Saaif Ahmed
661925946

10/7/19
ahmeds7

CdC HW #5

6.21:

What is wrong with cirait is that EN_L is tied to both enable pins of the 74x139. This is a problem because both will either be on or off instead of one off and one on. This could result in floating values or even shorts. To fix this, add a NOT gate before EN_L reaches one of the 74x139. This will ensure that one is on and one is off.

6.50: Truth Table

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	[illegible]	A	B	C	D
1	0	0	0	0	0	0	0	0	0		0	0	0	0
0	1	0	0	0	0	0	0	0	0		0	0	0	1
0	0	1	0	0	0	0	0	0	0		0	0	1	0
0	0	0	1	0	0	0	0	0	0		0	0	1	1
0	0	0	0	1	0	0	0	0	0		0	1	0	0
0	0	0	0	0	1	0	0	0	0		0	1	0	1
0	0	0	0	0	0	1	0	0	0		0	1	1	0
0	0	0	0	0	0	0	1	0	0		0	1	1	1
0	0	0	0	0	0	0	0	1	0		1	0	0	0
0	0	0	0	0	0	0	0	0	1		1	0	0	1

other values don't matter for BCD

6.50 (continued):

I will input this truth table into Espresso to simplify it.

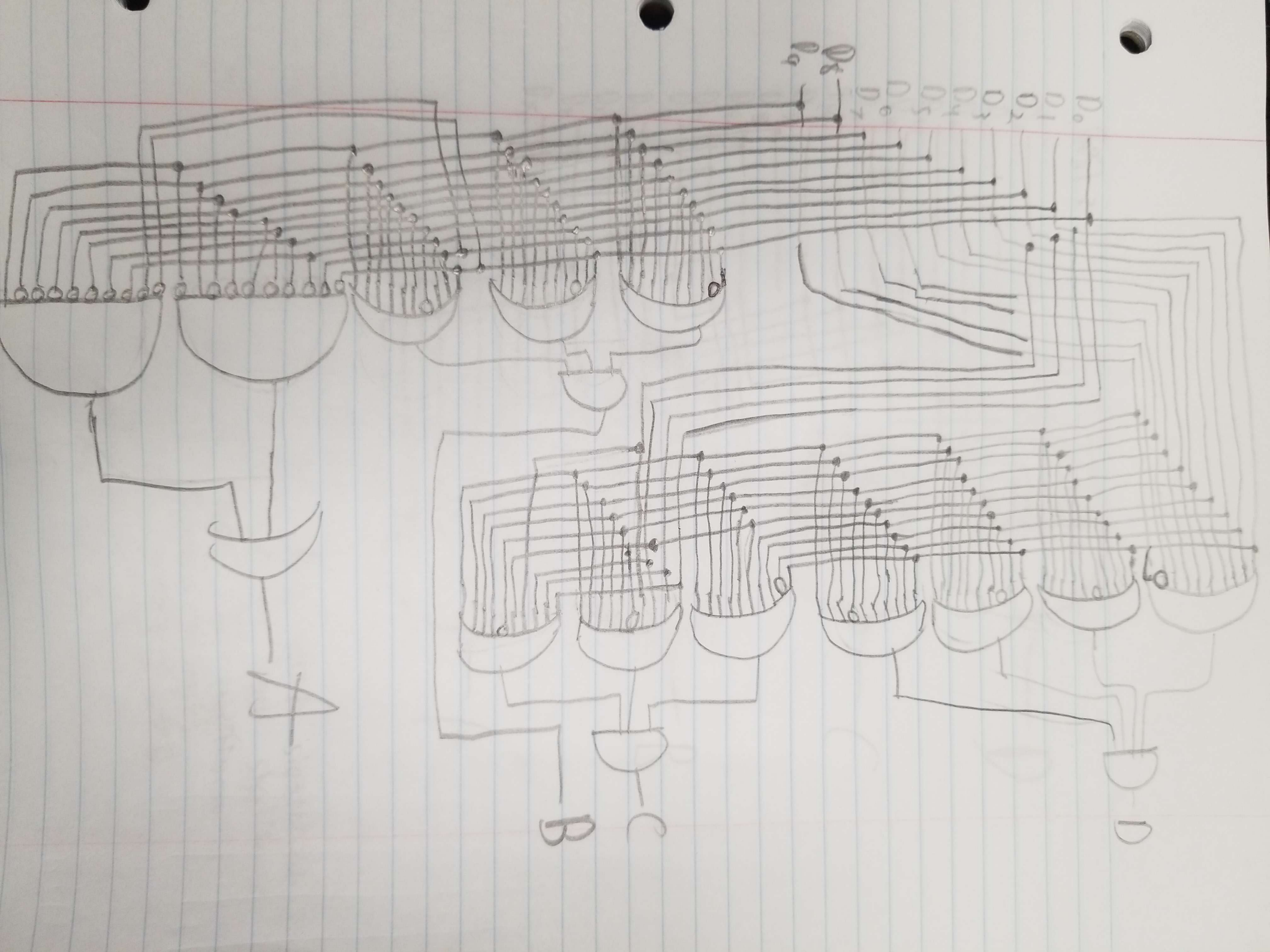
$$A = \bar{D}_0 \bar{D}_1 \bar{D}_2 \bar{D}_3 \bar{D}_4 \bar{D}_5 \bar{D}_6 \bar{D}_7 D_8 \bar{D}_9 + \bar{D}_0 \bar{D}_1 \bar{D}_2 \bar{D}_3 \bar{D}_4 \bar{D}_5 \bar{D}_6 \bar{D}_7 \bar{D}_8 D_9$$

$$B = (D_0 + \bar{D}_1 + D_2 + D_3 + D_4 + D_5 + D_6 + D_7 + D_8 + D_9) \cdot (D_0 + D_1 + \bar{D}_2 + D_3 + D_4 + D_5 + D_6 + D_7 + D_8 + D_9) \cdot (D_0 + D_1 + D_2 + \bar{D}_3 + D_4 + D_5 + D_6 + D_7 + D_8 + D_9)$$

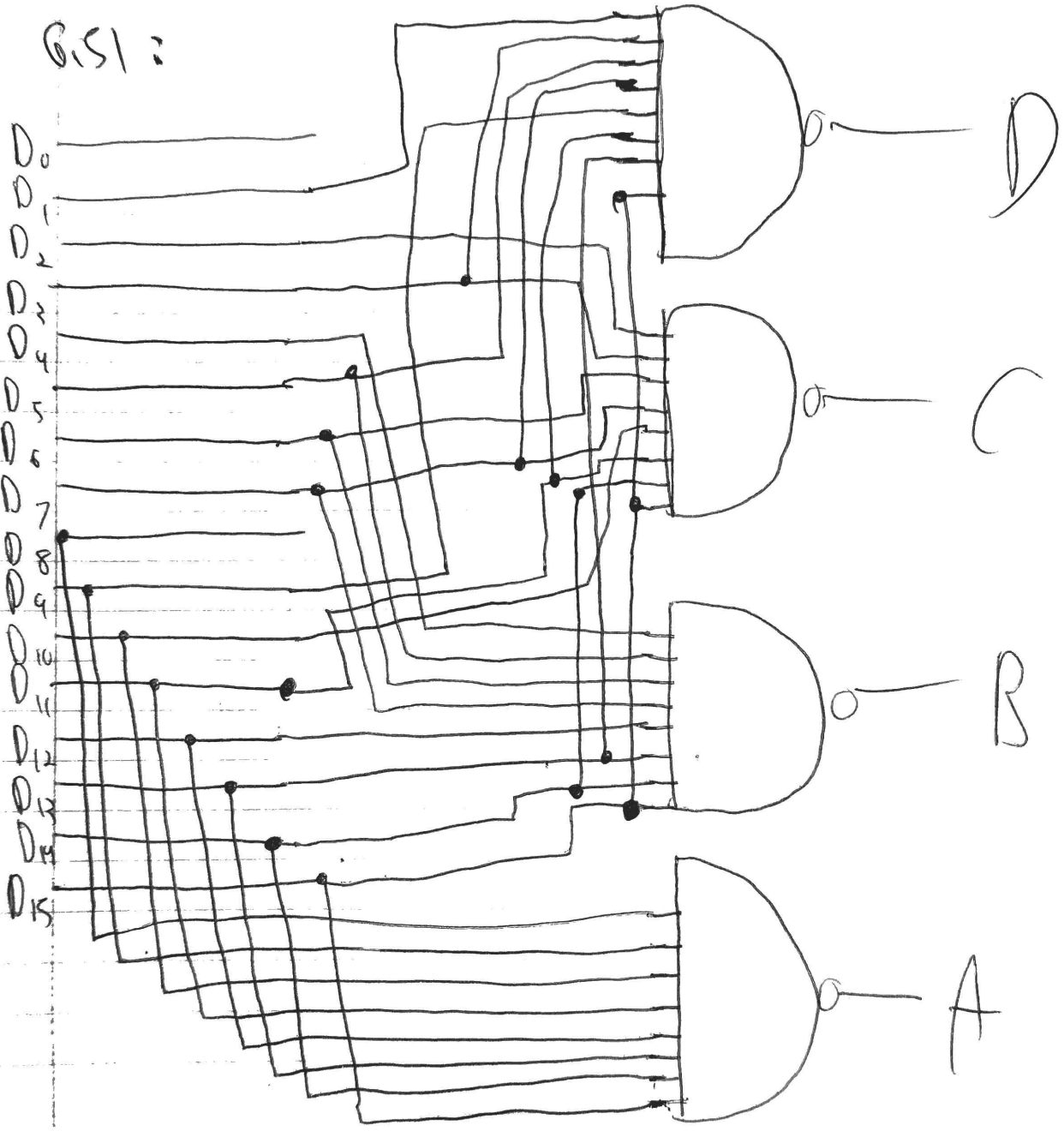
$$C = (D_0 + \bar{D}_1 + D_2 + D_3 + D_4 + D_5 + D_6 + D_7 + D_8 + D_9) \cdot (D_0 + D_1 + D_2 + D_3 + \bar{D}_4 + D_5 + D_6 + D_7 + D_8 + D_9) \cdot (D_0 + D_1 + D_2 + D_3 + D_4 + \bar{D}_5 + D_6 + D_7 + D_8 + D_9)$$

$$D = (D_0 + D_1 + D_2 + D_3 + D_4 + D_5 + D_6 + D_7 + \bar{D}_8 + D_9) \cdot (D_0 + D_1 + D_2 + D_3 + \bar{D}_4 + D_5 + D_6 + D_7 + D_8 + D_9) \cdot (D_0 + D_1 + \bar{D}_2 + D_3 + D_4 + D_5 + D_6 + D_7 + D_8 + D_9) \cdot (D_0 + D_1 + D_2 + D_3 + D_4 + D_5 + \bar{D}_6 + D_7 + D_8 + D_9)$$

Circuit


A
B
C
D

Q.5) :



A is most significant
D is least significant
order of significance is
(greatest to least)
A → B → C → D

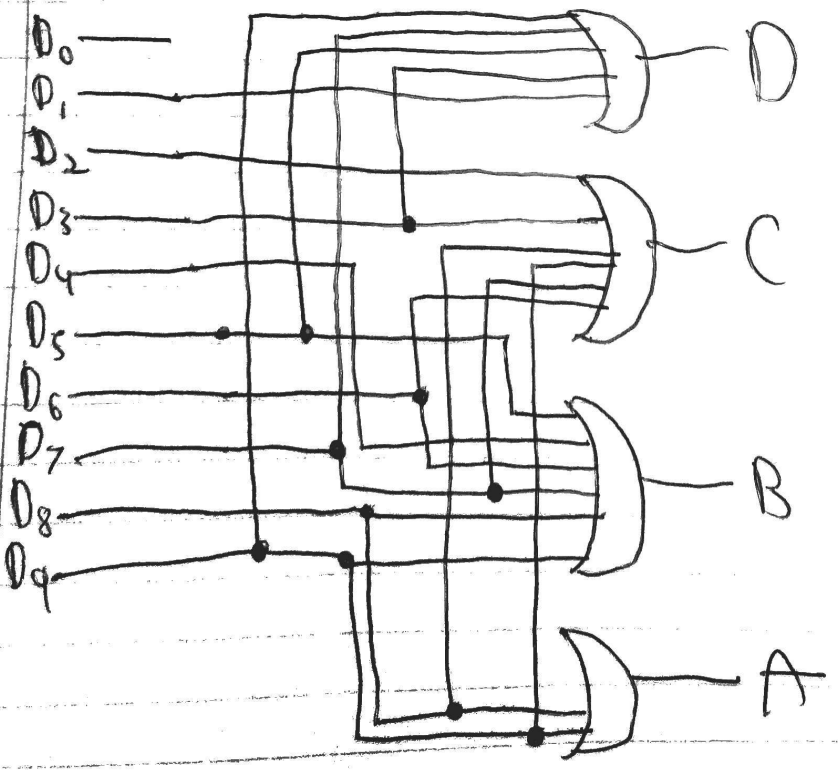
```
#hw_4 Problem 6.5 input
.i 10
.o 4
1000000000 0000
0100000000 0001
0010000000 0010
0001000000 0011
0000100000 0100
0000010000 0101
0000001000 0110
0000000100 0111
0000000010 1110
0000000001 1111
```

```
#hw_4 Problem 6.5 output
.i 10
.o 4
.p 9
0100000000 0001
0000000010 1110
0000100000 0100
0010000000 0010
0001000000 0011
0000010000 0101
0000000001 1111
0000001000 0110
0000000100 0111
.e
```

HW #5 Re Sub

6.21: In addition to what I mentioned before, I failed to mention what can happen if both chips are enabled. If more than 1 bit is allowed on the parity line problems can occur. If a 0 and 1 is output to the line there will be a short and the parity line will be floating instead of being at a logic level. The solution is the same. Add a NOT gate before one of the Enable pins to ensure that there isn't a bus contention.

6.50: Based off the truth table I drew I ~~should~~ should have realized that each bit is based off distinct 1 or 0 values rather than a logic combination of 10 bits. A is only HIGH when D_9 and P_9 are HIGH, so and so forth. From that you can design this circuit:



6.51: In addition to my drawing from before I forgot to mention that the inputs are Active LOW and the outputs are Active HIGH

Inputs: Active_LOW
Output: Active_High