Lab3Fa2019-Petrie

Wednesday, November 18, 2020 12:54 PM

Lab3Fa2019
-Petrie

CDA3201 x Intro to Logic Design x **Lab Assignment** **3**

Name: **Grade:**

3) [Petrie-100] Consider the following two functions:

$$X = \bar{A}BCD + \bar{A}B\bar{C}D + A\bar{B}\bar{C}D + BCD$$

$$Y = A\bar{B}CD + ABC\bar{} + ACD + ABD$$

3.1) [Petrie – 10] Write a Boolean Equation as a Sum of Minterms. There are four methods to do this: (1) completeTruth Table for X and Y, write the Sum of Minterm the row #s where each product =1; r (2) find minterms associated with each product and list in Sum of Minterm equation for X and for Y. Do both methods:

Method 1: Complete Truth Table to find Sum of Minterms

Product Minterm#	A B C D	A'BCD	A'BC'D	AB'C'D	BCD	X
	0000	0	0	0	0	0
	0001	0	0	0	0	0
	0010	0	0	0	0	0
	0011	0	0	0	0	0
	0100	0	0	0	0	0
	0101	0	0	0	0	0
	0110	0	0	0	0	0
A'BCD 7	0111	1	0	0	0	1
	1000	0	0	0	0	0
	1001	0	0	0	0	0
	1010	0	0	0	0	0
	1011	0	0	0	0	0
	1100	0	0	0	0	0
	1101	0	0	0	1	1
	1110	0	0	0	1	1
	1111	0	0	0	1	1

X = ∑m ( )

Method 2: Reverse engineer the minterms number(s) associated with each term in Sum of Minterms for X and for Y

A' B C D → minterm(s):
0 1 1 1 = m7

A' B C' D → minterm(s):
0101 = m5

A B' C' D → minterm(s):
1001 = m9

B C D → minterm(s):
111 : m7, m15

X = ∑m (5, 7, 9, 15)

Product Minterm#	A B C D	AB'CD	ABC'	ACD	ABD	Y
	0000	0	0	0	0	0
	0001	0	0	0	0	0
	0010	0	0	0	0	0
	0011	0	0	0	0	0
	0100	0	0	0	0	0
	0101	0	0	0	0	0
	0110	0	0	0	0	0
	0111	0	0	0	0	0
	1000	0	0	0	0	0
	1001	0	0	0	0	0
	1010	0	0	0	0	0
	1011	1	0	0	0	1
	1100	0	1	0	0	1
	1101	0	1	0	1	1
	1110	0	0	0	0	0
	1111	0	0	0	1	1

Y = ∑m ( )

A B' C D → minterm(s):
1011 = m11

A B C' → minterm(s):
110 : m12, 13

A C D → minterm(s):
1 11 : m11, 15

A B D → minterm(s):
11 1 : m13, 15

Y = ∑m (11, 12, 13, 15)

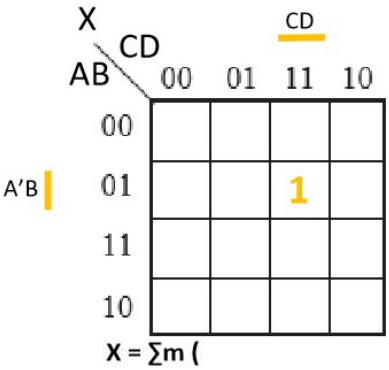
Name: Grade:

$$X = \bar{A}\,B\,C\,D + \bar{A}\,B\,\bar{C}\,D + A\,\bar{B}\,\bar{C}\,D + B\,C\,D$$

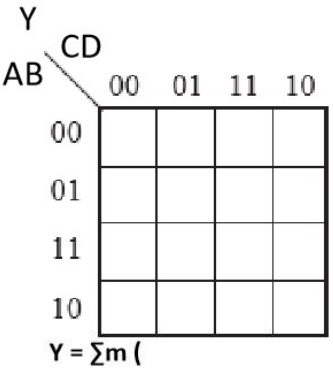
$$Y = A\,\bar{B}\,C\,D + A\,B\,\bar{C} + A\,C\,D + A\,B\,D$$

3.2) [Petrie 10] Two additional methods use Karnaugh Maps (K Maps).

Method 3: Reverse-engineer the cells each product covers using the K-maps. First figure out which row(s) and which column(s) are covered by the product, putting 1's in the intersection. Underline each term (Product) in X in a different color and mark in the X K-Map in that color the 1's each term represents. Number each cell and write down as Sum of Minterms. Repeat for Y.

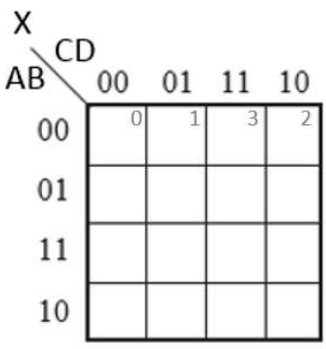


X = ∑m ( )

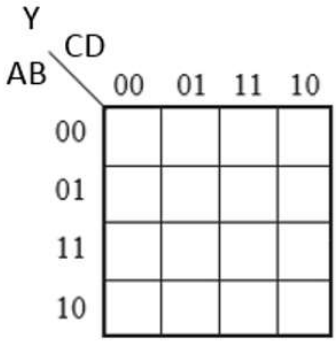


Y = ∑m ( )

Method 4: Label each cell in the following K-Maps with the Minterm # of the cell in the upper right corner, then, using the Truth Table in previous page, put the 1's and 0's in as indicated, write Sum of Minterms equations for X, Y.



X = ∑m ( )



Y = ∑m ( )

Check all methods agree, write the Sum of Minterms for X and Y below

X = f (A, B, C, D) = ∑m (_____)

Y = f (A, B, C, D) = ∑m (_____)

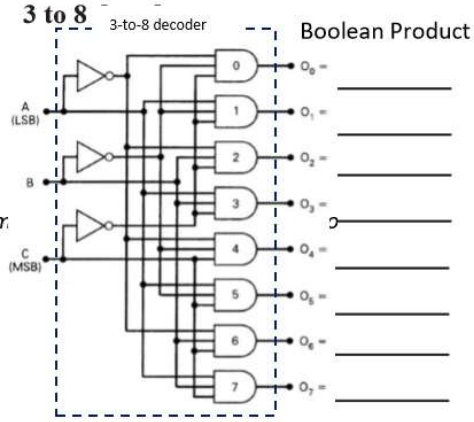
Name: **Grade:**

3.2) [10] In this sections we will be using **DECODERS to simplify building circuits**. A decoder is a circuit or chip that produces all the possible canonical products (provides output line for each midterm number). **Maximum of one of the 2^n output lines become active**, corresponding to the decimal equivalent of the n-bit binary number input, all other outputs remain inactivate. If input 10 for A_1A_0., line D_2 =1, all others equal 0.

[1] Give the Boolean Expression for the Truth Table of the **2 to 4 decoder:**

Minterm #	A_1	A_0	D_3	D_2	D_1	D_0	Boolean Equations
0	0	0	0	0	0	1	D_0 = _____
1	0	1	0	0	1	0	D_1 = _____
2	1	0	0	1	0	0	D_2 = _____
3	1	1	1	0	0	0	D_3 = _____

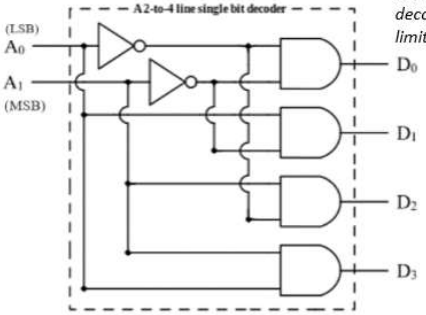
[1] Give the Boolean Product for each output **3 to 8**



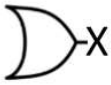
Since any Boolean function can be expressed as a sum of minterm function with a decoder and external OR gate(s)

[4] Complete circuit for **X** = $\overline{A_1}\overline{A_0}$ + $A_1\overline{A_0}$ using the 74HC4075

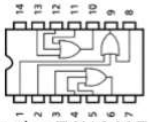
2 to 4 decoder circuit:



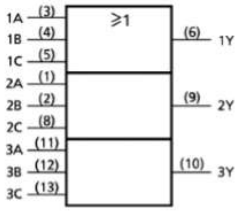
Complete the circuit for Y using the 3-to-8 decoder and an OR gate if there were no limits on the number of inputs in an OR gate:



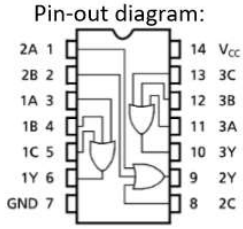
The maximum number of inputs for an OR gate is the 74HC4075 Triple 3-input OR, explained at right. Show how you would connect the decoder to get X



74HC4075
Triple 3-input OR gate
Functional diagram:

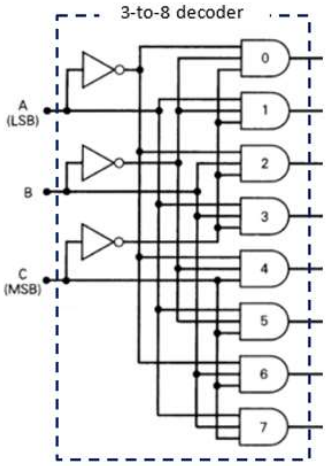


74HC4075
Pin-out diagram:



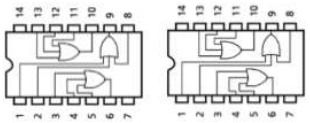
[4] Complete the circuit for **Y = f(C,B,A)** = $\overline{B}$ A + B $\overline{A}$ using the 74HC4075

3-to-8 decoder circuit:



Complete the circuit for Y using the 3-to-8 decoder and an OR gate if there were no limits on the number of inputs in an OR gate:

Show how you would connect the decoder to the two 74HC4075 to get Y:

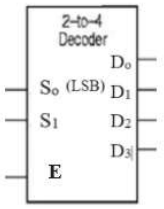


Name: **Grade:**

3. 3) [Petrie 5] Decoder Expansion. You can build bigger decoders as shown on the right, but you need many ANDs. To facilitate constructing with decoders, the *Enable* input is used.

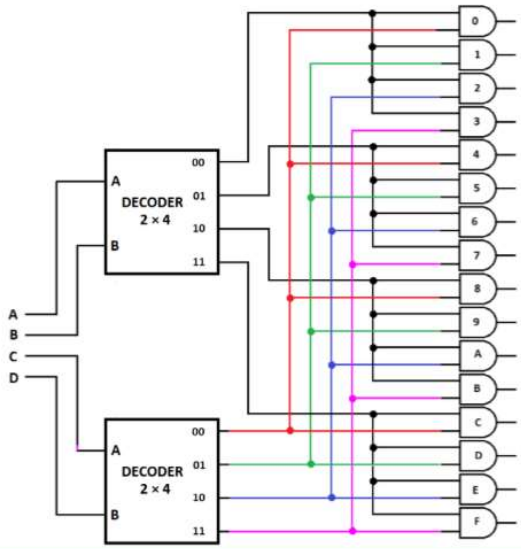
Suppose there was a 2-to-4 decoder chip with:

- *Inputs*: **3 input lines**
  - **Selection** input: $S_1 S_0$ (S_0 is the least significant bit, LSB), selects which output line will be "*active*"
  - **Enable** input: **E**
    - When E = 1, the chip is "*enabled*" making the appropriate output line "*active*", in this case = 1.
    - When E = 0, the chip is "*disabled*"; all output lines are "*inactive*", all = 0.



S_1	S_0	E	D_3	D_2	D_1	D_0
X	X	0	0	0	0	0
0	0	1	1	0	0	0
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	1

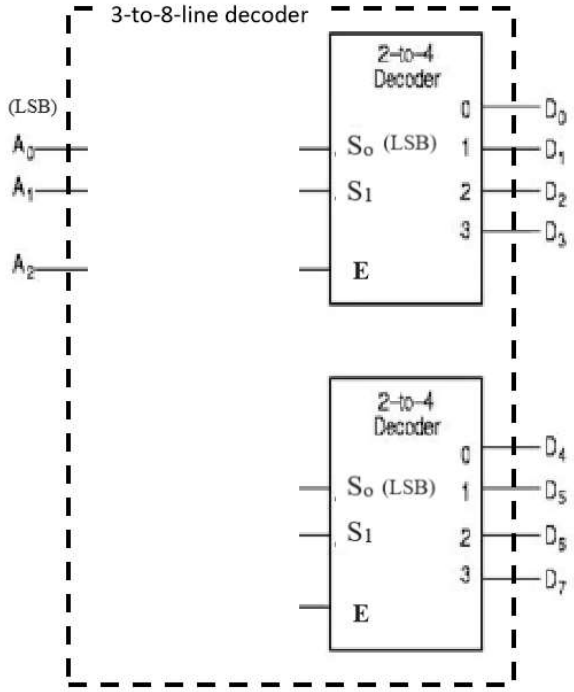
4-to-16 decoder built from two 2-to-4 decoders:



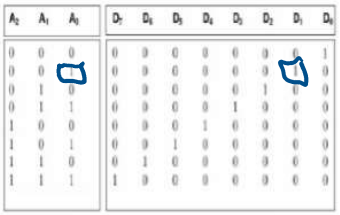
- *Outputs*: **4 output lines**: Data lines $D_3 D_2 D_1 D_0$.

Study the truth table and the two possible implementations for the circuit for the chip to find how it works.

3-to-8-line decoder

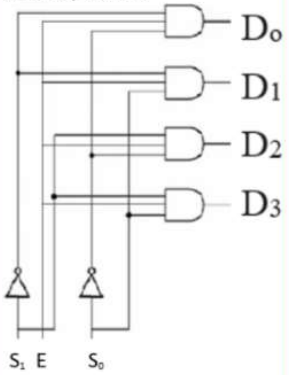


Build a 3-to-8 decoder on the left using two of these 2-to-4-line decoder chips as shown on left (You may need to add gates) and connect the $A_2A_1A_0$, where A_0 is LSB, to the components inside the rectangle. The outputs behave as in the truth table below:

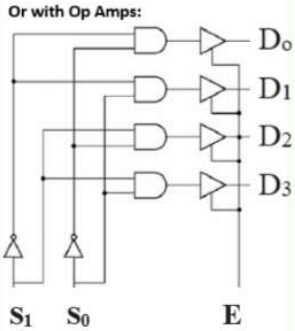


Two ways to build a 2-to-4 decoder with Enable:

With 3-input ANDs:

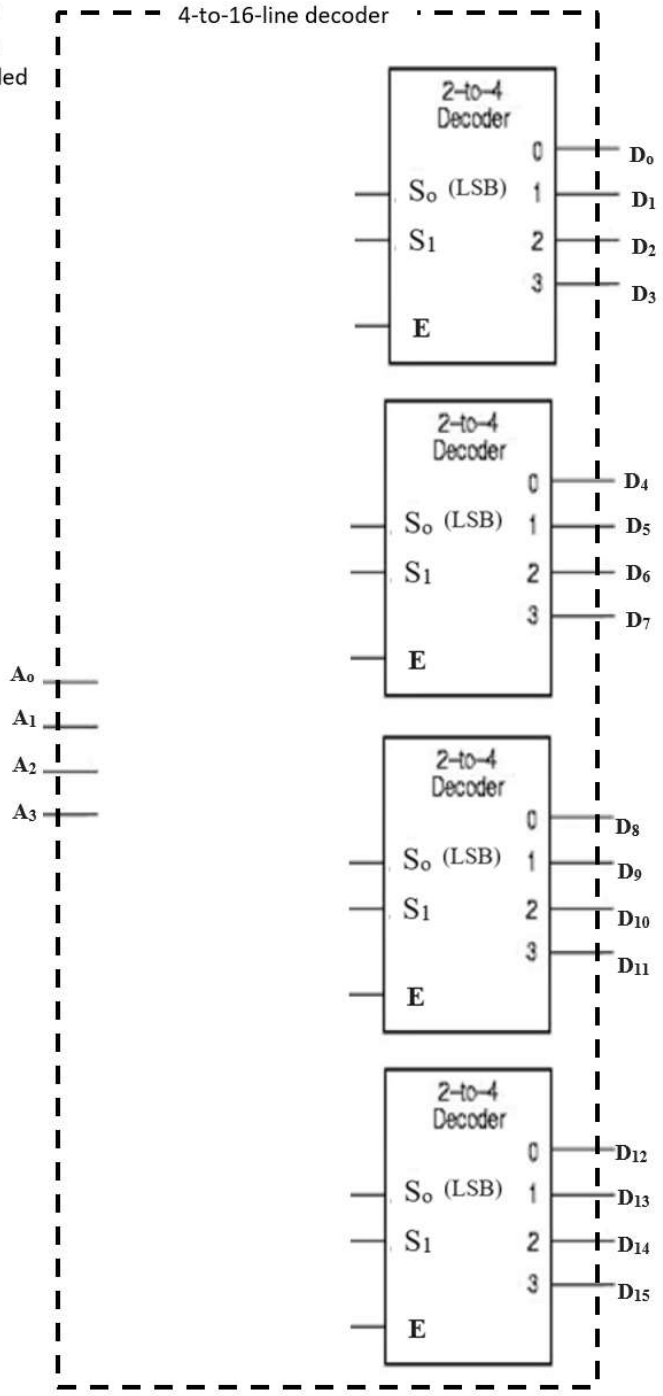


Or with Op Amps:



Name: Grade:

3.4 **[Petrie 10] Decoder Expansion.** Complete the 4-to-16 line decoder at right drawing in the dashed rectangle the components needed to complete the decoder.



Name: Grade:

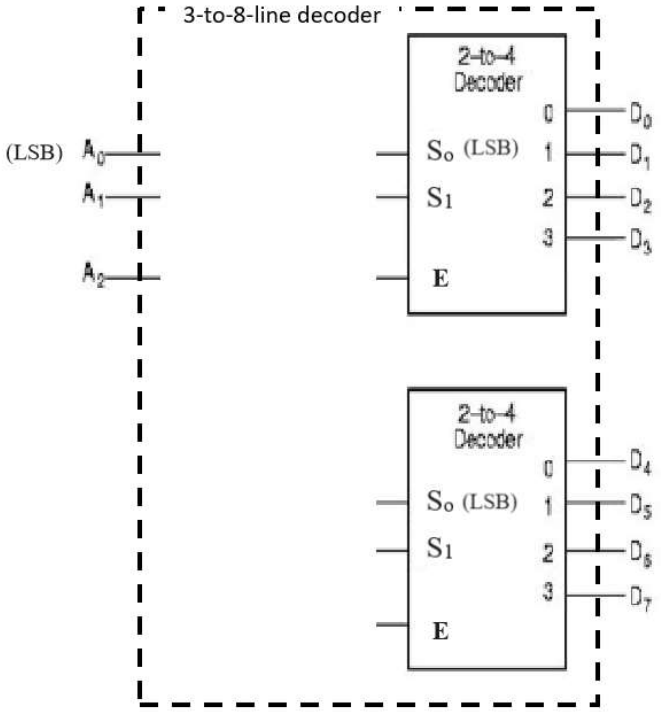
3.3 **Decoder Expansion.** Suppose there was a 2-to-4 decoder chip with:

- *Inputs*: **3 input lines**
  - **Selection** input: $S_1 S_0$ (S_0 is the least significant bit, LSB), selects which output line will be "*active*"
  - **Enable** input: **E**
    - When E = 1, the chip is "*enabled*" making the appropriate output line "*active*", in this case = 1.
    - When E = 0, the chip is "*disabled*"; all output lines are "*inactive*", all outputs = 0, no matter what the input equals.
- *Outputs*: **4 output lines**: **Data** output: $D_3 D_2 D_1 D_0$.

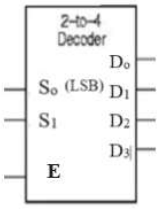
Study the truth table and the two possible implementations for the circuit for the chip to find how it works.

[5] Using two of these 2-to-4-line decoder chips add gates and connections inside the dashed rectangle to build a 3-to-8-line decoder. The input is $A_2A_1A_0$, where A_0 is LSB, the outputs behave as in the truth table at right:

A_2	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0



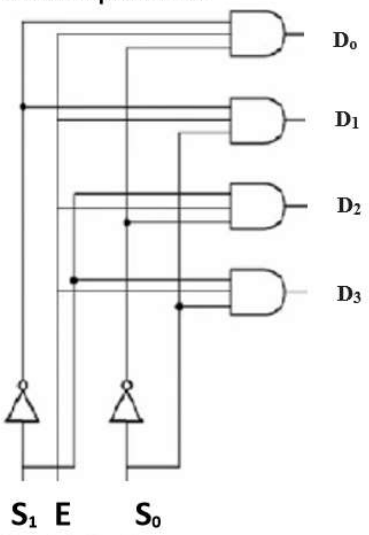
Given a **2-to-4-line decoder chip**:



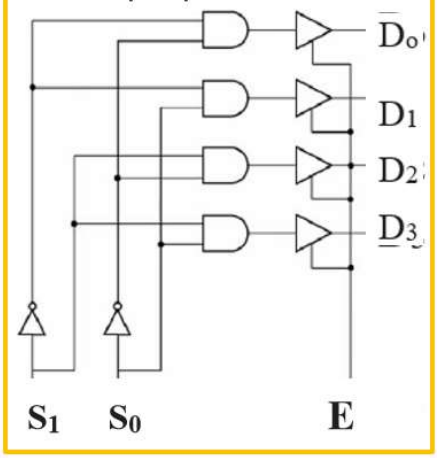
S_1	S_0	E	D_3	D_2	D_1	D_0
X	X	0	0	0	0	0
0	0	1	1	0	0	0
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	1

Two ways to build a 2-to-4 decoder with Enable:

With 3-input ANDs:



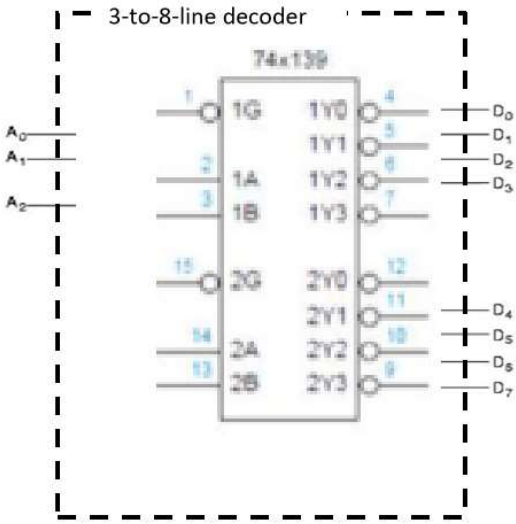
Or with Op Amps:

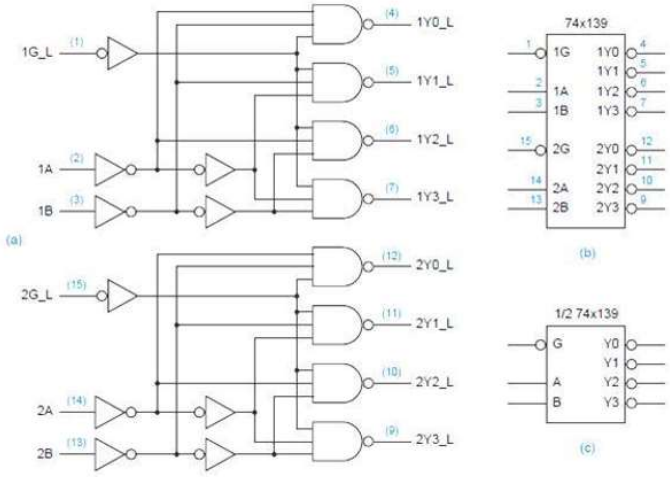


Name: **Grade:**

3.4 [Petrie-5] The 74139 is a chip that contains two 2-4 decoders. Look at the truth table and how it was built. Then show how you can connect 3 inputs A0 (lsb), A1, A2 to the 74x139 to create a 3-8 decoder, add whatever logic gates you need inside the dashed rectangle. Note that the resulting decoder is active low outputs, compared to the one in the previous exercise which was active high. Which logic chip would you use to connect the minterms for each of the two decoders you designed?

E	*A*	*B*	D_0	D_1	D_2	D_3
1	*X*	*X*	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

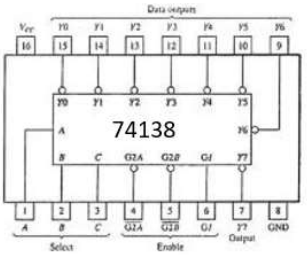


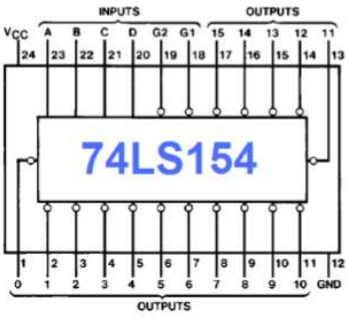


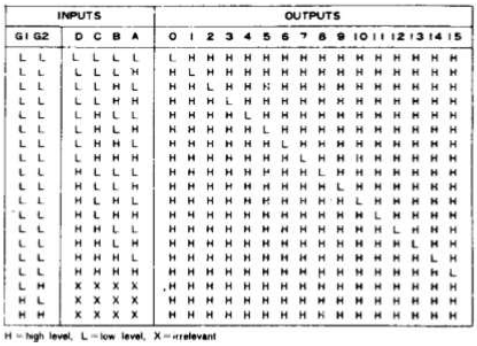
CDA3201 x Intro to Logic Design x **Lab Assignment** **3**

Name: **Grade:**

3.3) [TA-4, Petrie-30] The Functional Diagrams of the 74HC154 4 to 16 Decoder and of the 74138 3-8 Decoder are given, along with the PinOut Diagrams and the Truth Table for each. Plan the circuits for X and Y given in page 1. First by using the 74HC154 4-16 decoder and 1 7420 4-input NAND. You do not have a 4-16 decoder in your kit, plan the same circuits for X and Y using exactly two 3-to-8 Decoders, (2 of 74138 chips) and two 4-input NAND gates (1 - 7420 chip). No other gates are allowed.

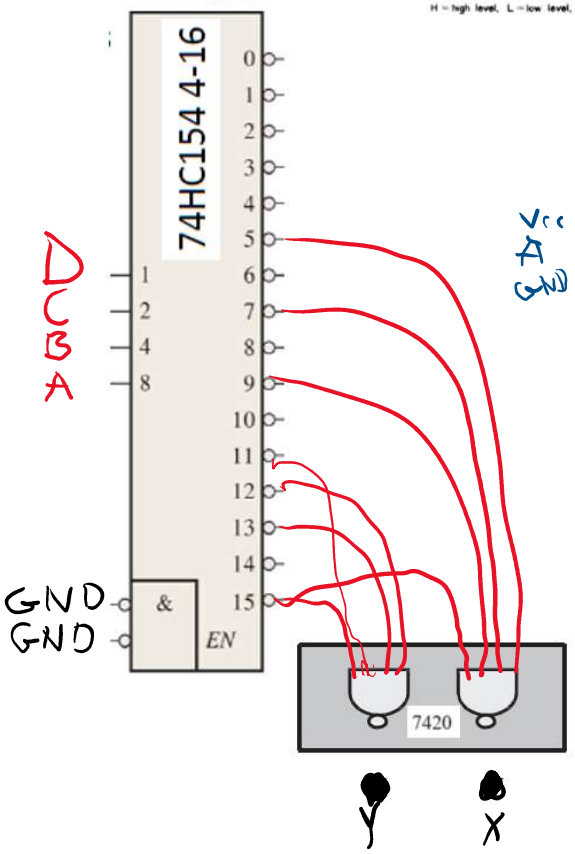


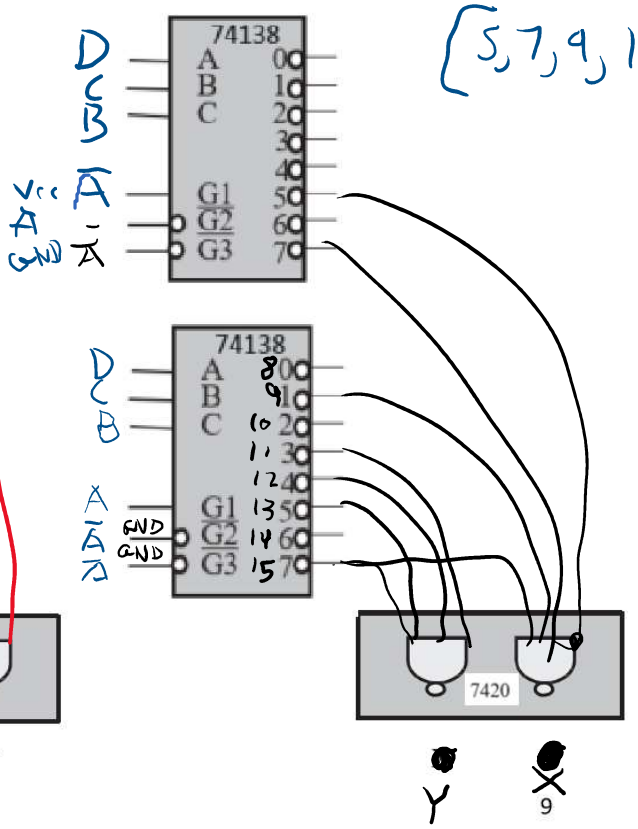




Inputs						Output							
Enable			Select										
G2A	G2B	G1	C	B	A	0	1	2	3	4	5	6	7
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

[5, 7, 9, 11, 12, 13, 15]

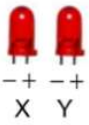


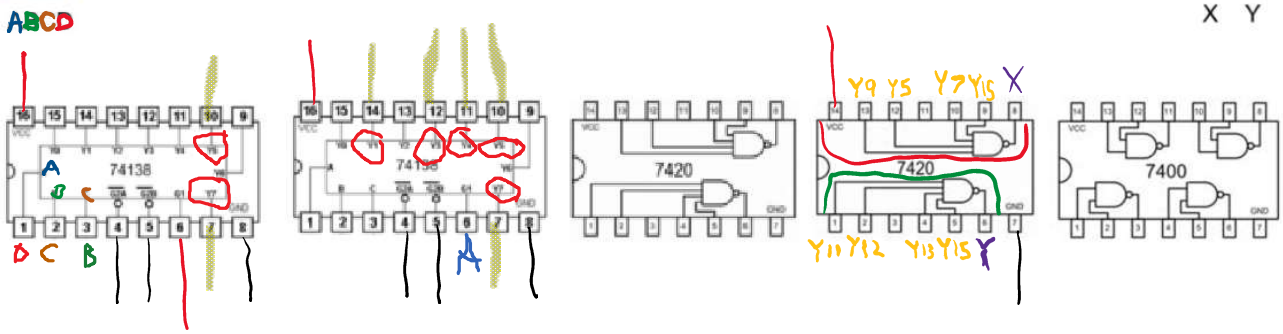


CDA3201 x Intro to Logic Design x	Lab Assignment	3
Name:	Grade:	

3.d) [4] Verify the circuit design/behavior for X and Y by implementing the 74138 circuit using Quartus and checking against the truth tables for X and Y on page 1. Do this before you actually build the 74138 circuit on the breadboard.

3.c) [10] Plan and build the above circuit using the specified logic chips on your breadboard and then connect to 4 logic switches as inputs A, B, C, D and 2 LEDs as outputs X and Y. You need to test all the 16 different input combinations of the inputs





3.3) [10] Build a "B" LED driver for the 7-segment display.

Using the two decoders on left add NAND gates to also build a driver circuit to control "b" LED.

1. Locate b LED resistor, disconnect the end connected to "B" in the 7-segment display driver and connect this end instead to the empty column at the end of the 7-segment display. This will be where you connect your output "b".
2. Connect ABCD switch inputs to WXYZ inputs of the driver (W is least significant bit)
3. Use the outputs of the 4-16 decoder you built get the minterms that turn on the "b" LED and group using the NAND chips below.
4. Connect the output "b" to the empty row where you connected the "b" resistor
5. Test running through the inputs 0 to 9

You do not have to do Quartus for this part of the lab.