Jonathan Chu

Name

00483222O

SID #

EEM16/CSM51A (Fall 2017)

Logic Design of Digital Systems

Prof. Ankur Mehta : mehtank@ucla.edu

Problem set 2 | **Assigned Monday Oct. 16, 2017**
Show all work. | **due 4pm Monday Oct. 30, 2017**

nstructions

his homework is to be done individually. You may consult with others to share thoughts and ideas, but all of your bmitted work must be yours alone. Be sure to indicate with whom you've collaborated and in what manner.

You may use any tools or refer to published papers, books, or course notes. You're allowed to make use of online ols such as Logisim, WolframAlpha, etc., provided you properly cite them in the space below.

You must submit this cover sheet plus all pages of your solutions based on the procedure below. Please write early and neatly — if we cannot easily decipher what you have written, you will get zero credit.

ubmission procedure

ou need to submit your solution online at Gradescope:

`https://gradescope.com/`

Please see the following guide from Gradescope for submitting homework. You will need to upload a PDF and ark where each question is answered.

`http://gradescope-static-assets.s3-us-west-2.amazonaws.com/help/submitting_hw_guide.pdf`

Collaborators

dentify with whom you've collaborated and in what manner, if any.

Angela Wu and Edward Chen, went over some answers

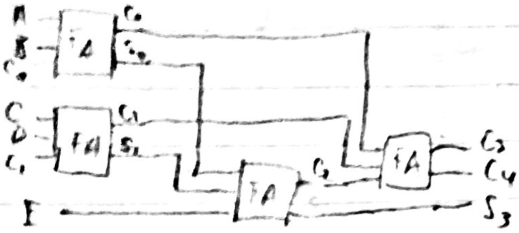
Online resources

dentify which online tools you've used, if any.

none

EEM16 Pset 2

1.1 a)



$$\begin{array}{r} A \\ + B \\ \hline C_0 S_0 \end{array} \qquad \begin{array}{r} C \\ + D \\ \hline C_1 S_1 \end{array}$$

$$\begin{array}{r} S_0 \\ S_1 \\ + E \\ \hline C_2 S_3 \end{array} \longrightarrow \begin{array}{r} C_0 \\ C_1 \\ + C_2 \\ \hline C_3 C_4 \end{array}$$

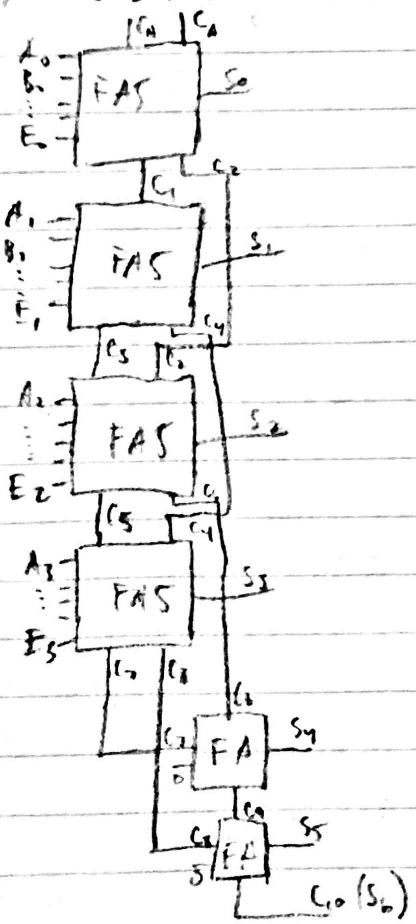
$S_3 C_3 C_4$

b) Propagation delay of FA5 = maximum of propagation delays of each path to output
$= \max(6t_{PD}, 9t_{PD}, 3t_{PD}) = 9t_{PD}$

1.2 a) 1111 + 1111 + 1111 + 1111 + 1111
$= 15 \times 5 = 75 < 127$

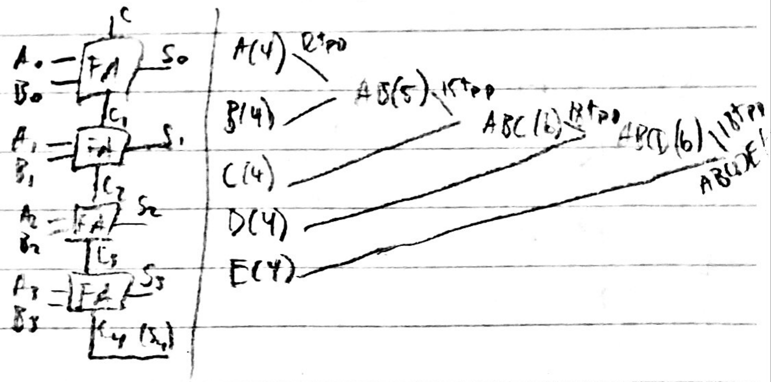
7 bits

b) $A_3A_2A_1A_0 + B_3B_2B_1B_0 + \dots + E_3E_2E_1E_0$
$= S_0S_1S_2S_3S_4S_5S_6$



c) $4(9t_{PD}) + 2(3t_{PD}) = 42t_{PD}$
$4 \cdot 4 + 2 = 18$ total FA's

d) 2 input 4-bit ripple carry adder. $12t_{PD}$ each



$12 + 15 + 18 + 17 = 63t_{PD}$
$4 + 5 + 6 + 6 = 21$ FA's

2.1 a)

E = 0

B_1B_0 \ A_1A_0	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

E = 1

B_1B_0 \ A_1A_0	00	01	11	10
00	1	0	0	0
01	1	1	0	0
11	1	1	1	1
10	1	1	0	1

(dotted loop) = 3D

$\square$ = in the page

b) $f = \bar{A}_1 B_0 \bar{E} + \bar{A}_1 B_1 + \bar{A}_1 \bar{A}_0 E + \bar{A}_1 \bar{A}_0 B_0 + B_1 B_0 E + \bar{A}_0 B_1 B_0 \bar{E} + A_1 \bar{A}_0 B_1 E$

c) no, it is not lenient since the $A_1\bar{A}_0B_1E$ and $\bar{A}_1\bar{A}_0E$ implicants are not connected and thus a glitch could result from A_1 changing from high to low.

2.2 a)

2.2 a)

A_1A_0 \ B_1B_0	00	01	11	10
00	x	0	0	0
01	1	x	0	0
11	1	1	x	1
10	1	1	0	x

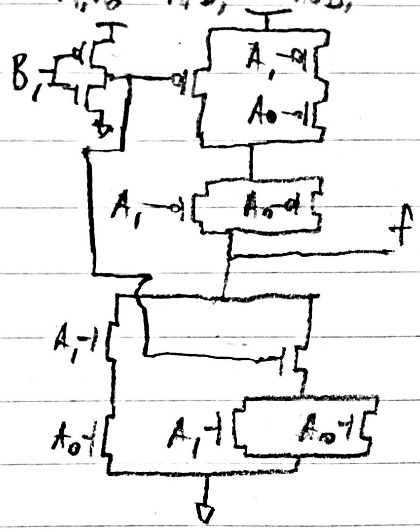
This K-Map is $\overline{F}$, where

$\overline{F}=1$ if $A>B$ and

$\overline{F}=0$ if $A<B$. I have set it up this way so that my CMOS gate doesn't need to be inverted at the end.

b) $f = A_1A_0 + A_1\overline{B_1} + A_0\overline{B_1} = A_1A_0 + \overline{B_1}(A_1 + A_0)$

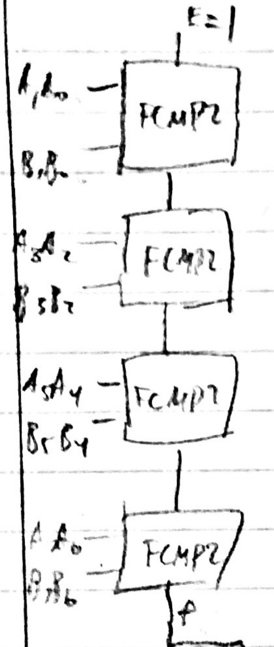
c)



2.3 L = Label ℓ = likelihood

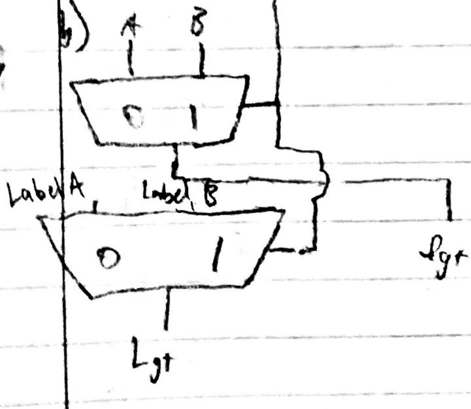
a) FCMP2: 2 2-bit inputs

$\underbrace{A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0}_{\text{likelihood of } A}$ $\underbrace{B_7 B_6 B_5 B_4 B_3 B_2 B_1 B_0}_{\text{likelihood of } B}$

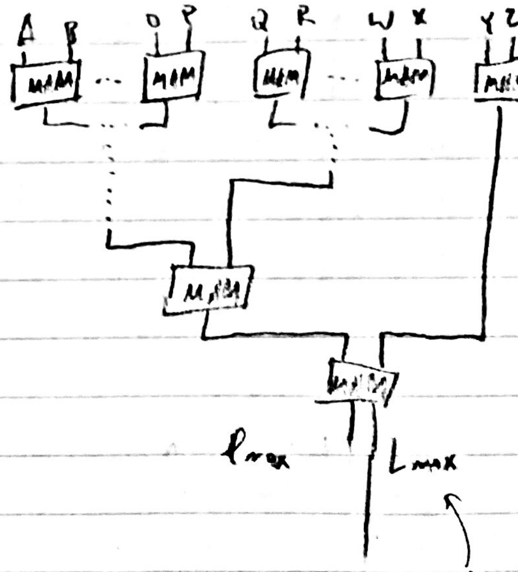


← A ≥ B

b)



c)



← likelihood vals

the output we care about is likelihood values

5-bit encoding for most likely letter

3.1 a) 32 rows (26 letters, 6 dashes)

7 columns (for each segment)

b)

	In	Out		In	Out		In	Out		In	Out
A	0x0	0x77	H	0x7	0x76	O	0xE	0x5C	V	0x15	0x1C
B	0x1	0x7C	I	0x8	0x06	P	0xF	0x73	W	0x16	0x2A
C	0x2	0x58	J	0x9	0x1E	Q	0x10	0x67	X	0x17	0x49
D	0x3	0x5E	K	0xA	0x78	R	0x11	0x50	Y	0x18	0x6E
E	0x4	0x79	L	0xB	0x38	S	0x12	0x6D	Z	0x19	0x5B
F	0x5	0x71	M	0xC	0x15	T	0x13	0x46	invalid	0x1A ⋮ 0x1F	0x40 ⋮ 0x40
G	0x6	0x6F	N	0xD	0x54	U	0x14	0x3E			

3.2 a) $\boxed{32}$

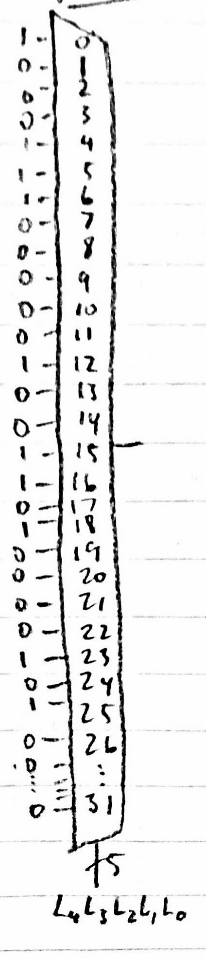
b) seg. A

seg G

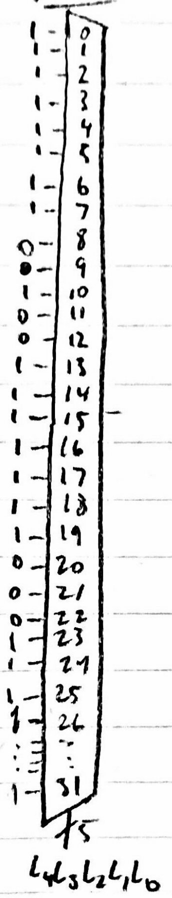
1-

3.2 a) /32/

b) eg: A



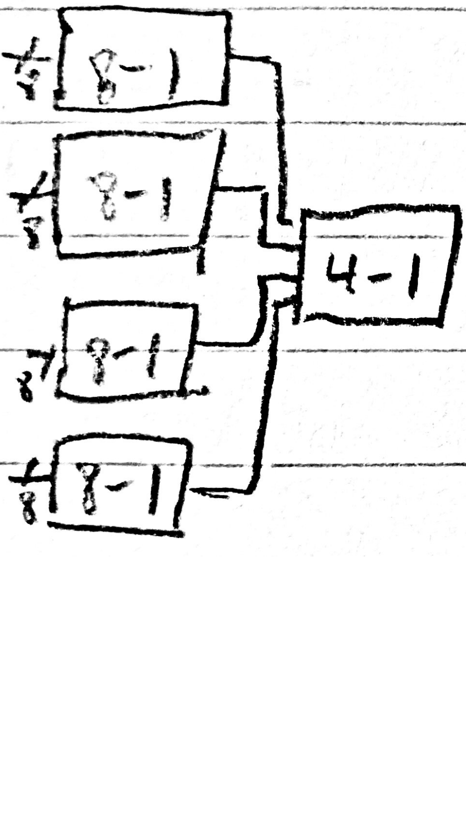
eg G



* Values in MUX are written in decimal to save space/preserve readability

Letter	Decimal
A	0
B	1
⋮	⋮
Z	25

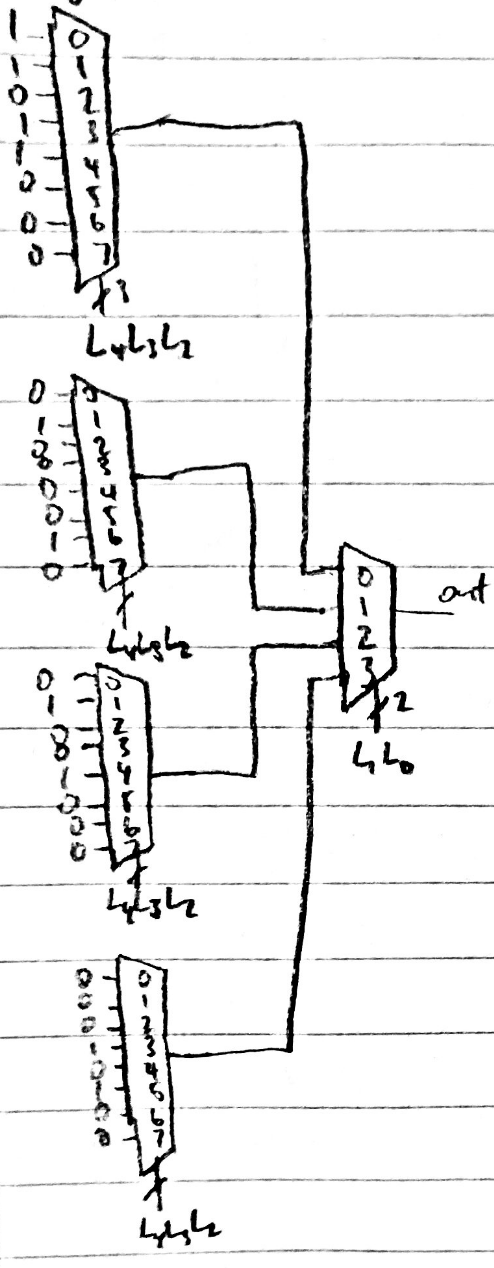
c)



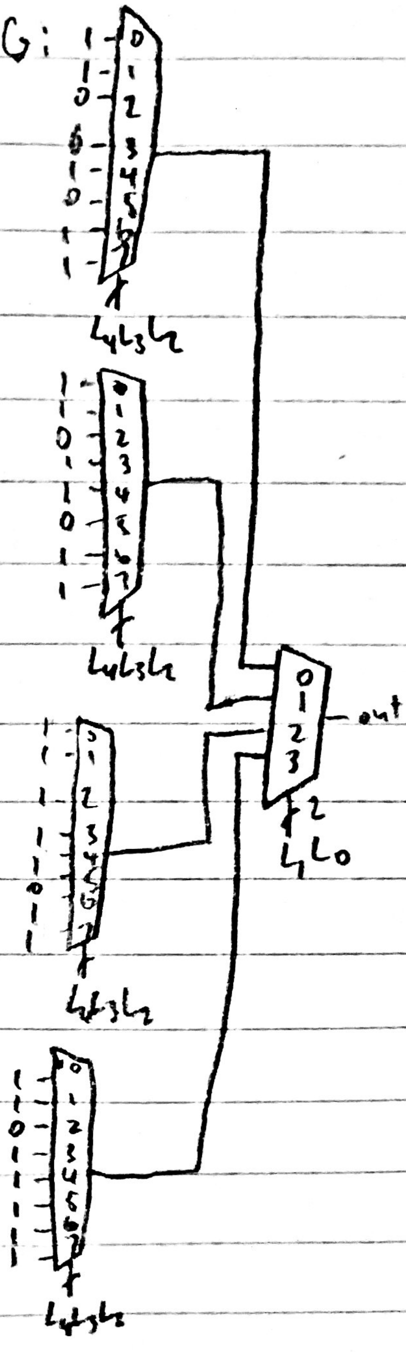
5 MUXes

d) Again, I will use decimal values in MUXes to save space

A:



G:



4. I will write this problem on FP/2's complement since all other major topics have been hit hard by these PSE3's already.

Consider the following floating point bit representation system FP1:

$$\underbrace{\_}_{S}\ \ \underbrace{\_\,\_\,\_}_{E}\ \ \underbrace{\_\,\_\,\_\,\_}_{M}$$

4.1 a) Consider the bits 10001010. Convert this to a decimal value assuming it is a value in:

i) binary
ii) hexadecimal
iii) FP1
iv) Two's Complement
v) Signed Magnitude

b) Do the same for the bit string 01011000.

i – v

4.2 a) Give the sum (in decimal) of the answers to (4.1a, iv) and (4.1a, v). Convert it to two's complement and then to signed magnitude.

b) Double your answer to (4.1b, iii) and add it to your answer to (4.1a, v). Explain why it cannot be represented exactly in FP1.

4.1 a) i. $2+8+128 = 138$ 10001010

ii. $16 + 16^3 + 16^7$

iii. 0

iv. $-128+2+8 = -118$

v. -10

b) i. $8+16+64 = 88$ 01011000

ii. $16^3+16^4+16^6$

iii. $\left(1+\frac{8}{16}\right)^{5-3} = (1.5)^2 = 2.25$

iv. 88

v. 88

4.2 a) -128

Two's Complement: 10000000

Signed Magnitude: 110000000

b) $2.25 \cdot 2 + (-10) = -5.5$

-5.5 cannot be represented in FPI because there do not exist M and E s.t. $(1+M)^{E-3} = 5.5$. M is a.t.f. $\frac{2^k}{16}$ where $k \le 15$, and there is no M that produces a number raised to an integer power that yields a value with magnitude >5 and fraction as clean as $\frac{1}{2}$