

Програмне забезпечення мікропроцесорних систем

Лекція 6

Основные принципы применения языка
LAD. Таймеры и счетчики
(на примере пакета CoDeSys)

Вопросы лекции

- Бинарная логика (на примере LAD)
- Библиотеки в CoDeSys (Standard)
 - Таймеры
 - Счетчики

Редактор LAD

- графический язык реализует структуры эл. цепей
 - подходит для построения логических переключателей
 - легко можно создавать сложные цепи управления
 - LD удобен для управления РОУ
- программа представлена в виде списка цепей
 - каждая цепь состоит из двух частей
 - в левой находится условие ЛО (контакты)
 - в правой - структура, состоящая из логических или арифметических операций, вызовов программ, функций или ФБ, инструкций перехода или возврата (обмотки)

Бинарная логика...

Язык LAD (ladder diagram)

- релейная схема – две верт. шины, соединенные с помощью контактов и обмоток
- компоновка контактов в *последовательные* и *параллельные* схемы **определяет** комбинирование бинарных сигнальных состояний датчиков

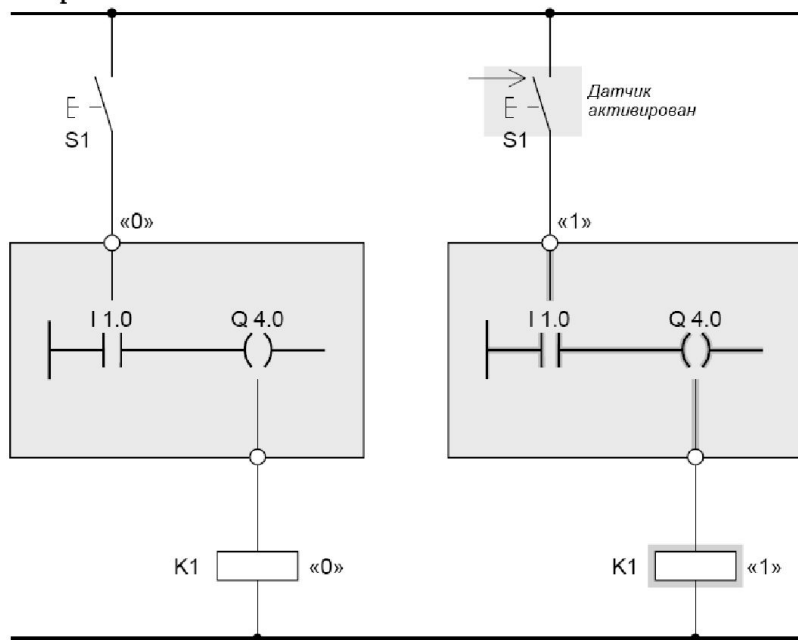
Бинарная логика...

Логическая переменная типа «Контакт»

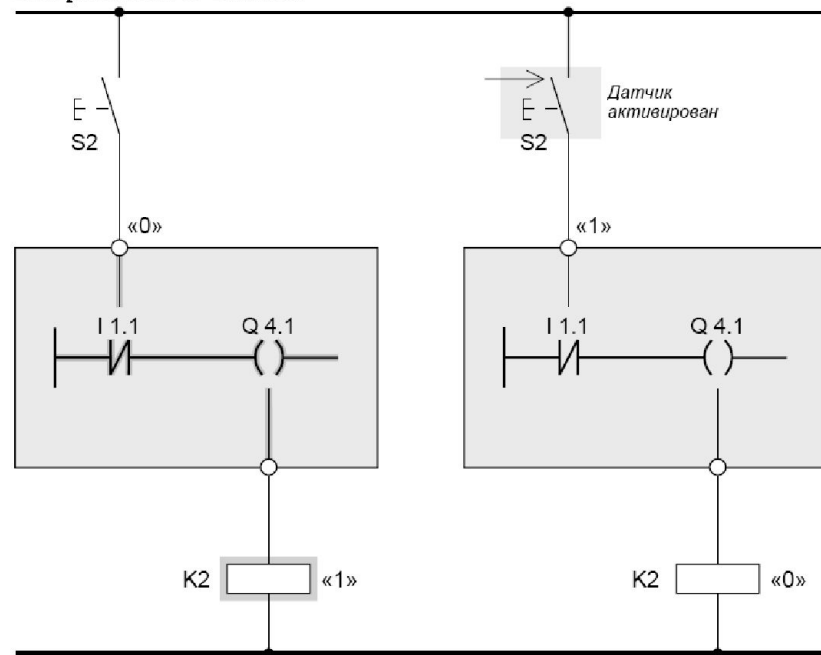
- проверяет сигнальное состояние двоичных операндов
 - входные и выходные биты
 - память меркеров
 - таймеры и счетчики
 - биты глобальных данных
 - биты временных локальных данных
 - биты слова состояния (оценка результатов вычислений)

Принцип работы РКС

Как работает NO-контакт



Как работает NC-контакт



Бинарная логика...

- NO-контакт соответствует сканированию с ожиданием сигнального состояния «1»
- NC-контакт проверяет вход на наличие сигнального состояния «0»

Нормально разомкнутый контакт
(Normally-open, NO)

Нормально замкнутый контакт
(Normally-closed, NC)

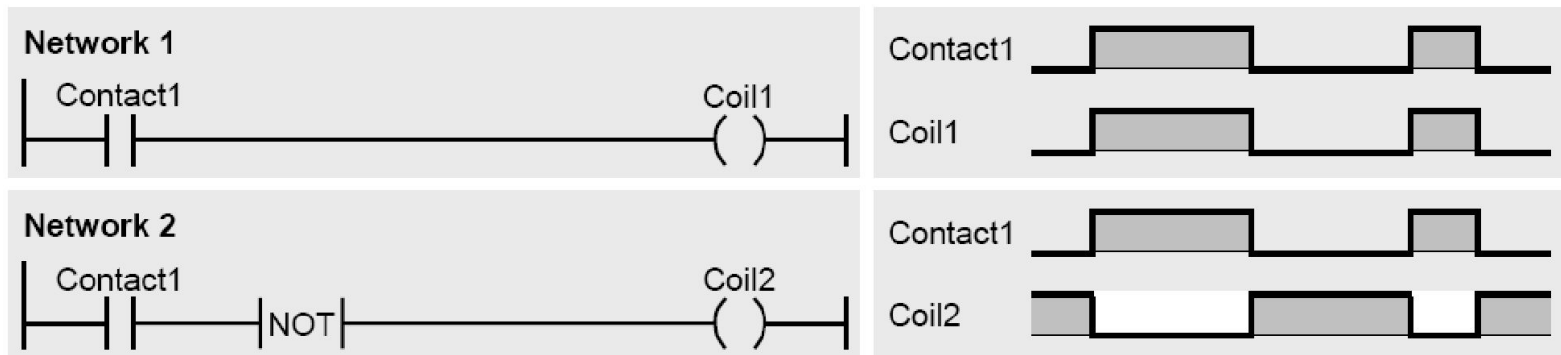
Бинарный операнд



Бинарный операнд

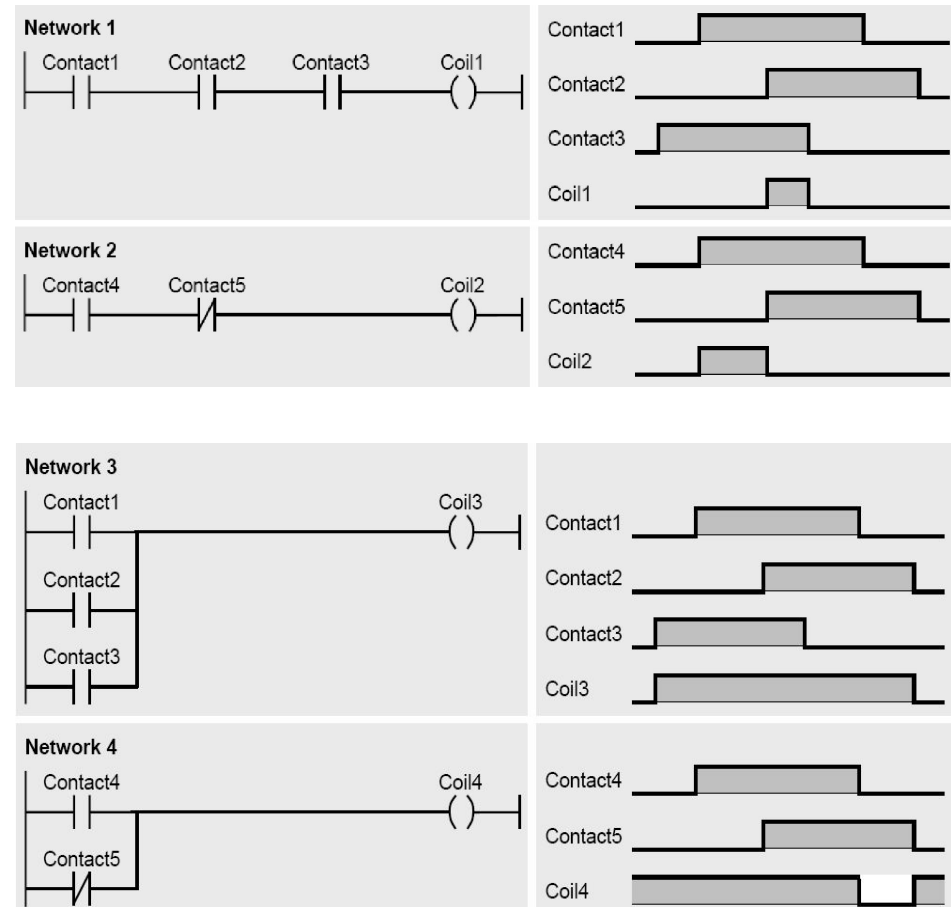


Одиночная катушка (single coil)



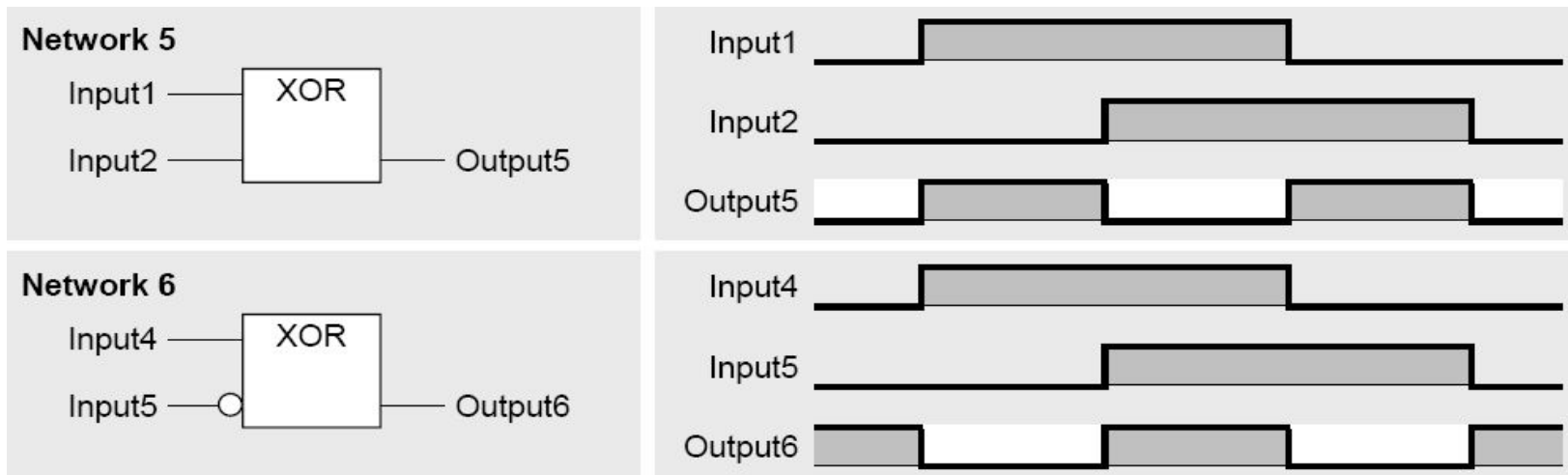
Бинарная логика...

- В последовательных схемах ток течет, если все контакты замкнуты
- В параллельных схемах ток течет, если хотя бы один из контактов замкнут



Бинарная логика...

- Функция Exclusive OR комбинирует друг с другом два бинарных состояния
 - возвращает RLO «1», когда два состояния (результаты сканирования) не являются одинаковыми
 - возвращает RLO «0», если два состояния (результаты сканирования) идентичны



Выходной элемент

- —() —

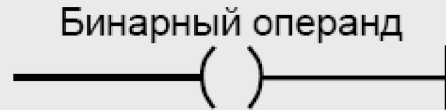
- выходная катушка, которая является терминатором (завершающим элементом) цепи и присваивает (assigns) RLO (результат логической операции) напрямую операнду

Функции для работы с памятью...

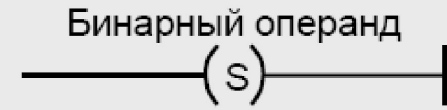
- одиночная катушка (coil) как присваивание RLO
- катушки R и S как индивидуально программируемые операции с памятью
- блочные элементы (boxes) RS и SR как функции, работающие с памятью
- коннекторы (midline outputs) как промежуточные буферы
- элементы оценки (обнаружения) фронта импульса электрического тока

Функции для работы с памятью...

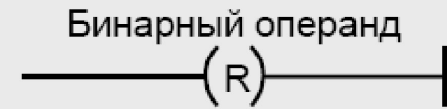
Одиночная катушка



Катушка установки



Катушка сброса



Одиночная катушка

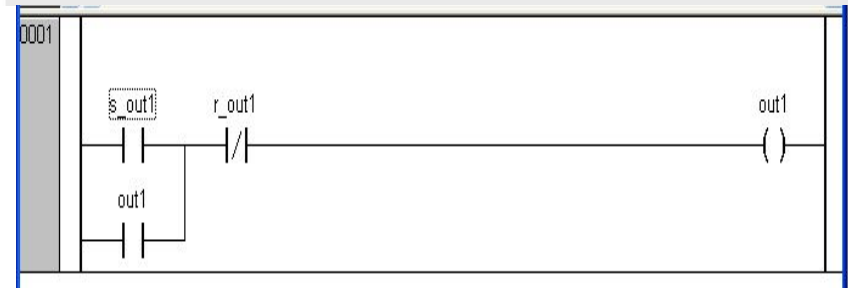
- терминатор цепи направляет (assigns) электрический ток

напрямую к операнду, расположенному при катушке

- параллельно можно составить до 16 катушек

Катушки установки и сброса (set coil, reset coil)

- становятся активными, только когда через них протекает ток



Функции для работы с памятью...

- Блочный элемент памяти (триггер)
 - объединяет функции катушек установки и сброса в блочном элементе функции для работы с общим операндом
 - RS (приоритет сброса)
 - SR (приоритет установки)



Функции для работы с памятью...

- Переключатель с доминантой **включения:**

$$Q1 = SR (\text{SET1}, \text{RESET})$$

означает:

$$Q1 = (\text{NOT RESET} \text{ AND } Q1) \text{ OR SET1}$$

- Переключатель с доминантой **выключения:**

$$Q1 = RS (\text{SET}, \text{RESET1})$$

означает:

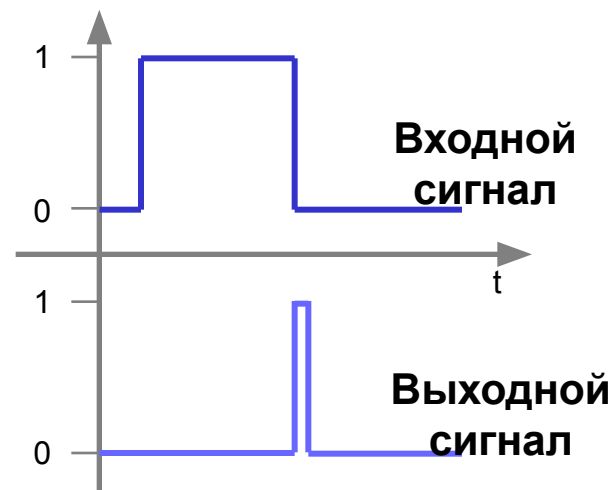
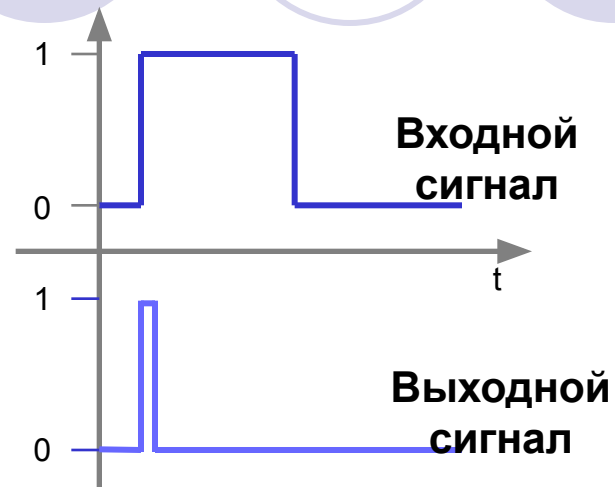
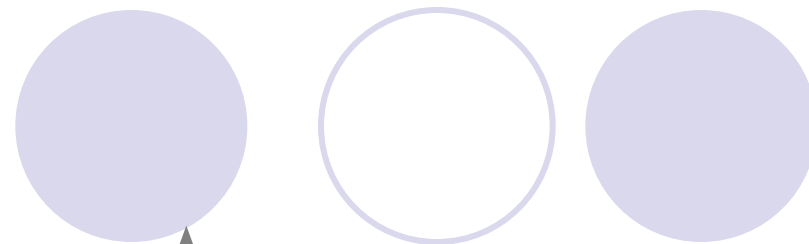
$$Q1 = \text{NOT RESET1} \text{ AND } (Q1 \text{ OR SET})$$

Функции оценки фронта импульса

- функция оценки фронта (edge evaluation)
 - обнаруживает изменение сигнального состояния, фронт сигнала
 - фронт является положительным, если сигнал меняется с «0» на «1»
 - ФБ R_TRIG, который служит для выделения переднего фронта импульса (FALSE -> TRUE) сигнала
 - фронт является отрицательным, если сигнал меняется с «1» на «0»
 - ФБ F_TRIG, который служит для выделения заднего фронта импульса (TRUE -> FALSE) сигнала
 - реализует импульсный контактный элемент
 - CPU сравнивает текущий RLO с сохраненным RLO

Детекторы фронтов

- **R_TRIG**
определяет
передний фронт
- **F_TRIG**
определяет задний фронт



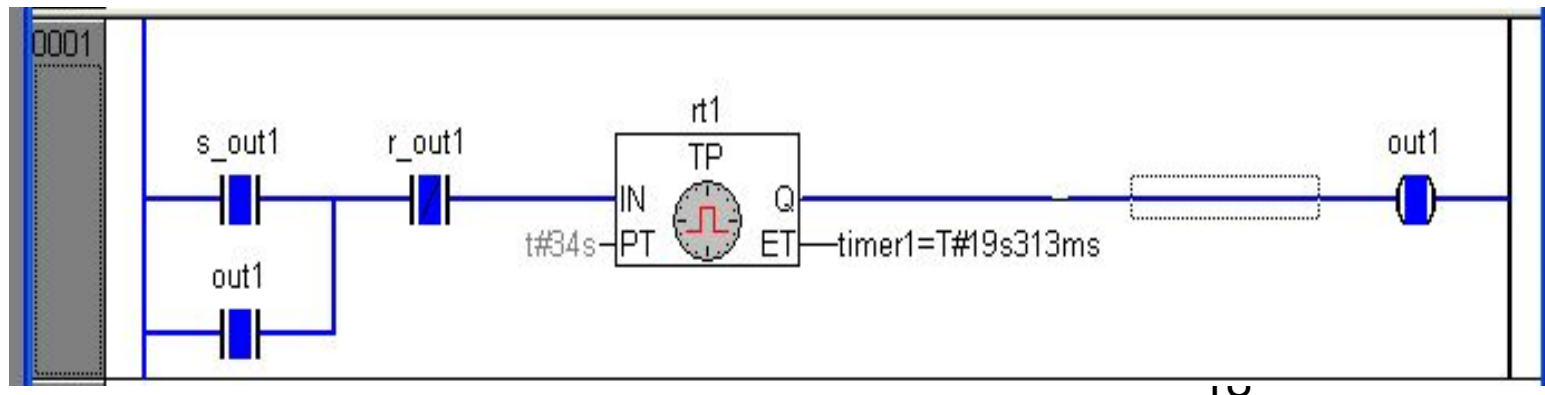
Таймеры...



- позволяют программно реализовать последовательности синхронизации, такие как интервалы ожидания и наблюдения, измерение интервалов или генерирование импульсов
 - имеют область, зарезервированную для них в памяти CPU
 - область памяти резервирует одно 16-битное слово для каждого таймерного адреса
 - виды таймеров
 - ФБ TP - импульсный таймер (pulse timer)
 - ФБ TON - таймер задержки включения (on-delay timer)
 - ФБ TOF - таймер задержки выключения (off-delay timer)

Программирование таймеров...

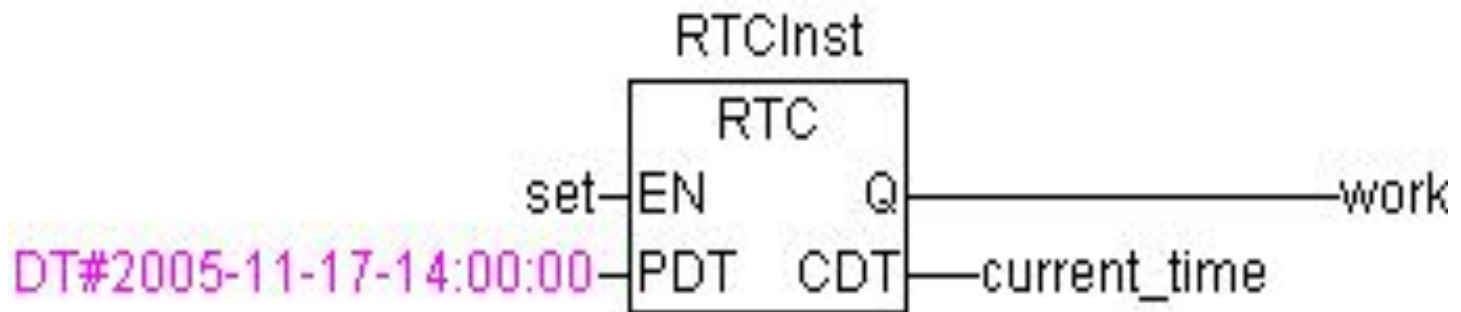
- операции над таймером
 - запуск таймера с заданием значения времени
 - обнуление (сброс) таймера
 - проверка состояния таймера (bool)
 - чтение значения таймера в цифровом виде



Часы реального времени RTC

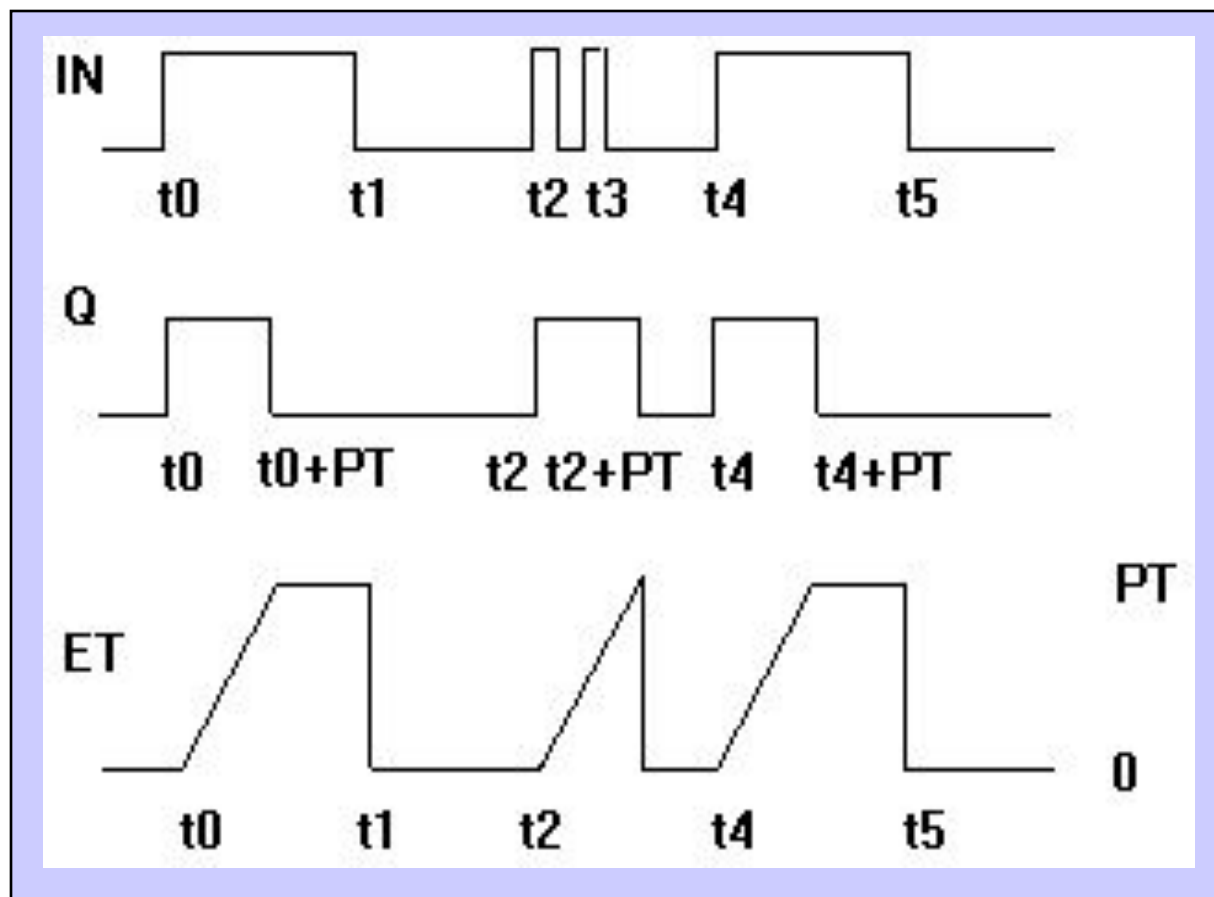
Функциональный блок RTC рассчитывает текущее время и дату, начиная от заданной точки PDT

- при этом ПЛК должен работать постоянно



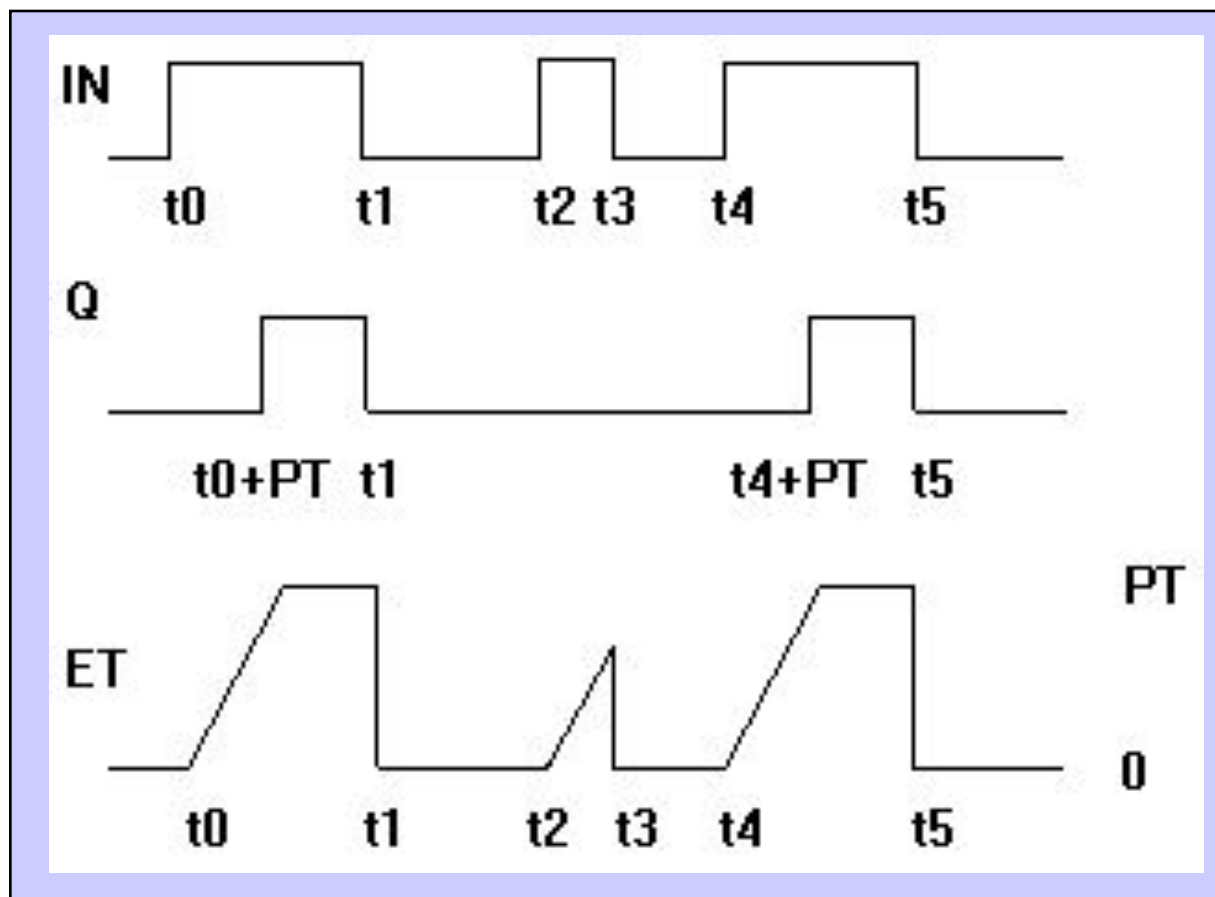
Таймер **TR**

Генерирует импульс заданной длительности



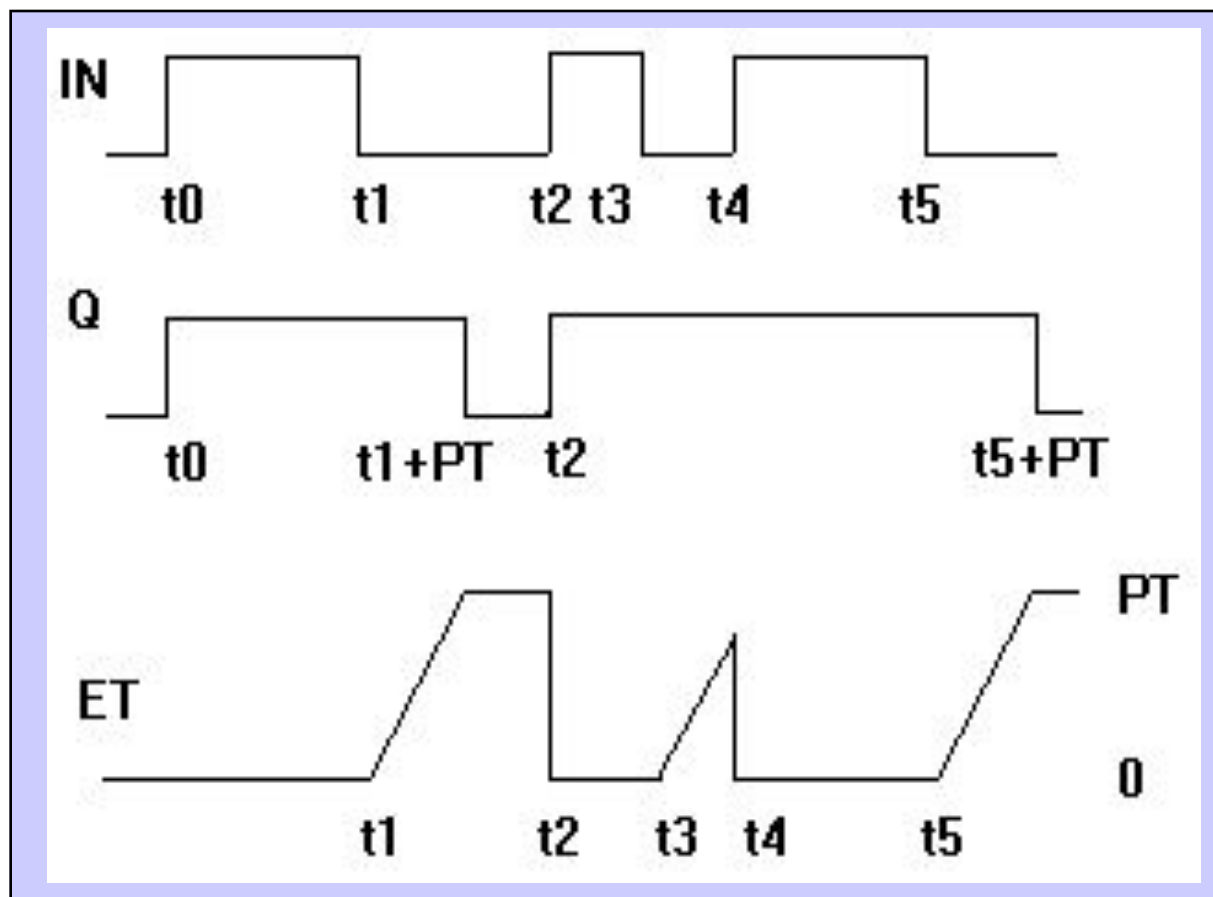
Таймер **TON**

Включает выход с задержкой по переднему фронту



Таймер **TOF**

Выключает выход с задержкой по заднему фронту



Опрос таймера...

- **Проверка состояния таймера**

- состояние таймера можно получить на выходе Q блочного элемента таймера
- состояние таймера можно с помощью NO-контакта (соответствует выходу Q) или с помощью NC-контакта (инверсия)
 - результаты считывания с помощью NO-контакта или выхода Q различаются в зависимости от типа таймера

Последовательность операций с таймерами

- Таймер будет работать правильно, если соблюдать следующий порядок
 - Запуск таймера (Start)
 - Сброс таймера (Reset)
 - Считывание значения времени или длительности
 - Проверка состояния таймера

Счетчики (counters) ...



- используют в вычислительных задачах для счета
 - по возрастанию (прямой счет)
 - по убыванию (обратный счет)
 - по возрастанию и убыванию

Счетчики располагаются в системной памяти CPU

Количество счетчиков определяется версией CPU

- счетчик сканируется путем считывания его состояния (нулевое или ненулевое значение счета) или текущего значения счетчика (значения счета, count value), которое можно получить десятичным коде

Счетчики

- **STU**

Инкрементируется по переднему фронту

- **STD**

Декрементируется по переднему фронту

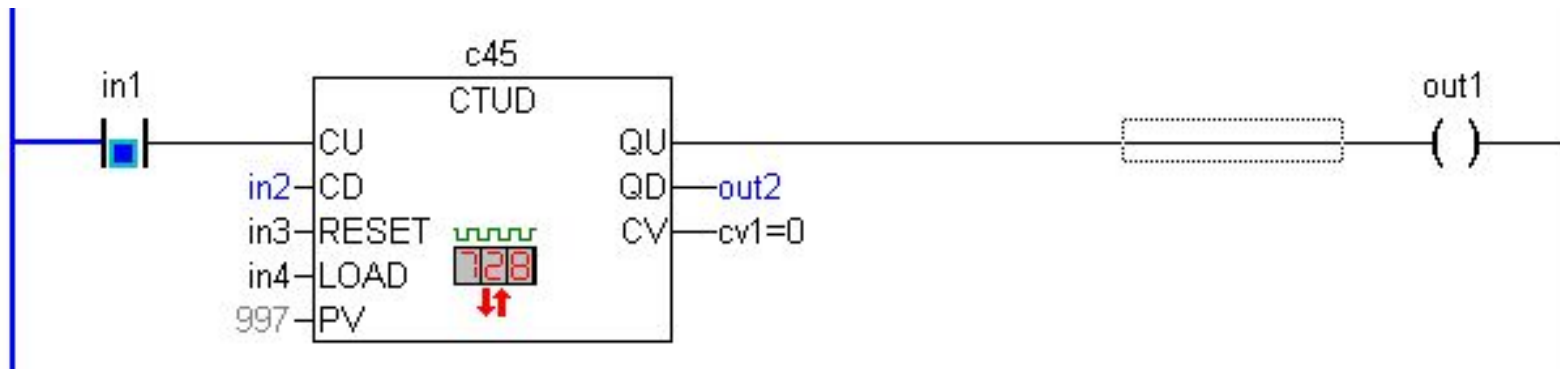
- **STUD**

Инкрементируется или
декрементируется по разным входам

Программирование счетчика...

Со счетчиками производятся следующие операции

- установка счетчика, задание значения счетчика
- прямой счет
- обратный счет
- сброс счетчика
- считывание числового значения счетчика



Последовательность операций со счетчиками....

- Счетчик будет работать правильно, если соблюдать следующий порядок
 - Счет (прямой или обратный в любом порядке)
 - Установка счетчика
 - Сброс счетчика
 - Проверка счета
 - Проверка состояния счетчика

Опрос счетчика...

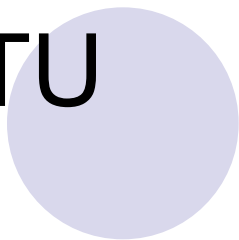
- **Проверка состояния счетчика (LAD)**

Состояние счетчика подается на выход Q блочного элемента счетчика

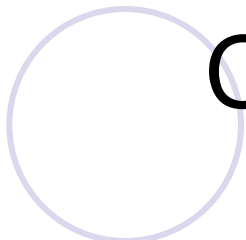
- состояние счетчика также можно проверить с использованием NO-контакта (соответствует выходу Q) или NC-контакта (инверсия)
 - выход Q содержит «1», если текущее значение счета больше нуля
 - выход Q содержит «0», если текущее значение счета равно нулю

Выход Q в блочном элементе счетчика может быть не подключен

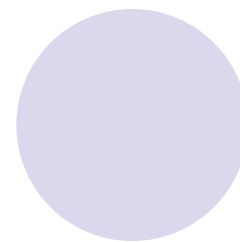
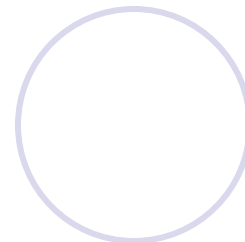
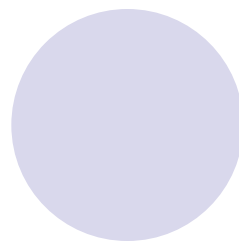
CTU



CTD



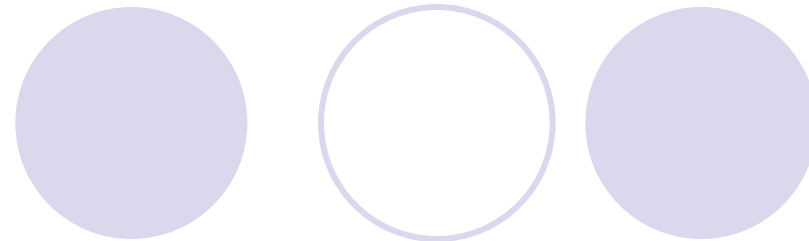
- по каждому фронту на входе CU (переход из FALSE в TRUE) выход CV увеличивается на 1
- выход Q устанавливается в TRUE, когда счетчик достигнет значения заданного PV
- счетчик CV сбрасывается в 0 по входу RESET = TRUE



- по каждому фронту на входе CD (переход из FALSE в TRUE) выход CV уменьшается на 1
- когда счетчик достигнет 0, счет останавливается, выход Q переключается в TRUE
- счетчик CV загружается начальным значением, равным PV по входу LOAD = TRUE.

СТUD TP

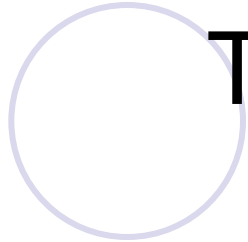
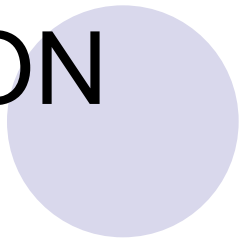
- по фронту на входе CU счетчик увеличивается на 1
- по фронту на входе CD счетчик уменьшается на 1 (до 0)
- QU устанавливается в TRUE, когда CV больше или равен PV
- QD устанавливается в TRUE, когда CV равен 0



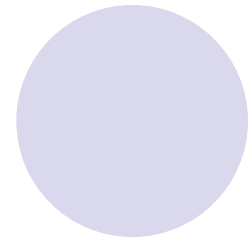
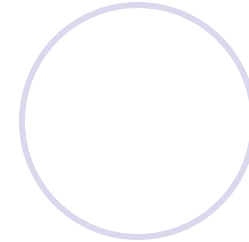
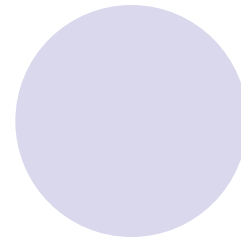
- пока IN равен FALSE, выход Q = FALSE, выход ET = 0
- при переходе IN в TRUE выход Q устанавливается в TRUE и таймер начинает отсчет времени (в мс) на выходе ET до достижения длительности, заданной PT
- далее счетчик не увеличивается

Таким образом, выход Q генерирует импульс длительностью PT по фронту входа IN

TON



TOF



- пока IN равен FALSE, выход Q = FALSE, выход ET = 0
- как только IN становится TRUE, начинается отсчет времени (в мс) на выходе ET до значения, равного PT
- далее счетчик не увеличивается
- Q равен TRUE, когда IN равен TRUE и ET равен PT, иначе FALSE

Таким образом, выход Q устанавливается с задержкой PT от фронта входа IN

- если IN равен TRUE, то выход Q = TRUE и выход ET = 0
- как только IN переходит в FALSE, начинается отсчет времени (в мс) на выходе ET
- при достижении заданной длительности отсчет останавливается
- выход Q равен FALSE, если IN равен FALSE и ET равен PT, иначе - TRUE

Таким образом, выход Q сбрасывается с задержкой PT от спада входа IN