Joseph Hyatt
CSE 3100-01/02
007131989

Homework 2

Chapter (3): 3.4 f, h; 3.7 b, d ; 3.15 a, b

3.4) f) F (w, x, y, z) = $\sum$ (8, 10, 12, 13, 14)

The K-map

wx \ yz	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12} 1	m_{13} 1	m_{15}	m_{14} 1
10	m_8 1	m_9	m_{11}	m_{10} 1

wz' wxy' wz'

There are 2 distinct combinations that are formed: the first, represented by the black is a quad (combo of 4 squars) with the value wz. The second represented by the color blue is a pair (2 squats) with value wxy. The simplified boolean function is the sum of the individual terms obtained. $F(w, x, y, z) = wz' + wxy'$

h) F (w, x, y, z) = $\sum$ (2,3,6,7,8,9,12,13)

The K-map

wx \ yz	00	01	11	10
00	m_0	m_1	m_3 1	m_2 1
01	m_4	m_5	m_7 1	m_6 1
11	m_{12} 1	m_{13} 1	m_{15}	m_{14}
10	m_8 1	m_9 1	m_{11}	m_{10}

$w'y$

wy'

There are 2 distinct combinations that are formed: the first, represented by the blue color is a quad (combination of 4 squares) with the value w'y, and the second, represented by the black color is a quad with value wy'. The simplified boolean function is the sum of the individual terms obtained. $F(w, x, y, z) = w'y + wy'$

3.7) b) AD' + B'C'D + BCD' + BC'D

The four variable map for the boolean function is

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

The K-map with regrouping minterms to simplify the functions are

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

The minterms B'C'D and BC'D are combined to form a larger cube C'D. The minterms AD' and B'C'D cannot combine to form larger cubes. The simplified boolean expression is the sum of C'D, AD', and B'C'D. $F = C'D + AD' + BCD'$.

D) wxy + xz + wx'z + w'x

The four variable map for the boolean function is

wx \ yz	00	01	11	10
00				
01	1	1	1	1
11		1	1	1
10		1	1	

The K-map with regrouping minterms to simplify the functions are

wx \ yz	00	01	11	10
00				
01	1	1	1	1
11		1	1	1
10		1	1	

The minterms AB'C, BCD, and A'B'C are combined to form a larger cube CD. The minterms AB'C, BCD, and ACD' are combined to form a larger cube AC. The minterms B'C'D, A'B'C, and ACD' are combined to form a larger cube B'D'. The minterms BCD, and A'BC'D are combined to form a larger cube A'BD. The simplified boolean expression is the sum of CD, AC, B'D', and A'BD. $F = CD + B'D' + AC + A'BD$

3.15) a) F (x, y, z) = Σ(0, 1, 4, 5, 6) d(x, y, z) = Σ(2, 3, 7)

Karnaugh map of the boolean function

x \ yz	00	01	11	10	
0	m_0 1	m_1 1	m_3 X	m_2 X	
1	m_4 1	m_5 1	m_7 X	m_6 1	1

1. Identify the squares of minterms: 000, 001, 100, 101 and 110. Mark these squares by 1's. Identify the square of minterms: 010, 011 and 111. Mark these squares by X's.

2. Find the possible adjacent squares and make a group combining with X's if necessary. The eight squares including don't care conditions (X's) at the top are adjacent and are surrounded by a blue rectangle.

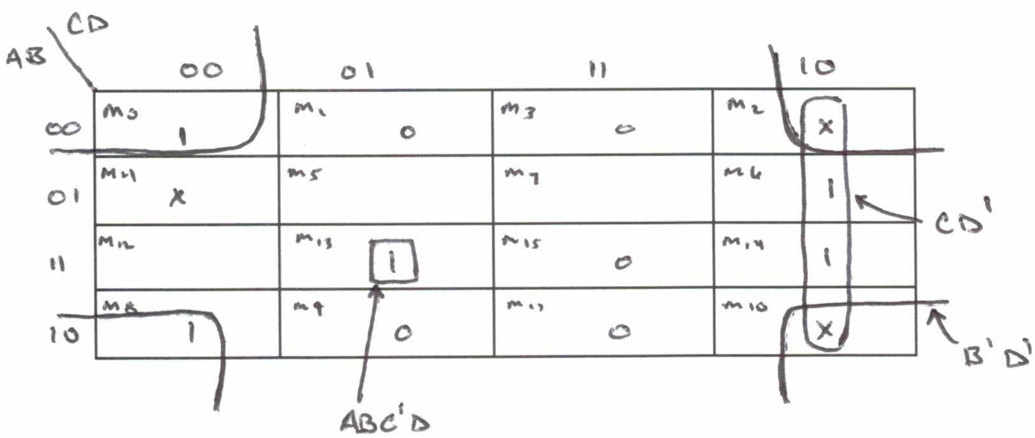
3. The next step is to express each pair of adjacent squares by the common expression. The pair denoted by the blue rectangle can be expressed as 1.

4. The next step is to combine all the expressions of each pair to yield in the sum-of-products. In this problem, the combined expression is 1.

The simplified boolean function is $F(x, y, z) = 1$. The simplified function in sum of minterms form is $F(x, y, z) = \sum(0, 1, 2, 3, 4, 5, 6, 7)$.

b) * F (A, B, C, D) = Σ(0, 6, 8, 13, 14) d(A, B, C, D) = Σ(2, 4, 10)

Karnaugh map of the boolean function



(1) Identify the squares of minterms: 0000, 0110, 1101, 1110 and 1000. Mark these squares by 1's. Identify the square of minterms: 0100, 0010 and 1010 in. Mark these squares by X's.

(2) Find the possible adjacent squares and make a group combining with X's if necessary. The four squares including don't care conditions (X's) at each corner are adjacent and are surrounded by red rectangles to make a pair. Another pair of four vertical squares in the right are adjacent that are surrounded by a black rectangle. Another square surrounded by a blue rectangle does not have any pair but should be considered. Any don't care conditions without its pair to 1's are discarded.

(3) The next step is to express each pair of adjacent squares by the common expression. The pair denoted by the red rectangles can be expressed as B'D'. The pair

denoted by the black rectangle can be expressed as CD' and the pair denoted by the blue rectangle can be expressed as ABC'D.

(4) The next step is to combine all the expressions of each pair to yield in the sum-of-products. In this problem, the combined expression is B'D' + CD' + ABC'D. The simplified boolean function is $F(A, B, C, D) = B'D' + CD' + ABC'D$.

The expression B'D' covers four squares which are 0000, 0010, 1000, and 1010. The corresponding minterms are m0, m2, m8, and m10. Similarly the expression CD' covers four squares which are 0010, 0110, 1010, and 1110. The corresponding minterms are m2, m6, m10 and m14. The expression ABC'D covers only one square which is 1101, the corresponding minterm is m13. The simplified function in sum of minterms is $F(A, B, C, D) = \sum(0, 2, 6, 8, 10, 13, 14)$.

Chapter (4): 4.1 a, b, c; 4.5; 4.35 a, b

4.1) Consider the combinational circuit shown in Fig. P4.1 . (HDL—see Problem 4.49)

(a)* Derive the Boolean expressions for T 1 through T 4 . Evaluate the outputs F 1 and F 2 as a function of the four inputs.

From the combinational circuit shown, T1 is the output of the AND gate whose inputs are B' and C. T1 = B'C

From the combinational circuit shown, T2 is the output of the AND gate whose inputs are A' and B. T2 = A'B

From the combinational circuit shown, T3 is the output of the OR gate whose inputs are A' and T1. T3 = A + T1. Substituting the value of T1 from the equation. T3 = A + B'C

From the combinational circuit shown, T4 is the output of the Ex-OR gate whose inputs are T2 and D and T1. T4 = T2D' + T'2D. Substitute the value of T2. T4 = (A'D)D' + (A'B)'D. Applying de-morgans law T4 = A'BD' + AD + B'D.

F1 is the overall output of the OR gate whose inputs are T3 and T4. F1 = T3 + T4

$$
\begin{aligned}
F_1 &= (A+B'C)+A'BD'+AD+B'D \\
&= A+AD+B'C+A'BD'+B'D \\
&= A(1+D)+B'C+A'BD'+B'D \\
&= A(1)+A'BD'+B'C+B'D
\end{aligned}
$$

$$F_1 = A+A'BD'+B'C+B'D$$

Applying distributive law to this equation

$$
\begin{aligned}
F_1 &= (A+A')(A+BD')+B'C+B'D \\
&= (1)(A+BD')+B'C+B'D \\
&= A+BD'+B'C+B'D
\end{aligned}
$$

The output boolean expression for F1 is A + BD' + B'C + B'D.

From the combination circuit F2 is the overall output of the OR gate whose inputs are D' and T2. F2 = D' + T2. Substitute the value T2 = F2 = D' + A'B

(b) List the truth table with 16 binary combinations of the four input variables. Then list the binary values for T 1 through T 4 and outputs F 1 and F 2 in the table.

A	B	C	D	$B'C$	$B'D$	BD'	$A'BD'$	T_1	T_2	T_3	T_4	F_1	F_2
0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	0	0	1	1	0
0	0	1	0	1	0	0	0	1	0	1	0	1	1
0	0	1	1	1	1	0	0	1	0	1	1	1	0
0	1	0	0	0	0	1	1	0	1	0	1	1	1
0	1	0	1	0	0	0	0	0	1	0	0	0	1
0	1	1	0	0	0	1	1	0	1	0	1	1	1
0	1	1	1	0	0	0	0	0	1	0	0	0	1
1	0	0	0	0	0	0	0	0	0	1	0	1	1
1	0	0	1	0	1	0	0	0	0	1	1	1	0
1	0	1	0	1	0	0	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0	1	0	1	1	1	0
1	1	0	0	0	0	1	0	0	0	1	0	1	1
1	1	0	1	0	0	0	0	0	0	1	1	1	0
1	1	1	0	0	0	1	0	0	0	1	0	1	1
1	1	1	1	0	0	0	0	0	0	1	1	1	0

(c) Plot the output Boolean functions obtained in part (b) on maps and show that the simplified Boolean expressions are equivalent to the ones obtained in part (a).

The karnaugh map for the output function F1 is

AB \ CD	00	01	11	10
00	0	1	1	1
01	1	0	0	1
11	1	1	1	1
10	1	1	1	1

B'C

BD'

A

B'D

The design procedure for the output function F1 is

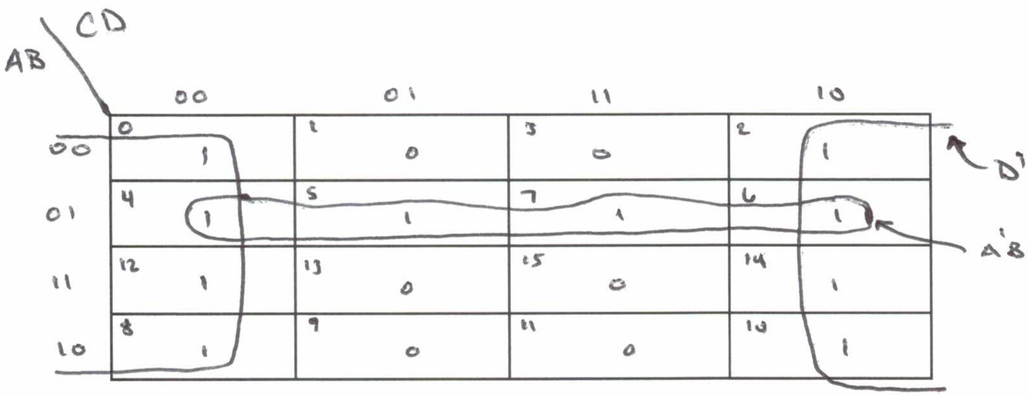
1. Express each pair of adjacent squares by their respective prime implicants. The group of eight 1's denoted by the blue color rectangle box can be expressed as A.

2. The group of four 1's at the right corners denoted by the green color rectangle boxes can be expressed as B'C.

3. The group of four 1's in the left and right denoted by the red color rectangle boxes can be expressed as BD'.

4. The group of four 1's at the top and bottom denoted by the black color rectangle boxes can be expressed as B'D.

Now we combine all the expressions of each group to yield the expression in the sum of products form is F = A + B'C + BD' + B'D. The simplified boolean expression for F1 obtained from the karnaugh map and obtained from the part (a) is the same.



Design procedure of Karnaugh map for the output function F2 is

1. Express each pair of adjacent squares by their respective prime implicants. The group of eight 1's denoted by the red color rectangle boxes in the left and right can be expressed as

2. The group of four 1's denoted by the blue color rectangle box can be expressed as

Now, combine all the expressions of each group to yield the expression in the sum-of-products form. The combination final expression is F2 = D' + A'B.

The simplified boolean expression for F2 obtained from the karnaugh map and obtained from par (a) is the same.

4.5) Design a combinational circuit with three inputs x, y, and z and three outputs A, B, and C. When the binary input is 0, 1, 2, or 3, the binary output is one greater than the input. When the binary input is 4, 5, 6, or 7, the binary output is two less than the input.

Truth Table:

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	0
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	0	1

Table 1

$$\begin{aligned} A &= x'yz + xyz' + xyz \\ &= x'yz + xy(z' + z) \\ &= x'yz + xy(1) \end{aligned}$$

$$A = y(x'z + x) \quad \ldots\ldots (1)$$

Applying distributive law

$$\begin{aligned} A &= y(x' + x)(z + x) \\ &= y(1)(z + x) \end{aligned}$$

$$A = xy + yz \quad \ldots\ldots (2)$$

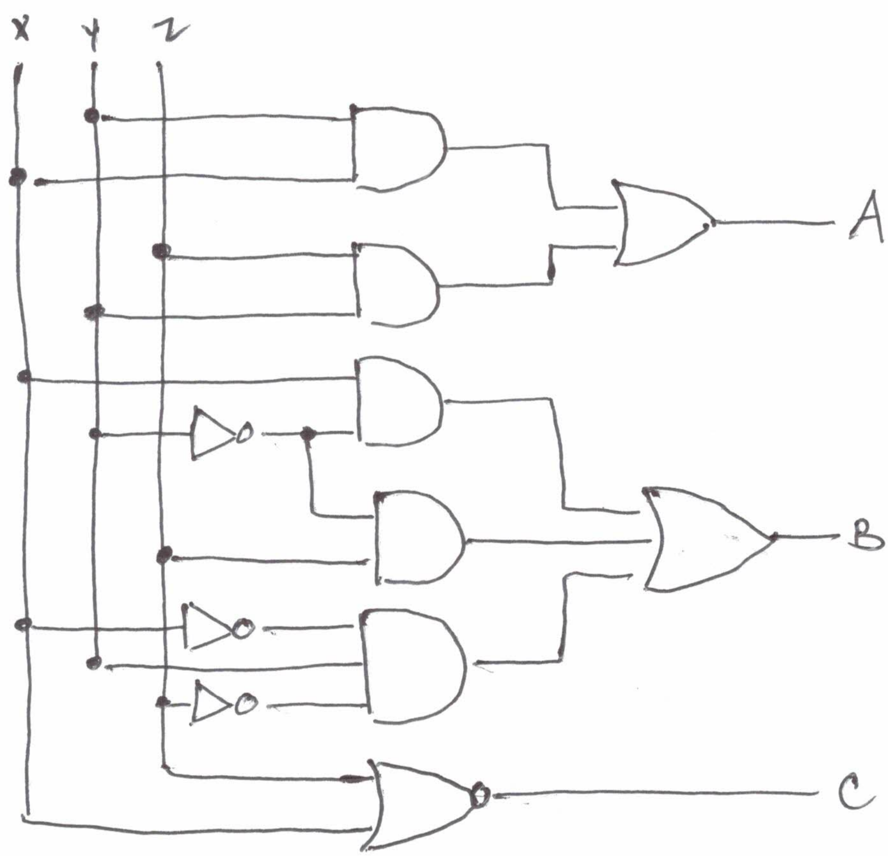
Determine the boolean expression for the binary output B

$$B = x'y'z + x'yz' + xy'z' + \underline{xy'z}$$
$$= x'y'z + x'yz' + xy'z' + xy'z + xy'z \qquad (\text{Since } X + X = X)$$
$$= (x' + x)y'z + x'yz' + xy'(z' + z)$$
$$= (1)y'z + x'yz' + xy'(1)$$
$$B = xy' + y'z + x'yz' \text{ ...... } (3)$$

Determine the boolean expression for the binary output C

$$C = x'y'z' + x'yz' + xy'z + xyz$$
$$= x'z'(y' + y) + xz(y' + y)$$
$$= x'z'(1) + xz(1)$$
$$= x'z' + xz$$
$$C = (x \oplus z)' \text{ ...... } (4)$$

The combinational logic circuit is

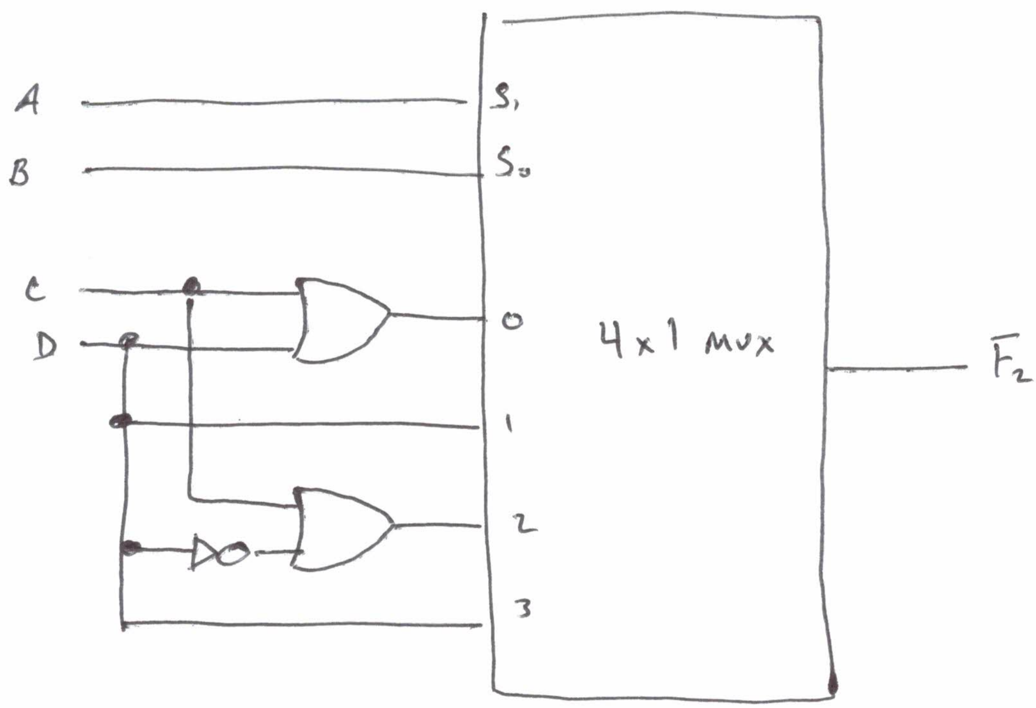


4.35) Implement the following Boolean function with a 4 × 1 multiplexer and external gates.

a)* $F1(A, B, C, D) = \Sigma(1, 3, 4, 11, 12, 13, 14, 15)$

A	B	C	D	F1	
0	0	0	0	0	$F_1 = D$
0	0	0	1	1	
0	0	1	0	0	
0	0	1	1	1	
0	1	0	0	1	$F_1 = C'D'$
0	1	0	1	0	
0	1	1	0	0	
0	1	1	1	0	
1	0	0	0	0	$F_1 = CD$
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	1	
1	1	0	0	1	$F_1 = 1$
1	1	0	1	1	
1	1	1	0	1	
1	1	1	1	1	

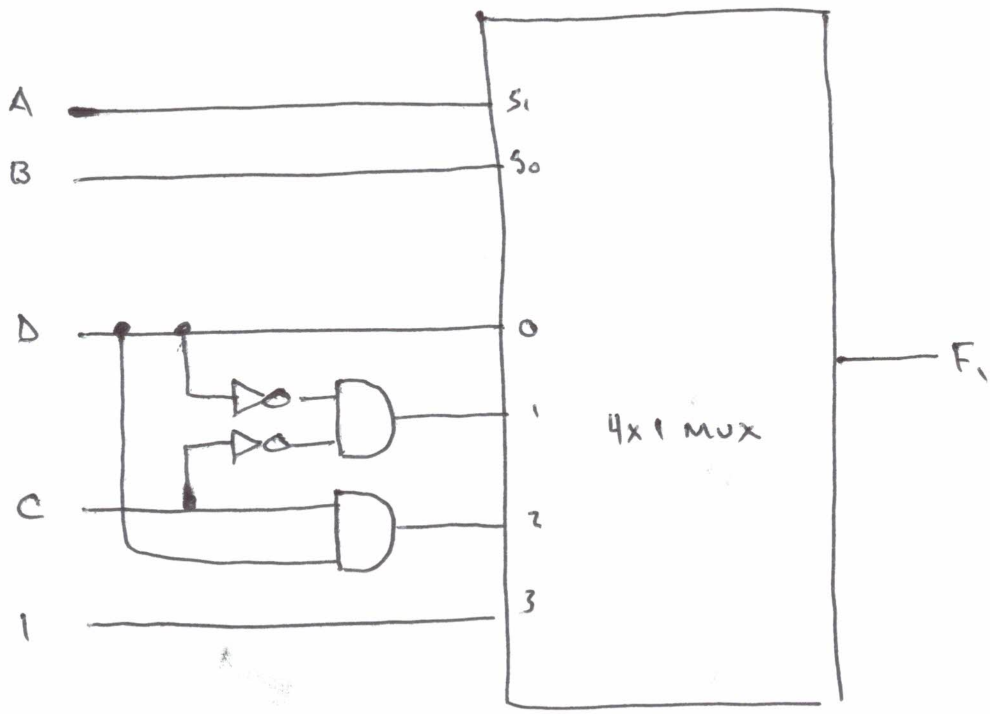
When AB = 00, F1 is = D because F1 = 0 when D = 0 and F1 = 1 when D = 1. The value C doesn't impact the output. When AB = 01 then F1 = C'D' because F1 = 1 when CD = 00 and F1 = 0 when CD = 01, 10, or 11. When AB = 11 then F1 = 1 because F = 1 when CD = 00, 01 or 11.



b) F 2 (A , B , C , D) = Σ (1 , 2 , 5 , 7 , 8 , 10 , 11 , 13 , 15)

<table>
<tr><th>A</th><th>B</th><th>C</th><th>D</th><th>F2</th><th></th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td rowspan="4">$F_2 = C'D + CD'$</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td rowspan="4">$F_2 = D$</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td rowspan="4">$F_2 = D' + C$</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td rowspan="4">$F_2 = D$</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>

When AB = 00 then F2 = C'D + CD' because F2 = 0 when CD = 00 or 11 and F2 = 1 when CD = 01 or 10. When AB = 01 then F2 = D because F2 = 0 whgehn D = 0 and F2 = 1 when D = 1. The value of C doesn't matter. When AB = 10 then F2 = D' + C because F2 = D' when CD = 00, 01, or 10 and F2 = 1 when CD = 10 or 11. When AB = 11 then F2 = D because F2 = 0 when D = 0 and F2 = 1 when D = 1.The value of C doesn't matter here also.


A
B
D
C
1
S1
S0
0
1
2
3
4x1 MUX
F1