Computer Organization and Architecture

Module 6

Prof. Indranil Sengupta

Dr. Sarani Bhattacharya

Department of Computer Science and Engineering

IIT Kharagpur

Direct Memory Access (DMA)

Introduction

- In the data transfer methods discussed under programmed I/O, it is assumed that machine instructions are used to transfer the data between I/O device and memory.
  - Not very suitable when large blocks of data are required to be transferred at high speed (e.g. transfer of a disk block).
- An alternate approach is *Direct Memory Access* (DMA).
  - Allows transfer of a block of data directly between an I/O device and memory, without continuous CPU intervention.

- Why programmed I/O is not suitable for high-speed data transfer?

    a) Several program instructions have to be executed for each data word transferred between the I/O device and memory.

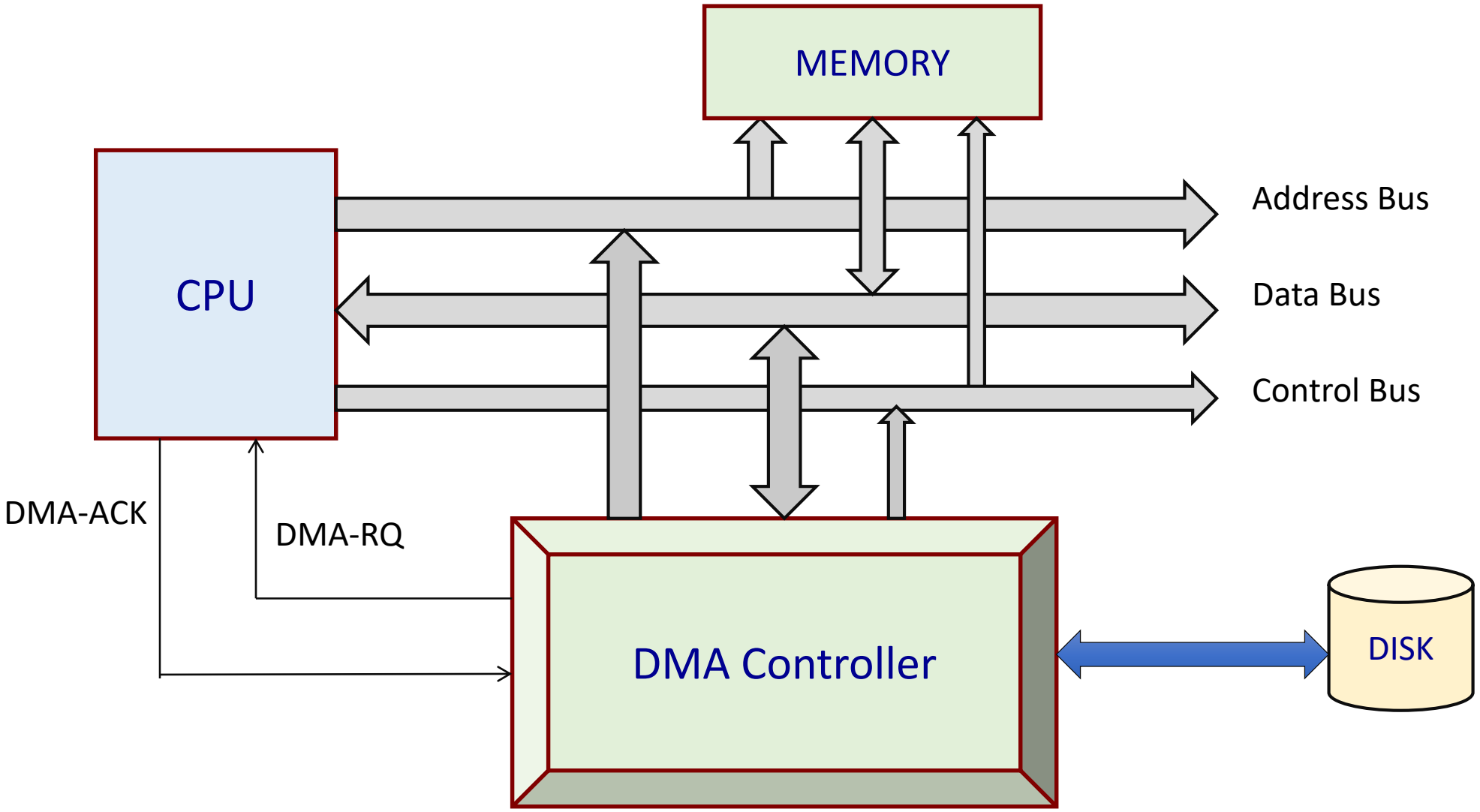
        - Suppose 20 instructions are required for each word transfer.
        - The CPI of the machine running at 1 GHz clock is 1.
        - So, 20 nsec is required for each word transfer → maximum 50 M words/sec
        - Data transfer rates of fast disks are higher than this figure.

b) Many high speed peripheral devices like disk have a synchronous mode of operation, where data are transferred at a fixed rate.

- Consider a disk rotating at 7200 rpm, with average rotational delay of 4.15 msec.
- Suppose there are 64 Kbytes of data recorded in every track.
- Once the disk head reaches the desired track, there will be a sustained data transfer at rate 64 Kbytes / 4.15 msec = 15.4 MBps.
- This sustained data transfer rate is comparable to the memory bandwidth, and cannot be handled by programmed I/O.

DMA Controller

- A hardwired controller called the *DMA controller* can enable direct data transfer between I/O device (e.g. disk) and memory without CPU intervention.
  - No need to execute instructions to carry out data transfer.
  - Maximum data transfer speed will be determined by the rate with which memory read and write operations can be carried out.
  - Much faster than programmed I/O.


MEMORY
CPU
Address Bus
Data Bus
Control Bus
DMA-ACK
DMA-RQ
DMA Controller
DISK

Steps Involved

a) When the CPU wants to transfer data, it initializes the DMA controller.
   - How many bytes to transfer, address in memory for the transfer.

b) When the I/O device is ready for the transfer, the DMA controller sends *DMA-RQ* signal to the CPU.

c) CPU waits till the next DMA breakpoint, relinquishes control of the bus (i.e. puts them in high impedance state), and sends *DMA-ACK* to DMA controller.

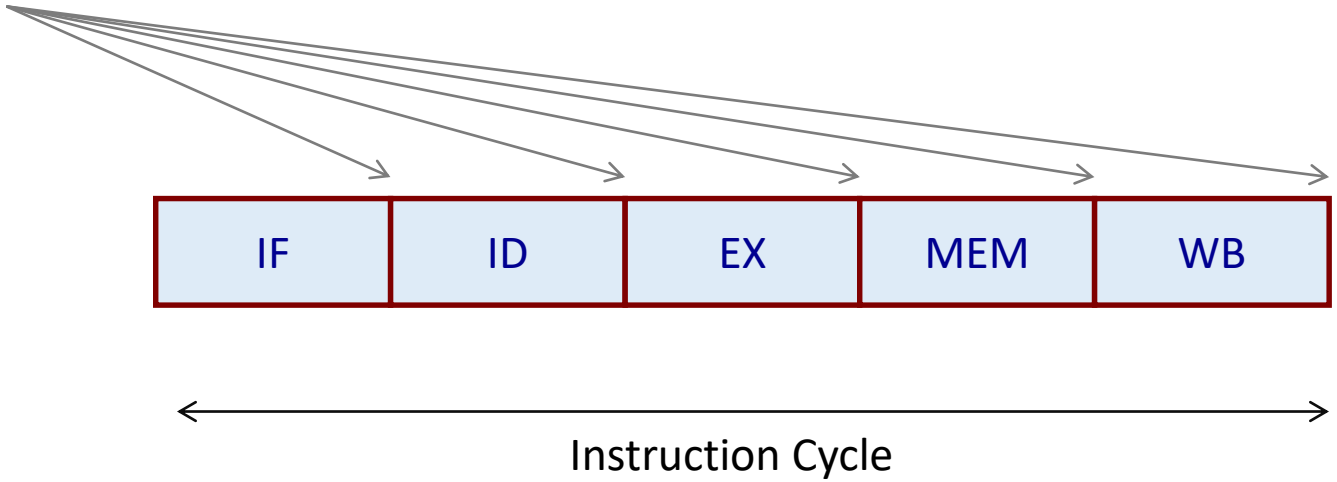
d) Now DMA controller enables its bus interface, and transfers data directly to/from memory.

e) When done, it deactivates the *DMA-RQ* signal.

f) The CPU again begins to use the bus to access memory.

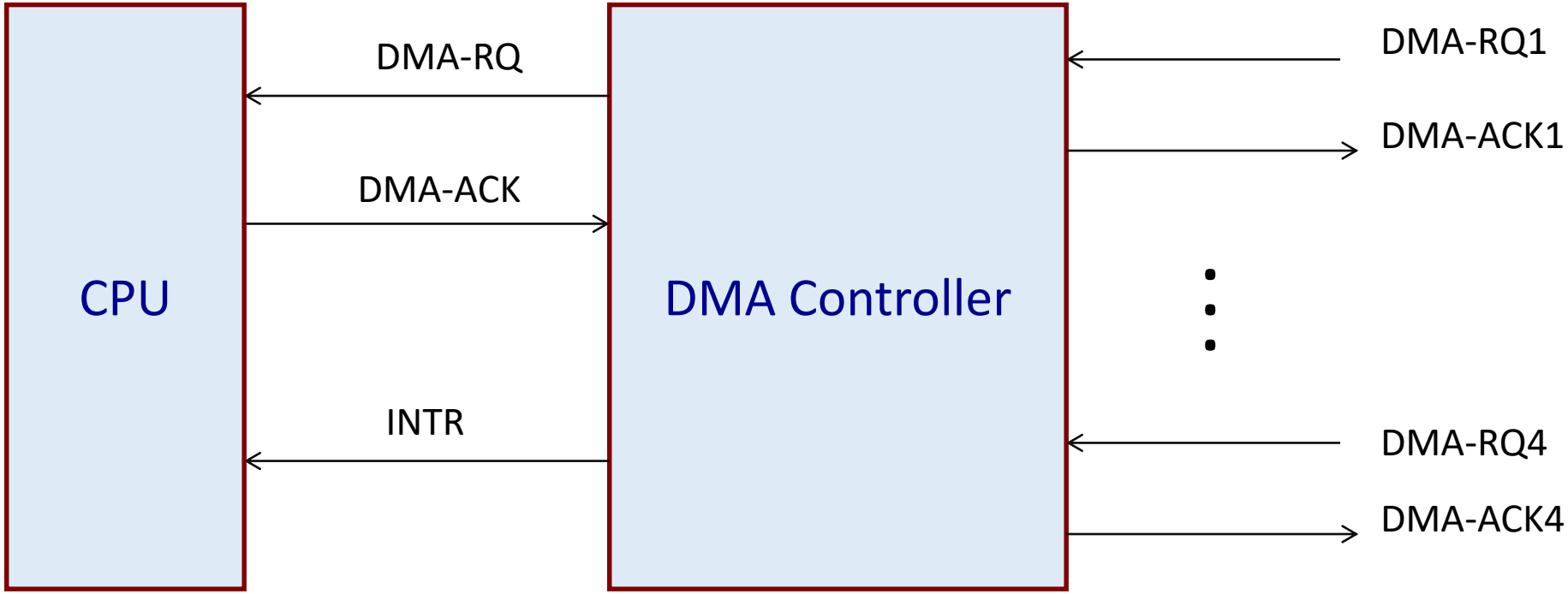
- The DMA breakpoints:
  - DMA request can be acknowledged at the end of any machine cycle.

DMA breakpoints



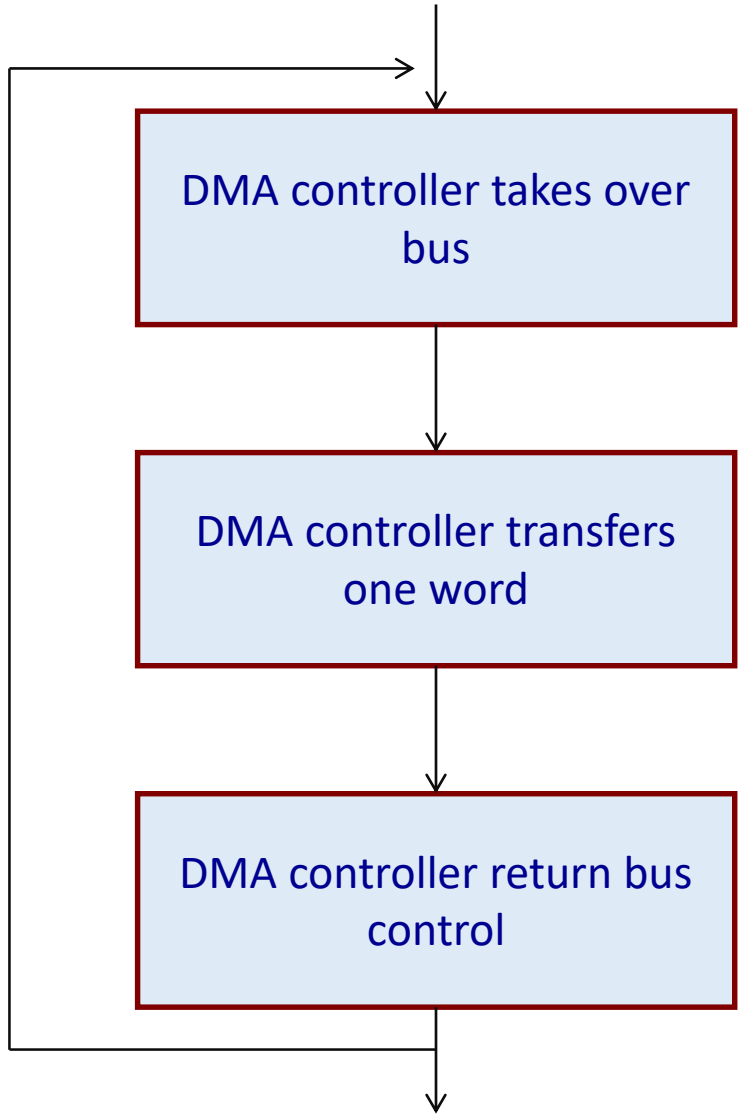
Why cannot we have interrupt breakpoints at the end of any machine cycle?

- For every DMA channel, the DMA controller will have three registers:
    a) Memory address
    b) Word count
    c) Address of data on disk
  - CPU initializes these registers before each DMA transfer operation.
  - Before the data transfer, DMA controller requests the memory bus from the CPU.
  - When the data transfer is complete, the DMA controller sends an interrupt signal to the CPU.

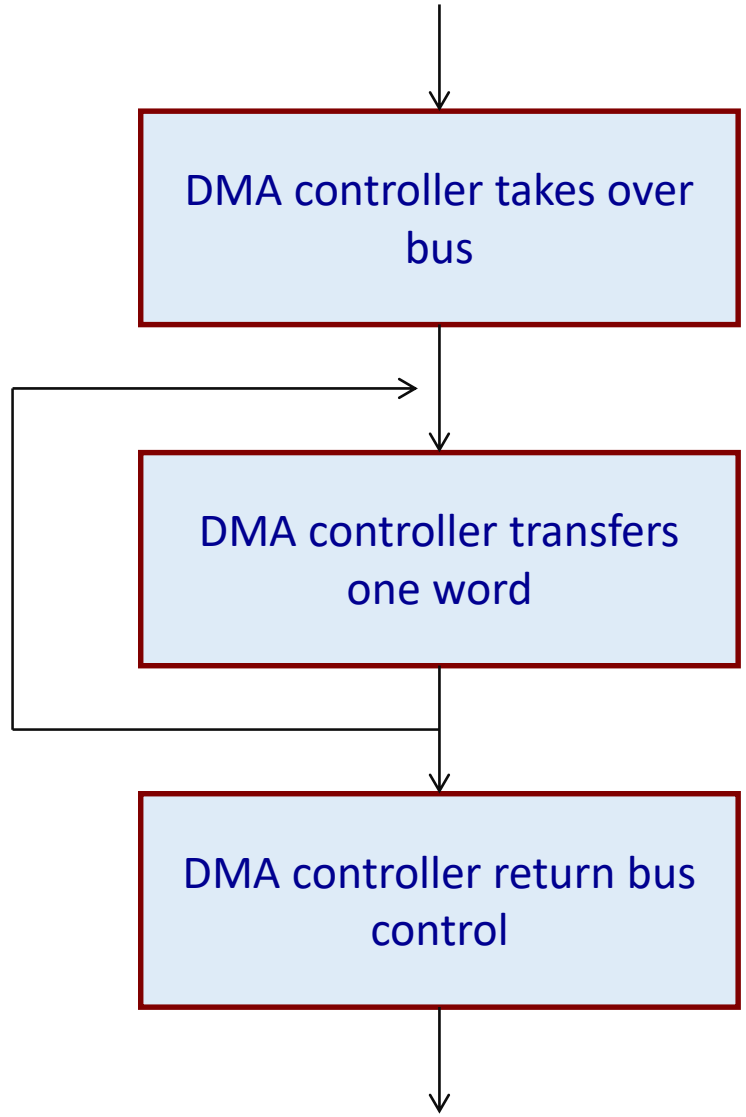

CPU
DMA-RQ
DMA-ACK
INTR
DMA Controller
DMA-RQ1
DMA-ACK1
DMA-RQ4
DMA-ACK4

DMA Transfer Modes

- DMA transfer can take place in two modes:
  - **a) DMA cycle stealing**
    - The DMA controller requests for the for a few cycles 1 or 2.
    - Preferably when the CPU is not using memory.
    - DMA controller is said to steal cycles from the CPU without the CPU knowing it.
  - **b) DMA block transfer**
    - The DMA controller transfers the whole block of data without interruption.
    - Results in maximum possible data transfer rate.
    - CPU will lie idle during this period as it cannot fetch any instructions from memory.



Cycle Stealing Mode



Block Transfer Mode

Others Applications of DMA

- Other than data transfer to/from high-speed peripheral devices, DMA can be used in some other areas as well:
  - High-speed memory-to-memory block move.
  - Refreshing dynamic memory systems, by periodically generating dummy read requests to the columns.

Practice Problems

Problem 1

- Suppose we want to read 2048 bytes in programmed I/O mode of transfer. The bus width is 32 bits. Each time an interrupt occurs, it takes 4 µsec to service it (i.e. transfer 32 bits). How much CPU time is required to read 2048 bytes?

Problem 2

- A DMA module is transferring bytes to main memory from an external device at 76800 bps. The CPU can fetch instructions at a rate of 2 million instructions per second. Assume instruction size is 32 bits. How much will the processor be slowed down due to DMA activity?

Problem 3

- A DMA controller transfers 32-bit words to memory using cycle stealing. The words are assembled from a device that transmits bytes at a rate of 2400 bytes per second. The CPU is fetching and executing instructions at an average rate of 1 million instructions per second. By how much time will the CPU be slowed down because of the DMA transfer?

Problem 4

- Consider a system employing interrupt-driven I/O for a device that transfers data at 8 KB/s on a continuous basis. The interrupt processing takes about 100µsec and the I/O device interrupts the CPU for every byte.

  While executing the ISR, the processor takes about 8µsec for the transfer of each byte. What is the fraction of CPU time consumed by the I/O device?

Problem 5

- Consider a disk drive with 16 surfaces, 512 tracks per surface, and 512 sectors per track, 1024 bytes per sector, and a rotation speed of 3600 rpm. The disk is operated in cycle stealing mode whereby whenever one 4-byte word is ready, it is sent to memory. Similarly for writing, the disk interface reads a 4-byte word from memory in each DMA cycle. The memory cycle time is 40 nsec. Find the maximum percentage of time that the CPU gets blocked during DMA operation.

Problem 6

- A hard disk is connected to a 50 MHz processor through a DMA controller. Assume that the initial set-up time for a DMA transfer takes 2000 clock cycles for the processor, and also assume that the handling of the interrupt on DMA completion requires 1000 clock cycles for the processor. The hard disk has a transfer rate of 4000 KB/s and average block size transferred is 8 KB. What fraction of the processor time is consumed by the disk, assuming that data are transferred only during the idle cycles of the CPU?

Problem 7

- A device with transfer rate of 20 KB/s is connected to a CPU. Data is transferred byte wise. Let the interrupt overhead be 6 µsec. The byte transfer time between the device interface register and CPU or memory is negligible. What is the minimum performance gain of operating the device under interrupt-driven mode?