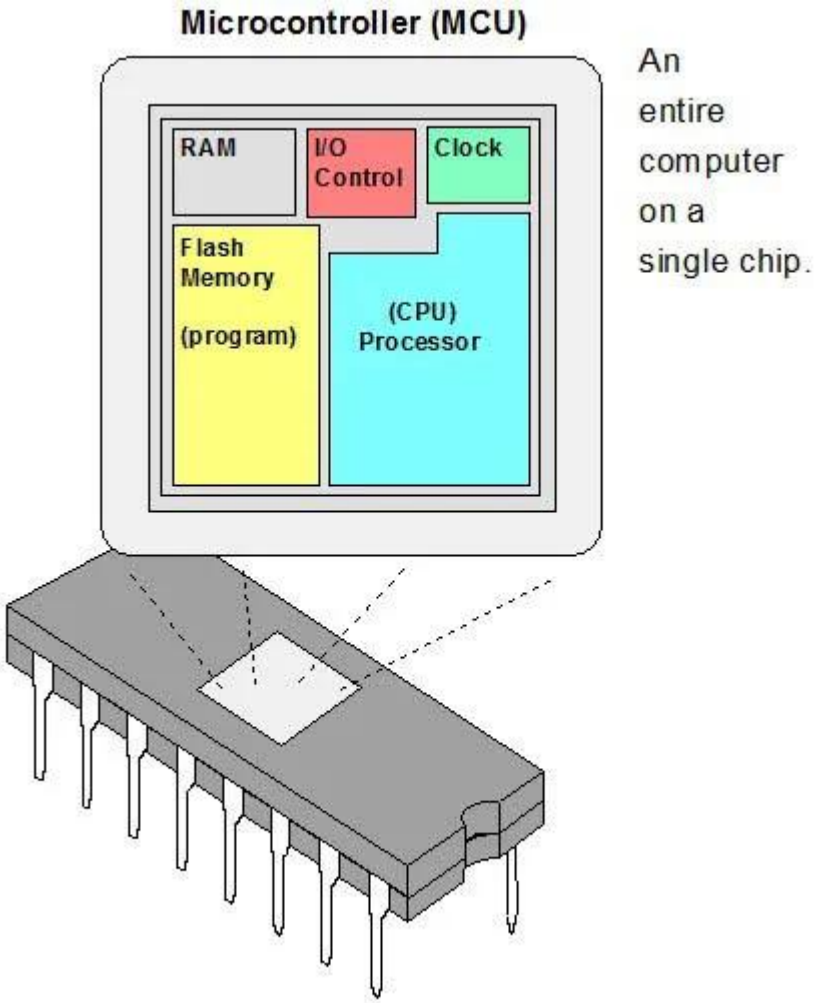
INTRODUCTION TO MICROCONTROLLERS

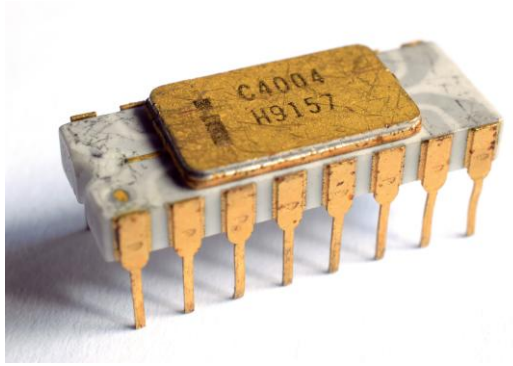
Definition of a microcontroller

- Small, self-contained computer on a single chip
- Helps to change the function without changing the electronic circuit
- Contains CPU, memory (RAM, Flash), and I/O peripherals
- Designed to perform specific tasks in embedded systems



Historical background

- 1971: 4-bits microcontroller – Intel 4004 used in calculators by a Japanese company named BUSICOM
- 1973: 12-bits microcontroller – Toshiba TLCS-12 used in Ford’s cars as engine controller
- 1980s–90s: Rise of 8051, PIC, and AVR families
- 2000s–Present: Dominance of ARM Cortex-M, IoT expansion




TLCS-12

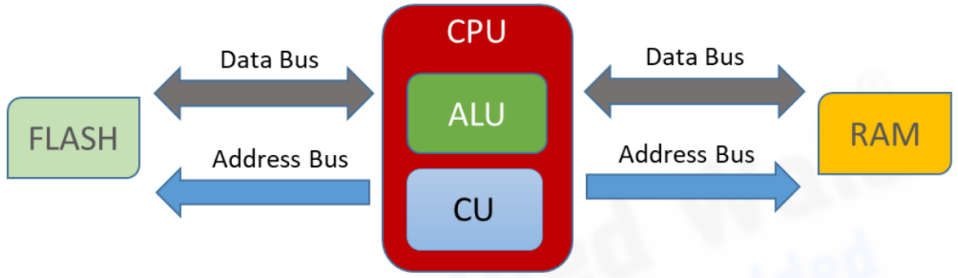

Ford's test car


EEC-1 Control module

Architecture overview

Harvard architecture

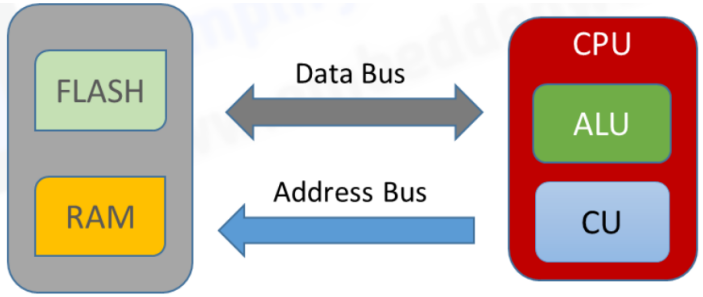
- Known as RISC (Reduced Instruction Set Computer)
- Newer than Von Neumann
- Memory bus and Data bus are separate which allows simultaneous access to data and instructions (Faster)



Harvard Architecture

Von Neumann architecture

- Known as CISC (Complex Instruction Set Computer)
- Shared bus, which can cause the von Neumann bottleneck (a limitation in data throughput)
- Simpler design



Von-Neumann Architecture

Architecture overview

RISC architecture

- Contains a hardwired programming unit
- Simple instructions decoding
- A single clock cycle for instruction execution
- Fixed size for instructions (32bits-4bytes)
- Performance is optimized, with an emphasis on software

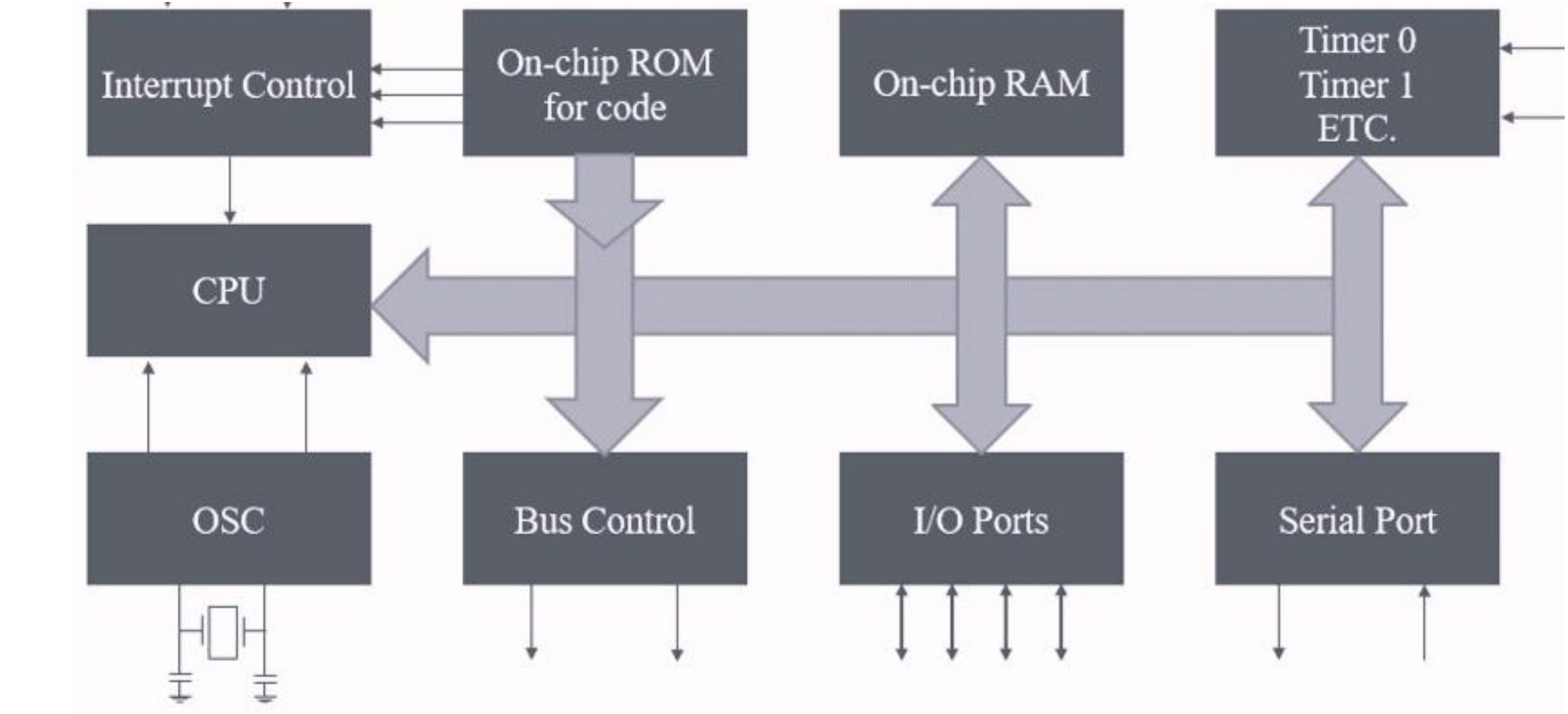
Commonly used in ARC, Alpha, ARC, ARM, AVR, PA-RISC, and SPARC.

CISC architecture

- Microcoded (implemented via small programs inside the CPU)
- Less usage of memory
- Less number of instructions
- Variable size for instructions (16 to 64 bits for each instruction)
- Performance is optimized with a focus on hardware

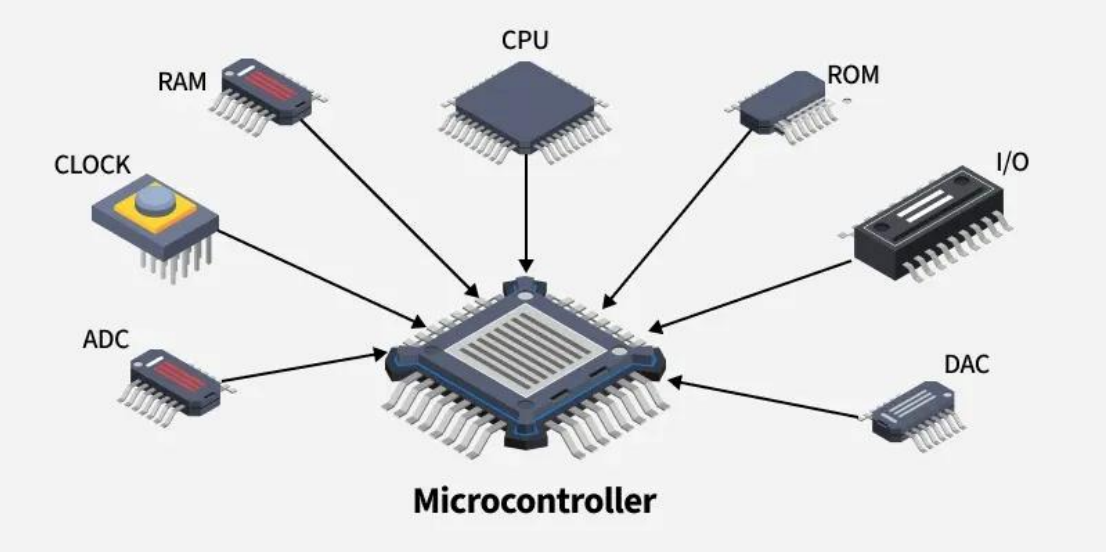
Commonly used in System/360, VAX, AMD, and Intel x86 processors.

Architecture overview



Some features and capabilities

- Digital & Analog I/O
- Timers, PWM, watchdogs
- Communication: UART, I2C, SPI, CAN
- Built-in ADC, DAC
- Low-power and sleep modes



Applications of microcontrollers

- Home appliances
- Automotive
- Medical devices
- IoT and smart devices
- Robotics and automation

Popular microcontroller platforms

- Arduino: Easy for beginners, based on AVR/ARM
- ESP32/ESP8266: Wi-Fi/Bluetooth enabled
- STM32/NXP: Industrial-grade, ARM Cortex-M
- Raspberry Pi Pico: Dual-core RP2040 MCU
- PIC & AVR: Widely used in legacy systems






Challenges and considerations

- Limited memory and processing power
- Real-time performance requirements
- Power management in portable devices
- Security in connected systems

Selecting a microcontroller

Hardware considerations

- Number of inputs/outputs (GPIO)
- Types of communication protocols needed

Software considerations

- CPU frequency needed and decide if an external clock is nedded
- Mathematical calculations (fixed point, floating point)

Microcontroller architecture

- Data size (8-bit, 16-bit or 32-bit) that a CPU can handle at a time

Selecting a microcontroller

Memory considerations

- Flash size and RAM size

Special features considerations

- Built-in peripherals (PWM, ADC…), Operating voltage …

Search for adequate microcontroller

- Search for communities and sources of support
- Examination of costs
- Examination of scalability (for future upgrades)
- Examination of development tools