

Приложения

В данной части настоящей книги представлены: инструкции по преобразованию программ, предназначенных для STEP 5, в программы для STEP 7, общий обзор содержимого библиотек блоков STEP 7 и общий обзор всех STL- и SCL-инструкций и функций.

Программное обеспечение **S5/S7 Converter (S5/S7-конвертер)** является опционным (то есть, поставляемым по отдельному заказу) программным продуктом. С помощью S5/S7-конвертера пользователь имеет возможность преобразовать имеющиеся программы, предназначенные для STEP 5, в STL-программы для STEP 7 в виде исходных файлов.

Наряду с другими программными продуктами в поставку ПО STEP 7 входят библиотеки блоков **Block Libraries (Библиотеки блоков)**, содержащие загружаемые функции и функциональные блоки, а также заголовки и описания интерфейсов системных функций SFC и системных функциональных блоков SFB.

Загружаемые функции FC и функциональные блоки SFB являются скомпилированными блоками, которые Вы можете копировать в Вашу пользовательскую программу (или, более точно, в автономный [offline] раздел *Blocks [Блоки]*), чтобы в последствии их вызывать. Эти блоки занимают в памяти такое же пространство как и "обычные" пользовательские блоки и, также как пользовательские блоки, загружаются в CPU.

Пользователь может переименовывать загружаемые функции и функциональные блоки, например, в случае, если их номера уже были назначены Вашим собственным блокам. Тем не менее, в случае возникновения проблем, Вы можете получить корректную контекстную помощь (Help). Если Вы выделили интересующий Вас блок, то диалоговое окно справочной системы может быть вызвано с помощью функциональной клавиши F1. Справочная система ориентирована на помощь, которая может касаться вопросов, связанных со свойствами блоков FAMILY (Семейство) и NAME (Имя).

Системные функции SFC и системные функциональные блоки SFB включены в операционную систему CPU. Для того чтобы можно было вызывать эти блоки в автономном (offline) режиме, стандартная библиотека содержит заголовки и описания интерфейсов этих блоков (сама программа, конечно, при этом располагается в CPU). Пользователь может скопировать описания интерфейсов как скомпилированные блоки в автономный (offline) раздел *Blocks (Блоки)* и в дальнейшем вызывать соответствующие системные блоки.

Редактор программ узнает из описания интерфейса, сколько параметров имеет системный блок, к какому типу данных они относятся и какие имена имеют эти параметры.

При инкрементном программировании пользователь может перенести библиотечные блоки из каталога элементов программы в окно программы и, таким образом, обеспечить их вызов. Редактор программы затем автоматически копирует эти блоки в Вашу программу.

Если при программировании путем создания исходной программы Вы вызываете библиотечные блоки с символьными именами из библиотечной таблицы символов, то на этапе компилирования стандартные блоки также будут автоматически скопированы в Вашу программу.

Книга заканчивается обзором STL-операторов и общим обзором SCL-инструкций.

32 S5/S7-конвертер (S5/S7 Converter)

Инструкции для преобразования программ, предназначенных для STEP 5, в программы для STEP 7.

33 Библиотеки блоков (Block Libraries)

Организационные блоки (OB), системные функции (SFC), системные функциональные блоки (SFB), функциональные IEC-блоки (загружаемые IEC-функции), блоки для S5-S7-преобразования (загружаемые функции преобразования), блоки ПИД-управления (функции для автоматического управления), коммуникационные блоки (DP-функции).

34 Обзор STL-операторов

Все STL-операторы.

35 Общий обзор SCL-инструкций

Все SCL-инструкции и SCL-функции.

32 S5/S7-конвертер

С помощью S5/S7-конвертера пользователь имеет возможность преобразовать имеющиеся программы, предназначенные для STEP 5, в STL-программы (для STEP 7) в виде исходных файлов. S5/S7-конвертер обращает непосредственно конвертируемые S5-инструкции в соответствующие инструкции STEP 7. Инструкции STEP 5, которые не могут быть непосредственно преобразованы в инструкции STEP 7, оказываются закомментированными. S5/S7-конвертер учитывает (сохраняет) все комментарии в программе. По выбору (опционно) список назначений может также быть преобразован в импортируемую таблицу символов.

Чтобы преобразовать последовательное управление посредством GRAPH 5 (sequential control) в программу STEP 7, Вы должны вновь создать программу с использованием S7-GRAPH.

S5/S7-конвертер входит в поставку наряду с другими программными продуктами ПО STEP 7. Пользователю не требуется проводить авторизацию для того, чтобы использовать это программное средство.

В электронном каталоге CA01 (CD) Вы найдете поддержку для аппаратного преобразования конфигурации SIMATIC S5 в конфигурацию SIMATIC S7 в пункте меню: *Selection Aids -> SIMATIC (Выбор поддержки -> SIMATIC)*. После выбора конфигурации S5 с помощью опций меню: *Edit -> Generate Signal List (Правка -> Создать список сигналов)* и *Edit -> Generate Configuration (Правка -> Создать конфигурацию)* Вы можете создать S7-станцию из спецификаций для конфигурации S5.

32.1 Общая информация

Для преобразования программы STEP 5 Вам потребуются файл программы с именем *nameST.S5D*, список перекрестных ссылок с именем *nameXR.INI* и, если есть в наличии, список назначений с именем *nameZ0.SEQ*. Кроме того, Вы можете создать макро-файл. Этот файл содержит последовательность инструкций, которые конвертор может использовать вместо определенных инструкций STEP 5. Из этих файлов конвертор создает исходный файл программы для STEP 7, и, если требуется, таблицу символов. Все созданные файлы сохраняются в том же разделе, что и файлы STEP 5.

Конвертор пересылает организационные блоки с программой пользователя в соответствующие организационные блоки STEP 7, а все остальные кодовые блоки - в функции FC.

Номера блоков FC начинаются с нуля и назначаются в возрастающей последовательности; Вы можете изменять присвоенные номера блоков с помощью соответствующего диалогового окна.

На рис. 32.1 показано соответствие исходных файлов и файлов, полученных в процессе преобразования (конвертации) программы.

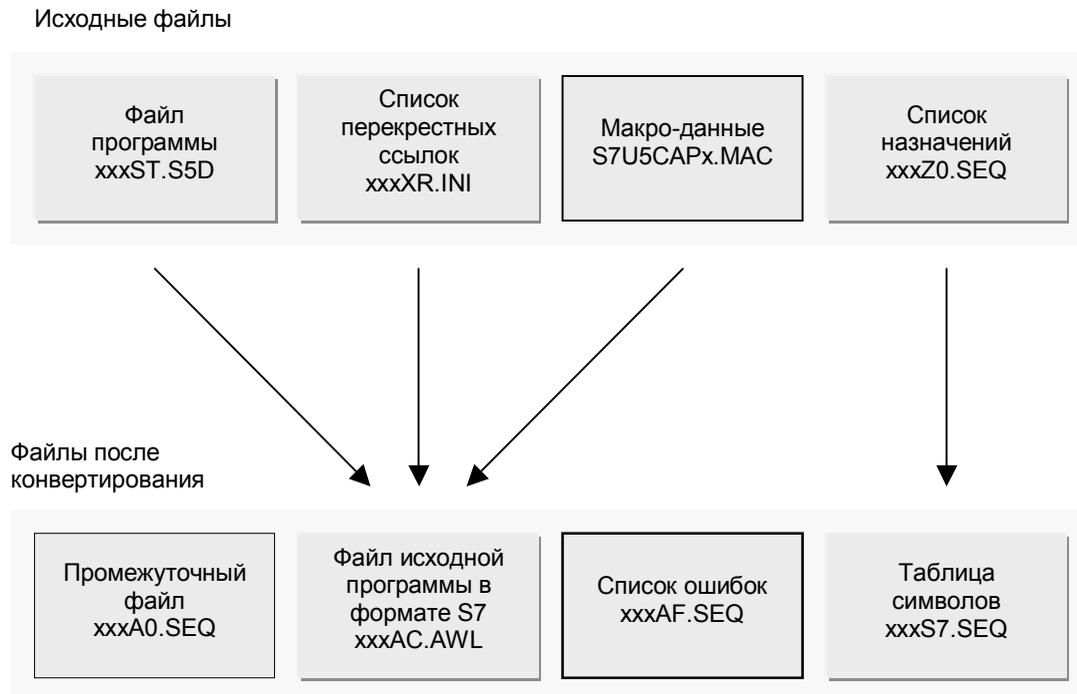


Рис. 32.1 Исходные файлы и файлы после конвертирования программы

Конвертер различает стандартные блоки из следующих прикладных пакетов программ Сименс (Siemens):

- арифметики с плавающей точкой;
- функций для обработки сигналов;
- базовых аналоговых функций;
- математических функций.

Наряду с другими программными продуктами в поставку ПО STEP 7 входит библиотека блоков замены при конвертировании программы *S5/S7 Converting Blocks (Блоки замены при S5/S7 конвертировании программы)*. Эта библиотека содержит блоки, которые заменяют стандартные блоки при S5/S7 конвертировании программы из перечисленных выше прикладных пакетов программ Сименс (Siemens). Вы также можете найти в этой библиотеке стандартные блоки ("integral functions" - "встроенные функции"), которые заменяют некоторые из функциональных блоков, встроенных в CPU S5-115U.

Если программа STEP 5 содержит блоки из указанных программных пакетов, то конвертер преобразует вызовы и сигналы, которые обрабатываются этими блоками в программе.

Вы можете скопировать соответствующие блоки из библиотеки в свою пользовательскую программу перед тем, как Вы начнете процесс компиляции конвертированной программы.

При конвертировании программы STEP 5 Вы можете следовать пунктам процедуры, представленной ниже:

- проверка выполнимости программы в среде STEP 7;
- если необходимо, подготовка STEP 5 - программы (например, удаление неконвертируемых разделов программы, которые должны заменяться параметризацией CPU);
- если необходимо, создание макросов (например, для управляемой замены инструкций STEP 5 на инструкции STEP 7 в процессе конвертации программы);
- процесс конвертации программы (создание исходной программы для STEP 7);
- создание проекта STEP 7 с последующим копированием в этот проект исходной программы и таблицы символов, а также, если это необходимо, копирование в этот проект используемых стандартных функциональных блоков;
- если это необходимо, после конвертирования внесение исправлений или добавлений в исходную программу STEP 7;
- компиляция исходной программы STEP 7.

Указанная последовательность шагов при конвертировании программы не является жестко заданной. Вы можете, например, начать конвертирование STEP 5 программы без ее предварительной подготовки и затем выполнить требуемые корректировки уже в исходной STEP 7 программе.

32.2 Подготовка

32.2.1 Проверка выполнимости программы в системе назначения (PLC)

Если Вам требуется использовать существующую программу STEP 5 в системе SIMATIC S7, Вы должны сначала убедиться, что программа может выполняться в системе назначения (PLC).

Например, необходимо проверить:

- Обладает ли CPU назначения требуемыми свойствами? Существуют ли требуемые характеристики выполнения программы?
- С какими модулями работает программа, предназначенная для STEP 5? К каким модулям обращается программа, предназначенная для STEP 7?
- Обладает ли CPU в системе STEP 7 требуемыми адресами (адресами входов, выходов, блоков)?

Вы можете использовать блок расширения S5 с интерфейсным модулем IM 463-2 или отдельные S5-модули в стойке станции S7-400. Также модули SIMATIC S5 могут быть подключены к системе SIMATIC S7 с использованием распределенной периферии (I/O) посредством шины PROFIBUS-DP.

32.2.2 Проверка параметров выполнения программы

Знакомые Вам по SIMATIC S5 "уровни выполнения программы" ("program execution levels") в основном соответствуют уровням выполнения программы в системе SIMATIC S7, где они называются "приоритетными классами" ("priority classes"). Вы можете заменить установки, которые Вы сделали в блоках данных DB1, или DX0, или, может быть, в системных данных при параметризации S7-CPU (например, параметры перезапуска, обработка таймерных ["watchdog"] прерываний).

Встроенные организационные блоки и встроенные функциональные блоки в S5 соответствуют системным блокам в S7. Если Вы использовали встроенные функции в S5, то в S7 Вам необходимо обеспечить выполнение этих функций с помощью системных блоков или с помощью параметризации CPU.

Блок данных DB1

В S5-115U параметры выполнения программы устанавливаются в блоке данных DB1 или в системных данных RS. В таблице 32.1.1 показано, как эти параметры могут быть обеспечены в SIMATIC S7.

Таблица 32.1.1 Установка параметров в SIMATIC S5 и в SIMATIC S7

| Блок данных DB1 и системные данные System Data (S5-115U) | | | |
|--|-----------|-----|--|
| Функция | 941 - 944 | 945 | В S7 заменяется на: |
| Restart delay (задержка перезапуска) | x | x | Параметр CPU "Restart" ("Перезапуск") |
| Retentive feature (реманентность) | x | x | Параметр CPU "Retentivity" ("Реманентность") |
| Cycle time monitoring (мониторинг времени цикла) | x | x | Параметр CPU "Cycle/Clock memory" ("Цикл/Тактовые меркеры") |
| Time interval for watchdog interrupt (интервал для таймерного прерывания) | x | x | Параметр CPU "Watchdog interrupt" ("Таймерное прерывание") |
| Software protection (защита программы) | x | x | Параметр CPU "Protection" ("Защита") |
| Output disable process images (блокировка выходов образа процесса) | x | x | Управление посредством образа подпроцессов SFC 26 UPDAT_PI, SFC 27 UPDAT_PO |
| Integral clock (встроенные часы) | x | x | Параметр CPU "Diagnostics/Clock" ("Диагностика/Часы") SFC 0 SET_CLK, SFC 1 READ_CLK |
| Delay interrupt OB 6 (прерывание с задержкой обработки) | | | |
| Time duration (продолжительность) | x | x | Параметр CPU "Interrupts" ("Прерывания") |
| Execution priority (приоритет выполнения) | x | - | Параметр CPU "Interrupts" ("Прерывания") |
| Sequential process image transfer (последовательная пересылка образа процесса) | - | x | - отсутствует - |
| Reduced PIQ transfer (сокращенная пересылка образа процесса) | - | x | - отсутствует - |

Системные утилиты

CPU S5-115U поддерживают системные утилиты, которые Вы можете применять с помощью организационного блока OB 250 (CPU 945) или с использованием системных данных RS 125 (CPU 941 ... CPU 944).

В таблице 32.1.2 содержатся указания к применению конвертирования этих системных утилит в SIMATIC S7.

Таблица 32.1.2 Установка параметров в SIMATIC S5 и в SIMATIC S7

| Системные утилиты OB 250 и BS 125 | | | |
|---|--------|--------|--|
| Функция | OB 250 | BS 125 | В S7 заменяется на: |
| Time intervals for watchdog interrupts (интервалы для таймерных прерываний) | x | - | Параметр CPU "Watchdog interrupt" ("Таймерное прерывание") |
| Time duration of delay interrupt (продолжительность интервала для прерывания с задержкой обработки) | x | - | Параметр CPU "Interrupts" ("Прерывания") |
| Reduced PIQ transfer (сокращенная пересылка образа процесса) | x | - | - отсутствует - |
| Read/write DBA/DBL register (чтение/запись в DBA/DBL регистр) | x | - | Чтение, например, с использованием L DBNO, L DBLG; прямая запись недопустима |
| Call DX/FX blocks indirectly (вызов блоков DX/FX с косвенной адресацией) | x | - | Вызов блоков с косвенной адресацией |
| Change block ID (смена идентификатора блока) | x | - | - отсутствует - |
| Update configuration image (обновление отображения конфигурации) | x | x | - отсутствует - |
| Set up block address list (создание списка адресов блоков) | x | x | - отсутствует - |
| Create data block (создание блока данных) | x | x | SFC 22 CREAT_DB |
| I/O accesses without QVZ (обращение к периферии (I/O) без QVZ) | x | x | Обработка синхронных ошибок: SFC 36 MSK_FLT, SFC 37 DMSK_FLT, SFC 38 READ_ERR, |
| Disable/Enable digital outputs (блокировка/разблокирование дискретных выходов) | x | x | Master Control Relay MCR (Главное управляющее реле) |
| Delete block (удаление блока) | x | x | Удаление блока данных: посредством SFC 23 DEL_DB |
| Update process image (обновление образа процесса) | - | x | Обработка с использованием блоков управления отображением подпроцессов: SFC 26 UPDAT_PI, SFC 27 UPDAT_PO |
| Interpret data block DB1 (пояснения к блоку данных DB1) | - | x | - отсутствует - |

Блок данных DX0

Для CPU, относящихся к категории высокопроизводительных CPU, информация в блоке данных DX0 определяет параметры выполнения программы.

В таблице 32.1.3 содержатся указания по переходу к SIMATIC S7.

Таблица 32.1.3 Установка параметров в SIMATIC S5 и в SIMATIC S7

| Блок данных DX0 | | | |
|---|------|------|--|
| Функция | 135U | 155U | В S7 заменяется на: |
| Restart characteristics (параметры перезапуска) | x | x | Параметр CPU "Restart" ("Перезапуск") |
| Number of processed timer cells (число обрабатываемых таймеров) | x | x | - отсутствует - (число фиксировано) |
| Cycle time monitoring (мониторинг времени цикла) | x | x | Параметр CPU "Cycle/Clock memory" ("Цикл/Тактовые меркеры") |
| Multiprocessor restart, interprocessor communication flags (перезапуск в мультипроцессорном режиме, флаги для межпроцессорных коммуникаций) | x | x | - отсутствует - |
| Accuracy of floating point arithmetic (точность выполнения арифметических операций с плавающей запятой) | x | - | - отсутствует - |
| Timed interrupt handling (обработка временных прерываний) | - | x | Параметр CPU "Watchdog interrupt" ("Таймерное прерывание") |
| Process interrupt handling (обработка прерываний процесса) | x | x | Параметр CPU "Interrupts" ("Прерывания") |
| Process interrupt level/ level-triggered interrupt (уровень прерываний процесса/ прерывания запускаемые от уровня) | x | - | Параметризация модуля |
| Addressing error monitoring (мониторинг ошибок адресации) | x | - | ОВ 122 (обработка ошибок доступа к входу/выходу [I/O]) |
| Error handling (system stop) (обработка ошибок (остановка системы)) | x | - | Заменяется ОВ обработки ошибок |

32.2.3 Проверка модулей

И/О модули

Сравните технические описания используемых И/О модулей со спецификациями SM-модулей для системы S7. Имеются ли аналоговые модули с соответствующими областями? Когда Вы организуете прямой доступ к аналоговым модулям, обратите внимание на различия в формате данных для S5 и S7.

Интеллектуальные И/О модули IP

Вы можете также использовать некоторые IP-модули в станции S7-400 вместе с адаптером:

- IP 240 Модуль позиционирования, модуль декодирования позиции, модуль счетчика;

- IP 242B Модуль счетчика;
- IP 244 Модуль контроля температуры;
- IP 246/247 Модули позиционирования;
- WF 721/723 Модули позиционирования;
- WF 705 Модуль декодирования позиции.

Стандартные блоки для этих модулей поставляются вместе с адаптером. Если Вы используете эти модули, Вы должны заменить блоки стандарта S5 на блоки стандарта S7 и привести Вашу программу в соответствие с новыми значениями инициализации для параметров. Для того, чтобы оставить для использования IP-модули, Вы должны применять соответствующие FM-модули.

Коммуникационные процессоры CP

Используемые в S5 коммуникационные процессоры необходимо заменить на CP-модули с соответствующими функциями. CP-модули в S7 обеспечивают доступ с помощью SFB-коммуникаций (коммуникаций посредством системных функциональных блоков) вместо блоков обработки данных стандарта S5. Их функции похожи, но их реализация обеспечивается ресурсами языков программирования STEP 7. Вы должны настроить соответствующую S5-программу с функциями обработки данных для работы с SFB-блоками.

S5-модули в станции S7-400

Вы можете также подключить модули расширения S5 к станции S7-400, используя *интерфейсный модуль IM 463-2*. К каждому из двух интерфейсов может быть подключено до четырех модулей расширения S5. При этом в одной центральной монтажной стойке может быть установлено до четырех интерфейсных модулей IM 463-2. Соединениями в модулях расширения S5 управляет интерфейсный модуль IM 314. При этом допускается использование только дискретных и аналоговых модулей. Прерывания процесса не переносятся (при конвертировании программы). Вы можете задавать области I/O для S5-модулей в S5-интерфейсном модуле IM 314 (как обычно в системе S5). Поддерживается доступ к областям входов/выходов (I/O) P, Q, IM3 и IM4.

Вы можете использовать в S7-400 некоторые модули IP и WF (см. выше) с адаптерами. Вы должны при этом, как обычно, задать S5-адреса модулей.

Параметризация S5-адресов с назначением S7-адресов производится с помощью утилиты конфигурирования оборудования Hardware Configuration. Вы можете найти интерфейсный модуль IM 463-2 и адаптер ("*adapter casing*") в каталоге модулей в соответствии с путем размещения: Simatic 400 -> IM-400 -> S5 Adapter. После размещения модулей в монтажной стойке, Вы должны назначить адреса для этих модулей, как для сигнальных S7-модулей, в области периферийных входов/выходов (I/O), отдельно для входных и выходных адресов.

Примечание:

Необходимо обеспечить, чтобы адресные области для адресов S7 и для адресов S5 не перекрывались.

32.2.4 Проверка адресации

Необходимо проверить число доступных адресов в выбранном CPU назначения. Имеется ли достаточное количество входов, выходов, меркеров, таймеров и счетчиков? Конвертер преобразует меркеры из "расширенной" ("extended") области (область S меркеров) в область меркеров, начиная с адреса M 256.0.

В S7 существует одна единая область периферийных входов/выходов (I/O). Все модули, адресованные в S5 в областях входов/выходов (I/O), P, Q, IM3 и IM4 и в "глобальной области" ("global area"), теперь будут адресованы в S7 в области P периферийных входов/выходов (I/O) (Вы должны внимательно отнестись к этому, если у Вас адресовано большое количество модулей в "расширенных" ["extended"] областях I/O и Вы подключаете эти модули к станции S7-400, например, с помощью IM 463-2). Область "страницы" памяти ("the page memory area") пропускается без замещения.

Конвертер преобразует все блоки с пользовательской программой (исключая организационные блоки) в функции, то есть, общее число всех программных блоков (PB), шаговых блоков без программы секвенсора (SB) и функциональных блоков (FB и FX) не должно превышать максимально допустимого числа функций (FC). Аналогично и общее число блоков данных (DB и DX) не должно превышать числа блоков данных S7. На практике эти ограничения касаются только тех случаев, когда Вы используете в качестве целевой системы (PLC) станцию S7-300.

Области системных данных RI, RJ, RS и RT пропускаются без замены в S7. Любая информация, которая располагалась в этих областях сохраняется в S7 в блоках глобальных данных или в области меркеров. После преобразования Вы можете получать системную информацию из RS-области посредством системных функций; инициализацию функций Вы можете выполнять с помощью данной области посредством системных функций или при параметризации CPU.

Подготовка программы STEP 5

Перед преобразованием Вы можете подготовить программу STEP 5 для ее дальнейшего использования в качестве программы для системы STEP 7 (но Вы не должны так непосредственно ее использовать; Вы должны будете сначала после преобразования выполнить все соответствующие корректировки в исходном файле программы STEP 7). С помощью предварительной подготовки программы STEP 5 Вы можете уменьшить число предупреждений (warnings) и сообщений об ошибках при проведении преобразования. Например, Вы можете подготовить программу STEP 5 перед преобразованием, выполнив следующие операции:

- Удаление блоков данных с параметрами программы DB1 или DX0;
- Удаление всех вызовов встроенных блоков или обращений к области системных данных RS; их функции могут быть обеспечены путем параметризации S7-CPU;
- Приведение адресов областей входов, выходов, периферийных входов/выходов (I/O) в соответствие с "новыми" адресами модулей; здесь необходимо обеспечить, чтобы не нарушался диапазон адресов STEP 5, иначе уже на первом проходе процесса преобразования будет

выдано сообщение об ошибке; при этом эти инструкции не будут конвертированы;

- При удалении многократно повторяющихся неконвертируемых разделов программы Вы можете удалять эти разделы, заменяя их на "уникальную" STEP 5-инструкцию в каждом разделе; для этого Вы должны назначить макросу (последовательности инструкций STEP 7) эту "уникальную" STEP 5-инструкцию, заменяющую раздел программы.
- Если Ваша программа содержит много блоков (или большой блок) неструктурированных данных (что используется, например, в буферах данных), то Вы можете значительно уменьшить число инструкций, которые должны быть скомпилированы, и, следовательно, исходный код программы, если Вы удаляете все кроме одного слова данных в данном блоке. После конвертирования программы (перед компиляцией) содержимое этого блока данных в исходном файле объявляется массивом, например:

```
Buffer : ARRAY[1..256] OF WORD;
```

Вы можете использовать конвертер не только для конвертирования целых программ, но также для конвертирования отдельных блоков.

32.3 Конвертирование

32.3.1 Создание макросов

Вы можете создавать макросы для замены неконвертируемых инструкций STEP 5 перед тем, как начать процесс конвертирования программы, или для выполнения изменений в программе после стандартного преобразования (конвертирования). Вы можете создавать "макросы конвертирования" ("conversion macros") при помощи конвертера. Если макрос определен дважды, то используется первое его определение. Макросы для набора инструкций SIMATIC с использованием немецких мнемоник хранятся в файле S7U5CAPA.MAC, а макросы для набора инструкций с использованием интернациональных (английских) мнемоник хранятся в файле S7U5CAPB.MAC. Конвертер распознает макросы для инструкций (instruction macros) и макросы для OB (OB macros). Пользователю предоставляется возможность создать 256 макросов для инструкций и 256 макросов для OB.

Макросы для инструкций (instruction macros) заменяют инструкции STEP 5 на соответствующие последовательности инструкций STEP 7. Общая структура макросов для инструкций (instruction macros) имеет следующий вид:

```
$MACRO: <инструкция STEP 5>  
  
<последовательность инструкций STEP 7>  
  
$ENDMACRO
```

Инструкции STEP 5 при этом должны быть определены со своими полными адресами. Тогда конвертер вставит вместо инструкций STEP 5 соответствующие последовательности инструкций STEP 7.

Пример:

Пусть в программе STEP 5 для CPU 945 использовалось прерывания с задержкой обработки ("delay interrupt" - организационный блок OB 6). Активация прерывания осуществляется посредством вызова специальной функции OB 250:

```
L   KF   +200
L   KB     1
JU  OB   250
```

В первой инструкции загрузки содержится число микросекунд, определяющих период задержки вызова организационного блока OB 6. Эта инструкция может быть оставлена, тогда как оставшиеся две инструкции должны быть заменены инструкциями STEP 5, которые не встречаются в Вашей программе, например, TB RT 200.0; таким образом, Ваша программа для STEP 5 перед конвертированием имеет следующий вид:

```
L   KF   +200
TB  RT   200.0
```

Теперь Вы можете записать следующую макро-инструкцию:

```
$MACRO: TB RT 200.0
T MD 250;
CALL SFC 32 (
OB_NO      := 20,
DTIME     := MD 250,
SIGN       := W#16#0000,
RET_VAL    := MW 254);
$ENDMACRO
```

Инструкция STEP 5 TB RT 200.0 заменяется при конвертировании программы определенной последовательностью инструкций STEP 7. Значение времени задержки, выраженное в миллисекундах, загружается в ("сверхоперативную память") в слово меркеров MW 250; после этого вызывается системная функция SFC 32. В диалоговом окне перед запуском преобразования конвертер предлагает номер 20 вместо номера 6 для организационного блока обработки прерывания.

Макросы для организационных блоков (OB macros) заменяют вызовы OB (JU OB или JC OB) в программе STEP 5 на определенные последовательности инструкций STEP 7. Общая структура макросов для организационных блоков (OB macros) имеет следующий вид:

```
$MACRO: <Номер OB>
<последовательность инструкций STEP 7>
$ENDMACRO
```

Пример:

Пусть в программе STEP 5 для CPU 945 Вы использовали организационный блок OB 160 для запуска прерывания с задержкой обработки.

В STEP 7 функция задержки времени осуществляется посредством вызова системной функции SFC 47 WAIT. Если Вы запрограммируете следующий макрос:

```
$OBCALL: 160  
T MW 250;  
CALL SFC 47 (WT := MW 250);  
$ENDMACRO
```

то конвертер будет заменять каждый вызов организационного блока OB 160 (в том числе и все вызовы по условию) определенной последовательностью инструкций.

Ввод макроса начинается с выбора следующих опций меню: *Edit -> Replace Macro (Редактор -> Макрос замены)*. При этом открывается файл S7U5CAPA.MAC, в который Вы вводите макрос. После окончания ввода Вы сохраняете файл: *File -> Save (Файл -> Сохранить)*. Завершение работы с файлом макроса выполняется с помощью опций меню: *File -> Exit (Файл -> Выход)*.

32.3.2 Подготовка к конвертированию

Если у Вас нет таблицы перекрестных ссылок *nameXR.INI* для Вашей программы в STEP 5, то для преобразования программы Вам необходимо создать эту таблицу (в системе STEP 5 с помощью опций меню: *Manage -> Create XREF (Управление -> Создание таблицы перекрестных ссылок)*).

Теперь Вы можете:

- создать Ваш собственный рабочий раздел (папку) для преобразованной программы и скопировать требуемые данные в этот раздел или
- запустить процесс преобразования программы в разделе (в папке), содержащем файлы STEP 5 (если Вы работаете с одним и тем же программатором в системах STEP 5 [и STEP 7]) или
- запустить процесс преобразования программы на дискете (если Вы создали файлы STEP 5 с помощью иного программатора).

Раздел (папка) для преобразованной программы должен содержать файлы *nameST.S5D*, *nameXR.INI*, а также, если есть назначение, файл *nameZ0.SEQ*. Конвертер в свою очередь добавит в этот раздел "целевые файлы" *nameAC.AWL*, *nameA0.SEQ* и, если есть назначение, *nameAF.SEQ* и *nameS7.SEQ*.

Файл S7S5CAPx.MAC сохраняется в разделе Windows.

32.3.3 Запуск конвертера

Вызов S5/S7-конвертера производится с помощью опций меню, панели задач Windows 95/NT: *Start -> Simatic -> Step 7 -> Convert File (Пуск ->*

Simatic -> *Step 7* -> *Конвертировать файл*). С помощью опций: *File* -> *Open* (*Файл* -> *Открыть*) Вы можете выбрать файл S5-программы, которую необходимо конвертировать. При щелчке на кнопке "ОК" конвертер отобразит исходный файл и целевой файл, так же как и назначение "новых" блоков "старым". Если это необходимо, Вы можете изменить имена целевых файлов (файлов назначения) в соответствующем текстовом поле. Для изменения назначенных номеров блоков, дважды щелкните кнопкой манипулятора "мышь" в соответствующей строке и введите новый номер блока в диалоговом окне.

Конвертер отмечает стандартные блоки звездочкой (Вы должны в дальнейшем скопировать эти блоки из библиотеки блоков в Ваш автономный (offline) пользовательский раздел перед тем, как начнется процесс компилирования исходного S7-файла).

Запуск конвертера осуществляется кнопкой "Start" ("Пуск"). При первом проходе исходной программы конвертер преобразует S5-программу в текстовый файл в формате S5-ASCII (nameA0.SEQ), а на втором проходе переводит этот файл в исходный файл S7-программы. Список (таблица) назначений преобразуется в таблицу символов. В конце процесса конвертирования программы на экране отображаются сообщения об ошибках и предупреждения. Записи обо всех ошибках и предупреждениях содержатся в файле протоколирования ошибок *nameAF.SEQ*.

Сообщения об ошибках поступают, если отдельные фрагменты S5-программы не могут быть конвертированы и могут быть перенесены в S7-программу только в виде комментариев. Предупреждения содержат информацию о возможных проблемах; предупреждения поступают, если конвертируемые инструкции вновь требуют проверки. Предупреждения могут касаться S5-программы (например, обнаружен недопустимый код MC 5) или же могут относиться к S7-программе (например, если обнаружена непреобразованная инструкция). Если Вы щелкнете кнопкой манипулятора "мышь" на строке сообщения, конвертер в специальном окне отобразит ситуацию, касающуюся этого сообщения.

Рекомендуется распечатывать список возникающих ошибок, чтобы эффективно выполнить отладку программы.

32.3.4 Конвертируемые функции

В таблице 32.2 представлены инструкции, которые по существу остаются неизменными при конвертировании S5-программы в S7-программу. К их числу относятся также инструкции с адресами, которые заменяются в системе STEP 7 другими адресами (например, с такими адресами, как меркеры из "расширенной S области" ["extended S memory bits"], которые заменяются на адреса меркеров из области M, начиная с адреса 256). При преобразовании инструкций из этого списка могут также производиться изменения в их синтаксисе (например, вместо +G будет записано +R). Обычно Вам не придется корректировать рассматриваемые инструкции.

Таблица 32.2 Преобразование функций

| Функции в STEP 5 | Функции в STEP 7 |
|---|--|
| Двоичные логические операции, операции с памятью | Двоичные логические операции, операции с памятью |
| Функции таймеров и счетчиков (Timers/counters functions) | Функции таймеров и счетчиков (Timers/counters functions) |
| Функции проверки битов (Bit test functions) | Заменяется операцией установки SET с последующей проверкой или операцией двойного инвертирования Set/Reset (Установка/Сброс) |
| Функции загрузки (load) и пересылки (transfer) (исключая системные данные и абсолютную адресацию) | Функции загрузки (load) и пересылки (transfer) |
| Функции сравнения (Comparison functions) | Функции сравнения (Comparison functions) |
| Функции вычисления (Calculation functions) | Функции вычисления (Calculation functions) |
| Логические операции с числами (Digital logic operations) | Логические операции со словами (WORD logic operations) |
| Функции сдвига (Shift functions) | Функции сдвига (Shift functions) |
| Функции перехода (Jump functions) | Функции перехода (Jump functions) |
| Функции преобразования (Conversion functions) | Функции преобразования (Conversion functions) |
| Блокировка/разблокировка прерываний (Disable/Enable interrupts) | Заменяется на SFC 41, SFC 42 |
| Функции Stop (Stop functions) | Заменяется на SFC 46 |
| Нуль-операции (NOP, ***, пустая строка) (Null operations) | NOP, NETWORK, // (пустая строка комментария) |

Инструкции замены (осуществляющие доступ к параметрам блока) большей частью конвертируются. Некоторые корректировки должны быть выполнены для инструкций, касающихся функций счетчиков и функций таймеров (например, SEC =*param* [*имя_параметра*]), так же как и для обработки параметров блока (DO =*param* [*имя_параметра*]). В этом случае и кодовые блоки и блоки данных могут быть использованы в качестве фактических операндов и (что очень важно!): в результате конвертирования номер блока может быть изменен.

Организационные блоки содержат номера, используемые в STEP 7. Все остальные блоки с пользовательской программой становятся функциями FC. Конвертер конвертирует блоки данных DB в блоки глобальных данных с такими же номерами. Блоки данных DX конвертируются в блоки данных DB, начиная с номера 256 (блок DX 1 становится блоком DB 257, и т.д.). Конвертер предлагает пользователю номера блоков; и пользователь может изменить все назначенные номера блоков в диалоговом окне перед запуском процесса конвертирования.

Конвертер принимает библиотечные номера блоков в строке AUTHOR (Автор) в заголовке блока. Имена функциональных блоков принимаются как NAME (имя) без учета специальных символов (другими словами, имя принимается без специальных символов с комментированием исходного имени).

Вызовы специальных функций не конвертируются (они должны быть заменены, например, системными функциями).

Адреса входов и выходов принимаются неизменными. В случае использования инструкций загрузки (load) и пересылки (transfer) с адресами из P-области конвертер использует периферийные входы PI и периферийные выходы PQ с неизменными адресами. Адреса из Q-области накладываются на адресное пространство P-области (периферийные входы/выходы [I/O]), начиная с адреса 256 (так, инструкция L OV 0 заменяется на L PIB 256; инструкция T OV 1 заменяется на T PQV 257 и т.д.).

Адреса меркеров области F принимаются без изменения. Это же распространяется на меркеры, используемые как "сверхоперативная память" ("scratchpad memory"), начиная с байта меркеров FY 200 до FY 255. Если Вы конвертируете Вашу программу для STEP 5 в основном без изменения, Вы можете оставить "сверхоперативную память" ("scratchpad memory") как обычно. Если Вам необходимо продолжить использование программы STEP 5 или ее фрагментов в среде STEP 7, то автор данной монографии рекомендует "сверхоперативную память" как блок во временных локальных данных. Это особенно касается случая, если Вы желаете переслать Ваши собственные стандарты из программы STEP 5 в программу STEP 7. Меркеры из "расширенной S области" ["extended S memory bits"] размещаются в адресном пространстве меркеров, начиная с адреса 256 (так, инструкция A S 0.0 заменяется на A M 256.0; инструкция L SY 2 заменяется на L MB 258 и т.д.).

Функции таймеров и счетчиков конвертируются без изменения. После конвертирования этих функций становится невозможным прямой доступ в системе STEP 7 к отдельным битам слова значения для таймера и для счетчика. Обработка фронта сигнала состояния в отдельных битах этих слов с помощью операторов проверки состояния бита может быть заменена с использованием операторов SET и CLR вместе с соответствующими операциями таймера и счетчика.

Необходимо отметить, что в системе STEP 7 данные адресуются побайтно (в отличие от STEP 7 в системе STEP 5 данные адресуются "пословно"). Так, DL 0 заменяется на DBB 0; DR 0 заменяется на DBB 1.

В таблице 32.3 Вы можете видеть результаты преобразования адресов при конвертировании программы.

При прямой и косвенной адресации конвертер использует корректные S7-адреса; при адресации данных с помощью параметров блока Вы должны выполнить преобразование для побайтной адресации самостоятельно.

Числа с плавающей запятой принимаются при конвертировании без изменения при том, что они определены как константы в операциях загрузки (load) или они используются как фактические параметры, и они трактуются при преобразовании как числа с плавающей запятой в системе STEP 7. Стандартные блоки, принимаемые для замены стандартных блоков STEP 5, также обрабатывают числа с плавающей запятой в формате STEP 7 (тип данных REAL).

Таблица 32.3 Преобразование адресов при конвертировании программы

| STEP 5 | STEP 7 |
|---------------|-------------------|
| DL [n] | DBB [2n] |
| DR [n] | DBB [2n+1] |
| DW [n] | DBW [2n] |
| DD [n] | DBD [2n] |
| D [(n).0..7] | DBX [(2n+1).0..7] |
| D [(n).8..15] | DBX [(2n).0..7] |

Если Вы в Вашей программе для STEP 5 самостоятельно набрали данные в формате чисел с плавающей запятой или, если Вы получили их от других устройств, например, посредством коммуникаций, то Вы должны преобразовать представление этих данных (чисел с плавающей запятой) в STEP 5 в тип данных REAL.

32.4 Последующее редактирование

32.4.1 Создание проекта в STEP 7

Для завершения процесса конвертирования Вы должны создать проект STEP 7, который должен соответствовать по структуре Вашей системе назначения (PLC) (если Вы еще не создали такого проекта во время ознакомления с адресацией модулей S7). Если необходимо изменить адреса модулей, параметризацию модулей или изменить параметры работы (execution properties) CPU, Вам необходимо выполнить конфигурирование оборудования (то есть, полностью установить проект). Если установки по умолчанию для параметров модулей не могут быть изменены, то достаточно установить независимую от модулей программу (module-independent program).

Итак для создания проекта:

- Вы создаете станцию (S7-300 или S7-400), открываете объект *Hardware (Оборудование)*, после чего конфигурируете станцию. Также с помощью утилиты конфигурирования оборудования Вы устанавливаете свойства CPU (например, номера ОВ прерывания). Вместе с CPU утилита SIMATIC Manager создает также разделы для объектов следующего уровня.
- Выделив объект *Sources (Исходные)*, Вы с помощью опций меню: *Insert -> External Sources File...* (*Вставка -> Внешний исходный файл...*) вносите созданный файл *nameAC.AWL* в раздел (в папку) для исходной программы.
- Если в Вашей программе используются стандартные S5-блоки, откройте библиотеку *S5/S7 Converting Blocks (Блоки для S5/S7-преобразования)* в разделе стандартной библиотеки *Standard Library* и

скопируйте в автономный раздел *Blocks (Блоки)* Вашего проекта стандартные S7-блоки, отмеченные конвертером в списке блоков звездочкой. Если Вы используете системные S7-блоки в конвертированной программе (например, SFC 20 BLKMOV), то откройте библиотеку *System Function Blocks (Системные функциональные блоки)* и скопируйте используемые системные блоки, в автономный раздел *Blocks (Блоки)* Вашего проекта.

- Если в Вашей программе используется символьная адресация, откройте (пустую) таблицу символов *Symbols* и выберите с помощью опций меню: *Symboltable -> Import... (Таблица символов -> Импорт...)* конвертированные символы *nameS7.SEQ* для программы.

Выполнив указанные подготовительные пункты, Вы можете теперь с помощью редактора обработать исходный файл программы перед тем, как начнете процесс ее компиляции (Вы можете уменьшить число сообщений об ошибках, если Вы выполните все корректировки до компиляции программы).

32.4.2 Неконвертируемые функции

После выполнения конвертирования программы обычно требуется редактирование исходного файла программы. Корректировка должна выполняться в отношении инструкций, перечисленных в таблице 32.4.

Таблица 32.4 Неконвертируемые функции

| Функции в STEP 5 | Примечание |
|--|---|
| Функции загрузки (load) и пересылки (transfer) с системными данными с абсолютными адресами | Заменяются системными функциями Должны заменяться новой программой |
| Операции с регистрами (LIR, TIR, LDI, TDI, MBA, MAB, MSA, MAS, MSB, MBR, ABR, ACR) | Должны заменяться новой программой |
| Функции пересылки блока (TNB, TNW, TXB, TXW) | Должны быть заменены системной функцией SFC 20 BLKMOV |
| Функции DO DO DW, DO FW DO RS | Конвертируется Должна заменяться новой программой |
| Вызов специальных функций | Заменяется специальными функциями SFC |
| Функции LIM, SIM, IAE, RAE | Могут заменяться функциями из диапазона SFC 39 ... SFC 42 |
| "Семафорные" функции (Semafore functions: SED, SEE, TSC, TSG) | Нет замещения |
| Другие функции (IAI, RAI, ASM, UBE) | Нет замещения |

32.4.3 Изменение адресов

Вопрос изменения адресов особенно касается входных и выходных модулей. Вы должны адаптировать систему доступа в конкретных условиях к входам и выходам, так же как и систему прямого доступа к периферии (I/O) к новым значениям адресов для модулей. Вы должны выполнить эту адаптацию в файле STEP 5 перед его конвертированием (если в системе STEP 5 поддерживается требуемый объем адресного пространства), или Вы можете заменить абсолютные адреса в исходном S7-файле с помощью функции замены "Replace", встроенной в редактор (будьте внимательны, особенно в случае, если "старые" и "новые" адресные области перекрываются).

В случае, если при программировании Вы используете символьную адресацию, Вы можете также создать исходную программу с символьными адресами; измените абсолютные адреса в таблице символов и затем вновь перекомпилируйте ее. Порядок выполнения действий предлагается ниже:

- Требуется, чтобы в распоряжении пользователя была таблица символов с символами для всех абсолютных адресов, которые должны быть изменены; требуется также, чтобы программа была скомпилирована без ошибок (блоки, содержащие абсолютные адреса, должны быть доступны в скомпилированной форме).
- Установите редактор на использование символьной адресации: посредством опций меню *Options -> Customize (Опции -> Установка пользователя)* откройте диалоговое окно, в котором на вкладке "Editor" ("Редактор") выберите опцию для включения режима символов *Symbolic Representation (Символьное представление)*.
- В редакторе посредством выбора опций меню: *File -> Generate Source File... (Файл -> Создать исходный файл...)* начните создание нового исходного файла. После ввода имени файла выберите все блоки в диалоговом окне, которые необходимы для исходного файла с символьной адресацией. Теперь новый исходный файл будет содержать инструкции с символьными адресами.
- Далее, измените все абсолютные адреса в таблице символов *Symbols* со "старых" значений для версии S5 на "новые" значения для версии S7.
- Если теперь Вы скомпилируете новый исходный файл, то в скомпилированных блоках будут содержаться новые абсолютные адреса.

32.4.4 Косвенная адресация

Конвертер может распознавать также косвенную адресацию в функциях DO MW и DO DW в системе STEP 7. Тем не менее, необходимо при этом преобразовать (конвертировать) указатель в формат STEP 7, что, с учетом буферизации содержимого аккумулятора и слова состояния, ведет к возрастанию требуемого объема памяти.

При правильном программировании обычно Вы можете использовать косвенную адресацию - или косвенную адресацию посредством памяти, или косвенную адресацию посредством регистра - с использованием небольшого количества инструкций, получив в результате программу с ясной структурой. Если косвенная адресация используется часто, то программирование в системе STEP 7 имеет определенные преимущества.

Программирование в системе STEP 7 имеет значительные преимущества, в случае, если косвенная адресация используется максимально часто.

- Косвенная адресация таймеров, счетчиков и блоков
Такая адресация конвертируется в косвенную адресацию посредством памяти с использованием одного слова в области временных локальных данных.
- Косвенная адресация блоков
Размещение номеров новых блоков не может приниматься в расчет (корректируется в ручном режиме).
- Косвенная адресация
Конвертируется в побитовом и "пословном" режиме посредством AR1, с буферированием содержимого слова состояния STW и аккумуляторов Асси 1 и Асси 2 во временных локальных данных (см. ниже).
- Косвенная адресация посредством регистра BR
Конвертирование невозможно. Корректируется в ручном режиме с использованием адресных регистров.
- Другие варианты косвенной адресации
Должны корректироваться в ручном режиме.

| | |
|---|---|
| Функции перехода (Jump functions) | Заменяются функцией распределителя переходов SPL |
| Функции сдвига (Shift functions) | Заменяются функцией сдвига с числом позиций, указываемых в аккумуляторе Ассим 2 |
| Функции TNB, TNW | Заменяются системной функцией SFC 20 BLKMOV с "переменной" указатель ANY. |
| Функции LIR, TIR | Прямая замена невозможна |
| Функции декрементирования или инкрементирования | Прямая замена невозможна |

Конвертер изменяет косвенную адресацию с DO MW и DO DW двоичных логических операций, функций с памятью и функций загрузки (load) и пересылки (transfer) для STEP 7 -программы. Указатель из программы для STEP 5 должен быть изменен на внутризонный указатель в формате STEP 7 (с буферированием содержимого аккумулятора и слова состояния). В результате получается длинная цепочка инструкций (см. пример ниже).

Если Вы в Вашей программе изменили большое число косвенных адресов, то ручной режим конвертирования программы может дать определенное преимущество.

У пользователя есть неограниченный доступ к двум адресным регистрам AR1 и AR2 как индексным регистрам (в функциях FC). Вы можете также адресовать меркеры или данные в режиме косвенной адресации посредством памяти как в системе STEP 5, но Вам при этом потребуется на индексный регистр одно двойное слово вместо одного машинного слова.

В таблице 32.5 представлен пример, в котором в первом столбце расположена программа для STEP 5, выполняющая сравнение значения входного слова с массивом данных, состоящим из битовых элементов; в случае совпадения сравниваемых значений устанавливается меркер. Во втором столбце содержится программа после конвертирования. Используя оба адресных регистра, Вы можете записать новую программу для прямого сравнения значений, для которой потребуется значительно меньшее число инструкций (см. третий столбец таблицы).

Сначала адресные регистры загружаются с указателями (примите во внимание побайтную адресацию данных!). Доступ к словам и к меркерам организован с помощью косвенной адресации посредством регистра. После каждой операции сравнения значение в адресном регистре AR1 инкрементируется на 2 байта, а адресный регистр AR 2 инкрементируется на 1 бит (конвертирование в байтовый адрес пропущено). В рассматриваемом примере указатель на слова данных используется в соответствии с критерием "точки разрыва" ("break criterion"), как в системе STEP 5; в такой точке система STEP 7 организует циклический переход LOOP.

32.4.5 Доступ к "чрезмерно большим" блокам данных

Доступ к "чрезмерно большим" блокам данных, то есть, доступ к данным, которые имеют значение байтового адреса больше 255, в системе STEP 5 выполнялось с помощью абсолютной адресации. При этом вычислялся начальный адрес блока данных, добавлялось адресное смещение и производилось обращение к данным или прямое (посредством функций LIR/TIR), или с помощью регистра BR (посредством функций LRW/TRW).

В системе STEP 7 Вы можете назначать адреса данных непосредственно (прямым указанием адреса) в разрешенном диапазоне вплоть до значения 8095 (в системе S7-300) и 32767 (в системе S7-400).

Следовательно, заменяя доступ с применением абсолютной адресации при переходе к системе STEP 7, Вы можете использовать адреса данных в "обычных" STL-инструкциях.

32.4.6 Использование абсолютных адресов

Абсолютная адресация может использоваться, если необходимо обрабатывать абсолютные адреса в памяти в системе STEP 5, если Вы назначаете адреса данных в "чрезмерно больших" блоках данных, или если Вы используете косвенную адресацию посредством BR-регистра, или если Вы используете передачу (transfer) блока. Доступ к абсолютным

Таблица 32.5 Конвертирование программы с косвенной адресацией

| Программа в STEP 5 | Программа после конвертирования | Программа после оптимизации |
|---|--|--|
| FB 174 Name : COMP | FUNCTION FC 4 VOID NAME: COMP VAR_TEMP conv_accum1 : dword; conv_accum2 : dword; conv_stw : word; END_VAR BEGIN NETWORK | FUNCTION FC 4 VOID NAME: COMP BEGIN |
| :L KB 20 :T DW 2 :L KB 50 :T DW 3 LOOP :L IW 10 | L 20; T DBW 4; L 50; T DBW 6; LOOP: L IW 10; | LAR1 P#40.0; LAR2 P#50.0; LOOP: L IW 10; |
| | T conv_accum1; L STW; T conv_stw; L DBB 5; SLW 4; LAR1; L conv_stw; T STW; L conv_accum1; | |
| :DO DW 2 :L DW 0 | L DBW[AR1,P#0.0]; | L DBW[AR1,P#0.0]; |
| :>F | >I; | >I; |
| | T conv_accum1; TAK; T conv_accum2; L STW; T conv_stw; L DBB 6; SLW 5; SRW 5; L DBB 7; SLW 3; OW; LAR1; L conv_stw; T STW; L conv_accum2; L conv_accum1; | |
| :DO DW 3 := F 0.0 | = M[AR1,P#0.0]; | = M[AR2,P#0.0]; |
| :L DW 2 :I 1 :T DW 2 :L KB 100 :>F :JC =END :L DL 3 :I 1 :T DL 3 :L KB 8 :<F :JC =LOOP :L DR 3 :I 1 :T DW 3 :JU =LOOP END :NOP 0 :BE | L DBW 4; INC 1; T DBW 4; L 100; >I; JC END; L DBB 6; INC 1; T DBB 6; L 8; <I; JC LOOP; L DBB 7; INC 1; T DBW 6; JU LOOP; END: NOP 0; END_FUNCTION | +AR1 P#2.0; CAR1; L P#200.0; >D; JC END; +AR2 P#0.1; JU LOOP; END: NOP 0; END_FUNCTION |

адресам далее не поддерживается в STEP 7; адресный счетчик STEP (для "связанных" операций) удаляется без замены.

Доступ к данным с адресами в "чрезмерно больших" блоках данных обеспечивается в STEP 7 непосредственно с помощью "обычных" инструкций. В связи с этим, вычисление адреса в блоках данных также пропускается. Очевидное решение для косвенной адресации с помощью BR-регистра заключается в использовании "косвенной адресации посредством регистра" и, если необходимо, то "межзонной адресации".

Системная функция SFC 20 BLKMOV заменяет операции в S5-программе по пересылке блоков (transfer). Вы определяете переменные или области памяти, которые должны быть скопированы непосредственно как параметры. Если Вы хотите переопределять исходные области памяти или области назначения во время выполнения программы, то Вы должны будете использовать "переменную" ANY-указатель в качестве фактического параметра.

32.4.7 Инициализация параметров

Конвертер принимает фактические параметры, которые используются при вызове блока без изменения. Если у Вас имеются определенные адреса с фактическими параметрами, Вы можете проверить их и, если необходимо, изменить.

Примеры:

- Определение числа в формате слова (WORD):
Такая адресация конвертируется в побайтную адресацию.
- Определение адреса (I/O):
При конвертировании должен использоваться новый адрес модуля.
- Передача (transfer) блока:
При конвертировании должен использоваться новый номер модуля.

32.4.8 Специальные функции организационных блоков

В STEP 7 Вы можете использовать системные функции или STL-инструкции для замены организационных блоков специальными функциями (см. табл. 32.6). Некоторые функции пропускаются полностью (например, такие, как страничная адресация ("page addressing"), доступ к системной программе).

32.4.9 Обработка ошибок

Способ сигнализации о нарушении диапазона допустимых значений посредством битов состояния OV и OS в STEP 7 аналогичен способу

Таблица 32.6 Конвертирование специальных функций организационных блоков

| Функция | 115U | 135U | 155U | Заменяется в STEP 7 |
|---|----------------|----------------|----------------|---|
| Байт условного кода для процесса (Process condition code byte) | - | 110 | - | Последовательность инструкций |
| Аккумуляторы процесса (Process accumulators) | - | 111-113 | 131-133 | Последовательность инструкций |
| Обработка прерываний (Handle interrupts) | - | 120-123 | 122 141-143 | SFC 39 DIS_IRT, SFC 40 EN_IRT, SFC 41 DIS_AIRT, SFC 42 EN_AIRT |
| Активация задания таймера (Activate a timer job) | - | 151 | 151 | SFC 28 SET_TINT, SFC 29 CAN_TINT, SFC 30 ACT_TINT, SFC 31 QRY_TINT |
| Обработка прерывания с задержкой обработки (Handle a delay interrupt) | - | 153 | 153 | SFC 32 SRT_DINT, SFC 33 QRY_DINT, SFC 34 CAN_DINT |
| Переменная времени ожидания (Variable waiting time) | 160 | - | - | SFC 43 WAIT |
| Удаление блока (Delete block) | - | - | 124 | Блок данных: SFC 23 DEL_DB |
| Создание блока (Create block) | 125 | - | 125 | Блок данных: SFC 22 CREAT_DB |
| Считывание стека блоков данных (Read block stack) | - | 170 | - | - отсутствует - |
| Проверка блока данных (Test data block) | - | 181 | - | SFC 24 TEST_DB |
| Обращение к блоку данных (Data block access) | - | 180 | - | - отсутствует - |
| Копирование блоков данных (Copy data blocks) | 183, 184 | 254, 255 | 254, 255 | SFC 20 BLKMOV (области данных) |
| Копирование области данных (Copy data areas) | 182 190-193 | 182 190-193 | - | SFC 20 BLKMOV |
| Задание и считывание времени суток (Set & read time-of-day) | - | 150 | 121, 150 | SFC 0 SET_CLK, SFC 1 READ_CLK |
| Данные по циклу (Cycle statistics) | - | 152 | - | Стартовая информация OB 1, SFC 6 RD_SINFO |
| Считывание информации о состоянии (Read status inf) | - | 228 | - | Стартовая информация, SFC 6 RD_SINFO |

Таблица 32.6 (продолжение) Конвертирование специальных функций организационных блоков

| Функция | 115U | 135U | 155U | Заменяется в STEP 7 |
|---|----------|----------|---------|---|
| Коммуникации мульти-процессорного режима (Multiprocessor communications) | - | 200-205 | 200-205 | Замена: GD-коммуникации |
| Сравнение типов перезапуска (Compare restart types) | - | 223 | 223 | - отсутствует - |
| Флаги передачи межпроцессорных коммуникаций (Transfer interprocessor communication flags) | - | 224 | - | GD-коммуникации |
| Задание времени цикла (Set cycle time) | - | 221 | - | Параметризация CPU |
| Перезапуск контроля цикла (Cycle time triggering) | - | 222 | 31, 222 | SFC 43 RE_TRIGR |
| Передача образа процесса (Transfer process images) | 254, 255 | - | 126 | SFC 26 UPDAT_PI, SFC 27 UPDAT_PO |
| Кольцевой счетчик (регистр) (Counter loop) | - | 160-163 | - | Последовательность инструкций |
| Заполнение знаком (Sign extension) | 220 | 220 | - | Последовательность инструкций |
| Постраничный доступ (Page accesses) | - | 216-218 | - | - отсутствует - |
| Доступ к системной программе (System program access) | - | 226, 227 | - | - отсутствует - |
| Регистр сдвига процесса (Process shift register) | - | 240-248 | - | - отсутствует - |
| Обработка блоков (Handling blocks) | - | 230-237 | - | Коммуникационные блоки SFC |
| ПИД-алгоритм (PID algorithm) | 251 | 250-251 | - | Стандартные блоки для ПИД-регулирования |
| Обслуживание работы системы (Execute system service) | 250 | - | - | (см. выше "Контроль параметров выполнения программы") |

сигнализации в STEP 5, но при этом существуют небольшие отличия. Если Вы проверяете биты состояния OV и OS, обращайтесь внимание на точность выполнения функций для соответствующих инструкций (например, арифметических функций).

Почти все системные функции SFC сигнализируют об ошибках посредством функционального значения (возвращаемого значения функции) RET_VAL.

Для обработки ошибок в STEP 7 существуют организационные блоки: для обработки синхронных ошибок - блоки OB 121, OB 122 и для обработки асинхронных ошибок - блоки с номерами от OB 80 до OB 87. В таблице 32.7 Вы можете видеть, как заменяются блоки обработки ошибок при переводе программы из STEP 5 в STEP 7.

Таблица 32.7 Конвертирование организационных блоков обработки ошибок

| Функция | S5-115 | S5-135 | S5-155 | Замена в S7 |
|--|--------|--------|--------|-------------|
| Вызов незагруженного блока | 19 | 19 | 19 | OB 121 |
| Задержка квитирования при прямом доступе к I/O модулям | 23 | 23 | 23 | OB 122 |
| Квитирование при обновлении образа процесса | 24 | 24 | 24 | OB 122 |
| Ошибки адресации | - | 25 | 25 | OB 122 |
| Превышено максимальное время цикла | 26 | 26 | 26 | OB 80 |
| Ошибки подстановки | 27 | 27 | 27 | - |
| Остановка по условию (Conditional STOP) | - | 28 | - | - |
| Задержка квитирования в случае входного байта IB 0 | - | - | 28 | OB 85 |
| Некорректный код операции | - | 29 | - | STOP |
| Задержка квитирования при прямом доступе к области периферийных входов/выходов (I/O) | - | - | 29 | OB 122 |
| Некорректные параметры | - | 30 | - | - |
| Ошибки проверки на четность или подтверждения в случае доступа к пользовательской памяти | - | - | 30 | OB 122 |
| Групповые ошибки специальной функции | - | 31 | - | - |
| Ошибки передачи в блоках данных | 32 | 32 | 32 | OB 121 |
| Ошибки таймера при запуске, управляемом по времени | 33 | 33 | 33 | OB 80 |
| Отказ батареи питания | 34 | - | - | OB 81 |
| Ошибки контроллера | - | 34 | - | - |
| Ошибки при создании блока данных | - | - | 34 | (SFC) |
| I/O ошибки | 35 | - | - | OB 86 |
| Ошибки интерфейса | - | 35 | - | OB 84 |
| Ошибки самоконтроля (Self-test errors) | - | - | 36 | - |

33 Библиотеки блоков

Базовое ПО STEP 7 включает в себя стандартную библиотеку *Standard Library*, которая содержит следующие разделы для блоков различных типов:

- Организационные блоки (OB);
- Системные функциональные блоки (SFB);
- Функциональные IEC-блоки (загружаемые IEC-функции);
- Блоки для S5-S7-преобразования (загружаемые функции преобразования);
- Блоки для T1-S7-преобразования (загружаемые функции преобразования);
- Блоки ПИД-управления (функции для автоматического управления);
- Коммуникационные блоки (DP-функции).

Вы можете копировать блоки или интерфейсы системных функций или системных функциональных блоков в свой собственный проект или библиотеки.

33.1 Организационные блоки (OB)

(Prio = приоритетный класс, принимаемый по умолчанию)

| OB | Prio | Назначение | |
|----|------|-------------------------------|---------------------------|
| 1 | 1 | Основная программа | (Main program) |
| 10 | 2 | Прерывание по времени суток 0 | (Time-of-day interrupt 0) |
| 11 | 2 | Прерывание по времени суток 1 | (Time-of-day interrupt 1) |
| 12 | 2 | Прерывание по времени суток 2 | (Time-of-day interrupt 2) |
| 13 | 2 | Прерывание по времени суток 3 | (Time-of-day interrupt 3) |
| 14 | 2 | Прерывание по времени суток 4 | (Time-of-day interrupt 4) |
| 15 | 2 | Прерывание по времени суток 5 | (Time-of-day interrupt 5) |

| ОВ | Prio | Назначение | |
|----|------|--|--|
| 16 | 2 | Прерывание по времени суток 6 | (Time-of-day interrupt 6) |
| 17 | 2 | Прерывание по времени суток 7 | (Time-of-day interrupt 7) |
| 20 | 3 | Прерывание с задержкой обработки 0 | (Time-delay interrupt 0) |
| 21 | 4 | Прерывание с задержкой обработки 1 | (Time-delay interrupt 1) |
| 22 | 5 | Прерывание с задержкой обработки 2 | (Time-delay interrupt 2) |
| 23 | 6 | Прерывание с задержкой обработки 3 | (Time-delay interrupt 3) |
| 30 | 7 | Прерывание таймерное 0 (5 с) | (Watchdog interrupt 0 [5 s]) |
| 31 | 8 | Прерывание таймерное 1 (2 с) | (Watchdog interrupt 1 [2 s]) |
| 32 | 9 | Прерывание таймерное 2 (1 с) | (Watchdog interrupt 2 [1 s]) |
| 33 | 10 | Прерывание таймерное 3 (500 мс) | (Watchdog interrupt 3 [500 ms]) |
| 34 | 11 | Прерывание таймерное 4 (200 мс) | (Watchdog interrupt 4 [200 ms]) |
| 35 | 12 | Прерывание таймерное 5 (100 мс) | (Watchdog interrupt 5 [100 ms]) |
| 36 | 13 | Прерывание таймерное 6 (50 мс) | (Watchdog interrupt 6 [50 ms]) |
| 37 | 14 | Прерывание таймерное 7 (20 мс) | (Watchdog interrupt 7 [20 ms]) |
| 38 | 15 | Прерывание таймерное 8 (10 мс) | (Watchdog interrupt 8 [10 ms]) |
| 40 | 16 | Аппаратное прерывание 0 | (Hardware interrupt 0) |
| 41 | 17 | Аппаратное прерывание 1 | (Hardware interrupt 1) |
| 42 | 18 | Аппаратное прерывание 2 | (Hardware interrupt 2) |
| 43 | 19 | Аппаратное прерывание 3 | (Hardware interrupt 3) |
| 44 | 20 | Аппаратное прерывание 4 | (Hardware interrupt 4) |
| 45 | 21 | Аппаратное прерывание 5 | (Hardware interrupt 5) |
| 46 | 22 | Аппаратное прерывание 6 | (Hardware interrupt 6) |
| 47 | 23 | Аппаратное прерывание 7 | (Hardware interrupt 7) |
| 60 | 25 | Мультипроцессорное прерывание | (Multiprocessor interrupt) |
| 70 | 25 | Ошибка резервирования I/O ¹⁾ | (I/O redundancy error) ¹⁾ |
| 72 | 28 | Ошибка резервирования CPU | (CPU redundancy error) |
| 73 | 25 | Ошибка резервирования коммуникаций | (Communication redundancy error) |
| 80 | 26 | Временная ошибка ¹⁾ | (Time error) ¹⁾ |
| 81 | 26 | Неисправность источника питания ¹⁾ | (Power supply fault) ¹⁾ |
| 82 | 26 | Диагностическое прерывание ¹⁾ | (Diagnostics interrupt) ¹⁾ |
| 83 | 26 | Прерывание вставки/удаления модуля ¹⁾ | (Insert/remove-module interrupt) ¹⁾ |
| 84 | 26 | Неисправность CPU ¹⁾ | (CPU hardware fault) ¹⁾ |
| 85 | 26 | Ошибка приоритетного класса ¹⁾ | (Priority class error) ¹⁾ |

| ОВ | Prio | Назначение | |
|-----|------|------------------------------------|-------------------------------------|
| 86 | 26 | DP-ошибка ¹⁾ | (DP error) ¹⁾ |
| 87 | 26 | Ошибка системы связи ¹⁾ | (Communication error) ¹⁾ |
| 90 | 29 | Обработка в фоновом режиме | (Background processing) |
| 100 | 27 | Полный перезапуск | (Complete restart) |
| 101 | 27 | Перезапуск | (Restart) |
| 102 | 27 | "Холодный" перезапуск | (Cold restart) |
| 121 | - | Ошибка программирования | (Programming error) |
| 122 | - | Ошибка доступа к I/O | (I/O access error) |

¹⁾ Prio = 28 при перезапуске (Prio = приоритетный класс, принимаемый по умолчанию)

33.2 Системные функциональные блоки (SFB)

IEC-функции таймеров и IEC-функции счетчиков

| SFB | Имя | Назначение | |
|-----|------|--|-------------------|
| 0 | CTU | Счетчик в режиме прямого счета | (Up counter) |
| 1 | CTD | Счетчик в режиме обратного счета | (Down counter) |
| 2 | CTUD | Счетчик в режиме прямого/обратного счета | (Up/down counter) |
| 3 | TP | Генерация импульса | (Pulse) |
| 4 | TON | Задержка включения импульса | (On delay) |
| 5 | TOF | Задержка выключения импульса | (Off delay) |

Коммуникации посредством сконфигурированных соединений

| SFB | Имя | Назначение | |
|-----|-------|-------------------------------|--------------------------|
| 8 | USEND | Нескоординированная передача | (Uncoordinated send) |
| 9 | URVC | Нескоординированный прием | (Uncoordinated receive) |
| 12 | BSEND | Поблочная передача | (Block-oriented send) |
| 13 | BRVC | Поблочный прием | (Block-oriented receive) |
| 14 | GET | Считывание данных от партнера | (Read data from partner) |
| 15 | PUT | Запись данных в партнере | (Write data to partner) |
| 16 | PRINT | Печать данных на принтере | (Write data to printer) |

Коммуникации посредством сконфигурированных соединений (*продолжение*)

| SFB | Имя | Назначение | |
|-----|---------|--|--|
| 19 | START | Инициация полного перезапуска партнера | (Initiate complete restart in partner) |
| 20 | STOP | Перевод партнера в режим STOP | (Set partner to STOP) |
| 21 | RESUME | Инициация перезапуска партнера | (Initiate restart in partner) |
| 22 | STATUS | Проверка состояния партнера | (Check status of partner) |
| 23 | USTATUS | Прием данных о состоянии партнера | (Receive status of partner) |

| SFC | Имя | Назначение | |
|-----|---------|---------------------------------|-------------------------------|
| 62 | CONTROL | Проверка состояния коммуникаций | (Check communications status) |

Встроенные функции CPU 312/314/614

| SFB | Имя | Назначение | |
|-----|----------|---------------------------------|----------------------------------|
| 29 | HS_COUNT | "Быстрый" счетчик | (High-speed counter) |
| 30 | FREQ_MES | Измеритель частоты | (Frequency meter) |
| 38 | HSC_A_B | Управление счетчиком "A/B" | (Control "Counter A/B") |
| 39 | POS | Функция позиционирования | (Control "Positioning") |
| 41 | CONT_C | Функция непрерывного управления | (Continuous closed-loop control) |
| 42 | CONT_S | Функция пошагового управления | (Step-action control) |
| 43 | PULSEGEN | Генерация импульса | (Generate pulse) |

| SFC | Имя | Назначение | |
|-----|---------|---|------------------------|
| 63 | AB_CALL | Функция вызова блока скомпонованного кода | (Call assembler block) |

Функции системной диагностики

| SFC | Имя | Назначение | |
|-----|----------|---------------------------------------|-----------------------------------|
| 6 | RD_SINFO | Считывание стартовой информации | (Read start information) |
| 51 | RDSYSST | Считывание информации подписка SYS ST | (Read SYS ST sublist) |
| 52 | WR_USMSG | Ввод в диагностический буфер | (Entry in the diagnostics buffer) |

Генерация сообщений, связанных с блоками

| SFB | Имя | Назначение | |
|-----|----------|---------------------------------------|---|
| 33 | ALARM | Сообщения с подтверждением | (Messages with acknowledgment display) |
| 34 | ALARM_8 | Сообщения без сопутствующих значений | (Messages without accompanying values) |
| 35 | ALARM_8P | Сообщения с сопутствующими значениями | (Messages with accompanying values) |
| 36 | NOTIFY | Сообщения без подтверждения | (Messages without acknowledgment display) |
| 37 | AR_SEND | Посылка архивных данных | (Send archive data) |

| SFC | Имя | Назначение | |
|-----|----------|---|---|
| 9 | EN_MSG | Разрешить сообщения | (Enable messages) |
| 10 | DIS_MSG | Заблокировать сообщения | (Disable messages) |
| 17 | ALARM_SQ | Сообщения, которые могут быть квитированы | (Messages that can be acknowledged) |
| 18 | ALARM_S | Сообщения, которые всегда квитировуются | (Messages that are always acknowledged) |
| 19 | ALARM_SC | Определение состояния квитирования | (Determine acknowledgment status) |

Встроенные часы CPU и измерение времени наработки

| SFC | Имя | Назначение | |
|-----|----------|---|----------------------------|
| 0 | SET_CLK | Установить часы | (Set clock) |
| 1 | READ_CLK | Считать значение времени | (Read clock) |
| 2 | SET_RTM | Установить измеритель времени наработки | (Set run-time meter) |
| 3 | CTRL_RTM | Скорректировать измеритель времени наработки | (Modify run-time meter) |
| 4 | READ_RTM | Считать значение измерителя времени наработки | (Read run-time meter) |
| 48 | SNC_RTCS | Синхронизировать вторичные (ведомые) часы | (Synchronize slave clocks) |
| 64 | TIME_TCK | Считать системное время | (Read system time) |

Генератор последовательностей Drum

| SFC | Имя | Назначение |
|-----|------|---|
| 32 | DRUM | Функция генератора последовательностей (Drum) |

Функции копирования и функции для работы с блоками

| SFC | Имя | Назначение |
|-----|----------|--|
| 20 | BLKMOV | Копирование данных из области памяти (Copy data area) |
| 21 | FILL | Инициализация области памяти (Pre-assign data area) |
| 22 | CREAT_DB | Создание блока данных (Generate data block) |
| 23 | DEL_DB | Удаление блока данных (Delete data block) |
| 24 | TEST_DB | Тестирование блока данных (Test data block) |
| 25 | COMPRESS | "Сжатие" памяти (Compress memory) |
| 44 | REPL_VAL | Введение заменяющего значения (Enter substitute value) |
| 81 | UBLKMOV | Копирование данных "без зазоров" (Copy Data area without gaps) |

Адреса модулей

| SFC | Имя | Назначение |
|-----|----------|---|
| 5 | GADR_LGC | Определение логического адреса (Determine logical address) |
| 49 | LGC_GADR | Определение слота (Determine slot) |
| 50 | RD_LGADR | Определение всех логических адресов (Determine all logical addresses) |

Распределенная периферия (I/O)

| SFC | Имя | Назначение |
|-----|----------|--|
| 7 | DP_PRAL | Инициация аппаратного прерывания (Initiate hardware interrupt) |
| 11 | DPSYN_FR | Синхронизация групп устройств (SYNC/FREEZE) |
| 12 | D_ACT_DP | Деактивация/активация ведомого (slave) DP-устройства (Deactivate or activate DP slave) |
| 13 | DPNRM_DG | Считывание диагностических данных (Read diagnostics data) |
| 14 | DPRD_DAT | Считывание данных из ведомого (slave) DP-устройства (Read slave data) |
| 15 | DPWR_DAT | Запись данных в ведомое (slave) DP-устройство (Write slave data) |

Управление программой

| SFC | Имя | Назначение | |
|-----|----------|-------------------------------------|--------------------------------|
| 43 | RE_TRIGR | Перезапуск контроля времени цикла | (Retrigger cycle time monitor) |
| 46 | STP | Переход в режим STOP | (Change to STOP state) |
| 47 | WAIT | Ожидание в течение времени задержки | (Wait for delay time) |

Пересылка записей данных

| SFC | Имя | Назначение | |
|-----|----------|--|------------------------------|
| 54 | RD_DPARM | Считывание предопределенного параметра | (Read predefined parameter) |
| 55 | WR_PARM | Запись динамического параметра | (Write dynamic parameter) |
| 56 | WR_DPARM | Запись предопределенного параметра | (Write predefined parameter) |
| 57 | PARM_MOD | Параметризация модуля | (Parameterize module) |
| 58 | WR_REC | Запись записи данных | (Write data record) |
| 59 | RD_REC | Считывание записи данных | (Read data record) |

Обновление образа процесса

| SFC | Имя | Назначение | |
|-----|----------|--|-------------------------------------|
| 26 | UPDAT_PI | Обновление таблицы входов образа процесса | (Update process-image input table) |
| 27 | UPDAT_PO | Обновление таблицы выходов образа процесса | (Update process-image output table) |
| 79 | SET | Установка битовых массивов I/O | (Set I/O bit field) |
| 80 | RSET | Сброс битовых массивов I/O | (Reset I/O bit field) |

События прерываний

| SFC | Имя | Назначение | |
|-----|----------|---------------------------------------|----------------------------------|
| 28 | SET_TINT | Установка прерывания по времени суток | (Set time-of-day interrupt) |
| 29 | CAN_TINT | Отмена прерывания по времени суток | (Cancel time-of-day interrupt) |
| 30 | ACT_TINT | Активация прерывания по времени суток | (Activate time-of day interrupt) |
| 31 | QRY_TINT | Запрос прерывания по времени суток | (Query time-of-day interrupt) |

События прерываний (продолжение)

| SFC | Имя | Назначение | |
|-----|----------|--|--------------------------------|
| 32 | SRT_DINT | Запуск прерывания с задержкой обработки | (Start time-delay interrupt) |
| 33 | CAN_DINT | Отмена прерывания с задержкой обработки | (Cancel time-delay interrupt) |
| 34 | QRY_DINT | Запрос прерывания с задержкой обработки | (Query time-delay interrupt) |
| 35 | MP_ALM | Запуск предупреждения для мультипроцессорного режима | (Trigger multiprocessor alarm) |
| 36 | MSK_FLT | Маскирование синхронных ошибок | (Mask synchronous errors) |
| 37 | DMSK_FLT | Демаскирование синхронных ошибок | (Unmask synchronous errors) |
| 38 | READ_ERR | Считывание регистра состояния событий | (Read event status register) |
| 39 | DIR_IRT | Блокировка обработки асинхронных ошибок | (Disable asynchronous errors) |
| 40 | EN_IRT | Разблокировка обработки асинхронных ошибок | (Enable asynchronous errors) |
| 41 | DIS_AIRT | Задержка обработки асинхронных ошибок | (Delay asynchronous errors) |
| 42 | EN_AIRT | Разрешение обработки асинхронных ошибок | (Enable asynchronous errors) |

Коммуникации посредством неконфигурированных соединений

| SFC | Имя | Назначение | |
|-----|---------|---|-----------------------------|
| 65 | X_SEND | Передача данных партнеру по связи вне локальной станции | (Send data externally) |
| 66 | X_RCV | Прием данных от партнера по связи вне локальной станции | (Receive data externally) |
| 67 | X_GET | Чтение данных от партнера по связи вне локальной станции | (Read data externally) |
| 68 | X_PUT | Запись данных в партнере по связи вне локальной станции | (Write data externally) |
| 69 | X_ABORT | Отмена связи вне локальной станции | (Abort external connection) |
| 72 | I_GET | Чтение данных от партнера по связи в пределах локальной станции | (Read data internally) |
| 73 | I_PUT | Запись данных в партнере по связи в пределах локальной станции | (Write data internally) |
| 74 | I_ABORT | Отмена связи в пределах локальной станции | (Abort internal connection) |

Коммуникации посредством глобальных данных

| SFC | Имя | Назначение | |
|-----|--------|-------------------|---------------------|
| 60 | GD_SND | Посылка GD-пакета | (Send GD packet) |
| 61 | GD_RCV | Прием GD-пакета | (Receive GD packet) |

H CPU

| SFC | Имя | Назначение | |
|-----|--------|----------------------------------|------------------------------------|
| 90 | H_CTRL | Управление режимами работы H-CPU | (Control Operating Modes on H-CPU) |

33.3 Функциональные ИЕС-блоки**Функции сравнения**

| FC | Имя | Назначение | |
|----|----------|--|---|
| 9 | EQ_DT | Проверка данных формата DT на равенство | (Compare DT for equal to) |
| 28 | NE_DT | Проверка данных формата DT на неравенство | (Compare DT for not equal to) |
| 14 | GT_DT | Сравнение данных формата DT по критерию "больше чем" | (Compare DT for greater than) |
| 12 | GE_DT | Сравнение данных формата DT по критерию "больше или равно" | (Compare DT for greater than or equal to) |
| 23 | LT_DT | Сравнение данных формата DT по критерию "меньше чем" | (Compare DT for less than) |
| 18 | LE_DT | Сравнение данных формата DT по критерию "меньше или равно" | (Compare DT for less than or equal to) |
| 10 | EQ_STRNG | Проверка данных формата STRING на равенство | (Compare STRING for equal to) |
| 29 | NE_STRNG | Проверка данных формата STRING на неравенство | (Compare STRING for not equal to) |
| 15 | GT_STRNG | Сравнение данных формата STRING по критерию "больше чем" | (Compare STRING for greater than) |
| 13 | GE_STRNG | Сравнение данных формата STRING по критерию "больше или равно" | (Compare STRING for greater than or equal to) |
| 24 | LT_STRNG | Сравнение данных формата STRING по критерию "меньше чем" | (Compare STRING for less than) |
| 19 | LE_STRNG | Сравнение данных формата STRING по критерию "меньше или равно" | (Compare STRING for less than or equal to) |

Функции даты и времени

| FC | Имя | Назначение | |
|----|----------|---|-----------------------------------|
| 3 | D_TOD_DT | Объединение данных типов DATE и TOD в значение формата DT | (Combine DATE and TOD to DT) |
| 6 | DT_DATE | Извлечение данных формата DATE из данных формата DT | (Extract DATE from DT) |
| 7 | DT_DAY | Извлечение данных о дне недели из данных формата DT | (Extract day-of-the-week from DT) |
| 8 | DT_TOD | Извлечение данных о времени суток из данных формата DT | (Extract TOD from DT) |
| 33 | S5TI_TIM | Конвертация данных формата S5TIME в значение формата TIME | (Convert S5TIME to TIME) |
| 40 | TIM_S5TI | Конвертация данных формата TIME в значение формата S5TIME | (Convert TIME to S5TIME) |
| 1 | AD_DT_TM | Прибавление значения формата TIME к значению формата DT | (Add TIME to DT) |
| 35 | SB_DT_TM | Вычитание значения формата TIME из значения формата DT | (Subtract TIME from DT) |
| 34 | SB_DT_DT | Вычитание значения формата DT из значения формата DT | (Subtract DT from DT) |

Математические функции

| FC | Имя | Назначение | |
|----|-------|---|---------------------|
| 22 | LIMIT | Функция-ограничитель | (Limiter) |
| 25 | MAX | Выбор максимального значения из трех значений | (Maximum selection) |
| 27 | MIN | Выбор минимального значения из трех значений | (Minimum selection) |
| 26 | SEL | Функция "двоичный переключатель" | (Binary selection) |

Функции для обработки данных типа STRING

| FC | Имя | Назначение | |
|----|---------|----------------------------------|------------------------------|
| 21 | LEN | Возврат длины строки символов | (Length of a STRING) |
| 20 | LEFT | Возврат части строки слева | (Left section of a STRING) |
| 32 | RIGHT | Возврат части строки справа | (Right section of a STRING) |
| 26 | MID | Возврат части строки из середины | (Middle section of a STRING) |
| 2 | CONCAT | "Сшивание" двух строк символов | (Concatenate STRINGS) |
| 17 | INSERT | Вставка строки символов | (Insert STRING) |
| 4 | DELETE | Удаление части строки символов | (Delete STRING) |
| 31 | REPLACE | Замена части строки символов | (Replace STRING) |

Функции для обработки данных типа STRING (продолжение)

| FC | Имя | Назначение | |
|----|----------|---|--------------------------|
| 11 | FIND | Поиск строки символов | (Find STRING) |
| 16 | I_STRNG | Конвертация данных формата INT в формат STRING | (Convert INT to STRING) |
| 5 | DI_STRNG | Конвертация данных формата DINT в формат STRING | (Convert DINT to STRING) |
| 30 | R_STRNG | Конвертация данных формата REAL в формат STRING | (Convert REAL to STRING) |
| 38 | STRNG_I | Конвертация данных формата STRING в формат INT | (Convert STRING to INT) |
| 37 | STRNG_DI | Конвертация данных формата STRING в формат DINT | (Convert STRING to DINT) |
| 39 | STRNG_R | Конвертация данных формата STRING в формат REAL | (Convert STRING to REAL) |

33.4 Блоки для S5-S7-преобразования

Арифметические операции для чисел в формате с плавающей запятой

| FC | Имя | Назначение | |
|----|---------|--|---|
| 61 | GP_FPGP | Конвертация данных из формата с фиксированной запятой в формат с плавающей запятой | (Convert fixed-point to floating-point) |
| 62 | GP_GPFP | Конвертация данных из формата с плавающей запятой в формат с фиксированной запятой | (Convert floating-point to fixed-point) |
| 63 | GP_ADD | Сложение чисел в формате с плавающей запятой | (Add floating-point numbers) |
| 64 | GP_SUB | Вычитание чисел в формате с плавающей запятой | (Subtract floating-point numbers) |
| 65 | GP_MUL | Перемножение чисел в формате с плавающей запятой | (Multiply floating-point numbers) |
| 66 | GP_DIV | Деление чисел в формате с плавающей запятой | (Divide floating-point numbers) |
| 67 | GP_VGL | Сравнение чисел в формате с плавающей запятой | (Compare floating-point numbers) |
| 68 | GP_RAD | Извлечение квадратного корня из числа в формате с плавающей запятой | (Find the square root of a floating-point number) |

Базовые функции

| FC | Имя | Назначение |
|----|----------|--|
| 85 | ADD_32 | Сложение 32-разрядных чисел в формате с фиксированной запятой (32-bit fixed-point adder) |
| 86 | SUB_32 | Вычитание 32-разрядных чисел в формате с фиксированной запятой (32-bit fixed-point subtractor) |
| 87 | MUL_32 | Перемножение 32-разрядных чисел в формате с фиксированной запятой (32 bit fixed-point multiplier) |
| 88 | DIV_32 | Деление 32-разрядных чисел в формате с фиксированной запятой (32-bit fixed-point divider) |
| 89 | RAD_16 | Извлечение квадратного корня из 16-разрядного числа в формате с фиксированной запятой (16-bit fixed point square root extractor) |
| 90 | REG_SCHB | Побитовый сдвиг в регистре (Bitwise shift register) |
| 91 | REG_SCHW | Пословный сдвиг в регистре (Wordwise shift register) |
| 92 | REG_FIFO | Буфер FIFO (Buffer (FIFO)) |
| 93 | REG_LIFO | Стек LIFO (Stack (LIFO)) |
| 94 | DB_COPY1 | Прямое копирование данных из области памяти (Copy data area (direct)) |
| 95 | DB_COPY2 | Косвенное копирование данных из области памяти (Copy data area (indirect)) |
| 96 | RETTEN | Сохранение содержимого "сверхоперативной" памяти (S5-155U) (Save scratchpad memory (S5-155U)) |
| 97 | LADEN | Загрузка содержимого "сверхоперативной" памяти (S5-155U) (Load scratchpad memory (S5-155U)) |
| 98 | COD_B8 | Конвертация данных из формата BCD в двоичный формат (для 8 декад) (BCD-binary conversion 8 decades) |
| 99 | COD_32 | Конвертация данных из двоичного формата в формат BCD (для 8 декад) (Binary-BCD conversion 8 decades) |

Функции для обработки сигнала

| FC | Имя | Назначение | |
|----|----------|--|---|
| 69 | MLD_TG | Генератор тактовых импульсов | (Clock pulse generator) |
| 70 | MLD_TGZ | Генератор тактовых импульсов с функцией таймера | (Clock pulse generator with timer function) |
| 71 | MLD_EZW | Начальное значение параметра одинарного сигнала (слово) | (Initial value single blinking wordwise) |
| 72 | MLD_EDW | Начальное значение параметра двойного сигнала (слово) | (Initial value double blinking wordwise) |
| 73 | MLD_SAMW | Групповой сигнал (слово) | (Group signal wordwise) |
| 74 | MLD_SAM | Групповой сигнал | (Group signal) |
| 75 | MLD_EZ | Начальное значение параметра одинарного сигнала | (Initial value single blinking) |
| 76 | MLD_ED | Начальное значение параметра двойного сигнала | (Initial value double blinking) |
| 77 | MLD_EZWK | Начальное значение параметра одинарного сигнала (слово меркеров) | (Initial value single blinking (wordwise) memory bit) |
| 78 | MLD_EZDK | Начальное значение параметра двойного сигнала (слово меркеров) | (Initial value double blinking (wordwise) memory bit) |
| 79 | MLD_EZK | Начальное значение параметра одинарного сигнала меркера | (Initial value single blinking memory bit) |
| 80 | MLD_EDK | Начальное значение параметра двойного сигнала меркера | (Initial value double blinking memory bit) |

Встроенные функции

| FC | Имя | Назначение | |
|----|--------|--|-----------------------------------|
| 81 | COD_B4 | Конвертация данных из формата BCD в двоичный формат (для 4 декад) | (BCD-binary conversion 4 decades) |
| 82 | COD_16 | Конвертация данных из двоичного формата в формат BCD (для 4 декад) | (Binary-BCD conversion 4 decades) |
| 83 | MUL_16 | Перемножение 16-разрядных чисел в формате с фиксированной запятой | (16-bit fixed-point multiplier) |
| 84 | DIV_16 | Деление 16-разрядных чисел в формате с фиксированной запятой | (16-bit fixed-point divider) |

Аналоговые функции

| FC | Имя | Назначение |
|-----|----------|---|
| 100 | AE_460_1 | Аналоговый входной модуль 460 (Analog input module 460) |
| 101 | AR_460_2 | Аналоговый входной модуль 460 (Analog input module 460) |
| 102 | AR_463_1 | Аналоговый входной модуль 463 (Analog input module 463) |
| 103 | AE_463_2 | Аналоговый входной модуль 463 (Analog input module 463) |
| 104 | AE_464_1 | Аналоговый входной модуль 464 (Analog input module 464) |
| 105 | AE_464_2 | Аналоговый входной модуль 464 (Analog input module 464) |
| 106 | AE_466_1 | Аналоговый входной модуль 466 (Analog input module 466) |
| 107 | AE_466_2 | Аналоговый входной модуль 466 (Analog input module 466) |
| 108 | RLG_AA1 | Аналоговый выходной модуль (Analog output module) |
| 109 | RLG_AA2 | Аналоговый выходной модуль (Analog output module) |
| 110 | PER_ET1 | Распределенные I/O ET 100 (ET 100 distributed I/O) |
| 111 | PER_ET2 | Распределенные I/O ET 100 (ET 100 distributed I/O) |

Математические функции

| FC | Имя | Назначение |
|-----|----------|--|
| 112 | SINUS | Синус (Sine) |
| 113 | COSINUS | Косинус (Cosine) |
| 114 | TANGENS | Тангенс (Tangent) |
| 115 | COTANG | Котангенс (Cotangent) |
| 116 | ARCSIN | Арксинус (Arc sine) |
| 117 | ARCCOS | Арккосинус (Arc cosine) |
| 118 | ARCTAN | Арктангенс (Arc tangent) |
| 119 | ARCCOT | Арккотангенс (Arc cotangent) |
| 120 | LN_X | Натуральный логарифм (Natural logarithm) |
| 121 | LG_X | Десятичный логарифм (Logarithm to base 10) |
| 122 | B_LOG_X | Логарифм с любым основанием (Logarithm to any base) |
| 123 | E_H_N | Экспоненциальная функция с основанием e (Exponential function with base e) |
| 124 | ZEHN_H_N | Экспоненциальная функция с основанием 10 (Exponential function with base 10) |
| 125 | A2_H_A1 | Экспоненциальная функция с любым основанием (Exponential function with any base) |

33.5 Блоки для TI-S7-преобразования

| FB | Имя | Назначение | |
|----|----------|---|-----------------------------------|
| 80 | LEAD_LAG | Алгоритм "опережение/отставание" | (Lead/lag algorithm) |
| 81 | DCAT | Дискретное управление прерыванием по времени | (Dircrete control time interrupt) |
| 82 | MCAT | Прерывание по времени для управления двигателем | (Motor control time interrupt) |
| 83 | IMC | Сравнение индексов матрицы | (Index matrix comparison) |
| 84 | SMC | Сканирование матрицы | (Matrix scanner) |
| 85 | DRUM | Маскирование последовательности событий | (Event maskable drum) |
| 86 | PACK | Сбор/распределение табличных данных | (Collect/distribute table data) |

| FC | Имя | Назначение | |
|----|----------|--|-------------------------------------|
| 80 | TONR | Фиксация задержки | (Latching ON delay) |
| 81 | IBLKMOV | Прямая передача данных области памяти | (Transfer data area indirectly) |
| 82 | RSET | Побитовый сброс образа процесса | (Reset process image bit by bit) |
| 83 | SET | Побитовая установка образа процесса | (Set process image bit by bit) |
| 84 | ATT | Входное значение в таблице | (Enter value in table) |
| 85 | FIFO | Выходное первое значение в таблице | (Output first value in table) |
| 86 | TBL_FIND | Нахождение значения в таблице | (Find value in table) |
| 87 | LIFO | Выходное последнее значение в таблице | (Output last value in table) |
| 88 | TBL | Выполнение обработки таблицы | (Execute table operation) |
| 89 | TBL_WRD | Копирование значения из таблицы | (Copy value from the table) |
| 90 | WSR | Сохранение данных | (Save datum) |
| 91 | WRD_TBL | Совместная обработка элементов таблицы | (Combine table element) |
| 92 | SHRB | Побитовый сдвиг в регистре сдвига | (Shift bit in bit shift register) |
| 93 | SEG | Распределение битов для 7-сегментного индикатора | (Bit pattern for 7-segment display) |

| FC | Имя | Назначение | |
|-----|---------|---|--------------------------------|
| 94 | ATH | Конвертация данных из формата ASCII в шестнадцатеричное представление | (ASCII hexadecimal conversion) |
| 95 | HTA | Конвертация данных из шестнадцатеричного формата в ASCII-формат | (Hexadecimal-ASCII conversion) |
| 96 | ENCO | Установка младшего бита | (Least significant set bit) |
| 97 | DECO | Установка бита в слове | (Set bit in word) |
| 98 | BCDCPL | Нахождение обратного BCD-числа | (Generate ten's complement) |
| 99 | BITSUM | Подсчет установленных битов | (Count set bits) |
| 100 | RSETI | Побайтовый сброс отображения выходов PQ процесса | (Reset PQ byte by byte) |
| 101 | SETI | Побайтовая установка образа выходов PQ процесса | (Set PQ byte by byte) |
| 102 | DEV | Вычисление стандартного отклонения | (Calculate standard deviation) |
| 103 | CDT | Таблица связанных данных | (Correlated data tables) |
| 104 | TBL_TBL | Объединение таблиц | (Table combination) |
| 105 | SCALE | Коэффициенты масштабирования | (Scale values) |
| 106 | UNSCALE | Обратные коэффициенты | (Unscale values) |

33.6 Блоки ПИД-управления

| FB | Имя | Назначение | |
|----|---------|------------------------|----------------------|
| 41 | CONT_C | Непрерывное управление | (Continuous control) |
| 42 | CONT_S | Пошаговое управление | (Step control) |
| 43 | PULSGEN | Генерация импульса | (Generate pulse) |

33.7 Коммуникационные блоки

| FC | Имя | Назначение | |
|----|---------|----------------|----------------|
| 1 | DP_SEND | Посылка данных | (Send data) |
| 2 | DP_RECV | Прием данных | (Receive data) |
| 3 | DP_DIAG | Диагностика | (Diagnostics) |
| 4 | DP_CTRL | Управление | (Control) |

34 Общий обзор STL-инструкций

В общем обзоре, предлагаемом ниже, представлен список инструкций с абсолютными адресами.

Возможны следующие типы адресации:

| | | |
|--------------------|--|--|
| A I [doubleword] | косвенная адресация посредством памяти MD двойное слово меркеров LD двойное слово локальных данных DBD двойное слово глобальных данных DID двойное слово в экземплярном DB | все адреса |
| A I [AR1,P#offset] | внутризонная косвенная адресация посредством регистра AR1 | кроме функций таймеров, счетчиков и блоков |
| A I [AR2,P#offset] | внутризонная косвенная адресация посредством регистра AR2 | |
| A [AR1,P#offset] | межзонная косвенная адресация посредством регистра AR1 | |
| A [AR2,P#offset] | межзонная косвенная адресация посредством регистра AR2 | |
| A #name | косвенная адресация посредством параметра | все адреса |

34.1 Базовые функции

34.1.1 Двоичные логические операции

| | | |
|------|-----|---|
| A | - | операция И (AND) для проверки присутствия уровня "1" |
| AN | - | операция И (AND) для проверки присутствия уровня "0" |
| O | - | операция ИЛИ (OR) для проверки присутствия уровня "1" |
| ON | - | операция ИЛИ (OR) для проверки присутствия уровня "0" |
| X | - | операция Исключающее ИЛИ (Exclusive OR) для проверки присутствия уровня "1" |
| XN | - | операция Исключающее ИЛИ (Exclusive OR) для проверки присутствия уровня "0" |
| - | I | вход |
| - | Q | выход |
| - | M | меркер |
| - | L | бит в области локальных данных |
| - | T | функция таймера |
| - | C | функция счетчика |
| - | DBX | бит в области глобальных данных |
| - | DIX | бит в экземплярном DB |
| - | ==0 | значение результата операции равно нулю |
| - | <>0 | значение результата операции не равно нулю |
| - | >0 | значение результата операции больше нуля |
| - | >=0 | значение результата операции больше нуля или равно нулю |
| - | <0 | значение результата операции меньше нуля |
| - | <=0 | значение результата операции меньше нуля или равно нулю |
| - | UO | значение результата операции неверно |
| - | OV | переполнение |
| - | OS | сохраненное переполнение |
| - | BR | двоичный результат |
| A(| | операция И (AND) с открывающей скобкой |
| AN(| | операция И (AND) с открывающей скобкой |
| O(| | операция ИЛИ (OR) с открывающей скобкой |
| ON(| | операция ИЛИ (OR) с открывающей скобкой |
| X(| | операция Исключающее ИЛИ (Exclusive OR) с открывающей скобкой |
| XN(| | операция Исключающее ИЛИ (Exclusive OR) с открывающей скобкой |
|) | | закрывающая скобка |
| O | | операция ИЛИ (OR), объединяющая операции И (AND) |
| NOT | | операция отрицания RLO |
| SET | | операция установки RLO |
| CLR | | операция сброса RLO |
| SAVE | | операция фиксации RLO в BR |

34.1.2 Операции с памятью

| | | |
|----|-----|---------------------------------|
| = | - | операция присвоения |
| S | - | операция установки |
| R | - | операция сброса |
| FP | - | положительный фронт сигнала |
| FN | - | отрицательный фронт сигнала |
| - | I | вход |
| - | Q | выход |
| - | M | меркер |
| - | L | бит в области локальных данных |
| - | DBX | бит в области глобальных данных |
| - | DIX | бит в экземплярном DB |

34.1.3 Функции передачи

| | | |
|---|-----|---------------------------------|
| L | - | операция загрузки (load) |
| T | - | операция передачи (transfer) |
| - | IB | входной байт |
| - | IW | входное слово |
| - | ID | входное двойное слово |
| - | QB | выходной байт |
| - | QW | выходное слово |
| - | QD | выходное двойное слово |
| - | MB | байт меркеров |
| - | MW | слово меркеров |
| - | MD | двойное слово меркеров |
| - | LB | байт локальных данных |
| - | LW | слово локальных данных |
| - | LD | двойное слово локальных данных |
| - | DBB | байт глобальных данных |
| - | DBW | слово глобальных данных |
| - | DBD | двойное слово глобальных данных |
| - | DIB | байт в экземплярном DB |
| - | DIW | слово в экземплярном DB |
| - | DID | двойное слово в экземплярном DB |
| - | STW | слово состояния |

| | | |
|----|--------------|--|
| L | PIB | загрузка (load) периферийного входного байта |
| L | PIW | загрузка (load) периферийного входного слова |
| L | PID | загрузка (load) периферийного входного двойного слова |
| T | PQB | передача (transfer) периферийного выходного байта |
| T | PQW | передача (transfer) периферийного выходного слова |
| T | PQD | передача (transfer) периферийного выходного двойного слова |
| L | T | "обычная" загрузка значения таймера |
| LC | T | загрузка значения таймера в BCD-коде |
| L | C | "обычная" загрузка значения счетчика |
| LC | C | загрузка значения счетчика в BCD-коде |
| L | <i>const</i> | загрузка (load) константы |
| L | P#.. | загрузка (load) указателя |
| L | P#var | загрузка (load) начального адреса переменной |

Функции аккумуляторов

| | |
|-------|---|
| PUSH | сдвиг содержимого аккумуляторов "вперед" |
| POP | сдвиг содержимого аккумуляторов "назад" |
| ENT | сдвиг содержимого аккумуляторов 2 и 3 "вперед" |
| LEAVE | сдвиг содержимого аккумуляторов 3 и 4 "вперед" |
| TAK | обмен содержимым между аккумуляторами 1 и 2 |
| CAW | обмен содержимым между байтами 0 и 1 аккумулятора 1 |
| CAD | обмен содержимым между всеми байтами аккумулятора 1 |

34.1.4 Функции таймеров

| | | |
|----|---|---|
| SP | T | запуск таймера в режиме "управляемого импульса" |
| SE | T | запуск таймера в режиме "расширенного импульса" |
| SD | T | запуск таймера в режиме "с задержкой включения" |
| SS | T | запуск таймера в режиме "с задержкой включения с памятью" |
| SF | T | запуск таймера в режиме "с задержкой выключения" |
| R | T | сброс таймера |
| FR | T | разрешение перезапуска таймера |

34.1.5 Функции счетчиков

| | | |
|----|---|--|
| CU | C | запуск счетчика в режиме "прямой счет" |
| CD | C | запуск счетчика в режиме "обратный счет" |
| S | C | установка счетчика |
| R | C | сброс счетчика |
| FR | C | разрешение перезапуска счетчика |

34.2 Функции для обработки чисел

34.2.1 Функции сравнения

| | |
|-----|--|
| ==I | проверка данных формата INT на равенство |
| <>I | проверка данных формата INT на неравенство |
| >I | сравнение данных формата INT по критерию "больше чем" |
| >=I | сравнение данных формата INT по критерию "больше или равно" |
| <I | сравнение данных формата INT по критерию "меньше чем" |
| <=I | сравнение данных формата INT по критерию "меньше или равно" |
| ==D | проверка данных формата DINT на равенство |
| <>D | проверка данных формата DINT на неравенство |
| >D | сравнение данных формата DINT по критерию "больше чем" |
| >=D | сравнение данных формата DINT по критерию "больше или равно" |
| <D | сравнение данных формата DINT по критерию "меньше чем" |
| <=D | сравнение данных формата DINT по критерию "меньше или равно" |
| ==R | проверка данных формата REAL на равенство |
| <>R | проверка данных формата REAL на неравенство |
| >R | сравнение данных формата REAL по критерию "больше чем" |
| >=R | сравнение данных формата REAL по критерию "больше или равно" |
| <R | сравнение данных формата REAL по критерию "меньше чем" |
| <=R | сравнение данных формата REAL по критерию "меньше или равно" |

34.2.2 Математические функции

| | |
|-----|---------|
| SIN | синус |
| COS | косинус |
| TAN | тангенс |

| | |
|------|---------------------------------------|
| ASIN | арксинус |
| ACOS | арккосинус |
| ATAN | арктангенс |
| SQR | нахождение квадрата числа |
| SQRT | извлечение квадратного корня из числа |
| EXP | экспонента по основанию e |
| LN | натуральный логарифм |

34.2.3 Арифметические функции

| | | |
|-----|--------------|---|
| +I | | сложение двух чисел формата INT |
| -I | | вычитание двух чисел формата INT |
| *I | | умножение двух чисел формата INT |
| /I | | деление двух чисел формата INT |
| +D | | сложение двух чисел формата DINT |
| -D | | вычитание двух чисел формата DINT |
| *D | | умножение двух чисел формата DINT |
| /D | | деление двух чисел формата DINT (целая часть) |
| MOD | | деление двух чисел формата DINT (остаток) |
| +R | | сложение двух чисел формата REAL |
| -R | | вычитание двух чисел формата REAL |
| *R | | умножение двух чисел формата REAL |
| /R | | деление двух чисел формата REAL |
| + | <i>const</i> | сложение с константой |
| + | <i>P#..</i> | сложение с указателем |
| DEC | <i>n</i> | декрементирование |
| INC | <i>n</i> | инкрементирование |

34.2.4 Функции преобразования

| | |
|-----|---|
| ITD | конвертирование данных формата INT в формат DINT |
| ITB | конвертирование данных формата INT в формат BCD |
| DTB | конвертирование данных формата DINT в формат DINT |
| DTR | конвертирование данных формата DINT в формат REAL |

| | |
|-------|--|
| BTI | конвертирование данных формата BCD в формат INT |
| BTD | конвертирование данных формата BCD в формат DINT |
| | Конвертирование данных формата REAL в формат DINT, при этом происходит: |
| RND+ | округление данных до ближайшего большего целого числа |
| RND- | округление данных до ближайшего меньшего целого числа |
| RND | округление данных до ближайшего целого числа |
| TRUNC | усечение дробной части числа |
| INVI | нахождение обратного кода двоичного числа формата INT |
| INVD | нахождение обратного кода двоичного числа формата DINT |
| NEGI | инвертирование числа формата INT |
| NEGD | инвертирование числа формата DINT |
| NEGR | инвертирование числа формата REAL |
| ABS | нахождение абсолютного значения числа формата REAL |

34.2.5 Функции сдвига

| | |
|------|---|
| SLW | - побитовый сдвиг влево содержимого младшего слова аккумулятора 1 |
| SLD | - побитовый сдвиг влево содержимого аккумулятора 1 |
| SRW | - побитовый сдвиг вправо содержимого младшего слова аккумулятора 1 |
| SRD | - побитовый сдвиг вправо содержимого аккумулятора 1 |
| SSI | - побитовый сдвиг со знаком содержимого младшего слова аккумулятора 1 |
| SSD | - побитовый сдвиг со знаком содержимого аккумулятора 1 |
| RLD | - циклический сдвиг влево содержимого аккумулятора 1 |
| RRD | - циклический сдвиг вправо содержимого аккумулятора 1 |
| - | <i>n</i> на <i>n</i> позиций |
| - | на число позиций, указанное в аккумуляторе Accum 2 |
| RLDA | - циклический сдвиг влево с использованием бита CC1 |
| RRDA | - циклический сдвиг вправо с использованием бита CC1 |

34.2.6 Логические функции для слов данных

| | |
|----|--|
| AW | - операция И (AND) для слова данных |
| AD | - операция И (AND) для двойного слова данных |

| | | |
|-----|--------------|---|
| OW | - | операция ИЛИ (OR) для слова данных |
| OD | - | операция ИЛИ (OR) для двойного слова данных |
| XOW | - | операция Исключающее ИЛИ (Exclusive OR) для слова данных |
| XOD | - | операция Исключающее ИЛИ (Exclusive OR) для двойного слова данных |
| - | <i>const</i> | с константой формата слова данных или двойного слова данных |
| - | | с содержимым аккумулятора Accum 2 |

34.3 Функции управления в программе

34.3.1 Функции перехода

| | | |
|------|--------------|--|
| JU | <i>метка</i> | безусловный переход |
| | | Выполняется переход, |
| JC | <i>метка</i> | если RLO = "1" |
| JCB | <i>метка</i> | если RLO = "1" с сохранением RLO |
| JCN | <i>метка</i> | если RLO = "0" |
| JNB | <i>метка</i> | если RLO = "0" с сохранением RLO |
| JBI | <i>метка</i> | если BR = "1" |
| JNBI | <i>метка</i> | если BR = "0" |
| | | Выполняется переход, |
| JZ | <i>метка</i> | если результат = "0" |
| JN | <i>метка</i> | если результат <> "0" |
| JP | <i>метка</i> | если результат > "0" |
| JPZ | <i>метка</i> | если результат >= "0" |
| JM | <i>метка</i> | если результат < "0" |
| JMZ | <i>метка</i> | если результат <= "0" |
| JUO | <i>метка</i> | если результат некорректен |
| JO | <i>метка</i> | переход выполняется при переполнении |
| JOS | <i>метка</i> | переход выполняется при запомненном переполнении |
| JL | <i>метка</i> | распределитель переходов |
| LOOP | <i>метка</i> | циклический переход |

34.3.2 Главное управляющее реле MCR

| | |
|------|-------------------------|
| MCRA | активация области MCR |
| MCRD | деактивация области MCR |
| MCR(| открытие зоны MCR |
|)MCR | закрытие зоны MCR |

34.3.3 Функции обработки блоков

| | | |
|------|----------|--|
| CALL | FB | вызов функционального блока |
| CALL | FC | вызов функции |
| CALL | SFB | вызов системного функционального блока |
| CALL | SFC | вызов системной функции |
| UC | FB | безусловный вызов функционального блока |
| CC | FB | вызов функционального блока по условию |
| UC | FC | безусловный вызов функции |
| CC | FC | вызов функции по условию |
| BEU | | безусловное завершение обработки блока |
| BEC | | завершение обработки блока по условию |
| BE | | безусловное завершение обработки блока |
| OPN | DB | вызов глобального блока данных |
| OPN | DI | вызов экземплярного блока данных |
| CDB | | обмен данными между регистрами блока |
| L | DBNO | загрузка (load) номера глобального блока данных |
| L | DINO | загрузка (load) номера экземплярного блока данных |
| L | DBLG | загрузка (load) размера глобального блока данных |
| L | DILG | загрузка (load) размера экземплярного блока данных |
| NOP | 0 | нуль-операция |
| NOP | 1 | нуль-операция |
| BLD | <i>n</i> | инструкции отображения программы |

34.4 Косвенная адресация

| | | |
|------|-------|---|
| LAR1 | - | загрузка в адресный регистр AR1 из |
| LAR2 | - | загрузка в адресный регистр AR2 из |
| - | MD | двойного слова меркеров |
| - | LD | двойного слова локальных данных |
| - | DBD | двойного слова глобальных данных |
| - | DID | двойного слова экземплярных данных |
| LAR1 | | загрузка в AR1 содержимого аккумулятора Ассум 1 |
| LAR2 | | загрузка в AR2 содержимого аккумулятора Ассум 1 |
| LAR1 | AR2 | загрузка в AR1 содержимого регистра AR2 |
| LAR1 | P#.. | загрузка в AR1 указателя |
| LAR2 | P#.. | загрузка в AR2 указателя |
| LAR1 | P#var | загрузка в AR1 начального адреса переменной |
| LAR2 | P#var | загрузка в AR2 начального адреса переменной |
| TAR1 | - | передача из адресного регистра AR1 |
| TAR2 | - | передача из адресного регистра AR2 |
| - | MD | в двойное слово меркеров |
| - | LD | в двойное слово локальных данных |
| - | DBD | в двойное слово глобальных данных |
| - | DID | в двойное слово экземплярных данных |
| TAR1 | - | передача из адресного регистра AR1 в аккумулятор Ассум 1 |
| TAR2 | - | передача из адресного регистра AR2 в аккумулятор Ассум 1 |
| TAR1 | AR2 | передача из адресного регистра AR1 в регистр AR2 |
| CAR | | обмен данными между регистрами AR1 и AR2 |
| +AR1 | | прибавить содержимое аккумулятора Ассум 1 к содержимому AR1 |
| +AR2 | | прибавить содержимое аккумулятора Ассум 1 к содержимому AR2 |
| +AR1 | P#.. | прибавить указатель к содержимому AR1 |
| +AR2 | P#.. | прибавить указатель к содержимому AR2 |

35 Общий обзор SCL-инструкций и функций

35.1 Операторы

| Тип оператора | Наименование | Оператор | Приоритет |
|-----------------|--|--------------|-----------|
| Скобки | (Выражение) | (,) | 1 |
| Арифметические | Степень | ** | 2 |
| | Унарный плюс, унарный минус (знак) | +, - | 3 |
| | Умножение, деление | *,/,DIV,MOD | 4 |
| | Сложение, вычитание | +, - | 5 |
| Сравнения | Меньше, меньше или равно, больше, больше или равно | <, <=, >, >= | 6 |
| | Равно, неравно | =, <> | 7 |
| Двоичная логика | Отрицание (унарная операция) | NOT | 3 |
| | Логическая операция И (AND) | AND, & | 8 |
| | Логическая операция Исключающее ИЛИ (Exclusive OR) | XOR | 9 |
| | Логическая операция ИЛИ (OR) | OR | 10 |
| Присвоение | Присвоение | := | 11 |

35.2 Управляющие операторы

| | |
|----------|---|
| IF | Ветвление программы, обусловленное значением булевой переменной |
| CASE | Ветвление программы, обусловленное значением целой переменной (INT) |
| FOR | Организация цикла в программе с использованием переменной "счетчик цикла" |
| WHILE | Организация цикла в программе с проверкой условия выполнения цикла |
| REPEAT | Организация цикла в программе с проверкой условия завершения цикла |
| CONTINUE | Прервать текущее выполнение (проход) программы цикла |
| EXIT | Выход из программы цикла |
| GOTO | Переход к метке перехода |
| RETURN | Выход из блока (прекращение выполнения блока) |

35.3 Вызов блоков

| | |
|---|---|
| Функции FC с функциональным значением | <i>Variable</i> := FCx(...); <i>Variable</i> := FCname(...); |
| Системные функции SFC с функциональным значением | <i>Variable</i> := SFCx(...); <i>Variable</i> := SFCname(...); |
| Функции FC без функционального значения | FCx(...); FCname(...); |
| Функциональные блоки FB с блоками данных | FBx.DBx(...); FBname.DBname(...); |
| Системные функциональные блоки SFB с блоками данных | SFBx.DBx(...); SFBname.DBname(...); |
| Функциональные блоки FB и системные функциональные блоки SFB как локальные экземпляры | <i>localname</i> (...); |

(Идентификаторы: *Variable* = идентификатор переменной,
name = идентификатор блока (функции))

Инициализация параметров блока обязательна для блоков FC и SFC и необязательна для блоков SFB.

35.4 Стандартные функции CSL

35.4.1 Функции таймеров

| Вызов | Тип данных | Комментарий |
|--|---|--|
| <pre>Time_BCD := Timer_function(T_NO = Timer_address, S = Start_input, TV = Timer_duration, R = Reset, Q = Timer_status, BI = Binary_time);</pre> | <p>WORD (см. ниже)</p> <p>TIMER</p> <p>BOOL</p> <p>S5TIME</p> <p>BOOL</p> <p>BOOL</p> <p>WORD</p> | <p>адрес таймера</p> <p>вход запуска</p> <p>длительность</p> <p>сброс</p> <p>состояние таймера</p> <p>время в двоичном формате</p> |

В качестве функции `Timer_function` могут быть использованы следующие функции, обеспечивающие работу таймеров в следующих рабочих режимах:

| | |
|---------|---|
| S_PULSE | Режим управляемого импульса (Pulse time) |
| S_PEXT | Режим расширенного импульса (Extended pulse) |
| S_ODT | Режим с задержкой включения (ON delay) |
| S_ODTS | Режим с задержкой включения с памятью (Latching OFF delay) |
| S_OFFDT | Режим с задержкой выключения (Off delay) |

35.4.2 Функции счетчиков

| Вызов в режиме прямого счета | Тип данных | Комментарий |
|---|--|--|
| <pre>BCD count value := S_CU(C_NO := Count_address, CU := Count_up, S := Set_input, PV := Count_value, R := Reset, Q := Counter_status, CV := Bin_count_val);</pre> | <p>WORD</p> <p>COUNTER</p> <p>BOOL</p> <p>BOOL</p> <p>WORD</p> <p>BOOL</p> <p>BOOL</p> <p>WORD</p> | <p>адрес счетчика</p> <p>режим прямого счета</p> <p>вход установки счетчика</p> <p>значение счетчика</p> <p>сброс</p> <p>состояние счетчика</p> <p>значение счетчика в двоичном формате</p> |
| Вызов в режиме обратного счета | | |
| <pre>BCD count value := S_CD(C_NO := Count_address, CD := Count_down, S := Set_input, PV := Count_value, R := Reset, Q := Counter_status, CV := Bin_count_val);</pre> | <p>WORD</p> <p>COUNTER</p> <p>BOOL</p> <p>BOOL</p> <p>WORD</p> <p>BOOL</p> <p>BOOL</p> <p>WORD</p> | <p>адрес счетчика</p> <p>режим обратного счета</p> <p>вход установки счетчика</p> <p>значение счетчика</p> <p>сброс</p> <p>состояние счетчика</p> <p>значение счетчика в двоичном формате</p> |
| Вызов в режиме прямого/обратного счета | | |
| <pre>BCD count value := S_CU(C_NO := Count_address, CU := Count_up, CD := Count_down, S := Set_input, PV := Count_value, R := Reset, Q := Counter_status, CV := Bin_count_val);</pre> | <p>WORD</p> <p>COUNTER</p> <p>BOOL</p> <p>BOOL</p> <p>BOOL</p> <p>WORD</p> <p>BOOL</p> <p>BOOL</p> <p>WORD</p> | <p>адрес счетчика</p> <p>режим прямого счета</p> <p>режим обратного счета</p> <p>вход установки счетчика</p> <p>значение счетчика</p> <p>сброс</p> <p>состояние счетчика</p> <p>значение счетчика в двоичном формате</p> |

35.4.3 Функции преобразования

Неявные функции преобразования

| Преобразование: тип исходный - тип целевой | Примечание |
|---|-------------------------------|
| BOOL_TO_BYTE BOOL_TO_WORD BOOL_TO_DWORD BYTE_TO_WORD BYTE_TO_DWORD WORD_TO_DWORD | заполнение лидирующими нулями |
| INT_TO_DINT INT_TO_REAL DINT_TO_REAL | заполнение знаком |
| CHAR_TO_STRING | |

Явные функции преобразования

| Преобразование: тип исходный - тип целевой | Примечание |
|---|--|
| BYTE_TO_BOOL WORD_TO_BOOL DWORD_TO_BOOL WORD_TO_BYTE DWORD_TO_BYTE DWORD_TO_WORD | заполняются значением младшие бит, байт, слово |
| CHAR_TO_BYTE BYTE_TO_CHAR CHAR_TO_INT INT_TO_CHAR | без изменения назначения битов |
| STRING_TO_CHAR | |

Явные функции преобразования (продолжение)

| Преобразование: тип исходный - тип целевой | Примечание |
|--|--|
| WORD_TO_INT DWORD_TO_DINT INT_TO_WORD DINT_TO_DWORD REAL_TO_DWORD DWORD_TO_REAL | без изменения назначения битов (без конвертирования) |
| DINT_TO_INT REAL_TO_DINT REAL_TO_INT | с округлением до целого INT или DINT |
| TRUNC ROUND | преобразование данных формата REAL в DINT |
| DINT_TO_TIME DINT_TO_TOD DINT_TO_DATE DATE_TO_DINT TIME_TO_DINT TOD_TO_DINT | без изменения назначения битов |
| BLOCK_DB_TO_WORD WORD_TO_BLOCK_DB | без изменения назначения битов |

35.4.4 Математические функции

| Вызов | Тип данных |
|---|-------------------------------|
| <pre>Result := Math_function(Input_value);</pre> | REAL (см. ниже) ANY_NUM |

(Идентификаторы: Result = идентификатор переменной,
Math_function = математическая функция,
Input_value = входное значение)

В качестве функции `Math_function` могут быть использованы следующие математические функции:

| | |
|------|---------------------------------------|
| SIN | синус |
| COS | косинус |
| TAN | тангенс |
| ASIN | арксинус |
| ACOS | арккосинус |
| ATAN | арктангенс |
| EXP | экспонента по основанию e |
| EXPD | экспонента по основанию 10 |
| LN | натуральный логарифм |
| LOG | десятичный логарифм |
| SQR | нахождение квадрата числа |
| SQRT | извлечение квадратного корня из числа |

ABS функция нахождения абсолютного значения числа:

| Вызов функции ABS | Тип данных |
|---------------------------------|------------|
| <code>Result :=</code> | ANY_NUM |
| <code>ABS (Input_value);</code> | ANY_NUM |

(Идентификаторы: `Result` = идентификатор переменной,
`Input_value` = входное значение)

35.4.5 Функции сдвига и циклического сдвига

| Вызов | Тип данных |
|-----------------------------------|------------|
| <code>Result :=</code> | ANY_BIT |
| <code>Shift_function(</code> | (см. ниже) |
| <code>IN := Input_value,</code> | ANY_NUM |
| <code>N := Num_of_places);</code> | INT |

(Идентификаторы: `Result` = идентификатор переменной,
`Shift_function` = функция сдвига,
`Input_value` = входное значение,
`Num_of_places` = число позиций сдвига)

В качестве функции `Shift_function` могут быть использованы следующие функции:

| | |
|-----|--------------------------|
| SHL | сдвиг влево |
| SHR | сдвиг вправо |
| ROL | циклический сдвиг влево |
| ROR | циклический сдвиг вправо |