



Периферия 16-битных контроллеров





Цели работы

- Когда мы завершим работу, вы будете иметь представление о:
 - Архитектуре PIC24F/H, dsPIC30/33
 - Системных модулях 16-битных контроллеров
 - Периферийных модулях 16-битных контроллеров



Содержание работы

- **Краткий обзор архитектуры**
 - Память программ
 - Память данных
 - Доступ к памяти программ
 - Стек
 - Режимы адресации
 - Набор инструкций
 - Прерывания и исключения

- **Системные модули**
 - Система тактирования и энергосберегающие режимы
 - Система сброса
 - сторожевой таймер
 - Защита кода CodeGuard™



Содержание работы (продолжение...)

❶ Периферийные модули

- Порты ввода-вывода
 - ❶ **Лабораторная работа #1**
- Таймеры
- Модули захвата и сравнения
- Компараторы и источник опорного напряжения
- Модуль I²C
- Параллельный порт
 - ❶ **Лабораторная работа #2**
- Модули управления двигателями и источниками питания



Содержание работы (продолжение...)

- ❖ Основные периферийные модули (продолжение)
 - Модуль часов реального времени и календаря
 - ❖ Лабораторная работа #3
 - UART
 - ❖ Лабораторная работа #4
 - Модуль вычисления CRC
 - 10- и 12-битные АЦП
 - ❖ Лабораторная работа #5
 - DMA
 - SPI
 - ❖ Лабораторная работа #6

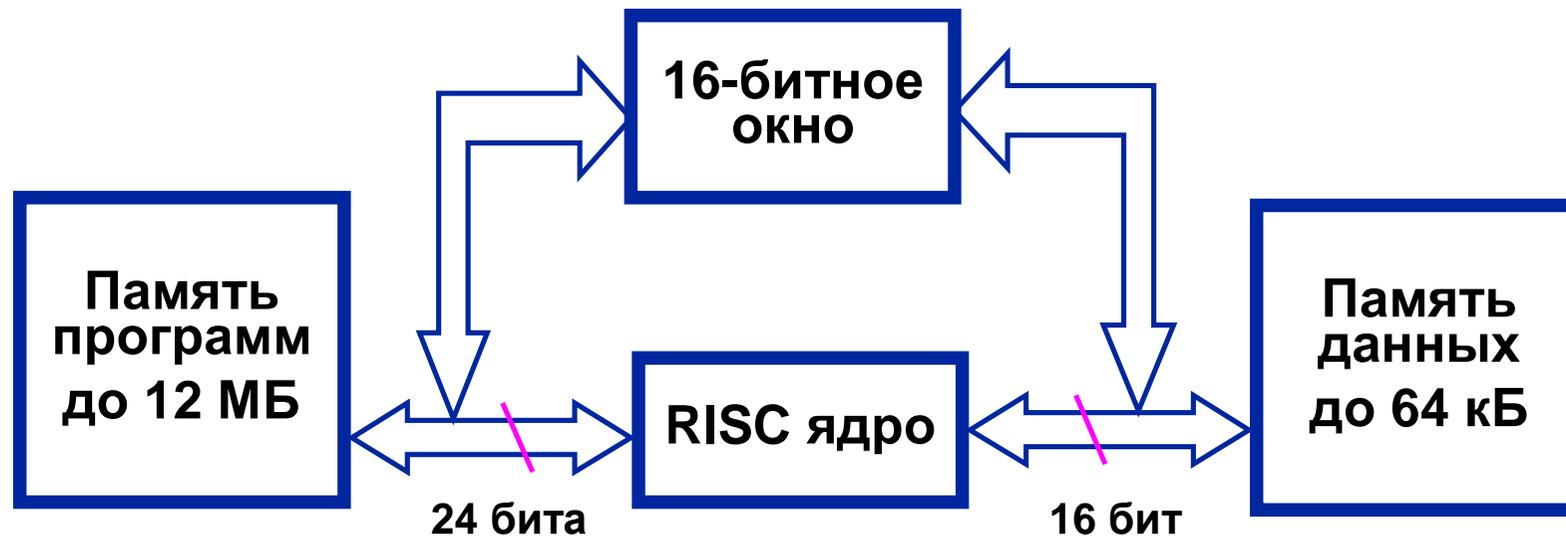


Основы 16-битной архитектуры



Гарвардская архитектура

- 16-битное ядро
- Размер инструкции - 24 бита
- Механизм передачи данных



Конвейер выполнения инструкций

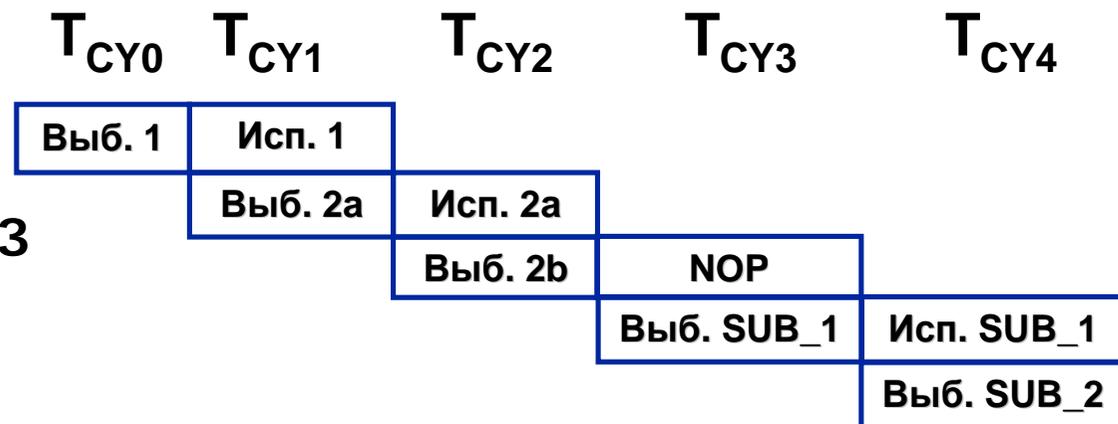
- ⌚ Одновременная выборка и выполнение
- ⌚ Почти все инструкции выполняются за один такт
- ⌚ Инструкции перехода (GOTO, CALL) выполняются за два такта

1. MOV W0, PORTB

2. CALL SUB_1

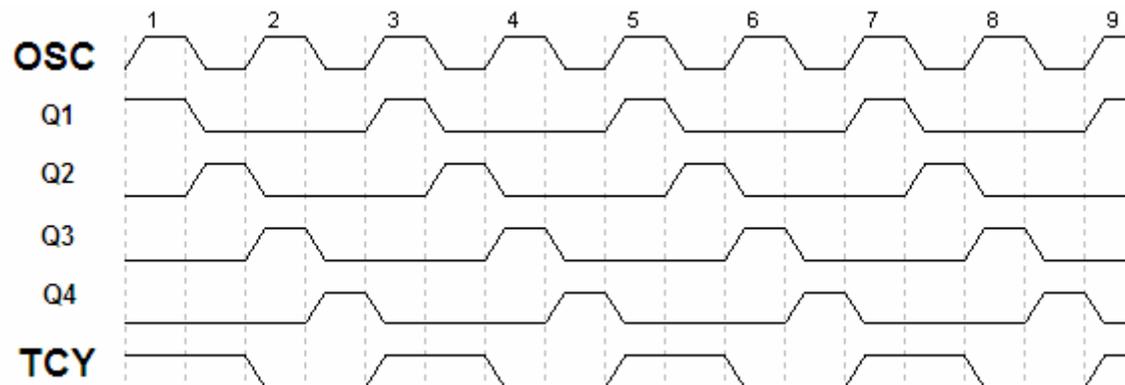
3. SUB_1

4. BSET PORTA, RA3

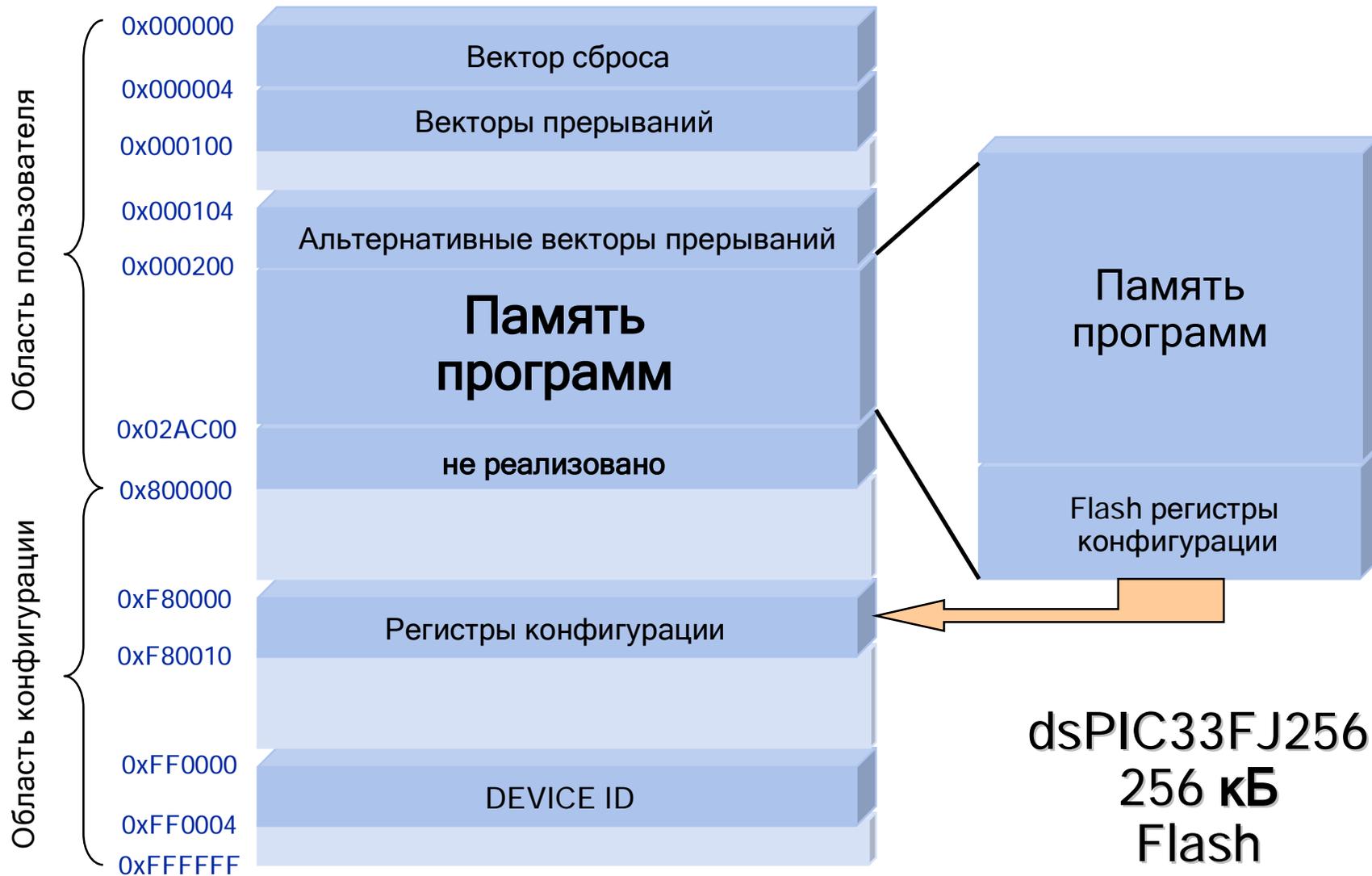


Тактирование

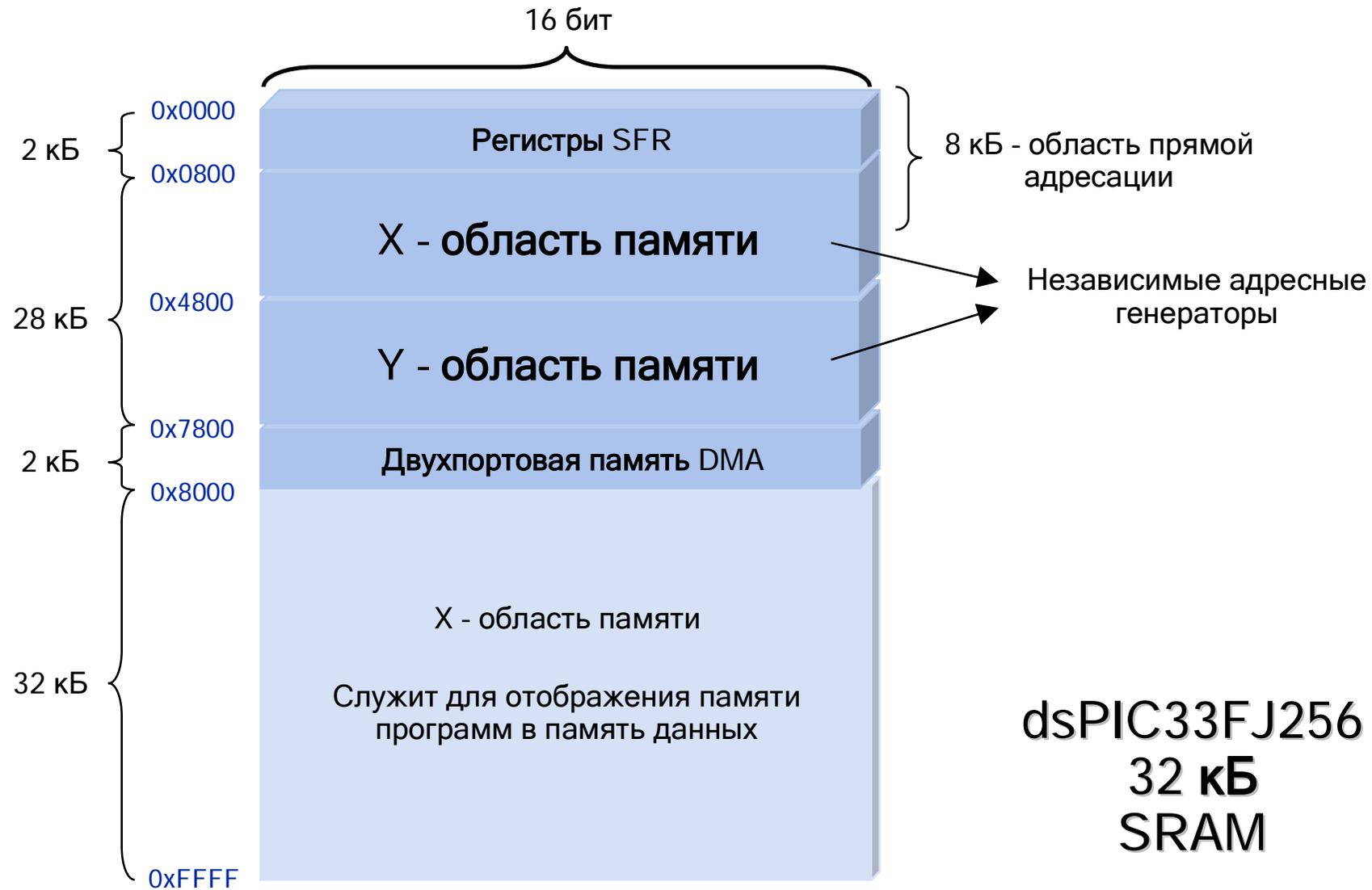
- Частота выполнения инструкций = $1/2$ частоты тактирования микроконтроллера
- PIC24F
 - 16 MIPS @ 32 МГц
- dsPIC30
 - 30 MIPS @ 120 МГц
- PIC24H / dsPIC33
 - 40 MIPS @ 80 МГц



Память программ



Память данных



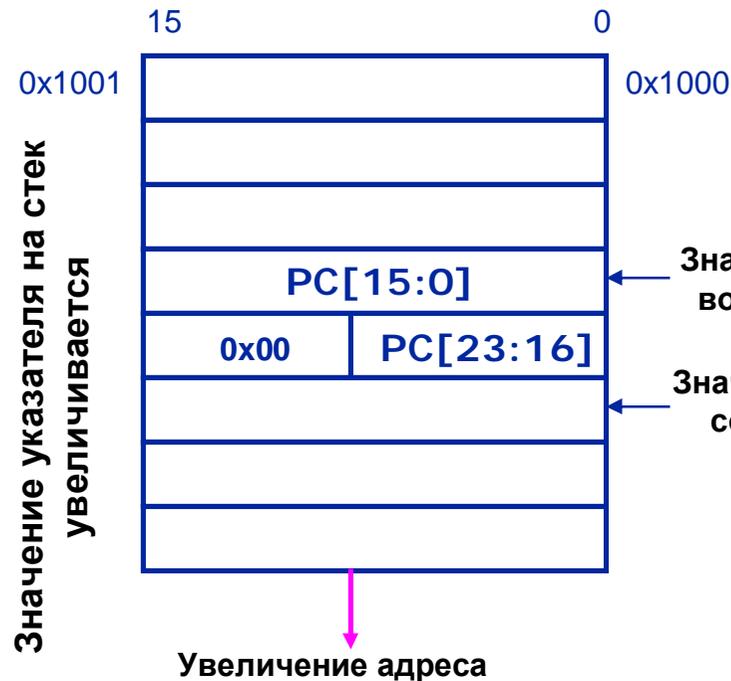


Доступ к памяти программ

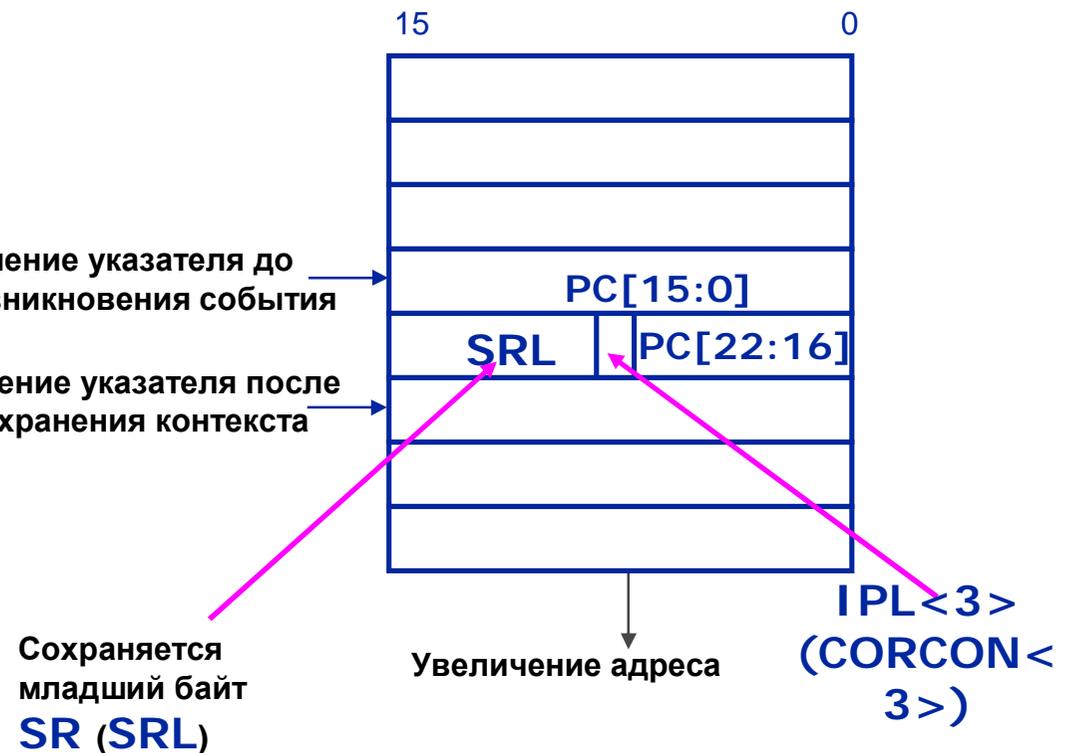
- **Выборка инструкций в ходе выполнения**
- **Инструкции табличного чтения/записи**
 - доступ ко всему слову инструкции
 - запись - посекторная
 - выполняются за два командных такта
 - пре- и пост- инкремент и декремент указателя
- **PSV - Programm Space Visibility**
 - доступ только к 16 младшим битам инструкции
 - запись - запрещена
 - доступно всем инструкциям работы с памятью

Программный стек

Вызов (call)



Прерывание или исключение



Замечание: SP всегда указывает на следующее свободное слово в стеке



Режимы адресации

Прямая адресация

Определяет
операцию как
байтовую

Адрес слова данных
($< 8 \text{ Кб}$)

ADD.b 0x900, **WREG**

Инструкция

Если определен в инструкции,
результат сохраняется в WREG (W0)

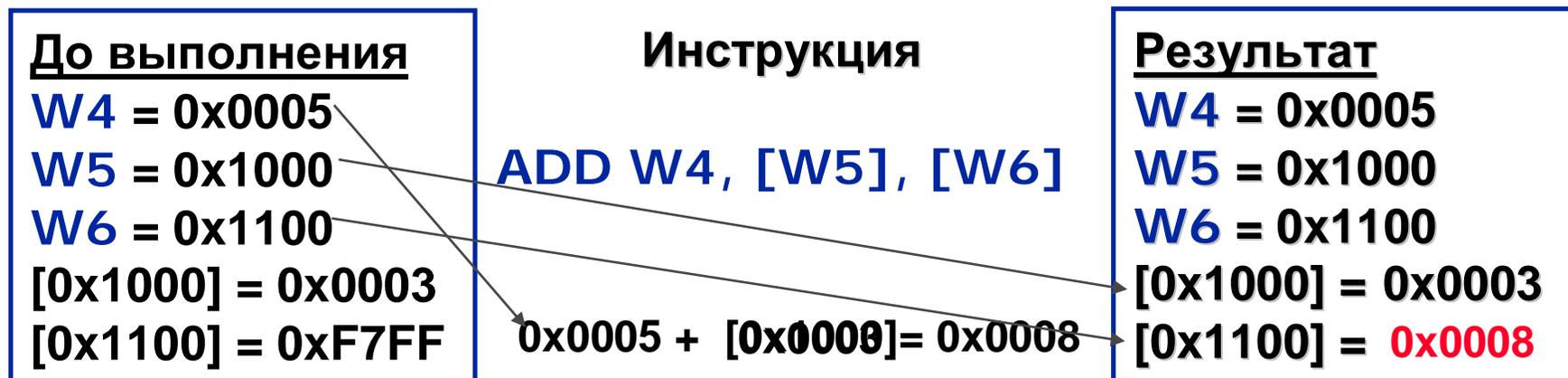


Прямая адресация регистров

- Доступ к набору W регистров
 - $W0$ – $W15$ ($0x0000$ - $0x001F$ в памяти данных)
 - Поддержка доступа к слову и байту слова
 - Используется, когда данные находятся в W регистрах
- Например,
 - $IOR \quad W2, W4, W6$
 - Побитовое ИЛИ значений $W2$ и $W4$, сохранение результата в $W6$

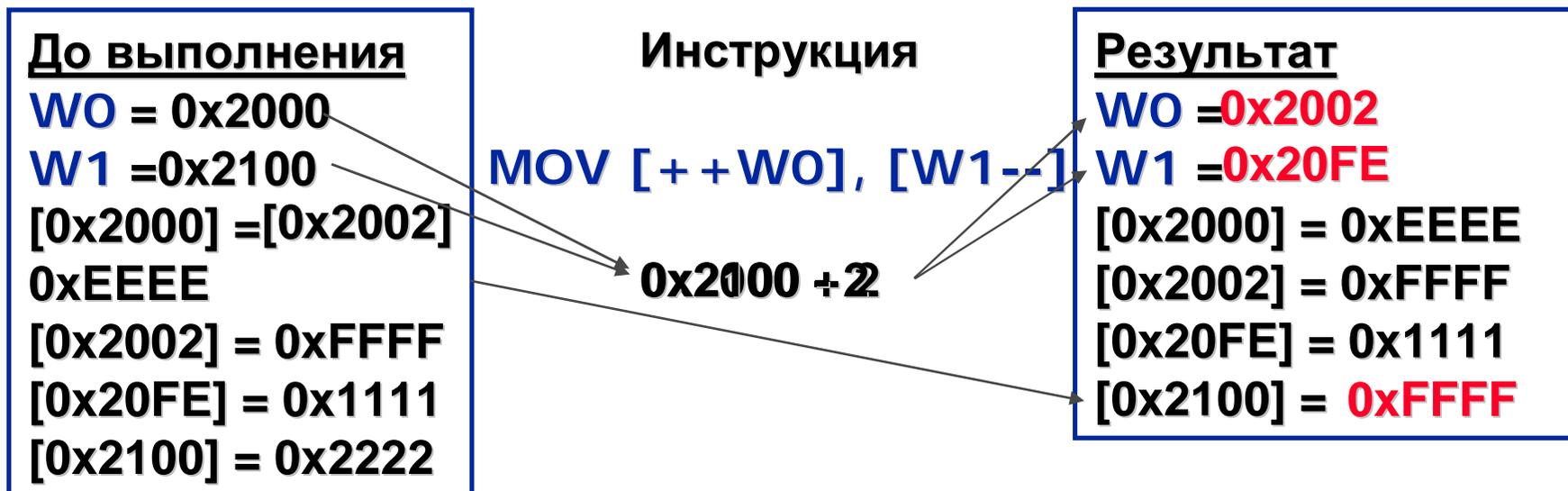
Косвенная адресация

- **Значения W регистров является указателями на память**
 - Обеспечивает доступ ко всему объему адресуемой памяти (64 КБ)
 - Режим косвенной адресации возможен в большинстве инструкций



Косвенная адресация с пре- и пост- модификацией

- ⊕ Позволяет изменять значения указателей (декремент, инкремент)
 - Си-подобная пре- и пост- модификация указателей
 - Изменение значения указателей на 1 при доступе к байту и на 2 при доступе к слову



Косвенная адресация со смещением указателя

- Указатель формируется путем сложения значений двух W регистров
 - При этом значения W регистров не меняются





Набор инструкций

Обзор системы команд

- **83 инструкции (dsPIC30/dsPIC33), 76 инструкций (PIC24F/H)**
 - Большинство инструкций занимают одно слово памяти программ (24 бита) за исключением некоторых, занимающих два слова
- **Все инструкции выполняются за 1 командный цикл, за исключением**
 - Инструкций управления ходом программы (2 или 3 цикла)
 - Табличных инструкций (2 цикла)
 - **MOV.D** инструкций (2 цикла)
 - **DO** инструкций (2 цикла)
 - Инструкций деления* (18 циклов)

* Инструкция деления является итерационной (выполняется в цикле REPEAT) и может быть прервана в любой момент (в соответствии с условиями прерывания цикла REPEAT)

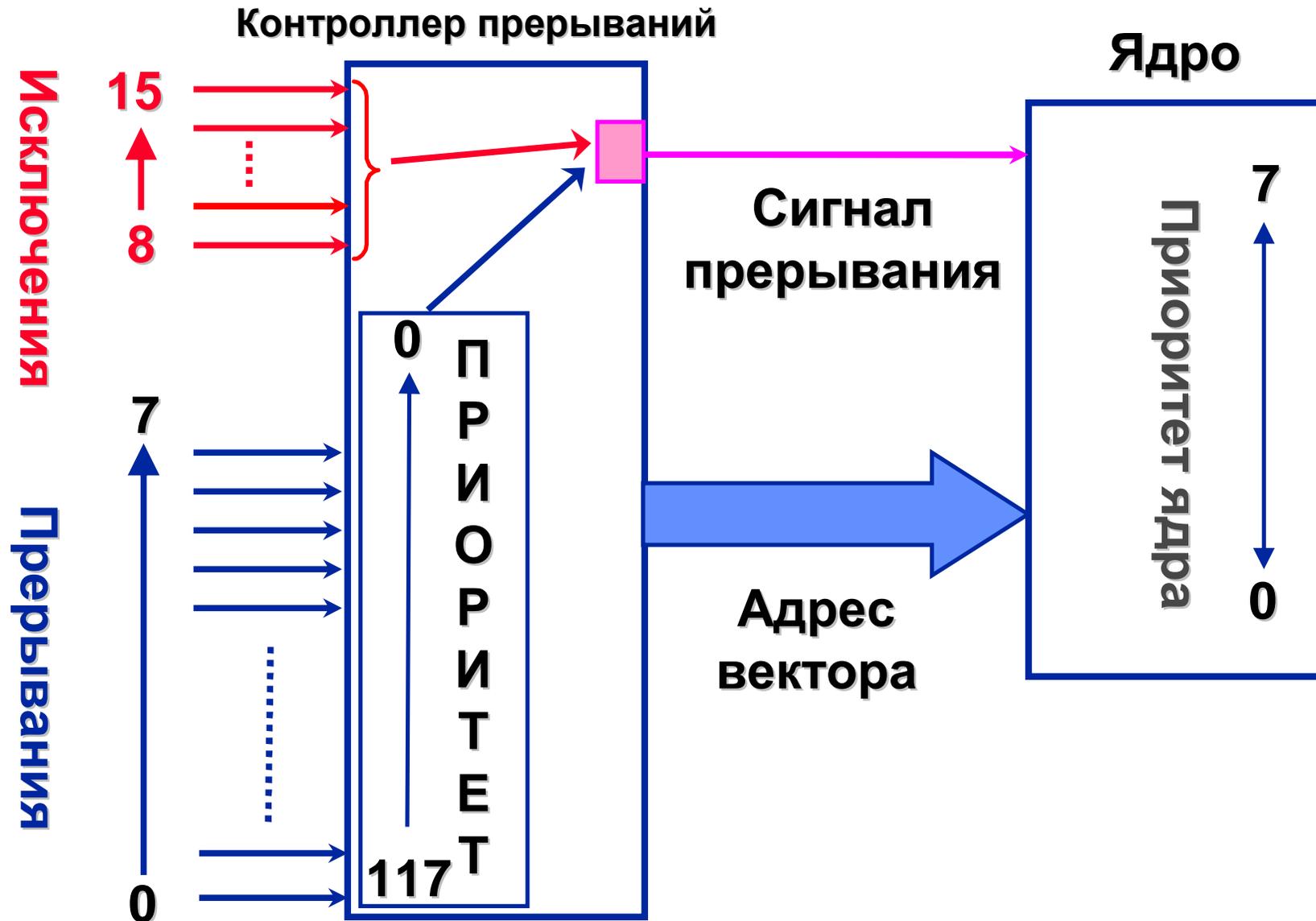
Наборы инструкций

- ❶ Систему команд можно условно разделить на следующие наборы инструкций:
 - (MOVE) Инструкции перемещения
 - (MATH) Математические инструкции
 - (LOGIC) Логические инструкции
 - (SHIFT / ROTATE) Инструкции сдвига
 - (BIT) Битовые инструкции
 - (STACK) Инструкции работы со стеком
 - (PROGRAM FLOW) Инструкции управления ходом программы
 - (CONTROL) Инструкции управления
 - (DSP) DSP инструкции (только для *dsPIC*)



Прерывания и исключения

Система прерываний



Прерывания

- ❶ **Таблица прерываний (IVT) имеет уникальный вектор прерывания для каждого источника**
 - 8 не маскируемых векторов для исключений (ловушек, «trap»)
 - от 54 (dsPIC30) до 118 (dsPIC33) векторов прерываний
- ❷ **Вектор содержит адрес функции обработки прерывания**
- ❸ **7 уровней приоритета, назначаемых пользователем для каждого прерывания**
- ❹ **Альтернативная таблица векторов**
- ❺ **Быстрая обработка прерываний**
 - Время реакции на прерывание - максимум 5 командных тактов
 - Возврат из прерывания - максимум 3 командных такта

Исключения

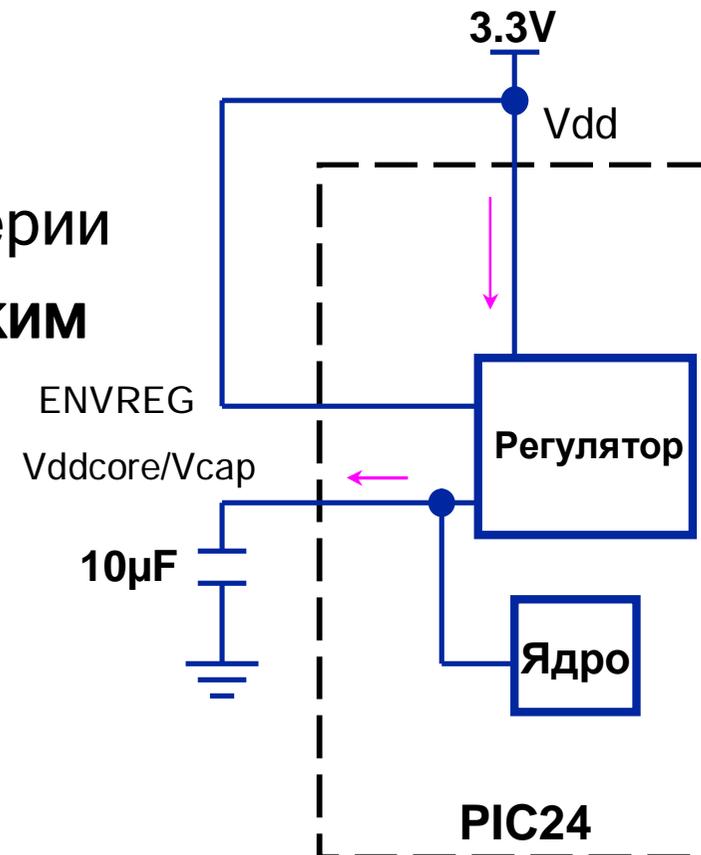
- ❖ **Программные («soft») исключения**
 - Уровень приоритета от 8 до 12
 - Возможно нормальное продолжение работы
 - Пример: ошибка АЛУ (деление на ноль)
- ❖ **Аппаратные («hard») исключения**
 - Уровень приоритета от 13 до 15
 - Выполнение программы приостанавливается
 - Продолжение работы возможно только в случае устранения источника исключения
 - Пример: ошибка тактового генератора, ошибка доступа к стеку



Системные модули

Система питания

- ❶ Встроенный регулятор напряжения питания ядра
 - 2.5 В - для питания ядра
 - 3.3 В - для питания периферии
- ❷ Необходим конденсатор с низким ESR
 - Рекомендуется 10 μF
- ❸ Вывод ENVREG (PIC24F)
 - включение/выключение регулятора
- ❹ BOR





Режимы энергосбережения

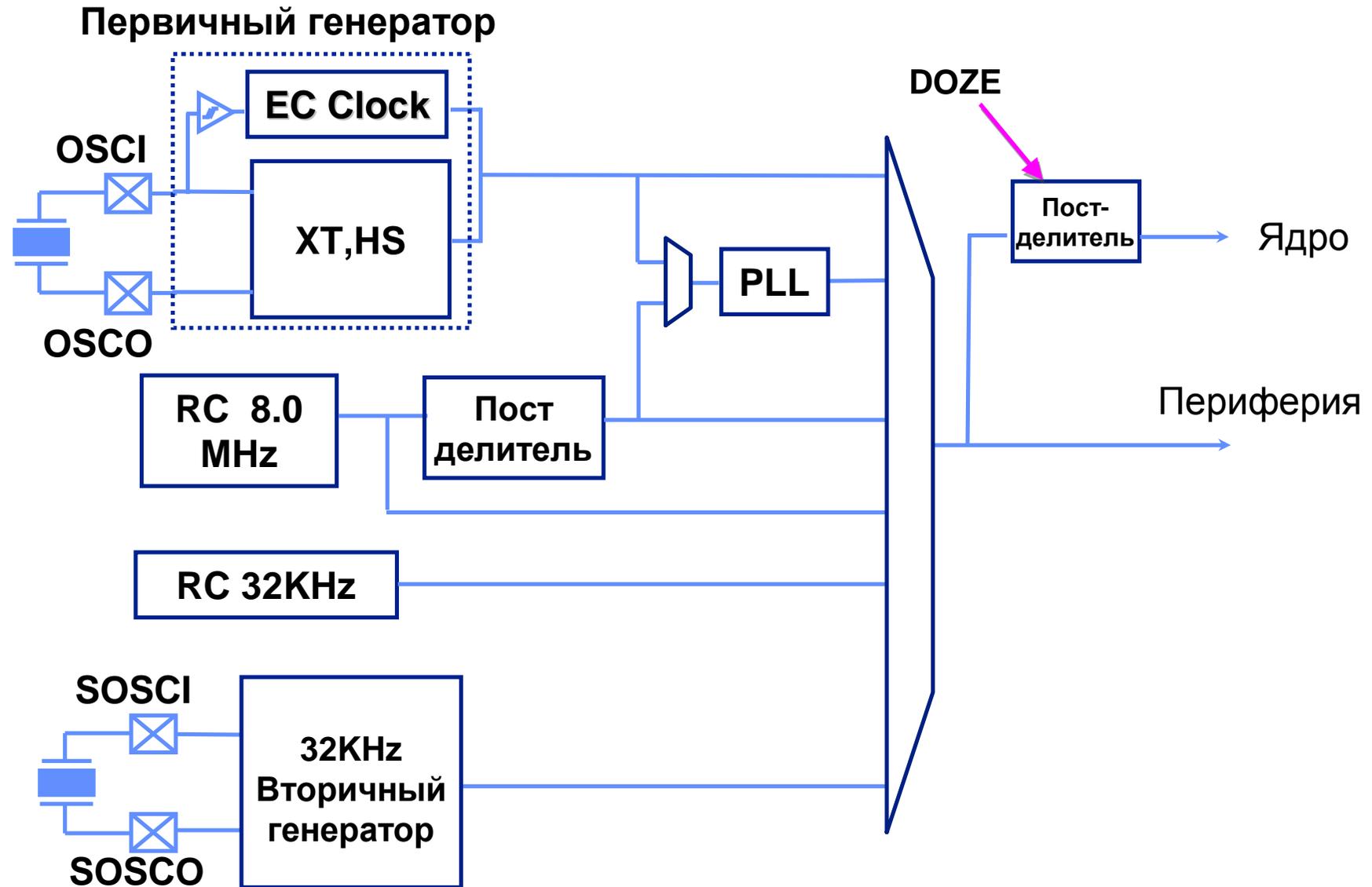
- ❖ Снижение напряжения питания при сохранении максимальной тактовой частоты
- ❖ Снижение тактовой частоты
 - Постделитель
 - Изменение источника тактирования в реальном времени
- ❖ Полное отключение неиспользуемой периферии
- ❖ Вход в режим энергосбережения - инструкция [pwrsav](#)
 - режим SLEEP
 - режим IDLE
- ❖ Режим DOZE



Режимы энергосбережения

- ❶ Выход из режимов энергосбережения
 - Сброс
 - сторожевой таймер
 - Любое немаскированное прерывание
 - ❶ Приоритет прерывания \leq приоритету ядра
 - Выполнение следующей инструкции
 - ❶ Приоритет прерывания $>$ приоритета ядра
 - Переход на вектор прерывания
- ❶ Как узнать какое событие «разбудило» контроллер?
 - проверить биты **RCON<SLEEP>** и **RCON<IDLE>**

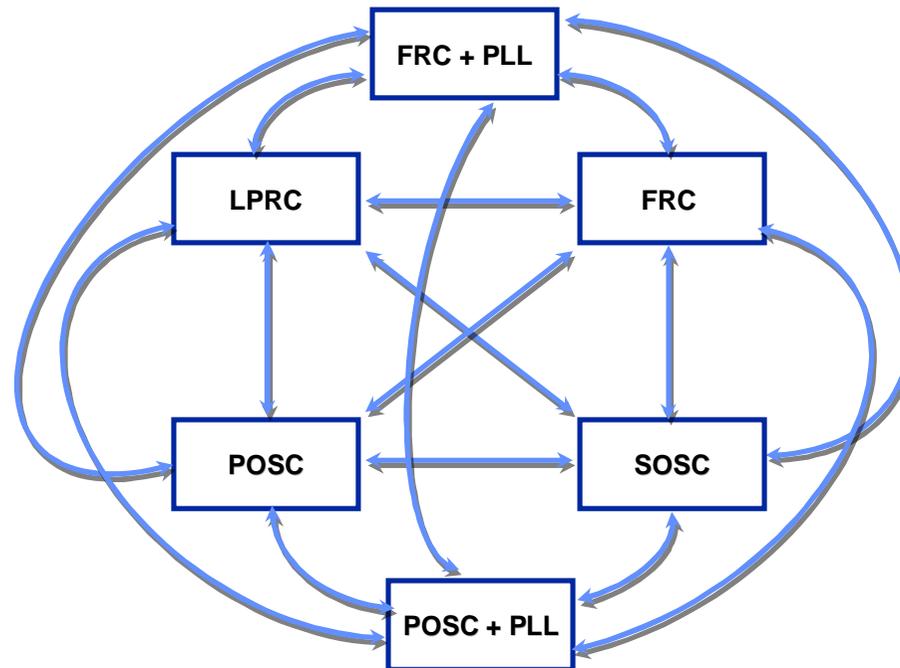
Схема тактирования



Технология энергосбережения

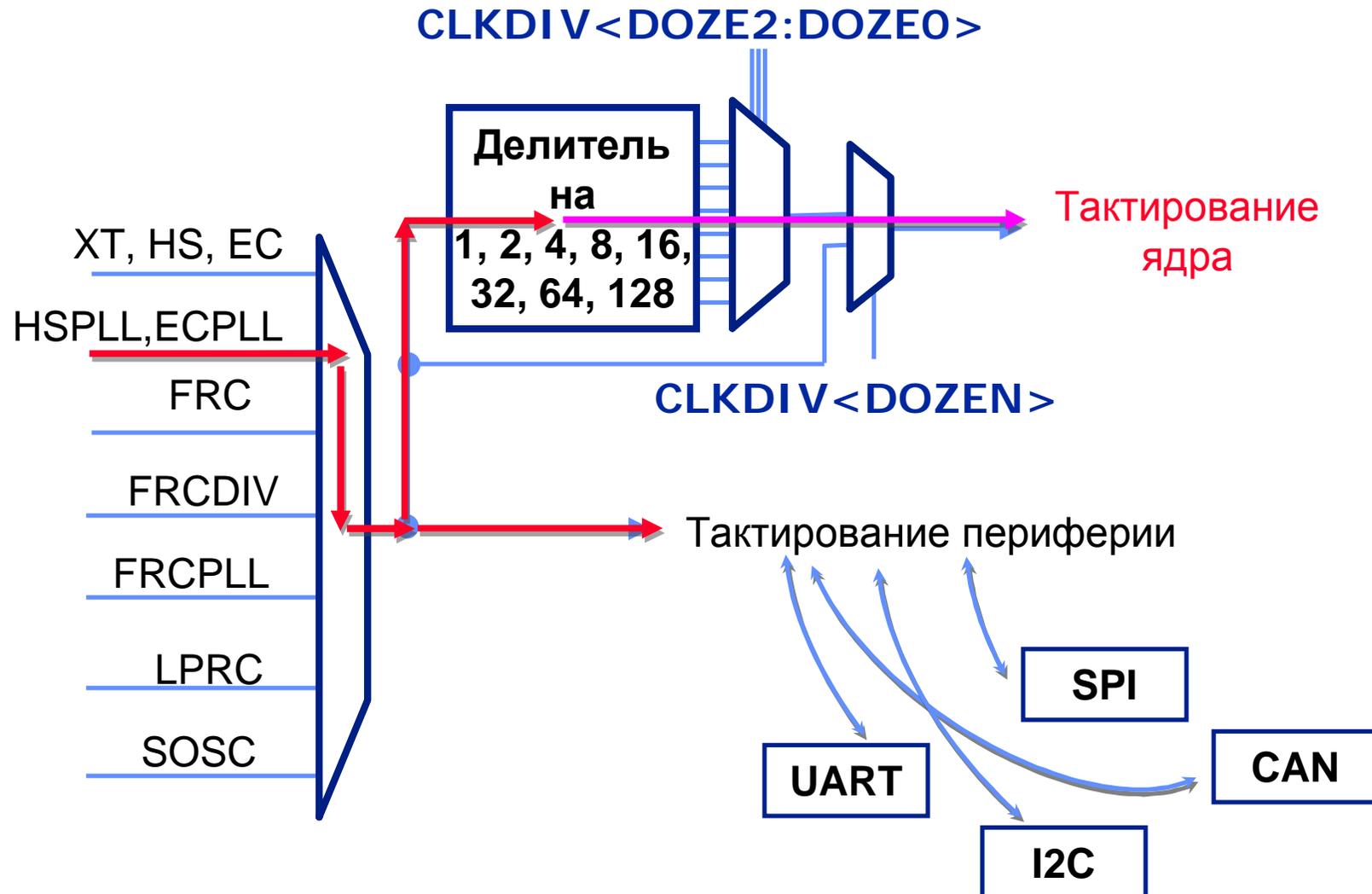


Переключение источников тактирования на лету



- `OSCON<OSWEN>` бит разрешает переключение источников тактирования
- `OSCON<NOSC2:NOSC0>` биты служат для выбора нового источника

Режим DOZE





Монитор схемы тактирования

- ❖ Что произойдет при сбое основного тактового генератора?
 - Схема тактирования автоматически переключится на FRC генератор, если мониторинг схемы тактирования разрешен
 - ❖ Мониторинг контролируется битом `FOSC<FCKSM>`
 - Генерируется аппаратное исключение
 - В обработчике исключения необходимо:
 - ❖ Сбросить флаг сбоя схемы тактирования `OSCCON<CF>`
 - ❖ Сбросить флаг аппаратного исключения `INTCON1<OSCFAIL>`
 - ❖ Переключиться на требуемый источник тактирования с FRC
 - ❖ Выйти из обработчика исключения

Система сброса

- ❶ **Возможные источники сброса**
 - Сброс при включении питания (**POR**)
 - Внешний сброс (**MCLR**)
 - Инструкция RESET (**SWR**)
 - Сторожевой таймер (**WDT**)
 - Снижение напряжения питания (**BOR**)
 - Конфликт исключения (**TRAPR**)
 - Неизвестный код инструкции (**IOPUWR**)
 - Неинициализированные W регистры (**UWR**)
 - Неизвестный код конфигурационного слова (**CM**)
- ❷ **Вектор сброса для всех источников = 0h**

Сторожевой таймер

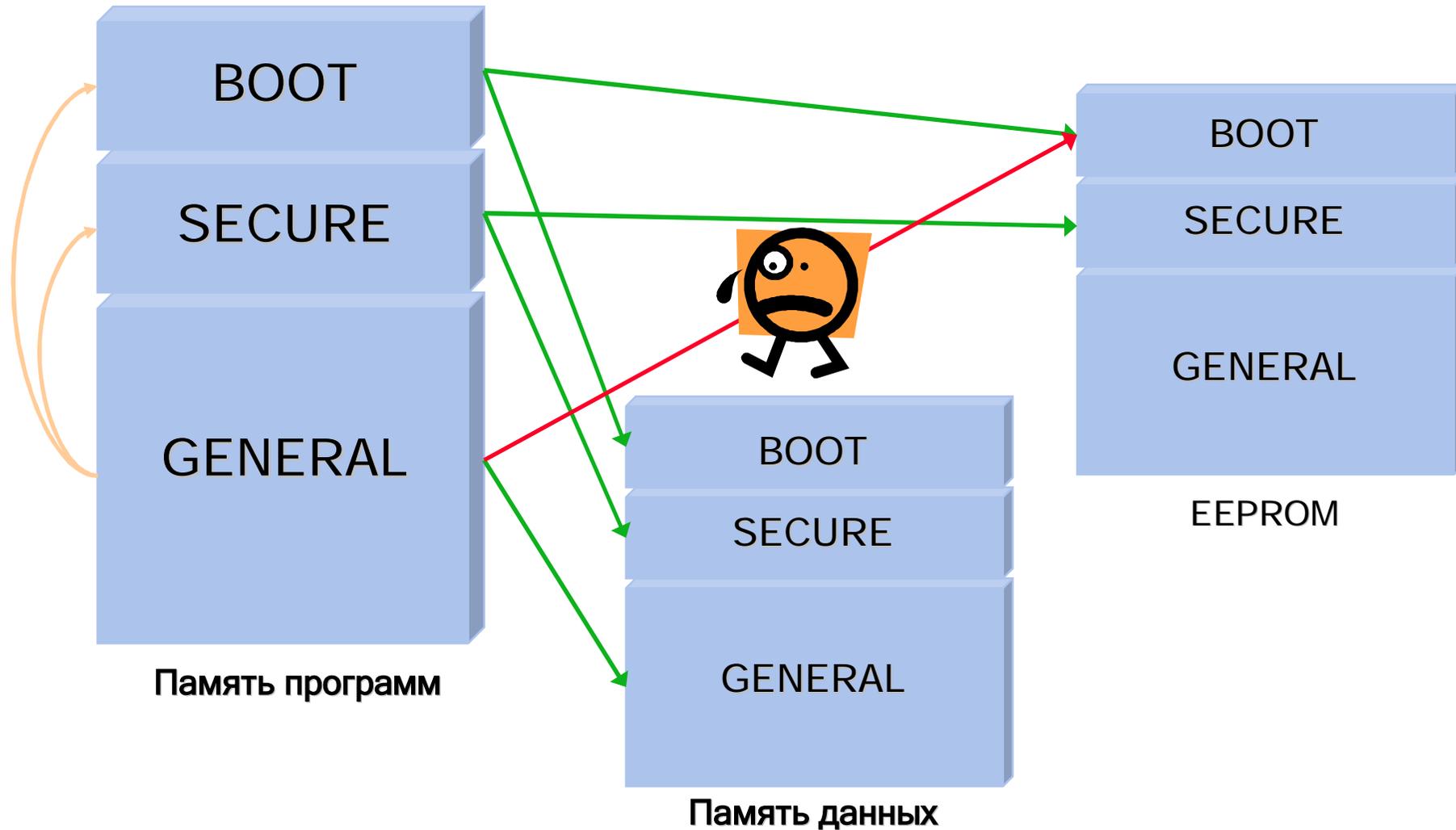
- ❖ Защита от сбоя программы
- ❖ Генерирует сброс, если таймер не обнулен за определенное время
 - Сброс таймера должен выполняться периодически (инструкция **CLRWDT**)
- ❖ Программируемый период
 - от 1 мс до 131 сек
- ❖ Программируемы пред- и пост- делитель в конфигурационном слове
- ❖ Разрешение работы как в конфигурационном слове, так и программно
- ❖ Оконный режим работы
 - **CLRWDT** инструкция должна быть выполнена в последней четверти периода таймера. В противном случае генерируется сброс.



Защита кода CodeGuard™

- ❖ Три уровня реализации
 - Основной
 - Промежуточный
 - Расширенный
- ❖ Разбивка областей памяти (программ данных, EEPROM) на три сегмента
 - Загрузочный (*boot*)
 - Секретный (*secure*)
 - Основной (*general*)
- ❖ Размеры сегментов конфигурируются
- ❖ Каждый сегмент может иметь два уровня защиты
 - Высокий - программный доступ возможен только к первым 32 словам сегмента
 - Стандартный - программный доступ возможен ко всему сегменту
- ❖ Секретный и основной сегменты имеют ограниченный доступ к загрузочному
- ❖ Основной сегмент имеет ограниченный доступ к защищенному

Защита кода CodeGuard™





Периферийные модули

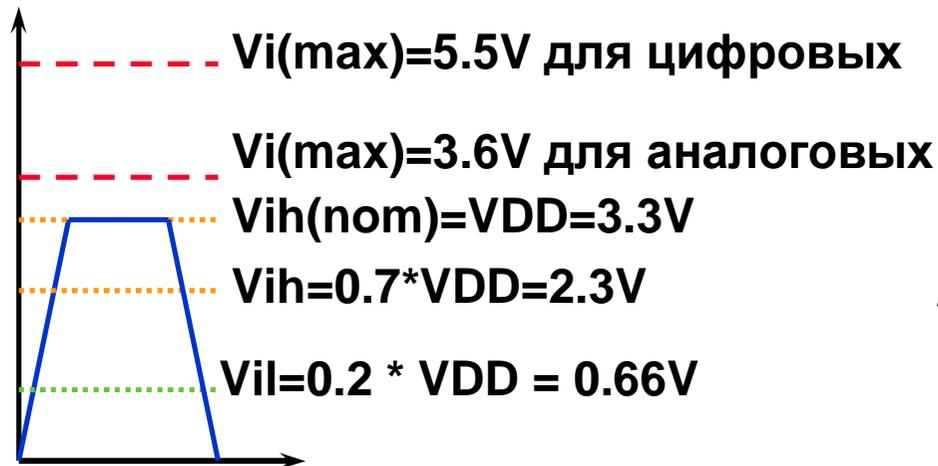


Порты ввода-вывода

- **Защитные диоды на всех выводах**
- **Триггеры Шмитта на всех выводах в режиме входа**
- **Генерирование прерывания по изменению состояния (24 вывода @ 80 выводной корпус)**
- **Мультиплексор периферии (18 и 28 выводные PIC24/dsPIC33)**

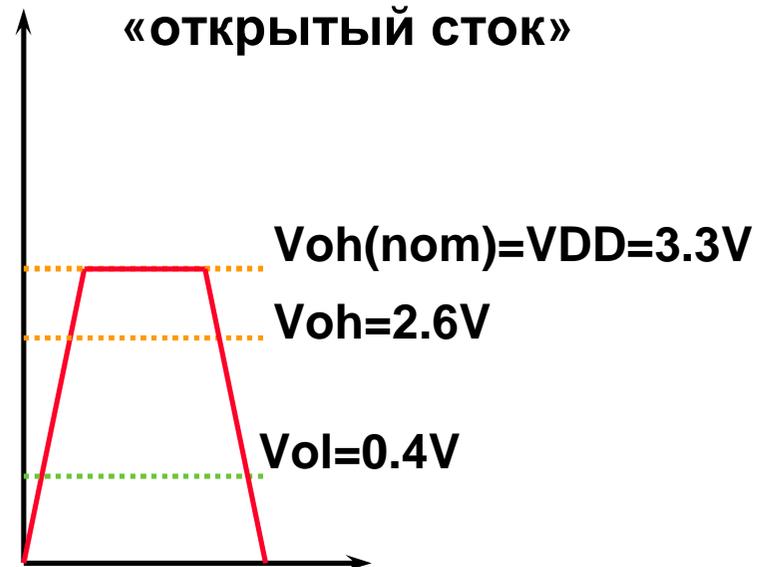
Логические уровни

Напряжение питания $V_{DD} = 3.3V$



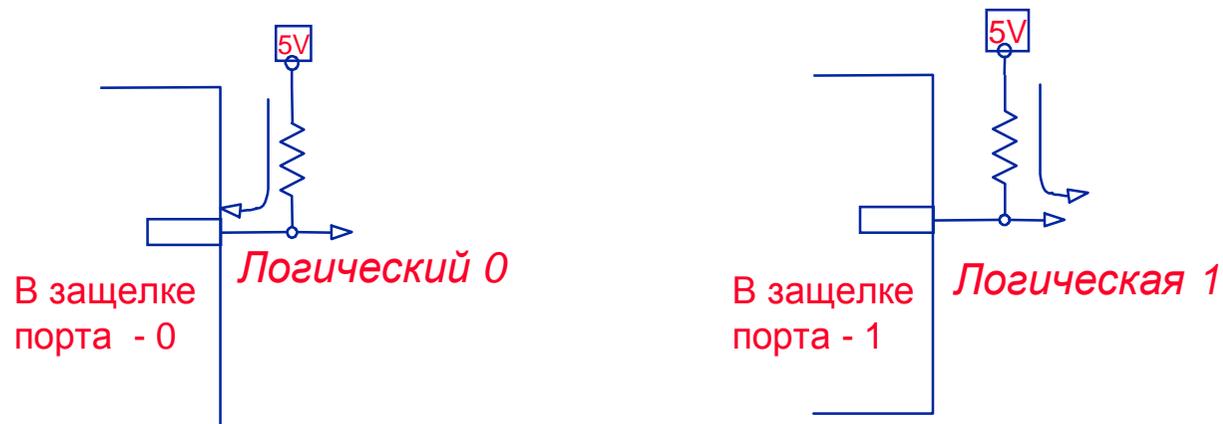
Все выводы обеспечивают
 втекающий/вытекающий ток:
 25 мА (dsPIC30)
 18 мА (PIC24F)
 4 мА (dsPIC33 / PIC24H)

Все цифровые
 выводы могут быть
 сконфигурированы
 для работы в режиме
 «открытый сток»



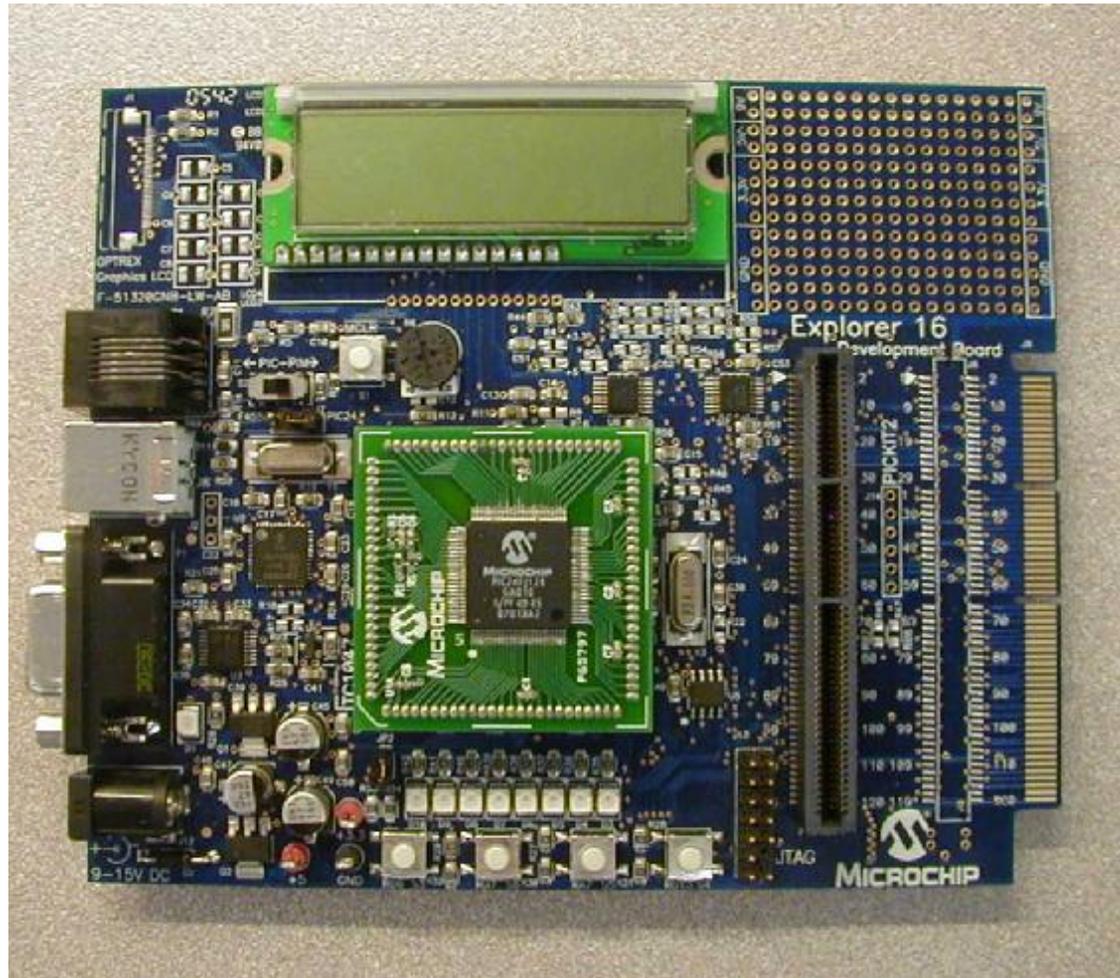
Выход с открытым стоком

- ❖ Интерфейс с 5 вольтовыми устройствами
 - Порт должен быть сконфигурирован на выход
 - Регистры ODCx обеспечивают переключение режима работы вывода



Лабораторная работа №1

🔧 Отладочная плата - Explorer 16





Лабораторная работа №1

🌀 Файл проекта

– `iooport`

🌀 Цель работы

– Конфигурация портов ввода-вывода

– Корректный доступ к портам

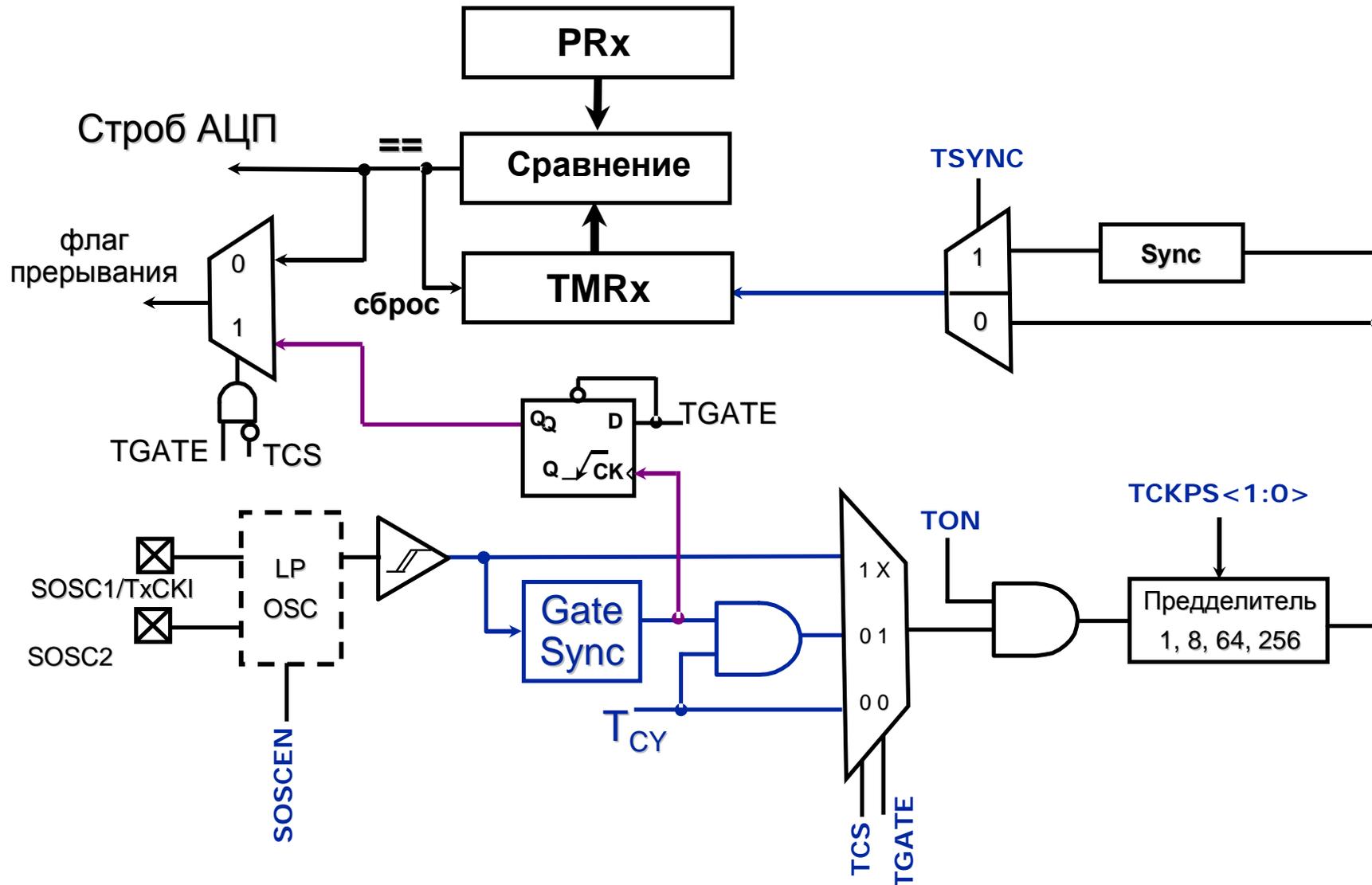
🌀 Для чтения - PORT

🌀 Для записи - LAT

Таймеры

- **От 5-и до 9-и 16 разрядных таймеров**
 - Таймеры могут быть объединены для работы в 32-битном режиме
 - Все таймеры имеют регистр периода
 - Все таймеры имеют предделитель
 - Режим разрешения счета
- **Три вида таймеров**
 - TMR A - кварцевый генератор 32 кГц
 - TMR B - синхронизация после предделителя
 - TMR C - строб для запуска АЦП

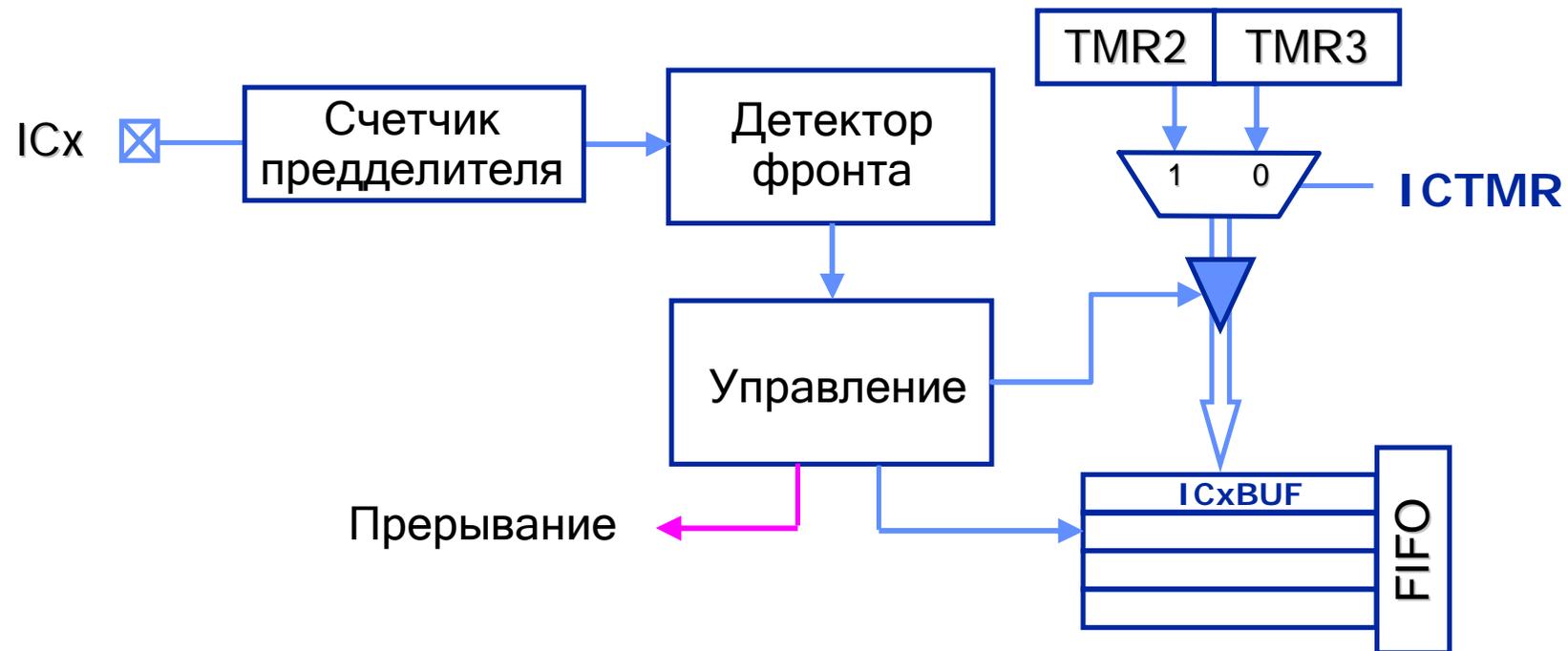
Структурная схема таймера



Модули захвата

- ❶ **Физически независимы**
 - Предделитель на 4 и 16
- ❷ **4-уровневый буфер FIFO**
 - Величина выбирается пользователем
 - Прерывание при заполнении буфера
- ❸ **Разрешение - 1 командный такт**
- ❹ **Возможность выбора базового таймера**
- ❺ **Конфигурация фронта захвата**

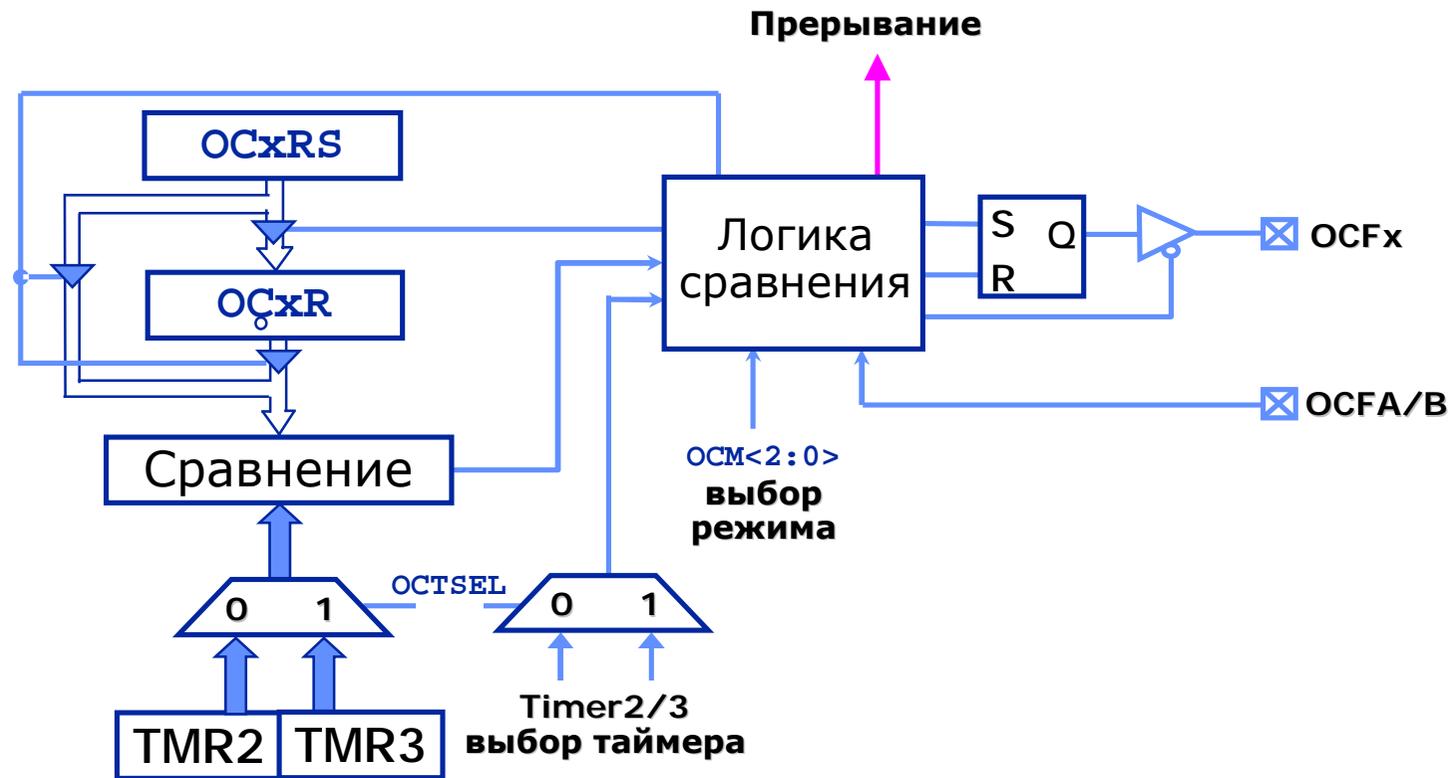
Структурная схема модуля захвата



Модули сравнения

- ❶ Два регистра сравнения
- ❶ Функция аппаратного отключения
- ❶ Возможность генерации набора сигналов для управления двигателями и источниками питания
 - С учетом паузы на переключения мощных ключей
- ❶ Разрядность 10 бит на частоте 39 кГц (40 MIPS)
- ❶ Минимальная длительность импульса - 1 T_{CY}

Структурная схема модуля сравнения

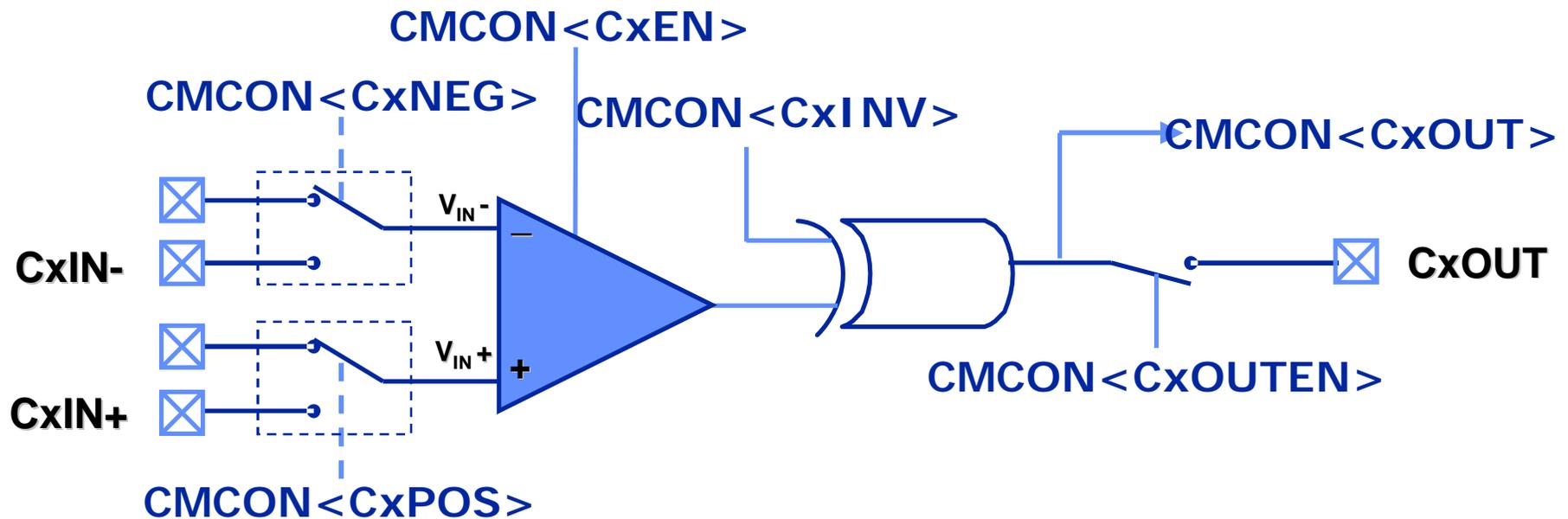




Компараторы

- **Только в PIC24F**
- **Два аналоговых компаратора**
 - гибкий мультиплексор выводов
 - аппаратная инверсия
 - конфигурируемый источник опорного напряжения
- **Физические параметры аналогичны компараторам PIC16/PIC18**

Структурная схема компаратора

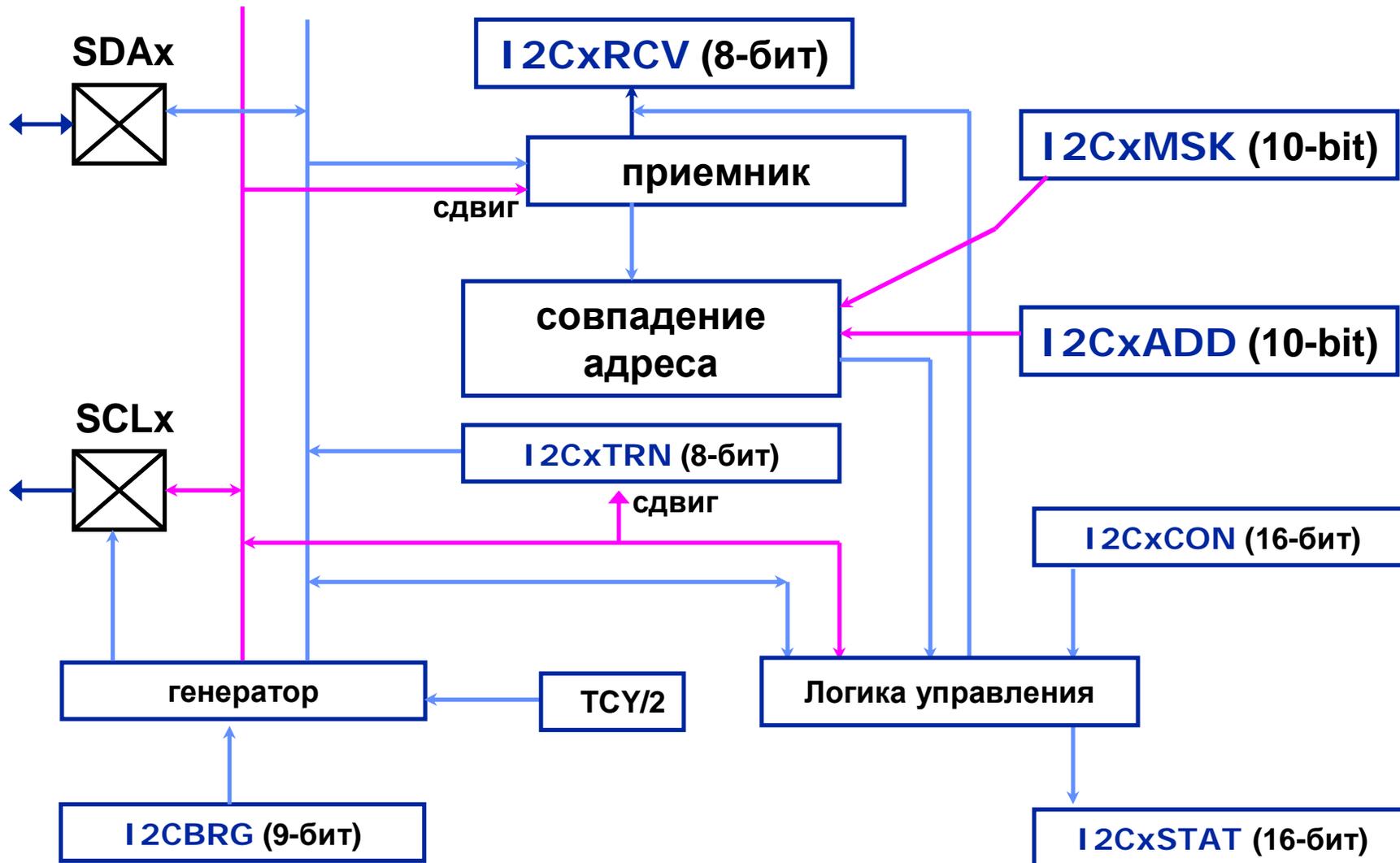




Модуль I²C

- ❖ **Физически независимый модуль**
- ❖ **Полная поддержка режимов**
 - ведущий
 - ведомый
- ❖ **8- и 10-битная адресация**
- ❖ **Поддержка общего вызова и зарегистрированных адресов**
- ❖ **Режим повторителя**
- ❖ **Уровни I²C и SMBus**
- ❖ **Прерывание при совпадении адреса с маской**
- ❖ **Спецификации**
 - 100 кГц
 - 400 кГц
 - 1 МГц

Структурная схема модуля I²C

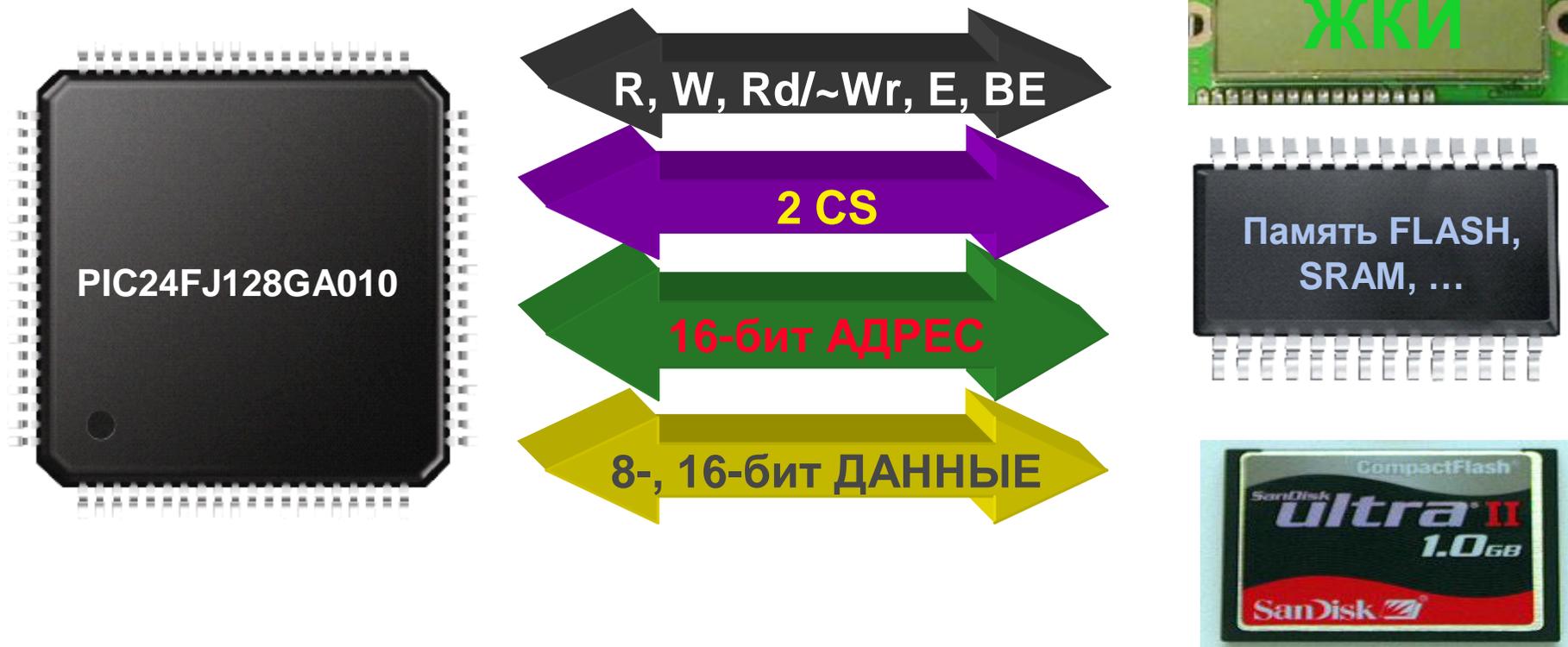




Параллельный порт

- ❖ Только в PIC24F
- ❖ 16-битная шина адреса
- ❖ 8- или 16-битная шина данных
- ❖ Две линии выбора устройства CS
- ❖ Автоинкремент и декремент адреса
- ❖ Программируемый активный уровень для всех управляющих сигналов
- ❖ Программируемое мультиплексирование шины данных и адреса
- ❖ Сигнал Byte Enable для работы с 16-битными устройствами
- ❖ Независимые стробы Read и Write или строб Rd/Wr и строб Enable
- ❖ Программируемые временные параметры
- ❖ Расширенный режим «ведомого»
 - 4-уровневые буферы на чтение и запись
- ❖ Выбор уровней сигналов
 - TTL
 - триггер Шмитта

Параллельный порт





Лабораторная работа №2

📁 Файл проекта

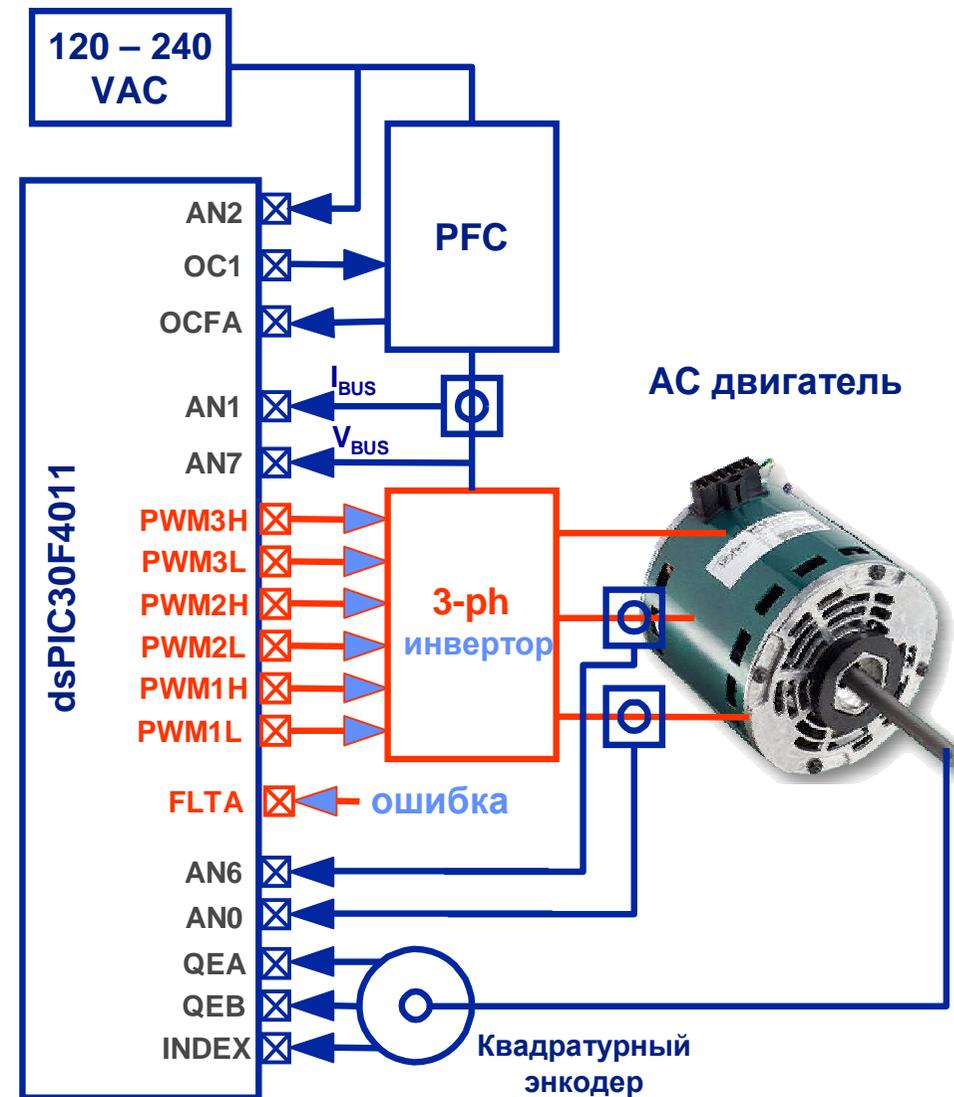
- [ртр](#)

📁 Цель работы

- Конфигурация параллельного порта для работы с модулем ЖКИ
- Конфигурация драйвера ЖКИ
- Вывод символов на ЖКИ

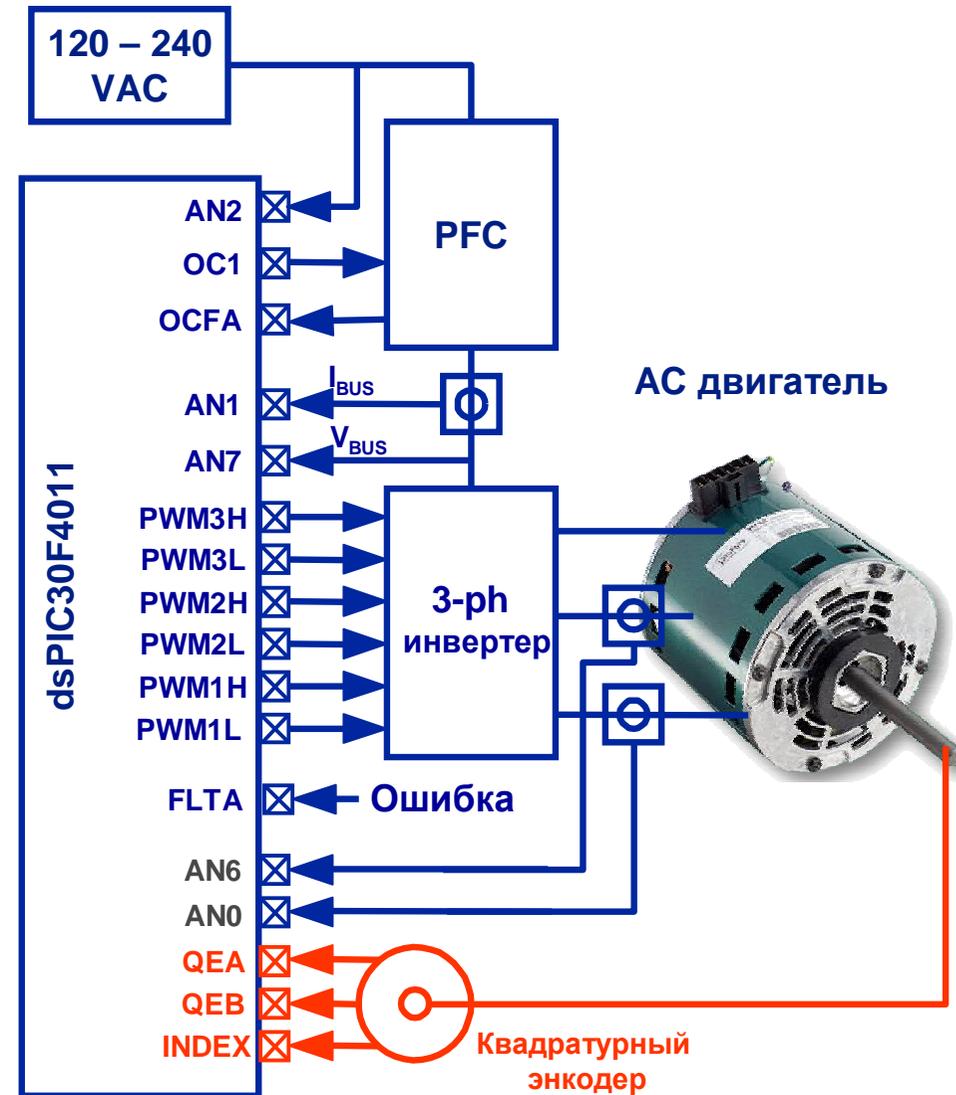
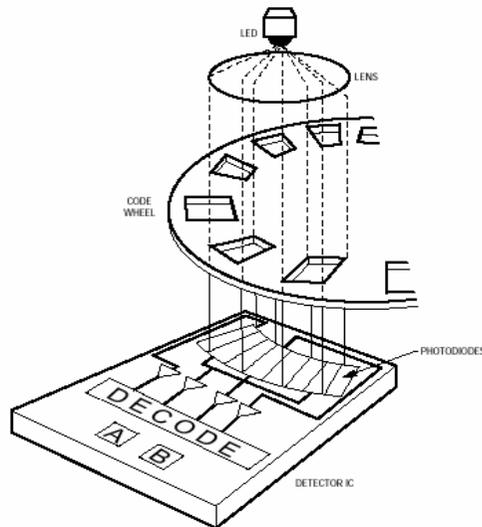
Модули управления двигателями

- ❖ Используется для управления двигателями
- ❖ До четырех генераторов ШИМ
- ❖ Различные настройки позволяют управлять
 - AC двигателями
 - DC двигателями
 - Источниками питания
- ❖ Высокая частота и высокое разрешение ШИМ позволяют реализовывать более сложные алгоритмы управления
- ❖ Аппаратный детектор ошибки



Модуль квадратурного энкодера

- ❖ Отслеживание позиции, скорости и направления вращения
- ❖ Три входа
 - Фаза А
 - Фаза В
 - Сигнал INDEX
- ❖ 16-битный счетчик положения





Модуль Power PWM

Только в dsPIC30F1010, dsPIC30F2020, dsPIC30F2023

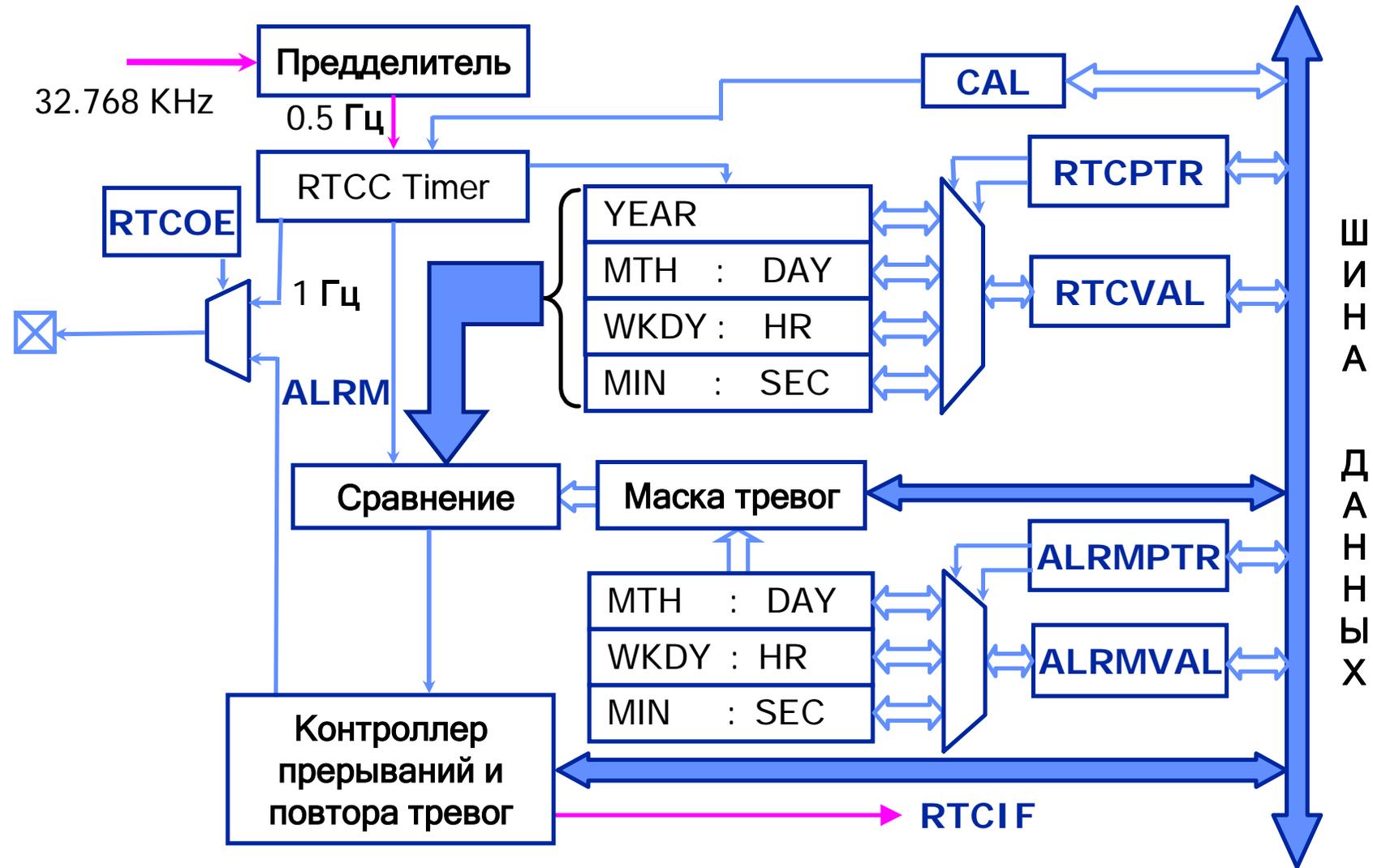
- ❖ **Большая гибкость настроек по сравнению с модулем Motor PWM**
- ❖ **Работает совместно с быстродействующим компаратором SMPS (16 входов)**
- ❖ **Четыре генератора ШИМ**
 - независимая временная база
 - два настраиваемых выхода у каждого генератора
- ❖ **Независимое изменение частоты, коэффициента заполнения, фазы**
- ❖ **Разрешающая способность - 1,1 нс**
 - 10-битный ШИМ с частотой 937 кГц (30 MIPS)
- ❖ **Внутренний стабильный источник опорного напряжения 1,2 В - 1%**
- ❖ **10-битный ЦАП**



Часы реального времени и календарь (RTC)

- ❖ **Только в PIC24F**
- ❖ **Часы**
 - подсчет секунд, минут, часов
 - 24-часовой формат
- ❖ **Календарь**
 - день недели, дата, месяц, год
 - учет високосных лет
- ❖ **BCD формат данных**
- ❖ **Программная и аппаратная калибровка**
- ❖ **Гибкая система тревог, импульс тревоги**
- ❖ **Защита от изменения данных**

Структурная схема модуля RTCC





Лабораторная работа №3

📁 Файл проекта

- `rtc`

📁 Цель работы

- Конфигурация кварцевого генератора 32 кГц
- Конфигурация модуля RTCC
- Инициализация RTCC
- Отображение времени и даты на ЖКИ

UART

- ❖ Полнодуплексный 8- и 9-битный обмен
- ❖ Альтернативные выводы
- ❖ Контроль и передача бита четности
- ❖ Один или два стоповых бита
- ❖ Собственный 16-разрядный генератор (от 29 бит/с до 1,875 Мбит/с @ 16 MIPS)
- ❖ FIFO буферы:
 - Приемник - 4 уровня
 - передатчик - 4 уровня

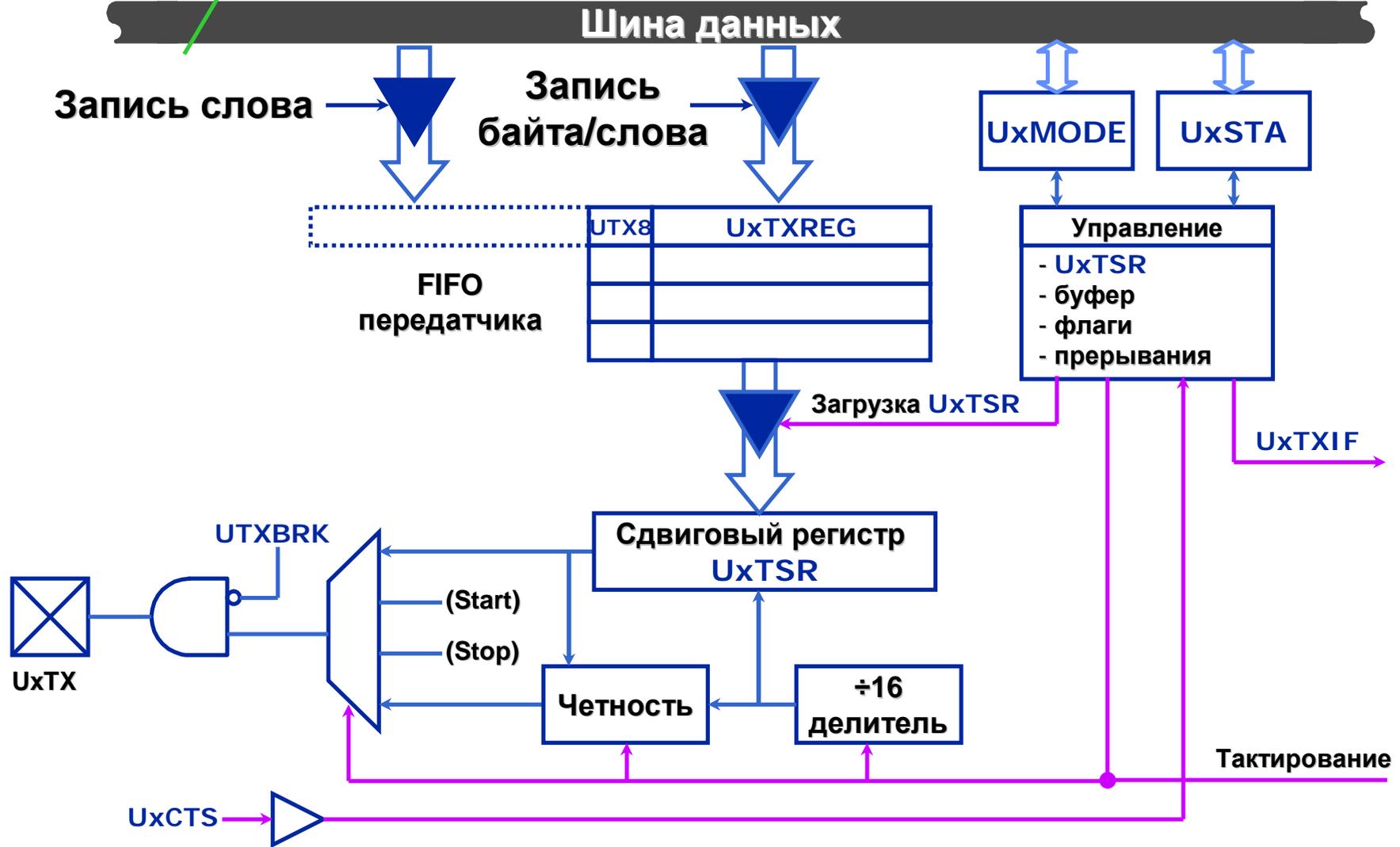
UART

- ❖ **Прерывание при ошибке приема**
 - ошибка четности
 - ошибка кадра
 - переполнение буфера
- ❖ **9-битный обмен с функцией детектирования адреса**
- ❖ **Режим «эхо»**
- ❖ **Полная поддержка спецификации LIN 2.0**
 - Автоопределение скорости обмена (PIC24/dsPIC33)
 - Генерация символа Break
- ❖ **Аппаратное управление потоком - линии CTS и RTS**
- ❖ **Поддержка физического уровня спецификации IrDA**



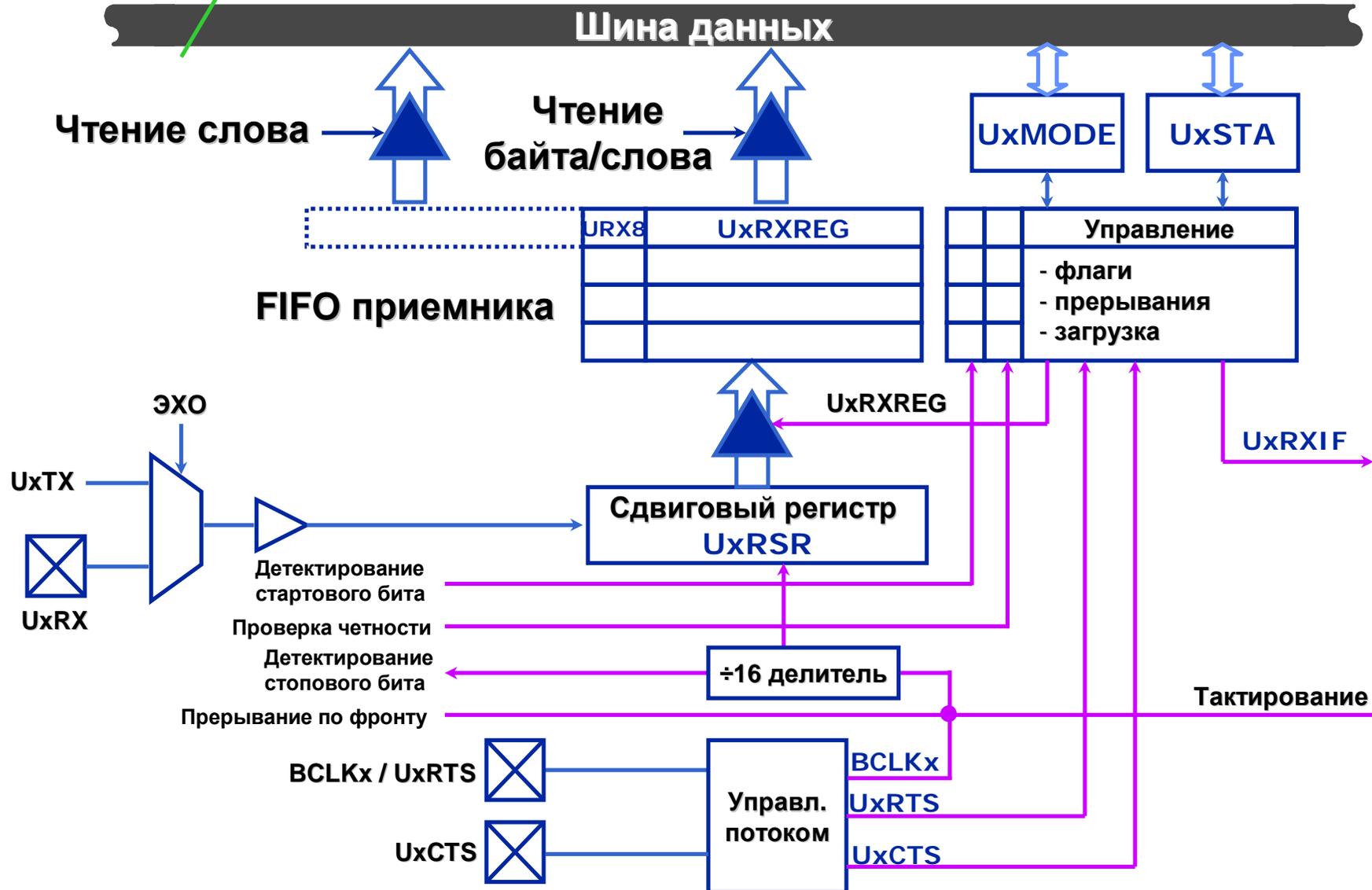
Модуль UART - передача

16



Модуль UART - прием

16





Лабораторная работа №4

🌀 Файл проекта

- `uart`

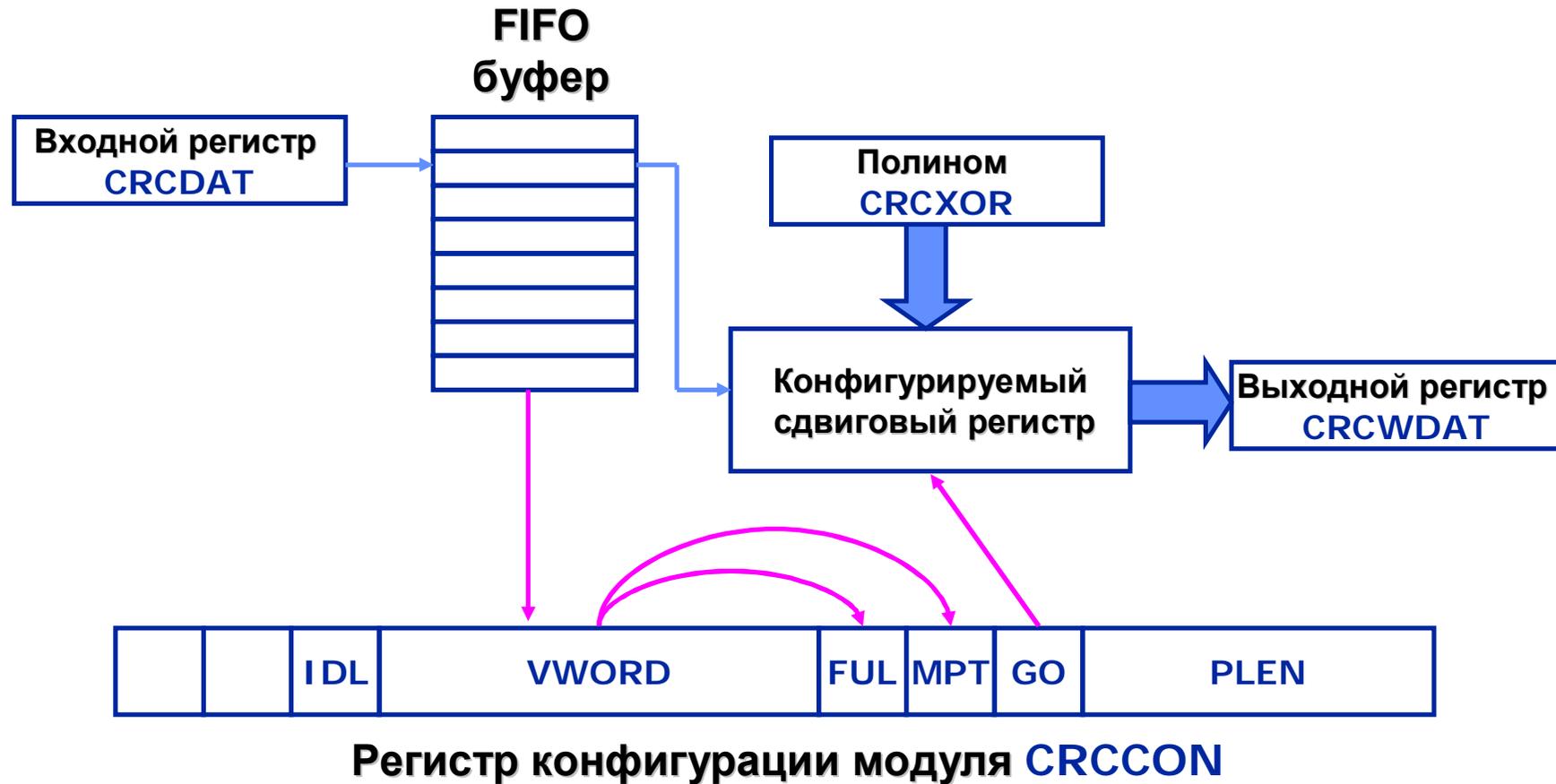
🌀 Цель работы

- Конфигурация модуля UART
- Прием символа
- Отображение принятого символа на ЖКИ
- Передача принятого символа

Модуль CRC

- ❖ **Аппаратное вычисление циклической избыточной суммы**
- ❖ **Произвольный полином от 1 до 16 бит**
- ❖ **FIFO буфер**
 - 8 уровней, если размер полинома больше 8 бит
 - 16 уровней, если размер полинома меньше или равен 8 бит
- ❖ **Прерывание по окончании вычисления**
- ❖ **Время вычисления CRC для очередного слова данных**
 - количество командных тактов, равное размеру полинома

Структурная схема модуля CRC

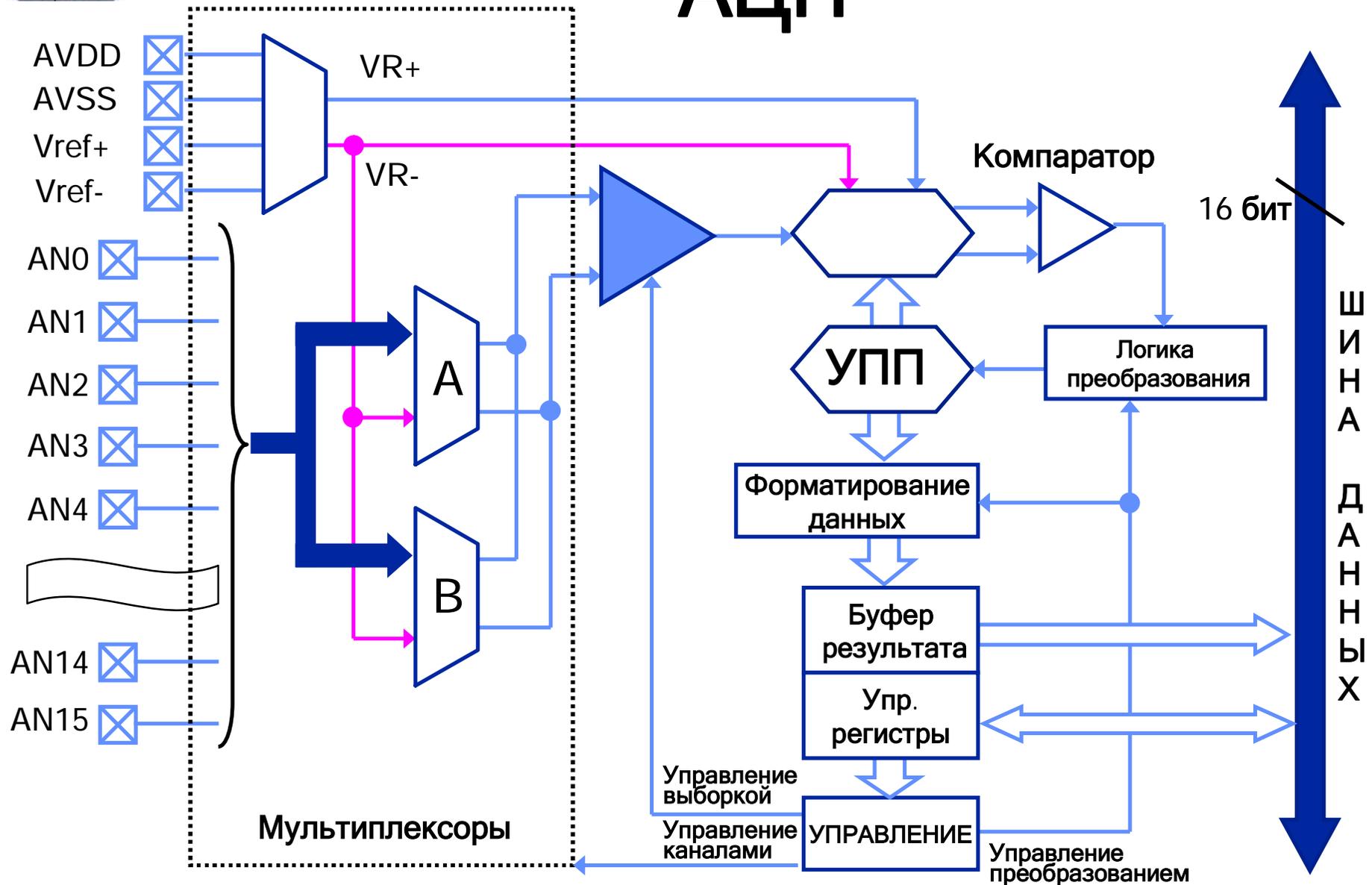


10-битный АЦП

Только в PIC24F

- ❖ До 16 каналов
- ❖ Скорость преобразования - 500 тыс. выборок в сек
- ❖ Гибкий мультиплексор каналов
- ❖ Автоматическая смена каналов
- ❖ Форматирование результата
 - знаковое число
 - беззнаковое число
 - формат Q.15 (с фиксированной точкой)
- ❖ 16-уровневый буфер результатов
- ❖ Настраиваемый порядок преобразования
- ❖ Гибкая конфигурация времени выборки и преобразования
- ❖ Один буфер с дифференциальным входом
- ❖ Программный и аппаратный запуск преобразования

Структурная схема 10-битного АЦП



10-БИТНЫЙ АЦП С 4 УВХ

Только в dsPIC30

- ❖ **Четыре схемы выборки-хранения**
- ❖ **Четыре буфера с дифференциальным входом**
- ❖ **Возможность одновременной выборки**

10/12-битный АЦП

Только в PIC24H / dsPIC33

- ❶ **Разрядность конфигурируется пользователем**
 - 4 УВХ, 10 бит, 1,1 МГц
 - 1 УВХ, 12 бит, 500 кГц
- ❷ **До двух АЦП в одном контроллере**
 - Возможно объединение модулей АЦП для удвоения скорости преобразования
- ❸ **Адресный генератор для канала DMA**



Лабораторная работа №5

📁 Файл проекта

- `adc10`

📁 Цель работы

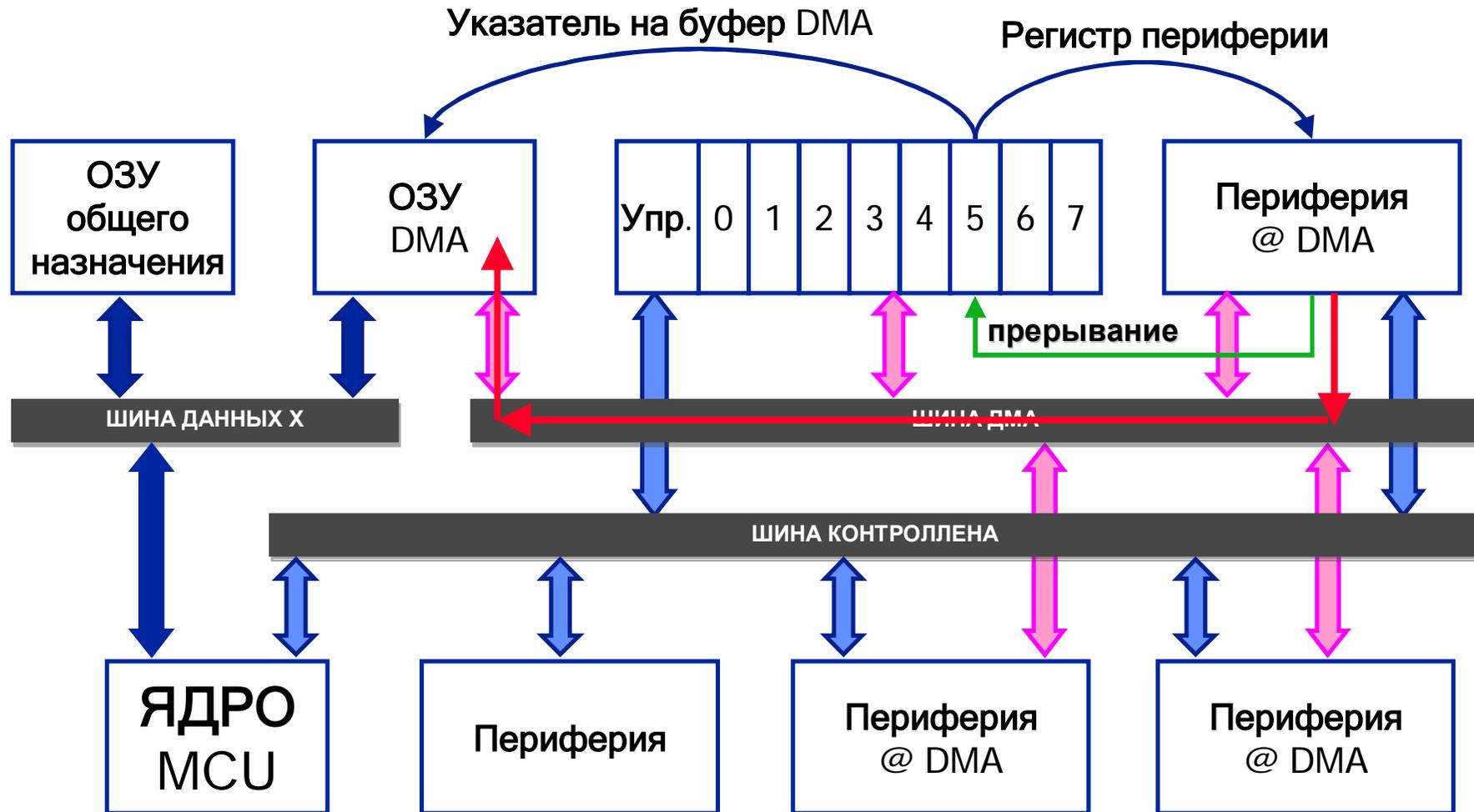
- Конфигурация АЦП в режиме автоматического семплирования двух каналов
- Вывод результатов преобразования на ЖКИ

DMA

Только в PIC24H / dsPIC33

- ⊗ Является частью ядра
- ⊗ Назначение - аппаратный обмен данными между памятью и периферией
- ⊗ 8 идентичных однонаправленных каналов
- ⊗ Обслуживание модулей
 - модули захвата
 - модули сравнения
 - SPI, UART, DCI, ADC, CAN
- ⊗ 2 кБ двухпортовой памяти SRAM
- ⊗ Одновременный доступ DMA и ядра контроллера к буферу DMA
- ⊗ 8- или 16-битный обмен
- ⊗ Возможность программного запуска транзакции

Структурная схема DMA

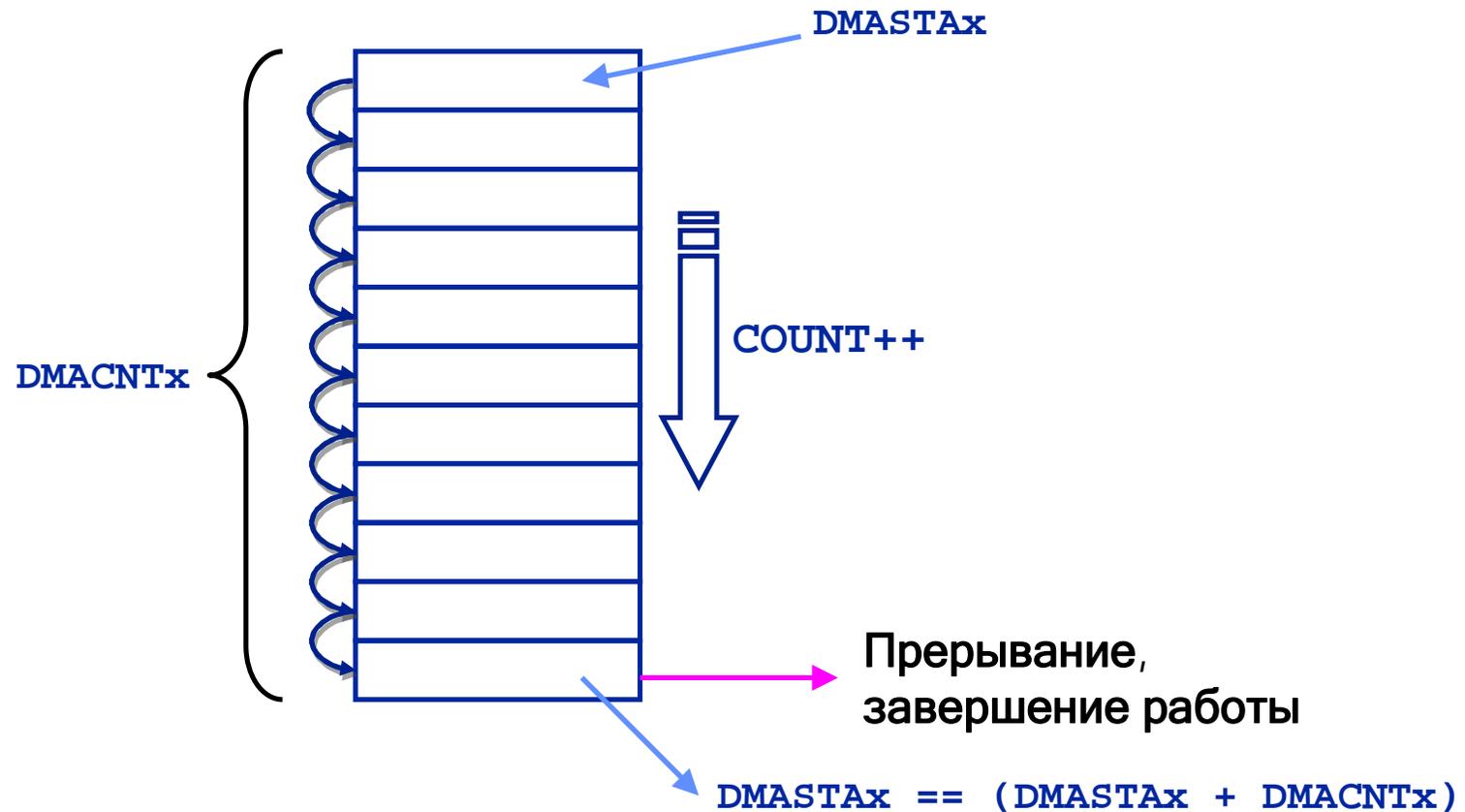


Режимы работы DMA

- ❶ **Однократная передача**
 - канал DMA необходимо переконфигурировать для продолжения обмена
- ❷ **Автоматический повтор**
 - возврат указателя на начало буфера DMA
- ❸ **Автоматический повтор с половинным заполнением**
 - прерывание возникает при половинном заполнении буфера
- ❹ **Режим «ring-pong»**
 - используется два буфера
- ❺ **Режим косвенной адресации периферии**
 - периферийный модуль формирует смещение для указателя на буфер DMA
- ❻ **Режим дополнения**
 - Автоматическая передача после приема (для SPI)

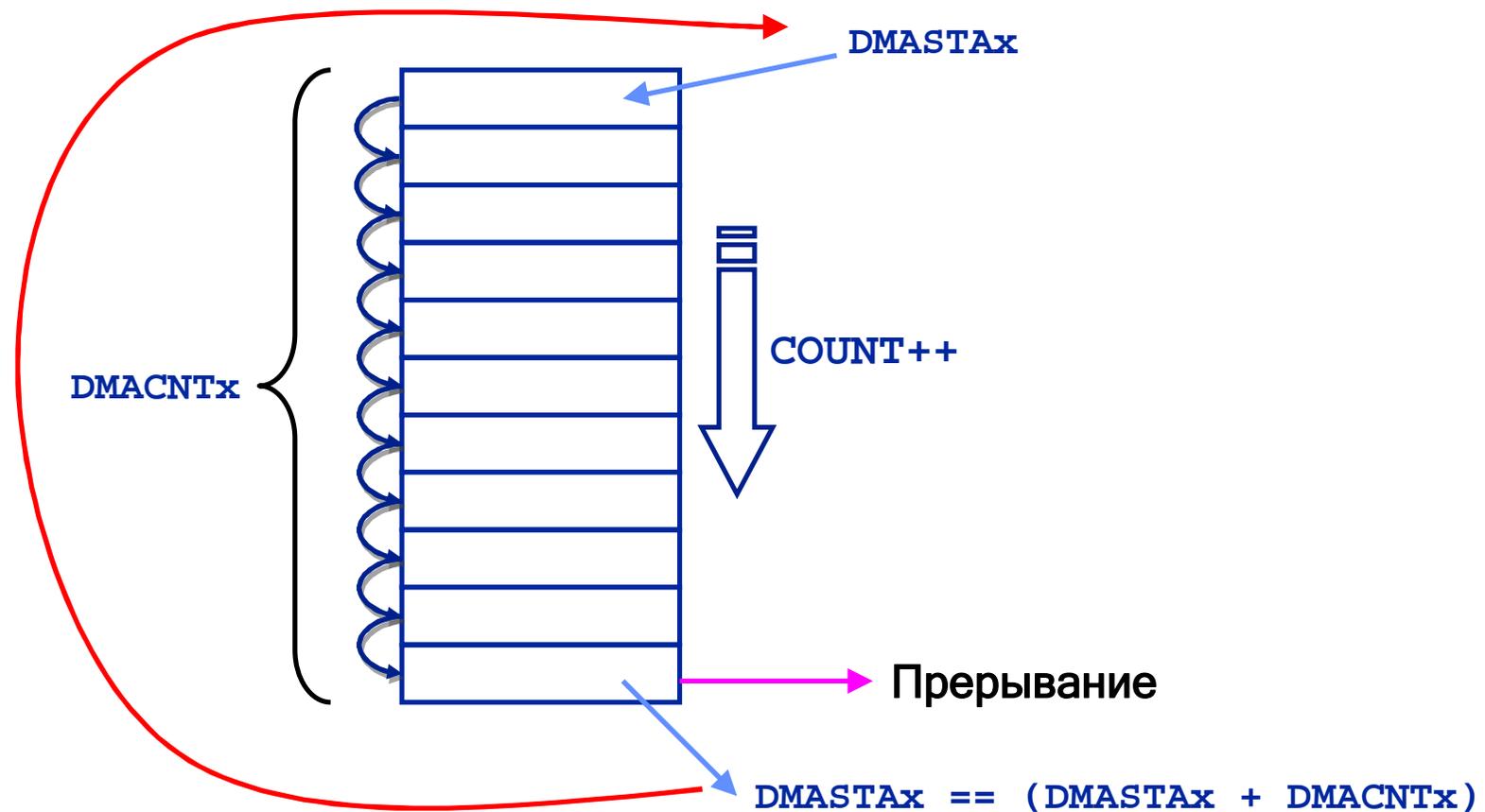
Режимы работы DMA

Однократная передача



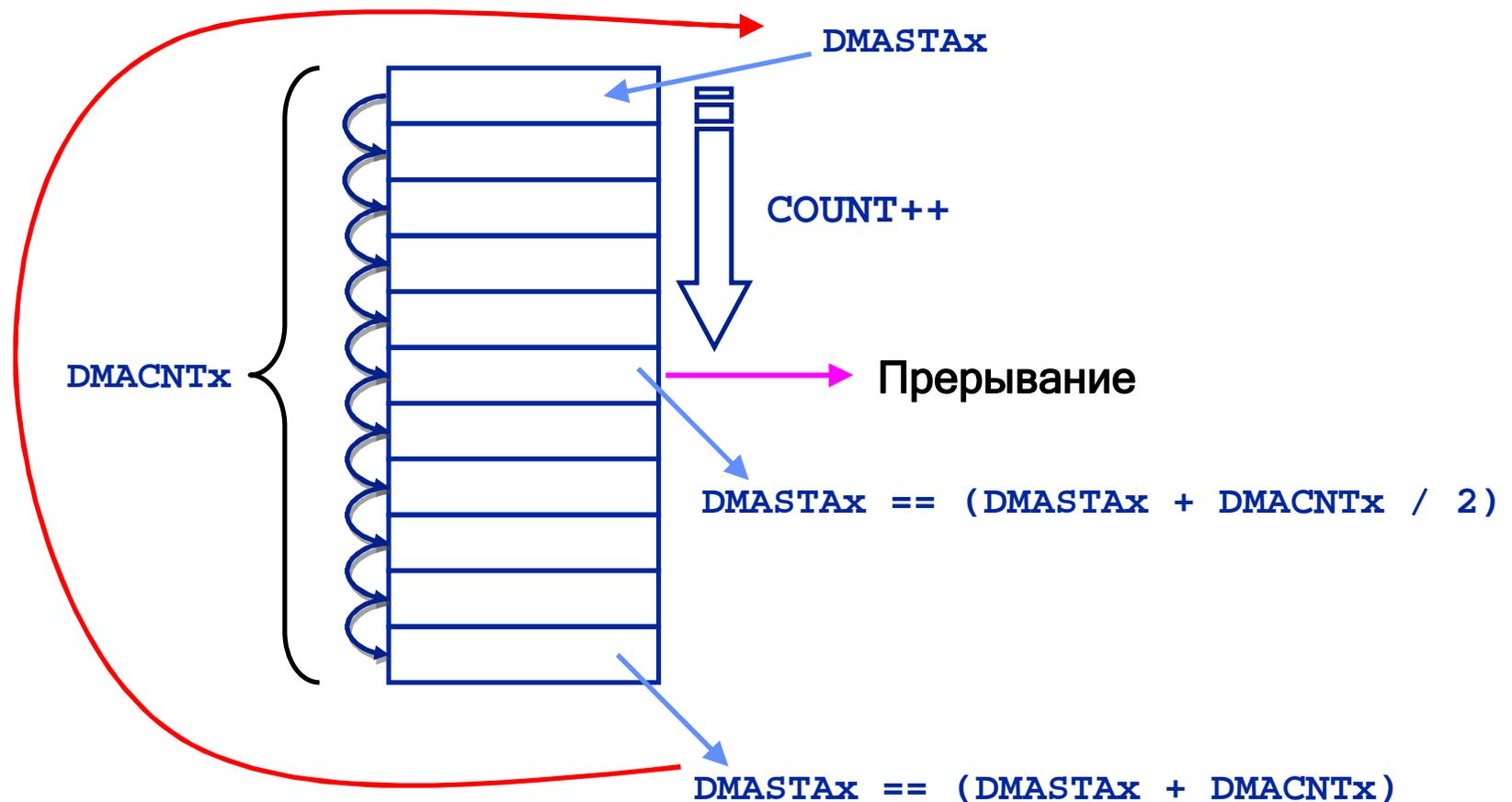
Режимы работы DMA

Автоматический повтор



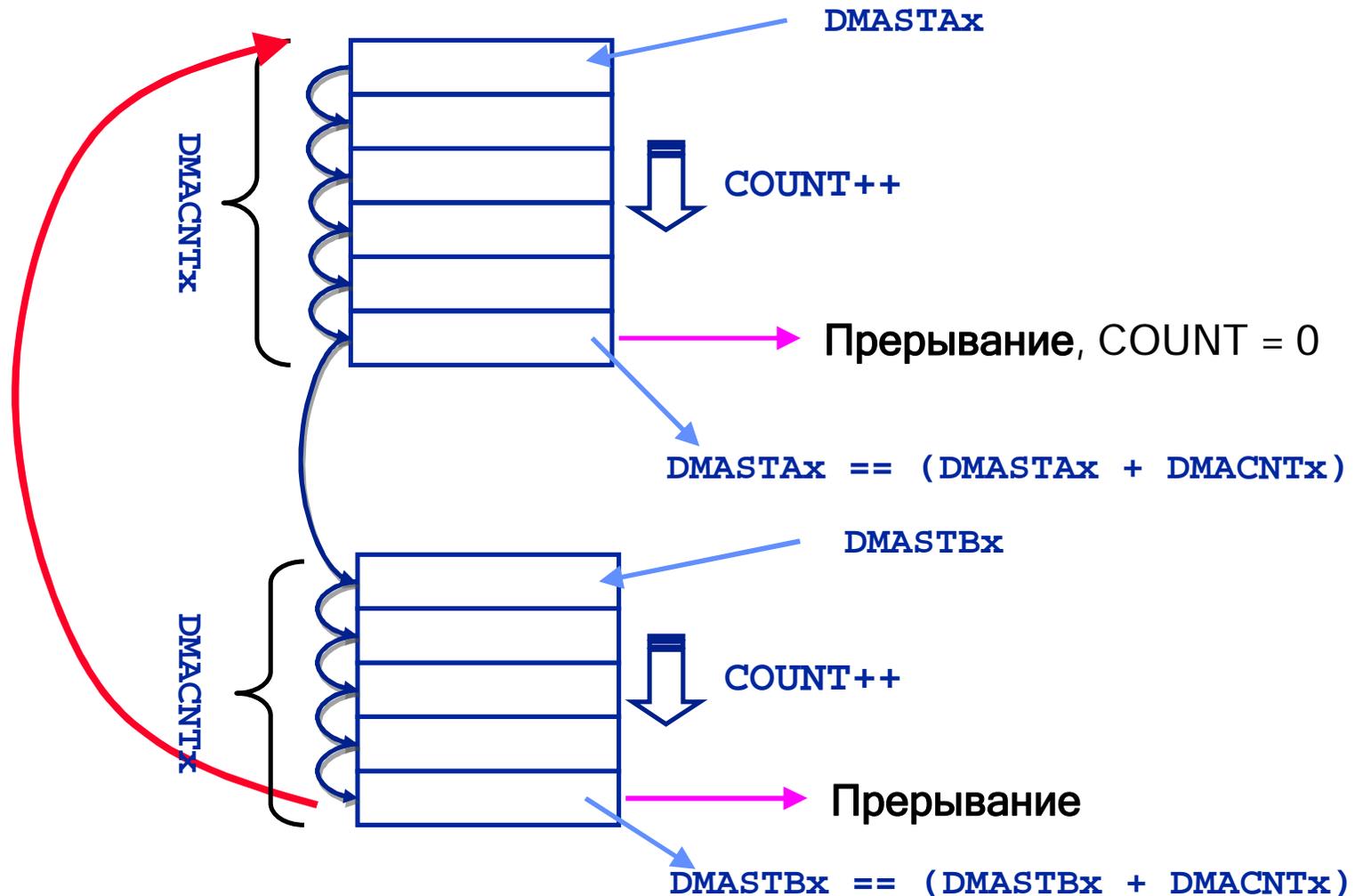
Режимы работы DMA

Автоматический повтор с половинным заполнением



Режимы работы DMA

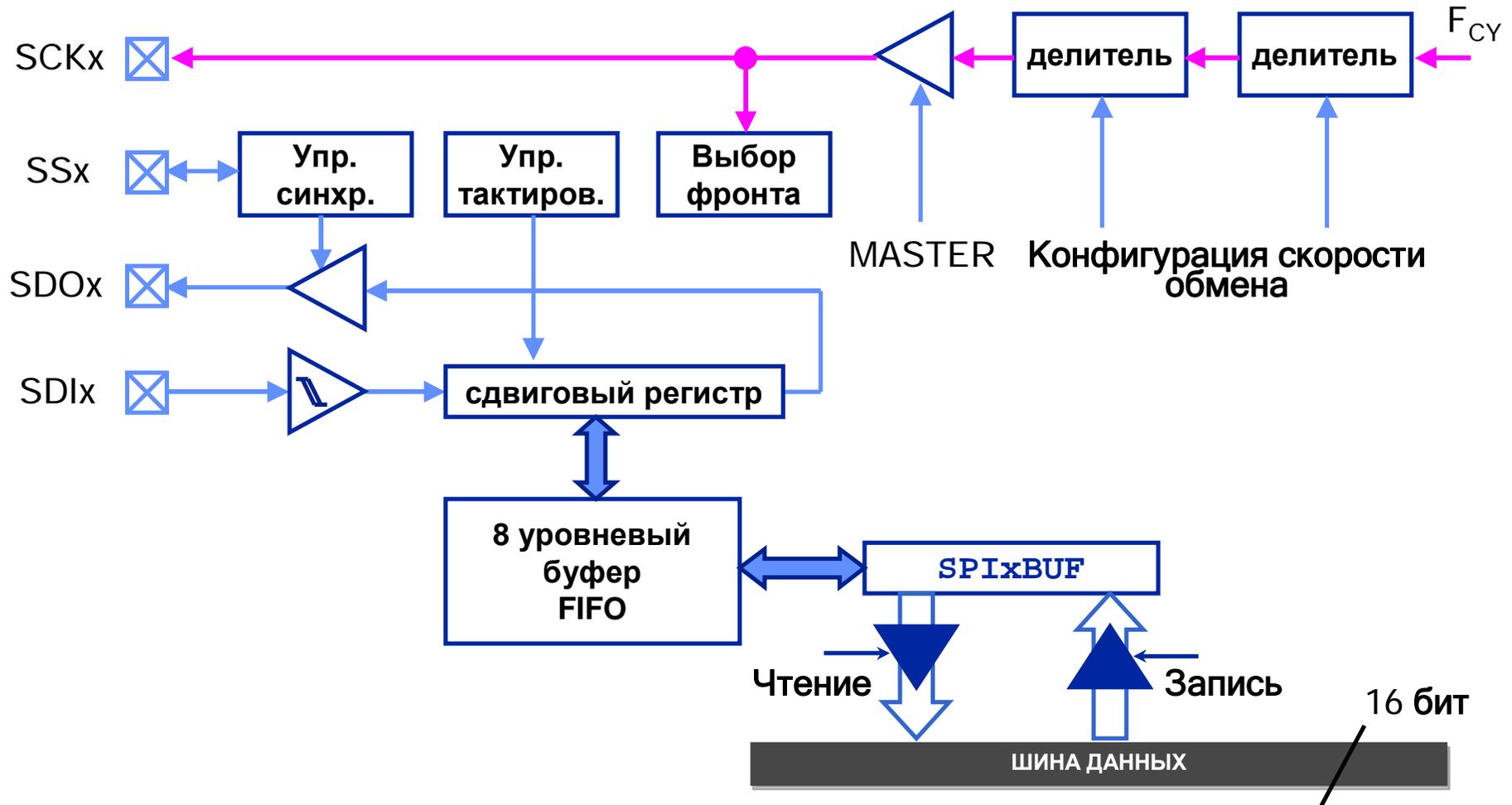
Режим Ping-Pong



Модуль SPI

- ❶ Физически независимый модуль
- ❷ 8 и 16-битный режим обмена
- ❸ Режимы работы
 - ведомый
 - ведущий
 - ❶ генератор тактового сигнала SCK с двумя делителями. $F_{max} = 10 \text{ МГц}$
- ❹ 8-уровневый буфер FIFO
- ❺ Фреймовый режим обмена
 - подключение современных кодеков и АЦП
- ❻ Выбор активных уровней на линиях

Структурная схема SPI





Лабораторная работа №6

🔗 Файл проекта

- `spi_dma`

🔗 Цель работы

- Сконфигурировать модуль SPI
- Сконфигурировать каналы DMA на прием и передачу
- Записать буфер данных в SPI EEPROM, установленную на плате
- Считать буфер данных из SPI EEPROM
- Сравнить записанный и считанный буферы



**Спасибо за
внимание!**