Introduction to Embedded Systems

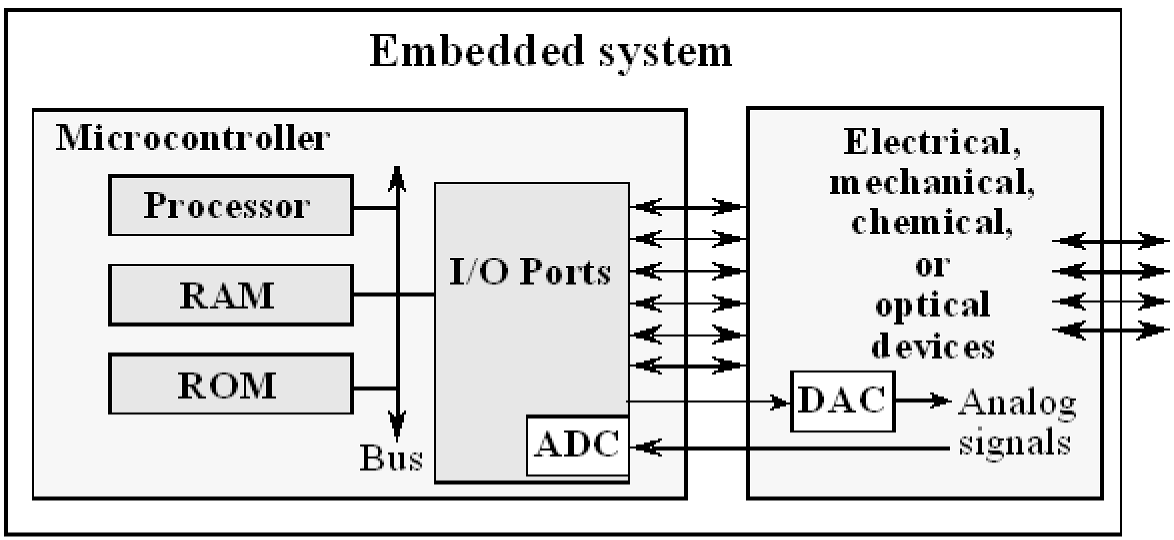
Cuauhtémoc Carbajal

ITESM CEM

13/08/2013

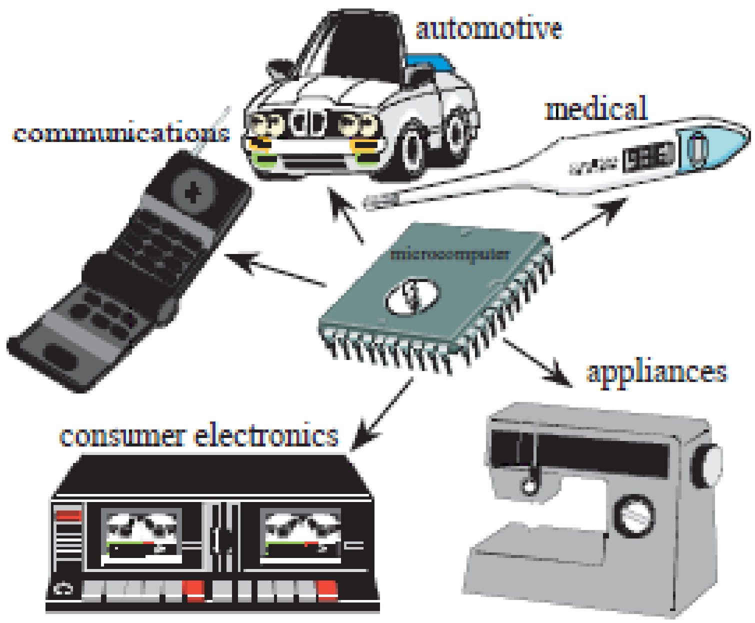
Embedded Systems

- *Embedded*
  - “hidden inside so one can’t see it”
  - *computer/processor*



Embedded Systems

- An *embedded* ***computer*** *system* includes a microcomputer
  - mechanical, chemical, and electrical devices
  - specific dedicated purpose, and
  - packaged as a complete system
- Applications
  - communications
  - automotive
  - military
  - medical
  - consumer
  - machine control



Embedded Systems

- An *embedded microcomputer* system
  - accepts inputs
  - performs calculations
  - generates outputs
  - runs in "*real time*"
- A *real time system*
  - specifies an upper bound on the time required to perform the input/calculation/output response to external events

What is a microprocessor?

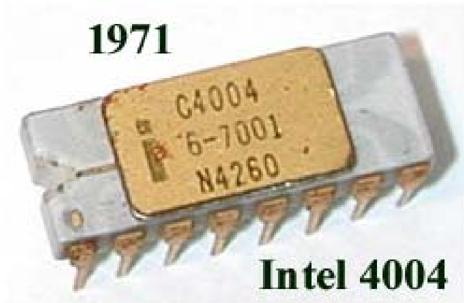
The microprocessor is the integration of a number of useful functions into a single IC package:

1. The ability to execute a stored set of instructions to carry out user defined tasks.

2. The ability to be able to access external memory chips to both read and write data from and to the memory.

http://data.bolton.ac.uk/learningresources/elearning/moodle/ami4655_micros/u01/micro01hist.html

History


1971
C4004
6-7001
N4260
Intel 4004


1974
Intel 8080


1974
Motorola 6800

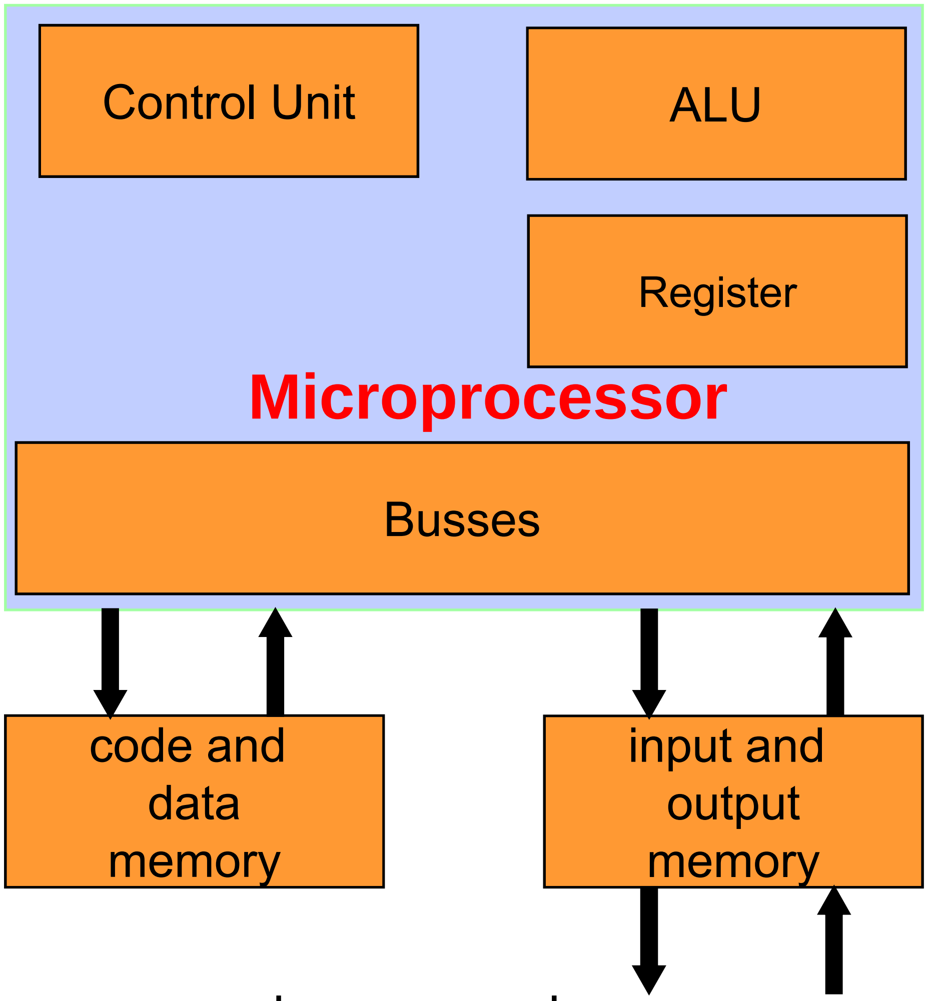

1978
MC6809L
CW38249
Motorola 6809

A Brief History of Microprocessors

- The first microprocessor was developed by what was then a small company called Intel (short for Integrated Electronics) in the early 1970s. The chipset was a success and within a short while Intel developed a general purpose 4 bit microprocessor called the 4004.
- In 1974 the more powerful second generation microprocessor (the 8008) was announced fabricated as a single chip. This was quickly followed by the Intel 8080.
- At about the same time Motorola released its first microprocessor, the 6800, which was also an 8 bit processor with about the same processing power as that of the Intel 8080.
- The architectures used in the Intel 8080 and the Motorola 6800 were very different.
- In due course the Intel 8080 core processor was used for a range of microcontrollers (8048 and 8051 to name but two).
- Motorola followed in a similar vein with a range of microcontrollers based on the 6800 (6805, 6808, 6811 which survive to this day).

http://mic.unn.ac.uk/miclearning/modules/micros/ch1/micro01hist.html

Block Diagram



- Note:
  - Modern microprocessors have a much finer granularity and sometimes parallel units.
  - However, the basics are still very much the same.

Arithmetic Logic Unit

- Arithmetic Logic Unit (ALU) calculates arithmetical and / or logical functions:
- At least:
  - Arithmetical: Addition (ADD)
  - Logical: Negation (NEG)
    Conjunction (AND)
- Typical:
  - Arithmetical: Subtraction (SUB)
    Multiplication (MUL)
  - Logical: Comparison (CMP)
    Disjunction (OR)
    Antivalence (EXOR)
  - Miscellaneous: Right- and Left Shift (ASR,ASL)
    Rotation (ROL, ROR)
    Register-Bit-Manipulation (set, clear, toggle, test)

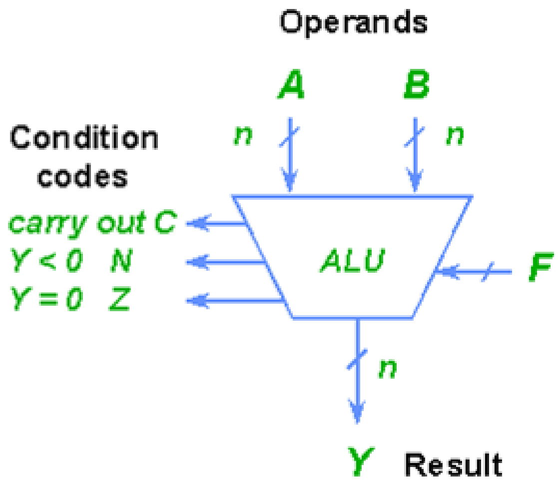
Arithmetic Logic Unit (cont.)

- An ALU is able to process two binary values with equal length (N)
  $\rightarrow$ N-Bit ALU with N = 4, 8,16, 32 or 64
- Most ALUs process Fixed Point Numbers
- A few ALUs, used especially those in Digital Signal Processors and desktop processors are capable of operating on both Floating Point Numbers and on Fixed Point Numbers.

Example: a simple ALU structure

Arithmetic/Logic Unit (ALU)

Purely combinational logic

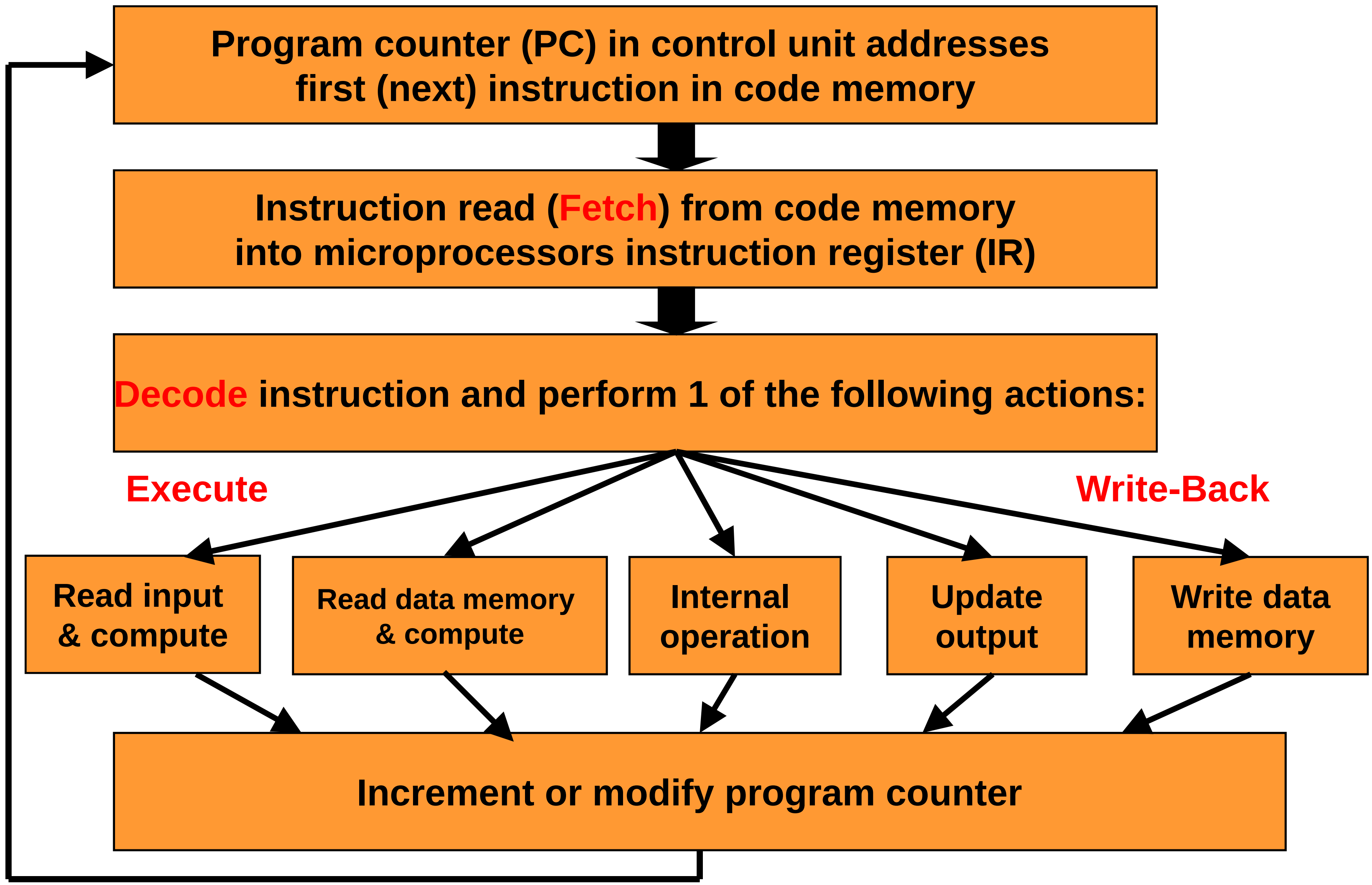


F	Y
000	A + B
001	A - B
010	A - 1
011	A and B
100	A or B
101	A * B
.	
.	

A, B, Y: Internal registers
F: Functional code
C: Carry – Bit
N: Negative – Bit
Z: Zero - Bit

- Note:
  - Most ALUs will generate a size of 2*n for register Y in case of a multiply operation Y = A * B
  - ALUs are also available as standalone ICs:
    - SN74LS181

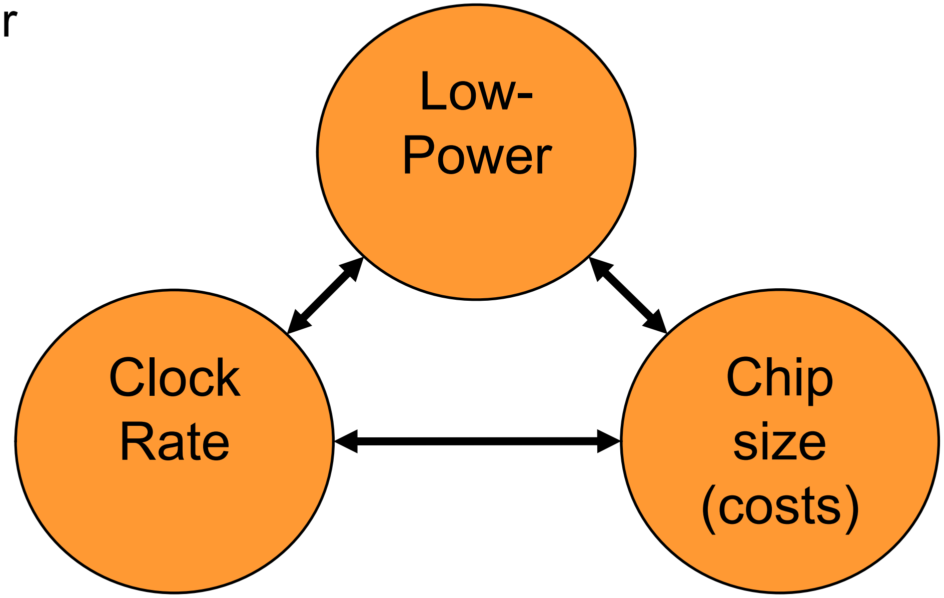
Control Unit: execution flow



Important selection features

- Word length:
  - Typical 16 or 32 bits
  - Important feature for performance
- Clock Cycles:
  - Million Instructions Per Second (MIPS)
  - Cycles Per Instruction (CPI)
  - depends on architecture
- Clock frequency [Hz] (f_{CLK}):
  - Frequency of an crystal oscillator
- Low-Power (CMOS):
  - $P = \sigma \cdot f_{CLK} \cdot C_L \cdot V_{DD}^2$
    - σ: switching activity
    - f_{CLK}: clock frequency
    - C_L: load capacitance
    - V_{DD}: supply voltage
  - Important for longer battery life
- Architecture:
  - Von Neumann, Harvard
- Instruction set:
  - CISC, RISC

Embedded System trade-offs



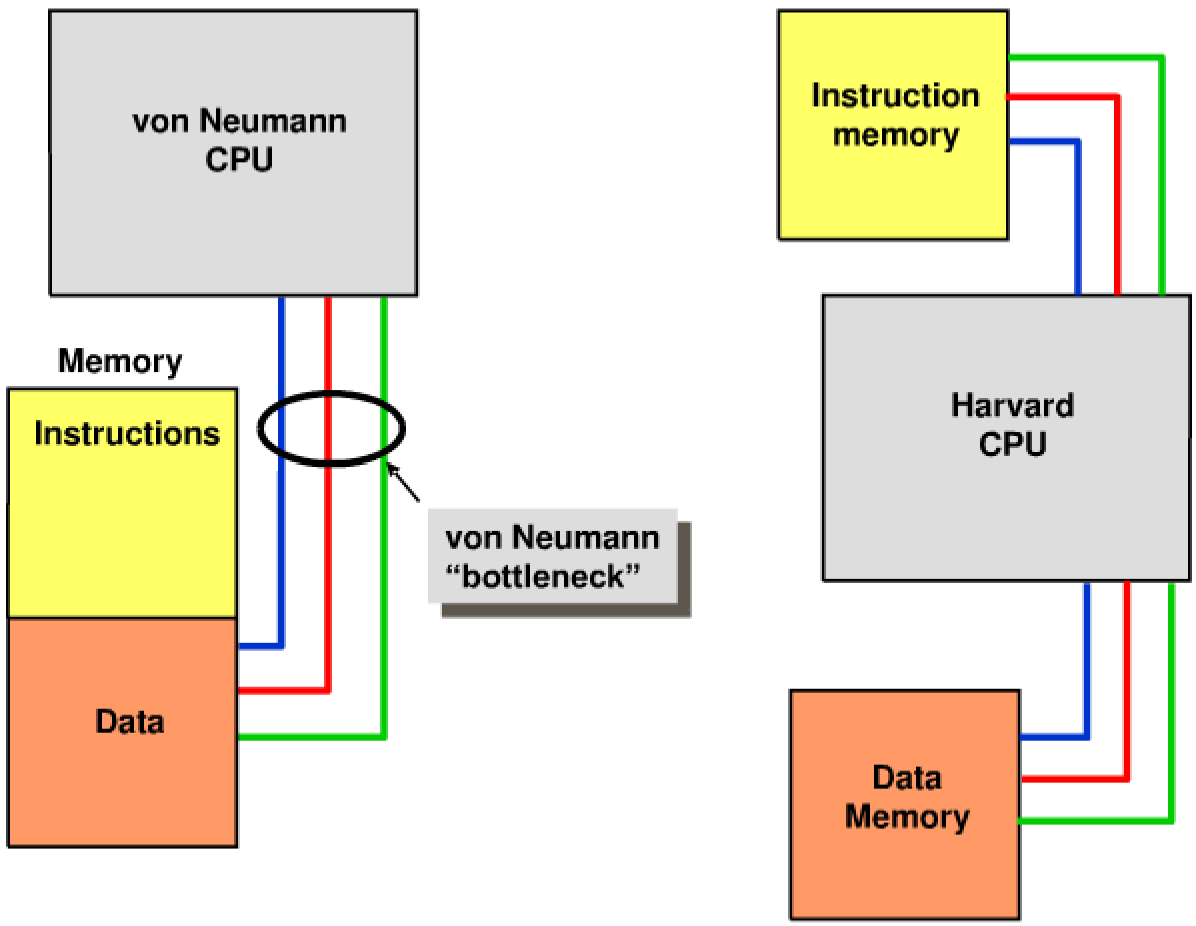
History

- First Microprocessor – Intel 4004 [5]:
  - Production start: 1971
  - Complexity: approx. 2,300 transistors; today: > 1,000,000,000 transistors
  - Integration: gate number
    - < 100 Small Scale Integration (SSI)
    - > today: 1 million gates
  - Clock rate (f_{CLK}): < 1 MHz; today: > 4 GHz
  - Word length: 4 bits; today > 64 bits
- First Microcontroller – TI TMS1000
  - Production start: 1974
  - Clock rate: 0.4 MHz
  - Word length: 4 bits
- Note: Typical features for Embedded Systems.
  - Clock rate: 100 MHz
  - Word length: 32 bits

Architectures

- Two basic microprocessor architectures:
  - "Von Neumann"- Architecture
  - "Harvard" - Architecture
- "Von Neumann" - Architecture:
  - Shared memory space between code and data
  - Shared memory busses between code and data
  - Example: typically microcontrollers such as the HCS12
- "Harvard" – Architecture:
  - Two independent memory spaces for code and data
  - Two memory bus systems for code and data
  - Example: typically Digital Signal Processors (DSPs) such as the TI C2000, C5000 and C6000 family

Architectures

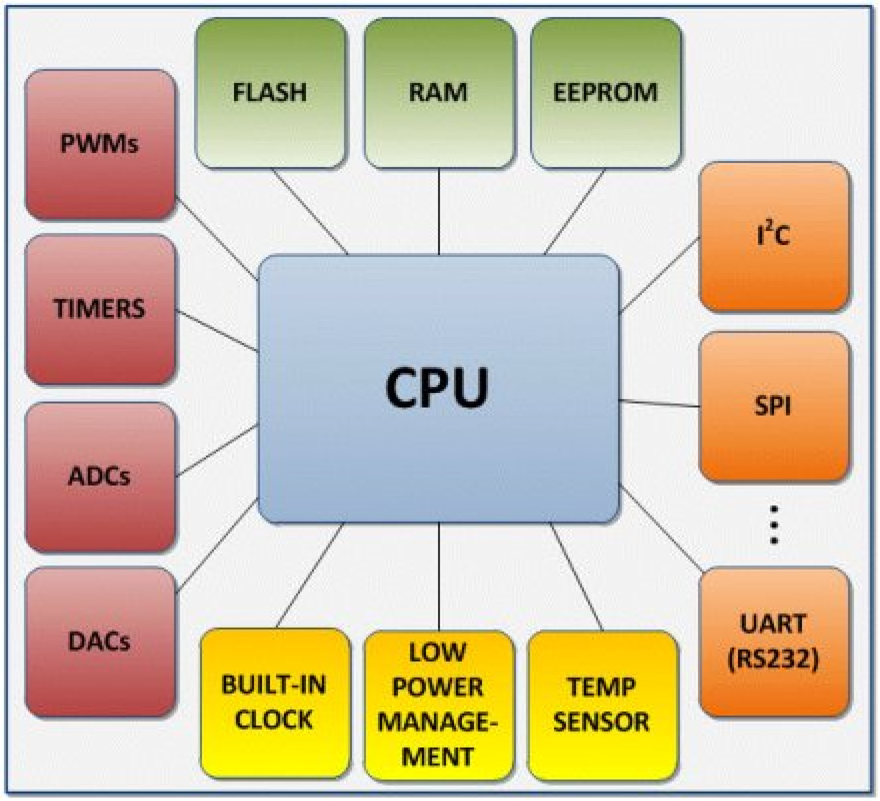


Buses:
- Data
- Address
- Control

CISC/RISC

- Complex Instruction Set Computer (CISC)
  - Between 1971 until ≈ 1980 favoured architecture for general purpose processors
  - Extensive and complex instructions sets
- Reduced Instruction Set Computer (RISC)
  - Since 1980
  - Features
    - Single cycle instructions: one instruction per clock
      - CPI=1; Clock Cycle per Instruction (CPI)
    - Uniform instructions: all instructions have the same format
    - Load/Store architecture: only a few commands have memory-access
    - High-level languages support: architectures and compilers are co-coordinated
- Note: Today's Microprocessor architectures have the advantages of both CISC and RISC – these architectures are called Hybrid Architectures

What is a microcontroller?



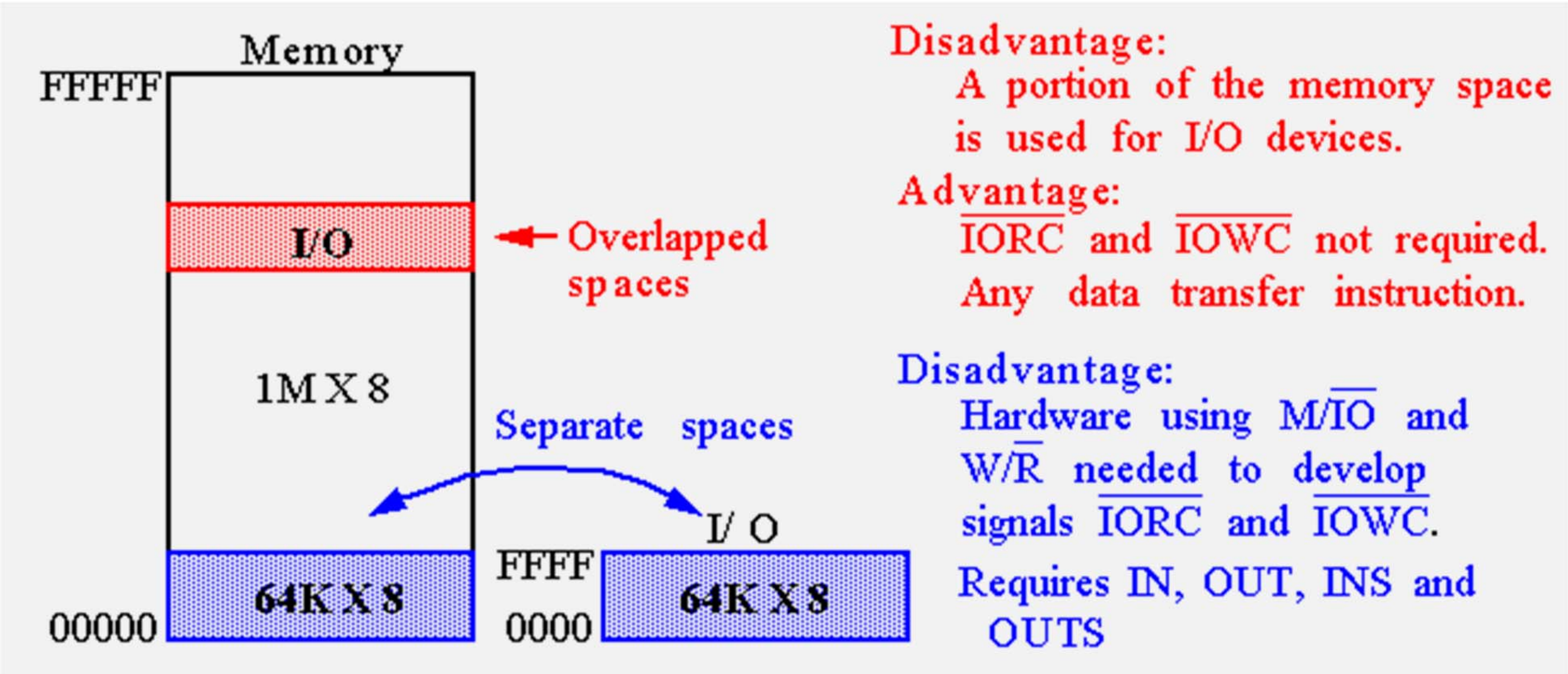
A microcontroller combines onto the same microchip :

1. The CPU core
2. Memory (both ROM and RAM)
3. Peripherals

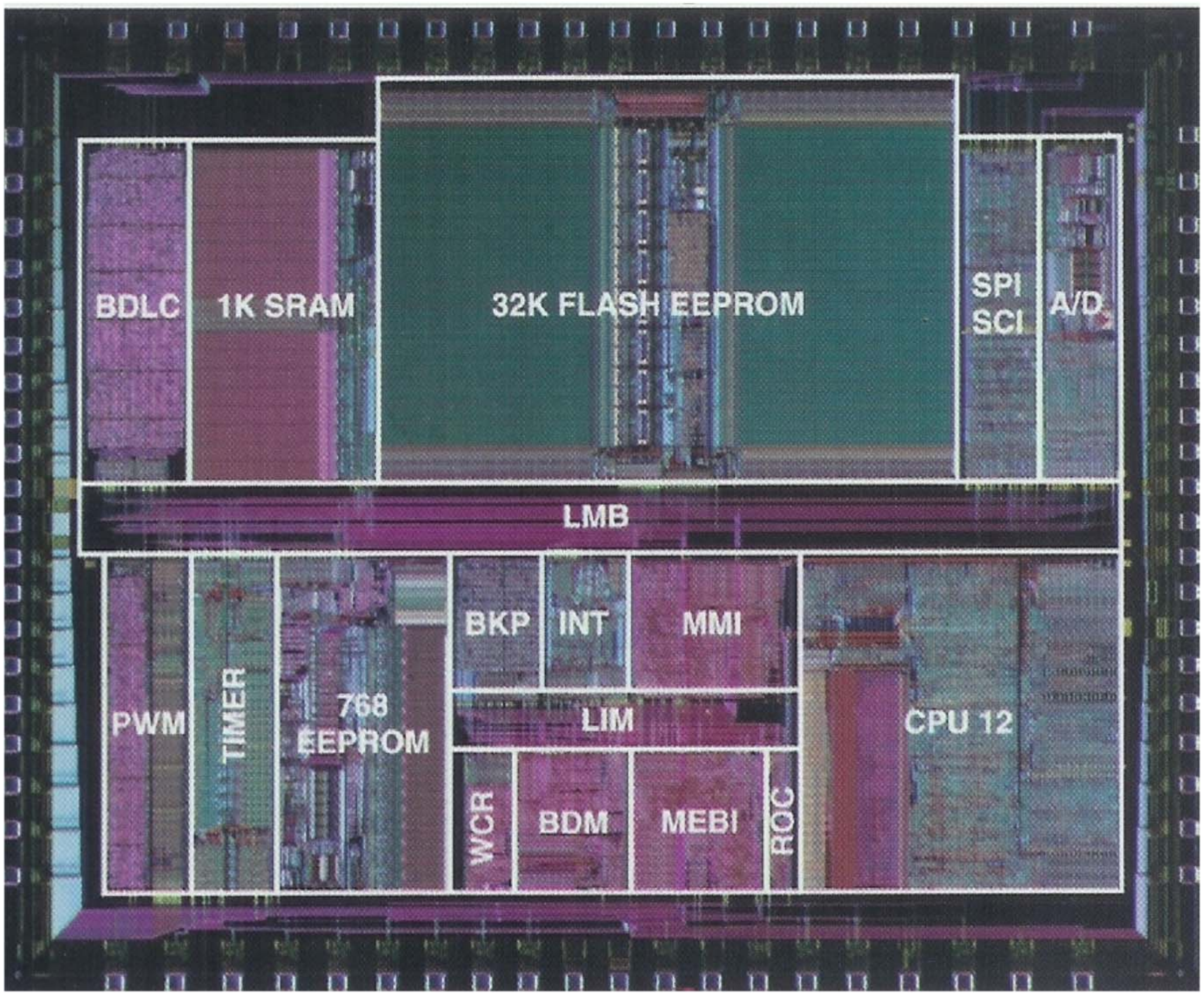
Isolated versus Memory-Mapped I/O

- Memory-mapped I/O
  - I/O ports/registers appear as addresses on common bus with memory
  - I/O ports/registers are accessed as though they are locations in memory
  - Employed on the STM32 microcontrollers
- Isolated I/O
  - I/O ports/registers have separate control signals from those used with memory
  - Special instructions are used to access I/O ports/registers
  - Employed on Intel x86 processors

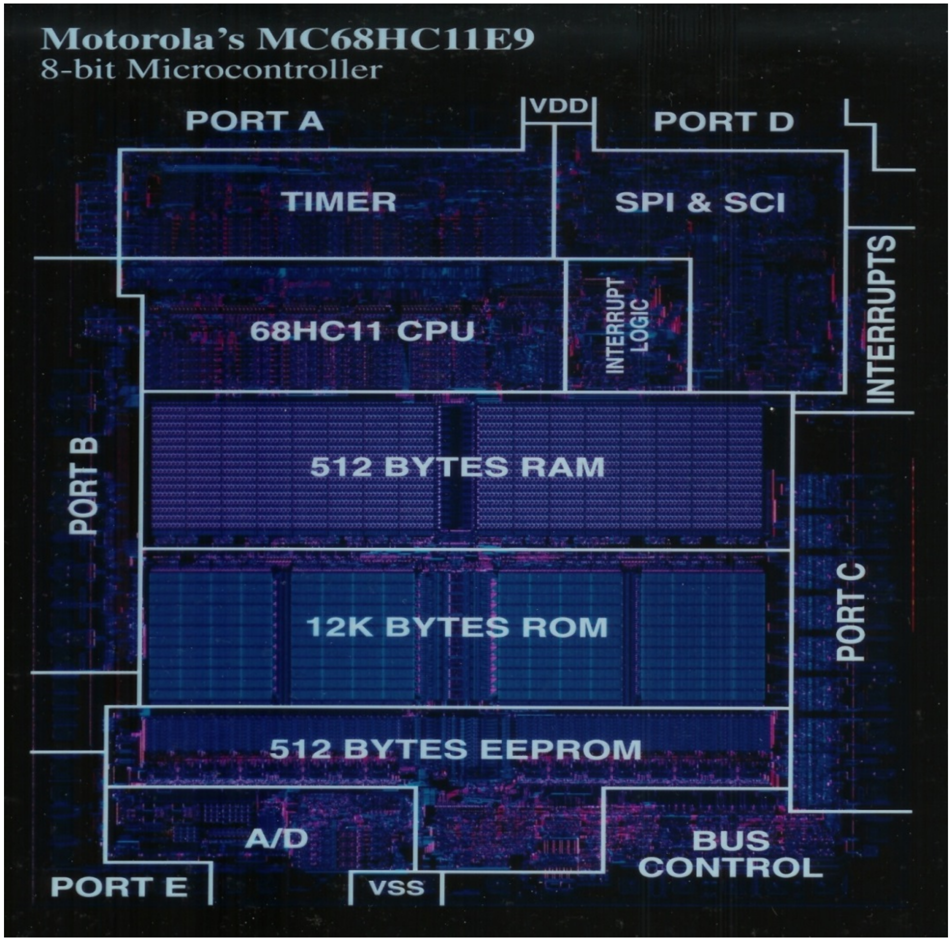
Isolated versus Memory-Mapped I/O



What is a microcontroller?



What is a microcontroller?



Keys to a successful education

- "*It is important that students bring a certain ragamuffin barefoot irreverence to their studies, they are here not to worship the known but to question it*" – Jacob Bronowski

Important take-aways

- *You are a professional*
- *Ethics matter*
- *Learn how to read*
- *Learn how to work efficiently*
- *Learn how to ask questions*
- *Be a good partner*
- *Learn how to debug*