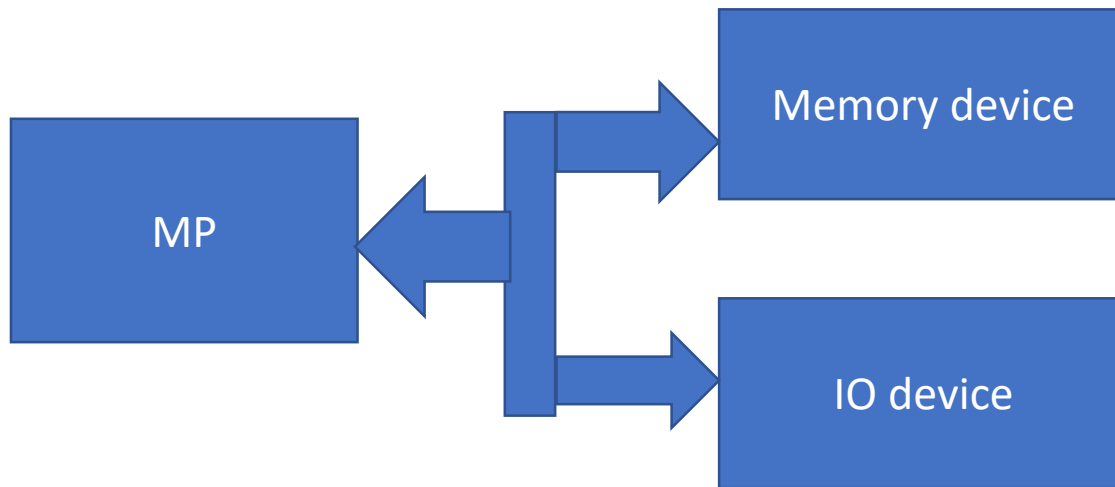


DATA TRANSFER SCHEMES & 8085 INTERRUPTS

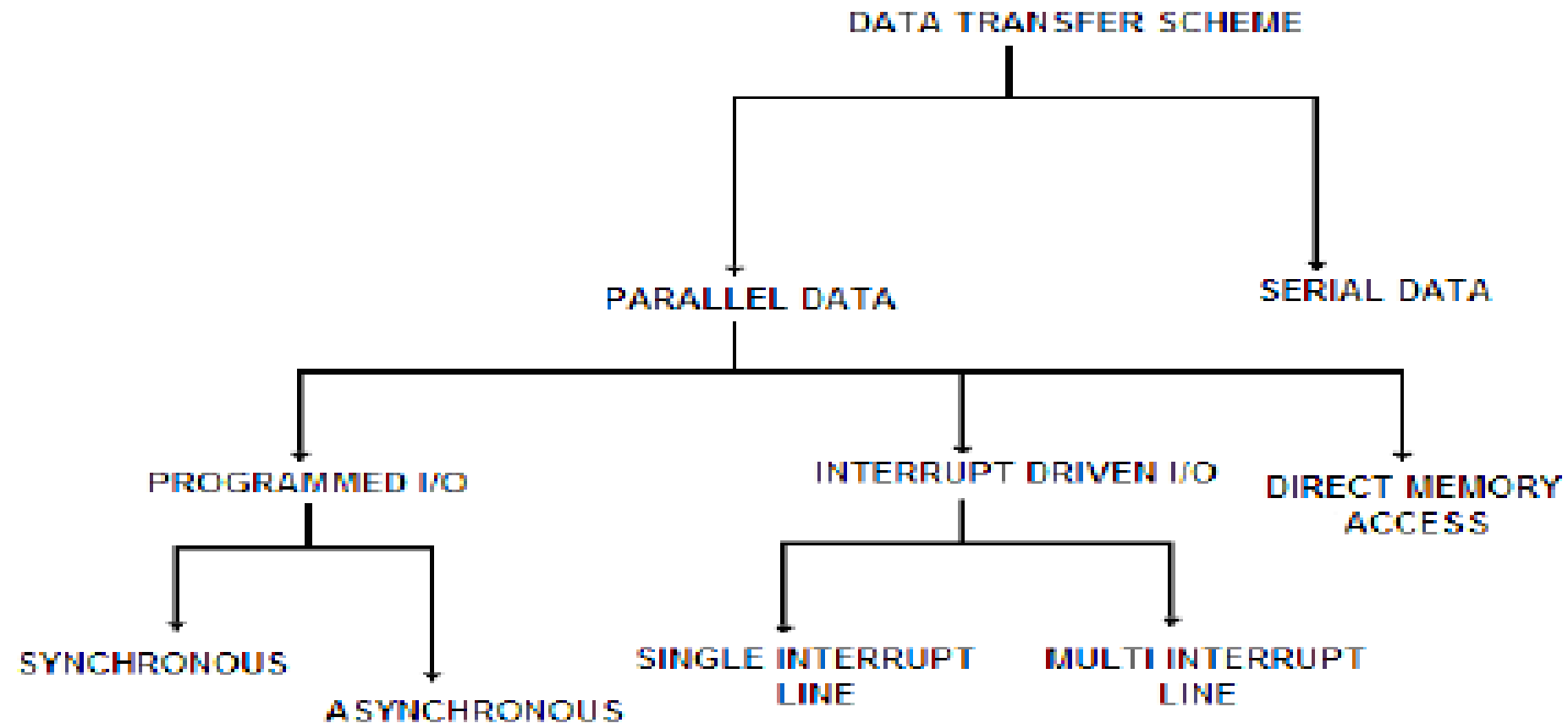


- There will be several IO and memory devices are connected to transfer data b/w memory and mp
- No much problem for transfer data communication b/w MP and M/y, since same technology is used in the m/y and MP. SPEED of both is compatible
- Problem : Data transfer b/w the mp and IO devices, the SPEED of the IO devices and the speed of MP or M/y is mismatch
- **TO OVERCOME the speed problem we need Mode of Transfer.**

NEEDS OF DATA TRANSFER SCHEME

- **A wide variety of IO devices having wide range of speed and other different characteristics are available .A slow responding IO device cannot transfer data when microprocessor issues instruction for it as it takes some time to get ready.**
- **Data codes and formats in peripheral differ from the word format of in the c.p.u and memory**
- **Transfers rates of peripherals is usually slower than the transfer rates of c.p.u.**
- **Operating modes of peripheral are different from each other and each must be controlled so as not to disturb the operation of each other peripherals connected to c.p.u**

Modes of data transfer schemes



The MP transmits/Receives data in 2 modes :

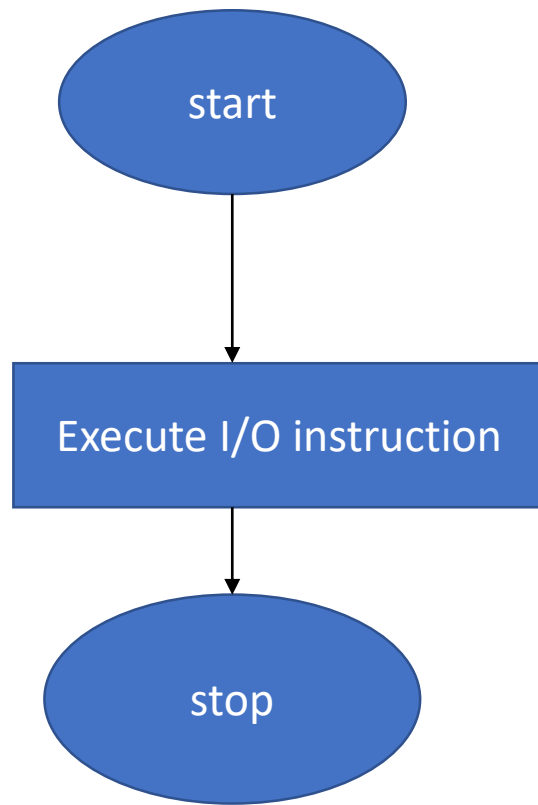
- * parallel IO mode
 - * Serial IO mode
-
- 8085 MP is a Parallel Device, since it transfers 8 bits of data at a time over 8 data lines.
 - In many situation parallel IO mode is either Impossible Eg: CRT terminal, That parallel data communication is not possible.
 - These device need serial IO mode is used which transfer a single bi on a single line at a time.
 - That's why there is parallel to serial conversion or serial to parallel conversion.

Programmed I/O Data transfer:-

- **—Synchronous Data Transfer: -**

- **Synchronous means “at the same time”. In this format of data transfer transmitter and receiver device are synchronized with the same clock pulse.**

- **This type of data transfer format is used in between the devices that match in speed. This method is invariably used in between memory and microprocessor as they are compatible.**



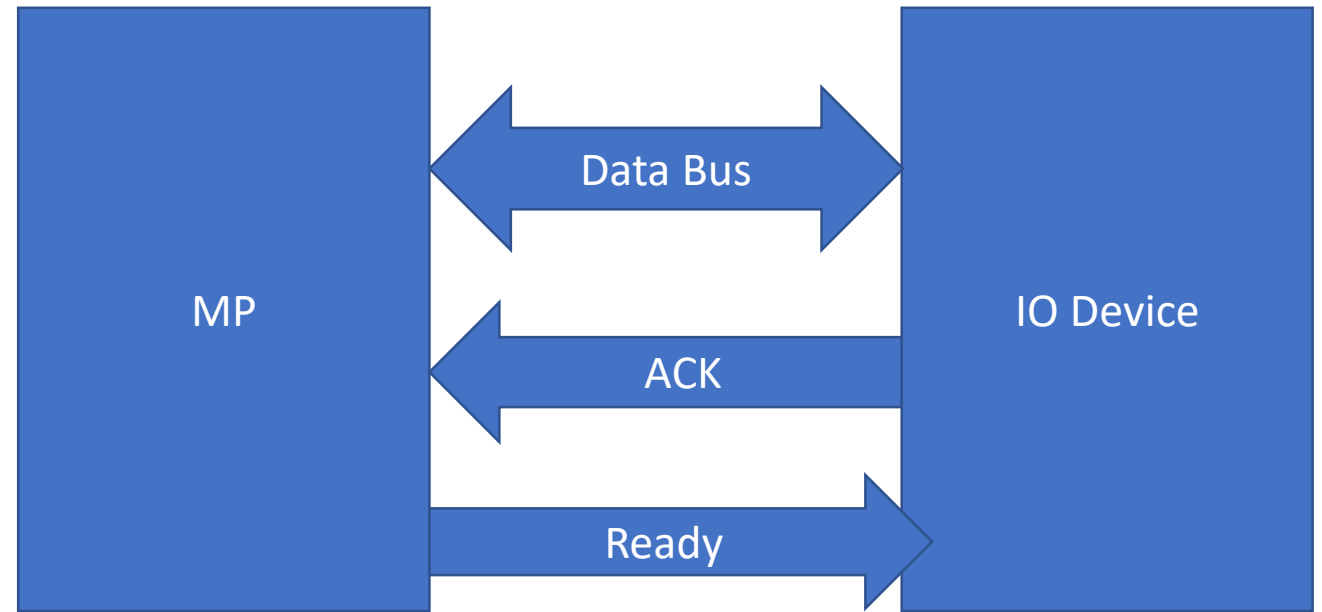
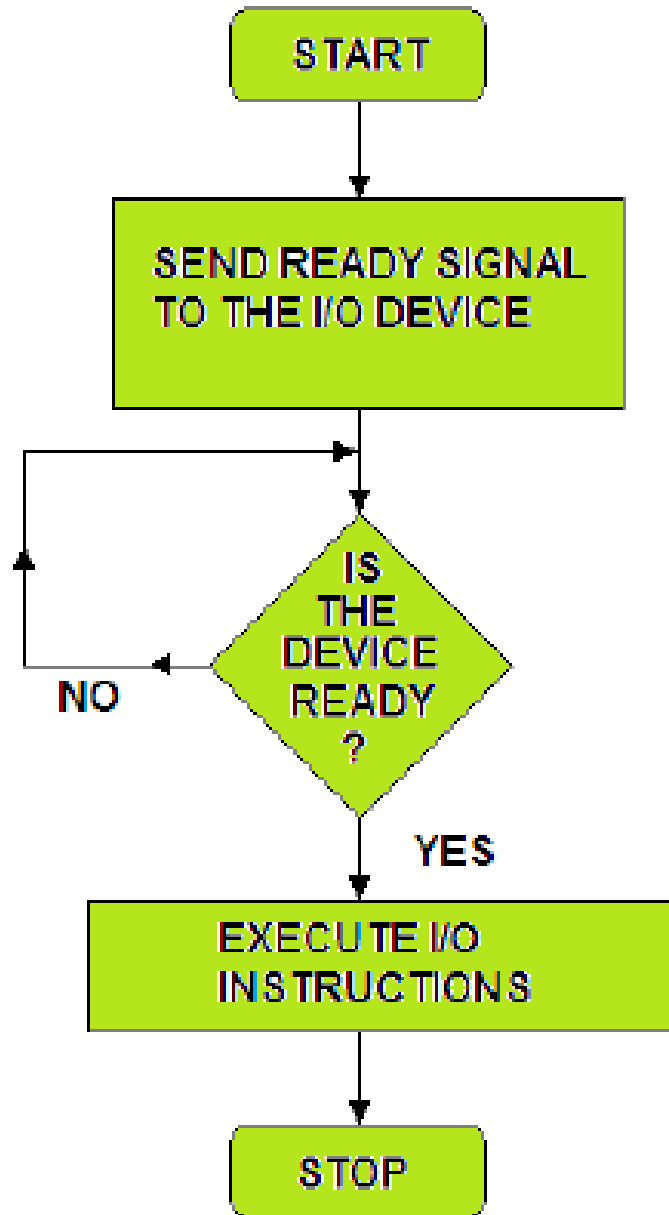
disadvantage:

Its not suitable when speed of character is not known.

- **Asynchronous Data Transfer: -**
- **Asynchronous means “at a regular interval”. In this method data transfer is not based on predetermined timing pattern in this technique the status of the IO device is checked by the microprocessor before the data is transferred. This method is invariably used in between microprocessor and IO devices.**

Its used when the speed of the IO devices is slower than the speed of the MP(MISMATCH) Its asynchronous to each other.

Its also called as “HANDSHAKING” mode.



Handshaking Mode

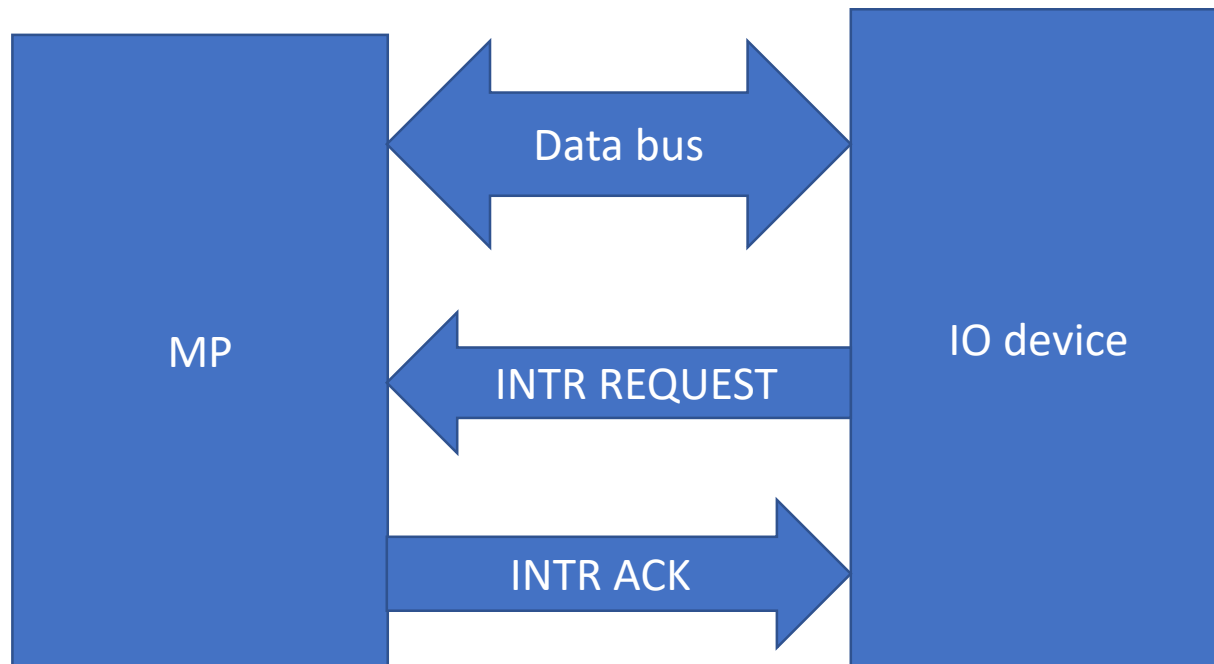
Advantage: can be used even when the speed character is not known

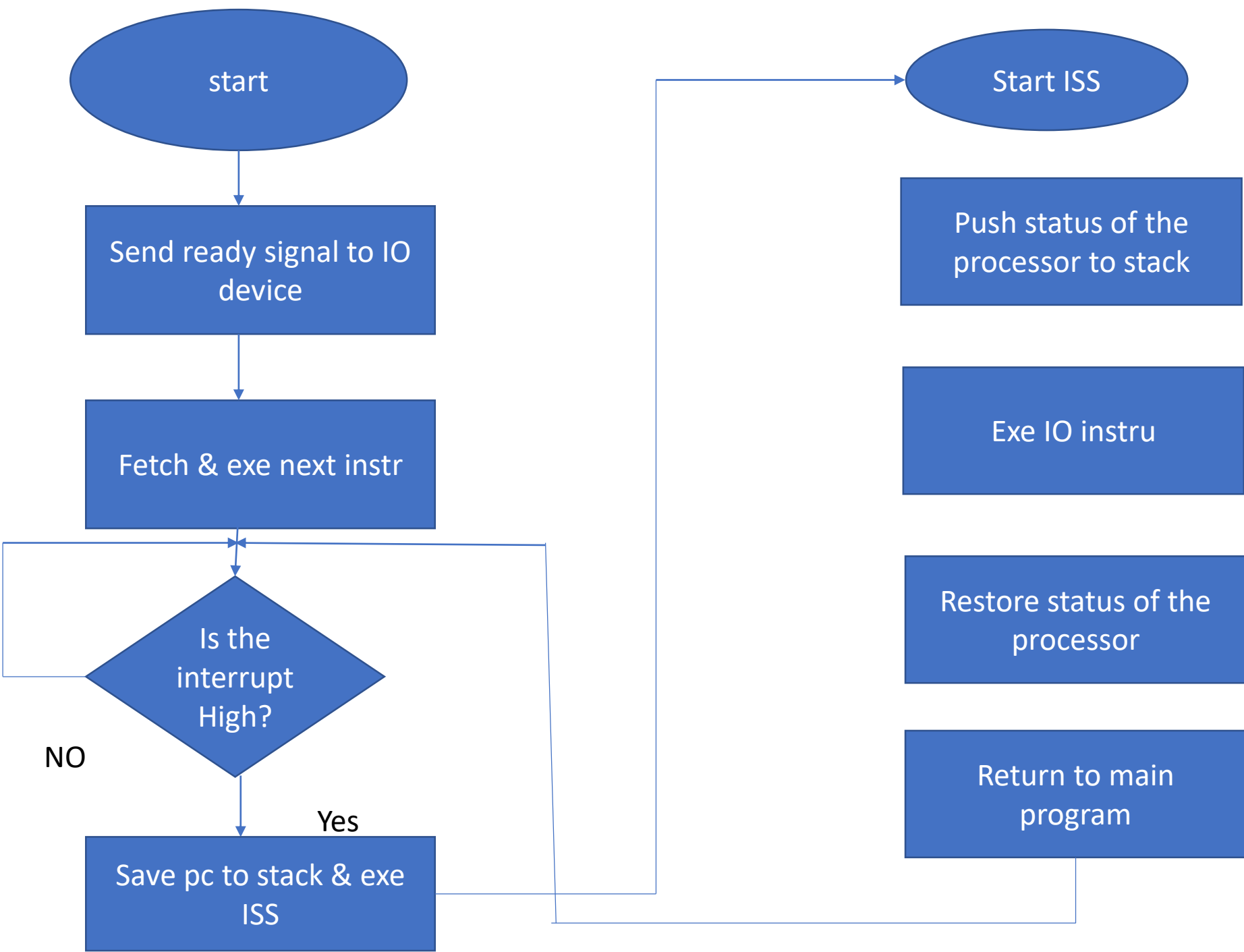
Disadvantages: It keeps checking the looping and the cpu time is wasted by just waiting and checking

To overcome this we have interrupt driven data transfer.

Interrupt Driven IO data transfer

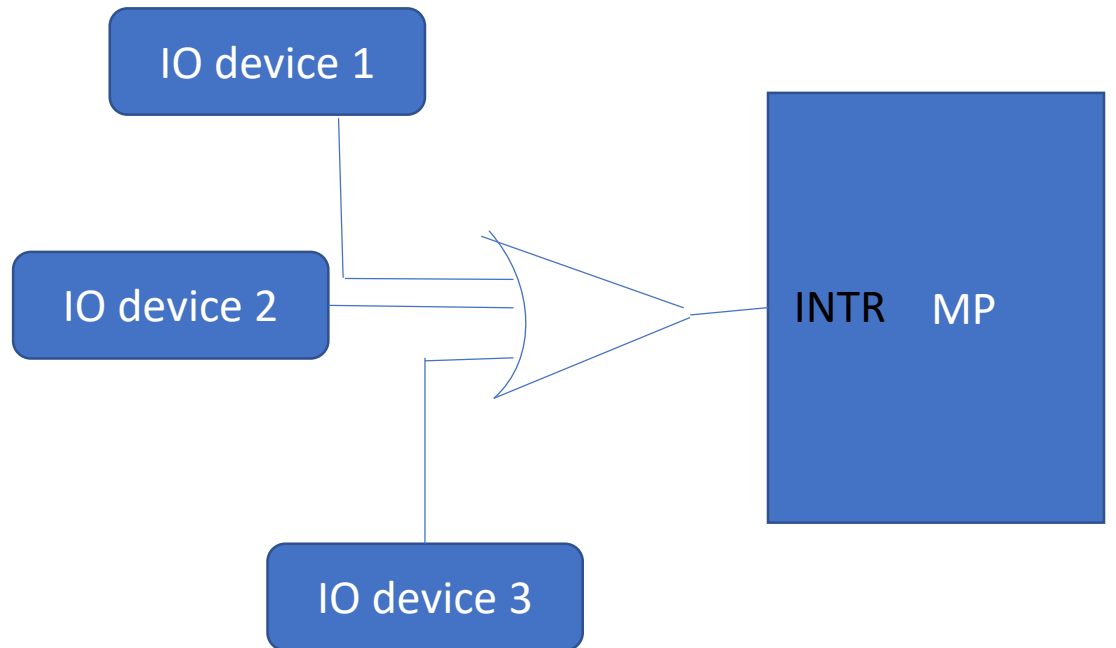
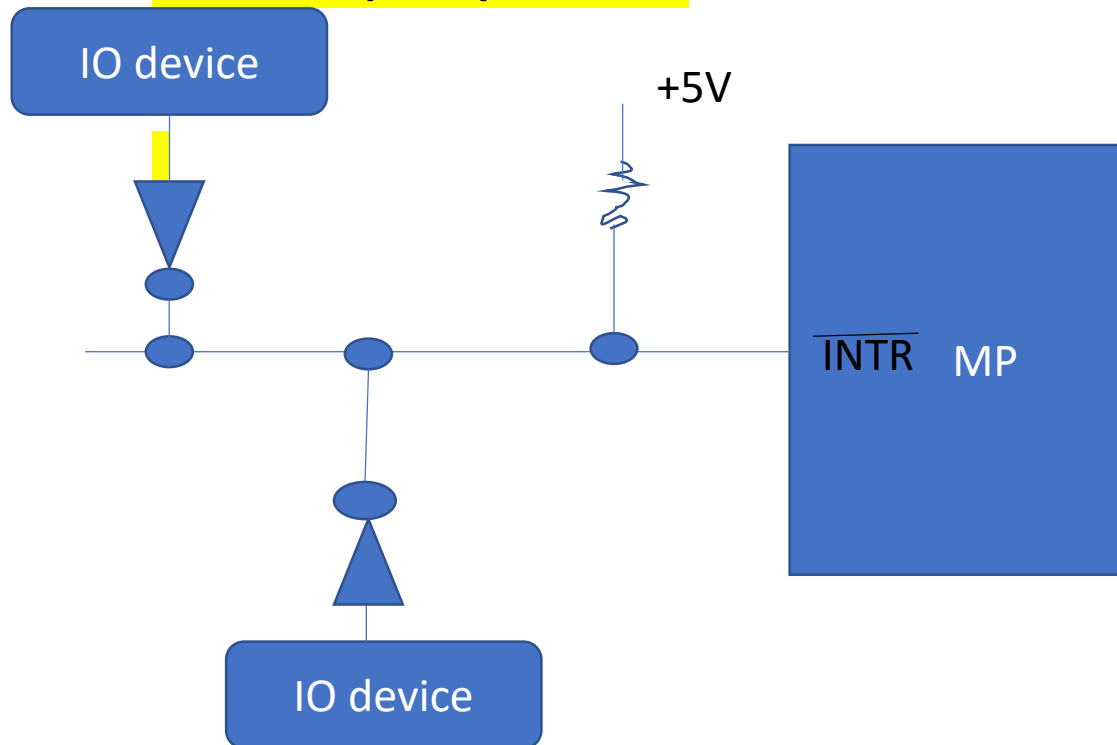
- Data transfer scheme the IO device informs the MP for the data transfer whenever the IO device is ready. This is achieved by Interrupting the MP.





Single Interrupt system

- When only one interrupt line is available with the MP and several IO devices are to be connected, then the method is known as **single Interrupt system**.



Problem :- when the interrupt line is active in either of 2 method, the MP do not identifies which devices has sent the interrupt signal.

- To solve this problem we have 3 techniques to identify the device

- 1) Polling
- 2) Daisy Chain
- 3) Multi Interrupt system

Interrupts

- An interrupt is considered to be an **emergency** signal that may be serviced.
 - The Microprocessor may respond to it **as soon as possible**.
- What happens when MP is interrupted ?
 - When the Microprocessor receives an interrupt signal, it **suspends the currently executing program** and **jumps to an Interrupt Service Routine (ISR)** to respond to the incoming interrupt.
 - Each interrupt will most probably have its own ISR.

8085 INTERRUPTS

Definition: Interrupt is a signals send by a external device to the processor, to request the processor in order to transfer data b/w the peripherals and the MP.

Its needed to transfer data B/w the peripheral and the MP.

Types of Interrupts are:

- 1) S/w interrupt
- 2) H/w interrupt
- 3) Multi Interrupt System

Software Interrupts

- 8085 MP has 8 s/w interrupts. RST 0, RST 1, RST 2, RST 3, RST 4, RST 5, RST 6, RST 7. its known as restart (RST) Instruction.
- Syntax: RST n (where n varies from 0 to 7)
- One byte call instruction that transfers the program execution to a specified m/y location on memory.
- RST instruction is similar to the call instruction.
- Address in the PC is stored on the stack before execution is transferred to the RST call location
- When the MP encounters a return Instruction in subroutine with RST instruction ie RET, the program returns to the address stored on stack.

RST instructions

- 8 RST instructions

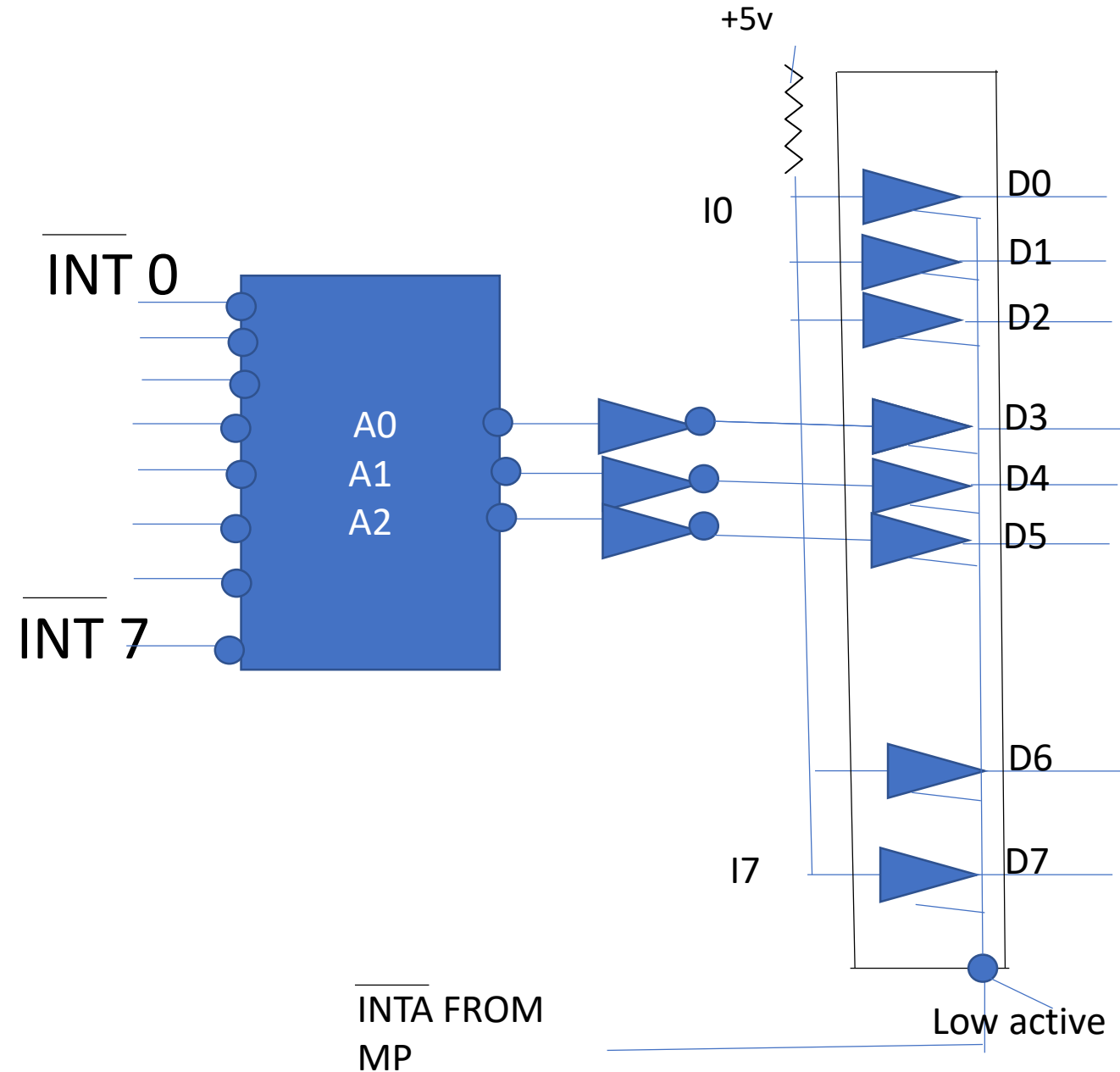
Mnemonics Or Instruction	Binary code								Hex	Call Location Effect
	D7	D6	D5	D4	D3	D2	D1	D0		
RST0	1	1	0	0	0	1	1	1	C7	CALL 0000 H
RST1	1	1	0	0	1	1	1	1	CF	CALL 0008H
RST2	1	1	0	1	0	1	1	1	D7	CALL 0010H
RST3	1	1	0	1	1	1	1	1	DF	CALL 0018H
RST4	1	1	1	0	0	1	1	1	E7	CALL 0020H
RST5	1	1	1	0	1	1	1	1	EF	CALL 0028H
RST6	1	1	1	1	0	1	1	1	F7	CALL 0030H
RST7	1	1	1	1	1	1	1	1	FF	CALL 0038H

Multi Interrupt system

- The 3 to 8 priority encoder generates a 3 bit binary code corresponding to active input.
- If 2/more I/p from IO device is high at the same time, the highest numbered I/p will be encoded.
- The three bits are inverted by the inverted buffer and then applied to an octal buffer
- D0 to D2 & D6 o D7 bits of the hex codes are same for all RST instruction
- D3 to D5 are different and are in the sequence of 3 bit, binary code number for RST 0 to RST 7.

[illegible]

- Assume INT low 010, then A0 A1 A2 will be 101 will be inverted by the Inverter buffer I3 I4 I5.
- The other lines will be connected to +5v becomes high.
- D7H places on data bus – low active INTA is signalled



Hardware Interrupts

- 8085 has 5 hardware interrupts inputs are

RST 5.5

RST 6.5

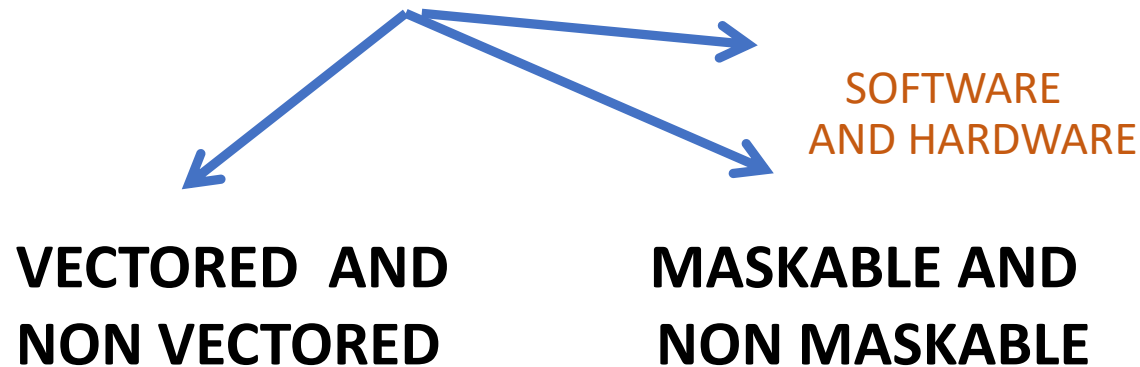
RST 7.5

TRAP and

INTR

THIS 5 PINS ARE PROVIDED IN MP TO TRANSFER
DATA BETWEEN THE IO DEVICE AND PROCESSOR

TYPES OF INTERRUPT:



- Classification of Interrupts

Interrupts can be classified into two types:

- Vectored (the address of the service routine is hard-wired) RST 7.5,6.5,5.5,TRAP
- Non-vectored (the address of the service routine needs to be supplied externally by the device) INTR

Interrupts can also be classified into:

- Maskable Interrupts (Can be delayed or Rejected) RST 7.5,6.5,5.5,INTR
- Non-Maskable Interrupts (Can not be delayed or Rejected) TRAP

Responding to Interrupts

- ▶ Responding to an interrupt may be **immediate** or **delayed** depending on whether the interrupt is maskable or non-maskable and whether interrupts are being masked or not.
- ▶ There are two ways of redirecting the execution to the ISR depending on whether the interrupt is vectored or non-vectored.
 - Vectored: The address of the subroutine is **already known** to the Microprocessor
 - Non Vectored: The **device will have to supply** the address of the subroutine to the Microprocessor

Interrupt Vectors and the Vector Table

- An **interrupt vector** is a pointer to where the ISR is stored in memory.
- All interrupts (vectored or otherwise) are mapped onto a memory area called the **Interrupt Vector Table** (IVT).
- The IVT is usually located in **memory page 00** (0000H - 00FFH).
- The purpose of the IVT is to hold the vectors that redirect the microprocessor to the right place when an interrupt arrives.

The 8085 Non-Vectored Interrupt Process

1. The interrupt process should be **enabled** using the **EI** instruction.
2. The 8085 checks for an interrupt during the execution of **every** instruction.
3. If INTR is high, MP completes current instruction, disables **DI** the interrupt and sends INTA (Interrupt acknowledge) signal to the device that interrupted
4. INTA allows the I/O device to send a RST instruction through data bus.
5. Upon receiving the INTA signal, MP saves the memory location of the next instruction on the stack and the program is transferred to 'call' location (ISR Call) specified by the RST instruction

The 8085 Non-Vectored Interrupt Process

6. Microprocessor Performs the ISR.
7. ISR must include the 'EI' instruction to enable the further interrupt within the program.
8. RET instruction at the end of the ISR allows the MP to retrieve the return address from the stack and the program is transferred back to where the program was interrupted.

The 8085 Maskable Interrupts

- 1) 8085 MP has 4 maskable interrupts RST 7.5, 6.5, 5.5 and INTR
- 2) Its controlled by a single FF which is inside the MP
- 3) Interrupt enable FF is controlled using 2 instructions

EI ---- Enable Interrupt

DI ---- Disable Interrupt

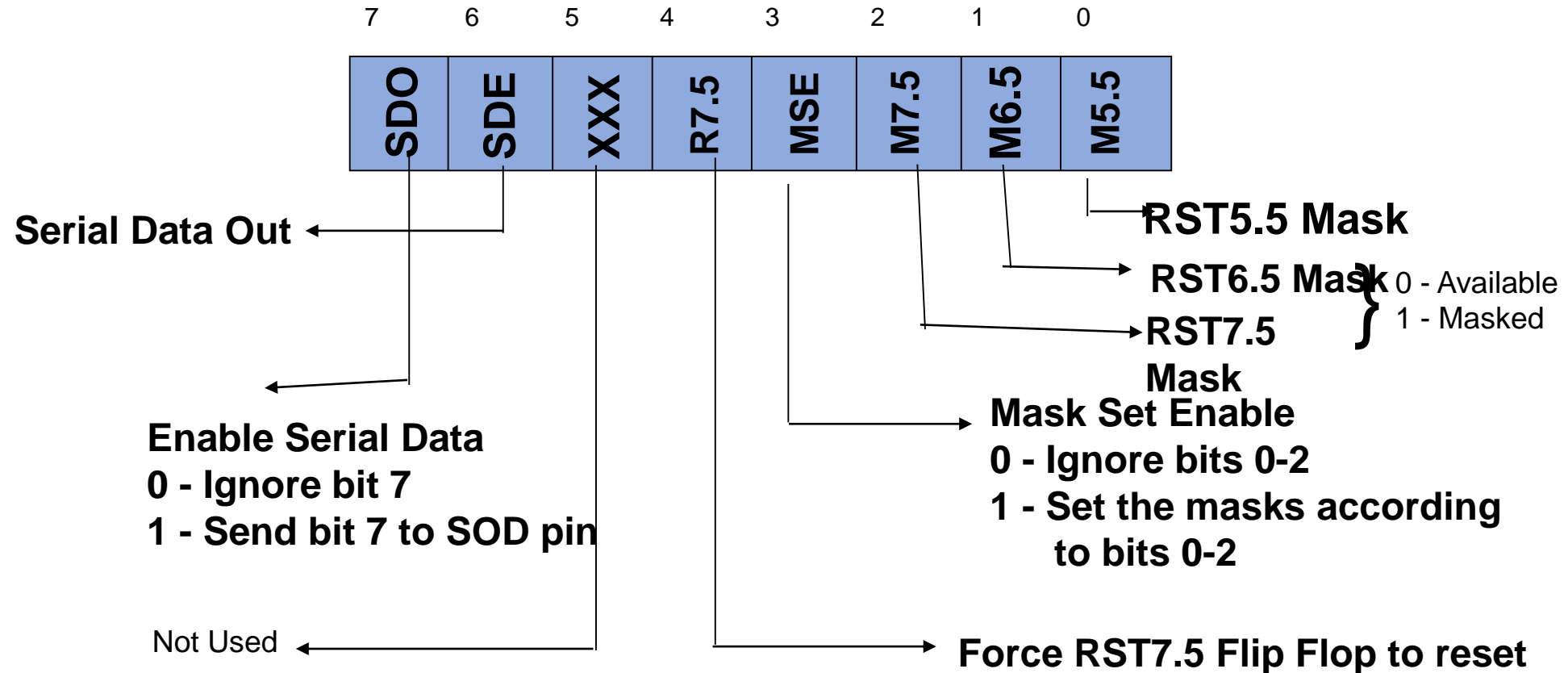
The 8085 NON Maskable Interrupts

- 1) 8085 has a single non-maskable interrupt called TRAP
- 2) Its not affected by the value of the interrupt enable FF
- 3) Its independent of EI and DI instructions
- 4) Trap is the highest and has a lower priority than the HOLD signal used for DMA

Manipulating the Masks / Interrupt Instruction

- The Interrupt Enable flip flop is manipulated using the EI/DI instructions.
- The individual **masks** for RST 5.5, RST 6.5 and RST 7.5 are manipulated using the **SIM** instruction.
 - This instruction takes the bit pattern in the Accumulator and applies it to the interrupt mask enabling and disabling the specific interrupts.

How SIM Interprets the Accumulator



SIM and the Interrupt Mask

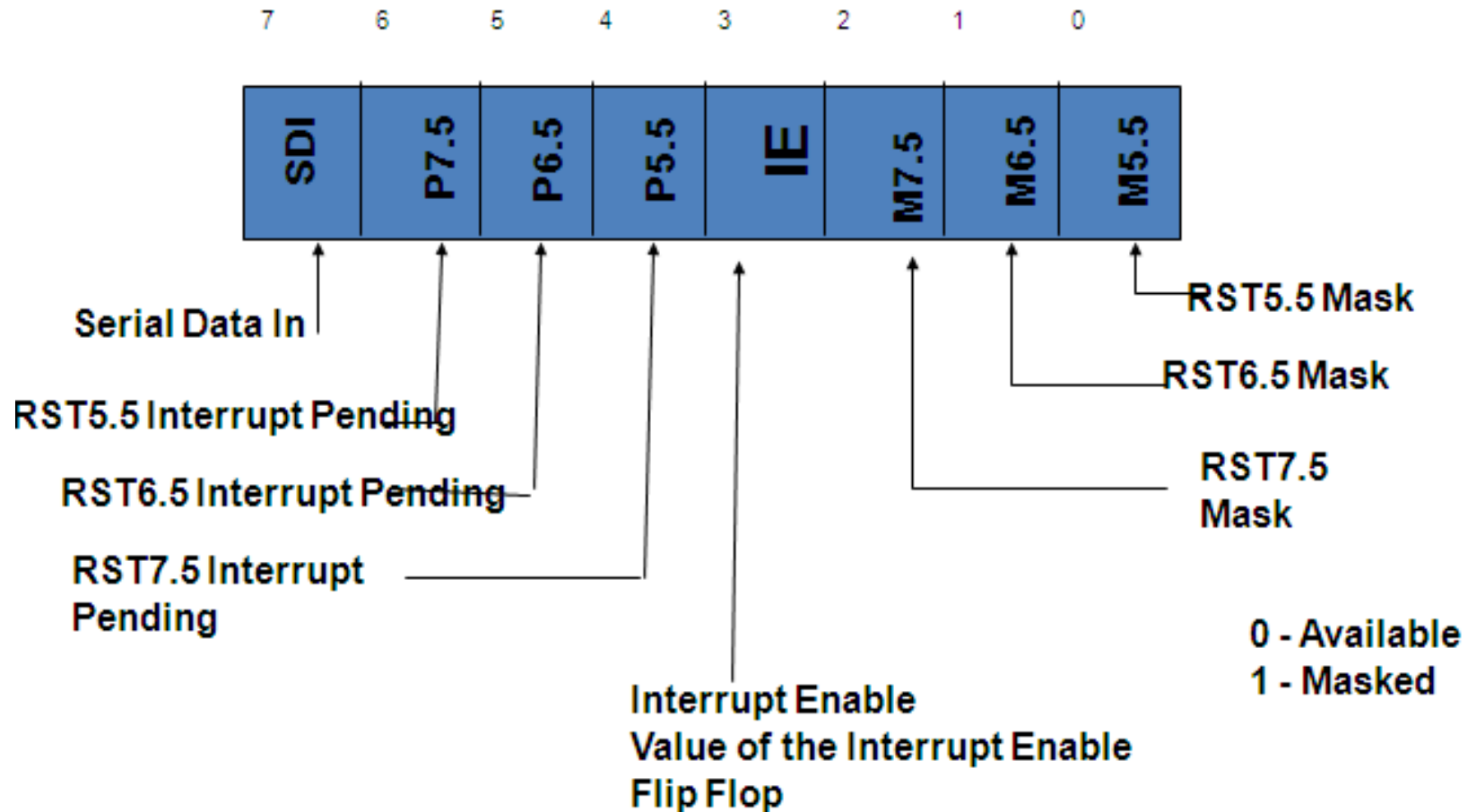
- Bit 0 is the **mask** for RST 5.5
- Bit 1 is the **mask** for RST 6.5
- Bit 2 is the **mask** for RST 7.5.
 - If the mask bit is 0, the interrupt is **available**.
 - If the mask bit is 1, the interrupt is **masked**.

- Bit 3 (Mask Set Enable - MSE) is an **enable for setting the mask**.
 - If it is set to 0 the mask is **ignored** and the old settings remain.
 - If it is set to 1, the new settings are **applied**.
 - The SIM instruction is used for multiple purposes and not only for setting interrupt masks.
 - It is also used to control functionality such as Serial Data Transmission.
 - Therefore, bit 3 is necessary to tell the microprocessor whether or not the interrupt masks should be modified

SIM and the Interrupt Mask

- The RST 7.5 interrupt is the **only** 8085 interrupt that has **memory**.
 - If a signal on RST7.5 arrives while it is masked, a flip flop will remember the signal.
 - When RST7.5 is unmasked, the microprocessor will be interrupted **even if the device has removed the interrupt signal**.
 - This flip flop will be **automatically reset** when the microprocessor **responds to an RST 7.5 interrupt**.
- Bit 4 of the accumulator in the SIM instruction allows **explicitly resetting** the RST 7.5 memory even if the microprocessor did not respond to it.
- Bit 5 is not used by the SIM instruction

RIM sets the Accumulator's different bits



The RIM Instruction and the Masks

- Bits 0-2 show the current **setting of the mask** for each of RST 7.5, RST 6.5 and RST 5.5
 - They return the contents of the three mask flip flops.
 - They can be used by a program to read the mask settings in order to modify only the right mask.
- Bit 3 shows whether the maskable interrupt process is **enabled or not**.
 - It returns the contents of the Interrupt Enable Flip Flop.
 - It can be used by a program to determine whether or not interrupts are enabled.

The RIM Instruction and the Masks

- Bits 4-6 show whether or not there are **pending interrupts** on RST 7.5, RST 6.5, and RST 5.5
 - Bits 4 and 5 return the current value of the RST5.5 and RST6.5 **pins**.
 - Bit 6 returns the current value of the RST7.5 memory **flip flop**.
- Bit 7 is used for **Serial Data Input**.
 - The RIM instruction reads the value of the **SID pin** on the microprocessor and returns it in this bit.

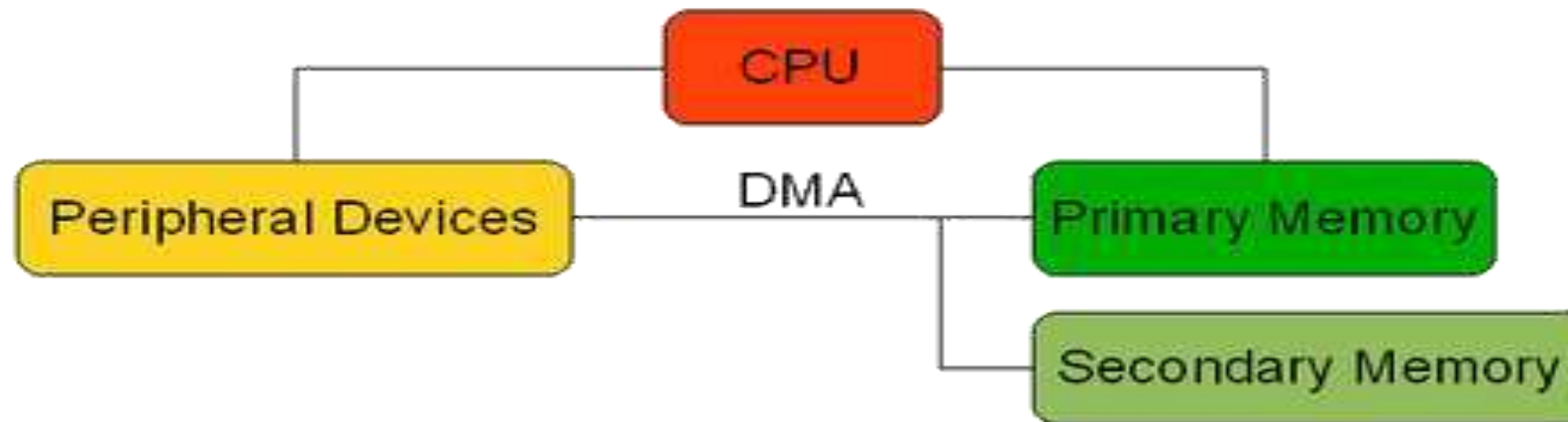
DIRECT MEMORY ACCESS (DMA) DATA TRANSFER

Direct memory access (DMA) Data Transfer

- The transfer between the I/O devices and external memory via the accumulator.
- When bulk of data is been transferred from IO devices to memory or vice-versa.. It takes more time because the data is 1st transferred to accumulator and then to other device.
- To overcome this problem the Direct memory Access (DMA) data transfer scheme is used.

DMA

- **Direct Memory Access (DMA)** is a method that allows an input/output (I/O) device to send or receive data directly to or from the main memory, bypassing the CPU to speed up memory operations. The process is managed by a chip known as a DMA controller (DMAC).

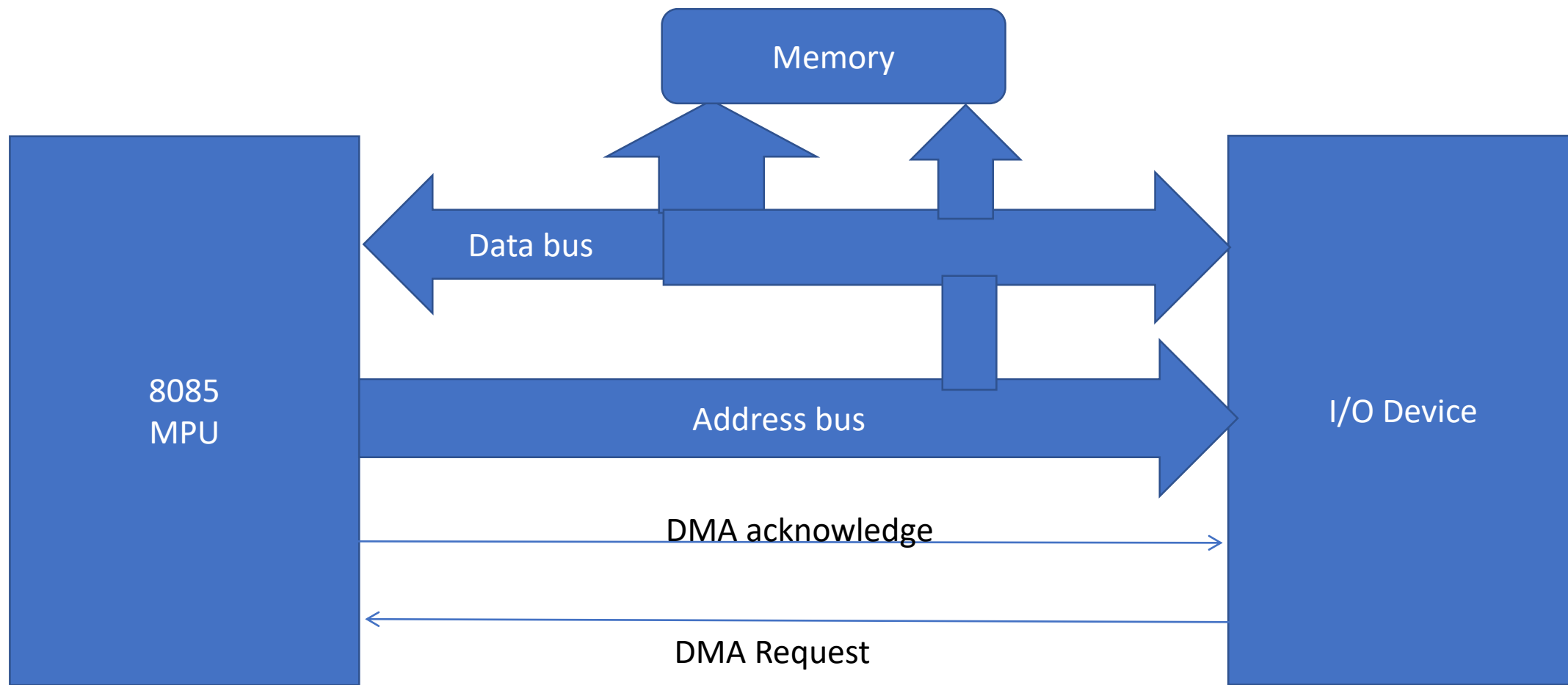


DMA

- DMA transfers are performed by a control circuit that is part of the I/O device interface.
- It refer to this circuit as a DMA controller.
- The DMA controller performs the function that would normally be carried out by the processor when accessing the main memory.
- Device wishing to perform DMA asserts the processors bus request signal
- Processor completes the current bus cycle and then asserts the bus grant signal to the device.

Basic DMA operation

- The direct memory access (DMA) I/O technique provides direct access to the memory while the microprocessor is temporarily disabled.
- A DMA controller temporarily borrows the address bus, data bus, and control bus from the microprocessor and transfers the data bytes directly between an I/O port and a series of memory locations.
- The DMA transfer is also used to do high-speed memory-to-memory transfers.
- Two control signals are used to request and acknowledge a DMA transfer in the microprocessor-based system.



- The external DMA controller sends a signal on HOLD pin to the mp
- The mp completes the current operation and sends signal on HLDA to make use of the address and data bus.
- For transfers the data between the two devices
- The DMA controller turns off the HOLD signal when the data transferred between the IO device and memory and then the mp takes back control of the buses.

