mood-book

UNIT-2

PART-1 : INTRODUCTION TO MICROCONTROLLERS :

Microcontroller :

Microcontroller is defined as an integrated circuit that contains a microprocessor along with memory, Input-output (I/o) ports, Peripherals like timer etc in a single silicon chip.

The prime use of the microcontroller is to control the operation of a system using a fixed program that is stored in ROM and that doesnot change over the lifetime of the system.

Eg 1) 8031 - intel's Microcontroller with no-on-chip EPROM

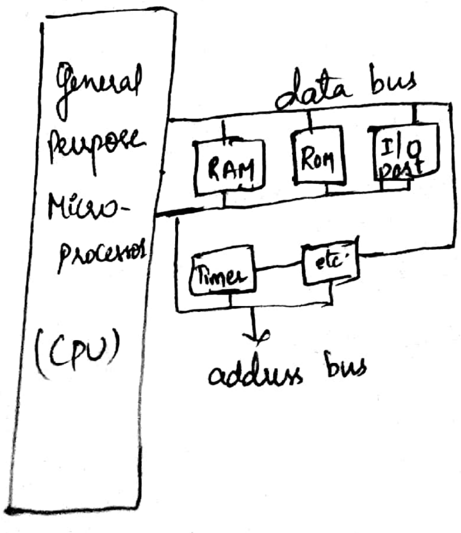
2) 8051 - Intel's Microcontroller with no-on-chip EPROM

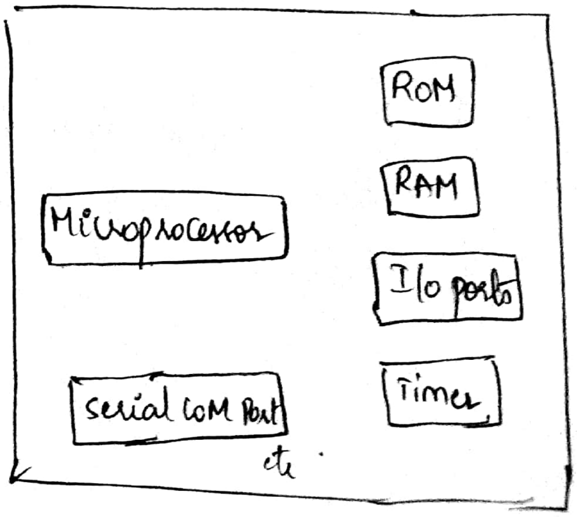
4) 8751 - Intel's Microcontroller with 4K on chip EPROM. & so on.

Difference between Microprocessor & Microcontroller :

Microprocessor	Microcontroller.
1. A General purpose Microprocessor contains → No RAM → No ROM → No I/o ports	1. Micro Controller has → Microprocessor (CPU) → RAM → ROM → I/o ports → Timer & other peripherals.
2. Must interface RAM, I/o ports and timers externally to make them functionals.	2) The fixed amount of on-chip ROM, RAM and number of I/o ports make them ideal for many applications in which cost & space are critical.

3. It has many instructions to move or transfers the data between memory and CPU	3. It has ~~no~~ only one or two instructions to move data between memory and CPU.
4. It has one or two instructions which can be used to handle bit-wise operations.	4. It has many bit handling instructions
5. It requires more hardware during the system implementation.	5. Microcontroller based system requires less hardware and hence ~~as~~ the size of the PCB decreases and thus the reliability.
6. It is more flexible in design point of view	6. It is less flexible in design point of view.
7. It has single memory map for data and code.	7. It has separate memory map for data and code.
8. Less number of multifunctional pins are present	8. More number of multifunctional pins are present
9. Access-time for memory & I/O devices is more	9. Access time for Memory & I/O devices is less.
10. ~~Archi~~ Basic Blocks of μP.	10: Basic Blocks of ~~80~~ μC





8051 Microcontroller & its features :

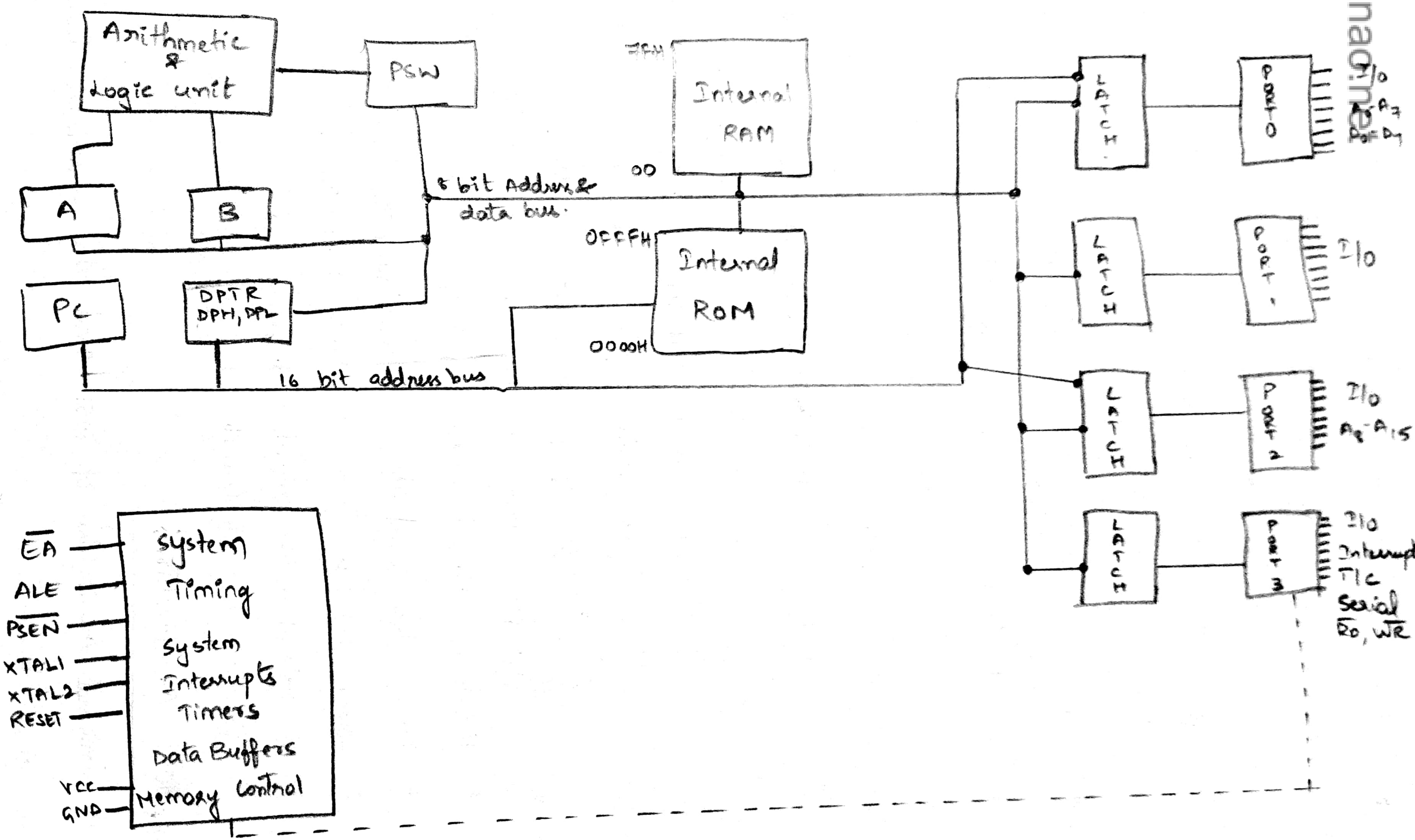
Intel has introduced 8051 Microcontroller in 1981. It is an 8-bit Microcontroller. It includes Internal ROM & RAM, I/O ports with programmable pins, Timers and counters, Serial data Communication.

The 8051 Architecture consists of the following Specific features

- * 8-bit CPU with register A (Accumulator) & B.
- * 16-bit Program Counter (PC) & data pointer (DPTR)
- * 8-bit Program Status word (PSW)
- * 8-bit Stack pointer (SP)
- * Internal ROM or EPROM) 0 (8031) to 4K (8051)
- * Internal RAM of 128 bytes
  - → 4 Register banks, Each containing eight registers
  - → 16 bytes, which may be addressed at the bit level.
  - → 80 bytes of general purpose data memory.
- * 32 I/O pin arranged as four 8-bit ports : P0-P3
- * ~~6~~ 2-16bit timers/counters : T0 & T1
- * Full duplex Serial data receiver/transmitter : SBUF
- * control registers : TCON, TMOD, SCON, PCON, IP & IE.
- * 2 External & 3-internal interrupt sources.
- * Oscillator & clock circuits.

** 8051 Architecture :

The internal Architecture of 8051 and the functional description is given below.


Arithmetic & Logic unit
PSW
A
B
PC
DPTR DPH, DPL
8 bit Address & data bus
16 bit address bus
7FH
Internal RAM
00
OFFFH
Internal ROM
0000H
LATCH
PORT 0
I/O
A0-A7
D0-D7
PORT 1
I/O
PORT 2
I/O
A8-A15
PORT 3
I/O
Interrupt
T/C
Serial
RD, WR
System Timing
System Interrupts
Timers
Data Buffers
Memory Control
EA
ALE
PSEN
XTAL1
XTAL2
RESET
VCC
GND

Accumulator: The accumulator register (A) acts as operand register which can be either implicit or specified in the instruction.

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

PSW (program status word): This set of flags contains the status information and is considered as one of the special function register.

Stack pointer: (SP): It is a 8-bit wide register which contains 8-bit stack top address. It may be defined anywhere in the on-chip 128 byte RAM.

Data pointer (DTPR): It is 16 bit register which contains the 16 bit external data RAM address. It can be accessed as 16-bit register or two 8-bit registers as DPH & DPL. It has been allotted two address in special function register bank.

Port 0 to 3 Latches & Drivers: These latches and drivers pairs are used by the user to communicate b/w I/O devices & μC.

Serial Data Buffer: It contains two independent registers one is serial transmit Buffer & the other is receive Buffer. It is used in the case of serial communication and is one of the special function register.

Timer Registers: Two 16-bit registers T0 & T1 are called as Timer registers. They can be accessed as 8 bit lower & upper bytes.

Eg TH0 represent Higher byte of Timer 0

TL1 represent Lower Byte of Timer 1

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

Timing and Control unit: This unit desires all the necessary timing & control Signals required for the internal operation of the Circuit.

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

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.

SFR Register Bank: This is a set of special function registers, which can be addressed using their respective address which lie in the range of 80H to FFH.

Register organisation of 8051:

The 8051 contains 34 general-purpose or working registers. of these 34 registers two are called as A (ACCUMULATOR) & B registers. A & B registers hold results of many instructions, like arithmetic, logical etc. The other 32 are arranged as a combination of 8 registers and as a part of internal RAM in Four Banks They are represented as B0, B1, B2, B3.. ~~All~~

All these 34 registers are 8 bit registers.

8051 has special function Register which are used to control special functions. like timers, serial communication etc.. These Special function Registers (SFR's) are allocated certain fixed Memory ~~local~~ address locations. The detailed Explanation and their addresses are shown in the table below

Name	Function	Byte Address	Purpose/usage.
A*	Accumulator	0E0H	used for holding data and status during Programming
B*	Arithmetic operations	0F0H	
PSW*	Program status word	0D0H	
SP	stack pointer	81H	used in instructions to point to memory
DPL	Lower Byte address of External memory	82H	
DPH	Higher Byte address of External memory	83H	
P0*	I/O PORT Latch	80H	used by the respective I/O ports
P1*	I/O Port Latch	90H	
P2*	I/O Port Latch	0A0H	
P3*	I/O Port Latch	0B0H	
SCON*	Serial port Control	98H	used by the Serial port
SBUF	serial port Data Buffer	99H	
TCON*	Timer/Counter Control	88H	used for Timer Control
TMOD	Timer/Counter mode Control	89H	
TL0	Timer 0 Lower Byte	8AH	
TL1	Timer 1 Lower Byte	8BH	
TH0	Timer 0 Higher Byte	8CH	
TH1	Timer 1 Higher Byte	8DH	
IE*	Interrupt Enable	0A8H	used for Interrupt control.
IP*	Interrupt Priority	0B8H	
PCON	Power Control	87H	used for power control

The SFR's which are marked with $\boxed{*}$ are bit-addressable.

Program Status Word (PSW)

Program Status Word is an 8-bit register. It is also Called as "Flag register" as it mainly contains the status flags. These flags indicate Status of the Current result. The Contents of the PSW ~~Co~~ will be changed by the ALU After each arithmetic & logic application. The flags can also be changed by the programmer. PSW is a bit-addressable register ie each bit Can be individually Set or reset by the programmer. The bits Can be refered to by their bit numbers or by their name.

Eg: PSW.4 (or) RS1 means the Same field of the PSW register. ~~The~~

~~&~~ The different fields of PSW are shown below.

PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
CY	AC	FO	RS1	RS0	OV	–	P

CY- Carry Flag:

It indicates the carry out of MSB, after any arithmetic operation.

If CY = 1 ; Carry is generated out of MSB

If CY = 0 ; no Carry ~~is~~

AC- Auxillary Carry Flag:

It indicates that there is a Carry out of the lower nibble

If AC = 1 ; Carry is generated out of lower nibble.

If AC = 0 ; Carry is not generated.

F0 - user defined flag:

It Can be used by the programmer to Store any user defined inform

This flag can be changed by simple instructions SETB & CLR.

RS1, RS0 - Register Bank Select:

The initial ~~to~~ 32 locations of Bytes the internal RAM are available to the programmer as register. If the number of registers are more then the opcodes also will be more and this is the main reason

If the 8051 needs to access internal ROM then $\overline{EA} = 0$.

why the registers are divided into register banks. At a time, only one of register banks are active and they can be selected by RS1 & RS0

RS [Register bank Select]

RS1	RS0	Register Bank.
0	0	Bank 0
0	1	Bank 1
1	0	Bank 2
1	1	Bank 3

OV- overflow flag:

overflow flag indicates if there was an overflow during a signed operation.

If OV=1 ; overflow in the result.

If OV=0 ; No overflow in the result.

P- Parity flag:

Parity flag indicates the even parity or odd parity in the result.

If P= 1 ; It indicates odd Parity

If P= 0 ; It indicates even Parity

Memory organisation of 8051:

The 8051 µC Supports Internal memory as well as External memory. (RAM & ROM). 8051 ~~can b~~ is designed on Harvard Architecture & hence it store Program & data in two different Spaces.

1) Programs will b stored in ROM as it is permanent

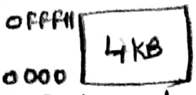
2) data will be stored in RAM as it is volatile

ROM organisation / Program Memory

ROM can be organised as a combination of Internal & External memory. and can be ~~d~~ used as

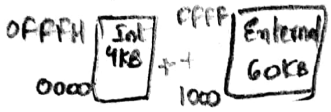
1) Internal ROM (4KBytes)
2) Internal ROM (4KB) + External ROM (60KB)
3) only External ROM (64KB)

Case-(i)



8051 has 4KBytes of Internal ROM. If the programmers wants to use only internal ROM then $\overline{EA}$ pin of 8051 must be = 1 [$\overline{EA}$ = 1] The range of address will be from 0000H to 0FFFH. all the oth addresses are invalid

Case-(ii) I



8051 has 4KB of Internal ROM along with this if the External R needs to be accessed then the maximum size of memory that can be accessed is 60KB. In this case also $\overline{EA}$ = 1. The range of addr for Internal ROM = 0000H to 0FFF H.

External ROM = 1000H to FFFF H

If the address is < 1000H, the operation will be in the internal ROM oth wise the operation will be in the External ROM.

Case-(iii)

FFFFH
64KB
0000H

only External ROM

8051 can be interfaced with External ROM and the maximum size of the memory that can be accessed is 64KBytes. If the μc (8051) wants to ac only External ROM then $\overline{EA}$ ~~pin~~ = 0 ie the External Access pin must be grounded. The valid address Range in this case is 0000H to FFFFH.

RAM organisation / Data memory:

8051 has a 128 Bytes of Internal RAM The address range is 00H to 7FH. The RAM is used for storing data. It is divided into three parts: Register Banks, Bit addressable area, General area

The first 32 Bytes of the internal RAM from 00H to 1FH are used for register Banks. Each bank has 8 registers $R_0, R_1 \cdots R_7$. A register can be addressed by its name or by its address.

Eg 00H to 07H Contains registers R_0 to R_7 of Bank0
08H to 0FH contains registers R_0 to R_7 of Bank 1
10H to 17H Contains register R_0 to R_7 of Bank 2
18H to 1FH Contains registers R_0 to R_7 of Bank 3

The Register Banks can be Selected using PSW register (RS1, RS0)

~~Register-eB~~

Bit Addressable Area:

The next 16-bytes of RAM from 20H to 2FH are Bit addressable Area. These bits can be addressed using their individual address 00H...7FH. If the programmer wants to perform Byte operation in this area, then 20H to 2FH addresses must be used. ie 20H will be a byte address, 21H will be the next byte address.

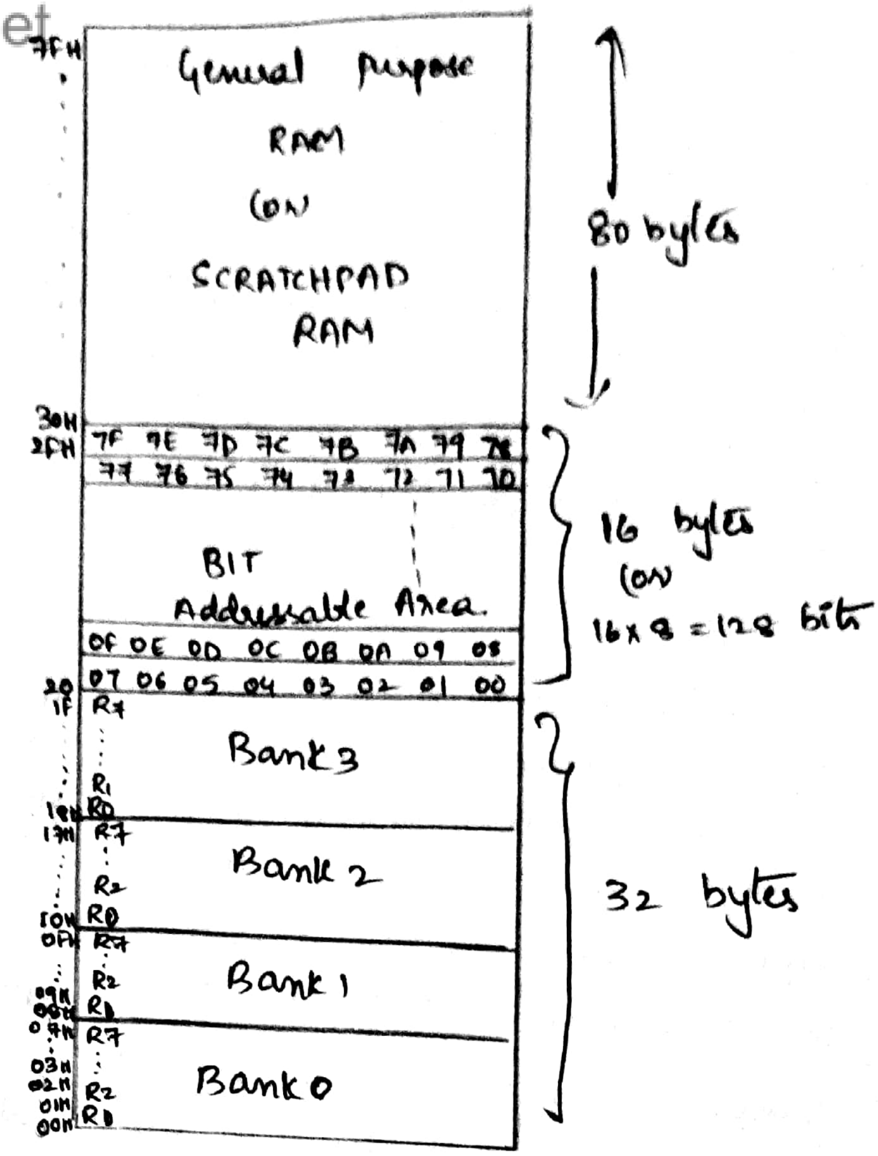
If the programmer wants to use bit addresses then they have to use addresses 00, 01, 02.... 7FH. The ~~μp~~ μicro Controller will check the ~~address and~~ instruction and then will understand whether it should pick bit or byte amount of data.

Eg 1) SETB 00H ; The Instruction is a bit Operation
It will make bit location 00H to "1"

2) MOV A, 00H ; This is a Byte operation
The Accumulator will get 8-bit data from Byte location 00H ~~which is~~

General Purpose Area:

The general purpose area ranges from location 30H to ~~7~~7FH. This is an 80-byte area which can be used for general data Storage.



Stack of 8051:

The stack is a set of memory locations operating in last in first out manner. It is used to store return addresses during ISRs and also used to store data during programs. In 8051, the stack can only be present in the internal RAM. It is because SP is an 8-bit register and can only contain an 8-bit address and External RAM has 16-bit address

The reset value of SP is 07H because on the first push, SP gets incremented and then data is pushed on to the stack. This means the very first data will be stored ~~at~~ location 08H. This does not affect the default bank (Bank0) ~~and Stillgives the stack~~.

The programmer can relocate the stack to any desired location by simply loading a new value into SP register

If the 8051 needs to access internal ROM then $\overline{EA} = 0$.

Pin 30 : ALE/PROG : The Address latch enable pins indicates that the valid address bits are available on their respective pins. This ALE signal is valid only for External memory access.

Pin 29 : $\overline{PSEN}$: Program store enable is an active - low signal that acts as a strobe to read the external program memory. It is '0' during External program memory acces.

Pins (39-32) : P0.0 to P0.7 : Port-0 is an 8-bit bidirectional bit addressable I/o port. ie it can by accessed as individual i/o pin. Port-0 also has an alternate function. It carries multiplexed address & data lines. It carries lower byte of address

If ALE = 1, the bus carries address (A_0-A_7)

· If ALE = 0, the bus carries data (D_0-D_7)

Pins 1-8 : P1.0 - P1.7 : Port 1 acts as an 8-bit bidirectional bit addressable Port. Port 1 doesnot have any alternate functions.

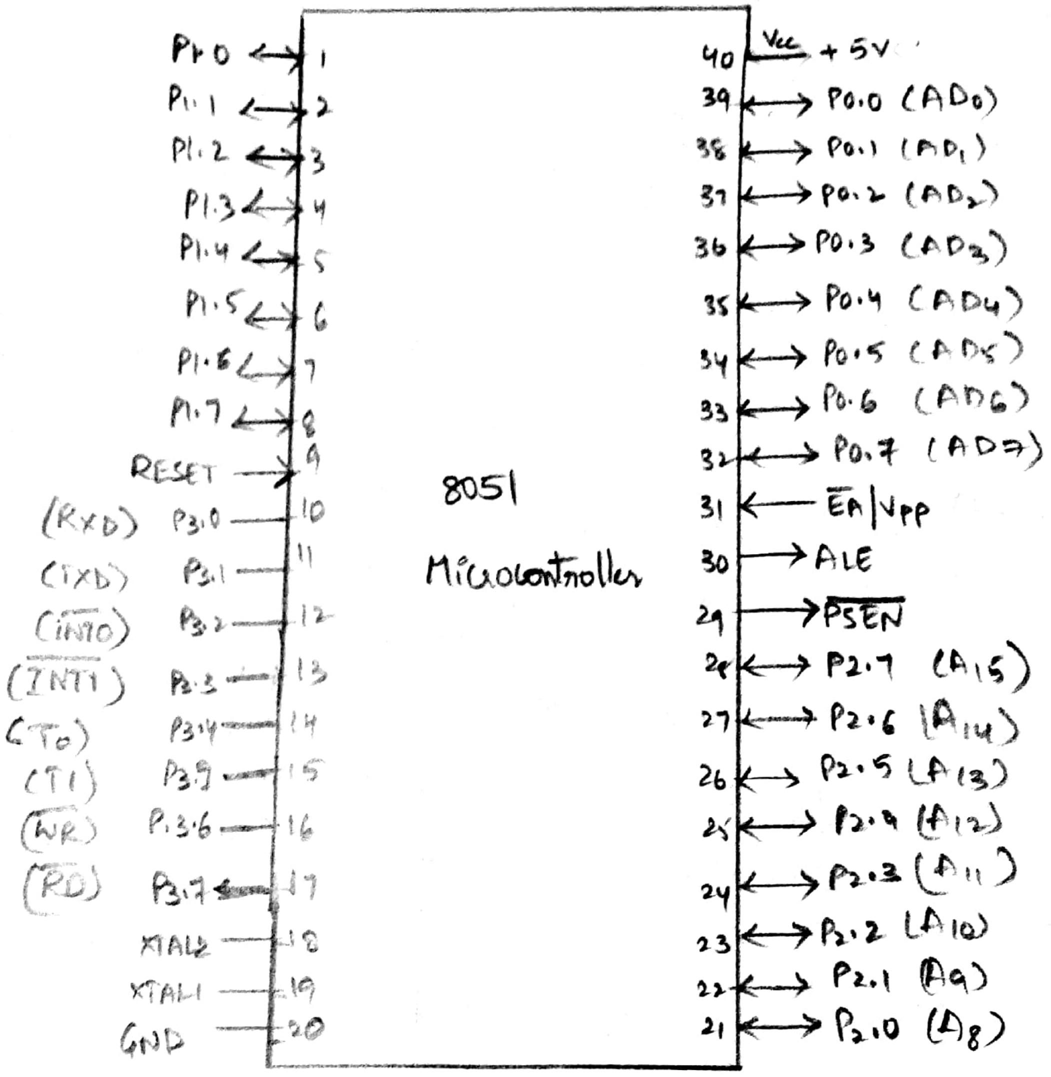
Pins 28-21 : P2.0 - P2.7 : Port 2 acts as an 8-bit bidirectional bit addressable I/o port. During external memory access, Port-2 emits higher eight bits of address (A_8-A_{15}) - This is the alternate function of Port-2

Pins 10-17 : P3.0 - P3.7 : Port-3 is an 8-bit bidirectional bit addressable I/o port. It also serves the alternative functions as shown in the table

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 pin 1 ($\overline{INT1}$)
P3.4	Acts as external input to timer 0 (T_0)
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 ($\overline{RD}$)

Pin/Signal Description of 8051 Micro-Controller :

8051 is available in a 40-pin plastic and ceramic DIP packages. The pin description of 8051 is as follows.



V_{cc} : (pin-40): +5V Supply Voltage

V_{ss} : (pin-20): Ground Voltage

Pin-9-Reset: A logic 1 on this pin erases/clears the Contents of Most of Registers and only when it is for two or more machine cycles.

Pin-18, 19: $XTAL_1$ & $XTAL_2$: $XTAL_1$ & $XTAL_2$ are the inputs & output of the respectively and a crystal oscillator is connected externally between these two pin which Specifies the operating frequency. usually the operating frequency of 8051 12MHz to 16MHz frequency.

Pin-31 : $\overline{EA}$/Vpp : External access enable pin. It indicates that the 8051 address external program memory. That means if $\overline{EA} = 0$ then 8051 uses only External Program memory.

ADDRESSING MODES OF 8051:

¥. 8051 supports the different types of addressing modes and they are as follows
1) Immediate addressing mode
2) Register addressing mode
3) Direct addressing mode
4) Indirect Addressing mode
5) Indexed addressing mode.

1. Immediate addressing mode:

In this addressing mode, the instruction contains the immediate data as as operand. The destination must be an register and the source will be the immediate data. The immediate data is used along with '#' symbol to inform the controller that it is not an address but a data.

Eg 1) MOV A, #35H ; A ← 35H
2) MOV DPTR, #3000H ; DPTR ← 3000H

It is important that the size of the source & destination must be same

2. Register Addressing Mode:

In this addressing mode, DATA is given by the register in the instruction. T Permitted registers are A, R7.. R0 of each memory bank. The data transfer between two RAM registers is not allowed

Eg 1) MOV A, R0 ; A ← R0
2) MOV R5, A ; R_5 ← A
3) MOV R_x, R_y ; invalid as both of them are a part of RAM
; here x & y can be any value from 0-7

3. Direct Addressing mode:

In this addressing mode, the operands are specified using 8 bit address field in the instruction only. The internal data RAM & SF are addressed directly.

Eg: MOV R0, 89H

here the contents present at the address 89H is copied into register. Here the 89H is the address of the SFR TMOD.

4. Indirect addressing mode :

Here the address of the operand is given in a register. Internal RAM and External RAM. Can be accessed using this mode.

Case-(i) : Internal RAM (8-bit address given by R0 & R1)

only R1 & R0 Can be called as Data pointer and can be used to Specify address. An @ sign is used before register to indicate that the register is ~~g~~ carrying an address.

Eg 1) MOV A, @R0 ; A ← [R0]

; A ← contents of internal RAM Location whose address is given by R0

Case-(ii) External RAM (16 bit address by DPTR & 8 bit address by R0 & R1)

for the external RAM, address is provided by R1 or R0 or by DPTR to indicate External RAM the symbol 'x' is present in the instruction

Eg MOV X A, @ DPTR ; A ← (DPTR)

; A gets the contents of External RAM loc whose address is given by the register DPTR.

5) Indexed Addressing mode :

This mode is used to access data from the Code Memory i.e either Internal Rom or External Rom. In this addressing mode, ad is indirectly specified as a sum of (A & DPTR) or (A & PC). to indicate Code memor here 'c' is present in such Instruction

Eg 1) MOVC A, @DPTR + A ; A ← contents of ROM pointe A + DPTR. If DPTR = 0400H & A = 05H, then go A g the contents of ROM whose address is 0405 H.

=0 then it means that the

Programming Timers/counters of 8051:

8051 has two, 16 bit Timers/counters T1 & T0. If the counter is programmed to count internal clock pulses then it is called as timer and if the counter is programmed to count external clock pulses it is called as counter.

The Counters/Timers are divided into two 8 bit registers called as TL0, TL1, TH0 & TH1.

The Timer/counter action is controller by the bits in the "Timer mode control register" (TMOD) and the timer/counter is controlled by TCON register which is "Timer/counter control register".

TCON Register organisation:

TCON has two parts

1) upper nibble contains → control bits & flags for the timers/counte

2) Lower nibble contains → control bits & flags for External Interru

The various fields of TCON register is as follows.

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

where the function is

TF1/TF0 → Timer 1/0 overflow flag.

TF1 = 1 means the timer 1 or timer 0 overflows respectively ie its bits roll over from all 1's to all 0's.

→ This bit is cleared when the processor executes the ISR.

TR1 & TR0 : Timer Run control bit.

If TR1 or TR0 = 1 then the timer 1 or Timer 0 starts counting.

TR1 or TR0 = 0 then Timer 1 or Timer 0 stops counting.

IE1 and IE0 : External interrupt flag.

If IE1 or IE0 = 1 then it means that an external interrupt l

occurred on $\overline{INT1}$ or $\overline{INT0}$ respectively.

IE1 or IE0 = 0 then it means that the interrupts are servic

IT1: External interrupt 1 Signal type control bit
It is set to 1 by the programmer to enable external interrupt $\overline{INT1}$ to be a edge triggered signal. If it is set to 0 by the programmer then it is said to be level trigger Signal interrupt. Basically it is used to decide the type of the interrupt that can trigger the Micro Controller.

IT0: It decides the type of external interrupt to be as a edge triggered or level triggerd type same as in the case of IT1.

TMOD - Timer Mode control register:

7	6	5	4	3	2	1	0
GATE	$C/\overline{T_1}$	M_1	MO	GATE	$C/\overline{T_0}$	M_1	Mo

The timers 0 & Timer 1 both used the same TMOD register. It is an 8 bit register where the lower 4 bits belong to timer 0 & upper 4 bits belong to timer 1.

The TMOD register fields are as shown above

GATE: gating control - ~~when~~ This bit allows the programmer to control the counter by Hardware interrupt connected to the External interrupts of Port 3. ie the time/counter can start or stop using this bit field using External Hardware.

$C/\overline{T}$: counter/Timer:

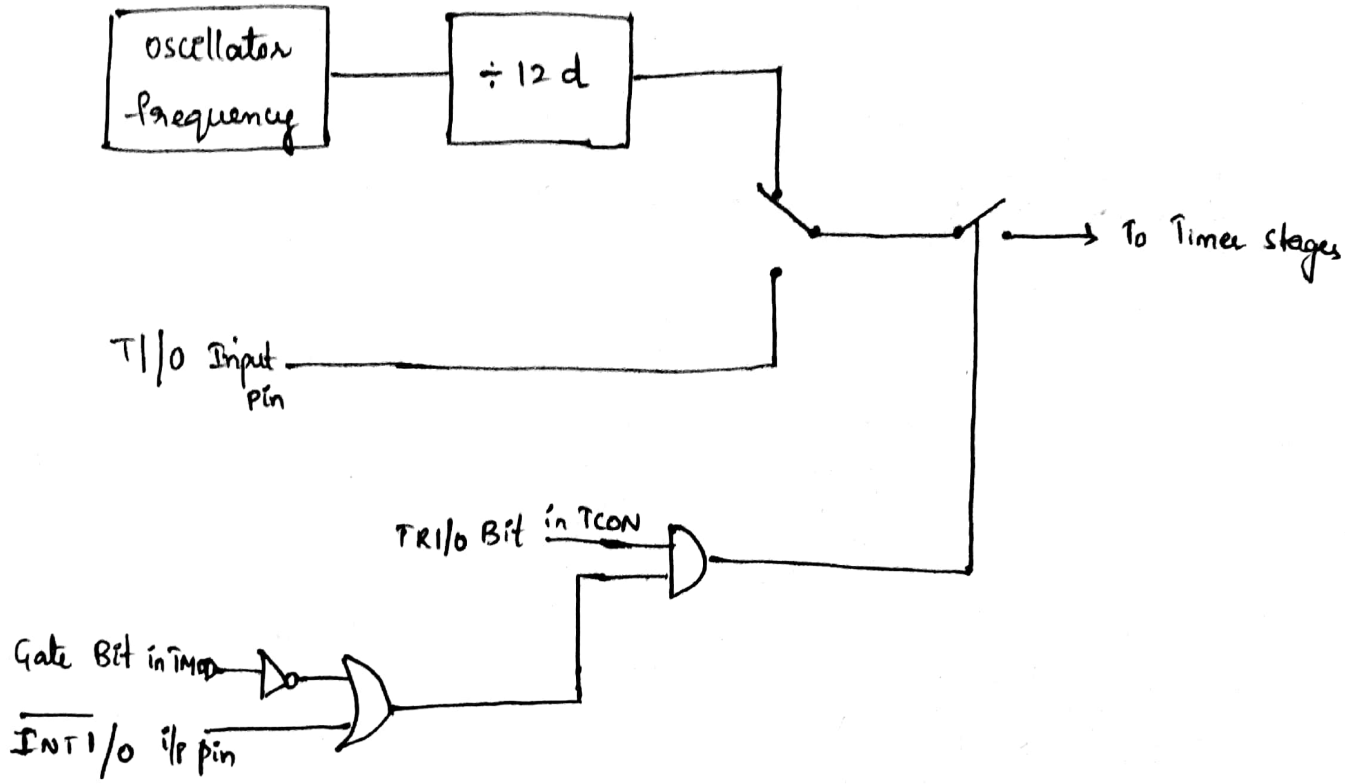
If this bit is set by the programmer, Timer 1/0 acts as an counter that takes the clock pulses from external input pins To or T1. If $C/\overline{T}$ is cleared to 0 by the programmer it ~~is call~~ acts as timer which takes the internal clock puls

M1, MO : Mode select bits:

It is used to select the timer modes. The timer ~~has~~ Can work in three modes. The Combination of M1 & MO will be use ~~Mode~~ for the purpose of selecting the modes of the timer.

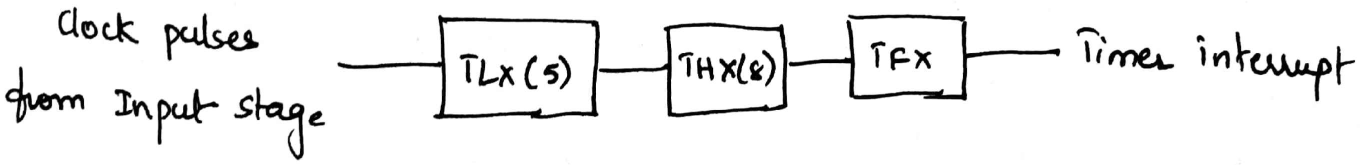
M	M	Timer Mode	
0	0	Mode 0	13-bit Timer
0	1	Mode 1	16-bit T/C
1	0	Mode 2	Auto reload
1	1	Mode 3	Two 8-bit timers Timer-0

Timer/counter Logic :



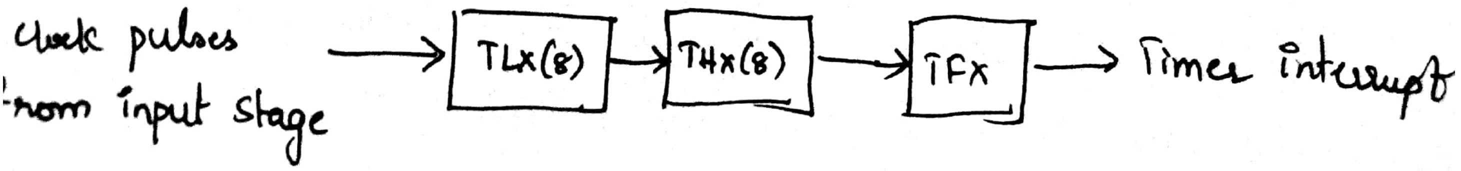
Timer Modes :

a) Timer Mode 0 (13-bit Timer/counter)



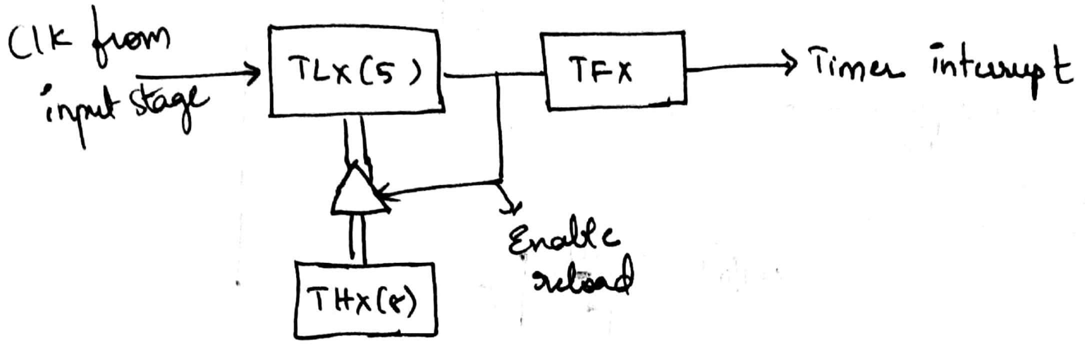
In time mode 0, THX is an 8-bit Count & TLX is a 5 bit Preset Total 13 bits are used for Counting. on each count the TLX increments & each time TLX reaches the Max value THX incremen by 1 bit. Thus the time Overflow flag overflows only when THX overflows. The max delay Produced is $2^{13} \times \frac{12}{fosc}$

b) Timer Mode 1 (16-bit timer/counter)



All the 16 bits of ~~E~~ Counters are used. on Each count the timer increments & The TFX flag is set only when the timer rolls-over from FFFF H to 0000 H. the max delay that can be produced is $2^{16} \times \left(\frac{12}{f_{osc}}\right)$

c) Timer Mode 2 (Auto reload TL ~~&~~ from TH)

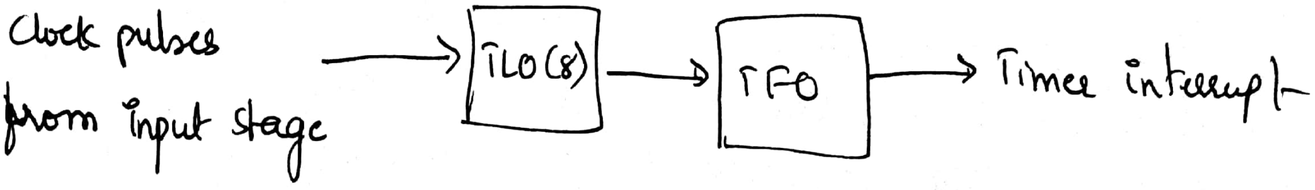


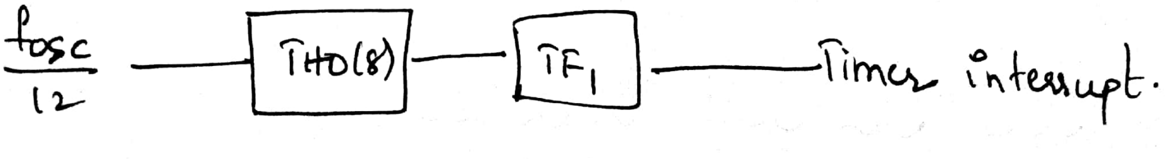
TLX is used as an 8-bit Counter. THX uses the Count value to be reloaded. on each count TLX increments. when TLX rolls over

1) Timer overflow flag TFX is set.
2) THX is copied into TLX. Hence TLX is Auto reloaded & so on
3) This process occurs continuously

Maximum delay that can be produced = $2^{8}\left(\frac{12}{f_{osc}}\right)$

d) Timer Mode 3 (Two 8-bit timers using Timer 0)





Timer 0 is used as 2 Separate 8-bit timers TH0 & TL0. The TL0 uses the control bits (TR0 & TF0) of timer 0. It can work as both timer or a Counter.

TH0 uses the control bits (TR1 & TF1) of Timer 1. It Can WORK only as a timer.

Ti~~mer 1~~ ~~Can be~~