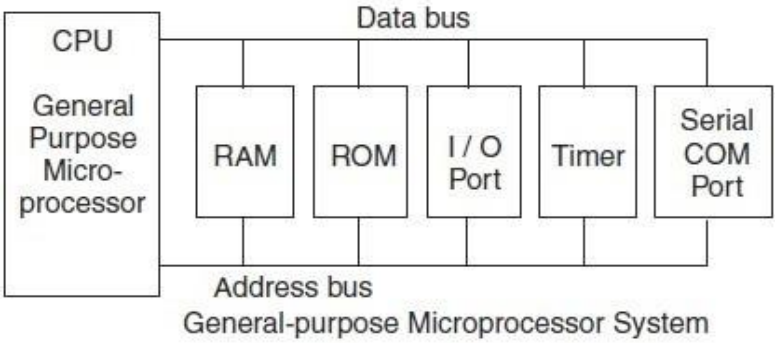
UNIT-4
INTEL 8051 MICROCONTROLLER

- Microprocessor vs Microcontroller,
- 8051 Microcontroller Architecture,
- 8051 pin diagram,
- 8051 Ports,
- Alternate functions of I/O pins,
- Memory organization,
- Internal RAM structure,
- Stack operation,
- Counters and Timers,
- Serial Communication in 8051,
- Interrupts in 8051.

INTRODUCTION

The microprocessor is a programmable chip that forms the CPU of a computer. Nowadays, many microprocessor chips are available in the market for users to select from depending on the application. In general, **processor chips** can be classified as general-purpose microprocessors, microcontrollers, and DSP processors

A general-purpose microprocessor is the CPU of a **digital computer** and needs external components such as memory, input devices, output devices, and decoders to function as a **microcomputer system**.



General-purpose Microprocessor System

Examples of **8-bit processors** are

- **Intel's 8085, Zilog 80, and Motorola 6800.**

Examples of **16-bit processors** are

- **Intel's** 8086 and 8088 and **Motorola's** 68000 and

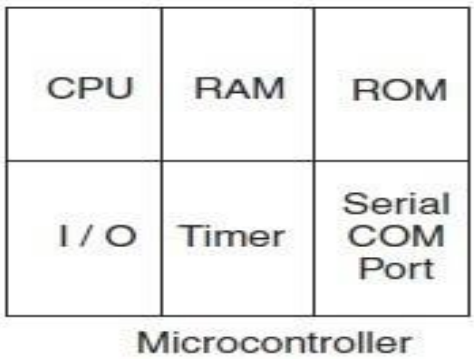
Examples of **32-bit processors** are

- **Intel's** 80186, 80286, and 80386, and **Motorola's** 68030.

In general, these **microprocessor-based systems** get **data from mass storage devices**, perform calculations, and store the results in **storage devices**. **General-purpose microprocessors** use **external memory** and a lot of **processor time** is involved in **data transfer** between the external memory and the processor.

MICROCONTROLLERS:

Microcontrollers are processor chips that generally have **memory, input ports, and output ports within the chip itself.** Therefore, they can also be called **single-chip computers**, **computer-on-a-chip**, or **system-on-a-chip**.



Microcontroller

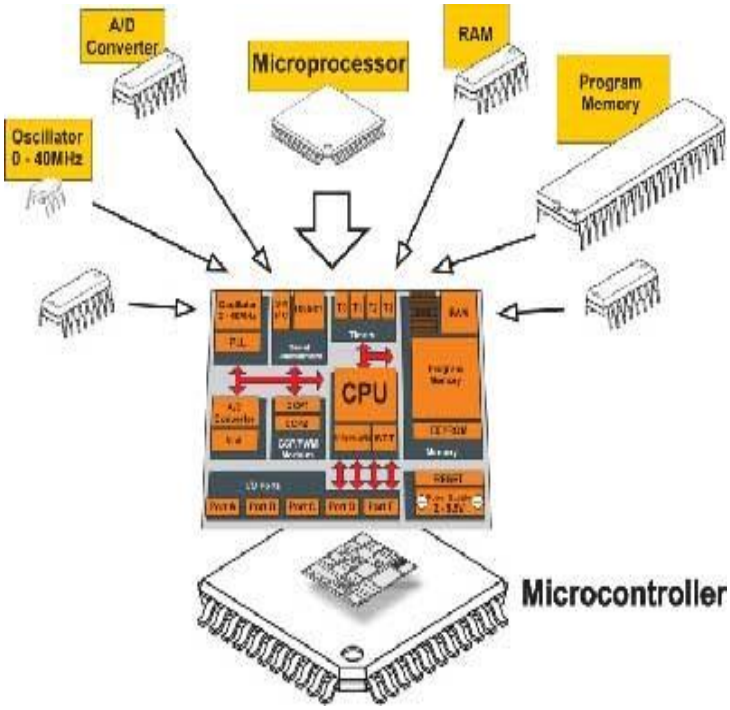
Microcontrollers are used in **machine control applications**, where there is **no need to change** the **program. Equipments** that use **microcontrollers** include computer printers, plotters, fax machines, Xerox machines,

telephones, automotive engine control mechanisms, and electronic instruments such as oscilloscopes, multi meters, planimeters, IC testers, etc.

Why the name Micro controller?

Basically **used for control applications**. It is used to control the **operation of machine** using **fixed program** that is stored in **ROM/EPROM** and that does not change **over the life time**.

The microcontroller incorporates all the features that are found in microprocessor. The microcontroller has built in ROM, RAM, Input Output ports, Serial Port, timers, interrupts and clock circuit. A microcontroller is an entire computer manufactured on a single chip. Microcontrollers are usually dedicated devices embedded within an application.



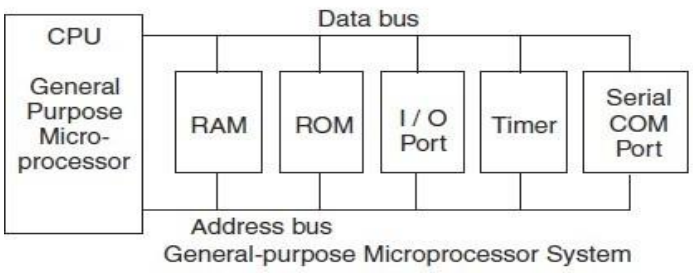
ADVANTAGES OF MICROCONTROLLER:

- Low cost
- Small size of product
- Easy to troubleshoot and maintain
- More reliable
- Additional memory, I/o can also be added
- Software security feature
- All features available with 40 pins.
- Useful for small dedicated applications and not for larger system designs which may require many more I/O ports.
- Mostly used to implement small control functions.

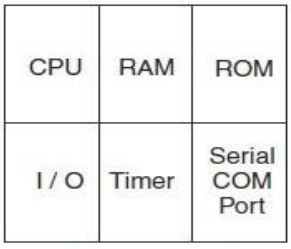
MICROPROCESSOR VS MICROCONTROLLER

It is very clear from figure that in microprocessor we have to interface additional circuitry for providing the function of memory and ports, for example we have to interface external RAM for data storage, ROM for program storage, programmable peripheral interface (PPI) 8255 for the Input Output ports, 8253 for timers, USART for serial port. While in the microcontroller RAM, ROM, I/O ports, timers and serial communication ports are in built. Because of this it is called as "system on chip". So in micro-controller there is no necessity of additional circuitry which is interfaced in the microprocessor because memory and input output ports are inbuilt in the microcontroller.

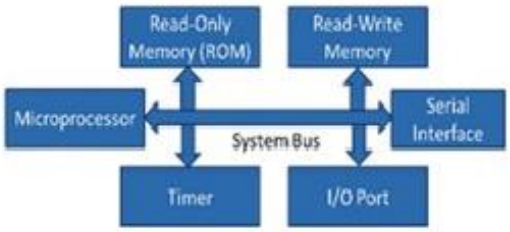
Microcontroller gives the satisfactory performance for small applications. But for large applications the memory requirement is limited because only 64 KB memory is available for program storage. So for large applications we prefer microprocessor than microcontroller due to its high processing speed.



General-purpose Microprocessor System



Microcontroller

Microprocessor	Micro Controller
	
Microprocessor is heart of Computer system.	Micro Controller is a heart of embedded system.
It is just a processor. Memory and I/O components have to be connected externally	Micro controller has external processor along with internal memory and i/O components
Since memory and I/O has to be connected externally, the circuit becomes large.	Since memory and I/O are present internally, the circuit is small.
Cannot be used in compact systems and hence inefficient	Can be used in compact systems and hence it is an efficient technique
Cost of the entire system increases	Cost of the entire system is low
Due to external components, the entire power consumption is high. Hence it is not suitable to used with devices running on stored power like batteries.	Since external components are low, total power consumption is less and can be used with devices running on stored power like batteries.
Most of the microprocessors do not have power saving features.	Most of the micro controllers have power saving modes like idle mode and power saving mode. This helps to reduce power consumption even further.
Since memory and I/O components are all external, each instruction will need external operation, hence it is relatively slower.	Since components are internal, most of the operations are internal instruction, hence speed is fast.
Microprocessor have less number of registers, hence more operations are memory based.	Micro controller have more number of registers, hence the programs are easier to write.
Microprocessors are based on von Neumann model/architecture where program and data are stored in same memory module	Micro controllers are based on Harvard architecture where program memory and Data memory are separate
Mainly used in personal computers	Used mainly in washing machine, MP3 players

Major **difference** is that **microcontrollers** are comparatively **faster** because of **reduced external memory accessing**.
Intel's 8031, 8051, and 8096 and Motorola's 68HC11 are examples of microcontrollers.

INTEL'S MCS 51 SERIES MICROCONTROLLERS:

Intel Corporation has many micro-controllers in both **8 bit and 16 bit** configuration. The **8 bit micro-controllers** -in many part numbers -MCS – 51 as the family name. All the microcontroller chips listed below have the same basic architecture.

- Intel Corporation has many micro-controllers in both 8 bit and 16 bit configuration.
- For example, 8XC51RD comes with an internal ROM of 64 KB, while 8XC51FC comes with only a 32 KB ROM.

Intel's 8 bit Micro-controllers:

Device Number	Data bus width	RAM capacity	ROM capacity
8031	8	128 bytes	Nil
8051	8	128 bytes	4Kbytes
8751H	8	128 bytes	4Kbytes EPROM
8052AH	8	256 bytes	8Kbytes
8752BH	8	256 bytes	8Kbytes EPROM

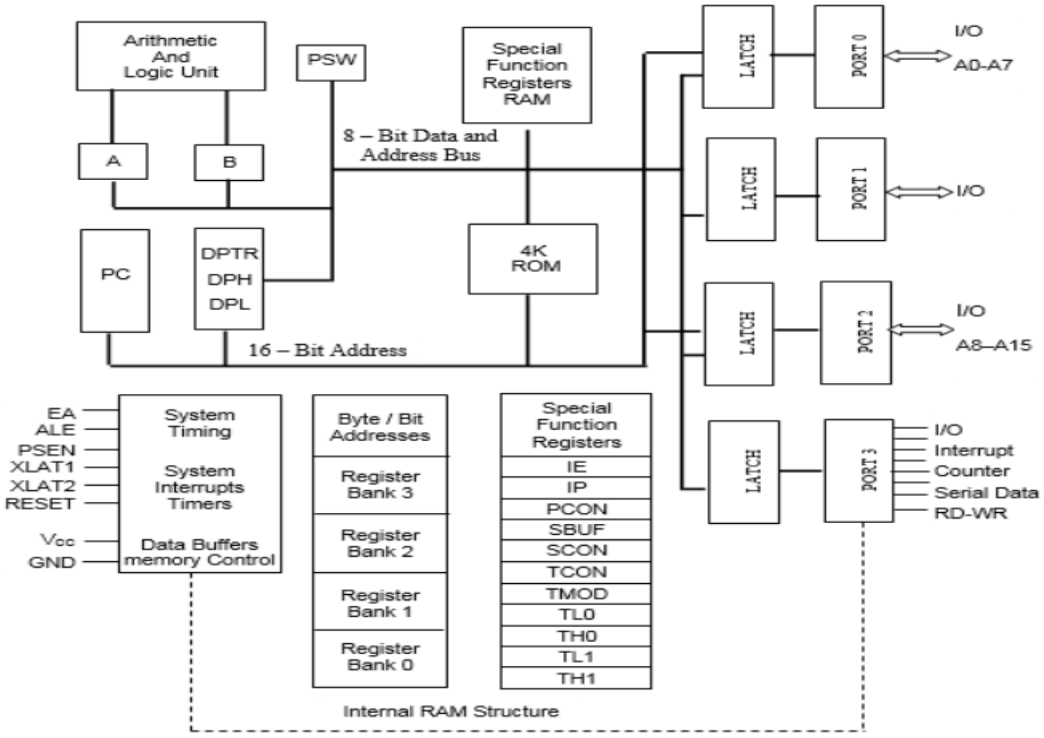
INTEL 8051 MICROCONTROLLER:

The **Intel MCS-51** (commonly referred to as **8051**) is a Harvard architecture, single chip microcontroller (µC) series which was developed by Intel in 1980 for use in embedded systems. Intel's original versions were popular in the 1980s and early 1990s. Intel's original MCS-51 family was developed using NMOS technology, but later versions, identified by a letter C in their name (e.g., 80C51) used CMOS technology and consume less power than their NMOS predecessors. This made them more suitable for battery-powered devices. The 8051 architecture provides many functions (CPU, RAM, ROM, I/O, interrupt logic, timer, etc.) in a single package.

The main features -8051chips are

- 8 bit CPU, i.e, 8-bit ALU, 8-bit Registers and 8-bit data bus
- 4Kbytes of on chip Program memory,
- 128 bytes of on chip data RAM,
- 4 ports of 8bit each (P0, P1, P3 & P4),
- Two 16 bit timers (T0 & T1),
- Full duplex serial port and
- On-chip clock oscillator.
- the 8051 provide Boolean processing;
- Two external and three internal vectored interrupts

8051 ARCHITECTURE:



Architecture Description:

- **8051 is an 8 bit micro-controller *i.e.* data bus within and outside the chip is 8 bits wide.**
- **The address bus of the 8051 is 16 bits wide, so it can address 64Kbytes of memory.**
- **The port0 and port2 pins of 8051 forms the multiplexed address and data bus.**

Accumulator (Acc): • Operand register
• Implicit or specified in the instruction
• Has an address in on chip SFR bank

B Register: *Used* to store one of the operands for multiplication and division, otherwise, scratch pad considered as a SFR.

Program Status Word (PSW): Set of flags contains status information.

Stack Pointer (SP): 8 bit wide register. Incremented before data is stored on to the stack using PUSH or CALL instructions. Stack defined anywhere on the 128 byte RAM

Data Pointer (DPTR): *1*6 bit register contains DPH and DPL Pointer to external RAM address. DPH and DPL allotted separate addresses in SFR bank

Port 0 To 3 Latches & Drivers: Each I/O port allotted a latch and a driver Latches allotted address in SFR. User can communicate via these ports P0, P1, P2, and P3.

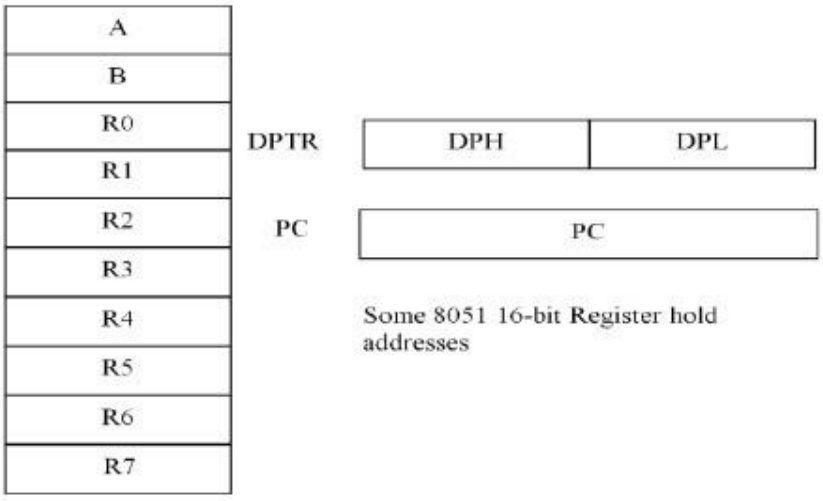
Serial Data Buffer: *I*nternally had TWO independent registers, TRANSMIT buffer (parallel in serial out – PISO) and RECEIVE buffer (serial in parallel out –SIPO) identified by SBUF and allotted an address in SFR.

Timer Registers: for Timer0 (16 bit register – TL0 & TH0) and for Timer1 (16 bit register – TL1 & TH1) four addresses allotted in SFR

Control Registers: Control registers are IP, IE, TMOD, TCON, SCON, and PCON. These registers contain control and status information for interrupts, timers/counters and serial port. Allotted separate address in SFR.

Timing and Control Unit: This unit derives necessary timing and control signals for internal circuit and external system bus
Oscillator: generates basic timing clock signal using crystal oscillator.
Instruction Register: decodes the opcode and gives information to timing and control unit.
EPROM & program address Register: provide on chip EPROM and mechanism to address it. All versions don't have EPROM.
RAM & RAM Address Register: provide internal 128 bytes RAM and a mechanism to address internally
ALU: Performs 8 bit arithmetic and logical operations over the operands held by TEMP1 and TEMP 2.User cannot access temporary registers.
SFR Register Bank: set of special function registers address range: 80 H to FF H. Interrupt, serial port and timer units control and perform specific functions under the control of timing and control unit

REGISTER SET OF 8051



Some 8-bitt Registers of the 8051

Accumulator ACC is the Accumulator register. The mnemonics for accumulator-specific instructions, however, refer to the accumulator simply as A.
B Register The B register is used during multiply and divide operations. For other instructions it can be treated as another scratch pad register.
Program Status Word The PSW register contains program status information as detailed in Table below **Table.** PSW: Program Status Word Register

CY	AC	F0	RS1	RS0	OV	–	P

CY	PSW.7	Carry flag.
AC	PSW.6	Auxiliary carry flag.
F0	PSW.5	Available to the user for general purpose.
RS1	PSW.4	Register Bank selector bit 1.
RS0	PSW.3	Register Bank selector bit 0.
OV	PSW.2	Overflow flag.
--	PSW.1	User-definable bit.
P	PSW.0	Parity flag. Set/cleared by hardware each instuction cycle to indicate an odd/even number of 1 bits in the accumulator.

RS1	RS0	Register Bank	Address
0	0	0	00H - 07H
0	1	1	08H - 0FH
1	0	2	10H - 17H
1	1	3	18H - 1FH

Stack Pointer The Stack Pointer register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H.

Data Pointer The Data Pointer (DPTR) consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16-bit address. It may be manipulated as a 16-bit register or as two independent 8-bit registers.

Ports 0 to 3 P0, P1, P2 and P3 are the SFR latches of Ports 0, 1, 2 and 3, respectively.

Serial Data Buffer The Serial Data Buffer is actually two separate registers, a transmit buffer and a receive buffer register. When data is moved to SBUF, it goes to the transmit buffer where it is held for serial transmission. (Moving a byte to SBUF is what initiates the transmission.) When data is moved from SBUF, it comes from the receive buffer.

Timer Registers Register pairs (TH0, TL0), (TH1, TL1), and (TH2, TL2) are the 16-bit counting registers for Timer/Counters 0, 1, and 2, respectively.

Control Registers: Special Function Registers IP, IE, TMOD, TCON, T2CON, SCON, and PCON contain control and status bits for the interrupt system, the timer/counters, and the serial port.

Special Function Registers (SFR) :

The set of Special Function Registers (SFRs) contains important registers such as Accumulator, Register B, I/O Port latch registers, Stack pointer, Data Pointer, Processor Status Word (PSW) and various control registers. Some of these registers are bit addressable. Addresses from 80H to FFH of all Special Function Registers

• PSW, P0-P3, IP, IE, TCON,SCON
– Bit addressable, 8bit each, 11 in number
• SP, DPH,DPL,TMOD,TH0,TL0,TH1,TL1,SBUF,PCON
– Byte addressable, 8bit each.
– DPTR – data pointer, accesses external memory. DPH + DPL = DPTR
• Starting 32 bytes of RAM – general purpose registers, divided into 4 register banks of 8 registers each. Only one of these banks accessible at one time. RS1 and RS0 of PSW used to select bank.
• TH0-TL0 and TH1-TL1
– 16 bit timer registers
• P0-P3 – port latches
• SP, PSW, IP – Interrupt Priority, IE – enable
• TCON – timer/counter control register to turn on/off the timers, interrupt control flags for external interrupts like INT1 and INT0
• TMOD – modes of operation of timer/counter
• SCON – serial port mode control register
• SBUF – serial data buffer for transmit and receive
• PCON – Power control register – power down bit, idle bit

Table 17.3 *SFR Registers, their Addresses and Contents after Reset*

Register	*Bit Addressable*	*Address (SFR)*	*Content After Reset*
ACC	Y	0E0H	0000 0000
B	Y	0F0H	0000 0000
PSW	Y	0D0H	0000 0000
SP	N	81H	0000 0111
DPH	N	82H	0000 0000
DPL	N	83H	0000 0000
P0	Y	80H	1111 1111
P1	Y	90H	1111 1111
P2	Y	0A0H	1111 1111
P3	Y	0B0H	1111 1111
IP	Y	0B8H	XX0 0000
IE	Y	0A8H	0XX0 0000
TMOD	N	89H	0000 0000
TCON	Y	88H	0000 0000
TH0	N	8CH	0000 0000
TL0	N	8AH	0000 0000
TH1	N	8DH	0000 0000
TL1	N	8BH	0000 0000
SCON	Y	98H	0000 0000
SBUF	N	99H	Indeterminate
PCON	N	87H	HMOS 0XXX XXXX
			CHMOS 0XXX 0000

8051 INSTRUCTION CYCLE:

- In 8051, **one instruction cycle** consists of **twelve (12)** clock cycles.
- **Instruction cycle** is sometimes called as **Machine cycle** by some authors.

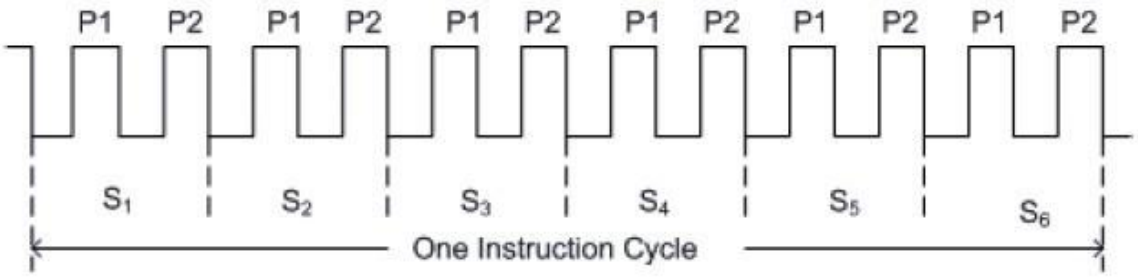


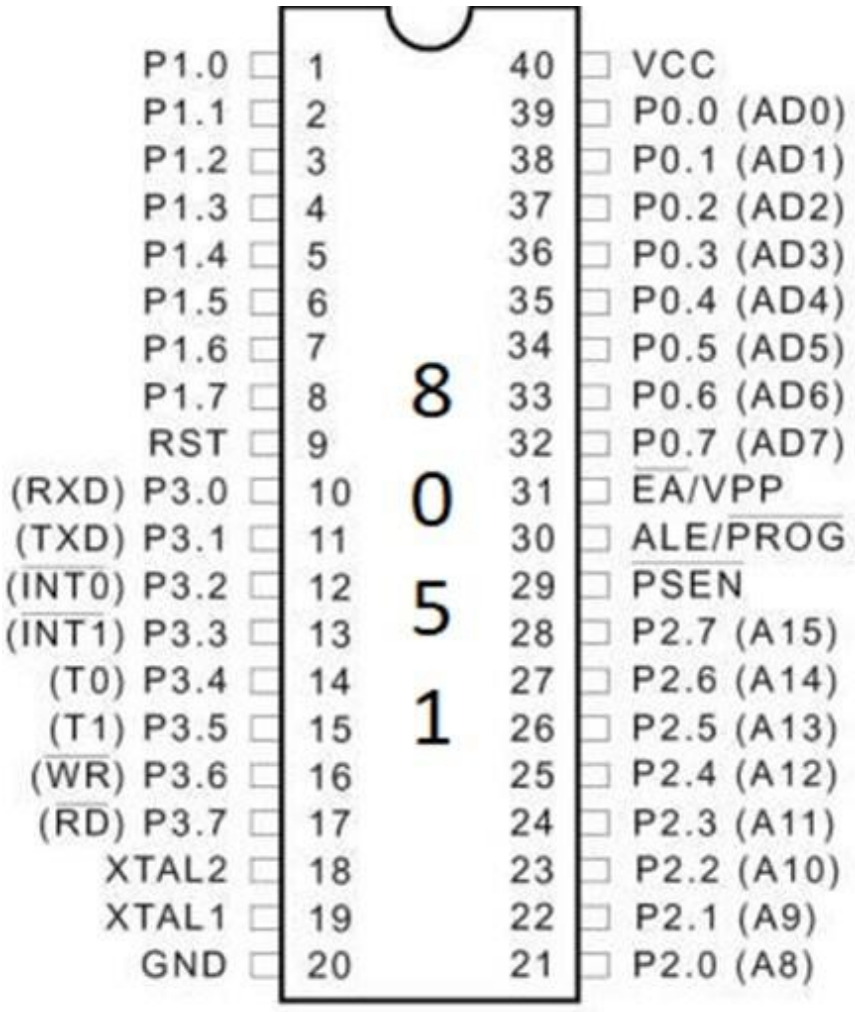
Fig 5.2 : Instruction cycle of 8051

In 8051, each instruction cycle has six states (S1 - S6). Each state has two pulses (P1 and P2)

PIN DETAILS OF 8051:

The 8051 is a 40-pin chip. The power supply +VCC and VSS takes two pins and the built-in clock oscillator requires two pins (–XTAL1 and XTAL2) for connecting the crystal.

- The four control signals pins of 8051 are PSEN, ALE, EA and RST. RST is an active high reset signal to restart the controller chip.
- 8051 responds to a **RST high input only** if the RST is held high for at **least two machine cycles**.
- In 8051, the **maximum number of clock cycles** taken for a **machine cycle is 12**.
- So, the RST pin must be high for at least **24 clock periods**.
- PSEN, ALE, EA are the signals used in conjunction with the **external memory access** of the 8051



Pin Diagram of 8051 microcontroller

Pin out Description:

Pins 1-8: Port 1 Each of these pins can be configured as an input or an output.

Pin 9: **RST** A logic one on this pin disables the microcontroller and clears the contents of most registers. In other words, the positive voltage on this pin resets the microcontroller. By applying logic zero to this pin, the program starts execution from the beginning.

Pins10-17: Port 3 Similar to port 1, each of these pins can serve as general input or output. Besides, all of them have **alternative functions:**

Pin 10: RXD Serial asynchronous communication input or Serial synchronous communication output.

Pin 11: TXD Serial asynchronous communication output or Serial synchronous communication clock output.

Pin 12: INT0 Interrupt 0 input.

Pin 13: INT1 Interrupt 1 input.

Pin 14: T0 Counter 0 clock input.

Pin 15: T1 Counter 1 clock input.

Pin 16: WR Write to external (additional) RAM.

Pin 17: RD Read from external RAM.

Port 3 Alternate Functions

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

Pin 18, 19: X2, X1 Internal oscillator input and output. A quartz crystal which specifies operating frequency is usually connected to these pins. Instead of it, miniature ceramics resonators can also be used for frequency stability. Later versions of microcontrollers operate at a frequency of 0 Hz up to over 50 Hz.

Pin 20: GND Ground.

Pin 21-28: Port 2 If there is no intention to use external memory then these port pins are configured as general inputs/outputs. In case external memory is used, the higher address byte, i.e. addresses A8-A15 will appear on this port. Even though memory with capacity of 64Kb is not used, which means that not all eight port bits are used for its addressing, the rest of them are not available as inputs/outputs.

Pin 29: PSEN If external ROM is used for storing program then a logic zero (0) appears on it every time the microcontroller reads a byte from memory.

Pin 30: ALE Prior to reading from external memory, the microcontroller puts the lower address byte (A0-A7) on P0 and activates the **ALE output**. After receiving signal from the **ALE pin**, uses **P0** as a memory chip address. Immediately after that, the ALE pin is returned its previous **logic state** and **P0 is now used as a Data Bus**. P0 port is used for both **data and address** transmission.

Pin 31: EA(External Access) By applying logic zero to this pin, P2 and P3 are used for data and address transmission with no regard to whether there is internal memory or not. It means that even there is a program written to the microcontroller, it will not be executed. Instead, the program written to external ROM will be executed. By applying logic one to the EA pin, the microcontroller will use both memories, first internal then external (if exists).

Pin 32-39: Port 0 Similar to P2, if external memory is not used, these pins can be used as general inputs/outputs. Otherwise, P0 is configured as address output (A0-A7) when the ALE pin is driven high (1) or as data output (Data Bus) when the ALE pin is driven low (0).

Pin 40: VCC +5V power supply.

Memory organization of 8051:

- In the 8051, the memory is organized logically into **program memory** and **data memory** separately.
- The **program memory** is **read-only type (ROM)**; the **data memory** is organized as **read–write memory (RAM)**.
- Again, both **program and data memories** can be **within** the chip or **outside**.
- The Intel 8051 has 128 bytes of RAM and 4 KB of ROM within the chip.
- The address bus of the 8051 is 16 bits wide. So it can access 64 KB of memory.

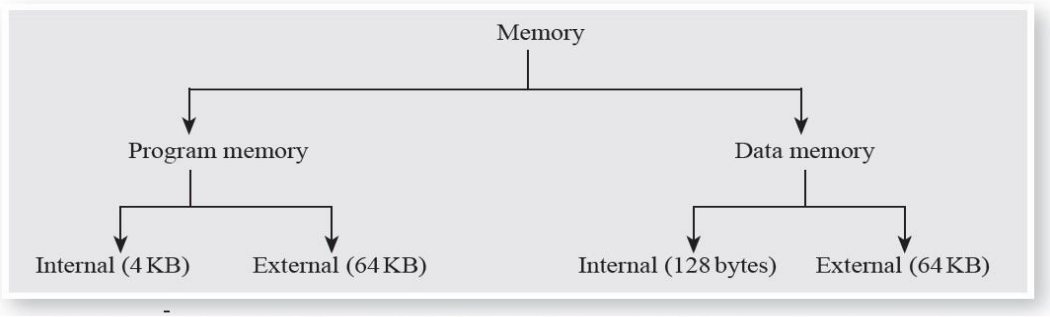
Program memory – EPROM:

- Intermediate results, variables, const

– 4KB internal from 0000 – 0FFFH
– 64KB external with PSEN, till FFFFH
– Internal –external difference PSEN

Data Memory – RAM:

– 64KB of external with DPTR signal
– Internal memory two parts - 128 bytes Internal RAM and secondly set of addresses from 80-FFH for SFR's
– 128 bytes from 00 – 7FH direct or indirect
–SFR addresses – only direct addressing mode



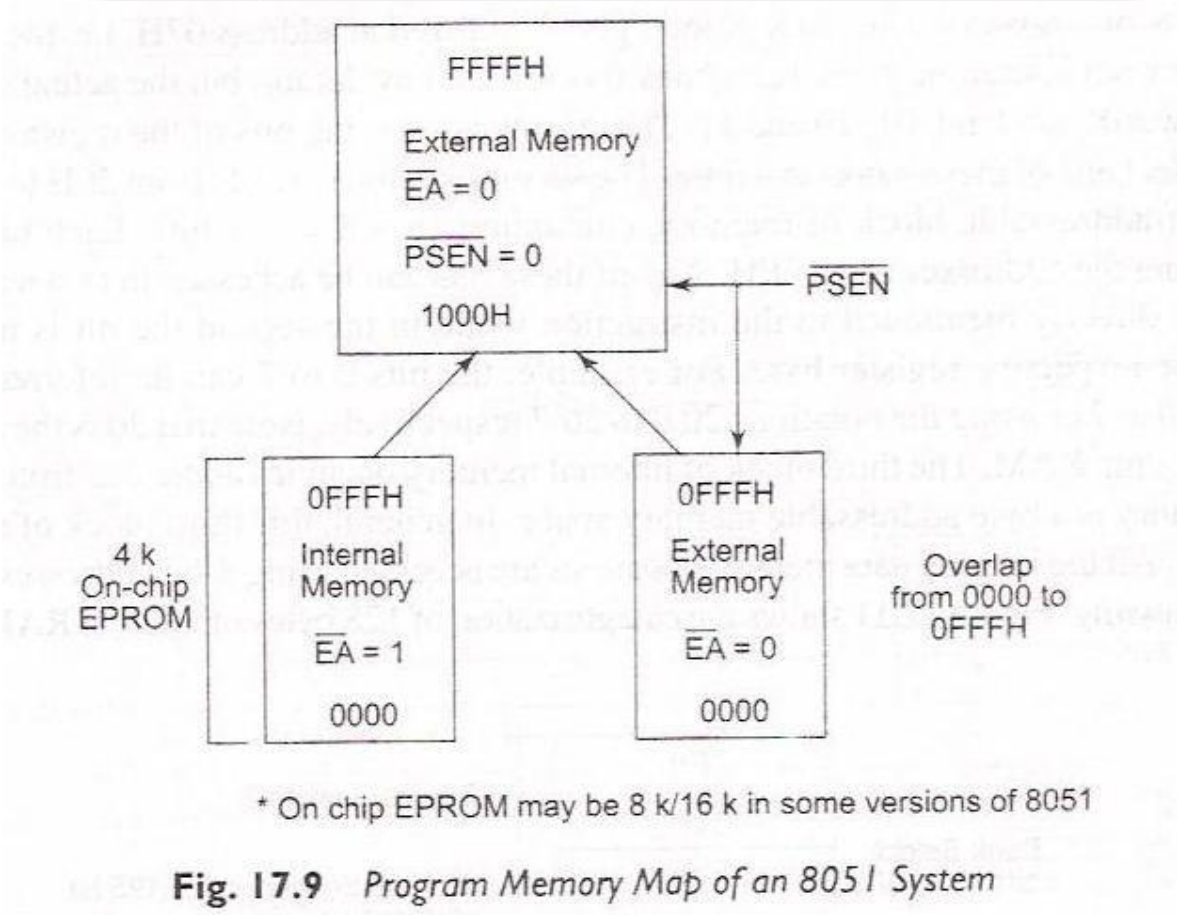


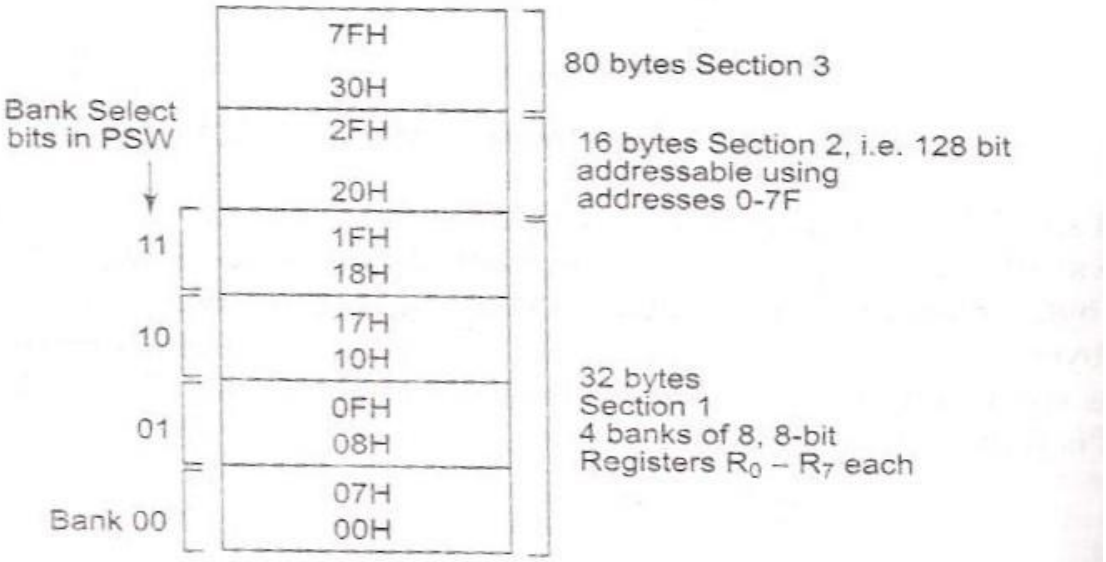
Fig. 17.9 *Program Memory Map of an 8051 System*

INTERNAL RAM STRUCTURE:

- The 8051 has **128 bytes of internal data RAM**, which is accessible as **bytes or sometimes as bits**.
- The **address of the internal RAM starts at 00H** and occupies **space up to 7FH**.

The **RAM** space is divided into **three blocks—the register banks (**00H-1FH:32 bytes) **, the bit-addressable memory (**20H-2FH – 16bytes**)**, and the **scratch pad memory (**30H-7FH: 80 bytes**)**.

- The 8051 has **four register banks** of **eight registers** each, with addresses from **00H to 1FH**. In assembly language, they are addressed by the names **R0–R7**.



ig. 17.11 *Functional Description of Internal Lower 128 Bytes of RAM*

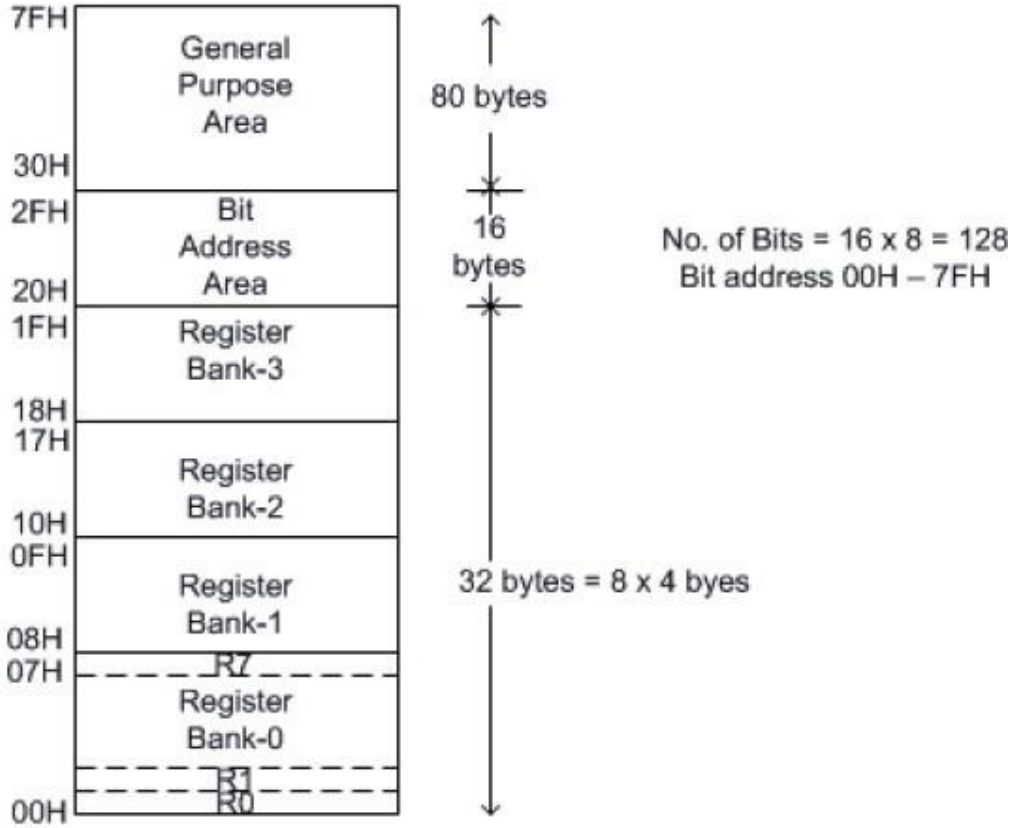


Fig: Internal RAM Structure

Internal Data Memory and Special Function Register (SFR) Map

- The register banks are identified with 2 bits in the **processor status word**.
- The PSW has two bits for identifying the register bank, i.e., 00 represents bank 0, 01 represents bank 1, 10 represents bank 2, and 11 represents bank 3.

- In the 8051, bitwise operations are also possible with special instructions using the bit addresses. The bit-addressable memory is both bit-addressable (from 00H to 7FH) and byte-addressable (from 20H to 2FH). Bit operations are helpful in many control algorithms.
- Using general-purpose scratch pad memory, programmers can read and write data at any time for any purpose. This memory ranges from the byte address 30H to the address 7FH.

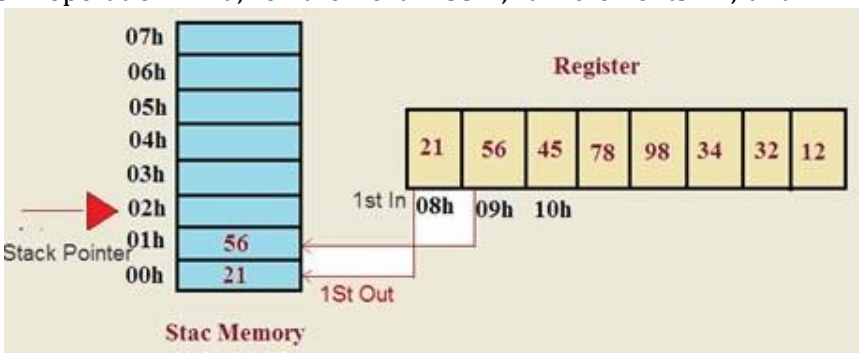
THE STACK OPERATION:

- In the 8051, the stack is configured as a series of memory locations following the Last-In First-Out (LIFO) pattern.
- In general, the stack is initialized in the internal RAM area. Any 8-bit data can be stored and retrieved from the stack using PUSH and POP instructions, with the help of the stack pointer.
- The stack pointer (SP) is an 8-bit register within the SFR area, with the address 81H.
- This register can hold one 8-bit address at a time, which is actually the memory location at top of the stack.
- A push operation in the 8051 is used to store an 8-bit data in the stack.
- The PUSH instruction first increments the value of SP and then stores the data mentioned in the instruction in the memory location pointed to by SP.
- Similarly, the POP instruction stores the value from the top of the stack in the register mentioned in the instruction and then decrements the value of SP.
- The stack pointer is initialized to the value 07H when the 8051 microcontroller is reset.
- The other instructions of the 8051 that affect the stack and the stack pointer are ACALL, LCALL, RET, and RETI.
- The stack pointer can be initialized to any internal RAM address by the programmer, by writing the required address in the SP SFR address 81H.

PUSH operation:

The 'PUSH' is used for taking the values from any register and storing in the starting address of the stack pointer, i.e., 00h by using 'PUSH' operation. And, for the next 'PUSH', it increments +1, and stores the value in the next address of the stack pointer, i.e., 01h.

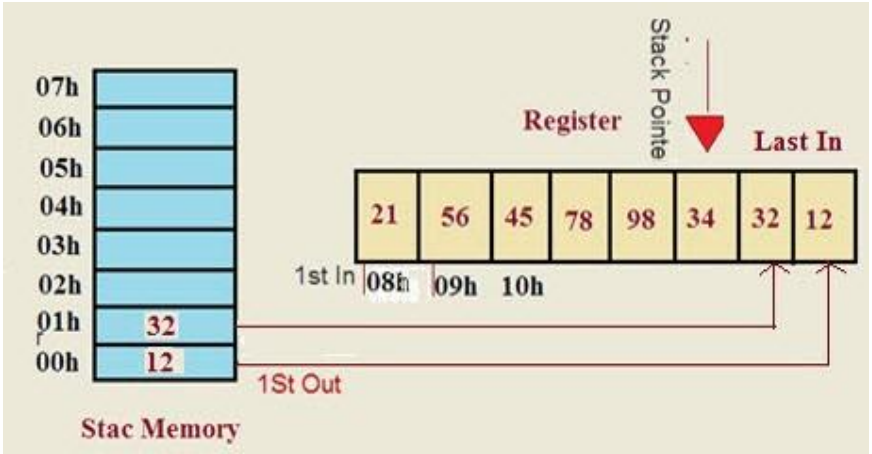
```
0000h
MOV 08h, #21h
MOV 09h, #56h
PUSH  00h
PUSH 01h
END
```



POP OPERATION:

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'.

```
000H
MOV 00H, #12H
MOV 01H, #32H
POP 1FH
POP 0EH
END
```



TIMERS/ COUNTERS:

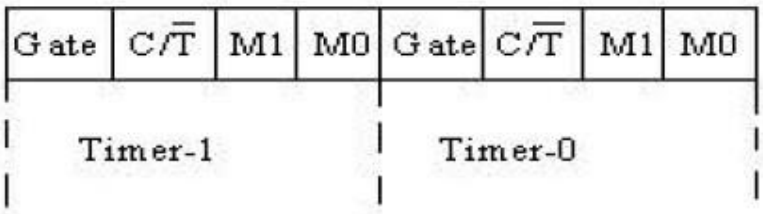
8051 has two 16-bit programmable UP timers/counters. They can be configured to operate either as timers or as event counters. The names of the two counters are T0 and T1 respectively. The timer content is available in four 8-bit special function registers, viz, TL0, TH0, TL1 and TH1 respectively.

In the "timer" function mode, the counter is incremented in every machine cycle. Thus, one can think of it as counting machine cycles. Hence the clock rate is 1/12 th of the oscillator frequency. In the "counter" function mode, the register is incremented in response to a 1 to 0 transition at its corresponding external input pin (T0 or T1). It requires 2 machine cycles to detect a high to low transition. Hence maximum count rate is 1/24 th of oscillator frequency. The operation of the timers/counters is controlled by two special function registers, TMOD and TCON respectively.

Timer Mode control (TMOD) Special Function Register:

TMOD register is not bit addressable.

TMOD Address: 89 H



Various bits of TMOD are described as follows -

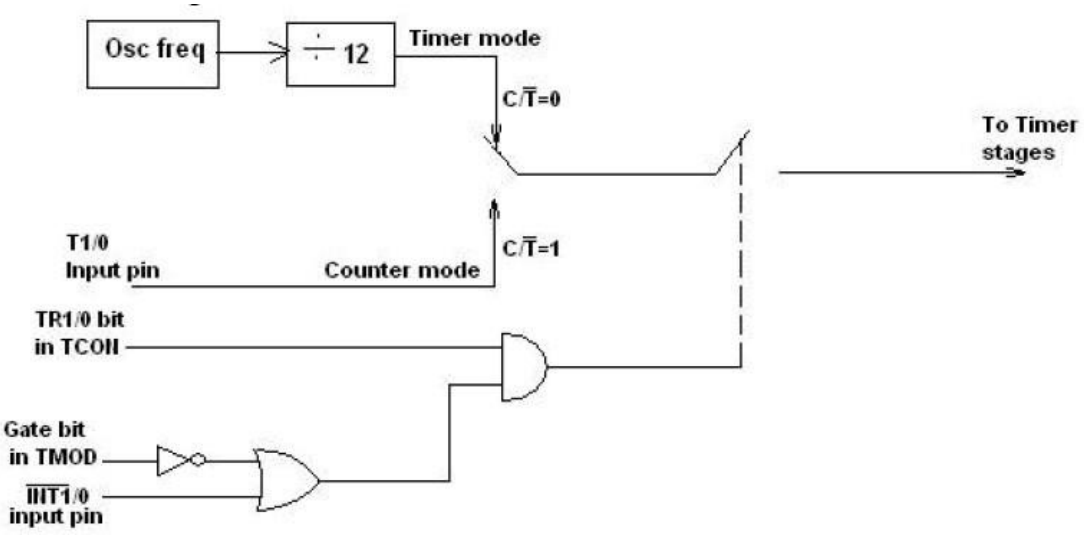
Gate: This is an OR Gate enabled bit which controls the effect of on START/STOP of Timer. It is set to one ('1') by the program to enable the interrupt to start/stop the timer. If TR1/0 in TCON is set and signal on pin is high then the timer starts counting using either internal clock (timer mode) or external pulses (counter mode).

It is used for the selection of Counter/Timer mode. Mode Select Bits:

M1 and M0 are mode select bits.

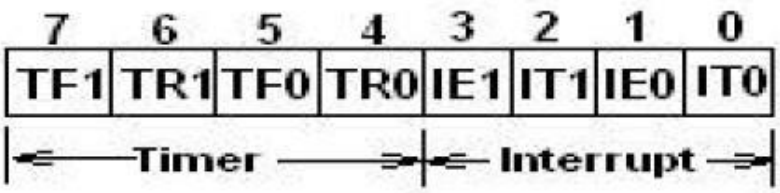
M1	M0	Mode
0	0	Mode 0
0	1	Mode 1
1	0	Mode 2
1	1	Mode 3

Timer/ Counter control logic:



Timer control (TCON) Special function register:

TCON is bit addressable. The address of TCON is 88H. It is partly related to Timer and partly to interrupt.



The various bits of TCON are as follows.

TF1: Timer1 overflow flag. It is set when timer rolls from all 1s to 0s. It is cleared when processor vectors to execute ISR located at address 001BH.

TR1: Timer1 run control bit. Set to 1 to start the timer / counter.

TF0: Timer0 overflow flag. (Similar to TF1) TR0: Timer0 run control bit.

IE1: Interrupt1 edge flag. Set by hardware when an external interrupt edge is detected. It is cleared when interrupt is processed.

IE0: Interrupt0 edge flag. (Similar to IE1)

IT1: Interrupt1 type control bit. Set/ cleared by software to specify falling edge / low level triggered external interrupt.

IT0: Interrupt0 type control bit. (Similar to IT1) As mentioned earlier, Timers can operate in four different modes. They are as follows

INTERRUPTS IN 8051:

8051 provides 5 vectored interrupts. They are

1. **INT0'**
2. **TF0**
3. **INT1"**
4. **TF1**
5. **RI/TI**

Out of these, INT0' and INT1' are external interrupts whereas Timer and Serial port interrupts are generated internally. The external interrupts could be negative edge triggered or low level triggered. All these interrupt, when activated, set the corresponding interrupt flags. Except for serial interrupt, the interrupt flags are cleared when the processor branches to the Interrupt Service Routine (ISR). The external interrupt flags are cleared on branching to Interrupt Service Routine (ISR), provided the interrupt is negative edge triggered. For low level triggered external interrupt as well as for serial interrupt, the corresponding flags have to be cleared by software by the programmer.

Interrupt Enable register (IE):
Address: A8H

7	6	5	4	3	2	1	0
EA	—	ET2	ES	ET1	EX1	ET0	EX0

EX0-- INT0' interrupt (External) enable bit
ET0-- Timer-0 interrupt enable bit
EX1 – INT1' interrupt (External) enable bit
ET1- - Timer-1 interrupt enable bit
ES-- Serial port interrupt enable bit
ET2-- Timer-2 interrupt enable bit
EA-- Enable/Disable all
Setting '1' -- Enable the corresponding interrupt
Setting '0'-- Disable the corresponding interrupt

Priority level structure:

Each interrupt source can be programmed to have one of the two priority levels by setting (high priority) or clearing (low priority) a bit in the IP (Interrupt Priority) Register. A low priority interrupt can itself be interrupted by a high priority interrupt, but not by another low priority interrupt. If two interrupts of different priority levels are received simultaneously, the request of higher priority level is served. If the requests of the same priority level are received simultaneously, an internal polling sequence determines which request is to be serviced. Thus, within each priority level, there is a second priority level determined by the polling sequence, as follows.

Interrupt Priority register (IP):

7	6	5	4	3	2	1	0
—	—	PT2	PS	PT1	PX1	PT0	PX0

Source	Priority level
IE0	Highest
TF0	
IE1	
TF1	
RI+TI	Lowest

SERIAL INTERFACE:

The serial port of 8051 is full duplex, i.e., it can transmit and receive simultaneously. The register SBUF is used to hold the data. The special function register SBUF is physically two registers. One is, write-only and is used to hold data to be transmitted out of the 8051 via TXD. The other is, read-only and holds the received data from external sources via RXD. Both mutually exclusive registers have the same address 099H.

Serial Port Control Register (SCON):

Register SCON controls serial data communication. Address: 098H (Bit addressable)

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Mode select bits:

SM2: multi processor communication bit
REN: Receive enable bit
TB8: Transmitted bit 8 (Normally we have 0-7 bits transmitted/received)
RB8: Received bit 8
TI: Transmit interrupt flag
RI: Receive interrupt flag

SM0	SM1	Mode
0	0	Mode 0
0	1	Mode 1
1	0	Mode 2
1	1	Mode 3

Power Mode control Register(PCON):

Register PCON controls processor power down, sleep modes and serial data baud rate. Only one bit of PCON is used with respect to serial communication. The seventh bit (b7) (SMOD) is used to generate the baud rate of serial communication.
Address: 87H

b7							b0
SMOD	—	—	—	GF1	GF0	PD	IDL

SMOD: Serial baud rate modify bit
GF1: General purpose user flag bit 1
GF0: General purpose user flag bit 0
PD: Power down bit IDL: Idle mode bit

Data Transmission

Transmission of serial data begins at any time when data is written to SBUF. Pin P3.1 (Alternate function bit TXD) is used to transmit data to the serial data network. TI is set to 1 when data has been transmitted. This signifies that SBUF is empty so that another byte can be sent.

Data Reception

Reception of serial data begins if the receive enable bit is set to 1 for all modes. Pin P3.0 (Alternate function bit RXD) is used to receive data from the serial data network. Receive interrupt flag, RI, is set after the data has been received in all modes. The data gets stored in SBUF register from where it can be read.

Serial Communication: Different modes:

Mode - 0 Shift register mode.

Serial data enters and exists through RXD. 8-bits are transmitted/recieved. Pin TXD is connected to the internal shift frequency pulse source to supply shift pulses to external circuits. The shift frequency or baud rate is fixed at 1/2 of the oscillator frequency.

Mode - 1 Standard UART

10 bits are transmitted (through TXD) or recieved through (RXD), a start bit(0), 8 data bits (LSB first), and a stop bit(1). Once recieved, the stop bits goes into RB8 in special function register SCON.The baud rate is variable.

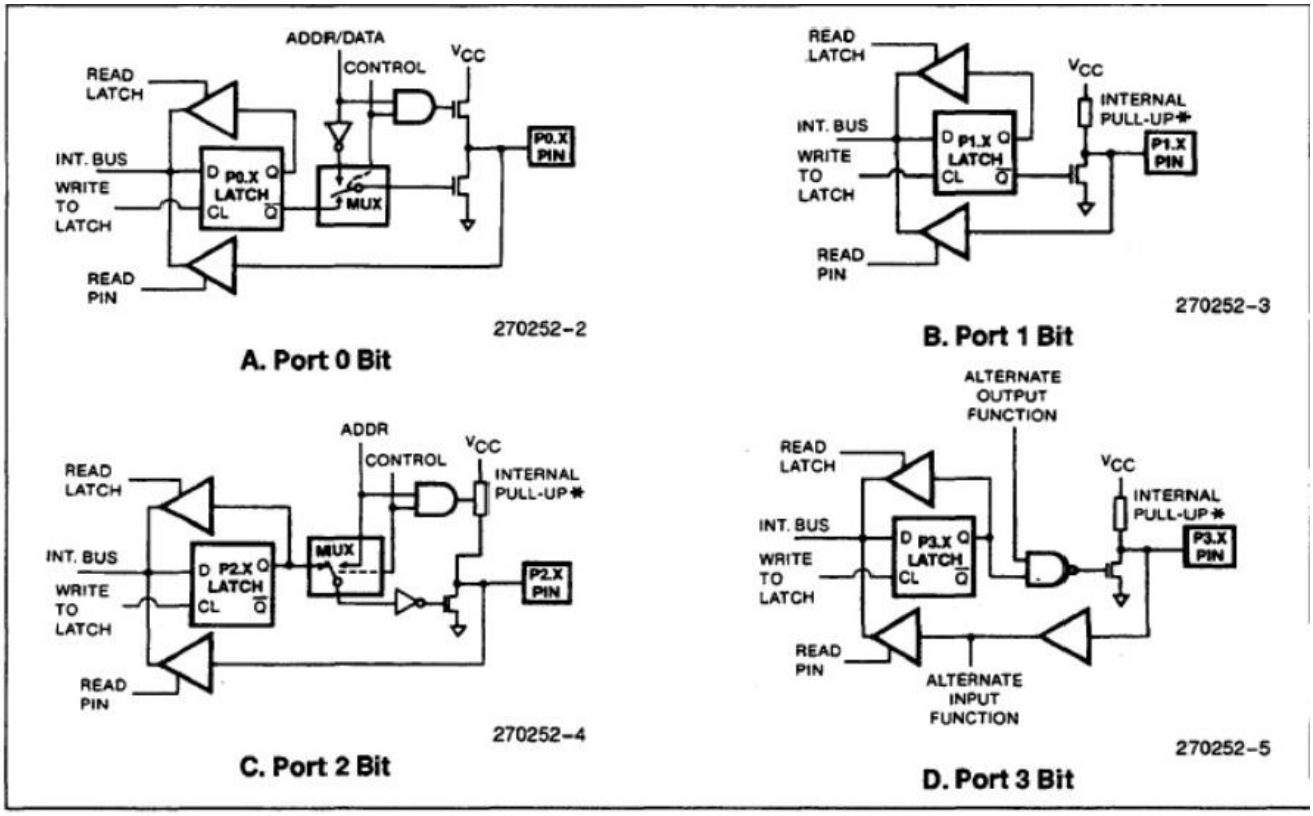
Mode - 2 Multiprocessor Mode.

11bits are transmitted through TXD or received through RXD, a start bit (0), 8 data bits (LSB first), a programmable 9th bit and a stop bit(1). On transmission, the 9th data bit (TB8 in SCON) can be assigned the value 0 or 1. Or, for example, the parity bit (P in the PSN) could be moved into TB8. On receive, the 9th bit goes into RB8 in SFR SCON, which the stop bit is ignored. The bandwidth is programmable to either 1/32 or 1/64 of oscillator frequency.

Mode - 3

11 bits are transmitted through TXD or received through RXD: a start bit, 8 data bits (LSB first), a programmable 9th bit, and a stop bit (1). In fact, Mode 3 is same as Mode 2 in all respects except the baud rate. The baud rate in Mode 3 is variable.

8051 Ports:



All four ports in the 8051 are bidirectional. Each consists of a latch (Special Function Registers PO through P3), an output driver, and an input buffer. The output drivers of Ports 0 and 2, and the input buffers of Port 0, are used in accesses to external memory.

In this application, Port 0 outputs the low byte of the external memory address, time-multiplexed with the byte being written or read. Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise the Port 2 pins continue to emit the P2 SFR content. All the Port 3 pins are multifunctional. They are not only port pins, but also serve the functions of various special features as listed on the following page.

Figure shows a functional diagram of a typical bit latch and I/O buffer in each of the four ports. The bit latch (one bit in the port's SFR) is represented as a Type D flip-flop, which will clock in a value from the internal bus in response to a "write to latch" signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a "read latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read pin" signal from the CPU. Some instructions that read a port activate the "read latch" signal, and others activate t e "read pin" signal.

More about that later. As shown in Figure, the output drivers of Ports 0 and 2 are switchable to an internal ADDR and ADDR/ DATA bus by an internal CONTROL signal for use in external memory accesses. During external memory accesses, the P2 SFR remains unchanged, but the PO SFR gets Is written to it.

Also shown in Figure , is that if a P3 bit latch contains a 1, then the output level is controlled by the signal labeled "alternate output function." The actual P3.X pin level is always available to the pin's alternate input function, if any. Ports 1, 2, and 3 have internal pullups. Port 0 has open drain outputs. Each I/O line can be independently used as an input or an output. (ports 0 and 2 may not be used as general purpose I/O when being used as the ADDR/DATA BUS). To be used as an input, the port bit latch must contain a 1, which turns off the output driver FET. Then, for Ports 1, 2, and 3, the pin is pulled high by the internal pullup, but can be pulled low by an external source.

Port 0 differs in not having internal pullups. The pullup FET in the PO output driver (see Figure 4) is used only when the Port is emitting Is during external memory accesses. Otherwise the pullup PET is off. Consequently PO lines that are being used as output port lines are open drain. Writing a 1 to the bit latch leaves both output FETs off, so the pin floats. In that condition it can be used a high-impedance input. Because Ports 1, 2, and 3 have fixed internal pull-ups they are sometimes called "quasi-bidirectional" ports.

When configured as inputs they pull high and will source current (IlL, in the data sheets) when externally pulled low. Port 0, on the other hand, is considered "true" bidirectional, because when configured as an input it floats. All the port latches in the 8051 have Is written to them by the reset function. If a 0 is subsequently written to a port latch, it can be reconfigured as an input by writing a 1 to it.