Module 1

Introduction

Introduction to MicroProcessors, Microcontroller & Microcomputer:

In 1971, Intel Corporation released the world's First Micro-processor - INTEL 4004, a 4-bit Microprocessor. It addresses 4096 memory locations of word size 4 bit. The instructions set consists of 45 different instructions.

First generation processors requires a lot of additional support IC's to form a system. It may require as high as 30 IC's to form a system. The 4-bit processors are provided with only 16 pins, but 8-bit & 16-bit processors are provided with 40 pins. Due to this limitation of pins, signals are multiplexed.

A by-product of microprocessor developement was the micro-contr-oller. The same fabrication techniques & programming concepts that make possible the general purpose microprocessor also yielded micro controllers.

Binary digit : Bit is the abbreviation of Binary digit

Bit : Binary numbers 0 (or) 1 is known as bit

Byte : A group of eight bits is known as byte

Nibble: A group of four bits is known as nibble

Data/Word: It is formed by combining number of bits for a given microprocessor.

Data length/ Word length : It is defined as the number of bits, the microprocessor can recognize & process at a time.

Example : Intel 8085 has word length of 8 bit

Intel 8086 —"— 16 bit

Pentium —"— 32 bit

Dual core Processor —"— 64 bit.

Other companies microprocessor are: FAIRCHILD F-8; MOTOROLA M6800; NATIONAL CMP-8, SIGNETICS 2650, ZILOG Z80 — All are 8-bit microprocessor.

Hardware: The arrangement of physical components such as µp, memory, input & output devices is known as hardware

Program: A set of instructions written for µp to perform a task is called a program.

Software: A group of program is known as software.

Microprocessor, Micro-computer & Micro controller:

Microprocessor: A microprocessor is a programmable integrated device that has computing & decision making capability.

A microprocessor is a multipurpose, programmable, clock-driven, register based electronic device that reads binary instructions from a storage device called memory, accepts binary data as input & processes data according to those instructions & provides results as output.

Microprocessor (MP) mainly consist of control unit, ALU (Arithmetic Logic Unit) & array of registers. MP is some time referred as CPU

* Fig (a) shows the MP (or) CPU consisting of ALU, control unit & Register array.
* MP (or) CPU consists of various registers to store data, the ALU to perform arithmetic & logic operations, instruction decoders, counters and control lines.
* The CPU communicates & operates in the binary numbers 0 & 1
* Timing of the communication process is controlled by the group of circuits called the control unit

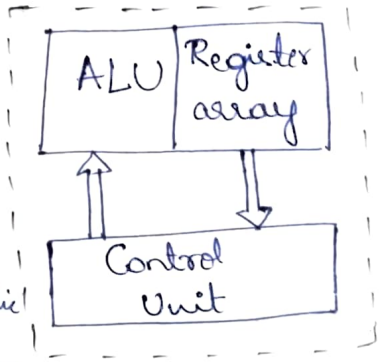


Fig. (a)

Ex: INTEL-8085

* A microcomputer is a compact integrated circuit designed to govern a specific operation in an embedded systems
* Fig (b) shows the microprocessor based system known as microcomputer.
* It consists microprocessor, input, output & memory.
* The microprocessor reads instructions from the memory & performs the tasks specified. Memory is used to store instructions & data.

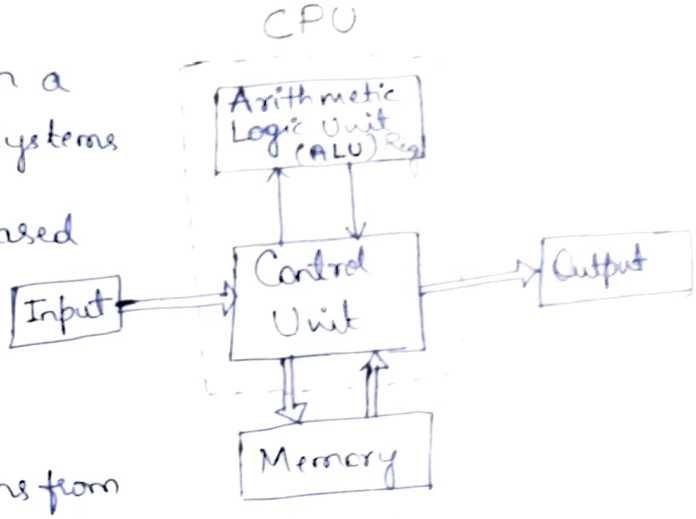


Fig (b)

* MP communicate with input/output devices either to accept (or) to send data
* Fig (b) can be replaced with fig (c). as CPU can also be referred as MP

Ex: PC, MP kits

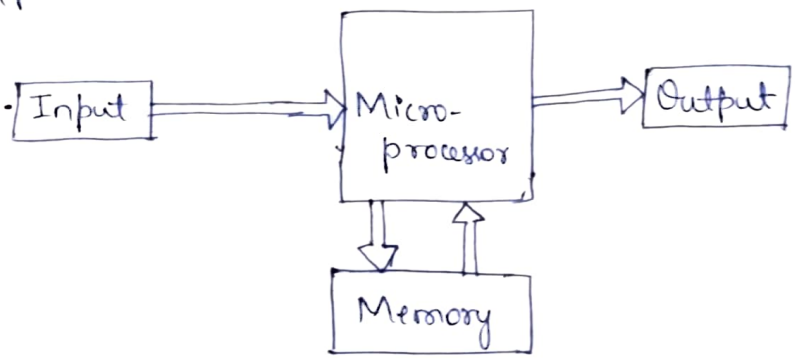


Fig (c).

* Micro Controller is a computer on a single chip consisting of CPU, memory, I/O ports & other peripherals
* As the technology advances, manufacturers are able to place CPU, memory, I/O interfacing circuit on a single chip & it is known as MC.
* A micro controller is essentially an entire computer on a single chip.
* Fig (d) shows microcontroller chip.

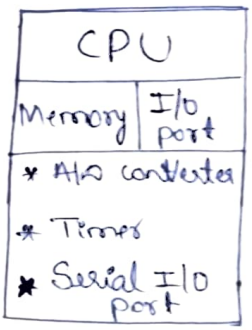


Fig (d)

Ex: INTEL 8051, PIC 16C74, Motorola 68HC11

Organization of a Microprocessor:

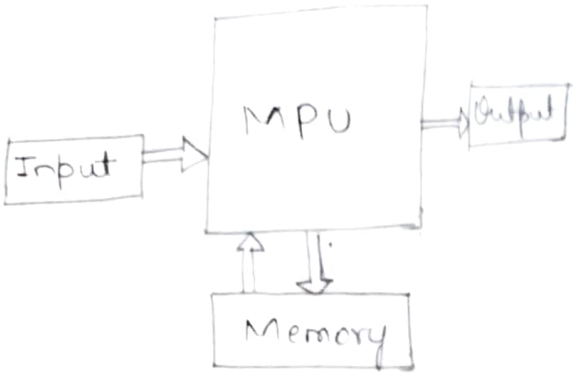


Fig 1(a) Simple block diagram

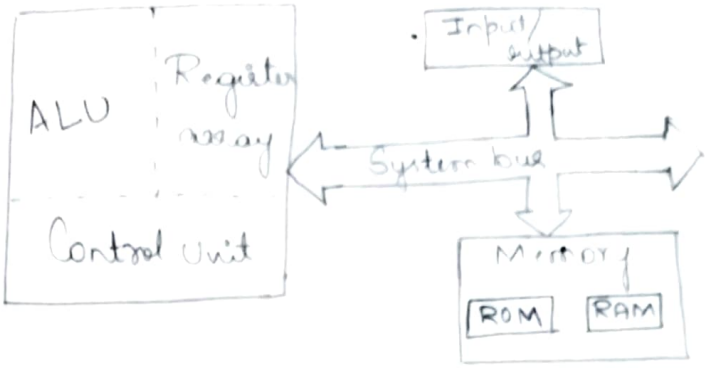


Fig 1(b). Block diagram with system bus

* Fig 1(b) shows simplified structure of a microprocessor organization
* It includes 3 components:
  (a) Microprocessor (b) Input/Output (c) Memory

These components are organized around a common communication path called a bus. The term peripherals is used for i/p/o/p devices.

(a) Microprocessor: It is a programmable, clock driven integrated electronic device & is used to perform the given tasks. It is capable of
* performing various computing functions
* Making decisions to change the sequence of program execution.

The µp can be subdivided into three segments as shown in figure 1(b)
(i) ALU (ii) Register array (iii) Control Unit (CU)

(i) ALU: The ALU unit performs arithmetic operations & logical operations.

(ii) Register array: These are used within CPU to store data that is actively being processed by processor. These registers holds operands for arithmetic & logic operations, intermediate results and addresses for memory operations.

(iii) CU: It provides the necessary timing & control signals to all the operations in µp. It controls flow of data b/w µp, memory & peripherals.

(b) Memory: Memory is made up of semiconductor materials to store instructions & data. µp reads instruction & data from

memory & performs the computing operations in its ALU section
There are 2 types of memory : (i) ROM (ii) RAM (R/W memory)

* ROM is used to store programs that do not need alterations
* R/W memory is also known as user memory. It is used to store user programs & data.

RAM → SRAM, DRAM

ROM → PROM, EPROM, EEPROM, FLASH

(c) Input / Output : The user can enter instructions & data into memory through input device such as keyboard, switches etc. The result can be displayed by a output device such as seven segment LED's, CRT, Video screen, Printer, Magnetic tape etc

(d) System bus : The system bus is a communication path b/w the µp & peripherals & memory. Bus is nothing but a group of wires to carry bits. System bus can be further divided into
→ Data bus → Address bus → Control signals

Data bus : Data bus is a group of communication path b/w µC & peripherals / memory to transfer data b/w them

Address bus : µC communicate with one of two peripherals (or) memory at any instant of time. It communicate with the peripherals (or) memory using unique address. So this address bus is a group of communication path that carry address from µC to peripheral (or) memory locations. (identifies one)

Control signals : µC uses certain signals to communicate with peripherals / memory are known as control signals.

Comparison b/w Microprocessors & Microcontrollers

Microprocessors	Microcontrollers
* MP is an IC with only CPU (ALU & control unit)	* MC has a CPU in addition with RAM, ROM & other peripherals embedded on a single chip.
* MP's are designed to perform different tasks for unspecific applications	* MC's are designed for dedicated applications most of the time.

* Memory & I/O components have to be connected externally
* The clock speed of MP is high
* MP system is much costlier
* Since memory & I/o has to be connected externally, the circuit becomes large
* Power consumption is large & hence cannot be operated with battery
* Has less number of registers, hence more operations are memory based.
* MP are based on Von-Neumann model where program & data are stored in same memory
* Mainly used in PC's

* Ex: INTEL 8085
INTEL 8086
INTEL PENTIUM
MOTORALA 6800
Zilog 80
* The architecture & instruction set are designed to handle byte size
* Many move instructions for data move from external memory to CPU & few (one @ Two) types of instructions to handle bit.

* Memory & I/O components are inbuilt
* Clock speed is low
* MC system is cheaper
* Since memory is inbuilt, circuit is small

* Power consumption is low & can be operated with battery.
* Have more number of registers.

* MC are mostly based on Harvard architecture where program memory & data memory are separate
* Mainly used in embedded sys such as: Washing mic, Cars, Printers, Toys; Calculator, Video games etc.
Ex: INTEL 8051
PIC16F887, PIC16C86, Z-8
MOTOROLA 6812

* The architecture & instruction set are optimized to handle data in bit & byte size.
* Many instructions are present to handle bit operations & few instructions are to move data from external memory to CPU

Microprocessor	Micro Computer
* It is a single chip	* It is the arrangement of more than one chip
* ALU is the heart of MP	* MPU is the heart of micro computer
* MP consists of ALU, control & timing circuit, register arrays as internal ports	* It is a system consisting of MPU, RAM, ROM, I/O devices.
* It has internal system bus	* It has system bus (Address bus, Data bus & control bus)
* MP has registers as storage devices	* It has RAM memory & ROM as storage device in addition to registers
* MP cannot alone work as a system	* It can work alone as a system.

8051 - Micro controller

* INTEL 4004 was first 4-bit processor developed in 1971.
* In 1974, Texas instrument introduced the first mc TMS1000
* A brief overview of some commercial micro controllers:
  INTEL, PIC, MOTOROLA, ATMEL

General Functional Block diagram of Micro controller

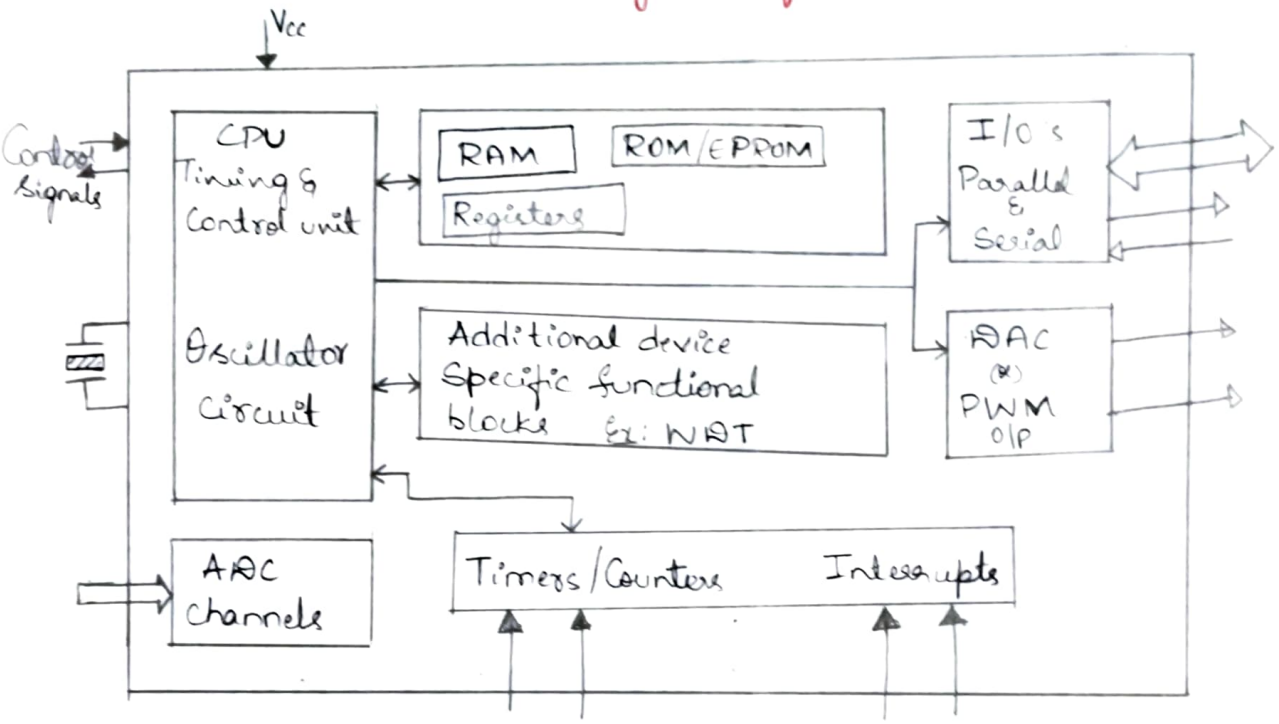


Fig. 2

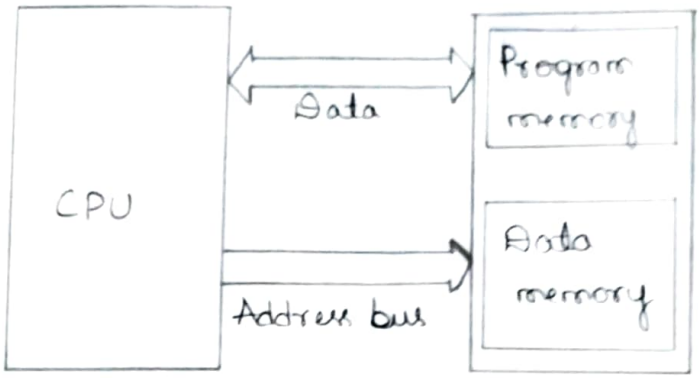
RISC & CISC CPU Architecture

* MC with small instruction set are called reduced instruction set computer (RISC) machines & those with complex instruction set are called CISC machine whereas micro chip PIC 18787X is an example of RISC machine

RISC	CISC
* Instructions takes one (or) two cycles	* Takes multiple cycles
* Only load/store instructions are used to access memory	* In addition to load & store instructions memory access is possible with other instructions also.
* Instructions executed by hardware	* Executed by micro-program.
* Fixed format instructions & few instructions	* Variable format instructions & complex instructions
* Few addressing modes & highly pipelined	* Many addressing modes & less pipelined
* Most of them have multiple register banks	* Single register bank
* Complexity is in compiler	* Complexity is in microprogram.

Harvard & Von-Neumann CPU Architecture :

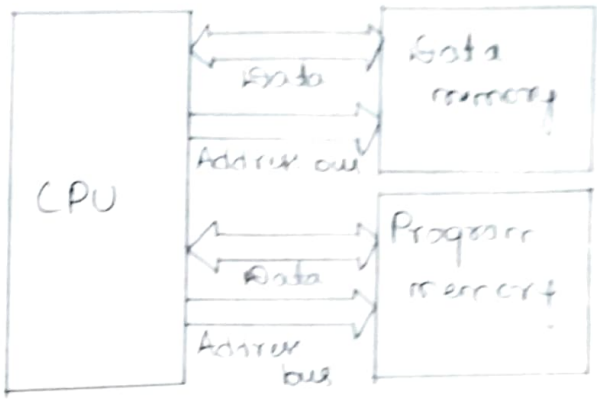
Von-Neumann (Princeton)



* It uses single memory space for both instruction & data
* It is not possible to fetch instruction code & data simultaneously.
* Execution of instruction takes more machine cycle.
* Uses CISC architecture
* Instruction pre-fetching is main feature
* Also known as control flow (or) control driven computers
* Simplifies the chip design because of single memory space

Ex: 8085, 8086, MC6800, MOTOROLA 68HC11

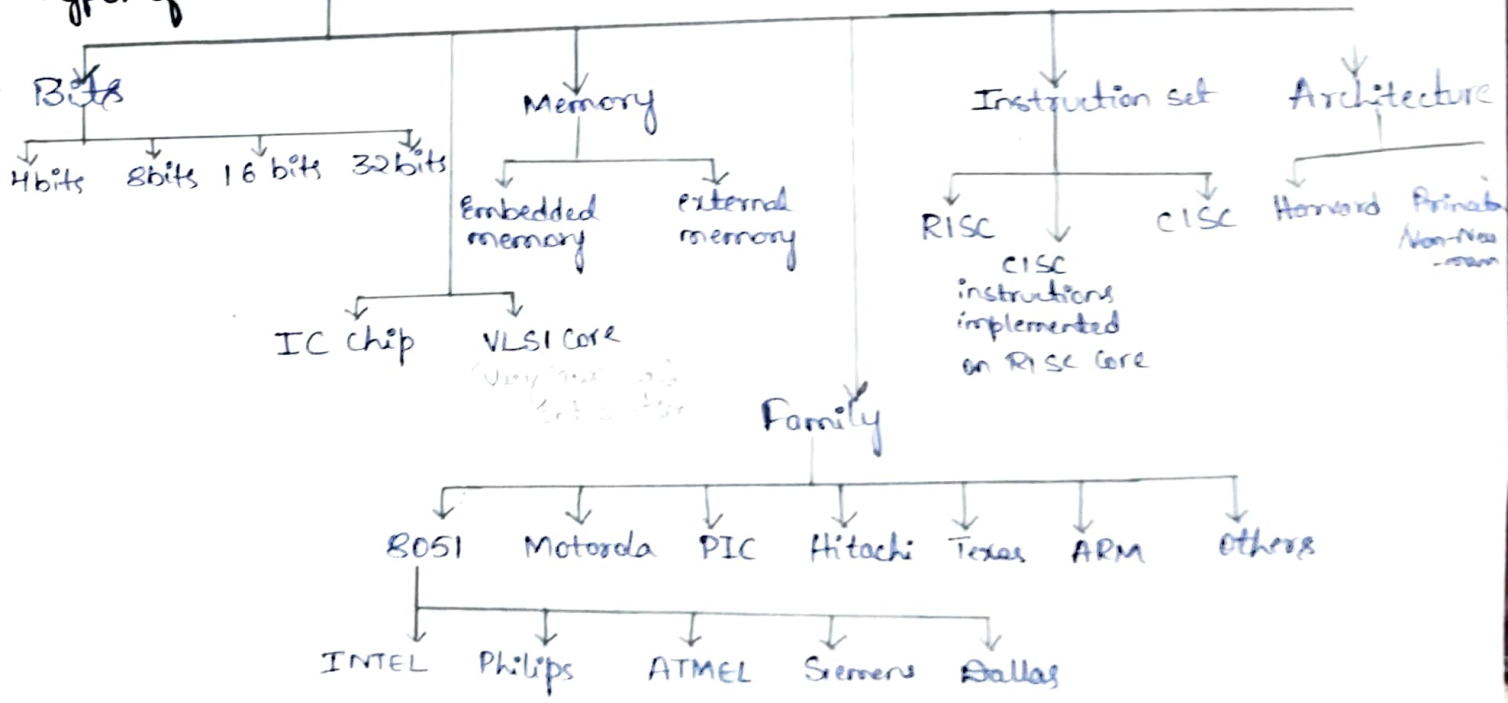
Harvard



* It has separate program memory & data memory
* Instruction & data can be fetched simultaneously
* Takes less machine cycle
* Uses RISC architecture
* Instruction parallelism is a main feature
* Also known as data flow (or) data driven computers
* Chip design is complex due to separate memory space

Ex: General purpose MCs, Special DSP chips etc

Types of Micro-controllers



Embedded System:

An embedded system is a single purpose computer (it has µp (or) µc) built into a larger system for the purpose of controlling and monitoring the system.

Ex: Cell phones, MP3 Players, ABS breaking system, Satellite guidance avionics etc.

An embedded system is a MC (or) MP based system which is defined to perform a specific task.

Simple block diagram of 8051 (Overview of 8051)

A microcontroller is a computer on a single chip (VLSI). The 8051 is a first MC of MCS-51 family, introduced by INTEL corporation in 1981. Simple block diagram of 8051 is as shown in figure:

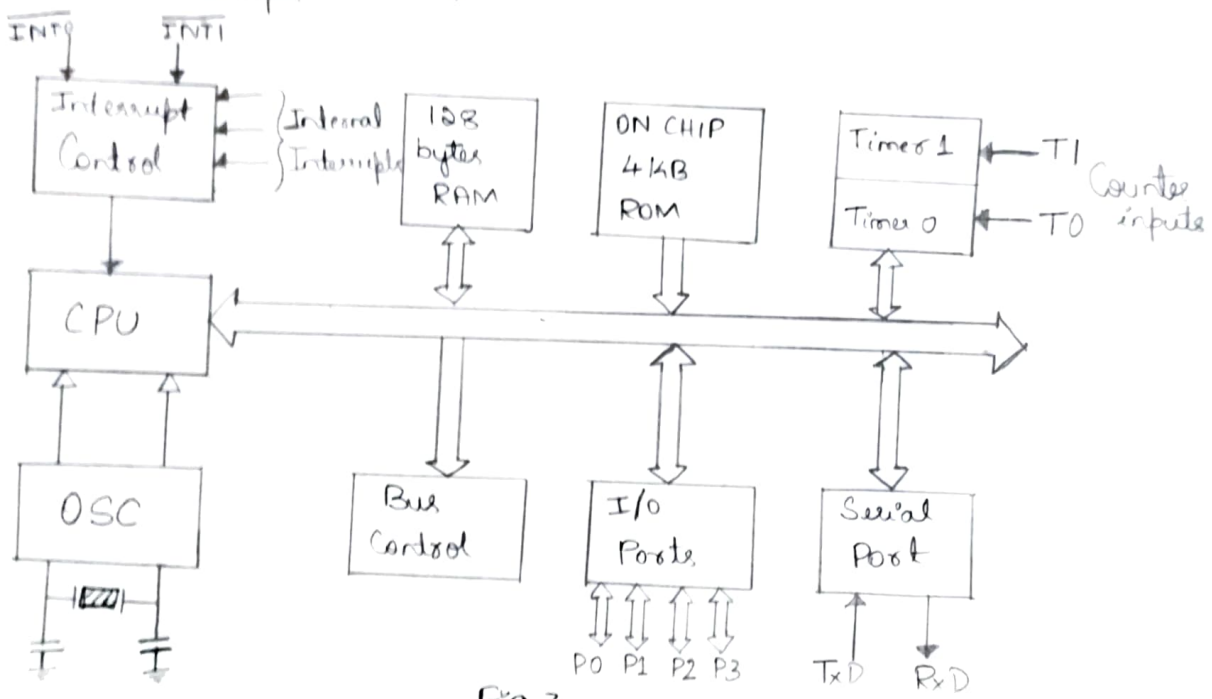


Fig 3.

Salient features of 8051 microcontroller are as follows:

* 8-bit CPU with registers A & B
* 16-bit program counter & data pointer (DPTR)
* 8-bit program status word (PSW)
* 8-bit stack pointer (SP)
* Internal ROM (or) EPROM (4kB)
* 128 bytes of on-chip RAM
* 32 bi-directional I/O pins that can be used as four 8-bit ports (P0, P1, P2 & P3)
* Oscillator & Clock circuits

* Two 16-bit timer / Counters : T0 & T1
* Full duplex serial data receiver / transmitter : SBUF
* Control registers : TCON, TMOD, SCON, PCON, IP & IE
* Two external & three internal interrupt sources.

Pin details of 8051 & their functions :

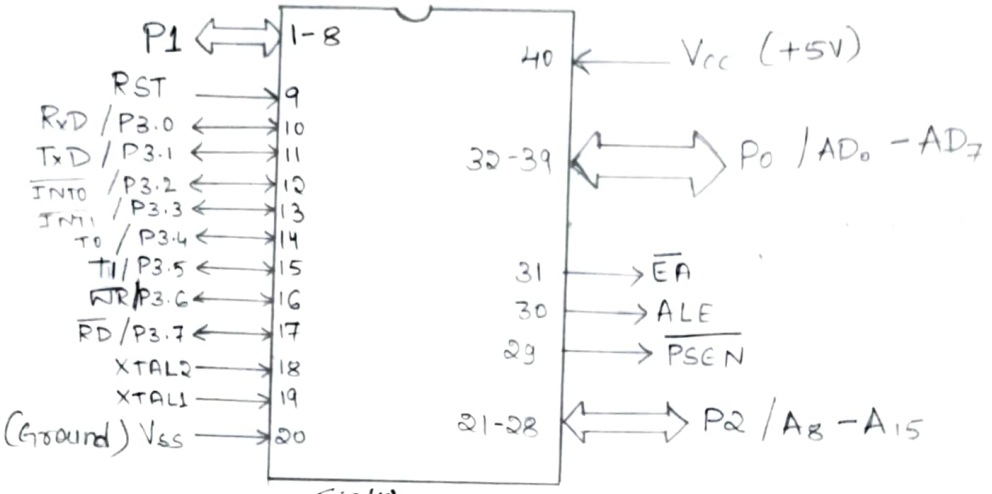


Fig (4)

* The 8051 μC is a 40 pin, dual in package (DIP), single IC as shown in above figure.
* It work with crystal frequency of range 1MHz to 16MHz & requires a power supply of +5V.

Vcc (40) : Vcc pin is connected to +5V power supply with rated current of 125mA & Pmax of 1W.

Vss (20) : Vss pin is connected to ground.

XTAL1 & XTAL2 (18 & 19) : Crystal of clock frequency 12 MHz is connected blw these two pins with capacitors of 30 pF.

RST (9) : Reset pin is made high for 2 machine cycles to initialize μC. Program Counter (PC) will be loaded to address of 0000H, Stack pointer (SP) is initialized to 07H & default Bank 0 will be selected when RST pin goes high.

P0 / AD0-AD7 (32-39) : Port 0 serves as true bidirectional I/O port & also it acts as low order address & data bus

P1 (1-8) : Port 1 serves as quasi bidirectional 8-bit I/O port

P2 (21-28) : Port 2 is also quasi-bidirectional 8-bit I/O port (A8-A15) Alternatively it serves as high order address bus.

ALE (30) : Address Latch Enable pin is used to demultiplex AD0-AD7.

→ Demultiplex is a process of separating low order address A_0-A_7 & data D_0-D_7.

The ALE is an active high pulse generated to latch the low order address during every operations.

PSEN (29): Program Store Enable is the o/p control signal. It is used to fetch program from external program memory by sending active low signal. During internal program execution it remains high.

$\overline{EA}$ (31): External Access pin, when held high ($\overline{EA}=1$), executes instructions from the internal ROM (Program memory) till address OFFFH; beyond this address, the instructions are fetched from external program memory. If $\overline{EA}=0$, all the instructions are fetched from external program memory (0000H to FFFFH).

P3(10-17): Port 3 serves as quasi bidirectional I/O port. It also acts as serial port (R_xD & T_xD), external interrupts (INT0 & INT1), timers (T0 & T1) & Control signals $\overline{WR}$ & $\overline{RD}$

Architecture of 8051

The block diagram of 8051 in figure 5 shows all the features unique to micro-controller. The 8051 mainly consists of

(i) 8 bit CPU
- ALU
- Instruction decoder & control
- CPU registers

(ii) Memory
- On chip RAM (Internal)
- On chip ROM (Internal)
- Stack

(iii) Four 8-bit ports

(iv) Two timers

(v) One serial port

(vi) Bus

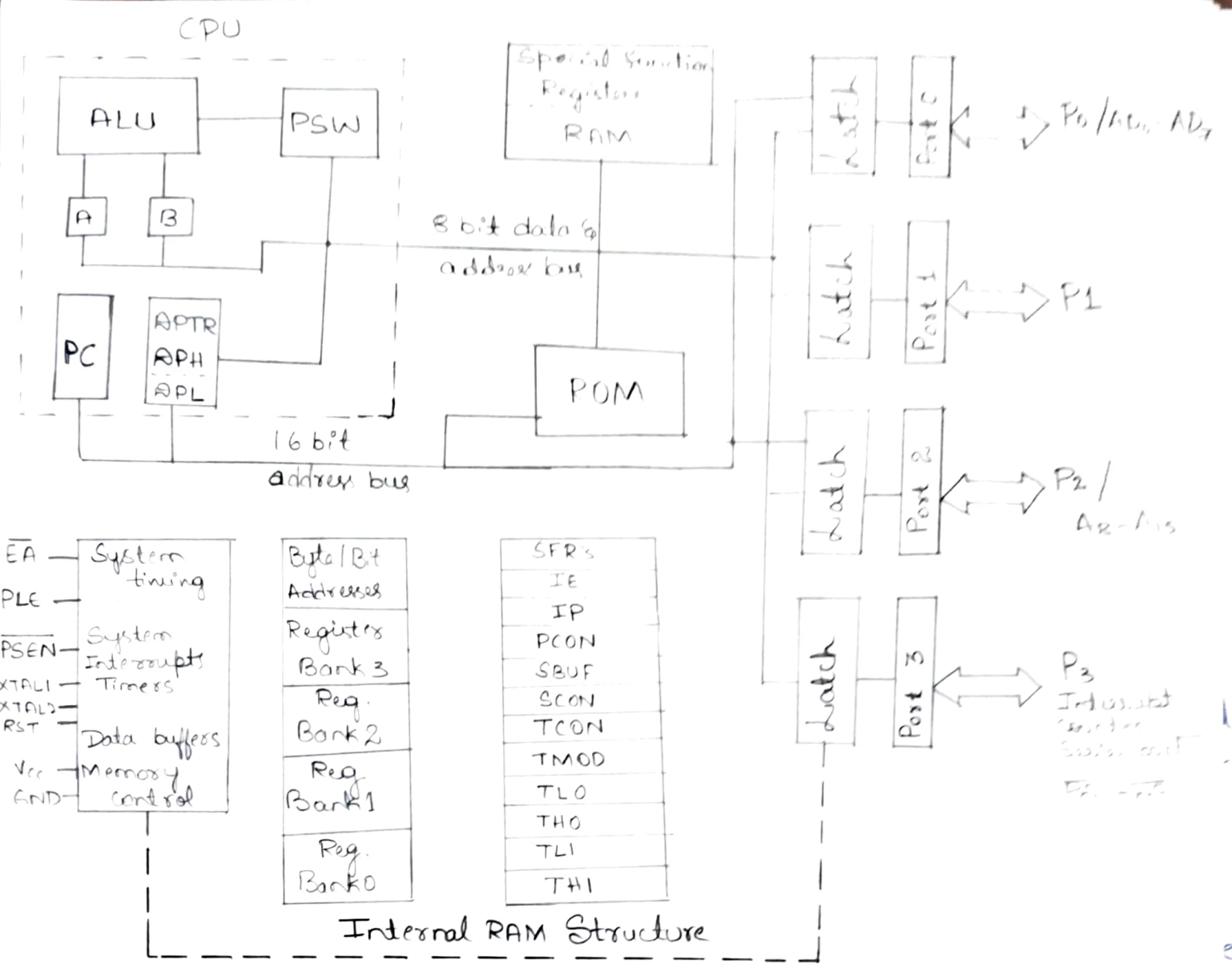


Fig. 5 Block diagram of 8051

(i) **CPU** : Central Processing Unit consists of ALU, instruction decoder & timing generation unit & registers

(a) Arithmetic Logic Unit: ALU performs the following functions

* Arithmetic operations
* Logical operations
* Decision making.

ALU performs these operations in association with registers A (&B), PSW, PC & @PTR. Reg. A stores one of the operand during arithmetic/logical operations that takes place in ALU. ALU stores result in reg. A & status of result is reflected in the flags of PSW.

(b) Instruction Decoder & Control : The instruction decoder & control are parts of timing & control unit. When an instruction is fetched from program memory, it is loaded in the instruction register. The decoder

decodes the instruction & establishes the sequence of events to follow.

The timing generation & control unit synchronizes all the μC operations with the clock & generates control signals necessary for communication b/w CPU & peripherals.

(c) CPU Registers:

'A' Register: Reg A is known as Accumulator & it is 8-bit in size. Accumulator has direct connection with ALU. Reg A performs the following functions:

* It is used to store 8-bit data temporarily
* for arithmetic / logic operations one of the operands is stored in reg A
* After arithmetic / logic operations result is stored in the accumulator
* It is used for transfering data b/w 8051 and external memory.
* Reg A is used along with Reg B multiplication & division operations.

'B' Register: It is also known as 8-bit register. It is used along with reg A for multiplication & division operations.

* It stores 8-bit data temporarily
* One of the 8-bit operands is stored in Reg B during multiplication operation & part of result is stored in reg after multiplication process
* During division operation, it holds 8-bit divisor & after the operation, the remainder is stored in reg B.

Program Status Word (PSW)

It is an 8-bit register consists of math flags, user flag & register bank select bits. This register reflect the status (condition) of ALU after arithmetic operations hence, it is known as Program status word.

Math flags such as CY, AC, P & OV will be set (or) reset according to the result after operations in ALU.

Format of PSW is as follows:

CY	AC	F0	RS1	RS0	OV	-	P
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

D7 - CY: This flag is used to reflect carry (or) borrow of result. If there is a carry from bit D7 then Carry flag will be set Otherwise it is reset & used in arithmetic, jump, rotate & boolean instructions.

D6 - AC (Auxiliary Carry): This flag reflect carry (or) borrow from the lower nibble. If there is a carry from bit D_3 to D_4, then AC flag will set, Otherwise it is reset. It is used for BCD arithmetic.

D5 - F0 (User flag 0): Flag 0 is User flag & user can use this flag to store certain conditions

D4 - D3 (RS1 & RS0): RS1 RS0 Register Bank select bits

RS1	RS0	Register Bank select bits
0	0	Bank 0
0	1	Bank 1
1	0	Bank 2
1	1	Bank 3

When µC is reset, by default Bank 0 is selected by setting 0 0 in bits D4 - D3 of PSW

D2 - OV (Over flow Flag): OV flag is used to detect errors in signed arithmetic operations. When 2 signed numbers are added, if the result exceeds the destination, OV flag is set, else it is reset. OV flag is set if there is a carry from D6 to D7 but no Carry from D7 (or) if there is a carry from D7 but no carry from D6 to D7

D1: Undefined, reserved for future use

D0 - P (Parity flag): Parity flag reflects number of 1's in reg. A after the operation. It is set if result contains an even no of 1's other it is reset for odd number of 1's.

... four registers bank are known as (8)

Stack & Stack Pointer:

Stack refers to area of internal RAM (00H - 7FH)
Stack is the memory location in the internal RAM that is used to store data temporarily during execution of program.

Stack pointer is an 8-bit register & it holds the address of stack RAM address hold by SP is called the top of the stack. The SP is initialised to 07H, when μC is power-up (or) reset

Program Counter (PC): PC is a 16-bit register Program instruction bytes are fetched from locations in ROM (Internal (or) External) those are addressed by the PC. PC holds the address of next instruction to be executed. The PC is automa-tically incremented after every instruction byte is fetched.

After reset (or) power up of μC, the PC will be set to 0000H & the CPU will start executing the first instruction stored at Program memory (ROM) location 0000H. PC is the only register that does not have an internal address.

Data Pointer (DPTR):

The data pointer - DPTR is a 16-bit register & can be divide into 8 bit register DPH (High byte) & DPL (Low byte). DPTR can hold a 16-bit address. Thus, it is used to furnish address informa-tion for internal & external Program memory (ROM) & external data memory (RAM).

Memory:

The 8051 has 4kB of internal ROM (Code (or) Program memory) & 128 bytes of internal RAM (Data memory).

Internal RAM

* 128 bytes of internal RAM is organized into three distinct area:

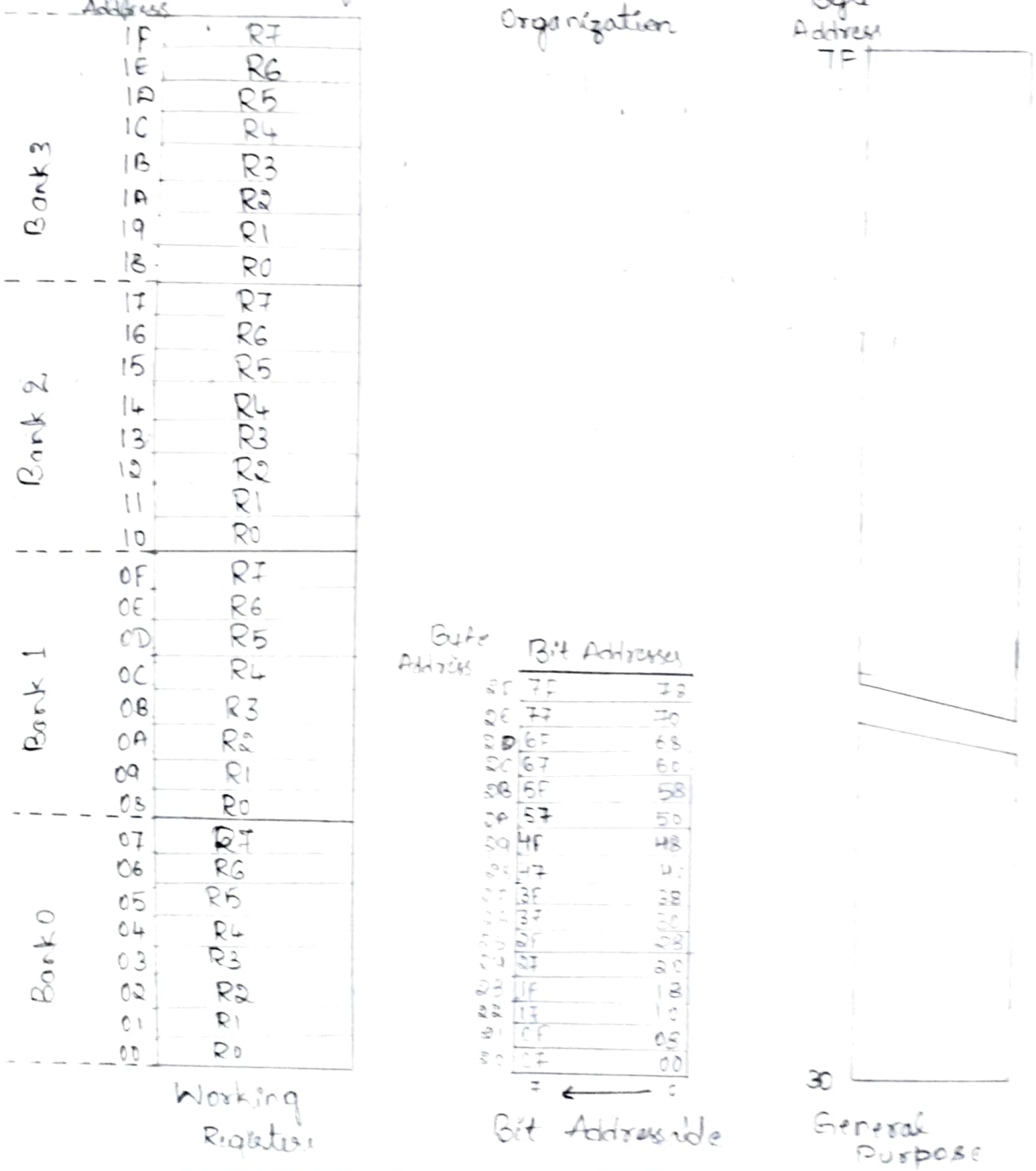
(a) Register Banks: 32 bytes are grouped into four banks of eight registers (R0 - R7). Four registers bank are known as

Bank0, Bank1, Bank2 & Bank3 & address is 00H - 1FH. Bits RS1 & RS0 of PSW select the bank, when 8051 is power-up or reset Bank0 is selected as default Bank.

(b) Bit addressable RAM: 16 bytes of RAM from 20H to 2FH is reserved as bit addressable area. This is where individual bits in internal RAM can be set (or) reset (cleared). There are 128 bits & they have address 00 to 7FH.

(c) General Purpose RAM: 80 bytes of internal RAM are available for general purpose storage. The address of this area is from 30H to 7FH Byte Address

Fig 6 Internal RAM Organization



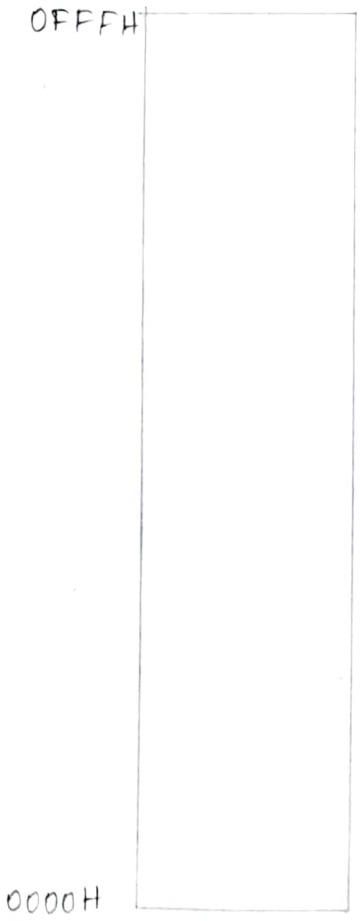
Special Function Register :

The 8051 operations that do not use the internal 128-byte RAM addresses from 00H to 7FH are done by a group of specific internal registers, each called a special-Function register (SFR), which may be addressed much like internal RAM, using addresses from 80H to FFH. Some SFR's are bit addressable.

PC is not a part of SFR & has no internal. SFR's can be grouped into
* Math Reg's : A & B
* Status Reg's : PSW
* Pointer Reg's : DPTR & SP
* I/O ports : P_0, P_1, P_2 & P_3
* Peripheral Control Reg's : SCON, TCON, PCON, TMOD, IE, IP
* Peripheral data Reg's : TL0, TL1, TH0, TH1, SBUF

		RAM address
Accumulator	A	0E0
Arithmetic	B	0F0
Addressing external memory	DPH	83
	DPL	82
Interrupt enable	IE	0A8
Interrupt Priority	IP	0B8
Input/Output port latch	P0	80
	P1	90
	P2	A0
	P3	0B0
Power Control	PCON	87
Program status word	PSW	0D0
Serial port control	SCON	98
Serial port data buffer	SBUF	99
(16 bit Pointer)	SP	81
Timer/Counter mode control	TMOD	89
	TCON	88
Timer 0 low byte	TL0	8A
Timer 0 high byte	TH0	8C
Timer 1 low byte	TL1	8B
Timer 1 high byte	TH1	8D

SFR's



Internal ROM

* SFRs are named in certain opcodes by their functional names such as A (ACC) TH0 and are referenced by other opcodes by their addresses

Port 0 structure can be ... DFOH (or) 8CH

Internal ROM:

Internal ROM of 8051 has address range from 0000H - OFFFH. Program addresses higher than OFFFH, which exceed the internal ROM capacity, will cause the 8051 to automatically fetch code bytes from external Program memory.

Oscillator Circuit & Reset Circuit:

8051 has on-chip oscillator circuit & only crystal need to be connected b/w XTAL1 & XTAL2. 30pF disc capacitors are recommended when a quartz crystal is used. 8051 can work with the frequency from 1MHz to 16MHz. 30 pF capacitors are selected to operate oscillator of 8051 at a crystal frequency of 12MHz. The crystal frequency is the basic internal clock frequency of the micro controller. Oscillator generates a train of pulses at the crystal frequency as shown below:

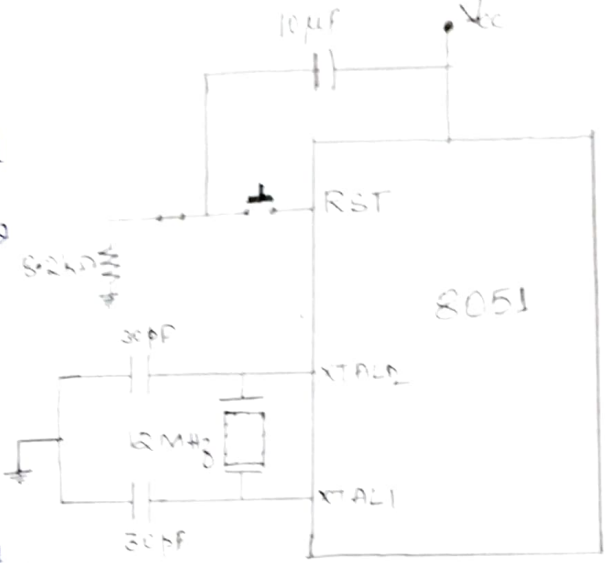
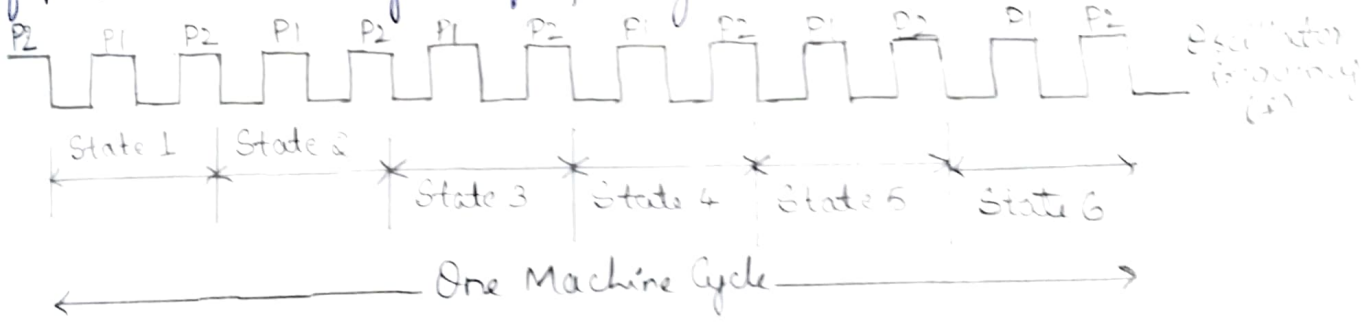


Fig: Crystal or ceramic Resonator Oscillator Circuit



Clock cycle: Time taken for one cycle is known as clock cycle. Clock cycle is reciprocal of clock frequency (or) crystal frequency

$$T_{clock} = \frac{1}{f_{crystal}}$$

State: Two clock cycles is equal to a state. A state is basic time interval for operations such as opcode fetch, opcode decoding opcode execution (or) writing a data byte.

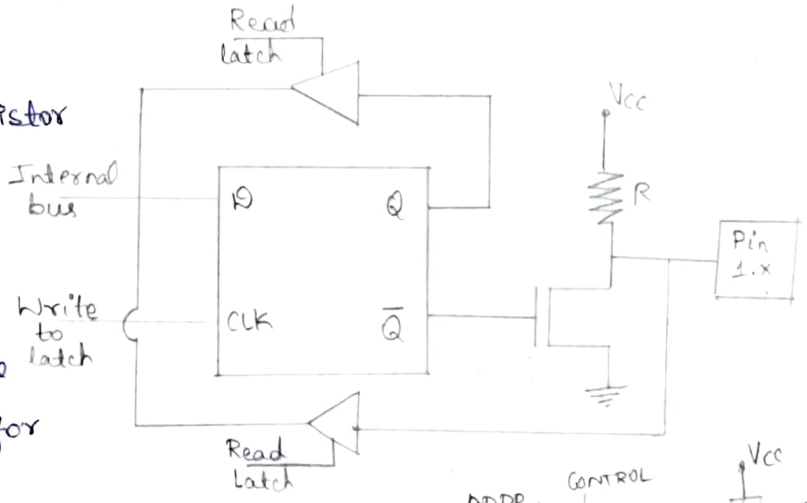
Port 0 structure can be seen from fig.1. It has two FET's & for normal operation, the upper pull up FET is OFF, providing open drain output pin & external pull-up resistors is required.

* If 1 is written to Port 0 latch, either FET's go OFF & the pin floats and can be used as high impedance input.
* Writing 0 to Port 0 pin, turns ON the lower FET & makes the pin suitable for output operations.

The upper FET operates when ADDR/Data bus & Control signal is provided. The pull up FET only operates when there is an access to external memory. Logic 1 on an address bit will turn the upper FET on & lower FET OFF to provide a logic high at the pin. When the address bit is a zero, lower FET is ON & upper FET is OFF to provide a logic low at the pin. After address has latched, the bus is turned around to become a data bus.

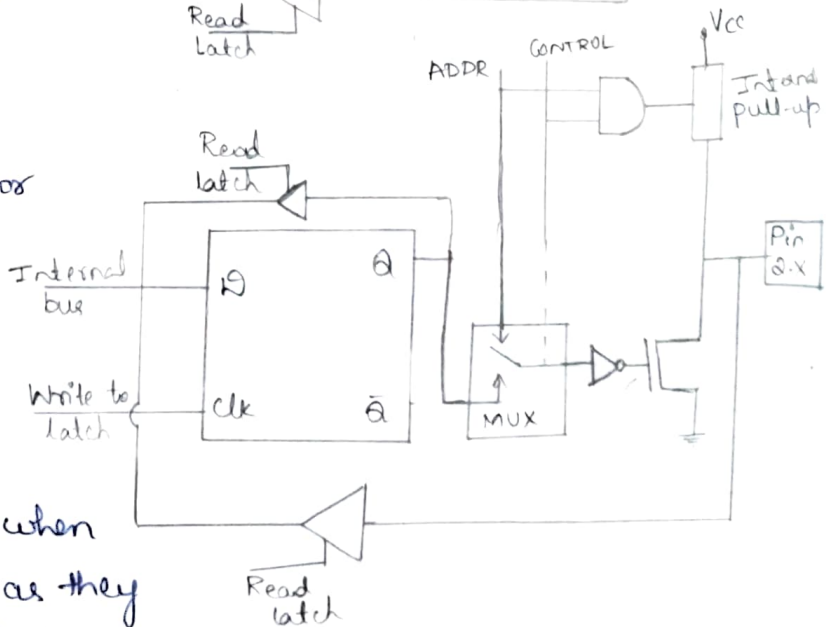
Port 1:

* It has internal pull-up resistor
* Writing 1 to the port 1 pin turns of FET & makes the pin suitable for input operation.
* Writing 0 to port 1, turns on the FET & makes the pin suitable for output operations.



Port 2:

* Do not need external pull-up resistor
* Writing 1 makes Port 2 as input
* Writing 0 makes Port 2 as output
* Port 2 is used to carry high order address when it receive control signal.
* Port 2 latches remain stable when external memory is addressed, as they do not have to be turned around for data input as is the case for Port 0



Mode 0: The timer is 13 bit wide

Machine Cycle: Smallest interval of time within the MC to accomplish execution of any simple instruction is known as machine cycle.

1 Machine cycle = 6 States.

An instruction may take one or more machine cycles to perform a task. Time required to execute an instruction is given by $T = \frac{CX12}{f_{crystal}}$ where C ⇒ No. of mc cycle

$f_{crystal}$ ⇒ frequency of Crystal

Ex: If an instruction take 1 M/c cycle to execute & f = 12MHz Then calculate time to execute the instruction

$$T = \frac{1 \times 12}{12 \times 10^6} = 1\mu s$$

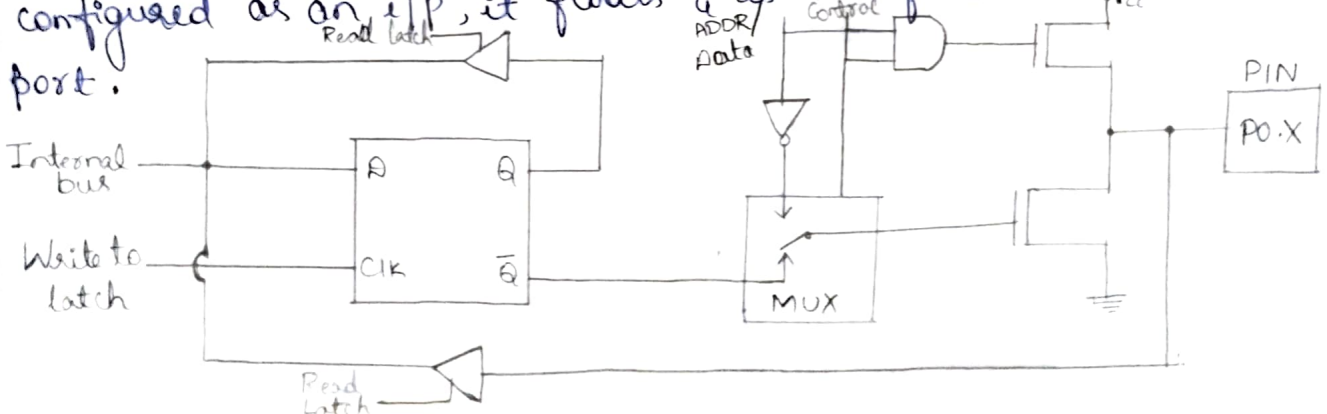
8051 can be reset when RST pin is made high for 2 machine cycles. Reset Circuit consist a capacitor of 10μF connected blw Vcc & RST pin & resistor of 8.2kΩ connected to RST pin w.r.t Vss. Capacitor is charged by +Vcc & apply active high pulse for minimum 2 machine cycles.

When 8051 is reset, PC holds the address 0000H, SP holds in RAM address 07H, Bank 0 is selected as default register bank

Input/Output Ports:

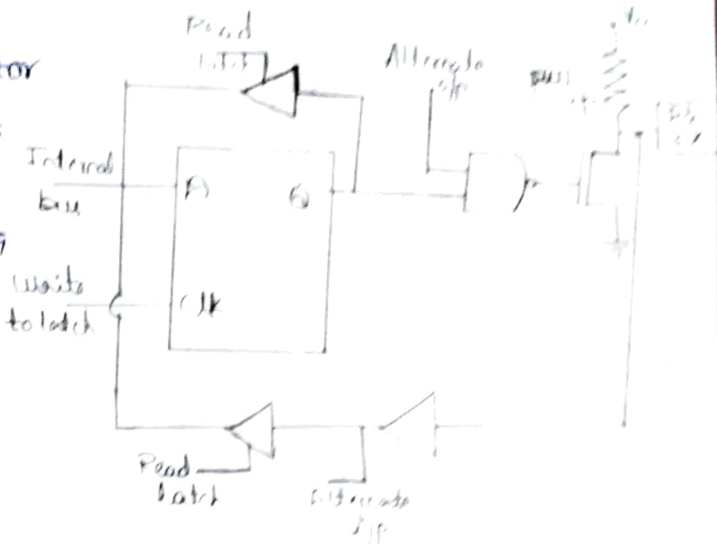
8051 has four 8-bit ports & the four ports are called Port 0 Port 1, Port 2 & port 3. Besides their use for I/O functions, these pins are alternate other functions

Port 0: Port 0 does not have internal pull up resistors, when configured as an i/p, it floats & is therefore a true bi-directional port.



Port 3:

* Do not need external pull-up resistor
* Writing 1 to Port 3 turns off FET & makes the pin as input
* Writing 0 to Port 3 turn ON FET & makes the pin as output.
* Port 3 pins need to be deselected for alternate functions.



Timers / Counters:

* These are 2 timers / Counters, useful for real time application such as width measurement, band rate generation etc. Pulse counting, frequency measurement, .
* Timer is used to provide time delay. Counting rate is oscillator frequency by 12.
* Counter is used to count external events happening outside the µC Thus it counts number based on pulse applied @ T0 (or) T1
* If a counter is programmed to be a timer, it will count the internal clock frequency of 8051 oscillator divided by 12d.

Timer 0 :- 16 bit register accessed as low & high bytes

Timer 1 :- 16-bit register accessed as low & high byte

TMOD:- Both timers 0 & 1 use the same register called TMOD to set various operation.

Gate	C/T	M1	M0	Gate	C/T	M1	M0
Timer 1				Timer 0			

Gate → Control when gate is set } when cleared timer is enabled
⇒ INT = 1 & TR_x = 1

C/T → Timer (or) Counter

M1, M0 – Mode bits

M1	M0	Mode	
0	0	0	13 bit timer mode
0	1	1	16 bit timer mode
1	0	2	8 bit auto reload
1	1	3	Split time mode.

TCON: (Bit Addressable)

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
For timer				For Interrupt			

TF1 – Timer 1 Overflow flag ; Set when timer rolls from 1 to 0

TR1 – Timer 1 run control unit

TF0 – Timer 0 Overflow flag ; Set when timer rolls from 1 to 0

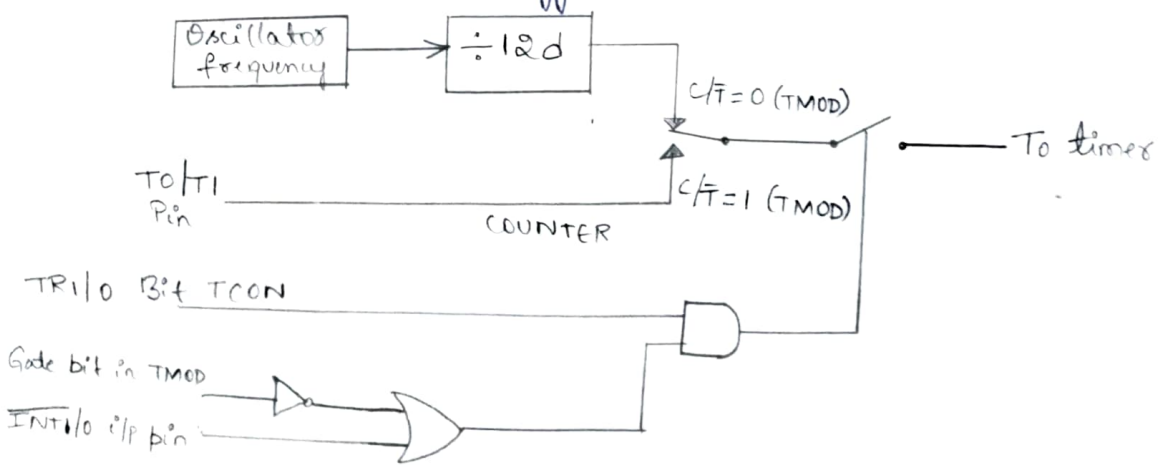
TR0 – Timer 0 run control bit

IE1 – External interrupt 1 Edge flag
Set to 1 when high to low signal to $\overline{INT1}$

IT1 – External interrupt 1 signal type control bit
↳ { 1 Falling edge triggered
0 Low level signal triggered

IE0 – External interrupt 0 Edge flag
Set to 1 when high to low signal is received to $\overline{INT0}$

IT0 – External interrupt 0 signal type control bit
↳ { 1 Falling edge triggered
0 Low level triggered.



Mode 0: The timer is 13 bit wide. It counts values from 0000H - 1FFFH. When count overflows from 1FFFH to 0000H timer interrupt flag (TFx) will be set

Fig Timer Mode 0 13-bit Timer/Counter

Mode 1: The timer is 16-bit wide. It counts values from 0000H - FFFFH when count overflows from FFFFH to 0000H, timer interrupt flag (TFx) will be set.

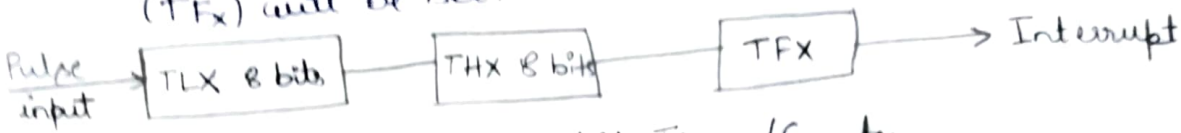


Fig Timer Mode 1 16-bit Timer/Counter

Mode 2: This mode is 8 bit auto-reload feature. TLx is load from THx. It counts from 00H - FFH. When count overflows from FFH to 00H, timer flag TFx will be set

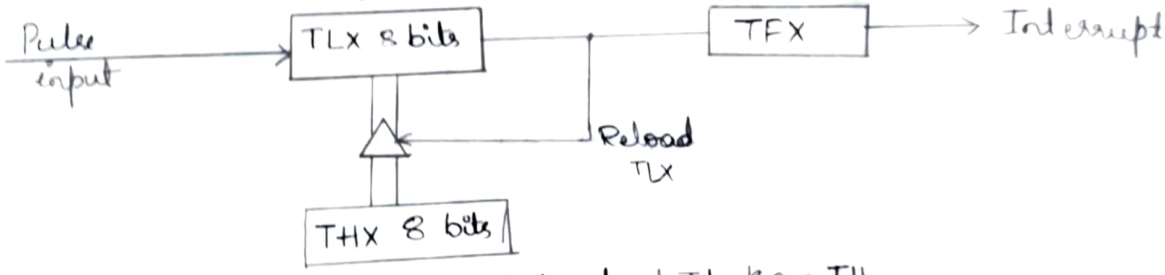


Fig. Timer Mode 2 Auto-Reload of TL from TH

Mode 3: Timer 0 register TH0 & TL0 are used as two separate timers. TL0 uses timer flag TF0 & TH0 uses timer flag TF1.

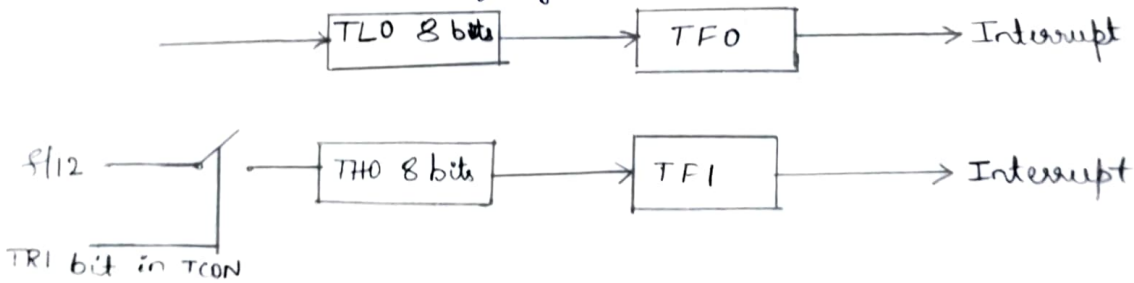


Fig. Timer mode 3 two 8 bit timers using Timer 0

Serial Ports:

Serial data Communication is very commonly used for digital data communications. 8051 supports a full duplex serial port i.e., it can transmit & receive a byte simultaneously. 8051 has TxD & RxD pins for transmission & reception of serial data respectively.

The data to be transmitted is placed in the SFR known as SBUF & similarly the data received is present in the SBUF register

SBUF register is a SFR with 8-bit in wide.

SCON

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SM1	MODE
baud = f/12	0	0	8 bit shift reg
baud = variable	0	1	8 bit UART,
baud = f/32 (or) f/64	1	0	9 bit UART
baud = variable	1	1	9 bit UART

SM2 : Multiprocessor Communication bit

REN : Receive enable bit

TB8 : Transmitted bit 8

RB8 : Received bit 8

TI : Transmit Interrupt flag

RI : Receive Interrupt flag.

UART : Universal asynchronous receiver Transmitter

SMOD :

7	6	5	4	3	2	1	0
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.

Interrupts :

Interrupt is an internal (or) external hardware signal given to micro-controller to perform a dedicated task.

- Interrupts
  - Internal interrupts
    - TF0
    - TF1
    - TI (or) RI
  - External interrupts
    - $\overline{INT0}$
    - $\overline{INT1}$

Priority —
1. $\overline{INT0}$
2. TF0
3. $\overline{INT1}$
4. TF1
5. TI (or) RI

Interrupt Enable [IE]

EA	-	ET2	ES	ET1	EX1	ET0	EX0

- EA : Enable All Interrupt
- ET2 : Enable Serial Port interrupt
- ES : Enable external interrupt 1
- ET1 : Enable Timer 1 overflow interrupt
- EX1 : Enable external interrupt 0
- ET0 : Enable timer 0 overflow interrupt
- EX0 : Enable external interrupt 0

Interrupt Priority (IP) :

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

PS – Priority of serial port interrupt
PT1 – Priority of timer1 Overflow interrupt
PT0 – Priority of timer0 Overflow interrupt
PX0 – Priority of external interrupt 0
PX1 – Priority of external interrupt 1

Programming - 8051 MicroController

Instruction Set of 8051

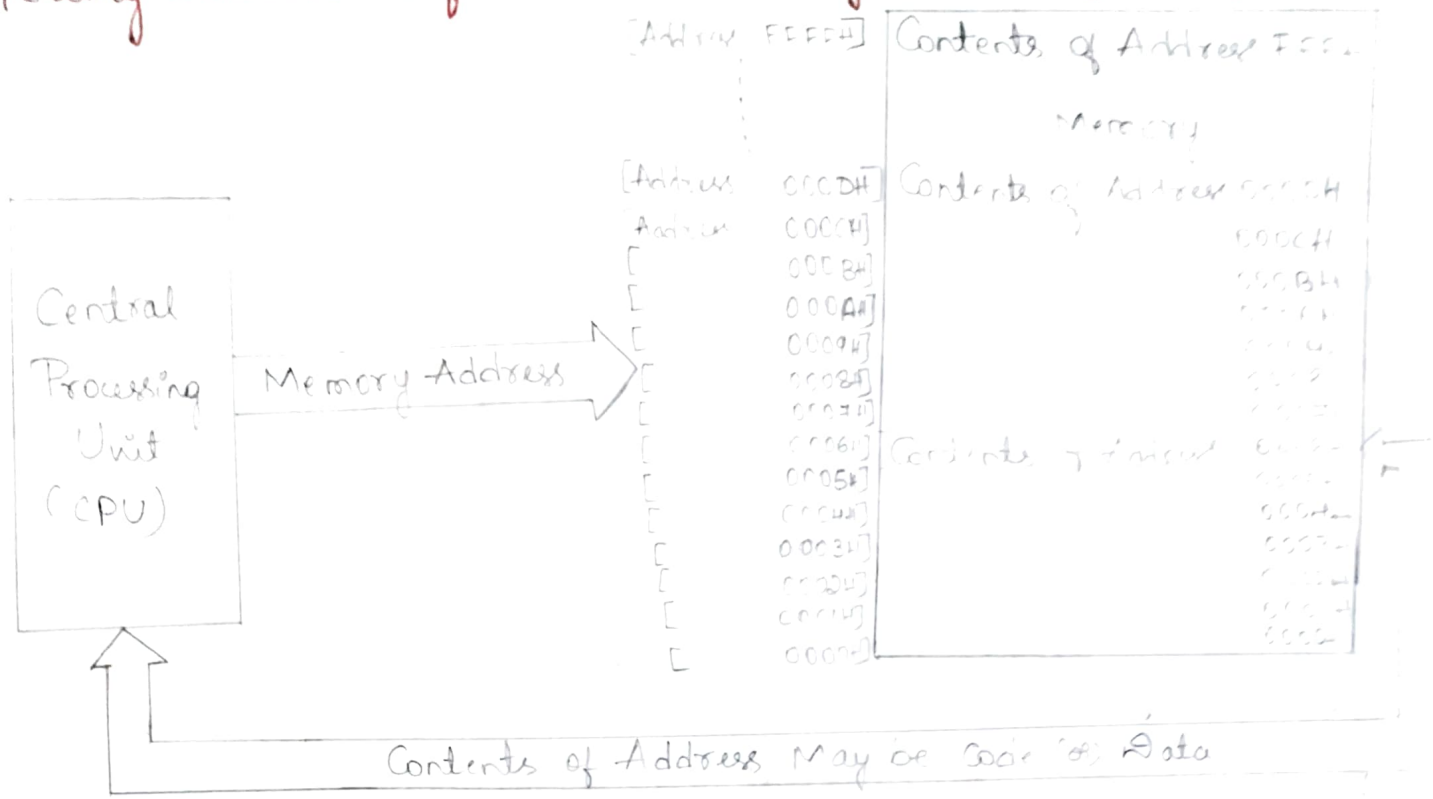
Hardware Concepts : Memory

Circuits that are meant to store binary data in electrical form are known as solid-state memories & are made of millions of storage devices. Typically, bits are transferred b/w CPU & memory as positive-logic voltages. There are two types of memory ROM & RAM

ROM : ROM is also known as Program memory. Memory circuits where binary numbers are stored permanently are known as read only memory (or) ROM. Contents of the memory cell in ROM may not be changed by CPU, but CPU can read the contents of memory ROM is also called non-volatile, because the contents of memory will be present even when the power is off. Program (or) Code in the form of binary numbers are stored in the ROM.

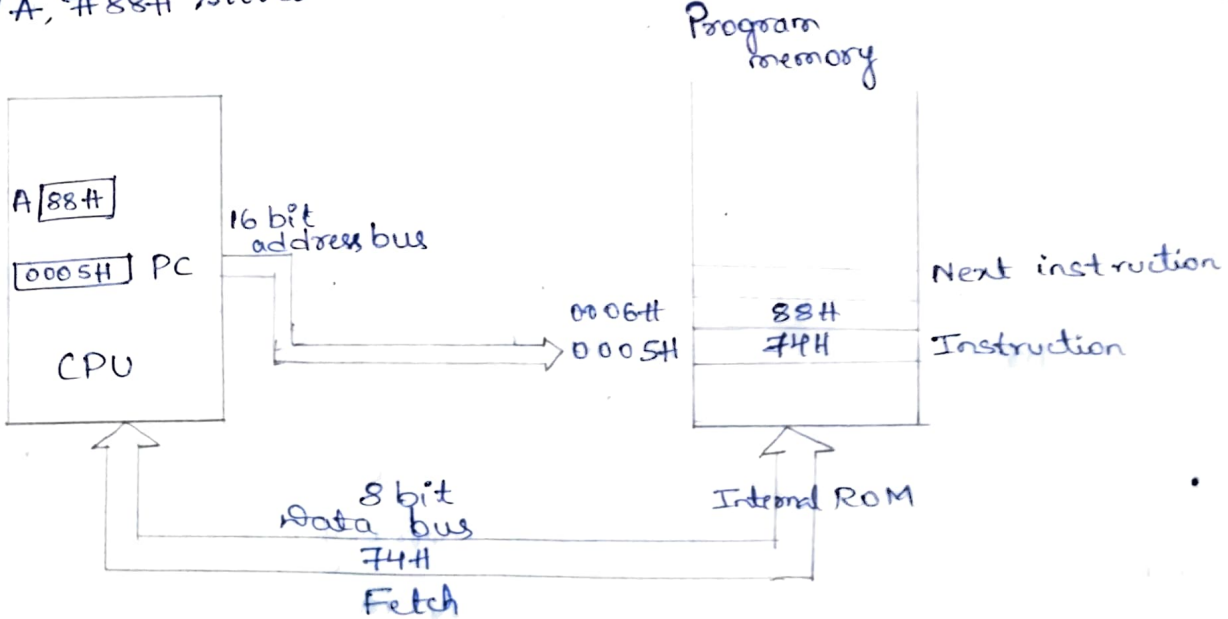
RAM : Memory circuit where CPU can read & write binary numbers is known as Random Access memory (RAM) (or) R/W memory RAM is also referred as data memory in µC. RAM is volatile memory because its contents are lost when Power is OFF. RAM can be classified as SRAM (Static RAM) & DRAM (Dynamic RAM)

Getting instructions from Code memory [Opcode fetch from ROM]



Set of instructions are stored in the program memory is known as program. Getting instruction from code memory into the CPU is called fetching instruction. CPU performs decoding & execution of instruction after fetching an instruction from code memory. Ex: MOV A, #88H

Let MOV A, #88H stored in two locations of internal ROM starting at 0005H



On start-up (Reset or Power up), PC will be automatically loaded with starting address of program memory (0000H in 8051) The CPU fetch first instruction from this location & executes

the operation that is indicated by the first instruction.

After fetching first instruction, PC is automatically incre-mented to point the next instruction. This means PC holds address of next instruction. Thus, CPU fetches next instruction & executes that & this process continues until the µC is turned OFF (or) Otherwise halted. PC holds address of instruction stored at 0005H Therefore CPU fetches code from this address (74H) & decode that in instruction decoder. Thus execution of coping task of data from address 000FH to 'A' will be carried by CPU.

Machine Level Language:

Program is group of instructions. Such instructions written in the form of different bit pattern is known as Machine level language Ex:

Instruction in MLL	Common
0111 0100 1000 1000	Move 8 bit content to accumulator

* It is difficult to write program using 0's & 1's
* Writing program in MLL is time consuming & is very tedious task to bug the program in MLL.
* Other programmer cannot easily read & understand the program.

High level Language:

It is the language written using meaningful english words. It is machine independent because the internal circuits of the computer used to perform a task in a high level language which is not the part of language. And it is not tied to any particular processor type.

Ex: Basic, Pascal, C, FORTRAN

Assembly level language:

Group of instructions written in the short form of words that resemble the operations to be performed is knowns as assembly level language

Short form of words that resemble the operations are known as mnemonics

Ex: MOV A, #51H ; ADD A, R1

Merits of Assembly level Programming:

1. Speed of operation of MC is more in ALP compared to HLP
2. Reduced size of code
3. System size is small if ALP is used & hence it is economic.
4. Better understanding of programming is possible.
5. Possible to write program for specified purpose.

Assembler: Assembler is the program that converts assembly level program (Source code) to machine level (Object code)

Compiler: Compiler is the software that converts higher level program into machine level.

Assembler directives:

Assembler directives are the instructions that direct the assembler to do something. Some functions of assembler directives are:

1. They tell the assembler to set aside space for variables.
2. Tell the assembler to include additional source files.
3. They establish the start address for the program.

ORG (Origin):

The ORG directive is used to indicate the beginning of the address. The number that comes after ORG can be either in hex (or) in decimals. If the number is not followed by H, it is decimal & the assembler will convert it to hex.

```
ORG  00H ; Starting address is 00h
```

EQU (Equate):

This is used to define a constant without occupying a memory location. The EQU directive does not set aside storage for a data item but associates a constant value with a data

label so that when the label appears in the program, a constant value will be substituted for the label

```
COUNT   EQU 25
        .
        .
MOV     R3, #COUNT
```

END directive:

This indicates to the assembler the end of the source (asm) file. The END directive is the last line of a 8051 program i.e. Anything after the END is ignored by the assembler.

DB (Define Byte):

It is used to define the 8-bit data, the numbers with DB can be in decimal, hex, ASCII, binary formats.

```
ORG 500H
DT1: DB 28
DT2: DB 00110011B
DT3: DB 50H
DT4: DB "A"
```

Assembly Level Programming of 8051

In assembly language, all operations are written in the form of mnemonics. Mnemonics are short form of words that resemble the operations to be performed by the CPU.

Instructions:

Instructions are commands to the μC to perform the given task. Instruction format consists of mnemonic followed by destination & source fields, code as shown in figure.

Mnemonics	Destination operand	Source Operand
Opcode	Operand	

Opcode:

Opcode contains a symbolic representation of the operation to be performed by the μC.

Ex : MOV, ADD etc

Operand: Operand indicates the data to be operated on by the µC & it gives the address of operands.

Comment: To improve program clarity, a programmer uses comments throughout the program. A comment always begins with a semicolon (;)

Label : The label is the symbolic address for the instruction. It can be any combination of upto 8 letters (A-Z), numbers (0-9) & period

Instruction Set:

Instruction set can be classified as:

* Data transfering instructions.
* Arithmetic instructions.
* Logical instructions.
* Bit manipulation instructions.
* Branch control transfer instructions.

Addressing Mode:

Various methods for accessing data needed in the execution of an instruction is known as addressing modes.

Addressing modes define the way in which the operands are accessed by the instruction. There are five types of addressing mode

1. Immediate addressing mode
2. Register addressing
3. Direct —11—
4. Indirect —11—
5. Indexed —11—

1. Immediate addressing

Immediate addressing uses data as a part of the instruction

Ex: MOV A, #45H [# symbol preceding the constant indicates the immediate data type]

2. Register addressing

Register addressing mode involves the use of registers to hold the data to be manipulated (operated).

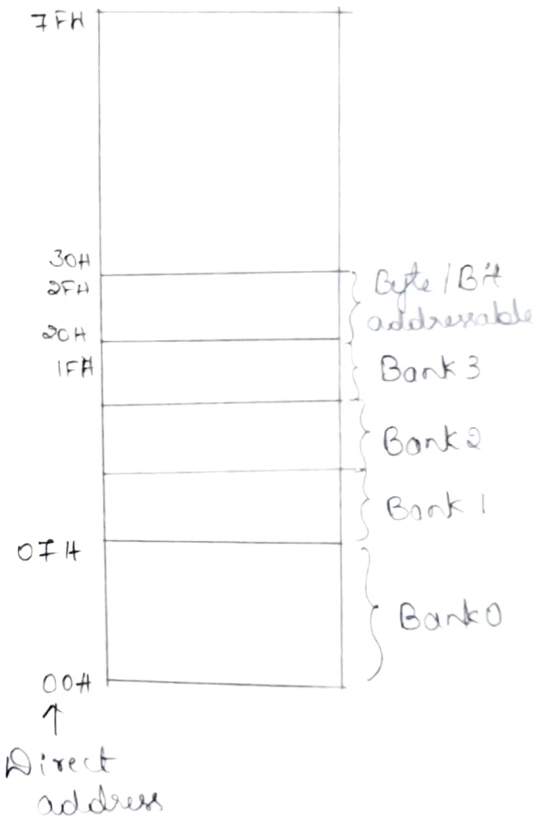
In register addressing, registers R0-R7 from the selected bank, accumulator A, reg. B, Carry bit & DPTR are used.

Ex: MOV A, R1

3. Direct Addressing:

Direct addressing mode allows to use the direct address operand in the instruction itself. It uses the lower 128 bytes of internal RAM (00h-7Fh) & the SFR's.

SFR	Address (Direct)
A	0E0
B	0F0
DPL	82
DPH	83
IE	0A8
IP	0B8
P0	80
P1	90
P2	0A0
P3	0B0
PCON	87
PSW	0D0
SBUF	99
SCON	98
SP	81
TCON	88
TMOD	89
TH0	8C
TL0	8A
TH1	8D
TL1	8B



4. Indirect Addressing:

Indirect addressing uses a register to hold the actual address that will finally be used in the data move; the register itself is not the address but rather the value in the register.

Registers R0 (or) R1 & DPTR are the only registers that can used as data pointers. R0 (or) R1 can point data locations in internal RAM from address 00h to 7Fh (128 bytes)

The mnemonic symbol used for indirect addressing is the sign '@' (at sign). Indirect addressing cannot be used to refer to SFR's.

Ex: MOV A, @R0

5. Indexed Addressing:

In indexed addressing, a separate register, either PC (or) DPTC is used to hold the base address & the A is used to hold the offset address. Adding the value of the base address to the value of the offset address forms the effective address.
Accumulator is known as indexed register.

MOVC A, @A+DPTR.