

---

## Содержание главы 14

|                  |                                                       |          |
|------------------|-------------------------------------------------------|----------|
| <b><u>14</u></b> | <b><u>Побитовые логические операции</u></b> .....     | <b>4</b> |
| <u>14.1</u>      | <u>Выполнение побитовой логической операции</u> ..... | 4        |
| <u>14.2</u>      | <u>Описание побитовых логических операций</u> .....   | 8        |

# 14 Побитовые логические операции

Побитовые логические операции (числовая логика) комбинируют побитно значения двух переменных в соответствии с операциями AND, OR и исключающего OR. Логическая операция может применяться к словам или двойным словам. Доступные логические операции со словами приведены в таблице 14.1.

Таблица 14.1 Обзор логических операций со словами

| Логическая операция  | С переменной размером в слово | С переменной размером в двойное слово |
|----------------------|-------------------------------|---------------------------------------|
| AND (И)              | WAND_W                        | WAND_DW                               |
| OR (ИЛИ)             | WOR_W                         | WOR_DW                                |
| Исключающее OR (ИЛИ) | WXOR_W                        | WXOR_DW                               |

## 14.1 Выполнение побитовой логической операции

### Представление

Кроме разрешающего входа (enable input) EN и разрешающего выхода (enable output) ENO блочный элемент побитовой логической операции оснащен двумя входами IN1 и IN2 и одним выходом OUT. «Заголовок» блочного элемента идентифицирует выполняемую логическую операцию (например, WAND\_W означает объединение слов по AND).

#### Блочный элемент побитовой логической операции (в примере: объединение по AND)



Комбинируемые значения подаются на входы IN1 и IN2, результат операции находится на выходе OUT. Входы и выход имеют различные типы данных в зависимости от операции: WORD для операции со словами (16 бит) и DWORD для операций с двойными словами (32 бита). Применяемые переменные должны быть того же типа данных, что и входы или выход.

Описание битов различных форматов данных содержится в параграфе 3.5.4 «Простые типы данных».

## Функция

Побитовая логическая операция выполняется, когда на разрешающем входе присутствует «1» (когда через вход EN протекает ток). Если выполнение операции не допускается (EN сброшен в «0»), то операция не производится, и ENO также имеет значение «0».

| ЕСЛИ EN == "1" или не используется |                   |
|------------------------------------|-------------------|
| ТО                                 | ИНАЧЕ             |
| <b>OUT := IN1 Wlog IN2</b>         | <b>ENO := "0"</b> |
| <b>ENO := "1"</b>                  |                   |

Wlog – побитовая логическая операция

Если главное реле управления (MCR) активировано, то при исполнении побитовой логической операции (EN = «1») выход OUT установлен в ноль. MCR на выход ENO не воздействует.

Побитовая логическая операция генерирует результат побитно. Бит 0 входа IN1 комбинируется с битом 0 входа IN2, и результат сохраняется в бите 0 выхода OUT. То же самое происходит с битами 1, 2 и так далее до бита 15 или 31 соответственно. В таблице 14.2 показано формирование результата для отдельного бита.

В главе 15 «Биты состояния» объясняется, как побитные логические операции управляют битами состояния.

**Таблица 14.2** Формирование результата побитовой логической операций

|                                    |   |   |   |   |
|------------------------------------|---|---|---|---|
| Содержимое входа IN1               | 0 | 0 | 1 | 1 |
| Содержимое входа IN2               | 0 | 1 | 0 | 1 |
| Результат операции AND             | 0 | 0 | 0 | 1 |
| Результат операции OR              | 0 | 1 | 1 | 1 |
| Результат операции исключающего OR | 0 | 1 | 1 | 0 |

## Примеры

Рисунок 14.1 показывает по одному примеру для каждой побитовой логической операции.

При пошаговом программировании побитовые логические операции вы можете найти в каталоге программных элементов (Program Element Catalog), вызываемого командой меню View → Catalog (Вид → Каталог) [Ctrl + K] или Insert → Program Elements (Вставка → Программные элементы), в разделе «Word Logic» («Побитовая логика»).

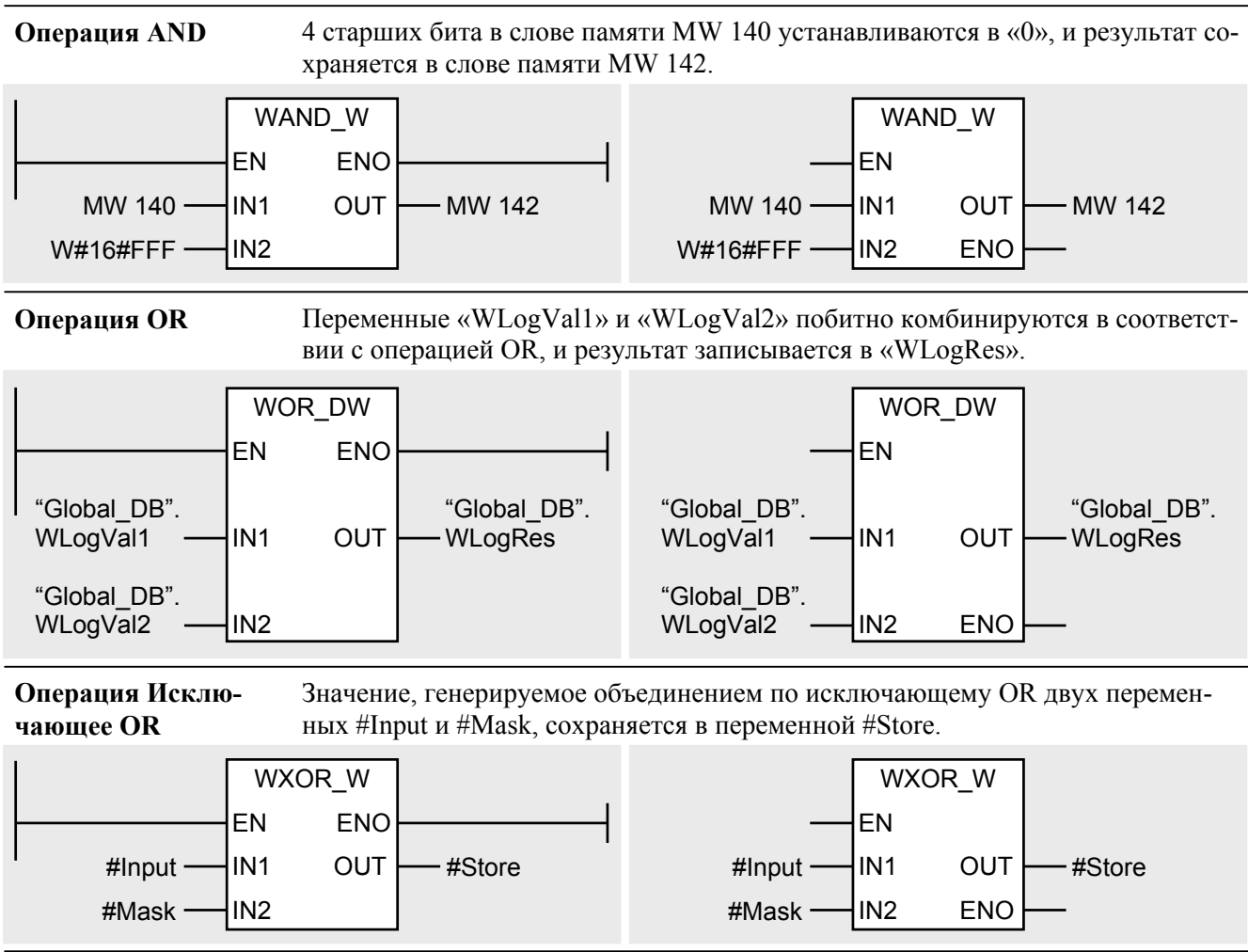


Рисунок 14.1 Примеры побитовых логических операций

Побитовые логические операции в цепи (LAD)

Вы можете соединить контакты перед входом EN и после выхода ENO последовательно и параллельно.

Сам блочный элемент побитовой логической операции может быть помещен после Т-ветви или вставлен в ветвь, соединенную непосредственно с левой направляющей (шиной питания). В этой ветви также могут быть контакты перед входом IN, и она не может быть самой верхней ветвью.

Прямое подключение к левой питающей направляющей позволяет вам соединить блочные элементы побитовой логики параллельно. Когда блочные элементы подключаются параллельно, требуется катушка для завершения цепи. Если вы не производите проверку на наличие ошибки, назначьте катушке «пустой» операнд, например, бит временных локальных данных.

Можно соединить блочные элементы побитовой логической операции последовательно. Если выход ENO предыдущего блочного элемента соединен с входом EN следующего, то последний обработается всегда. Если вы хотите использовать ре-

зультат из предыдущего блочного элемента в качестве входного значения для следующего блока, то переменные из области временных локальных данных предоставляют удобные промежуточные буферы.

Если вы компонуете несколько блочных элементов побитовой логики в одной цепи (параллельно левой питающей направляющей и в дальнейшем последовательно), то сначала обрабатываются блочные элементы самой верхней ветви слева направо, затем элементы второй ветви слева направо и так далее.

Библиотека «LAD\_Book» на дискете, прилагаемой к книге, содержит примеры представления и группировки операций побитовой логики (FB 114 в программе «Digital Functions», «Числовые функции»).

### **Побитовые операции в логической схеме (FBD)**

EN и ENO могут быть не назначены. Если требуется обработать блочный элемент побитовой логической операции в зависимости от определенных условий, то можно скомпоновать бинарные логические операции перед входом EN. Вы можете соединить выход ENO с бинарными входами других функций. Например, можно соединить блочные элементы побитовой логики последовательно, таким образом, выход ENO предыдущего блочного элемента будет подключен к входу EN следующего. Если вы хотите использовать результат вычислений предыдущего блочного элемента в качестве входного значения для последующего элемента, то переменные из области временных локальных данных предоставляют удобные промежуточные буферы.

Библиотека «FBD\_Book» на дискете, прилагаемой к книге, содержит примеры представления и группировки побитовых логических операций (FB 114 в программе «Digital Functions», «Числовые функции»).

## 14.2 Описание побитовых логических операций

### Операция AND

AND комбинирует отдельные биты значения на входе IN1 с соответствующими битами значения на входе IN2 в соответствии с правилами объединения по AND. Бит в результирующем слове OUT будет иметь значение «1», если соответствующие биты в обоих комбинируемых значениях равны «1».

Так как биты, равные «0» на входе IN2, также устанавливают соответствующие биты результата в «0» в независимости от значения этих битов на входе IN1, мы называем эти биты «маскированными». Маскирование – основное применение (числовой) операции AND.

### Операция OR

OR комбинирует отдельные биты значения на входе IN1 с соответствующими битами значения на входе IN2 в соответствии с правилами объединения по OR. Бит в результирующем слове OUT будет иметь значение «0», если соответствующие биты в обоих комбинируемых значениях равны «0».

Так как биты, равные «1» на входе IN2, также устанавливают соответствующие биты результата в «1» независимо от значения этих битов на входе IN1, мы называем эти биты «маскированными». Маскирование – основное применение (числовой) операции OR.

### Операция Исключающее OR

Исключающее OR комбинирует отдельные биты значения на входе IN1 с соответствующими битами значения на входе IN2 в соответствии с правилами объединения по исключающему OR. Бит в результирующем слове OUT будет иметь значение «1», если соответствующий бит только одного из двух комбинируемых значений равен «1». Если бит на входе IN2 равен «1», то соответствующий бит результата будет иметь обратное состояние по отношению к биту, расположенному в той же позиции на входе IN1.

В результате равны «1» будут только биты IN1 и IN2, имеющие противоположные сигнальные состояния до числовой операции исключающего OR. Основное использование (числовой) операции исключающего OR – обнаружение битов с противоположными сигнальными состояниями или «инвертирование» сигнальных состояний отдельных битов.