CS2100 Computer Organisation

Tutorial #8: MSI Components

(Week 10: 22 – 26 March 2021)

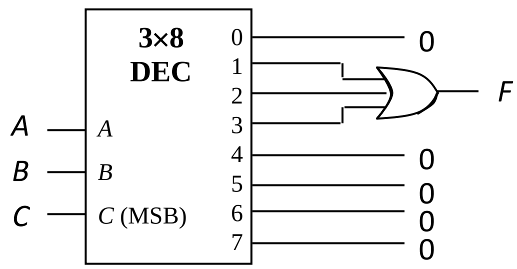
LumiNUS Discussion Questions:

These questions will not be discussed in tutorial. Please discuss on the forum.

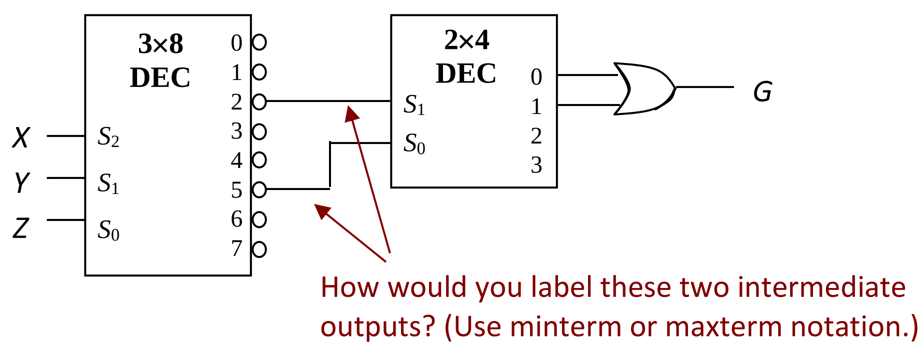
D1. Given this Boolean function:

$$F(A,B,C) = \Sigma\, m(1, 2, 3)$$

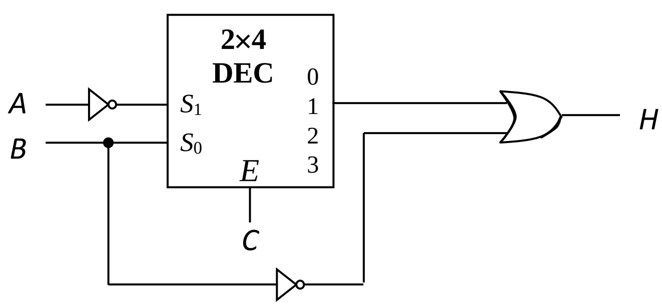
We want to implement this function using a **3×8 decoder with normal outputs** as shown below. Point out the mistakes in the solution below.



D2. Given the following circuit comprising a **3×8 decoder with negated outputs** and a **2×4 decoder with normal outputs**, what is the Boolean function *G*(*X,Y,Z*)?



D3. Given the following circuit comprising a **one-enabled 2×4 decoder with normal outputs**, what is the simplified SOP expression of Boolean function *H(A,B,C)*?



Tutorial Questions:

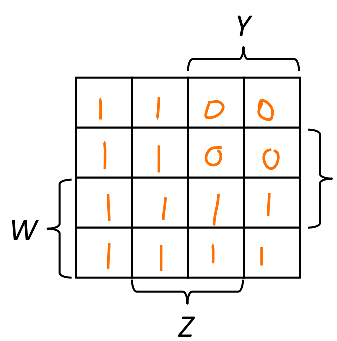
1. Realize the following function with (a) **an 8:1 multiplexer**, and (b) **a 4:1 multiplexer** using the first 2 input variables as the selector inputs.

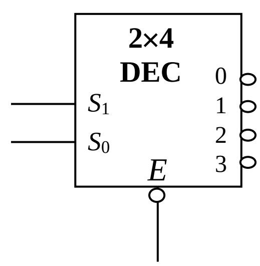
$$\boldsymbol{F(X, Y, Z) = \Pi M(1, 5, 6) \cdot D(4)}$$

You may write complemented variables instead of drawing an inverter to derive it. If you have several choices for your answer, choose the simplest one (constant logic values 0 and 1 are simpler than literals). You may write “x” or “d” for “don’t-care” values.

What if we use the last 2 input variables as the selector inputs instead for the 4:1 multiplexer?

2. Given the following **zero-enabled 2×4 decoder with negated outputs**, how would you implement the Boolean function $J(W,X,Y,Z) = \prod M(2, 3, 6, 7)$ <u>without</u> any additional logic gates?





3. [AY2011/2 Semester 2 Exam question]
You are to design a converter that takes in 4-bit input *ABCD* and generates a 3-bit output *FGH* as shown in Table 1 below.

Input				**Output**		
A	***B***	***C***	***D***	***F***	***G***	***H***
0	0	0	0	0	0	0
1	0	0	0	0	0	1
1	1	0	0	0	1	0
1	1	1	0	0	1	1
1	1	1	1	1	0	0
0	1	1	1	1	0	1
0	0	1	1	1	1	0
0	0	0	1	1	1	1

Table 1

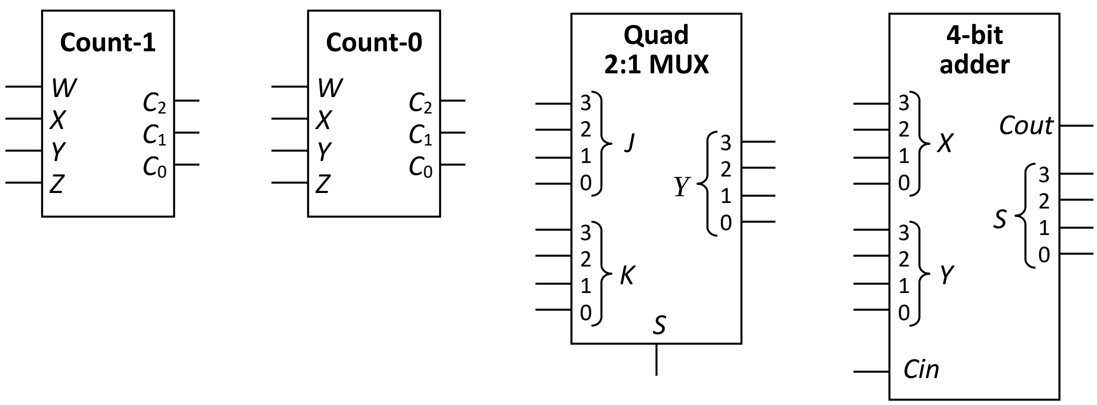
S	$Y_3Y_2Y_1Y_0$
0	$J_3J_2J_1J_0$
1	$K_3K_2K_1K_0$

Table 2

You are given the following components:

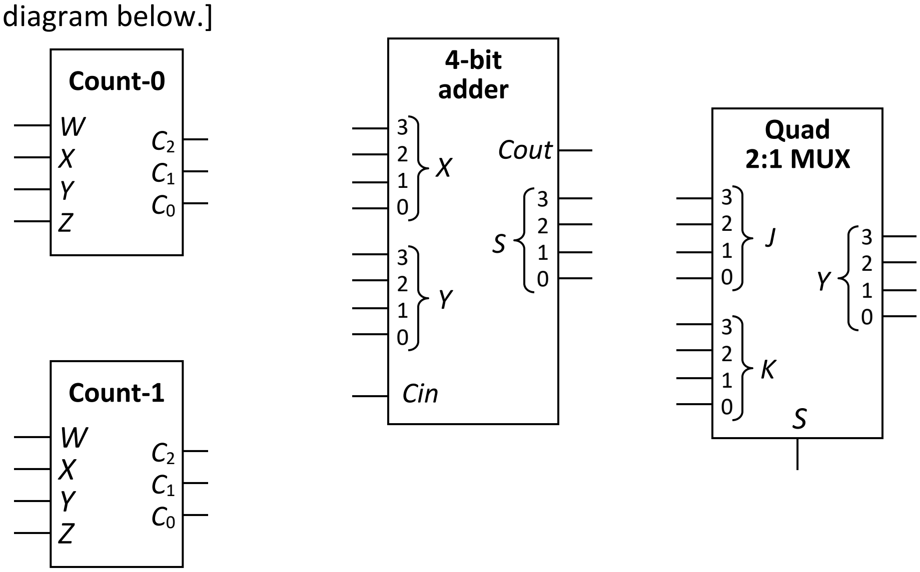
a. A **Count-1** device that takes in a 4-bit input *WXYZ* and generates a 3-bit output $C_2C_1C_0$ which is the number of 1s in the input. For example, if $WXYZ = 0111$, then $C_2C_1C_0 = 011$ (or 3).
b. A **Count-0** device that takes in a 4-bit input *WXYZ* and generates a 3-bit output $C_2C_1C_0$ which is the number of 0s in the input. For example, if $WXYZ = 0111$, then $C_2C_1C_0 = 001$ (or 1).
c. A **quad 2:1 multiplexer** that takes in two 4-bit inputs $J_3J_2J_1J_0$ and $K_3K_2K_1K_0$, and directs one of the inputs to its output $Y_3Y_2Y_1Y_0$ depending on its control signal *S*, as shown in Table 2 above.
d. A **4-bit parallel adder** that takes in two 4-bit unsigned binary numbers and outputs the sum.

The block diagrams of these components are shown below:



Given the above 4 components, you are to employ block-level design to design the converter, <u>without using any additional logic gate or other devices</u>. You may observe that if $A = 1$, then the output *FGH* is simply the number of 1s in the input *ABCD*. You are to make your own observation for the case when $A = 0$.

[Hint (not given in exam): You need only use one of each of the components. Complete the diagram below.]



4. Implement the following Boolean function using the fewest number of **2×4 decoder with 1-enable and normal outputs**, and at most two logic gates.

$$F(a,b,c,d) = \Sigma m\ (0,1,3,4,6,7,8,9,11,12,14,15)$$

(There is a solution with two decoders and one logic gate which is easy to obtain. A more challenging solution uses one decoder and two logic gates. We will discuss the former and leave the latter as an exercise for your own attempt.)