

Операции управления программой

Операция **Условное окончание обработки** заканчивает обработку главной программы в зависимости от результата предшествующей логической операции .

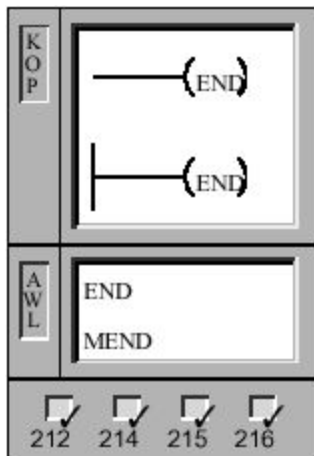
Главная программа должна заканчиваться катушкой

Абсолютное окончание обработки

В AWL операция **Абсолютное окончание обработки**

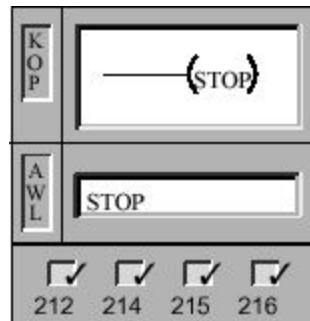
представляется командой MEND.

Операнды : нет



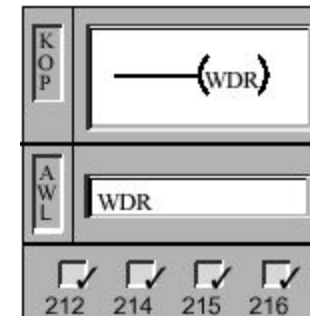
Операция **STOP** заканчивает обработку программы пользователя немедленно , переводя CPU из режима работы RUN в режим работы STOP.

Операнды : нет



Посредством операции **Сброс времени контроля** можно перезапустить время контроля CPU. Благодаря этому удлиняется максимально допустимое время цикла без сообщения об ошибке времени контроля .

Операнды : нет



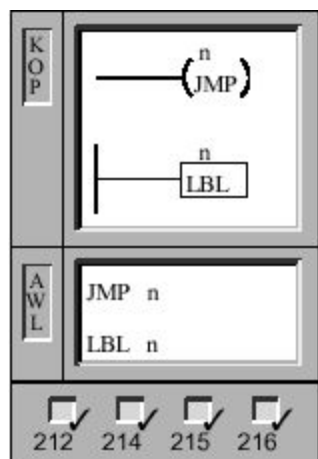
Если операция **STOP** выполняется в программе обработки прерываний , то последняя немедленно завершается и все стоящие в очереди прерывания игнорируются . Остаток программы обрабатывается , и в конце цикла CPU переходит в состояние STOP.

Операции управления программой

Операция **Переход на метку** выполняет ответвление программы к заданной метке перехода (n).

Операция **Определение метки перехода** указывает целевой пункт (n), в который нужно перейти .

Операнды :n: 0 - 255



	KOP	AWL
<p>Network 14</p> <p>Network 33</p>	<p>Если реманентные данные не потеряны, то перейти на LBL 4.</p> <p>Можно использовать JMP и LBL в главной программе, в подпрограммах или в программах обработки прерываний. Операция перехода и относящаяся к ней метка перехода должны всегда находиться в в одной и той же части кода (обе в главной программе, обе в подпрограмме или обе в программе обработки прерываний).</p>	<pre> Network LDN SM0.2 JMP 4 . . . Network LBL 4 </pre>

Операция перехода и относящаяся к ней метка перехода должны обе находиться либо в главной программе , либо в одной подпрограмме , либо в одной программе обработки прерываний . Вы не можете перейти из главной программы на метку , расположенную в подпрограмме или в программе обработки прерываний . Вы также не можете из подпрограммы или программы обработки прерываний перейти на метку , расположенную вне соответствующей подпрограммы или программы обработки прерываний .

Операции управления программой

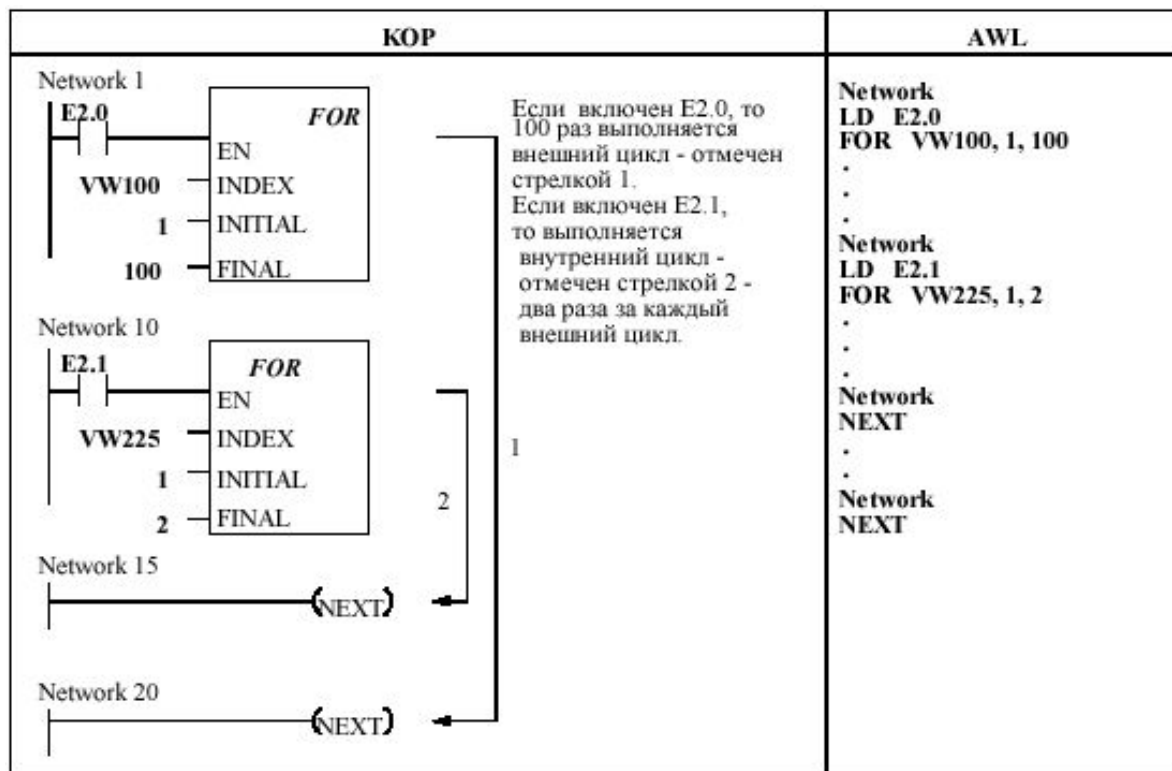
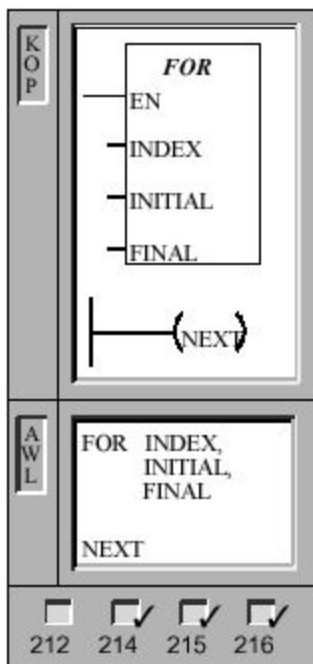
Операция **Программный цикл с FOR** выполняет операции между FOR и NEXT. Вы должны задать текущее значение счетчика программного цикла (**INDEX**), начальное значение (**INITIAL**) и конечное значение (**FINAL**).

Операция **Конец программного цикла с FOR** (NEXT) отмечает конец программного цикла с FOR и устанавливает вершину стека в "1".

Операнды : **INDEX:** VW, T, Z, EW, AW, MW, SMW, AC, *VD, *AC, SW

INITIAL: VW, T, Z, EW, AW, MW, SMW, AC, AEW, константа, *VD, *AC, SW

FINAL: VW, T, Z, EW, AW, MW, SMW, AC, AEW, константа, *VD, *AC, SW



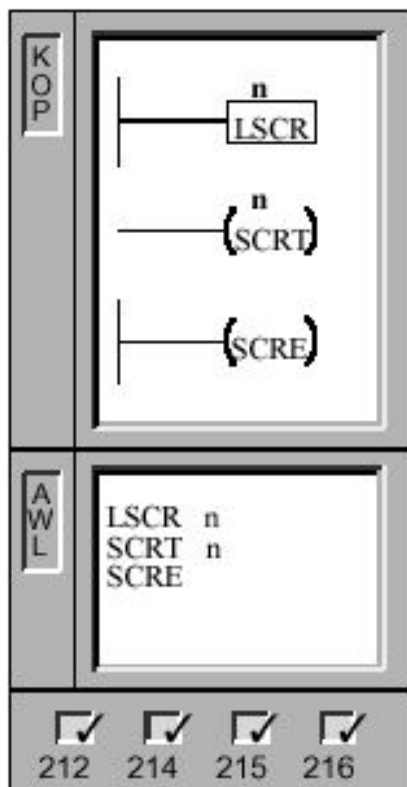
Операции управления программой

Операция **Загрузка реле шагового управления** отмечает начало сегмента SCR. Если $n = 1$, то разблокируется поток сигнала к сегменту SCR. Сегмент SCR должен заканчиваться операцией SCRE.

Операция **Фронт реле шагового управления** отмечает SCR–бит, который должен разблокироваться (S–бит, который должен быть установлен). Если поток сигнала поступает к катушке, то включается указанный S–бит и выключается S–бит операции LSCR (разблокировал данный сегмент SCR).

Операция **Конец реле шагового управления** отмечает конец сегмента SCR.

Операнды :n: бит в области S, размер которой 16 байт (SB0... SB15)

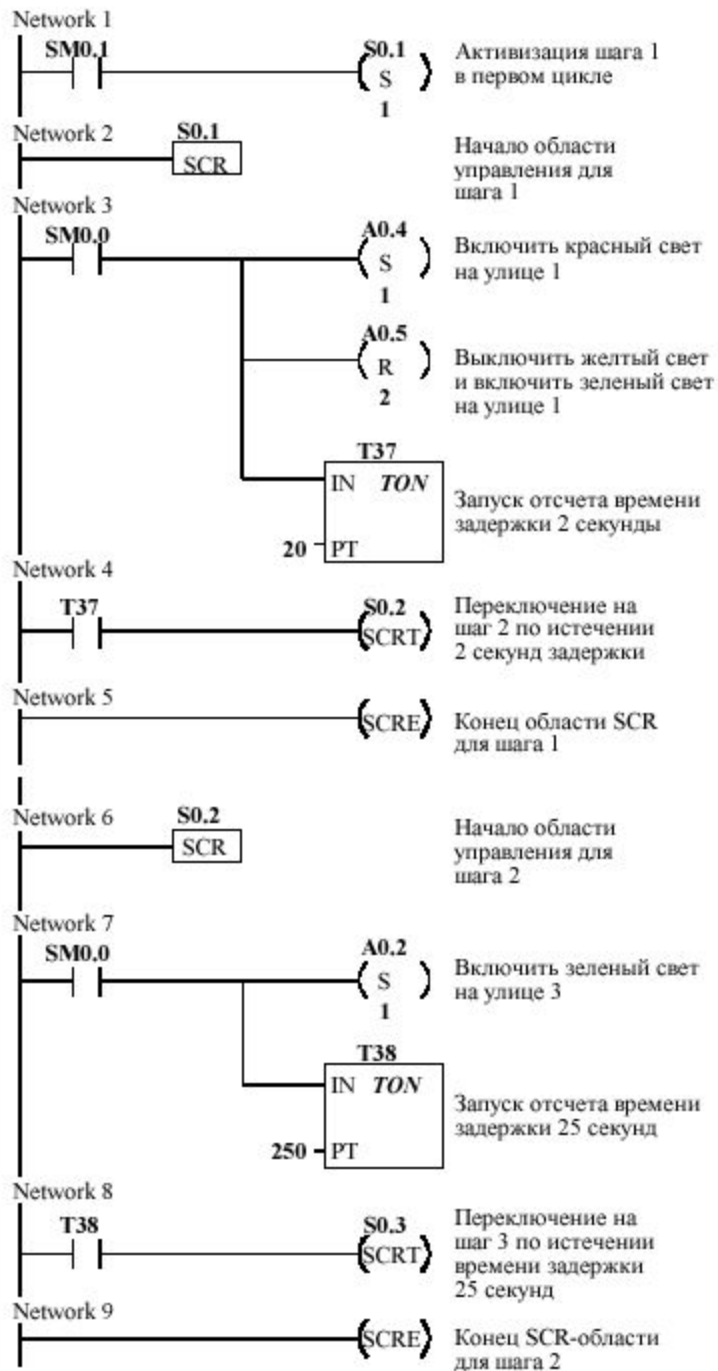


Указание

Вы можете использовать реле шагового управления в главной программе, однако их нельзя вставлять в подпрограммы или программы обработки прерываний.

С помощью операций перехода можно совершать переходы внутри сегментов SCR, а также перескакивать через сегменты SCR. Однако нельзя совершать переходы снаружи внутрь сегмента SCR или изнутри сегмента SCR наружу.

Пример программы шагового управления

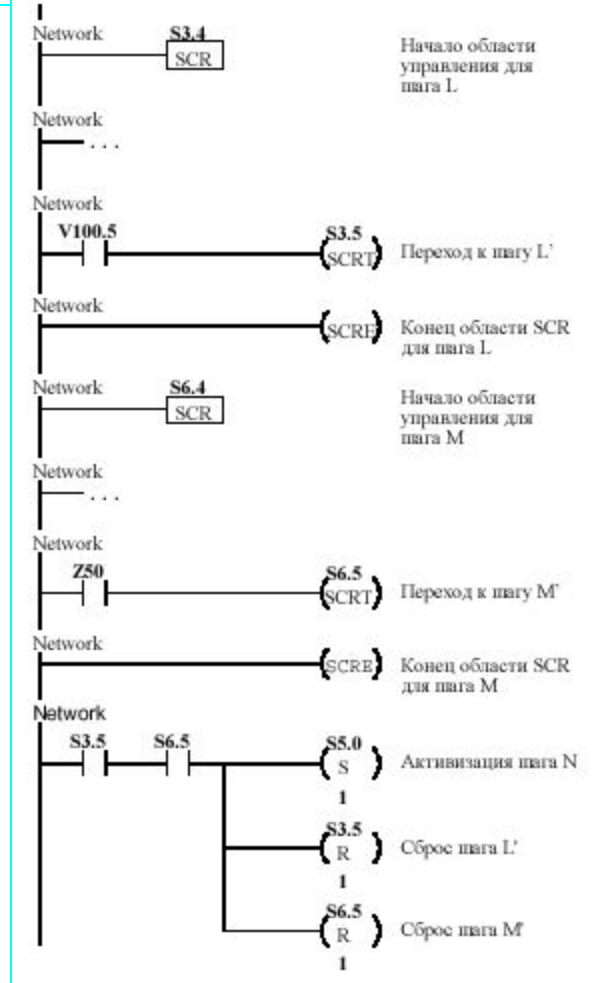
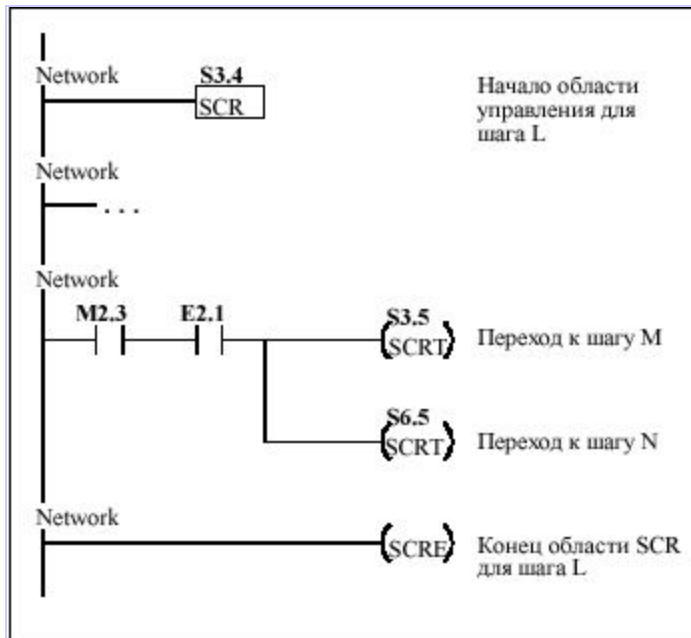
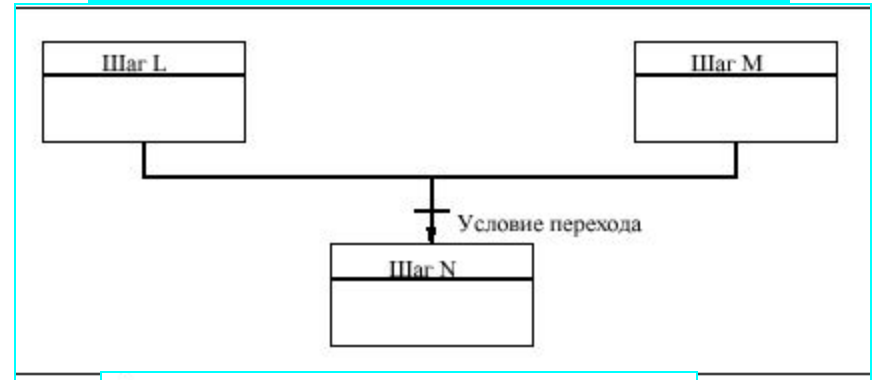


- В этом примере с помощью специального меркера SM0. (первого цикла) устанавливается S0.1. В первом цикле S0.1 находится на активном шаге 1 .
- После 2-секундной задержки T37 вызывает переключение на следующий шаг 2. Это переключение деактивирует сегмент SCR для шага 1 (S0.1) и активизирует сегмент SCR для шага 2 (S0.2).

Сведение потоков управления

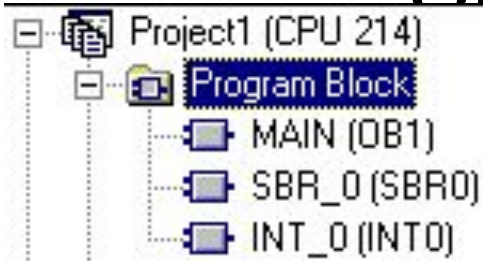
Управление потоками

Разделение потоков управления

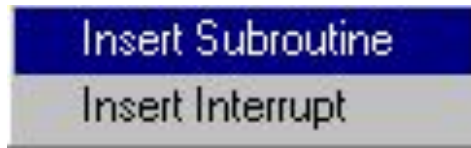


Операции управления программой.

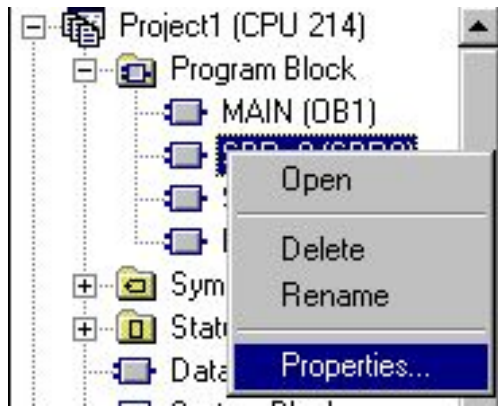
Операции управления программой



1. В разделе Program Block дерева инструкций необходимо вызвать контекстное меню (правой клавишей мыши).

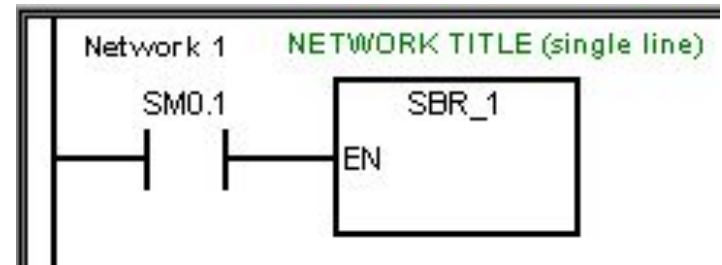


2. Выбрать команду Insert Subroutine.



3. Вы можете изменить имя подпрограммы, выбором пункта Properties в контекстном меню выбранной подпрограммы.

4. Список созданных подпрограмм отображается в разделе Subroutines дерева инструкций, и в цепочку программы, подпрограмма вставляется двойным щелчком мыши.



Номер события	Описание прерываний	CPU 212	CPU 214	CPU 215	CPU 216
0	Нарастающий фронт, E0.0*	Да	Да	Да	Да
1	Спадающий фронт, E0.0*	Да	Да	Да	Да
2	Нарастающий фронт, E0.1		Да	Да	Да
3	Спадающий фронт, E0.1		Да	Да	Да
4	Нарастающий фронт, E0.2		Да	Да	Да
5	Спадающий фронт, E0.2		Да	Да	Да
6	Нарастающий фронт, E0.3		Да	Да	Да
7	Спадающий фронт, E0.3		Да	Да	Да
8	Порт 0: Принят символ	Да	Да	Да	Да
9	Порт 0: Закончена передача	Да	Да	Да	Да
10	Управляемое временем прерывание 0	Да	Да	Да	Да
11	Управляемое временем прерывание 1		Да	Да	Да
12	HSC0: CV = PV (текущее значение = предварительно установленному значению)*	Да	Да	Да	Да
13	HSC1: CV = PV (текущее значение = предварительно установленному значению)*		Да	Да	Да
14	HSC1: Смена направления счета		Да	Да	Да
15	HSC1: Внешний сброс		Да	Да	Да

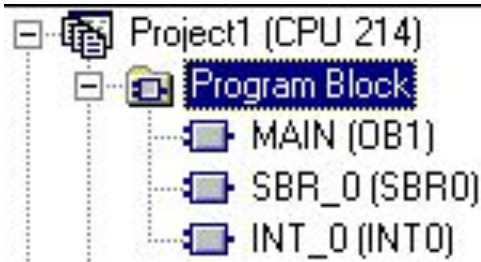
П Р Е Р Ы В А Н И Я

Номер события	Описание прерываний	CPU 212	CPU 214	CPU 215	CPU 216
16	HSC2: CV = PV (текущее значение = предварительно установленному значению)		Да	Да	Да
17	HSC2: Смена направления счета		Да	Да	Да
18	HSC2: Внешний сброс		Да	Да	Да
19	PLS0: Отсчет импульсов завершен		Да	Да	Да
20	PLS1: Отсчет импульсов завершен		Да	Да	Да
21	Таймер T32: Прерывание CT = PT			Да	Да
22	Таймер T96: Прерывание CT = PT			Да	Да
23	Порт 0: Закончен прием сообщения			Да	Да
24	Порт 1: Закончен прием сообщения				Да
25	Порт 1: Принят символ				Да
26	Порт 1: Закончена передача				Да

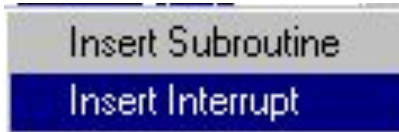
* Если событие 12 (HSC0, PV = CV) поставлено в соответствие прерыванию, то события 0 и 1 не могут быть сопоставлены никакому прерыванию. Если одно из событий 0 и 1 поставлено в соответствие прерыванию, то событие 12 не может быть сопоставлено никакому прерыванию.

П Р Е Р Ы В А Н И Я

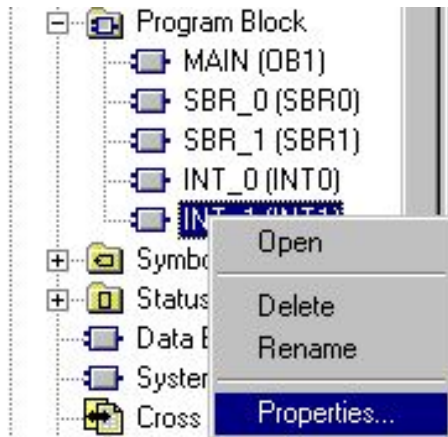
Определение программы обработки прерываний



1. В разделе Program Block дерева инструкций необходимо вызвать контекстное меню (правой клавишей мыши).



2. Выбрать команду Insert Interrupt.



3. Вы можете изменить имя программы обработки прерывания, выбором пункта Properties в контекстном меню выбранной программы.

4. Соответствие между программой обработки прерывания и событием прерывания вы устанавливаете при использовании специальной функции в структуре программы циклической обработки.

• В программах обработки прерываний нельзя использовать операции DISI, ENI, CALL, HDEF, FOR/NEXT, LSCR, SCRE, SCRT и END.

П Р Е Р Ы В А Н И Я

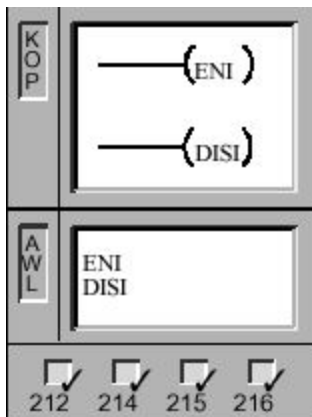
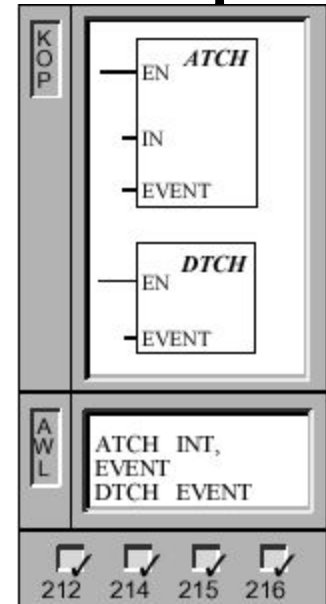
Определение прерываний в программе циклической обработки

Операция **Назначение прерывания** назначает событию прерывания (EVENT) номер программы обработки прерываний (INT) и затем разблокирует это событие .

Операция **Отделение прерывания** отделяет событие прерывания (EVENT) от всех программ обработки прерываний и затем блокирует это событие .

Операнды :INT : от 0 до 127

EVENT: от 0 до 20



Операция **Разблокировка всех событий прерываний** разблокирует обработку всех назначенных событий прерываний .

Операция **Блокировка всех событий прерываний** блокирует обработку всех событий прерываний .

Операнды : нет

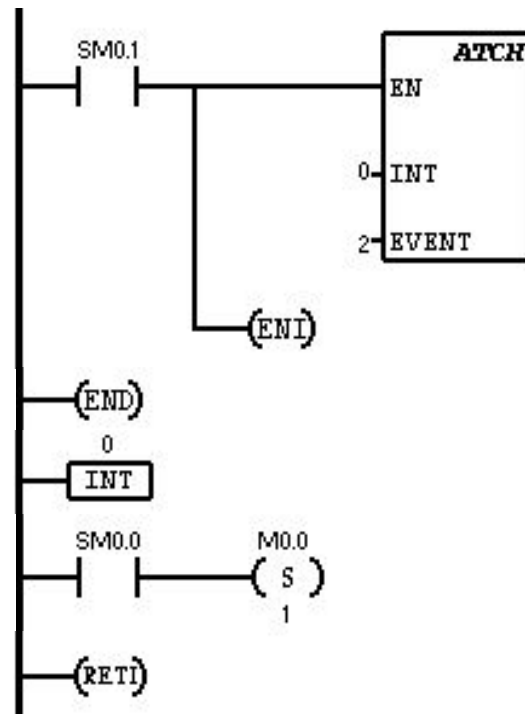
При переходе в режим RUN Вы блокируете прерывания . Если CPU находится в режиме RUN, то Вы можете с помощью операции ENI разблокировать все события прерываний . Команда “Блокировка всех событий прерываний ” допускает постановку прерываний в очередь , но не разрешает вызывать программы обработки прерываний .

Прерывания от ввода/вывода

К прерываниям от ввода / относятся прерывания при нарастающем или спадающем фронте , прерывания от быстрых счетчиков и прерывания от последовательности импульсов . CPU может создавать прерывание при нарастающем и /или спадающем фронте на входе . В таблице приведены входы , доступные для прерываний в разных CPU. События “Нарастающий фронт ” и “Спадающий фронт ” могут восприниматься по каждому из этих входов . С помощью этих событий могут также отображаться сбойные ситуации , которые должны сразу приниматься во внимание при появлении события .

Прерывания от ввода/вывода	CPU 212	CPU 214	CPU 215	CPU 216
Входы и выходы	E0.0	от E0.0 до E0.3	от E0.0 до E0.3	от E0.0 до E0.3

В данном примере используется событие прерывания 2 для установки маркера



Высокоскоростные счетчики

Определение режима для быстрых счетчиков и их активизация

Операция **Определение режима для быстрых счетчиков** назначает заданному быстрому счетчику (HSC) режим счета (MODE)

Операция **Активизация быстрых счетчиков** конфигурирует и управляет режимом работы быстрых счетчиков через сигнальные состояния битов специальных меркеров HSC. Параметр N задает номер быстрого счетчика .

Вы можете использовать для каждого счетчика только один блок HDEF.

Операнды :HSC: от 0 до 2

MODE: 0 (HSC0) от 0 до 11 (HSC1 или 2)

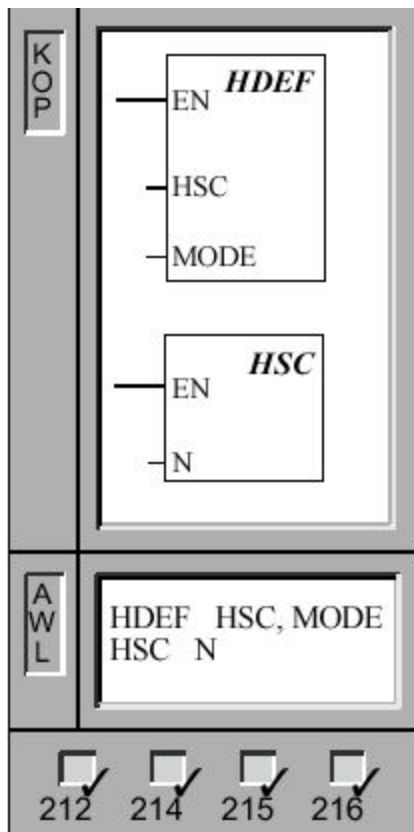
N: от 0 до 2

Быстрые счетчики подсчитывают быстрые события , которыми невозможно управлять с частотой циклической обработки программы контроллера .

- HSC0 представляет собой реверсивный счетчик , который поддерживает тактовый вход . Ваша программа управляет направлением счета (или назад) через бит управления направлением . Максимальная частота счета данного счетчика составляет 2 кГц .

- HSC1 и HSC2 представляют собой универсальные счетчики , которые можно конфигурировать согласно одному из двенадцати различных режимов счета . Различные режимы счета приведены в таблице 9–6. Максимальная частота счета счетчиков HSC1 и HSC2 определяется Вашим CPU

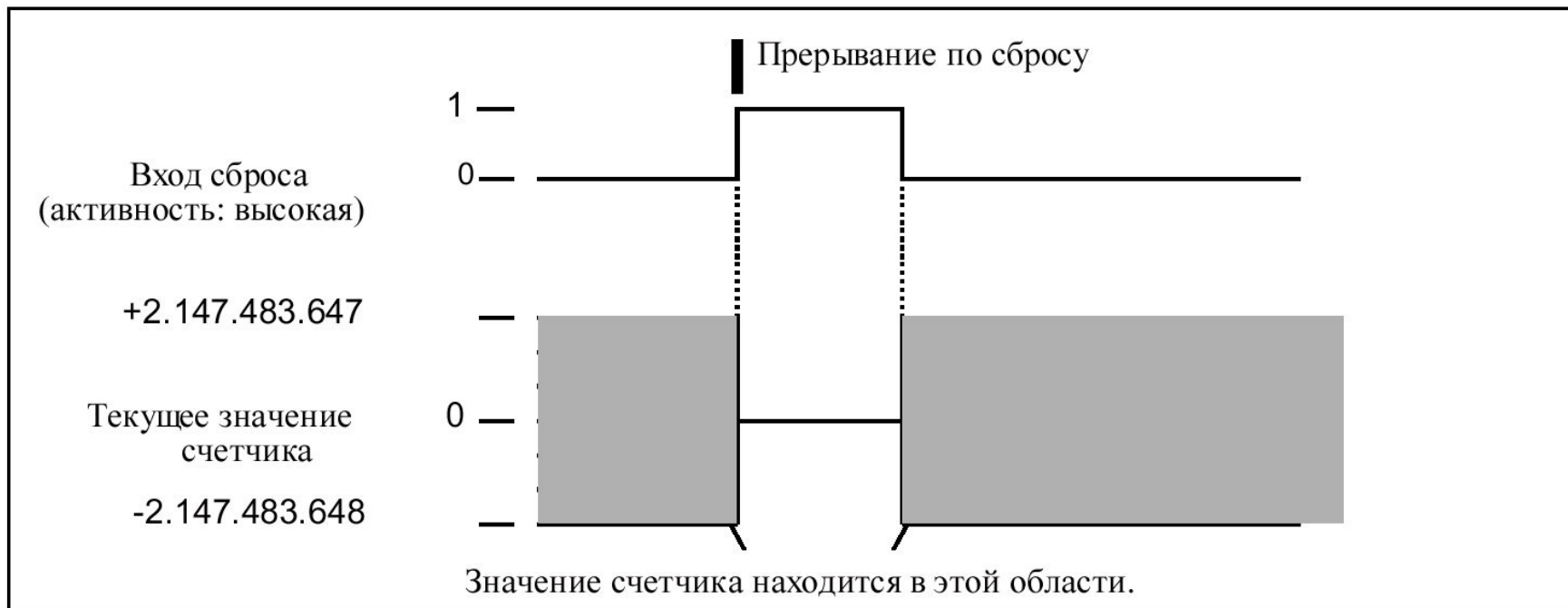
Каждый счетчик имеет в своем распоряжении особые входы , поддерживающие такие функции , как датчик тактовых импульсов , управление направлением счета , сброс и запуск . Для двухфазных счетчиков оба датчика тактовых импульсов могут работать с максимальной частотой . В случае А/В– счетчиков (квадратурных режимах) Вы можете выбирать однократную или четырехкратную скорость счета . HSC1 и HSC2 полностью не зависят друг от друга и не влияют на другие быстрые операции . Оба счетчика работают с максимальной частотой , не оказывая друг на друга отрицательного воздействия .



Работа счетчика со входом сброса и без входа запуска

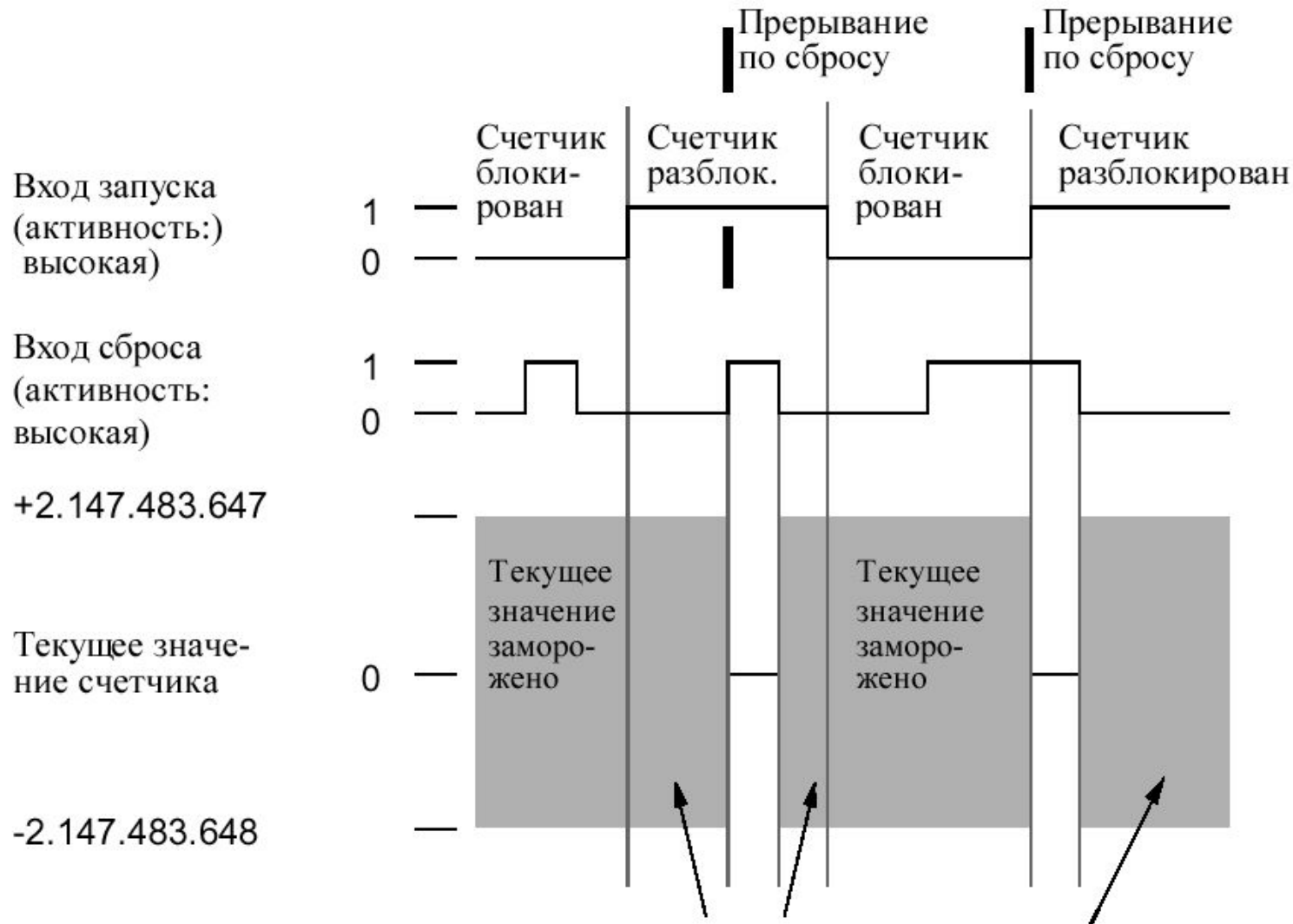
Работа счетчика со входом сброса и запуска

Работа счетчика со входом сброса и без входа запуска



назад

Работа счетчика со входом сброса и запуска

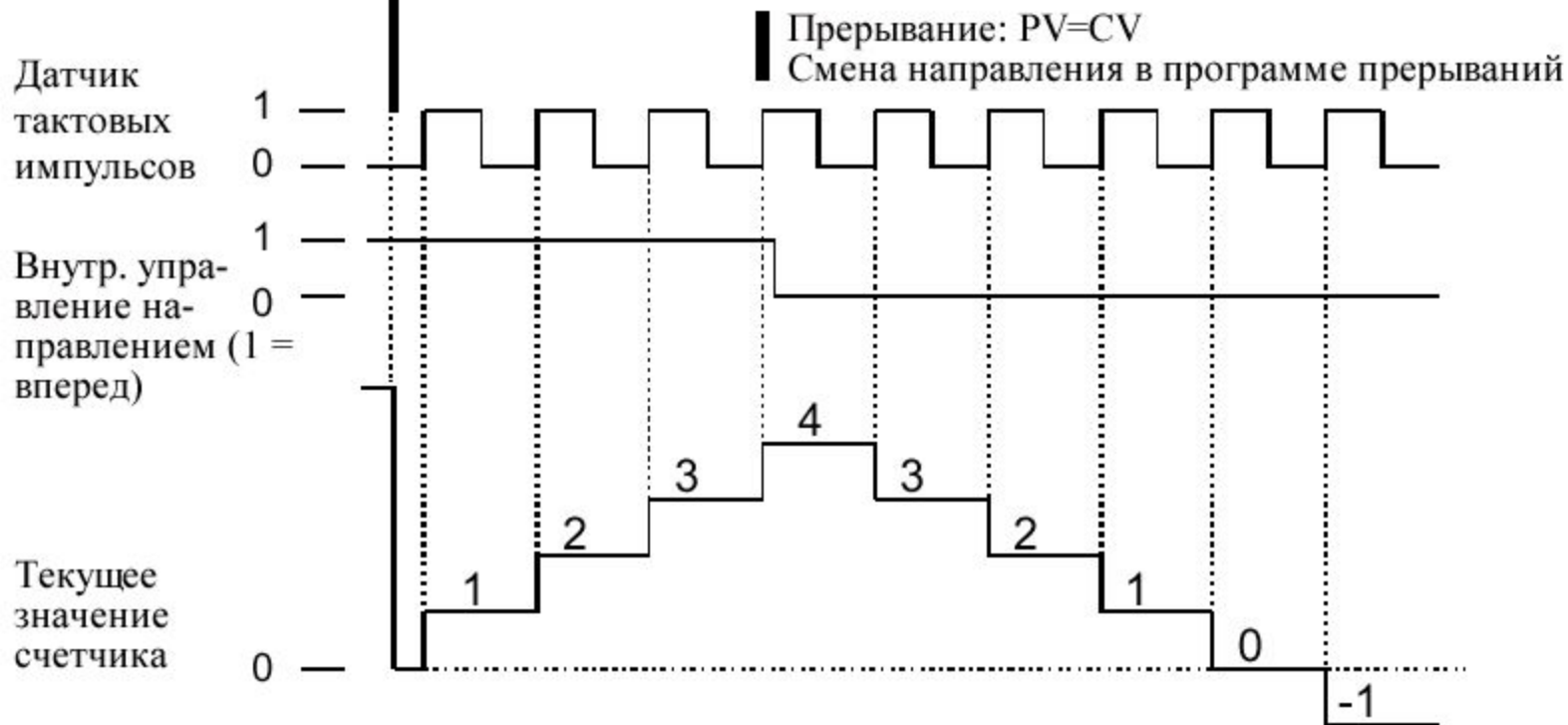


Значение счетчика находится в этой области.

Режимы быстрых счетчиков

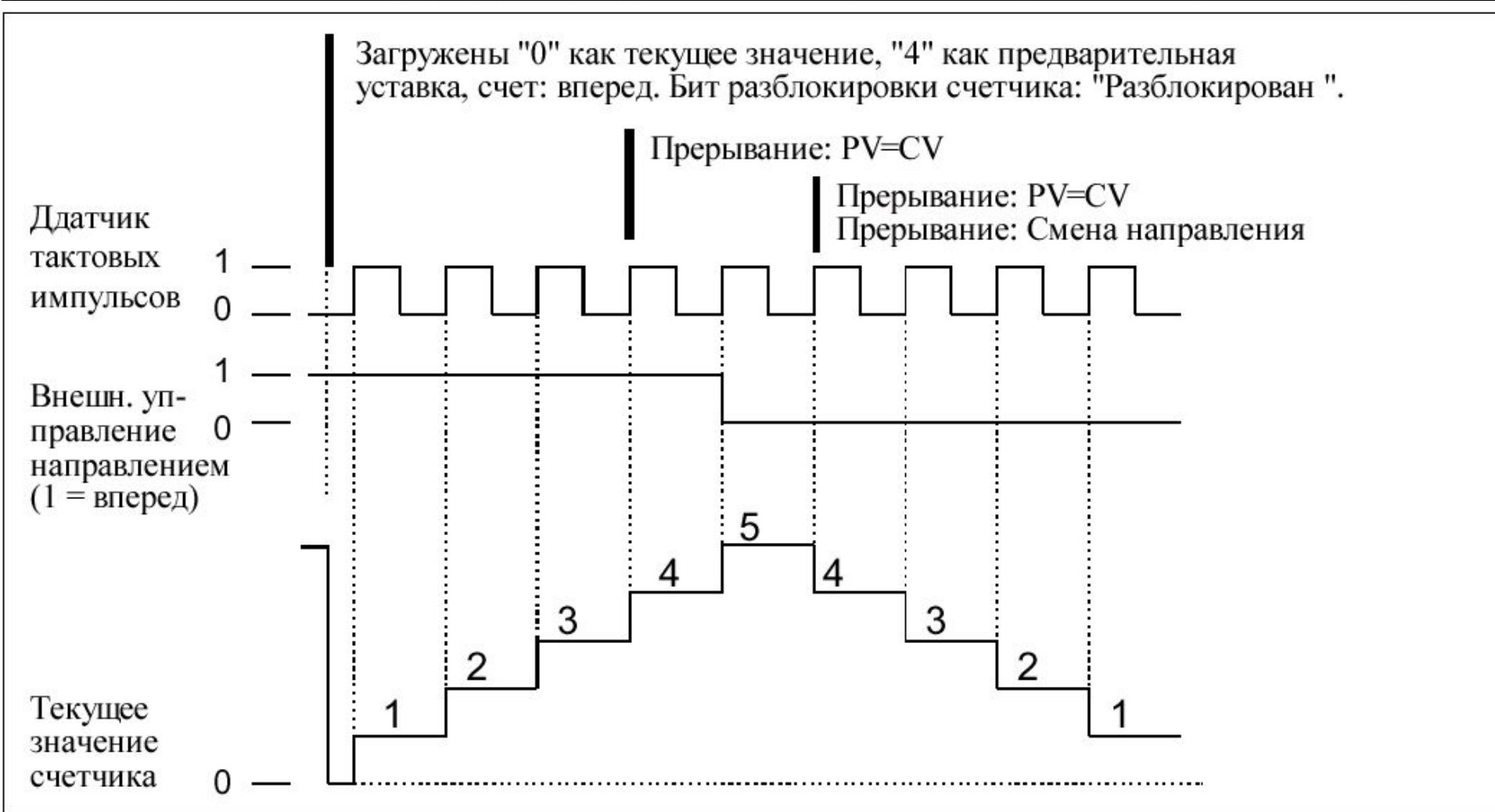
Режим	Описание	E0.0			
0	Однофазный реверсивный счетчик с внутренним управлением направлением счета SM37.3 = 0, обратный счет SM37.3 = 1, прямой счет	Датчик тактовых импульсов			
HSC1					
Режим	Описание	E0.6	E0.7	E1.0	E1.1
0	Однофазный реверсивный счетчик с внутренним управлением направлением счета	Датчик такт. имп.			
1	SM47.3 = 0, обратный счет				
2	SM47.3 = 1, прямой счет				Запуск

Загружены "0" как текущее значение, "4" как предварительная уставка, прямой счет. Бит разблокировки счетчика: "Разблокирован"



Режимы быстрых счетчиков

Режим	Описание	E0.6	E0.7	E1.0	E1.1	
3	Однофазный реверсивный счетчик с внешним управлением направлением счета	Датчик такт. имп.	Направление			
4				E0.7 = 0, обратный счет	Сброс	
5				E0.7 = 1, прямой счет		Запуск



Режимы быстрых счетчиков

Режим	Описание	E0.6	E0.7	E1.0	E1.1
6	Двухфазный счетчик с тактовыми входами для прямого и обратного счета	Датчик такт. имп. (вперед)	Датчик такт. имп. (назад)	Сброс	Запуск
7					
8					

Загружены "0" как текущее значение, "4" как предварительная уставка, счет: вперед. Бит разблокировки счетчика: "Разблокирован"

Датчик тактовых импульсов

1

Счет вперед

0

Датчик такт. имп.

1

Счет назад

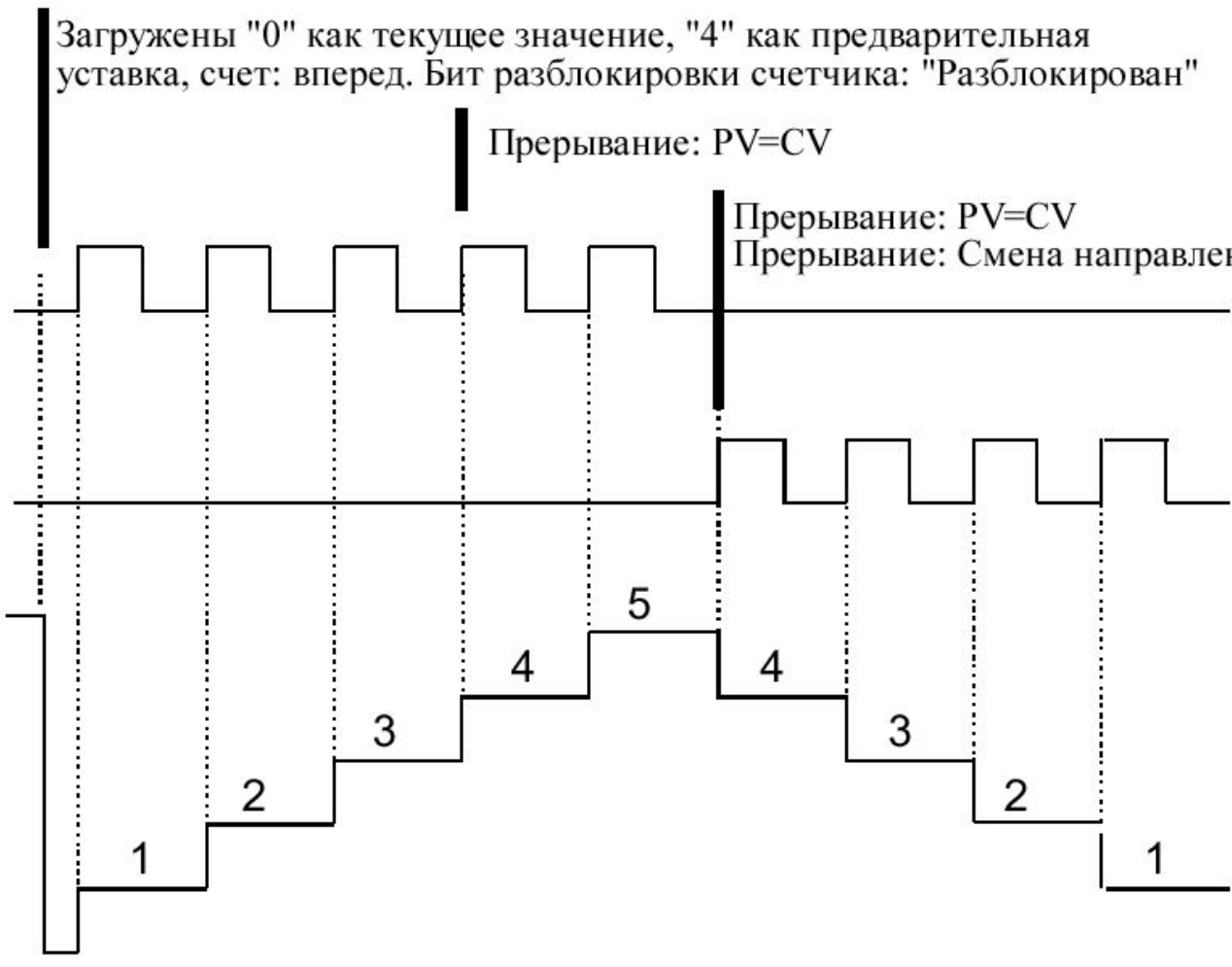
0

Текущее значение счетчика

0

Прерывание: PV=CV

Прерывание: PV=CV
Прерывание: Смена направления



Режимы быстрых счетчиков

Режим	Описание	E0.6	E0.7	E1.0	E1.1
9	A/B-счетчик Фаза А опережает В на 90 градусов по часовой стрелке	Датчик такт. имп. (фаза А)	Датчик такт. имп. (фаза В)	Сброс	Запуск
10	Фаза В опережает А на 90 градусов против часовой стрелки				
11					

Загружены "0" как текущее значение, "3" как предварительная уставка, счет: вперед. Бит разблокировки счетчика: "Разблокирован".

Датчик тактовых импульсов
Фаза А

1 —
0 —

Датчик тактовых импульсов
Фаза В

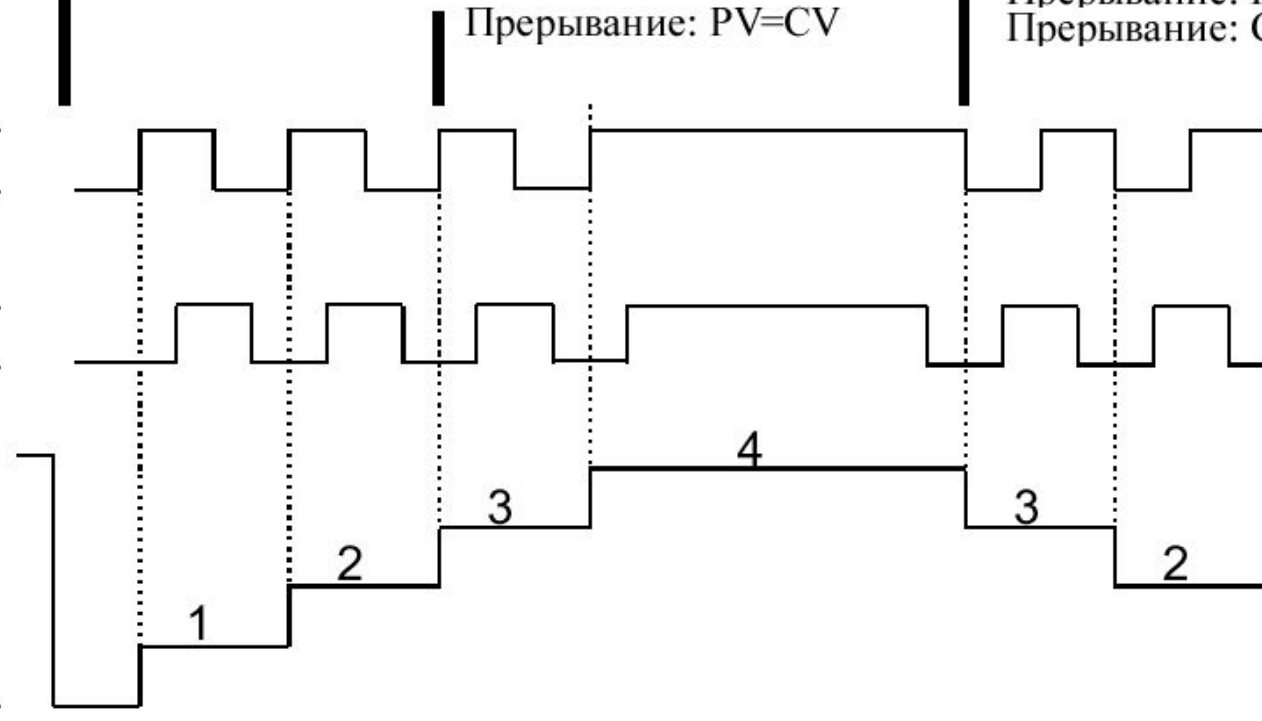
1 —
0 —

Текущее значение счетчика

0 —

Прерывание: PV=CV

Прерывание: PV=CV
Прерывание: Смена напр.



Операнды управления

HSC1 и HSC2 имеют в своем распоряжении три управляющих бита, с помощью которых можно конфигурировать активное состояние входов сброса и запуска и выбирать однократную или четырехкратную скорость счета (в А/В-счетчиках). Эти биты находятся в управляющем байте соответствующего счетчика и используются только тогда, когда выполняется операция HDEF.

HSC1	HSC2	Описание (только при исполнении HDEF)
SM47.0	SM57.0	Бит управления активностью входа сброса: 0 - высокая, 1 - низкая.
SM47.1	SM57.1	Бит управления активностью входа запуска: 0 - высокая, 1 - низкая.
SM47.2	SM57.2	Скорость счета А/В-счетчиков: 0 - четырехкратная скорость счета, 1 - однократная скорость счета.

После определения счетчика и режима счета Вы можете программировать динамические параметры счетчика. Каждый быстрый счетчик имеет управляющий байт, который активизирует или блокирует счетчик, а также устанавливает, в каком направлении должен происходить счет (режимы 0, 1 и 2). Управляющий байт задает также начальное направление счета для всех других режимов счета, а также текущее и предварительно установленное значения, которые должны загружаться. Управляющий байт и назначенные текущее и предварительно установленное значения проверяются при выполнении операции HSC.

HSC0	HSC1	HSC2	Описание
SM37.0	SM47.0	SM57.0	После выполнения HDEF не используются (в HSC0 никогда).
SM37.1	SM47.1	SM57.1	После выполнения HDEF не используются (в HSC0 никогда).
SM37.2	SM47.2	SM57.2	После выполнения HDEF не используются (в HSC0 никогда).
SM37.3	SM47.3	SM57.3	Бит управления направлением счета: 0 - назад, 1 - вперед.
SM37.4	SM47.4	SM57.4	Запись направления счета в HSC: 0 - не актуализировать, 1 - актуализировать направление.
SM37.5	SM47.5	SM57.5	Запись нового предварительно установленного значения в HSC: 0 - не актуализировать, 1 - актуализировать предварительно установленное значение.
SM37.6	SM47.6	SM57.6	Запись нового текущего значения в HSC: 0 - не актуализировать, 1 - актуализировать текущее значение.
SM37.7	SM47.7	SM57.7	Разблокировка HSC: 0 - HSC блокировать, 1 - HSC активизировать.

Операнты управления

Каждый быстрый счетчик имеет в своем распоряжении текущее значение и предварительно установленное значение размером 32 бита в каждом случае . Оба значения являются целыми числами со знаком . Чтобы загрузить новое текущее или предварительно установленное значение в быстрый счетчик , Вам нужно настроить управляющий байт и байты специальных меркеров , которые содержат текущие и /или предварительно установленные значения . Потом выполните операцию HSC, чтобы передать новые значения в быстрые счетчики . Таблица описывает байты специальных меркеров , которые содержат новые текущие и предварительно установленные значения .

Текущее значение HSC0, HSC1 и HSC2			
HSC0	HSC1	HSC2	Описание
SM38	SM48	SM58	Самый старший байт нового текущего значения (32 бита).
SM39	SM49	SM59	Второй по старшинству байт нового текущего значения (32 бита).
SM40	SM50	SM60	Третий по старшинству байт нового текущего значения (32 бита).
SM41	SM51	SM61	Младший байт нового текущего значения (32 бита).
Предварительно установленное значение HSC0, HSC1 и HSC2			
HSC0	HSC1	HSC2	Описание
SM42	SM52	SM62	Самый старший байт нового предварительно установленного значения (32 бита).
SM43	SM53	SM63	Второй по старшинству байт нового предварительно установленного значения (32 бита).
SM44	SM54	SM64	Третий по старшинству байт нового предварительно установленного значения (32 бита).
SM45	SM55	SM65	Младший байт нового предварительно установленного значения (32 бита).

Дополнительно к управляющим байтам и байтам , содержащим новые текущие и предварительно установленные значения , можно считывать текущее значение быстрого счетчика также путем задания области памяти НС (значение быстрого счетчика) и номера счетчика (0, 1 или 2). Таким способом , Вы можете считывать текущее значение непосредственно . Однако для записи Вам нужно использовать описанную выше операцию HSC.

Операнды управления

Каждый быстрый счетчик имеет байт состояния , предоставляющий в распоряжение меркеры состояния . Эти биты состояния задают текущее направление счета . Кроме того они указывают , является ли текущее значение равным предварительно установленному значению или превышает его . Таблица описывает биты состояния быстрых счетчиков .

HSC0	HSC1	HSC2	Описание
SM36.0	SM46.0	SM56.0	Не используются.
SM36.1	SM46.1	SM56.1	Не используются.
SM36.2	SM46.2	SM56.2	Не используются.
SM36.3	SM46.3	SM56.3	Не используются.
SM36.4	SM46.4	SM56.4	Не используются.
SM36.5	SM46.5	SM56.5	Бит состояния: Текущее направление счета: 0 - назад, 1 - вперед.
SM36.6	SM46.6	SM56.6	Бит состояния: Текущее значение равно предварительно установленному значению: 0 - не равно, 1 - равно.
SM36.7	SM46.7	SM56.7	Бит состояния: Текущее значение больше предварительно установленного значения: 0 - меньше или равно, 1 - больше.

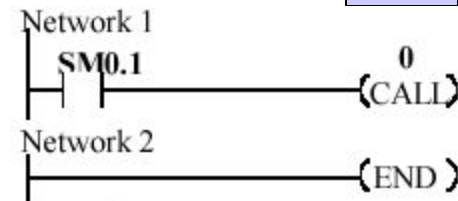
Биты состояния HSC0, HSC1 и HSC2 действительны только во время обработки программы прерываний для быстрых счетчиков . Когда Вы контролируете состояния быстрых счетчиков Вы можете разблокировать прерывания для событий , влияющих на обрабатываемую операцию .

Пример инициализация счетчика

Чтобы инициализировать HSC1 например как однофазный реверсивный счетчик с внешним управлением направлением счета (режим 3, 4 или 5), действуйте следующим образом :

Реж
им
3,4,5

1. Вызовите с помощью меркера первого цикла подпрограмму , в которой выполняется инициализация . Если Вы вызываете эту подпрограмму , то последующие циклы ее больше не вызывают , за счет чего сокращается время цикла и программа имеет более наглядную структуру .



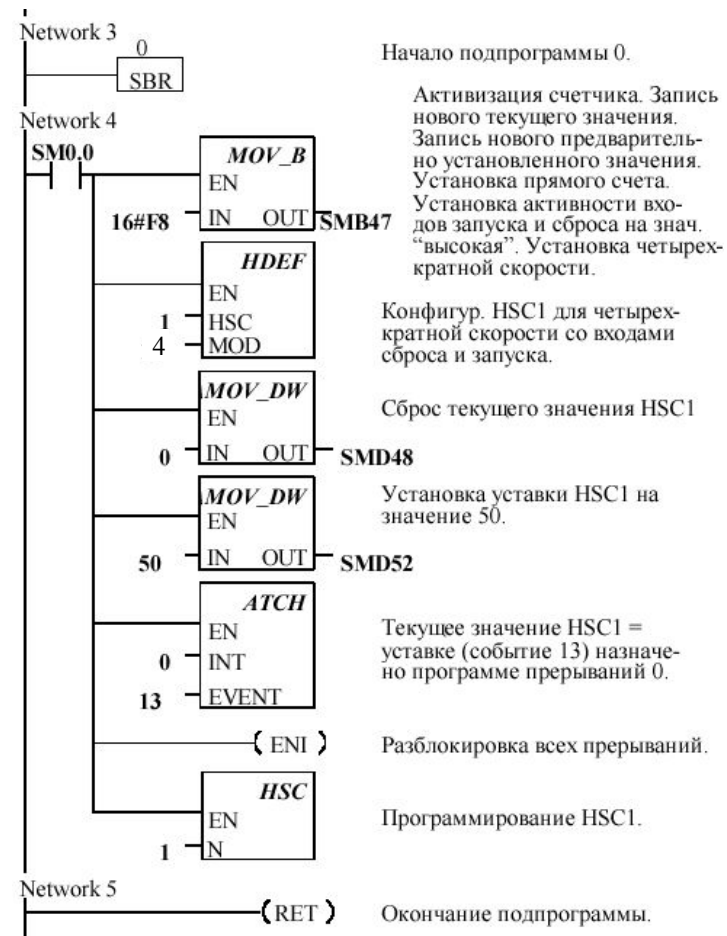
2. В подпрограмме инициализации загрузите SM47 желаемыми установками . Например :

SM47 = 16#F8, активизирует счетчик записывает новое текущее значение записывает новое предварительно установленное значение устанавливает прямой счет устанавливает активность входов запуска и сброса на значение “высокая”.

3. Выполните операцию HDEF. При этом вход HSC установлен в “1”, а вход MODE установлен либо в “3”, если нет внешнего сброса или запуска , либо в “4”, если есть внешний сброс и нет запуска , либо в “5”, если есть внешний сброс и запуск .

4. Загрузите желаемое текущее значение в SM48 (слово). (Если Вы загружаете значение “0”, то меркер сбрасывается).

5. Загрузите желаемое предварительно установленное значение в SM52 (слово).



6. Запрограммируйте прерывание , назначив событие прерывания PV = CV (13) соответствующей программе обработке

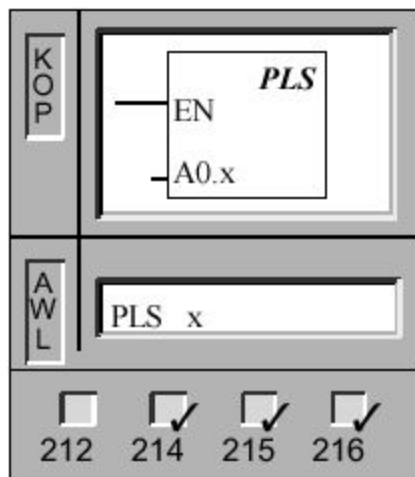
Операции с быстрыми выходами

Некоторые CPU могут через выходы A0.0 и A0.1 либо порождать быстрые последовательности импульсов (**PTO** = pulse train output), либо управлять широтно –импульсной модуляцией (**PWM** = pulse width modulation).



Функция **PTO** обеспечивает на выходе последовательность прямоугольных импульсов (относительная длительность включения 50%) с определенным количеством импульсов и фиксированным периодом следования .

Количество импульсов может лежать в диапазоне от 1 до 4.294.967.295. Период может задаваться в микросекундах (250 до 65.535) или в миллисекундах (2 до 65.535). Нечетное число микросекунд или миллисекунд вызывает искажение относительной длительности включения .



Функция **PWM** обеспечивает фиксированный период следования импульсов с переменной относительной длительностью включения . Период следования и длительность импульсов могут задаваться в микро - или миллисекундах . Период следования лежит в диапазоне от 250 до 65.535 микросекунд или в диапазоне от 2 до 65.535 миллисекунд . Длительность импульсов лежит в диапазоне от 0 до 65.535 микросекунд или в диапазоне от 0 до 65.535 миллисекунд . Если длительность импульсов и период следования равны , то относительная длительность включения составляет 100%, и выход является постоянно включенным . Если длительность импульсов равна нулю , то относительная длительность включения составляет 0%, и выход выключается . Если период следования задается меньшим , чем две единицы времени , то период следования принимает значение по умолчанию , равное двум единицам времени.

Операция **Вывод импульсов** проверяет специальные меркеры для этого импульсного выхода (x). Потом вызывается импульсная операция , определенная в специальных меркерах .

Операнды :x: от 0 до 1

Специальные меркеры быстрых выходов

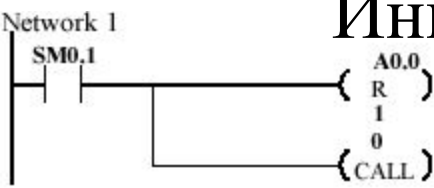
A 0.0	A 0.1	Биты состояния для импульсных выходов
SM66.6	SM76.6	Переполнение конвейера РТО - нет переполнения, 1 - переполнение 0
SM66.7	SM76.7	Холостой ход РТО обработка, 1 - холостой ход 0 -
A 0.0	A 0.1	Биты управления для выходов РТО/PWM
SM67.0	SM77.0	РТО/PWM: актуализация периода следования 0 - нет актуализации, 1 - актуализация периода следования
SM67.1	SM77.1	PWM: актуализация длительности импульсов 0 - нет актуализации, 1 - актуализация длительности импульсов
SM67.2	SM77.2	РТО: актуализация значения счетчика импульсов актуализации, 1 - актуализация счетчика импульсов 0 - нет
SM67.3	SM77.3	РТО/PWM: выбор базы времени 0 - 1 мкс/такт; 1 - 1 мс/такт
SM67.4	SM77.4	Не используется.
SM67.5	SM77.5	Не используется.
SM67.6	SM77.6	РТО/PWM: выбор функции 0 - РТО; 1 - PWM
SM67.7	SM77.7	РТО/PWM: разблокировка блокировка РТО/PWM, 1 - разблокировка РТО/PWM 0 -
A 0.0	A 0.1	Значения периода следования для выходов РТО/PWM (диапазон: 2 - 65.535)
SM68	SM78	Старший байт периода следования РТО/PWM
SM69	SM79	Младший байт периода следования РТО/PWM
A 0.0	A 0.1	Значения длительности импульсов для выходов PWM (диапазон: 0 - 65.535)
SM70	SM80	Старший байт длительности импульсов PWM
SM71	SM81	Младший байт длительности импульсов PWM
A 0.0	A 0.1	Значения счетчика импульсов для импульсных выходов (диапазон: 1 - 4.294.967.295)
SM72	SM82	Самый старший байт значения счетчика импульсов РТО
SM73	SM83	Второй по старшинству байт значения счетчика импульсов РТО
SM74	SM84	Третий по старшинству байт значения счетчика импульсов РТО
SM75	SM85	Младший байт значения счетчика импульсов РТО

Специальные меркеры быстрых выходов

Управляющий регистр (16-ричное значение)	Результат выполнения операции PLS					
	Разблокировка	Функция	База времени	Значение счетчика импульсов	Длительность импульсов	Период следования
16#81	Да	PТО	1 мкс/такт			Загрузка
16#84	Да	PТО	1 мкс/такт	Загрузка		
16#85	Да	PТО	1 мкс/такт	Загрузка		Загрузка
16#89	Да	PТО	1 мс/ такт			Загрузка
16#8C	Да	PТО	1 мс/ такт	Загрузка		
16#8D	Да	PТО	1 мс/ такт	Загрузка		Загрузка
16#C1	Да	PWM	1 мкс/такт			Загрузка
16#C2	Да	PWM	1 мкс/такт		Загрузка	
16#C3	Да	PWM	1 мкс/такт		Загрузка	Загрузка
16#C9	Да	PWM	1 мс/ такт			Загрузка
16#CA	Да	PWM	1 мс/ такт		Загрузка	Загрузка
16#CB	Да	PWM	1 мс/ такт		Загрузка	Загрузка



Инициализация выхода в режиме PTO

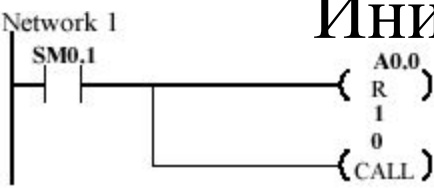


1. Сбросьте выход в “0” с помощью меркера первого цикла и вызовите подпрограмму , в которой выполняется инициализация .

2. Загрузите в подпрограмме инициализации значение 16#8D в SM67, указывая этим для функции PTO, что инкрементирование должно производиться в миллисекундах. Это значение устанавливает управляющий байт , с помощью которого разблокируется операция PTO/PWM и выбирается функция PTO. Кроме того , управляющий байт указывает , что инкрементирование производится в миллисекундах и что должны актуализироваться значения счетчика и периода следования импульсов .
3. Загрузите желаемый период следования в SM68.
4. Загрузите желаемое количество импульсов в SM72.
5. Этот шаг является необязательным : если Вы хотите после выполнения операции “ импульсов ” выполнить поставленную в соответствие функцию , то можете запрограммировать прерывание , сопоставляя событие “ импульсов закончена ” (прерываний 19) программе обработки прерываний и выполняя операцию “Разблокировка всех событий прерываний ”.
6. Выполните операцию PLS, чтобы S7-200 запрограммировал генератор PTO/PWM.
7. Закончите подпрограмму .

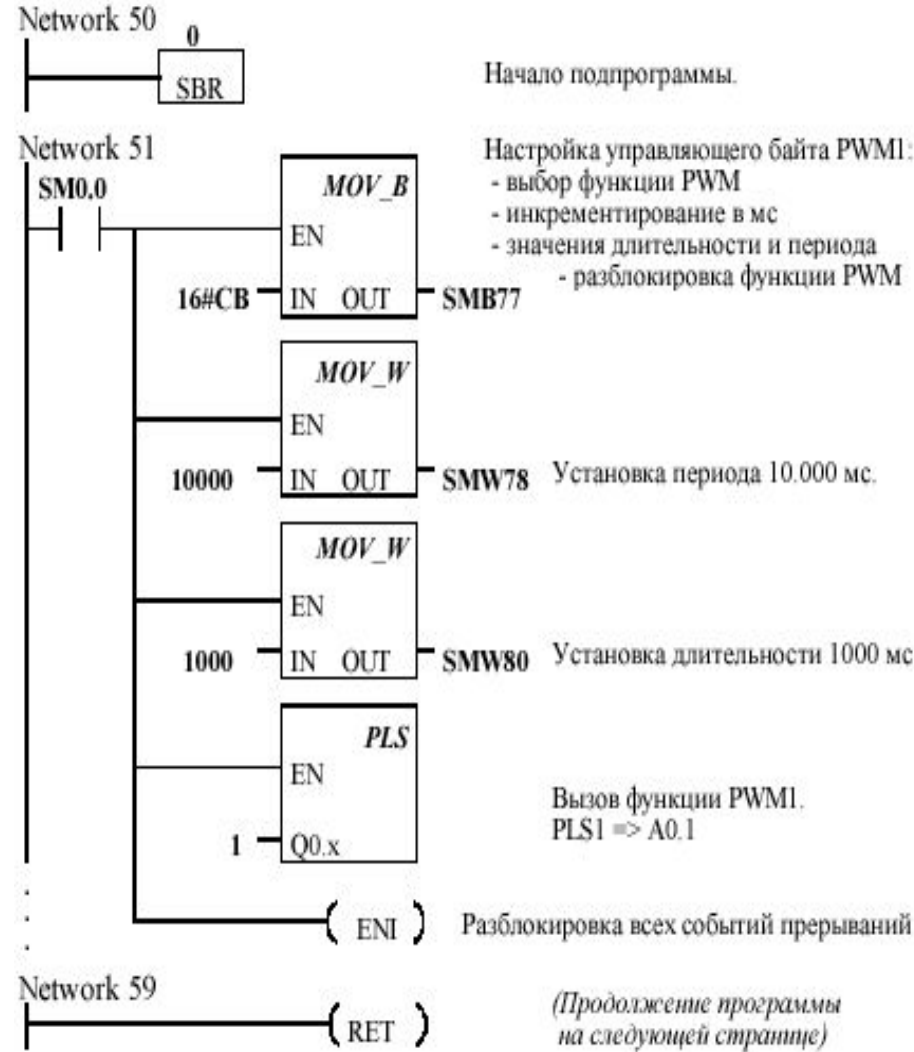


Инициализация выхода в режиме PWM



1. Сбросьте выход в “0” с помощью меркера первого цикла и вызовите подпрограмму , в которой выполняется инициализация .

2. Загрузите значение 16# CB в SM77 в подпрограмме инициализации , указывая этим для функции PWM, что инкрементирование должно производиться в миллисекундах. Это значение устанавливает управляющий байт , с помощью которого разблокируется операция PTO/PWM и выбирается функция PWM. Кроме того , управляющий байт указывает , что инкрементирование производится в миллисекундах и что должны актуализироваться значения для длительности импульса и периода следования .
3. Загрузите желаемый период следования в SM78.
4. Загрузите желаемую длительность импульса в SM80.
5. Выполните операцию PLS, чтобы S7-200 запрограммировал генератор PTO/PWM.
6. Закончите подпрограмму .



Прерывания управляемые временем

S7-200 может поддерживать одно или несколько прерываний, управляемых временем. С помощью прерываний, управляемых временем, Вы можете определять действия, которые должны выполняться периодически. Период задается с шагом 1 мс, значения лежат в диапазоне от 5 мс до 255 мс. Период для управляемого временем прерывания 0 запишите в SMB34, период для управляемого временем прерывания 1 запишите в SMB35.

Прерывания, управляемые временем	CPU 212	CPU 214	CPU 215	CPU 216
Количество поддерживаемых прерываний, управляемых временем	1	2	2	2

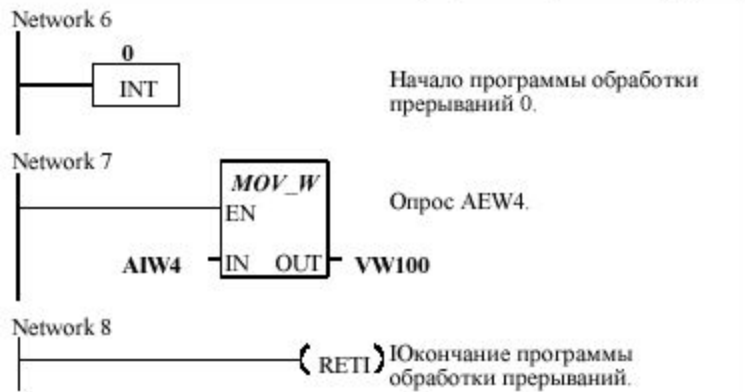
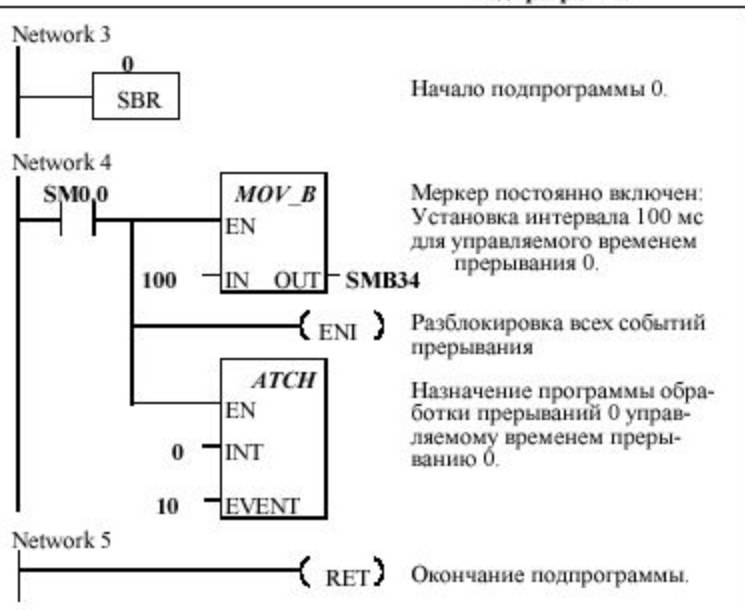
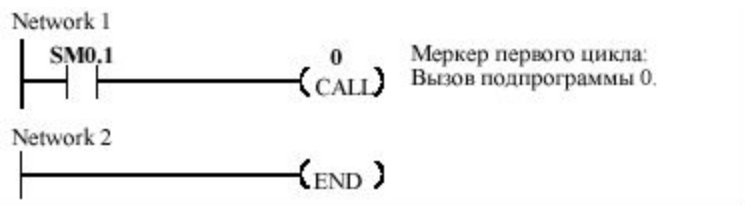
Управляемое временем событие прерывания вызывает соответствующую программу обработки прерываний каждый раз, когда истекает время. В общем случае с помощью управляемых временем событий прерываний Вы управляете регулярным опросом аналоговых входов. Управляемое временем прерывание разблокируется и время начинает отсчитываться, когда Вы назначаете программу обработки прерываний управляемому временем событию прерывания. При этом система фиксирует период, чтобы последующие изменения не влияли на период. Если Вы хотите изменить период, то Вам нужно задать новое значение для периода и затем снова назначить программу обработки прерываний управляемому временем событию прерывания. При новом назначении эта функция стирает накопленное значение времени предыдущего назначения, и время начинает отсчитываться с новым значением периода. После разблокировки прерывание, управляемое временем, функционирует непрерывно и обрабатывается каждый раз, когда истекает заданный интервал времени. Если Вы выходите из режима RUN или отделяете прерывание от программы обработки прерываний (DTCH), то управляемое временем прерывание блокируется. Если Вы выполняете операцию “Блокировка всех событий прерываний”, то управляемые временем прерывания в дальнейшем хотя и появляются, однако ставятся в очередь (тех пор, пока либо прерывания снова не разблокируются, либо очередь не переполнится).

Пример считывания состояния аналогового входа

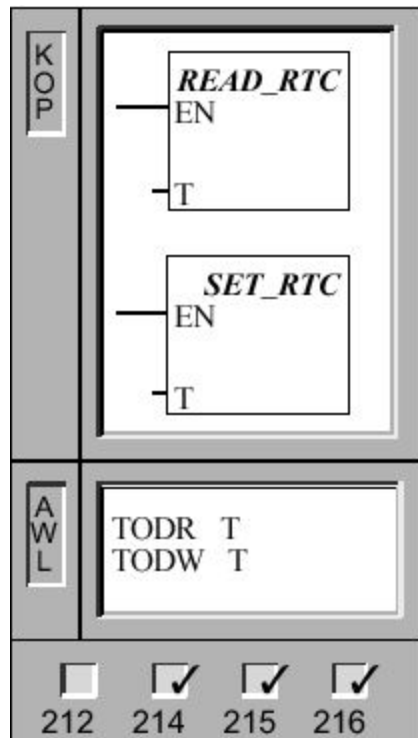
1. Под управлением бита первого скана, вызывается подпрограмма инициализации управляемого временем прерывания

2. Командой перемещения устанавливается временной интервал 100 мс. Разблокируются все прерывания, и событию 10 присваивается программа обработки прерывания с номером 0.

3. Программа обработки прерывания будет исполняться с периодичностью 100 мс., тем самым считывая значение аналогового входа в память VW100



Операции с часами реального времени



Операция **Чтение часов реального времени** считывает текущее время суток и текущую дату из часов реального времени и загружает их в 8–байтный буфер (адрес T).

Операция **Запись в часы реального времени** записывает текущее время суток и текущую дату , загруженные в 8–байтный буфер (адресT), в часы реального времени .

В AWL операции Read_RTC и Set_RTC представляются посредством мнемокодов TODR (чтение часов реального времени) и TODW (запись в часы реального времени).

Операнды : T: VB, EB, AB, MB, SMB, *VD, *AC

Вы должны кодировать дату и время в BCD–формате (например, 16#97 для года 1997). Используйте для этого следующие форматы данных :

год / jjmm jj - 0 - 99 mm - 1 - 12

день / tthh tt - 1 - 31 hh - 0 - 23

минут / mmss mm - 0 - 59 ss - 0 - 59

день недели 000t t - 0 - 7 1 = воскресенье

0 = день недели выключается