

Содержание главы 16

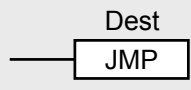
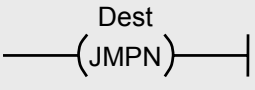
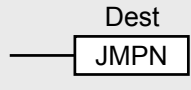
<u>16</u>	<u>Функции перехода</u>	4
<u>16.1</u>	<u>Обработка функции перехода</u>	4
<u>16.2</u>	<u>Безусловный переход</u>	6
<u>16.3</u>	<u>Переход, если RLO = «1»</u>	7
<u>16.4</u>	<u>Переход, если RLO = «0»</u>	8

16 Функции перехода

Функции перехода вы можете использовать для прерывания линейного потока программы и продолжения с другой точки блока. Это программное ветвление может быть выполнено безусловно или в зависимости от RLO (результата логической операции).

16.1 Обработка функции перехода

Представление

Представление LAD		Представление FBD	
Переход, если RLO = «1»		Переход, если RLO = «1»	
Переход, если RLO = «0»		Переход, если RLO = «0»	
Место назначения, метка перехода		Место назначения, метка перехода	

Функция перехода состоит из операции перехода в форме катушки (LAD) или блочного элемента (FBD) и метки перехода, обозначающей место в программе, с которого после перехода продолжится обработка. Метка перехода указывается над операцией перехода.

Переходы нельзя программировать совместно с Т-ветвью.

Метка перехода может включать в себя до 4 символов, которые могут быть буквами, цифрами и знаками подчеркивания. Начинается она с буквы. Метка перехода в блочном элементе обозначает сеть, которая будет обрабатываться после окончания операции перехода. Блочный элемент с меткой перехода должен располагаться в начале сети (в каталоге программных элементов – «LABEL», «Метка»).

Для пошагового программирования функции перехода можно найти в каталоге программных элементов (Program Elements Catalog), вызвав его командой View → Catalog (Вид → Каталог) [Ctrl – K], или выбрав команду Insert → Program Elements (Вставка → Программные элементы), раздел «Logic Control / Jump» («Логическое управление / Переход»).

Функция

Переход выполняется либо всегда (абсолютный или безусловный переход), либо в зависимости от результата логической операции (RLO) (условный переход). В случае перехода, зависящего от RLO, вы можете решить, при RLO, равным «1» или «0», этот переход будет выполнен.

Переходы можно выполнять как вперед (в направлении обработки программы или увеличения сетевых номеров), так и назад. Переход может осуществляться только внутри блока, то есть место назначения перехода должно быть в том же блоке, что и функция перехода. Если вы используете главное реле управления (Master Control Relay, MCR), метка перехода должна находиться в той же MCR-зоне или в MCR-области, что и функция перехода.

Место назначения перехода должно быть обозначено однозначно, то есть вы должны назначить метку перехода в блоке только один раз. Перейти к месту назначения можно из нескольких точек программы.

Программный редактор сохраняет имена меток переходов в неисполняемом разделе соответствующих блоков в среде данных устройства программирования. В пользовательской памяти CPU (в скомпилированном блоке) хранятся только размеры переходов (смещение перехода). Когда в онлайн-режиме производятся изменения программы в блоке в CPU, эти модификации всегда должны быть проведены в среде данных устройства программирования с тем, чтобы сохранить исходные имена меток. Если это обновление не осуществляется, или если блоки передаются из CPU в программатор, то неисполняемые разделы блоков будут перезаписаны или удалены. Тогда на дисплее или при печати редактор генерирует для меток переходов символы замены (M001, M002 и т.д.).

16.2 Безусловный переход

Абсолютным переходом, который всегда выполняется, является функция перехода JMP, чья катушка соединена с левой направляющей (питающей шиной) (LAD), или ее блочный элемент не имеет предшествующей логической операции (FBD). Эта функция перехода выполняется всегда, когда встречается в программе. CPU прерывает линейный поток программы и продолжает обработку в сети, обозначенной меткой перехода. Пример (рисунки 16.1 и 16.2): В сети 3 (Network 3) представлен безусловный переход к метке перехода M2. После обработки этой сети CPU продолжает выполнение программы с метки перехода M2 в сети 5 (Network 5). Чтобы обработать сеть 4 (Network 4), требуется переход из другой точки программы.

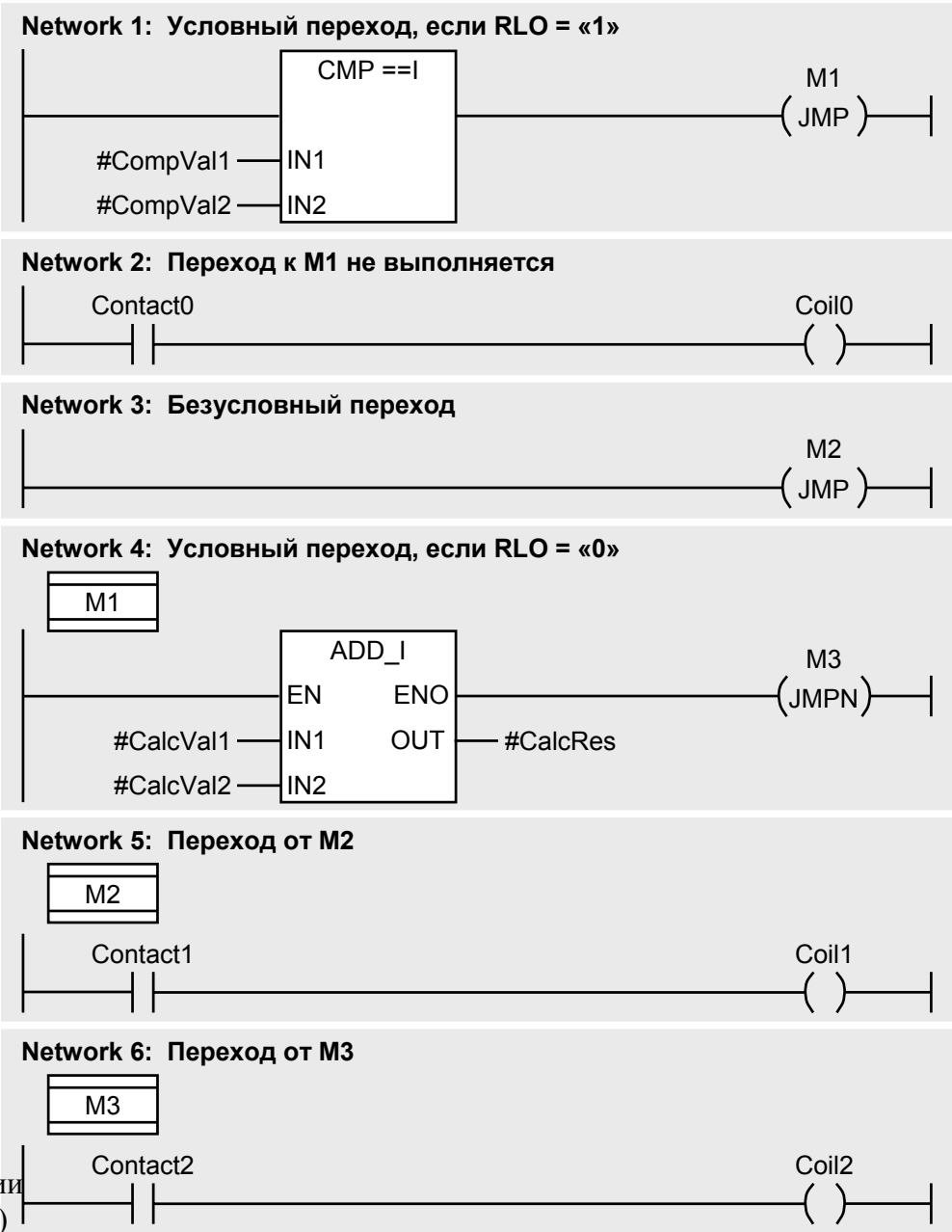


Рисунок 16.1
Пример функции
перехода (LAD)

16.3 Переход, если RLO = «1»

Условный переход при RLO = «1» выполняет функция JMP, чья катушка не имеет прямого соединения с левой направляющей (шиной питания) (LAD), или блочному элементу которой предшествует логическая операция (FBD). Логическая операция, предшествующая этой катушке, может быть выполнена в любом случае. Если RLO = «1» (в случае выполнения предшествующей логической операции), то CPU прерывает линейный поток программы и продолжает обработку в сети, обозначенной меткой перехода. Если предшествующая логическая операция не выполнена, то CPU продолжает выполнять программу в следующей сети.

Пример (рисунки 16.1 и 16.2): Если условие сравнения в сети 1 (Network 1) выполнено, то программа продолжит работу в сети 4 (Network 4). Если условие сравнения не удовлетворено, то обрабатывается следующая сеть, сеть 2 (Network 2).

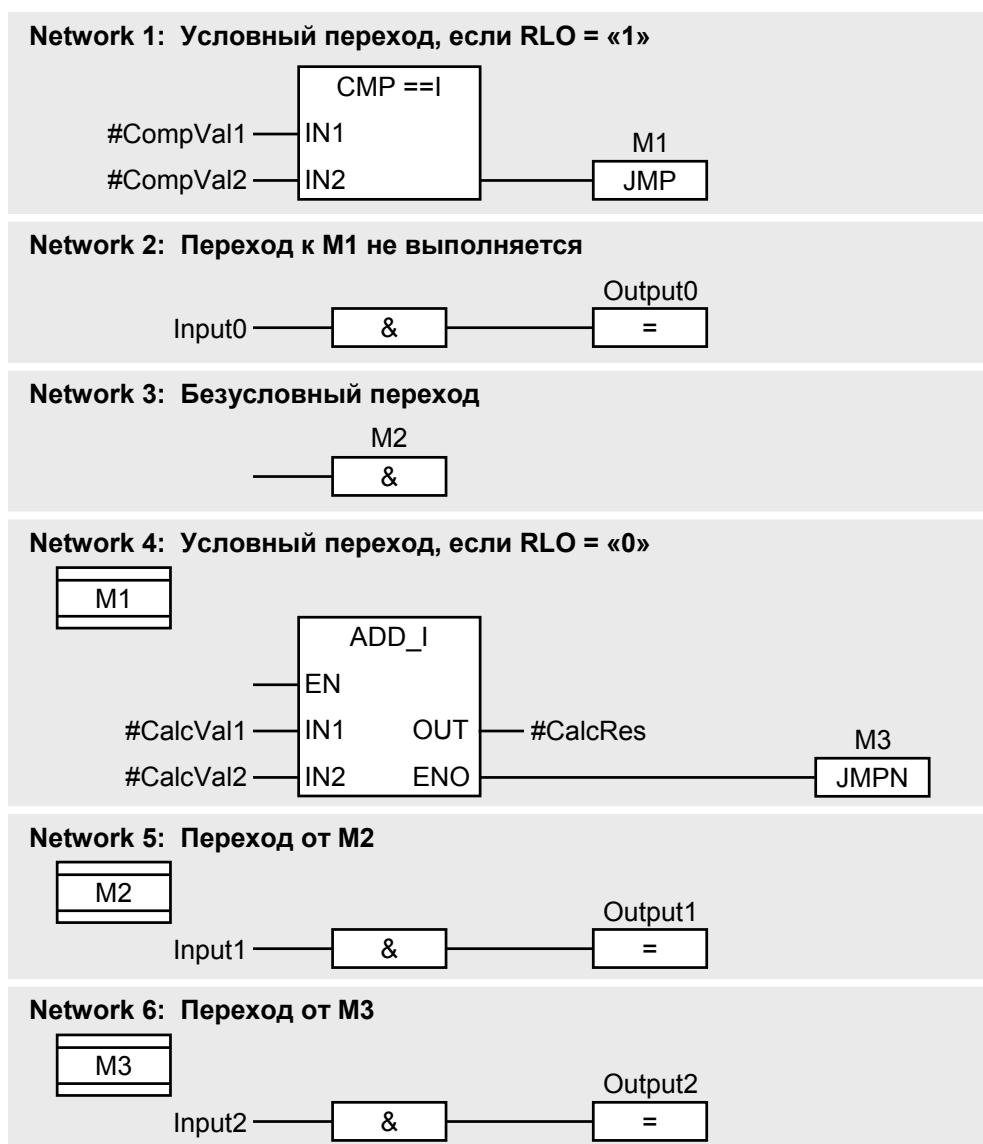


Рисунок 16.2 Пример функций перехода (FBD)

16.4 Переход, если $RLO = \langle 0 \rangle$

Условный переход при $RLO = \langle 0 \rangle$ выполняет функция JMPN, чья катушка не соединена напрямую с левой направляющей (шиной питания) (LAD), или блочному элементу которой предшествует логическая операция (FBD). Логическая операция, предшествующая этой катушке, может быть выполнена в любом случае. Если $RLO = \langle 0 \rangle$ (в случае неудовлетворения предшествующей логической операции), то CPU прерывает линейный поток программы и продолжает обработку в сети, обозначенной меткой перехода. Если предшествующая логическая операция выполнена, то CPU продолжает выполнять программу в следующей сети.

Пример (рисунки 16.1 и 16.2): Если сумматор в сети 4 (Network 4) сигнализирует об ошибке, то выполняется переход к сети 6 (Network 6) (метка перехода M3). Если ошибка не возникает, то обрабатывается следующая сеть, сеть 5 (Network 5).