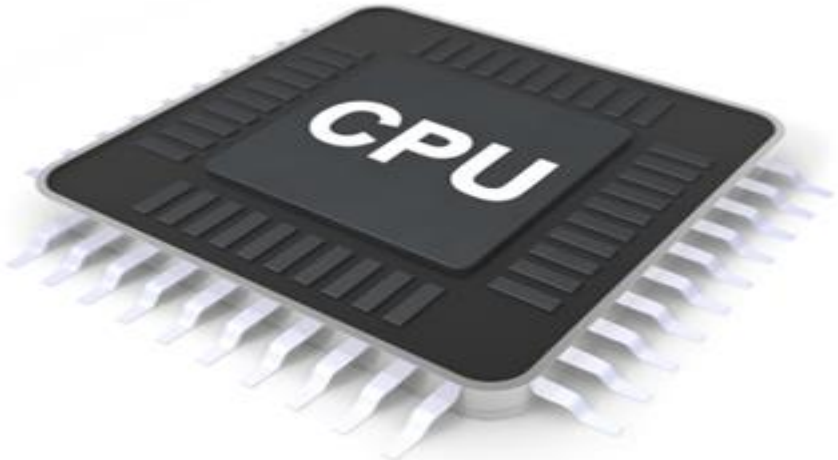


Intro to Embedded Microp & AVR

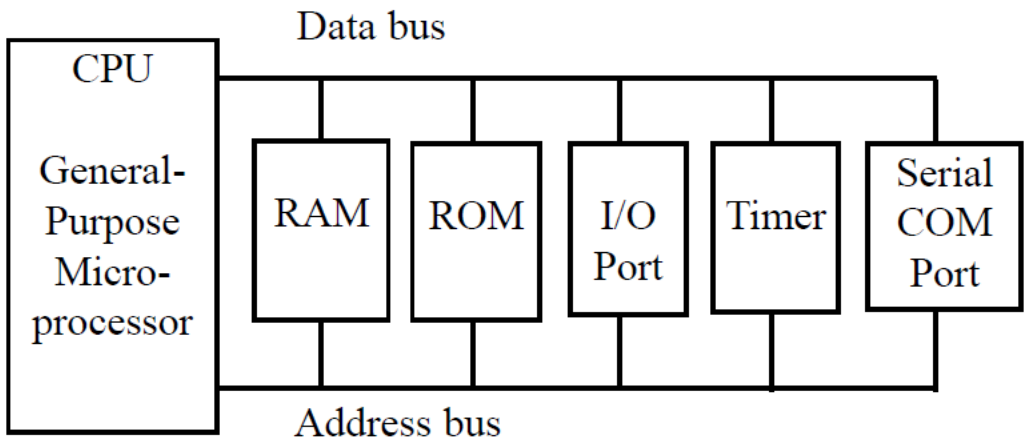
ENEE 3582

Microp

General Purpose vs Embedded

- Microprocessors in computers
- Processes information
- Runs many apps
- Software and OS
- Newer (started in 1980)
- Lots of RAM
- Many pins
- Power hungry
- Non-RT apps
- Advanced, expensive, large die
- Multicore
- Multiple human interfaces
- Not typically rated

- Microcontrollers for devices
- Controls mech/elect/electronic
- Runs single/limited apps
- Firmware, no OS
- Older (1970)
- Tiny RAM
- Limited pins
- Low power
- RT apps
- Simple, cost-effective, small die
- Peripheral support, System-on-a-chip(SoC)
- No/limited human interface
- Rated for temp/pressure/vibration



CPU	RAM	ROM
I/O ----- ADC	Timer	Serial COM Port

A. Mazidi

Choices of Microcontrollers

- Choose a microcontroller based on:
  - Speed/computation power
  - Peripheral support (sub-systems in SoC)
  - Power consumption
  - RAM/ROM needs
  - Number of IO pins needed
    - Packaging: DIP vs QFP
    - https://www.eesemi.com/ic-package-types.htm
  - Cost

Choices of Microcontrollers

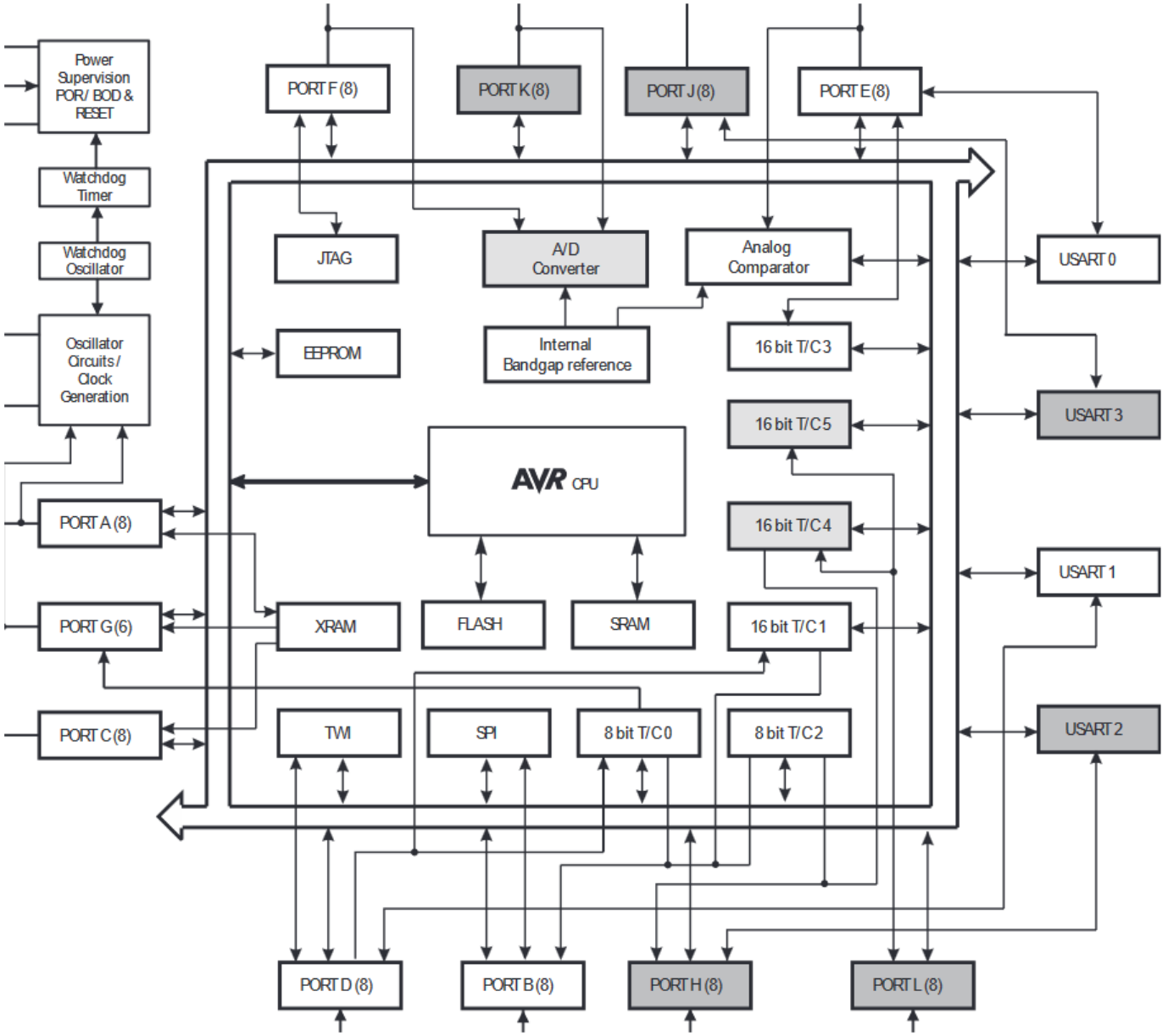
- 4-bit: E.g. COP400, EM73201, W741E260, HD404358
- 8-bit: E.g. 6502, Z80, 8051, AVR, PIC
  - A few bytes to a few hundred KB of RAM
  - Software is in asm and C
  - Still dominate both numbers and dollar volume
  - Two kinds:
    - Old-style CISC, E.g. 6502, Z80, 8051
      - These are >20 years old and doing well
    - Newer style RISC, E.g. AVR, PIC
- 16-bit: E.g. ARM, MIPS, MN10300, PPC
- 32-bit: E.g. 386, AVR32

AVR

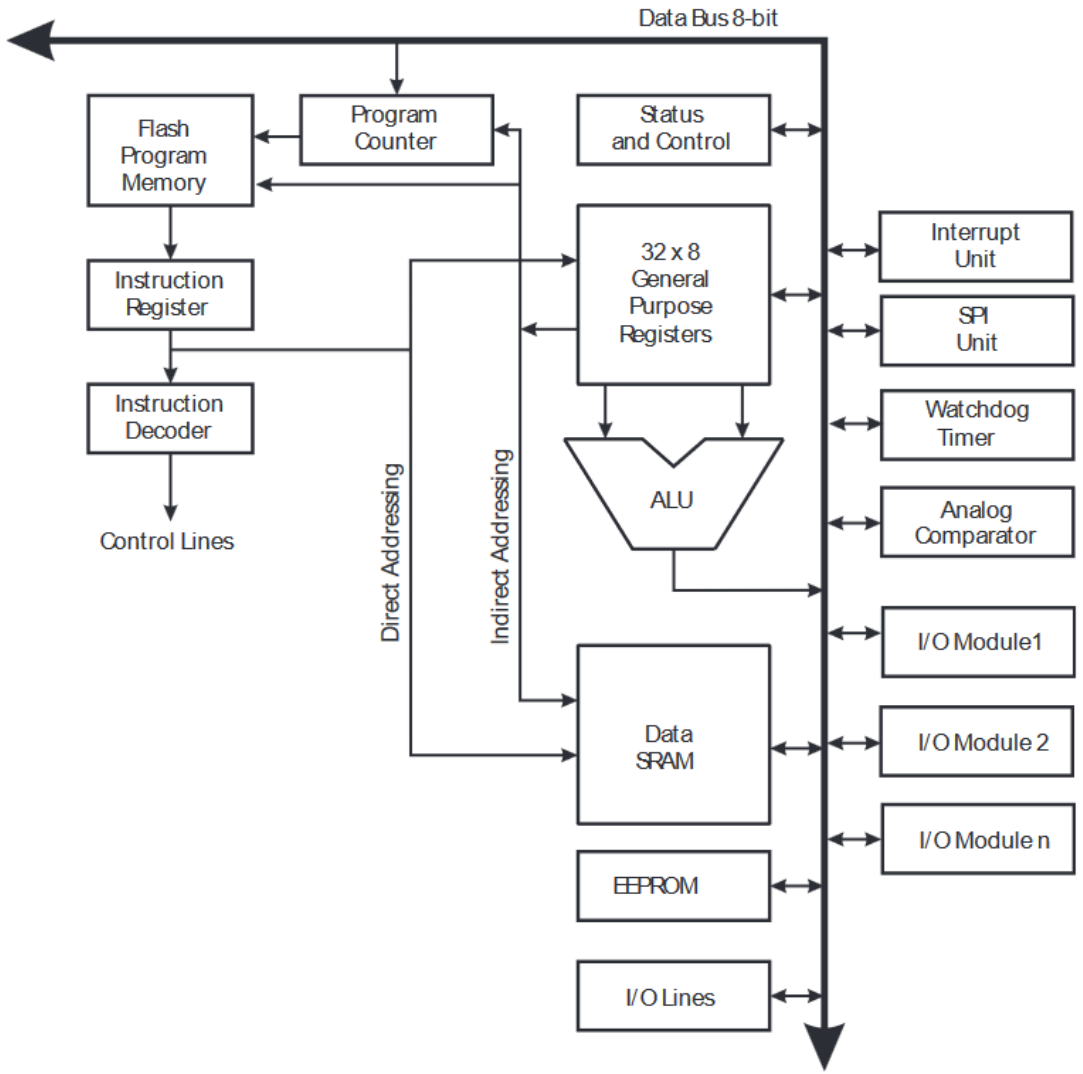
- Designed by two students of Norwegian Institute of Technology (NTH)
  - Bought and developed by Atmel in 1996. Atmel bought by Microchip in 2016
- AVR 8-bit family:
  - Classic AVR (AT90Sxxxx)
    - Replaced. Not recommended for new design
  - Mega AVR (ATmegaxxxx)
    - 120 instructions, expandable
    - Extensive peripheral sets
  - Tiny AVR (ATtinyxxxx)
    - Smaller, limited, low power, low cost
  - Special AVR
    - Unique peripherals: USB, ethernet, Zigbee, CAN
- AVR32: 32bit

ATmega 2560

- Datasheets: https://www.microchip.com/en-us/product/ATmega2560#document-table
- AVR CPU Core:
  - 32, 8-bit registers: R0-R31
  - 3, 16-bit indexing registers: X, Y, Z
- 4KB EEPROM, 8KB RAM
- Data registers: address 0x0000-0x001F
- Internal data RAM (IRAM): address 0x0200-0x21FF
- External data RAM (XRAM): address 0x2200-0xFFFF
- EEPROM: address 0x0000
- FLASH address 0x000000-



https://www.microchip.com/en-us/product/ATmega2560#document-table


Data Bus 8-bit
Flash Program Memory
Program Counter
Status and Control
Instruction Register
Instruction Decoder
Control Lines
32 x 8 General Purpose Registers
ALU
Direct Addressing
Indirect Addressing
Data SRAM
EEPROM
I/O Lines
Interrupt Unit
SPI Unit
Watchdog Timer
Analog Comparator
I/O Module1
I/O Module 2
I/O Module n

Device	Flash	EEPROM	RAM	General Purpose I/O pins	16 bits resolution PWM channels	Serial USARTs	ADC Channels
ATmega640	64KB	4KB	8KB	86	12	4	16
ATmega1280	128KB	4KB	8KB	86	12	4	16
ATmega1281	128KB	4KB	8KB	54	6	2	8
ATmega2560	256KB	4KB	8KB	86	12	4	16
ATmega2561	256KB	4KB	8KB	54	6	2	8

Speed (MHz)[2]	Power Supply	Ordering Code	Package[1][3]	Operation Range
8	1.8V - 5.5V	ATmega2560V-8AU ATmega2560V-8AUR[4] ATmega2560V-8CU ATmega2560V-8CUR[4]	100A 100A 100C1 100C1	Industrial (-40°C to 85°C)
16	4.5V - 5.5V	ATmega2560-16AU ATmega2560-16AUR[4] ATmega2560-16CU ATmega2560-16CUR[4]	100A 100A 100C1 100C1	

RAM

- Random-access memory (RAM):
  - Allows read and writing of memory
  - Same amount of time is required to access any location on the same chip
  - Volatile: information is lost without power.
  - Used to store user's programs
- Dynamic RAM (DRAM):
  - Uses 1 transistor and 1 capacitor for 1 bit.
  - Periodic refresh is required to maintain the contents of chip.
  - Asynchronous
  - SDRAM: synchronous DRAM (edge triggered)
- Static RAM (SRAM):
  - No periodic refresh is required
  - Uses 4-6 transistors for 1 bit

ROM

- Read-only memory (ROM):
  - Can only be read but not written by the processor
  - Nonvolatile
- Mask-programmed ROM (MROM):
  - Programmed when being manufactured
- Programmable ROM (PROM):
  - Programmed by the end user
- Erasable programmable ROM (EPROM)
  - Electrically programmable many times
  - Erased by ultraviolet light
  - Erasable in bulk (whole chip in one erasure operation)

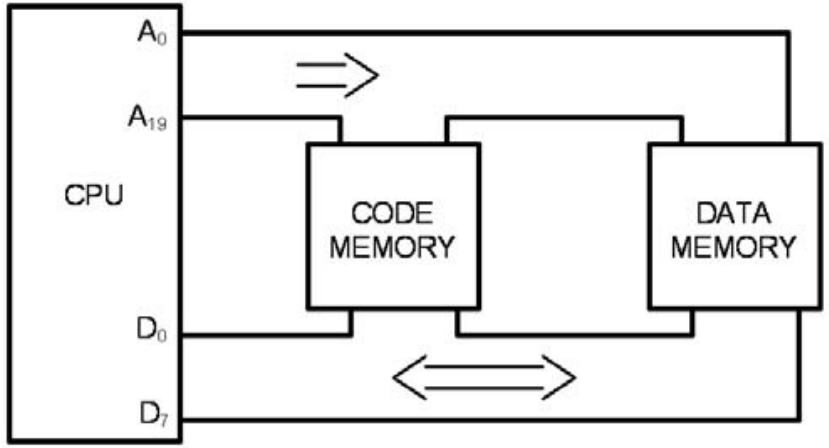
ROM

- Electrically erasable programmable ROM (EEPROM)
  - Electrically programmable many times
  - Electrically erasable many times
  - Erased one location, one row, or whole chip in one operation
  - Erased without removing unit from device.
- Flash EEPROM
  - Electrically programmable many times
  - Electrically erasable many times
  - Erased in bulk/large blocks => faster than EEPROM

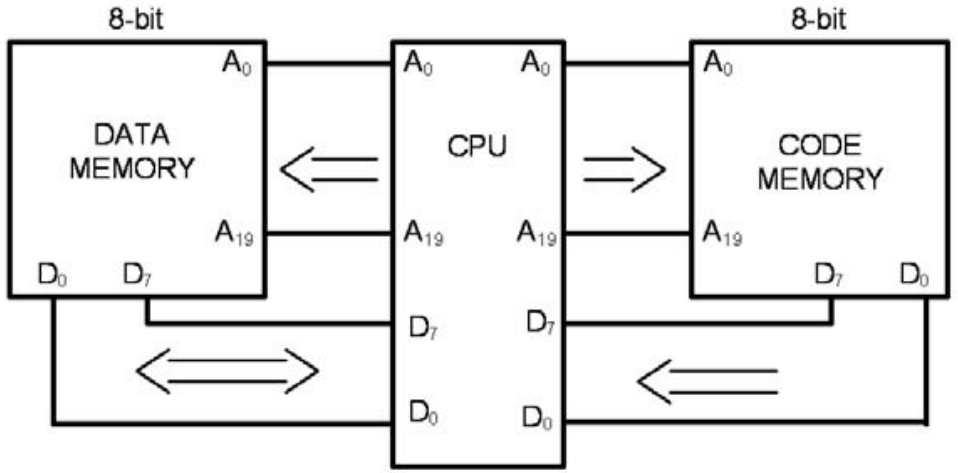
Harvard vs Princeton Memory Architectures

- Von Neumann (Princeton) architecture
  - One memory, one bus
  - Code and Data accessed on the same bus
  - Memory “collisions”
- Harvard architecture
  - Separate memory for code and data memory
  - Separate busses
  - Expensive: Double memory pins for external memory
- AVR Solution:
  - Harvard internal memory
  - Von Neumann for external memory
  -

von Neumann Architecture



Harvard Architecture

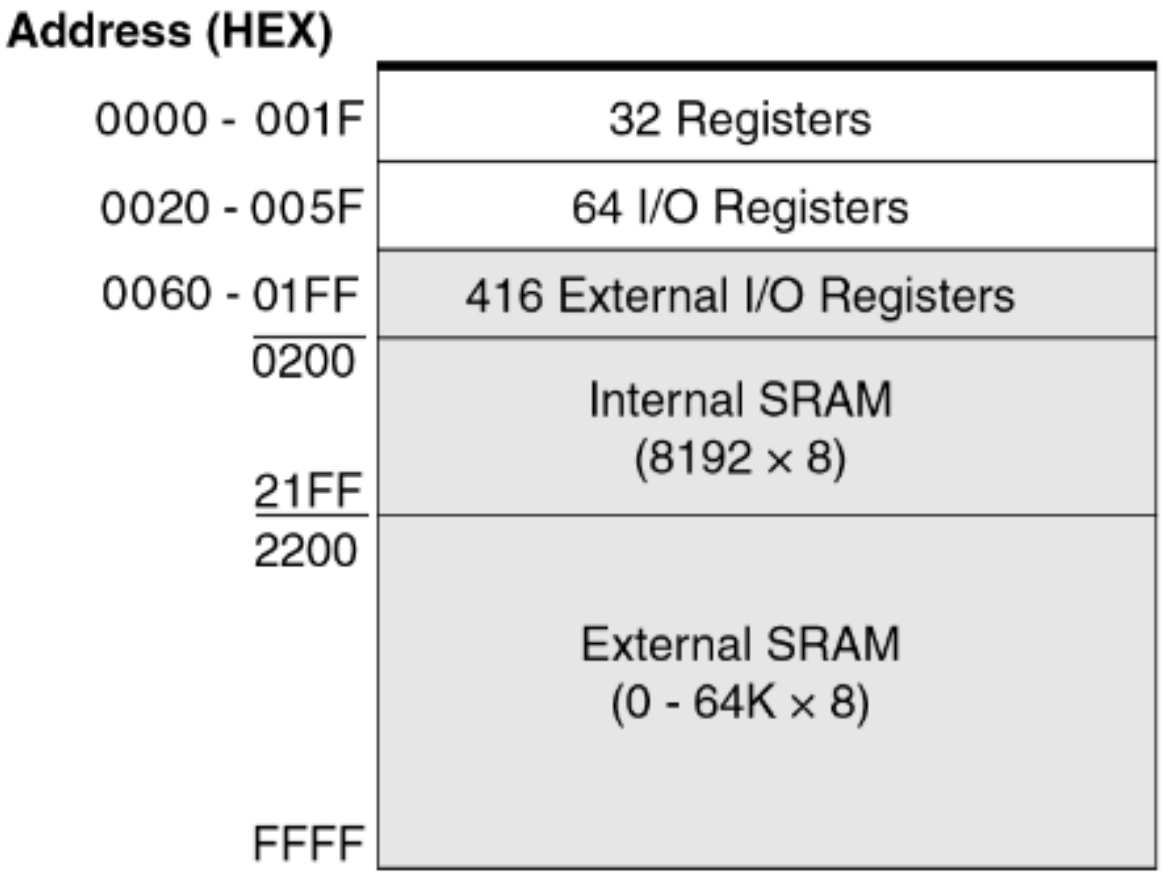


A. Mazidi

Memory Organization

- Memory is organized by addresses
  - Each address points to a memory location
  - n bit address = 2^n locations
- Memory is grouped (measured) into byte locations
- Program Memory:
  - Mega2560: 256KB Flash
  - Address in PC (17bits)
  - Each location is 2B
  - Starting location = 0x00000
- Data Memory:
  - Mega2560: 8KB internal SRAM, 0-64KB external SRAM
  - Each location is a 1B
  - Starting address = 0x0200 (internal), 0x2200 (external)

Mega 2560


Address (HEX)
0000 - 001F
32 Registers
0020 - 005F
64 I/O Registers
0060 - 01FF
416 External I/O Registers
0200
Internal SRAM
(8192 × 8)
21FF
2200
External SRAM
(0 - 64K × 8)
FFFF