Master of Computer Applications

CAPOL403R01: Computer Organization & Architecture

Unit V: Lecture 3: Part 2
Interrupt Driven IO – Design Issues

Dr. D. MURALIDHARAN
School of Computing
SASTRA Deemed to be University

Implementation issues – Multiple devices

- Device Identification
  - which device raises the interrupt?
- Handling multiple interrupts
  - Which device has to be serviced when multiple devices need the service?

Multiple Interrupt lines

- Device Identification
  - Many interrupt pins are supported by the processor
  - Each may be used to connect an IO module
  - Processor could identify the device from the pin
  - Processor with many pins for interrupt is not advisable
  - Device ID may not be possible when many devices are connected through a single interrupt line
- Priority
  - The pins may be checked according to a pre-determined order

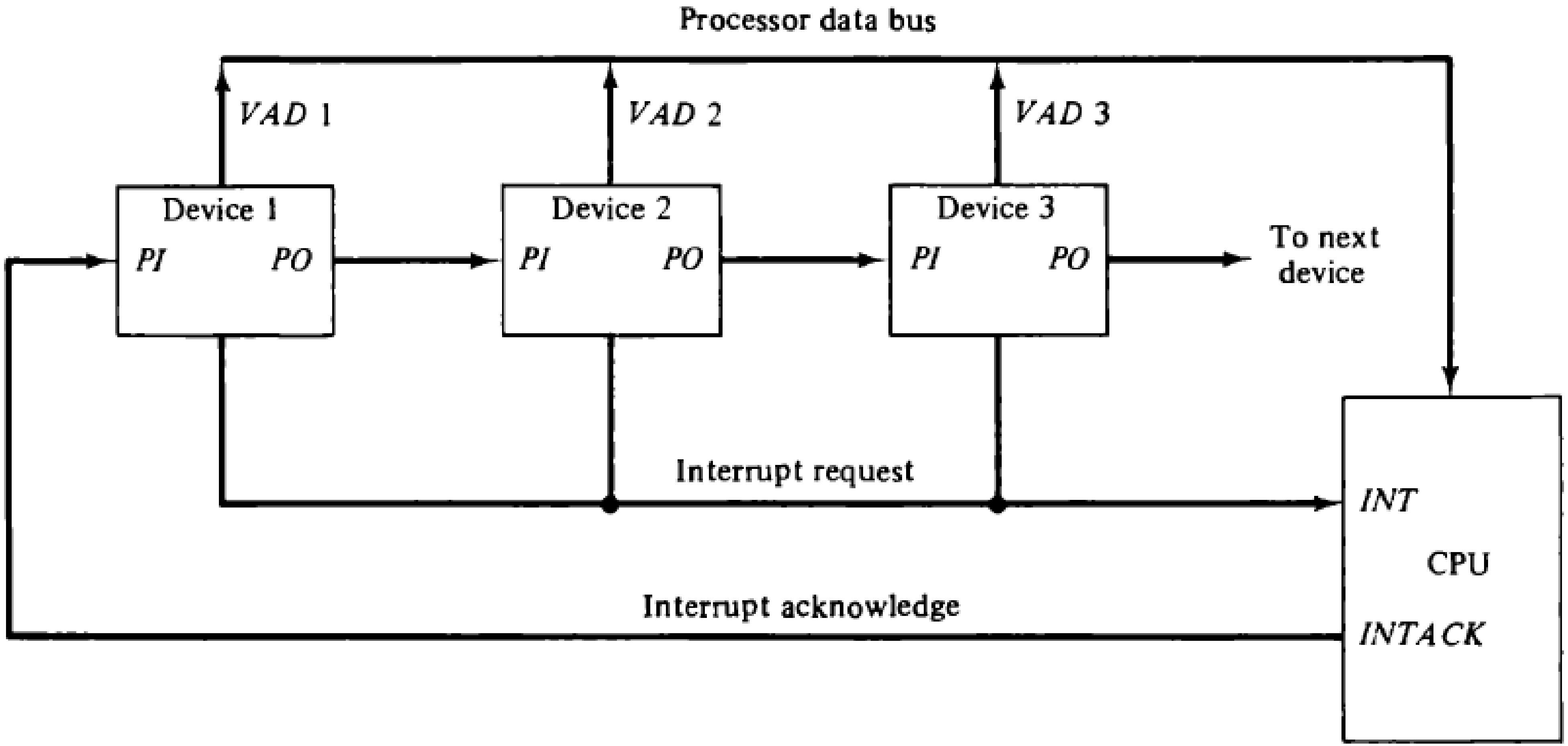
Software polling

- Device Identification
  - Processor checks the devices one by one during the ISR
  - It places the TESTIO signal with the first device ID
  - If the device raised the interrupt, it responds positively
  - If not, the processor places the TESTIO signal with the second device ID and so on…
  - Once the device is identified, the processor enters into the device service routine
  - Time consuming process
- Priority
  - The addresses may be placed according to a pre-determined order

Daisy chaining (Hardware poll)

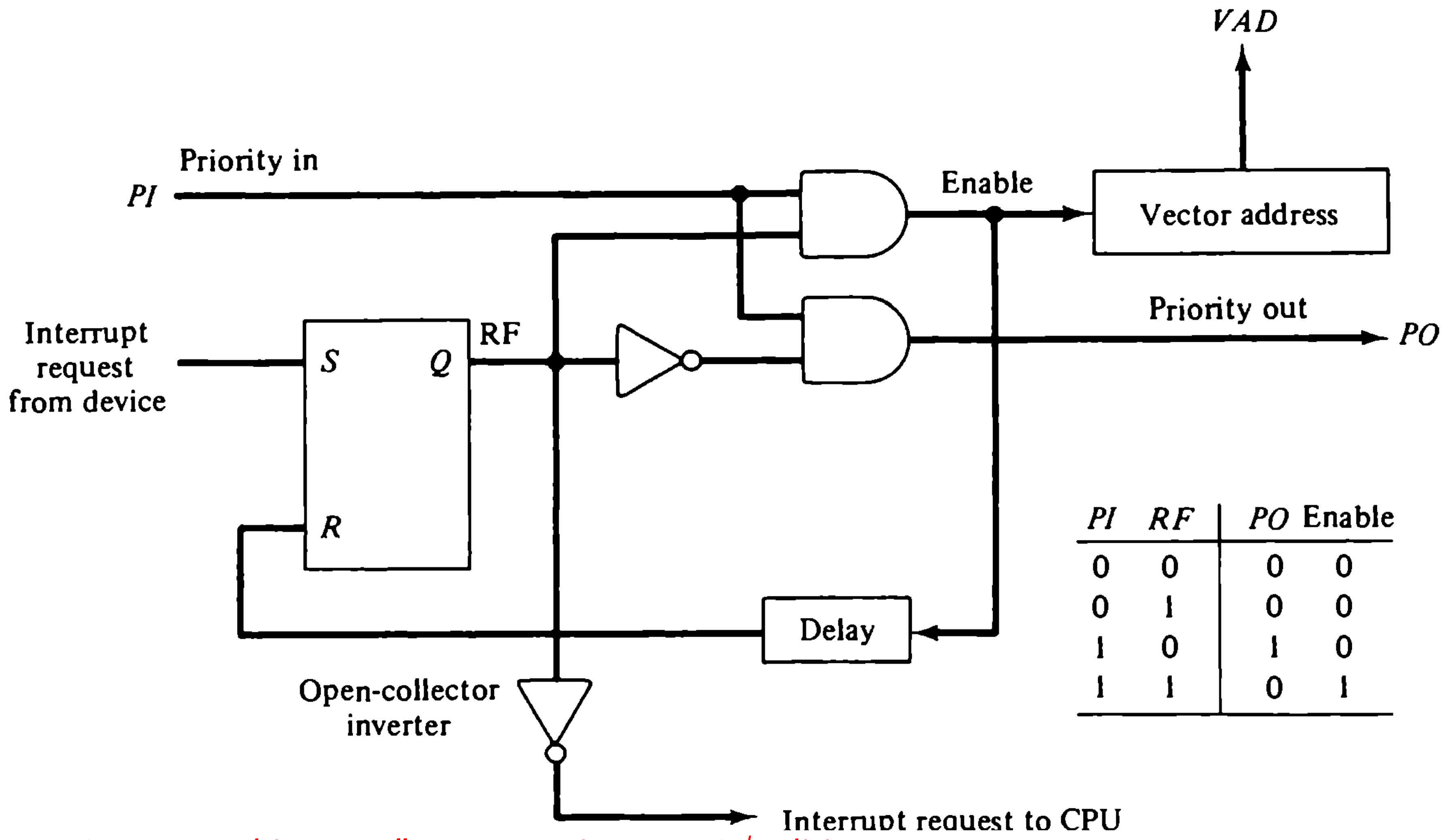
- Device Identification
  - Processor places the INTA signal
  - It goes to the first device
  - If the device did not send the interrupt, it passes the INTA to the next device
  - The device which has raised the interrupt places a word on the data line
  - The word is called as "vector"
    - It may be some unique identifier
    - For example, it may be the address of the device
    - The processor uses it to invoke the appropriate device service routing
    - In this way, processor avoids to check each and every device

Daisy chaining…



Courtesy: "Computer System Architecture" – M. Morrismano, 3rd Edition

Daisy chaining…



PI	RF	PO	Enable
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1

Courtesy: "Computer System Architecture" – M. Morrismano, 3rd Edition

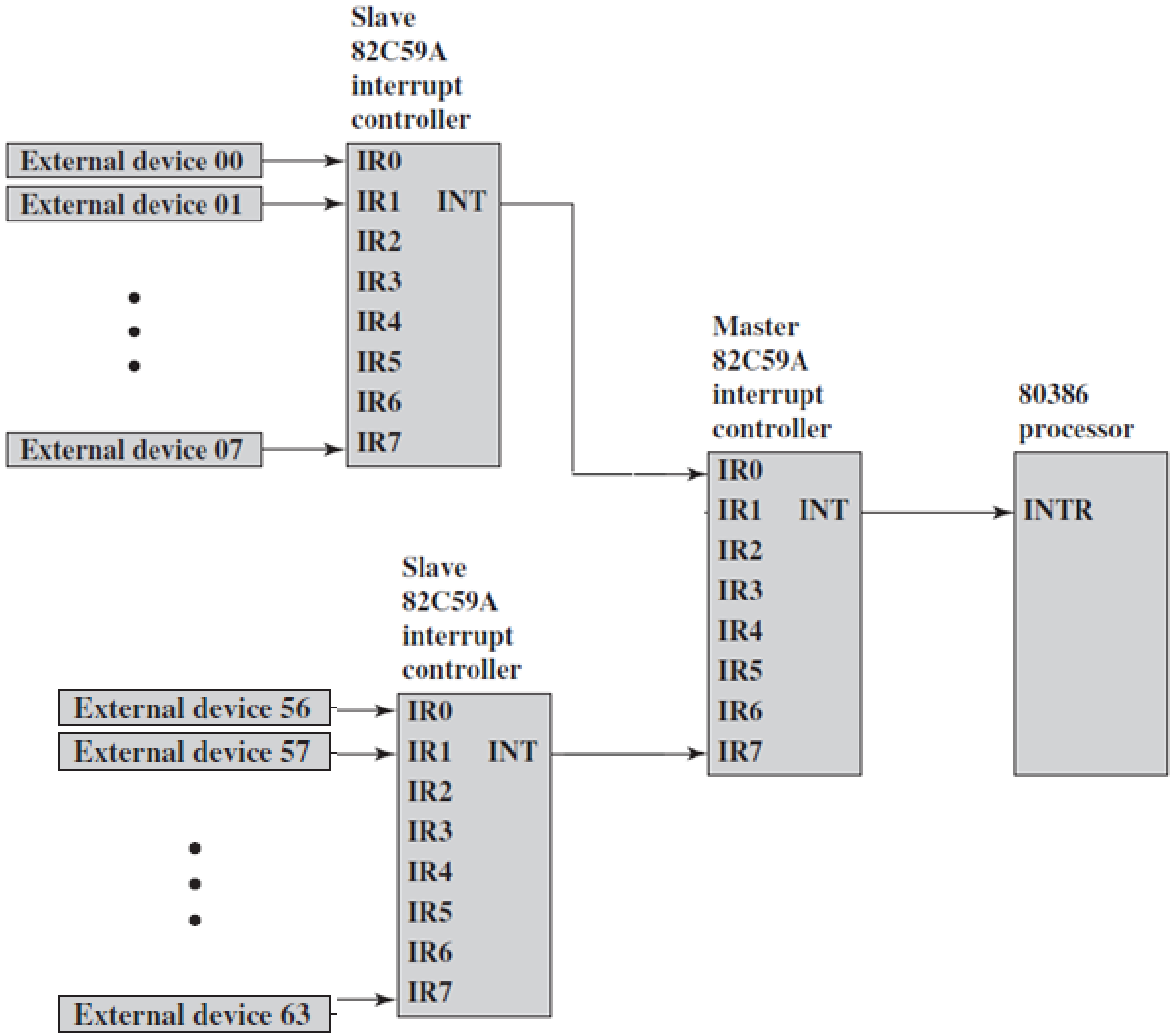
Bus arbitration

- Device Identification
  - The device holds the bus
  - It raises the interrupt
  - Processor responds only to this device
- Multiple interrupts
  - Many methods are available
  - In a centralized method, an arbiter may choose the device and gives the bus control
  - In a distributed method, all devices will compete by placing their address and knows whether they are in the competition or not
  - The winner gains the bus control

Example 1: 8259 PIC

- INTEL 80386 has one INTR and one INTA lines only
- To service multiple devices, usually it is configured with 8259
- 8259 is responsible for interrupt management
- A single 8259 handles up to eight modules
- It accepts interrupt requests from the attached modules
- If more than one module needs the service from the processor, 8259 determines the highest priority module
- It signals the processor by raising INTR
- After getting INTA, it places the vector on the data line
- By cascading, more than 8 modules can be handled

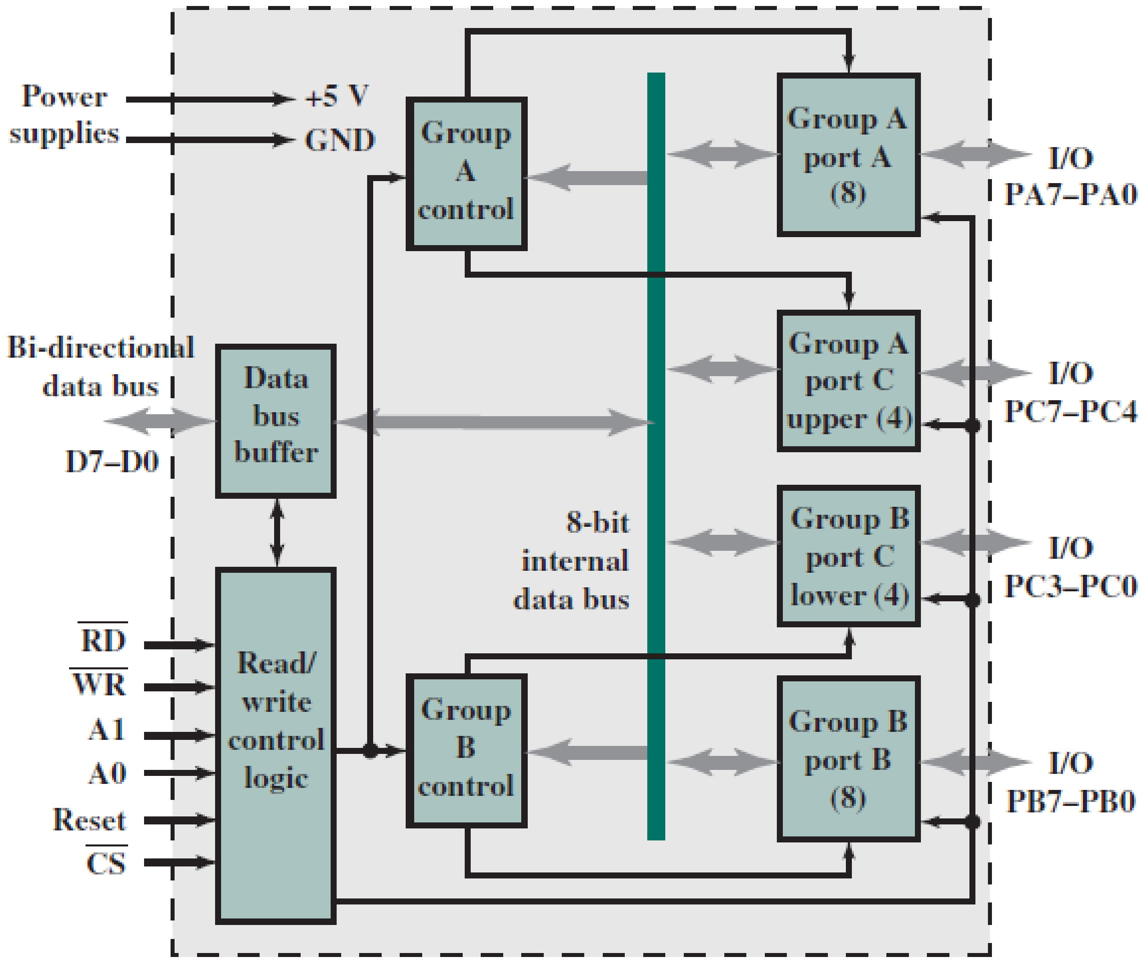
8259…

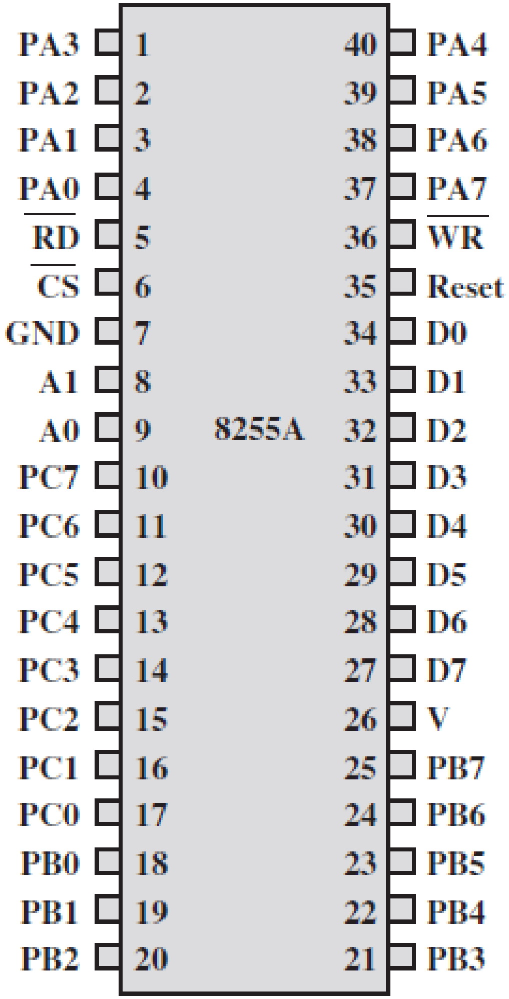

Slave
82C59A
interrupt
controller
External device 00
External device 01
External device 07
IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7
INT
Master
82C59A
interrupt
controller
IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7
INT
80386
processor
INTR
Slave
82C59A
interrupt
controller
External device 56
External device 57
External device 63
IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7
INT

8259… (Interrupt modes)

- Interrupt modes
  - Fully nested: The interrupt requests are ordered in priority from 0 to 7
  - Rotating(Equal priority): The one which is serviced currently gets the lowest priority
  - Special mask: This mode allows the processor to disable some devices

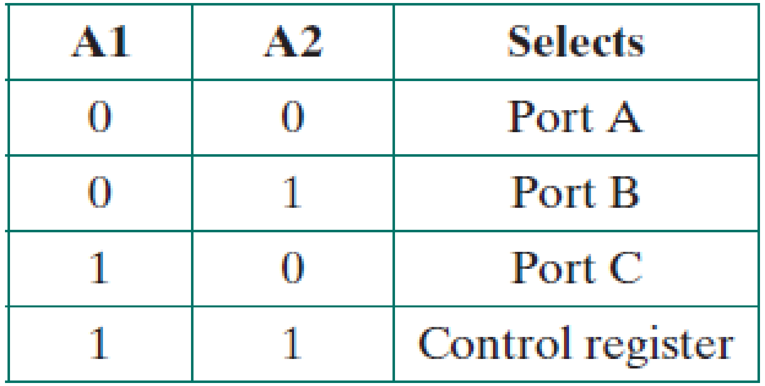
Example 2: 8255 PPI





8255…

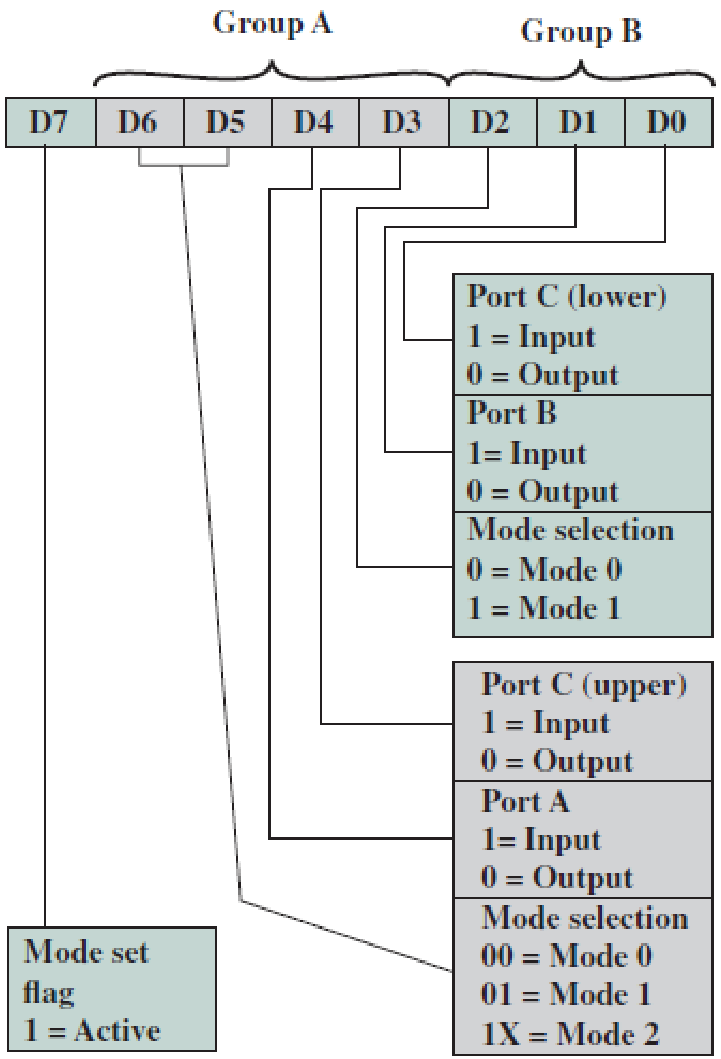
- The processor selects a mode by placing a control word in the control word register of the PPI



A1	A2	Selects
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Control register

- Mode 0: Basic I/O mode – All 24 pins are IO lines
- Mode 1: Handshake IO - Ports A & B are data ports and Port C provides control lines to these ports
- Mode 2: Bidirectional IO – Port A one direction, Port B the other direction, Port C provides control lines
- When D7 of CW is ‘0’, individual bits of port C can be set or reset

8255…



Group A

Group B

D7 D6 D5 D4 D3 D2 D1 D0

Port C (lower)
1 = Input
0 = Output

Port B
1= Input
0 = Output

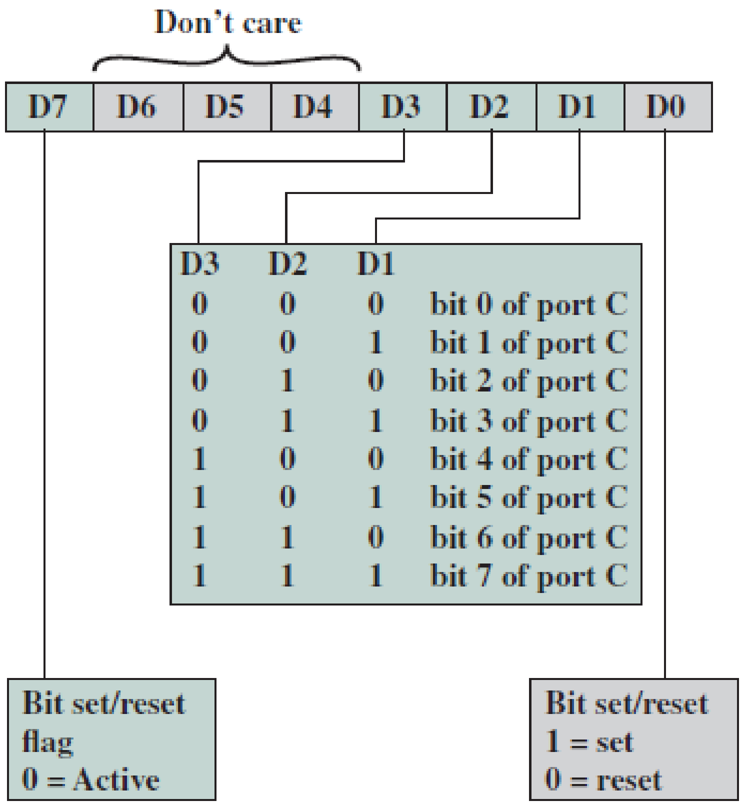
Mode selection
0 = Mode 0
1 = Mode 1

Port C (upper)
1 = Input
0 = Output

Port A
1= Input
0 = Output

Mode selection
00 = Mode 0
01 = Mode 1
1X = Mode 2

Mode set
flag
1 = Active



Don't care

D7 D6 D5 D4 D3 D2 D1 D0

D3	D2	D1	
0	0	0	bit 0 of port C
0	0	1	bit 1 of port C
0	1	0	bit 2 of port C
0	1	1	bit 3 of port C
1	0	0	bit 4 of port C
1	0	1	bit 5 of port C
1	1	0	bit 6 of port C
1	1	1	bit 7 of port C

Bit set/reset
flag
0 = Active

Bit set/reset
1 = set
0 = reset

Examples

- What is the control word to program port A as handshake Input and port B as basic Output? Can port C pins be used as IO in this case?
- What is the control word to program port A as basic input and port B as handshake input? What is the function of port C pins in this case?
- Tell the functions of the PPI when the processor places the control word (i) “10010000” (ii) “01110011” in the control register.

Thank you