

Установление связи и актуализация

6

| В разделе | Вы найдете | на стр. |
|-----------|---|---------|
| 6.1 | Воздействия при установлении связи и актуализации | 6–2 |
| 6.2 | Процесс установления связи и актуализации | 6–3 |
| 6.3 | Контроль времени | 6–15 |
| 6.4 | Особенности при установлении связи и актуализации | 6–28 |

6.1 Воздействия при установлении связи и актуализации

Установление связи и актуализация отображаются светодиодами REDF на обоих CPU. При установлении связи эти светодиоды мигают с частотой 0,5 Гц, а при актуализации с частотой 2 Гц.

Установление связи и актуализация оказывают различные воздействия на выполнение программы пользователя и коммуникационные функции.

Таблица 6–1. Свойства связи и актуализации

| Процесс | Установление связи | Актуализация |
|---|---|--|
| Выполнение программы пользователя | Обрабатываются все классы приоритета (OB). | Обработка классов приоритета задерживается частями. Выполнение всех требований восполняется после актуализации. Подробности вы найдете в следующих разделах. |
| Удаление, загрузка, создание и сжатие блоков | Блоки не могут удаляться, загружаться, создаваться и сжиматься. Если эти действия как раз выполняются, то установление связи и актуализация невозможны. | Блоки не могут удаляться, загружаться, создаваться и сжиматься. |
| Выполнение коммуникационных функций, управление с помощью PG | Коммуникационные функции выполняются. | Выполнение этих функций ограничивается и задерживается частями. Выполнение задержанных функций наверстывается после актуализации. Подробности вы найдете в следующих главах. |
| Самотестирование CPU | Не выполняется | Не выполняется |
| Функции тестирования и ввода в действие, например, «Наблюдение и управление переменными», «Контроль (вкл/выкл)» | Функции тестирования и ввода в действие невозможны. Если эти действия как раз выполняются, то установление связи и актуализация невозможны. | Функции тестирования и ввода в действие невозможны. |
| Обработка соединений с главным CPU | Все соединения сохраняются; создание новых соединений невозможно. | Все соединения сохраняются; создание новых соединений невозможно. Разорванные соединения восстанавливаются только после актуализации |
| Обработка соединений с резервным CPU | Все соединения разрываются; создание новых соединений невозможно. | Все соединения уже разорваны. Разрыв произошел при установлении связи. |

6.2 Процесс установления связи и актуализации

Имеются два вида установления связи и актуализации:

- При «нормальном» установлении связи и актуализации отказоустойчивая система должна быть переведена из одиночного режима в состояние **резервирования**. После этого оба CPU синхронно обрабатывают одну и ту же программу.
- При установлении связи и актуализации с **переключением главный/резервный** управление процессом может взять на себя второй CPU с измененными компонентами. Может быть изменена аппаратная конфигурация, конфигурация памяти или операционная система.
Чтобы вернуться в состояние резервирования, должны быть последовательно выполнены «нормальное» установление связи и актуализация.

Как запустить установление связи и актуализацию

Начальная ситуация: Одиночный режим, т.е. только один из CPU отказоустойчивой системы, соединенных между собой волоконно-оптическими кабелями, находится в режиме RUN.

Установление связи и актуализация для перехода в состояние резервирования могут быть инициированы следующим образом:

- Перевод переключателя режимов работы на резервном CPU из STOP в RUN или RUN-P
- Включение питания на резервном CPU (положение переключателя режимов работы RUN или RUN-P), если перед выключением питания CPU не находился в состоянии STOP
- Управление на PG/ES.

Установление связи и актуализацию с переключением главный/резервный можно запустить только с помощью управления оператора на PG/ES.

Внимание

Если установление связи и актуализация на резервном CPU были прерваны (например, из-за выключения питания или перехода в STOP), то несогласованность данных может привести к появлению запроса на сброс памяти этого CPU. После сброса памяти резервного CPU установление связи и актуализация снова становятся возможными

Диаграмма процесса установления связи и актуализации

На следующем рисунке в общих чертах представлен процесс установления связи и актуализации. В исходном состоянии главный CPU находится в одиночном режиме. На рисунке предполагается, что главным является CPU 0.

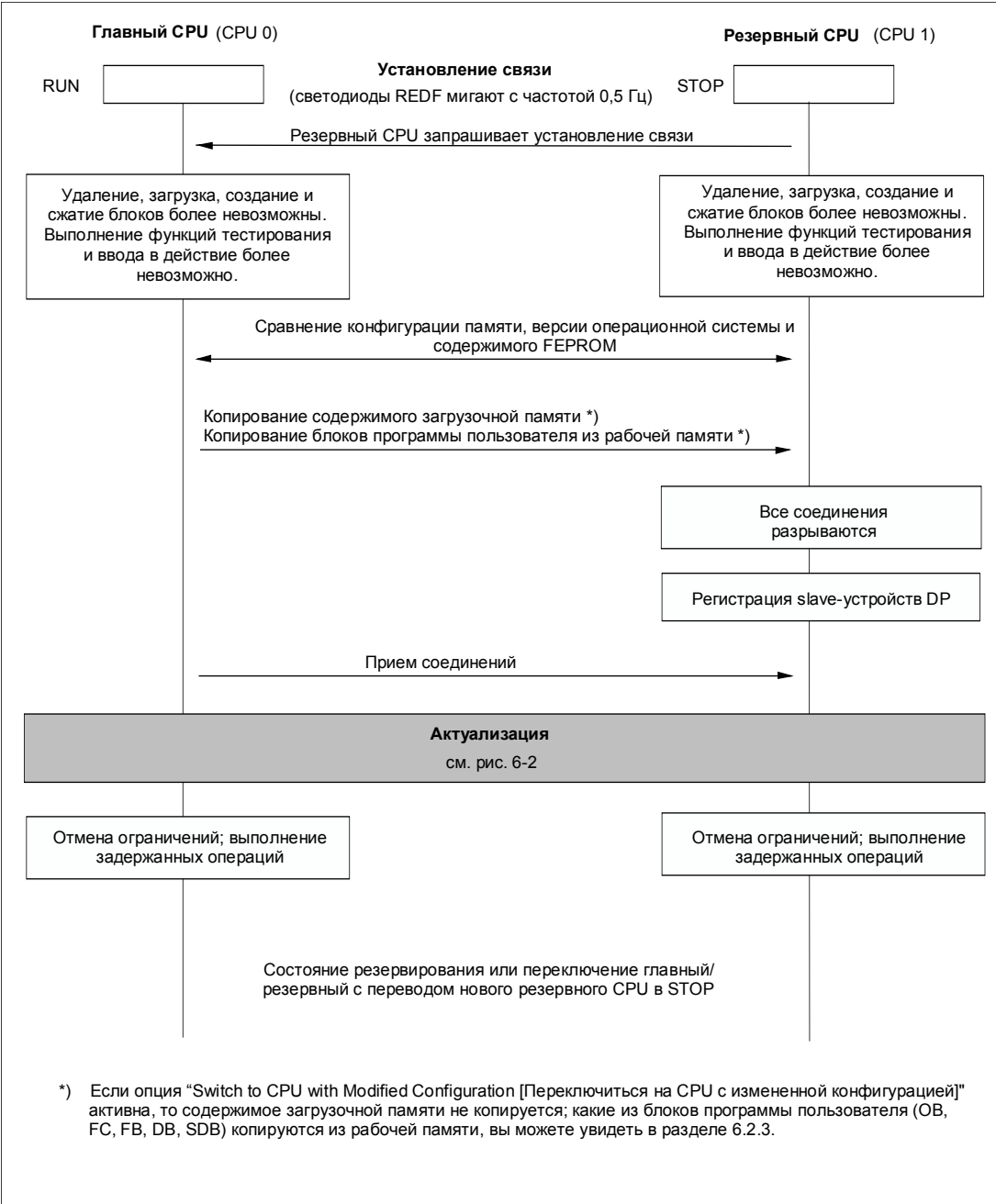


Рис. 6–1. Процесс установления связи и актуализации

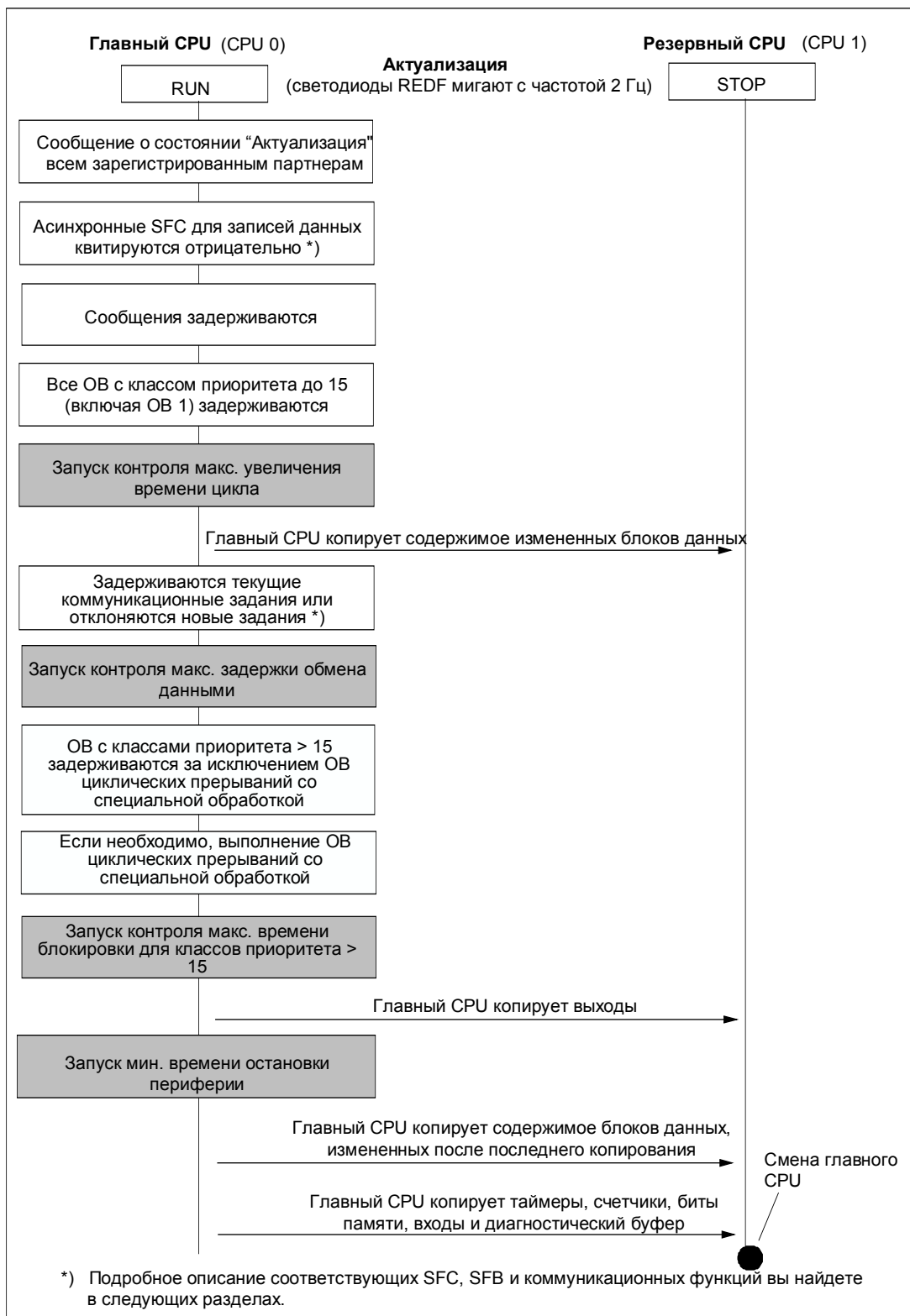


Рис. 6–2. Процесс актуализации

Минимальная длительность входных сигналов во время актуализации

Во время актуализации обработка программы на определенное время останавливается (позже мы обсудим эту тему более подробно). Чтобы изменение входного сигнала могло быть надежно обнаружено CPU даже во время актуализации, должны быть выполнены следующие условия:

Мин. длительность сигнала $> 2 \times$ время актуализации периферии (только для DP)
 + интервал вызова класса приоритета
 + время обработки программы класса приоритета
 + время для актуализации
 + время обработки программ, имеющих более высокий класс приоритета

Пример:

Минимальная длительность входного сигнала, анализируемая в классе приоритета > 15 (например, OB 40).

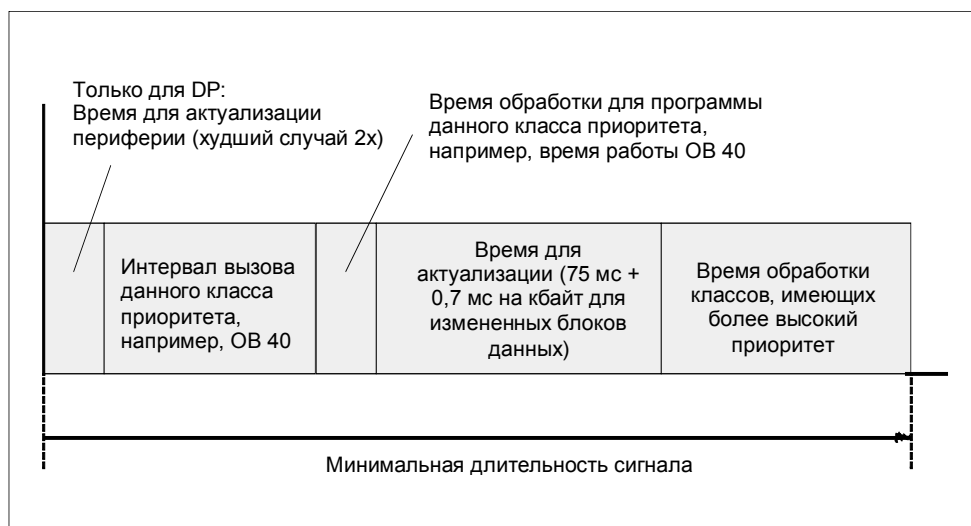


Рис. 6–3. Пример минимальной длительности входного сигнала во время актуализации

6.2.1 Процесс установления связи

В процессе установления связи следует различать, нужно ли системе перейти в состояние резервирования, или нужно главный CPU сделать резервным и наоборот (произвести переключение главный/резервный).

Установление связи для перехода в состояние резервирования

Чтобы исключить различия в обеих подсистемах, главный и резервный CPU выполняют следующие сравнения.

Проверяются:

1. идентичность конфигураций памяти
2. идентичность версий операционных систем
3. идентичность содержимого загрузочной памяти (плата флэш-ПЗУ)
4. идентичность содержимого загрузочной памяти (встроенное статическое ОЗУ и плата ОЗУ)

Если 1., 2. или 3. не одинаковы, то резервный CPU переходит в STOP с сообщением об ошибке.

Если не одинаковы 4., то главный CPU копирует программу пользователя в загрузочную память из своего ОЗУ в резервный CPU.

Программа пользователя, находящаяся в загрузочной памяти флэш-ПЗУ, не копируется. Она должна быть идентична уже до установления связи.

Установление связи с переключением главный/резервный

В STEP 7 вы можете выбрать одну из следующих опций и деблокировать тем самым:

- «Switch to CPU with modified configuration [Переключение на CPU с измененной конфигурацией]»
- «Switch to CPU with expanded memory configuration [Переключение на CPU с расширенной конфигурацией памяти]»

Опция «Switch to CPU with modified operating system [Переключение на CPU с измененной операционной системой]» не может быть использована с существовавшими до сих пор версиями операционных систем.

Переключение на CPU с измененной конфигурацией

Вы могли изменить на резервном CPU:

- аппаратную конфигурацию
- вид загрузочной памяти (напр., заменить плату ОЗУ платой FLASH-памяти); новая загрузочная память может быть больше или меньше старой.

При установлении связи блоки из главного CPU в резервный не копируются (точное положение вещей писано в разделе 6.2.3.).

Шаги, которые необходимо выполнить в вышеупомянутых сценариях (изменение аппаратной конфигурации, изменение вида загрузочной памяти), описаны в главе 11.

Замечание

Если вы не изменяли ни аппаратную конфигурацию, ни вид загрузочной памяти на резервном CPU, переключение главный/резервный все же выполняется, и прежний главный CPU переходит в состояние STOP.

Переключение на CPU с расширенной конфигурацией памяти

Вы могли выполнить на резервном CPU следующие изменения памяти:

- увеличение рабочей памяти и/или
- увеличение загрузочной памяти. При этом модули загрузочной памяти должны быть одного вида, т.е. платы ОЗУ или платы FLASH-памяти; у плат FLASH-памяти содержимое должно совпадать.

При установлении связи блоки программы пользователя (OB, FC, FB, DB, SDB) главного CPU копируются из загрузочной и рабочей памяти в резервный CPU (исключение: если модули загрузочной памяти являются платами флэш-памяти, то копируются только блоки из рабочей памяти).

Шаги, которые необходимо выполнить в вышеупомянутых сценариях (увеличение рабочей памяти, увеличение загрузочной памяти), описаны в главе 11.

Внимание

Если вы изменили вид загрузочной памяти или операционную систему на резервном CPU, то он не перейдет в RUN, а вернется в состояние STOP с соответствующей записью в диагностическом буфере.

Если вы не увеличили ни рабочую, ни загрузочную память на резервном CPU, то он не перейдет в RUN, а вернется в состояние STOP с соответствующей записью в диагностическом буфере.

Переключение главный/резервный не выполняется, и CPU, который до сих пор был главным, останется в режиме RUN.

6.2.2 Процесс актуализации

Что происходит при актуализации?

При актуализации обработка коммуникационных функций и ОВ ограничивается частями. Точно также, все динамические данные (содержимое блоков данных, таймеры, счетчики и биты памяти) передаются в резервный CPU.

Процесс актуализации происходит следующим образом:

1. Все асинхронно исполняющиеся SFC, которые обращаются к записям данных периферийных модулей (SFC 13, 51 и 55–59), до конца актуализации получают «отрицательное» квитирование:
 - Текущее задание возвращает BUSY = TRUE. Оно полностью выполняется по окончании актуализации.
 - Задание, которое прерывается во время актуализации, по окончании актуализации возвращает значение W#16#80C3 (SFC 13, 55–59) или W#16#8085 (SFC 51). Задания, возвращающие эти значения, по окончании актуализации должны быть повторены программой пользователя.
 - Задание, которое вы захотите запустить во время актуализации, отвергается с возвращаемым значением W#16#80C3 (SFC 13, 55–59) или W#16#8085 (SFC 51). Задания, возвращающие эти значения, по окончании актуализации должны быть повторены программой пользователя.
2. Функции сообщений задерживаются до конца актуализации (см. следующий список).
3. Выполнение ОВ 1 и всех ОВ до класса приоритета 15 включительно задерживается.
 В случае циклических прерываний генерирование новых запросов ОВ блокируется, так что новые циклические прерывания не сохраняются и, следовательно, не возникает новых ошибок запроса.
 Только по окончании актуализации генерируется и обрабатывается максимум один запрос для каждого ОВ циклических прерываний. Метка времени циклических прерываний, сгенерированных с задержкой, не может быть проанализирована.
4. Передача содержимого всех блоков данных, которые были изменены после установления связи.
5. Задания на обмен данными, из которых CPU сам выводит задания для других модулей (напр., периферии), получают отрицательное квитирование (см. следующий список).
6. Инициализирующие вызовы (т.е. вызовы, которые имеют следствием манипуляции с рабочей памятью, см. также *Системное программное обеспечение для S7–300/400, Системные и стандартные функции*) коммуникационных функций квитируются отрицательно. Все остальные коммуникационные функции задерживаются и выполняются по окончании актуализации.

7. Блокируется генерирование новых запросов ко всем ОВ (в том числе также и к имеющим класс приоритета > 15), так что новые прерывания не сохраняются и, следовательно, не происходит ошибок запроса.

Прерывания, находящиеся в очереди, снова запрашиваются и обрабатываются только по окончании актуализации. Метка времени прерываний, сгенерированных с задержкой, не может быть проанализирована.

Обработка программы пользователя и актуализация периферии больше не выполняются.

8. Генерирование стартового события для ОВ циклических прерываний со специальной обработкой, если его класс приоритета > 15, и исполнение этого ОВ, если необходимо.

Замечание

ОВ циклических прерываний со специальной обработкой имеют значение прежде всего в тех случаях, когда вам нужно в течение определенного времени обратиться к модулям или сегментам программы. Это типично для отказоустойчивых систем. Подробности см. в руководствах *S7-400F and S7-400FH Programmable Controllers* [Программируемые контроллеры S7-400F и S7-400FH] и *S7-300 Programmable Controllers; Fail-Safe Signal Modules* [Программируемые контроллеры S7-300; Отказоустойчивые сигнальные модули].

9. Передача выходов и всего содержимого блоков данных, которые снова были изменены. Передача таймеров, счетчиков, битов памяти и входов. Передача диагностического буфера.

Во время этой синхронизации данных тактовые импульсы для циклических прерываний, прерываний с задержкой и таймеров S7 останавливаются. Это приводит к потере возможно существовавшей синхронности между циклическими прерываниями и прерываниями по времени.

10. Отмена всех ограничений. Выполняются задержанные прерывания и коммуникационные функции. Все ОВ снова обрабатываются.

Для задержанных ОВ циклических прерываний не гарантируется эквидистантность по отношению к предыдущим вызовам.

Внимание

Прерывания от процесса и диагностические прерывания сохраняются периферией. Если такие прерывания были установлены модулями децентрализованной периферии, то они выполняются после снятия блокировки. Если они были установлены модулями центральной периферии, то все они могут быть воспроизведены только в том случае, если запрос на одно и то же прерывание во время блокировки не выдавался несколько раз.

Если из PG/ES был сделан запрос на переключение главный/резервный, то по окончании актуализации CPU, бывший ранее резервным, становится главным, а CPU, бывший ранее главным, переходит в состояние STOP. В противном случае оба CPU переходят в RUN (состояние резервирования) и выполняют программу пользователя синхронно.

Если было выполнено переключение главный/резервный, то в первом цикле после актуализации OB 1 имеет свой собственный идентификатор (см. справочное руководство *Системное программное обеспечение для S7-300/400, системные и стандартные функции*). О других особенностях, связанных с изменением конфигурации, см. раздел 6.2.3.

Задержанные функции сообщений

Перечисленные SFC, SFB и служебные программы операционной системы запускают сообщения всем зарегистрированным в каждом случае партнерам. После начала актуализации эти функции задерживаются.

- SFC 17 «ALARM_SQ», SFC 18 «ALARM_S», SFC 107 «ALARM_DQ», SFC 108 «ALARM_D»
- SFC 52 «WR_USMSG»
- SFB 31 «NOTIFY_8P», SFB 33 «ALARM», SFB 34 «ALARM_8», SFB 35 «ALARM_8P», SFB 36 «NOTIFY», SFB 37 «AR_SEND»
- Состояния
- Системные диагностические сообщения

Задания на блокировку и деблокировку сообщений с помощью SFC 9 «EN_MSG» и SFC 10 «DIS_MSG», начиная с этого момента, отклоняются с отрицательным возвращаемым значением.

Коммуникационные функции с вторичными заданиями

Если CPU получает одно из перечисленных ниже заданий, он, в свою очередь, должен генерировать из них коммуникационные задания и посылать их другим модулям. Это могут быть, например, команды на запись в модули децентрализованной периферии или чтение из них записей данных для параметризации. Эти задания отклоняются вплоть до завершения актуализации.

- Запись и чтение записей данных с помощью функций визуализации
- Чтение записей данных через списки состояний системы (SSL)
- Блокировка и деблокировка сообщений
- Регистрация и отмена регистрации сообщений
- Квитирование сообщений

Замечание

Последние три функции регистрируются системой WinCC и автоматически повторяются по окончании актуализации.

6.2.3 Переключение на CPU с измененной конфигурацией

Если процедура установления связи и актуализации была запущена из STEP 7 с помощью опции «Switch to CPU with modified configuration [Переключение на CPU с измененной конфигурацией]», то этот процесс отличается от обработки содержимого памяти.

Загрузочная память

Компоненты загрузочной памяти не копируются в резервный CPU из главного CPU.

Рабочая память

Следующие компоненты передаются из рабочей памяти главного CPU в резервный CPU:

- Содержимое всех блоков данных, которые имеют одинаковую метку времени интерфейса в той и другой загрузочной памяти и у которых не установлены атрибуты «Read Only [только чтение]» и «unlinked [не связан]».
- Блоки данных, сгенерированные SFC в главном CPU.
Блоки данных, сгенерированные SFC в резервном CPU, стираются.
Если в загрузочной памяти резервного CPU содержится блок данных с таким же номером, то установление связи прерывается с записью в диагностический буфер.
- Образы процесса, таймеры, счетчики и биты памяти
- Диагностический буфер

Если диагностический буфер в резервном CPU спроектирован меньше, чем в главном CPU, то передается лишь такое количество записей, на которое спроектирован резервный CPU. Для этого из главного CPU выбираются самые новые записи.

Если памяти недостаточно, то установление связи прерывается с записью в диагностический буфер.

Если были изменены блоки данных, содержащие экземпляры SFB S7-связи, то эти экземпляры возвращаются в состояние, в котором они находились перед первым вызовом.

Замечание

При переключении на CPU с измененной конфигурацией загрузочная память главного и резервного CPU могут иметь различные размеры.

6.2.4 Блокировка установления связи и актуализации

Установление связи и актуализация связаны с увеличением времени цикла. В нем есть интервал времени, в котором не выполняется актуализация периферии (см. раздел 6.3 «Контроль времени»). Это особенно нужно принимать во внимание в тех случаях, если используется децентрализованная периферия, и после актуализации производится замена главного CPU резервным и наоборот (переключение главный/резервный) (т.е. при изменении конфигурации во время работы).



Предостережение

Выполняйте установление связи и актуализацию только в не критических состояниях процесса.

Чтобы самому задать время начала установления связи и актуализации, в вашем распоряжении имеется функция SFC 90 «H_CTRL». Подробное описание этой SFC вы найдете в руководстве *Системное программное обеспечение для S7-300/400, Системные и стандартные функции*.

Внимание

Если процесс допускает увеличение времени цикла в любой момент времени, то в вызове SFC 90 «H_CTRL» нет необходимости.

Во время установления связи и актуализации самотестирование CPU не выполняется. Поэтому у отказобезопасных систем обратите внимание на то, чтобы не затягивать процесс актуализации на слишком большой интервал времени. За дополнительными подробностями обратитесь к руководству *S7-400F and S7-400FH Programmable Controllers [Программируемые контроллеры S7-400F и S7-400FH]*.

Пример процесса, критического ко времени

Пусть каретка с кулачком длиной 50 мм движется вдоль оси с постоянной скоростью $v = 10 \text{ км/час} = 2,78 \text{ м/с} = 2,78 \text{ мм/мс}$. На оси находится выключатель. Таким образом, выключатель активизируется кулачком в течение интервала времени $\Delta t = 18 \text{ мс}$.

Чтобы активизация выключателя могла быть распознана CPU, время блокировки для классов приоритета > 15 (определение см. ниже) должно быть заметно меньше 18 мс.

Так как в STEP 7 для классов приоритета > 15 можно установить максимальное время блокировки только 0 мс или значение между 100 и 60000 мс, то вы должны воспользоваться одним из следующих мероприятий:

- Перенесите начало установления связи и актуализации на время, когда состояние процесса не является критическим. Для этого воспользуйтесь SFC 90 «H_CTRL» (см. выше).
- Используйте значительно более длинный кулачок и/или значительно уменьшите скорость перемещения каретки перед тем, как она достигнет выключателя.

6.3 Контроль времени

Во время актуализации обработка программы останавливается на определенный интервал времени. Раздел 6.3 имеет значение для вас, если этот интервал критичен для вашего процесса. Если это так, запроектируйте описанные ниже времена контроля.

Во время актуализации отказоустойчивая система контролирует, чтобы увеличение времени цикла, задержка связи и время блокировки для классов приоритета > 15 не превысили запроектированные вами максимальные значения; одновременно она обеспечивает соблюдение запроектированного минимального времени останова периферии.

Внимание

Если вы не задали значений для времен контроля, то вы должны учесть актуализацию во времени контроля цикла. В этом случае, если актуализация прерывается, то отказоустойчивая система переходит в одиночный режим: CPU, который был до сих пор главным, остается в режиме RUN, а резервный CPU переходит в состояние STOP.

Вы можете запроектировать все времена контроля или ни одного.

При проектировании времен контроля вы приняли во внимание технологические требования.

Ниже времена контроля разъясняются более подробно.

- Максимальное увеличение времени цикла
 - Увеличение времени цикла: интервал во время актуализации, в течение которого не выполняется ОВ 1 (а также и другие ОВ с классом приоритета до 15). На этом интервале «нормальный» контроль времени не действует.
 - Максимальное увеличение времени цикла: максимально допустимое увеличение времени цикла, которое вы запроектировали.
- Максимальная задержка связи
 - Задержка связи: интервал во время актуализации, в течение которого не выполняются коммуникационные функции. (Примечание: существующие коммуникационные связи главного CPU сохраняются.)
 - Максимальная задержка связи: максимально допустимая задержка связи, которую вы запроектировали.
- Максимальное время блокировки классов приоритета > 15
 - Время блокировки классов приоритета > 15 : интервал во время актуализации, в течение которого никакие ОВ не выполняются (таким образом, не выполняется и программа пользователя) и больше не производится актуализация периферии.
 - Максимальное время блокировки классов приоритета > 15 : максимально допустимое время блокировки, которое вы запроектировали для классов приоритета > 15 .

- Минимальное время останова периферии:

Это интервал времени между копированием выходов из главного CPU в резервный и моментом времени перехода в состояние резервирования или переключения главный/резервный (момент времени, когда бывший главный CPU переходит в состояние STOP, а новый главный CPU переключается в режим RUN). В течение этого времени управляются выходы обоих CPU. Это предотвращает отключение периферии даже в случае актуализации с переключением главный/резервный.

Минимальное время останова периферии имеет особое значение при актуализации с переключением главный/резервный. Если минимальное время останова периферии вы спроектируете равным 0, то при изменении системы во время работы выходы могут отключиться.

Моменты запуска различных видов контроля времени показаны на рис. 6–2 (блоки с серым фоном). В любом случае эти времена завершаются, когда достигается состояние резервирования системы или происходит переключение главный/резервный (т.е. когда новый главный CPU переходит в режим RUN) в конце актуализации.

На следующем рисунке представлены времена, имеющие значение при актуализации.

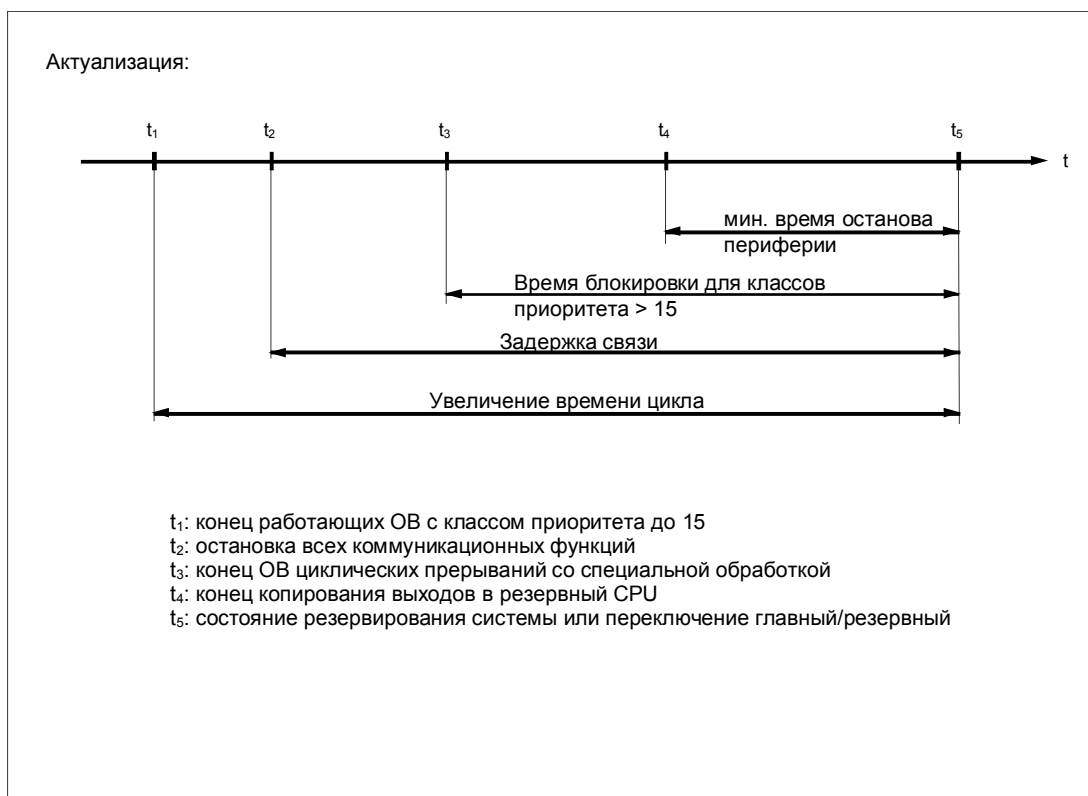


Рис. 6–4. Времена, имеющие значение при актуализации

Реакция на превышение времени

Если одно из контролируемых времен превысит максимальное запрооектированное значение, то запускается следующий процесс:

1. Актуализация прекращается
2. Отказоустойчивая система остается в одиночном режиме с существующим главным CPU в режиме RUN
3. Причина прекращения актуализации вносится в диагностический буфер
4. Вызывается OB 72 (с соответствующей стартовой информацией)

Затем резервный CPU снова анализирует свои системные блоки данных. После этого, но не раньше, чем через минуту, делается новая попытка установления связи и актуализации. Если 10 попыток в целом оказываются безуспешными, то новые попытки не делаются. Тогда вы должны снова запустить установление связи и актуализацию.

Причинами для истечения времен контроля могут быть:

- большая загрузка прерываниями (напр., от периферийных модулей)
- высокая коммуникационная загрузка, так что обработка текущих функций длится дольше
- на конечном этапе актуализации в резервный CPU должны быть скопированы большие объемы данных.

6.3.1 Временные характеристики

Временные характеристики при установлении связи

Во время установления связи на контроллер вашей системы должно оказываться как можно меньше воздействий. Длительность установления связи возрастает с ростом текущей нагрузки вашего программируемого логического контроллера. Длительность установления связи зависит в первую очередь от

- коммуникационной загрузки
- времени цикла

Следующее относится к незагруженной системе автоматизации:

Время установления связи = величине загрузочной и рабочей памяти в Мбайтах × 1 с + основная загрузка

Основная загрузка составляет несколько секунд.

При более высокой загрузке вашей системы автоматизации зависящая от памяти часть может возрасти до 1 минуты на Мбайт.

Временные характеристики при актуализации

Время передачи при актуализации зависит от количества и общей длины измененных блоков данных; оно не зависит от объема измененных данных внутри блока. Оно зависит также от текущего состояния процесса и от коммуникационной загрузки.

В первом приближении подлежащее проектированию максимальное время блокировки для классов приоритета > 15 может рассматриваться как функция объема данных в рабочей памяти. Объем кода в рабочей памяти значения не имеет.

6.3.2 Определение времен контроля

Определение с помощью STEP7 или с использованием формул

Времена контроля, перечисленные ниже, автоматически рассчитываются в STEP 7 версии 5.2 и выше для каждого нового проекта. Вы их можете также рассчитать с помощью формул и процедур, описанных ниже. Они эквивалентны формулам, используемым в STEP7.

- Максимальное увеличение времени цикла
- Максимальная задержка связи
- Максимальное время блокировки для классов приоритета
- Минимальное время останова периферии

Вы можете также автоматически рассчитать времена контроля в HW Config диалоге Properties CPU [Свойства CPU]-> Trigger H Parameters [Запустить параметры обеспечения отказоустойчивости].

Точность времен контроля

Замечание

Времена контроля, определенные с помощью STEP 7 или по формулам, являются всего лишь рекомендациями.

Они ориентируются на отказоустойчивую систему с двумя партнерами по обмену данными и средней коммуникационной нагрузкой.

Так как профиль вашей системы может сильно отличаться от этого предположения, то вы должны принять во внимание следующие правила.

- Время цикла может сильно увеличиться при возрастании коммуникационной нагрузки.
- Если вы выполняете изменения своей системы во время работы, то это может существенно увеличить время цикла.
- Чем большая часть программы (особенно коммуникационные блоки) обрабатывается в классах приоритета > 15, тем больше задержка связи и увеличение времени цикла.
- В небольших системах с высокими требованиями к производительности вы можете уменьшить рассчитанные времена контроля.

Использование резервных модулей ввода и вывода

Внимание

Если вы резервируете периферийные модули и соответствующим образом приняли их во внимание в своей программе, то вы должны увеличить рассчитанные времена контроля во избежание толчков на модулях вывода.

Добавка нужна только тогда, когда вы эксплуатируете с резервированием модули из следующей таблицы.

Таблица 6–2. Увеличение времен контроля при использовании периферии с резервом

| Тип модуля | Добавка в мс |
|---|--------------|
| ET200M: стандартные модули вывода | 2 |
| ET200M: модули вывода типа HART | 10 |
| ET200M: отказобезопасные модули вывода | 50 |
| ET200L–SC с аналоговым выводом | ≤ 80 |
| ET200S с аналоговым выводом или с технологическими модулями | ≤ 20 |

Действуйте следующим образом:

- Определите из таблицы добавку. Если в режиме резервирования используется несколько типов модулей из таблицы, то возьмите наибольшую добавку.
- Добавьте ее к уже определенным временам контроля.

Проектирование времен контроля

При проектировании времен контроля необходимо принять во внимание следующие зависимости; их соблюдение проверяется STEP 7:

Максимальное увеличение времени цикла

> макс. задержки связи

> (макс. времени блокировки для классов приоритета > 15)

> мин. времени останова периферии

Если при установлении связи и актуализации с переключением главный/резервный CPU были спроектированы с различными значениями для функций контроля, то применяется большее из двух значений.

Расчет минимального времени остановки периферии (T_{PH})

Следующее относится к расчету минимального времени остановки периферии:

- для центральной периферии: $T_{PH} = 30 \text{ мс}$
- для децентрализованной периферии I/O: $T_{PH} = 3 \times T_{TRmax}$
где T_{TRmax} = наибольшее время сканирования цели всех master-систем DP отказоустойчивой станции

При использовании центральной и децентрализованной периферии результирующий минимум времени остановки периферии составляет:

$$T_{PH} = \text{MAX} (30 \text{ мс}, 3 \times T_{TRmax})$$

На рис. 6–5 показано соотношение между мин. временем остановки периферии и макс. временем блокирования классов приоритета > 15.

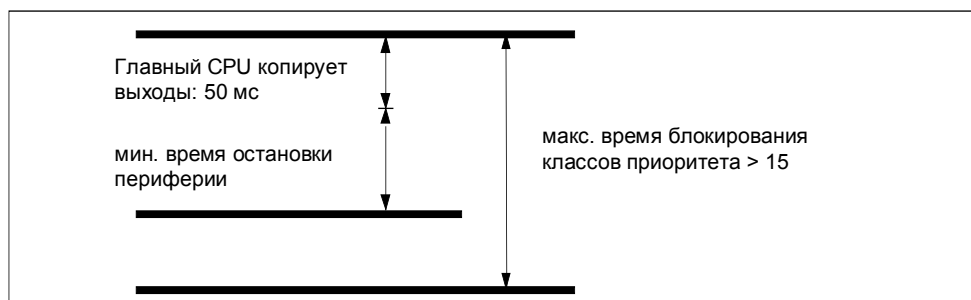


Рис. 6–5. Соотношение между мин. временем остановки периферии и макс. временем блокирования классов приоритета > 15

Обратите внимание на следующее условие:

$50 \text{ мс} + \text{мин. время остановки периферии} \leq (\text{макс. времени блокирования классов приоритета} > 15)$.

Из этого вытекает, что выбранное большим мин. время остановки периферии может оказаться определяющим для макс. времени блокирования классов приоритета > 15.

Расчет макс. времени блокирования классов приоритета > 15 (T_{P15})

Макс. время блокирования классов приоритета > 15 определяется 4 факторами:

- Как показано на рис. 6–2, после последнего копирования в резервный CPU, снова передается в резервный CPU. **Количество и структура блоков данных**, которые вы описываете в классах с высоким приоритетом, определяет длительность этого процесса и, таким образом, макс. время блокирования классов приоритета > 15. Указание вы получите в нижеприведенных способах устранения.
- На последнем этапе актуализации все ОВ задерживаются или блокируются. Во избежание необоснованного увеличения макс. времени блокирования классов приоритета > 15 в результате неблагоприятного программирования обрабатывайте наиболее критичные к времени компоненты периферии в **выбранном циклическом прерывании**. Это имеет особое значение в случае отказобезопасных программ пользователя. Это циклическое прерывание определяется при проектировании; затем оно снова исполняется непосредственно после начала отсчета макс. времени блокирования для классов приоритета > 15, но только в том случае, если вы ему назначили класс приоритета > 15.
- При установлении связи и актуализации с переключением главный/резервный (см. раздел 6.2.1) после завершения актуализации еще активный канал связи у включенных slave-устройств DP должен быть переключен. Это увеличивает время, в течение которого допустимые значения не могут считываться или выводиться. Длительность этого процесса определяется вашей **аппаратной конфигурацией**.

- Из **технологических условий вашего процесса** вытекают требования относительно того, как долго может откладываться актуализация периферии. Это особенно важно в случае контролируемых временем процессов в отказобезопасных системах.

Указание

Другие факторы, на которые следует обратить внимание при использовании отказобезопасных модулей, описаны в руководствах: *S7-400 F and S7-400 FH Programmable Controllers* [Программируемые контроллеры S7-400 F и S7-400 FH] и *S7-300 Programmable Controllers; Fail-Safe Signal Modules* [Программируемые контроллеры S7-300; Отказобезопасные сигнальные модули]. Это относится, в частности, к внутренним временам исполнения в отказобезопасных модулях.

1. Для каждой master-системы DP определите из параметров шины в STEP 7
 - T_{TR} для master-системы DP
 - время переключения DP (обозначенное ниже как T_{DP_UM})
2. Для каждой master-системы DP определите из технических данных для включенных slave-устройств DP
 - максимальное время переключения для активного канала связи (обозначенное ниже как T_{SLAVE_UM}).
3. Из технологических спецификаций своей установки определите
 - максимально допустимый интервал времени, в течение которого не происходит актуализация ваших периферийных модулей (обозначенный ниже как T_{PTO}).
4. Из своей пользовательской программы определите
 - время цикла для циклического прерывания, имеющего самый высокий приоритет, или выбранного вами (см. выше) (T_{WA})
 - время исполнения вашей программы в этом циклическом прерывании (T_{PROG})
5. Для каждой master-системы DP из этого получается

$$T_{P15} (\text{master-система DP}) = T_{PTO} - (2 \times T_{TR} + T_{WA} + T_{PROG} + T_{DP_UM} + T_{SLAVE_UM}) \quad [1]$$

Указание

Если T_{P15} (master-система DP) < 0 , то расчет здесь следует остановить. Возможные способы устранения приведены после следующего примера расчета. Выполните необходимые изменения и начните расчет снова с пункта 1.

6. Выберите минимальное из всех значение T_{P15} (master-система DP). В дальнейшем это значение называется T_{P15_HW} .
7. Определите долю максимального времени блокирования для классов приоритета > 15 , обусловленную минимальным временем остановки периферии (T_{P15_OD}):

$$T_{P15_OD} = 50 \text{ мс} + \text{мин. время остановки периферии} \quad [2]$$

Указание

Если $T_{P15_OD} > T_{P15_HW}$, то расчет здесь следует остановить. Возможные способы устранения приведены после следующего примера расчета. Выполните необходимые изменения и начните расчет снова с пункта 1.

8. Из раздела 6.3.4 выясните долю максимального времени блокирования для классов приоритета > 15 , обусловленную программой пользователя (T_{P15_AWP}).

Указание

Если $T_{P15_AWP} > T_{P15_HW}$, то расчет здесь следует остановить. Возможные способы устранения приведены после следующего примера расчета. Выполните необходимые изменения и начните расчет снова с пункта 1.

9. Рекомендуемое значение для максимального времени блокирования для классов приоритета > 15 теперь получается из формулы:

$$T_{P15} = \text{MAX} (T_{P15_AWP}, T_{P15_OD}) \quad [3]$$

Пример расчета T_{P15}

В дальнейшем максимально допустимый интервал времени при актуализации, в течение которого операционная система не выполняет обработку программы и не актуализирует периферию, определяется для следующей конфигурации системы.

Пусть имеются две master-системы DP: пусть master-система DP_1 «соединена» с CPU через интерфейс MPI/DP CPU, а master-система DP_2 через внешний интерфейсный master-модуль DP.

1. Из параметров шины в STEP 7:

$$T_{TR_1} = 25 \text{ мс}$$

$$T_{TR_2} = 30 \text{ мс}$$

$$T_{DP_UM_1} = 100 \text{ мс}$$

$$T_{DP_UM_2} = 80 \text{ мс}$$

2. Из технических данных для используемых slave-устройств DP:

$$T_{SLAVE_UM_1} = 30 \text{ мс}$$

$$T_{SLAVE_UM_2} = 50 \text{ мс}$$

3. Из технологических спецификаций для вашей системы:

$$T_{PTQ_1} = 1250 \text{ мс}$$

$$T_{PTQ_2} = 1200 \text{ мс}$$

4. Из программы пользователя:

$$T_{WA} = 300 \text{ мс}$$

$$T_{PROG} = 50 \text{ мс}$$

5. Из формулы [1]:

$$T_{P15} \text{ (master-система DP_1)} \\ = 1250 \text{ мс} - (2 \times 25 \text{ мс} + 300 \text{ мс} + 50 \text{ мс} + 100 \text{ мс} + 30 \text{ мс}) = 720 \text{ мс}$$

$$T_{P15} \text{ (master-система DP_2)} \\ = 1200 \text{ мс} - (2 \times 30 \text{ мс} + 300 \text{ мс} + 50 \text{ мс} + 80 \text{ мс} + 50 \text{ мс}) = 660 \text{ мс}$$

Проверка: так как $T_{P15} > 0$, продолжить

6. $T_{P15_HW} = \text{MIN}(720 \text{ мс}, 660 \text{ мс}) = 660 \text{ мс}$

7. Из формулы [2]:

$$T_{P15_OD} = 50 \text{ мс} + T_{PH} = 50 \text{ мс} + 90 \text{ мс} = 140 \text{ мс}$$

Проверка: так как $T_{P15_OD} = 140 \text{ мс} < T_{P15_HW} = 660 \text{ мс}$, продолжить

8. Из раздела 6.3.4 для 170 Кбайт данных программы пользователя:

$$T_{P15_AWP} = 194 \text{ мс}$$

Проверка: так как $T_{P15_AWP} = 194 \text{ мс} < T_{P15_HW} = 660 \text{ мс}$, продолжить

9. Из формулы [3] получается рекомендуемое макс. время блокирования для классов приоритета > 15 :

$$T_{P15} = \text{MAX}(194 \text{ мс}, 140 \text{ мс})$$

$$T_{P15} = 194 \text{ мс}$$

Если вы введете 194 мс в качестве максимального времени блокирования для классов приоритета > 15 в STEP 7, то гарантируется, что при длительностях сигналов 1250 мс или 1200 мс изменение сигнала при актуализации всегда будет распознано.

Устранение затруднений, если нельзя рассчитать T_{P15}

Если при расчете максимального времени блокирования для классов приоритета > 15 не получается никаких рекомендаций, то вы можете устранить это затруднение с помощью следующих мероприятий:

- Сократите время цикла для запроецированного циклического прерывания.
- В случае особенно больших времен T_{TR} распределите slave-устройства между несколькими master-системами DP.
- Увеличьте скорость передачи у затронутых master-систем DP.
- Спроектируйте связи DP/PA и Y-связи в отдельных master-системах DP.
- Если slave-устройства DP имеют очень разные времена переключения и, таким образом, очень различные, как правило, T_{PTO} , распределите эти slave-устройства среди нескольких master-систем DP.
- Если в отдельных master-системах DP следует ожидать лишь небольшой загрузки из-за прерываний или параметризации, то вы можете также сократить рассчитанные времена T_{TR} примерно на 20-30%. Но при этом возрастает опасность выхода из строя станции в децентрализованной периферии.

- Время T_{P15_AWP} дает ориентировочное значение; оно зависит от структуры вашей программы. Вы можете его сократить, например, с помощью следующих мероприятий:

- Сохраняйте часто и редко изменяемые данные в разных DB.
- Задавайте меньший размер рабочей памяти для DB.

Если вы сократите время T_{P15_AWP} , не приняв указанных мер, то это увеличит опасность прекращения актуализации из-за истечения времен контроля.

Расчет максимальной задержки связи

Мы рекомендуем использовать следующую формулу:

Максимальная задержка связи = $4 \times$ (максимальное время блокирования для классов приоритета > 15)

Это время в решающей степени определяется состоянием процесса и коммуникационной загрузкой вашей системы. Под этим следует понимать как абсолютную загрузку, так и загрузку относительно размера вашей пользовательской программы. В случае необходимости вы должны это время скорректировать.

Расчет максимального увеличения времени цикла

Мы рекомендуем использовать следующую формулу:

Максимальное увеличение времени цикла = $10 \times$ (максимальное время блокирования для классов приоритета > 15)

Это время в решающей степени определяется состоянием процесса и коммуникационной загрузкой вашей системы. Под этим следует понимать как абсолютную загрузку, так и загрузку относительно размера вашей пользовательской программы. В случае необходимости вы должны это время скорректировать.

6.3.3 Влияние на временные характеристики

Интервал времени, в течение которого не происходит актуализация периферии, в первую очередь определяется следующими факторами:

- количеством и размером блоков данных, изменяемых во время актуализации
- количеством экземпляров SFB S7-связи и SFB для генерирования сообщений, относящихся к блокам
- изменениями системы во время работы
- настройками через динамические количественные структуры
- конфигурацией децентрализованной периферии (с уменьшением скорости передачи и увеличением количества slave-устройств время, необходимое для актуализации периферии, увеличивается.)

В наиболее неблагоприятных условиях этот интервал увеличивается на следующие величины:

- максимальное время цикла используемого циклического прерывания
- длительность всех ОВ циклических прерываний
- длительность ОВ прерываний, имеющих высокий приоритет, которые исполнялись до задержки прерываний

Целенаправленная задержка актуализации

Задержите актуализацию с помощью SFC 90 «H_CTRL» и разблокируйте ее только тогда, когда наступит состояние меньшей коммуникационной загрузки или загрузки прерываниями.



Осторожно

Задержка актуализации увеличивает время, в течение которого отказоустойчивая система работает в одиночном режиме.

6.3.4 Значения производительности для установления связи и актуализации

Доля программы пользователя T_{P15_AWP} в максимальном времени блокирования для классов приоритета > 15

Доля программы пользователя T_{P15_AWP} в максимальном времени блокирования для классов приоритета > 15 может быть рассчитана с помощью следующей формулы:

$$T_{P15_AWP} \text{ в мс} = 0,7 \times \text{размер блоков данных в рабочей памяти в килобайтах} + 75$$

В следующей таблице указаны вытекающие отсюда времена для некоторых типичных значений данных рабочей памяти.

Таблица 6–3. Типичные значения для доли программы пользователя T_{P15_AWP} в максимальном времени блокирования для классов приоритета > 15

| Данные рабочей памяти | T_{P15_AWP} |
|-----------------------|----------------|
| 500 Кбайт | 430 мс |
| 1 Мбайт | 800 мс |
| 2 Мбайт | 1,51 с |
| 5 Мбайт | 3,66 с |
| 10 Мбайт | 7,24 с |

Для этой формулы были сделаны следующие предположения:

- 80% блоков данных изменяются перед задержкой прерываний, имеющих классы приоритета > 15 .
Для отказобезопасных систем это значение должно быть определено более точно, в частности, во избежание простоя драйверных блоков (см. раздел 6.3.2).
- Приблизительно 100 мс времени актуализации на каждый мегабайт рабочей памяти, занятой блоками данных, отводятся для исполняющихся в данный момент или отложенных коммуникационных функций.
В зависимости от коммуникационной загрузки вашей системы автоматизации вы должны увеличить или уменьшить T_{P15_AWP} при настройке.

6.4 Особенности при установлении связи и актуализации

Требования к входным сигналам во время актуализации

При актуализации считанные ранее сигналы процесса сохраняются и не обновляются. Изменение сигнала процесса во время актуализации распознается CPU только в том случае, если измененное состояние сигнала сохраняется и по окончании актуализации.

Импульсы (изменения сигнала «0 → 1 → 0» или «1 → 0 → 1»), возникающие во время актуализации, не распознаются CPU.

Поэтому позаботьтесь о том, чтобы время между двумя изменениями сигнала (длительность импульса) всегда было больше, чем время, необходимое для актуализации.

Коммуникационные соединения и функции

В отличие от более ранних версий ПЗУ, связи с главным CPU больше не разрываются. Однако во время актуализации соответствующие коммуникационные задания не обрабатываются. Они сохраняются и выполняются, как только наступает один из следующих случаев:

- актуализация завершена, и система находится в режиме резервирования
- актуализация и переключение главный/резервный завершены, система находится в одиночном режиме
- актуализация была прекращена (напр., из-за превышения времени контроля), и система снова находится в одиночном режиме.

Во время актуализации невозможен первоначальный вызов коммуникационных блоков.

Запрос на общее стирание при прерывании связи

Если установление связи прерывается во время копирования содержимого загрузочной памяти из главного CPU в резервный, то резервный CPU делает запрос на общее стирание. Это сигнализируется записью в диагностический идентификатор события W#16#6523.