Chapter 11: Input/Output Organisation

Lesson 10:
Direct Memory Access (DMA)

Objective

- Understand the concept of Direct memory access to memory for data transfers
- Learn how DMA used when multiple bytes are to be transferred between memory and IO devices
- Learn DMA Data transfer mechanism between I/O devices and system memory with the least processor intervention using DMAC
- Know about the DMAC registers

Direct memory access

Multi-byte data set or burst of data or block of data

- A DMA is required when a multi-byte data set or a burst of data or a block of data is to be transferred between the external device and system or two systems
- A device facilitates DMA transfer with a processing element called DMAC (DMA Controller)

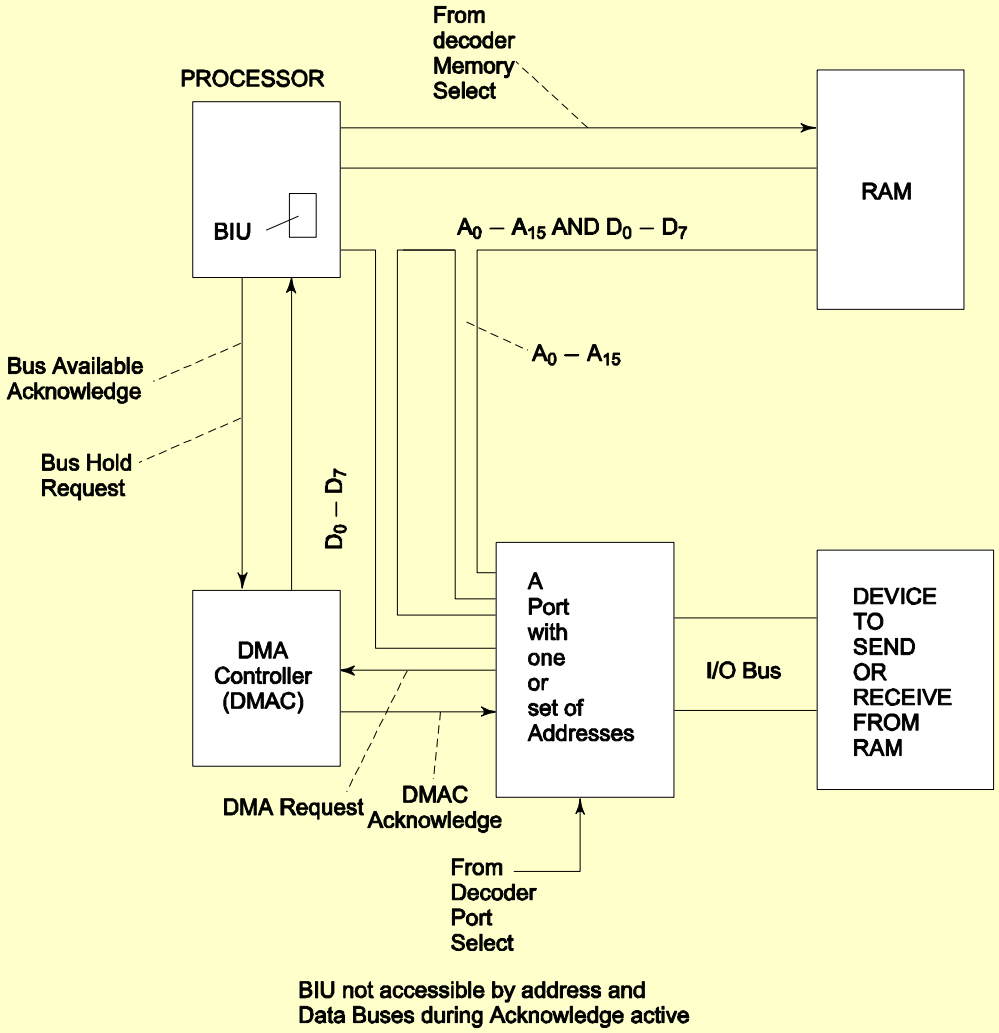
Using a DMA controller (DMAC)

- DMA based method useful, when a block of bytes are transferred, for example, from disk to the RAM or RAM to the disk
- Repeatedly interrupting the processor for transfer of every byte during bulk transfer of data will waste too much of processor time in context switching

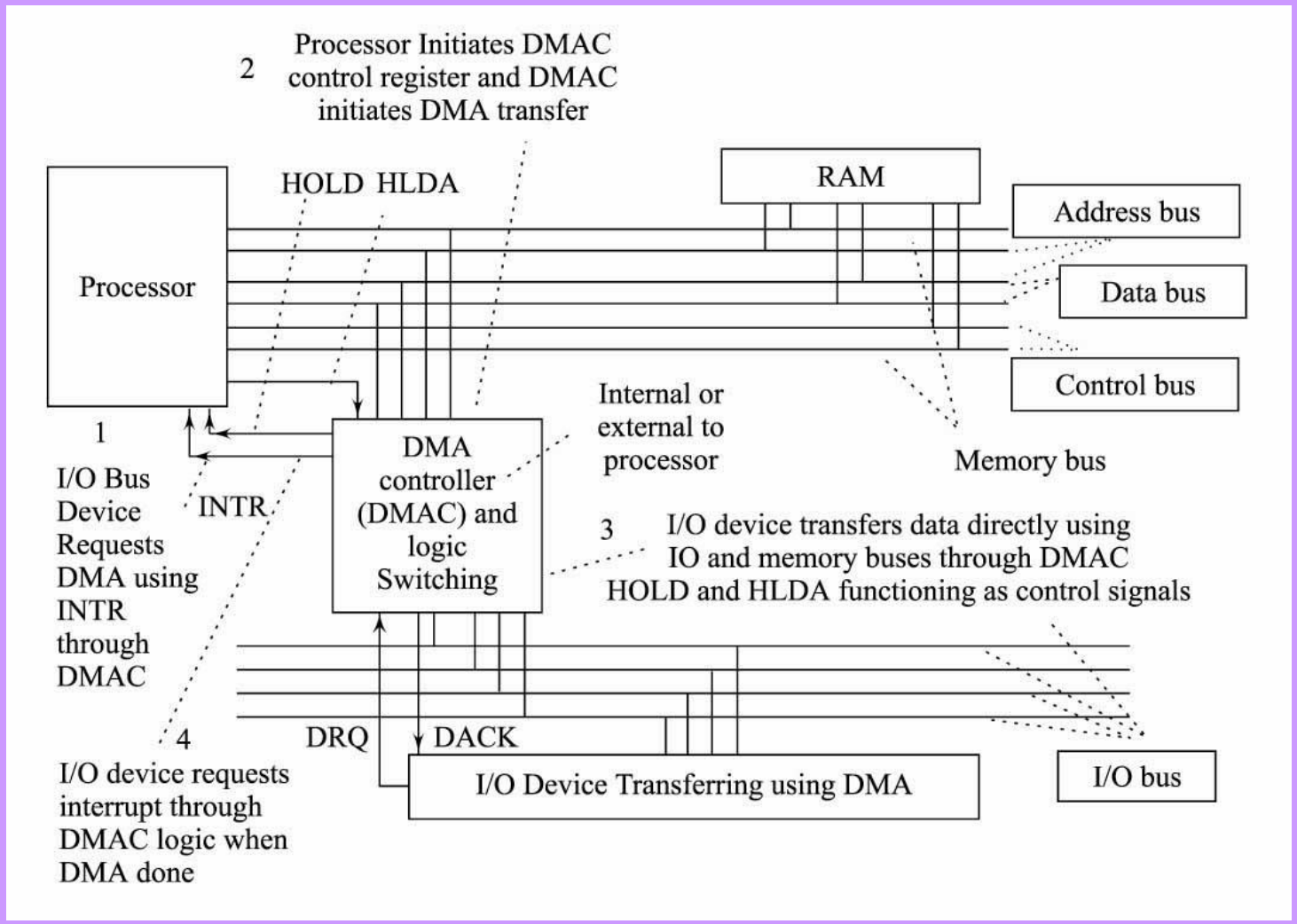
DMAC

- System performance improves by separate processing of the transfers from and to the peripherals (for example, between camera memory and USB port)

Interconnections using a DMAC



Sequence of Events Involved in a DMA Transfer to copy the Results of an I/O Operation into the Main Memory



DMAC hold request

- After an ISR initiates and programs the DMAC, the DMAC sends a hold request to the CPU
- CPU acknowledges that if the system memory buses are free to use

Three modes of Direct memory access

Three modes

- Single transfer at a time and then release of the hold on the system bus
- Burst transfer at a time and then release of the hold on the system bus. A burst may be of a few kB
- Bulk transfer and then release of the hold on the system bus after the transfer is completed

DMA proceeds without the CPU intervening

- Except (i) at the start for DMAC programming and initializing and (ii) at the end
- Whenever a DMA request by external device is made to the DMAC, the CPU is requested (using interrupt signal) the DMA transfer by DMAC at the start to initiate the DMA and at the end to notify (using interrupt signal) the end of the DMA by DMAC

Using a DMA controller

When a DMA controller used to transfer a block of bytes─ ISRs are not called during the transfer of bytes

- An ISR is called only at the beginning of the transfer to program the controller (DMAC)
- Another ISR is called only at the end of the transfer

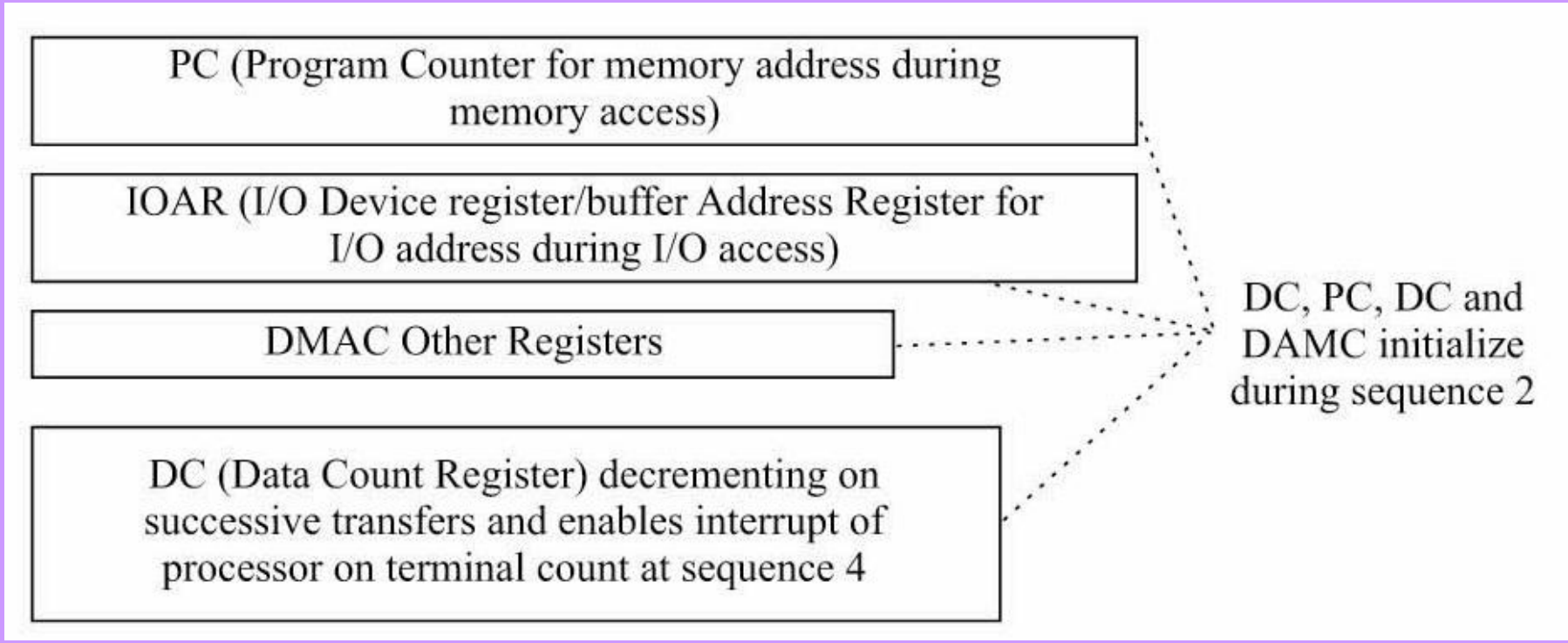
Programming the DMAC registers

Programming the DMAC registers

The ISR that initiates the DMA (Direct Memory Access) to the interrupting source, simply programs the DMA registers for the:

- command (for mode of transfer— bulk or burst or bytes),
- data-count (number of bytes to be transferred),
- memory block address where access to data is made and
- I/O bus for start address of external device

DMAC control registers for the DMAC operations for an IO device channel

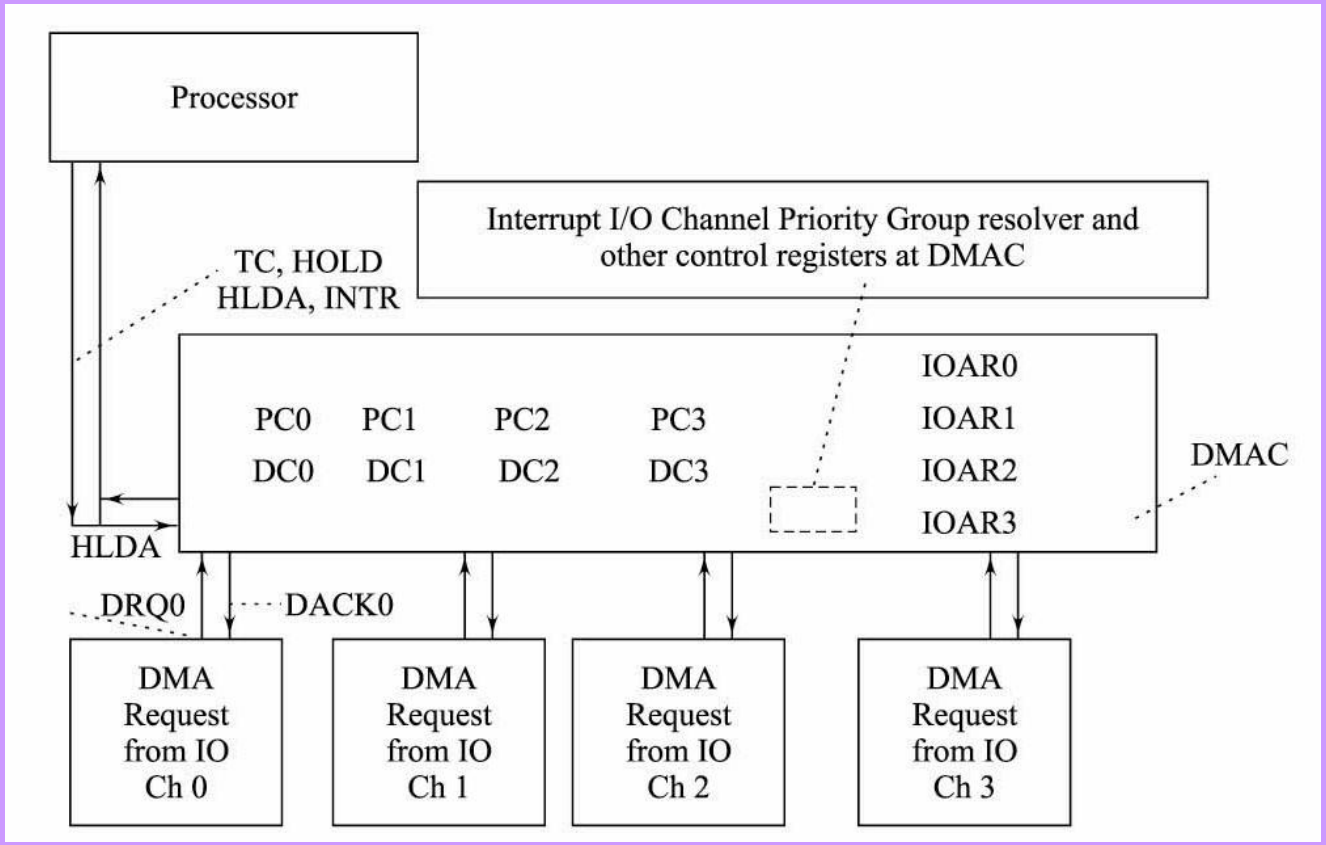


DMA IO Channels

Multichannel DMAC for use with several devices

- Each channel has separate PC, IOAR, and DC
- Every channel has a separate pair of DMA request DRQ and DMA ACK (DACK)
- One I/O channel may be allocated for DRAM refresh controller, another for hard disk 1, third for hard disk 2 and a fourth for the floppy drive

Four I/O Channel DMAC, Separate Control Register(s) and I/O Channel Priority Resolver



Summary

We learnt

- DMA used when multiple bytes are to be transferred between memory and IO devices
- Data transfer occurs efficiently between I/O devices and system memory with the least processor intervention using DMAC
- DMAC has registers for command (for mode of transfer— bulk or burst or bytes), data-count (number of bytes to be transferred), memory block address where access to data is made

We learnt

- DMAC facilitates fast direct byte transfers between memory and I/O devices compared of interrupt driven DMA as it has in-built processing element and uses the system buses as and when they are made available by the processor

End of Lesson 10 on

Direct Memory Access (DMA)