Micro controller through processor

- 1st Generation - Vaccum tube, Porch cards
  2nd Generation - Transistor, Diode
  3rd Generation - IC → (Diffital Era)
  4th Generation - VLSI microprocessor
  5th Generation - ULSI, Super computer.
  6th Generation — Quantum computing.
- Micro processor → small + functional / operation
- Wherever automation comes to picture, microcontroller plays important role like whiper in car, automatic door lock in car, home appliances etc...
- Micro controller => Advancement of microprocessor

Micro processor

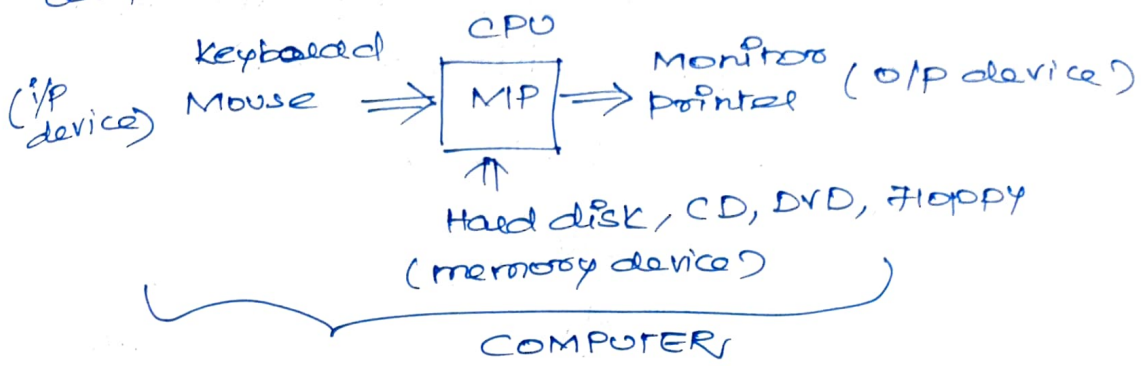
- Manually some mathematical operation like division, we can go upto 2 to 5 decimal points but if we want to go more and accurate then we opt for machine/computer calculation.
- Huge data calculation
- Eg: 3x3 matrix but with MP we can do any specification with accuracy.
- MP => combination of different mathematical operation based circuits in one place => IC
  → combination of different circuits like adder, multitier, divider, differentiator, integrator .... logical circuits like ADD, OR, XOR... all placed in a single place to do process.

I/P device ⟹ MP ⟹ O/P device

Peripheral devices (external devices)
Keyboard, mouse, printer, monitor...
Hard disk

Computer

computer is a microprocessor based system



Buses

- How we are connecting i/p devices to the processor?
- How we are connecting o/p devices to the processor?
- Through wires – Technically ⇒ Buses.
- Non-Technically, Bus?
  - Transport vechicle
  - transfer of data/person from one place to another.
- Technically, Bus?
  - Transfer of data, address, controls through wires
  - Group of wires.
  - to transfer the data from i/p to processor or from processor to o/p.
- As MP is a digital device, the data will be in the format of binary 0's, 1's (two level signal)

Ex: High state +5V/-5V/+3V/-3V ⇒ 1 } Binary
low state 0V ⇒ 0 }

Negative logic ↑ Positive logic ↑

Representing binary in diff form like +5V-0V / 1-0 (imagination)

The data, when feed as i/p will be in the form of 0's, 1's & when we get the o/p also in the format of 0's & 1's. (Digital format)

A	B
0	0
0	1
1	0
1	1

→ 2 wires/buses

2^2 (2 wires) ⇒ 4 (0's & 1's)

1 wire/bus — Either 0 or 1

3 wires/buses ⇒ 2^3 → 8

4 wires/buses 2^4 → 16

5 wires/buses 2^5 → 32

Buses → Data bus (information for the processor) (Data/)

→ control bus

→ Address bus

(wires) purpose

Data bus:

- Group of wires used for data/information
- Ex:- 5 (Decimal) → 101 (2^2 & 2^0) 4 + 0 + 1 (Binary)

For representing 5 (decimal no) i need three wires/buses, likewise if I need to represent higher numbers, then I need more no. of buses.

1st Microprocessor ⇒ 4 bit (Intel 4004)

(ff.) length of the data bus & processor's width

2nd Microprocessor ⇒ 8 bit (Intel 8008)

(8085, 8086, ...)

If we are giving any information, then it will be in the format of 8 bit ... so on ... o/p also

(ix.) very 1st computer comes under 8086 (8 bit)

IA-32

- 8086 - 8bit
- 80186 - 16bit
- 80286 - 32bit
- 80386
- 80486

Single processor

↓↓

combining 2 processor

↳ hence the name core 2 dual means in a place of single processor we have '2'.

Ex:- In 8085 processor, (8 bit)

5 (Decimal) → 0000 0101

In core I3 processor, (32 bit)

5 (Decimal) → 0000 0000 0000 0000 0000 0000 0000 0101

then think about 64 bit that we are using now,

So why we keep on increasing the data bus from 8 bit to 64 bit?

(easy) (looks messy/bigger)

(A-Z)

Ex: to speak in English we need 26 alphabets

to speak in Tamil we need 247 letters

Each language has some basic characters, likewise when we communicate with computer, we need those basic concepts so we are increasing the data bus from 4bit → 8bit → 16bit → 32 bit → 64bit.

Here we are not changing the data but how many binary formats is important.

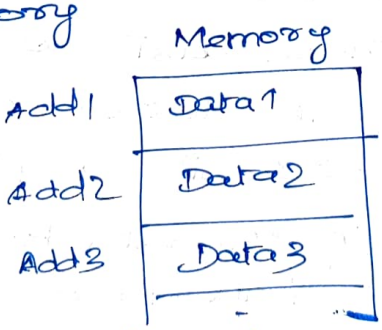
In early stage we used only english, but right now we are using many formats like ASCII, numbers, alphabets, other languages, pictures, videos, many formats, for this we need represent in binary formats so we are increasing the data bus length/wire/buses.

- Bidirectional data bus (I/P, O/P) - (⟺)

Address bus :-

- Every person is called by some name, so that's called address, every home/flat are called by some numbers & street name — address.
- Identification process.
- Every processor is connected by some I/P & O/P devices, as a human we know all these peripherals but microprocessor need to know the location where it is connected, it needs identification, so processor needs to identify the peripherals by address through ~~address~~/wires by the processor.

- Address bus is not only for peripherals but also memory

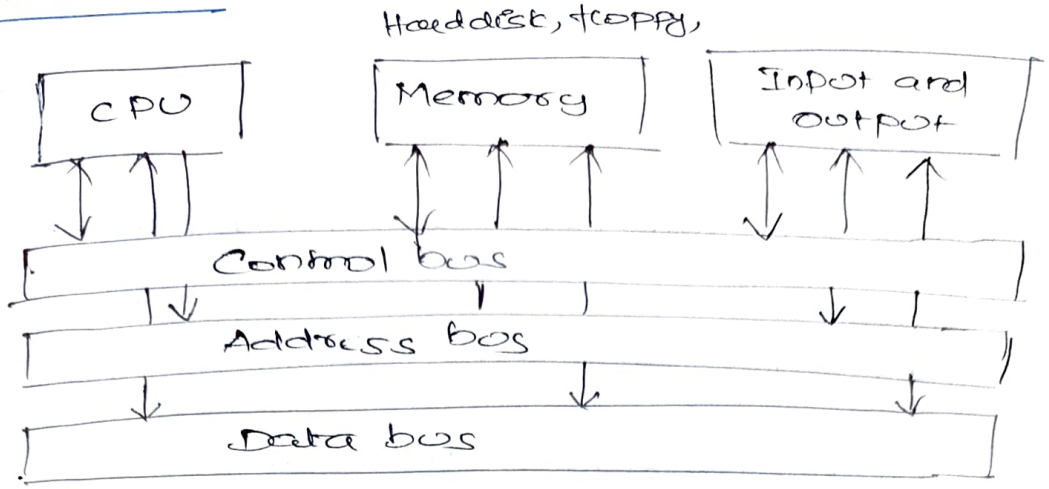


It's like Indian Bank in VIT campus, Mannivakkam branch, Anna Nagar branch ....

- more data/memory, we need more address bus...

When increasing address bus, we can use many peripherals and also memory like 1TB, 512GB memory in our laptop/system.

Control bus :-



consider when you are giving print a document, then printer is not connected to your laptop or PC, then we get notifications like printer not ready / printer not connected. Suppose you are connected printer, but did not turn ON power supply, then also we get notification, in another situation we connected printer, turned ON power supply of printer, then also we get notification if we ~~dont~~ dont paper in the tray.

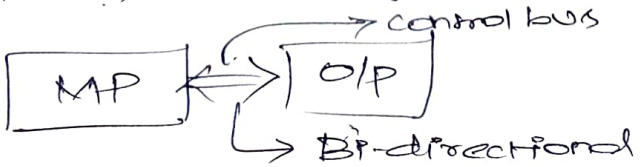
So what processor do is simply check the address of the peripheral & if not connected means, it will notify.

Here the processor checks the status of the printer (connected to PC, power supply is on ON condition?, Paper status?) - if memory alloted to printer is empty then it sends back next set of data to memory.

(or) control the operation of

To verify the status, for the peripherals, we use control bus.

Ex: ACK signal → Acknowledgement
RD signal → Read signal
WR signal → write signal



Microprocessor

- Digital IC
- Digital signals (High/low state ⇒ 0/1)
- I/p & o/p ⇒ digital signal
- A computer is developed on the basis of microprocessor.
- group of wires & Buses — control, Data (i/p/o/p), Address (memory)
- Data length ⇒ word length ⇒ bus width

In 8085 processor,

Data bus → 8bit

Address bus → 16 (2×8)

most of the processor, when compare to the data bus, double the time we have for address bus

we can control a device with the help of smart phone through simple SMS, but cannot control any devices with the help of computer. Hence one biggest drawback with processor is we cannot control any device with computer/processor (micro)

8253, 8251 → peripheral ICs
8255, for controlling devices

Address bus → memory purpose.

We cannot control any device with the help of processor

Microcontroller

- Main purpose is controlling the peripherals / controlling the electrical equipments.
- MC ⇒ built in controlling circuit with processor.
- Fans { speed — 1, 2, 3, 4 ; tubelight — ON, OFF }

So control unit differs for each devices, it wont be same for all.

- In microcontroller, we won't be having all the general purpose options, we have specific to the device that we are using.
- When we are removing some general purpose options from processor, we are adding the memory like RAM, ROM.. (external)

Ex:- Security camera
- it saves audio + video
- So we need more memory
- it won't perform other operations
- External hard disks with more memory.

In general,

Microcontroller = min processing unit + controlling CKTS + Memory.

(Memory won't be there at MP)

- MC → Specific purpose (8051) (Dedicated system) Ex: Smart EB meters
- MP → General purpose
  - more processing unit → Ex: Adder, subtractor, multiplier, integrator ...
  - min memory
  - No control circuits
  - 8085

⇒ MC → 8051 → 8bit → 16bit (Address bus) → Port together 24bits, Sometime we can use the reduced version,
⇒ MP → 8085 → 8bit → 16bit

(Data width / Data bus)

when we are utilizing address, data has no use, when we are using data, address has no use, so here we will be using 16bits address bus for both purpose.

⇒ Multiplexing

AD0 to AD7 / A8 - A15

lower order → Address / Data

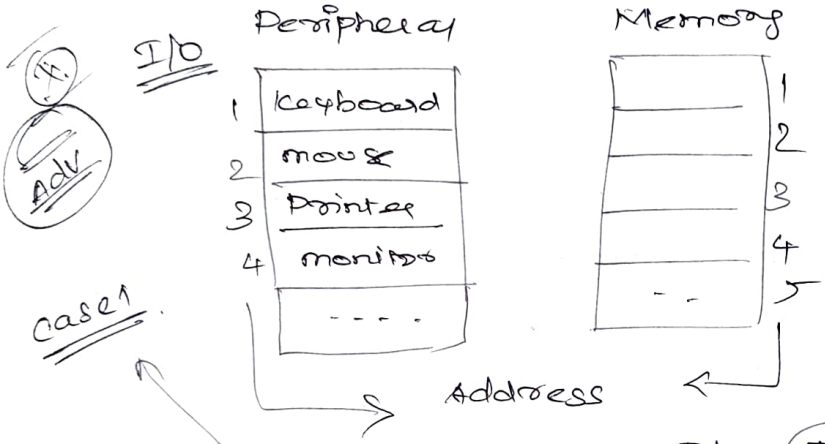
Higher Orders → Address

Size of the IC will be reduced. no. of wires are small

Addressing of memory & peripherals

Both MC and MP are connected by some peripherals and ~~address~~ some memories. So by default we need to assign some address for all. what if we process like assume 100 address are there, assume 20 address for peripherals and the remaining address for memory, in this situation it act as limitations, meaning we cannot add more than 20 peripherals & allocate more than 80 address for memory.

unrestricted addressing

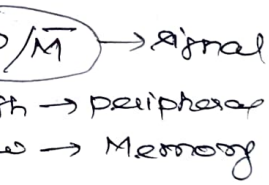


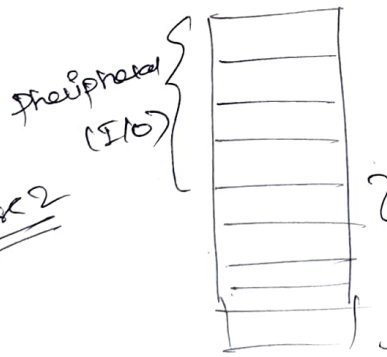
same no? for both peripherals & memory.

adv

case1

Memory mapped I/O





case2

⇒ IO Mapped memory (no need of using signal like IO/M, but the disadvantage here is usage of peripherals or memory is limited.

- Restricted addressing.

Blocks of MC and MP:-

- CPU → control processing unit
- I/O → Input/output
- Bus → Address bus / data bus.
- Memory → RAM & ROM
- Timer
- Serial & Parallel communication
- Interrupt.

Bit?

573 → 3 digit no
↳ character in a decimal no system

Decimal no. of system (0-9)

Binary no. system → 0, 1

↳ 1010 → 4 bit
101 → 3 bit
01 → 2 bit
↳ character in a binary no system

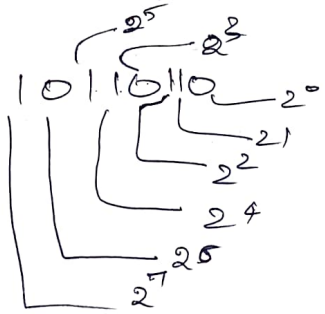
Bytes?

↳ group of 8 bits – 1 byte (10101011)

Group of 4 bits – 1 nibble (1010)

1 byte = 2 nibble = 8 bit

573 → Hundred (10^2), units (10^0), ten (10^1)



↳ we are giving position for the bits / binary

$2^0 \rightarrow 1 \rightarrow 1$
$2^1 \rightarrow 2 \rightarrow$
$2^2 \rightarrow 4$
$2^3 \rightarrow 8$
$2^4 \rightarrow 16$
$2^5 \rightarrow 32$
$2^6 \rightarrow 64$
$2^7 \rightarrow 128$
$2^8 \rightarrow 256$
$2^9 \rightarrow 512$
$2^{10} \rightarrow 1024 \Rightarrow$ 1kb
↳ lines/bits

$2^{11} \rightarrow 2k$
$2^{12} \rightarrow 4k$
$2^{13} \rightarrow 8k$

$2^{14} \rightarrow 16k$
$2^{15} \rightarrow 32k$
$2^{16} \rightarrow 64k$
$2^{17} \rightarrow 128k$

$2^{18} \rightarrow 256k$
$2^{19} \rightarrow 512k$
$2^{20} \rightarrow 1024k \rightarrow$ 1mb.

(X) 1024 → Address, we need 10 address lines
we can generalise

Word length/bus length ≠ data length

↓↓
no. of binary bits representing the data.

8085 → Data → 8 bit
word length → 8

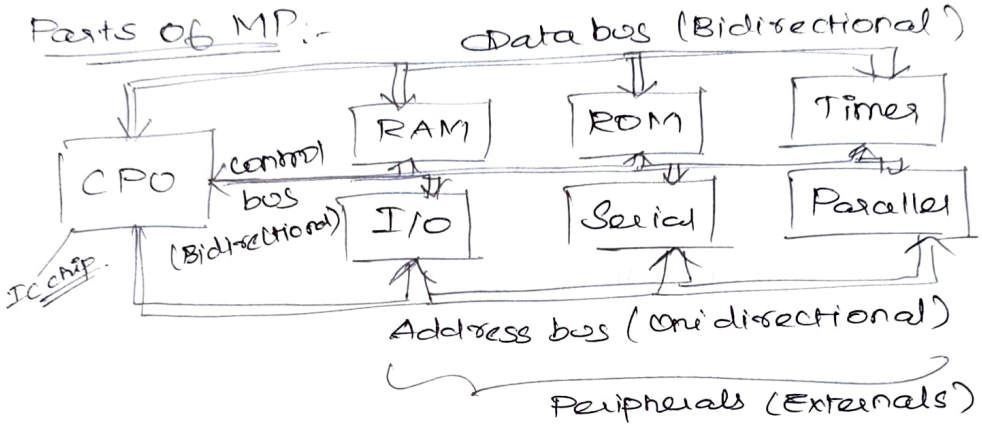
8086 → Data → 16 bit
word length = 16.

Port :-

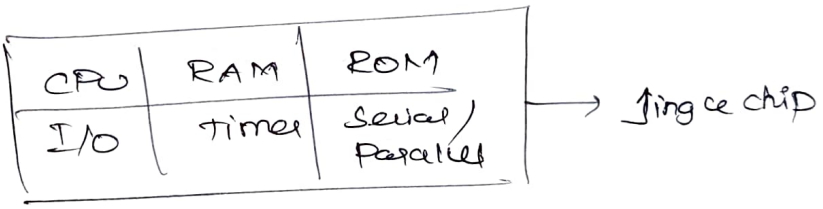
- area in which I/P & O/P devices are connected
- data plays important role
- serial port, USB-type A, Parallel, display port, HDMI, Ethernet port,
- IO ports

Serial → bit by bit (internet)
Parallel → group of bits (printer)

Parts of MP :-



Parts of MC :-



Difference b/w MC and MP :-

MP	MC
1. CPU - standalone RAM, ROM, Timer, Serial/Parallel ↳ separate	1. All are in 1 chip
2. Increase the size of memory RAM & ROM	2. Fixed
3. General purpose	3. Specific
4. Expensive	4. Not
5. Versatility	5. -

Applications of MC :-

- Home appliances ← TV
  - Fridge
  - AC
  - cellphone
- Office – camera,
  - printer
- Mechanical – Automatic machines
  - Air bags in car
  - Door closing in car

8051 MC family

- 8051
- Zilog ~~Z80~~ Z8 series
- Motorolla 6811
- PIC

} 8 bit MC

8051 Microcontroller

- MC51 series Intel, 1981 (MCS-51)
- Atmel 8051

8051 Family

	(ROM)	(RAM)	(IO Ports)	(Timer)	(Serial port)
- 8051	– 4K	– 128bytes	– 32 lines	– 2	– 1
- 8052	– 8K	– 256bytes	– 32 lines	– 3	– 1
- ~~8053~~ 8031	– 0K	– 128bytes	– 32 lines	– 2	– 1

(ROM-less)

(*) 8051 is a subset of 8052.

ROM & RAM :-

- ROM (Read Only Memory) / code memory
  - ↳ only Read not to edit ⇒ 4K
  - Storing the program (Specific Application)
- RAM (Random Access Memory) / Data memory
  - Read & write
  - 128 Bytes
  - Data essential for the program (ROM) will be stored.

Ports:

- 4 ports
- Each port is 8 bit / 8 line
- 32 lines used for I/O devices

Timer / counter

- Timer 0
- Timer 1

Serial port

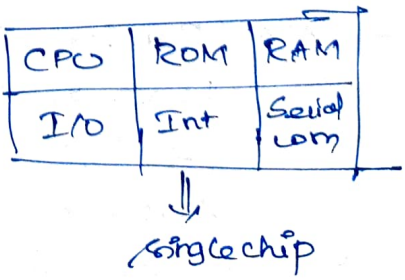
- Bit by bit communication

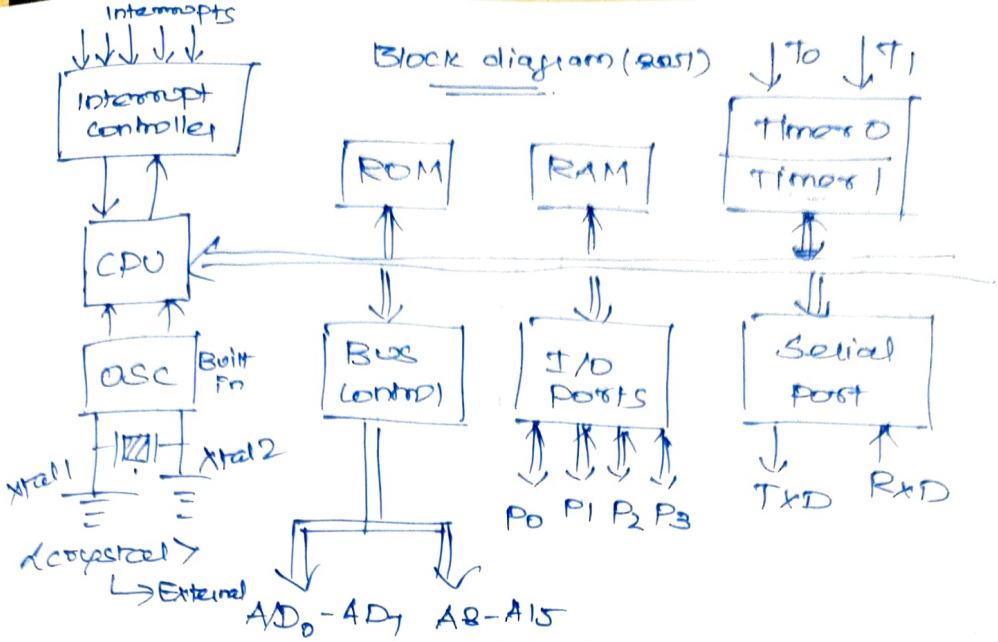
Features

- 4K built in ROM/code/program memory.
- 128 bytes of built in RAM/data memory
- 4 I/O ports, each port 8 pin (P0, P1, P2, P3)
- Full duplex UART (universal asynchronous Receiver and Transmitter)
  - TV, Radio → ~~Half duplex~~ Duplex
  - Walkie talkie → Half duplex
  - Cell phone/Smart → Full duplex
- 6 sources / 5 vectored interrupt
- 64K of External ROM can be connected.
- 64K of external RAM can be connected

8051 - Basic components

- 4K ROM
- 128 bytes RAM
- 4 I/O Ports (P0-P3)
- 6/5 Interrupt
- Serial communication



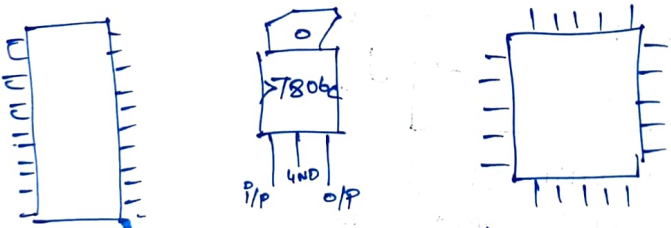


Why oscillator?

- Time required to perform certain operation
- generates pluse

Pin-details

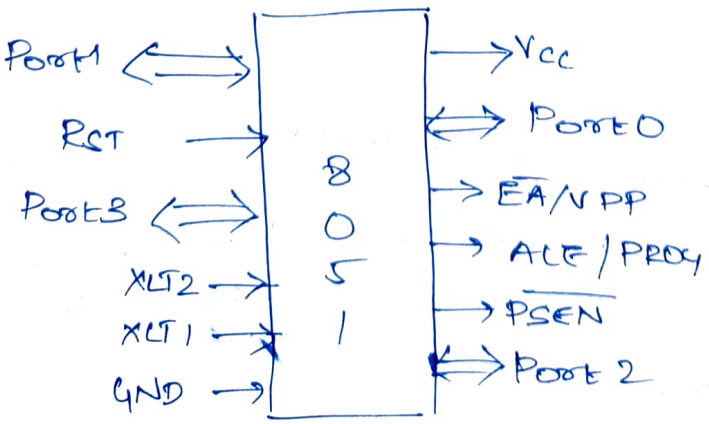
- 40 pin
- DIP configuration of IC (Dual In package) ⇒ 

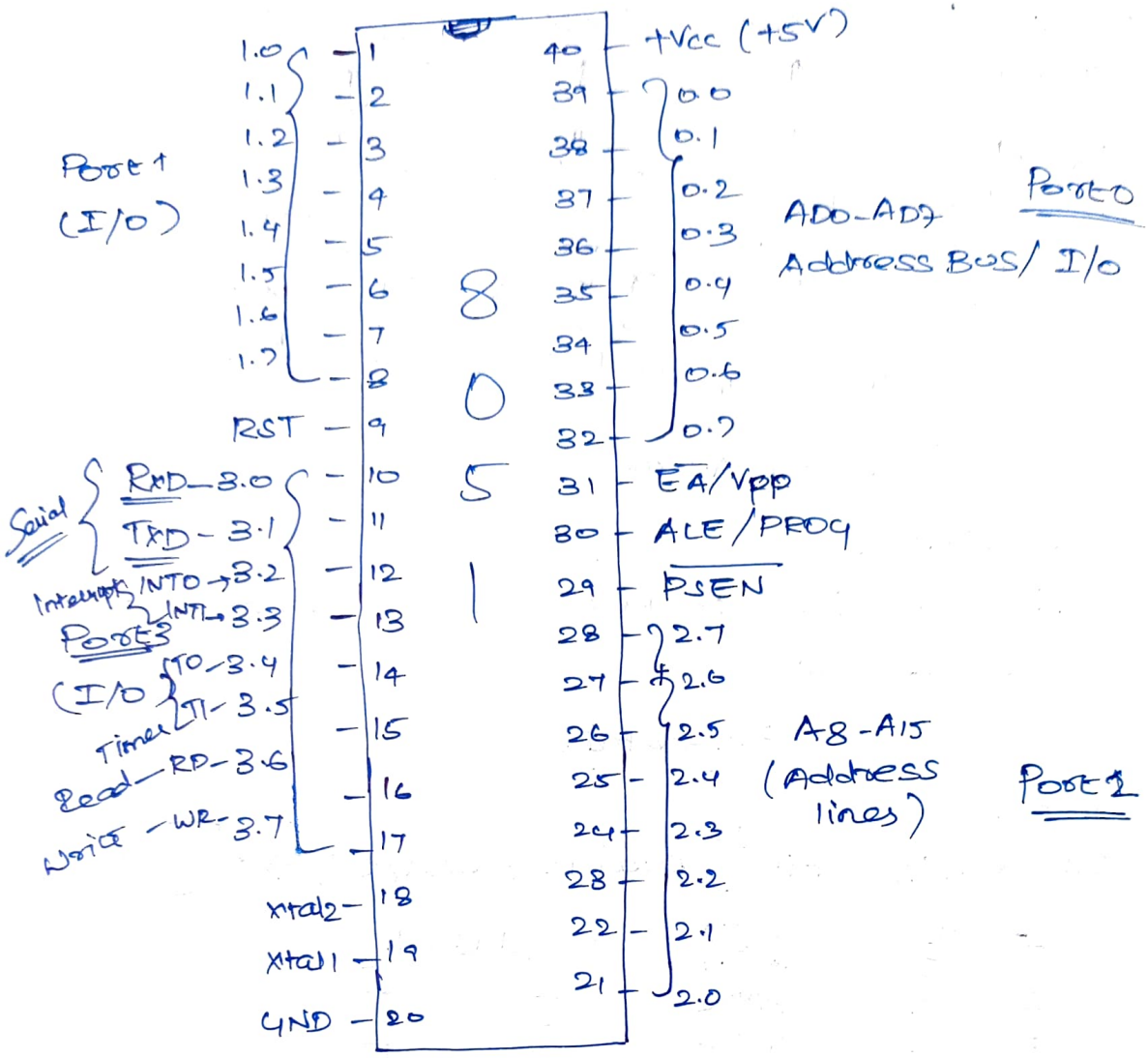


DIP (Dual In Package) — Born side

SIP (Single In Package) — one side

QIP (Quad In-line Package) — 4 side





- Power supply ⇒ GND (20)
  +Vcc (+5V) (40)
- Crystal Oscillator connection (2 pins) ⇒ Xtal 1 (19), Xtal 2 (18) ⇒ In order CPU to work
- Reset (RST) ⇒ (9)
- Port 1 → 1 - 8
- Port 3 → 10 - 17
- Port 0 → 32 - 39
- Port 2 → 21 - 28 } external memory
- $\overline{PSEN}$ - program status enable pin (29) [0 - ROM, 1 - RAM] (External)
- ALE - Address Latch enable [0 - data, 1 - Address] ⇒ Port 0
- EA - External Access [0 - External memory, 1 - Internal memory]