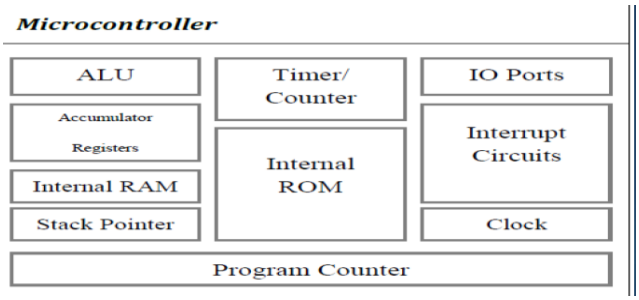
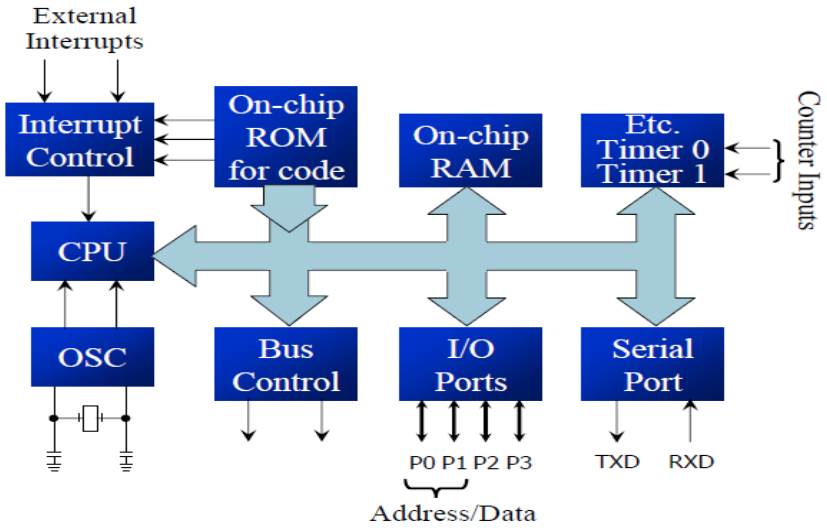
MODULE 5 MICROCONTROLLERS

- A microcontroller is a microcomputer with few other application-specific devices on a single chip or VLSI core.
- It has the specified computational capabilities as well as the enhanced IO operation and control capabilities.
- It is an integrated part in a real-time control or communication system.
- Henceforth, a microcontroller will also be referred to as MCU.
- An MCU consists of a microcomputer circuit or unit may have a provision to run in a power-down or idle mode.



MICROPROCESSORS	MICROCONTROLLERS
Microprocessor contains ALU, General purpose registers, stack pointer, program counter, clock timing circuit, interrupt circuit	Microcontroller contains the circuitry of microprocessor, and in addition it has built in ROM, RAM, I/O Devices, Timers/Counters etc.
It has many instructions to move data between memory and CPU	It has few instructions to move data between memory and CPU
Few bit handling instruction	It has many bit handling instructions
Less number of pins are multifunctional	More number of pins are multifunctional
Single memory map for data and code (program)	Separate memory map for data and code (program)
Access time for memory and IO are more	Less access time for built in memory and IO.
Microprocessor based system requires additional hardware	It requires less additional hardwares
More flexible in the design point of view	Less flexible since the additional circuits which is residing inside the microcontroller is fixed for a particular microcontroller
Large number of instructions with flexible addressing modes	Limited number of instructions with few addressing modes

ARCHITECTURE OF 8051



- In the architecture of 8051 the system bus is connected to all the supporting devices to CPU.
- The system bus consists of 8 bit data bus, 16 bit address bus and bus control signals.

- It also contains program memory, data memory, ports, serial interface, interrupt control, timer and CPU interfaced together to system bus.

CPU: The CPU is the brain of processing machine. It consists of following:

a. Accumulator: The accumulator register (ACC or A) acts as an operand register, in case of some instructions. The ACC register has been allotted an address in the on-chip special function register bank.

b. B Register: This register is used to store one of the operands for multiply and divide instructions. In other instructions, it may just be used as a scratch pad. This register is considered as a special function register.

c. Program Status Word (PSW) This set of flags contains the status information and is considered as one of the special function registers.

d. SP Stack pointer: this 8 bit wide register is incremented before the data is stored onto stack using push or call instructions. This register contains 8-bit stack top address. The stack may be defined anywhere in the on-chip 128-byte RAM. After reset, the SP register is initialized to 07. After each write to stack operation, the 8-bit contents of the operand are stored onto the stack, after incrementing the SP register by one.

e. Data Pointer (DTPR) This 16-bit register contains a higher byte (DPH) and the lower byte (DPL) of a 16 bit external RAM address. It is accessed as a 16 bit register or two 8 bit registers.

Interrupt controller: these provides a method to postpone or delay a current process perform a subroutine task and restart the standard program linking.

Oscillators: This circuit generates the basic timing clock signal for the operation of the circuit using crystal oscillator.

Memory:

- The RAM and RAM address blocks provide 128 bytes of RAM and its mechanism to address internally.
- It has a ROM of 4kB.

Bus control: bus is a group of wires which uses a common communication channel for data transfer. The different bus configurations include 8, 16 or more cables connected together. Busses are address bus and data bus.

Timers: These two 16-bit registers can be accessed as their lower and upper bytes. For example, TLO represents the lower byte of the timing register 0, while THO represents higher bytes of the timing register 0. Similarly, TL1 and TH1 represent lower and higher bytes of timing register 1. All these registers can be accessed using the four addresses allotted to them which lie in the special function registers SFR address range, i.e. 80 H to FF.

Input/ output ports: The i/p, o/p port provide a microcontroller a physical connection to outside world. It provides a gateway for passing data from the outside with the help of sensors. The output ports allow to control external devices.

Serial Ports: The serial data buffer internally contains two independent registers. One of them is a transmit buffer which is necessarily a parallel-in serial-out register. The other is called receive buffer which is a serial-in parallel-out register. Loading a byte to the transmit buffer initiates serial transmission of that byte. The serial data buffer is identifed as SBUF and is one of the special function registers.

SIGNAL DESCRIPTION OF 8051

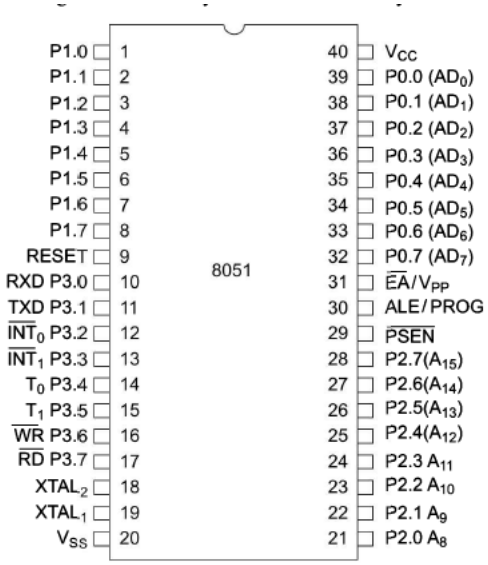


Fig. 17.3 *8051 Pin Configuration (Intel Corp.)*

- 8051 is available in 40 pin plastic and ceramic DIP packages.

Pins	Description
Vcc	+ 5V power supply.
Vss	Return pin for the supply.
RESET	Resets 8051, only when it goes high for two or more cycles.
ALE/PROG	The Address Latch Enable output pulse indicates that valid address is available on in their respective pins. Valid only for external memory accesses.
$\overline{EA}$/Vpp	External accessible pin, of tied low indicates that 8051 can address external program memory. For execution of internal memory programs this should be high.
$\overline{PSEN}$	Program store enable act as a strobe to read external program memory.
Port 0 (P0.0 – P0.7)	Port 0 is an 8 bit bidirectional bit addressable I/O port. It act as multiplexed address/data lines during external memory access. It receives code bytes during programming of internal EPROM.
Port 1(P1.0 – P1.7)	It act as 8 bit bidirectional bit addressable port.
Port 2 (P2.0 – P2.7)	Port 2 is an 8 bit bidirectional bit addressable I/O port. During external memory accesses, port 2 emits higher eight bits of address . P2 also receives higher order address bits during programming of the on-chip EPROM.
Port 3(P3.0 – P3.7)	**Table 17.2** *Alternate Functions of Pins of Port 3 (Intel Corp.)* *Port 3 Pin* — *Alternative Function* P3.0 — Acts as serial input data pin (RXD) P3.1 — Acts as serial output data pin (TXD) P3.2 — Acts as external interrupt pin 0 ($\overline{INT_0}$) P3.3 — Acts as external interrupt input pin 1 ($\overline{INT_1}$) P3.4 — Acts as external input to timer 0 (T0) P3.5 — Acts as external input to timer 1(T1) P3.6 — Acts as write control signal for external data memory ($\overline{WR}$) P3.7 — Acts as read control signal for external data memory read operation ($\overline{RD}$)
$XTAL_1$ and $XTAL_2$	There is an inbuilt oscillator which derives the necessary clock frequency for the operation of the controller. XTAL, is the input of amplifier and XTAL, is the output of the amplifier. A crystal is to be connected externally between these two pins to complete the feedback path to start oscillations.

REGISTER ORGANIZATION OF 8051

- 8051 has two 8-bit registers, registers A and B, which can be used to store operands, as allowed by the instruction set.

- Internal temporary registers of 8051 are not user accessible.

- Including these A and B registers, 8051 has a family of special purpose registers known as, Special Function Registers (SFRs).
- There are, in total, 21-bit addressable, 8-bit registers.
- ACC (A), B, PSW, PO, P1, P2, P3, IP, IE, TCON and SCON are all 8-bit, bit addressable registers.
- The remaining registers, namely, SP, DPH, DPL, TMOD, THO, TLO, TH1, TL1, SBUF and PCON registers are to be addressed as bytes, i.e. they are not bit-addressable.
- The registers DPH and DPL are the higher and lower bytes of a 16-bit register DPTR, i.e. data pointer, which is used for accessing external data memory.
- Starting 32-bytes of on-chip RAM may be used as general purpose registers. They have been allotted addresses in the range from 0000H to 001FH.
- These 32, 8-bit registers are divided into four groups of 8 registers each, called register banks. At a time only one of these four groups, i.e. banks can be accessed.
- The register bank to be accessed can be selected using the RS1 and RSO bits of an internal register called program status word.
- The registers THO and TLO form a 16-bit counter/timer register with H indicating the upper byte and L indicating the lower byte of the 16-bit timer register TO.
- Similarly, TH1 and TL1 form the 16-bit count for the timer T1.
- The four port latches are represented by PO, P1, P2 and P3. Any communication with these ports is established using the SFR addresses to these registers.

- Register SP is a stack pointer register.
- Register PSW is a flag register and contains status information.
- Register IP can be programmed to control the interrupt priority.
- Register IE can be programmed to control interrupts, i.e. enable or disable the interrupts.
- TCON is called timer/ counter control register. Some of the bits of this register are used to turn the timers on or off. This register also contains interrupt control flags for external interrupts INT; and INT.
- The register TMOD is used for programming the modes of operation of the timers/counters.
- The SCON register is a serial port mode control register and is used to control the operation of the serial port.
- The SBUF register acts as a serial data buffer for transmit and receive operations.
- The PCON register is called power control register. This register contains power down bit and idle bit which activate the power down mode and idle mode in 80C51BH. The PCON register also contains two general purpose flags and a double baud rate bit.

MEMORY AND I/O ADDRESSING

- The total memory of an 8051 system is logically divided into program memory and data memory.
- Program memory stores the programs to be executed, while data memory stores the data like intermediate results, variables and constants required for the execution of the program.

- Program memory is invariably implemented using EPROM, because it stores only program code which is to be executed and thus it need not be written into.
- However, the data memory may be read from or written to and thus it is implemented using RAM.

- Further, the program memory and data memory both may be categorized as on-chip (internal) and external memory, depending upon whether the memory physically exists on the chip or it is externally interfaced.

- The 8051 can address 4 Kbytes on-chip program memory whose map starts from 0000H and ends at 0FFFH.

- It can address 64 Kbytes of external program memory under the control of PSEN signal, whose address map is from 0000H to FFFFH.

- 8051 supports 64 Kbytes of external data memory whose map starts at 0000H and ends at FFFFH. This external data memory can be accessed under the control register DPTR, which stores the addresses for external data memory accesses.

- 8051 generates RD and WR signals during external data memory accesses. The chip select line of the external data memory may be derived from the address lines as in the case of other microprocessors.

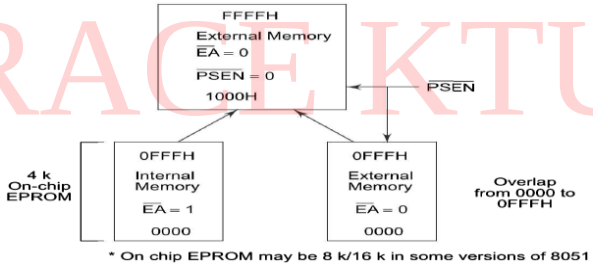


Fig. 17.5 *Program Memory Map of an 8051 System*

- Internal data memory of 8051 consists of two parts; the first is the RAM block of 128 bytes (256 bytes in case of some versions of 8051) and the second is the set of addresses from 80H to FFH, which includes the addresses allotted to the special function registers.

- The address map of the 8051 internal 128 bytes RAM starts from 00 and ends at 7FH. This RAM can be addressed by using direct or indirect mode of addressing.

- However, the special function register address map, i.e. from 80H to FFH is accessible only with direct addressing mode.

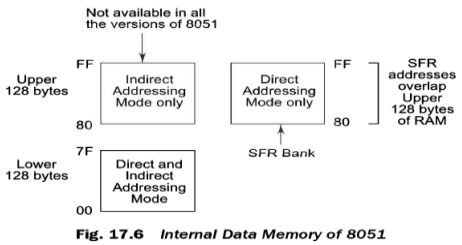


Fig. 17.6 *Internal Data Memory of 8051*

- The lower 128 bytes of RAM whose address map is from 00 to 7FH is functionally organised in three sections.

First Section:

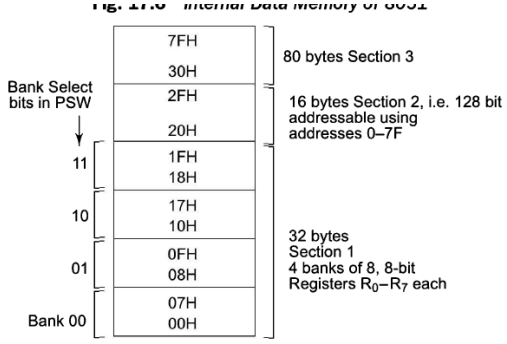
- The address block from 00 to 1FH, i.e. the lowest 32 bytes which form the first section, is divided into four banks of 8-bit registers, denoted as bank 00,01,10 and 11.
- Each of these banks contain eight 8-bit registers. The stack pointer gets initialized at address 07H, i.e. the last address of the bank 00, after reset operation.
- After reset bank 0 is selected by default but the actual stack data is stored from 08H onwards, i.e. bank 01, 10 and 11.
- These bank addressing bits of the register banks are present in PSW, to select one of these banks at a time.

Second Section:

- The second section extends from 20H to 2FH, i.e. 16 bytes, which is a bit-addressable block of memory, containing 16 x 8 = 128 bits.
- Each of these bits can be addressed using the addresses 00 to 7FH.
- Any of these bits can be accessed in two ways. In the first, its bit number is directly mentioned in the instruction while in the second the bit is mentioned with its position in the respective register byte. For example, the bits 0 to 7 can be referred directly by their numbers, i.e. 0 to 7 or using the notations 20.0 to 20.7 respectively.

Third Section:

- The third block of internal memory occupies addresses from 30H to 7FH.
- This block of memory is a byte addressable memory space.
- In general, this third block of memory is used as stack memory.
- All the internal data memory locations are accessed using 8-bit addresses under appropriate modes of addressing.



. 17.7 Functional Description of Internal Lower 128 Bytes of RAM

INTERRUPTS OF 8051

- 8051 provides five sources of interrupts.
- INT0 and INT, are the two external interrupt inputs.
- These can either be edge-sensitive or level-sensitive, as programmed with bits IT, and IT, in register TCON.
- These interrupts are processed internally by the flags IE, and IE.
- If the interrupts are programmed as edge-sensitive, these flags are automatically cleared after the control is transferred to the respective vector.

- On the other hand, if the interrupts are programmed level-sensitive, these flags are controlled by the external interrupts sources themselves.
- The timer 0 and timer 1 interrupt sources are generated by TF) and TF, bits of the register TCON, which are set, if a rollover takes place in their respective timer registers, except timer 0 in mode 3. When these interrupts are generated, the respective flags are automatically cleared after the control is transferred to the respective interrupt service routines.

- Serial Communication Interrupt (R1/T1): The serial port interrupt is generated, if at least one of the two bits RI and TI is set. Neither of the flags is cleared, after the control is transferred to the interrupt service routine. The RI and TI flags need to be cleared using software, after deciding, which one of these two caused the interrupt. This is accomplished in the interrupt service routine.

- In addition to these five interrupts, 8051 also allows single step interrupts to be generated with help of software. The external interrupts, if programmed level-sensitive, should remain high for at least two machine cycles for being sensed. If the external interrupts are programmed edge-sensitive, they should remain high for at least one machine cycle and low for at least one machine cycle, for being sensed.

Interrupt Enable Register

- This register is responsible for enabling or disabling the interrupt.
- Each interrupt can be enabled or disabled by setting a bit in the addressable register.
- The corresponding bit in the register enables the particular interrupts like timer, serial i/p and external i/p
- In the IE register, the bit corresponding to 1 activate the interrupt and 0 disables the interrupt.

EA	-	-	ES	ET1	EX1	ET0	EX0

i. EA – IE.7: It disables all the interrupts. If EA = 0 no interrupts will be acknowledged. If EA = 1 interrupt source is individually enabled or disabled.
ii. IE.6, IE.5: not implemented, reserved for future use.
iii. ES – IE.4 : enables or disables the serial port interrupt.
iv. ET1 – IE.3: Enable or disable timer 1 overflow interrupt
v. EX1 – IE.2: Enables or disables external interrupt 1
vi. ET0 – IE.1: enables or disables timer 0 overflow interrupt
vii. EX0 – IE.0: enable or disable external interrupt 0.

Interrupt Priority Register

- We can change the priority level of the interrupts by changing the corresponding bit in the interrupt priority register.
- Low priority interrupt can be interrupted by high interrupts but not vice versa.
- If two interrupt of different priorities are received simultaneously the request of higher level is served.
- If the request of same priority is received simultaneously, then the interrupt polling sequence determine which request is to be served first.

-	-	PT2	PS	PT1	PX1	PT0	PX0

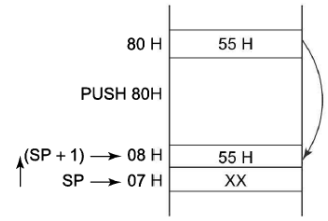
i. IP.6, IP.5 – Reserved for future use
ii. PS: IP.4 – Defines serial port priority level
iii. PT1: IP.3 – defines timer priority 1 level
iv. PX1: IP.2 – defines external interrupt priority.
v. PT0: IP.1 – defines timer 0 interrupt priority
vi. PX0: IP.0 – defines external interrupt of 0 priority

The following steps are taken to service the interrupts:

1. The current instruction of the main program is completed. If the current instruction is RETI or write to IE/IP register, the next one is also completed.
2. The content of the program counter pointing to the address of the next instruction of the main program to be executed after the interrupt service routine will be pushed on to the stack (data memory) at the current stack top address. The lower byte PCL is pushed first and PCH, the higher byte is pushed later.
3. The overflow flags are cleared if it is a timer interrupt. Corresponding IE0 and IE1 flags are cleared if it is an edge triggered (configured accordingly) interrupt.
4. The interrupt in progress for the priority level flip flop is set.
5. The control of execution is transferred to the interrupt service routine by generating a long call (LCALL). This is also called as vectoring of an interrupt.
6. The execution of the ISR starts. During execution of the ISR, low priority level or equal priority level interrupts are discarded. Only higher priority level interrupts will receive service.
7. At the end of the ISR, RETI instruction is executed. The RETI instruction clears the interrupt in progress flag. The address of the next instruction that was stored on to stack in step2 is popped back from the stack top and loaded into PC. The execution of the main program continues. Other lower or equal priority interrupts can be sensed and serviced further.

STACK STRUCTURE OF 8051

- 8051 stack operations are 8-bit wide i.e. in an operation using PUSH or POP instruction one byte of data is stored on to stack or retrieved from the stack.
- In case of internal 16 bit address push or pop to/from the stack, the operation is implemented byte by byte i.e. lower byte first followed by higher byte.
- The SP register is an 8-bit register and is initialized to internal RAM address 07H after reset.
- Obviously, the capabilities of stack in 8051 are limited compared to microprocessors.
- Fig. 17.9(a) shows operation of PUSH instruction.
- The SP registers points to stack top. The stack top is always assumed to be preoccupied. So the SP is incremented first.
- Then 8 bit content of the 8-bit address provided as operand is pushed on to the stack memory address available in SP



ig. 17.9 *(a) Storing into Stack Memory*

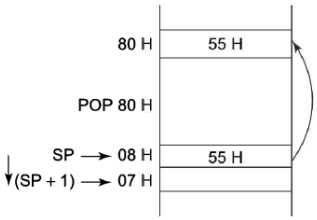


Fig. 17.9 (b) *Retrieving from Stack Memory*

- Thus the PUSH instruction has following two steps.

1. Increment stack by 1.
2. Store 8-bit content of the 8-bit address specified in the instruction to the address pointed to by SP.

- Complementarily, POP operation has the following two steps as shown in Fig. 17.9(b).
  1. 1. Store the content of top of stack pointed to by SP register to the 8 bit memory specified in the instruction.
  2. Decrement SP by 1.

TIMERS AND COUNTERS OF 8051

- It has two timer circuits timer 0 and timer 1
- These circuits can be used either as timer or counter and it is 16 bit wide.
- These 16 bit is accessed as 2 separate registers of lower byte and higher byte.

- these can be accessed like any other registers.

Timer 0 register

- It is a 16 bit register and can be accessed by low byte and high byte
- Low byte is referred as TL0 and higher byte as TH0

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Timer 1 register

- It is also a 16 bit register divided into two separate registers of lower and higher byte. Lower byte: TL1 and higher byte: TH1

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Timer Mode Register (TMOD)

- It is an 8 bit register used by both the timers timer 0 and timer 1 to set various timer modes
- In this TMOD register lower 4 bits are set for timer 0 and higher 4 bits are for timer 1
- In each cases lower 2 bits are used to set the timer mode and upper 2 bits are used to specify the operation.

GATE	C/T	M1	M0	GATE	C/T	M1	M0

- the upper or lower 4bit, the first bit is the gate very timer has a mean of starting and stopping, sometimes aid this with the help of software or some by means of hardware.
- The hardware way of starting and stopping the timer is by an external source is achieved by making GATE = 1.
- The second bit C/T is used to decide whether the timer is used as a time delay generator or an event counter. If this bit is 0 it is used as timer and if it is 1 then used a counter.

Timer Counter Register (TCON)

- This register specifies the type of external interrupt to 8051 micro controller.
- Two external interrupt whether it is an edge or level triggered specify by this register by a set or cleared by appropriate bits in it.
- It is also an addressable register.

TF1	TR1	TF0	TR0	IE1	IE0	IT0

- TF1 – TCON7 – Timer 1 overflow flag
- TR1 – TCON6 – Timer 1 control bit
- TF0 – TCON5 – Timer 0 overflow flag
- TR0 – TCON4 – Timer 0 control bit
- IE1 – TCON3 – External interrupt 1 edge
- IT1 – TCON2 – Interrupt 1 control bit
- IE0 – TCON1 – External interrupt 0 edge
- IT0 – TCON0 – Interrupt 0 type control bit

Modes of operation:

M0	M1	Mode	Operating modes
0	0	0	13 bit timer mode, 8 bit timer/ counter TH_x and TL_x as 5 bit pre scale register
0	1	1	16 bit timer, 16 bit timer/ counter TH_x and TL_x are cascaded no pre scalable
1	0	2	8 bit auto reload mode, 8 bit auto timer/ counter. THx hold value is to be 1 reloaded to TLx, each time overflow occurs
1	1	3	Split timer mode

Mode 1:

- Mode 1 is 1 16 bit timer and its value starts from location 0000H to FFFFH to be loaded in the timer register T_L and T_H. only after loading the timer will start
- It is done by giving the instruction SET B TR0 for timer 0 and SET B TR1 for timer 1
- Once the timer is started, it will count upto FFFFH
- When it roll over from FFFFH to 0000H a timer flag is set.
- When this timer flag is set, the timer can be stopped by giving instruction CLR TR0 for timer 0 and CLR TR1 for timer 1.
- When the timer reaches its limit in order to repeat the process the register must be reloaded with the original value and Tf must be reset to 0

Mode 0:

- It is exactly same as mode 1 except that in this 1 13 bit timer is used instead of 16 bit.
- It can hold values between 0000H to 1FFFH therefore when the timer reaches maximum of 1FFFH and it rolls over to 0000H and Tf is in raised condition.

Mode 2:

- It is an 8 bit timer that allows the value from 00 to FF to be loaded in the timer register TH.
- Once it is loaded with the 8 bit value, it gives a copy of it to TL register.
- After the timer is started it starts counting by incrementing the TL register.
- Once it reaches its limit FFH, it rolls to 00H, the timer flag is set.
- To repeat the process, we must clear the TF register and reload it to the original value.

Mode 3:

- It is also known as Split Timer Mode
- Timer 0 and timer 1 may be programmed to be in mode 0, mode 1, mode 2 independently of similar mode for other timer.

ADDRESSING MODES OF 8051

- 8051 instruction set supports six addressing modes as listed:

1. Immediate Addressing
2. Register Addressing
3. Direct addressing
4. Register indirect addressing
5. Indexed addressing
6. Register specific

- Immediate Addressing: In this mode, an immediate data, i.e. a constant is specified in the instruction, after the opcode byte. E.g., ADD A, #100

- Register Instructions: In this addressing mode, operands are stored in the registers Ry—R, of the selected register bank. One of these eight registers (Ry—R7) is specified in the instruction using the 3-bit register specification field of the opcode format. E.g. MOV A, R5

- Direct Addressing: In this mode of addressing, the operands are specified using the 8-bit address field, in the instruction format. Only internal data RAM and SFRS can be directly addressed. Example MOV RO, 89H.

- Indirect Addressing _In this mode of addressing, the 8-bit address of an operand is stored in a register and the register, instead of the 8-bit address, is specified in the instruction. The registers Rg and R, of the selected bank of registers or stack pointer can be used as address registers for storing the 8-bit addresses. The address register for 16-bit addresses can only be ‘data pointer’ (DPTR). Example ADD A, @ RO

- Indexed Addressing: with this mode the effective address of the operand is the sum of base register and the offset register. The base register can be either a data pointer of program counter with offset address as accumulator. e.g., MOV A, @A + DTPR

- Register Specific Instructions: In this type of instructions, the operand is implicitly specified using one of the registers. Some of the instructions always operate only on a specific register. These type of instructions fall under this category. E.g. RLA rotates accumulator left.