21W-COMSCIM51A-1 Homework 9

CHARLES ZHANG

TOTAL POINTS

120 / 120

QUESTION 1

1 1 10 / 10

✓ **- 0 pts** **Correct**

![1.PNG](/files/6bb474e1-4bef-485c-bf87-b10128fbbbd4)

- **0.5 pts** Missing enable input
- **1 pts** Mixed MSB with LSB (xyz input instead of zyx)
- **2 pts** $$z_{2}$$ 1 term incorrect
- **2 pts** $$z_{1}$$ 1 term incorrect
- **3 pts** $$z_{1}$$ 2 terms incorrect
- **3 pts** $$z_{2}$$ 2 terms incorrect
- **4 pts** $$z_{1}$$ all 3 terms incorrect
- **4 pts** $$z_{2}$$ all 3 terms incorrect

QUESTION 2

2 2 10 / 10

✓ **- 0 pts** **Correct**

![Screen_Shot_2021-03-05_at_10.00.50_PM.png](/files/f6825f85-7ca8-43d9-b12f-113f3c3ed4fd)

- **1 pts** Minor error in the truth table
- **3 pts** Major error in the truth table
- **5 pts** Blank in the truth table
- **2.5 pts** Wrong $$C_{out}$$ (no partial credit)
- **2.5 pts** Wrong $$z$$ (no partial credit)

QUESTION 3

3 3 10 / 10

✓ **- 0 pts** **Correct**

![3.PNG](/files/0c329f92-4737-473a-8cfb-8449dd27f875)

12 AND gates: 4 AND gates going into 8 AND gates

3 NOT gates

- **1 pts** Minor error
- **3 pts** More than 12 AND gates
- **4 pts** More than 3 inverters
- **6 pts** Used 3 input AND gates
- **8 pts** Major design error

QUESTION 4

4 4 10 / 10

✓ **- 0 pts** **Correct**

![Screen_Shot_2021-03-05_at_10.02.57_PM.png](/files/9b63086b-049d-4ebe-9591-aff3c805b739)

- **1 pts** Error in x0
- **1 pts** Error in x1
- **1 pts** Error in x2
- **1 pts** Error in x3
- **3 pts** Error in s0
- **3 pts** Error in s1

QUESTION 5

5 5 10 / 10

✓ **- 0 pts** **Correct**

![5.PNG](/files/6b2269e4-7ca8-459c-9fd7-ee8669d2eb76)

$$x$$ input into both DEMUX

$$s_{1}$$ goes into enable input (inverted for least significant)

$$s_{0}$$ goes into select input

- **1 pts** Minor error
- **1 pts** $$s_{1}$$ inverted for $$y_{2}$$ and $$y_{3}$$ DEMUX instead of $$y_{0}$$ and $$y_{1}$$ DEMUX
- **2 pts** Some other form of error with the order of outputs (eg. switching $$s_{1}$$ and $$s_{0}$$, but not switching the order of $$y$$)
- **3 pts** $$s_{1}$$ (or $$s_{0}$$) is not inverted for one of the inputs
- **3 pts** Either $$s_{1}$$ or $$s_{0}$$ does not go to

the enable input

- **4 pts** Data input to both DEMUX is not $$x$$
- **5 pts** Enable bit was not used
- **6 pts** Missing an input

QUESTION 6

6 20 pts

6.1 a **10 / 10**

✓ **- 0 pts Correct**

![Screen_Shot_2021-03-05_at_10.04.55_PM.png](/files/890f3bc8-b001-4891-83eb-571d8cd637e1)

- **2 pts** Minor error in S0-S2
- **4 pts** Major error in S0-S2
- **2 pts** Minor error in D0-D7
- **4 pts** Major error in D0-D7
- **10 pts** Blank

6.2 b **10 / 10**

✓ **- 0 pts Correct**

![Screen_Shot_2021-03-05_at_10.07.06_PM.png](/files/4a406346-d137-49e7-bcf0-4cbbda3a89c7)

- **2 pts** Minor error in S0-S2
- **4 pts** Major error in S0-S2
- **2 pts** Minor error in D0-D7
- **4 pts** Major error in D0-D7
- **10 pts** Blank

QUESTION 7

7 30 pts

7.1 a **10 / 10**

✓ **- 0 pts Correct**

![Screen_Shot_2021-03-05_at_10.08.48_PM.png](/files/57ae4783-e1a3-4200-9457-5cb5f01e5145)

- **2.5 pts** Error in 1-4th row (left)
- **2.5 pts** Error in 4-8th row (left)
- **2.5 pts** Error in 1-4th row (right)
- **2.5 pts** Error in 4-8th row (right)

7.2 b **10 / 10**

✓ **- 0 pts Correct**

![Screen_Shot_2021-03-05_at_10.10.17_PM.png](/files/70b6d359-09f4-4e68-81a6-2251a40dd4d7)

- **2 pts** Error in S0-S3
- **2 pts** Error in D0-D3
- **2 pts** Error in D4-D7
- **2 pts** Error in D8-D11
- **2 pts** Error in D12-D15

7.3 c **10 / 10**

✓ **- 0 pts Correct**

![Screen_Shot_2021-03-05_at_10.12.13_PM.png](/files/5915ce3b-4a0a-43f6-a79c-4dadc0fa17e0)

- **2 pts** Error in S0-S2
- **4 pts** Error in D0-D3
- **4 pts** Error in D4-D7

QUESTION 8

20 pts

8.1 a **10 / 10**

✓ **- 0 pts Correct**

![Screen_Shot_2021-03-05_at_10.13.13_PM.png](/files/e697a4d7-b6e1-4187-84db-37ed5f7d6640)

- **2.5 pts** Error in 1-2nd row
- **2.5 pts** Error in 3-4th row
- **2.5 pts** Error in 5-6th row
- **2.5 pts** Error in 7-8th row

8.2 b **10 / 10**

✓ **- 0 pts Correct**

![Screen_Shot_2021-03-05_at_10.14.40_PM.png](/files/e22c963d-3662-4774-bd37-4d9528644b19)

- **2 pts** Error in S0-S1
- **4 pts** Error in D0-D1
- **4 pts** Error in D2-D3

CS M51A, Winter 2021, Assignment 9
(Total Mark: 120 points, 12%)

Due: Wed Mar 10rd, 10:00 AM Pacific Time

Student Name: Charles Zhang

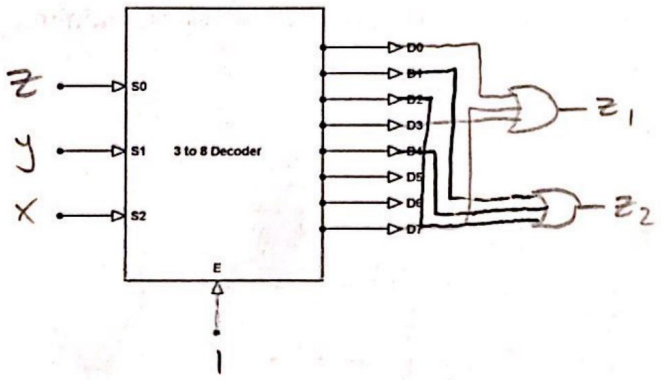
Student ID: 305413659

Note: You must complete the assignments entirely on your own, without discussing with others.

1. (10 Points) Using the following decoder, AND and OR Gates, implement the following function. Please show which decoder's input must be connected to x, y and z.

   z_1 = xyz+x'y'z'+x'yz -> $\Sigma m(7,0,3)$

   z_2 = xy'z'+x'yz'+x'y'z -> $\Sigma m(4,2,1)$



1 1 **10 / 10**

✓ **- 0 pts** **Correct**

![1.PNG](/files/6bb474e1-4bef-485c-bf87-b10128fbbbd4)

- 0.5 pts Missing enable input

- 1 pts Mixed MSB with LSB (xyz input instead of zyx)

- 2 pts $$z_{2}$$ 1 term incorrect

- 2 pts $$z_{1}$$ 1 term incorrect

- 3 pts $$z_{1}$$ 2 terms incorrect

- 3 pts $$z_{2}$$ 2 terms incorrect

- 4 pts $$z_{1}$$ all 3 terms incorrect

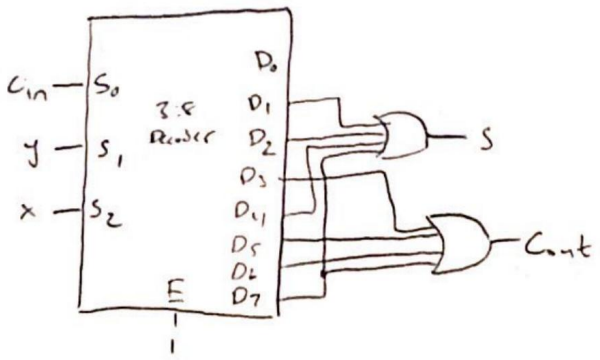
- 4 pts $$z_{2}$$ all 3 terms incorrect

2. (10 Points) Using 3-to-8 Decoders and a few other gates (such as OR, AND, or Inverters), design a full-adder. First, draw the truth table for full-adder. Then draw your design.

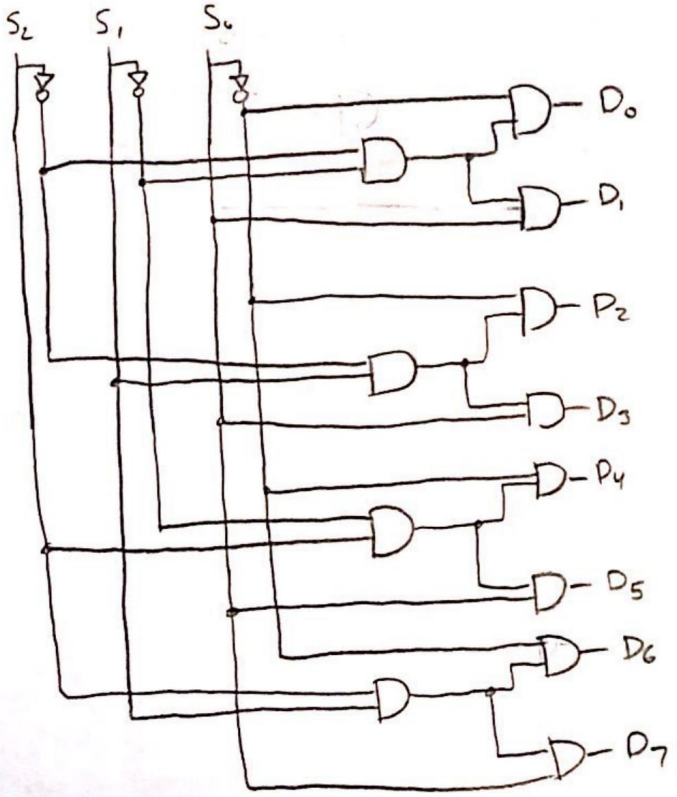
x	y	C_{in}	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$S = x'y'C_{in} + x'yC_{in}' + xy'C_{in}' + xyC_{in} = \Sigma m(1,2,4,7)$

$Cout = \Sigma m(3,5,6,7)$



3. (10 Points) Design a 3-to-8 decoder (without the enable input) using only inverters and 2-input ANDs. Your design must use the minimum number of gates possible.



2 2 **10 / 10**

✓ **- 0 pts** **Correct**

![Screen_Shot_2021-03-05_at_10.00.50_PM.png](/files/f6825f85-7ca8-43d9-b12f-113f3c3ed4fd)

- 1 pts Minor error in the truth table

- 3 pts Major error in the truth table

- 5 pts Blank in the truth table

- 2.5 pts Wrong $$C_{out}$$ (no partial credit)

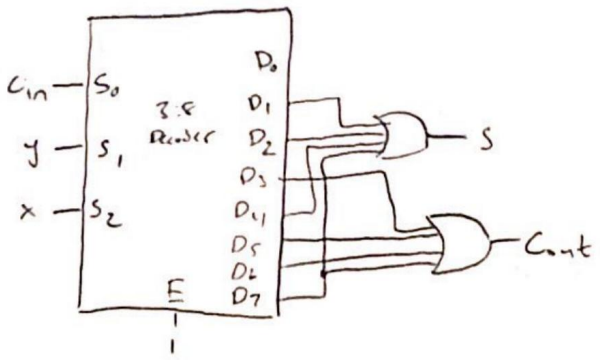
- 2.5 pts Wrong $$z$$ (no partial credit)

2. (10 Points) Using 3-to-8 Decoders and a few other gates (such as OR, AND, or Inverters), design a full-adder. First, draw the truth table for full-adder. Then draw your design.

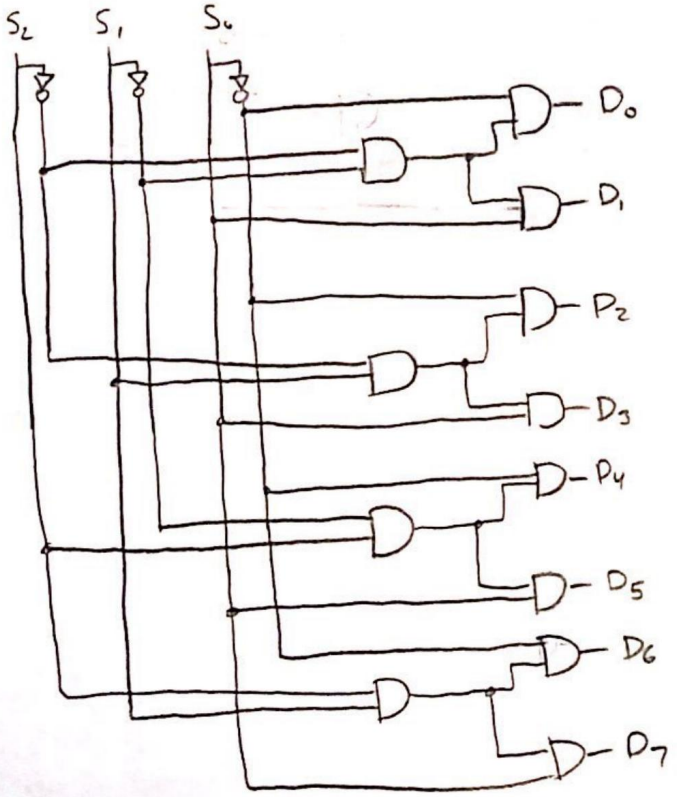
x	y	C_{in}	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$S = x'y'C_{in} + x'yC_{in}' + xy'C_{in}' + xyC_{in} = \Sigma m(1,2,4,7)$

$Cout = \Sigma m(3,5,6,7)$



3. (10 Points) Design a 3-to-8 decoder (without the enable input) using only inverters and 2-input ANDs. Your design must use the minimum number of gates possible.



3 3 **10 / 10**

✓ **- 0 pts** **Correct**

![3.PNG](/files/0c329f92-4737-473a-8cfb-8449dd27f875)

12 AND gates: 4 AND gates going into 8 AND gates

3 NOT gates

- 1 pts Minor error

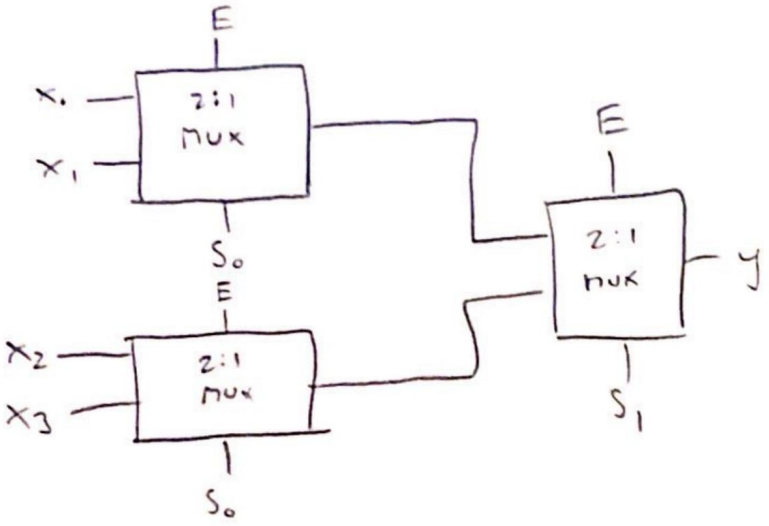
- 3 pts More than 12 AND gates

- 4 pts More than 3 inverters

- 6 pts Used 3 input AND gates

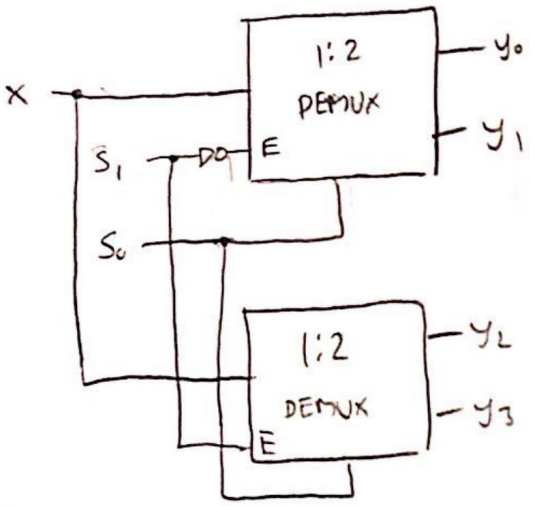
- 8 pts Major design error

4. (10 Points) Design a 4-to-1 MUX using <u>three</u> 2-to-1 MUXes. Label your inputs as x_0, x_1, x_2, x_3, output as y, and the select bits as s_1, s_0.



S_1	S_0	F
0	0	x_0
0	1	x_1
1	0	x_2
1	1	x_3

5. (10 Points) Design a 1-to-4 DEMUX using two 1-to-2 DEMUXes. Label your input as x, outputs as y_0, y_1, y_2, y_3 and the select bits as s_1, s_0.



S_1	S_0	y_0	y_1	y_2	y_3
0	0	x	0	0	0
0	1	0	x	0	0
1	0	0	0	x	0
1	1	0	0	0	x

S	y_0	y_1
0	x	0
1	0	x

4 4 **10 / 10**

✓ **- 0 pts** **Correct**

![Screen_Shot_2021-03-05_at_10.02.57_PM.png](/files/9b63086b-049d-4ebe-9591-aff3c805b739)

- 1 pts Error in x0

- 1 pts Error in x1

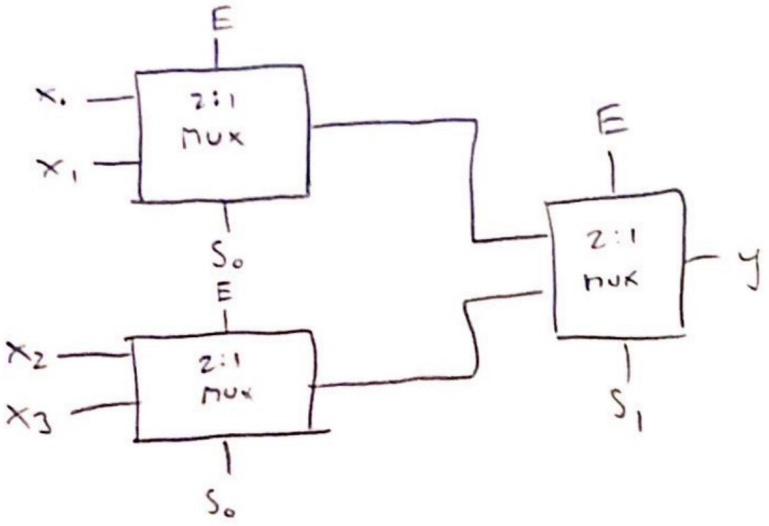
- 1 pts Error in x2

- 1 pts Error in x3

- 3 pts Error in s0

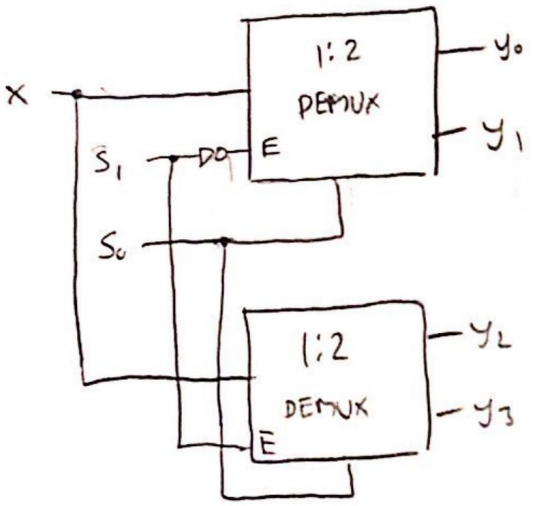
- 3 pts Error in s1

4. (10 Points) Design a 4-to-1 MUX using <u>three</u> 2-to-1 MUXes. Label your inputs as x_0, x_1, x_2, x_3, output as y, and the select bits as s_1, s_0.



S_1	S_0	F
0	0	x_0
0	1	x_1
1	0	x_2
1	1	x_3

5. (10 Points) Design a 1-to-4 DEMUX using two 1-to-2 DEMUXes. Label your input as x, outputs as y_0, y_1, y_2, y_3 and the select bits as s_1, s_0.



S_1	S_0	y_0	y_1	y_2	y_3
0	0	x	0	0	0
0	1	0	x	0	0
1	0	0	0	x	0
1	1	0	0	0	x

S	y_0	y_1
0	x	0
1	0	x

5 5 **10 / 10**

✓ **- 0 pts** **Correct**

![5.PNG](/files/6b2269e4-7ca8-459c-9fd7-ee8669d2eb76)

$$x$$ input into both DEMUX

$$s_{1}$$ goes into enable input (inverted for least significant)

$$s_{0}$$ goes into select input

- 1 pts Minor error

- 1 pts $$s_{1}$$ inverted for $$y_{2}$$ and $$y_{3}$$ DEMUX instead of $$y_{0}$$ and $$y_{1}$$ DEMUX

- 2 pts Some other form of error with the order of outputs (eg. switching $$s_{1}$$ and $$s_{0}$$, but not switching the order of $$y$$)

- 3 pts $$s_{1}$$ (or $$s_{0}$$) is not inverted for one of the inputs

- 3 pts Either $$s_{1}$$ or $$s_{0}$$ does not go to the enable input

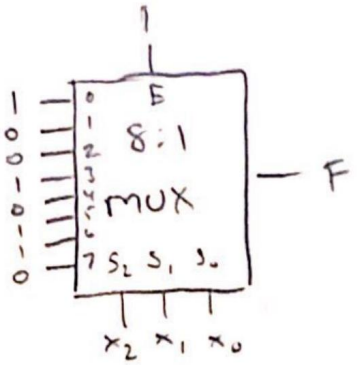
- 4 pts Data input to both DEMUX is not $$x$$

- 5 pts Enable bit was not used

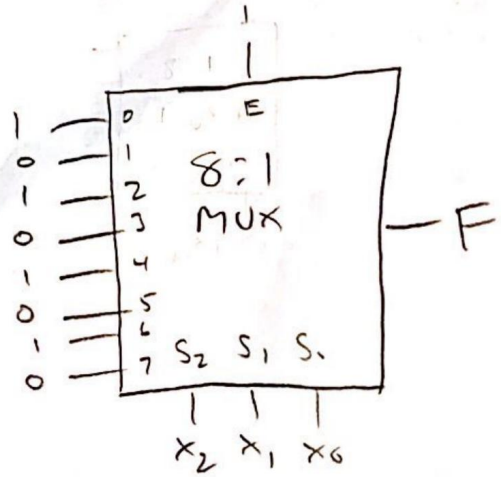
- 6 pts Missing an input

6. Assume we have three binary input (x_2,x_1,x_0), draw a design that computes the following logic expressions using 8-to-1 MUX respectively (**one** MUX for each expression). You may use "0", "1", x_2, x_1 or x_0 as your inputs.

(a) (10 Points) $\Sigma m(0,3,5,6)$



(b) (10 Points) $\Pi M(1,3,5,7) = \Sigma_{1,\wedge}(0,2,4,6)$



6.1 a **10 / 10**

✓ **- 0 pts** **Correct**

![Screen_Shot_2021-03-05_at_10.04.55_PM.png](/files/890f3bc8-b001-4891-83eb-571d8cd637e1)

- 2 pts Minor error in S0-S2

- 4 pts Major error in S0-S2

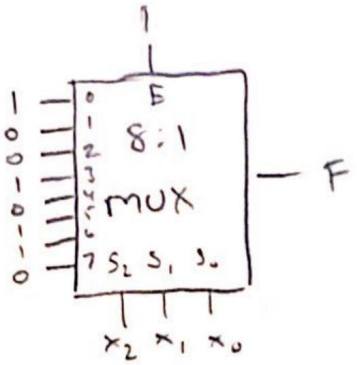
- 2 pts Minor error in D0-D7

- 4 pts Major error in D0-D7

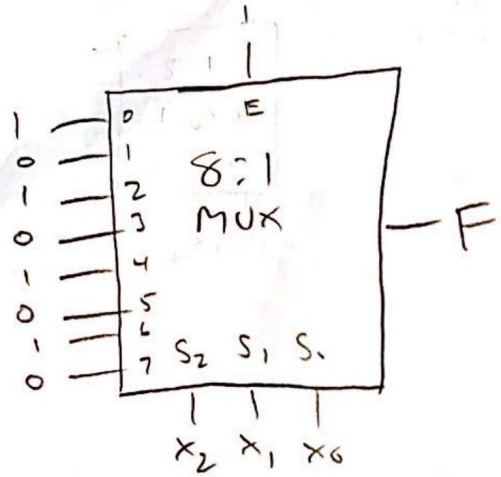
- 10 pts Blank

6. Assume we have three binary input (x_2,x_1,x_0), draw a design that computes the following logic expressions using 8-to-1 MUX respectively (**one** MUX for each expression). You may use "0", "1", x_2, x_1 or x_0 as your inputs.

(a) (10 Points) $\Sigma m(0,3,5,6)$



(b) (10 Points) $\Pi M(1,3,5,7) = \Sigma_{1,\wedge}(0,2,4,6)$



6.2 b **10 / 10**

✓ **- 0 pts** **Correct**

![Screen_Shot_2021-03-05_at_10.07.06_PM.png](/files/4a406346-d137-49e7-bcf0-4cbbda3a89c7)

- 2 pts Minor error in S0-S2

- 4 pts Major error in S0-S2

- 2 pts Minor error in D0-D7

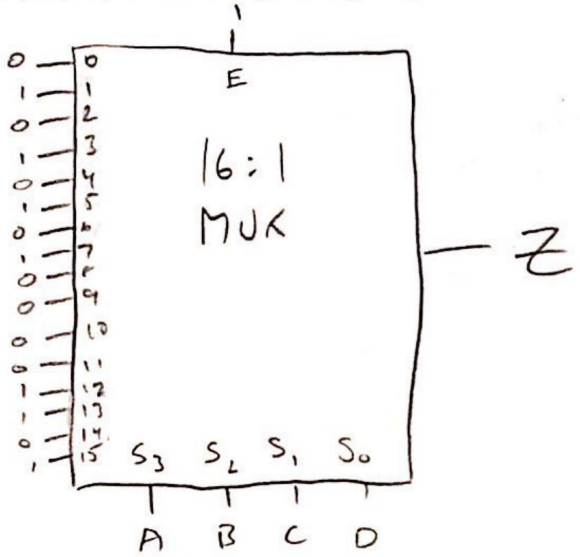
- 4 pts Major error in D0-D7

- 10 pts Blank

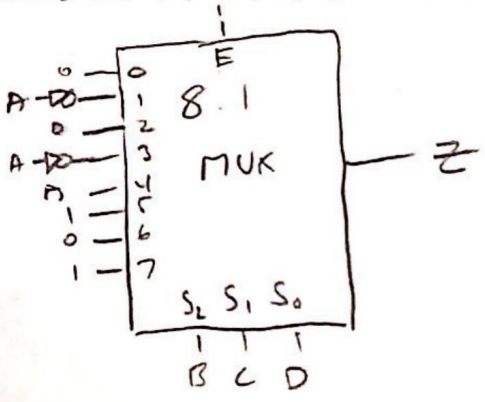
7. (a) (10 Points) Draw the truth table for the following function: Z(A,B,C,D)= ABCD+A'D+ABC'

A	B	C	D	Z	
0	0	0	0	0	
0	0	0	1	1	←
0	0	1	0	0	
0	0	1	1	1	←
0	1	0	0	0	←
0	1	0	1	1	
0	1	1	0	0	
0	1	1	1	1	
1	0	0	0	0	
1	0	0	1	0	←
1	0	1	0	0	
1	0	1	1	0	←
1	1	0	0	1	←
1	1	0	1	1	
1	1	1	0	0	
1	1	1	1	1	

(b) (10 Points) Given the truth table of part (a), implement Z using **one** 16-to-1 MUX. (You may use "0", "1", A, B, C or D as your inputs)



(c) (10 Points) Given the truth table of part (a), implement Z using only inverters and **one** 8-to-1 MUX. (You may use "0", "1", A, B, C or D as your inputs)



7.1 a **10 / 10**

✓ **- 0 pts** **Correct**

![Screen_Shot_2021-03-05_at_10.08.48_PM.png](/files/57ae4783-e1a3-4200-9457-5cb5f01e5145)

- 2.5 pts Error in 1-4th row (left)

- 2.5 pts Error in 4-8th row (left)

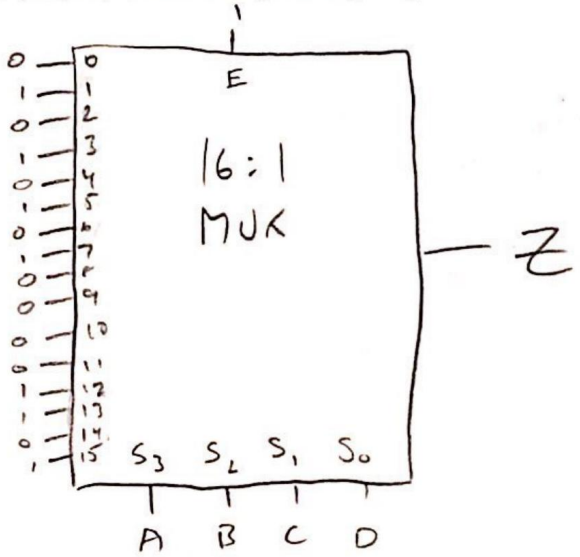
- 2.5 pts Error in 1-4th row (right)

- 2.5 pts Error in 4-8th row (right)

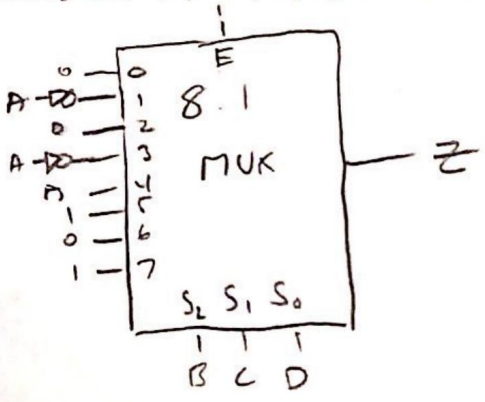
7. (a) (10 Points) Draw the truth table for the following function: Z(A,B,C,D)= ABCD+A'D+ABC'

A	B	C	D	Z	
0	0	0	0	0	
0	0	0	1	1	←
0	0	1	0	0	
0	0	1	1	1	←
0	1	0	0	0	←
0	1	0	1	1	
0	1	1	0	0	
0	1	1	1	1	
1	0	0	0	0	
1	0	0	1	0	←
1	0	1	0	0	
1	0	1	1	0	←
1	1	0	0	1	←
1	1	0	1	1	
1	1	1	0	0	
1	1	1	1	1	

(b) (10 Points) Given the truth table of part (a), implement Z using **one** 16-to-1 MUX. (You may use "0", "1", A, B, C or D as your inputs)



(c) (10 Points) Given the truth table of part (a), implement Z using only inverters and **one** 8-to-1 MUX. (You may use "0", "1", A, B, C or D as your inputs)



7.2 b **10 / 10**

✓ **- 0 pts** **Correct**

![Screen_Shot_2021-03-05_at_10.10.17_PM.png](/files/70b6d359-09f4-4e68-81a6-2251a40dd4d7)

- 2 pts Error in S0-S3

- 2 pts Error in D0-D3

- 2 pts Error in D4-D7

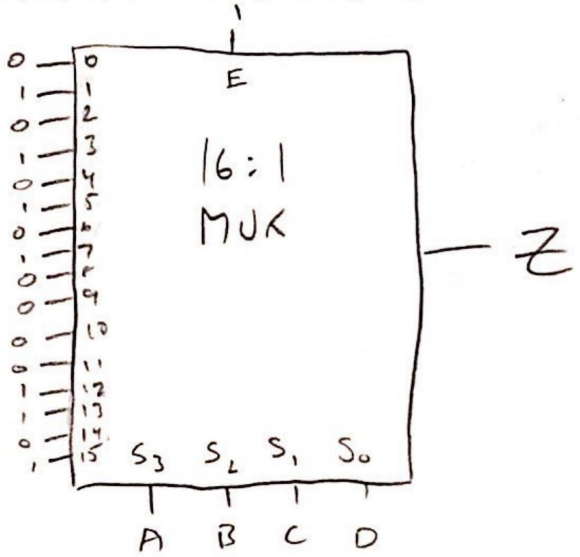
- 2 pts Error in D8-D11

- 2 pts Error in D12-D15

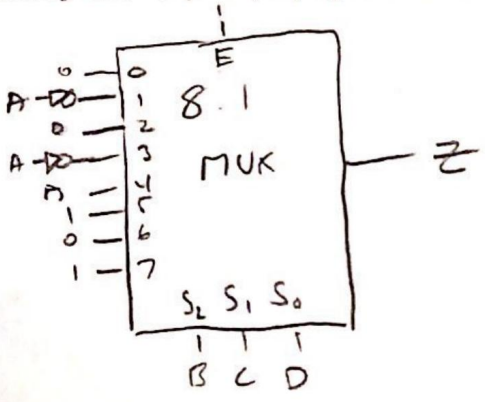
7. (a) (10 Points) Draw the truth table for the following function: Z(A,B,C,D)= ABCD+A'D+ABC'

A	B	C	D	Z	
0	0	0	0	0	
0	0	0	1	1	←
0	0	1	0	0	
0	0	1	1	1	←
0	1	0	0	0	←
0	1	0	1	1	
0	1	1	0	0	
0	1	1	1	1	
1	0	0	0	0	
1	0	0	1	0	←
1	0	1	0	0	
1	0	1	1	0	←
1	1	0	0	1	←
1	1	0	1	1	
1	1	1	0	0	
1	1	1	1	1	

(b) (10 Points) Given the truth table of part (a), implement Z using **one** 16-to-1 MUX. (You may use "0", "1", A, B, C or D as your inputs)



(c) (10 Points) Given the truth table of part (a), implement Z using only inverters and **one** 8-to-1 MUX. (You may use "0", "1", A, B, C or D as your inputs)



7.3 C **10 / 10**

✓ **- 0 pts** **Correct**

![Screen_Shot_2021-03-05_at_10.12.13_PM.png](/files/5915ce3b-4a0a-43f6-a79c-4dadc0fa17e0)

- 2 pts Error in S0-S2

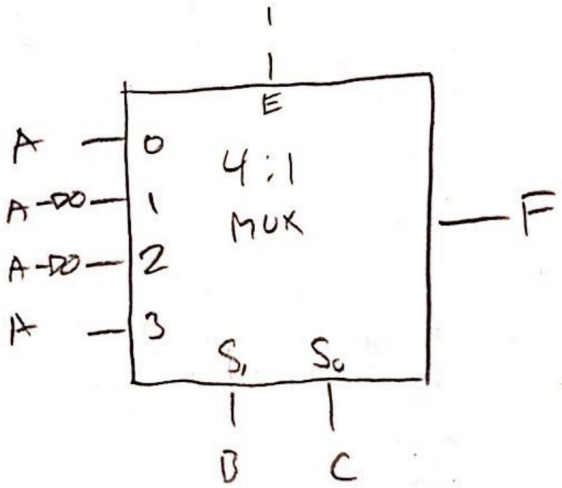
- 4 pts Error in D0-D3

- 4 pts Error in D4-D7

8. (a) (10 Points) Draw the truth table of a 3-input XOR function. (i.e. F=XOR(A,B,C))

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(b) (10 Points) Draw a design which computes F using only inverters and **one** 4-to-1 MUX. (You may use "0", "1", A, B, or C as your inputs)



8.1 a **10 / 10**

✓ **- 0 pts** **Correct**

![Screen_Shot_2021-03-05_at_10.13.13_PM.png](/files/e697a4d7-b6e1-4187-84db-37ed5f7d6640)

- 2.5 pts Error in 1-2nd row

- 2.5 pts Error in 3-4th row

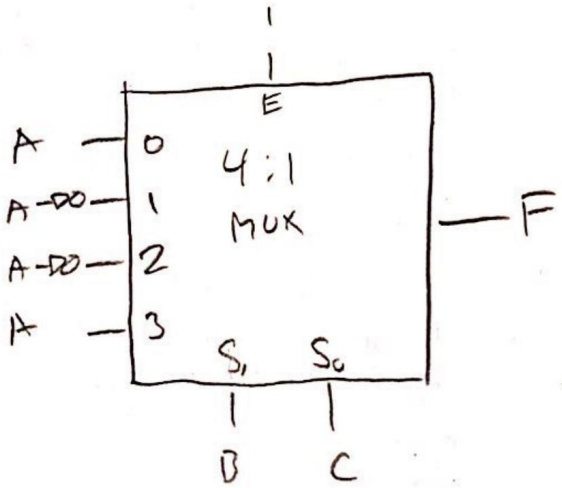
- 2.5 pts Error in 5-6th row

- 2.5 pts Error in 7-8th row

8. (a) (10 Points) Draw the truth table of a 3-input XOR function. (i.e. F=XOR(A,B,C))

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(b) (10 Points) Draw a design which computes F using only inverters and **one** 4-to-1 MUX. (You may use "0", "1", A, B, or C as your inputs)



8.2 b **10 / 10**

✓ **- 0 pts** **Correct**

![Screen_Shot_2021-03-05_at_10.14.40_PM.png](/files/e22c963d-3662-4774-bd37-4d9528644b19)

- 2 pts Error in S0-S1

- 4 pts Error in D0-D1

- 4 pts Error in D2-D3