Microcontrollers

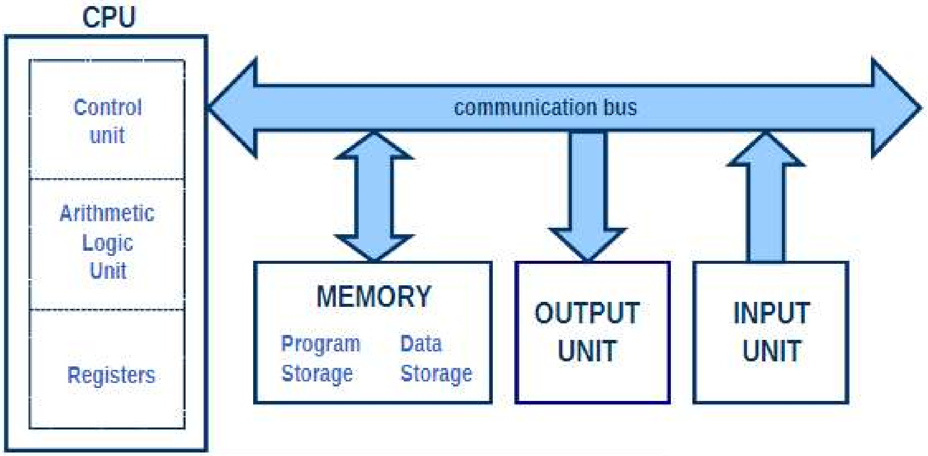
What is Microcontroller

- A microcontroller is a small computer on a single integrated circuit designed to govern a specific operation.

- A typical microcontroller includes a processor, memory and input/output (I/O) peripherals on a single chip.

- Sometimes referred to as an embedded controller or microcontroller unit (MCU)

- microcontrollers are found in vehicles, robots, office machines, medical devices, mobile radio transceivers, vending machines and home appliances, among other devices.



What is Microcontroller. Cont

- Microcontrollers are purchased blank and then programmed with a specific control program
- Once programmed, microcontroller is built into product to make the product smarter and easier to use
- A designer will use microcontroller to:
  - Gather information from various sensors
  - Process the input into a set of actions
  - Use the output mechanisms on the Microcontroller to do a certain function

Microcontroller Application Example

- Air conditioning system
  - A user input a setpoint for the desired room temperature
  - Temperature sensor measure the room temperature
  - Microcontroller compares the actual temperature with the desired temperature
  - If the actual temperature is higher than the desires → MC turns the condition on
  - If the actual temperature is less than the desired temperature → MC turns the condition off.

The difference between computers and microcontrollers

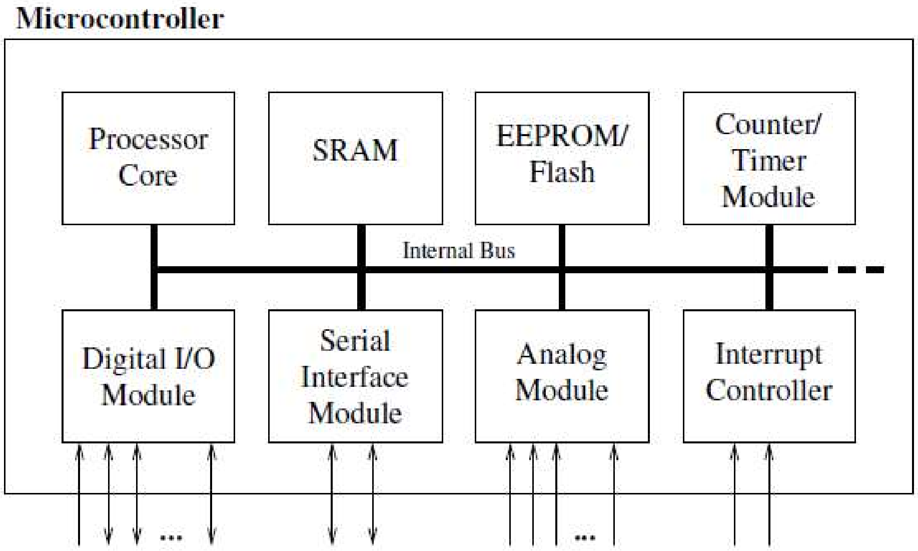
- Computer is a general-purpose function device that can run many applications
- Larger than microcontrollers
- Requires large amount of power to operate
- Expensive

- Microcontrollers preform specific tasks
- Size is smaller than computer
- Less power consumption than computers
- Less expensive than computer

Examples of Microcontrollers

- Home appliances: Fridge, Microwave, Oven
- Medical monitoring device: ECG, Blood pressure meter, Blood pumps
- Security systems,: Alarm, remote surveillance
- Smart card readers
- Cars: A/C, Transmission, Ant-lock Brake System (ABS), Air bags

Microcontroller Layout



Microcontroller Basic Parts

- **Processor Core:** The CPU of the controller. It contains the arithmetic logic unit, the control unit, and the registers (stack pointer, program counter, accumulator register, register file, . . .).

- **Memory:** The memory is sometimes split into program memory and data memory.

- **Digital I/O:** Parallel digital I/O ports are one of the main features of microcontrollers. The number of I/O pins varies from 3-4 to over 90, depending on the controller family and the controller type.

- **Analog I/O:** Apart from a few small controllers, most microcontrollers have integrated analog/digital converters, which differ in the number of channels (2-16) and their resolution (8-12 bits).

- **A2D and D2A:** A2D converts analog signals to digital, D2A converts digital signal signal to analog signal

- **Interfaces:** Controllers generally have at least one serial interface which can be used to download the program and for communication with the development PC in general. Since serial interfaces can also be used to communicate with external peripheral devices, most controllers offer several and varied interfaces like SPI and I2C.

Microcontroller Basic Parts

- **Interrupt Controller:** Interrupts are useful for interrupting the normal program flow in case of (important) external or internal events. In conjunction with sleep modes, they help to conserve power.

- **Timer/Counter:** Most controllers have at least one and more likely 2-3 Timer/Counters, which can be used to timestamp events, measure intervals, or count events.

- **PWM (pulse width modulation) outputs:** which can be used to drive motors or for safe breaking (antilock brake system, ABS). Furthermore, the PWM output can, in conjunction with an external filter, be used to realize a cheap digital/analog converter.

- **Watchdog Timer:** Since safety-critical systems form a major application area of microcontrollers, it is important to guard against errors in the program and/or the hardware. The watchdog timer is used to reset the controller in case of software “crashes”.

- **Debugging Unit:** Some controllers are equipped with additional hardware to allow remote debugging of the chip from the PC.

Memory In Microcontrollers

Register File: A (usually) relatively small memory embedded on the CPU. It is used as a scratchpad for temporary storage of values the CPU is working with - you could call it the CPU's short-term memory.

Data Memory: For longer term storage, generic CPUs usually employ an external memory which is much larger than the register file. Data that is stored there may be short-lived but may also be valid for as long as the CPU is running.

Instruction Memory: Like the data memory, the instruction memory is usually a relatively large external memory (at least with general CPUs). Actually, with von-Neumann-architectures, it may even be the same physical memory as the data memory.

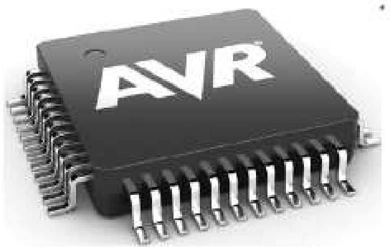
Cache memory (optional): Cache memory is a small-sized type of volatile computer memory that provides high-speed data access to a processor and stores frequently used computer programs, applications and data.

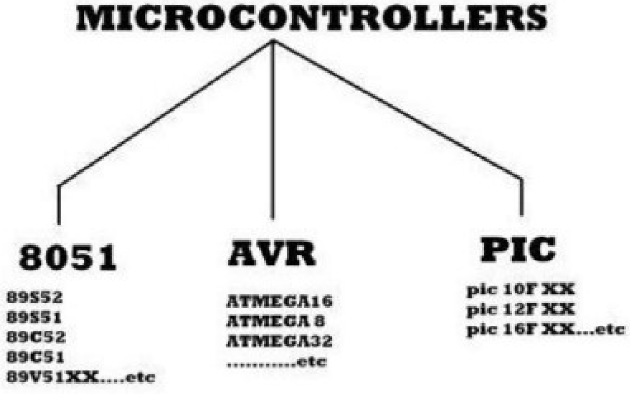
Cache memory is the fastest memory available and acts as a buffer between RAM and the CPU.

Microcontrollers Classifications

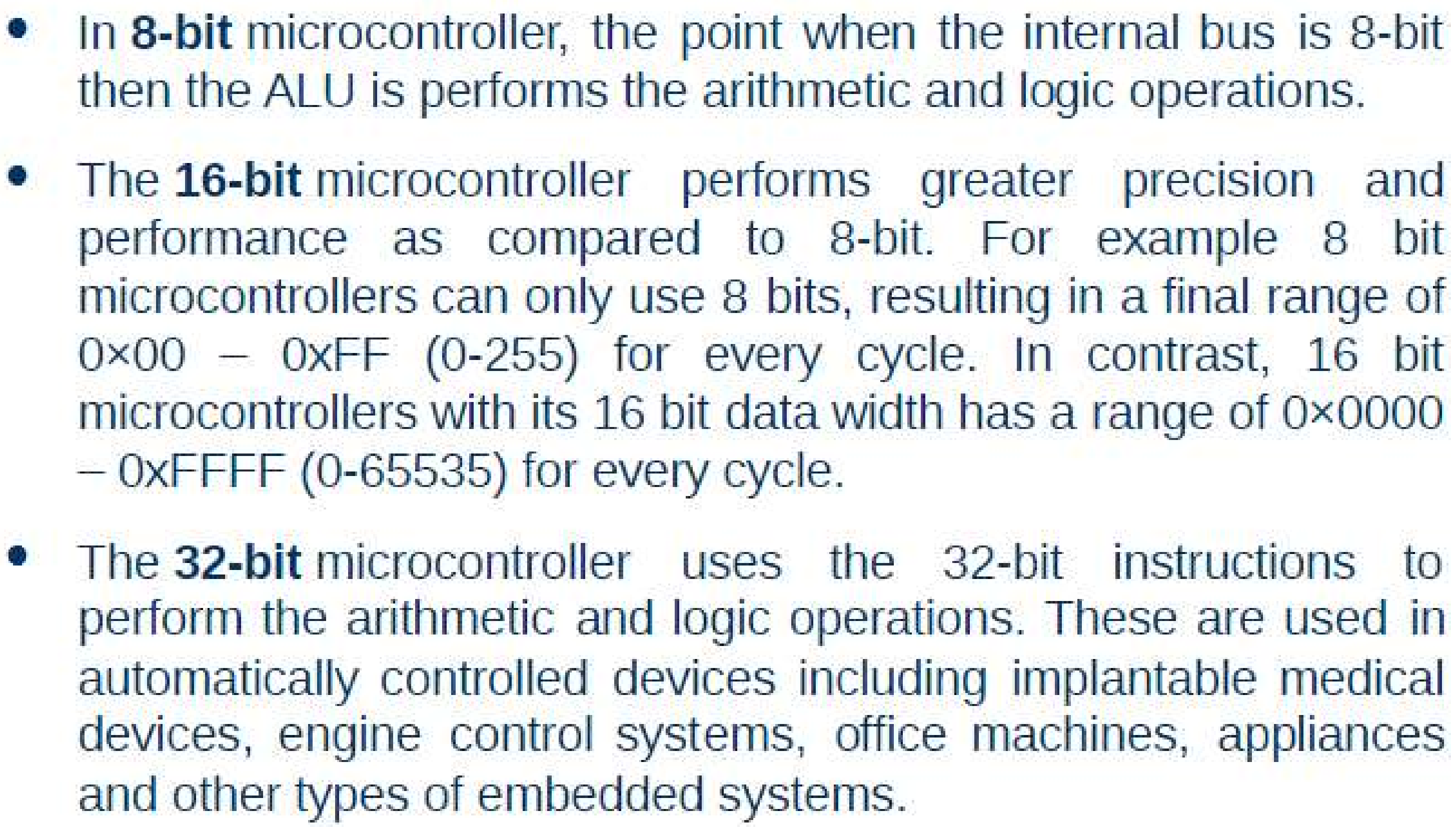
The microcontrollers are characterized by their:

- Family
- bits,
- and memory structure.



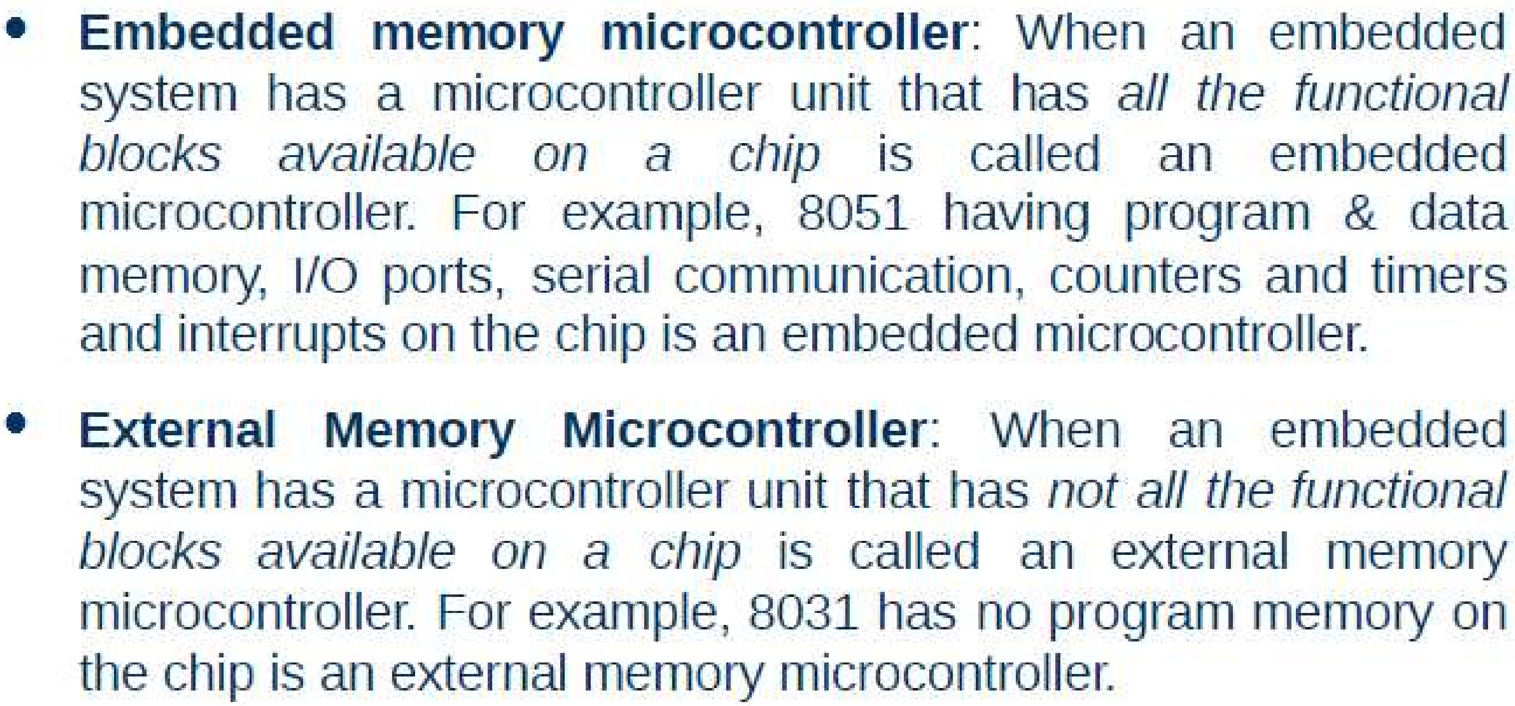


Classification According to Number of Bits



- In **8-bit** microcontroller, the point when the internal bus is 8-bit then the ALU is performs the arithmetic and logic operations.
- The **16-bit** microcontroller performs greater precision and performance as compared to 8-bit. For example 8 bit microcontrollers can only use 8 bits, resulting in a final range of 0×00 – 0xFF (0-255) for every cycle. In contrast, 16 bit microcontrollers with its 16 bit data width has a range of 0×0000 – 0xFFFF (0-65535) for every cycle.
- The **32-bit** microcontroller uses the 32-bit instructions to perform the arithmetic and logic operations. These are used in automatically controlled devices including implantable medical devices, engine control systems, office machines, appliances and other types of embedded systems.

Classification According to Memory Device

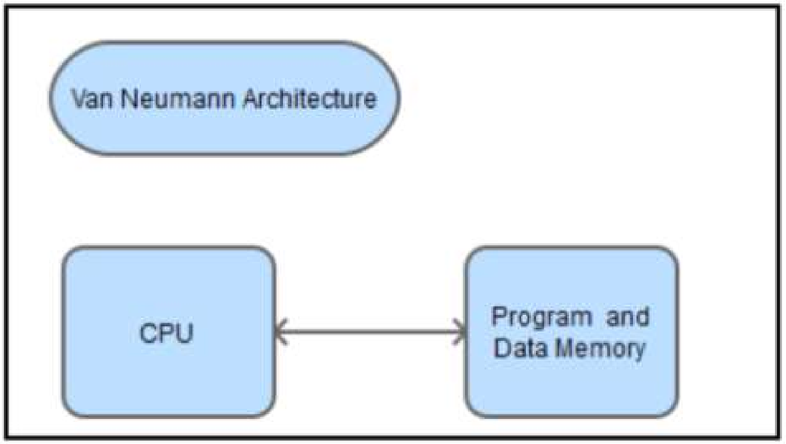


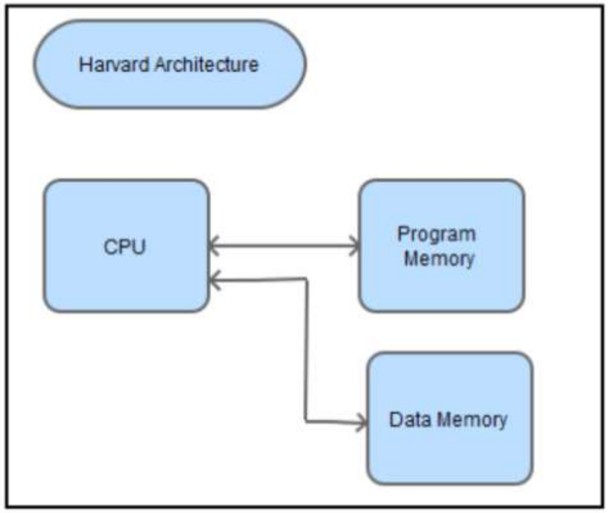
- **Embedded memory microcontroller**: When an embedded system has a microcontroller unit that has *all the functional blocks available on a chip* is called an embedded microcontroller. For example, 8051 having program & data memory, I/O ports, serial communication, counters and timers and interrupts on the chip is an embedded microcontroller.
- **External Memory Microcontroller**: When an embedded system has a microcontroller unit that has *not all the functional blocks available on a chip* is called an external memory microcontroller. For example, 8031 has no program memory on the chip is an external memory microcontroller.

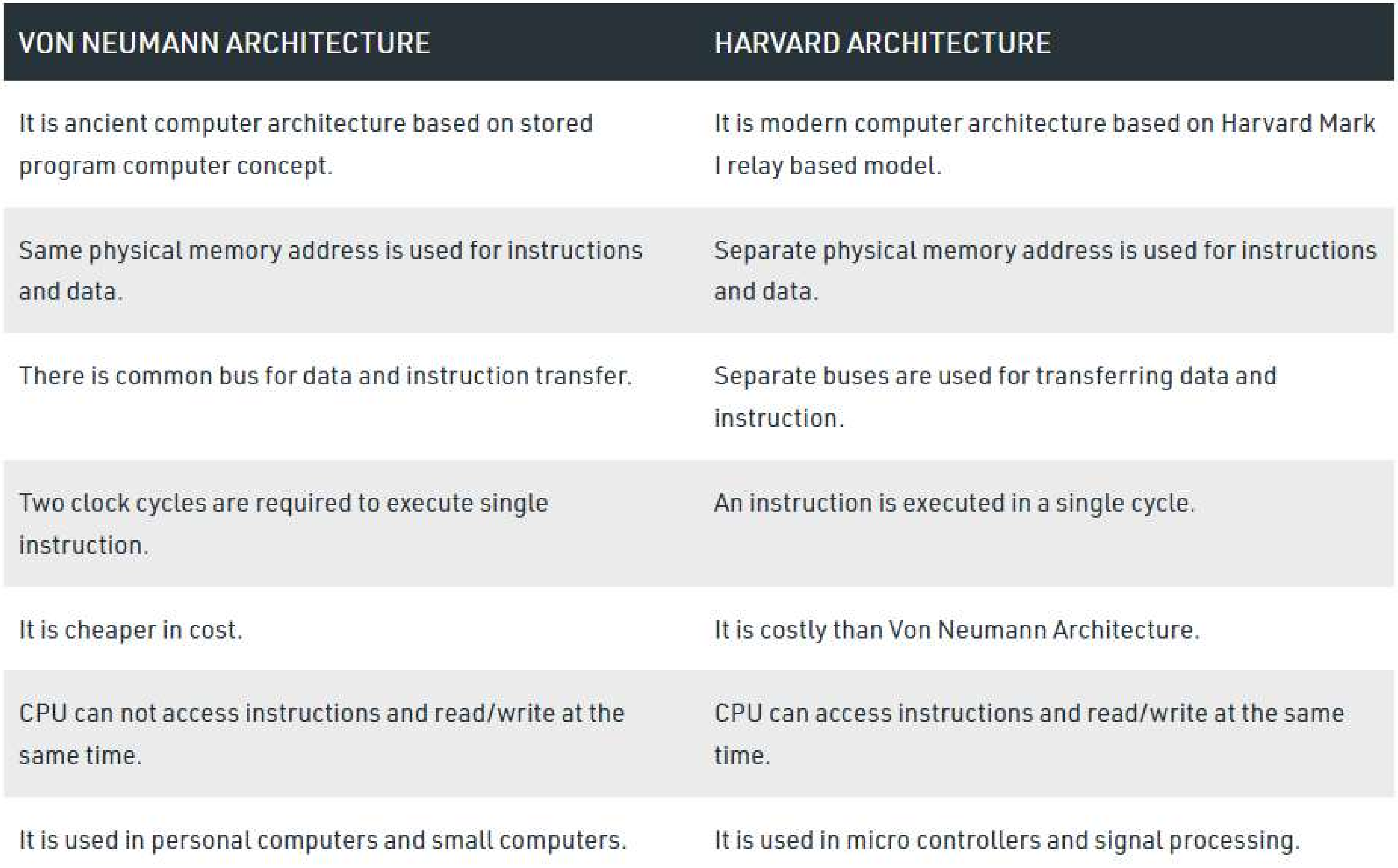
Classification According to Memory Architecture

Von Neumann Architecture: In this architecture, program and data are stored together and are accessed through the same bus. Unfortunately, this implies that program and data accesses may conflict (resulting in the famous von Neumann bottleneck), leading to unwelcome delays.

Harvard Architecture: This architecture demands that program and data are in separate memories which are accessed via separate buses. In consequence, code accesses do not conflict with data accesses which improves system performance. As a slight drawback, this architecture requires more hardware, since it needs two busses and either two memory chips or a dual-ported memory (a memory chip which allows two independent accesses at the same time).



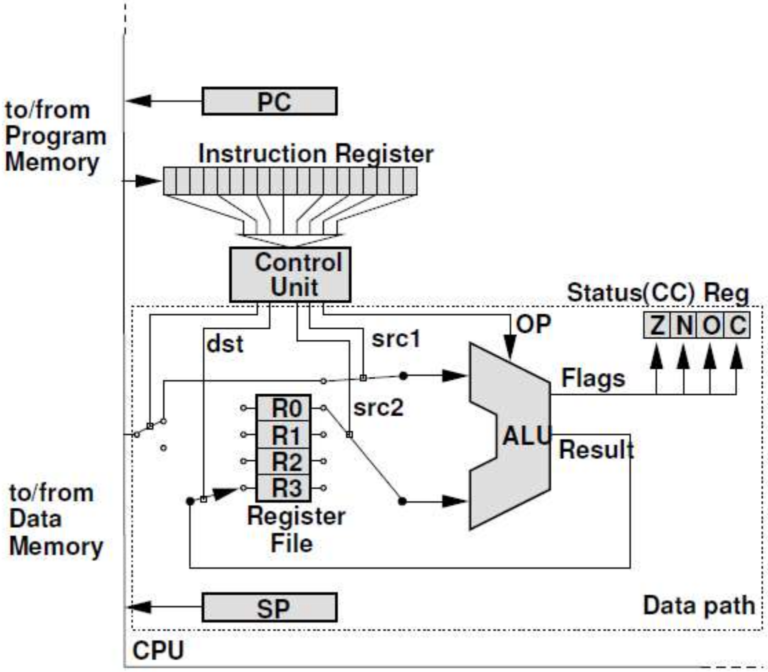




VON NEUMANN ARCHITECTURE	HARVARD ARCHITECTURE
It is ancient computer architecture based on stored program computer concept.	It is modern computer architecture based on Harvard Mark I relay based model.
Same physical memory address is used for instructions and data.	Separate physical memory address is used for instructions and data.
There is common bus for data and instruction transfer.	Separate buses are used for transferring data and instruction.
Two clock cycles are required to execute single instruction.	An instruction is executed in a single cycle.
It is cheaper in cost.	It is costly than Von Neumann Architecture.
CPU can not access instructions and read/write at the same time.	CPU can access instructions and read/write at the same time.
It is used in personal computers and small computers.	It is used in micro controllers and signal processing.

Microprocessors

Processor Architecture



1- Arithmetic Logic Unit

At the core of the CPU is the arithmetic logic unit (ALU), which is used to perform computations (AND, ADD, INC, . . .). Several control lines select which operation the ALU should perform on the input data. The ALU takes two inputs and returns the result of the operation as its output. Source and destination are taken from registers or from memory

2- Registers

Status register:

Z	N	O	C

Z (Zero): The result of the operation is zero.

N (Negative): The result of the operation is negative, that is, the *most significant bit* (msb) of the result is set (1).

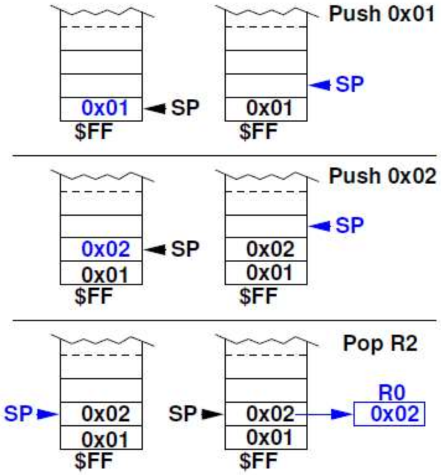
O (Overflow): The operation produced an overflow, that is, there was a change of sign in a two's-complement operation.

C (Carry): The operation produced a carry.

Data Register:

Data registers are used as source or destination for the ALU to perform the operations. Data registers can be 8 or 16 registers. Some registers can be merged to form 32 bits registers if the arithmetic operation requires that.

Alternatively, operands/result can come from/be stored to the memory. However, memory access is much slower than access to the register file, so it is usually wise to use the register file if possible.



Stack Pointer register (SP)

The stack is used by the CPU to store return addresses and possibly register contents during subroutine and interrupt service routine calls. It is accessed with the commands PUSH (put something on the stack) and POP (remove something from the stack).

Program counter register (PC)

Is used to store the address of the next program instruction. The PC is either incremented to point to the next instruction in the sequence or is loaded with a new address in the case of a jump or subroutine call. After a reset, the PC is typically initialized to $0000.

Instruction Register (IR)

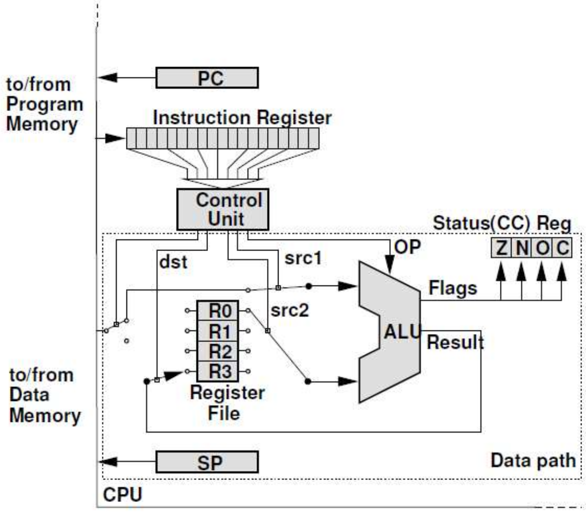
Is used to store the opcode of the next instruction (e.g. add, sub, inc, ..etc.)

3- Control Unit

- The task of the control unit to determine which operation should be executed next and to configure the data path accordingly.

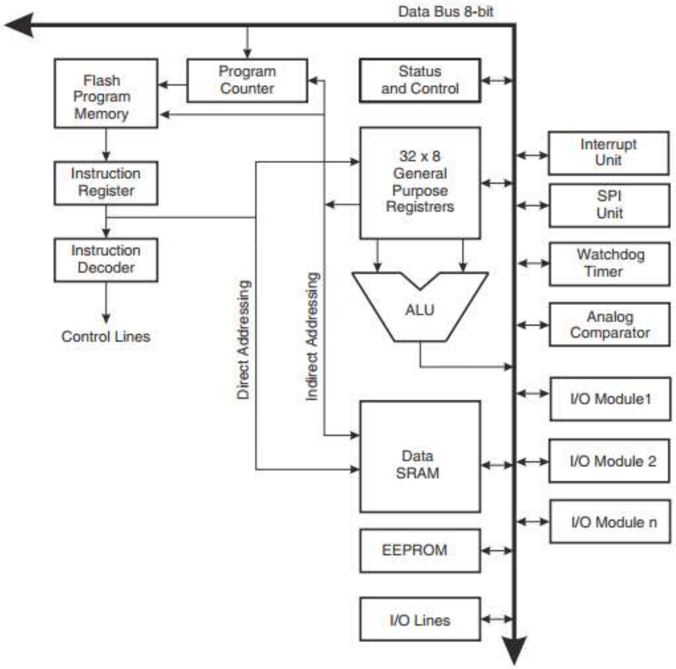
- The control unit loads this instruction into the instruction register (IR), decodes the instruction, and sets up the data path to execute it.

- Data path configuration includes providing the appropriate inputs for the ALU (from registers or memory), selecting the right ALU operation, and making sure that the result is written to the correct destination (register or memory).

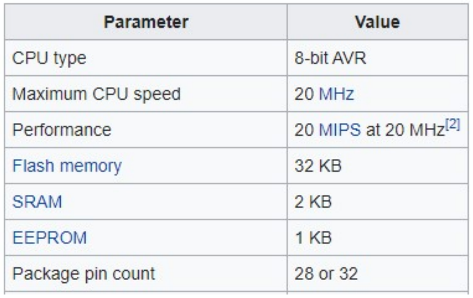


Example: 8-bit AVR Microprocessor

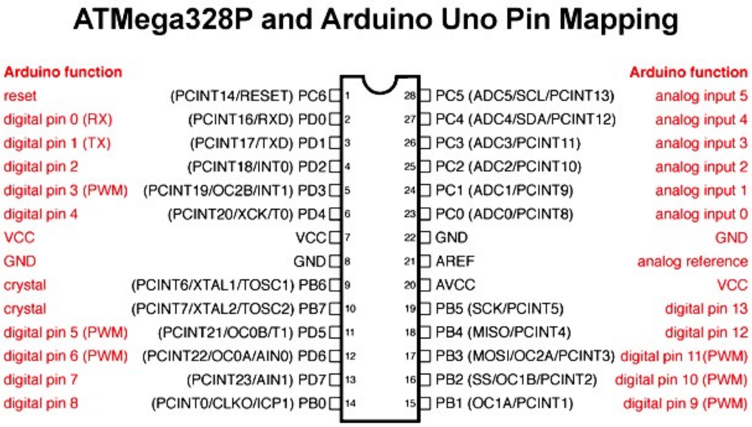
- ATmega328P is the microcontroller used in the Arduino uno board.
- ATmega328P has the 8-Bit AVR CPU
- AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining.
- The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time.
- Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations.



Link to 8 bit AVR CPU specs:
https://www.arnabkumardas.com/arduino-tutorial/avr-cpu-core/



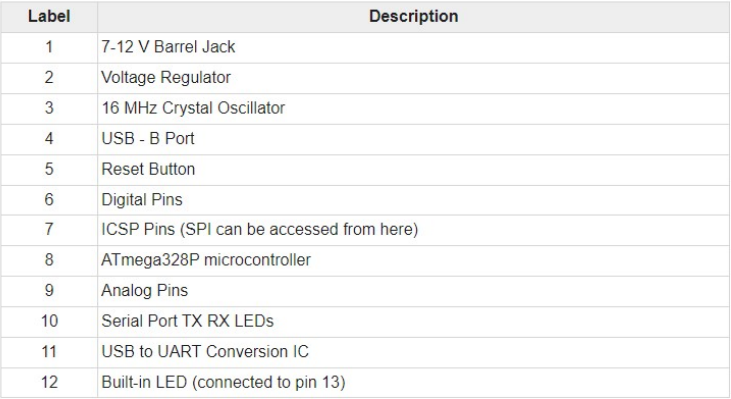
Parameter	Value
CPU type	8-bit AVR
Maximum CPU speed	20 MHz
Performance	20 MIPS at 20 MHz[2]
Flash memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Package pin count	28 or 32



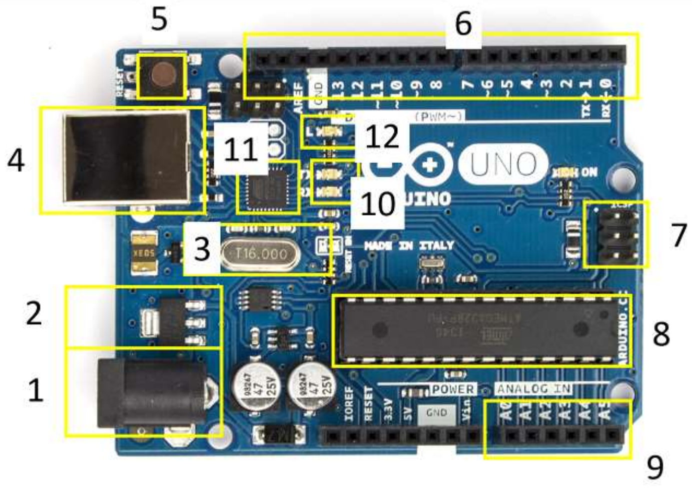
ATMega328P and Arduino Uno Pin Mapping

Arduino function					Arduino function
reset	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)	analog input 0
VCC	VCC	7	22	GND	GND
GND	GND	8	21	AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)	digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

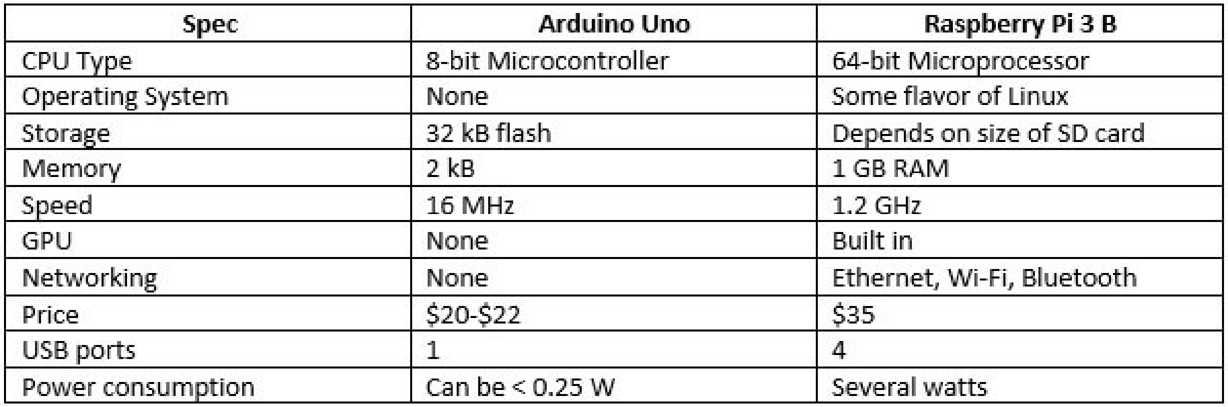
Data sheet:
https://web.archive.org/web/20160412093102/http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Summary.pdf



Label	Description
1	7-12 V Barrel Jack
2	Voltage Regulator
3	16 MHz Crystal Oscillator
4	USB - B Port
5	Reset Button
6	Digital Pins
7	ICSP Pins (SPI can be accessed from here)
8	ATmega328P microcontroller
9	Analog Pins
10	Serial Port TX RX LEDs
11	USB to UART Conversion IC
12	Built-in LED (connected to pin 13)



How to Read and Compare Microcontrollers



Spec	Arduino Uno	Raspberry Pi 3 B
CPU Type	8-bit Microcontroller	64-bit Microprocessor
Operating System	None	Some flavor of Linux
Storage	32 kB flash	Depends on size of SD card
Memory	2 kB	1 GB RAM
Speed	16 MHz	1.2 GHz
GPU	None	Built in
Networking	None	Ethernet, Wi-Fi, Bluetooth
Price	$20-$22	$35
USB ports	1	4
Power consumption	Can be < 0.25 W	Several watts