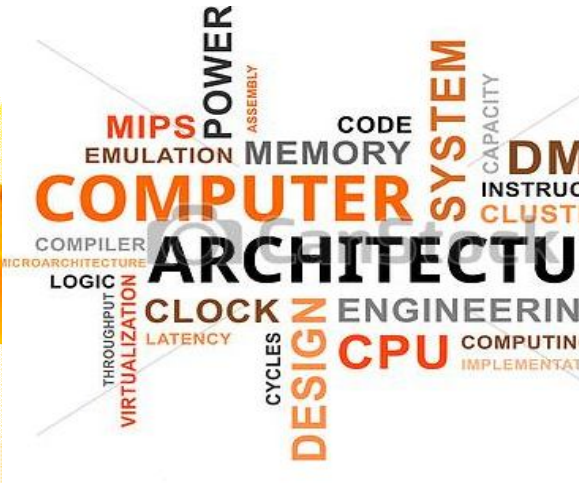
# UNIT V
# I/O ORGANIZATION AND PARALLELISM

Accessing I/O devices – Interrupts – **Direct Memory Access** – Buses– Interface circuits – Standard I/O Interfaces (PCI, SCSI, USB) –Instruction Level Parallelism : Concepts and Challenges – Introduction to multicore processor – Graphics Processing Unit



# Recap the previous Class

# 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.

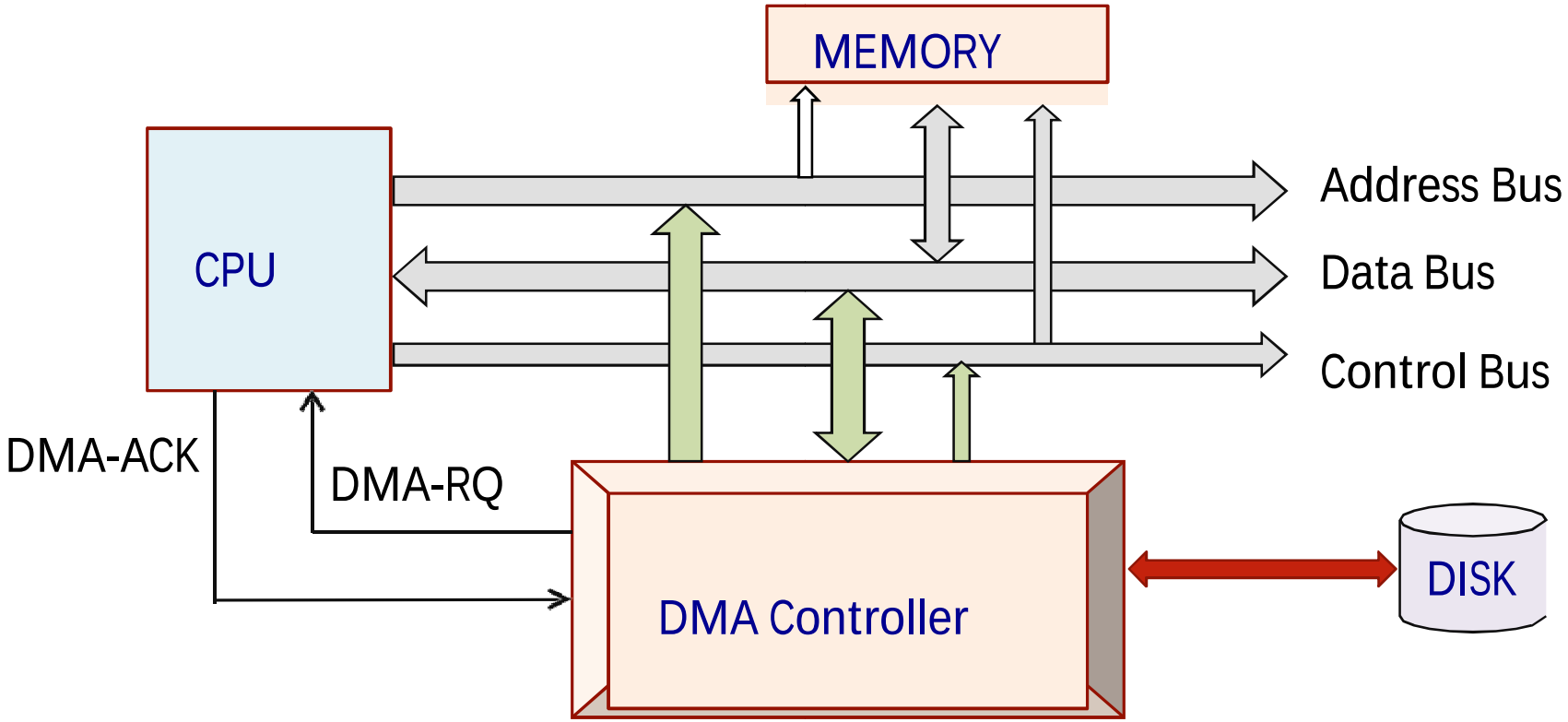
- Suppose **20 instructions are required** for each word transfer.
- The CPI of the machine running at 1 GHz **clock is 1**.
- So, **20 nsec** is required for each word transfer → maximum 50 M words/sec
- Data transfer rates of fast disks are higher than this figure.

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

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.

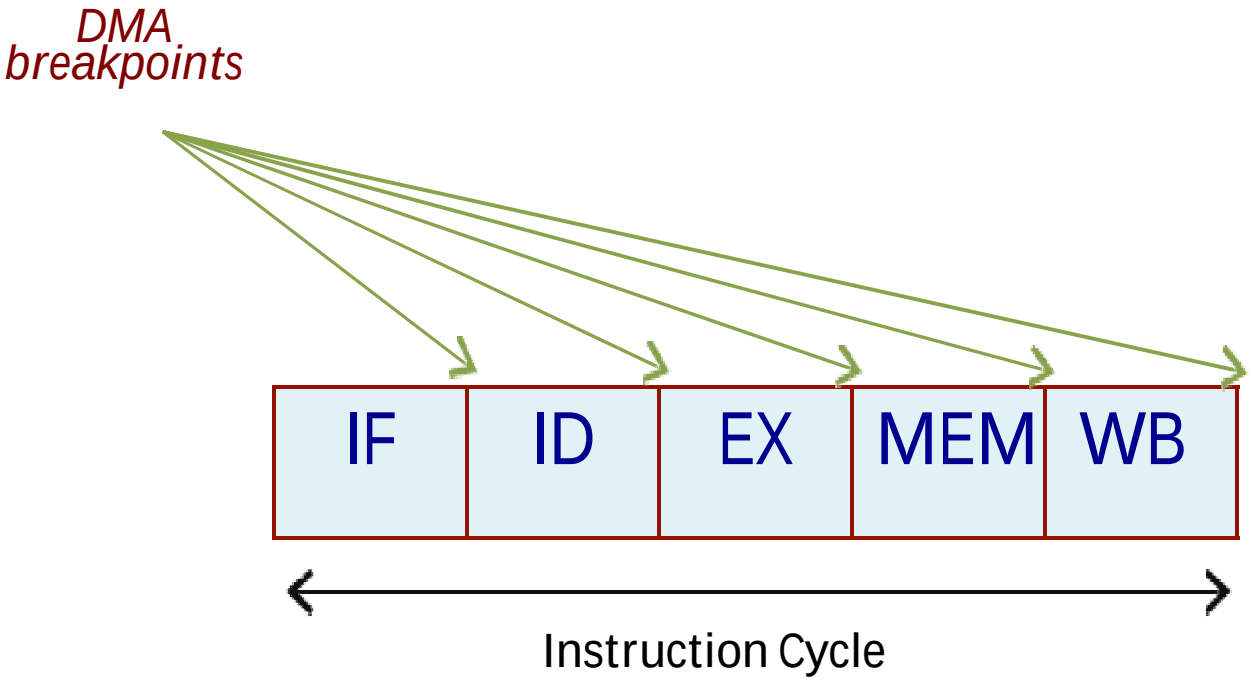
c) CPU **waits till the next DMA breakpoint**, relinquishes control of the bus, and sends DMA-ACK to DMA controller.

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.



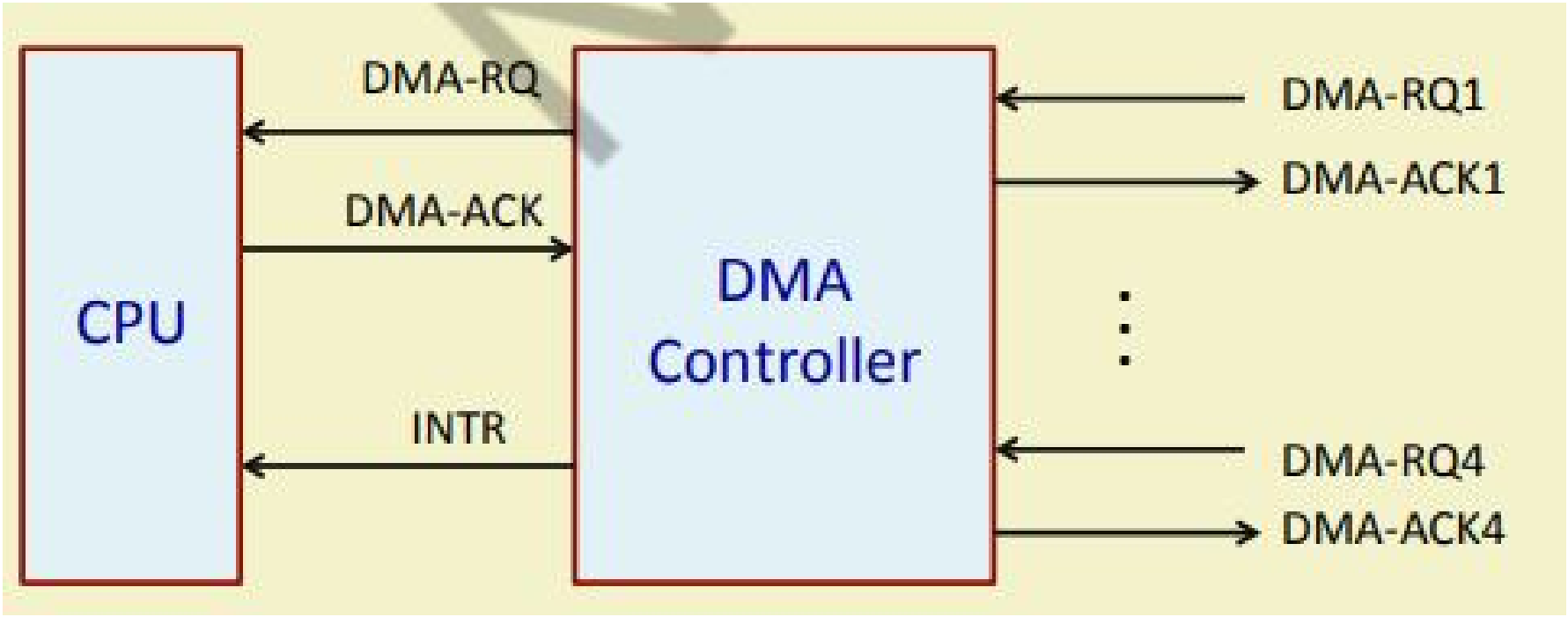
- For every DMA channel, the DMA controller will **have three registers**:

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

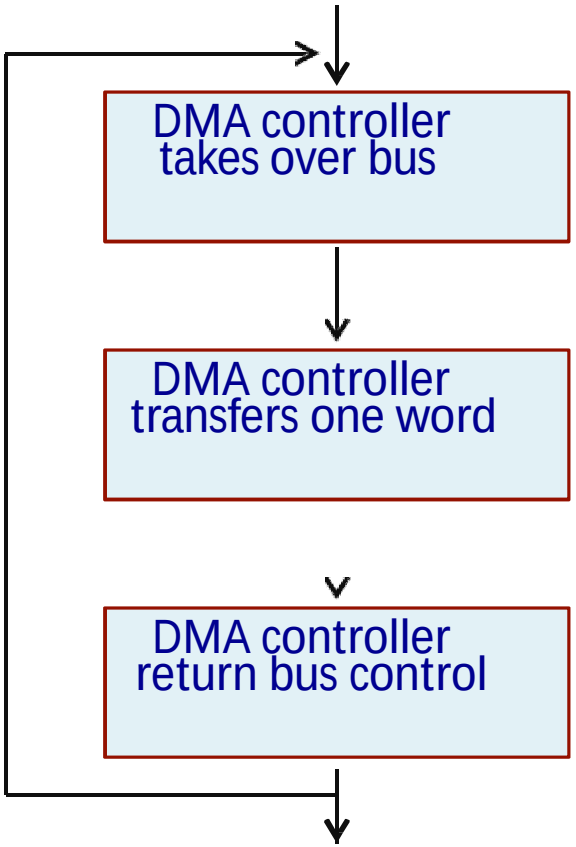
- DMA transfer can take place in two modes:

## a) DMA cycle stealing

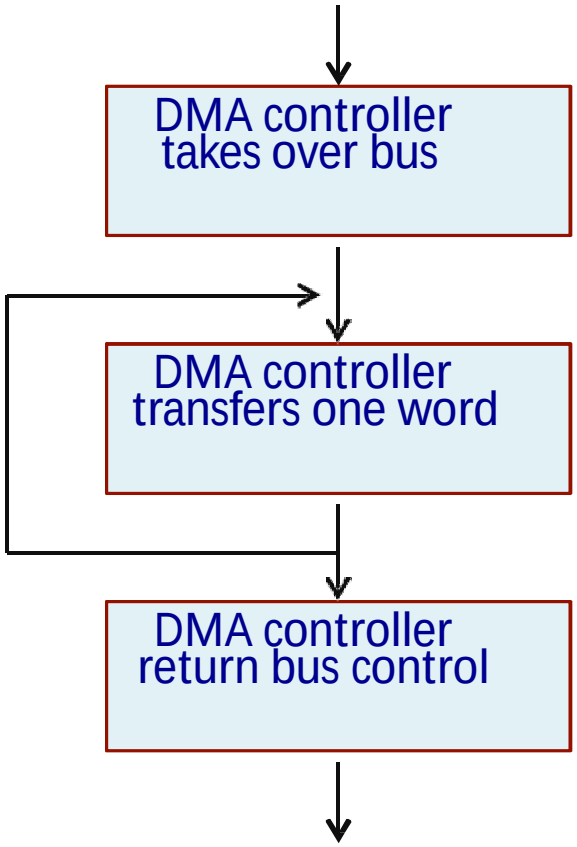
- The DMA controller requests for a **few cycles 1 or 2.**
- Preferably when the CPU is not using memory.
- DMA controller is said to **steal cycles from the CPU without the CPU knowing it**.

## 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.

# TEXT BOOK

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, “Computer Organization”, McGraw-Hill, 6th Edition 2012.

## REFERENCES


1. David A. Patterson and John L. Hennessey, “Computer organization and design”, MorganKauffman ,Elsevier, 5th edition, 2014.
2. William Stallings, “Computer Organization and Architecture designing for Performance”, Pearson Education 8th Edition, 2010
3. John P.Hayes, “Computer Architecture and Organization”, McGraw Hill, 3rd Edition, 2002
4. M. Morris R. Mano “Computer System Architecture” 3rd Edition 2007
5. David A. Patterson “Computer Architecture: A Quantitative Approach”, Morgan Kaufmann; 5th edition 2011


# THANK YOU