Digital Design

Lecture of week 11 part 2

Dr Manal Tantawi

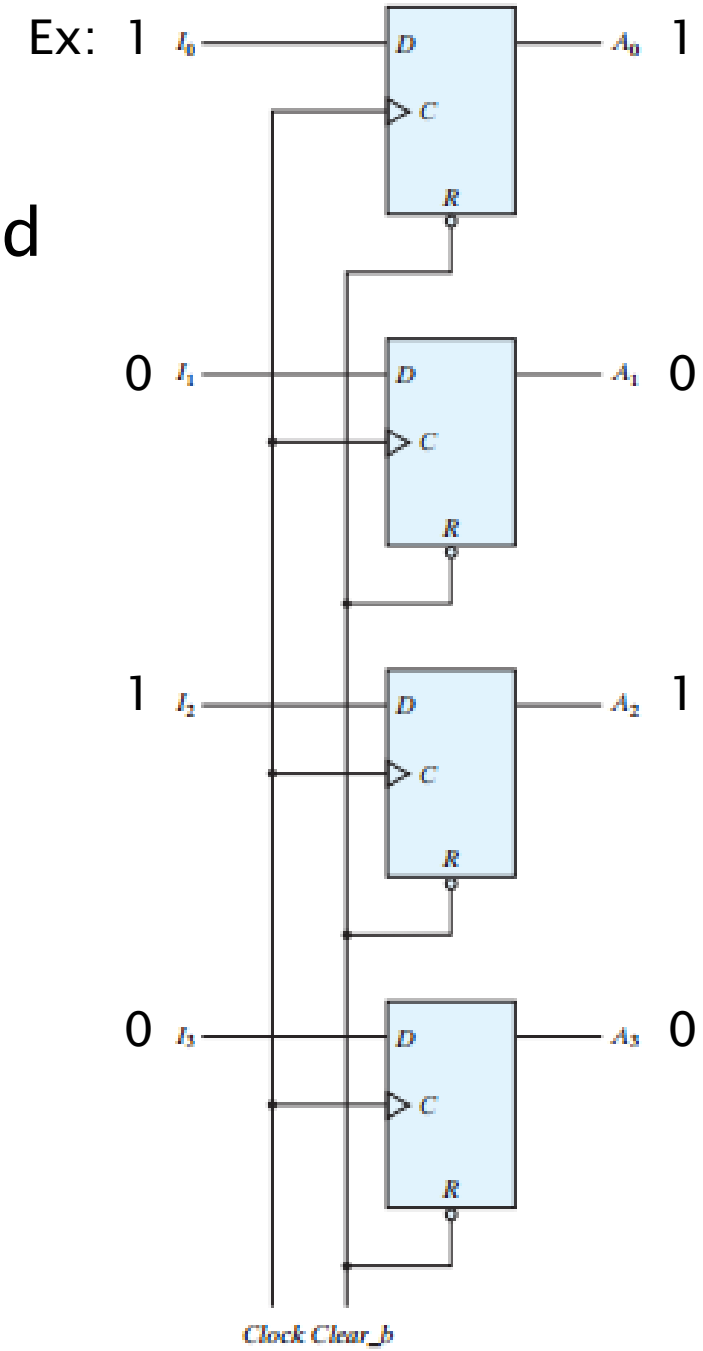
Computer Memories

1) Registers √√
2) RAM (self read not included in final exam)
3) ROM (self read not included in final exam)

Registers

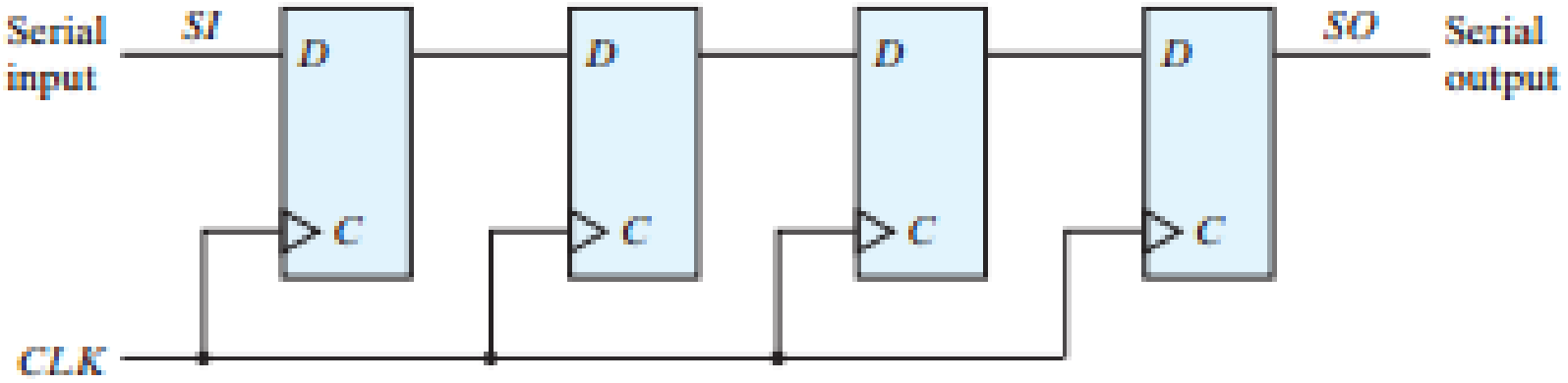
- 4 bit register with parallel load

Only one clock is needed to upload an input



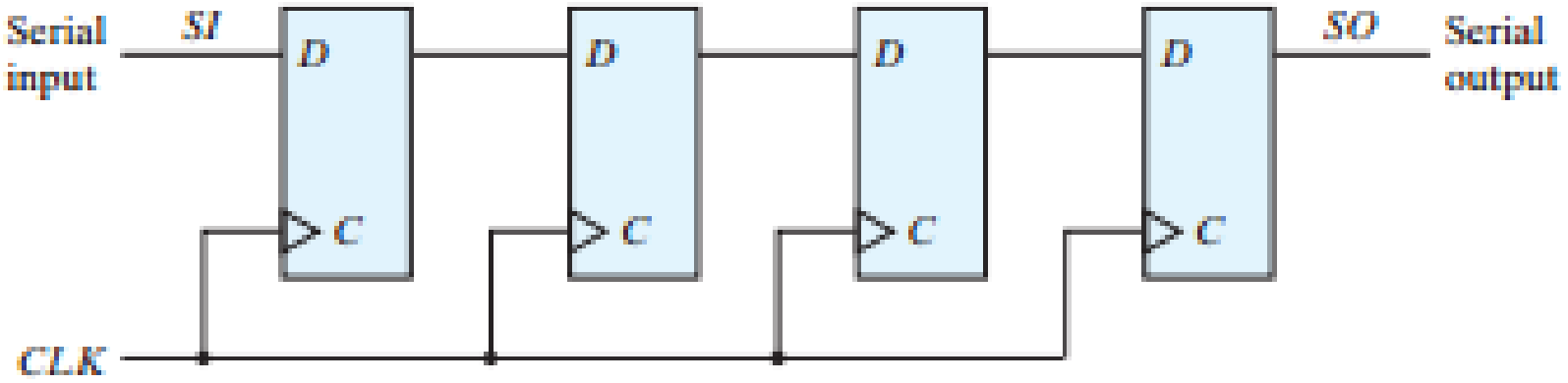
Registers

- 4 bit shift register (right or left)



Registers

- 4 bit shift register (right)

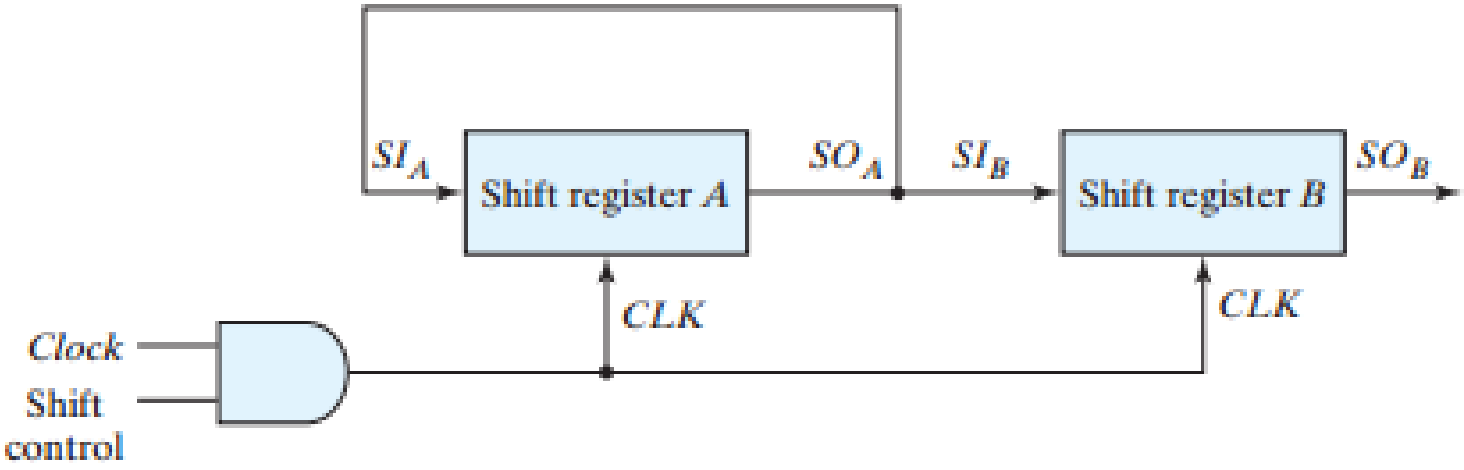


	1	0	1	0
CLK 1	0	1	0	1
CLK 2	0	0	1	0
CLK 3	0	0	0	1
CLK 4	0	0	0	0

4 clocks are needed

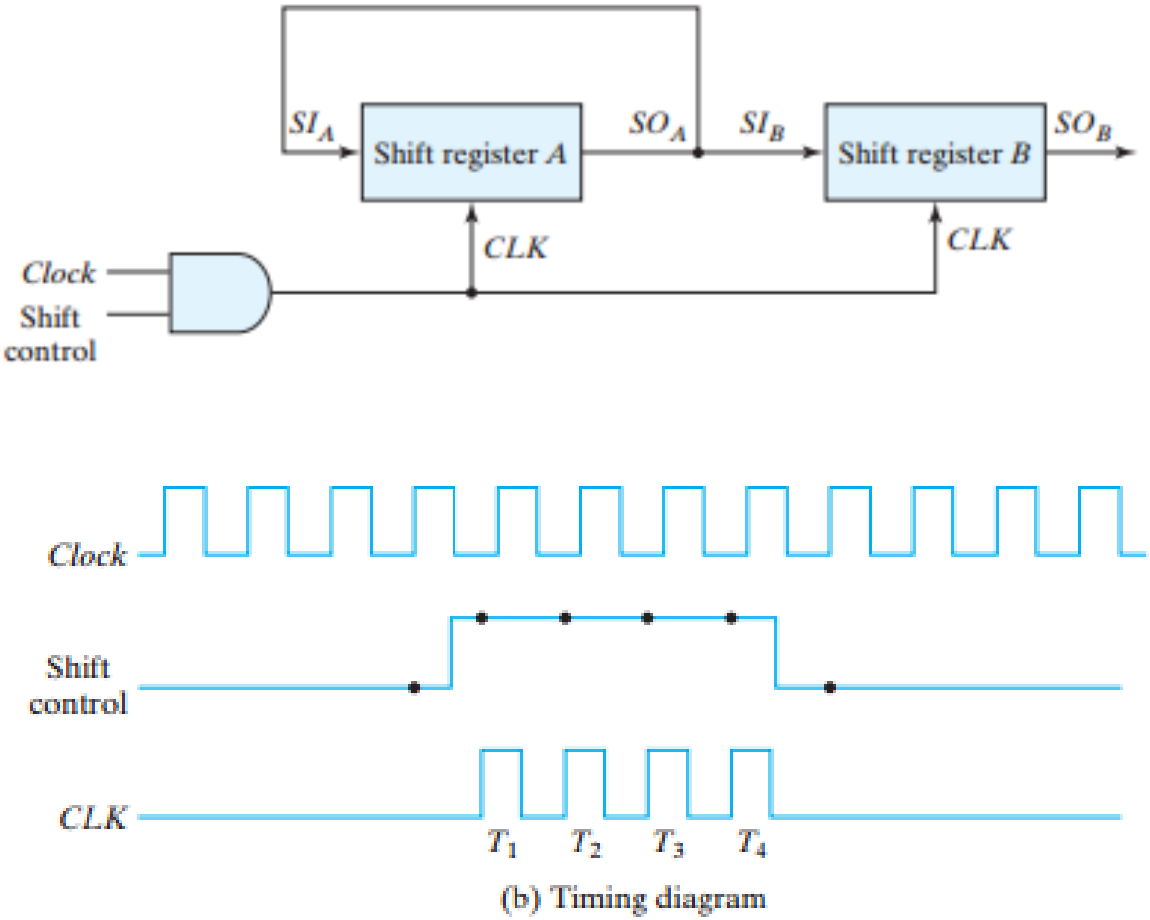
Shift register

- Serial transfer (B = A)



Shift register

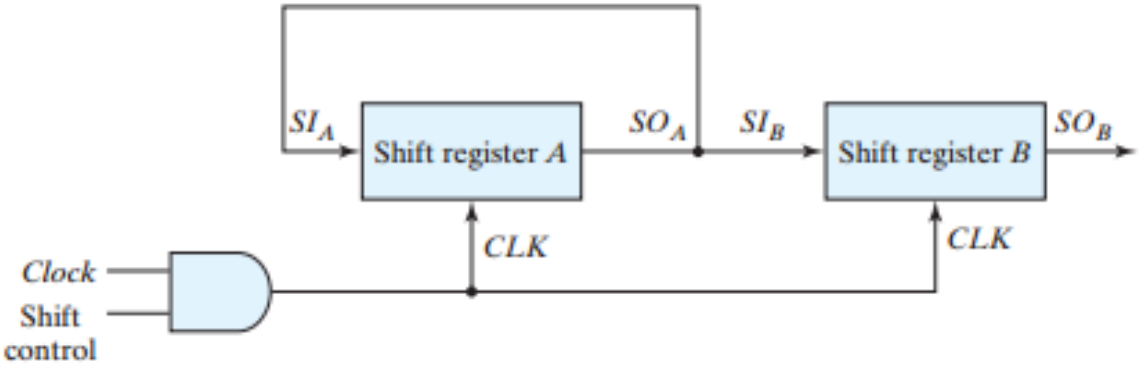
- Serial transfer (B = A)



(b) Timing diagram

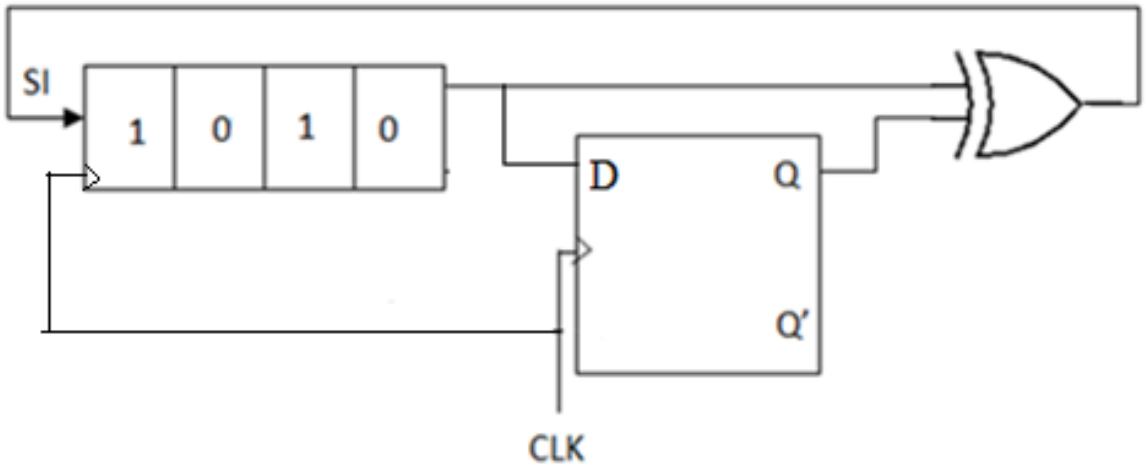
Shift register

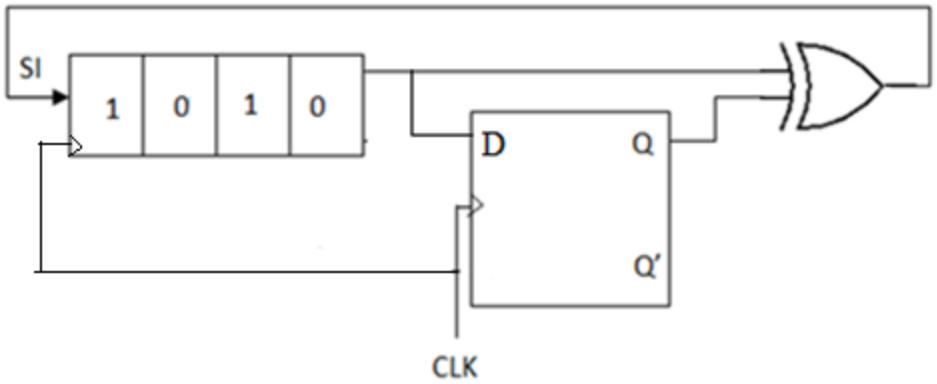
- Serial transfer (B = A)



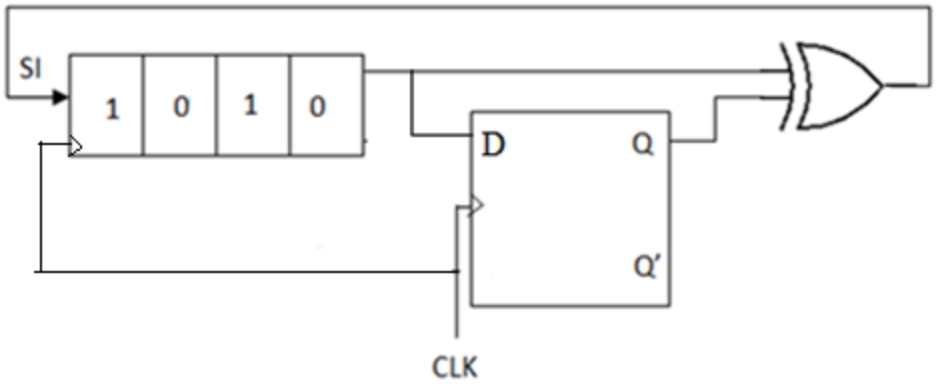
Timing Pulse	Shift Register *A*				Shift Register *B*			
Initial value	1	0	1	1	0	0	1	0
After T_1	1	1	0	1	1	0	0	1
After T_2	1	1	1	0	1	1	0	0
After T_3	0	1	1	1	0	1	1	0
After T_4	1	0	1	1	1	0	1	1

Find the content of the register after 4 clocks (initial value for Q =0)

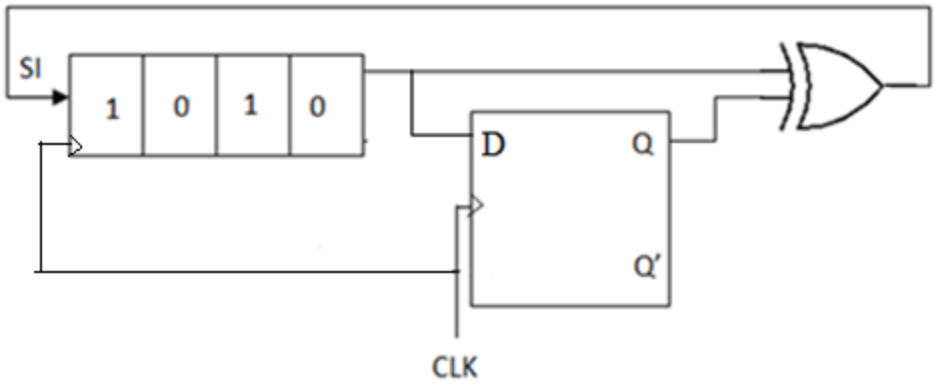




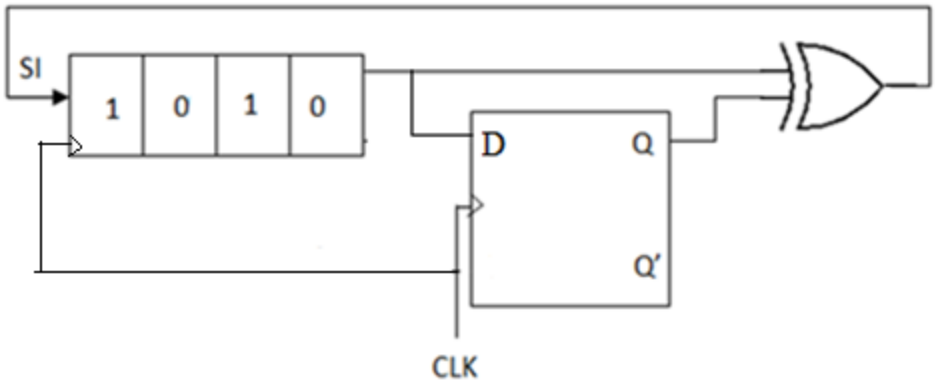
clocks	SI Last bit ⊕ Q	Register content	D =Last bit	Q
	0	1 0 1 **0**	0	**0**
CLK 1	1	0 1 0 **1**	1	**0**



clocks	SI Last bit ⊕ Q	Register content	D =Last bit	Q
	0	1 0 1 **0**	0	**0**
CLK 1	1	0 1 0 **1**	1	**0**
CLK 2	1	1 0 1 **0**	0	**1**



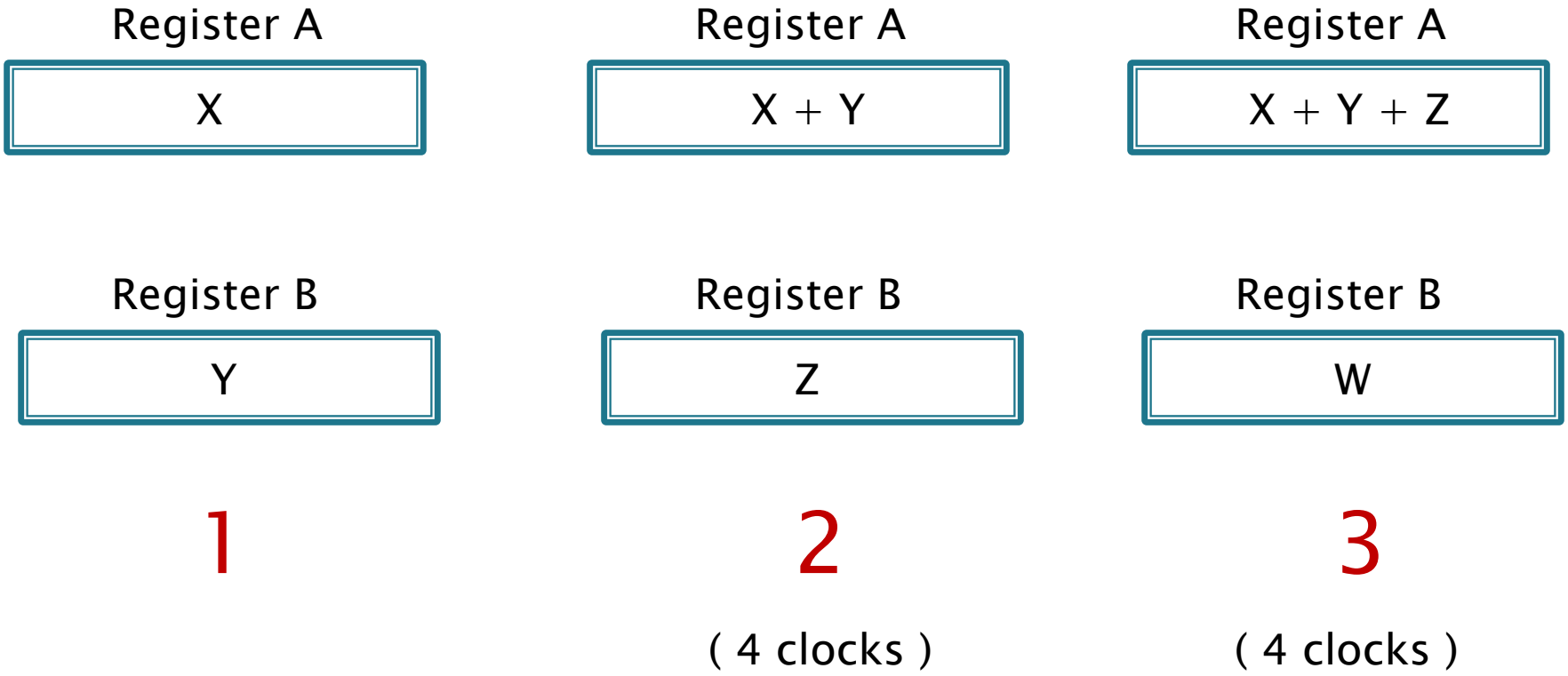
clocks	SI Last bit ⊕ Q	Register content	D =Last bit	Q
	0	1 0 1 **0**	0	**0**
CLK 1	1	0 1 0 **1**	1	**0**
CLK 2	1	1 0 1 **0**	0	**1**
CLK 3	1	1 1 0 **1**	1	**0**



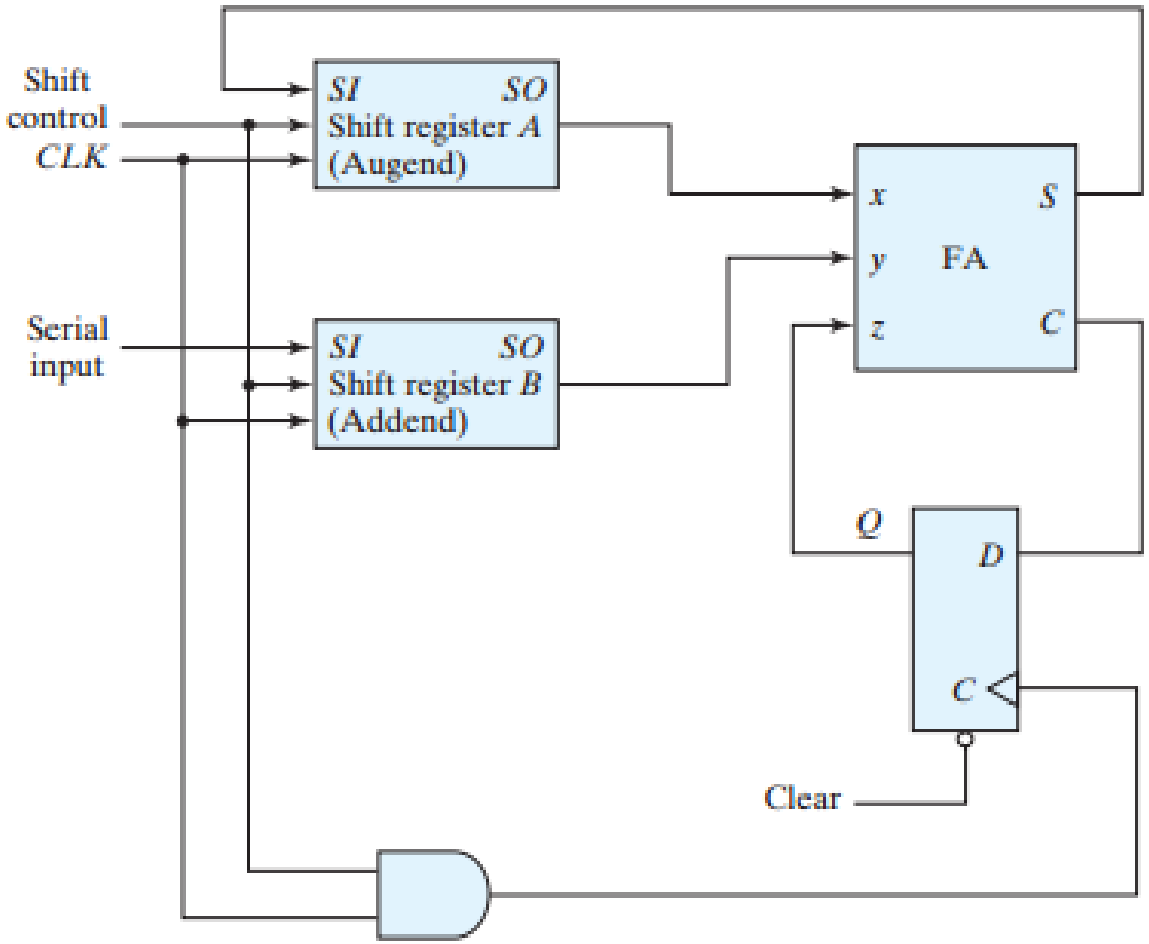
clocks	SI Last bit ⊕ Q	Register content	D =Last bit	Q
	0	1 0 1 **0**	0	**0**
CLK 1	1	0 1 0 **1**	1	**0**
CLK 2	1	1 0 1 **0**	0	**1**
CLK 3	1	1 1 0 **1**	1	**0**
CLK 4		1 1 1 **0**		**1**

Serial Adder

X + Y + Z + W (each of them is 4 bits)



Serial Adder



Serial Adder

X = 0 1 0 1 Y = 0 0 1 1 Z= 0 1 1 1

Register A Register B

Clocks	SI_A =S	Register A	SI_B	Register B	D =C	Q	C	S
	0	0 1 0 <u>1</u>	1	0 0 1 <u>1</u>	1	<u>0</u>	1	0

Serial Adder

X = 0 1 0 1 Y = 0 0 1 1 Z= 0 1 1 1

Register A Register B

Clocks	SI_A =S	Register A	SI_B	Register B	D =C	Q	C	S
	0	0 1 0 **1**	1	0 0 1 **1**	1	**0**	1	0
CLK1	0	0 0 1 **0**	1	1 0 0 **1**	1	**1**	1	0

Serial Adder

X = 0 1 0 1 Y = 0 0 1 1 Z= 0 1 1 1

Register A Register B

Clocks	SI_A =S	Register A	SI_B	Register B	D =C	Q	C	S
	0	0 1 0 **1**	1	0 0 1 **1**	1	**0**	1	0
CLK1	0	0 0 1 **0**	1	1 0 0 **1**	1	**1**	1	0
CLK2	0	0 0 0 **1**	1	1 1 0 **0**	1	**1**	1	0

Serial Adder

X = 0 1 0 1 Y = 0 0 1 1 Z= 0 1 1 1

Register A Register B

Clocks	SI_A =S	Register A	SI_B	Register B	D =C	Q	C	S
	0	0 1 0 **1**	1	0 0 1 **1**	1	**0**	1	0
CLK1	0	0 0 1 **0**	1	1 0 0 **1**	1	**1**	1	0
CLK2	0	0 0 0 **1**	1	1 1 0 **0**	1	**1**	1	0
CLK3	1	0 0 0 **0**	0	1 1 1 **0**	0	**1**	0	1

Serial Adder

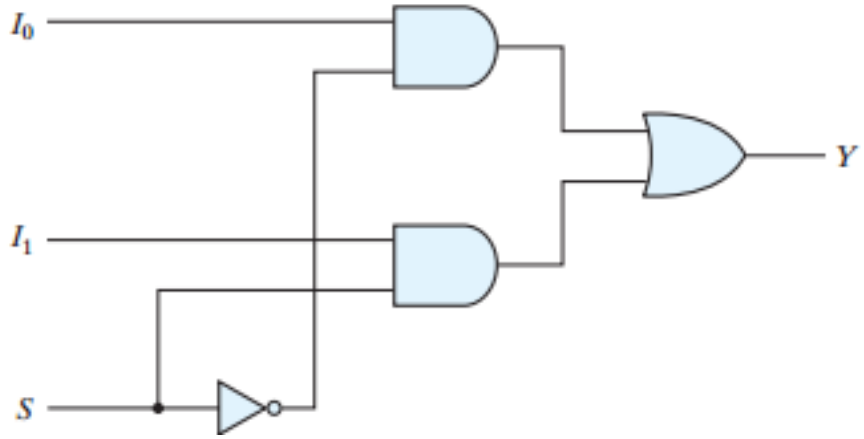
X = 0 1 0 1 Y = 0 0 1 1 Z= 0 1 1 1

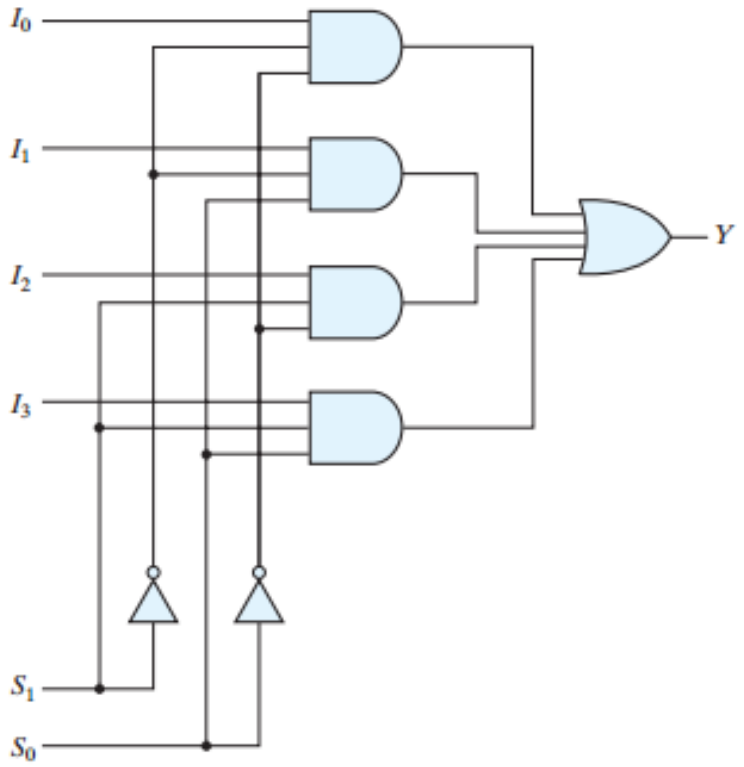
Register A Register B

Clocks	SI_A =S	Register A	SI_B	Register B	D =C	Q	C	S
	0	0 1 0 **1**	1	0 0 1 **1**	1	**0**	1	0
CLK1	0	0 0 1 **0**	1	1 0 0 **1**	1	**1**	1	0
CLK2	0	0 0 0 **1**	1	1 1 0 **0**	1	**1**	1	0
CLK3	1	0 0 0 **0**	0	1 1 1 **0**	0	**1**	0	1
CLK4		1 0 0 **0**		0 1 1 **1**		**0**		

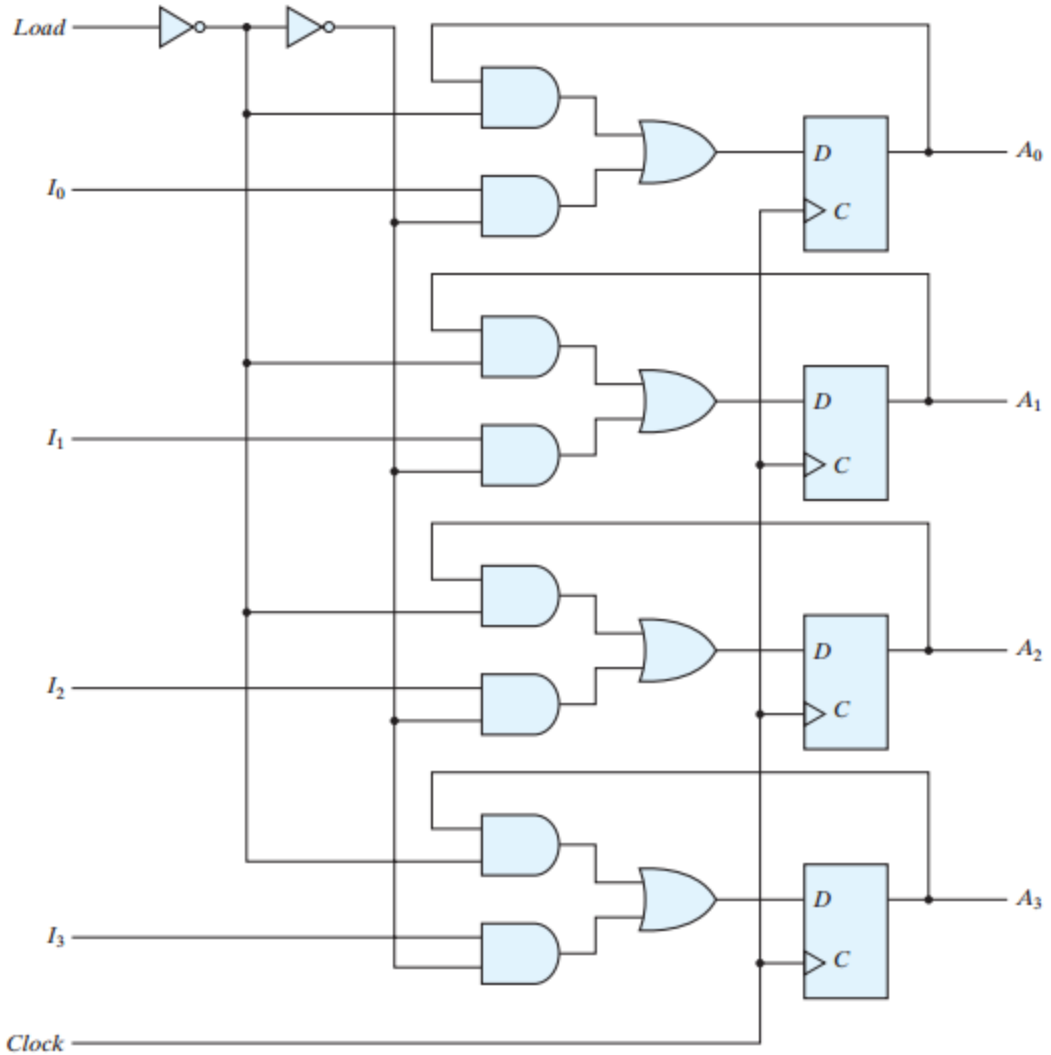
X+ Y z

Selection circuit (remember)



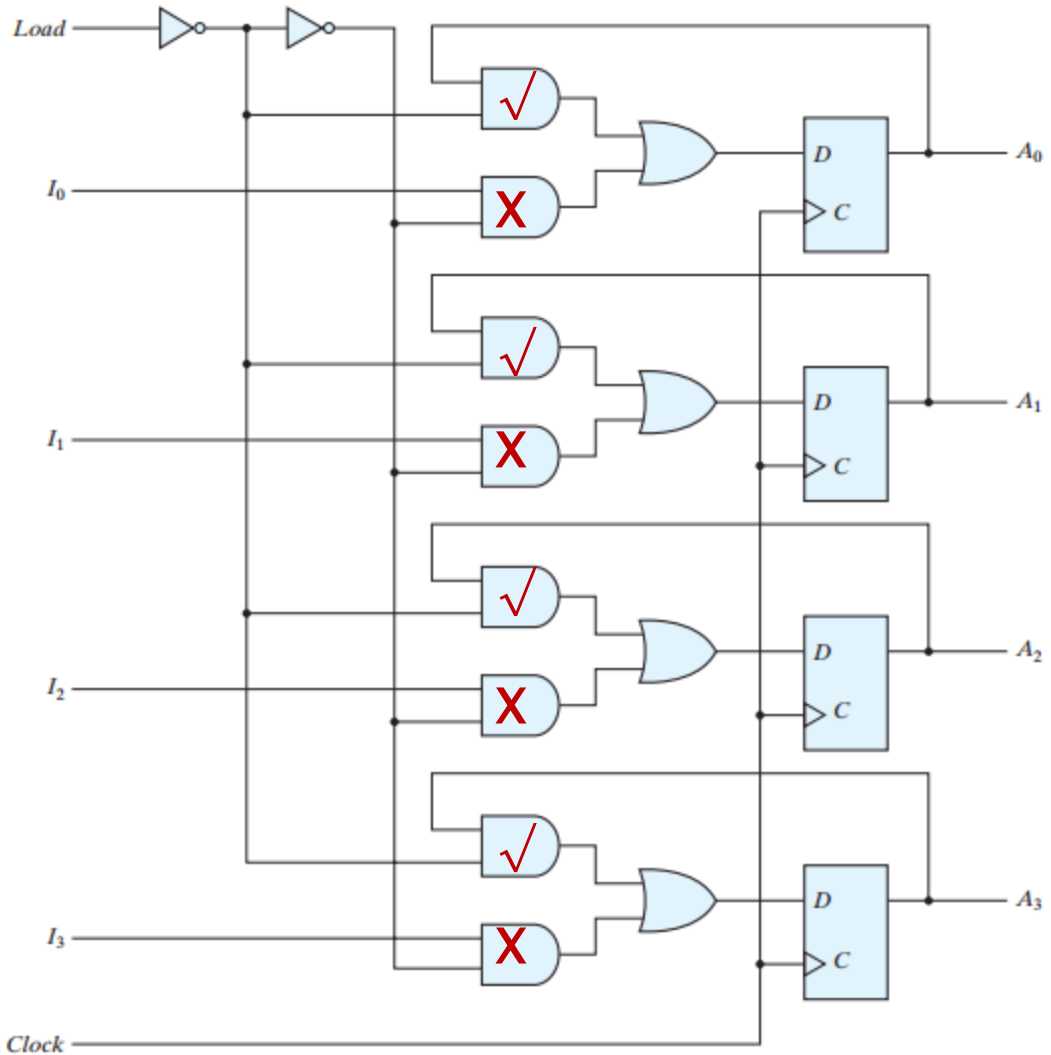


4 bit register with parallel load \ no change



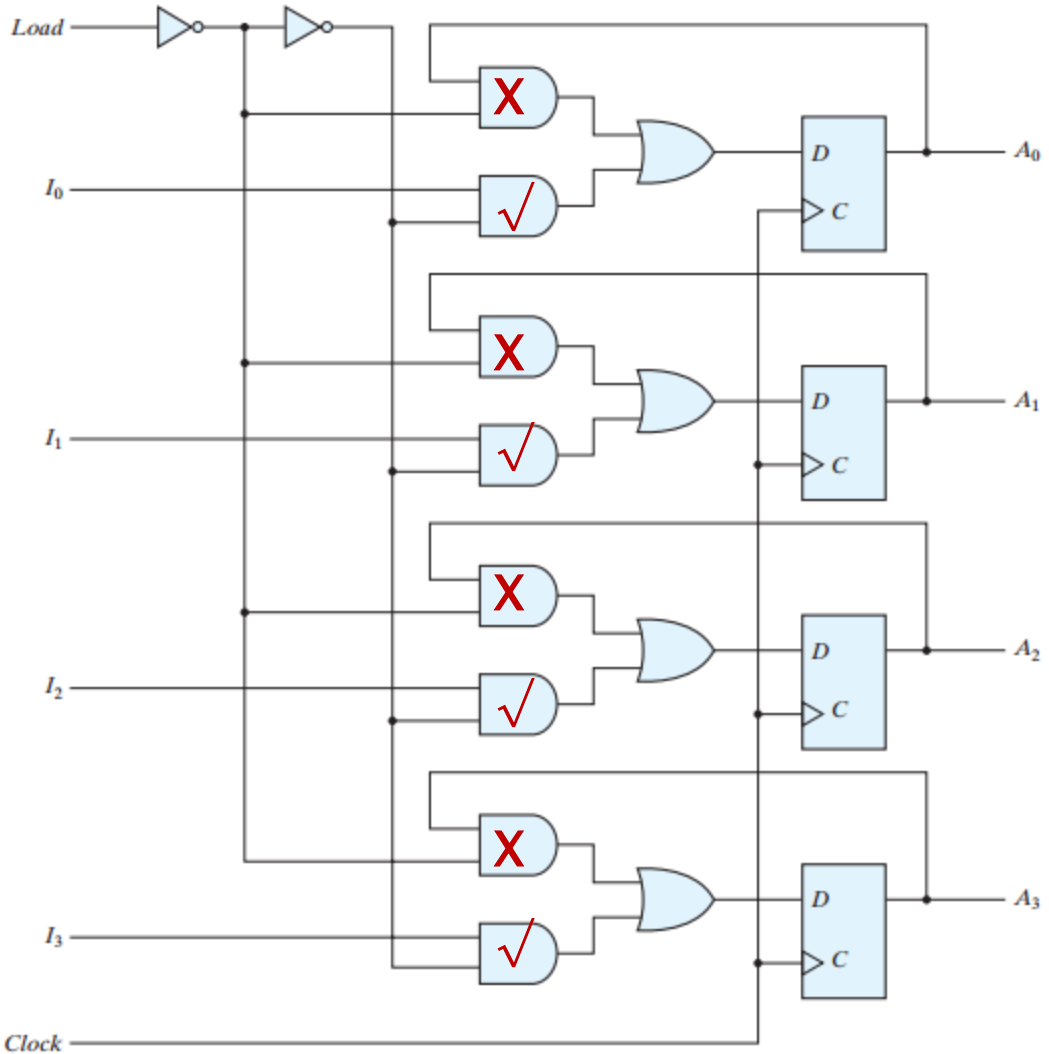
4 bit register with parallel load \ no change

Load = 0
no change

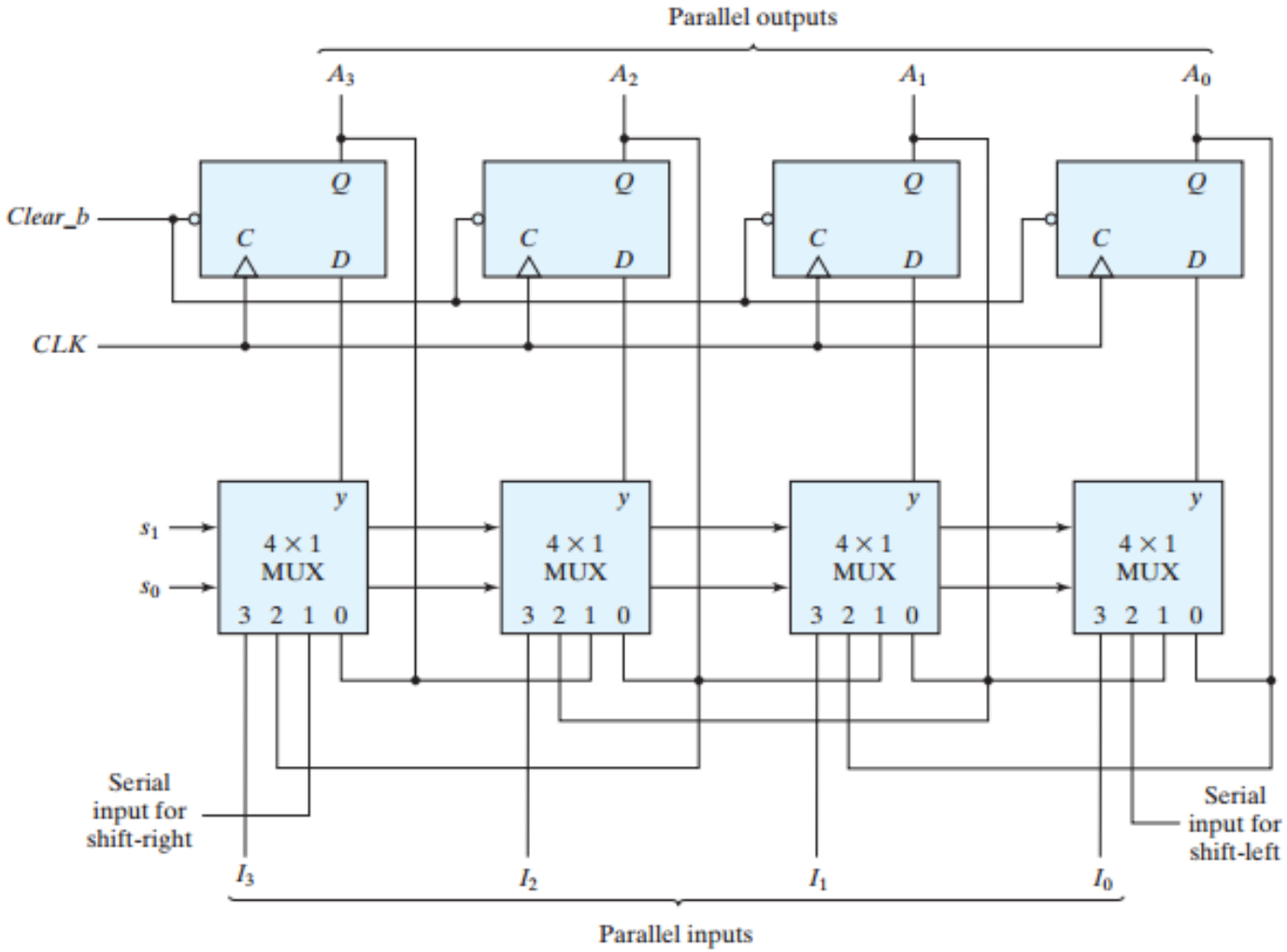


4 bit register with parallel load \ no change

Load = 1
Parallel load



4 bit Universal Shift Register example 1



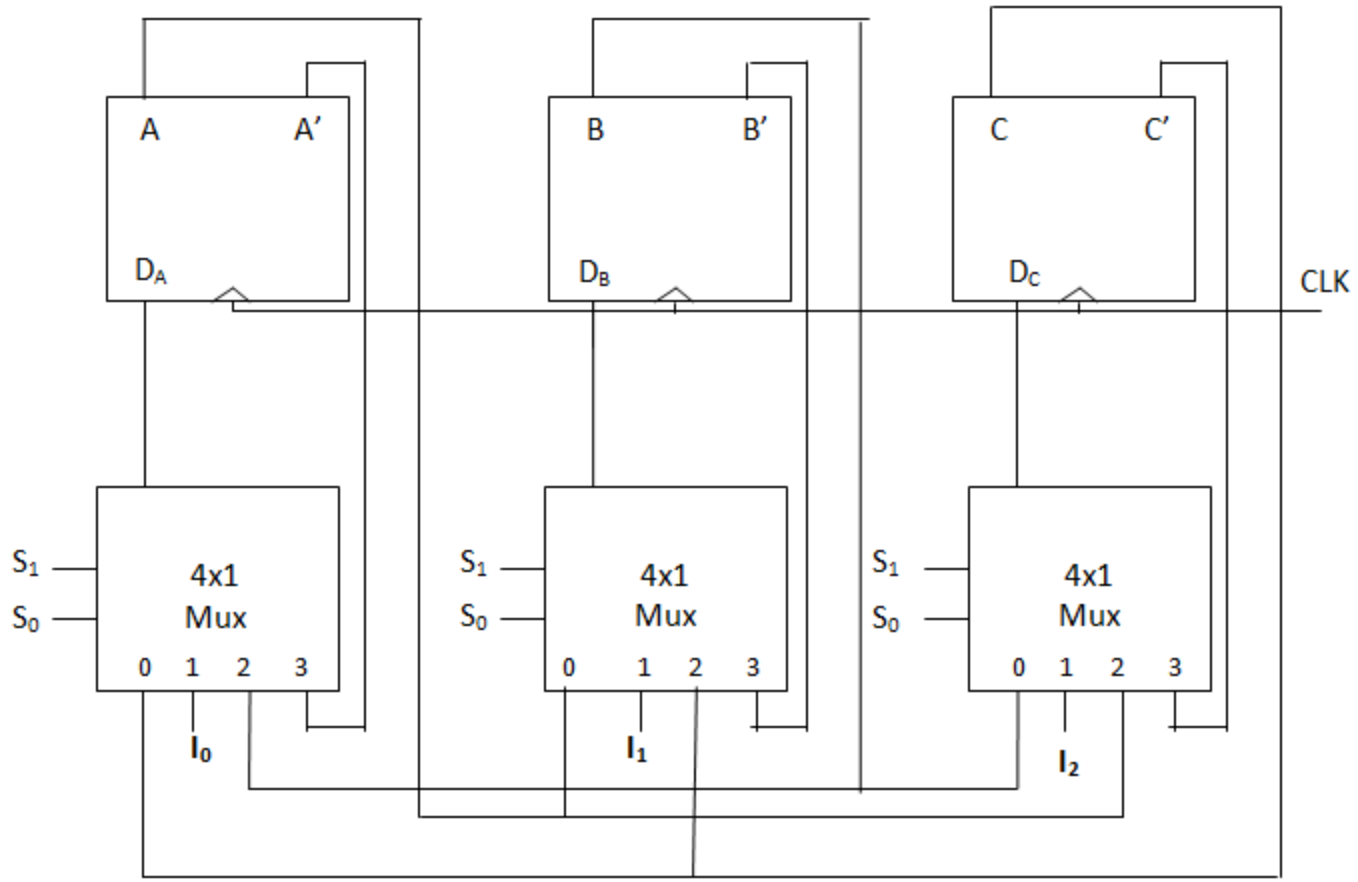
4 bit Universal Shift Register (cont.)

example 1

Function table

Mode Control		
s_1	s_0	**Register Operation**
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

3 bit Universal Shift Register(cont.) example 2



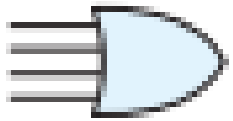
3 bit Universal Shift Register (cont.)
example 2

Function table

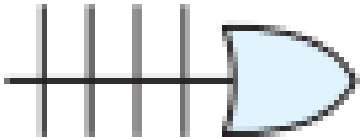
S_0	S_1	Register operation
0	0	Circular shift right
0	1	Parallel Load
1	0	Circular shift left
1	1	complement

Random Access Memory (RAM)

Different representations for OR gate

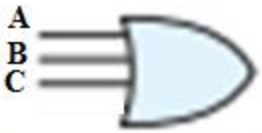


(a) Conventional symbol

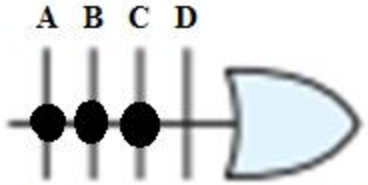


(b) Array logic symbol

Ex: A + B + C

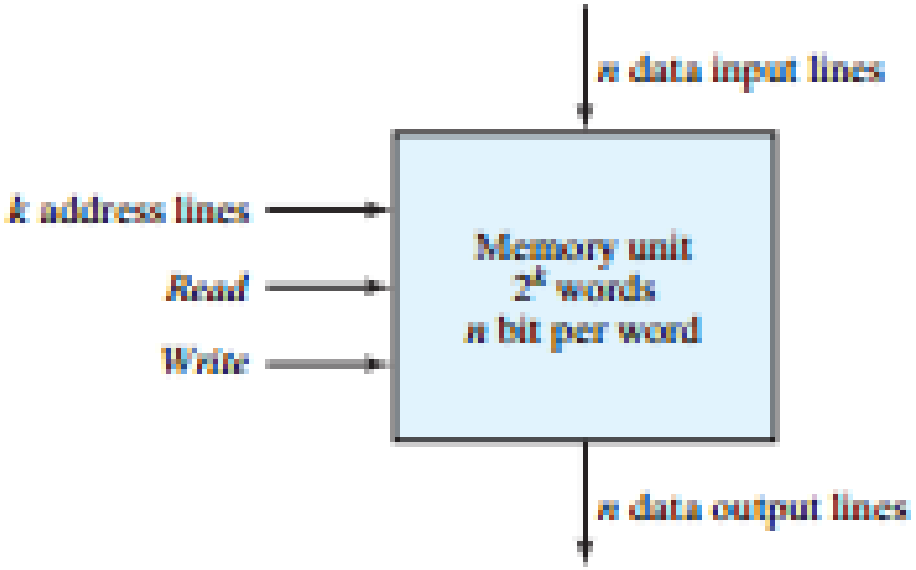


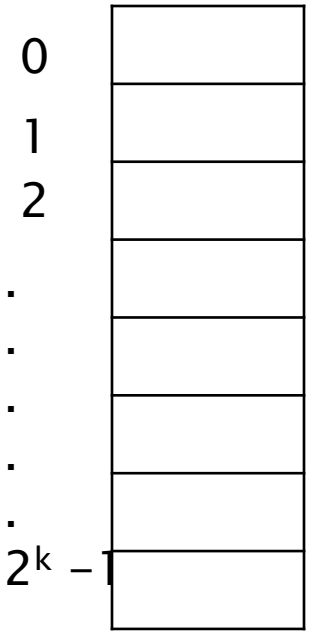
(a) Conventional symbol



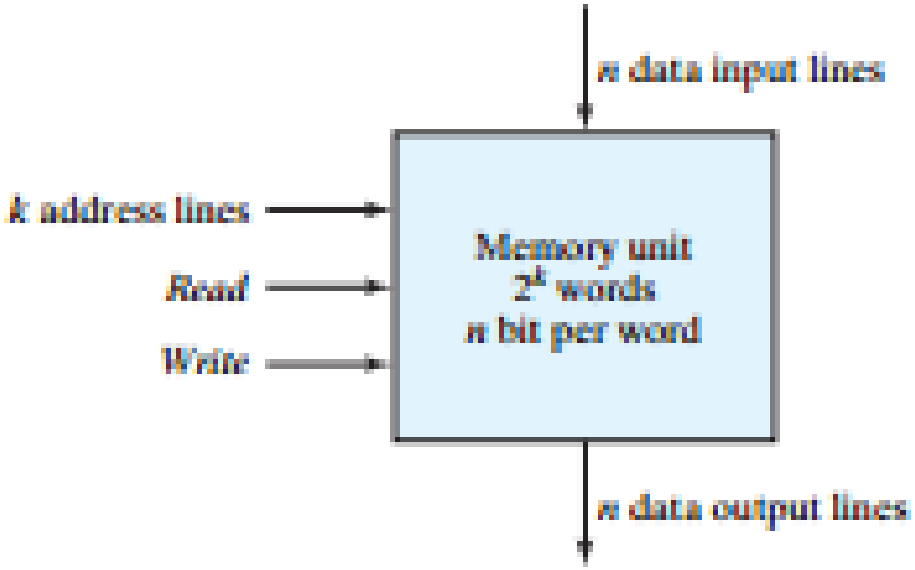
(b) Array logic symbol

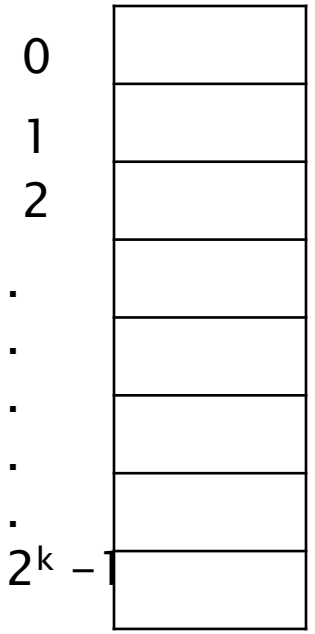
Memory unit (RAM)





Memory unit (RAM)





RAM size = # of words x word size

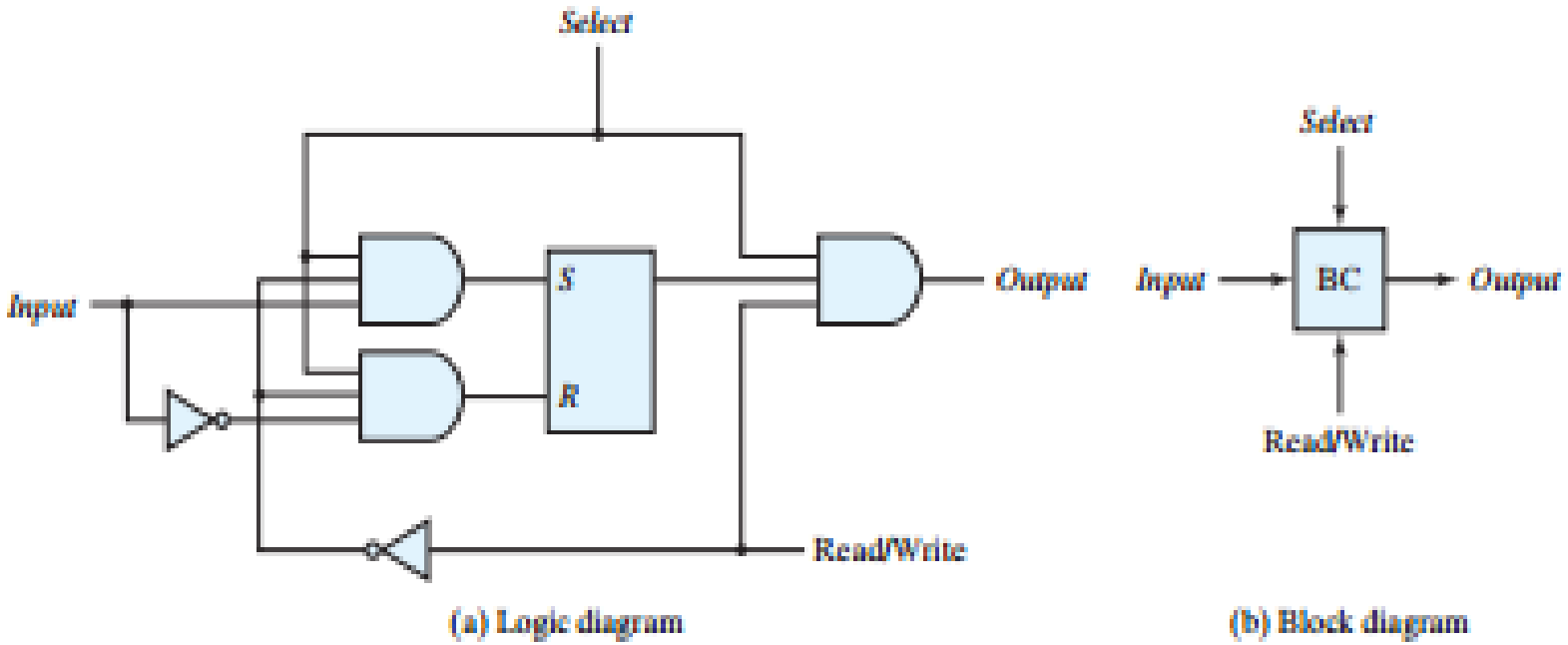
ex: 8 x 4 RAM

number of words = 8 = 2^3

Word size = number of inputs = number of outputs = 4

number of address lines = 3

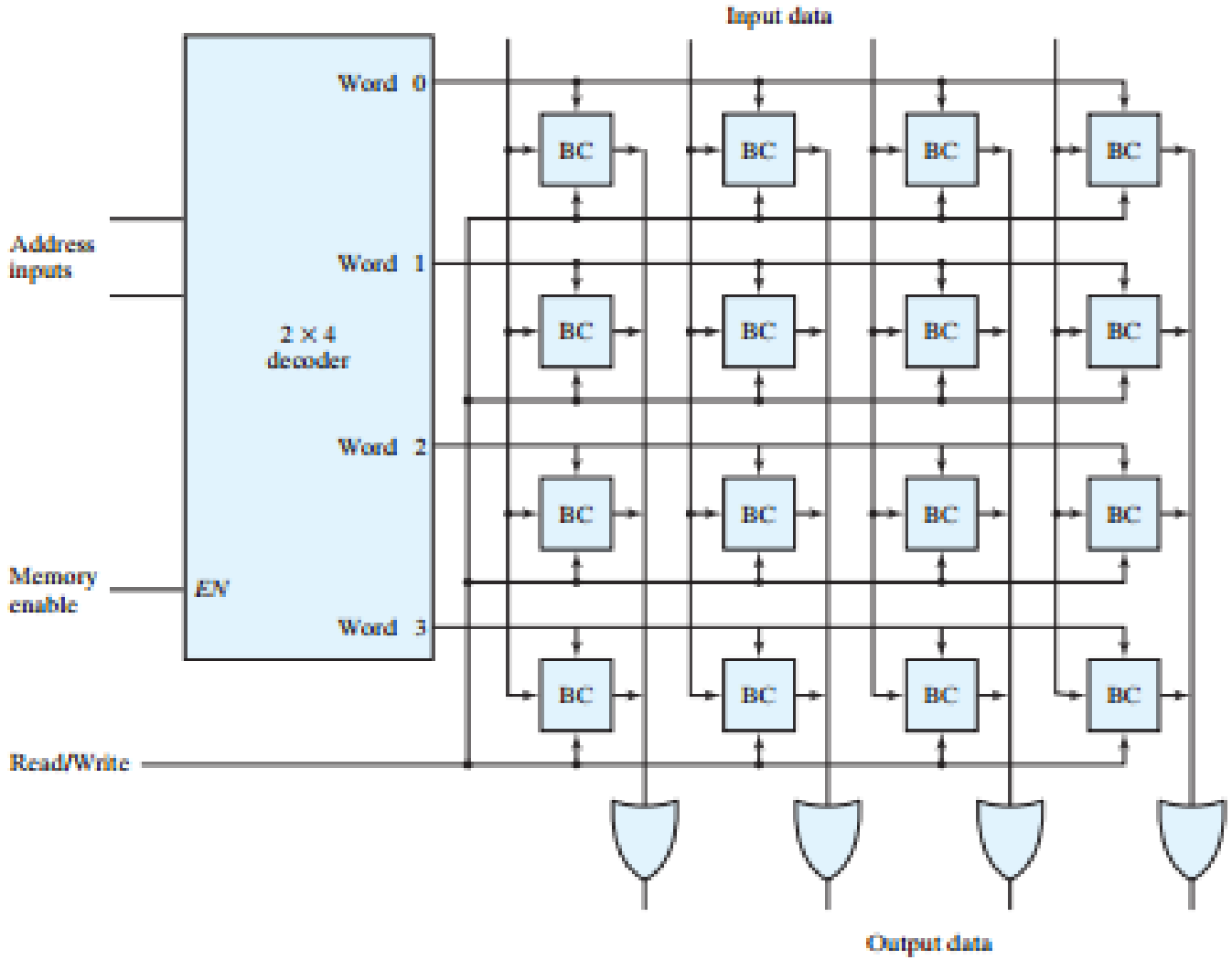
A Binary Cell



(a) Logic diagram

(b) Block diagram

Diagram of a 4 x 4 RAM



If we have 32M x 8 RAM then

- Kilobyte = 2^{10} Mega = 2^{20} Gaga = 2^{30}

Number of words = $2^5 * 2^{20} = 2^{25}$

Word size = 8 bits

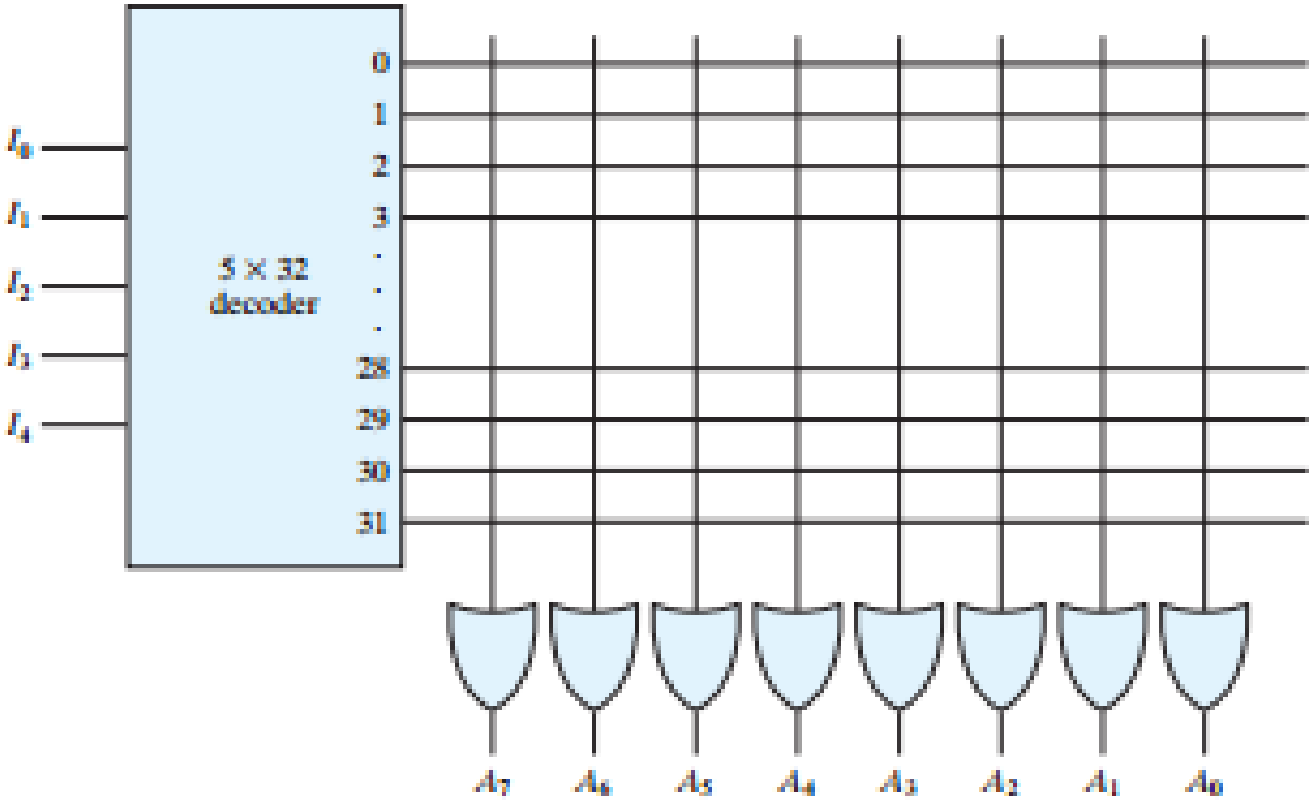
Number of inputs = 8
Number of outputs = 8

Number of address lines = 25

Decoder size = 25 x 2^{25}

Read-only Memory (ROM)

32 x 8 ROM



Example

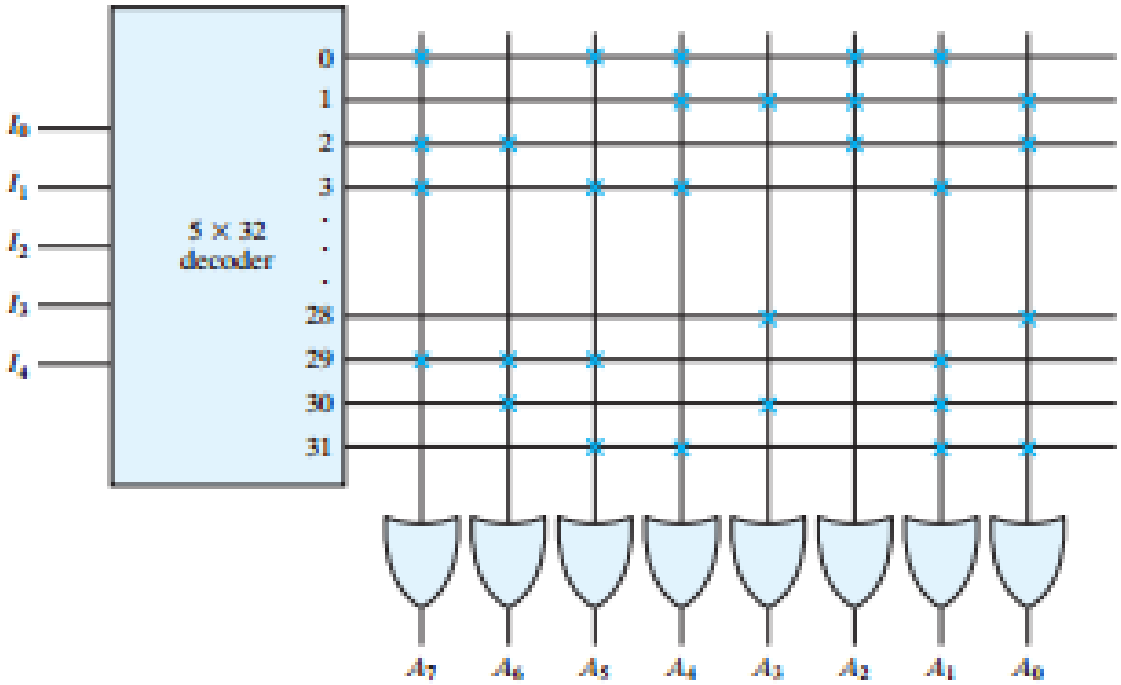
ROM Truth Table (Partial)

Inputs					Outputs							
I_4	I_3	I_2	I_1	I_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
		⋮							⋮			
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1

Programmed ROM

ROM Truth Table (Partial)

Inputs					Outputs							
I_4	I_3	I_2	I_1	I_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
		⋮						⋮				
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1



Thank you