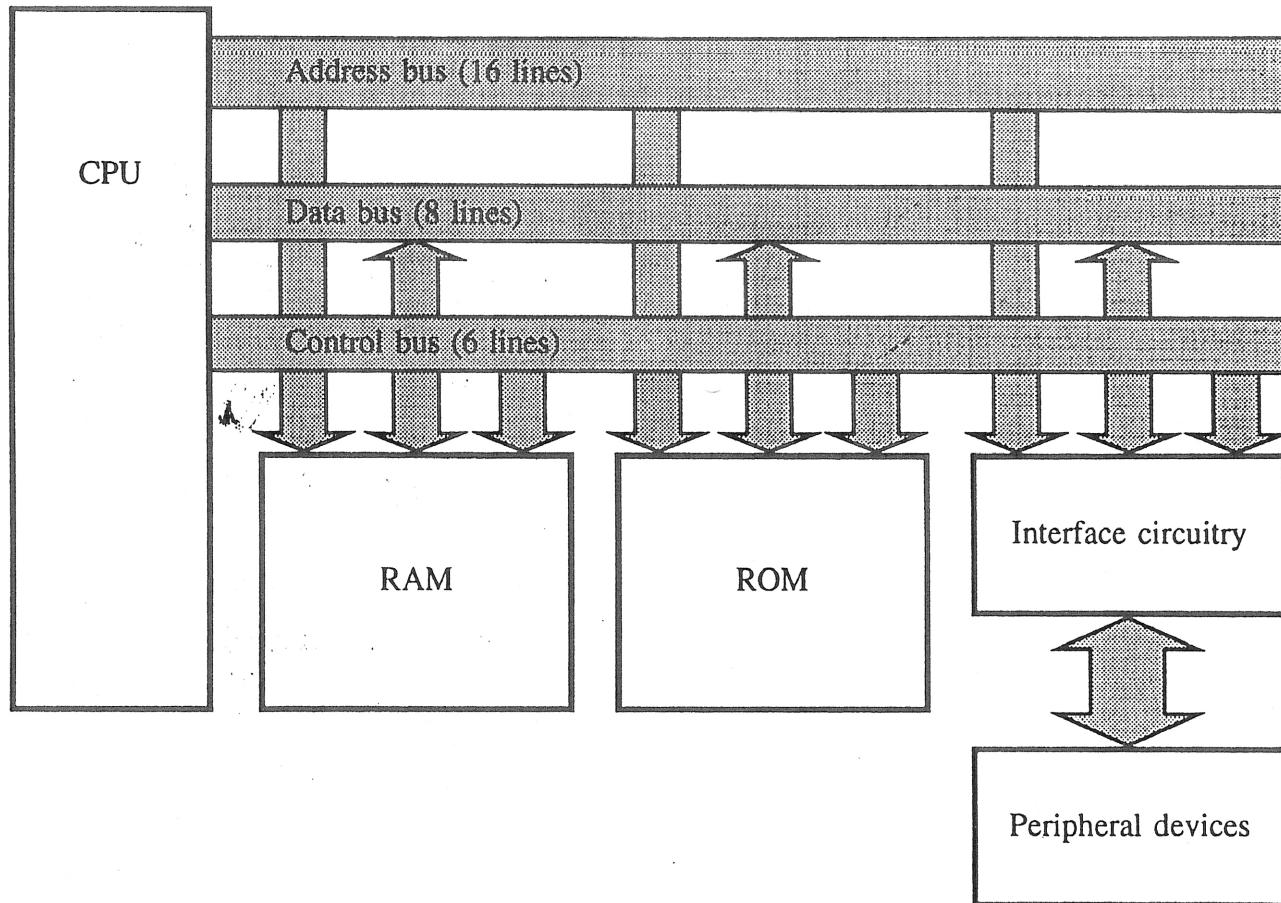


# Memory Chips, Circuits, and Decoders

*Computer Science and Engineering Department*  
**Computer Engineering**

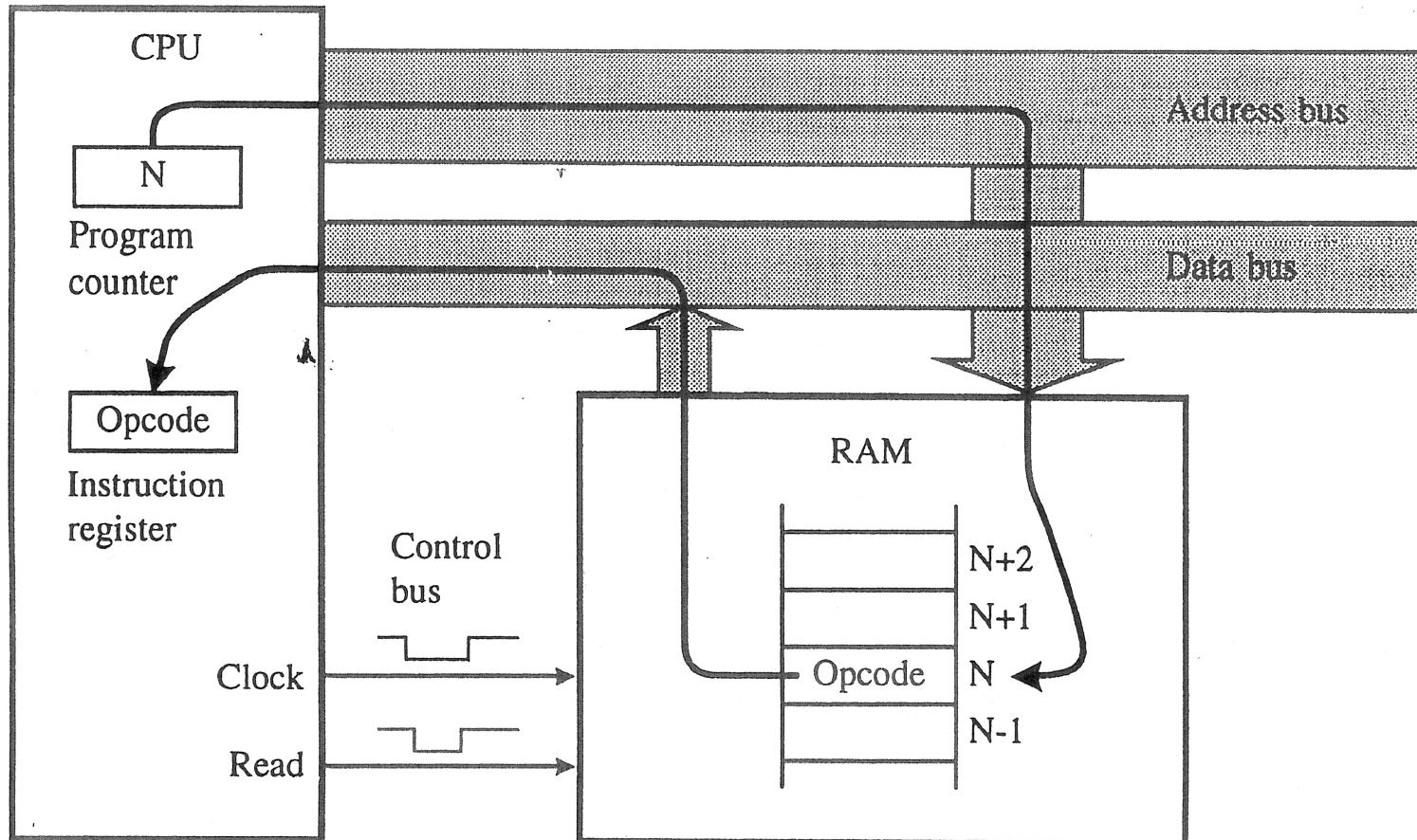
UNIVERSITY  
OF NEVADA  
RENO



**FIGURE 1–2**  
Block diagram of a microcomputer system

*Computer Science and Engineering Department*  
**Computer Engineering**

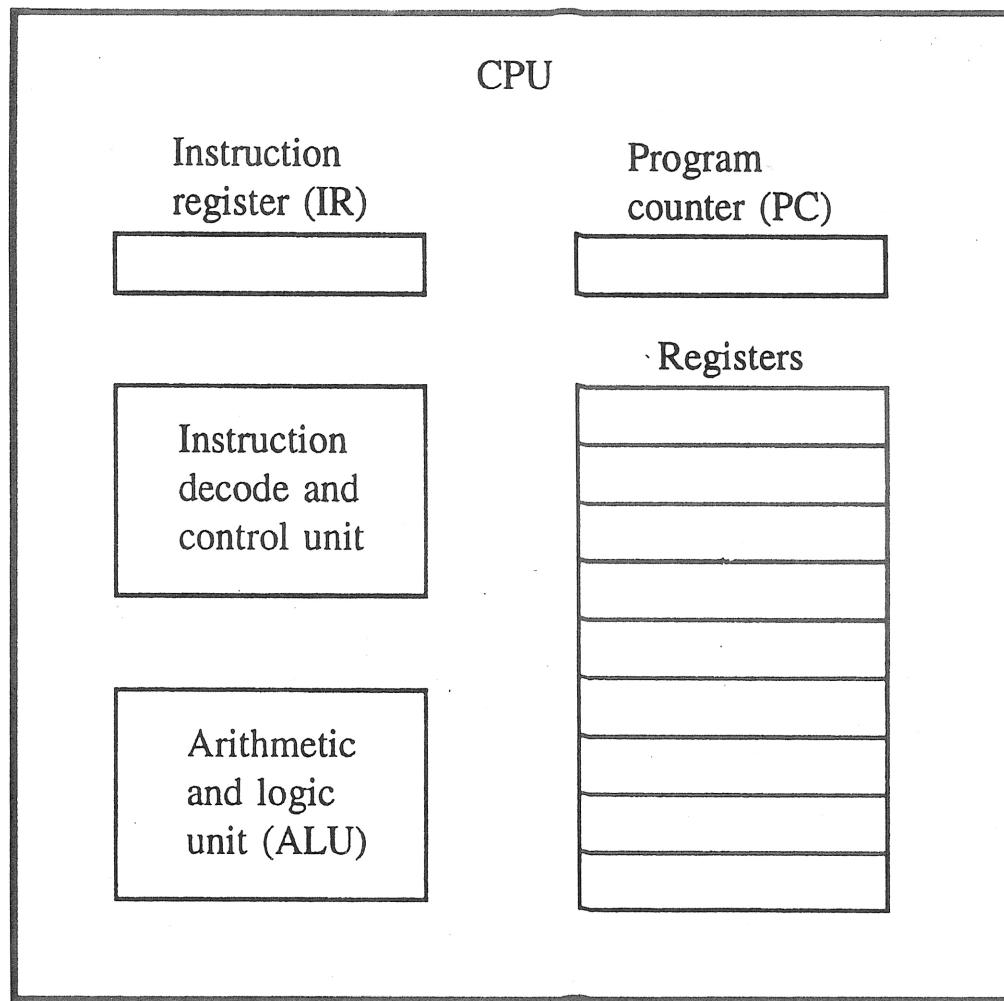
UNIVERSITY  
OF NEVADA  
RENO



**FIGURE 1–4**  
Bus activity for an opcode fetch cycle

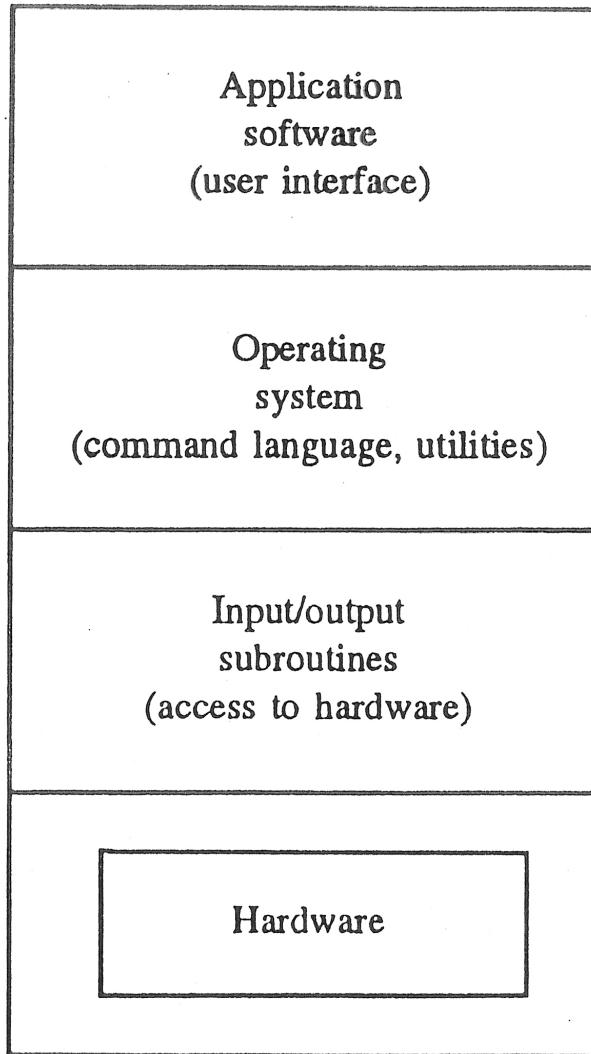
*Computer Science and Engineering Department*  
**Computer Engineering**

**FIGURE 1–3**  
The central processing  
unit (CPU)



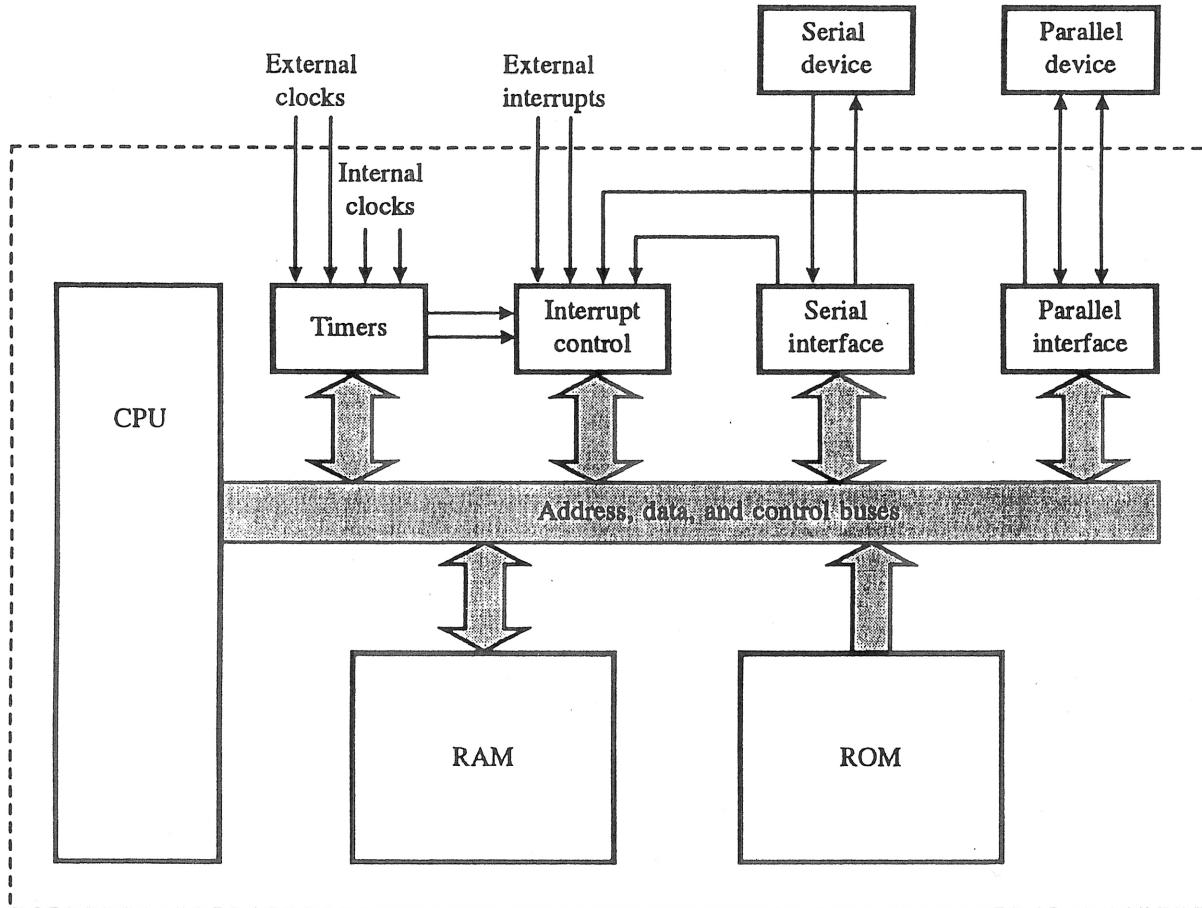
*Computer Science and Engineering Department*  
**Computer Engineering**

**FIGURE 1–5**  
Levels of software



*Computer Science and Engineering Department*  
**Computer Engineering**

UNIVERSITY  
OF NEVADA  
RENO



**FIGURE 1–6**  
Detailed block diagram of a microcomputer system

# Computer Science and Engineering Department

## Computer Engineering

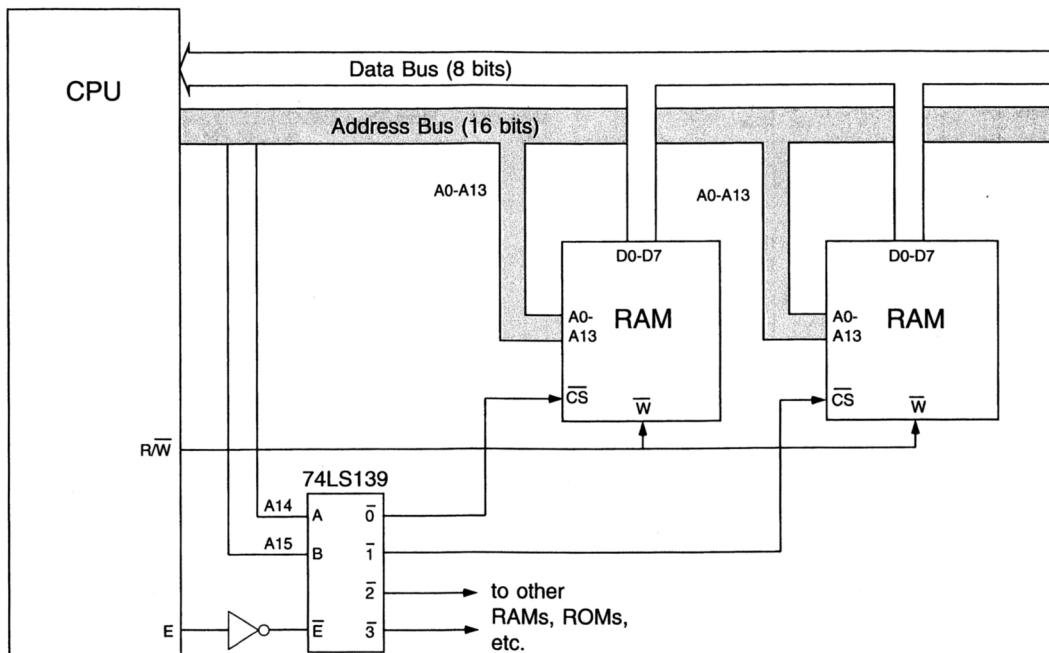


Figure 0-35. Interface between RAMs and a CPU.

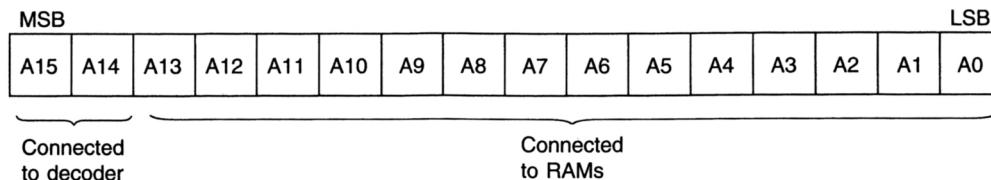
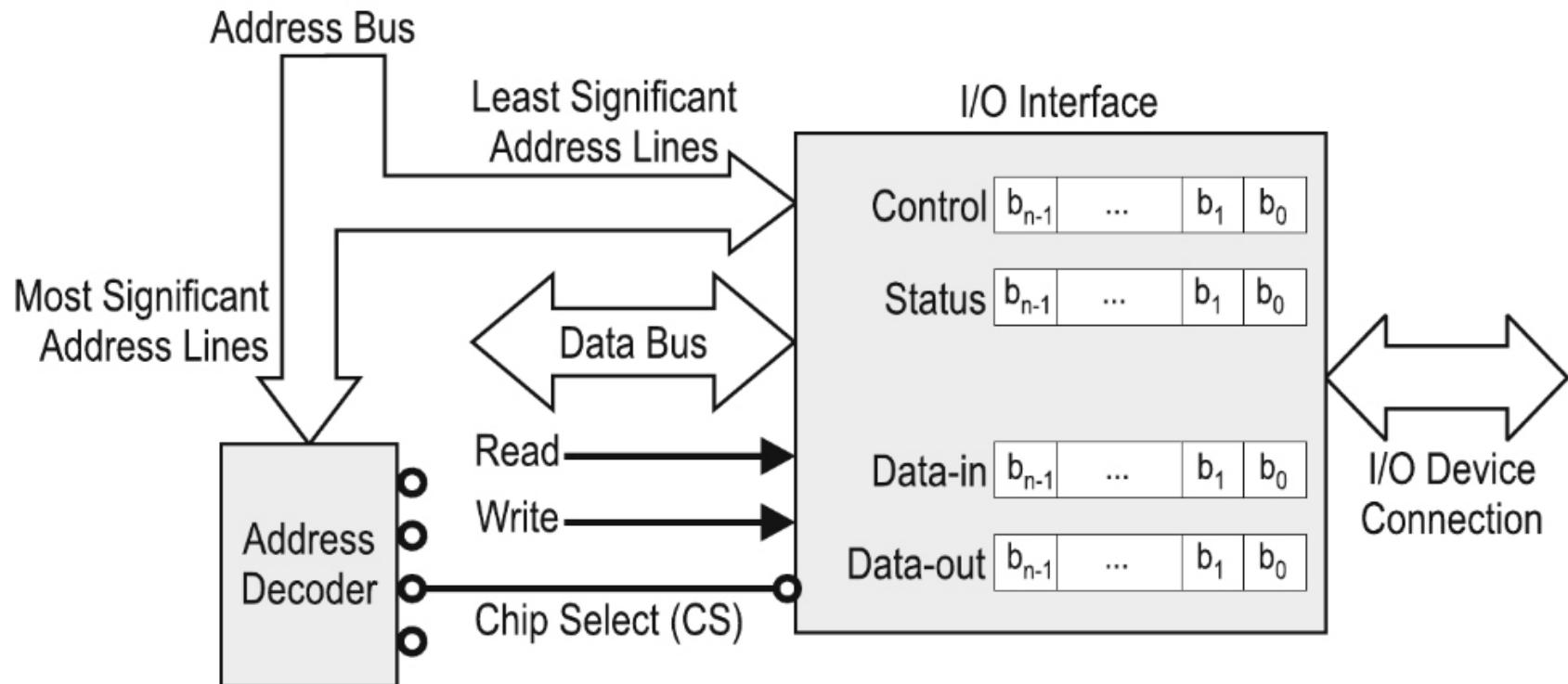


Figure 0-36. Address decomposition for decoder example.

From: "Introduction to Embedded Systems: Using Microcontrollers and the MSP430", by Manuel Jiménez, Rogelio Palomera, Isidoro Couvertier

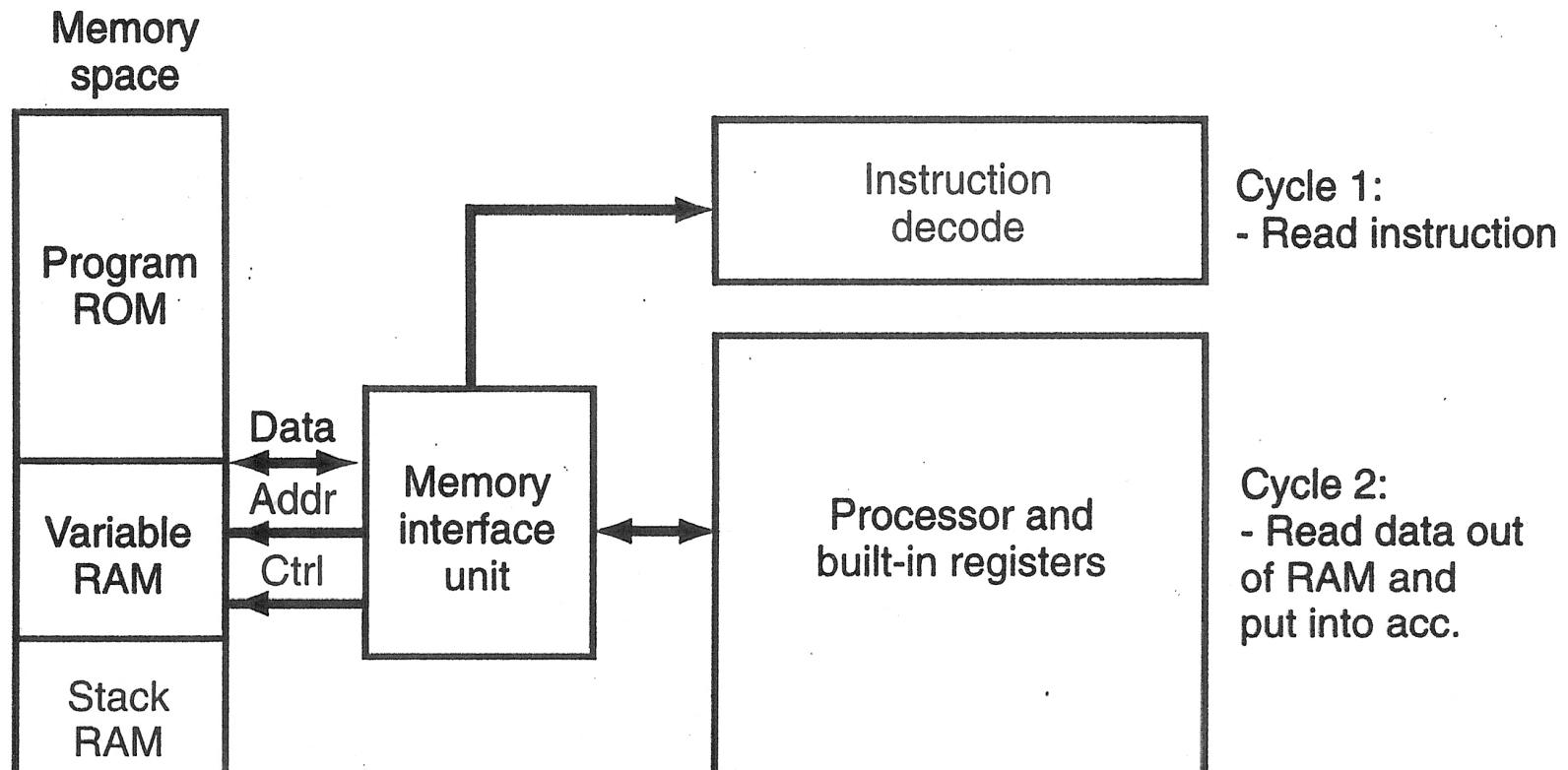
112

### 3 Microcomputer Organization



**Fig. 3.16** Anatomy of an input/output (I/O) interface

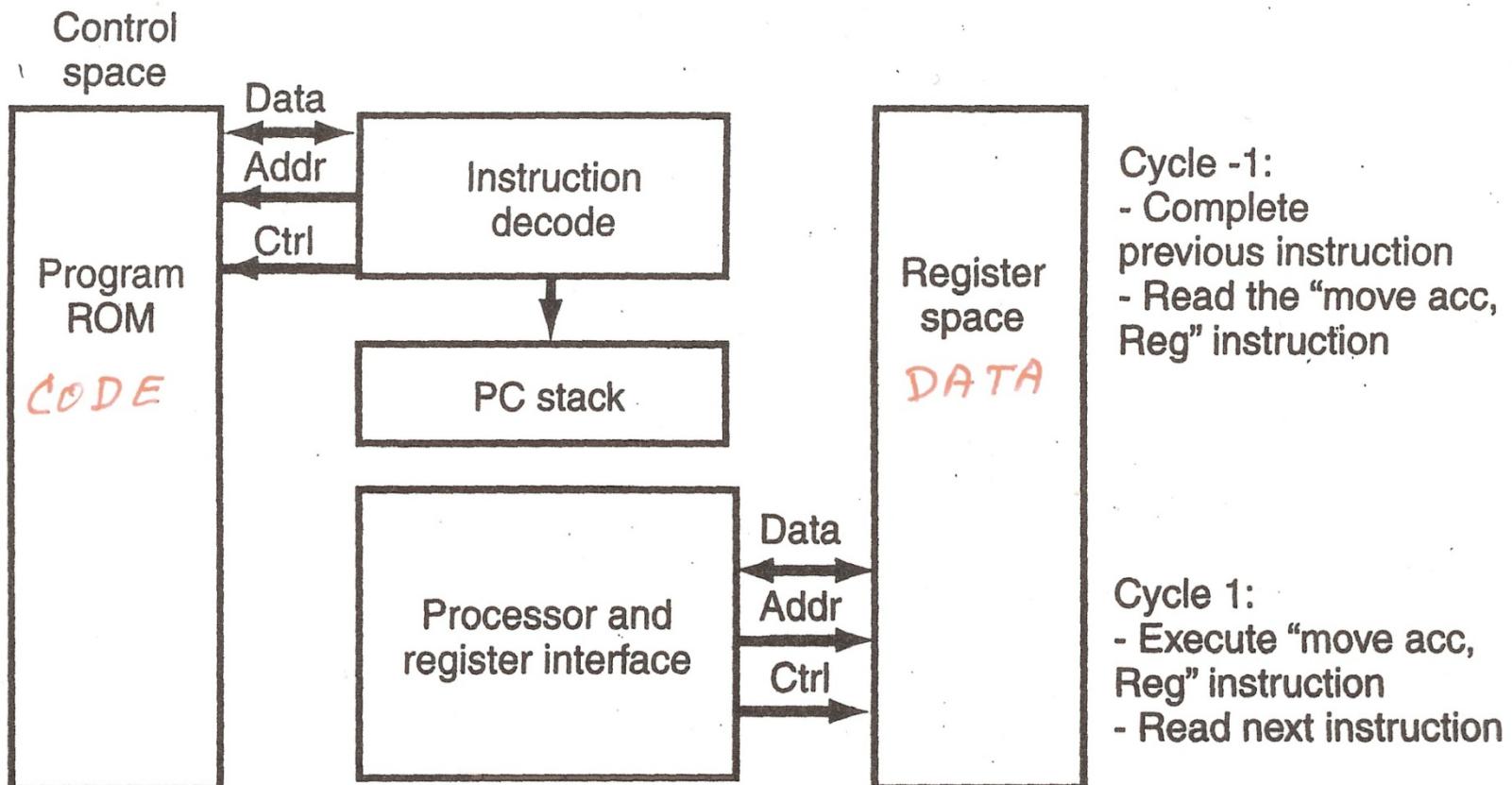
*Computer Science and Engineering Department*  
**Computer Engineering**



**FIGURE 1-6** mov Acc., Reg in the Princeton architecture.

*Computer Science and Engineering Department*  
**Computer Engineering**

UNIVERSITY  
OF NEVADA  
RENO



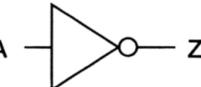
**FIGURE 1-7** mov Acc, Reg in the Harvard architecture.

## **Design of Memory Chips**

**All things are a) AND, b) OR, or c) NOT**

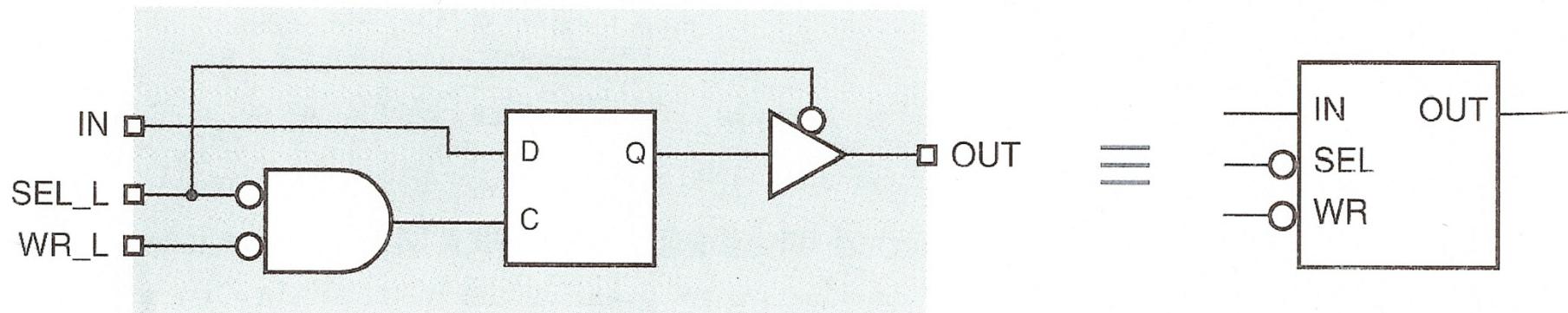
*Computer Science and Engineering Department*  
**Computer Engineering**

**Table 0-11. Basic Logic Gates**

Operation	Symbol	TTL Part #	Equation	Truth Table															
INVERTER		74LS04	$Z = \overline{A}$	<table border="1"><tr><td>A</td><td>Z</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	Z	0	1	1	0									
A	Z																		
0	1																		
1	0																		
OR		74LS32	$Z = A + B$ $Z = A \vee B$	<table border="1"><tr><td>A</td><td>B</td><td>Z</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Z	0	0	0	0	1	1	1	0	1	1	1	1
A	B	Z																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
AND		74LS08	$Z = A \bullet B$ $Z = A \wedge B$	<table border="1"><tr><td>A</td><td>B</td><td>Z</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Z	0	0	0	0	1	0	1	0	0	1	1	1
A	B	Z																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	

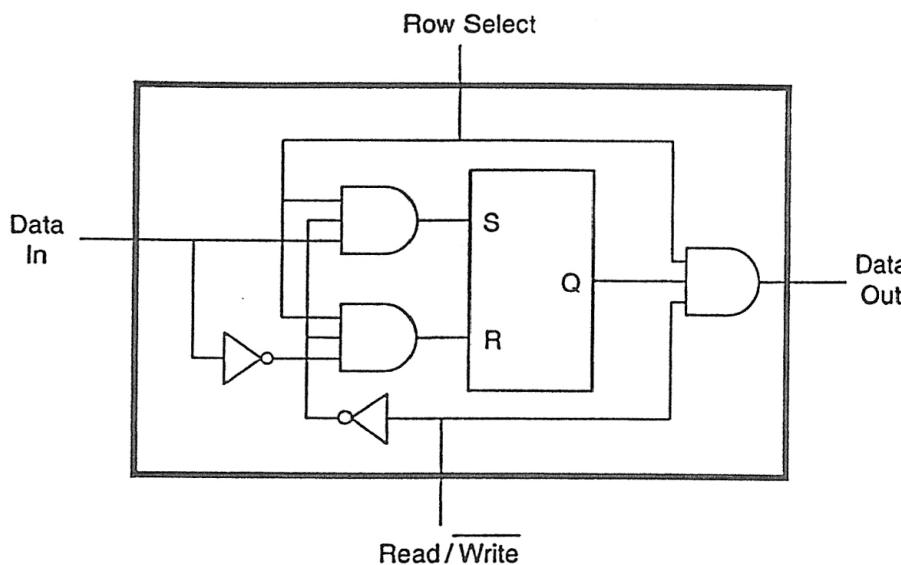
*Computer Science and Engineering Department*  
**Computer Engineering**

Figure 10-20 Functional behavior of a static-RAM cell.



*Computer Science and Engineering Department*  
**Computer Engineering**

UNIVERSITY  
OF NEVADA  
RENO



(a)

Row Select	<u>Read/ Write</u>	Data In	Q	Data Out	Function
0	x	x	x	0	deselect
1	0	1/0	1/0	0	write
1	1	x	1/0	1/0	read

(b)

Figure 0-41. Memory cell made from an SR latch. (a) schematic, (b) function table.

# Computer Science and Engineering Department

# Computer Engineering

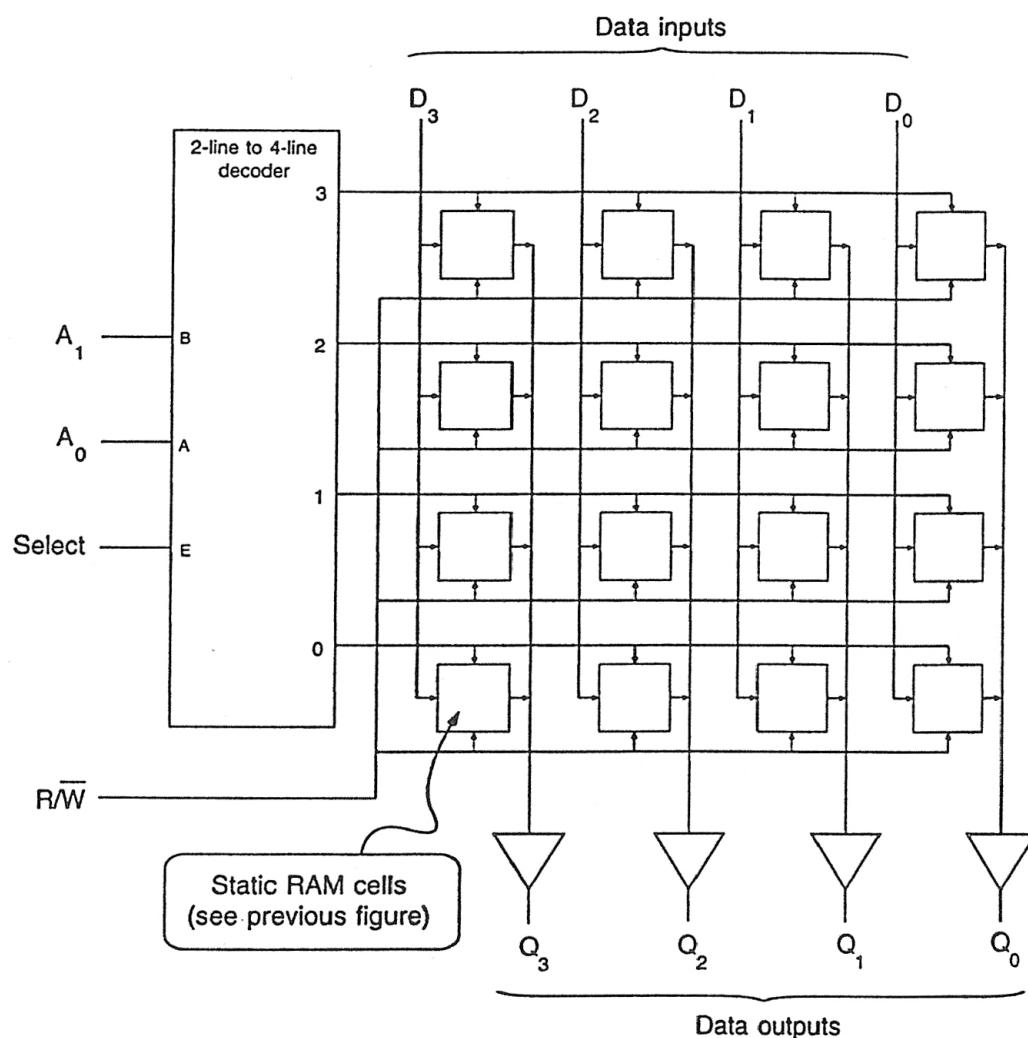
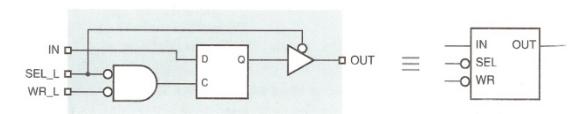


Figure 0-42. Static RAM IC.

Figure 10-20 Functional behavior of a static-RAM cell.



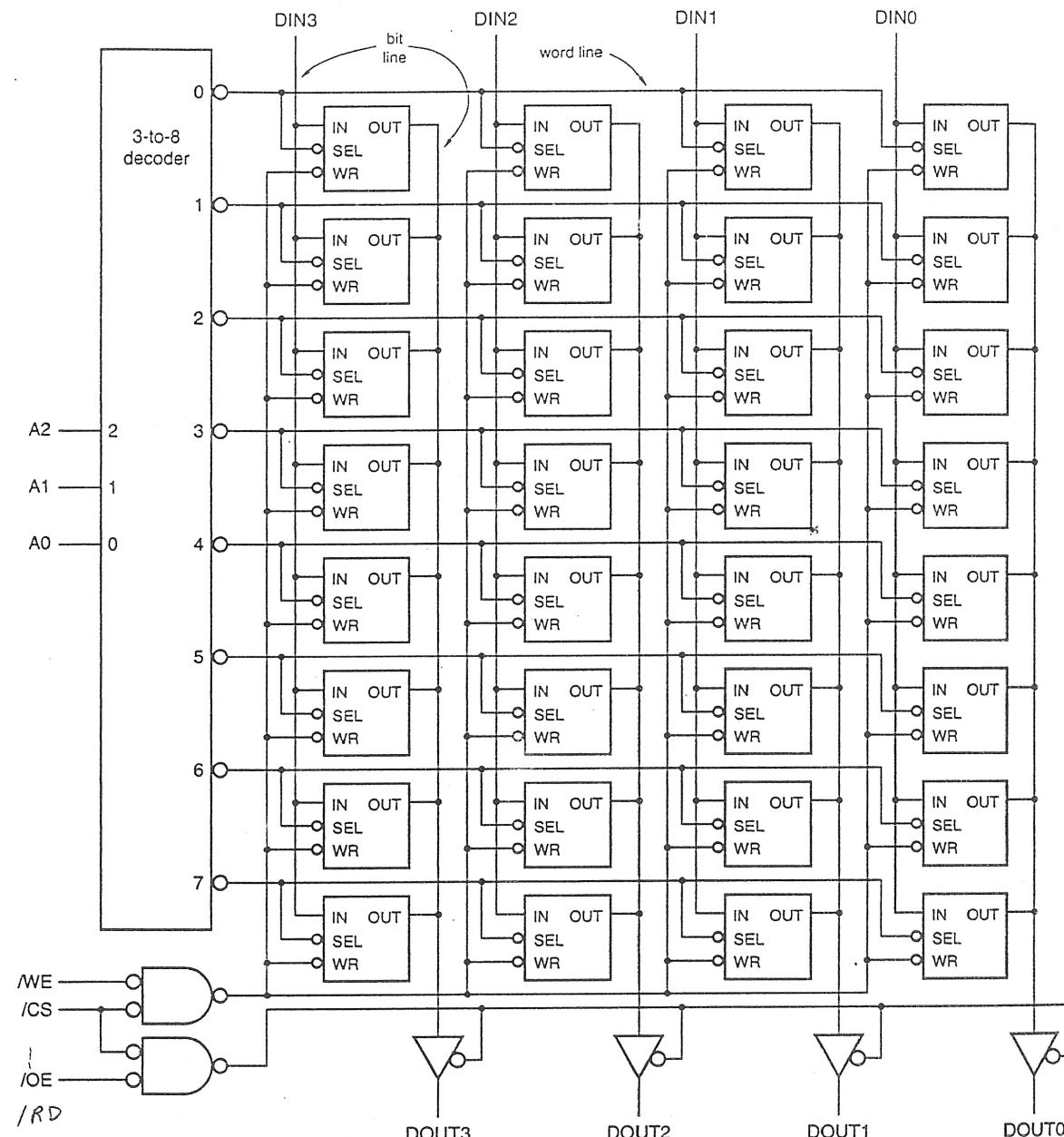


Figure 11-21 Internal structure of an 8x4 static RAM.

*Computer Science and Engineering Department*  
**Computer Engineering**

UNIVERSITY  
OF NEVADA  
RENO

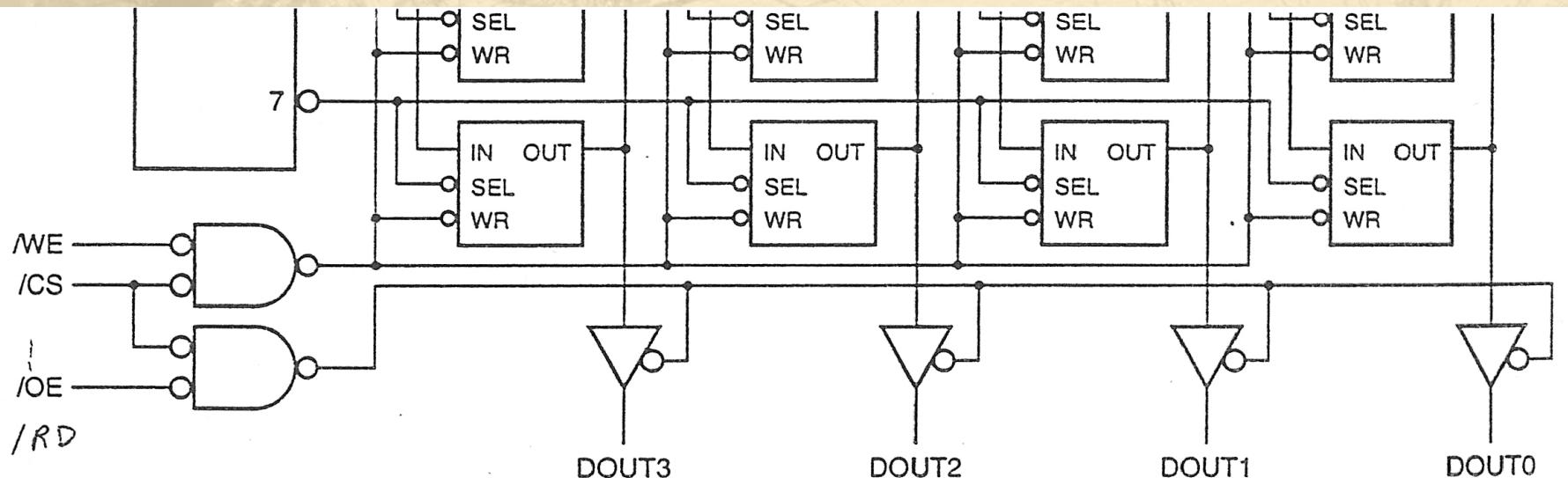


Figure 11-21 Internal structure of an 8x4 static RAM.

*Computer Science and Engineering Department*  
**Computer Engineering**

UNIVERSITY  
OF NEVADA  
RENO

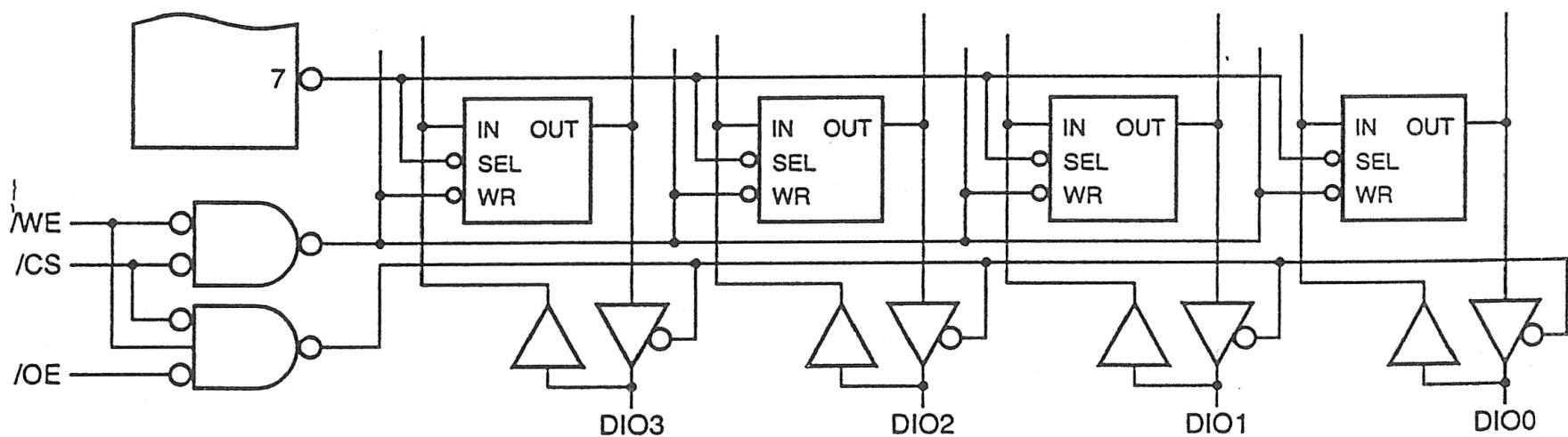


Figure 11–25 Output-buffer control in an SRAM with a bidirectional data bus.

*Computer Science and Engineering Department*  
**Computer Engineering**

UNIVERSITY  
OF NEVADA  
RENO

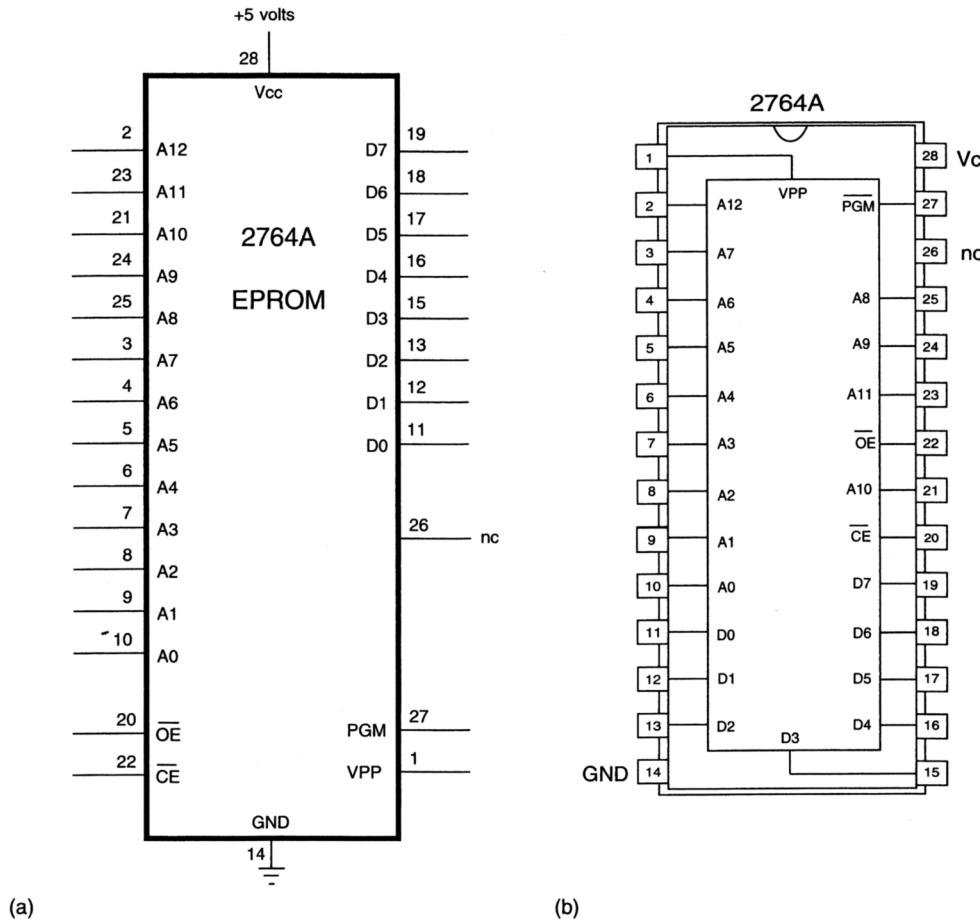


Figure 0-44. 2764A 8K byte EPROM. (a) functional symbol, (b) pinouts.

*Computer Science and Engineering Department*  
**Computer Engineering**

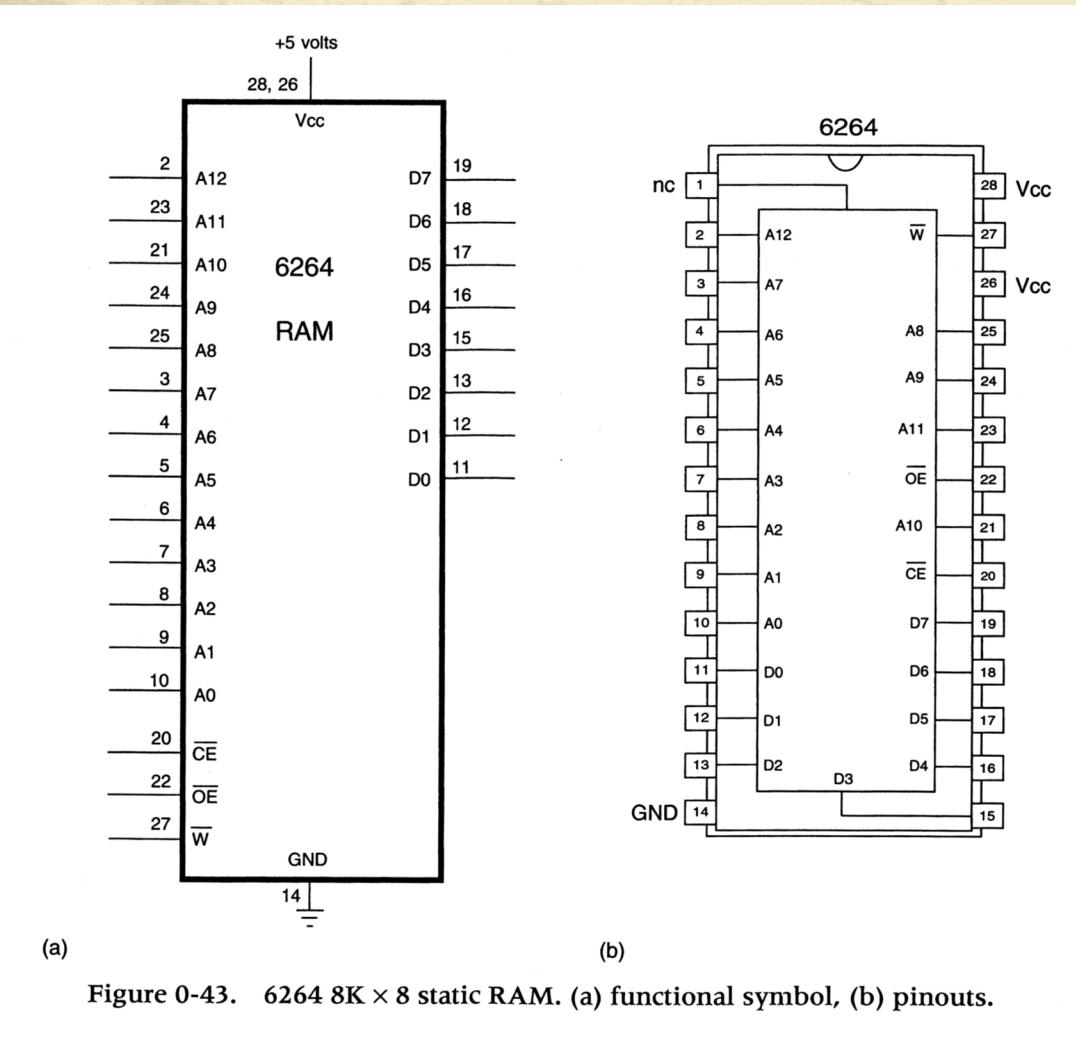


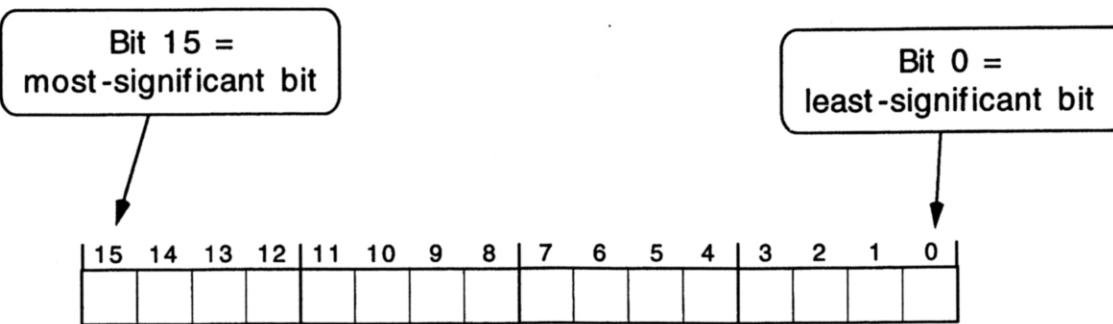
Figure 0-43. 6264 8K × 8 static RAM. (a) functional symbol, (b) pinouts.

**Memory Addresses**

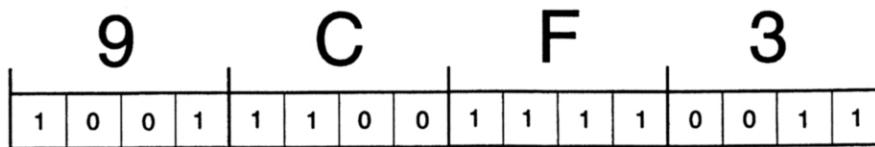
**Memory Map**

**Memory Address Table**

*Computer Science and Engineering Department*  
**Computer Engineering**



(a)



(b)

Figure 1-8. 16-bit addresses. (a) bit numbering, (b) address example.

*Computer Science and Engineering Department*  
**Computer Engineering**

UNIVERSITY  
OF NEVADA  
RENO

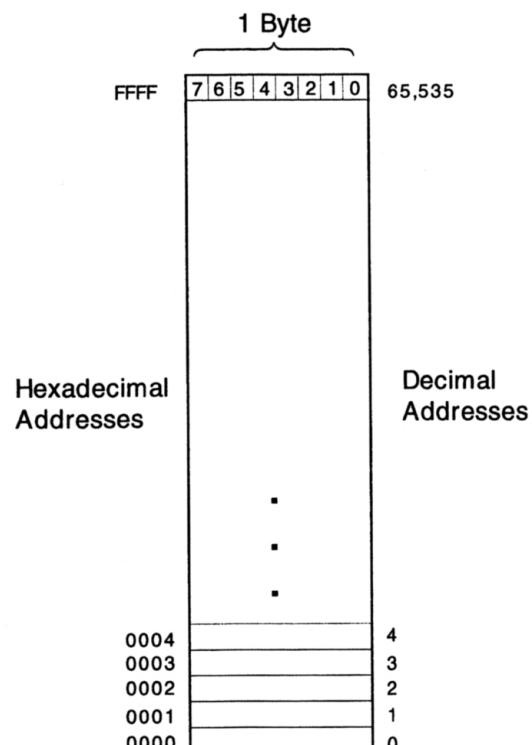


Figure 1-6. Memory map.

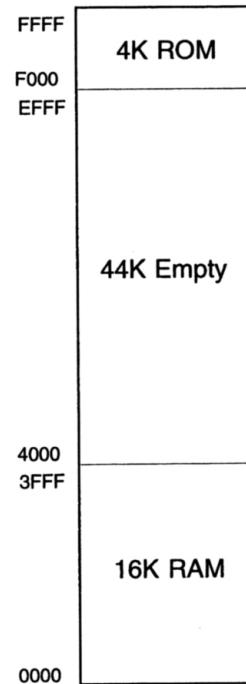


Figure 1-7. A sample memory map.

# Computer Science and Engineering Department

# Computer Engineering

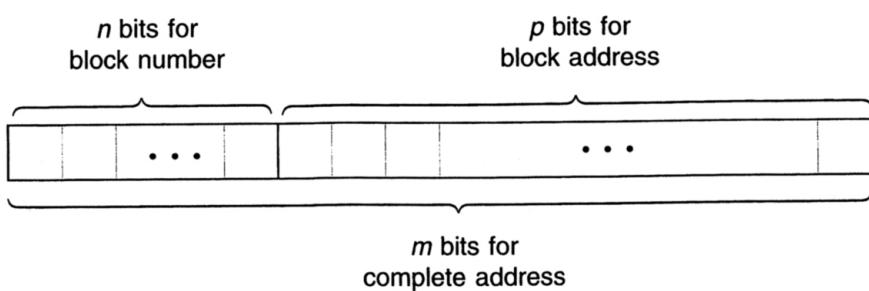


Figure 1-9. Dividing a memory space into blocks.

Table 1-2. Dividing a 64K Memory Space

$n$	$p$	Number of Blocks ( $2^n$ )	Block Size ( $2^p$ )
0	16	1	65536
1	15	2	32768
2	14	4	16384
3	13	8	8192
4	12	16	4096
5	11	32	2048
6	10	64	1024
7	9	128	512
8	8	256	256
9	7	512	128
10	6	1024	64
11	5	2048	32
12	4	4096	16
13	3	8192	8
14	2	16384	4
15	1	32768	2
16	0	65536	1

# Computer Science and Engineering Department

## Computer Engineering

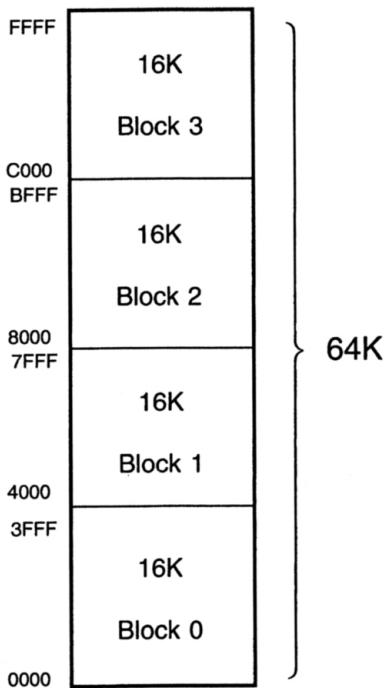


Figure 1-10. Memory map showing four 16K blocks.

Block Number	***** Binary Addresses *****		*** Hex ***	
	From	To	From	To
0	00 00000000000000	00 11111111111111	0000	3FFF
1	01 00000000000000	01 11111111111111	4000	7FFF
2	10 00000000000000	10 11111111111111	8000	BFFF
3	11 00000000000000	11 11111111111111	C000	FFFF

## Memory Address Table

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Chip1-Start

Chip1-End

Chip2-Start

Chip2-End

Chip3-Start

Chip3-End

Chip4-Start

Chip4-End

## Sample Memory Mapping Question

1. Design an ATmega2560 microcontroller (Harvard architecture) system memory address decoding circuit which has a single 4Kx8 memory chip starting at address 0x4000 and an ADC0820 Analog to Digital Converter (data sheet attached) starting at 0x6000 so that they are both mapped as **external data memory**.

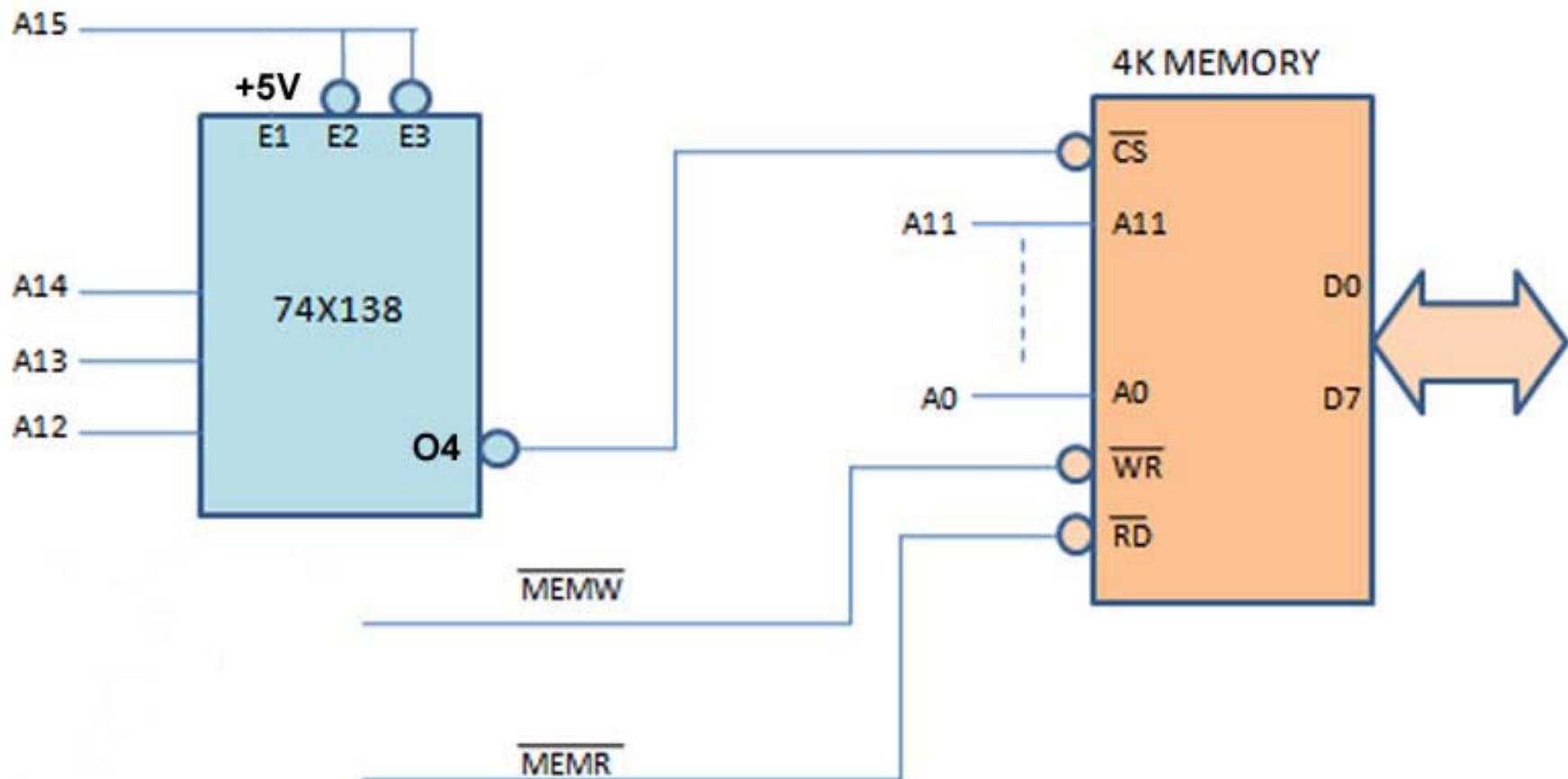
Use ABSOLUTE decoding with no fold-back for the external DATA memory.

Show your work and include the following items:

- a. MEMORY MAP
- b. MEMORY ADDRESS TABLE
- c. BOOLEAN EQUATIONS for each chip select signal
- d. Full schematic logic diagram of your DECODING CIRCUITS

Computer Science and Engineering Department  
**Computer Engineering**

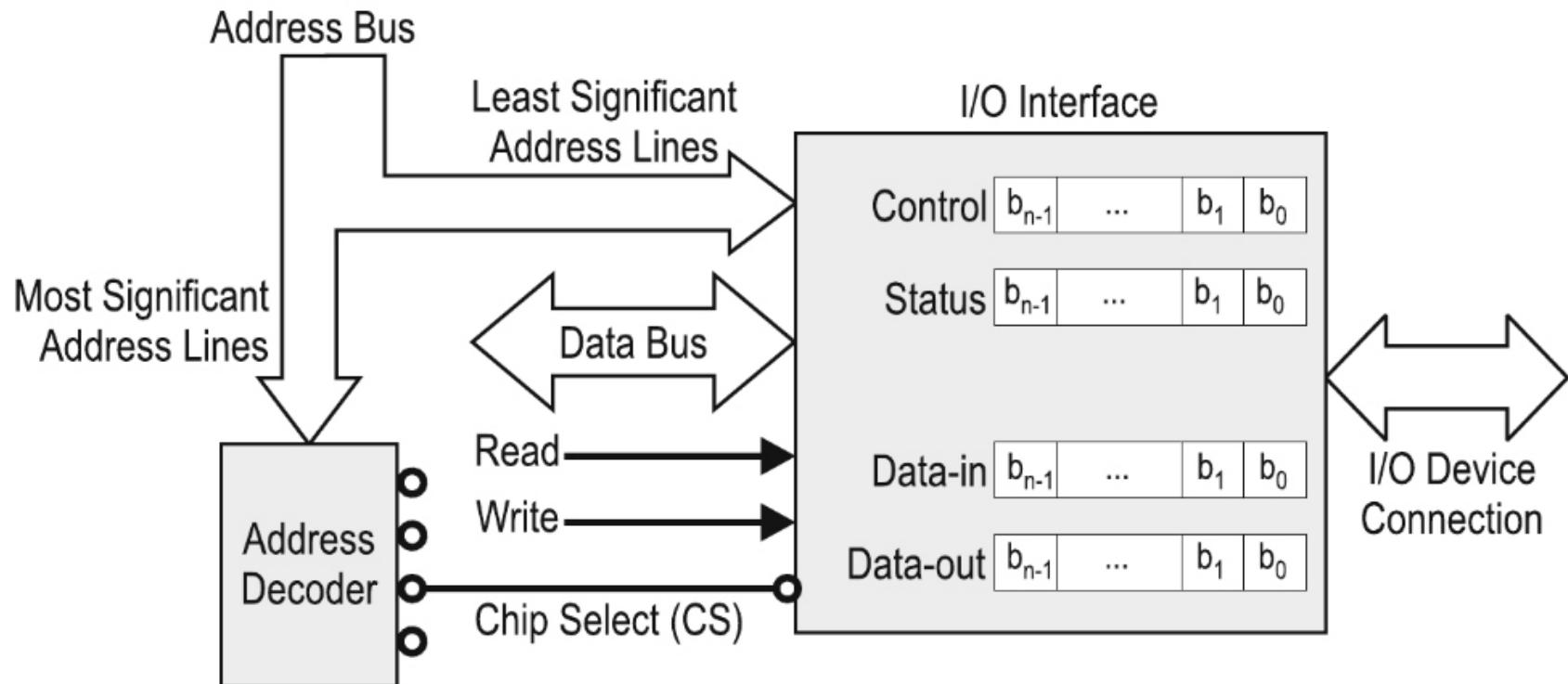
4Kx8 RAM Chip alone mapped to start at 0x4000



From: "Introduction to Embedded Systems: Using Microcontrollers and the MSP430", by Manuel Jiménez, Rogelio Palomera, Isidoro Couvertier

112

### 3 Microcomputer Organization



**Fig. 3.16** Anatomy of an input/output (I/O) interface

# Computer Science and Engineering Department

# Computer Engineering

## 8-Bit Resolution Configuration

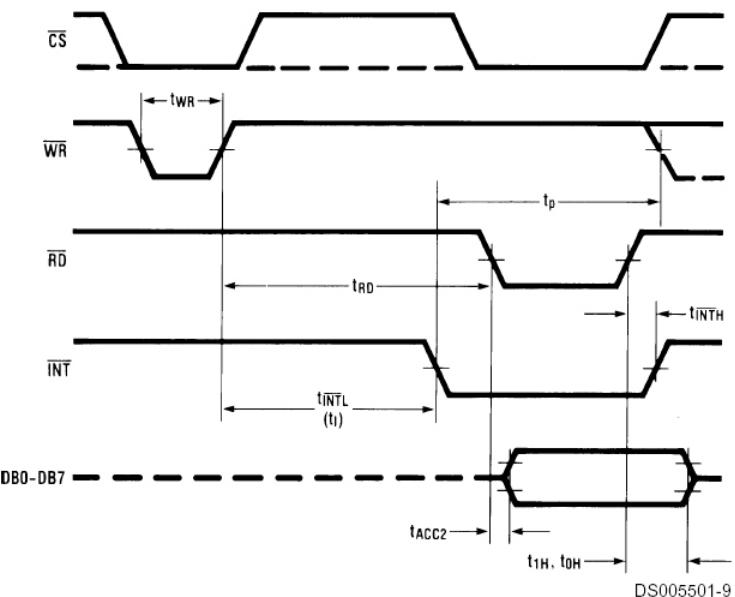
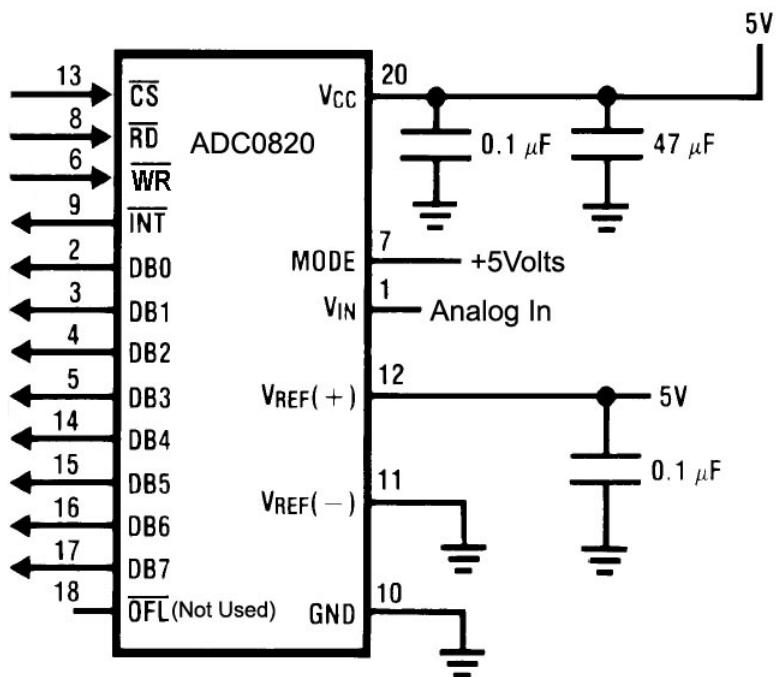
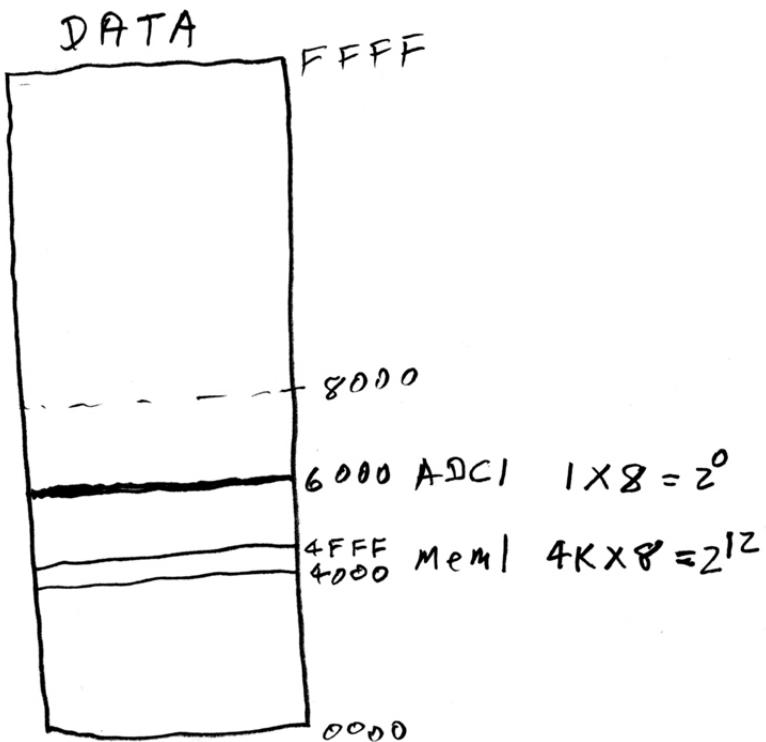


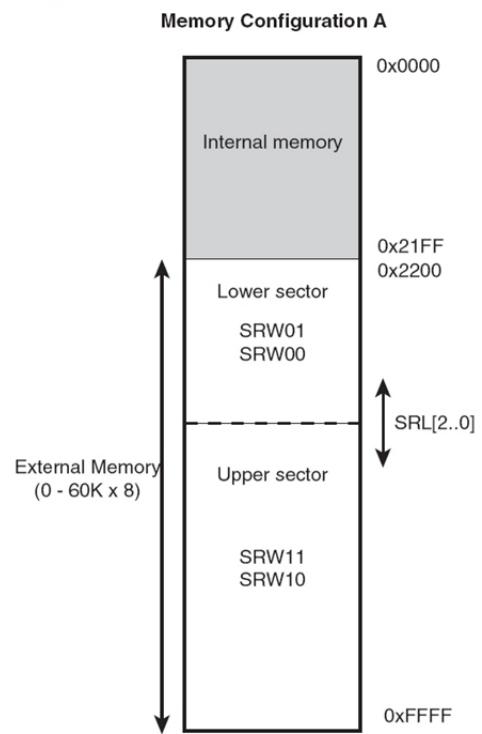
FIGURE 4. WR-RD Mode (Pin 7 is High and  $t_{RD} > t_I$ )

*Computer Science and Engineering Department*  
**Computer Engineering**

(a)



**Figure 9-1.** External Memory with Sector Select



*Computer Science and Engineering Department*  
**Computer Engineering**

(b)

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Mem1 - S  
 Mem1 - E

ADC1 - S  
 ADC1 - E

(c)

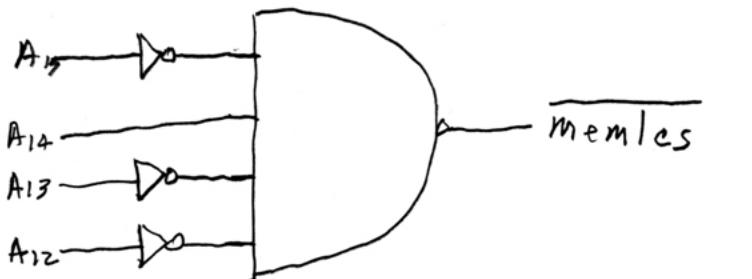
$$\text{Mem1cs} = A15^* \cdot A41 \cdot A13^* \cdot A12^*$$

$$\text{ADC1cs} = A15^* \cdot A41 \cdot A13 \cdot A12^* \cdot A11^* \cdot A10^* \cdot A9^* \cdot A8^* \cdot A7^* \cdot A6^* \cdot A5^* \cdot A4^* \cdot A3^* \cdot A2^* \cdot A1^* \cdot A0^*$$

$$(\text{ADC1cs}^* = A15 + A41^* + A13^* + A12 + A11 + A10 + A9 + A8 + A7 + A6 + A5 + A4 + A3 + A2 + A1 + A0)$$

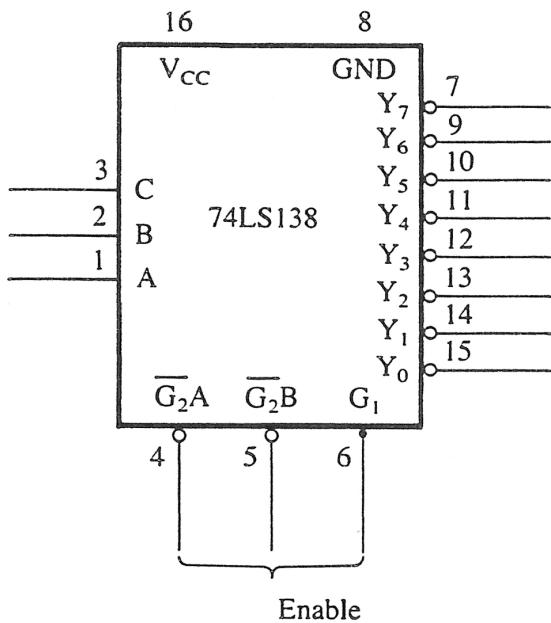
Computer Science and Engineering Department  
**Computer Engineering**

(d)



# Computer Science and Engineering Department

## Computer Engineering



LS138, S138  
Function Table

Inputs		Outputs							
Enable	Select	Y <sub>0</sub>	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>	Y <sub>4</sub>	Y <sub>5</sub>	Y <sub>6</sub>	Y <sub>7</sub>
G <sub>1</sub> G <sub>2</sub>	C B A	X X X	H H H	H H H	H H H	H H H	H H H	H H H	H H H
X H	X X X	H H H	H H H	H H H	H H H	H H H	H H H	H H H	H H H
L X	X X X	H H H	H H H	H H H	H H H	H H H	H H H	H H H	H H H
H L	L L L	L H H	H H H	H H H	H H H	H H H	H H H	H H H	H H H
H L	L L H	H L H	H H H	H H H	H H H	H H H	H H H	H H H	H H H
H L	L H L	H H L	H H H	H H H	H H H	H H H	H H H	H H H	H H H
H L	L H H	H H L	H H H	H H H	H H H	H H H	H H H	H H H	H H H
H L	L H H	H H H	H H H	H H H	H H H	H H H	H H H	H H H	H H H
H L	H L L	H H H	H H H	H H H	H H H	H L H	H H H	H H H	H H H
H L	H L H	H H H	H H H	H H H	H H H	H H L	H H H	H H H	H H H
H L	H H L	H H H	H H H	H H H	H H H	H H H	H L H	H H H	H H H
H L	H H H	H H H	H H H	H H H	H H H	H H H	H H H	H L H	H H H

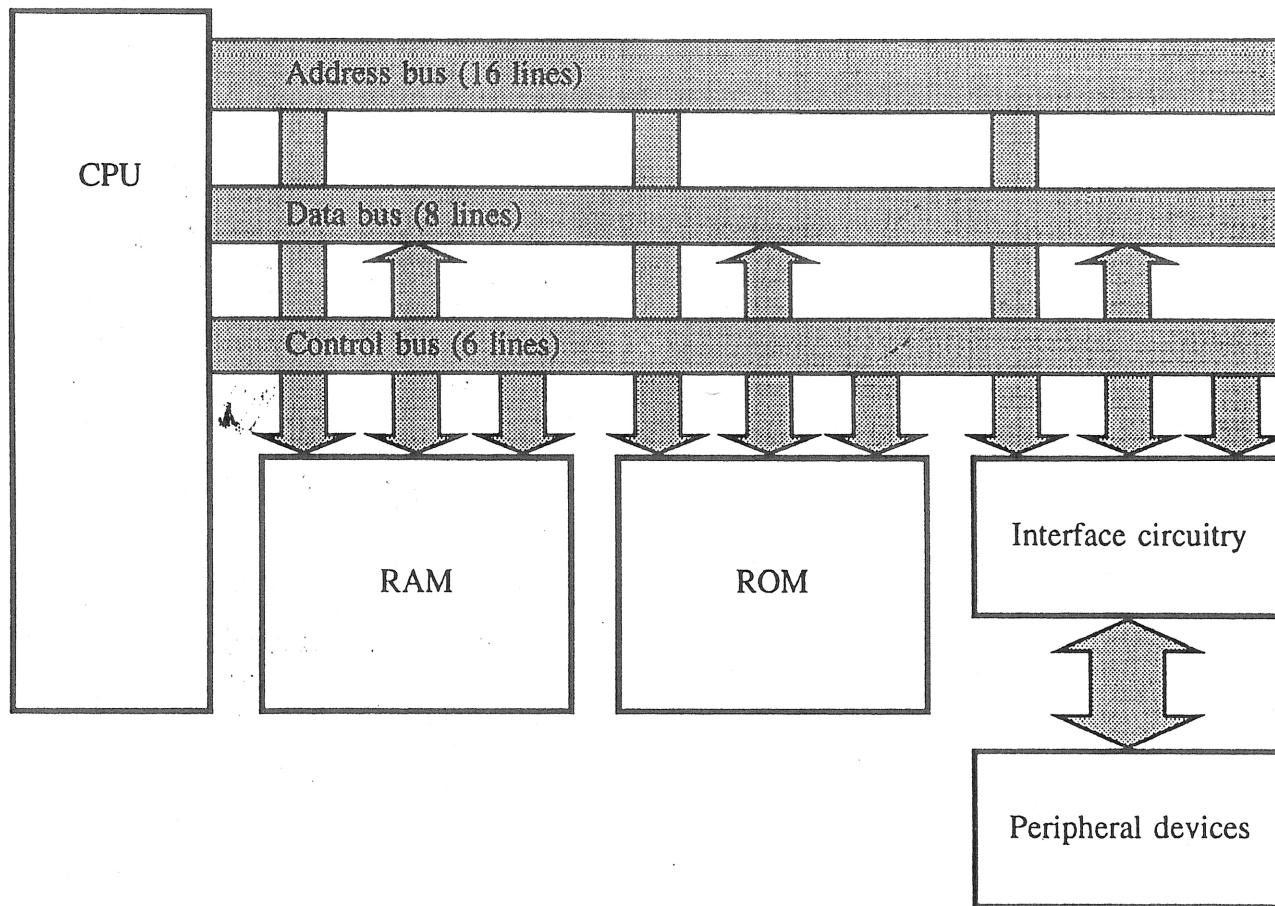
H = high level, L = low level, X = irrelevant

# Memory Access

## Timing Diagrams

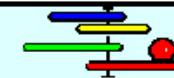
*Computer Science and Engineering Department*  
**Computer Engineering**

UNIVERSITY  
OF NEVADA  
RENO



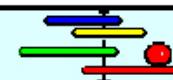
**FIGURE 1–2**  
Block diagram of a microcomputer system

## **Timing Diagrams**

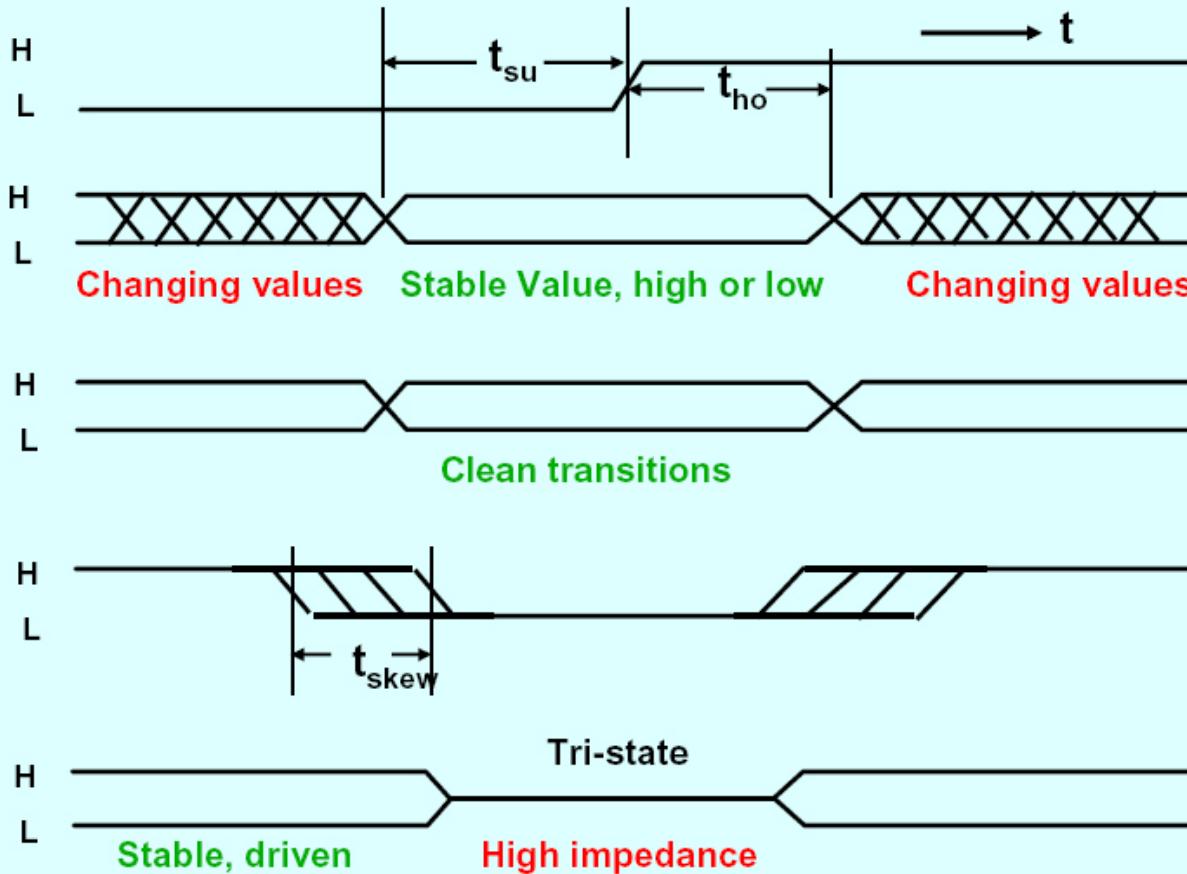


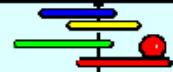
- **Graphical representation of circuit behavior over time**
  - illustrates the logic behavior of signals in a digital circuit as a function of time
- **May be used as a device specification**
  - illustrates device performance
- **May be used as a module or system specification**
  - identifies a requirement for system performance
- **May be used as a tool in system analysis**

2



## Timing Diagram Notation





## Timing Terminology

**Caution:** terminology may vary slightly between vendors.

*Set-up time ,  $t_{su}$ :* the minimum length of time that a signal must be valid at a circuit input before a second triggering signal arrives at a second input.  
Usually a clock

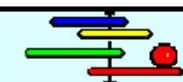
*Delay time ,  $t_{co}$ :* the length of time that a circuit requires for its output(s) to begin to change in response to a triggering signal arriving at an input. (also called **propagation delay**)

*Hold time ,  $t_{ho}$ :* the minimum length of time that a signal must be kept valid at a circuit input after a triggering signal has been received at a second input.

*Timing skew ,  $t_{skew}$ :* the maximum range of times over which a particular signal transition can occur.

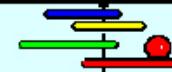
- Due to variations in driver output impedance
- Problems in clock distribution

### *Nomenclature for Timing Diagrams*



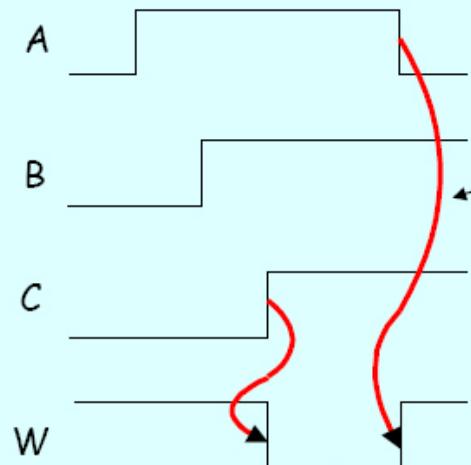
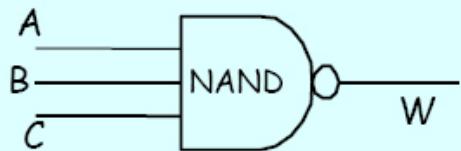
Symbol	Input	Output
—	The input must be valid	The output will be valid
—	If the input were to fall	Then the output will fall
—	If the input were to rise	Then the output will rise
X X X X X	Don't care, it will work regardless	Don't know, the output value is indeterminate
—	Nonsense	High impedance, tristate, HiZ, Not driven, floating

### Example Timing Diagram



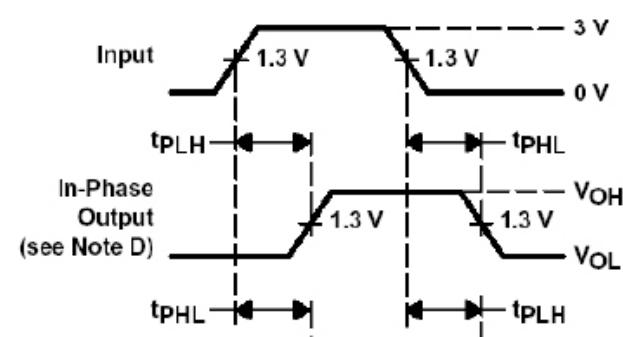
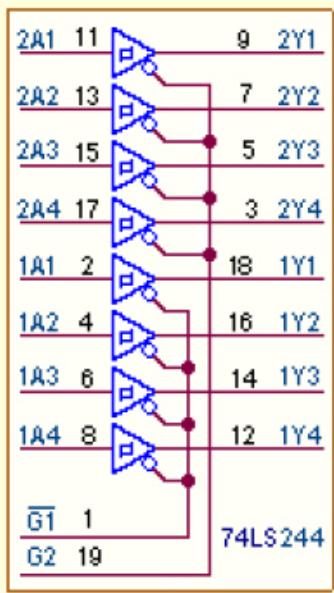
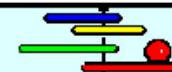
- **Functional Timing Diagram (idealized)**

- assumes zero delays
- simply demonstrates logic relations



**Arrows show cause and effect**  
which input transactions cause which output  
transactions,  
especially in complex timing diagram

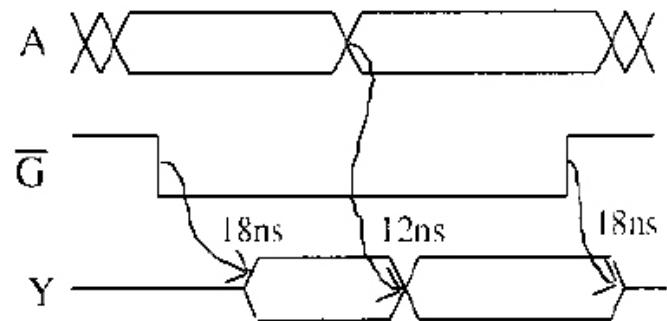
## 74LS244 Buffer 1



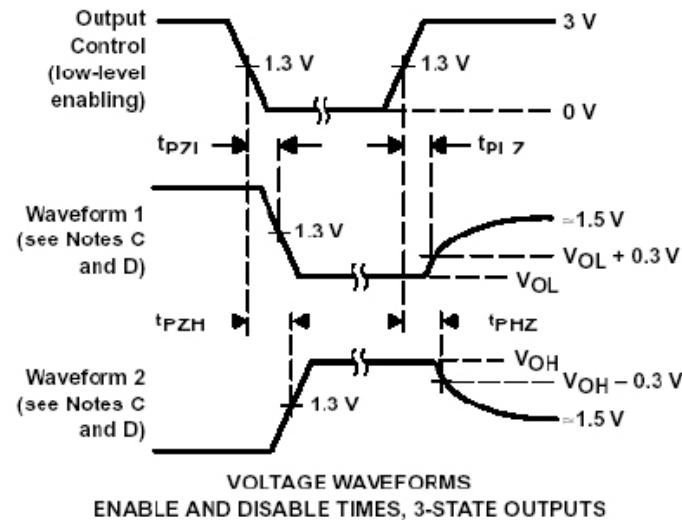
PARAMETER	TEST CONDITIONS	'LS240			'LS241, 'LS244			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
t <sub>PLH</sub>	R <sub>L</sub> = 667 Ω, C <sub>L</sub> = 45 pF				9	14	12	18
t <sub>PHL</sub>					12	18	12	18

- Propagation delay = 12nS (enabled) - typical value  
→ Analysis (design) should use MAXIMUM values for worst case.

## 74LS244 Buffer 2



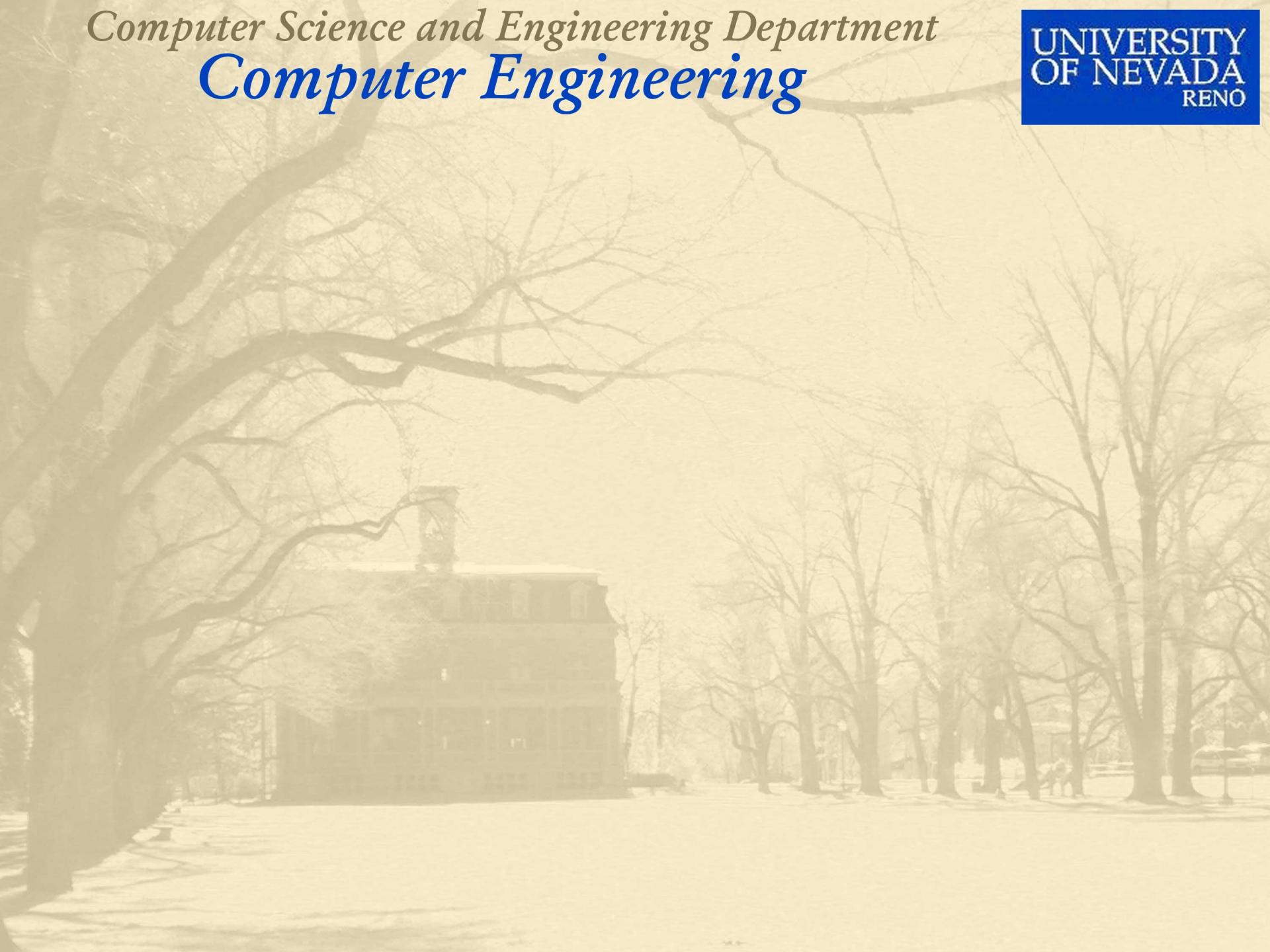
- tri-state buffer (see later slide)
- Propagation delay = 12nS (enabled)



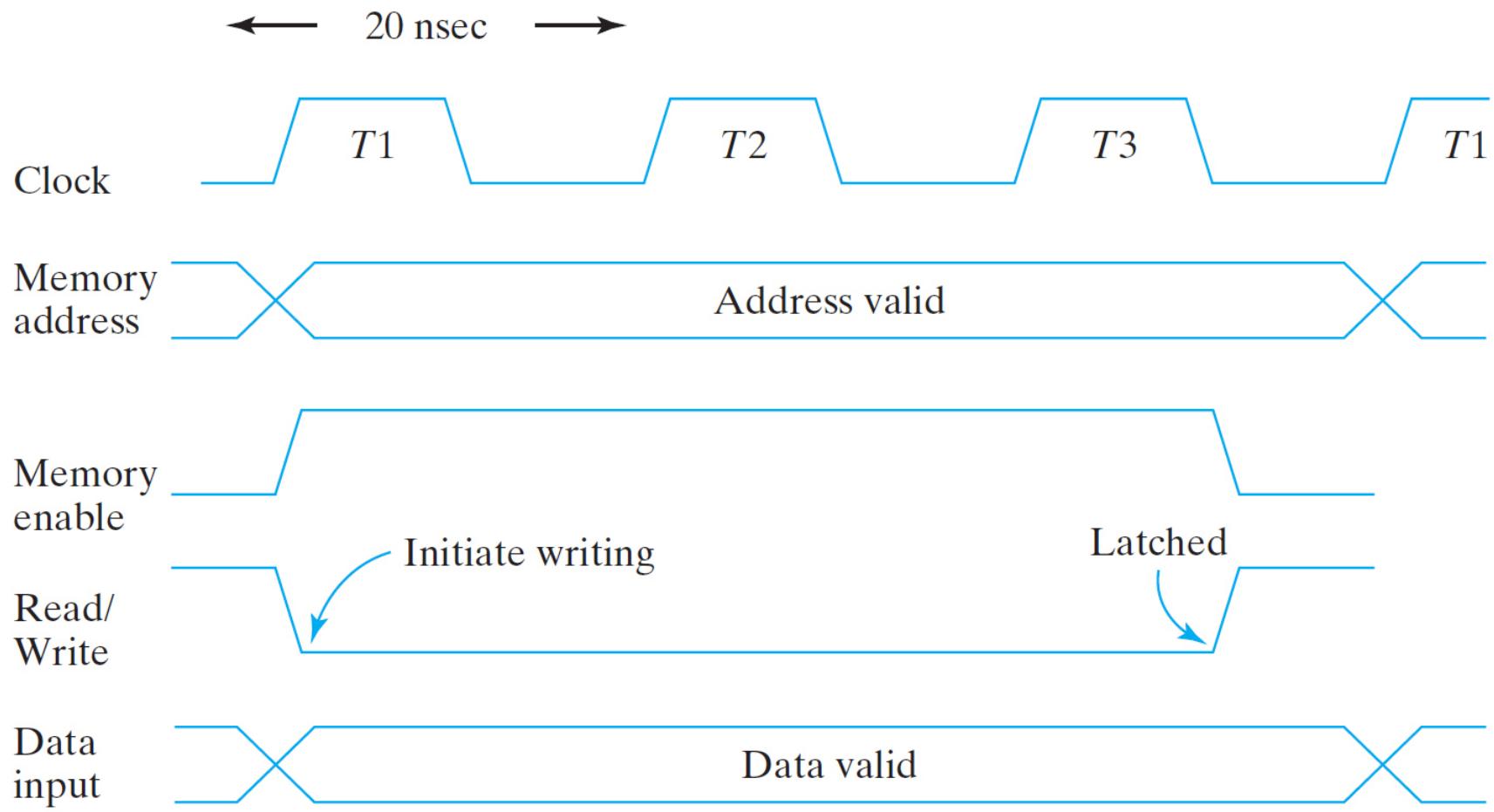
PARAMETER	TEST CONDITIONS	'LS240			'LS241, 'LS244			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
t <sub>PLH</sub>	$R_L = 667 \Omega$ , $C_L = 45 \text{ pF}$	9	14		12	18		ns
t <sub>PHL</sub>		12	18		12	18		
t <sub>PZL</sub>	$R_L = 667 \Omega$ , $C_L = 45 \text{ pF}$	20	30		20	30		ns
t <sub>PZH</sub>		15	23		15	23		
t <sub>PLZ</sub>	$R_L = 667 \Omega$ , $C_L = 5 \text{ pF}$	10	20		10	20		ns
t <sub>PHZ</sub>		15	25		15	25		

*Computer Science and Engineering Department*  
**Computer Engineering**

UNIVERSITY  
OF NEVADA  
RENO

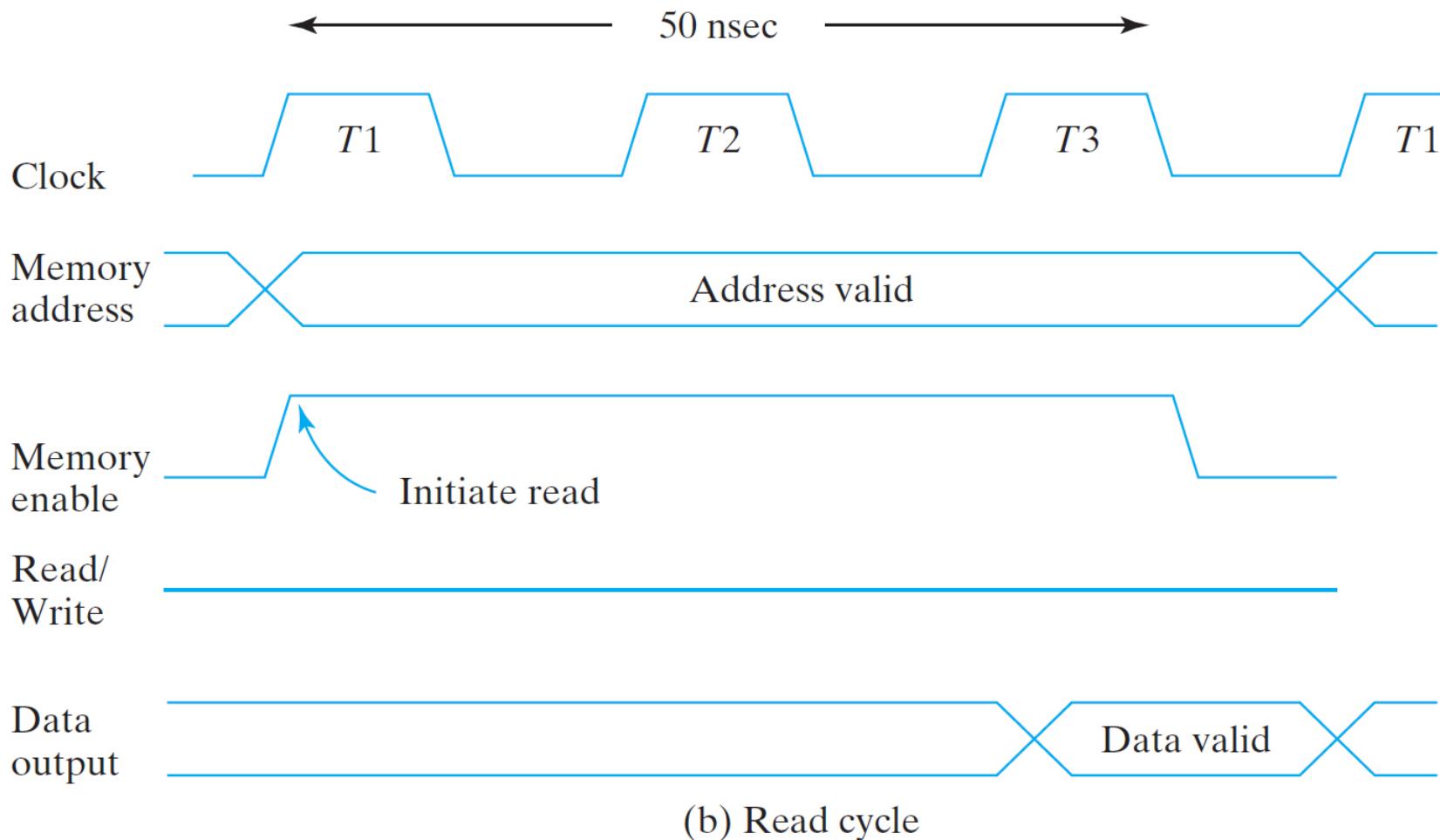


*Computer Science and Engineering Department*  
**Computer Engineering**



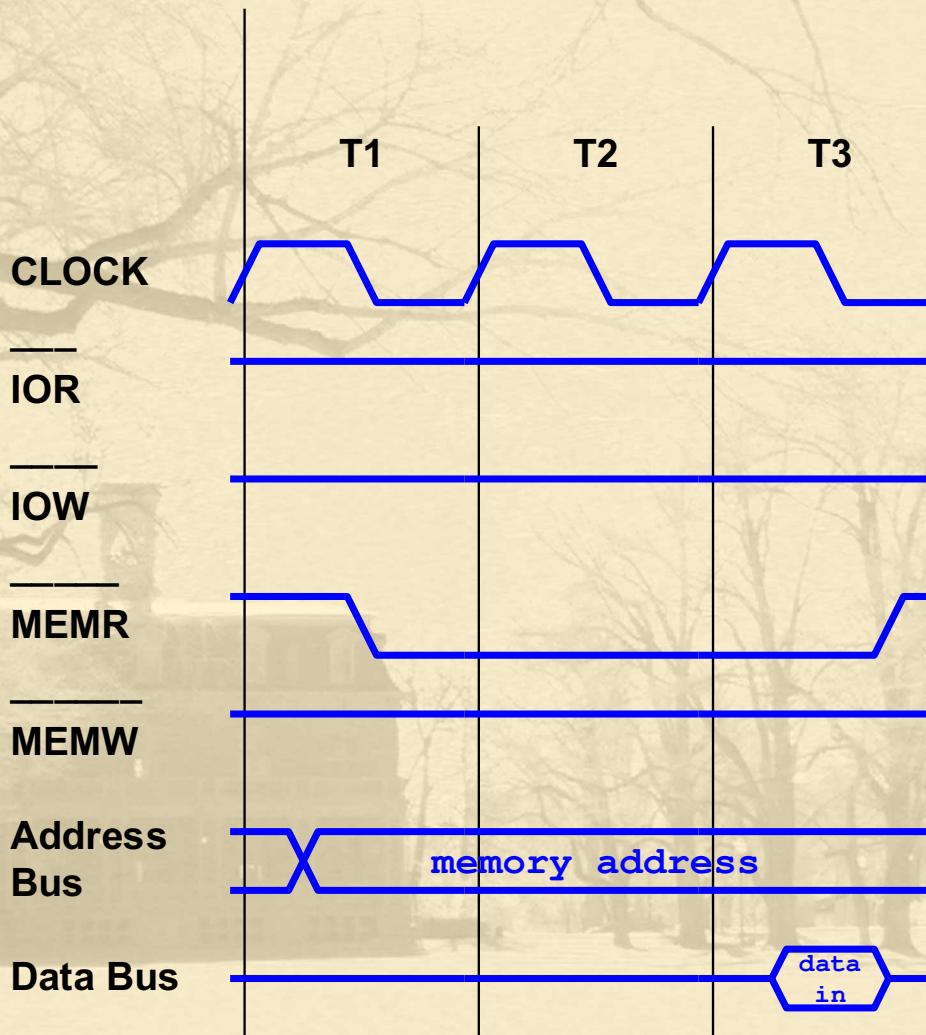
(a) Write cycle

*Computer Science and Engineering Department*  
**Computer Engineering**

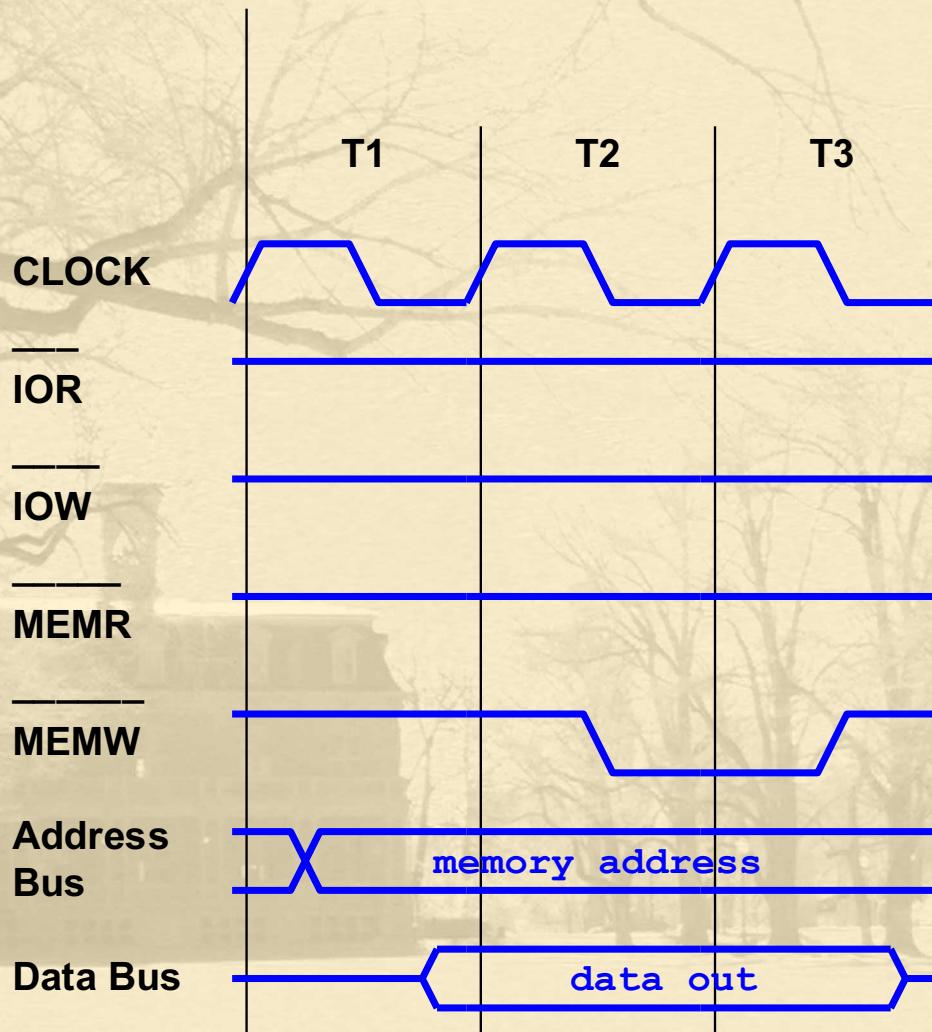


**FIGURE 7.4**  
Memory cycle timing waveforms

*Computer Science and Engineering Department*  
**Computer Engineering**

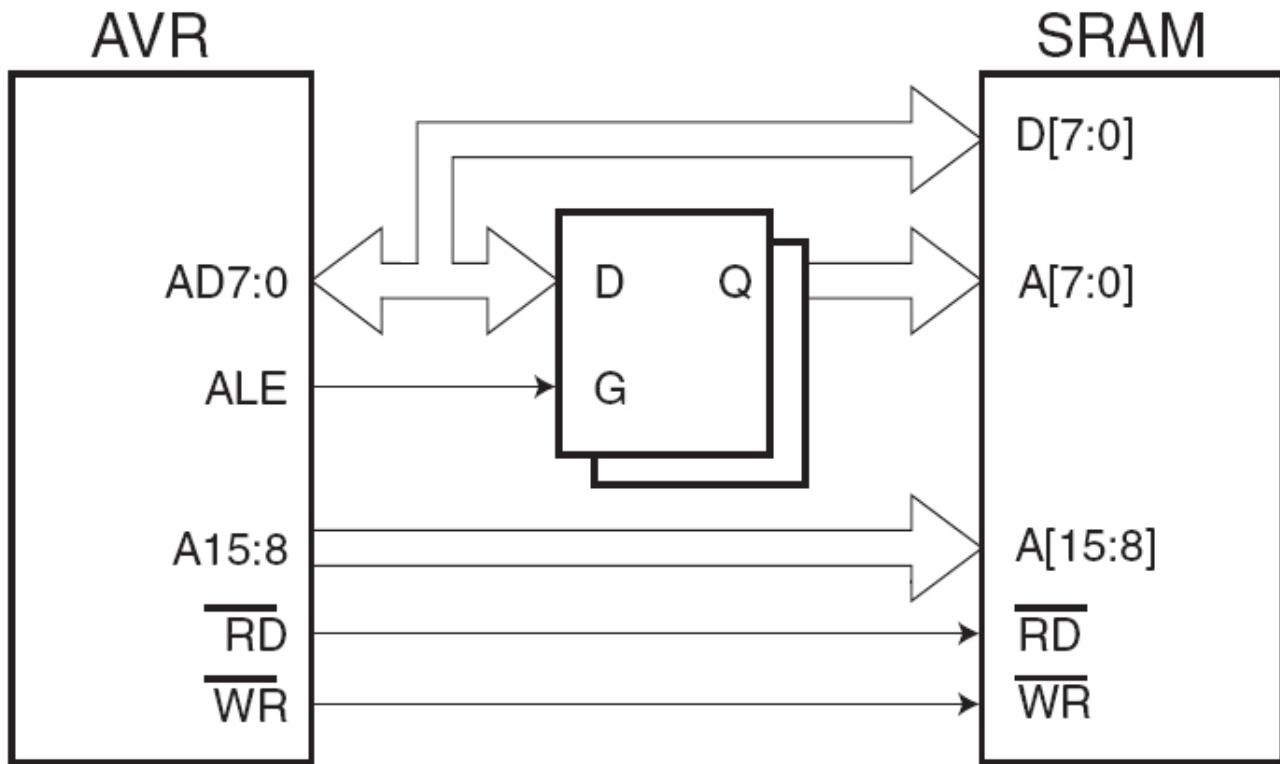


*Computer Science and Engineering Department*  
**Computer Engineering**



*Computer Science and Engineering Department*  
**Computer Engineering**

**Figure 9-2.** External SRAM Connected to the AVR



# Computer Science and Engineering Department

## Computer Engineering

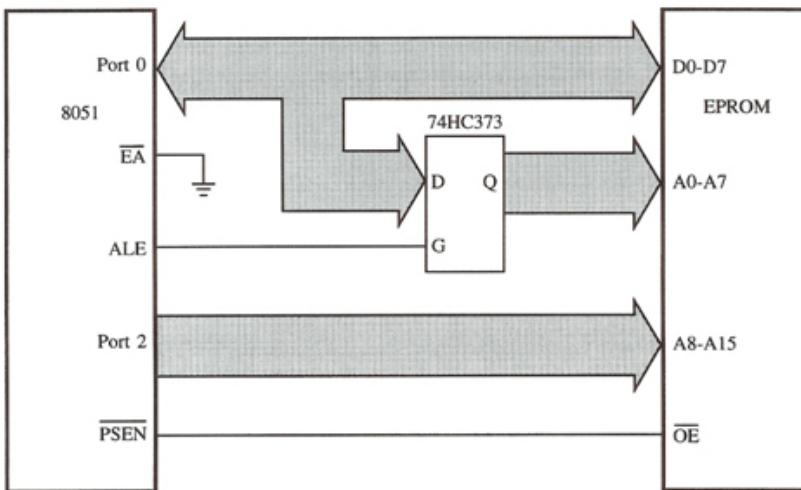
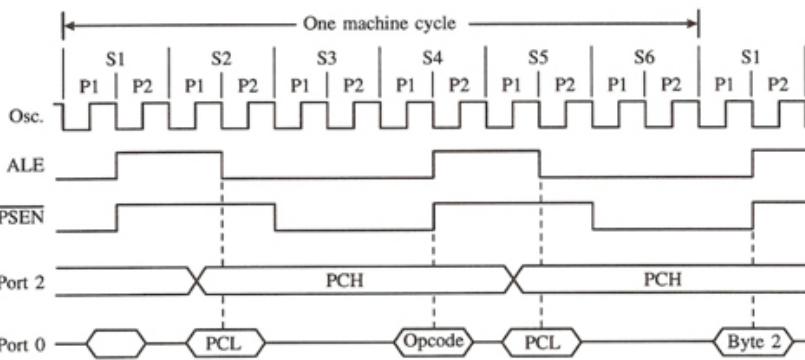


FIGURE 2–8  
Accessing external code memory



Note: PCH = Program counter high byte  
PCL = Program counter low byte

FIGURE 2–9  
Read timing for external code memory

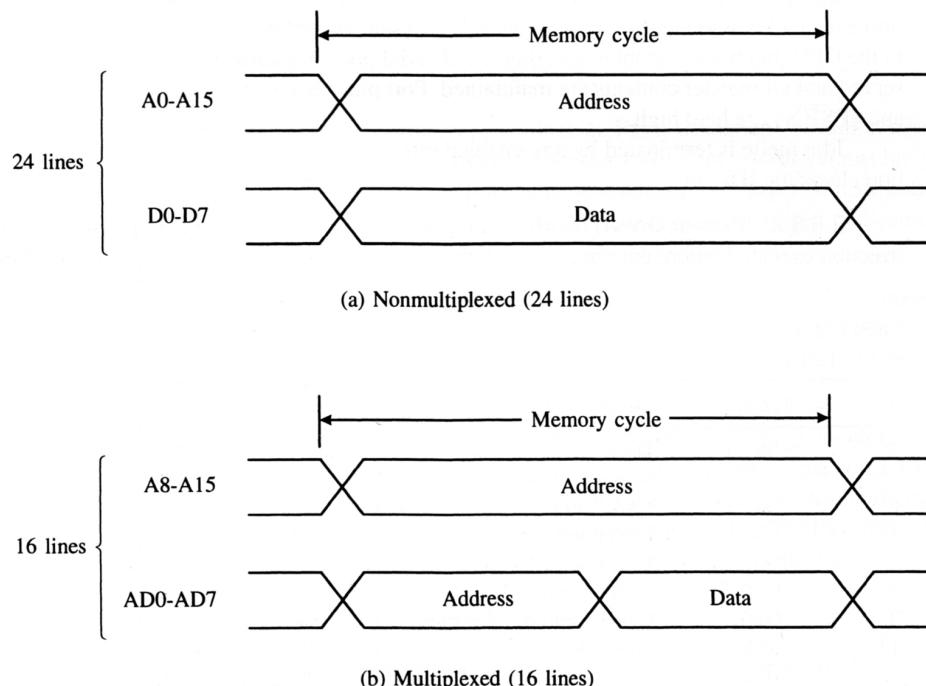
# Computer Science and Engineering Department

# Computer Engineering

## 2.7 EXTERNAL MEMORY

It is important that microcontrollers have expansion capabilities beyond the on-chip resources to avoid a potential design bottleneck. If any resources must be expanded (memory, I/O, etc.), then the capability must exist. The MCS-51™ architecture provides this in the form of a 64K external code memory space and a 64K external data memory space. Extra ROM and RAM can be added as needed. Peripheral interface ICs can also be added to expand the I/O capability. These become part of the external data memory space using memory-mapped I/O.

When external memory is used, Port 0 is unavailable as an I/O port. It becomes a multiplexed address (A0–A7) and data (D0–D7) bus, with ALE latching the low-byte of the address at the beginning of each external memory cycle. Port 2 is usually (but not always) employed for the high-byte of the address bus.



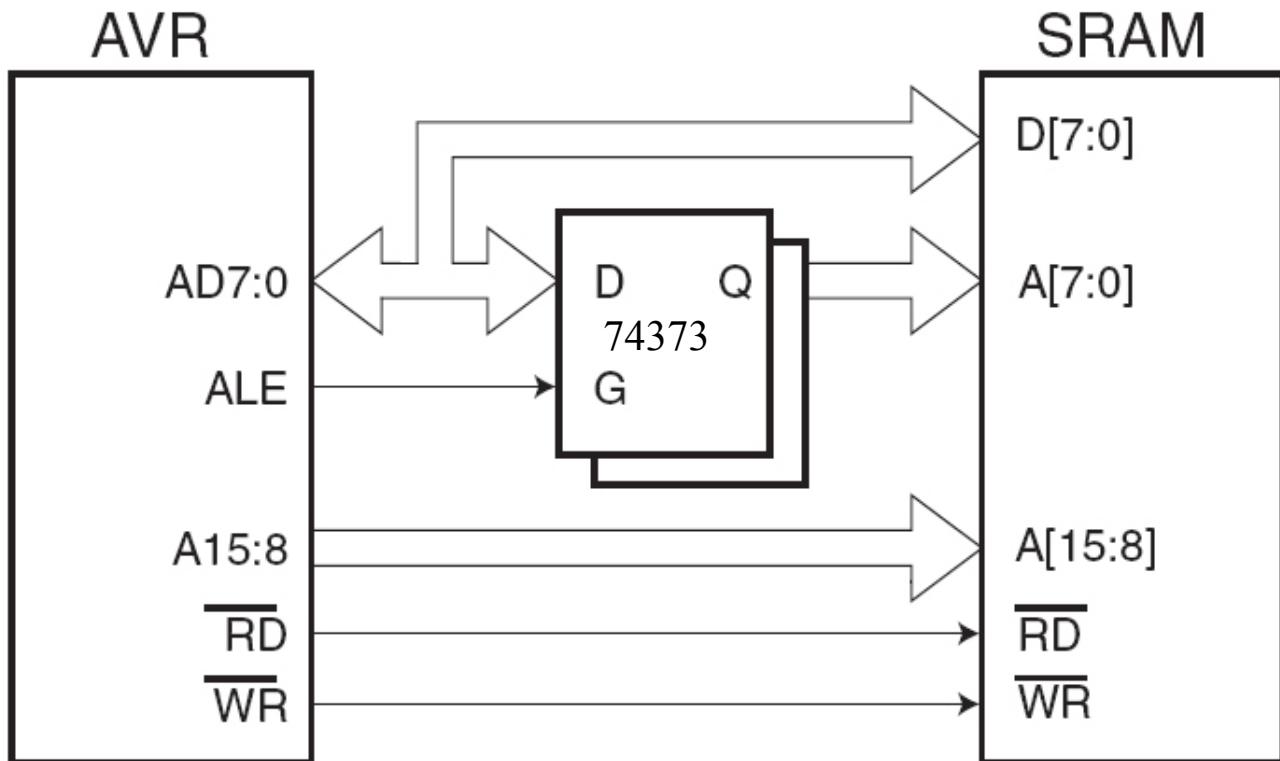
**FIGURE 2-8**

Multiplexing the address bus (low-byte) and data bus

*Computer Science and Engineering Department*  
**Computer Engineering**

UNIVERSITY  
OF NEVADA  
RENO

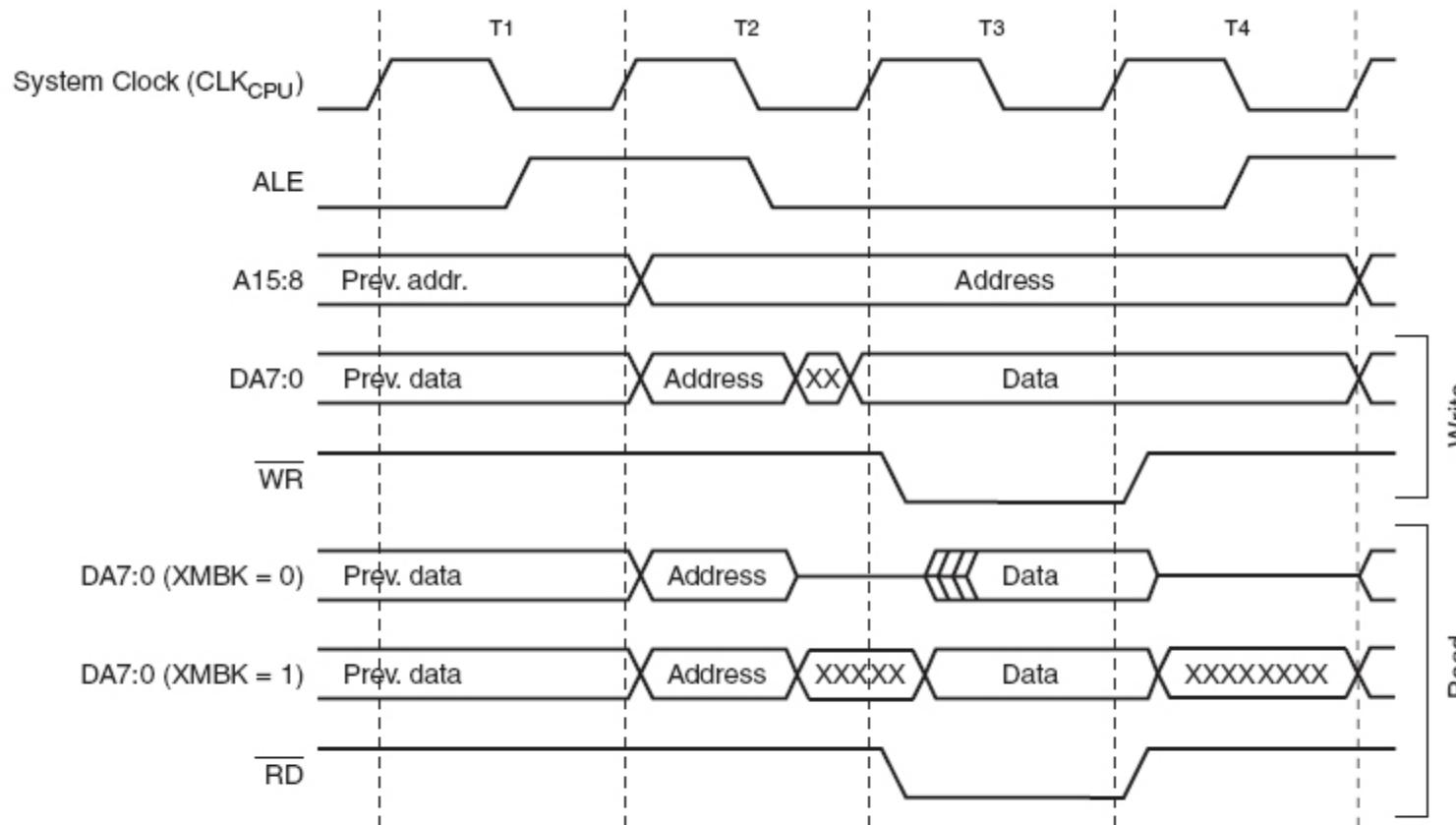
**Figure 9-2.** External SRAM Connected to the AVR



# Computer Science and Engineering Department

## Computer Engineering

Figure 9-3. External Data Memory Cycles without Wait-state (SRWn1=0 and SRWn0=0)<sup>(1)</sup>



Note: 1. SRWn1 = SRW11 (upper sector) or SRW01 (lower sector), SRWn0 = SRW10 (upper sector) or SRW00 (lower sector). The ALE pulse in period T4 is only present if the next instruction accesses the RAM (internal or external).

# Computer Science and Engineering Department

# Computer Engineering

Figure 9-2. External SRAM Connected to the AVR

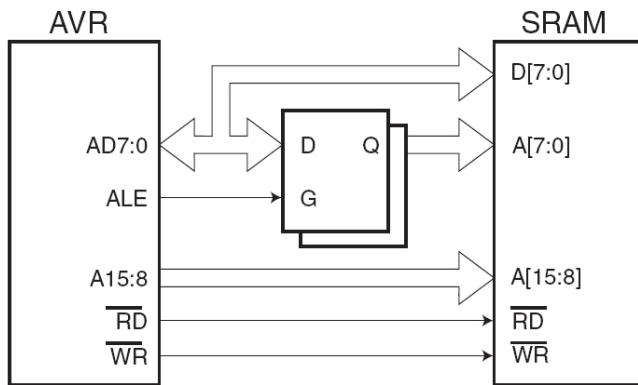
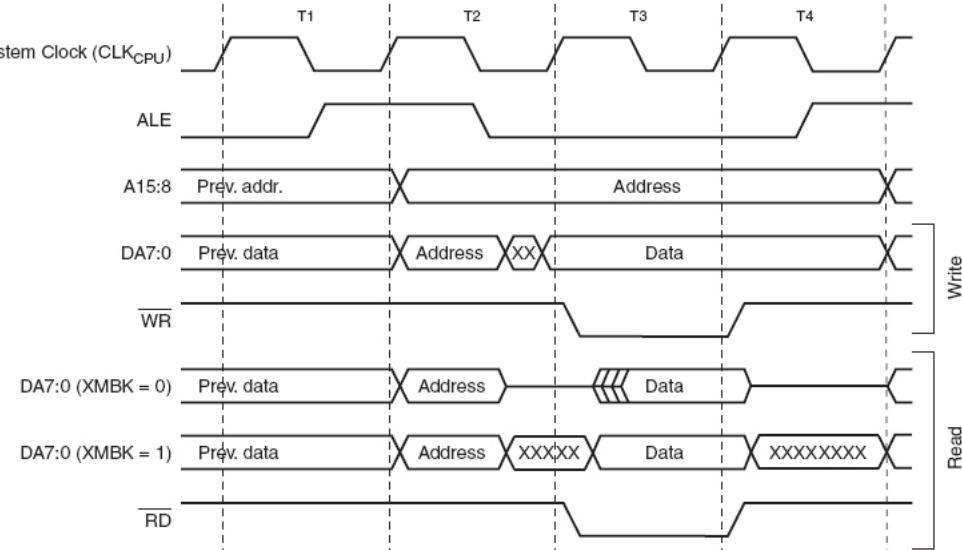


Figure 9-3. External Data Memory Cycles without Wait-state (SRWn1=0 and SRWn0=0)<sup>(1)</sup>

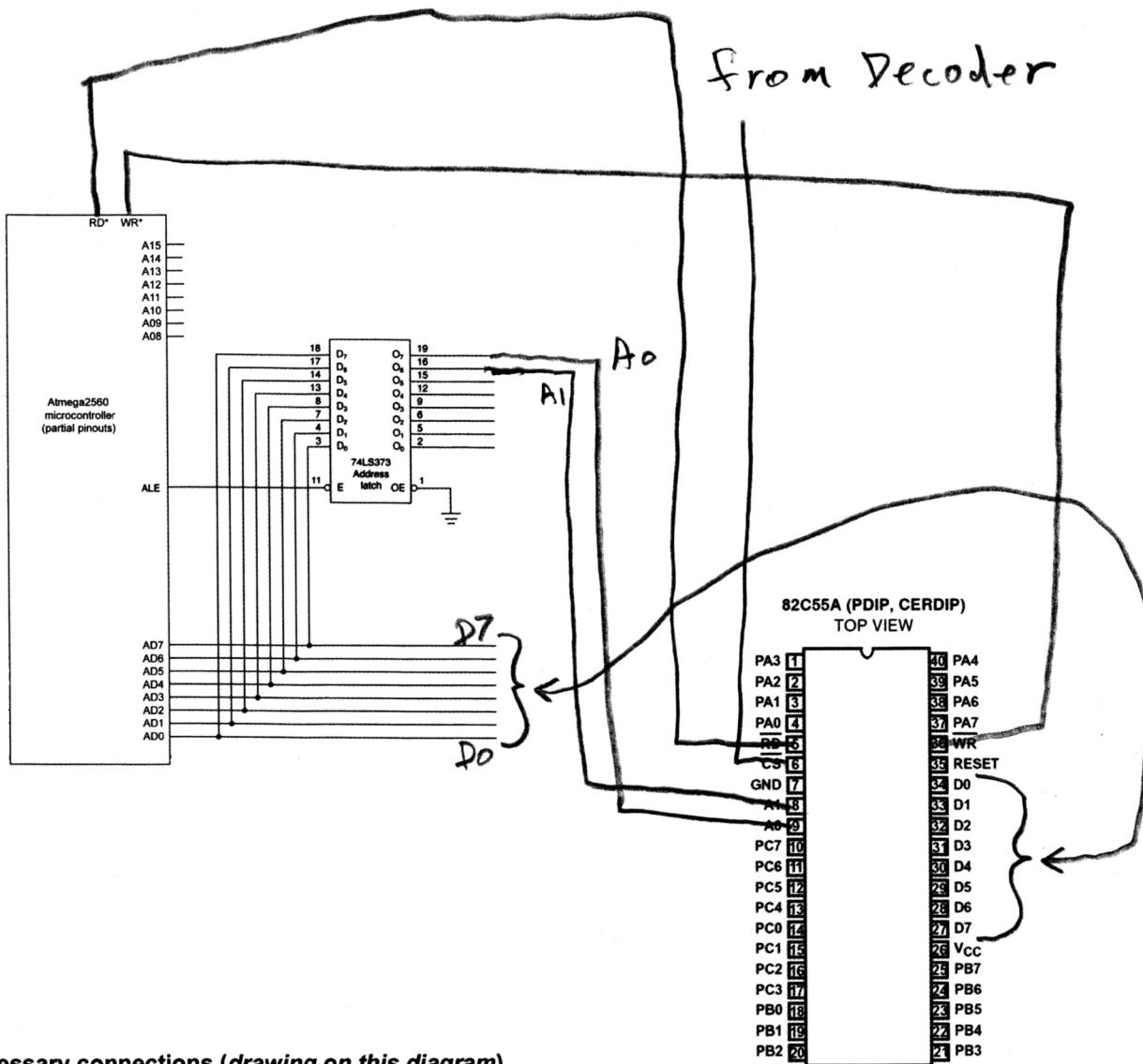


Note: 1. SRWn1 = SRW11 (upper sector) or SRW01 (lower sector), SRWn0 = SRW10 (upper sector) or SRW00 (lower sector). The ALE pulse in period T4 is only present if the next instruction accesses the RAM (internal or external).

# Computer Science and Engineering Department

# Computer Engineering

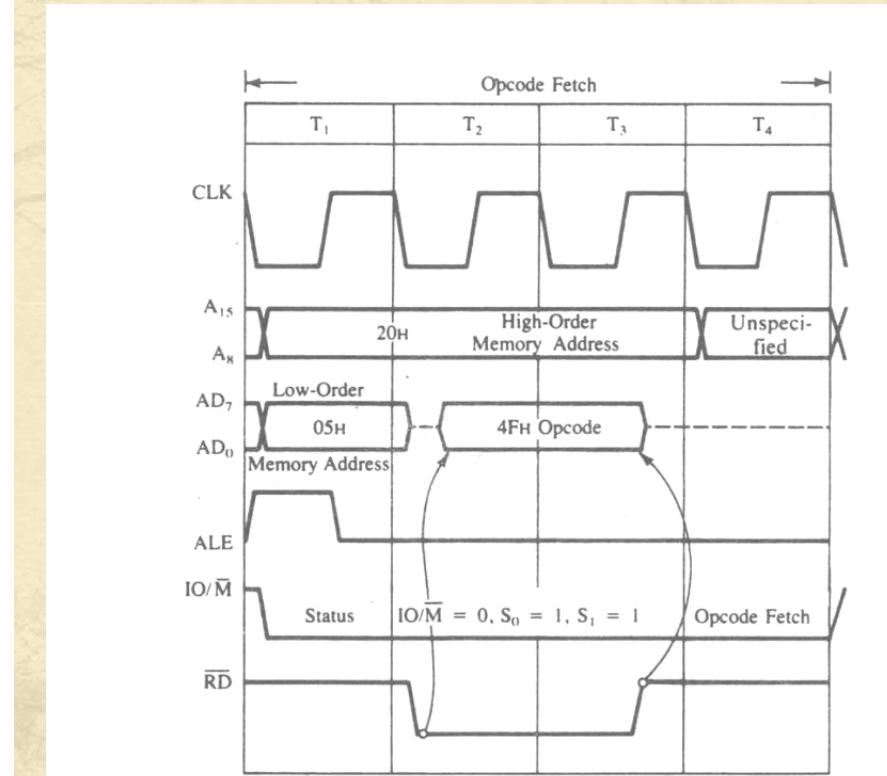
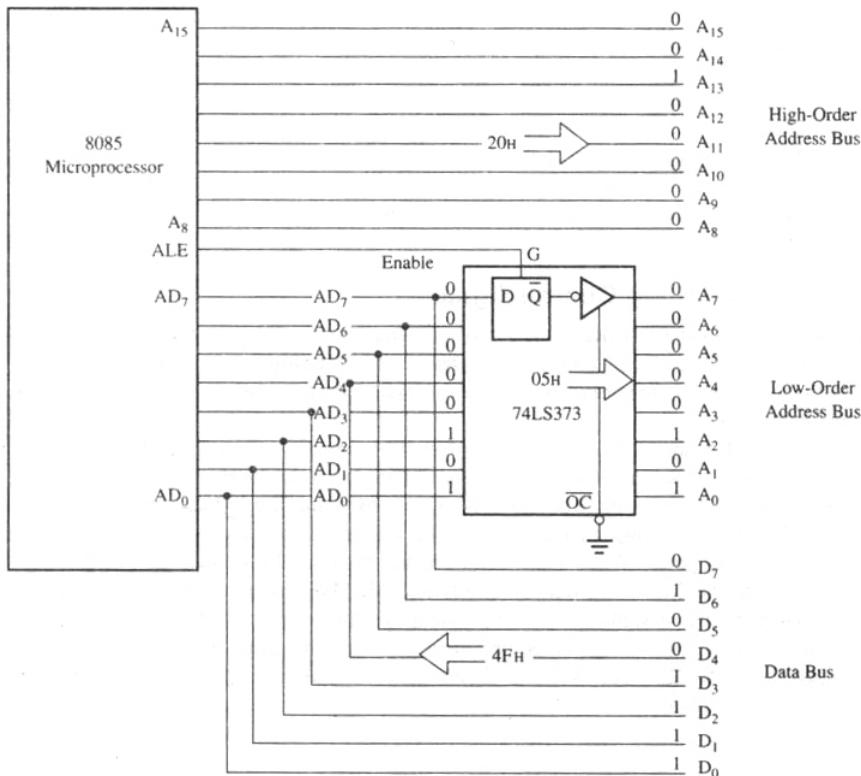
UNIVERSITY  
OF NEVADA  
RENO



# Computer Science and Engineering Department

# Computer Engineering

UNIVERSITY  
OF NEVADA  
RENO



*Computer Science and Engineering Department*  
**Computer Engineering**

UNIVERSITY  
OF NEVADA  
RENO

