

Что описывает эта глава?

Прикладная программа обязательно должна содержать кодовые блоки. Однако при многих постановках задач Вы можете обратиться к встроенным в центральный модуль блокам или к предложенным стандартным функциональным блокам. Однако некоторое минимальное количество кодовых блоков Вы в любом случае должны будете создать самостоятельно. Эта глава описывает, как нужно при этом действовать в редакторе КОР.

Обзор главы

В разделе	Вы найдете	на стр.
3.1	Создание кодовых блоков. Обзор	3–2
3.2	Кодовые блоки в редакторе	3–4
3.3	Структура таблицы описания переменных	3–7
3.4	Редактирование таблицы описания переменных	3–9
3.5	Описание мультиэкземпляров	3–11
3.6	Редактирование операторной части. Обзор	3–12
3.7	Основные правила ввода элементов КОР	3–14
3.8	Ввод элементов КОР	3–17
3.9	Создание параллельных ветвлений	3–20
3.10	Редактирование адресов и параметров	3–22
3.11	Символическая адресация	3–23
3.12	Изменения в режиме замены	3–25
3.13	Ввод заголовков и комментариев	3–27

3.1. Создание кодовых блоков. Обзор

Кодовые блоки

Кодовые блоки (OB, FB, FC) состоят из раздела описания переменных, операторной части и, кроме того, обладают свойствами. Таким образом, при программировании Вы должны редактировать следующие три части:

- **Таблица описания переменных.** В таблице описания переменных Вы задаете параметры или локальные переменные блока.
- **Операторная часть.** В операторной части Вы программируете коды блока, которые должны обрабатываться автоматизированной системой. Она состоит из одной или нескольких сетей с элементами контактного плана.
- ⑩ **Свойства блока.** Свойства блока содержат дополнительную информацию, такую как метки времени и указание путей, которая вносится системой. Кроме того, Вы можете сами ввести информацию об имени, семействе, версии и авторе (см. главу 5)

Редактирование кодového блока

Принципиальным является, то в какой последовательности Вы редактируете три части кодового блока. Само собой разумеется, что Вы можете проводить также последующие дополнения и корректировку. Если Вы хотите обратиться к символу из таблицы символов, то Вы должны сначала проверить ее на полноту и при необходимости дополнить.

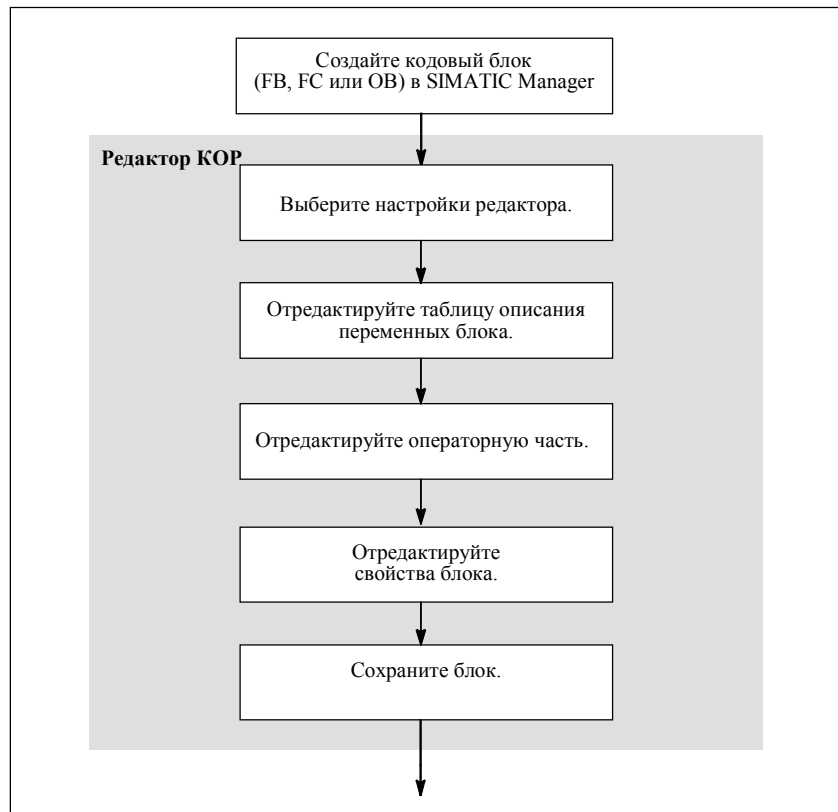


Рис 3-1. Последовательность программирования кодовых блоков в КОР

3.2. Кодовые блоки в редакторе

Обзор

Прежде чем начинать программирование в редакторе KOP, Вы должны познакомиться с возможностями настроек, чтобы иметь возможность работать удобно и в соответствии с Вашими привычками.

Настройка редактора

Воспользовавшись командой меню **Extras ► Einstellungen** (Дополнительные функции ► Настройка), Вы откройте окно регистрового диалога. На вкладке "Editor" (Редактор) Вы можете сделать предварительные установки для программирования блоков, а именно:

- шрифт (вид, стиль, величину) в тексте и таблицах
- предпочитаемый Вами язык (KOP или AWL). В соответствии с этим указанием вновь создаваемый блок открывается в KOP или AWL. Несмотря на это Вы при необходимости можете впоследствии, при учете ограничений, просматривать блок в другом языке.
- хотели бы Вы получить при работе с новым блоком в первую очередь символику и комментарии.

Настройки для языка, комментариев и символики Вы можете изменять прямо во время редактирования через команду меню **Ansicht ► ...** (Вид ► ...).

Настройки для KOP

На вкладке "KOP", которая также вызывается с помощью команды меню **Extras ► Einstellungen** (Дополнительные функции ► Настройка), Вы можете сделать следующие основные установки:

- KOP-Layout (Компоновка KOP): Здесь задается размер отображения Ваших сетей. В зависимости от выбранной величины Вы можете разместить больше или меньше элементов KOP в одной ветви. Эта установка оказывает также влияние при распечатке Вашего блока.
- Operandenfeldbreite (Ширина поля операнда): Здесь Вы задаете ширину текстового поля для операндов, например, для адресов. При превышении ширины поля операндов происходит разрыв строки. При символической адресации требуется большее поле операндов, а при абсолютной меньшее.
- Linie/Farbe für (Линия/цвет для): выбранный элемент, цвет, контакты, линия, статус, цвет и линия

Основное окно редактора KOP

После открытия кодового блока появляется окно с:

- **таблицей описания переменных** блока в верхней части и
- **операторной частью** в нижней части, в которой Вы редактируете собственный код блока.

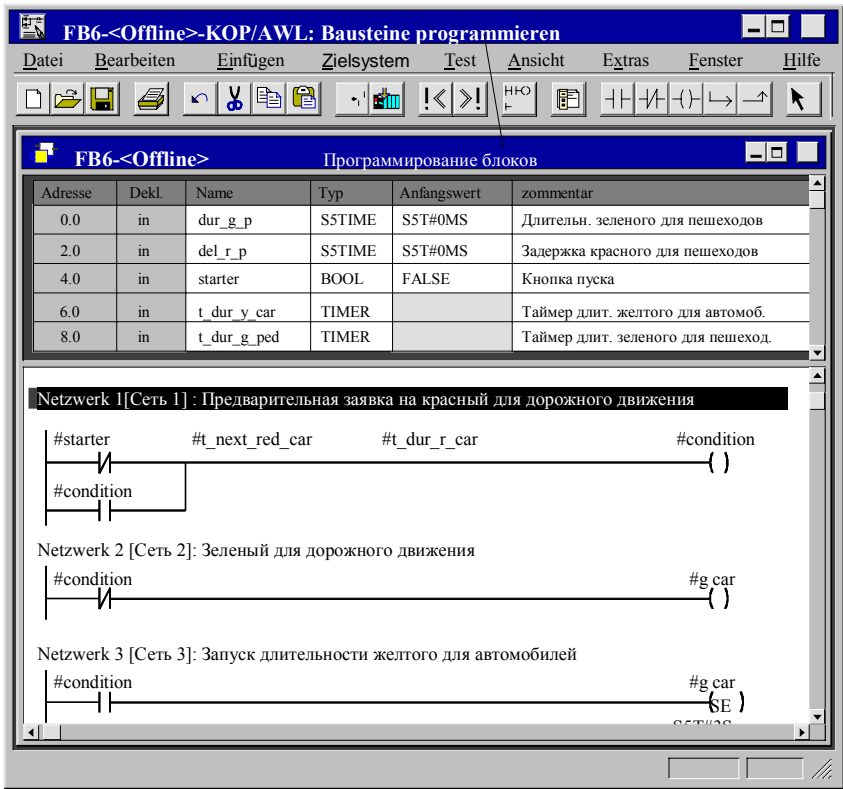


Рис 3-2. Таблица описания переменных и операторная часть в KOP

Свойства блока Вы обрабатываете в собственном диалоге (см. главу 5).

В редакторе Вы можете открыть несколько блоков и при желании обрабатывать их поочередно.

Взаимодействие таблицы описания переменных и операторной части

Таблица описания переменных и операторная часть кодового блока тесно связаны друг с другом, так как в операторной части используются имена из таблицы описания переменных. Изменения в таблице описания переменных воздействуют на операторную часть и на весь блок в целом.

Таблица 3–1. Взаимодействие таблицы описания переменных и операторной части

Действия в описании переменных	Реакция в операторной части
корректный новый ввод	если присутствует некорректный код, то теперь становится корректной ранее не описанная переменная.
корректное изменение имени без изменения типа	символ представляется везде и сразу новым именем.
корректное имя изменяется на некорректное	корректный код становится некорректным.
некорректное имя изменяется на корректное	если присутствует некорректный код, то он становится корректным.
изменение типа	если присутствует некорректный код, то он становится корректным, а если присутствует корректный код, то он становится некорректным.
стирание переменной (символа), которая применяется в коде	корректный код становится некорректным.
изменение комментариев	нет
ошибочный ввод новой переменной	нет
стирание не применяемой переменной	нет
изменение начальных значений	нет

3.3. Структура таблицы описания переменных

Обзор

В таблице описания переменных Вы задаете локальные переменные, включая формальные параметры блока. Это имеет, среди прочего, следующие результаты:

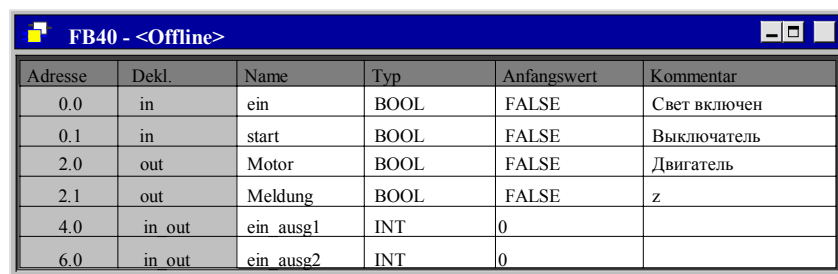
- благодаря описанию резервируется память в стеке локальных данных или в экземпляре блока данных.
- с помощью выбора входных, выходных и проходных параметров Вы задаете "интерфейс" для вызова блоков в программе.
- ⑩ если Вы описали переменные в функциональном блоке, то эти переменные определяют также структуру данных для каждого экземпляра блока данных, который соответствует этому функциональному блоку.

Структура таблицы описания переменных

После того как Вы открыли вновь созданный кодовый блок, выводится предварительно настроенная таблица описания переменных. Она содержит только допустимые для выбранного вида блока описания типов (in, out, in_out, stat, temp) и, кроме того, в предписанном порядке.

При создании нового организационного блока выводится стандартное описание переменных, значения в котором Вы можете менять.

Таблица описания переменных содержит записи для адреса, типа описания, имени, типа данных, начального значения и комментариев для переменной. Каждая строка таблицы предназначена для одного описания переменной. Переменные типа массив или структура нуждаются в нескольких строках.



Adresse	Dekl.	Name	Typ	Anfangswert	Kommentar
0.0	in	ein	BOOL	FALSE	Свет включен
0.1	in	start	BOOL	FALSE	Выключатель
2.0	out	Motor	BOOL	FALSE	Двигатель
2.1	out	Meldung	BOOL	FALSE	z
4.0	in out	ein ausg1	INT	0	
6.0	in out	ein ausg2	INT	0	

Рис 3-3. Пример таблицы описания переменных

Изменение ширины столбца

Вы можете изменять ширину столбца таблицы. Для этого расположите указатель мыши между двумя столбцами и раздвигайте столбцы в горизонтальном направлении при нажатой левой клавише мыши. Также изменить ширину столбца Вы можете с помощью команды меню **Ansicht ► Spaltenbreite** (Вид ► Ширина столбца). Если Вы хотите отказаться от обязательного ввода комментариев или начальных значений, то Вы можете этими способами минимизировать соответствующие столбцы, чтобы иметь возможность сконцентрироваться на других столбцах.

Значение столбцов Столбцы таблицы переменных имеют следующее значение:

Таблица 3–2. Столбцы таблицы описания переменных

Столбец	Значение	Примечание	Обработка
Adresse (Адрес)	Адрес в формате BYTE.BIT.	Если для типа данных требуется больше чем байт, то присвоение адреса производится посредством перехода к следующему адресу байта. Пояснение к символу: * : величина элемента массива в байтах. + : начальный адрес, относительно начала структуры = : общая потребность структуры в памяти	Системная запись: Адрес предоставляется системой и выводится по окончании ввода описания.
Name (Имя)	Символическое имя переменной	Имя должно начинаться с буквы. Зарезервированные ключевые слова запрещены.	необходимо
Deklaration (Описание)	Тип описания, "цель использования" переменной.	Возможны в зависимости от типа блока: Входной параметр "in" Выходной параметр "out" Проходной параметр "in_out" Статическая переменная "stat" Временная переменная "temp"	Системное задание в соответствии с типом блока.
Typ (Тип)	Тип данных переменной (BOOL, INT, WORD, ARRAY и т.д.).	Элементарные типы данных Вы можете выбирать в меню при помощи правой клавиши мыши.	необходимо
Anfangswert (Начальное значение)	Начальное значение, если программное обеспечение не берет на себя значение по умолчанию.	Должно быть совместимо с типом данных. Начальное значение принимается при первом сохранении блока данных как текущее значение переменной, если Вы явно не задали текущего значения.	не обязательно
Kommentar (Комментарий)	Комментарии для документации		не обязательно

3.4. Редактирование таблицы описания данных

Последовательность действий При новом вводе описания Вы вводите за желаемым типом описания по очереди имя, тип, начальное значение (необязательно) и комментарий (необязательно). При этом Вы передвигаете курсор в соседнее поле нажатием клавиши TAB. По завершении строки переменной присваивается адрес.

После каждой обработки поля таблицы производится проверка синтаксиса, в ходе которой обнаруженные ошибки выделяются красным. Вы не обязаны проводить исправления сразу же, а можете проделать это при дальнейшем редактировании и проводить корректировку в более поздний момент времени.

Функции редактирования Для редактирования таблицы в Вашем распоряжении находятся известные функции меню **Bearbeiten** (Редактирование). Для простой обработки Вы можете вызвать правой клавишей мыши контекстно-зависимое меню.

При вводе типа Вам также оказывает поддержку меню правой клавиши мыши. Под меню "Datentyp" (тип данных) выведены все элементарные типы данных.

Отдельные строки Вы помечаете щелчком на защищенном от записи адресном поле данной строки. Дополнительные строки того же самого типа описания помечаются при нажатой клавише SHIFT. Помеченные имеют черный фон.

Изменение типа описания Столбец "Dekl." (Описание) защищен от записи. Тип описания определяется по месту описания внутри таблицы. Благодаря чему обеспечивается то, что переменные могут вводиться только в правильном порядке типов описаний. Если Вы хотите изменить тип описания, то Вы должны сначала вырезать описание, а затем снова вставить его под новым типом описания.

Ввод структур Если Вы хотите ввести структуру как переменную, то Вы вводите в столбце для имени имя, а в столбце для типа данных ключевое слово **STRUCT**. При нажатии клавиши TAB или клавиши RETURN вставляется пустая строка и заключительная строка для структуры (**END_STRUCT**). В пустую строку Вы вводите элемент структуры, задавая имя, тип и при желании начальное значение. Последующие элементы (строки) Вы вставляете с помощью команды меню или клавиши RETURN.

Если Вы хотите **позначить структуру**, то произведите щелчок мышью в адресном поле первой или последней строки (в которой находится ключевое слово **STRUCT** или **END_STRUCT**). Отдельные описания внутри структуры помечаются с помощью щелчка мышью в адресном поле соответствующей строки.

Если Вы вводите структуру внутри другой структуры, то Ваша иерархия отображается с помощью соответствующего перечисления имен переменных.

Ввод массивов Если Вы хотите ввести в качестве типа данных массив, то Вы вводите в соответствующем столбце "Name" имя, а в столбце "Datentyp" (тип данных) ключевое слово **ARRAY** с заданием размерности массива, например, `array[1..20,3..24]` для двухмерного массива. При нажатии клавиши TAB вставляется строка, в которую Вы вводите тип данных массива.

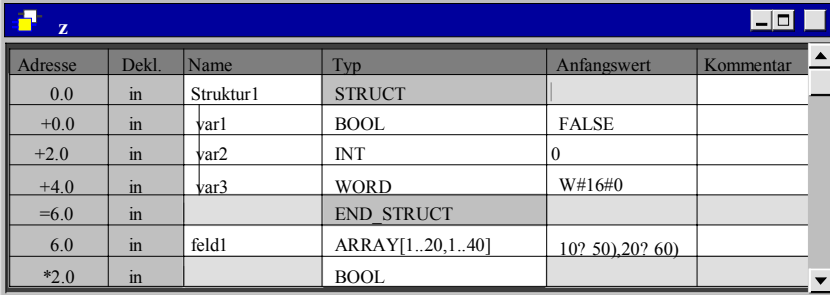
Если Вы хотите **позначить массив**, то щелкните мышью в поле адреса соответствующей строки.

Начальные значения для отдельных элементов массива Вы можете задавать либо также отдельно либо с помощью коэффициента повторения (см рис. 3-4):

- Отдельное задание: Вы присваиваете отдельному элементу собственное начальное значение. Значения перечисляются через запятую.
- Коэффициент повторения: Вы присваиваете нескольким элементам одинаковые начальные значения. Значение задается в круглых скобках, перед которыми ставится коэффициент повторения для всего количества элементов.

Пример

На рис 3.4 приведен пример таблицы описания переменных:



Adresse	Dekl.	Name	Typ	Anfangswert	Kommentar
0.0	in	Struktur1	STRUCT		
+0.0	in	var1	BOOL	FALSE	
+2.0	in	var2	INT	0	
+4.0	in	var3	WORD	W#16#0	
=6.0	in		END_STRUCT		
6.0	in	feld1	ARRAY[1..20,1..40]	10? 50),20? 60)	
*2.0	in		BOOL		

Рис. 3-4. Структуры и массивы в таблице описания переменных

Указание

Если Вы затем измените описание переменных блока, вызов которого Вы уже запрограммировали прежде, то может произойти конфликт меток времени. Поэтому программируйте по возможности в первую очередь все вызываемые блоки, а уже затем блоки, из которых они вызываются. У функциональных блоков в данном случае следует создать заново соответствующие экземпляры DB.

Если Вы затем изменили UDT, который Вы указали в описании переменных как тип данных, то Вы должны перепроверить описание переменных блока и сохранить блок заново.

3.5 Описание мультиэкземпляров

Мультиэкземпляры Мультиэкземплядность возникает в результате описания статической переменной, имеющей в качестве типа данных функциональный блок. В операторной части экземпляр вызывается как элемент KOP.

Дальнейшую информацию о роли мультиэкземпляров Вы найдете в руководстве по программированию /234/. Объяснение синтаксиса вызова мультиэкземпляров Вы найдете в части 2 этого руководства.

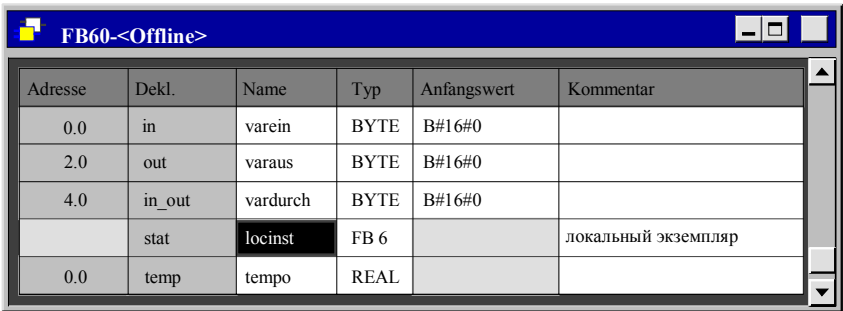
Правила

Для описания мультиэкземпляров действуют следующие правила:

- Описание мультиэкземпляров возможно только в функциональных блоках.
- Функциональному блоку, в котором описывается мультиэкземпляр, должен быть поставлен в соответствие экземпляр DB.
- ⑩ Мультиэкземпляры могут описываться только как статические переменные (тип описания "stat").

Ввод

мультиэкземпляров Для описания мультиэкземпляра введите в столбец “Name” за типом описания "stat" имя переменной. В качестве типа данных укажите функциональный блок. Функциональный блок может указываться абсолютным или символическим именем. При желании Вы можете внести комментарий.



Adresse	Dekl.	Name	Typ	Anfangswert	Kommentar
0.0	in	varein	BYTE	B#16#0	
2.0	out	varaus	BYTE	B#16#0	
4.0	in_out	vardurch	BYTE	B#16#0	
	stat	locinst	FB 6		локальный экземпляр
0.0	temp	tempo	REAL		

Рис. 3-5. Описание мультиэкземпляров (пример)

3.6. Редактирование операторной части. Обзор

Операторная часть В операторной части Вы описываете выполнение программы Вашего кодового блока. Для этого Вы вводите элементы КОР в сети. Операторная часть для кодовых блоков в наиболее частом случае состоит из нескольких сетей. Редактор сразу после ввода элемента КОР проводит проверку синтаксиса и выделяет Ваши ошибки красным цветом. Недопустимое расположение элементов отклоняется с сообщением об ошибке.

Редактируемые части раздела операторов В операторной части Вы можете редактировать заголовок блока, комментарии к блоку, заголовки сетей, комментарии к сетям и строки команд внутри сетей.

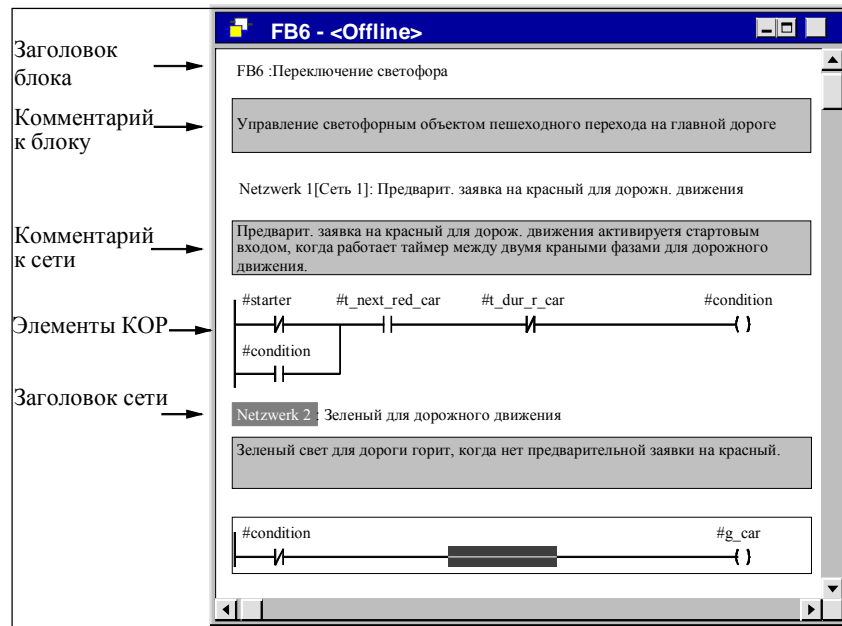


Рис. 3-6. Структура операторной части

Ввод новых блоков

Составные части раздела операторов Вы в принципе можете редактировать в любом порядке. Если Вы в первый раз программируете блок, то рекомендуется следующая процедура:

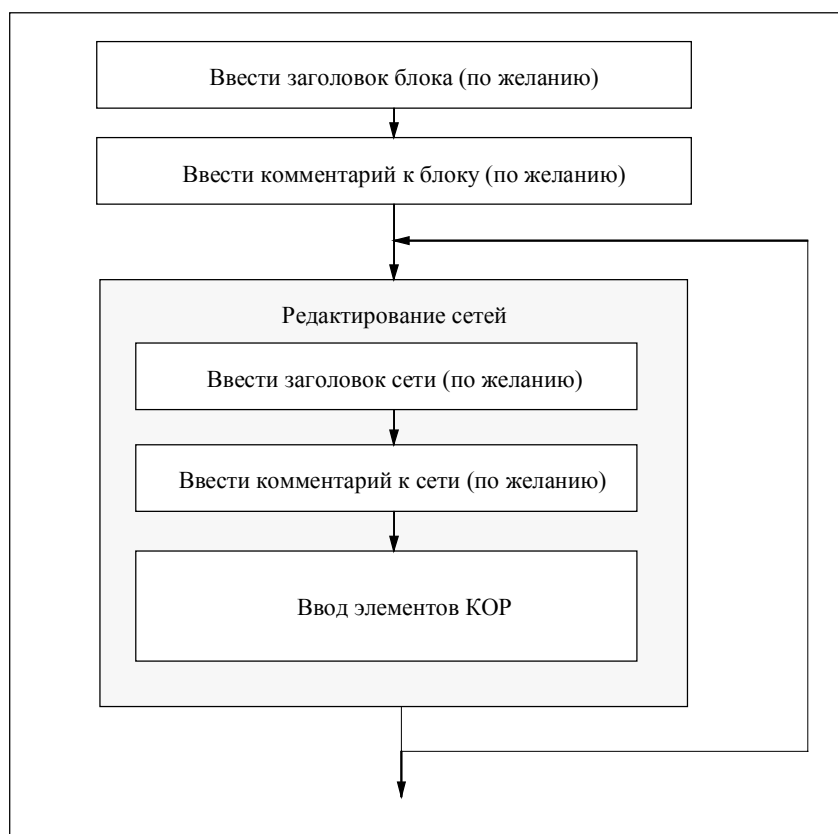


Рис. 3-7. Редактирование операторной части

Изменения Вы можете проводить как в режиме вставки, так и в режиме замены. Переключение между ними производится с помощью клавиши INSERT.

3.7. Основные правила ввода элементов КОР

Общие положения

Сеть КОР может состоять из многих элементов в нескольких ветвях.

Все элементы и ветви должны быть связаны друг с другом, причем левая токовая шина не считается связью (IEC 1131-3).

При программировании в КОР Вы должны соблюдать некоторые правила. На возможные ошибки указывают сообщения об ошибках.

Окончание сети КОР

Каждая сеть КОР должна иметь окончание в форме катушки или блока.

Следующие элементы КОР не могут быть использованы для окончания сети:

- блоки сравнения
- катушки коннекторов $—(\#)—$
- ⑩ катушки для анализа положительного $—(P)—$ или отрицательного $—(N)—$ фронта.

Протекание тока

Нельзя редактировать ветви, которые могут вызвать протекание тока в противоположном направлении. На рис. 3.8 приведен пример. При состоянии сигнала "0" на E 1.4 ток потек бы справа налево через E 6.8. Это недопустимо.

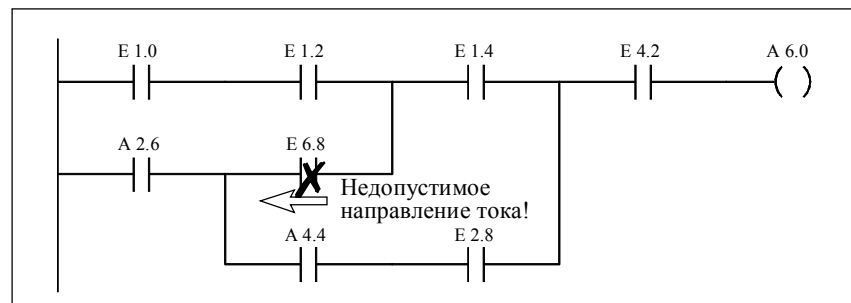


Рис. 3-8. Протекание тока в противоположном направлении ? недопустимо)

Короткое замыкание

Нельзя редактировать ветви, вызывающие короткое замыкание. На рис. 3.9 приведен пример.

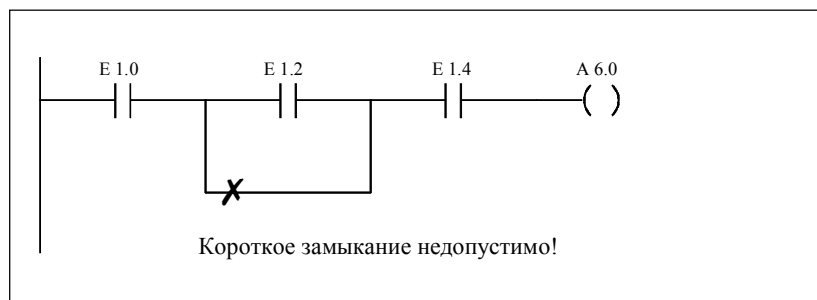


Рис. 3-9. Короткое замыкание в сети КОР (недопустимо)

Размещение блоков

Исходной точкой ветви для подключения блока всегда должна быть левая токовая шина. В ветви перед блоком могут, однако, содержаться логические функции или другие блоки.

Блоки не могут располагаться внутри Т-образных разветвлений. Исключением из этого правила являются блоки сравнения. На рис 3.10 приведен пример.

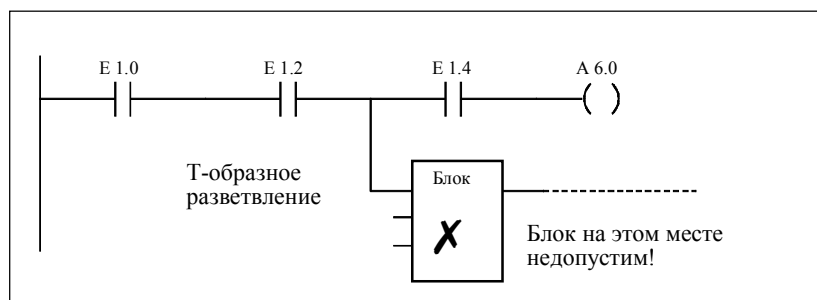


Рис. 3-10. Блок в Т-образном разветвлении (недопустимо)

Размещение катушек

Катушки размещаются автоматически у правого края сети, где они образуют окончание сети.

Исключения: Катушки для коннекторов —(#)— и катушки для анализа положительных —(Р)— или отрицательных —(N)— фронтов не могут размещаться в ветви ни с левого, ни с правого края. Также их нельзя размещать в параллельных ветвях.

Среди катушек есть такие, которые требуют булевых логических операций, и такие, которые не могут их иметь.

- **Катушки, требующие логических операций:**

выход —(), установить выход —(S), сбросить выход —(R), коннектор —(#)—, положительный фронт —(P) —, отрицательный фронт —(N) —

все катушки счетчиков и таймеров

перейти, если 0 —(JMPN)

включить Master Control Relais —(MCR<)

загрузить результат логического соединения в BR-регистр —(SAVE)

возврат —(RET)

- **Катушки, которые не допускают логических операций:**

начало Master Control Relais —(MCRA)

конец Master Control Relais —(MCRD)

открыть блок данных —(OPN)

выключить Master Control Relais —(MCR>)

Все другие катушки могут иметь логические операции или не иметь их.

Следующие катушки Вы не можете использовать в качестве параллельных выходов:

перейти, если 0 —(JMPN)

перейти, если 1 —(JMP)

вызов блока —(CALL)

возврат —(RET)

Разрешающий Подключение разрешающего входа "EN" или разрешающего выхода **ВХОД/ВЫХОД** "ENO" блоков возможно, но не является настоятельно необходимым.

Удаление и изменение

Если ответвление состоит только из одного элемента, то при удалении данного элемента полностью удаляется вся ветвь.

При удалении блока удаляются за исключением главной ветви все ветви, которые связаны с блоком через логические входы.

Для простой смены элементов одинакового типа в Вашем распоряжении имеется режим замены (см. главу 3.12).

3.8. Ввод элементов KOP

Операции над сетями

Чтобы пометить сеть, в которую Вы можете вводить элементы KOP, щелкните мышью на обозначении сети (например, "Netzwerk 1"). Помеченную таким образом сеть Вы можете вырезать, снова вставить или копировать.

Чтобы создать новую сеть, выберите команду меню **Einfügen ► Netzwerk** (Вставка ► Сеть) или щелкните на соответствующем символе в панели инструментов. Новая сеть вставляется под помеченной сетью. Она содержит только одну ветвь с катушкой.

Если Вы ввели больше элементов, чем может отображаться на Вашем экране, то происходит сдвиг сети за левую границу экрана. С помощью команды меню **Ansicht ► Verkleinern/Vergrößern/Zoomfaktor...** (Вид ► уменьшение/увеличение/коэффициент масштабирования...) Вы можете настроить изображение так, чтобы получить наилучший обзор.

Маркирование в сети

В сеть Вы попадаете с помощью щелчка мышью на соответствующем элементе KOP. Внутри сети Вы в принципе можете пометить мышью три области:

- элементы KOP, такие как, например, контакт или блок,
- точку пересечения
- пустые элементы (отрезок проводника или открытую ветвь).

Вы можете выделить одновременно только одну область. (На рис. 3-11 приведены примеры маркирования, причем показано сразу несколько маркированных объектов).

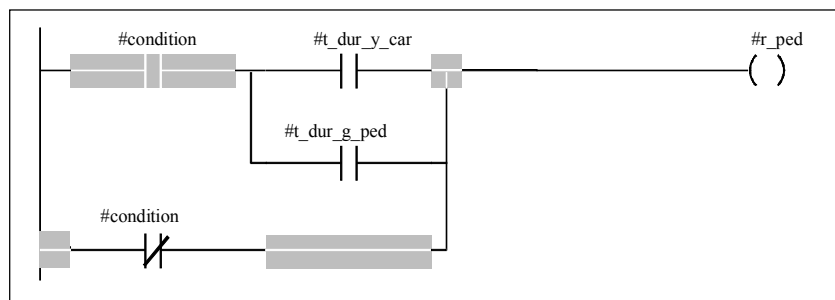


Рис. 3-11. Возможности маркирования в сети KOP

Цвет маркирования Вы можете определить сами. Для этого откройте с помощью команды меню **Extras ► Einstellungen** (Дополнительные функции ► Настройка) вкладку "KOP".

Возможности ввода Вам предоставляются следующие возможности для вставки элементов элементов KOP

KOP:

- Введите замыкающий, размыкающий контакт или катушку функциональной клавишей F2, F3 или F4 соответственно.
- Выберите символ для замыкающего, размыкающего контакта или катушки на панели инструментов.
- Выберите элемент из меню **Einfügen ▶ ...** (Вставка...).
- Выберите элемент из окна диалога **KOP-Element einfügen ▶** (Вставка элемента KOP ▶) (см. рис. 3-12). Для этого выберите команду меню **Einfügen ▶ KOP-Element ...** (Вставка ▶ Элемент KOP), символ на панели инструментов или функциональную клавишу F11.

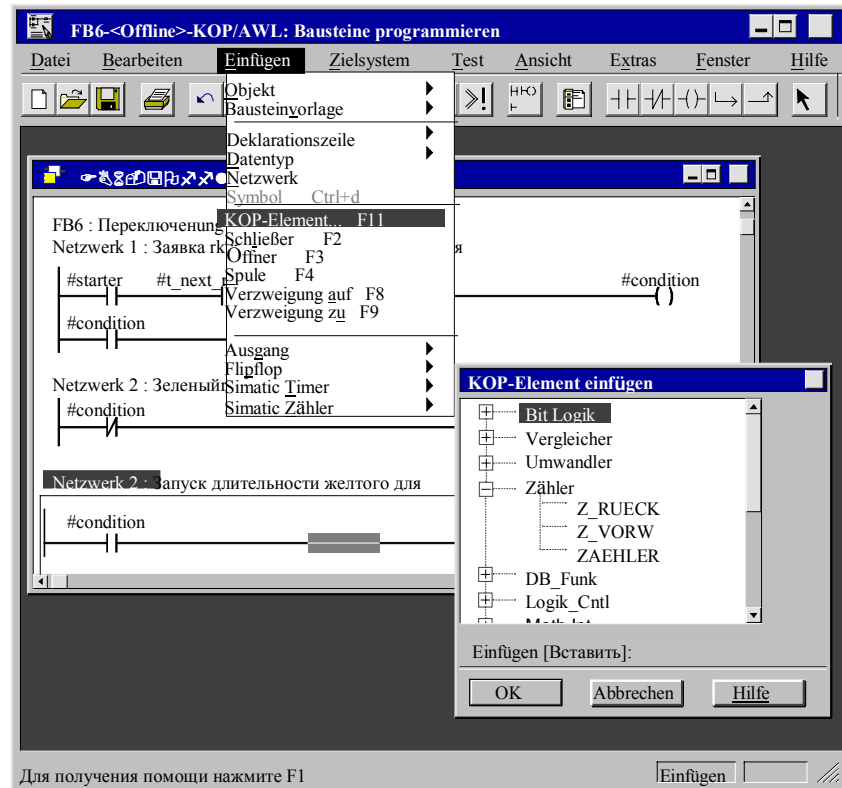


Рис. 3-12. Вставка элемента KOP через диалог для выбора einfügen

Пояснения к рис. 3-12:

Меню Einfügen (Вставка): Objekt - объект; Bausteinvorlage - шаблон для блока; Deklarationszeile - строка описания; Datentyp - тип данных; Netzwerk - сеть; Symbol - символ; KOP-Element - элемент KOP; Schließer - замыкающий контакт; Öffner - размыкающий контакт; Spule - катушка; Verzweigung auf - открыть ветвление; Verzweigung zu - закрыть ветвление; Ausgang - выход; Flipflop - триггер; Simatic Timer - таймер Simatic; Simatic Zähler -счетчик Simatic.

Окно KOP-Element einfügen (Вставка элемента KOP): Bit Logik - битовая логика; Vergleicher - элемент сравнения; Umwandler - преобразователь; Zähler - счетчик; Z_RUECK - обратный счет; Z_VORW - прямой счет; ZAEHLER - счетчик (прямой и обратный счет).

Элементы KOP вставляются всегда за помеченным элементом.

Указание

Если Вы выбрали семейство "FB Bausteine" или "FC Bausteine", то под ним приводятся все соответствующие блоки, которые находятся в программе CPU. Если Вы выбрали семейство "SFB Bausteine" или "SFC Bausteine", то приводятся все доступные в CPU системные функциональные блоки или системные функции. Таким образом, Вы можете включать в свою сеть целые блоки и очень быстро программировать вызов других блоков.

Вызов

мультиэкземпляров переменных, вызываются как элемент KOP. Для этого выбирается команда меню **Einfügen ► KOP-Element ...** (Вставка ► Элемент KOP). Затем в списке элементов KOP Вы находите группу "Multiinstanzen" ("Мультиэкземпляры"), под которой приводятся все мультиэкземпляры.

3.9. Создание параллельных ветвлений

Применение	Чтобы реализовать в языке программирования КОР логическую операцию "ИЛИ", Вы должны создать параллельные ветви.
Правила	<p>При редактировании параллельных ветвлений действуют следующие правила:</p> <ul style="list-style-type: none">• Вставляйте ветвление ИЛИ слева направо.• Параллельные ветви открываются вниз, а закрываются вверх.• Открывайте параллельную ветвь при помощи команды меню Einfügen ► Verzweigung auf (Вставка ► Открыть ветвление), функциональной клавиши F8 или соответствующего символа на панели инструментов.• Закрывайте параллельную ветвь при помощи команды меню Einfügen ► Verzweigung zu (Вставка ► Закрыть ветвление), функциональной клавиши F9 или соответствующего символа на панели инструментов.• Параллельная ветвь открывается всегда перед помеченным элементом.• Параллельная ветвь закрывается всегда позади помеченного элемента. <p>⑩ Чтобы стереть параллельную ветвь, Вы должны стереть все элементы этого ответвления. При удалении последнего элемента ответвления удаляется и остаток ветви.</p>
Создание новых ветвей	Если Вы хотите открыть новую ветвь, пометьте начальный пункт ветви, под которой Вы хотите вставить новую ветвь. При помощи клавиши F8 создайте новую ветвь (ср. рис. 3-13).
Создание замкнутого ветвления	<p>Чтобы создать замкнутое ветвление, отметьте элемент, перед которым Вы хотите открыть параллельную ветвь. Откройте параллельную ветвь с помощью F8, вставьте элемент КОР и закройте ветвь с помощью клавиши F9.</p> <p>При замыкании параллельная ветвь дополняется необходимыми пустыми элементами. Если требуется, то ветви располагаются так, чтобы избежать перекрещивания ветвей. Если Вы замыкаете ответвление прямо из параллельной ветви, то оно замыкается за ближайшим следующим элементом КОР.</p> <p>На рис 3-13 приведен пример создания параллельного ветвления только с помощью функциональных клавиш или символов панели инструментов.</p>

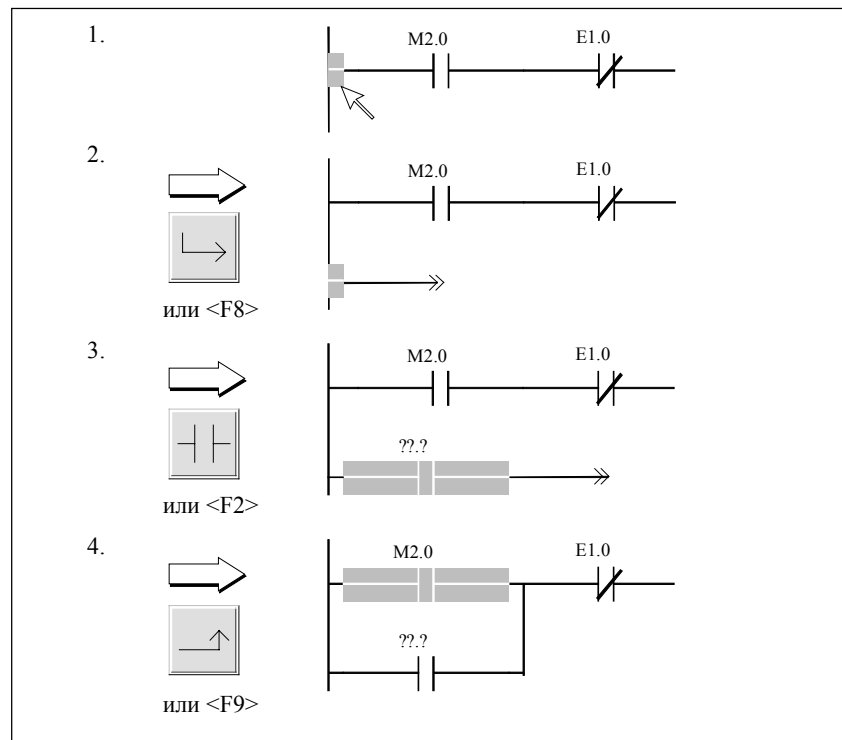


Рис. 3-13. Создание параллельного ветвления в сети КОР

Разделение замкнутых параллельных ветвей

Замкнутую параллельную ветвь можно снова отделить, вырезав точку перекрещивания, в которой параллельная ветвь снова встречается с главной ветвью.

3.10. Редактирование адресов и параметров

Применение

Редактор блоков использует при вставке элементов КОР последовательность знаков ???.? как метку-заполнитель для адресов или параметров. В исполняемой операторной части должны быть заданы все адреса и параметры. За исключением FB и SFB, а также блоков для таймеров и счетчиков, которые не должны быть полностью параметризованы. Адреса и параметры Вы можете задавать в абсолютной или символической форме.

Последовательность действий

Чтобы редактировать адрес или параметр, откройте щелчком мыши на метке ???.? соответствующее текстовое поле. После того как Вы закончили ввод, производится проверка синтаксиса. При наличии ошибки адрес или параметр выделяются красным цветом и в строке статуса выдается сообщение об ошибке. При корректном синтаксисе открывается следующее текстовое поле, которое еще не редактировалось.

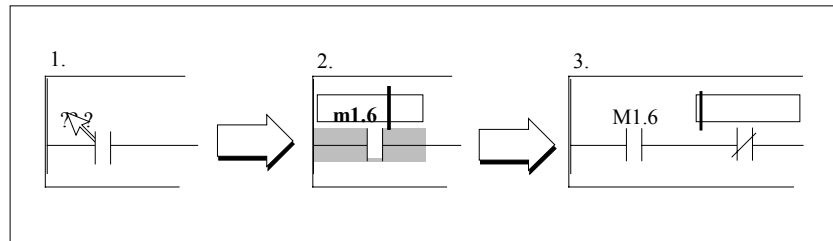


Рис. 3-14. Ввод адресов элементов КОР

Как только Вы познакомились с инструментами редактирования в КОР, рекомендуется сначала ввести все элементы в сеть, а затем присвоить каждому элементу адрес и параметр.

Поиск и устранение ошибок

Благодаря выделению красным цветом ошибки легко распознать. Чтобы легче разбираться в ошибках, которые лежат за пределами непосредственно видимой области, редактор предлагает две функции поиска **Bearbeiten ► Gehe zu vorherigem Fehler / nächstem Fehler** (Редактирование ► Иди к предыдущей ошибке/к следующей ошибке), которые Вы также можете исполнить с помощью символа на панели инструментов.

Поиск ошибок выходит за пределы сети. То есть поиск происходит полностью во всей операторной части, а не только внутри одной сети или непосредственно видимой области. Если Вы включите строку состояния командой меню **Ansicht ► Statuszeile** (Вид ► Строка состояния), то там отображаются указания к ошибкам.

Корректировку ошибок и изменения Вы можете проводить в режиме замены (см. главу 3.12).

3.11. Символическая адресация

Использование символической адресации

В языке программирования KOP Вы можете вводить адреса, параметры и имена блоков абсолютно или как символы. С помощью команды меню **Ansicht ► Symbolische Darstellung** (Вид ► Символическое представление) Вы переключаетесь между абсолютной и символической адресацией.

Определение символов

Если Вы хотите использовать глобальные символы, то Вы должны внести их в таблицу символов. Для чего:

- откройте таблицу символов с помощью команды меню **Extras ► Symboltabelle** (Дополнительные функции ► Таблица символов).
- откройте с помощью команды меню **Einfügen ► Symbol** (Вставка ► Символ) диалоговое окно, в котором Вы можете заново определять или изменять отдельные символы.

Дальнейшую информацию по редактированию символов Вы найдете в руководстве пользователя /231/.

Изображение

Чаше всего Вам не нужно отмечать, идет ли речь о локальном для блока или о глобальном символе. Однако, если возможна путаница, так как, например, были использованы одинаковые символы в таблице символов и в описании переменной, то Вы можете различить символы следующими способами:

- Символ из **таблицы символов** изображается в кавычках **".."**.
- Символ из **таблицы описания переменных** блока изображается со стоящим впереди значком **"#"**.

Обозначение **".."** или **"#"** Вы вводить не должны. Если в таблице описания переменных или в таблице символов содержится символический адрес, то обозначение добавляется после проверки синтаксиса.

Удобная информация о символе

Для более легкого программирования с символической адресацией Вы можете выводить для каждого символа соответствующий абсолютный адрес в комментариях к символу. Этого Вы можете достичь с помощью команды меню **Ansicht ► Symbolinformation** (Вид ► Информация о символе). Вследствие этого действия вслед за каждой сетью отображается соответствующее информационное поле. Это изображение Вы не можете редактировать; изменения Вы должны производить в таблице символов или в таблице описания переменных.

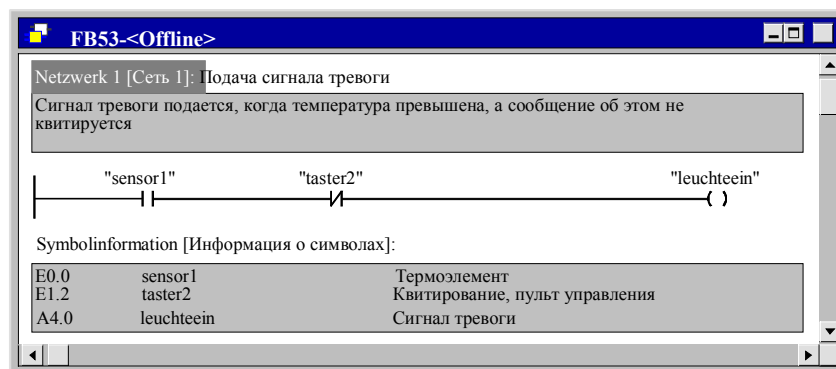


Рис. 3-15. Информация о символах в КОР

При печати блока воспроизводится текущее изображение экрана с комментариями к командам или информацией о символах.

Указание

При загрузке программы в процессор таблица символов не переносится. Поэтому, если Вы хотите редактировать прикладную программу и не располагаете оригинальной программой в системе разработки, то Вы также не располагаете первоначальной символикой.

3.12. Изменения в режиме замены

Замена адресов и параметров В редакторе блока Вы можете удобно изменять адреса и параметры в режиме замены. Переключение между этими режимами производится с помощью клавиши INSERT. В текстовых полях для параметров или адресов Вы можете таким способом переписать Ваши данные.

Замена элементов КОР

Режим замены делает возможным простой обмен элементами КОР одинакового типа. При этом сохраняются все логические связи и все настройки параметров. Преимущество состоит в том, что Вы не должны вводить адреса и параметры заново. Перезаписываемый элемент КОР может заменяться только элементом КОР того же типа. Так, например, взаимозаменяемы замыкающий и размыкающий контакты, R/S - и S/R - триггеры или различные таймеры. Чтобы заменить имеющийся элемент КОР, пометьте его и перейдите с помощью клавиши INSERT в режим замены. Элемент заменяется, как только Вы вставите элемент КОР того же типа.

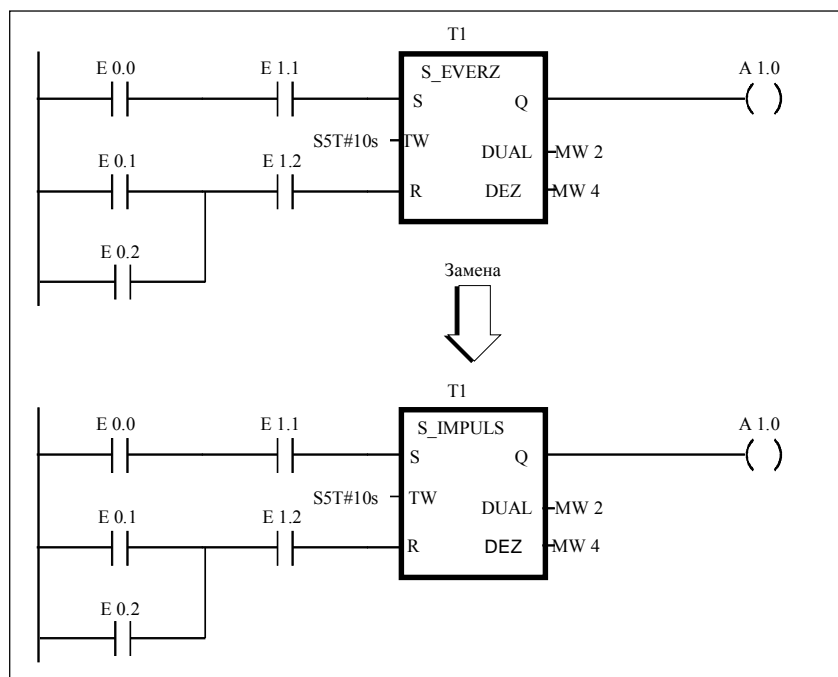


Рис. 3-16. Замена блоков

**Особый случай:
расщепление узла**

Если в одной точке ветви одновременно закрывается параллельная ветвь и открывается следующая, то возникает узел. Такой узел Вы можете расщепить в режиме замены, пометив верхнюю и нижнюю точку соединения и вставив элемент КОР. Узел расщепляется и вставляется элемент КОР.

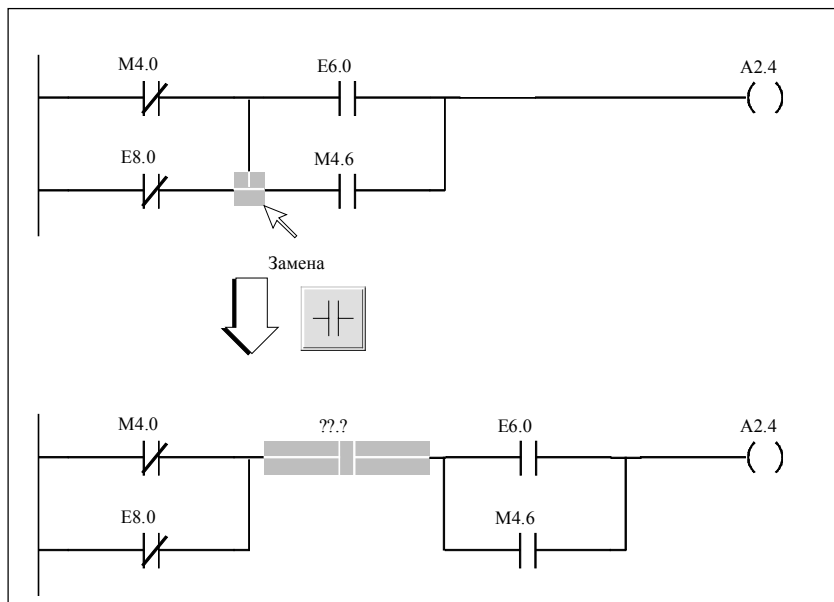


Рис. 3-17. Расщепление узла

Указание

Комментарии и заголовки Вы можете корректировать в режиме вставки.

3.13 Вод заголовков и комментариев

Обзор В операторной части кодового блока Вы можете ввести заголовок блока и заголовок сети, а также комментарий к блоку или комментарий к сети. Этот ввод является необязательным и не отражается на работе программы.

Ввод заголовков блока и сети Чтобы ввести заголовок блока или сети, поместите курсор на трех вопросительных знаках справа от имени блока или сети (например Netzwerk 1 : ???). Открывается текстовое поле, в которое Вы вводите заголовок. Максимальная длина заголовка - 64 знака.

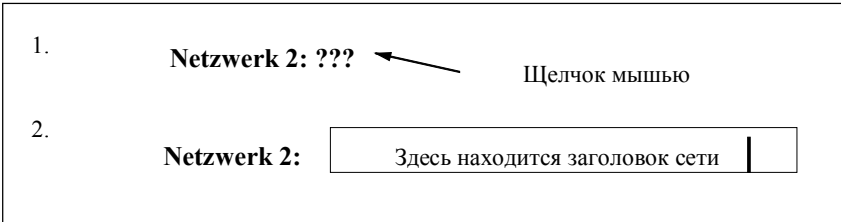


Рис. 3-18. Ввод заголовков

Ввод комментариев Вы можете вывести на экран или удалить с экрана поля комментариев с помощью команды меню **Ansicht ► Kommentar** (Вид ► Комментарий). При двойном щелчке на таком поле комментариев открывается текстовое поле, в которое Вы можете теперь вводить Ваши пояснения. В вашем распоряжении находятся 64 Кбайта на блок для комментариев к блоку и сетям.



Рис. 3-19. Ввод комментариев

Указание

При загрузке блока в процессор комментарии не переносятся. Если Вы загружаете блок из процессора и не располагаете оригинальным блоком в системе разработки, то Вы не можете просмотреть или отредактировать первоначальные комментарии.
