# *Embedded Systems*

## *Memory and memory map*

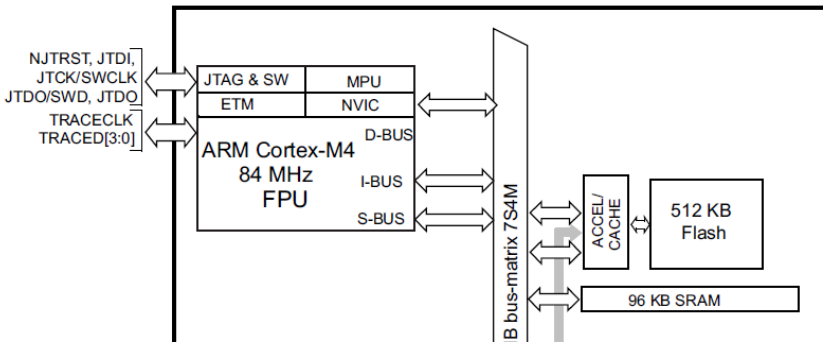
## *Lesson 05*

**Francesco Menichelli**

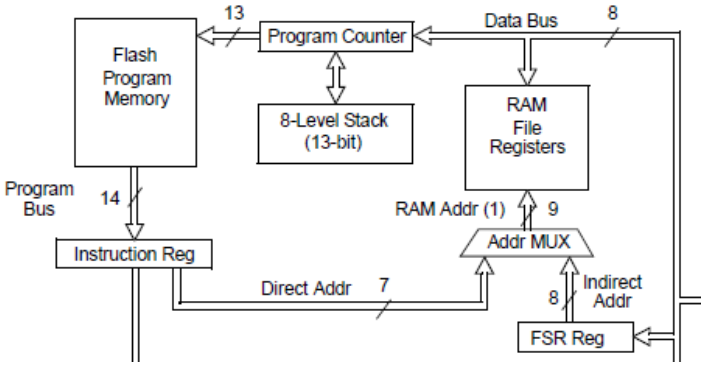
**francesco.menichelli@uniroma1.it**

# Memory – Data retention

- Memory devices are used to store program and data for the CPU
- They can be divided in two groups depending on a parameter known as **data retention**
  - Volatile (data is lost at power off)
    - SRAM, DRAM
  - Non volatile (data is retained after a power off)
    - EPROM, EEPROM, FLASH



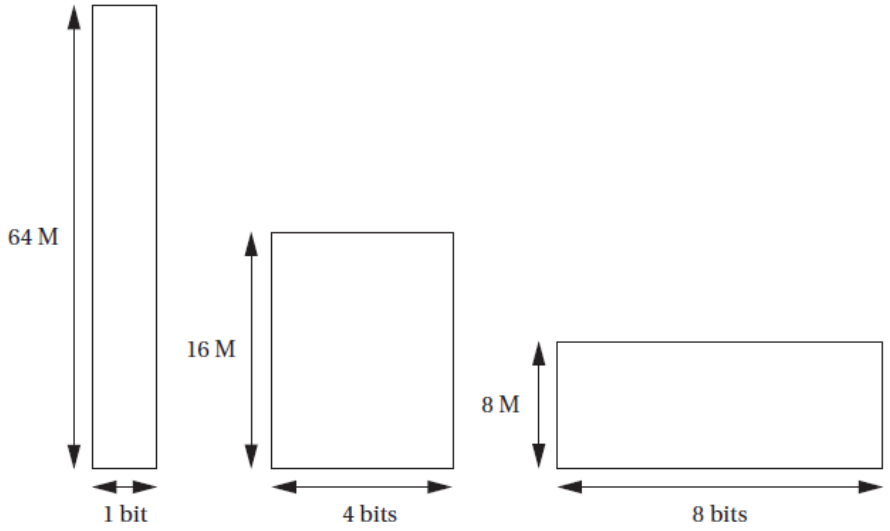

STM32F401 (ST Microelectronics)




PIC16F648 (Microchip)

# Memory – Size

- The most basic way to characterize a memory is by its capacity
  - E.g. 64 Mbit.
- However, given size, manufacturers usually make several versions of a memory with a different data width. This parameter is called **aspect ratio**



# Memory - RAM

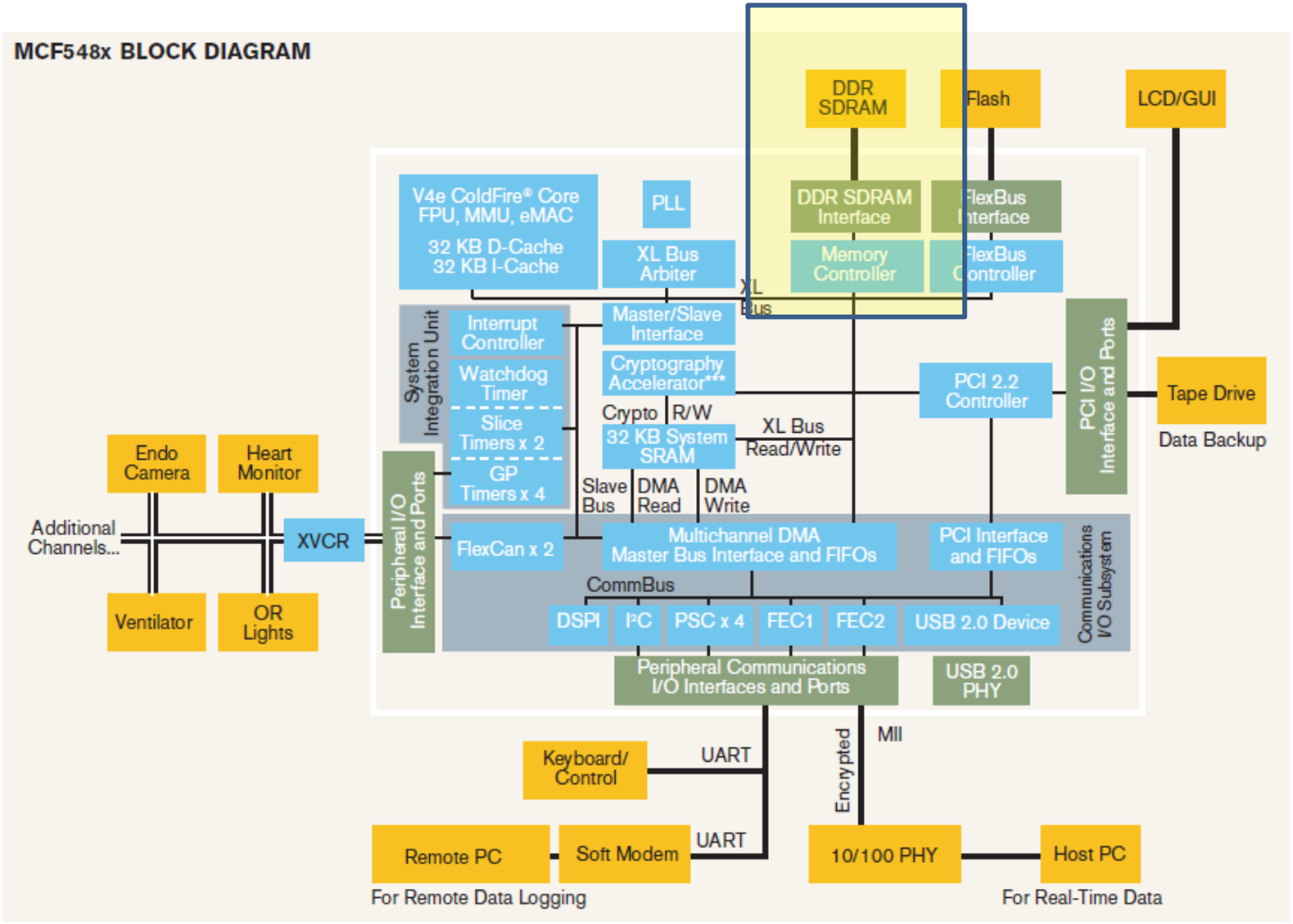
- Static RAM (SRAM)
  - Manufactured using standard logic
  - Consume only leakage current if not used
  - It basic cell is composed by 6 transistors
  - Higher speed, Higher cost
- Dynamic RAM (DRAM)
  - Manufactured using dedicated processes
  - Must be refreshed if not used (consumes power)
  - It basic cell is composed by 1 transistor
  - Lower speed, Lower cost

# Memory - RAM

- With present technology, SRAM in the order of 1-1000 kbyte can ben easily integrated in the microcontroller
- If system requires above 1Mbyte of RAM, external DRAM could be the only solution.
  - Embedded Linux requires at least 4-8Mbyte to run smoothly
- DRAM requires hardware for refresh function (DRAM controller)

- Using integrated SRAM is the preferred choice:
  - Cost
  - Speed
  - Reliability

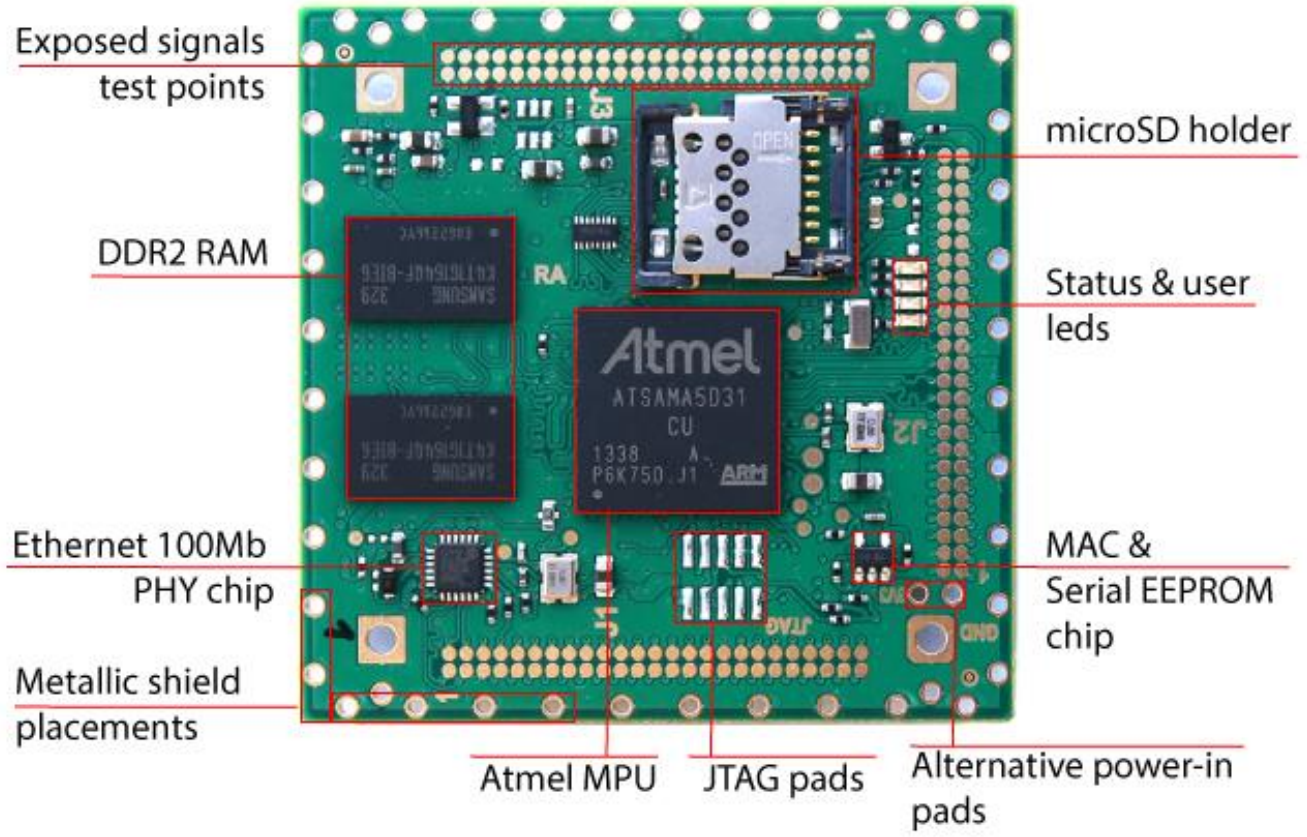
# Example – Coldfire MCF548x (Freescale)



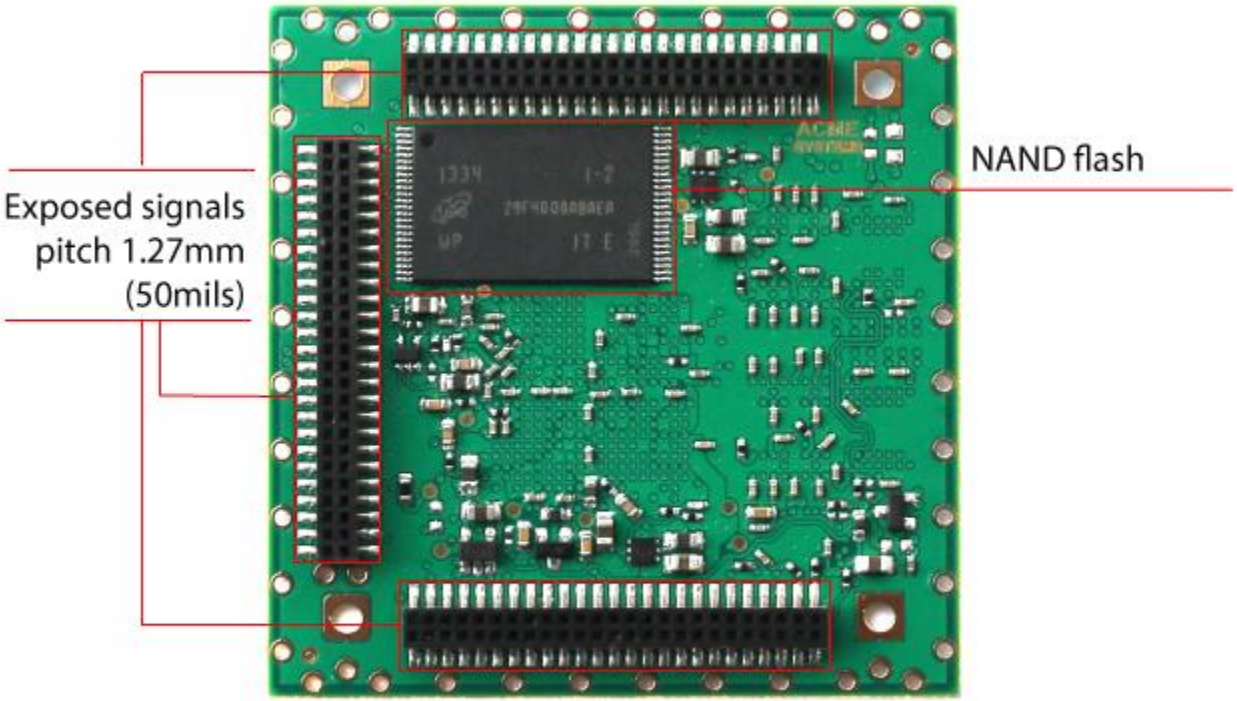
# Memory - FLASH

- The most common non-volatile memory technology
  - Slow write, reduced lifetime
  - Read is faster than write, but not as SRAM
- Used for program memory in microcontrollers
  - Integrated or external
- For medium/high performance CPU (approx. above 50MHz clock), can be too slow to respond in 1 cycle
  - architectural techniques to raise bandwidth

# Board example – external memory

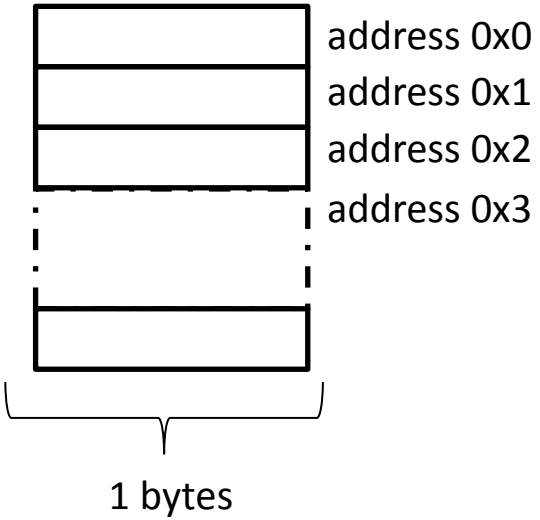


# Board example – external memory



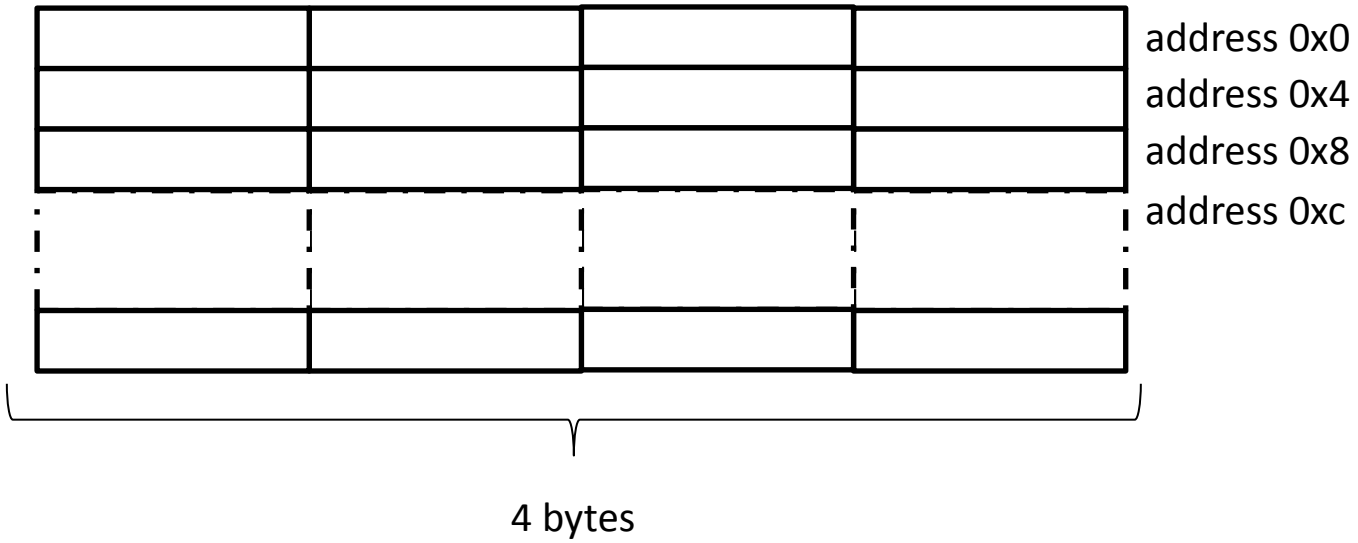
# Memory in 8-bit architectures

- The smallest element of a memory is a byte (8-bit)
- Memory address space is indexed on bytes
  - byte 0 is at address 0, byte 1 is at address 1
- Processor can access memory in bytes.



# Memory in 32-bit architectures

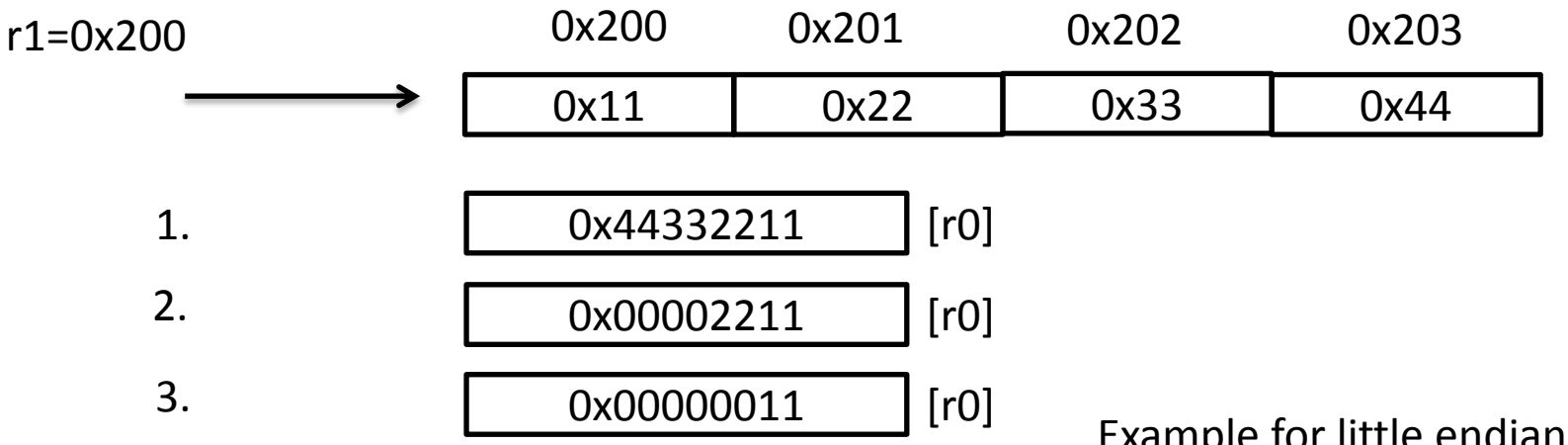
- The smallest element of a memory is a byte (8-bit)
- Memory address space is indexed on bytes
  - byte 0 is at address 0, byte 1 is at address 1
- Processor can access memory in words (4 bytes) half words (2 bytes) or bytes.



# Memory in 32-bit architectures

- The ARM processor can generate 3 kinds of memory accesses: word, half word, bytes:

1. LDR r0,[r1] ; read a word (4 byte) in one step
2. LDRH r0,[r1] ; read a half word and put it in r0 (bits 31-16 in r0 are zeroed)
3. LDRB r0,[r1] ; read a byte and put it in r0 (bits 31-8 in r0 are zeroed)




Example for little endian format

# Memory mapped I/O

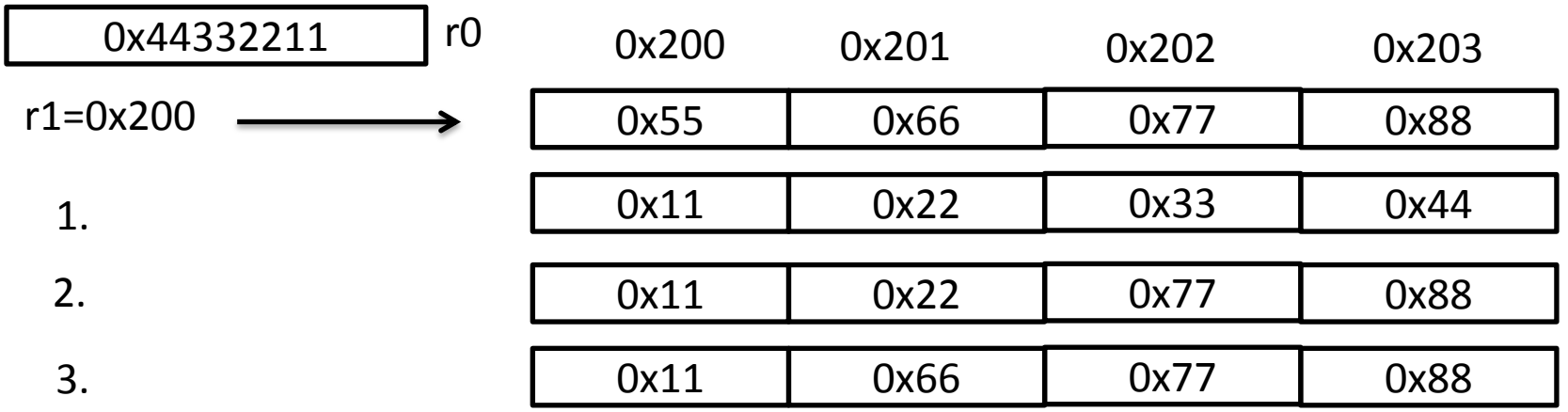
- Device registers are mapped within the CPU addressable memory space
- In order to read or change the value of a register, CPU performs load and store operations, as for memory
  - LDR r0,[r1]
  - STR r0,[r1]

  *Where r1 has been previously loaded with the address of the register.*

# Memory mapped I/O

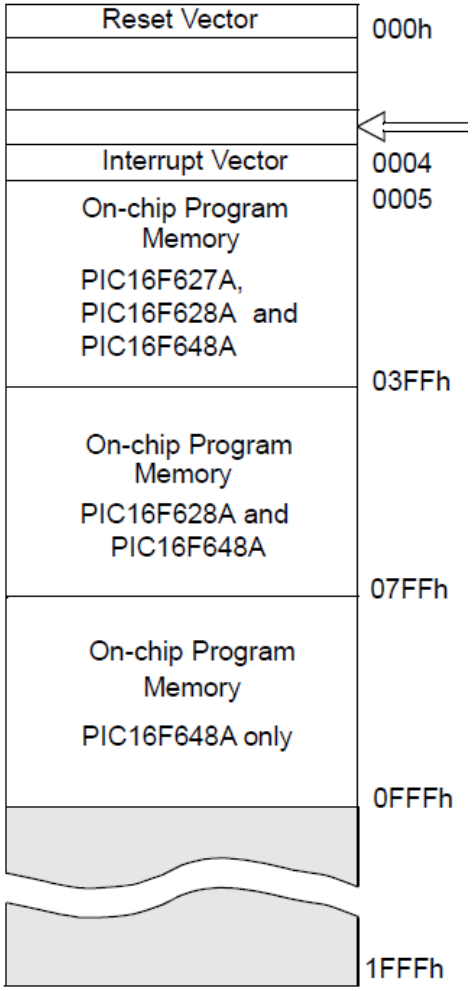
- Access size for registers

1. STR r0,[r1] ; write a word (4 bytes) in one step
2. STRH r0,[r1] ; write a half word (2 bytes)
3. STRB r0,[r1] ; write a byte



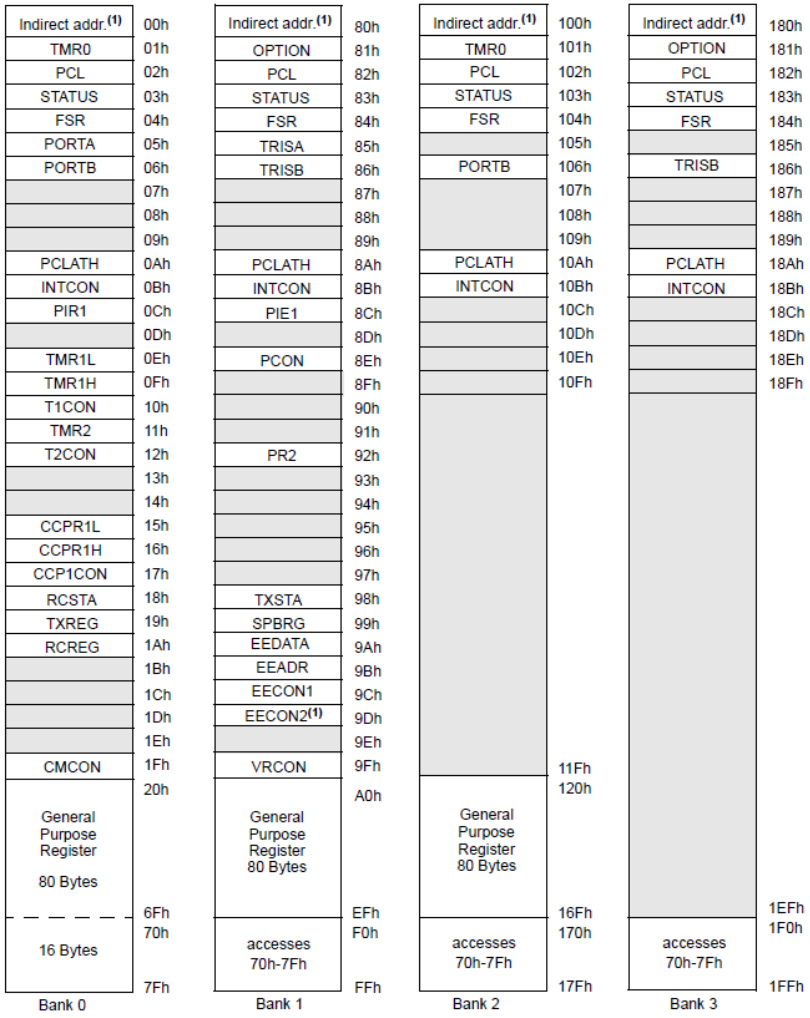

Example for little endian format

# Memory map example 8-bit microcontroller




PIC16F648 (Microchip) – Program memory

# Memory map example 8-bit microcontroller

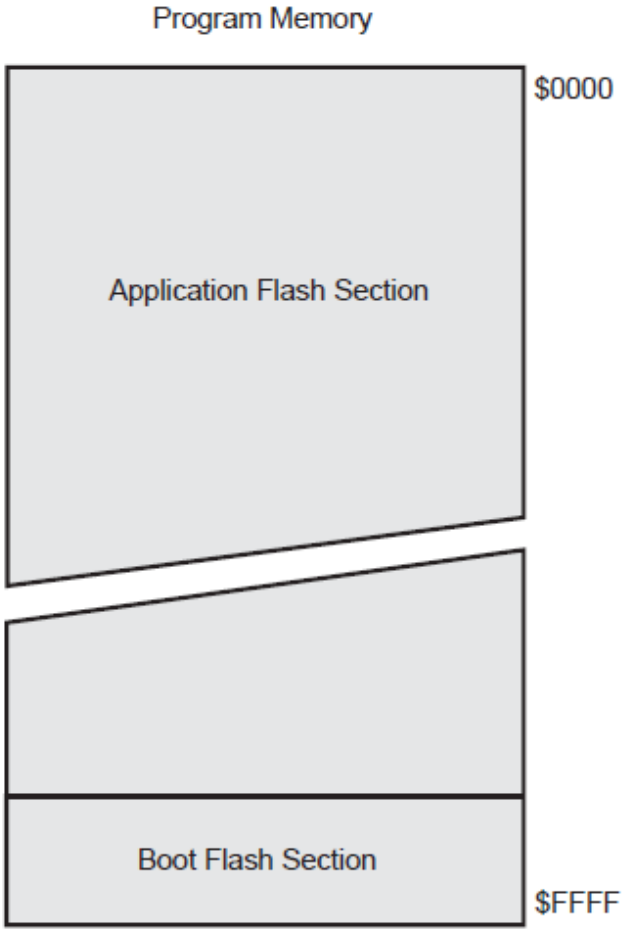



PIC16F648 (Microchip) – Data Memory

# Resume - PIC16F648 (Microchip)

- Program and data memory spaces are independent
- Program memory address space is linear
- Program memory word size is 13-bit
  - Custom design, general purpose memories are in 8-bit multiples
- Data memory address space is banked
  - ISA limitation: only 7-bit addresses can be generated (128 locations)
- Data memory cell size is 8-bit

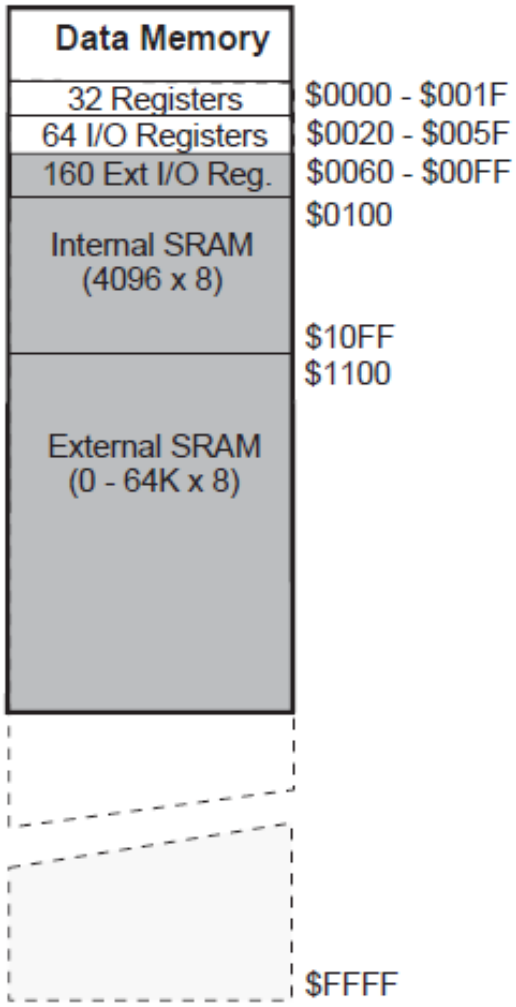
# Memory map example 8-bit microcontroller




16-bit words

ATmega128 (ATMEL)

# Memory map example 8-bit microcontroller




ATmega128 (ATMEL)

# Memory map example 8-bit microcontroller

## Register Summary

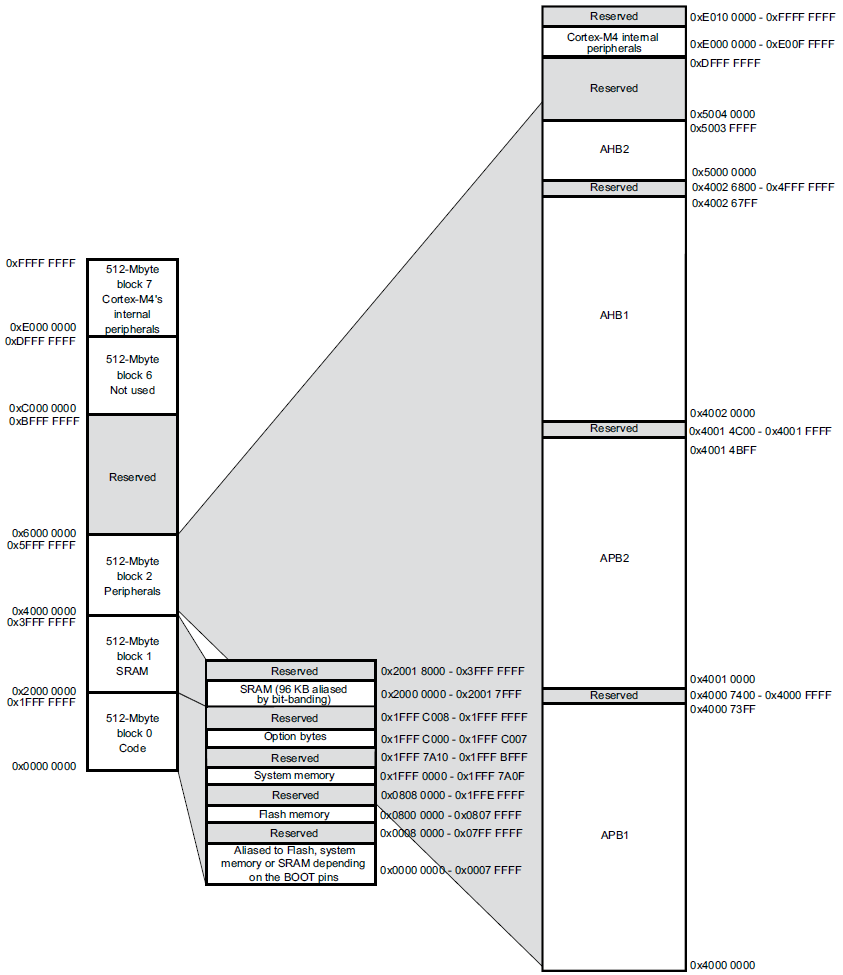
| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|---|---|---|---|---|---|---|---|---|---|---|
| ($FF) | Reserved | – | – | – | – | – | – | – | – | |
| .. | Reserved | – | – | – | – | – | – | – | – | |
| ($9E) | Reserved | – | – | – | – | – | – | – | – | |
| ($9D) | UCSR1C | – | UMSEL1 | UPM11 | UPM10 | USBS1 | UCSZ11 | UCSZ10 | UCPOL1 | 190 |
| ($9C) | UDR1 | USART1 I/O Data Register | | | | | | | | 188 |
| ($9B) | UCSR1A | RXC1 | TXC1 | UDRE1 | FE1 | DOR1 | UPE1 | U2X1 | MPCM1 | 188 |
| ($9A) | UCSR1B | RXCIE1 | TXCIE1 | UDRIE1 | RXEN1 | TXEN1 | UCSZ12 | RXB81 | TXB81 | 189 |
| ($99) | UBRR1L | USART1 Baud Rate Register Low | | | | | | | | 192 |
| ($98) | UBRR1H | – | – | – | – | USART1 Baud Rate Register High | | | | 192 |
| ($97) | Reserved | – | – | – | – | – | – | – | – | |
| ($96) | Reserved | – | – | – | – | – | – | – | – | |
| ($95) | UCSR0C | – | UMSEL0 | UPM01 | UPM00 | USBS0 | UCSZ01 | UCSZ00 | UCPOL0 | 190 |
| ($94) | Reserved | – | – | – | – | – | – | – | – | |
| ($93) | Reserved | – | – | – | – | – | – | – | – | |
| ($92) | Reserved | – | – | – | – | – | – | – | – | |
| ($91) | Reserved | – | – | – | – | – | – | – | – | |
| ($90) | UBRR0H | – | – | – | – | USART0 Baud Rate Register High | | | | 192 |
| ($8F) | Reserved | – | – | – | – | – | – | – | – | |
| ($8E) | Reserved | – | – | – | – | – | – | – | – | |
| ($8D) | Reserved | – | – | – | – | – | – | – | – | |
| ($8C) | TCCR3C | FOC3A | FOC3B | FOC3C | – | – | – | – | – | 135 |
| ($8B) | TCCR3A | COM3A1 | COM3A0 | COM3B1 | COM3B0 | COM3C1 | COM3C0 | WGM31 | WGM30 | 130 |
| ($8A) | TCCR3B | ICNC3 | ICES3 | – | WGM33 | WGM32 | CS32 | CS31 | CS30 | 134 |
| ($89) | TCNT3H | Timer/Counter3 – Counter Register High Byte | | | | | | | | 136 |
| ($88) | TCNT3L | Timer/Counter3 – Counter Register Low Byte | | | | | | | | 136 |
| ($87) | OCR3AH | Timer/Counter3 – Output Compare Register A High Byte | | | | | | | | 136 |
| ($86) | OCR3AL | Timer/Counter3 – Output Compare Register A Low Byte | | | | | | | | 136 |
| ($85) | OCR3BH | Timer/Counter3 – Output Compare Register B High Byte | | | | | | | | 137 |
| ($84) | OCR3BL | Timer/Counter3 – Output Compare Register B Low Byte | | | | | | | | 137 |
| ($83) | OCR3CH | Timer/Counter3 – Output Compare Register C High Byte | | | | | | | | 137 |
| ($82) | OCR3CL | Timer/Counter3 – Output Compare Register C Low Byte | | | | | | | | 137 |
| ($81) | ICR3H | Timer/Counter3 – Input Capture Register High Byte | | | | | | | | 137 |
| ($80) | ICR3L | Timer/Counter3 – Input Capture Register Low Byte | | | | | | | | 137 |
| ($7F) | Reserved | – | – | – | – | – | – | – | – | |

ATmega128 (ATMEL)

# Resume - ATmega128 (ATMEL)

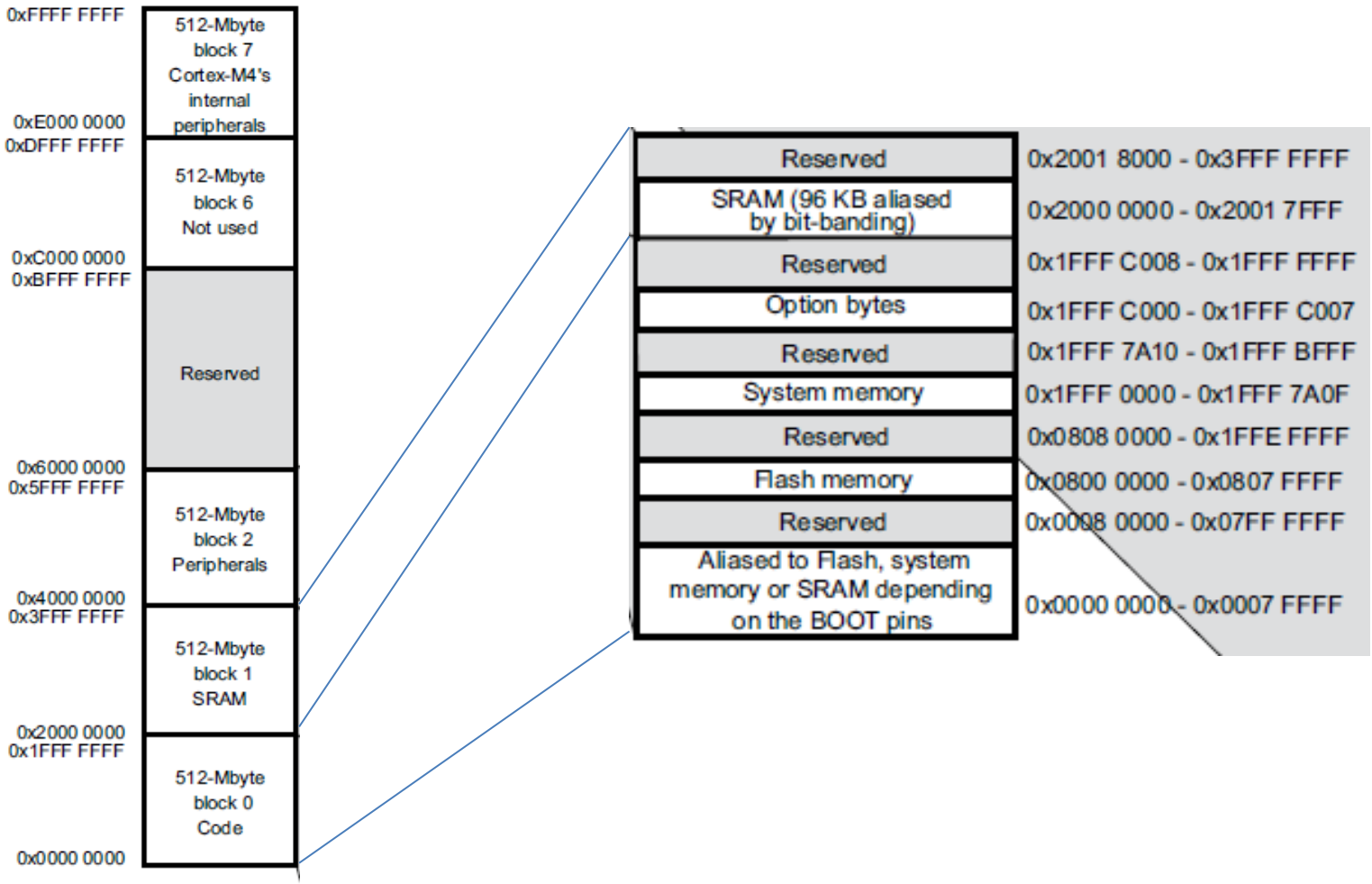
- Program and data memory spaces are independent
- Program memory address space is linear
- Program memory word size is 16-bit
- Data memory address space is linear
- Data memory cell size is 8-bit

# Memory map example 32-bit microcontroller



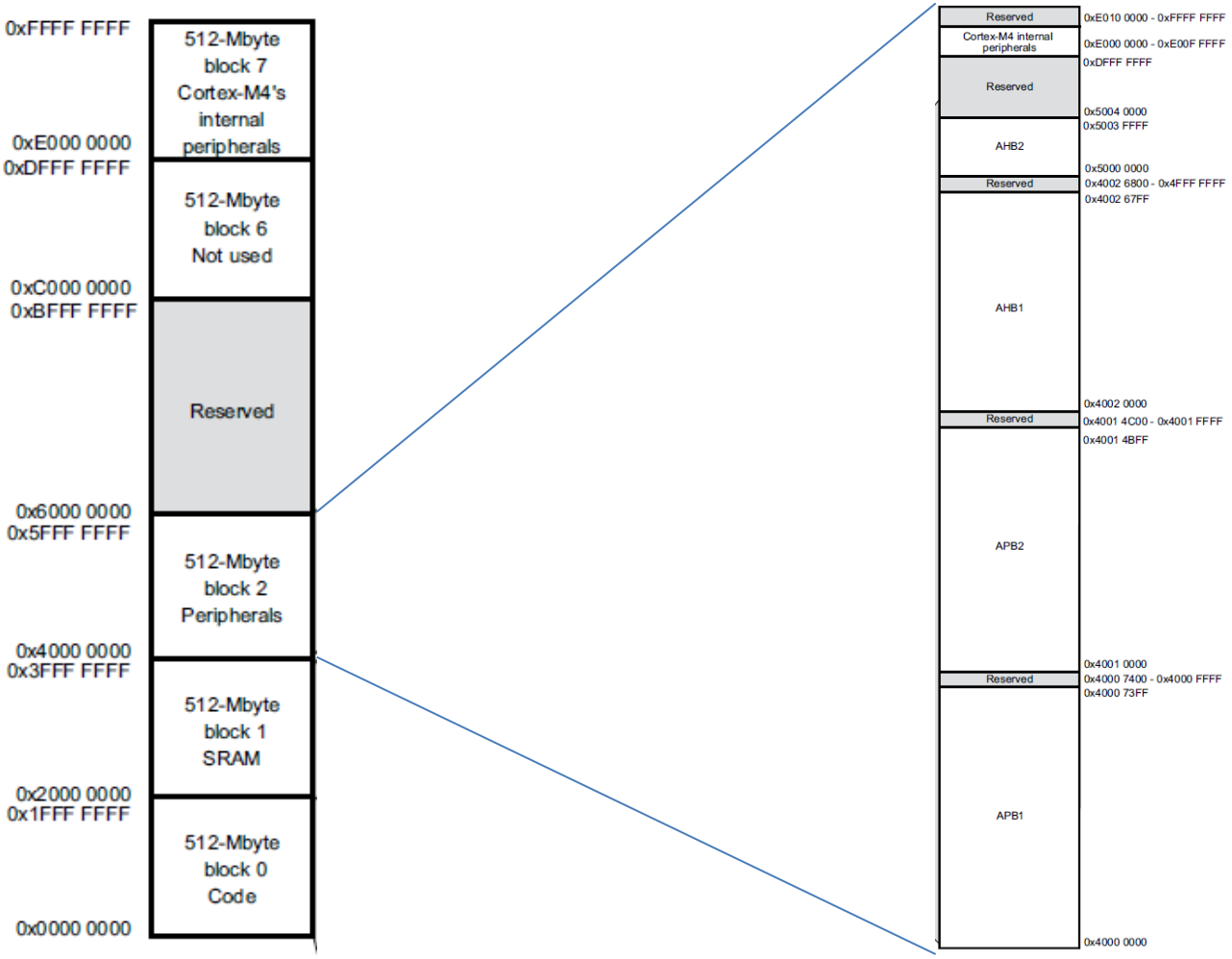

STM32F401 (ST Microelectronics)

# Memory map example 32-bit microcontroller




STM32F401 (ST Microelectronics)

# Memory map example 32-bit microcontroller




STM32F401 (ST Microelectronics)

# Memory map example 32-bit microcontroller

| Bus | Boundary address | Peripheral |
|---|---|---|
| APB2 | 0x4001 4C00- 0x4001 FFFF | Reserved |
| | 0x4001 4800 - 0x4001 4BFF | TIM11 |
| | 0x4001 4400 - 0x4001 47FF | TIM10 |
| | 0x4001 4000 - 0x4001 43FF | TIM9 |
| | 0x4001 3C00 - 0x4001 3FFF | EXTI |
| | 0x4001 3800 - 0x4001 3BFF | SYSCFG |
| | 0x4001 3400 - 0x4001 37FF | SPI4/I2S4 |
| | 0x4001 3000 - 0x4001 33FF | SPI1 |
| | 0x4001 2C00 - 0x4001 2FFF | SDIO |
| | 0x4001 2400 - 0x4001 2BFF | Reserved |
| | 0x4001 2000 - 0x4001 23FF | ADC1 |
| | 0x4001 1800 - 0x4001 1FFF | Reserved |
| | 0x4001 1400 - 0x4001 17FF | USART6 |
| | 0x4001 1000 - 0x4001 13FF | USART1 |
| | 0x4001 0800 - 0x4001 0FFF | Reserved |
| | 0x4001 0400 - 0x4001 07FF | TIM8 |
| | 0x4001 0000 - 0x4001 03FF | TIM1 |
| | 0x4000 7400 - 0x4000 FFFF | Reserved |

STM32F401 (ST Microelectronics)

# Resume - STM32F401 (STM)

- Program and data memory spaces are unified
- Address space is linear, word size is 32-bit
- Bit banding (ARM Cortex specific)
  - Bit banding regions map each word in an alias region of memory to a bit in a bit-band region of memory. Writing to a word in the alias region has the same effect as a read-modify-write operation on the targeted bit in the bit-band region.
  - Allows atomic read-modify-write operation without specialized instruction and hardware

# Bit banding

A mapping formula shows how to reference each word in the alias region to a corresponding bit in the bit-band region. The mapping formula is:

*bit_word_addr* = *bit_band_base* + (*byte_offset* x 32) + (*bit_number* × 4)

where:

– *bit_word_addr* is the address of the word in the alias memory region that maps to the targeted bit

– *bit_band_base* is the starting address of the alias region

– *byte_offset* is the number of the byte in the bit-band region that contains the targeted bit

– *bit_number* is the bit position (0-7) of the targeted bit

**Example**

The following example shows how to map bit 2 of the byte located at SRAM address 0x20000300 to the alias region:

0x22006008 = 0x22000000 + (0x300*32) + (2*4)

Writing to address 0x22006008 has the same effect as a read-modify-write operation on bit 2 of the byte at SRAM address 0x20000300.