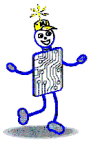


CHAPTER 9: Input / Output

The Architecture of Computer Hardware, Systems Software & Networking:

An Information Technology Approach

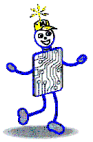
5th Edition, Irv Englander

John Wiley and Sons ©2013

PowerPoint slides authored by Angela Clark, University of South Alabama

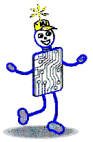
PowerPoint slides for the 4th edition were authored by Wilson Wong, Bentley University

PowerPoint slides modified by Gianluca Amato, Univ. di Chieti-Pescara



Remeber the IPO model

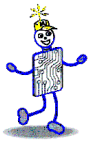
- Processing speed or program execution
  - for many applications, determined primarily by ability of I/O operations to stay ahead of processor.



Examples of I/O Devices

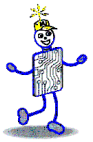
Device	Input/Output	Data Rate*	Control	Type
Keyboard	Input	Very low	External & Program	Character
Mouse	Input	Low	External	Character
Touchpad				
Touch screen				
Scanner	Input	Medium	External & Program**	Block burst
Voice	Input	Low to medium	External & Program	Block burst
Sound	Input/Output	Medium	Program	Block burst or steady
USB	Input/Output	Low to very high	External & Program**	Block burst
Network	Input/Output	High to very high	External & Program**	Block burst
Printer	Output	Low to medium	Program	Block Burst
Graphics display	Output	High	Program	Steady
Flash drive	Storage	Medium	External & Program**	Block burst
Magnetic Disk	Storage	Medium	Program	Block burst
Solid State Drive	Storage	Medium to High	Program	Block burst
Optical Drive	Storage	Medium to High	External & Program**	Block burst or steady
Magnetic tape	Storage	Low to medium	External & Program**	Block burst or steady

* Very low <500 bps; low <5 kbps; medium <10 Mbps; high 10 -100 Mbps; very high 100-5000 Mbps
**External initiation features, mostly program control



I/O Techniques

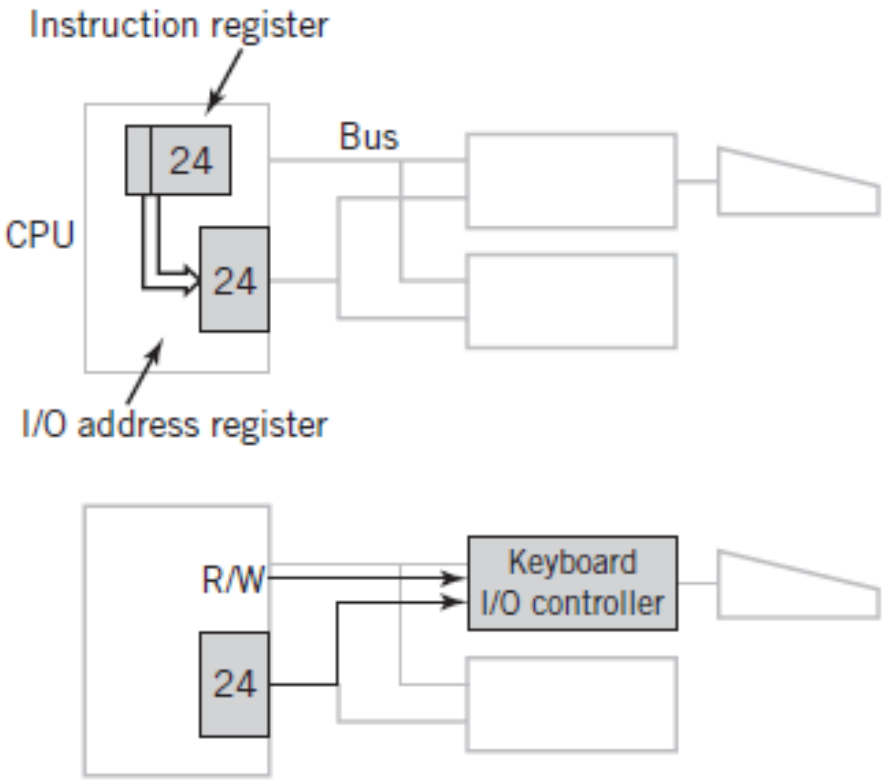
- Programmed I/O
  - CPU controlled I/O
- Interrupt Driven I/O
  - External input controls
- Direct Memory Access Controllers
  - Method for transferring data between main memory and a device that bypasses the CPU



Programmed I/O

- I/O data and address registers in CPU
- One word transfer per I/O instruction
- Address information for each I/O device
  - LMC I/O capability for 100 devices
- Full instruction fetch/execute cycle
- Primary use:
  - keyboards
  - communication with I/O modules (see DMA)

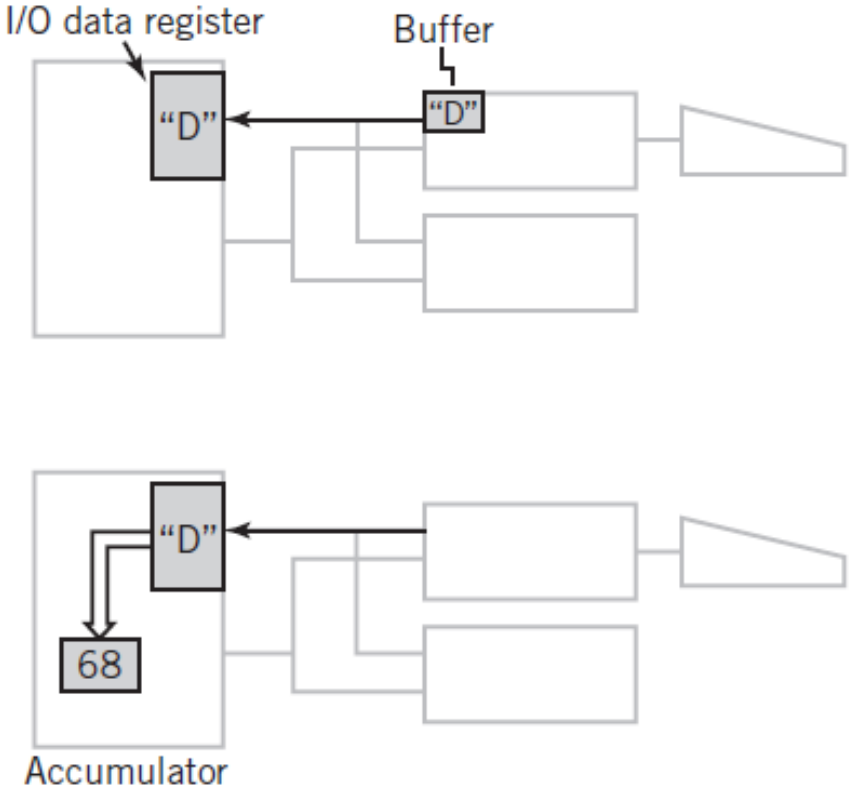
Programmed I/O Example



1. CPU executes INPUT 24 instruction. Address 24 is copied to the I/O address register.

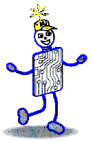
2. Address 24 is recognized by the keyboard I/O controller. A read/write control line indicates that the instruction is an INPUT.

Programmed I/O Example



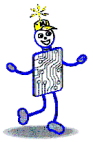
3. A buffer in the I/O controller holds a keystroke, in this case ASCII 68, the letter "D". The data is transferred to the I/O data register.

4. From there it is copied to the appropriate accumulator or general-purpose register, completing the operation.



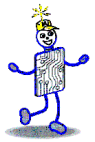
Interrupts

- Signal that causes the CPU to alter its normal flow of instruction execution
  - frees CPU from waiting for events
  - provides control for external I/O initiation
  - Interrupt lines (hardware)
    - One or more special control lines to the CPU
  - Interrupt handlers
    - Program that services the interrupt
    - Also known as an interrupt routine or device driver

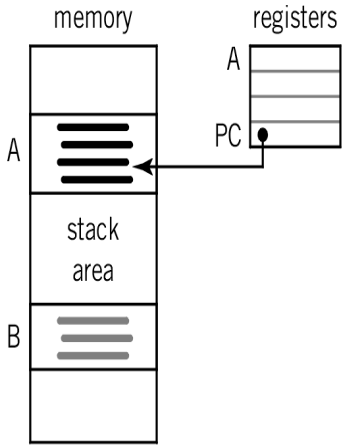


Interrupt Terminology

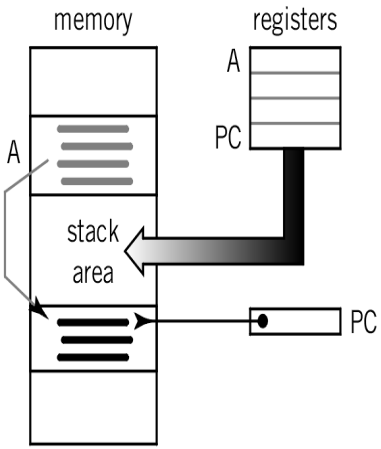
- Interrupt lines (hardware)
  - One or more special control lines to the CPU
- Interrupt request
- Interrupt handlers
  - Program that services the interrupt
  - Also known as an interrupt routine or device driver
- Context
  - Saved registers of a program before control is transferred to the interrupt handler
  - Allows program to resume exactly where it left off when control returns to interrupted program



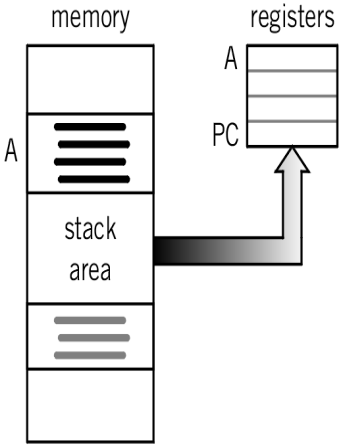
Servicing an Interrupt



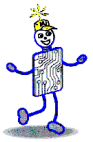
1. Before interrupt arrives, program A is executing. The program counter points to the current instruction.



2. When the interrupt is received by the CPU, the current instruction is completed, all the registers are saved in the stack area (or in a special area known as a process control block). The PC is loaded with the starting location of program B, the interrupt handler program. This causes a jump to program B, which becomes the executing program.

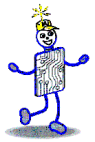


3. When the interrupt routine is complete, the registers are restored, including the program counter, and the original program resumes exactly where it left off.

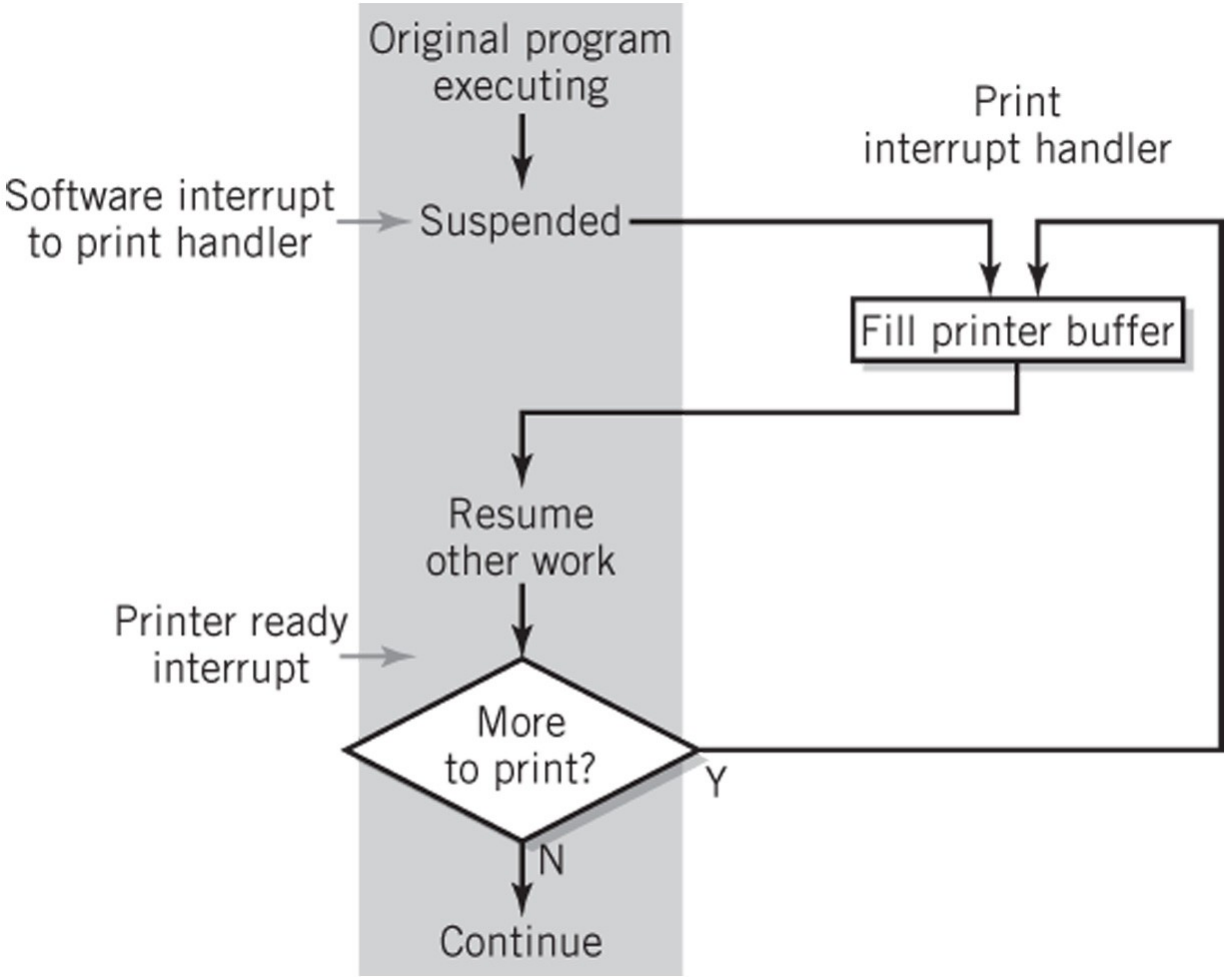


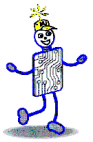
Use of Interrupts

- Notify that an external event has occurred
  - real-time or time-sensitive
- Signal completion
  - printer ready or buffer full
- Allocate CPU time
  - time sharing
- Indicate abnormal event (CPU originates for notification and recovery)
  - illegal operation, hardware error
- Software interrupts

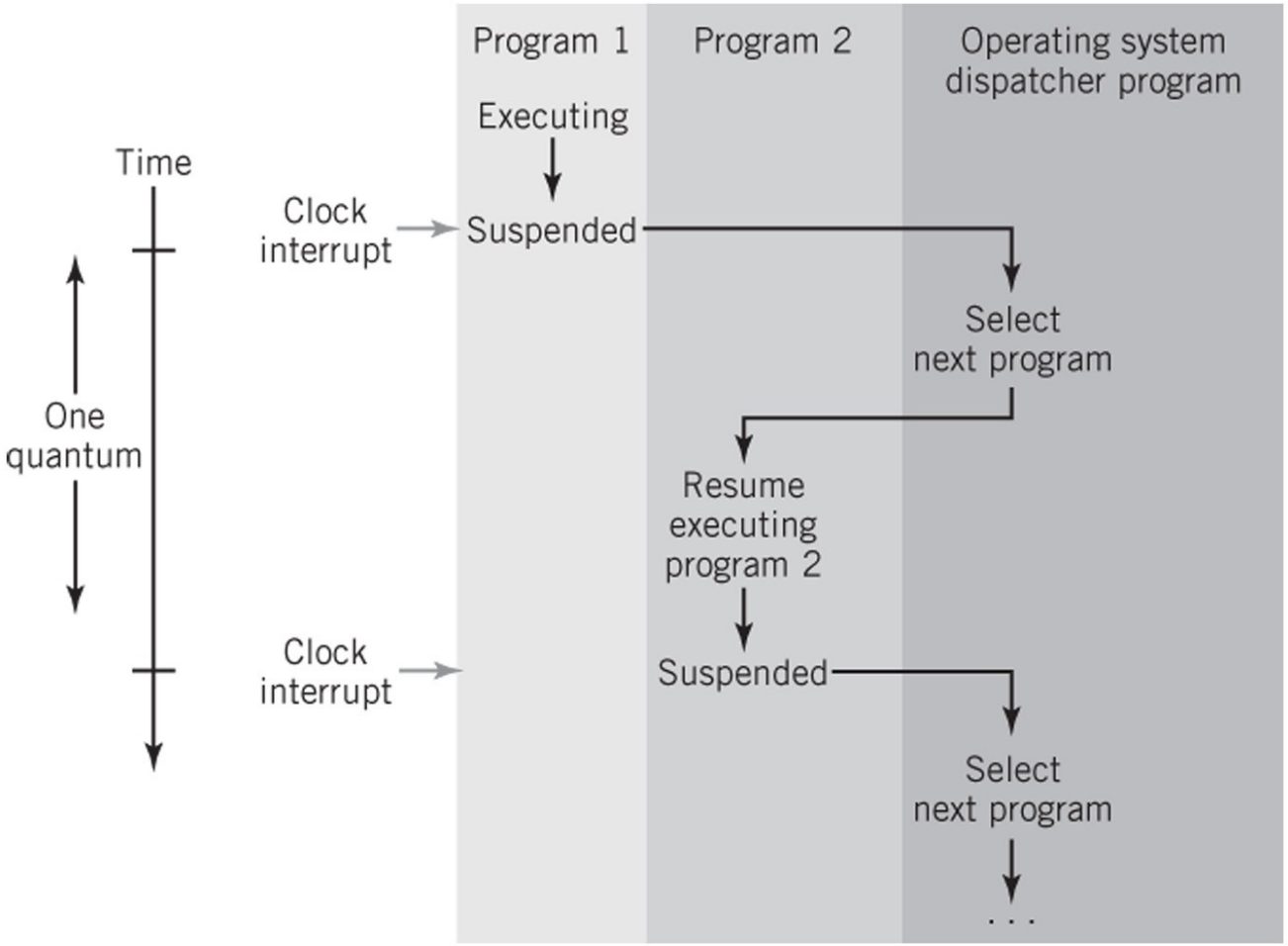


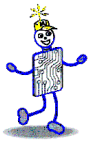
Print Handler Interrupt





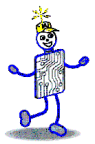
Using an Interrupt for Time Sharing



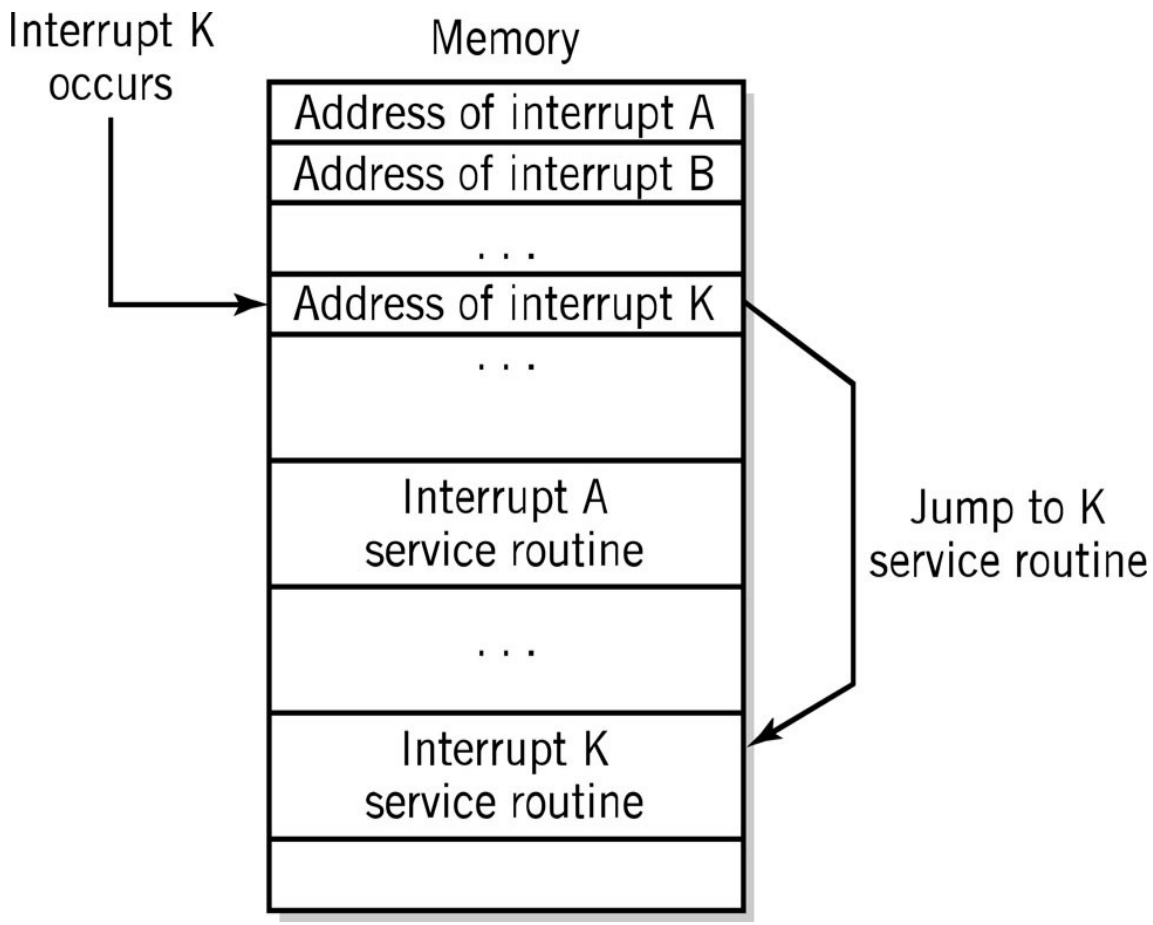


Interrupt Processing Methods

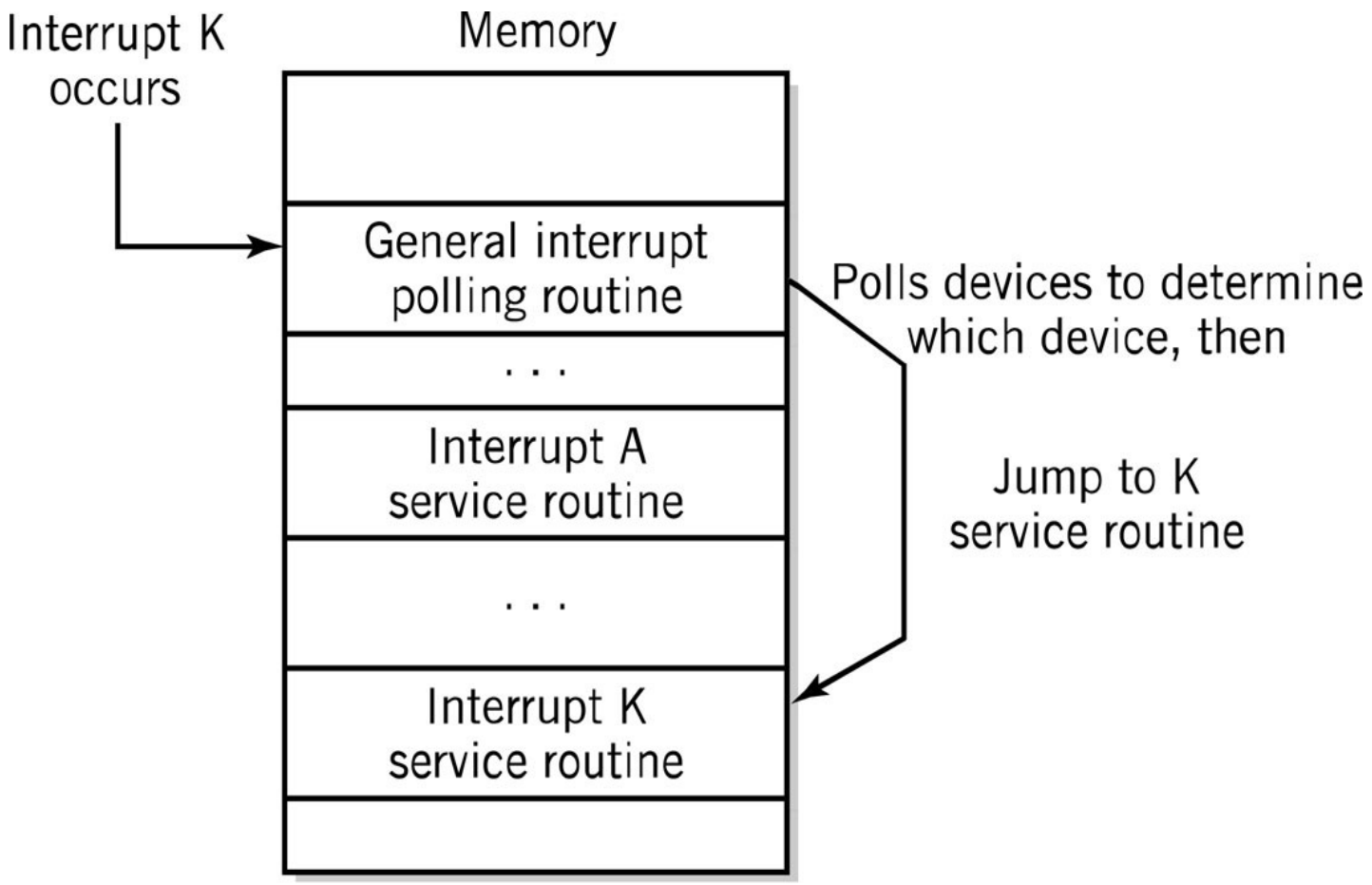
- Vectored interrupt
  - Address of interrupting device is included in the interrupt
  - Requires additional hardware to implement
- Polling
  - Identifies interrupting device by polling each device
  - General interrupt is shared by all devices

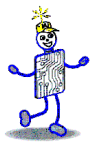


Vectored Interrupts

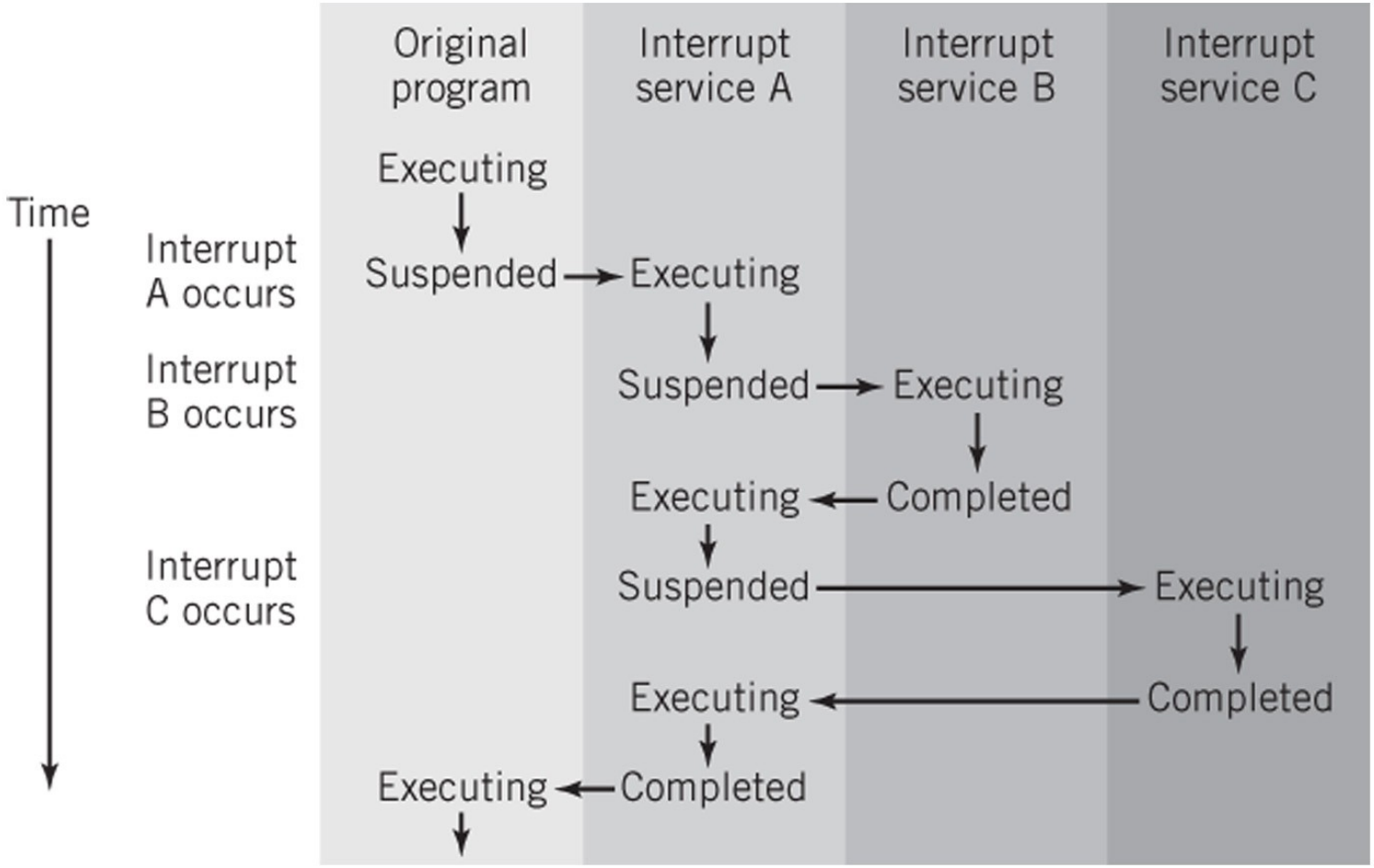


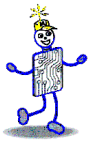
Polled Interrupts





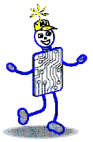
Multiple Interrupts Example





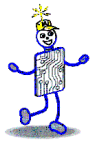
Direct Memory Access

- Transferring large blocks of data
- Direct transfer to and from memory
- CPU not actively involved in transfer itself
- Required conditions for DMA
  - The I/O module must be capable of reading and writing to memory
  - Conflicts between the CPU and the I/O module must be avoided

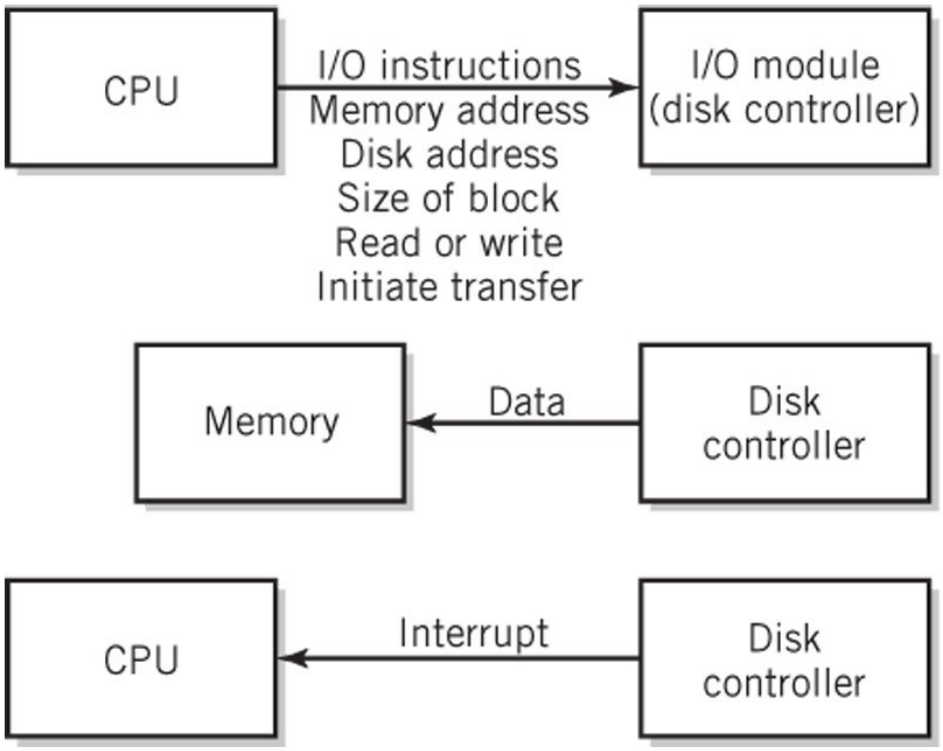


DMA Instructions

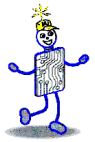
- To initiate DMA, programmed I/O is used to send the following information:
  1. Location of data on I/O device
  2. Starting location in memory
  3. Size of the block
  4. Direction of transfer: read or write
- Interrupt to CPU upon completion of DMA



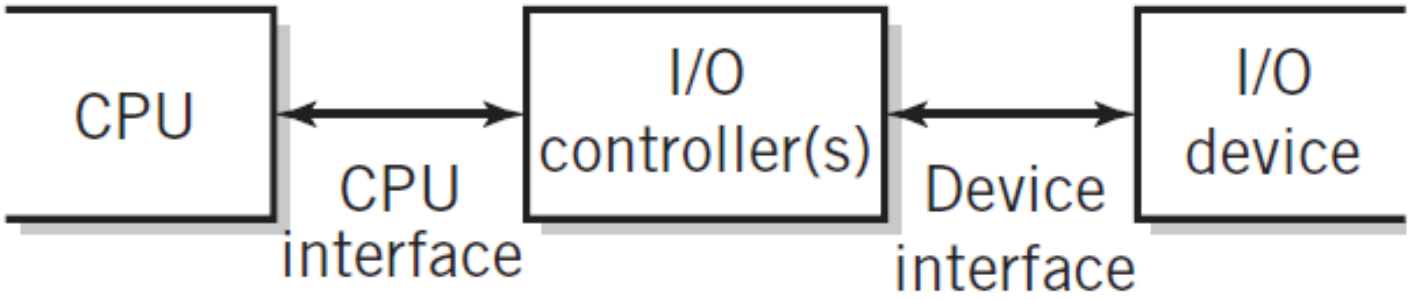
DMA Initiation and control

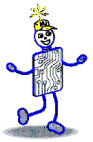


1. Programmed I/O used to prepare I/O module for transfer by providing required information and initiating transfer.
2. DMA transfer. In this case data is transferred from disk to memory.
3. Upon completion, disk controller sends *completion* interrupt to CPU.



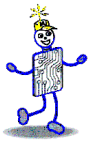
I/O Controller Interfaces





I/O Controller Functions

- Recognizes messages from device(s) addressed to it and accepts commands from the CPU
- Provides a buffer where the data from memory can be held until it can be transferred to the device
- Provides the necessary registers and controls to perform a direct memory transfer
- Physically controls the device
- Copies data from its buffer to the device/from the CPU to its buffer
- Communicates with CPU



Copyright 2013 John Wiley & Sons