

* Cache Initialization

* Cache Write through & cache writeback.

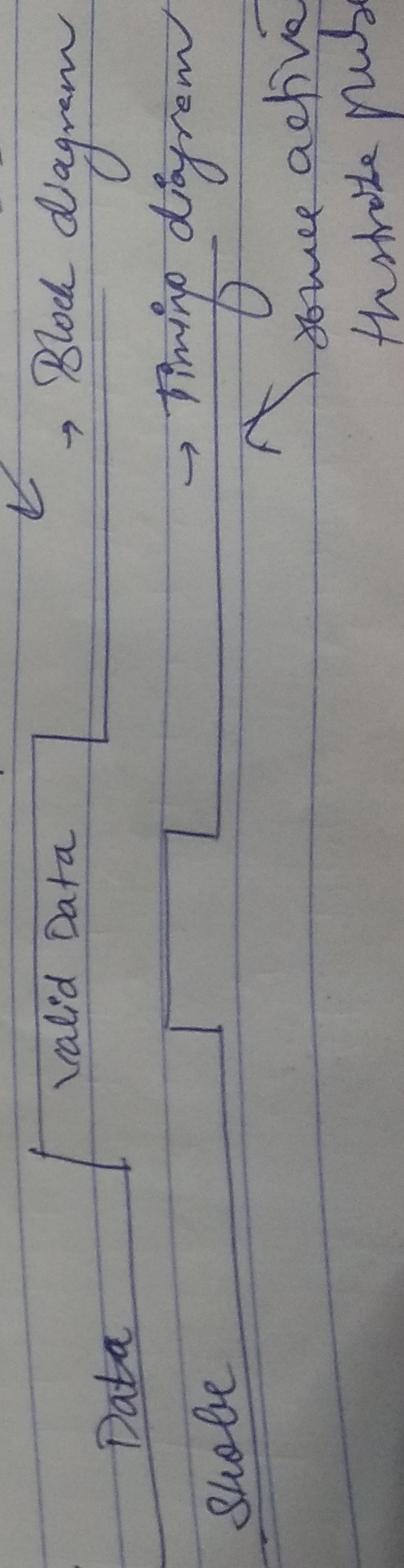
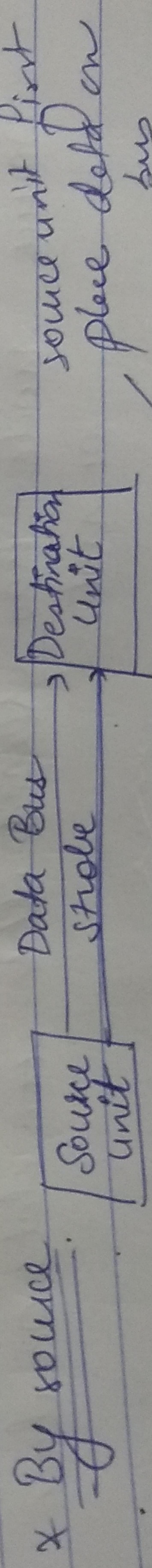
- * interrupt mechanism.
- * branch instruction call subroutine & program interrupt & interrupt of 5 external interrupts, 5 bits of internal interrupt (of no interrupt subroutine)
- * Input Output Organization — Asynchronous Data Transfer

- * This scheme is used when speed of I/O devices do not match with microprocessor, and timing characteristics of I/O devices is not predictable.
- * In this method, process initiates the device & checks its status.
- * CPU has to wait till I/O device is ready to transfer data.
- * When device is ready CPU issues instruction for I/O transfer. In this method 2 types of techniques are used based on signals before data transfer.

1. Strobe Control 2. Handshaking

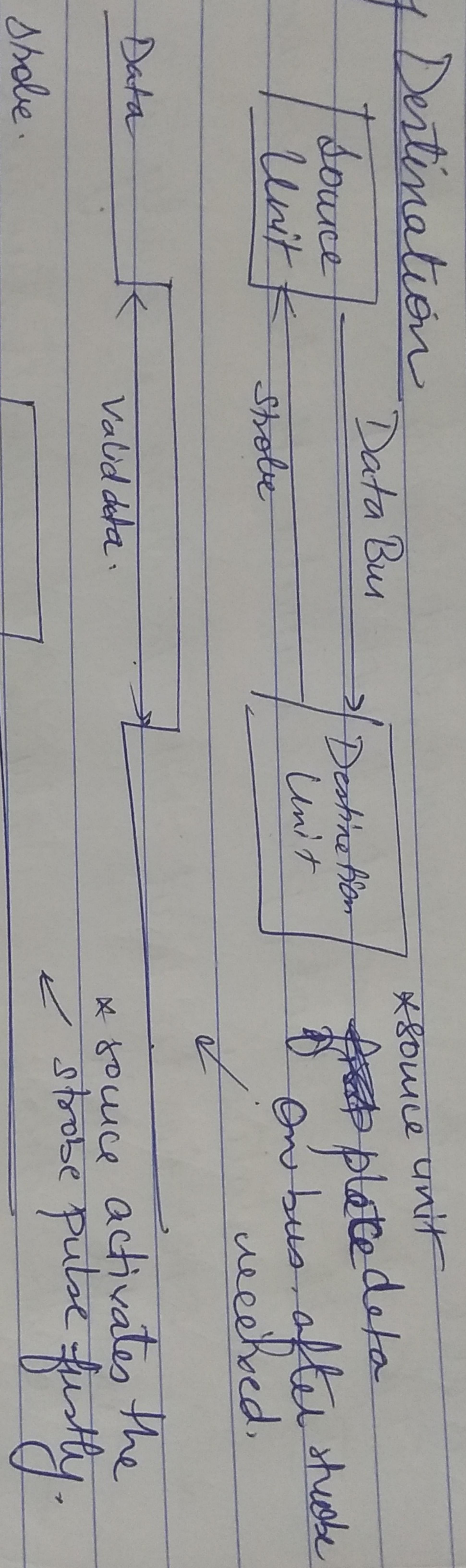
Strobe Control : \Rightarrow

- * It employs a single control line to time each transfer.
- * The strobe signal can be activated by either a source or a destination unit.



- * The data bus carries binary info from source to destination unit
- * The bus has multiple lines to transfer an entire byte at once
- * The strobe is a single line that informs the destination unit when a valid data word is available.

* By Destination



- ② The destination unit activates the strobe pulse, to inform source to provide the data. The source will respond by placing the requested binary information on the data bus.

Disadvantage of Strobe Signal

- * source don't does not know whether destination unit has received the data item placed in the bus.
- * destination unit does not know whether some node has placed the data on bus or not.

Handshaking

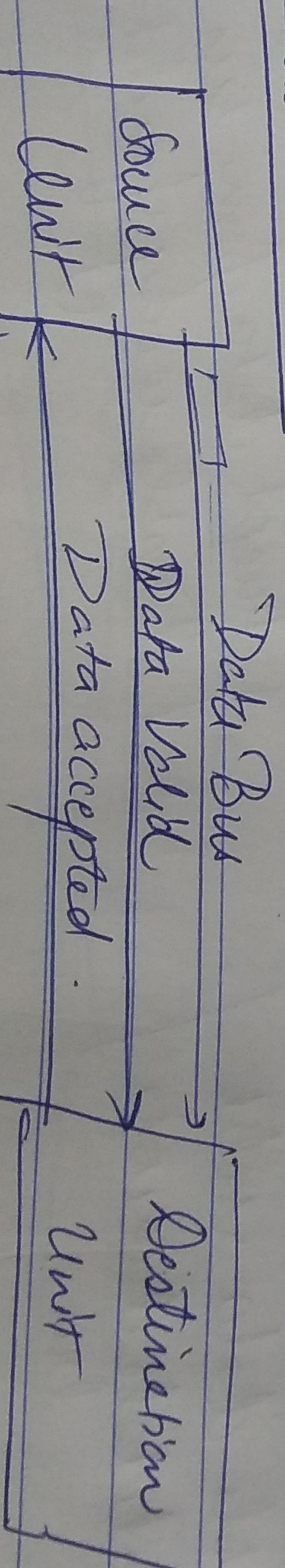
- a second control signal is introduced that provides a reply to the unit that initiates the transfer.

Principle of Handshaking :-

The basic principle of 2-wire handshaking method of data transfer is as follows:-

- * One control line is in same direction as data flows in the bus from source to destination.
- * It is used by source unit to inform the destination unit whether there is a valid data in the bus.
- * The other control line is in the other direction from the destination to the source.
- * It is used by the destination unit to inform the source whether it can accept the data.
- * The sequence of control during the transfer depends on the unit that initiates the transfer.

* Source initiated →



- * Source unit places data on data bus & enables data validation
- * When destination unit accepts data, it activates data accepted
- * The source disables signal & system goes into initial state'

Source unit

Places data on bus
Enable data valid

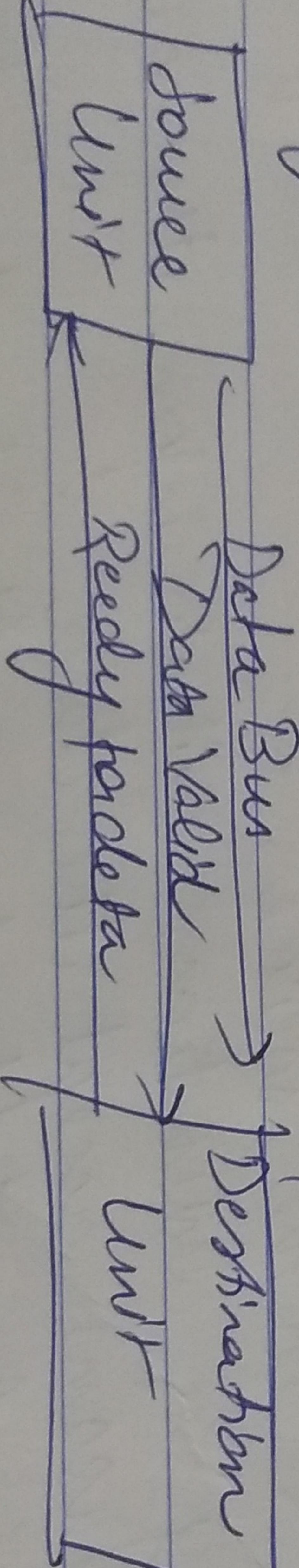
→ 'Accept data from bus'
Enable data accepted.

Disable data valid
Invalid data on bus

→ 'Disable data accepted,'
Ready to accept data

* Destination unit initiated

- * destination unit tells → ready ~~to accept~~ for data to reflect.
- * source unit in this case does not place data on the bus until after it receives the ready for data signal from the destination unit.
- * Diff b/w source initiated & destination initiated transfer is in the choice of initial state



Ready to accept data
Enable ready for data

Disable data valid
Invalid data on bus
Invalidate data on bus
Invalidate bus

Advantage of Handshaking method

- * The handshaking scheme provides degree of flexibility & reliability bcz successful completion of data transfer relies on active participation by both subits.
- * If any one is faulty, data transfer will not be completed. Such can even be detected by means of a timeout mechanism where Microcles can alarm if data is not completed within time.

Modes of Data Transfer

- * All the internal operations in a digital system are synchronized by means of clock pulses supplied by common clock pulse generator.
The data transfer can be
① Synchronous or ② Asynchronous.
 - ① When both the transmitting & receiving units use same clock pulse then such data transfer is called synchronous process.
 - ② If there is no concert of clock pulses and sender operates at diff instant then the receiver which data transfer is called asynchronous data transfer.

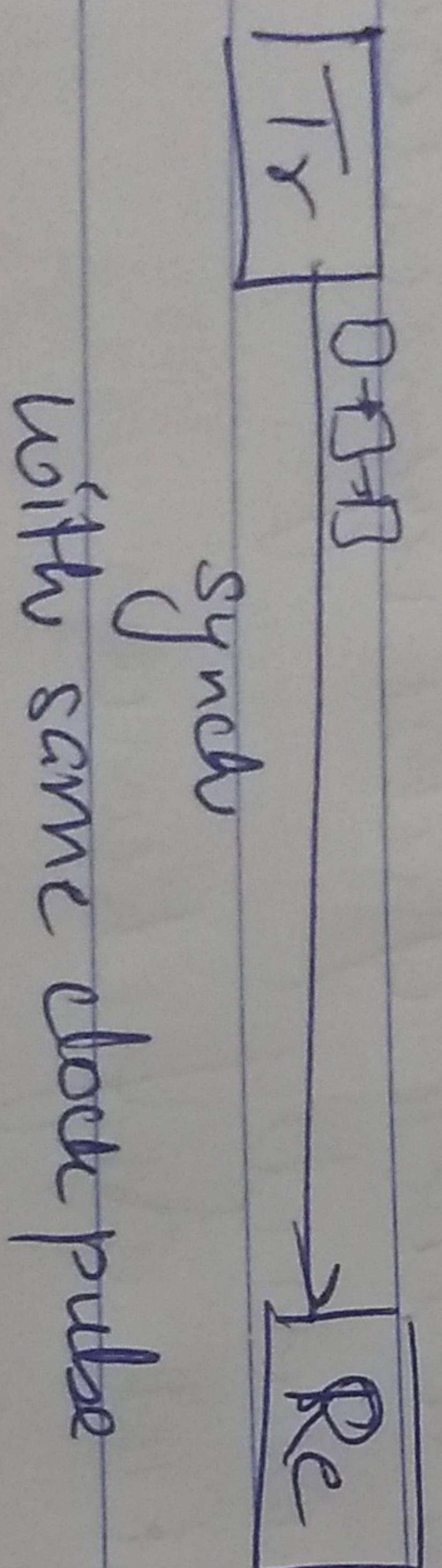
The data transfer can be handled by various modes, some of the modes use CPU to handle intermediate path, others transfer the data directly to & from the memory unit & this can be handled by 3 ways.

- i) Programmed I/O
- ii) Interrupt-initiated I/O (DMA)
- iii) Direct Memory Access (DMA)

I/O Organization

1. Synchronous Data Transfer

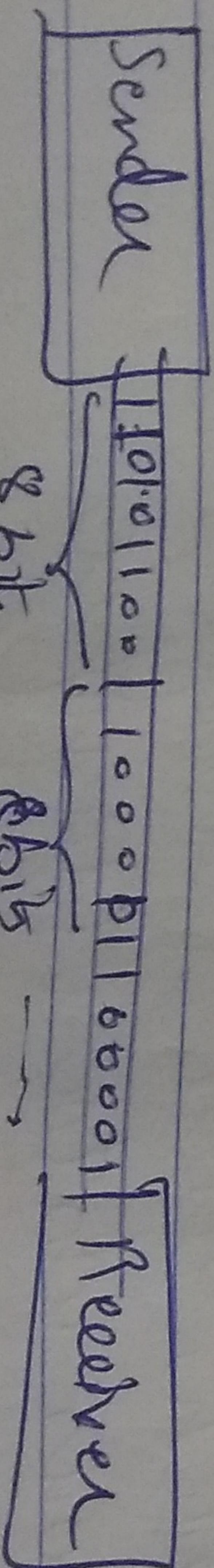
at same time



with same clock pulse

- timing signals
- 8bit → 1byte (Receiver)
 - ↓
 - Count no. of bit set

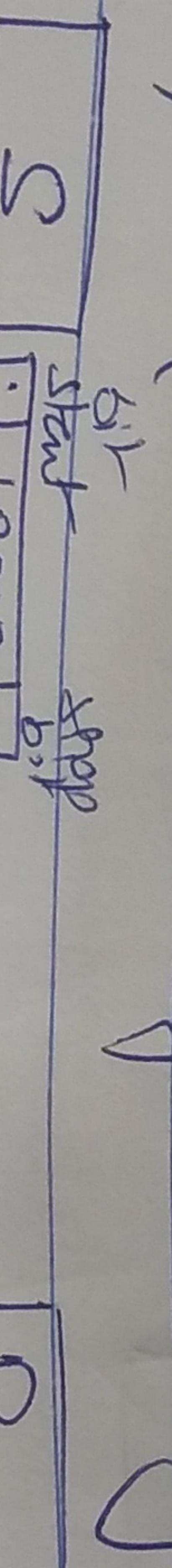
- timing must be accurate
- fastest method
- without start bit or stop bit



Note → In synch. trans. we send bits one after another without start / stop bits or gaps. It is responsibility of receiver to group the bits.

Synchronous transfer

- * regular intervals
- * 1 start bit ('0') at beginning
- * 1 stop bit ('1') at end of each byte.



- * In the transmitter & receiver data bytes at any instant of time.
- * Only 1 byte is sent at a time. There is fixed time b/w 2 data bytes.
- * Transmitter & receiver operate at different clock frequencies.
- * To help receiver start & stop bits are used along with data in middle.

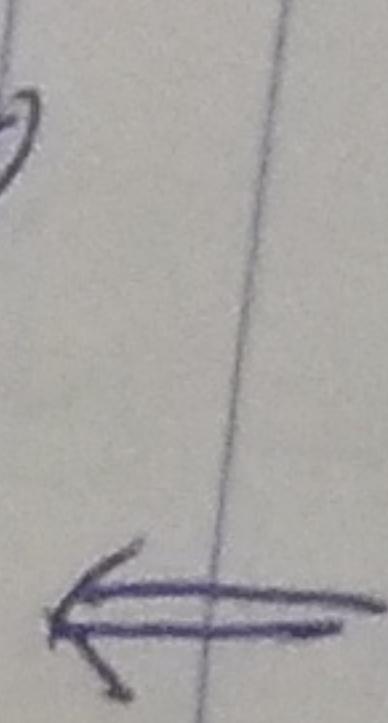
Strobe control

destination source initiated

Handshaking

source destination initiated

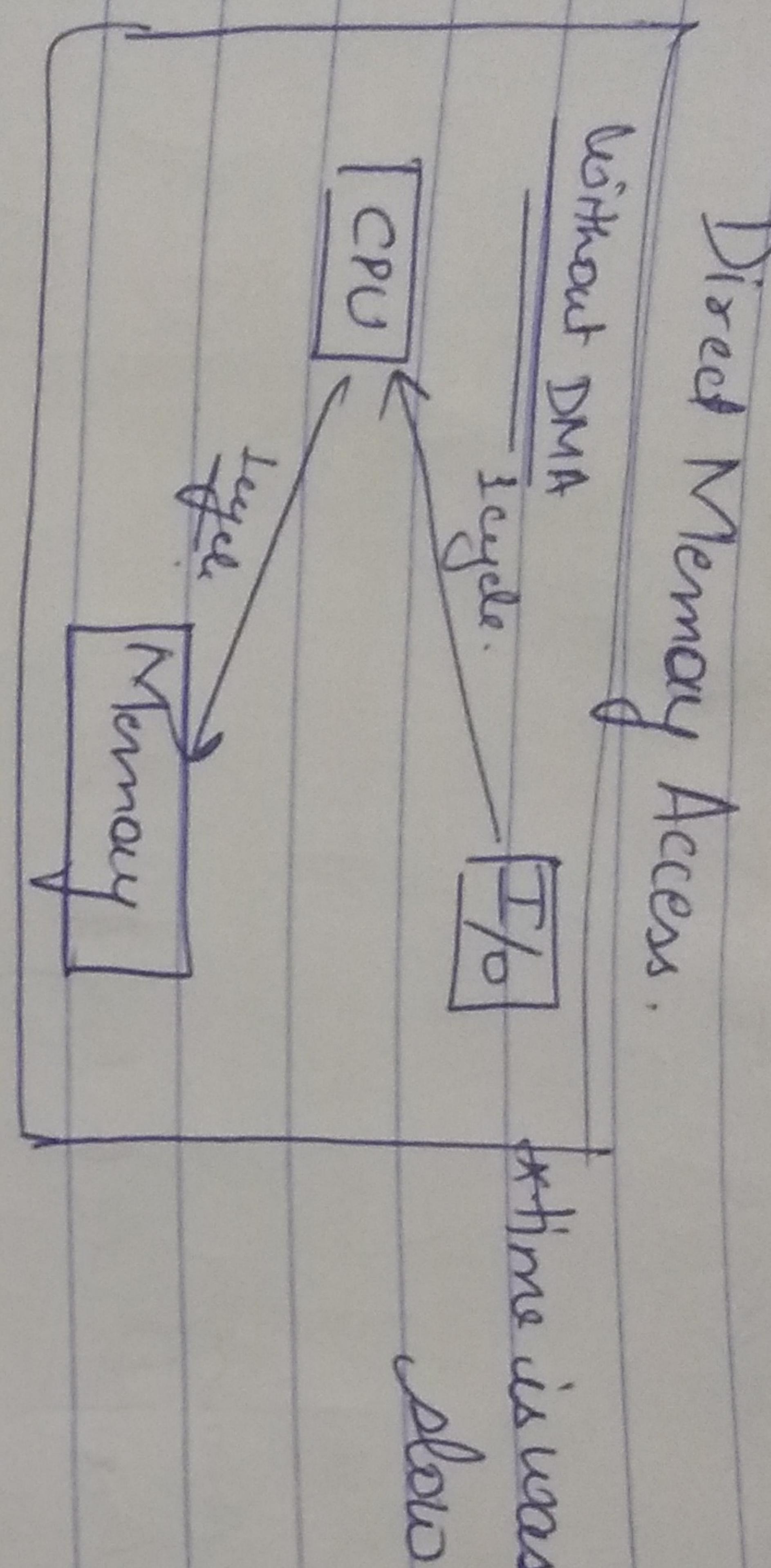
DMA Controller



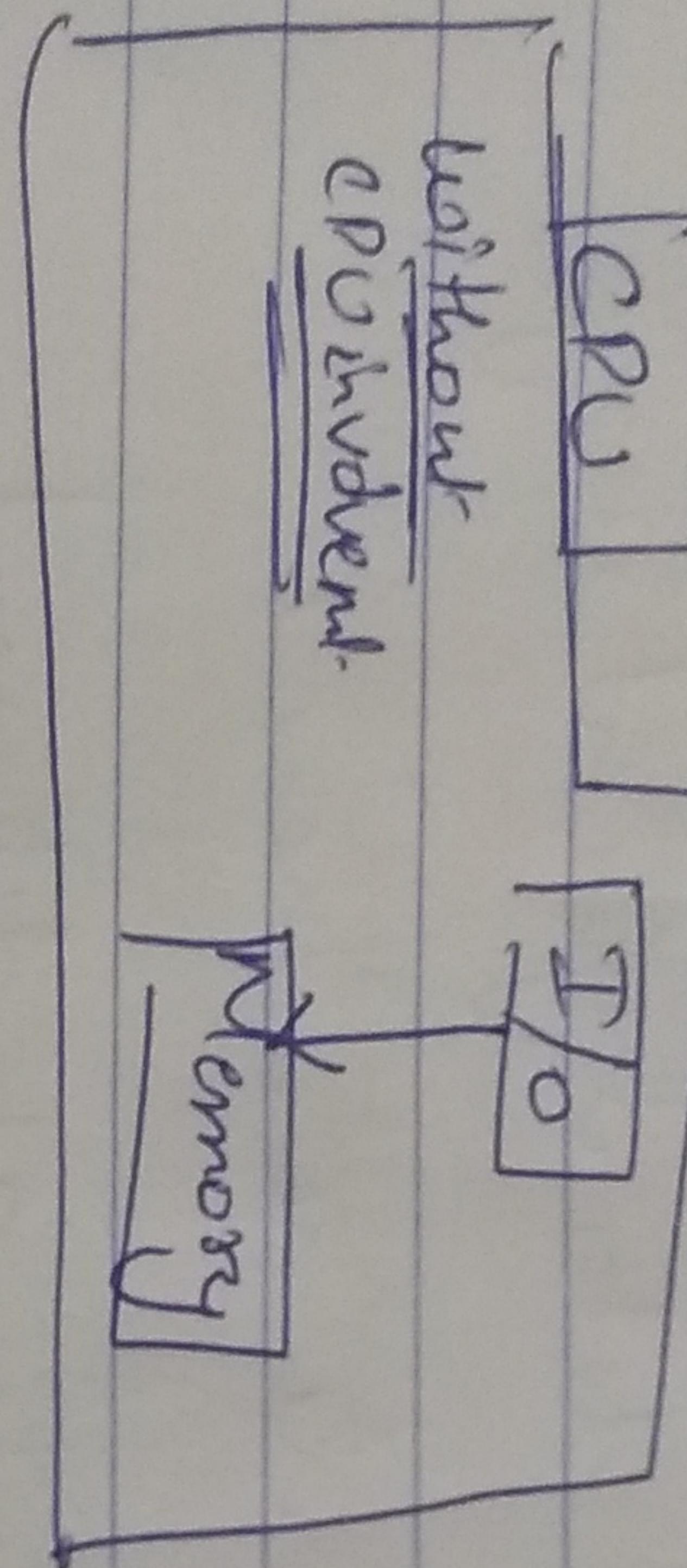
Direct Memory Access.

Without DMA

cycle.



With DMA



* DMA is designed by Intel to transfer data at the fastest rate.

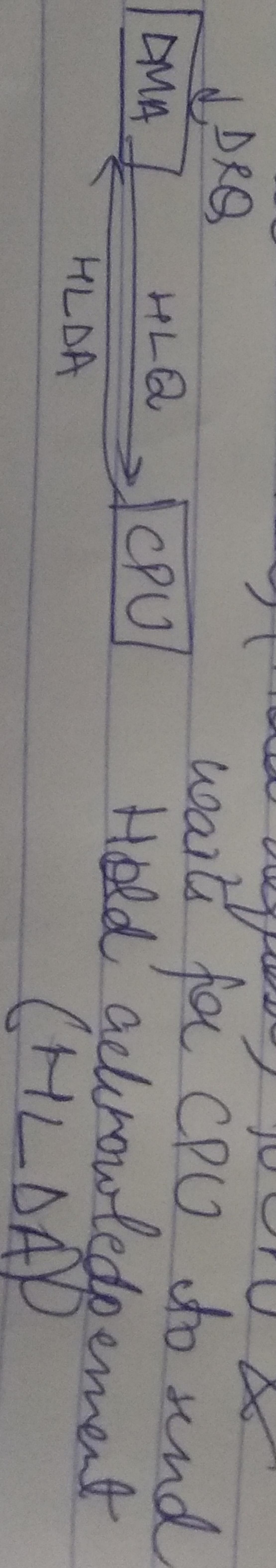
* It allows the device to transfer data directly to/from memory without CPU involvement.

How DMA are performed?

1. I/O device wants to transfer the data to memory.

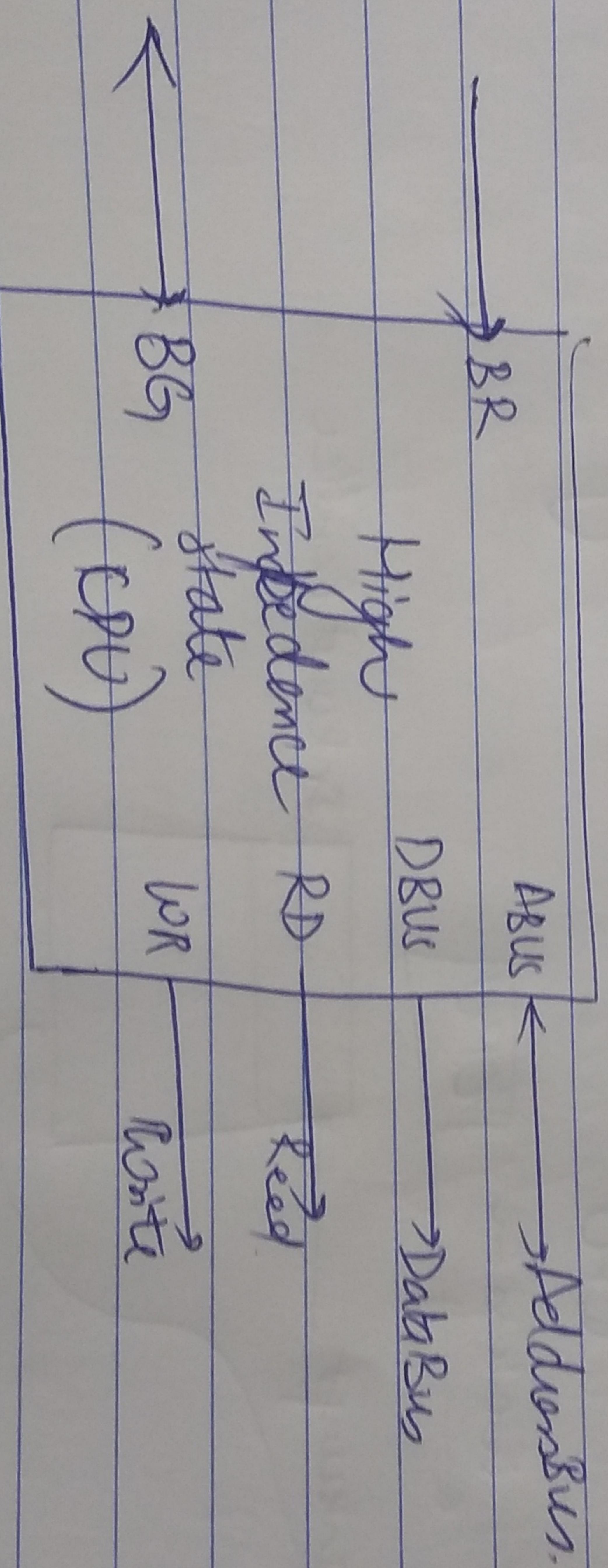
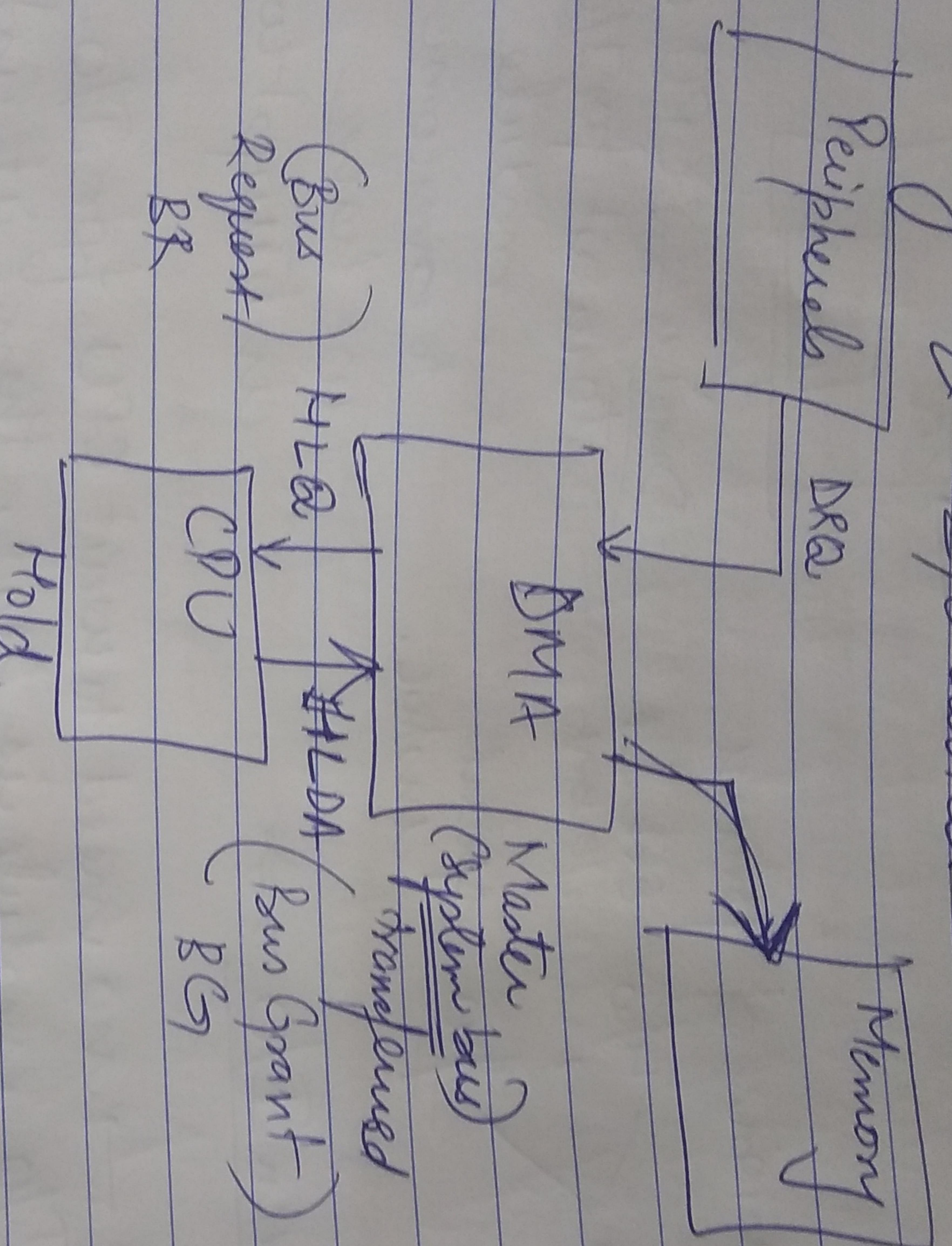
2. First device has to send DMA request (DRQ) to CPU.

3. DMA controller sends HLDA (Hold request) to CPU &



③ The CPU loses the control over bus & lock the MDA signal to DMA controller.

④ Now CPU is in HOLD state, DMA controller has to manage operation over buses & CPU, memory & I/O devices.



I/O Processor

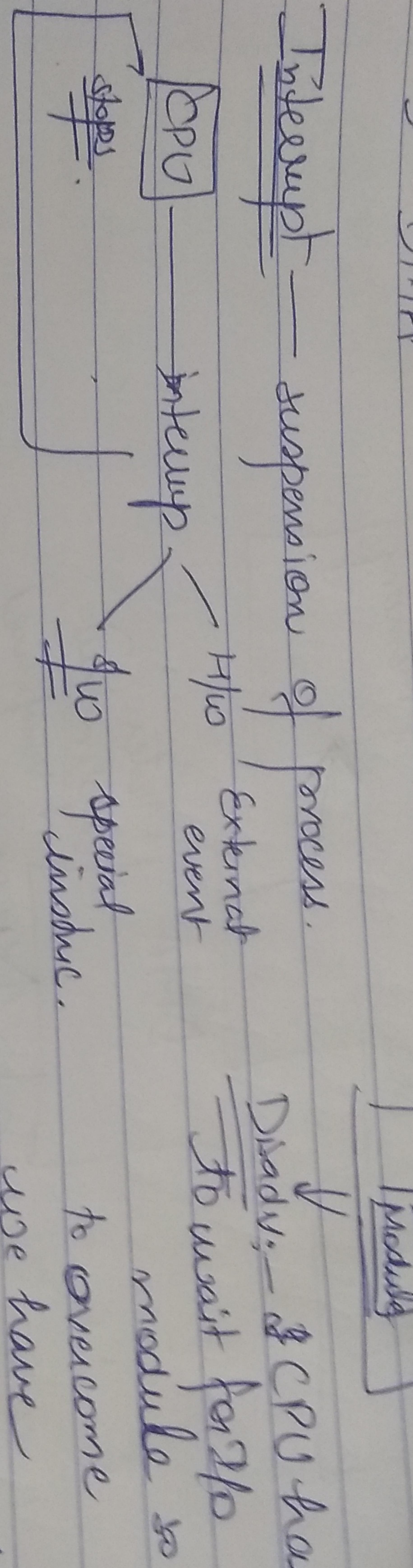
- * It is processor with DMA capability that communicates with I/O devices.
- * IOP is similar to CPU except that it is designed to handle the details of I/O operations.

Interrupt I/O

3 techniques used for exchanging data or transfer the data

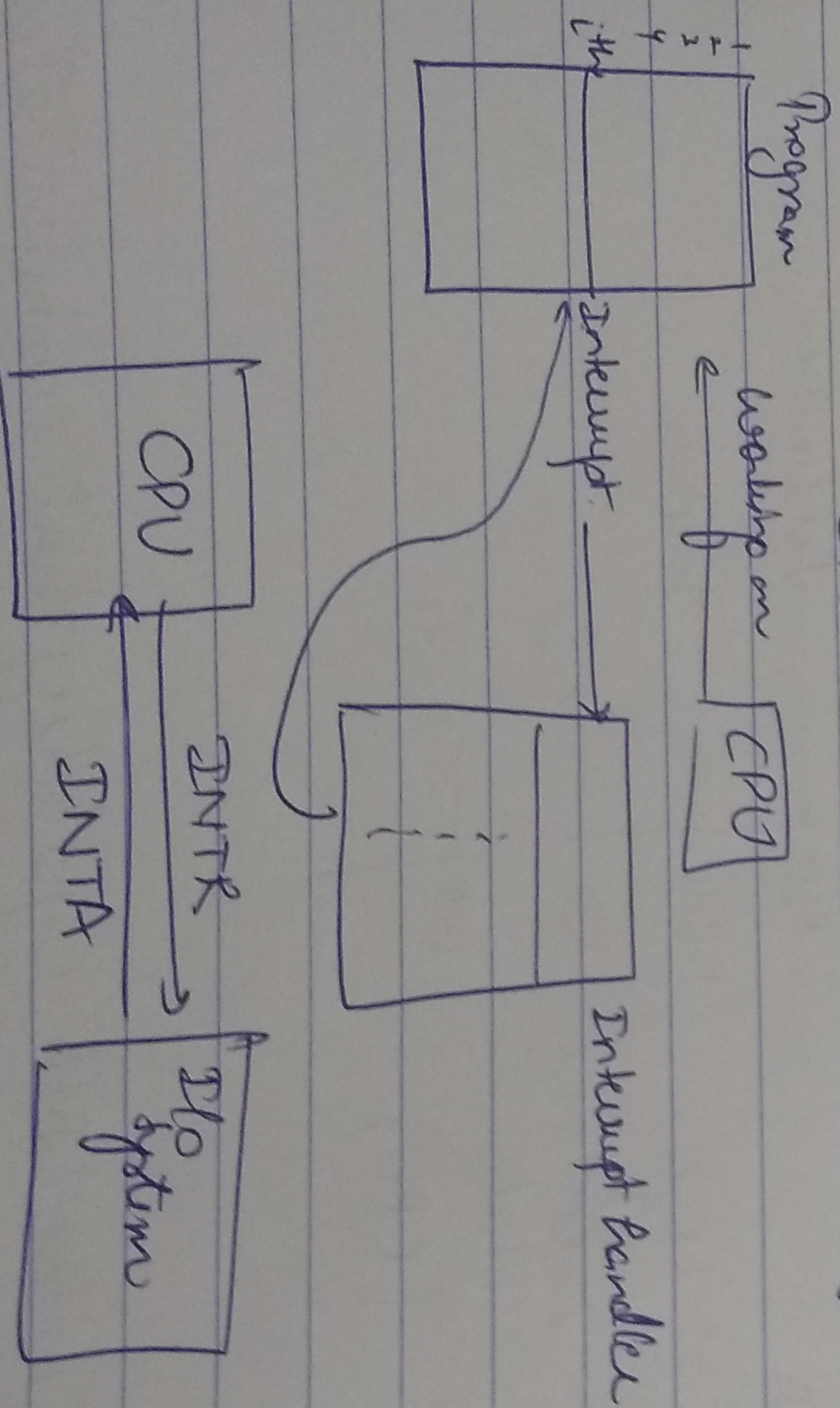
1. Programmed I/O
2. Interrupt driven I/O
3. DMA

Interrupt — suspension of process.



2. Interrupt driven I/O :=> interrupt controller.

Interrupt Controller → no need to wait CPU for I/O.
I/O module is ready then it interrupts CPU.



Q. If 50 kbps device is connected to the process. The interrupt overhead is 50 μ sec. What is min. performance achieved when interrupt initiated data transferred is used instead of programmed I/O.

Q. five external & five internal interrupts.
Diff b/w sw interrupt & subroutine call.

External: — Initiated by H/W of system.

- * I/O devices
- * Power failure interrupt
- * Request transferring of data by I/O device
- * Time that elapsed by an event
- * Timeout interrupt

Internal: — when an illegal instruction occurs & also known as traps.

- * Stack overflow
- * Stack underflow
- * Divide by zero
- * By invalid operation code
- * Violation of protection

Subroutine is portion of a code within larger program which performs a specific task & is relatively independent of remaining code. It runs when you call it.

How to determine who initiated interrupt?

- * For processing I/O device interrupt there is special dedicated chip (called) as PIC (Programmable Interrupt Controller)
- * When a hit interrupt through Keyboard, it will be initially sent to PIC & PIC sends a signal on INT pin of processor (due to which processor will aware that there is something to interrupt due to some device), when processor will see it will send Ready signal to PIC then PIC sends info regarding interrupt device
- * interrupt types to processor via data bus line & finally interrupt is executed.

Branch instruction, call subroutine, program interrupt

- * can cause a computer to begin executing a different instruction sequence & thus deviate from its default behaviour of executing instruction (a function, subroutine, method)
- * can cause a computer to begin executing a different instructions that perhaps a specific task, packaged as a unit.

Q1 Numerical

$$50 \text{ Kbps} \rightarrow 1 \times 10^6 \text{ usec}$$
$$1B \rightarrow 50 \times 10^6 \text{ usec}$$
$$80 \text{ usec.}$$

$$\frac{T_p}{T_i} = \frac{80 \text{ usec}}{50 \text{ usec}} \rightarrow 0.4$$

$$\frac{5 \mu\text{s}}{1 \mu\text{s}} = 5$$