Computer Organization and Architecture

17 Input / Output

Tongwei Ren

Dec. 17, 2019

南京大學
NANJING UNIVERSITY

Review

- Micro-operations
  - Concept, different cycles
- Control unit
  - Input and output
  - Function
- Control signal
- Hardwired implementation
- Microprogrammed control unit

What is a Computer in Your Eyes

Peripheral Device

- I/O operations are accomplished through a wide assortment of external devices that provide a means of exchanging data between the external environment and the computer
- An external device connected to an I/O module is often referred to as a **peripheral device** or, simply, a **peripheral**
- Type
  - Human readable: communicate with the computer user
    - Video display terminal, printer, ……
  - Machine readable: communicate with equipment
    - Disk, tape, ……
  - Communication: communicate with remote devices

Can we connect peripherals to system bus directly?

Can we connect peripherals to system bus directly?

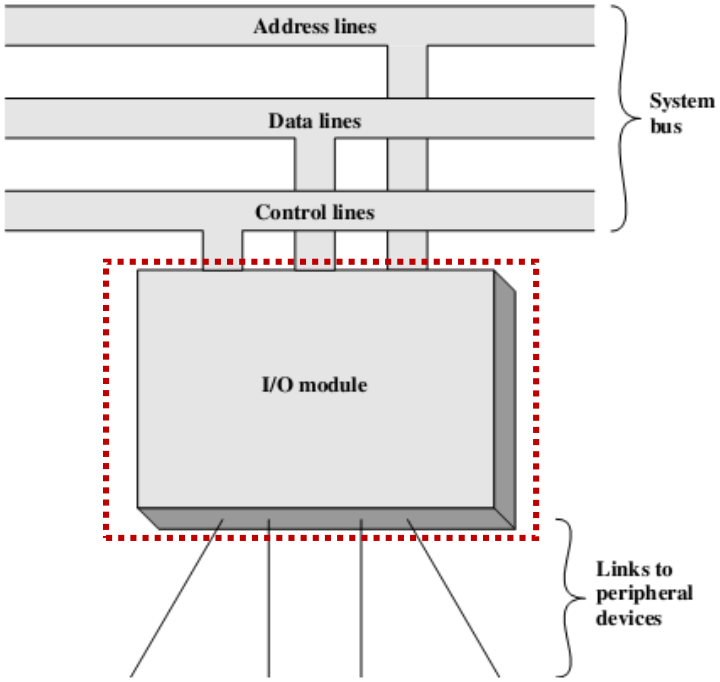
NO

Why NOT Connect Peripherals to Bus

- There are a wide variety of peripherals with various methods of operation
- The data transfer rate of peripherals is often much slower than that of the memory or processor
- The data transfer rate of some peripherals is faster than that of the memory or processor
- Peripherals often use different data formats and word lengths than the computer to which they are attached

I/O Module

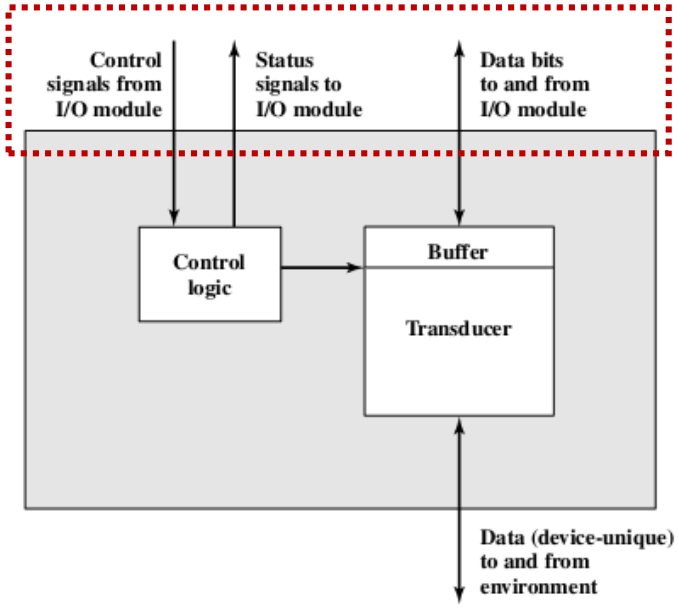
- Interface to the processor and memory via the system bus or central switch
- Interface to one or more peripheral devices by tailored data links



I/O Module is a bridge between **internal system** and **peripherals**

Interface to Peripheral

- The interface to the I/O module is in the form of control, data, and status signals
- Control logic associated with the device controls the device's operation in response to direction from the I/O module
- Buffer is associated with the transducer to temporarily hold data being transferred between the I/O module and the external environment with size of 8 to 16 bits



I/O Module Functions

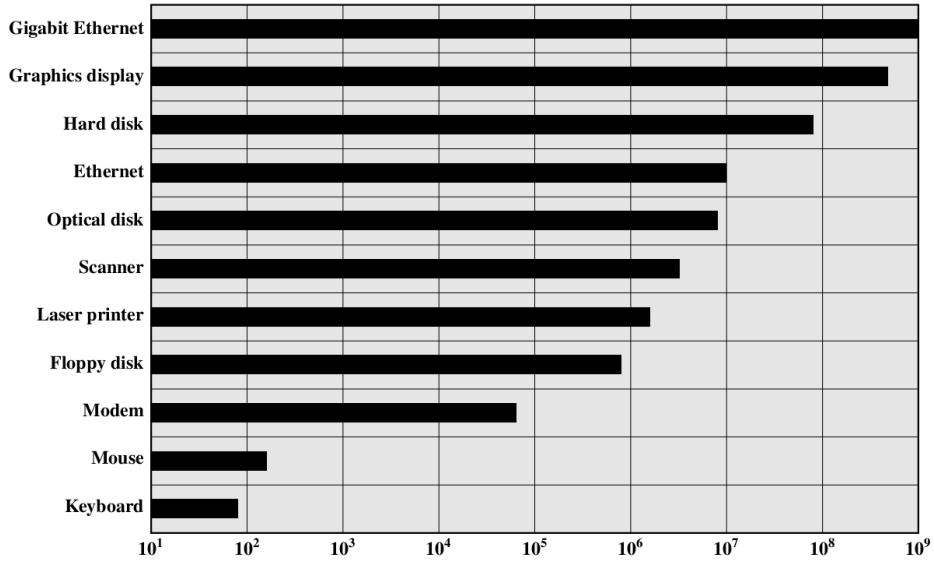
- Processor communication
  - Command decoding: I/O module accepts commands from the processor, typically sent as signals on the control bus
  - Data: data are exchanged between processor and I/O module over data bus
  - Status reporting: for peripherals are so slow, it is important to know the status of the I/O module
  - Address recognition: I/O module must recognize one unique address for each peripheral it controls
- Device communication
  - Involve commands, status information, and data

I/O Module Functions

- Control and timing
  - Processor communicates with external devices in unpredictable patterns
  - Internal resources, such as memory and system bus, are shared
  - Example: transfer of data from external device to processor
    - Processor interrogates I/O module to check the device status
    - I/O module returns the device status
    - If prepared, processor requests the transfer of data, by means of a command to I/O module
    - I/O module obtains a unit of data from the external device
    - Data are transferred from I/O module to processor

I/O Module Functions

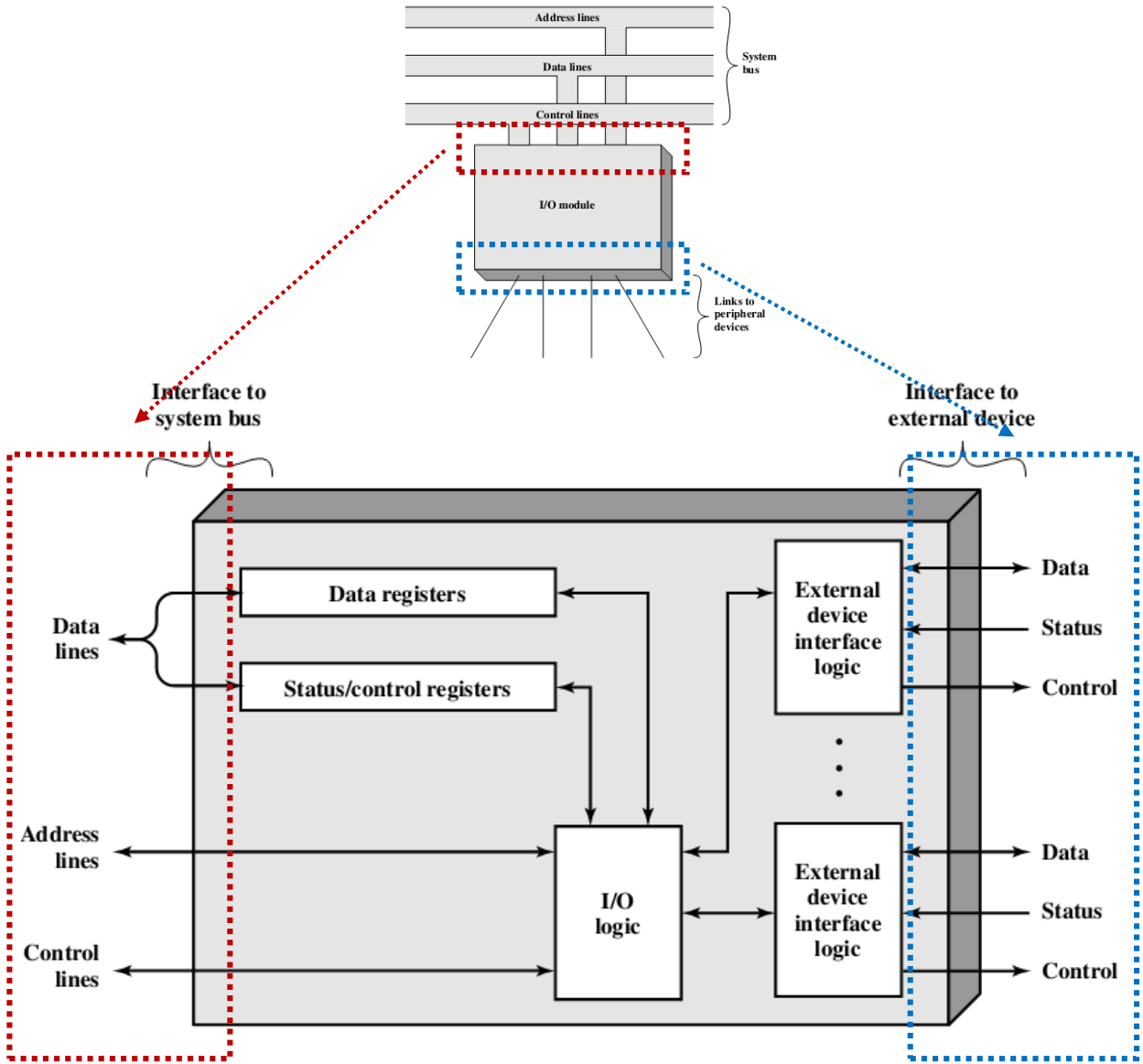
- Data buffering
  - Transfer rate into and out of main memory or processor is quite high
  - Transfer rate is orders of magnitude lower for many peripheral devices and covers a wide range



I/O Module Functions

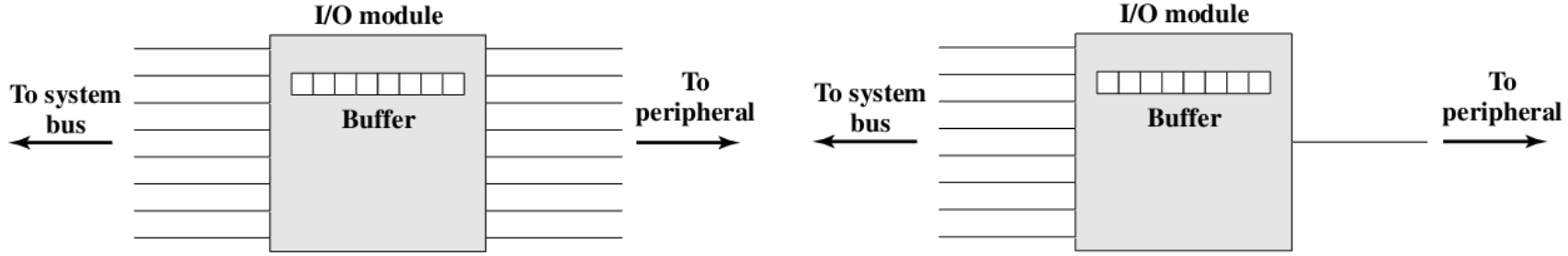
- Error detection
  - Detect errors and report errors to processor
  - Error type
    - Mechanical and electrical malfunctions reported by the device
    - Unintentional changes to the bit pattern in transfer

I/O Module Structure



External Interface

- Type
  - Parallel interface: there are multiple lines connecting I/O module and the peripheral, and multiple bits are transferred simultaneously, just as all of the bits of a word are transferred simultaneously over the data bus
  - Serial interface: there is only one line used to transmit data, and bits must be transmitted one at a time
- For parallel interface requires timing on each time, the bus clock frequency is limited when the transfer speed and bus length increase



FireWire and USB

- FireWire: IEEE 1394
- USB: Universal Serial Bus

	USB				FireWire		
	1.1	2.0	3.0	3.1	400	800	3200
Data transfer rate	12Mbps	480Mbps	5Gbps	10Gbps	400Mbps	800Mbps	3.2Gbps
Device number	127				63		
Type	Serial						
Architecture	Master/slave				Peer-to-peer		

[刘洪禹，171840773]

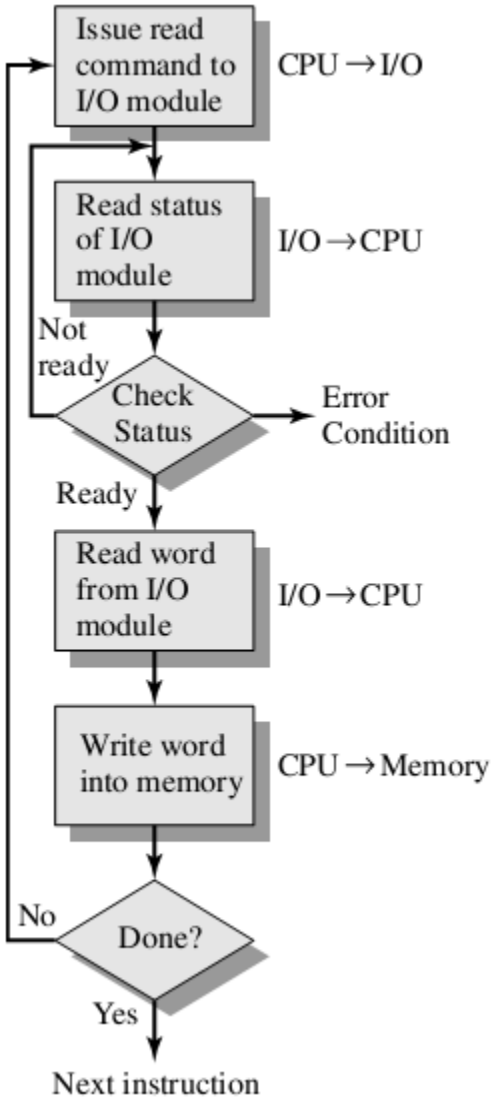
I/O Operation Technique

- Programmed I/O: when transfer data, processor executes a program that gives it direct control of the I/O operation, and wait until the I/O operation is complete
- Interrupt-driven I/O: processor issues an I/O command, continues to execute other instructions, and is interrupted by the I/O module when the latter has completed its work
- Direct memory access(DMA): I/O module and main memory exchange data directly, without processor involvement

	No Interrupts	**Use of Interrupts**
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)

Programmed I/O

- When processor is executing a program and encounters an instruction relating to I/O, it executes that instruction by issuing a command to the appropriate I/O module
- I/O module will perform the requested action and then set the appropriate bits in the I/O status register
- I/O module does not interrupt processor, so it is the responsibility of processor periodically to check the status of I/O module until it finds that the operation is complete



Programmed I/O: I/O Command

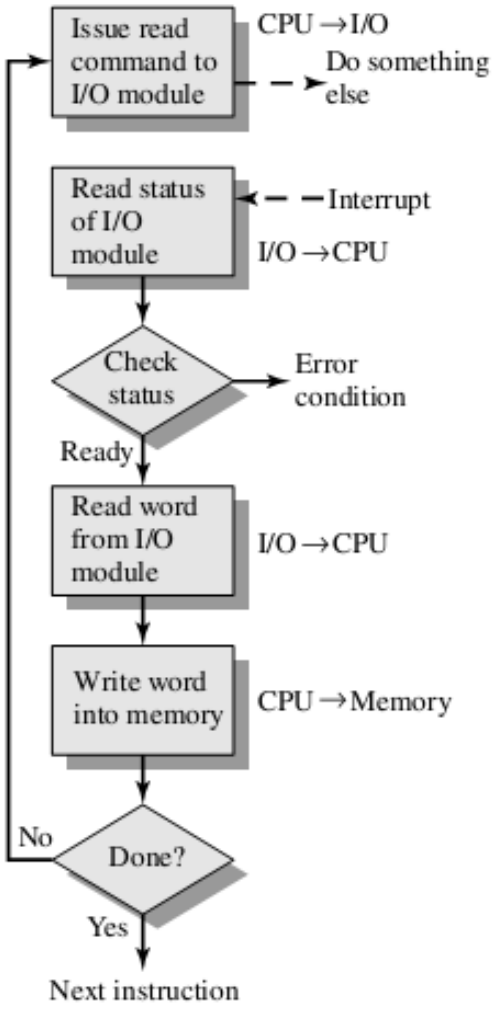
- To execute an I/O-related instruction, the processor issues an address, specifying the particular I/O module and external device, and an **I/O command**
- Type
  - Control: activate a peripheral and tell it what to do
  - Test: test various status conditions associated with an I/O module and its peripherals
  - Read: cause I/O module to obtain an item of data from the peripheral and place it in an internal buffer
  - Write: cause the I/O module to take an item of data from the data bus and subsequently transmit that data item to the peripheral

Programmed I/O: I/O Instruction

- I/O instructions are easily mapped into I/O commands, and there is often a simple one-to-one relationship
  - The form of the instruction depends on the way in which external devices are addressed
- Addressing
  - Memory-mapped I/O: a single address space for memory locations and I/O devices
    - Large repertoire of instructions can be used, allowing more efficient programming
  - Isolated I/O: equipped with memory read and write plus input and output command lines

Interrupt Driven I/O

- Processor issues an I/O command to a module and then go on to do some other useful work
- I/O module interrupts the processor to request service, when it is ready to exchange data with the processor
- Processor executes data transfer, and then resumes its former processing



Interrupt Driven I/O (cont.)

- View point of I/O module
  - For input, I/O module receives a READ command from the processor
  - I/O module proceeds to read data in from an associated peripheral
  - Once the data are in the module's data register, I/O module signals an interrupt to the processor over a control line
  - I/O module waits until its data are requested by the processor
  - When the request is made, I/O module places its data on the data bus and is then ready for another I/O operation

Interrupt Driven I/O (cont.)

- View point of processor
  - Processor issues a READ command
  - Processor goes off and does something else, and checks for interrupts at the end of each instruction cycle
  - When the interrupt from I/O module occurs, processor saves the context of the current program
  - Processor reads the word of data from I/O module and stores it in memory
  - Processor restores the context of the program it was working on (or some other program) and resumes execution

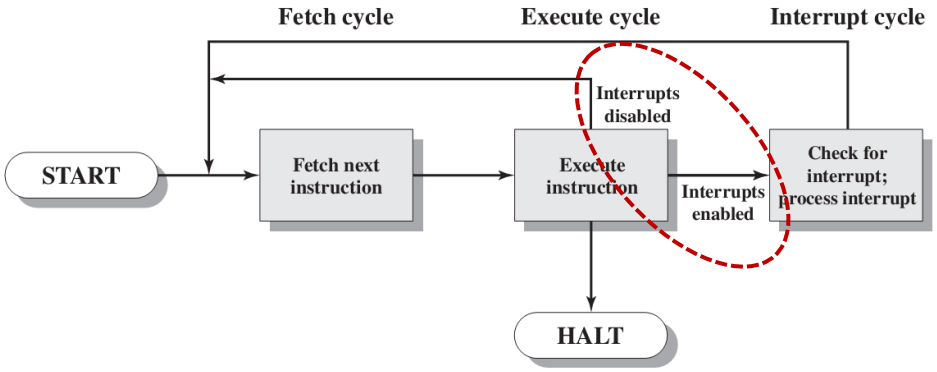
Interrupt Driven I/O: Device Identification

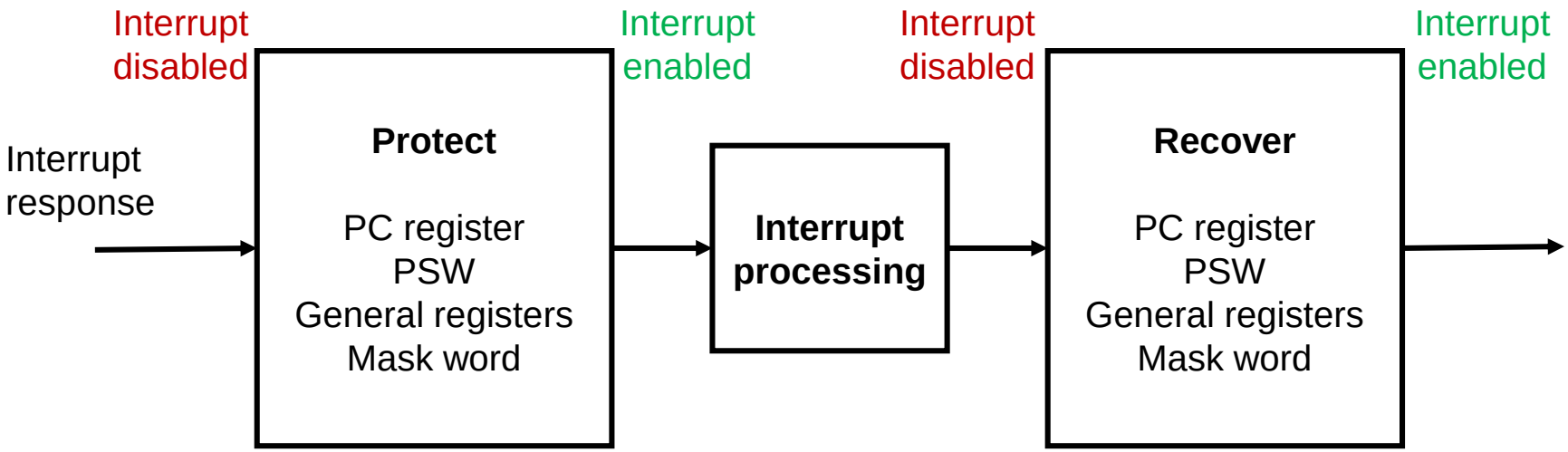
- Multiple interrupt lines
  - Even if multiple lines are used, one of the other three techniques must be used on each line
- Software poll
  - Processor checks the address of each I/O module in turn
- Daisy chain (vectored interrupt)
  - All I/O modules share a common interrupt request line, and the interrupt acknowledge line is daisy chained through the modules
- Independent request (vectored interrupt)
  - A specific interrupt controller is used to analyze priorities and decoding

Interrupt Driven I/O: Assigning Priority

- Multiple interrupt lines: processor picks the interrupt line with the highest priority
- Software poll: the order in which modules are polled determines their priority
- Daisy chain: the order of modules determines their priority
- Independent request : the interrupt controller determines

Interrupt Driven I/O: Interrupts Enabled and Disabled





Interrupt Driven I/O: Response Priority and Processing Priority

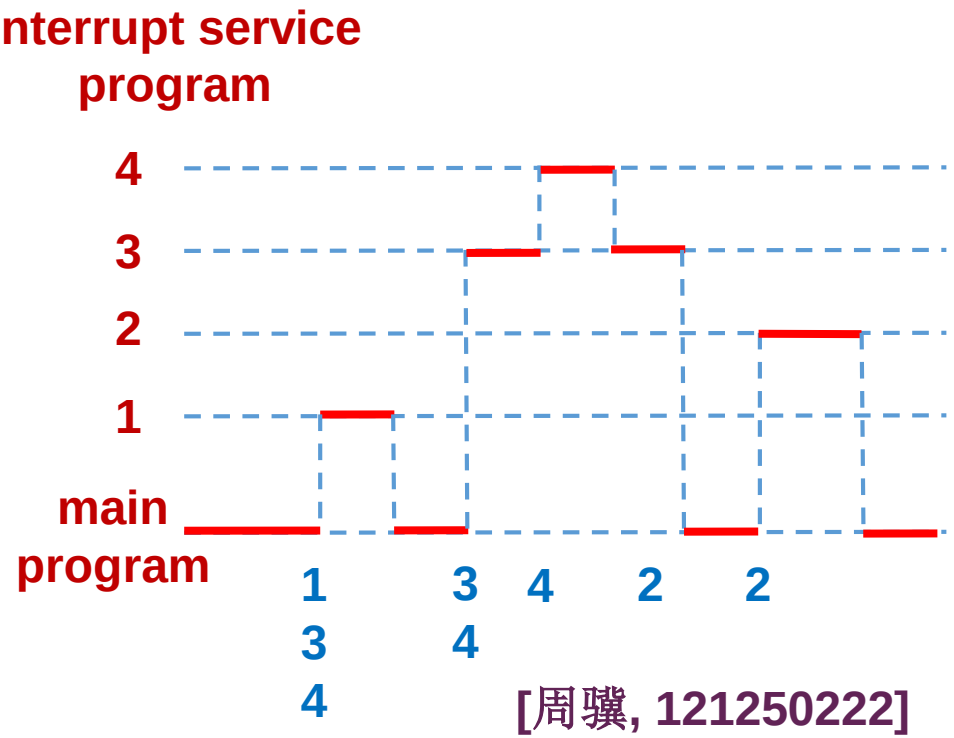
- Example
  - Assume there are 4 interrupt sources in an interrupt system, whose **response priorities** are L1>L2>L3>L4 and **processing priorities** are L1>L4>L3>L2. If the L1, L3 and L4 interrupts occur at the same time when the main program executes, and L2 interrupt occurs in the procedure of processing L3 interrupt, write the **mask words** and the procedure of all interrupt service programs.

Interrupt Driven I/O (cont.)

- Example (cont.)
  - Mask words

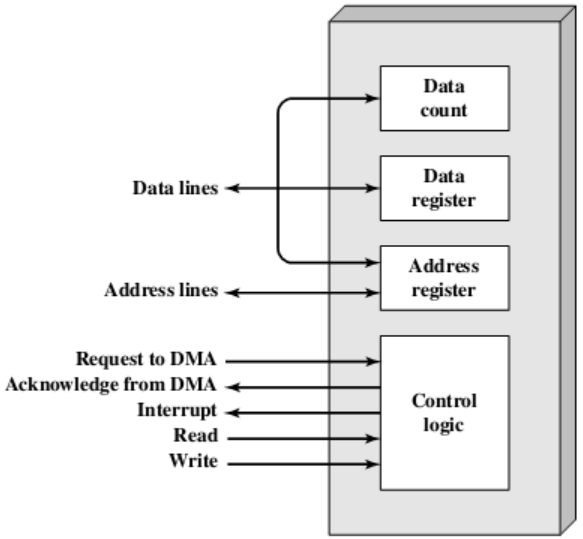
	mask word			
	L1	L2	L3	L4
L1	1	1	1	1
L2	0	1	0	0
L3	0	1	1	0
L4	0	1	1	1

  - Interrupt service program



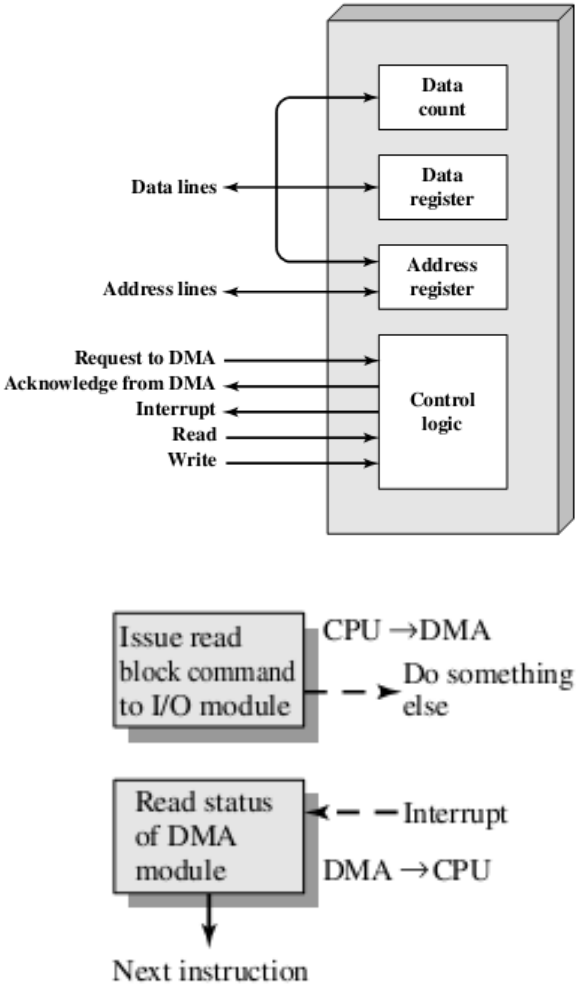
Direct Memory Access

- Problem
  - I/O transfer rate is limited by the speed with which processor can test and service a device
  - Processor is tied up in managing an I/O transfer; a number of instructions must be executed for each I/O transfer
- DMA
  - A module for direct memory access without going through processor



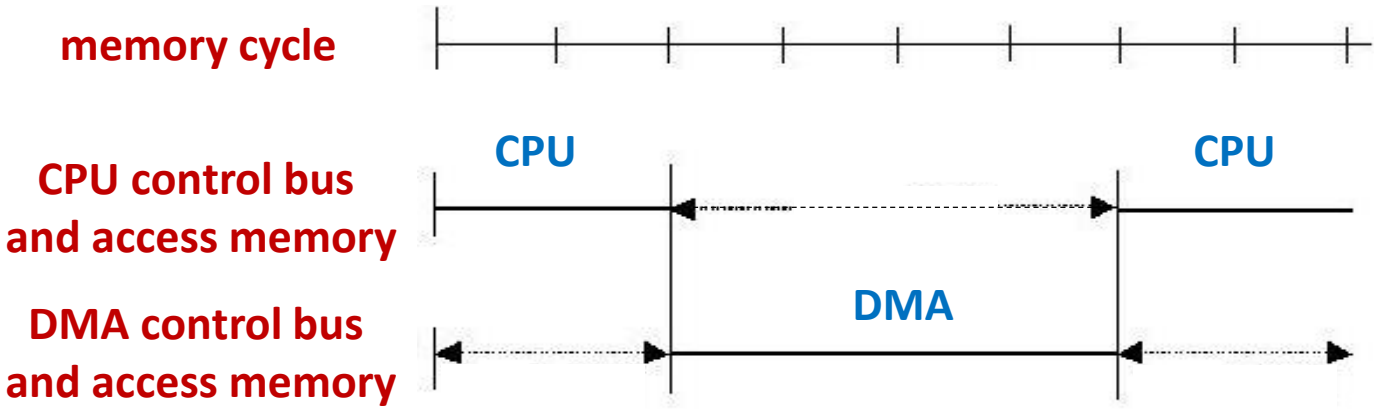
Direct Memory Access (cont.)

- Processor issues a command to the DMA module by sending the following information: read/write, I/O device address, starting location in memory, number of words
- Processor continues with other work
- DMA module transfers the entire block of data, one word at a time, directly to or from memory, without going through processor
- When the transfer is complete, DMA module sends an interrupt signal to processor



Direct Memory Access: Memory Access

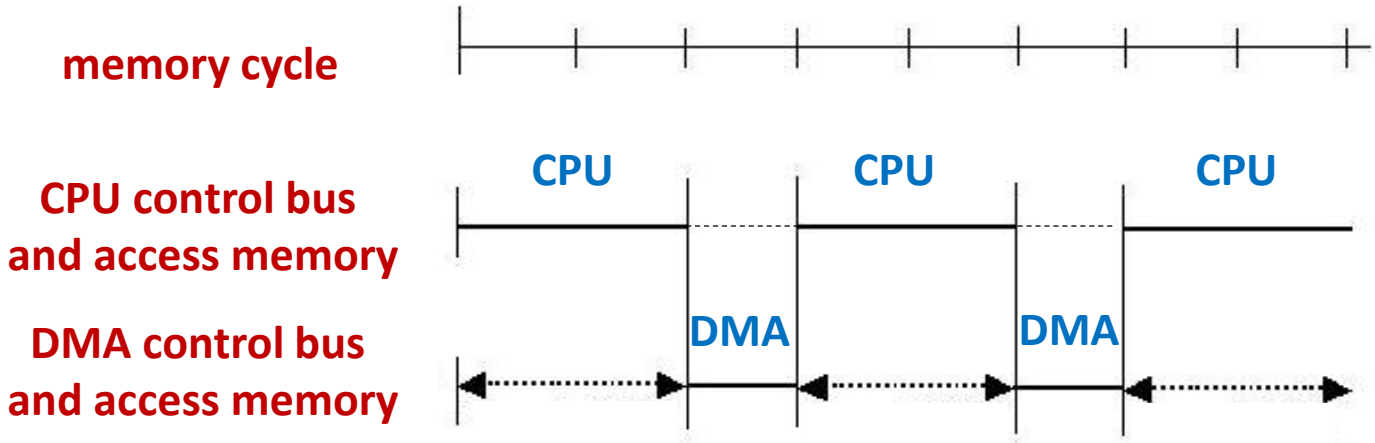
- Stop CPU



- Advantage: simple in control
- Disadvantage: influence CPU, not make full use of memory
- Suitable: block transfer by high speed I/O device

Direct Memory Access: Memory Access

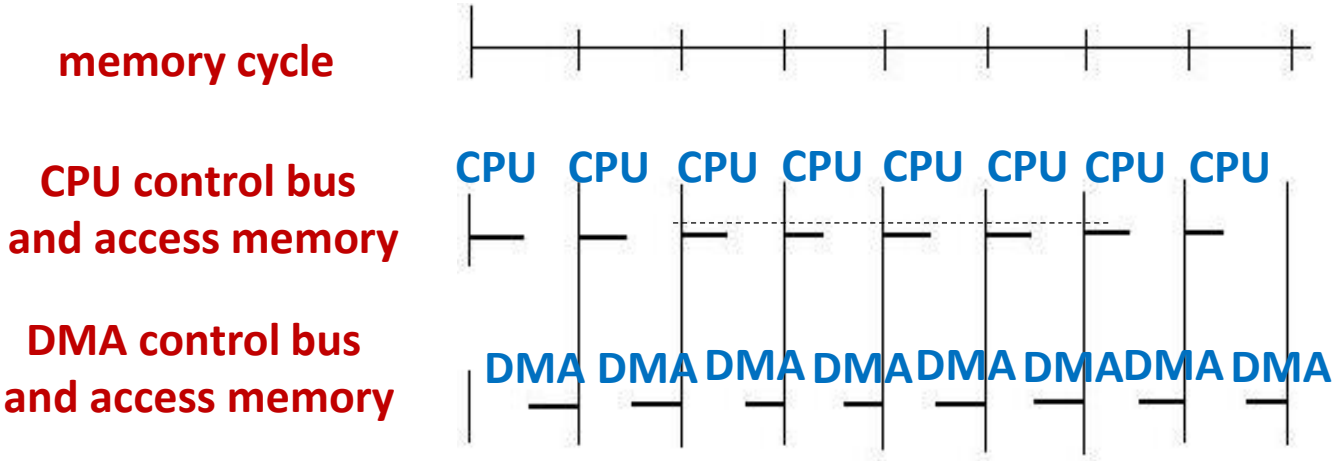
- Cycle stealing



  - Advantage: make full use of CPU and memory, and response I/O request in time
  - Disadvantage: DMA should request bus each time
  - Suitable: I/O cycle is larger than memory cycle

Direct Memory Access: Memory Access

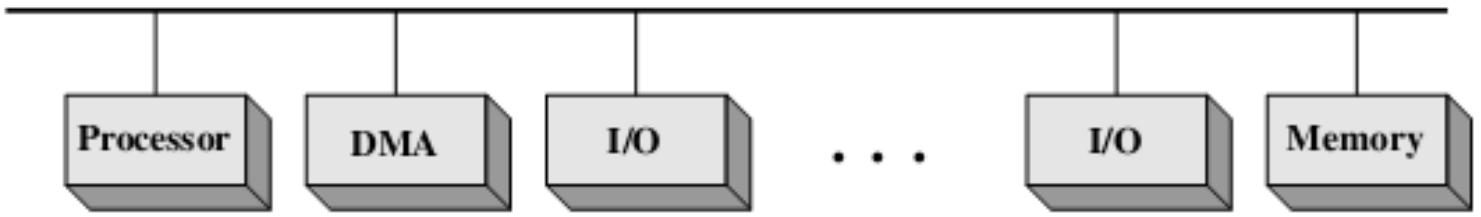
- Alternate (time sharing) access



  - Advantage: CPU is not stopped or waited, and DMA doesn't request bus
  - Suitable: CPU cycle is larger than memory cycle

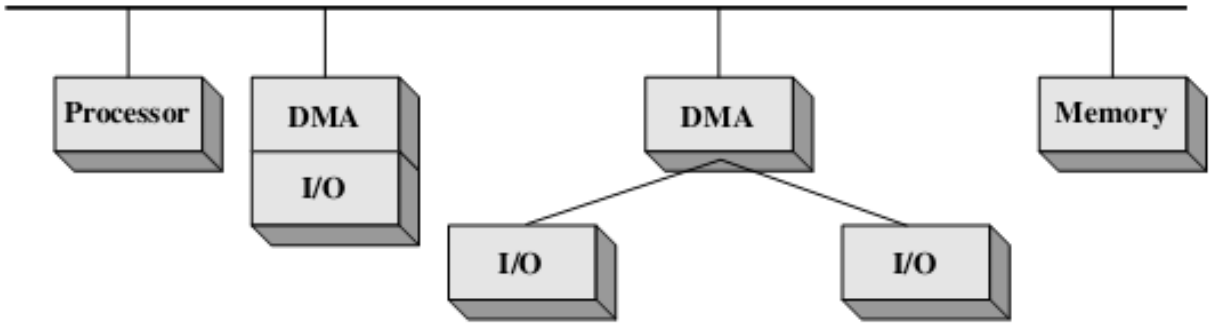
Direct Memory Access: Connection

- DMA mechanism configuration
  - Single-bus, detached DMA
    - All modules share the same system bus
    - DMA module uses programmed I/O to exchange data between memory and an I/O module through DMA module
    - Inexpensive but inefficient



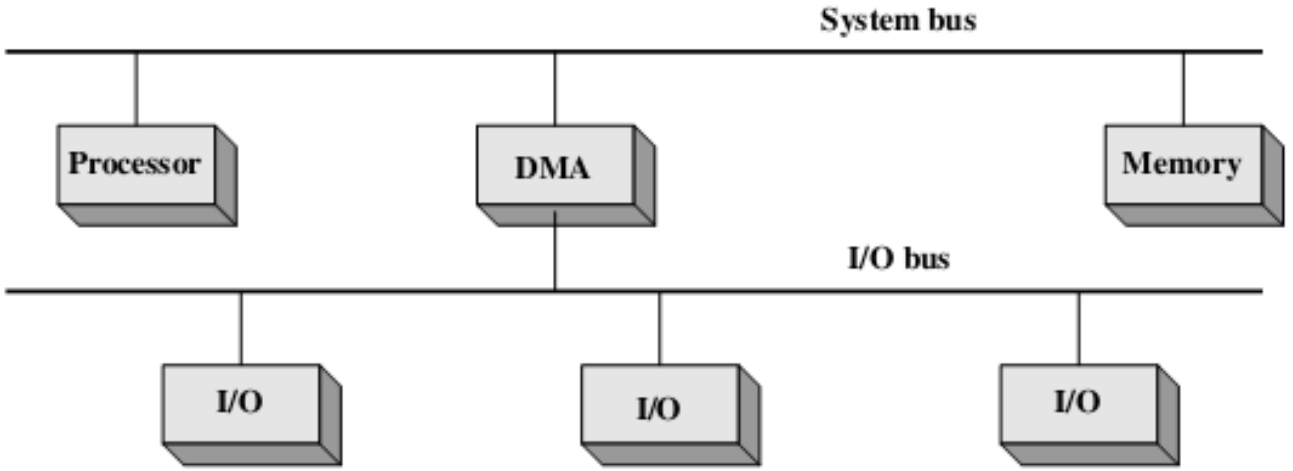
Direct Memory Access : Connection

- DMA mechanism configuration (cont.)
  - Single-bus, integrated DMA-I/O
    - Besides system bus, there is a path between DMA module and one or more I/O modules
    - DMA logic may actually be a part of an I/O module, or it may be a separate module that controls one or more I/O modules
    - Reduce the bus cycles



Direct Memory Access: Connection

- DMA mechanism configuration (cont.)
  - I/O bus
    - Connect I/O modules to the DMA module using an I/O bus
    - Reduce the number of I/O interfaces in DMA module to one and provides for an easily expandable configuration



Direct Memory Access (cont.)

- Example
  - Assume the clock frequency of CPU is 500MHz; hard disk transfers data with 4 words block (32 bits per words), and the transfer speed is 32Mbps
  - If use interrupt driven I/O, each block transfer requires 500 clock cycles (including interrupt response and processing), what the time percentage does CPU use in hard disk I/O?
  - If use DMA, CPU requires 1000 clock cycles for DMA transfer initialization, and 500 clock cycles for interrupt processing after transfer. If 8KB data can be transferred in each DMA transfer, what the time percentage does CPU use in hard disk I/O?

Direct Memory Access (cont.)

- Example (cont.)
  - Interrupt driven I/O

    Interrupt number per second: 32Mb / 16B = 250K

    Cycle number per second: 250K * 500 = 125M

    CPU time percentage: 125M / 500M = 25%
  - DMA

    Time for each DMA transfer: 8KB / 32Mbps = 2ms

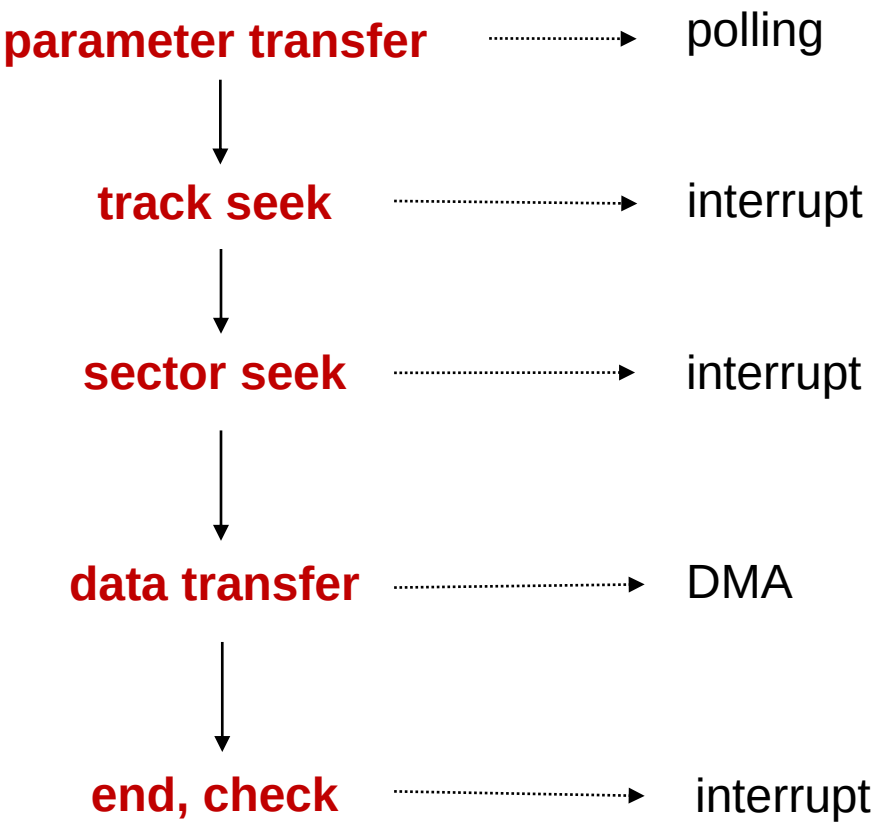
    DMA transfer number per second: 1s / 2ms = 500

    Cycle number per second: (1000 + 500) * 500 = 750K

    CPU time percentage: 750K / 500M = 0.15%

Direct Memory Access: Hard Disk Access

- Full procedure



Evolutionary of I/O Module

- CPU directly controls a peripheral device
- A controller or I/O module is added, and CPU uses programmed I/O without interrupts (divorce CPU from external device details)
- Interrupts are employed (CPU need not spend time waiting for an I/O operation to be performed)
- I/O module is given direct access to memory via DMA (move a block of data to or from memory without involving the CPU, except at the beginning and end of the transfer)

Evolutionary of I/O Module (cont.)

- **I/O channel**: I/O module has its own processor, with a specialized instruction set tailored for I/O (CPU directs I/O processor to execute an I/O program in memory, and it is interrupted only when the entire sequence has been performed)
- **I/O processor**: I/O module has a local memory (a large set of I/O devices can be controlled, with minimal CPU involvement)

Summary

- Peripheral Device
- I/O module
  - Function
  - Structure
- I/O operation technique
  - Programmed I/O
  - Interrupt driven I/O
  - DMA
- Evolutionary of I/O module

Thank You

rentw@nju.edu.cn

南京大學
NANJING UNIVERSITY