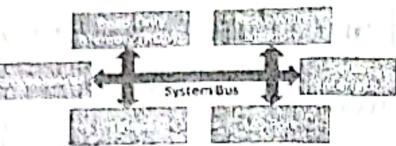
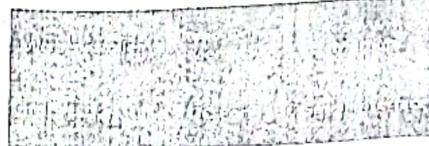


To make a complete micro computer system, only CPU is not sufficient. It is necessary to add other peripherals such as read only memory (ROM), read / write memory (RAM), decoders, drivers, number of input / output devices to make a complete micro computer system. In addition, special purpose devices, such as interrupt controller, programmable timers, programmable I/O devices, DMA controllers may be added to improve the capability and performance and flexibility of a micro computer system.

The micro controller incorporates all the features that are found in microprocessor. It has also added features to make a complete micro computer system on its own. The microcontroller has built-in ROM, RAM, parallel I/O, serial I/O counters and a clock circuit.

The micro controller has on-chip (built-in) peripheral devices. These on-chip peripherals make it possible to have single chip microcomputer system. These are few more advantages of built-in peripherals:

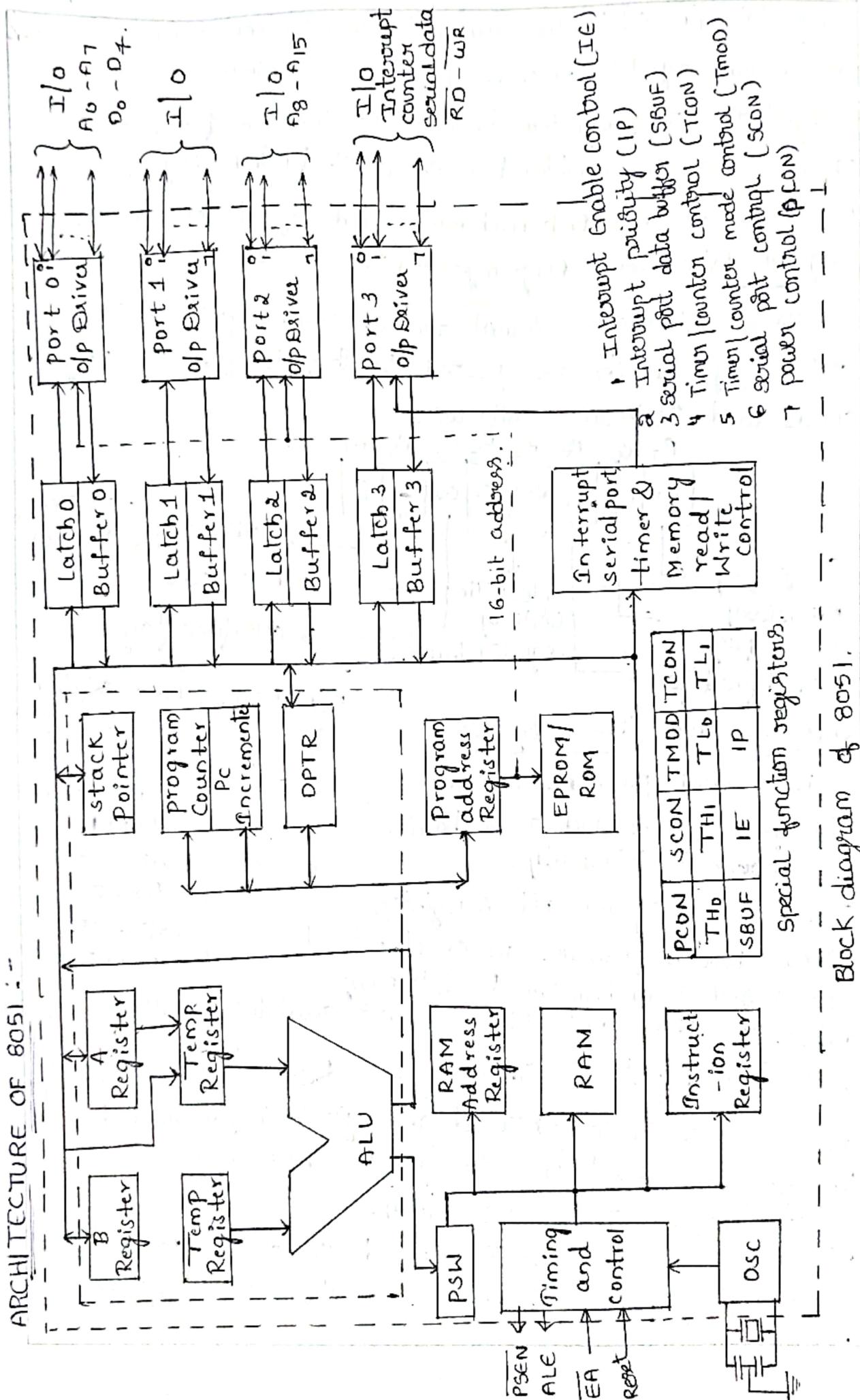
- Built-in peripherals have smaller access times hence speed is more.
- hardware reduces due to single chip micro computer system.
- less hardware, reduces PCB size and increases reliability of the system.

Microprocessor	Microcontroller
	
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 use 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

### Features of 8051 microcontroller

- 4 KB on chip program memory.
- 128 bytes on chip data memory(RAM).
- 128 user defined software flags.
- 8-bit data bus
- 16-bit address bus
- 32 general purpose registers each of 8 bits
- 16 bit timers (usually 2, but may have more, or less).
- 3 internal and 2 external interrupts.
- Bit as well as byte addressable RAM area of 16 bytes.
- Four 8-bit ports, (short models have two 8-bit ports).
- 16-bit program counter and data pointer.
- 1 Microsecond instruction cycle with 12 MHz Crystal.

## ARCHITECTURE OF 8051 :-



Special function registers.

Block diagram of 8051.

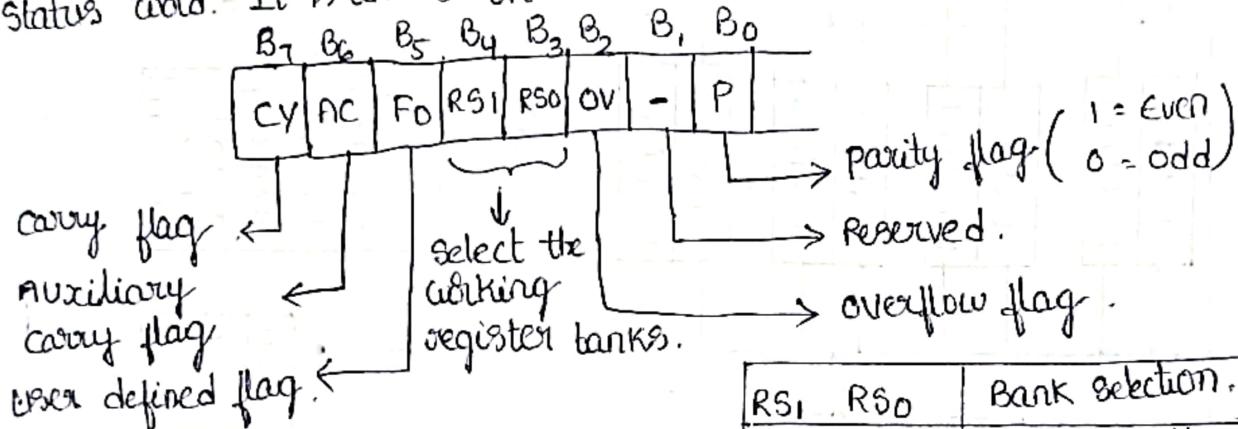
Accumulator (ACC) :- The accumulator register (ACC 81 A) acts as an operand register, in case of some instructions.

B Register :- This register is used to store one of the operands. This register is considered as a special function register.

ALU :- It performs Both Arithmetic and logical operations.

Program Status word (Flag Register) :-

Many instructions internal and external affect several status flags, which are grouped together to form the program status word. It is an 8-bit word.



RAM and RAM address Register :-

These blocks provides internal 128 bytes of RAM and a mechanism to address it internally.

EPROM and program address Registers :-

These blocks provides an on-chip EPROM and a mechanism to internally address it. Note that EPROM is not available in all 8051 versions.

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.

Timing and control unit :- This unit derives all the necessary timing and control signals required for the internal operation of the circuit.

RS <sub>1</sub>	RS <sub>0</sub>	Bank Selection
0	0	00H - 07H Bank 0
0	1	08H - 0FH Bank 1
1	0	10H - 17H Bank 2
1	1	18H - 1FH Bank 3

Stack pointer (SP) :- This 8-bit wide register is incremented before the data is stored onto the stack using push & call instructions.

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

Port 0 to 3 latches and drivers :- There four latches and drivers pairs are allotted to each of the four on-chip I/O ports. These are identified as P<sub>0</sub>, P<sub>1</sub>, P<sub>2</sub> and P<sub>3</sub>.

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

Timer Registers :- These two 16-bit registers can be accessed as their lower and upper bytes. TLO and TH0 represents the lower byte and higher bytes of the timing register 0. Similarly TL1 and TH1 represents the lower byte and higher bytes of the timing register 1.

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

Serial Data Buffer :- 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.

## Pin diagram of 8051 :-

8051 is packaged in a 40 pin plastic and ceramic dual line package.

→ V<sub>CC</sub> this is a +5V supply voltage

Pin. V<sub>SS</sub> is a return pin for the supply. Reset input pin resets

the 8051, only when it goes high for a proper reinitialization.

→ ALE/PROG:- The address latch enable output pulse indicates that the valid address bits are available on their respective pins. This ALE signal is valid only for external memory accesses.

→ EA/VPP:- External access enable pin, if tied low, indicates that the 8051 can address external program memory. For execution of programs in internal memory, the EA must be tied high. This pin also receives 21 volts for programming of the on-chip EPROM.

→ PSEN :- (Program store enable) that acts as a strobe to read the external program memory.

→ Port 0 (P<sub>0.0</sub> - P<sub>0.7</sub>):- Port 0 is an 8-bit bidirectional bit addressable I/O port. Port 0 act as multiplexed address/data lines during external memory access when EA is low.

→ Port 1 (P<sub>1.0</sub> - P<sub>1.7</sub>):- Port 1 acts as an 8-bit bi-directional bit addressable port.

→ Port 2 (P<sub>2.0</sub> - P<sub>2.7</sub>):- Port 2 acts as 8-bit bi-directional bit addressable I/O port. During external memory access, port 2 emits higher eight bits of address (A<sub>8</sub>-A<sub>15</sub>) which are valid, if ALE goes high and EA is low.

P <sub>1.0</sub>	1	micro	40	V <sub>CC</sub>
P <sub>1.1</sub>	2		39	P <sub>0.0</sub> (AD <sub>0</sub> )
P <sub>1.2</sub>	3		38	P <sub>0.1</sub> (AD <sub>1</sub> )
P <sub>1.3</sub>	4	Controller	37	P <sub>0.2</sub> (AD <sub>2</sub> )
P <sub>1.4</sub>	5		36	P <sub>0.3</sub> (AD <sub>3</sub> )
P <sub>1.5</sub>	6		35	P <sub>0.4</sub> (AD <sub>4</sub> )
P <sub>1.6</sub>	7		34	P <sub>0.5</sub> (AD <sub>5</sub> )
P <sub>1.7</sub>	8		33	P <sub>0.6</sub> (AD <sub>6</sub> )
Reset	9		32	P <sub>0.7</sub> (AD <sub>7</sub> )
RxD	P <sub>3.0</sub>	10	31	EA/VPP
TxD	P <sub>3.1</sub>	11	30	ALE/PROG
INT <sub>0</sub>	P <sub>3.2</sub>	12	29	PSEN
INT <sub>1</sub>	P <sub>3.3</sub>	13	28	P <sub>2.7</sub> (A <sub>15</sub> )
T <sub>0</sub>	P <sub>3.4</sub>	14	27	P <sub>2.6</sub> (A <sub>14</sub> )
T <sub>1</sub>	P <sub>3.5</sub>	15	26	P <sub>2.5</sub> (A <sub>13</sub> )
WR	P <sub>3.6</sub>	16	25	P <sub>2.4</sub> (A <sub>12</sub> )
RD	P <sub>3.7</sub>	17	24	P <sub>2.3</sub> (A <sub>11</sub> )
XTAL2	P <sub>3.8</sub>	18	23	P <sub>2.2</sub> (A <sub>10</sub> )
XTAL1	P <sub>3.9</sub>	19	22	P <sub>2.1</sub> (A <sub>9</sub> )
V <sub>SS</sub>		20	21	P <sub>2.0</sub> (A <sub>8</sub> )

Port 3 ( $P_{3.0} - P_{3.7}$ ) :- Port 3 is an 8-bit bidirectional bit addressable I/O port which serve the alternative functions.

XTAL<sub>1</sub> and XTAL<sub>2</sub> :- There is an inbuilt oscillator which derives the necessary clock frequency for the operation of the controller. XTAL<sub>1</sub> is the input of amplifier and XTAL<sub>2</sub> is the output of the amplifier. A crystal is to be connected externally between these two pins. The external clock is fed to the controller at pin XTAL<sub>2</sub> and XTAL<sub>1</sub> pin should be grounded. commercially available versions of 8051 run on 12MHz to 16MHz frequency.

Interrupts of 8051 :-

Interrupts are the events that temporarily suspend the main program, pass the control to the external sources and execute their task. 8051 has 5 interrupt signals. INT<sub>0</sub>, INT<sub>1</sub>, TFO, TFI, R<sub>1</sub> T<sub>1</sub> (serial port).

INT<sub>0</sub> and INT<sub>1</sub> are two external interrupt inputs. These can either be edge-sensitive or level-sensitive, as programmed with bits IT<sub>0</sub> and IT<sub>1</sub> in register TCON. These interrupts are processed internally by the flags IE<sub>0</sub> and IE<sub>1</sub>.

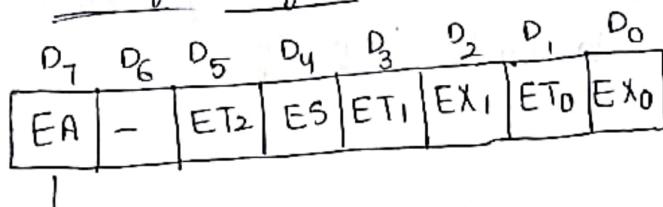
The Timer 0 and Timer 1 interrupt sources are generated by TFO and TFI bits of the register TCON.

The serial port interrupt is generated, if at least one of the two bits R<sub>1</sub> and T<sub>1</sub> is set. The flag is cleared, after the control is cleared is transferred to the interrupt service routine.

All these Interrupts are enabled using a special function register called interrupt enable register (IE) and their priorities are programmed using another special function register called interrupt priority register (IP).

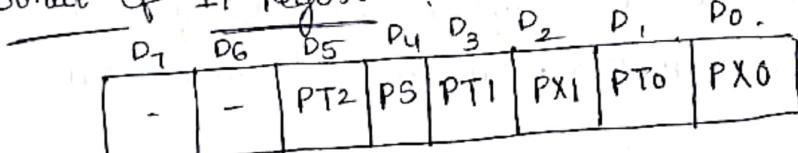
Interrupt source	Priority within a level	vector address
IE0 (External INT0)	Highest	0003h
TF0 (Timer0)		000Bh
IE1 (External INT1)		0013h
TF1 (Timer1)		001Bh
RI = TI (Serial port)	Lowest	0023h

Format of IE Register :-



- D<sub>7</sub> - No interrupt will be acknowledged.
- D<sub>6</sub> - Each interrupt source is individually enabled or disabled.
- D<sub>6</sub> - Not implemented, reserved for future use.
- D<sub>5</sub> - This enables or disables Timer 2.
- D<sub>4</sub> - This enables or disables the Serial port interrupt.
- D<sub>3</sub> - This enables or disables the Timer 1.
- D<sub>2</sub> - This enables or disables external interrupt 1.
- D<sub>1</sub> - This enables or disables the timer 0.
- D<sub>0</sub> - This enables or disables external interrupt 0.

Format of IP Register :-



- D<sub>7</sub>, D<sub>6</sub> - Not implemented, reserved for future use.
- PT<sub>2</sub> - D<sub>5</sub> - This defines the Timer 2 interrupt priority level (8052 only).
- D<sub>4</sub> - PS - Serial port interrupt priority level.
- D<sub>3</sub> | D<sub>1</sub> - PT<sub>1</sub> | PT<sub>0</sub> - Timer 1 | Timer 0 interrupt priority level.
- D<sub>2</sub> | D<sub>0</sub> - PX<sub>1</sub> | PX<sub>0</sub> - This defines External INT1 | INT0 priority level.

## Instruction set of 8051 :-

8051 instruction set supports six addressing modes.

1. direct addressing
2. Indirect addressing
3. Register Instructions
4. Register specific
5. Immediate mode
6. Indexed addressing.

→ Direct addressing :- In this mode of addressing, the operands are specified using the 8-bit address field, in the instruction format.

Ex:- MOV R0, 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.

Ex:- ADD A, @R0. (The address register for 16-bit addresses can only be DPTR).

→ Register Instructions :- In this addressing mode, operands are stored in the register R0-R7 of the selected register bank.

Ex:- ADD A, R7.

→ 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.

Ex:- RLA (This instruction rotates accumulator left).

→ Immediate mode :- In this mode, an immediate data, a constant is specified in the instruction after the opcode byte.

Ex:- ADD A, #100.

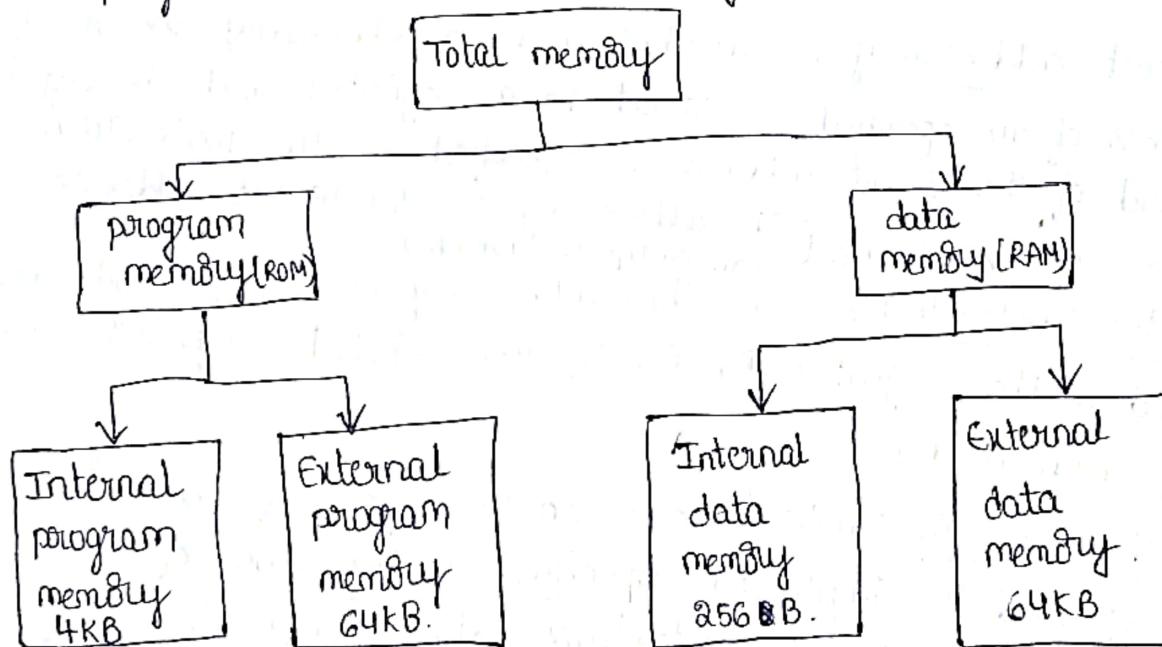
→ Indexed Addressing :- only program memory can be accessed using this addressing mode. Program counter or data pointer are the allowed 16-bit address storage registers, in this mode of addressing. This addressing mode is basically used for accessing data from look up table.

Ex:- MOVC A,@ A + DPTR

here 'C' means code. here the content of A register is added with content of DPTR and the resultant is the address of memory location from where the data is copied to A register.

→ Memory Organization :-

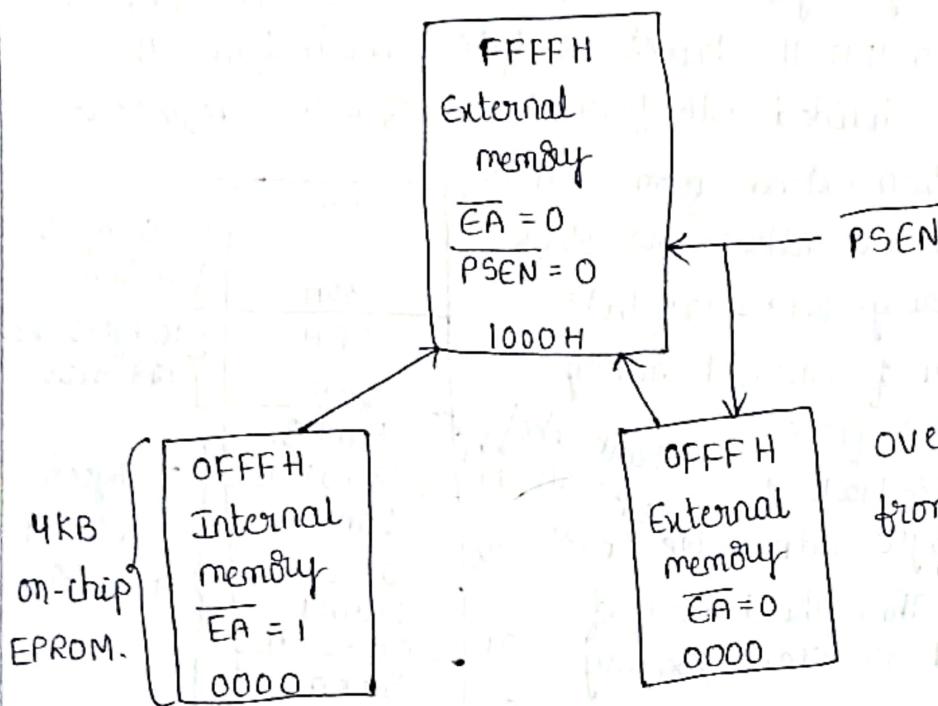
The total memory of an 8051 system is logically divided into program memory and data memory.



Classification of total memory.

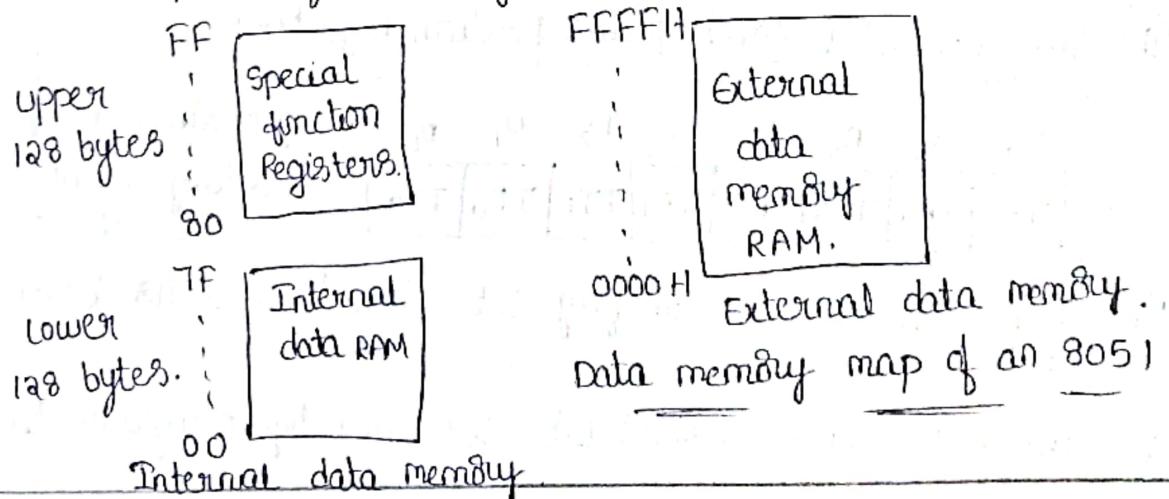
The 8051 can address 4K bytes on-chip program, whose map starts from 0000H and ends at 0FFFH. It can address 64K bytes of external program memory under the control of PSEN signal, whose address map is from 0000H to FFFFH. Here, one may note that the map of internal program memory overlaps with the external program memory. These two memory spaces

can be distinguished using the PSEN signal.



program memory map of an 8051 system

8051 supports 64 K bytes of external data memory which map starts at 0000H and ends at FFFFH. This external data memory can be accessed under the control of Register DPTR, which stores the addresses for external data memory accesses. Internal data memory of 8051 consists of two parts; the first is the RAM block of 128 bytes and the second is the set of addresses from 80H to FFH, which includes the addresses allotted to the Special function registers.



Data memory map of an 8051

The lower 128 bytes of RAM whose address map is from 00 to 7FH is functionally organised in three sections. The address block from 00 to 1FH, the lowest 32 bytes which form the first section is divided into four banks of 8-bit registers.

The second section extends from 20H to 2FH which is a bit addressable block of memory, containing  $16 \times 8 = 128$  bits.

The third section of internal memory

occupies addresses from 30H to 7FH. This block of memory is a byte addressable PSW space. This third block of memory is used as stack memory.

Counters/Timers :-

Many microcontroller applications require the counting of external events, such as the frequency of a pulse train, or the generation of precise internal time delays between computer actions.

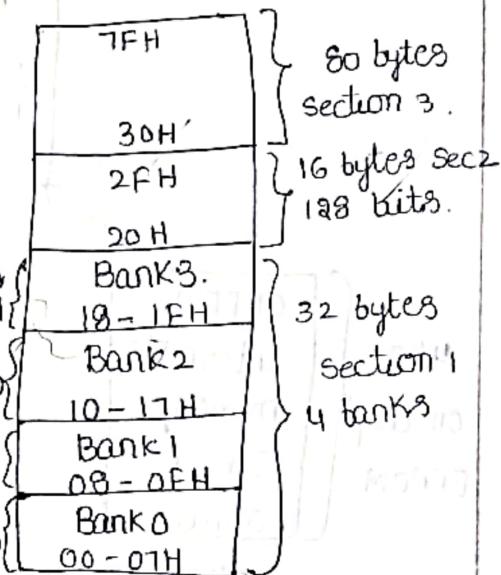
The counters are divided into two 8-bit registers called the timer low (TLO, TL1) and high (TH0, TH1) bytes. All counter action is controlled by bit states in the timer mode control register (TMOD), the timer/counter control register (TCON)

The timer control (TCON) special function Register :-

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	generating delay events.
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	↑ counter

D<sub>7</sub> - TF1 → Timer 1 overflow flag. Set when timer rolls from all 1's to 0.

D<sub>6</sub> - TR1 → Timer 1 run control bit. Set to 1 by program to



enable timer to count.

D<sub>5</sub> - TF<sub>0</sub> → Timer 0 overflow flag set when timer rolls from all is to 0.

D<sub>4</sub> - TR<sub>0</sub> → Timer 0 own control bit set to 1 by program to enable timer to count.

D<sub>3</sub> → IE<sub>1</sub> → External interrupt 1 Edge flag set to 1 when a high-to-low edge signal is received on port 3 pin 3.3 (INT1).

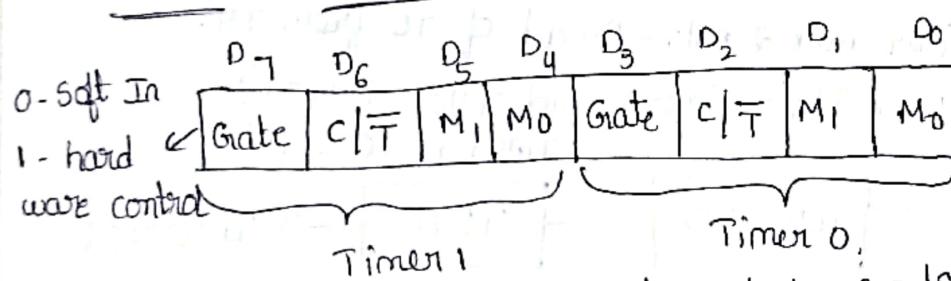
D<sub>2</sub> → IT<sub>1</sub> → External interrupt 1 signal type control bit.

D<sub>1</sub> → set to 1 by program to enable external interrupt 1 to be triggered by a falling edge signal. set to 0 by program to enable low-level signal on external interrupt 1 to generate an interrupt.

D<sub>0</sub> - IE<sub>0</sub> → External interrupt 0 Edge flag set to 1 when a high-to-low edge signal received on port 3 pin 3.2 (INT0).

D<sub>0</sub> - IT<sub>0</sub> → External interrupt 0 signal type control bit.

The timer mode control (TMOD) special function register:-



D<sub>7</sub> & D<sub>3</sub> - Gate → enable bit which controls run/stop of timer 1/0. Set to 1 by program to enable timer to own if bit TR<sub>1</sub> / TR<sub>0</sub> in TCON is set.

D<sub>6</sub> & D<sub>2</sub> - C/T → set to 1 by program to make timer 1/timer 0 act as a counter by counting pulses from external input. Cleared to 0 by program to make timer act as a timer by counting internal frequency.

D<sub>5</sub> & D<sub>1</sub> M<sub>1</sub>

D<sub>4</sub> & D<sub>0</sub> M<sub>0</sub>

Timer / counter operating mode select bit.

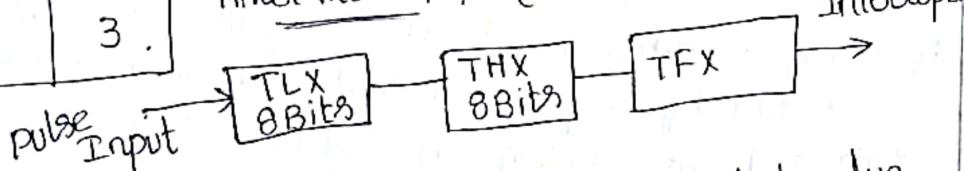
Timer mode 0 :- (TLX = 5 bits, THX = 8 bits)

M <sub>1</sub>	M <sub>0</sub>	mode
0	0	0
0	1	1
1	0	2
1	1	3

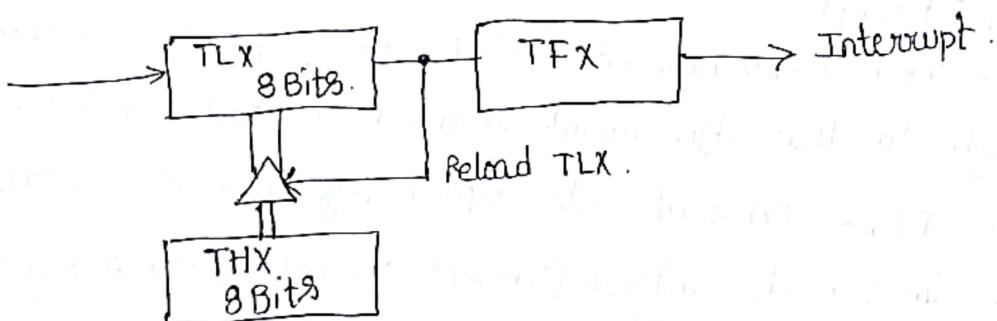


Timer mode 0 13-bit Timer / counter.

Timer mode 1 :- (TLX = 8 bits, THX = 8 bits)

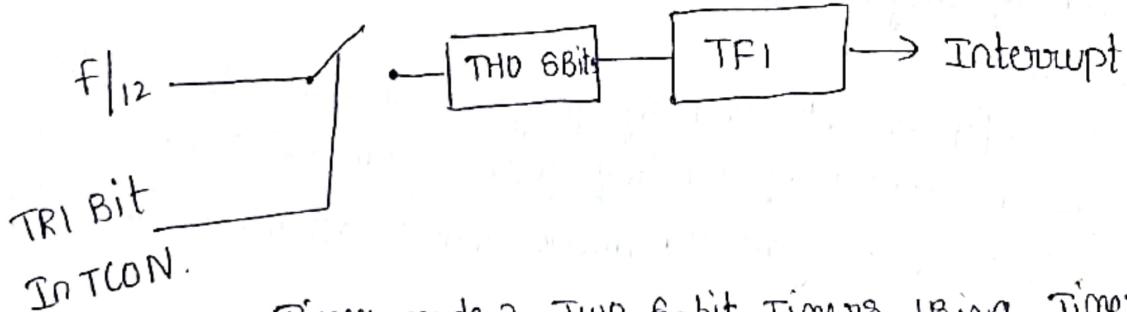
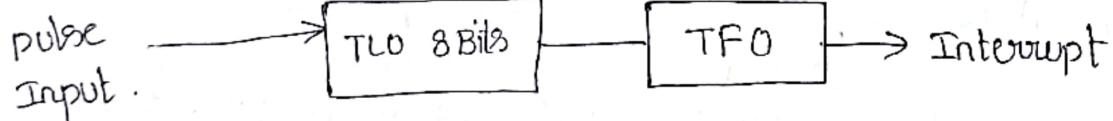


Timer mode 2 :- (TLX = 8 bits, THX = for storing initial value)



timer mode 2 Auto - Reload of TL from TH.

Timer mode 3 :- (TL0 = Timer0 and TH0 = Timer1)  
(timer 1 = 16 bit).



Timer mode 3 Two 8-bit Timers Using Timer0.

## Serial data Input / output :-

Computers must be able to communicate with other computers in modern multiprocessor distributed systems. One effective way to communicate is to send and receive data bits serially. The 8051 has a serial data communication circuit that uses register SBUF to hold data. Register SCON controls data communication, register PCON controls date rates, and pins RXD and TXD connect to the serial data network.

### The serial port control (SCON)

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
SM <sub>0</sub>	SM <sub>1</sub>	SM <sub>2</sub>	REN	TB8	RB8	T <sub>I</sub>	R <sub>I</sub>

Bit D<sub>7</sub>&D<sub>6</sub> - Serial port mode bits. set/cleared by program.

SM<sub>0</sub> SM<sub>1</sub> mode description.

0 0 0 Shift register, baud = f/12

0 1 1 8-bit UART, baud = variable

1 0 0 9-bit UART, baud = f/32 or f/64.

1 1 1 9-bit UART, baud = variable.

Bit D<sub>5</sub> - SM<sub>2</sub> → multiprocessor communications bit.

Bit D<sub>4</sub> - REN → Receive enable bit.

Bit D<sub>3</sub> - TB8 → Transmitted bit 8.

Bit D<sub>2</sub> - RB8 → Received bit 8.

Bit D<sub>1</sub> - T<sub>I</sub> → Transmit interrupt flag.

Bit D<sub>0</sub> - R<sub>I</sub> → Receive interrupt flag.

### The power mode control (PCON)

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
Sm0D	-	-	-	GFI	GFO	PO	IDL

D<sub>7</sub> → SMOD → Serial baud rate modify bit.

D<sub>6</sub>, D<sub>5</sub> & D<sub>4</sub> → Not used.

D<sub>3</sub> → G<sub>F1</sub> → General purpose user flag bit 1.

D<sub>2</sub> → G<sub>F0</sub> → General purpose user flag bit 0.

D<sub>1</sub> → PD → Power down bit.

D<sub>0</sub> → IDL → Idle mode bit.

Instruction set in 8051 :-

1. Data Transfer instructions.

2. Arithmetic instructions.

3. Logical instructions.

4. Boolean instruction (Bit oriented instructions).

5. Branch instructions.

→ Data transfer Instructions :- These instructions are used to transfer the data from one register to another register.

Ex:- MOV B, R<sub>0</sub> → The contents of R<sub>0</sub> is moved into B.

→ Arithmetic instructions :- These instructions are used to perform arithmetic operations.

Ex:- ADD A, R<sub>0</sub> → It performs addition operation between the contents of Register A and Register R<sub>0</sub>.

→ Logical instructions :- These instructions are used to perform basic logical operations along with rotate and clear operations.

Ex:- 1 ANL A, R<sub>2</sub> → And operation.

2 XRL A, R<sub>3</sub> → EX-OR operation

3 RL A → Rotate left operation.

DATA TRANSFER	ARITHMETIC	LOGICAL	BOOLEAN	PROGRAM BRANCHING
MOV (move)	ADD (addition)	ANL (Bit wise And)	CLR (clear)	LJMP (long jump)
MOVC (move code)	ADDC (Add with carry)	ORL (OR)	SETB (Set bit)	AJMP (Absolute jump)
MOVX (move Extended memory)	SUBB (Subtract with Borrow)	XRL (EX-OR)	MOV (move)	SJMP (short jump)
PUSH (push value onto stack)	INC (Increment)	CLR (Clear)	JC (Jump if carry set)	JZ (Jump if zero)
POP (Pop value from stack)	DEC (Decrement)	CPL (Complement)	JNC (Jump if not carry set)	JNZ (Jump if not zero)
XCH (Exchange)	MUL (multiplication)	RL (Rotate ACC left)	JB (Jump if Bit Set)	CJNE (Compare and jump if not equal)
XCHD (Exchange digits).	DIV (Division)	RLC (Rotate ACC with carry)	JNB (Jump if not Bit Set)	DJNZ (Dec flags if not zero)
	DAA (decimal adjust after addition)	RR (rotate Right)	JBC (Jump if Bit Set and clear bit)	NOP (No operation)
		RRC (rotate Right ACC with carry)	ANL	LCALL Long Call
		SWAP	ORL	ACALL Absolute Call
		(Swap Accumulator bubbles)	CPL	RET Return from subroutine
				RETI Return From Interrupt
				JMP (Jump)

→ Boolean or bit oriented instructions :-

Bit oriented instructions perform logic operations.

The difference is that these are performed upon single bits.

Ex:- CPL C → complements the carry flag

CLR C → clear the carry flag

SETB C → set bit carry flag.

→ Branch instructions :- These are two types . 1. unconditional  
2. conditional.

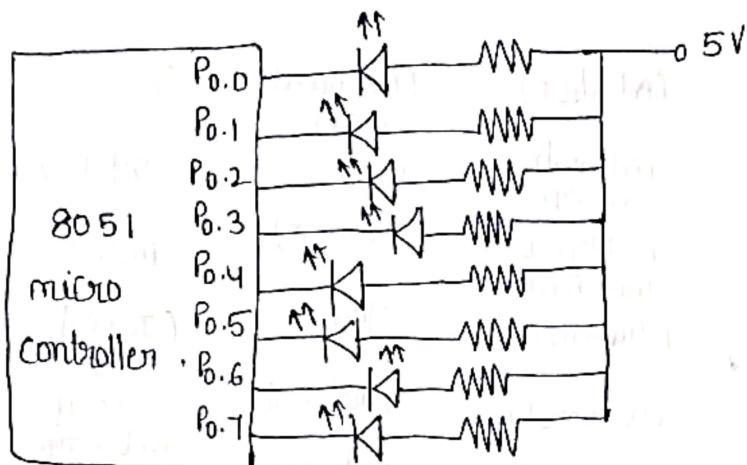
Ex:- A call

RET → return from sub routine.

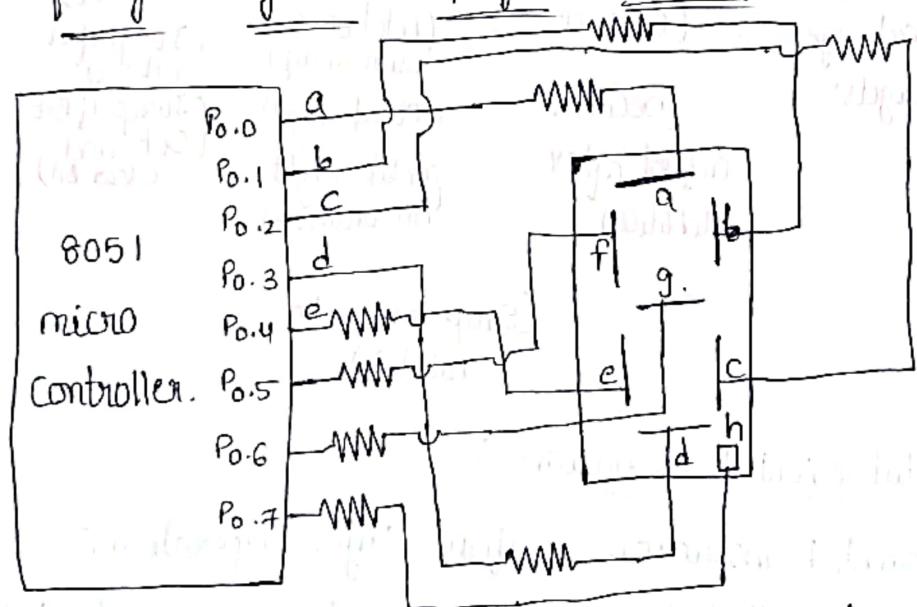
RETI → return from interrupt sub routine.

AjMP → Absolute Jump.

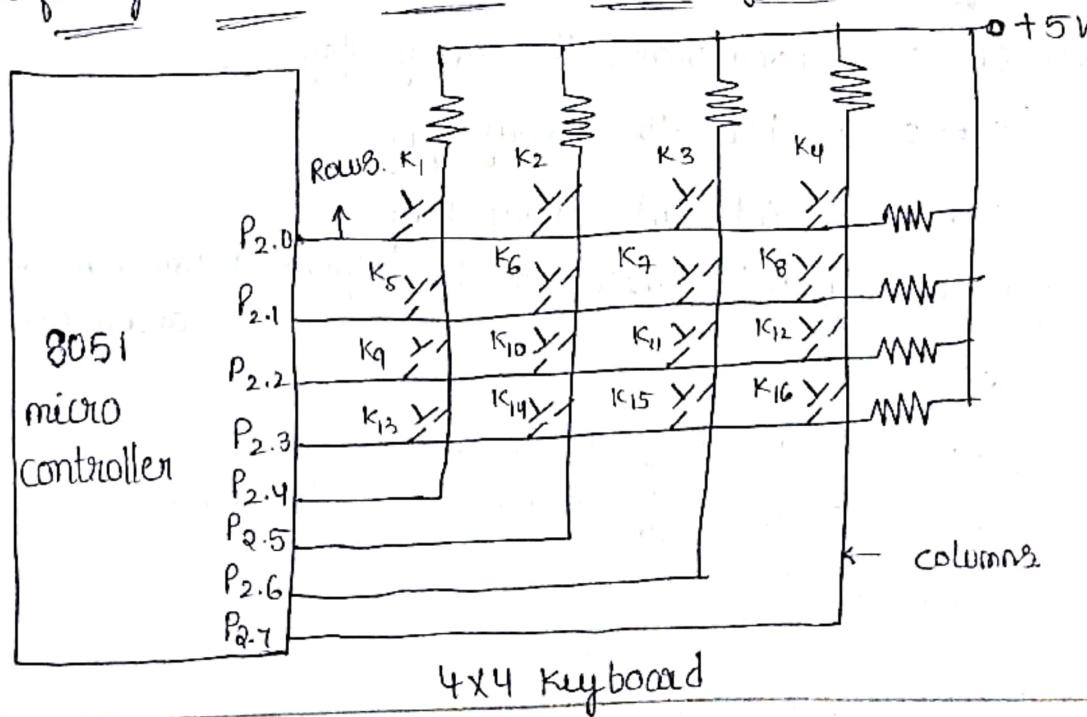
## Interfacing 8051 micro controller with LED's :-



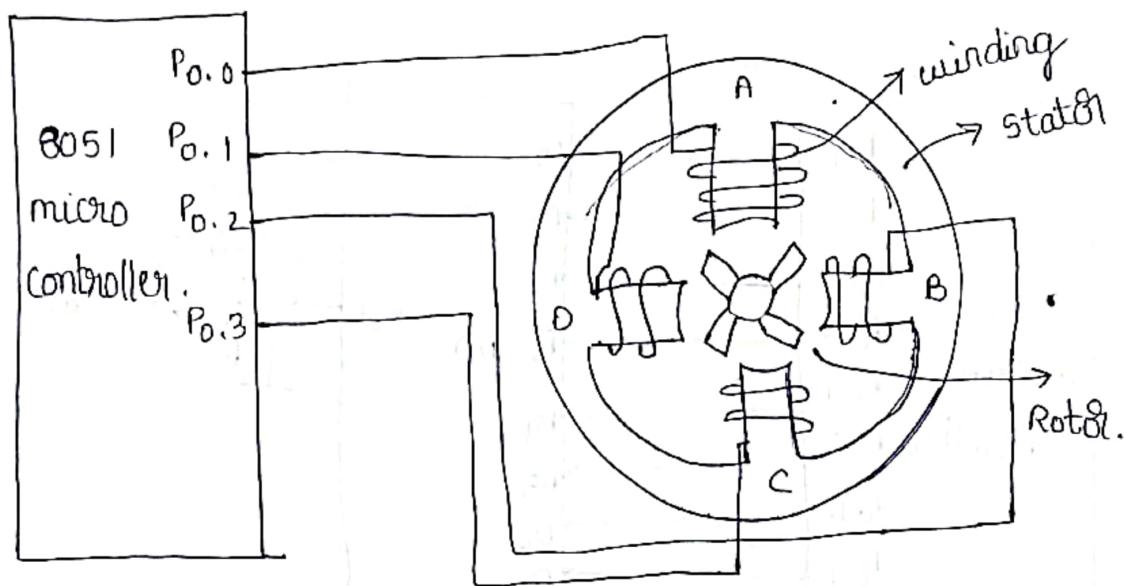
## Interfacing 7-segment display with 8051 micro controller



## Interfacing 8051 micro controller with Keyboard :-

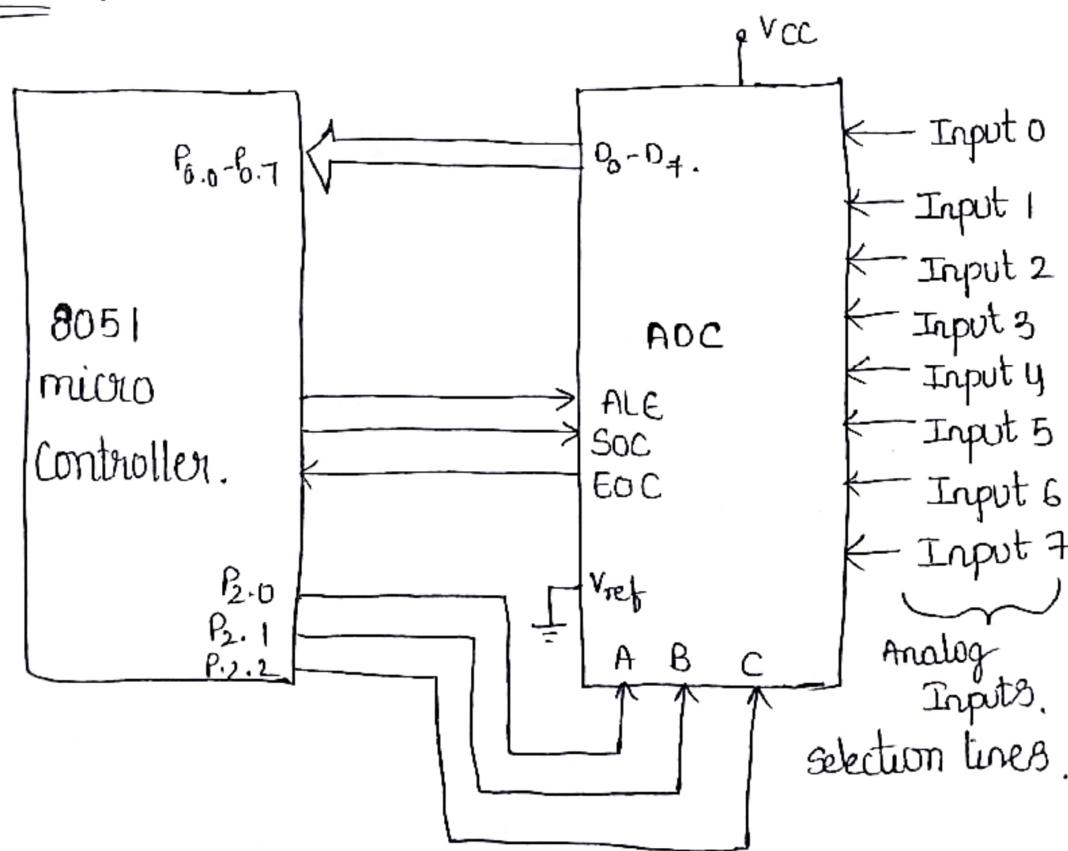


## Interfacing stepper motor with 8051 micro controller.



## Interfacing 8051 micro controller with Analog to Digital Converter

Converter :-



# Interfacing 8051 micro controller with digital to analog converter :-

