8051 Microcontroller

Microcontrollers *are small computing systems on a single chip.*

-Central Processing Unit (CPU)
-Program memory
-Random Access Memory (RAM)
-EEPROM - Electrically Erasable Programmable Read Only Memory
-USARTs, Timer/Counters, ADC, DAC, I/O Ports, CANs, SPIs, etc.

Microprocessor vs Microcontroller:

• The major difference between both of them is the life of external peripherals, where microcontroller have memory (RAM, ROM, EEPROM) embedded in it while we have to use exterior circuits in case of microprocessors.
• As all the microcontroller peripherals are on single IC(Integrated circuit) it is reduced while the microprocessor is huge.
• Microcontrollers are set up by utilizing CMOS (reciprocal metal oxide semiconductor) technology so they are far cheaper than microprocessors. Generally, the applications prepared with these are reasonable because they need less external components, while the entire charge of systems complete with microprocessors is high because of the number of exterior components required for such systems.
• The other difference is that the microcontroller has a power saving system while in microprocessor generally there is no power saving system. In microcontroller use low power and also for external components need consumption of power is less. The compare to microprocessor, it requires high power consumption mode and also several external components are used in different high power consumption.
• The design of Microcontroller depends on Harvard architecture where program and information memory are particular while microprocessors depend on the von-Neumann architecture where both the memories are put away in the equivalent memory module.

Important Features of 8051

- 4K bytes ROM, 128 bytes RAM, Four 8-bit I/O ports, Two 8/16 bit timers, Serial port, 64K external code memory space
64K data memory space, Multiple internal and external interrupt source.

Pin Description of 8051:

- The 8051 is a 40 pin device, but out of these 40 pins, 32are used for I/O. 24 of these are dual purpose, i.e. they can operate as I/ or a control line or as part of address or data bus.

8051 CPU Registers:

- *A (8-bit Accumulator)*
- *B (8-bit register for Mul&Div)*
- *PSW (8-bit Program Status Word)*
- *SP (8-bit Stack Pointer)*
- *PC (16-bit Program Counter)*
- *DPTR (16-bit Data Pointer)*

Types of Computer Architecture:

Basically, Microprocessors or Microcontrollers are classified based on the two types of Computer Architecture: ***Von Neumann Architecture*** *and* ***Harvard Architecture.***

Von Neumann Architecture:

Von Neumann Architecture or Princeton Architecture is a Computer Architecture, where the Program i.e. the Instructions and the Data are stored in a single memory.
Since the Instruction Memory ***(ROM)*** *and the Data Memory* ***(RAM)*** *are the same, the Processor or CPU cannot access both Instructions and Data at the same time as they use a single bus.*
This type of architecture has severe limitations to the performance of the system as it creates a bottleneck while accessing the memory.

Harvard Architecture:

Harvard Architecture, in contrast to Von Neumann Architecture, uses separate memory for Instruction (Program) and Data. Since the Instruction Memory and Data Memory are separate in a Harvard Architecture, their signal paths i.e. buses are also different and hence, the CPU can access both Instructions and Data at the same time.
Almost all Microcontrollers, including 8051 Microcontroller implement Harvard Architecture.

Program Memory (ROM) of 8051 Microcontroller:

- In 8051 Microcontroller, the code or instructions to be executed are stored in the Program Memory, which is also called as the ROM of the Microcontroller. The 8051 has 4KB of internal ROM.

- Some variants of 8051 like the 8031 and 8032 series doesn't have any internal ROM (Program Memory) and must be interfaced with external Program Memory with instructions loaded in it.

- Almost all modern 8051 Microcontrollers, like 8052 Series, have 8KB of Internal Program Memory (ROM) in the form of Flash Memory (ROM) and provide the option of reprogramming the memory.

- In case of 4KB of Internal ROM, the address space is 0000H to 0FFFH. If the address space i.e. the program addresses exceed this value, then the CPU will automatically fetch the code from the external Program Memory.

- For this, the External Access Pin (EA Pin) must be pulled HIGH i.e. when the EA Pin is high, the CPU first fetches instructions from the Internal Program Memory in the address range of 0000H to 0FFFFH and if the memory addresses exceed the limit, then the instructions are fetched from the external ROM in the address range of 1000H to FFFFH.

*- There is another way to fetch the instructions: ignore the Internal ROM and fetch all the instructions only from the External Program Memory (External ROM). For this scenario, the EA Pin must be connected to GND(**ground**). In this case, the memory addresses of the external ROM will be from 0000H to FFFFH.*

Data Memory (RAM) of 8051 Microcontroller

- The Data Memory or RAM of the 8051 Microcontroller stores temporary data and intermediate results that are generated and used during the normal operation of the microcontroller. Original Intel's 8051 Microcontroller had 128B of internal RAM.

- But almost all modern variants of 8051 Microcontroller have 256B of RAM. In this 256B, the first 128B i.e. memory addresses from 00H to 7FH is divided in to Working Registers (organized as Register Banks), Bit – Addressable Area and General Purpose RAM (also known as Scratchpad area).

- In the first 128B of RAM (from 00H to 7FH), the first 32B i.e. memory from addresses 00H to 1FH consists of 32 Working Registers that are organized as four banks with 8 Registers in each Bank.

- The 4 banks are named as Bank0, Bank1, Bank2 and Bank3. Each Bank consists of 8 registers named as R0 – R7. Each Register can be addressed in two ways: either by name or by address.

- To address the register by name, first the corresponding Bank must be selected. In order to select the bank, we have to use the RS0 and RS1 bits of the Program Status Word (PSW) Register (RS0 and RS1 are 3rd and 4th bits in the PSW Register).

- When addressing the Register using its address i.e. 12H for example, the corresponding Bank may or may not be selected. (12H corresponds to R2 in Bank2).

- The next 16B of the RAM i.e. from 20H to 2FH are Bit – Addressable memory locations. There are totally 128 bits that can be addressed individually using 00H to 7FH or the entire byte can be addressed as 20H to 2FH.

***For example:** 32H is the bit 2 of the internal RAM location 26H.*

- The final 80B of the internal RAM i.e. addresses from 30H to 7FH, is the general purpose RAM area which are byte addressable.

- These lower 128B of RAM can be addressed directly or indirectly.

- The upper 128B of the RAM i.e. memory addresses from 80H to FFH is allocated for Special Function Registers (SFRs). SFRs control specific functions of the 8051 Microcontroller. Some of the SFRs are I/O Port Registers (P0, P1, P2 and P3), PSW (Program Status Word), A (Accumulator), IE (Interrupt Enable), PCON (Power Control), etc.

- SRFs Memory addresses are only direct addressable. Even though some of the addresses between 80H and FFH are not assigned to any SFR, they cannot be used as additional RAM area.

In some microcontrollers, there is an additional 128B of RAM, which share the memory address with SFRs i.e. 80H to FFH. But, this additional RAM block is only accessed by indirect addressing.

Interfacing External Memory with 8051 Microcontroller

- We have seen that a typical 8051 Microcontroller has 4KB of ROM and 128B of RAM (most modern 8051 Microcontroller variants have 8K ROM and 256B of RAM).

- The designer of an 8051 Microcontroller based system is not limited to the internal RAM and ROM present in the 8051 Microcontroller. There is a provision of connecting both external RAM and ROM i.e. Data Memory and Program.

- The reason for interfacing external Program Memory or ROM is that complex programs written in high – level languages often tend to be larger and occupy more memory.

- Another important reason is that chips like 8031 or 8032, which doesn't have any internal ROM, have to be interfaced with external ROM.

- A maximum of 64B of Program Memory (ROM) and Data Memory (RAM) each can be interface with the 8051 Microcontroller.

Timers /Counters:

The 8051 has 2 timers/counters: Timer/Counter 0 & Timer/Counter 1

Timer :

- Used as a time delay generator.

- Many microcontroller application requires the counting of external events such as frequency, time delay.

Registers Used in Timer/Counter:

- 8051 has two 16-bit Timer registers ,Timer 0 & Timer 1.

*- As 8051 has 8-bit architecture , each Timer register is treated as two 8-bit registers namely **TH0, TL0, TH1, TL1.***

*- One 8-bit mode register **-TMOD**.*

*- One 8-bit control register-**TCON.***

Counter:

• Count the number of events

• Show the number of events on registers

• External input from T0 input pin (P3.4) for Counter 0

• External input from T1 input pin (P3.5) for Counter 1

• External input from Tx input pin.

• We use Tx to denote T0 or T1.

TMOD Register:

• Both Timer 0 & Timer 1 use the same Mode register TMOD.

• It is an-8-bit register .

• The lower 4-bits are meant for Timer 0 & the upper 4-bits are meant for Timer 1

• It is used similar to any other register of 8051 .

• C/T : Timer or counter selected cleared for timer operation (input from internal system clock).

• M1,M0 : Used for mode selection. Because the Timers of 8051 can be set in 4-different modes.

Gate:

- Every timer has a mean of starting and stopping.

- GATE=0: Internal control, The start and stop of the timer are controlled by way of software.

- GATE=1: External control, The hardware way of starting and stopping the timer by software and an external source.

- Timer/counter is enabled only while the INT pin is high and the TR control pin is set (TR).

TCON Register:

- Timer control register TMOD is a 8-bit register which is bit addressable and in which Upper nibble is fortimer/counter, lower nibble is for interrupts.

TR (Timer run control bit)

- *TR0 for Timer/counter 0; TR1 for Timer/counter 1.*
- *TR is set by programmer to turn timer/counter on/off.*

  TR=0 : off (stop)

  TR=1 : on (start)

TF (timer flag, control flag)

- *TF0 for timer/counter 0; TF1 for timer/counter 1.*
- *TF is like a carry. Originally, TF=0. When TH-TL roll over to 0000 from FFFFH, the TF is set to 1.*

  TF=0 : not reach

  TF=1: reach

  If we enable interrupt, TF=1 will trigger ISR.

RxD and TxD pins:

- The 8051 has two pins for transferring and receiving data by serial communication.

Interrupt Sources:

8051 has 6 sources of interrupts

- *Reset*
- *Timer 0 overflow*
- *Timer 1 overflow*
- *External Interrupt 0*
- *External Interrupt 1*
- *Serial Port events (buffer full, buffer empty, etc)*

- Upon reset all Interrupts are disabled and do not respond to the Microcontroller These interrupts must be enabled by software in order for the Microcontroller to respond to them. This is done by an 8-bit register called Interrupt Enable Register (IE).

Interrupt Priorities

If two interrupt sources interrupt at the same time ,the interrupt with the highest PRIORITY gets serviced first.
All interrupts have a power on default priority order.

1. *External interrupt 0 (INT0)*
2. *Timer interrupt0 (TF0)*
3. *External interrupt 1 (INT1)*
4. *Timer interrupt1 (TF1)*
5. *Serial communication (RI+TI)*

A high-priority interrupt can interrupt a low-priority interrupt

- *All interrupt are latched internally*
- *Low-priority interrupt wait until 8051 has finished servicing the high-priority interrupt*