

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL New Scheme
Based On AICTE Flexible Curricula Computer Science & Engineering, IV-Semester
CS-404:Computer System Organization

Topic Covered

**Data Transmission, Programmed I/O, Interrupt Driven I/O, DMA
Data Transfer, Asynchronous and Synchronous Data Transmission**

Programmed I/O

Programmed I/O (PIO) refers to data transfers initiated by a CPU under driver software control to access registers or memory on a device.

The CPU issues a command then waits for I/O operations to be complete. As the CPU is faster than the I/O module, the problem with programmed I/O is that the CPU has to wait a long time for the I/O module of concern to be ready for either reception or transmission of data. The CPU, while waiting, must repeatedly check the status of the I/O module, and this process is known as Polling. As a result, the level of the performance of the entire system is severely degraded.

Programmed I/O basically works in these ways:

- CPU requests I/O operation
- I/O module performs operation
- I/O module sets status bits
- CPU checks status bits periodically
- I/O module does not inform CPU directly
- I/O module does not interrupt CPU
- CPU may wait or come back later

Interrupt

The CPU issues commands to the I/O module then proceeds with its normal work until interrupted by I/O device on completion of its work.

For input, the device interrupts the CPU when new data has arrived and is ready to be retrieved by the system processor. The actual actions to perform depend on whether the device uses I/O ports, memory mapping.

For output, the device delivers an interrupt either when it is ready to accept new data or to acknowledge a successful data transfer. Memory-mapped and DMA-capable devices usually generate interrupts to tell the system they are done with the buffer.

Although Interrupt relieves the CPU of having to wait for the devices, but it is still inefficient in data transfer of large amount because the CPU has to transfer the data word by word between I/O module and memory.

Below are the basic operations of Interrupt:

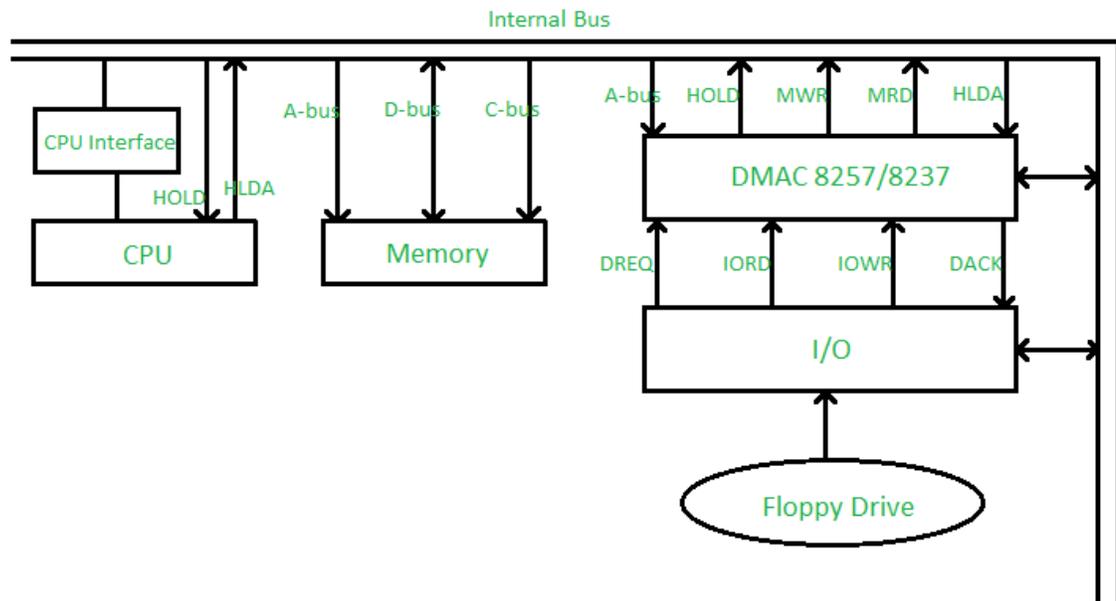
- CPU issues read command
- I/O module gets data from peripheral whilst CPU does other work
- I/O module interrupts CPU
- CPU requests data
- I/O module transfers data

Direct Memory Access (DMA)

Suppose any device which is connected at input-output port wants to transfer data to transfer data to memory, first of all it will send input-output port address and control signal, input-output read to input-output port, then it will send memory address and memory write signal to memory where data has to be transferred. In normal input-output technique the processor becomes busy in checking whether any input-output operation is completed or not for next input-output operation, therefore this technique is slow.

This problem of slow data transfer between input-output port and memory or between two memories is avoided by implementing Direct Memory Access (DMA) technique. This is faster as the microprocessor/computer is bypassed and the control of address bus and data bus is given to the DMA controller.

- HOLD – hold signal
- HLDA – hold acknowledgment
- DREQ – DMA request
- DACK – DMA acknowledgment



Suppose a floppy drive which is connected at input-output port wants to transfer data to memory, the following steps are performed:

- **Step-1:** First of all the floppy drive will send DMA request (DREQ) to the DMAC, it means the floppy drive wants its DMA service.
- **Step-2:** Now the DMAC will send HOLD signal to the CPU.
- **Step-3:** After accepting the DMA service request from the DMAC, the CPU will send hold acknowledgement (HLDA) to the DMAC, it means the microprocessor has released control of the address bus the data bus to DMAC and the microprocessor/computer is bypassed during DMA service.
- **Step-4:** Now the DMAC will send one acknowledgement (DACK) to the floppy drive which is connected at the input-output port. It means the DMAC tells the floppy drive be ready for its DMA service.
- **Step-5:** Now with the help of input-output read and memory write signal the data is transferred from the floppy drive to the memory.

WHAT IS ASYNCHRONOUS TRANSMISSION

In contrast, asynchronous transmission works in spurts and must insert a start bit before each data character and a stop bit at its termination to inform the receiver where it begins and ends.

The term *asynchronous* is used to describe the process where transmitted data is encoded with start and stop bits, specifying the beginning and end of each character.

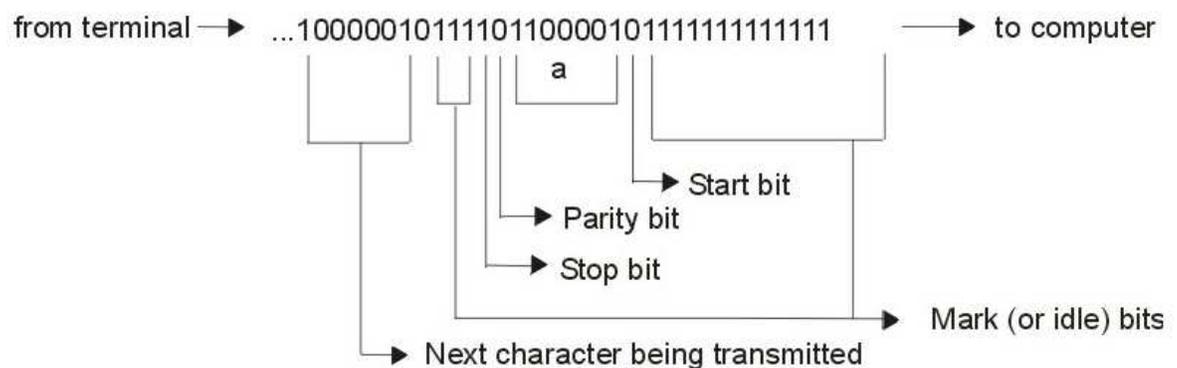
An example of synchronous transmission is shown in the following figure.

Figure 2. Asynchronous transmission



These additional bits provide the timing or synchronization for the connection by indicating when a complete character has been sent or received; thus, timing for each character begins with the start bit and ends with the stop bit.

When gaps appear between character transmissions, the asynchronous line is said to be in a mark state. A mark is a binary 1 (or negative voltage) that is sent during periods of inactivity on the line as shown in the following figure



When the mark state is interrupted by a positive voltage (a binary 0), the receiving system knows that data characters are going to follow. It is for this reason that the start bit, which precedes the data character, is always a space bit (binary 0) and that the stop bit, which signals the end of a character, is always a mark bit (binary 1).

The following is a list of characteristics specific to asynchronous communication:

- Each character is preceded by a start bit and followed by one or more stop bits.
- Gaps or spaces between characters may exist.

With asynchronous transmission, a large text document is organized into long strings of letters (or characters) that make up the words within the sentences and paragraphs. These characters are sent over the communication link one at a time and reassembled at the remote location.

In asynchronous transmission, ASCII character would actually be transmitted using 10 bits. For example, "0100 0001" would become "**1** 0100 0001 **0**". The extra one (or zero, depending on parity bit) at the start and end of the transmission tells the receiver first that a character is coming and secondly that the character has ended. This method of transmission is used when data are sent intermittently as opposed to in a solid stream. In the previous example the start and stop bits are in bold.

The start and stop bits must be of opposite polarity. This allows the receiver to recognize when the second packet of information is being sent.

Asynchronous transmission is used commonly for communications over telephone lines.

WHAT IS SYNCHRONOUS TRANSMISSION?

- The term *synchronous* is used to describe a continuous and timed bound /clock based transfer of data blocks.
- It is a data transfer method in which a continuous stream of data signals is accompanied by timing signals (generated by an electronic clock) to ensure that the transmitter and the receiver are in step (synchronized) with one another.
- The data is sent in blocks (called frames or packets) spaced by fixed time intervals.
- Synchronous transmission modes are used when large amounts of data must be transferred very quickly from one location to the other.
- Synchronous transmission synchronizes transmission speeds at both the receiving and sending end of the transmission by using clock signals.
- A continual stream of data is then sent between the two nodes.
- The data blocks are grouped and spaced in regular intervals and are preceded by special characters called syn or synchronous idle characters. See the following illustration.

Figure 1. Synchronous transmission



After the syn characters are received by the remote device, they are decoded and used to synchronize the connection. After the connection is correctly synchronized, data transmission may begin.

An analogy of synchronous transmission would be the transmission of a large text document. Before the document is transferred across the synchronous line, it is first broken into blocks of sentences or paragraphs. The blocks are then sent over the communication link to the remote site.

The timing needed for synchronous connections is obtained from the devices located on the communication link. All devices on the synchronous link must be set to the same clocking.

The following is a list of characteristics specific to synchronous communication:

- There are no gaps between characters being transmitted.
- Timing is supplied by modems or other devices at each end of the connection.
- Special syn characters precede the data being transmitted.
- The syn characters are used between blocks of data for timing purposes.

Due to there being no start and stop bits the data transfer rate is quicker although more errors will occur, as the clocks will eventually get out of sync, and the receiving device would have the wrong time that had been agreed in the protocol for sending/receiving data, so some bytes could become corrupted (by losing bits).

Ways to get around this problem include re-synchronization of the clocks and use of check digits to ensure the bytes is correctly interpreted and received. Most network protocols (such as Ethernet, SONET, and Token Ring) use synchronous transmission.

Bibliography

- Morris Mano , “Computer System Organization ”PHI
- Alan Clements: “Computer Organization and Architecture”, Cengage Learning
- <https://www.geeksforgeeks.org/direct-memory-access-with-dma-controller-8257-8237/>
- <https://www.louiewong.com/archives/137>
- <https://www.fiberoptics4sale.com/blogs/archive-posts/95042950-what-is-synchronous-transmission-and-asynchronous-transmission>
- <https://teachcomputerscience.com/synchronous-and-asynchronous/>
-