INTERRUPTS II

CPTS 260

How does the processor know which device is interrupting?

Interrupt Vectors

- Each device is assigned a unique number which is recognized by the bus.
- Processor hardware interrupts the number as an index into an array of pointers at a reserved location in memory.
- The interrupt handler reads the item in this array to handle the device, thus we say interrupts are vectored.

Multiple Levels of Interrupts

- How do we handle multiple types of devices?
- Some interrupts require interrupts be serviced in short time and others do not need it now!
- We need them to have Priorities.
- We operating at a priority level K, a processor can only be interrupted by a device that has been assigned to level k+1 or higher.
- A processor can have interrupt priorities level 0 to N and they assign 0 to application programs and can have N interrupts in progress at a time. However only ONE interrupt can be in progress at any priority level.

Assignment of Interrupts Vectors & Priorities

- Fixed: Manual assignment, used on small scale and embedded systems.

  Example: Some devices are manufactured with physical switches on circuit board, and interrupt vector address is entered in these switches.

- Flexible: Automated assignment used on general purpose system.

  Example: When user connect a new device the PC automatically assigns the interrupt vector assignment.

- USB Devices?

Smart Devices & Improved I/O

- Smart device does not interrupt as often and does not require the processor to handle each step.

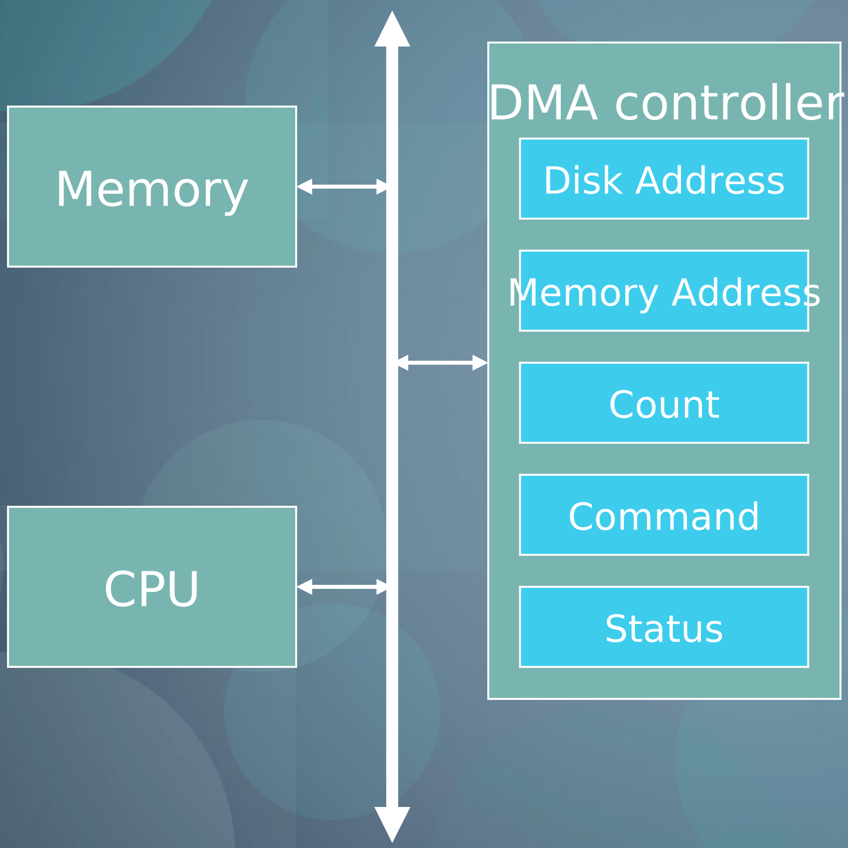
  Example: A disc device.

- The Interrupt processing is complex and when an error occurs the processor must access the CSR and determine which operation is successful and which is not and proceed.

Direct Memory Access (DMA)

- Allows external devices to access memory without processor intervention
- Requires a DMA interface device
- Must be “set up” or programmed and transfer initiated

DMA Interface



CPU:

fetch instruction
decode
fetch operand
operate

DMA Cycle Steal

Example: DMA Transfer from disk

<u>Process</u> (running)
…
`fread(…)`
/* will free CPU? */
//(non-blocking
I/O?)

<u>Device Driver</u>
told to
transfer data
from disk to
memory
buffer

<u>Disk</u>

<u>Controller</u>
Initiates
transfer
Sends bytes to
DMA controller

<u>DMA Controller</u>
Transfers bytes
to memory buffer
When completed,
interrupts the CPU

<u>CPU</u>

(running
other processes)
…
ISR

Interrupt

Benefits of DMA

- DMA: saves CPU time by controlling data transfer between I/O device and memory.
- Advantages of DMA
  - Computer system performance is improved by direct transfer of data between memory and I/O devices, bypassing the CPU.
  - CPU is free to perform operations that do not use system buses.
- Disadvantages of DMA
  - In case of Burst Mode data transfer, the CPU is rendered inactive for relatively long periods of time.

Buffer Chaining

- Can we optimize DMA?   YES!
- Smart I/O devices use buffer chaining (The processor allocated multiple buffers, and creates a linked list in memory, the processor then passes the list to I/O device and allows devices to fill each buffer).
- The technology used to create new operation is called operation chaining. (Addition to buffer chaining it includes a read or write operation with the block number).

SUMMARY

- Programmed I/O and Interrupt driven I/O
- Interrupts and Interrupt Vectors
- Interrupt priorities.
- Benefits of Interrupts driven I/O
- Smart I/O
- DMA
- Buffer chaining