

EEE 410 – Microprocessors I

Spring 04/05 – Lecture Notes # 16

Outline of the Lecture

- **Input/Output (I/O) and Device Interfacing**

INPUT/OUTPUT (I/O) AND DEVICE INTERFACING

In addition to memory, 80x86 microprocessors are able to access I/O ports. Ports are accessed either to bring data from the port into the CPU (inputting) or to send data from the CPU to the port (outputting).

Input/Output Instructions in 8086 CPU

The 8086 microprocessor can access information from ports as well as from the memory. There are two instructions for this purpose “IN” and “OUT”. These instructions can send data from the accumulator (AX, AL or AH) to ports or receive data from ports into the accumulator.

Case 1: 8-bit data ports

	Inputting Data		Outputting Data	
Format	IN	dest,source	OUT	dest,source
(1)	IN	AL,port#	OUT	port#,AL
(2)	MOV	DX,port#	MOV	DX,port#
	IN	AL,DX	OUT	DX,AL

In format (1) above, *port# is the address of the port*, and can be from 00 to FFH. This 8-bit address allows 256 input ports and 256 output ports. No segment register is involved in computing the address, in contrast to the data accessed from memory.

In format (2), *port# is the address of the port*, and can be from 0000 to FFFFH. This 16-bit address allows 65,536 input ports and 65,536 output ports. No segment register (DS) is involved.

Ex: Write a sequence of instructions that will output FFH to a byte-wide output port at the address ABH of the I/O address space?

```
MOV AL,FFH
OUT ABH,AL
```

Ex: Write a sequence of instructions that will output FFH to an output port at the address B000H of the I/O address space?

```
MOV DX,B000H           ;Note that 16-bit address must be in DX
MOV AL,FFH
OUT DX,AL
```

Ex: Assume that the port address 22H is an input port for monitoring the temperature. Write Assembly language instructions to monitor the port continuously for the temperature of 100 degrees. If it reaches 100, then BH should contain ‘Y’.

```
BACK:  IN  AL,22H      ;get the temperature data from port# 22H
        CMP AL,100    ;is temp =100?
        JNZ BACK     ;if not, keep monitoring
        MOV BH,'Y'   ;temp =100, load 'Y' into BH
```

Case 2: 16-bit data ports

	Inputting Data		Outputting Data	
Format	IN	dest,source	OUT	dest,source
(1)	IN	AX,port#	OUT	port#,AX
(2)	MOV	DX,port#	MOV	DX,port#
	IN	AX,DX	OUT	DX,AX

Ex: Assume that AX=98F6H and the output port address is 47H, then

OUT 47H,AX

- In the above case F6H, the content of AL, goes to port address 47H and 98H, the content of AH goes into the port address 48H.
- In other words the low byte goes to the low port address, and the high byte goes to the high port address.

- **Types of Input / Output**

-

Isolated Input / Output

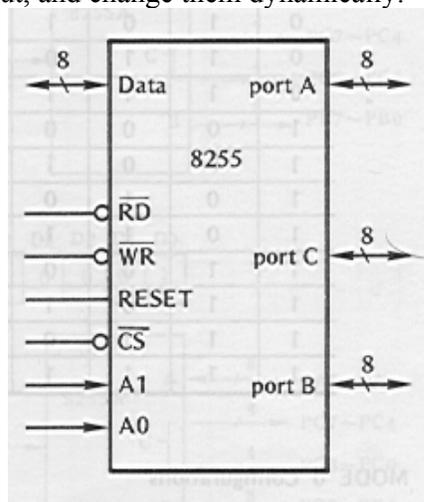
Isolated I/O (also referred as peripheral I/O or direct I/O) uses dedicated input (IN) and output (OUT) instructions to transfer data between the I/O device and the microprocessor. Isolated I/O uses separate map for the I/O space, freeing the entire memory for use by the program.

Memory-mapped Input / Output

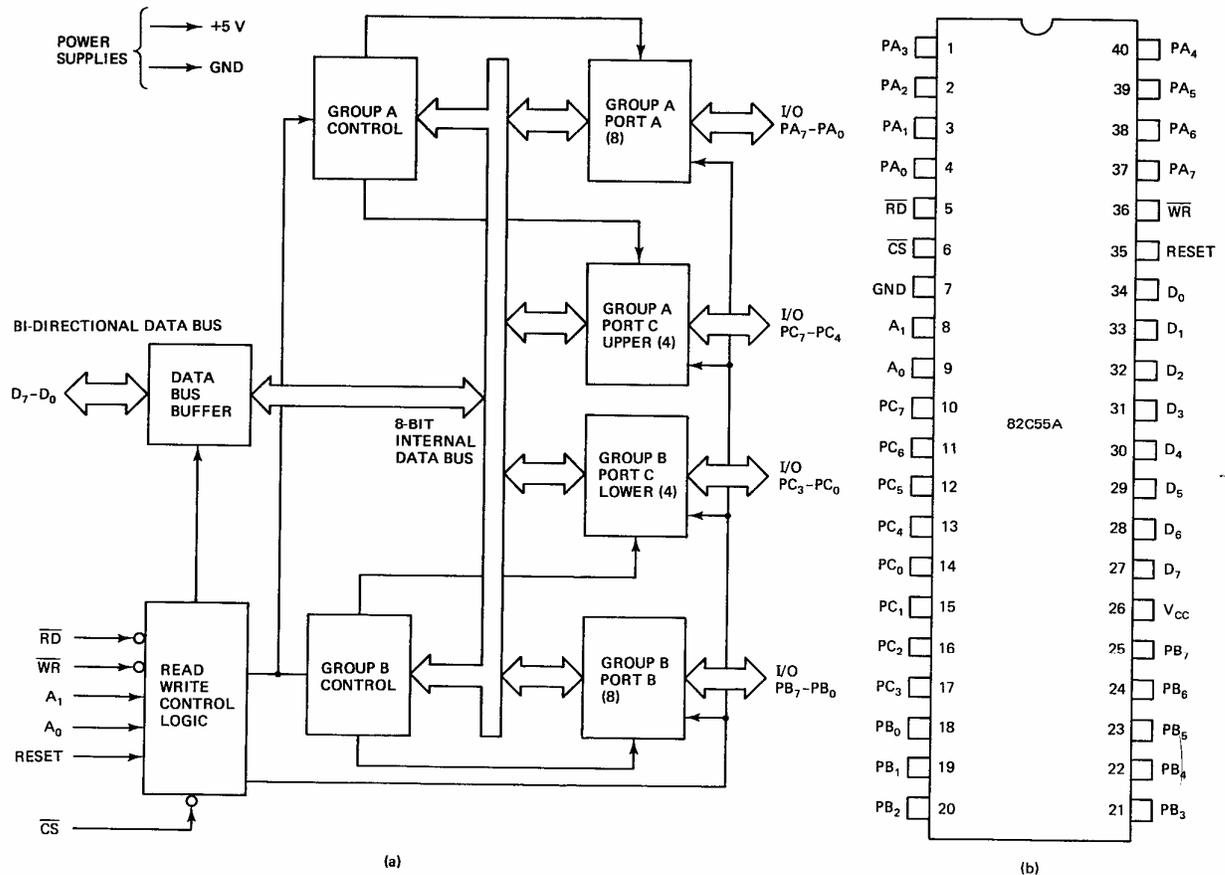
In memory-mapped I/O, a memory location is assigned to be input or output port, so it uses a portion of memory space for I/O transfers. This reduces the amount of memory available, but simplifies the hardware. “MOV” instruction is used instead of “IN” and “OUT” instructions. M6800, M68000 and RISC processors are examples to this type.

The 8255 programmable Peripheral Interface (PPI)

The 8255 PPI chip is designed to permit easy implementation of *parallel I/O* into the microcomputer. The 8255 is one of the most widely used I/O chips. It has three separately accessible ports, A, B, and C. Above all the user can program the individual ports to be input or output, and change them dynamically.



The block diagram of the 8255 Chip



Detailed Block Diagram of 8255 (a), and the Pin layout (b)

- **PA0-PA7 (Port A)** : This 8-bit port can be programmed all as input or all as output or all bits as bi-directional input/output.
- **PB0-PB7 (Port B)** : This 8-bit port can be programmed all as input or all as output or all bits as bi-directional port.
- **PC0-PC7 (Port C)** : This 8-bit port can be all input or all output. It can also be split into two parts:

CU (Upper 4 bits PC4-PC7) and

CL (Lower 4 bits PC0-PC3).

Each can be used for input or output.

- **RD and WR** : These two active low signals are inputs to the 8255. If the 8255 is using isolated I/O design, IOR or IOW of the system bus are connected to these two pins. If the port uses memory-mapped I/O, MEMR and MEMW activate them.
- **RESET**: This is an active high signal input to the 8255 used to clear the control register. When RESET is activated, all ports are initialized as input ports.
- **A0, A1, and CS**: While CS selects the entire chip, it is A0 and A1 that select the specific port. A0 and A1 are used to access ports A, B, C, or the control register according to the table below.

A1	A0	Selects:
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Control Register

Table 1:

Programming Modes of PPI

1) Mode 0: Simple input or output mode:

- This mode 0 is the *basic input output* mode. In this mode any ports A,B or C can be programmed as input or output.
- Note that in this mode a given port cannot be both input or output port at the same time.
- One major characteristic of port C is that one can program CL (PC0-PC3) and CU (PC4-PC7) independent of each other.

The Control Register; is used to configure the individual ports as to be in input or output mode.

The Control Register in Mode 0

D7	D6	D5	D4	D3	D2	D1	D0
1=Mode 0 I/O Mode	Mode selection 00 = Mode 0 01 = Mode 1 1X = Mode 2		Port A 1=input 0=output	Port C (PC7-PC4) 1=input 0=output	Mode Selection 0=Mode 0 1=Mode 1	Port B 1=input 0=output	Port C (PC3-PC0) 1=input 0=output

The port Addresses for the 8255

Port	Address	A0 A1
A	300H	0 0
B	301H	0 1
C	302H	1 0
Control Register	303H	1 1

Note that first two bits of the Address are used for the mode selection as shown in the table given above.

Ex: Configure 8255 as follows: port A as input, B as output, and all the bits in C as output. Determine the content of the Control Register and, Program the ports to input data from A and send it to both B and C. (Assume the standard port addresses of 8255 given above)

Soln: a) Control register:

1	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---

= 90H

- b)
- | | |
|-------------|---|
| MOV AL,90H | ; This control byte is used to initialize the PPI |
| MOV DX,303H | ;get the address of Control Register |
| OUT DX,AL | ;output the Control Byte to the Control register |
| MOV DX,300H | ;get the address of Port A |
| IN AL,DX | ;input data from Port A |
| MOV DX,301H | ;get the address of Port B |
| OUT DX,AL | ;output data to Port B |
| MOV DX,302H | ;get the address of Port C |
| OUT DX,AL | ;output data to Port C |

Ex: Configure the ports of 8255 as follows: port A=input, B=output/ PC0 –PC3=input, and PC4-PC7=output.

Determine the content of the Control Register and,

Program the 8255 to get data from port A and send it to port B. In addition, input data from PCL and send out to PCU.

(PCL=Lower 4 bits (nibble) of Port C, PCU;Upper 4 bits(nibble) of Port C).

(Assume the standard port addresses of 8255 given above)

Soln: a) Control register:

1	0	0	1	0	0	0	1
---	---	---	---	---	---	---	---

= 91H

```

b)  MOV  AL,91H           ; This control byte is used to initialize the PPI
     MOV  DX,303H        ;get the address of Control Register
     OUT  DX,AL          ;output the Control Byte to the Control register
     MOV  DX,300H        ;get the address of Port A
     IN   AL,DX          ;input data from Port A
     MOV  DX,301H        ;get the address of Port B
     OUT  DX,AL          ;output data to Port B
     MOV  DX,302H        ;get the address of Port C
     IN   AL,DX          ;get the data from Port C
     AND  AL,0FH         ;mask the upper bits
     MOV  CL,4           ;rotate count =4
     ROL  AL,CL          ;shift the bits to upper position
     OUT  DX,AL          ;output PCL to PCU

```

2) Mode 1: I/O with handshaking capability:

One of the most powerful features of the 8255 is the ability to handle handshaking signals. Handshaking refers to the process of communicating back and forth between two intelligent devices. Printer is a good example for Mode 1 Interfacing.

Port A and B are used for input or output while Port C is used for handshaking signals.

3) Mode 2: Bi-directional I/O with handshaking:

In this mode data is transferred both in and out via the same port with handshaking capability. Port A is used as a bi-directional port and port C is used for handshaking signals. Port B can be configured to be in Mode 0 or 1.