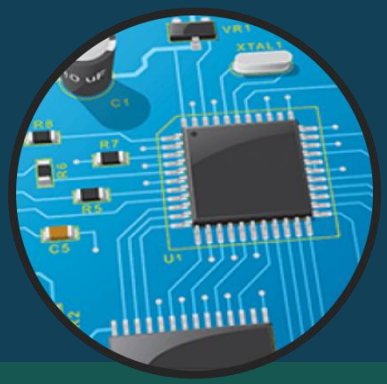
IO Addressing & Basic Interfacing



Topic#8

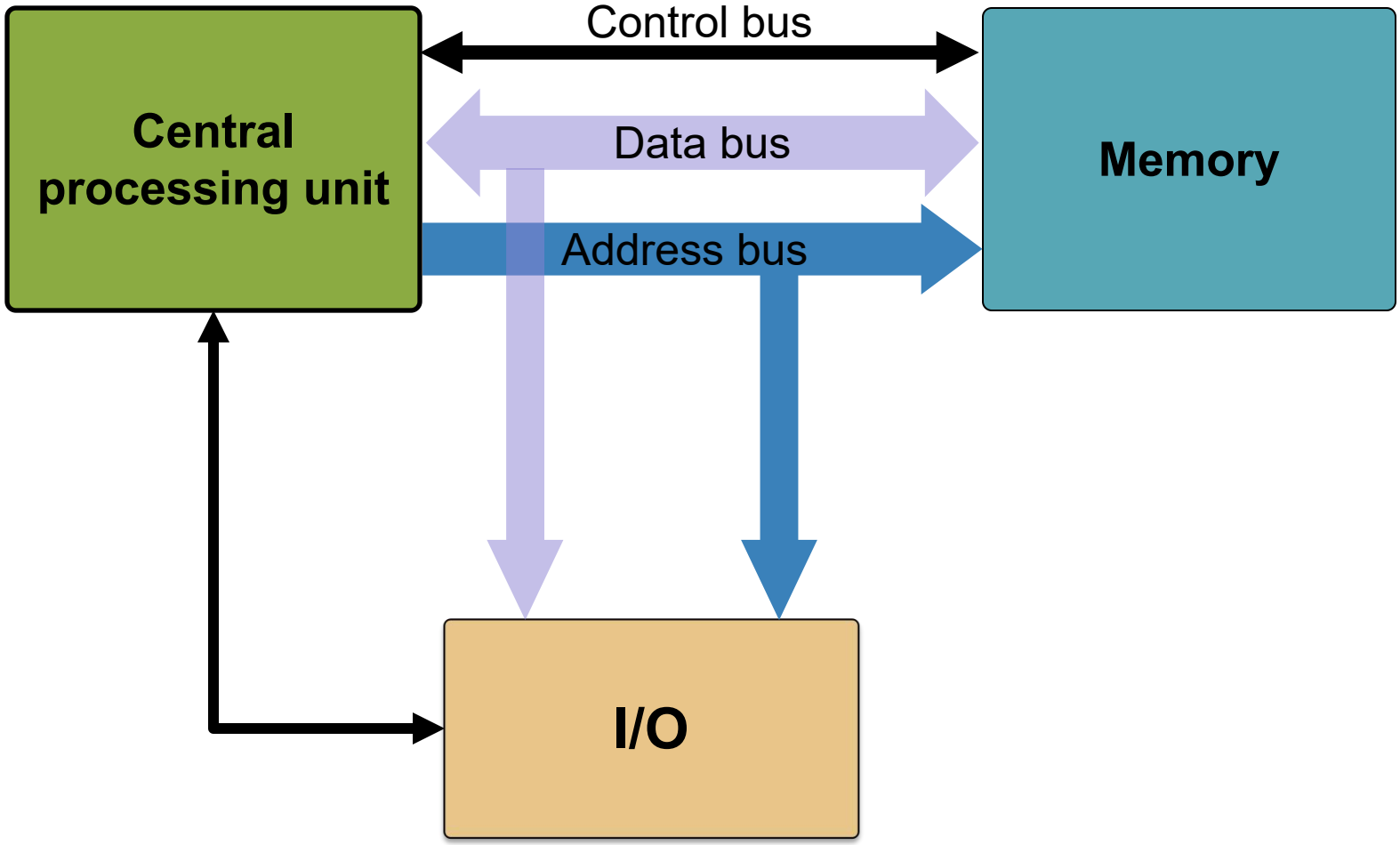
Spring 2019

Engr. Taufique-ur-Rehman <taufique.rehman@seecs.edu.pk>
Muhammad Imran Abeel <imran.abeel@seecs.edu.pk>

Outlines

- Input/output Addressing
  - Port Based I/O
  - Bus Based I/O
    - Memory Mapped I/O
    - Standard/Isolated I/O
- I/O Instructions

A System Diagram



Computer System Block Diagram

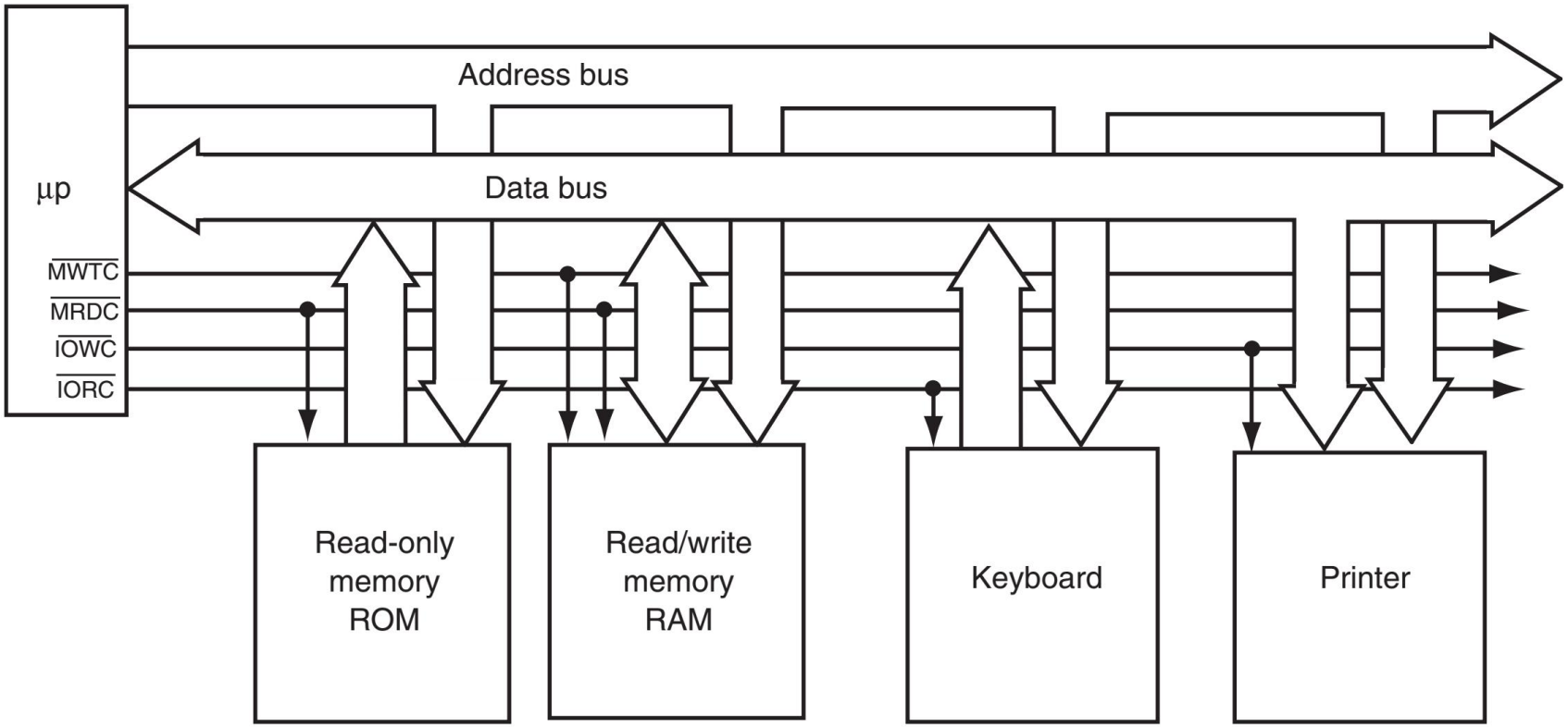


FIGURE 1–12 The block diagram of a computer system showing the address, data, and control bus structure.

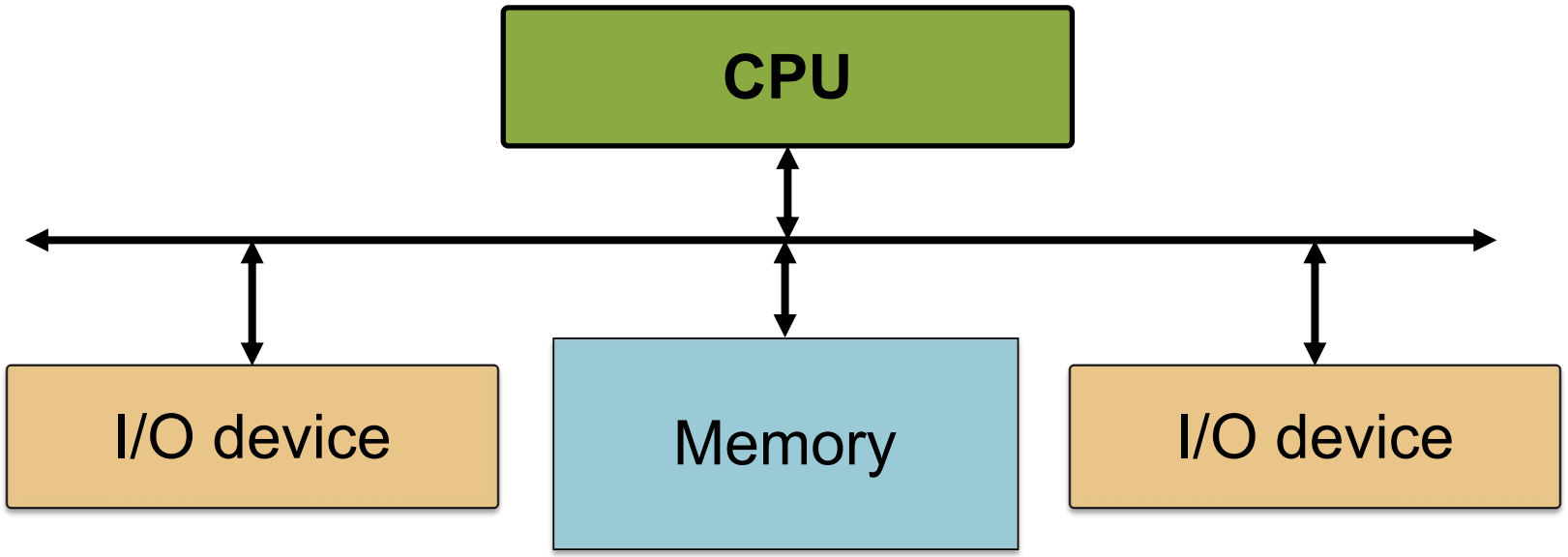
I/O Addressing

- Every Microprocessor possesses certain pins to communicate with other devices

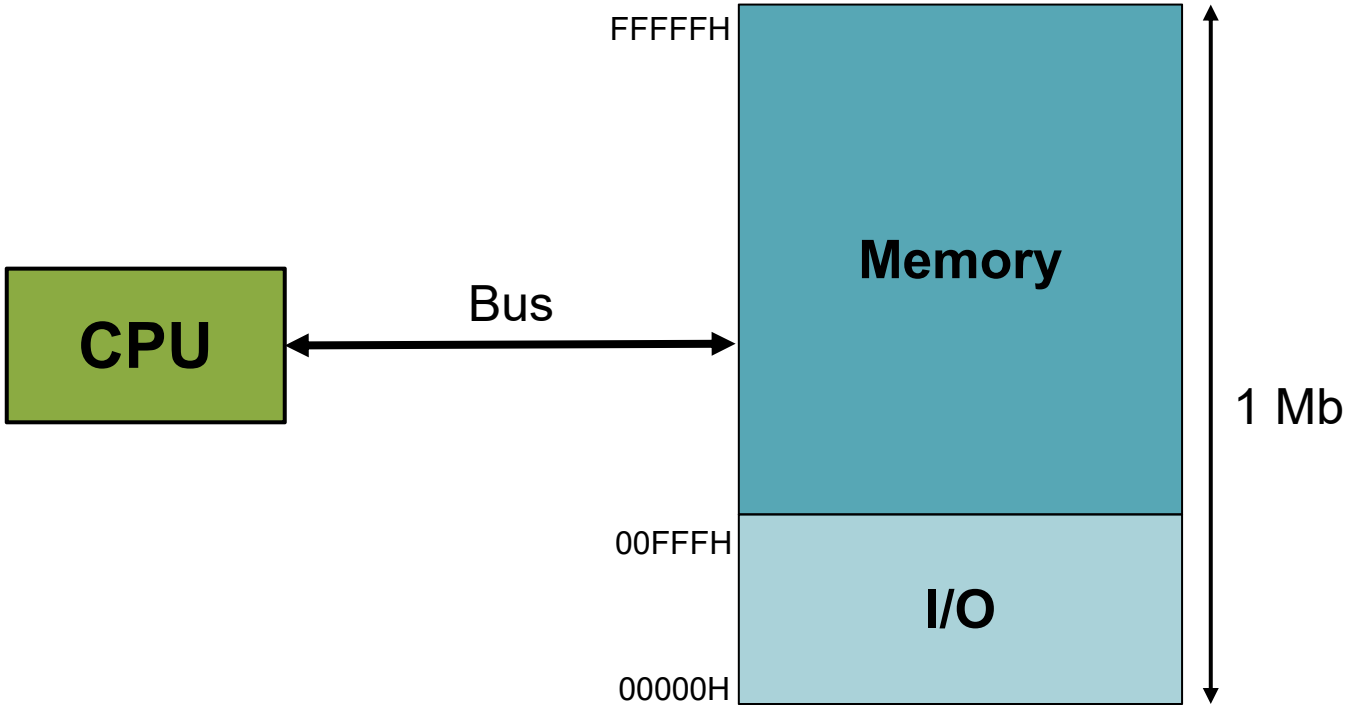
Port Based I/O	Bus Based I/O
Processor has one or more N ports	Shared communication channel for all devices
Processor access the ports just like a register	Communication protocol is built into the processor
	Single instruction is required to complete one process

Buses

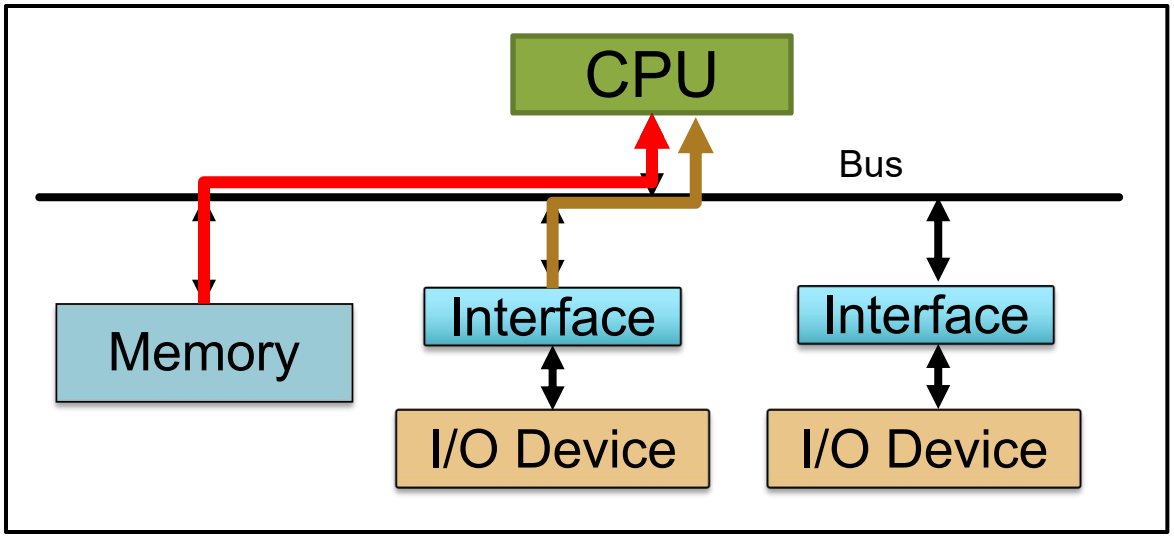
- Shared communication link to connect multiple devices

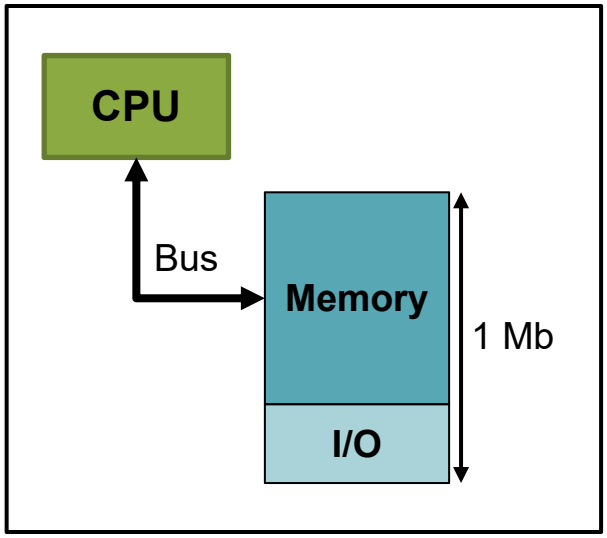


Memory Mapped I/O



Memory Mapped I/O

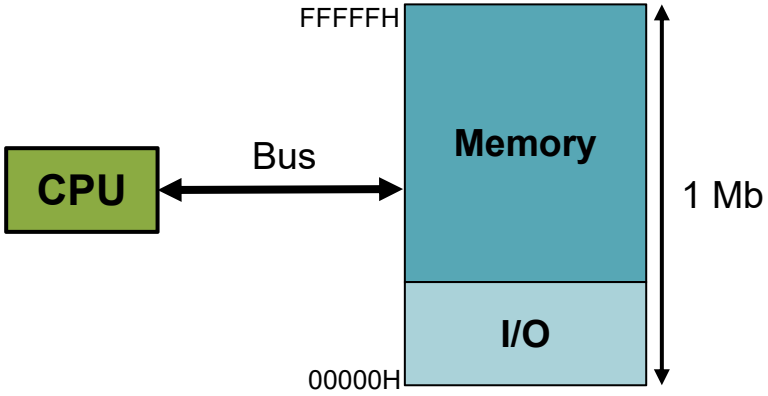




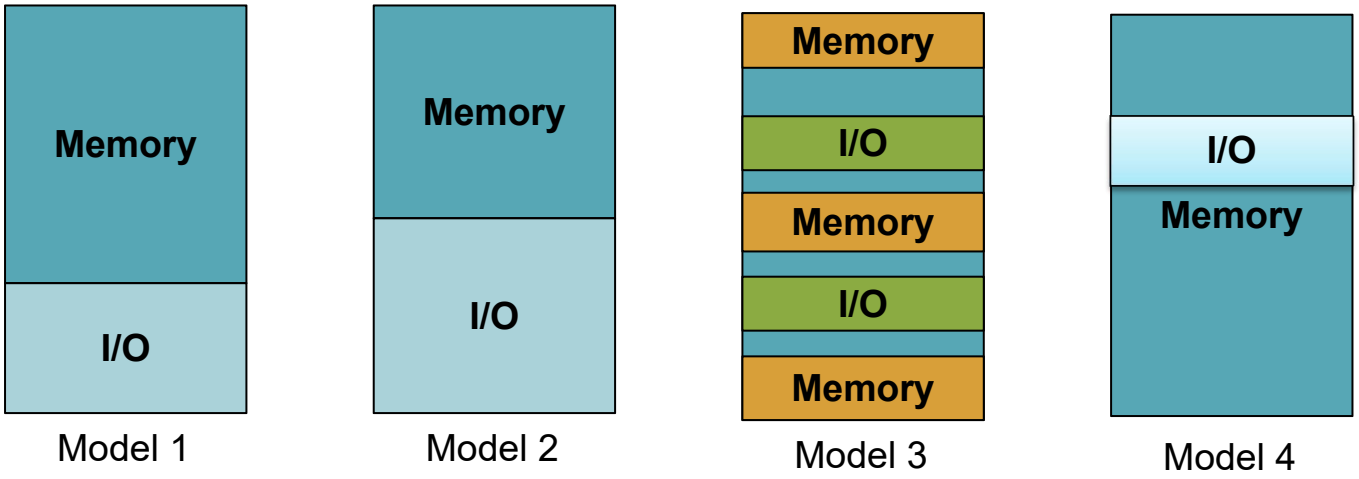
- IO/Memory shares the same bus.
- Devices can be accessed in the same way as we access memory in general scenario.

Memory Mapped I/O

- ***Bus has 20-bit address***
  - Lower 512K Addresses are mapped for Memory
  - Higher 512K Addresses correspond to Peripherals

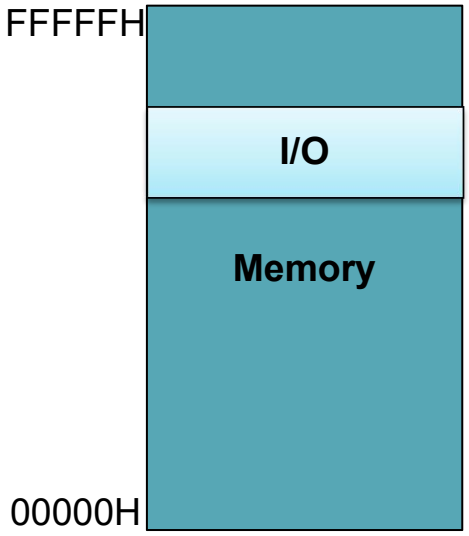


- ***Shared memory models***



Memory Mapped I/O Address (8088/8086)

Memory + I/O



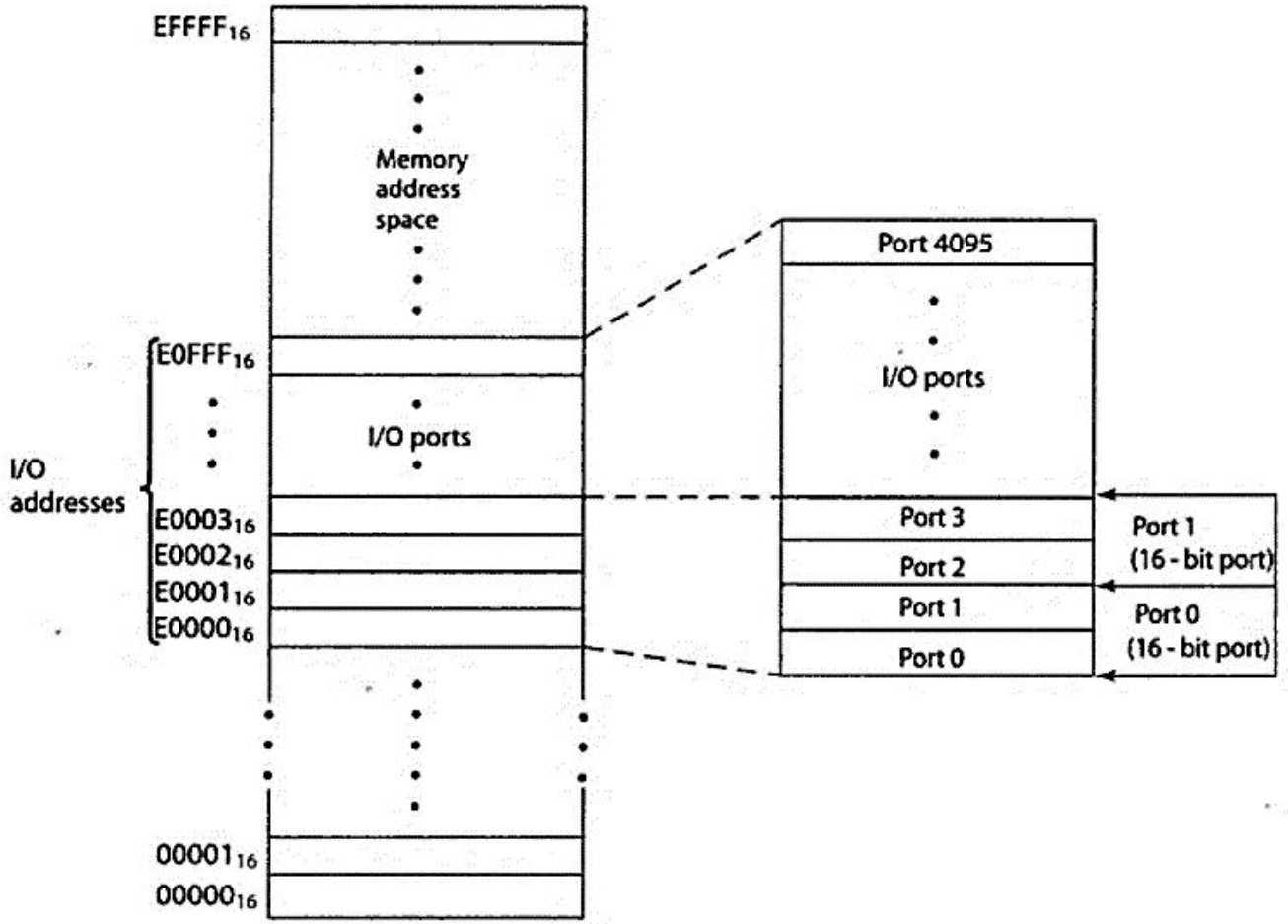
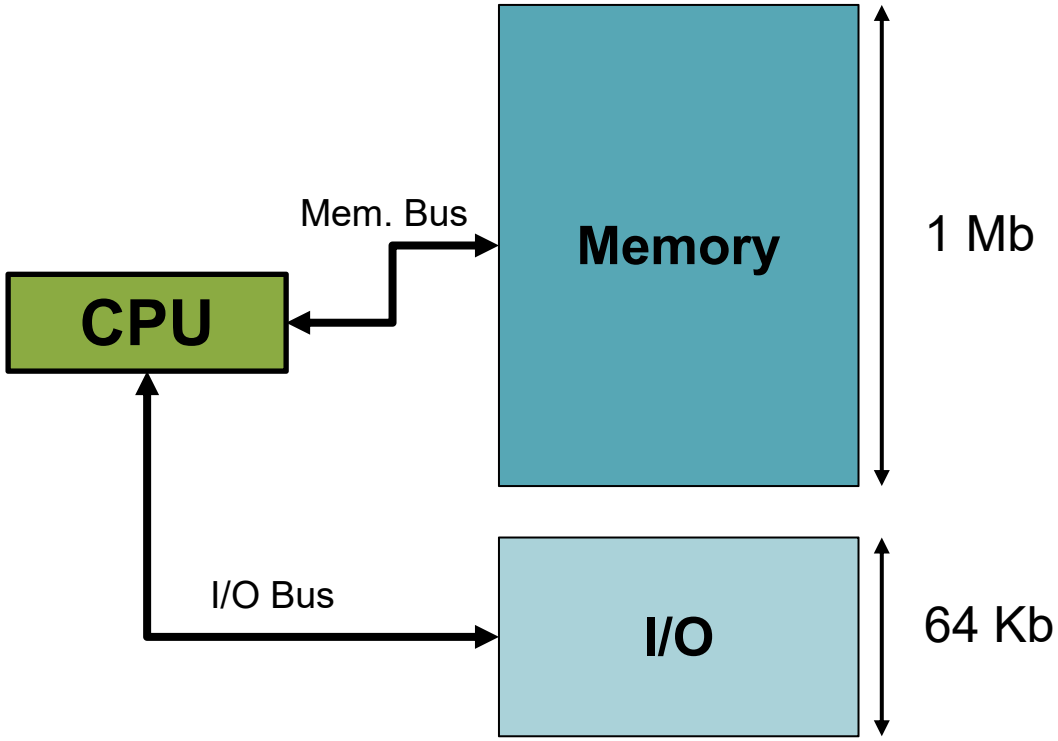


Figure 8–40 (a) Isolated I/O ports. (b) Memory-mapped I/O ports.

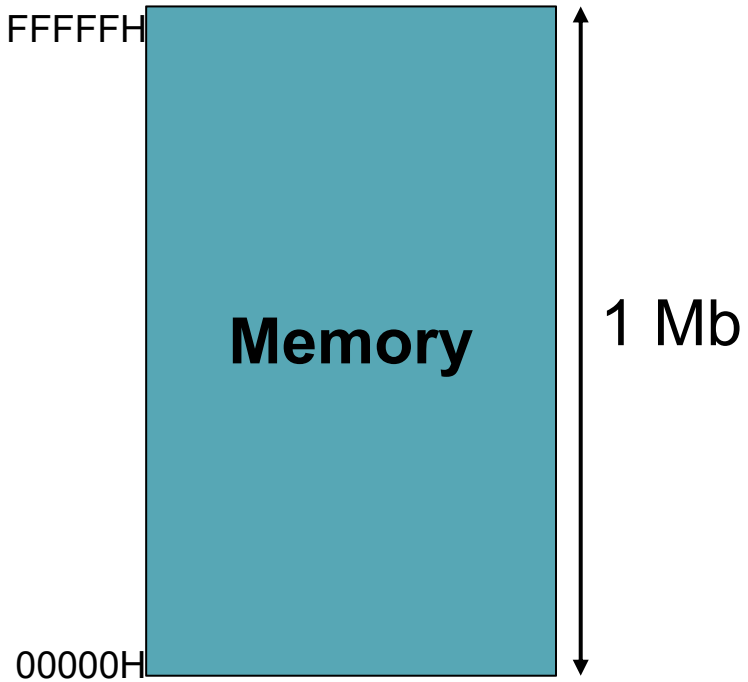
I/O Mapped I/O Addresses



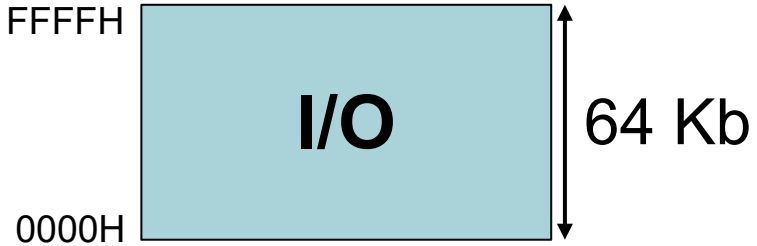
- **Note: I/O Mapped I/O is also known as Standard I/O or Isolated I/O**

Standard I/O Addressing

- For 16-bit processor, standard I/O Address Space is:



2^{20} = 1Mbytes × 8



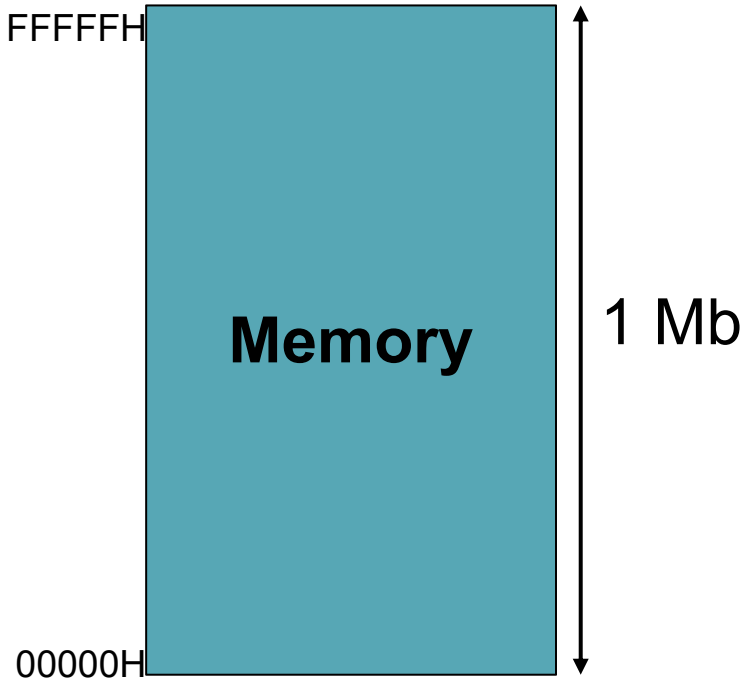
2^{16} = 64Kbytes × 8

Standard I/O Addressing

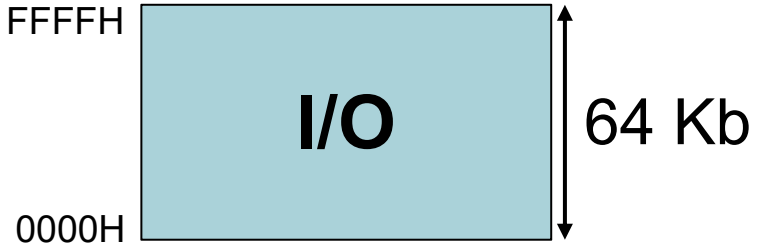
- For 16-bit processor, standard I/O address space is:

When $IO/\overline{M}$ = 0

All 1Mbytes contains the memory address



2^{20} = 1Mbytes × 8



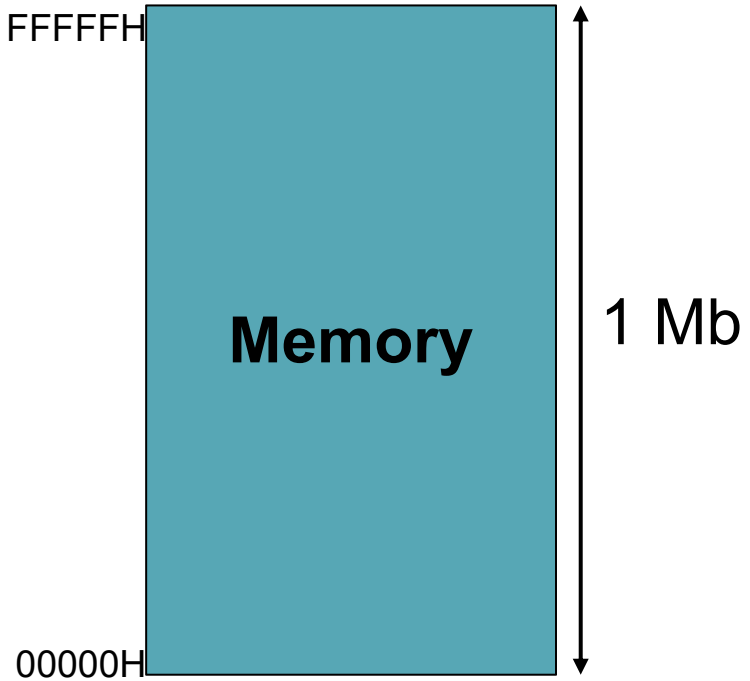
2^{16} = 64Kbytes × 8

Standard I/O Addressing

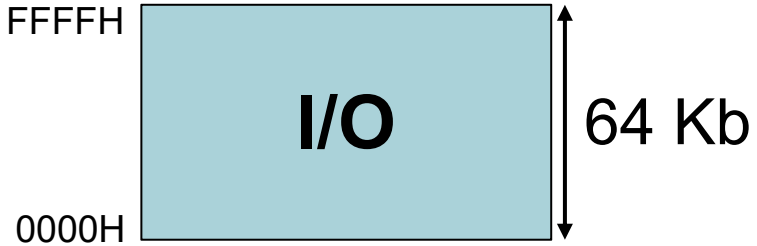
- For 16-bit processor, standard I/O address space is;

When $IO/\overline{M}$ = 1

64kbytes out of 1MB are for I/O address.

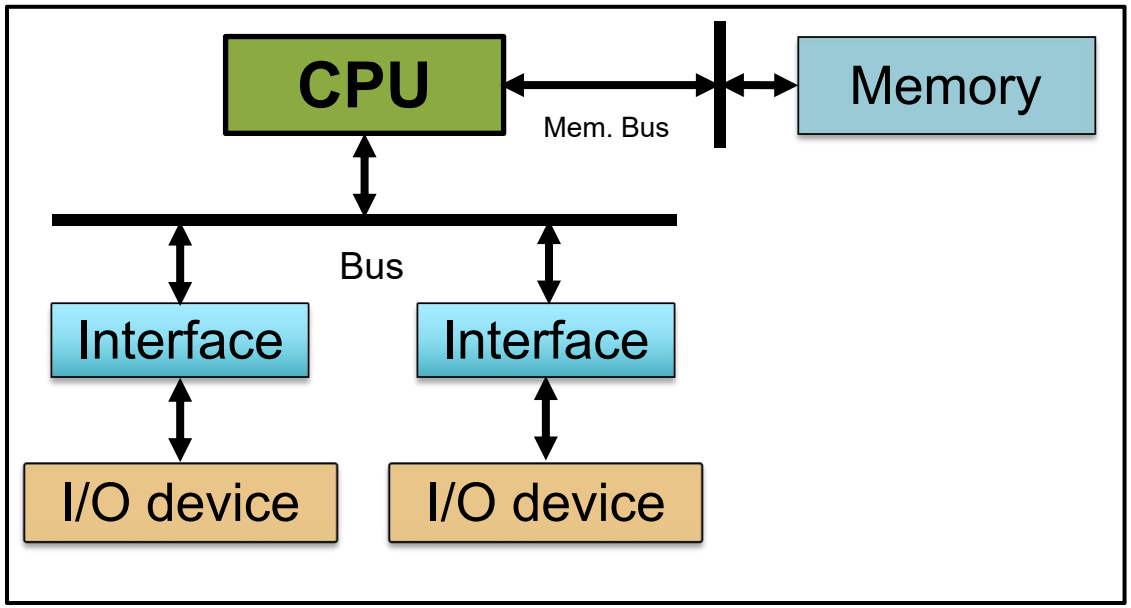


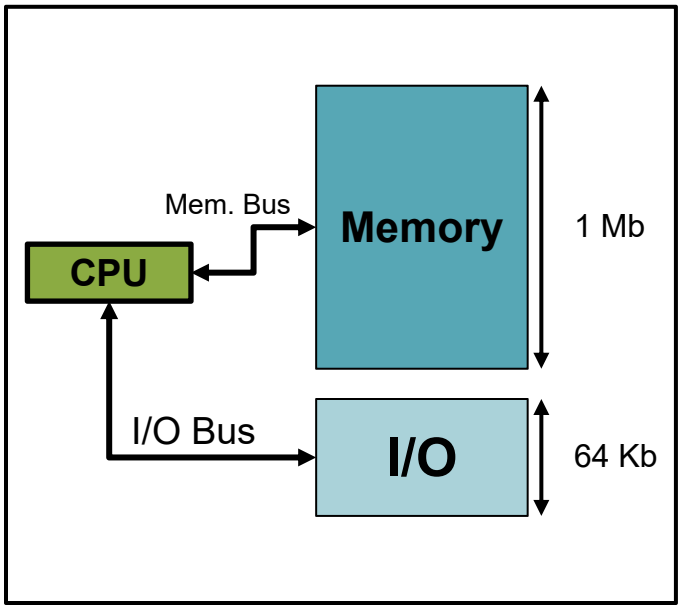
2^{20} = 1Mbytes × 8



2^{16} = 64Kbytes × 8

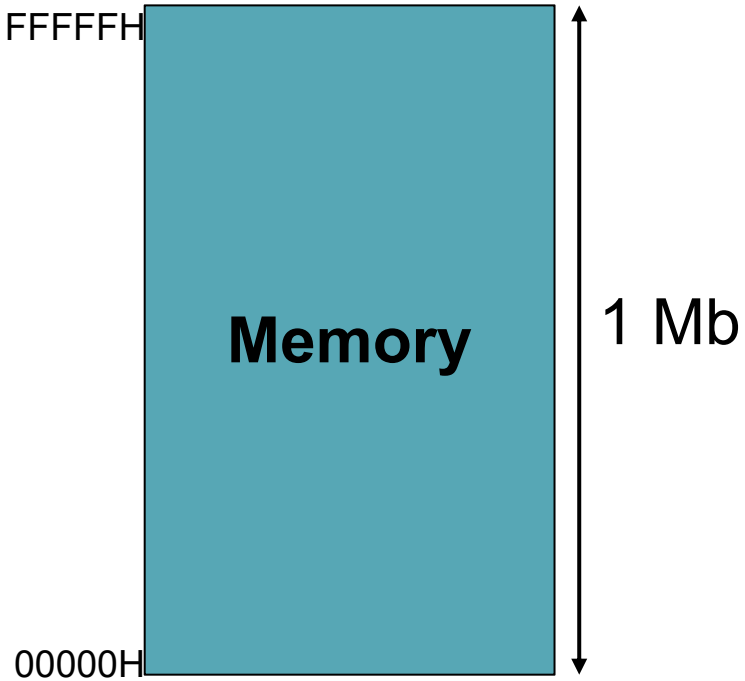
Standard I/O Addressing



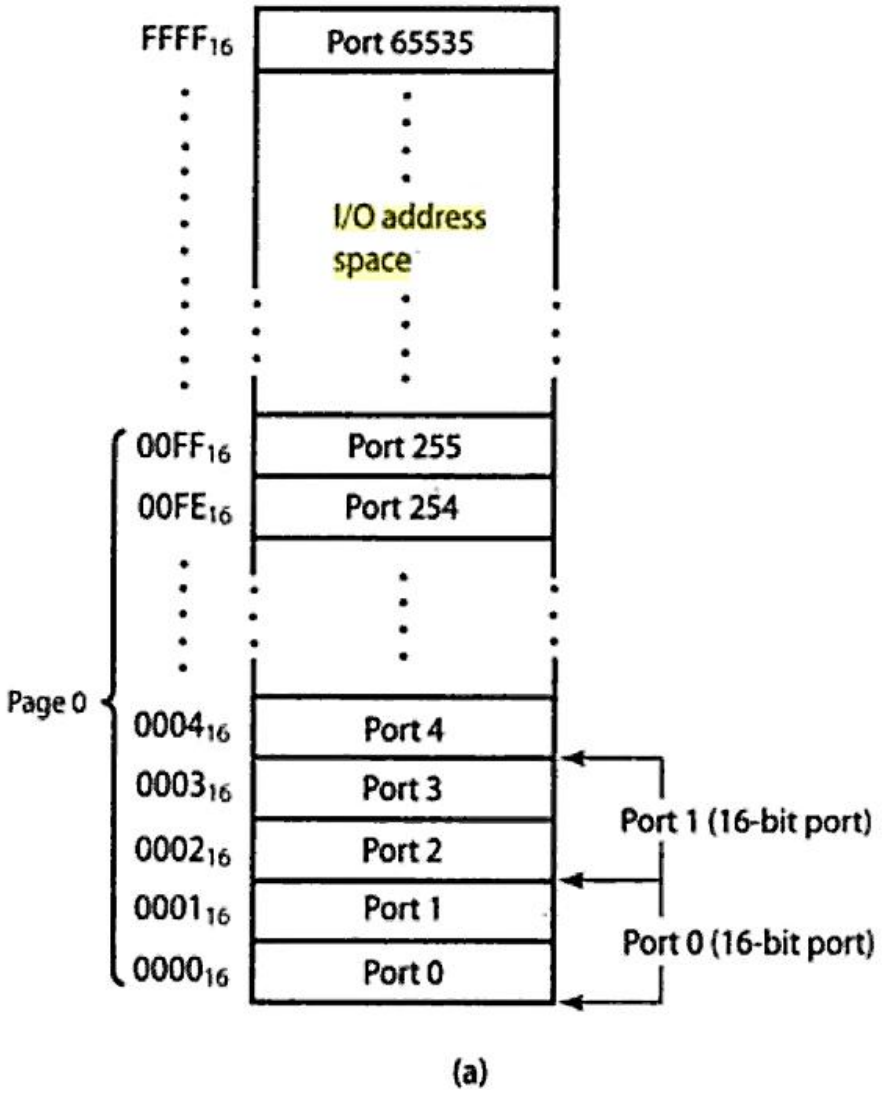


- Dedicated Address Space for Memory & IO
- Additional Pins on Bus indicate that either Memory or I/O is being accessed

Standard I/O Addressing (8088/8086)



2^{20} = 1Mbytes × 8



Memory vs I/O Mapped I/O

Memory Mapped I/O

Peripheral Registers and Memory occupy same Address Space

I/O Mapped I/O

$IO/\overline{M}$ signal points towards Memory or Peripherals

Memory vs I/O Mapped I/O

Memory Mapped I/O

Peripheral Registers and Memory occupy same Address Space

Bus Address is 16-bits

- Lower 32K Address Space for Mem
- Higher 32K Address Space for I/O

I/O Mapped I/O

$IO/\overline{M}$ signal points towards Memory or Peripherals

Bus Address is 16-bits

- 64K Address Space is available for Mem
- Also 64K Addresses are available for I/O

Memory vs I/O Mapped I/O

Memory Mapped I/O

Peripheral Registers and Memory occupy same Address Space

Bus Address is 16-bits

- Lower 32K Address Space for Mem
- Higher 32K Address Space for I/O

MOV, ADD works for both Memory and I/O

I/O Mapped I/O

$IO/\overline{M}$ signal points towards Memory or Peripherals

Bus Address is 16-bits

- 64K Address Space is available for Mem
- Also 64K Addresses are available for I/O

Special Instructions like IN, OUT, INS and OUTS are required for I/O

Memory vs I/O Mapped I/O

Memory Mapped I/O

Peripheral Registers and Memory occupy same Address Space

Bus Address is 16-bits

- Lower 32K Address Space for Mem
- Higher 32K Address Space for I/O

MOV, ADD works for both Memory and I/O

Example:

$$ADD\ A,\ B$$

I/O Mapped I/O

$IO/\bar{M}$ signal points towards Memory or Peripherals

Bus Address is 16-bits

- 64K Address Space is available for Mem
- Also 64K Addresses are available for I/O

Special Instructions like IN, OUT, INS and OUTS are required for I/O

Example:

```
IN R0,A
IN R1,B
ADD R0,R1
OUT A,R0
```

Standard I/O

I/O Devices

- Switches, LEDs, Keyboard, Mouse etc.

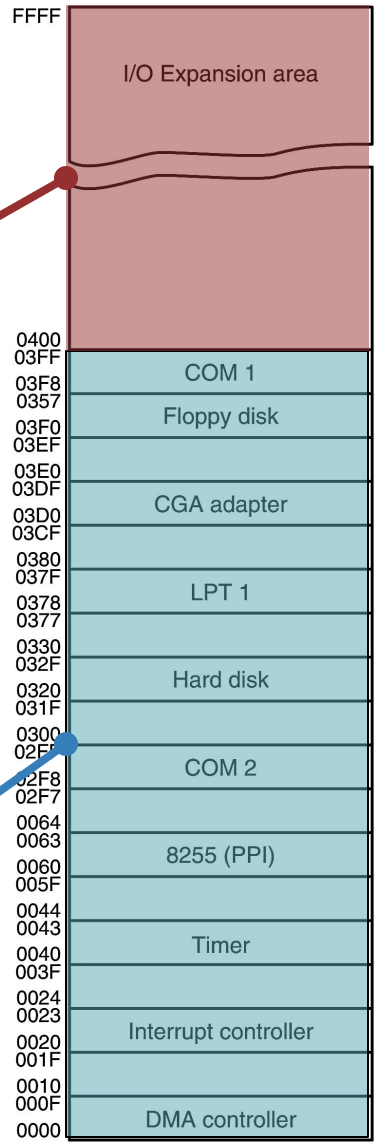
I/O Interface

- Receives Address from CPU
- Decodes the I/O Address for Port Number
- Latch the Output Data
- Sample Input Data
- Synchronize Data Transfer
- Voltage Level translation to operate I/O Devices

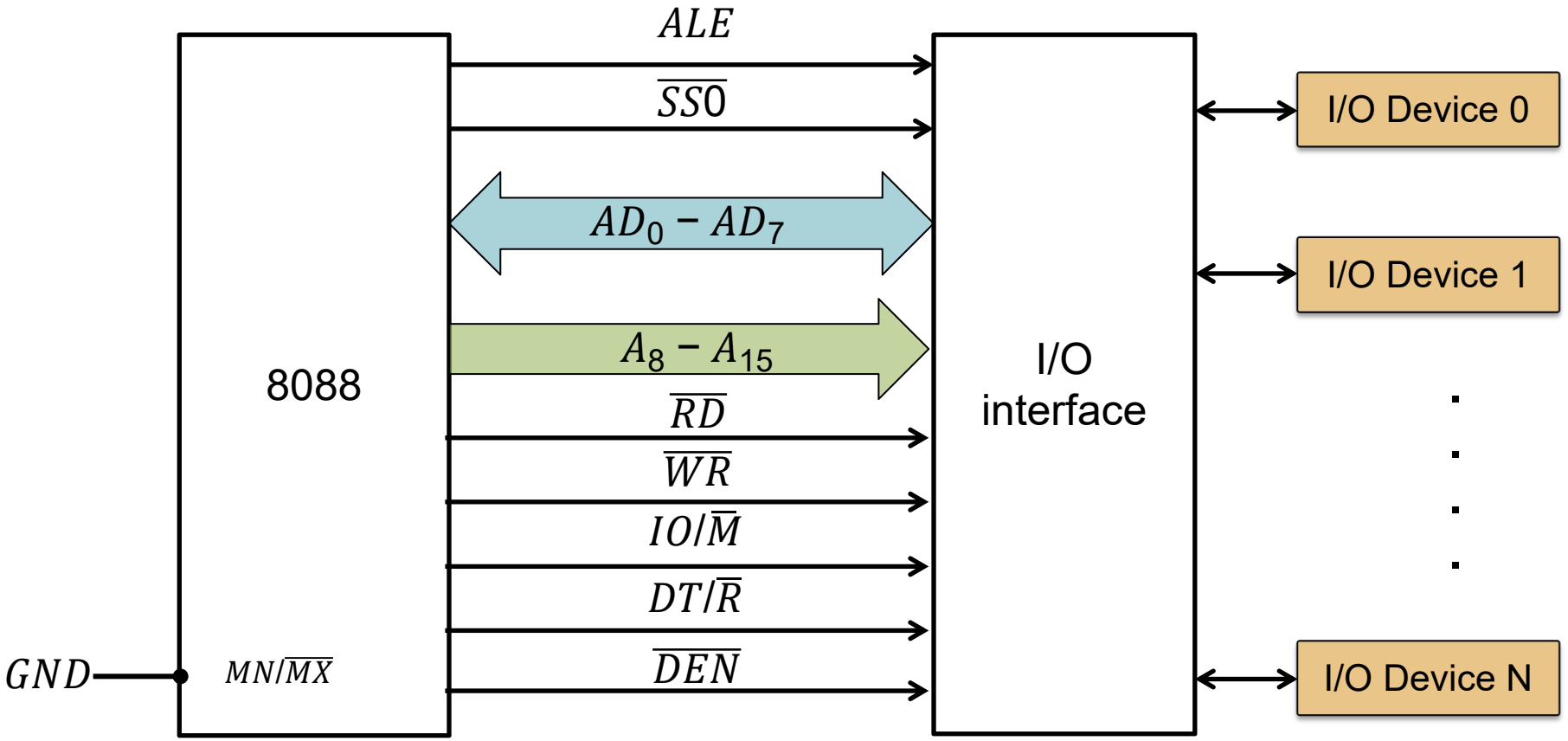
I/O Map of Personal Computer

Available for User Specific Applications

Reserved for Computer System (0000H-0400H)

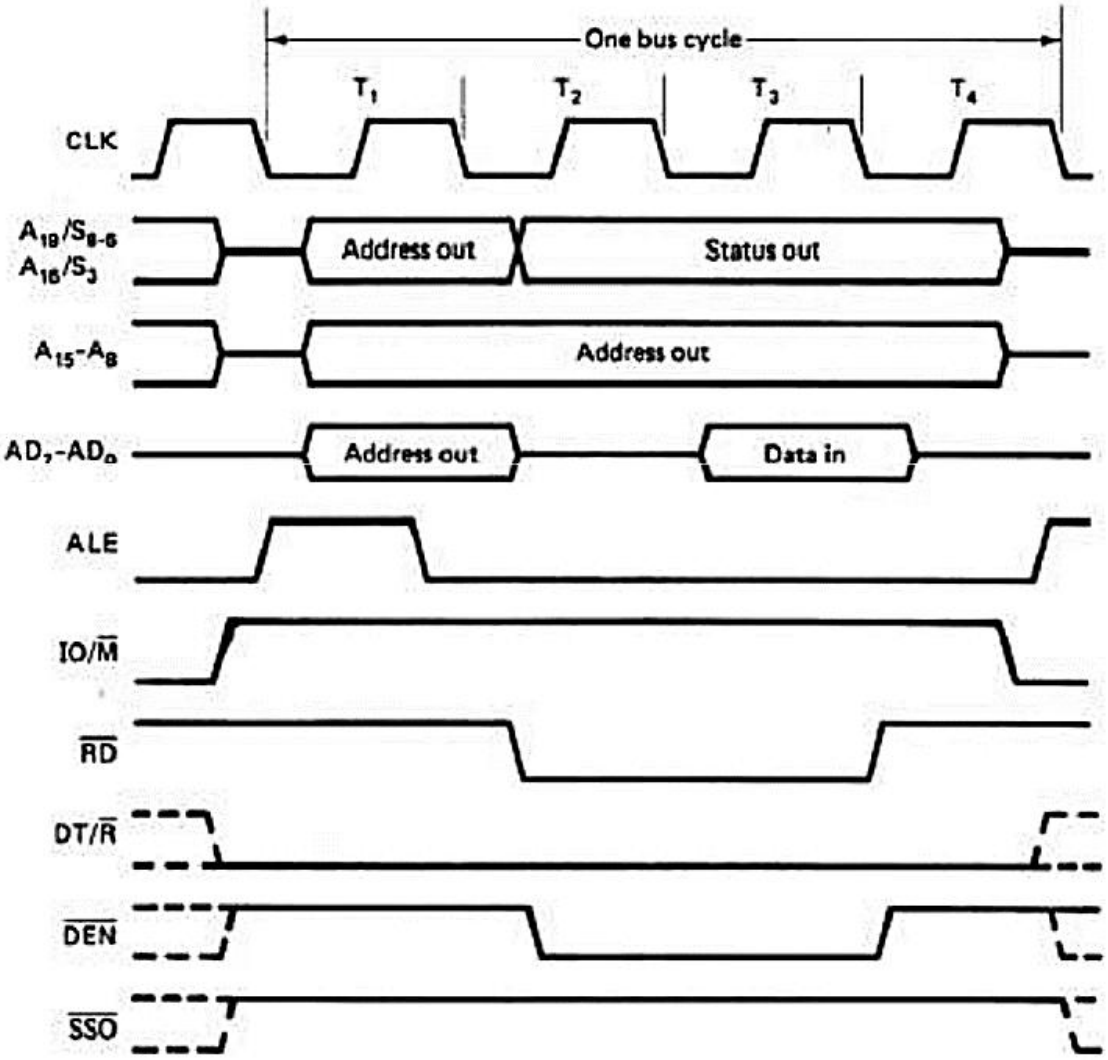


Minimum Mode Interface

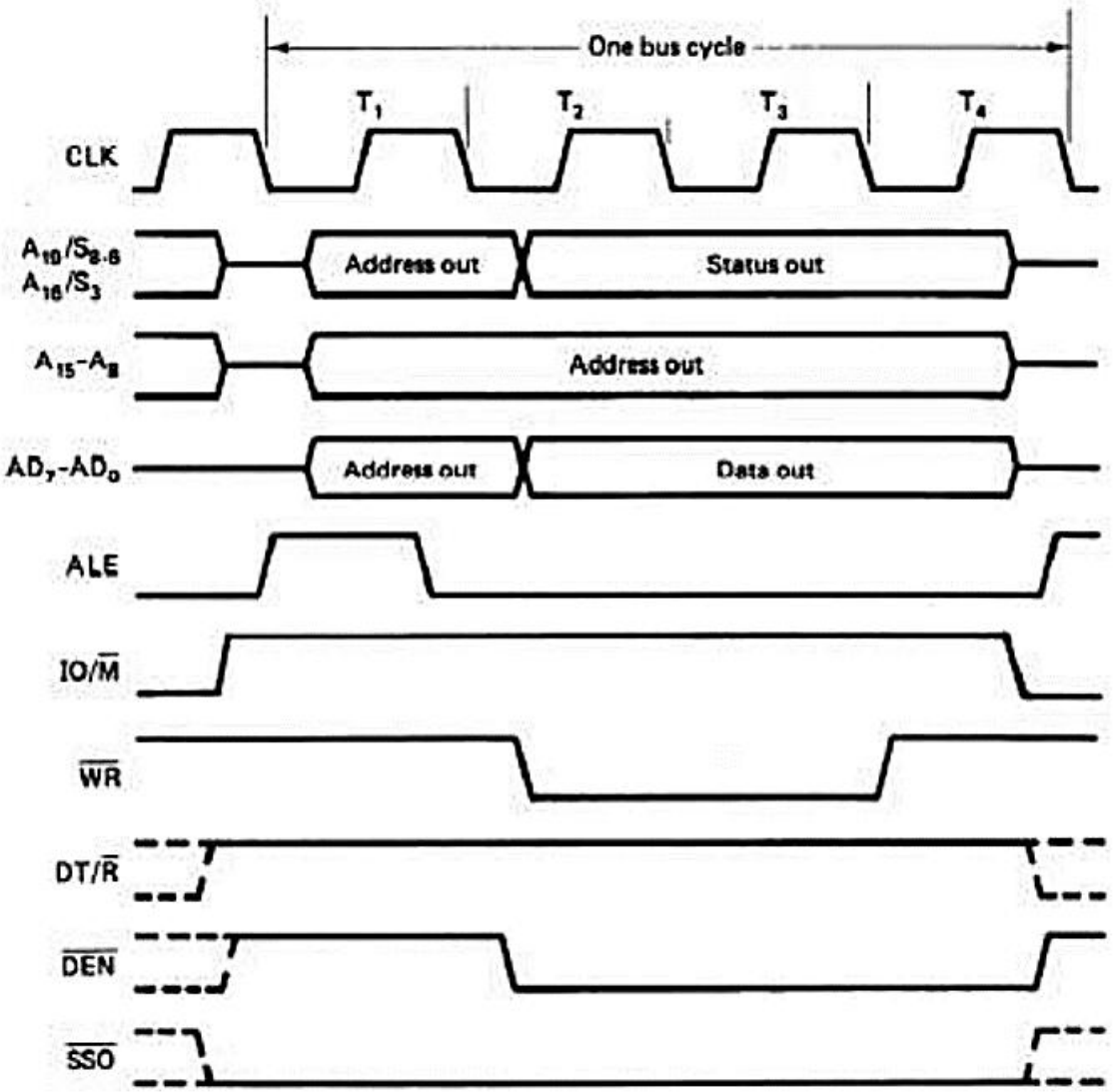


- Signals for I/O Read Cycle??

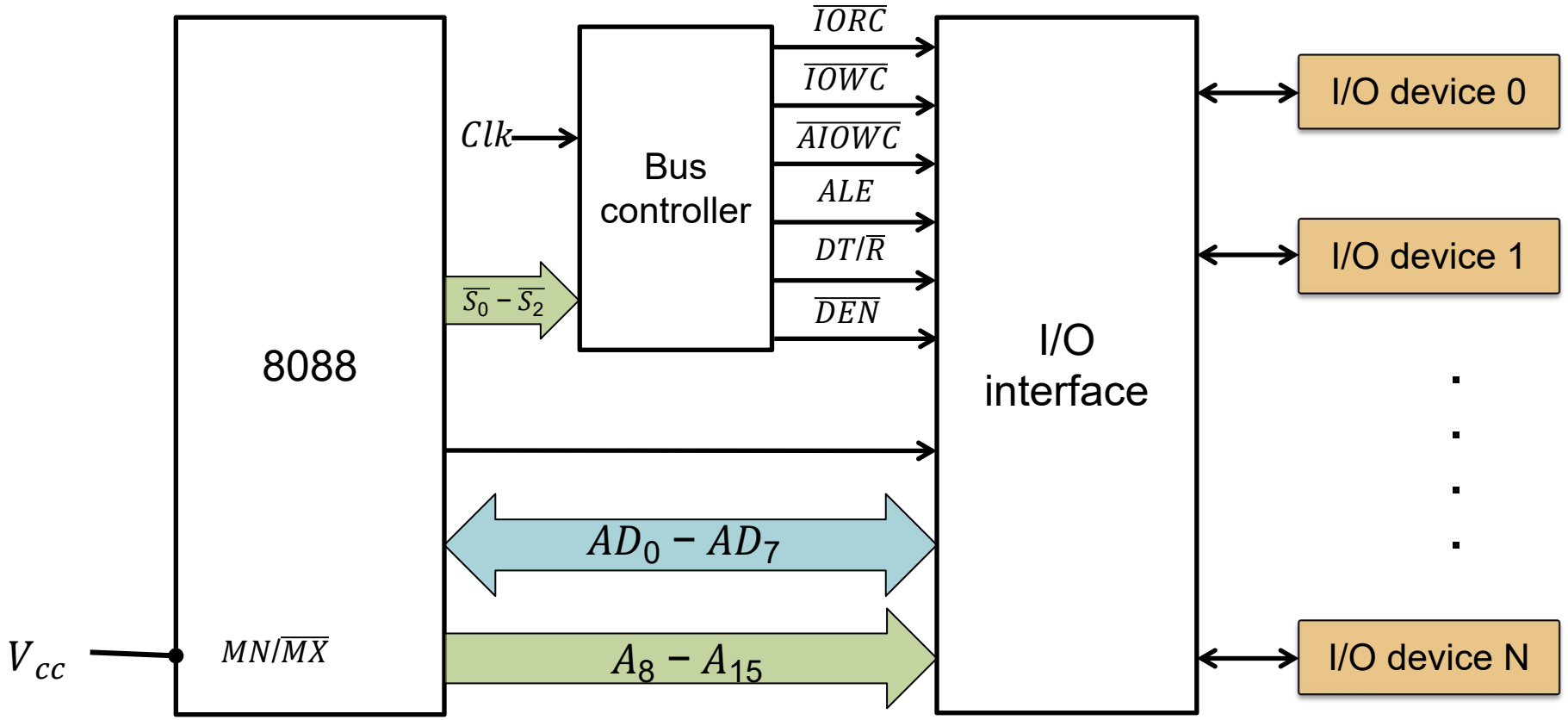
Bus Read Cycle



Bus Read Cycle



Maximum Mode Interface



I/O Bus Cycle Status Code

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	*Function*
0	0	0	Interrupt acknowledge
0	0	1	I/O read
0	1	0	I/O write
0	1	1	Halt
1	0	0	Opcode fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive

I/O Instructions

Fixed Address

Byte: $IN\ AL, p8$

Word: $IN\ AX, p8$

DWord: $IN\ EAX, p8$

Can only access first 256 locations using fixed address

Variable Address

Byte: $IN\ AL, DX$

Word: $IN\ AX, DX$

Can access all 64K locations using variable Address

I/O Instructions

Instruction	Format	Meaning	Operation
IN	IN Acc, Port	Input direct	Acc ← Port
	IN Acc, DX	Input indirect	Acc ← ((DX))
OUT	OUT Port, Acc	Output direct	Port ← Acc
	OUT DX, Acc	Output indirect	((DX)) ← Acc

Memory Mapped vs Standard I/O

Example: (Memory Mapped I/O)

- $ADD\ A, B$

Example: (Standard Mapped I/O)

Memory Mapped vs Standard I/O

Example: (Memory Mapped I/O)

- $ADD\ A, B$

Example: (Standard Mapped I/O)

- $IN\ R0, A$
- $IN\ R1, B$
- $ADD\ R0, R1$
- $OUT\ A, R0$

I/O Instructions

Reading Assignment

Example

– Write a sequence of instructions that will output the data **FFh** to a byte wide output port at address **00ABh** of the I/O address space

I/O Instructions

Reading Assignment

Example

– Write a sequence of instructions that will output the data **FFh** to a byte wide output port at address **00ABh** of the I/O address space

Solution:

- $MOV\ AL,\ FFH$
- $OUT\ AB, AL$

I/O Instructions

Reading Assignment

Example

– Write a series of instructions that will output **FFh** to an output port located at address **B000h** of I/O address space.

I/O Instructions

Reading Assignment

Example

– Write a series of instructions that will output **FFh** to an output port located at address **B000h** of I/O address space

Solution:

– $MOV\ DX,\ B000H$

– $MOV\ AL, FFh$

– $OUT\ DX,\ AL$

I/O Instructions

Reading Assignment

Example

– Data are to be read in from 2 bytes wide input ports at Addresses **00AAh** and **00A9h** and then output as a WORD to output port at Address **B000h**

I/O Instructions

Reading Assignment

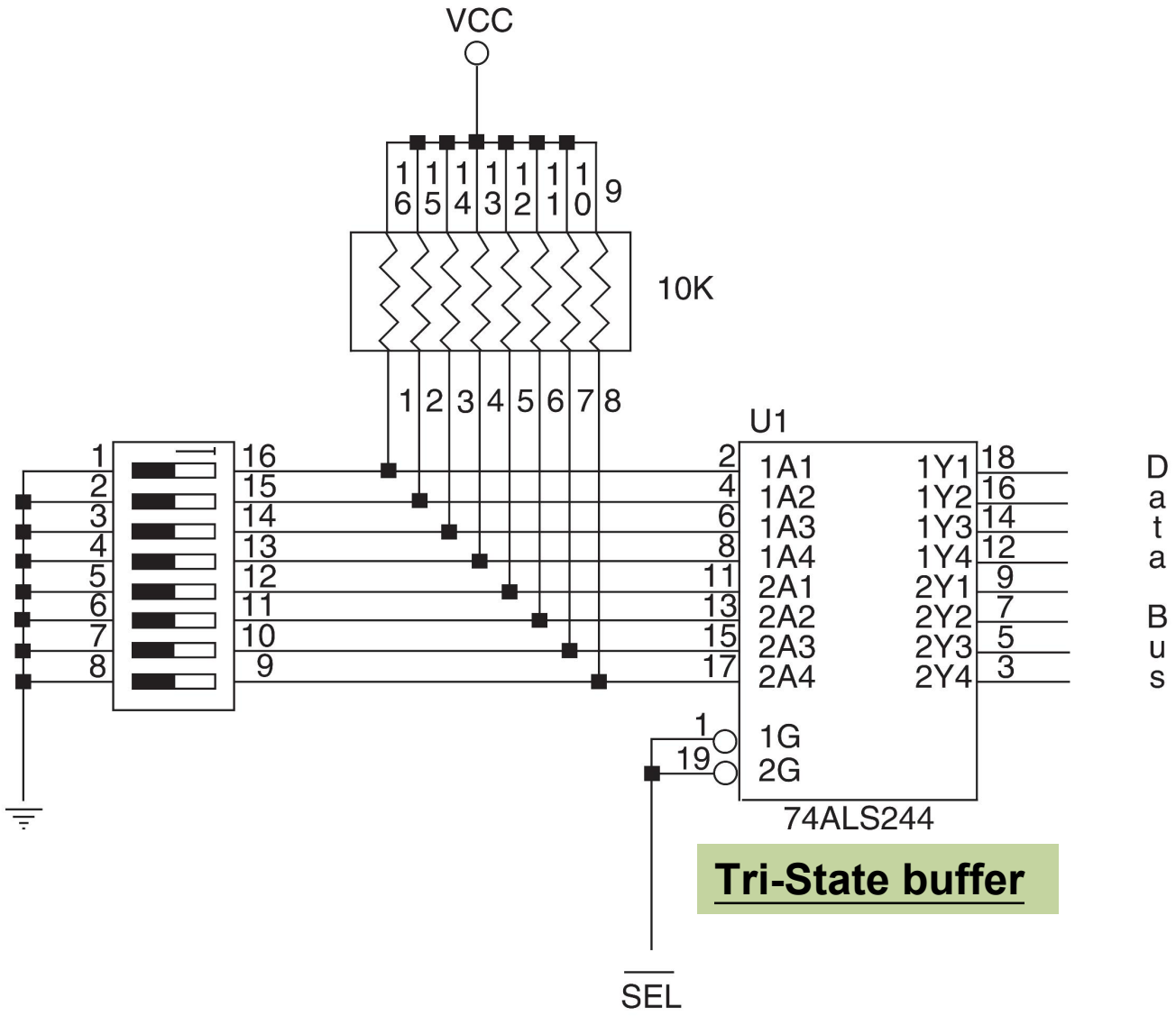
Example

– Data are to be read in from 2 bytes wide input ports at Addresses **00AAh** and **00A9h** and then output as a WORD to output port at Address **B000h**

Solution:

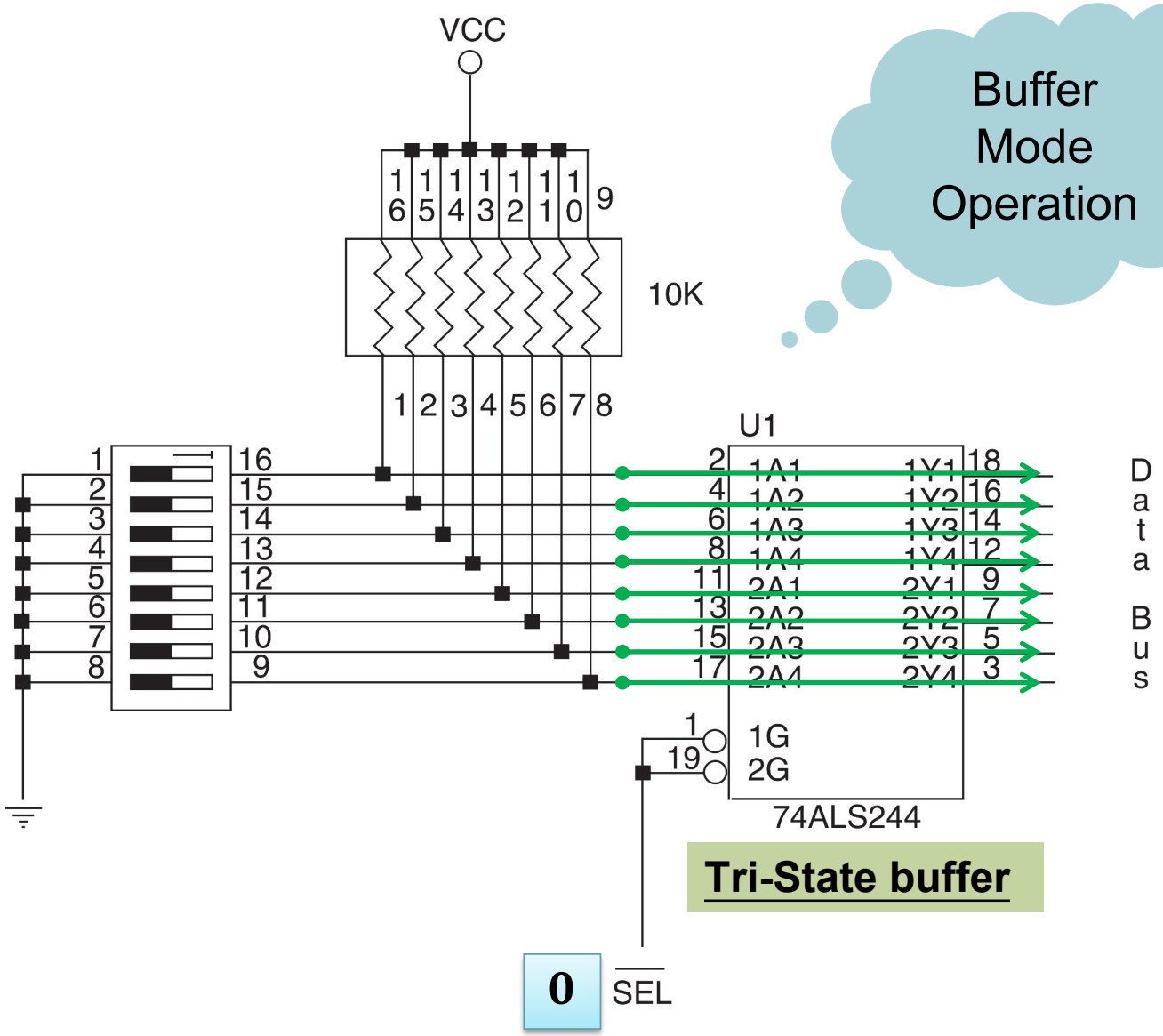
- IN AL, AAh
- MOV AH,AL
- IN AL,A9h
- MOV DX, B000h
- OUT DX,AX

Basic Input Interface

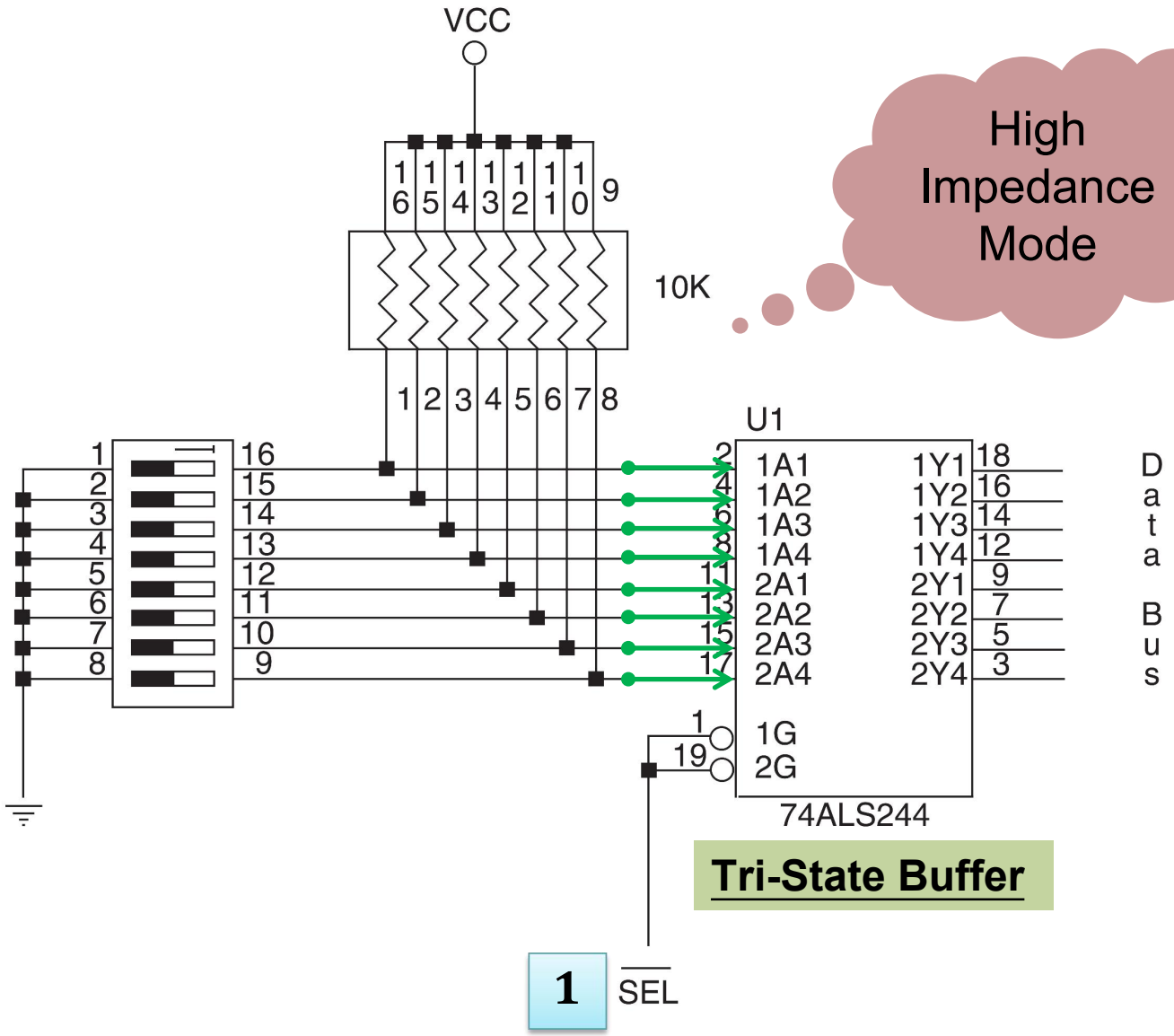


Tri-State buffer

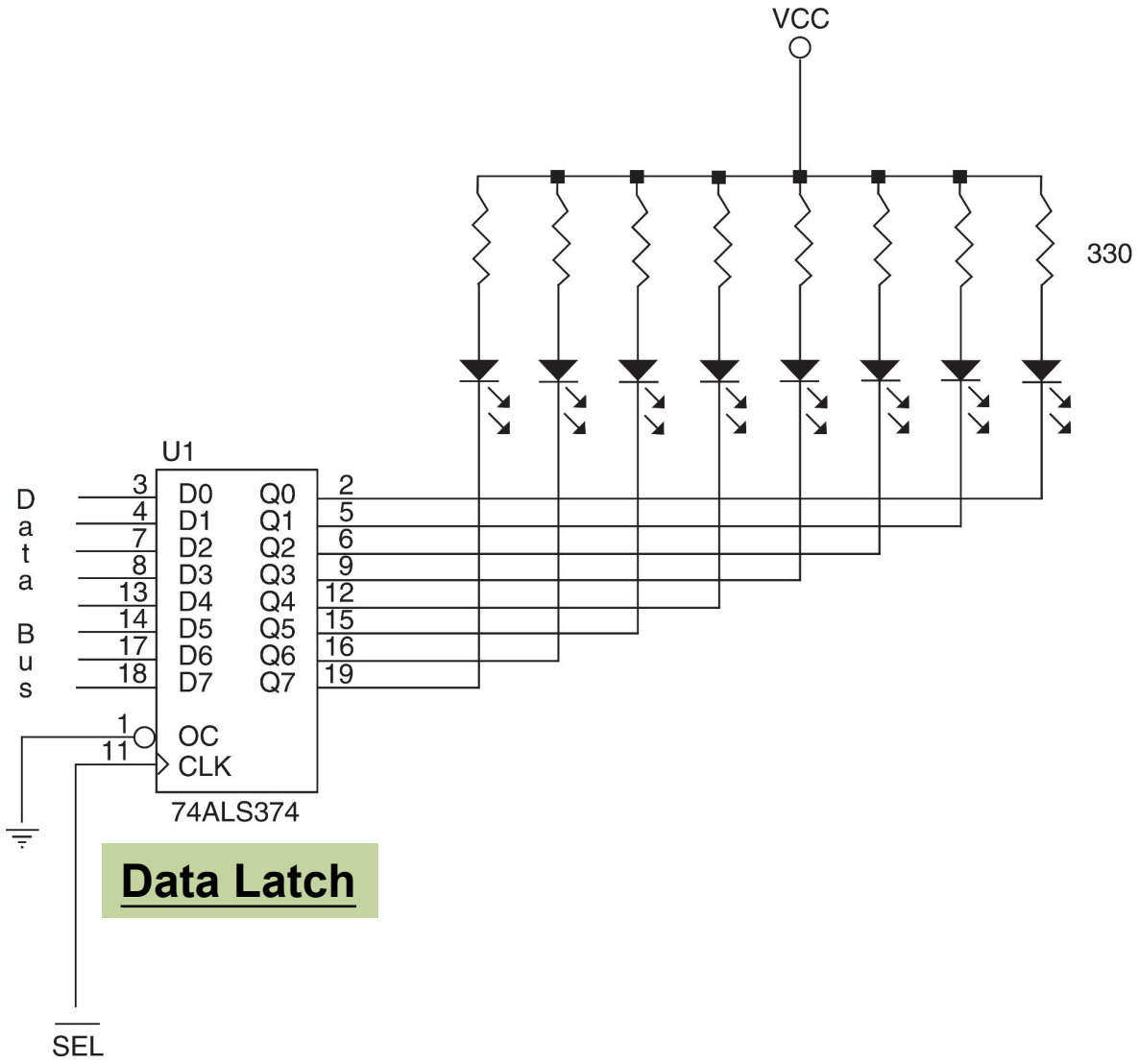
Basic Input Interface



Basic Input Interface



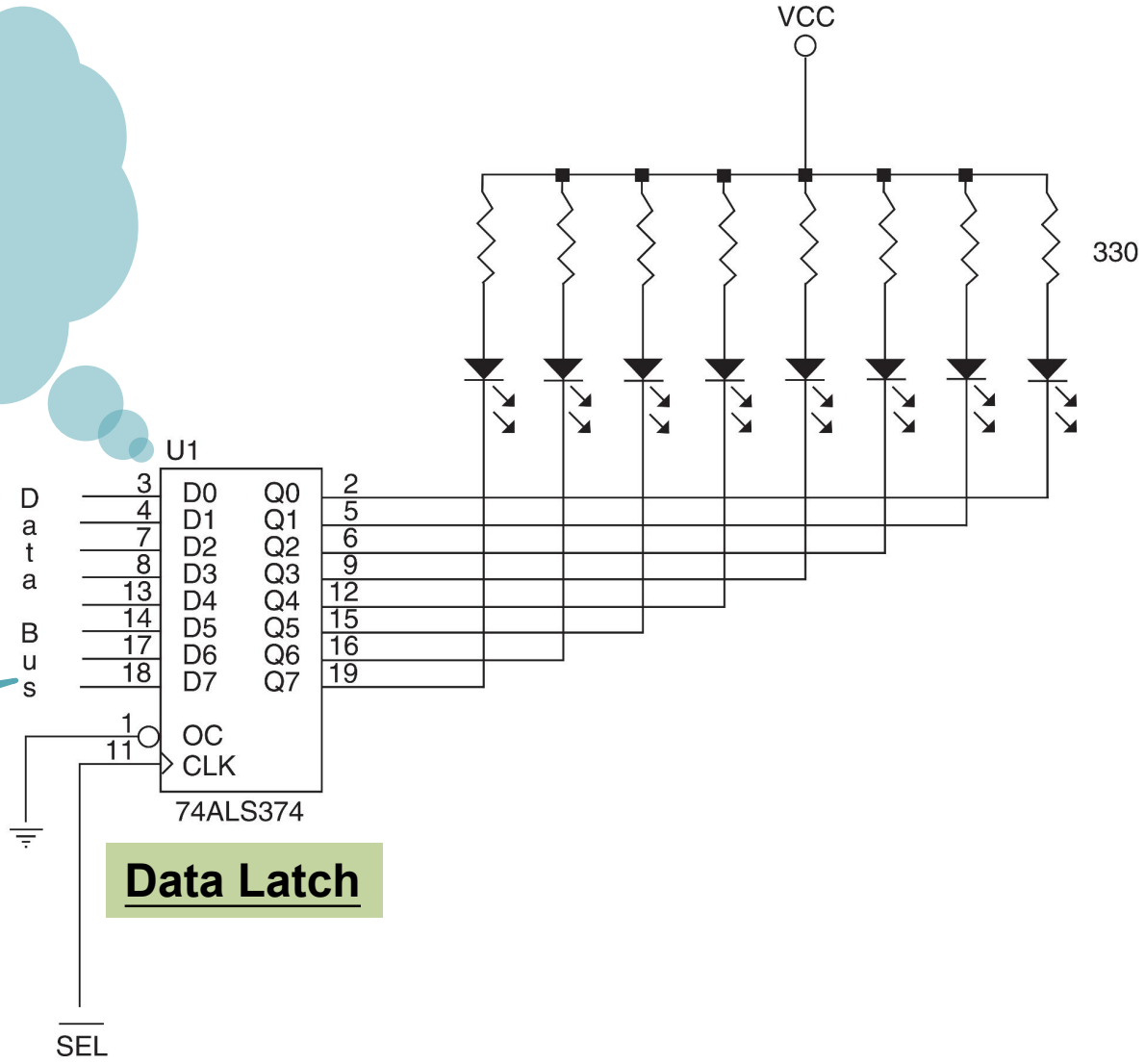
Basic Output Interface



Data Latch

Basic Output Interface

Latch is required to hold the output data for LED



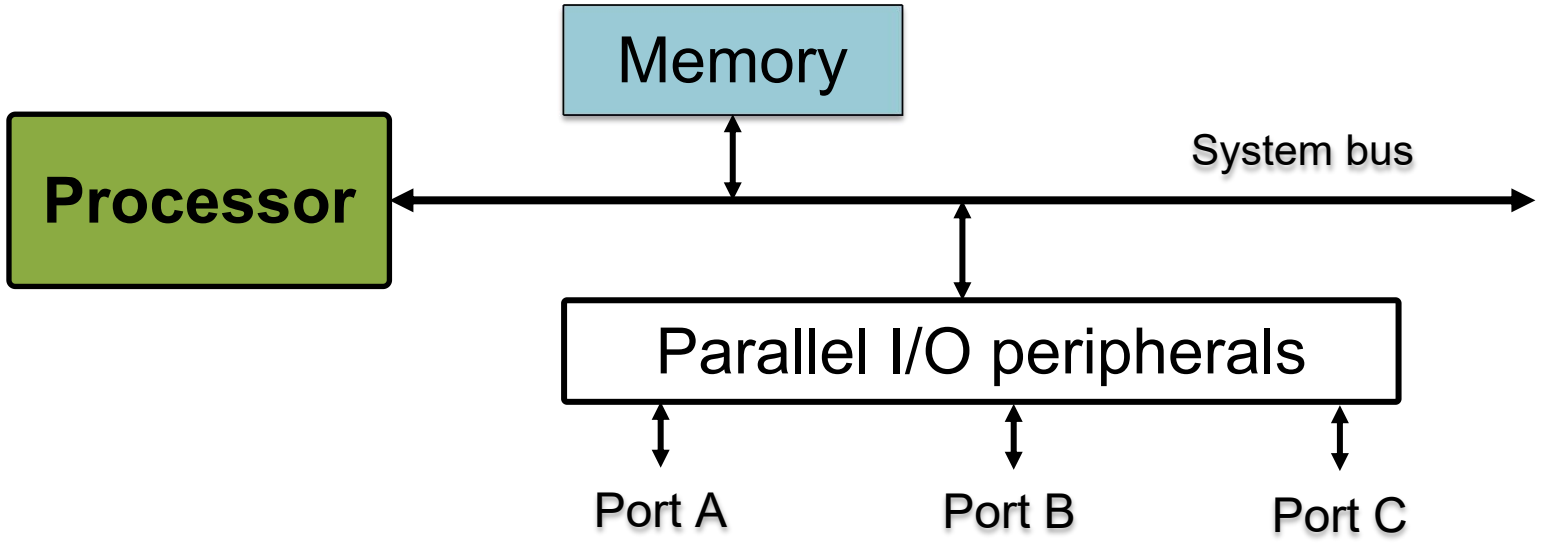
The microprocessor only provides data for 1 μs.

Data Latch

Compromises/ Extensions

Parallel I/O Peripherals:

- When processor only supports bus based I/O and parallel ports are required
- Each peripheral ports are connected to a register within peripheral that is read/written by processor

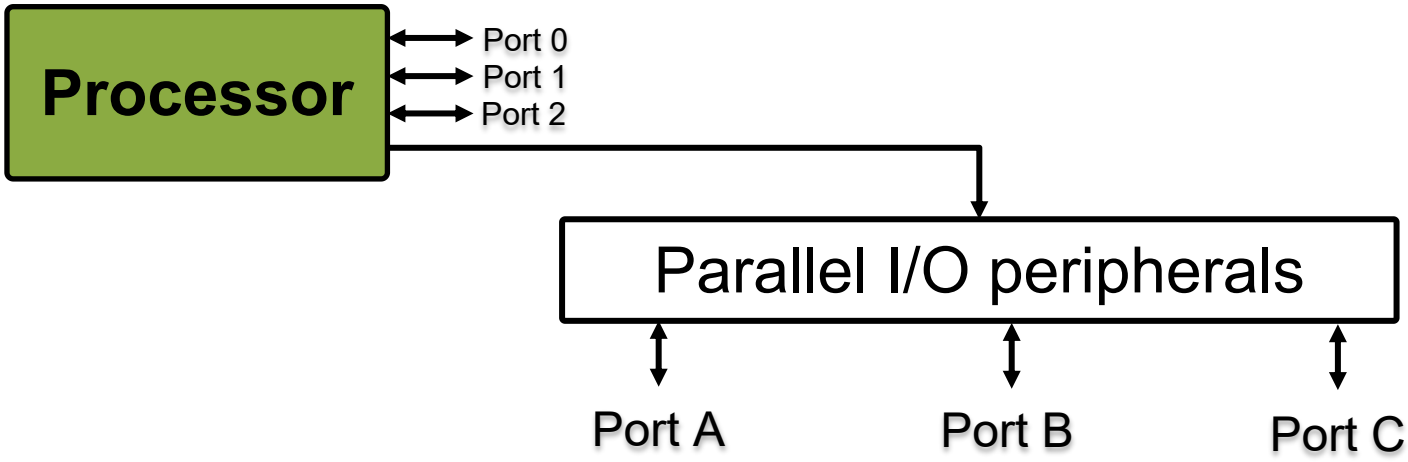


- Parallel I/O for Bus Based System

Compromises/ Extensions

Extended Parallel I/O:

- Processor supports port based I/O but additional ports are required.
- One or multiple processor ports are connected to parallel I/O peripheral to extend the number of available ports.

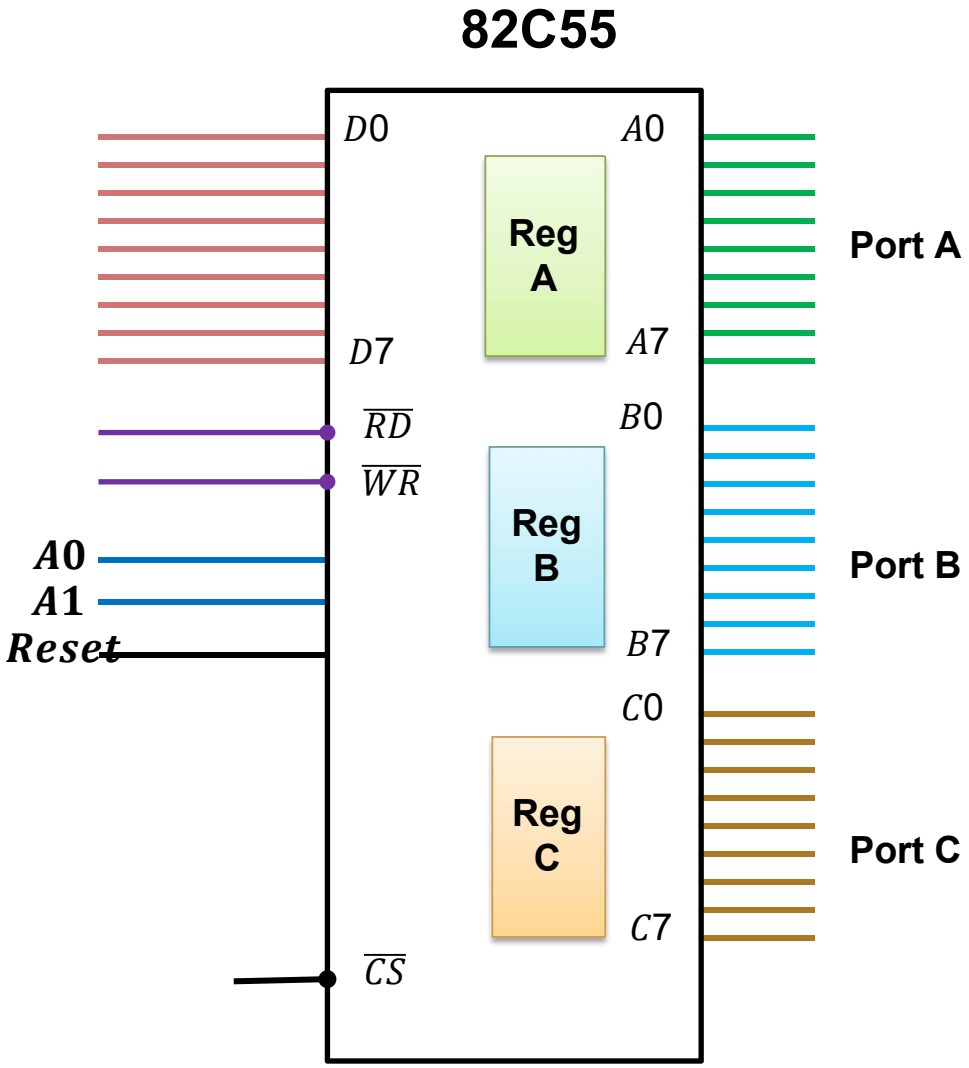


- 4 ports are extended to 6 ports.

Peripheral I/O 82C55

- For every port in 82C55 there is an internal register that can be read or written by processor

A_1	A_0	Function
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Command register



Peripheral I/O 82C55

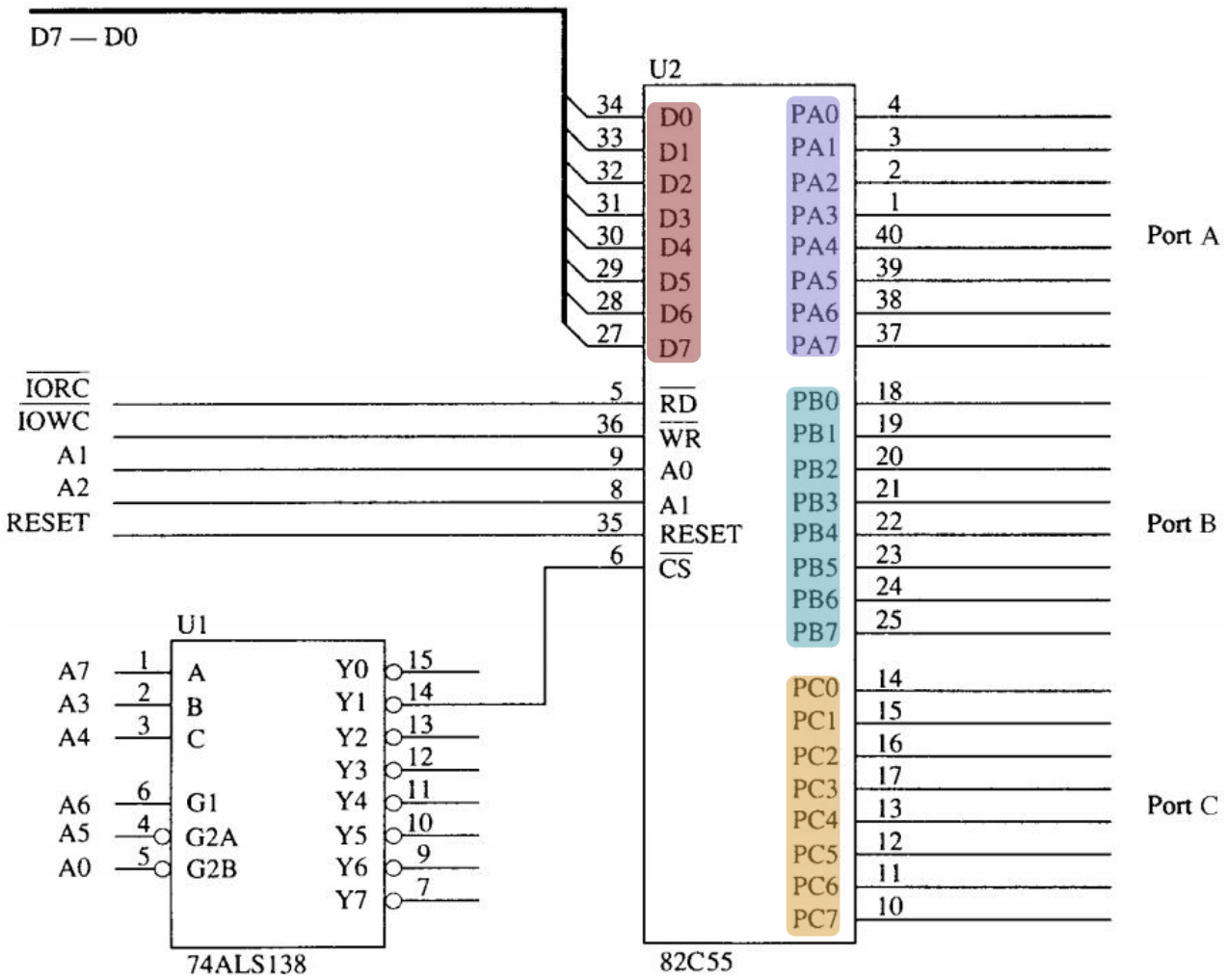


FIGURE 11–19 The 82C55 interfaced to the low bank of the 80386SX microprocessor.

Peripheral I/O 82C55

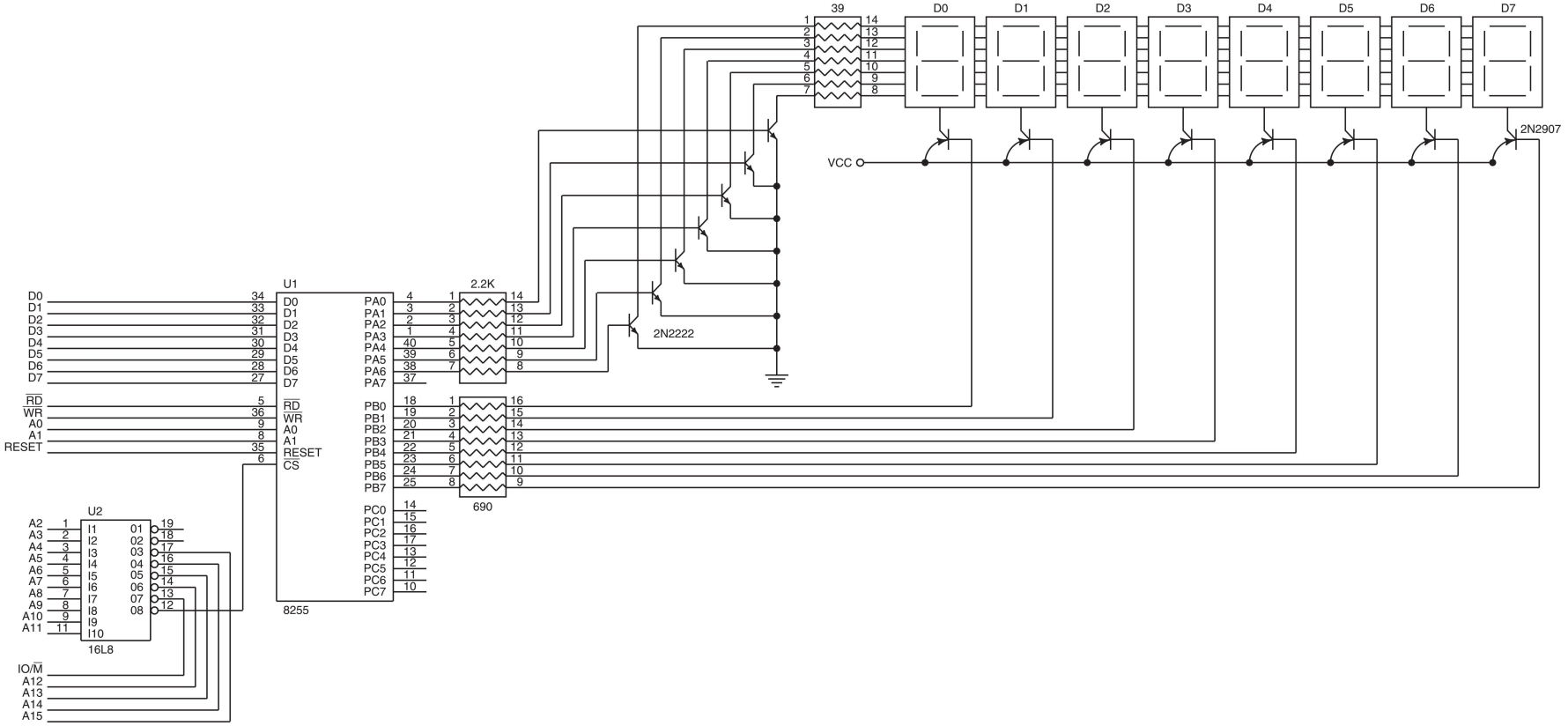
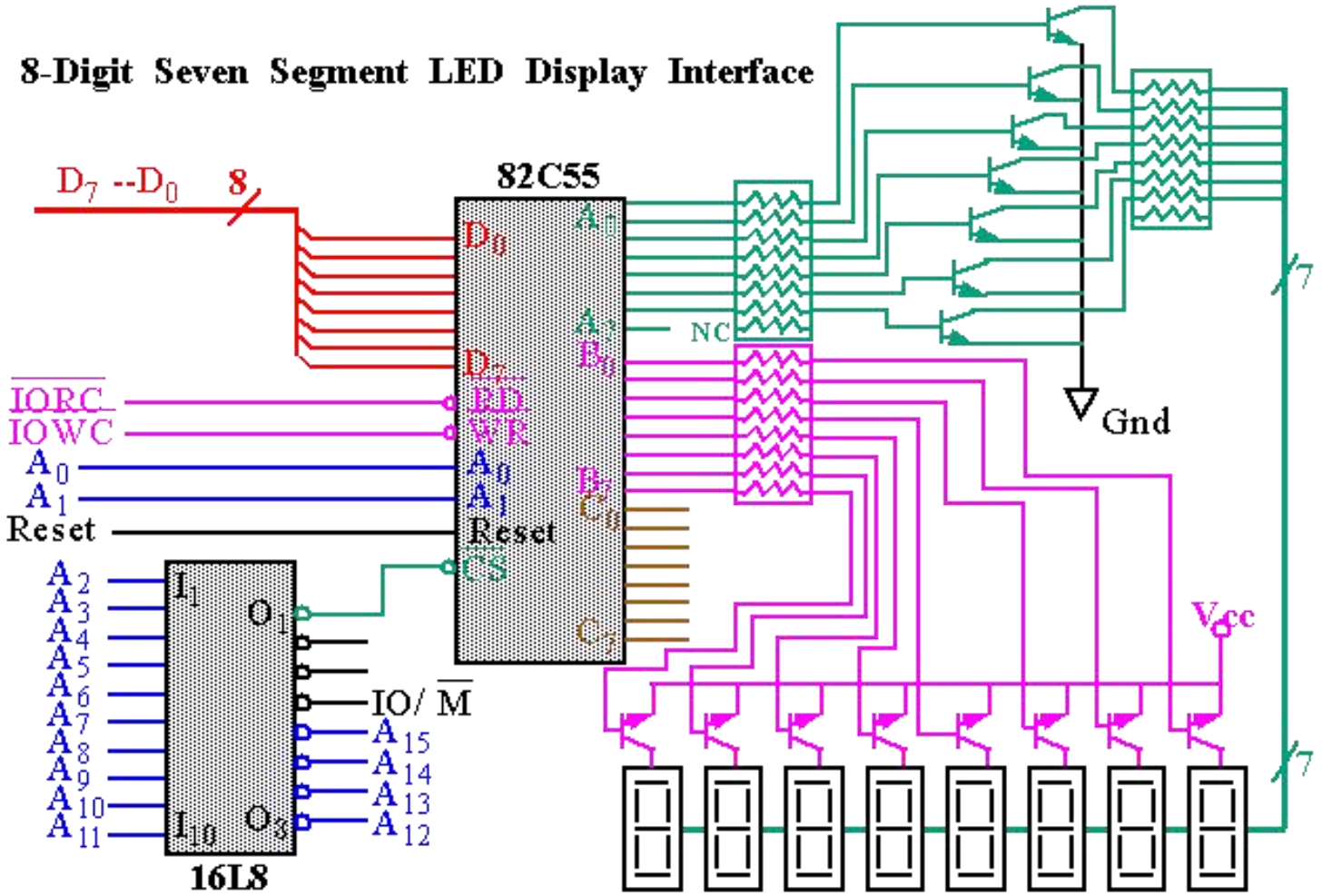


FIGURE 11–21 An 8-digit LED display interfaced to the 8088 microprocessor through an 82C55 PIA.

Peripheral I/O 82C55



Questions?

Thank You!