



P32

Знакомство с PIC32

Начало работы
Отладка контроллеров

Class Objectives

- | **После изучения этого класса вы будете знать:**
 - | Архитектуру контроллеров PIC32
 - | Основные особенности периферии контроллеров PIC32
 - | Доступные аппаратные и программные решения, предлагаемые Microchip

План

- | **Обзор PIC32**
- | **Архитектура**
- | **Периферия**
- | **Аппаратные и Программные средства**

- | **Несколько демонстраций**

План

- | **Обзор PIC32**
- | **Архитектура**
- | **Периферия**
- | **Аппаратные и Программные средства**

Решения Microchip

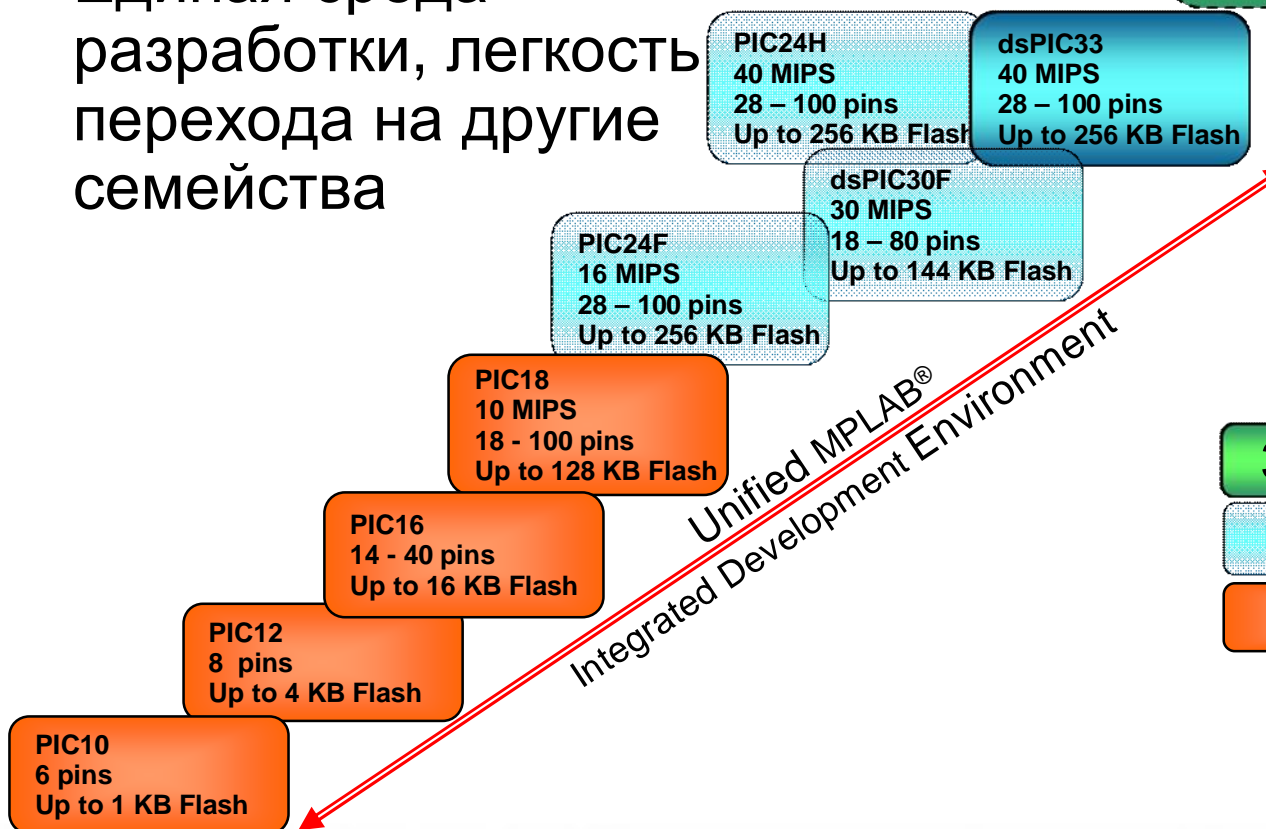
>550 PIC[®]

контроллеров

Единая среда
разработки, легкость
перехода на другие
семейства



Performance



Unified MPLAB[®]
Integrated Development Environment

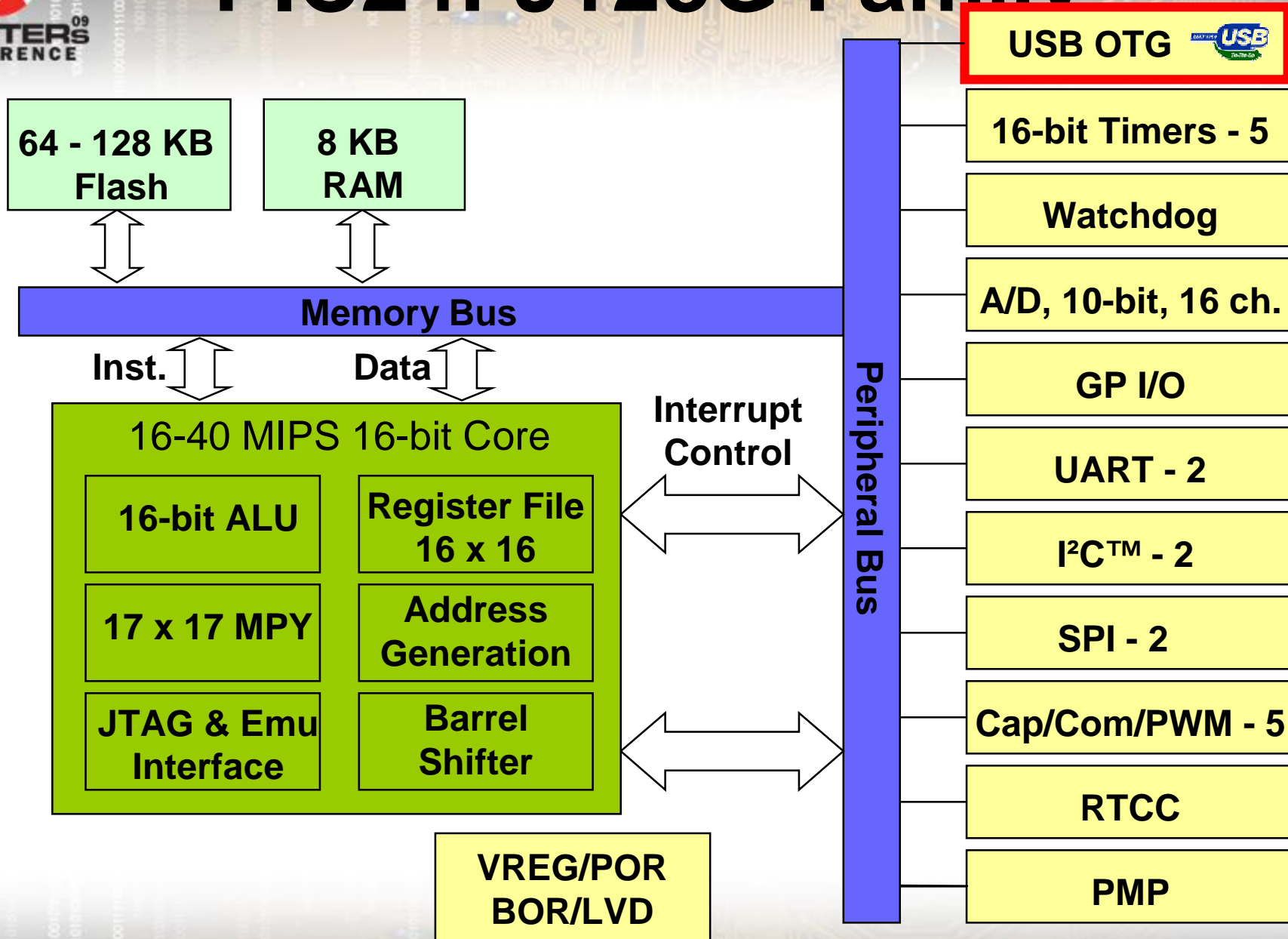
32-bit

16-bit

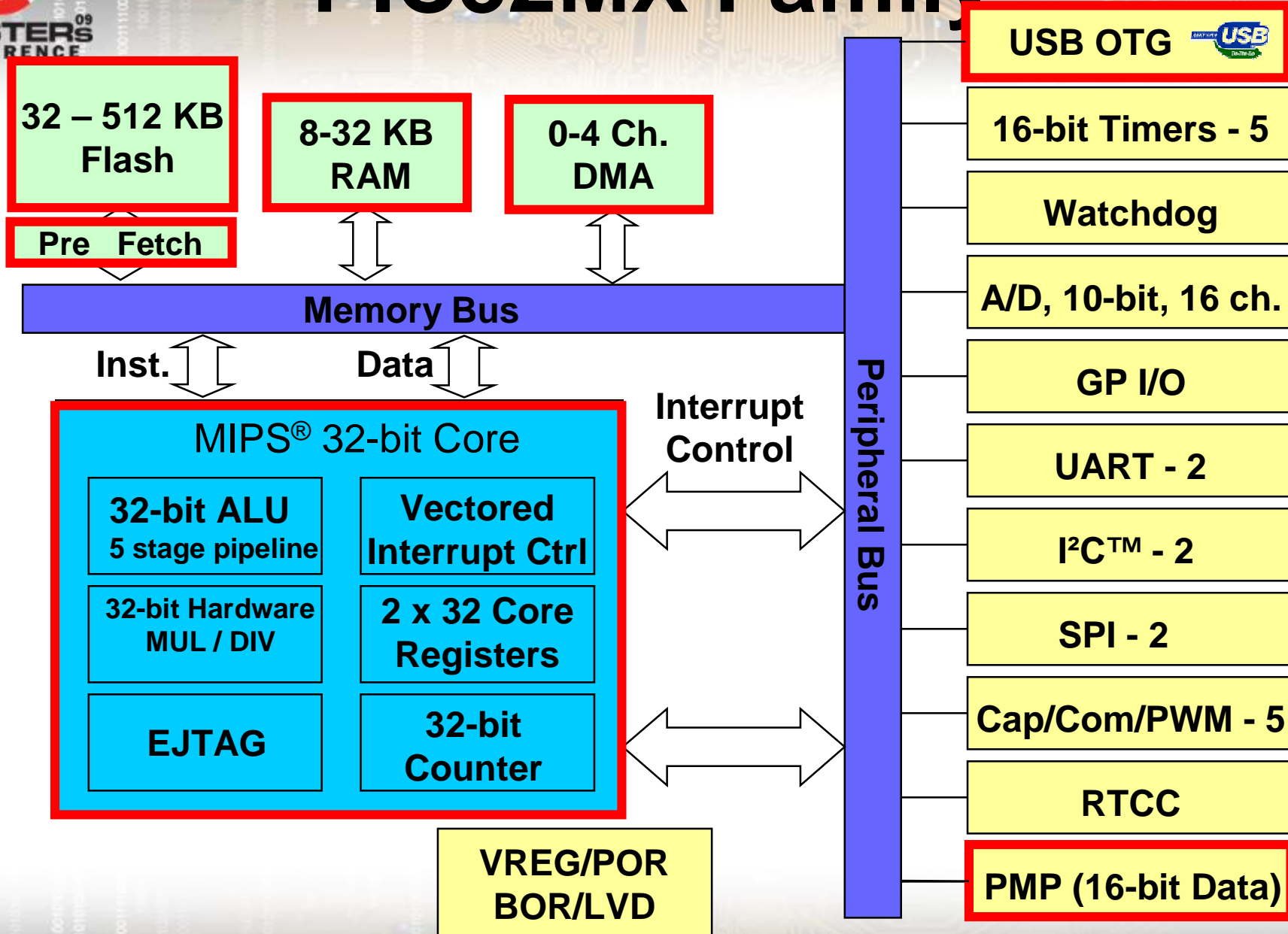
8-bit

Migration

PIC24FJ128G Family



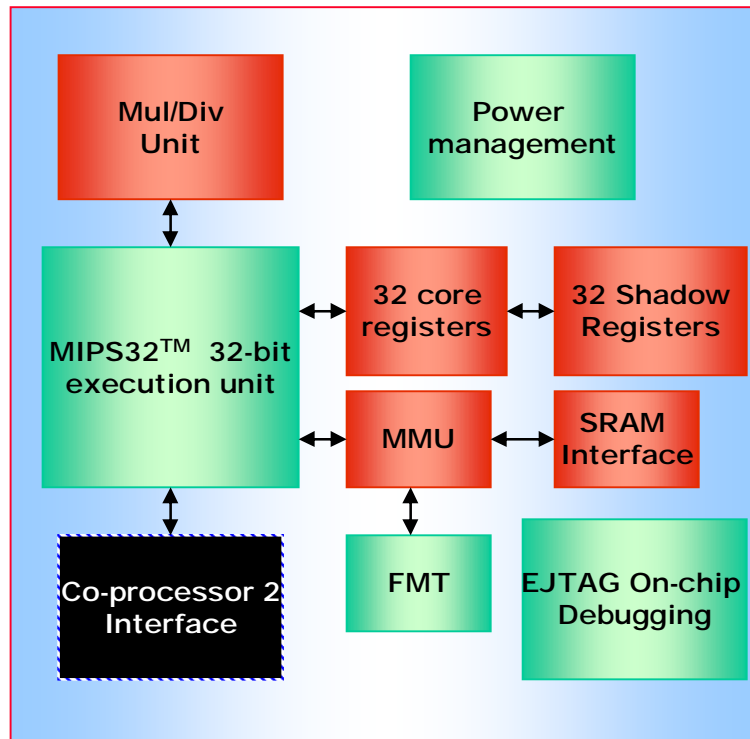
PIC32MX Family





MIPS TECHNOLOGIES

MIPS32[®] M4K[®] Core



**Высокая
производительность**

Низкое потребление

**Компактный размер
кода**

**Поддержка сторонними
производителями**

Особенности контроллеров

Совместимость с PIC24 и dsPIC33

- По: выводам, Периферии, Библиотекам

Высокая производительность

- Ядро MIPS32[®], 80 МГц
- До 4 каналов DMA

Много памяти

- 512 + 12 KB Flash, 32 KB RAM
- Буфер предвыборки с кэшем

Быстрые прерывания

- Аппаратный контроллер прерываний
- Полный набор из 32 теневых регистров



Семейство PIC32

3 – общего применения

4 – с USB OTG

Семейство

Объем Flash в кБ

PIC32MX360F512L

20 – без DMA

40 – с DMA

60 – с DMA и Trace

H – 64 вывода

L – 100 выводов



Multimedia Demo (Demo 1)

- | **Purpose**

- | Performance and Capabilities

- | **Hardware**

- | PIC32 Multimedia Board

- | **Software**

- | Graphics Demo App by SEGGER

Multimedia Demo



- | **65K цветов, QVGA дисплей**
- | **USB-питание/зарядка, Li-ion аккумулятор**
- | **Интегрированный отладчик**
- | **USB OTG, Device и Host**
- | **Микрофон, динамик и microSD карта памяти**

Multimedia Demo



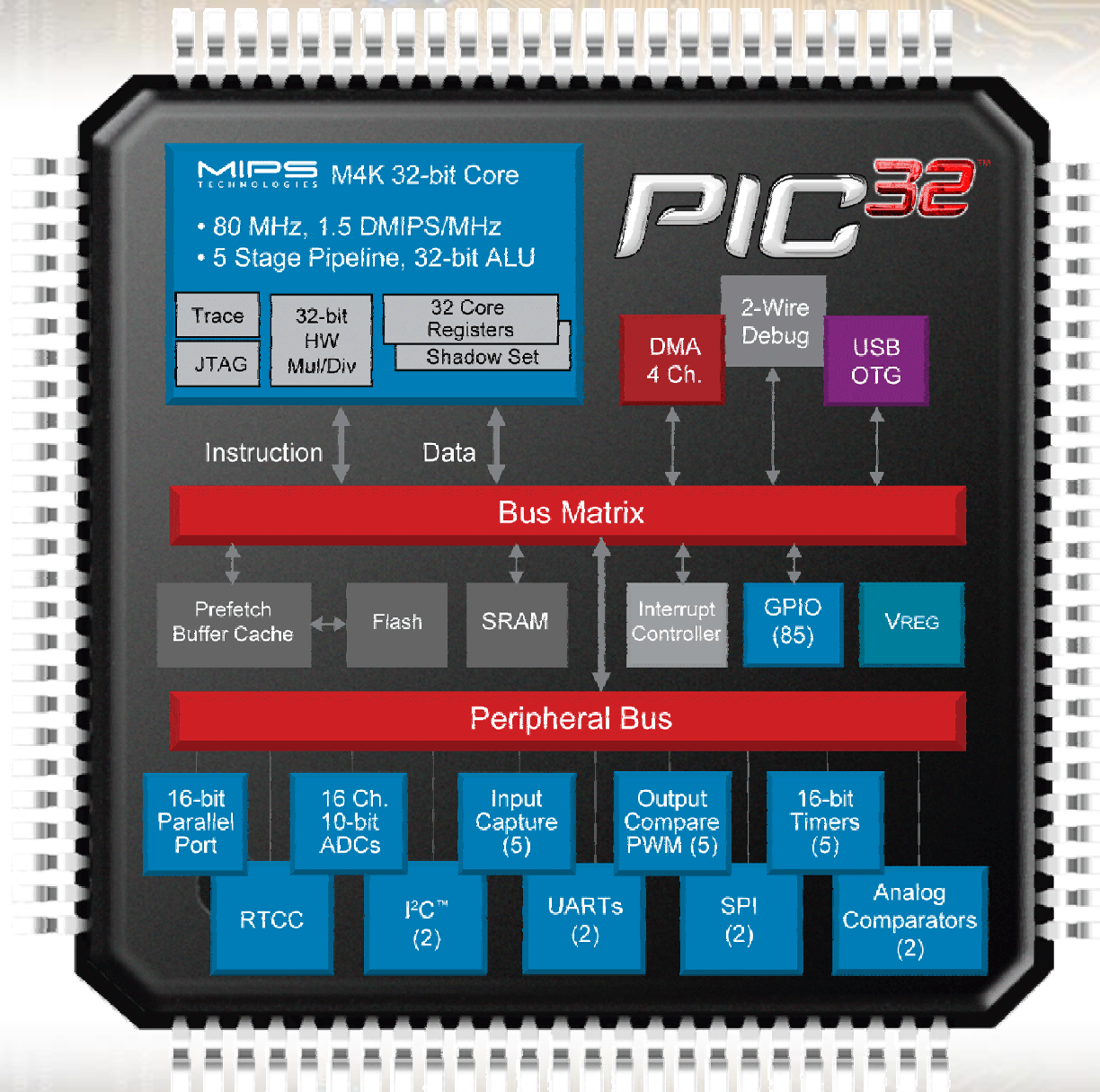
Вопрос - I

- | PIC32 имеют до 512 / 524 КБ Flash памяти
- | Максимальная частота 80 МГц

План

- | Обзор PIC32
- | **Архитектура**
- | Периферия
- | **Аппаратные и Программные средства**

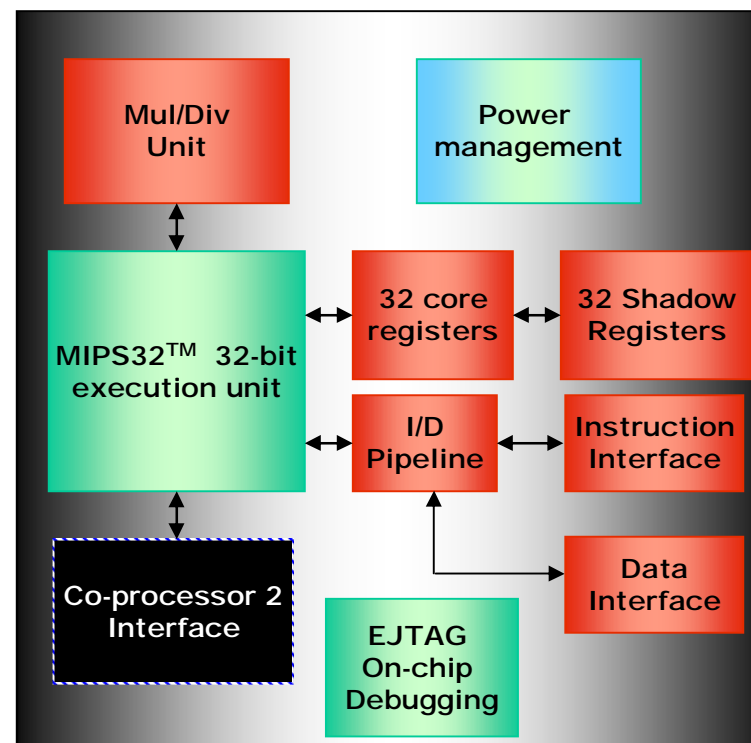
Блок диаграмма



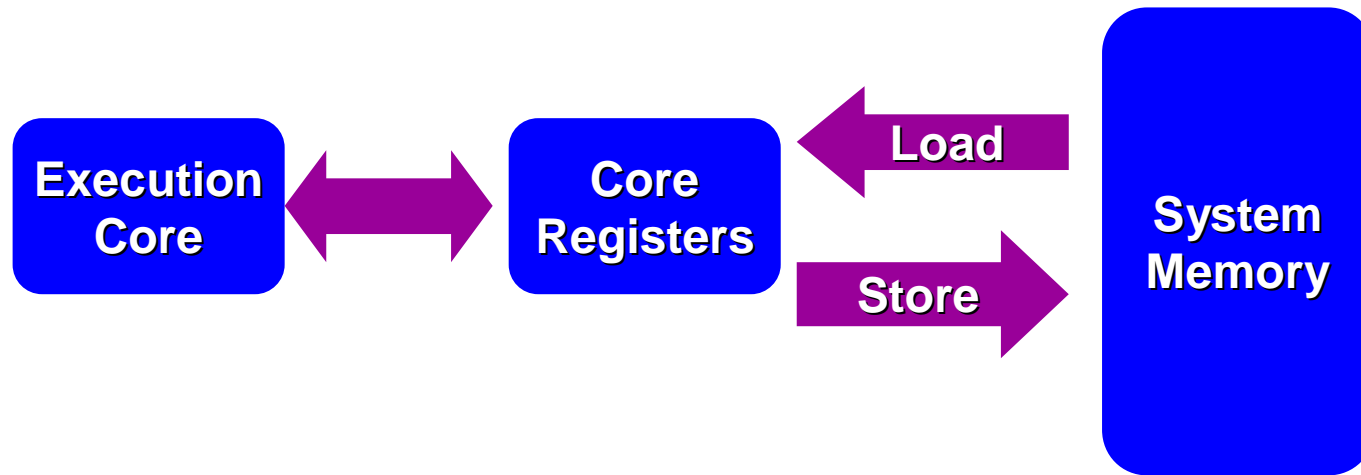
Свойства ядра

- | **Быстродействие 1.56 DMIPS/МГц**
- | **80 МГц 32-бит ЦПУ**
 - | Одноцикловые команды MDU/MAC
 - | Автономный блок MDU
- | **32 регистров ядра**
 - | В два раза больше чем у конкурентов
 - | Плюс 32 теневых регистра
- | **Две шины Инструкций и Данных**
 - | Возможно выполнение кода из RAM
- | **Поддержка MIPS16e™ инструкций**
 - | Уменьшает размер кода на ~40%

MIPS32® M4K® Core

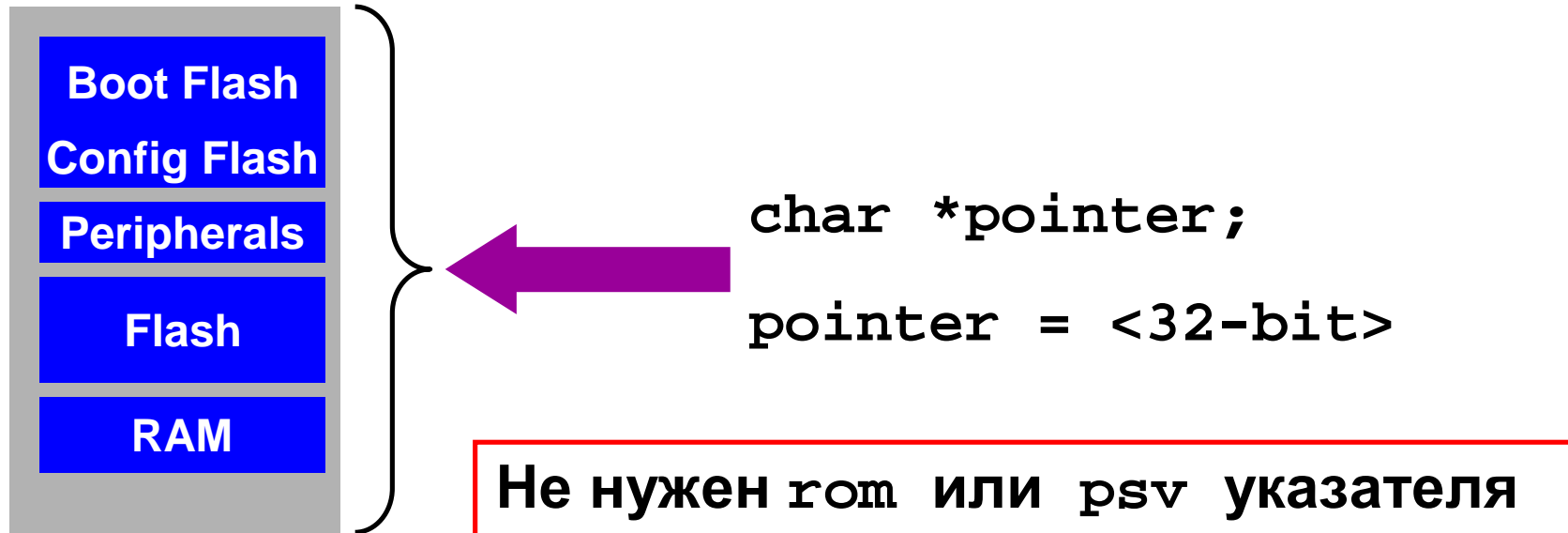


Архитектура Load/Store



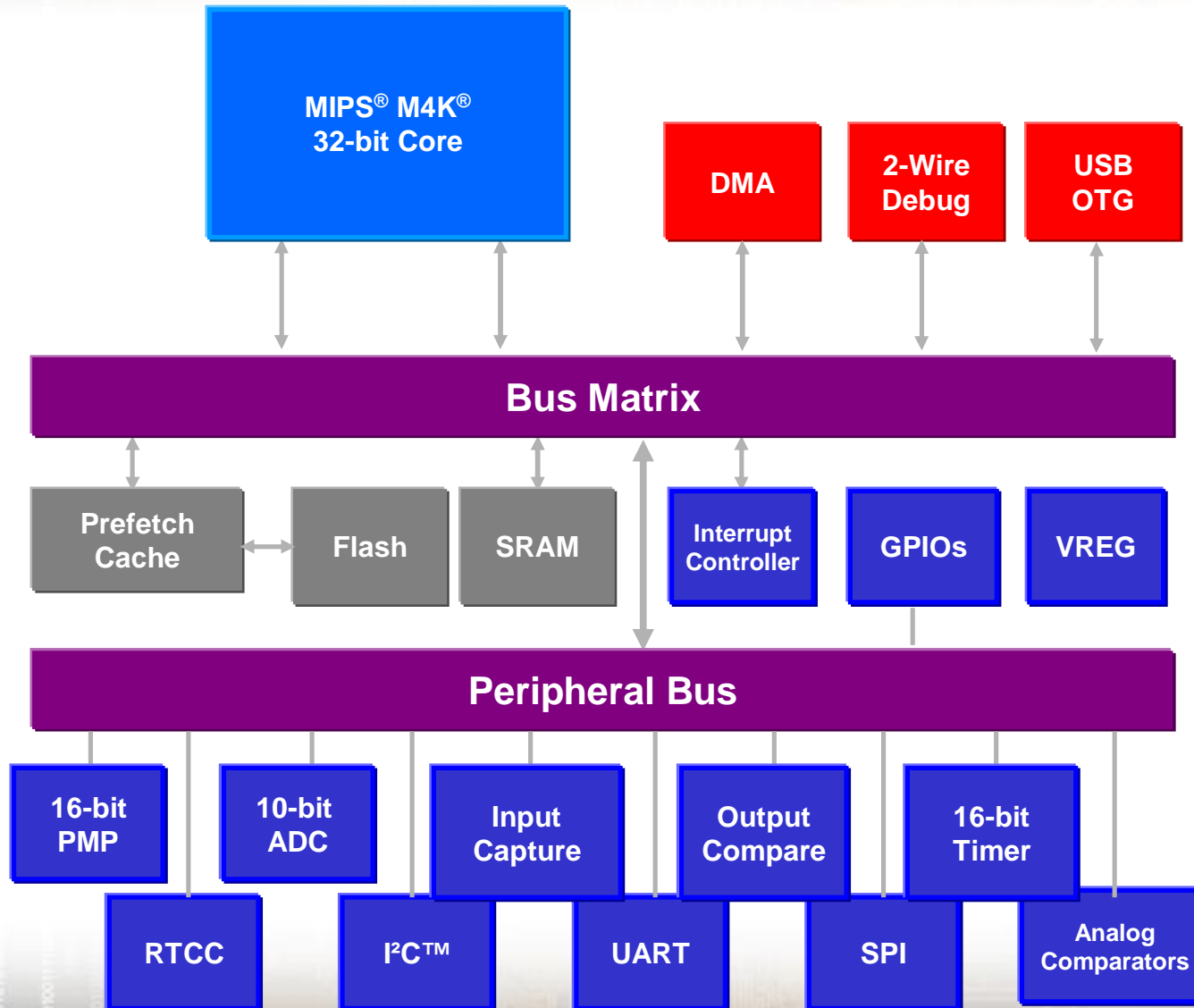
- | 1 инструкция за клок
- | ЦПУ может работать быстрее чем память

Унифицированное адресное пространство

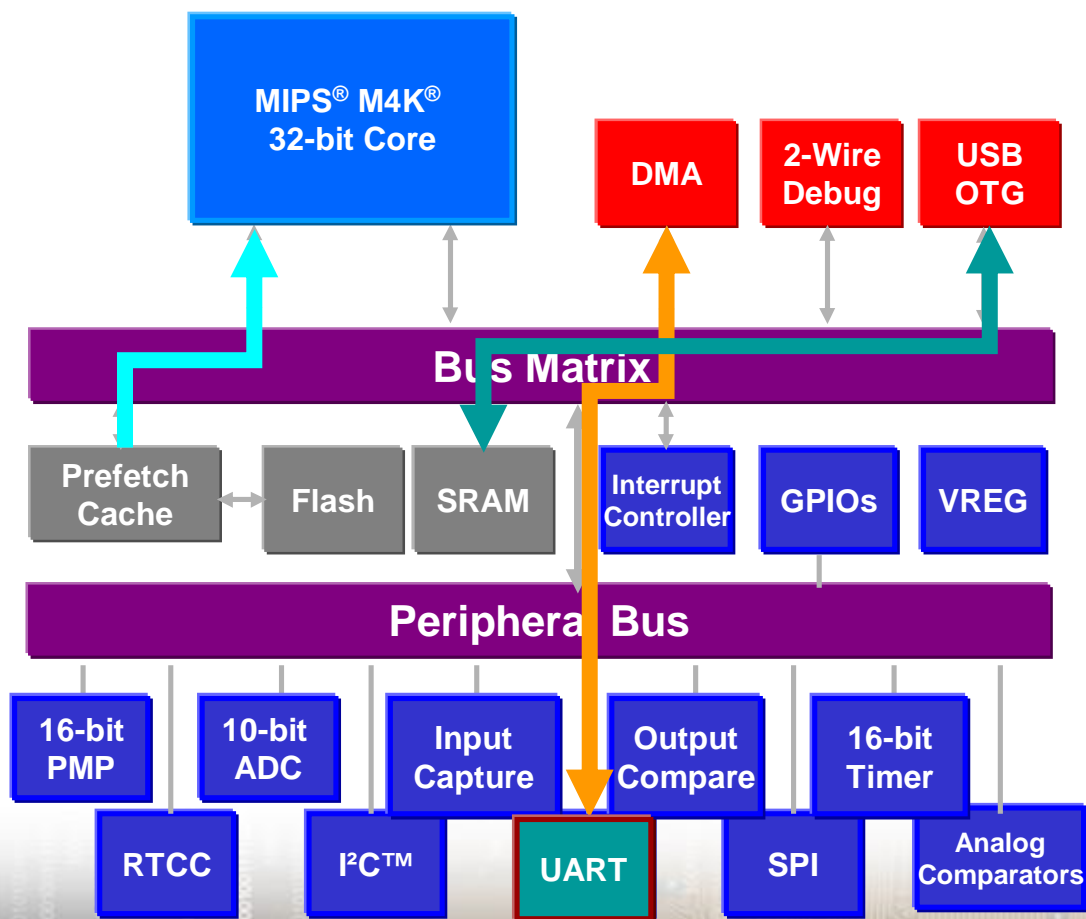


Аппаратная защита от некорректного обращения

Шинная матрица



Высокопроизводительная шинная матрица



- Высокопроизводительная шинная матрица
 - Ядро ЦПУ
 - DMA
 - Модуль отладки
 - USB OTG
- Одновременный обмен данными между ведущими и ведомыми модулями

Предвыборка команд и Кэш

- | захватывает 128-bit данные

 - | 4x32-bit инструкций

 - | 8x16-bit инструкций

- | До 80 MIPS при последовательном выполнении

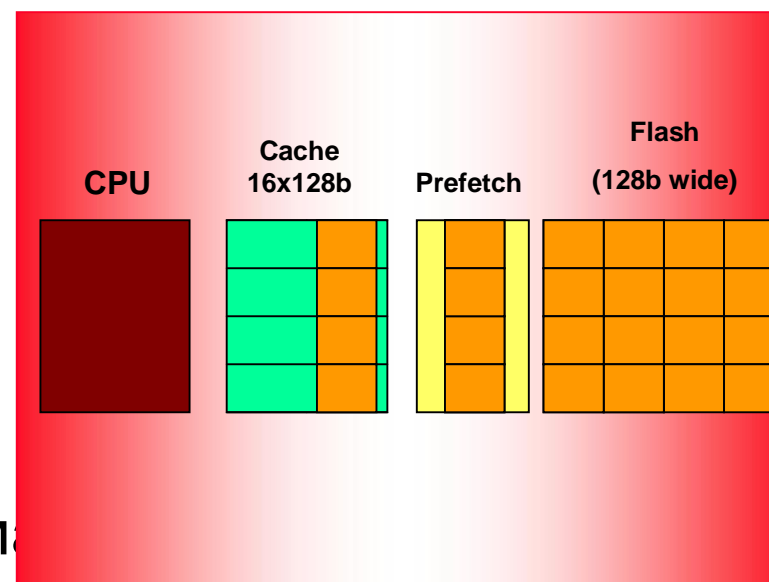
- | 256 Байт кэша

 - | 16 строк

 - | по 4 инструкции в строке

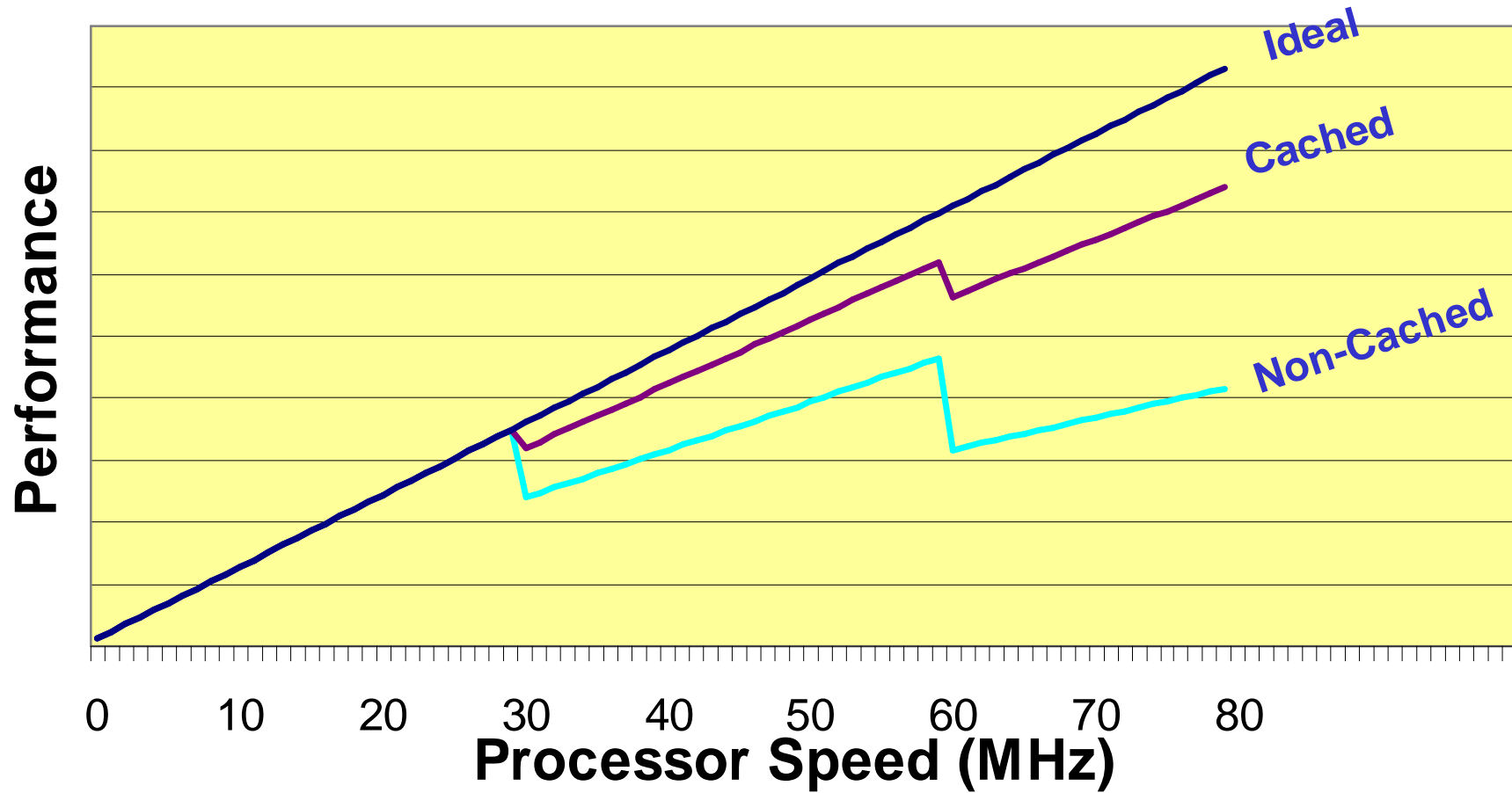
 - | 2 строки имеют адресную м

 - | 4 строки можно использовать для кэширования данных





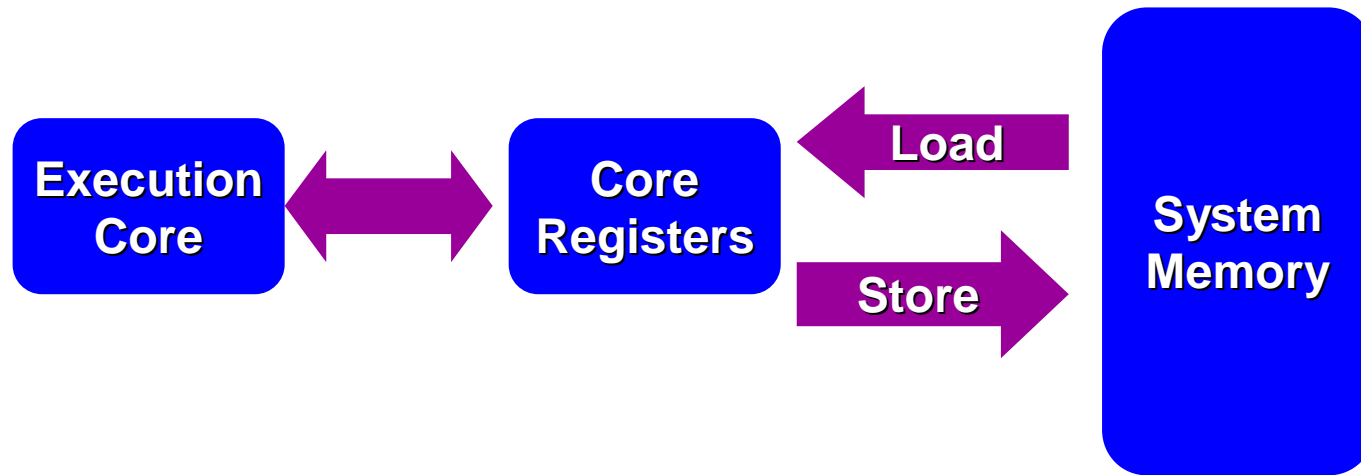
Effect of Prefetch Cache



Prefetch Cache Demo

```
Dhrystone - HyperTerminal
File Edit View Call Transfer Help
Dhrystone Benchmark, Version 2.1 (Language: C)
Board: Microchip Explorer16
Microcontroller: PIC32MX360F
Running at: 80 MHz
Runs through Dhrystone: 10000
-----
Running with Cache and PF disabled:
CPU Cycles/Dhrystone Loop = 996
DMIPS/MHz (x1000) = 571
Time (ms) = 124
-----
Running with Cache and PF enabled:
CPU Cycles/Dhrystone Loop = 478
DMIPS/MHz (x1000) = 1190
Time (ms) = 59
-----
End of Benchmark
Connected 0:00:36  Auto detect  19200 8-N-1  SCROLL  CAPS  NUM  Capture  Print echo
```

Архитектура Load/Store



- | 1 инструкция за клок
- | ЦПУ может работать быстрее чем память

Atomic Bit Operations

LATA

RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
-----	-----	-----	-----	-----	-----	-----	-----

LATA SET

-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---

LATA CLR

-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---

LATA INVERT

-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---

Одноцикловая манипуляция с битами!



Atomic Clear Operation

LATA

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

LATA CLR

1	0	1	0	1	0	1	0
---	---	---	---	---	---	---	---



Atomic Set Operation

LATA

0	1	0	1	0	1	0	1
---	---	---	---	---	---	---	---

LATA SET

0	0	0	0	1	0	1	0
---	---	---	---	---	---	---	---

Atomic Invert Operation

LATA

1	0	1	0	1	1	1	1
---	---	---	---	---	---	---	---

LATA INVERT

1	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---

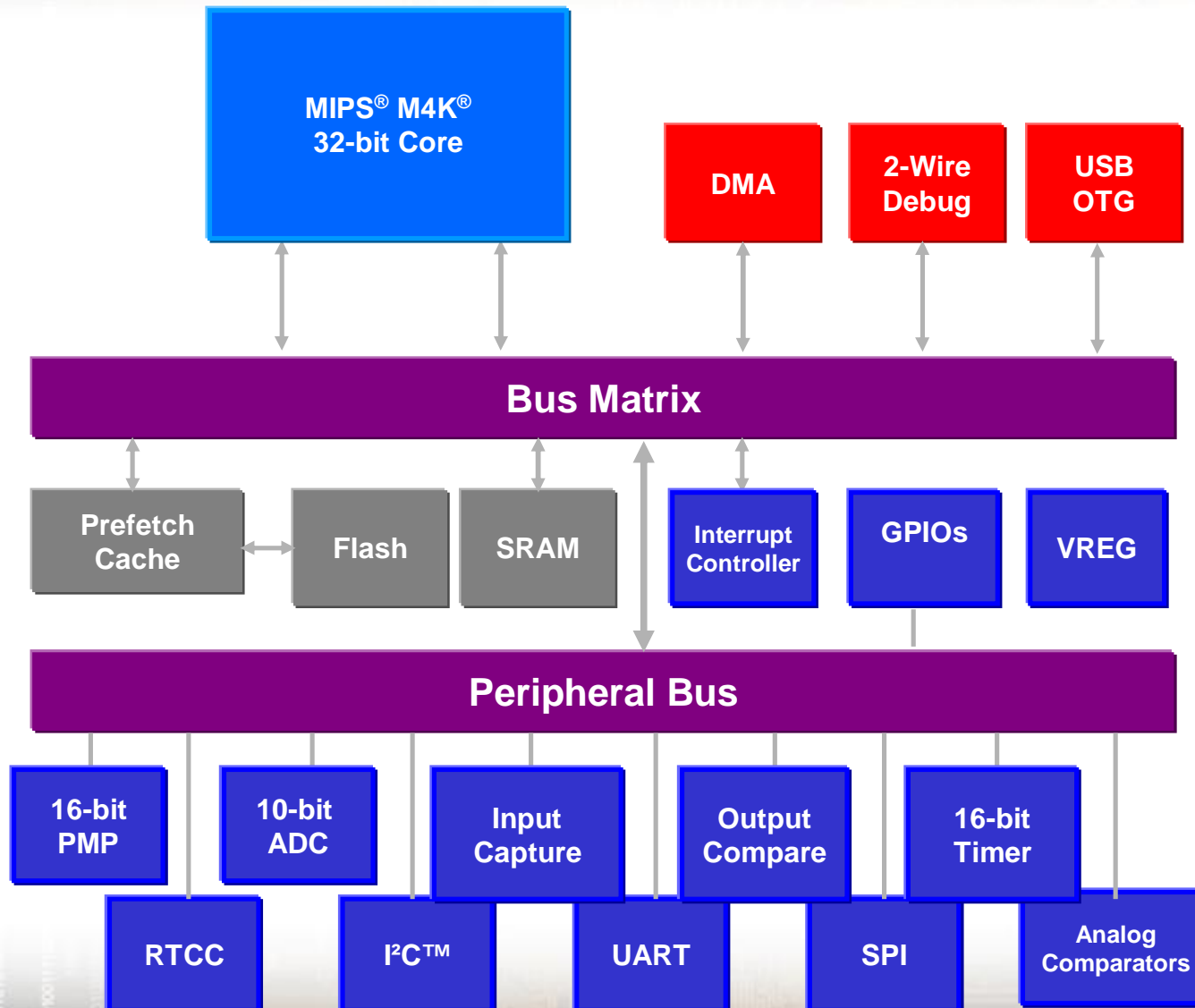
Вопрос - II

- | PIC32 имеет 2 набора регистров
- | Режим MIPS16e™ улучшает Размер кода
- | PIC32 может захватить 8 команд за одно обращение к Flash памяти

План

- | Обзор PIC32
- | Архитектура
- | **Периферия**
- | Аппаратные и Программные средства

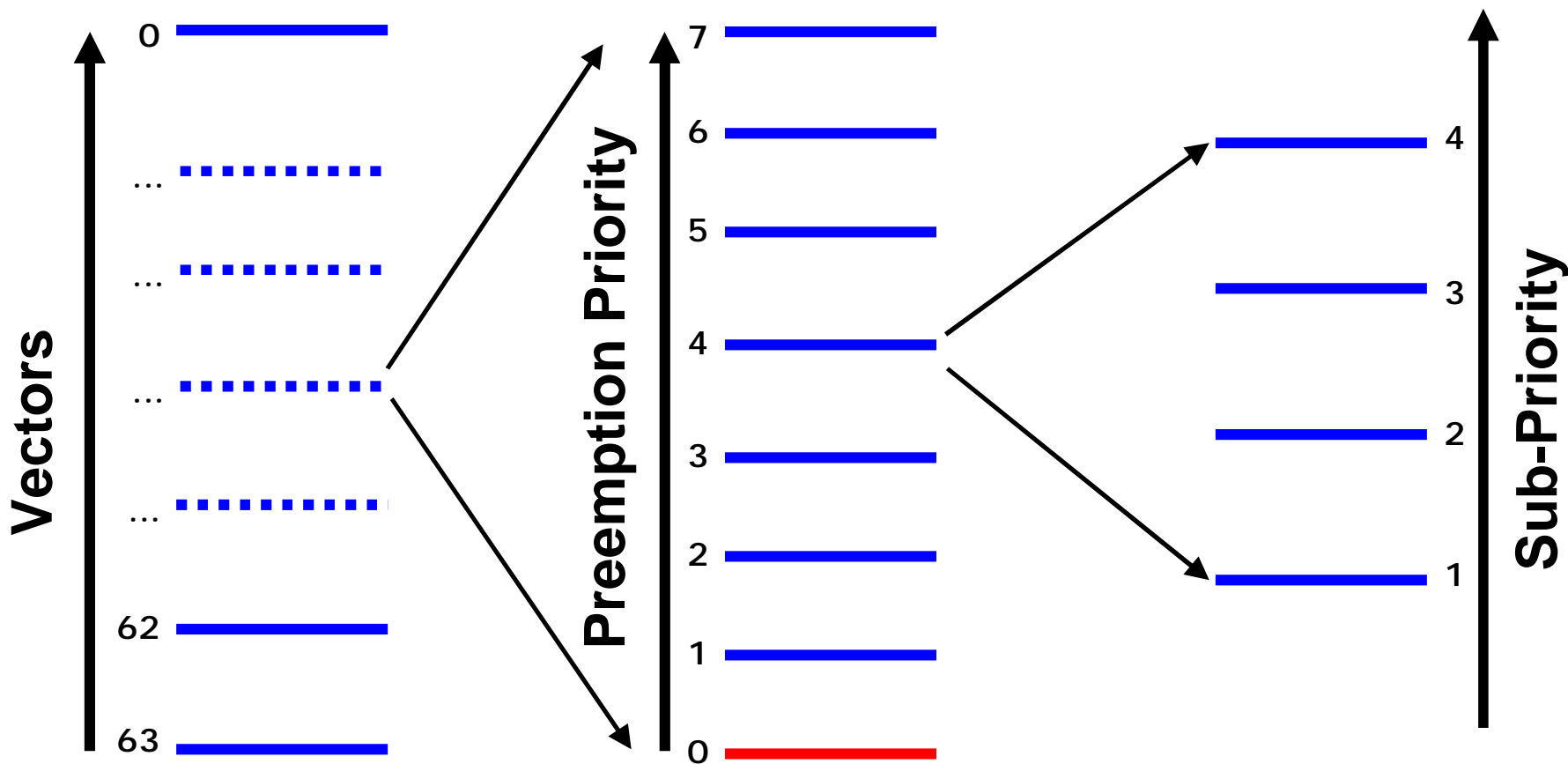
Контроллер прерываний



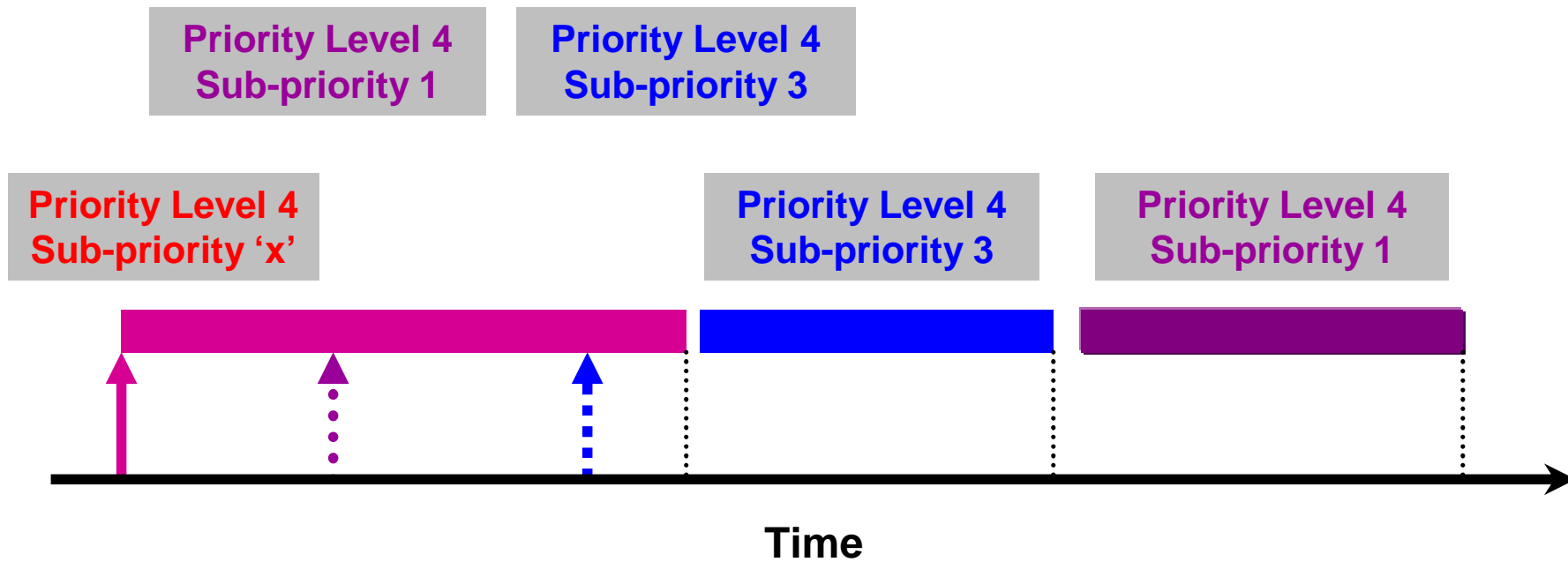
High Performance Interrupts

- | **PIC32 имеет мощную систему прерываний**
- | **Векторный контроллер прерываний**
 - | Приоритетность / на каждый приоритет 4 подприоритета / возможность вложенных прерываний
 - | Организация прерываний совместима с PIC24/dsPIC33
 - | Уникальные вектора на каждый источник
 - | Режим одного вектора для RTOS

Прерывания



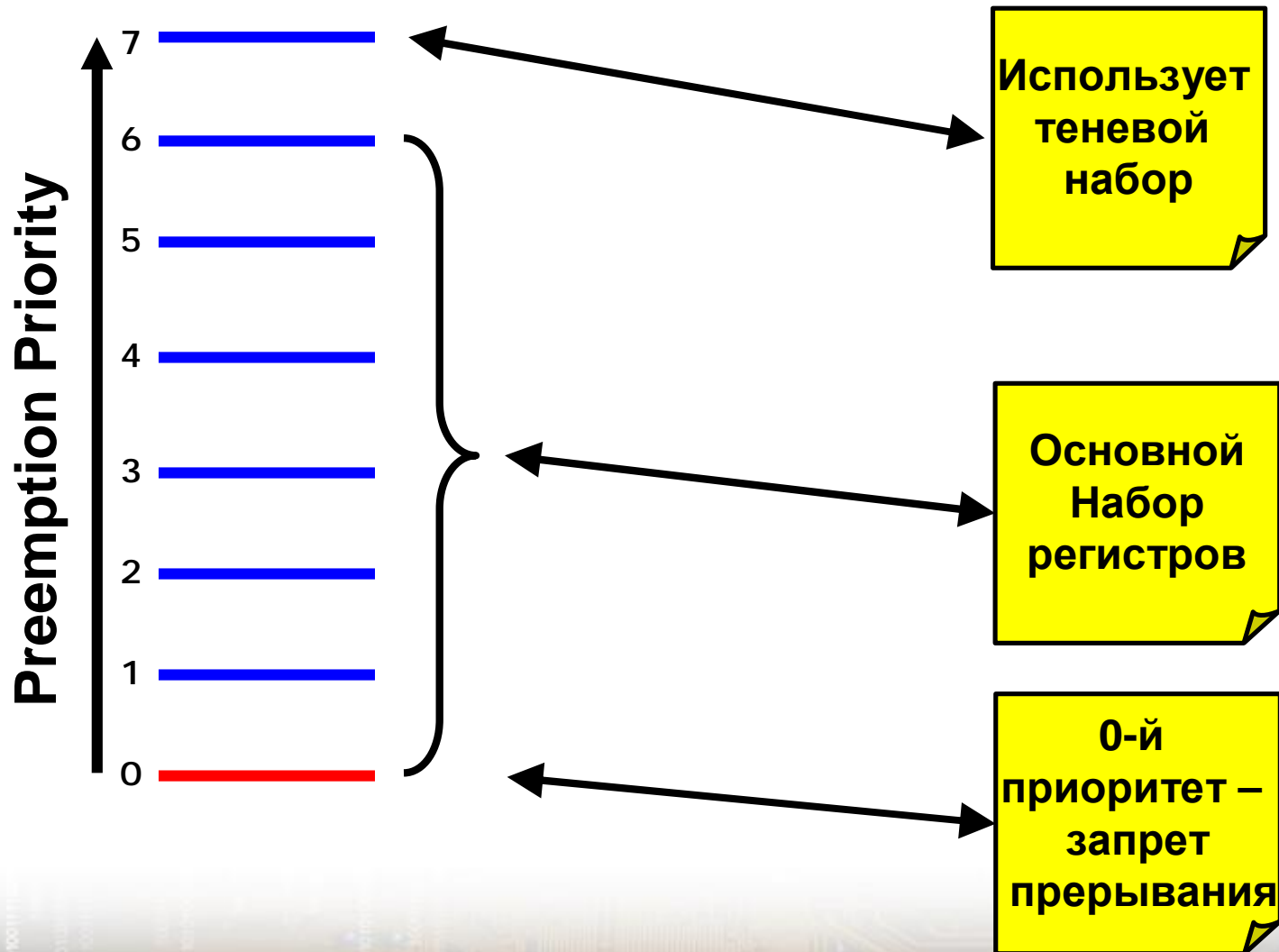
Sub-Priority пример



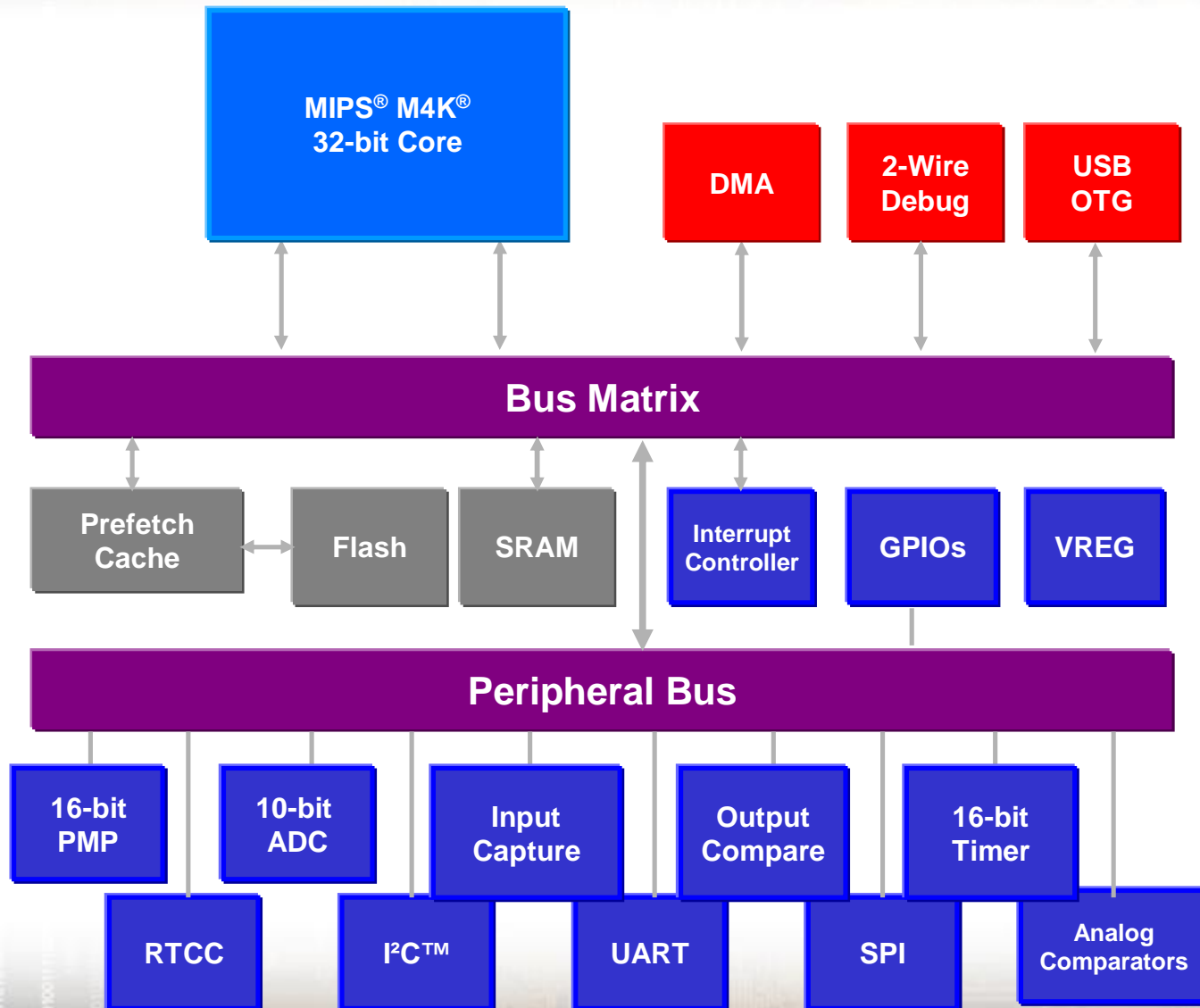
High Performance Interrupts

- | PIC32 имеет мощную систему прерываний
- | Векторный контроллер прерываний
 - | Приоритетность / на каждый приоритет 4 подприоритета / возможность вложенных прерываний
 - | Организация прерываний совместима с PIC24/dsPIC33
 - | Уникальные вектора на каждый источник
 - | Режим одного вектора для RTOS
- | **Быстрое сохранение контекста: малое время входа в прерывания!**
 - | Полный набор теневых регистров
 - | Кэш предвыборки → 2 строки имеют адресную маску → вектора прерываний имеют одинаковые старшие адреса → Линии кэша с адресной маской могут постоянно хранить пролог для всех прерываний

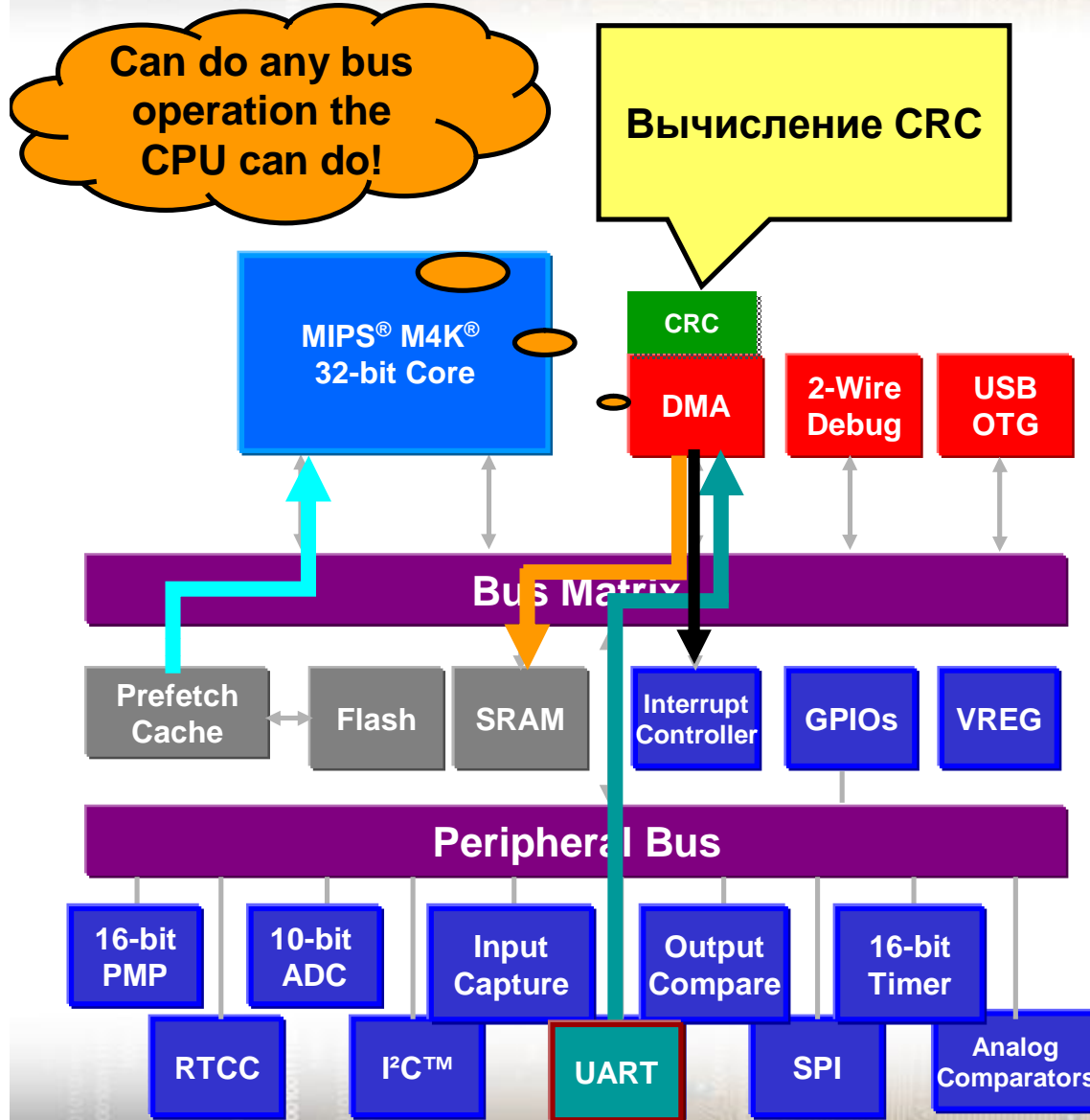
Теневой набор регистров



DMA

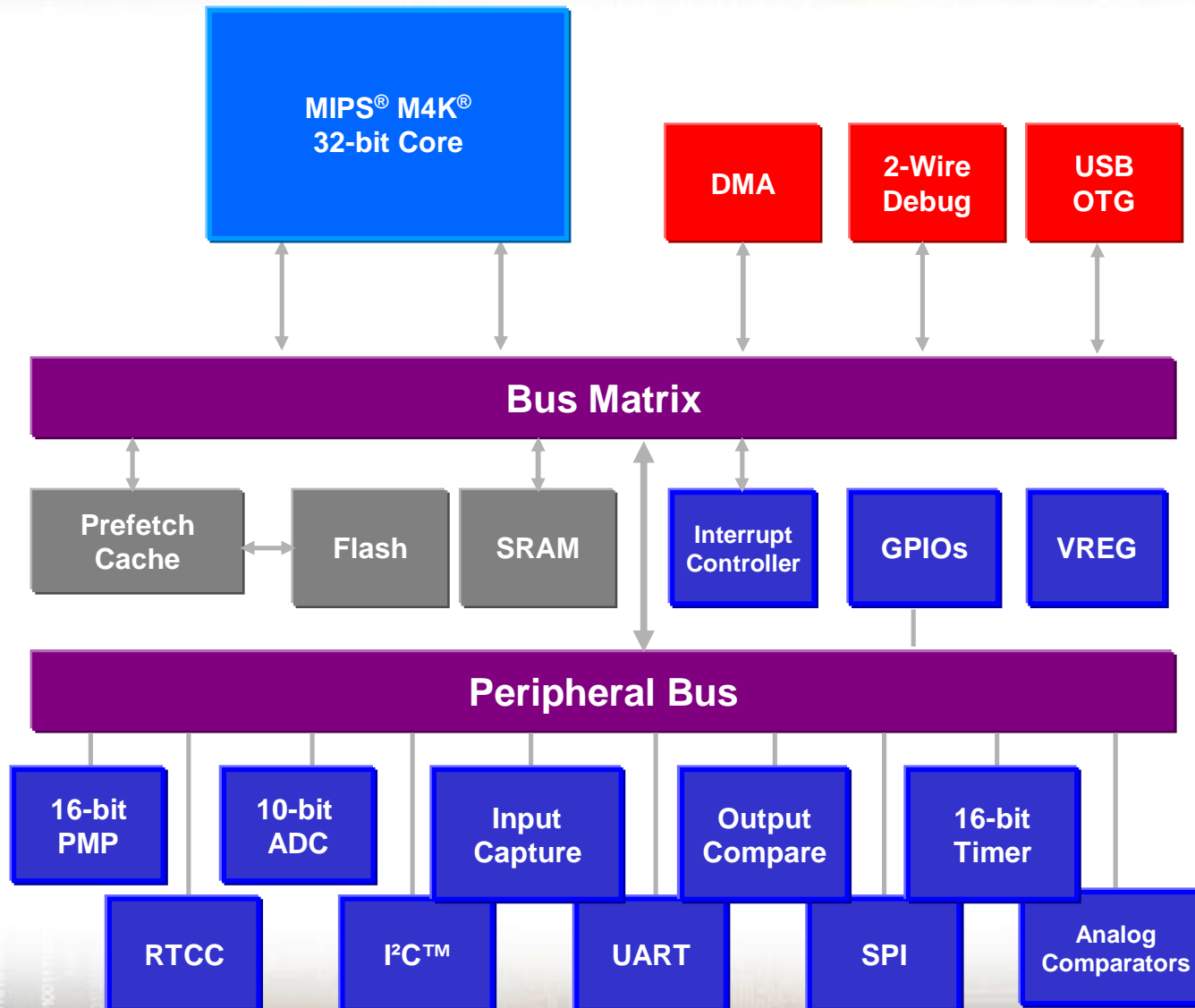


UART to Data Memory DMA Example

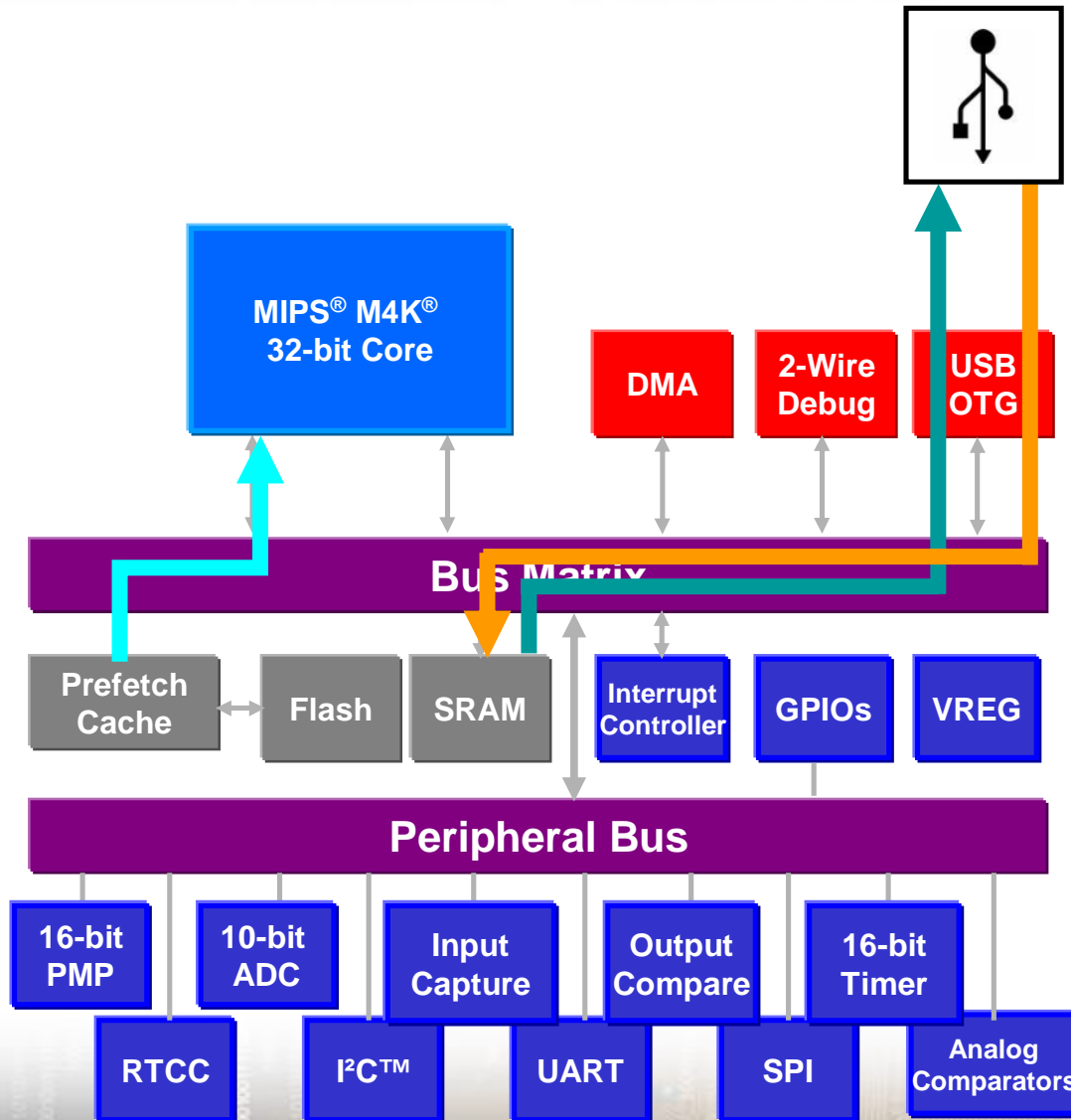


1. UART asserts interrupt to DMA
 2. Данные поступают из UART
 3. Ядро продолжает выполнять команды во время работы DMA!
 4. Возможно вычисления CRC
 5. Данные посылаются в память
 6. DMA может сформировать прерывание
- | DMA может прервать передачу при получении заданного числа байт или по получению заданного символа (например символа NULL)
- | **Варианты работы DMA:**
- | Память - память
 - | Память - периферия
 - | Периферия - память
 - | Периферия - Периферия
 - | Flash - память
 - | Flash - периферия

USB



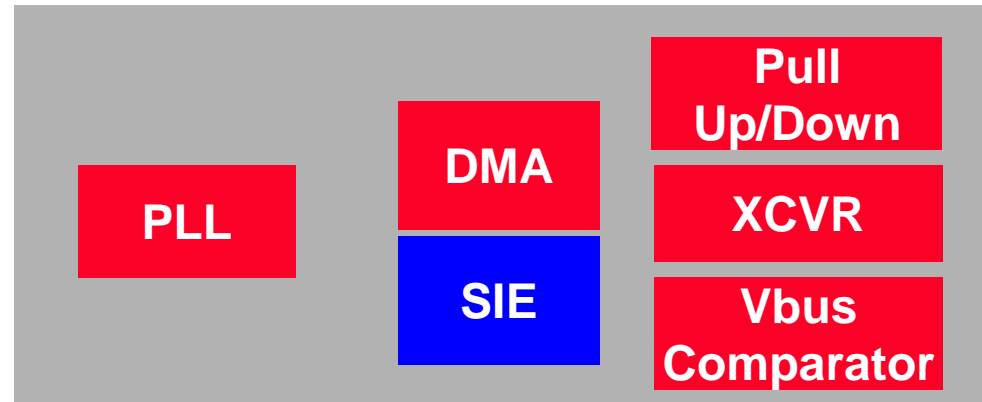
Работа USB



- | USB module operates as descriptor-based master
- | Дескрипторы это таблицы, которые содержат информацию для каждой конечной точки
- | USB имеет свой DMA

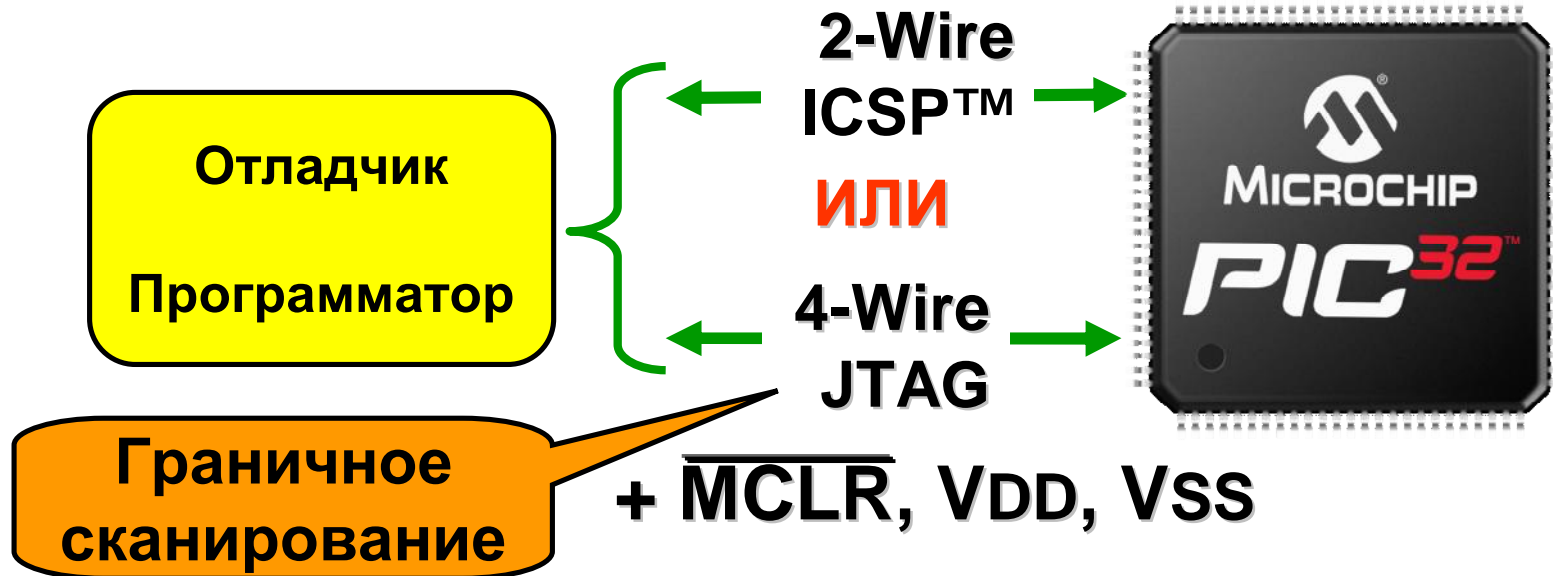
1. Данные из USB напрямую попадают в SRAM
2. Данные из SRAM поступают в USB
3. Ядро продолжает выполнять команды во время пересылки данных через USB

USB



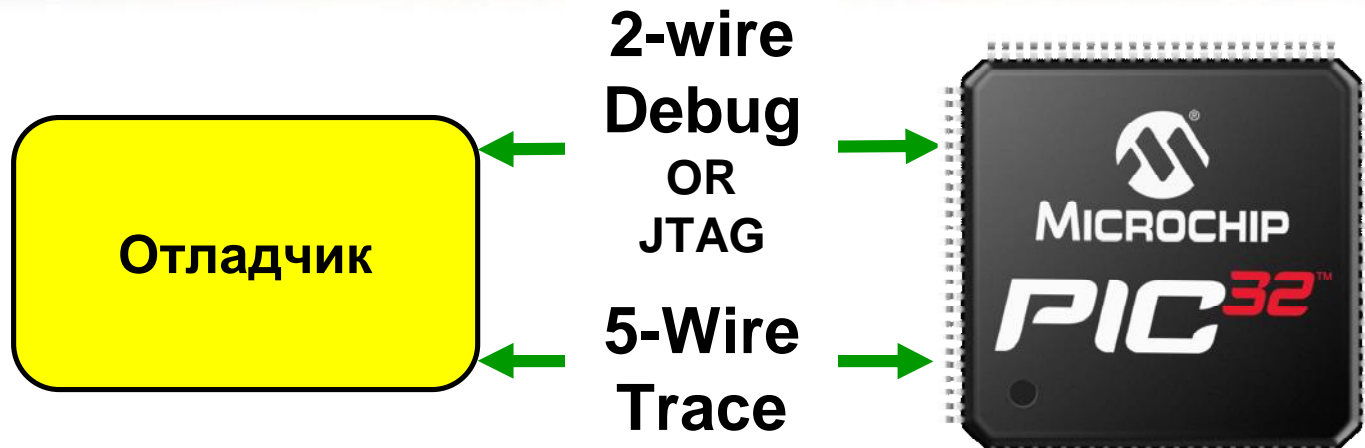
- | **2.0 FS поддерживает Device, Host, OTG**
- | **Интегрированный PHY и необходимые элементы**
 - | Подтягивающие резисторы
 - | Компараторы напряжения Vbus
 - | Мастер шины
- | **Тактирование USB независимо от тактирования CPU**

Отладка / Программирование



- | Точки останова: 6 на инструкции и 2 на Данные
- | 2 Комплексные точки останова
 - | Последовательность, по числу проходов, Секундомер
- | Доступ в реальном времени ко всей RAM и SFR-ам
- | Программные точки останова

Аппаратная трассировка



Функция ядра, требует мало выводов

- Неразрушающая трассировка, основана на аппаратных ресурсах
- ПК реконструирует ход программы
- 5 выводов, могут быть использованы как V/B

Триггер на простые и комплексные точки останова

Поддерживается внутрисхемным эмулятором MPLAB[®] REAL ICE[™] с Кабелем для трассировки

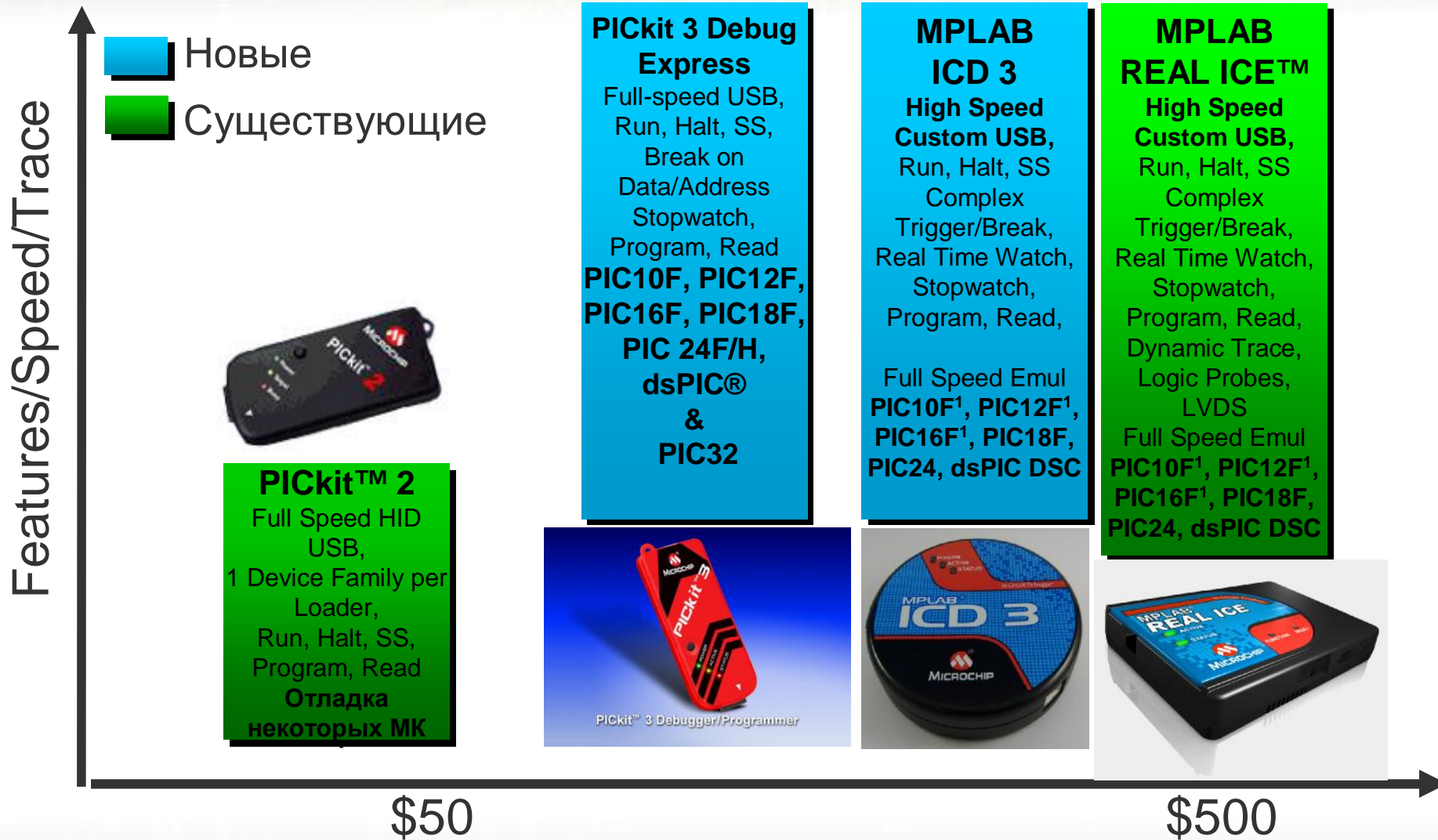
Вопрос - III

- | DMA может работать когда ЦПУ в режиме – Верно/Нет? **Верно**
- | Если я хочу автоматизировать передачу данных через USB, я должен использовать 2 DMA каналов – Верно/Нет? **Нет**
- | Комплексные точки останова влияют на время выполнения программы – Верно/Нет? **Нет**

План

- | Обзор PIC32
- | Архитектура
- | Периферия
- | **Аппаратные и Программные средства**

Средства для внутрисхемной отладки



MPLAB® ICD 3 – Следующее поколение дебаггера

- | Отладчик/Программатор для Flash контроллеров
- | Отличные возможности при низкой цене
- | Полная поддержка МК
- | Много других улучшений
 - | Возможность питания отладочной платы (100 мА)
 - | Быстрый – USB HS, аппаратное ускорение, буфер ОЗУ
 - | Программные точки останова (1000)



Что такое MPLAB[®] REAL ICE[™] эмулятор?



MPLAB[®] REAL ICE[™] In-Circuit Emulator

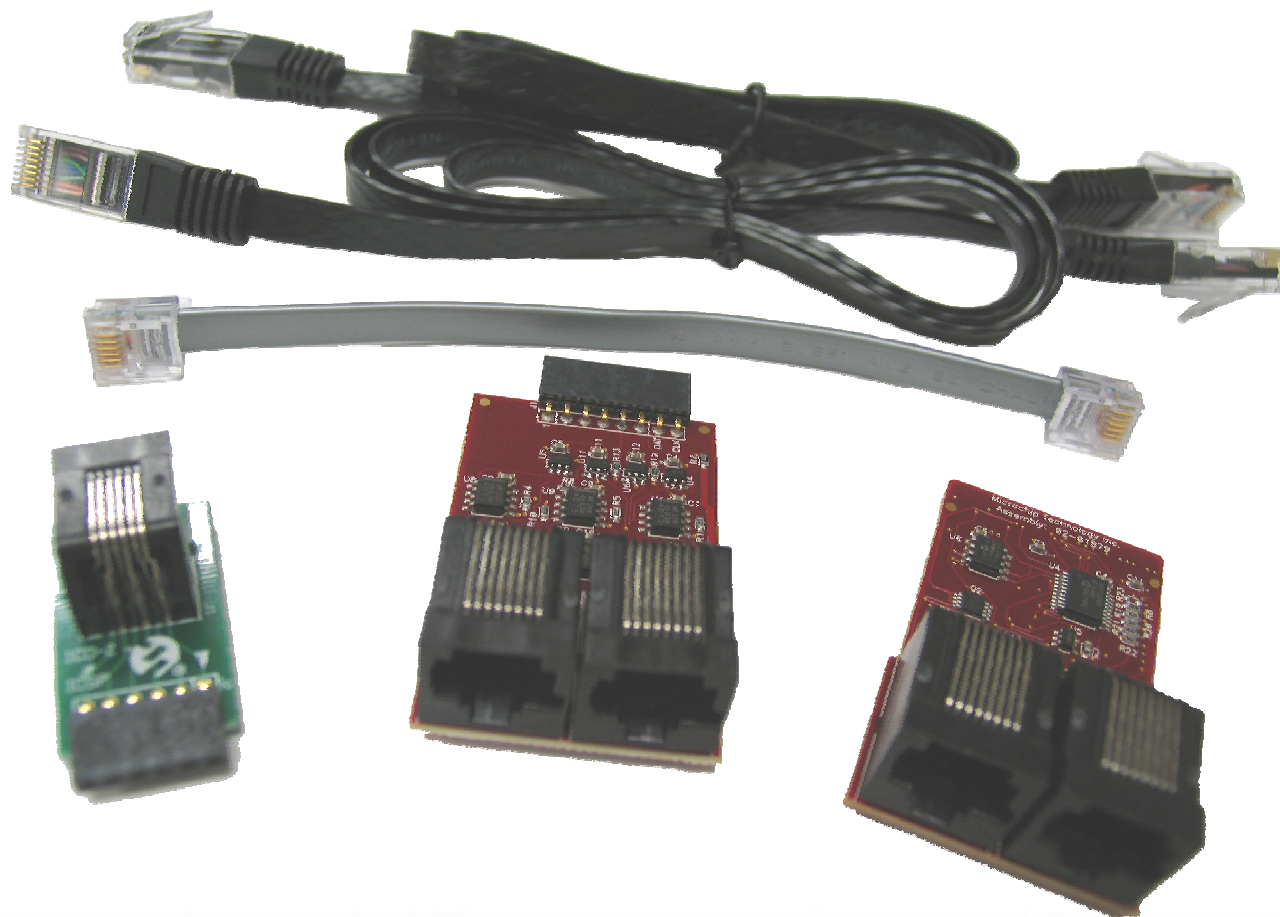
Эмулятор MPLAB® REAL ICE™

- | MPLAB REAL ICE (DV244005)
 - | RJ45 Driver Board и кабель
 - | Кабель логического пробника
 - | Плата для самотестирования





High Speed Performance Pak (AC244002)



“Базовые” особенности

- | **Интеграция с MPLAB® IDE**
- | **Работает как программатор и отладчик**
- | **Точки останова**
- | **Окно наблюдения переменных (Watch window)**
- | **Управление программой**
 - | **Run**
 - | **Animate**
 - | **Halt**
 - | **Step Over**
 - | **Single Step**
 - | **Reset**



MPLAB® REAL ICE™

Продвинутые возможности

- | **Наблюдение за переменной в реальном времени**
- | **DMCI – Data Monitoring and Control Interface**
- | **Трассировка**
- | **Секундомер**
- | **Расширенные точки останова**



MPLAB® REAL ICE™ Эмулятор vs. Обычный эмулятор

Особенности по отношению к обычному эмулятору

- | Реальный контроллер а не эмуляционный чип
- | Работает в системе
- | Возможна высокоскоростная отладка при низкой цене

Расширенные ВОЗМОЖНОСТИ для PIC32



Расширенные точки останова

- | **Точки останова по изменению данных**
- | **Event Breakpoints**
- | **ANDed Breakpoints**
- | **Sequenced Breakpoints**
- | **Секундомер**



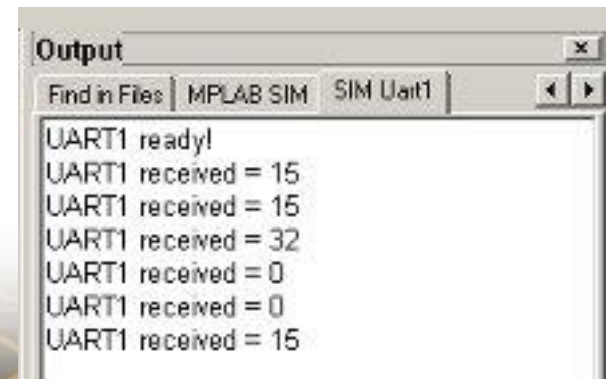
Симулятор

| **Может являться мощным
средством отладки**

Ввод-вывод данных в симуляторе

| Ввод-вывод данных через порт UART в симуляторе

```
while(1) {  
    while(!DataRdyUART1()); //Wait for data in the UARTRx.  
    data = (char)ReadUART1(); // Read data from Rx.  
    putcUART1(data); // Write data into Tx.  
}
```



Форматированный вывод в симуляторе

ANSI библиотечные функции ввода-вывода

- | Библиотечные функции C32 направляют поток **stdout** в порт **UART2**, а MPLAB IDE выводит в окно **Output** данные, направляемые в **UART1**.
 - | Для перенаправления потока в UART1 нужно определить функцию **_mon_putc (char c)**. Без этого определения компилятор будет использовать функцию из библиотеки.

```
while (1) {
```

```
    OpenUART1( UART_EN, UART_RX_ENABLE | UART_TX_ENABLE,  
pbClk/16/DESIRED_BAUDRATE-1);
```

```
    printf("UART1 ready!\n");
```

```
    ...
```


PIC32 Starter Kit

- | Все необходимое для начала работы с PIC32
- | MPLAB® IDE и MPLAB C Компилятор для PIC32
- | 37 примеров с исходными кодами
- | 2 TCP/IP стека
- | Питание от USB
- | Интегрированный программатор и дебаггер



DM320001



Explorer + PIC32 Starter

USB Starter Board (DM320003)



Starter Kit PIM Adapter (MA320002)



Explorer 16 (DM240001)



┆ PIC32 Starter Kit

┆ **DBINIT(void)**

┆ **DBPUTC(const unsigned char *c)**

┆ **DBPUTS(const unsigned char *s)**

┆ **DBGETC(unsigned char *c)**

┆ **DBGETS(unsigned char *s, int len)**

┆ **DBPRINTF(const char *format, ...)**

Контекстный ввод-вывод

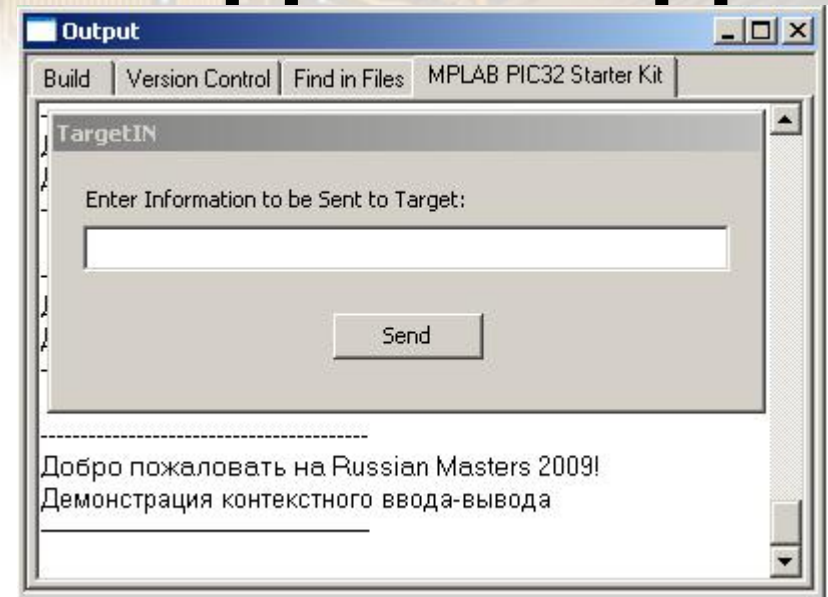
```
int main (void) {
```

```
    DBINIT();
```

```
    DBPUTS("Добро пожаловать на  
Russian Masters 2009!\n");
```

```
    DBGGETS(buffer, sizeof(buffer));
```

```
    ...
```

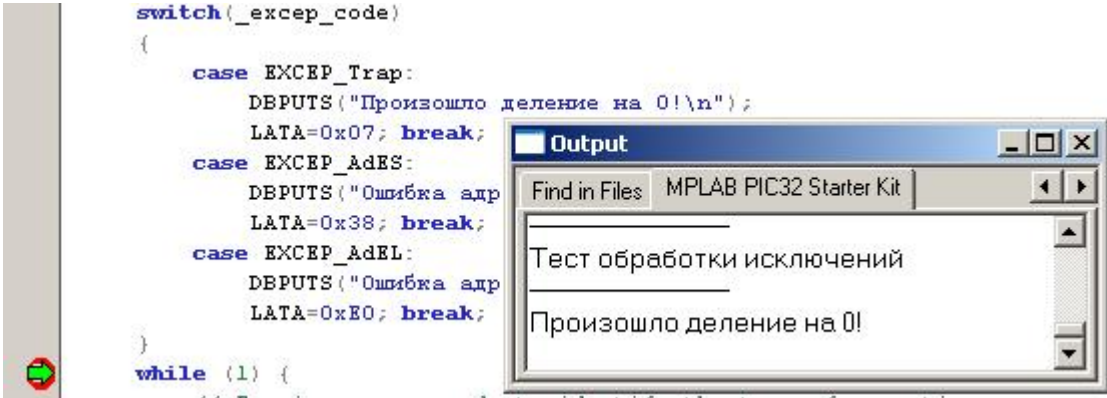


Обработка исключений

С помощью отладчика можно увидеть что произошло «Исключение» (ошибка адреса, переполнение арифметических операций, деление на ноль, и т.п.)

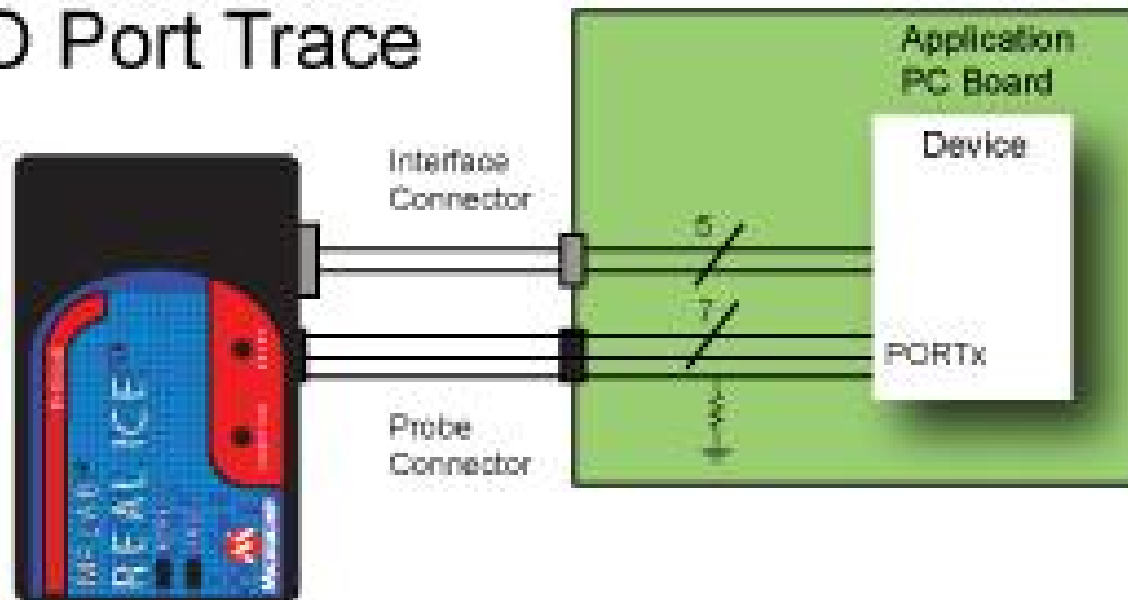
REAL ICE имеет функцию трассировки кода

```
switch(_excep_code)
{
    case EXCEP_Trap:
        DBPUTS("Произошло деление на 0!\n");
        LATA=0x07; break;
    case EXCEP_AdES:
        DBPUTS("Ошибка адр");
        LATA=0x38; break;
    case EXCEP_AdEL:
        DBPUTS("Ошибка адр");
        LATA=0xE0; break;
}
while (1) {
```



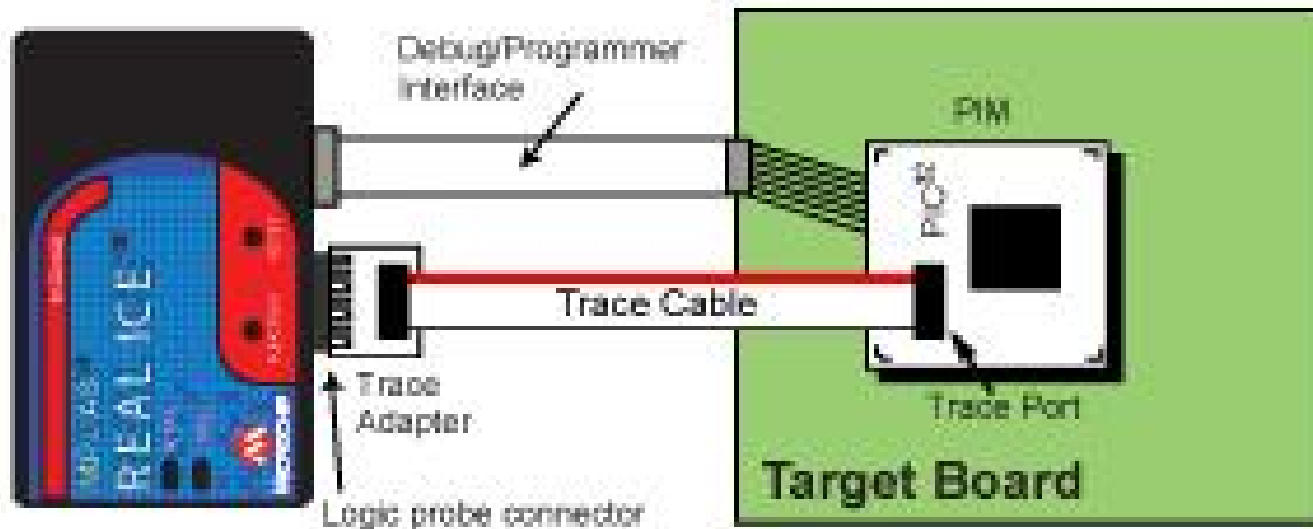
Требует 8 выводов

I/O Port Trace



Трассировка инструкций

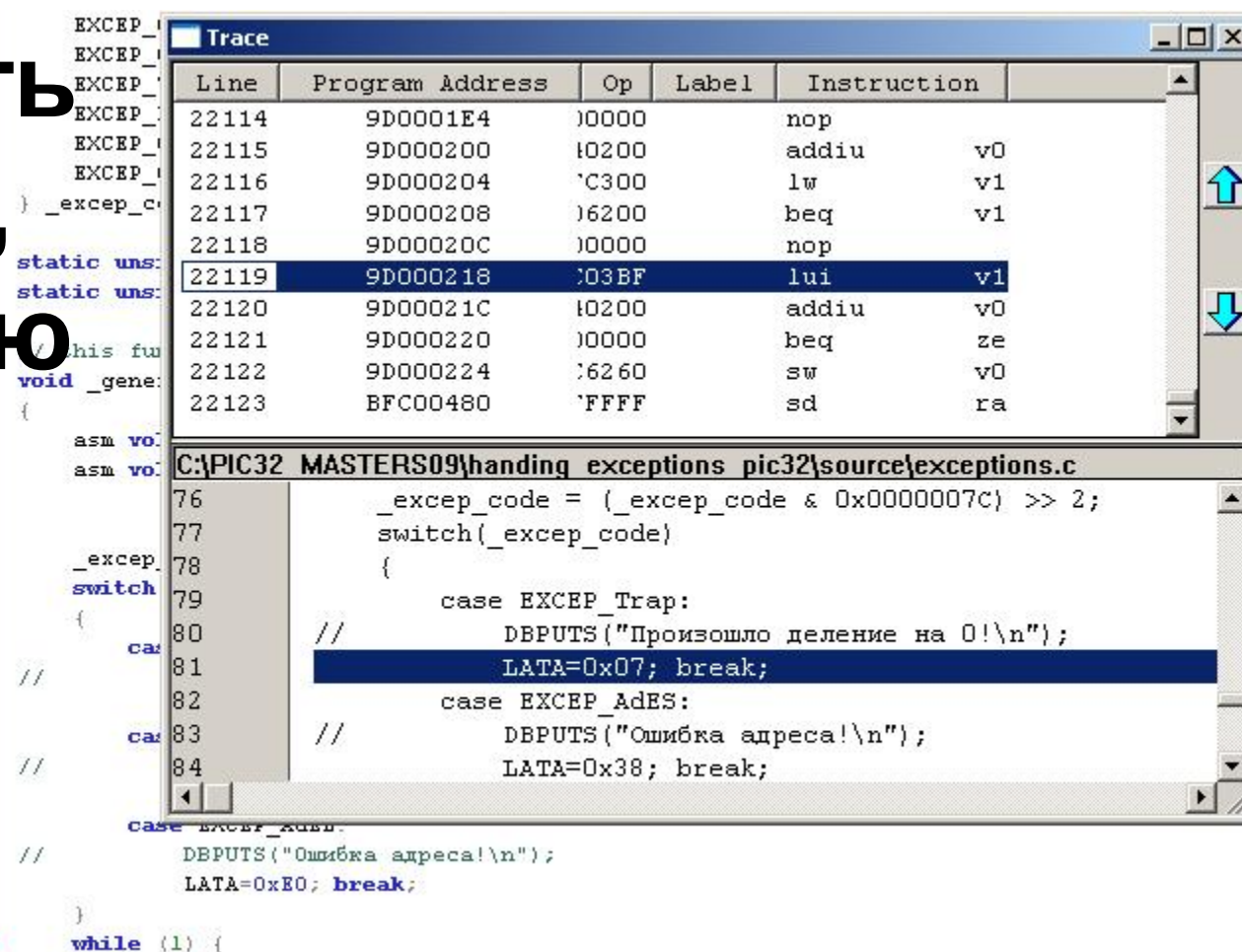
- | Встроенная периферия, не требуются макросы
- | Показывает выполнение всех инструкций
- | Пока только для PIC32MX
- | Требуется 5 выделенных выводов
- | MPLAB® REAL ICE™ Trace Kit (AC244006)



Трассировка кода

Можно «отмотать» назад и

определить
операцию,
приведшую
к ошибке



The screenshot displays a debugger interface with two windows. The top window, titled "Trace", shows a list of instructions with columns for Line, Program Address, Op, Label, and Instruction. The instruction at line 22119 is highlighted in blue. The bottom window shows the source code for "exceptions.c" with the line `LATA=0x07; break;` highlighted in blue, corresponding to the instruction in the trace window.

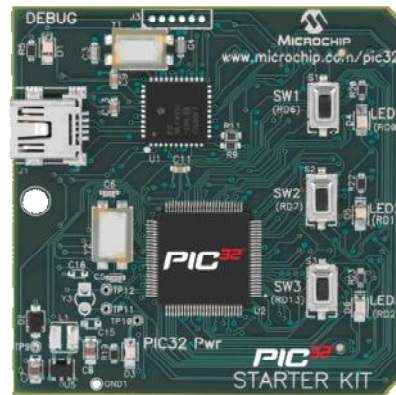
Line	Program Address	Op	Label	Instruction
22114	9D0001E4	00000		nop
22115	9D000200	02000		addiu v0
22116	9D000204	03000		lw v1
22117	9D000208	06200		beq v1
22118	9D00020C	00000		nop
22119	9D000218	03BF		lui v1
22120	9D00021C	02000		addiu v0
22121	9D000220	00000		beq ze
22122	9D000224	06260		sw v0
22123	BFC00480	0FFF		sd ra

```
C:\PIC32 MASTERS09\handing exceptions pic32\source\exceptions.c
76     _excep_code = (_excep_code & 0x0000007C) >> 2;
77     switch(_excep_code)
78     {
79         case EXCEP_Trap:
80             // DBPUTS("Произошло деление на 0!\n");
81             LATA=0x07; break;
82         case EXCEP_AdES:
83             // DBPUTS("Ошибка адреса!\n");
84             LATA=0x38; break;
85     }
86     // DBPUTS("Ошибка адреса!\n");
87     LATA=0xE0; break;
88 }
89 while (1) {
```

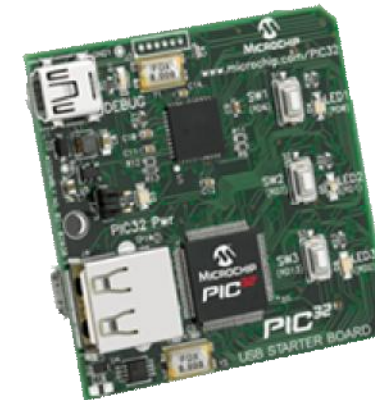


PIC32-Specific Tools (By Microchip)

**General Starter Kit
(DM320001)**

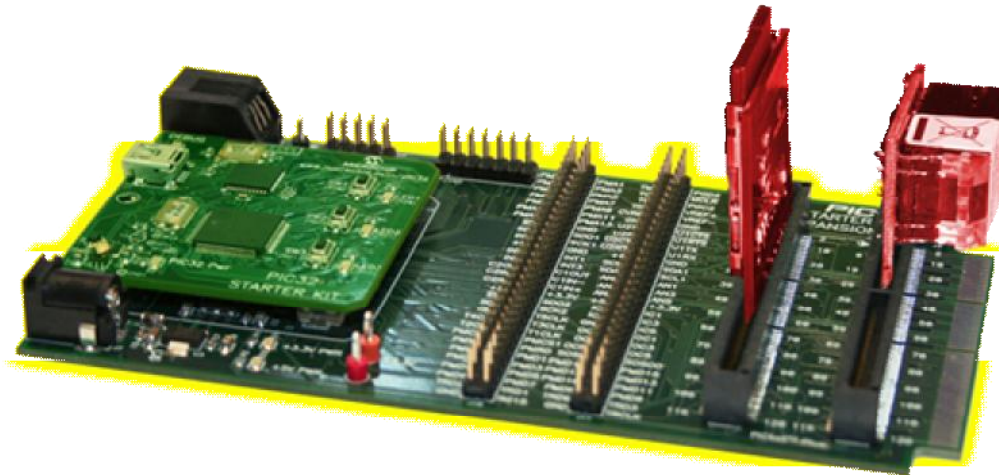


**USB Starter Board
(DM320003)**



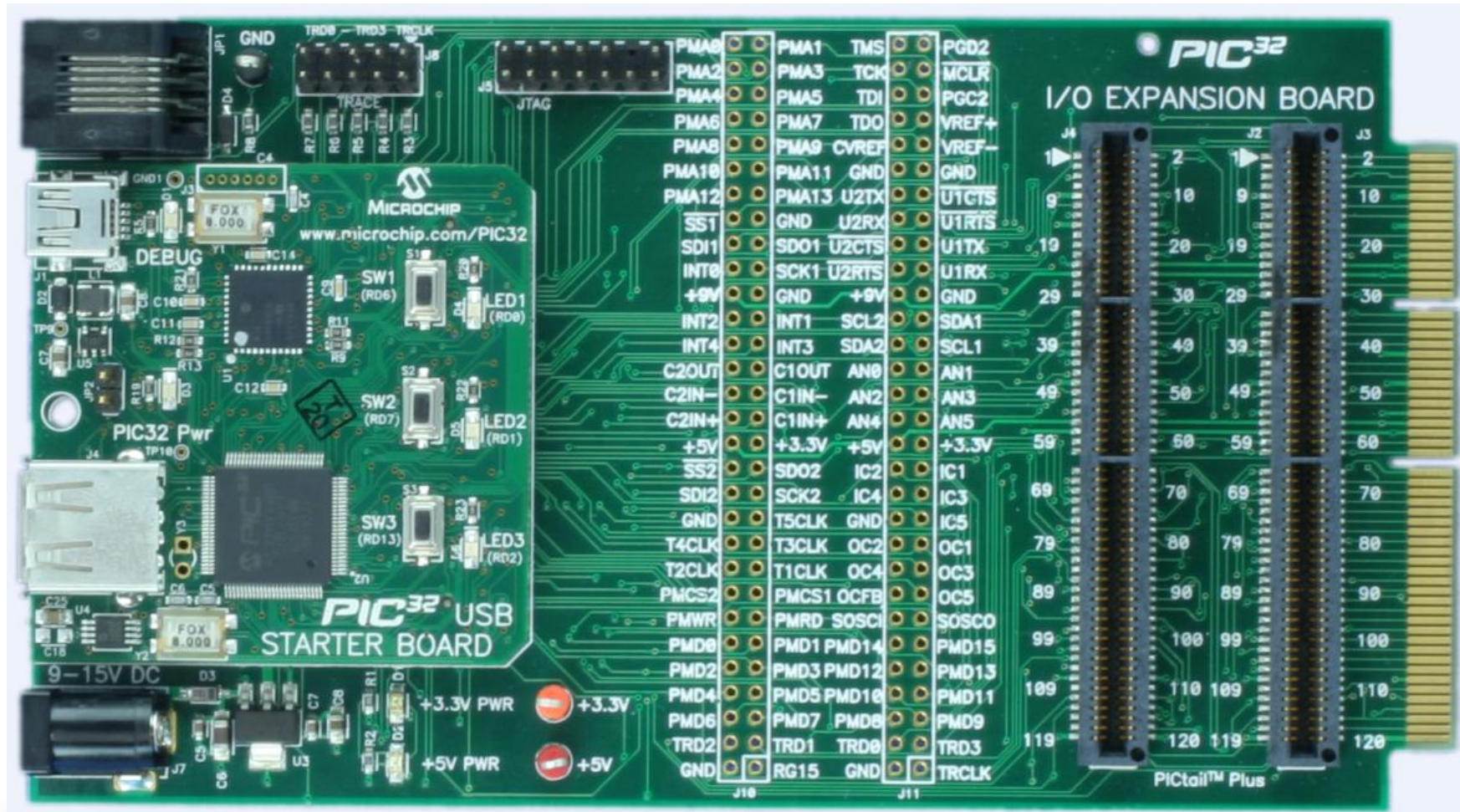
MPLAB® C32

(Software006015)

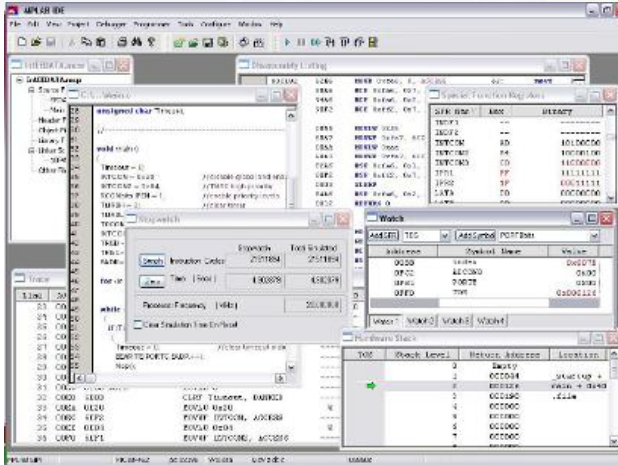


**I/O Expansion Board
(DM320002)**

PIC32-Specific Tools (By Microchip)



Compatible Tools (By Microchip)



MPLAB® IDE
(Software007002)



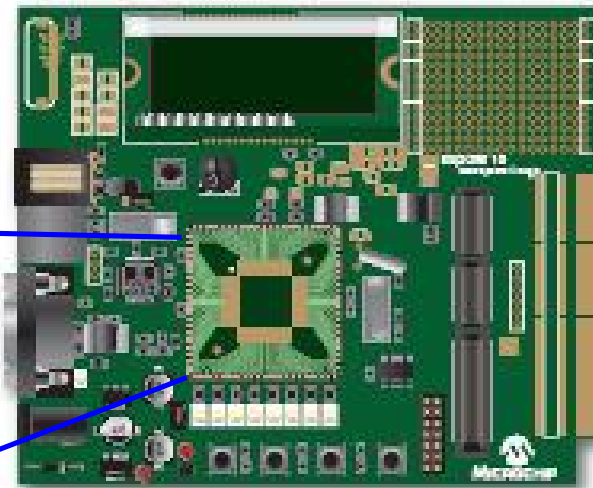
MPLAB® REAL ICE™
In-Circuit Emulator
(DV244005)



MPLAB® ICD 2/3
(DV164005 / DV164035)



PIC32 PIM
(MA320001, 2)



Explorer 16
Development Board
(DM24000x)



PICkit™ 2/3
(DV164121 / DV164131)



Microchip Software Solutions

Connectivity	<ul style="list-style-type: none"> Microchip TCP/IP with SSL and BSD ZigBee® Protocol Stack for 802.15.4* MIWi™ Protocol Stack for 802.15.4 Networks
USB	<ul style="list-style-type: none"> USB Host, Device USB OTG, Dual Role* USB Class Drivers – HID, MSD, CDC, Custom, etc.
Graphics	<ul style="list-style-type: none"> Microchip Graphics Library
CAN	<ul style="list-style-type: none"> Standalone CAN Library
Audio	<ul style="list-style-type: none"> Speex, ADPCM, PCM Encoding/Decoding Library
Encryption	<ul style="list-style-type: none"> Public Key Cryptography Library (RSA)
Basic Libraries	<ul style="list-style-type: none"> 32-bit File System Library 16-bit File System Library Math Library Peripheral Library EEPROM Emulation DSP Library
Bootloader	<ul style="list-style-type: none"> Serial Port Bootloader USB Host Bootloader*

* Available in Beta or scheduled for future release



PIC32 Third Party Support

IDE, C/C++ Compiler and Debugger

Now part of the Microchip Family



Embedded RTOS Support



Graphics GUI Support



Up-to-date list at www.microchip.com/pic32

Что Будет в ближайшем Будущем

- Доступность в корпусах 10x10 BGA & 9x9 QFN

Следующее поколение PIC32

- Больше DMA каналов
- Два CAN 2.0B
- 8-128 KB SRAM
- 10/100 Ethernet MAC
- До 6 UART, 5 I²C™ и 4 SPI



Дополнительные ресурсы

| www.microchip.com/pic32

| Примеры конкурса дизайнов на PIC32
www.myPIC32.com

| E-Learning - www.microchip.com/training

| Книжки

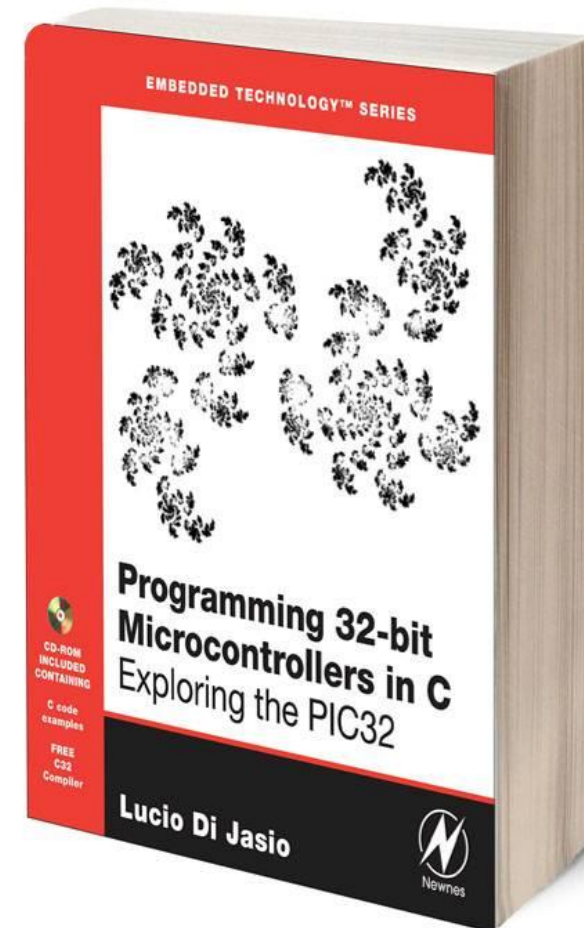
- | *Exploring the PIC32* by Lucio Di Jasio
- | *See MIPS Run* by Dominic Sweetman



Programming 32-bit Microcontrollers in C: Exploring the PIC32

- | Accelerate the learning curve
- | Experience the simplicity of PIC32
- | Utilize compatibility with PIC24

15 примеров



Written by long-time Microchip employee & a member of the PIC32 team

Итоги

- | **Сегодня мы изучили:**
 - | Архитектуру PIC32
 - | Периферию
 - | Программные и Аппаратные Средства



Thank You!

Questions?



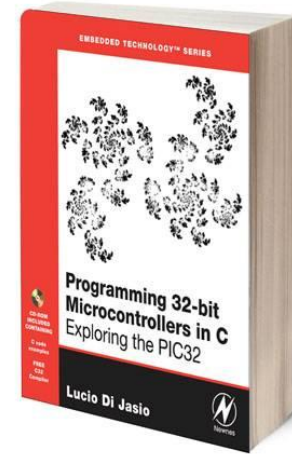
Getting Started with PIC32

Introduction to PIC32 Hands-on Workshop (MCU4101)

Prerequisites: None

Attendees learn the basics of programming the PIC32

www.microchip.com/training

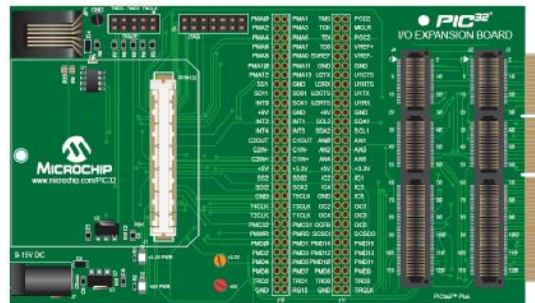


Recommended Development Tools



PIC32 Starter Kit
DM32001

USB version: DM32003



PIC32 I/O
Expansion Board
DM32002



PIC32 Plug-in Module
(Explorer 16 Users Only)
MA32001

USB Version: MA32002



Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KeeLoq, KeeLoq logo, MPLAB, PIC, PICmicro, PICSTART, rPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, nanoWatt XLP, Omniscient Code Generation, PICC, PICC-18, PICkit, PICDEM, PICDEM.net, PICtail, PIC32 logo, REAL ICE, rLAB, Select Mode, Total Endurance, TSHARC, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2009, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.