

## Обзор главы

В разделе	Вы найдете	на стр.
11.1	Логические операции с битовыми операндами	11–2
11.2	Битовые логические операции и релейно-контактные схемы	11–6
11.3	Анализ условий с помощью И, ИЛИ и исключающего ИЛИ	11–10
11.4	Логические операции со скобочными выражениями и И перед ИЛИ	11–13
11.5	Операции оценки фронтов: FP, FN	11–16
11.6	Завершение цепи логических операций	11–20
11.7	Операции установки и сброса: S и R	11–21
11.8	Операция присваивания (=)	11–24
11.9	Отрицание, установка, сброс и сохранение VKE	11–26

## 11.1 Логические операции с битовыми операндами

### Объяснение

Для логических операций с битовыми операндами существуют следующие основные операции:

- U (И) и ее форма с отрицанием UN (И-НЕ).
- O (ИЛИ) и ее форма с отрицанием ON (ИЛИ-НЕ).
- X (ИСКЛЮЧАЮЩЕЕ ИЛИ) и ее форма с отрицанием XN (ИСКЛЮЧАЮЩЕЕ ИЛИ - НЕ).

Эти операции выполняют следующие основные функции:

- Опрос состояния битового операнда на равенство "1" (активизирован, включен) или на равенство "0" (не активизирован, выключен).
- Опрос состояния таймера (Timer) или счетчика на равенство "0" (значение ячейки = 0) или на равенство "1" (значение ячейки > 0).

Результат логической операции образуется в зависимости от значения /ER–бита:

- Если /ER = "0", то результат опроса состояния оставляется в ячейке результата логической операции без изменения (начало цепи логических операций).
- Если /ER = "1", то результат опроса состояния сопрягается логически в соответствии с таблицей истинности логической операции (U, O, X) и откладывается в ячейке результата логической операции.

**Таблица истинности** Результат логической операции в начале цепи логических операций может определяться на основании следующей таблицы истинности:

**в начале цепи логических операций**

Мнемоника	Операция	Состояние операнда	Результат в VKE
U	И	0 1	0 1
UN	И-НЕ	0 1	1 0
O	ИЛИ	0 1	0 1
ON	ИЛИ-НЕ	0 1	1 0
X	ИСКЛЮЧАЮЩЕЕ ИЛИ	0 1	0 1
XN	ИСКЛЮЧАЮЩЕЕ ИЛИ - НЕ	0 1	1 0

**Таблица истинности** Результат логической операции, начиная со второй логической операции **внутри цепи** над битами, может определяться на основании следующей таблицы истинности:  
**логических операций**

Мнемоника	Операция	VKE перед выполне- нием	Состоя- ние операнда	Результат в VKE
U	И	0	0	0
		0	1	0
		1	0	0
		1	1	1
UN	И-НЕ	0	0	0
		0	1	0
		1	0	1
		1	1	0
O	ИЛИ	0	0	0
		0	1	1
		1	0	1
		1	1	1
ON	ИЛИ-НЕ	0	0	1
		0	1	0
		1	0	1
		1	1	1
X	ИСКЛЮЧАЮЩЕЕ ИЛИ	0	0	0
		0	1	1
		1	0	1
		1	1	0
XN	ИСКЛЮЧАЮЩЕЕ ИЛИ - НЕ	0	0	1
		0	1	0
		1	0	0
		1	1	1

**Операнды  
основных  
операций (U, UN,  
O, ON, X, XN)**

Операнд, опрашиваемый операциями, может быть битом, таймером или счетчиком. Операция обращается к контакту посредством одного из следующих видов операндов:

- Признак операнда и адрес внутри области памяти, задаваемой этим признаком операнда (смотрите таблицы с 11–1 по 11–3).
- Бит, таймер или счетчик как параметр (смотрите таблицу 11–4).
- Условие, выраженное через биты слова состояния (смотрите таблицу 11–8).

Таблица 11–1. Операнды: прямая и косвенная адресация

Признак операнда	Максимальный диапазон адресов в зависимости от способа адресации				
	прямая	косвенная через память		косвенная регистровая, внутри области	
E A	от 0.0 до 65 535.7	[DBD] [DID] [LD] [MD]	от 0 до 65 532	[AR1, P#байт.бит] [AR2, P#байт.бит]	от 0.0 до 8 191.7
M	от 0.0 до 65 535.7	[DBD] [DID] [LD] [MD]	от 0 до 65 532	[AR1, P#байт.бит] [AR2, P#байт.бит]	от 0.0 до 8 191.7
DBX DIX L	от 0.0 до 65 535.7	[DBD] [DID] [LD] [MD]	от 0 до 65 532	[AR1, P#байт.бит] [AR2, P#байт.бит]	от 0.0 до 8 191.7

Таблица 11–2. Операнды: косвенная регистровая адресация с указанием области памяти

Признак операнда 1)	Максимальный диапазон адресов
E, A, M, DBX, DIX или L	[AR1, P#байт.бит] от 0.0 до [AR2, P#байт.бит] 8 191.7

1) Область памяти закодирована в битах указателя 24, 25 и 26 (см. гл. 2.8).

Таблица 11–3. Операнды: таймеры и счетчики

Признак операнда	Максимальный диапазон адресов в зависимости от способа адресации	
	прямая	косвенная через память
T Z	от 0 до 65535	[DBW] от 0 □□□ до 65 [DIW] 534 [LW] [MW]

Таблица 11–4. Операнды: биты, таймеры или счетчики, передаваемые как параметры

Операнд	Формат адресного параметра
Символическое имя	Бит, таймер или счетчик, передаваемый как параметр.

Таблица 11–5. Операнды битовых логических операций: биты слова состояния

Диапазон адресов или ссылка на адрес	Биты слова состояния
>0, <0, <=0, >=0, <=0, ==0	7 и 6: индикаторные биты (область памяти)
UO	7 и 6: индикаторные биты (область памяти) □
BIE	8: двоичный результат (адрес)
OV	5: переполнение (адрес)
OS	4: переполнение, с запоминанием (адрес)

## Изменение битов слова состояния

Операция	OR	STA	VKE	/ER
U	x	x	x	1
UN	x	x	x	1
U(	0	1	-	0
UN(	0	1	-	0
O	0	x	x	1
ON	0	x	x	1
O(	0	1	-	0
ON(	0	1	-	0
X	0	x	x	1
XN	0	x	x	1
X(	0	1	-	0
XN(	0	1	-	0
=	0	x	-	0
CLR	0	0	0	0
FN	0	x	x	1
FP	0	x	x	1
NOT	-	1	x	-
R	0	x	-	0
S	0	x	-	0
SAVE	-	-	-	-
SET	0	1	1	0

## 11.2. Битовые логические операции и релейно-контактные схемы

### Введение

Логические операции над битами называются также релейными операциями, так как с их помощью можно типовым образом реализовать такие последовательности команд, которые заменяют релейные схемы. В дальнейшем объясняется то, как с помощью последовательностей команд в форме AWL могут воспроизводиться типичные релейные схемы.

### Замыкающий контакт

На рис. 11-1 представлена релейно-контактная схема с замыкающим контактом между токовой шиной и катушкой. В нормальном состоянии контакт разомкнут. Пока контакт не активизирован, он остается разомкнутым. Состояние сигнала разомкнутого контакта равно "0" (не активизирован). Пока контакт остается разомкнутым, ток от токовой шины не может протекать к катушке в конце схемы. Когда контакт активизируется (состояние сигнала контакта равно "1"), ток протекает к катушке.

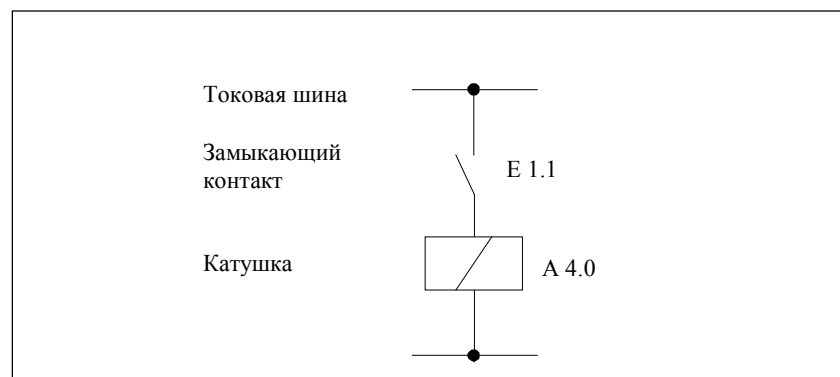


Рис. 11-1. Релейная схема с замыкающим контактом

С помощью операции U (И) или O (ИЛИ) Вы можете опросить состояние сигнала замыкающего контакта. Если замыкающий контакт разомкнут ( $E\ 1.1 = 0$ ), то результат опроса равен "0", а если он замкнут, то результат опроса равен "1".

**Размыкающий  
контакт**

На рис. 11–2 представлена релейно-контактная схема с размыкающим контактом между токовой шиной и катушкой. В нормальном состоянии контакт замкнут. Пока контакт не активизирован, он остается замкнутым. Состояние сигнала замкнутого контакта равно "0" (не активизирован). Пока контакт остается замкнутым, ток от токовой шины может протекать через контакт, и катушка в конце схемы проводит ток. При активизации контакта (состояние сигнала контакта равно "1") контакт размыкается, и поток сигнала к катушке прерывается.

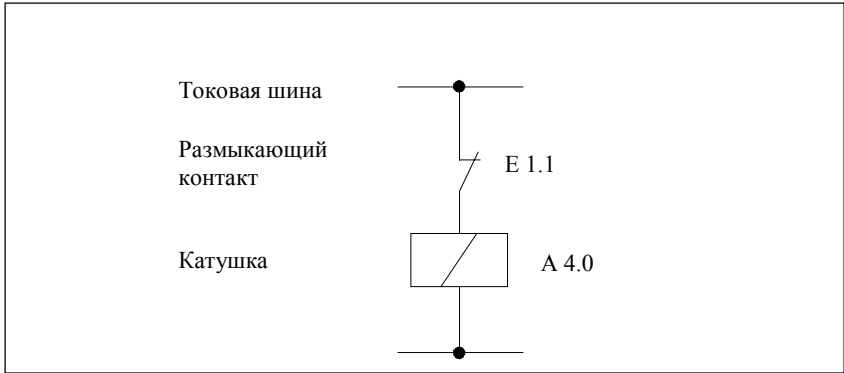


Рис. 11-2. Релейная схема с размыкающим контактом

С помощью операции UN (И-НЕ) или ON (ИЛИ-НЕ) Вы можете опросить состояние сигнала размыкающего контакта. Если размыкающий контакт замкнут ( $E\ 1.1 = 0$ ), то результат опроса равен "1", а если он разомкнут, то результат опроса равен "0".

**Поток сигнала в  
последовательной  
цепи**

Рисунок 11–3 показывает пример списка команд, где с помощью операции U (И) два замыкающих контакта включаются последовательно. Выход A 4.0 равен "1" и потому катушка проводит ток только тогда, когда оба замыкающих контакта имеют состояние сигнала "1" (замкнуты).

Программа на AWL	Релейная схема
U E 1.0	E 1.0
U E 1.1	E 1.1
= A 4.0	A 4.0

Рис. 11-3. Последовательное включение контактов, операция И

**Поток сигналов  
в параллельной  
цепи**

Рисунок 11–4 показывает список команд, где с помощью операции О (ИЛИ) два замыкающих контакта включаются параллельно.

Выход А 4.0 равен ”1” и потому катушка проводит ток тогда, когда один из двух замыкающих контактов имеет состояние сигнала ”1” (замкнут).

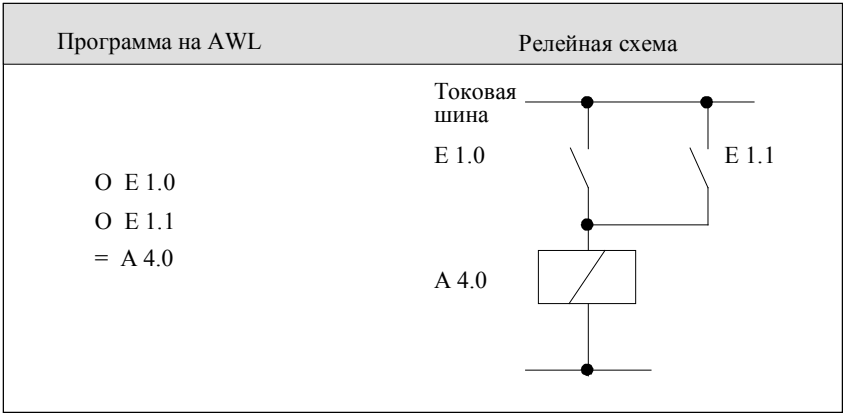


Рис. 11-4. Параллельное включение контактов с помощью операции ИЛИ

**ИСКЛЮЧАЮЩЕЕ  
ИЛИ**

Операция ИСКЛЮЧАЮЩЕЕ ИЛИ в форме AWL соответствует включению реле согласно рисунку 11–5, где в каждом случае один размыкающий контакт и один замыкающий контакт взаимно связаны механически. А 4.0 равен ”1” тогда, когда контакты E 1.0 и E 1.1 имеют разные значения.

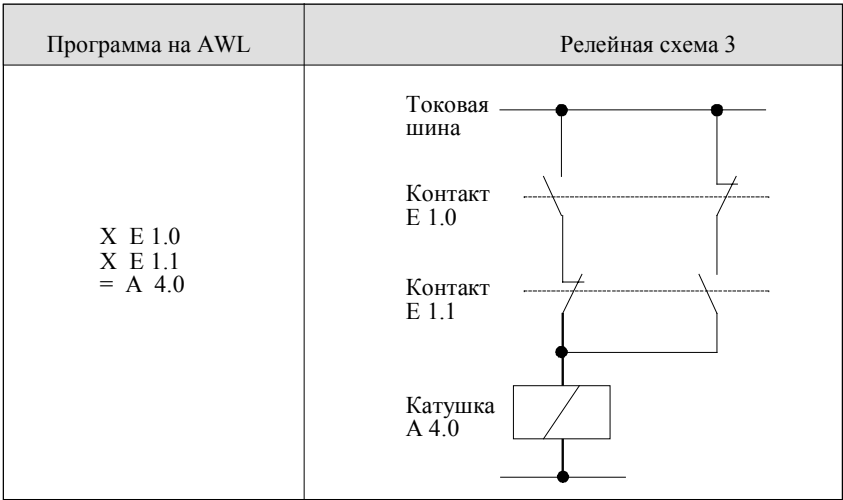
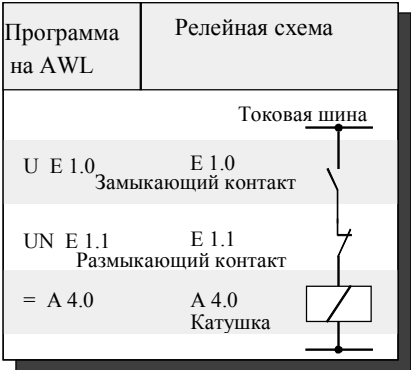


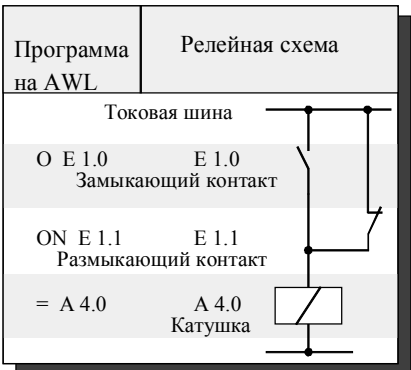
Рис. 11-5. Параллельное включение контактов с помощью операции ИСКЛЮЧАЮЩЕЕ ИЛИ



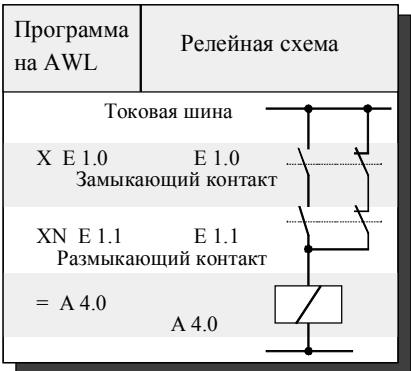
UN: реализовать последовательное включение размыкающего контакта



ON: реализовать параллельное включение размыкающего контакта



XN: реализовать последовательное включение параллельных размыкающего/ замыкающего контактов



### 11.3. Анализ условий с помощью И, ИЛИ и исключающего ИЛИ

#### Описание

С помощью логических операций над битами Вы можете опрашивать биты слова состояния A0, A1, BIE, OV и OS. Эти биты могут быть перед этим установлены следующими операциями (таблица 11–6).

Таблица 11–6. Операции, воздействующие на биты слова состояния A0, A1, BIE, OV и OS

Вид операции	Операция	Глава данного руководства
Арифметика с фиксированной точкой	+I, –I, *I, /I, +D, –D, *D, /D, MOD	15.1
Операции сравнения (арифметика с фиксированной точкой)	==I, <>I, <I, <=I, >I, >=I, ==D, <>D, <D, <=D, >D, >=D	17.2
Арифметика с плавающей точкой	+R, –R, *R, /R, SQRT, SQR, LN, EXP, SIN, COS, TAN, ASIN, ACOS, ATAN	16.1
Числа с плавающей точкой: операции сравнения	==R, <>R, <R, <=R, >R, >=R	17.3
Операции преобразования	ITB, DTB, RND, RND–, RND+, TRUNC, NEGI, NEGD	18.1, 18.2 и 18.4
Функции сдвига и циклического сдвига	SLW, SRW, SLD, SRD, SSI, SSD, RLD, RRD, RLDA, RRDA	20.1 и 20.2
Логические операции со словами	UW, OW, XOW, UD, OD, XOD	19.2
Выражения со скобками	)	11.4
Сохранение VKE в BIE–флаге	SAVE	11.9
Операции перехода	SPBB, SPBNB, SPS	22.1
Управление программой	BEB, BE, BEA, CC, UC	23.6
Операции передачи	T STW	14.3

## Отношение результата к 0

Комбинацию битов A1 и A0 в слове состояния можно опрашивать также упрощенно через “эквивалентные операнды” (например, >0, ==0, <0, и т.д.). Таблица 11–7 представляет связь разных комбинаций битов с упрощенным опросом. Так, например, комбинация A1=0 и A0=1 может опрашиваться в операции И с U<0

Таблица 11–7. Комбинации состояний A0 и A1 и соответствующая возможность опроса

Если имеет место следующая комбинация сигналов в слове состояния,		то опрос может происходить с помощью
состояние сигнала A1	состояние сигнала A0	
1	0	>0
0	1	<0
0 или 1	1 или 0	<>0
1 или 0	0 или 0	>=0
0 или 0	1 или 0	<=0
0	0	==0
1	1	UO

AWL	Объяснение
L +10 // НИЖНЯЯ ГРАНИЦА L MW30	Загрузить целое число 10 как нижнее граничное значение в АККУ1–L.  Загрузить значение меркерного слова MW30 в АККУ1–L и передать целочисленное значение 10 в АККУ2–L.
<=I	Число 10 меньше или равно значению MW30? Если да, то установить VKE в "1"; в противном случае сбросить VKE в "0".
L +100 // ВЕРХНЯЯ ГРАНИЦА	Загрузить целое число 100 как верхнее граничное значение в АККУ1–L и передать значение меркерного слова MW30, которое запомнилось в АККУ1–L, в АККУ2–L.
–I	Вычесть 100 из значения MW30. Этот результат устанавливает A1 и A0 посредством логического сопряжения битов, которое показывает отношение результата к "0" (смотрите таблицу 3–25). VKE остается неизменным.
U <=0	Выполнено ли условие <=0 соответственно логическому сопряжению битов A1 и A0? Ответ "да" доставляет "1", а ответ "нет" – "0" (смотрите таблицу 3–24). Произвести логическое сопряжение этой "1" или "0" с VKE в соответствии с таблицей истинности операции И (смотрите таблицу 3–3). Запомнить результат в VKE–бите.
= A 4.0	Записать значение VKE в ячейку состояния сигнала выхода A 4.0. Катушка на выходе A 4.0 становится токопроводящей (имеет состояние сигнала "1"), когда значение MW30 больше или равно 10, либо меньше или равно 100.

Логические операции над битами дают Вашей программе возможность реагировать также в тех случаях, когда результат арифметической операции над числами с плавающей точкой является недопустимым, так как одно из чисел не является действительным числом с плавающей точкой (недействительное, UO). Операция опрашивает состояние сигнала бита UO слова состояния (смотрите таблицу 11–7).

## Переполнение и двоичный результат

Некоторые из операций из таблицы 11–6 могут устанавливать в “1” бит двоичного результата (BIE) или биты переполнения (OV и OS) слова состояния. Вы можете использовать логические операции над битами U, UN, O, ON, X и XN вместе со следующими областями памяти, чтобы Ваша программа могла реагировать на биты BIE, OV или OS с состоянием сигнала “1”.

AWL	Объяснение
L MW10	Загрузить целое число из меркерного слова MW10 в AKKU1–L.
L MW20	Загрузить целое число из меркерного слова MW20 в AKKU1–L и передать значение MW10 в AKKU2–L.
+I	Сложить два целых числа из аккумуляторов.
T MW30	Передать результат из AKKU1–L в MW30.
U E 0.0	Опросить состояние сигнала на входе E 0.0 на равенство “1” или “0”.
U OV	Опросить бит OV слова состояния на равенство “1” или “0”.
= A 4.0	Если состояние сигнала E 0.0 равно “1” и имеется “1” в бите OV слова состояния (например: во время последней арифметической операции произошло переполнение), то установить состояние сигнала выхода A 4.0 в “1”; в противном случае установить его в “0”.

AWL	Объяснение
U BIE	Опросить бит BIE слова состояния на равенство “1” или “0”.
= A 4.0	Если в бите BIE слова состояния имеется “1”, то установить состояние сигнала выхода A 4.0 в “1”; в противном случае установить его в “0”.

**Адресация битов  
в слове состояния**

Логические операции над битами оценивают условия с помощью операндов, приведенных в таблице 11–8:

Таблица 11–8. Операнды логических операций над битами: биты слова состояния

Область памяти или ссылка на адрес	Биты слова состояния
>0, <0, <>0, >=0, <=0, ==0	7 и 6: индикаторные биты (область памяти)
UO	7 и 6: индикаторные биты(область памяти)
BIE	8: двоичный результат (адрес)
OV	5: переполнение (адрес)
OS	4: переполнение, с запоминанием (адрес)

## 11.4. Логические операции со скобочными выражениями и И перед ИЛИ

### Описание

Вы можете с помощью операций U (И), O (ИЛИ) и X (ИСКЛЮЧАЮЩЕЕ ИЛИ) и их форм с отрицанием UN, ON и XN логически связывать фрагменты последовательностей логических операций, заключенные в скобки (скобочные выражения). Если фрагмент последовательности логических операций стоит в скобках, то программа выполняет операции в скобках до того, как выполнит логическую операцию, стоящую перед скобочным выражением.

Вы можете задавать операции И и ИЛИ в последовательности логических операций также без скобок. Существует установка о том, что вначале обрабатывается операция И, а затем результаты логически связываются в соответствии с таблицей истинности операции ИЛИ.

### Результат логической операции

Операция, открывающая скобочное выражение, сохраняет VKE предыдущей операции в стеке скобок. Позже программа логически связывает этот сохраненный VKE с результатом логических сопряжений, выполненных внутри скобок.

AWL	Объяснение
U( O E 0.0 O M 10.0 )	Операции между U( и ) образуют обычную логическую операцию ИЛИ. Результат первичного опроса запоминается в VKE-бите. Результат опроса логически связывается с VKE предыдущей операции в соответствии с таблицей истинности операции ИЛИ. Данное логическое сопряжение создает новый результат, который заменяет имеющееся значение в VKE-бите.
U(  O E 0.2  O M 10.3  )	U( копирует текущее значение VKE-бита, сохраняет его в стеке скобок и завершает предыдущую цепь логических операций. Поэтому следующая логическая операция начинает новую цепь и выполняет первичный опрос.  Операции между U( и ) образуют обычную логическую операцию ИЛИ. Результат первичного опроса запоминается в VKE-бите.  Результат опроса логически связывается с VKE предыдущей операции в соответствии с таблицей истинности операции ИЛИ. Данное логическое сопряжение создает новый результат, который заменяет имеющееся значение в VKE-бите.
)	Операция ) логически связывает VKE, сохраненное в стеке скобок (смотрите шаг 3), с текущим VKE в соответствии с таблицей истинности операции И. Эта операция использует таблицу истинности операции И, так как символ ) завершает скобочное выражение, начатое символом U. Данное логическое сопряжение образует новый VKE.
U M 10.1	Данная обычная операция И логически связывает новый VKE из шага 6 с результатом опроса в соответствии с таблицей истинности операции И.
= A 4.0	Операция присваивания (=, смотрите главу 11.8.) присваивает значение VKE выходной катушке.

## И перед ИЛИ

Приведенный ниже список команд использует принцип И-перед-ИЛИ для программирования переключательной схемы. По определению, программа сначала обрабатывает операцию И, а затем логически связывает результаты операции И в соответствии с таблицей истинности операции ИЛИ. Скобки не требуются. Принцип, применяемый здесь, называется “И перед ИЛИ”.

AWL	Объяснение
U E 0.0	Результат первичного опроса сохраняется в VKE-бите.
U M 10.0	Результат опроса логически связывается с VKE предыдущей операции в соответствии с таблицей истинности операции И. Данное логическое сопряжение создает новый результат, который заменяет имеющееся значение в VKE-бите.
O	Операция О копирует текущее значение VKE-бита, сохраняет его в OR-бите и завершает предыдущую цепь логических операций. Операция О сохраняет VKE как одно из двух значений, которое она будет использовать для того, чтобы выполнить операцию ИЛИ в соответствии с принципом И-перед-ИЛИ.
U E 0.2	Результат первичного опроса сохраняется в VKE-бите. В каждой операции И, которая следует за операцией ИЛИ, образуемое VKE, в дополнение к обычной операции, логически связывается с OR-битом.
U M 0.3	Результат опроса логически связывается с VKE, образованным предыдущей операцией, в соответствии с таблицей истинности операции И. Данная логическая операция создает новый результат, который заменяет имеющееся значение в VKE-бите. В каждой операции И, которая следует за операцией ИЛИ, образуемый VKE, в дополнение к обычной операции, логически связывается с OR-битом.
	Первая операция О повторяет VKE, сохраненный в стеке скобок, и логически связывает его с текущим VKE. Эта операция образует новое значение, которое запоминается в VKE-бите как результат логической операции И-перед-ИЛИ. (Специальной операции для завершения логического сопряжения И-перед-ИЛИ не существует. Специальный процессор битов в контроллере находит последнюю операцию U в логическом сопряжении И-перед-ИЛИ. Операция, которая следует за последней операцией U, (например, =, S, R или O), автоматически закрывает логическое сопряжение И-перед-ИЛИ при вычислении VKE.)
	Следующая операция О логически связывает результат логического сопряжения И-перед-ИЛИ с результатом опроса второй операции О.
O M 10.1	Операция присваивания (=, смотрите главу 11.8.) присваивает значение VKE выходной катушке.
= A 4.0	

Выход А 4.0 становится токопроводящим (состояние его сигнала равно “1”), когда либо результат одной **или** другой из пары операций И равен “1”, **или** результат обычной операции ИЛИ над М 10.1 равен “1”.

11.5. Операции оценки фронтов: FP, FN

Описание

Вы можете использовать операции FP (положительный фронт) и FN (отрицательный фронт) как распознающие фронт сигнала контакты релейно-контактной схемы. Эти операции регистрируют смену результата логического сопряжения и реагируют на нее. Смена с "0" на "1" называется "нарастающим (положительным) фронтом", а смена с "1" на "0" называется "падающим (отрицательным) фронтом" (см. рис. 11–6).

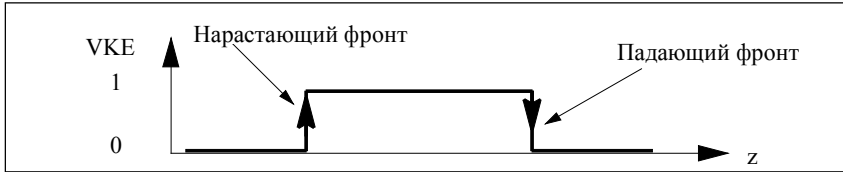


Рис. 11-6. Изображение нарастающего и падающего фронтов

Реакция на нарастающий фронт сигнала

Рисунок 11–7 показывает список команд, который дает Вашей программе возможность реагировать на нарастающий фронт сигнала. Объяснение следует после рисунка.

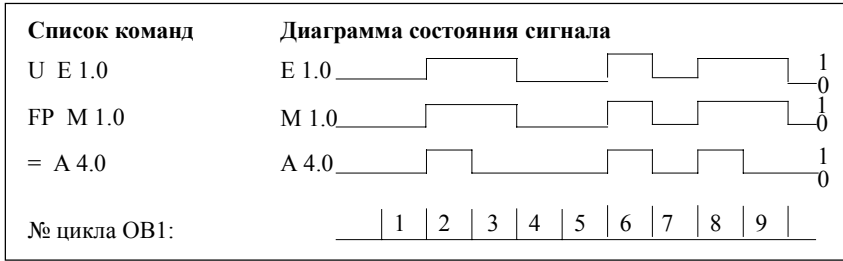


Рис. 11-7. Программирование реакции на нарастающий фронт

Когда контроллер обнаруживает нарастающий фронт на контакте E 1.0, он активизирует выход A 4.0 на один цикл OB1. Контроллер запоминает результат логической операции, полученный после выполнения U, в меркере фронта M 1.0 и сравнивает его с VKE предыдущего цикла. (В примере на рисунке 11–7 VKE команды "U E 1.0" случайным образом равен состоянию сигнала входа E 1.0. Однако это не будет иметь места в каждой программе.) Если текущий VKE равен "1", а VKE предыдущего цикла, сохраненный в меркере фронта M 1.0, равен "0", то операция FP устанавливает VKE в "1". Операция FP распознает нарастающий фронт на контакте (в частности, смену VKE "0" на "1"). Если смены VKE нет (и текущий VKE, и предыдущий VKE, сохраненный в меркере фронта, оба равны "0" или "1"), то операция FP сбрасывает VKE в "0".



Таблица 11–9. Опрос на наличие нарастающего фронта на входе E 1.0

Номер цикла OB1	Состояние сигнала на входе в предыдущем цикле	Состояние сигнала на входе в текущем цикле	Состояние сигнала сменилось с "0" на "1"?	Катушка на выходе A 4.0 проводит ток?
1	0 (значение по умолчанию)	0	Нет.	Нет.
2	0	1	Да.	Да.
3	1	1	Нет.	Нет.
4	1	0	Нет.	Нет.
5	0	0	Нет.	Нет.
6	0	1	Да.	Да.
7	1	0	Нет.	Нет.
8	0	1	Да.	Да.
9	1	1	Нет.	Нет.

Таблица 11–9 относится к AWL–программе на рисунке 11–7. В общем случае Вы должны рассматривать изменения, обнаруживаемые операциями FP и FN, как изменения, которые находят свое отражение в VKE, а не в состояниях сигналов контактов. Например, цепь логических операций может образовать VKE, который не связан непосредственно с состоянием сигнала контакта.

#### Реакция на падающий фронт сигнала

Рисунок 11–8 показывает список команд, который дает Вашей программе возможность реагировать на падающий фронт сигнала. Объяснение следует за этим рисунком.



Рис. 11–8. Программирование реакции на падающий фронт

Когда контроллер обнаруживает падающий фронт на контакте E 1.0, он активизирует выход A 4.0 на один цикл OB1. Контроллер запоминает результат логической операции, полученный после выполнения U, в меркере фронта M 1.0 и сравнивает его с VKE предыдущего цикла (смотрите таблицу 11–10). (В примере на рисунке 11–8 VKE команды "U E 1.0" случайным образом равен состоянию сигнала входа E 1.0. Однако это не будет иметь места в каждой программе.) Если текущий VKE равен "0", а VKE предыдущего цикла, сохраненный в меркере фронта M 1.0, равен "1", то операция FN устанавливает VKE в "1".

Операция FN распознает падающий фронт на контакте (например, когда состояние сигнала VKE сменилось с "1" на "0"). Если смены VKE нет (и текущий VKE, и предыдущий VKE, сохраненный в меркере фронта, оба равны "0" или "1"), то операция FN сбрасывает VKE в "0".

Таблица 11–10. Опрос на наличие падающего фронта на входе E 1.0

Номер цикла OB1	Состояние сигнала на входе в предыдущем цикле	Состояние сигнала на входе в текущем цикле	Состояние сигнала сменилось с "1" на "0"?	Катушка на выходе A 4.0 проводит ток?
1	0 (значение по умолчанию)	0	Нет.	Нет.
2	0	1	Нет.	Нет.
3	1	0	Да.	Да.
4	0	0	Нет.	Нет.
5	0	1	Нет.	Нет.
6	1	1	Нет.	Нет.
7	1	1	Нет.	Нет.
8	1	0	Да.	Да.
9	0	0	Нет.	Нет.

Таблица 11–10 относится к AWL–программе на рисунке 11–8. В общем случае Вы должны рассматривать изменения, обнаруживаемые операциями FP и FN, как изменения, которые находят свое отражение в VKE, а не в состояниях сигналов контактов. Например, цепь логических операций может образовать VKE, который не связан непосредственно с состоянием сигнала контакта.

#### **Задействованный операнд**

Операнд, задействованный операцией FP или FN, является битом.

Операция обращается к выходу посредством одного из следующих видов операндов:

- Признак операнда и адрес внутри области памяти, заданной признаком операнда (см. таблицы 11–11 и 11–12).
- Бит, передаваемый как параметр (см. таблицу 11–13).

Таблица 11–11. Операнды FP и FN: прямая и косвенная адресация

Признак операнда <sup>1)</sup>	Максимальный диапазон адресов в зависимости от способа адресации			
	Прямая	Косвенная через память	Косвенная регистровая внутри области	
E <sup>2)</sup> A <sup>3)</sup> M DBX DIX	от 0.0 до 65 535.7	[DBD] [DID] [LD] [MD] от 0 до 65 532	[AR 1, P#байт.бит] [AR 2, P#байт.бит]	от 0.0 до 8 191.7

- 1) Примите во внимание **предупреждение**, следующее за данной таблицей.
- 2) Так как операционная система в начале каждого цикла переписывает отображение процесса на входах, то сохраненный операцией FP или FN бит VKE, который использует входной бит операнда, искажается. Примите во внимание **предупреждение**, следующее за данной таблицей.
- 3) Не рекомендуется использовать выход в качестве операнда операции FP или FN. Если Вы желаете воздействовать на выход, то используйте для этого операцию S, R или =.



#### Осторожно

Искажение сохраненного результата логической операции.

Это может иметь своим следствием небольшой материальный ущерб.

Если Вы используете в Вашей программе операцию FP или FN, то меркер, являющийся операндом данной операции, используется FP или FN исключительно для своих собственных целей запоминания. Поэтому Вы не должны использовать операции, изменяющие этот бит, так как в противном случае Вы исказите сохраненный VKE. Данное предупреждение имеет силу для всех задаваемых в операциях областей памяти, которые приведены в таблице 11–11.

Таблица 11–12. Операнды FP и FN: косвенная регистровая адресация с указанием области памяти

Признак операнда <sup>1)</sup>	Диапазон адресов
E, A, M, DBX, DIX	[AR 1, P#байт.бит] от 0.0 до 8191.7 [AR 2, P#байт.бит]

- 1) Область памяти закодирована в AR 1 или AR 2 (см. главу 9.6).

Таблица 11–13. Операнды FP и FN: бит, передаваемый как параметр

Операнд	Формат адресного параметра
Символическое имя	Бит, передаваемый как параметр

## 11.6. Завершение цепи логических операций

### Описание

Вы можете завершить цепь логических операций одной из следующих трех AWL-операций. Каждая из этих операций может воздействовать на бит, представляющий конец данной цепи.

- S (установить): если VKE в предыдущей команде был установлен в "1", то посредством S устанавливается в "1" состояние сигнала контакта или катушки, к которой происходит обращение.
- R (сбросить): если VKE в предыдущей команде был установлен в "1", то посредством R сбрасывается в "0" состояние сигнала контакта или катушки, к которой происходит обращение.
- = (присваивание): независимо от состояния VKE значение VKE присваивается операнду, к которому происходит обращение.

### Завершение цепи логических операций

Когда цепь логических операций завершается, бит первичного опроса (/ER) сбрасывается. Если значение /ER-бита равно "0", то это означает, что следующая операция в программе представляет первую операцию новой цепи логических операций (смотрите главу 9.4, первичный опрос). (Команды условного перехода также сбрасывают /ER-бит в "0"; смотрите главы 22.2–22.4).

Цепи логических операций, которые были начаты операциями U(, UN(, O( и так далее, должны завершаться операцией ). Так как эти команды могут использоваться также посредине цепи логических операций, то они образуют в ней прерывание. Это означает, что новая логическая операция начинается прежде, чем завершилась старая. Для того, чтобы иметь возможность надлежащим образом продолжить старую цепь логических операций, после завершения команд, выполняемых в скобках, снова "восстанавливается" старый /ER-бит, сохраненный при открытии скобки. Следовательно, ограниченные скобками части программы можно представить себе как "промежуточное вычисление", после завершения которого "восстанавливается старая нить".

## 11.7 Операции установки и сброса: S и R

### Описание

С помощью операции S (установить) Вы можете установить в "1" состояние сигнала бита, к которому происходит обращение. (Информацию о том, как Вам с помощью операции S установить определенное значение в счетчике, к которому происходит обращение, смотрите в главе 13.2).

С помощью операции R (сбросить) Вы можете сбросить в "0" состояние сигнала бита, к которому происходит обращение. Кроме того, операция R может сбрасывать в "0" таймер или счетчик, к которому происходит обращение (смотрите главы 12.3 и 13.2). Операции S и R завершают цепь логических операций (смотрите главу 11.6).

### Установка бита

Операция S устанавливает в "1" бит, к которому происходит обращение, если результат логической операции предыдущей команды равен "1" и Master Control Relay (MCR) является токопроводящим (то есть состояние его сигнала равно "1"). Если MCR не активизируется (состояние его сигнала равно "0"), то бит, к которому происходит обращение, не изменяется. Операция S завершает цепь логически операций.

Рисунок 11–9 показывает, как операция S удерживает на уровне "1" состояние сигнала катушки A 4.0, к которой она обращается, до тех пор, пока операция R не сбрасывает это состояние сигнала в "0". Тот факт, что состояние сигнала катушки, к которой происходит обращение, остается в "1" до тех пор, пока операция R не сбрасывает его в "0", поясняет статический характер операции S.

Если в релейной схеме замыкающий контакт на входе E 1.0 был приведен в действие (состояние сигнала становится "1"), то контакт замыкается. Ток течет через контакт на входе E 1.0 и через размыкающий контакт под ним и активизирует при этом выход A 4.0 (состояние сигнала A 4.0 становится "1").

Когда катушка активизируется, замыкающий контакт на выходе A 4.0 напротив E 1.0 замыкается. После этого катушка на выходе A 4.0 остается активизированной (состояние сигнала "1"), независимо от того, разомкнут или замкнут контакт на входе E 1.0. Катушка держит сама себя под напряжением.

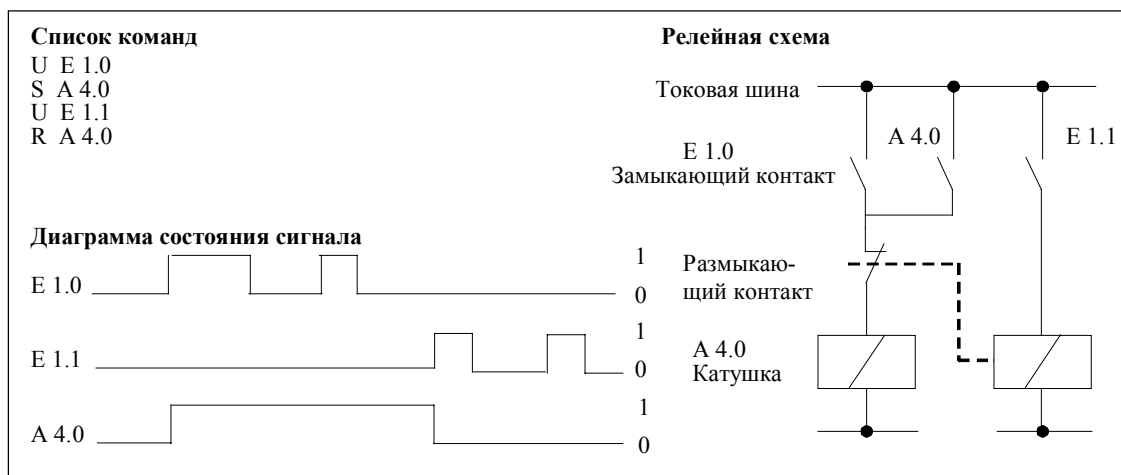


Рис. 11-9. Статическая установка и сброс бита

## Сброс бита

Операция R сбрасывает в "0" бит, к которому происходит обращение, если результат предыдущей логической операции равен "1" и активизируется Master Control Relay (MCR) (то есть состояние его сигнала равно "1"). Если MCR не активизируется (состояние его сигнала равно "0"), то бит, к которому происходит обращение, не изменяется. Операция R завершает цепь логических операций.

Рисунок 11–9 показывает, как операция R удерживает на уровне "0" состояние сигнала катушки A 4.0, к которой она обращается, независимо от изменения состояния сигнала контакта, вызвавшего сброс (E 1.1). Тот факт, что состояние сигнала катушки, к которой происходит обращение, остается в "0" до тех пор, пока операция S не установит его обратно в "1", поясняет статический характер операции R.

В релейной схеме катушка на выходе A 4.0, активизированная операцией S, теперь обесточивается путем замыкания замыкающего контакта на входе E 1.1 (состояние его сигнала становится "0"). Вследствие замыкания контакта E 1.1 ток может течь к катушке под ним. Эта катушка размыкает размыкающий контакт над катушкой на выходе A 4.0, причем ток сигнала к катушке прерывается. Вследствие замыкания контакта E 1.1 вызывается операция R.

## Катушка, к которой происходит обращение

Операнд, к которому происходит обращение со стороны операции S, может быть битом. Операнд, к которому происходит обращение со стороны операции R, может быть битом, таймером или счетчиком. Эти операции обращаются к катушке через один из следующих операндов:

- Признак операнда и адрес внутри области памяти, заданной признаком операнда (смотрите таблицы с 11–14 по 11–16).
- Бит, таймер или счетчик, передаваемый как параметр (см. таблицу 11–17).

Таблица 11–14. Операнды операций S и R: прямая и косвенная адресация

Признак операнда	Максимальный диапазон адресов в зависимости от способа адресации		
	прямая	косвенная через память	косвенная регистровая, внутри области
E A	от 0.0 до 65535.7	[DBD] от 0 до 65 [DID] 532 [LD] [MD]	[AR 1, P#байт.бит] от 0.0 до 8 191.7 [AR 2, P#байт.бит]
M	от 0.0 до 255.7	[DBD] от 0 до 65 [DID] 532 [LD] [MD]	[AR 1, P#байт.бит] от 0.0 до 8 191.7 [AR 2, P#байт.бит]
DBX DIX L	от 0.0 до 65535.7	[DBD] от 0 до 65 [DID] 532 [LD] [MD]	[AR 1, P#байт.бит] от 0.0 до 8 191.7 [AR 2, P#байт.бит]

Таблица 11–15. Операнды операций S и R: косвенная регистровая адресация с указанием области памяти

Признак операнда <sup>1)</sup>	Максимальный диапазон адресов
E, A, M, D, DBX, DIX или L	[AR1, P#байт.бит] от 0.0 до 8191.7 [AR2, P#байт.бит]

<sup>1)</sup> Область памяти закодирована в битах указателя 24, 25 и 26 (смотрите главу 9.6).

Таблица 11–16. Операнды операции R: таймеры и счетчики

Признак операнда	Максимальный диапазон адресов в зависимости от способа адресации	
	прямая	косвенная через память
T <sup>1)</sup> Z	от 0 до 65535	[DW] от 0 до 65 534 [DXW] [LW] [MW]

<sup>1)</sup> Операция S, устанавливающая в “1” бит, к которому происходит обращение, не имеет силы для таймеров и счетчиков. Операция S, используемая счетчиком, устанавливает в этот счетчик определенное значение. Таймеры запускаются операциями для определенных типов таймеров (смотрите главы 12.2, 12.3 и 13.2).

Таблица 11–17. Операнды операций S и R: бит, таймер или счетчик, передаваемый как параметр

Операнд	Формат адресного параметра
Символическое имя	Бит, таймер <sup>1)</sup> или счетчик, передаваемый как параметр

<sup>1)</sup> Операция S не имеет силы для таймеров. Таймеры запускаются операциями для определенных типов таймеров (смотрите главы 12.2, 12.3).

# 11.8. Операция присваивание (=)

**Описание** Каждая логическая операция дает результат, называемый “результатом логической операции” (VKE). Этот VKE равен либо ”1”, либо ”0”. В случае контактов и катушек ”1” показывает, что ток течет, а ”0” показывает, что ток не течет.

Вы можете с помощью операции присваивания (=) скопировать VKE предыдущей операции в цепи логических операций и затем присвоить его в качестве состояния сигнала катушке, к которой обращается операция =. Операция = завершает цепь логических операций (смотрите главу 11.6).

**Установка или сброс бита** Значение, присваиваемое операцией = катушке, к которой происходит обращение, может быть равно ”1” или ”0”, в зависимости от VKE операции, предшествующей операции =. В отличие от операций S и R операция = имеет динамический характер. Она присваивает VKE в качестве состояния сигнала выходу, к которому обращается операция =. Рисунок 11–10 показывает, как изменяется это значение, когда изменяется VKE команды ”U E 1.0”.

На рисунке 11–10 операция = дает возможность входному сигналу на контакте (E 1.0) активизировать или деактивизировать выход (A 4.0) (то есть операция = путем присваивания VKE предшествующей операции устанавливает или сбрасывает выход A 4.0 ).

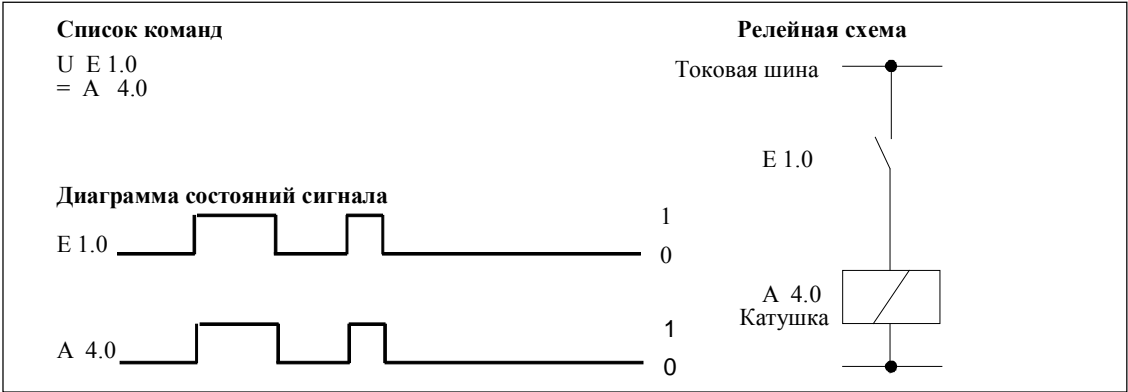


Рис. 11-10. Динамическая установка и сброс бита



## Операнды

Операнд, к которому обращается операция =, может быть битом. Операция обращается к катушке через один из следующих операндов:

- Признак операнда и адрес внутри области памяти, заданной признаком операнда (смотрите таблицы с 11–18 по 11–19).
- Бит, передаваемый как параметр (смотрите таблицу 11–20).

Таблица 11–18. Операнды операции =: прямая и косвенная адресация

Признак операнда	Максимальный диапазон адресов в зависимости от способа адресации				
	прямая	косвенная через память		косвенная регистровая, внутри области	
E A	от 0.0 до 65 535.7	[DBD] [DID] [LD] [MD]	от 0 до 65 532	[AR1, P#байт.бит] [AR2, P#байт.бит]	от 0.0 до 8 191.7
M	от 0.0 до 65 535.7	[DBD] [DID] [LD] [MD]	от 0 до 65 532	[AR1, P#байт.бит] [AR2, P#байт.бит]	от 0.0 до 8 191.7
DBX DIX L	от 0.0 до 65 535.7	[DBD] [DID] [LD] [MD]	от 0 до 65 532	[AR1, P#байт.бит] [AR2, P#байт.бит]	от 0.0 до 8 191.7

Таблица 11–19. Операнды операции =: косвенная регистровая адресация с указанием области памяти

Признак операнда <sup>1)</sup>	Максимальный диапазон адресного параметра
E, A, M, D, DBX, DIX или L	[AR1, P#байт.бит] от 0.0 до 8 191.7 [AR2, P#байт.бит]

<sup>1)</sup> Область памяти закодирована в старших 8 битах AR 1 или AR 2

Таблица 11–20. Операнды операции =: бит, передаваемый как параметр

Операнд	Формат адресного параметра
Символическое имя	Бит, передаваемый как параметр

## 11.9. Отрицание, установка, сброс и сохранение VKE

### Описание

Вы можете изменять результат логической операции (VKE), который сохранен в VKE-бите слова состояния контроллера, с помощью следующих операций (см. главу 11.8):

Мнемоника	Операция	Значение
NOT	Инвертировать VKE	Отрицание (обращение) текущего VKE
SET	Установить VKE	Установка текущего VKE в "1"
CLR	Сбросить VKE	Сброс текущего VKE в "0"
SAVE	Сохранить VKE в BIE-флаге	Сохранение текущего VKE в бите слова состояния

Так как эти операции воздействуют непосредственно на VKE, они не имеют операндов.

### Отрицание VKE

Вы можете в Вашей программе с помощью операции NOT инвертировать (обратить) текущий VKE. Если текущий VKE равен "0", то операция NOT изменяет его на "1", если OR-бит не установлен. Если текущий VKE равен "1", то операция NOT изменяет его на "0", если OR-бит не установлен. Эта операция является выгодной для того, чтобы, например, сократить Вашу программу путем замены положительного переключения отрицательным (смотрите пример в главе В.3: таймеры).

### Установка VKE в "1"

Вы можете в Вашей программе с помощью операции SET установить VKE-бит абсолютно в "1". Рисунок 11-11 показывает, как работает операция SET в программе.

### Сброс VKE в "0"

Вы можете в Вашей программе с помощью операции CLR сбросить VKE-бит абсолютно в "0". CLR сбрасывает в "0" также /ER-, OR- и STA-биты. После этого цепь логических операций заканчивается. Рисунок 11-11 показывает, как работает операция CLR в программе.

### Сохранение VKE

Вы можете в Вашей программе с помощью операции SAVE сохранить VKE для будущего использования или воздействовать на BIE-бит слова состояния контроллера, например, когда Вы программируете функциональные блоки (FB) и функции (FC) для программных блоков KOP.

Операция	Воздействие на биты слова состояния								
	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
NOT						-	1	x	-
SET						0	1	1	0
CLR						0	0	0	0
SAVE	x								

**Применение  
SET и CLR**

Программа на рисунке 11–11 показывает применение операций SET и CLR для абсолютной установки или абсолютного сброса бита.

Список команд	Состояние сигнала	Результат логической операции ? VKE)
SET		1
= M 10.0	1	
= M 15.1	1	
= M 16.0	1	
CLR		0
= M 10.1	0	
= M 10.2	0	

Рис. 11-11. Абсолютная установка и сброс бита с помощью SET и CLR

Вы можете использовать команды программы, изображенной на рисунке 11–11, в организационном блоке запуска (ОВ). После того, как Вы включите Ваш контроллер, он обрабатывает ОВ запуска вместе с содержащимися в нем командами. После того, как контроллер выполнит все команды, следующие меркерные биты независимо от условий имеют определенное состояние сигнала:

- состояние сигнала меркеров M 10.0, M 15.1 и M 16.0 равно "1".
- состояние сигнала меркеров M 10.1 и M 10.2 равно "0".