Презентация по теме Язык релейных диаграмм(LD)

Выполнили: Саидзода С Д и Саидзода С К

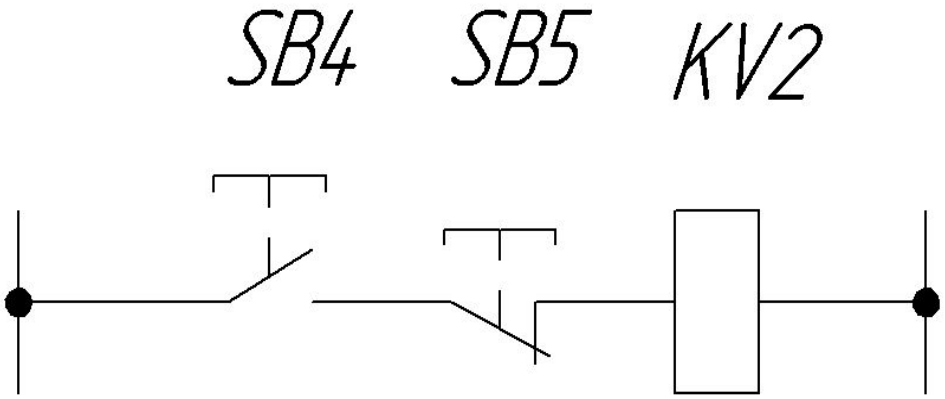
Язык релейных диаграмм**(LD)**

- Графический язык
- Программа состоит из схем
- Использовался для программирования практически всех классических ПЛК
- Удобен для программирования логических выражений
- Сложно использовать для работы с аналоговыми типами данных
- Переключение между FBD и LD

История появления языка **LD**

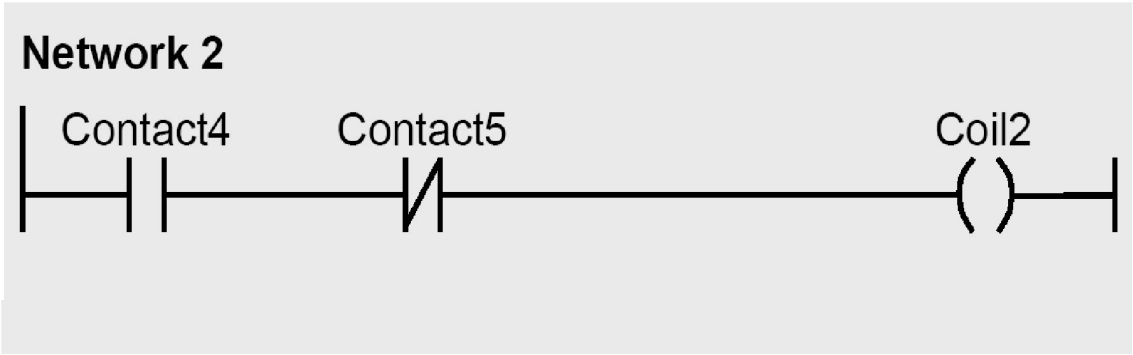
Необходимо было создать управляющее устройство, алгоритм работы которого можно было бы менять, не переделывая монтажную схему системы управления, и в результате возникла логичная идея заменить системы управления с «жесткой» логикой работы (совокупность реле, регуляторов, таймеров и т.д.) на автоматы с программно заданной логикой работы. Так родились ПЛК. Впервые ПЛК были применены в США для автоматизации конвейерного сборочного производства в автомобильной промышленности (1969г.). Поскольку в определении «программируемый логический контроллер» главным являлось «программируемый», то практически сразу возник вопрос, как программировать ПЛК? Идеальным вариантом могла бы стать автоматическая трансляция принципиальных схем релейных автоматов в программы для ПЛК. Почему бы и нет? Так в ПЛК появился язык релейно-контактных схем (РКС или LD в английских источниках Ladder Diagram). Специалист-технолог мог “перерисовать” схему управления на дисплее программирующей станции ПЛК. Естественно схема изображалась не графически а посредством условных символов.

Пример перехода от принципиальной схемы к схеме на языке **LD**



Фрагмент принципиальной схемы

Эта же схема на языке LD



Операции бинарной логики **(LD)**

Последовательные и параллельные схемы

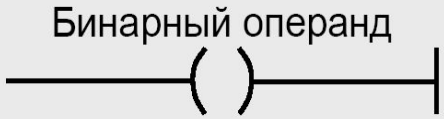
Бинарные сигнальные состояния группируются в LD (контактные планы) посредством последовательных (series) и параллельных (parallel) соединений контактов. Последовательное соединение соответствует функции AND (И), а параллельное со-единение – функции OR (ИЛИ). Вы будете использовать контакты для проверки сигнальных состояний двоичных операндов

LD использует два вида контактов для сканирования битовых операндов: NO-контакт и NC-контакт. Одиночная катушка, как терминатор (завершающий элемент) цепи назначает или направляет электрический ток напрямую к операнду, расположенному при катушке

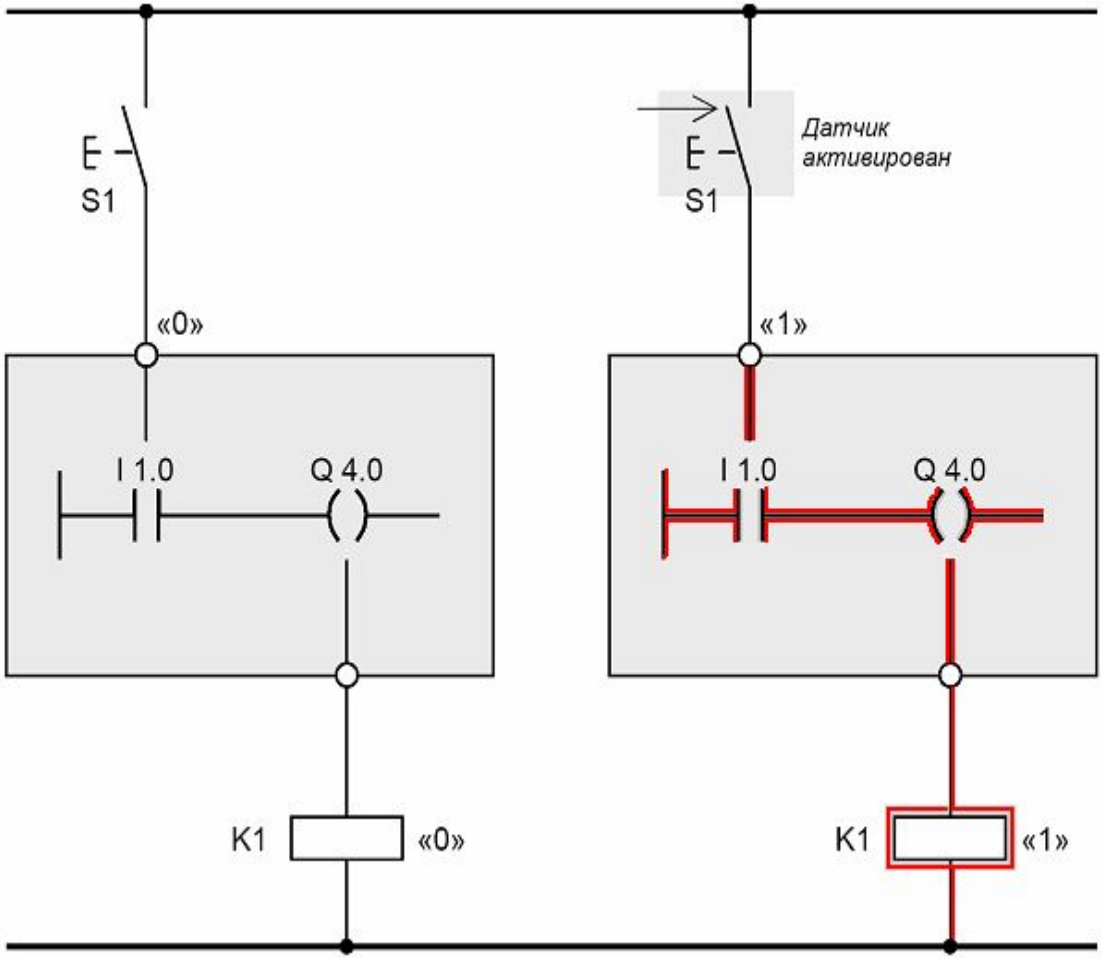
Нормально разомкнутый контакт (Normally-open, NO)

Нормально замкнутый контакт (Normally-closed, NC)

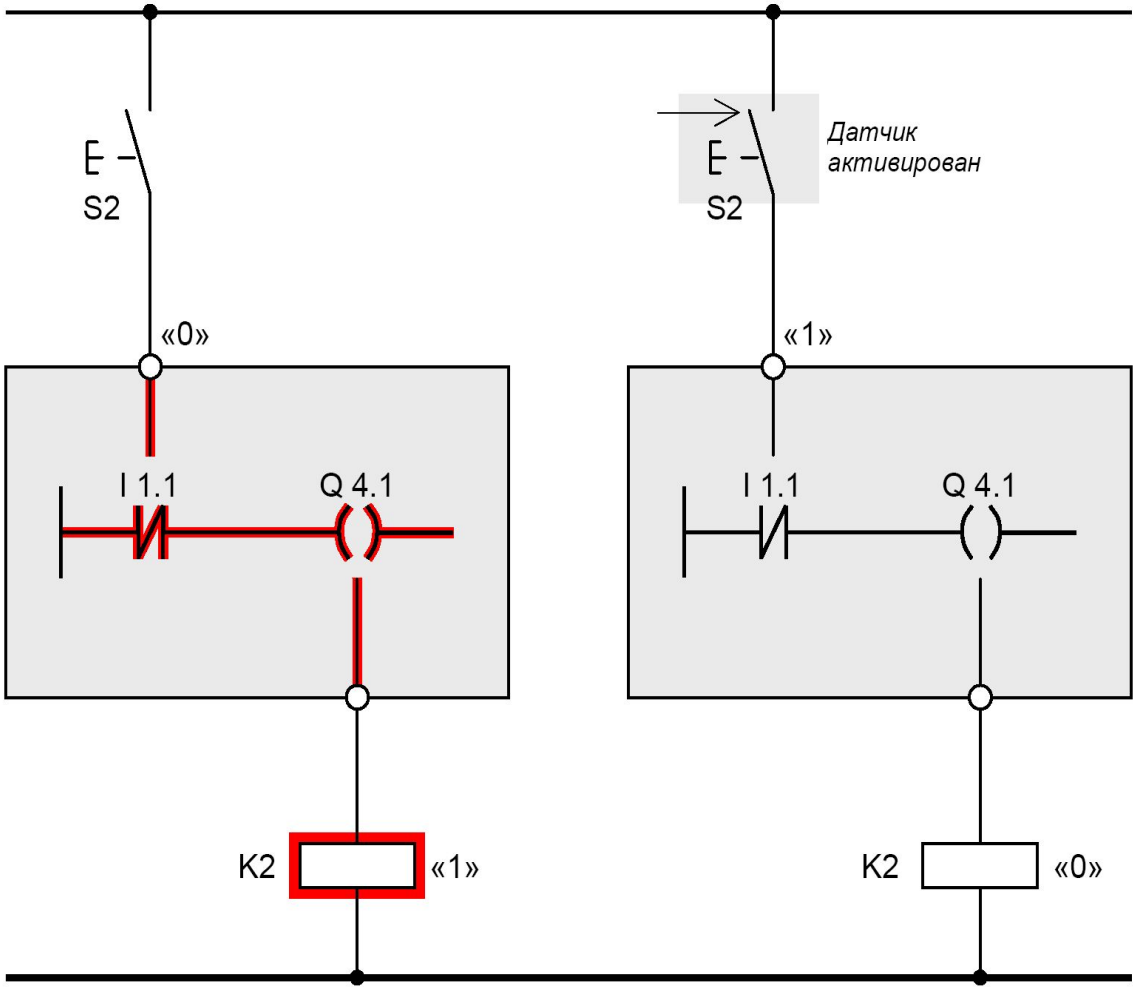
Одиночная катушка



Работа **NO-**контакта

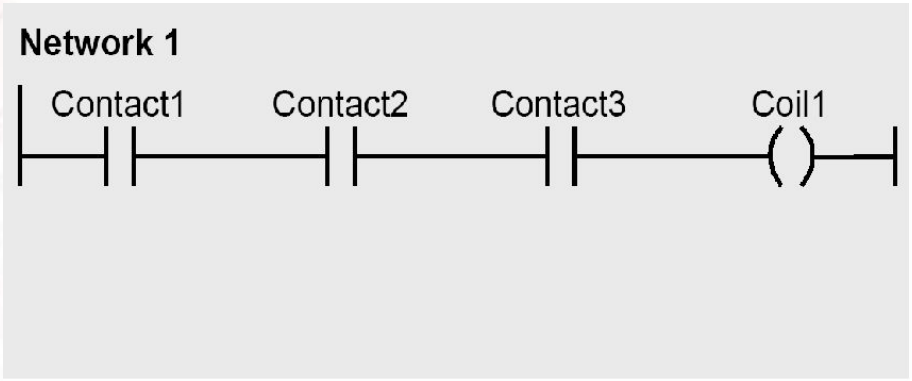


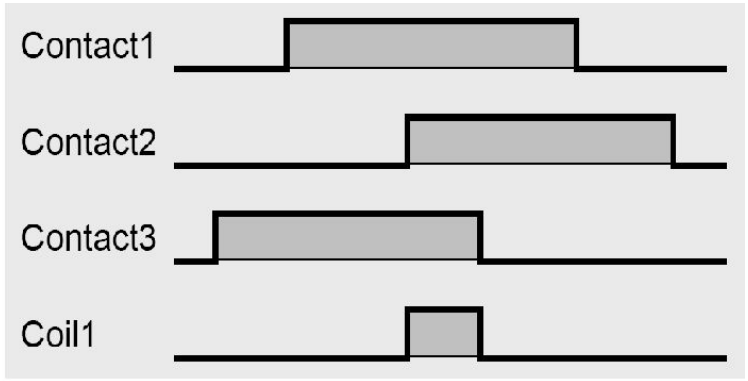
Работа **NC-**контакта

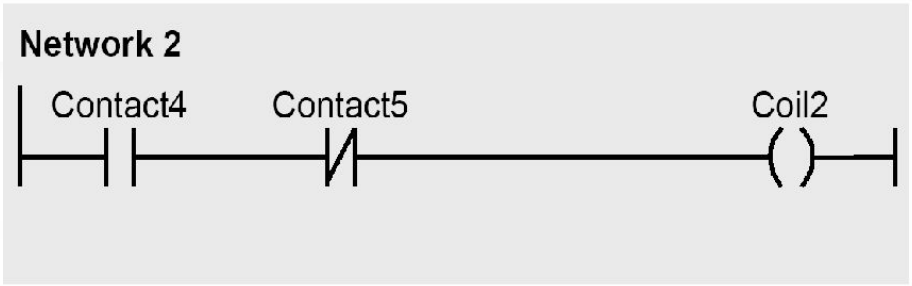


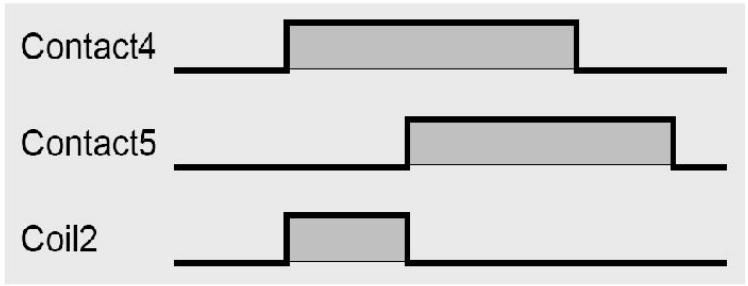
Последовательные схемы

В последовательных схемах два или более контактов соединены последовательно.
Ток в последовательной схеме течет, когда все контакты замкнуты.



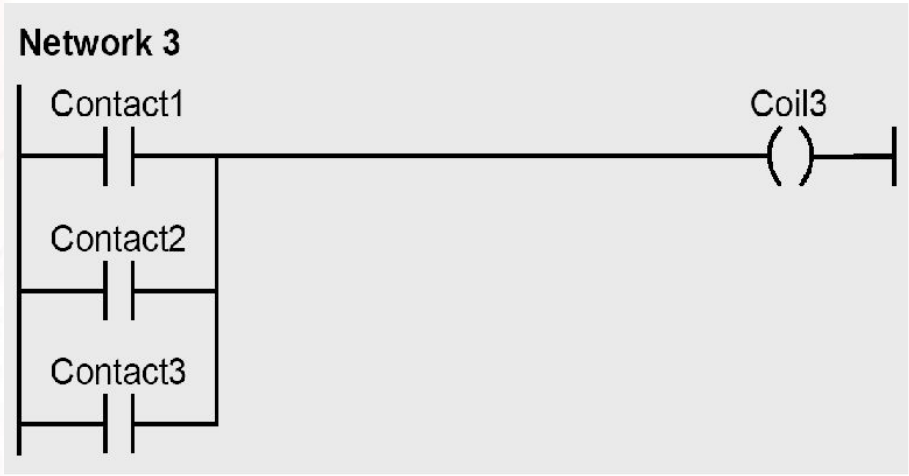


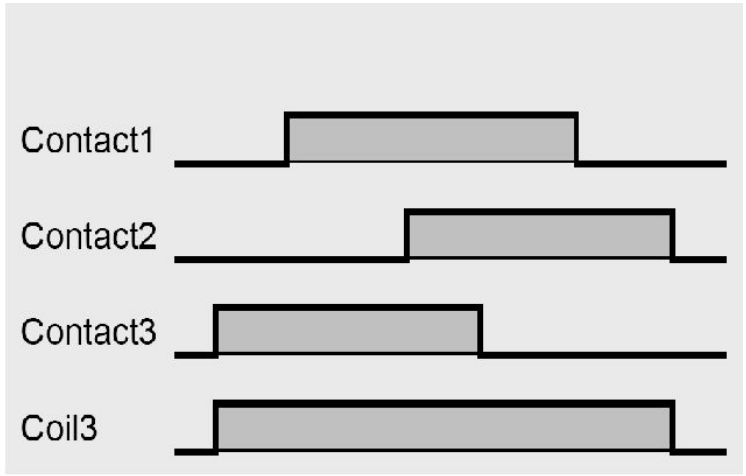


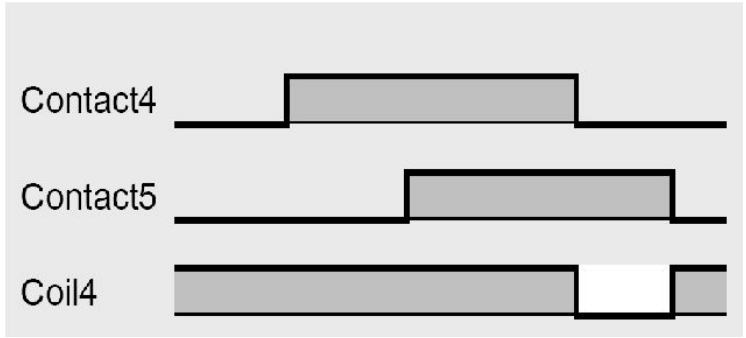


Параллельные схемы

Ток протекает через параллельную схему, если один из контактов замкнут.

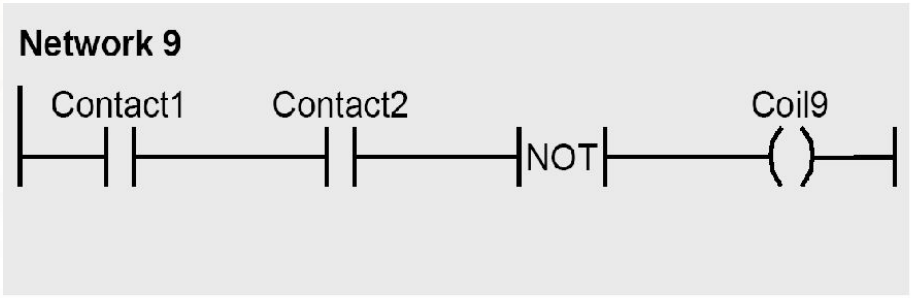


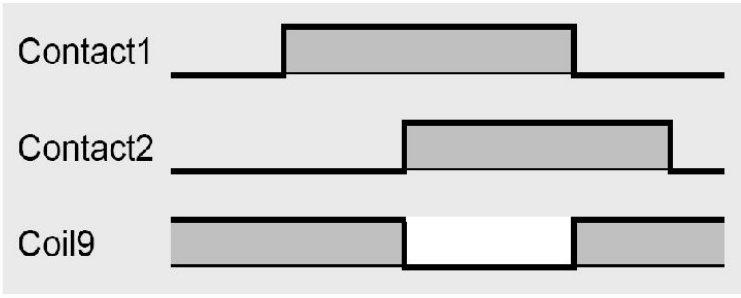


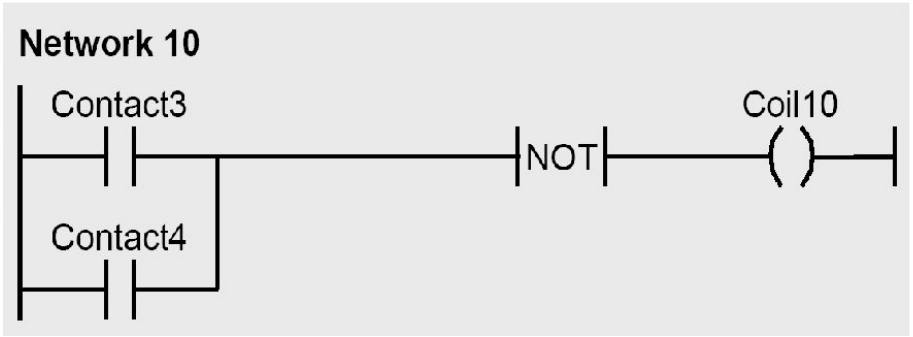


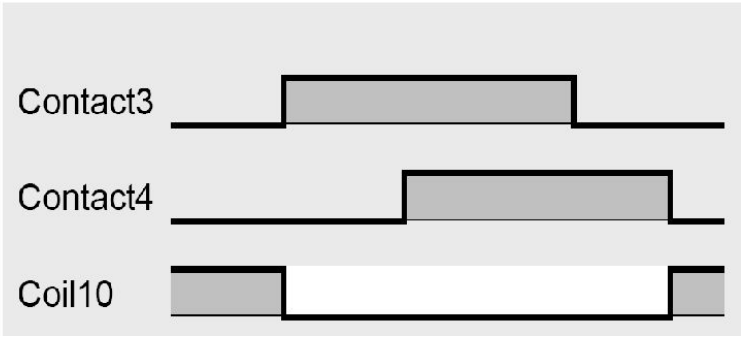
Инвертирование результата логической операции

NOT-контакт инвертирует результат логической операции









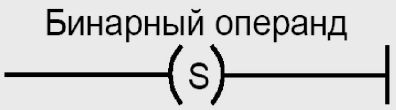
Катушки установки и сброса

Катушки установки и сброса (set coil, reset coil) также могут завершать цепь. Эти катушки становятся активными, только когда через них протекает ток.

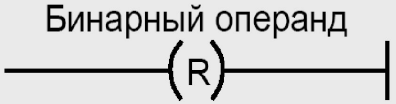
Если ток течет в катушке установки, то операнд над катушкой устанавливается в сигнальное состояние «1». Если ток течет в катушке сброса, то операнд над катушкой переустанавливается в сигнальное состояние «0» (сбрасывается). При отсутствии тока в катушке установки или сброса бинарный операнд остается без изменений

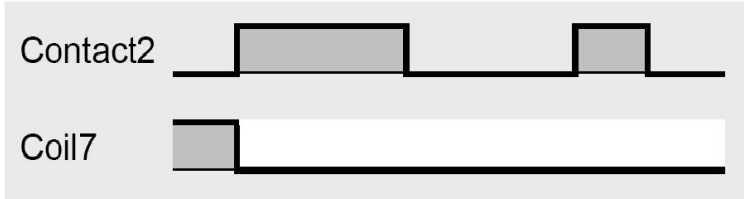
Диаграммы работы катушек установки и сброса

Катушка установки



Катушка сброса





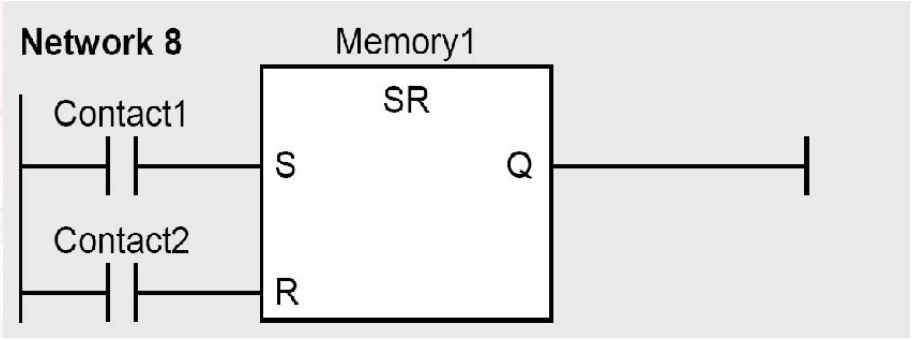
Блочный элемент памяти (триггер)

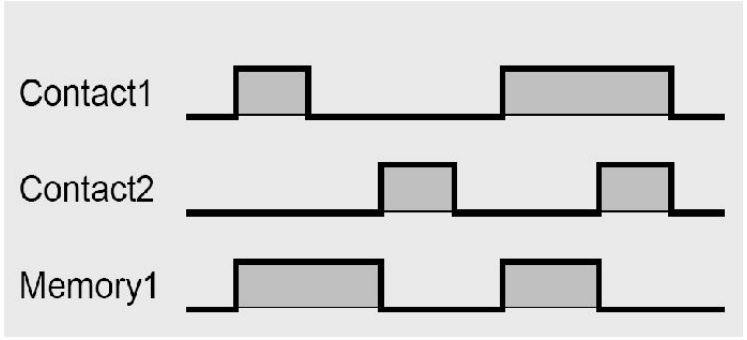
Функции катушек установки и сброса объединяются в блочном элементе функции для работы с памятью (memory box). Общий бинарный операнд располагается над блочным элементом. Вход S (set input) блочного элемента в данном случае соответствует катушке установки, вход R (reset input) – катушке сброса.

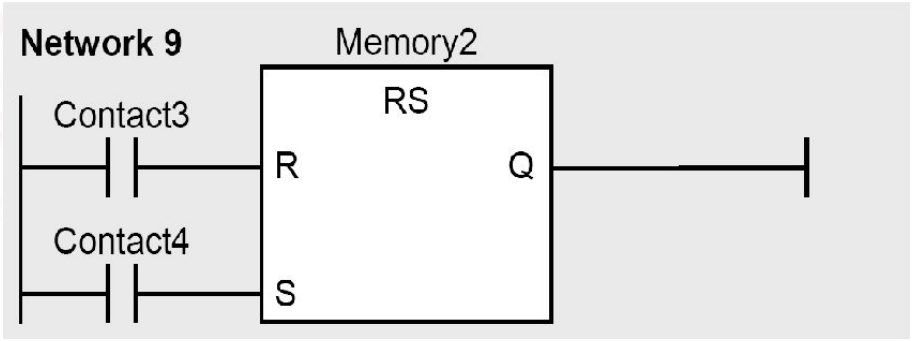
Блочный элемент SR

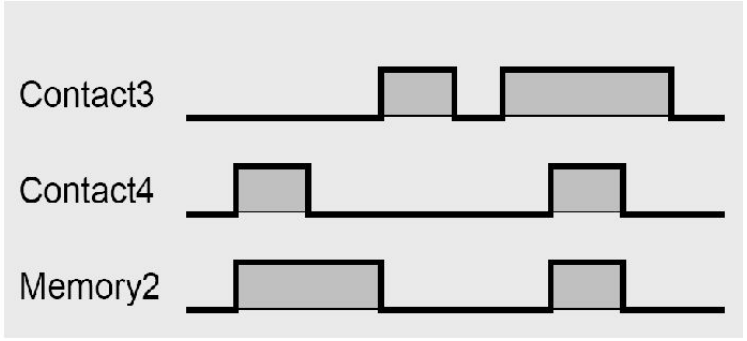
Блочный элемент RS

SR - триггер с приоритетом сброса
RS - триггер с приоритетом установки



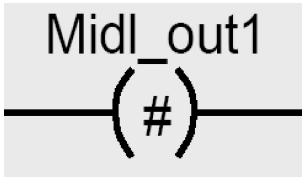




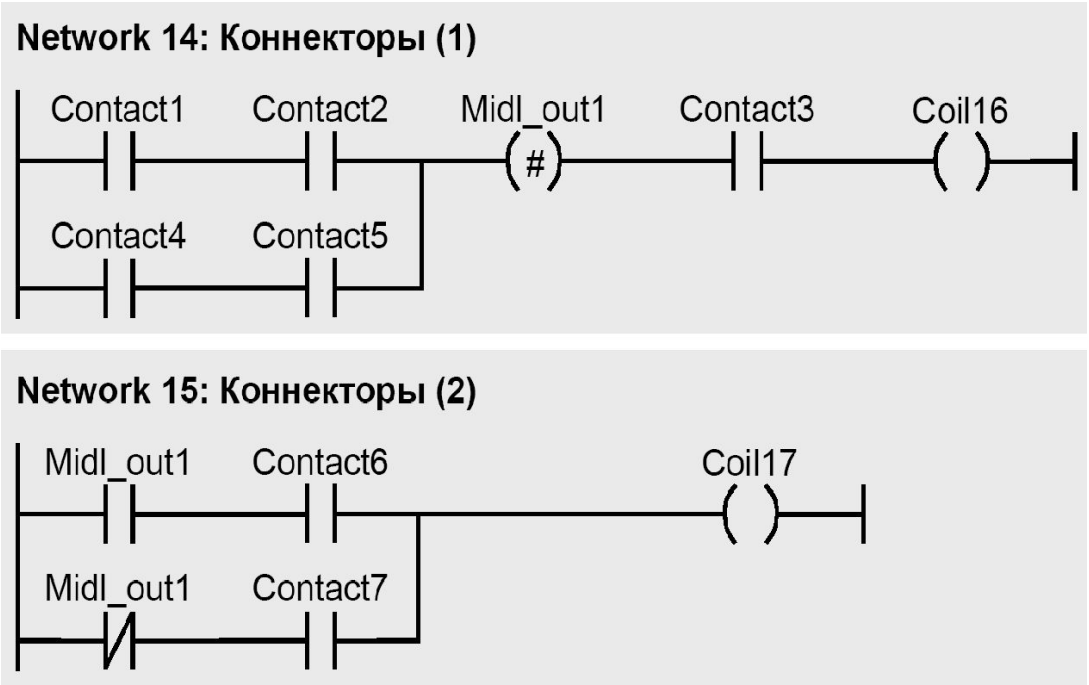


Коннекторы в LD

Коннектор является одиночной катушкой в цепи. RLO, действительный для этой точки (электрический ток, который течет в цепи, в данной точке), хранится в двоичном операнде над коннектором. Сам коннектор не оказывает влияния на электрический ток. Коннектор не может завершать цепь; для этой цели применяется одиночная катушка.



Пример использования коннекторов в LD



RLO из цепи, формируемый контактами *Contact1*, *Contact2*, *Contact4* и *Contact5*, сохраняется в коннекторе *Midl_out1*. Если условие логической операции выполняется (ток течет в коннекторе), и если *Contact3* замкнут, то *Coil16* возбуждается. Хранимый RLO используется в следующей сети (network 15) двумя способами. С одной стороны, производится проверка выполнения условия логической операции и битовой логической комбинации, осуществленной с *Contact6*, а с другой стороны, производится проверка невыполнения условия логической операции и битовой логической комбинации, осуществленной с *Contact7*.