EE 113 – Intro to Electrical Engg Practice: 2021-22/I

Problem Sheet – 4: Digital Electronics

Topics: Combinatorial Circuits – K-map minimization, truth tables, Combinatorial circuit design using Multiplexers

Part A– K-map minimization

Student to do these by themselves

Part B – Logic design using Multiplexers

6. Implement $F(x, y, z)$ using an 8-to-1 MUX. Given: $F(x, y, z) = \sum(0, 2, 4, 5, 6)$.

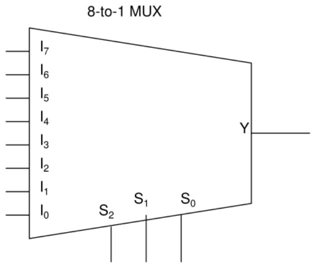
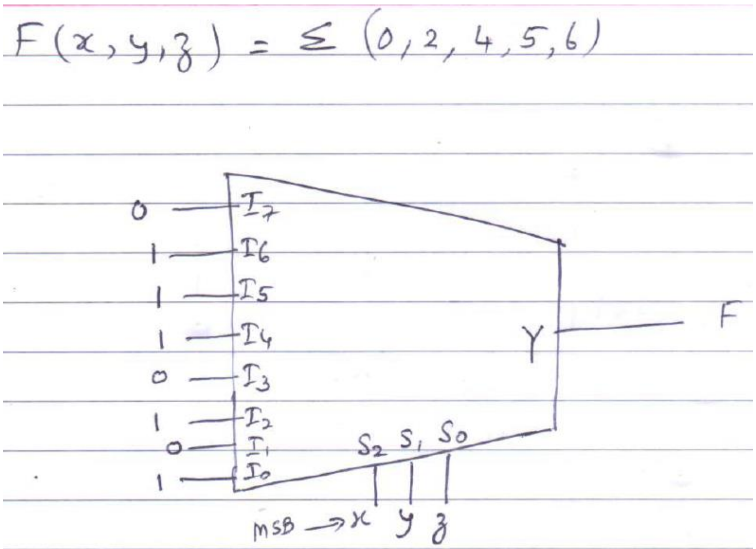


Fig.P6 8-to-1 MUX

Solution



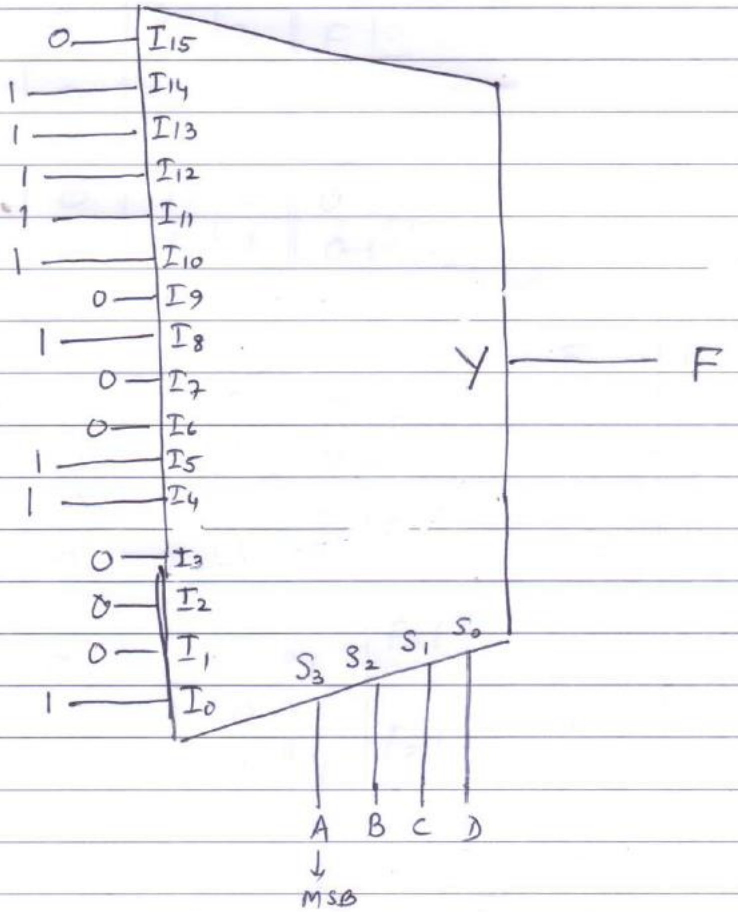
<u>Note:</u> For a 3-variable Boolean expression, if we have an 8-to-1 MUX, then the data inputs I_0 to I_7 are nothing but the truth table entries. But this is an over kill, and in most cases, it will not be an interesting solution. It may be a simple solution, but a costly one. See the next problem and the solution to appreciate the problem with this approach.

Here, in this problem, the total number of pins of the IC required will be: VCC + GND + 8 data inputs + 3 Select inputs = 13 pins. This requires a 14-pin or a 16 pin IC (which are standard packages). However, if two outputs are required as in the case of a full-adder, then it will require two ICs.
A much better solution is to use 4-to-1 MUX for this.

7. Implement F (A, B, C, D) using a 16-to-1 MUX. Given: F (A, B, C, D) = ∑ (0, 4, 5, 8, 10, 11, 12, 13, 14).

Solution

$F(A, B, C, D) = \Sigma(0, 4, 5, 8, 10, 11, 12, 13, 14)$

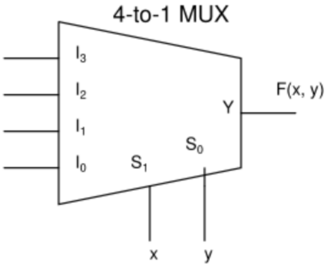


Note: For a 4-variable Boolean expression, if we have a 16-to-1 MUX, then the data inputs I_0 to I_{15} are nothing but the truth table entries. This is an over kill, and in most cases, it will not be an interesting solution. It may be a simple solution, but a costly one.

Here, in this problem, the total number of pins of the IC required will be: VCC + GND + 15 data inputs + 4 Select inputs = 21 pins. This requires a 24-pin IC (that is the standard package after 20 pins). Both costly and occupies more space.

The above solution is seldom used for a 4-variable problem. A much better solution is to use 8-to-1 MUX. See the solution to problem 9.

8. Implement $F(x, y, z)$ using a 4-to-1 MUX. Given: $F(x, y, z) = \sum(0, 2, 4, 5, 6)$.

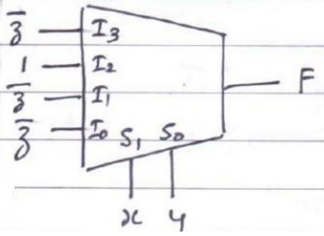


Solution

Addr. lines

$F = \Sigma(0,2,4,5,6)$

x	y	z	F	
0	0	0	1	$F = \bar{z}$
0	0	1	0	
0	1	0	1	$F = \bar{z}$
0	1	1	0	
1	0	0	1	$F = 1$
1	0	1	1	
1	1	0	1	$F = \bar{z}$
1	1	1	0	



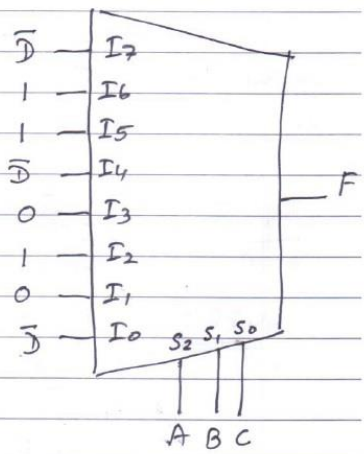
Note: Here, we are able to implement a 3-variable expression with a 4-to-1 MUX. A much more elegant and cheaper solution. The total number of pins: Vcc + GND + 4 data pins + 2 select pins = 8. In fact, we can implement a full-adder with just one 16-pin IC, as the select pins will be shared between the two functions (Carry Out and Sum). The only requirement here is the availability of both z and $\bar{z}$, which are normally available.

9. Implement F (A, B, C, D) using an 8-to-1 MUX. Given: F (A, B, C, D) = ∑ (0, 4, 5, 8, 10, 11, 12, 13, 14).

$F(A,B,C,D) = \sum(0,4,5,8,10,11,12,13,14)$

Addr Lines

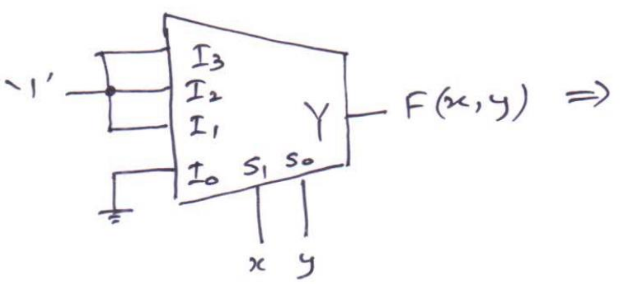
A	B	C	D	F	
0	0	0	0	1	$F=\bar{D}$
0	0	0	1	0	
0	0	1	0	0	$F=0$
0	0	1	1	0	
0	1	0	0	1	$F=1$
0	1	0	1	1	
0	1	1	0	0	$F=0$
0	1	1	1	0	
1	0	0	0	1	$F=\bar{D}$
1	0	0	1	0	
1	0	1	0	1	$F=1$
1	0	1	1	1	
1	1	0	0	1	$F=1$
1	1	0	1	1	
1	1	1	0	1	$F=\bar{D}$
1	1	1	1	0	



Note: Here, we are able to implement a 4-variable expression with an 8-to-1 MUX. A much more elegant and cheaper solution. The total number of pins: Vcc + GND + 8 data pins + 3 select pins = 13. Much simpler and cheaper than the earlier MUX with a 24-pin IC. Again, it is assumed that D and $\bar{D}$ which are normally available.

See problem 11 for the simplicity of implementing a very complex decoder with an 8-to-1 MUX.

10. The implementation of a combinatorial circuit using a 4-to-1 MUX is shown in Fig.P10. Write the corresponding Boolean function F(x, y).



x	y	
0	0	0
0	1	1
1	0	1
1	1	1

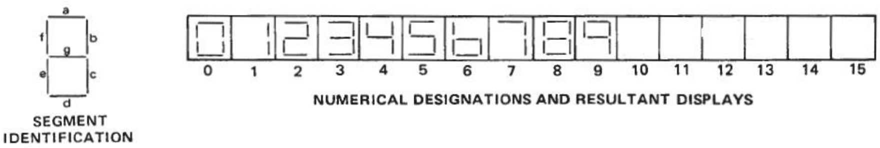
$$F(x,y) = \bar{x}y + x\bar{y} + xy$$

Part C – Decoders

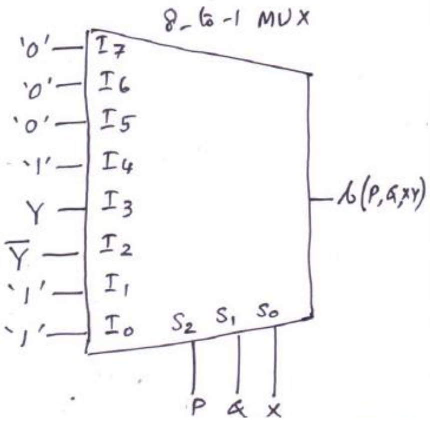
11. In order to display numerals 0 to 9 corresponding to their BCD codes, we need a BCD-to-7 segment decoder circuit with 7 outputs and a 7-segment LED display. Each LED of the 7-segment LED display is connected to the corresponding output (**a** to **g**) of the BCD-to-7-segment decoder. Segment identification and numerical displays of the 7-segment LED display are shown in Fig.5B2. Assume that for an LED segment to light up, the corresponding decoder output should be '1'. When a 4-bit BCD code **(PQXY)** of numerals 0 to 9 is given to the inputs of the decoder, the appropriate outputs will be '1'. As shown in Fig.P11, all the decoder outputs will be '0' for BCD codes **(PQXY): 1010 to 1111**.

    i) Write the truth table for the decoder output **b (P,Q,X,Y)**. Note that **P** is the MSB.

    ii) Implement the above function **b (P,Q,X,Y)** using an 8-to-1 multiplexer. Block diagram of the 8-to-1 MUX to be used for your implementation is shown in Fig. P6.



P	Q	X	Y	b	
0	0	0	0	1	$I_0 = 1$
0	0	0	1	1	
0	0	1	0	1	$I_1 = 1$
0	0	1	1	1	
0	1	0	0	1	$I_2 = \overline{Y}$
0	1	0	1	0	
0	1	1	0	0	$I_3 = Y$
0	1	1	1	1	
1	0	0	0	1	$I_4 = 1$
1	0	0	1	1	
1	0	1	0	0	$I_5 = 0$
1	0	1	1	0	
1	1	0	0	0	$I_6 = 0$
1	1	0	1	0	
1	1	1	0	0	$I_7 = 0$
1	1	1	1	0	



12. The pin diagram of a very popular decoder, viz. 74LS138 3-to-8 decoder is shown below. This IC is commonly used as an address decoder for RAM and ROM ICS (for their Chip Select inputs) in microprocessor applications.

Summary of its working:

<u>Inputs:</u>

Address inputs (Select): A, B, C (Note: C is the MSB and A is the LSB. Hence, when C=1, B=0, A=0, the Y4 output will get selected; similarly is C=0, B=0, A=1, the Y1 output will get selected)

Enable inputs: G2A, G2B and G1 (Note: G2A and G2B are active-low inputs, whereas G1 is an active-high input. For the IC to be enabled, the following need to be satisfied: G2A = G2B = 0, and G1= ‘1’).

<u>Outputs</u>

Decoder Outputs: Y0, Y1,….Y7. All are active-low outputs, meaning when an output gets selected, it becomes low. At a time only one output will be low.

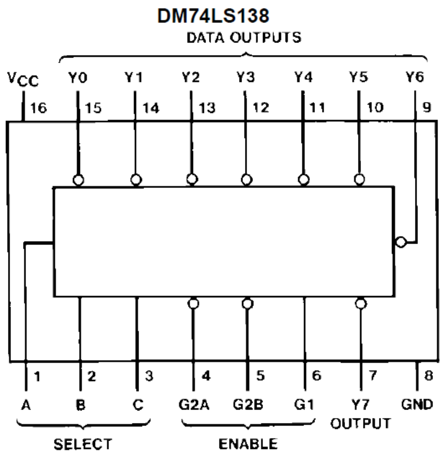
For an output to get selected (i.e. to become low), two conditions must be satisfied:

a) The CBA inputs should give the address of the output.

b) The outputs should be enabled. i.e. the enable inputs should be G2A = G2B = 0, and G1= ‘1’.

For example, if C=0, B=0, A=1, the Y1 output will be at ‘0’ provided, G2A = G2B = 0, and G1= ‘1’.

If any of the enable inputs do not have the above condition, then all the outputs will be at ‘1’.



Design problem statement:

You are given four LEDs: Red, Yellow, Green and Blue.

Red should turn ON if the decimal equivalent of ‘CBA’ input is 2

Yellow should turn ON if the decimal equivalent of ‘CBA’ input is 3

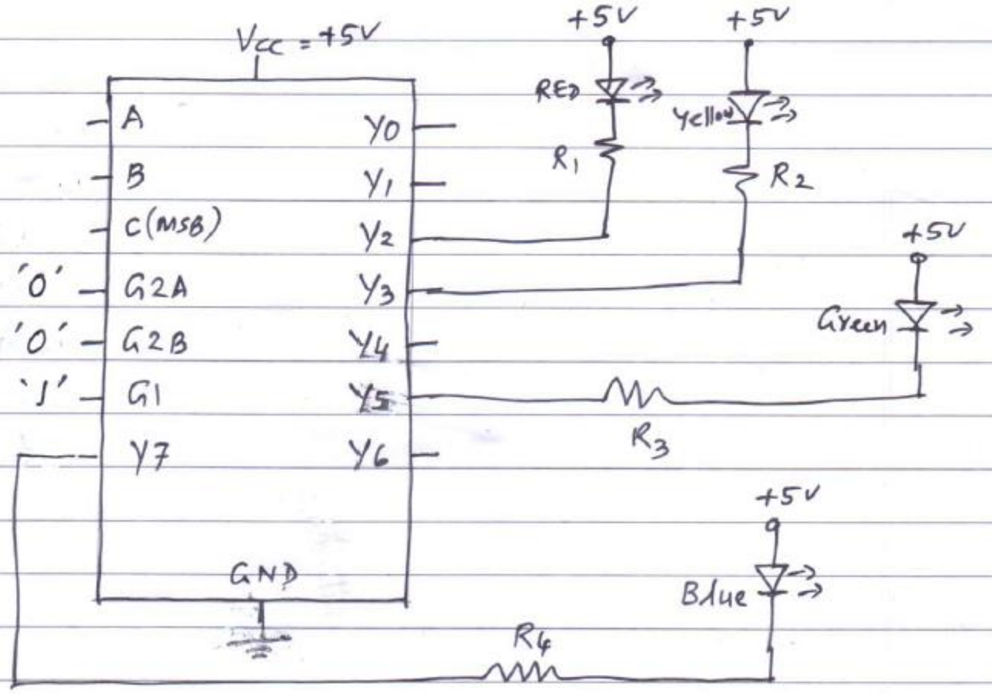
Green should turn ON if the decimal equivalent of ‘CBA’ input is 5

Blue should turn ON if the decimal equivalent of ‘CBA’ input is 7

Assume that G2A = G2B = 0, and G1= ‘1’. Show the circuit connections. Assume that for an LED to turn ON, its current should be around 10 mA. Assume LED forward voltage to be 1.75 volts. You should not use any other IC or gate other than the 74LS138 decider. You may use any resistor values.

Specifications of the Y0 to Y7 outputs: $V_{OL} \leq 0.4$ V, $I_{OL} = 16$ mA. $V_{OH} \geq 2.4$ V, $I_{OH} = 0.4$ mA.

Solution



LEDs will turn ON only when the corresponding CBA inputs are there (i.e. when the o/p are at '0')

Choice of R_1, R_2, R_3, R_4

$$R_1 = \frac{V_{cc} - V_{OL} - V_{LED}}{I_{LED}} = \frac{5 - 0.4 - 1.75}{10\,mA} = \frac{2.85}{10\,mA}$$

$$= 0.285\,k\Omega,\ 285\,\Omega$$

$R_1 = R_2 = R_3 = R_4$

C	B	A	LED
0	1	0	RED
0	1	1	YELLOW
1	0	1	GREEN
1	1	1	BLUE