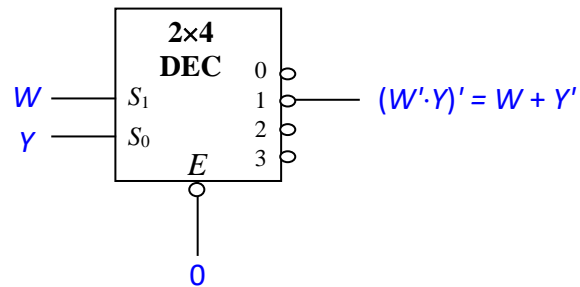
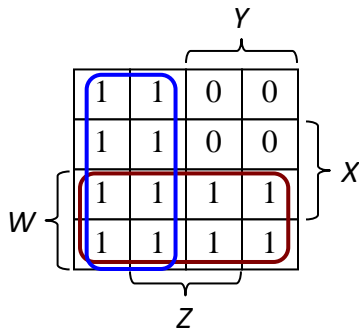


CS2100 Computer Organisation
Tutorial #8: MSI Components
 (Week 10: 22 – 26 March 2021)
Answers to Selected Questions

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

Answer: $J(W,X,Y,Z) = \prod M(2, 3, 6, 7) = W + Y'$ (from K-map)

The expression consists of only two variables. A 2×4 decoder is sufficient to implement such an expression on two variables since $W + Y'$ is maxterm $M1$ of $J(W,Y)$. Alternative solution possible (for example, swapping the inputs W and Y and selecting output number 2).



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

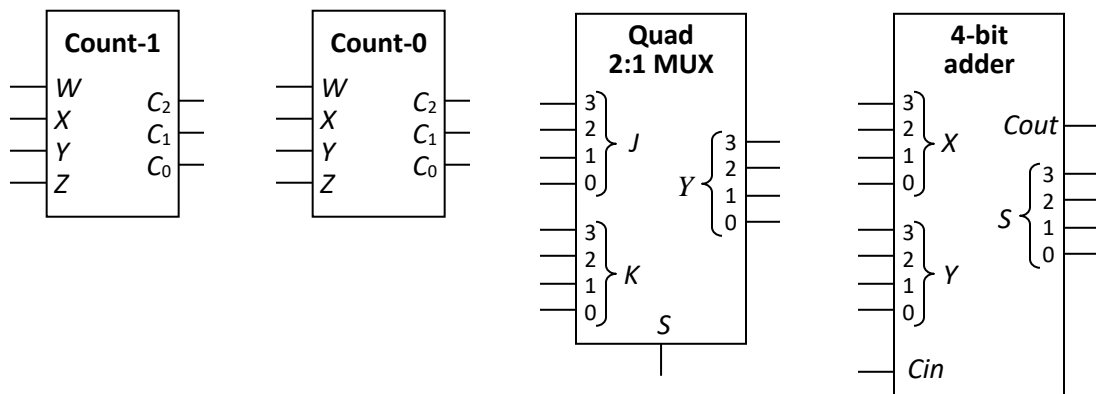
S	$Y_3Y_2Y_1Y_0$
0	$J_3J_2J_1J_0$
1	$K_3K_2K_1K_0$

Table 2

You are given the following components:

- A **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).
- 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).
- 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.
- 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, without using any additional logic gate or other devices. 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 $A = 0$.

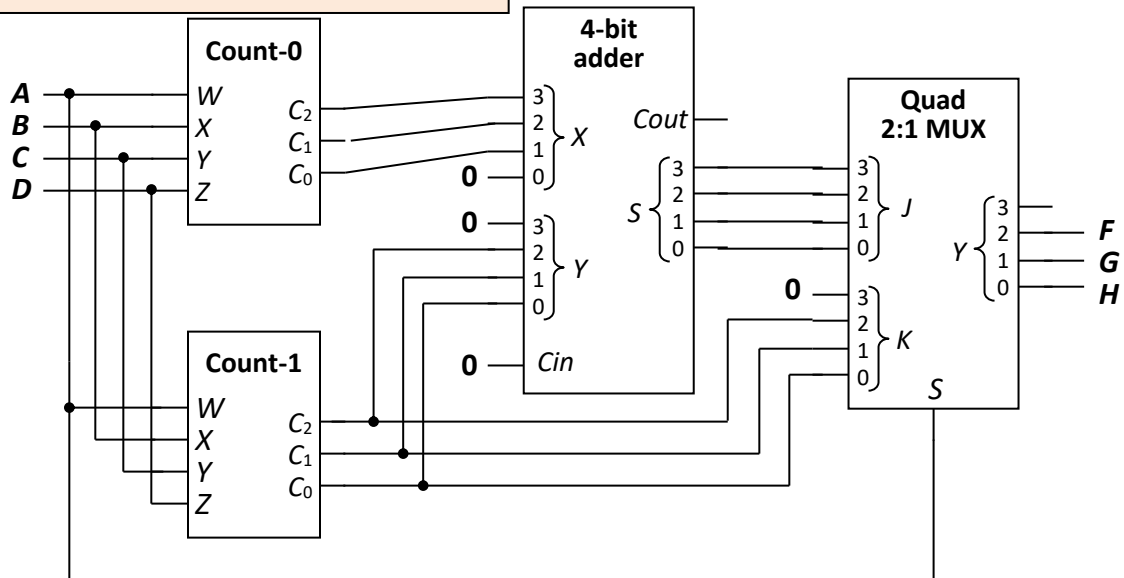
[Hint (not given in exam): You need only use one of each of the components.
Complete the diagram below.]

To tutors:

To save time, you may want to project the incomplete diagram on the whiteboard and get student to complete it on the whiteboard.

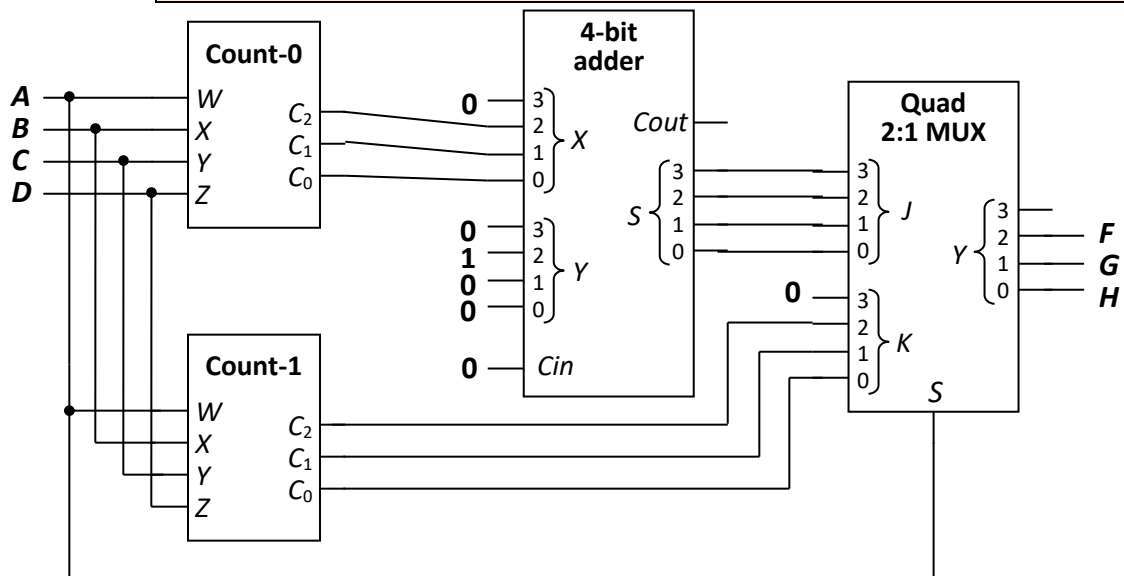
Key ideas:

1. If $A = 1$ (or $D = 0$), count #1s in $ABCD$.
2. If $A = 0$ (or $D = 1$), either
 - a. $\#1s + 2 \times \#0s$; or
 - b. $4 + \#0s$



OR

Tutors: Check that all the inputs of the components are connected to some values, not left unconnected. Tell students that in the exam, the incomplete diagram will **not** be given!



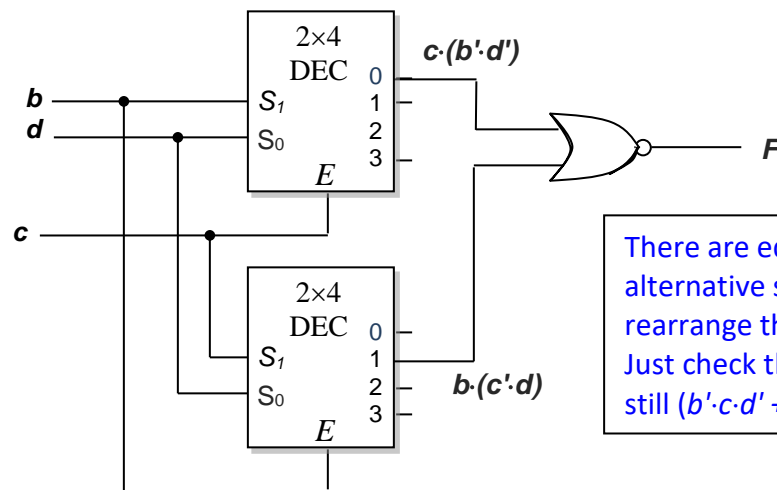
4. Implement the following Boolean function using the fewest number of **2×4 decoder** with **1-enable and normal outputs**, and at most two logic gates.

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

(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.)

Answer:

It is easier to think about implementing F' (which is $\sum m(2,5,10,13)$ or $b' \cdot c \cdot d' + b \cdot c' \cdot d$), and then adding an inverter to invert it back to F .



There are equivalent alternative solutions that rearrange the inputs b , c and d . Just check that the output is still $(b' \cdot c \cdot d' + b \cdot c' \cdot d)'$.