MODULE 5 : MICROCONTROLLERS

- Micro controller is a single chip computer or a CPU with all the peripherals like RAM, ROM, I/O ports, Timers etc. on the same chip
- It is a small computer on a single IC
- It can be called as a heart of embedded systems
- Microcontrollers are designed to execute a single specific task to control a single system.
- Microcontrollers are generally used in projects and applications that required direct control of user
- Eg: 8051, Motorola's 6811, Z8, PIC 16x etc.

MICRO PROCESSOR	MICROCONTROLLER
General purpose device which is called a CPU	Dedicated chip which is also called single chip computer
Do not contain on-chip I/O port, timers, memories etc.	It include RAM, ROM, serial and parallel interface, timers etc.
Instructions are mainly nibble or byte addressable	Instructions are both bit addressable as well as byte addressable
Design is complex and expensive	Design is rather simple and cost effective
Large number of instructions and is complex	Less number of instructions and is simple
Microprocessor has Zero status flag	Microcontroller has no zero flag
Heart of computer system	Heart of embedded system

MICRO PROCESSOR	MICROCONTROLLER
Since memory and I/O has to be connected externally the circuit become large	Since memory and I/O are present internally, the circuit is small
Less number of registers, hence more operations are memory based	Have more number of registers, hence the programs are easier to write
Mainly used in personal computers	Used in washing machine, MP3 players etc.
These are based on Von Neumann model	Based on Harvard architecture

TYPES OF MICROCONTROLLERS

- Microcontrollers are classified in terms of
  - Internal bus width
  - Embedded microcontroller
  - Instruction set
  - Memory architecture
  - IC chip or VLSI core file and family

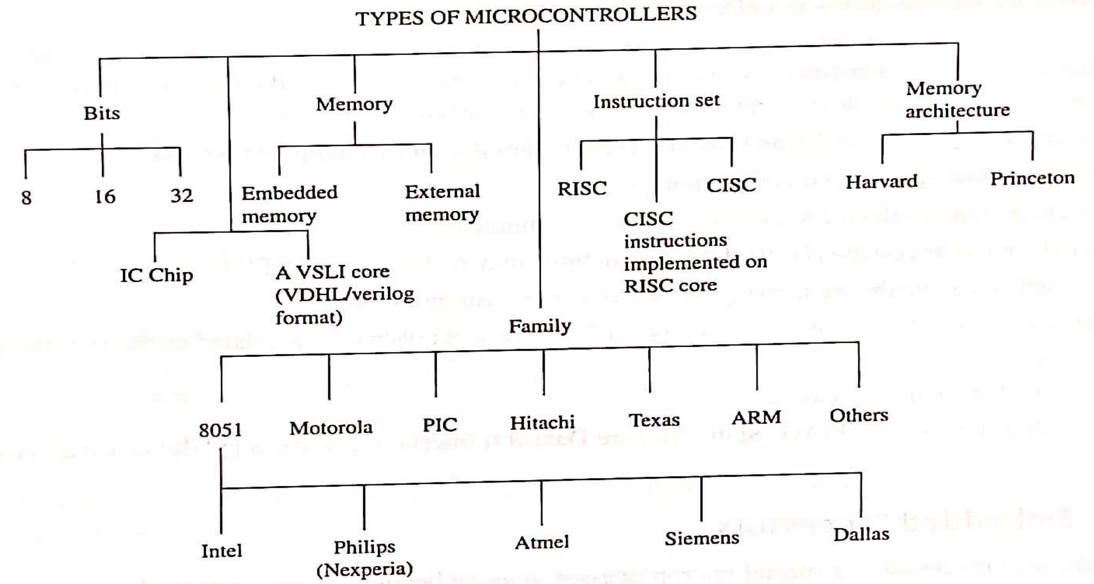


Figure 1.5 Various Types of Microcontrollers

1. The 8, 16 and 32 bit microcontrollers

➢ 8 bit microcontrollers

- Internal bus in an MCU is 8 bit bus
- ALU performs Arithmetic and logical operations on a byte at an instruction
- Eg: INTEL 8031/ 8051, PIC 1x

➢ 16 bit microcontrollers

- Internal bus in an MCU is 16 bit bus
- ALU performs Arithmetic and logical operations on the operand word of 16 bits at the instruction
- Provide greater precision and performance
- Eg: INTEL 8096, PIC 2x

- 32 bit microcontrollers
  - Internal bus is 32 bit bus
  - ALU performs Arithmetic and logical operations on the operand word of 32 bit at the instruction
  - Used in mobile phones, MP3 audio systems and aerospace systems
  - Eg: PIC 3x, INTEL/Atmel 251 family

2. Embedded and External memory microcontrollers

- Embedded microcontroller
  - MCU has all the hardware and software units in a single unit
  - Used in an embedded system
  - Eg: Telephone handset circuit uses an embedded microcontroller
- External memory microcontroller
  - MCU has all the hardware and software units not as a single unit
  - Used in an embedded system
  - All or part of the memory unit externally interfaced using an interface circuit called **glue circuit**
  - Eg: 8031 has a program memory which is interfaced externally to it
  - Eg: 8051 has both internal as well as external program memory

3. CISC and RISC architecture microcontrollers

➢CISC architecture microcontroller

- CISC –Complex Instruction Set Computers
- Instruction set of MCU support many addressing modes for arithmetic and logical instructions
- Memory access during ALU operations and data transfer instructions
- Provide flexibility in choosing various ways of performing the data transfer , arithmetic and other operations.
- Instructions are variable number of bytes in CISC .
- Eg: INTEL 8096

➢RISC microcontrollers

- RISC- Reduced Instruction Set Computer
- Instruction set of MCU support a few addressing modes for arithmetic and logical instructions
- A few instructions for data transfer (load, store, push & pop)
- RISC implements each instruction in a single cycle using a distinct hardwired control
- Uses lesser amount of circuitry
- Less power dissipation
- Instructions are of fixed number of bytes
- Many registers. Memory fetching is generally reduced
- Eg: ARM processor family- based MCU

4. **Harvard and Princeton memory architecture microcontrollers**

➢ Harvard memory architecture

- MCU has a distinct memory address space for program and data memory.
- MCU has separate instructions
- Has separate control signals for data transfer from the two memories
- Eg: 8051 has

address space between 0x0000 and 0xFFFF for the program memory bank

separate memory between 0x0000 and 0xFFFF for the data memory bank

➢ Princeton memory architecture

- Common memory address space is used for the program memory and data memory
- Eg: 68HC II has an address space between 0x0000 and 0xFFFF for the program memory codes and the same space between 0x0000 and 0xFFFF for the data bytes.

Criteria for selecting a microcontroller

1. Basic features to consider while selecting an MCU (Microcontroller Unit)

- 8 bit or 16 bit or 32 bit MCU
- Power dissipation and the clock speed required
- RISC/CISC instruction set
- Program storage structure- Harvard or Princeton memory
- External & internal memory required
- DMA controller requirement if any
- Cache memory requirement if any
- Intensive computation at fast rate required if any

2. **Additional factors to consider while selecting an MCU**

- Cost when organized as single chip and with MCU interfaces to external devices
- Major building hardware blocks and their cost and availability
- Ease of integration
- Major building blocks of software, their cost and availability
- Availability of design team expertise
- Availability and cost of software development and hardware tools
- Ease and availability of testing and debugging facilities
- Easy and reliable availability of MCU

3. Selection of a processor and processor family

- Firstly, decide whether to use a microprocessor or a microcontroller base design and also whether to use an 8-bit or 32-bit MCU

If there are intensive computations and large memory needed, then select microprocessor based design

If computations are not intensive and chip or single unit design is preferred, then select MCU based design

- Secondly the selection of the processor family is made

While selecting family, consider MCU cost, Availability, Cost of development tools and effort needed for the design

- Finally, consider the selection of a processor family member as a chip or a VLSI core

4. Selection based on on-chip resource

- In the MCU family, the available resources can be variable.
- The cost of an MCU depends on the on-chip resource (internal hardware blocks)
- Eg: Some MCU of 8051 member may have 4KB internal program memory and other 8KB.
- Select according to the program size , if cost differential is minimal.

5.Selection based on Development Tool

- Selection of appropriate hardware and software development tool is also important when deciding an MCU
- IDE (Integrated Development Environment) provides the project management tools. IDE gives the make facility. IDE enables the following

✓ Creation of source file

✓ Organization of source files in to a project

✓ Provision of databases for many devices

✓ Automatically it compiles, points errors and interactively correct errors, Assembles and link the application

- Example of IDE is **MicroVision 2 .** It is Keil Software (Now Keil is an ARM company). The two models of IDE are

❖ **Build Mode** → Build mode of MicroVision2 is for installing the application files and for generating executable files.

❖ **Debug Mode** → For thorough testing of the application from the generated executable files.

- Eg: MPLAB (IDE) → It is for PIC microcontrollers. It is MS Window based tool set.

Applications of Microcontrollers

- In biomedical instruments
  - ECG LCD display cum recorder
  - Blood cell recorder cum analyzer
  - Patient monitor system
- In communication systems
  - Numeric pagers
  - Cellular phones
  - Cable TV terminals
  - Video games

- Peripheral controllers of a computer
  - Keyboard controller
  - Printer controller
  - LAN controller
  - Disk drive controller
- Instruments
  - Industrial process controller
  - Electronic smart weight display system
- Automatic signal tracker
- Robotics system
- CNC machine controller
- CRT display controller

Microcontroller On-Chip Resources

1. **Basic processing unit, Internal buses & Interrupt handling**

Basic processing unit :

- Processing unit fetches, decodes , read operands, executes and write back the result during processing an instruction.
- After each fetch of the instruction, the program counter (PC) sets for the next instruction.
- The processing unit has an IR (Instruction Register) contains fetched instruction. It has an ID (Instruction Decoder) which decodes the fetched instruction.

- Contains ALU and PSW (Processor Status Word). The PSW save the flags after each ALU operation. It also save the processor status, for example what is the status of an interrupt, enabled or disabled.
- Contains Oscillator circuit, which controls the generation of the sequence of control signals by the control and sequence logic.
- Contains internal reset circuitry connected to an external reset circuit through the reset pin. Reset unit controls the starting of the processor on power-up.

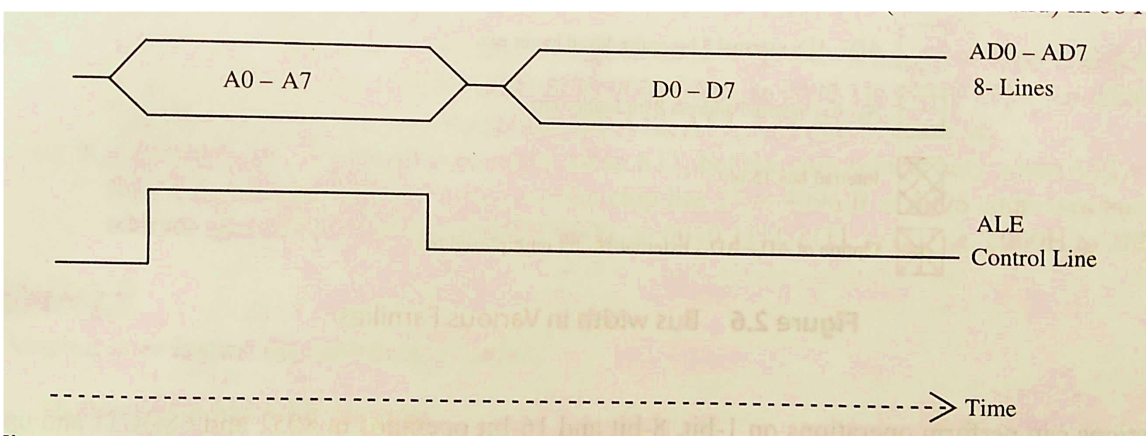
Internal Buses:

- Subunits in the microcontroller/microprocessor are interconnected through a set of parallel lines called internal buses
- Internal buses carries signals(bits) from one subunits to another.

External Buses:

- Many units in the system such as microprocessor or microcontroller, RAM and Flash memory and I/O devices are interconnected through a set of parallel lines called external buses
- Three set of external buses are
  - ➢Control bus
  - ➢Address bus
  - ➢Data bus
- Buses carry signals (bits) from one unit to another at an instant
- External data bus D_0 to D_7 multiplexes with lower address bus A_0 to A_7 by multiplexing in 8051

- In the following figure(timing diagram) when ALE (Address Latch Enable) is active (=1), there is an address at bus. When ALE is inactive (=0) there can be the data on the bus



Interrupt Handling

- Processing unit has an interrupt handler and connects to the external interrupt sources through the INTR pin.
- Eg: $\overline{\text{INT0}}$, $\overline{\text{INT1}}$ pins in 8051

2.Program and Data Memory

- Memory has large number of addresses. Each address save 1 byte (8 bit)
- When data bus width is 8 bit , then one byte can be write and read at an instant.
- When data bus width is 32 bit, then 4 byte can be write and read.

- Writing and Reading by the controller. If $\overline{\text{RD}}$ bus signal is active , then there is a read from the address. If $\overline{\text{WR}}$ control signal is active, then there is a write to the address in the memory.
- Each memory chip has a Chip-Select input ($\overline{\text{CS}}$). The address decoder circuit decodes the input address and is used for selecting the appropriate memory chip.

Program Memory

- Program memory can be On- Chip or external in an MCU (Mostly a non-volatile ROM)
- Program memory stores read only codes , constant input data and parameters of the program.

❖EPROM → ROM can be EPROM. EPROM can be erased by exposing the chip to the UV light.

❖Flash Memory → Can be erased by electric signals. Complete sector of memory is erased and all bytes at that sector become 1111 1111b (FFH) in one cycle. A sector may be 1KB, 2KB or 4KB.

❖EEPROM → Can be erased by electrical signals. Byte can be erased (1 or 2 or 4 or 16 or 64) in one cycle.

Data Memory

- Data and stack in an MCU are in RAM
- MCU has an internal RAM (128 Byte in 8051)

3. Parallel ports

- Input port → bits are placed by an external device and are taken into register or memory address using an instruction.
- Output port → bits are send for the output using an instruction.

ports are two types : Parallel & Serial

- A parallel 8 bit port means all the 8 bit can be placed at all its pins simultaneously
- A serial 8 bit port means all the 8 bits of a byte can be placed at its pin in 8 time slots.
- An MCU parallel port is of 8 bits
- MCUs have memory mapped I/O devices

4. On- Chip registers

- 8051 has 32 on-chip registers. These are partitioned in to 4 sets of register bank.
- Register – set select bits are **RS_0** and **RS_1**

- **RS_1 - RS_0 set as 11 , then bank 3 is accessed**
- **RS_1 - RS_0 set as 10 , then bank 2 is accessed**
- **RS_1 - RS_0 set as 01 , then bank 1 is accessed**
- **RS_1 - RS_0 set as 00 , then bank 0 is accessed**

5. Special Function Registers

Accumulator	**Serial buffer**
B register	**Serial control**
PSW	**Interrupt enable**
Timer Control Register	**Interrupt Priority**
Timer Mode Register	**Power control**
T_0 Timer counts register	**Stack pointer**
T_1 Timer count register	**Data pointer lower byte**
Port	**Data pointer upper byte**

6. UART (Universal Asynchronous Receiver and Transmission)

- Popular mode of serial communication
- In serial communication , a stream of 1s and 0s is sent to or received at successive time intervals of Δt on a single line called serial line.
- Transmission and reception in UART mode is asynchronous (Asynchronous serial communication)

Prepared By EBIN P.M APCSE MEA ENGINEERING COLLEGE , MALAPPURAM 31

7. Synchronous Serial Communication

- MCU can possess a synchronous serial line. It also accompanies a synchronizing clock
- Each byte (frame of bits) on the serial line needs to maintain the same phase difference
- Transmitter can communicate explicitly or implicitly to the receiver using clock bits
- Explicitly means, data and clock bits are send on two separate lines to the receiver.
- Implicitly means, the clock and data bits are combined and the receiver can separate bits back

Prepared By EBIN P.M APCSE MEA ENGINEERING COLLEGE , MALAPPURAM 32

- Transmitter buffer sends a byte to parallel in serial out (PISO) shift register at each clock pulse.
- The receiver buffer receives a byte from serial in parallel out (SIPO) shift register at each clock pulse.
- Transmitter can also be called master. Receiver can also be called slave

8. Timers / Counters

- Timers are must for real time operations in any system
- A 16 bit timer/counter can count from 0000H to FFFFH
- An internal clock is used in the timer mode. The internal clock pulses increment the counter.
- 8051 family MCU has 2 programmable timers, T_0 and T_1

9. On-Chip A/D converters (ADC)

- Microcontroller has ADC conversion feature. The ADC starts conversion on each negative going pulse.

10. Watchdog Timer (WDT)

- It is a timing device that resets the system after a pre-defined time out
- Most microcontrollers have an On-Chip WDT
- WDT rescues the system if a fault develop in between

11. Bit-wise manipulation capability

- 8051 family MCU has a powerful bit manipulation capability
- The carry flag bit (C Flag) playing the role of an accumulator of one bit. The C flag can be set or reset

12. Power down mode

- MCU based system need not be switched off at any time. In this case , the processor runs in the power down mode.
- MCU initiates certain actions on execution of the WAIT and STOP instructions
- During the STOP state an MCU disconnects external devices and internal clock circuit is deactivated.

13. Real-Time clock

- Using this, an OS sets the system clock, schedules the task and time delay functions.
- It is an on-chip device made from the timer working in non-reset, non-loadable and non-stop mode

14. Reset circuit

- A reset circuit or device resets the microcontroller so that a processor starts smoothly
- All the processor registers and devices and interfaced devices get the default values
- It forces the processor to start the processing of instructions from a starting address smoothly

15. Oscillator circuit

- It is required for generating internal states and clock pulses

8051 Microcontroller - Architecture

- 8051 has 4KB internal program memory.
- The 8051 has
  - Processor
  - ROM & RAM
  - Interrupt control circuit
  - Internal timing devices (timers T0 & T1)
  - Serial Interface (SI)
  - Special Function Registers (SFR)
  - Four Ports (P0, P1, P2 and P3)

➢Features in 8051 are given below:

1. 8051 can be used as a single- chip micro computer with embedded program in ROM or Flash. The ROM is 4KB in 8051
2. 8051 has 128Byte RAM. There are a number of Special Function Registers
3. Each program needs a stack. 8051 has a stack . SP(Stack pointer) is 8 bit register
4. There is a program counter. The lower bytes of program counter is sent at the bus A_0-A_7. The A_0-A_7 bus pins are also common to the data bus D_0-D_7
5. There is a Program Counter higher byte at bus A_8-A_{15}. The A_8-A_{15} bus pins are used in the expanded chip mode.

6. A microcontroller can be connected to I/O devices, using ports

7. There are two external pins for interrupts INT0 & INT1. There is interrupt control circuit for the 8051 interrupt service mechanism. Two special function registers IP (Interrupt Priority) and IE (Interrupt Enable) used for priorities and mask.

8. There are two programmable timers/event counters , T_0 & T_1 to do real time control of event and task.

The internal architecture of 8051 contains the following blocks:

1. **Accumulator (A) :** 8 bit register, saves the operand for operations by ALU. Its important function is to accumulate the result after an ALU operation.
2. **B register (B) :** 8 bit register B saves a second operand for the ALU. It accumulates the part of the result of multiplication or division.
3. **PSW (Processor Status Word):** 8 bit register to save the status information
4. **Stack Pointer (SP):** 8 bit register is incremented before the data is stored on to the stack using PUSH instruction.

5. **Data Pointer (DPTR) :** 16 bit register , holds the external data memory address of the data being currently fetched or to be fetched. Data is fetched using indirect addressing mode. The DPTR consist of 2 bytes DPH(Higher) DPL (Lower)
6. **Port P0:** 8 bit port **P_0** is for I/O in a single chip mode. In the expanded mode P_0 is for the data bus- cum-lower order address signals AD_0-AD_7
7. **Port P2:** 8 bit port P_2 is for the I/O in a single chip mode. In the expanded mode P_2 is for higher order address signals A_8-A_{15}
8. **Port P1:** 8 bit port P_1 is for the I/O in a single chip mode . In the expanded mode it is used for few interfacing signals.

9. Port P3: 8 bit port P3 is for the I/O in a single chip mode.

- Also used for serial interface signals
- P3 pin also used for timer T0 and T1 inputs, and interrupts $\overline{\text{INT0}}$ and INT1 inputs.
- P3 pin is also used for sending $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals for the memory read , write in the expanded mode.

10. Serial Data Buffer (SBUF): It internally contains two independent registers

- Transmit buffer → Parallel-in Serial – out register
- Receive buffer → Serial – in Parallel-out register

11. Timer Register: These are two 16 bit registers ,can be accessed as their lower and upper bytes

- TL0 → Lower byte of the timing register 0
- TH0 → Higher byte of the timing register 0
- TL1 → Lower byte of the timing register 1
- TH1 → Higher byte of the timing register 1

12. Control Registers : The special function registers IP, IE, TMOD, TCON, SCON and PCON contain control and status information for interrupts, timers/counters and serial port.

- IP – Interrupt Priority register
- IE –Interrupt Enable register
- TMOD – Timer mode register

- TCON – Timer Control register
- SCON – Serial Control register
- PCON – Power Control register

13. Timing & Control unit : This unit derives all the necessary timing and control signals required for the internal operation of the circuit. It also derives control signals required for controlling the external system bus.

14. Oscillator : This circuit generate the basic timing clock signals for the operation of the circuit using crystal oscillator

15. Instruction Register: This register decodes the opcode of an instruction to be executed , and gives information to the timing and control unit to generate necessary signals for the execution of the instruction.

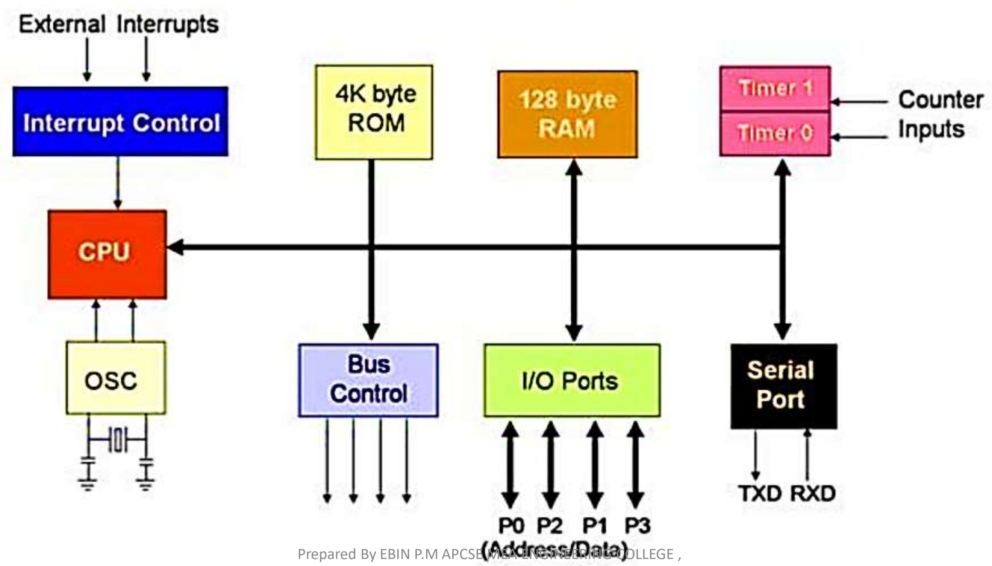
16. RAM : RAM is 128 byte memory for the read and write and is indirectly and directly addressable. A RAM address is between 0x00 and 0x7F

17. ROM: EPROM or flash EEPROM of 4KB or 8KB or 16KB

18. ALU: Perform 8 bit arithmetic and Logical operations over the operands held by the temporary registers TMP1 and TMP2.

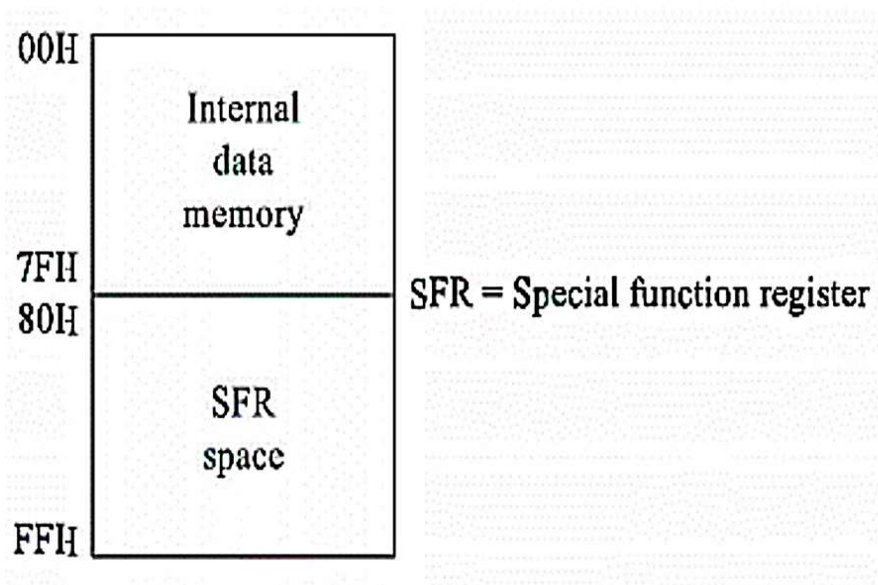
19. SFR Register Bank: 4 register banks each of 8 registers and these are also part of the internal RAM.

Block Diagram of 8051 Microcontroller



Register Organization of 8051

- The internal data memory of 8051 is divided into 2 groups
  - Set of eight registers (R0 to R7)
  - Scratch pad memory (High speed internal memory)
- The address range 00H to 07H is used to access the registers, and the rest are scratch pad memory
- 8051 provides 4 register banks, but only one register bank can be used at any point of time.
- To select the register bank, two bits of PSW are used.



- The following addressing can be used to select register bank

Address Range	Register Bank
00H to 07H	Register Bank 0
08H to 0FH	Register Bank 1
10H to 17H	Register Bank 2
18H to 1FH	Register Bank 3

- The concept of 4 register bank is very useful. For servicing the interrupts, this feature is very good
- The interrupt program can use one bank and the interrupt service subroutine can access another bank for better performance.

Special Function Registers (SFRs)

- Register A & B is used to store operands
- PSW – Processor Status Word
- IP- Interrupt Priority Register
- IE- Interrupt Enable Register
- SBUF - Serial Buffer
- SCON - Serial Control Register
- TH_1 - Timer1 higher 8 bits
- TH_0 - Timer 0 higher 8 bits
- TL_1 - Timer 1 lower 8 bits
- TL_0 - Timer0 lower 8 bits

- TMOD – Timer mode Register
- TCON – Timer Control Register
- PCON – Power Control Register
- SP - Stack Pointer
- P0 – Port 0
- P1 – Port1
- P2 – Port2
- P3 – Port3

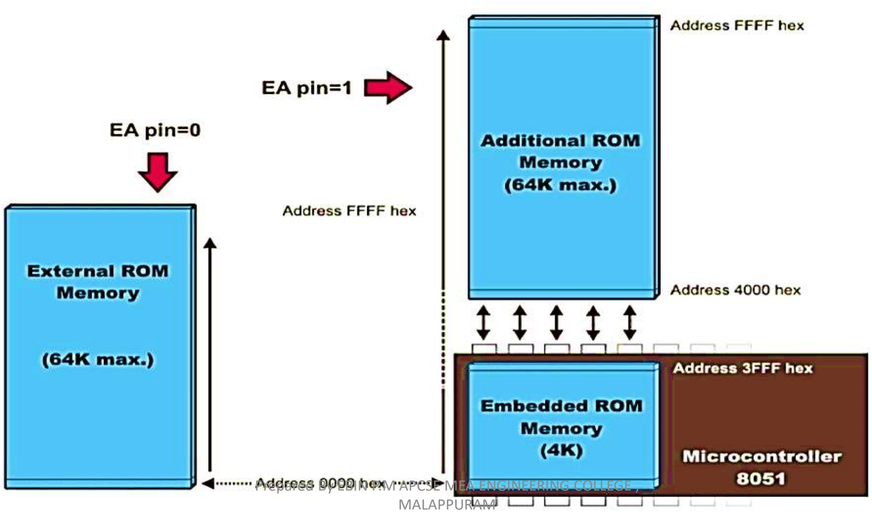
Memory Organization of 8051

➢In 8051 microcontroller , the memory is divided in to program memory and data memory.

Program Memory

- ROM is used for permanent saving program (CODE) being executed. The memory is read only
- Program memory may also used to store constant variables. The 8051 executes programs stored in program memory only
- 8051 memory organization allows external program memory to be added. Microcontroller handle external memory depends on the pin EA (External Access) logical state.

- External ROM is accessed whenever the $\overline{EA}$ pin is connected to ground (=0) or when the PC contains an address higher than the last address in the internal 4KB ROM (0FFFH). 8051 designs can use internal and external ROM automatically.



Internal Data Memory (RAM)

- The 128 byte internal RAM is organized into three distinct areas.

A) 32 bytes from address 00H to 1FH, that create 32 working registers organized as 4 banks of 8 registers each.

- The 4 register banks are numbered 0 to 3 and are made up of 8 registers named R_0 to R_7.
- Each register can be accessed by name or by its RAM address.

Eg: If bank 3 is currently selected, then R_0 of bank 3 is R_0 or address 18H

- Bits RS_0 and RS_1 in the PSW determine which bank of registers is currently in use at any time when the program is running.

➢Bank 0 is selected on reset.

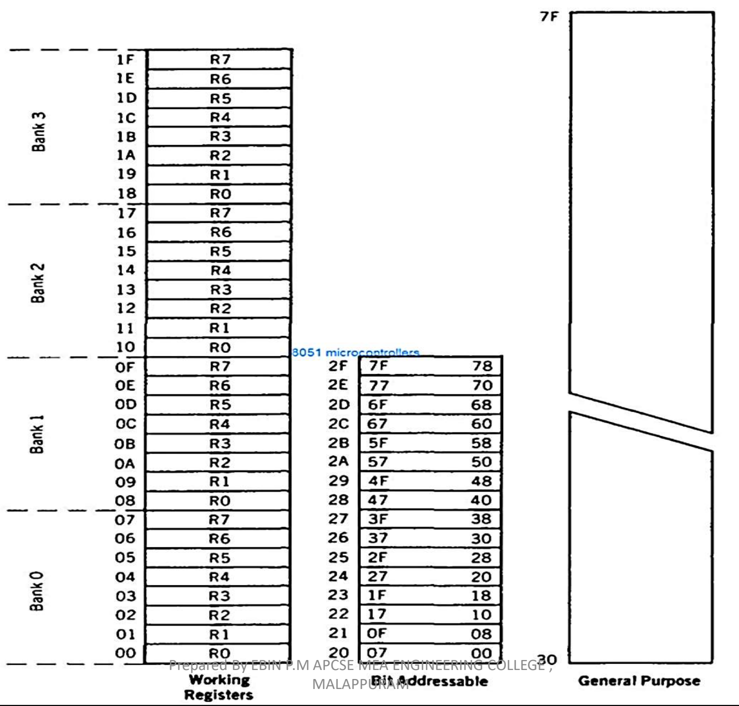
B) A bit-addressable area of 16 bytes occupies RAM byte addresses 20H to 2FH; forming a total of 128 addressable bits.

- An addressable bit maybe specified by its bit address of 00H to 7FH.

Eg: bit address 4FH is also bit 7 of byte address 29H

- Addressable bits are useful when the program need only remember a binary event.(switch on, light off, etc.)

c) A general purpose RAM area above the bit area rom 30H to 7FH, addressable as bytes.



External Data Memory

- Access to external memory is slower than access to internal data memory.
- There maybe up to 64K Bytes of external data memory. Several 8051 devices provide on-chip XRAM space.
- It can be accessed through the indirect addressing mode.

I/O addressing:

- 8051 microcontroller have 4 I/O ports each of 8-bit, which can be configured as input or output.
- Total 32 I/O pins allows the microcontroller to be connected with the peripheral devices.

Interrupts in 8051

- Interrupt is a signal send by an external device to the processor so as to request the processor to perform a particular task or work.
- When a processor recognizes an interrupt, it saves the processor status in stack. Then it call and execute interrupt service routine(ISR). At the end of ISR, it restores the processor status and the program control is transferred to main program.
- Types of interrupts are:

1) **Software & Hardware interrupt**
2) **Vectored & non-vectored interrupt**
3) **Maskable & non-maskable interrupt**

Types of Interrupt in 8051

➢ The 5 sources of interrupt in 8051 are:

1) Timer 0 overflow interrupt – TF0
2) Timer 1 overflow interrupt – TF1
3) External hardware interrupt – INT0
4) External hardware interrupt – INT1
5) Serial communication interrupt – RI/TI

There are two SFR's for interrupt handling

IE -> Interrupt Enable Register

IP -> Interrupt Priority Register

Interrupt Enable (IE) Register

- Responsible for enabling and disabling the interrupt.
- It is a bit addressable register.

7	6	5	4	3	2	1	0
EA	**X**	**X**	**ES**	**ET1**	**EX1**	**ET0**	**EX0**

- EX0 -> Enable/Disable – External Interrupt 0
- ET0 -> Enable/Disable – Timer 0 overflow interrupt
- EX1 -> Enable/Disable – External Interrupt 1
- ET1 -> Enable/Disable – Timer 1 overflow interrupt
- ES -> Enable/Disable – Serial port interrupt
- EA -> Enable/Disable – All interrupts

Interrupt Priority (IP) register

- It is possible to change the priority levels of the interrupts by setting or clearing the corresponding bit in the IP register.
- If the interrupt priorities are not programmed, the microcontroller executes in predefined manner in the order INTO, TF0, INT1, TF1, SI

TCON Register

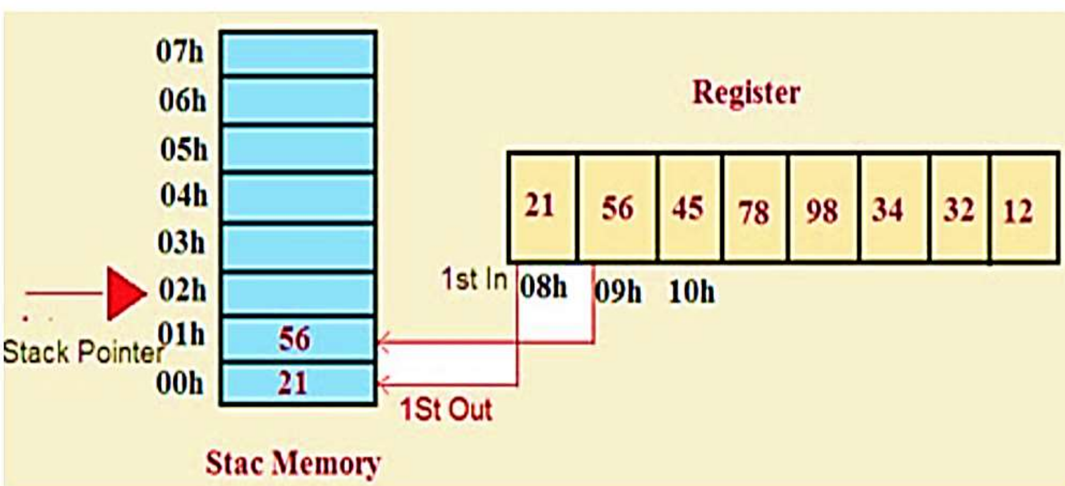
- TCON register specifies the type of external interrupt to the 8051 microcontroller.

INTERRUPT	ADDRESS
IE0	0003
TF0	000B
IE1	0013
TF1	001B
SERIAL	0023

STACK in 8051

- Stack is an area of RAM allocated to hold temporarily all the parameters of the variable
- Whenever the function is called , the parameters and local variables are added to the stack (PUSH)
- When the function returns, the parameter and the variables are removed(POP) from the stack.
- The register used to access the stack is called Stack Pointer register. The Stack Pointer (SP) is a small register used to point at the stack.
- When we push something in to the stack memory, the stack pointer increases.

PUSH Operation

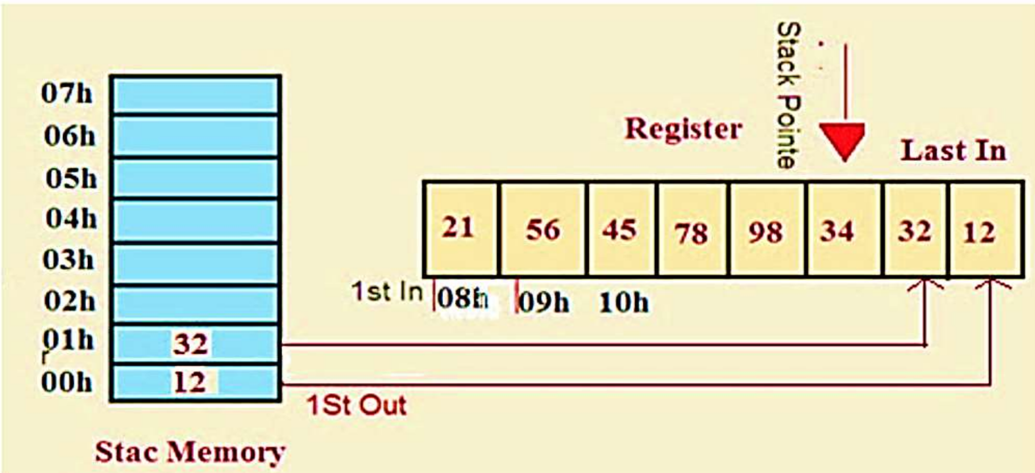


PUSH operation of Stack

- The PUSH is used for taking the values from any register and storing in the starting address of the stack pointer. ie; 00H by using PUSH operation
- For the next PUSH, it increments +1 , and stores the value in the next address of the stack pointer. ie, 01H
- Eg: 0000H

```
MOV 08H, #21H
MOV 09H, #56H
PUSH 00H
PUSH 01H
END
```


POP Operation



POP Operation in Stack

- It is used for placing the values from the stack pointer's maximum address to any other register's address
- If we use this POP again , then it decrements by 1 and the value stored in any register is given as POP
- Eg: 0000H

```
MOV 00H, #12H
MOV 01H, #32H
POP 1FH
POP 0EH
END
```