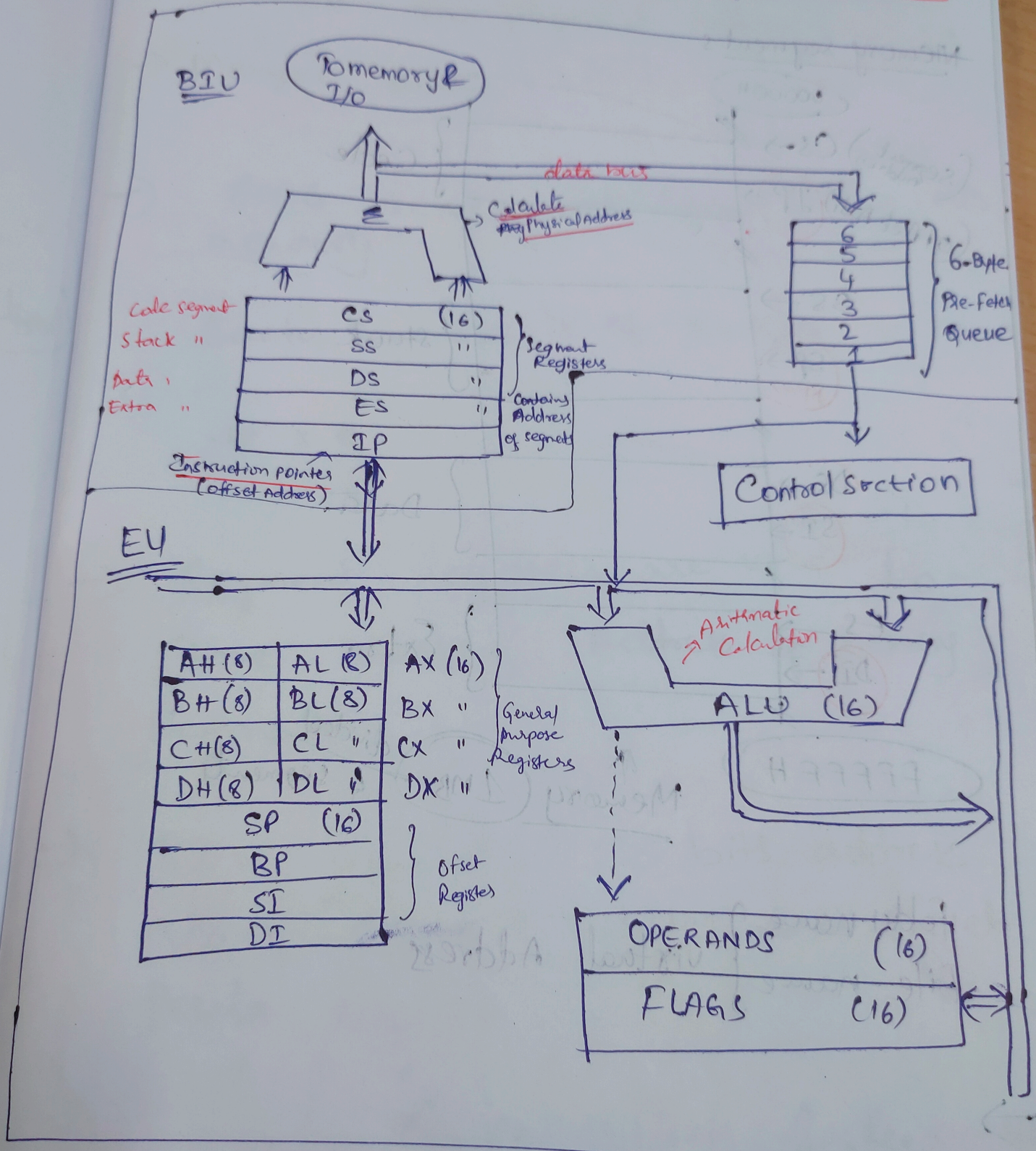
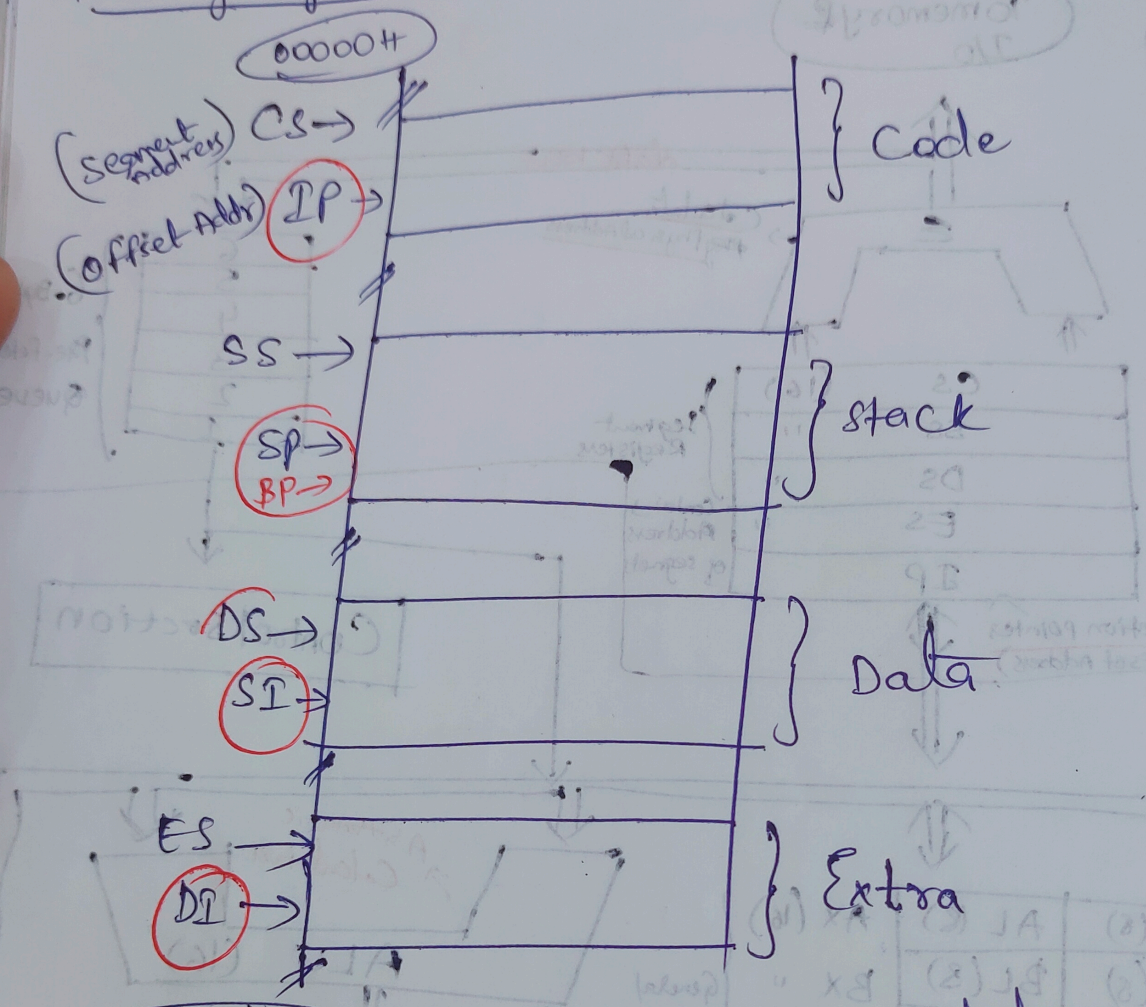
UNIT- III

Introduction of 8086 Architecture



Physical Address = Segment Address $\times 10_H \times$ Offset Address

Memory Segments



Memory (1 MB) divided 4 segments

Folder name
File name
} virtual Address

BIU

→ Fetch the Next Instruction

→ Calculate the physical Address

→ Manges Queue

→ 8086 Microprocessor have 2^{20} ⇒ 1MB memory.

Each segment is of 64KB

~~64~~ x ~~8~~

→ Segment Registers are used to Store Starting Address of memory Segment

⇒ BIU generates 20 bits address Using Segment register and offset Pointer registers

Physical Address = Seg Reg. X10+Offset

(I) Bus Interface Unit (BIU)

Responsible for Establishing Communications with External Peripheral devices and memory Via system bus.

1) It fetches the instructions from memory
2) It Reads the data from I/O and memory
3) It writes the data into I/O & memory
4) It provides the Address Relocation facility

BIU have 3 parts

① segment Registers
⇒ ② Instruction Queue → Pre-fetches Instructions from memory → (8086 uses pipe lining. fetching & Execution done in parallel
③ Instruction pointer

→ Stores in instruction Queue

→ FIFO

→ BIU & EU Operates Simultaneously

* The process of fetching the next instruction in Advance while the EU is Executing the current instruction pipeling

* Segment Registers.

⇒ 8086 microprocessor has a capability to store (i.e) 2^{20} bits ⇒ 1 MB memory which is divided into 16 bit segment ⇒ 1MB/16

⇒ Each segment contains 64KB memory

Physical Address = 20 bits
Segment Register = 16 bits
Offset pointer = 4 bits

Physical Address = S.R × 10 × OP

II EXECUTION UNIT (EU)

EU informs BU from where the Next instruction (or) data to be fetched

→ It picks instruction from instruction queue in BIU

→ It decodes and the Executes the instruction

→ And it updates the status of flag Registers

* Control unit

→ It is Controlling and coordinating all the Activities of Sub units.

→ It fetches, decodes and Executes Instructions.

→ It gives Control signals like Read, write etc

Design of Control unit

1) Control unit is a Component of CPU that directs the Operations of processor

→ All the Computer resources are managed by the CU (Control unit)

→ CU Generates the Control signals to perform the Operations

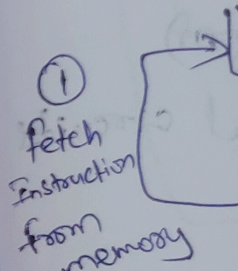
→ It also instructs the ALU, which Operation should be performed on data

Components

1) Sequ

2) Regist

3) Con



* Des

Co

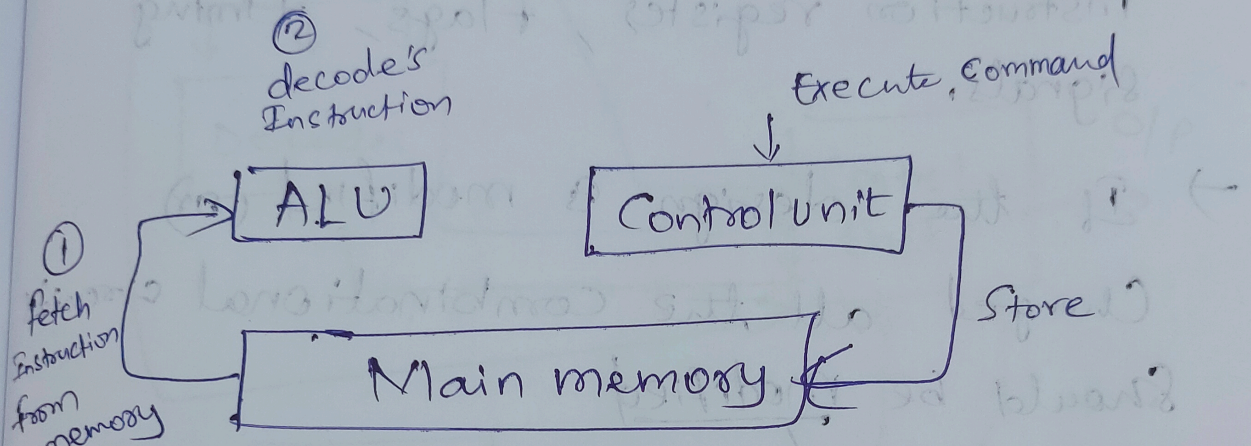
t

1

2

Components of Control unit

1) Sequencing logic
2) Registers, decoders of CPU
3) Control memory



* Design of Control unit

Control unit can be designed in two methods

① Hardwired Control unit

② Micro-programmed Control unit

① Hardwired Control unit

→ It is implemented with the help of physical components which is hard-wired (Gates, Flipflops, decoders) in the hard ware.
(Gates → AND, OR, NOR, XOR, NAND; Flipflops → 0 or 1)

→ The Input to Control unit are the instruction register, flags, timing signals.

→ If the design is modified (or) Changed all the combinational circuits should be modified

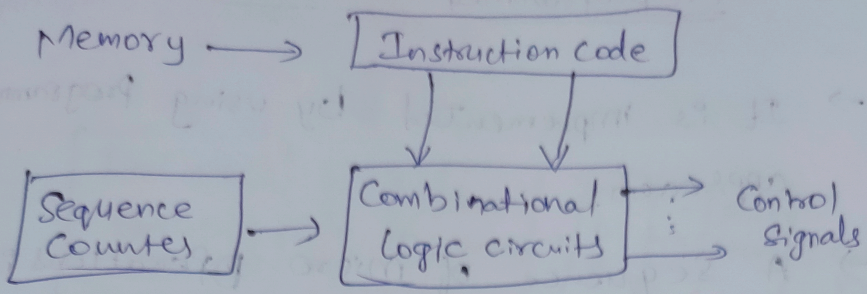
→ It is very difficult

* The Sequence of the operation Carried out by this machine is determined by Wiring of the logic Elements

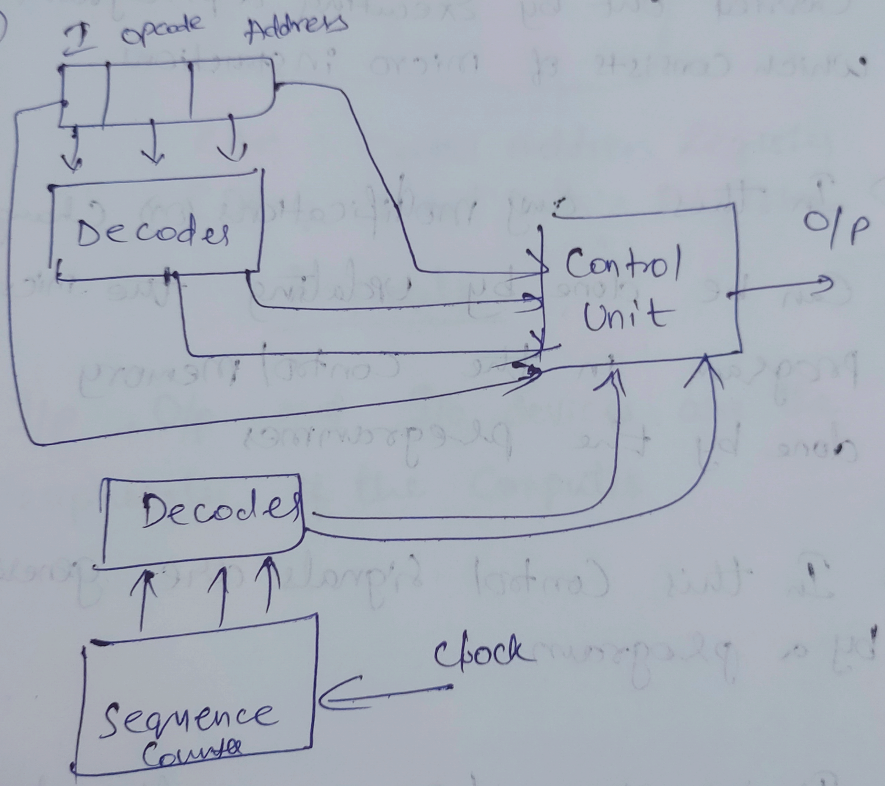
So, it is known as "Hardwired".

H/w CU

①


Memory
Instruction code
Sequence Counter
Combinational Logic circuits
Control Signals

②


I
opcode
Address
Decoder
Control Unit
O/P
Decoder
Clock
Sequence Counter

(2) Microprogrammed Control unit

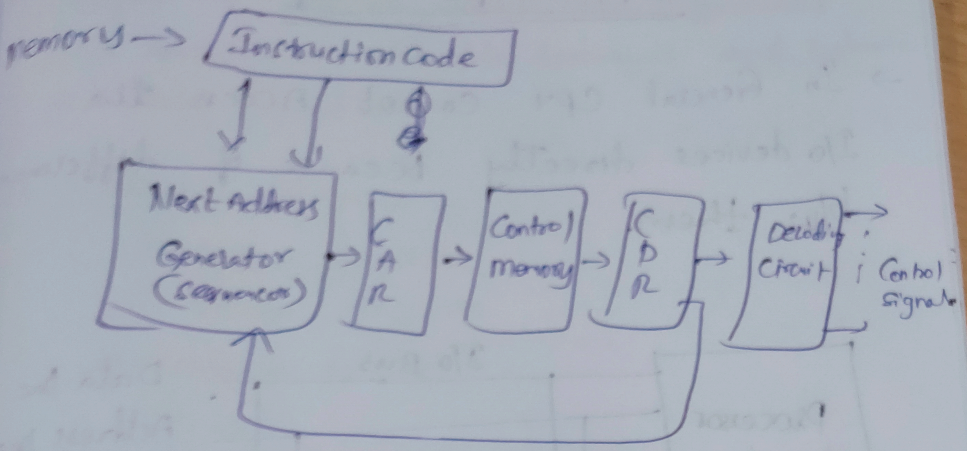
→ It is implemented by using Programming approach.

→ A sequence of micro operations is carried out by executing a program which consists of micro instructions.

→ In this any modifications (or) changes can be done by updating the micro program in the control memory done by the programmes

→ In this control signals are generated by a program

→ It is slow, because of the time it takes to fetch microinstructions from control memory.



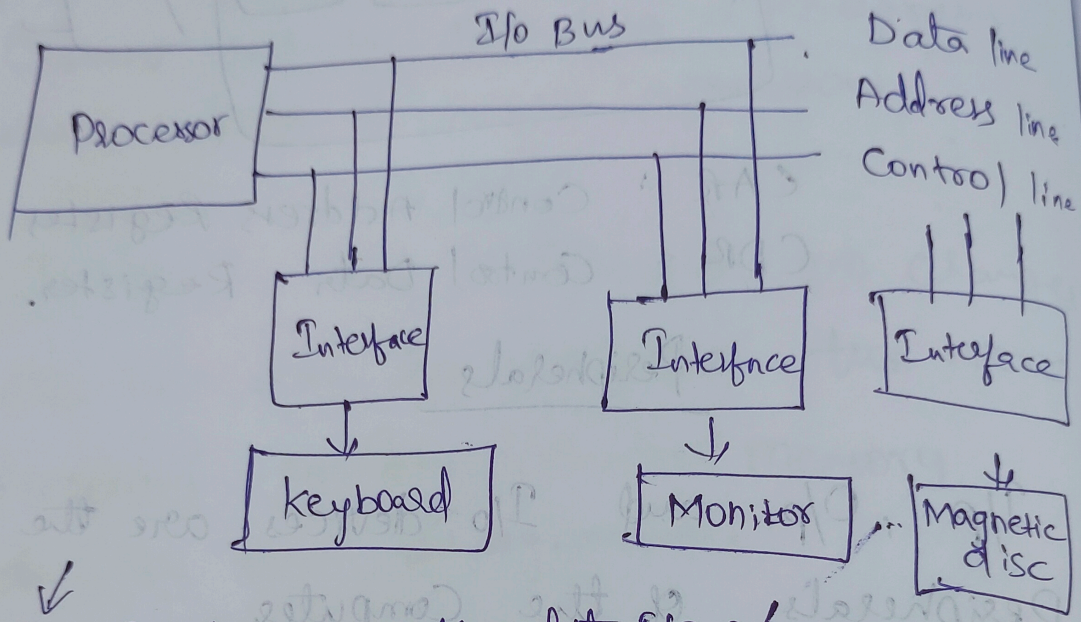
CAR : Control Address Register
CDR : Control Data Register.

Peripherals

I/P , O/P and I/O devices are the Peripherals of the Computer.

Input-output Interface

→ In General CPU cannot Access the I/O devices directly because of difference b/w them.



→ It synchronises the data flow/communication

* In Order to Resolve those differences We use Interface, so, that the CPU Can Communicate with I/O devices

Differences b/w I/O devices and CPU

Electro Mechanical / Electro magnetic	Electronic device
(1) I/O devices are slower (Data transfer)	(1) faster (Synchronise mechanism)
(2) Data format is Bytes (codes)	(2) Word (Executes Instruction in word)
(3) Serial manner Byte/Byte	(3) parallel manner Executed simultaneously

* Interface is used to resolve the differences and synchronises the data transfer and Balance the data format etc

→ Interface Contains

① Buffer Registers
② Status Registers
③ Address decoder

There are 4 types of I/O Commands to Execute the instruction.

Status Register → 1) Status Command → CPU uses this to know the Status
2) Control Command → for Read and Write Which operation to be performed
3) I/P Command
4) O/P Command

* I/P Command is used to store data in the interface

* O/P Command is used to get the data for the CPU.

I/O Interface

Commands Four commands that interface Receive

① Control command:

It is issued to activate the peripheral and inform it what operation should be done (sent by Interface)

② Status Command:

It is used to test various status Conditions which peripheral is idle (or) busy and Interface Status

③ Data O/p Command:

~~Causes the~~

By sending this data O/p Command processor causes the interface to Respond by transfering data from bus into one of its registers

④ Data I/p Command

It is opposite to data output.

→ In this Case the interface receives an data from peripherals and place it in its buffer registers.

→ The processor checks if data are

Available by meanse of a status command and then issues a data input command.

→ The Interface places the data on the data lines, where they are accepted by the processor.

CPU Can Access Memory as well as I/O

① Isolated I/O

② Memory I/O.

Isolated I/O