

# Команды IEC 1131-3

# 10

Эта глава описывает стандартные команды IEC 1131-3. Есть несколько команд SIMATIC, которые могут использоваться в программе IEC. Эти команды называются нестандартными командами IEC и представляются в начале каждого раздела.

## Обзор главы

Раздел	Описание	Стр.
10.1	Поразрядные логические команды IEC	10–2
10.2	Команды сравнения IEC	10–8
10.3	Команды таймеров IEC	10–11
10.4	Команды счетчиков IEC	10–15
10.5	Арифметические команды IEC	10–18
10.6	Числовые функции IEC	10–22
10.7	Команды пересылки IEC	10–24
10.8	Логические команды IEC	10–26
10.9	Команды сдвига и циклического сдвига IEC	10–28
10.10	Команды преобразования IEC	10–31

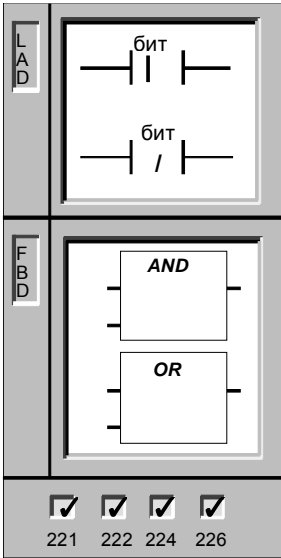
10.1 Поразрядные логические команды IEC

Таблица 10–1 дает постраничные ссылки для нестандартных поразрядных логических команд IEC.

Таблица 10–1. Нестандартные поразрядные логические команды IEC

Описание	Страница
Стандартные контакты	9–2, 10–2
Контакты непосредственного опроса	9–3
Контакт НЕ (NOT)	9–4
Положительный и отрицательный фронт	9–4
Выход	9–6
Непосредственный выход	9–6
Установка и сброс (N битов)	9–7

Стандартные контакты (нестандартные команды IEC 1131-3)



**Нормально разомкнутый** контакт замыкается (включается), когда бит равен 1.

**Нормально замкнутый** контакт замыкается (включается), когда бит равен 0.

Эти команды получают указанное значение из памяти или из регистра образа процесса, если память имеет тип I или Q.

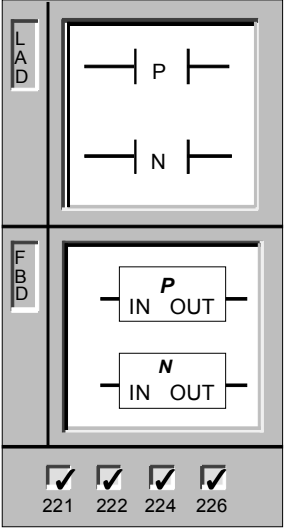
В LAD команды нормально разомкнутого и нормально замкнутого контактов представляются в виде контактов.

В FBD команды, соответствующие нормально разомкнутому контакту, представляются в виде AND/OR-блоков [И/ИЛИ-блоков]. Эти команды можно использовать для манипулирования булевыми сигналами тем же самым способом, что и контакты многозвенной логической схемы.

Команды, соответствующие нормально замкнутому контакту, тоже представляются в виде блоков. Команда нормально замкнутого контакта строится путем размещения символа отрицания на отметке входного сигнала. Число входов как AND-блока, так и OR-блока может расширяться максимум до семи.

Входы/выходы	Операнды	Типы данных
Бит (LAD)	I, Q, M, SM, T, C, V, S, L	BOOL
Вход (LAD)	Поток сигнала	BOOL
Входы (FBD)	I, Q, M, SM, T, C, V, S, L, поток сигнала	BOOL
Выход (LAD, FBD)	I, Q, M, SM, T, C, V, S, L, поток сигнала	BOOL

Положительный, отрицательный фронт



Контакт **Положительный фронт** позволяет сигналу протекать в течение одного цикла обработки программы при каждом переходе "выключено → включено".

Контакт **Отрицательный фронт** позволяет сигналу протекать в течение одного цикла обработки программы при каждом переходе "включено → выключено".

В LAD команды положительного и отрицательного переходов представляются в виде контактов.

В FBD эти команды представляются блоками P и N.

Входы/выходы	Операнды	Типы данных
IN (LAD)	Поток сигнала	BOOL
IN (FBD)	I, Q, M, SM, T, C, V, S, L, поток сигнала	BOOL
OUT (LAD)	Поток сигнала	BOOL
OUT (FBD)	I, Q, M, SM, T, C, V, S, L, поток сигнала	BOOL

Примеры контактов

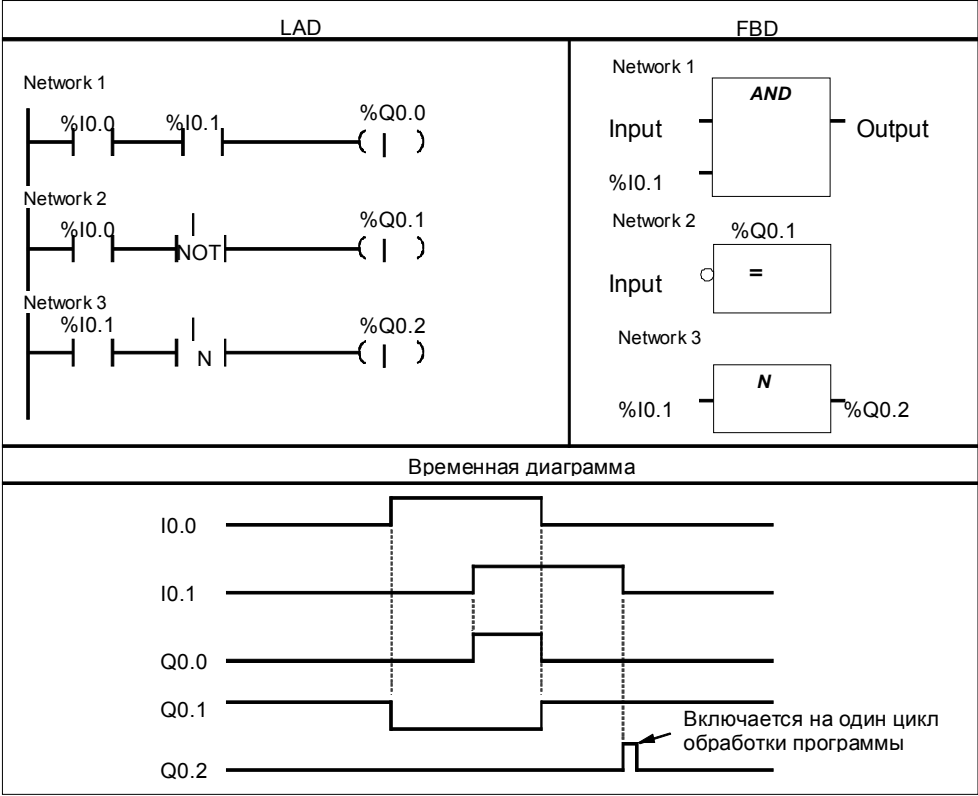
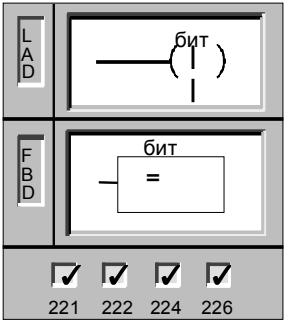


Рис. 10–1. Примеры команд булевых контактов для LAD и FBD

Вывод



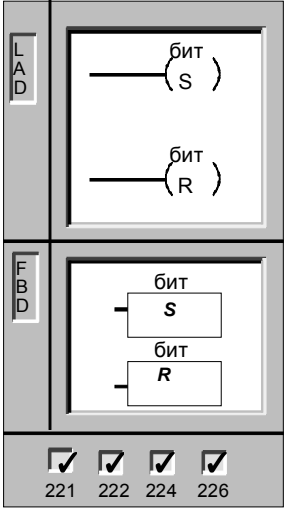
Когда выполняется команда **Вывод**, включается выход.

В форме LAD команда Вывод представляется в виде катушки.

В форме FBD эта команда представляется в виде блока присваивания (=).

Входы/выходы	Операнды	Типы данных
Бит (LAD/FBD)	I, Q, M, SM, T, C, V, S, L	BOOL
Вход (LAD)	Поток сигнала	BOOL
Вход (FBD)	I, Q, M, SM, T, C, V, S, L, поток сигнала	BOOL

Установка, сброс



Когда выполняются команды **Установка** и **Сброс**, устанавливается или сбрасывается значение, указываемое битом или параметром OUT.

Ошибки, устанавливающие ENO в 0: SM4.3 (этап выполнения), 0006 (косвенная адресация), 0091 (операнд вне диапазона).

Входы/выходы	Операнды	Типы данных
Бит (LAD, FBD)	I, Q, M, SM, T, C, V, S, L	BOOL
Вход (LAD)	Поток сигнала	BOOL
Вход (FBD)	I, Q, M, SM, T, C, V, S, L, поток сигнала	BOOL

Примеры вывода

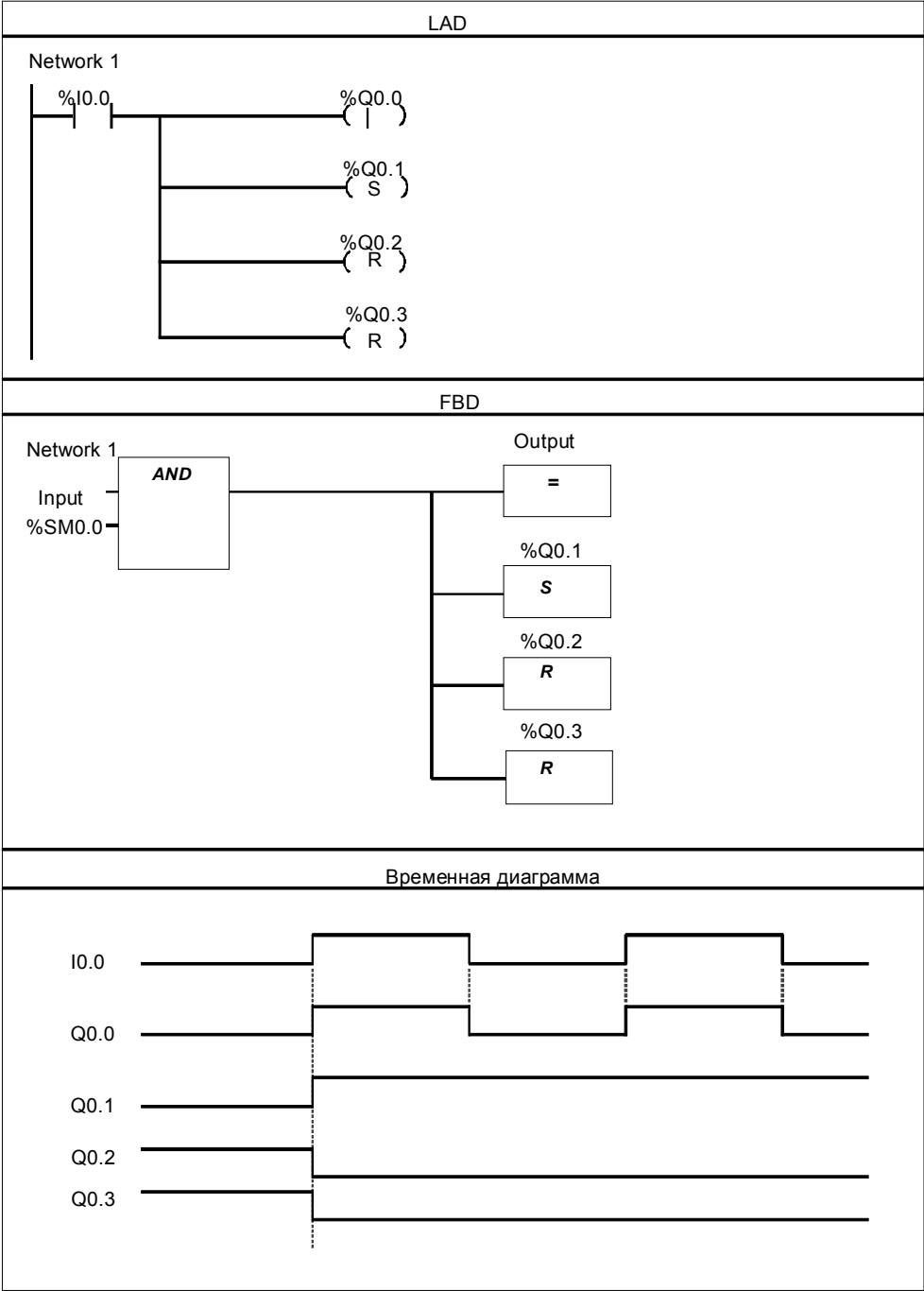
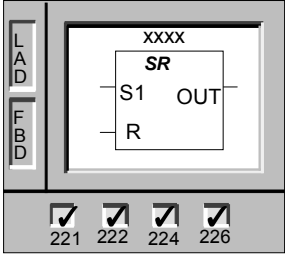


Рис. 10–2. Примеры команд вывода для LAD и FBD

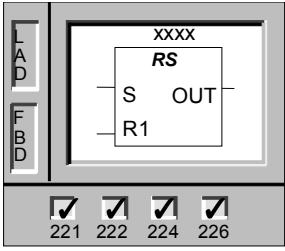
### Триггер с приоритетом установки



**Триггер с приоритетом установки** представляет собой фиксатор, где преобладает установка. Если истинны и сигнал установки (S1), и сигнал сброса (R), то выход (OUT) будет истинным. Параметр функционального блока xxxx определяет булев параметр, который устанавливается или сбрасывается. Необязательный выход отображает состояние сигнала параметра xxxx.

Входы/выходы	Операнды	Типы данных
S1, R (LAD)	Поток сигнала	BOOL
S1, R (FBD)	I, Q, M, SM, T, C, V, S, поток сигнала	BOOL
OUT (LAD)	Поток сигнала	BOOL
OUT (FBD)	I, Q, M, SM, T, C, V, S, L, поток сигнала	BOOL
xxxx	I, Q, M, V, S	BOOL

### Триггер с приоритетом сброса



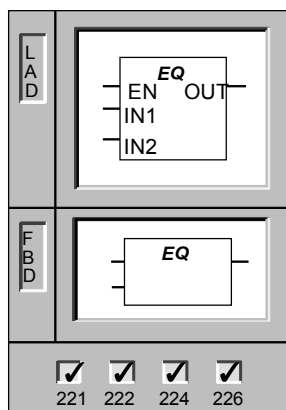
**Триггер с приоритетом сброса** представляет собой фиксатор, где преобладает сброс. Если истинны и сигнал установки (S), и сигнал сброса (R1), то выход (OUT) будет ложным. Параметр функционального блока xxxx определяет булев параметр, который устанавливается или сбрасывается. Необязательный выход отображает состояние сигнала параметра xxxx.

Входы/выходы	Операнды	Типы данных
S, R1 (LAD)	Поток сигнала	BOOL
S, R1 (FBD)	I, Q, M, SM, T, C, V, S, L, поток сигнала	BOOL
OUT (LAD)	Поток сигнала	BOOL
OUT (FBD)	I, Q, M, SM, T, C, V, S, L, поток сигнала	BOOL
xxxx	I, Q, M, V, S	BOOL

## 10.2 Команды сравнения IEC

Нестандартных команд сравнения IEC не существует.

### Сравнение на “равно”

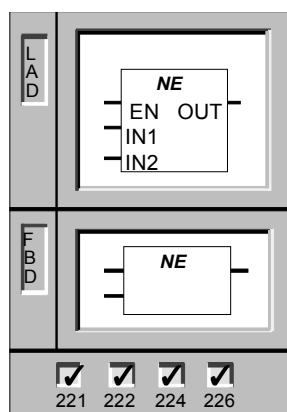


Функция **Сравнение на “равно”** сравнивает IN1 и IN2 с образованием булева результата, помещаемого в OUT. Типы входных и выходных данных могут изменяться, но должны быть одинаковыми.

Сравниваемые байты не имеют знака.  
Сравниваемые целые числа, двойные целые числа и вещественные числа имеют знак.  
Операции времени - целые числа со знаком.

Входы/выходы	Операнды	Типы данных
Входы (LAD и FBD)	IB, QB, MB, SB, SMB, VB, LB, IW, QW, MW, SW, SMW, VW, LW, T, C, AIW, ID, QD, MD, SD, SMD, VD, LD, HC, AC, константа, *VD, *AC, *LD	BYTE, INT, DINT, REAL
OUT (только LAD)	Поток сигнала	BOOL
OUT (только FBD)	I, Q, M, SM, T, C, V, S, L, поток сигнала	BOOL

### Сравнение на “не равно”



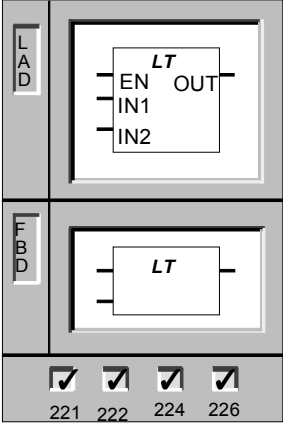
Функция **Сравнение на “не равно”** сравнивает IN1 и IN2 с образованием булева результата, помещаемого в OUT. Типы входных и выходных данных могут изменяться, но должны быть одинаковыми.

Сравниваемые байты не имеют знака.  
Сравниваемые целые числа, двойные целые числа и вещественные числа имеют знак.  
Операции времени - целые числа со знаком.

Входы/выходы	Операнды	Типы данных
Входы (LAD & FBD)	IB, QB, MB, SB, SMB, VB, LB, IW, QW, MW, SW, SMW, VW, LW, T, C, AIW, ID, QD, MD, SD, SMD, VD, LD, HC, AC, константа, *VD, *AC, *LD	BYTE, INT, DINT, REAL
OUT (только LAD)	Поток сигнала	BOOL
OUT(только FBD)	I, Q, M, SM, T, C, V, S, L, поток сигнала	BOOL



Сравнение на “меньше”

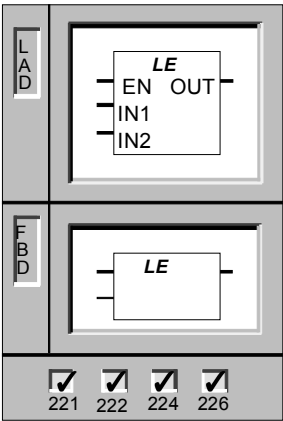


Функция **Сравнение на “меньше”** проверяет, действительно ли IN1 меньше, чем IN2, с образованием булева результата, помещаемого в OUT. Типы входных и выходных данных могут изменяться, но должны быть одинаковыми.

Сравниваемые байты не имеют знака.  
Сравниваемые целые числа, двойные целые числа и вещественные числа имеют знак.  
Операции времени - целые числа со знаком.

Входы/выходы	Операнды	Типы данных
Входы (LAD & FBD)	IB, QB, MB, SB, SMB, VB, LB, IW, QW, MW, SW, SMW, VW, LW, T, C, AIW, ID, QD, MD, SD, SMD, VD, LD, HC, AC, константа, *VD, *AC, *LD	BYTE, INT, DINT, REAL
OUT (только LAD)	Поток сигнала	BOOL
OUT (только FBD)	I, Q, M, SM, V, S, L, поток сигнала	BOOL

Сравнение на “меньше или равно”

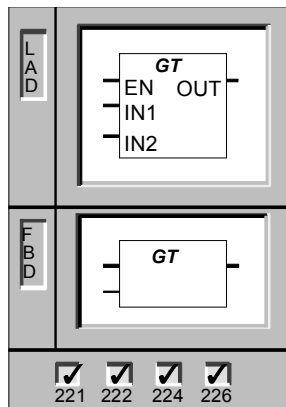


Функция **Сравнение на “меньше или равно”** проверяет, действительно ли IN1 меньше или равен IN2, с образованием булева результата, помещаемого в OUT. Типы входных и выходных данных могут изменяться, но должны быть одинаковыми.

Сравниваемые байты не имеют знака.  
Сравниваемые целые числа, двойные целые числа и вещественные числа имеют знак.  
Операции времени - целые числа со знаком.

Входы/выходы	Операнды	Типы данных
Входы (LAD & FBD)	IB, QB, MB, SB, SMB, VB, LB, IW, QW, MW, SW, SMW, VW, LW, T, C, AIW, ID, QD, MD, SD, SMD, VD, LD, HC, AC, константа, *VD, *AC, *LD	BYTE, INT, DINT, REAL
OUT (только LAD)	Поток сигнала	BOOL
OUT (только FBD)	I, Q, M, SM, V, S, L, поток сигнала	BOOL

## Сравнение на “больше”

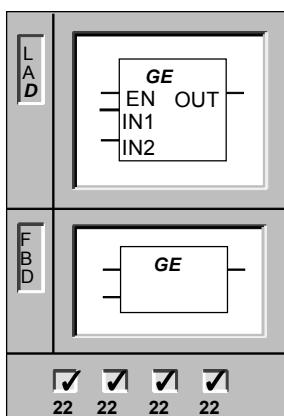


Функция **Сравнение на “больше”** проверяет, действительно ли IN1 больше, чем IN2, с образованием булева результата, помещаемого в OUT. Типы входных и выходных данных могут изменяться, но должны быть одинаковыми.

Сравниваемые байты не имеют знака.  
Сравниваемые целые числа, двойные целые числа и вещественные числа имеют знак.  
Операции времени - целые числа со знаком.

Входы/выходы	Операнды	Типы данных
Входы (LAD & FBD)	IB, QB, MB, SB, SMB, VB, LB, IW, QW, MW, SW, SMV, VW, LW, T, C, AIW, ID, QD, MD, SD, SMD, VD, LD, HC, AC, константа, *VD, *AC, *LD	BYTE, INT, DINT, REAL
OUT (только LAD)	Поток сигнала	BOOL
OUT (только FBD)	I, Q, M, SM, V, S, L, поток сигнала	BOOL

## Сравнение на “больше или равно”



Функция **Сравнение на “больше или равно”** проверяет, действительно ли IN1 больше или равен IN2, с образованием булева результата, помещаемого в OUT. Типы входных и выходных данных могут изменяться, но должны быть одинаковыми.

Сравниваемые байты не имеют знака.  
Сравниваемые целые числа, двойные целые числа и вещественные числа имеют знак.

Входы/выходы	Операнды	Типы данных
Входы (LAD & FBD)	IB, QB, MB, SB, SMB, VB, LB, IW, QW, MW, SW, SMV, VW, LW, T, C, AIW, ID, QD, MD, SD, SMD, VD, LD, HC, AC, константа, *VD, *AC, *LD	BYTE, INT, DINT, REAL
OUT (только LAD)	Поток сигнала	BOOL
OUT (только FBD)	I, Q, M, SM, V, S, L, поток сигнала	BOOL

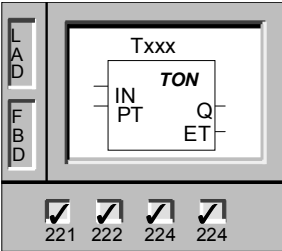
10.3 Команды таймеров IEC

Таблица 10–2 дает постраничные ссылки для нестандартных таймерных команд IEC.

Таблица 10–2. Нестандартные таймерные команды IEC

Описание	Страница
Таймер с задержкой включения с запоминанием	9–15

Таймер с задержкой включения

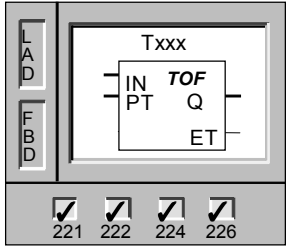


Функциональный блок **Таймер с задержкой включения** ведет отсчет времени до предварительно заданного значения с момента, когда разрешающий вход (IN) становится истинным. Когда истекшее время (ET) больше или равно заданному времени (PT), выходной бит таймера (Q) включается.

Выходной бит сбрасывается, когда разрешающий вход становится ложным. Когда заданное время (PT) достигнуто, отсчет времени прекращается и таймер блокируется.

Входы/выходы	Операнды	Типы данных
IN (LAD)	Поток сигнала	BOOL
IN (FBD)	I, Q, M, SM, T, C, V, S, L, поток сигнала	BOOL
PT (LAD & FBD)	VW, IW, QW, MW, SMW, LW, SW, AIW, AC, константа, *VD, *AC, *LD	INT
Q (LAD & FBD)	I, Q, M, SM, V, S, L	BOOL
ET (LAD & FBD)	VW, IW, QW, MW, SMW, LW, SW, AQW, AC, *VD, *AC, *LD	INT
Txxx	Константа, обратитесь к таблице 10–3, страница 10–12.	TON

Таймер с задержкой выключения

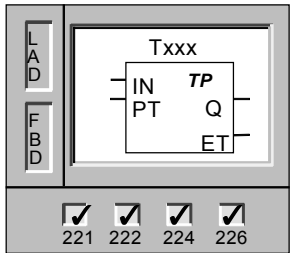


Функциональный блок **Таймер с задержкой выключения** используется для того, чтобы задерживать установку выходного сигнала "ложь" на установленный период времени после того, как входной сигнал примет значение "ложь". Он ведет отсчет времени до предварительно установленного значения с момента, когда разрешающий вход (IN) примет значение "ложь".

Когда истекшее время (ET) больше или равно заданному времени (PT), выходной бит таймера (Q) включается. Как только предварительная установка достигнута, выходной бит таймера переключается в состояние "ложь" и истекшее время сохраняется, пока разрешающий вход (IN) не переключится в состояние "истина". Если разрешающий вход (IN) переключается в состояние "ложь" на время, более короткое, чем заданное время (PT), то выходной бит остается в состоянии "истина". Для получения информации о номерах и разрешающих способностях таймеров обратитесь к таблице 10–3.

Входы/выходы	Операнды	Типы данных
IN (LAD)	Поток сигнала	BOOL
IN (FBD)	I, Q, M, SM, T, C, V, S, L, поток сигнала	BOOL
PT (LAD & FBD)	VW, IW, QW, MW, SMW, LW, SW, AIW, AC, константа, *VD, *AC, *LD	INT
Q (LAD & FBD)	I, Q, M, SM, V, S, L	BOOL
ET (LAD & FBD)	VW, IW, QW, MW, SMW, LW, SW, AQW, AC, *VD, *AC, *LD	INT
Txxx	Константа, обратитесь к таблице 10–3, страница 10–12.	TOF

Импульсный таймер



Функциональный блок **Импульсный таймер** используется для генерирования импульсов заданной длительности. Когда разрешающий вход (IN) становится истинным, выходной бит таймера (Q) включается. Выходной бит для заданного импульса остается истинным в течение предварительно установленного времени (PT). Как только истекшее время (ET) достигнет предварительной установки (PT), выходной бит (Q) становится ложным.

Истекшее время (ET) сохраняется, пока разрешающий вход (IN) не станет ложным. Как только выходной бит (Q) становится истинным, он остается истинным, пока не истекло время импульса (PT). Для получения информации о номерах и разрешающих способностях таймеров обратитесь к таблице 10–3.

Входы/выходы	Операнды	Типы данных
IN LAD)	Поток сигнала	BOOL
IN (FBD)	I, Q, M, SM, T, C, V, S, L, поток сигнала	BOOL
PT (LAD & FBD)	VW, IW, QW, MW, SMW, LW, SW, AIW, AC, константа, *VD, *AC, *LD	INT
Q (LAD & FBD)	I, Q, M, SM, S, V, L	BOOL
ET (LAD & FBD)	VW, IW, QW, MW, SW, SMW, LW, AQW, AC, *VD, *AC, *LD	INT
Txxx	Константа, обратитесь к таблице 10–3	TP

Описание команд таймеров IEC 1131-3

Таймеры TON, TOF и TP доступны с тремя разрешающими способностями. Разрешающая способность определяется номером таймера и представлена в таблице 10–3. Каждое отсчет текущего значения кратен базе времени. Например, значение 50 в 10–миллисекундном таймере представляет величину 500 мс.

Таблица 10–3. Номера и разрешающие способности таймеров

Тип таймера	Разрешение миллисекундах (мс)	Максимальное значение в секундах (с)	Номер таймера
TON, TOF, TP	1 мс	32,767 с	T32, T96
	10 мс	327,67 с	T33 - T36, T97 - T100
	100 мс	3276,7 с	T37 - T63, T101 - T255

Примечание

Нельзя совместно использовать одинаковые номера таймеров для TOF, TP и TON. Например, вы не можете иметь TON T32 и TOF T32.

Пример таймера с задержкой включения

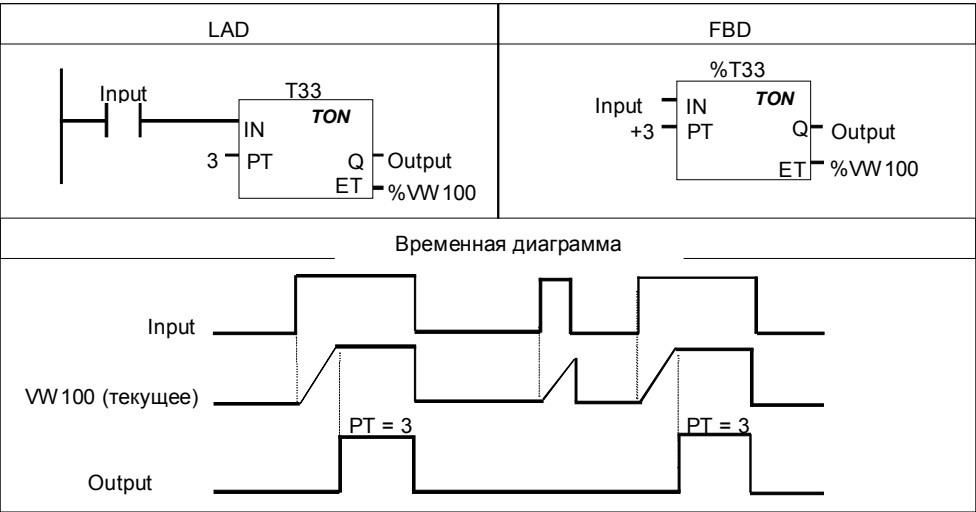


Рис. 10–3. Пример таймера с задержкой включения для LAD и FBD

Пример таймера с задержкой выключения

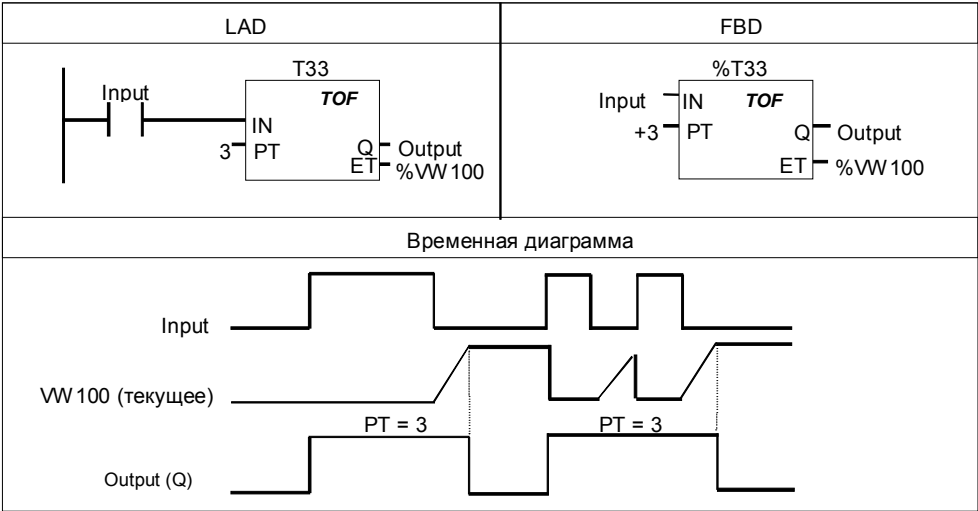


Рис. 10–4. Пример таймера с задержкой выключения для LAD и FBD

Пример импульсного таймера

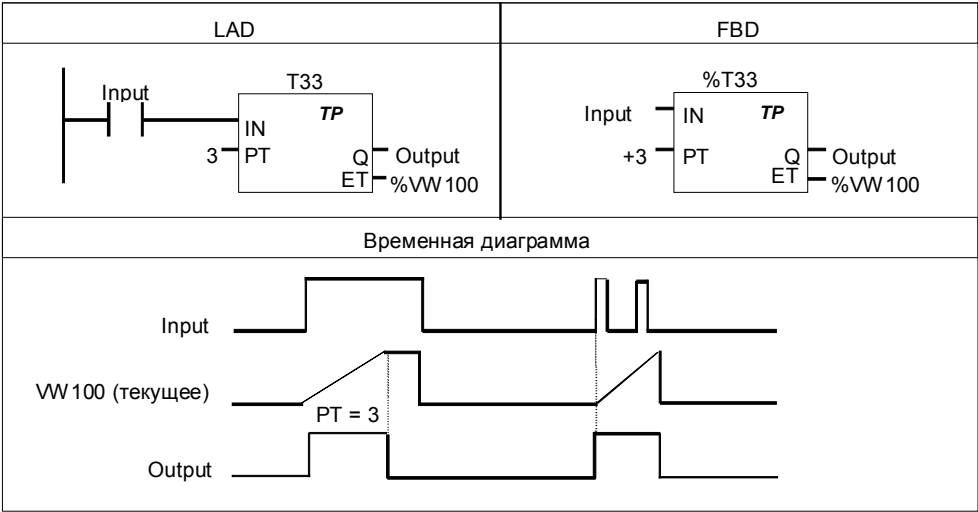


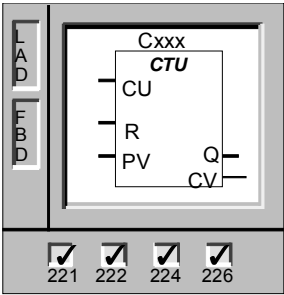
Рис. 10–5. Пример команды импульсного таймера для LAD и FBD

### 10.4 Команды счетчиков IEC

Таблица 10–4 дает постраничные ссылки для нестандартных команд счетчиков IEC.

Таблица 10–4 Нестандартные команды счетчиков IEC	
Описание	Страница
Скоростной счетчик	9–27
Определение скоростного счетчика	9–27
Импульсный вывод	9–50

#### Суммирующий счетчик



Функциональный блок **Суммирующий счетчик** ведет счет вверх от текущего значения к предварительно установленному значению по положительным фронтам сигнала на входе CU (Count Up). Когда текущее значение (CV) больше или равно предварительно установленному значению (PV), выходной бит счетчика (Q) включается. Счетчик сбрасывается, когда включается вход сброса (R).

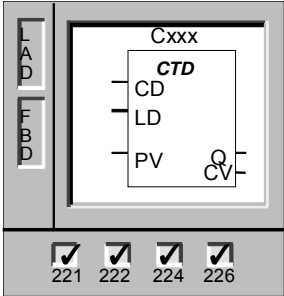
Суммирующий счетчик прекращает счет, когда он достигает предварительно установленного значения (PV).

#### Примечание

Так как для каждого счетчика имеется одно текущее значение, не назначайте один и тот же номер более чем одному счетчику. (Суммирующие, вычитающие и реверсивные счетчики обращаются к одному и тому же текущему значению.)

Входы/выходы	Операнды	Типы данных
CU, R (только LAD)	Поток сигнала	BOOL
CU, R (только FBD)	I, Q, M, SM, V, S, L, T, C, поток сигнала	BOOL
PV (LAD & FBD)	VW, IW, QW, MW, SMW, LW, SW, AIW, AC, константа, *VD, *AC, *LD	INT
Q (LAD & FBD)	I, Q, M, SM, V, S, L	BOOL
CV (LAD & FBD)	VW, IW, QW, MW, SW, LW, AC, *VD, *AC, *LD	INT
Cxxx (LAD & FBD)	Константа	CTU

Вычитающий счетчик



Функциональный блок **Вычитающий счетчик** ведет счет вниз от предварительно установленного значения (PV) по положительным фронтам сигнала на входе CD (**C**ount **D**own). Когда текущее значение (CV) равно нулю, выходной бит счетчика (Q) включается. Счетчик сбрасывается и загружает в качестве текущего значения (CV) предварительно установленное значение (PV), когда включается вход загрузки (LD). Вычитающий счетчик прекращает счет, когда он достигает нуля.

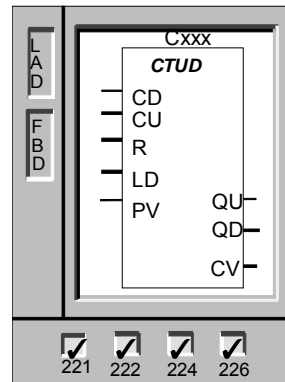
Примечание

Так как для каждого счетчика имеется одно текущее значение, не назначайте один и тот же номер более чем одному счетчику. (Суммирующие, вычитающие и реверсивные счетчики обращаются к одному и тому же текущему значению.)

Входы/выходы	Операнды	Типы данных
CD, LD (LAD)	Поток сигнала	BOOL
CD, LD (FBD)	I, Q, M, SM, V, S, L, T, C, поток сигнала	BOOL
PV (LAD, FBD)	VW, IW, QW, MW, SMW, LW, SW, AIW, AC, константа, *VD, *AC, *LD	INT
Q (LAD & FBD)	I, Q, M, SM, V, S, L	BOOL
CV (LAD & FBD)	VW, IW, QW, MW, SW, LW, AC, *VD, *AC, *LD	INT
Cxxx	Константа	CTD



### Реверсивный счетчик



Функциональный блок **Реверсивный счетчик** ведет счет вверх или вниз от предварительно установленного значения по положительным фронтам сигнала на входе CU (**C**ount **U**p) или CD (**C**ount **D**own). Когда текущее значение (CV) равно предварительно установленному значению, включается выход (QU). Когда текущее значение (CV) равно нулю, включается выход (QD).

Счетчик загружает в качестве текущего значения (CV) предварительно установленное значение (PV), когда включается вход загрузки (LD).

Аналогично, счетчик сбрасывается и загружает в качестве текущего значения (CV) нуль, когда включается сброс (R). Счетчик прекращает счет, когда он достигает предварительно установленного значения или нуля.

#### Примечание

Так как для каждого счетчика имеется одно текущее значение, не назначайте один и тот же номер более чем одному счетчику.  
(Суммирующие, вычитающие и реверсивные счетчики обращаются к одному и тому же текущему значению.)

Входы/выходы	Операнды	Типы данных
CU, CD, R, LD (только LAD)	Поток сигнала	BOOL
CU, CD, R, LD (только FBD)	I, Q, M, SM, V, S, L, T, C, поток сигнала	BOOL
PV (LAD & FBD)	VW, IW, QW, MW, SMW, LW, SW, AIW, AC, константа, *VD, *AC, *LD	INT
QU (LAD & FBD)	I, Q, M, SM, V, S, L	BOOL
QD (LAD & FBD)	I, Q, M, SM, V, S, L	BOOL
CV (LAD & FBD)	VW, IW, QW, MW, SW, LW, AC, *VD, *AC, *LD	INT
Cxxx	Константа	CTUD

Пример счетчика

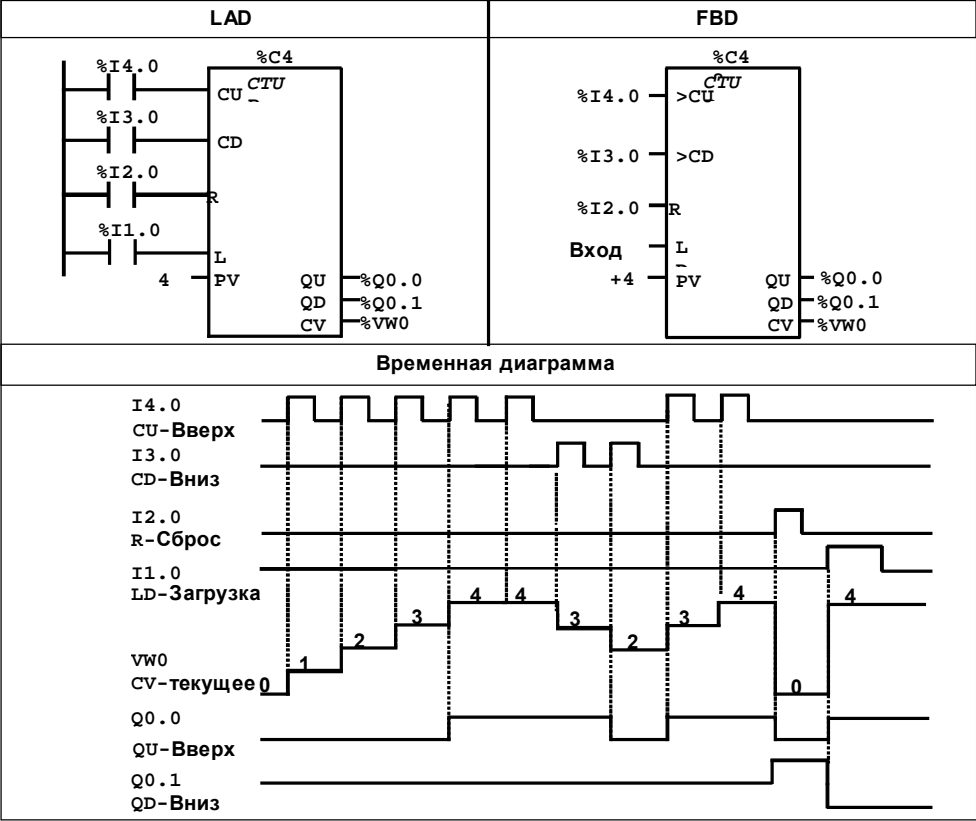
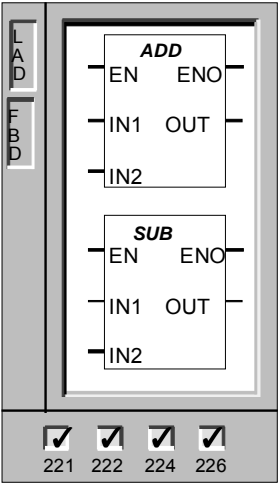


Рис. 10–6. Пример команды счетчика для LAD и FBD

# 10.5 Арифметические команды IEC

## Сложение, вычитание



Функции **Сложение (ADD)** и **Вычитание (SUB)** складывают или вычисляют разность IN1 и IN2 и помещают результат в OUT. Типы входных и выходных данных могут изменяться, но должны быть одинаковыми. Например, две 16-разрядные переменные могут складываться или вычитаться, но результат должен помещаться в 16-разрядную переменную; результат сложения или вычитания двух 32-разрядных переменных должен помещаться в 32-разрядную переменную.

В форме LAD:

IN1 + IN2 = OUT

IN1 - IN2 = OUT

Ошибки, устанавливающие ENO в 0: SM1.1 (переполнение), SM4.3 (этап выполнения), 0006 (косвенная адресация).

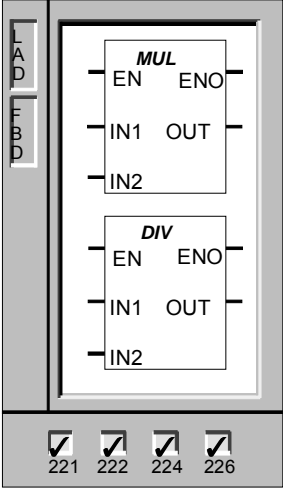
Эти функции влияют на следующие биты специальной памяти: SM1.0 (нуль); SM1.1 (переполнение); SM1.2 (отрицательный результат).

Входы/выходы	Операнды	Типы данных
IN1, IN2	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, VD, ID, QD, MD, SMD, SD, LD, HC, AC, константа, *VD, *AC, *LD	INT, DINT, REAL
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	INT, DINT, REAL

### Примечание

Вещественные числа или числа с плавающей точкой представляются в формате, описанном в стандарте ANSI/IEEE 754-1985 (с одинарной точностью). Для получения дополнительной информации обратитесь к этому стандарту.

Умножение, деление



Функция **Умножение (MUL)** умножает IN1 на IN2 и помещает результат в переменную, заданную через OUT.

Функция **Деление (DIV)** делит IN1 на IN2 и помещает результат в переменную, заданную через OUT.

Типы входных и выходных данных могут изменяться, но должны быть одинаковыми. Например, произведение двух 16-разрядных переменных должно помещаться в 16-разрядную переменную, произведение двух 32-разрядных переменных должно помещаться в 32-разрядную переменную.

В форме LAD:  $IN1 * IN2 = OUT$   
 $IN1 / IN2 = OUT$

Ошибки, устанавливающие ENO в 0: SM1.1 (переполнение), SM1.3 (деление на ноль), 0006 (косвенная адресация).

Эти функции влияют на следующие биты специальной памяти: SM1.0 (ноль); SM1.1 (переполнение); SM1.2 (отрицательный результат); SM1.3 (деление на ноль).

Если устанавливается SM1.1 (бит переполнения), то другие математические биты состояния сбрасываются и выходной операнд не изменяется. В случае целочисленных операций, если во время операции деления устанавливается SM1.3, то другие математические биты состояния остаются неизменными и первоначальные входные операнды не изменяются. В ином случае все поддерживаемые математические биты состояния содержат действительное состояние на момент завершения математической операции.

Входы/выходы	Операнды	Типы данных
IN1, IN2	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, VD, ID, QD, MD, SMD, SD, LD, HC, AC, константа, *VD, *AC, *LD	INT, DINT, REAL
OUT	VW, IW, QW, MW, SW, SMW, T, C, LW, VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	INT, DINT, REAL

Примечание

Вещественные числа или числа с плавающей точкой представляются в формате, описанном в стандарте ANSI/IEEE 754-1985 (с одинарной точностью). Для получения дополнительной информации обратитесь к этому стандарту.

Математические примеры

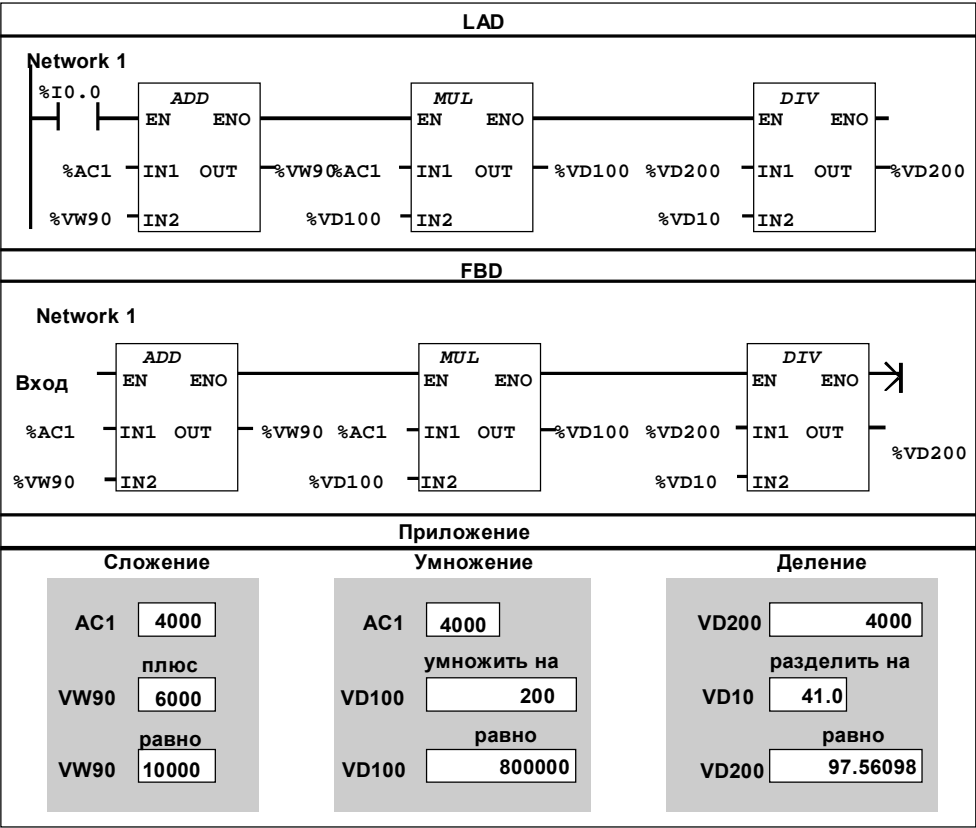
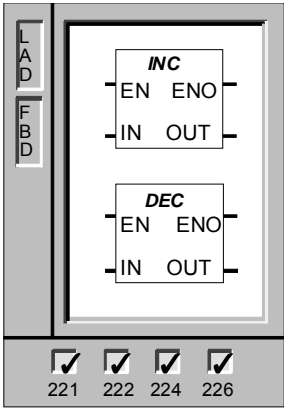


Рис. 10–7. Примеры математических функций для LAD и FBD

Инкремент, декремент



Функции **Инкремент** и **Декремент** увеличивают или уменьшают IN на 1 и помещают результат в OUT.  
Операции инкрементирования и декрементирования применяются к байтам без знака.  
Ошибки, устанавливающие ENO в 0: SM1.1 (переполнение), SM4.3 (этап выполнения), 0006 (косвенная адресация).  
Эти функции влияют на следующие биты специальной памяти: SM1.0 (нуль); SM1.1 (переполнение); SM1.2 (отрицательный результат).

Входы/выходы	Операнды	Типы данных
IN	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, VD, ID, QD, MD, SD, SMD, LD, HC, AC, константа, *VD, *AC, *LD	BYTE, INT, DINT
OUT	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, T, C, LW, VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	BYTE, INT, DINT

Пример инкрементирования и декрементирования

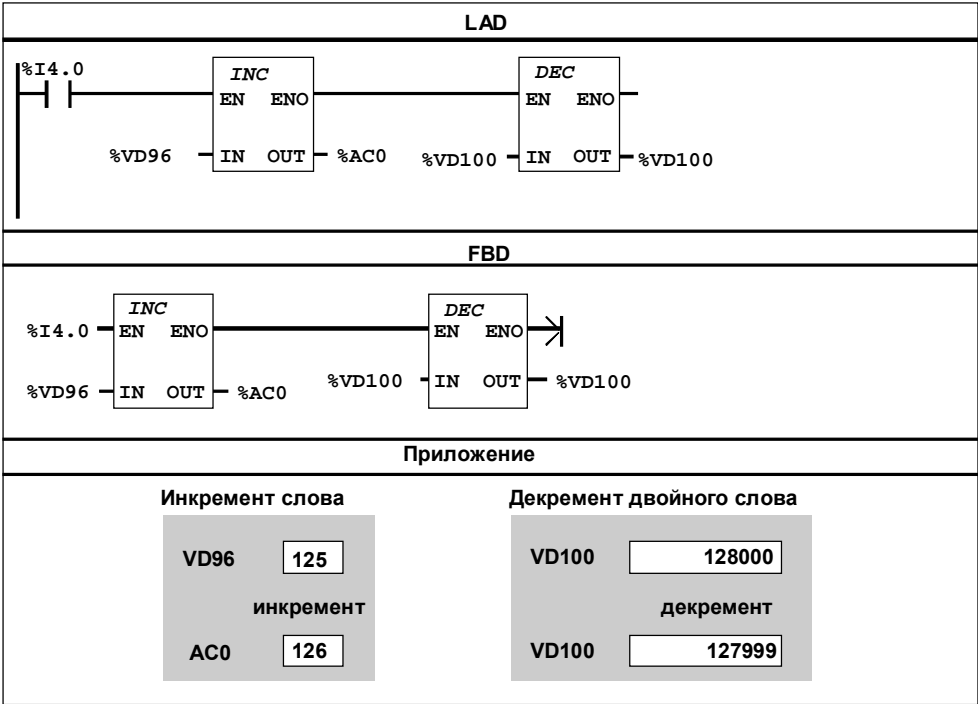


Рис. 10–8. Пример функций инкрементирования и декрементирования для LAD и FBD

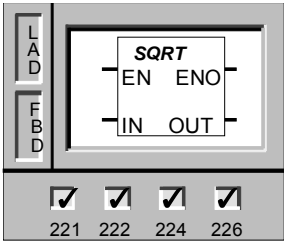
### 10.6 Числовые функции IEC

Таблица 10–5 дает постраничные ссылки для нестандартных числовых функций IEC.

Таблица 10–5. Нестандартные числовые функции IEC

Описание	Страница
PID-регулятор	9–88

#### Квадратный корень



Команда **Квадратный корень** извлекает квадратный корень из значения, поступающего на IN, и помещает результат в OUT.

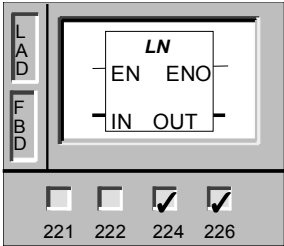
Ошибки, устанавливающие ENO в 0: SM1.1 (переполнение), SM4.3 (этап выполнения), 0006 (косвенная адресация).

Эта функция влияет на следующие биты специальной памяти: SM1.0 (нуль); SM1.1 (переполнение); SM1.2 (отрицательный результат).

Если устанавливается SM1.1 (бит переполнения), то другие математические биты состояния сбрасываются и выходной операнд не изменяется.

Входы/выходы	Операнды	Типы данных
IN	VD, ID, QD, MD, SMD, SD, LD, AC, константа, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	REAL

#### Натуральный логарифм



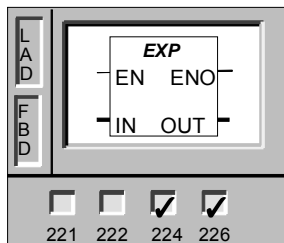
Команда **Натуральный логарифм** вычисляет натуральный логарифм значения в IN и помещает результат в OUT.

Ошибки, устанавливающие ENO в 0: SM1.1 (переполнение), 0006 (косвенная адресация).

Эта команда влияет на следующие биты специальной памяти: SM1.0 (нуль); SM1.1 (переполнение); SM1.2 (отрицательный результат).

Входы/выходы	Операнды	Типы данных
IN	VD, ID, QD, MD, SMD, SD, LD, AC, константа, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	REAL

## Натуральная экспонента



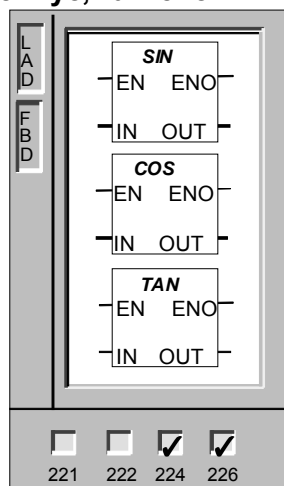
Команда **Натуральная экспонента** выполняет экспоненциальную операцию возведения числа  $e$  в степень, равную значению в IN, и помещает результат в OUT.

Ошибки, устанавливающие ENO в 0: SM1.1 (переполнение), 0006 (косвенная адресация).

Эта команда влияет на следующие биты специальной памяти: SM1.0 (нуль); SM1.1 (переполнение); SM1.2 (отрицательный результат).

Входы/выходы	Операнды	Типы данных
IN	VD, ID, QD, MD, SMD, SD, LD, AC, константа, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	REAL

## Синус, косинус, тангенс



Команда **Синус** вычисляет синус угла IN и помещает результат в OUT. Входной угол выражен в радианах. Если угол выражен в градусах, то его нужно умножить на  $\pi/180$ , чтобы преобразовать в радианы.

Команда **Косинус** вычисляет косинус угла IN и помещает результат в OUT. Входной угол выражен в радианах. Если угол выражен в градусах, то его нужно умножить на  $\pi/180$ , чтобы преобразовать в радианы.

Команда **Тангенс** вычисляет тангенс угла IN и помещает результат в OUT. Входной угол выражен в радианах. Если угол выражен в градусах, то его нужно умножить на  $\pi/180$ , чтобы преобразовать в радианы.

Ошибки, устанавливающие ENO в 0: SM1.1 (переполнение), 0006 (косвенная адресация).

Эти команды влияют на следующие биты специальной памяти: SM1.0 (нуль); SM1.1 (переполнение); SM1.2 (отрицательный результат).

Входы/выходы	Операнды	Типы данных
IN	VD, ID, QD, MD, SMD, SD, LD, AC, константа, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	REAL



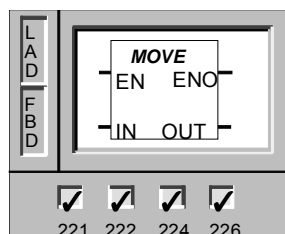
## 10.7 Команды пересылки IEC

Таблица 10–6 дает постраничные ссылки для нестандартных команд пересылки IEC.

Таблица 10–6. Нестандартные команды пересылки IEC

Описание	Страница
Команды перестановки	9–108
Пересылка непосредственно считанного байта	9–109
Пересылка байта для непосредственной записи	9–109

### Пересылка



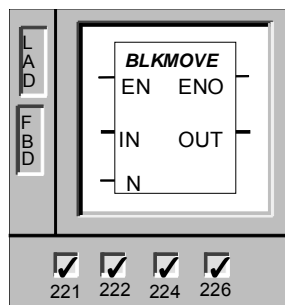
Функция **Пересылка и присваивание значений** пересылает значение IN по адресу OUT. Эта команда выполняет операцию присваивания. Входной параметр во время выполнения не изменяется.

Типы входных и выходных данных могут изменяться, но должны быть одинаковыми.

Ошибки, устанавливающие ENO в 0: SM4.3 (этап выполнения), 0006 (косвенная адресация).

Входы/выходы	Операнды	Типы данных
IN	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SM, SMW, LW, T, C, AIW, VD, ID, QD, MD, SMD, SD, LD, HC, &VB, &IB, &QB, &MB, &SB, AC, константа, *VD, *AC, *LD	BYTE, WORD, INT, DWORD, DINT, REAL
OUT	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, LW, T, C, AQW, VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	BYTE, WORD, INT, DWORD, DINT, REAL

### Групповая пересылка



Функция **Групповая пересылка** пересылает N слов, заданных посредством адреса IN, по адресу OUT. N имеет диапазон значений от 1 до 255.

Типы входных и выходных данных могут изменяться, но должны быть одинаковыми.

Групповая пересылка является нестандартной функцией только в IEC.

Ошибки, устанавливающие ENO в 0: SM4.3 (этап выполнения), 0006 (косвенная адресация), 0091 (операнд вне диапазона).

Входы/выходы	Операнды	Типы данных
IN	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SM, SMW, LW, T, C, AIW, VD, ID, QD, MD, SMD, SD, LD, *VD, *AC, *LD	BYTE, WORD, DWORD, INT, DINT
OUT	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, LW, T, C, AQW, VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	BYTE, WORD, DWORD, INT, DINT
N	VB, IB, QB, MB, SB, SMB, LB, AC, константа, *VD, *AC, *LD	BYTE

Примеры пересылки

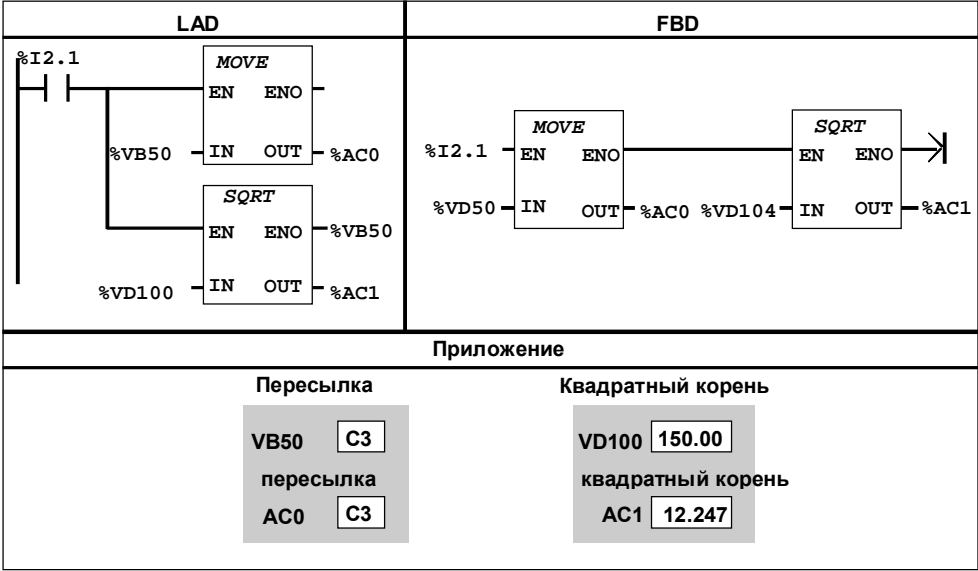
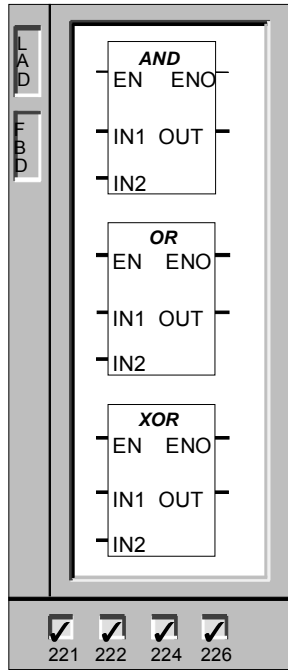


Рис. 10–9. Пример пересылки для LAD и FBD

10.8 Логические команды IEC

Нестандартные логические команды IEC не существуют.

И, ИЛИ, Исключающее ИЛИ



Функция **И (AND)** выполняет операцию И над соответствующими битами IN1 и IN2 и загружает результат в OUT.

Функция **ИЛИ (OR)** выполняет операцию ИЛИ над соответствующими битами IN1 и IN2 и загружает результат в OUT.

Функция **Исключающее ИЛИ (XOR)** выполняет операцию Исключающее ИЛИ над соответствующими битами IN1 и IN2 и загружает результат в OUT.

Типы входных и выходных данных могут изменяться, но должны быть одинаковыми.

Ошибки, устанавливающие ENO в 0: SM4.3 (этап выполнения), 0006 (косвенная адресация).

Эти команды влияют на следующие биты специальной памяти: SM1.0 (нуль).

Входы/выходы	Операнды	Типы данных
IN1, IN2	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, AIW, T, C, LW, VD, ID, QD, MD, SD, SMD, LD, HC, AC, константа, *VD, *AC, *LD	BYTE, WORD DWORD
OUT	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, T, C, LW, VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	BYTE, WORD DWORD

Пример команд И, ИЛИ, Исключающее ИЛИ

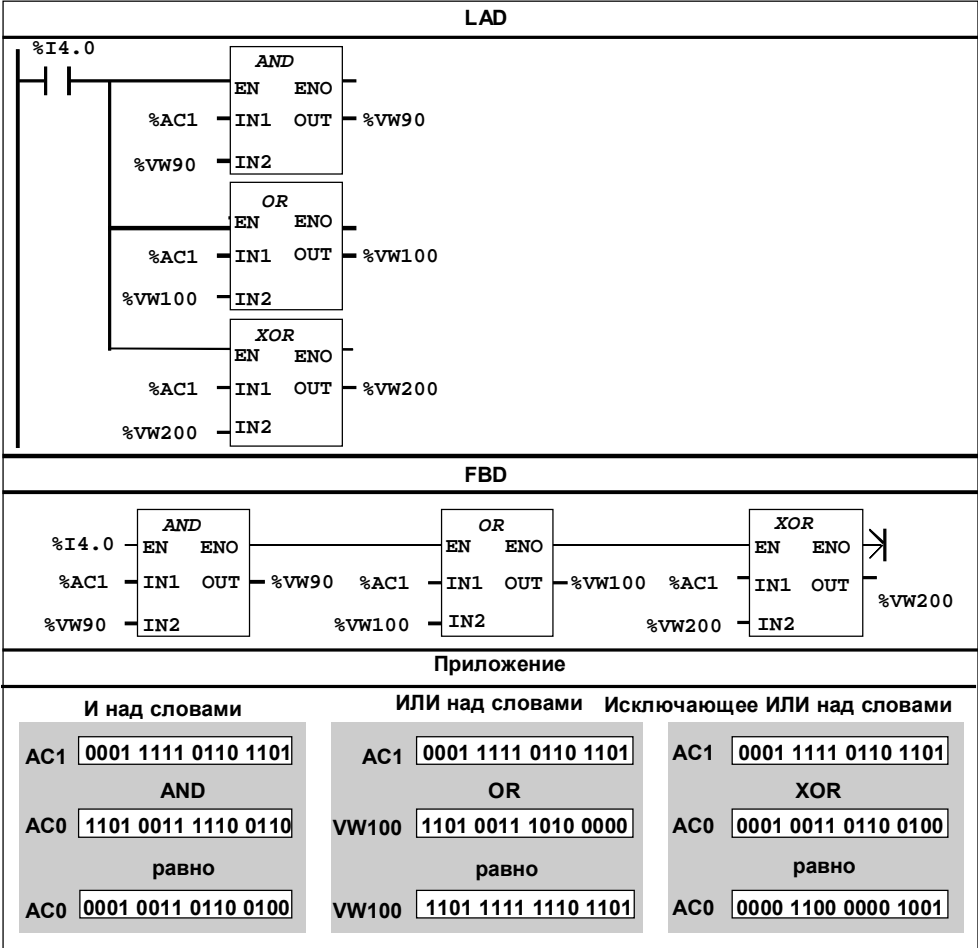
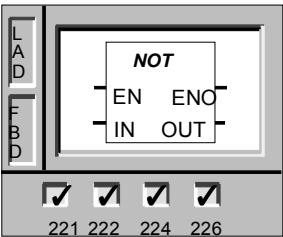


Рис. 10–10. Пример функций И, ИЛИ и Исключающее ИЛИ

NOT [НЕ]



Функция **NOT [НЕ]** инвертирует соответствующие биты IN и загружает результат в OUT.

Типы входных и выходных данных могут изменяться, но должны быть одинаковыми.

Ошибки, устанавливающие ENO в 0: SM4.3 (этап выполнения), 0006 (косвенная адресация).

Эта команда влияет на следующие биты специальной памяти: SM1.0 (нуль).

Входы/выходы	Операнды	Типы данных
IN	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, AIW, T, C, LW, VD, ID, QD, MD, SD, SMD, LD, HC, AC, константа, *VD, *AC, *LD	BYTE, WORD DWORD
OUT	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, T, C, LW, VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	BYTE, WORD DWORD

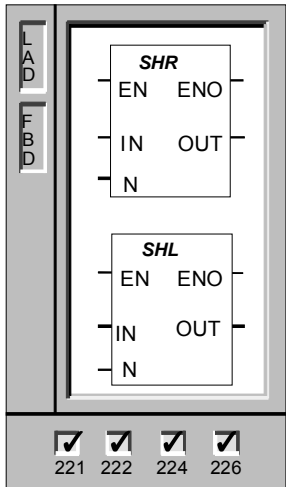
### 10.9 Команды сдвига и циклического сдвига IEC

Таблица 10–7 дает постраничные ссылки для нестандартных команд сдвига IEC.

Таблица 10–7 Нестандартные команды IEC

Описание	Страница
Вдвигание бита в регистр сдвига	9–130

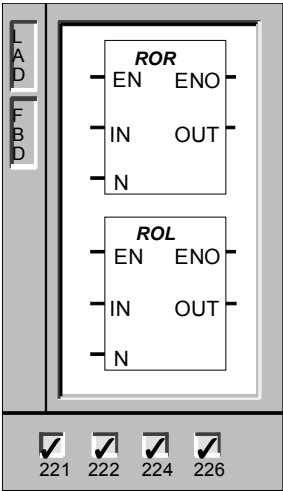
#### Логический сдвиг вправо, логический сдвиг влево



Функция **Сдвиг вправо (SHR)** сдвигает значение, заданное посредством переменной IN, вправо на количество разрядов, заданное через N. Результат помещается в переменную, заданную через OUT. Каждый бит заполняется нулем, когда он сдвигается вправо. Заметьте, что знаковый разряд сдвигается, когда вы используете типы данных, имеющие знак. Функция **Сдвиг влево (SHL)** сдвигает значение, заданное посредством переменной IN, влево на количество разрядов, заданное через N. Результат помещается в переменную, заданную через OUT. Каждый бит заполняется нулем, когда он сдвигается влево. Заметьте, что знаковый разряд сдвигается, когда вы используете типы данных, имеющие знак. Ошибки, устанавливающие ENO в 0: SM4.3 (этап выполнения), 0006 (косвенная адресация).

Входы/выходы	Операнды	Типы данных
IN	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, VD, ID, QD, MD, SD, SMD, LD, HC, AC, константа, *VD, *LD, *AC	BYTE, WORD DWORD
N	VB, IB, QB, MB, SB, SMB, LB, AC, константа, *VD, *LD, *AC	BYTE
OUT	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, LW, T, C, VD, ID, QD, MD, SD, SMD, LD, AC *VD, *LD, *AC	BYTE, WORD DWORD

Логический циклический сдвиг вправо, логический циклический сдвиг влево



Функции Циклический сдвиг вправо (ROR) и Циклический сдвиг влево (ROL) циклически сдвигают входное значение (IN) вправо или влево на заданное число разрядов (N) и загружают результат в выход (OUT).

Циклический сдвиг является круговым. В команде ROR бит с номером нуль циклически сдвигается в старший значащий бит. В команде ROL старший значащий бит циклически сдвигается в бит с номером нуль. Заметьте, что знаковый разряд сдвигается, когда вы используете типы данных, имеющие знак.

Ошибки, устанавливающие ENO в 0: SM4.3 (этап выполнения), 0006 (косвенная адресация).

Входы/выходы	Операнды	Типы данных
IN	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, VD, ID, QD, MD, SD, SMD, LD, HC, AC, константа, *VD, *LD, *AC	BYTE, WORD, DWORD
N	VB, IB, QB, MB, SB, SMB, LB, AC, константа, *VD, *LD, *AC	BYTE
OUT	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, LW, T, C, VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	BYTE, WORD, DWORD

Примеры сдвига и циклического сдвига

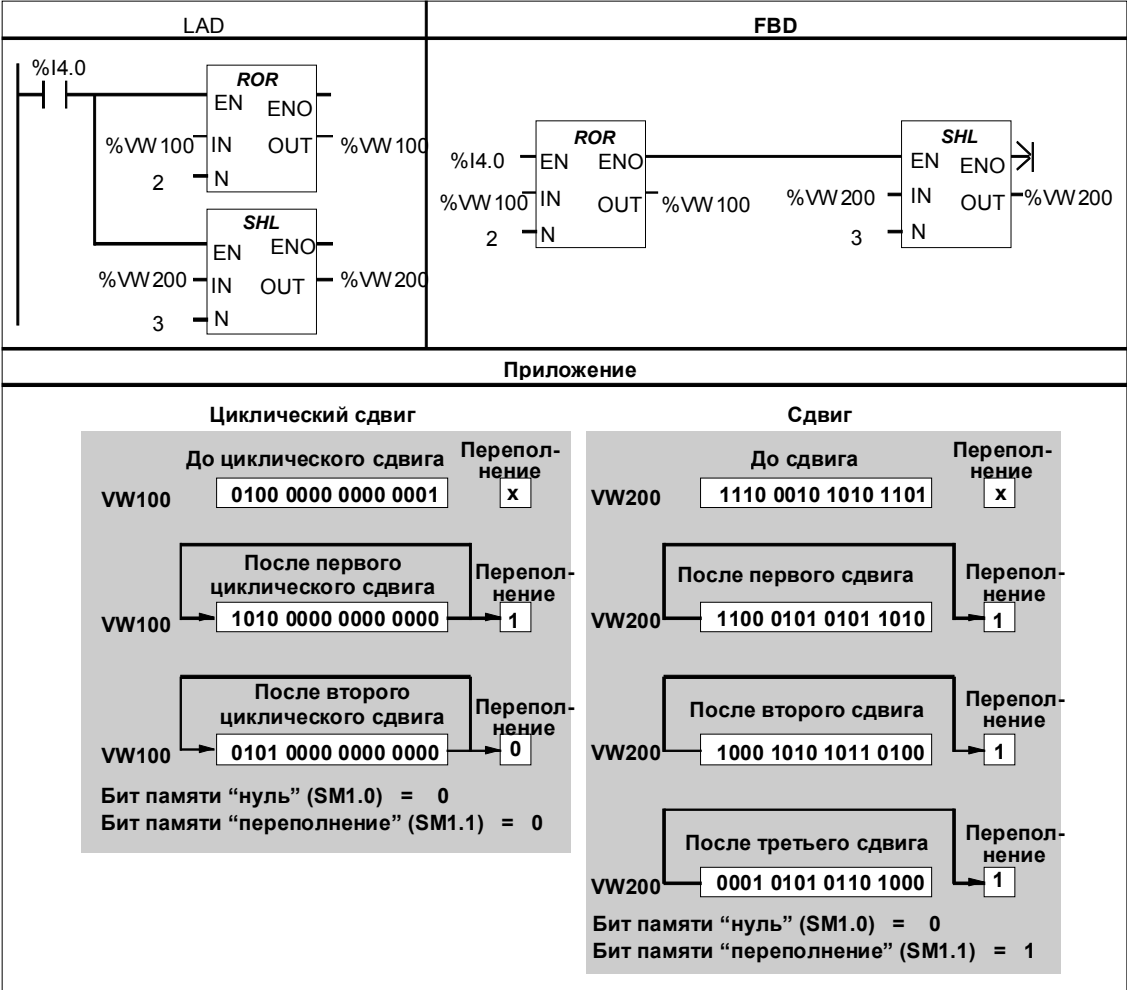


Рис. 10–11. Пример функций сдвига и циклического сдвига для LAD и FBD

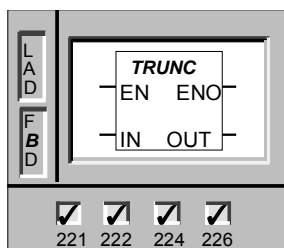
## 10.10 Команды преобразования IEC

Таблица 10–8 дает постраничные ссылки для нестандартных команд преобразования IEC.

Таблица 10–8. Нестандартные команды преобразования IEC

Описание	Страница
Декодирование	9–138
Кодирование	9–138
Управление 7-сегментным дисплеем (Сегмент)	9–140
Преобразование ASCII в 16-ричный код и 16-ричного кода в ASCII	9–142
Преобразование целого числа в строку ASCII	9–143
Преобразование двойного целого числа в строку ASCII	9–145
Преобразование вещественного числа в строку ASCII	9–146

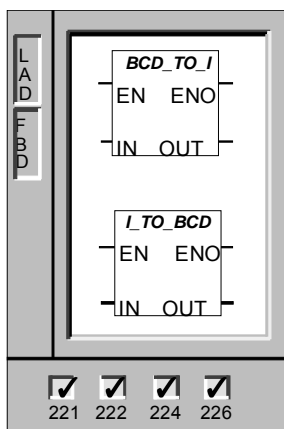
### Выделение целой части



Функция **Выделение целой части** преобразует вещественное число (IN) в значение двойного целого числа и помещает результат в OUT. Округление не выполняется. Ошибки, устанавливающие ENO в 0: SM1.1 (переполнение), SM4.3 (этап выполнения), 0006 (косвенная адресация). Эта функция влияет на следующие биты специальной памяти: SM1.1 (переполнение).

Входы/выходы	Операнды	Типы данных
IN	VD, ID, QD, MD, SD, SMD, LD, AC, константа, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	DINT

### Двоично-десятичный код (BCD) в целое число, целое число в BCD

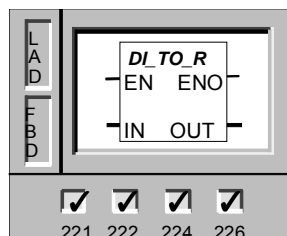


Функция **BCD в целое число** преобразует входное значение двоично-десятичного кода (IN) в целочисленное значение и загружает результат в переменную, заданную через OUT. Функция **Целое число в BCD** преобразует входное целочисленное значение в значение двоично-десятичного кода и загружает результат в OUT. Ошибки, устанавливающие ENO в 0: SM1.6 (BCD), SM4.3 (этап выполнения), 0006 (косвенная адресация). Эти функции влияют на следующие биты специальной памяти: SM1.6 (недействительный BCD).

Входы/выходы	Операнды	Типы данных
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, константа, *VD, *LD, *AC	WORD
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD	WORD



### Двойное целое число в вещественное число

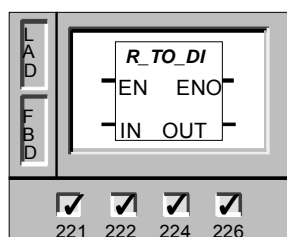


Функция **Двойное целое число в вещественное число** преобразует 32-разрядное целое число со знаком (IN) в 32-разрядное вещественное число и загружает результат в переменную, заданную через OUT.

Ошибки, устанавливающие ENO в 0: SM4.3 (этап выполнения), 0006 (косвенная адресация).

Входы/выходы	Операнды	Типы данных
IN	VD, ID, QD, MD, SD, SMD, LD, HC, AC, константа *VD, *LD, *AC	DINT
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	REAL

### Вещественное число в двойное целое число

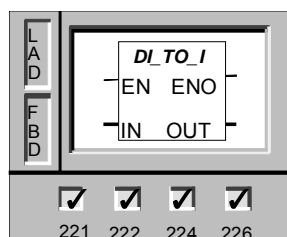


Функция **Вещественное число в двойное целое число** преобразует значение вещественного числа (IN) в значение двойного целого числа и загружает результат в переменную, заданную через OUT.

Ошибки, устанавливающие ENO в 0: SM1.1 (переполнение), SM4.3 (этап выполнения), 0006 (косвенная адресация).

Входы/выходы	Операнды	Типы данных
IN	VD, ID, QD, MD, SD, SMD, LD, AC, константа, *VD, *LD, *AC	REAL
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	DINT

### Двойное целое число в целое число



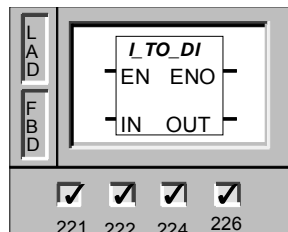
Функция **Двойное целое число в целое число** преобразует значение двойного целого числа (IN) в значение целого числа и загружает результат в переменную, заданную через OUT.

Ошибки, устанавливающие ENO в 0: SM1.1 (переполнение), SM4.3 (этап выполнения), 0006 (косвенная адресация).

Эта функция влияет на следующие биты специальной памяти: SM1.1 (переполнение).

Входы/выходы	Операнды	Типы данных
IN	VD, ID, QD, MD, SD, SMD, LD, HC, AC, константа, *VD, *LD, *AC	DINT
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC	INT

## Целое число в двойное целое число

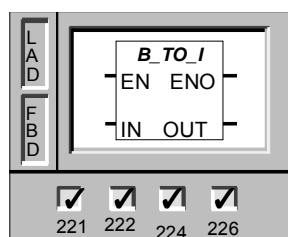


Функция **Целое число в двойное целое число** преобразует значение целого числа, заданное через (IN), в значение двойного целого числа и загружает результат в переменную, заданную через OUT.

Ошибки, устанавливающие ENO в 0: SM4.3 (этап выполнения), 0006 (косвенная адресация).

Входы/выходы	Операнды	Типы данных
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, константа, AC, *VD, *LD, *AC	INT
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	DINT

## Байт в целое число

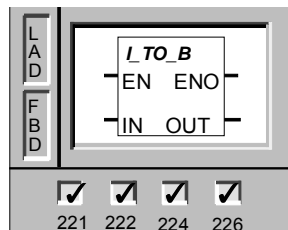


Функция **Байт в целое число** преобразует значение байта (IN) в значение целого числа и загружает результат в переменную, заданную через OUT.

Ошибки, устанавливающие ENO в 0: SM4.3 (этап выполнения), 0006 (косвенная адресация).

Входы/выходы	Операнды	Типы данных
IN	VB, IB, QB, MB, SB, SMB, LB, AC, константа, *VD, *LD, *AC	BYTE
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC	INT

## Целое число в байт



Функция **Целое число в байт** преобразует значение целого числа (IN) в значение байта и загружает результат в переменную, заданную через OUT.

Ошибки, устанавливающие ENO в 0: SM1.1 (переполнение), SM4.3 (этап выполнения), 0006 (косвенная адресация).

Эта функция влияет на следующие биты специальной памяти: SM1.1 (переполнение).

Входы/выходы	Операнды	Типы данных
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, константа, *VD, *LD, *AC	INT
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC	BYTE

Пример преобразования

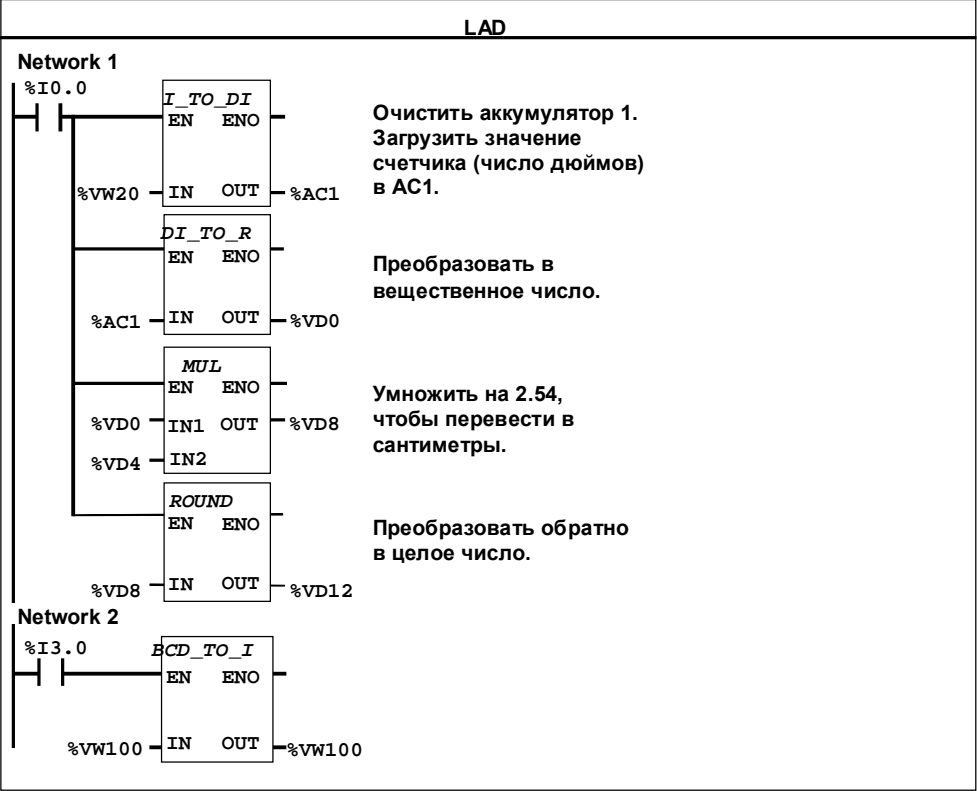


Рис. 10–12. Пример команды преобразования вещественного числа для LAD

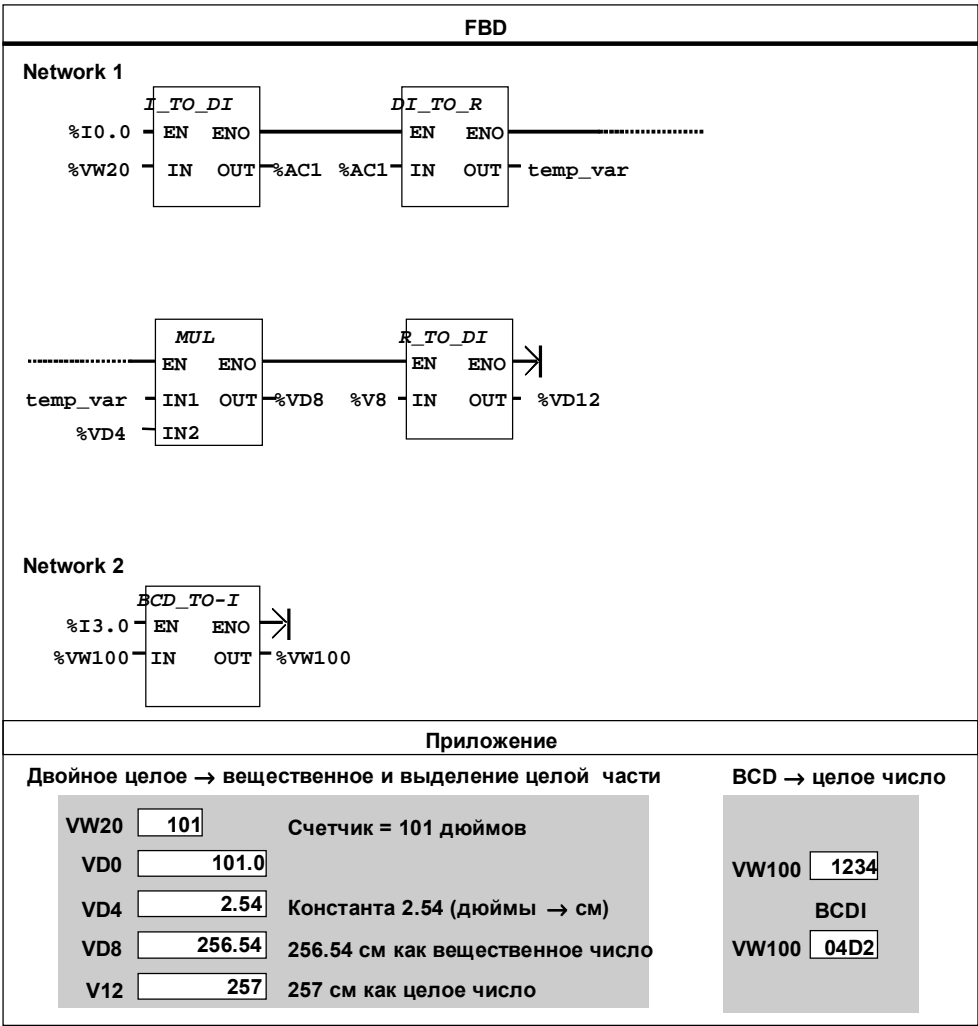


Рис. 10–13. Пример команды преобразования вещественного числа для FBD