DEVICE DRIVERS AND INTERRUPTS SERVICE MECHANISM

Lesson-12: 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 (single purpose processor) and that device is called DMAC (DMA Controller).

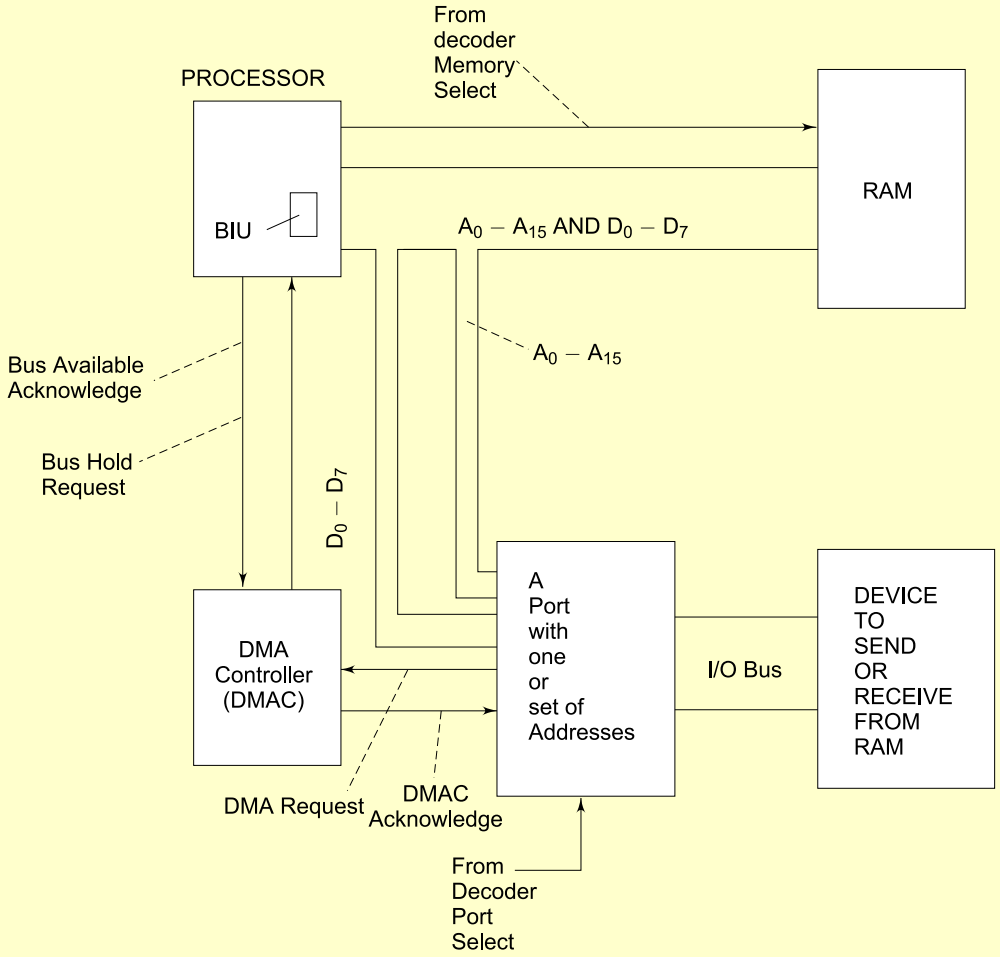
Using a DMA controller

- 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



BIU not accessible by address and
Data Buses during Acknowledge active

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

- 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 is 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

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

Use of DMA Channel for Facilitating the Small Interrupt-Latency Period Sources

Small latency periods can be set when using a DMA channel when multiple interrupt from IO sources exist.
The ISR run period from start to end can now be very small, [only short code for programming DMAC and for short code on end of DMA transfer for initiating new data transfer or new task.]

Multiple channels DMAC

- Provides DMA action from system memories and two (or more IO) devices.
- Separate set of registers for programming each channel.
- Separate interrupt signals in the case of a multi-channel DMAC

On Chip DMAC in Microcontrollers

- 8051 family member 83C152JA (and its sister JB, JC and JD microcontrollers) ─two DMA channels on-chip.
- 80196KC has a PTS (Peripheral Transactions Server) that supports DMA functions. [Only single and bulk transfer modes are supported, not the burst transfer mode.]
- MC68340 microcontroller has two DMA channels on chip.

Summary

We learnt

- DMA controller is a device with single purpose processor and 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

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 12 of Chapter 4