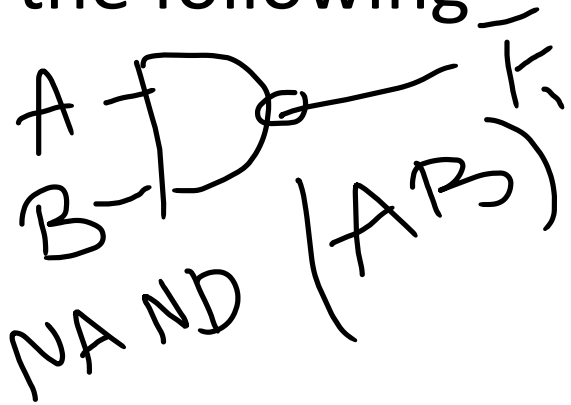
ECE124: Discussion

Discussion #8

Yeonsik Noh, PhD

3.21 Draw (a) the multiple-level NAND circuit for the following expression and (b) repeat (a) for a NOR circuit.

$$w(x + y + z) + xyz$$



F(wxyz) w(x+y+z)+xyz

(w(x+y+z)+xyz)'

(wx+wy+wz+xyz)'

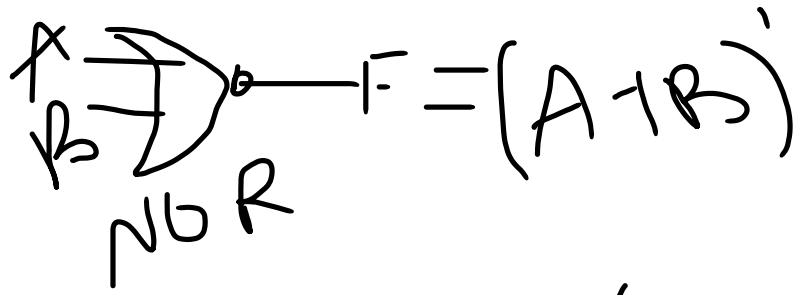
(wz)'(wy)'(wz)'(xyz)'

(F')' =((wz)'(wy)'(wz)'(xyz)')'

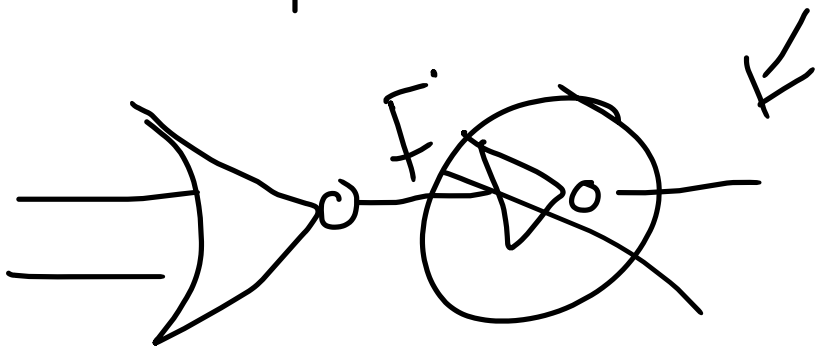
F' = (w(x'y'z')'+xyz)'
(w(x'y'z')')'(xyz)'
((w(x'y'z')')'(xyz)')'

3.21 (b) repeat (a) for a NOR circuit.

w(x + y + z) + xyz



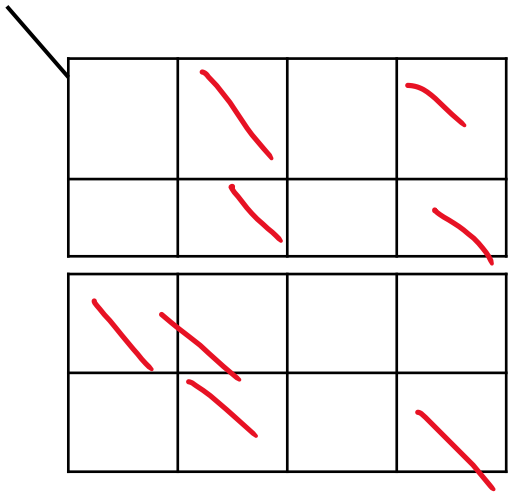
F' (w(x+y+z)+xyz)'
(w(x+y_z))'(xyz)'
((w'+(x+y+z)')(x'+y'+z'))'
(((w'+(x+y+z)')'+(x'+y'+z')')')'

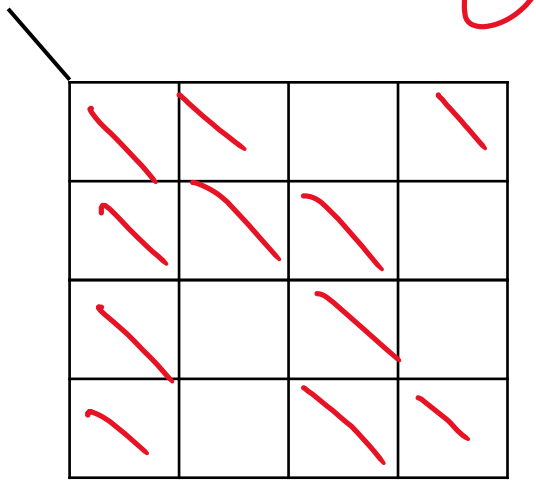


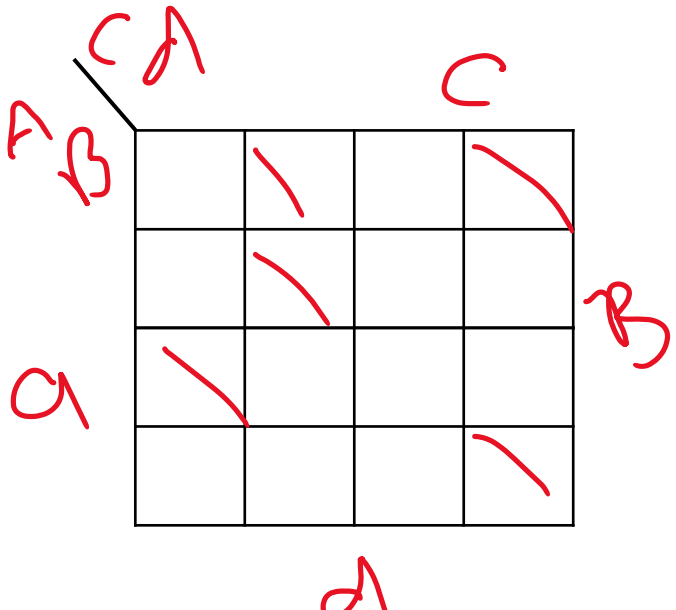
3.26 With the use of maps, find the simplest sum-of-products form of the function F = *fg*, where

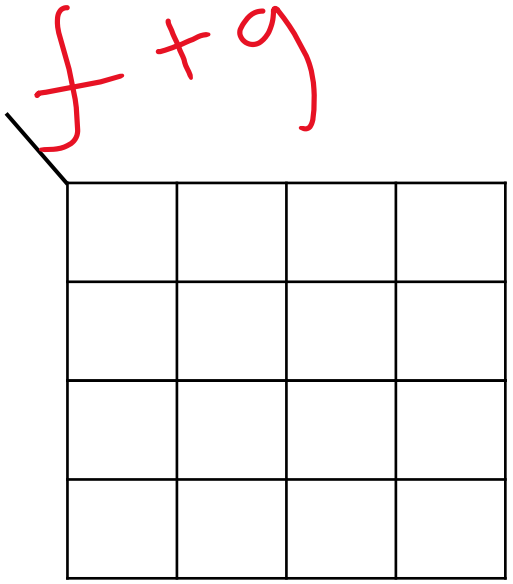
f = abc’ + c’d +a’cd’ + b’cd’

g = (a + b + c’ + d’)(b’ +c’ + d)(a’ + c + d’)

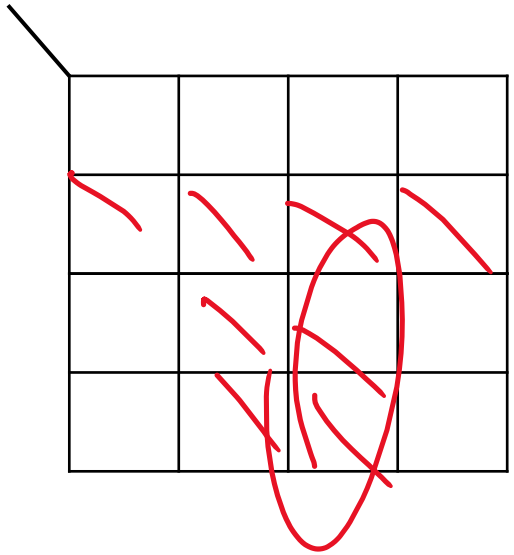


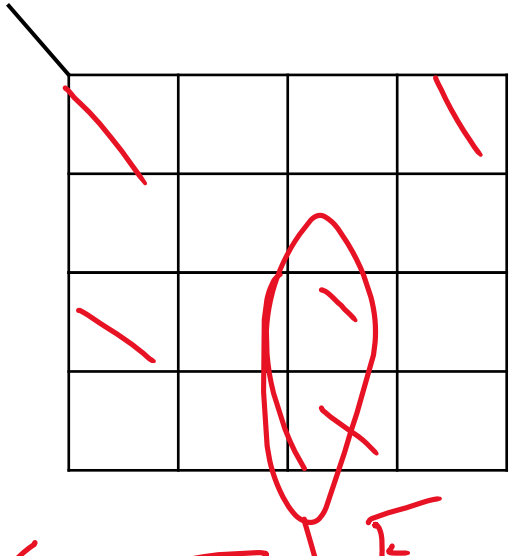


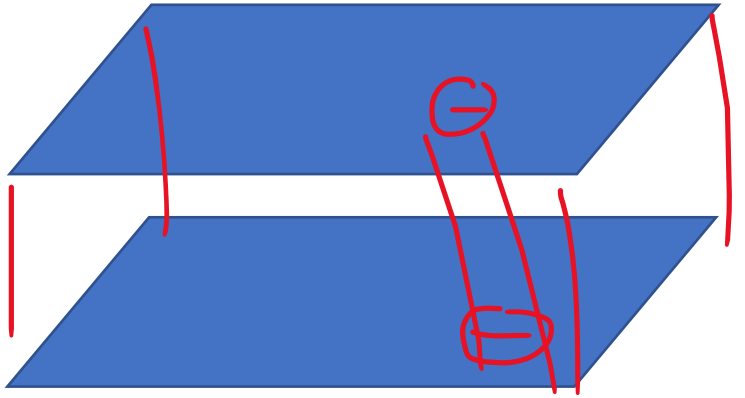


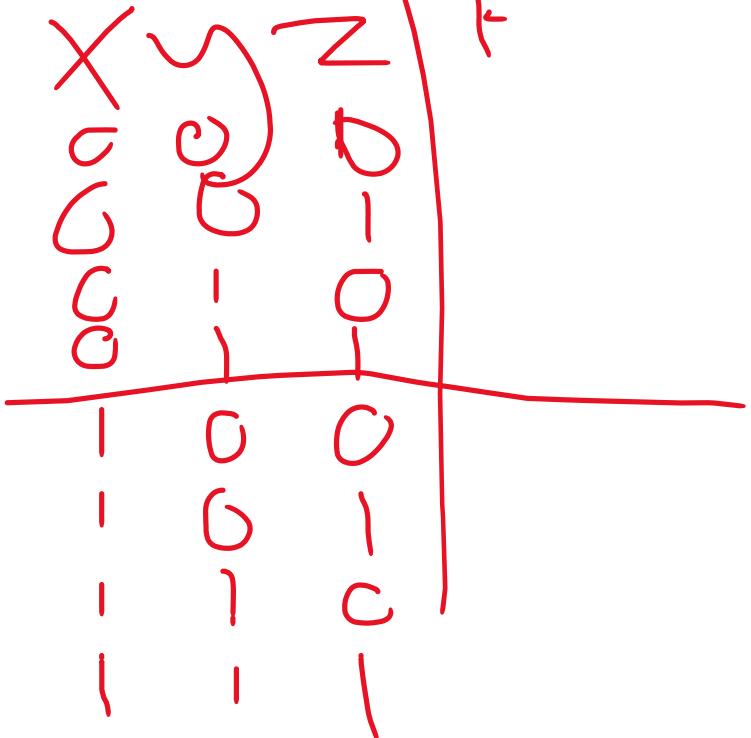


- 5 variables K-map: F(A, B, C, D, E) = Σ(4, 5, 6, 7, 9, 11, 13, 15, 16, 18, 27, 28, 31)



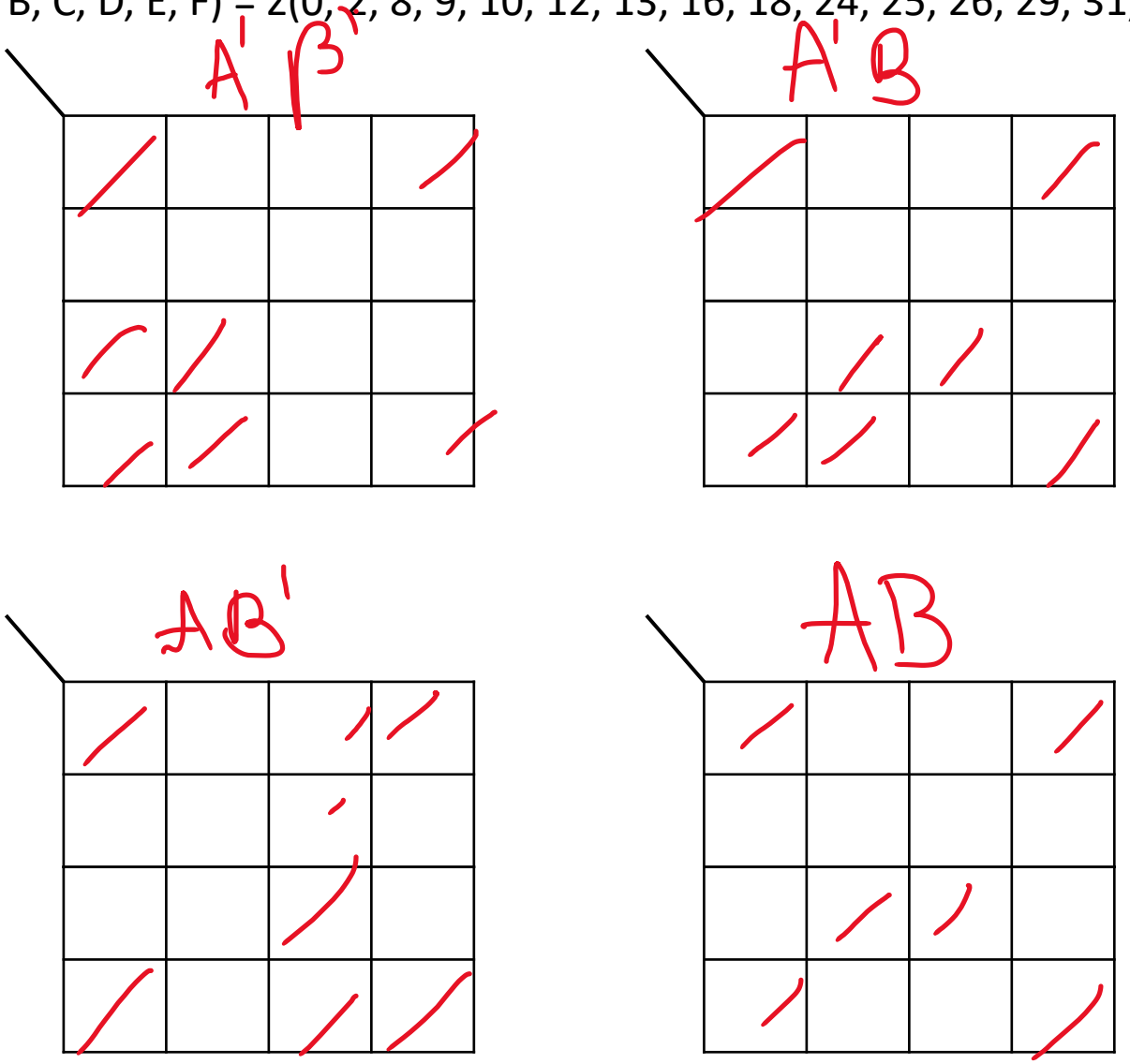


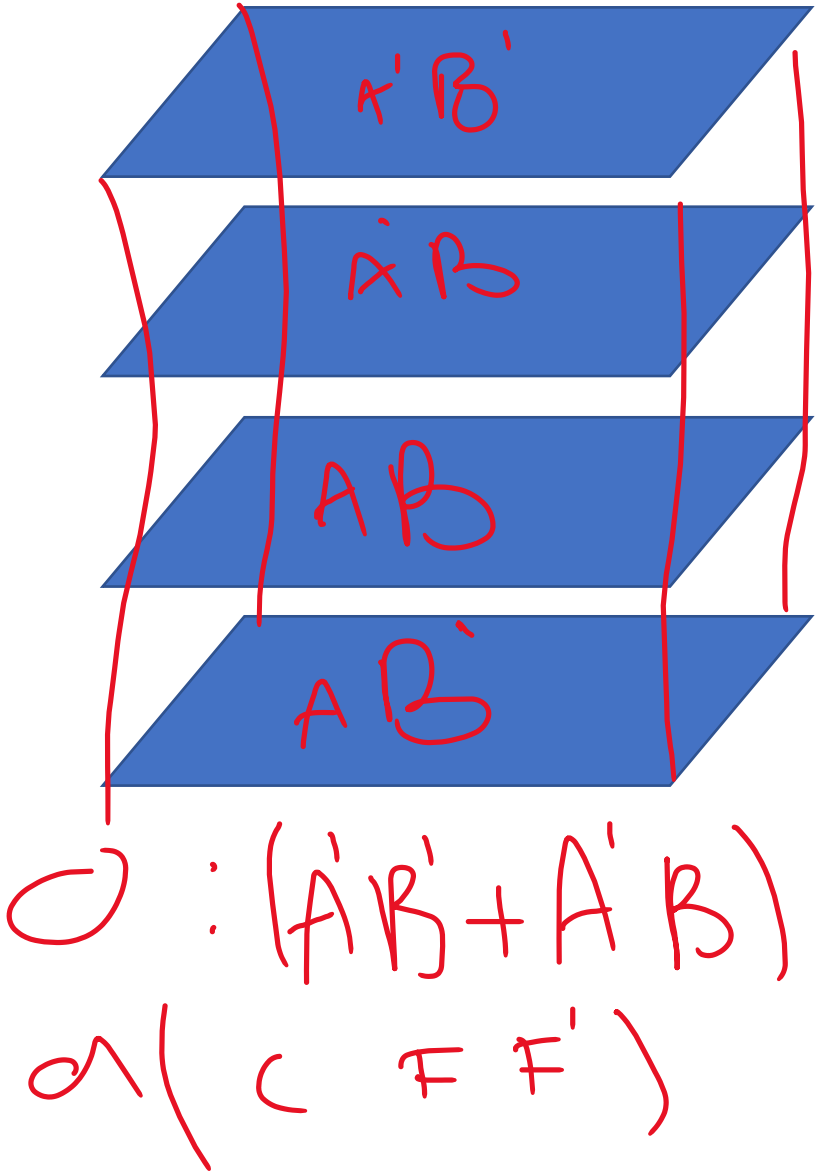




- 6 variables K-map:

F(A, B, C, D, E, F) = Σ(0, 2, 8, 9, 10, 12, 13, 16, 18, 24, 25, 26, 29, 31, 32, 34, 35, 39, 40, 42, 43, 47, 48, 50, 56, 58, 61, 63)





4.13 The adder-substractor circuit of Fig. 4.13 has the following values for mode input M and data inputs A and B.

In each case, determine the values of the four SUM outputs,

the carry C, and overflow V.

	M	A	B
(a)	0	0111	0110
(b)	1	1000	1001

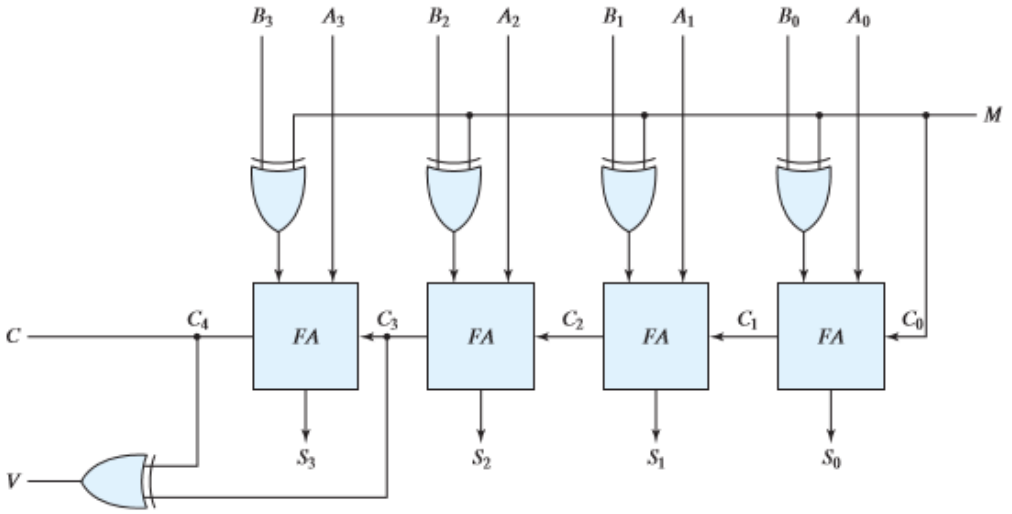


FIGURE 4.13
Four-bit adder–subtractor (with overflow detection)

4.13 The adder-substractor circuit of Fig. 4.13 has the following values for mode input M and data inputs A and B. determine the values of the four SUM outputs, the carry C, and overflow V.

	M	A	B
(a)	0	0111	~~0110~~ 1011(?)

-8<4 bit signed # < +7

7+(-5) = 2

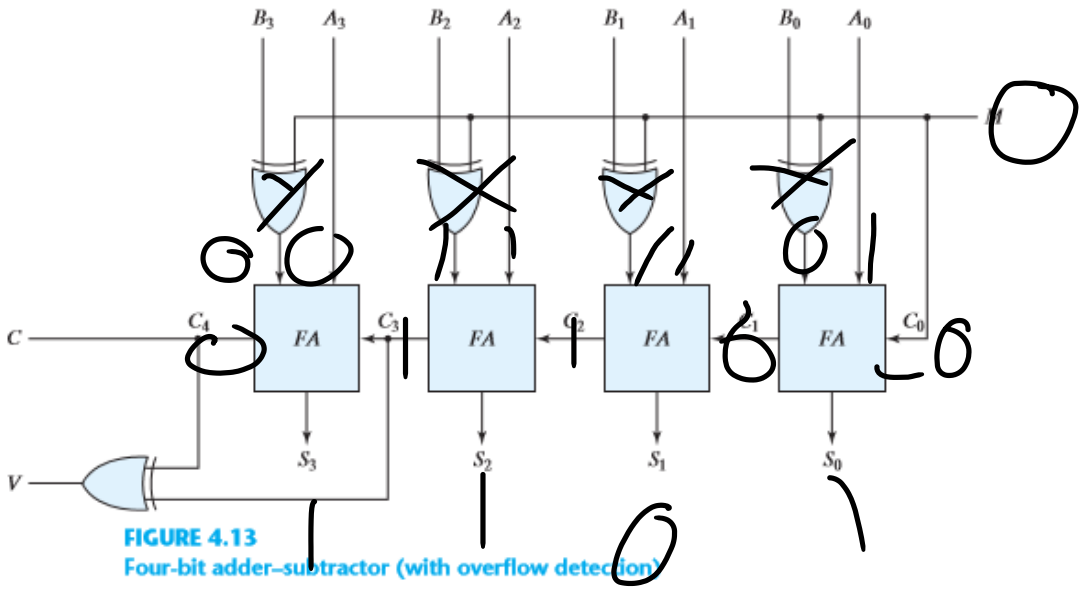
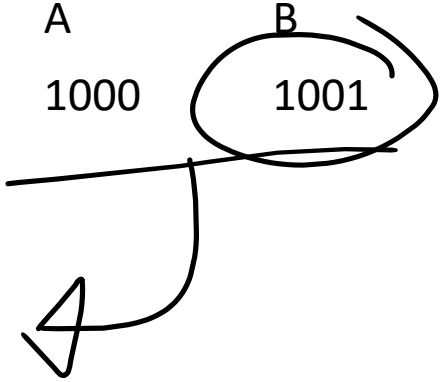
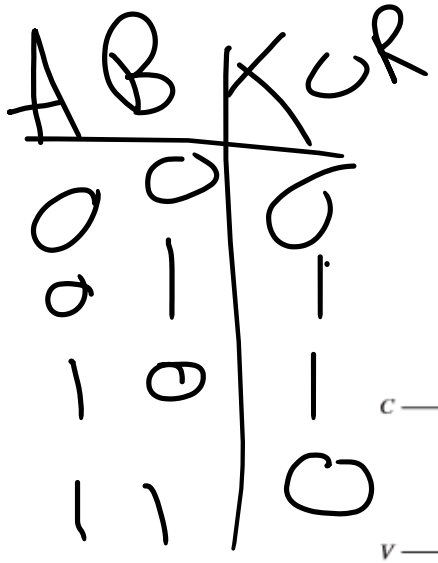


FIGURE 4.13
Four-bit adder–subtractor (with overflow detection)

4.13 The adder-substractor circuit of Fig. 4.13 has the following values for mode input M and data inputs A and B. determine the values of the four SUM outputs, the carry C, and overflow V.

	M	A	B
(b)	1	1000	1001





(a)-(b)
(-a)-(-b)
(-a)+(-(-b))

1000
1000

1001
0111

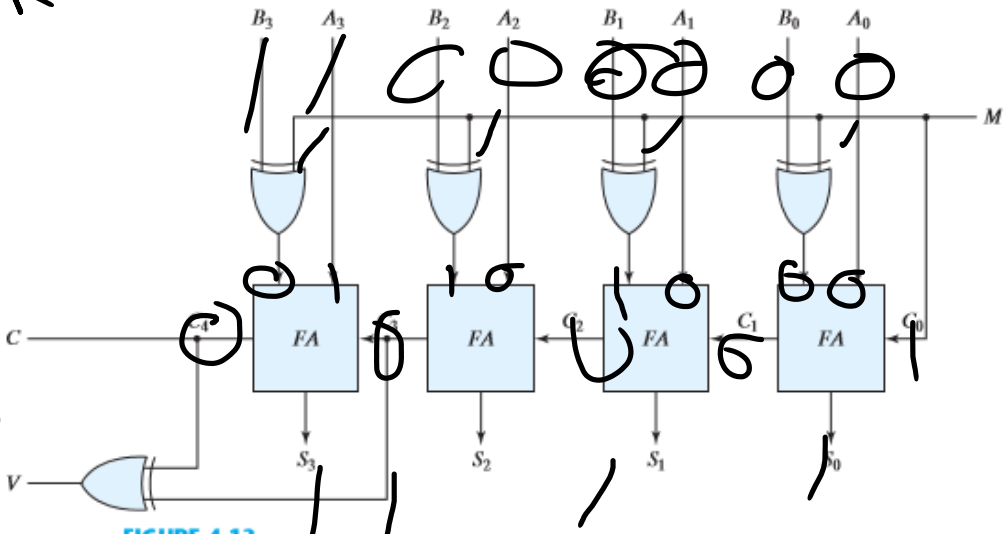


FIGURE 4.13
Four-bit adder–subtractor (with overflow detection)

- Carry Lookahead Equations

Recall
- $G_i = A_i B_i$ ← "carry generate" for stage i
- $P_i = A_i \oplus B_i$ ← "carry propagate" for "

Full adder
- $C_{i+1} = A_i B_i + (A_i \oplus B_i) C_i$
- $S_i = A_i \oplus B_i \oplus C_i$

- $C_{i+1} = G_i + P_i C_i$
- $S_i = P_i \oplus C_i$

- C_0 = input carry
- $C_1 = G_0 + P_0 C_0$
- $C_2 = G_1 + P_1 C_1 = G_1 + P_1(G_0 + P_0 C_0) = G_1 + P_1 G_0 + P_1 P_0 C_0$
- $C_3 = G_2 + P_2 C_2 = G_2 + P_2(G_1 + P_1 G_0 + P_1 P_0 C_0) = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$
- $C_4 = G_3 + P_3 C_3 = G_3 + P_3(G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0) = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$

- $S_0 = P_0 \oplus C_0$
- $S_1 = P_1 \oplus C_1$
- $S_2 = P_2 \oplus C_2$
- $S_3 = P_3 \oplus C_3$

- Design a combinational circuits for parity generator/checker