сдвигается в сторону младших адресов.

o Вызов регистра сдвига в циклической программе пользователя

Вводятся данные в конкретные байты маркеров.

Вызывается организационный блок, который связан с регистром сдвига (OB241…OB248) .

Интерпретируются данные в конкретных байтах данных.

Так как каждый вызов регистра, сдвига соответствует одному тактовому импульсу, последовательной программе должен был бы предшествовать опрос фронта.

7.3.3 Деблокировка области памяти регистра сдвига (OB 221)

Организационный блок OB240 резервирует, область памяти в DB RAM процессорного модуля для каждого инициализированного регистра сдвига (128 слов на каждый регистр сдвига). Память пользователя деблокируется вызовом организационного блока OB221 (предпочтительно во время запуска)

7.4 Изменение времени цикла (OB222)

С помощью вызова организационного блока OB22 пользователь может продлить контрольное время цикла, которое равно 100 мсек, измеряя его с момента вызова OB, т.е. "внутренние часы" перезапускаются программой пользователя. Перед тем, как программа пользователя будет вызвана в OB1 или FBO (см. Рис. 11), системная программа переключает время цикла.

7.5 Расширение разрядности чисел с фиксированной запятой с 16 до 32 бит (OB220)

Вызовом организационного блока OB220 16—битовое число со знаком и фиксированной запятой в аккумуляторе АКК 1 расширяется до 32 бит. Это, например, полезно для преобразования в число с плавающей запятой (FDG, 32 бит) отрицательного числа с фиксированной запятой.

7.6 Сравнение видов запусков в многопроцессорном режиме (OB223)

Если организационный блок OB223 вызывается в многопроцессорном режиме работы, соответствующий процессор проверяет, все ли процессоры выполнили запуск. Если нет, то процессор переходит в "СТОП" и выдает сообщение об ошибке SFF (см. Раздел 5.7).

7.7 Считывание контрольной суммы операционной системы на EPROMe (OB227)

Если вызывается организационный блок OB227, в аккумулятор АКК 1-L передается контрольная сумма системной программы EPROMa.

Если вызывается OB226, содержимое памяти по адресу, который должен быть предварительно загружен в АКК 1, из этого EPROMa передается в АКК 1.

Во время выполнения циклической программы может быть проверена контрольная сумма EPROMa посредством сложения содержимого веек байтов в виде числа с фиксированной запятой (адреса с 0 по 7FFF:).

7.8 Передача блоков маркеров связи (OB224)

Обычно маркеры межпроцессорной связи, указанные пользователем в DB1, побайтно передаются системной программой в процессор (смотри Раздел 1.4.1 и Рис.11). При многопроцессорном режиме работы эта передача выполняется каждым процессором независимо через координатор с целью минимизировать время блокирования шины процессором. Поэтому в качестве когерентных блоков информации могут передаваться байты.

Вызов OB224 используется при начальном запуске в любом режиме запуска, и при включении каждого процессора пользователь может передавать в блоки все маркеры межпроцессорной связи, указанные в DB1. Каждый процессор может начать передачу своих маркеров межпроцессорной связи только тогда, когда другой процессор завершит свою передачу. Выполнение циклической программы будет соответственно задержано (время цикла!). С помощью этой специальной функции обновление маркеров межпроцессорной связи будет отдельным для каждого процессора, чтобы можно было наверняка быть уверенным в новом состоянии информации всех маркеров межпроцессорной связи. Функция будет работать, только если координатор находится в состоянии "RUN" с деблокировкой шины < 8 мкс (смотри Руководство по координатору KOR).

7.9 Параметрирование характеристик запуска (OB225)

Если вызывается специальная функция OB225, состояние ячеек таймеров и счетчиков сохраняется при ручном или автоматическом новом пуске без запоминания. OB225 должен вызываться до завершения выполнения OB21 или OB22. Он становится активным при следующем новом пуске без запоминания. Следует заметить, что при всех последующих новых пусках с запоминанием действует стандартная установка "стирание ячеек таймеров и счетчиков" до тех пор, пока не будет вызова OB255.

8. Специальные команды языка STEP-5

Начиная с версии S0 AO3 программного обеспечения для программатора PG675, в языке STEP-5 допускается использование специальных функций:

Создание блока данных	EDBxxx
Установка семафора	SES
деблокировка семафора	SEF

8.1 Создание блока данных

По команде E DBxxк создается блок данных с номером xxx (начиная с 2 по 255) в области памяти RAM процессора. В аккумуляторе АКК 1 находится информация о количестве слов данных (максимальная длина блока данных — 4096 слов, включая заголовок блока). Если соответствующий блок данных уже существует или в памяти RAM процессора не достаточно места для его формирования, то выдается сообщение об. ошибке SFF.

8.2 Установка/сброс семафора

Команды управления "Установка семафора" SESxxx и "Деблокировка семафора" SEFxxx служат для обмена данными между центральными процессорами (CPU) или коммуникационными процессорами (CP) при многопроцессорном режиме работы. При установке семафора SESxxx область данных (маркеры межпроцессорной связи), которые должны быть определены в программе пользователя,. будут блокированы для других процессоров. По команде SEFxxx эта область данных может быть считана или в нее может быть записана информация других процессоров. Семафор может быть деблокирован . только тем процессором, который его установил. Команды SES/SEF влияют на коды условий (см. Раздел 6.1) следующим образом;

ANZ1	ANZO	Значение
0	0	Семафор был установлен другим процессором и не может быть установлен/сброшен
1	0	Семафор устанавливается/сбрасывается

Состояние этих битов условий может быть использовано с помощью команд перехода по условию (см. Раздел 6.3)

8 Обзор операций команд языка STEP—5

Основные операции

Операция			Параметр
o Двоичные логические операции			
U	E		0.0 ... 127.7
U	A		0.0 ... 127.7
U	M		0.0 ... 255.7
U	D	1)	0.0 ... 255.15
U	T		0 ... 127
U	Z		0 ... 127
UN	E		0.0 ... 127.7
UN	A		0.0 ... 127.7
UN	M		0.0 ... 255.7
UN	D	1)	0.0 ... 255.15
UN	T		0 ... 127
UN	Z		0 ... 127
O	E		0.0 ... 127.7
O	A		0.0 ... 127.7
O	M		0.0 ... 255.7
O	D	1)	0.0 ... 255.15
O	T		0 ... 127
O	Z		0 ... 127
ON	E		0.0 ... 127.7
ON	A		0.0 ... 127.7
ON	M		0.0 ... 255.7
ON	D	1)	0.0 ... 255.15
ON	T		0 ... 127
ON	Z		0 ... 127
)			
U(			
O(			
O			
o Функции сравнения			
!=F			
><F			
>F			
>=F			
<F			
<=F			
!=D			
><D			
>D			
>=D			
<D			
<=D			
!=G			
><G			
>G			
>=G			
<G			
<=G			

Операция			Параметр
o Операции с памятью			
S	E		0.0 ... 127.7
S	A		0.0 ... 127.7
S	M		0.0 ... 255.7
S	D	1)	0.0 ... 255.15
R	E		0.0 ... 127.7
R	A		0.0 ... 127.7
R	M		0.0 ... 255.7
R	D	1)	0.0 ... 255.15
=	E		0.0 ... 127.7
=	A		0.0 ... 127.7
=	M		0.0 ... 255.7
=	D	1)	0.0 ... 255.15
o Функции загрузки			
L	EB		0 ... 127
L	EW		0 ... 126
L	ED		0 ... 124
L	AB		0 ... 127
L	AW		0 ... 126
L	AD		0 ... 124
L	MB		0 ... 255
L	MW		0 ... 254
L	MD		0 ... 252
L	DL		0 ... 255
L	DR		0 ... 255
L	DW		0 ... 255
L	DD		0 ... 254
L	T		0 ... 127
L	Z		0 ... 127
L	PB		0 ... 127
			128 ... 255
L	PW		0 ... 126
			128 ... 254
L	QB		0 ... 255
L	QW		0 ... 254
LD	T		0 ... 127
LD	Z		0 ... 127
L	KB		0 ... 255
L	KC		2 алфав.-цифр. символа
L	KM		Набор 16 бит
L	KH		0 ... FFFF
L	KF		– 32768 ... ... + 32767
L	KY		0 ... 255 для каждого байта
L	KT		0.0...999.3
L	KZ		0...999
L	KG		2)

1) Слово 1: B2 + бит адреса; B3 + относительный адрес

2) ё0.1469368 x 10^{-39} ... ё0.1701412 x 10^{39}

Операция	Параметр
o Операции с таймерами и счётчиками	
SI T	0 ... 127
SV T	0 ... 127
SE T	0 ... 127
SS T	0 ... 127
SA T	0 ... 127
R T	0 ... 127
S Z	0 ... 127
R Z	0 ... 127
ZV Z	0 ... 127
ZR Z	0 ... 127
o Функции передачи	
T EB	0 ... 127
T EW	0 ... 126
T ED	0 ... 124
T AB	0 ... 127
T AW	0 ... 126
T AD	0 ... 124
T MB	0 ... 255
T MW	0 ... 254
T MD	0 ... 252
T DR	0 ... 255
T DL	0 ... 255
T DW	0 ... 255
T DD	0 ... 254
T PB	0 ... 127
	128 ... 255
T PW	0 ... 126
	128 ... 254
T QB	0 ... 255
T QW	0 ... 254
o Вызов блоков	
SPA PB	0 ... 255
SPA FB	0 ... 255
SPA SB	0 ... 255
SPB PB	0 ... 255
SPB FB	0 ... 255
SPB SB	0 ... 255
A DB	2 ... 255
BE	
BEB	
BEA	
SPA OB	1 ... 39
SPB OB	1 ... 39
AX DX	1 ... 255
BA FX	0 ... 255
BAB FX	0 ... 255
SPA OB [1]	40 ... 255
SPB OB [1]	40 ... 255

[1] Вызов спецфункции

Операция	Параметр
o Арифметические операции	
+F	
-F	
xF	
:F	
+G	
-G	
xG	
:G	
o Специальные функции	
NOP 0	
NOP 1	
STP	
BLD	

Дополнительные функции

Операция	Параметр
o Логические функции с словами	
UW	
OW	
XOW	
o Операции с таймерами и счетчиками	
FR T	0 ... 127
FR Z	0 ... 127
FR =	Форм.операнд
SI =	Форм.операнд
SE =	Форм.операнд
SVZ =	Форм.операнд
SSV =	Форм.операнд
SAR =	Форм.операнд
RD =	Форм.операнд
o Функции загрузки и передачи	
L =	Форм.операнд
LC =	Форм.операнд
LW =	Форм.операнд
LD =	Форм.операнд
T =	Форм.операнд
L BS	0 ... 255
L BA	0 ... 255
T BA [2]	0 ... 255
T BS [2]	0 ... 255
LIR [2]	0 ... 15

[2] Системная операция

Операция	Параметр
o Функция загрузки и пересылки (продолжение)	
TIR [1]	0 ... 15
TNB [1]	0 ... 255
TNW [1]	0 ... 255
o Двоичные логические операции	
U =	Форм.операнд
UN =	Форм.операнд
O =	Форм.операнд
ON =	Форм.операнд
o Функции преобразования	
KEW	
KZW	
KZD	
DEF	
DUF	
DED	
DUD	
FDG	
GFD	
o Функции сдвига	
SLW	0 ... 15
SRW	0 ... 15
SLD	0 ... 32
SVD	0 ... 32
RLD	0 ... 32
RRD	0 ... 32
SVW	0 ... 15
o Функции перехода	
SPA =	Символич.адрес
SPB =	Символич.адрес
SPZ =	Символич.адрес
SPN =	Символич.адрес
SPP =	Символич.адрес
SPM =	Символич.адрес
SPO =	Символич.адрес
SPS [2] =	Символич.адрес
SPR [1] =	-32768...+32767

Операция	Параметр
o Операции установки	
S =	Форм.операнд
RB =	Форм.операнд
= =	Форм.операнд
o Другие функции	
AF	
AS	
ENT	
D	0 ... 255
I	0 ... 255
B =	Форм.операнд
B DW	0 ... 255
B MW	0 ... 255
BI [1]	
B BS [1]	0 ... 255
TAK	
BLD	0 ... 255
o Арифметические операции	
ADD BF	-127 ... +127
ADD KF	-32768 ... +32767
o Специальные функции	
E DB	0 ... 255
SES 1)	0 ... 31
SEF 1)	0 ... 31

[1] Системная операция

[2] Слово 1: дистанция перехода (2 байта)

10 Информация о неисправностях

Информация о всех возможных неисправностях хранится в системных словах SD3 и SD4

Идентификатор неисправности SD3 SD4	Значение ошибки	Реакция системы
0101 абс	Неправильный идентификатор блока или длина блока пользователя	Переход в СТОП" по причине DBO
0102 абс	Неправильный идентификатор блока или длина блока в области блоков данных RAM	Переход в "СТОП" по причине DBO
0103 абс	Неправильная передача DB1	Переход в "СТОП" по причине DB1
0104 -	EPROM пуст или не установлен	Системная программа требует общего сброса. Без сброса запуск невозможен
0105 -	Неправильное содержимое памяти RAM пользователя Неправильный идентификатор	Системная программа требует общего сброса Без сброса запуск невозможен
0106 -	RAM процессора содержит неправильный идентификатор	Системная программа требует общего сброса Без сброса запуск невозможен
0201 -	Отсутствие адресного списка в многопроцессорном режиме	Остается в состоянии "СТОП"
0202 -	Выбран неправильный режим запуска	"СТОП"
0301 укз	Ошибка адресного списка: присутствуют цифровые входы, модули которых не подключены (при одно- и многопроцессорном режиме работы)	Возврат снова в цикл "СТОП"
0302 укз	Ошибка адресного списка: присутствуют цифровые выходы, модули которых не подключены (в многопроцессорном режиме работы)	Возврат снова в цикл "СТОП"
0303 укз	Ошибка адресного списка: присутствуют цифровые выходы, модули которых не подключены (в многопроцессорном режиме работы.)	

абс = Абсолютный адрес неверных данных
укз = Указатель списка адресов области отображения процесса
- = не имеет значения

Инициализация регистров сдвига

Идентификатор неисправности SD3 SD4	Значение ошибки	Реакция системы
0701 адр	Нет места в памяти DB RAM	"СТОП"
0702 адр	Регистр сдвига слишком длинный или положение указателя выходит за пределы регистра	"СТОП"

Вызов регистра сдвига

Идентификатор неисправности SD3 SD4	Значение ошибки	Реакция системы
0703 87xx	Сдвиговый регистр не инициализирован xx — номер регистра сдвига	"СТОП"

Инициализация PID—регулятора

Идентификатор неисправности SD3 SD4	Значение ошибки	Реакция системы
0704 адр	Нет места в DB RAM	"СТОП"
0705 адр	DBx No = 255	"СТОП"
0706 адр	Неправильная длина DB	"СТОП"
0707 адр	DB+1 уже существует в памяти пользователя	"СТОП"

адр = счетчик адресов

Вызов PID—регулятора

Идентификатор неисправности SD3 504	Значение ошибки	Реакция системы
0708 87xx	DBx+1 уже существует в памяти пользователя, ошибка дополнительного идентификатора PID регулятора (цикл) xx – No. DB	"СТОП"

Идентификатор неисправности SD3 SD4	Значение ошибки	Реакция системы
2101 –	Неисправность питания NAU	"СТОП"
2102 -	Неисправность входов/выходов PEU	"СТОП"
2103 –	Неисправность батареи BAU	"СТОП"
2104 –	Сигнал останова HALT	"СТОП"
2105 –	Сигнал "СТОП" STOP	"СТОП"
2106 *x*	Задержка сигнала квитирования QVZ x — во время формирования отображения процесса (см USTACK) = указатель элемента в списке адресов отображения процесса x — при обработке сигнала от модулей входов/выходов = адрес	"СТОП"
2107 адр	Ошибка адресации ADF	"СТОП"
2108 адр	Ошибка цикла ZYKF	"СТОП"
3110 адр	Недопустимый вызов DBO	"СТОП"
3111 адр	Вызов несуществующего блока данных	"СТОП"
3112 адр	Неправильная многословная команда в BS RAM	"СТОП"
3117 адр	Запрос останова от пользователя	

адр = счетчик адресов

- = не имеет значения

Идентификатор неисправности SD3 SD4	Значение ошибки	Реакция системы
3118 адр	Эта команда STEP-5 не существует	"СТОП"
3119 адр	Эта команда STEP-5 не существует	"СТОП"
3120 адр	Эта команда STEP-5 не существует	"СТОП"
3121 адр	Эта команда STEP-5 не существует	"СТОП"

Специальная функция OB255

Идентификатор неисправности SD3 SD4	Значение ошибки	Реакция системы
3403 адр	DB не загружен	"СТОП"
3404 адр	Недостаточно места в памяти	"СТОП"
3405 адр	DB уже есть в памяти DB RAM	"СТОП".

Идентификатор неисправности SD3 SD4	Значение ошибки	Реакция системы
35xx адр	xx — номер ячейки вызванного таймера больше, чем номер ячейки таймера, указанной в DB1 или больше 127	"СТОП"
3610 адр	Переполнение стека блоков	"СТОП"
3611 адр	Вызываемый блок не загружен	"СТОП"
3616 адр	Ошибка прерывания по времени	"СТОП"
3622 адр	Вызываемый OB специальной функции отсутствует	"СТОП"

адр = счетчик адресов

Идентификатор неисправности SD3 SD4	Значение ошибки	Реакция системы
3701 адр	Недопустимая команда многословная команда BMW/BOW	"СТОП"
3702 адр	Параметры блока с командами L=/T = недопустимы	"СТОП"
3711 адр	Формальный параметр в команде BI= равен нулю	"СТОП"
3712 адр	Формальный параметр в команде BI= больше 126	"СТОП"
3801 адр	Недопустимый параметр блока в команде (U=, UN=, 0=, ON=, ==, S=, RB=)	"СТОП"
3810 адр	Стирание DBO	"СТОП"
3812 адр	Недостаточно места в памяти при выполнении команды EDB	"СТОП"
3813 адр	DB уже существует (при выполнении команды EDB)	"СТОП"

адр = счетчик адресов

Приложение А:

Некоторые отличия действия команд STEP-5 в контроллере 135 U по сравнению со 150 S/U:

Функции загрузки

Не все байтовые или словные функции загрузки стирают старшее слово в аккумуляторе АКК 1. Кроме того, только младшее слово АКК I сохраняется в младшем слове АКК 2. Старшие 16 битов в аккумуляторе АКК 1 и АКК 2, поэтому, не изменяются этими командами.

Операции с таймерами и счетчиками

Область параметров для всех операций с таймерами и счетчиками ограничивается диапазоном от О...127, т.е. максимально может быть запрограммировано 128 таймеров и счетчиков. Обычно это ограничение накладывается и на двоичные логические операции с таймерами и счетчиками.

Арифметические операции

При операциях с операндами в виде чисел с плавающей запятой +G, -G, XG и :G используется только 16-битовая мантисса, в противоположность 24-битовой мантиссе в контроллере S5—150 S. Младшие 3 битов мантиссы устанавливаются в "О".

Приложение В

Замечания по программированию S—процессора в контроллере 135 U

Программировать контроллер 135 U и тестовые функции можно с помощью программаторов PG670 (версия программного обеспечения A08) и PG675 (.версия программного обеспечения SO AO1) с. предустановкой типа контроллера 150 S (словные инструкции). В этом случае действуют следующие ограничения:

- Набор команд и ограничения параметров (смотри Раздел 9).

- Во время выполнения байтовых и словных команд загрузки старшее слово в аккумуляторе АКК 1 не стирается, и передается только младшее слово из аккумулятора АКК 1 в акккумулятор АКК 2.
- Отображение сигнала состояния в зависимости от программы и проверка (контроль) выполнения программы могут быть выполнены только при использовании памяти пользователя типа RAM. Отображение состояния переменных (СОСТ ПЕР) выполняется также при использовании памяти EPROM.
- Блоки данных DBO и DB1 не должны использоваться в программе пользователя ни при однопроцессорной, ни при многопроцессорной работе контроллера (они зарезервированы для системных программ).

* Вызов организационных блоков с номерами больше 39 в программе пользователя не допускается.
* Конфигурация памяти в контроллере: Значение для 16-битовой памяти пользователя и для 8-битовой памяти DB RAM выводятся поочередно.
* Системные параметры в контроллере: Расположение информации на экране может быть неверным.
* Адрес памяти в контроллере: Если выдаются адреса > OBOOH, положение адреса на программаторе не верно.
* Отображение сигнала на экране программатора в зависимости от выполнения программы/непосредственное отображение сигналов/контроль программы: При отображении таймеров, счетчиков и слов данных выводятся некорректные значения.
* Управление отображением процесса (УПР ПЕР): Этот режим не действителен.
* Считывание стека прерываний: Расположение битов управления на экране и идентификаторов в байте отображения состояния не верно.
* Переключение режимов работы (СТАРТ/СТОП) контроллера: Эти функции не работают.
* Тест блоков: Тест блоков не может быть введен.

Выше перечисленные ограничения, обозначенные звездочкой, снимаются, если используется версия программного обеспечения для PG675U-S0-A02 или PG675U-S1-A01.

SIEMENS

S5-135U

ПРОГРАММИРУЕМЫЙ КОНТРОЛЛЕР S5-135U

R—процессор

Руководство по программированию	Каталог.No .C79000—B8500—C364—01

Оглавление Страница

1 Пояснения
1.1 Применение
1.2 Однопроцессорный режим Много-
1.3 процессорный режим

2 Программирование
2.1 Язык программирования и структура программы
2.1.1 Язык программирования STEP 5
2.1.2 Структура программы
2.2 Организация, хранение и обработка программы
2.2.1 Организация программы
2.2.2 Хранение программы
2.2.3 Выполнение программы
2.3 Программирование в многопроцессорном режиме
2.3.1 Маркеры межпроцессорной связи
2.3.2 Распределение периферии
2.4 Оптимизация программы пользователя и ограничения по времени

2.5 Программные блоки
2.5.1 Программирование программных блоков
2.5.2 Вызов программных блоков
2.6 Блоки данных
2.6.1 Программирование блоков данных
2.6.2 Вызов блоков данных
2.6.3 Специальные блоки данных
2.7 Функциональные блоки
2.7.1 Структура функциональных блоков
2.7.2 Вызов и параметрирование функциональных блоков Программиро-
2.7.3 вание функциональных блоков
2..8 Организационные блоки

3 Работа процессора
3.1 Обзор рабочих состояний
3.2 Инициализация
3.3 Состояние останова
3.3.1 Быстрое мерцание светодиода "STOP"
3.3.2 Непрерывное свечение светодиода "STOP"
3.4 Медленное мерцание светодиода "STOP"
3.5 Режим тестирования
3.5.1 Запуск
3.5.2 Новый пуск
3.5.3 Ручной перезапуск
3.5.4 Автоматический перезапуск
3.6 Запуск в многопроцессорном режиме
3.6.1 Обработка программы
3.6.2 Циклическая работа
3.6.3 Обработка программы прерывания от процесса
3.7 Обработка программы прерывания по времени
3.7.1 Обработка аппаратных прерываний
3.7.2 Прерывание на границе команд STEP 5
3.7.3 Прерывание в других случаях
4 Управляющие биты и стек прерываний
4.1 Набор команд STEP-5 с примерами программирования
4.2 Основные правила
4.3 Основной набор операций дополнительный набор операций
5 Специальные функции системной программы
5.1 Расширение 16-битового числа с фиксированной запятой до 32 бит
5.2 Новая установка времени цикла
5.3 Переключение контрольного времени цикла
5.4 Чтение ячейки памяти EPROM системной программы
5.5 Чтение контрольной суммы EPROM системной программы
5.6 Регистр сдвига
5.6.1 Режим работы
5.6.2 Инициализация и вызов регистра сдвига
5.6.3 Стирание регистра сдвига
5.7 PID-регулятор
5.7.1 PID-алгоритм
5.7.2 Блоки данных для PID-регулятора
5.7.3 Инициализация и вызов PID-регулятора в программе STEP-5
5.7.4 Формат контроллера ввода/вывода
5.7.5 Общие замечания
5.7.6 Характеристики регулятора
5.7.7 Двоично/десятичные дроби
5.8 Функция копирования блоков данных
5.9 Блок передачи маркеров межпроцессорной связи
5.10 Сравнение режимов запуска
5.11 Обращение к буферам связи
5.11.1 Запись данных в буфер связи
5.11.2 Считывание данных из буфера связи
5.11.3 Заполнение буфера связи
5.12 Вспомогательные Функции операционной системы
6 Обзор операций STEP 5
7. Информация об ошибках
7.1 Информация об ошибках в системных байтах 3 и 4
7.2 Информация об ошибках через Акк 1 и Акк 2

Аббревиатура.

AG,	Программируемый контроллер
ПК АКК 1(2)-L(H)	Аккумулятор 1(2) младшее (старшее) слово = 16 бит
AWL	Список команд
BARB	Контроль выполнения программы
BARBEND	Конец контроля программы
BCD	Двоично-десятичный код
CPU	Модуль центрального процессора Коммуника-
CP	ционный процессор
DB	Блок данных
DX	Дополнительный блок данных
FB	Функциональный блок
FX	Дополнительный Функциональный ёлок Функ-
FUP	циональная схема
IP	"Интеллигентные" модули периферии
KOP	Релейно-контактная схема
KOR	Координатор
OB	Организационный блок
PA	Отображение процесса
PAA	Отображение процесса на выходах Отображе-
PAE	ния процесса на входах Программатор
PG	Шаговый блок
SB	Стек прерываний
USTACK	Результат логической операции
VKE	

Дополнительная литература

Следующие руководства содержат введение в программирование на языке STEP 5 и документацию по использованию стандартных Функциональных блоков:

Программирование логического управления на STEP 5

Том 1, программирование основных функций
SIEMENS AG, ISBN 3-8009-1407-7
Том 2, использование стандартных Функциональных блоков
SIEMENS AG, ISBN 3-3009-1373-9
Том 3, программирование функциональных блоков
SIEMENS AG, ISBN 3—3009-1366-6.

1 Пояснения

1.1 Применение

Программируемый контроллер (ПК) 135U представляет собой высокоэффективное многопроцессорное устройство, предназначенное для автоматизации процессов (разомкнутого и замкнутого циклов управления, сигнализации, контроля, регистрации).

Его можно использовать как для простейшего логического управления, так и для решения сложным задач автоматизации, для его программирования используется язык STEP 5.

Центральное устройство контроллера может быть оборудовано следующими модулями:
— одним модулем центрального процессора (CPU) для однопроцессорного режима работы или
— одним координатором (KOR) и несколькими модулями CPU (в количестве до 4) для многопроцессорной работы, а также
— процессорами связи или коммуникационными процессорами (CP) количеством до S для однопроцессорного режима или от 4 до 7 для многопроцессорного

Оставшиеся свободными ячейки центрального устройства 135U могут быть заполнены модулями входов/выходов. В соответствии с конфигурацией периферии к центральному устройству может быть подключено устройство расширения.
Распределение мест установки и возможные комбинации центральным устройств и устройств расширения имеется в каталоге ST 54.

1.2 Однопроцессорный режим

Имеется два типа процессоров, предназначенным для решения задач автоматизации и ориентированным на управление или регулирование:
— S-процессор - для задач управления или побитовой обработки
— R-процессор - для задач регулирования или пословной обработки
В однопроцессорном режиме для обычным задач автоматизации в зависимости от типа задачи в контроллере используется либо S- либо R-процессор. Обмен данными с модулями периферии и коммуникационными процессорами производится по шине S5.

1.3 Многопроцессорный режим

Для сложным задач автоматизации предусмотрена одновременная установка в центральное устройство нескольким S- или R-процессоров для работы в многопроцессорном режиме.

В многопроцессорном контроллере каждый CPU обрабатывает свою часть программы пользователя.

При установке более одного CPU необходимо использовать координатор. Он управляет передачей данным по шине S5. Для этого пользователь обязан для каждого CPU запрограммировать блок данным DB 1 (см. Раздел 2.3).

Для обмена данными между отдельными CPU в координаторе создается область маркеров межпроцессорного обмена данными.

2. Программирование

2.1 Язык программирования и структура программы

2.1.1 Язык программирования STEP 5

Использование языка STEP 5 делает возможным программирование целого ряда Функций - от двоичной логики по основных арифметических операций и сложных числовых расчетов.

Программа может быть представлена тремя способами; в виде релейно-контактной схемы (КОР), в виде функциональной схемы (FUP) и в виде списка команд языка STEP-5 (AWL). Выброс способа записи программы определяется конкретной задачей. Машинные коды команд для всех трех способов одинаковы. Используя определенные правила совместимости видов представления, с помощью программатора можно перевести программу из одного вида представления в другой (см. Рис.1).

Команды расширенных операций могут быть запрограммированы только в функциональных блоках и представлены только в виде AWL.

2.1.2 Структура программы

Полная программа контроллера состоит из системной программы и программы пользователя. Системная программа содержит инструкции и установки для внутренних Функций (например, хранение данных в случае сбоя питания, сообщения и т.п.). Эта программа — неотъемлемая часть контроллера (содержится в EPROMe) и не может быть изменена пользователем.

Программа пользователя состоит из команд и указаний, запрограммированных пользователем для обработки и выдачи сигналов, необходимых для управления (регулирования) процессом.

ПК 1351-1 позволяет осуществлять структурное программирование, т.е. полная программа делится на отдельные программные блоки. Такой метод дает пользователю следующие преимущества:

— простое и наглядное программирование даже сложных программ,
— программные блоки могут быть стандартизированы
— простая организация программ
— легко модифицируется программа
— простое тестирование программы
— простота эксплуатации

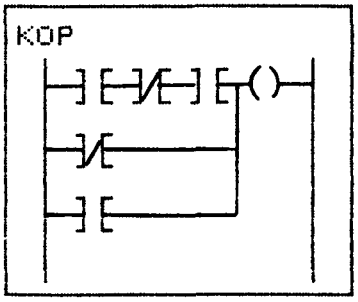
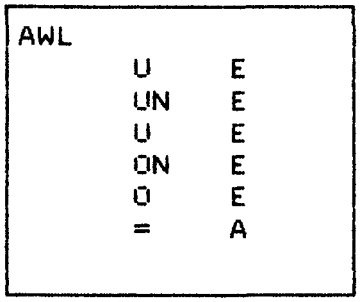
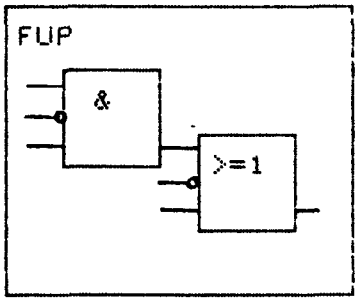
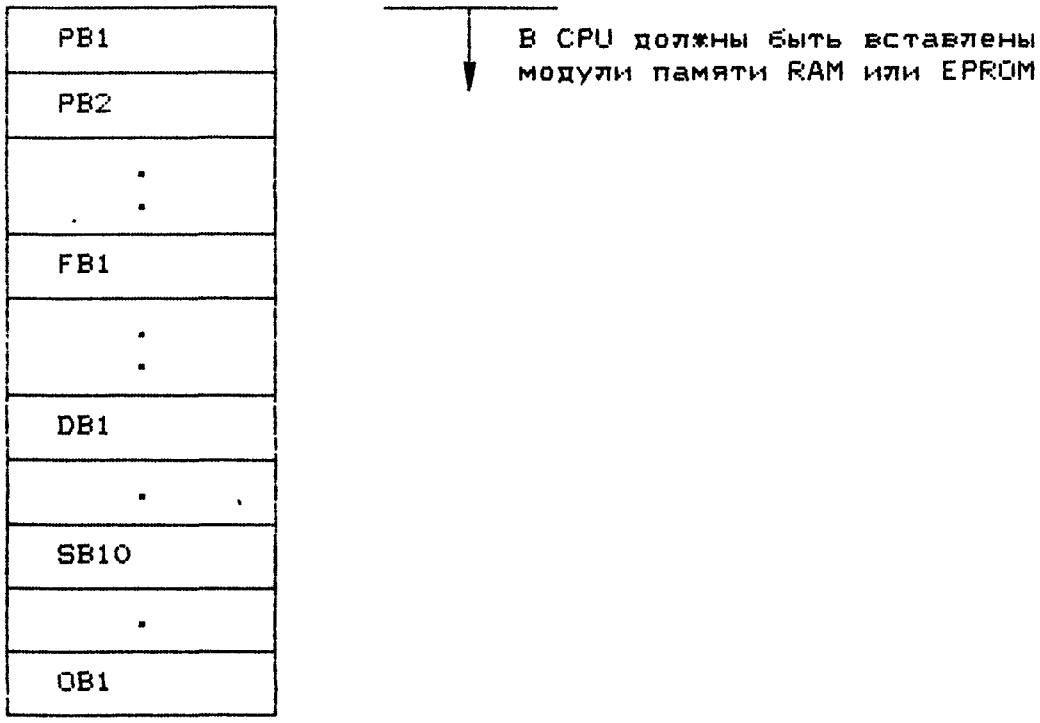
Релейно-конт. схема	Список команд	Функциональная схема
Программируется с помощью графических символов, как схема прохождения тока. Соответствует DIN 19239 (проект)	Программируется с помощью мнемонических сокращений функциональных операций. Соответствует DIN 19239 (проект)	Программируется с помощью графических символов. Соответствует IEC 117-15 DIN 40700 DIN 40719 DIN 19239 (проект)
		

Рис. 1 Виды представления языка программирования STEP 5.



Размещение блоков в программной памяти (порядок любой)

для составления программы можно использовать различные типы программных блоков (каждый для своего круга задач):

- Организационные блоки (OB)

Они обеспечивают связь между системной программой и программой пользователя. Есть организационные блоки, которые могут программироваться пользователем. Они программируются в некоторых ситуациям, когда должна учитываться реакция пользователя. Есть также организационные блоки, которые недоступны пользователю, они содержат системные программы специальным функций, и им можно лишь вызывать (см. Раздел 2.8).

- Программные блоки (PB)

Программные блоки PB содержат части программы пользователя, отражающие отдельные технологические операции (см. Раздел 2.5).

- Функциональные (FB) и дополнительные функциональные блоки (FX).

Функциональные блоки FB содержат программы часто используемых или наиболее сложных функций (например, внутреннего контроля, сигнализации, арифметических или функций контроля замкнутого цикла). Исключение: FB0 (см. Раздел 2.7).

o Шаговые блоки (SB)

Шаговые блоки используются для программирования каскадов или цепочек процесса.

o Блоки данных (DB) и дополнительные блоки данные (DX)

Блоки данных используются для хранения данных или текстов. Функции блоков данных коренным образом отличаются от других блоков.

DBO, DB1, DB2 и DXO зарезервированы для специфических целей (см. Раздел 2.6).

Можно запрограммировать максимум 256 программных, функциональных и шаговых блоков, 253 блока данных и 39 организационных блоков. Каждый блок может занимать в программной памяти максимум 4096 слов. В случае приема/передачи блоков с помощью PG необходимо принимать во внимание объем памяти PG.

Все запрограммированные блоки могут быть в любом порядке занесены программаторам в программную память, которая в виде модуля RAM или EPROM вставляется в CPU (Рис. 2).

2.2 Организация, хранение и обработка программы

2.2.1 Организация программы

Организация программы зависит от того, как, и в какой последовательности должны обрабатываться блоки программы пользователя (рис. 3). Переходы (условные и безусловные) к тем или иным блокам программируются в организационных блоках.

Дополнительные программные функциональные и шаговые блоки могут быть вызваны определенным сочетанием организационных, программных, функциональных и шаговых блоков.

Максимальная глубина вложений — 24 уровня. Эту величину можно рассматривать, как результирующую глубину обращения, с учетом всех возможных операций (циклических, обработки прерываний, временных функций и т.п., см. Разделы 3.6—3.7).

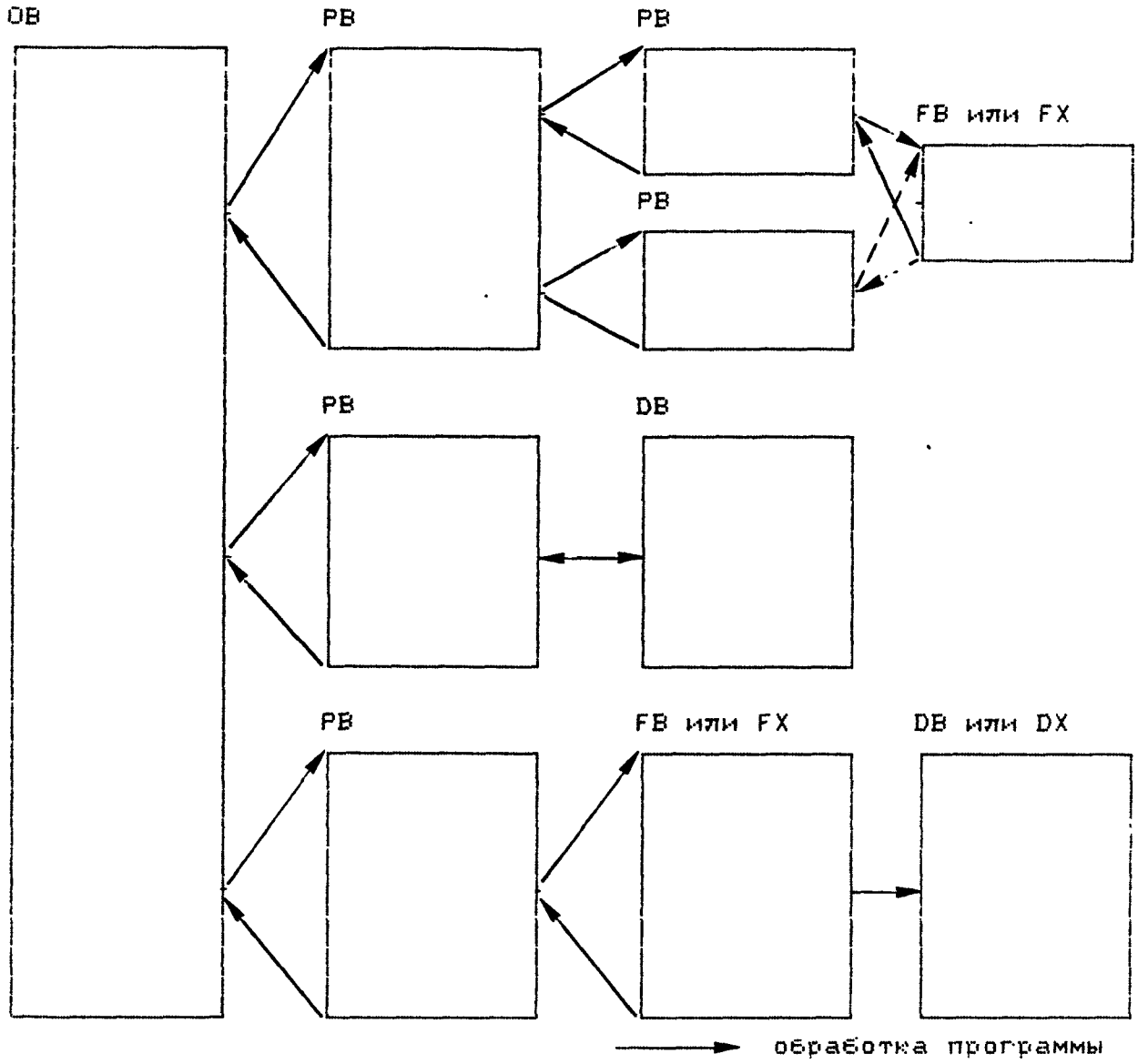


Рис. 3 Организация программы STEP 5

2.2.2 Хранение программы

Если в CPU .используется субмодуль RAM, программу пользователя можно непосредственно передать из PG в CPU. Запрограммированные блоки могут содержаться в памяти в любом порядке. Когда субмодуль RAM полон, следующие блоки данные помещаются в область памяти RAM CPU (RAM для блоков данных см. Рис.17). Объем RAM CPU — 3792 слова. Но при использовании регистров сдвига этот объем сокращается на 128 слов; по мере заполнения RAM CPU адресация смещается в сторону уменьшения.

Если используется блок EPROM, то все запрограммированные блоки должны быть занесены в него. Блоки данным, которые содержат изменяющиеся в процессе выполнения программы данные, должны быть скопированы с субмодуля EPROM в область RAM CPU во время нового пуска (см. Раздел 7.1). Исключение составляют блоки данным DBO и DB1, которые задействованы в системной программе.

2.2.3 Выполнение программы

Обработка программы пользователя может идти по одному из трех путей (Рис. 4):

o Циклическая работа программы (см. Раздел 3.6.1)

при циклической отработке программы задействован либо организационный блок OB1 , либо Функциональный блок FBO;

— OBI выполняется циклически, вызывая блоки программы пользователя.

— FBO функционирует также, как OB1, но, в дополнение к ЭТОМУ, он дает возможность пользоваться некоторыми дополнительными операциями. FBO больше всего подводит для выполнения небольшим, ограниченных по времени программ, которые не требуют структурного программирования, а также не содержат обращения к блокам.

Если запрограммированы и OB1 и FBO, то работать будет лишь OB1.

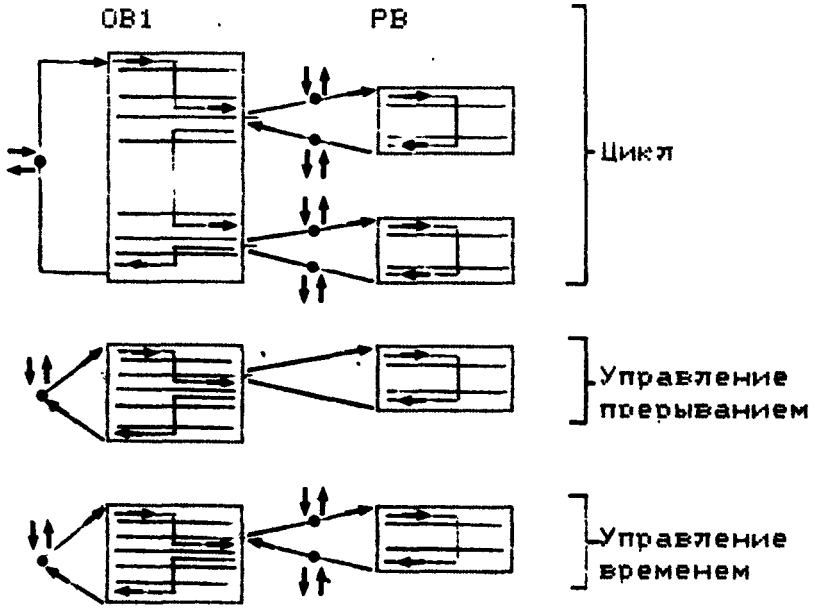
o Обработка программы прерывания (см. Раздел 3.6.2)

В таком режиме работы циклическая обработка программы прерывается по инициативе внешних устройств. Для вызова стандартным программ прерывания служит OB2.

o Обработка программы, управляемой по времени (см. Раздел 3.6.3)

В этом режиме определенные участки программы (вызываемые OB13) автоматически включаются в циклическую программу, используя временную установку (см. Рис. 4).

Такой режим работы необходим, когда решаются задачи управления с замкнутым циклом.



Точка прерывания, в которой прекращается обработка циклической программы, предыдущей программы прерывания или программы, управляемой по времени, и начинается выполнение актуальной программы прерывания или программы, управляемой по времени. Программа, управляемая по времени, может быть прервана только программой обработки прерывания, но не наоборот (см. Раздел 3.6.3)

Рис. 4 Типы выполняемых программ

2.3 Программирование в многопроцессорном режиме

Программирование контроллера для многопроцессорного режима сводится к операции программирования каждого CPU, описанной в Разделе 2.2. В дополнение к этому отметим некоторые особенности многопроцессорного режима;

— Процессоры могут обмениваться данными через маркеры межпроцессорной связи.

— Вся программа контроллера может быть разбита на индивидуальные программы для каждого CPU в соответствии с общей задачей.

— Каждому CPU должны быть назначены свои периферийные устройства ввода/вывода.

— Доступом каждого CPU к шине контроллера управляет координатор. Количество работающих CPU устанавливается при помощи перемычек в координаторе (см. Руководство по координатору).

Распределение программы:

Процессоры максимум до 4) в многопроцессорном режиме выполняют каждый свою программу независимо от другим, что дает возможность пользователю разделить всю программу контроллера на индивидуальные программы для каждого CPU. Таким образом, многопроцессорный режим дает следующие преимущества:

— Разделение программы между CPU, которые затем работают параллельно, сокращает время выполнения всей программы.

— Программы с малым временем выполнения для обработки быстрых процессов могут быть помещены каждая в своем CPU. В этом случае время обработки программы пользователя в дальнейшем можно сократить за счет использования FBO вместо OB1,а также за счет точного определения длины таймерного набора (см. Раздел 2.4").

— Программы пользователя с большим временем выполнения для обработки процессов, которые некритичны ко времени, могут быть запрограммированы в своим "медленным" CPU отдельно от "быстрым".

— Каждый CPU может быть ориентирован на свою часть станка в зависимости от ее функции. Установив M-, R- и S- процессоры в один контроллер, можно оптимальным образом решать задачи регулирования и управления.

2.3.1 Маркеры межпроцессорной связи

Маркеры межпроцессорной связи представляют собой байты, которые назначаются пользователем в CPU как входные и выходные. Они предназначены для побайтного циклического обмена данными между CPU и располагаются в координаторе.

Байт маркеров связи определяется как выходной для CPU, который передает данные в циклическом режиме посредством координатора и как входной (с тем же номером) для CPU, принимающего данные.

При работе с маркерами связи необходимо соблюдать следующие правила:

— Маркер, назначенный для одного или нескольких CPU, как входной, должен быть определен для другого CPU, как выходной.

— Если в одном CPU маркер назначен в качестве выходного, то он не может быть определен, как выходной для других CPU (только как входной).

— Маркерные байты, определенные, как маркеры связи в каком-либо CPU, пригодны в этом CPU лишь для передачи данных. Остальные маркеры, не определенные, как маркеры связи, могут использоваться по своему обычному назначению.

— Можно указывать только маркерные байты, которые установлены в координаторе или коммуникационных процессорах (см. Руководство по эксплуатации KOR и CP's).

— Маркер связи задействовать не обязательно.

Маркеры связи можно использовать также для передачи данных между CPU и CP (коммуникационными процессорами). Это возможно как в однопроцессорном, так и в многопроцессорном режиме.

Область маркеров связи, имеющая максимальный объем 256 входных и выходных байтов, может быть разбита в координаторе и/или CP на 32-байтные области (см. Инструкцию по эксплуатации координатора и коммуникационного процессора). Если маркеры связи на KOR или CP не квитируются, то необходима обработка прерываний (см. Раздел 3.7). Ввод списка адресов маркеров связи в DB1 описан в Разделе 2.6.

2.3.2 Распределение периферии

В многопроцессорном режиме пользователь должен распределить (назначить) периферийные модули входов/выходов (только P-периферию с относительными адресами байтов 0...127) и, побайтно, необходимые маркеры связи для каждого CPU. Для этой цели существует блок данных DB1, в который пользователь заносит распределение входов/выходов и маркеров в виде таблицы адресов, используемых в PG (см. Раздел 2.6). При однопроцессорной работе DB1 можно запрограммировать таким образом, чтобы время выполнения программы было оптимизировано.

При программировании DB1 как в одно-, так и в многопроцессорном режимах необходимо помнить следующие правила:

– После того, как список адресов был сформирован в CPU при новом старте, обращение к модулям периферии через область отображения процесса допускается только для адресов байтов, указанных в DB1, (команды L/T E, EB, ED, A, AB, AD и логические операции со входами/выходами).

- Непосредственный доступ к периферийным модулям в обход области отображения процесса с помощью команд загрузки и передачи (L/T PB, PM, QB, QW) возможен для загрузки *всех* вводов. квитируемых на соответствующих модулях независимо от списка адресов.

- Непосредственная передача возможна только для выходов, указанных в DB1, если имеется список адресов, т.к. при непосредственной передаче область отображения процесса описывается дополнительно. Область отображения процесса имеется только для P-периферии с относительными адресами байтов O...127.

При новом старте операционная система воспринимает введенный пользователем адресный список и проверяет, отвечают эти указанные в DB1 входы, выходы на соответствующих модулях или маркеры связи процессору. Если нет, то CPU переходит в состояние "СТОП" с сообщением об ошибке DB1 (см. Раздел 3.7) с медленным мерцанием светодиода STOP. Программа пользователя не обрабатывается.

2.4. Оптимизация программы пользователя и ограничения по времени

Оптимизация программы пользователя

o Структура программы

В однопроцессорном режиме, так же как и *в* многопроцессорном, время обработки программы пользователя можно уменьшить, если просто применить структурное программирование.

Поскольку каждое изменение блока требует дополнительного времени, структурного программирования следует избегать при создании коротких, критичных по времени программ и использовать только FBO. В FBO можно использовать весь набор команд контроллера ПК 135U (см. Раздел 4).

o Назначение (установка) входов/выходов

При распределении периферии в однопроцессорном и в многопроцессорном режимах очень важно, чтобы в списке адресов входов/выходов и маркеров связи были указаны только те адреса, к которым имеет доступ программа данного CPU.
Наличие в адресном списке адресов входов/выходов и маркеров связи, которые не используются программой, но периодически опрашиваются в соответствии с DB1, неоправданно увеличивает время обработки программы пользователя.

o Длина таймерного набора

В DB1 пользователь может указать также количество таймерных позиций, входящих в таймерный набор. В этом случае время обновления таймеров, не вошедших в таймерный набор, будет сэкономлено, время цикла (см. Раздел 3.6..1) сократится.

Однако, это возможно лишь в том случае, если количество таймеров, задействованных пользователем, меньше числа, указанного в длине таймерного набора.

Если длина таймерного набора = 0, то таймеры вообще не будут обрабатываться. Если не указывать длину таймерного набора, тогда будут доступны все таймеры от О до 127.

В однопроцессорном н многопроцессорном режимах также можно указать длину таймерного набора. Но в этом случае пользователь должен записывать в DB1 полный адресный список, т.е. указывать н адресный список входов/выходов.

Если обрабатывается таймер, находящийся за пределами таймерного набора или номер которого больше 127, то CPU воспринимает это как ошибку параметрирования (см. Раздел 3.7).

Ограничения

Стандартные функциональные блоки занимают номера с О по 199. Поэтому, функциональные блоки пользователя могут быть созданы только с номерами от 200 до 255, если имеются стандартные функциональные блоки.

FBO может быть использован только для программирования циклической обработки программ (см. Раздел 3.1).

Если имеются стандартные функциональные блоки, то маркерные байты с 200 по 255 будут заняты и не могут быть использованы пользователем.

Блоки данных DB 0, 1, 2 и DX 0 зарезервированы для определенных целей:

- DB 0 создается операционной системой и содержит список адресов всех; блоков загруженной программы пользователя.
- DB 1 задается пользователем как список адресов и обрабатывается операционной системой.
- DB 2 служит для параметрирования компактной регулировки, которая может быть заказана в виде пакета программ. Он написан на языке Ассемблера в целях оптимизации по времени и поддерживается дополнительно операционной системой R-процессора (см. Руководство пользователя по компактной регулировке в AG 135U/R-процессор, Каталоговый No 6ES5 842-OBB10).
- Дополнительный блок данных DX О служит для предварительной установки пользователем определенных функций операционной системы (см. Раздел 2.6).

Для организационных, программных и шаговых блоков можно использовать только основной набор операций STEP 5. Команды дополнительного набора программируются только в функциональных блоках FB и FX.

2.5 Программные блоки

2.5.1 Программирование программных блоков

Следующее описание подходит для программирования организационных, программных и шаговых блоков. Эти три типа блоков в программировании почти не отличаются. Они могут программироваться в виде релейно-контактной схемы (KOP), функциональной схемы (FUP) и в виде списка команд (AWL) языка STEP 5. Программирование начинается с присвоения номера блоку:

— Программные блоки O...255
— Шаговые блоки O...255
— Организационные блоки 1...39 (см. Раздел 2.8)

Затем идет непосредственно программа управления, которая завершается командой "BE". Для программирования может использоваться только набор функций языка STEP 5. Блок программы STEP 5 не может занимать более 2000 слов в памяти пользователя.

Заголовок блока автоматически записывается программатором в начало блока. Он занимает 5 слов в памяти пользователя.

Программный блок обязательно должен содержать законченную программу. Попытка выйти за границу блока, чтобы продолжить программу, воспринимается как ошибка.

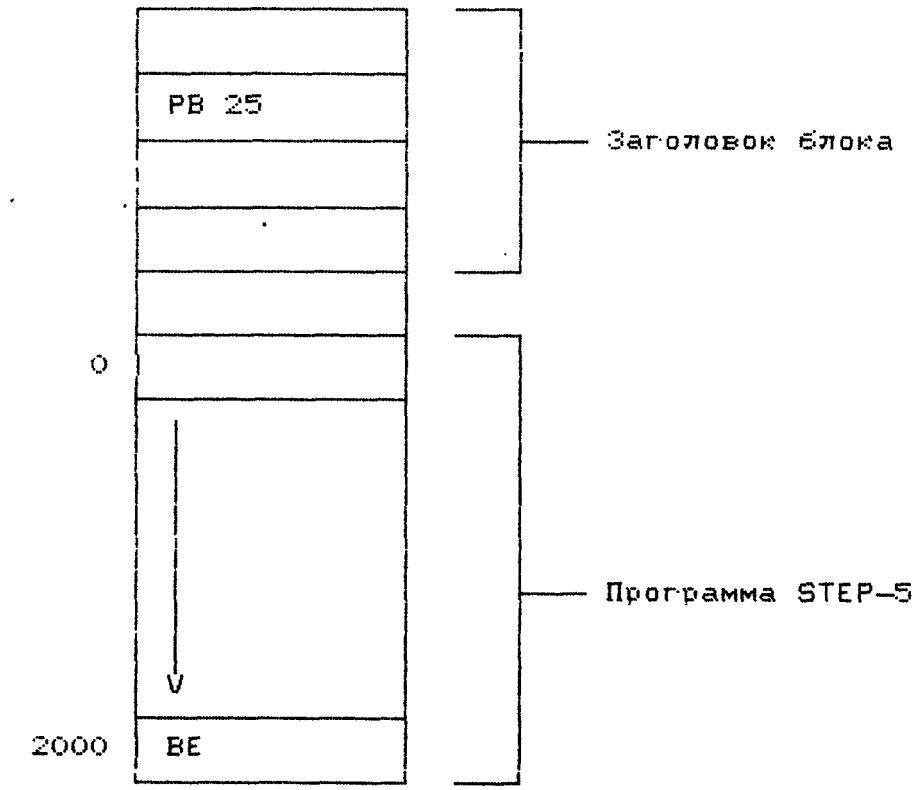


Рис. 5 Структура организационного, программного и шагового блоков

2.5.2 Вызов программных блоков

Программные блоки активируются с помощью операций вызова (Рис.6). Эти вызовы можно запрограммировать в любом программном, функциональном или шаговом блоке. Они напоминают команды перехода к подпрограмме и могут быть безусловными (SPA PBX) или условными (SPB PBX).

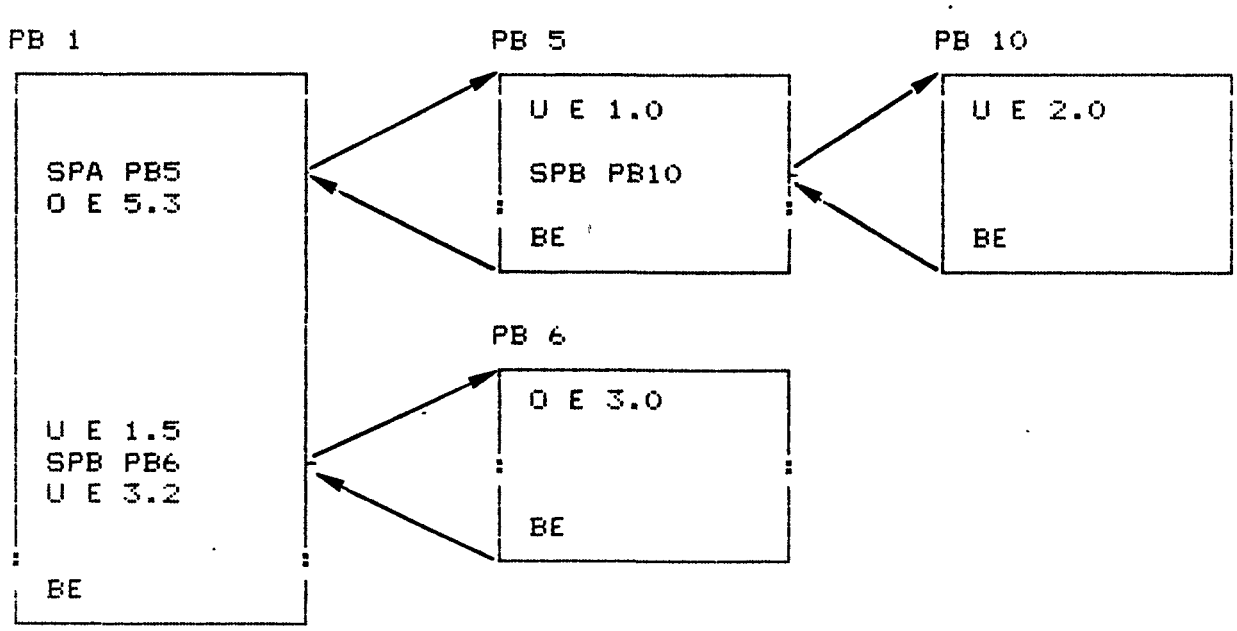


Рис.6 Вызовы блоков, которые выполняют обработку программных блоков.

После команды "BE" (конец блока.) происходит возврат в блок, из которого произведен вызов, и выполняется команда STEP 5, которая следует за вызовом блока. При этом после команд вызова блока или "BE" невозможно использовать результат логической операции: VKE (см. Раздел 4.1) для последующий операций, т.к. эти команды ограничивают VKE. Однако VKE тоже переходит в "новый" блок и там может быть использован.

o Безусловный вызов: SPAxx

Программный блок вызывается независимо от результата предыдущей логической операции (VKE).

o Условный вызов: SPBxx

Вызов программного блока зависит от результата предыдущей логической операции.

Если VKE=1, команда перехода выполняется, если VKE=0 - нет. В обоих случаях команда перехода устанавливает VKE в 1. Такая же зависимость от VKE для команды условного конца блока BEB.

2.6 Блоки данных

2.6.1 Программирование блоков данных

В блоках данных хранятся данные, необходимые для обработки программы пользователя. В блоках данных не может быть команд языка STEP-5. Данными могут быть:

— последовательность битов;
— числа (десятичные, двоичные), например, для таймеров, результаты арифметических операций;
— буквенно-цифровые значения, например, для текстовых сообщений.

Блоки данных имеют такую же структуру, как и программные. Программирование начинается с указания номера блока данных (максимум 255, например DB25). Каждый блок данных состоит из 16--битовых слов данных, которые должны вводиться в блок в порядке возрастания нумерации слов данных, начиная с нулевого слова DW0 (см. Рис. 7).

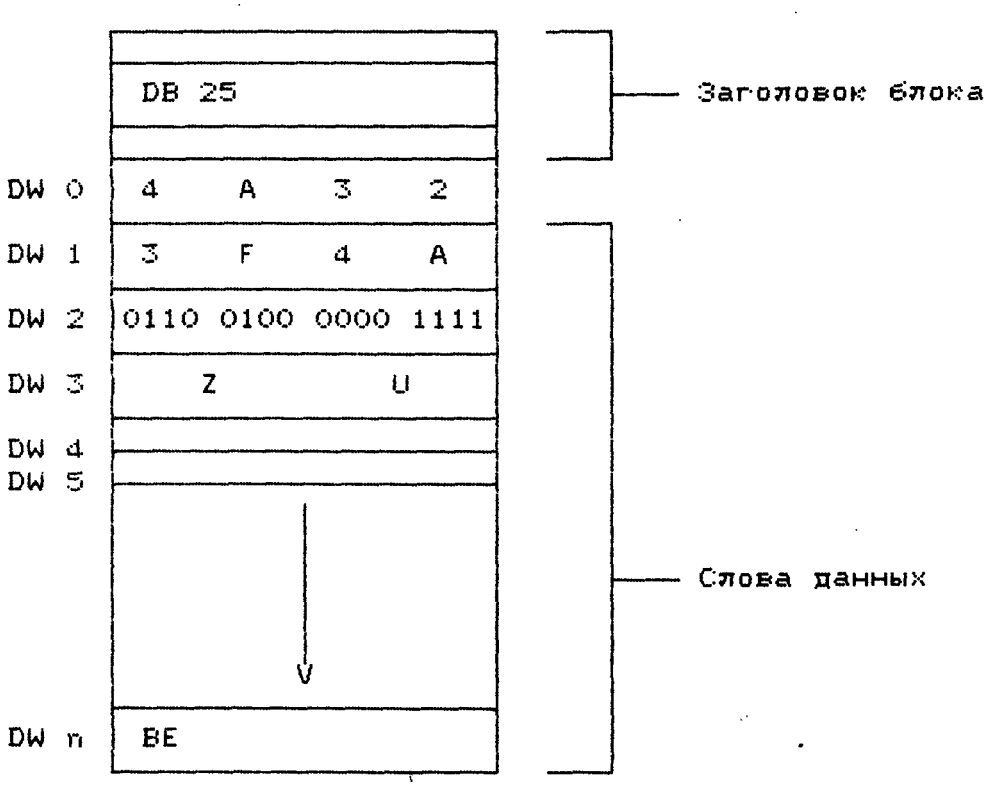


Рис.7 Структура блока данных

Одно слово в памяти резервируется для слова данных в памяти программ. Следующие пять слов в памяти программ занимает. заголовок блока, создаваемый программатором для каждого блока данных. Блок данных может занимать максимально 4096 слов в программной памяти CPU. При передаче и приеме данных с программатора следует учитывать его объем памяти.

Внимание ! С. помощью команд загрузки/передачи L/T DW, можно обращаться к словам данных с номерами до 255. Блоки данных DB 0, 1, 2 зарезервированы для определенных функций, и не могут быть задействованы пользователем. Дополнительные блоки данных DX создаются и используются как DB, но DX O зарезервирован (см. Раздел 2.6.3).

2.6.2 Вызов блоков данных

Блоки данным можно вызывать только безусловно. Вызов остается действительным до тек пор, пока не произойдет следующий вызов. Вызов блока данных можно запрограммировать в организационном, программном, функциональном или шаговом блоке с помощью команды A DB..., а дополнительный блок данных по команде AX DX... . Блоки данных DB O, 1, 2 и DX O обрабатываются операционной системой и не могут вызываться пользователем (см. Раздел 2.6.3).

Внимание ! Перед тем, как выполнить команду загрузки/передачи слова данных, надо вызвать блок данных. Адресуемое слово данных должно находиться в блоке данных, в противном случае операционная система опознает ошибку при передаче (TRAP, см. Раздел 3.7).

Пример

Содержимое слова данных 1 блока данных DB10 переслать в слово данных 1 блока DB20 (Рис. 8).
Если программный блок, в котором уже был адресован блок данных, вызывает другой программный блок, а в этом блоке адресован другой блок данных, то последний блок данных действителен лишь в вызванном программном блоке. После возвращения в вызвавший программный блок вновь начинает действовать предыдущий блок данных (Рис.9).

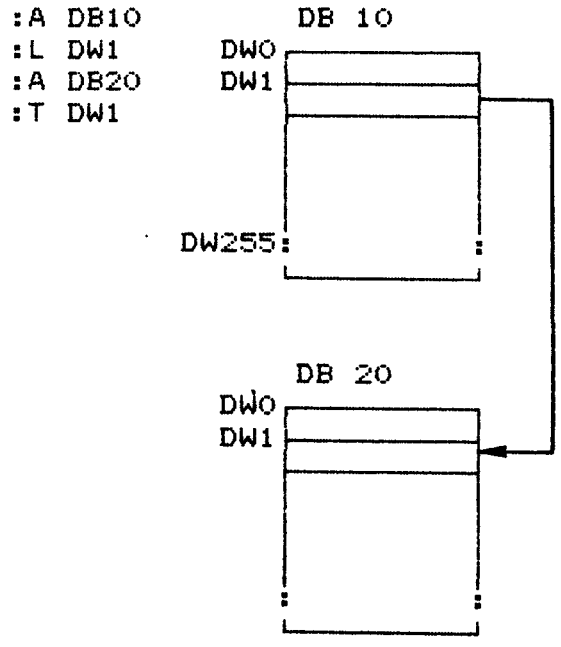


Рис.8 Адресация блоков данных

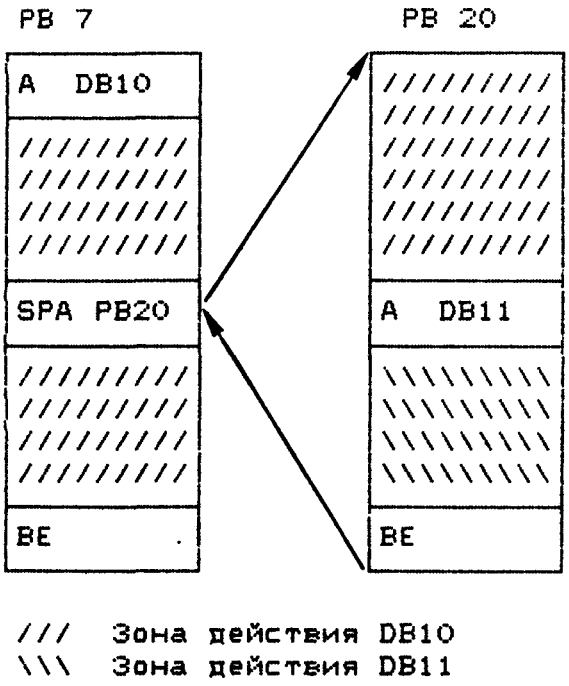


Рис.9. Зона действия выбранных данных блоков данных

Если программный блок, в котором уже был адресован блок данных, вызывает другой программный блок, а в этом блоке адресован другой блок данный, то последний блок данных действителен лишь в вызванном программном блоке. После возвращения в вызвавший программный блок вновь начинает действовать предыдущий блок данные (Рис.9).

Пример (см. Рис.9)

В программном блоке 7 вызван DB10. В дальнейшем обрабатываются данные этого блока.

Следующим для обработки вызывается PB20. Однако, DB10 по-прежнему является действительным. Только после вызова OB11 меняется область данных. DB11 действует до конца PB20.

После перехода назад в PB7 опять становится действительным DB10.

2.6.3 Специальные блоки данных

Блоки данных DB O, 1, 2 и DX O зарезервированы для определенных функций. Они обрабатываются операционной системой и не могут быть задействованы пользователем:

- Блок данных DB O содержит список адресов с начальными адресами всех блоков, которые находятся в субмодуле памяти пользователя или в памяти RAM блоков данных CPU. Этот адресный список создается операционной системой при инициализации (см. Раздел 3.2), и при вводе или изменении блоков с помощью PG автоматически обновляется.

- Блок данных DB 1 содержит список цифровых входов и выходов (P - периферия с относительными адресами байтов - от O до 127), а также входных и выходных маркеров связи, принадлежащих CPU, и, в данном случае, длину таймерного набора (см. Раздел 3.3). DB 1 создается пользователем. При новом старте операционной системой создаются внутренние списки адресов. Если указанные в DB 1 входы, выходы или маркеры связи отсутствуют в соответствующих блоках, то CPU переходит в состояние "СТОП" с сообщением об ошибке DB 1 (см. Раздел 3.7).

- Блок данных DB 2 служит для параметрирования пользователем компактной регулировки. Компактная регулировка может быть заказана в виде пакета программ и работает с поддержкой операционной системы.

- Дополнительный блок данных DX O служит для предварительной установки пользователем определенных функций операционной системы.

o Блок данных DB1

Ввод/изменение DB 1 возможно:

— Посредством PG, когда CPU в состоянии "СТОП", и если контроллер оснащен памятью пользователя RAM. DB1 вводится или изменяется только во время нового пуска (см. Раздел 3.5.1).

— Посредством программирования модуля EPROM.

Организация DB 1 с помощью PG

Слова данных 0, 1 и 2 должны быть установлены соответственно:

КН = 4D41, 534B, 3031.

После третьего слова идут списки адресов. Каждый адресный список начинается с ключевого слова. Ключевыми словами являются:

для цифровых входов	КН = DE00
Для цифровых выходов	КН = DA00
Для входных маркеров связи	КН = CE 00
Для выходных маркеров связи	КН = CA00

Следом за ключевым словом идут в формате данных с фиксированной запятой адреса установленных входов/выходов и маркеров, принадлежащие соответствующему списку адресов. Порядок записей в списке адресов произвольный, так же как и самих списков.

Задание длины таймерного набора производится через ключевое слово:

КН = BBOO

Затем вводится длина таймерного набора в формате с фиксированной запятой.

Последним, как символ окончания, в DB1 должно быть записано слово

КН = EE 00

В многопроцессорном режиме DB1 должен быть организован для каждого CPU. В однопроцессорном режиме он может быть использован для программирования оптимального времени обработки. Если DB 1 был запрограммирован, то должны быть указаны входы и выходы, т.к. только они циклически обновляются, на маркеры связи и длину таймерного набора это не распространяется.

Пример: для DB1

0	KH = 4D41;	
1	KH = 534B;	
2	KH = 3031;	
3	KH = DEOO;	Ключевое слово цифровых входов
4	KF = +00000;	Байт входов 0
5	KF = +00001;	Байт входов 1
6	KF = +00002;	Байт входов 2
7	KF = +00003;	•
8	KF = +00007;	•
9	KF = +00010;	•
10	KH = DAOO;	Ключевое слово цифровых выходов
11	KF = +00000;	Байт выходов 0
12	KF = +00002;	Байт выходов 2
13	KF = +00004;	•
14	KF = +00012;	•
15	KH = CEOO;	Ключевое слово выходных маркеров связи
16	KF = +00050;	Байт маркеров 50
17	KH = +00051;	•
18	KF = +00060;	•
19	KH = CAOO;	Ключевое слово входных маркеров связи
20	KF = +00070;	Байт маркеров 70
21	KF = +00072;	•
22	KF = +00100;	•
23	KH = BBOO;	Ключевое слово длины таймерного набора
24	KF = +00040;	
25	KH = EEOO;	Идентификатор окончания

Начиная с версий матобеспечения PG 675 SO A03 или S1 A01 можно, используя функциональные клавиши F1 и F3, работать с экранной формой записи DB1.

НАЗНАЧЕНИЕ ПЕРИФЕРИИ:

ЦИФРОВЫЕ ВХОДЫ	,	0,	1,	2,	3,	7,	10,	,	,	, ,
ЦИФРОВЫЕ ВЫХОДЫ	,	0,	2,	4,	12,	,	,	,	,	, ,
ВХОДНЫЕ МАРКЕРЫ СВЯЗИ	,	50,	51,	60,	,	,	,	,	,	, ,
ВЫХОДНЫЕ МАРКЕРЫ СВЯЗИ	,	70,	72,	100,	,	,	,	,	,	, ,
ДЛИНА БЛОКА ТАЙМЕРОВ	,	40,	,	,	,	,	,	,	,	, ,

•

o Дополнительный блок данных DX O

С помощью DX O пользователь может согласовать определенные Функции операционной системы со своими требованиями, заменив в нем стандартные предустановкой (V) своими. При этом, следует помнить, что стандартные предустановки (V) устанавливаются только при новом пуске, а ввод или изменение DX O возможны только в режиме запуска.

Построение DX 0

Блок DX 0 делится на информационные блоки различной длины.

Каждый блок начинается с метки блока, указания длины информационного блока в словах и, собственно, информации. Каждый блок соответствует определенной функции операционной системы. С помощью указания длины возможно определить объем информационного блока. Информация состоит из 1 до n слов данных, значение которых зависит от типа блока. По завершении последнего блока в качестве метки окончания устанавливается EEEE.

Формальная конструкция:

No бита	15 8	7 0	
DW 0	'M'	'A'	Символы 4D41
1	'S'	'K'	ASCII 534B
2	'X'	'O'	5830
3	Метка блока 1, длина информ. блока 1		
.	Параметр		Информация
.	Параметр		
.	Параметр		
.	Метка блока 1, длина информ. блока 1		
.	Параметр		Информация
.	Метка блока 1, длина информ. блока 1		
.	Параметр		Информация
.	Параметр		
.	Параметр		
m	Метка окончания = EEEE		

Система может определить, какие информационные блоки действительны в данный момент (ИНИЦИАЛИЗАЦИЯ, СТОП, ЗАПУСК).

Для опций процесса регулирования, выбираемых пользователем, специфицируются выполняемые в дальнейшем блоки. При этом действуют следующие правила:

- Не следует вводить ненужные блоки и параметры.

- Блок может встречаться в DX 0 несколько раз; в этом случае действует последнее параметрирование.

- Последовательность параметров блока соблюдать не обязательно;
при многократном назначении одного параметра действителен последний ввод.

- При новом пуске параметры устанавливаются сначала таким образом, чтобы определить нужные функции или реакцию операционной системы. При всех трех видах запуска обрабатывается DX 0, и устанавливаются соответствующие параметры.

Значения вводятся в шестнадцатиричном формате. Изменяемые функции операционной системы описаны в Разделе 3.

Метка блока	Параметр	Значение [2]
Обработка запуска (см. Раздел 3.5):		
02xx [1]	1000	V Автоматический перезапуск при включении напряжении
	1001	Автоматический новый пуск при включении напряжения
	2000	V Синхронизация запуска в многопроцессорном режиме
	2001	Отсутствие синхронизации запуска в многопроцессорном режиме (временная синхронизация (см. Раздел 3.5.4)
	3000	V Контроль ошибки адресации
	3001	Отсутствие контроля ошибки адресации
	BB00 00yy	Установка количества активизируемых таймеров (см. Длина таймерного набора в DB 1; Предустановка: 128 таймеров Допустимо: от 0 до 128 таймеров
Циклическая обработка (см. Раздел 3.6.1):		
04xx [1]	1000 yyyy	Длина временного цикла в миллисекундах Предустановка: yyyy = 150 мС Допустимо: от 1 до 4000 мС
	4000	V Актуализация области отображения маркеров связи без семафорной защиты
	4001	Актуализация области маркеров связи с семафорной защитой (см. Раздел 5.9)

[1] xx = длина информационного блока (в словах)

[2] V = предустановка при новом пуске

Метка блока	Параметр	Значение *)
Обработка прерываний (см. Раздел 3.6.1):		
06xx [1]	100C	V Обработка прерываний от регулятора, по времени и аппаратных на границе блока
	1006	Обработка прерываний от регулятора, по времени и аппаратных на границе команды
	1008	Обработка прерываний от регулятора и аппаратных на границе команды, по времени - на границе блока
	100A	Обработка аппаратных прерываний на границе команды, от регулятора и по времени - на границе блока
	2000	V Сигнал аппаратного прерывания, установка по уровню
	2001	Аппаратное прерывание, установка по фронту
Обработка ошибок (см. Раздел 3.7):		
10xx		Обработка ошибок по времени
	1000	V Системный "СТОП", если результат есть, а соответствующий OB не загружен
	1001	Нет останова системы, если результат есть, а соответствующий OB не загружен
		Обработка ошибок регулятора
	1200	V Системный "СТОП", если результат есть, а соответствующий OB не загружен
	1201	Нет останова системы, если результат есть, а соответствующий OB не загружен
		Обработка ошибок цикла
	1400	V Системный "СТОП", если результат есть, а соответствующий OB не загружен
	1401	Нет останова системы, если результат есть, а соответствующий OB не загружен

[1] xx = длина информационного блока (в словах)
[2] V = предустановка при новом пуске
Аппаратное прерывание = обработка программы прерывания от процесса
Прерывание по времени = обработка программы, управляемой по времени
Прерывание от регулятора = обработка блока компактного регулятора (см. Руководство по программному обеспечению AG 135U)

Метка блока	Параметр	Значение [1]
		Обработка ошибок кодов команд
	1800	V Системный "СТОП", если результат есть, а соответствующий OB не загружен
	1801	Нет останова системы, если результат есть, а соответствующий OB не загружен
		Обработка ошибок времени обработки [2]
	1A00	V Системный "СТОП", если результат есть, а соответствующий OB не загружен
	1A01	Нет останова системы, если результат есть, а соответствующий OB не загружен
		Обработка ошибки адресации
	1C00	V Системный "СТОП", если результат есть, а соответствующий OB не загружен
	1C01	Нет останова системы, если результат есть, а соответствующий OB не загружен
		Обработка ошибки задержки квитирования
	1E00	V Системный "СТОП", если результат есть, а соответствующий OB не загружен
	1E01	Нет останова системы, если результат есть, а соответствующий OB не загружен
EEEE		Метка окончания

[1] V = предустановка при новом пуске

[2] Ошибка времени обработки = вызов незагруженного блока, общая ошибка специальной функции или ошибка передачи

2.6 Функциональные блоки

Функциональные блоки являются такими же частями программы, как например, программные блоки. дополнительные функциональные блоки (FX) имеют ту же структуру, что и FB, и также программируются. Функциональные блоки имеют 4 существенных отличия от организационных, программных и шаговые блоков;

— Функциональные блоки могут быть параметрированы, т.е. могут быть заданы фактические параметры, с которыми работает FB, заменяющие формальные операнды.
— В противоположность организационным, программным и шаговым блокам для программирования функциональных блоков используется более широкий набор команд. Эти дополнительные команды можно использовать только в функциональные блоках.
— Программа функционального блока может быть представлена только в виде списка команд AWL
— Графически при вызове функциональный блок выглядит как "черный ящик"

Внутри программы пользователя функциональные блоки представляют сложную, законченную функцию. Они могут храниться в операционной системе или на гибких дисках, которые ф.SIEMENS поставляет в качестве стандартного матобеспечения ("стандартные функциональные блоки на мини-флопи-дисках", см. Каталог ST 57), а также могут быть созданы самим пользователем.

2.7.1 Структура Функциональных блоков

Функциональный блок состоит из заголовка и "тела" (Рис.10).

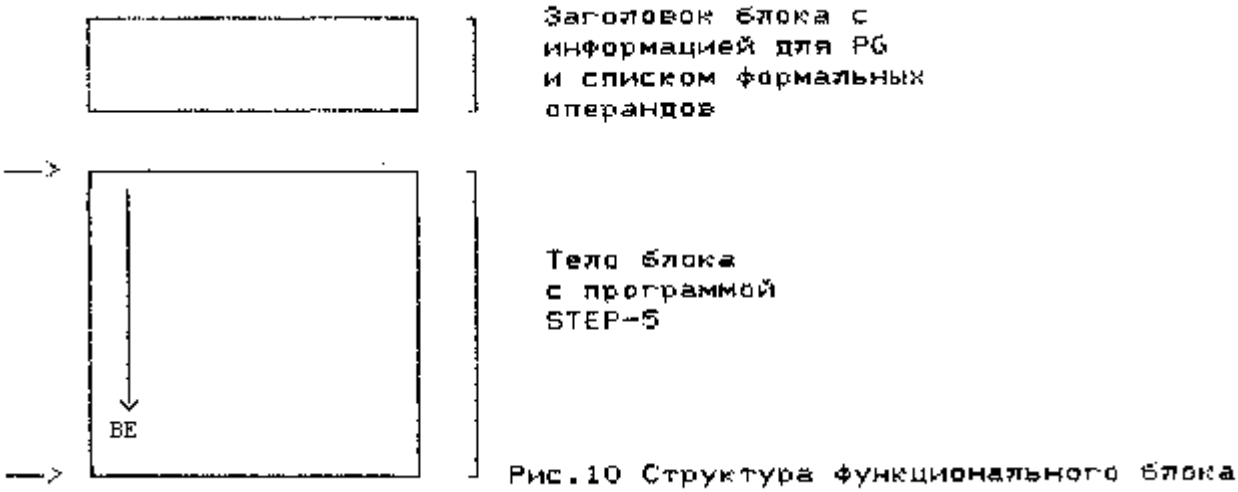


Рис.10 Структура функционального блока

o Заголовок блока

Заголовок блока содержит все данные, которые необходимы для программатора, чтобы представить FB и чтобы проконтролировать операнды при параметрировании FB. Перед программированием FB заголовок вводится пользователем с помощью программатора.

Он хранится в памяти CPU и содержит команды перехода, которые выполняются при вызове функционального блока, но не отображаются при его чтении (переход через список формальные операндов).

o "Тело" блока

"Тело" блока содержит собственно программу FB. Выполняемые функции записаны на языке STEP 5 и хранятся в памяти. При вызове FB выполняется только "тело" блока. Для программирования FB может использоваться более широкий набор команд по сравнению с программными блоками (см. Раздел 4.2).

2.7.2 Вызов и параметрирование функционального блока

Функциональные блоки могут использоваться для повторяющихся или довольно сложных функций. Они загружаются в память один раз и могут вызываться многократно, причем, параметры блока могут меняться от вызова к вызову.

FB записываются в память с определенным номером (от FBO до FB255). Номера FB от 240 до 243 заняты встроенными FB. Функциональные блоки пользователя должны адресоваться в сторону убывания, начиная с FB 255, чтобы не пересечься со стандартными функциональными блоками, имеющими номера с 1 по 199.

FB O должен в отсутствие 0B 1 использоваться только для программирования циклической обработки программы (см. Раздел 3.6.1).

Вызов FB можно программировать в программном или другом FB. Вызов состоит из команды вызова и списка параметров.

o Команды вызова

- Безусловный вызов - SPA FBn (для функционального блока) или BA FXn (для дополнительного функционального блока):
  Вызванный FB обрабатывается независимо от результата предыдущей логической операции (VKE).

- Вызов по условию - SPB FBn (для функционального блока) или BA FXn (для дополнительного функционального блока):

Вызванный FB обрабатывается, только если VKE=1. Если VKE=0, то переход не выполняется. В обоих случаях при условном переходе VKE устанавливается в 1.

После безусловного и условного вызова результат логической операции может быть обработан, но не может использоваться дальше в логических цепочках. При переходе он используется в опрошенном функциональном блоке.

o Список параметров

Список параметров следует непосредственно за командой вызова (Рис. 11). В нем определены входные/выгодные переменные и данные (см. Раздел 2.7.3 "Типы параметров блоков'").

Список может содержать максимум 40 параметров. Они присваивают формальным параметрам (операндам) функционального блока значения переменных (Фактические операнды, см. пример ниже).

При обработке программы FB происходит подстановка вместо формальных параметров переменных из списка. Последовательность переменных в списке параметров контролируется программирующим устройством.

Команда перехода после вызова FB автоматически активируется программатором, но не индицируется на дисплее. Вызов FB, команда перехода и каждый параметр занимают в памяти отдельное слово, а вызов FX - два слова, исключение - числа с плавающей запятой (тип параметра KG): они занимают два слова.

Пример: (вызов функционального блока и передача параметров в программном блоке в видах представления программы AWL и KOP/FUP).

- Вид представления AWL

```
PB25

      : SPA FB 201
NAME  : E-ANTR
ZU-E  :     DW  1
RME   :     E  3.5
ESB   :     M  2.5
UEZ   :     T   2
ZEIT  :     KT 10.1
ZU-A  :     DW  1
BEU   :     A  2.3
LSL   :     A  6.0
```

Формальные операнды — Фактические операнды

- Вид представления KOP/FUP

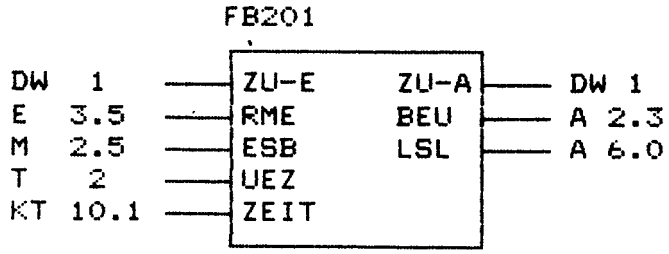


Рис. 11 Вызов функционального блока

Идентификаторы входов/выходов функционального блока, которые появляются на экране программатора в режиме программирования, и имя FB хранятся в самом FB. Поэтому перед программированием на программаторе все необходимые функциональные блоки должны быть перенесены на рабочую дискету или непосредственно в память программатора. (Для получения более детальной информации обратитесь к Руководству по программатору).

2.7.3 Программирование функциональных блоков

FB создается в два приема в соответствии с его структурой: Введение заголовка и введение "тела" блока.

Перед введением "тела" блока (на языке STEP 5) вводится заголовок. Заголовок включает:

— библиотечный номер
— номер FB
— формальный операнд (имена параметров блока)
— тип параметров блока
— тип блока данным

o Библиотечный номер

В качестве библиотечного номера можно использовать любое число от 0 до 65535. FB определяется этим номером независимо от его символьных или абсолютных параметров. Достаточно один раз присвоить библиотечный номер, чтобы полностью идентифицировать FB. Стандартные FB имеют строго определенные библиотечные номера.

o Имя Функционального блока

Имя FB может иметь до 8 символов. Имя идентифицирует функциональный блок символом.

o Формальный операнд

Формальный операнд может иметь до 4 символов и должен обязательно начинаться с буквы. Каждый FB может программировать до 40 параметров.

o Параметры блока

<u>Вид параметров блока</u>

В качестве парамеров блока E, A, D, B, T или Z могут быть запрограммированы:

— E - вход
— A - выход
— D - данные
— B - команда вызова блока
— T - таймер
— Z - счетчик

На графическом изображении FB параметры E, A, D, B, T и Z расположены слева, параметр A - справа.

Тип параметров блока

Для видов параметров E, A и D должен быть указан тип параметров

BI/BY/W/D	-	для классов E и A
KM/KH/KY/KS/KF/KT/KC/KG	-	для класса D

Для параметров E,A и D тип параметра определяет, какой размер - битовый, байтовый, словный или размер двойного слова - используется, и какой формат данных определен для параметра D (см. Руководство по программированию PG).

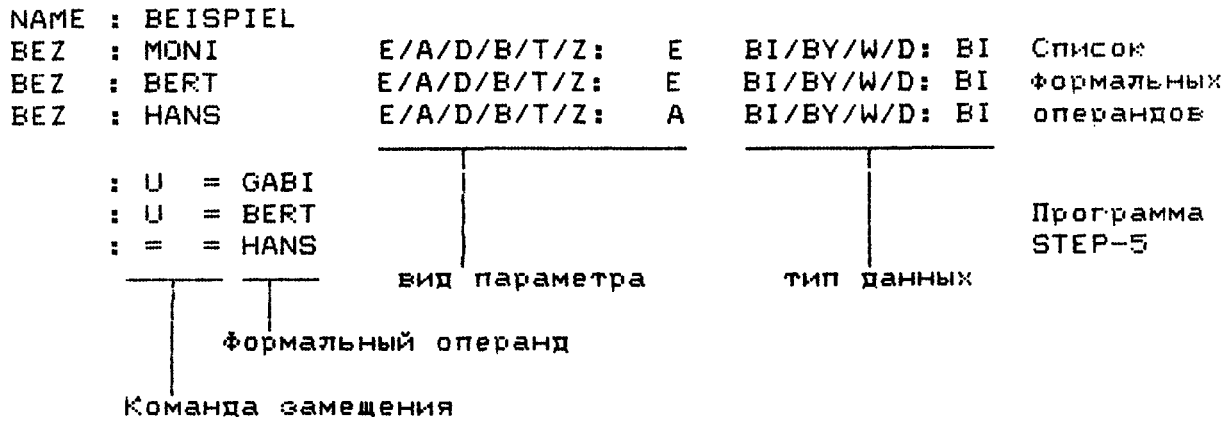
Вид параметра	Тип параметра	Допустимые актуальные операнды
E, A	BI — для операнда с адресом битов	E n.m — входы A n.m — выходы M n.m — маркеры
	BY - для операнда с адресом байтов	EB n — входные байты AB и — выходные байты MB n — байты маркеров DL n — старший байт слова данных DR n — младший байт слова данных PB n — периферийные байты
	W - для операнда с адресом слов	EM n — входные слова AW n — выходные слова MW n — слова маркеров DW n — слова данных PW n — периферийные слова
D	KM — для двоичного числа (16 разрядов) KY — для двух байтовых чисел от 0 до 255	Константы

Вид параметра 1	Тип параметра	Допустимые актуальные операнды
D	KH — для 16-ричных чисел (макс. 4 Позиции) KC — для символов (макс.2 буквенно-цифровых знака KT — для времени (BCO—код) в диапазоне 1...999.3 KZ — для счетчиков (BCD) диапазон: O...999 KF — для чисел с фиксированной запятой в диапазоне от -32768 до +32768	
B	Указание типа не требуется	DB n - блоки данных, выполняется команда ADB FB n - функц. блоки (допустимы только без параметров) вызываются безусловно SPA PB n - прогр. блоки вызываются безусловно SPA SB n – шагов блоки вызываются безусловно SPA
T	указание типа не требуется	T 0...127 Таймер [1)]
Z	указание типа не требуется	Z 0...127 Счетчик [1)]

[1)] Значение параметра таймера или счетчика должно быть указано в качестве данные или должно быть - запрограммировано в качестве константы в функциональном блоке.

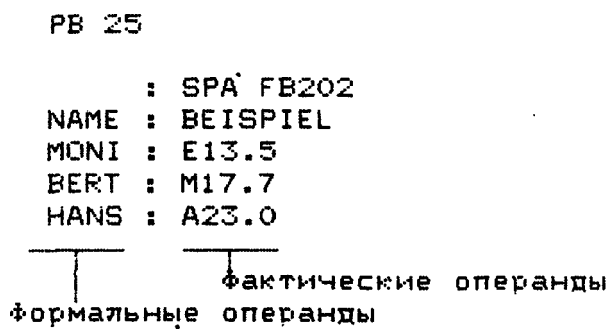
Пример (Программа в функциональном блоке)

FB 202

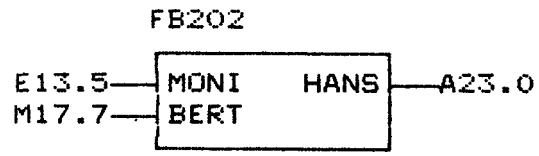


Пример (Вызов функционального блока из программного блока)

- Вид представления AWL



- Вид представления KOP/ FUP



- Выполняемая программа

```
:U  E13.5
:U  M17.7
:=  A23.0
```

Операции (команды замещения), к которым относятся параметры, должны быть запрограммированы в функциональном блоке с формальными параметрами в качестве операндов. Кроме того, к формальным операндам можно обращаться несколько раз в различных местах функционального блока.

Во время вызова функционального блока формальные операнды замещаются фактическими параметрами из списка параметров.

Внимание! Если порядок или количество формальных операндов в заголовке функционального блока изменяется, то должны быть изменены соответствующие команды и список параметров в блоке, в котором происходит вызов данного функционального блока.

Пример (стандартный функциональный блок)

RAD:	FB6 для 115U
Извлечение корня из числа с плавающей запятой	FB6 для 135U
GP	FBI 9 для 150U

Функциональный блок RAD:GP извлекает квадратный корень из числа с плавающей запятой (8-битовая экспонента и 24-битовая мантисса). Результат - также число с плавающей запятой (8-битовая экспонента 24-битовая мантисса), как бы младший бит мантиссы не округлялся.

Если необходимо, функциональный блок для последующих операций устанавливает идентификатор "Корень отрицательный".

Числовые значения:
Радикал -0.1469368 Exp -38...+0.1701412 Exp +39
Корень +0.3833234 Exp -19...+0.1304384 Exp +20

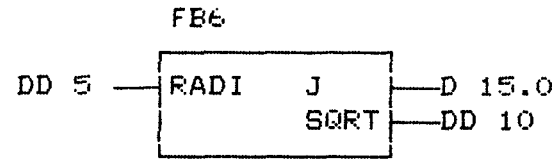
Функция : Y = ±A
Y = SQRT; A = EXTR

Вызов функционального блока :

В виде AWL:

```
      : SPA FB 6
ИМЯ   : RAD :GP
RADI: DD  5
J     : D 15.0
SQRT: DD 10
```

В виде KOP/FUP



DD - двойное слово данных

В приведенном выше примере извлекается корень из числа с плавающей запятой, которое записано в двойном слове данных DD5 в виде 8-битовой экспоненты и 24-битовой мантиссы. Результат, который тоже является 32-битовым числом с плавающей запятой, записывается в DD10. Перед выполнением функции должен быть вызван соответствующий блок данных. Параметр J (вид параметра M, тип параметра BI) определяет знак результата: J=1 для отрицательного результата. Используемые слова маркеров : с MW238 по MW254.

В каталоге ST57 приведены стандартные функциональные блоки для ПК 135U, время их выполнения, необходимая память и используемые ими переменные.

Общие замечания

Если используется стандартный функциональный блок, то байты маркеров с 200 по 255 резервируются, и пользователь не должен их использовать.

Таймер TO, счетчик ZO и блоки данных DB O. 1, 2 и DX O также зарезервированы.

Стандартные функциональные блоки занимают номера с 1 по 199. Функциональные блоки пользователя могут, поэтому, иметь номера с 200 по 255, если используются стандартные функциональные блоки.

Функциональный блок FBO обрабатывается системой циклически вместо организационного блока OB1, если OB1 не запрограммирован.

2.8 Организационные блоки

Организационные блоки осуществляют связь между системной программой и программой пользователя. Организационные блоки с OB1 по OB39 являются частью программы пользователя, также как и программные, шаговые и функциональные блоки. Организационные блоки вызываются системной программой. Пользователь имеет возможность программировать организационные блоки с OB1 по OB39 и, таким образом, косвенно иметь доступ в системную программу.

Для тестирования эти организационные блоки могут быть вызваны пользователем (по команде SPA/SPB OBxxx).

Соответствующим программированием организационных блоков могут быть установлены следующие режимы работы (см. Разделы с 3.5 по 3.7):

— Циклическое выполнение программы (OB1 или FBO);
— Обработка прерываний от процесса (OB2);
— Обработка прерываний по времени (OB13);
— Новый пуск (OB20);
— Ручной перезапуск (OB21);
— Автоматический перезапуск (OB22);
— Обработка прерываний (OB 19, 23...34).

Абсолютный параметр	Назначение или причина для инициализации
OB для циклической обработки процесса (см. Раздел 3.6.1):	
OB1	Циклическая обработка
OB для обработки прерываний процесса (см. Раздел 3.6.2):	
OB2	Обработка прерываний от процесса
OB для обработки прерываний по времени (см. Раздел 3.6.3):	
OB13	База времени 0.1сек
OB для запуска:	
OB20 OB21 OB22	Новый пуск Ручной перезапуск Автоматический перезапуск

Абсолютный параметр	Назначение или причина для инициализации	Реакция при отсутствии OB
OB для обработки прерываний (см. Раздел 3.7):		
OB19	Вызов незагруженного блока	"СТОП"
OB23	Задержка квитирования при обращении к модулям периферии или другим адресам шины S5 (QVZ)	нет
OB24	Задержка квитирования при обновлении области отображения процесса и при передаче маркеров связи (QVZ)	нет
OB25	Ошибка адресации (ADF)	"СТОП"
OB26	Превышение времени цикла (ZYK)	"СТОП"
OB27	Ошибка замещения (подстановки)	"СТОП"
OB28	"СТОП" от PG и останов по шине S5 [1])	"СТОП"
OB29	Недопустимый код операции	"СТОП"
OB30	Недопустимый параметр	"СТОП"
OB31	Общая ошибка специальных функций	"СТОП"
OB32	Ошибка передачи в блок данных	"СТОП"
OB33	Обработка прерывания по времени вызвана в тот момент, когда еще активна обработка предыдущего прерывания	"СТОП"
OB34	Ошибка при обработке регулятора	"СТОП"

[1]) OB28 вызывается перед переходом в состояние "СТОП". Останов происходит всегда независимо от того, есть эти OB28 и что в нем запрограммировано.

Абсолютный параметр	Назначение или причина для инициализации
OB для специальных функций:	
OB40...255	Смотри список ниже

Помимо организационных блоков с OB1 по OB39 специальные функции системной программы могут быть вызваны с помощью организационных блоков с номерами больше 39. Эти блоки для специальных функций не могут быть запрограммированы пользователем — они могут быть только вызваны. Они не содержат команд программы STEP—5. Специальные функции отдельно описаны в Разделе 5.

Организационные блоки специальных функций могут вызываться из организационных блоков OB1...0B39 (при использовании версий программного обеспечения программатора PG675, начиная с SO A03 или S1 A01).

Список встроенных на настоящий момент в R-процессор специальных функций.

OB 216-218	Доступ к буферу связи
OB220	Расширение числа с фиксированной запятой в Акк 1 от 16 до 32 бит путем расширения знака
OB221	Установка нового времени цикла
OB222	Триггер времени цикла
OB223	Останов при неодновременном запуске в многопроцессорном режиме
OB224	Передача меток связи по блокам в многопроцессорном режиме работы
OB226	Чтение по байтам содержания ячеек памяти операционной системы
OB227	Чтение контрольной суммы памяти операционной системы
OB230-237	Вспомогательные функции операционной системы
OB240	Инициализация регистра сдвига
OB241	Вызов регистра сдвига
OB242	Стирание регистра сдвига
OB250	Инициализация PID—регулятора
OB251	Вызов PID—регулятора
OB254	Копирование блока данных DX
OB255	Копирование блока данных DB

3 Работа

3.1 Обзор рабочих состояний

И при однопроцессорной работе контроллера, и при многопроцессорной различают следующие рабочие состояния:

— Состояние "СТОП"
— Запуск
— Выполнение программы

Каждое рабочее состояние делится на три типа (см. Рис 11).

"СТОП"	Запуск	Выполнение программы
Светодиод "STOP" быстро мерцает= Предупреждение	Новый старт	Циклическое
Светодиод "STOP" постоянно горит	Ручной перезапуск	Обработка прерывания от процесса
Светодиод "STOP" медленно мерцает = Ошибка	Автоматический перезапуск	Обработка прерывания по времени

Рис.12 Рабочие состояния

Рабочие состояния отражаются на светодиодах передней панели CPU:

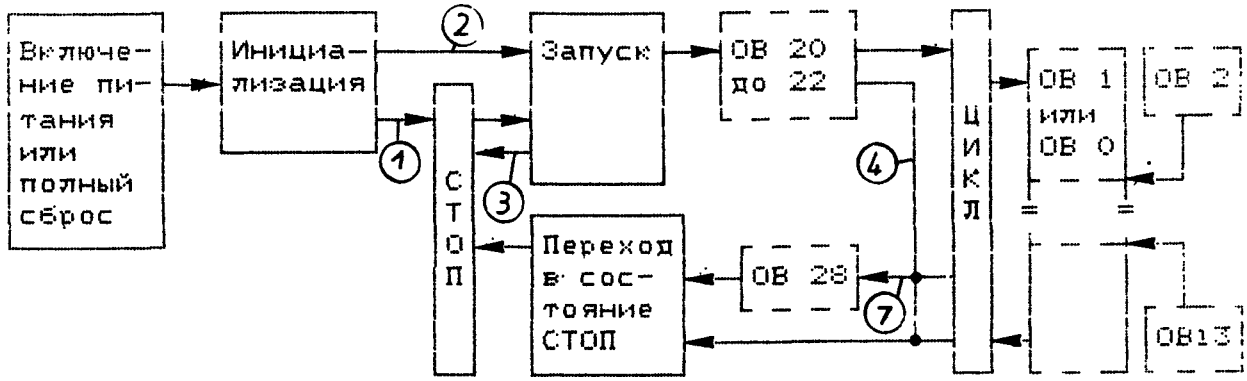
- В состоянии "СТОП" горит красный светодиод "STOP", причем либо постоянно, либо мерцая быстро или медленно в зависимости от причины останова.

- Во время циклической работы программы постоянно горит зеленый светодиод "RUN".

- Цифровые выводы перед началом циклической обработки программы деблокируются. В состоянии запуска или останова от CPU выдается сигнал BASP. горит светодиод BASP, и все выходы блокированы (исключения см. в Разделе 3.4)

Рисунок 13 показывает структуру расположения системных программ. Организационные блоки осуществляют интерфейс между системными программами и программой пользователя. Они вызываются операционной системой в различных рабочих состояниях и служат для программирования реакции пользователя. Операционная система выполняет свои функции независимо от того, запрограммированы организационные блоки или нет. Если. например, отсутствует ОВ запуска, то CPU перейдет в циклическую работу.

Рабата CPU при одно- и многопроцессорном режимах с модулями памяти RAM и EPROM описана в Руководстве по эксплуатации центрального устройства.



——————— Системная программа
——————— Программа пользователя

a Включение питания или полный сброс
b Инициализация
c СТОП
d Запуск'
e Переход в состояние "СТОП"
f ОВ 20 - 22
g Цикл
h ОВ1 или FBO

1 Ошибки, которые были обнаружены во время инициализации (см. Раздел 3.2) или переключатель режима работы в положении "STOP", или перед запуском было состояние "СТОП", или автоматический перезапуск.
2 Автоматический перезапуск после подачи напряжения (см. Раздел 3.5.3) с продолжением программы пользователя с места прерывания.
3 Ошибка во время запуска перед тем, как был вызван 0В запуска, например, ошибка DB1 (см. Раздел 3.7).
4 Ошибка или неисправность во время выполнения программы пользователя в 0В запуска (Раздел 3.7).
5 Выполнение программы обработки прерывания процесса (Раздел 3.6.2)
6 Выполнение программы обработки прерывания по времени ('Раздел 3.6.3)
7 Вызов соответствующего организационного блока, если возникла определенная ошибка (Раздел 3.7).
8 Продолжение выполнения программы пользователя с места прерывания.
9 Причина ошибки - в 0В ошибки.

Рис.13 Структура операционной системы

3.2 Инициализация

После включения напряжения литания и при выполнении полного стирания (см. Раздел 3.3.1). CPU независимо от предыдущего рабочего состояния выполняет программу инициализации, которая включает:

После "Напряжение включено";

- Проверка содержимого памяти пользователя и CPU-RAM.
- Стирание списка адресов блоков (DBO) и конфигурирование заново памяти пользователя и DB-RAM

После полного стирания:

- Стирается вся память RAM
- Дальше как при "Напряжение включено"

Если при инициализации возникла ошибка, CPU переходит в состояние "СТОП" с быстрым мерцанием светодиода "STOP":

а) Содержимое RAM неисправно, например из-за прерывания буферизации при включении
б) Не установлен EPROM пользователя, или установленный EPROM пуст.

Быстрое мерцание означает требование полного стирания системы. Причина ошибки должна быть устранена, и после этого очищен CPU (см. Раздел 3.3.1), то же самое необходимо проделать и при первоначальной установке CPU (аналогично а).

Если перед "Напряжение выключено" CPU был в циклическом режиме, то после "Напряжение включено" операционная система при неизменном положении переключателей на CPU и KOR выполняет автоматический перезапуск (см. Раздел 3.5.3).

3.3 Состояние останова

3.3.1 Быстрое мерцание светодиода STOP (предупреждение)

Значение; Требование полного стирания от системы или от пользователя

o Требование полного стирания от системы

Если получено требование полного стирания от системы после возникновения ошибки при инициализации, то полное стирание CPU производится следующим образом:

- Переключатель выбора устанавливается в положение "OVERALL RESET"; одновременно переключатель режима работы переводится из "STOP" в "RUN" и снова в "STOP", или
- функцией PG "Полное стирание AG" (см. Руководство пользователя PG).

Результат: Выполняется полное стирание (см. Раздел 3.2). Светодиод "STOP" горит постоянно.

o Требование полного стирания от пользователя

Пользователь, производит полное стирание следующим образом;

- Перевод контроллера в состояние "СТОП", путем перевода переключателя режимов работы из "RUN" в "STOP".
- Переключатель режимов устанавливается в положение "OVERALL RESET"; одновременно переключатель режимов работы переводится из STOP в "RUN" и снова в "STOP".

Результат: Светодиод "STOP" быстро мерцает (предупреждение: "Требуется полное стирание").

Затем выполняется полное стирание [1]):

- Переключатель выбора устанавливается в положение "OVERALL RESET" ; одновременно переключатель режимов работы переводится из "STOP" в "RUN" и снова в "STOP", или
- функцией PG "Полное стирание AO" (см. Руководство пользователя PG).

Результат: Выполняется полное стирание (см. Раздел 3.2). Светодиод "STOP" горит постоянно.

3.3.2 Светодиод "STOP" горит постоянно

CPU переходит в состояние останова

- при переводе переключателя режимов работы из "RUN" в "STOP"" (сюда не относятся такие предыдущие ошибки, как выбор недопустимого режима запуска или ошибка DB1; CPU при этом еще не в цикле);
- при аппаратных ошибках, которые не являются частными ошибками отдельных CPU (BAU, PEU, NAU, см. Раздел 3.7);
- при функции PG "СТОП AO" на CPU в многопроцессорном режиме;
- после полного стирания;
- в многопроцессорном режиме при функции PG "СТОП AG", BASP, BARBEND на других CPU;
- в многопроцессорном режиме по сигналу останова от координатора (при переводе переключателя режимов работ координатора в положение "STOP" или при останове других CPU по причинам, не являвшимся ошибками CPU, исключение - см. Раздел 3.4).

3.3.3 Медленное мерцание светодиода "STOP"

CPU в одно- и многопроцессорном режимах переходит в состояние останова при следующих ошибках CPU:

- при программных и аппаратных ошибках (ADF, ZYK, SUF, QVZ, TRAF,.., см. Раздел 3.7);
- при обработке программы (ошибка DB-1, выбор недопустимого вида запуска);
- при наличии команды останова в программе пользователя;
- при функции PG "СТОП AG" (только в однопроцессорном режиме) и при функции FG BARBEND в CPU;

[1]) Если затребованное пользователем полное стирание не должно выполняться, то необходимо после требования полного стирания выбрать режим запуска (см. Раздел 3.5).

o Реакция на состояние останова

- Выдача сигнала BASP, который блокирует все цифровые выходы. Это индицируется светодиодом "BASP" (исключение - см. Раздел 3.4);
- Вызов организационного блока OB2S и создание стека прерываний при определенных ошибках (см Раздел 3.7);
- При многопроцессорной работе, если один из CPU переходит в состояние "СТОП", то остальные переходят в "СТОП" автоматически (исключение: смотри Раздел 3.4).

o Пути выхода из состояния останова

- Полный сброс, затем новый пуск
- Тестовая операция
- Выбор режима запуска

3.4 Режим тестирования

С помощью режима тестирования можно запустить отдельный процессор в многопроцессорной системе (или любую требуемую комбинацию CPU) без CPU, находящимся в состоянии "СТОП", блокирующих весь контроллер в целом. Здесь следует отметить следующие моменты:

- Запуск отдельного процессора не синхронизирован. В зависимости от длины вызываемых при запуске организационных блоков (OB20, OB21, OB22) цикл запуска CPU может иметь различное время.

Невозможен останов нескольких CPU в случае ошибки. При ошибках в состояние останова переходит только неисправный CPU. Исключение: Во время полного стирания, при функциях PG "СТОП АО", BARB, BARBEND и при ошибке DB1 на одном из CPU в состояние останова переходит весь контроллер.

Сигнал BASP не выдается. В случае ошибки цифровые выходы не блокируются (исключения смотри выше).

Тестовый режим можно отключить установкой на KOR, чтобы было невозможно вызвать эту функцию с помощью переключателя, если при ней может возникнуть опасная ситуация. Функция "Тестовый режим" в состоянии поставки по этим соображениям заблокирована.

o Инициализация тестовых функций

Тестовая функция должна быть деблокирована на KOR (координаторе) (смотри Руководство по координатору).

- Переключатель на координаторе должен быть переключен из положения "STOP" в положение "TEST", после этого светодиод "BASP" должен погаснуть.

- Режим запуска должен быть установлен на CPU, которые должны быть переведены в режим выполнения циклической программы (см. Раздел 3.5).

3.5 Запуск

Системная программа CPU имеет три различных режима запуска контроллера:

- Новый пуск
- Ручной перезапуск
- Автоматический перезапуск

Допустимость режима запуска отображается на программаторе при анализе ошибок в состоянии "СТОП" (см. Раздел 3.7, биты управления NEU-ZUL, MWA-ZUL).

Для каждого режима запуска системная программа вызывает организационный блок, который пользователь может запрограммировать для конкретизации событий во время запуска. Если это не требуется, то данные организационные блоки можно не программировать.

3.5.1 Новый пуск

Для выполнения нового пуска необходимо:
— удерживая переключатель в положении "RESET", перевести переключатель режима работы из положения "STOP" в "RUN";
— после этого при многопроцессорном режиме работы выполняется запуск отдельных процессоров: Запуск KOR (исключение см. Раздел 3.4) или используется функция программатора "СТАРТ АО". Подразумевается, что переключатели режимов работы на всем процессорах и на координаторе находятся *в* положении "RUN".

Новый старт будет выполнен только в том случае, если в каждом CPU присутствует блок DB1.

Тогда системная программа:

— Сбрасывает все переменные (маркеры, маркеры связи, таймеры и счетчики, PAA); цифровая и аналоговая периферия стирается.
— Создает список адресов (DB1) для входов, выходов или маркеров связи и сравнивает с периферией или маркерами связи, которые подтверждаются соответствующими модулями.
— Вызывает организационный блок OB20.

Данные инициализации сохраняются (особенно блок списка адресов DBO). Если в многопроцессорном режиме не запрограммирован DB1 или указанные в DB1 цифровые входы, выходы или маркеры связи не подтверждаются соответствующими модулями, то CPU переходит в состояние останова с сообщением об ошибке (ошибка DB1, см. Раздел 3.7). После устранения ошибки необходим новый пуск.

В организационном блоке OB20 пользователь может записать программу, которая выполняет отдельные операции инициализации один раз перед запуском циклической программы, например установка маркеров, запуск таймеров, установка выходов и, если необходимо, подготовка данных для обмена данными между контроллером и устройствами ввода/вывода. OB20 должен быть завершен командой BE (конец блока). После обработки OB20 начинается циклическое выполнение программы в OB1 или FBO.

3.5.2 Ручной перезапуск

Для выполнения ручного перезапуска необходимо:

— поставить переключатель в среднее положение;
— перевести переключатель режима работы из положения "STOP" в положение "RUN";
— после .этого при многопроцессорном режиме работы выполняется новый пуск отдельных процессоров: запуск KOR (исключение см. Раздел 3.4) или используется функция программатора "СТАРТ AG"; подразумевается, что переключатели режимов работы на всех процессорах и на координаторе находятся в положении "RUN".

Во время выполнения ручного перезапуска результат достигается до того, как контроллер переходит в состояние останова, и состояние предыдущей операции запишется в счетчик, т.е. маркеры и маркеры связи не стираются. Текущие значение таймеров и счетчиков на момент прерывания циклической обработки программы сохраняются, так же как данные нового пуска (DB1) и инициализации (DBO). В качестве интерфейса пользователя вызывается OB21, Функции которого соответствуют OB20 при новом пуске.

По завершении обработки программы по прерыванию (не программы запуска) со следующей команды после прерывания (при BASP) продолжается циклическая обработка программы. PAA и цифровые выходы, соответствующие CPU, в конце прерванного цикла стираются, для того чтобы избежать влияния выходов при прерывании. После этого начинается новый цикл с вызова OB1 или FBO после сброса сигнала BASP, который до этого блокировал все цифровые выходы.

3.5.3 Автоматический перезапуск

Для выполнения автоматического перезапуска необходимо:

— во время выполнения циклической программы выключить и затем вновь включить питание, если
— положение остальных переключателей оставалось в прежнем состоянии "RUN";
— не было ошибок инициализации, и программа пользователя не менялась

Если был сбой напряжения во время выполнения программы, ПК автоматически выполняет новый пуск при появлении питания. В этом случае системная программа сначала вызывает организационный блок OB22, в котором пользователь может запрограммировать предустановки отдельных состояний. В противном случае процедура выполнения автоматического перезапуска идентична ручному перезапуску. Если контроллер не должен выполнять автоматический перезапуск, то в OB22 должна быть запрограммирована команда STP (СТОП).

```
OB22      : STP  (STOP)
          : BE   (конец блока)
```

3.5.4 Запуск в многопроцессорном режиме

В многопроцессорном режиме необходимо после запуска отдельных CPU запустить KOR (исключение см. Раздел 3.4):

— Переключатель выбора на KOR необходимо перевести из "STOP" в "RUN" или из "STOP" в "TEST";

- или с помощью функции PG "ПУСК AG" на CPU, вызвавшем состояние останова, если переключатели на всех CPU и на KOR неизменно стояли в положении "RUN".

Пои запуске ПК в многопроцессорном режиме путем запуска KOR необходимо:

- установить переключатели режимов работ всех CPU в положение "RUN"

- установить вид запуска каждого отдельного CPU по их предыдущим условиям в состоянии останова. При этом возможно, что некоторые CPU перейдут в режим перезапуска, а другие - в режим нового пуска. Посредством маркеров связи можно "неверную" установочную информацию передать от одного CPU к другому, если ПК перед этим был в цикле и этому не препятствуют соответствующие программы в организационных блоках запуска OB20. . . OB22. Маркеры при разных видах запуска обрабатываются по-разному.

В многопроцессорном режиме можно запустить ПК только единственным образом, используя запуск KOR, если причиной перехода процессоров с состояние "СТОП" явился только KOR. В этом случае все процессоры выполняют ручной перезапуск. Этот вид запуска действителен также для тех CPU, которые не были причиной останова ПК и были переведены в состояние останова сигналом от KOR (не предполагая при снятии сигнала останова, что они не должны запускаться своими переключателями режимов работы).

При автоматическом перезапуске KOR автоматически запускается. Вид пуска для всех CPU в ПК одинаков.

Запуск отдельным процессоров в многопроцессорном режиме (но не в тестовом) синхронизируется по времени, т.е. процессоры остаются в цикле ожидания до тех пор, пока все процессоры не закончат процедуру запуска и затем все вместе переходят к циклической работе.
При запуске от PG переключатели режимов работы всех процессоров должны быть в положении "RUN". Тот CPU, который работает с PG, выполняет выбранный режим запуска, а остальные - ручной перезапуск, если они не управляются переключателями режимов работы (как при снятии сигнала останова от KOR).

3.6 Обработка программы

3.6.1 Циклическая работа

Циклическая обработка программы — нормальная работа программируемого контроллера (Рис.14). Процессор начинает обработку программы с начала программы STEP-5, по очереди выполняются команды языка STEP—5 до конца программы и затем снова с начала.

Организационный блок OB1 или функциональный блок FBO является интерфейсом между системной программой и циклически выполняемой программой пользователя. Первая команда STEP—5 в OB1 является первой командой программы пользователя (если нет OB20...22). Если запрограммированы и OB1 и FBO, то системной программой обрабатывается только OB1.

В OB1 или FBO вызываются программные, функциональные и шаговые блоки. Дальше из этих блоков могут быть вызваны другие программные, функциональные и шаговые блоки, и т.д. Максимальная глубина вложений - 24 уровня. Это значение считается как сумма уровней глубины вложения всех возможных режимов работы (циклический, прерывание от процесса, прерывание по времени) и если выполняется аппаратное прерывание.

o Время цикла

Время выполнения программы пользователя — сумма времен выполнения блоков, которые вызываются. Если блок вызывается n раз. время его выполнения должно складываться n раз. Сумма времен всех обрабатываемых; частей программы пользователя (циклической + прерывание от процесса + прерывание по времени + возможно, аппаратное прерывание) за исключением программы запуска (OB 20...0B 22) составляет время цикла программы пользователя при однократном прогоне программы - от вызова 0B1 или FB0 до конца обработки (BE).

Общее время цикла является суммой времени цикла программы пользователя и времени, затраченного на циклическую часть системной программы (см. Рис.14). Это общее время цикла контролируется операционной системой. Оно стандартно установлено на максимально допустимое значение 150 мс. Пользователь может установить контрольное время цикла в OB 221 (см. Раздел 5.3). Если оно завершится прежде, чем будет установлено системной программой (Рис.14), то CPU перейдет в состояние останова с сообщением об ошибке ZYK (см. Раздел 3.7).

Вызвав специальную функцию операционной системы "Установка времени цикла" (0B 222, см. Раздел 5.3), можно заново запустить время цикла из программы пользователя, т.е. при каждом вызове OB 222 внутренние часы со стандартным или заданным в OB 221 контрольным временем запускаются заново.

o Область отображения процесса

Модули периферии и CPU обмениваются данными о состоянии цифровых входов и выходов (P-периферия с адресами байтов O...127) только один раз за цикл обработки программы пользователя. Буферная память образуется в области отображения процессов входов и выходов в области системных данных CPU. При этом, по сравнению с прямым доступом к модулям периферии время обработки соответствующей команды (см. Раздел 4.2) увеличивается и исключается "дребезг" выводов от часты;; переключении внутри цикла обработки.

o Маркеры связи

Маркеры связи служат для обмена данными между отдельными CPU или между CPU и CP (см. Раздел 2.3.1). Они, как и область отображения процесса, циклически считываются в CPU (вводные маркеры связи) или выдаются (выходные маркеры связи).

o Индикация

Во время безошибочной циклической работы процессора горит зеленый светодиод "RUN". Цифровые выходы деблокируются в начале цикла; красный светодиод "BASP" гаснет. Красный светодиод "STOP" и светодиоды ошибок не горят.

o Места прерываний

Выполнение циклической программы может быть прервано:
- — выполнением программы обработки прерываний от процесса (Раздел 3.6.2)
- — выполнением программы обработки прерываний по времени (Раздел 3.6.3)
- — обработкой регулятора (см. Руководство пользователя по компактному регулятору в AG 135U/K-процессор);
  аппаратным прерыванием (см. Раздел 3.7).

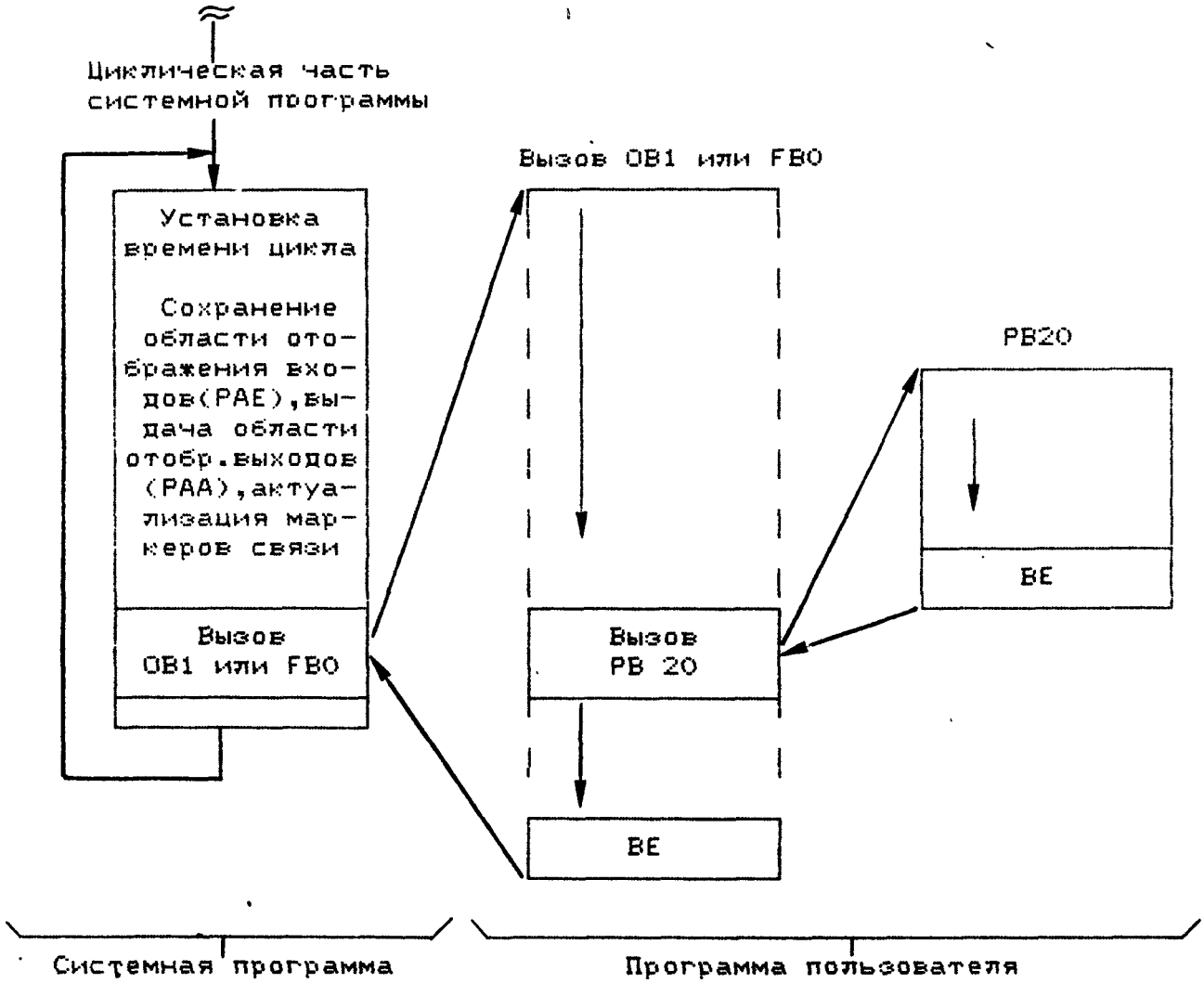


Рис.14 Выполнение циклической программы

o Общая структура программы пользователя

OB1 или FBO содержит общую структуру программы пользователя. Документирование этого блока предназначается для показа основной структуры программы (Рис.15) или для придания выразительности частям системы, которые соединяются в группы (Рис. 16).

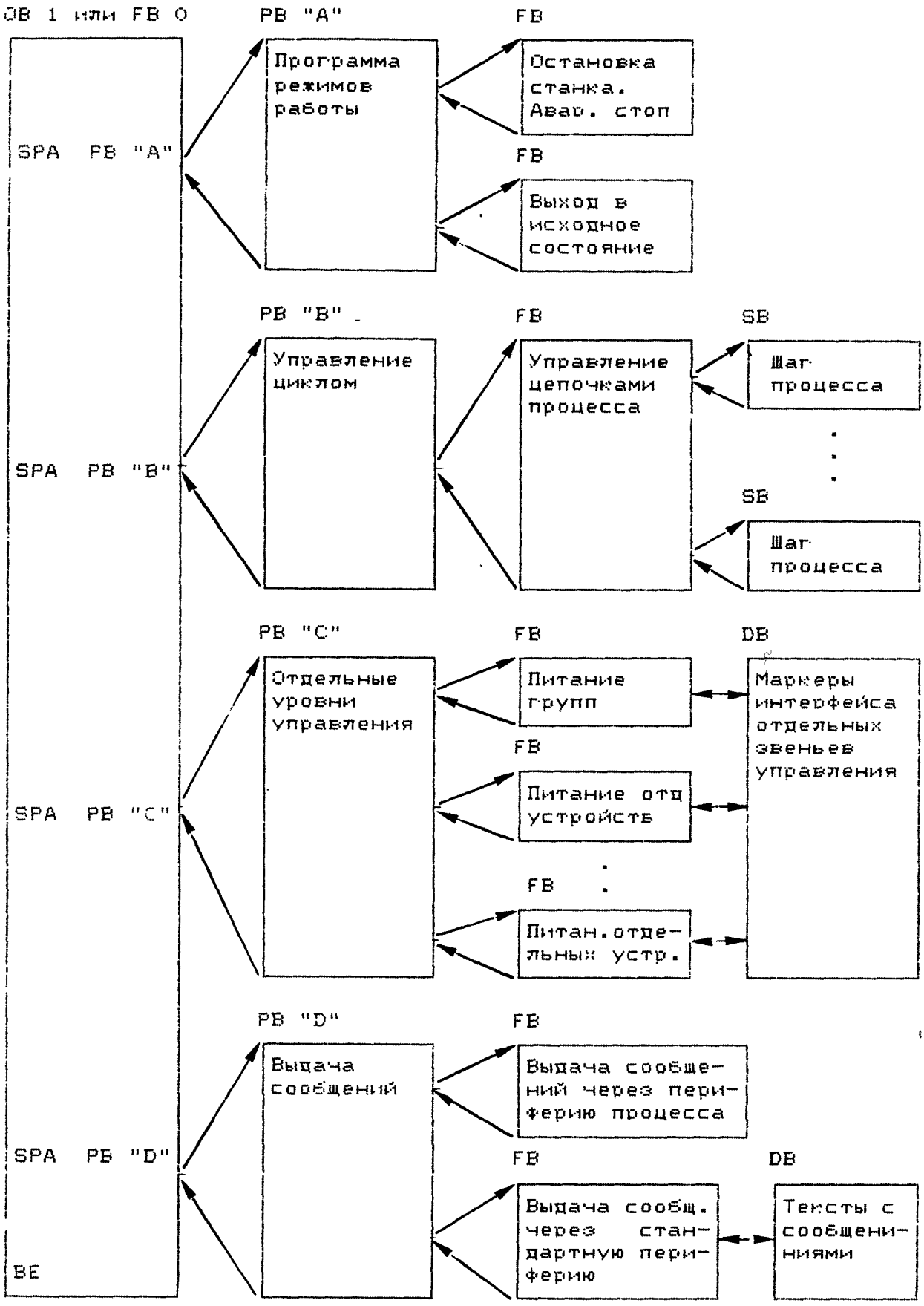


Рис. 15. Базовая структура программы пользователя по отношению к программной структуре

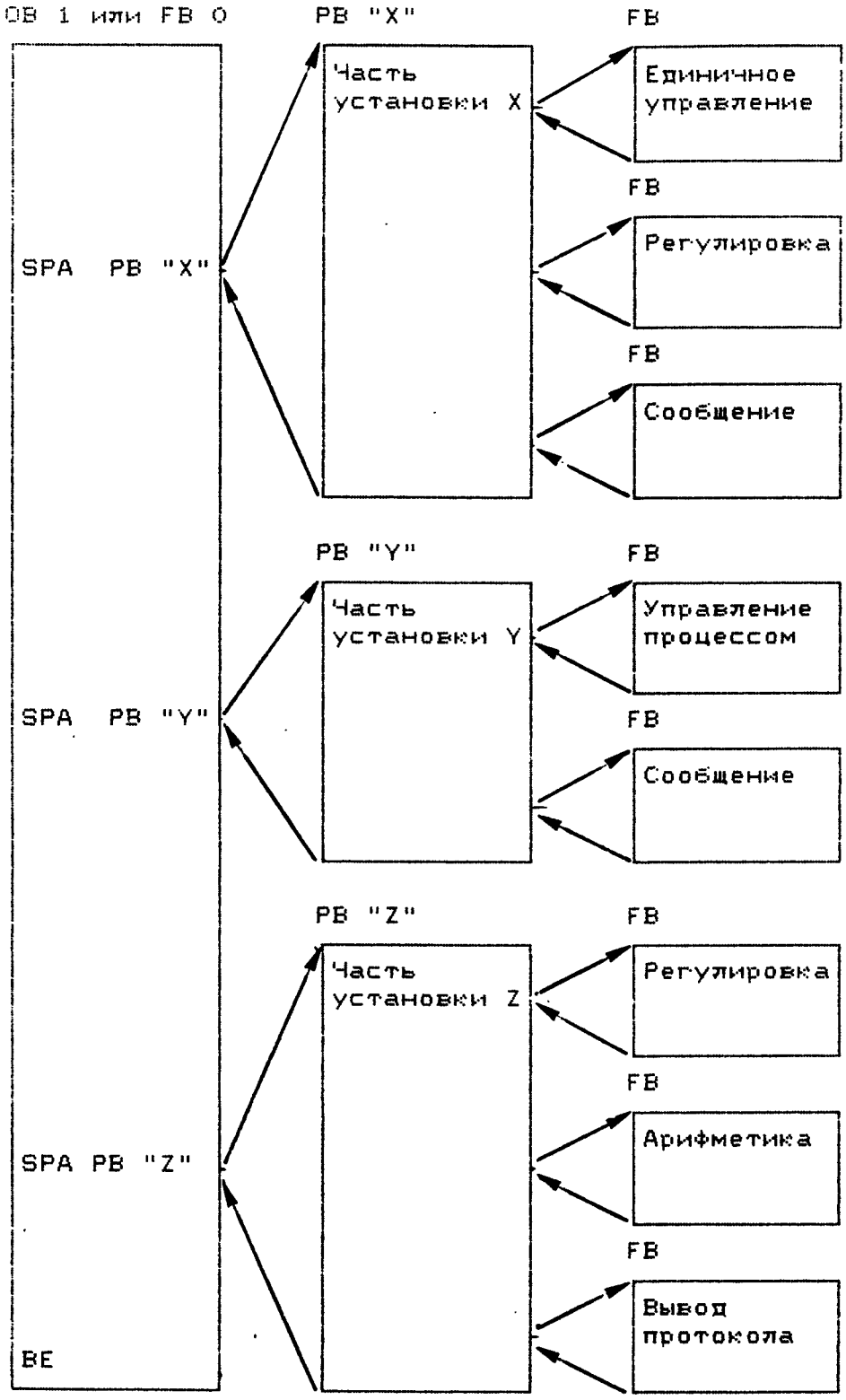


Рис.16 Базовая структура программы пользователя по отношению к системной структуре

Указание

Регистры вычислений .АКК 1,2,3 и 4 нельзя использовать для хранения данных за границей цикла - с конца цикла программы до начала следующего, - т.к. они используются там системной программой.

3.6.2 Обработка программы прерывания от процесса

Контроллер 135U может обрабатывать прерывания, поступающие в контроллер от процесса. В этом режиме работы выполнение циклической программы прерывается на границе блока сигналом прерывания на шине S5 (смотри Руководство по S-процессору и центральному устройству). После этого системная программа вызывает организационный блок OB2, в котором пользователь может запрограммировать требуемую реакцию на возникший сигнал прерывания. После того, как эта программа будет, выполнена, процессор возвращается в точку прерывания и продолжает выполнять циклическую программу, начиная с этой точки.

Следующие запросы прерывания принимаются после обработки OB2. Циклическая обработка программы при следующей смене блока пути команды STEP-5 (зависит от предустановки) вновь прерывается обработкой программы прерывания от процесса.

Обработка прерываний от процесса означает, что пользователь может непосредственно реагировать на сигналы от процесса.

o Точки прерываний

Прерывание циклической программы не может быть выполнено в произвольной точке по запросу прерывания от процесса. Обычно это возможно только на границе блоков. Когда происходит переход от одного блока к другому (командой вызова блока или при возврате в предыдущий блок по команде конца блока), системная программа может вызвать организационный блок для обработки реакции на сигнал прерывания от процесса.

Пользователь может, выбрав другую предустановку в DX O (см. Раздел 2.6), установить прерывание от процесса на границе команды STEP-5. Блок регулятора, который написан на языке Ассемблера, содержит границы псевдокоманд.

Обработка прерывания от процесса может быть прервана только неисправностью устройства (см. Раздел 3.7), и не может быть прервана запросом на прерывание по времени или вновь поступившим сигналом запроса на прерывание от процесса.

o Блокировка прерываний

Программа обработки прерываний вставляется в циклическую программу на границе блоков или границе команд STEP-5. В этой точке выполнение циклической программы прерывается. Это прерывание может иметь отрицательный эффект, если выполнение циклической программы критично по времени, например, когда должно быть очень быстрое время реакции.

Если фрагмент программы не должен прерываться, возможны следующие способы достижения этого:

— Программа не должна делиться на блоки при стандартной установке "Прерывание на границе блока". Поэтому нет переводов и нет возможности прерываний.

Сама программа представляет собой программу обработки прерываний. В этом случае тоже не может возникнуть прерывания на границе блоков или команд.
Запрет прерывания командой AS. Команда AF снова деблокирует обработку прерываний. Сегмент программы между командами AS и AF не может быть прерван сигналами запроса на прерывание.

3.6.3 Обработка программы прерывания по времени

ПК 1351-1 может выполнять программы прерывания по времени. Прерывание по времени выполняется, если приходит сигнал от "внутренних часов" к процессору, нормальная циклическая обработка прерывается и выполняется специфическая программа.

После того, как программа прерывания выполнена, процессор возвращается в ту же точку основной программы, в которой возникло прерывание и продолжает выполнять циклическую программу.

- Интерфейс между системной программой и программой обработки прерываний по времени

OB13 является интерфейсом между системной программой и программой обработки прерываний по времени. Он может вызываться системной программой каждые 1000 мс.

- Точки прерываний

Прерывание циклической программы может быть выполнено либо по умолчанию на границе блока, либо на границе команды STEP-5 при выборе соответствующей предустановки в DX O (см. Раздел 2.6). Программа обработки прерываний по времени может быть прервана, в свою очередь, сигналом запроса на прерывание от процесса, обработкой регулятора (см. Руководство пользователя по компактному регулятору в AG 135 U (R процессор)) или при возникновении аппаратной неисправности. Но вновь поступившим сигналом прерывания по времени не прерывается.

Если после 100 мс приходит новый запрос на прерывание по времени до завершения выполнения программы обработки первого прерывания, операционная система опознает "критическое" состояние и вызывает 0B 33. В 0B 33 пользователь может запрограммировать желаемую реакцию на данное состояние. После обработки 0B 33 программа продолжает выполняться с точки прерывания. Если 0B 33 не запрограммирован, процессор переходит в состояние "СТОП", и выдается сообщение об ошибке WECKFE (см. Раздел 3.7). Если запросы на обработку прерываний от процесса и по времени возникают одновременно, то программа обработки прерывания от процесса обрабатывается в первую очередь. Обработка прерывания по времени в этом случае имеет меньший приоритет.

o Время реакции

Время реакции на запрос прерывания по времени соответствует времени обработки блока или команды STEP-5 в зависимости от выбранной предустановки. Если в это время выполняется программа прерывания от процесса, то программа прерывания по времени будет выполняться только после завершения обслуживания всех запросов на прерывание от процесса. Максимальное время реакции на сигнал прерывания по времени в этом случае увеличивается на время выполнения программы прерывания от процесса. Время обработки прерывания по времени, включая возможно возникающие прерывания от процесса, не может превышать 100 мс (см. выше).

3.7 Обработка аппаратных прерываний

Операционная система может определять безошибочную работу CPU, ошибку в операционной системе или влияние неверно написанной программы пользователя.

3.7.1 Прерывания на границе команд STEP-5

Обработка программы может быть прервана на границе команды STEP-5 или даже не принята. CPU идет в состояние останова по следующим причинам:

a) "СТОП" от PG, переключателя режимов работ на CPU в "STOP" или сигнал останова от KOR (переключатель на КОР: в "STOP" или "СТОП" других CPU);
б) Команда STOP в программе пользователя
c) Сигнал от шины S5:

BAU - сбой буферной батареи в центральном устройстве
NAU - сбой напряжения питания в центральном устройстве
PEU - сбой напряжения питания в устройстве расширения;

d) Переполнение стека прерываний (STUEU) или
переполнение стека блоков (STUEB) при слишком большой глубине вложений;

e) Отсутствие списка адресов DB 1 (только в многопроцессорном режиме),
неверный список адресов (ошибка DB 1, см. Раздел 2.6) при новом пуске,
неверный блок параметрирования регулятора DB 2 (ошибка DB 2) при перезапуске
или неверный DX O.

f) Ошибка при инициализации (см. Раздел 3.2)

При вызове операционной системой в качестве интерфейса пользователя 0B 28, пользователь может запрограммировать в нем определенную реакцию, например, сохранение переменных. После обработки OB 28 CPU всегда переходит в состояние останова. Переход в состояние останова происходит независимо от того, был ли запрограммирован 0B 28 или нет. Причина неисправности в 0B 28, как в циклической программе, может быть определена и обработана.

Причины неисправности различаются следующим образом:

- a и c - постоянно горит светодиод "STOP";

- b, d, e - светодиоды "STOP" медленно мерцает;

- f - светодиод "STOP" быстро мерцает.

Причины неисправности могут быть проанализированы с помощь PG (см. Разделы 3.7.2 и 3.7.3).

Если ошибка возникла после вызова программы пользователя (только для f и d), то стек прерываний заполняется текущим состоянием индикационных байтов и аккумуляторов. Стек прерываний можно посчитать с помощью PG (см. Пример). Дополнительно к функциям в рамках вывода стека прерываний в случаях e и f причину ошибки можно уточнить в системных байтах 3 и 4 (см. Раздел 7.1).

3.7.2 Прерывания в других случаях

Другие причины неисправности препятствуют работе CPU не в такой мере. Операционная система в этих случаях дает возможность пользователю самому определять дальнейшие действия. Приведенные ниже организационные блоки вызываются в следующих случаях:

Причины неисправности	Вызов блока	Реакция при отсутствии ОВ
Вызов незагруженного блока	ОВ 19	"СТОП"
Задержка квитирования при обращении к модулям периферии	ОВ 23	нет
Задержка квитирования при обновлении области отображения процесса	ОВ 24	нет
Ошибка адресации	ОВ 25	"СТОП"
Превышение времени цикла	ОВ 26	"СТОП"
Ошибка замещения	ОВ 27	"СТОП"
Недопустимый код операции	ОВ 29	"СТОП"
Недопустимый параметр	ОВ30	"СТОП"
Общая ошибка специальных функций	ОВ 31	"СТОП"
Ошибка передачи в блок данных	ОВ 32	"СТОП"
Ошибка при обработке прерывания по времени = Ошибка будильника	ОВ 33	"СТОП"
Ошибка при обработке регулятора	ОВ 34	"СТОП"

Вызвав организационный блок ошибки, пользователь получает через аккумуляторы 1 и 2 дополнительную информацию об ошибке (см. Раздел 7.2).

Пользователь может при возникновении такой ошибки предложить CPU работать дальше, перевести его в "СТОП", запрограммировав команду останова в соответствующем организационном блоке, или заставить выполнять специальную программу. Если вызываемый организационный блок не запрограммирован, то обработка программы продолжается (при ошибках ОБ 23, 24) или CPU переходит в состояние останова (при отсутствии других организационных блоков ошибки).

o Вызов незагруженного блока

Если в программе пользователя вызван блок, который недействителен или отсутствует, то операционная система распознает такую ситуацию. Это имеет место и для условного и для безусловного вызовов и для всех видов блоков.

Если опознан вызов незагруженного блока, то операционная система вызывает ОВ 19. В ОВ 19 могут быть определены дальнейшие действия CPU. Если ОВ 19 содержит только команду BE (конец блока.), то вызов незагруженного блока обрабатывается как нулевая операция (NOP) . Обработка прерываний программы STEP-5 продолжается со следующей команды. Если ОВ 19 не запрограммирован, то при вызове незагруженного блока CPU переходит в "СТОП".

o Задержка квитирования (QVZ)

Задержка квитирования появляется, если какой-либо модуль после обращения к нему в течение определенного времени не выдает сигнал RDY (готовность). Причиной задержки квитирования может быть неисправность модуля или снятие модуля во время работы.

Следующие ошибки задержки квитирования прерывают обработку программы и вызывают соответствующие организационные блоки;

- Задержка квитирования при обращении по шине S5 к CP, IP, KOR или к модулю периферии (например, командами загрузки и передачи (L/T P... или Q...):
  Операционная система вызывает организационный блок 0B 29.

- Задержка квитирования при обновлении области отображения процесса для вводов и выводов и передаче маркеров связи:
  Операционная система вызывает организационный блок ОВ 24.

Если вызванный организационный блок не запрограммирован, то продолжится обработка программы пользователя. Задержка квитирования удлиняет время выполнения команды STEP-5, которая явилась причиной сбоя. Если задержка квитирования должна ввести CPU в "СТОП", то надо запрограммировать в ОВ 23 или ОВ 24 команду останова STP.

o Ошибка адресации (ADF)

Ошибка адресации возникает, если команда STEP—5 (см Раздел 4.2) опрашивает вход или выход в области отображения процесса, а соответствующий этому входу или выходу модуль периферии при последнем новом пуске не подтвердил своего наличия (модуль не был установлен, неисправен или не указан в блоке данных DB1 CPU).

Операционная система прерывает обработку программы пользователя и вызывает организационный блок ОВ25. После обработки программы, записанной в ОВ25, продолжается выполнение прерванной программы пользователя со следующей команды.

Если ОВ25 не запрограммирован, то CPU при появлении ошибки адресации переходит в "СТОП".

o Превышение времени цикла (ZYK)

Превышение времени цикла может возникнуть, например, из-за ошибки программирования, если при определенном состоянии процесса CPU попадает в бесконечный цикл, из-за неисправности тактового генератора или из-за временной перегрузки CPU (см. Раздел 3.6.1).

Если возникает превышение времени цикла, то операционная система прерывает обработку программы пользователя и вызывает организационный блок OB26. Время цикла обновляется.

Если OB26. не запрограммирован, то при возникновении ошибки превышения времени цикла CPU переходит в состояние "СТСП". Независимо от программирования OB26 при превышении времени цикла выдается сигнал BASP, блокирующий цифровые выходы. Сигнал BASP сбрасывается при возвращении к циклической обработке программы.

o Ошибка кода команды

Ошибка кода команды возникает в случае, если CPU не может интерпретировать или выполнить команду. Можно различить следующие ошибки кодов команд;

- Ошибка замещения

CPU при обработке программы пользователя внутри функционального блока выполняет подстановку, если выполняются операции с формальными операндами в функциональном блоке. формальные операнды замещаются при этом на указанные в вызове функционального блока фактические операнды (см. Раздел 2.7).

Недопустимая подстановка распознается процессором. Операционная система прерывает обработку программы пользователя и вызывает блок OB27.

— Недопустимый код операции

Недопустимый код операции возникает, если была запрограммирована команда, которая не вводит в список команд STEP—5 данного процессора (например, можно запрограммировать с помощью программатора команды RU и SU, однако R- и S-процессорами контроллера 135U они не интерпретируются").

При определении недопустимого кода операции обработка программы пользователя на этом месте прерывается и вызывается организационный блок OB29.

— Недопустимый параметр

Недопустимый параметр появляется, если программируется команда с параметром, который для данного процессора недопустим (например, операции с таймерами и счетчиками со значением параметров > 127).

Если процессор опознает недопустимый параметр, то операционная система прерывает обработку программы пользователя и вызывает организационный блок OB3O.

Команда, явившаяся причиной соответствующей ошибки кода команд, не обрабатывается, а вызывается соответствующий организационный блок. После того, как обработана программа в организационном блоке, продолжается обработка прерванной программы пользователя со следующей команды.
Если соответствующий организационный блок не запрограммирован, CPU переходит в состояние останова.

o Общая ошибка специальных функций

Общая ошибка специальных функций опознается операционной системой, если при обработке организационного блока специальной функции возникает ошибка или специальная функция отсутствует. В этом случае вызывается организационный блок OB31. Специальная функция дальше не выполняется. После обработки организационного блока OB31 программа продолжается с команды, следующей за вызовом организационного блока специальной функции. При отсутствии OB31 CPU переходит в состояние останова.

o Ошибка передачи

Если при передаче данных в блок данных (DB, DX) указанный параметр команды передачи превышает длину блока данных, то это определяется как ошибка передачи для защиты от ошибочной перезаписи данных в памяти. Ошибка передачи опознается также в том случае, если передача слова данных состоялась, хотя блок данных еще не открыт (с A DBxxx или AX DXxxx).

При определении ошибки передачи операционная система вызывает организационный блок OB32. Команда, ставшая причиной ошибки, больше не выполняется. Если OB32 не запрограммирован, то CPU переходит в режим останова.

o Ошибка контрольного времени

Если при обработке прерывания по времени (смотри Раздел 3.6.3) возникает запрос на следующее прерывание по времени, то операционная система вызывает организационный блок OB33. Если OB33 не запрограммирован, то CPU переходит в "СТОП".

o Ошибка регулятора

Ошибка при обработке функционального блока компактного регулятора, поддерживаемого операционной системой, распознается как ошибка регулятора. При возникновении ошибки регулятора обрабатывается организационный блок OB34. Если OB34 не запрограммирован, то CPU переходит в "СТОП". После обработки OB34 продолжается выполнение функций регулятора.

Если при возникновении вышеназванных причин неисправностей необходимо продолжать выполнение программы без какой-либо реакции, то достаточно в соответствующем организационном блоке ошибки запрограммировать единственную команду BE (конец блока). Обработка программы — циклической, прерывания по времени или прерывания от процесса — продолжится с команды STEP—5, на которой она была прервана.

При возникновении ошибки в организационном блоке ошибки происходит то же, что и в циклической программе, т.е. вызывается соответствующий организационный блок ошибки. При этом возможно до 7 уровней вложения блоков обработки ошибок. Затем происходит немедленный останов без нового вызова 0B; то же самое произойдет, если возникнет во второй раз ошибка того же типа (с тем же организационным блоком о шибки).

В программе запуска (организационные блоки с OB20 по OB22) возникшая ошибка обрабатывается как и в циклической программе. При 3.7.1 а организационный блок OB28 вызываться не будет. Если программа запуска прервана переходом в "СТОП", то допустим только новый пуск. Прерванный запуск не может быть продолжен. Пои запуске время цикла не контролируется.

Если необходимо прервать обработку программы при возникновении вышеописанных ошибок, когда организационный блок ошибки запрограммирован, то достаточно запрограммировать в нем команду останова. Процессор немедленно перейдет в "СТОП" (см. Раздел 3.7.1 b).

3.7.3 Управляющие биты и стек прерываний

С помощью функции On-line "AG-Инфо", а затем "Вывод USTACK" Вы можете проанализировать режим работы, свойства процессора и программы пользователя, а также возможные причины ошибок и прерываний.

Важно!
Вызов управляющих битов Вы можете произвести из любого режима, а вызов USTACK только в режиме "Стоп"

- Управляющие биты показывают текущее или предшествующее рабочее состояние с последней причиной неисправности. Если возникнет несколько ошибок, то управляющие биты укажут на все имеющиеся ошибки.

- В USTACK выдается текущая точка прерывания с текущим состоянием байтов индикации и содержимым аккумуляторов, а также причина неисправности.
Если возникнет несколько прерываний, то организуется многоступенчатый стек прерываний:
Уровень 01: = последняя причина прерывания,
Уровень 02: = предпоследняя причина прерывания и т.д.

При переполнении USTACK происходит немедленный переход в состояние останова.
После этого необходим новый пуск.

Значение аббревиатур управляющих битов и стека прерываний поясняется ниже.

Указание:
Приведенная здесь маска управляющим битов выводится на PG 635 с программным обеспечением PG S5-DOS

Что делать, если текст на экране Вашего программатора отличается от приведенного здесь?

Ориентируйтесь, в таком случае на позиции сокращений на экране.

Примеры:

1. Крестик стоит под управляющим битом в строке 6 на позиции 1. На Вашем экране он обозначен аббревиатурой "CHS-FE", а в данном руководстве по программированию на этой позиции стоит сокращение "DXO-FE" (см. следующую страницу). Описание "DXO-FE" в данном руководстве по программированию будет действительным для Вас!

2. В строке 8 помечена крестиком предпоследняя позиция, однако соответствующее сокращение отсутствует. Действительным в этом случае будет описание "REG-FE" в данном руководстве по программированию (см. ниже).

БИТЫ УПРАВЛЕНИЯ

>>STP<< STP-6	FE-STP	BARBEND	PGSTP	STPSCH	STP-BEF	MP-STP	
>>ANL<< ANL-6	NEUST	M W A	A W A	ANL-2	NEU-ZUL	MWA-ZUL	
	X				X	X	
>>RUN<< RUN-6	EINPROZ	BARB	OB1GEL	FBOGEL	OBPROZA	OBWECKA	
	X						
32KWRAM 16KWRAM	8KWRAM	EPROM	KM-AUS	KM-EIN	DIG-EIN	DIG-AUS	
	X				X	X	
URGELO URL-IA	STP-VER	ANL-ABB	UA-PG	UA-SYS	UA-PRFE	UA-SCH	
DXO-FE FE-22	MOD-FE	RAM-FE	DBO-FE	DB1-FE	DB2-FE	KOR-FE	
N A U P E U	B A U	STUE-FE	Z Y K	Q V Z	A D F	WECK-FE	
B C F FE-6	FE-5	FE-4	FE-3	L Z F	REG-FE	DOPP-FE	

При вызове USTACK на экран PG сначала появляется маска, показывающий состояние управляющих битов.

Следующие управляющие биты отражают текущее или предшествующее рабочее состояние процессора и дают информацию об определенных свойствах процессора и программы пользователя STEP 5.

Управляющие биты вызываются из любого режима работы!

Таким образом, Вы можете, например, одновременно увидеть . загружен ли организационный блок OB2 и возможна ли обработка программы управляемой по прерыванию от процесса.

STP Процессор в состоянии останова ("СТОП"); следующие управляющие биты выдают причину останова:

STP-6	не используется
FE-STP	Ошибка стоп: состояние останова после NAU (сбой питания:). PEU (неисправность периферии), BAU (отказ батареи), STUEB (переполнение BSTACK), STUEU (переполнение USTACK), DOPP (ошибка дублирования) или ошибка процессора.
BARBEND	Конец контроля обработки; Процессор в состояние; "СТОП" после выполнения функции ON-LINE "КОНЕЦ КОНТРОЛЯ ПРОГРАММЫ" (необходим новый пуск)
PG-STP	"СТОП" от PG: останов по команде от PG в однопроцессорном режиме или в многопроцессорном тестовом режиме
STP-SCH	Стоп от переключателя: останов от переключения переключателя режимов в положение "STOP"
STP-BEF	Стоп по команде: а) останов при обработке операции STEP5 'STP' б) останов по команде останова из операционной системы, если не запрограммирован организационный блок ошибки
MP-STP	Стоп в многопроцессорном режиме: а) переключатель выбора на KOR в "STOP" или б) останов другого CPU в многопроцессорном режиме в) команда стоп от PG в многопроцессорном режиме

ANL Процессор в режиме запуска;

ANL-6	Не используется
NEUST	Новый пуск запрашивается или активен или выполнен при последнем запуске
M W A	Ручной перезапуск запрашивается или активен или выполнен при последнем запуске
A W	Автоматический перезапуск или активен или выполнен при последнем запуске
ANL-2	Не используется
NEU-ZUL	Допускается новый пуск в качестве следующего запуска

MWA—ZUL	Допускается ручной перезапуск в качестве следующего запуска.
RUN	Процессор в режиме работы RUN (идет выполнение циклической программы):
RUN-6	не используется
EINPROZ	Однопроцессорный режим работы
BARB	Выполняется функция программатора ON-LINE "КОНТРОЛЬ ПРОГРАММЫ"
OB1GEL	Загружен OB1 в память пользователя. Выполняется циклическая программа, определяемая OB1.
FBOGEL	Загружен FBO в память пользователя. Циклическая программа определяется FBO, если при загрузке не загружен OB1. Если загружены и FBO, и OB1, циклически выполняется только OB1.
OBPROZA	Загружен организационный блок обработки прерываний OB2, т.е. допускается прерывание процесса
OBWECKA	Загружен организационный блок обработки прерываний по времени, т.е. допускается обработка прерываний, управляемых по времени.
32KRAM	Память пользователя RAM с 32К слов
16KRAM	Память пользователя RAM с 16К слов
SKRAM	Память пользователя RAM с 8К слов
EPROM	Память пользователя EPROM
KM-AUS	Адресный список для выходных маркеров связи из DB1 существует
KM-EIN	Адресный список для входных маркеров связи из DB1 существует
DIG-EIN	Адресный список для цифровым входов существует
DIG-AUS	Адресный список для цифровых входов существует
URGELOE	Выполнен полный сброс CPU (необходим новый ПУСК)
URL-IA	Выполняется полный серое
STP-VER	Процессор является причиной остановки контроллера
ANL-ABB	Прерывание во время запуска (требуется новый пуск)
UA-PG	Запрос полного сброса с программатора

UA-SYS Запрос на полный сброс от системной программы (не возможен запуск): должен быть выполнен полный сброс.

UA-PRFE Не используется

UA-SCH Подготовка к полному сбросу с помощью переключателей: Выполняется полный сброс или выбирается вид запуска, если запрошенный полный сброс выполнять не надо.

Следующие управляющие биты отмечают ошибки, которые могут возникнуть в режимах работы ЗАПУСК (например при первом новом пуске) и РАБОТА (например, при обработке программы управляемой по времени).

Если возникает несколько ошибок, то в последних трех строках управляющих битов будут указаны все, на данный момент имеющиеся и еще не обработанные, причины прерываний. Обратите внимание на системные данные BS2: они содержат UAMK (общее слово индикации прерываний, 16 бит), в которое заносятся все возникшие и еще не обработанные ошибки.

Ошибки при ЗАПУСКЕ:

DXO-FE Ошибка параметрирования в DX 0.

FE-22 не используется

MOD-FE Ошибка модуля памяти пользователя (необходим полный сброс)

RAM-FE Ошибка в памяти RAM системной программы или DB-RAM (необходим полный сброс).

DBO-FE Ошибка при создании списка адресов блоков в DBO

DB1-FE Ошибка при создании списка адресов периферии в DBO для обновления отображения процесса:

а) DB1 при установленном координаторе в многопроцессорном режиме не программируется;

б) адреса байтов вводов и выходов или маркеров связи, указанные в DB1 при новом пуске, не дают подтверждения на соответствующих модулях.

DB2-FE Ошибка при параметрировании блока данных DB 2 структуры регулятора R64

Ошибки при ЗАПУСКЕ или РАБОТЕ;

KOR-FE Ошибка во время обмена данными с координатором

NAU Сбой питания в центральном устройстве

PEU Входы/выходы не работают = отсутствует напряжение в устройстве расширения

BAU	Неисправность батареи = мало напряжение буферной батареи (в центральном устройстве)
STUE-FE	Переполнение стека прерываний или блоков (слишком велика глубина вложения; необходим новый пуск)
ZYK	Превышение допустимого времени цикла
CUZ	Задержка квитирования при обмене данными с периферией.
ADF	Ошибка адресации входов или выходов ошибка возникает при обращении к области отображения процесса модулей периферии, которые не были установлены при последнем новом пуске, дефектны или не указаны в DB1
WECK-FE	Ошибка будильника: новый запрос на обработку прерывания по времени перед или во время обработки прерывания
BCF	Ошибка кода команды: a) ошибка -замещения! обрабатываемая команда STEP5 не замещаема b) ошибка операции: обрабатываемая команда STEP5 не верна c) ошибка параметризации: параметр обрабатываемой команды STEP5 не верен
FE-6.	не используется
FE-5	не используется
FE-4	не используется
FE-3	не используется
LZF	0шибка выполнения: a) вызываемый блок не загружен b) ошибка передачи в блок данных[i] c) специальная ошибка выполнения (раньше общая ошибка спец. функций)
REG-FE	Ошибка при обработке структуры регулятора F64 в цикле
DOP-FE	Ошибка дублирования: a) повторная активизация активного уровня обработки программы ошибки (ADF, BCF, LZF, QVZ, REG, ZYK) (необходим новый старт). b) ошибка системы (необходим полный сброс)

После вывода управляющих битов на экран PG и нажатия клавиши подтверждения на экране появляется изображение стека прерываний (USTACK). Операционная система выдает сюда, при переходе и состояние останова, всю информацию, необходимую при новом пуске или перезапуске.

Указание:
Приведенную ниже маску USTACK Вы можете увидеть на PG 685 при использовании программного обеспечения PG-S5-DOS.

Что делать если текст на экране Вашего программатора не совпадает с приведенным здесь?

Ориентируйтесь в этом случае на позицию сокращения на экране.

Примеры:

1. В USTACK в качестве причины неисправности помечена крестом позиция 3 в строке 2. На вашем экране здесь стоит сокращение "TRAF", а в данном руководстве по программированию на этой позиции стоит сокращение "LZF" (см. следующую страницу). Описание "LZF" в данном руководстве по программированию будет действительно и для Вас.

2. В качестве причины неисправности помечена крестом последняя позиция в строке 2, однако соответствующее сокращение отсутствует. В этом случае будет действительно описание "DOPP" из данного руководства по программированию (см. ниже)!

СТЕК ПРЕРЫВАНИЙ

```
TIEFE:      02

BEF-REG:   C70A    SAZ:     00F3      DB-ADR:   0000    BA-ADR:   0038
BST-STP:   0000    OB-NR:    226      DB-NR:              -NR.:
                   REL-SAZ:0006       DBL-REG:  0000
VEK-ADR:   0004    UAMK:    0100      UALW:     0000

AKKU1:0000 C464 AKKU2:0000 00FF  AKKU3:0000 0000  AKKU4:0000 0000

СКОБКИ: KE1   111   KE2   100   KE3   111

БИТЫ РЕЗУЛЬТАТА:    ANZ1   ANZO   OVFL   OVFLS ODER   STATUS VKE   ERAB
                           X                          X      X
ПРИЧИНА ОШИБКИ:     NAU    PEU    BAU    MPSTP ZYK    QVZ    ADF   STP
                                                                X
                    BCF    S-6    LZF    REG   STUEB  STUEU  WECK  DOPP
```

Следующие метки USTACK содержат указания на место ошибки, с помощью которые можно найти в программе пользователя инструкцию, при обработке которой процессор перешел в стоп.

TIEFE — Уровень стека прерываний при вложенных ошибках
TIEFE 01 = последняя по времени причина прерывания
TIEFE 02 = предпоследняя по времени причина прерывания

BEF-REG — Регистр команд: Содержит машинные коды (первое слово) следующей выполняемой команды прерванного уровня обработки программы

BST-STP — Указатель стека блоков

EBENE Z — Указывает уровень обработки программы, который был прерван:
Z: 0002 = новый пуск или ручной перезапись
0004 = цикл
0006 = прерывание по времени
0008 = регулятор
000A = прерывание от процесса
000C = прерывание (переход в "СТОП" при сигнале останова от переключателя, от PG или ROR)
0010 = ошибка "будильника"
0012 = ошибка регулятора
0014 = ошибка цикла
0016 = автоматический перезапуск
0018 = ошибка кода команды
001A = ошибка выполнения команды
001C = ошибка адресации
001E = задержка квитирования

SAZ — Счетчик адреса:
содержит абсолютный адрес команды, которая должна выполняться следующей на прерванном уровне обработки программ, в программной памяти. (Абсолютный адрес - 1 или - 2 = причина прерывания одно- или двухсловная команда!)
Если ошибка наводится на уровне STEP5, например, при обработке регулятора, то SAZ устанавливается в "О", а содержимое BEF-REG не имеет значения.

...-NR — Тип и номер последнего обрабатываемого блока

REL-SAZ — Счетчик относительного адреса: содержит адрес команды, которая будет выполняться следующей, относительно начала текущего блока (относительный адрес показывается программатором при установке ключевого переключателя в положение "блокировка ввода", или при выводе блока на печать)

UAMK	Слово маски отображения прерываний; в UAMK указываются все имеющиеся на данный момент и еще не обработанные причины прерываний.
UALW	Слово стирания отображения прерываний
DB-ADR	Абсолютный начальный адрес выбранного последним блока данных в программной области (DB 0) (DB-ADR = OOOOH, если DB не был адресован)
DB-NR	Номер вызванного в данный момент блока данных
DBL-P.EG	Длина вызванного в данный момент блока данных
BA-ADR	Абсолютный адрес следующей команды, которая должна была быть выполнена в последнем вызывающем блоке
...-NR	Тип и номер блока последнего вызывающего блока
AKKU1...4	Содержимое регистров вычислений на момент прерывания В определенным случаям ошибок операционная система в момент прерывания заносит в Акк 1 и 2 метки ошибок, которые подробнее поясняют причину прерывания.
СКОБКИ	Количество уровней скобок: 'KEx abc' x = от 1 до 7 уровней a = OR (или, см. индикацию битовых операций) b = VKE (результат логической операции, см. индикацию битовых операций) c = 1: U(c = 0: 0(
БИТЫ РЕЗУЛЬТАТА	См раздел 4.1

Следующие сокращения представляют собой важнейшие причины ошибок. Крестом помечаются только те причины прерываний, которые возникли в индицируемом в данный момент уровне обработки программ (см. Уровни!).

При указании причин ошибок они трактуются из отображения слова маски отображения прерываний (UAMK, 16 бит). Частично указания здесь идентичны управляющим битам.

NAU	Неисправность сетевого питания в центральном устройстве
PE 1-1	Блокировка входов/выходов неисправность питания в устройстве расширения
BAU	Неисправность батареи = Недостаточно напряжение буферной батареи в центральном устройстве
MPSTP	Стоп в многопроцессорном режиме: a) переключатель выбора на KOR в "STOP" или b) останов другого CPU в многопроцессорном режиме в) команда стоп от PG в многопроцессорном режиме

ZYK	Превышение времени цикла
QVZ	Задержка квитирования при обмене данными с периферией
ADF	Ошибка адресации входов или выходов
STP	Останов при выборе переключателем режима работы "STOP" Останов по команде от PG при однопроцессорном или многопроцессорном тестовом режиме Останов при обработке операции STEPS 'STP' Останов по команде останова от операционной системы, если не запрограммирован организационный блок ошибки
BCF	Ошибка кода команды: a) ошибка замещения: обрабатываемая команда STEP5 не замещаема b) ошибка операции; обрабатываемая команда STEP5 не верна c) ошибка параметризации: параметр обрабатываемой команды STEP5 не верен
S-6	Не используется
LZF	Ошибка выполнения: a) вызываемый блок не загружен b) ошибка передачи в блок данных c) специальная ошибка выполнения (раньше общая ошибка спец функций)
REG	Ошибка при обработке структуры регулятора R64 в цикле
STUEB	Переполнение стека блоков- (превышение глубины вложения; необходим новый пуск)
STUEU	Переполнение стека прерываний (превышение глубины вложения; необходим новый пуск)
WECK	Ошибка будильника: новый запрос на обработку прерывания по времени перед или во время обработки прерывания
DOPP	Ошибка дублирования: a) повторная активизация активного уровня обработки программы ошибки (ADF, BCF, LZF, QVZ, REG, ZYK:) (необходим новый старт). b) ошибка системы (необходим полный сброс)

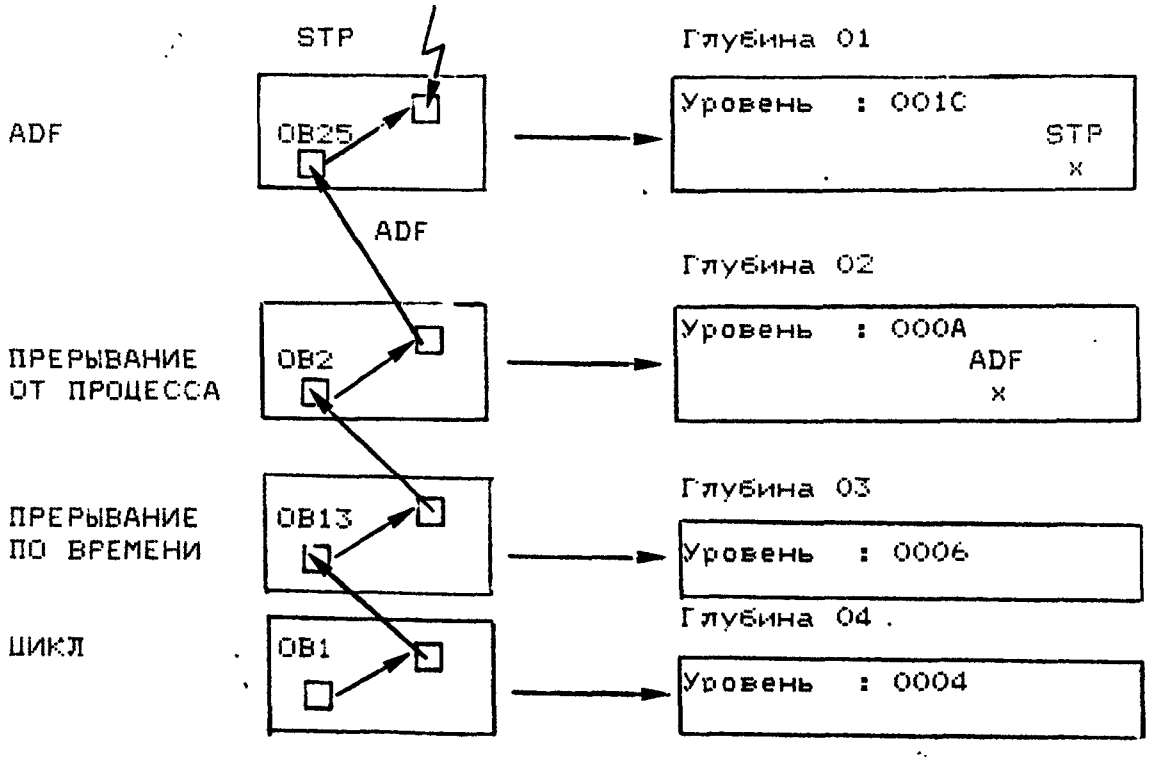
Примеры по обработке USTACK

Пример 1:

На следующем изображении Вы видите структуру USTACK в зависимости от возникающих прерываний.

1 . Уровень обработки программ "ЦИКЛ" (OB1) прерван появлением прерывания по времени.
2. При этом активизируется уровень обработки программ "ПРЕРЫВАНИЕ ПО ВРЕМЕНИ" и обрабатывается OB13.
3. Из-за возникновения прерывания от процесса происходит выход из уровня "ПРЕРЫВАНИЕ ПО ВРЕМЕНИ", активируется уровень "ПРЕРЫВАНИЕ ОТ ПРОЦЕССА" и обрабатывается OB2.
4. Неверная команда адресации ведет к тому, что активизируется уровень ."ADF" и на нем обрабатывается OB25. В его программе обработки ошибок пользователь запрограммировал команду останова (STP). Процессор прервал обработку программы.

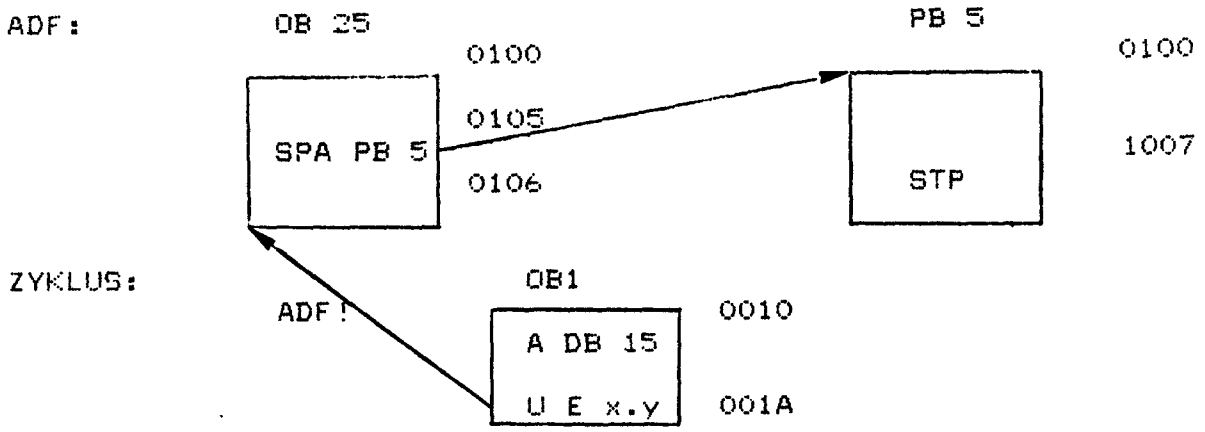
Перед завершающим переходом в состояние останова было прервано с различных уровнях обработки программ. Если Вы теперь вызовете на экран PG USTACK, то Вы получите четырехуровневый USTACK с глубиной вложения 01, в котором содержатся метки уровня обработан программ прерванного последним (=ADF). Вы можете теперь "погружаться" в USTACK пока не достигнете USTACK с глубиной вложения 04, который содержит уровень обработки программ ЦИКЛ, прерванный первым.



Уровни обработки программ . USTACK

Пример 2:
В данном примере процессор при выполнении команды 'U E x.y' в OB1 опознает ошибку адресации. Это приводит *к* обработке 0B 25. При обработке команды STP в PB5 процессор переходит в СТОП.

Два прерванных уровня обработки программ создают двухуровневый USTACK:




```
С Т Е К   П Р Е Р Ы В А Н И Й

TIEFE:      01

BEF-REG:  STP     SAZ:     1007     DB-ADR:          BA-ADR:  0106
BST-STP:    3     OB-NR:    226     DB-NR:    16       -NR.:  25
                  REL-SAZ:0006      DBL-REG:

УРОВЕНЬ:  001C    UAMK:    0300     UALW:     0000

AKKU1:

БИТЫ РЕЗУЛЬТАТА:    ........

ПРИЧИНА ОШИБКИ:                                          STP
                                                          X
```

```
С Т Е К   П Р Е Р Ы В А Н И Й

TIEFE:      02

BEF-REG: U Ex.y   SAZ:     100A     DB-ADR:          BA-ADR:  0000
BST-STP:    1     OB-NR:       1     DB-NR:    16       -NR.:
                  REL-SAZ:000A      DBL-REG:

УРОВЕНЬ:  0004    UAMK:    0200     UALW:     0000

AKKU1:

БИТЫ РЕЗУЛЬТАТА:    ........

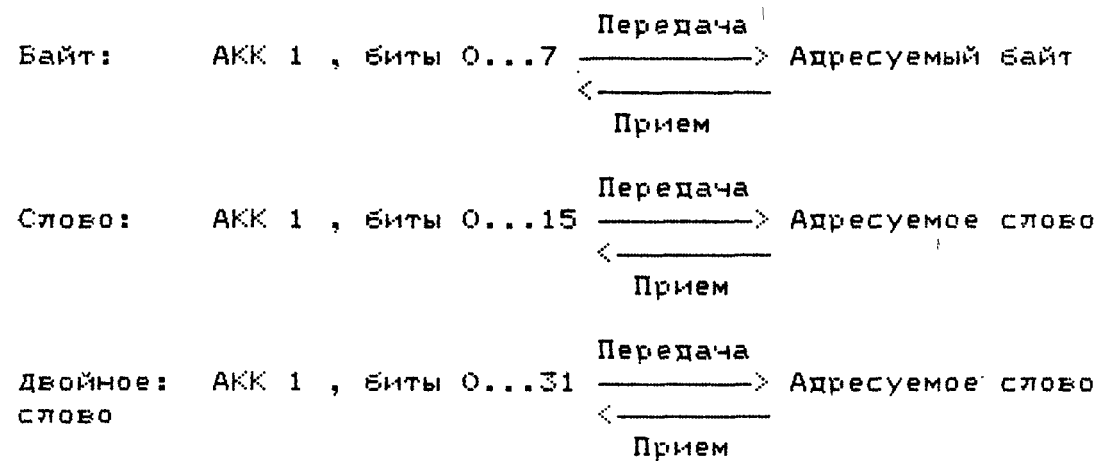
ПРИЧИНА ОШИБКИ:                                          ADF
                                                          X
```

4 Набор команд языка STEP—5 с примерами программирования

4.1 Общие правила .

Большинство операций STEP—5 используют два регистра (32 бита) в качестве исходного операнда и в качестве назначения для результата. Это аккумулятор 1 (АКК 1) и аккумулятор 2 (АКК 2).

В зависимости от метода адресации (байтовая, словная или двойными словами) команды загрузки и передачи используют содержимое аккумуляторов следующим образом:



АКК 1 всегда является источником для операции передачи и назначением для операций загрузки.

При операциях загрузки байта или слова старшие биты в АКК. 1, которые не используются, заполняются нулями. Перед этим старое содержимое передается из АКК 1 в АКК 2.
При пословных операциях загрузки/передачи из/в побайтово организованную область памяти (PAA, PAE, P-, Q- периферия, маркеры, шина S5) загружаются/передаются байты n и n + 1.
При выполнении операций передачи содержимое АКК 1 и 2 остается без изменений. Вспомогательные регистры (АКК 3 и 4) остаются без изменений во время выполнения всех операций загрузки и передачи.
Обработка команд STEP-5 аналогична ПК 150 S. Отличия оговариваются.

o Представление чисел

В качестве операндов для команд языка STEP—5, которые выполняют логические действия, сравнение или изменение содержимого АКК 1 и 2, могут быть использованы числа в различных видах представления. В зависимости от выполняемой операции содержимое АКК 1 и 2 интерпретируется в одном из следующие; видов:

а") Число с фиксированной запятой; интерпретируется 16—битовым двоичным числом в виде двоичного дополнения (расширение числа с фиксированной запятой с 16 до 32 бит, (см. Раздел 5.1). Ввод с PG: L KF Z, причем -32768 <.Z < +32767.

b) Число в коде BCD; со знаком и 3 цифры; значение битов в АКК 1;

Виты	15...12	11...8	7...4	3...0
	Знак	Сотни	Десятки	Единицы

Отдельные цифры представляются Б виде положительных 4-битовые чисел в двоичном представлении.

Знак: 0000 - если число положительное
1111 - если число отрицательное

c) Числа с плавающей запятой; Интерпретируются как 32-битовые двоичные числа с 8-битовой экспонентой и 24-битовой мантиссой. При выполнении операций с числами с плавающей .запятой +G, -G, xG и :G в ПК 135U мантисса представляется 16-битовым числом; 8 младших битов устанавливаются в О.

Пример: (Ввод с программатора числа Z с плавающей запятой')

```
Z = 12.34567                    Z = —0.005
L KG + 1234567 + 02             L KG — 50000000 — 02
       ———————  ——                    ———————  ——
          |      |                       |      |
      Мантисса  Экспонента            Мантисса  Экспонента
                (основание 10)                  (основание 10)

Z = +0.1234567x10⁺²=12.34567    Z = —0.5 x 10⁻² = —0.005
```

Примечание: Внутреннее представление может не соответствовать Формату, в котором числа вводятся при создании программы на программаторе. (Смотри Руководство по программатору). Программатор создает представления чисел, показанные выше.

o Биты результата

Есть команды для обработки информации, состоящей из отдельных битов, и команды для обработки информации в виде байтов и слов. (8, 16 или 32 бит). В обеих группах есть команды, которые устанавливают коды условий и команды, которые интерпретируют их (смотри приложение: Список инструкций, влияние на коды условий). Есть коды условий битовые (биты с О до 3) и коды условий словные (биты с 4 по 7), которые соответствуют группам команд. Байт кодов условий может быть выведен с помощью программатора в виде:

	Коды условий словные				Коды условий битовые			
	ANZ1	ANZO	OV	OS	OR	STA	VKE	ERAB
Бит	7	6	5	4	3	2	1	0

Объяснение битовых кодов условий:

ERAB — Первый опрос; Начало логической операции. В конце цепочки логической операции (операций с памятью) ERAB устанавливается в состояние 0. Команды, которые устанавливают ERAB в 0 (например, команда присвоение = Ax.x), ограничивают VKE (смотри приложение), т.е. в дальнейшем результат логической операции остается постоянным. Он может быть интерпретирован (например, в командах зависимых от VKE), но не может быть изменен. Только после первой логической команды (означающей начало цикла сканирования) VKE может быть изменено.

VKE — Результат логической операции; Результат битовой логической операции. Команда "истинно" при выполнении команды сравнения (смотри Приложение: Список команд, двоичные логические операции или операции сравнения).

STA — Статус; В битовых командах определяется логическое состояние бита, который только что был опрошен или установлен. Статус .обновляется при двоичных логических операциях < исключая U(, 0(,), 0> и операциях с памятью.

OR — ИЛИ; информация для CPU, что следующая логическая операция "И" должна быть выполнена перед логической операцией "ИЛИ" (И перед ИЛИ).

Объяснение словных кодов условий;

OV — Переполнение; Определяется, произошло ли переполнение (превышение допустимых границ) при выполнении последней арифметической операции.

OS — Запоминание переполнения; Бит переполнения запоминается. Этот бит используется для индикации уровня, на котором произошло переполнение при выполнении нескольких арифметических операций.

ANZ1 и ANZO - биты кодируемого результата, которые интерпретируются в соответствии со следующей таблицей.

Биты результата		Результат вычисления с фиксированн. запятой	Цифровая логическая операция	Сравнение содержимого АКК 1 и АКК 2	Сдвиг: последний сдвинутый бит
ANZ1	ANZO				
0	0	Результат = 0	= 0	АКК 2 = АКК 1	0
0	1	Результат <.0	-	АКК 2 < АКК 1	-
1	0	Результат > 0	<>0	АКК 2 > АКК 1	1

Операции переходов позволяют непосредственным образом интерпретировать коды (Раздел 4.3).

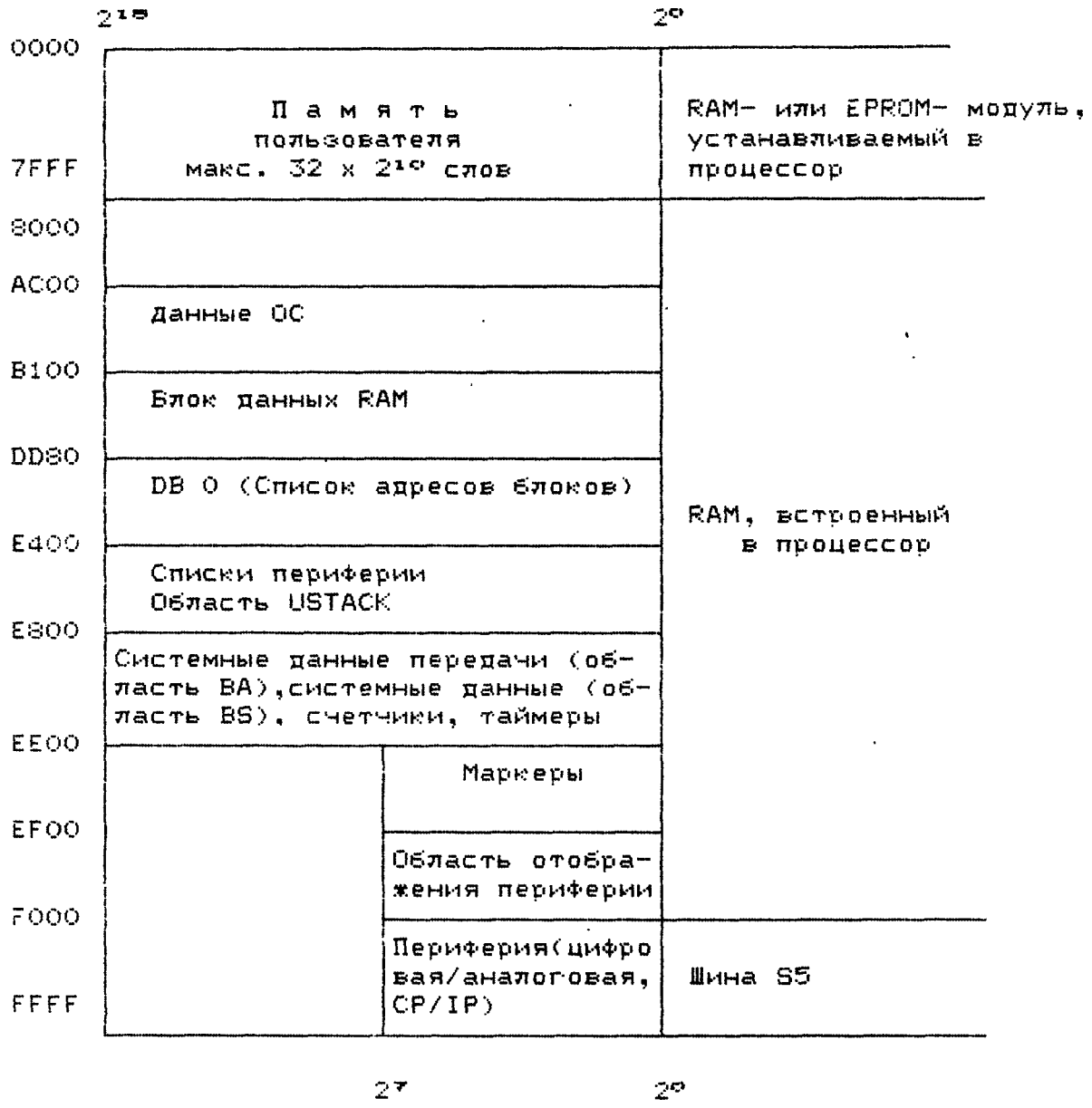


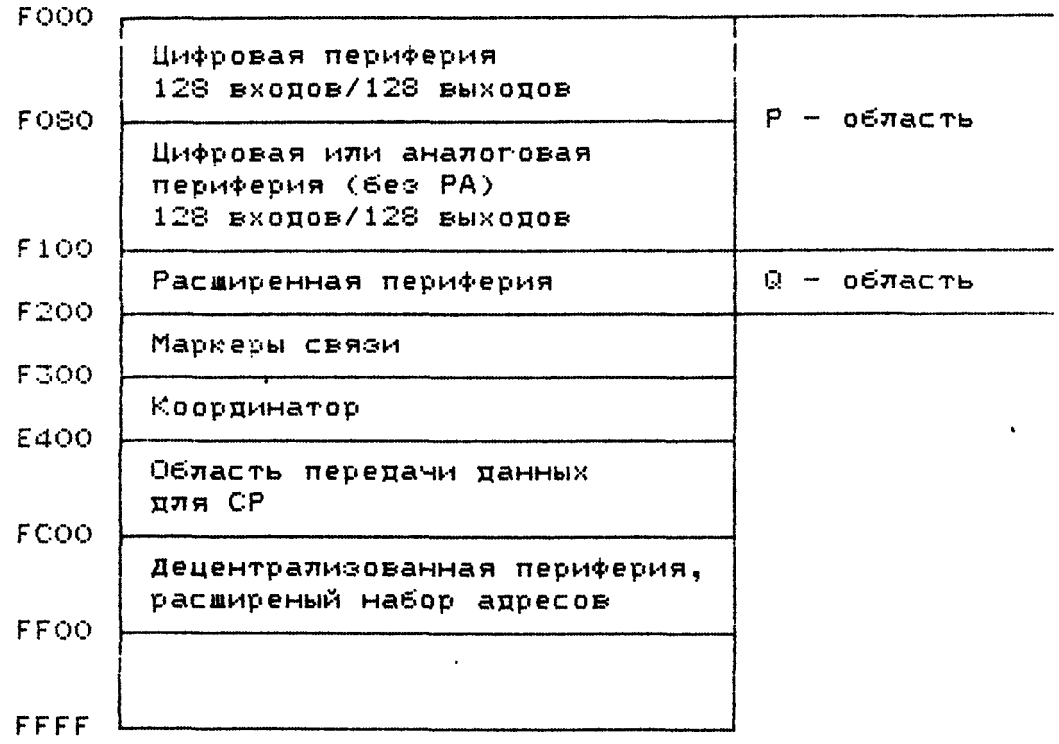
Рис 17. Распределение памяти

о Адресные области для входов/выходов/программ

Область (абсолютный адрес)		Адресуется с помощью	Параметр
EF00 EF7F	PAE (отображение входов)	L EB/T EB L EW/T EW L ED/T ED L E/UN E/O E/ON E S E/R E/=E	0...127 0...126 0...124 0.0...127.7
EF80 EFFF	PAA (отображение выходов)	L AB/T AB L AW/T AW L AD/T AD L A/UN A/O A/ON A S A/R A/= A	0...127 0...126 0...124 0.0...127.7
F000 F07F	Цифровые входы/выходы	L PB/T PB L PW/T PW	0...127 0...126
P—периферия с отображением процесса			
F080 F0FF ~~EF7F~~	Цифровые или аналоговые входы/выходы	L PB/T PB L PW/T PW	128...255 128...254
P—периферия без отображения бражения процесса			
F100 ~~F080~~ F1FF ~~EF7F~~	Область расширенной периферии	L QB/T QB L QW/T QW	0...255 0...254
Q — периферия			

Рис.18 Распределение адресов в ПК S5-135U

o Область периферии на шине S5



4.2 Основной набор операций

о Двоичные логические операции

Операция	Параметр	Функция
)		Скобку закрыть
U(		Операция "И" для выражения в скобках
O(		Операция "ИЛИ" для выражения в скобках
O		Операция "ИЛИ" над функциями "И"
U		Операция "И"
O		Операция "ИЛИ"
E	0.0...127.7	с опросом входа на состояние "1"
A	0.0...127.7	с опросом выхода на состояние "1"
M	0.0...255.7	с опросом маркера на состояние "1"
N E	0.0...127.7	с опросом входа на состояние "0"
N A	0.0...127.7	с опросом выхода на состояние "0"
N M	0.0...255.7	с опросом маркера на состояние "0"
T	1...127	с опросом таймера на состояние "1"
N T	1...127	с опросом таймера на состояние "0"
Z	1...127	с опросом счетчика на состояние >0
N Z	1...127	с опросом счетчика на состояние =0

Двоичные логические операции создают результат логической операции (VKE).

В начале выполнения цепочки логических операций (первый опрос) результаты первой логической операции зависят только от состояния опрашиваемых сигналов и от того, используется или нет их инверсное состояние (N - отрицание); однако, они не зависят от типа логической операции (U = UND <"И"> , O = ODER <"ИЛИ">).

Во время выполнения цепочки логических операций VKE формируется в зависимости от типа логической операции, предыдущего состояния VKE и состояния опрашиваемого сигнала. Цепочка логических операций завершается командой, ограничивающей VKE (ERAB = O) (например операция установки).

VKE остается без изменений до тех пор, пока не начнется следующий "первый опрос". После этого VKE может быть интерпретирован, но не может быть установлен.

Программа	Статус	VKE	ERAB	
:				
= A 0.0	0	0	0	← ограничение VKE
U E 1.0	1	1	1	← первый опрос
U E 1.1	1	1	1	
U E 1.2	0	0	1	
= A 0.1	0	0	0	← ограничение VKE, конец цепочки логической операции

o Операции установки/сброса

Операция	Параметр	Функция
S		Установка
R		Сброс
=		Присвоение
E	0.0...127.7	входов в PAE
A	0.0...127.7	выходов в PAA
M	0.0...255.7	битов маркеров
D	0.0...255.15	битов слова данных

o Функции загрузки, передачи и сравнения

Операция		Параметр	Функция
L			Загрузка
T			Передача
E	B	0....127	входного байта в/из PAE
E	W	0....126	входного слова в/из PAE
E	D	0....124	входного двойного слова в/из PAE
A	B	0....127	выходного байта в/из PAA
A	W	0....126	выходного слова в/из PAA
A	D	0....124	выходного двойного слова в/из PAA
M	B	0....255	маркерного байта
M	W	0....254	маркерного слова
M	D	0....252	маркерного двойного слова
D	R	1....255	младшего байта слова данных из DB
D	L	1... 255	старшего байта слова данных из DB
D	W	1....255	слова данных из DB
P	B	0....127	периферийного байта модулей цифровых входов/выходов (зависит от PAE/PAA)
P	B	128...255	периферийного байта модулей аналоговых входов/выходов
Q	B	0...255	байта расширенной периферии (область Q)

Операция	Параметр	Функция
L		Загрузка
T		Передача
P W	0...126	периферийного слова модулей цифровых входов/выходов (область P)
P W	128...254	периферийного слова модулей аналоговых входов/выходов (область P)
Q W	0...254	слова расширенной периферии (область Q)
L		Загрузка
K M	любая битовая комб.(16 бит)	константы в виде битовой комбинации
K H	0...FFFF	константы в 16-ричном коде
K F	от -32768 до +32768	константы в виде числа с фиксированной запятой
K Y	0...255 для кажд. байта	двухбайтовой константы
K B	0...255	однобайтовой константы
K C	2 любых алфав-цифр.символа	константы (2 символа в коде ASCII)
K T	0.0...999.3	константы времени
K Z	0...999	константы счета
K G	1)	константы в виде числа с плавающей запятой (32 бита)
T	0...127	значения таймера
Z	0...127	значения счетчика
LC T	0...127	BCD-кодированное значения таймера
LC Z	0...127	BCD-кодированное значения счетчика
<>		Сравнение на равенство
><		Сравнение на неравенство
>		Сравнение на больше
>=		Сравнение на больше/равно
<		Сравнение на меньше
<=		Сравнение на меньше/равно
F		двух чисел с фиксированной запятой (16 бит)
D		двух чисел с фиксированной запятой (32 бита)
G		двух чисел с плав. запятой (32 бита)

1) ё 0.1469368x10^{-39} ....ё 0.1701412x 10^{39}

Операции загрузки и передачи не влияют на коды условий. Команды сравнения устанавливают в качестве результата VKE и коды ANZO и ANZ1. Всегда сравнивается содержимое аккумуляторов АКК 1 и АКК 2 (смотри примеры программ и список команд).

Для операций загрузки и передачи следует обратить внимание на инструкции в Разделе 6.1. Обращение к входам/выходам может быть выполнено непосредственным способом — с помощью операций загрузки и передачи L/T PB, PM, QB, QW или через область отображения процесса — с помощью операций L/T EB, EW, ED, AB, AW, AD и с помощью логических операций. При использовании операций T PB O...127 и T PW O...126 одновременно устанавливается и PAA (PAE/PAA — отображение процесса входов/выходов для 128 байтов периферии с адресами от O до 127).

Отображение процесса представляет собой область памяти, содержание которой только выдается на периферию (PAA) или считывается с периферии (PAE) один раз за цикл выполнения программы пользователя (см. Рис. 13). Это исключает частое изменение логического состояния битов в процессе выполнения циклической программы пользователя, которое может привести к "дребезгу" соответствующих выходных устройств.

Q-область может быть адресована только при использовании интерфейсных модулей 300 или 301, так как модули входов/выходов, адресуемые в Q-области, могут быть установлены только в устройство расширения. Для всей Q- и P-области с относительными адресами байтов от 128 до 255 отображения процесса нет.

С помощью операций загрузки и передачи слов для адресации области, имеющей байтовую структуру (PAA, PAE, маркеры, шина S5), загружается/передается байт n и n+1; при использовании двойных слов загружаются/передаются байты с n по n+3.

Пример

L EW 5 в АКК 1 загружаются байты 5 и 6 из области PAE

L MD 10 загружаются 10,11,12 и 13 байты маркеров

o Операции с таймерами и счетчиками

Перед тем, как загрузить таймер или счетчик, необходимо предварительно значение времени или счета загрузить в АКК 1.

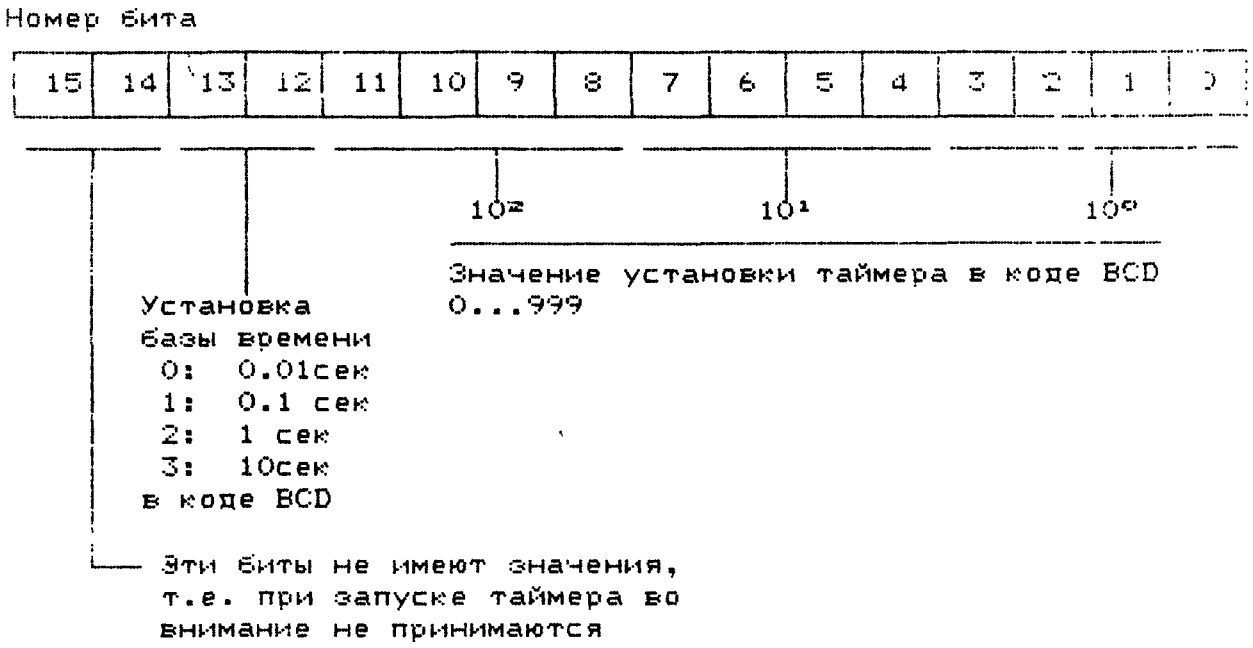
Имеют смысл следующие операции загрузки:
Для таймеров: L KT, L .EW, L AW, L MW, L DW
Для счетчиков: L KZ, L EW, L AW, L MW, L DW

Операция	Параметр	Функция
SI T	1…127	Запуск таймера в качестве импульса
SV T	1…127	Запуск таймера в качестве удлиненного импульса За-
SE T	1…127	пуск таймера в качестве задержки включения
SS T	1…127	Запуск таймера в качестве запоминающей задержки включения
SA T	1…127	Запуск таймера в качестве задержки выключения
R T	1…127	Сброс таймера
S Z	1…127	Установка счетчика
R Z	1…127	Сброс счетчика
ZV Z	1…127	Прямой счет
ZR Z	1…127	Обратный счет

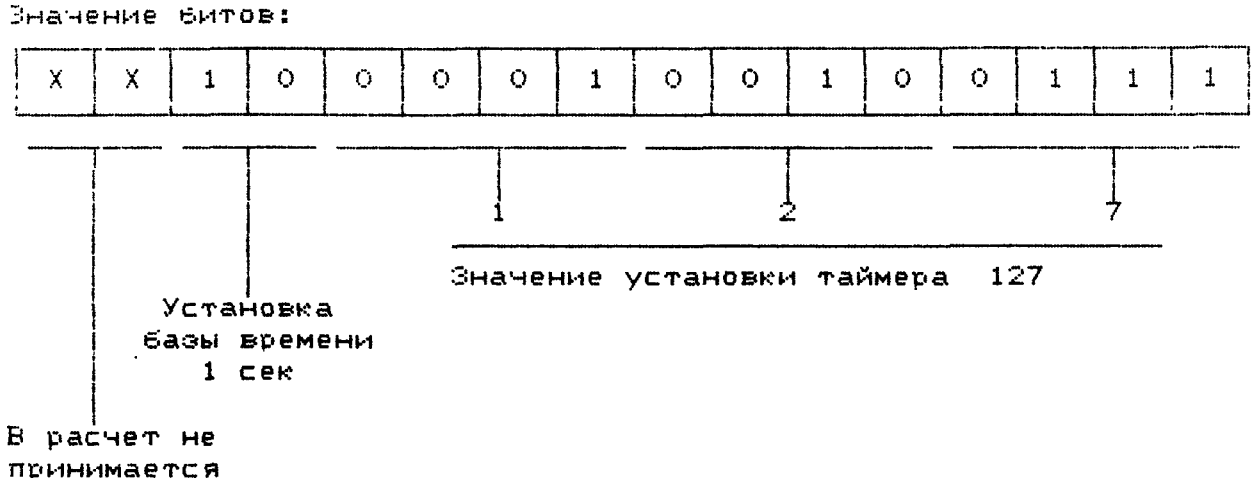
При использовании операций таймеров SI, SE, SV, SS, SA и S или счетчиков, значение, записанное в аккумуляторе 1, переносится в область памяти, соответствующую данному таймеру/счетчику (команде передачи), и начинает выполняться соответствующая операция.

Если ..значение таймера или значение счетчика загружается при помощи EW, AW, MW или DW, то соответствующее слово должно иметь следующую структуру:

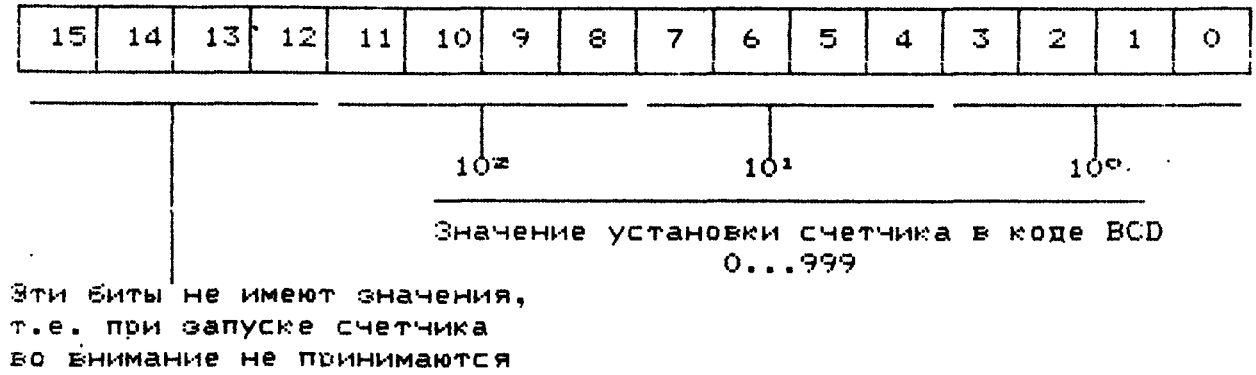
Для значения таймера



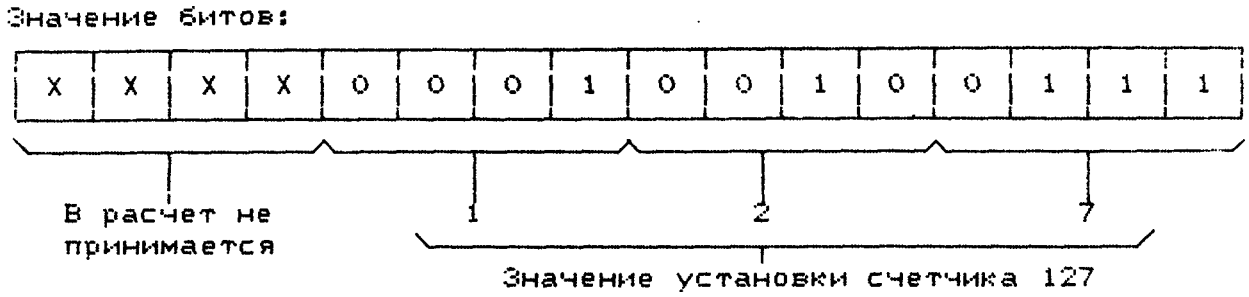
Пример: Установка времени 127 секунд



Для значения счетчика

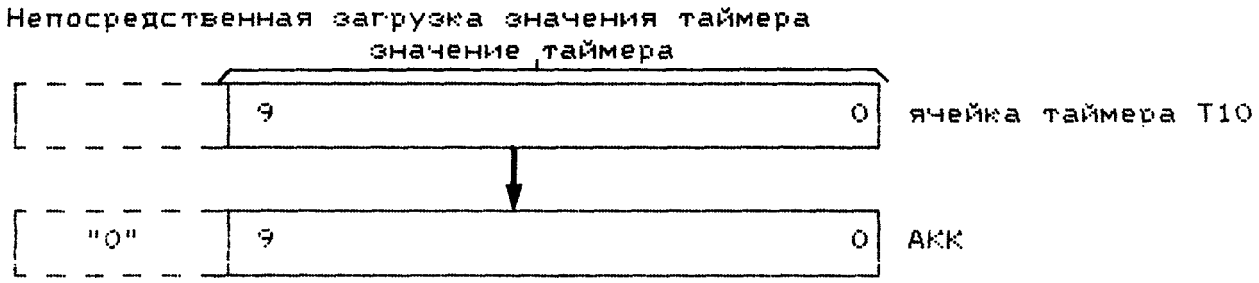


Пример: Установка значения счетчика 127



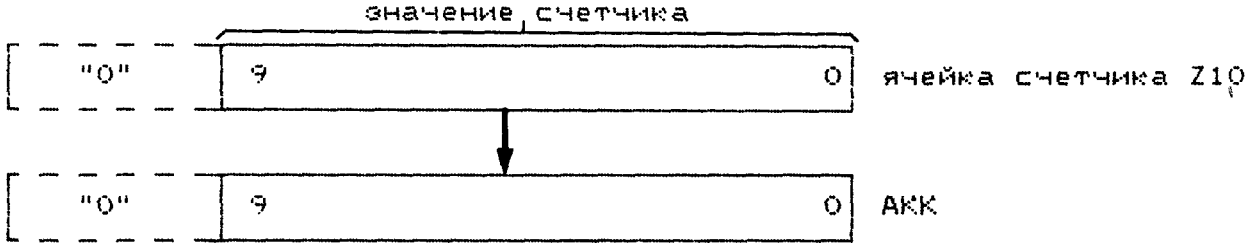
Значение таймера или счетчика хранится в области памяти, отведенной для этих целей, в виде двоичных кодов. Для того, чтобы опросить таймер или счетчик, значение из области таймеров или счетчиков должно быть перенесено в АКК 1 непосредственна или в коде BCD.

Пример:



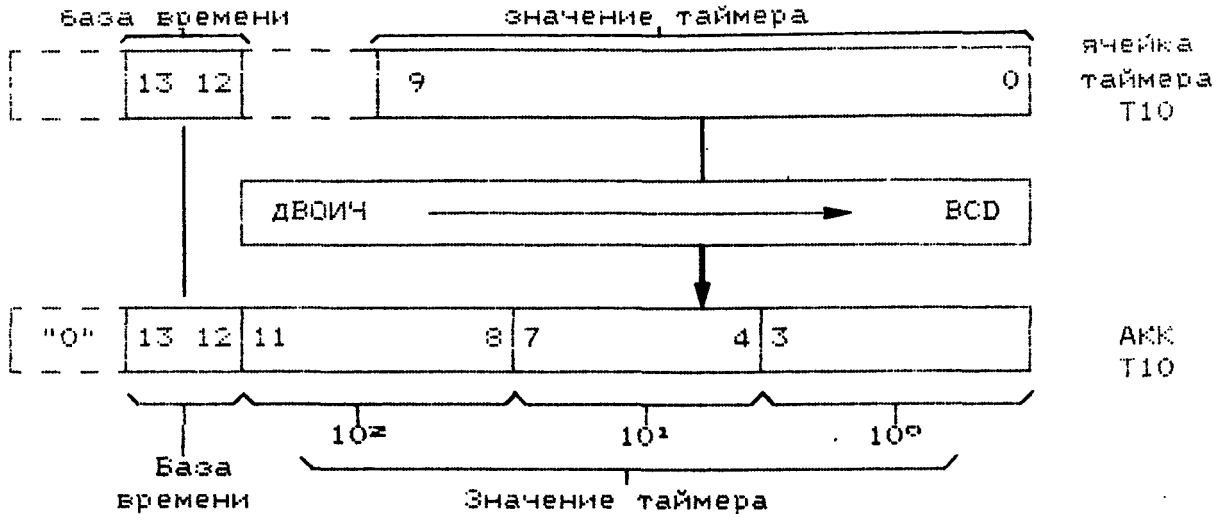
L T 10 непосредственная загрузка двоичного значения таймера T10 в АКК
Значение базы времени здесь не загружается

Непосредственная загрузка значения счетчика;



L Z 10 непосредственная загрузка двоичного значения счетчика Z10 в АКК

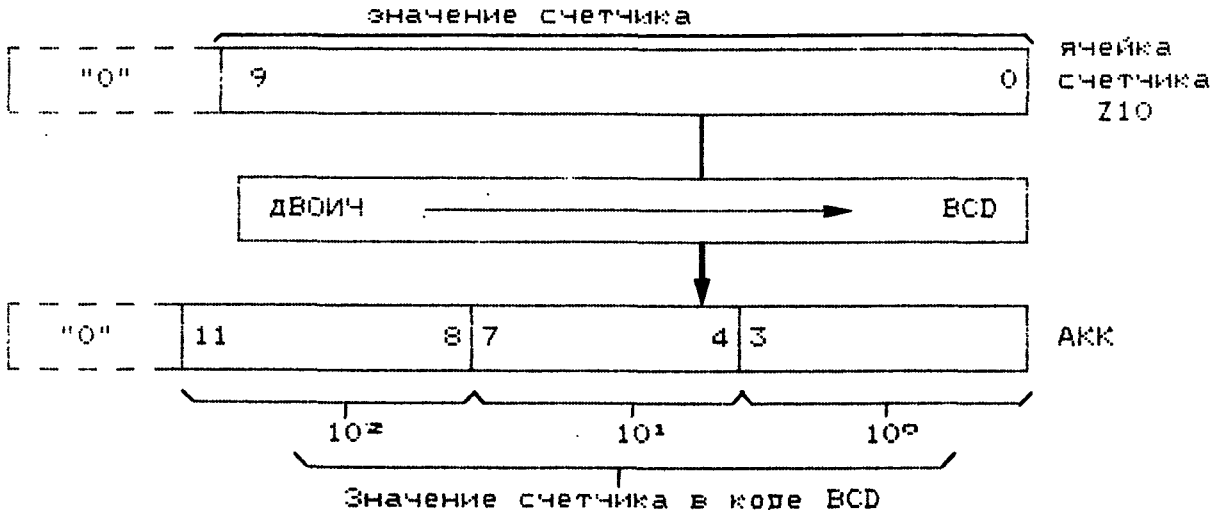
Кодированная загрузка значения таймера;



LC T 10 кодированная загрузка значения таймера T10 и базы времени в АКК

Значение базы времени здесь тоже загружается.

Кодированная загрузка значения счетчика:



LC Z 10 кодированная загрузка значения счетчика Z10 в АКК

При кодированной загрузке значение битов 14 и 15 ячейки таймера и значение битов с 12 по 15 ячейки счетчика не загружается. На их место в АКК 1 записывается "О". После этого содержимое АКК может быть нужным образам обработана.

o Арифметические операции

Операция	Параметр	Функция
+ F		Сложение двух чисел с фиксированной запятой (16 бит)
-F		Вычитание двух чисел с фиксированной запятой (16 бит)
x F		Умножение двух чисел с фиксированной запятой (16 бит)
: F		Деление двух чисел с фиксированной запятой
+ G		Сложение двух чисел с плавающей запятой
- G		Вычитание двух чисел с плавающей запятой
x G		Умножение двух чисел с плавающей запятой
: G		Деление двух чисел с плавающей запятой

Арифметические операции выполняются над содержимым аккумуляторов 1 и 2 (смотри список операций). Затем результат помещается в АКК 1. Арифметические регистры изменяются при арифметических операциях следующим образом:

```
<АКК 1> : = результат
<АКК 2> : = <АКК 3>
<АКК 3> : = <АКК 4>
<АКК 4> : = <АКК 4>
```

Предыдущее содержимое аккумулятора (АКК 2) теряется.

a Вызов блоков

Операция	Параметр	Функция
SPA		Переход безусловный
SPB		Переход по условию (зависит от VKE)
O B	1...39	к организационному блоку
O B	40...255	к системной программе специальной функции
P B	1...255	к программному блоку
F B	1...255	к функциональному блоку
S B	1...255	к шаговому блоку
A D B	3...255	Вызов блока данных
AX D X	1...255	Вызов дополнительного блока данных DX
BE		Конец блока
BEA		Безусловный конец блока
BEB		Конец блока условный, зависит от VKE

o Пустые операции

Операция	Параметр	Функция
NOP NOP BLD	0 1 0... 255	Пустая операция Пустая операция Команды формирования изображения на экране програм-матора (не интерпретируется как команда CPU)

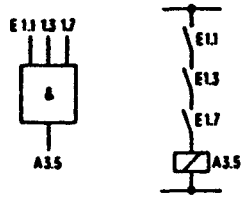
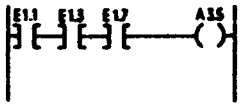
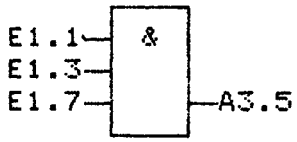
o Команда остановки

Операция	Параметр	Функция
STP		CPU переходит в состояние "СТОП"

Примеры программирования логических функций, таймеров, счетчиков и функций сравнения

о Логические функции

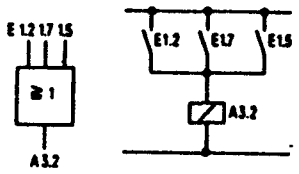
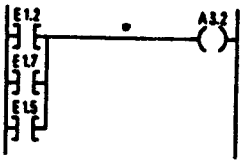
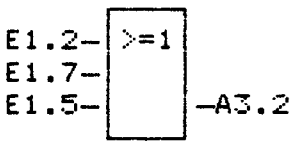
Операция "И"

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E1.1 U E1.3 U E1.7 = A3.5		

На выходе A3.5 индицируется "1", если на всех входах одновременно присутствует "1".

На выходе A3.5 "О", если хотя бы на одном из входов "О". Количество опросов и последовательность программирования произвольны.

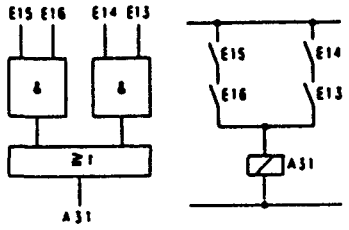
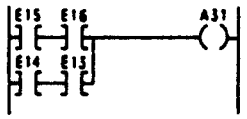
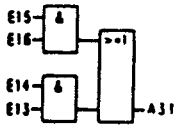
Операция "ИЛИ"

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	O E1.2 O E1.7 O E1.5 = A3.2		

На выходе A3.2 "1", если по крайней мере на одном из входов "1". На выходе A3.2 "О", если на всех входах одновременно "О". Количество опросов и последовательность программирования произвольны.

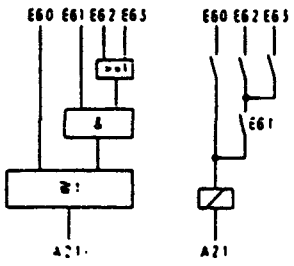
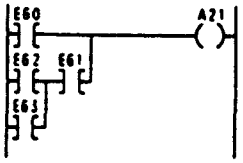
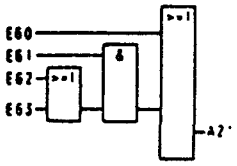
o Логические функции (продолжение)

Операция "И" перед "ИЛИ"

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E1.5 U E1.6 O U E1.4 U E1.3 = A3.1		

На выходе A3.1 "1", если по крайней мере одна из операций "И" выполнена (имеет на выводе "1").
На выходе A3.1 "О", если ни одна из операций "И" не выполнена.

Операция "ИЛИ-И-ИЛИ"

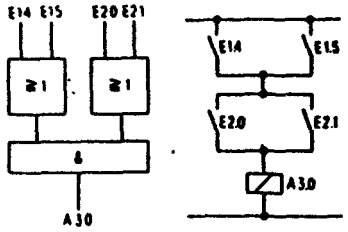
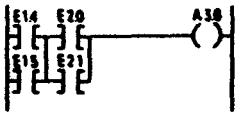
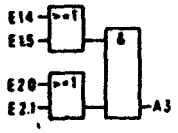
Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	O E6.0 O U E6.1 U(O E6.2 O E6.3) = A2.1		

На выходе A2.1 "1", если на входе E6.0 или на входе E6.1 и одном из входов E6.2 или E6.3 "1".

На выходе A2.1 "О", если на входе E6.0 "О" и операция "И" не выполнена.

о Логические функции (продолжение)

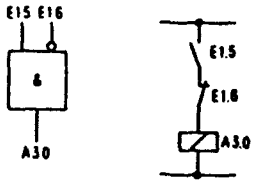
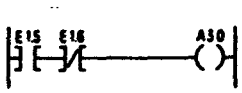
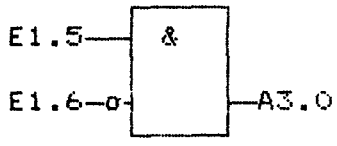
Операция "ИЛИ" перед "И"

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U(O E1.4 O E1.5) U(O E2.0 O E2.1) = A3.0		

На выходе А3.О "1", если обе операции "ИЛИ" выполнены.

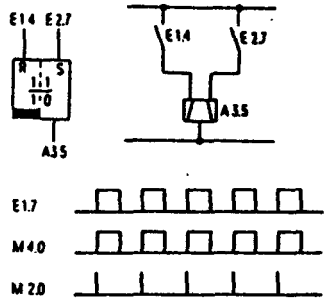
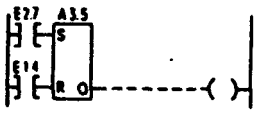
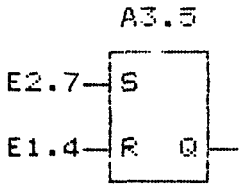
На выводе А3.О "О", если не выполнена хотя бы одна операция "ИЛИ"

Опрос на 0-состояние сигнала

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E1.5 UN E1.6 = A3.0		

На выходе А3.О "1" только в том случае, если на входе Е1.5 "1" (нормально разомкнутый контакт активирован), а на входе Е1.6 "О" (нормально замкнутый контакт не активирован).

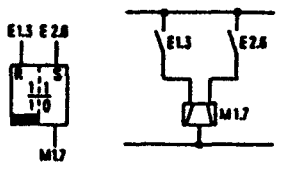
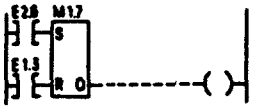
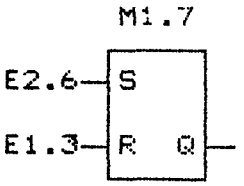
o Функции памяти RS-триггер (элемент памяти) для запоминания сигналов

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E2.7 S A3.5 U E1.4 R A3.5		

Состояние сигнала "1" на входе E2.7 вызывает установку элемента памяти.
Если состояние сигнала на входе E2.7 изменится на "О", то это не повлияет на состояние RS-триггера, т.е. запоминается предыдущее состояние.
Появление "1" на входе E1.4 приводит к сбросу RS-триггера. Если состояние сигнала на входе E1.4 изменится на "О", то это не повлияет на состояние триггера, т.е. запоминается предыдущее состояние.
При одновременном поступлении "1" на оба входа — установки (E2.7) и сброса (E1.4) приоритет у последнего опроса, т.е. в данном случае у сброса (E1.4).
Команда NOP О используется в том случае, если программу потребуется представлять в KOP или FUP на PG675/PG670. При программировании RS—триггера в KOP или FUP эта операция заносится в программу автоматически.

o Функции памяти (продолжение")

RS-элемент памяти с маркером

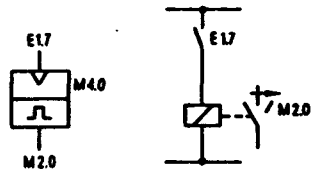
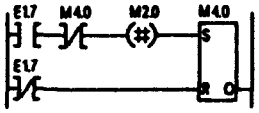
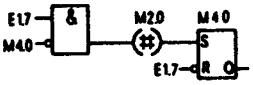
Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E2.6 S M1.7 U E1.3 R M1.7		

Состояние сигнала "1" на входе E2.8 вызывает установку элемента памяти.
Если состояние сигнала на входе E2.6 изменится на "О", то это не повлияет на состояние RS—триггера, т.е. запоминается предыдущее состояние.
Появление "1" на входе E1.3 приводит к сбросу RS—триггера. Если состояние сигнала на входе E1.3 изменится на "О", то это не повлияет на состояние триггера, т.е. запоминается предыдущее состояние.

Если сигнал установки (вход E2.6) и сигнал сброса (вход E1.3) появляются одновременно, имеет приоритет операция, запрограммированная последней (в данном случае U E1.3) - сброс имеет приоритет.

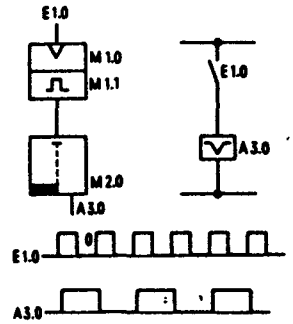
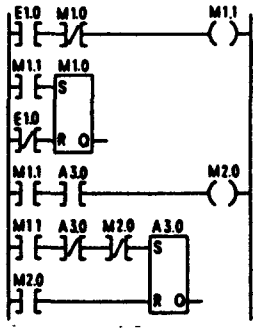
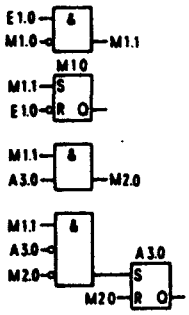
о Функции памяти (продолжение)

Эквивалентная схема импульсного контакта (анализ положительного фронта)

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E1.7 UN M4.0 = M2.0 U M2.0 S M4.0 UN E1.7 R M4.0		

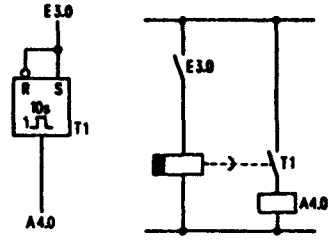
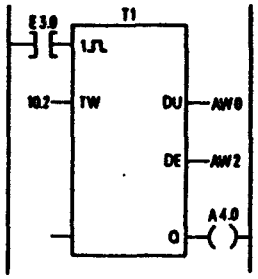
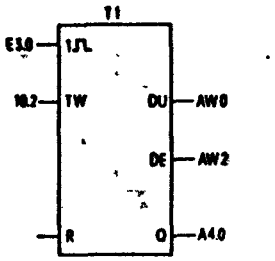
При любом положительном фронте на входе E1.7 выполняется операция "И" (U El.7 и UN M4.0), и результатом логической операции (VKE) устанавливаются маркеры M4.0 (маркер фронта.) и M2.0. При следующем цикле обработки программы операция "И" (U El.7 и UN M4.0) не выполняется, т.к. маркер M4.0 установлен. Маркер M2.0 возвращается в исходное состояние. Таким образом, маркер. M2.0 только во время одного единственного цикла программы имеет состояние "1I".

Делитель частоты на 2 (Т—триггер)

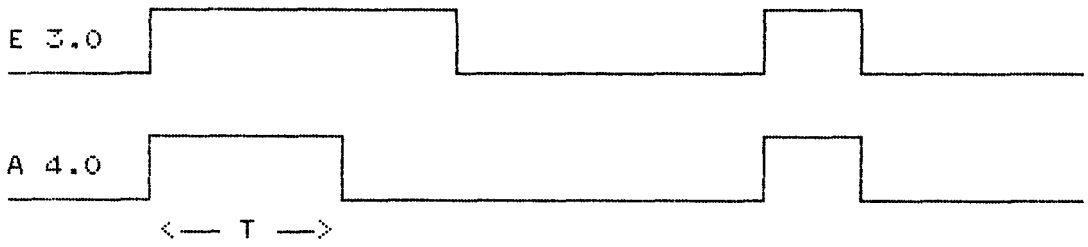
Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E1.0 UN M1.0 = M1.1 U M1.1 S M1.0 UN E1.0 R M1.0 U M1.1 U A3.0 = M2.0 U M1.1 UN A3.0 UN M2.0 S A3.0 U M2.0 R A3.0		

Делитель частоты на 2 (выход A3.0) меняет свое состояние при каждом положительном фронте сигнала на входе E1.0. Поэтому на выходе элемента памяти частота сигнала в два раза меньше, чем на входе.

o Функции таймера Импульс

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E 3.0 L KT10.2 SI T 1 U T 1 = A 4.0		

Таймер запускается во время первого же цикла обработки программы, если результат логической операции (VKE) = 1. Таймер не меняет своего состояния, пока VKE=1, и не истекло запрограммированное время.
Таймер сбрасывается в "О", когда VKE становится равным "О" или истекло время, на которое таймер запрограммирован.



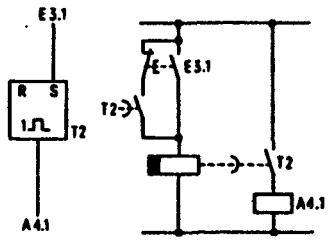
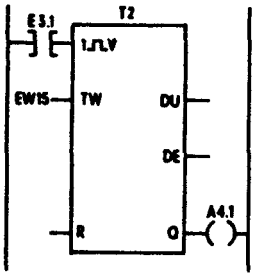
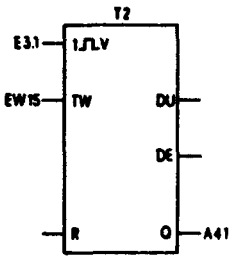
Значение KT 10.2 :
Таймер загружается заданным значением (10) Число справа от точки указывает на масштаб времени:

О = 0.01 сек 2 = 1 сек
1 = 0.1 сек 3 = 10 сек

Выходы DU и DE - цифровые выходы ячейки таймера. С выхода DU выдается состояние таймера в двоичном виде, а DE - в двоично-десятичном виде с базой времени.

о Функции таймера (продолжение)

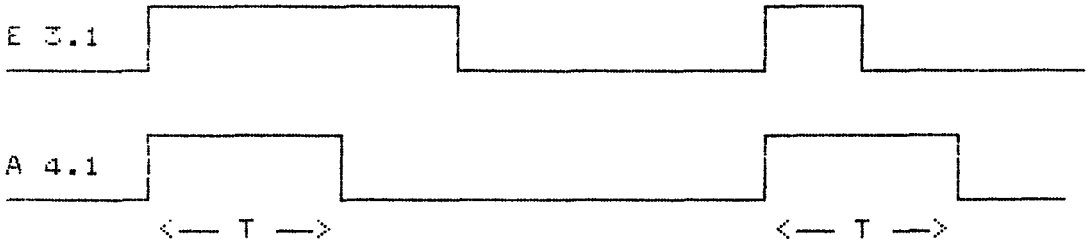
Удлиненный импульс

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E 3.1 L EW15 SV T 2 U T 2 = A 4.1		

Таймер запускается во время первого же цикла обработки программы, если результат логической операции (VKE) = 1.

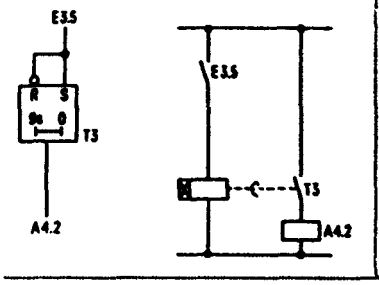
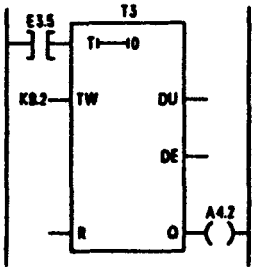
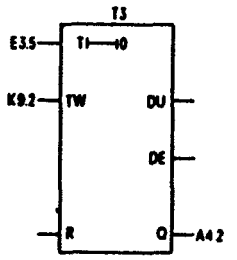
Таймер не меняет своего состояния, пока не истекло запрограммированное время, даже если VKE стал равен "О".

Таймер сбрасывается в "О" по истечении времени, на которое он .запрограммирован.



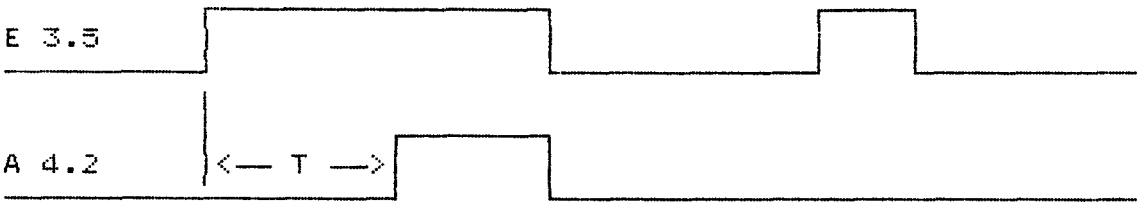
Установка значения величины времени с помощью заданного в двоично-десятичном коде (BCD) значения операндов E, A, M или D (в примере - слово входов 15: EM15).



Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E 3.5 L KT9.2 SE T 3 U T 3 = A4.2		

Таймер запускается во время первого же цикла обработки программы. если результат логической операции (VKE) = 1, но он не устанавливается в "1" до тех пор, пока не истечет время, на которое запрограммирована задержка включения.

Таймер сбрасывается в "О", когда VKE устанавливается в "О".



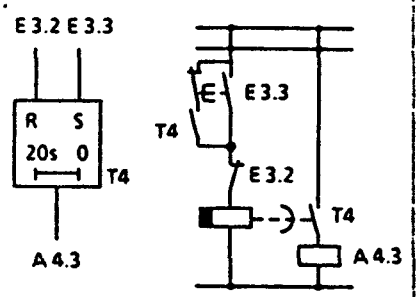
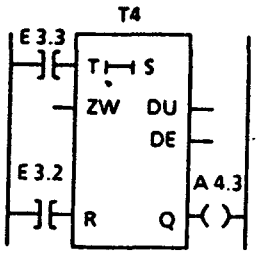
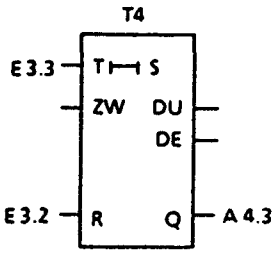
KT 9.2:

Таймер загружается заданным значением (9 сек.). Число справа от точки указывает на базу времени:

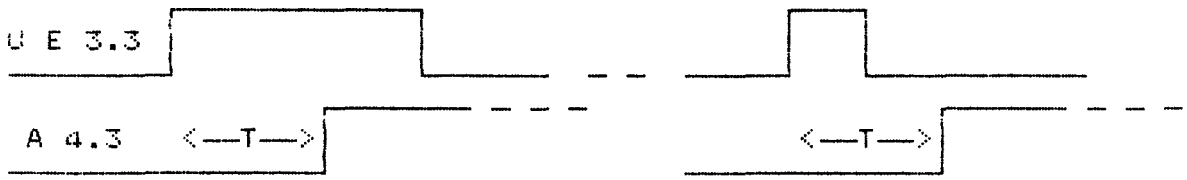
O = 0.01 сек 2 = 1 сек
1 = 0.1 сек 3 = 10 сек

о Функции таймера (продолжение.)

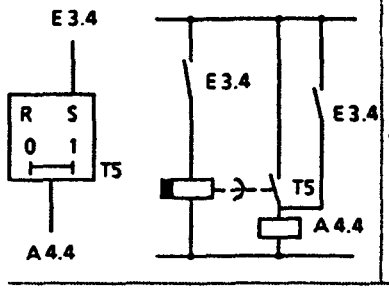
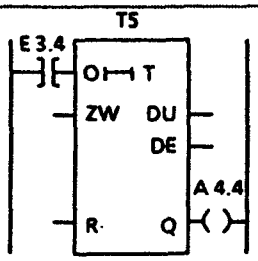
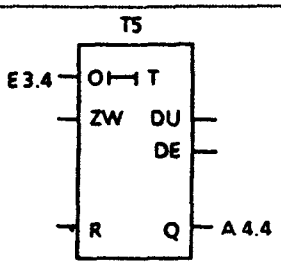
Задержка включения с запоминанием

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E 3.3 L DW21 SS T 4 U E 3.2 R T 4 U T 4 = A 4.3		

По положительному фронту сигнала на входе E3.3 начинается отсчет времени, на которое запрограммирована задержка включения. По истечении этого времени таймер устанавливается в "1". Таймер сбрасывается в "О" только функцией R T.



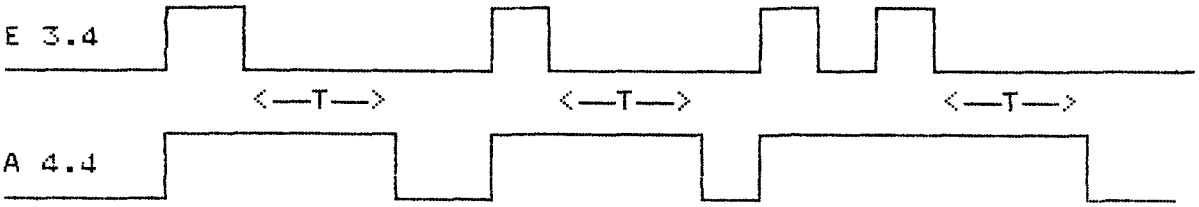
Задержка выключения

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E 3.4 L MW13 SA T 5 U T 5 = A 4.4		

Таймер устанавливается в "1" во время первого же цикла обработки программы положительным фронтом сигнала на входе E3.4.

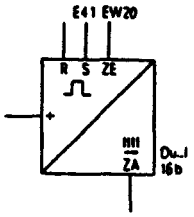
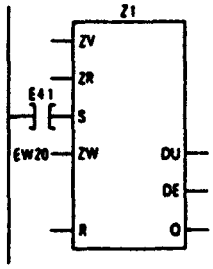
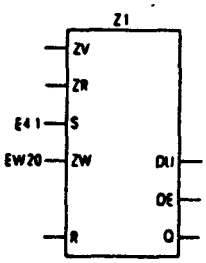
По отрицательному фронту сигнала на входе E3.4 начинается отсчет времени, на которое запрограммирована задержка выключения.

По истечении этого времени таймер сбрасывается в "О".



o Функции счетчика

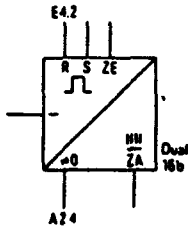
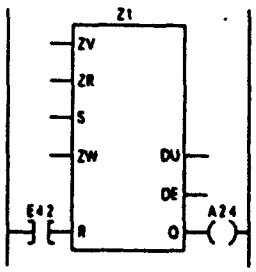
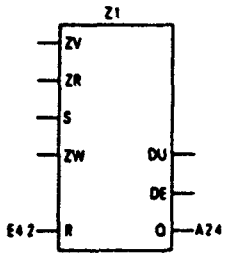
Установка счетчика

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E 4.1 L EW20 S Z 1		

Счетчик устанавливается по входу S во время первого же цикла обработки программы и в течение этого цикла не меняет своего состояния независимо от VKE. Счетчик перезапускается в начале следующего цикла, если к его началу VKE=1.
Указание для работы с положительным фронтом содержится в слове счетчика, которым может быть слово входов EW, слово выводов AW, слово маркеров MW или слово данных DW. В слове счетчика задано также значение счета. Слово счетчика задается в BCD—коде и имеет формат 16 бит.

Выходы DU и DE - цифровые выходы ячейки счетчика. С выхода DU выдается состояние счетчика в двоичном виде, а DE - в двоично-десятичном виде.

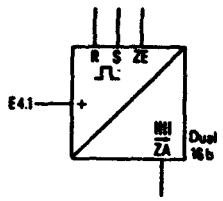
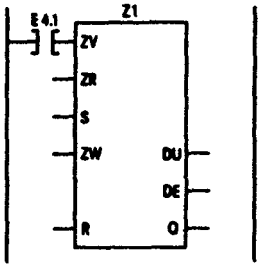
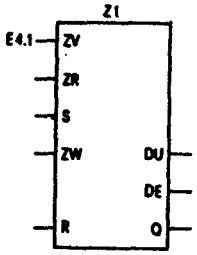
Сброс счетчика

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E 4.2 R Z 1 U Z 1 = A 2.4		

Счетчик сбрасывается по входу R, если VKE = 1 . Счетчик после этого остается без изменений, даже если результат логической операции становится "О" на входе R.

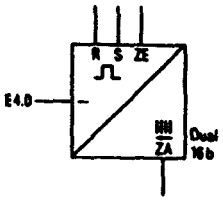
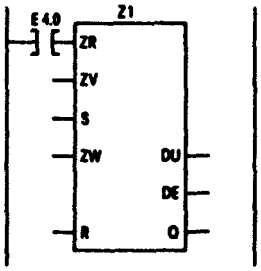
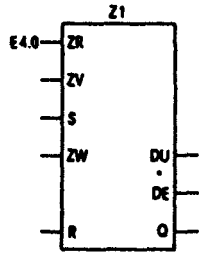
o Функции счетчика (продолжение)

Прямой счет

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E4.1 ZV Z 1		

Значение счетчика увеличивается на 1 при поступлении на счетный вход ZV положительного фронта сигнала от операнда, запрограммированного перед оператором ZV. Необходимые для работы с положительными фронтами указания содержатся в слове счетчика. Используя два различных маркера для работы со счетными входами ZV и ZR, счетчик можно использовать для прямого и обратного счёта.

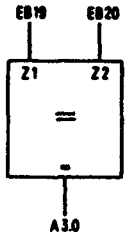
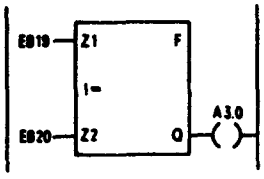
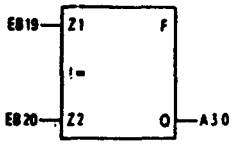
Обратный счет

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	U E4.0 ZR Z 1		

Значение счетчика уменьшается на 1 при поступлении на счетный вход ZR положительного фронта сигнала от операнда, запрограммированного перед оператором ZR. Необходимые для работы с положительными фронтами метки содержатся в слове счетчика. Используя два различных маркера для работы со счетными входами ZV и ZR, счетчик можно использовать для прямого и обратного счета.

о Функции сравнения

Сравнение на равенство

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	L EB19 L EB20 != F = A3.0		

Операнд, заданный первым, сравнивается с последующим операндом в соответствии с функцией сравнения (Z1=Z2). По результату сравнения выставляется VKE:

VKE = 1, если операнды равны (содержимое EB19 равно содержимому EB20)

АКК 1-L = АКК 2-L

VKE = O, если операнды не равны.

АКК 1-L ≠ АКК 2-L

Коды условий ANZ1 и ANZO устанавливаются, как описано в Разделе 6.1.

O	EB19	АКК 2-L
O	EB20	АКК 1-L

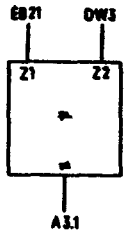
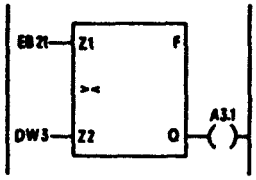
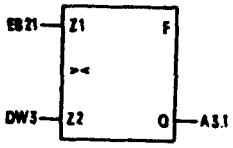
АКК 2—H и АКК 1-H остаются без изменений во время сравнения чисел с фиксированной запятой.

При сравнении чисел с фиксированной запятой (!=F) и чисел с плавающей запятой (!=G) сравнивается полное содержимое аккумуляторов АКК 1 и АКК 2 (32 бита).

При сравнении учитывается численное представление операндов, т.е. содержимое аккумуляторов АКК 1-L и АКК 2-L интерпретируется как числа с фиксированной запятой.

o Функции сравнения (продолжение)

Сравнение на неравенство

Исходная схема	Тип представления STEP 5		
	Список команд (AWL)	Контактный план (KOP)	Функциональный план (FUP)
	L EB21 L DW3 >< F = A3.1		

Операнд, заданный первым, сравнивается с последующим операндом в соответствии с функцией сравнения (Z1XZ2). По результату сравнения выставляется VKE:

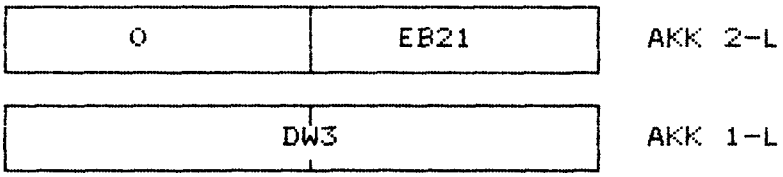
VKE = 1, если операнды не равны (содержимое EB21 не равно содержимому EB22)

АКК 1-L 4 ≠ АКК 2-L

VKE =0. если операнды равны.

АКК 1-L = АКК 2-L

Коды условий ANZ1 и ANZO устанавливаются, как описано в Разделе 6.1.



АКК 2-H и АКК. 1-H остаются без изменений во время 16-битового сравнения чисел с фиксированной запятой.

При сравнении 32-битовых чисел с фиксированной запятой сравнивается полное содержимое аккумуляторов АКК 1 и АКК 2 (включая АКК 1-H и АКК 2-H)

Это применимо и к сравнению на больше, больше или равно, меньше, меньше или равно (смотри список команд).

При сравнении учитывается численное представление операндов, т.е. содержимое аккумуляторов АКК 1-L и АКК 2-L интерпретируется как числа с фиксированной запятой.

4.3 Дополнительный набор операций

В противоположность другим блокам, функциональные блоки могут программироваться с помощью расширенного набора операция языка STEP 5. Полный набор операций для использования в функциональных блоках состоит из основного и расширенного набора операций.

Основной набор, расширенный набор операций и системные функции вместе составляют полный набор операций языка программирования STEP 5.

С помощью системных функций можно вмешиваться в работу системной программы; память может быть перезаписана в любой момент, и может быть изменено содержимое рабочих регистров центрального процессора. Поэтому, системные функции следует использовать с повышенным вниманием.

Системные функции представляются на следующих страницах.

Программа функциональных блоков может быть представлена только в виде AWL. Поэтому они не могут быть запрограммированы в графическом виде представления (KOP и FUP).

Следующее описание показывает дополнительные операции и системные функции, которые могут быть использованы только в функциональных блоках. Кроме того, показана возможная комбинация операций дополнительного набора с актуальными операндами.

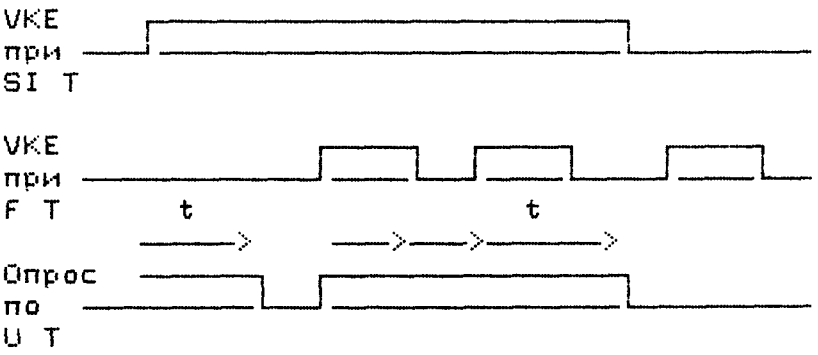
o Двоичная логика

Операция	Описание
U = ____	Функция "И", опрос формального операнда на состояние "1".
UN = ____	Функция "И", опрос формального операнда на состояние "0".
O = ____	Функция "ИЛИ", опрос формального операнда на состояние "1"
ON = ____	Функция "ИЛИ", опрос формального операнда на состояние "0"
	Вставить формальный операнд
	В качестве фактического операнда допустимы двоично-адресованные входы/выходы и маркеры (вид параметра E,A; тип данных:BI), таймеры и счетчики (вид параметра: T, Z).

o Функции запоминания

Операция	Описание
S = ____ RB = ____ = = ____ ^ |	Установка (двоичная) формального операнда. Сброс (двоичный) формального операнда. Вывод результата соединения по формальному операнду.
└────	Вставить формальный операнд
	В качестве фактического операнда допустимы двоично-адресованные входы,выходы,данные и маркеры (вид параметра E,A; тип данных:BI)

o Функции таймеров и счетчиков

Операция	Описание
FR T 0...127	Деблокировка таймера для запуска Операция выполняется только по переднему фронту результата логической операции VKE. Таймер перезапускается, если VKE = "1" во время операции запуска
	
FR Z 0...127	Деблокировка счетчика Операция выполняется только по переднему фронту результата логической операции VKE. Счетчик устанавливается (счет "вперед" или "назад"), если результат логической операции VKE=1 в соответствующей операции
FR = ____	Деблокирование формального операнда для нового пуска (см. "Описание FR T или FR Z" в зависимости от формального операнда, вид параметра: T,Z).
RD = ----	Сброс (цифровой) формального операнда (вид параметра:T,Z).

Операция	Описание
SI = ----	Запуск таймера, заданного формальным операндом, в качестве импульса с длительностью, закодированной в АКК (вид параметра:T).
SE = ----	Запуск таймера, заданного формальным операндом, в качестве задержки включения с длительностью, закодированной в аккумуляторе (вид параметра: T).
SVZ = ----	Запуск таймера, заданного формальным операндом, в качестве удлиненного импульса с длительностью, закодированной в аккумуляторе или установка счетчика, заданного формальным операндом, с указанным впоследствии счетным значением (вид параметра: T,Z).
SSV = ----	Запуск таймера, заданного формальным операнда, в качестве запоминающей задержки включения со значением, закодированным в аккумуляторе, или счет вперед счетчика, заданного формальным операндом (вид параметра: T,Z).
SAR = ----	Запуск таймера, заданного формальным операндом, в качестве задержки включения с длительностью, закодированной в аккумуляторе, или счет назад счетчика, заданного формальным операндом (вид параметра:T,Z).
	Вставить формальный операнд
	Таймеры и счетчики можно использовать в качестве актуальных операндов. Исключения: SI и SE (только таймеры). Значение таймера или счетчика может быть установлено как с помощью основного набора операций, так и с помощью формальных операндов следующим образом: Установка значения таймера или счетчика в виде значения в коде BCD с помощью операндов EW, AW, MW, DW, указанных в качестве формальных операндов (класс параметра E, тип параметра W) или с помощью константы (вид параметра D, тип данных KT, KZ).

Примеры

Вызов функциональ-ного блока	Программа в функцио-нальном блоке	Выполняемая программа
: SPA FB203		
ИМЯ : BEISPIEL	: U =ANNA	U E 10.3
ANNA : E 10.3	: L KT 010.2	L KT 010.2
BERT : T 17	: SSV =BERT	SS T 17
HANS : A 18.4	: U =BERT	U T 17
	: = =HANS	= A 18.4
: SPA FB204		
ИМЯ : BEISPIEL	: U =MAXI	U E 10.5
MAXI : E 10.5	: SSV =DORA	ZV Z 15
IRMA : E 10.6	: U =IRMA	U E 10.6
EVA : E 10.7	: SAR =DORA	ZR Z 15
DORA : Z 15	: U =EVA	U E 10.7
EMMA : M 58.3	: L KZ100	L KZ100
	: SVZ =DORA	S Z 15
	: UN =DORA	UN Z 15
	: = =EMMA	= M 58.3
: SPA FB205		
ИМЯ : BEISPIEL	: U =KURT	U E 10.4
KURT : E 10.4	: L =EGON	L EW 20
CARL : T 18	: SVZ =CARL	SV T 18
EGON : EW20	: U =CARL	U T 18
MAUS : M 100.7	: = =MAUS	= M 100.7

- Операции загрузки и пересылки

Операция	Описание
L = ____	Загрузка формального операнда. Значение, заданное в качестве формального операнда; загружается в аккумулятор (вид параметра: E,A,T,Z; тип данных: BY,W,D).
LC = ____	Кодированная загрузка формального операнда Значение временной или счетной ячейки в качестве формальн. операнда загружается в аккумулятор в BCD-коде (вид параметра:T,Z)
LW = ____	Загрузка набора битов формального операнда Формальный операнд в битовом виде загружается в аккумулятор (вид параметра:D; тип данных: KF, KH, KM, KY, KC, KT, KZ)
LD = ____	Загрузка набора битов формального операнда Формальный операнд в битовом виде загружается в аккумулятор (вид параметра:D; тип данных: KG)
T = ____ (^ — Вставить формальный операнд)	Пересылка в формальный операнд. Содержимое аккумулятора пересылается в формальный операнд (вид параметра: A, E; тип данных: BY, W, D).

В качестве Фактических операндов допускается использовать операнды, соответствующие основным операциям. При LW допускаются данные в двоичном виде (KM) или шестнадцатиричном виде (KH), в виде двоично-десятичного числа (KY), символов ASCII (KC), числа с фиксированной запятой (KF), значение в формате таймера (KT) и счетчика (KZ). При использовании LD в качестве данных допускаются числа с плавающей запятой.

Операция	Параметры	Описание
L BA	0...255	Загрузка слова в аккумулятор АКК 1 из области "интерфейса данных".
L BS	0...255	Загрузка слова в аккумулятор АКК 1 из области "системных данных".
T BA [1])	0...255	Передача слова из аккумулятора АКК 1 в область "интерфейсных данных".
T BS [5)]	0...255	Передача слова из аккумулятора АКК 1 в область "системных данных".
LIR [1)]	0...15	Загрузка регистра (косвенная): значение содержимого слова из памяти[2)], адресованного значением из АКК 1
TIR [1)]	0...15	Передача содержимого регистра (косвенная): в слово памяти[2)] , адресованное значением из АКК 1
TN& [1] >	0...255	Передача блока по байтам: источник- в АКК 2, назначение - в АКК 1 [3)]
TNW [1)]	0...255	Передача блока по словам: источник - в АКК 2, назначение - в АКК 1[3)]

[1)] Системная функция

[2)] Регистры при LIR и TIR (ширина регистра = 16 бит)

Номер регистра	Обозначение регистра
0	АКК 1 — H старшее слово (16 бит) аккумулятора 1
1	АКК 1 — L младшее слово (16 бит) аккумулятора 1
2	АКК 2 — H старшее слово (16 бит) аккумулятора 2
3	АКК 2 — L младшее слово (16 бит) аккумулятора 2
4	-
5	BSP — указатель стека блоков
6	DBA — начальный адрес вызванного блока данных
7, 8	-
9	АКК 3 — H старшее слово (16 бит) аккумулятора 3
10	АКК 3 — L младшее слово (16 бит) аккумулятора 3
11	АКК 4 — H старшее слово (16 бит) аккумулятора 4
12	АКК 4 — L младшее слово (16 бит) аккумулятора 4
13, 14	-
15	SAZ — счетчик адресов

a) Регистры 4,7,8,13,14 не представимы. LIR/TIR на этих номерах регистров обрабатываются как нулевая операция (NOP).
b) Обращение к 8-битовой памяти (для адресов слов памяти >EEOOH):
TIR: старший байт регистра теряется.
LIR: в старший байт регистра записывается FFH.

[3]) Параметр для TNW/TNB указывает длину передаваемой области. Конечный адрес исходной области должен быть предварительно загружен в АКК 2, а конечный адрес области приема в АКК 1. Исходная область и область приема должны располагаться полностью каждая в своей области памяти. При этом различаются следующие области памяти:

	Адреса (16-ричные)
16 — Битовая память пользователя	0000...7FFF
для 16К-модуля	0000...3FFF
16 - Битовая память RAM CPU	AC00...EDFF
8 - Битовая память RAM CPU	EE00...EFFF
8 — Битовая область периферии	F000...FFFF

Смотри также распределение памяти (Рис. 17).

o Арифметические операции

Операция	Описание
ENT	Ввод данных в память арифметики Результат команды ENT загружается в АКК 3 и АКК 4, которые в арифметических операциях используются тоже. АКК 4: = АКК 3 АКК 1: = АКК 1 АКК 3: = АКК 2 АКК 2: = АКК 2

Старое содержимое АКК 4 теряется.
Пример
Должно быть вычислено выражение: (30 + 3 X 4)/6 =7

	АКК 1	АКК 2	АКК 3	АКК 4
Содержимое аккумулятора до выполнения арифметической операции	a	b	c	d
L KF 30	30	a	c	d
L KF 3	3	30	c	d
ENT	3	30	30	c
L KF 4	4	3	30	c
* F	12	30	c	c
+ F	42	c	c	c
L KF 6	6	42	c	c
/ F	7	c	c	c

Операция	Параметры	Описание
ADD BF	-127...+127	Прибавление байта константы (с фиксированной запятой) к содержимому АКК 1 [1]
ADD KF	-32768...+ 32767	Прибавление слова константы (с фиксированной запятой) к содержимому АКК 1 [1]
ТАК		Обмен содержимым АКК 1 и АКК [2]

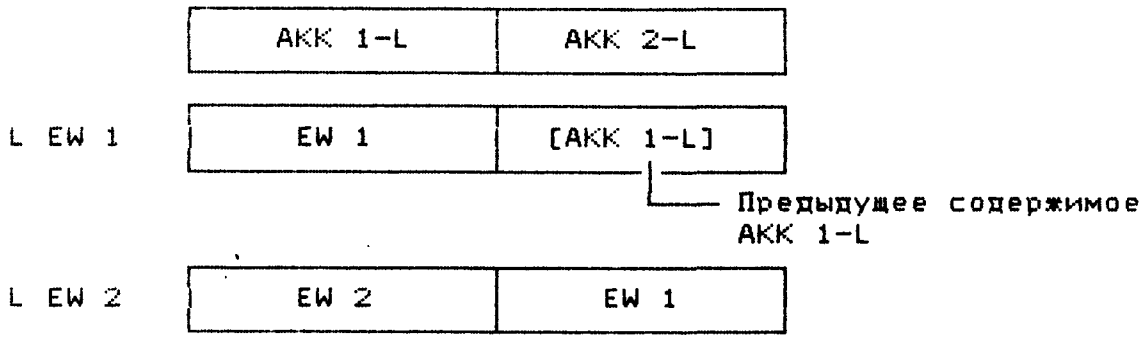
o Цифровые логические операции

Операция	Описание
UW	Операция цифрового логического "И" над содержимым аккумуляторов АКК 1 и АКК 2
OW	Операция цифрового логического "ИЛИ" над содержимым аккумуляторов АКК 1 и АКК 2
XOW	Операция цифрового логического "ИСКЛ.ИЛИ" над содержимым аккумуляторов АКК 1 и АКК 2

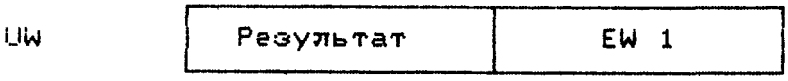
Аккумуляторы 3 и 4 не используются, а используются коды условий ANZ1 и ANZO (смотри Раздел 4.1).

При использовании двум операций загрузки аккумуляторы могут быть загружены соответствующими операндами. Затем содержимое обоих аккумуляторов может быть использовано для цифровых логических операций.

Пример



Операция логического цифрового "И" над EW2 и EW1:



[1] Содержимое АКК 1, 3 и 4 не меняется
[2]) Содержимое АКК 3 и 4 не меняется

Организационные функции

o Функции перехода

Назначение условного или безусловного перехода указывается символически (максимально 4 знака, начинающиеся с буквы). Символический параметр команды перехода идентичен символическому адресу команды, к которой осуществляется переход. При программировании следует принимать во внимание, что абсолютное состояние перевода не может превышать ё127 слов и что команда STEP-5 может состоять из более, чем одного слова. Переход может выполняться только внутри одного блока; переход между сегментами не допустим.

Примечание; Команда перехода и назначение должны быть в одном сегменте. Для сегмента в качестве назначения можно указывать только один символический адрес. Эти условия не применяются для команды перехода SPR, для которой в качестве параметра указывается абсолютный адрес расстояния перехода.

Операция	Описание
SPA = адр	Безусловный переход Безусловный переход выполняется при всех условиях.
SPB = адр	Переход по условию Условный переход выполняется, если VKE=1. Если \VKE=1, команда не выполняется, и результат логической операции VKE устанавливается в 1.
SPZ = адр	Переход по условию : ANZ1, ANZ0 Переход выполняется, если только ANZ1=0 и ANZO=0. Результат логической операции не изменяется.
SPN = адр	Переход по условию : ANZ1, ANZO Переход выполняется, если ANZ1 ≠ ANZO. Результат логической операции не меняется.
SPP = адр	Переход по условию : ANZ1, ANZO Переход выполняется, если только ANZ1=1 и ANZO=0. Результат логической операции не изменяется.
SPM = адр	Переход по условию: ANZ1, ANZO Переход выполняется, если только ANZ = O, и ANZO = 1. Результат логической операции не изменяется.
SPO = адр	Переход по условию переполнения: Переход выполняется, если только условие переполнения OV=1. Если переполнения нет, (OV=0), переход не выполняется. Результат логической операции не изменяется. Если превышены допустимые границы результирующего числа при выполнении арифметических операций, происходит переполнение.

Операция	Описание
SPS = адр	Перепад, если установлен код условия OS (запоминание переполнения) 0S=1.
SPR [1)] -32768...+32767	Переход к системному матобеспечению.

адр = символический адрес (максимально 4 символа)

o Функции сдвига

Операция	Описание
SLW O...15	Сдвиг влево (справа заполнение нулями)
SRW O...15	Сдвиг вправо (слева заполнение нулями)
SLD O...32	Сдвиг двойного слова влево .(справа заполнение нулями)
SVW O...15	Сдвиг вправо со знаком
SVD 0...32	Сдвиг двойного слова вправо со знаком (знак устанавливается слева)
RLD 0...32	Циклический сдвиг влево
RRD 0...32 Параметр	Циклический сдвиг вправо

При операциях сдвига используется только аккумулятор АКК 1. Параметр команды определяет, на сколько позиций производится сдвиг содержимого аккумулятора. В операциях SLW, SRW, SVW используется только младшее слово аккумулятора, в операциях SLD. SVD, RLD, RRD используется полное содержимое аккумулятора (32 бита).

Операции сдвига выполняются безусловно. Последний сдвинутый бит может быть обработан с помощью команд перехода. Происходит установка кодов ANZO и ANZ1 (смотри Раздел 4.1).

Если последний сдвинутый бит равен нулю, используется команда перехода SPZ. Если использовать команду перехода SPN, переход выполняется, если сдвинутый бит был равен 1.

[1)] Системная функция

Пример

Программа STEP-5	Содержимое слова данных
: L DW52	H = 14AF
: SLW 4	
: T DW53	H = 4AF0

Программа STEP-5	Содержимое АКК 1 (16-ричн.)
: L EDO	2348 ABCD
: SLW 4	2348 BCD0
: SRW 4	2348 0BCD
: SLD 4	3480 BCD0
: SVW 4	3480 FBCD
: SVD 4	0348 0FBC
: RLD 4	3480 FBC0
: RRD 4	0348 0FBC
: BE	

o Операции преобразования

Операция	Описание
KEW	Формирование первого дополнения АКК 1 (16 бит)
KZW	Формирование двоичного дополнения АКК 1 (16 бит)
KZD	Формирование двоичного дополнения АКК 1 (32 бит)
DEF	Преобразование числа с фиксированной запятой (16 бит) из кода BCD в двоичный вид
DUF	Преобразование числа с фиксированной запятой (16 бит) из двоичного вида в код BCD
DED	Преобразование числа с фиксированной запятой (32 бит) из кода BCD в двоичный вид
DUD	Преобразование числа с фиксированной запятой (32 бит) из двоичного вида в код BCD
FDG	Преобразование числа с фиксированной запятой (32 бит) в число с плавающей запятой
GFD	Преобразование числа с плавающей запятой (32 бит) в число с фиксированной запятой

Пример

Содержимое слова данных 64 инвертируется и хранится в слове данных 78.

Программа STEP-58	Содержимое слова данных:
: L DW64	BM = 0011111001011011
: KEW	
: T DW78	BM = 1100000110100100

Содержимое слова данных 207 интерпретируется в виде числа с фиксированной запятой и записывается в слово данных 51 с противоположным знаком.

Программа STEP-5		Содержимое слова данных
: L	DW207	F: +51
: KZW		
: T	DW51	F: -51

o Декремент/Инкремент

Операция	Описание
D 1 ... 255	Декремент
I 1 ... 255 Параметр	Инкремент

Содержимое аккумулятора АКК 1 уменьшается (декремент) или увеличивается (инкремент) на число, указанное в качестве параметра. Операция безусловная. Операция ограничивается правым байтом (без переноса).

Пример

Программа STEP-5:		Содержимое слова данных:
: L	DW7	H: 1010
: I 16		
: T	DW8	H: 1020
: D 33		
: T	DW9	H: 10FF

o Функции обработки

Операция	Описание
B DWO ... 255 (операция)	Обработка слова данных Следующая указанная операция будет скомбинирована с параметром, указанным в слове данных и затем выполнена.
B MWO ... 254 (операция)	Обработка слова маркеров Следующая указанная операция будет скомбинирована с параметром, указанным в слове маркеров и затем выполнена.

Операция	Описание
B = _____ (формальный операнд)	Обработка формального операнда (вид параметра B) Могут быть замещены только операции A DB, SPA PB, SPA FB, SPA SB
BI [1] [2]	Обработка, использующая формальный операнд (косвенная). Номер формального операнда находится в аккумуляторе AKK 1.
B BS [1] [2]	Выполнение команды из системной области данных (RS).

Все операции, за исключением перечисленных ниже, могут комбинироваться с командой B DW или B MW;

- A DBO, A DB1, A DB2,
- все двух- или трехслойные команды (допускаются E DB, EX DX, S ES, S EF, AX DX, BA FX и BAB FX)
- операции с формальными операндами в функциональных блоках,
- SPA OBxx и SPB OBxx.

Программатор не проверяет допустимость операции.

Пример (обработка слова данных)

Содержимое слов данных с DW20 по DW100 устанавливается в состояние "O". Индексный регистр для параметров слов данных - DW1.

	KF 20	тановка индексного регистра
	DW 1	
	KF O	рос
	DW 1	
01	DW O	
	DW 1	кремент индексного регистра
	KF 1	
	F	
	DW 1	

	KF 100 F PB =M001 ….	ереход, если индекс в пределах . одолжение программы STEP-5

[1)] Системная функция

[2)] Значение, которое в системных данных или в. формальном операнде в функциональном блоке интерпретируется в качестве кода команды STEP-5, которая затем выполняется. Операция допустима как с командой B DW, так и с командой B MW. Только системные данные 60...63 резервируются для пользователя и не используются системной программой.

o Команды блокировки/деблокировки

Операция	Описание
AS	Блокировка прерывания от процесса
AF	Деблокировка прерывания от процесса

Функция "Блокировка/деблокировка прерываний" может быть использована, например, если во время выполнения программы обработки прерываний по времени необходимо подавить обработку прерываний от процесса. В части программы, заключенной между инструкциями AS и AF, обработка прерываний от процесса невозможна (см. Раздел 3.6.2).

o Специальные функции

Операция	Параметр	Описание
E DB	3...255	Создание блока данных в DB—RAM
	1...255	Создание дополнительного блока данных в области DB-ДАМ
	0....31	Установка семафора
SES [1])	0....31	Сброс семафора
SEF [1])		

— Создание блока данных

Команда E DBxxx создает блок данных с номером xxx (от 3 до 255) в RAM блоков данных CPU. При этом, количество слов данных должно быть загружено в младшее слово аккумулятора АКК 1. Если соответствующий блок данных уже существует или недостаточно места в памяти DB-RAM, то CPU переходит в состояние останова с сообщением об ошибке SFF. Команда EXDXxxx создает дополнительный блок данных и работает как EDBxxx, допустимые параметры с 1 по 255.

— Установка/сброс семафора

Команды SESxxx (установка семафора) и SEFxxx (сброс семафора) регулируют обмен данными между CPU или между CPU и СР *в* многопроцессорном режиме. Установив семафор SESxxx, мы блокируем область данных (маркеры межпроцессорной связи, область ВА) с указанным номером xxx (с О по 31), который должен быть установлен в программе пользователя, для других CPU. Команда SEFxxx снова разрешает другим CPU читать и записывать данные этой области. Семафор может быть сброшен только тем CPU, который его установил. Команды SES/SEF оказывают влияние на биты индикации (смотри Раздел 4.1):

[1]) системные операции

ANZ1	ANZO	Значение
0	0	Семафор установлен другим CPU и не может быть установлен или сброшен
1	0	Семафор установлен/сброшен

Индикация может быть использована функциями условных переходов.

5 Специальные Функции системной программы

Организационные блоки OB40...OB255 зарезервированы для специальных функций и не могут программироваться пользователем, однако, они могут быть вызваны с помощью команд условного или безусловного вызова блока SPA OBxx или SPB OBxx. Если при обработке специальной функции возникает ошибка или если вызванная специальная функция отсутствует, то CPU дает сообщение об ошибке SFF (см Раздел 3.7).

Список имеющихся в данный момент в R-процессоре встроенных специальных функций

OB216...0B218	Обращение к буферам связи
OB220	Преобразование АКК 1 из 16 бит в 32 бита
OB221	Установка и запуск нового времени цикла
OB222	Сброс времени цикла
OB223	"СТОП" при неодинаковом виде запуска в многопроцессорном режиме работы контроллера
OB224	Передача блока маркеров межпроцессорной связи в многопроцессорном режиме
OB226	Считывание содержимого ячеек системной памяти
OB227	Считывание контрольной суммы памяти системных программ
OB230...0B237	Функции управления
OB240	Инициализация регистра сдвига
OB241	Вызов регистра сдвига
OB242	Стирание регистра сдвига
OB250	Инициализация PID—регулятора
OB251	Вызов PID—регулятора
OB254	Копирование блока DX
OB255	Копирование блока DB

5.1 Расширение 16-битового числа с фиксированной запятой до 32 бит (OB220)

Эта специальная функция расширяет 16-битовое число с фиксированной запятой со знаком в АКК 1—L на старшее слово: Если бит 15 = O (положительное число), то в старшее слово загружаются нули, в противном случае (при отрицательном числе) загружается FFFFH. Это например, необходимо для преобразования в число с плавающей запятой (FDG, 32 бит) отрицательного числа с фиксированной запятой.

Параметры: отсутствуют. Случаи ошибок: отсутствуют.

5.2 Новая установка времени цикла (OB 221)

С помощью этой специальной функции пользователь может изменять контрольное время цикла, обычное значение которого 150 мс. Время цикла может меняться от 1 до 4000 мс и должно передаваться в АКК1 в виде числа, равного количеству миллисекунд. Вызвав эту специальную функцию, можно заново запустить контрольный таймер, т.е. установить время цикла. Оно удлиняется, таким образом, на заданное значение времени цикла.
Параметры: Новое время цикла в миллисекундам в АКК 1.

Ошибка: Время цикла не попадает в область 1 мс < ZYKZ < 4000мс

5.3 Переключение контрольного времени цикла (0B 222)

Специальная функция 0B 222 позволяет переключать контрольное. время цикла, т.е перезапускать контрольный таймер. Максимально допустимое значение времени цикла удлиняется, таким образом, с помощью вызова этой специальной функции на заданное значение - стандартное в 150 мс или установленное в 0B 221.
Параметры: Отсутствуют

Ошибки: Отсутствуют

5.4 Чтение ячейки памяти EPROM системной программы (OB 226)

Содержимое ячейки программной памяти (байта) EPROM системной программы загружается в младший байт АКК 1. Остальное содержимое АКК 1 стирается. Старое содержимое АКК 1 копируется в АКК 2. Адрес считываемой ячейки программной памяти должен быть перед вызовом 0B 226 загружен в АКК 1.
Параметры: Адрес загружаемой ячейки программной памяти в АКК 1.

Ошибки: Отсутствуют

5.5 Чтение контрольной суммы EPROM системной программы (OB 227)

Контрольная сумма EPROM системное программы загружается в младшее слово АКК1. Остальное содержимое АКК1 стирается. Старое содержимое АКК1 копируется в АКК2.

Вместе со специальной функцией 0B 226 пользователь может проверить во время циклической обработки программы содержимое EPROM системной программы, в которой он суммирует содержимое отдельных ячеек памяти EPROM с помощью операции сложения с фиксированной запятой и конечную сумму сравнивает с контрольной.

Параметры: Отсутствуют

Ошибки: Отсутствуют

5.6 Регистр сдвига (OB240, 241, 242)

5.6.1 Режим работы

Следующая диаграмма иллюстрирует принцип построения программного сдвигового регистра. Он состоит из ряда 8-битовых позиций хранения в блоке памяти RAM контроллера 135U.

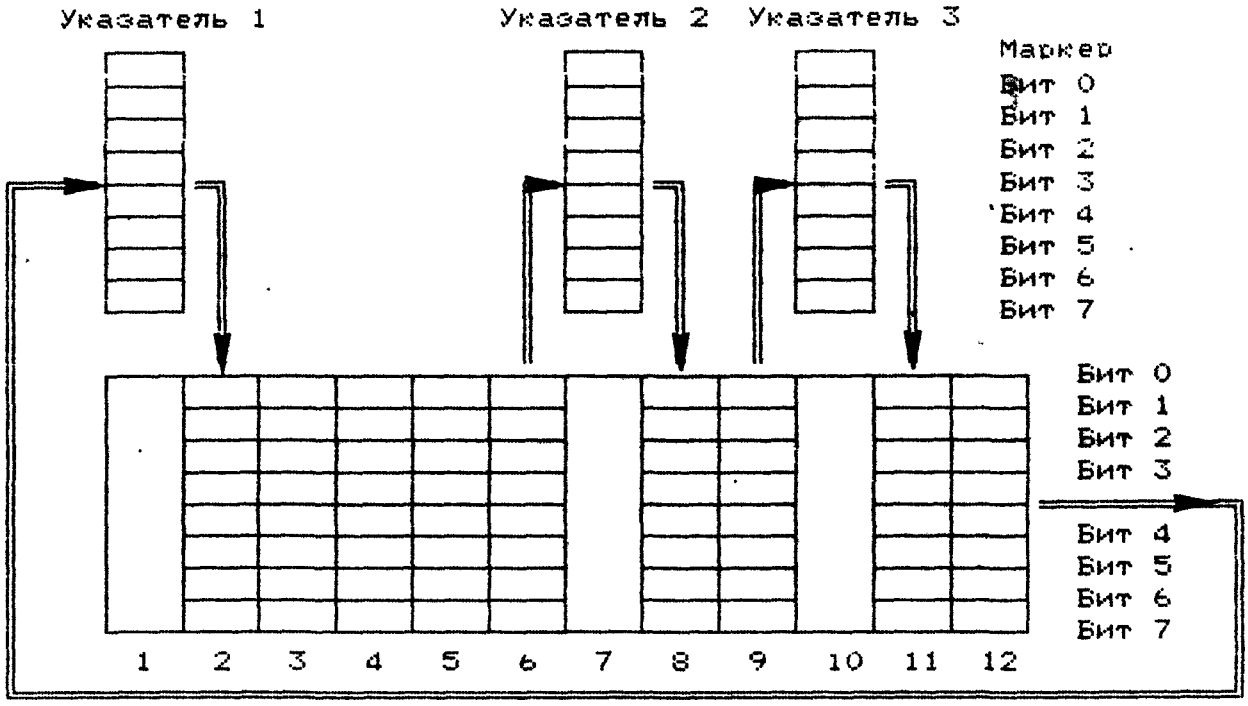


Рис.19 Схематическая диаграмма сдвигового регистра с 3 указателями и 12 ячейками хранения

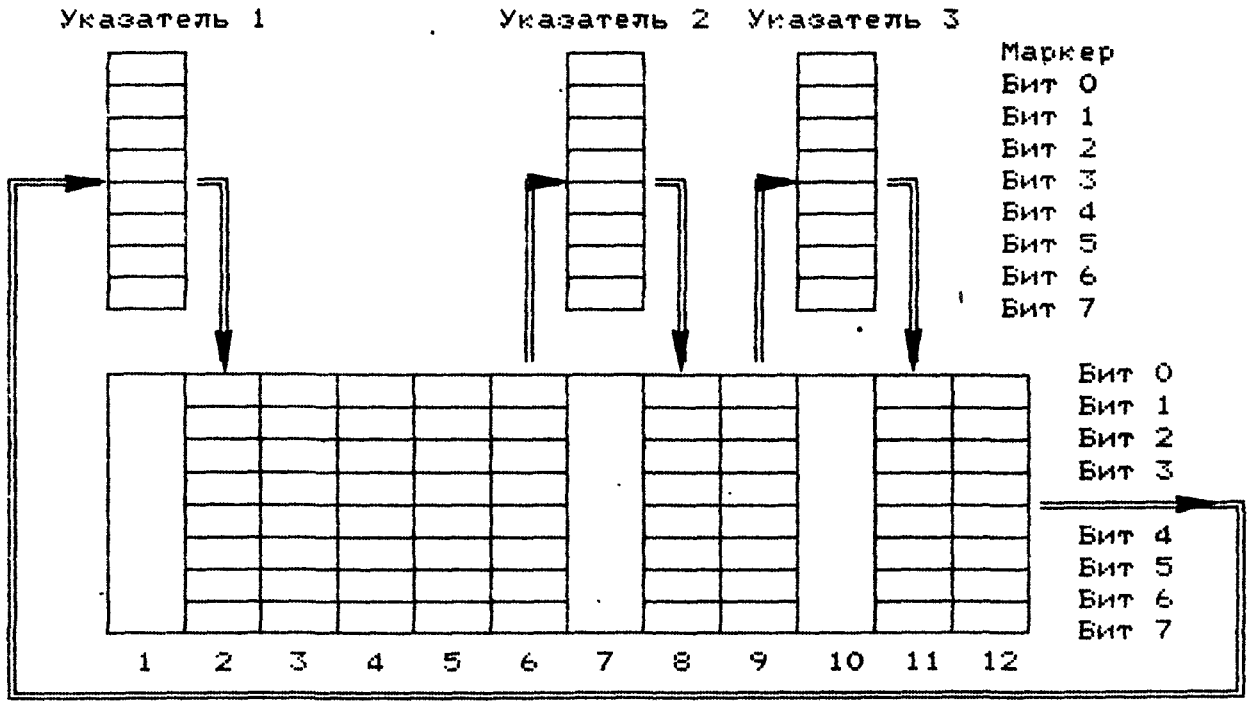


Рис.20 Схематическая диаграмма сдвигового регистра с 3 указателями и 12 ячейками хранения перед первым тактовым импульсом

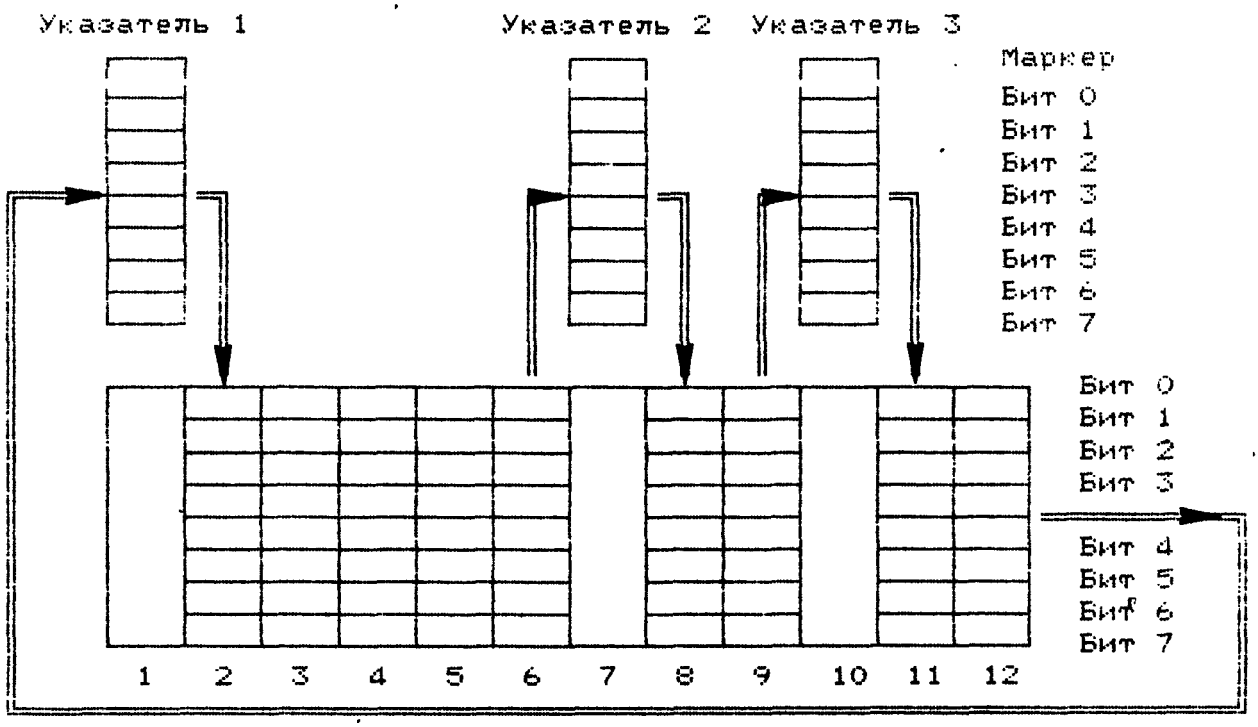


Рис.21 Схематическая диаграмма сдвигового регистра с 3 указателями и 12 ячейками хранения после первого тактового импульса

Количество ячеек хранения может быть выбрано пользователем в пределах от 2 до 256. Отдельные сдвиговые регистры находятся вне памяти RAM в байтам маркеров, которые после их назначения используются в этом качестве,

Установка первого указателя (базовый указатель) фиксировано располагается в первой ячейке. Все остальные указатели могут располагаться пользователем относительно базового указателя произвольно. Количество указателей определяется пользователем от одного (базовый) до шести. Как в случае аппаратно-реализуемых сдвиговых регистров, информация сдвигается побайтно от позиции 1, как показано на рисунке, к позиции L. Отсюда информация возвращается снова в первую позицию. Каждый вызов функции сдвигового регистра заставляет всю информацию сдвигаться на 1 позицию хранения —> 1 тактовый импульс.

Пользователь может вводить информацию в регистр сдвига или выбирать информацию из него. Это возможно только при использовании указателей, т.е. байты маркеров используются в качестве указателей.

— Перед вызовом функции SR: Установить/сбросить биты маркеров (Рис. 20).

Пример:	Установить бит маркера О указателя 1 Установить бит маркера 3 указателя 2 Установить бит маркера 2 указателя 3	STEP-5 S M 0.0 S M 1.3 S M 2.2

— Вызвать функцию сдвигового регистра SPA OB 241

Результат: Информация (8 бит) в ячейке сдвигается на 1 позицию.

— Интерпретация информации, которая теперь находится в указателе L MB О

Пример: Опрос маркерных битов О, 3 и 2 (Рис. 21) в базовом указателе.

Вся информация в базовом указателе, которая приходит от всех других указателей, может быть считана (в приведенном выше примере это возможно только после 12 тактовых импульсов).

5.6.2 Инициализация и вызов регистра сдвига

o Вызов регистра сдвига <OB 241>

В R—процессоре может быть вызвано максимально 64 регистра сдвига. Вызов производится из программы пользователя блоком .OB241. Перед вызовом необходимо занести номер регистра сдвига в аккумулятор АКК 1-L (младшее слово), причем, номер лежит в пределах от 182 до 255.

Перед вызовом регистра сдвига в циклической программе пользователя необходимо каждый регистр сдвига запараметрировать, например при запуске через блок данных с тем же номером. Каждому регистру сдвига в качестве указателя присваивается маркерный байт (максимально 6 указателей = 6 маркерных байтов на один регистр сдвига).

*

o Инициализация регистра сдвига (OB 240)

Перед вызовом регистра сдвига необходимо его сначала один раз — например, при запуске — параметрировать и инициализировать. Для этого надо создать блок с тем же номером, что и инициализируемый регистр сдвига, а затем вызвать специальную функцию 0B 240.

Блок данных должен быть построен по жесткой схеме, которую должен создать пользователь. Он содержит все необходимые для параметрирования регистра сдвига данные.

0	DW 0
Длина регистра сдвига	DW 1
Номер первого байта маркеров	DW 2
Расстояние n_2	DW 3
	. . .
Расстояние n_6	DW 7
0	DW 8

Рис.22 Конфигурация блока данных для инициализации регистра сдвига

Слово данных О: Всегда содержит нули

Слово данных 1: Длина регистра сдвига в виде числа байтов памяти, отведенных под регистр сдвига. Оно может быть в предела от 2 до 256.

Слово данных 2: Содержит номер первого маркера блока маркеров, являющегося указателем. Если пользователь параметризует, например, два указателя, то они вместе с базовым указателем составят три указателя. Тогда они указываются в блоке данных и резервируются два следующих друг за другом маркера. Следует предусмотреть, чтобы до конца маркерной области было достаточно места для всех запараметрированных указателей.

Слова данные 3...7: Максимально 5 вводов, т.е. расстояние от указателя до базового указателя:

n_2 = расстояние от указателя 2 до базового указателя

n_3 = расстояние от указателя 3 до базового указателя

Число указателей, включая базовый, не может быть больше длины, регистра сдвига. Также расстояние от указателя до базового указателя не может быть больше длины регистра сдвига.

Слово данных после последнего расстояния указателя должно содержать нули.

На основании информации из этого блока данных резервируется и инициализируется определенная область памяти в конце памяти RAM блоков данных (DB-RAM). для каждого регистра сдвига требуется n=L/2+8 слов данных, т.е. длина DB—RAM уменьшается на n слов данных, причем конечный адрес DB—RAM сдвигается к младшим адресам.

Если регистр сдвига, который должен быть инициализирован, уже имеется, то, при одинаковой длине, новый и уже имеющийся регистры сдвига занятую область инициализируют заново. В противном случае уже занятая область объявляется недействительной, и открывается новая область.
Параметры: Открываемый блок данных с номером от 192 до 255

Ошибки:

— Недопустимый номер блока данных;
— Имеющейся памяти недостаточно;
— формальная ошибка в конфигурации блока данных;
— Ввод недопустимой длины регистра сдвига;
— Ошибка параметрирования указателей.

5.6.3 Стирание регистра сдвига (ОВ 242)

С помощью этой функции регистр сдвига, чей номер передается в АКК' 2, стирается; т.е. стирается список, и в DB-RAM метка соответствующего регистра объявляется недействительной.

Параметры: Номер стираемого регистра сдвига в АКК 1—L.

Ошибки: Недопустимый номер регистра сдвига.

5.7 PID-регулятор (ОВ 250 и 0В 251)

Пользователь может иметь в процессоре один или более PID-регуляторов в ПК 135U.

Каждый регулятор должен быть инициализирован в стартовом организационном блоке. Для передачи параметров используется данных.

Текущий алгоритм управления встроен в операционную систему и может быть только вызван пользователем. Блок данных используется в качестве интерфейса данных между алгоритмом управл. программой пользователя.

о Функции PID-регулятора

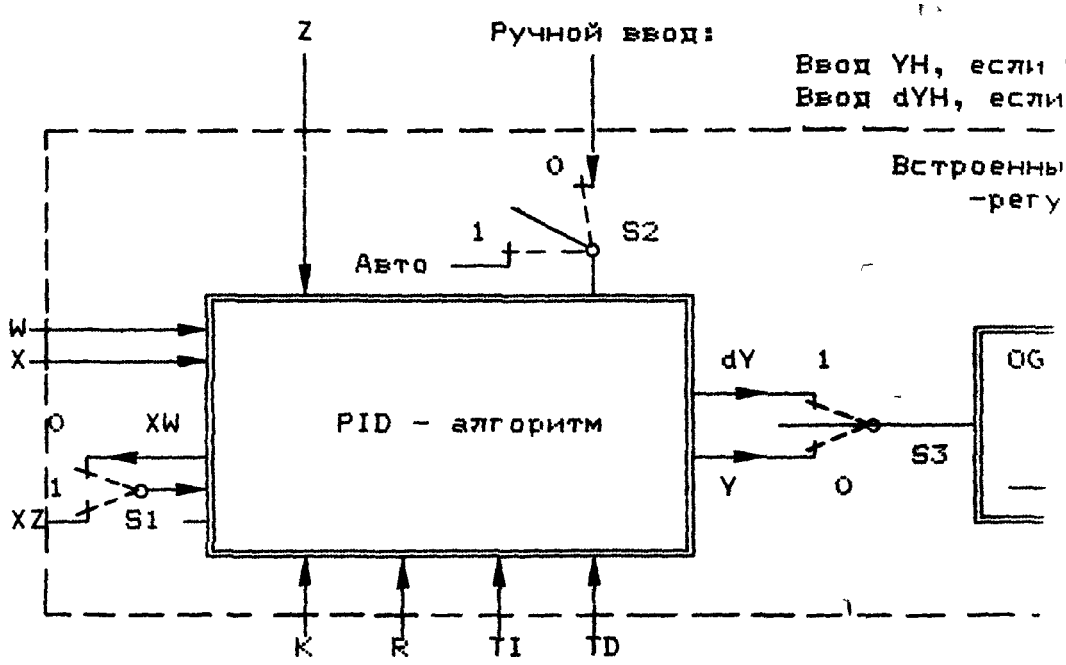


Рис.23 Блок-схема PID—регулятора.

Индекс k: k-опрос

Переключатель	1 Положение	Эффект
SI STEU бит 1	0 1	Дифференциатор выполняет дифференциацию регулировки XW_k Через XZ может быть подан другой сигнал?
S2 STEU бит 0	0 1	Ручное управление Автоматическое управление
S3 STEU бит 3	0 1	Алгоритм управления положением Алгоритм управления скоростью
S4 STEU бит 4	0 1	С прямым управлением Без прямого управления

Функции, относящиеся к положению переключателей на этой блок-схеме, устанавливаются с помощью параметрирования установками соответствующим битов управления в слове управления STEU PID-регулятора (см. Раздел 5.7.2).

Последовательный регулятор предназначен для управления высокоскоростными процессами, например, процессами управления давлением, температурой или потоком.

Сам регулятор основывается на PID-алгоритме (система пропорционально-интегрально-дифференциального регулирования). Выходной сигнал может быть выдан в виде управляемой переменной (алгоритм управления положением) или в виде управляемой скорости изменение переменной (алгоритм управления скоростью).

Отдельные компоненты P, I и D могут быть отключены установкой соответствующие параметров R, TI и TD (см. Раздел 5.7.2) в нулевое значение. Это означает, что легко может быть получена требуемая структура системы регулирования — PI, PID, PD.

Сигнал отклонения XU, (через выход XZ) любая влияющая величина или инвертированное актуальное значение X могут быть использованы в блоке отклонений.

Для компенсации влияния изменений осуществляется управление скоростью подачи .без контроля времени. Сигнал помехи Z, связанный с процессом, подается для алгоритма управления. При ручном управлении в эту точку подается предварительно устанавливаемая переменная YN (см. таблицу значений управляющего слова STEU дальше в тексте).

Если требуется изменение направления, следует устанавливать отрицательное значение параметра K.

Если информация по управлению (dY или Y) на пределе, компонента I выключается автоматически для того, чтобы избежать ухудшения отклика регулирования.

Программа регулирования может содержать фиксированные значения или адаптивные (динамические) параметры (K, R, TI, TD). Они вводятся через установку в памяти отдельных параметров.

5.7.1 PID—алгоритм

PID—регулятор, основан на алгоритме управления скоростью, согласно которому производится вычисление соответствующего управляюоего приращения dY, в отдельной точке во время t=k TA по формуле:

$$dY_k = K[(XW_k - XW_{k-1})R + \frac{TA}{2TN}(XW_k + XW_{k-1}) +$$

$$\frac{1}{2}\frac{TV}{TA}(XU_k - 2XU_{k-1} + XU_{k-2}) + dD_{k-1}\}]$$

$$= K(dPW_kR + dI_k + dD_k)$$

P—часть I—часть D—часть

dXXXk Изменение величины XXX во время t.

U может быть W или Z, в зависимости от того, XW или XZ подключено к отклоняющему входу. Выполняется следующее!

При подключении XW_k При подключении XZ:

$$PW_k = W_k - X_k$$

$$PW_k = XW_k - XW_{k-1} \qquad PZ_k = XZ_k - XZ_{k-1}$$

$$QW_k = PW_k - PW_{k-1} \qquad QZ_k = PZ_k - PZ_{k-1}$$

$$QW_k = XW_k - 2XW_{k-1} + XW_{k-2} \qquad QZ_k = XZ_k - 2XZ_{k-1} + XZ_{k-2}$$

$$dPW_k = (XW_k - XW_{k-1})R$$

$$dI = TI\ XW_k \qquad TI = \frac{TA}{TN}$$

$$dD_k = \frac{1}{2}(TD\ QU_k + dD_{k-1}) \qquad TD = \frac{TV}{TA}$$

Если управляемая переменная Y_K используется в качестве управляющего выхода во время t_k, то она формируется в соответствии с формулой:

$$Y_k = \sum_{m=0}^{m=k} dY_m$$

В большинстве проектируемых процессов управления можно допустить, что R=1, если требуется реакция P.

Величина R может быть использована для установки пропорциональной компоненты PID—регулятора.

5.7.2 Блоки данных для PID—регулятора

Специальные данные для регулятора вводятся с использованием блока данных (для инициализации и вызова PID—регулятора смотри Раздел 7.2.3).

Пользователь должен ввести следующие данные в передаваемый блок данных:

K, R, TI, TD, W, STEU, YH, BGOG, BGUG

Структура передаваемого блока, который состоит из 49 слов данных с номерами от O до 48, более детально описана ниже.

o Структура передаваемого блока данных

Номер слова	Имя	Вх/вых	Формат числа	Формат PG	Примечание
0	—	—	—	—	Зарезервировано
1	K	Вх	GP	KG	Коэффициент пропорциональности K>0 : Положительное направление управления, т.е. изменение уставки и изменяемой переменной в одинаковых направлениях K<0 : Отрицательное направление управления; В пределах числа с плавающей запятой
3	R	вх	GP	KG	R параметр, обычно = 1 для регуляторов с компонентой P; В пределах числа с плавающей запятой
5	TI	вх	GP	KG	TI = TA/TN; В пределах числа с плавающей запятой
7	TD	вх	GP	KG	TD = TV/TA; В пределах числа с плавающей запятой
9	W_k	вх	GP	KG	Здесь вход точки установки, если STEU бит 6 = 1, в противном случае вход в слове No.19 ($-1 < W_k < 1$)
11	STEU	вх	BM	KM	Управляющее слово (смотри страницу)
12	YH_k	вх	GP	KG	Ручной вход здесь, если STEU бит 6 = 1; в противном случае вход в слове No.18 ($-1< YH_k <1$). При алгоритме управления скоростью здесь должен быть указан инкремент управляющей переменной
14	BGOG	вх	GP	KG	Значение верхнего предела $0 < BGOG < 1$ (YA_k макс.)
16	BGUG	вх	GP	KG	Значение нижнего предела $-1 < BGUG < 0$ (YA_k мин.)

Номер слова	Имя	Вх/вых	Формат числа	Формат PG	Примечание
18	YH_k	вх	LP	KF	Здесь ручной ввод, если STEU бит 6= 0 ($-1 < W_k < 1$) При алгоритме управления скоростью здесь должен быть указан инкремент управляющей переменной
19	Wk	вх	LP	KF	Здесь вход точки установки, если STEU бит 6 = 0. ($-1 < W_k < 1$)
20	MERK				Зарезервирован
21	X_k	вх	LP	KF	Ввод актуального значения для STEU бит 7 = 0 ($-1 < X_k < 1$)
22	X_k	вх	GP	KG	Ввод актуального значения для STEU бит 7 = 1 ($-1 < X_k < 1$)
24	Z_k	вх	LP	KF	Воздействующая величина ($-1 < Z_k < 1$)
25	Z_k	вх	GP	KG	Воздействующая величина вводится здесь, если STEU бит7 = 1 ($-1 < Z_k < 1$)
27	Z_{k-1}		GP	KG	Предыдущее значение воздействующей величины
29	XZ_k	вх	LP	KF	Значение, прикладываемое к блоку источника через вход XZ ($-1 < XZ_k < 1$); вход здесь, если STEU бит 7 = 0
30	XZ_k	вх	GP	KG	XZ вводится здесь, если STEU бит 7 = 1 ($-1<XZ_k<1$)
32	XZ_{k-1}		GP	KG	Предыдущее ограничение XZ_k
34	PZ_{k-1}		GP	KG	$XZ_{k-1} - XZ_{k-2}$
36	dD_{k-1}		GP	KG	Исходная компонента
38	XW_{k-1}		GP	KG	Предыдущее значение источника управления
40	PW_{k-1}		GP	KG	$XW_{k-1} - XW_{k-2}$
42	—	—	—	—	Зарезервировано

Номер слова	Имя	Вх/вых	Формат числа	Формат PG	Примечание
44	Y_{k-1}		GP	KG	Предыдущее значение вычисленной управляемой величины Y_{k-1} или dY_{k-1} до ограничения
46	YA_k	вых	GP	KG	Выход величины
48	YA_k	вых	LP	KF	Выход величины BGUG < YA < BGOG

Рекомендуемый формат
(KH и KM тоже допустимы)

GP = Формат числа с плавающей запятой
LP = Двоичная или десятичная дробь

вх — вход, вых — выход

o Значение слова управления STEU (слово 11 в передаваемом DB)

Бит	Имя	Значение
0	AUTO	= 1 : Автоматический режим работы = 0 : Ручной режим работы
1	XZ_ON	= 1 : Дифференциальная величина, которая не должна быть XW_k, подается на блок источника через вход XZ = 0 : XW_k подается на блок источника. XZ игнорируется
2	REG_AUS	= 1 : При вызове регулятора (OB251) все значения в блоке данных регулятора (DW20...DW48) сразу стираются (кроме K, R, TI, TD, BGOG, BGUG, STEU, YH_k и W_k). Регулятор выключается. Предыдущее значение входной величины обновляется = 0 : Регулятор включается
3	VELOC	= 1 : Алгоритм управления скоростью = 0 : Алгоритм управления положением
4[1]	MAN	= 1 : Для GESCHW = 0 (алгоритм управления положением) последняя управляемая выдаваемая переменная остается. Для GESCHW=1 (алгоритм управления скоростью) устанавливается приращение позиционирования $dY_k = 0$. = 0 : Для GESCHW=0, после переключения в ручной режим, значение управляемой переменной YA экспоненциально изменяется к значению, установленному вручную за 4 шага. После этого значение, устанавливаемое вручную, передается непосредственно на выход. Для GESCHW=1, значение, установленное вручную, передается непосредственно на выход регулятора. При ручном режиме работы действуют ограничители, и выполняются следующие соотношения: 1) X_k, XW_{k-1}, PW_{k-1} 2) XZ_k, XZ_{k-1}, PZ_{k-1}, если STEU бит 1 = 1 3) Z_k и Z_{k-1}, если STEU бит 5 = 0 Значение dD_{k-1} устанавливается в "0". Алгоритм не выполняется.
5	NO_Z	= 1 : Без прямого управления = 0 : С прямым управлением
6	PGDG	= 1 : W_k–, YH_k– вводятся в виде чисел с плавающей запятой. = 0 : Ввод в виде двоичной/десятичной дроби.
7	VAR_GP	= 1 : Переменные Xk, XZk, Zk вводятся в виде чисел с плавающей запятой = 0 : Ввод переменной в виде двоичной/десятичной дроби
8	STOS	= 1 : Нет безударного переключения ручной/автомат. = 0 : Безударное переключение ручной/автоматический
9–15		Не имеют значения

[1] Действительно только в ручном режиме (AUTO=0)

5.7.3 Инициализация и вызов PID—регулятора в программе STEP—5

o Инициализация в организационном блоке запуска 0B 20/21/22

—- Выбор передаваемого блока данных DBx (содержащего параметры)
— Вызов OB250 (программа инициализации регулятора)

Для передачи данных каждый регулятор должен использовать свой собственный блок данных DBx (x<255). Системная программа создает другой блок данных DBx+1 в памяти пользователя RAM процессора, который регулятор использует в качестве поля данных во время выполнения циклической программы; поэтому соответствующие номера блоков данных должны быть допустимы к использованию. Эти DBx+1 блоки данных являются интерфейсом данных между регулятором и пользователем или периферией.

Внимание ! Если DBx+1 нельзя использовать во время инициализации, он будет использован без какой-либо индикации от операционной системы в качестве поля данных регулятора; подразумевается, что его длина такая же, как и DB регулятора (48 слов данных), и слова DW20...DW48 стираются. В противном случае контроллер переходит в состояние "СТОП".

Вместо блока данных DB можно использовать дополнительный блок данных DX. Инициализация проводится аналогично.

o Вызов регулятора во время выполнения цикла

Вызов регулятора, после того как время опроса прошло:

— Выбор поля данных DBx+1
— Загрузка входных данных X_k, XZ_k, Z_k, YH_k, или их комбинации
— Входные данные преобразуются в нужный формат и передаются в DBx+1
— Вызов 0B 251 (PID-регулятор)
— Загрузка выходных данных YA_k из DB x + 1
— Обмен данными и передача их процессу входов/выходов

5.7.4 Формат входа и выхода регулятора

Алгоритм PID—регулятора использует формат чисел с плавающей запятой для внутреннего представления чисел и может параметрироваться значениями с плавающей запятой. Кроме того, могут использоваться двоичные и десятичные дроби (смотри бит 6 и 7 в слове управления STEU). В этом случае регулятор автоматически преобразует слова в формат с плавающей запятой.

В языке STEP—5 требуется меньше времени на преобразование слов входов и выходов, если они вводятся в двоичном или десятичном виде.

o Входы

W, YH, X, Z и XZ могут быть введены как в виде чисел с плавающей запятой, так и в двоичном/десятичном виде. Блок передачи данных для каждого значения имеет различные позиции в памяти.

o Ввод в виде двоичной/десятичной дроби

Более детальную информацию по этим числам смотри в Разделе 5.7.7.

Внимание ! Для того, чтобы быть в пределах номинальных пределов модуля аналоговых входов, важно помнить, что значение отдельных битов будет различно в зависимости от того, какие пределы входного сигнала используются. Это следует обязательно учитывать, особенно когда устанавливается прямое управление. В противном случае может возникнуть ситуация, когда прямое управление, введенное с помощью программатора, не может быть выполнено, хотя фактическое значение намного больше, чем требуемое значение.

Если аналого-цифровой преобразователь используется для отрицательных чисел в виде значения со знаком, то из этого значения сначала должно формироваться его двоичное дополнение, перед тем как оно передается в блок данных регулятора. Двоичное значение 15 бита должно быть установлено в "1".

Если при использовании аналого-цифрового преобразователя получается число "О", то оно представляется в виде:

1000000000000000

Затем Формируется двоичное дополнение. Вместо этого числа в блок данных записывается +0:

0000000000000000

o Выходы

Выход регулятора YA формируется в DB в виде числа с фиксированной запятой, расположенной слева. Формат двоичной/десятичной дроби входа и выхода в зависимости от используемого модуля входов/выходов (аналого-цифровой преобразователь, цифро-аналоговый преобразователь) должен быть преобразован до и после вызова регулятора в программе пользователя STEP—5, перед тем как значения передаются в- или из блока данных регулятора.

5.7.5 Общие замечания

Если значение STOS (бит 8 STEU) = О, происходит переключение из ручного в автоматический режим безусловно, т.е. подбираемый любой величины дифференциал регулятора управляется только через составляющую I. Однако, если был выбран TI = TA/TN = О (P- или PD-регулятор), дифференциал регулятора при переключении не вызывает изменения установленной величины.

Это можно исключить, установив STOS = 1. Дифференциал регулятора при этом при переключении ручной/автоматический быстро перерегулируется, независимо от того T 1=0 или нет. Возникающее при этом изменение установленной величины соответствует величине дифференциала регулятора и таким образом не является ошибкой в работе регулятора.

Виты О и 1 MERK могут, если надо, использоваться для индикации, чтобы показать, что установленная величина (при алгоритме скорости установленного приращения) находится на верхней или нижней границе. Так, как эти биты обрабатываются алгоритмом для отключения I-составляющей, то они не могут быть изменены.

Блок данных регулятора DBx+1 не может быть перезагружен во время циклической работы.

Если каскадируется управление с двумя или более регуляторами, следует учитывать следующее:

— Если каскад должен быть разделен на части, то или все регуляторы должны быть переведены в ручное управление одновременно, так, чтобы ни один из регуляторов не изменял компоненту I, или, по крайней мере, регулятор внешней петли должен быть в ручном режиме работы так, чтобы последняя управляемая переменная, которая относится к точке установки внутренней петли была установлена или могла бы быть установлена в безопасное значение.

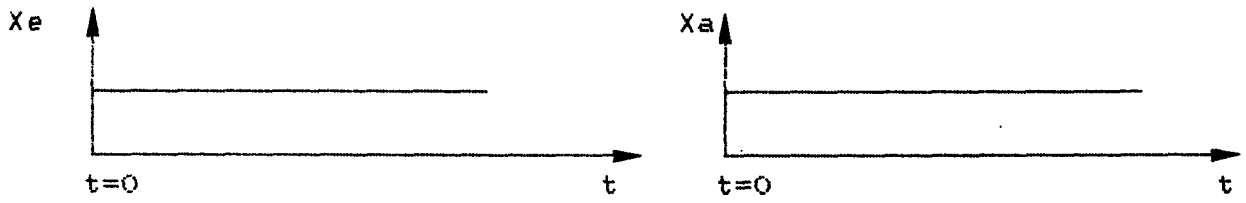
— Если каскад должен быть замкнут, то обе петли должны находиться в автоматическом режиме управления или, по крайней мере, внутренняя петля так, чтобы управляемая переменная внешней петли могла бы быть принята за точку установки.

Если во время переключения в ручной режим работы управляемая система отделяется от регулятора и регулируется непосредственно конечный элемент управления, то результирующая управляемая переменная должна быть подключена к ручному входу регулятора. Этим обеспечивается выдача набора управляемых переменных, соответствующих работе в ручном режиме при переключении из ручного режима работы в автоматический. При использовании алгоритма управления скоростью - это управляемая переменная отклонения.

5.7.6 Характеристики регулятора

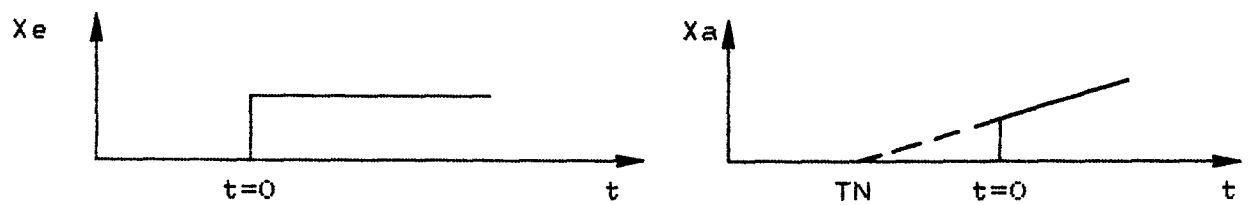
o P—регулятор

Количественная характеристика для P-регулятора — это K. Это отношение входной и выходной величин: K = Xa/Xe



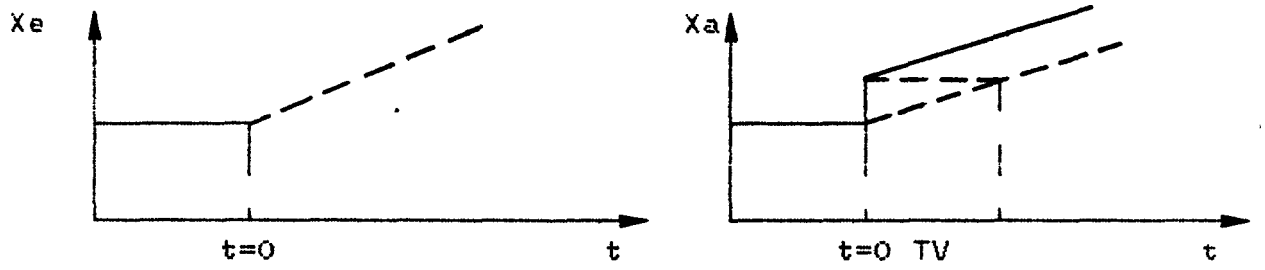
o PI—регулятор

Коэффициент пропорциональности K и интегральная константа времени TN являются количественными характеристиками для PI—регулятора. Коэффициент пропорциональности K — отношение входной и выходной величин определяет реакцию P. Время сброса TN — это время, требуемое для реакции и достижения того же значения регулируемой величины с помощью воздействия I, как и для компоненты P.



o PD—регулятор

Коэффициент пропорциональности K (см выше) и константа времени отклонения TV являются количественными характеристиками PD-регулятора. Время прямого управления — это время, которое необходимо P—регулятору, чтобы произошло такое же изменение на выходе, какое PD—регулятор выполняет в результате наличия компоненты D. Для определения времени подачи линейного изменения входное значение используется вместо шаговой функции.



o PID—регулятор

Количественные характеристики PID—регулятора — это коэффициент пропорциональности K, время сброса TN и время подачи TV. Они определяют P, I и D реакцию.

Сокращения для PID — регулятора

dY_k	Вычисляемое приращение позиционирования
dZ_k	Приращение воздействия
GP	Представление в виде числа с плавающей запятой
k	к-тый опрос
K	Коэффициент пропорциональности
LP	Представление в виде числа с запятой слева (0,123)
UL	Верхний предел
R	R—параметр
TA	Время опроса
TD	TV/TA
TI	TA/TN
t	Точка опроса (время) = к TA
TN	Время сброса
TV	Время подачи вперед
LL	Нижний предел
W_k	Точка установки
X_k:	Актуальное значение
XW_k	Отклонение управляющего воздействия
Y_k	Вычисленная управляемая переменная
YA_k	Значение управляемой переменной (приращение положения или управляемой переменной)
Z_k	Воздействующая величина

5.7.7 Двоичные и десятичные дроби

Для представления двоичных и десятичных дробей в блоке данных требуется одно слово. Связь между десятичным числом, двоичным числом и представлением в виде числа в формате с фиксированной запятой в программаторе показана на следующем примере.

Дробь		Число с фиксированной запятой
десятичная	двоичная	
– 0.999...	1000000000000001	– 32767
– 0.75	1010000000000000	– 24576
– 0.5	1100000000000000	– 16384
– 0.25	1110000000000000	– 8192
0	0000000000000000	0
+ 0.25	0010000000000000	+ 8192
+ 0.5	0100000000000000	+ 16384
+ 0.75	0110000000000000	+ 24576
+ 0.999...	0111111111111111	+ 32767

В двоичном представлении отрицательные числа отличаются от положительных тем, что они образованы двоичным дополнением.

Двоичные/десятичные дроби (LP) могут быть преобразованы в значения, отображаемые программатором в формате KF, по Формуле:

LP x 32767 = KF

если -1 < LP <1 и -32767 < KF < +32767

5.S Функция копирования блоков данных (ОВ 254, ОБ 255)

Специальные функции ОВ 254 и ОВ 255 работают одинаково, причем программа ОВ 254 используется для блоков DX, а ОВ 255 — для DB. В их: обработке- различают два режима:

о Копирование блока данных из памяти пользователя в DB-RAM

Блок данных в памяти пользователя копируется в DB-RAM, сохраняя свой первоначальный номер блока. Новый начальный адрес блока данных заносится в Список адресов DBO, причем старый адрес блока переписывается.

Параметры: АКК 1-L: Номер копируемого блока

АКК 1-H: 0

Ошибки:

— Копируемый блок отсутствует.

— Блок в DB-RAM уже имеется (функция выполняется только один раз, обычно при запуске.)

— Недостаточно места в памяти DB—RAM.

Если возникает одна из этих ошибок, то функция не выполняется, и выдается сообщение об ошибке SFF (смотри Раздел 3.7).

o Создание нового блока данных в DB-RAM

Первоначальный блок данных может находиться и в памяти пользователя и в DB-RAM, если копировать его в DB-RAM под другим номером. Начальный адрес нового блока данных заносится в список адресов *a*-DBO. Начальный адрес старого блока также сохраняется в DBO.

Ввод начального адреса в DBO происходит в том случае, если передача полностью завершена и все метки в заголовке блока указаны правильно. Скопированный блок, таким образом, сначала опознается операционной системой после полной передачи как действительный или существующий.

Параметры: АКК 1-L:	Номер копируемого блока данных
АКК 1-H:	Номер нового блока данных

Ошибки:

— Копируемый блок отсутствует.
— Новый блок уже имеется.
— Недостаточно памяти в DB--RAM.

5.9 Блок передачи маркеров межпроцессорной связи

Обычно, маркеры межпроцессорной связи, указанные пользователем в DB1, побайтно передаются системной программой в CPU при многопроцессорном режиме (смотри Раздел 2.3.1 и Рис.14). При многопроцессорном режиме работы эта передача выполняется каждым CPU независимо через координатор с целью минимизировать время блокирования шины процессором. Поэтому в качестве когерентных блоков информации могут передаваться байты.

Вызов 0B 224 используется при начальном запуске в любом режиме запуска, и при включении каждого CPU пользователь может передавать в блоки все маркеры межпроцессорной связи, указанные в DB1. Каждый CPU может начать передачу своих маркеров межпроцессорной связи только тогда, когда другой CPU завершит свою передачу. Выполнение циклической программы будет соответственно задержано (время цикла!). С помощью этой специальной функции обновление маркеров межпроцессорной связи будет отдельным для каждого CPU, чтобы можно было бы наверняка быть уверенным в новом состоянии информации всех маркеров межпроцессорной связи. Функция будет работать, только если координатор находится в состоянии "RUN" с деблокировкой шины < 8 мкс (смотри Руководство по координатору KOR).

5.10 Сравнение режимов запуска

Вызвав 0B 233, например, при запуске или в начале циклической обработки программы, можно в многопроцессорном режиме проверить дли соответствующего CPU, одинаковы ли виды запуска у всех участвующих в работе CPU. Если это не так, то соответствующий CPU переходит в "СТОП" с сообщением об ошибке SFF (смотри Раздел 3.7).

5.11 Обращение к буферам связи (OB 216…OB218)

Эти организационные блоки позволяют общаться с так называемыми буферами связи. Буфер связи — это область памяти, которая используется при необходимости коммуникационными процессорами некоторыми модулями интеллигентной периферии и некоторыми координаторами в многопроцессорном режиме один раз или много. При размере буфера связи в 1024 байта они занимают место в памяти с F400H по F7FFH.

При размере буфера связи в 2048 байт они занимают область по FBFFH.

Выбор (адресация текущего буфера связи производится через регистр выбора или идентификации (аналогично CHIP — выбор микросхемы).

Организационные блоки содержат следующие Функции:

08216 запись байта/слова/двойного слова в буфер связи;
08217 чтение байта/слова/двойного слова из буфера связи;
08218 заполнение буфера связи от CPU (служит для рования в многопроцессорном режиме)

Эти функции служат, с одной стороны, целям тестирования и одновременно эти элементарные функции позволяют программировать блоки управления или аналогичные блоки.

Способ записи:

АКК 1:	АКК 1	32 байта
АКК 1-L:	Младшее слово аккумулятора АКК 1	16 байта
АКК 1-LL:	Младшее слово, младший байт АКК 1	8 байтов
АКК 1-LH:	Младшее слово, старший байт АКК 1	8 байтов

и т.д.

5.11.1 Запись данных в буфер связи (OB 216)

Блок передает байт/слово/двойное слово из АКК 1 (начального бита) в буфер связи. Адрес назначения в буфере. устанавливается в АКК 2-L, а номер буфера связи в АКК3-LH содержит О, 1 или 2 в качестве указания, что байт, слово или двойное слово соответственно. Номер • может принимать значение от О до 25?', допустимы а 2-L.) от О до 2047.

Адресация буфера связи и передача данных <1/2/4 байта представляют собой неделимое единое действие.

Если передача невозможна из-за того, что:

— АКК 3—LH содержит недопустимое значение;
— указанный адрес недопустим или в этом буфере с адреса не существует;
— указанный бу4'ер связи отсутствует или
— буферов связи вообще нет,

то содержимое аккумуляторов остается неизменным;) индикации в логическом устройстве (например, VKE') CTI

Если передача выполнена успешно, то содержимое АКК 1 и АКК. 3 остается неизменным; АКК 2-L содержит значение, возросшее на 1/2/4 (в зависимости от длины данных); устанавливается VKE; остальные биты индикаций в логическом устройстве стираются.

Пример (без оптимизации по времени)

В буфере связи 7 стирается область с адресами с 50H по 69H (=20 байт) (область существует):

```
        L   KY   1,7          ; 1=передача слова, 7=номер буфера связи
        L   KB   50           ; Начальных адрес
        ENT                   ; Описание АКК 3
        L   KB   0            ; Описание АКК 3 и АКК 1
MARK:   SPA OB 216            ; Слово стирается, адрес повышается на 2
        + F                   ; Можно использовать ТАК
        L   KB   70           ; Конечное значение счетчика циклов
        >< F                  ; Адрес = конечный адрес ?
        SLD   16              ; АКК 1-L стирается, АКК 2-L содержит адрес
        SPB   =MARK           ; Переход, если адрес < конечный адрес
              .               ; продолжение
              .               ; по завершении цикла
```

Подобный пример, только с проверкой существования области:

```
                              ; Начало инициализации
        L KY 1,7              ; 1=передача слова, 7=номер буфера связи
        L KB 50               ; Начальный адрес
        ENT                   ; Описание АКК 3
        L KB 0                ; Описание АКК 2, АКК 1 неопределен
                              ; Конец инициализации
                              ; Начало цикла
SCHL:   TAK                   ; Можно также +F
        L KB 70               ; Конечное значение счетчика циклов
        i= F                  ; Адрес = конечный адрес ?
        SPB =ENDE             ; Если да, выйти из цикла
        SLD 16                ; АКК 1-L стирается, АКК 2-L содержит адрес
        SPA OB216             ; Слово стирается, адрес увеличивается на 2
        SPB =SCHL             ; Переход, если передача без ошибки
                              ; Конец цикла

                              ; Начало обработки ошибок
        FEHL: .               ;
              .               ;
              BEA             ; Конец обработки ошибок

                              ; Начало продолжения
        ENDE:  .              ;
               .              ;
               .              ;
               BE             ; Конец продолжения
```

5.11.2 Считывание данных из буфера связи (OB 217)

Блок передает байт/слово/двойное слово из буфера связи в АКК .1. (начиная с правого бита). Адрес назначения в буфере связи устанавливается в АКК 2-L, а номер буфера связи в АКК 3-LL. АКК 3-LH содержит О, 1 или 2 в качестве указания того, что передается байт, слово или двойное слово, соответственно. Номер буфера связи может принимать значение от О до 255, допустимы адреса (в АКК 2-L) от О до 2047.

Адресация буфера связи и передача данных (1/2/4 байта представляют собой неделимое единое действие.

Если передача невозможна из-за того, что;

— АКК 3-LH содержит недопустимое значение;
— указанный адрес недопустим или в этом буфере связи такого адреса не существует;
— указанный буфер связи отсутствует или
— буфера связи вообще нет,

то содержимое аккумуляторов остается неизменным; все (1) биты индикации в логическом устройстве (например, VKE) стираются.

Если передача выполнена успешно, то содержимое АКК 3 остается неизменным; АКК 2-L содержит значение, возросшее на 1/2/4 (в зависимости от длины данных); АКК 1 в правой части содержит считанное значение (остальные биты равны нулю); устанавливается VKE; остальные биты индикации в логическом устройстве .стираются.

Пример

Блок данных с адресами с 100 по 107 (=8 байт) из буфера связи 7 передается в область маркеров с 200 по 207:

```
   L KY 2,7          ; 2=передача двойного слова, 7=номер буфера
                       связи
   L KB 100          ; Начальный адрес в буфере связи
   ENT               ; Описание АКК 3
   L KB 0            ; Описание АКК 2, АКК 1 не определен
   SPA 0B 217        ; Считываются байты с 0 по 3,
                     ; адрес увеличивается на 4
   T MD 200          ; ПВ200...MB203
   SPA OB 217        ; Считываются байты с 4 по 7,
                     ; адрес увеличивается на 4
   T MD 204          ; MB204...MB207
```

5.11.3 Заполнение буфера связи (OB218)

Блок передает метку установки данного процессора в буфер связи, если содержимое адресуемой ячейки равно нулю. Адрес назначения в буфере связи определяется через АКК 1-L, а номер буфера связи в АКК 2-LL. Номер буфера связи может принимать значение от О до 255, допустимы адреса (в АКК 1-L) от О до 2047.

Адресация буфера связи, чтение и возможная запись метки места установки модуля представляют собой неделимое единое действие.

Если ввод метки места установки невозможен из-за того, что:

— указанный адрес имеет значение, не равное нулю;
— указанный адрес недопустим или не существует в данном буфере связи;
— указанный буфер связи отсутствует или
— буфера связи вообще нет,

то содержимое аккумуляторов остается неизменным; все (!) биты индикации в логическом устройстве (например, VKE) стираются.

Если передача выполнена успешно, то содержимое аккумуляторов остается неизменным; устанавливается VKE; остальные биты индикации в логическом устройстве стираются.

5.12 Вспомогательные функции операционной системы (OB230...OB237)

Специальные функции OB 230...0B 237 содержат вспомогательные функции. Их вызов недопустим и приводит к ошибке.

6. Обзор операций языка STEP-5

Основные операции

Операция	Параметр
o Двоичные логические операции	
U E	0.0 ... 127.7
U A	0.0 ... 127.7
U M	0.0 ... 255.7
U D ¹)	0.0 ... 255.15
U T	0 ... 127
U Z	0 ... 127
UN E	0.0 ... 127.7
UN A	0.0 ... 127.7
UN M	0.0 ... 255.7
UN D ¹)	0.0 ... 255.15
UN T	0 ... 127
UN Z	0 ... 127
O E	0.0 ... 127.7
O A	0.0 ... 127.7
O M	0.0 ... 255.7
O D ¹)	0.0 ... 255.15
O T	0 ... 127
O Z	0 ... 127
ON E	0.0 ... 127.7
ON A	0.0 ... 127.7
ON M	0.0 ... 255.7
ON D ¹)	0.0 ... 255.15
ON T	0 ... 127
ON Z	0 ... 127
)	
U(	
O(	
O	
o Функции сравнения	
!=F	
><F	
>F	
>=F	
<F	
<=F	
!=D	
><D	
>D	
>=D	
<D	
<=D	
!=G	
><G	
>G	
>=G	
<G	
<=G	

Операция	Параметр
o Операции с памятью	
S E	0.0 ... 127.7
S A	0.0 ... 127.7
S M	0.0 ... 255.7
S D ¹)	0.0 ... 255.15
R E	0.0 ... 127.7
R A	0.0 ... 127.7
R M	0.0 ... 255.7
R D ¹)	0.0 ... 255.15
= E	0.0 ... 127.7
= A	0.0 ... 127.7
= M	0.0 ... 255.7
= D ¹)	0.0 ... 255.15
o Функции загрузки	
L EB	0 ... 127
L EW	0 ... 126
L ED	0 ... 124
L AB	0 ... 127
L AW	0 ... 126
L AD	0 ... 124
L MB	0 ... 255
L MW	0 ... 254
L MD	0 ... 252
L DL	0 ... 255
L DR	0 ... 255
L DW	0 ... 255
L DD	0 ... 254
L T	0 ... 127
L Z	0 ... 127
L PB	0 ... 127
	128 ... 255
L PW	0 ... 126
	128 ... 254
L QB	0 ... 255
L QW	0 ... 254
LD T	0 ... 127
LD Z	0 ... 127
L KB	0 ... 255
L KC	2 алфав.-цифр. символа
L KM	Набор 16 бит
L KH	0 ... FFFF
L KF	− 32768 ... ... + 32767
L KY	0 ... 255 для каждого байта
L KT	0.0...999.3
L KZ	0...999
L KG	²)

[1)] Слово 1: B2 + бит адреса; $0.1469368 \times 10^{-39}$…

B3 + относительный адрес $0.1701412 \times 10^{-39}$

Операция	Параметр
o Операции с таймерами и счетчиками	
SI T	0 ... 127
SV T	0 ... 127
SE T	0 ... 127
SS T	0 ... 127
SA T	0 ... 127
R T	0 ... 127
S Z	0 ... 127
R Z	0 ... 127
ZV Z	0 ... 127
ZR Z	0 ... 127
o Функции передачи	
T EB	0 ... 127
T EW	0 ... 126
T ED	0 ... 124
T AB	0 ... 127
T AW	0 ... 126
T AD	0 ... 124
T MB	0 ... 255
T MW	0 ... 254
T MD	0 ... 252
T DR	0 ... 255
T DL	0 ... 255
T DW	0 ... 255
T DD	0 ... 254
T PB	0 ... 127
	128 ... 255
T PW	0 ... 126
	128 ... 254
T QB	0 ... 255
T QW	0 ... 254
o Вызов блоков	
SPA PB	0 ... 255
SPA FB	0 ... 255
SPA SB	0 ... 255
SPB PB	0 ... 255
SPB FB	0 ... 255
SPB SB	0 ... 255
A DB	2 ... 255
BE	
BEB	
BEA	
SPA OB	1 ... 39
SPB OB	1 ... 39
AX DX	1 ... 255
BA FX	0 ... 255
BAB FX	0 ... 255
SPA OB [1]	40 ... 255
SPB OB [1]	40 ... 255

[1] Вызов спецфункции

Операция	Параметр
o Арифметические операции	
+F	
-F	
xF	
:F	
+G	
-G	
xG	
:G	
o Специальные функции	
NOP 0	
NOP 1	
STP	
BLD	

Дополнительные функции

Операция	Параметр
o Логические функции с словами	
UW	
OW	
XOW	
o Операции с таймерами и счетчиками	
FR T	0 ... 127
FR Z	0 ... 127
FR =	Форм.операнд
SI =	Форм.операнд
SE =	Форм.операнд
SVZ =	Форм.операнд
SSV =	Форм.операнд
SAR =	Форм.операнд
RD =	Форм.операнд
o Функции загрузки и передачи	
L =	Форм.операнд
LC =	Форм.операнд
LW =	Форм.операнд
LD =	Форм.операнд
T =	Форм.операнд
L BS	0 ... 255
L BA	0 ... 255
T BA [2]	0 ... 255
T BS [2]	0 ... 255
LIR [2]	0 ... 15

[2] Системная операция

Операция	Параметр
o Функция загрузки и пересылки (продолжение)	
TIR [1]	0 ... 15
TNB [1]	0 ... 255
TNW [1]	0 ... 255
o Двоичные логические операции	
U =	Форм.операнд
UN =	Форм.операнд
O =	Форм.операнд
ON =	Форм.операнд
o Функции преобразования	
KEW	
KZW	
KZD	
DEF	
DUF	
DED	
DUD	
FDG	
GFD	
o Функции сдвига	
SLW	0 ... 15
SRW	0 ... 15
SLD	0 ... 32
SVD	0 ... 32
RLD	0 ... 32
RRD	0 ... 32
SVW	0 ... 15
o Функции перехода	
SPA =	Символич.адрес
SPB =	Символич.адрес
SPZ =	Символич.адрес
SPN =	Символич.адрес
SPP =	Символич.адрес
SPM =	Символич.адрес
SPO =	Символич.адрес
SPS [2] =	Символич.адрес
SPR [1] =	-32768...+32767

Операция	Параметр
o Операции установки	
S =	Форм.операнд
RB =	Форм.операнд
= =	Форм.операнд
o Другие функции	
AF	
AS	
ENT	
D	0 ... 255
I	0 ... 255
B =	Форм.операнд
B DW	0 ... 255
B MW	0 ... 255
BI [1]	
B BS [1]	0 ... 255
TAK	
BLD	0 ... 255
o Арифметические операции	
ADD BF	-127 ... +127
ADD KF	-32768 ... +32767
o Специальные функции	
E DB	0 ... 255
SES 1)	0 ... 31
SEF 1)	0 ... 31

[1] Системная операция

[2] Слово 1: Дистанция перехода (2 байта)

7 Информация об ошибках

7.1 Информация об ошибках в системных байтах 3 и 4

o Информация об ошибках при создании списка адресов блоков

Метка ошибки SD3 SD4	Пояснения	Реакция при запуске
8001H yyyyH	Неверная длина блока yyyy=адрес блока с неверной длиной	Прерывание
8002H yyyyH	Вычисленный конечный адрес блока в памяти неверен yyyy = адрес блока	Прерывание
8003H yyyyH	Недействительный адрес блока yyyy = адрес неверной метки	Прерывание
8004H yyyyH	Слишком большой номер организационного блока. Допустимо OB1..0B47 yyyy = адрес неверной метки	Прерывание

o Информация об ошибках при создании списка адресов для обновления области отображения информации

Метка ошибки SD3 SD4	Пояснения	Реакция при запуске
0400H –	Ошибки не обнаружено	Дальше
0410H yyyyH	Недействительная метка yyyy = недействительная метка	Прерывание
0411H yyyyH	Неверный параметр в списке адресов цифровых входов yyyy = адрес неверно введенного байта входов	Прерывание
0412H yyyyH	Неверный параметр в списке адресов цифровых выходов yyyy = адрес неверно введенного байта выходов	Прерывание
0413H yyyyH	Неверный параметр в списке адресов входных маркеров связи yyyy = адрес неверно указанного байта маркеров	Прерывание

Метка ошибки SD3 SD4	Пояснения	Реакция при запуске
0414H yyyyH	Неверный параметр в списке адресов выходных маркеров связи УУУУ = адрес неверно указанного байта маркеров	Прерывание
0415H yyyyH	Неверное число таймеров (допустимое количество 128) yyyy = неверное число таймеров	Прерывание
0419H yyyyH	Задержка квитирования от цифровых входов yyyy = адрес неквитируемого байта входов	Прерывание
041AH yyyyH	Задержка квитирования от цифровых выводов yyyy = адрес неквитируемого байта выходов	Прерывание
041BH yyyyH	Задержка квитирования от входных маркеров межпроцессорной связи yyyy = адрес неквитируемого байта маркеров	Прерывание
041CH yyyyH	Задержка квитирования от выходных маркеров межпроцессорной связи yyyy = адрес неквитируемого байта маркеров	Прерывание

о Информация об ошибках при обработке DB2 - инициализация распределителя вызовов регулятора

Метка ошибки SD3 SD4	Пояснения	Реакция при запуске
0421H DByyH	Незагруженный блок данных yy = номер незагруженного блока	Прерывание
0422H FByyH	Незагруженный функциональный блок yy = номер незагруженного блока	Прерывание

о Информация об ошибках при обработке DXO — параметров операционной системы

Метка ошибки 3D3 SD4	Пояснения	Реакция при запуске
0431К уууuН	Недействительная метка блока уууу = неверная метка	Прерывание
0432Н уууuН	Неизвестный параметр уууу = неверный параметр	Прерывание
0433Н УУУуН	Недопустимый параметр УУУУ = неверный параметр	Прерывание
0434Н УУУуН	Недопустимое число таймеров (допустимое количество 128) уууу = неверное число таймеров	Прерывание
043SH уууuН	Недопустимое время цикла допустимо от 1мс до 4с уууу = неверное время	Прерывание

7.2 Информация об ошибках через АКК 1 и АКК 2

o Информация об ошибках при обработке функции регулятора

Метка ошибки АКК 1 АКК 2	Пояснения	Реакция при запуске
O801H DByyH	Ошибка времени опроса yy= номер соответствующего блока данных регулятора	Вызов OB34 (если не маскировано.)
0802H DByyH	Блок данные регулятора не загружен yy= номер незагруженного блока	Вызов OB34
0803H FByyH	Функциональный блок регулятора не загружен yy= номер незагруженного блока	Вызов OB34
0880H yyyyH	Задержка квитирования (QVZ) во время обновления области отображения процесса регулятора	Вызов OB34

o Информация об ошибках при ошибках кодов команд MC5

Метка ошибки АКК 1 АКК 2		Пояснения	Реакция при запуске
1S01H	—	Ошибка замещения в команде BBS	Вызов OB27
1802H	—	Ошибка замещения в команде BDW/BMW	Вызов OB27
1803H	—	Ошибка замещения в команде BX/BIX	Вызов OB27
1804H	—	Ошибка замещения в команде LX/TX	Вызов OB27
1805H	—	Ошибка замещения в команде UX/UNX/ OX/ONX/ = X/SX/RBX	Вызов OB27
1811H	—	Команда с недопустимым кодом	Вызов OB29
1812H	—	Недопустимая команда ACU	Вызов OB29
1813H	—	Недопустимая команда BEFE	Вызов OB29

Метка ошибки АКК 1 АКК 2	Пояснения	Реакция при запуске
1814H —	Недопустимая команда BESY	Вызов OB29
1315H —	Недопустимая команда RECH	Вызов OB29
1821H —	Недопустимый параметр при ADBO,1,2	Вызов OB3O
1822H —	Недействительный номер таймера *в* команде LT	Вызов OB3O
1823H —	Недействительный номер таймера в команде FT	Вызов OB3O
1824H —	Недействительный номер таймера в команде LCT	Вызов OB3O
1825H —	Недействительный номер таймера в команде RT	Вызов OB3O
1826H —	Недействительный номер таймера в команде SAT	Вызов OB3O
1827H —	Недействительный номер таймера в команде SET	Вызов OS30
1828H —	Недействительный номер таймера в команде SIT	Вызов OB3O
1829H —	Недействительный номер таймера в команде SST	Вызов OB3O
182AH —	Недействительный номер таймера в команде SVT	Вызов OB3O
182BH —	Недопустимый параметр в SPA (B) OBO	Вызов OB3O
182CH —	Недопустимый параметр в команде SPA OB>39: специальные функции отсутствуют	Вызов OB3O

о Информация об ошибках при выполнении команд MC5

Метка ошибки АКК 1 АКК 2	Пояснения	Реакция при запуске
1A01H —	Нет загруженного блока данных при выполнении команды ADB	Вызов 0B19
1A02H —	Нет загруженного блока данных при выполнении команды AXDX	Вызов 0B19
1A03H —	Нет загруженного блока данных при выполнении SPA (B) FB, OB, PB, SB	Вызов 0B 19
1A04H —	Нет загруженного блока данных при выполнении BA (B) FX	Вызов 0B 19
1A05H —	Нет загруженного блока данных при выполнении OB254 или OB255	Вызов 0B 19
1A11H —	Ошибка передачи при BIT MD в неопределенное слово данных	Вызов OB32
1A12H —	Ошибка передачи при TDR в неопределенное слово данных	Вызов OB32
1A13H —	Ошибка передачи при TDL в неопределенное слово данных	Вызов OB32
1A14H —	Ошибка передачи при TDW в неопределенное слово данных	Вызов OB32
1A15H —	Ошибка передачи при TDD в неопределенное слово данных	Вызов OB32
1A21H —	Ошибка специальной функции при EDB или EXDX. Блок данных уже существ.	Вызов OB31
1A22H —	Ошибка специальной функции при EDB или EXDX. Недопустимая длина блока данных (5слов или > 4Кслов)	Вызов OB31
1A23H —	Ошибка специальной функции при EDB или EXDX. Недостаточно места в памяти RAM	Вызов OB31
1A25H —	Ошибка специальной функции при BIX недопустимый параметр в АКК 1	Вызов OB.31
1A27H —	Ошибка специальной функции при U(переполнение стека скобок	Вызов OB31
1A28H —	Ошибка специальной функции при 0(переполнение стека скобок	Вызов OB31
1A31H —	OB254 или OB255 блок данных уже имеется в RAM	Вызов OB31

Метка ошибки АКК 1 АКК 2	Пояснения	Реакция при запуске
1A32H —	OB254 или OB255 новый блок данных уже существует	Вызов OB31
1A33H —	OB254 или OB255 недостаточно места в памяти RAM	Вызов OB31

o Информация об ошибках при обработке задержки квитирования

Метка ошибки АКК 1 АКК 2	Пояснения	Реакция пои запуске
1E23H yyyyH	Задержка квитирования (QVZ) при обращении к периферии yyyy = адрес QVZ	Вызов OB23
1E25H yyyyH	Задержка квитирования (QVZ) при обновлении области отображения процесса цифровых выходов УУУУ = адрес неквитируемого выходного байта	Вызов OB24
1E26H yyyyH	Задержка квитирования (QVZ) при обновлении области отображения процесса цифровых входов yyyy = адрес неквитируемого входного байта	Вызов OB24
1E27H yyyyH	Задержка квитирования (QVZ) при обновлении области отображения выходных маркеров связи yyyy = адрес неквитируемого маркерного байта	Вызов OB24
1E2SH yyyyH	Задержка квитирования (QVZ) при обновлении области отображения входных маркеров связи yyyy = адрес неквитируемого маркерного байта	Вызов OB24

Дополнение к руководству по программированию R-процессора C79000-B8500 C364-01

Новые встроенные функции в R-процессоре
(программирование начиная с выпуска 9, версии операционной системы 4.0)

Обзор

OB 110	Обращение к регистрам индикации
OB 120	Прерывание от процесса
OB 122	Задержка прерывания от процесса
OB 160 -163	Цикл счета
OB 170	Считывание стека блоков (BSTACK)
OB 181	Тестирование блоков данных (DB/DX)
OB 190, 192	Передача маркеров в блок данным
OB 191, 193	Передача блоков данных в маркерную область
OB 228	Считывание информации о состоянии уровня обработки программ

Обработка ошибок

Для новых спецфункций не имеется в настоящее время 0B ошибок. Однако, Вы можете в таких случаях обработать результат логической операции как сигнал о том есть ошибка или нет ошибки и в зависимости от этого предусмотреть, в случае ошибки, специальную программу обработки ошибок.

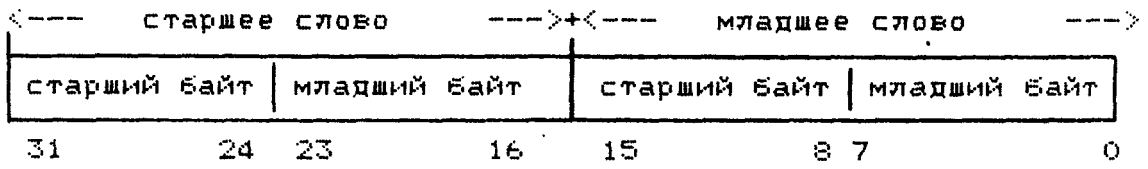
Если при обработке этой специальной функции возникла ошибка, в большинстве случаев, устанавливается VKE (VKE=1) и это может быть обработано с помощью команды SPB (перевод по условию).

При некоторых спецфункциях индикация результата ANZO н ANZ1 проводят влияют на обработку спецфункций и, поэтому могут опрашиваться.

Указание:

При параметрировании отдельных организационных блоков спецфункций примите во внимание, пожалуйста, следующие принципы записи:

Akku 1:	Akku 1,	32 бита
Akku 1-L:	Akku 1, младшее слово, младший байт,	16 бит
Akku 1-LL:	Akku 1, младшее слово, старший байт,	8 бит
Akku 1-LH:	Akku 1, младшее слово, старший байт,	8 бит



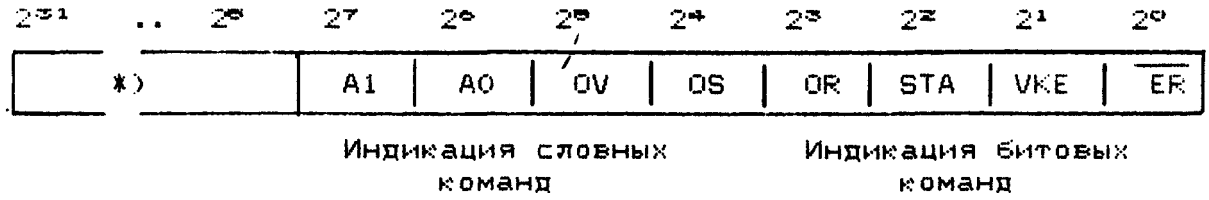
Указание:

В понятие Параметр в нижеследующем описании спецфункций входят все данные, которые необходимы процессору для того, чтобы он мог корректно выполнить спецфункцию. Перед вызовом спецфункции в программе STEP5 Вы должны загрузить эти данные в аккумулятор или в соответствующие ячейки памяти.

1 . Обращение к регистру индикации (OB 110)

Регистр индикации содержит информацию о результатах арифметических или логических операций и может быть обработан с помощью специальных, зависящих от индикации, команд. С помощью организационного блока спецфункции OB 110. Вы можете загрузить регистр индикации в аккумулятор 1 или записать в него содержимое аккумулятора 1. Дополнительно Вы можете установить биты индикации в "0" или "1".

Содержимое аккумулятора 1 при обращении к регистру индикации;



*) Биты с 2^5 по 2^{31} зарезервированы для дополнительных действий и при записи в регистр индикации должны быть равными "0" . Вы должны при считывании регистра индикации игнорировать их.

Параметры;

1. Akku 2-L: Метка функции

   возможные значения: 1, 2 или 3

2. Akku1: новый байт индикации или маска

Akku 2-L	Akku 1		Функция:
	до	после	
1	новый байт индикации маска	новый байт индикации	Содержимое аккумулятора 1 загружается в регистр индикации
2	маска	новый байт индикации	В маске индикации в Akku 1 все биты установленные в "1", устанавливаются в "1" и в регистре индикации. Новый байт индикации загружается в аккумулятор 1.
3		новый байт индикации	В маске индикации в Akku 1 все биты установленные в "1", устанавливаются в "0" в регистре индикации. Новый байт индикации загружается в аккумулятор 1.

Случаи ошибок:

- Метка функции в аккумуляторе Akku-2-L не равна 1, 2 или 3.

- В аккумуляторе 1 установлен в "1" один из битов от 2^5 по 2^{31}

В случае ошибки вызывается 0B 31 и в аккумулятор Akku 1-L загружается метка ошибки 1A49H.

2 Блокировка обработки прерывания от процесса/по времени или задержка (OB 120 и 0B 122)

Программа STEP5 может быть прервана на границе блока или на границе команды программой с более высоким приоритетом. К таким более приоритетным уровням обработки программ принадлежат прерывание от процесса и прерывание по времени. Выполнение прерванной программы продолжается при этом после выполнения этой вложенной программы.

С помощью организационных блоков спецфункций OB 120 и OB 122 Вы можете блокировать прерывание от процесса и прерывание по времени на одной или нескольких, идущих друг за другом границах блоков или команд.

OB 120: Включение/выключение блокировки прерывания

Этот организационный блок спецфункций имеют влияние на прием прерывания.

Блокировка прерывания включена означает, что начиная с этого момента прерывания больше не регистрируются и соответствующие прерывания уже были зарегистрированы (например, ожидают на границе блока) стираются. Только в случае OB 2 (прерывание от процесса.) или OB 13 (прерывание по времени) если они уже начались, то эти прерывания обрабатываются полностью.

"Блокировка прерывания выключена" означает, что с этого момента все возникающие прерывания регистрируются и будут вызваны и обработаны на следующей границе блока или команды.

ОБ 122: Включение/выключение задержки прерывания

Этот организационный блок спецфункций имеют влияние на обработку прерывания.

"Задержка прерывания включена" означает, что все возникающие впоследствии прерывания регистрируются и уже имеющиеся прерывания остаются зарегистрированными. Обработка зарегистрированных прерываний, однако, не начинается сразу. Проход всех границ команд или блоков для обработки прерываний становится недействительным. Только, если уже была начата обработка OB 2 (прерывание от процесса) или OB 13 (прерывание по времени), то они обрабатываются полностью.

"Задержка прерывания включена" означает, что зарегистрированные прерывания будут вызваны и обработаны на следующей границе блока или команды.

ОВ 120 и ОВ 122 помечают блокированные или задержанные прерывания в управляющем слове со следующими значениями:

Бит 2°: Прерывание по времени

Вит 2^2: Прерывание от процесса

Бит 2^1, 2^3 по 2^{31} : зарезервированы; эти биты должны быть равны О

Параметры:

1. Akku 2-L: Метка функции
   возможные значения: 1, 2 или 3

2. Akku 1: новое управляющее слово или маска

Akku 2-L	Akku 1		Функция:
	до	после	
1	управ-ляющее слово	Управ-ляющее слово	Содержимое аккумулятора 1 загружается в управляющее слово
2	маска	Новое управ-ляющее слово	В маске аккумулятора 1 все биты установленные в "1", устанавливается в "1" и в управляющем слове. Новое управляющее слово загружается в аккумулятор 1.
3	маска	новое управ-ляющее слово	В маске аккумулятора 1 все биты, установлен-ные в "1", устанавливаются в "0" в управляющем слове. Новое управляющее слово загружается в аккумулятор 1.

Случаи ошибок:

- Недопустимая метка функции в аккумуляторе Akku 2-L

- В аккумуляторе 1 установлен в "1" один из битов от 2^5 по 2^{31}.

В случае ошибки вызывается 0В 31 и в аккумулятор Akku 1 передается метка ошибки:

1A47H при ОВ 120
1A48H при 0В 122

Указание:

- Состояние обработки прерывания может быть определено путем считывания системных слов данных BS 131 и BS 132.

BS 131 управляющее слово блокировки прерывания

BS 131 управляющее слово задержки прерывания

-Для блокировки и деблокировки прерываний от процесса можно использовать также в случае OB 120 команды AS и AF:

AS соответствует

```
:L    KB2
:L    KM00000000 00000100
:SPA  OB120
```

AF соответствует

```
:L    KB3
:L    KM00000000 00000100
:SPA  OB120
```

3 Цикл счета (OB 160 по 163)

С помощью этих организационных блоков спецфункций Вы можете реализовать программные циклы с более удобным выполнением.

Каждому из 4 OB-спецФункций соответствует определенное системное слово данных

```
BS 60  --->  OB 160
BS 61  --->  OB 161
BS 62  --->  OB 162
BS 63  --->  OB 163
```

В эти системные слова данных Вы можете передать нужное Вам количество выполнении цикла. Если теперь Вы вызываете соответствующий OB-спецфункции, то счетчик цикла в системном слове данных уменьшается на 1. Цикл выполняется столь долго, пока счетчик циклов не будет содержать ноль. (Если счетчик цикла уже при вызове OB-спецфункции содержит значение ноль, то он уменьшается при вызове соответственно на единицу: таким образом, получится 65 536 выполнении цикла!)

Счетчик цикла в системном слове данных > O: VKE установлен (VKE = 1)

Счетчик цикла в системном слове данных = O: VKE сброшен (VKE = 0)

Остальные биты индикации битовых и словных команд всегда сброшены!

Содержимое аккумуляторов не сбрасывается и не обрабатывается. Таким образом, то что имелось в них к началу следующего выполнения цикла может быть использовано и не должно быть заново переустановлено.

Четыре организационных блока 0B 160, 161, 162 и 163 позволяют четырехкратное вложение циклов. Вы можете, при этом, в системные слова данных с BS 60 по 63 занести четыре различных количества циклов.

В данном случае, можно использовать эти спецфункции вместе с командой B BS (обработка системных данных).

Параметры:

1. Системные слова данных BS 60 – 63: счетчики циклов

   возможные значения:

   0 - 65 535 (FFFFH)

Случаи ошибки: нет

Пример:

В маркерном слове x указано необходимое число выполнения циклов.

```
Инициализация цикла:          :L KB0
                              :L MWx        счетчик циклов
                              :!=F
                              :SPB =M002
                              :T BS62       счетчик циклов
                                            передается в системное
                                            слово данных
'Программа цикла':       M001:  .
                              :  .
                              :  .
                              :  .
                              :
Обработка цикла:              :SPA OB162    цикл счета
                              :SPB =M001    при VKE = 1 цикл
                                            продолжает выполняться
дальше                   M002:  .
                              :  .
                              :  .
                              :  .
```

4 Считывание стека блоков (BSTACK)(OB 170)

В стек блоков заносятся все блоки начиная с OB 1 или FB 0, которые были вызваны друг за другом и чья обработка еще не завершена.

С помощью организационного блока спецфункции OB 170 Вы можете считать имеющуюся в BSTACK в настоящее время информацию в блок данных. Таким образом, Вы можете сохранять имеющееся в BSTACK количество информации и, вместе с тем, иметь в своем распоряжении резерв, который можно использовать для дальнейшего ввода информации.

Для каждого ввода информации Вы имеете соответствующий адрес возврата (счетчик адресов = SAZ.), абсолютный начальный адрес действительного для данного блока данных (DBA), а также его длину (число слов данных = DBL).

ВАЖНО! Перед вызовом 0B 170 необходимо вызвать достаточно длинный блок данных (DB или DX)! Для каждого желаемого ввода информации в BSTACK Вам необходимо иметь четыре слова данных.

Параметры

1. Akku 2-L: Номер слова данных (DM rr), начиная с которого в вызванном DB должны

заноситься вводы информации ("смещение")

Akku 1-L: Необходимое количество элементов BSTACK

возможные значения: 1 – 20

Пример: если Akku 1-L содержит значение '1'; то Вы получите последний ввод информации в BSTACK, при '2' последний и предпоследний и т.д.

После успешного вызова 0B 170

- в Akku 2-L устанавливается смещение блока данных

-" в Akku 1-L устанавливается действительно имеющееся число элементов BSTACK,

возможные значения; O - 20, причем
имеющееся число =< необходимого числа
O = 'нет вводов информации в BSTACK' или 'ошибка'

(Содержимое Akku 1-L умноженное на 4 дает число записанных слов данных в вызванном DB!)

- стирается VKE,

- могут быть обработаны биты индикации результата ANZO и ANZ1 (см. ниже),

- все остальные биты индикации стираются.

Случаи ошибки:

- не вызван блок данных

- вызванный блок отсутствует или имеет недостаточную длину

- недопустимые параметры в Akku 1 и 2

В случае ошибки VKE, а также биты индикации результата ANZO и ANZ1 устанавливаются (VKE, ANZO и ANZ1 = 1), остальные биты индикации битовых и словных команд сбрасываются. Содержимое аккумулятора Akku 1 равняется 'O'.

Влияние битов индикации результата VKE, ANZO и ANZ1

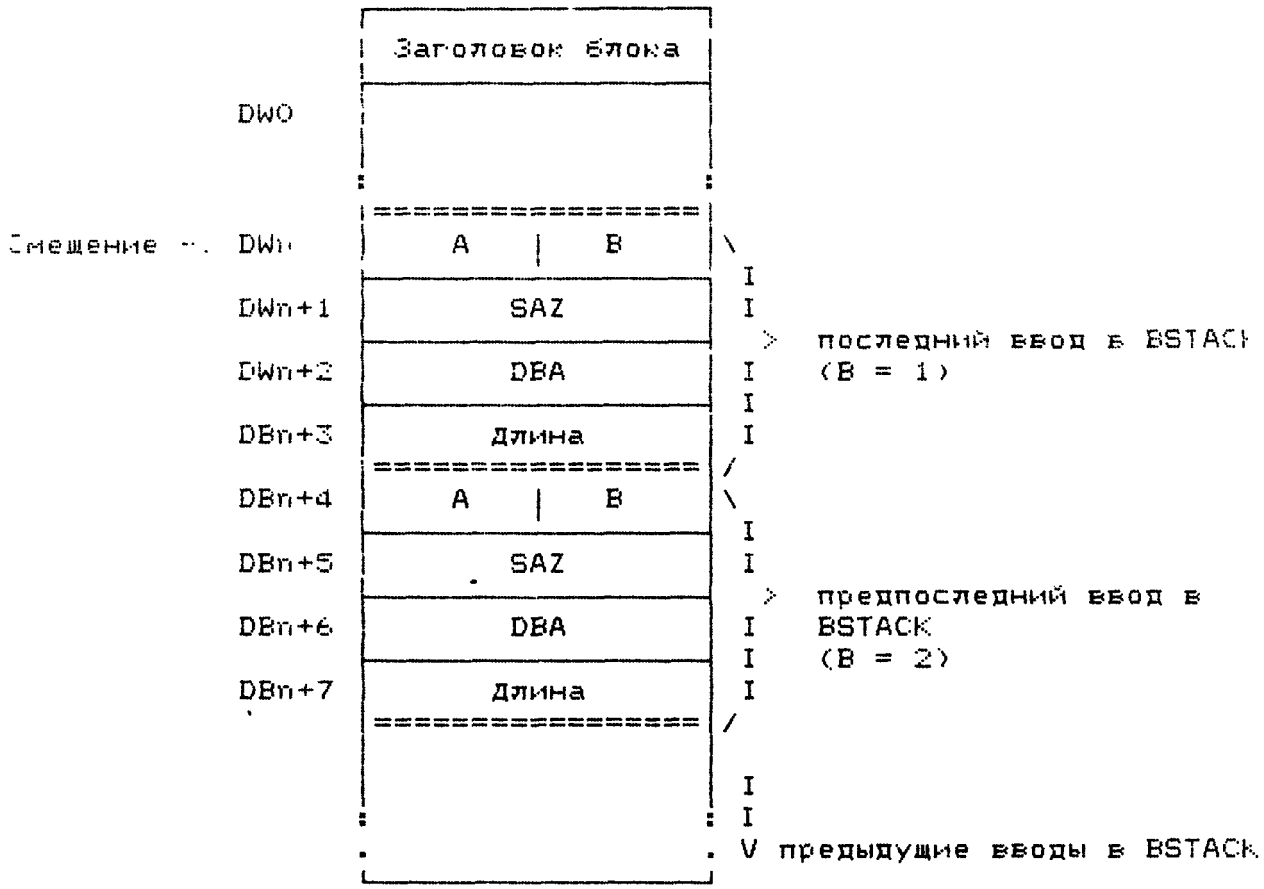
VKE	ANZO	ANZ1	Опрашивается с	Значение
0	1	0	SPM	имеющееся число элементов BSTACK < указанного числа
0	0	0	SPZ	имеющееся число элементов BSTACK =
0	0	1	SPP	имеющееся число элементов BSTACK > указанного числа
1	1	1	SPB	ошибка

Так содержимое BSTACK пои вызове 0B 170 заносится в вызванный блок данных

A = номер элемента BSTACK (20 - 1)

(Уже при выдаче последнего элемента BSTACK получаем соответствующий резерв: A = 17 -> резерв = 16).

B = число элементов BSTACK (1 - 20, глубина стека)

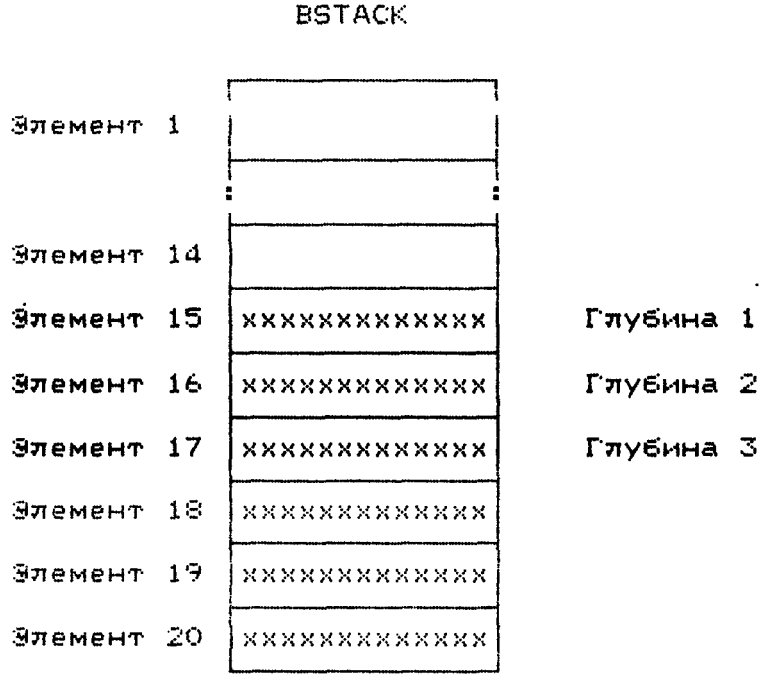


Пример:

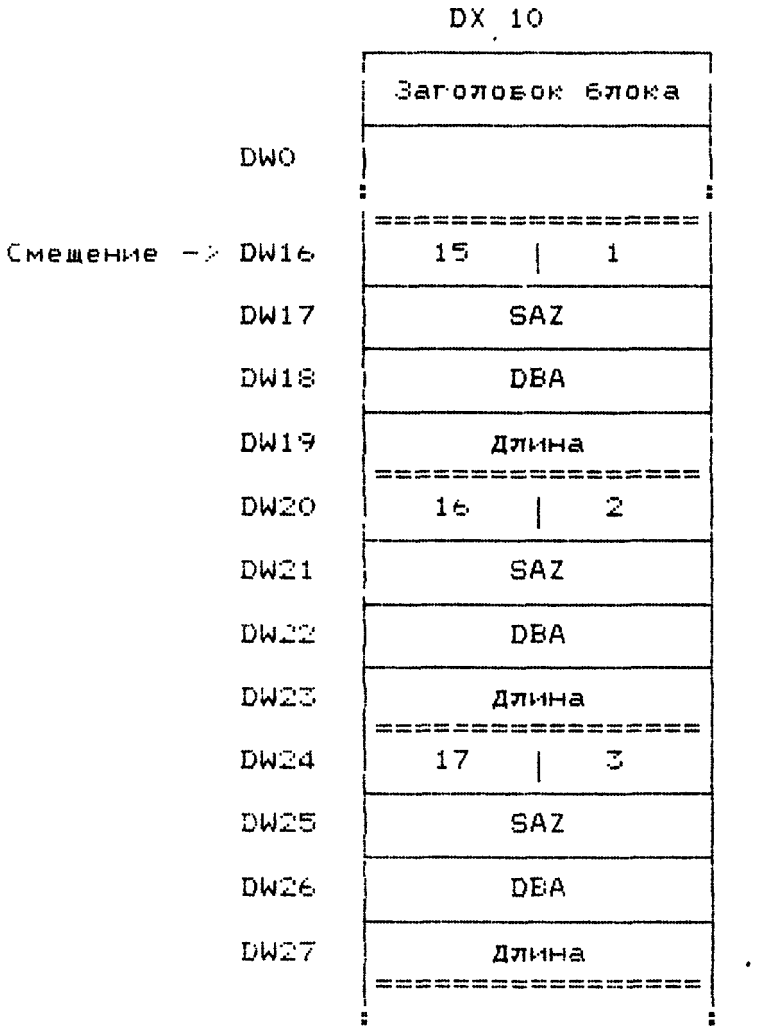
Вы хотите считать в блок данным DX 10 последние 3 ввода информации в BSTACK. Вводы должны быть занесены в DX 10 начиная со слова данных DW 16.

```
:AX  DX  10              ; вызов DX 10
:L   KY   16             ;вводы в BSTACK должны заноситься с DW 16
:L   KY    3             ;заказаны последние 3 ввода в BSTACK
:SPA OB 170
```

В BSTACK занесены в данный момент 6 блоков:



После вызова OB спец функции DX 10 заполняется следующим образом:



Akku 2-L: 16
Akku 1-L: 3

VKE = 0
ANZO = 0
AN21 = 1

5 Тестирование блоков данных (DB/DX) (OB 181)

С помощью организационного блока спецфункции OB 181 можно проверить,

a) имеется ли определенный DB- или DX-блок данных,
b) по какому адресу расположено первое слово данных блока данных,

c) сколько слов данных содержит указанный блок данных
d) тип памяти и область (память пользователя: RAM или EPROM, DB-RAM).

Применение функции "Тестирование DB/DX" имеет смысл перед командами TNB/TNM, E DB/EX DX и перед вызовом организационных блоков спецфункций OB 254 и OB 255.

Так Вы можете, например, перед передачей блока слов данных вызвать OB 181 для того, чтобы удостовериться, что блок данных назначения действительно существует и имеет достаточную длину, чтобы принять все копируемые слова данным.

Параметры:

1. Akku 1-LL: Номер блока
   возможные значения: от 1 до 255

2. Akku 1-LH: Метка блока
   возможные значения: 1 = DB
   2 = DX

Если проверяемый блок имеется в процессоре,

- в Akku 1-L заносится адрес первого слова данных (DB O),

- в Akku 2-L заносится длина блока данных в словах (без заголовка блока),
Пример: в Akku 2-L стоит значение '7' --> блок данных состоит из слов от DW O до DW 6.

- VKE сбрасывается (VKE = O),
- биты индикации словных команд ANZO и ANZ1 обрабатываются (см. следующий список)
- остальные биты индикации битовых и словных команд сбрасываются.

Если проверяемый блок отсутствует в памяти или параметрирование неверно,

- то устанавливается VKE (VKE = 1),

- биты индикации словных команд ANZO и ANZ1 обрабатываются (см. следующий список),

- остальные биты индикации битовых и словных команд сбрасываются

- содержимое аккумуляторов не изменяется.

Случаи ошибок:

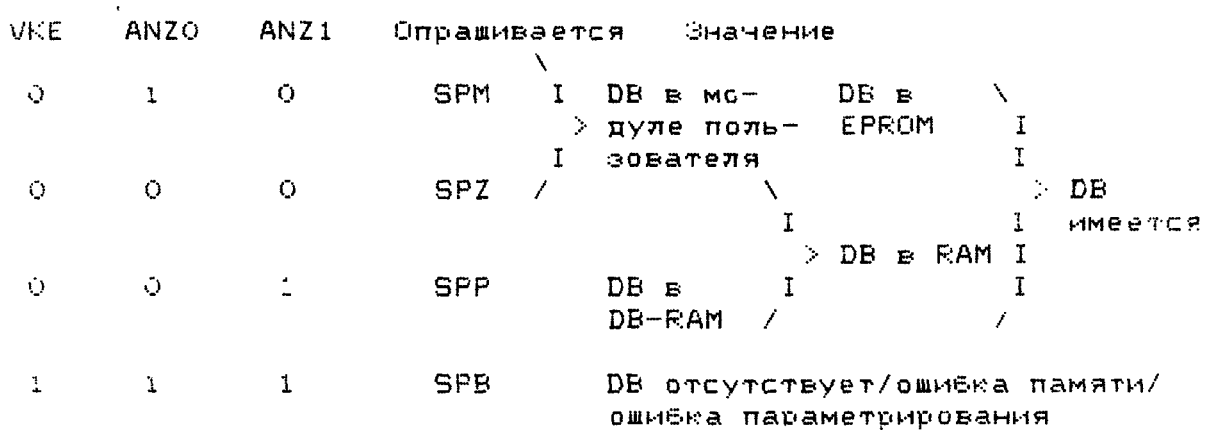
- неверный номер блока (недопустимо: О)

- неверная метка блока (недопустимо; О, 3 – 255)

- ошибка памяти

Обзор: влияние битов индикации результата VKE, ANZO и ANZ1

VKE = 0	DB имеется
VKE = 1	DB отсутствует или ошибка
ANZ1 = 0	DB в модуле пользователя
ANZ1 = 1	DB в DB-RAM
ANZO = 0	DB в памяти 'запись-чтение' (RAM) DB в памяти
ANZO = 1	'только чтение' (EPROM')



6. Передача маркеров в блок данных (ОВ 190 и 0В 192)

Организационные блоки ОВ 190 и ОВ 192 указанное пользователем число маркерных байтов в предусмотренный для этого блок данных.

Это может, например, иметь преимущество перед вызовом блоков в организационных блоках ошибки или при прерывании циклической обработки программ от программы управляемой по времени или по прерыванию от процесса.

С помощью OB 190 и OB 192 Вы можете эти маркерные байты снова восстановить из блока данных.

ВАЖНО!
Перед соответствующим вызовом необходимо вызвать блок данных (DB/DX)!

После вызова 0B 190/192 в вызванном блоке данных сохраняются маркерные байты, начиная с указанного адреса слова данных. Область сохраняемых маркеров 0B 190/192 берет из аккумулятора.

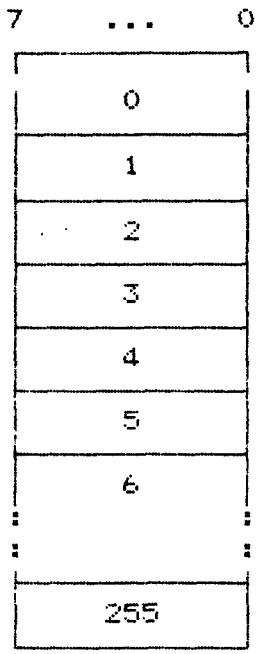
Организационные блоки спецфункцнй ОВ 190 и 0B 192 работают аналогично, за исключением типа и способа передачи маркерных байтов:

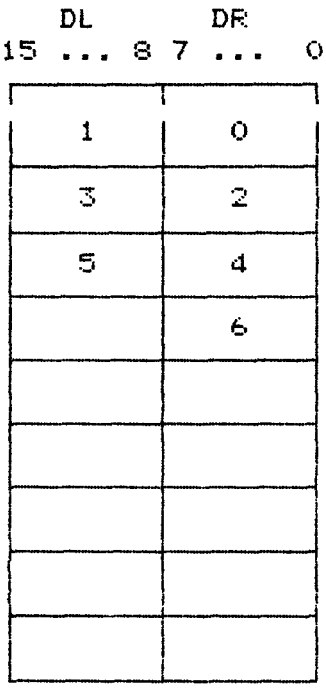
- ОВ 190 передает маркеры побайтно
- ОВ 192 передает маркеры пословно

Это важно, если данные, передаваемые в блок данных, после этого должны быть обработаны, а блок данных используется не только как простая промежуточная память.

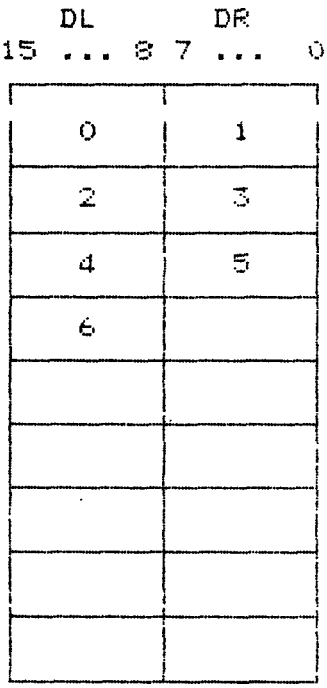
Следующий рисунок должен пояснить эти различия:

Маркер копируется с помощью ОВ 190:





ОВ 192:



Указание: В случае, если передается нечетное число маркерных байтов, то последнее используемое слово данных блока данных используется только наполовину. При ОВ 190 свободной остается левая половина слова данных, при ОВ 192 - левая.

Параметры:

Указание на источник:

1. Akku 2-LH: первый передаваемый маркерный байт возможные значения: от О до 255
2. Akku 2-LL: последний передаваемый маркерный байт возможные значения: от О до 255

   (последний маркерный байт ≥ первый маркерный байт!)

Указание на приемник:

3. Akku 1-L: адрес первого используемого слова данные в вызванном блоке данных

Если ОВ спецфункций 190/192 обработаны корректно, то VKE сбрасывается (VKE = О).. Содержимое аккумулятора не изменяется.

В случае сшибки VKE устанавливается (VKE = 1), содержимое аккумулятора не изменяется.

Случаи ошибок:

- отсутствует вызванный DB- или DX- блок данных

- неверная область маркеров (последний маркерный байт < первого маркерного байта)

- адрес слова данных отсутствует

- длина DB- или DX- блока данных недостаточна

7 Передача блоков данных в маркерную область (ОВ 191 и ОВ 193)

С помощью ОВ 191 и ОВ 193 Вы можете передавать данные из Слова данных в маркерную область. Так, например, можно сохраненное перед этим в блоке данных маркерные байты снова вернуть в область маркеров.

ОВ 191/193 также как и ОВ 190/192 различаются тем, что меняются местами источник и приемник:

ОВ 190/193 Область маркеров --> Маркер →. Блок данных

ОВ 191/193: Область маркеров ← Данные ← Блок данных

Важно!

Перед соответствующим вызовом необходимо вызвать

блок данных (DB/DX)!

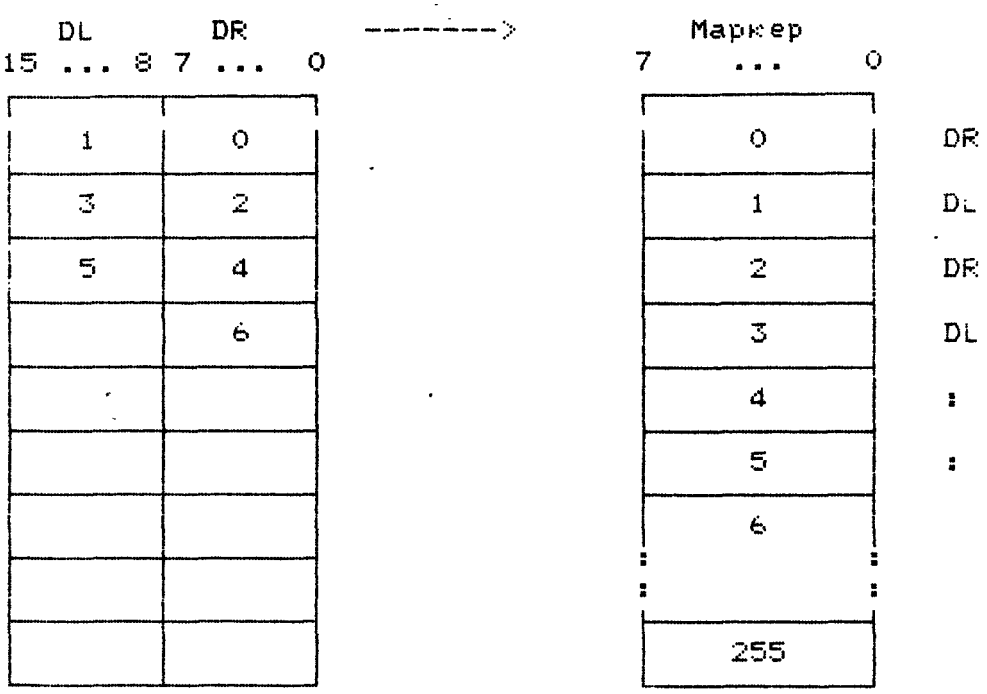
Организационные блоки спецфункций ОВ 191 и ОВ 193 работают аналогично, за исключением вида и способа передачи данных:

- ОВ 191 передает слова данных побайтно

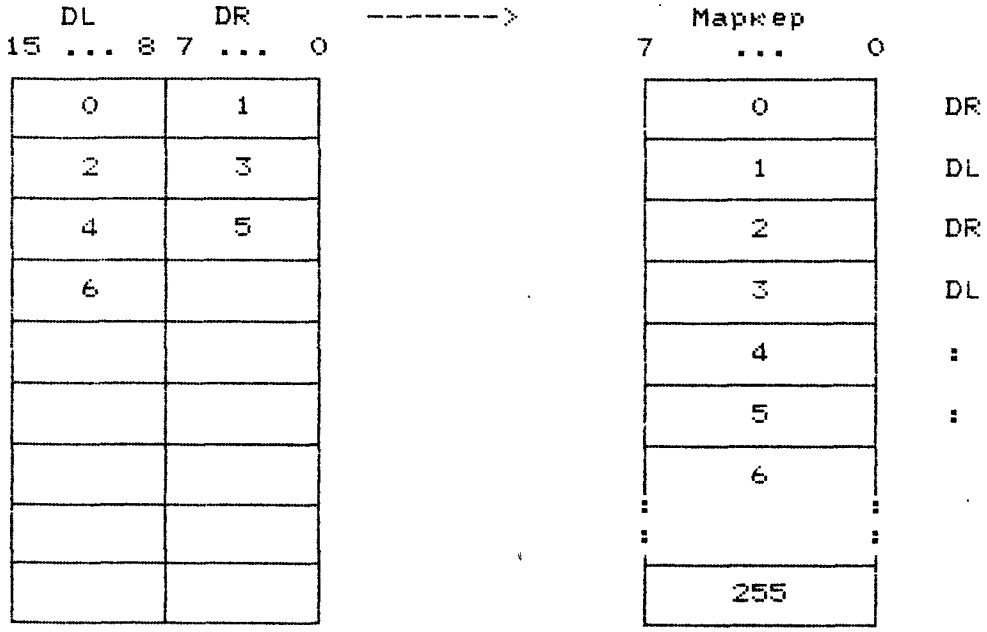
- ОВ 193 передает слова данных пословно

Следующий рисунок должен пояснить эти различия;

Данные передаются с помощью OB191:



Данные передаются с помощью **OB193:**



Параметры:

Указание источника:

1. Akku 2-L: адрес первого передаваемого слова данных в вызванном блоке данных

Указание приемника:

2. Akku 1-LH: первый записываемый маркерный байт возможные значения: от О до 255

3. Akku 1-LL: последний записываемый маркерный байт: возможные значения: от O до 255.

(последний маркерный байт ≥ первый маркерный байт!)

Если OB спецфункций 191/193 обработаны корректно, VKE сбрасывается (VKE = O). Содержимое аккумулятора не изменяется.

В случае ошибки VKE устанавливается (VKE = 1), содержимое аккумулятора не изменяется.

Случаи ошибок: см. OB 190/192

Пример:

Перед вызовом программного блока PB 12 необходимо сохранить все маркеры (от MB O до MB 255) в блоке данных DX 37 начиная с адреса 100 и затем снова их восстановить.

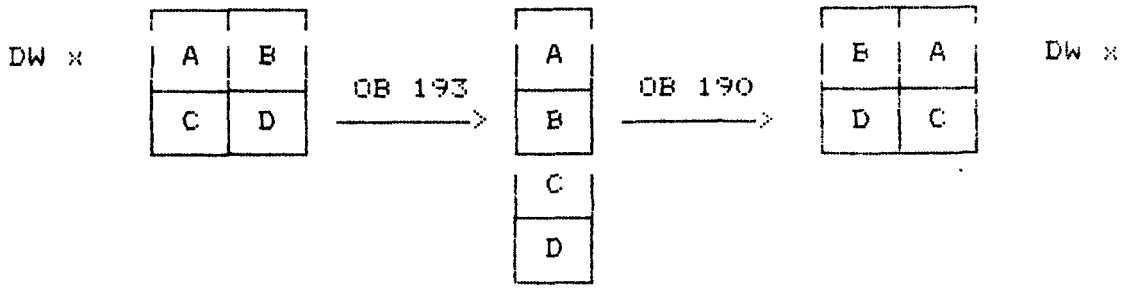
Сохранение:	:AX DX37	вызов блока данных
	:L KYO,255	маркерная область MBO - M255
	:L KB 100	адрес первого слова данных
	:SPA OB 190	сохранение маркеров
Смена блока:	:SPA PB12	
Восстановление:	:	(блок данных уже вызван)
	:L KB100	адрес первого слова данных
	:L KYO,255	маркерная область MBO - M255
	:SPA OB191	восстановление маркеров

Другие применения организационных блоков OB 190 и OB 193

- Вы можете передвигать блоки данных внутри блоков данных, если Вы при обратной передаче из маркерной области укажете тот же самый номер DB, однако, другое слово данных.

- Кроме того, легко передавать блоки данных (максимум 255 байт) в другой блок данных (возможно перед этим произвести вызов OB 181 'тестирование блоков данных (DB/DX)').

- Без больших затрат, можно поменять местами в блоке данных старшие и младшие байты, передавая слова данных с помощью соответствующего OB в область маркеров и назад:



8 Считывание информации о состоянии уровня обработки программ (OB 228)

При получении определенных результатов, операционная система вызывает соответствующий уровень обработки программ. При этом, уровень обработки программ 'активизируется'.

С: помощью организационного блока 0B 228 Вы можете установить активен ли в данный момент определенный уровень обработки программ или нет. В аккумулятор 1 Вы передаете соответствующий уровень обработки программ, чье состояние должно быть опрошено. (Номер соответствует номеру, введенному в USTACK по меткам 'УРОВЕНЬ' или 'VEK-ADR' в зависимости от используемого пакета программ PG.)

При вызове спецфункций OB 228 передает информацию о состоянии указанного уровня обработки программ в аккумулятор 1.

Параметры:

1. Akku 1-L: номер уровня обработки программ (см. USTACK, УРОВЕНЬ)

   возможные значения (шестнадцатеричные)
   02 = новый пуск или ручной перезапуск
   04 = цикл
   06 = прерывание по времени
   08 = регулятор
   0A = прерывание от процесса
   0C = аппаратное прерывание
   10 = ошибка будильника
   12 = ошибка регулятора
   14 = ошибка цикла
   16 = автоматический перезапуск
   18 = ошибка кода команды
   1A = ошибка выполнения команды
   1C = ошибка адресации
   1E = задержка квитирования

Случаи ошибок: отсутствуют

После вызова OB 228

- информация о состоянии находится в аккумуляторе 1:

содержимое Akku 1 = O : уровень обработки программ не вызван

содержимое Akku 1 # O : уровень обработки программ активирован

предыдущее содержимое Akku 1 заносится в Akku 2,

предыдущее содержимое Akku 2 теряется.

Таким образом, Вы можете свою обработку программы выполнять в зависимости от состояния других уровней обработки программ.

Пример:

Задержка квитирования должна игнорироваться при новом пуске, но не на других уровнях обработки программ.

Вы вызываете в начале OB 23 организационный блок спецфункции OB 228 для того, чтобы установить был ли активизирован при возникновении QVZ уровень обработки программы НОВЫЙ ПУСК (номер 02), или нет. Дальнейшую обработку ошибки Вы выполняете в зависимости от информации состояния, которую Вы получите:

Akku 1 = O: НОВЫЙ СТАРТ пассивен --> QVZ не возникало при новом старте
--> программа ошибки должна быть обработана

Anku 1 ≠ НОВЫЙ СТАРТ активизирован → QVZ возникло при новом старте
→ QVZ может быть игнорирован

OB 228 позволяет Вам, таким образом, производить дифференцированную обработку ошибки.

SIEMENS

SIMATIC S5

Многопроцессорный режим работы
программируемых контроллеров S5-135U и S5-155U

Руководство по эксплуатации Заказной No . C79000-B8500-C500-02

Оглавление

1 Введение

2 Ввод в эксплуатацию

2.1 Необходимые условия
2.2 Порядок ввода в эксплуатацию
2.3 Режимы работы координатора
2.3.1 Обычный режим работы
2.3.2 Тестовый режим работы" '
2.4 PG - мультиплексор 'координатора 923C

3 Поиск неисправностей

1 Введение

Программируемые контролеры S5-135U и S5-155U являются представителями семейства контроллеров SIMATIC S5.

Программируемый контроллер может эксплуатироваться как в однопроцессорном, так и в многопроцессорном режиме. При работе в многопроцессорном режиме могут быть использованы до четырех процессоров одновременно.

В качестве центральных процессорных модулей (CPU) могут использоваться :

Для AG S5-135U:

CPU 921 (S—процессор):	предназначен для быстрой обработки двоичных сигналов (решение задач управления). Программируется на языке STEP5.
CPU 922 (R-процессор):	предназначен для быстрой обработки слов задачи регулирования, вычислений, контроля, сообщений). Программируется на языке STEPS.
CPU 928:	предназначен для быстрой обработки слов и очень быстрой обработки двоичных сигналов. Программируется на языке STEP5.
CPU 920 (M-процессор):	предназначен для программирования на языках высокого уровня (Бейсик, Си) и Ассемблере для решения задач арифметики, сортировки, статистики в рамках системных функций S5 (новый пуск, перезапуск, связь, отображение процесса)

Для AG S5-155U:

CPU 946/947:	предназначен для очень быстрой обработки слов и двоичных сигналов, особенно быстрой обработки двойных слов и чисел с плавающей запятой, а также для обработки больших программ, требующих значительных объемов памяти.

CPU 922 (R-процессор), CPU 928 и CPU 920 (M-процессор) могут работать также и в AG 155U.

CPU в данных центральных устройствах могут использоваться в любых комбинациях, если нет ограничений на установку из-за ширины передней панели или из-за перегрузки шины S5:

- CPU 920, 921 и 922 занимают каждый одно установочное место,
- CPU 928 занимает два установочных места,
- CPU 926/927 может занимать в общей сложности до 5-ти установочных мест (3 для CPU 946/947 и до 2 для модулей памяти 355) и устанавливаются только в двух установочных местах центрального устройства 155U.

При работе в многопроцессорном режиме каждый CPU, независимо от всех остальных, выполняет свою индивидуальную прикладную программу.

Обмен данными с периферийными модулями, коммуникационными процессорами, интеллигентными модулями периферии и другими CPU осуществляется через общую шину S5. При работе в многопроцессорном режиме доступ CPU к шине S5 регулируется координатором. Управление шиной описано в Руководстве по эксплуатации координаторов 923C и 923A.

Обмен данными между CPU в многопроцессорном режиме может происходить через

- "Маркеры связи":
  для циклического обмена двоичными данными (см. Руководство по программированию CPU) или через

- "Специальные функции для межпроцессорной связи":

  для программно управляемого обмена комплектных блоков данных только через координатор 923C, но не с S-процессором CPU 921 (см. Руководство по эксплуатации "Многопроцессорная коммуникация" в данном руководстве по устройству).

  В следующем разделе Руководства (смотри Раздел 2) приводятся методические указания по вводу в эксплуатацию программируемого контроллера S5-135U в многопроцессорном режиме.

Раздел 3 Руководства содержит описание внешних проявлений неисправностей. В этом Разделе также рассматриваются и возможные причины неисправностей.

2 Ввод в эксплуатацию

В этом Разделе описывается порядок ввода в эксплуатацию программируемого контроллера S5-135U в многопроцессорном режиме. Кроме того, рассматривается реакция модулей на выполнение отдельных шагов процедуры ввода в эксплуатацию.

2.1 Необходимые условия

Излагаемый далее материал рассчитан на пользователя, знакомого с облуживанием или программированием отдельных модулей в однопроцессорном режиме работы контроллера. С информацией по обслуживанию или программированию отдельных модулей в однопроцессорном режиме работы можно ознакомиться, обратившись к справочному Руководству по программируемому контроллеру S5-135U.

Также предполагается наличие рабочей программы и работоспособных модулей CPU.

2.2 Порядок ввода в эксплуатацию

В AG S5-135U и S5-155U можно использовать до четырех CPU. Допустимые места установки описаны в Руководствах по эксплуатации обоих устройств автоматизации.

Для работы в многопроцессорном режиме необходим координатор. В AG S5-135U можно использовать координаторы 923A и 923C, в AG S5-155U - только координатор 923C.

Координатор выделяет каждому CPU отрезок времени, в течение которого процессор получает доступ к шине и имеет глобальную память для обмена данными между CPU посредством маркеров связи. Кроме того KOR C (координатор C) содержит дополнительную память с четырьмя буферами связи для обеспечения функции "Многопроцессорная коммуникация", а также мультиплексор для последовательного PG-интерфейса PG-MUX".

Важно !
В координаторах 923 C и A не допускается изменение жестко
установленного времени деблокировки шины !

Указание:
Если координатор установлен в центральное устройство 135U или 155U, то все имеющиеся CPU автоматически переходят в многопроцессорный режим, даже если KOR работает всего с одним CPU, для всех CPU действительны условия многопроцессорного режима (DB1 необходим, DX0 может потребоваться, CPU 946/947 работает только в режиме 155U и т.д.).

Процедуру ввода в эксплуатацию можно разбить на следующие шесть шагов:

o Шаг 1:

Задание на координаторе количества используемых CPU и деблокировка маркеров сопряжения.

а) Координатор A (6ES5 923-3UA11)

*

Кодировка посредством перемычек на кодировочном штеккере EP62:

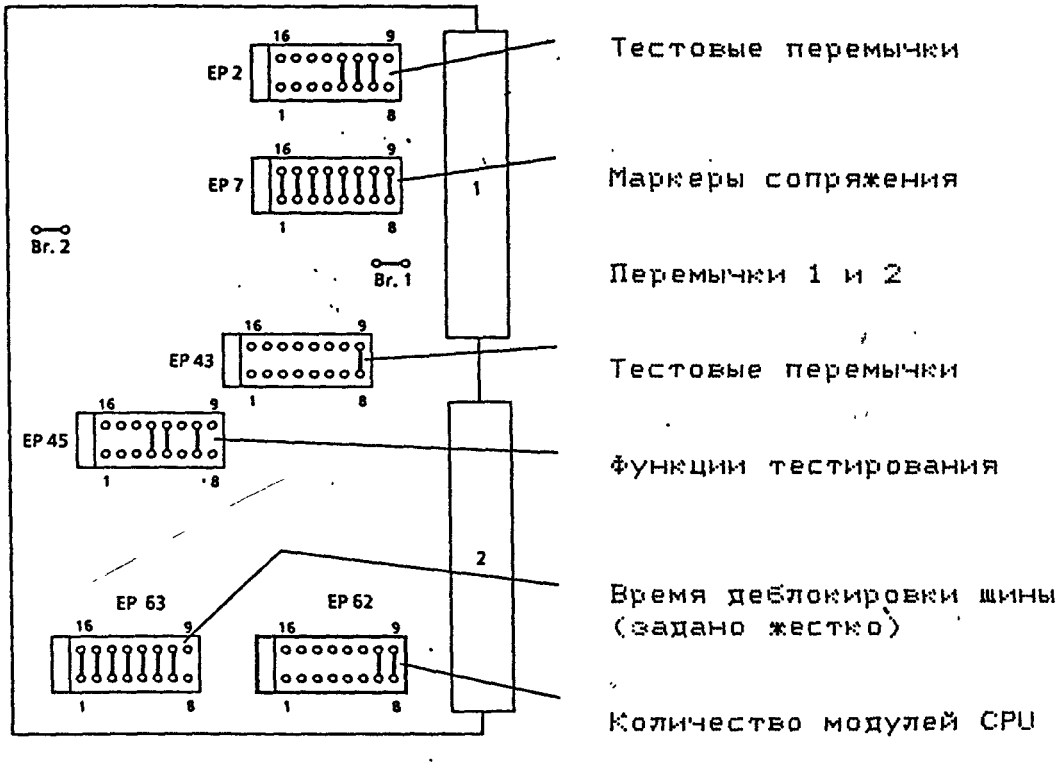


Рис.1 Расположение кодировочного штеккера на координаторе типа A

Количество центральных модулей	Перемычка на EP62
2	7–10, 8–9
3	7–10
4	8–9

Важно!
CPU 928 занимает в ПК S5-135U два установочных места. В многопроцессорном режиме он всегда должен располагаться справа от остальных CPU, чтобы не терялось установочное место для CPU и не было пустых мест.

б) Координатор C (6ES5 923-3UC11)

Кодировка выполняется установкой микропереключателей DIP – S1.4, S1.5 или S1.6, которые расположены в углублении лицевой панели:

	Включен	Выключен	
S1.1		X	–
1.2		X	–
1.3		X	Тестовый режим (см. Раздел 2.3.2)
1.4	X		Количество занимаемых CPU мест = 2
1.5		X	Количество занимаемых CPU мест = 3
1.6		X	Количество занимаемых CPU мест = 4

(X =положение переключателей при поставке)

Следующий этап работ заключается в установке, в случае необходимости, адресации сопряжения на координаторе.

256 байтов маркеров межпроцессорной связи можно отключать группами по 32 байта. Отключение выполняется удалением перемычек на копировочном штеккере EP7 (см. Рис. 1) координатора типа A или на кодировочном штеккере EP60 координатора типа C (см. Рис. 2).

Необходимость отключения блокировки маркеров межпроцессорной связи возникает в случае, когда эти маркеры используются центральными процессорами. В этой ситуации соответствующие области для предотвращения двойной адресации отключаются в обязательном порядке (см. соответствующие Руководства по устройствам).

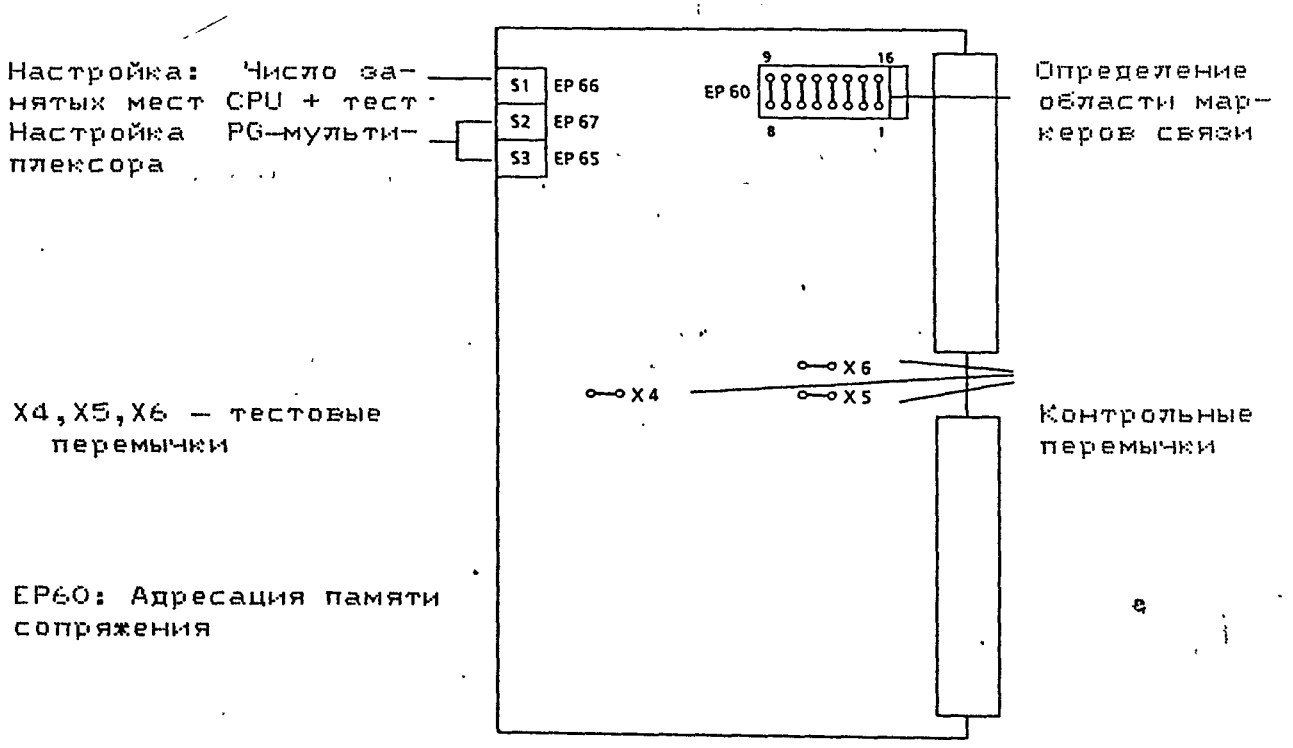
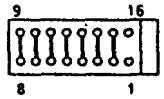


Рис.2 Расположение кодировочных перемычек на координаторе типа C.



В состоянии поставки включены все области маркеров связи

EP60 на координаторе C

EP7 на координаторе A

Перемычка	Байты маркеров связи	Адреса
8 – 9	с 0 по 31	с F200H по F21FH
7 – 10	с 32 по 63	с F220H по F23FH
6 – 11	с 64 по 95	с F240H по F25FH
5 – 12	с 96 по 127	с F260H по F27FH
4 – 13	с 128 по 159	с F280H по F29FH
3 – 14	с 160 по 191	с F2A0H по F2BFH
2 – 15	с 192 по 223	с F2C0H по F2DFH
1 – 16	с 224 по 255	с F2E0H по F2FFH

Перемычка установлена: область включена (область квитирована в координаторе).
Перемычка не установлена: область отключена (область квитирована в центральном процессоре).

o Шаг 2:

Центральное устройство выключено. CPU и координатор устанавливаются на предусмотренные для них места в центральном устройстве. После этого подается напряжение питания.

Важно установить CPU так, чтобы между ними не было свободных мест. Поэтому CPU 928 в AG 135YU всегда должен стоять справа от остальных CPU.

Все субмодули памяти (EPROM или RAM) устанавливаются в CPU или в модуль памяти 355 в соответствии с выбранной конфигурацией. Модуль EPROM предварительно должен быть запрограммирован на PG (программаторе).

Можно установить все селекторы режимов работ (на процессорах и координаторе) в STOP, но необязательно. После этого можно включить напряжение питания.

Реакция:

После подачи напряжения питания выдается запрос на перевод в состояние "общий сброс" всех CPU, т.е. красный светодиод "STOP" мигает быстро. Каждый CPU модуль выдает сигнал BASP (в случае, если на координаторе не выбрана функция тестирования и переключатель режимов работы координатора не установлен в положение TEST (см. Раздел 2.3.2): цифровые выходы при этом блокируются. При этом на передней панели горит красный светодиод "BASP".

o Шаг 3:

Установка селектора режимов работы координатора в положение STOP (если это не было сделано в шаге 2) и общий сброс всех используемых CPU.

Общий сброс каждого CPU: необходимо зафиксировать клавишу переключения режимов работы в положении "URLOSCHEN - OVERALL RESET" (общий сброс) и, при указанном положении клавиши переключения режимов работы, перевести селектор режимов работы из положения STOP в положение RUN, а затем снова вернуть в положение STOP.

Реакция

|

Красные светодиоды "STOP" модулей CPU, селекторы режимов работы которых находятся в положении STOP, светятся непрерывно. Каждый центральный процессор выдает сигнал BASP (горят светодиоды "BASP").

o Шаг 4:

Загрузка прикладных программ в установленный модуль памяти RAM.

Если при шаге 2 в CPU не были установлены модули EPROM с записанными в них программами пользователя, то надо теперь загрузить их в RAM.

Комбинация CPU, работающих с модулями EPROM и RAM или обоими типами памяти одновременно (CPU 946/947), может быть любой. Прежде чем многопроцессорная система перейдет в циклический режим работы, необходимо распределить и запрограммировать периферию CPU: должен быть загружен DB1. В CPU 946/947, кроме того, дополнительно надо запрограммировать в DXO многопроцессорный режим (режим 155U).
Остальные программы пользователя могут быть также загружены в CPU на шаге 6. (в циклическом режиме).

Загрузка прикладной программы в CPU 920 (M-процессор) возможна только в том случае, если переключатель режимов работы находится в положении RUN. При переключении из STOP в RUN для загрузки программы в многопроцессорном режиме работы светодиод "STOP" горит постоянно (при работе в однопроцессорном режиме вместо непрерывного свечения выдается медленно мерцающий сигнал, который переходит в непрерывный только после загрузки программы пользователя).

Реакция

Полностью совпадает с реакцией, которая описана для шага 3.

o Шаг 5:

Новый пуск всех CPU 920, 922, 928 и 926/927, новый пуск всех CPU 921 (S-процессоров) без запоминания.

Для выполнения шага 5 последовательно на всех CPU селектор режимов работы, при зафиксированной в положении RESET (сброс) клавише переключения режимов работы, переводится из положения STOP в положение RUN.

Реакция:

На каждом CPU непрерывно светится красный светодиод "STOP", каждым процессором выдается сигнал BASP.

o Шаг 6:

а) Перевод селектора режимов работы координатора из положения STOP в положение RUN.

б) В случае, когда на координаторе установлен тестовый режим работы (смотри Раздел 2.3.2), селектор режимов работы можно также перевести в положение TEST.

Реакция:

В обоих случаях зеленые светодиоды "RUN" всех CPU выдают непрерывный световой сигнал. Все CPU одновременно переходят в циклический режим работы. Сигнал BASP не выдается (светодиод "BASP" выключен).

Если на координаторе не установлен тестовый режим, то реакция на перевод селектора режима работы координатора из позиции STOP в позицию TEST отсутствует.

Замечание по запуску в многопроцессорном режиме

На этапе запуска (обработка OB20, или OB21, или OB22) светодиоды "STOP" и "RUN" CPU 920, 922, 928, 946/947 не выдают световых .сигналов. Светодиод "RUN" выдает непрерывный световой сигнал только при переходе CPU на циклическую обработку программы. В отличие от светодиодов этих процессоров, светодиод "RUN" CPU 921 (S-процессора) выдает непрерывный световой сигнал уже на этапе запуска.

Таким образом, возможна ситуация, когда в обычном режиме работы координатора (см. Раздел 2.3.1) оба светодиода "STOP" и "RUN" CPU 920, 922, 928, 946/947 не выдают световых сигналов, а светодиод "RUN" CPU 921 S-процессора в это время уже выдает непрерывный световой сигнал (на самом деле CPU 921 тоже ждет момента завершения этапа запуска всеми CPU).

Указание

В случае, когда используется координатор С, но при этом к передней панели не подключен интерфейс программатора, координатор С выдает сообщение светодиодом: "IF FAULT" (неисправность интерфейса). В данном случае указанное сообщение можно проигнорировать.

2.3 Режимы работы координатора

2.3.1 Обычный режим работы

Перевод отдельных CPU в режим циклической обработки программы осуществляется, синхронно, если блок данные DXO для CPU 922, 92S, 946/947 не запрограммирован иным образом (смотри Руководства по программированию CPU). Это означает, что все CPU переходят в режим циклической обработки программы одновременно по завершении этапа запуска всех CPU.
В случае, когда селектор режимов работы координатора установлен в положение "RUN", сбой одного CPU (т.е. переход этого модуля в состояние останова) приводит к переходу в состояние останова всех остальных CPU. Красный светодиод "STOP" . центрального процессора (или центральных процессоров), явившегося причиной останова, медленно мигает. Светодиоды "STOP" остальных CPU горят непрерывно.

На CPU, явившемся причиной перехода в состояние останова, возможна выдача светодиодами сигнала индикации ошибки. Одновременно с этим все CPU выдают сигнал BASP.

2.3.2 Тестовый режим работы

Деблокировка тестового режима работы выполняется установкой дополнительной перемычки 3-14 на кодировочном штеккере EP45 координатора A или включением переключателя DIP S1.3 на координаторе C.

В случае перевода селектора режима работы координатора из положения STOP в положение TEST CPU могут переходить в рабочий режим независимо друг от друга. Таким образом, переход в режим циклической обработки программы не является синхронным. Во всех CPU сигнал BASP подавляется (и даже в случае сбоя !).

При сбое одного из находящихся в состоянии RUN центральных процессоров в тестовом режиме работы в состояние останова перейдет только этот, отказавший модуль. Сбой индицируется медленным миганием светодиода "STOP" на центральном процессоре. Таким образом, сбой одного CPU не влияет на остальные.

Важно !
В тестовом режиме работы ни один из CPU не выдает при сбое сигнал BASP. В связи с этим по окончании работ по вводу в эксплуатацию для предотвращения неправильного обслуживания, необходимо отключить тестовый режим работы !

В случае, когда функция тестирования не переведена в активное состояние CPU не будут реагировать на переключение из STOP в TEST.

2.4. PG - мультиплексор координатора C

PG — мультиплексор, который находится на плате координатора C, обеспечивает через последовательный интерфейс, расположенный на передней панели координатора, доступ программатора к любому CPU. Максимальное количество модулей, к последовательным интерфейсам которых обеспечивается центральный доступ, равно 8.

Для работы мультиплексора необходимо программное обеспечение программатора S5—DOS. Для CPU 920 (M-процессора) в настоящее время указанный тип доступа не поддерживается. I

Места установки, обслуживаемые PG-мультиплексором и координатором, указаны в Руководствах по эксплуатации AG 135U и 155U.

Каждому из этих модулей назначается абонентский номер, который должен находиться в диапазоне от 1 до 31 (десятичные числа) . Младший абонентский номер, являющийся базовым адресом, жестко закреплен за установочным местам 11. Назначение абонентских номеров остальным модулям осуществляется в соответствии с приведенными таблицами:

AG S5-135U:

Установ. место	Абонентский номер
11	Базовый адрес
19	Базовый адрес + 1
27	Базовый адрес + 2
35	Базовый адрес + 3
43	Базовый адрес + 4
51	Базовый адрес + 5
59	Базовый адрес + 6
67	Базовый адрес + 7

AG S5-155U:

Установ. место	Абонентский номер
11/27	Базовый адрес
51/67	Базовый адрес + 1
91	Базовый адрес + 2
99	Базовый адрес + 3
107	Базовый адрес + 4
115	Базовый адрес + 5
123	Базовый адрес + 6
131	Базовый адрес + 7

Установка базового адреса выполняется с помощью переключателя DIP S2, который расположен в углублении передней панели координатора C. Базовый адрес формируется как сумма двоичных величин, задаваемых включенным положением указанного переключателя;

	Выключен	Включен	
S 2.1	X		—
S 2.2	X		16
S 2.3	X		8
S 2.4	X		4
S 2.5	X		2
S 2.6		X	1

Q

(X = состояние при поставке) (в данном случае базовый адрес =1)

Важно ! При установке базового адреса необходимо помнить о том, что величина абонентского номера не может превышать 31.

Посредством переключателя S3, который также расположен в углублении лицевой панели координатора C, необходимо выполнить деблокировку установочных мест, обслуживаемых PG-

мультиплексором;

	Выключен	Включен	Установочное место	
S3.1	X		11	Место установки CPU 1
S3.2	X		19	Место установки CPU 2
S3.3	X		27	Место установки CPU 3
S3.4	X		35	Место установки CPU 4
S3.5	X		43	
S3.6	X		51	
S3.7	X		59	
S3.8	X		67	

Состояние при поставке: все установочные места блокированы.

В случае, когда установочные места не заняты или когда обслуживание процессорных модулей должно производиться через разъемы интерфейсов, расположенные на лицевых панелях модулей, необходимо обеспечить блокировку (= отключение) соответствующих установочных мест. При использовании CPU 920 (M-процессора) отведенное под него установочное место блокируется в обязательном порядке.

Передний разъем интерфейса модуля, обслуживаемого через мультиплексор, не должен использоваться для подключения.

Указание:
В микропереключателе DIP SI положению "ВКЛ" соответствует левая позиция, а в переключателях S2 и S3 — правая.

В программаторах должна использоваться операционная система S5--DOS. Необходимое соединение между программатором и абонентом осуществляется с помощью пакета "ВЫБОР ТРАКТА (BUSANWAHL). Более подробно вопрос соединения рассматривается в Руководстве пользователя для соответствующего программатора.

Пример установки переключателей на координаторе C (ПК S5—135U):

	Выключен	Включен		
S2.1	X			
S2.2	X		16	
S2.3		X	8	---> 8
S2.4		X	4	---> 4
S2.5	X		2	
S2.6		X	1	---> 1
			Базовый адрес	13

	Выключен	Включен	Установочные места	Адреса установочных мест
S3.1		X	11	13
S3.2	X			
S3.3	X			
S3.4	X			
S3.5		X	43	17
S3.6		X	51	18
S3.7	X			
S3.8		X	67	20

3 Поиск неисправностей

В Разделе 2 была рассмотрена процедура ввода в эксплуатацию программируемого контроллера AG S5-135U или S5-155U в многопроцессорном режиме работы. Данный Раздел содержит указания, которые должны помочь при поиске и устранении неисправностей.

A … После шага 2

Внешнее проявление неисправности:
Светодиоды "RUN" и "STOP" отдельных CPU светятся, другие CPU выдают запрос на выполнение общего сброса. Все CPU выдают сигнал BASP.

Необходимо проверить:

Правильность установленного на координаторе занятого числа мест установки CPU.

Нет ли свободных мест между CPU?.

B ... После шага 5

Внешнее проявление неисправности 1:

Светодиод "STOP" CPU медленно мигает. В управляющих битах (их можно посмотреть с помощью программатора) присутствует наряду с обычными данными метка ошибки в DB1. Содержимое U-стека не выдается.

Необходимо проверить;

Запрограммирован ли блок данных DB1 для этого CPU.

Внешнее проявление неисправности 2:

После перезапуска:

- CPU 920 (M-процессор) сразу выдает сигнал QVZ (вместе с сигналом BASP), красный светодиод "STOP" медленно мерцает.

- Остальные CPU (других типов) остаются в состоянии останова (светодиод "STOP" горит постоянно) до тех пор, пока на всех процессорах не будет выполнен новый старт (без запоминания).

После этого: - CPU 922 (R-процессор) или CPU 928;
Светодиод "STOP" медленно мигает;

- CPU 921 (S-процессор):
Светодиод "STOP" горит постоянно.

Индикация управляющих битов (ее можно прочитать с помощью программатора) содержит, наряду с обычными записями, ANL-ABB (аварийный останов во время запуска) и метку ошибки DB1.

Последовательность выполнения перезапуска произвольна.

После перевода селекторов режимов работы всех CPU в положение STOP светодиоды "STOP" всех CPU будут гореть постоянно.

При переводе селектора режимов работы из позиции STOP в RUN

- CPU 920 (M-процессор):
  выдает сигнал QVZ, а светодиод "STOP" медленно мигает;

- CPU 922 и 921 (R- и S-процессоры) и CPU 928:
  светодиод "STOP" медленно мигает.

Переключение обратно в позицию STOP на CPU 921 (R-процессоре) не заставит светодиод "STOP" гореть постоянно. Это может произойти только после нового пуска (см выше).

Необходимо проверить:

Подключен ли координатор.

C ... После шага 6

Внешнее проявление неисправности 1:

Все CPU продолжают находиться в состоянии останова.

Необходимо проверить:

Находятся ли селекторы режимов работы всех CPU в позиции RUN?

Последующий запуск отдельных CPU невозможен. Необходимо переключить координатор в STOP, а все CPU в RUN. После этого координатор вновь можно переключить в RUN.

В режиме теста переключение координатора из STOP в TEST приведет к запуску только тех CPU, селекторы которых находятся в положении RUN.

Внешнее проявление неисправности 2:

CPU 921 (S-процессор):
Светодиод "STOP" сразу начинает медленно мигать. Этот режим индикации сохраняется и при переключении из позиции RUN в STOP.

CPU 922 (R-процессор), CPU 928, CPU 946/947;
Светодиод "STOP" сразу начинает медленно мигать. Этот режим индикации сменяется непрерывным свечением при переключении из позиции RUN в STOP.

CPU 920 (M-процессор):
Красный светодиод "STOP" продолжает гореть непрерывно. При переключении координатора из STOP в RUN светодиод "STOP" CPU 920 (M-процессора) начинает медленно мигать.

Необходимо проверить:

Выполнен ли новый пуск без запоминания всех CPU 921 (S-процессоров) и новый пуск всех остальных CPU?

При работе координатора в режиме теста запуск CPU, для которых выполняется новый пуск, можно осуществить переключением координатора из STOP в TEST.

Внешнее проявление неисправности 3:

При использовании координатора C:

На шаге 6 координатор C переключается из STOP в RUN, и все CPU остаются в состоянии останова (непрерывное свечение светодиода "STOP"). На программаторе невозможно просмотреть U-стек. Все CPU выдают сигнал BASP.

Если штекерный переключатель S3.1 координатора C установлен в положение "включено", т.е. задана функция тестирования, то и CPU 921 (S-процессор) и CPU 922 (R-процессор) не будут реагировать на перевод селектора из положения STOP в TEST. В отличии от них M-процессор реагирует выдачей непрерывного сигнала зеленым светодиодом "RUN". Все CPU выдают сигнал BASP. При этом отсутствует возможность управлять работой CPU (а также и координатора) селектором, расположенным на передней панели.

Выключение и последующее включение сетевого напряжения центрального устройства приводит к тому, что светодиоды "STOP" всех; CPU (в том числе, CPU 920, если селектор в позиции STOP) выдают снова непрерывный световой сигнал.

Необходимо проверить:

Правильно ли задано количество CPU на координаторе C.

SIEMENS

SIMATIC S5

Многопроцессорная коммуникация

Программируемый контроллер S5-135U, CPU922 (R-процессор) и CPU928
Программируемый контроллер S5—155U, CPU946/947

Руководство по эксплуатации **Заказной No.C79000—B8500-C468-04**

Оглавление 1

Введение

1.1 Конфигурация
1.2 Принцип передачи данных
1.3 Идентификация передатчика и приемника
1.4 Промежуточное хранение данных
1.5 Запуск системы
1.6 Вызов и глубина вложения организационных блоков специальных Функций OB200 и OB202...OB205
1.7 Параллельная обработка в многопроцессорном программируемом контроллере
1.8 Занимаемые области памяти
1.9 Время выполнения

2 Работа с параметрами

2.1 Обработка выходных параметров
2.1.1 Индикация результатов
2.1.2 Конфликт инициализации, ошибки, предупреждения

3 Функция инициализации <OB200>

3.1 Входные параметры
3.1.1 Режим работы "Автоматический/Ручной"
3.1.2 Количество CPU
3.1.3 Идентификатор блока
3.1.4 Номер блока
Начальный адрес списка назначений
3.2 Входные параметры
3.2.1 Конфликт инициализации
3.2.2 Общий объем назначенной памяти

4 Функция ПЕРЕДАЧА (OB202)

4.1 Входные параметры
4.1.1 CPU—приемник
4.1.2 Идентификатор блока
4.1.3 Номер блока
4.1.4 Номер зоны
4.2 Выходные параметры
4.2.1 Ошибки и предупреждения
4.2.2 Обеспечиваемый объем передачи

5 Функция ТЕСТ ПЕРЕДАЧИ (OB203)

5.1 Входные параметры
5.1.1 CPU-приемник
5.2 Выходные параметры
5.2.1 Ошибки
5.2.2 Обеспечиваемый объем передачи

6 Функция ПРИЕМ (OB204)

6.1 Входные параметры

6.1.1 CPU—передатчик
6.2 Выходные параметры
6.2.1 Ошибки и предупреждения
6.2.2 Обеспечиваемый объем приема
6.2.3 Идентификатор блока
6.2.4 Номер блока
6.2.5 Адрес первого принятого слова данных
Адрес последнего принятого слова данных

7 Функция ТЕСТ ПРИЕМА (OB205)

7.1 Входные параметры

7.1.1 CPU—передатчик

7.2 Выходные параметры
7.2.1 Ошибки
7.2.2 Обеспечиваемый объем приема

8 Использование средств передачи данных

8.1 Вызов организационных блоков специальных функций из Функциональных блоков

8.1.1 Предварительное назначение ресурсов (FB200)
8.1.2 Передача зоны данных (FB202)
8.1.3 Проверка возможности передачи (FB203)
8.1.4 Прием зоны данных (FB204)
8.1.5 Проверка возможности приема (FB205)

8.2 Передача блоков данных

8.2.1 Описание принципа передачи
8.2.2 Передача блоков данных (FB110)
8.3 Расширение области маркеров межпроцессорной связи
8.3.1 Постановка задачи
8.3.2 Решение
8.3.3 Структура данных
8.3.4 Структура программы
8.3.5 Передача из области-слов данных (FB100)
8.3.6 Прием в области слов данных (FB101)
8.3.7 Пример

9 Литература

1 Введение

В состав многопроцессорного программируемого контроллера S5—135U могут входить до четырех CPU одновременно. Для обмена данными между CPU предусмотрен ряд вспомогательных средств, которые могут использоваться по отдельности или в различных комбинациях друг с другом. Такими вспомогательными средствами являются:

1. Средства для передачи маркеров реализующие возможность передачи маркеров межпроцессорной связи в случае, когда в одном процессоре они определены как выходные, а в другом (или в нескольких других) — как входные.

2. Средства для передачи блоков данных, точнее зон данных размером 32 слова данных (64 байта) максимум. Для передачи зон данных предусмотрены следующие Функции:

   ИНИЦИАЛИЗАЦИЯ (OB200): Предварительное назначение ресурсов

   ПЕРЕДАЧА (OB202): Передача зон данных

   ТЕСТ ПЕРЕДАЧИ (OB203): Проверка возможности передачи

   ПРИЁМ (OB204): Прием зоны данных

   ТЕСТ ПРИЁМА (OB205): Проверка возможности приема

   Для использования перечисленных Функций нужно обладать определенным уровнем знаний: необходимо быть знакомым с основами языка программирования STEP—5 и с принципом работы программируемых контроллеров семейства SIMATIC S5. Необходимую информацию по программированию и программируемым контроллерам можно найти в документации, которая указана в перечне литературы.

Маркеры межпроцессорной связи актуализируются операционной системой автоматически. В отличие от маркеров межпроцессорной связи вызов функций ИНИЦИАЛИЗАЦИЯ, ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЕМ и ТЕСТ ПРИЕМА должен оформляться с помощью команд SPA OB или SPB OB как вызов организационных блоков специальных Функций.

В приводимом далее вводном примере рассматриваются наиболее существенные характеристики упоминавшихся организационных блоков специальных Функций (пример представлен в упрощенном виде).

Вводный пример

Необходимо выполнить передачу зоны данных 1 блока данных 17 (смотри ниже) из CPU 3 в CPU2. Таким образом, CPU 3 является передатчиком, а CPU 2 — приемником.

Для решения поставленной задачи в прикладной программе CPU 3 следует подготовить параметры Функции ПЕРЕДАЧА и осуществить вызов этой Функции. Передача параметров должна выполняться через массив маркеров.

CPU 3:

```
L  KB 2          2 — номер CPU—приемника
T  MB 100
L  KB 1          1 — идентификатор блока данных
T  MB 101
L  KB 17         17 — номер блока данных
T  MB 102
L  KB 1          1 — номер зоны данных
T  MB 103
L  KB 100        100 - начало массива параметров <MB100>
SPA OB202        Функция ПЕРЕДАЧА
L  MB 104        индикация . . .
. . . . .          . . . . Обработка
```

Блок данных 17

Заголовок блока <5 слов>

Зона данных О	Слово данных О	
	• • •	
	Слово данных 31	
Зона данных 1	Слово данных 32	
	. . .	===> в CPU 2
	Слово данных 63	
Зона данных 2	Слово данных 64	
	. . .	
Зона данных N 1	. . . |	

В прикладной программе CPU 2 должен выполняться "регулярный" вызов Функции ПРИЕМ, которая осуществляет проверку на передачу данных. Если данные были переданы, то эта Функция размещает переданные данные в блоке данных.

Передатчиком уже заданы тип блока данных (DB,DX), номер и позиция (адрес) в блоке данных. В связи с этим, указанные атрибуты в CPU—приемнике всегда будут точно совпадать с атрибутами CPU—передатчика!

cpu 2:

```
L  KB 3       3 — номер CPU—передатчика

T  MB70
L  KB 70      70 — начало массива параметров (MB70)
SPA OB204     функция ПРИЕМ

L  MB 72      индикация . . . .

. . . . .          . . . . Обработка
```

Блок данных 17

Заголовок блока (5 слов)

Зона данных О	Слово данных О	
	. . .	
	Слово данных 31	
Зона данных 1	Слово данных 32	
	. . .	<== от CPU 3
	Слово данных 63	
Зона данных 2	Слово данных 64	
	• • •	
Зона данных М	. . .	

1.1 Конфигурация:

1 AG (программируемый контроллер) S5—135U или AG 155U

Имеют периферийную шину SIMATIC S5 и следующие компоненты;

- 1 Координатор 923C (KOR C)

В координаторе имеется четыре буфера связи. Это области памяти емкостью 1024 байта каждый. Все буферы связи занимают одно и то же адресное пространство — с F400H по F7FFH. Выбор (адресация) "актуального" буфера осуществляется посредством так называемого регистра выбора или идентификации) (т.е. аналогично выбору микросхемы). За четырьмя буферами связи KOR C жестко закреплены номера: 252, 253, 254 и 255. Перечисленные номера предназначены для осуществления выбора конкретного буфера связи и используются в рассматриваемом здесь многопроцессорном .сопряжении.

- 2-4 CPU

Для AG 135U, например, CPU 922 (R—процессоры), CPU 928 или CPU 920 (M—процессоры).

Для AG 155U CPU 946/947, CPU 922 (R-процессоры), CPU 928 или CPU 920 (M-процессоры).

Эти CPU могут в произвольной комбинации обмениваться друг с другом данными, одновременно с процессом обмена допускается, без каких-либо ограничений, использование "управляющих блоков" (управляющие блоки также используют адресацию буферов связи).

CPU 921 (S-процессоры), которые дополнительно подключены к этой же раме, не могут принимать участия в описываемом здесь многопроцессорном сопряжении. Управляющие блоки S—процессора нельзя вызывать во время обработки управляющим блоков R—, M—процессоров и CPU 928, а также во время многопроцессорного сопряжения (т.е. межпроцессорного обмена данными) между R-, M-процессорами н CPU 928, CPU 946/947. Необходимо отметить, что межпроцессорную связь можно обеспечить в любом случае, используя для этой цели маркеры межпроцессорной связи.

1.2 Принцип передачи данных

Для передачи данных в CPU-передатчике необходимо перевести в активное состояние Функцию ПЕРЕДАЧА, а в CPU—приемнике — Функцию ПРИЕМ.

При этом осуществляется пересылка расположенных друг за другом слов данных из блока данных DB или DX, которые находятся в CPU-передатчике. Пересылка в CPU—приемник выполняется через координатор C. В CPU—приемнике слова данных размещаются в блоке DB или DX под номерами н по адресам, которые идентичны соответствующим атрибутам в CPU—передатчике. Таким образом, при передаче данных обеспечивается копирование 1:1.

Пример

	Передаваемые данные в CPU—передатчике	Принимаемые данные в CPU—приемнике
Блок данных Адрес слова данных	DB 17 с DW32 по DW63	DB17 с DM32 по DW63

Объем данных, который, как правило, передается посредством Функции ПЕРЕДАЧА и Функции ПРИЕМ, равен 32 словам.

В случае, когда размер блока (без учета заголовка) не кратен 32 словам, в качестве последней зоны данных передается, в виде исключения, группа данных, состоящая менее чем из 32 слов.

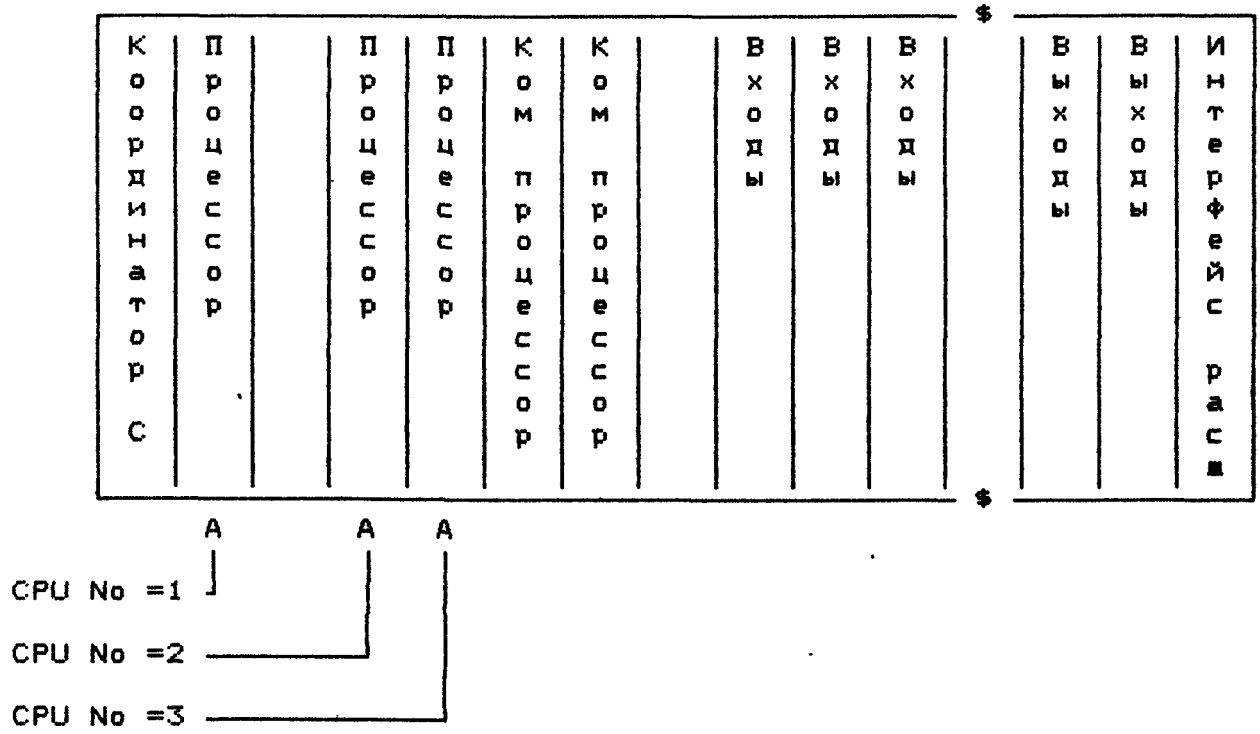
Блок данных в CPU—приемнике может быть больше или меньше блока данных CPU—передатчика. Существенно наличие в принимающем блоке данных всех слов данных, которые были переданы при выполнении Функции ПЕРЕДАЧА. Отсутствие переданных данных в принимающем блоке обрабатывается Функцией ПРИЕМ как состояние ошибки.

1.3 Идентификация передатчика и приемника

CPU нумеруются следующим образом: CPU, который находится в крайнем левом установочном месте, назначается номер 1. Номер каждого следующего CPU больше на единицу номера CPU, расположенного слева от него.

Пример:

AG 135U/AG 155U:



1.4 Промежуточное хранение данных

Время цикла CPU определяется совокупностью выполняемых задач и характеристиками производительности самого CPU. На время цикла CPU влияют:

— Объем отдельных программных частей;
— Количество выполнении отдельных частей программы (многократный вызов, циклы);
— Количество запрограммированных регуляторов.

Кроме того, на время цикла CPU влияют условные вызовы блоков <SPB PBxx), прерывания (OB2, обработка с управлением по прерываниям) и т.д.

Итак, в многопроцессорном режиме работы циклическая обработка программы в каждом отдельно взятом CPU происходит асинхронно по отношению к циклической обработке программы в других CPU.

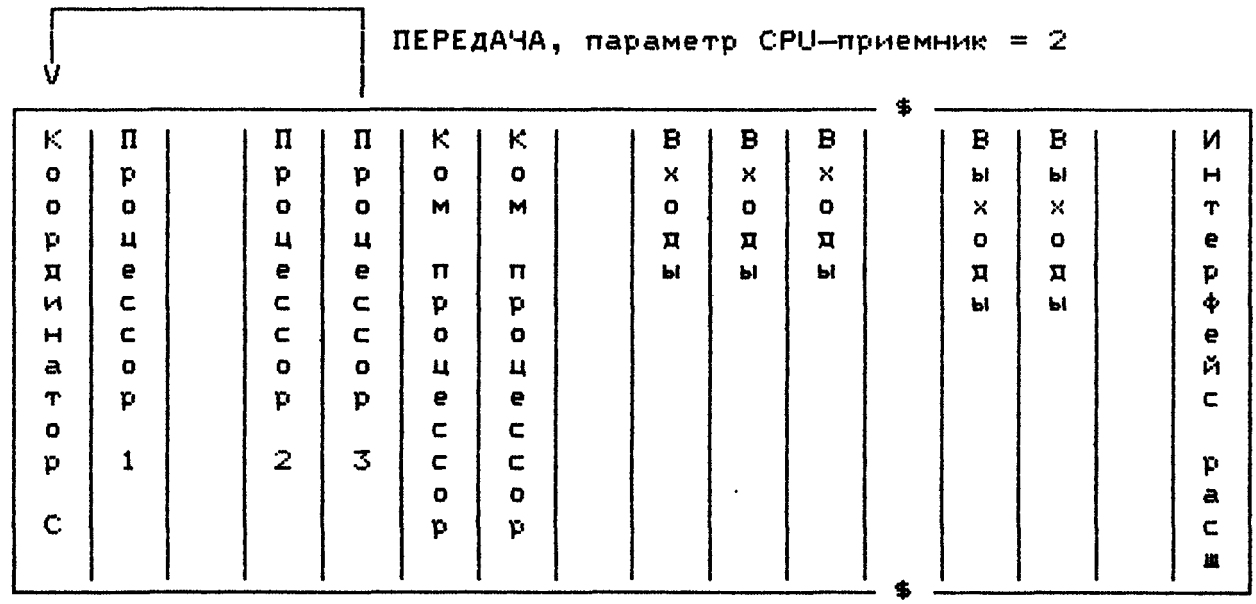
В отличие от циклической обработки программы, обработка программы с управлением по прерываниям по времени выполняется периодически по тактирующему сигналу, в R—процессоре и в CPU '928 такая обработка выполняется через каждые 100мс. Взаимное смешение тактирующего сигнала CPU относительно другого CPU может достигать 100мс.

Асинхронность процессов обусловливает необходимость промежуточного хранения передаваемых данных в координаторе С.

Источник данных и назначение определяются номером "собственного" CPU и номером приемника (функция ПЕРЕДАЧА) или номером передатчика (Функция ПРИЕМ).

Пример: Передача данных из CPU 3 в CPU 2

1. Шаг :



2. Шаг:



1. В промежуточной памяти используется алгоритм FIFO. Таким образом, последовательность приема совпадает с последовательностью Передачи. Это справедливо для каждого отдельного участка связи (идентифицируемого парой CPU — CPU-передатчиком и CPU-приемником). На работу промежуточной памяти, закрепленной за отдельным участком связи, другие соединения влияния не оказывают.

2. Промежуточная память имеет буферное питание от батареи, благодаря этому выполнение Функции "автоматический перезапуск после сбоя сетевого напряжения питания" может осуществляться без каких-либо ограничений. Программируемый контроллер обрабатывает ситуацию отказа сетевого питания в ходе передачи данных без потери информации.

Объем информации координатора С равен 48 блокам памяти по 32 слова в каждом. Функция ИНИЦИАЛИЗАЦИЯ выполняет закрепление этих блоков памяти за отдельными участками связи.

Каждый блок памяти (размер блока памяти всегда 32 слова) используется для хранения только одной зоны данных (размер зоны данных — от одного до 32 слов данных). Зона данных записывается в блок памяти Функцией ПЕРЕДАЧА и выбирается из него Функцией ПРИЕМ.

Количество блоков памяти, отводимых для участка связи, находится в прямой зависимости от параметров — обеспечиваемый объем передачи (функция ПЕРЕДАЧА и Функция ТЕСТ ПЕРЕДАЧИ) и обеспечиваемый объем приема (Функция ПРИЕМ и Функция ТЕСТ ПРИЕМА).

Обеспечиваемый объем передачи — это величина, равная количеству свободных, в определенный момент времени, блоков памяти из числа блоков памяти, зарезервированных для некоторого участка передачи.

Обеспечиваемый объем приема — это величина, равная количеству занятых к определенному моменту времени блоков памяти из числа блоков памяти, зарезервированных для некоторого участка приема.

Сумма обеспечиваемых объемов передачи и приема в любой момент равна количеству блоков памяти, закрепленных за участком связи.

Пример;

Участку связи "от CPU 3 к CPU 2" посредством Функции ИНИЦИАЛИЗАЦИЯ назначены семь блоков памяти.

Передатчик: CPU 3		Приемник: CPU 2	
Процесс	Объем передачи (свободные блоки памяти)	Объем приема (занятые блоки памяти)	Процесс
ПЕРЕДАЧА одной зоны данных A	7	0	ИНИЦИАЛИЗАЦИЯ
	6	1	
ПЕРЕДАЧА четырех зон данных (B,C,D,E)	2	5	
ПЕРЕДАЧА четырех зон данных (F,G,H,I)	4	3	ПРИЕМ двух зон данных A, B
	0	7	ПРИЕМ пяти зон данных <D,E,F,G)
ПЕРЕДАЧА двух зон данных (K, L)	5	2	
	5	2	ПРИЕМ двух зон данных H,I

Указание 1:

ПЕРЕДАЧА/ПРИЕМ n зон данных означает, что осуществляется n следующих друг за другом вызовов соответствующей Функции.

Указание 2:

Для большей наглядности в этом примере сначала выполняется или ПЕРЕДАЧА или ПРИЕМ. Необходимо при этом отметить, что одновременное выполнение ПЕРЕДАЧИ (CPU 3) и ПРИЕМА (CPU 2) является не только возможным, но и целесообразным решением (см. Раздел "Параллельная обработка в многопроцессорном программируемом контроллере"). В рассмотренном примере во время передачи зон данных K и L осуществляется прием зон H и I.

Приведенный пример также иллюстрирует алгоритм очереди, используемый в промежуточной памяти: Первые переданные зоны данных (A,B,C ...) были приняты первыми (A,B,C ...).

Выводы;

Промежуточное хранение данных в координаторе C реализовано для согласования асинхронных процессов, выполняемых CPU—передатчиком и CPU—приемником, с рабочей скоростью процессоров.

Объем промежуточной памяти ограничен, в связи с этим приемник должен "часто" и "регулярно" проверять, занесены ли данные в память (Функция ТЕСТ ПРИЕМА), (обеспечиваемый объем приема больше нуля) и осуществлять попытки выборки данных, хранящихся в памяти (Функция ПРИЕМ). Имеет смысл повторять вызовы Функции ПРИЕМ до тех пор, пока обеспечиваемый объем ПРИЕМА не станет равным нулю. При таком подходе переданные данные не будут долго холиться в промежуточной памяти и, сохранив актуальность, будут приняты CPU—приемником. Кроме того, в этом случае освобождаются блоки памяти (возрастает обеспечиваемый объем приема) и предотвращается блокировка передатчика т.е. предотвращается ситуация, когда обеспечиваемый объем передачи "исчерпан" (равен нулю).

Следовательно, если нулевой обеспечиваемый объем приема является отображением идеального состояния (все переданные данные получены приемником), то нулевой обеспечиваемый объем передачи, напротив, свидетельствует о наличии ошибок проектирования:

— Слишком часто выполняется вызов Функции ПЕРЕДАЧА;

— Недостаточно часто выполняется вызов Функции ПРИЕМ;

— Участку передачи назначено слишком мало блоков памяти. Объема промежуточной памяти недостаточно для компенсации временного дисбаланса между частотой передачи и частотой приема данных.

1.5 Запуск системы

При многопроцессорном сопряжении необходимо,, чтобы у всех задействованных в сопряжении CPU процесс перехода из состояния STOP (останов) в состояние RUN (работа) выполнялся единообразно, то есть, либо для всех этих CPU должен выполняться новый пуск либо перезапуск.

Единообразный вид запуска, по крайней мере хотя бы тех CPU, которые задействованы в сопряжении, должен обеспечиваться посредством (смотри перечень литературы)

- o соответствующего обслуживания (переключатель на лицевой панели, программатор)
- o соответствующей подготовки параметров (DXO)

и/или o соответствующего программирования (с использованием организационного блока специальной функции OB223 "Останов из-за нарушения единообразия запуска в многопроцессорном режиме работы").

Новый пуск:

В пользовательском интерфейсе OB20, учитывающем специфику запуска, один из должен с помощью Функции ИНИЦИАЛИЗАЦИЯ выделить промежуточную память (в координаторе C). При этом находящиеся в памяти данные будут уничтожены. Затем, т.е. еще в процессе запуска, в отдельных CPU может осуществляться вызов Функций ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЕМ, и ТЕСТ ПРИЕМА. Необходимо выполнить программирование таким образом, чтобы вызов перечисленных Функций в отдельных CPU происходил только после корректной инициализации промежуточной памяти в координаторе KOR. После завершения запуска, т.е. в режиме RUN (работа), прикладная программа обрабатывается с самого начала, т.е. обработка начинается с первой команды блока OB1 или FB0.

Перезапуск:

В интерфейсах пользователя OB21 и OB22, учитывающих специфику запуска, нельзя использовать Функцию ИНИЦИАЛИЗАЦИИ. Вызов Функций ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ ПРИЕМ и ТЕСТ ПРИЕМА может привести к осложнениям в работе (смотри указания в Разделе "Вызов и вложенность организационных блоков специальных Функций"). После завершения запуска, т.е. в режиме RUN (работа), прикладная программа обрабатывается не с самого начала, а, в отличии от нового старта, обработка программы продолжается с места прерывания. . Точка прерывания может, например, находиться в теле Функции ПЕРЕДАЧА.

1.6 Вызов и глубина вложения организационных блоков специальных Функций OB200 и OB202... OB206.

Наиболее простым является следующий способ:

Вызов Функции ИНИЦИАЛИЗАЦИЯ только в организационном блоке нового старта OB20;
Вызов Функций ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЕМ, ТЕСТ ПРИЕМА только при циклической обработке программы или только при обработке программы прерывания по времени.

Указание:

Возможно, что выполнение одной из Функций ИНИЦИАЛИЗАЦИЯ, ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЕМ, ТЕСТ ПРИЕМА будет, прервано в соответствии с заданными параметрами DXO <прерывание на- границах команд) и режимом обработки программы (перезапуск, обработка прерывания, например OB26, при превышении времени цикла). Если во включенном в точку прерывания пользовательском интерфейсе (например OB13 при разрешении прерывания на границах команд или OB22 после сбоя сетевого питания) также содержится одна из Функций ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЕМ, ТЕСТ ПРИЕМА, то эти Функции идентифицируют ситуацию недопустимого вызова (дублирование вызова) и сообщают об этом выдачей индикации ошибки (ошибка No.3, смотри далее).

1.7 Параллельная обработка в многопроцессорном программируемом контроллере.

После завершения предварительного выделения промежуточной памяти (Функция ИНИЦИАЛИЗАЦИЯ) на всех CPU одновременно и параллельно, в любой комбинации и с любыми параметрами могут обрабатываться Функции ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЕМ, ТЕСТ ПРИЕМА.

Если ограничиться рассмотрением одного единственного участка связи <от CPU S до CPU E), то можно отметить возможность одновременной обработки Функции ПЕРЕДАЧА (CPU S) и Функции ПРИЕМ (CPU E). В то время, когда CPU S пересылает следующие зоны данным в координатор C, то CPU E может осуществлять прием (выборку) зон данных, которые уже находятся в промежуточной памяти координатора C.

1.8 Занимаемые области памяти

Организационные блоки специальных Функций OB200 и OB202...OB205 не используют рабочую область (например, для промежуточного хранения данных) и не вызывают блоки данных. Естественно, эти блоки обращаются к областям, содержащим параметры. При этом изменяются только те параметры, которые отмечены как выходные. Кроме того, эти блоки оказывают влияние на индикацию результата (ANZ1, VKE и т.д. (смотри Раздел "Обработка выходных параметров" и Раздел "Индикация результатов"). Организационные блоки OB200 и OB202...OB205 также не изменяют содержимое аккумуляторов и регистров.

CPU 922, CPU 928: Содержимое аккумуляторов АКК 1...4 не зависит от 0B специальных Функций межпроцессорной связи

CPU 946/947: Содержимое регистров, а также АКК 1...3 остается без изменений, но меняется содержимое АКК 4.

Мс = миллисекунды
Мкс = микросекунды

0B специальных Функций I Время выполнения

Блок/ Название	Функция	R—процессор	CPU 928	CPU 946/947
0B 200 / ИНИЦИА-ЛИЗАЦИЯ	Предварительное задание ресурсов	230мс	130мс	128мс
0B 202 / ПЕРЕДАЧА	Передача одной зоны данных (32 слова данных)	806мкс (294мкс основная нагрузка + 16 МКС /слово) 118 мкс при выдаче предупреждения	666мкс (250мкс основная нагрузка + 13мкс /слово) 115 мкс при выдаче предупреждения	762мкс (426мкс основная нагрузка + 21мкс /двойное слово) 243 мкс при выдаче предупреждения
0B 203 / ТЕСТ ПЕРЕДАЧИ	Проверка воз-можности передачи	72мкс	50мкс	207мкс
0B 204 / ПРИЕМ	Прием одной зоны данных (32 слова данных)	825мкс (281мкс основная нагрузка + 17мкс /слово) 115мкс при выдаче пре-дупреждения	660мкс(244мкс основная нагрузка + 13мкс /слово) 98 мкс при выдаче предупреждения	772мкс (421мкс основная нагрузка + 22мкс /двойное слово) 243 мкс при выдаче предупреждения
0B 205 / ТЕСТ ПРИЕМА	Проверка воз-можности приема	70мкс	48мкс	223мкс

"Время выполнения" — это время обработки организационных блоков специальных Функций. Время (интервал времени) между вызовом блока и завершением выполнения блока может существенно превышать это время выполнения в случае прерывания обработки блока от процесса, имеющего более высокий приоритет (например, актуализация ячеек таймеров, обработка регуляторов и т.д.).

Время выполнения, приведенное в таблице, получено в предположении, что из четырех установленных CPU доступ к шине периферии SIMATIC S5 осуществляет только тот CPU, для которого измеряется время выполнения. При интенсивном использовании шины другими CPU, время выполнения, особенно при приеме и передаче, возрастает.

Важной характеристикой производительности участка связи (от CPU S до CPU E) является общее время передачи данных, которое складывается из следующих трех компонентов;

— Время передачи (см. Время выполнения)
— Длительность промежуточного хранения (в координаторе KOR C)
— Время приема (см. Время выполнения)

Следовательно, время, в течение которого передаваемые данные будут "в пути", в значительной степени определяется длительностью промежуточного хранения^ т.е. определяется структурой прикладной программы (смотри Раздел "Промежуточное хранение данных").

2 Работа с параметрами

Непосредственно "сами" параметры находятся в массиве параметров, который располагается в области маркеров. Максимальный размер массива параметров — 10 байтов. Младшее слово аккумулятора указывает на первый байт массива параметров, это слово должно содержать значение в диапазоне от 0 до 246.

Массив параметров делится ' на область для входных параметров, область для выходных параметров и, возможно, на дополняющую область (которая не используется:).

Входные параметры:

Входные параметры полностью или частично считываются и обрабатываются Функциями, доступа к записи нет.

Выходные параметры:
Выходные параметры полностью или частично перезаписываются Функциями, доступа к чтению нет.

Не используемая область:

Эти байты (байты маркеров — MB) не считываются и не перезаписываются Функциями.

Пример: Массив параметров Функции ПРИЕМ <OB204>.

MB n+O CPU—передатчик	Входной параметр
MB n+1	Не используется
MB n+2 Ошибка/предупреждение	Выходной параметр
MB n+3 Обеспечиваемый объем приема	"
MB n+4 Идентификатор блока	"
MB n+5 Номер блока	"
MB n+6 Адрес первого принятого слова данных	"
MB n+7: ——————"——————	"
MB n+8: Адрес последнего принятого слова данных	"
MB n+9: ——————"——————	"

Этот пример наглядно показывает, что указатель массива параметров не может иметь значение, превышающее <MB> 246, так. как в противном случае происходит выход 10-байтового массива параметров за границу области маркеров <MB255>.

2.1 Обработка выходных параметров

Выходные параметры - это данные, предоставляемые Функцией прикладной программе для последующей обработки. Выходные параметры содержат, в частности, информацию о возможности выполнения Функции. В случае, когда выполнение Функции невозможно, выходные параметры содержат указание причины аварийного завершения Функции.

2.1.1 Индикация результатов

Функции ИНИЦИАЛИЗАЦИЯ, ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЕМ, ТЕСТ ПРИЕМА оказывают влияние на индикацию результатов <'смотри руководство по программированию R—процессора или CPU 928, Раздел "Общие указания по STEP—5").

— Виты OV и OS (индикация слов) всегда находятся в сброшенном состоянии,
— Биты OR, STA, ERAB (индикация битов) всегда находятся в сброшенном состоянии,
— VKE, ANZO и ANZ1 дают информацию о том, была ли Функция выполнена правильно и полностью.

VKE=0; Функция отработана правильно и полностью

VKE=1: Аварийное завершение выполнения функции:

Возможно, что указатель массива параметров в области маркеров содержит недопустимое значение, т.е. младшее слово аккумулятора содержит величину большую, чем 246. Материал следующих разделов излагается в предположении, что указатель массива параметров содержит допустимое значение.

Первый байт выходных параметров содержит подробную информацию о причине аварийного завершения.

ANZO=1 Дополнительная индикация ошибки (номер ошибки 1...9)

ANZ1=1 Дополнительная индикация предупреждения (номер предупреждения — 1 или 2)

Состояние	Индикация VKE	ANZ0	ANZ1	Типичные команды обработки индикации
Функция выполнена корректно И полностью	0	0	0	SPB=
Обработка Функции прервана, недопустимое значение указателя массива параметров	1	0	0	SPB=
Обработка Функции прервана из-за конфликта инициализации	1	0	0	SPB=
Обработка Функции прервана из-за ошибки	1	1	0	SPB=, SPM=
Обработка Функции прервана из-за предупреждения	1	0	1	SPB=, SPP=

2.1.2 Конфликт инициализации, ошибки, предупреждения

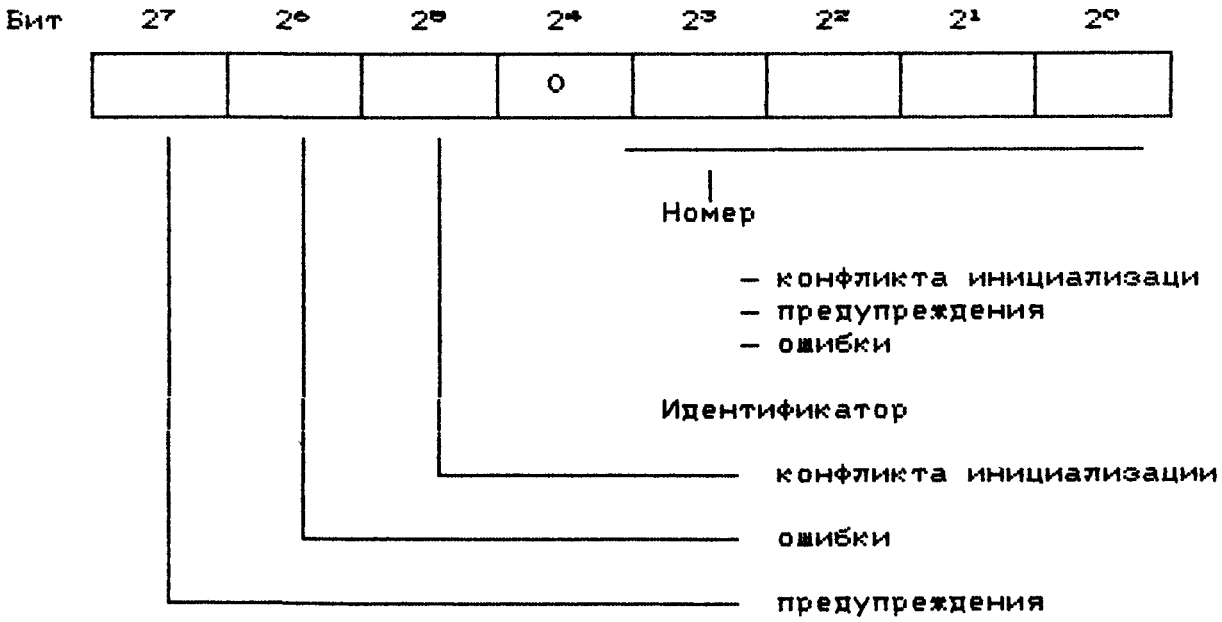
Первый байт массива выходных параметров также сигнализирует о том, была ли соответствующая Функция обработана корректно и полностью. Причина аварийного завершения обработки Функции отображается так же подробно, как и при индикации результата.

При выполнении упоминавшегося ранее соглашения о том, что хотя бы указатель массива параметров содержит правильное значение, значение первого байта массива выходных параметров всегда будет нести смысловую нагрузку.

В случае корректного и полного выполнения Функции все биты первого байта массива выходных параметров будут сброшены в ноль <=0>, а все остальные выходные параметры также будут иметь определенный смысл.

При завершении обработки Функции с выдачей предупреждения (бит 7 = 1) полезная информация будет содержаться только в индикации обеспечиваемого объема ПРИЕМА/ПЕРЕДАЧИ, все остальные выходные параметры (при их наличии) не изменяется.

В случае аварийного завершения обработки Функции из-за ошибки <бит 6 == 1> или из-за конфликта инициализации <бит 5 = 1> все остальные выходные параметры не изменяются.



Обработка

Идентификаторы в битах 5, 6 и 7 указывают, в частности, какой смысл несут номера, занимающие биты с 0 по 3.

Наряду с побитовой обработкой весь байт можно интерпретировать как беззнаковое число с Фиксированной запятой (это число будет указываться далее в скобках под соответствующим номером):

Диапазон значений	Соответствующее состояние
0	Функция обработана корректно и полностью
33... 42	Аварийное завершение обработки Функции из-за конфликта инициализации
65... 73	Аварийное завершение обработки Функции из-за ошибки
129...130	Аварийное завершение обработки Функции из-за

Перечисленные ошибки или конфликты инициализации обычно идентифицируются Функциями в порядке возрастания номеров ошибок или конфликтов инициализации. Это означает, что в случае, например, когда Функция ПЕРЕДАЧА выдает сообщение об ошибке и не "запускается", внесение изменений в программу и/или параметры, приведет к устранению одной ошибки (из нескольким), которой соответствует максимальный номер ошибки.

Конфликт инициализации

Конфликт инициализации возможен только при использовании Функции ИНИЦИАЛИЗАЦИЯ. В случае конфликта инициализации необходимо внесение изменений в программу и/или в параметры.

Номера конфликтов инициализации

1.(33) Нет или недостаточно буферов связи (номера с 252 по 255), необходимых для многопроцессорного сопряжения.

2.(34) Дефект буферов связи (номера с 252 по 255:), необходимых для многопроцессорного сопряжения.

3.(35) Недопустимый параметр "Автоматический/ручной режим работы"
Необходимо различать следующие варианты:
— Значение идентификатора "Автоматический/ручной" меньше 1
— Значение идентификатора "Автоматический/ручной" больше 2

4.(36) Недопустимый параметр "Количество CPU" <36>
Необходимо различать следующие варианты:
— Количество CPU меньше 2
— Количество CPU больше 4

5.(37) Недопустимый параметр "Идентификатор блока"

Необходимо различать следующие варианты:
— Значение идентификатора блока меньше 1
— Значение идентификатора блока больше 2

6.(38) Недопустимый параметр "Номер блока" — соответствующий блок является блоком специального назначения

Необходимо различать следующие варианты:
— Если идентификатор блока = 1, то DBO, DB1, DB2
— Если идентификатор блока = 2, то DXO

7.(39) Ошибка в параметре "Номер блока" - блока с указанными параметрами не существует.

8.(40) Слишком большое значение параметра "Начальный адрес списка назначений" или недостаточный размер блока данных.

9.(41) В блоке данных некорректно построен список назначений

10.(42) Суммарное количество выделяемых блоков памяти превышает 48

Ошибки

В случае выявления ошибки необходимо внести изменения в программу и/или в параметры.

Номер ошибки:

1.(65) Недопустимый параметр "CPU-приемник" (ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ)

Необходимо различать следующие варианты!
— Номер CPU—приемника превышает число 4
— Номер CPU-чприемника меньше 1
— Номер CPU—приемника совпадает с "собственным" номером.

2.(66) Недопустимый параметр "CPU-передатчик" (ПРИЕМ, ТЕСТ ПРИЕМА).

Необходимо различать следующие варианты:
— Номер CPU-ттередатчика превышает число 4
— Номер CPU—передатчика меньше 1
— Номер CPU—передатчика совпадает с "собственным" номером.

3.(67) Ошибка при вызове организационного блока специальной Функции (ПЕРЕДАЧА, ПРИЕМ, ТЕСТ ПЕРЕДАЧИ, ТЕСТ ПРИЕМА)

Необходимо различать следующие варианты:

— Ошибка является ошибкой-следствием, т.к. не был осуществлен вызов Функции ИНИЦИАЛИЗАЦИЯ или же вызов Функции ИНИЦИАЛИЗАЦИЯ привел к конфликту инициализации.

— дублирование .вызова: Вызов одной из Функций — т.е. Функций ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЕМ или ТЕСТ ПРИЕМА (смотри Раздел "Вызов и глубина вложения организационных блоков специальных Функций").

— Ошибка в "собственном" номере CPU (разрушены системные данные). После выключения и включения сетевого питания системная программа вновь генерирует номер CPU.

4. (68) Ошибка в управляющих данных (управление очередью) выбранного участка связи.

Посредством Функции ИНИЦИАЛИЗАЦИЯ необходимо заново выполнить распределение промежуточной памяти в координаторе С (ПЕРЕДАЧА, ПРИЕМ, ТЕСТ ПРИЕМА, ТЕСТ ПЕРЕДАЧИ).

5.(69) Недопустимый параметр "Идентификатор блока" (ПЕРЕДАЧА) или недопустимый идентификатор блока, поступившего с передатчика (ПРИЕМ).

Необходимо различать следующие варианты;

— Идентификатор блока меньше 1

— Иденти4'икатор блока больше 2

6.(70) Недопустимый параметр "Номер блока" (ПЕРЕДАЧА) или недопустимый номер (70) блока, поступившего с передатчика(ПРИЕМ).

Ошибка заключается в том, что этот блок является блоком специального назначения.

Необходимо различать следующие варианты:

— Если идентификатор блока = 1, то DBO, DB1, DB2

— Если идентификатор блока = 2, то DXO

7.(71) Ошибка в параметре "Номер блока" (ПЕРЕДАЧА) или ошибка в номере блока, поступившего с передатчика (ПРИЕМ). Блок с указанными параметрами не существует.

8.(72) Ошибка в параметре "Номер зоны" (ПЕРЕДАЧА) недостаточный размер блока данных или слишком большой номер зоны.

9.(73) Размер блока данных недостаточен для приема зон данных, поступивших с передатчика (ПРИЕМ).

Предупреждение

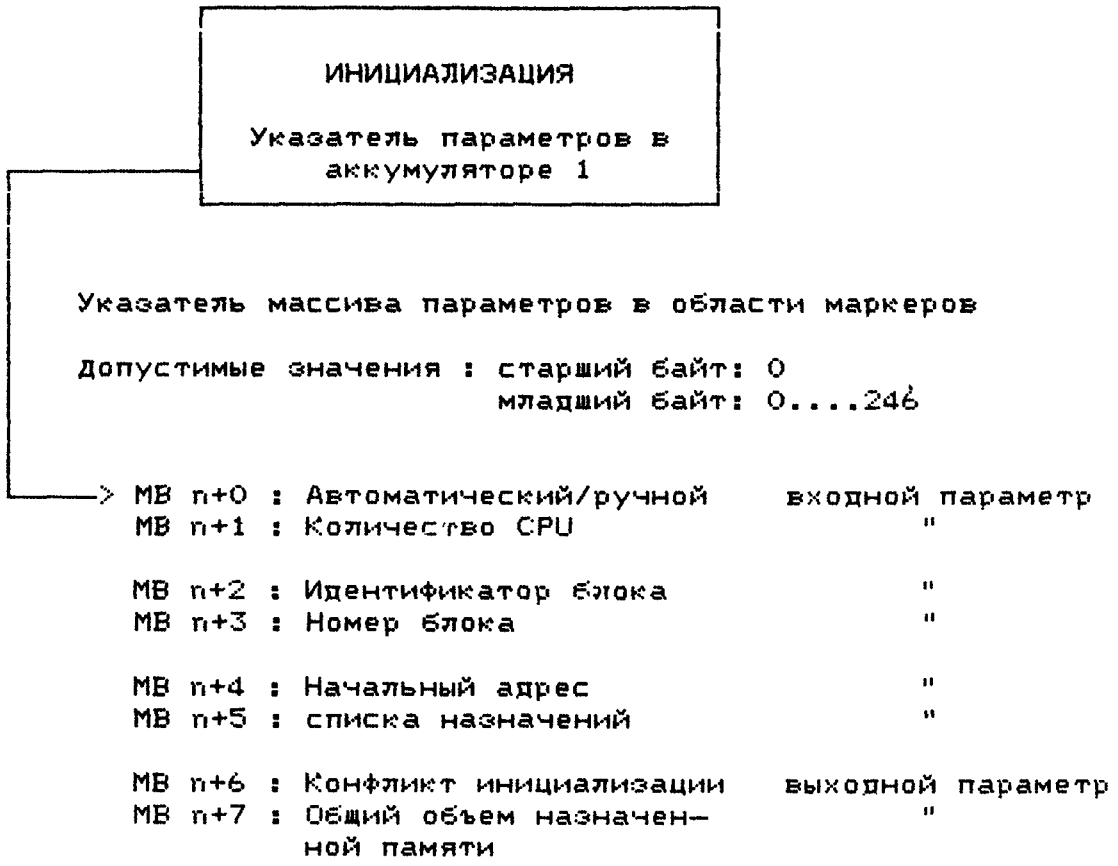
Предупреждение выдается в случае, когда Функция не может быть выполнена. Следует повторить вызов Функции, например, в следующем цикле.

Номер предупреждения:

1.(129) Функция ПЕРЕДАЧА не может выполнить передачу данным, так как обеспечиваемый объем передачи уже при вызове Функции был равен нулю.

2.(130) Функция ПРИЕМ не может выполнить прием данных так как уже при вызове Функции обеспечиваемый объем приема был равен нулю.

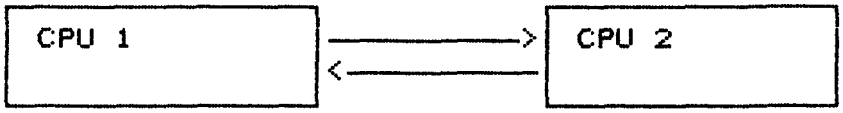
2 Функция ИНИЦИАЛИЗАЦИЯ (OB200)



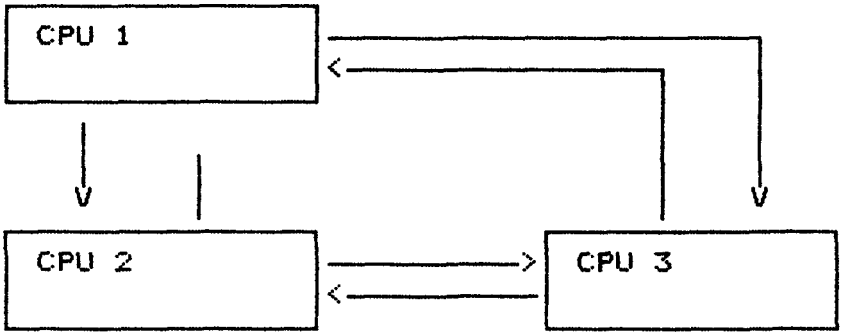
Для того, чтобы можно было выполнить передачу данных из CPU S в CPU E, необходимо обеспечить временное хранение их в промежуточной памяти. Функция ИНИЦИАЛИЗАЦИЯ Формирует для этой цели промежуточную память в координаторе С.

Объем памяти измеряется и выражается в блоках (размером в 32 слова).
Блок памяти (его размер всегда равен 32 словам) отводится только под одну единственную зону данных (размер зоны данных — от одного до 32 слов данных). Зона данных загружается в блок памяти организационным блоком Функции ПЕРЕДАЧА и выбирается из памяти организационным блоком Функции ПРИЕМ.

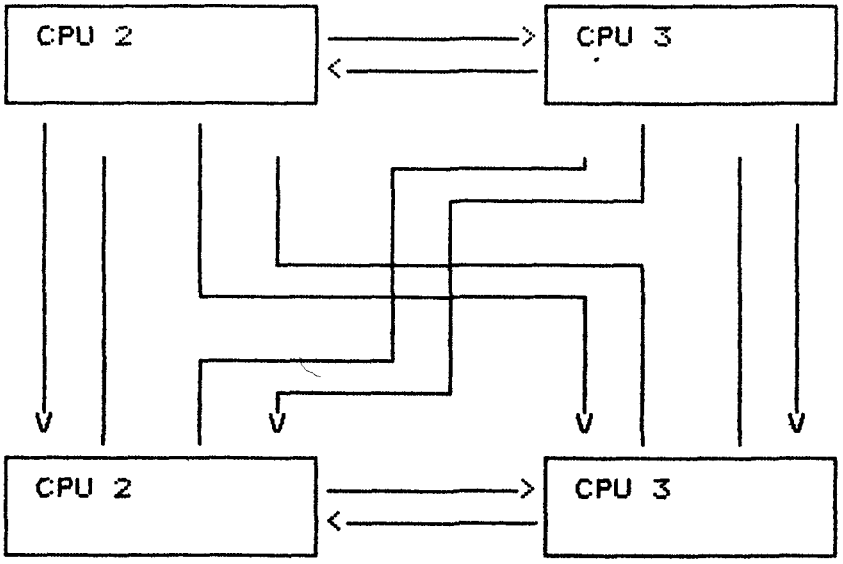
В случае использования двух CPU получаются два участка связи (направления передачи, каналы):



В случае использования трех CPU, получается шесть участков связи:



В случае использования четырех CPU, получается двенадцать участков связи:



Посредством Функции ИНИЦИАЛИЗАЦИЯ определяется, как 48 имеющихся блоков памяти будут распределены между участниками связи. При этом распределение может выполняться максимум между двенадцатью участниками связи.

Это означает, что любому из возможных участников связи, идентифицируемому параметрами "CPU—передатчик" и "CPU—приемник", выделяется определенный объем памяти.

Вызов Функций ПЕРЕДАЧА/ПРИЕМ/ТЕСТ ПЕРЕДАЧИ/ТЕСТ ПРИЕМА в CPU возможен только после того, как на одном из CPU .будет вызвана и полностью выполнена функция ИНИЦИАЛИЗАЦИЯ.

В случае нескольких следующих друг за другом вызовов Функции ИНИЦИАЛИЗАЦИЯ действительным значениями параметров будут те значения, которые задаются в последнем вызове. Во время обработки Функции ИНИЦИАЛИЗАЦИЯ одним из CPU на остальных CPU не должен осуществляться вызов каких-либо Функций, в том числе и Функции ИНИЦИАЛИЗАЦИЯ.

3.1 Входные параметры

3.1.1 Режим работы "Автоматический/ручной"

1: Автоматический
2: Ручной
0, 3—255: Недопустимые значения, использование которых приводит к конфликту инициализации.

Режим работы "Автоматический"

В случае выбора режима работы "Автоматический" имеющиеся блоки памяти распределяются равномерно в соответствии с количеством CPU.

Количество CPU	Количество участков связи	Количество блоков памяти, приходящихся на участок связи
2	2	24
3	6	8
4	12	4
0, 1, 5-255	Недопустимые значения инициализации, их использование приводит к конфликту	

Режим работы "Ручной"

В случае выбора режима работы "Ручной" пользователь должен подготовить в блоке данных список назначений, в котором согласно определенной схеме 48-ми (или меньшему количеству) имеющихся блоков памяти назначаются максимум 12 участков связи.

Использование этого режима наиболее целесообразно при неравномерном обмене данными между CPU. Так, например, в описываемом здесь многопроцессорном сопряжении не могут принимать участие S—процессоры, следовательно, соответствующие участки связи не нуждаются в блоках памяти, и этим участкам просто не следует назначать блоки памяти. Место расположения списка назначений определяется следующими параметрами:

— идентификатор блока
— номер блока
— начальный адрес списка назначений.

Таким образом, перечисленные параметры имеют смысл только в режиме работы "Ручной".

3.1.2 Количество CPU

Данный параметр имеет смысл только в режиме работы "Автоматический". Возможные значения параметра были рассмотрены ранее.

3.1.3 Идентификатор блока

1: Блок данных DB
2: Блок данных DX
0, 3—255: Недопустимые значения, использование которых приводит к конфликту инициализации

3.1.4 Номер блока
Начальный адрес списка назначений

Вместе с идентификатором блока, номер блока и начальный адрес списка назначений определяют область (точнее — начальный адрес области), в которой размещается список назначений. Список назначений содержит ряд входных параметров функции ИНИЦИАЛИЗАЦИЯ, это значит, что к списку назначений осуществляется доступ только для чтения, который не изменяет содержимое самого списка назначений. Структура списка определена следующим образом;

Список назначений:

Слово данных	Формат	Величина	Пояснения	
DW n+0	KC	SI	Передатчик	= CPU 1
DW n+1	KY	2, a	Приемник	= CPU 2
DW л+2	KY	3, b	Приемник	= CPU 3
DM n+3	KY	1, c	Приемник	== CPU 4
DW n+4	KC	S2	Передатчик	= CPU 2
DM n+5	KY	1, d	Приемник	= CPU 1
DW n+6	KY	3, e	Приемник	= CPU 3
DM n+7	KY	4, f	Приемник	= CPU 4
DW n+8	KC	S3	Передатчик	= CPU 3
DW n+9	KY	1, q	Приемник	= CPU 1
DW n+10	KY	2, h	Приемник	= CPU 2
DW n+11	KY	4, I	Приемник	== CPU 4
DW n+12	KC	S4	Передатчик	= CPU 4
DW n+13	KY	1, k	Приемник	= CPU 1
DW n+14	KY	2. l	Приемник	= CPU 2
DW n+15	KY	3, m	Приемник	= CPU 3

Приведенная структура должна сохраняться и в том случае, когда установлено менее четырех CPU.

Вместо букв с a по m должны подставляться числа в диапазоне от О -до 48. Сумма подставленных чисел не должна превышать 48.

Пример:

Подключены три CPU. CPU 2 будет передавать большое количество данных двум остальным CPU, которые, в свою очередь, будут передавать CPU 2 небольшое количество данных — подтверждения, пересылаемые в процессе логического квитирования. Обмен данными между CPU 1 и 3 осуществляться не будет.

Список назначений, пример!

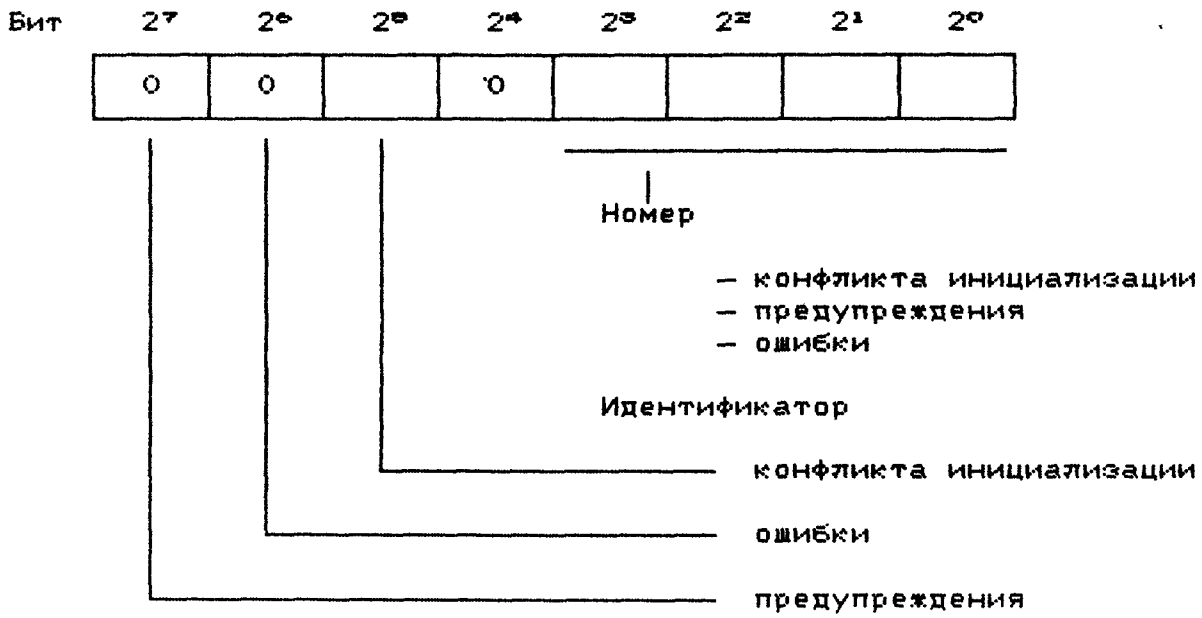
Слово данных	Формат	Величина	Пояснения	
DW n+0	KC	S1	Передатчик =	= CPU 1
DM n+1	KY	2,2	Приемник =	= CPU 2
DW n+2	KY	3,0	Приемник =	= CPU 3
DW n+3	KY	4,0	Приемник =	= CPU 4
DW n+4	KC	S2	Передатчик =	= CPU 2
DW n+5	KY	1,22	Приемник =	= CPU 1
DW n+6	KY	3,22	Приемник =	= CPU 3
DW n+7	KY	4,0	Приемник =	= CPU 4
DW n+8	KC	S3	Передатчик a	s CPU 3
DW n+9	KY	1,0	Приемник =	= CPU 1
DW n+10	KY	2,2	Приемник =	= CPU 2
DW n+11	KY	4,0	Приемник =	= CPU 4
DW n+12	KC	S4	Передатчик =	= CPU 4
DW n+13	KY	1,0	Приемник =	= CPU 1
DW n+14	KY	2,0	Приемник =	= CPU 2
DW n+15	KY	3,0	Приемник =	= CPU 3

3.2 Выходные параметры

3.2.1 Конфликт инициализации

Этот байт сигнализирует о том, была эти Функция ИНИЦИАЛИЗАЦИЯ обработана корректно и полностью.

Рассматриваемые конфликты инициализации идентифицируются и индицируются Функцией в порядке возрастания их номеров.



Конфликт инициализации

В случае конфликта инициализации необходимо внести изменения в программу и/или в параметры.

Номера конфликтов инициализации

1. Нет или недостаточно буферов связи (номера с 252 по 255),
<33> необходимых для многопроцессорного сопряжения.

2. дефект буферов связи (номера с 252 по 255), необходимых
(34) для многопроцессорного сопряжения.

3.(35) Недопустимый параметр "Автоматический/ручной режим работы". Необходимо различать следующие варианты;

— Значение идентификатора "Автоматический/ручной" меньше 1
— Значение идентификатора "Автоматический/ручной" больше 2

4.(36) Недопустимый параметр "Количество CPU".
Необходимо различать следующие варианты:

— Количество CPU меньше 2
— Количество CPU больше 4

5.(37) Недопустимый параметр "Идентификатор блока". Необходимо различать следующие варианты:

— Значение идентификатора блока меньше 1
— Значение идентификатора блока больше 2

6. (38) Недопустимый параметр "Номер блока"; соответствующий блок является блоком специального назначения Необходимо различать следующие варианты:

— Если идентификатор блока =1, то DBO, DB1, DB2
— Если идентификатор блока =2, то DXO

7.(39) Ошибка в параметре "Номер блока", блока с указанными параметрами не существует.

8.(40) Слишком большое значение параметра "Начальный адрес списка назначений" или недостаточный размер блока данных.

9.(41) В блоке данных некорректно построен список назначений.

10.(42) Суммарное количество выделяемых блоков памяти превышает 48.

Предупреждение При выполнении Функции ИНИЦИАЛИЗАЦИЯ предупреждения не выдаются.

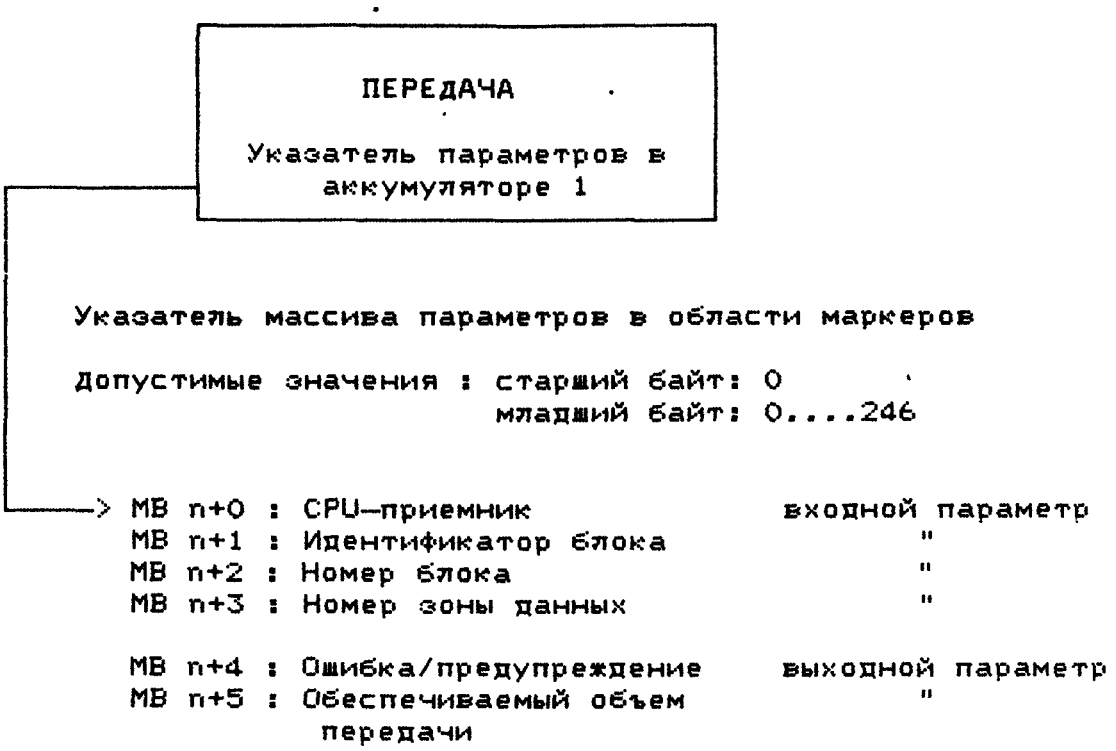
Ошибка

При выполнении Функции ИНИЦИАЛИЗАЦИЯ невозможно возникновение ошибки.

3.2.2 Общий объем назначенной памяти

Этот параметр указывает, сколько из имеющихся в распоряжении 48 блоков памяти закреплено за участками связи.
В режиме работы "Автоматический" этот параметр всегда равен 48. В случае выбора режима работы "Ручной" этот параметр может иметь значение меньше 48. Значение рассматриваемого параметра меньшее 48 в режиме "Ручной" означает, что имеющаяся память просто не используется.

4 Функция ПЕРЕДАЧА (OB202)



Функция ПЕРЕДАЧА обеспечивает передачу зоны данных в промежуточную память координатора KOR C. Кроме того, эта Функция выдает справочную информацию о том, сколько еще зон данных может быть передано и сколько еще зон данных может быть занесено *в* промежуточную память.

4.1 Входные параметры

4.1.1 CPU—приемник

Передаваемые данные предназначены для CPU—приемника. Допустимые значения этого параметра от 1 до 4, но при этом значение этого параметра не должно совпадать с "собственным" номером.

4.1.2 Идентификатор блока

1: Блок данных
2: Блок данных

0,3—255: Недопустимые значения, использование которых приводит к выдаче сообщения об ошибке.

4.1.3 Номер блока

Номер блока вместе с идентификатором блока (этот параметр уже был рассмотрен) и с номером зоны (этот параметр будет рассмотрен далее) определяет область, из которой должны выбираться передаваемые данные (а также область, в которую они будут заноситься на CPU—приемнике).

При этом следует учитывать, что определенные блоки данных являются блоками специального назначения. Так в R—процессоре и в CPU 928 блоками специального назначения являются блоки данных DBO, DB1, DB2 и DX0. Блоки специального назначения нельзя использовать в описываемой здесь передаче данных !

4.1.4 Номер зоны Номер зоны обозначает передаваемую область данных.

Номер зоны	Область данных	
	Первое слово данных	Последнее слово данных
0	DW 0	DW 31
1	DW 32	DW 63
2	DW 64	DW 95
3	DW 96	DW 127
4	DW 128	DW 159
5	DW 160	DW 191
6	DW 192	DW 223
7	DW 224	DW 255
8	DW 256	DW 287
9	DW 288	DW 319
и т.д.	и т.д.	и т.д.

Возможны следующие варианты:

1. При достаточном размере блока данных будет получена, в соответствии с приведенной таблицей, область данных из 32 слов.

2. В случае, когда конец блока данных находится в определенной параметрами зоне* будет передаваться область данных из 31 слова.

3. Если сформированный адрес первого слова данных находится за пределами блока данных, то функция ПЕРЕДАЧА идентифицирует это состояние как ошибочное и выдает соответствующее сообщение.

Пример:

Блок данных состоит из 80 слов — слова данных с DW0 по DW74, и 5 слов — заголовок блока данных.

Номер зоны	Область данных Первое слово данных Последнее слово данных		Длина
0	DW 0	DM 31	32 слова
1	DM 32	DM 63	32 слова
2	DM 64	DW 74	11 слова
3 и более		ошибка в задании параметра	

4.2 Выходные параметры

4.2.1 Ошибки и предупреждения

Этот байт сигнализирует о том, была эти Функция ПЕРЕДАЧА обработана корректно и полностью.



Ошибки

В случае выявления ошибки необходимо внести изменения в программу и/или в параметры.

Номера ошибок:

1.(65) Недопустимый параметр "CPU—приемник". Необходимо различать следующие варианты:

— Номер CPU—приемника превышает 4
— Номер CPU—приемника меньше 1
— Номер CPU—приемника совладает с "собственным номером

3.(67) Ошибка при вызове организационного блока специальной Функции.

Необходимо различать следующие варианты:

— Ошибка является ошибкой-следствием, т.к. не был осуществлен вызов функции ИНИЦИАЛИЗАЦИЯ или же вызов Функции ИНИЦИАЛИЗАЦИЯ привел к конфликту инициализации.

— Дублирование вызова: Вызов одной из Функций — т.е. Функций ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЕМ или ТЕСТ ПРИЕМА, недопустим, т.к. в рассматриваемом CPU на одном из подчиненных уровней обработки (например на уровне обработки циклической программы) уже был осуществлен вызов одной из Функций ИНИЦИАЛИЗАЦИЯ, ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЁМ, или ТЕСТ ПРИЕМА <см. Раздел "Вызов и глубина вложения организационных блоков специальных Функций).

— Ошибка в "собственном номере CPU (разрушены системные данные). После включения и выключения напряжения питания системная программа вновь сгенерирует номер CPU.

4.(68) Ошибка в управляющих данных (управление очередью) выбранного участка связи; посредством Функции ИНИЦИАЛИЗАЦИЯ

необходимо заново выполнить распределение промежуточной памяти координатора С (KOR C).

5.(69) Недопустимый параметр "Идентификатор блока". Необходимо различать следующие варианты:

— Значение идентификатора блока меньше 1
— Значение идентификатора блока больше 2

6.(70) Недопустимый параметр "Номер блока": соответствующий ёлок является блоком специального назначения. Необходимо различать следующие варианты:

— Если идентификатор блока = 1, то DBO, DB1, DB2
— Если идентификатор блока =2, то DXO

7.(71) Ошибка в параметре "Номер блока", блока с указанными параметрами не существует.

8.(72) Ошибка в параметре "Номер зоны": Недостаточный размер данных или слишком большой номер зоны.

Предупреждение

Предупреждение выдается в том случае, когда Функция не может быть выполнена. Следует повторить вызов Функции, например, в следующем цикле.

Номер предупреждения:

1: Функция ПЕРЕДАЧА не может выполнить передачу данных, т.к. *(129)* уже при вызове Функции обеспечиваемый объем передачи (смотри далее) был равен нулю.

Конфликт инициализации

При выполнении Функции ПЕРЕДАЧА невозможна возникновение ситуации конфликта инициализации.

4.2.2 Обеспечиваемый объем передачи

Для того, чтобы можно было выполнить передачу данных из CPU S в CPU E, необходимо обеспечить временное их хранение в промежуточной памяти. Каждому возможному участку связи и направлении" связи, которые определяются парой параметров CPU—передатчик и CPU—приемник, назначается вполне определенный объем памяти.

Память, назначаемая участку связи и направлении" связи, измеряется и выражается в блоках (размером в 32 слова) (смотри описание Функции ИНИЦИАЛИЗАЦИЯ). Блок памяти (его размер всегда равен 32 словам) отводится только под одну единственную зону данных (размер зоны данных — от одного до 32 слов данных). Зона данных загружается в блок памяти организационным блоком функции ПЕРЕДАЧА и выбирается из блока памяти организационным блоком Функции ПРИЕМ.

Параметр "Обеспечиваемый объем передачи" содержит информацию о том, сколько зон данных может быть еще передано и записано в промежуточную память.

5 Функция ТЕСТ ПЕРЕДАЧИ <OB 203>

ТЕСТ ПЕРЕДАЧИ

Указатель параметров в
аккумуляторе 1

Указатель массива параметров в области маркеров

Допустимые значения:	старший байт: О младший байт; О....246	
MB n+0: CPU—приемник		входной параметр не
MB n+1: /		используется
MB n+2: Ошибка		выходной параметр
MB n+3 : Обеспечиваемый объем передачи		

Функция ТЕСТ ПЕРЕДАЧИ определяет количество свободных блоков памяти в промежуточной памяти координатора KOR C. В соответствии с полученным результатом вызова Функции — числом, которое является количеством свободных блоков промежуточной памяти, можно для передачи М зон данных выполнить М раз вызовов функции ПЕРЕДАЧА.

5.1 Входные параметры 5.1.1

CPU—приемник

Участок связи, для которого определяется обеспечиваемый объем передачи, (см. далее), идентифицируется номером "собственного" CPU и номером CPU—приемника.

5.2 Выходные параметры 5.2.1

Ошибки

Этот байт сигнализирует о том, была ли Функция ТЕСТ ПЕРЕДАЧИ обработана корректно и полностью.



Ошибки

В случае выявления ошибки необходимо внести изменения в программу и/или в параметры.

Номера ошибок!

1. Недопустимый параметр "CPU—приемник". (65) Необходимо различать следующие варианты!

— Номер CPU—приемника превышает 4
— Номер CPU—приемника меньше 1
— Номер CPU—приемника совпадает с "собственным" номером

3. Ошибка при вызове организационного блока специальной Функции (67) Необходимо различать следующие варианты;

— Ошибка является ошибкой—следствием, т.к. не был осуществлен вызов Функции ИНИЦИАЛИЗАЦИЯ или же вызов Функции ИНИЦИАЛИЗАЦИЯ привел к конфликту инициализации.
— Дублирование вызова: Вызов одной из Функций — т.е. Функций ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЕМ или ТЕСТ ПРИЕМА, недопустим, т.к. в рассматриваемом CPU на одном из подчиненных уровней обработки (например на уровне обработки циклической программы) уже был осуществлен вызов- одной из Функций ИНИЦИАЛИЗАЦИЯ, ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЕМ, или ТЕСТ ПРИЕМА <см. Раздел "Вызов и глубина вложения организационных блоков. специальных Функций").

— Ошибка в "собственном" номере CPU (разрушены системные данные). После включения и выключения напряжения питания системная программа вновь сгенерирует номер CPU.

4.(68) Ошибка в управляющих данных (управление очередью) выбранного участка связи; посредством Функции ИНИЦИАЛИЗАЦИЯ

необходимо заново выполнить распределение промежуточной памяти координатора C (KOR C).

Предупреждение

При выполнении Функции ТЕСТ ПЕРЕДАЧИ предупреждения не выдаются.

Конфликт инициализации

При выполнении Функции ТЕСТ ПЕРЕДАЧИ невозможно возникновение ситуации конфликта инициализации.

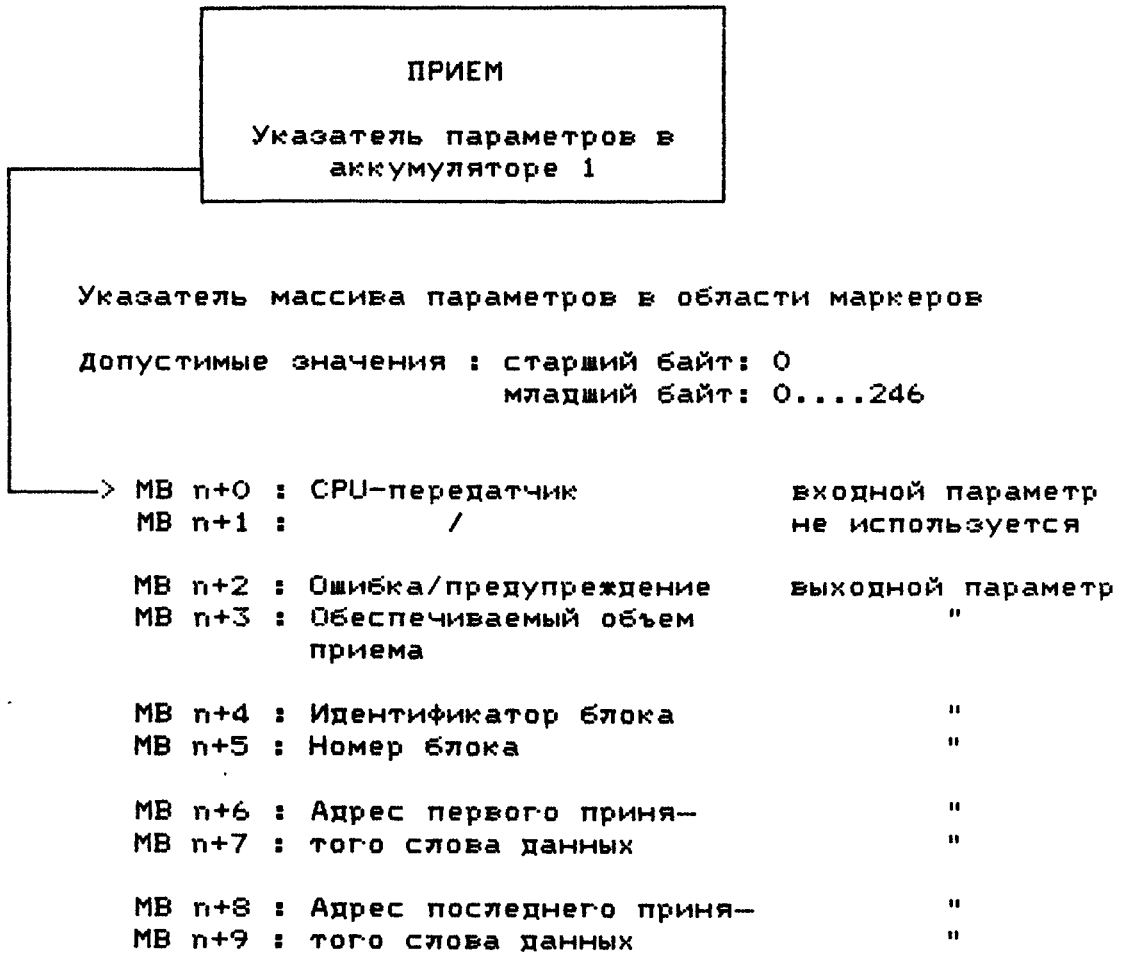
5.2.2 Обеспечиваемый объем передачи

Для того, чтобы можно было выполнить передачу данных из CPU S в CPU E, необходимо обеспечить временное их хранение в промежуточной памяти. Каждому возможному участку связи и направлению связи, которые определяются парой параметров CPU—передатчик и CPU—приемник, назначается вполне определенный объем памяти.

Память, назначаемая участку связи и направлению связи, измеряется и выражается в блоках (размером в 32 слова) (смотри описание Функции ИНИЦИАЛИЗАЦИЯ). Блок памяти (его размер всегда равен 32 словам) отводится только под одну единственную зону данных (размер зоны данных — от одного до 32 слов данных). Зона данных загружается в блок памяти организационным блоком Функции ПЕРЕДАЧА и выбирается из блока памяти организационным блоком Функции ПРИЕМ.

Параметр "Обеспечиваемый объем передачи" содержит информацию о том, сколько зон данных может быть еще передано и записано в промежуточную память.

6 Функция ПРИЕМ (OB 204)



Функция ПРИЕМ обеспечивает выборку зоны данных из промежуточной памяти координатора KOR C. Кроме того, эта Функция выдает справочную информацию о том, сколько еще зон данных находится в промежуточной памяти и сколько еще зон данных может быть принято.

Вызов Функции ПРИЕМ должен выполняться в цикле до тех пор, пока из промежуточной памяти не будут выбраны все находящиеся в ней зоны данных.

6.1 Входные параметры

6.1.1 CPU-передатчик

Организационный блок Функции ПРИЕМ осуществляет ПРИЕМ данных, которые были переданы CPU-передатчиком. Параметр "CPU-передатчик" может иметь значения от 1 до 4, но при этом значение рассматриваемого параметра должно отличаться от "собственного" номера.

6.2 Выходные параметры

6.2.1 Ошибки и предупреждения

Этот байт сигнализирует о том, была эти Функция ПРИЕМ обработана корректно и полностью.



Ошибки

В случае выявления ошибки необходимо внести изменения в программу и/или в параметры.

Номера ошибок:

2.(66) Недопустимый параметр "CPU—передатчик".

Необходимо различать следующие варианты:

— Номер CPU—передатчика превышает 4
— Номер CPU—передатчика меньше 1
— Номер CPU—передатчика совпадает с "собственным" номером.

3.(67) Ошибка при вызове организационного блока специальной Функции Необходимо различать следующие варианты:

— Ошибка является ошибкой—следствием, т.к. не был осуществлен вызов Функции ИНИЦИАЛИЗАЦИЯ или же вызов Функции ИНИЦИАЛИЗАЦИЯ привел к конфликту инициализации.
— Дублирование вызова: Вызов одной из Функций — т.е. Функций ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЕМ или ТЕСТ ПРИЕМА, недопустим, т.к. в рассматриваемом CPU на одном из подчиненных уровней обработки (например на уровне обработки циклической программы) уже был осуществлен вызов одной из функций ИНИЦИАЛИЗАЦИЯ, ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЕМ, или ТЕСТ ПРИЕМА <см. Раздел "Вызов и глубина вложения организационных блоков специальных Функций">.
— Ошибка в "собственном" номере CPU (разрушены системные данные). После включения и выключения напряжения питания системная программа вновь сгенерирует номер CPU.

4.(68) Ошибка в управляющих данных (управление очередью) выбранного участка связи; посредством Функции ИНИЦИАЛИЗАЦИЯ

необходимо заново выполнить распределение промежуточной памяти координатора C (KOR C).

5.(69) Недопустимый параметр "Идентификатор блока", который был передан CPU—передатчиком.

Необходимо различать следующие варианты:
— Значение идентификатора блока меньше 1
— Значение идентификатора блока больше 2

6. (70) Недопустимый параметр "Номер блока" — соответствующий блок является блоком специального назначения. Необходимо различать следующие варианты:

— Если идентификатор блока = 1, то DB0, DB1, DB2
— Если идентификатор блока =2, то DX0

7.(71) Ошибка в параметре "Номер блока", который был передан CPU передатчиком. Блока с указанными параметрами не существует.

9.(73) Размер блока данных недостаточен для приема зоны данных, переданных CPU—передатчиком.

Предупреждение

Предупреждение выдается в том случае, когда Функция не может быть выполнена. Следует повторить вызов Функции, например, в еле думаем цикле.

Номер предупреждения!

2.(130) Функция ПРИЕМ не может выполнить ПРИЕМ данных, т.к. уже (130) при вызове Функции обеспечиваемый объем приема был равен нулю.

Конфликт инициализации

При выполнении Функции ПРИЕМ невозможно возникновение ситуации конфликта инициализации.

6.2.2 Обеспечиваемый объем приема

Для того, чтобы можно было выполнить передачу данным из CPU S в CPU E, необходимо обеспечить временное их хранение в промежуточной памяти. Каждому возможному участку связи и направлению связи, которые определяются парой параметров CPU—передатчик и CPU—приемник, назначается вполне определенный объем памяти.

Память, назначаемая участку связи и направлению связи, измеряется и выражается в блоках (размером в 32 слова) (смотри описание Функции ИНИЦИАЛИЗАЦИЯ). Блок памяти (его размер всегда равен 32 словам) отводится только под одну единственную зону данных (размер зоны данных — от одного до 32 слов данных). Зона данных загружается в блок памяти организационным блоком Функции ПЕРЕДАЧА и выбирается из блока памяти организационным блоком Функции ПРИЕМ.

Параметр "Обеспечиваемый объем приема" содержит информацию о том, сколько зон данных может быть еще записано в промежуточную память и принято.

6.2.3 Идентификатор блока

1: Блок данных DB

2: Блок данных DX

6.2.4 Номер блока

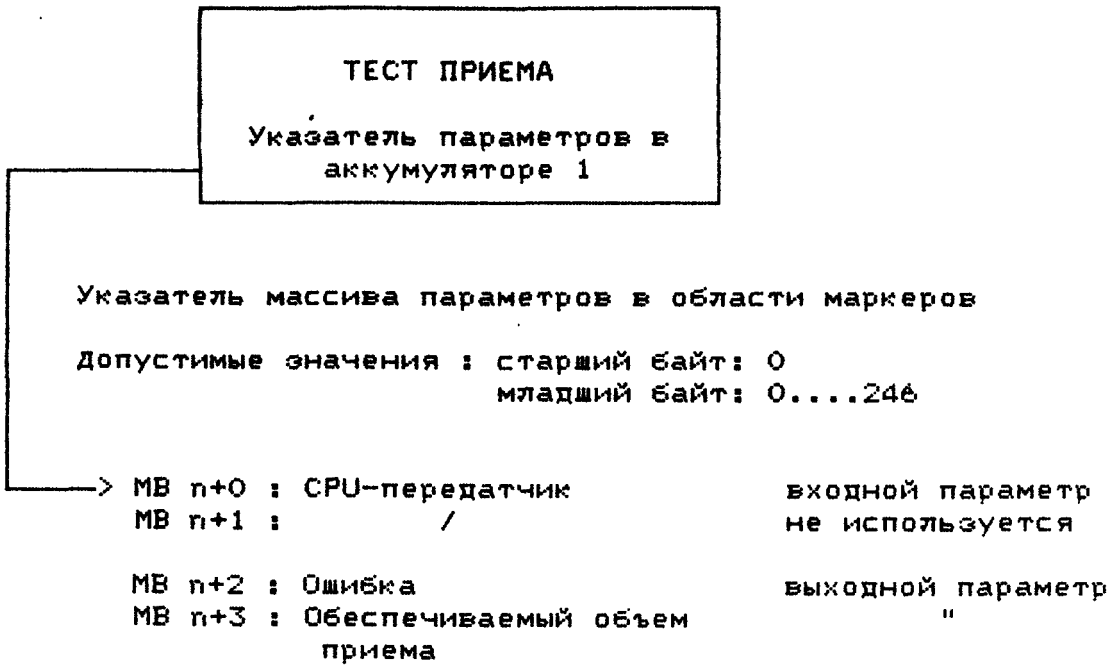
Номер блока вместе с иденти4'икатором блока (был рассмотрен ранее) и с адресами первого и последнего слов данных (эти параметры будут рассмотрены далее) определяют область, в которую принимаемые данные заносятся Функцией ПРИЕМ (и из которой они в CPU—передатчике были выбраны Функцией ПЕРЕДАЧА).

Следует обратить внимание на то, что блоки памяти, используемые для принимаемых данных, должны размещаться в памяти, обеспечивающей доступ для чтения и доступ для записи (RAM). Использование памяти, которая обеспечивает доступ только для чтения (EPROM), может быть целесообразно только для блоков памяти, выделяемых под передаваемые данные.

6.2.5 Адрес первого принятого слова данных
Адрес последнего принятого слова данных

Разность между адресами первого и последнего слова данных, переданных при выполнении Функции, не превышает 31. Указанное ограничение разности адресов объясняется тем, что один вызов Функции приводит к передаче максимум 32 слов.

7 Функция ТЕСТ ПРИЕМА (ОВ 205)



Функция ТЕСТ ПРИЕМА определяет количество занятых блоков памяти в промежуточной памяти координатора KOR C. В соответствии с полученным результатом вызова Функции — числом M, которое является количествам занятых блоков промежуточной памяти, можно, для приема M зон данных, выполнить M раз вызов Функции ПРИЕМ.

7.1 Входные параметры

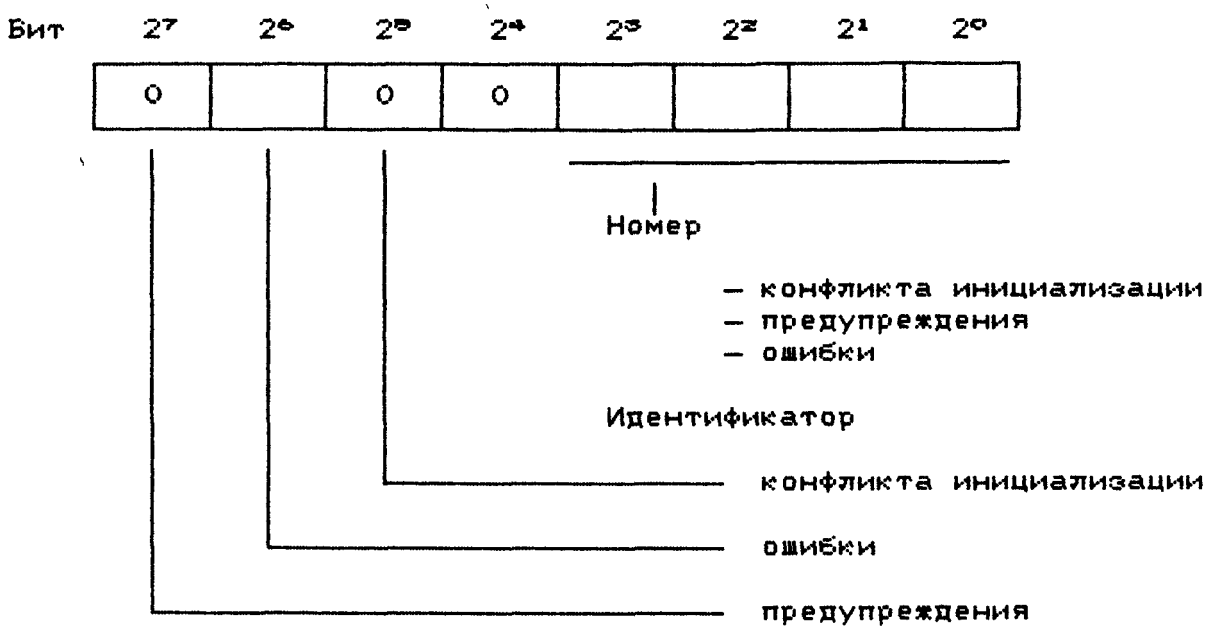
7.1.1 CPU—передатчик

Участок связи, для которого определяется объем приема (см. далее), идентифицируется номером "собственного" CPU и номером CPU—передатчика.

7.2 Выходные параметры

7.2.1 Ошибки

Этот байт сигнализирует о том, была эти Функция ТЕСТ ПРИЕМА обработана корректно и полностью.



Ошибка

В случае выявления ошибки необходимо внести изменения в программу и/или в параметры.

Номера ошибок

2.(66) Недопустимый параметр "CPU—передатчик".
Необходимо различать следующие варианты:

— Номер CPU—передатчика превышает 4
— Номер CPU—передатчика меньше 1
— Номер CPU—передатчика совладает с "собственным" номером

3.(67) Ошибка при вызове организационного блока специальной Функции

Необходимо различать следующие варианты:

— Ошибка является ошибкой—следствием, т.к. не был осуществлен вызов Функции ИНИЦИАЛИЗАЦИЯ или же вызов Функции ИНИЦИАЛИЗАЦИЯ привел к конфликту инициализации.

— Дублирование вызова; Вызов одной из Функций — т.е. Функций ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЕМ или ТЕСТ ПРИЕМА, недопустим, т.к. в рассматриваемой CPU на одном из подчиненных уровней обработки (например на уровне обработки циклической программы) уже был осуществлен вызов одной из Функций ИНИЦИАЛИЗАЦИЯ, ПЕРЕДАЧА, ТЕСТ ПЕРЕДАЧИ, ПРИЕМ, или ТЕСТ ПРИЕМА <см. Раздел "Вызов и глубина вложения организационных блоков специальные Функций).

— Ошибка в "собственном" номере CPU (разрушены системные данные). После включения и выключения напряжения питания системная программа вновь сгенерирует номер CPU.

4.(68) Ошибка в управляющих данных (управление очередью) выбранного участка связи; посредством Функции ИНИЦИАЛИЗАЦИЯ

необходимо заново выполнить распределение промежуточной памяти координатора C.(KOR C).

Предупреждение

При выполнении Функции ТЕСТ ПРИЕМА предупреждения не выдаются.

Конфликт инициализации

При выполнении Функции ТЕСТ ПРИЕМА невозможно возникновение ситуации конфликта инициализации.

7.2.2 Обеспечиваемый объем приема

Для того, чтобы можно было выполнить передачу данных из CPU S в CPU E, необходимо обеспечить временное их хранение в промежуточной памяти. Каждому возможному участку связи и направлению связи, которые определяются парой параметров CPU—передатчик и CPU—приемник, назначается вполне определенный объем памяти.

Память, назначаемая участку связи и направлению связи, измеряется и выражается в блоках (размером в"32 слова) (смотри описание Функции ИНИЦИАЛИЗАЦИЯ). Блок памяти (его размер всегда равен 32 словам) отводится только под одну единственную зону данных (размер зоны данных — от одного до 32 слов данных). Зона данных загружается в блок памяти организационным блоком Функции ПЕРЕДАЧА и выбирается из блока памяти организационным блоком Функции ПРИЕМ.

Параметр обеспечивающий объем приема содержит информацию о том, сколько зон данных может быть еще записано в промежуточную память и принято.

8 Использование средств передачи данных

8.1 Вызов организационных блоков специальных Функций через Функциональные блоки.

Рассматриваемые далее пять Функциональных блоков (FB200, FB202...FB205) содержат вызов соответствующих организационных блоков специальных функций для осуществления многопроцессорной связи (OB200, OB202...OB205).

Номера Функциональных блоков выбраны произвольно, и их можно менять. Параметры организационных блоков специальных Функций передаются при вызове Функциональных блоков как фактические параметры. Прямой вызов организационных блоков специальных Функций выполняется быстрее, однако, в этом случае отсутствуют Формальные параметры, что приводит к снижению наглядности.

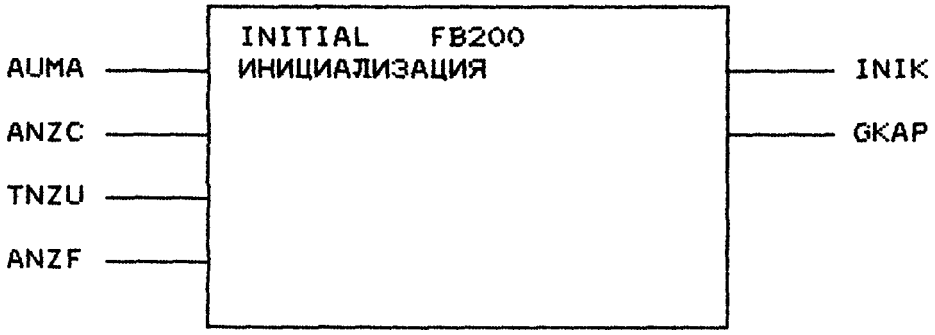
Номер Функционального блока	Имя Функционального блока	Функция
FB200	ИНИЦИАЛИЗАЦИЯ	Предварительное назначение ресурсов
FB202	ПЕРЕДАЧА	Передача зоны данных
FB203	ТЕСТ ПЕРЕДАЧИ	Проверка возможности передачи
FB204	ПРИЕМ	Прием зоны данных
FB205	ТЕСТ ПРИЕМА	Проверка возможности приема

Область маркеров с MB246 до, максимально, MB255 используется функциональными блоками как массив параметров для организационных блоков специальных Функций.

Входные и выходные параметры подробно рассматриваются в описании организационных блоков специальных Функций.

8.1.1 Предварительное назначение ресурсов <FB200>.

Имя параметра	Значение	Вид	Тип	Массив параметров
AUMA	Автоматический/ручной Количество	E	BY	MB246
ANZC	CPU Тип (старший байт) и номер	E	BY	MB247
TNZU	(младший байт) блока данных,	E	14	MW248
ANFZ	содержащего список назначений	E	W	MW250
INIK	Начальный адрес списка назначений	A	BY	MB252
GKAP	Конфликт инициализации Общий объем назначенной памяти	A	BY	MB253

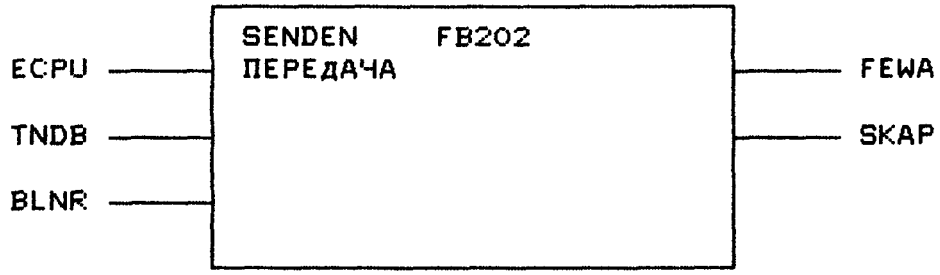



```
FB200                          ДЛН=45   АБС
СЕГМЕНТ  1
ИМЯ   INITIAL
МЕТКА AUMA        E/A/D/B/T/Z: E   BI/BY/W/D: BY
МЕТКА ANZC        E/A/D/B/T/Z: E   BI/BY/W/D: BY
МЕТКА TNZU        E/A/D/B/T/Z: E   BI/BY/W/D; W
МЕТКА ANFZ        E/A/D/B/T/Z: E   BI/BY/W/D: W
МЕТКА -INIK       E/A/D/B/T/Z: A   BI/BY/W/D: BY
МЕТКА GKAP        E/A/D/B/T/Z: A   BI/BY/W/D: BY

 0017     L   =AUMA        Автоматический/ручной
 0018     T   MB246
 0019     L   =ANZC        Количество CPU
00 la     T   MB247
 001B     L   =TNZU        Тип и номер блока данных
001 C     T   MW248
 001D     L   =ANFZ        Начальный адрес списка
001E      T   MW250         назначений
001 F
 0020     L   KB246        Организационный блок спец. Функции:
 0021    SPA OB200          ИНИЦИАЛИЗАЦИЯ
 0022
 0023     L   MB252        Конфликт инициализации
 0024     T   =INIK
 0025     L   MB253        Общий объем назначенной памяти
 0026     T   =GKAP
 0027     BE
```

8.1.2 Передача зоны данных <FB202>

Имя параметра	Значение	Вид	Тип	Массив параметров
ECPLI	CPU—приемник Тип (старший байт) и	E	BY	MB246.
TNDB	номер (младший байт) блока данных	E	M	MW247
BLNR	— источника Номер зоны	E	BY	MB249
FEMA	Ошибка/предупреждение назначений	A	BY	MB250
SKAP	Обеспечиваемый объем передачи	A	BY	MB251



FB202 ДЛН=40 ABC

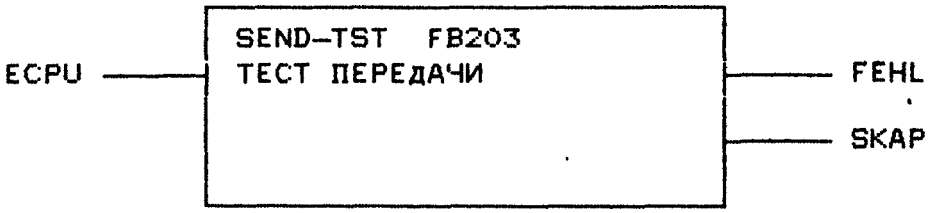
СЕГМЕНТ 1

ИМЯ : SENDEN
METKA:ECPU E/A/D/B/T/Z: E BI/BY/W/D: BY
METKA:TNDB E/A/D/B/T/Z: E BI/BY/M/D: W
METKA:BLNR E/A/D/B/T/Z: E BI/BY/M/D: BY
METKA:FEWA E/A/D/B/T/Z: A BI/BY/M/D: BY
METKA:SKAP E/A/D/B/T/Z: A BI/BY/W/D: BY

```
0014   : L   =ECPU          CPU-приемник
0015   : T   MB246.
0016   : L   =TNDB          Тип и номер блока данных
0017   : T   MW247
0018   : L   =BLNR          Номер зоны данных
0019   : T   MB249 001A    ;
001B   : L   KB246          Организационный блок спей. Функции:
001 C   ; SPA OB202          ПЕРЕДАЧА ЗОНЫ ДАННЫХ
001D
001E   : L   MB250          Ошибка/предупреждение
001F   : T   =FEMA
 0020   : L • MB251          Обеспечиваемый объем передачи
 0021   : T   =SKAP
 0022   : BE
```

8.1.3 Проверка возможности передачи (FB203)

Имя параметра	Значение	Вид	Тип	Массив параметров
ECPU	CPU—приемник	E	BY	MB246
FEHL	Ошибка	A	BY	MB248
SKAP	Обеспечиваемый объем передачи	A	BY	MB249

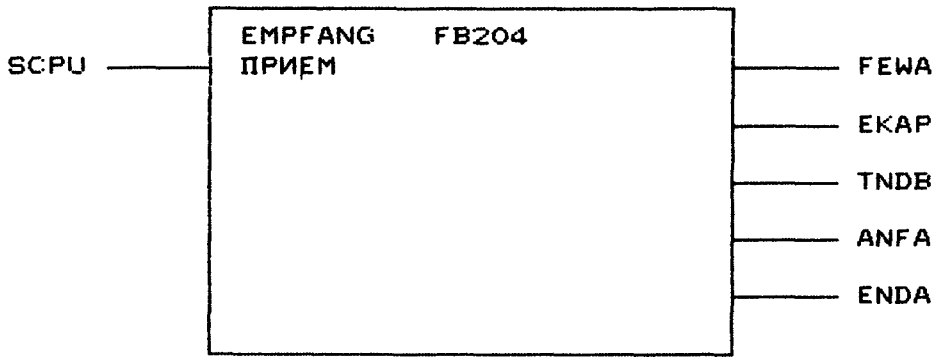



```
FB20.3                          ДЛН=30   АБС:
СЕГМЕНТ  1
ИМЯ  8 SEND-TST
МЕТКА:ECPU        E/A/D/B/T/Z: E   BI/BY/W/D: BY
МЕТКА:FEHL        E/A/D/B/T/Z: A   BI/BY/W/D: BY
МЕТКА:SKAP        E/A/D/B/T/Z: A   BI/BY/W/D: BY

OOOE     L   ==ECPLI        CPU-приемник OOOF     T   MB246
 0010
 0011     L   KB246         Организационный блок спец. функции:
 0012     SPA OB203           ПРОВЕРКА ВОЗМОЖНОСТИ ПЕРЕДАЧИ
 0013
0014-     L   MB248         Ошибка
 0015     T   =FEHL
 0016     L   MB249         Обеспечиваемый объем передачи
 0017     T   =SKAP
 0018     BE
```

8.1.4 Прием зоны данных <FB204>

Имя параметра	Значение	Вид	Тип	Массив параметров
SCPU	CPU—передатчик	E	BY	MB246
FEWA	Ошибка/предупреждение			
EKAP	Обеспечиваемый объем приема	A	BY	MB24S
TNDB	Тип (старший байт) и номер (младший байт) блока данных назначения	A	BY	MB249
		A	W	MM250
ANFA	Адрес первого принятого слова данных (начальный адрес)	A	W	MM252
ENDA	Адрес последнего принятого слова данных (начальный адрес)	A	U	MW254



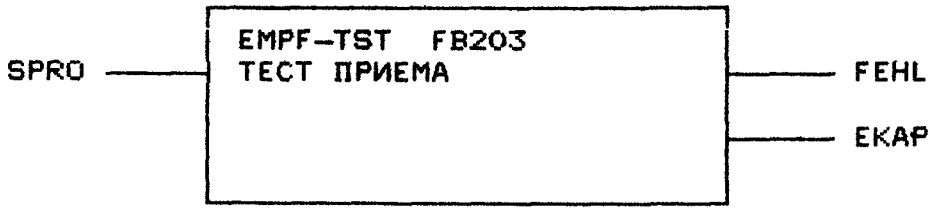
FB204 • ДЛН=45 ABC

СЕГМЕНТ 1
ИМЯ : EMPFANG
METKA:SCPL1 E/A/D/B/T/Z; E BI/BY/M/D: BY
METKA:FEMA E/A/D/B/T/Z: A BI/BY/W/D: BY
METKA:EKAP E/A/D/B/T/Z: A BI/BY/W/D: BY
METKA:TNDB E/A/D/B/T/Z: A BI/BY/M/D: W
METKA:ANFA E/A/D/B/T/Z: A BI/BY/M/D: И
METKA:ENDA E/A/D/B/T/Z: A BI/BY/W/D; M

```
 0017   : L   =SCPU        CPU-передатчик
 0018   : T   MB246
 0019   :
001 A   : L   KB246        Организационный блок спец. Функции:
001B    : SPA OB204          ПРИЕМ ЗОНЫ ДАННЫХ 001C   :
001D    : L   MB24S        Ошибка/предупреждение
001E      T   =FEWA
001F      L   MB249        Обеспечиваемый объем приема
 0020     T   =EKAP
 0021     L   MW250         Тип и номер блока данных
 0022    T  =TNDB
 0023   8 L   MW252         Начальный адрес
 0024    : T   =ANPA
 0025    ; L   MW254        Конечный адрес
 0026    : T   =ENDA
 0027    : BE
```

8.1.5 Проверка возможности ПРИЕМА <FB205>

Имя параметра	Значение	Вия	Тип	Массив параметров
SCPU	CPU—передатчик	E	BY	MB246
FEHL	Ошибка	A	BY	MB248
EKAP	Обеспечиваемый объем приема	A	BY	MB249




```
FB205                          ДЛН=30  ABC
СЕГМЕНТ  1
ИМЯ  : EMPF-TST
МЕТКА:SCPU        E/A/D/B/T/Zs E   BI/BY/W/Ds BY
МЕТКА:FEHL        E/A/D/B/T/Zs A  .BI/BY/W/D: BY
МЕТКА:EKAP        E/A/D/B/T/Zs A   BI/BY/W/Ds BY

000E    L   =SCPU        CPU-передатчик

0000    T   MB246
0010
0011    L   KB246        Организационный блок спец. Функции:
0012    SPA OB205          ПРОВЕРКА-ВОЗМОЖНОСТИ ПРИЕМА
0013
0014    L   MB248        Ошибка
0015    T   =FEHL
0016    L   MB249        Обеспечиваемый объем приема
0017    T   =EKAP
0018    BE
```

i

8.2 Передача блоков данных

8.2.1 Описание принципа передачи

Функциональный блок UEBT-DAT <FB110> пересылает зоны данных, количество которых определяется посредством параметров, из блока данных CPU в блок данных другого CPU. Типы и номера блоков данных, участвующих в процессе передачи т.е. блок данных, из которого выбираются зоны данных, и блок данных, в который зоны данных заносятся, совпадают друг с другом (описание списка параметров, Функционального блока и программы на языке STEP—5 приводятся далее). Номер Функционального блока выбран произвольным образом и может быть изменен.

8.2.2 Передача блока данных (FBI 10)

Подлежащая передаче область данных определяется входными параметрами ERSB (номер первой подлежащей передаче зоны данных) и ANZB (=количество подлежащих передаче зон данных). Обычно зона данных состоит из 32 слов. При передаче последней зоны данных может, в зависимости от размера блока данных, пересылаться менее 32 слов.

Процесс передачи инициируется поступлением положительного Фронта на стартовый вход STAR. Если при этом выходной параметр REST равен нулю, то Функциональный блок UEBT—DAT может выполнить передачу всех (в соответствии с параметром ANZB) зон данных.

В случае, когда выходкой параметр REST имеет значение больше нуля, необходимо, например в следующем цикле, выполнить завершающие вызовы. В такой ситуации изменение всего набора параметров (т.е. значений всех параметров) может быть осуществлено пользователем (прикладной программой) только после того, как при завершающем вызове будет получен выходной параметр REST равный нулю, свидетельствующий о том, что передача области данных закончена.

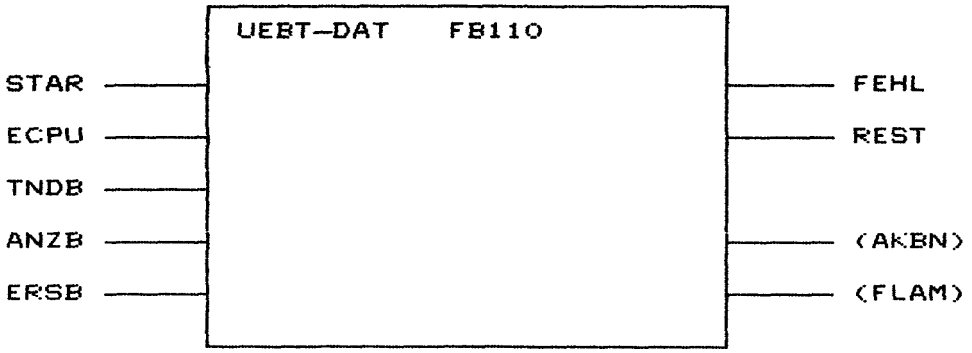
Допускается многократный вызов Функционального блока с использованием при каждом вызове другого набора параметров. При этом будет происходить одновременная передача разных ("вложенных" друг в друга) областей данных. Кроме того, возможно "непосредственное" использование организационных блоков специальных Функций поддержки многопроцессорного сопряжения — т.е. организационных блоков с OB202 по OB205. Такой вариант использования организационных блоков рассматривается в приводимом далее примере.

В случае, когда Функциональный блок UEBT—DAT не может обеспечить корректную обработку Функции ПЕРЕДАЧА (OB202), в выходной параметр FEHL заносится номер соответствующей ошибки, и VKE устанавливается в значение 1.

Функциональный блок UEBT-DAT использует байты маркеров с MB246 по MB251 в качестве промежуточных или интерфейсных маркеров. Всем остальным переменным, значения которых имеют смысл до тех пор, пока после многократного вызова Функционального блока UEBT—DAT не будет получен выходной параметр REST равный нулю, память назначается механизмом поддержки работы с Формальными и Фактическими параметрами. Такой подход необходим для обеспечения возможности одновременной передачи различных блоков данных.

Имя параметра	Значение	Вид	Тип
STAR	Инициирование передачи блока данных по положительному Фронту CPU—приемник	E	BI
ECPU		E	BY
TNDB	Тип (старший байт) и номер <младший байт> передаваемого блока данных	E	W
ANZB		E	BY
ERSB	Количество передаваемых зон данных	E	BY
FEHL	Номер первой передаваемой зоны данных	A	BY
REST	Ошибка	A	BY
AKBN	Количество еще не переданных зон данных	A	BY
FLAM	Номер текущей зоны Маркер Фронта	A	BI

Внутренние интерфейсные маркеры, не предназначенные для обработки.



FB110 ДЛН=8-9 АБС

СЕГМЕНТ 1

ИМЯ : UEBT-DAT

METKA:STAR E/A/D/B/T/Z E BI/BY/W/D BI
METKA:ECPU E/A/D/B/T/Z E BI/BY/W/D BY
METKA:TNDB E/A/D/B/T/Z E BI/BY/W/D M
METKA:ANZB E/A/D/B/T/Z E BI/BY/M/D BY
METKA:ERSB E/A/D/B/T/Z E BI/BY/W/D BY
METKA:FEHL E/A/D/B/T/Z A BI/BY/W/D BY
METKA:REST E/A/D/B/T/Z A BI/BY/M/D BY
METKA:AKBN E/A/D/B/T/Z A BI/BY/W/D BY
METKA:FLAM E/A/D/B/T/Z A BI/BY/M/D BI

```
0020    L   =ECPU       Предварительное назначение массива
0021    T   MB246       параметров для организ. блока спец. Функции 0В 202
0022    L   =TNDB
0023    T   MW247
0024
0025    L   =REST•      Сначала должна быть выполнена пересылка
0026    L   “BO         еще не переданных зон данных
0027    ><  F
0028    SPB =UEBT
```

```
0029
002A     UN  =STAR
002B     RB  =FLAM
002C     ON  =STAR
002D     0   =PLAM
002E     SPB =GUT
002F     S   =FLAM
0030
0031     L   =ANZB          Инициализация глобальных маркеров
0032     T   =REST          при поступлении положительного
0033     L   =ERSB          Фронта на стартовый вход
0034     T   =AKBN
0035
0036     L   =REST          До тех пор, пока REST не равен 0,
0038SCHL  L   KF+0          продолжать попытки пересылки зон
0039     !=F                данных
OO3A      SPB =GUT
003BUEBT  L   =AKBN
OO3C:      T   MB249
003D      L   KB246        Организационный блок спец. Функций;
OO3E      SPA OB202        ПЕРЕДАЧА ЗОНЫ "ДАННЫХ
003F      L   MB250
0040      SPM =FEHL        Аварийное завершение по ошибке
0041      SPP =GUT         Аварийное завершение по обеспечиваемому объему передачи,
.                          равному О
0042      L   =AKBN        Увеличение номера зоны
0043      I   1
0044      T   =AKBN
0045      L   =REST        Уменьшение  количества  оставшихся
0046      D   1            зон данных
0047      T   =REST
0048      SPA =SCHL
0049
004A   GUT   U   MO.O      Нормальное завершение программы
004B      UN  MO.O
004C      L   KBO          VKE=0, FEHL=0
004D      T   =FEHL
004E      BEA 004F
0050    FEHL  T   =FEHL        Завершение программы в случае ошибки
0051      L   KBO
0052    T   =REST        VKE=1, FEHL содержит номер ошибки
0053    : BE
```

8.2.3 Пример

CPU 1 должен в циклической прикладной программе передавать на CPU 2 блоки данных DB3 (зоны данных с 2 по 5) и DB4 (зоны данных с 1 по 3). В CPU 2, также в циклической программе, должен выполняться вызов Функции ПРИЕМ <OB204>.

В CPU должны быть загружены следующие блоки:

Функция	CPU 1	CPU 2
Блок перезапуска	OB20	
Блок цикла	FBO	FBO
Передаваемые блоки	DB3 и DB4	
Принимаемые блоки		DB3 и DB4

OB20 осуществляет вызов Функции ИНИЦИАЛИЗАЦИЯ (OB200) и резервирует для участка связи от CPU 1 до CPU 2 несколько блоков памяти.

В циклической прикладной программе Функционального блока FBO CPU 1 содержатся два вызова Функционального блока UEBT—DAT, причем для вызовов используются разные наборы параметров.

Приход положительного Фронта на входе EO.O приводит к началу передачи первого блока данных DB3. Положительный Фронт сигнала на входе E0.1 запускает передачу второго блока данных DB4.

```
FB0                                ДЛН=66  АБС
СЕГМЕНТ 1
ИМЯ :DEMO
 0005     L    KB2        На CPU 2 ...
 0006     T    MBO
 0007     L    KY1,3       ... из блока данных DB3
 0009     T    MM1
000A      L    KB4        ...пересылаются четыре зоны данных,
000 B     T    MB3
 000C     L    KB2        ...начиная с зоны данных 2
000D      T    MB4
000E
000F      SPA  FBI 10             <
 0010 имя  UEBT-DAT
 0011STAR      E 2.0
 0012 EPRO      MBO
 0013 TNDB      MM I
 0014 ANZB      MB3
 0015 ERSB      MB4
 0016 FEHL      MB5
 0017 REST      MB6
00 18 AKNB      MB7
0019 FLAM      M 8.0
001A
001B
001C      SPB  =HALT        Аварийное завершение по ошибке
001D
 00 IE     L    KB2        На CPU 2 ... 59
```

```
001F    :T   MB10
0020     L   KY1,4        ... из блока данных DB4
0022     T   MM 11
0023     L   KB3          ... пересылаются три зоны данных,
0024     T   MB 13
0025     L   KB1          ... начиная с зоны данных 1
0026     T   MB 14
0027
0028     SPA  FBI 10
0029 имя  UEBT-DAT
002A STAR     E 2.1
002B EPRO     MB 10
002C TNDB     MM 11
002D ANZB     MB13
002E ERSB     MB 14
002F FEHL     MB5
0030 REST     MB 16.
0031 AKNB     MB 17
0032 FLAM     M S.I.
0033
0034
0035     SPB  =HALT       Аварийное завершение по ошибке
0036     BEA
0037
0038 HALT:STP
0039     : BE
```

В CPU 2 Функция ПРИЕМ <OB204>, вызываемая в Функциональном блоке, FBO заносит каждую переданную зону данных *в* соответствующий блок данных. Прием всего блока данных может реализоваться за несколько циклов.

FBO ДЛН=26 АБС

СЕГМЕНТ 1

Имя : EMPF-DAT

```
0005    :L   KB1              Прием данных от CPU I
0006     T   MB246
0007
0008 SCHL L   KB246          Организационный блок спец. Функции:
0009     SPA  OB204          ПРИЕМ
000A     SPM  =FEHL          Аварийное завершение по ошибке
000B     L   MB249           Функция ПРИЕМ вызывается до тех
000C     L   KBO             пор, пока в промежуточной памяти не
000D     XF                  останется ни одной свободной зоны
000E     SPB  =SCHL          данных, т.е. обеспечиваемый объем
000F                         приема = О
0010     BEA
OOH FEHL STP
0012    :BE
```

8.3 Расширение области маркеров межпроцессорной связи

8.3.1 Постановка задачи

В многопроцессорном программируемом контроллере S5—135U каждый из 256 маркерных байтов CPU можно, путем занесения соответствующей записи в блок данных DB1, объявить входным или выходным маркером межпроцессорной связи. Однако, при этом уменьшается количество маркерных байтов, которые могли бы использоваться обычным образом. Для передачи записи данных, состоящей из нескольких байтов, необходимо предусмотреть дополнительные меры (переменная—семафор или специальная Функция OB224), предотвращающие обработку приемником не полностью переданной записи данных.

8.3.2 Решение

Следующие друг- за другом и начинающиеся с DWO слова данных блока данных DB или DX определяются как "слова данных межпроцессорной связи". За каждым участком связи закрепляется "свой" блок данных и, таким образом, участки связи совершенно не зависят друг от друга.

Выполнение блока циклической программы <OB1 или FBO> начинается с приема слов данных межпроцессорной связи, которая реализуется организационными блоками специальных Функций поддержки многопроцессорного сопряжения. Затем выполняется "стандартная" циклическая программа, которая осуществляет обработку принятых данных и генерирует передаваемые данные. Сгенерированные передаваемые данные в конце цикла передаются с помощью организационных блоков специальных Функций поддержки многопроцессорного сопряжения. Таким образом, эти данные могут быть приняты другими CPU в начале их цикла.

Для каждого из 12 возможных участков связи, независимо от всех остальных участков связи, справедливо следующее:

— CPU—передатчик переходит в активное состояние только после полной выборки CPU—приемником "старых" данных из промежуточной памяти координатора KOR C.
— CPU—приемник переходит в активное состояние только после того, как CPU—передатчик полностью запишет "новые" данные в промежуточную память координатора KOR C.

При этом в распоряжении CPU—приемника будет полная новая запись данных или же будет сохранена без каких-либо изменений старая запись данных. Такой подход предотвращает смешение "старых" и "новых" данных !

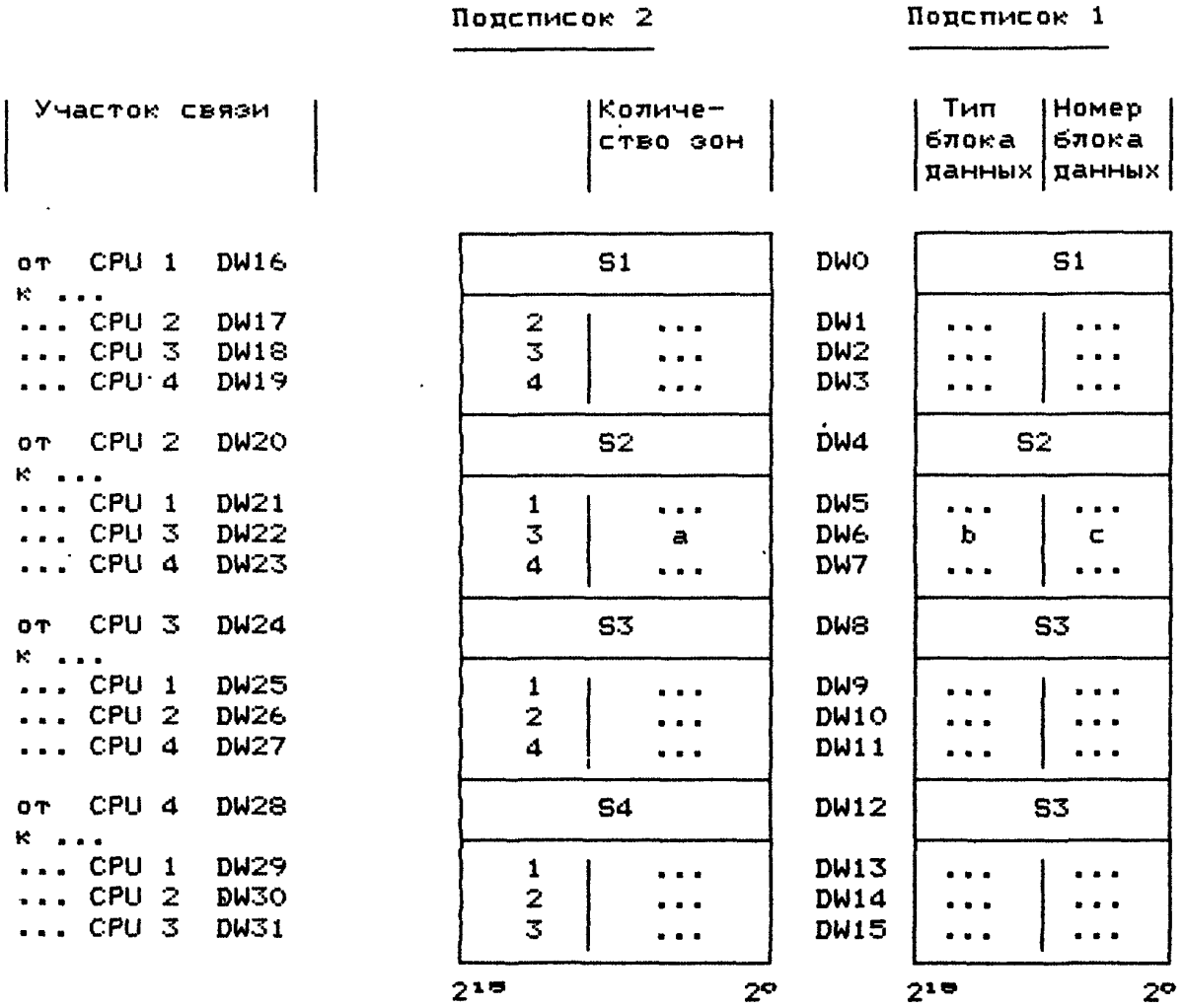
8.3.3 Структура данных

Список управления связями (он будет описан далее) содержит информацию о том, какие слова данных (далее вместо термина "слова данных" будет использоваться термин "область слов данных") и по какому маршруту (т.е. от какого CPU и на какой CPU) будут передаваться при реализации многопроцессорного сопряжения. Список ++управления связями находится в дополнительном блоке данных, наличие которого обязательно во всех участвующих в сопряжении CPU.

Области слов данных всегда начинаются со слова данных DWO. Размер областей слов данных указывается в зонах. При этом необходимо учитывать следующее:

— Полная зона данных состоит из 32 слов данных.
— Если последняя зона передаваемого блока данных "усечена", т.е. ее размер может быть от одного до 31 слова данных, то передается соответственно меньшее количество слов данных.
— В случае, когда размер передаваемого блока данных превышает размер, указанный в зонах в списке управления связями, лишние слова данных могут использоваться самим CPU.
— В случае, когда размер принимающего блока данных превышает размер принятой области слов данных, лишние слова данных могут использоваться самим CPU.

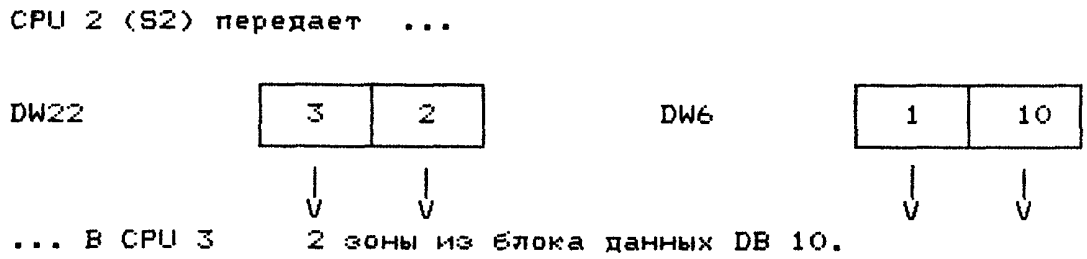
Структура списка управления связью



Список управление связью состоит из двух, имеющих аналогичную структуру и равных по размеру, подсписков. Каждый подсписок занимает 16 слов данных. С учетом того, что любой из четырех CPU может быть CPU—передатчиком <S1,S2,S3,S4>, каждый участок связи описывается тремя следующими записями:

a) **Количество зон**
   Запись "количество зон" определяет ' размер (^количество слов данных) передаваемой области слов данных (для отсутствующих или неиспользуемых участков связи в качестве количества зон указывается число О, аналогично в этом случае задаются тип и номер блока данных).
b) **Тип блока данных**
   Запись "тип блока данных" используется для определения типа блока данных, содержащего передаваемую область слов данных.
c) **Номер блока данных**
   Запись "номер блока данных" используется для определения номера блока данных, содержащего передаваемую область слов данных.

В соответствии с рассмотренной ранее структурой эти записи можно построчно считывать и заполнять. Так, например, для передачи из CPU 2 (S2) в CPU 3 первых двух зон данных блока DB10 необходима следующая запись:



Подсписок 2 идентичен списку назначений (режим работы "ручной"), который используется для Функции ИНИЦИАЛИЗАЦИЯ (ОВ200). В блоке данных подсписок 1 должен занимать адреса с О по 15, а подсписок 2 — адреса с 16 по 31. Записи, выделенные жирным шрифтом, изменению не подлежат.

8.3.4 Структура программы

При запуске CPU, путем вызова Функции ИНИЦИАЛИЗАЦИЯ (OB200), резервирует под участки связи ровно столько блоков памяти в координаторе, сколько зон данных должно передаваться на соответствующем участке.

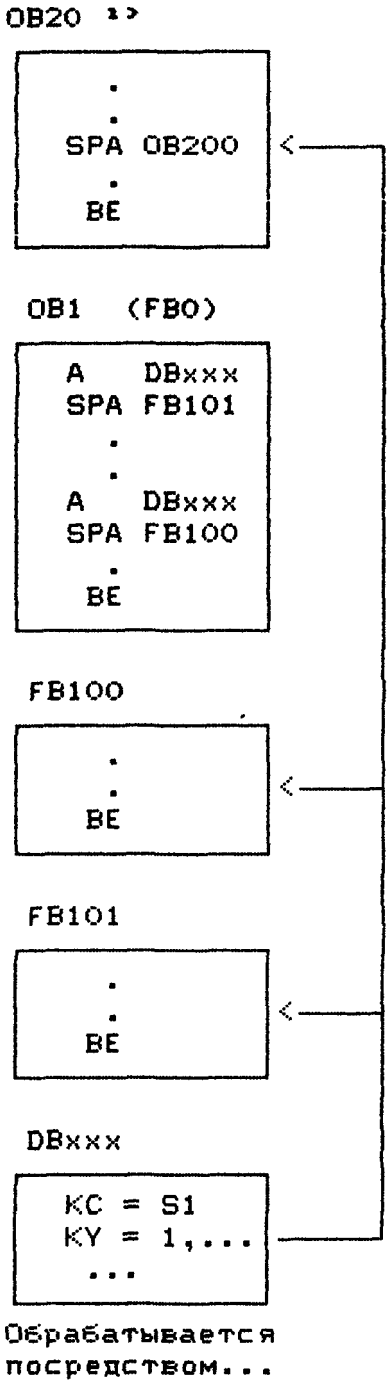
Для ПЕРЕДАЧИ и ПРИЕМА областей слов данных на каждом CPU используются два Функциональных блока:

Номер Функционального блока	Имя	Функция
FB100	SEND-DAT	Передача областей слов данных остальным CPU
FB101	EMPF-DAT	Прием областей слов данных с остальных CPU

Номера Функциональных блоков выбраны произвольно, и их можно изменять.

Функциональные блоки SEND—DAT и EMPF—DAT выбирают из списка управления связью информацию о том, какие области слов данных и из каких блоков данных должны передаваться или приниматься. Передается и принимается полная область слов данных. Если же это невозможно из-за недостаточного обеспечиваемого объема приема или передачи до тех пор, пока ресурсов не будет достаточно для всей области слов данных.

Обзор блоков, используемых в каждом CPU



Организационный блок нового пуска для предварительного распределения промежуточной памяти в координаторе KOR C

1> организационный блок OB200 может вызываться только в одном CPU.

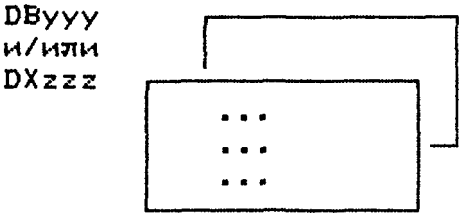
Циклическая прикладная программа, расширенная включением вызовов Функциональных блоков EMPF-DAT и SEND-DAT.

Функциональный блок: SEND—DAT
Передача областей слов данных

Функциональный блок: EMPF—DAT
Прием областей слов данных

Блок данных, содержащий список управления связью

Максимум три блока входных данных и три блока выходных данных



Внимание!
Функциональные блоки SEND—DAT и EMPF—DAT содержат организационные блоки специальных Функций поддержки многопроцессорного сопряжения — организационные блоки с OB202 по OB205. Дополнительный вызов этих организационных блоков вне SEND—DAT и EMPF—DAT недопустим!

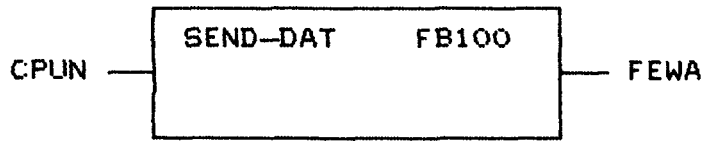
8.3.5 Передача областей слов данных <FB100>

Перед вызовом FB100 необходимо открыть ёлок данных, который содержит список управления связью. Функциональный ёлок SEND—DAT использует для обработки информации списка управления связью номер CPU, осуществляющего вызов этого функционального блока.

В случае, когда в Функциональном блоке невозможна корректная обработка Функции ПЕРЕДАЧА (SENDEN) (OB202) в выходной параметр FEMA заносится соответствующий номер ошибки или предупреждения, а VKE устанавливается в 1. Кроме того, в случае недопустимого входного параметра CPUN, параметру FEWA присваивается значение 16 (бит 4 =1).

Функциональный блок SEND—DAT использует маркерные байты с MB239 по MB251 в качестве промежуточных или интерфейсных маркеров.

Имя параметра	Значение	Вид	Тип
CPUN	Номер CPU, вызвавшего FB100. Допустимые номера; 1, 2, 3, 4.	D	KF




```
FB100                                   ДЛН=90
СЕГМЕНТ 1
ИМЯ  :SEND-DAT
МЕТКА:CPUN        E/A/D/B/T/Z: D   KM/KH/KY/KC/KF/KT/KZ/KGs KF
МЕТКА:FEMA        E/A/D/B/T/Z: A   BI/BY/W/Ds BY
000B     LW   =CPUN          CPUN = CPUN - 1
000C     L    KB 1           Ошибка в случае:
000D     -F
000E     SPM  =FEWA           номер CPU меньше 1
000F     L    KB 3
0010     >F
0011     SPB  =FEWA           номер CPU больше 4
0012     TAK
0013
0014     SLW  2               CPUN = CPUN x 4
0015     T    MB245           Базовый адрес
0016
0017     L    KB 1
0018     T    MB244           Счетчик участков связи
0019
001A SCHL:L    MB245          Базовый адрес
001B       :+F  MB244          + счетчик
001C       :+F
```

```
001D          :T      MW240
001E          :ADD    BF+16 + смещение
001 F         : T     MW242
0020          :       -
0021          :B      MW242
0022          :L      DRO Количество зарезервированных
0023          :T      MB239 блоков - О ?
0024          :L      KB 0
0025          :!=F
0026          SPB     =LEER
0027          .
0028          :B      MM242
0029          :L      DLO Номер CPU-приемника
002A          :T      MB246
002B          :L      KB246          Организационный блок спец. Функций:
002C          :SPA    OB203          ПРОВЕРКА ВОЗМОЖНОСТИ ПЕРЕДАЧИ
002D          :L      MB248
002E          :SPB    =OBFE
002 F
0030      L   MB249       Обеспечиваемый объем передачи
0031      L   MB239        не равен количеству зарезервиро
0032      ><F               ванных блоков ?
0033      SPB  =LEER
0034
0035      L   KB 0        Счетчик блоков
0036      T   MB249
0037
0038      B    MW240
0039      L    DUO        Тип и номер блока данных,
                                          являющегося источником
003A      T    MW247
003B
003C UEBT L    KB246       Организационный блок спец. функции
003D      SPA  OB202       ПЕРЕДАЧА ЗОНЫ ДАННЫХ
003E      L    MB250       Аварийное завершение по ошибке/предупреждению
003F      SPB  =OBFE
0040
0041      L    MB249       Номер блока = номер блока + 1
0042      I    1
0043      T    MB249       Пересланы все блоки ?
0044      L    MB239
0045      < F
0046      SPB  =UEBT
0047
0048 LEER L    MB244       Увеличение значения
0049      I    1          счетчика участков связи
004A      T    MB244
004B      L    KB4        Обработаны все три участка
004C      < F              связи ?
004D      SPM  =SCHL
004E      L    KBO        Нормальное завершение программы;
004F      T    =FEWA       VKE = O , FEWA = O
0050      BEA
0051
0052 FEWA L    KB16        Завершение программы в случае ошибки:
0053 OBFE T    =FEMA       VKE = 1, FEWA содержит номер
0054      BE              ошибки/предупреждения
```

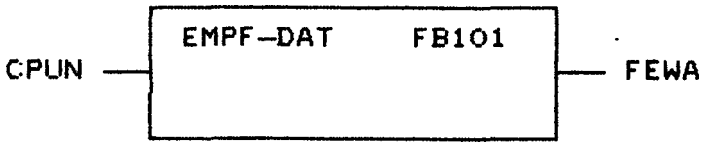
8.3.6 Прием областей слов данных (FB101)

Перед вызовом FB101 необходимо открыть блок данных, который содержит список управления связью. Функциональный блок. EMPF—DAT использует для обработки информации списка управления связью номер CPU, осуществившего вызов этого Функционального блока.

В случае, когда в Функциональном блоке невозможна корректная обработка Функции ПРИЕМ <OB204>, в выходной параметр FEWA заносится соответствующий номер ошибки или предупреждения, а VKE устанавливается равным единице. Кроме того, в случае недопустимого входного параметра PRNR параметру FEWA присваивается значение 16 (бит 4 = 1).

Функциональный блок EMPF—DAT использует байты маркеров с MB239 по MB255 в качестве промежуточных или интерфейсных маркеров.

Имя параметра	Значение	Вид	Тип
C PUN	Номер CPU, вызвавшего блок FB101. допустимые номера: 1, 2, 3, 4.	D	KF




```
FB101                                   ДЛН=88
СЕГМЕНТ 1
ИМЯ  :EMPF-DAT
МЕТКА:CPUN        E/A/D/B/T/Z: D   KM/KH/KY/KC/KF/KT/KZ/KG: KF
МЕТКА:FEMA        E/A/D/B/T/Z: A   BI/BY/W/D: BY
                                                              >

 000B    :LM  =CPUN          Ошибка в случае:
000C     :L    KB 1
 000D    :< F
 000E    :SPM =FEWA           Номер CPU меньше 1
000F     :LW  =CPUN
 0010    ;L    KB 4
 0011    :>F
 0012    ;SPB =FEWA           Номер CPU больше 4
 0013    :
 0014    ;L    KB 1          Счетчик участков связи
 0015    IT    MB242
 0016    :
 0017    :L    KB 16
 0018    :T    MW244          Указатель подсписка 2
 0019    :
```

```
001A SUCH L    MW244          Просмотр подсписка 2 до тех пор,
001B      I    1              пока не будет найдена следующая
001 C     T    MW244          запись для CPU—приемника
00ID      B    MW244          с номером
001E      L    DLO
00 1F     LW   =CPUN
0020      >< F
0021      SPB =SUCH
0022
0023      B    MW244
0024      L    DRO            Количество зарезервированных
0025      T    MB243          блоков памяти = О ?
0026      L    KB O
0027      !=F
0028      SPB  =LEER
0029
002A      L    MW244          Определение номера CPU—
002B      L  KM00000000 00001100  передатчика  по указателю под-
002D      UW                  списка 2
002E      SRM  2
002 F     I    1
0030      T    MB246
0031
0032      L    KB246          Организационный блок спец. функций
0033      SPA  OB205          ПРОВЕРКА ВОЗМОЖНОСТИ ПЕРЕДАЧИ
0034      L    MB248
0035      SPB =OBFE           Аварийное завершение по ошибке
0036
0037      L    MB249          Обеспечиваемый объем передачи
0038      L    MB243          не равен количеству зарезервиро-
0039      XF                  ванным блоков ?
003A      SPB   =LEER
003B
003C EMPF L     KB246         Организационный блок спец. Функций;
003D      SPA   OB204         ПРИЕМ ЗОНЫ ДАННЫХ
003E      L     MB248
003F      SPB  =OBFE          Аварийное завершение по ошибке
0040
0041      L     MB249         При предупреждении обрабатывается
0042      L     KBO           следующий участок связи
0043      >< F
0044      SPB  =EMPF
0045
0046 LEER L     MB242         Увеличение значения счетчика
0047      I     1             участков связи
0048      T     MB242
0049      L     KB4           Обработаны все участки связи ?
004A      <F
004B      SPM  =SLICH
004C      L     KB O          Нормальное завершение программы:
004D      T    =FEWA          VKE=0, FEWA=0 004E      BEA 004 F
0050 FEWA L     KB16          Завершение программы в случае
0051 OBFE T    =FEWA          ошибки: VKE=1, FEWA содержит
0052     :BE                   код ошибки
```

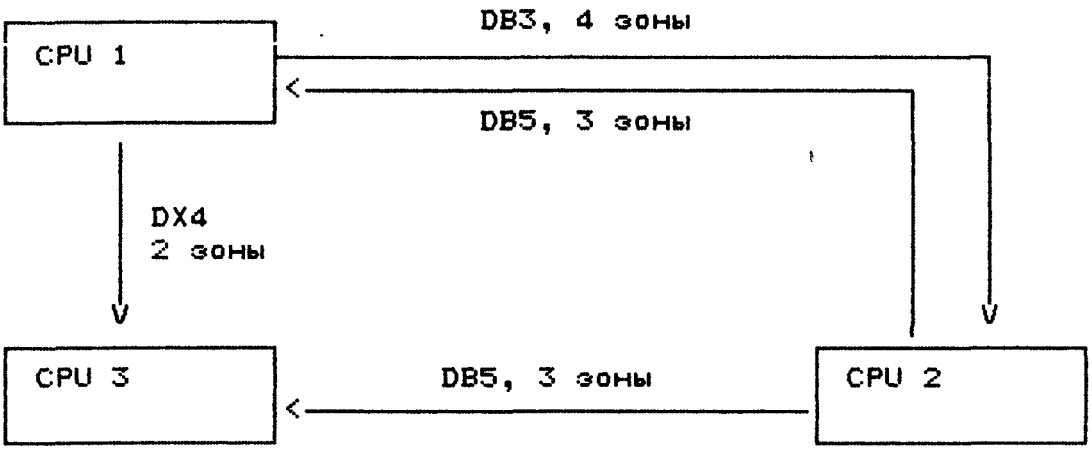
8.3.7 Пример <для AO S5-135U>

Между тремя CPU должен осуществляться следующий обмен данными;

- Из CPU 1 в CPU 2:
  Блок данных; DB3, слова данных с DWO по DW127 (=4 зоны)

- Из CPU I е CPU 3;
  Блок данных DX4, слова данных с DWO по DW63 (=2 зоны)

- Из CPU 2 в CPU 1 и в CPU 3:
  Блок данных DB5, слова данных с DWO по DW957 (=3 зоны)



Интерфейс с циклической прикладной программой на всех трех CPU должен образовывать Функциональный ёлок FBO. При перезапуске CPU 1 должен выполнять вызов Функции ИНИЦИАЛИЗАЦИЯ (OB200). Список управления связью должен размещаться в блоке данных DB100.

Перечень блоков, которые должны быть загружены в отдельные CPUs

Функция	CPU 1	CPU 2	CPU 3
Организационный блок запуска	OB20	—	—
Прикладная программа	FBO	FBO	FBO
Функц. блок SEND—DAT	FB100	FB100	FB100
Функц. блок EMPF—DAT	FB101	FB101	FB101
Список управления связью	DB100	DB100	DB100
Блоки входных данных	DB5	DB3	DBS, DX4
Блоки выходных данных	DB3, DX4	DBS	—

Сначала создается и заносится в блок данных DB100 список управления связью, структура которого была рассмотрена в Разделе "Структура данных""

```
DB100                              ДЛН=37  АБС ЛИСТ 1
        Подсписок 1 ——
  0 : КС= 51            Передать из CPU 1 ...
  1 : KY= 001,003:        ... DB3 в CPU 2
  2 : KY= 002,004:        ... DX4 в CPU 3
  3   KY= 000,000:
  4   КС= S2            Передать из CPU 2 ...
  5   KY= 001,005:        ... DBS 9 CPU 1
  6   KY= <001,005:         ... DBS 9 CPU 3
  7   KY= 000,000;
  8   КС= S3
  9   KY= 000,000;
 10   KY= 000,000:
 11   KY= 000,000;
 12   КС= S4
 13   KY= 000,000;
 14   KY= 000,000;
 1Э   KY= 000,000:
 —— Подсписок 2 ——
 16   КС= SI            Передать из CPU 1 ...
 17   KY= 002,004;        ... 4 зоны данных в CPU 2
 18   KY= 003,002;        ... 2 зоны данных в CPU 3
 19   KY= 004,000;
 20   КС= S2            Передать из CPU 2 ...
 21   KY= 001,003;        ... 3 зоны данных в CPU 1
 22   KY= 003,003;        ... 3 зоны данных в CPU 3
 23   KY= 004,000;
 24   КС= S3
 25   KY= 001,000;
 26   KY= 002,000;
 27   KY= 004,000;
 28   КС= S4
 29   KY== 001,000;
 30   KY= 002,000;
 31   KY= 003,000;
```

Список назначений, который необходим для Функции ручной инициализации — Функция инициализация (OB200), содержится в области слов данных с DW16 по DW31. Организационный блок OB200 вызывается в ходе запуска организационного блока OB20 CPU 1. Ниже приводится листинг блока OB20.

```
OB20                               ДЛН=19    АБС
СЕГМЕНТ 1
0000     L   KB2          Ручная инициализация буферов
0001     T   MB246          связи
0002
0003     L   KY1,100       Начиная со слова данных DW16,
0005     T   MW248          в блок данных DB100 занесен
0006     L   KF+16         список назначений
0008     T   MW250
0009
000A     L   KB246        Организационный блок спец. Функций;
000B     SPA  OB200         ИНИЦИАЛИЗАЦИЯ
000C
000D     BE
```

Прикладная программа на каждом CPU расширена включением вызова Функциональных блоков EMPF—DAT и SEND—DAT. Функциональный блок FBO, листинг которого приведен ниже, предназначен для CPU 1. Для выполнения этого функционального блока на других CPU достаточно всего лишь изменить соответствующим образом параметр PRNR.

```
FB0                                          ДЛН=31    ABC

СЕГМЕНТ 1

ИМЯ . :PROG-1

0005    :A   DB100          Список управления связью —> DB100
0006    :SPA  FB101         Прием блоков входных данных
0007 имя :EPMF-DAT
0008 CPUN;    KF+1
0009 FEWA:    MB0
000A    : SPA  =FEMA        Аварийное завершение по ошибке/предупреждению
000B    :
000C    :

        Здесь включается циклическая прикладная программа, считывающая
        данные из блоков выходных данных и записывающая данные в блоки
        выходных данных.
000D    :
000E    :
000F    :
0010    :A   DB100         Список управления связью -> DB100
00H    :SPA  FB100         Передача выходных блоков данных
0012 имя :SEND-DAT
0013 CPUN:    KF+l
0014 FEWA:    MBO
0015    :SPA =FEWA          Аварийное завершение по ошибке/предупреждению
0016    :
0017    :BEA
0018    :                   При ошибке/предупреждении

          Здесь вставляется обработка ошибок, (например, останов, вывод
          сообщения об ошибке на печать или монитор, или...

0019    BE
```

Важное указание!

Правильное выполнение рассмотренного примера (расширение области маркеров межпроцессорной связи посредством Функциональных блоков SEND-43AT и EMPF--DAT) возможно только в том случае, если ни на одном из CPU не выполняется дополнительно вызов организационных блоков специальных Функций поддержки многопроцессорного сопряжения — т.е. организационных блоков с OB202 по OB205.

Слово данные DW16 по DW31 содержат список соответствий, необходимый для Функции ручной ИНИЦИАЛИЗАЦИИ <'OB200>. OB200 вызывается в момент запуска из блока OB20 CPU 1.

```
OB20                ДЛН==23   АБС СЕГМЕНТ 1

0000   L   KB2         Ручная инициализация буфера связи
0001   T   MB24fc       буфера связи
0002
0003   T   KY 1,100    В DB100 вводится список соответствий,
0005   T   MM248       начиная со слова 16
0006   L   KF+16
0008   T   MM250
0009
OOOA   L   KB246       Программный 0В:
OOOB   SPA OB200       ИНИЦИАЛИЗАЦИЯ
OOOC
OOOD   UN  M252.5      Конец блока, если нет конфликта
OOOE   BEB             инициализации
OOOF
```

Здесь вводится обработка ошибок в случае конфликта инициализации (например останов, вывод сообщения на печать или...)

```
0010   BE
```

9 Литература

Необходимая информация по языку программирования STEP—5 и описание принципа работы программируемых контроллеров семейства SIMATIC S5 содержатся *в* следующих документам:

Руководство по программированию R—процессор
Каталоговый номер: C79000-B8500-C364

Руководство по программированию CPU 928
Каталоговый номер: C79000--B8500--C6.33

Руководство по программированию CPU 946/947
Каталоговый номер: C79000—B8500—C637

Руководство по эксплуатации координатора 923C
Каталоговый номер:. C79000—B8500-C349

Руководство по экспэтуатации центрального устройства 135U
Каталоговый номер:. C79000—B8500-C260

Руководство по экспэтуатации центрального устройства 155U
Каталоговый номер:. C79000—B8500-C360

SIEMENS

SIMATIC S5

ПРОГРАММИРУЕМЫЕ КОНТРОЛЛЕРЫ СЕРИИ U

Указания по монтажу Заказной No. C79000-B8500-C452-03

Предисловие

Предлагаемое описание содержит правила по установке в шкафах программируемых контроллеров серии U с центральными устройствами (EG 135U, 150U, 155U) и устройствами расширения (EG 183U, 185U, 186U и 187U).

Описание включает централизованное и децентрализованное расположение шкафов, устанавливаемые в них блоки питания и вентиляторы, заземление батарей и блоков питания, подключение периферии процесса, особенно деблокирующих входов.

Кроме того, даны указания по вентиляции шкафов, контролю температуры, подключению соединительных проводов на установке и в шкафах, а также предписания по разводке проводов. Сюда входят экранирование, заземление и длина укладываемых кабелей, меры против паразитных напряжений и скачков, а также выравнивание потенциалов.

Кроме того, в описании Вы найдете примеры конфигураций, схемы размещений, список мощностей потери модулей и таблицу по технике подключения передних разъемов.

Это описание может быть использовано обслуживающим персоналом со знаниями по SIMATIC и с допуском к работам по подключению кабелей питания.

Если у Вас появятся вопросы, которые в этом руководстве не освещены или освещены не полностью, обратитесь, пожалуйста, в ближайшее к Вам представительство Фирмы Сименс.

Указания по применению

Следующая информация должна помочь Вам в использовании предлагаемого описания.

Оглавление

* **Глава 1: Размещение в шкафах**
  Эта глава информирует о централизованном и децентрализованном размещении шкафов и о мерах по установке в них центральных устройств, устройств расширения и вентиляторов.

* **Глава 2: Электропитание**
  Здесь содержится информация о питании центральных устройств и устройств расширения, а также модулей периферии, датчиков и исполнительных устройств. Описаны питание от заземленной/незаземленной/центральнозаземленной батареи или заземленного/незаземленного/центральнозаземленного блока питания. Кроме того, питание силовой нагрузки от двух источников питания и питание деблокировочных входов модулей периферии.

* **Глава 3: Электрический монтаж установок (без учета заземления)**
  Описывает монтаж установки и необходимые поперечные сечения кабелей.

* **Глава 4: Контроль вентилятора, батареи и температурного режима**
  Рассказывает о перемычках на блоке питания и дает указания по контролю за вентилятором, батареей и температурой при установке нескольких устройств в одном шкафу.

* **Глава 5; Вентиляция шкафа и рассеиваемая мощность**
  В этой главе Вы получите информацию о вентиляции шкафа, рассеиваемой мощности модулей и максимальной температуре окружающей среды для различных видов шкафов.

* **Глава 6: Компоновка оборудования в шкафу**
  Описывает размеры отдельных устройств в шкафу.

* **Глава 7: Электрический монтаж в шкафу**
  В этой главе описывается техника подключения передних разъемов, разводка проводов внутри и снаружи шкафа. Сюда входит подключение к массе, защита при косвенных контактах, выравнивание потенциалов и меры против паразитных токов. Здесь же содержатся указания по экранированию проводов и их допустимой длине, а также меры по защите от молний.

* **Глава 8: Защита и контроль**

  Описываются меры по защите от поражений.

* **Предметный указатель**

  В конце описания Вы найдете предметный указатель. В нем собраны ключевые слова встречающиеся в этом описании с указанием глав и номеров страниц где они описаны.

* Обзор страниц

  После предметного указателя расположен обзор страниц. Он содержит информацию о страницах, на которых имеются изменения и добавления по сравнению с предыдущим изданием данной документации.

* **Формуляр "Замечания пользователя"**

  После обзора страниц расположен формуляр, предусмотренный для Ваших сообщений и предложений нам.

Обучение

Информацию об обучении, которое облегчит Вам освоение устройств Вы получите в ближайшем к Вам представительстве Фирмы Сименс.

Обзор литературы

Мы рекомендуем Вам следующую литературу по системам семейства Simatic S5:

Каталог ST54.1, Программируемые контроллеры S5-135U и S5-150U
(заказной номер; ES6010-K4654-A111-A3-7600)

Каталог IT 12, документация для SIMATIC S5
(заказной номер: ES0850-C314-X-A2)

Периферийные устройства серии U
(заказной номер: 6ES5998-0CP11)

ОГЛАВЛЕНИЕ Страница

Предисловие

Указания по применению

Глава 1 Размещение в шкафах

1.1 Централизованная схема размещения
1.2 Децентрализованная схема размещения

Глава 2 Электропитание

2.1 Встроенные источники питания
2.2 Источник питания нагрузки
2.3 Электропитание центральных устройств, устройств расширения и периферии от заземленной батареи или от заземленных блоков питания
2.4 Электропитание центральных устройств, устройств расширения и периферии от незаземленной батареи или от незаземленных блоков питания
2.5 Электропитание центральных устройств, устройств расширения и периферии от батареи с централизованным заземлением или от блоков питания с централизованным заземлением
2.6 Схема обеспечения тока нагрузки от двух блоков питания
2.6.1 Модули без потенциальной развязки
2.6.2 Модули с потенциальной развязкой
2.7 Буферная батарея
2.8 Электропитание деблокировочных входов периферийных модулей

Глава 3 Электрический монтаж установок (без учета заземления)

Глава 4 Контроль вентиляторов, батареи и температурного режима

4.1 Контроль вентиляторов, батареи и температурного режима при размещении в шкафу нескольких устройств

Глава 5 Вентиляция шкафа и рассеиваемая мощность

5.1 Рассеиваемая мощность модулей
5.2 Рассеиваемая в шкафу мощность и охлаждение шкафа

Глава 6 Компоновка оборудования в шкафу

6.1 Монтажные размеры
6.2 Примеры компоновки в шкафу

Глава 7 Электрический монтаж в шкафу

7.1 Правила подключения
7.2 Прокладка кабелей
7.3 Заземление на корпус
7.4 Защита от тока, опасного для человеческого организма, защита при косвенном контакте
7.5 Выравнивание потенциалов
7.6 Меры по защите от напряжения помех
7.7 Экранирование, ограничения по длине кабелей
7.8 Грозозащита
7.9 Подключение программаторов к сети питания

Глава 8 Зажита и контроль

Предметный указатель

Обзор страниц

Рисунки

1.1 Компактный корпус: центральное устройство и устройство расширения серии U с блоком питания и вентилятором (1)
1.2.1 Установочные размеры центрального устройства и устройства расширения (исключая EG 187U)
1.2.2 Установочные размеры центрального устройства и устройства расширения (исключая EG 187U)
1.3 Пример централизованной схемы размещения ZG 135U/150U/155U и EG 184U
1.4 Пример децентрализованной схемы размещения ZG 135U/150U/155U и EG 183U/184U
2.1 Указание; электрическое разделение
2.2 Возможные варианты подключения датчиков/исполнительных блоков периферии к заземленным блокам питания
2.3 Возможные варианты подключения датчиков/исполнительных блоков периферии к незаземленным блокам питания
2.4 Возможные варианты подключения датчиков/исполнительных блоков периферии к центральнозаземленным блокам питания
2.5 Питание периферийных модулей без потенциальной развязки от двух блоков питания
2.6 Питание периферийных модулей с потенциальной развязкой от двух блоков питания
2.7 Подключение блока питания нагрузки
2.8 Деблокировочные напряжения при сетевом питании 220В
2.9 Деблокировочные напряжения при питании 24В
2.10 Питание деблокировки
2.11 Указание; сигнал деблокировки
3.1 Электрический монтаж установок
4.1 Релейные контакты
4.2 Соединительные клеммы блоков питания устройств серии U
4.3 Напряжения питания
4.4 Расположение перемычек в блоках питания
4.5 Неисправность батареи
4.6 Один из вариантов подключения контроля вентиляторов и температурного режима при размещении в одном шкафу центрального устройства 135 U и устройства расширения 183 U

5.1 Максимальная температура окружающей среды шкафа в зависимости от рассеиваемой мощности установленного оборудования
5.2 Конструкция шкафов
6.1 Монтажные размеры
6.2 Пример компоновки и
6.2.1 Уровни крепления левая/правая сторона
7.1 Обмен сигналами при децентрализованном размещении
7.2 Размещение конденсаторов подавления помех
7.3 Меры по подавлению помех, создаваемые в шкафу люминесцентными лампами
7.4 Крепление экранов кабелей к экранной шине
7.5 Размещение элементов грозозащиты

Таблицы

1.1 Блоки питания

3.1 Площадь сечения кабеля
7.1 Подключение передних разъемов
7.2 Максимально допустимые длины кабелей цифровых сигналов при прокладке в общем канале
7.3 Длины проводов и разность потенциалов

УКАЗАНИЕ

Это руководство не может ни охватить все подробности и варианты исполнения устройств, ни предусмотреть все ситуации, которые могут возникнуть при установке, эксплуатации и обслуживании. Если Вы захотите получить дополнительную информацию или ответы на вопросы, которые возникли у Вас при работе и которых Вы не нашли здесь, обратитесь в ближайшее представительство Фирмы Сименс.

Содержание данного руководства не может изменить уже имеющиеся соглашения, обязательства или правовые отношения. Контракт содержит весь объем обязательств фирмы Сименс. Гарантии, содержащиеся в контракте, являются универсальными гарантиями фирмы Сименс. Возможные пояснения в данной книге, которые идут сверх этого не дают права на дополнительные гарантии.

1 Размещение в шкафах

Положения данного документа необходимо учитывать при проведении проектных работ в случае, когда центральные устройства (ZG 135U, 150U, l55U) и устройства расширения (EG 183U, 184U, 185U, 186U, 187U) размещаются вместе в одном или рядом стоящих шкафах. Предлагаемые правила дополняют документацию по отдельным компонентам.

Рисунок 1.1 показывает компактное устройства и устройства расширения с расположенным в нижней части блоком исполнении туда же встроен блок питания. исполнение центрального кабельным коробом и вентиляторов. В таком

Устройства расширения могут располагаться как в одном шкафу с соответствующим центральным устройством, так и в разных шкафах (централизовано или децентрализовано). Размеры устройств показаны на рисунке 1.2.

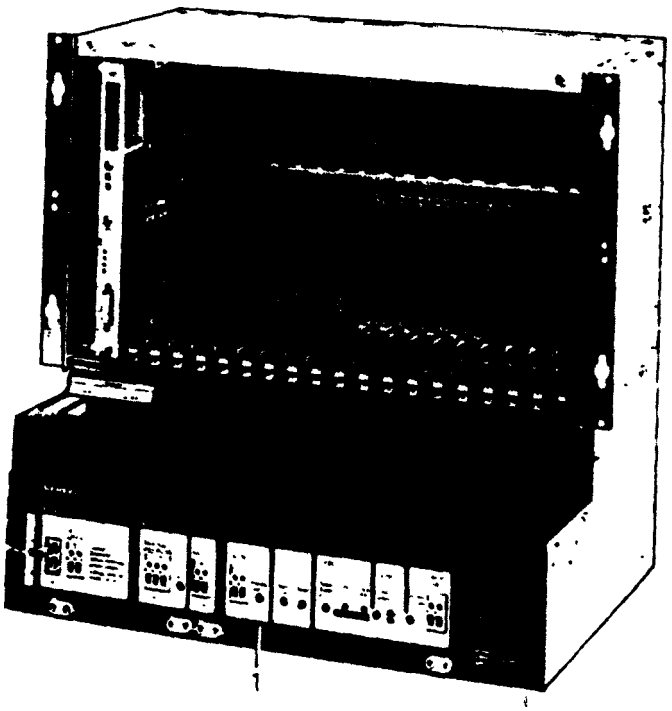


Рисунок 1.1 Компактный корпус: центральное устройство и устройства расширения серии U с блоком питания и вентилятором (1)

Каждое из устройств может быть оснащено различными блоками питания и вентиляторными блоками, которые вместе с мощностью рассеивания модулей влияют на повышение температуры внутри шкафа. Поэтому следует обратить внимание на указания в главе 5.

Если устройства расширения без блока питания подключены к центральному устройству через модули связи, то необходимо помнить, что кабель шины может выдержать не более 5А.

Входное напряжение	Выходной ток при 5В =	24В = Сумма1)	15В =	ZG 135U	ZG 150U 155U	EG 183U	EG 184U	EG 185U	EG 184U
Блок питания с вентиляторами:									
115/230В ~ потенциально развязанное	18А	0.8А	0.5А	*		*		*	
115/230В ~ потенциально развязанное	40А	2.8А	2.0А	*	*			*	
24В = потенциально связанное	10А	0.8А	0.5А	*					
24В = потенциально развязанное	18А	0.8А	0.5А	*		*		*	
24В = потенциально развязанное	40А	2.8А	2.0А	*	*			*	
Блок вентиляторов без блока питания	115/240В ~						*		
	24В =						*		
Блок питания без вентиляторов:									
115/230В ~ потенциально развязанное	15А	0.4А							*
24В =	15А	0.4А							*

1) Суммарный ток через переднюю клемму и задний разъем.

Таблица 1.1 Блоки питания.

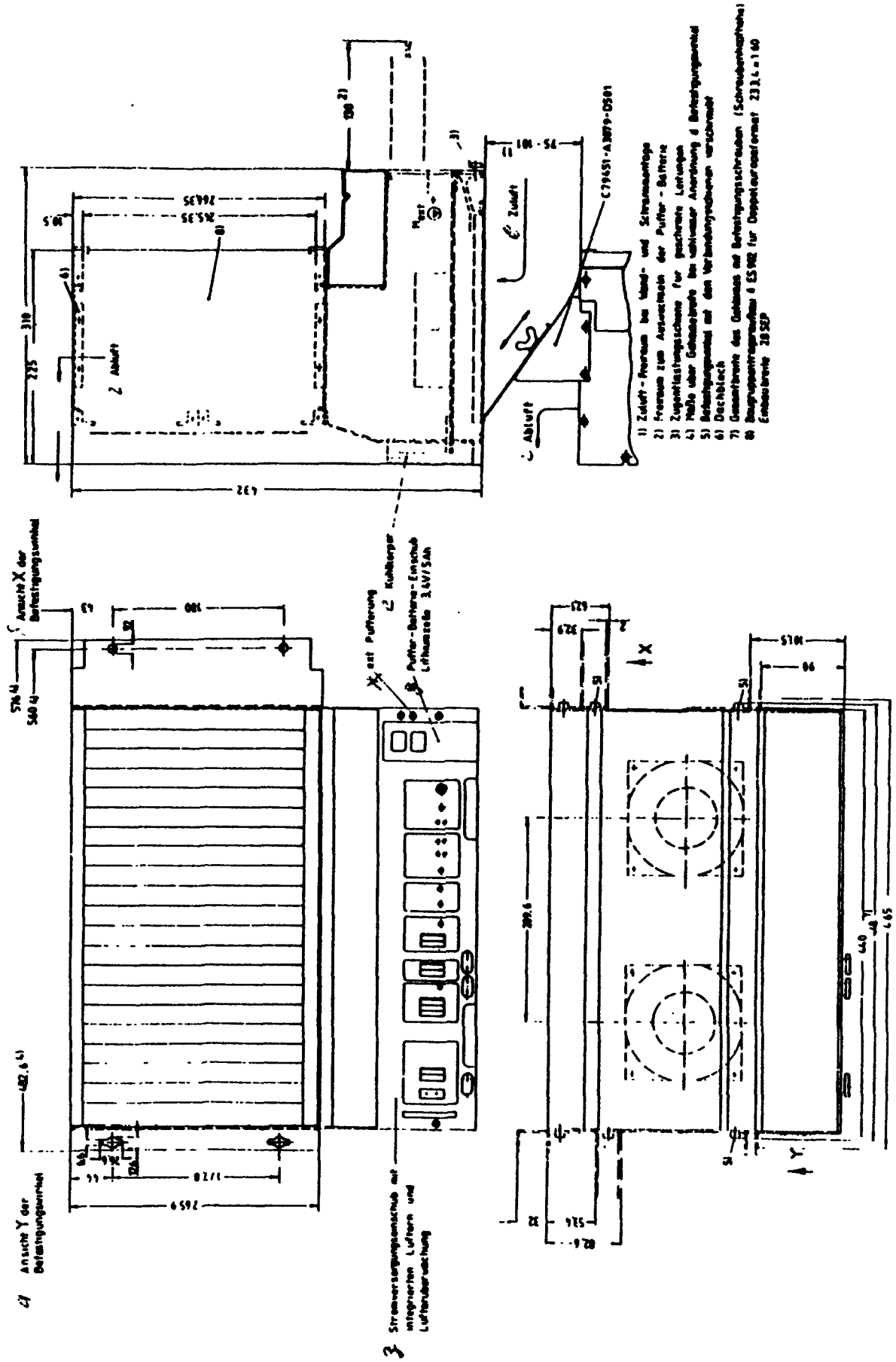


Рис. 1.2.1 Установочные размеры центрального устройства и устройства расширения (исключая EG 187U).

а - вид по оси Y крепящего уголка
б - вид по оси X крепящего уголка
в - приточный воздух
г - отводимый воздух
д - выдвижной блок буферной батареи
литиевая батарея 3,4 В / 5 А ч
е - радиатор
ж - внешняя буферизация
з - выдвижной блок питания со встроенными вентиляторами
и системой контроля вентиляторов
и - отводимый воздух

1) Свободное пространство для приточного воздуха при настенной установке или при монтаже в шкафах
2) Свободное пространство для замены буферной батареи
3) Нина снятия натяжения экранированных проводов
4) Ширина корпуса при произвольном положении крепящего уголка
5) Крепящий уголок привинчен к соединительным шинам
6) Крышка
7) Общая ширина корпуса с крепящими винтами (с учетом высоты головки винта)
8) Базовая несущая конструкция в соответствии с ES 902 для двойного европейского формата 233,4 x 160.Установочная ширина 28 SEP (SEP - стандартное установочное место)

Рис. 1.2.2 Установочные размеры центрального устройства и устройства расширения (исключая EG 187).

1.1 Централизованная схема размещения

Централизованным называется размещение, когда устройства расширения размещены в том же шкафу, что и центральное устройство или установлены в стоящем рядом шкафу. Минимальное расстояние между устройствами при такой схеме размещения равно 75 мм максимальное - 100 мм). Суммарная длина проводки от ZG до наиболее удаленного EG не должна прерывать 2 метров.

Приводимый далее пример (см. Рис 1.3) показывает один из возможных вариантов размещения ZG 135 U, 150 U или 155 U вместе с EG 184 U и соответствующими средствами подключения.

С помощью блока связи ZG 312-5 к ZG можно подключить от одного до трех EG 184 U. длина соединительного провода блока связи ZG 312-5 не должна превышать 150 см.

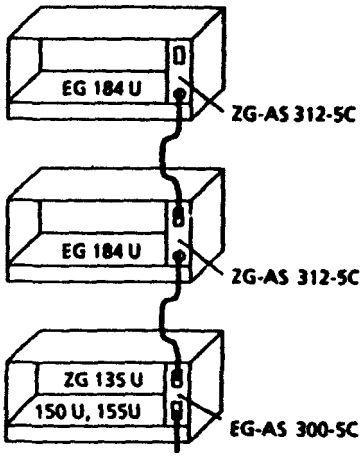


Рис.1.3 Пример централизованной схемы размещения ZG 135 U / 150 U / 155U и EG 184 U

Другие возможности сопряжения рассматриваются в каталоге ST 54.1 и в руководстве "Периферия серии U" 6ES5 998-OPC11.

1.2 Децентрализованная схема размещения

Децентрализованным называется размещение, при котором устройства расширения установлены в шкафу, который находится на значительном удалении от центрального устройства. Суммарная длина кабеля от ZG до наиболее удаленного EG не должна превышать 200 метров. Расстояние между устройствами должно быть не меньше 75 мм (макс. 100 мм). Приводимый далее пример показывает один из возможных вариантов реализации децентрализованной схемы размещения (.см. Рис.1.4), при котором подключение X EG к ZG выполняется через блоки связи EG 301-5 и блоки связи ZG/EG 310-3, а через блок связи ZG/EG 312-5 к EG 183 U можно централизованно подключить до трех дополнительных EG 184 U.

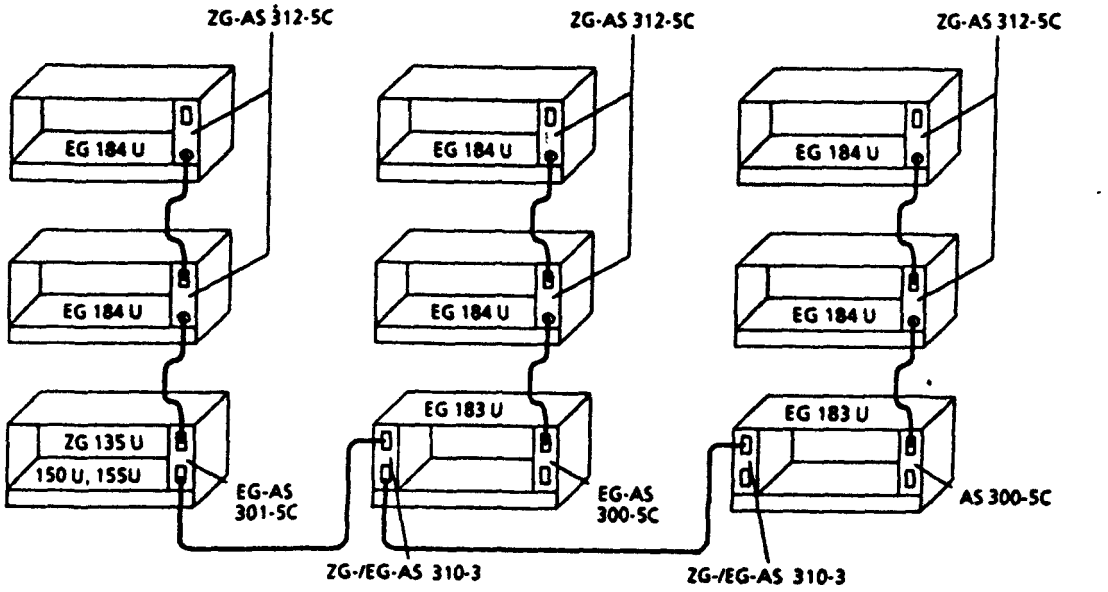


Рис. 1.4 Пример децентрализованной схемы размещения
ZG 135 U / 150 U / 155U и EG 183 U / 184 U

Обратите внимание на то, что

* в последнем блоке связи 310-3 в обязательном порядке используется замыкающий штекер
* применение блока связи устройства расширения 301-5 возможно только в сочетании с устройством расширения EG 183U, используемым в децентрализованной схеме
* в одном центральном устройстве может использоваться до четырех блоков связи устройств расширения 301
* с помощью блоков связи 304 и 314 в децентрализованной конфигурации можно достичь удаления до 600 м.

Описание возможностей сопряжения с использованием других блоков связи приводится в каталоге ST 54.1 и в руководстве "Периферия серии U" 6ES5 998-OPC11.

ВАЖНО

Для связи блоков связи между собой можно использовать только фирменный кабель.
Экран этого кабеля должен иметься с обоих сторон (не разрезан!) Низкоомное подсоединение экрана через контактную поверхность на проводящей плате и фидере массы проводящей шины не может быть оборвано.

Глава 2

Электропитание

В контроллерах SIMATIC-S5 применяются следующие типы источников питания:

- встроенные источники питания для центральных устройств и устройств расширения,
- источники нагрузочного тока для периферийных модулей (PBG), датчиков и исполнительных блоков.

2.1 Встроенные источники питания

Встроенные блоки питания центральных устройств и устройств расширения преобразуют входное напряжение 115/230 В переменного тока или 24 В постоянного тока во внутреннее постоянное напряжение 5В, 15 В и 24 В.

При комплектации центральных устройств и устройств расширения необходимо следить за тем, чтобы не превышался номинальный ток встроенных блоков питания.

Данные по потреблению тока отдельными модулями при напряжении 5 В приводятся в каталоге ST 54 (технические характеристики).

Для входного напряжения 24 В постоянного тока предусмотрены блоки питания с- и без потенциальной развязки (см. Таблицу 1). Допустимо следующее входное напряжение:

- статическое - от 20 до 30 В постоянного тока,

- кратковременный выброс - 36 В длительностью 100 мс.

Для блоков питания с номинальным входным напряжением в 115/230 В переменного тока допустимо следующее входное напряжение!

- номинальное напряжение 115 В: от 97 до 132 В перем. тока,
- номинальное напряжение 240 В: от 187 до 253 В перем. тока.

2.2 Источники питания нагрузки

Питание периферийных модулей (PBG), а также центральных устройств и устройств расширения входным напряжением 24 В постоянного тока может осуществляться от вторичных источников питания серии 6EV13... (выходной ток 20 и 40 А) фирмы SIEMENS.

Более подробную информацию по этому вопросу Вы найдете в каталоге ET 1.

При проработке вопроса использования источников тока нагрузки для модулей цифровых выходов постоянного тока (серия U) необходимо учитывать следующие моменты:

- Для защиты кабелей и проводов от тока перегрузки и для защиты модуля от короткого замыкания в модуле, наряду с электронной защитой от короткого замыкания, дополнительно используются плавкие предохранители. Плавкие предохранители также применяются для защиты от подключения с нарушением полярности выводов.

- Сопротивление R электронной защиты от короткого замыкания, которое указано в технических характеристиках модулей цифровых выходов (15 Ом для выходов 0,5 А, 4,75 Ом для выходов 2 А), соответствует максимально допустимому активному сопротивлению линии.

- При выборе источника тока нагрузки необходимо помнить о том, что тактируемая защита от короткого замыкания срабатывает только после того, .как ток на выходе кратковременно превысит, с учетом всех подключенных выходных нагрузок (учет фактора одновременности), номинальный выходной ток в 2 - 3 раза. Такое превышение по току, как правило, имеет место в нерегулируемых источниках тока нагрузки.

- Необходимо учитывать соответствующее превышение по току в регулируемых источниках тока нагрузки, особенно - при небольших выходных мощностях до 20 А.

ВНИМАНИЕ!

Для всех вторичных источников электропитания, которые используются для устройств и модулей, необходимо предусмотреть надежное электрическое разделение в соответствии с нормой VDE 0160.

Рис 2.1 Указание: Электрическое разделение

2.3 Электропитание центральных устройств, устройств расширения и периферии от заземленной батареи или от заземленных блоков питания

Следует обратить внимание на то, что внутреннее опорное напряжение (5 В) центральных устройств и устройств расширения серии U заземлено. Эксплуатация с заземлением обеспечивает лучшую, по сравнению с эксплуатацией без заземления или с эксплуатацией с централизованным заземлением, помехозащищенность. В связи с этим эксплуатация с заземлением является предпочтительной.

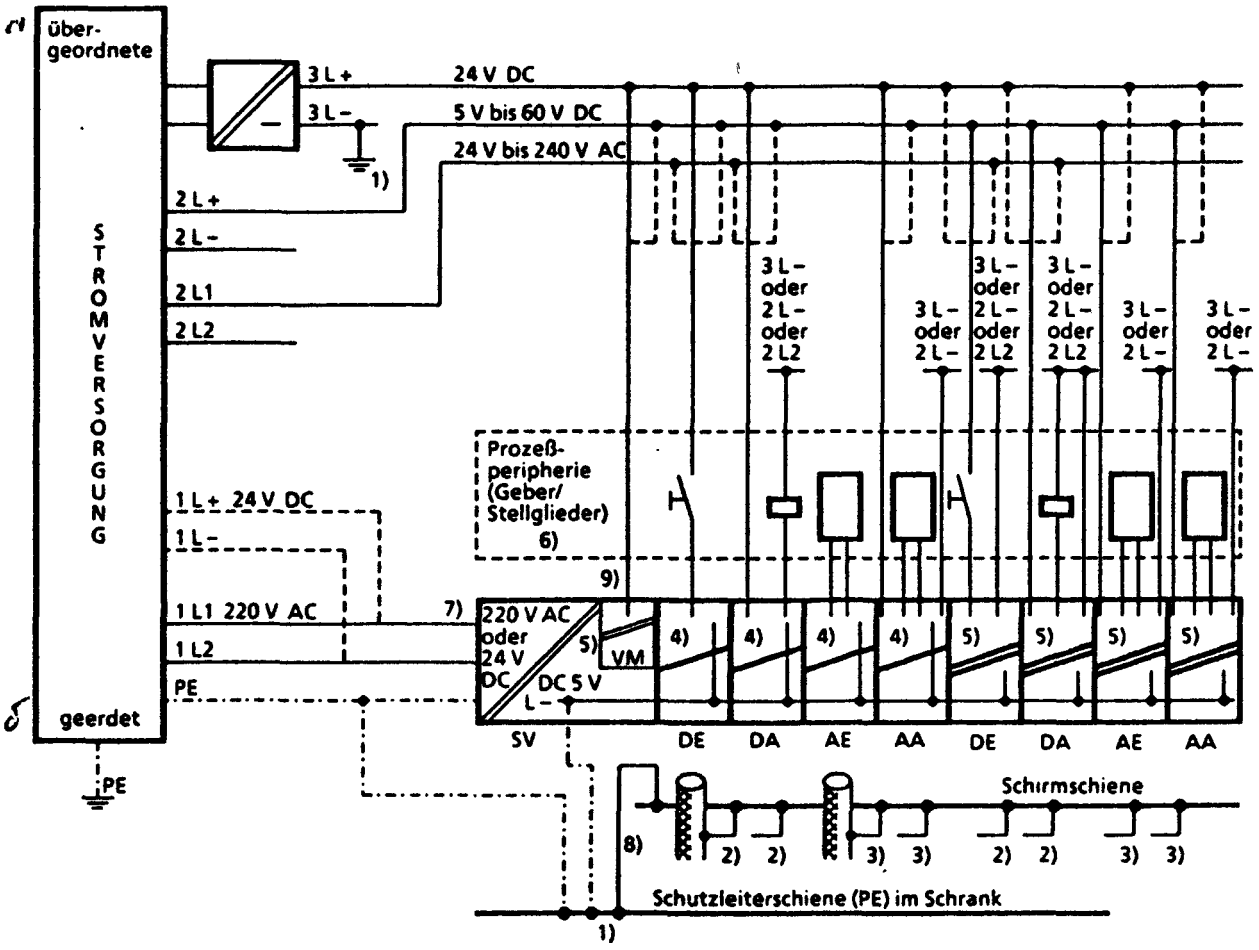


Рис. 2.2 Возможные варианты подключения датчиков/исполнительных блоков периферии к заземленным блокам питания

1) Потенциал на корпусе (потенциал шкафа) = защитный провод.
2) У цифровых модулей - экран, если он имеется. В случае относительно большой длины проводов экран необходим. Возможно одностороннее или двухстороннее подключение.
3) В случае аналоговых модулей экран подключается только с одной стороны у входа в шкаф; прокладку необходимо выполнить до самого модуля.
4) Модуль без потенциальной развязки.
5) Модуль с потенциальной развязкой.

6) Необходим защитный провод к корпусам датчиков и исполнительных блоков.
7) В блоке питания 24 В/10 А отсутствует развязка по потенциалу. Надежная эксплуатация обеспечивается только при использовании заземленного блока питания.
8) Провод с максимально возможным сечением (черного цвета); в случае, если экран .(предполагается двухстороннее подключение) используется в качестве защитного провода, то цвет зелено-желтый.
9) Контроль напряжения нагрузки L+ (24 В постоянного тока).

а- блок питания более высокого уровня
б- заземленный
в- три линии L+
г- три линии L-
д- две линии L+
е- две линии L-
ж- две линии L1
з- две линии L2
и- одна линия L+ 24 В постоянного тока
к- одна линия L-
л- одна линия L1 220 В переменного тока
м- одна линия L2
н- PE (защитное заземление)
о- PE (защитное заземление)
п- 24 В постоянного тока
р- от 5 В до 60 В постоянного тока
с- от 24 В до 240 В переменного тока
т- три линии L- или две линии L- или две линии L2
у- три линии L- или две линии L-
ф- три линии L- или две линии L- или две линии L2
х- три линии L- или две линии L- или две линии L2
ц- три линии L- или две линии L-
ч- три линии L- или две линии L-
ш- периферия процесса (датчики/исполнительные блоки)
щ- 220 В переменного тока или 24 В постоянного тока
э- постоянный ток, 5 В
ю- вольтметр
я- блок питания

аа - цифровой вход
аб - цифровой выход
ав - аналоговый вход
аг - аналоговый выход
ад - цифровой вход ,
ае - цифровой выход
аж - аналоговый вход
аз - аналоговый выход
аи - экранированная шина
ак - шина защитного провода (PE) в шкафу

2.4 Электропитание центральных устройств, устройств расширения и периферии от незаземленной батареи или от незаземленных блоков питания

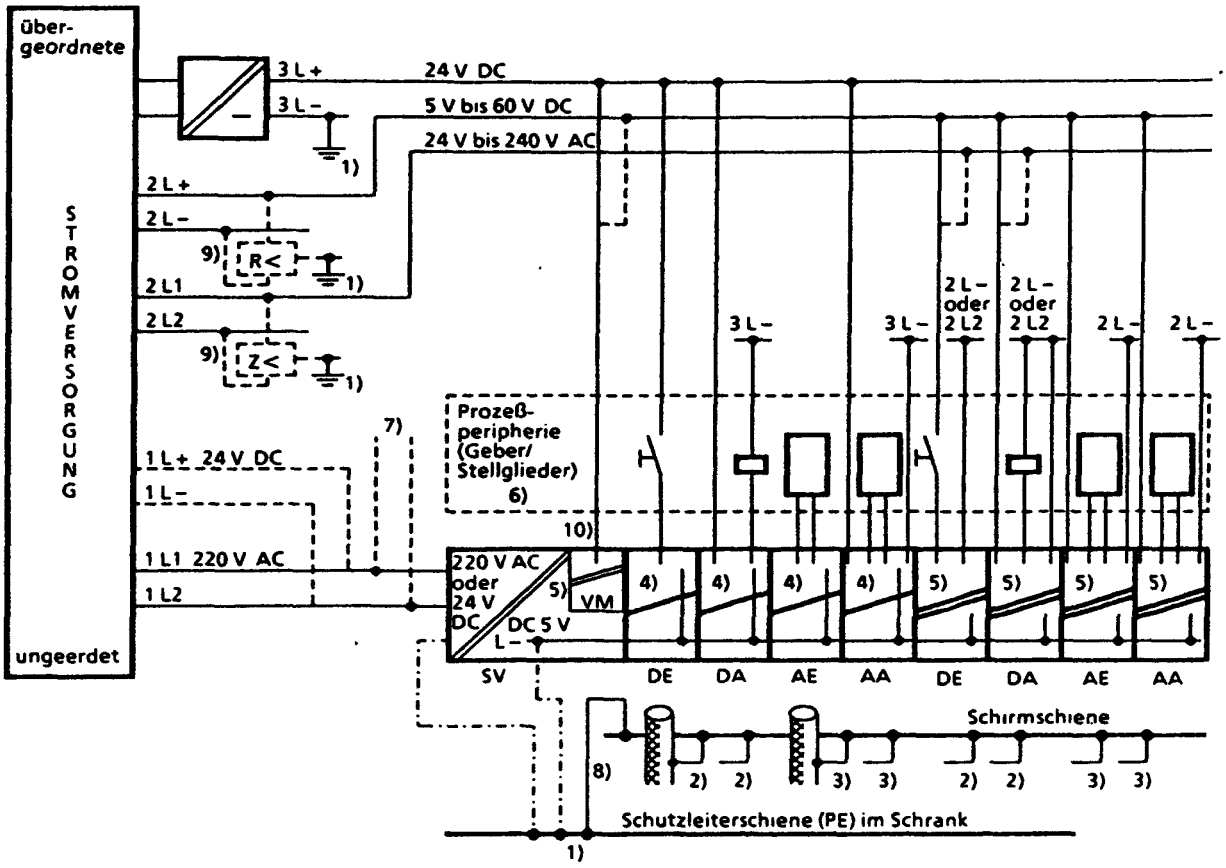


Рис. 2.3 Возможные варианты подключения датчиков/исполнительных блоков периферии процесса к незаземленным блокам питания

1) Потенциал на корпусе (потенциал шкафа) = защитный провод.
2) У цифровых модулей - экран, если он имеется. В случае относительно большой длины проводов экран необходим. Возможно одностороннее или двухстороннее подключение проводов.
3) Экран аналоговых модулей подключается только с одной стороны у входа в шкаф; прокладку нужно вести до самого модуля.
4) Модуль без потенциальной развязки.
5) Модуль с потенциальной развязкой..
6) Необходим защитный провод к корпусам датчиков и исполнительных блоков. При небольших рабочих напряжениях необходимость в защитном проводе отпадает.
7) В блоке .питания 24В/10А нет потенциальной развязки. Поэтому непосредственное подключение к незаземленному блоку питания невозможно. Необходим подвод напряжения через 3 L+/-.
8) Провод с максимально возможным сечением (черного цвета);
в случае, когда экран (двухстороннее подключение) используется в качестве защитного провода, - цвет зелено-желтый.
9) Контроль напряжения нагрузки L+ (24В постоянного тока).
10) В случае, когда из-за двойной ошибки возможны опасные перемещения, а также при напряжениях, превышающих 50 В переменного тока или 120 В постоянного тока, необходимы средства контроля изоляции. Для одной системы электроснабжения необходим только один комплект средств контроля изоляции.

а - блок питания более высокого уровня
б - не заземленный
в - три линии L+
г три линии.L-
д - две линии L+
е - две линии L-
ж - две линии L1
з - две линии L2
и - одна линия L+ 24 В постоянного тока
к - одна линия L-
л - одна линия L1 220 В переменного тока
м - одна линия L2
н - 24 В постоянного тока
о - от 5 В до 60 В постоянного тока
п - от 24 В до 240 В переменного тока
р - три линии L-
с - три линии L-
т - две линии L- или две линии L2
у - две линии L- или две линии L2
ф - две линии L-
ц - две линии L-
ч - периферия процесса (датчики/исполнительные блоки)
ш - 220 В переменного тока или 24 В постоянного тока
щ - постоянный ток, 5 В
э - волтьтметр
я - блок питания

аа - цифровой вход
аб - цифровой выход
ав - аналоговый вход
аг - аналоговый выход
ад - цифровой вход
ае - цифровой выход
аж - аналоговый вход
аз - аналоговый выход
аи - экранированная шина
ак - шина защитного провода (PE) в шкафу

2.5 Электропитание центральных устройств, устройств расширения и периферии от батареи с централизованным заземлением или от блоков питания с централизованным заземлением

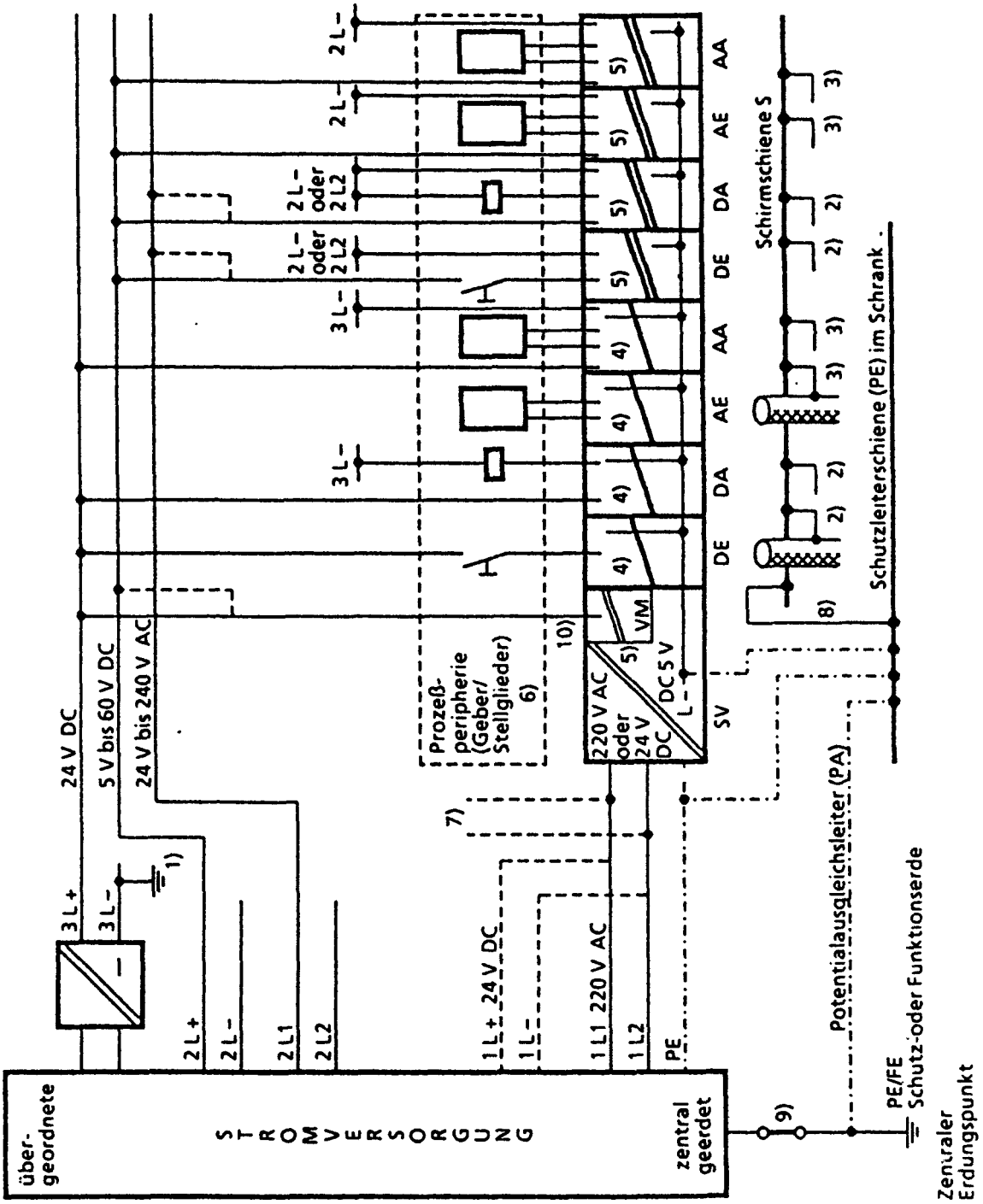


Рис. 2.4 Возможные варианты подключения датчиков/исполнительных блоков периферии процесса к блокам питания с централизованным заземлением

1) Потенциал на корпусе (потенциал шкафа) = защитный провод.
2) У цифровых модулей - экран, если он имеется. В случае относительно большой длины проводов экран необходим. Возможно одностороннее или двухстороннее подключение проводов.
3) Экран аналоговых модулей подключается только с одной стороны у входа в шкаф; прокладку нужно вести до самого модуля.
4) Модуль без потенциальной развязки.
5) Модуль с потенциальной развязкой.
6) Необходим защитный провод к корпусам датчиков и исполнительных блоков. При небольших рабочих напряжениях необходимость в этом защитном проводе отпадает.
7) В блоке питания 24В/10А нет потенциальной развязки. Поэтому непосредственное подключение к незаземленному блоку питания невозможно. Необходим подвод напряжения через 3 L+/-.
8) Соединительный провод с максимальным сечением (черного цвета); в случае, когда экран (двухстороннее подключение) используется в качестве защитного провода, - цвет зелено-желтый.
9) Контроль напряжения нагрузки L+ (24 В постоянного тока).
10) Разъемное соединение для измерений.

а - блок питания более высокого уровня
б - централизованное заземление
в - три линии L+
г - три линии L-
д - две линии L+
е - две линии L-
ж - две линии L1
з - две линии L2
и - одна линия L+ 24 В постоянного тока
к - одна линия L-
л - одна линия L1 220 В переменного тока
м - одна линия L2
н - PE
о - линия выравнивания потенциалов (PA)
п - PE/FE защитное заземление или функциональное заземление
р - централизованная точка заземления
с - 24 В постоянного тока
т - от 5 В до 60 В постоянного тока
у - от 24 В до 240 В переменного тока
ф - три линии L-
х - три линии L-
ц - две линии L- или две линии L2
ч - две линии L- или две линии L2
щ -две линии L-
щ - две линии L-
э - периферия процесса (датчики/исполнительные блоки)
ю - 220 В переменного тока или 24 В постоянного тока
я - постоянный ток, 5 В
ъ - вольтметр
аа - блок питания
аб - цифровой вход
ав - цифровой выход
аг - аналоговый вход
ад - аналоговый выход
ае - цифровой вход
аж - цифровой выход
аз - аналоговый вход
аи - аналоговый выход
ак - экранированная шина S
ал - шина защитного провода (PE) в шкафу

2.6 Схема обеспечения тока нагрузки от двух блоков питания

В приводимых далее двух примерах на рис. 2.4, 2.5) рассматривается схема питания входов и выходов разных модулей от двух блоков питания.

2.6.1 Модули бед потенциально развязки

В связанных по потенциалу модулях входов-выходов отрицательный полюс (L-) блока питания должен соединяться с опорным потенциалом (корпус устройств серии U или корпус шкафа). Это необходимо, т.к. входы модуля 420, который фигурирует в примере, связаны с массой (корпус устройств серии U).

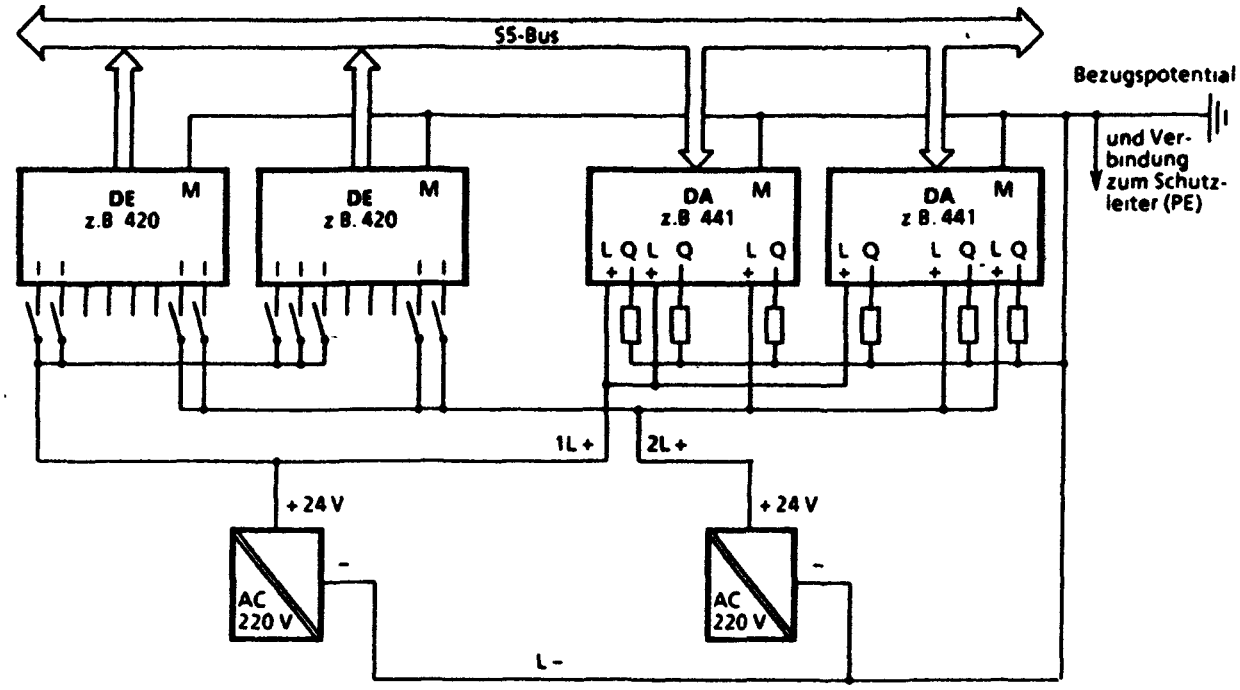


Рис. 2.5 Питание периферийных модулей без потенциальной развязки от двух блоков питания

а - шина S5
б - опорный потенциал
в - и подключение к защитному проводу (PE)
г- цифровой выход, например, 441
д - цифровой выход, например, 441
е - цифровой вход, например, 420
ж - цифровой вход, например, 420
з - одна линия L+
и - две линии L+
к - + 24 В
л - блок питания - переменный ток, 220 В
м - линия L-
н - блок питания - переменный ток, 220 В
о - + 24 В

2.6.2 Модули с потенциальной развязкой

Питание модулей входов и выходов с потенциальной развязкой от двух блоков питания может выполняться путем разбивки на развязанные по потенциалу группы.

Необходимо обратить внимание на то, что подключение входов или выходов двух

развязанных по потенциалу групп к одному блоку питания приведет к потере развязки по потенциалу между группами.

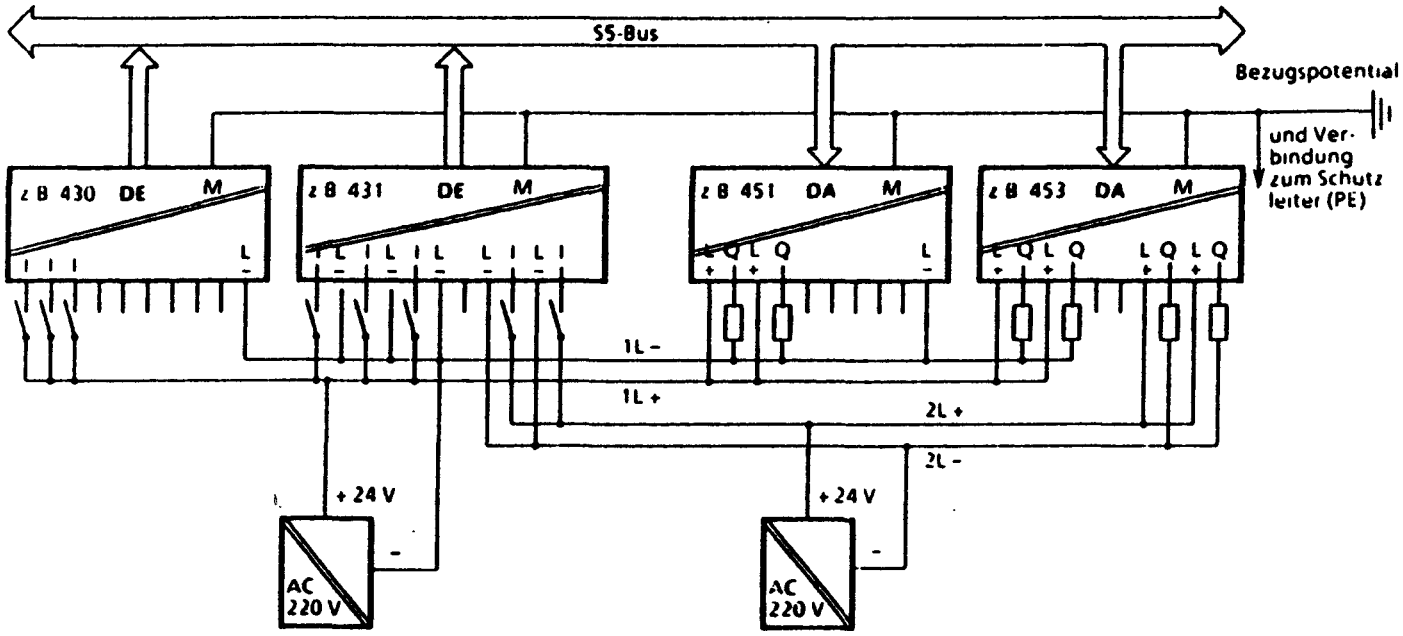


Рис. 2.6 Питание периферийных модулей с потенциальной развязкой от двух блоков питания

а - шина S5
б - опорный потенциал
в - и подключение к защитному проводу (PE)
г- - цифровой выход, например, 453
д - цифровой выход, например, 451
е - цифровой вход, например, 431
ж - цифровой вход, например, 430
з - одна линия L-
и - одна линия L+
к - две линии L+
л - две линии L-
м - + 24 В
н - блок питания - переменный ток, 220 В
о - + 24 В
п - блок питания - переменный ток, 220 В

В случае использования центральных устройств и устройств расширения с подключением к сети для электропитания периферии процесса (напряжение нагрузки) рекомендуется применять вторичный источник питания серии 6EV13... или 6ES5950-8MD ... фирмы СИМЕНС с гальванической развязкой.

Напряжение нагрузки 24 В должно подключаться к входу контроля напряжения (Monitor Input).

2.7 Буферная батарея

В центральных устройствах 135 U, 15O U и 155 U рядом с выдвижным блоком питания в устройствах размещается буферная батарея. Замену батареи можно выполнять без потери содержимого памяти, если замена осуществляется при включенном сетевом питании или при подведенном к клеммам "Ext. Batt." ("Внешняя батарея") внешнем напряжении (3,4 В),

ВАЖНО

Обратите внимание на полярность внешнего напряжения. Неисправность батареи может при желании быть определена с помощью реле сообщений ("Monitor Output").

2.8 Электропитание деблокировочных входов периферийных модулей

В состав модулей серии U включена схема отключения. Через деблокировочные входы можно выполнять блокировку определенных модулей или осуществлять отключение отдельных модулей в ходе работы программируемого контроллера. Это возможно во время работы.

В модулях с входами или выходами постоянного тока отключение выполняется путем подачи внешнего напряжения на входы F+/F-.

В модулях с входами или выходами переменного тока отключение выполняется с помощью перемычки на переднем разъеме.

При отсоединении переднего разъема от лицевой панели модуля прекращается подача напряжения на деблокировочный вход. Модуль отключается и перестает выдавать квитирующий сигнал на запрос CPU. При этом возникает ошибка "Задержка квитирования" (QVZ)

ВАЖНО

CPU 921 (S-процессор) при QVZ переходит в состояние СТОП.
CPU 922 (R-процессор), 928, 946/947 (AG 155U) и AG 150U продолжают работать или могут начать выполнение программы, записанной в 0B-интерфейсе вызываемом при QVZ. Если периферийный модуль отсутствует или не отвечает на запрос, то CPU 923 и CPU 946/947 для всех неисправных входов определяет состояние "O" в области отображения процесса. Все другие CPU определяют их как "1".

Примеры использования блокирующих входов:

- Выполнение отключения отдельных подпроцессов с минимальным потреблением мощности. Это значит, что работа с выходами различных модулей может обеспечиваться общим блоком питания, при этом выходы модулей переводятся в активное состояние раздельно.
- Контроль напряжения нагрузки каждого отдельного модуля без каких-либо дополнительных затрат. В организационном блоке QVZ можно запрограммировать любую реакцию на исчезновение напряжения нагрузки.

При проектировании установок необходимо учитывать следующее;

Включение: Напряжение на деблокировочных входах модулей входов и выходов должно подаваться не позже, чем через 100 мС после включения программируемого контроллера.

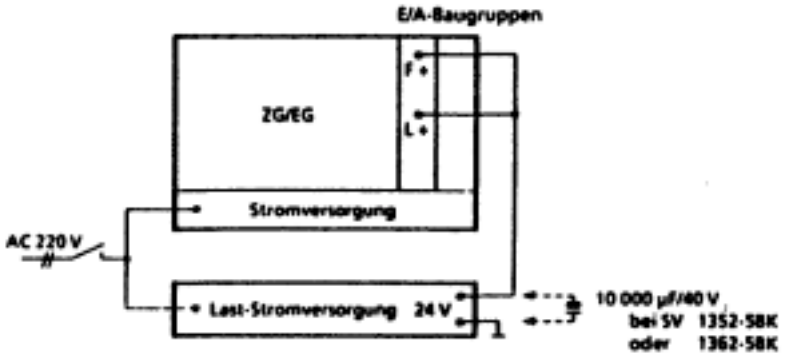
Выключение: После выключения программируемого контроллера напряжение на деблокировочных входах модулей входов и выходов должно сохраняться до тех пор, пока обеспечивается внутреннее напряжение 5 В.

Указания по отключению программируемых контроллеров и источников питания деблокировочных входов:

1. Общее отключение центрального устройства/устройства расширения и питания сети нагрузки при сетевом питании 220 В

При использовании блока питания нагрузки 6EV1334-4AK (220В/24В 10 А) обеспечивается нормальное функционирование, так как питание нагрузки 24 В имеет выходную емкость не менее 4700 мкФ. при токе нагрузки 10 А (номинальный ток).

При использовании других блоков питания, которые не удовлетворяют этому условию, следует выполнить параллельное подключение конденсатора 10000 мкФ/40 В. Параллельное подключение конденсатора следует выполнить при использовании блока питания нагрузки 6EV1352-5BK (380 В/24 В, 20 А) и блока питания нагрузки 6EV1362-6BK (380 В/24 В, 40 А).



а - модули входов/выходов
б - центральное устройство/ устройство расширения
в - блок питания
г - сеть переменного тока 220В
д - блок питания нагрузки 24 В
е - 10 000 мкФ/40 В

для блока питания 1352-5BK или блока питания 1362-6BK

Рис. 2.7 Подключение блока питания нагрузки

2. Раздельное или общее отключение центрального устройства/устройства расширения и питания сети нагрузки

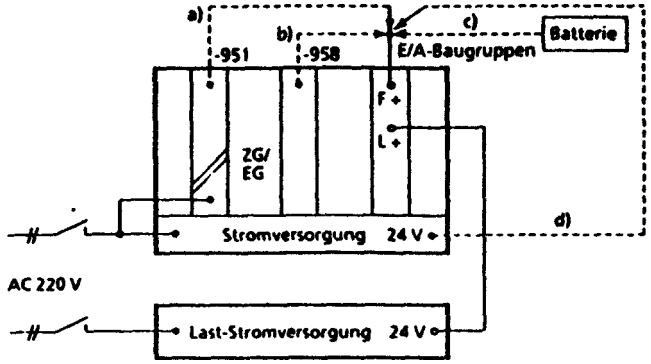
В случае, когда должна обеспечиваться возможность отдельного отключения питания сети нагрузки без изменения состояния деблокировочных входов модулей, необходимо предусмотреть меры по поддержке деблокировочного напряжения. Рассматриваемые далее варианты поддержки деблокировочного напряжения могут использоваться также в случае применения блока питания нагрузки без дополнительного конденсатора и при общем отключении.

o Питание от сети 220 В для центрального устройства/устройства расширения и блока питания нагрузки

Питание деблокировочных входов модулей входов и выходов осуществляется от:

a) Блока питания нагрузки 6ES5951-4LB11

в) Блока питания деблокировочных входов 6ES5958-4UA1I c) Батареи
d) Питание деблокировки непосредственно от источника питания
(следите за нагрузкой и MLFB-номером)

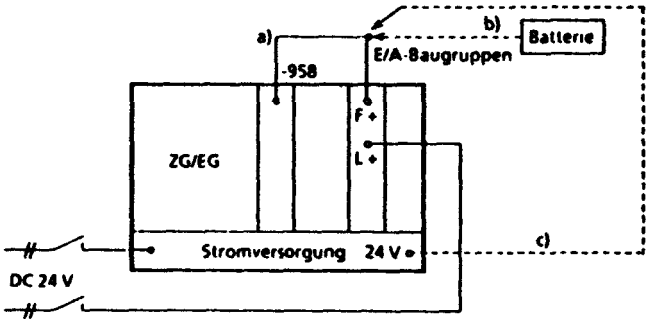


1 - батарея
2 - модули входов/выходов
3 - центральное устройство/ устройство расширения
4 - блок питания
5 - сеть переменного тока 220 В
6 - блок питания нагрузки 24 В

Рис. 2.8 Деблокировочные напряжения при сетевом питании 220 В

o Питание 24 В для центрального устройства/устройства расширения и периферии

Питание деблокировочных входов модулей входов и выходов осуществляется от:
a) Блока питания деблокировочных входов 6ES595S-4UA11
в) Батареи
c) Питание деблокировки непосредственно от источника питания



1 - батарея
2 - модули входов/выходов
3 - центральный устройство/ устройство расширения
4 - блок питания
5 - 24 В постоянного тока

Рис. 2.9 Деблокировочные напряжения при питании 24 В

Следует помнить, что блок питания деблокировочных входов 6ES5958-4UA11 может подключаться только к центральным устройствам и устройствам расширения, которые оснащены блоком питания.

Подключение деблокировочных вводов модулей вводов и выходов в устройстве расширения без блока питания (например, устройства расширения EG 184 U, EG 187 U) должно осуществляться через блок питания деблокирующих входов 958. Этот блок устанавливается в устройстве с блоком питания.

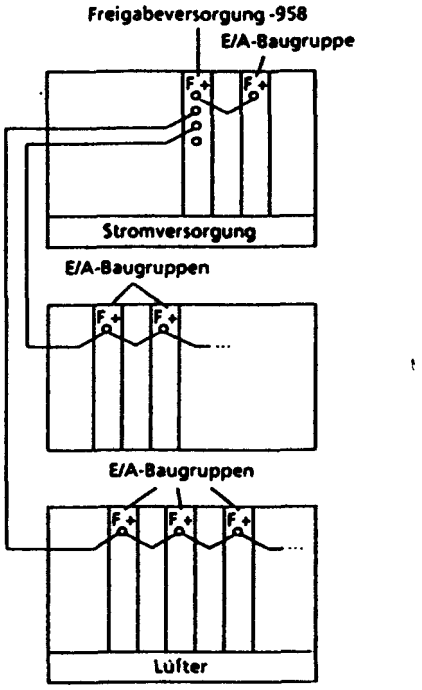


Рис. 2.10 Питание деблокировки

а - блок питания деблокировочных входов 958
б - модуль входов/выходов
в - блок питания
г - модули входов/выходов
д - модули входов/выходов
е - вентилятор

Снятие и установка периферийных модулей в ходе эксплуатации

В устройствах автоматизации - программируемых контроллерах 130 K, 130 W и 135 U (только S-процессор) отсоединение переднего разъема (прекращение подачи напряжения на деблокировочные входы) приводит к останову программируемого контроллера с задержкой квитирования.

Снятие и установка периферийных модулей возможны в ходе эксплуатации на следующих типах программируемых контроллеров:
AO 135U (CPU 922/ R-процессор, CPU 928), AG 150K, AG 150S, AG 150U и AG 155U (CPU 946/947).

ВНИМАНИЕ!

При отсутствующем деблокировочном сигнале, при отключенном переднем разъеме или при снятом модуле со всех входов соответствующего модуля считывается сигнал 1. Однако CPU 928 и CPU 946/947 (AG 155U) записывают в этом случае в область отображения входов состояние сигнала "О".

Для отключения подпроцессов достаточно прекратить "подачу деблокировочного напряжения на соответствующий модуль выходов!

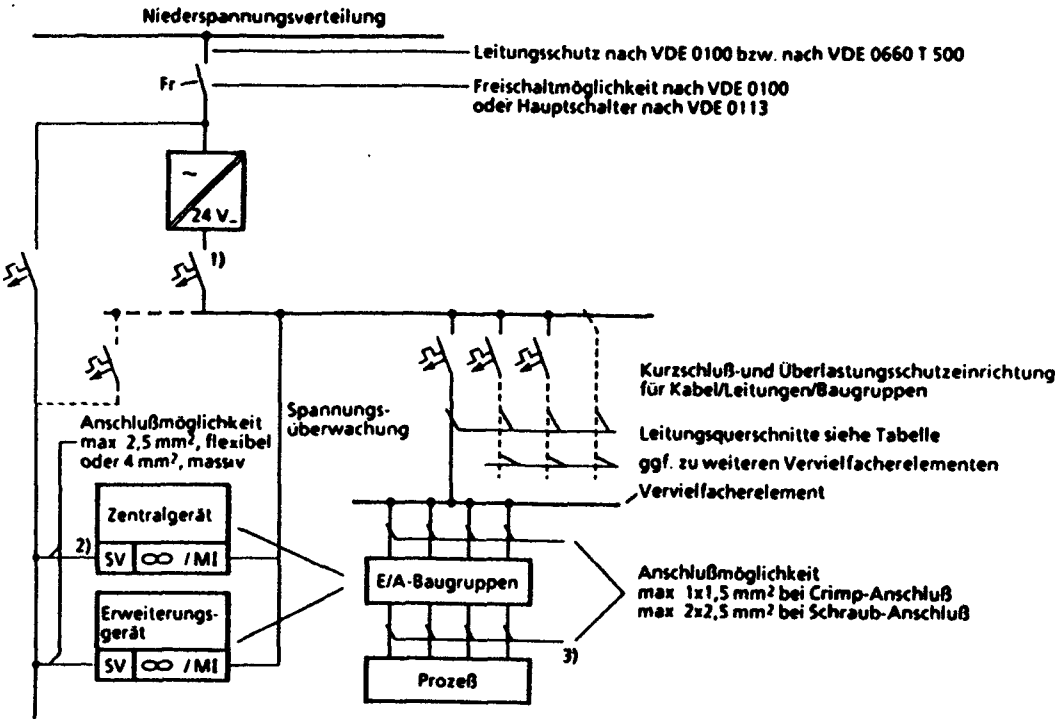
Программист путем соответствующего программирования организационного блока обработки QVZ Должен предотвратить возникновение опасных состояний процесса и станка в случае ошибки или при замене модуля.

Глава 3

Электрический монтаж установок (без учета заземления)

Обратите внимание, что при использовании для питания периферии напряжения 24 В постоянного тока и для питания центрального устройства и устройства расширения - напряжения 240 В переменного тока или напряжения 24 В постоянного тока, необходимо пользоваться только следующим подключением.

При питании периферии процесса напряжением 240 В переменного тока используется то же принципиальное решение. Соответствующие изменения значений сведены в Таблицу 3.1.



1) Может отсутствовать при наличии защитного провода в блоке питания.
2) Потребление тока каждым центральным устройством и устройством расширения:
Например, при 24 В постоянного тока (1_A = 18 А) будет потребляться около 6,9 А, а при 240 В переменного тока (1_A = 18 А) будет потребляться около 1,2 А.

3) В случае выбора подходящих диаметров сечений и длин проводов защита выходных линий от короткого замыкания и перегрузки может быть обеспечена предохранителями, которые находятся на модулях выходов. В отличии от выходных линий для входных сигнальных линий всегда требуется дополнительная защита, которая может быть осуществлена с помощью выхода.

Рис. 3.1 Электрический монтаж установок

а - низковольтная распределительная система
б - защита сети согласно норме VDE 0660 T 500
в - отключение
г - возможность отключения согласно норме VDE 0100 или
главный выключатель согласно норме VDE 0113
д - 24 В постоянного тока
е - возможные варианты подключения:
гибкий кабель - максимум 2,5 мм^2
кабель для фиксированного монтажа - 4 мм^2
ж - центральное устройство
з - блок питания
и - устройство расширения
к - блок питания
л - контроль напряжения
м - модули входов/выходов
н – процесс
о - защита от короткого замыкания и перегрузки для кабелей/проводов/модулей
п - площадь сечения проводов приводится в таблице
р - возможно - к другим умножителям
с - умножитель
т - возможности подключения:
максимум 1 х 1,5 мм^2 при зажимном подсоединении максимум 2 х 2,5мм^2 при винтовом подсоединении

Номинальный ток элемента защиты	< 2 2. . .6	6...10	10...16	16...25[1)]	25...40[1)]
Длина кабеля (м)	Площадь сечения кабеля (мм^2)				
< 1	1 1,5	2,5	4	6	10
1. . .2,5	1,5 2,5	4	6	10	16

[1>] При выполнении защиты под указанные значения тока провода от умножителя *к* модулю должны, при определенных обстоятельствах, прокладываться с защитой от короткого замыкания.

Таблица 3.1 Площадь сечения кабеля

Глава 4
Контроль вентиляторов, батареи и температурного режима

В компактных устройствах серии U аккумуляция тепла предотвращается вентиляторами, которые входят в состав выдвижного блока питания и располагаются в нижней части устройства.

Для обеспечения постоянного достаточного теплоотвода, в каждом устройстве имеются два вентилятора. Осуществляется контроль потока воздуха.

Сигнал о остановке одного или обоих вентиляторов и (если задано) неисправности батареи выдается релейным контактом "Monitor Output" ("Контрольный выход"), см. Рис. 4.2) и светодиодом:

- Замкнут релейный контакт 1-2: вентилятор работает и (если задано) батарея в порядке
- Замкнут релейный контакт 2-3: дефект вентилятора и (если задано) неисправна батарея

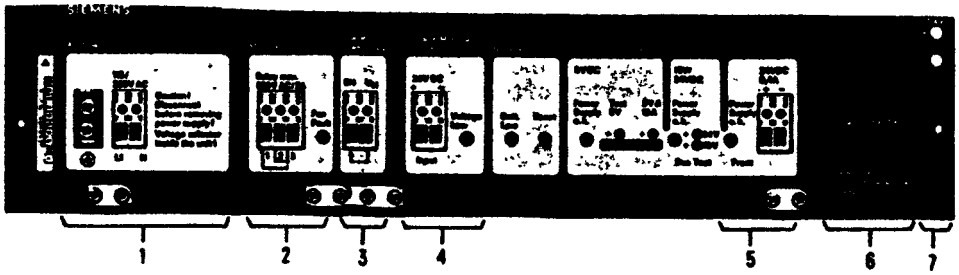


Рис 4.1 Релейные контакты

1 Напряжение питания 115/230 В переменного тока или 24 В постоянного тока
2 Сигнальный выход контроля вентиляторов ("Monitor Output") ("Контрольный выход") и (если задано) контроль батареи.
3 Разрешение внутреннего напряжения питания + 5 В/+ 15 В/+ 24 В (напр. установка перемычки E_N-U_N)
4 Контроль напряжения нагрузки ("Voltage Monitor") ("Контроль напряжения")
5 Питание деблокировки 24 В
6 Буферная батарея
7 Внешняя батарея

Рис. 4.2 Соединительные клеммы блоков питания устройств серии U

С помощью перемычки F-R (см. Рис.4.4) в *блоке* питания можно определить реакцию контроля потока воздуха на отказ вентилятора, а именно - будет эти схема контроля потока воздуха при отказе вентилятора отключать внутреннее напряжений питания U_A (5 В) или нет:

Перемычка F-R	Информация
установлена	отключение U_A.
не установлена	отключение U_A не производится (сообщение через релейный контакт).

Рис. 4.3 Напряжения питания

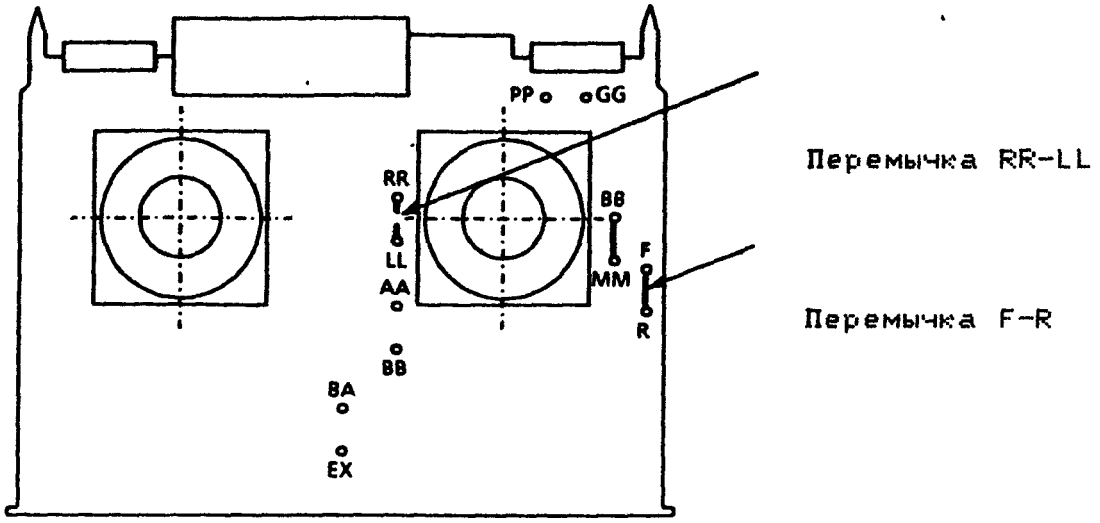


Рис. 4.4 Расположение перемычек в блоках питания (напр.6ES5 955-3LC12)

УКАЗАНИЕ

Перемычка F-R не устанавливается в случае, когда немедленное отключение невозможно по каким-либо особым причинам. В этом случае необходимо позаботиться о том, чтобы питание отключалось другим способом не позднее чем через 60 секунд, например, с помощью реле времени (см. Рис.4.5).

С помощью перемычки RR-LL вы можете выбрать, должно ли срабатывать реле сообщений ("Monitor- Output") не только при неисправности вентилятора, но и при неисправности батареи:

Перемычка RR-LL	Информация
разомкнута (состояние поставки)	реле сообщений срабатывает только при неисправности вентилятора
замкнута	реле сообщений срабатывает при неисправности батареи и вентилятора

Рис. 4.5 Неисправность батареи

Функциональное назначение и положение остальных перемычек описывается в Руководстве по эксплуатации соответствующего блока питания.

4.1 Контроль вентиляторов, батареи и температурного режима при размещении в шкафу нескольких устройств

В случае, когда должен обеспечиваться общий контроль нескольких устройств, оснащенных блоками вентиляторов, для контроля вентиляторов необходимо, в соответствии с приводимой далее схемой, выполнить подключение к клеммам EN и U_N.

При отказе вентилятора отключаются все устройства.
Отключение устройств выполняется термоконтактами, или средствами контроля вентиляторов. Термоконтакты расположены в теплообменнике, в потоке отводимого воздуха верхнего устройства. Термоконтакты представляют собой реле времени К, с задержкой размыкания.

Помните, что один выход Un можно использовать для управления не более чем семью входами EN.

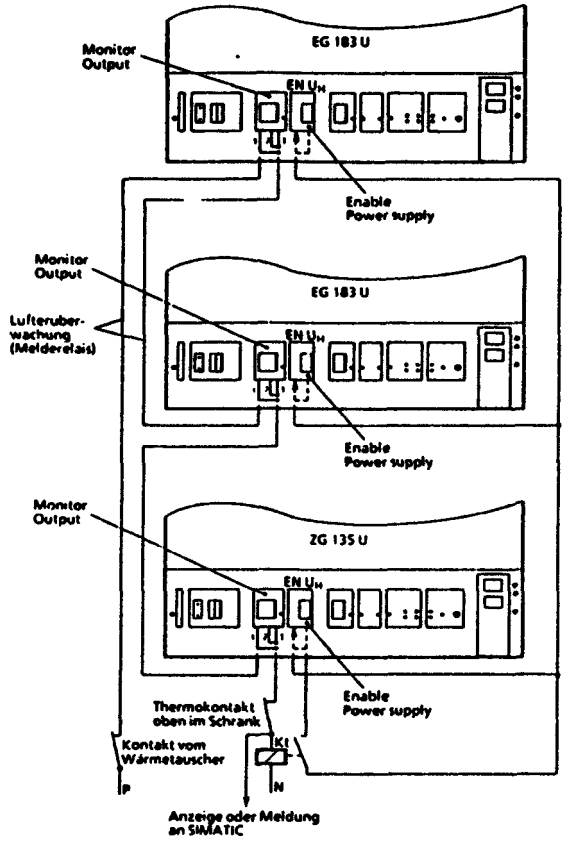


Рис. 4.6. Один из вариантов подключения контроля вентиляторов и температурного режима при размещении в одном шкафу центрального устройства 135 U и устройства расширения 183 U

а - устройство расширения EG 183 U
б - контрольный выход
в - разрешение подачи питания
г - контрольный выход
д - устройства расширения EG 183 U
е - контроль вентиляторов (сигнальное реле)
ж - разрешение подачи питания
з - контрольный выход
и - центральное устройство
к - термоконтакт в верхней части шкафа
л - разрешение подачи литания
м - контакт теплообменника
н - индикация или сообщение на SIMATIC

Глава 5

Вентиляция шкафа и рассеиваемая мощность

При комплектации компактных устройств необходимо помнить о том, что рассеиваемая мощность не должны превышать той мощности, отвод которой обеспечивается вентиляторами устройства.

Максимальная величина отводимой встроенными вентиляторами мощность потерь равна 250 Вт при температуре приточного воздуха 55 °C. С уменьшением температуры приточного воздуха эта величина увеличивается на 20 Вт/градус.

5.1 Рассеиваемая мощность модулей

Сводный перечень рассеиваемой мощности модулей, используемых в компактных устройствах приведен ниже.
Максимальные значения можно использовать для оценки полной нагрузки и для оценки допустимых потерь мощности с учетом коэффициента одновременности работы.

Блоки питания	
6ES5 955-3LC14	макс. 75,0 Вт
6ES5 955-3LF12	макс. 140,0 Вт
6ES5 955-3NC12	макс. 55,0 Вт
6ES5 955-3NA12	макс. 50,0 Вт
6ES5 955-5LB11	макс. 150,0 Вт
6ES5 955-5NB11	макс. 25,0 Вт
6ES5 955-3NF11	макс. 25,0 Вт

S5-135U		
CPU 920 (M-процессор)	6ES5 920-3UA11	11,0 Вт
CPU 921 (S-процессор)	6ES5 921-3UA11/12	15,0 Вт
CPU 922 (S-.процессор)	6ES5 922-3UA11	11,0 Вт
Координатор 923A	6ES5 923-3UA11	2,5 Вт
Координатор 923C	6ES5 923-3UC11	5,5 Вт
Процессор 928	6ES5 928-3UA11	15,0 Вт
	6ES5 928-3UA12	17.5 Вт

S5-150 U		
Центральные модули	6ES5 924…927S	63,5 Вт
Интерфейс подключения внешней памяти 341	341	7,5 Вт
Модуль контроля четности	342	7,5 Вт
S5-155U		
CPU 946/947	6ES5 946-3UA11	30,0 Вт
	6ES5 947-3UA11	10,0 Вт
Блоки связи		
устройства расширения	6ES5 300	3,0 Вт
устройства расширения	6ES5 301	4,0 Вт
устройства расширения	6ES5 302	10,0 Вт
устройства расширения	6ES5 304	7,5 Вт
устройства расширения	6ES5 308	2,5 Вт
устройства расширения и центрального устройства	6ES5 310	3,0 Вт
устройства расширения и центрального устройства	6ES5 310 H	2,0 Вт
устройства расширения и центрального устройства	6ES5 311	7,5 Вт
центрального устройства	6ES5 312	1,5 Вт
центрального устройства	6ES5 314	5,0 Вт
центрального устройства	6ES5 318-3U	2,0 Вт
программатора	6ES5 511	8,5 Вт
Интерфейсный блок	6ES5 512 C/H	7,5 Вт
Коммуникационные процессоры		
CP 524	6ES5 524	7,5 Вт
CP 525	6ES5 525	10,0 Вт
CP 526	6ES5 526	11,0 Вт
CP 535	6ES5 535	18,5 Вт
CP 536	6ES5 536	12,0 Вт
Модули памяти		
Модуль памяти	6ES5 340-3	макс. 4,5 Вт
Модуль памяти с субмодулями	6ES5 350-3	макс. 10,0 Вт
Модуль памяти с субмодулями	6ES5 355	макс. 9,0 Вт
Периферийные модули, цифровые		
Модуль входов	6ES5 420-4U..	макс. 7,0 Вт
Модуль входов	6ES5 430-4U..	макс 8,3 Вт
Модуль входов	6ES5 431-4U..	2,2…7,7 Вт (24..60 В)
Модуль входов	6ES5 432-4U..	макс. 7,5 Вт
Модуль входов	6ES5 434-4U..	макс. 5,5 Вт
Модуль входов	6ES5 435-4U..	макс. 3,5 Вт при 24 В макс 18,0 Вт при 60 В
Модуль входов	6ES5 436-4UA	макс. 3,5 Вт при 115 В макс. 17,0 Вт при 240 В
Модуль входов	6ES5 436-4UB	макс. 2,0 Вт при 115В макс. 8,5 Вт при 240 В

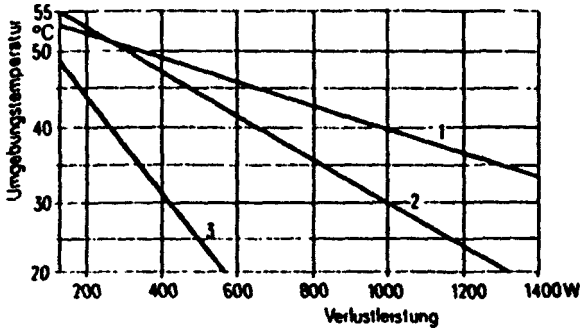
Модуль выходов	6ES5 441-4U..	макс.17,0 вт
Модуль выходов	6ES5 451-4U..	макс.17,0 вт
Модуль выходов	6ES5 453-4U..	макс.49,0 вт
Модуль выходов	6ES5 454-4U..	макс.17,5 вт
Модуль выходов	6ES5 455-4U..	макс.39,0 вт
Модуль выходов	6ES5 456-4UA..	макс.39,0 вт
Модуль выходов	6ES5 456-4UB..	макс.18,0 вт
Модуль выходов	6ES5 457-4U..	макс.13,0 вт
Модуль выходов	6ES5 458-4U..	макс.5,2 вт
Периферийные модули		
Модуль входов	6ES5 460-4U..	макс.3,5 вт
Модуль входов	6ES5 463-4U	макс.5,0 вт
Модуль входов	6ES5 465-4U	макс.1,5 вт
Модуль выходов	6ES5 470-4U	макс.9,0 вт

5.2 Рассеиваемая в шкафу мощность и охлаждение шкафа

Величина отводимой из шкафа мощности зависит от конструктивного исполнения шкафа, температуры окружающей среды и от размещения оборудования в шкафу.

Ориентировочные значения допустимой температуры окружающей среды для шкафа размерами 600мм x 600мм x 2200мм выбираются по графику (смотри Рис.5.1) в зависимости от рассеиваемой мощности установленного оборудования.

Обратите внимание, что эти значения действительны только в случае компоновки, приведенной на Рис. 6.1, 6.2 и 6.3.



а - температура окружающей среды
б - рассеиваемая мощность

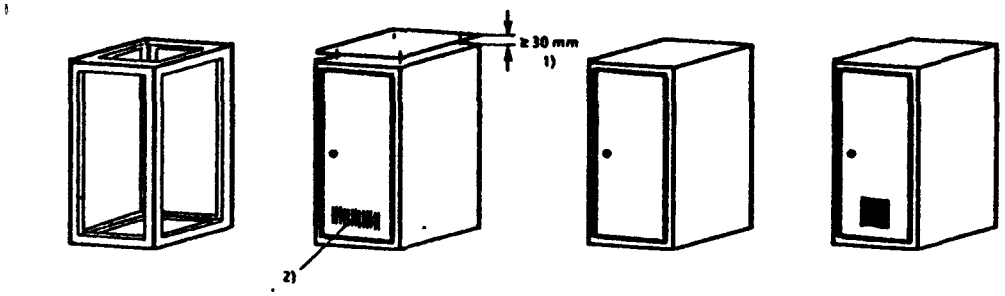
1 Закрытый шкаф с теплообменником
2 Шкаф с вентиляционными щелями
3 Закрытый шкаф с естественной конвекцией и принудительной циркуляцией, обеспечиваемой вентиляторами устройств

Рис. 5.1 Максимальная температура окружающей среды шкафа в зависимости от рассеиваемой мощности установленного оборудования

Пример

1	Центральное устройство	200 Вт
2	Устройства расширения, рассеиваемая мощность составляет по 250Вт на каждом	500 Вт
1	Силовой блок питания, 24В/40А, 6EV1360 (полная нагрузка)	200 Вт
	Суммарная рассеиваемая мощность:	900 Вт

На выбор рода защиты шкафа решающее влияние оказывают условия окружающей среды в месте установки.



Каркас/настенный вариант	Открытый шкаф с принудительной вентиляцией	Закрытый шкаф с естественной конвекцией и принудительной вентиляцией	Закрытый шкаф с теплообменником
< IP1X	> IP2X	>IP5X	>IP5X
только в чистых и сухих помещениях		без ограничений по месту установки	
только на электроучастках или на замкнутых электроучастках			

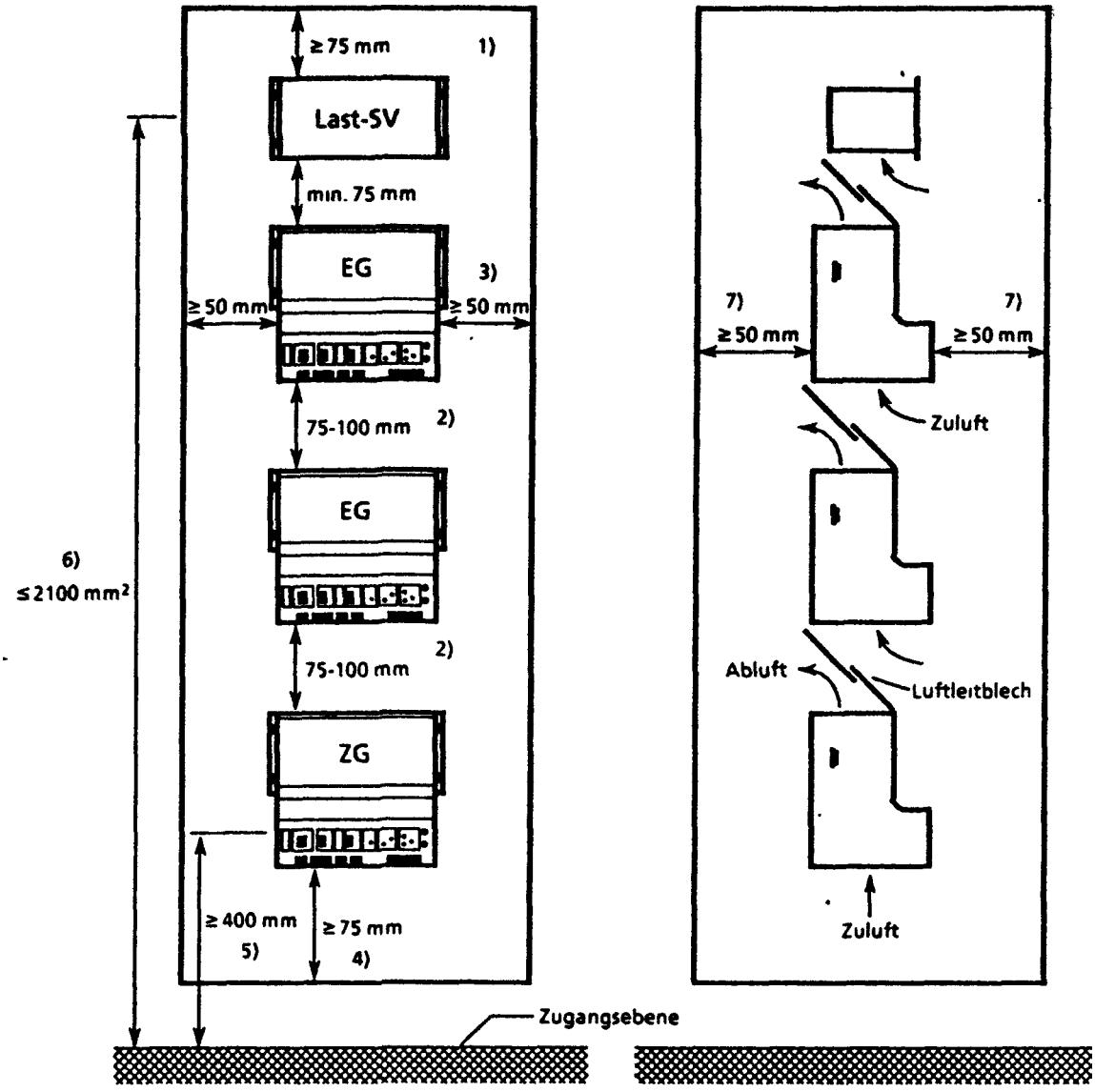
1) Обратите внимание на зазор.
2) Площадь охлаждающих щелей > 600 см^2
3) Максимальная температура окружающей среды

Рис. 5.2 Конструкция шкафов

Глава 6
Компоновка оборудования в шкафу

6.1 Монтажные размеры

Размещение оборудования в шкафу выполняется с обеспечением необходимого расстояния между устройствами и с учетом ограничения по высоте расположения органов управления. Вследствие этого в шкафу можно разместить друг над другом не более 3 устройств серии U.



а Блок питания	е Приточный воздух
б Устройство расширения	ж Воздухонаправляющая пластина
в Устройство расширения	з Отводимый воздух
г Центральное устройство	и Приточный воздух
д Поверхность пола	

Рис. 6.1 Монтажные размеры

1) Минимум 75мм при закрытой крышке. Это расстояние можно уменьшить в случае использования перфорированной крышки и при снятой вентиляционной крышке.

2) Минимальное расстояние для всасывания и отвода воздуха составляет 75мм, максимальное - 100мм. Ограничение на максимальное расстояние накладывается длиной соединительных кабелей между модулями связи центрального устройства и устройства расширения.

3) При параллельном подключении устройств (с использованием интерфейсного модуля AS312) возможно максимальное расстояние 400мм.

4) Минимум 75мм до предметов, препятствующих всасыванию воздуха (такими препятствиями являются устройства или оборудование с большой площадью поверхности).

5) Минимальная высота органов управления над уровнем пола равна 400мм, минимальная высота расположения элементов подключения над уровнем пола -/ 200мм.

6) Максимальная высота расположения органов управления:
2100Мм согласно норме VDE0106, часть 100 и 2000мм
согласно норме VDE0113.

7) Свободное пространство для циркуляции воздуха (необходимые условия обеспечиваются при глубине шкафа 400мм).

6.2 Примеры компоновки в шкафу

В качестве примера был рассмотрен монтаж в шкафу 8MF (2200мм x 600мм x 600мм) с использованием профильных направляющих С. Такой вариант монтажа, наряду с невысокой стоимостью, отличается следующими преимуществами:

- Монтаж во всех шкафах может осуществляться без использования дополнительных деталей.
- Компоновка не зависит от ширины шкафа (диапазон ширины шкафа от 350 до 1200мм).

- - Устройства можно размещать асимметрично. За счет асимметричности с одной стороны можно обеспечить большее пространство для прокладки сигнального кабеля.
- Установку и снятие устройств (включая блоки питания нагрузки) можно (и при последующей работе) выполнять спереди. При этом, на устройствах достаточно освободить (не вывинчивая до конца) гайку с фиксатором. Установка и снятие устройств выполняется с использованием гаек (монтаж силами одного человека).
- За счет прокладки аналоговых, цифровых и сетевых кабелей в раздельных каналах обеспечивается помехозащищенность - уменьшение помех, обусловленных взаимным влиянием кабелей.
- - Для подавления помех в сетевых кабелях на вводе в шкаф рекомендуется установить сетевой фильтр.

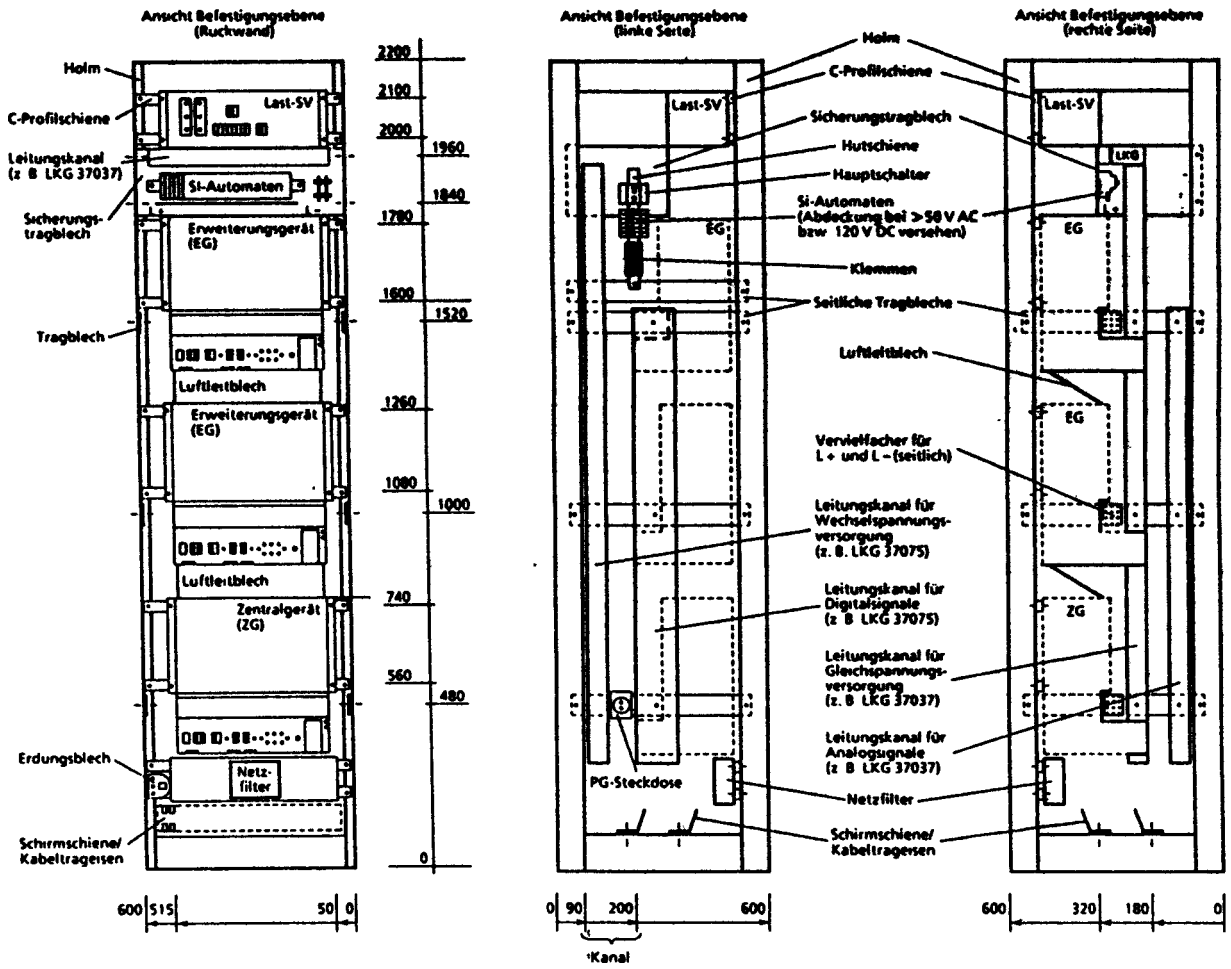


Рис. 6.2 Пример компоновки и
Рис. 6.2/1 Уровни крепления левая/правая сторона

а - Монтажная схема компоновки (вид сзади)
б - Балка
в - Профильная направляющая С
г - Канал для прокладки кабеля (например LKG37037)
д - Элемент несущей конструкции для установки предохранителей
е - Поддерживающий элемент
ж - Пластина заземления
э - Экранированная шина/опорный элемент крепления кабеля
и - Сетевой фильтр
к- Центральное устройство (ZG)
л - Воздухонаправляющая пэтастина
м - Устройство расширения (EG)
н - Воздухонаправляющая пластина
о - Устройство расширения (EG)
п - Установочные автоматы
р - Блок питания нагрузки
с - Монтажная схема компоновки (вид слева)

т - Балка
у - Профильная направляющая С
ф - Элемент несущей конструкции для установки предохранителей
х - Верхняя шина
ц - Главный выключатель
ч - Установочные автоматы (при напряжении, превышающем 50В переменного тока или 120В постоянного тока, необходимо предусмотреть защитную крышку)
ш - Клеммы
щ - Боковые элементы несущей конструкции
э - Воздухонаправляюшая пластина
ю - Расширитель для линий L+ и L- (сбоку)
я - Канал для прокладки кабеля питания (переменное напряжение)
(например LKG37075)
аа - Канал для прокладки кабеля цифровых сигналов
(например LKG37075)
аб - Канал для прокладки кабеля питания (постоянное напряжения)
(например LKG37073)
ав - Канал для прокладки кабеля аналоговых сигналов
(например LKG37073)
аг - Сетевой фильтр
ад - Экранированная шина/опорный элемент крепления кабеля
ае - Канал
аж - Розетка программатора
аз - Устройство расширения
аи - Блок питания нагрузки
ак - Монтажная схема компоновки (вид справа)
ал - Блок питания нагрузки
ам - Канал для прокладки кабеля
ан - Устройство расширения
ао - Устройство расширения
ап - Центральное устройство

Глава 7
Электрический монтаж в шкафу

7.1 Правила подключения

o Подключение питания

Соединительные клеммы компактных устройств рассчитаны максимум на
- одножильный провод с сечением до 4 мм^2 или
- многожильный провод с наконечником, сечением до 2,5 мм^2

o Подключение сигнальных кабелей к периферийным модулям

Модули оснащены 20- или 42-контактными разъемами, размер ножевого контакта – 2,4мм x 0,8мм.

Для подключения сигнальных проводов предусмотрены передние разъемы установочной ширины 20 и 40 мм. Можно выбрать передний разъем с зажимными или винтовыми контактами (см таблицу 7.1).

Для работы с передними разъемами более удобны многожильные провода.

В случае использования винтового подключения отпадает необходимость в наконечниках на проводниках, т.к. и без них обеспечивается надежное крепление.

Ширина рабочего конца отвертки: для 4UB31 2.8 мм
для остальных 3.5 мм

Anschluß-art	Stecker-Typ 6ES5497-	Max. Pol-zahl	Querschnitt der Signal- oder Ver-sorgungsleitung	Stecker für Nennspan-nung	Gabel-breite mm 2)	Front-stecker-breite	Baugruppen-Typ 6ES5... Betriebsart mit Lüfter	ohne Lüfter
Crimp-kontakt 1)	4UA12	42	0,5...1,5 mm²	5...60 V DC	12	20 mm	420, 430, 431, 432, 434, 441, 451, 458	-
	4UA22	42	0,5...1,5 mm²	5...60 V DC	12	40 mm	453, 454	420, 430, 431, 432, 434, 441, 451, 453, (454), 458
	4UA42	20	0,5...1,5 mm²	24 ..240 V AC	14	40 mm	435, 436, 455, 456	
Schraub-kontakt	4UB12	42	0,5...2 x 2,5 mm²	5 ..60 V DC	12	40 mm	420, 430, 431, 432, 434, 441, 451, 453, (454), 458	
	4UB22	25	0,5...2 x 2,5 mm²	5 ..60 V DC	12	40 mm	454	
	4UB31	42	0,5...1,5 mm²	5.. 60 V DC	12	20 mm	wie 4UA12	-
	4UB42	20	0,5...2 x 2,5 mm²	24...240 V AC	14	40 mm	435, 436, 455, 456	

Таблица 7.1 Подключение передних разъемов

1) Описание контактов и инструментов для обжимки приведены в каталоге ET1.

2) для предотвращения неправильного подключения модулей передние разъемы подвода переменного напряжения (AC) оснащены вилкой шириной 14мм, а передние разъемы подвода постоянного напряжения (DC) - 12мм.

7.2 Прокладка кабелей

По возмущающим воздействиям кабели образуют следующие группы:
- Кабели питания для подвода напряжения от 110 до 38ОВ переменного тока к программируемым контроллерам, устройствам расширения и блокам питания нагрузки.
- Кабели питания для подвода напряжения 24В постоянного тока.
- Кабели цифровых сигналов для переменного напряжения.
- Кабели цифровых сигналов для постоянного напряжения.
- Кабели аналоговых сигналов.

При прокладке сигнальных кабелей и кабелей питания внутри шкафа необходимо учитывать требования нормы VDE 0660, часть 500, а при прокладке сигнальных кабелей и кабелей питания за пределами шкафа - требования нормы VDE 0100, части 430, 510 и 523.

Для аналоговых сигналов должны, в принципе, как в шкафу, так и за его пределами, должны использоваться экранированные кабели (см. Раздел 7.7*.).*

o Прокладка кабелей в шкафу

Все указанные группы кабелей должны прокладываться в шкафу отдельно друг от друга. Прокладка считается раздельной, если:
- разделены каналы прокладки (минимальное расстояние между каналами 2 см)
- разделены жгуты
- используется отдельный кабель
- кабель экранирован

Кабели питания для подвода напряжения питания 220В переменного тока могут прокладываться в одном канале с сигнальными кабелями только в том случае, если эти кабели экранированы.

Отдельные кабели цифровых сигналов для постоянного напряжения и переменного напряжения не должны прокладываться в шкафу с использованием одного и того же канала.

o Прокладка кабелей за пределами шкафа

Проводка цифровых сигналов для переменного напряжения, постоянного напряжения и проводка аналоговых сигналов должна выполняться с использованием разных кабелей.

Сигнальные кабели могут прокладываться в одном канале вместе с силовыми кабелями.

Сигнальный кабель и силовой кабель напряжения до 380В переменного тока можно прокладывать рядом друг с другом. Однако, для уменьшения возмущающих воздействий, эти кабели рекомендуется прокладывать на расстоянии около 10 см друг от друга.

Между сигнальными кабелями и силовыми кабелями напряжением более 500В переменного тока должно быть расстояние не менее 10 см, а между сигнальными кабелями и силовыми кабелями напряжением более 1кВ переменного тока должно быть расстояние более 30см.

7.3 Заземление на корпус

Под заземлением на корпус следует понимать проводящее соединение неактивных металлических частей (VDE06O). В соответствии с этим при креплении рам компактных устройств к несущим элементам-балкам, используются обеспечивающие хорошую проводимость зубчатые шайбы. Балки, в свою очередь, также соединяются с корпусом шкафа с обеспечением хорошей проводимости. При настенном монтаже корпуса устройств соединяются с потенциалом земли (например, шиной защитного провода) проводом с площадью поперечного сечения не менее 10мм^3.

7.4 Защита от тока, опасного для человеческого организма, зажита при косвенном контакте

Посредством защитного провода с площадью сечения не менее 10мм^2 необходимо выполнить подсоединение шкафа к защитному проводу системы распределения энергии, к которой подключено питание 220В переменного тока устройств, размещенных в шкафу.

В случае, когда питание осуществляется от источника напряжения 24В постоянного тока, устройство необходимо соединить с централизованной точкой заземления или с защитным проводом системы распределения энергии, от которой обеспечивается питание розетки программатора (площадь поперечного сечения используемого провода - не менее 10мм^2). Защитные провода систем распределения питания центральных устройств, устройств расширения и блоков питания должны подключаться к клеммам защитного провода этих устройств.

Несколько стоящих рядом шкафов можно соединить друг с другом привинчиванием, при этом должна быть обеспечена хорошая проводимость. Возможен и другой вариант — к каждому из шкафов подключается защитный провод с площадью поперечного сечения не менее Юмм'^{2}. В результате выполнения описанных мероприятий шкаф и размещенное в нем оборудование будут подключены к системе защиты от тока, опасного для человеческого организма (защита при косвенном контакте).

7.5 Выравнивание потенциалов

Необходимо различать следующие варианты децентрализованной схемы размещения:

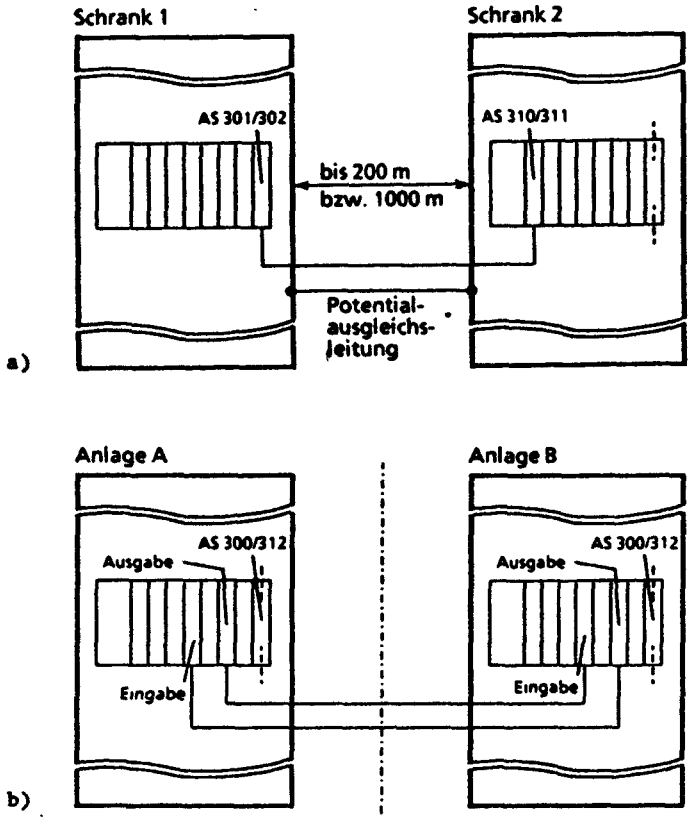
- Разнесенное размещение центрального устройства и устройств расширения расстояние между ними - до 200м) с соединением через модули связи 301/310 (см. Рис.7.1а). Модули связи не развязаны по потенциалу. Допустима разность потенциалов до 7В.

- Разнесенное размещение центрального устройства и устройств расширения (расстояние между ними до 1000м) с последовательным соединением через модули связи 302/311 (см. Рнс.7.1а). Модули связи развязаны по потенциалу.

- Обмен сигналами между разнесенными установками через развязанные по потенциалу модули входов и выходов (см. Рис.7.16).

Прокладка провода выравнивания потенциалов обязательна в первом случае (соединительный провод с площадью поперечного сечения не менее Юмм^2 прокладывается между несущими балками шкафов и рамами корпусов) и желательна при использовании всех остальных вариантов децентрализованной схемы размещения.

Если при централизованной схеме размещения в одном из шкафов используются связанные по потенциалу модули связи 301/310, то через связанные по потенциалу периферийные модули может выполняться дополнительный обмен сигналами. В этом случае необходима прокладка провода выравнивания потенциалов.



а - Шкаф 1
б - Шкаф 2
в - до 200м или до 1000м
г- - Провод выравнивания потенциалов
д - Установка 1
е- Установка
ж- Вход
з- Выход
и- Вход
к- Выход

Рис.7.1 Обмен сигналами при использовании децентрализованной схемы размещения.

7.6 Меры по защите от напряжения помех

Для предотвращения возникновения пиков напряжения помех в кабелях питания и сигнальных кабелях шкафа необходимо соблюдать приводимые далее указания по подавлению помех на установках.

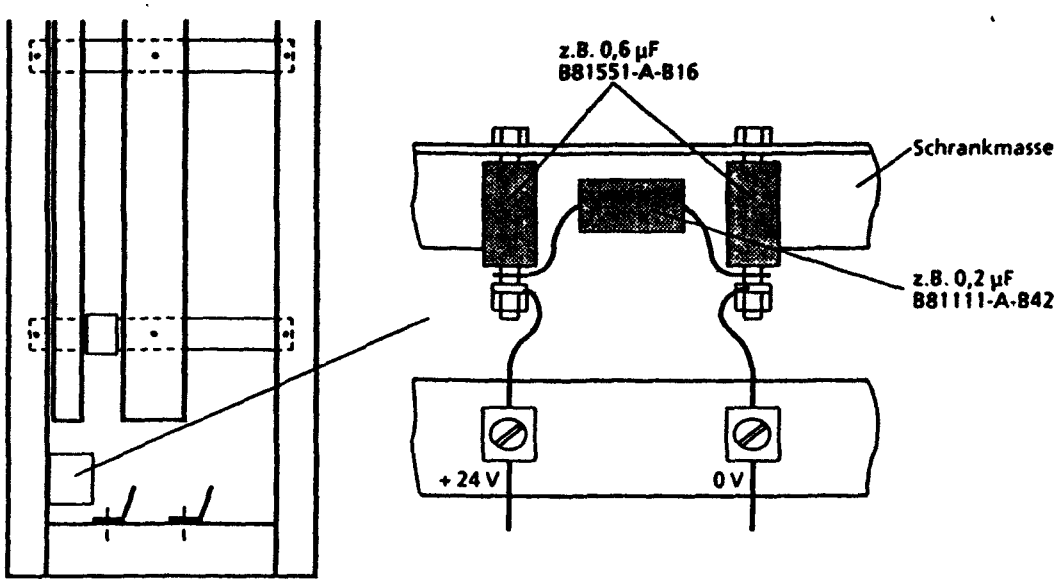
о Подавление помех в сетевых кабелях

При питании шкафа от сети 230В на сетевом кабеле необходимо установить сетевой фильтр (например B84299-K64, 250В переменного тока / 10А (см. Рис.6.2). Расстояние между вводом в шкаф и сетевым фильтром должно быть по возможности минимальным. Клемма подключения к корпусу сетевого фильтра подсоединяется по кратчайшему пути с централизованной точкой заземления в шкафу.

o шунтирующие конденсаторы при питании от источника постоянного напряжения

В случае, когда шкаф подключен к централизованному источнику питания с напряжением питания 24В, напряжения помехи могут попадать в шкаф через кабели литания.

В этой ситуации рекомендуется на вводе в шкаф питания 24В установить конденсаторы подавления помех, которые монтируются на корпусе шкафа (соединение с корпусом) или на экранной шине.



а – например, 0,6 мкф B81551-A-B16
б – корпус или "масса" шкафа
в – например, 0,2 мкф B81111-A-B42
г- + 24
д- 0 В

Рис. 7.2 Размещение конденсаторов подавления помех

o Монтаж индуктивностей

Для индуктивностей, расположенных в том же шкафу, и которые не регулируются непосредственно выходами SIMATIC (например, контактор, катушки реле) необходимо установить гасящие элементы (например, звенья RC).

o Отделение индуктивностей

Рекомендуется отделить разделительной перегородкой из металла ту часть шкафа, которая содержит большие индуктивности, например, трансформаторы или контакторы.

o Защита от электростатического разряда

Для защиты от электростатического разряда устройств и модулей

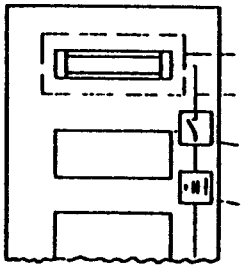
необходимо использовать закрытые со всех сторон металлические корпуса или шкафы, соединенные с обеспечением хорошей проводимости с точкой заземления на месте установки.

При использовании клеммных коробок для небольших устройств (таких, как, например, программируемый контроллер AG /ETI00U) желательно работать с литыми корпусами или с корпусами из листовой стали, но не с пластмассовыми корпусами.

Двери или крышки корпусов должны соединяться посредством лент с корпусом или посредством контактных пружин с заземленным корпусом устройства.

В случае, когда при проведении работ на установке шкаф должен находиться в открытом состоянии, необходимо соблюдать инструкции по обеспечению защиты элементов и модулей, которые могут быть повреждены электростатическим зарядом (EGB) (см. Техническое извещение EMA/TM6008, 04/1985 или Указания по работе с элементами и модулями, которые могут быть повреждены электростатическим зарядом, номер для заказа C79000-B0600-C070-01).

Для освещения шкафа не должны использоваться люминесцентные лампы, т.к. они оказывают возмущающее воздействие на работу устройств. Если же отказ от применения люминесцентных ламп невозможен, то для подавления возможных помех подключение должно выполняться в соответствии со схемой, приведенной на рисунке 7.3. Лучше всего для освещения подходят лампы марки LINESTRA.



В исключительных случаях (например, при небольшом расстоянии от системы управления")

- экранирующая сетка на лампе
- полностью экранированный кабель
- переключатель, заключенный в металлический корпус
- сетевой фильтр или экранированный сетевой ввод

Рис.7.3 Меры по подавлению помех, создаваемых в шкафу люминесцентными лампами

7.7 Экранирование, ограничения по длине кабелей

Проложенные кабели аналоговых сигналов должны быть экранированы по всей длине. В отличие от кабелей аналоговых сигналов, определенная часть кабеля цифрового сигнала может прокладываться без экранирования. Данные о максимально допустимых длинах кабелей приведены в таблице 7.2 и 7.3.

o Цифровые модули

Модули	Длина кабеля 9 (м)	
	без экрана	с экраном
	400	1000
Входы, 24В постоянного тока	600	1000
Входы 220В переменного тока	600	1000
Входы 431, от 24 до 60В пост.	400	1000
Входы 434, от 5 до 15В пост.	200 (600) [1)]	600 (1000) [1)]

1) данные .по длине, которые указаны в скобках, действительны при подключении датчиков NAMUR.

Таблица 7.2 Максимально допустимые длины кабелей цифровых сигналов при прокладке в общем кабеле

o Аналоговые модули

Модуль 6ES5....	Длина проводов 1)	Разница потенциалов [2)]
460	50м (Ue = 50мВ) 200м (Ue = 500мВ)	60В/75В пост. тока
463	200м	25В переменного тока/ 60В постоянного тока
465	50м (Ue = 50мВ) 200м (Ue = 500мВ)	+-1,0 В
470	200м	60В переменного тока/ 75В постоянного тока (только относительно М)

1) Значения длины проводов входных и выходных кабелей аналоговых модулей при прокладке в общем экранированном кабеле.

2) Допустимая разница потенциалов входных и выходных каналов относительно точки заземления (М) и между каналами.

Таблица 7.3. Длина проводов и разница потенциалов

Указания по подключению экранов:

Экраны кабелей должны монтироваться в шкафу на экранной шине в непосредственной близости от кабельного ввода. При работе с экранами-оплетками необходимо обеспечить крепление к экранной. шине максимально возможной поверхности экрана кабеля (например, крепление шланговым металлическим фиксатором, охватывающим экран, или крепление PUK-скобами для фиксации кабеля).

Для кабелей с экраном из Фольги необходимо по кратчайшему пути (около 3 см) подключить прокладываемый вместе с этим кабелем экранный провод к экранной шине. Для экранной шины должно обеспечиваться хорошо проводящее соединение с несущей балкой, корпусом шкафа и с централизованной точкой заземления в шкафу.

Экранные шины для аналоговых сигнальных кабелей можно монтировать изолированно и подключать их к опорному потенциалу или к земле в централизованной точке.

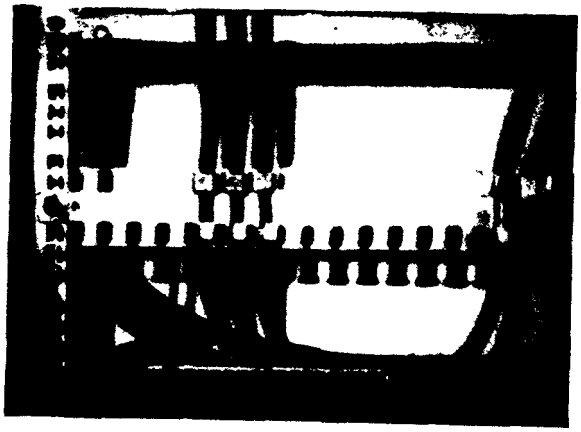


Рис. 7.4 Крепление экранов кабелей к экранной шине

o Одностороннее заземление экранов кабелей

Для экранов аналоговых сигнальных кабелей, по которым проходят небольшие сигналы (мВ или мкА), в шкафу выполняется одностороннее подключение к экранной шине.

Может оказаться целесообразным продолжение экрана от экранной шины до модуля. В этом случае не осуществляется подключение экрана к модулю.

У цифровых сигнальных кабелей экран кончается на экранной шине.

Одностороннее заземление экрана кабеля целесообразно в том случае, если к другому концу кабеля невозможно проложить низкоомный провод выравнивания потенциалов или в случае, когда должно учитываться только низкочастотное или статическое возмущение.

о Двухстороннее заземление экранов кабелей

Для двухстороннего заземления экрана цифрового сигнального кабеля, обеспечивающего наиболее благоприятные условия для отвода высокочастотных помех, необходимо наличие низкоомного провода выравнивания потенциалов.

Сопротивление должно составлять около 10% от сопротивления оплетки экрана.

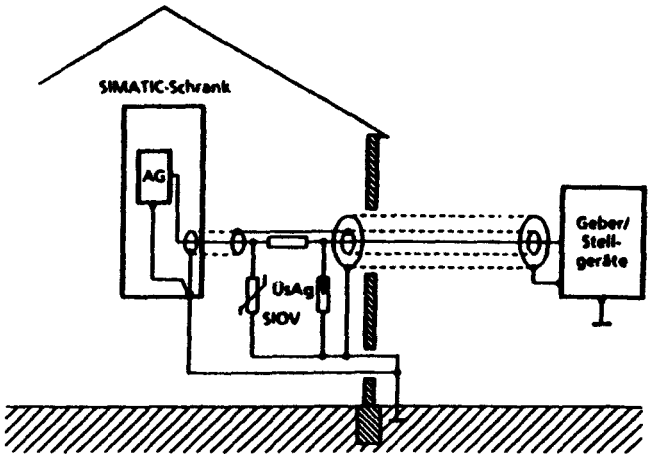
У кабелей шин, поставляемых заводом-изготовителем, выполнено двухстороннее заземление экрана. Посредством этих кабелей шин полностью или частично реализуется внутренняя вина устройства (например, в интерфейсных модулях устройств .расширения). Отключение заземлений экранов таких кабелей недопустимо.

Средства подключения шин, которые предусмотрены на модулях, должны использоваться лишь в исключительных случаях (например, экономия на экранной шине необходима при наличии одного единственного экранированного кабеля).

7.8 Грозозащита

При прокладке кабелей и проводов устройств S5 за пределами зданий должны применяться только полностью экранированные кабели. Экран должен быть стойким к воздействию электрического тока, обе стороны экрана должны соединяться с землей, для аналоговых сигналов следует применять в этом случае кабели с двойной экранировкой. Внутренний экран кабеля аналоговых сигналов заземляется с одной стороны (см.Раздел 7.7*).*

Кроме того, к сигнальным кабелям должны подключаться элементы защиты от перенапряжения (варисторы SIOV) и грозовые разрядники, наполненные инертным газом (USAG). Элементы защиты от перенапряжения должны размещаться, по возможности, у кабельного ввода в здание или, в крайнем случае, у шкафа.



а - Шкаф SIMATIC
б - Программируемый контроллер
в - Грозовой разрядник
г - Варистор
д - Датчик/исполнительный блок

Рис. 7.5 Размещение элементов

7.9 Подключение программаторов к сети питания

Для подключения программаторов в каждой группе шкафов предусмотрена штепсельная розетка с защитным контактом. Подвод питания на штепсельные розетки с распределителя, к которому подключен защитный провод шкафа.

Глава 8
Защита и контроль

При проектировании программируемых контроллеров, как и в случае контакторных систем управления, необходимо соблюдать действующие нормы VDE (например, VDE 0113, VDE 0100). Особое внимание при этом необходимо уделять реализации мероприятий, .направленных на предотвращение возникновения опасных ситуаций:

o Необходимо предотвращать ситуации, возникновение которых опасно для людей или может привести к повреждению оборудования или материалов.

o Нельзя допускать автоматический запуск станков и машин после восстановления исчезнувшего перед этим сетевого напряжения питания или после деблокировки устройства аварийного выключения.

o При сбоях программируемого контроллера в любом случае должно сохраняться действие команд устройства аварийного выключения (NOT-AUS) и клавиш границ безопасного расстояния. Должно обеспечиваться непосредственное срабатывание этих защитных приспособлений на исполнительных блоках силовой части системы.

o Использование устройства аварийного отключения (NOT-AUS) должно приводить станок к обеспечению безопасного состояния для людей и установок:

- Должны отключаться исполнительные блоки и приводы, которые могут привести к возникновению опасного состояния (например, приводы главного шпинделя станка),
- Устройство аварийного выключения (NOT—AUS) не должно отключать те исполнительные блоки и приводы, отключение которых может создать опасную ситуацию для людей и установки (например, зажимные приспособления.).

o Применение устройства аварийного выключения должно дополнительно регистрироваться программируемым контроллером и обрабатываться прикладной программой.