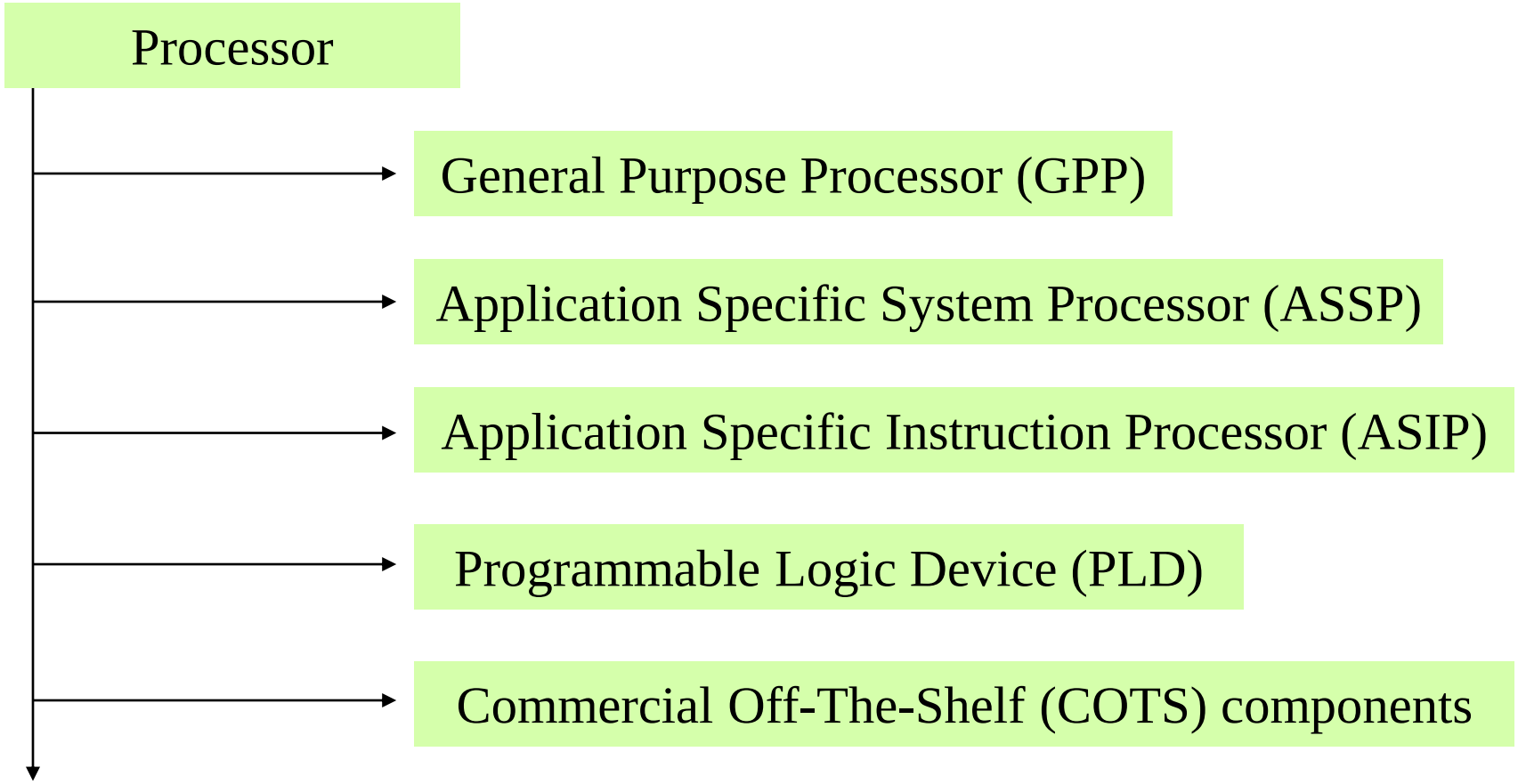
Professional Elective-II: Embedded Systems (PECCSE601B)

Module - II

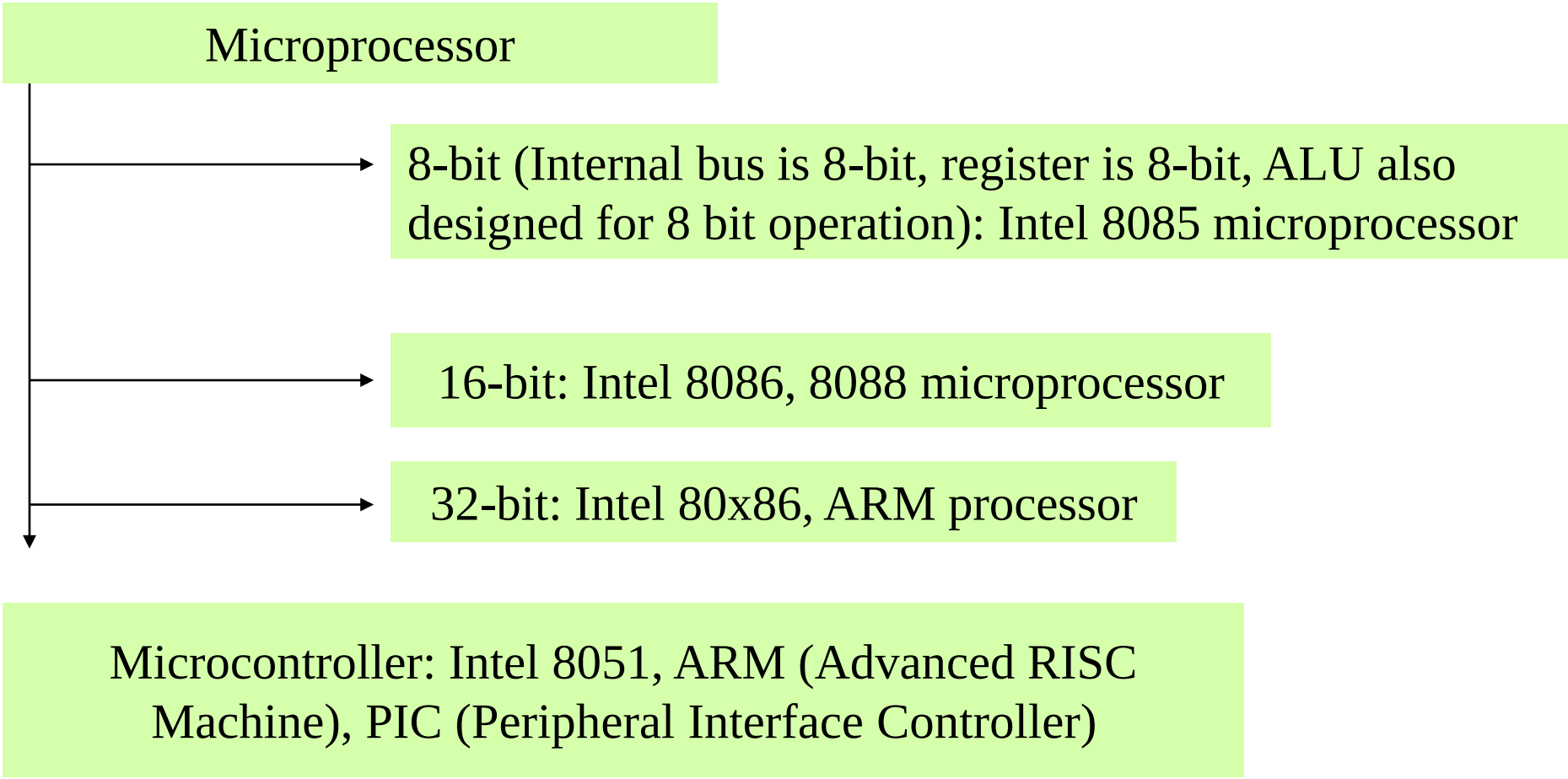
The typical embedded system

Prepared by – Dr. Susmita Biswas
Associate Professor
University of Engineering and Management, Kolkata

processors

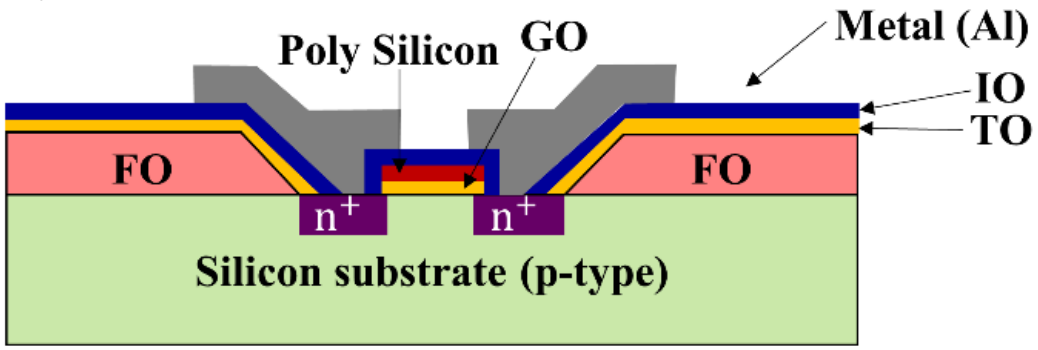

Processor
General Purpose Processor (GPP)
Application Specific System Processor (ASSP)
Application Specific Instruction Processor (ASIP)
Programmable Logic Device (PLD)
Commercial Off-The-Shelf (COTS) components

General Purpose Processor (GPP)


Microprocessor
8-bit (Internal bus is 8-bit, register is 8-bit, ALU also designed for 8 bit operation): Intel 8085 microprocessor
16-bit: Intel 8086, 8088 microprocessor
32-bit: Intel 80x86, ARM processor
Microcontroller: Intel 8051, ARM (Advanced RISC Machine), PIC (Peripheral Interface Controller)

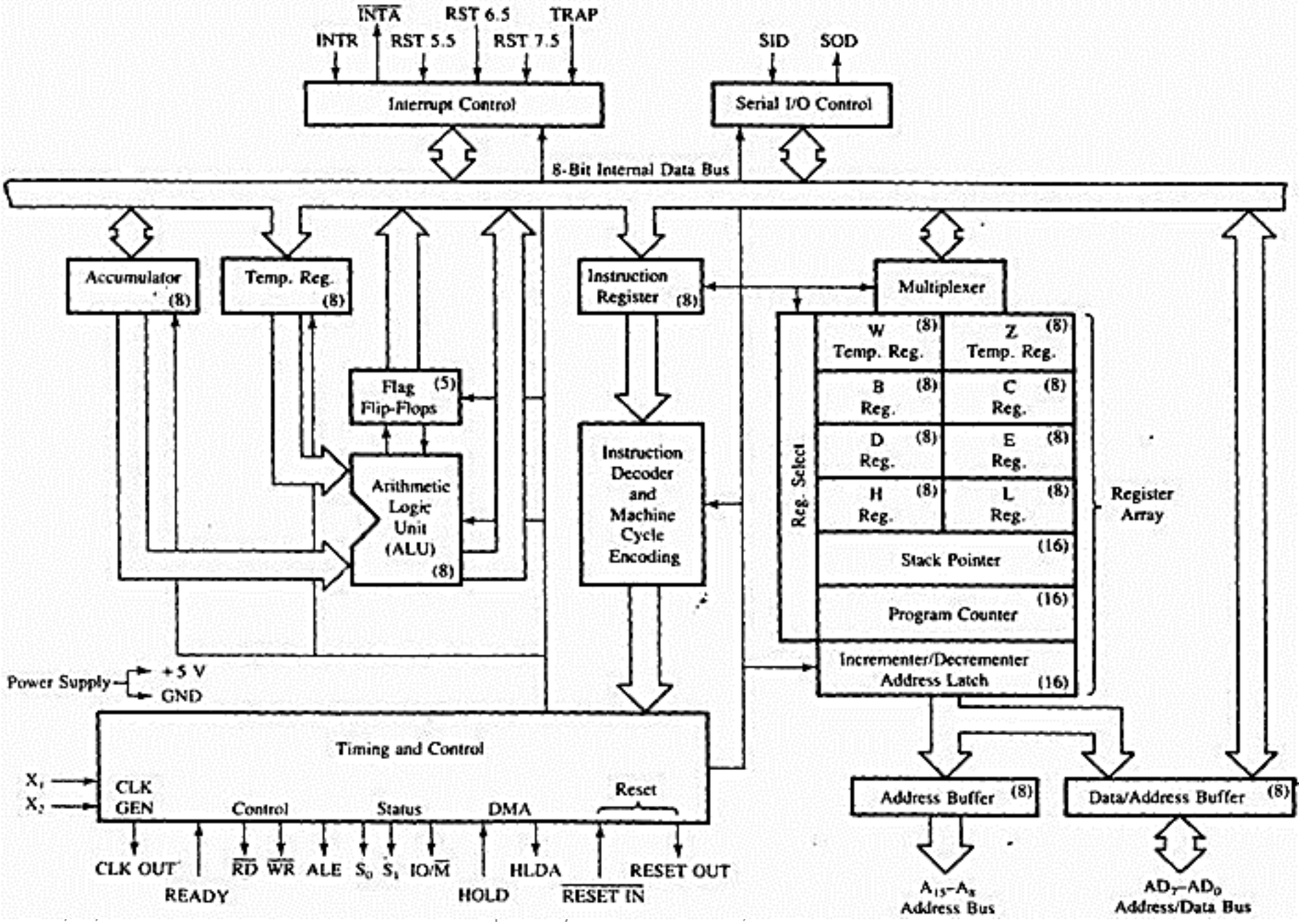
Intel 8085 microprocessor architecture

- The Intel 8085 microprocessor is an **8-bit microprocessor** designed by Intel in 1977 using **NMOS (n-channel metal-oxide semiconductor) technology**.



- Its architecture consists of several key components, including the accumulator, registers, program counter, stack pointer, instruction register, flags register, data bus, address bus, and control bus.

- The 8085 is based on **Von-Neumann architecture**, where the data and instructions are in the same memory space without any distinction between them.
- The Intel 8085 comes as a 40-pin IC package.



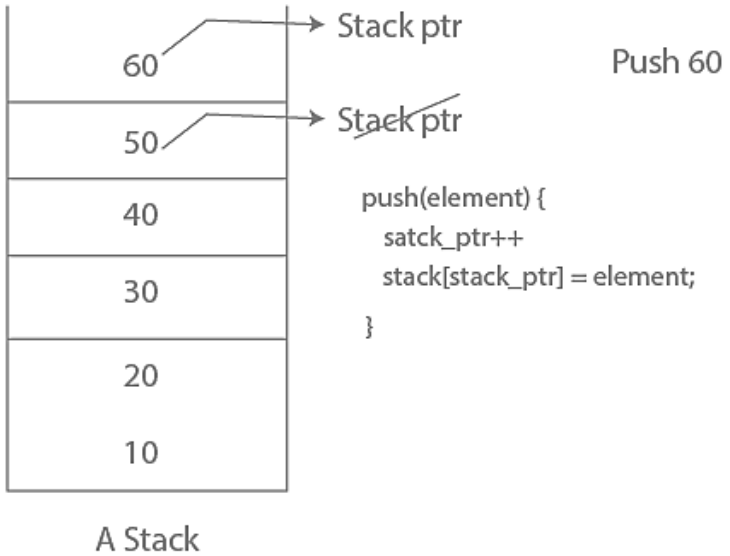
Intel 8085 microprocessor architecture

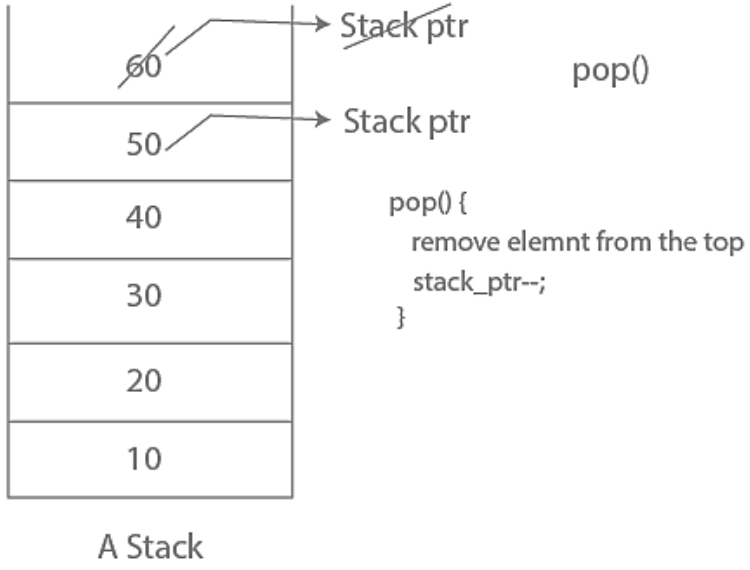
1. **Arithmetic and Logic Unit (ALU):** Arithmetic Operations, Logical operations, Bit-Shifting Operations.
2. **Accumulator:** Accumulator is used to perform I/O, arithmetic, and logical operations. It is connected to ALU and the internal data bus. For all arithmetic operations Accumulator's 8-bit pin will always there connected with ALU and in most-off times results of all the operations, carried by different instructions, will be stored in the accumulator after operation performance.
3. **General Purpose Registers:** There are six general-purpose registers. These registers can hold 8-bit values. These 8-bit registers are B,C,D,E,H,L. These registers work as 16-bit registers when they work in pairs like B-C, D-E, and H-L.

Here registers **W and Z are reserved registers**. We can't use these registers in arithmetic operations. It is reserved for microprocessors for internal operations like swapping two 16-bit numbers. We know that to swap two numbers we need a third variable hence here W-Z register pair works as **temporary registers** and we can swap two 16-bit numbers using this pair.

Intel 8085 microprocessor architecture

4. **Program Counter:** Program Counter holds the address value of the memory to the next instruction that is to be executed. It is a 16-bit register. Program Counter(PC) works in random memory locations.
5. **Stack Pointer :** It is a 16-bit special register. A stack pointer stores the memory address of the last data element added to the stack or, in some cases, the first available address in the stack. Stack pointer works in a continuous and contiguous part of the memory. In stack, the content of the register is stored that is later used in the program. This pointer is very useful in stack-related operations like PUSH, POP, and nested CALL requests initiated by Microprocessor.





Intel 8085 microprocessor architecture

6. **Flag Register:** Flag Register contains 8-bit out of which 5-bits are important. Those 5 flip-flops are set/reset after an operation according to data condition of the result in accumulator or other register.

- **Carry Flag (CY):** Set (q=1) when arithmetic operation results in a carry otherwise reset (q=0).
- **Parity Flag (P):** Set (q=1) when result has an even number of 1, otherwise reset (q=0).
- **Auxiliary Carry Flag (AC):** Set (q=1) when in an arithmetic operation carry is generated by digit D3 and propagated to digit D4 otherwise reset (q=0).
- **Zero Flag (Z):** Set (q=1) when result of an arithmetic operation is zero otherwise reset (q=0).
- **Sign Flag (S):** Set (q=1) when sign bit (MSB) of the result of an arithmetic operation is 1 (negative result) otherwise reset (q=0).

Intel 8085 microprocessor architecture

7. **Instruction register and decoder:** It is an 8-bit register that holds the instruction that is just fetched from the memory and being decoded and executed by the processor.
8. **Timing and control unit:** The timing and control unit comes under the CPU section, and it controls the flow of data from the CPU to other devices. There are certain timing and control signals like Control signals, DMA Signals, RESET signals and Status signals.

•**ALE –** It is an **Address Latch Enable** signal. It goes high (1) during first T state of a machine cycle and enables the lower 8-bits of the address, otherwise data bus is activated.

•**IO/$\overline{M}$ –** It is a status signal which determines whether the address is for input-output or memory. When it is high(1) the address on the address bus is for input-output devices. When it is low(0) the address on the address bus is for the memory.

•**SO, S1 –** These are status signals. They distinguish the various types of operations such as halt, reading, instruction fetching or writing.

•**$\overline{\text{RD}}$ –** It is a signal to control READ operation. When it is low the selected memory or input-output device is read.
•**$\overline{\text{WR}}$ –** It is a signal to control WRITE operation. When it goes low the data on the data bus is written into the selected memory or I/O location.
•**READY –** It senses whether a peripheral is ready to transfer data or not. If READY is high(1) the peripheral is ready. If it is low(0) the microprocessor waits till it goes high. It is useful for interfacing low speed devices.
•**$\overline{\text{RESET IN}}$** – When the signal on this pin is low(0), the program-counter is set to zero.
•**RESET OUT** – This signal indicates that the MPU is being reset. The signal can be used to reset other devices.
•**HOLD** – It indicates that another device is requesting the use of the address and data bus. Having received HOLD request the microprocessor relinquishes the use of the buses as soon as the current machine cycle is completed. Internal processing may continue. After the removal of the HOLD signal the processor regains the bus.
•**HLDA** – It is a signal which indicates that the hold request has been received after the removal of a HOLD request, the HLDA goes low.

Intel 8085 microprocessor architecture

9. System bus:

- The **data bus** is **8-bit bidirectional** and carries the data which is to be stored. The 8 data lines are manipulating 8-bit data ranging from 00 to FF i.e. (2^8 = 256) numbers from 0000 0000 -1111 1111.

- The **address bus** is **16-bit unidirectional** and carries the location where information (instruction/ data) is to be stored. MPU carries 16-bit address i.e. 2^{16} = 65,536 or 64K memory locations.

- The **control bus** is having various **single lines** used for sending control signals in the form of the pulse to the memory and I/O devices.

10. **Interrupt control:** Whenever a microprocessor is executing the main program and if suddenly an interrupt occurs, the microprocessor shifts the control from the main program to process the incoming request. After the request is completed, the control goes back to the main program.

There are 5 interrupt signals in 8085 microprocessors. :

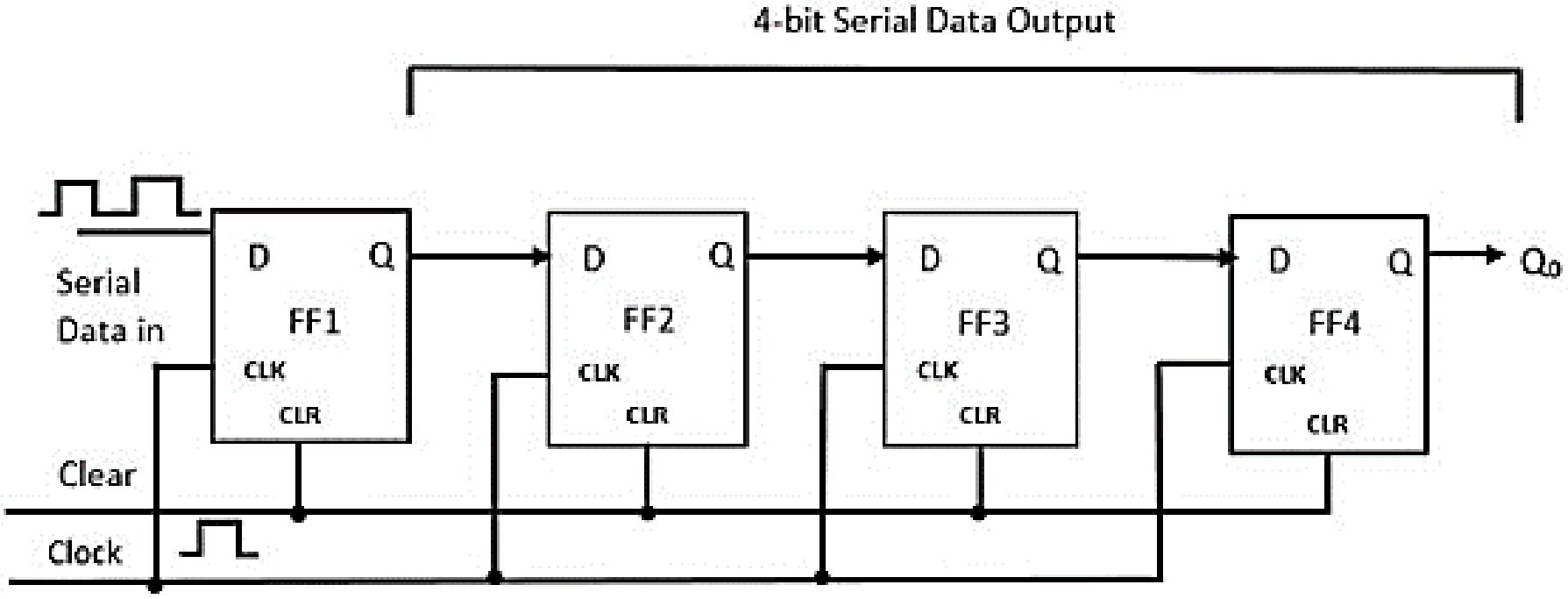
- **TRAP:** The TRAP interrupt is a **non-maskable interrupt** that is generated by an external device, such as a power failure or a hardware malfunction. The TRAP interrupt has the **highest priority and cannot be disabled or ignored by the instructions of CPU**.
- **RST 7.5:** The RST 7.5 interrupt is a **maskable** interrupt (can be disabled or ignored by the instructions of CPU) that is generated by a software instruction. It has the second highest priority.
- **RST 6.5:** The RST 6.5 interrupt is a **maskable** interrupt that is generated by a software instruction. It has the third highest priority.
- **RST 5.5:** The RST 5.5 interrupt is a **maskable** interrupt that is generated by a software instruction. It has the fourth highest priority.
- **INTR:** The INTR interrupt is a **maskable** interrupt that is generated by an external device, such as a keyboard or a mouse. It has the **lowest priority** and can be disabled. $\overline{\textbf{INTA}}$ is an interrupt acknowledgement sent by the microprocessor after INTR is received.

11. Serial Input/output control:

- It controls the serial data communication by using Serial input data and Serial output data. **Serial Input/Output control in the 8085 microprocessor refers to the communication of data between the microprocessor and external devices in a serial manner, i.e., one bit at a time.**
- The 8085 has a serial I/O port (SID/SOD) for serial communication. The **SID (Serial Data Input)** pin is used for serial input. In SID, the RIM Instruction is initiated to input data in a serial manner. **Read Interrupt Mask (RIM)** instruction reads the current value of the **IMR (Interrupt Mask Register)** and copies it to the accumulator. The IMR is a register that controls which interrupts are allowed to be processed by the microprocessor.
- The **SOD (Serial Data Output)** pin is used for serial output. The **SIM (Set Interrupt Mask)** instruction should be initiated in order to output data in serial manner. SIM takes one byte as an operand and sets the corresponding bits in the IMR based on the value of the operand.

11. Serial Input/output control:

- The timing and control of serial communication is managed by the 8085's internal circuitry (Timing and control unit). The 8085 also has two special purpose registers, the **Serial Control Register (SC)** and the **Serial Shift Register (SS)**, which are used to control and monitor the serial communication. These are nothing but shift registers.

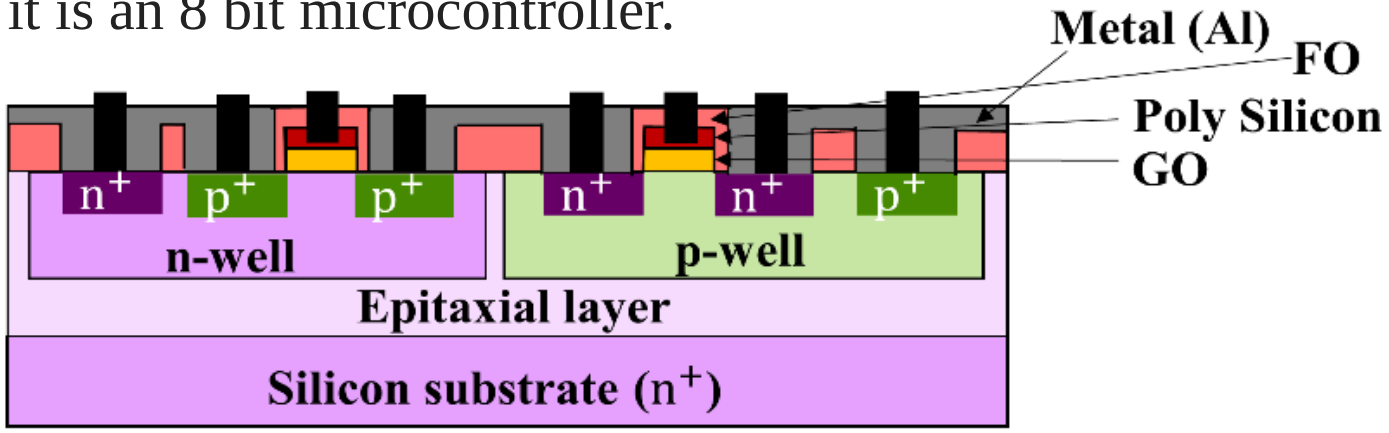


(16) (PDF) Area and Speed Efficient Layout Design of Shift Registers using Nanometer Technology (researchgate.net)

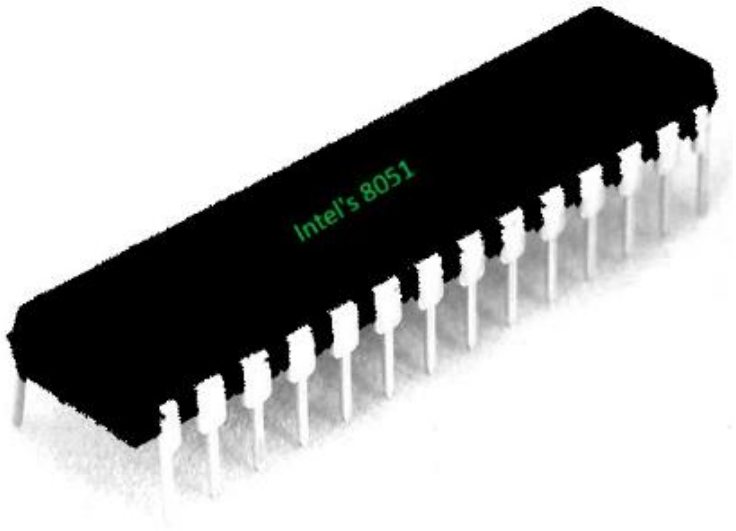
Intel 8051 microcontroller architecture

- When microprocessor and additional outside circuitry (memory, input/output devices, timers) are **built in a single semiconductor chip (System on a Chip (SoC))**, this is called as **microcomputer**. Microcomputers are used in control applications – they are also designated as **microcontroller**.
- The 8051 Microcontroller is an **8-bit microcontroller designed by Intel in 1981**. Number of transistors utilized in this microcontroller is ~10 K.

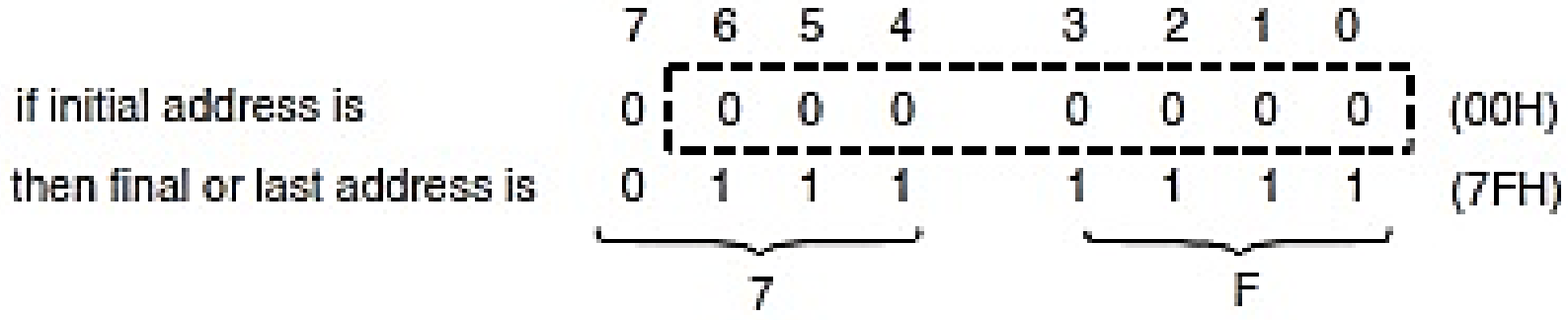
Normally, this microcontroller was developed using **NMOS** technology, which requires more power to operate. Therefore, Intel redesigned Microcontroller 8051 using **CMOS (Complementary Metal Oxide Semiconductor)** technology and their updated versions came with a letter C in their name, for instance an 80C51 it is an 8 bit microcontroller.



Intel 8051 microcontroller architecture

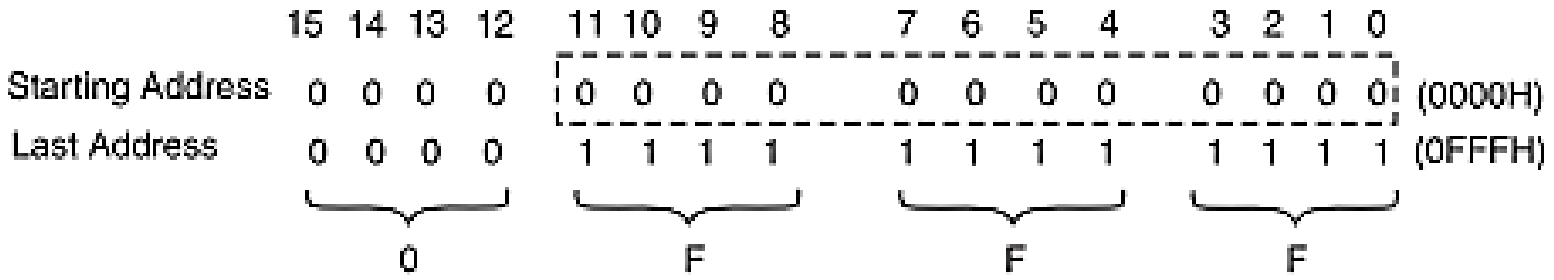


- It is based on a **Harvard architecture**.
- It includes 128 (2^7) bytes of Random Access memory (RAM).



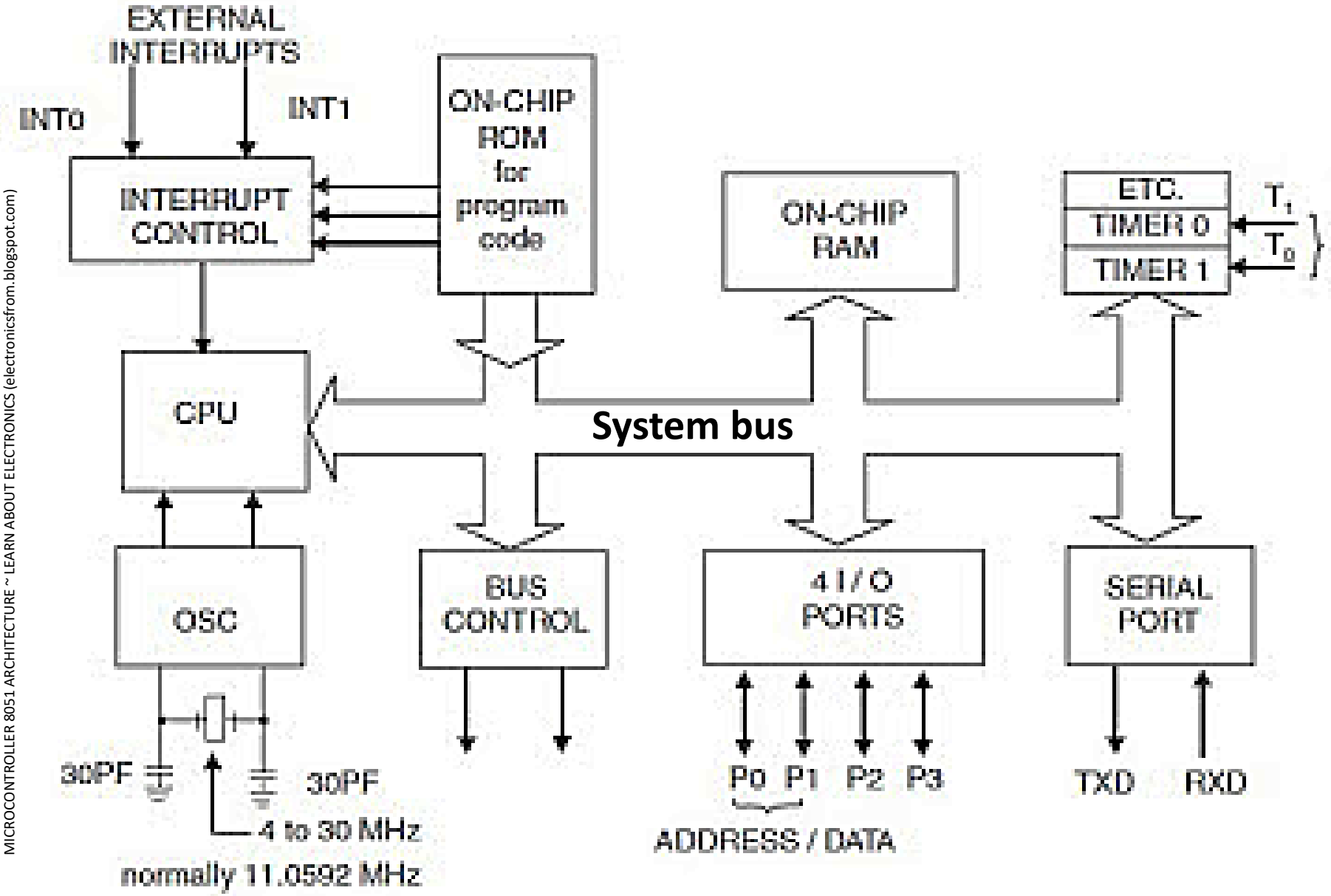
- It includes 4KB read only memory (ROM) is available for program storage. This is used for permanent data storage.

$$4\text{KB} = 2^2 \cdot 2^{10}\,\text{B}\,(\text{since } 1\text{KB} = 2^{10}\,\text{B})$$
$$= 2^{12}\,\text{Byte}$$



- It includes 4KB of on-chip flash memory, and an 8-bit CPU with an instruction set that includes a variety of arithmetic and logic operations.
- It has four parallel 8-bit ports that are programmable and addressable based on the requirement.
- It also includes two 16-bit timers and an on-chip crystal oscillator of frequency 12 MHz.
- A UART (Universal Asynchronous Receiver/Transmitter) receives serial data and stores it as parallel data (usually one byte), and takes parallel data and transmits it as serial data.

Intel 8051 microcontroller architecture



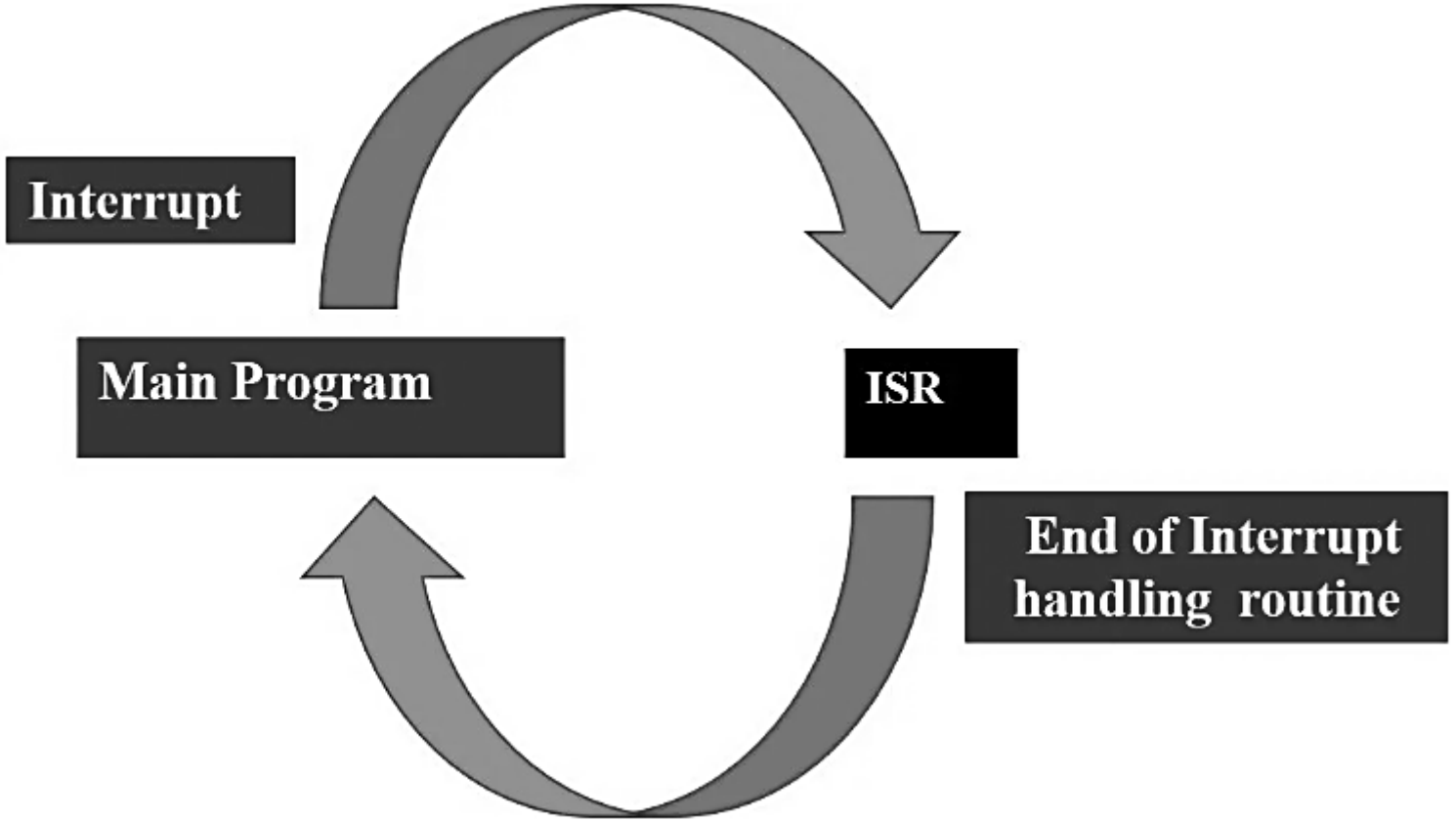
Intel 8051 microcontroller architecture

Central Processor Unit (CPU): As we know that the CPU is the **brain** of any processing device of the microcontroller. **It monitors and controls all operations that are performed on the Microcontroller units.** The User has no control over the work of the CPU directly . It reads program written in **program memory (ROM)**, fetch data from **data memory (RAM)** and executes them and do the expected task of that application. Generated output will be stored in data memory (RAM) or transferred to the output port.

Interrupts: The **interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention.** It alerts the microcontroller to a high-priority process requiring interruption of the current working process. microcontroller services the interrupt by executing a subroutine called **interrupt service routine (ISR)**. The feature of Interrupt is very useful as it helps in case of emergency operations. An Interrupts gives us a mechanism to put on hold the ongoing operations, execute a subroutine and then again resumes to another type of operations.

Intel 8051 microcontroller architecture

Interrupts: The Microcontroller 8051 can be configured in such a way that it temporarily terminates or pause the main program at the occurrence of interrupts. When a subroutine is completed, Then the execution of main program starts.



Intel 8051 microcontroller architecture

Interrupt Vector Table: A fixed memory area is assigned for each interrupt inside the microcontroller. The **Interrupt Vector Table contains the starting address of the memory location of every interrupt**.

When an interrupt occurs, the controller transfers the contents of the program counter onto the stack. Then it jumps to the memory location, which is specified by the Interrupt Vector Table (IVT).

The code written in this memory area by the programmer starts its execution. We refer to this code as an **Interrupt Service Routine (ISR)** or an interrupt handler on the 8051 microcontroller.

Interrupts	Memory Location	Pin	Flag Clearing
Reset	0000	9	Auto
Timer0	000B		Auto
Timer1	001B		Auto
INT0	0003	12	Auto
INT1	0013	13	Auto
Serial com	0023		Cleared by programmer

Intel 8051 microcontroller architecture

8051 Interrupt Types: The 8051 microcontroller can recognize **six different types of events**. These events can request the microcontroller to temporarily stop the execution of the current program and instead run a special block of code first. The interrupt sources present in 8051 microcontrollers are:

Reset interrupt: When the **reset pin activates, the program execution flow jumps to execute code from the 0000H memory location**. Generally, we don't use this pin. It is also known as power-on reset.

Timer0 overflow interrupt (TF0) and timer1 overflow interrupt (TF1): Two timers (T0 and T1) are present in the 8051 microcontroller and are responsible for a Timer interrupt. **A timer interrupt informs the microcontroller that the corresponding timer has finished counting.** Memory locations 000BH and 001BH in the interrupt vector table belong to Timer0 and Timer1, respectively.

Intel 8051 microcontroller architecture

8051 Interrupt Types:

External hardware interrupt (INT0 and INT1): There are two external interrupts (**INT0 and INT1**) to serve external devices. **Pin numbers 12 and 13 in port 3 are for the external hardware interrupts**. Both of these interrupts are **active low**. An **external interrupt informs the microcontroller that an external device needs routine service**. Memory locations 0003H and 0013H in the interrupt vector table belong to INT0 and INT1, respectively.

Serial communication interrupt (RI/TI): This interrupt is useful for serial communication. It has a single interrupt that belongs to both receive and transmit. **When enabled, it notifies the controller whether a byte has been received or transmitted.** This interrupt vector table assigns location 0023H to this interrupt.

Intel 8051 microcontroller architecture

Memory: Intel 8051 microcontroller is based on a **Harvard architecture.** The memory which is used to store the program of the microcontroller is known as **code memory or Program memory**. It is known as **ROM** memory. Microcontroller also requires a memory to store data or operands temporarily. The **data memory** of the 8051 is used to store data temporarily for operation is known **RAM** memory. 8051 microcontroller has 4KB of code memory or program memory, that has 4KB ROM and also 128 bytes of data memory of RAM.

Intel 8051 microcontroller architecture

BUS: Basically **Bus is a collection of wires which work as a communication channel or medium for transfer of Data**. These buses consists of 8, 16 or more wires of the microcontroller. Thus, these can carry 8 bits,16 bits simultaneously.

Address Bus: Microcontroller 8051 has a **16 bit address bus** for transferring the data. It is used to address memory locations and to transfer the address from CPU to Memory of the microcontroller. **It has four addressing modes that are Immediate addressing modes, Bank address (or) Register addressing mode, Direct Addressing mode, Register indirect addressing mode.**

Data Bus: Microcontroller 8051 has 8 bits of the data bus, which is used to carry data of particular applications.

Intel 8051 microcontroller architecture

Oscillator: Generally, we know that the microcontroller is a computing device, therefore it requires clock pulses for its operation of microcontroller applications. For this purpose, microcontroller 8051 has an **on-chip crystal oscillator of frequency 12 MHz** which works as **a clock source for Central Processing Unit** of the microcontroller. The output pulses of oscillator are stable. Therefore, it enables synchronized work of all parts of the 8051 Microcontroller.

Input/Output Port: Normally microcontroller is used in embedded systems to control the operation of machines in the microcontroller. Therefore, to connect it to other machines, devices or peripherals we require I/O interfacing ports in the microcontroller interface. For this purpose microcontroller 8051 has 4 input, output ports to connect it to the other peripherals

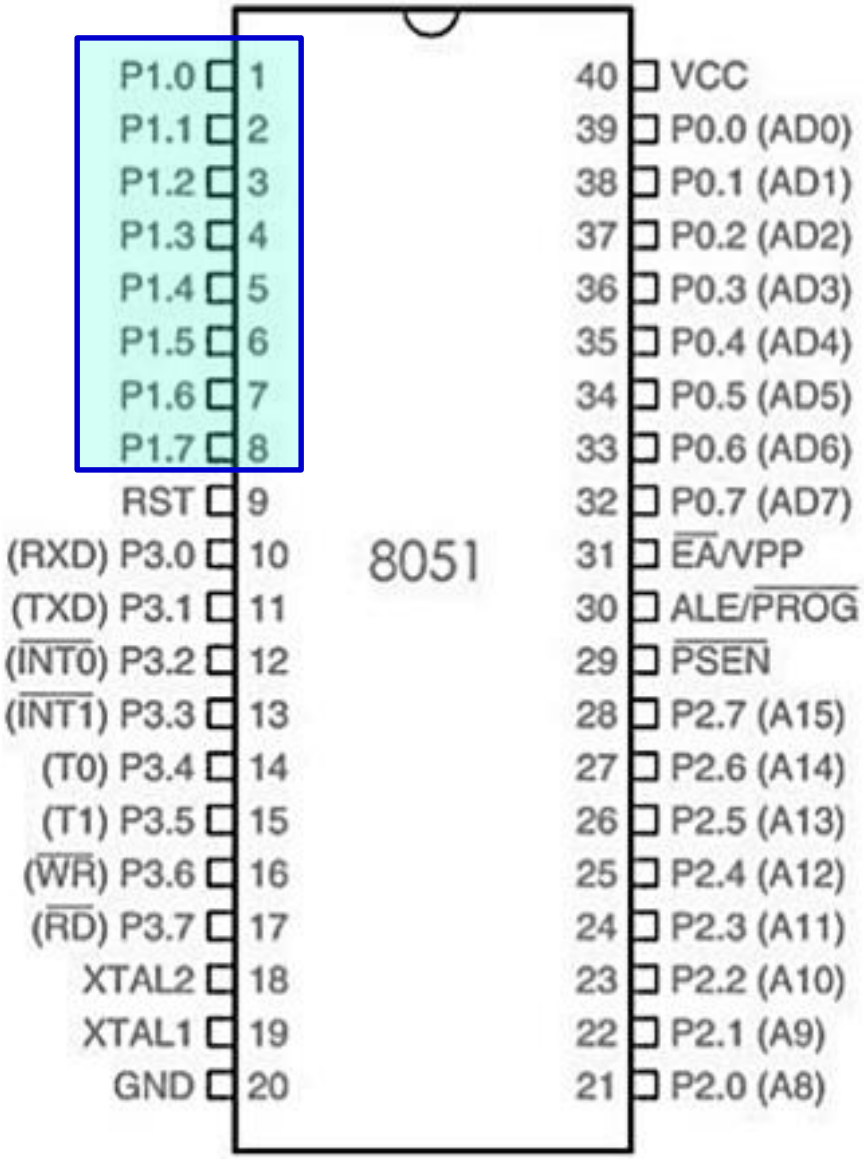
Intel 8051 microcontroller architecture

8051 microcontroller has two 16 bit timers and counters. These counters are again divided into a 8 bit register. The timers are used for measurement of intervals to determine the pulse width of pulses.

Pin diagram of 8051 Microcontroller

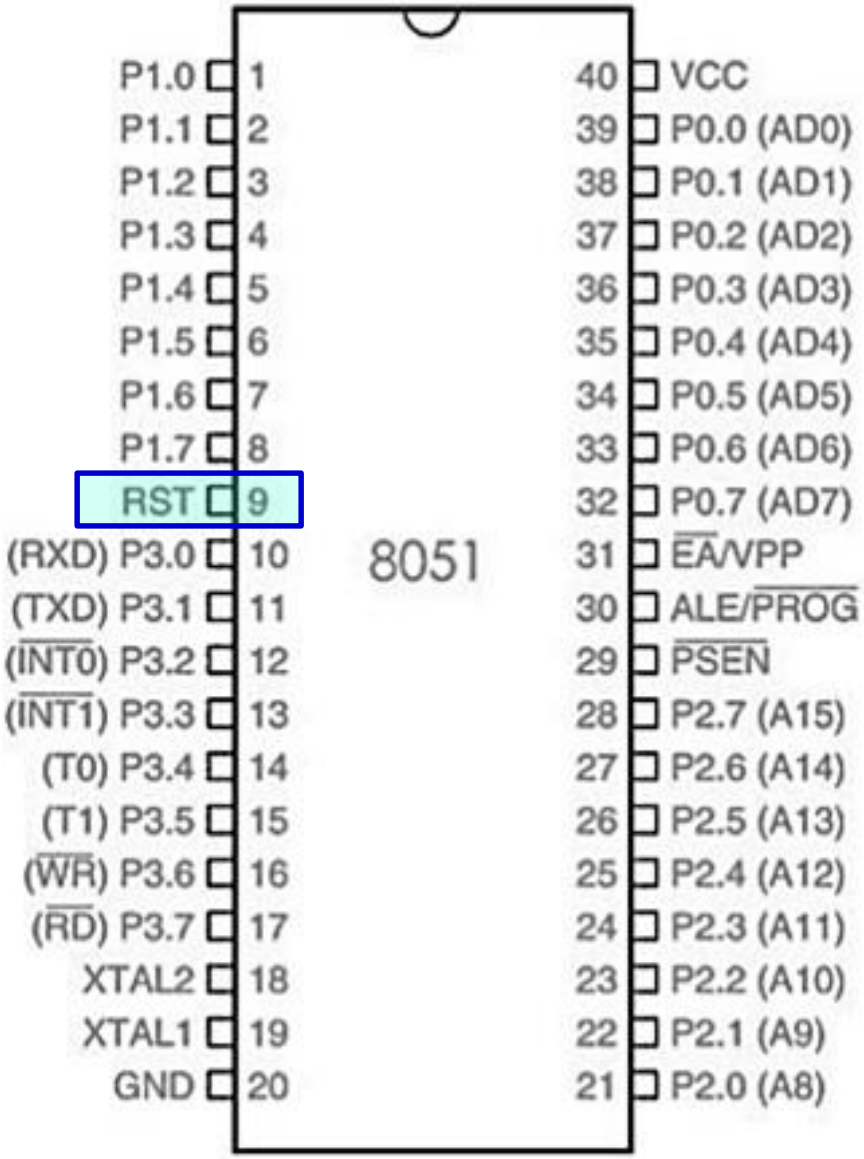
40 - PIN DIP

Pin diagram of 8051 Microcontroller



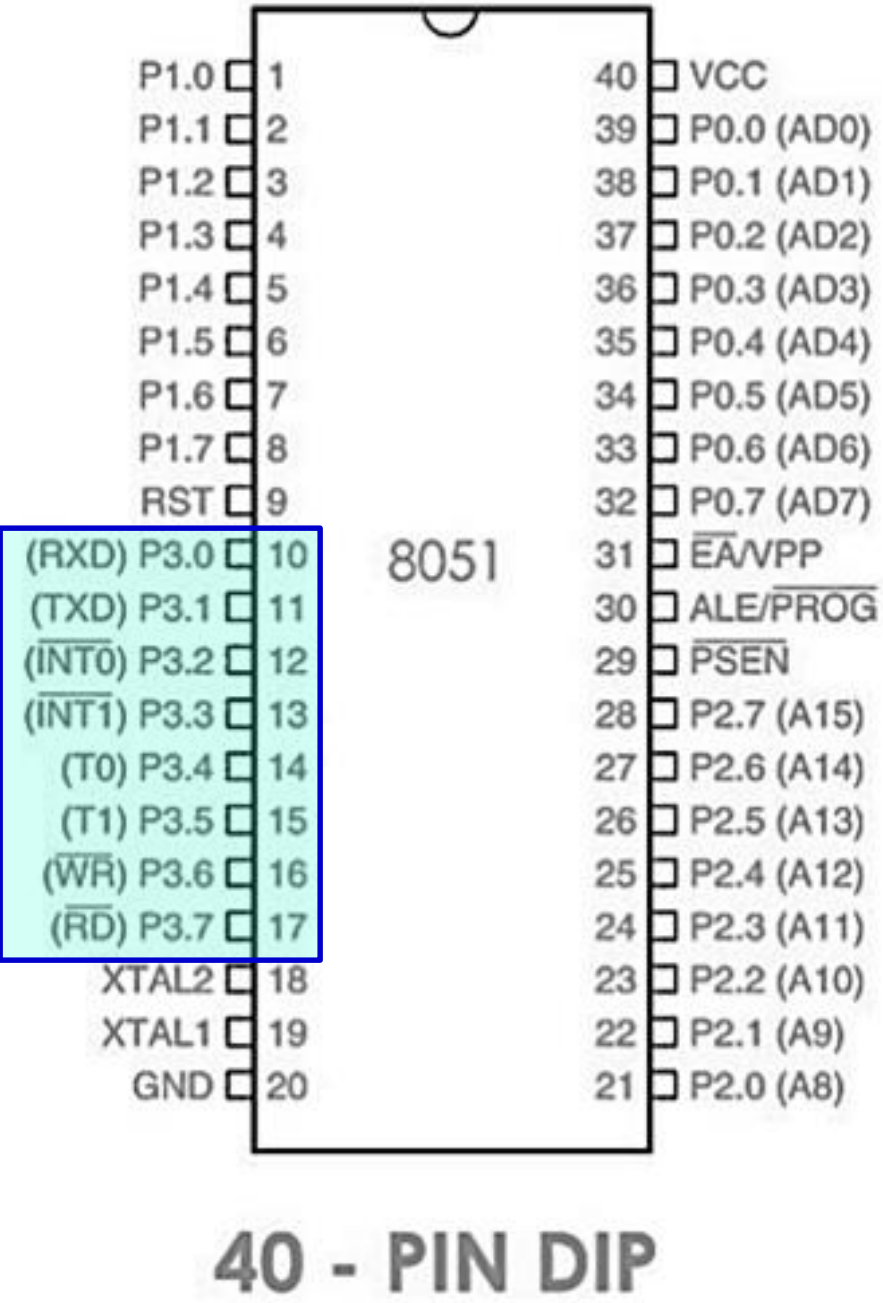
•**Pin 1 to Pin 8 (Port 1) – Pin 1 to Pin 8 are assigned to Port 1 for simple I/O operations.** They can be configured as input or output pins depending on the logic control i.e. **if logic zero (0) is applied to the I/O port it will act as an output pin** and **if logic one (1) is applied the pin will act as an input pin.** These pins are also referred to as P1.0 to P1.7 (where **P1 indicates that it is a pin in port 1** and **the number after '.' tells the pin number i.e. 0 indicates first pin of the port**. So, P1.0 means first pin of port 1, P1.1 means second pin of the port 1 and so on). These pins are bidirectional pins.

Pin diagram of 8051 Microcontroller



•**Pin 9 (RST) – Reset pin**. It is an **active-high, input pin**. Therefore if the **RST pin is high for a minimum of 2 machine cycles**, the microcontroller will reset i.e. it will close and terminate all activities. It is often referred as "power-on-reset" pin because it is used to reset the microcontroller to it's initial values when power is on (high).

Pin diagram of 8051 Microcontroller

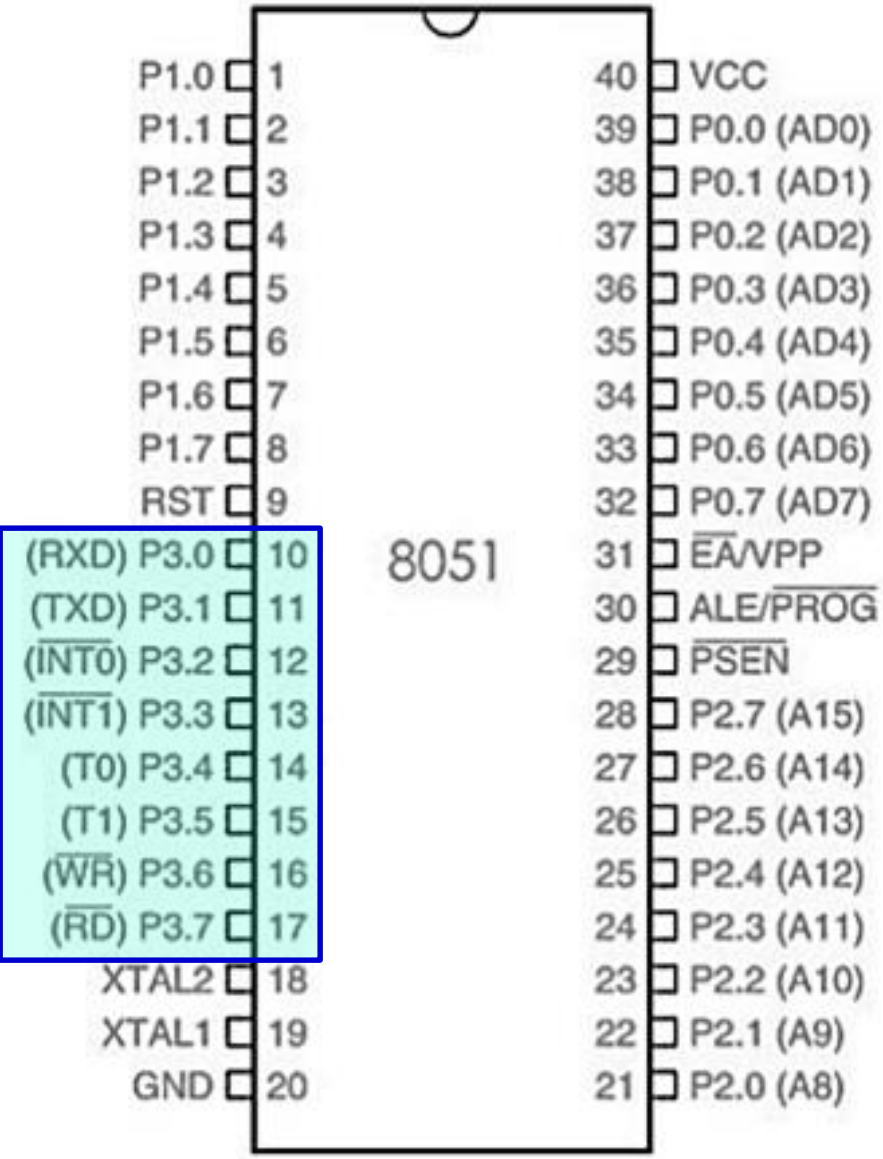


Pin 10 to Pin 17 (Port 3) – Pin 10 to pin 17 are port 3 pins which are also referred to as P3.0 to P3.7. These pins are similar to port 1 and can be used as universal input or output pins. These pins are bidirectional pins. These pins also have some additional functions which are as follows:

P3.0 (RXD): 10th pin is RXD (**serial data receive pin**) which is for **serial input**. Through this input signal microcontroller receives data for serial communication.

P3.1 (TXD): 11th pin is TXD (**serial data transmit pin**) which is **serial output pin**. Through this output signal microcontroller transmits data for serial communication.

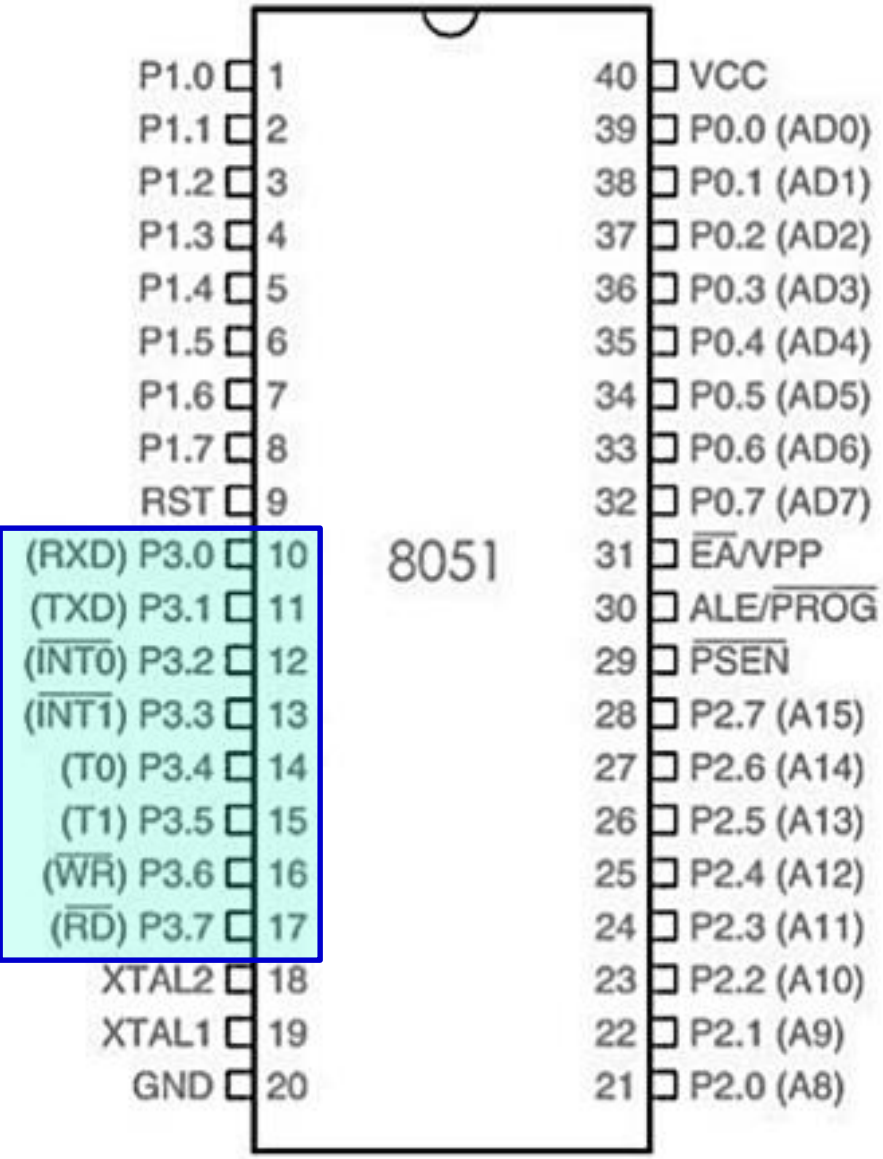
Pin diagram of 8051 Microcontroller



P3.2 and P3.3 (INT0′, INT1′) : 12th and 13th pins are for **External Hardware Interrupt 0 and Interrupt 1 respectively**. When this interrupt is activated(i.e. when it is low), 8051 gets interrupted in whatever it is doing and jumps to the vector value of the interrupt (**0003H for INT0 and 0013H for INT1**) and starts performing Interrupt Service Routine (ISR) from that vector location.

P3.4 and P3.5 (T0 and T1) : 14th and 15th pin are for **Timer 0 and Timer 1 external input**. **They can be connected with 16 bit timer/counter.**

Pin diagram of 8051 Microcontroller

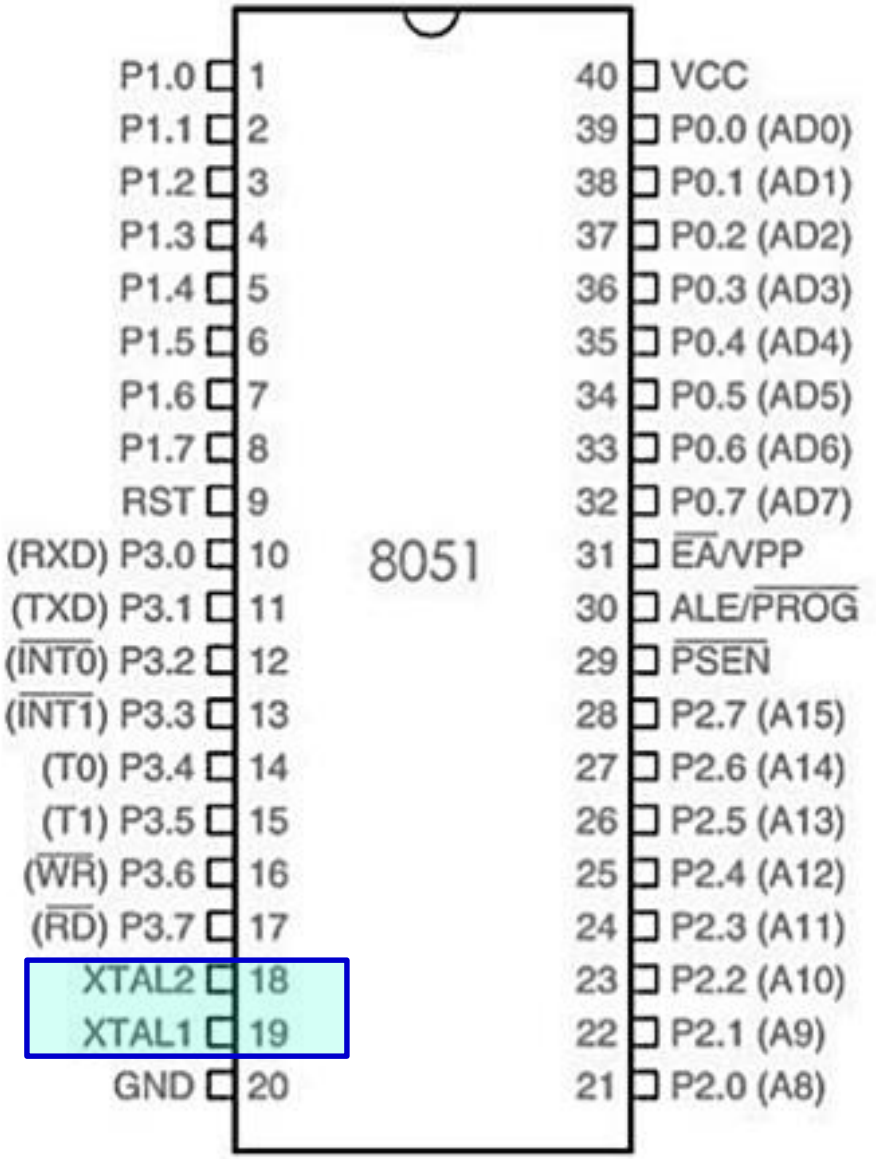


40 - PIN DIP

P3.6 (WR') : 16th pin is for **external memory write** i.e. writing data to the external memory.

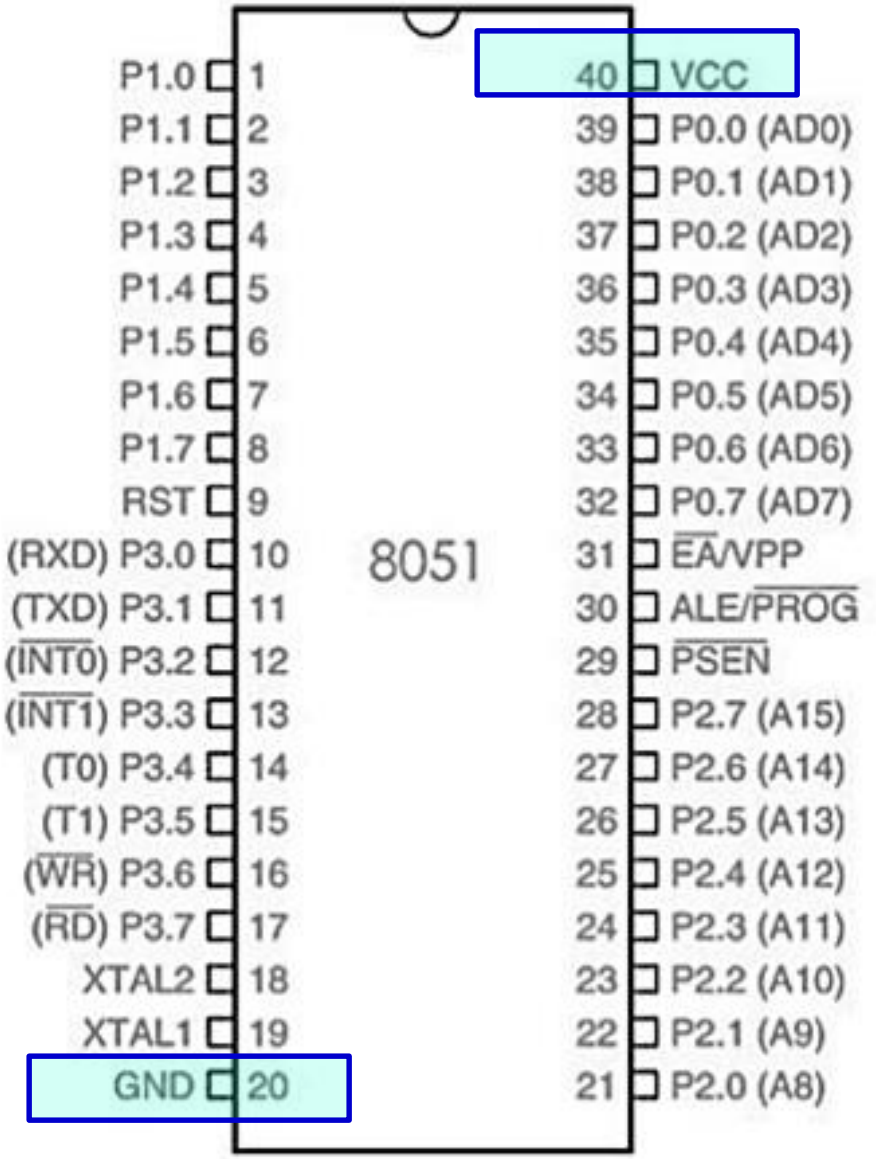
P3.7 (RD') : 17th pin is for **external memory read** i.e. reading data from external memory.

Pin diagram of 8051 Microcontroller



Pin 18 and Pin 19 (XTAL2 And XTAL1) – These pins are **connected to an external oscillator** which is generally a quartz crystal oscillator. They are used to provide an external clock frequency of **4MHz to 30MHz**.

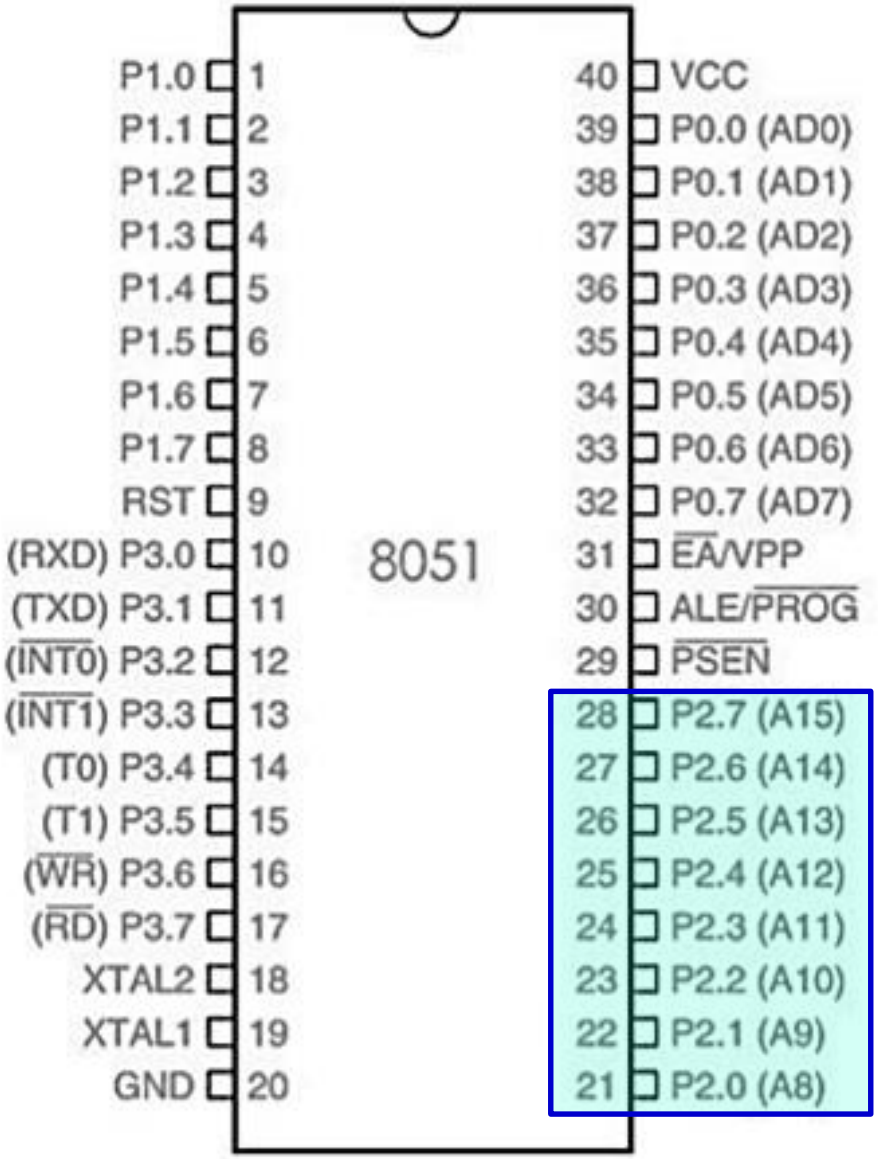
Pin diagram of 8051 Microcontroller



Pin 20 (GND) – This pin is connected to the ground. It has to be provided with 0V power supply. Hence it is connected to the negative terminal of the power supply.

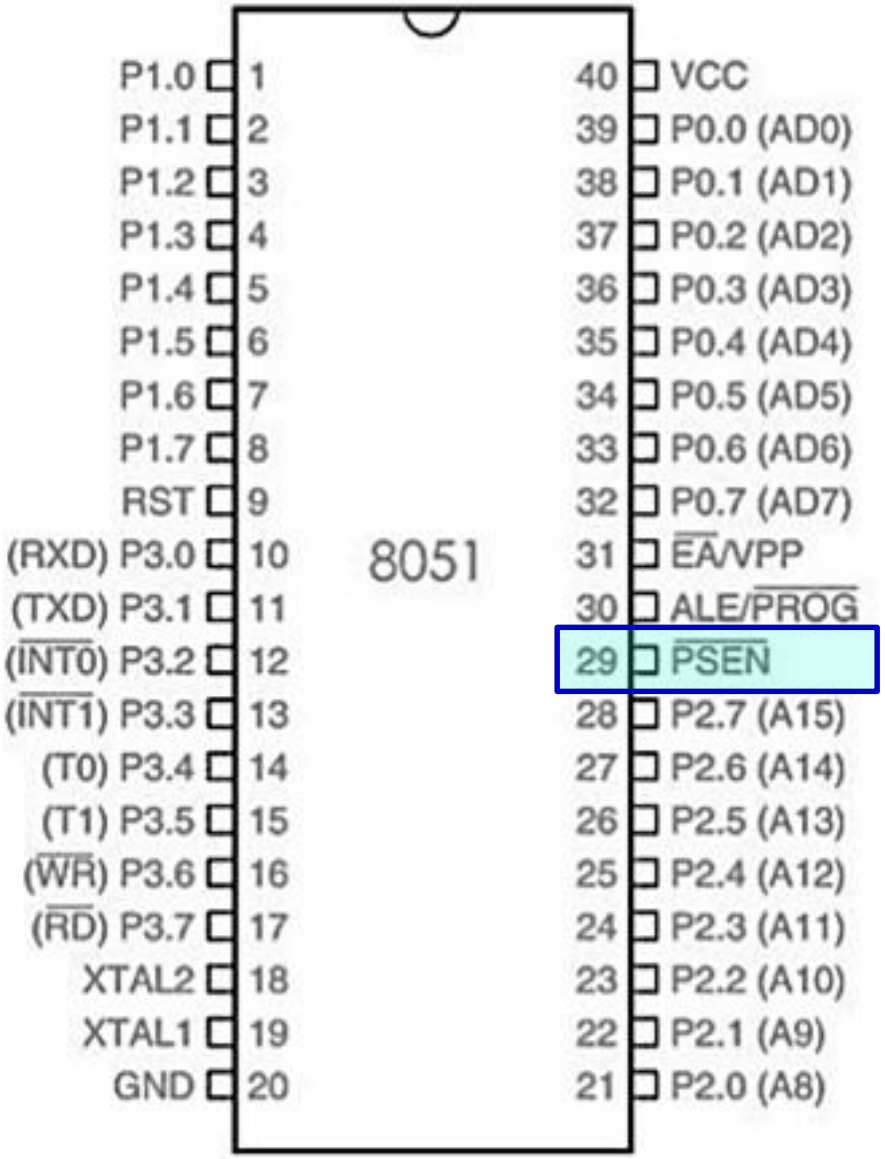
Pin 40 (VCC) – This pin provides power supply voltage i.e. +5 Volts to the circuit.

Pin diagram of 8051 Microcontroller



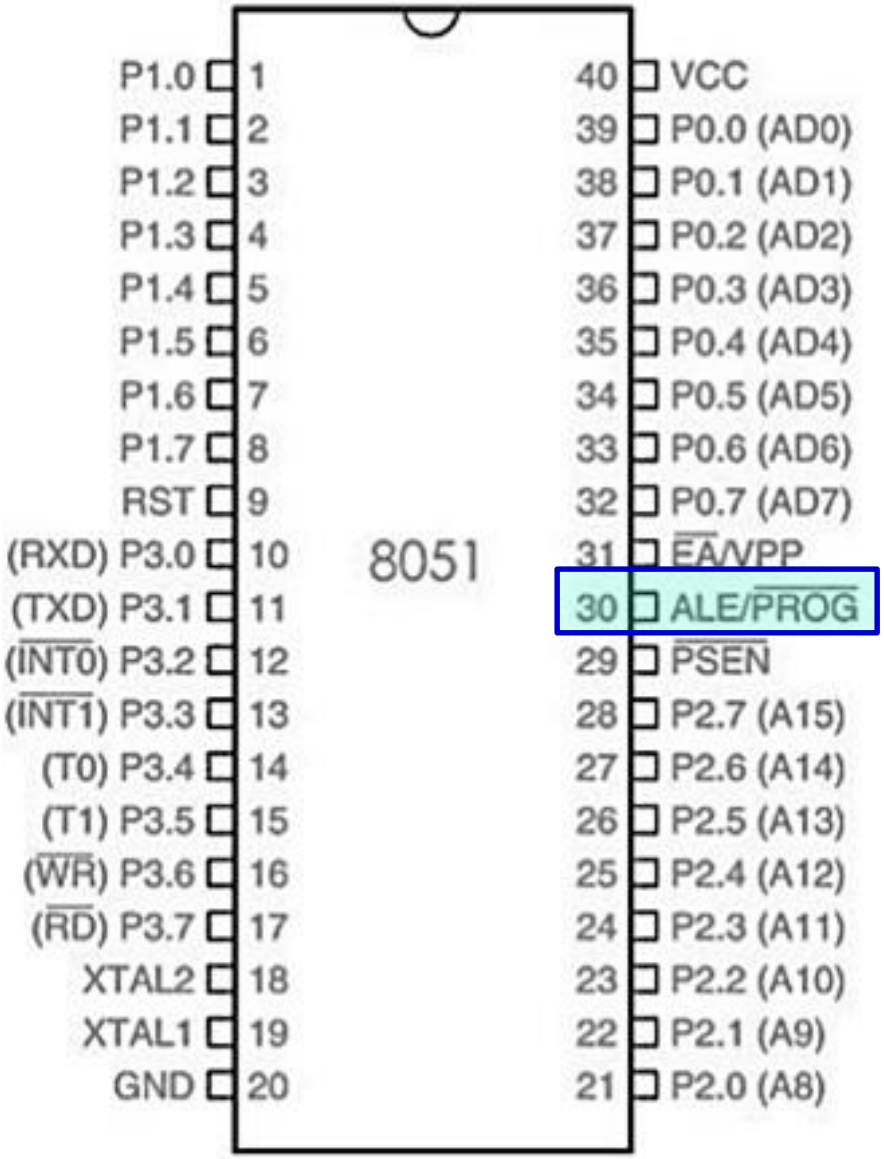
Pin 21 to Pin 28 (Port 2) – Pin 21 to pin 28 are **port 2 pins** also referred to as P2.0 to P2.7. **When additional external memory is interfaced with the 8051 microcontroller, pins of port 2 act as higher-order address bytes. These pins are bidirectional.**

Pin diagram of 8051 Microcontroller



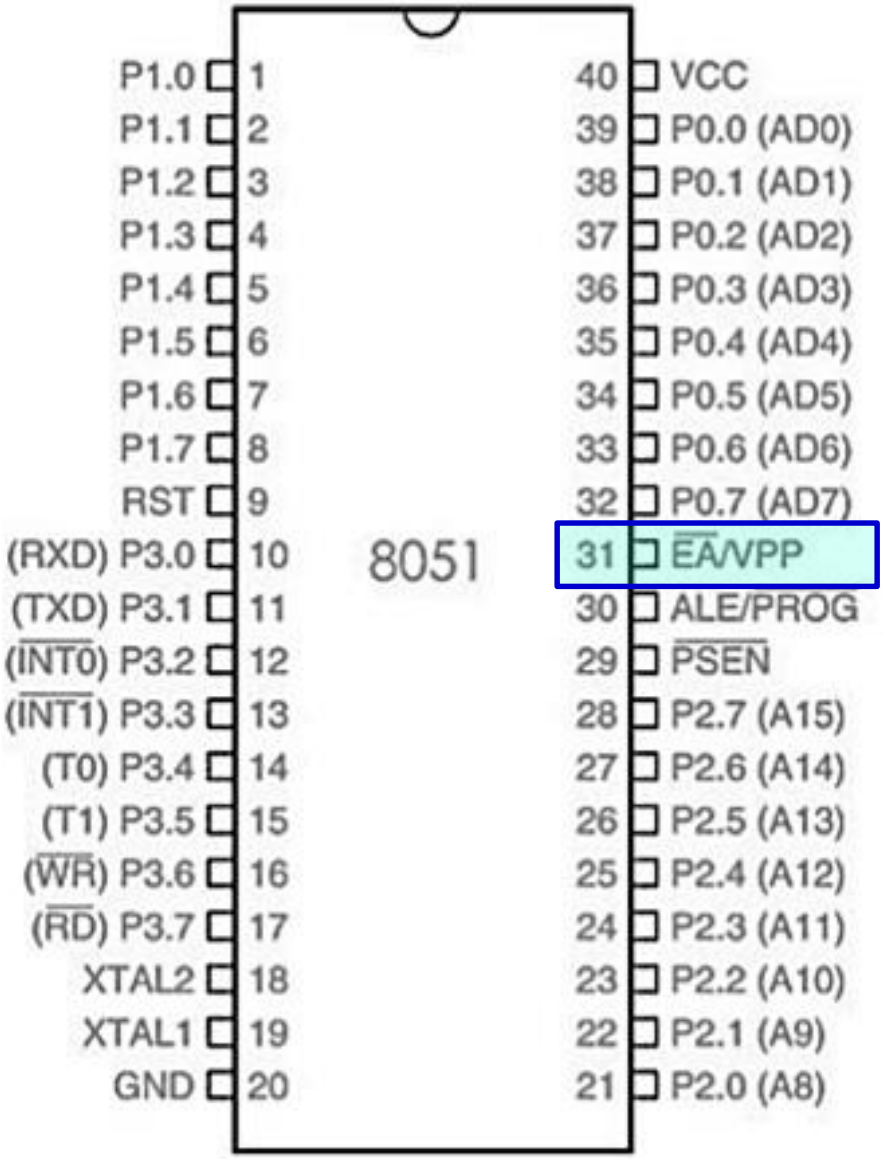
Pin 29 (PSEN) – PSEN stands for **Program Store Enable**. **It is output, active-low pin. This is used to read external memory**.

Pin diagram of 8051 Microcontroller



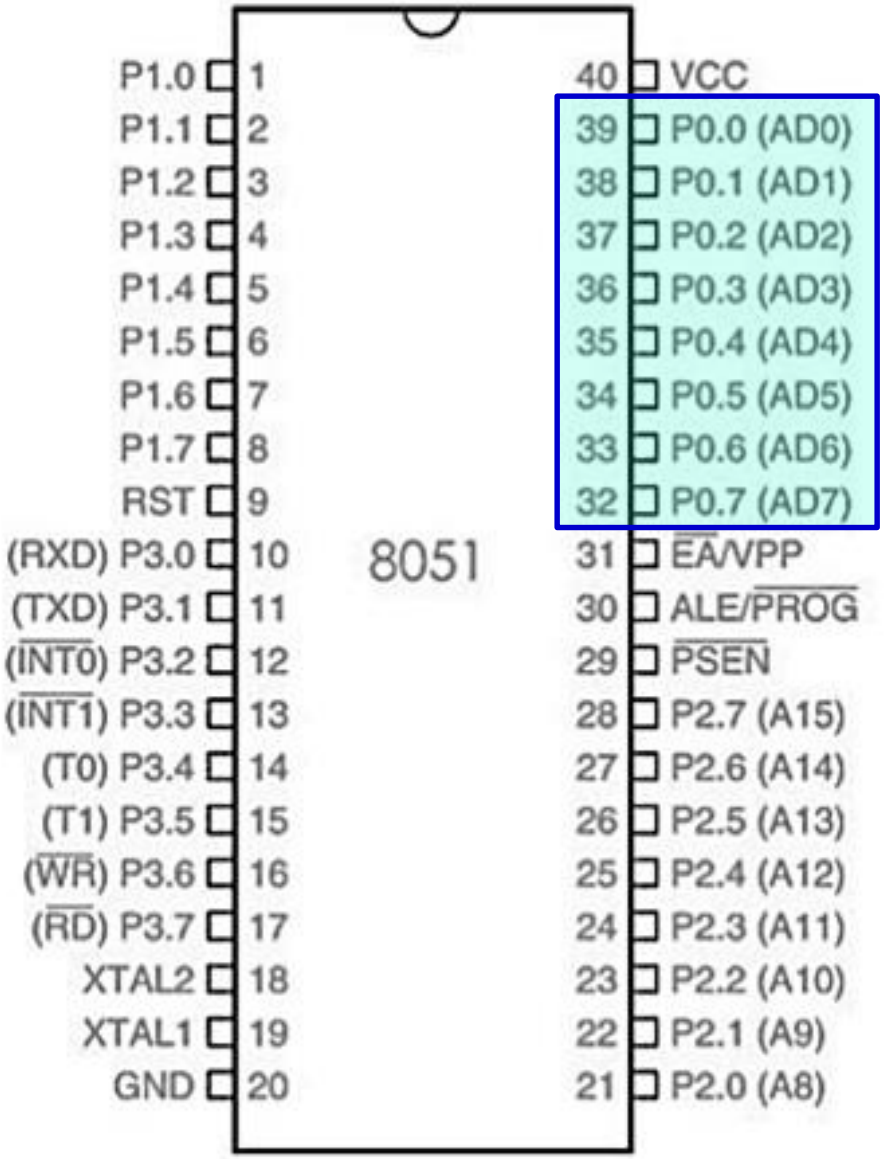
Pin 30 (ALE/ PROG) – ALE stands for **Address Latch Enable**. **It is input, active-high pin**. This pin is **used to distinguish between memory chips when multiple memory chips are used**. It is also **used to de-multiplex the multiplexed address and data signals available at port 0**. During flash programming i.e. Programming of EPROM, this pin acts as program pulse input (PROG).

Pin diagram of 8051 Microcontroller



Pin 31 (EA/ VPP) – EA stands for **External Access input**. It is **used to enable/disable external memory interfacing**. In 8051, EA is connected to Vcc as it comes with on-chip ROM to store programs.

Pin diagram of 8051 Microcontroller



Pin 32 to Pin 39 (Port 0) – Pin 32 to pin 39 are port 0 pins also referred to as P0.0 to P0.7. **They are bidirectional input/output pins. Port 0 is also designated as AD0-AD7 because 8051 multiplexes address and data through port 0 to save pins.**

Difference between RISC and CISC architecture

Parameters	RISC	CISC
Full form	Reduced Instruction Set Architecture	Complex Instruction Set Architecture
Instruction Size	Fixed size	Variable size
Instruction fetch time	Same for all instruction	Vary with respect to instructions
Instruction set	Small and simple	Large and complex
Dealing with	Registers	Registers as well as memory
Addressing modes	Less modes as most instructions are based on registers. Only the load and store operation deals with memory.	More modes as complex instructions are available with different verities.

Difference between RISC and CISC architecture

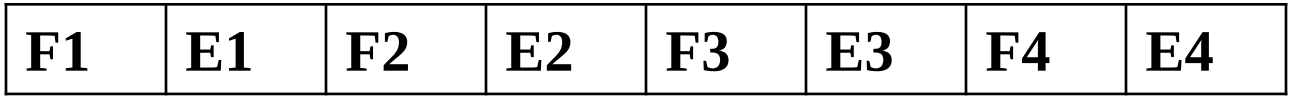
Parameters	RISC	CISC
Number of registers	Many	Few
Compiler design	Simple: small instruction set and less number of addressing modes	Complex: instruction set is complex and many addressing modes are required
Program size	Long (Weak code density)	Small (Better code density): complex instructions are available
Number of operands	Fixed (Mainly in registers)	Variable (Can be in registers and memory)
Control unit	Hardware controlled	Micro program controlled: if we use hardware control it will be very costly

Difference between RISC and CISC architecture

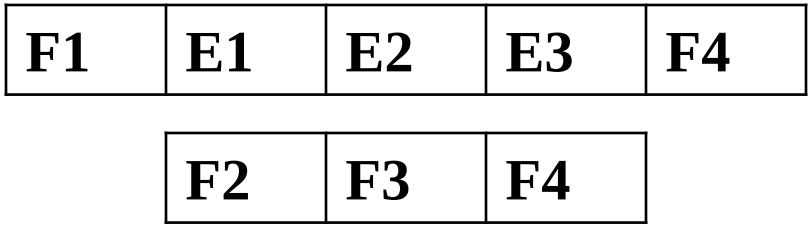
Parameters	RISC	CISC
Execution speed	Faster	Slower
Pipelining	Efficient: As instruction sizes are fixed	Inefficient: As instruction sizes are variable

Pipelining

Instruction execution without pipelining



Instruction execution with pipelining



ARM microcontroller architecture

ARM is short for "**Advanced RISC Machines**". The ARM processor belongs to the family of CPUs which are based primarily on **Reduced Instruction Set Computer (RISC)**. The ARM processors could be of **32 bit or 64 bit**. The RISC processors are higher in speed because they perform a small number of instructions.

The **ARM processors have a less number of transistors because they have a reduced instruction set**, which allows a **smaller size for the IC**. Thereby being **space efficient** also. Most of the electronic devices such as **tablets, mobiles, smart phones and other mobile devices** consist of these processors.

By combining the ARM microprocessor with RAM, ROM and other peripherals in one single chip, we get an ARM microcontroller, for example, LPC2148.

Developed by Acorn computer limited.

ARM microcontroller architecture

ARM architecture is basically a **RISC architecture** (large uniform register file, load/store architecture, simple addressing modes, uniform and fixed length instruction fields).

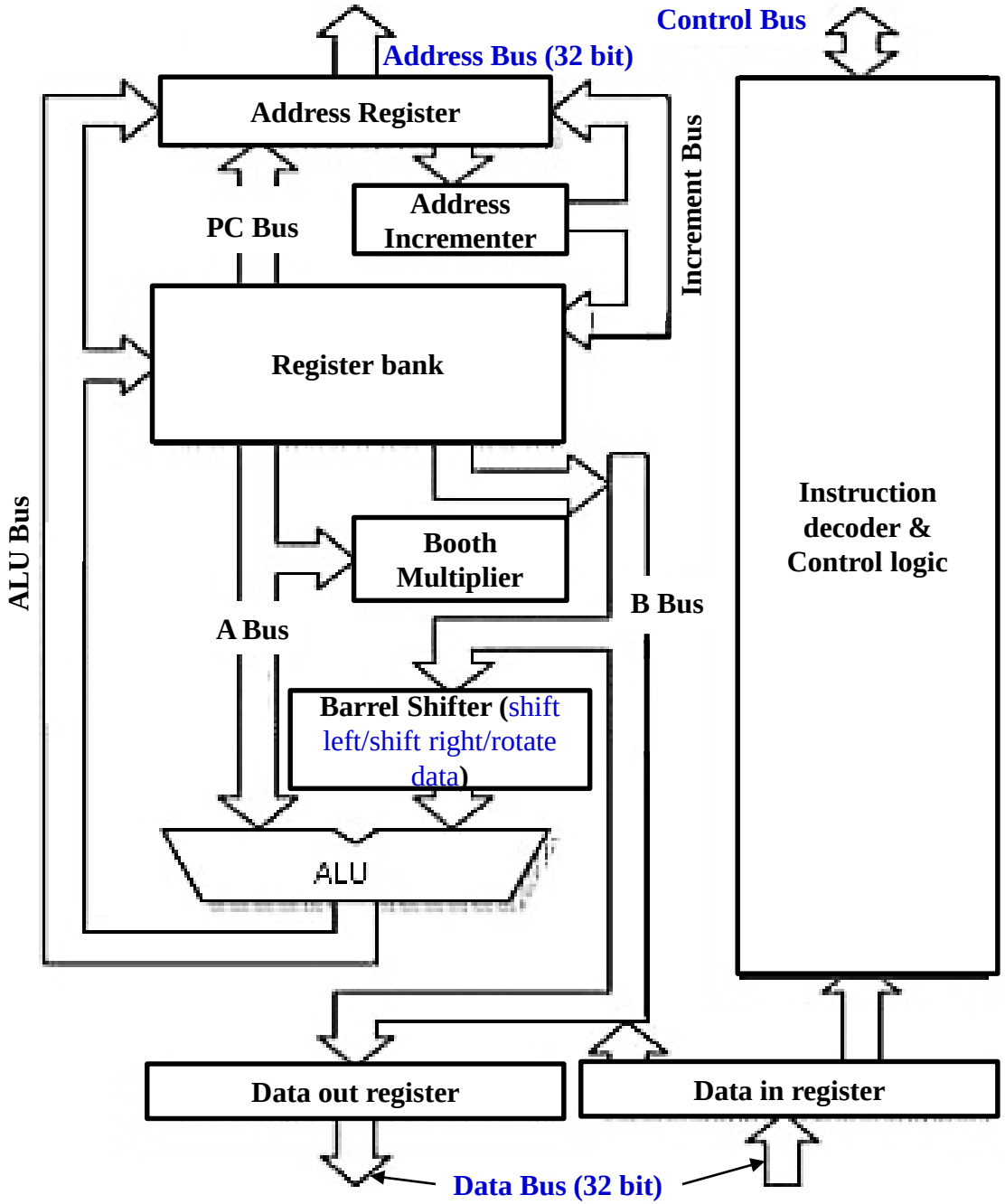
Advancements, that have done in **ARM architecture** compared to any other RISC architecture is that each instruction controls the ALU and shifter, auto increment and auto decrement addressing modes, multiple load/store architecture (conditional execution).

Features of ARM architecture:

- High performance
- Low code size
- Low power consumption
- Consists of a 3 stage pipeline which fetches the instructions, then decodes it and then finally executes the instruction
- Low silicon area in IC chip

ARM microcontroller implements 2 type of instruction sets: 32 bit ARM instruction set and 16 bit thumb instruction set.

ARM microcontroller architecture



ARM microcontroller architecture

1. **Arithmetic and Logic Unit (ALU):** It is a combinational logic circuit that performs Arithmetic Operations and Logical operations.
2. **Booth multiplier:** It multiplies two signed binary numbers in 2's complement notation.
3. **Barrel shifter:** 32 bit barrel shifter is digital circuit (combinational) that can shift data (shift left/shift right) by a specific number of bits without the use of any sequential logic. Barrel shifter performs preprocessing of data before execution of a particular operation in ALU. Thus **it speeds up the execution process**.
4. **Control unit:** It is responsible for the system operation and it controls the flow of data between processor and other devices.

ARM microcontroller architecture

5. **Register file:** ARM microcontroller has **load (copy data from memory to register)/store (copy data from register to memory) RISC architecture**. Here, arithmetic and logical operation only use register operands and cannot directly operate with memory locations.

ARM Cortex M3 processor consist of 37 register set (31 general purpose register (GPR) and 6 status register). These registers are part of the programmer's model.

a. Core Registers: These are the fundamental registers used for general-purpose computation and control. They include:

- **General-purpose registers (R0-R12):** Used for temporary storage during arithmetic operations.
- **R13: Stack Pointer (SP):** Points to the top of the stack.
- **R14: Link Register (LR):** Stores the return address after a function call.
- **R15: Program Counter (PC):** Holds the address of the next instruction to be executed.

ARM microcontroller architecture

b. Special Registers:

- **xPSR (Program Status Register):** Contains flags and status information.
- **MSP (Main Stack Pointer):** Used for exception handling.
- **PSP (Process Stack Pointer):** Used for thread mode.
- **Control:** Determines the privilege level and stack selection.

c. Floating-Point Registers:

- **S0-S15:** Single-precision floating-point registers.
- **FPSCR (Floating-Point Status and Control Register):** Manages floating-point exceptions.

d. Debug Registers:

- **DHCSR (Debug Halting Control and Status Register):** Controls debugging features.
- **DCRSR (Debug Core Register Selector Register):** Selects the register to access during debugging.

Difference between 8051 and ARM microcontroller

8051 Microcontroller	ARM microcontroller
8 bit for standard core bus width is present in 8051 microcontroller – average performance.	32 bit/64 bit bus width is present in ARM microcontroller – high performance.
Average power consumption and less expensive.	Less power consumption and more expensive.
It is based on CISC Instruction set Architecture.	It is based on RISC Instruction Set Architecture.
Smaller number of built-in peripherals.	Greater number of built-in peripherals.
Von Neumann architecture.	Modified Harvard architecture.
Flash, ROM, SRAM memory is used in 8051 microcontroller.	Flash, EEPROM, SDRAM memory is used in ARM microcontroller.
UART, USART, I2C, SPI, communication protocols are used.	UART, USART, Ethernet, I2S, DSP, SPI, CAN, LIN, I2C communication protocols are used.

Difference between 8051 and ARM microcontroller

8051 Microcontroller	ARM microcontroller
8051 microcontroller costs very low as compared to features provided.	ARM microcontroller costs low as compared to features provided.
Developed by Intel.	Developed by Acorn computer limited.

PIC microcontroller architecture

PIC stands for **Peripheral Interface Controller**. PIC microcontroller was developed by **microchip technology** in 1993. It was developed for supporting PDP computers to control its peripheral devices and that's why it was named Peripheral Interface Controller.

PIC microcontrollers are of **low cost, very fast and easy for the programming and execution of program**. Their interfacing with other peripherals is also very easy.

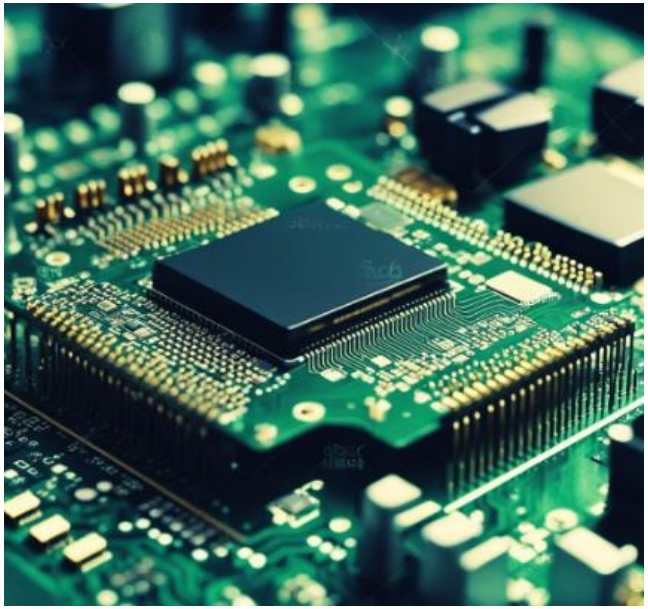
PIC Microcontrollers from Microchip Company are divided into 4 large families.

- **First family:** PIC10 (10FXXX) called Low End
- **Second family:** PIC12 (PIC12FXXX) called Mid-Range
- **Third family:** PIC16 (16FXXX)
- **Fourth family:** PIC 17/18 (18FXXX)

Each family has a variety of components along with built in special features. It offers a lot of memory sizes and pin packages and different clock ratings.

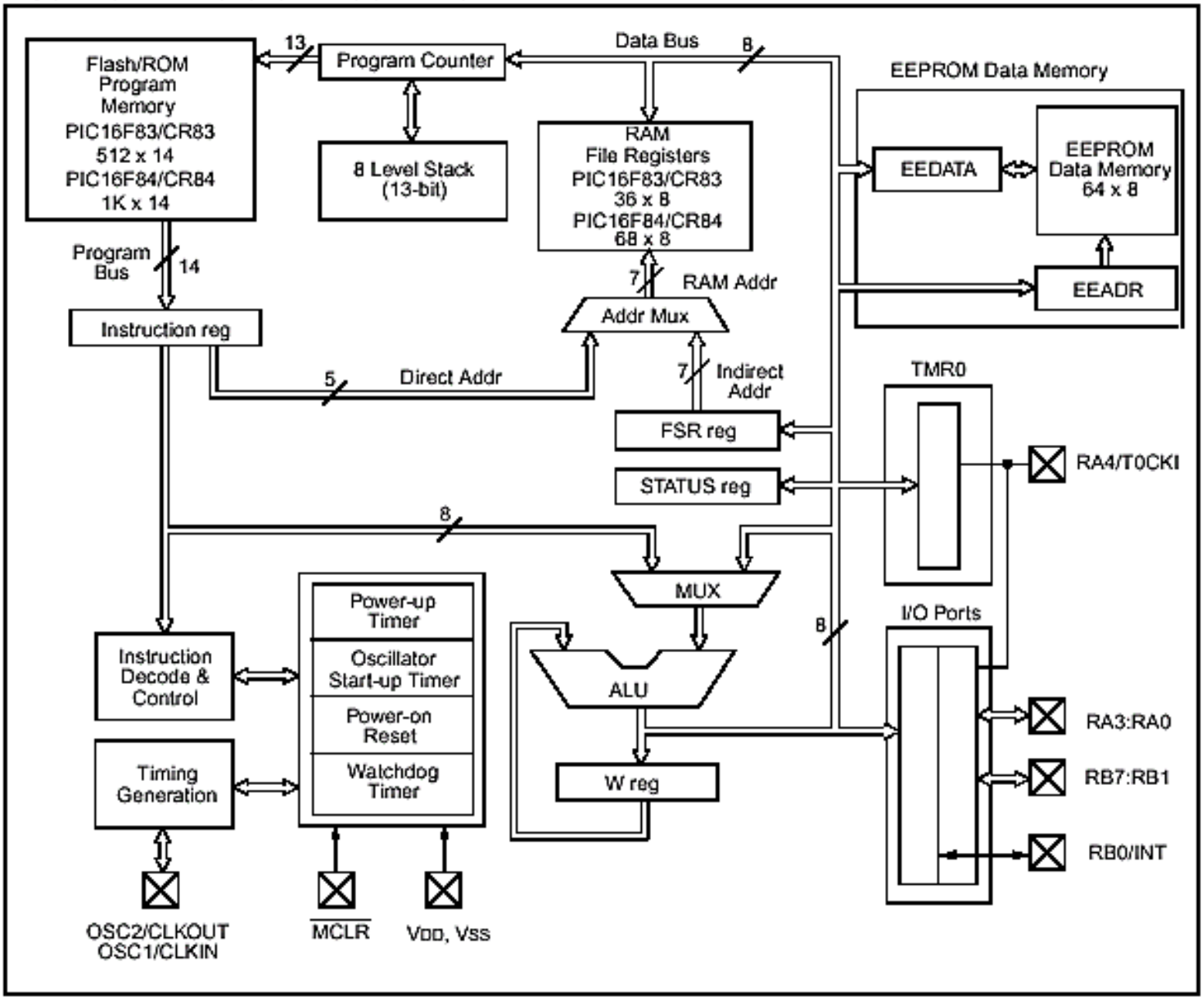
PIC microcontroller architecture

PIC Microcontroller architecture is based on **Harvard architecture** and supports **RISC architecture (Reduced Instruction Set Computer)**.
PIC microcontroller architecture consists of memory organization (ram, rom, stack), CPU, timers, counter, ADC, DAC, serial communication, CCP module and I/O ports. PIC microcontroller also supports the protocols like CAN, SPI, UART for interfacing with other peripherals. PIC microcontrollers can be programmed using a variety of programming languages, including assembly language, C, and C++.



PIC microcontroller architecture block diagram

FIGURE 3-1: PIC16F8X BLOCK DIAGRAM



PIC microcontroller architecture

1. CPU (Central Processing Unit):

PIC microcontroller's CPU consists of

- Arithmetic logic unit (ALU)
- Memory unit (MU)
- Control unit (CU)
- Accumulator

ALU is used for arithmetic operations and for logical decisions. Memory is used for storing the instructions after processing. Control unit is used to control the internal and external peripherals which are connected to the CPU and accumulator is used for storing the results.

2. Memory organization:

PIC microcontroller memory module consists of mainly 3 types of memories: Program Memory, Data Memory, Data EEPROM.

PIC microcontroller architecture

2.1. Program Memory:

It contains the written program after we burned it in microcontroller. Program Counter executes commands stored in the program memory, one after the other. PIC microcontroller can have 8K words x 14 bits of Flash program memory that can be electrically erased and reprogrammed. Whenever we burn program into the micro, we erase an old program and write a new one.

2.2. Data Memory:

It is a RAM type which is used to store the data temporarily in its registers. The RAM memory is classified into banks. Each bank extends up to 7Fh (128 bytes). Number of banks may vary depending on the microcontroller. PIC16F84 has only two banks. Banks contain **Special Function Registers (SFR)** and **General Purpose Registers (GPR)**. The lower locations of each bank are reserved for the Special Function Registers and upper locations are for General Purpose Registers.

PIC microcontroller architecture

2.2.1.General Purpose Registers (GPR):

These registers don't have any special function. These are used for general purpose for multiplying, addition or subtraction and then storing the results in other registers. CPU can easily access the data in these registers.

2.2.2. Special Function Registers (SFR):

These registers are used for special purposes and they cannot be used as normal registers. **Their function is set at the time of manufacturing**. They perform the function assigned to them and user cannot change the function of SFR. These registers are used by the CPU and peripheral modules for controlling the desired operation of the device. SFRs are the gateway to interaction between the CPU and the peripherals. Three important SFRs for programming are:

- **STATUS register:** It changes the bank
- **PORT registers:** It assigns logic values 0 or 1 to the ports
- **TRIS registers:** It is a data direction register for input and output

PIC microcontroller architecture

2.3. Data EEPROM:

This memory allows storing the variables as a result of burning the written program. It is readable and writable during normal operation (over the full VDD range). This memory is not directly mapped in the register file. It is indirectly addressed through the SFRs. There are six SFRs which are used to read and write to this memory:

- **EECON1:** is memory accessible control register. The control bit, EEPGD, determines the type of memory that will be access (program memory or data memory). When clear, as it is when reset, any subsequent operations will operate on the data memory. When set, any subsequent operations will operate on the program memory.

- **EECON2:** is used exclusively in the EEPROM write sequence.

PIC microcontroller architecture

- **EEDATA:** holds 8 bit data (for reading and writing)
- **EEDATH:** EEDATA and EEDATH registers form a two-byte word that holds the 14-bit data for read/write
- **EEADR:** holds the address of the EEPROM location (The address range of EEPROM memory starts from 00h to FFh (256 byte devices) and 80h to FFh (128 bytes devices))
- **EEADRH:** EEADR and EEADRH registers form a two-byte word that holds the 13-bit address of the program memory location being accessed

PIC microcontroller architecture

3. Serial communication:

The transfer of one bit of data at time consecutively over a communication channel is called Serial Communication. There are three protocols of serial communication:

3.1. USART: It stands for **Universal synchronous and Asynchronous Receiver and Transmitter** which provides a serial communication in two devices. In this protocol data is transmitted and received bit by bit through a single wire according to the clock pulses. To send and receive data serially the PIC microcontroller has two pins TXD and RXD.

PIC microcontroller architecture

3.2. SPI Protocol: SPI stands for **Serial Peripheral Interface. It is used to send data between PIC microcontrollers and other peripherals like sensors, shift registers and SD cards.** Three wire SPI communications is supported in PIC microcontroller between two devices on a common clock source. SPI protocol has greater data handling capability than that of the USART.

3.3. I2C Protocol: I2C stands for **Inter Integrated Circuit**, and **this protocol is used to connect low speed devices like microcontrollers, EEPROMS and A/D converters.** PIC microcontroller support two wire Interface or I2C communication between two devices which can work as both Master and Slave device.

PIC microcontroller architecture

4. Interrupts: There are **20 internal interrupts and three external interrupt sources** in PIC microcontrollers which are related with different peripherals like ADC, USART, Timers, and CCP etc.

5. I/O Ports: Let us take PIC16 series, it consists of five ports, such as **Port A, Port B, Port C, Port D and Port E**.

- **Port A: This port is 7-bit wide and can be used for both input and output**. The status of TRISA register decided whether it is used as input or output port.

- **Port B:** It is an 8-bit port. This port also can be used as input and output. Moreover in input mode four of its bits are variable according to the interrupt signals.

- **Port C:** It is also an 8-bit port and can be used as both input and output port which is determined by the status of the TRISC register.

PIC microcontroller architecture

- **Port D:** This 8-bit port, unlike Port A, B and C is not an input/output port, but is used as acts as a slave port for the connection to the microprocessor When in I/O mode Port D all pins should have Schmitt Trigger buffers.

- **Port E: It is a 3-bit port which is used as the additional feature of the control signals to the A/D converter**.

6. CCP Module: A CCP module works in the following three modes:

- **Capture Mode:** In this mode time is captured when a signal is arrived, or we can say that, when the CCP pin goes high it captures the value of the Timer1.
- **Compare Mode:** It works same as an analog comparator, which means that when timer 1's value reaches some reference value it will give an output signal.
- **PWM (pulse width modulation) Mode:** This mode provides a 10 bit resolution pulse and duty cycle that is programmable.

PIC microcontroller architecture

7. Timers:

Timers and counters are important as timers can tell the time and count. PIC microcontroller can have up to four timers (depending upon the family) Timer0, Timer1, Timer2 and Timer3. Timer0 and Timer2 are of 8-bits while the Timer1 and Timer3 are of 16-bits, which can also be used as a counter. These timers work according to the selected modes.

8. D/A Converter:

There are no analog outputs in PIC Microcontroller. To get analog output we have to use external Digital-to-Analog Converter (DAC). It can convert 8 bits of digital number from the eight digital outputs of PIC microcontroller.

PIC microcontroller architecture

9. A/D Converter:

It converts the analog voltage levels to digital voltage values. In PIC Microcontroller, ADC has 8-channels and has resolution of 10-bit, which means that if we have to convert an analog voltage between 0V to 5V the converter will divide it to 2^10 levels (1024 levels). The special function registers ADCON0 and ADCON1 control the operation of ADC. The converter stores the lower 8 bits in ADRESL register and the upper bits in the ADRESH register. Reference voltage of 5V is required for the operation of the converter.

10. Oscillator: Utilized to generate clock pulses in PIC microcontroller.

Applications Of PIC Microcontrollers:

- PIC microcontrollers are widely used in various applications, ranging from simple LED blinking projects to complex automation systems.
- They are commonly used in embedded systems, automotive electronics, industrial control systems, medical devices, and consumer electronics.
- Due to their low power consumption, small size, and cost-effectiveness, they are an ideal choice for many applications that require microcontrollers.

Difference between 8051 and PIC microcontroller

8051 Microcontroller	PIC microcontroller
8 bit for standard core bus width is present in 8051 microcontroller – average performance	8 bit/16 bit/32 bit bus width is present in PIC microcontroller – high performance
Average power consumption	Low power consumption
It is based on CISC Instruction set Architecture	It is based on RISC Instruction Set Architecture
12 clock/ instruction cycle	4 clock/instruction cycle
Von Neumann architecture	Harvard architecture
Flash, ROM, SRAM memory is used in 8051 microcontroller.	Flash, SRAM memory is used in PIC microcontroller.
Slower	Faster

Digital signal processor

A digital signal processor is a specialized microprocessor chip, with its architecture optimized for the operational needs of digital signal processing.

Digital Signal Processing (DSP) is a crucial component of many embedded systems, enhancing their ability to process and manipulate digital signals efficiently.

DSP is used in embedded systems to process signals in real-time or near real-time. For example, DSP can be employed in audio processing for noise reduction, equalization, and audio effects in products like headphones or smartphones. In automotive systems, DSP can be used for engine control and noise cancellation.

ASIC Processors

- The **application-specific integrated circuits** are built for specific applications.
- These chips are small in size and consume low power. The design cost of ASIC is high and this is the main disadvantage.
- The application-specific integrated circuit chips are used in satellites, modems, computers, etc. Some of the top ASICs manufacturer companies are Ams AG. Listed Company, Bitfury. Private Company, XMOS Semiconductor Private Company, Analogix Semiconductor Private Company, EDAptive Computing Private Company, Lumen Radio Private Company, Integrated Device Technology, Hookit. Private Company, etc.

Application Specific System Processor (ASSP)

- The application-specific system processor is a semiconductor integrated circuit product used to **implements a specific function**. **Different applications require unique set of system processor.**
- The performance, characteristics and die size of the application-specific system processor is the same as the ASIC.
- The ASSP's are used in various types of industries to perform video encoding or decoding and audio encoding or decoding.
- In place of embedded software, the application-specific system processor is used to run the application and it provides the solution faster. Example: IIM7100, W3100A.

Application Specific Instruction Processor (ASIP)

- The **application-specific instruction-set processors** are designed for specific applications. They have two parts: **minimum ISA (instruction set architecture) and configurable logic**. They are more flexible than ASSP.
- These processors have low power consumption, high computational speed, and good flexibility. They have better performance and low cost than FPGA.
- Due to programmability, the data path utilization is high in ASIPs, and the performance of this instruction set processor is good.

Programmable Logic Device (PLD)

- Programmable Logic Devices PLDs are the integrated circuits. They contain an array of AND gates & another array of OR gates. There are three kinds of PLDs based on the type of arrays, which has programmable feature
- **Classification:**

a. Programmable Read Only Memory

b. Programmable Array Logic

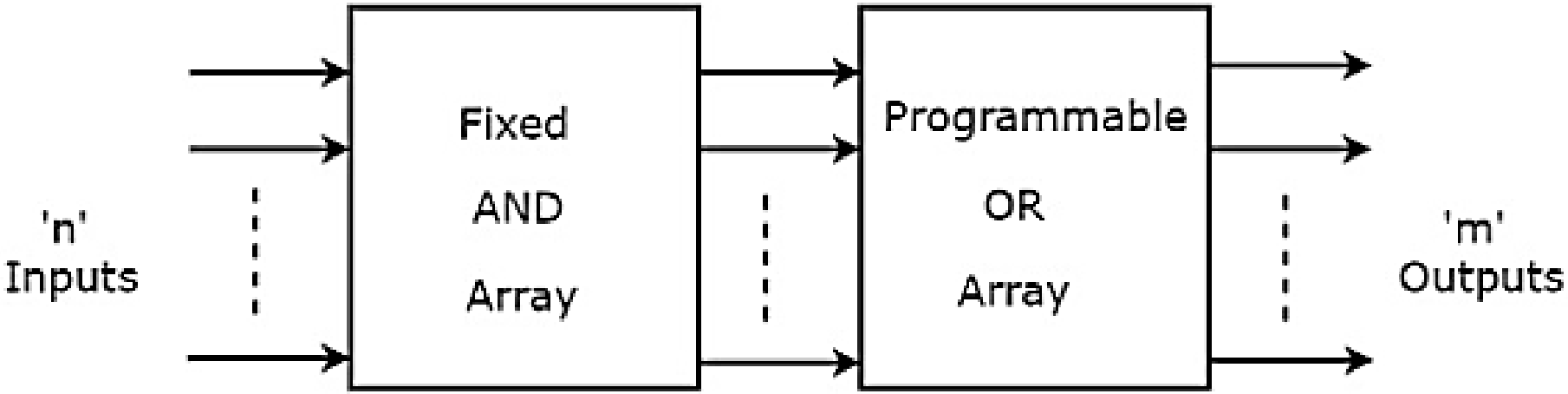
c. Programmable Logic Array

The process of entering the information into these devices is known as **programming**. Basically, users can program these devices or ICs electrically in order to implement the Boolean functions based on the requirement. Here, the term programming refers to hardware programming but not software programming.

Programmable Logic Device (PLD)

Programmable Read Only Memory: Here, the inputs of AND gates are not of programmable type. So, we have to generate 2^n product terms by using 2^n AND gates having n inputs each. We can implement these product terms by using n$\times$ 2^n decoder. So, this decoder generates ‘n’ min terms.

Here, the inputs of OR gates are programmable. That means, we can program any number of required product terms, since all the outputs of AND gates are applied as inputs to each OR gate. Therefore, the outputs of PROM will be in the form of sum of min terms.

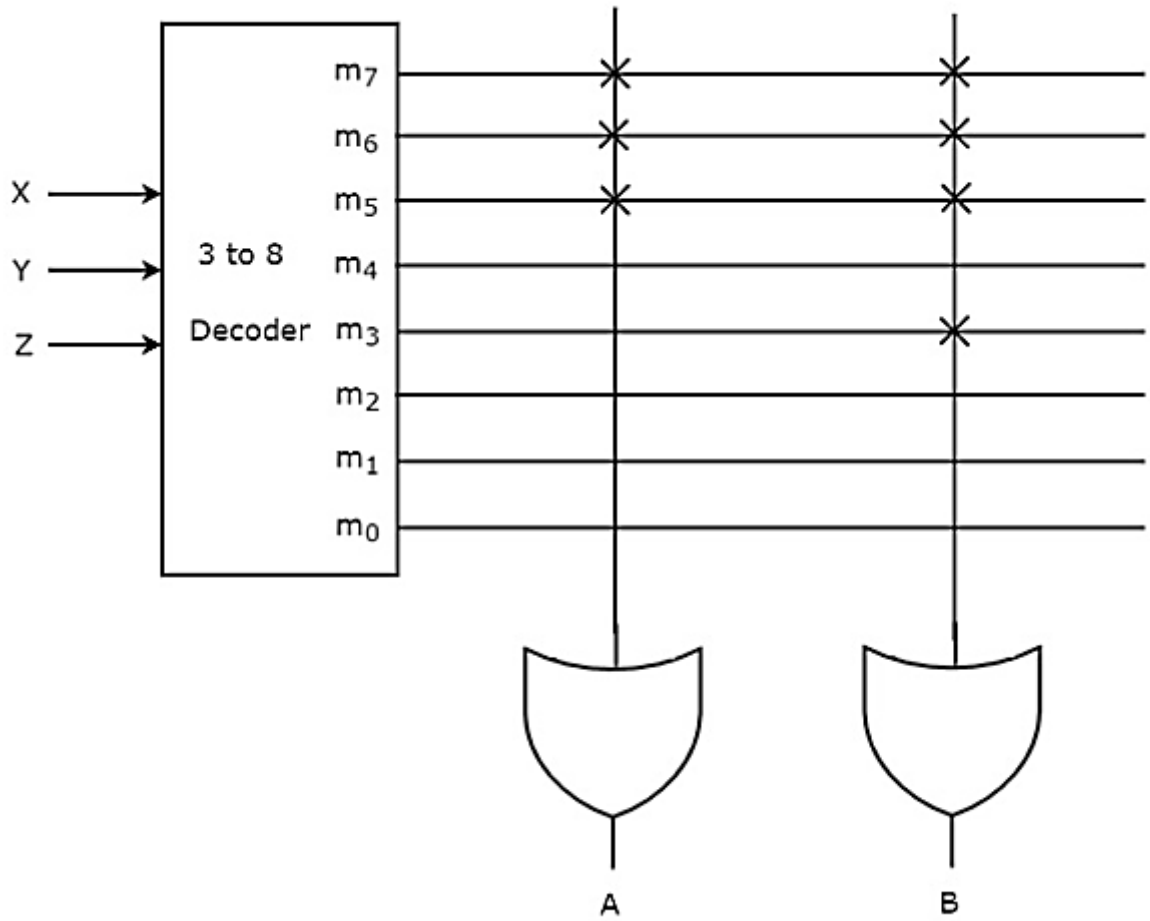


Programmable Logic Device (PLD)

Programmable Read Only Memory: Read Only Memory (ROM) is a memory device, which stores the binary information permanently. That means, we can't change that stored information by any means later.

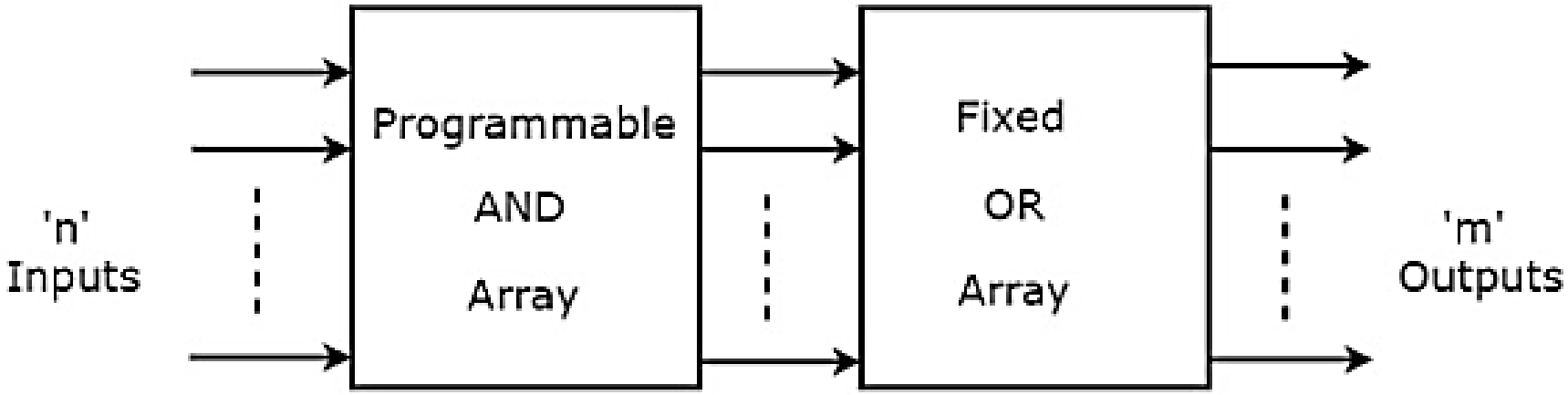
If the ROM has programmable feature, then it is called as Programmable ROM (PROM). The user has the flexibility to program the binary information electrically once by using PROM programmer.

PROM is a programmable logic device that has **fixed AND array & Programmable OR array**. The block diagram of PROM is shown in the following figure.


X
Y
Z
3 to 8
Decoder
m_7
m_6
m_5
m_4
m_3
m_2
m_1
m_0
A
B

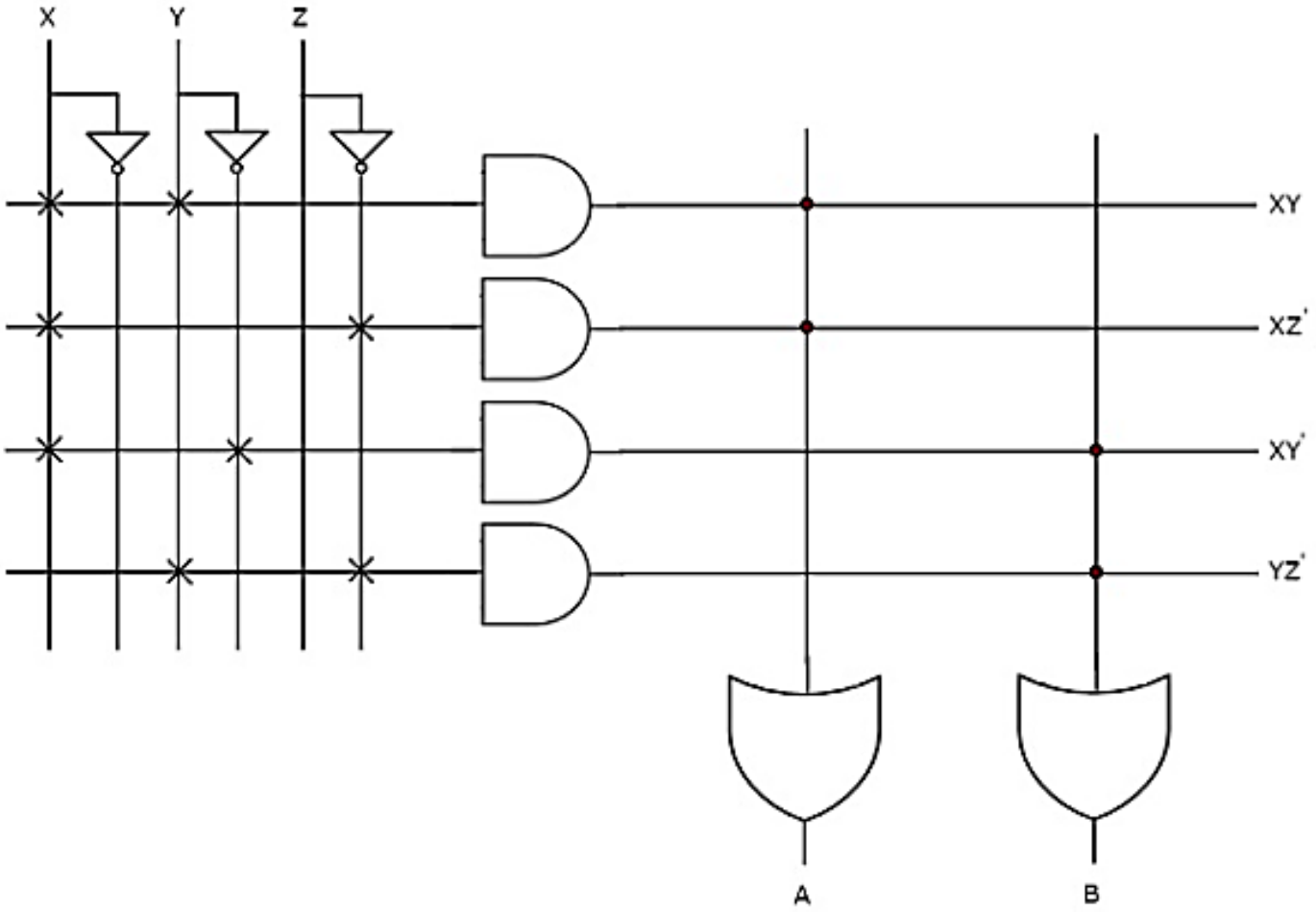
Programmable Logic Device (PLD)

Programmable Array Logic: PAL is a programmable logic device that has **Programmable AND array & fixed OR array**. The advantage of PAL is that we can generate only the required product terms of Boolean function instead of generating all the min terms by using programmable AND gates. The block diagram of PAL is shown in the following figure.



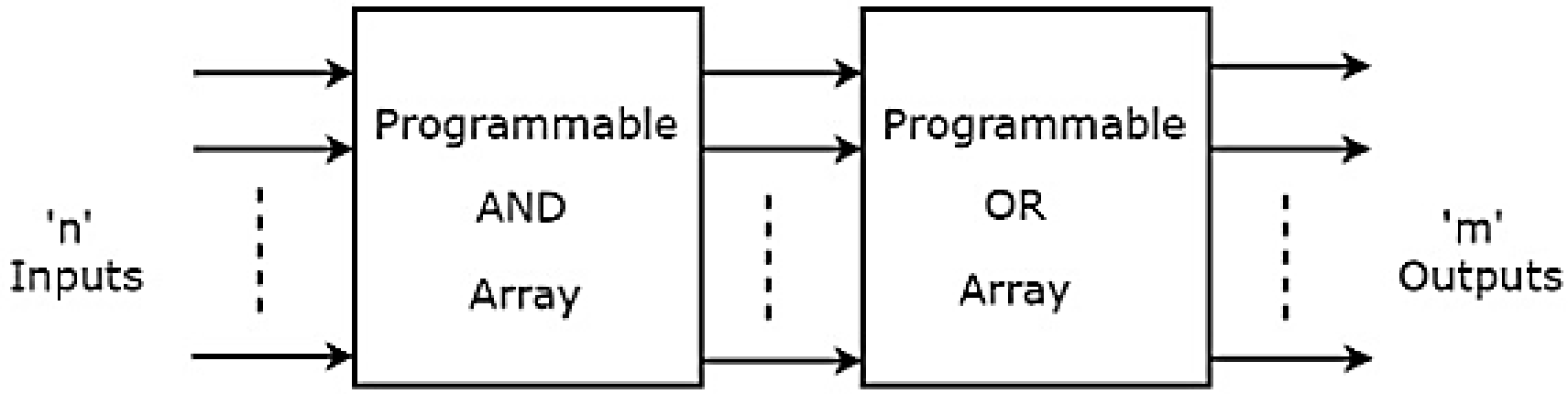
Programmable Logic Device (PLD)

Programmable Array Logic:



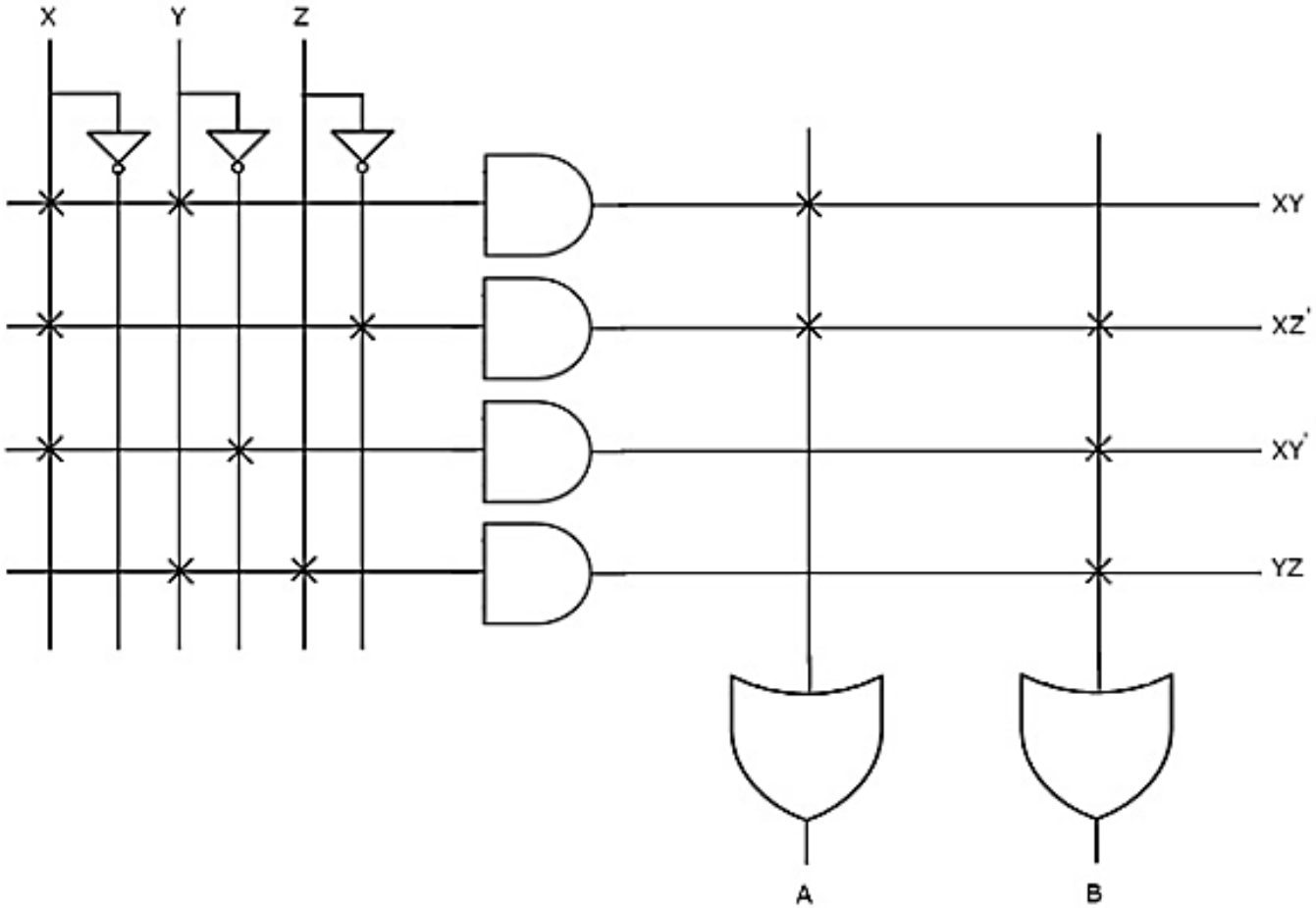
Programmable Logic Device (PLD)

Programmable Logic Array: PLA is a programmable logic device that has both Programmable AND array & Programmable OR array. Hence, it is the most flexible PLD. The block diagram of PLA is shown in the following figure.



Programmable Logic Device (PLD)

Programmable Logic Array:

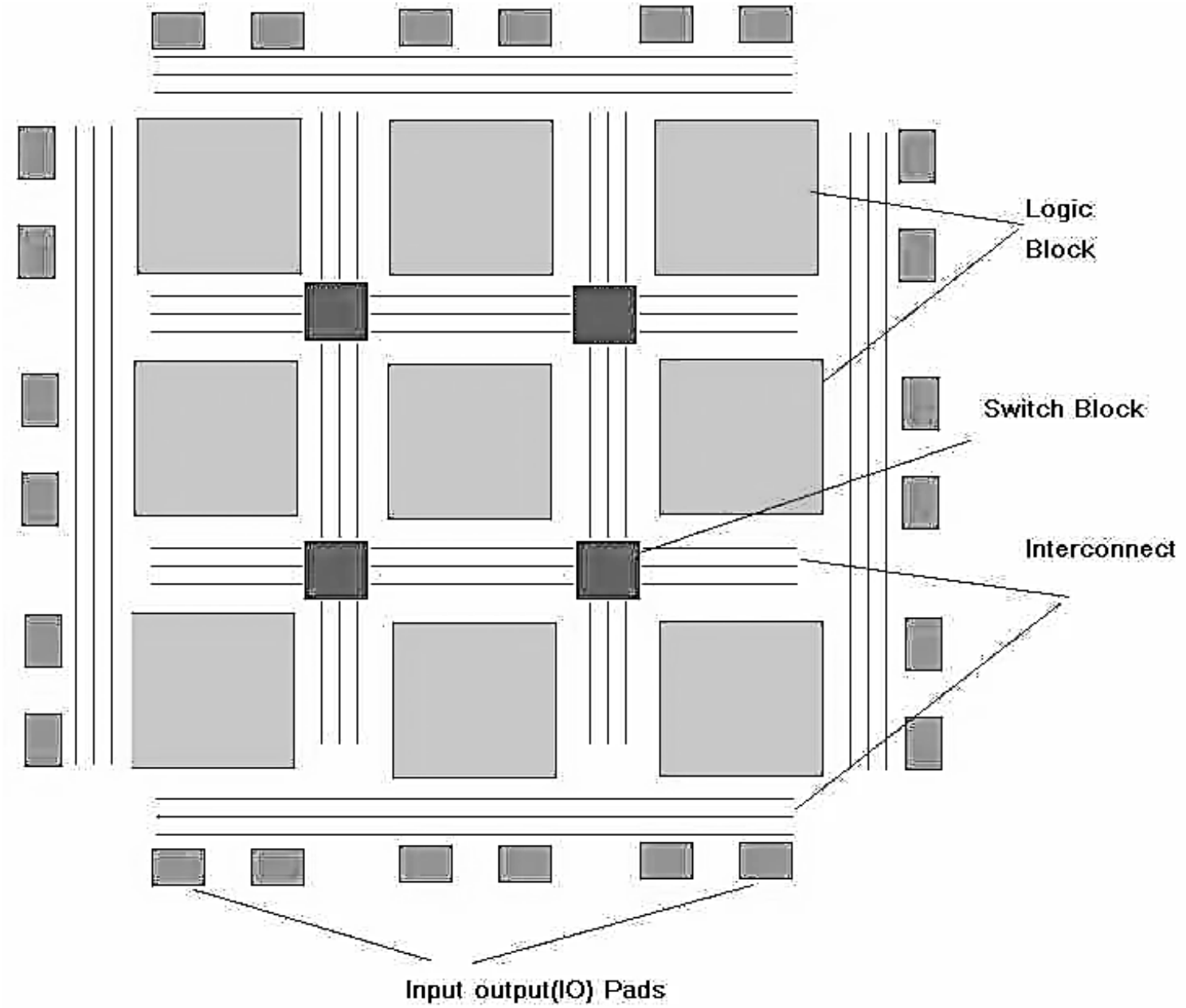


Field Programmable Gate Array

FPGA stands for Field Programmable Gate Array which is an IC that can be programmed to perform a customized operation for a specific application. They have thousands of gates. In the field of VLSI FPGAs have been very popular. Languages such as VHDL and Verilog are used to write the code for FPGA programming.

FPGAs have become a game-changing technology that are transforming embedded systems. **With their unparalleled versatility, these robust semiconductor devices allow engineers to design application-specific hardware combinations that can be reprogrammed and improved as necessary.**

Field Programmable Gate Array



Commercial Off-The-Shelf (COTS) components

COTS, or commercial-off-the-shelf, is a product that remains "as is." This means that the hardware is a standard product that already exists and is available from commercial sources.

COTS products are designed to be easily installed and interoperate with existing system components.

Some of the benefits of using COTS products are

- **lower costs**
- **reduced development time**
- **faster insertion of new technology**
- **lower lifecycle costs resulting from using readily available**
- **up-to-date products**

In a spacecraft: payload, electrical and power subsystem, command and data handling system, communication subsystem, thermal control subsystem, attitude determination and control subsystem, propulsion control subsystem, mechanisms, and actuators.

Memory hierarchy

Memory Hierarchy is one of the most required things in Computer Memory as it helps in optimizing the memory available in the computer. There are multiple levels present in the memory, each one having a different size, different cost, etc.

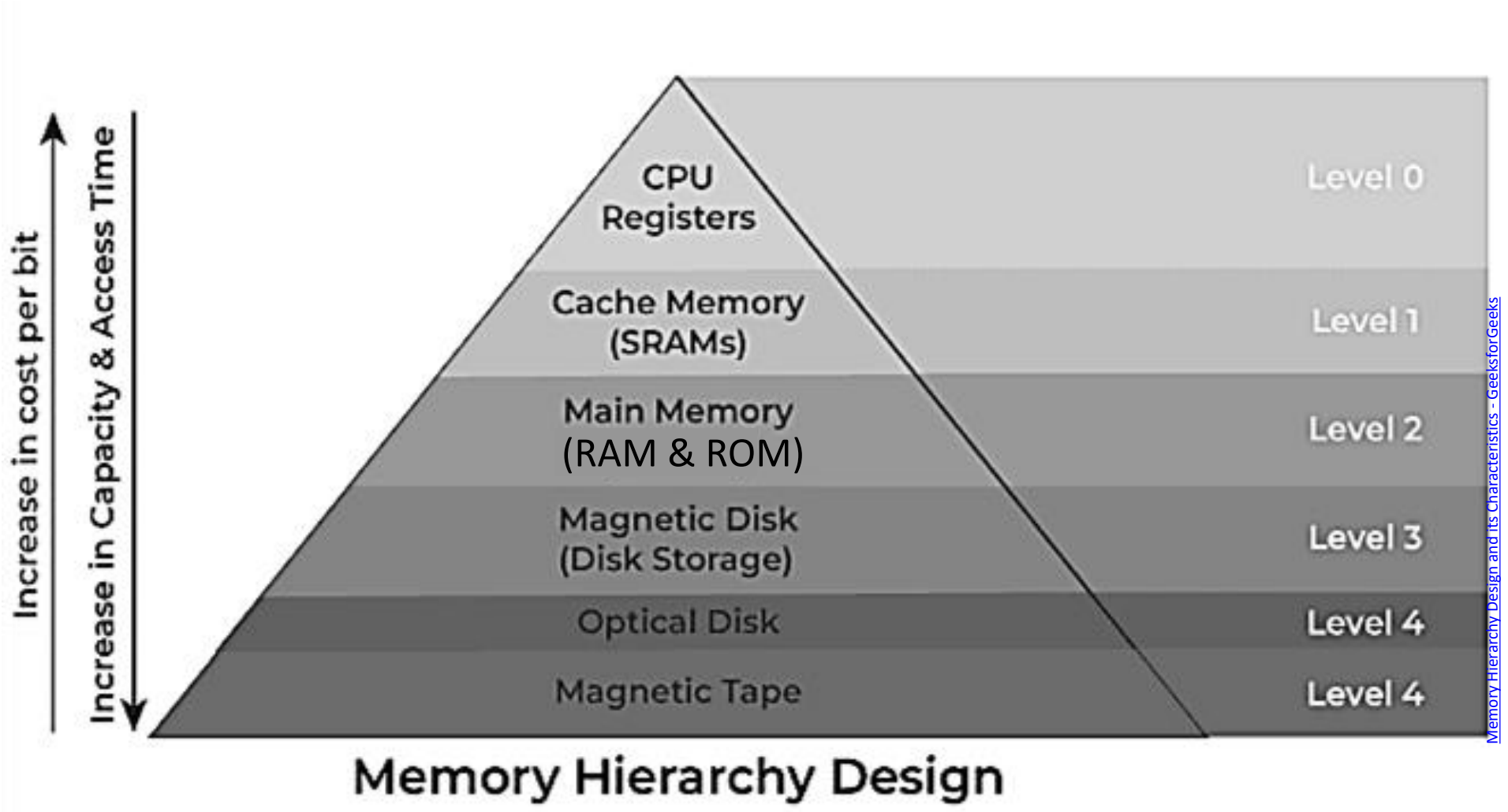
Types of Memory Hierarchy

This Memory Hierarchy Design is divided into 2 main types:

1. **External Memory or Secondary Memory:** Comprising of Magnetic Disk, Optical Disk, and Magnetic Tape i.e. peripheral storage devices which are accessible by the processor via an I/O Module.

2. **Internal Memory or Primary Memory:** Comprising of Main Memory, Cache Memory & CPU registers. This is directly accessible by the processor.

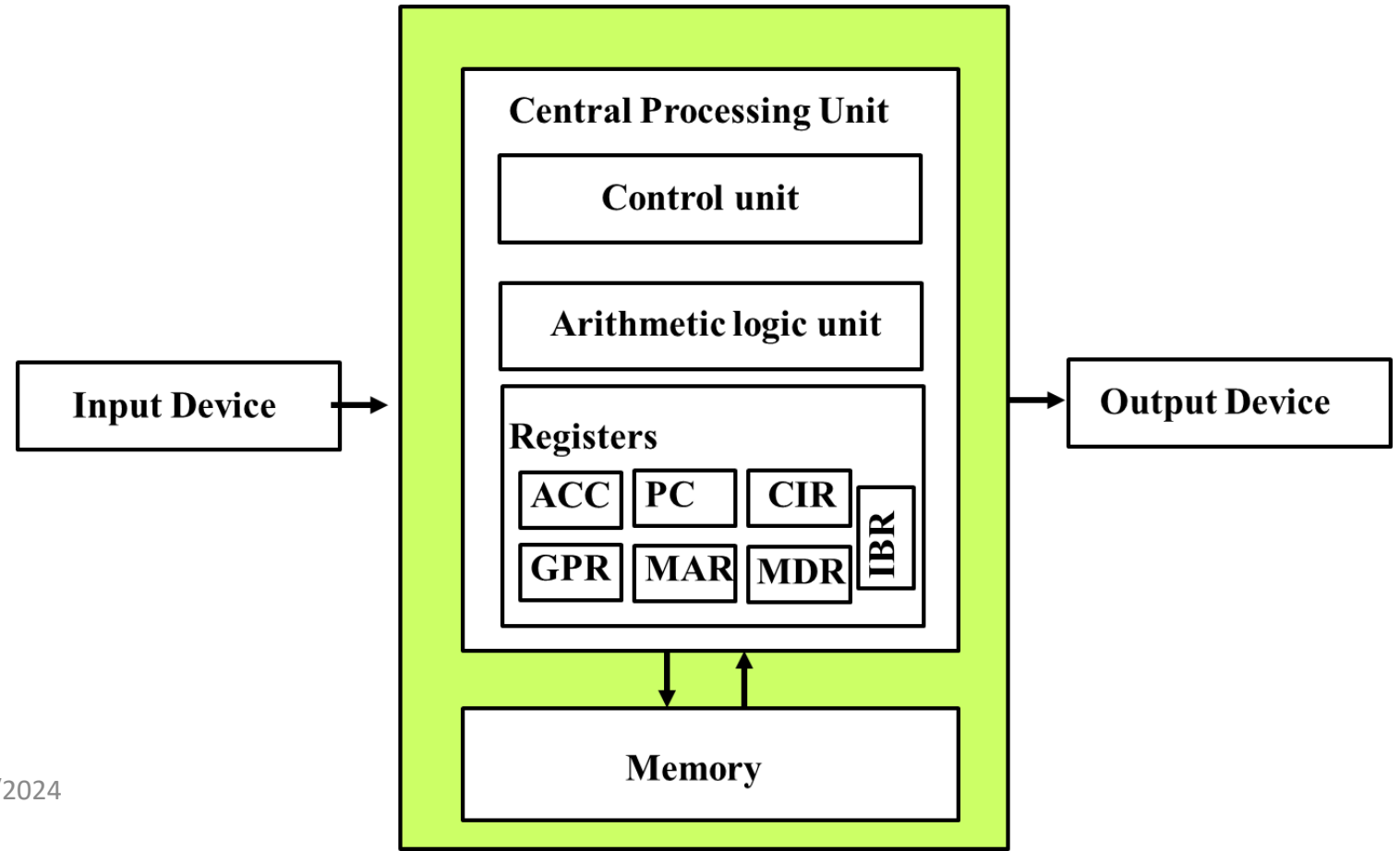
Memory hierarchy



Memory hierarchy

1. Registers

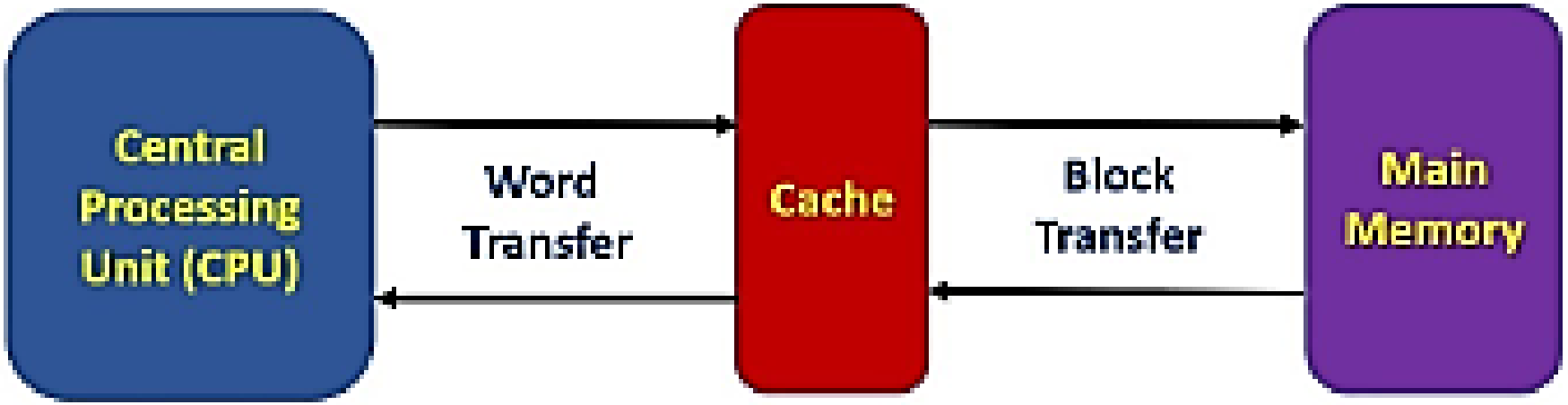
Registers are small, high-speed memory units located in the CPU. They are used to store the most frequently used data and instructions. Registers have the fastest access time and the smallest storage capacity, typically ranging from 16 to 64 bits.



Memory hierarchy

2. Cache Memory

Cache memory is a small, fast memory unit located close to the CPU. It stores frequently used data and instructions that have been recently accessed from the main memory. **Cache memory is designed to minimize the time it takes to access data by providing the CPU with quick access to frequently used data.**



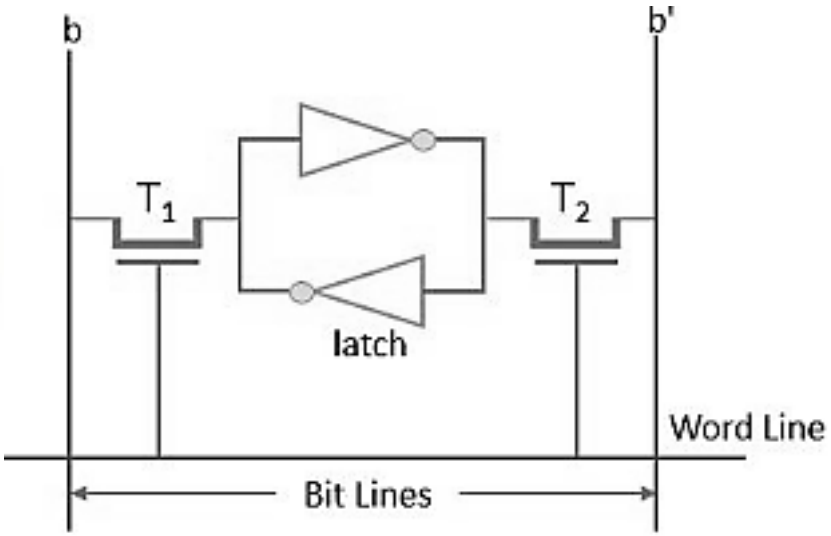
Memory hierarchy

3. Main Memory

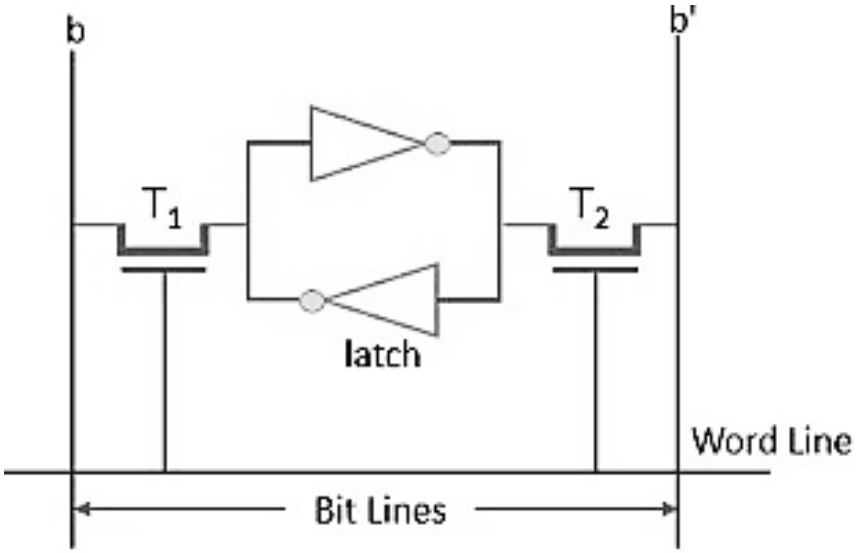
Main memory, also known as RAM (Random Access Memory), is the primary memory of a computer system. It has a larger storage capacity than cache memory, but it is slower. Main memory is used to store data and instructions that are currently in use by the CPU.

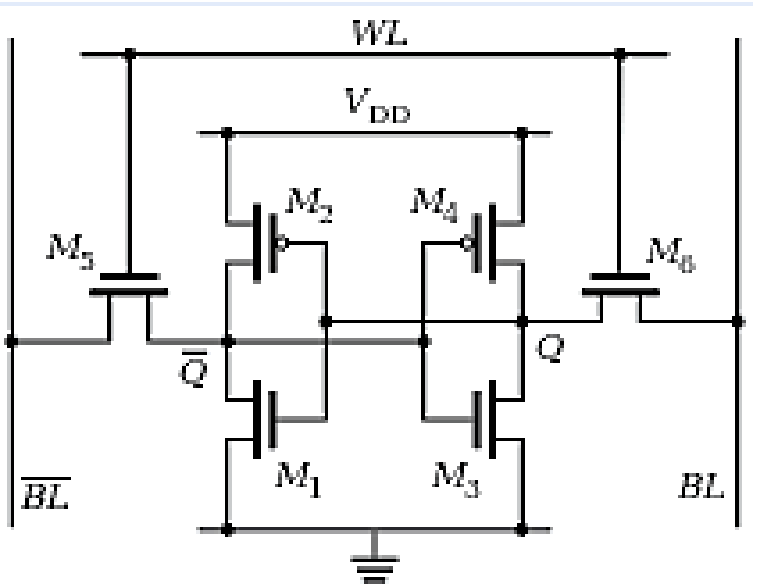
Types of Main Memory

Static RAM: Static RAM stores the binary information in flip flops and information remains valid until power is supplied. It has a faster access time and is used in implementing cache memory.



A typical SRAM cell is made up of six MOSFETs, and is often called a **6T SRAM cell**. Each bit in the cell is stored on four transistors (M1, M2, M3, M4) that form two cross-coupled inverters. This storage cell has two stable states which are used to denote 0 and 1. Two additional access transistors serve to control the access to a storage cell during read and write operations. 6T SRAM is the most common kind of SRAM.

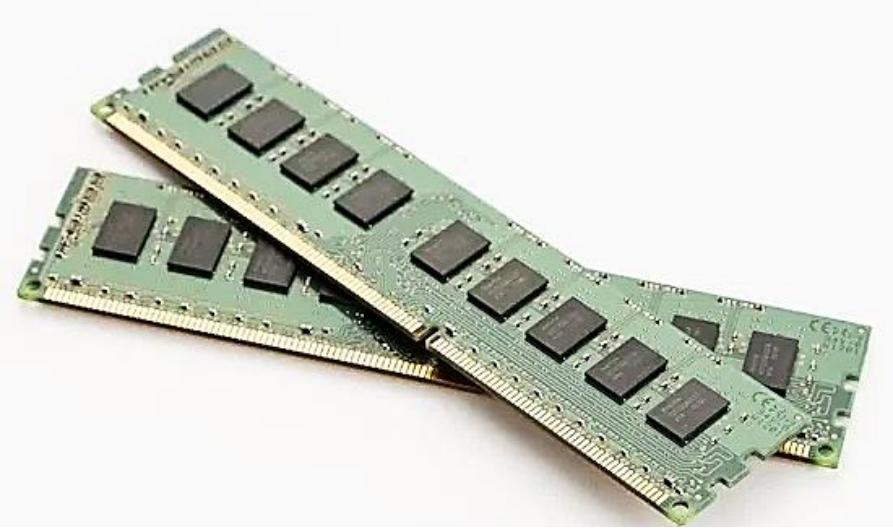


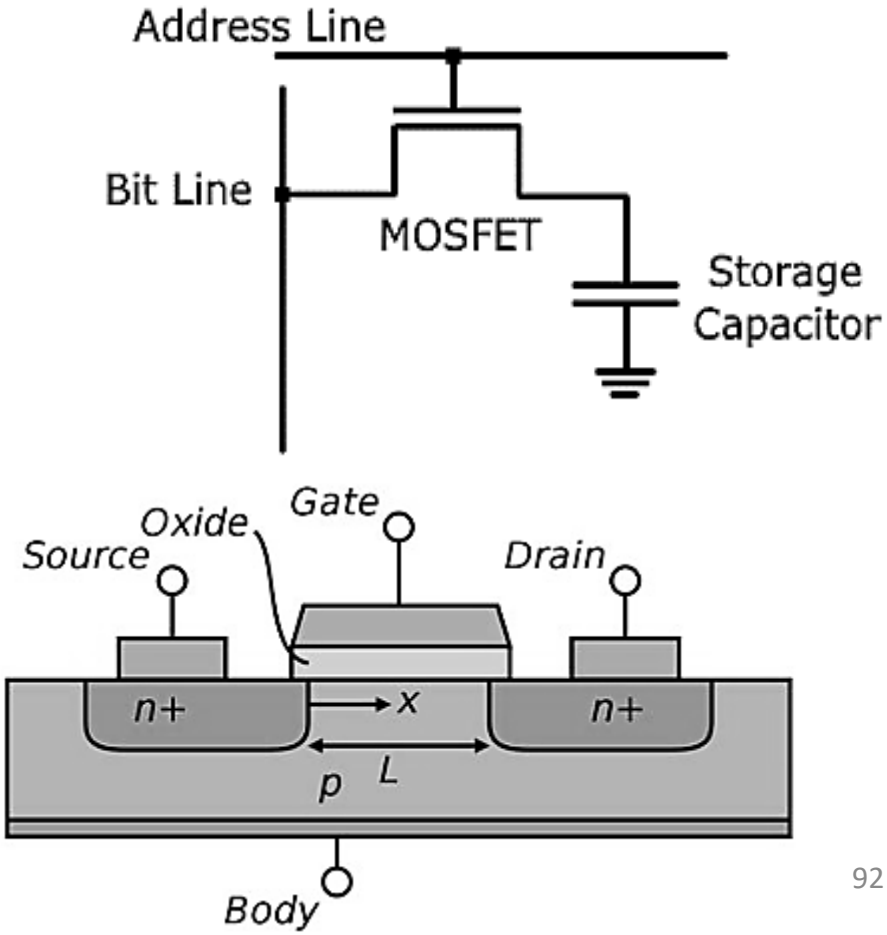


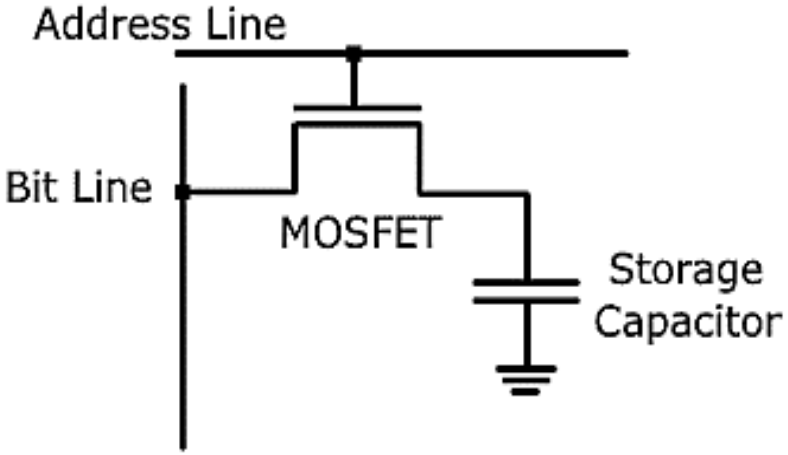
Access to the cell is enabled by the word line (WL in figure) which controls the two *access* transistors M_5 and M_6 which, in turn, control whether the cell should be connected to the bit lines: BL and BL.

Memory hierarchy

Dynamic RAM: It stores the binary information as a charge on the capacitor. It requires refreshing circuitry to maintain the charge on the capacitors after a few milliseconds. It contains more memory cells per unit area as compared to SRAM.







C in storage capacitor determined by:

better dielectric more area

$$C = \frac{\varepsilon A}{d}$$

thinner film

The capacitor stores charge. If the charge is above a threshold value then it will store logic 1. There is no circuitry to hold the static charge of capacitor. Stored charge will leak. The refreshing is necessary after a few milliseconds.

Memory hierarchy

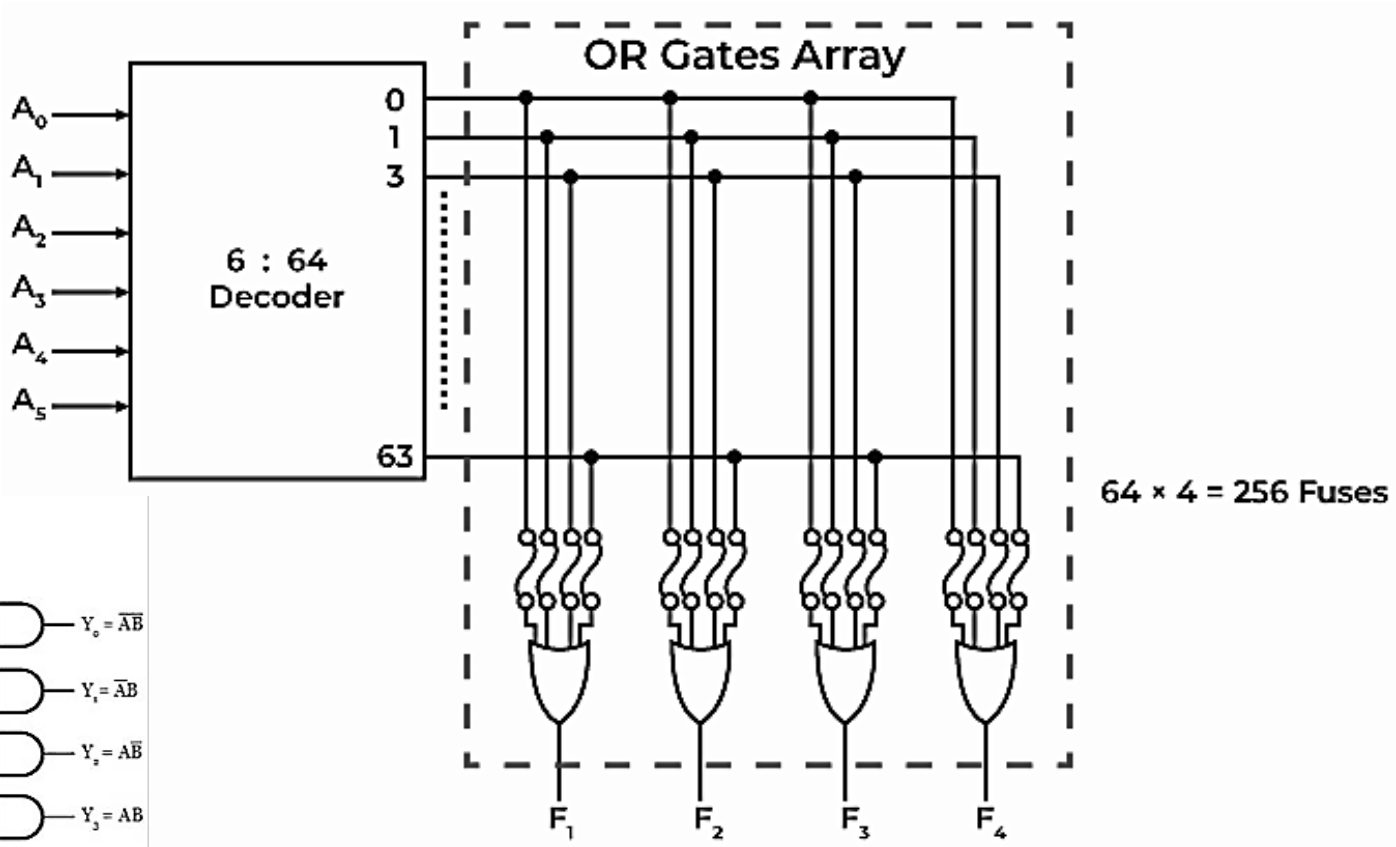
ROM: ROM stands for Read-Only Memory. It is a non-volatile memory that is used to store important information which is used to operate the system. As its name refers to read-only memory, we can only read the programs and data stored on it.

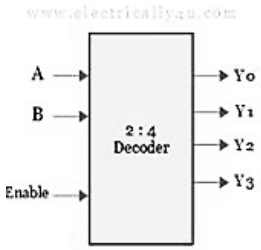
Internal Structure of ROM:

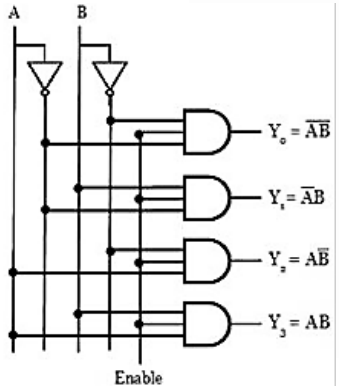
The internal structure of ROM have two basic components.

Decoder

OR gates







Memory hierarchy

Types of ROM:

Mask ROM:

Mask ROM refers to a type of ROM in which the contents are programmed by the IC manufacturer. In other words, it is not a user-programmable ROM. Since the process is costly, mask ROM is used when the needed volume is high (hundreds of thousands) and it is absolutely certain that the contents will not change.

PROM (programmable ROM):

PROM refers to the kind of ROM that the user can burn information into. In other words, PROM is a **user-programmable memory**. For every bit of the PROM, there exists a fuse. PROM is programmed by blowing the fuses. If the information burned into PROM is wrong, that PROM must be discarded since its internal fuses are blown permanently. For this reason, PROM is also referred to as OTP (**One Time Programmable**). Programming ROM also called burning ROM, requires special equipment called a ROM burner or ROM programmer.

Memory hierarchy

Types of ROM:

EPROM (erasable programmable ROM) and UV-EPROM:

EPROM was invented to **allow making changes** in the contents of PROM after it is burned. In EPROM, one can program the memory chip and erase it thousands of times. This is especially necessary during the development of the prototype of a **microprocessor-based project**. A widely used EPROM is called **UV-EPROM**, where UV stands for ultraviolet. The only problem with UV-EPROM is that erasing its contents can take up to 20 minutes.

All UV-EPROM chips have a window through which the programmer can shine ultraviolet (UV) radiation to erase the chip's contents. For this reason, EPROM is also referred to as **UV-erasable EPROM** or simply UV-EPROM.

Memory hierarchy

Types of ROM:

EEPROM (electrically erasable programmable ROM):
EEPROM has several advantages over EPROM, such as the fact that its method of erasure is electrical and therefore instant as opposed to the 20-minute erasure time required for UV-EPROM. In addition, in EEPROM one **can select which byte to be erased**, in contrast to UV-EPROM, in which the entire contents of ROM are erased. To utilize EEPROM fully, the designer must incorporate the circuitry to program the EEPROM into the system board. In general, the cost per bit for EEPROM is much higher than for UV-EPROM.

Memory hierarchy

Types of ROM:

Flash Memory EPROM:

Since the early 1990s, Flash EPROM has become a popular user-programmable memory chip. and for good reasons.

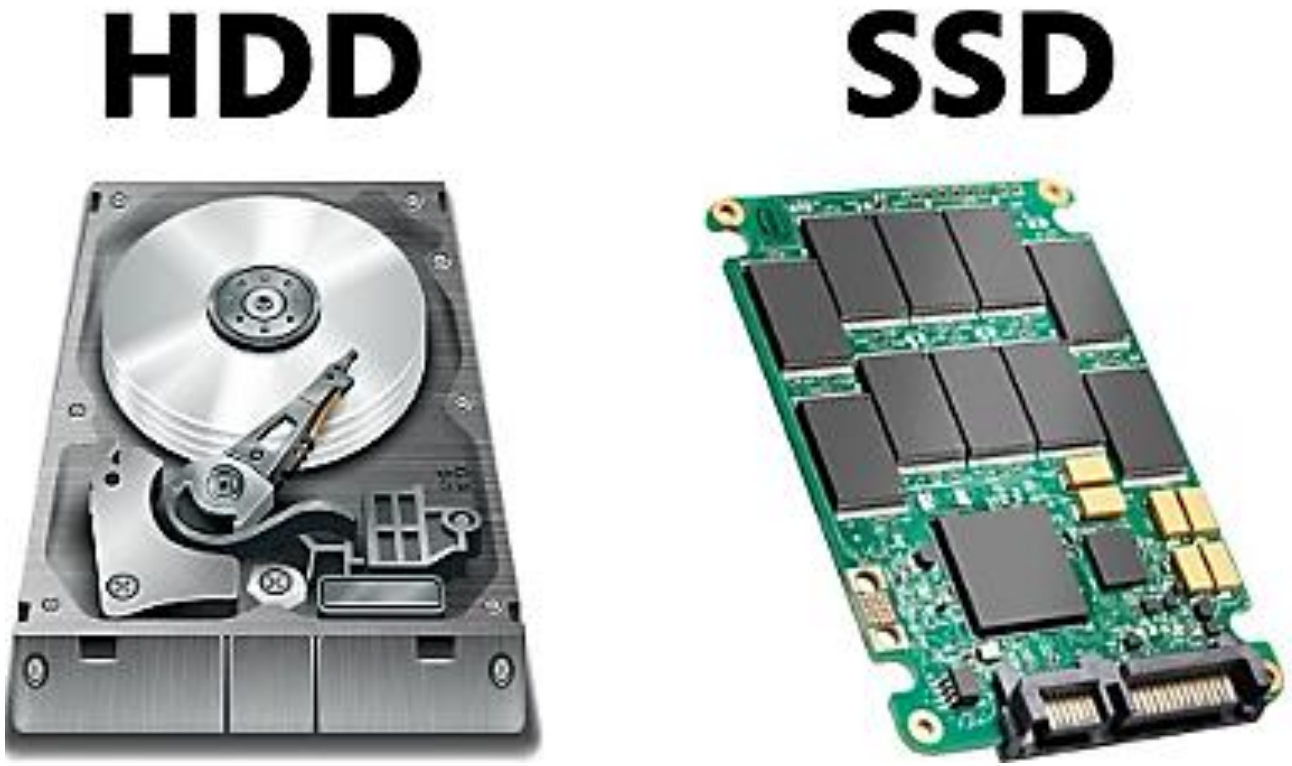
First, the erasure of the entire contents takes less than a second, or one might say in a flash, hence its name, Flash memory.

In addition, the erasure method is electrical, and for this reason, it is sometimes referred to as Flash EEPROM. To avoid confusion, it is commonly called Flash memory.

Memory hierarchy

4. Secondary Storage

Secondary storage, such as hard disk drives (HDD) and solid-state drives (SSD), is a non-volatile memory unit that has a larger storage capacity than main memory. It is used to store data and instructions that are not currently in use by the CPU. Secondary storage has the slowest access time and is typically the least expensive type of memory in the memory hierarchy.



Memory hierarchy

Magnetic Disk

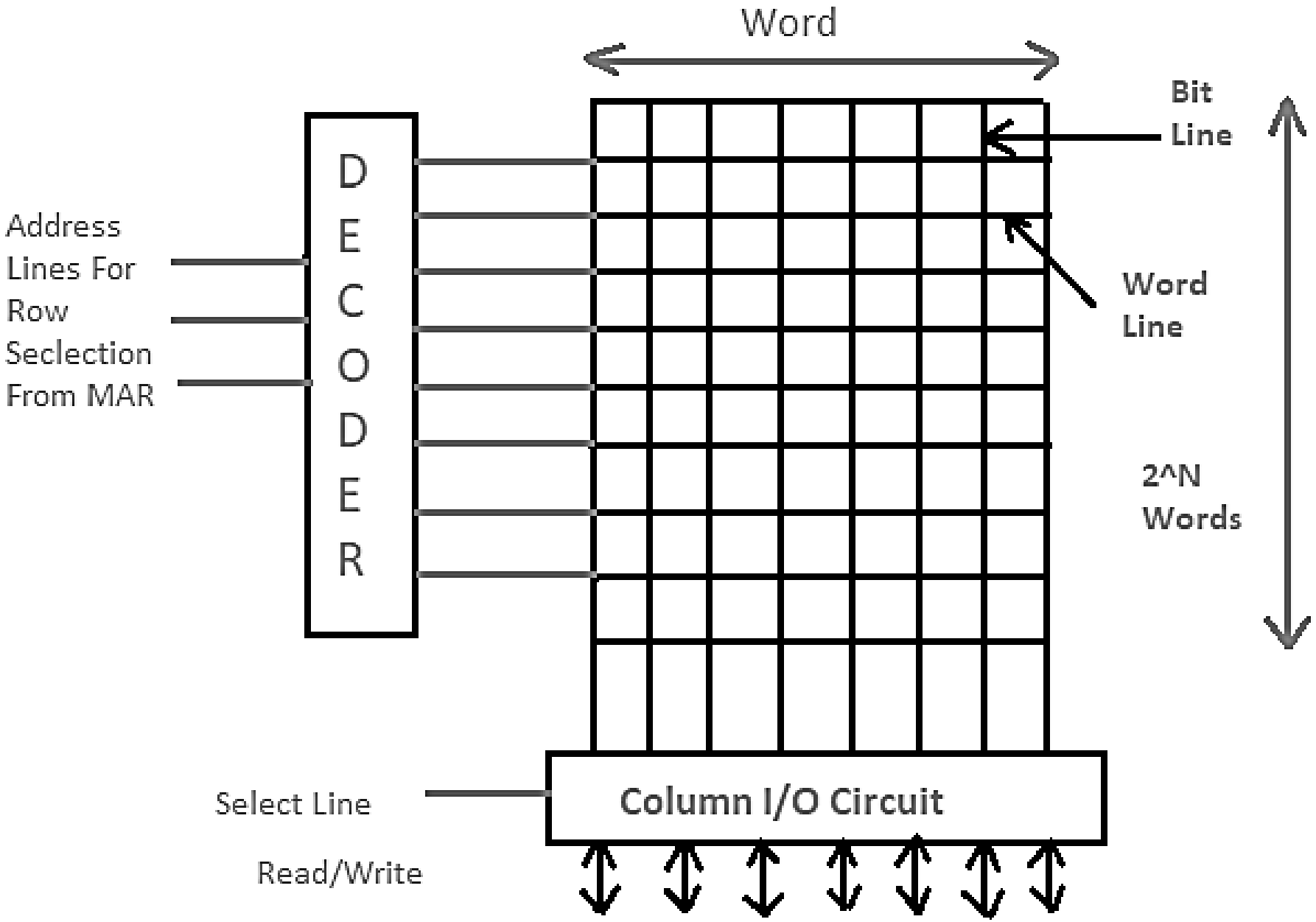
Magnetic Disks are simply circular plates that are fabricated with either a metal or a plastic or a magnetized material. The Magnetic disks work at a high speed inside the computer and these are frequently used.

Memory hierarchy

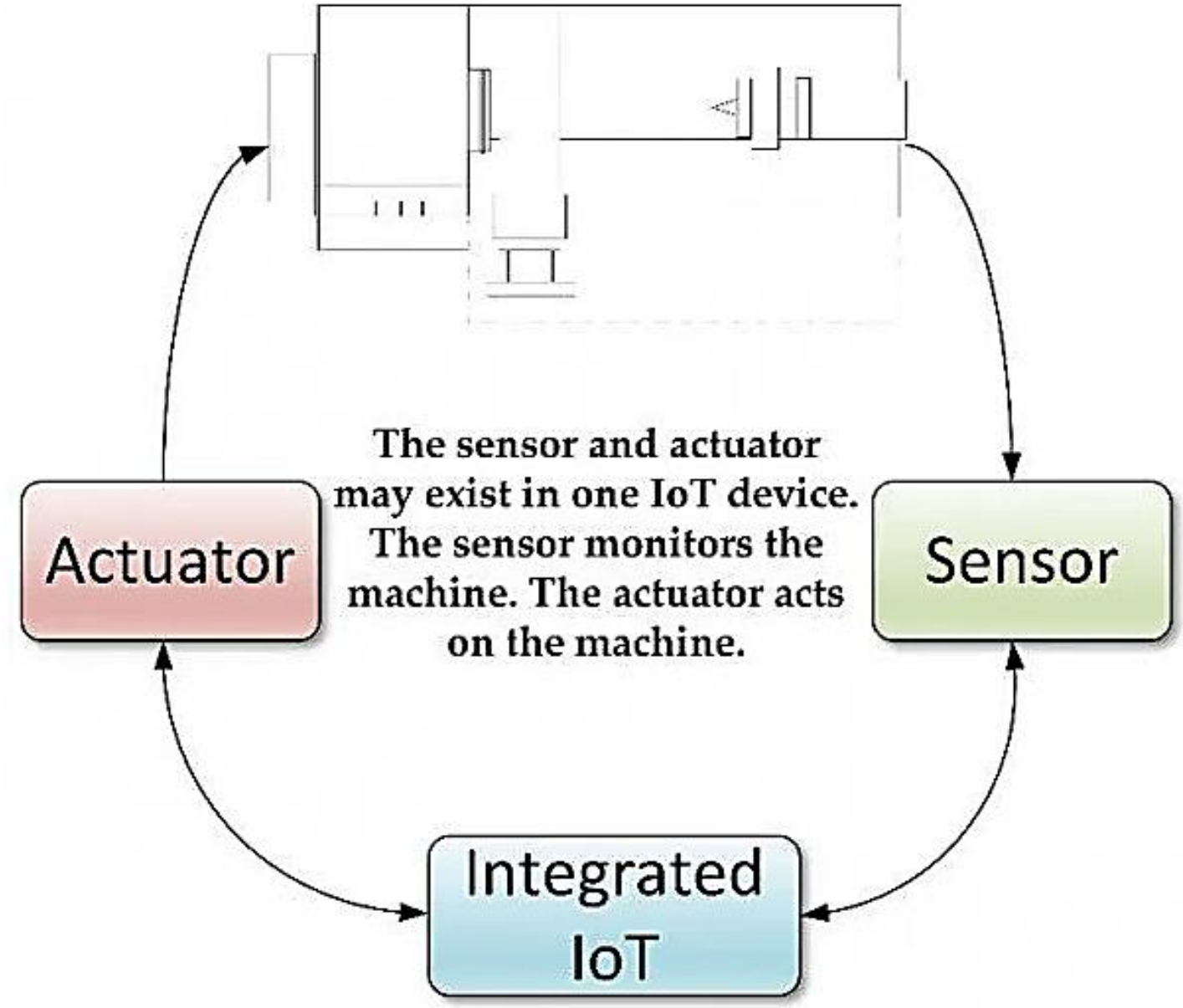
Magnetic Tape

Magnetic Tape is simply a magnetic recording device that is covered with a plastic film. It is generally used for the backup of data. In the case of a magnetic tape, the access time for a computer is a little slower and therefore, it requires some amount of time for accessing the strip.

Memory hierarchy



Role of sensor and actuator in embedded system



Role of sensor in embedded system: Sensors gather data from the device's environment. They can measure things like temperature, pressure, motion, light, and more. The data collected by sensors is used by the embedded system to make decisions or trigger specific actions.

Role of actuator in embedded system design: Actuators are components that convert electrical signals from the embedded system into physical actions. They enable the system to manipulate or control various devices and mechanisms.

Role of Real Time Clock in embedded system

A **Real-Time Clock (RTC)** is a crucial component in many embedded systems, providing accurate timekeeping and datekeeping functions. Here are the primary purposes and functionalities of an RTC in an embedded system:

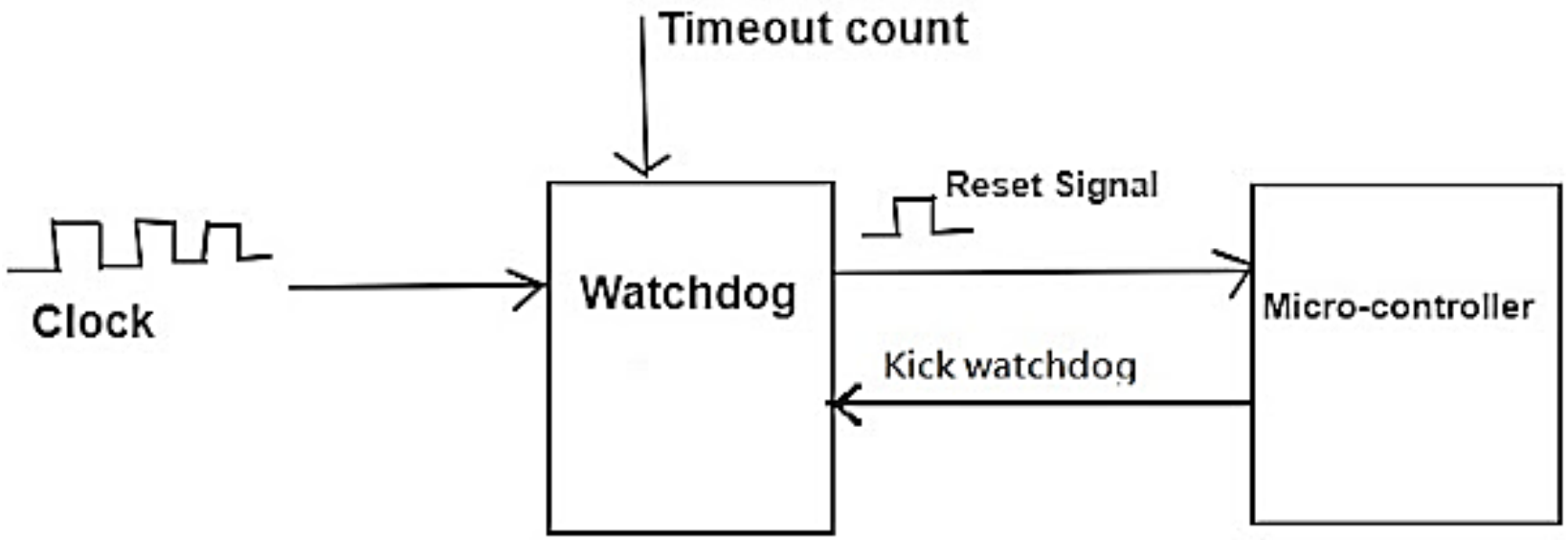
1. **Accurate Timekeeping:** The main function of an RTC is to keep track of the current time and date even when the system is powered off. This is critical for applications that require precise timing information.
2. **Maintaining a Calendar:** An RTC provides the ability to track the current date, including the day, month, and year. This information is essential for applications that involve date-based calculations.
3. **Battery Backup for Timekeeping:** An RTC typically has a backup power source (often a small coin cell battery) that allows it to continue operating and keeping track of time even when the main system power is turned off.
4. **Synchronization and Timestamping:** An RTC provides a reliable time reference for coordinating activities, events, or data logging across multiple devices or systems.

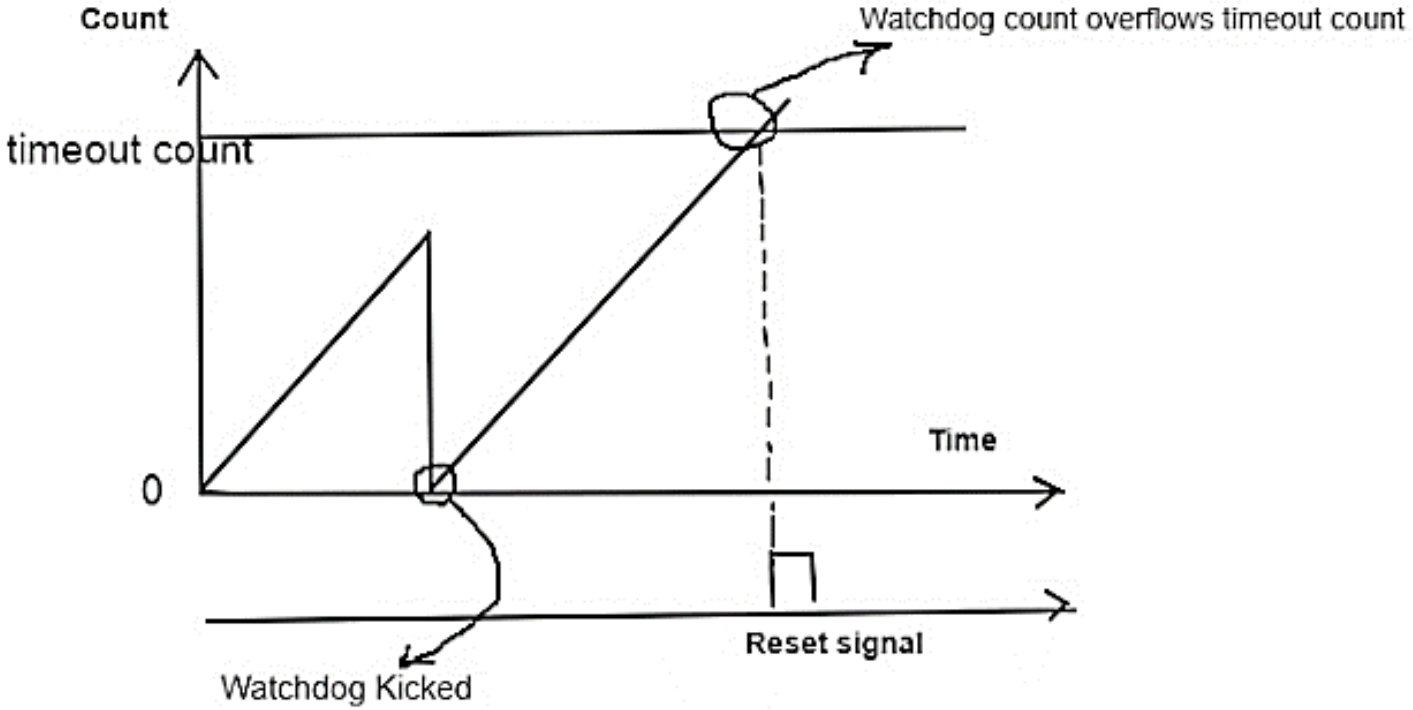
Role of Watch dog Timer in embedded system

Watchdog timer is a piece of hardware in micro-controller. **Watchdog timer is used to generates system reset if system gets stuck somewhere** i.e. if system goes into endless loop of execution watchdog timer will reset the system to come out of endless loop. Watchdog is safety mechanism in embedded system which makes your system reliable, but **it depends on how you make use of watchdog timer.**

Watchdog is basically a counter, which starts from counting zero and reaches to a certain value. **If counter reaches to certain value then watchdog hardware will generates a watchdog reset. To avoid system reset, software needs to kick the watchdog i.e. need to reset the counter to zero.** In case software stuck into endless loop it system will not able to kick the watchdog hence counter reaches to certain value and resets the system.

Role of Watch dog Timer in embedded system





Role of I/O circuit in embedded system

In an **embedded system**, **I/O (Input/Output) ports** play a crucial role in enabling communication between the microcontroller or microprocessor and the external world.
Significance of I/O circuits:

1. **Interfacing with External Devices**:
   - I/O ports allow the microcontroller to **interact with sensors, actuators, and other external devices**.
2. **Data Transfer**:
   - I/O ports facilitate the **transfer of data** between the microcontroller and the external environment.
3. **Real-Time Responsiveness**:
   - Many embedded systems require **real-time responses** to external events.
4. **Energy Efficiency**:
   - In battery-powered or energy-efficient systems, minimizing power consumption is crucial.
5. **Customizability and Flexibility**:
   - Microcontrollers offer a variety of **built-in peripherals** (such as analog-to-digital converters, digital I/O pins, UART, SPI, and I2C).

Communication Protocols in Embedded Systems

Communication Protocols are a set of rules that allow two or more communication systems to communicate data via any physical medium. The rules, regulations, synchronization between communication systems, syntax to be followed and semantics are all defined by the term protocol. Protocols can be implemented by both hardware and software or combination of both.

In embedded System a processor or controller takes input from the physical world peripherals like sensors, processes the same through appropriate software and provides the desired output. In this case, the components have to communicate with each other to provide the anticipated output. Each communicating entity should agree to some protocol to exchange information.

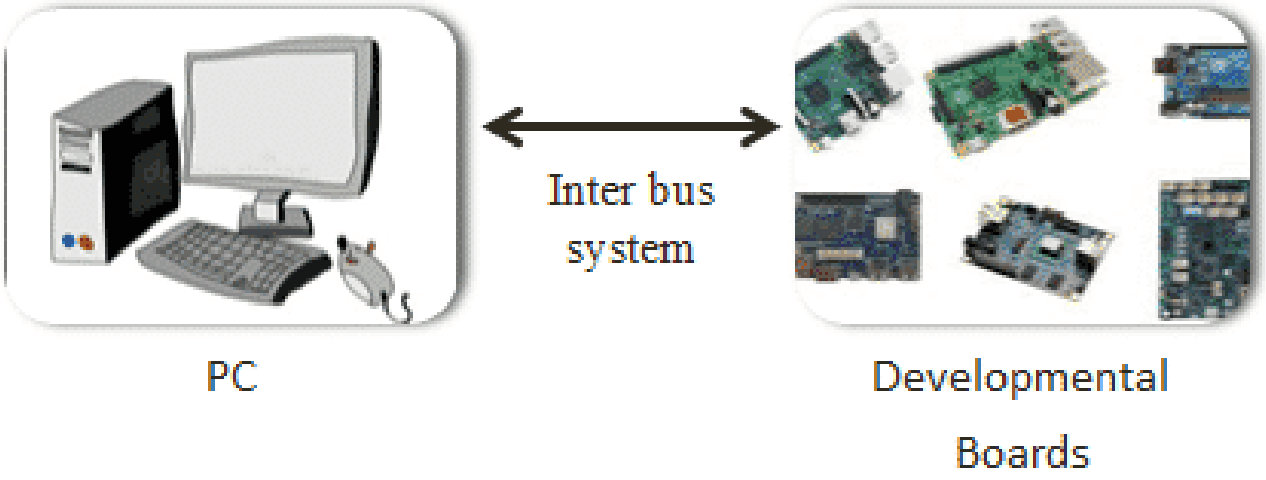
A **Protocol** is a set of rules agreed by both the sender and receiver on

- How the data is packed
- How many bits constitute a character
- When the data begins and ends

Inter system and intra system communication protocols

Inter System Communication Protocols:

Inter system protocols establish **communication between two communicating devices (External) i.e. between PC and microprocessor kit, developmental boards, etc**. In this case, the communication is achieved through inter bus system. E.g: USB, UART/USART.



Inter system and intra system communication protocols

Intra System Communication Protocols:

The Intra system protocol establishes **communication (Internal) between components within the circuit board**. In embedded systems, intra system protocol increases the number of components connected to the controller. Increase in components lead to circuit complexity and increase in power consumption. Intra system protocol promises secure access of data from the peripherals. Example, I2C Protocol, SPI Protocol, CAN Protocol.

Serial and parallel communication

Parallel Communication:

- In parallel communication, all the bits of data are transmitted simultaneously on separate communication lines.
- Used for shorter distance.
- In order to transmit n bit, n wires or lines are used.
- More costly.
- Faster than serial transmission.
- Data can be transmitted in less time.

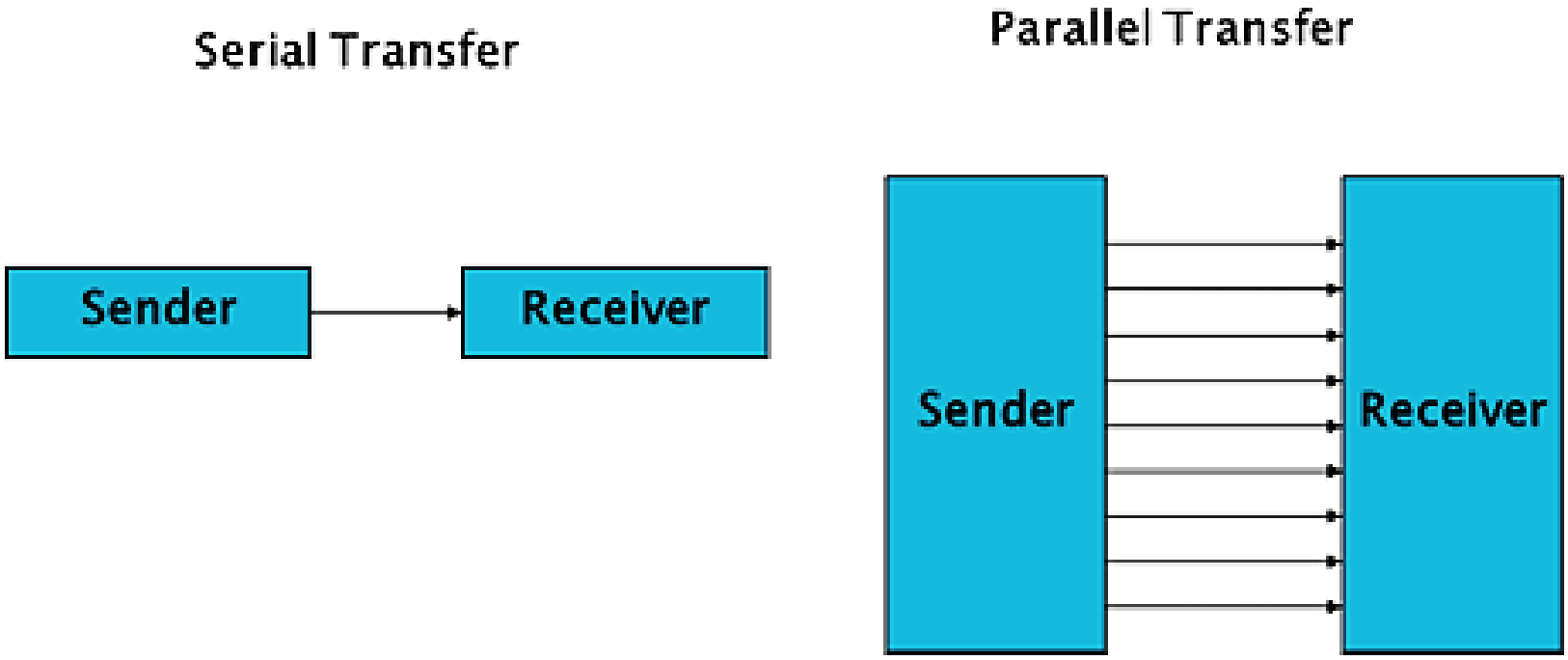
Example: printers and hard disk

Serial Communication Basics:

- In serial communication the data bits are transmitted serially one by one i.e. bit by bit on single communication line
- It requires only one communication line rather than n lines to transmit data from sender to receiver.
- Thus all the bits of data are transmitted on single lines in serial fashion.
- Less costly.
- Long distance transmission.

Example: Telephone.

Serial and parallel communication



Asynchronous and synchronous communication

Serial communication uses two methods:

- Asynchronous.
- Synchronous.

Asynchronous:

⇨ Transfers single byte at a time.

⇨ No need of clock signal

❖ Example: UART (universal asynchronous receiver transmitter)

Synchronous:

⇨ Transfers a block of data (characters) at a time.

⇨ Requires clock signal

❖ Example: SPI (serial peripheral interface),
I2C (inter integrated circuit).

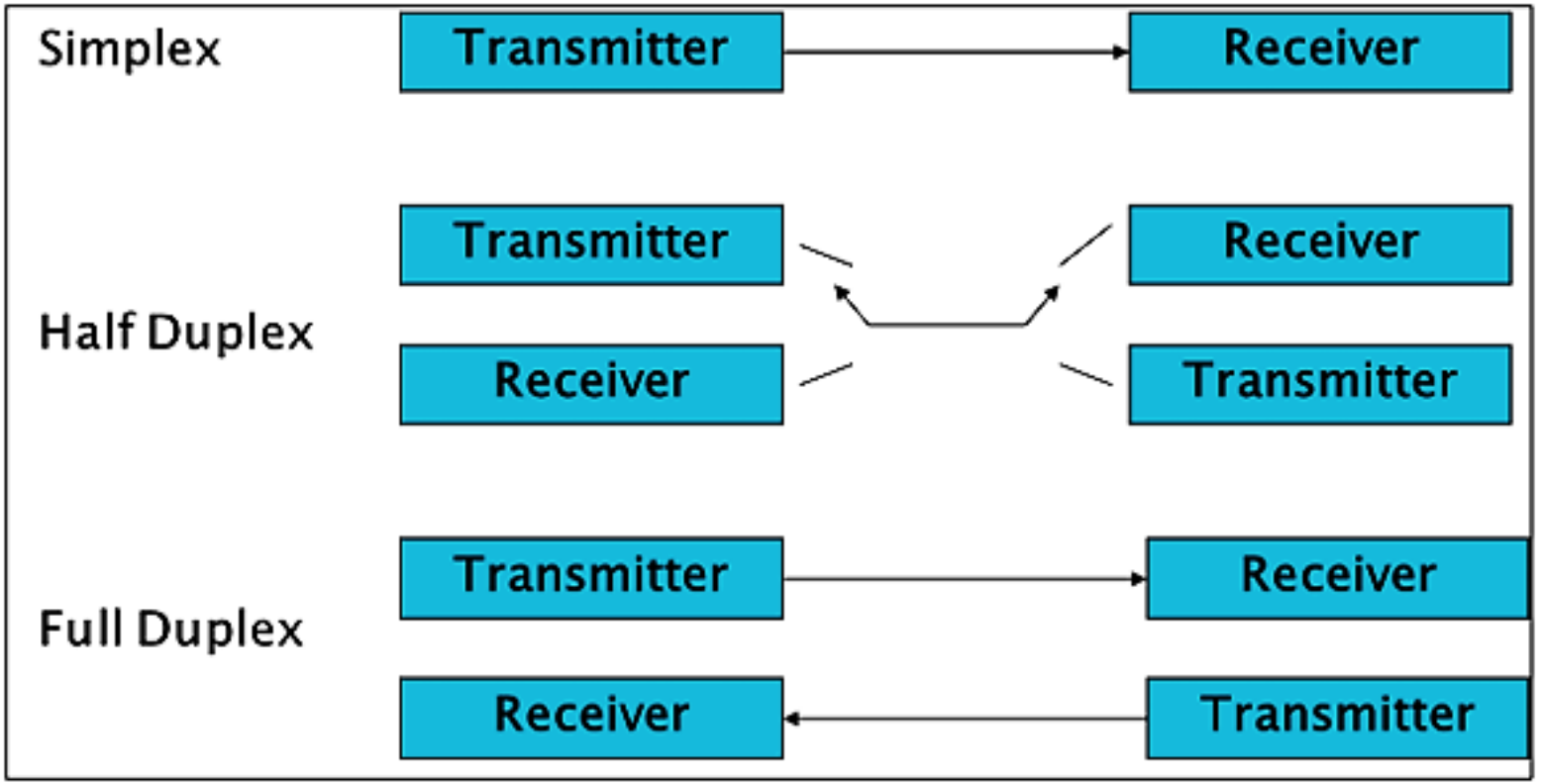
Data transmission

Data Transmission: In data transmission if the data can be transmitted and received, it is a duplex transmission.

Simplex: Data is transmitted in only one direction i.e. from TX to RX only one TX and one RX only

Half duplex: Data is transmitted in two directions but only one way at a time i.e. two TX's, two RX's and one line

Full duplex: Data is transmitted both ways at the same time i.e. two TX's, two RX's and two lines

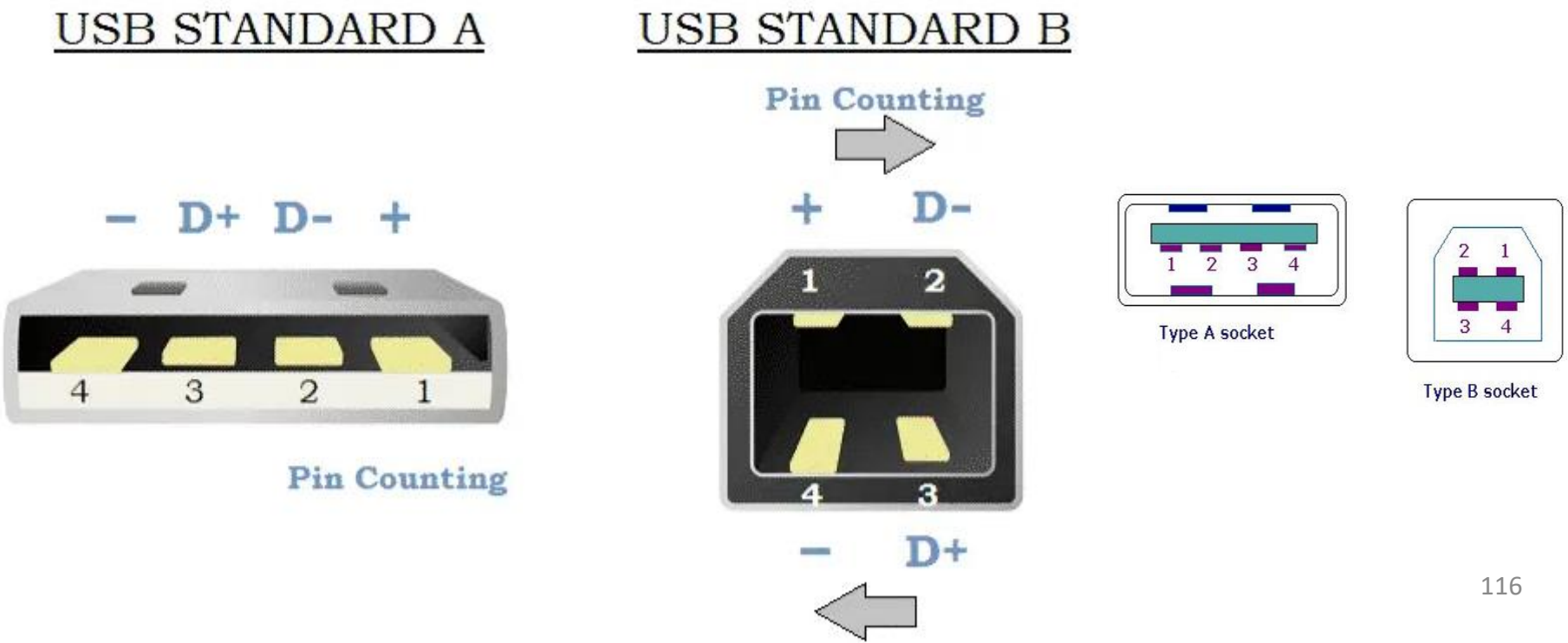


Classify inter system communication protocols of embedded system

USB Communication Protocols:

Universal Serial Bus (USB) is a two-wired serial communication protocol. It allows 127 devices to be connected at any given time. USB supports plug & play functionality.

USB protocol sends and receives the data serially between host and external peripheral devices through data signal lines D+ and D-. Apart from two data lines, USB has V_{CC} and Ground signals to power up the



Classify inter system communication protocols of embedded system

USB Communication Protocols: Data is transmitted in the form of packets where two devices communicate each other. Data packets compose of 8 bits (byte) with LSB (Least Significant Bit) transmitted first.

The advantages of USB Communication Protocol are as follows:

- Fast and simple.
- It is of low cost.
- Plug and Play hardware.

The disadvantages of USB Communication Protocol are as follows:

- Needs powerful master device.
- Specific drivers are required.

Classify inter system communication protocols of embedded system

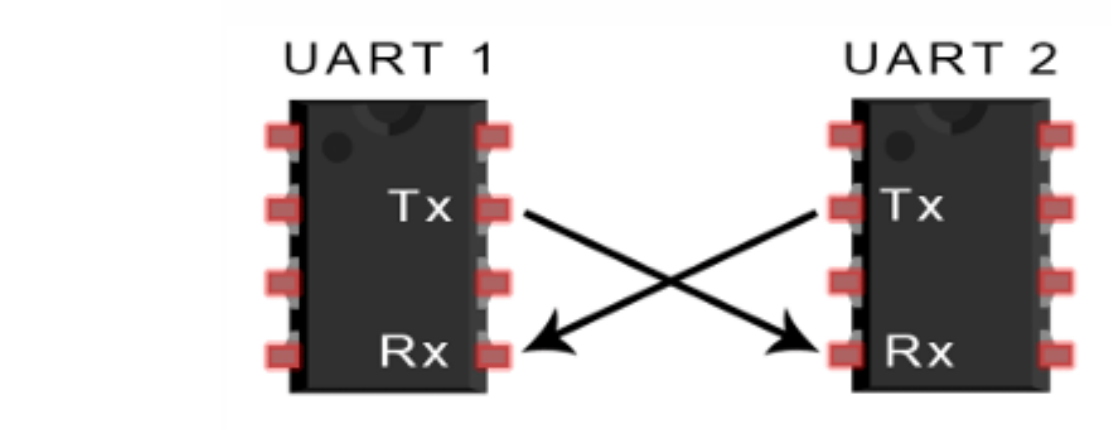
UART Communication Protocols:

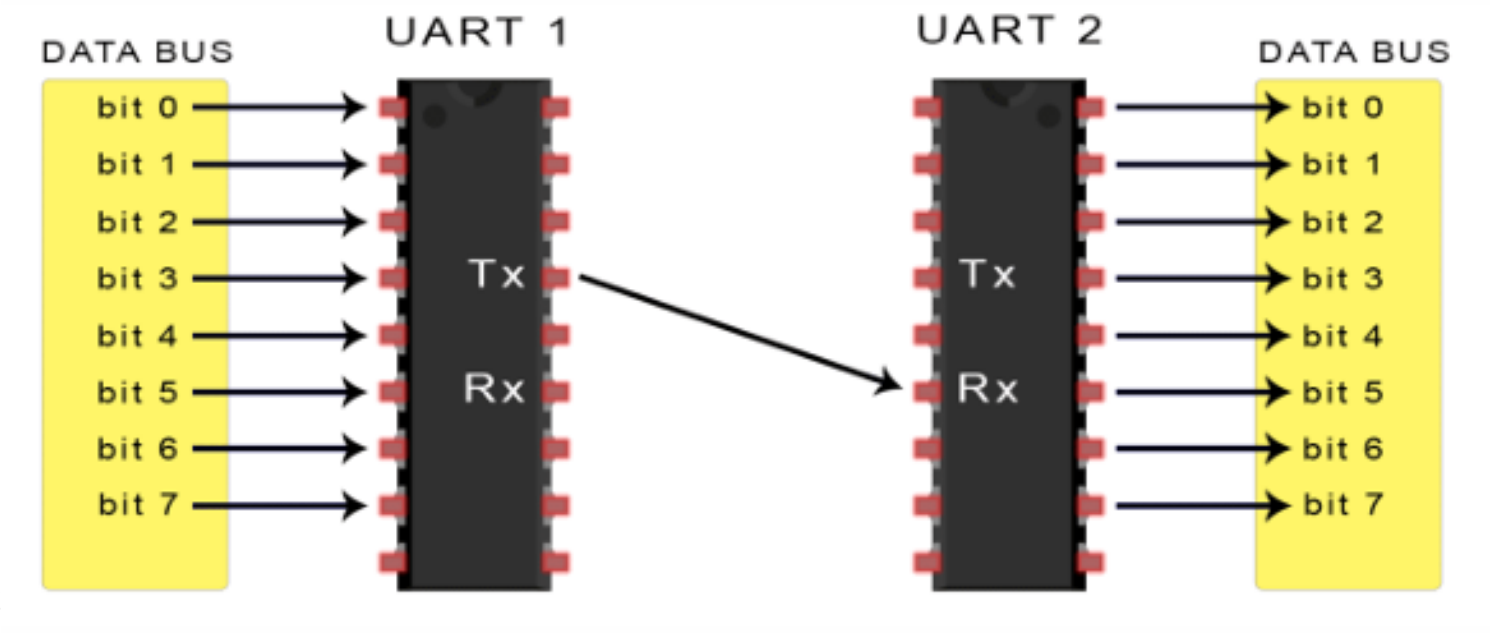
Universal **A**synchronous **R**eceiver/**T**ransmitter (UART) is not a communication protocol but just a physical piece of hardware which converts parallel data into serial data. Its main purpose is to transmit and receive data serially. UART is also two-wired i.e., the serial data is handled by Tx (Transmitter) and Rx (Receiver) pins.

UART transmits data asynchronously, which induces that no clock signal is associated in transmitting and receiving data. Instead of clock signal, UART embed start and stop bits with actual data bits, which defines the start and end of data packet. When receiver end detects the start bit, it starts to read the data bits at specific baud rate meaning both transmitting and receiving peripherals should work under same baud rate. **UART works under half duplex communication mode meaning it either transmits or receives at a time.**

Classify inter system communication protocols of embedded system

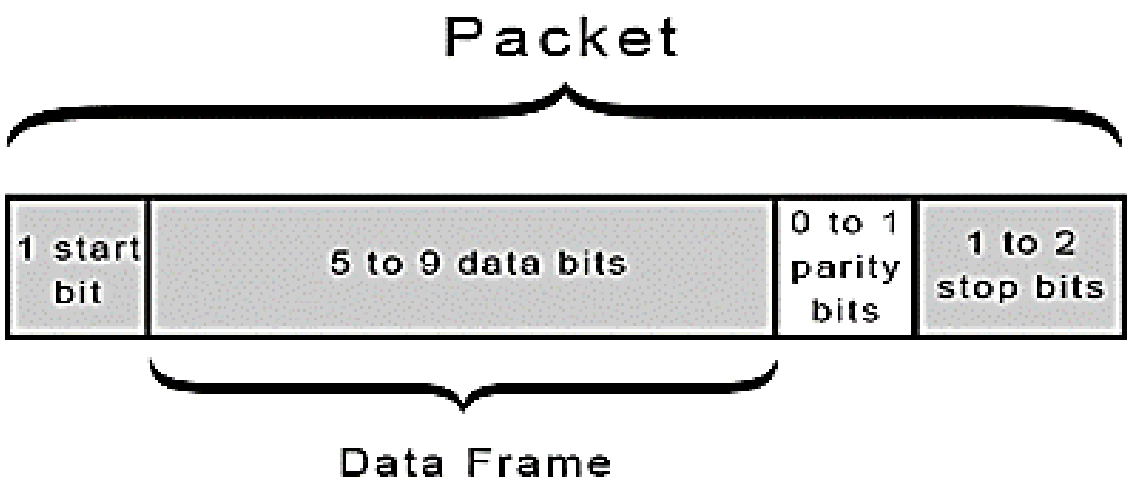
UART Communication Protocols:

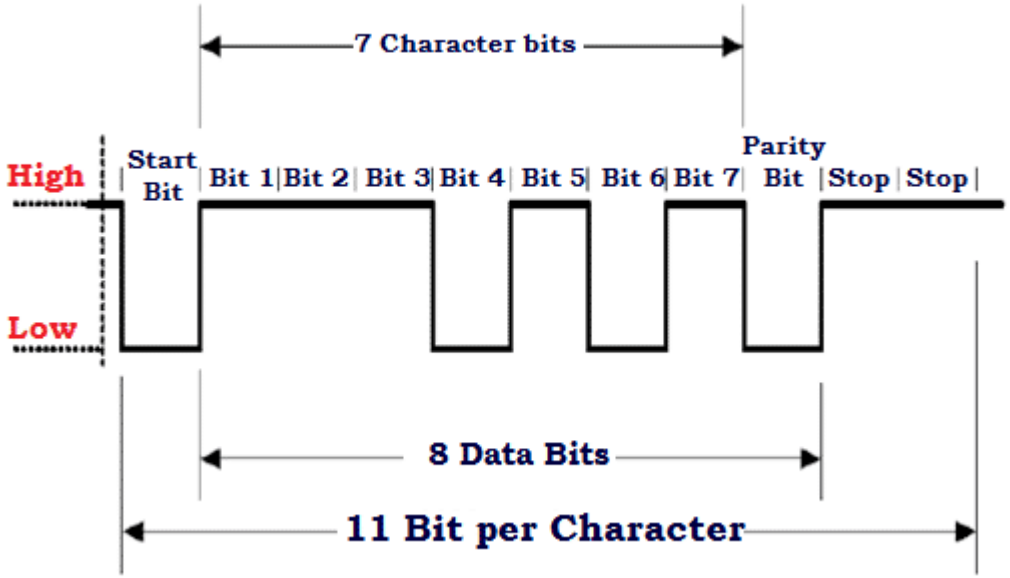




Classify inter system communication protocols of embedded system

UART transmitted data is organized into *packets*. Each packet contains 1 start bit, 5 to 9 data bits (depending on the UART), an optional *parity* bit, and 1 or 2 stop bits:





Classify inter system communication protocols of embedded system

USART Communication Protocol:

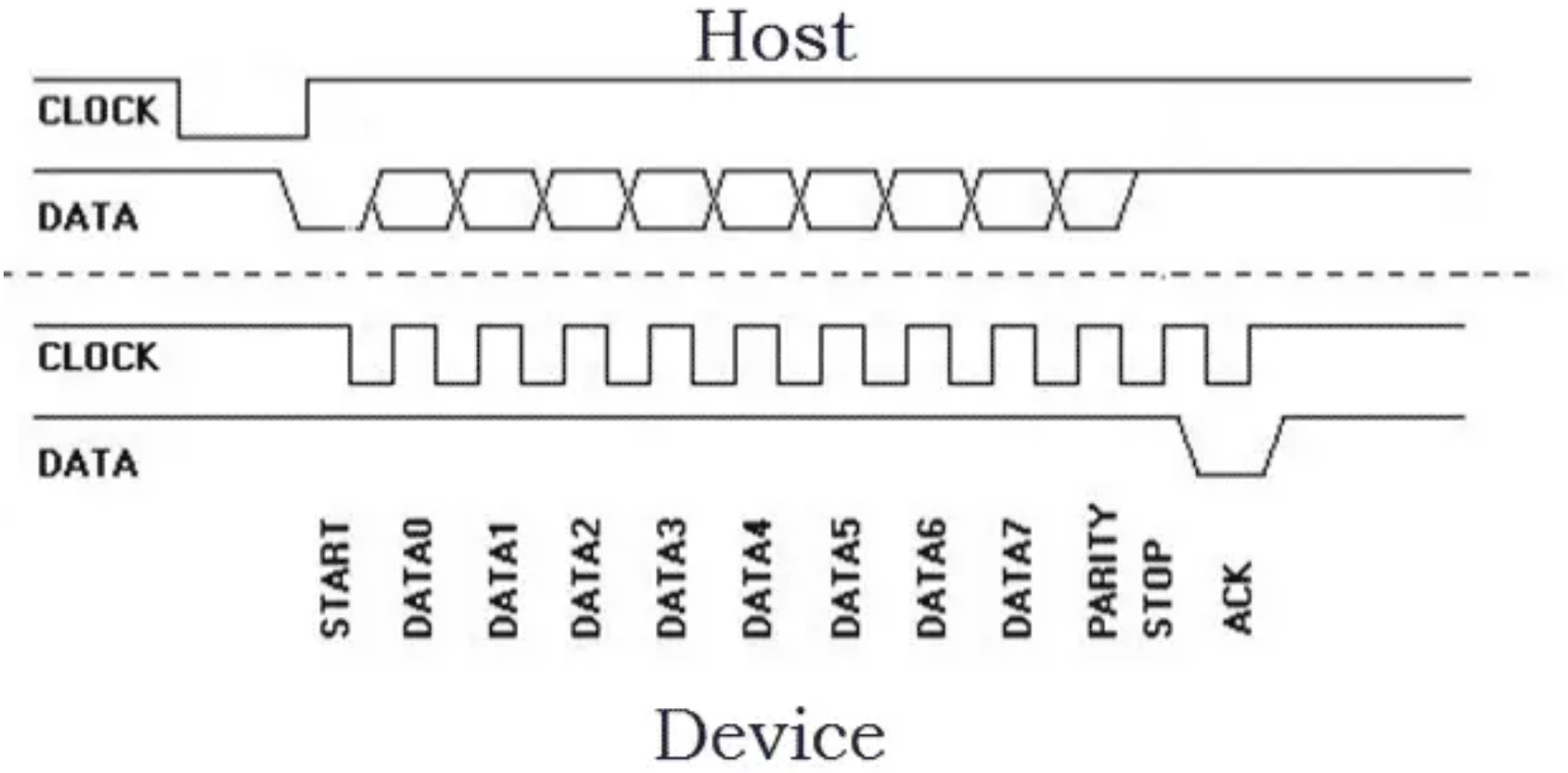
Universal Synchronous Asynchronous Receiver/Transmitter (USART) is identical to that of UART with only added functionality synchronous. That is, the transmitter will generate a clock signal which will be recovered at the receiver end from the data stream transmitted without knowing baud rate ahead.

USART works under full duplex communication mode meaning it can transmit and receive data at same time.

USART encompass the abilities of UART, which enables application of both depending on the applications area.

Classify inter system communication protocols of embedded system

USART Communication Protocol:



Classify inter system communication protocols of embedded system

The advantages of UART/ USART Communication Protocol are as follows:

- Clock signal is not required
- Cost effective
- Uses parity bit for error detection
- Requires only 2 wires for data communication

The disadvantages of UART/ USART Communication Protocol are as follows:

- Doesn’t support multiple master slave functionality
- Baud rate of communicating UART should be within 10 percent of each other

Classify intra system communication protocols of embedded system.

I2C Communication Protocols

Inter Integrated Circuit (I2C) is a serial communication protocol developed by Philips Semiconductors. In embedded systems, all peripheral devices are connected as memory mapped devices to the microcontroller.

I2C necessitates two wires SDA (Serial Data Line) and SCL (Serial Clock Line) to carry information between devices. These two active wires are said to be bidirectional.

I2C protocol is a master to slave communication protocol. Each slave is been provided with unique address. In order to establish communication, master device initially sends the target slave address along with R/W (Read/Write) flag. The corresponding slave device will move into active mode leaving other devices in off state.

Once the slave device is ready, communication starts between master and slave devices. One bit acknowledgment is replied by the receiver if transmitter transmits 1 byte (8 bits) of data. A stop condition is issued at the end of communication between devices.

Classify intra system communication protocols of embedded system.

The advantages of I2C Communication Protocols are as follows:

- Provides good communication between onboard devices which are accessed infrequently
- Addressing mechanism eases master slave communication
- Cost and circuit complexity does not end up on number of devices

Disadvantages of I2C Communication Protocols:

- The biggest disadvantage of I2C Communication Protocols is its limited speed.

Classify intra system communication protocols of embedded system.

Serial Peripheral Interface (SPI) Communication Protocols:

SPI (Serial Peripheral Interface) is one of the serial communication protocol developed by Motorola. It is a 4-wire protocol namely MOSI (Master Out Slave In), MISO (Master In Slave Out, SS (Slave Select), and SCLK (Serial Clock).

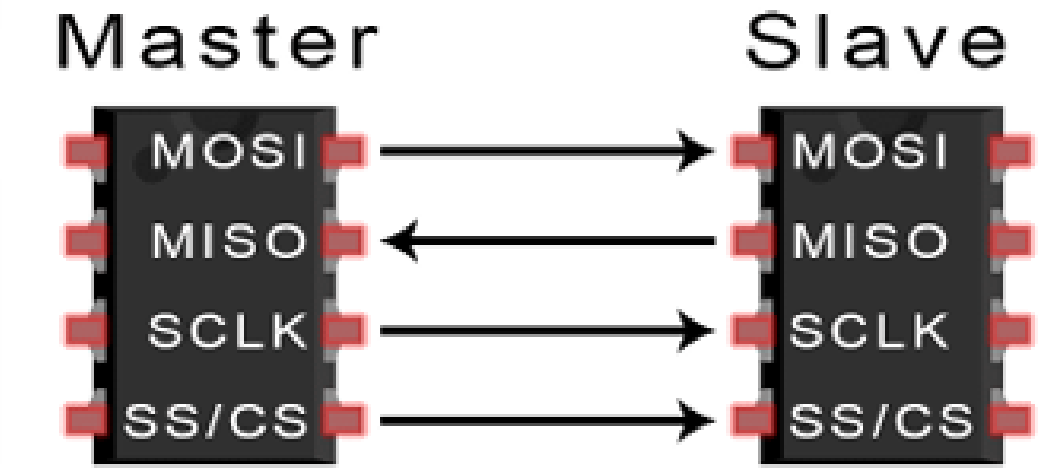
As I2C protocol, SPI is also a master to slave communication protocol. In SPI, the master device first configures the clock at a particular frequency. Furthermore the SS line is used to select the appropriate slave by pulling the SS line low where it is normally held high.

The communication is established between the selected slave and the master device as soon as appropriate slave device is selected.

SPI is a full duplex communication protocol. SPI doesn't limit data transfer to 8 bit words.

Classify intra system communication protocols of embedded system.

Serial Peripheral Interface (SPI) Communication Protocols:



MOSI (Master Output/Slave Input) – Line for the master to send data to the slave.

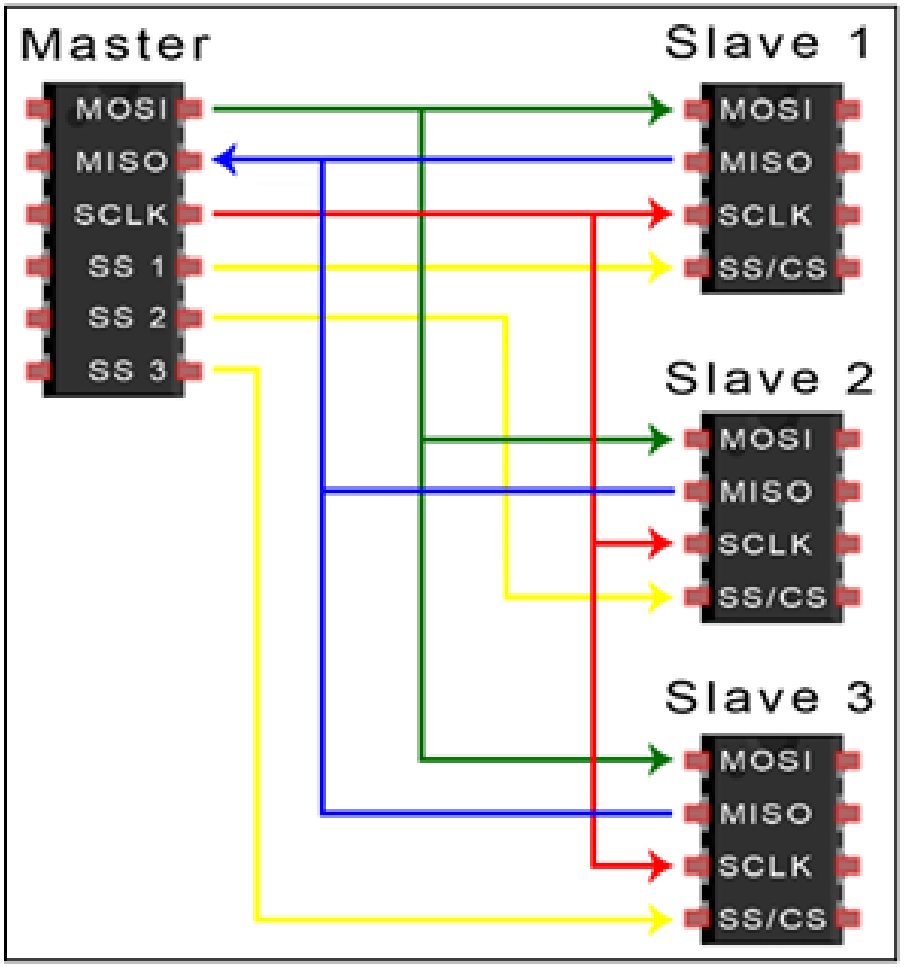
MISO (Master Input/Slave Output) – Line for the slave to send data to the master

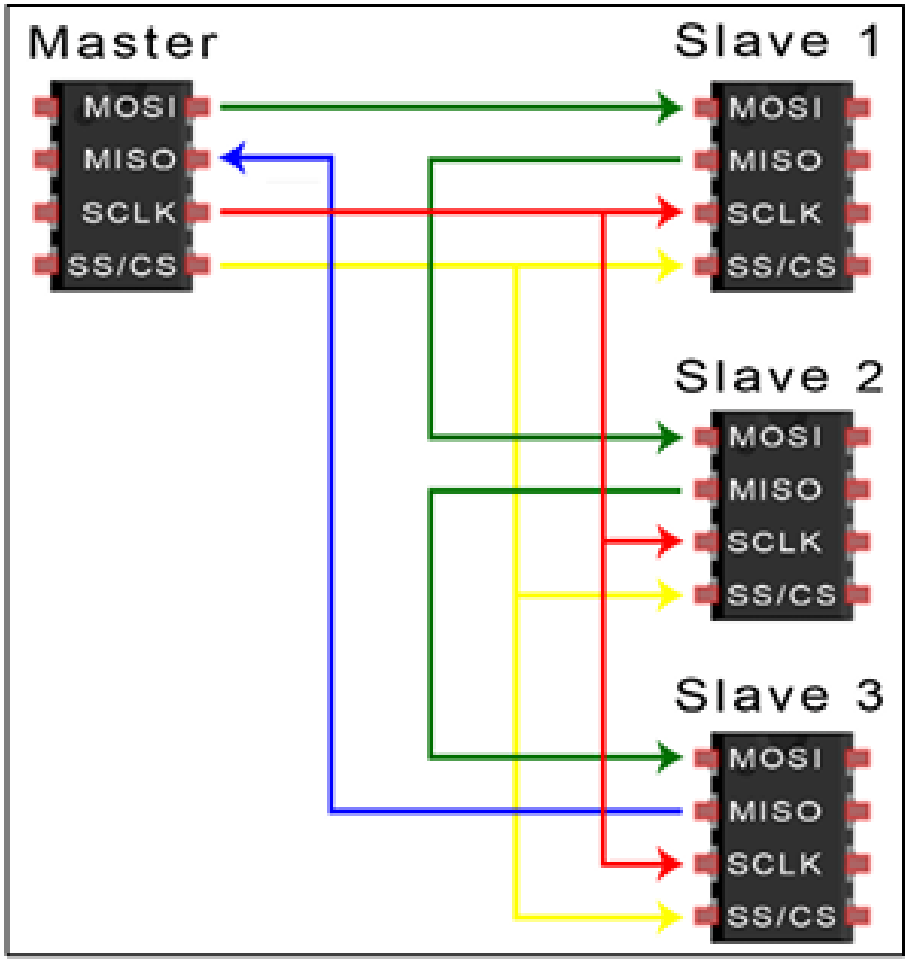
SCLK (Clock) – Line for the clock signal.

SS/CS (Slave Select/Chip Select) – Line for the master to select which slave to send data to.

Classify intra system communication protocols of embedded system.

Serial Peripheral Interface (SPI) Communication Protocols:





Classify intra system communication protocols of embedded system.

The advantages of SPI Communication Protocol are as follows:
•Faster than asynchronous serial communication protocol.
•Support multiple slaves connectivity.
•Universally accepted protocol and low cost.

The disadvantages of SPI Communication Protocol are as follows:
•Requires more wires than other communication protocols.
•Master device should control all slave communications (slave-slave communication is impossible).
•Numerous slave devices leads to circuit complexity.

Classify intra system communication protocols of embedded system.

Controller Area Network (CAN) Communication Protocol:
CAN (Controller Area Network) is a serial communication protocol developed by the Robert Bosch for intra vehicular communication. It requires two wires CAN High (H+) and CAN low (H-) for data transmission. CAN protocol is based on a message oriented communication protocol.

Classify intra system communication protocols of embedded system.

The advantages of CAN Communication Protocols are as follows:

- Low cost and reliable
- Shows robust performance
- Secured and fast protocol

The disadvantages of CAN Communication Protocol are as follows:

- Automotive oriented