Modern Digital System Design

ECE 2372 / Fall 2018 / Lecture 09

ADD, SUB, Adders, Subtractors & Comparator, ROM, PLA

Texas Tech University
Dr. Tooraj Nikoubin

Add operation for two bits

	0	+		1	+		
	0			0			
0	0		0	1			
C	S		C	S			

	0	+		1	+		
	1			1			
0	1		1	0			
C	S		C	S			

Add operation for two bits

	0	+
	0	
	0	
0	0	
C	S	

Add operation for two bits

	0	+
	0	
	0	
0	0	
C	S	

	1	+
	0	
	0	
0	1	
C	S	

Add operation for two bits

0	1	0
+	+	+
0	0	1
0	0	0
0 0	0 1	0 1
C S	C S	C S

Add operation for two bits

	0	+		1	+		0	+		0	+
	0			0			1			0	
	0			0			0			1	
0	0		0	1		0	1		0	1	
C	S		C	S		C	S		C	S	

Add operation for three bits

	1	+		0	+		1	+		1	+
	1			1			0			1	
	0			1			1			1	
1	0		1	0		1	0		1	1	
C	S		C	S		C	S		C	S	

SUB operation for two bits

	0	-		1	-	1	0	-		1	-
	0			0			1			1	
0	0		0	1		1	1		0	0	
B	D		B	D		B	D		B	D	

SUB operation for three bits

	0	-		1	-	1	0	-		1	-
	0			0			0			0	
	0			0			1			1	
0	0		0	1		1	1		0	0	
B	D		B	D		B	D		B	D	

SUB operation for three bits

0	-
1	
1	
B	D

SUB operation for three bits

1	0	-
	1	
	1	
B	D	

SUB operation for three bits

1	0	-
	1	
	1	
1	0	
B	D	

	1	-
	1	
	1	
B	D	

SUB operation for three bits

1	0	-
	1	
	1	
1	0	
B	D	

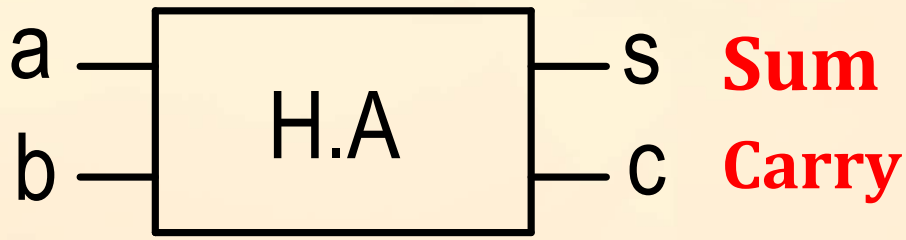
1	1	-
	1	
	1	
B	D	

SUB operation for three bits

1	0	-
	1	
	1	
1	0	
B	D	

1	1	-
	1	
	1	
1	1	
B	D	

Half Adder



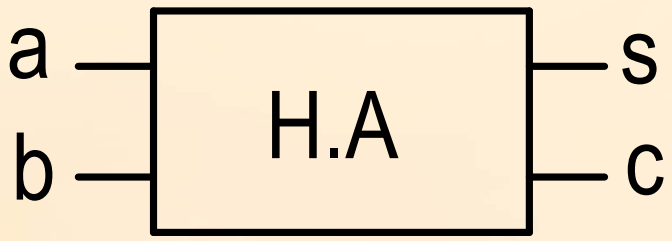
Half Adder

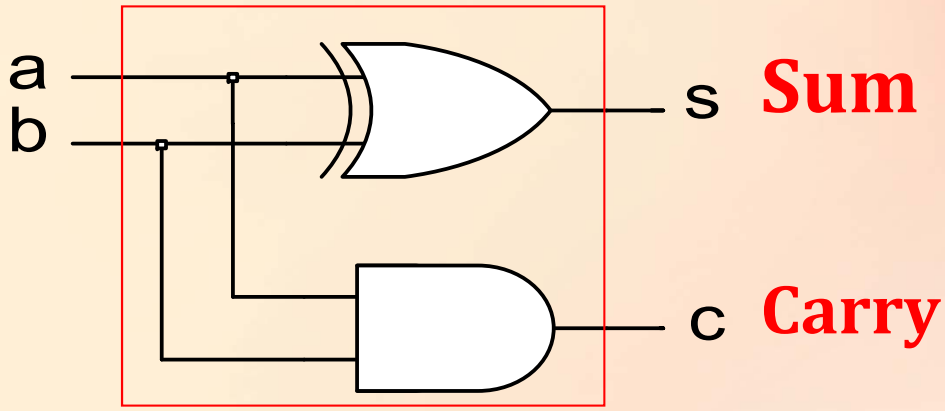
b	a	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Half Adder

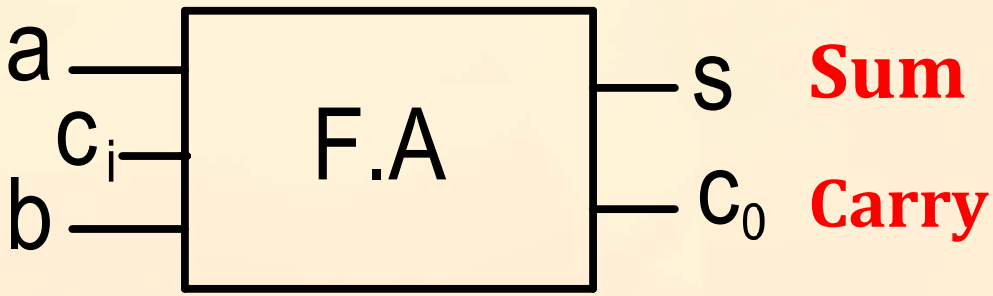
b	a	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = a \oplus b$$
$$C = ab$$





Half Adder



Full Adder

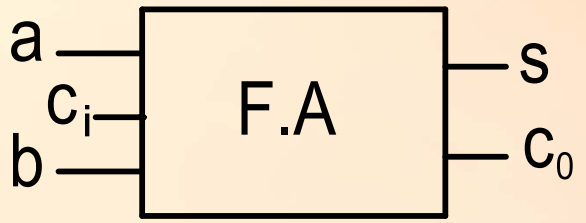
Ci	b	a	C0	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Sum

Carry

Full Adder

Ci	b	a	C0	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

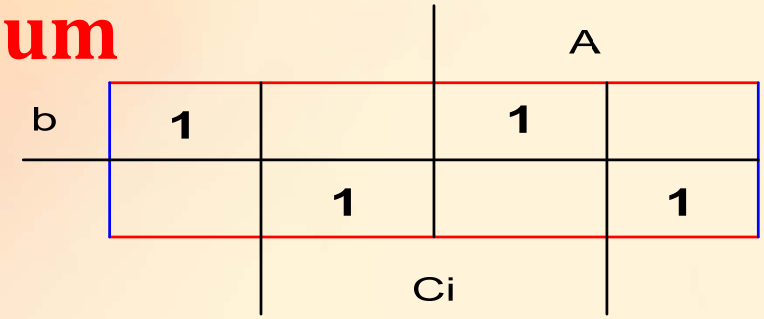


$$S = \sum m(1,2,4,7)$$

$$Co = \sum m(3,5,6,7)$$

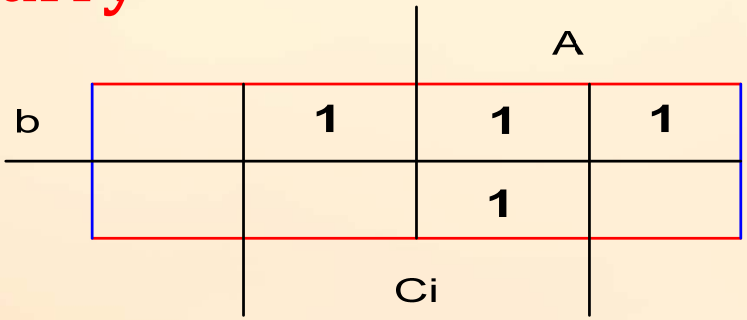
Full Adder

Sum

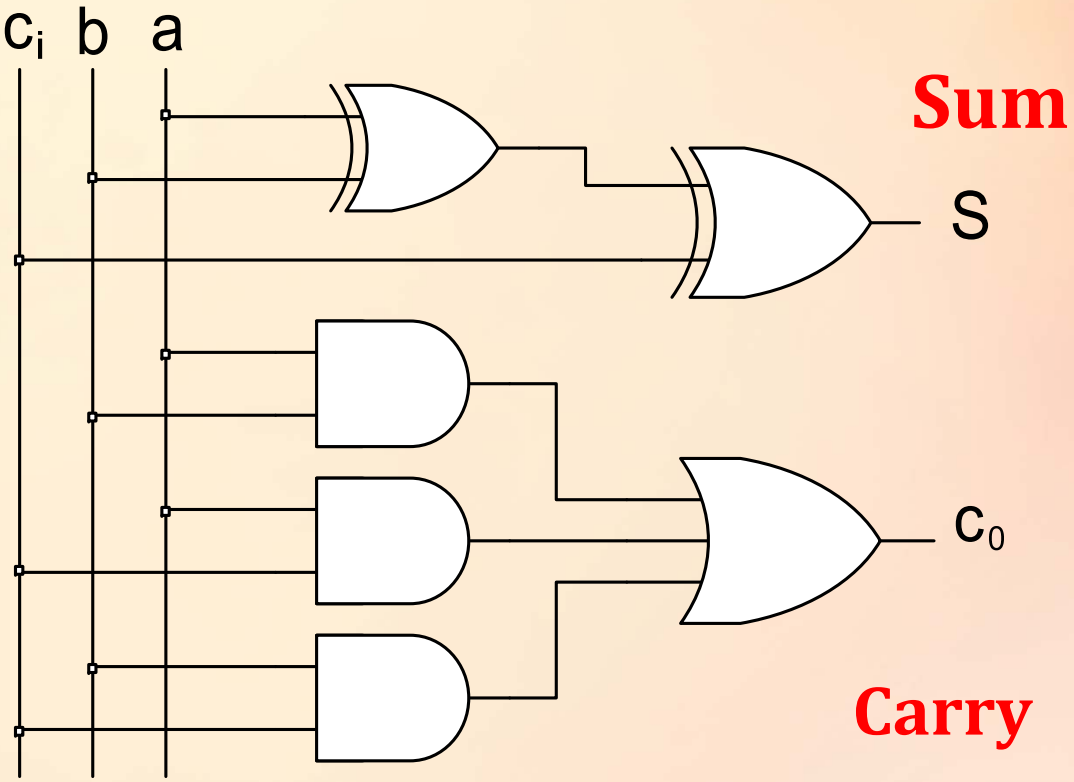


$$S = a \oplus b \oplus C_i$$

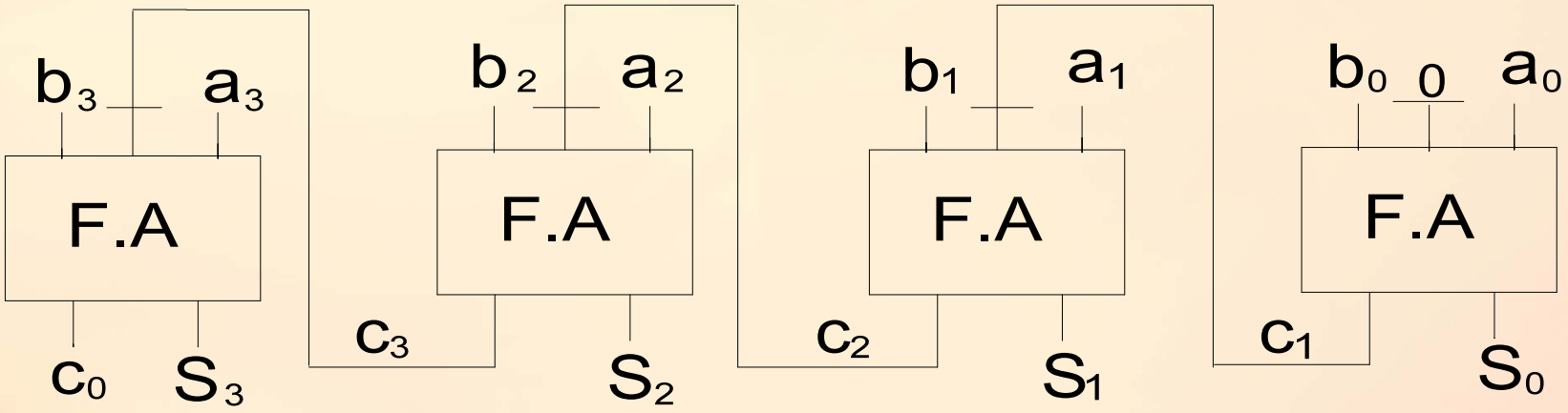
Carry



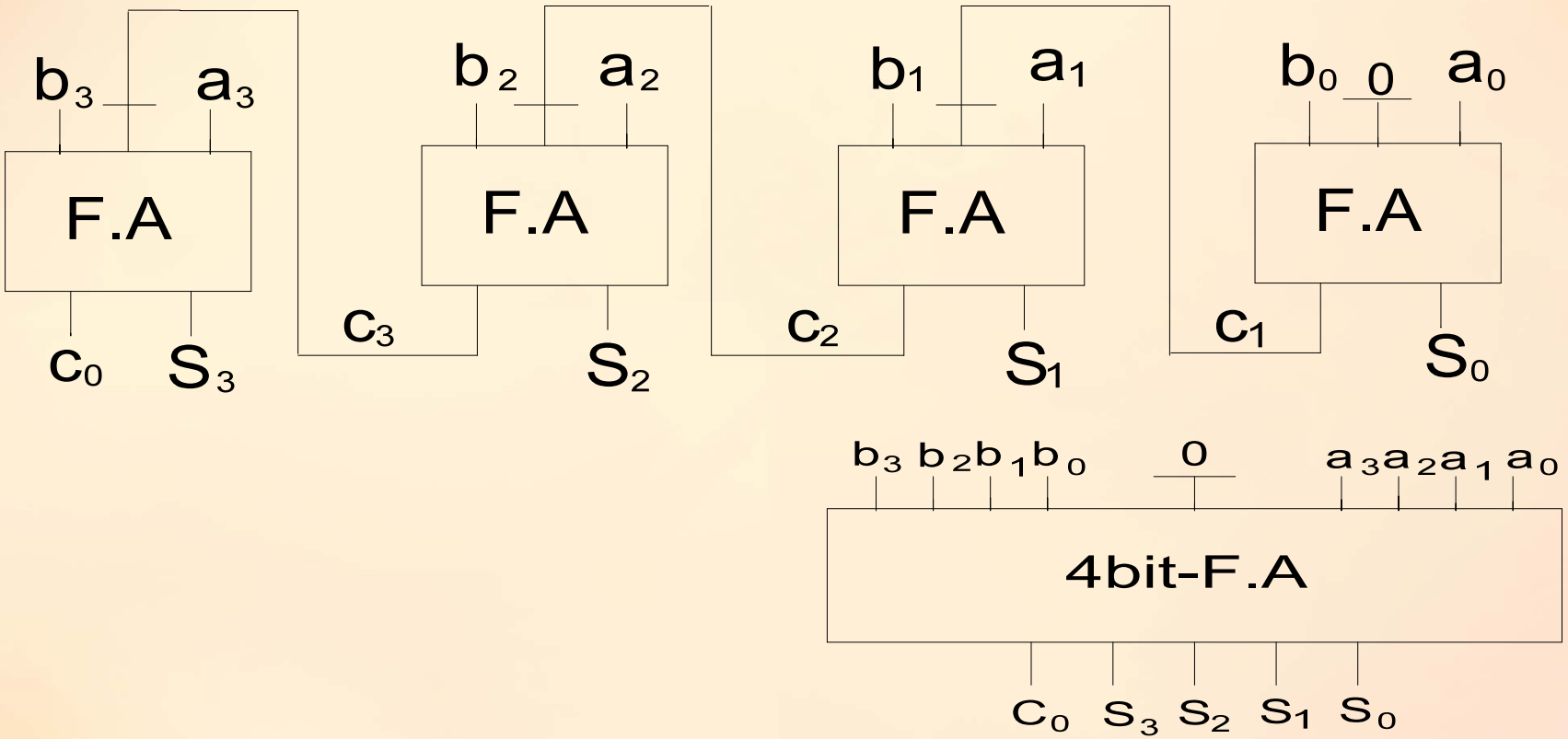
$$Co = ab + aC_i + bC_i$$



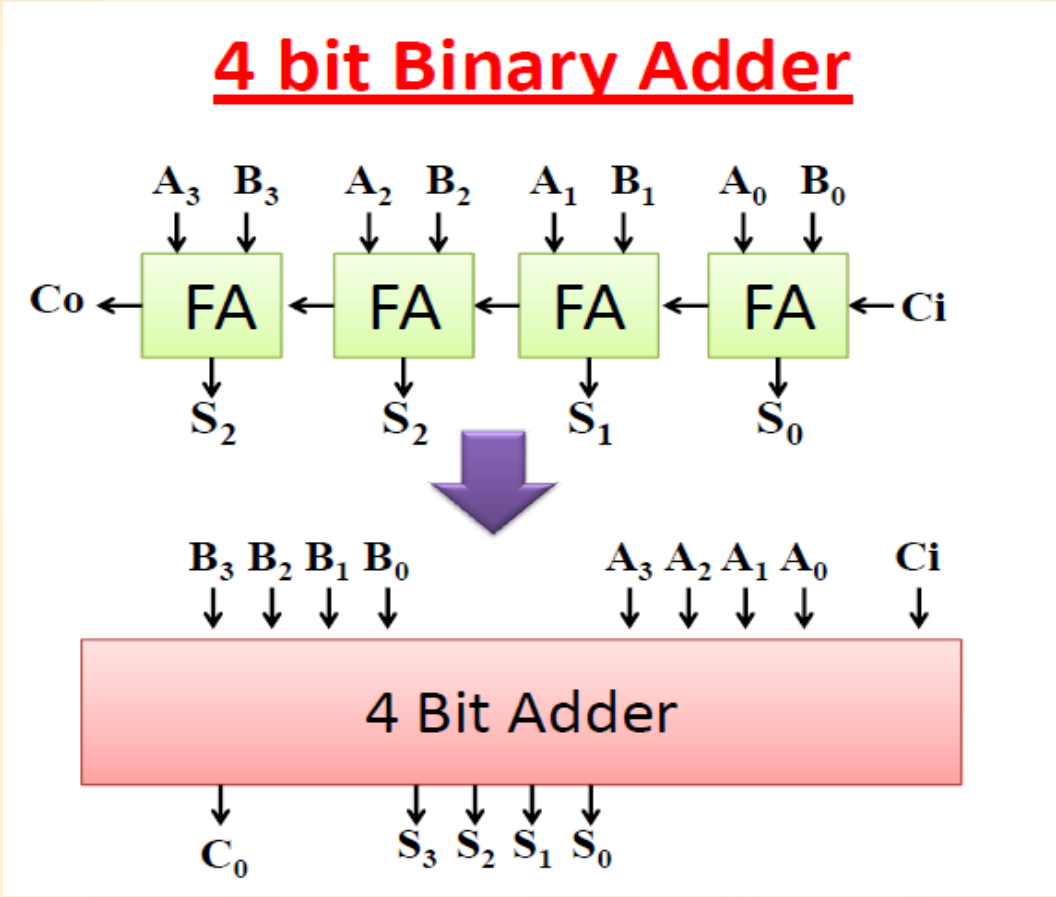
4 Bit Binary Adder



4 Bit Binary Adder



4 Bit Binary Adder



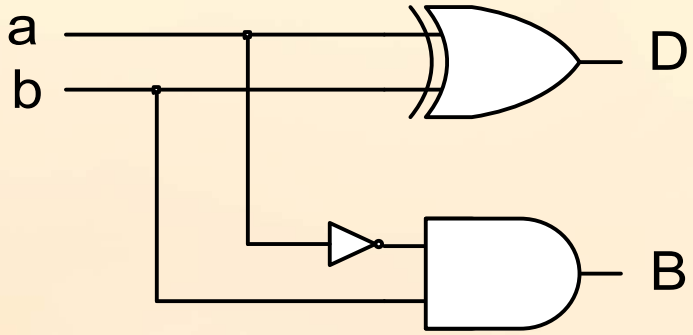
Half Subtractor

Differential

Borrow

b	a	B	D
0	0	0	0
0	1	0	1
1	0	1	1
1	1	0	0

Half Subtractor

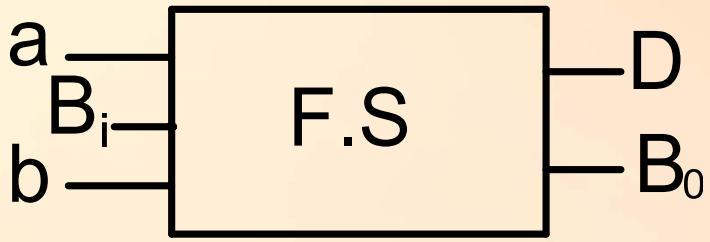


b	a	B	D
0	0	0	0
0	1	0	1
1	0	1	1
1	1	0	0

$$D = a \oplus b$$
$$B = \overline{a}\, b$$

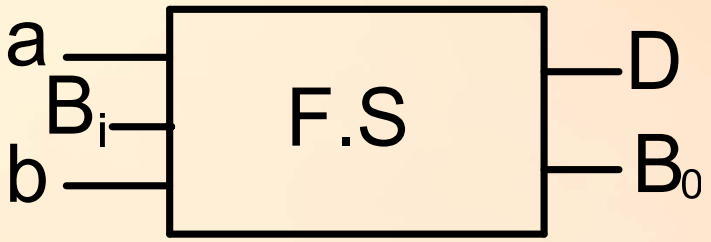
Full Subtractor

B_i	b	a	B_0	D
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1



Full Subtractor

B_i	b	a	B_0	D
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1



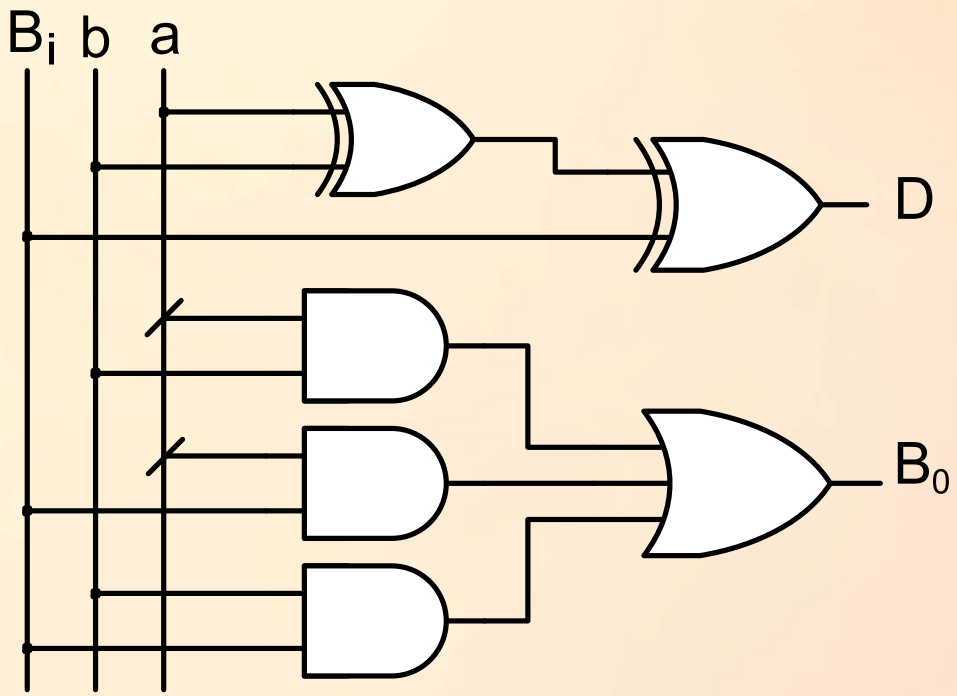
$$D = a \oplus b \oplus B_i$$

$$Bo = \bar{a}b + \bar{a}B_i + bB_i$$

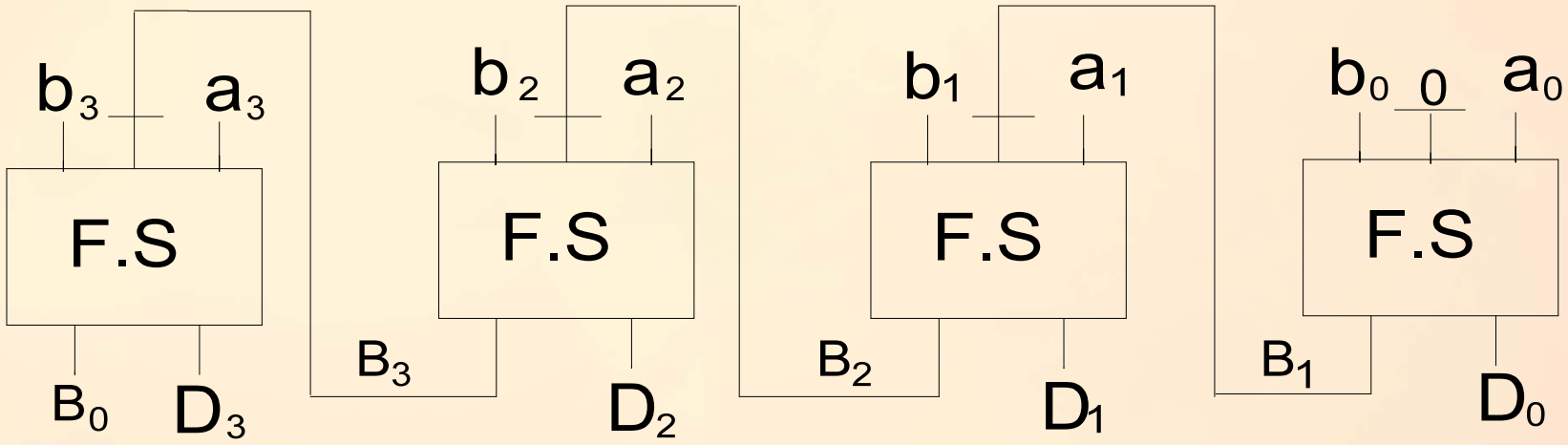
Full Subtractor

$$D = a \oplus b \oplus B_i$$

$$B_o = \bar{a}b + \bar{a}B_i + bB_i$$

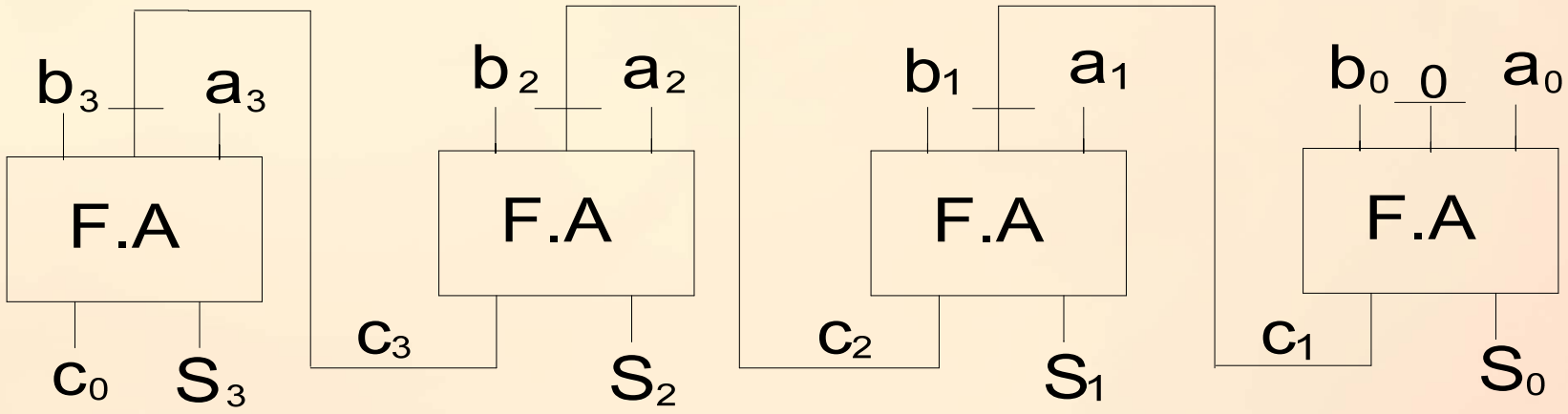


Example for 4 Bit Adder



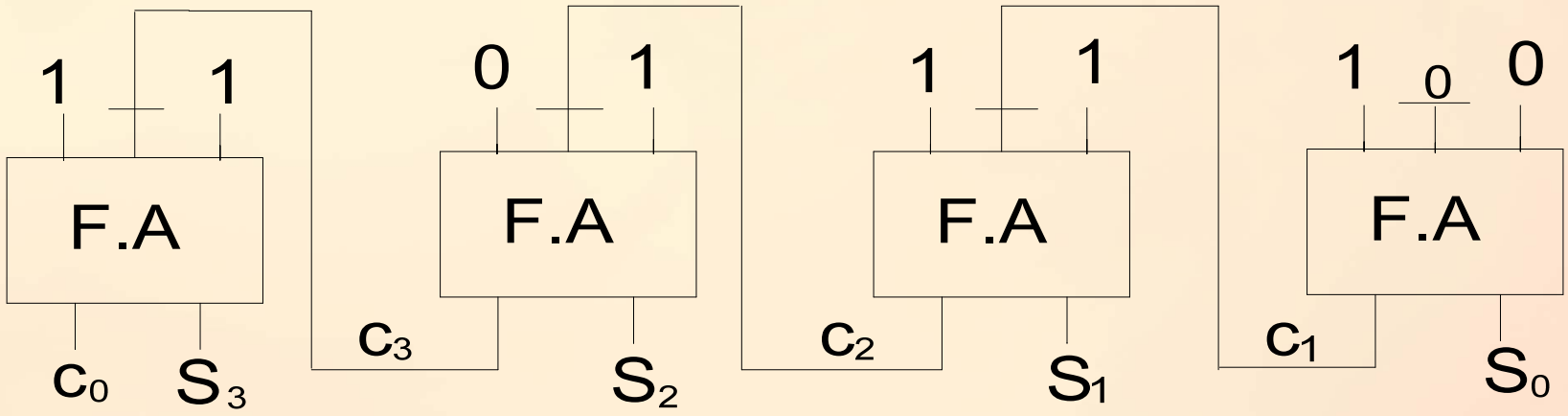
Example for 4 Bit Adder

$a = 1\ 1\ 1\ 0 \qquad b = 1\ 0\ 1\ 1$



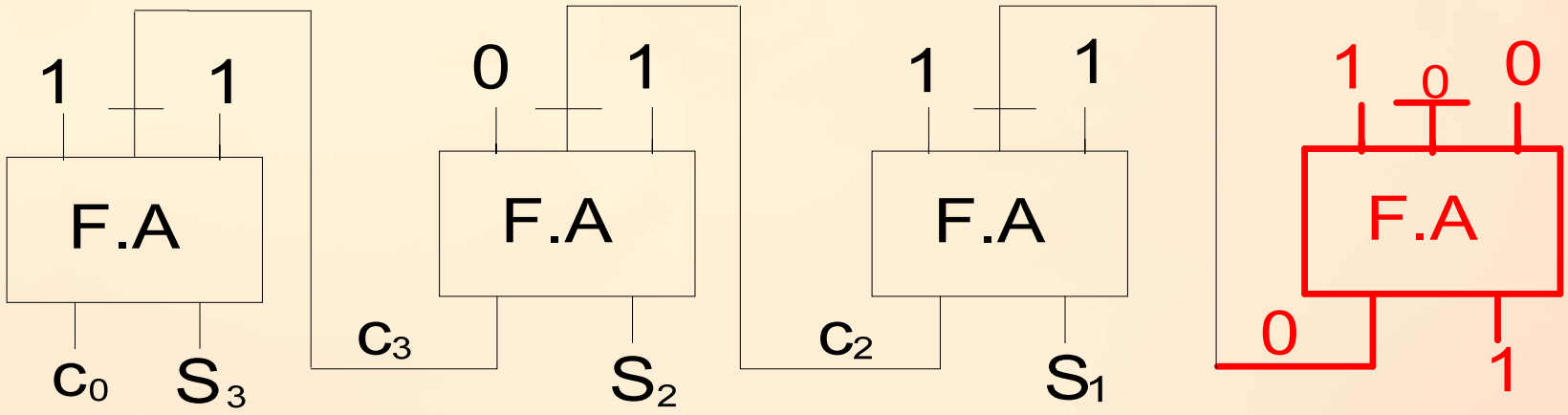
Example for 4 Bit Adder

$a = 1\ 1\ 1\ 0 \qquad b = 1\ 0\ 1\ 1$



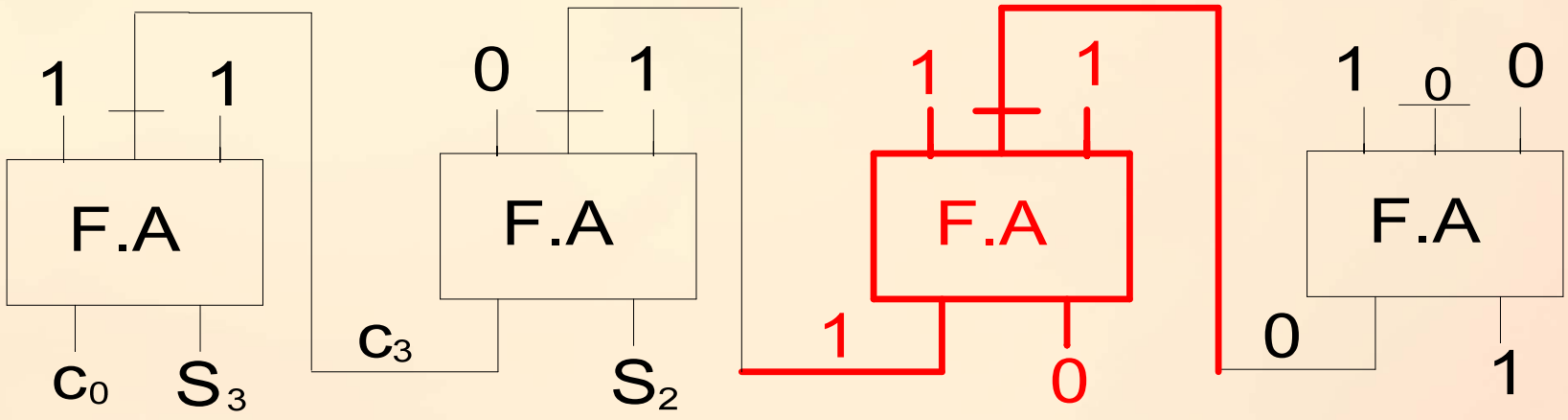
Example for 4 Bit Adder

$a = 1\ 1\ 1\ 0 \qquad b = 1\ 0\ 1\ 1$



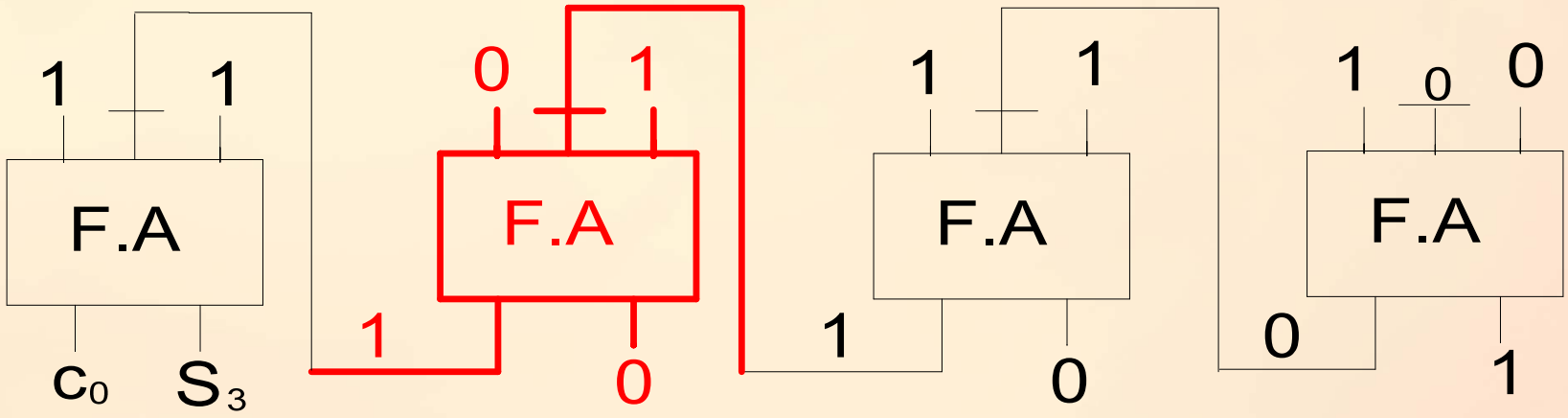
Example for 4 Bit Adder

$a = 1\ 1\ 1\ 0 \qquad b = 1\ 0\ 1\ 1$



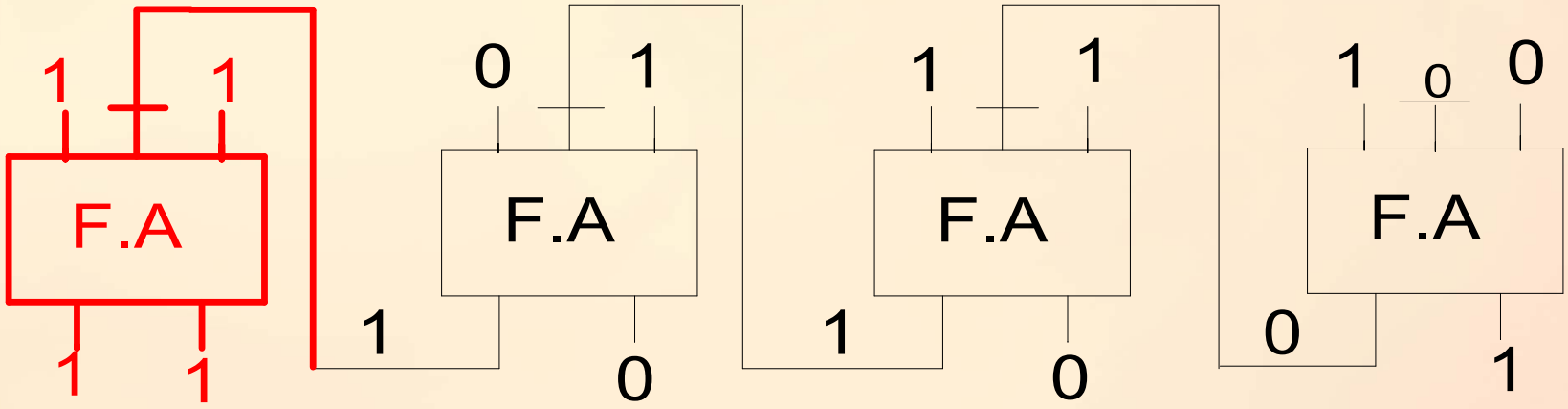
Example for 4 Bit Adder

$a = 1\ 1\ 1\ 0 \qquad b = 1\ 0\ 1\ 1$

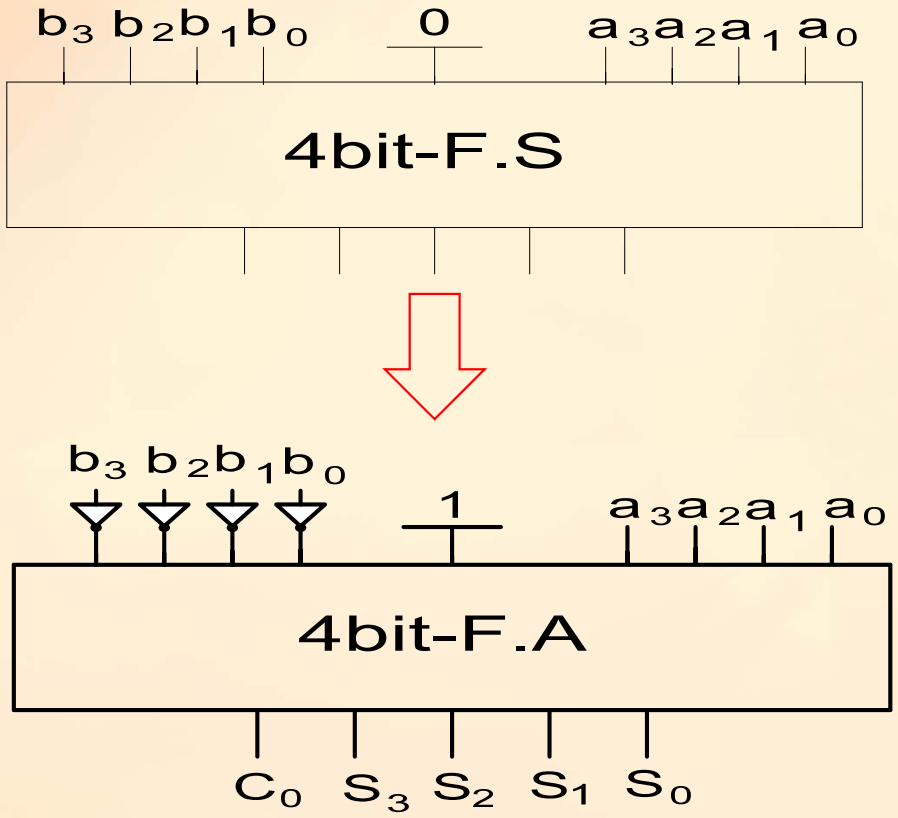


Example for 4 Bit Adder

$a = 1\ 1\ 1\ 0 \qquad b = 1\ 0\ 1\ 1$

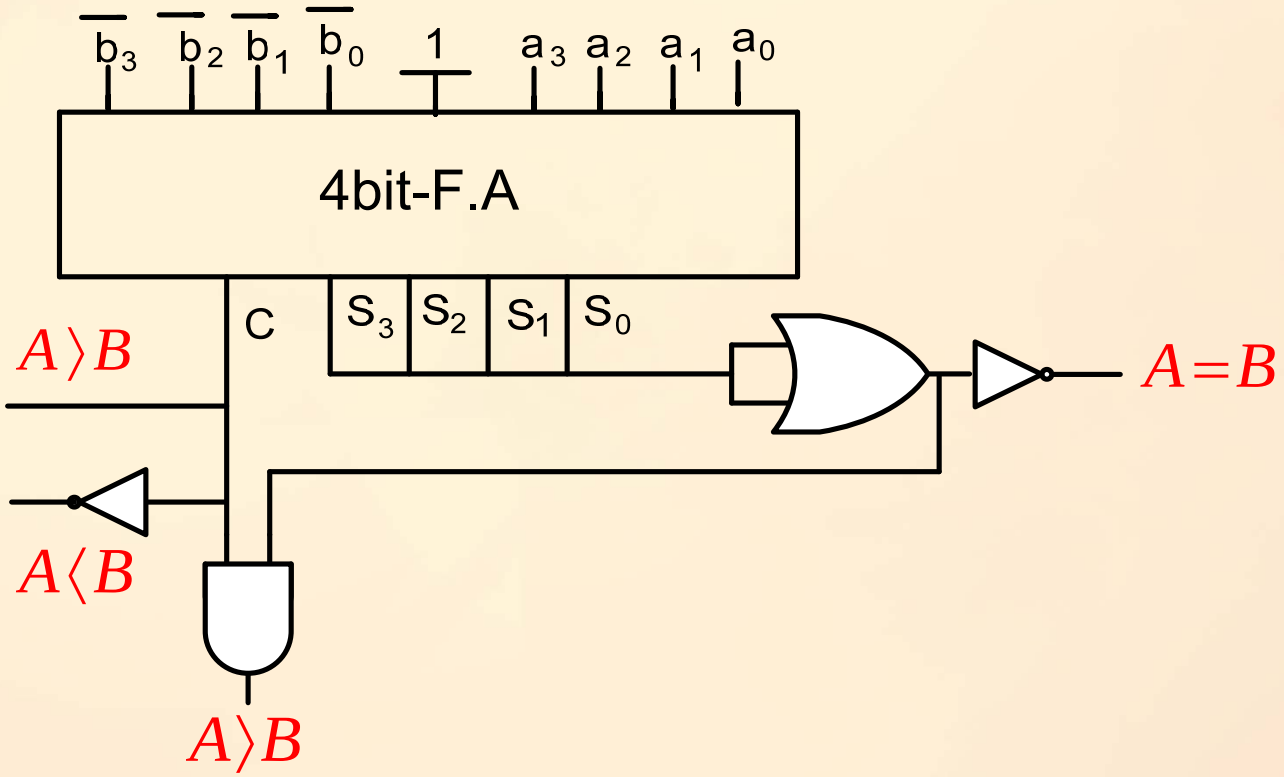


Example for 4 Bit Adder

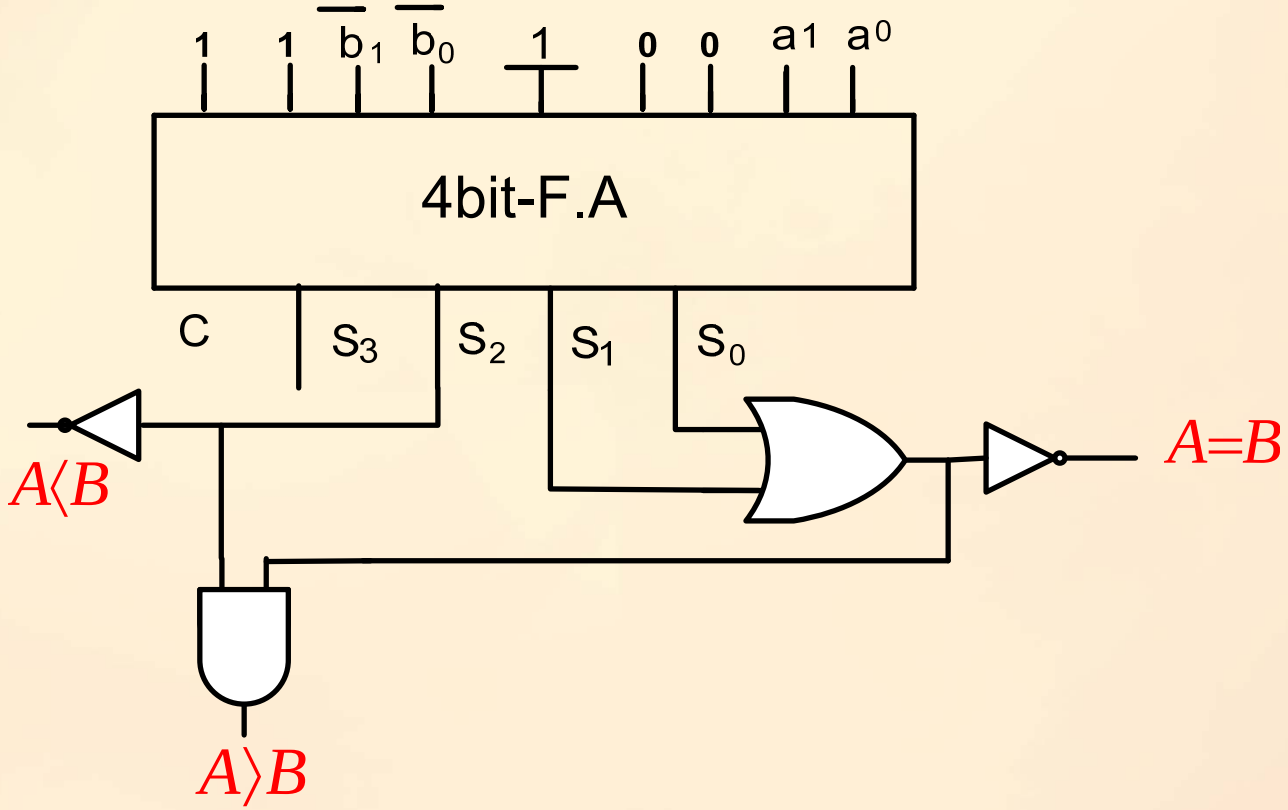


$$A - B = A + \overline{B} + 1$$

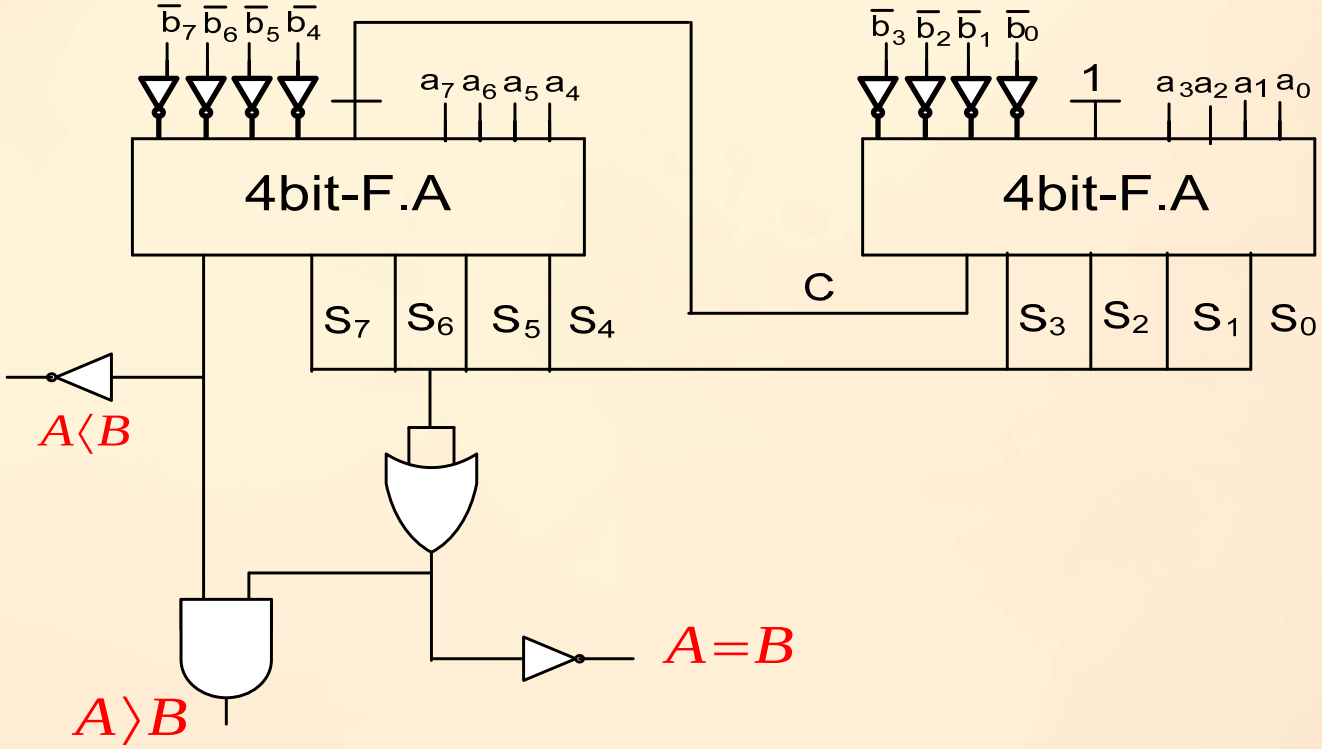
4 Bit Adder Comparator

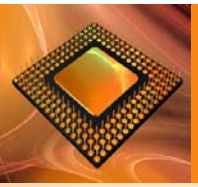


4 Bit Adder Comparator



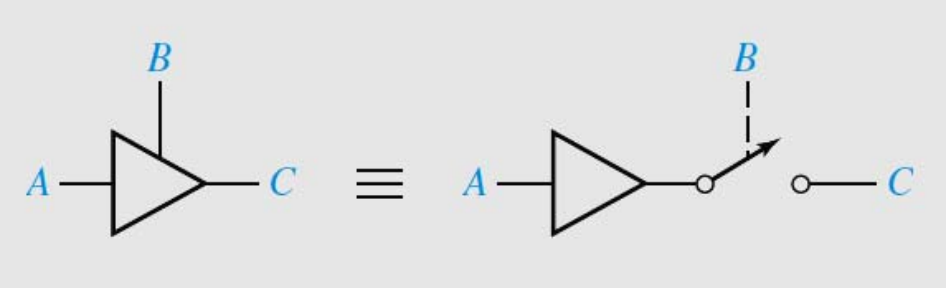
8 Bit Adder Comparator

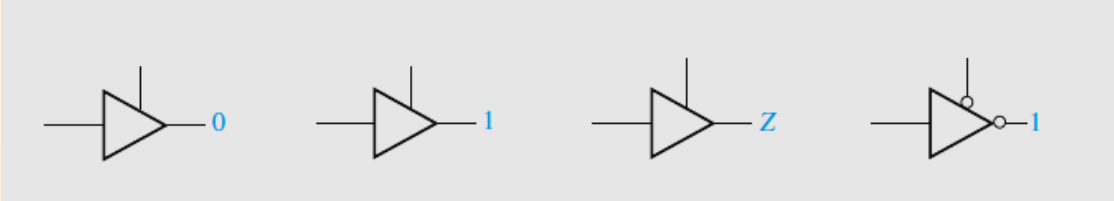


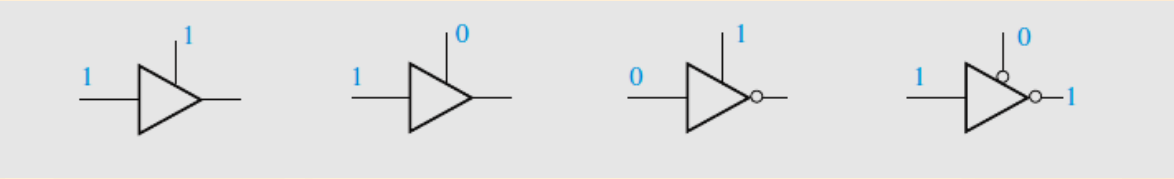


Add/Sub Module Design

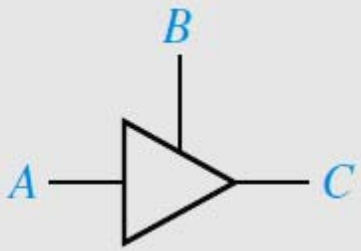
Three State Buffer





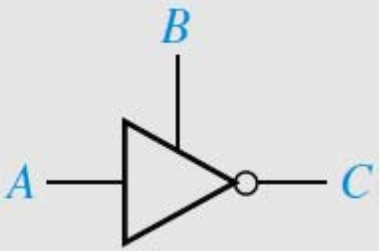


Three State Buffer



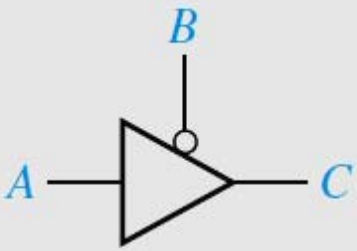
B	A	C
0	0	Z
0	1	Z
1	0	0
1	1	1

(a)



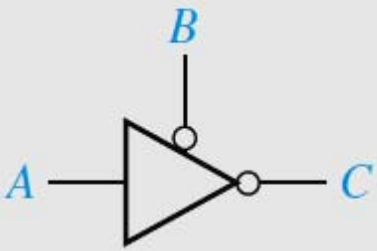
B	A	C
0	0	Z
0	1	Z
1	0	1
1	1	0

(b)



B	A	C
0	0	0
0	1	1
1	0	Z
1	1	Z

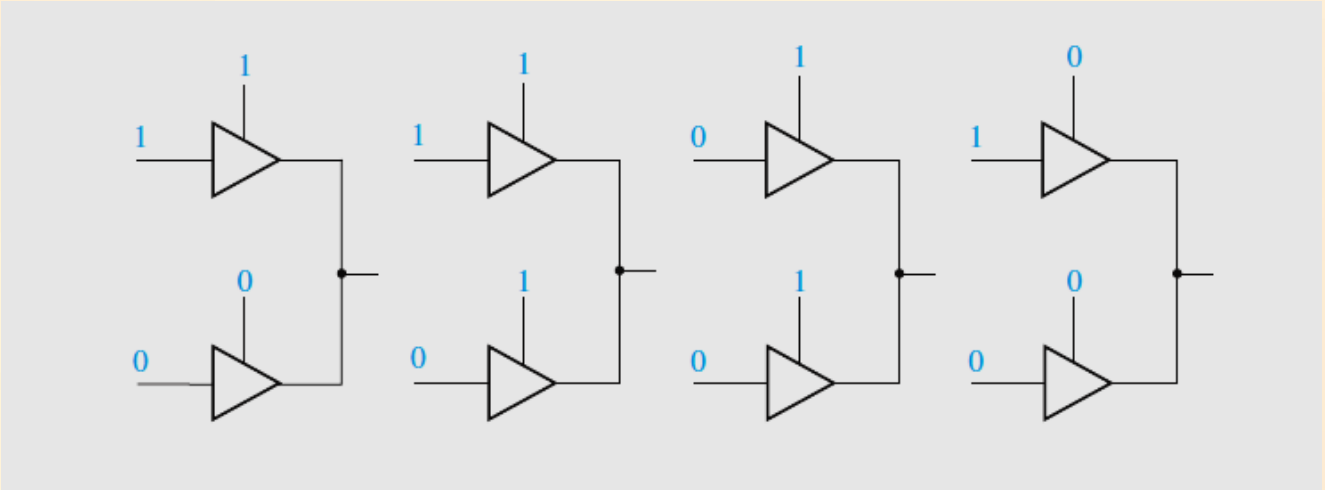
(c)



B	A	C
0	0	1
0	1	0
1	0	Z
1	1	Z

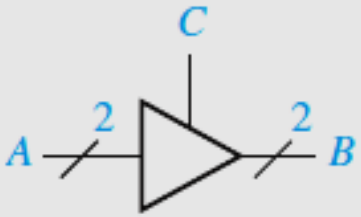
(d)

Three State Buffer

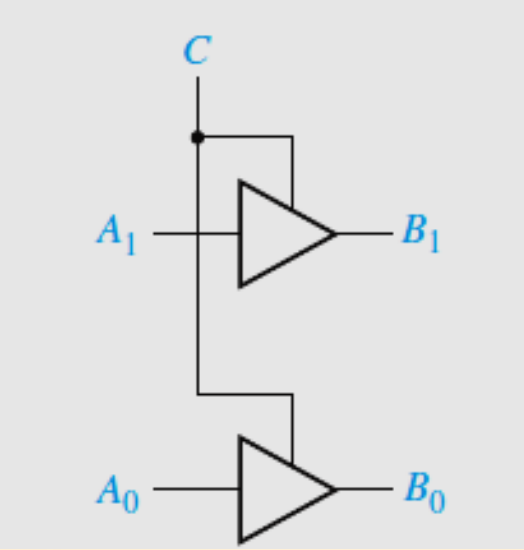


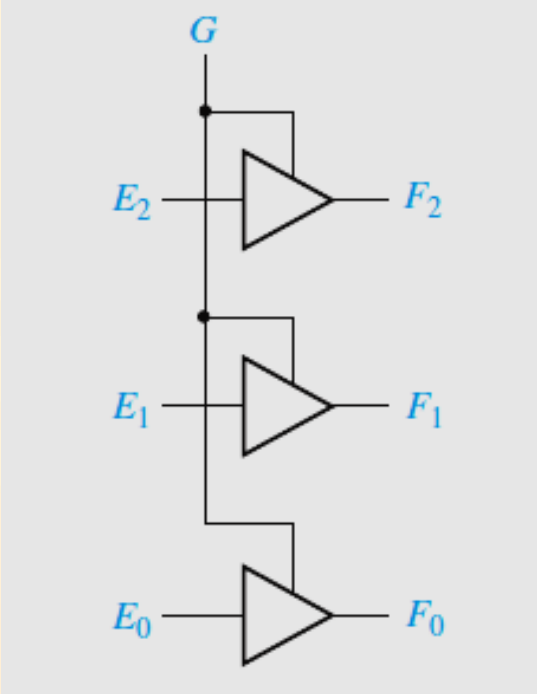
The bus symbol A 

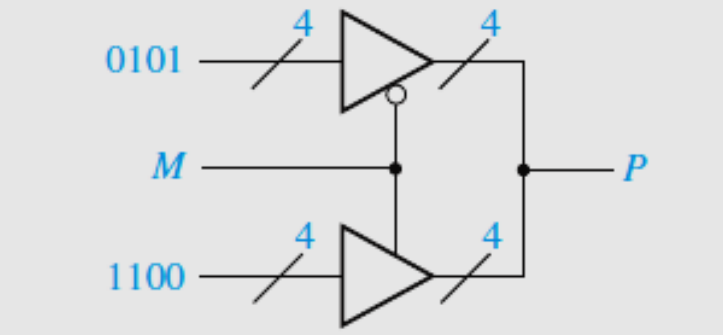
control input:



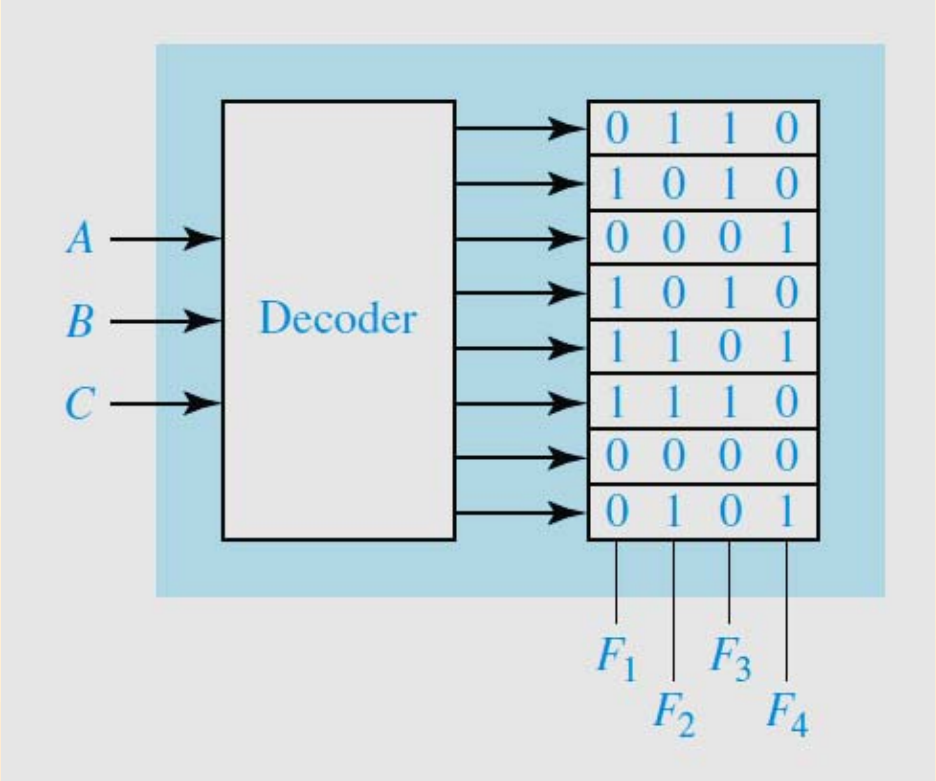
Using bus notation, draw an equivalent circuit for:



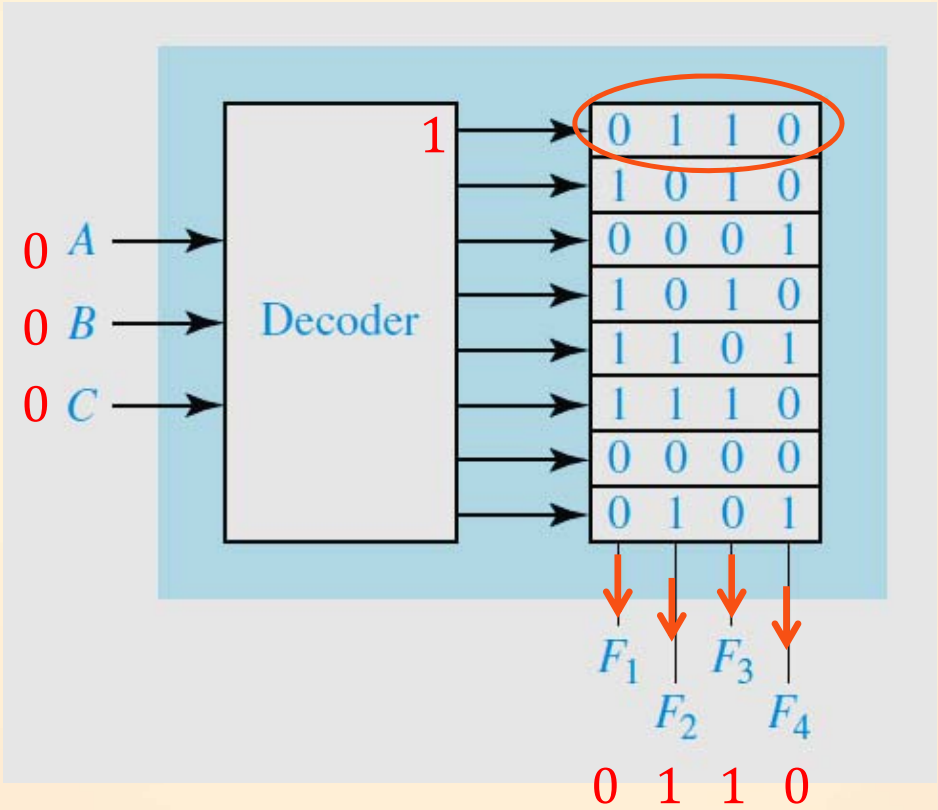




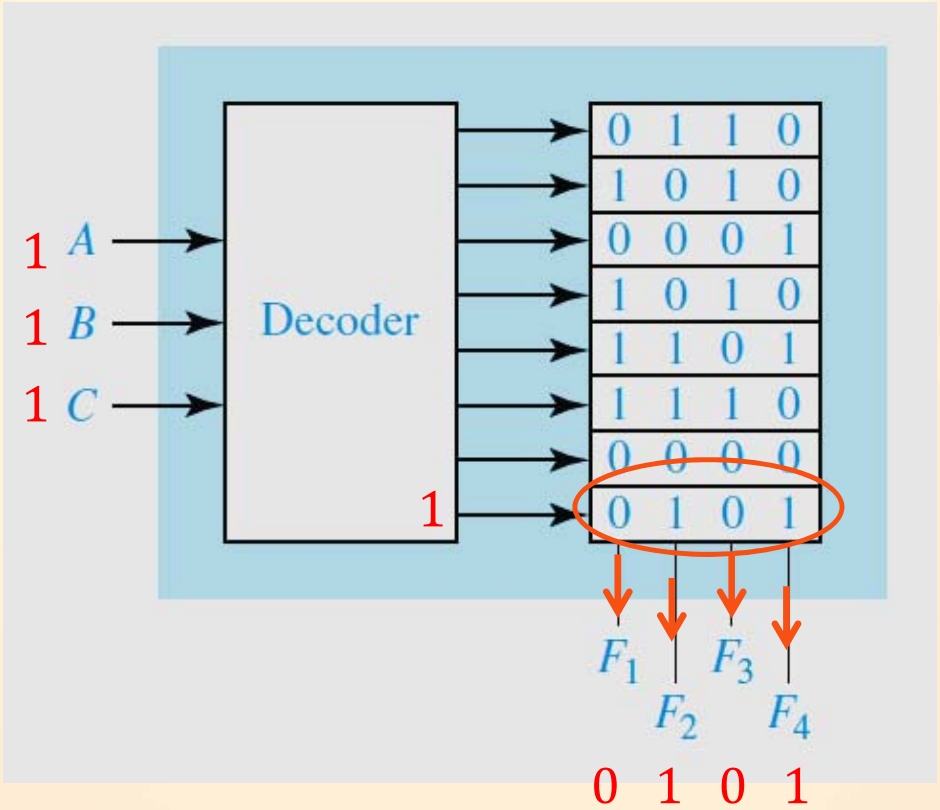
Read Only Memory (ROM)



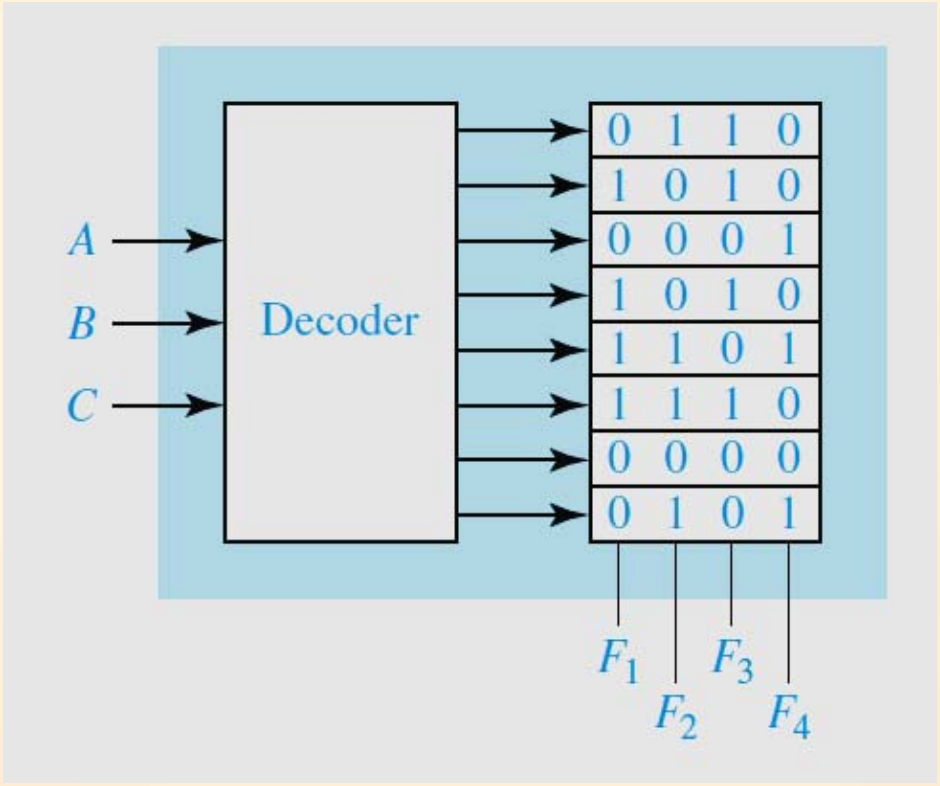
Read Only Memory (ROM)



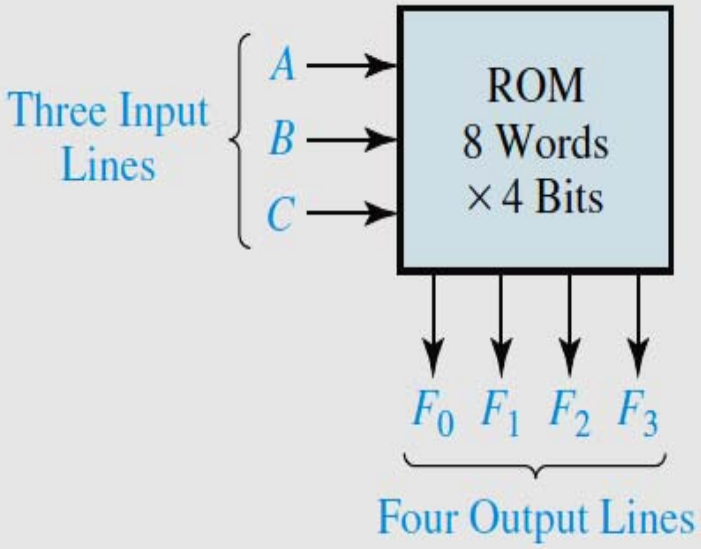
Read Only Memory (ROM)



Read Only Memory (ROM)



Read Only Memory (ROM)



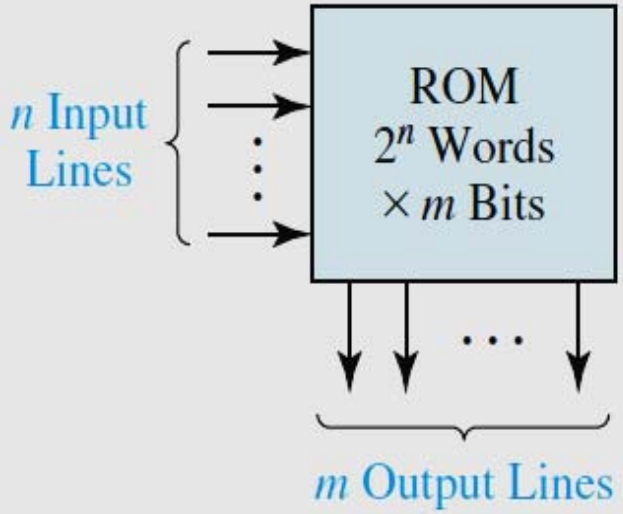
(a) Block diagram

A B C	F_0 F_1 F_2 F_3
0 0 0	1 0 1 0
0 0 1	1 0 1 0
0 1 0	0 1 1 1
0 1 1	0 1 0 1
1 0 0	1 1 0 0
1 0 1	0 0 0 1
1 1 0	1 1 1 1
1 1 1	0 1 0 1

Typical Data Stored in ROM (2^3 words of 4 bits each)

(b) Truth table for ROM

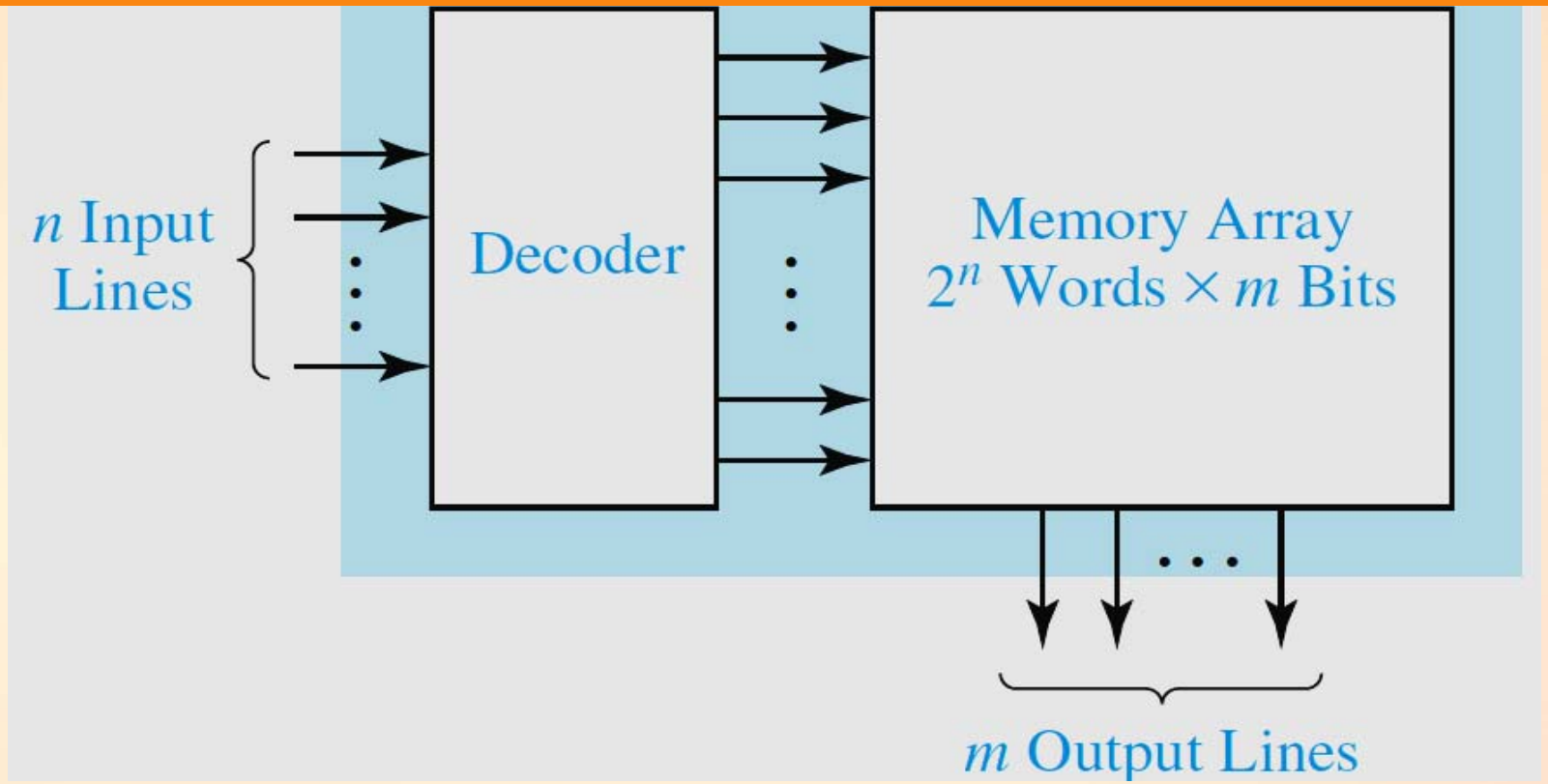
Read Only Memory (ROM)



n Input Variables	m Output Variables
00 · · · 00	100 · · · 110
00 · · · 01	010 · · · 111
00 · · · 10	101 · · · 101
00 · · · 11	110 · · · 010
⋮	⋮
11 · · · 00	001 · · · 011
11 · · · 01	110 · · · 110
11 · · · 10	011 · · · 000
11 · · · 11	111 · · · 101

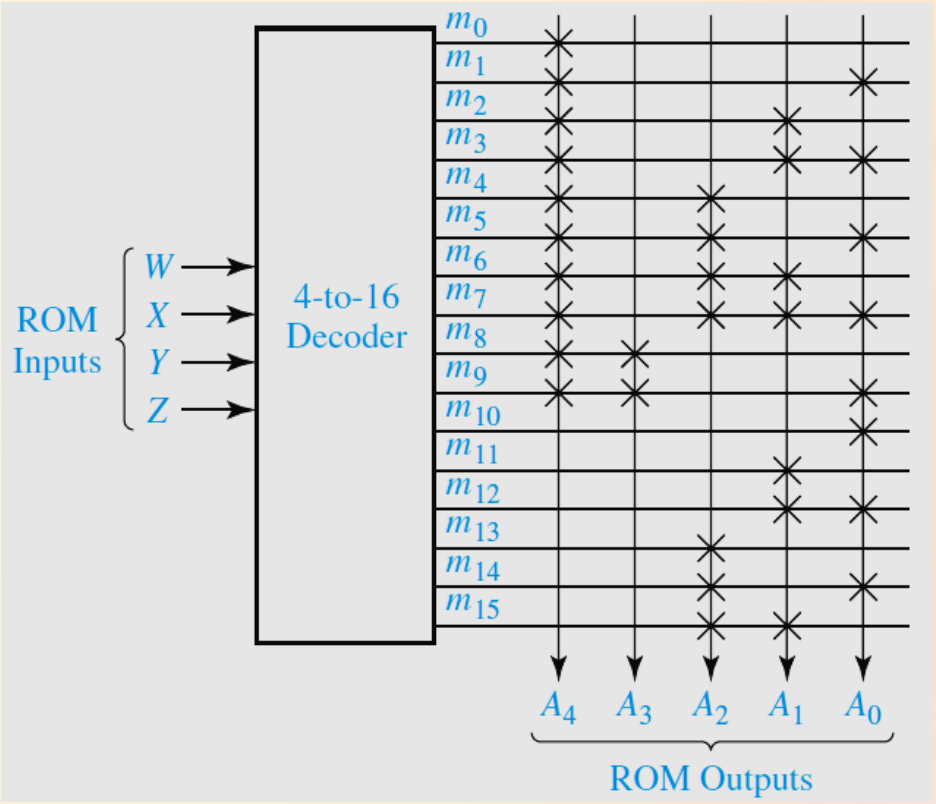
Typical Data Array Stored in ROM (2^n words of m bits each)

Read Only Memory (ROM)

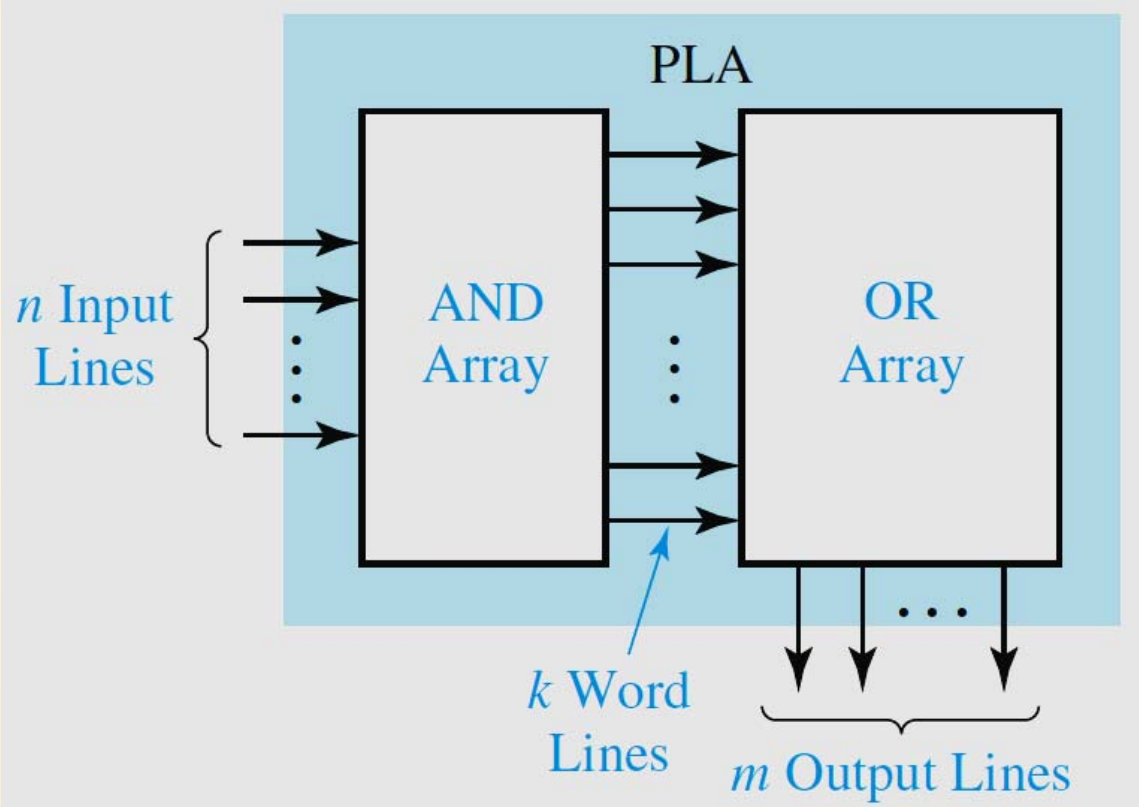


Read Only Memory (ROM)

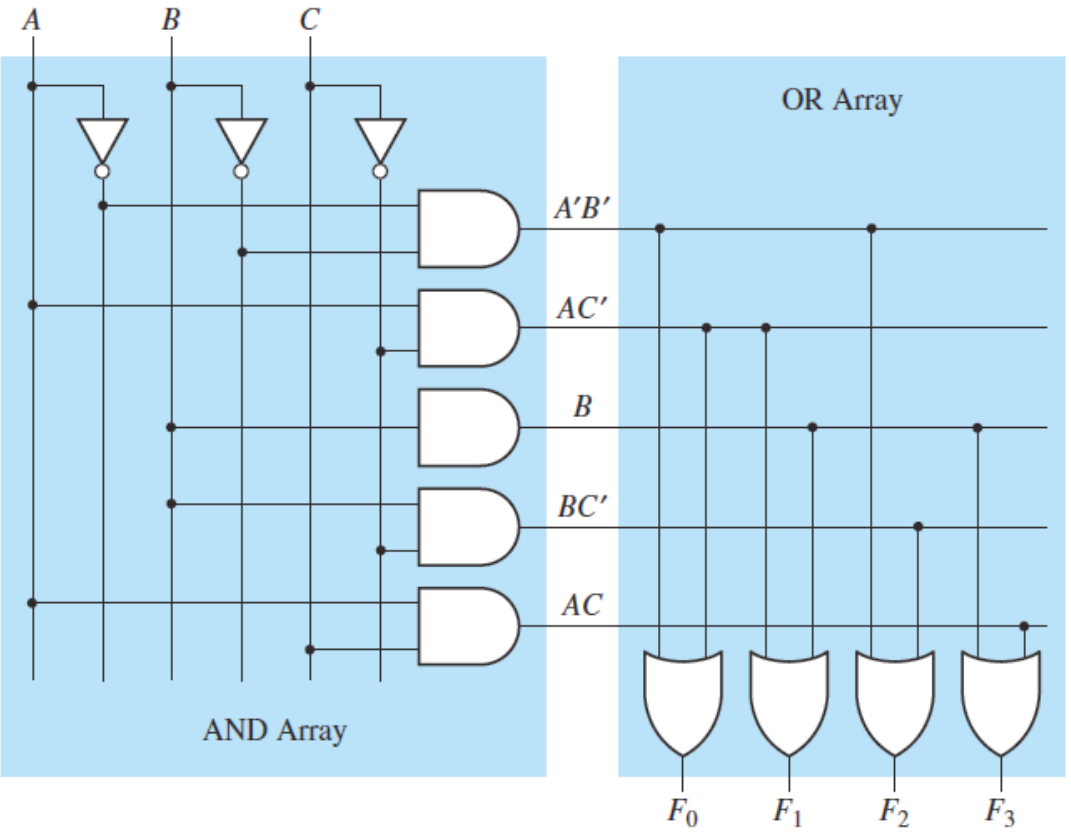
Input W X Y Z	Hex Digit	ASCII Code for Hex Digit A_6 A_5 A_4 A_3 A_2 A_1 A_0
0 0 0 0	0	0 1 1 0 0 0 0
0 0 0 1	1	0 1 1 0 0 0 1
0 0 1 0	2	0 1 1 0 0 1 0
0 0 1 1	3	0 1 1 0 0 1 1
0 1 0 0	4	0 1 1 0 1 0 0
0 1 0 1	5	0 1 1 0 1 0 1
0 1 1 0	6	0 1 1 0 1 1 0
0 1 1 1	7	0 1 1 0 1 1 1
1 0 0 0	8	0 1 1 1 0 0 0
1 0 0 1	9	0 1 1 1 0 0 1
1 0 1 0	A	1 0 0 0 0 0 1
1 0 1 1	B	1 0 0 0 0 1 0
1 1 0 0	C	1 0 0 0 0 1 1
1 1 0 1	D	1 0 0 0 1 0 0
1 1 1 0	E	1 0 0 0 1 0 1
1 1 1 1	F	1 0 0 0 1 1 0



Programmable Logic Array

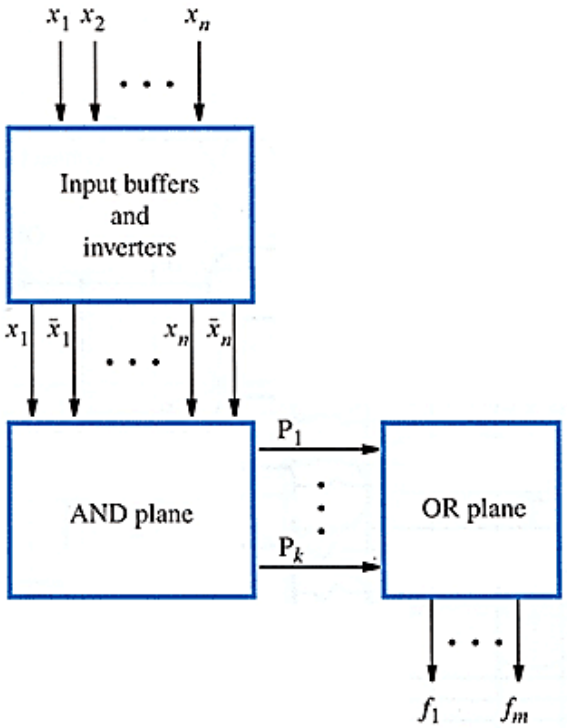


Programmable Logic Array

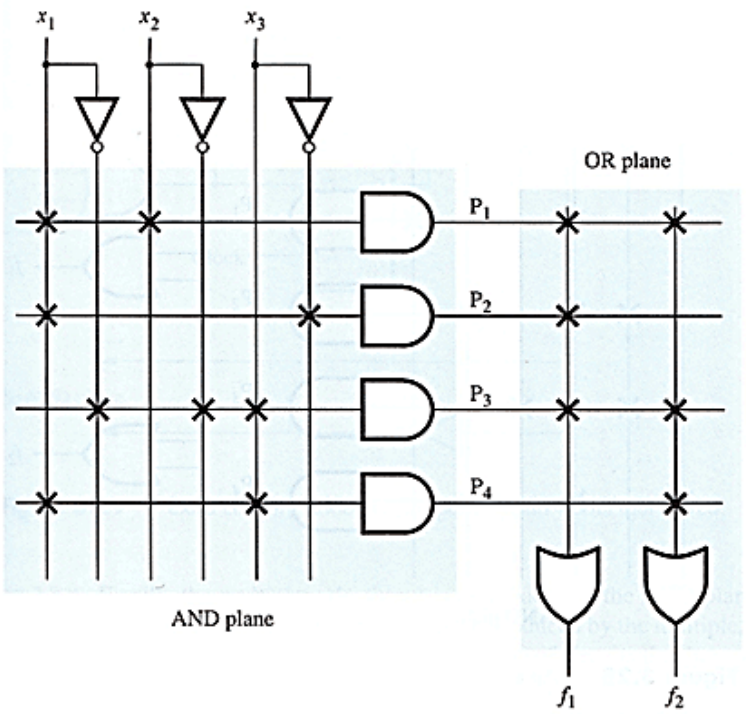


Product Term	Inputs A B C	Outputs F_0 F_1 F_2 F_3	
$A'B'$	0 0 –	1 0 1 0	$F_0 = A'B' + AC'$
AC'	1 – 0	1 1 0 0	$F_1 = AC' + B$
B	– 1 –	0 1 0 1	$F_2 = A'B' + BC'$
BC'	– 1 0	0 0 1 0	$F_3 = B + AC$
AC	1 – 1	0 0 0 1	

Programmable Logic Array

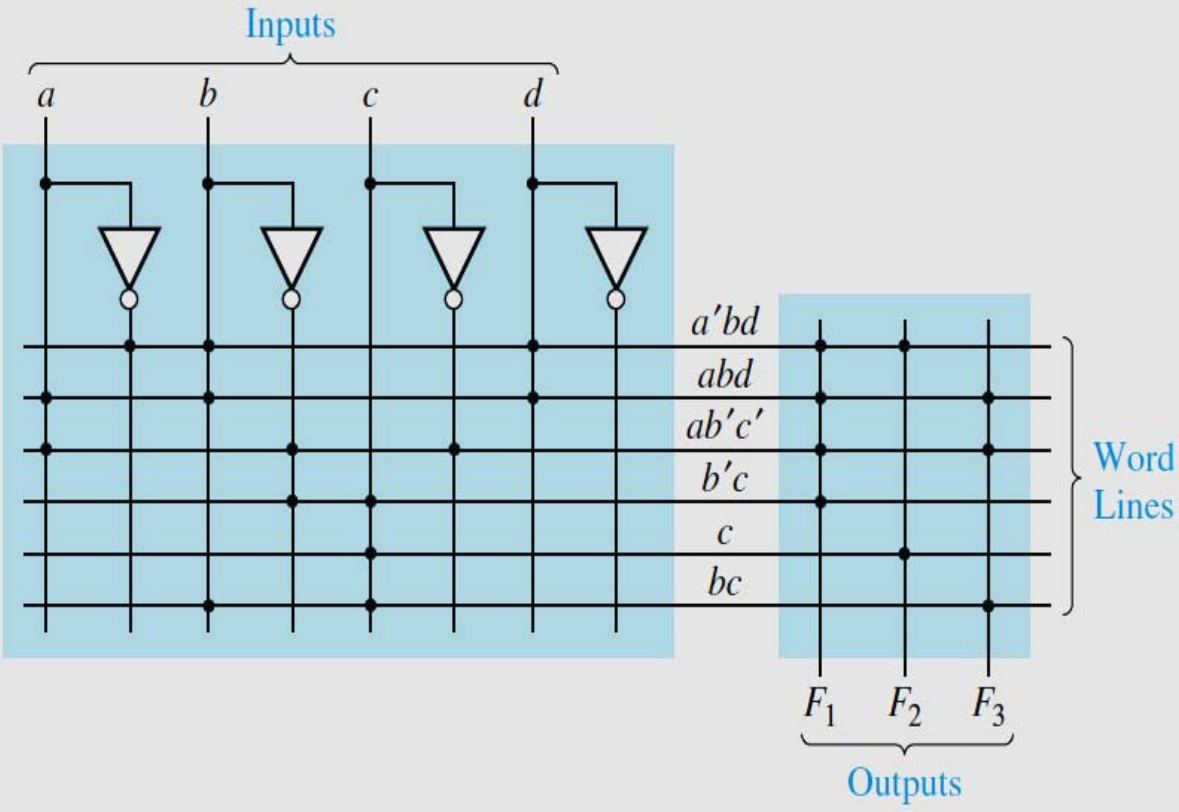


General structure of a PLA



Customary schematic for PLA

Programmable Logic Array

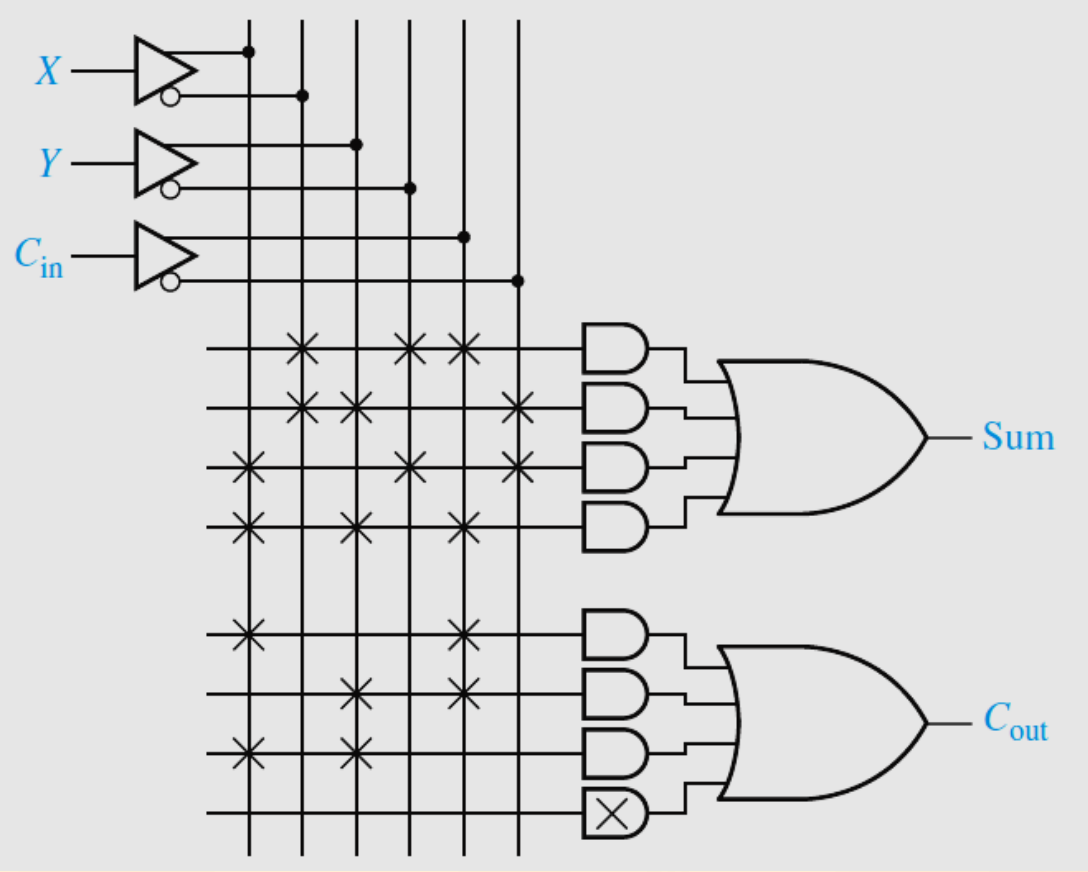


(b) PLA structure

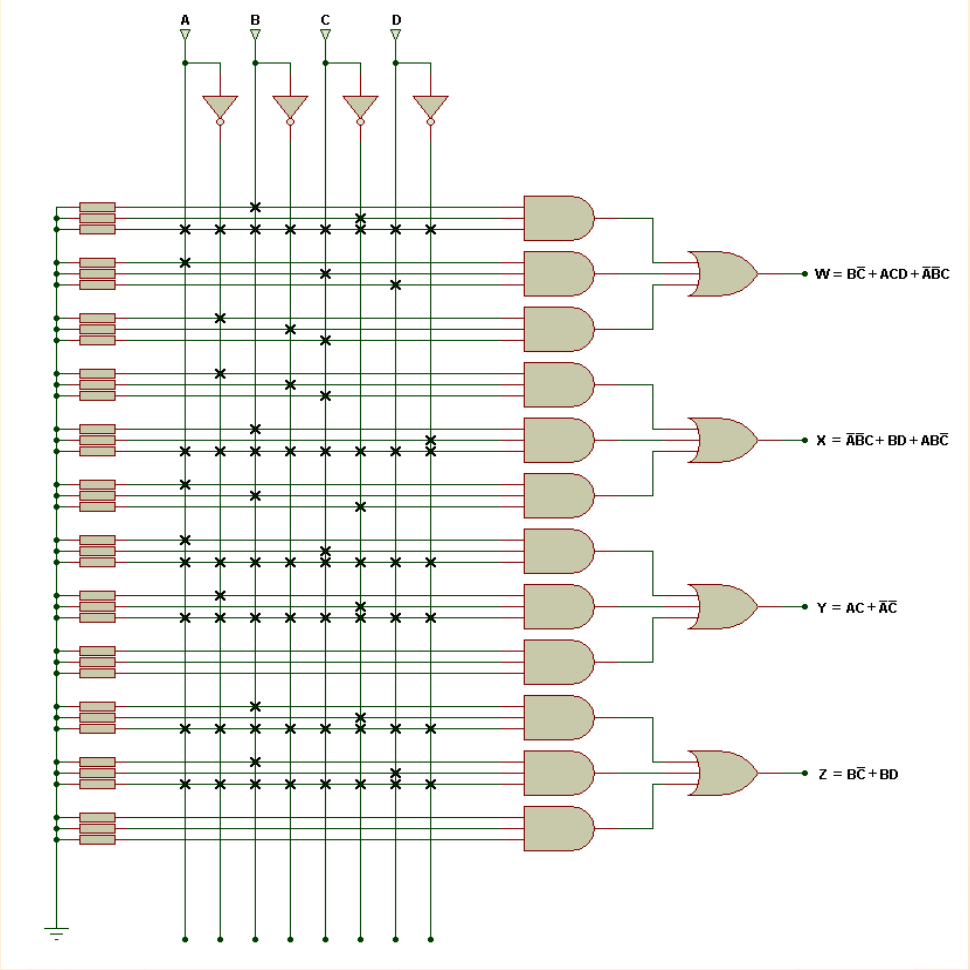
a b c d	f_1 f_2 f_3
0 1 – 1	1 1 0
1 1 – 1	1 0 1
1 0 0 –	1 0 1
– 0 1 –	1 0 0
– – 1 –	0 1 0
– 1 1 –	0 0 1

(a) PLA table

Programmable Logic Array



Programmable Logic Array



THANK YOU