Input-Output Organization

MCA-05/MSCIT-05/PGDCA-05/BCA-11

Input-Output Organization

11-1 Peripheral Devices

I/O Subsystem

Provides an efficient mode of communication between the central system and the outside environment

Peripheral (or I/O Device)

Input or Output devices attached to the computer

Monitor (*Visual Output Device*) : CRT, LCD

KBD (*Input Device*) : light pen, mouse, touch screen, joy stick, digitizer

Printer (Hard Copy Device) : Dot matrix (*impact*), thermal, ink jet, laser (*non-impact*)

Storage Device : Magnetic tape, magnetic disk, optical disk

11-2 Input-Output Interface

Interface

1) A conversion of signal values may be required

2) A synchronization mechanism may be needed

The data transfer rate of peripherals is usually slower than the transfer rate of the CPU

3) Data codes and formats in peripherals differ from the word format in the CPU and Memory

4) The operating modes of peripherals are different from each other

Each peripherals must be controlled so as not to disturb the operation of other peripherals connected to the CPU

2) A synchronization mechanism may be needed

The data transfer rate of peripherals is usually slower than the transfer rate of the CPU

3) Data codes and formats in peripherals differ from the word format in the CPU and Memory

4) The operating modes of peripherals are different from each other

Each peripherals must be controlled so as not to disturb the operation of other peripherals connected to the CPU

Interface

Special hardware components between the CPU and peripherals

Supervise and Synchronize all input and output transfers

I/O Bus and Interface Modules :

I/O Bus

Data lines

Address lines

Control lines

Interface Modules : *VLSI Chip*

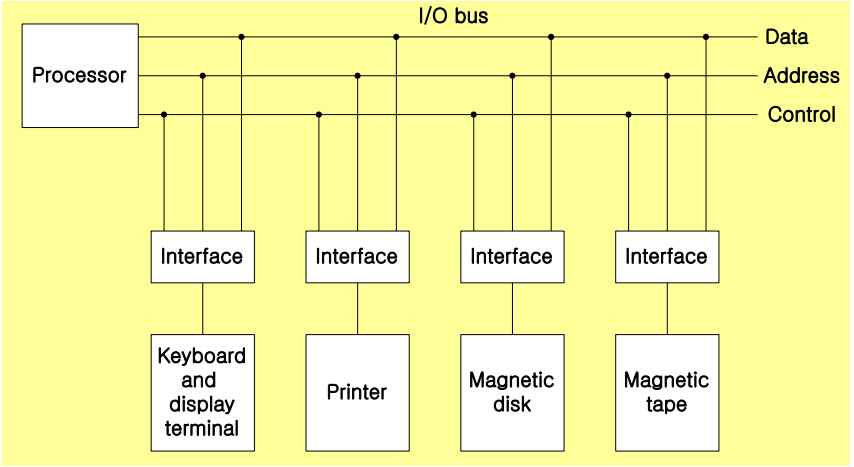
SCSI (Small Computer System Interface)

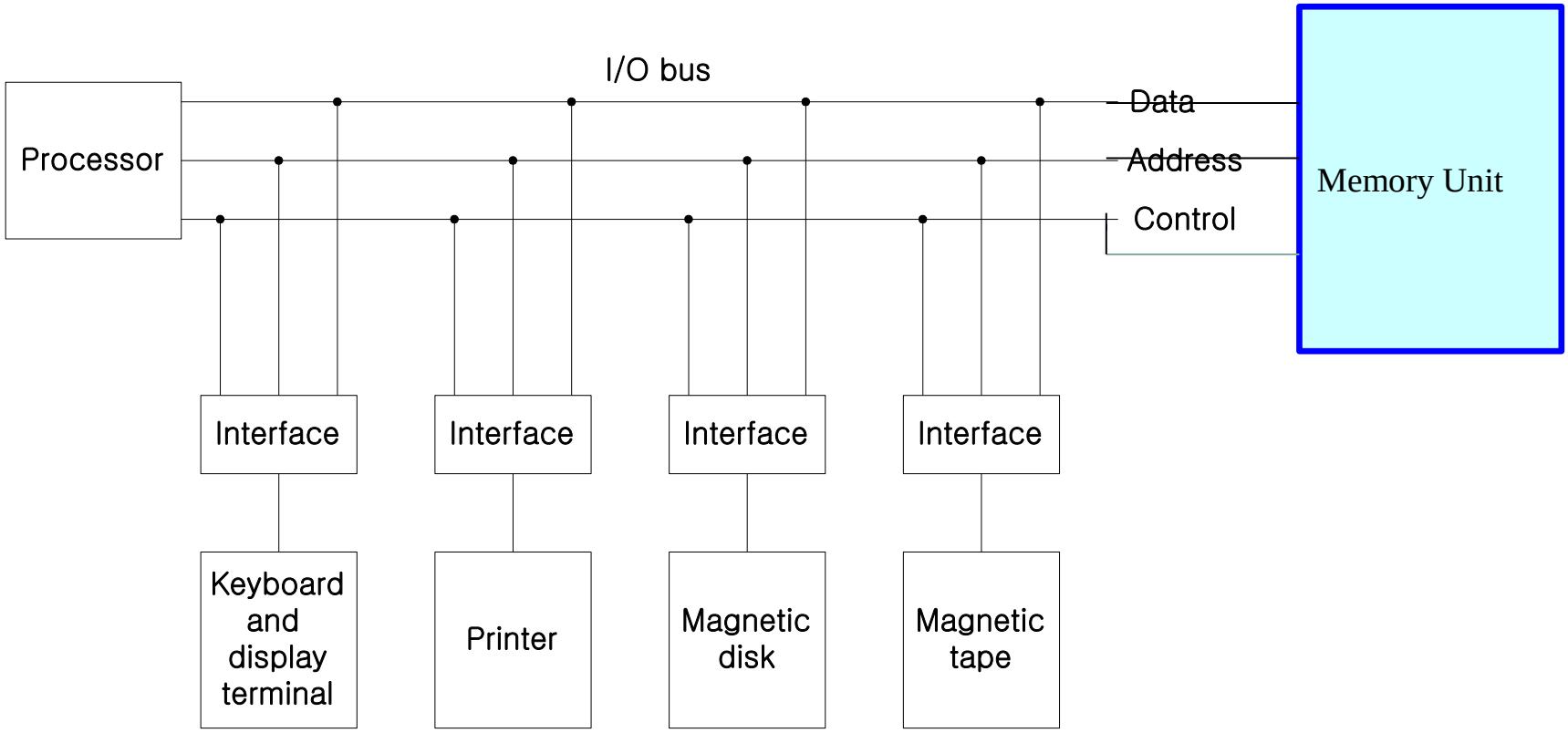
IDE (Integrated Device Electronics)

Centronics

RS-232

IEEE-488 (GPIB)




I/O bus
Processor
Data
Address
Control
Memory Unit
Interface
Interface
Interface
Interface
Keyboard and display terminal
Printer
Magnetic disk
Magnetic tape

I/O command :

1. Control Command
2. Status Command
3. Input Command
4. Output Command

I/O Bus versus Memory Bus

Computer buses can be used to communicate with memory and I/O

1) Use two separate buses, one for memory and the other for I/O

I/O Processor

2) Use one common bus for both memory and I/O but have separate control lines for each : *Isolated I/O* or *I/O Mapped I/O*

Example of I/O Interface :

4 I/O port : Data port A, Data port B, Control, Status

8255 PIO (port A, B, C, Control/Status)

Address Decode : CS, RS1, RS0

11-3 Asynchronous Data Transfer

Synchronous Data Transfer

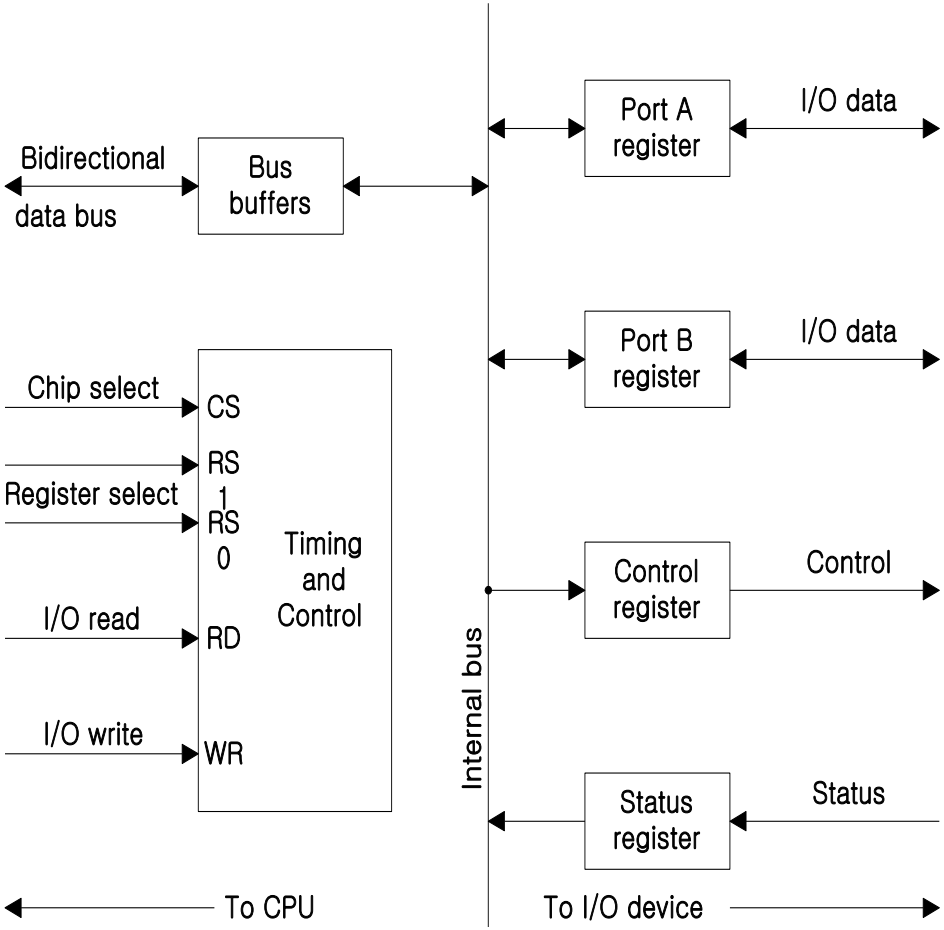
All data transfers occur simultaneously during the occurrence of a clock pulse

*Registers in the **interface** share a common clock with **CPU** registers*

Asynchronous Data Transfer

Internal timing in each unit (CPU and Interface) is independent

Each unit uses its own private clock for internal registers



CS	RS 1	RS 0	Register selected
0	×	×	None : data bus in high-impedance
1	0	0	Port A register
1	0	1	Port B register
1	1	0	Control register
1	1	1	Status register

11-4 Modes of Transfer

Data transfer to and from peripherals

1) Programmed I/O

2) Interrupt-initiated I/O

3) Direct Memory Access (DMA)

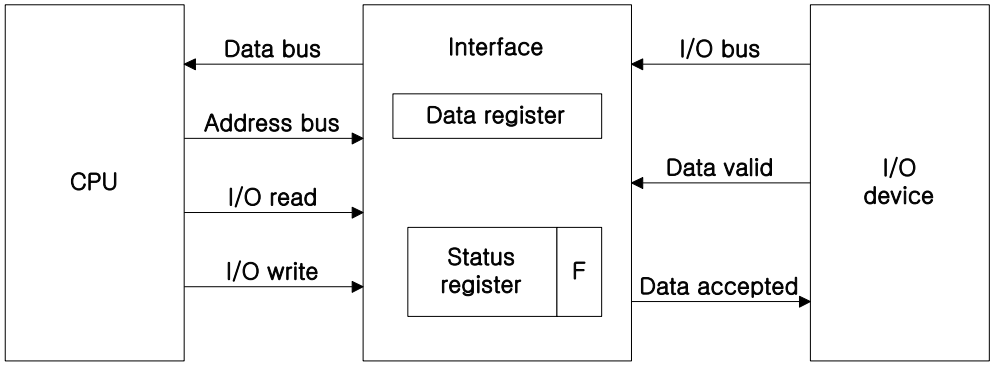
4) I/O Processor (IOP)

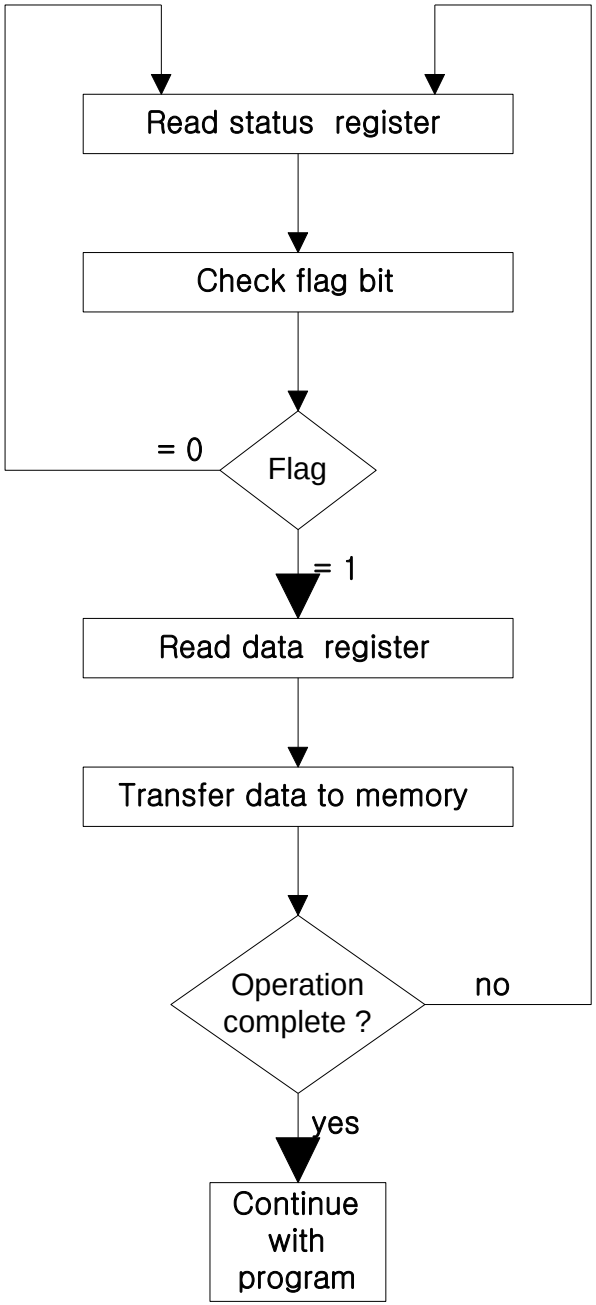
Example of Programmed I/O

Interrupt-initiated I/O

1) Non-vectored : fixed branch address

2) Vectored : interrupt source supplies the branch address (interrupt vector)


CPU
Data bus
Address bus
I/O read
I/O write
Interface
Data register
Status register
F
I/O bus
Data valid
Data accepted
I/O device
F = Flag bit


Read status register
Check flag bit
= 0
Flag
= 1
Read data register
Transfer data to memory
Operation complete ?
no
yes
Continue with program

Software Considerations

I/O routines

software routines for controlling peripherals and for transfer of data between the processor and peripherals

I/O routines for standard peripherals are provided by the manufacturer (Device driver, OS or BIOS)

I/O routines are usually included within the operating system

I/O routines are usually available as operating system procedures (OS or BIOS function call)

11-5 Priority Interrupt

Priority Interrupt

Identify the source of the interrupt when several sources will request service simultaneously

Determine which condition is to be serviced first when two or more requests arrive simultaneously :

1) Software : Polling
2) Hardware : Daisy chain, Parallel priority

Polling

Identify the highest-priority source by software means

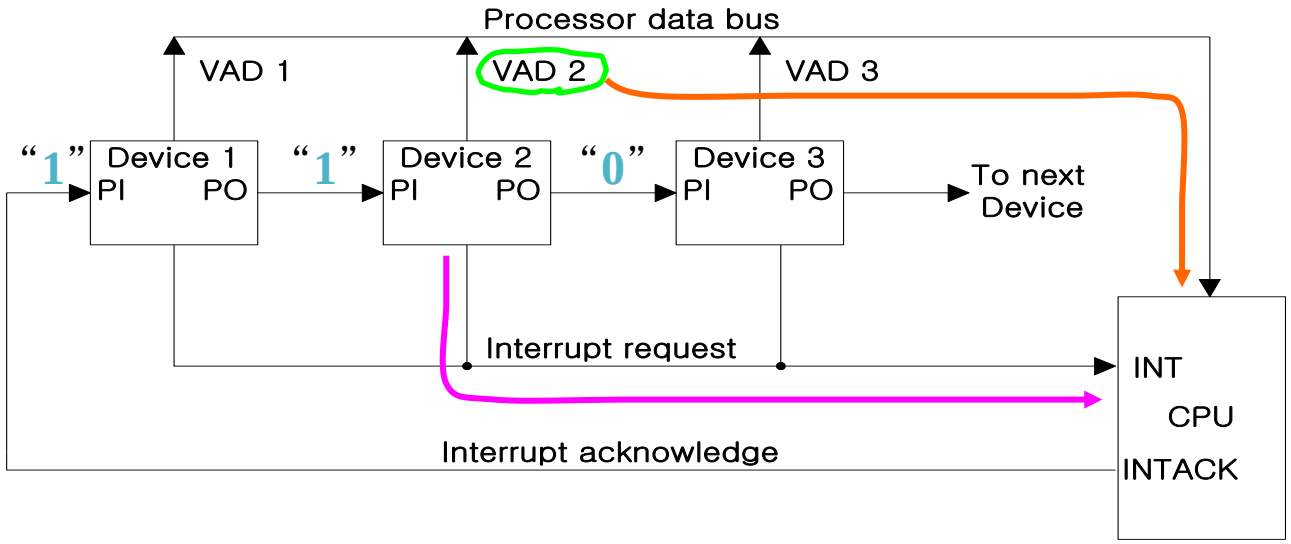
- One common branch address is used for all interrupts
- Program polls the interrupt sources in sequence
- The highest-priority source is tested first

Polling priority interrupt

- If there are many interrupt sources, the time required to poll them can exceed the time available to service the I/O device
- ardware priority interrupt

Device 2 Interrupt Request

-Chaining :


Processor data bus
VAD 1
VAD 2
VAD 3
"1"
Device 1
PI
PO
"1"
Device 2
PI
PO
"0"
Device 3
PI
PO
To next
Device
Interrupt request
INT
CPU
INTACK
Interrupt acknowledge

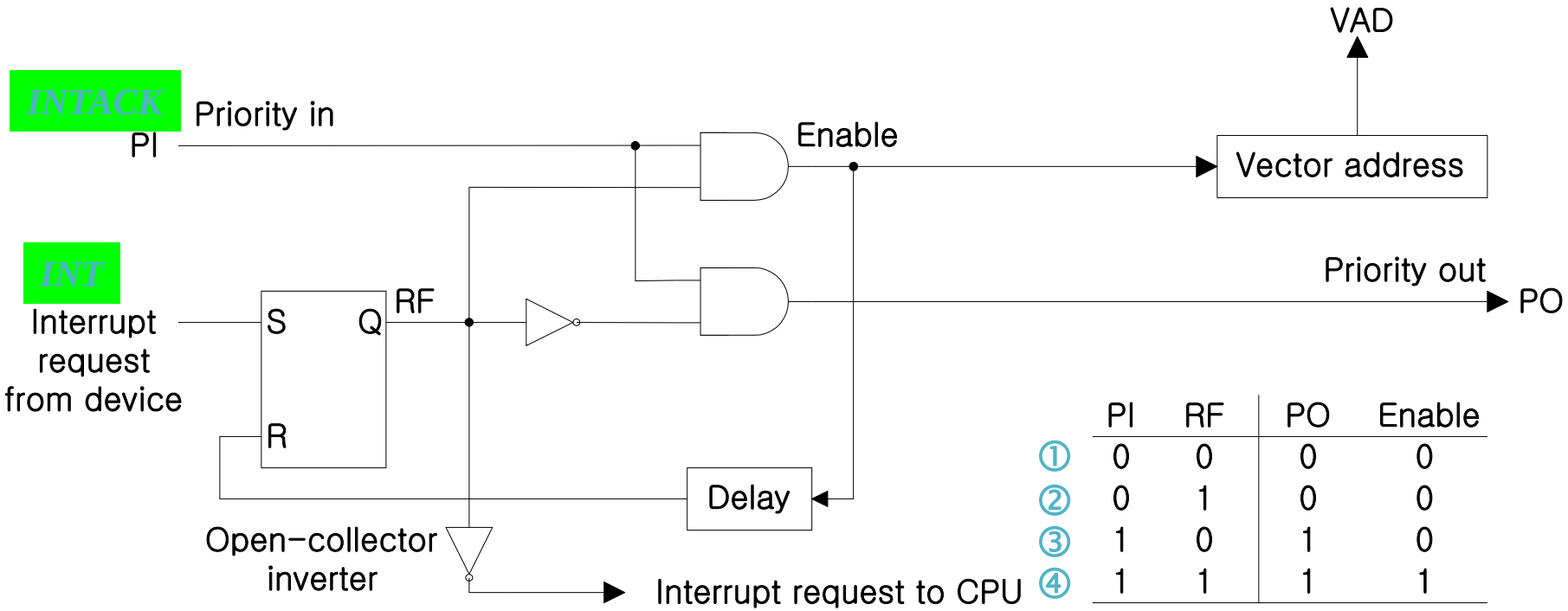
One stage of the daisy-chain priority arrangement :

① *No interrupt request*

② *Invalid : interrupt request, but no acknowledge*

③ *No interrupt request : Pass to other device (other device requested interrupt)*

④ *Interrupt request*



	PI	RF	PO	Enable
①	0	0	0	0
②	0	1	0	0
③	1	0	1	0
④	1	1	1	1

Parallel Priority

Priority Encoder Parallel Priority :

Interrupt Enable F/F (IEN) : set or cleared by the program

Interrupt Status F/F (IST) : set or cleared by the encoder output

Priority Encoder Truth Table :

I_0

Interrupt Cycle

At the end of each instruction cycle, CPU checks IEN and IST

if both IEN and IST equal to "1"

CPU goes to an Instruction Cycle

Sequence of microoperation during Instruction Cycle

$SP \leftarrow SP - 1$: Decrement stack point

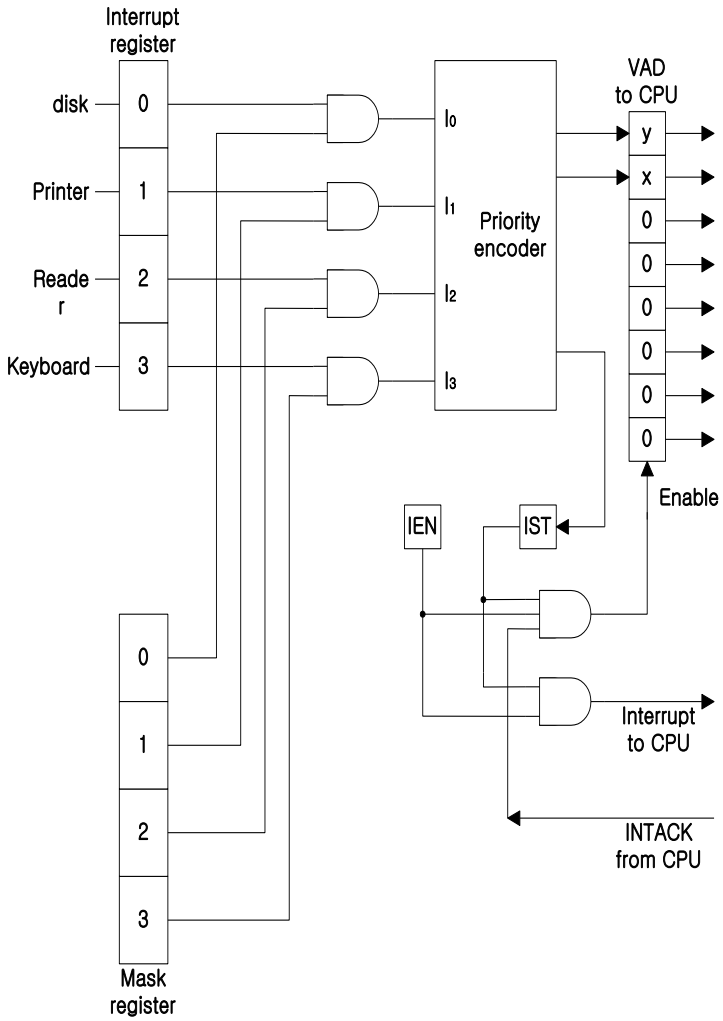
$M[SP] \leftarrow PC$: Push PC into stack

$INTACK \leftarrow 1$: Enable INTACK

$PC \leftarrow VAD$: Transfer VAD to PC

$IEN \leftarrow 0$: Disable further interrupts

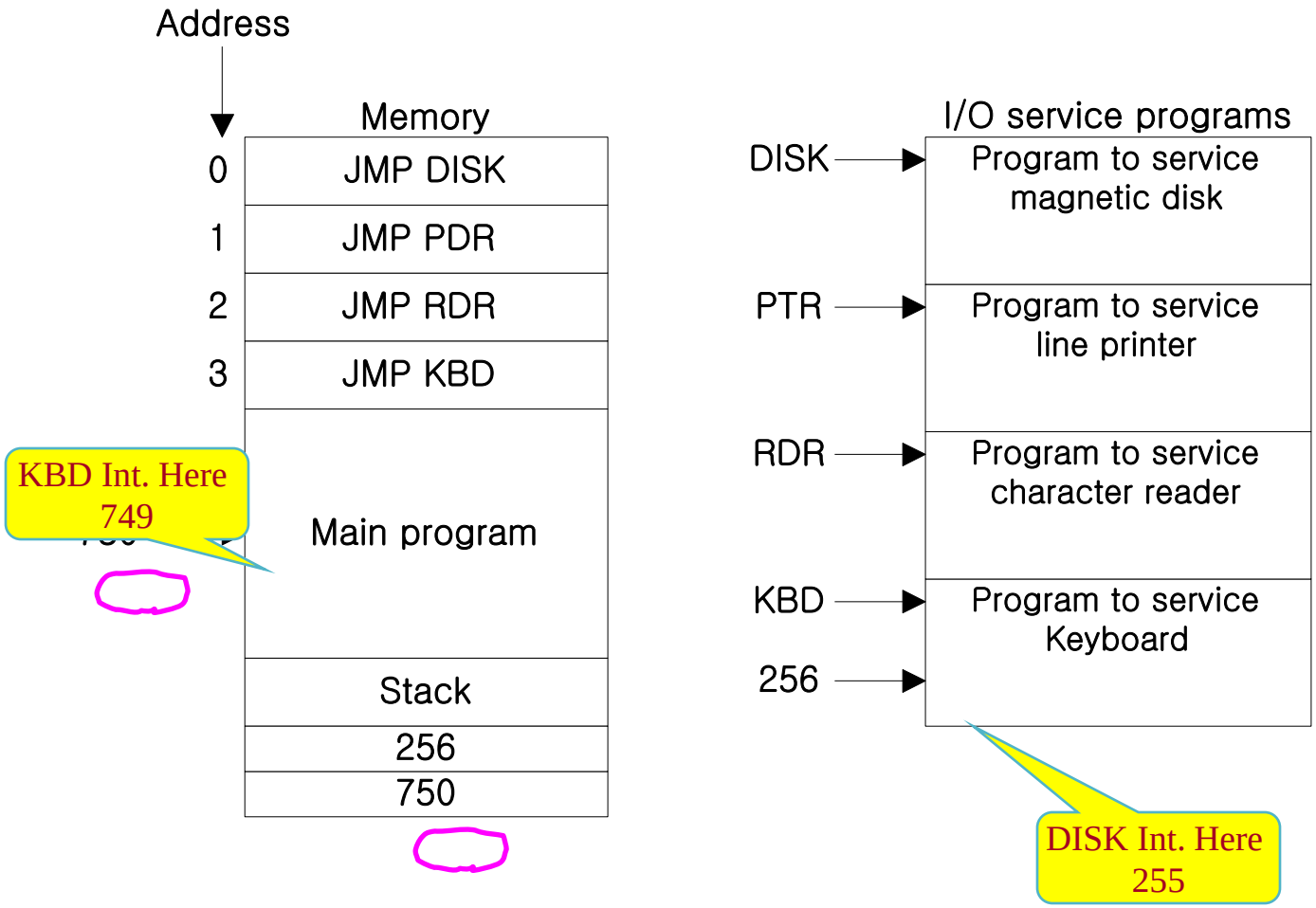
Go to Fetch next instruction



Software Routines

CPU main program 749 KBD interrupt

KBD service program 255 DISK interrupt



Initial Operation of ISR

1) Clear lower-level mask register bit
2) Clear interrupt status bit IST
3) Save contents of processor registers
4) Set interrupt enable bit IEN
5) Proceed with service routine

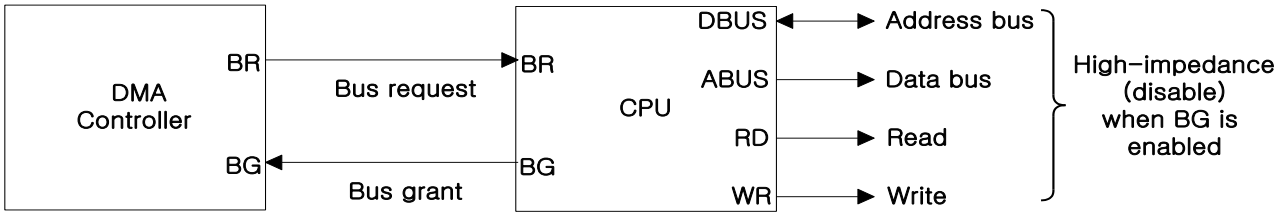
Final Operation of ISR

1) Clear interrupt enable bit IEN
2) Restore contents of processor registers
3) Clear the bit in the **interrupt register** belonging to the source that has been serviced
4) Set lower-level priority bits in the mask register
5) Restore return address into PC and set IEN

11-6 Direct Memory Access (DMA)

DMA

*DMA controller takes over the buses to manage the transfer directly between the **I/O device** and **memory** (Bus Request/Grant)*



Transfer Modes

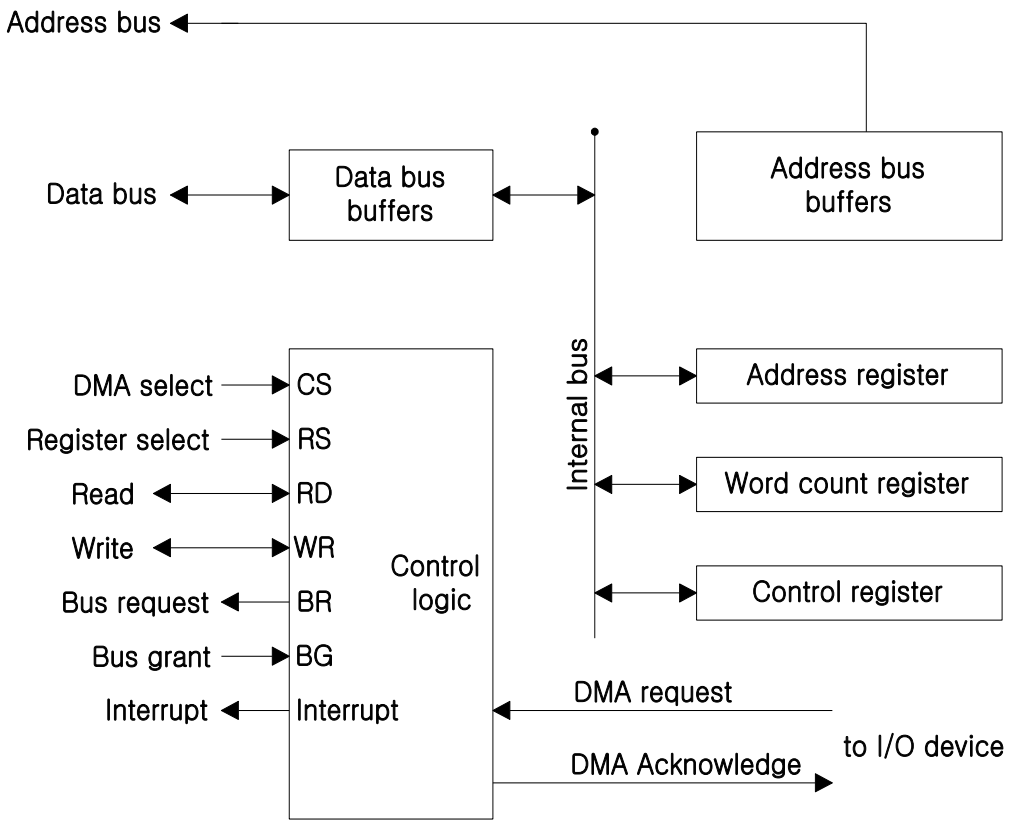
1) Burst transfer : Block

2) Cycle stealing transfer : Byte

DMA Controller (Intel 8237 DMAC) :

DMA Initialization Process

1) Set Address register :
   memory address for read/write
2) Set Word count register :
   the number of words to transfer
3) Set transfer mode :
   read/write,
   burst/cycle stealing,
   I/O to I/O,
   I/O to Memory,
   Memory to Memory
   Memory search
   I/O search
4) DMA transfer start : *next section*
5) EOT (End of Transfer) :
   Interrupt



DMA Transfer (I/O to Memory)

1) I/O Device sends a DMA request

*2) DMAC activates the **BR** line*

*3) CPU responds with **BG** line*

4) DMAC sends a DMA acknowledge to the I/O device

5) I/O device puts a word in the data bus (for memory write)

*6) DMAC write a data to the address specified by **Address register***

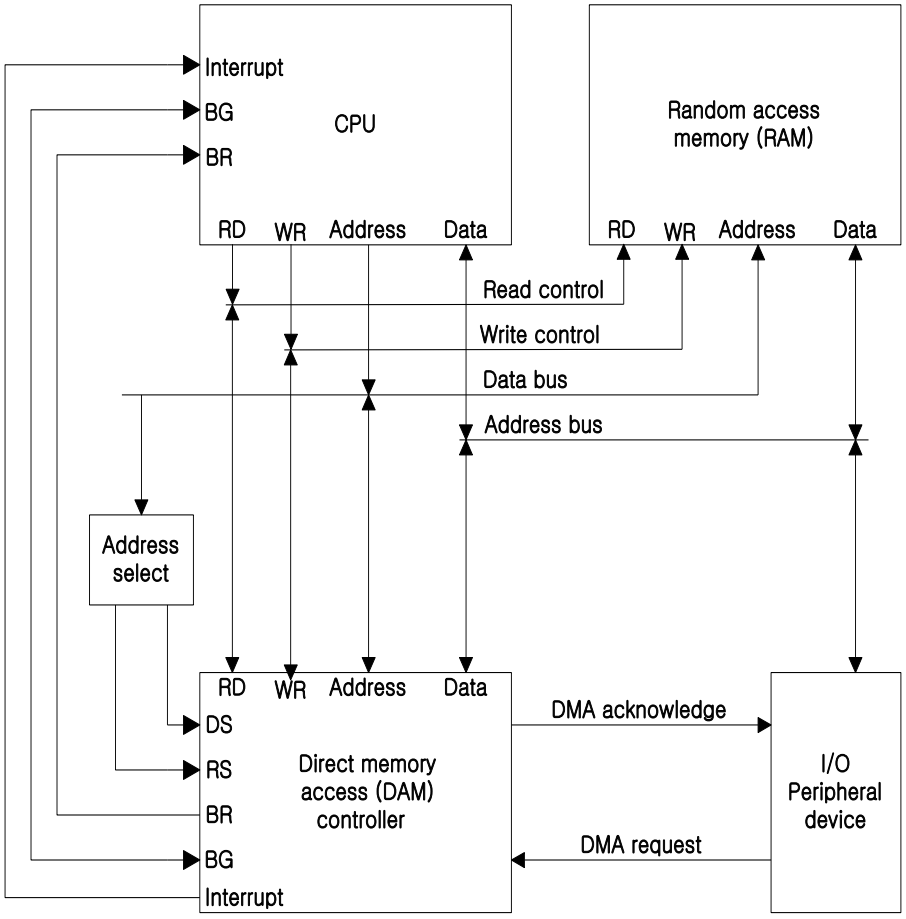
*7) Decrement **Word count register***

*8) **Word count register** = 0*

***EOT** interrupt CPU*

*9) **Word count register** ≠ 0*

DMAC checks the DMA request from I/O device



Reference

Mano, M. Morris (October 1992). Computer System Architecture (3rd ed.). Prentice-Hall. ISBN 0-13-175563-3

Lecture notes of Dept. of Info. & Comm., Korea Univ. of Tech. & Edu., Korea