I/O Organization

Revya Naik V

Assistant Professor

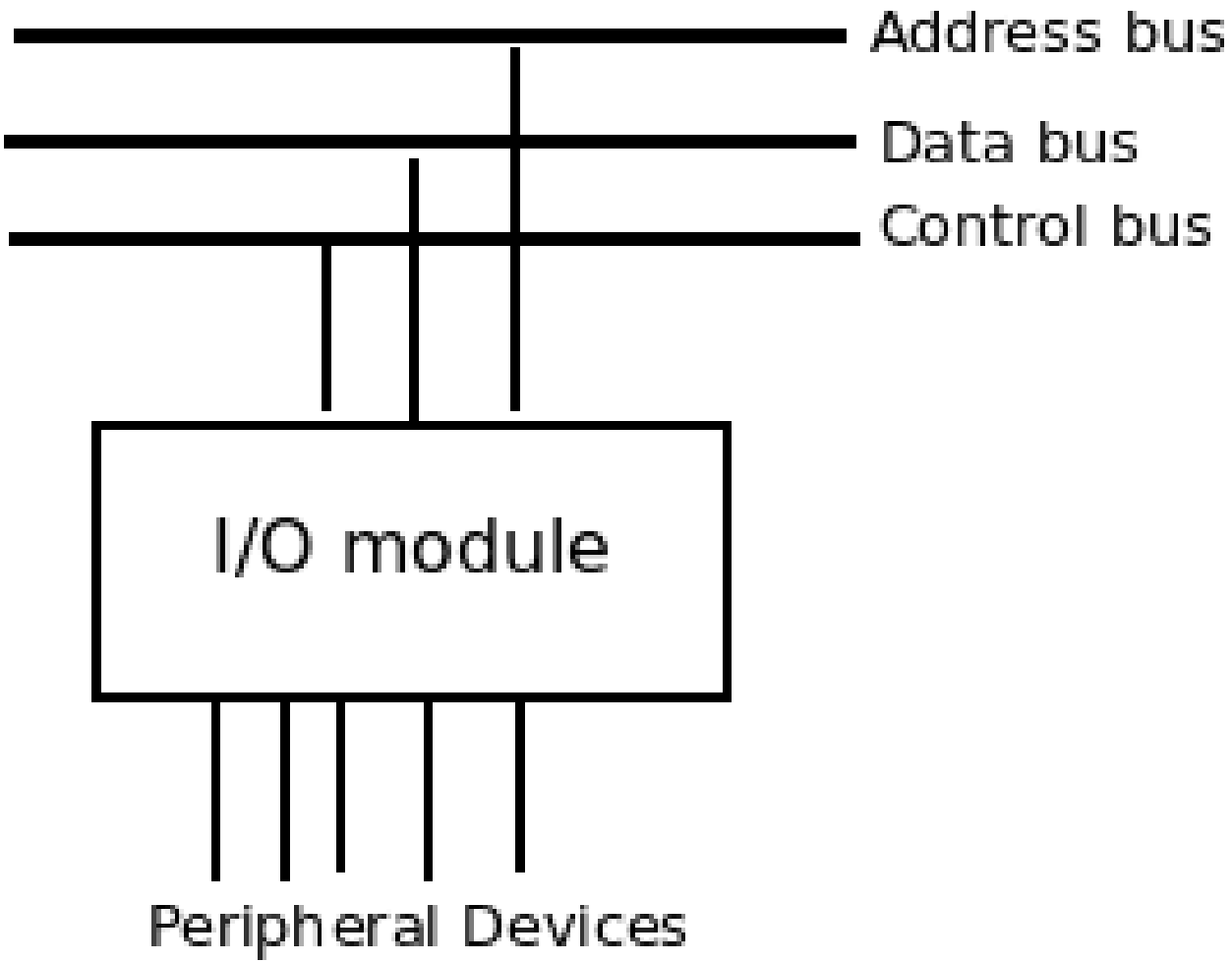
Computer science & engineering dept

RGUKT Basar

I/O module

- I/O module contains logic for performing a communication function between the peripheral and system bus.
- Data transfer of peripheral devices are much slower than processor, so its impractical to use high speed system directly with peripherals.
- Data transfer rate of some peripherals is faster than memory or processor.
- I/O module required to interface memory and processor via system bus.

I/O module


Address bus
Data bus
Control bus
I/O module
Peripheral Devices

I/O module functions

- Control and timing
- Processor communication
- Device communication
- Data buffering
- Error detection

Control and timing

Control and timing requirement, to coordinate the flow of traffic between internal resource and external devices

Ex:the control of the transfer of data from an external to the processor might involve following steps

- The procssor intergorates the I/O module to check the status of the attached device.
- The I/O moudel returns the device status.
- If the device is operational and ready to transmit the processor requests the transfer of data, by means of a command to the I/O moudle
- The I/O module obtains unit of data (e.g, 8 or 16 bits) from the external device.
- The data are transferred from the I/O moudule to the processor.

Processor communications

Commond decoding:

- The I/O module accepts commands from the processor, typically sent as signals on the control bus.
- Ex: an I/O module for a disk drive might accept the following commands: READ SECTOR,WRITE SECTOR, SEEK track number.

Data: Data are exchanged between the processor and the I/O module over the data bus.

Status reporting: Because peripherals are so slow, it is important to know the status of the I/O module. Status singnals are BUSY and READY.

Address recognition: I/O module must recognize one unique address for each peripheral it controls.

I/O module functions

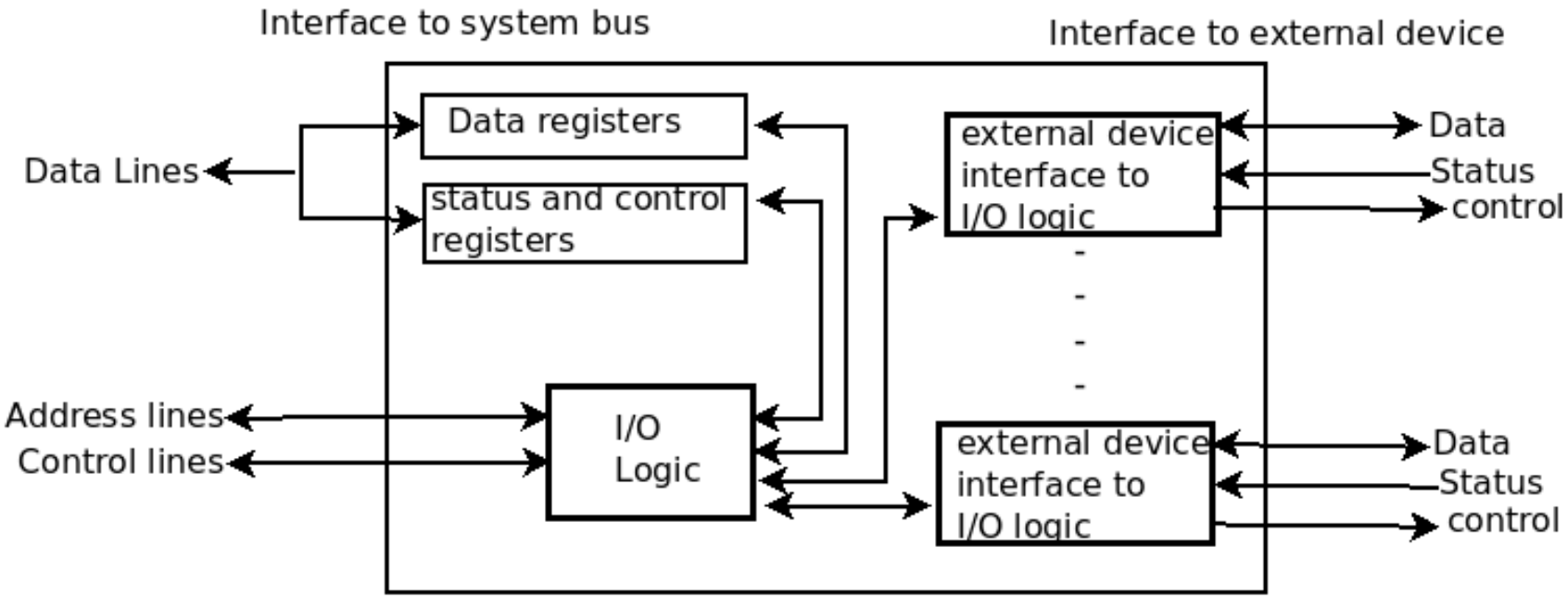
Device communication involves command, status information and control

- Status signals from I/O module
- Status signals to I/O module
- Data bits to and from I/O module

Data Buffering: Data coming from main memory are sent to an I/O module in a rapid burst. The data are buffered in the I/O module and then sent to the pripheral device at its data rate.

- I/O module **detect errors** and reporting to the processor.
- Ex: paper jam, transmission error (parity checker)

I/O module



Block Diagram of I/O module

Memory mapped I/O & I/O mapped I/O

- Processor, main memory and I/O share common bus. Single address bus for memory and I/O devices.
- Size of address bus of a processor is 16, then there are 2^16 combinations, some addres lines for I/O devices and other addresses used for memory locations
- IO/M complement command line used to identify a memory location or an I/O device
- IO/M complement=1 indicates address present in address bus is address of an I/O device
- IO/M complement=0 indicates address present in address bus is the address of memory location

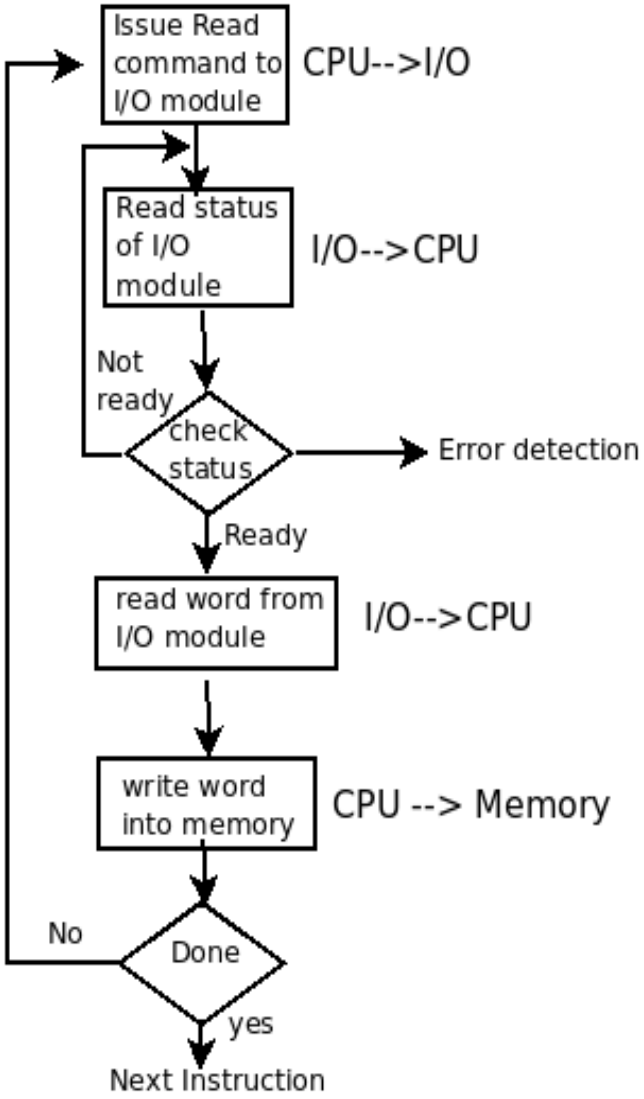
Programmed I/O

- With programmed I/O, data are exchanged between the processor and the I/O module.
- Processor executes a program that gives it direct control of the I/O operation, including sensing device status, sending a read or write command, and transferring the data.
- When the processor issues a command to the I/O module, it must wait unitl the I/O operation complete.
- There are four types of I/O commands that an I/O module may receive when it is addressed by a prccessor

Programmed I/O

- **Control**: used to activate a peripheral and tell it what to do
- **Test**: used to test various status conditions associated with an I/O module and its peripherals.
- **Read**: the I/O to obtain an item of data from the perpheral and place it in internal buffer.The processor can obtain the data item by requesting that the I/O module place it on data bus.
- **Write** :I/O module take from data bus and transimit data to perpheral.

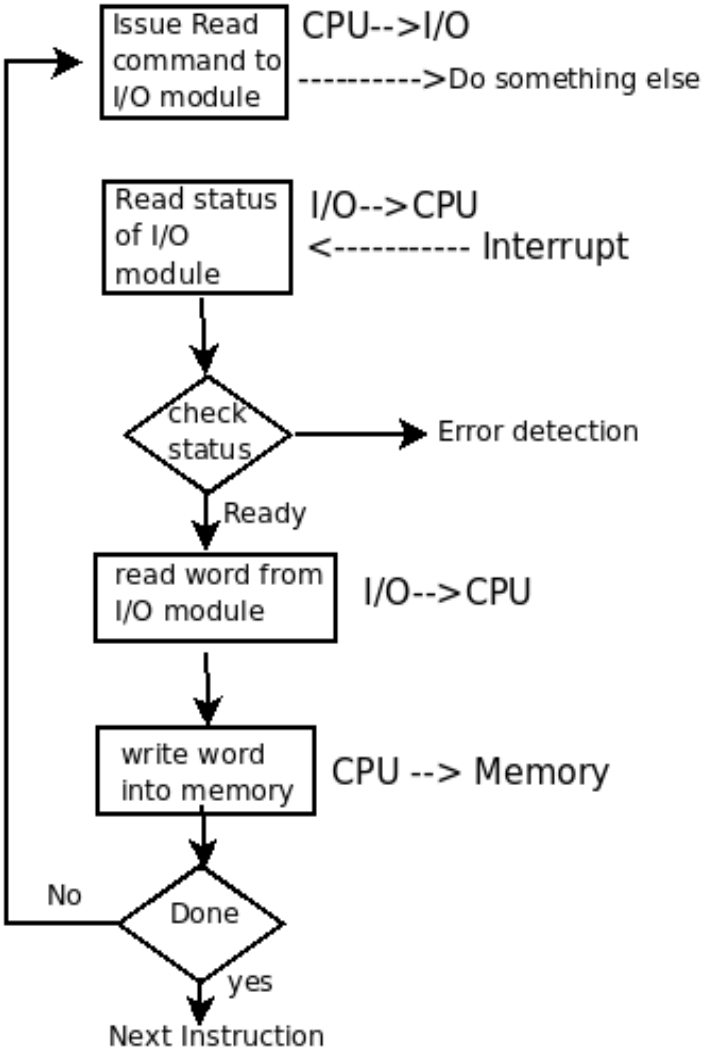
Programmed I/O flowchart



Interrupt Driven I/O

- Program I/O processor repeatly checks I/O status of I/O module when its ready. Processor wasting lot of time for checking I/O module.
- In Interrupt driven I/O checks I/O module status and do its other work. When I/O ready to exchange data with processor then I/O module send interrupt to processor.

Interrupt driven I/O flowchart



Interrupt line

Software poll: Interrupt service routine whose job is to poll(processor raise Test/IO) each I/O module to determine which module caused interrupt.

Daisy chain: all I/O modules are connected to common interrupt line then I/O module raise interrupt to processor. Processor sends INTA through all I/O modules. Only particular I/O module accepts INTA.

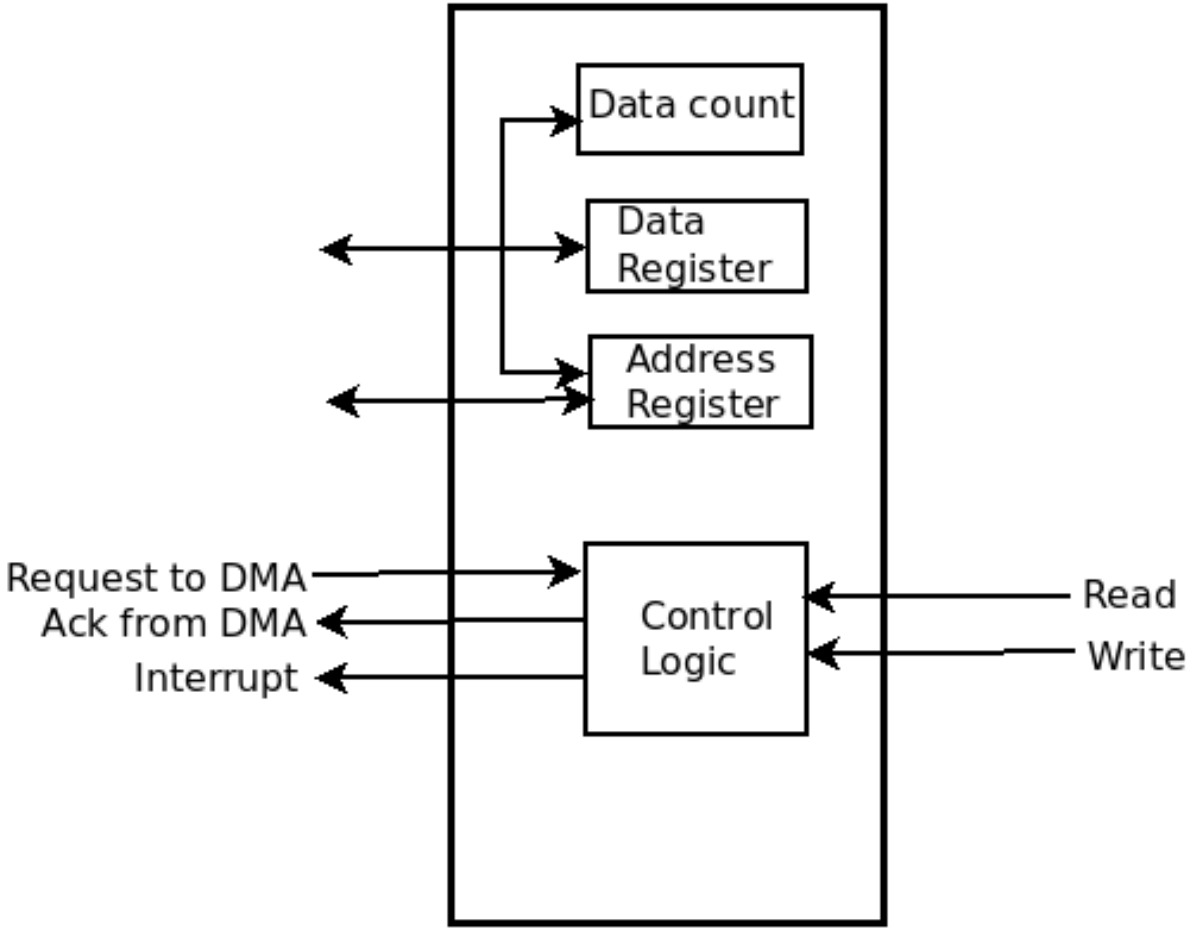
Disadvantage: daisy chain is efficient and software poll is time consumming.

Bus arbitration: I/O module first gain control of the bus before it raise the interrupt request line, thus only one module raise the interrupt line at a time. Processor detects interrupt, it responds on the INTA line

Direct memory acees(DMA)

- Interrupt-driven I/O more efficient than simple programmed I/O, still processor requires to transfer data between memory and an I/O module, and any data transfer must traverse throguh the processor.
- DMA is more efficient technique to transfers large volume of data between I/O module and memory.
- The DMA module takes control of system bus to transfer data to and from memory over system bus.
- DMA must use the bus only when the processor does not need it, or it must force the processor to suspend operation temporarily. This technique referred to as cycle stealing.

Direct memory access(DMA)



Block Diagram of DMA

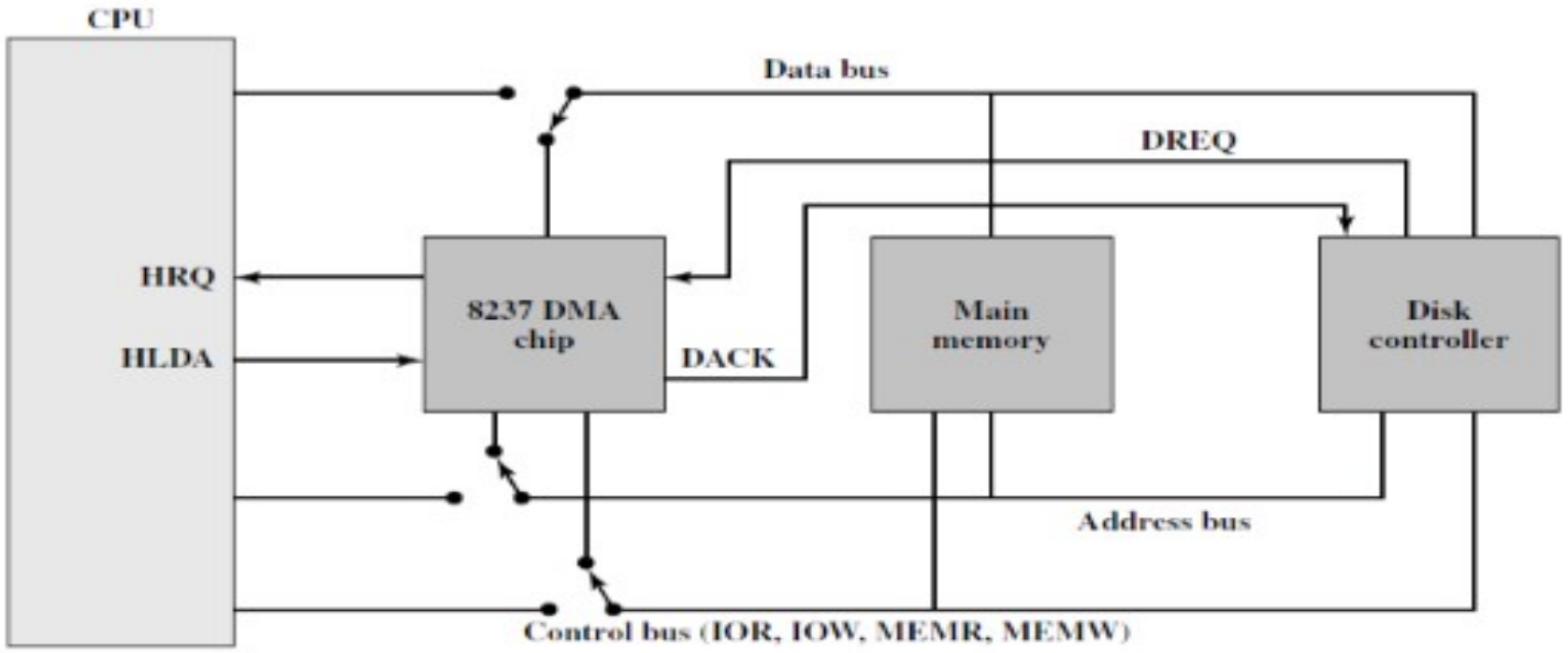
Direct memory access(DMA)

- Whether a read or write is requested, using the read or write control line between the processor and the DMA module.
- The address of the I/O device involved, communicated on the data lines.
- The starting location in memory to read from or wirte to , communicated on the data lines and stored by the DMA module in its address register.
- The number of words to be read or written , again communicated via the data lines and stored in the data count register.
- The DMA module transfers the entire block of data one word at a time, directly to or from memory, without going through the processor. When the transfer is complete, the DMA module sends an interrupt signal to the processor.
- The processor involved at the beginning and end of the transfer.

DMA configuration

- All modules share the same system bus. Uses programmed I/O to exchange data between memory and I/O module through the DMA module. each transfer of a word consumes two bus cycles.
- The number of bus cycles can be reduced by integrating the DMA and I/O functions. There is a path between the DMA module and one or more I/O modules that does not include system bus.
- I/O modules and the DMA modules connected using I/O bus. For exchanging data between DMA and memory system bus is used. But system bus is off transfeering data between DMA and I/O modules.

DMA controller



DACK = DMA acknowledge
DREQ = DMA request
HLDA = HOLD acknowledge
HRQ = HOLD request

Figure 9

DMA controller

Following steps to transfer a block of data from memory to disk

- The peripheral device will request the service of DMA by pulling DREQ (DMA request) high.
- The DMA will put a high on its HRQ (hold request), signaling the CPU through its HOLD pin that it needs to use the buses.
- The CPU will finish the present bus cycle and respond to the DMA request by putting high on its HDLA (hold acknowledge), thus telling the 8237 DMA that it can go ahead and use the buses to perform its task. Hold must remain active high as long as DMA is performing its task.

DMA controller

- DMA will activate DACK which tells the perpheral device that it will start to transfer data.
- DMA starts to transfer the data from memory to peripheral by putting the address of the first byte of the block on the address bus and activating MEMR. Then DMA drerements the counter and increment the addres pointer and repeats this process untill reaches zero and the task is finished.
- Aafter DMA finsihed, deactivate HRQ, signaling the CPU that it can regain control over its buses.

Alternate DMA configuration

Privileged and non-privileged instructions

- Operating system has two types modes, user modes and kernel mode.
- The processor is said to be in privileged mode when the functions in OS kernel are executing
- Privileged instructions are shutdown, change contents of control register, jump into kernel code and commands sending to I/O device.
- Non-privileged instructions (load,store, add, subtract) executed user mode. Suppose privileged instruction attempt made to execute in usermode which causes run time error.