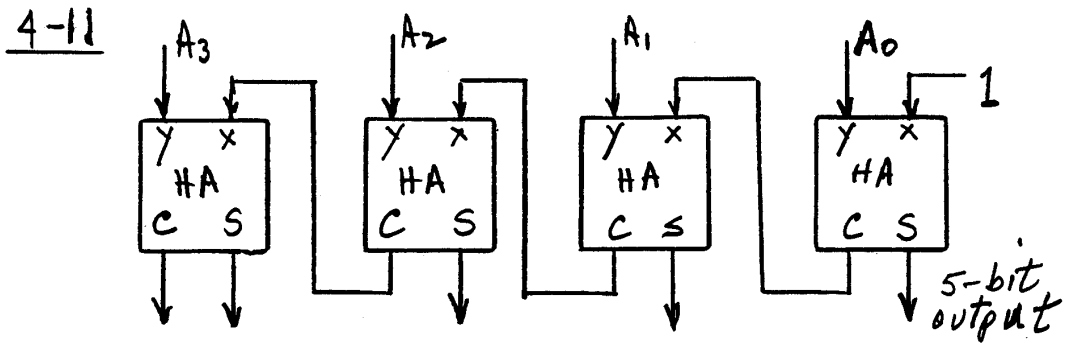
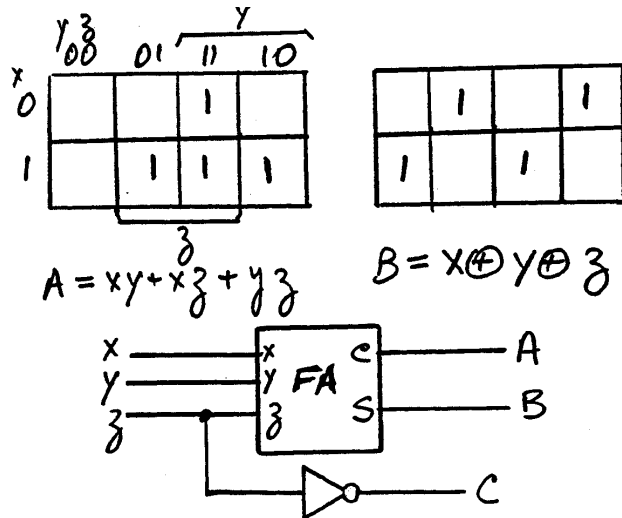


4-5

inputs			outputs		
x	y	z	A	B	C
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	1	0

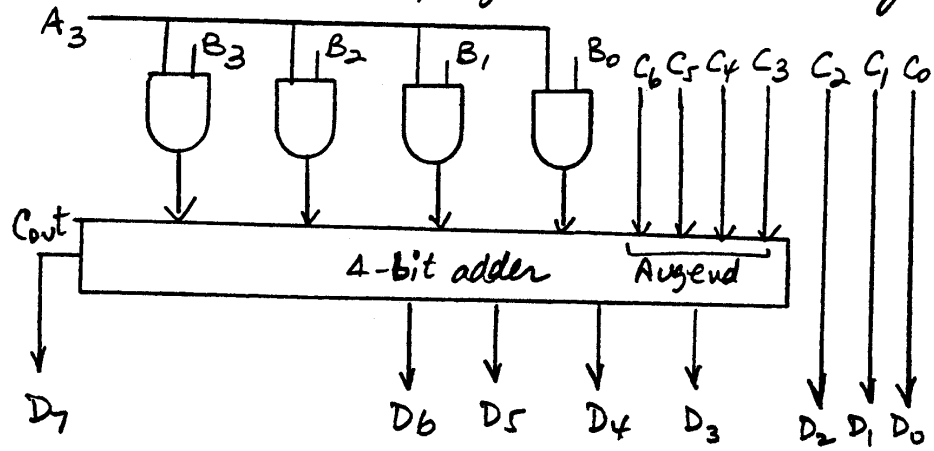


4-13

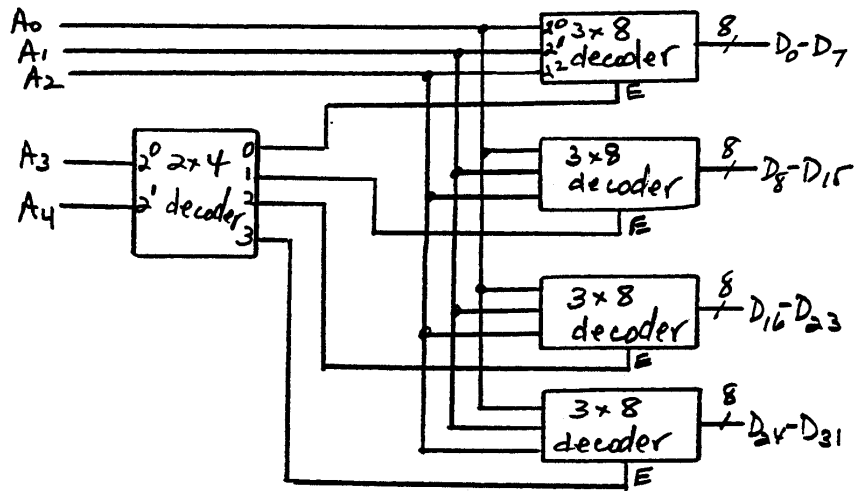
	SUM	C	V
(a)	1101	0	1
(b)	0001	1	1
(c)	0100	1	0
(d)	1011	0	1
(e)	1111	0	0

4-20

Add to the circuit of Figure 4-16 the following:

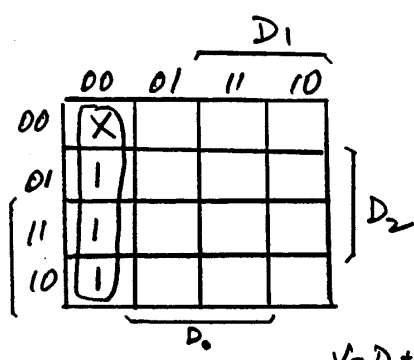


4-25

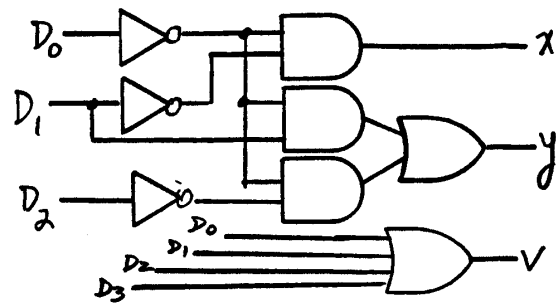


4-29

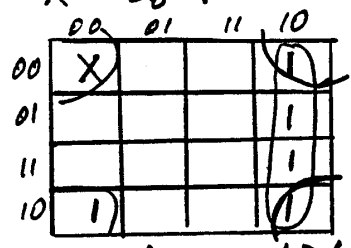
Inputs				Outputs		
D_3	D_2	D_1	D_0	X	Y	V
0	0	0	0	x	x	0
x	x	x	1	0	0	1
x	x	1	0	0	1	1
x	1	0	0	1	0	1
1	0	0	0	1	1	1



$V = D_0 + D_1 + D_2 + D_3$

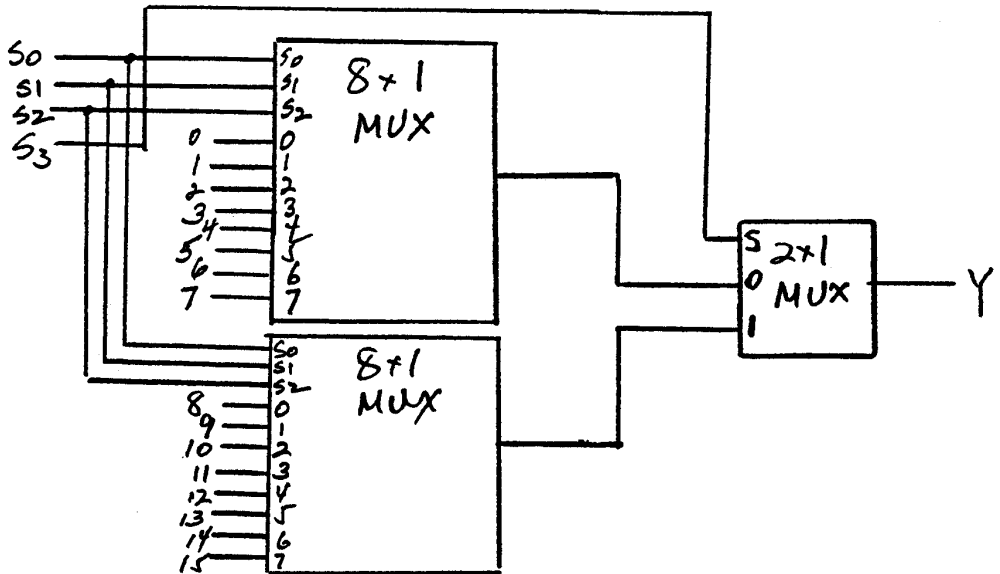


$X = D_0' D_1$



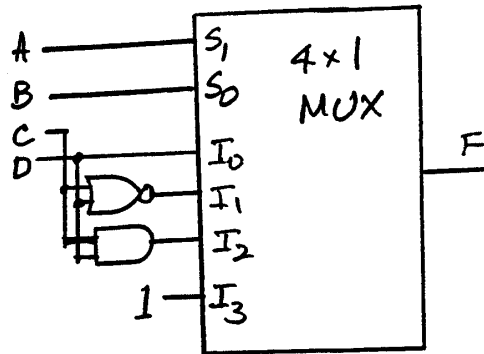
$Y = D_0' D_1 + D_0' D_2$

4-31



4-35

ABCD	F	
0000	0	AB=00
0001	1	F=D
0010	0	
0011	1	
0100	1	AB=01
0101	0	F=C'D'
0110	0	=(C+D)'
0111	0	
1000	0	AB=10
1001	0	F=CD
1010	0	
1011	1	
1100	1	AB=11
1101	1	F=1
1110	1	
1111	1	



4-37

```

module _4bit_add_sub(S,C,A,B,M);
input [3:0] A,B;
input M;
output [3:0] S;
output C;
wire [3:0] BM; //BM = B xor M
wire C1,C2,C3,C; //C = C4 output cary
xor xor0(BM[0],B[0],M),
    xor1(BM[1],B[1],M),
    xor2(BM[2],B[2],M),
    xor3(BM[3],B[3],M);
//Instantiate full adder (HDL Expl 4-2)
fulladder FA0(S[0],C1,A[0],BM[0],M),
           FA1(S[1],C2,A[1],BM[1],C1),
           FA2(S[2],C3,A[2],BM[2],C2),
           FA3(S[3],C,A[3],BM[3],C3);
endmodule

```