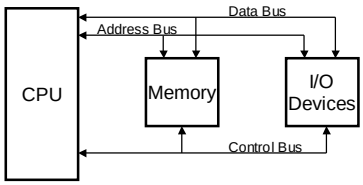
Parallel I/O Examples

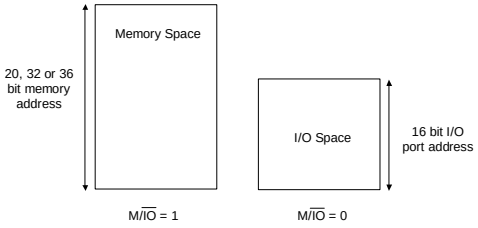
```
in   al,    40h    ;al gets 1 byte from port 40h
in   ax,    255    ;ax gets 2 bytes from port ffh
in   al,    dx     ;ax gets 1 byte from port address in dx
in   eax,   dx     ;eax gets 4 bytes from port addr. in dx
out 80h,    al     ;send contents of al to port 80h
out dx,     eax    ;send contents of eax to port addr. in dx
```



Address Bus is Shared - Control Bus Indicates I/O or Memory

Parallel I/O

- I/O Port Similar to Memory Address
- x86 Uses M/$\overline{IO}$ Control Bus Output to "Signal" I/O *NOT* Memory
- Port Address (Number) 16 Bits Long (**0000h** – **ffffh**)
- Port Width is 8, 16 or 32 Bits (Parallel)



I/O Port Addressing Modes

- Two Addressing Modes
  1) Immediate Port Address
     - Can only be 1 byte
     - Can only Address Ports **00h** through **ffh**
  2) Port Address Present in **DX**
     - Can Address all Ports **0000h** through **ffffh**
- Can only Use **DX** for Port Addresses
- Can only Use **AL, AX, EAX** for Port Data

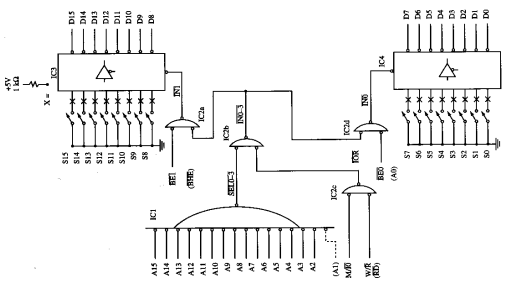
I/O Data Transfer

```
in   al,    40h    ;al gets 1 byte from port 40h
in   ax,    255    ;ax gets 2 bytes from port ffh
in   al,    dx     ;ax gets 1 byte from port address in dx
in   eax,   dx     ;eax gets 4 bytes from port addr. in dx
out 80h,    al     ;send contents of al to port 80h
out dx,     eax    ;send contents of eax to port addr. in dx
```

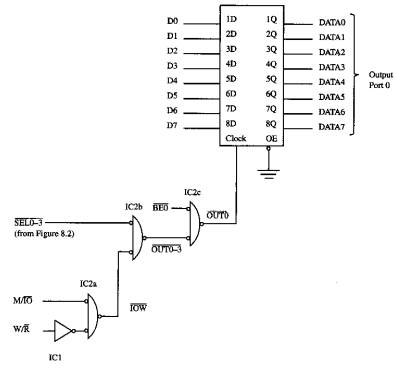
386+

```
insb  72           ;receive byte string from port 72
                   ;store in location at es:di
insw  dx           ;receive word strng from port addr. in dx
insd  dx           ;receive doub. word string
outsb ffh          ;send byte string to port 255
                   ;source string located in memory at ds:si
outsw dx           ;send word string to port addr. in dx
outsd dx           ;send doub. word string to port
```

Input I/O Port Interface



Output I/O Port Interface

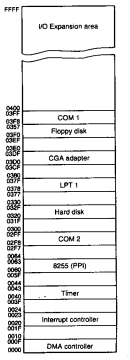


Standard IBM AT I/O Memory Map

ALSO:

0000h-03ffh	Reserved for Computer System
0400h-0fffh	Available for User
00f8h-00ffh	Reserved for Intel
00f8h-00fdh	Used for x87 FPU

BE CAREFUL WHICH PORTS YOU INTERFACE TO!!!



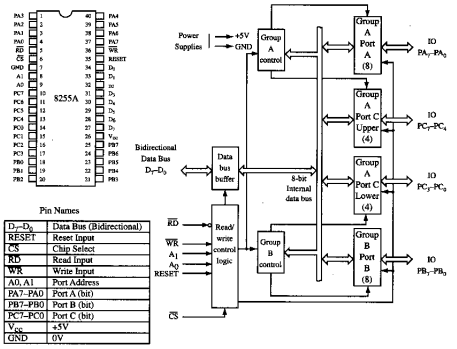
Memory Mapped I/O

- Possible to Design Address Decoder to "Divert" Read or Write to a Device Instead of Memory
- Replace Memory Chip with Set of Latches (CPU Write) and Set of Tri-State Buffers (CPU Read)
- In Hardware, Appears to be I/O Port Interface
- In Software, Appears to be Memory Location
  - Use **`mov`** Instead of **`in`**, **`out`**
- Advantage is Large Variations in Addressing Modes
- Disadvantage is Particular Memory Address can NEVER be Used for Data Storage

Packaged I/O Interface Circuits

- Several Developed by Intel to Ease Design Burden
  - Provide a Complete I/O Interface on a Single Chip
- Examples of Common I/O Interface Chips:
  - 8255A Programmable Peripheral Interface (PPI)
  - 8259 Programmable Interrupt Controller (PIC)
  - 8253/4 Programmable Interval Timer (PIT)
  - 8237 Programmable DMA Controller
- IBM PC/XT had these Chips on System Board
- Modern PCs have Functionality Included in System Chipset

8255A PPI – Parallel I/O Interface



Pin Names

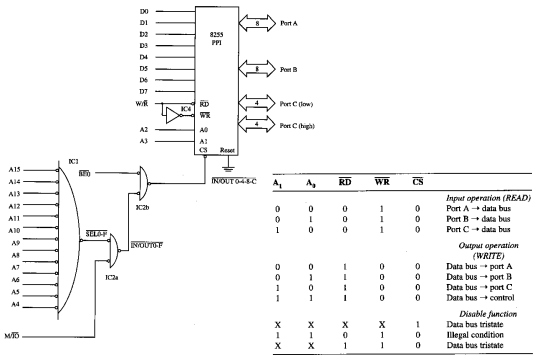
D_7–D_0	Data Bus (Bidirectional)
RESET	Reset Input
$\overline{CS}$	Chip Select
$\overline{RD}$	Read Input
$\overline{WR}$	Write Input
A0, A1	Port Address
PA7–PA0	Port A (bit)
PB7–PB0	Port B (bit)
PC7–PC0	Port C (bit)
V_{CC}	+5V
GND	0V

8255A PPI – Assets and Capabilities

- 24 I/O Lines in 3 8-bit Port Groups – A, B, C
- A, B can be 8-bit Input or Output Ports
- C can Serve as 2 4-bit Input or Output Ports
- 3 Modes of Operation:
  - Mode 0: A, B, C Simple Input or Output Level Sensitive Ports
  - Mode 1: A, B Input or Output Ports with Strobe Control in C
  - Mode 2: A is Bidirectional with Control/Handshake in B and C
- A, B can only Change 1 Byte at a Time
- C has Individual Bit Set/Reset Capability

Advantage is Non-Dedicated Circuit can Change Port Configuration with Software and No "Glue Logic"

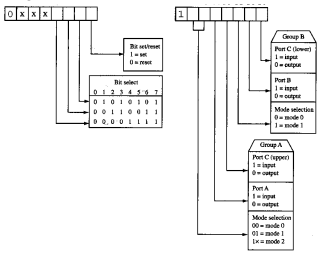
PPI Interface to 486



A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
					Input operation (READ)
0	0	0	1	0	Port A → data bus
0	1	0	1	0	Port B → data bus
1	0	0	1	0	Port C → data bus
					Output operation (WRITE)
0	0	1	0	0	Data bus → port A
0	1	1	0	0	Data bus → port B
1	0	1	0	0	Data bus → port C
1	1	1	0	0	Data bus → control
					Disable function
X	X	X	X	1	Data bus tristate
1	1	0	1	0	Illegal condition
X	X	1	1	0	Data bus tristate

Programming/Controlling the PPI

- 8255A PPI Contains a "write-only" Control Register
- Accessed when **A0**=**A1**=1
- When PPI is Reset (Active high strobe on **RESET** pin):
  - Default is A, B, C are Mode 0 Input Ports
- Control Register Also Used to set/reset Port C Bits Individually



PPI Programming Example

```
;Assume Address Decoder Designed for PPI Base Address of 0400h
;PPI Connected to D7-D0 on 486 data bus
       mov    dh,    04h
       mov    dl,    0ch
       mov    al,    82h
       out    dx,    al
```

What is the above Code Doing????

Control Word Example

```
;Assume Address Decoder Designed for PPI Base Address of 0400h
;PPI Connected to D7-D0 on 486 data bus
       mov    dh,    04h    ;Let dh point to base address
       mov    dl,    0ch    ;Select the control register
       mov    al,    82h    ;Place 82h Control Word into al
       out    dx,    al     ;Write al contents to PPI
```

D_7=1	Register Receives Control Word (not bit set/reset)
D_6D_5=00	A and C_7-C_4 are Mode 0 (Group A)
D_4=0	A is Level Sensitive Output
D_3=0	C_7-C_4 is Level Sensitive Output
D_2=0	B and C_3-C_0 are Mode 0 (Group B)
D_1=1	B is Level Sensitive Input
D_0=0	C_3-C_0 are Level Sensitive Output