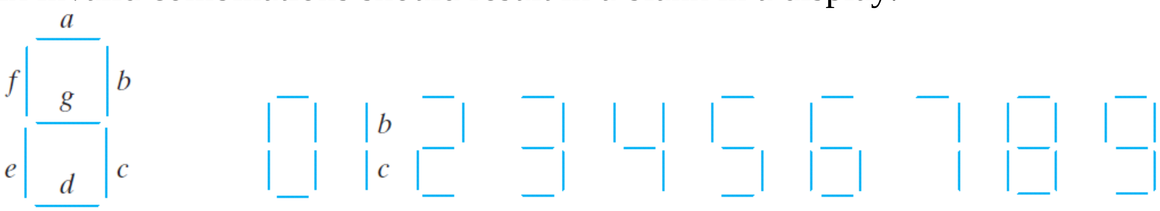
IET Faculty

Dr.Tallal Elshabrawy& Dr.Wassim Alexan

ELCT201 Digital Logic Design
Sheet 5

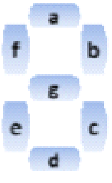
Problem I:

A BCD–to–seven–segment decoder is a combinational circuit that converts a decimal digit in BCD to an appropriate code for the selection of segments in a display indicator used for displaying the decimal digit in a familiar form. The seven outputs of the decoder (a,b,c,d,e,f and g) select the corresponding segments in the display, as shown in the following figure (a). The numeric display chosen to represent the decimal digit is shown in figure (b). Using a truth table and K–maps, design the BCD–to–seven–segment decoder using a minimum number of gates. The six invalid combinations should result in a blank in a display.

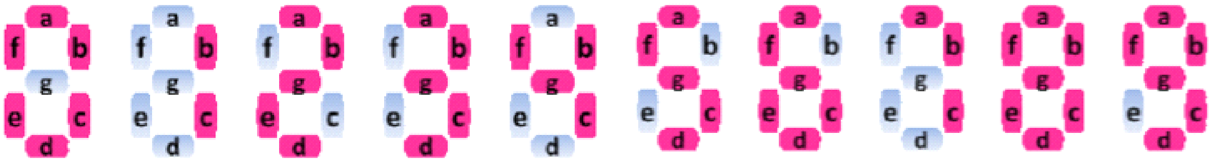


(a) Segment designation (b) Numerical designation for display

Solution:

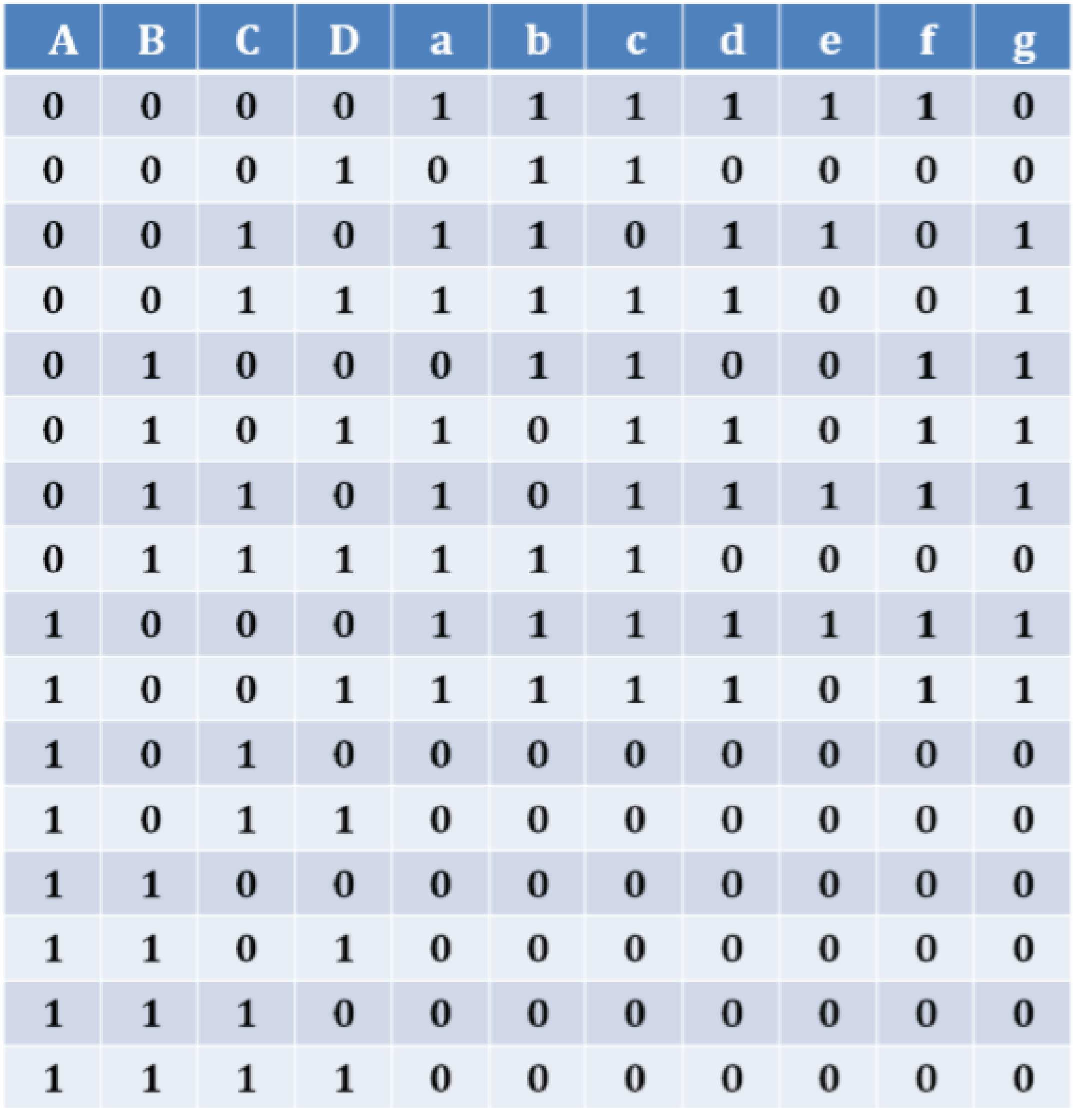


P.2a



1:7- segment display.

- The input to the 7-segment display is number to be displayed, largest number to be displayed on a single 7-segment display is 9, which can be represented in 4 bits. Hence, **input** is 4 bit number (ABCD) to be displayed and **output** is 7-bit number which are 7 segments (a to g).

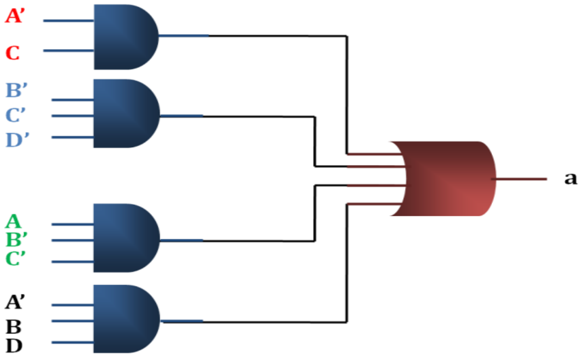


A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

a

AB \ CD	00	01	11	10
00	1		1	1
01		1	1	1
11				
10	1	1		

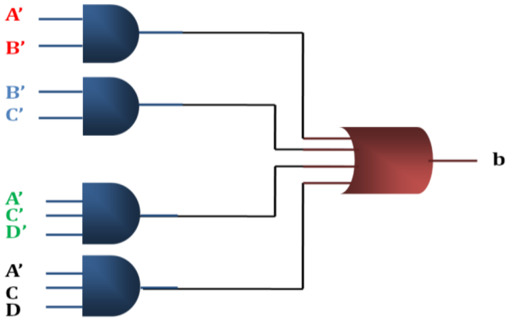
a = A’C+ AB’C’ + B’C’D’ + A’BD



b

AB \ CD	00	01	11	10
00	1	1	1	1
01	1		1	
11				
10	1	1		

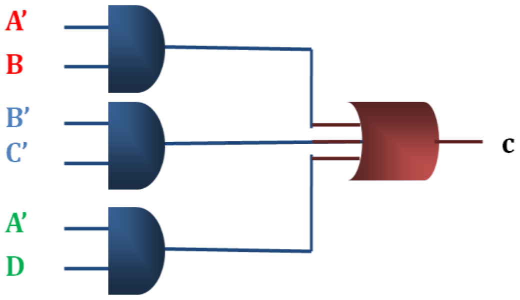
b = A’B’+ A’C’D’ + B’C’ + A’CD



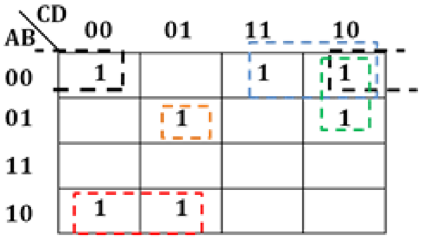
c

AB \ CD	00	01	11	10
00	1	1	1	
01	1	1	1	1
11				
10	1	1		

c = A’B+ A’D + B’C’

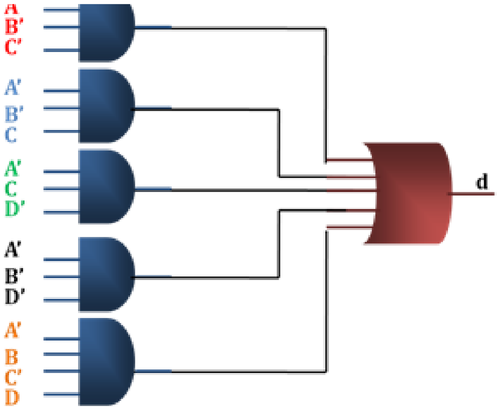


d

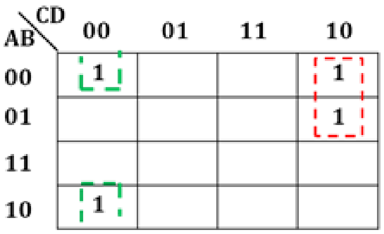


AB \ CD	00	01	11	10
00	1		1	1
01		1		1
11				
10	1	1		

d = AB'C'+ A'CD' + A'B'C + A'B'D' + A'BC'D

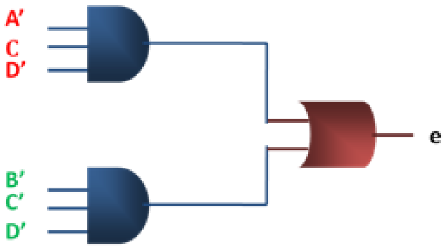


e

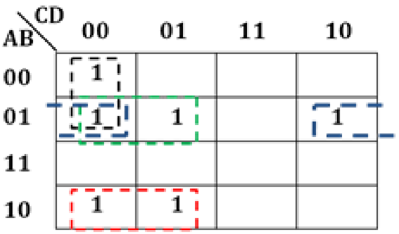


AB \ CD	00	01	11	10
00	1			1
01				1
11				
10	1			

e = A'CD'+ B'C'D'

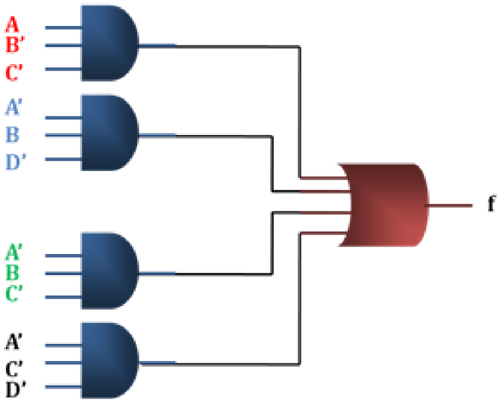


f

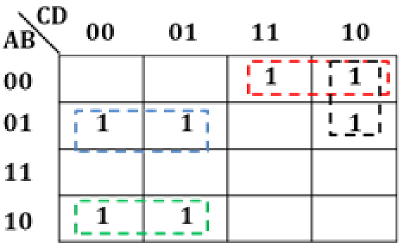


AB \ CD	00	01	11	10
00	1			
01	1	1		1
11				
10	1	1		

f = AB'C'+ A'BC' + A'BD' + A'C'D'

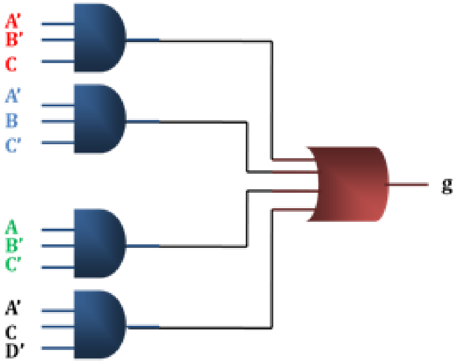


g



AB \ CD	00	01	11	10
00			1	1
01	1	1		1
11				
10	1	1		

g = A'B'C+ AB'C' + A'BC'+ A'CD'



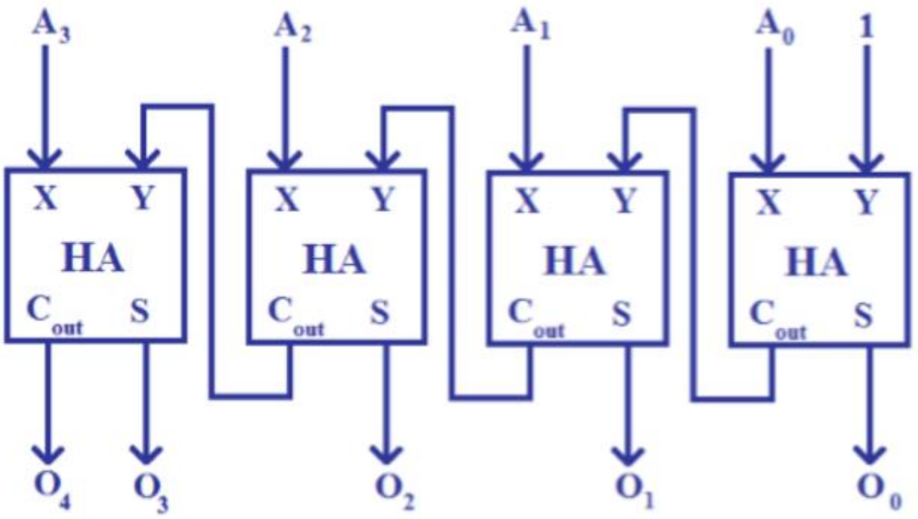
Problem II: Design a 4–bit combinational circuit incrementer (a circuit that adds one to a 4–bit binary number). The circuit can be designed using four half–adders.

Solution:

4 bit incrementer:

$$\begin{array}{r} A_3\ A_2\ A_1\ A_0 \\ 1 \\ \hline 0_4\ 0_3\ 0_2\ 0_1\ 0_0 \end{array} \quad +$$

Half Adder is used to add two input variables, so we will need 4 half adders to perform the 4-bit incrementer operation.



Problem III: Design a binary multiplier that multiplies two 4–bit numbers. Use AND gates and binary adders.

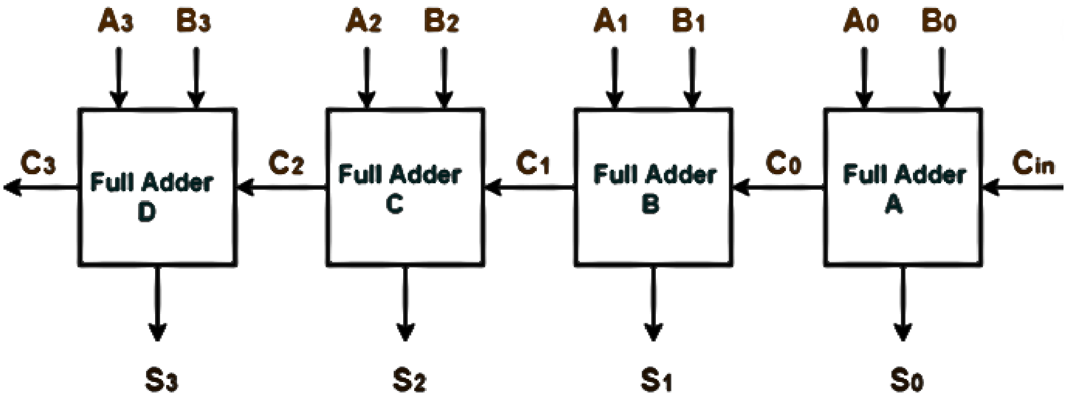
Solution:

Multiplying two 4-bit numbers Multiplications yields the following:

$$\begin{array}{cccccccc}
 & & & \times & a_3 & a_2 & a_1 & a_0 \\
 & & & & b_3 & b_2 & b_1 & b_0 \\ \cline{5-8}
 & & & & a_3b_0 & a_2b_0 & a_1b_0 & a_0b_0 \\
 & & & a_3b_1 & a_2b_1 & a_1b_1 & a_0b_1 & \\
 & & a_3b_2 & a_2b_2 & a_1b_2 & a_0b_2 & & \\
 & a_3b_3 & a_2b_3 & a_1b_3 & a_0b_3 & & & \\ \hline
P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
\end{array}$$

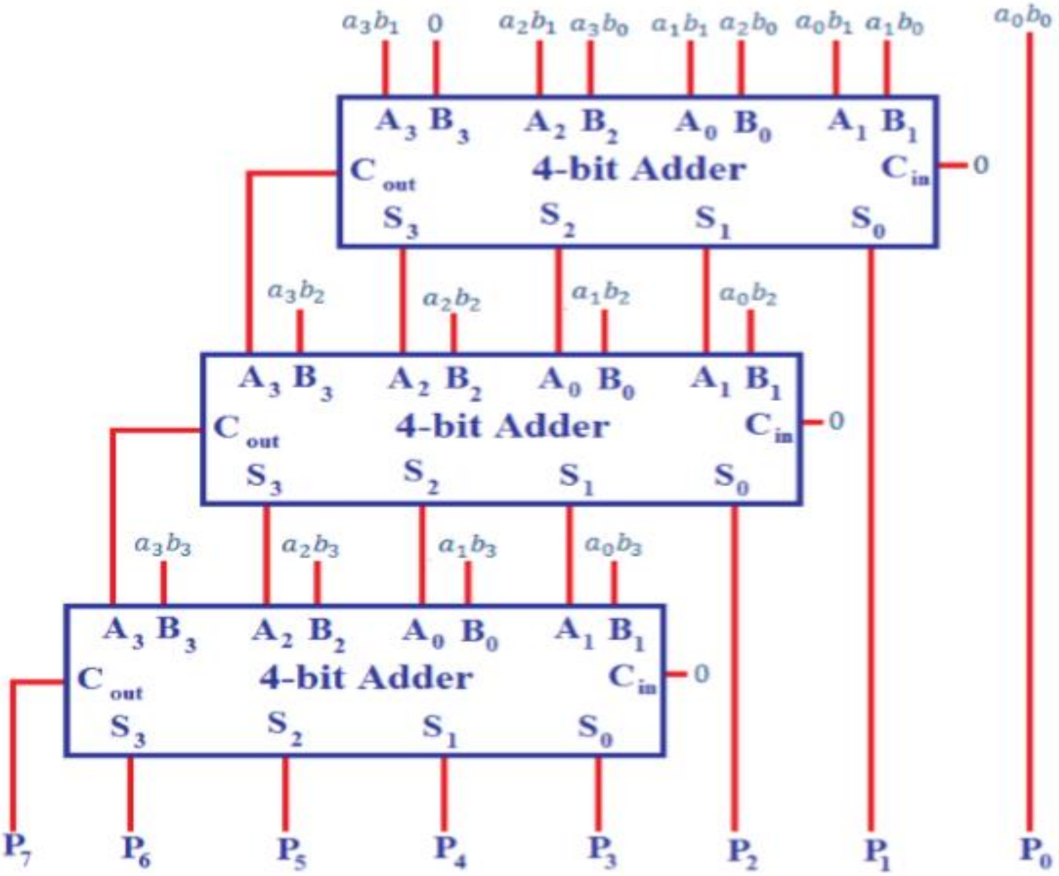
Implementation using 4-bit Full adder Blocks:

4-bit Full Adder:



4 bit Multiplier:

The 4-bit multiplier can be implemented using 3 4bit adder blocks as follows:

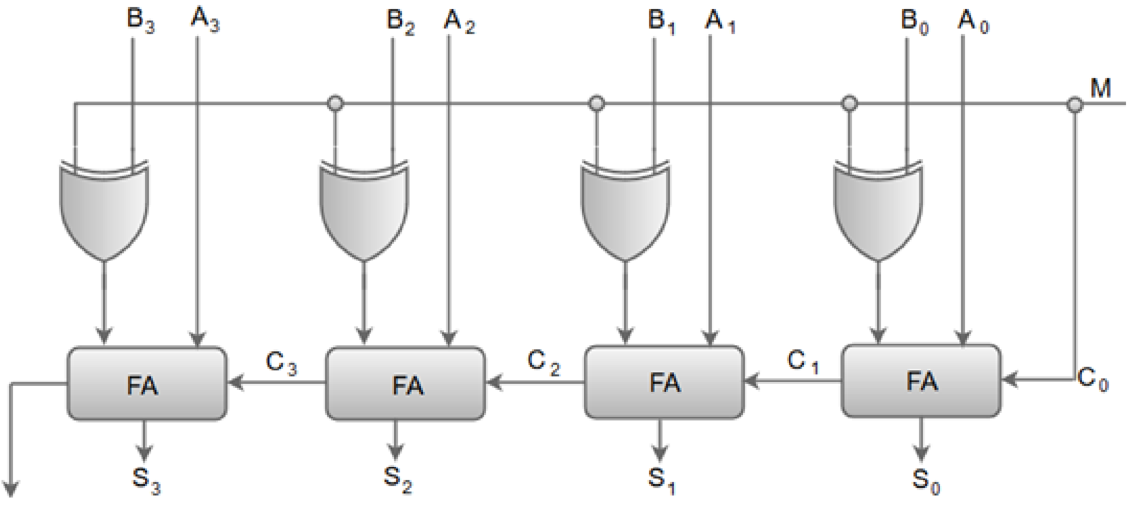


Problem IV: Design a combinational circuit that takes two 4–bit numbers A and B and a 1–bit input C. The circuit should function as an adder or a subtractor of the inputs A and B, such that based on the input C it toggles between addition and subtraction. Assume that A>B.

Solution:

- Xor Function can be used as a controller to obtain A or A’ , XOR outputs inverse of input1 (A) if input2 (B) is 1

A	B	A **XOR** B
0	0	0
0	1	1
1	0	1
1	1	0



B_3
A_3
B_2
A_2
B_1
A_1
B_0
A_0
M
FA
C_3
FA
C_2
FA
C_1
FA
C_0
S_3
S_2
S_1
S_0