1.5 I/O Ports and Data Transfer Concepts

I/O Ports

There are two methods in which I/O devices can be connected to the Microprocessor.

- Memory mapped I/O
- I/O mapped I/O

Memory mapped I/O:

In this method I/O device is treated like the memory. Here there is no IO/M signal. If the processor wants to read the data from a I/O device it will place the address of the I/O device on the address bus. Then the I/O device will get selected. The memory which is having the same address will also get selected.so we have to use separate address for memory and separate address for I/O device.

I/O mapped I/O:

Here we have the IO/M signal. So we can select either the memory or I/O device for read and write operation.

Data Transfer Concepts

- Parallel data transfer
- Serial data transfer

Parallel data transfer

- Programmed I/O
- Interrupt I/O
- DMA

Programmed I/O:

Here the processor has to check whether the I/O device is ready or not through the Ready signal of the I/O device. If the ready signal is high, then it will send the data to the I/O device. Otherwise it will continuously check the Ready signal. The processor is busy in checking the Ready signal. The drawback is wastage of time.

Interrupt I/O:

In this method the I/O device will interrupt the Processor through the INTR signal to indicate to the processor that it is ready to accept the next data. Then the processor will

send the INTA signal. Then the processor stops its normal execution and start transferring the data to the I/O device.

DMA:

Using DMA I/O device can directly transfer the data to the Memory using the Address and Data buses of Processor.

Serial data Transfer

Some of the external I/0 devices receive only the serial data. Normally serial communication is used in the MultiProcessor environment. 8051 has two pins for serial communication.

- SID- Serial Input data.
- SOD-Serial Output data.

Serial data communication can be categorized on the basis of how data transmission occurs. These are:

1. **Simplex:**

   In simplex communication, the hardware is set such that the data exchange takes place in only one direction. Example: Computer to Printer communication.

2. **Half Duplex:**

   The half-duplex communication allows the data exchange in both directions, but not at the same time. Example: Walkie talkie.

3. **Full Duplex:**

   It permits the information transfer in both directions at the same time. Example: Telephone lines.

Interrupts

The processor has the following interrupts:

INTR is a maskable hardware interrupt. The interrupt can be enabled/disabled using STI/CLI instructions or using more complicated method of updating the FLAGS register with the help of the POPF instruction.

When an interrupt occurs, the processor stores FLAGS register into stack, disables further interrupts, fetches from the bus one byte representing interrupt type, and jumps to

interrupt processing routine address of which is stored in location 4 * <interrupt type>. Interrupt processing routine should return with the IRET instruction.

NMI is a non-maskable interrupt. Interrupt is processed in the same way as the INTR interrupt. Interrupt type of the NMI is 2, i.e. the address of the NMI processing routine is stored in location 0008h. This interrupt has higher priority than the maskable interrupt.

Software interrupts can be caused by:

- INT instruction - breakpoint interrupt. This is a type 3 interrupt.
- INT <interrupt number> instruction - any one interrupt from available 256 interrupts.
- INTO instruction - interrupt on overflow
- Single-step interrupt - generated if the TF flag is set. This is a type 1 interrupt. When the CPU processes this interrupt it clears TF flag before calling the interrupt processing routine.

Synchronous Data Transfer

- The word 'Synchronous 'means 'taking place at the same time.'
- Thus, to establish communication between our processor and the device, we need to set a common clock pulse. This common pulse synchronizes the peripheral device with the 8085 microprocessor.
- This method is used when the speed of the microprocessor, Intel 8085, in this case, and the external peripheral device match with each other.
- If the device is ready to send data, it can indicate via the READY pin of 8085. Once the speeds match, the data transfer immediately begins, once a signal is issued by the microprocessor to begin transferring. The microprocessor need not wait for an extended period because of the matching speeds.
- This technique of data transfer is seldom used to communicate with I/O devices though. Because I/O devices compatible with the microprocessor's speed are usually not found.
- Hence, this method of data transfer is most commonly employed for communicating with compatible memory devices.

Asynchronous Data Transfer

- When the speed of the I/O device is slower than that of the microprocessor, we prefer the Asynchronous Data Transfer Method. As the speeds of both the devices differ, the I/O device's internal timing is entirely independent of the microprocessor.
- Thus, they are termed to be 'asynchronous 'from each other. The term asynchronous means 'at irregular intervals.'