# Assignment-2
## Subject Name :COA(CS21002)
## Branch: CSE

**<u>Full Marks=10</u>**

**(Draw the diagrams wherever required)**

### 1. What is program controlled I/O? State the disadvantages associated with program-controlled I/O.

Ans:

In program-controlled I/O, the processor program controls the complete data transfer. So only when an I/O transfer instruction is executed, the transfer could take place. It is required to check that device is ready/not for the data transfer in most cases. Usually, the transfer is to & from a CPU register & peripheral.

Disadvantages:

✓ ties up CPU for long period with no useful work

✓ Unsuitable for high speed data transfer

✓ CPU spends most of its time in a tight loop waiting for the device to become ready(Busy waiting)

### 2. Explain the sequence of events that takes place when an INTR interrupt request (INTR) is received by the processor.

Ans: The sequence of events involved in handling an IRQ:

✓ Devices raise an IRQ.

✓ The processor interrupts the program currently being executed.

✓ The device is informed that its request has been recognized and the device deactivates the request signal.

✓ The requested action is performed.

An interrupt is enabled and the interrupted program is resumed.

### 3. Explain Direct Memory Access method with it's requirement. Also explain the registers used in a DMA operation.

In DMA approach the I/O module and main memory exchange data directly without the processor involvement. DMA involves an additional module on the system bus i.e. DMA controller, to control the data transfer between i/O module and main memory. The block diagram of DMA controller is shown in fig 7.11.

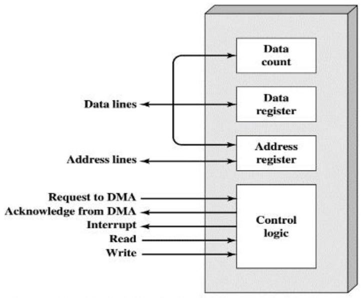


Figure 7.11 Typical DMA Block Diagram

When the processor wishes to read or write a block of data, it issues a command to the DMA controller. The processor sends the read or write request, address of the IO device, the starting memory location and number of words to be read or write. The processor then continue with other work. The MDA controller transfers the entire block of data with out going through the processor. When the transfer is complete, the DMA module sends an interrupt signal to the processor.

Ans: The internal registers of a Direct Memory Access (DMA) Controller are:-

1. Base Address Register (16 bit)
2. Base Word Count Register (16 bit)
3. Current Address Register (16 bit)
4. Current Word Count Register (16 bit)
5. Temporary Address Register (16 bit)
6. Temporary Word Count Register (16 bit)
7. Status Register (8 bit)
8. Command Register (8 bit
9. Temporary Register (8 bit)
10. Mode Register (8 bit)
11. Mask Register (4 bit)
12. Request Register (4 bit)

1. **Base Address Register:**
   It is a 16 bit register that stores the initial address from where the data transfer will take place in a DMA Controller. It is used to reload the Current Address Register after every operation.
2. **Base Word Count Register:**
   It is a 16 bit register that stores the number of transfers to be performed during an operation. It is used to reload the Current Word Count Register after every operation.
3. **Current Address Register:**
   It is a 16 bit register that stores the memory address for DMA data transfer. The value automatically increases or decreases after every operation based on how it is programmed. Each channel has its own Current Address Register.
4. **Current Word Count Register:**
   It is a 16 bit register that stores the number of transfers remaining to be performed during an operation. The value automatically decreases after every operation.
5. **Temporary Address Register:**
   It is a 16 bit register that stores the address of data during memory to memory transfer in a DMA Controller.
6. **Temporary Word Count Register:**
   It is a 16 bit register that stores the number of transfers to be performed during a memory to memory transfer in a DMA Controller.
7. **Status Register:**
   It is a 8 bit register that indicates which channel is currently under DMA services or which channels has reached its terminal count. It basically gives the status of the channels. The terminal counts(TC) bits indicates if the channel has reached its terminal count. If terminal count is reached, the transfers are terminated.
8. **Command Register:**
   It is a 8 bit register that programs the DMA operation and initializes the channel to be used for data transfer.
9. **Temporary Register:**
   It is a 8 bit register that holds data during memory to memory data transfer. It always contain the last byte transferred in previous memory to memory transfer operation

10. **Mode Register:**

It is a 8 bit register that determines the operating mode, i.e., the transfer mode and other transfer parameters, for a channel. Each channel has its own mode register which is selected by bit positions 0 and 1.

11. **Mask Register:**

It is a 4 bit register that is used to mask a channel from requesting the DMA Services. When the mask on a channel is SET, the channel is disabled. It sets or clears all the mask on all the channels with just one command.

12. **Request Register:**

It is a 4 bit register that is used to request DMA data transfer by the software. It determines which channel is requesting for the data transfer.

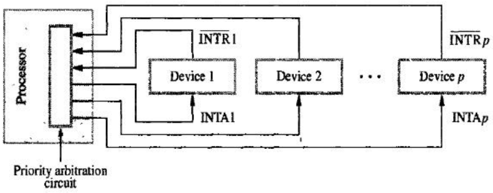
## 4. What is the need of I/O interface.

I/O interface co-ordinate the data transfer between the IO devices and system bus. Interface Circuit consists of Address Decoder,Control Circuits,Data registers, Status registers.
Address decoder enables the device to recognize its address when it is available in data line.
The data register holds the data being transfer to or from the processor. The status register contains the information relevant to the operation of I/O devices.

## 5. How does the processor resolve among simultaneous interrupt requests?

If simultaneous interrupt request arrives to the processor from two or more devices, the processor follows different mechanism to decide which device will service first.

Priority Scheme:
Organizing I/O devices in a prioritized structure. Each of the interrupt-request lines is assigned a
different priority level. The processor is interrupted only by a high priority device.


Processor
$\overline{\text{INTR1}}$
$\overline{\text{INTR}}p$
Device 1
Device 2
Device p
INTA1
INTAp
Priority arbitration circuit

Daisy-chain:
A widely used scheme is to connect the devices to form a daisy chain, as shown in figure 3a. The interrupt-request line INTR is common to all devices. The interrupt-acknowledge line, INTA, is connected in a daisy-chain fashion, such that the INTA signal propagates serially through the devices. Device that is electrically closest to the processor gets
high priority.

Daisy chain with priority:
The priority and daisy chain schemes may be combined to produce the more general structure in figure 3b. Devices are organized in groups, and each group is connected at a different priority level. Within a group, devices are connected in a daisy chain.

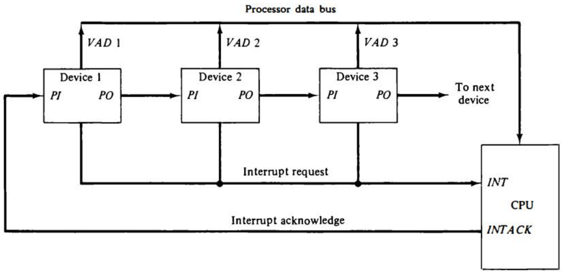


Figure 12 Daisy-chain priority interrupt.

### 6. What are the different DMA transfer techniques? Distinguish between cycle stealing and burst mode data transfer in DMA.

Cycle Stealing mode- The DMA module return the system bus to the processor after every data transfer and acquired after every instruction cycle.
Data transfer rate is slow.
CPU utilization is high.

Burst mode- The DMA module return the system bus after completion of entire data transfer.
Data transfer rate is fast.
CPU utilization is low.

(1)Burst mode DMA transfers data in a continuous sequence, while cycle stealing DMA temporarily takes control of the system bus to transfer smaller amounts of data before returning control to the CPU.
(2) Burst mode is used for faster device,while cycle stealing is used for slower device.

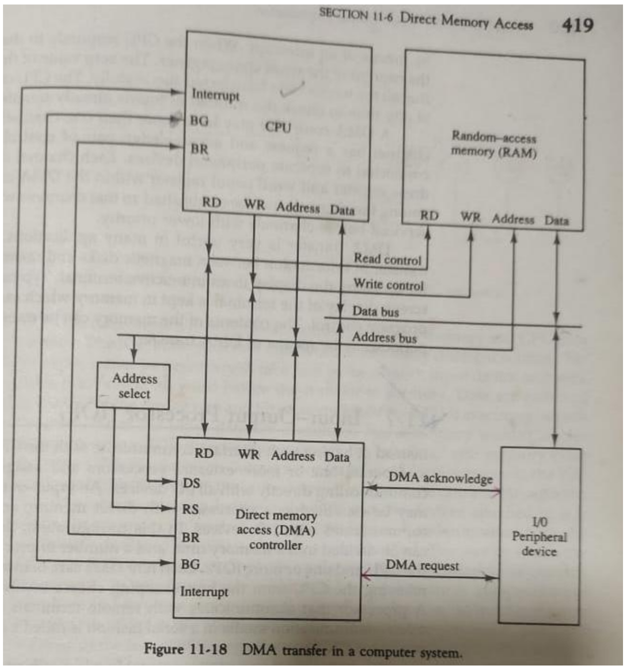
### 7. What is vectored interrupt technique?

An I/O device requesting an interrupt can identifies itself to the processor by sending a special code to the processor. The special code contains the identification of the device, starting address for the ISR, address of the branch to the ISR. The processor executes the corresponding ISR after receiving the special code. This mechanism is called vectored interrupt.

### 8. Explain the working principle of I/O transfer method that facilitates transfer of bulk data from hard disk to main memory with the highest throughput?

DMA Transfer Technique

DMA controller can transfer a block of data from an external device to the processor, without any intervention from the processor. However, the operation of the DMA controller must be under the control of a program executed by the processor. That is, the processor must initiate the DMA transfer. To initiate the DMA transfer, the processor informs the DMA controller of: Starting address, Number of words in the block, Direction of transfer (I/O device to the memory, or memory to the I/O device). Once the DMA controller completes the DMA transfer, it informs the processor by raising an interrupt signal.



SECTION 11-6 Direct Memory Access 419

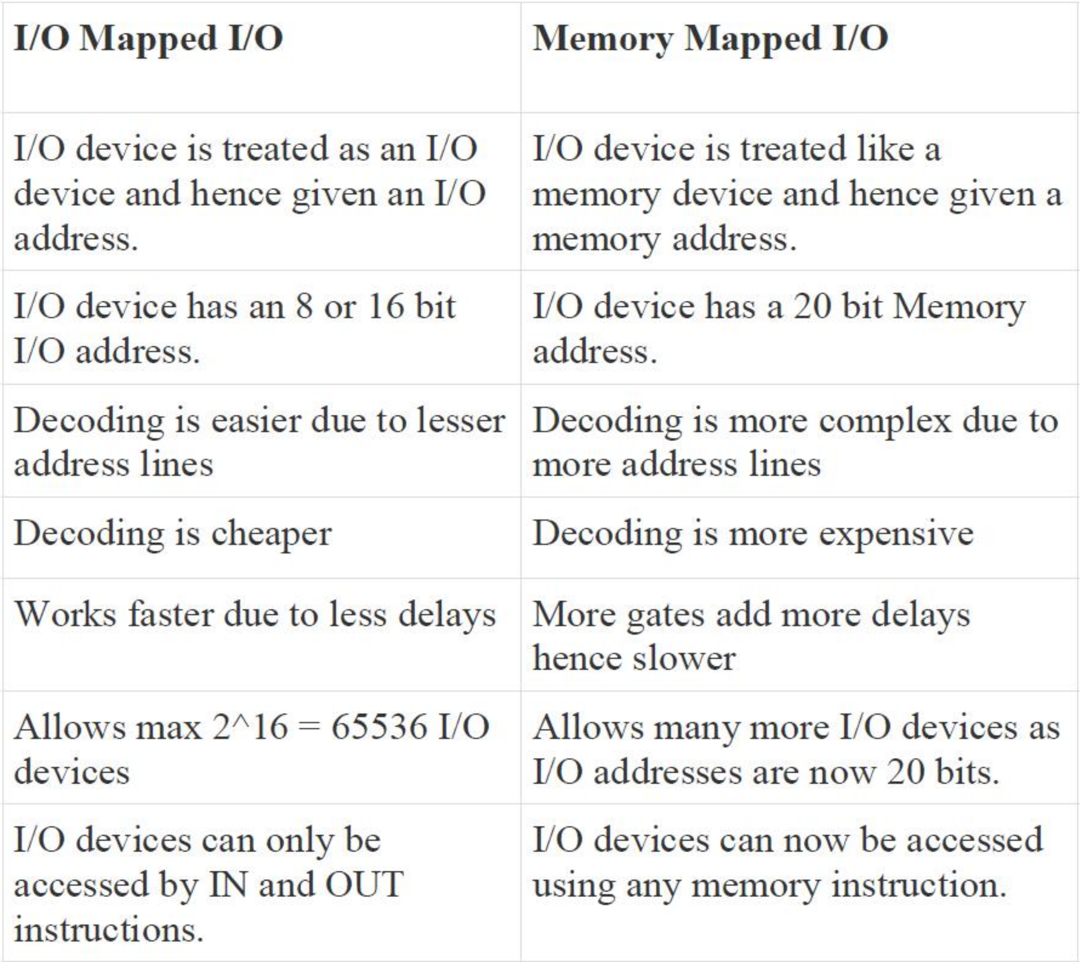
Figure 11-18 DMA transfer in a computer system.

## 9. State how isolated I/O(I/O mapped I/O) is different from memory mapped I/O. Explain nesting of interrupt requests.

In isolated I/O memory and I/O have different address space, but in memory mapped I/O memory and I/O have same address space.

In Isolated I/O Separate instruction control read and write operation in I/O and Memory, but in memory mapped I/O Same instructions can control both I/O and Memory.

Isolated IO is complex due to separate separate logic is used to control both, whereas Memory mapped I/O is simpler logic is used as I/O is also treated as memory only.



| I/O Mapped I/O | Memory Mapped I/O |
|---|---|
| I/O device is treated as an I/O device and hence given an I/O address. | I/O device is treated like a memory device and hence given a memory address. |
| I/O device has an 8 or 16 bit I/O address. | I/O device has a 20 bit Memory address. |
| Decoding is easier due to lesser address lines | Decoding is more complex due to more address lines |
| Decoding is cheaper | Decoding is more expensive |
| Works faster due to less delays | More gates add more delays hence slower |
| Allows max 2^16 = 65536 I/O devices | Allows many more I/O devices as I/O addresses are now 20 bits. |
| I/O devices can only be accessed by IN and OUT instructions. | I/O devices can now be accessed using any memory instruction. |

## 10. What is interrupt? Briefly explain the types of interrupts.

When a Process is executed by the CPU and when a user Request for another Process then this will create disturbance for the Running Process. This is also called as the Interrupt.

Interrupts can be generated by User, Some Error Conditions and also by Software's and the hardware's. But CPU will handle all the Interrupts very carefully because when Interrupts are generated then the CPU must handle all the Interrupts Very carefully means the CPU will also Provide Response to the Various Interrupts those are generated. So that When an interrupt has Occurred then the CPU will handle by using the Fetch, decode and Execute Operations.

Interrupts allow the operating system to take notice of an external event, such as a mouse click. Software interrupts, better known as exceptions, allow the OS to handle unusual events like divide-by-zero errors coming from code execution.

The sequence of events is usually like this:

Hardware signals an interrupt to the processor

The processor notices the interrupt and suspends the currently running software

The processor jumps to the matching interrupt handler function in the OS

The interrupt handler runs its course and returns from the interrupt

The processor resumes where it left off in the previously running software

The most important interrupt for the operating system is the timer tick interrupt. The timer tic interrupt allows the OS to periodically regain control from the currently running user process. The OS can then decide to schedule another process, return back to the same process, do housekeeping, etc. The timer tick interrupt provides the foundation for the concept of preemptive multitasking.

TYPES OF INTERRUPTS

Generally there are three types of Interrupts those are Occurred For Example

1) Internal Interrupt
2) External Interrupt.
3) Software Interrupt.

1.Internal Interrupt:

• When the hardware detects that the program is doing something wrong, it will usually generate an interrupt usually generate an interrupt.

– Arithmetic error - Invalid Instruction

– Addressing error - Hardware malfunction

– Page fault – Debugging

• A Page Fault interrupt is not the result of a program error, but it does require the operating system to get control.

The Internal Interrupts are those which are occurred due to Some Problem in the Execution For Example When a user performing any Operation which contains any Error and which contains any type of Error. So that Internal Interrupts are those which are occurred by the Some Operations or by Some Instructions and the Operations those are not Possible but a user is trying for that Operation. And The Software Interrupts are those which are made some call to the System for Example while we are Processing Some Instructions and when we wants to Execute one more Application Programs.

2.External Interrupt:

• I/O devices tell the CPU that an I/O request has completed by sending an interrupt signal to the processor.

• I/O errors may also generate an interrupt.

• Most computers have a timer which interrupts the CPU every so many interrupts the CPU every so many milliseconds.

The External Interrupt occurs when any Input and Output Device request for any Operation and the CPU will Execute that instructions first For Example When a Program is executed and when we move the Mouse on the Screen then the CPU will handle this External interrupt first and after that he will resume with his Operation.

3.Software interrupts:

These types if interrupts can occur only during the execution of an instruction. They can be used by a programmer to cause interrupts if need be. The primary purpose of such interrupts is to switch from user mode to supervisor mode.

A software interrupt occurs when the processor executes an INT instruction. Written in the program, typically used to invoke a system service. A processor interrupt is caused by an electrical signal on a processor pin. Typically used by devices to tell a driver that they require attention. The clock tick interrupt is very common; it wakes up the processor from a halt state and allows the scheduler to pick other work to perform.

A processor fault like access violation is triggered by the processor itself when it encounters a condition that prevents it from executing code. Typically when it tries to read or write from unmapped memory or encounters an invalid instruction.