# ELC4438: Embedded System Design

## Embedded Processor

Liang Dong

Electrical and Computer Engineering

Baylor University

# 1. Processor Architecture

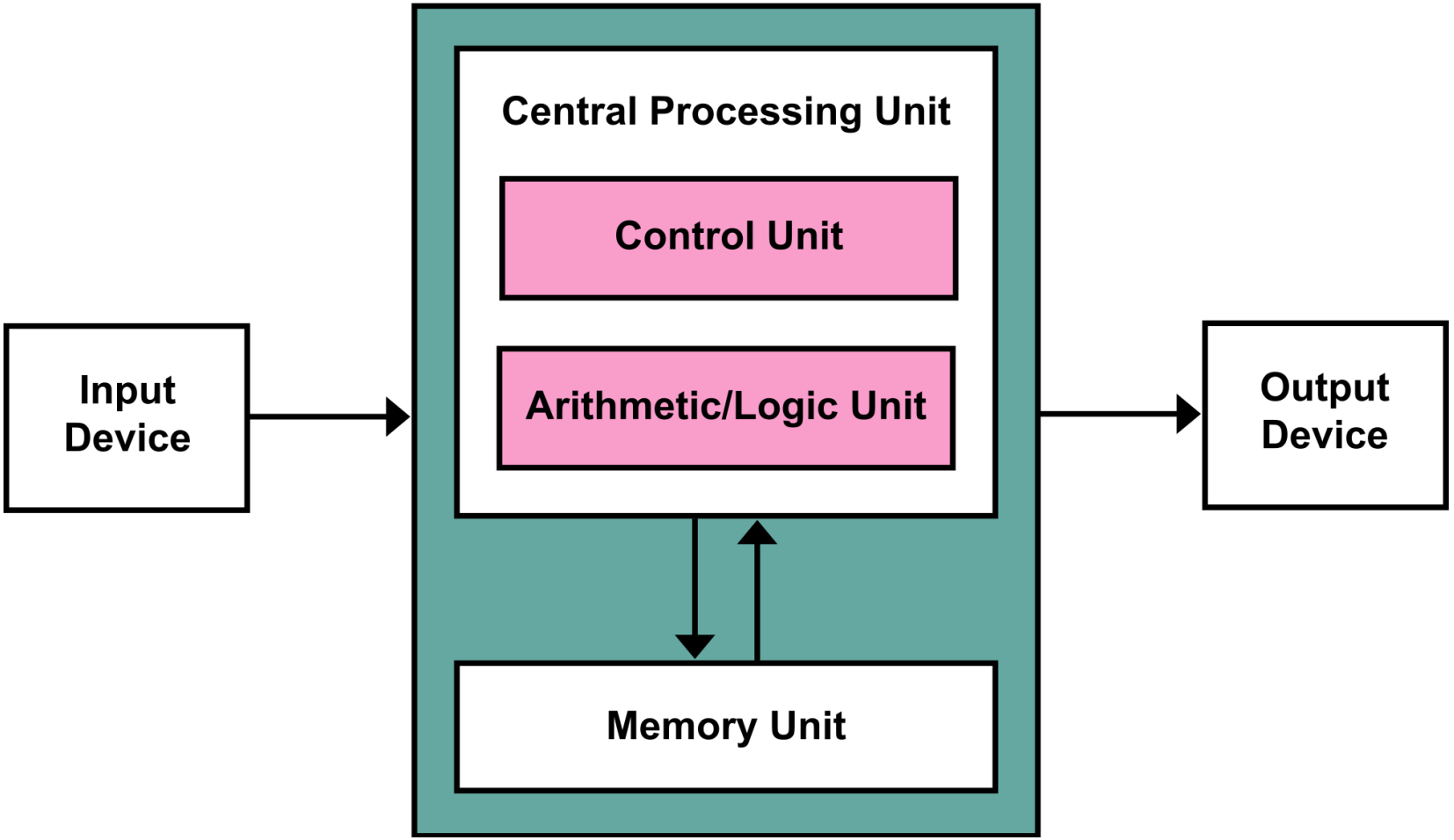
- General PC – Von Neumann Architecture
  - a.k.a. Princeton Architecture

- Embedded Processor – Harvard Architecture
  - Harvard Physicist A. Howard 1930

# Von Neumann Architecture

- Program memory and data memory are one
- Program address and data address point to different physical positions of the same memory
- The width (wordlength) of program instruction are the same as the width of data
- A unified bus

# Von Neumann Architecture



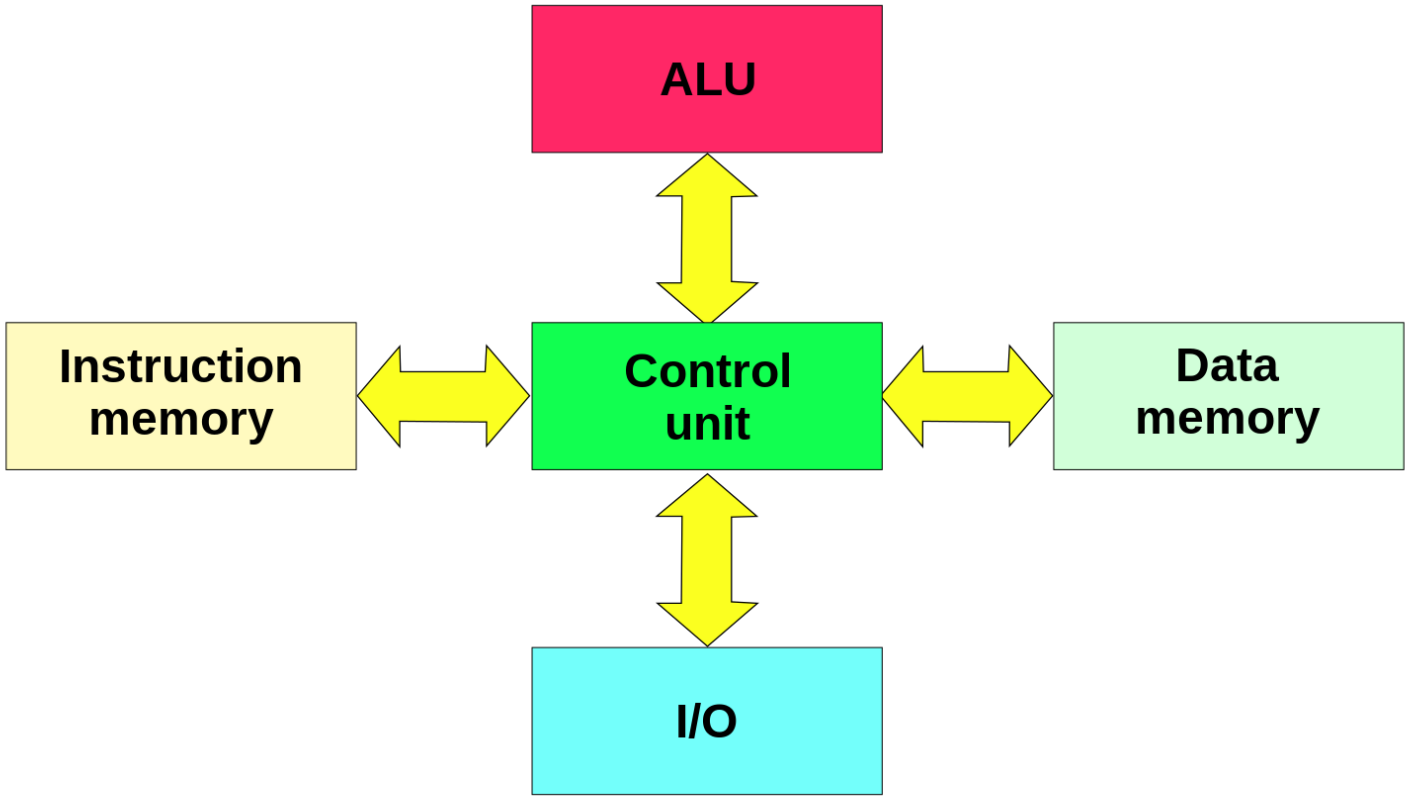
# Von Neumann Architecture

- Main contribution of Von Neumann – Introduced the concept of “stored instructions” – “program”
- Instruction and data are similar binary codes – naturally, instruction and data share the same bus
- Pros: Unified addressing, reduced I/O interface
- Cons: The flow of information becomes the bottleneck

# Harvard Architecture

- Program and data are in different memories
- Independent addressing for each memory
- Processing unit has both program/instruction bus and data bus
- Efficient instruction fetching and execution
- Pipelining is implementable – simultaneous execution and pre-read the next instruction

# Harvard Architecture



# Harvard Architecture

- Intel 8051 microcontroller series
- Motorola MC68
- Zilog Z8
- ATMEL AVR
- ARM ARM9, ARM10, ARM11, Cortex-M3, Cortex-M4, etc.

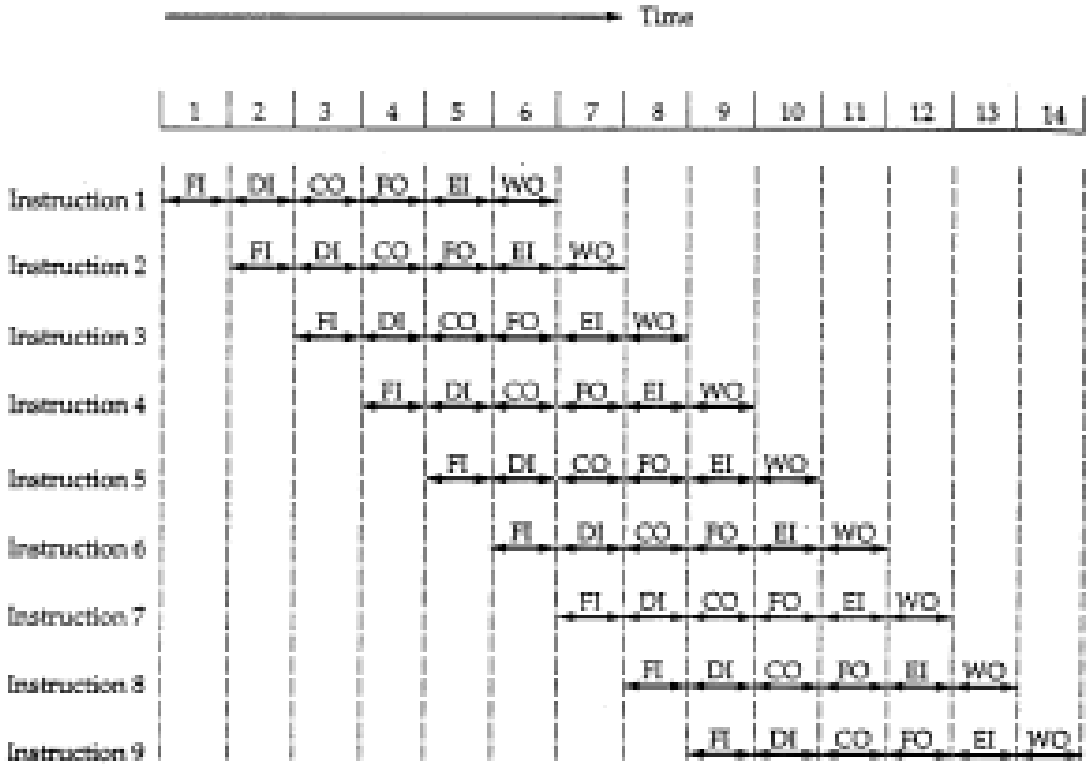
# Harvard Architecture

- No conflict between instruction fetching and data reading
- Advanced Harvard Arch.:
  - data in program memory for algorithmic instruction
  - Instruction in high-speed buffer instead of memory

# 2. CISC and RISC

- The complexity of the instruction set is an important parameter of CPU
- Complex instruction set computing (CISC) vs. Reduced instruction set computing (RISC)
- CISC: ease programming, increase addressing efficiency of memory
- RISC: increase processing speed
  - Key technology: pipelining

# Pipelining




- Fetch Instruction (FI)
- Decode Instruction (DI)
- Calculate Operands (CO)
- Fetch Operands (FO)
- Execute Instruction (EI)
- Write Operand (WO)

# CISC and RISC

- CISC: Intel 80x86, Motorola 68K series, etc.

- RISC: *non-x86* microprocessor, ARM (today's most advanced 32-bit RISC)

# CISC

- Pros:
  - Rich instructions → easy programming
  - Instruction length is not specified → save memory space
  - Direct operation of registers → reduce the number of registers
- Cons:
  - Multiple addressing methods and instruction formats, variant-length instructions → increase hardware complexity and design cost
  - Complex instructions and long cycles → difficult to compile into high-efficiency machine language
  - Some instructions have low frequency of usage → lower performance-price ratio

# RICS

- In general, a few tens of instructions
- Reduce hardware complexity and increase software flexibility and intelligence
- Features
  - Reduced instruction set (simple action, one clock cycle); complex procedure uses subroutine/function instead of (complex) instruction
  - Instruction lengths are the same
  - Most instructions execute in one clock cycle

# RISC

- Pros:
  - Simple controller hardware for decoding and execution → reduce footprint area, VLSI
  - High processing speed
  - Directly support high-level language, simplify compiling program
  - Reduce design cycle comp. w. CISC

# RISC

- Cons:
  - Increase the burden of assembly programmer; Increase the size of machine-language program → more memory
  - High requirement on the complier → Rely on the efficiency of the compiler

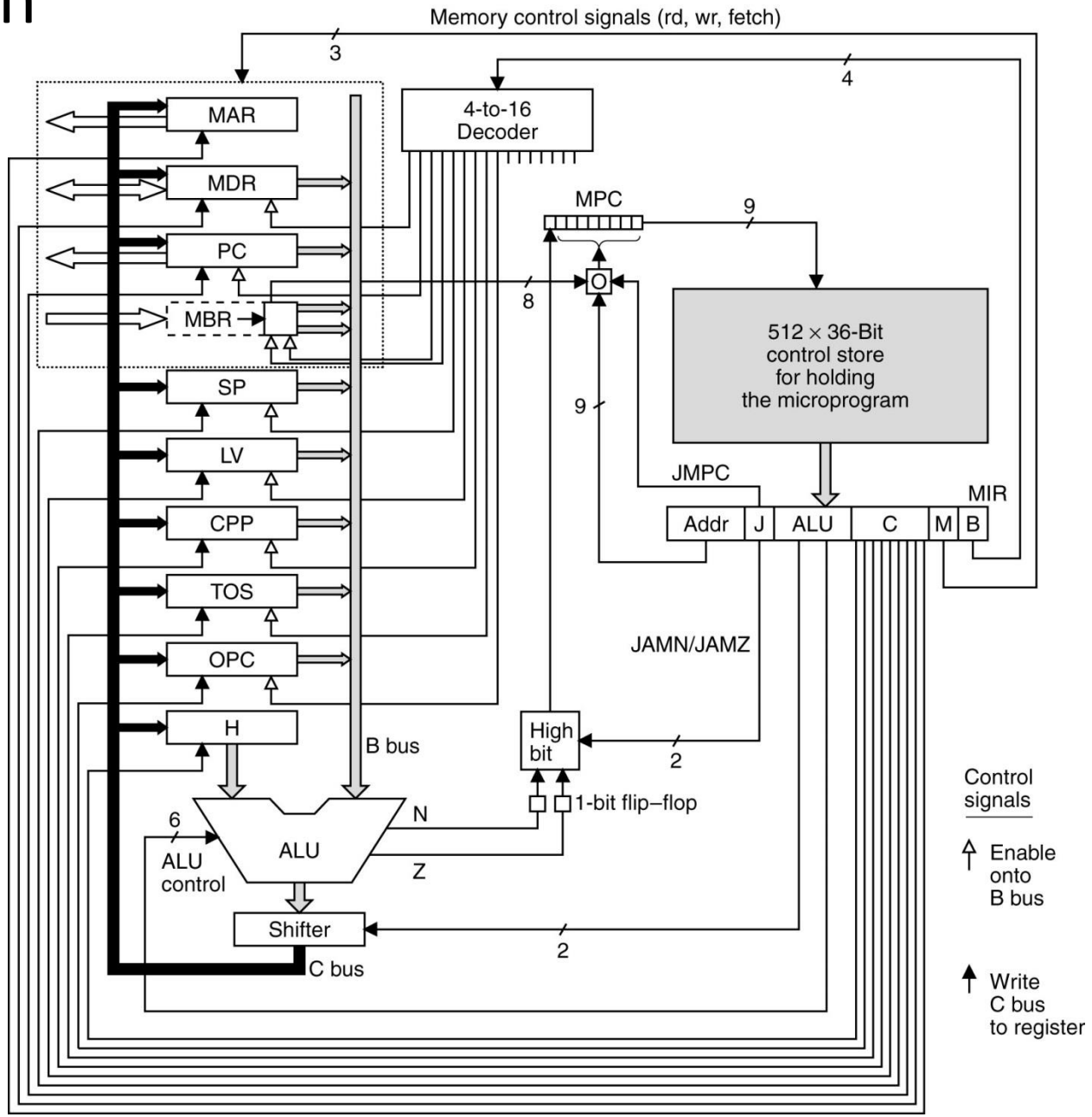
# RICS Summary

- Instruction: 90% directly accomplished by hardware, 10% software combination
- Addressing: simple and short cycle
- Instruction cycle: pipelining → one instruction per clock cycle
- High usage of registers: instructions execute data among registers → increase processing speed

# 3. x86 and ARM Instruction Sets

- Intel developed x86 instruction sets for its 16-bit CPU i8086
- x86 Instruction Set Features
  - 8 general-purpose registers → most of the time, CPU accesses data in memory not in register
  - Decode Unit. Decode various-length instruction into fixed-length instructions. Hardware decoding and micro decoding
  - Addressing range is limited

# Microinstruction Control:



# ARM Instruction Set

- RICS. Simplified hardware and increased processing speed. Complex action compiled by a few simple instructions
- Features
  - Low volume, low energy, low cost
  - Support Thumb (16-bit) and ARM (32-bit) dual instruction sets
  - Use many registers → increase processing speed
  - Simplified addressing
  - Fixed instruction length, pipelining

# ARM Instruction Sets

- PC uses x86 instruction sets for compatibility
- Large servers no longer use CISC
- The most popular embedded processor today: ARM processor
- Pentium processor adopts some RISC structures in its internal implementation

# 4. Classification of Embedded Processors

1. Micro Controller Unit (MCU)
2. Micro Processor Unit (MPU)
3. Digital Signal Processor (DSP)
4. System on Chip (SoC)

# Micro Controller Unit (MCU)

- Single-chip machine
- Chip has peripherals such as bus, bus logic, flash, RAM, timer, counter
- Only needs clock, power circuitry to form an embedded system
- Low volume, low power, reliable, low cost
- Most shares in the embedded system market

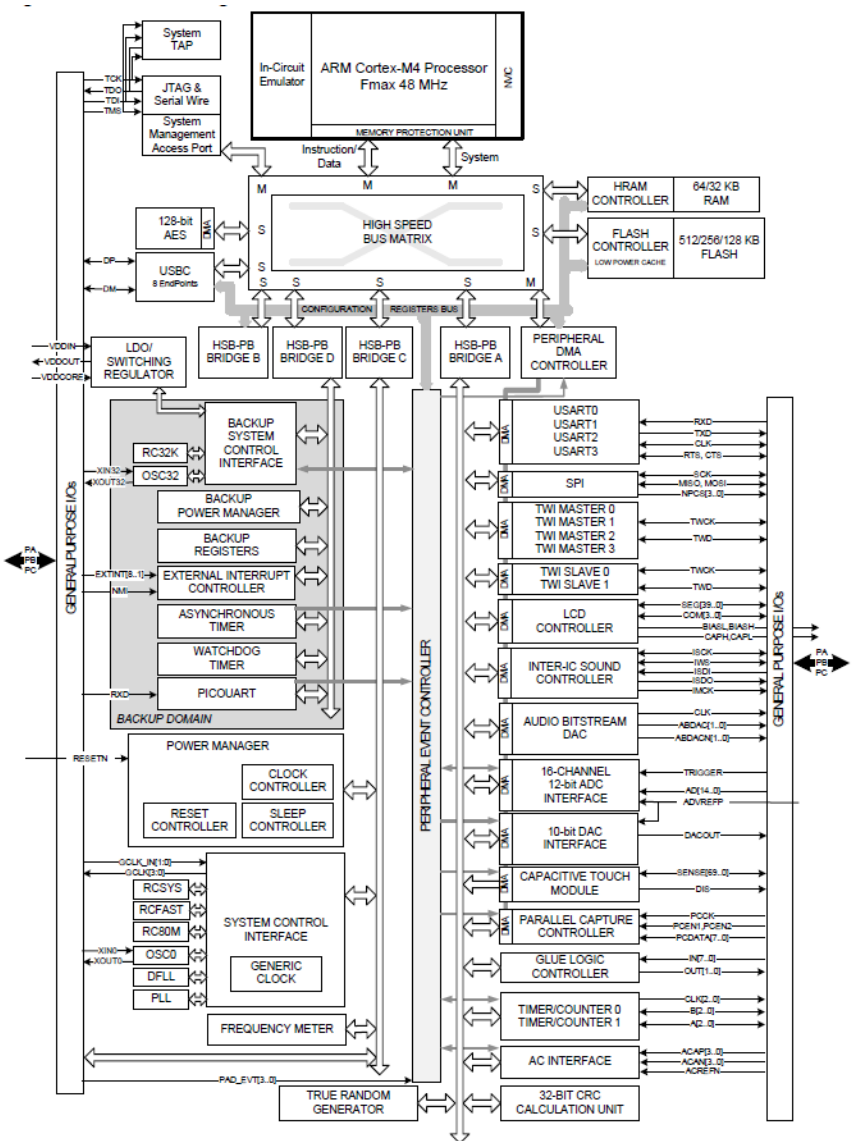
# Micro Controller Unit (MCU)

- MC51
- Atmel 8-bit AVR
- Microchip 8-bit PIC
- TI MSP430, CC2430 (Zigbee)

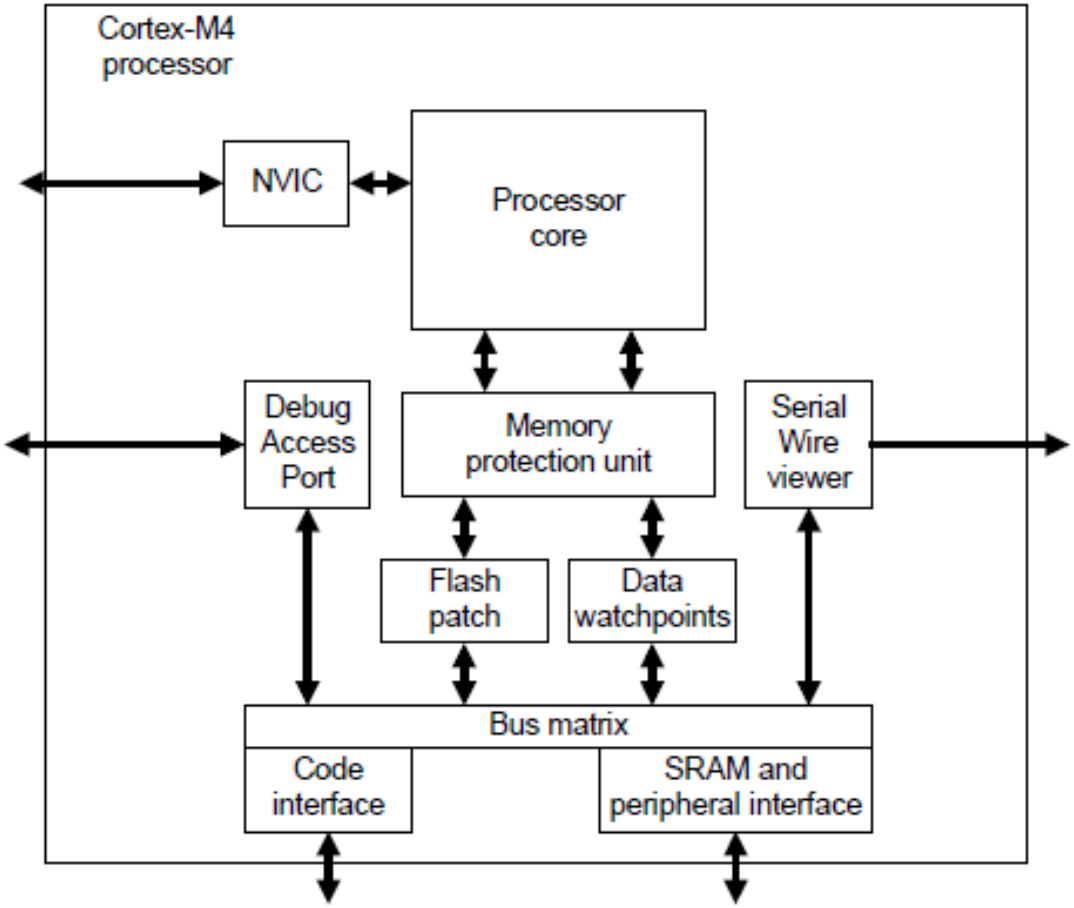
# Micro Processor Unit (MPU)

- Evolved from general-purpose CPU (>= 32bit)
- High performance
- Only keeps relevant functional components of CPU
- Typical example: ARM (Advanced RISC Machine) processor
- MIPS (1984 Stanford spin-off company)
- PowerPC (IBM, Motorola, ~~Apple~~)

# Atmel SAM4L (32-bit ARM Cortex-M4 RISC Processor)



# ARM Cortex-M4



# Digital Signal Processor (DSP)

- Specifically designed for signal processing
- Appropriate for implementing DSP algorithms, e.g. digital filtering, FFT
- 1978 First single-chip DSP. Ten times faster than MCU. Voice synthesis, coding and decoding

# Digital Signal Processor (DSP)

- Parallel processing of multiple operations
- Support pipelining
- Fast interrupt service
- One multiplexing one addition in one clock cycle
- Separation of program and data
- Fast on-chip RAM
- Hardware support for low/no-overhead repeat and jump
- Multiple hardware address generator (in a single clock cycle)

# System on Chip (SoC)

- Seamless integration of software and hardware
- Embed operating system code module in chip
- Use hardware description language (HDL) to implement a complex system on chip
- Low volume, low energy, high reliability