EEM16 Midterm

[redacted]

TOTAL POINTS

51.5 / 65

QUESTION 1

1 Problem #1 13 / 18

✓ **- 0 pts a) is Correct**

- 0 pts b) is correct

- 0 pts c) is correct

✓ **- 0 pts d) is correct**

✓ **- 0 pts e) is correct**

- 1 pts a) added don't cares as minterms

- 3 pts a) is incorrect

✓ **- 2 pts b) is partially correct**

- 5 pts b) is incorrect

✓ **- 3 pts c) is incorrect**

- 3 pts d) is incorrect

- 2 pts e) is partially incorrect

- 4 pts e) is incorrect

- 1 pts d) partially incorrect

- 2 pts c) partially incorrect

QUESTION 2

2 Problem #2 13 / 14

- 0 pts all correct

- 2 pts Part (a): answer other than 2, 3 or 4

- 1 pts Part (b): deduct for wrong demorgan

- 0.5 pts Part(b): partial improper SoP

- 1 pts Part(b): For improper SoP form

- 2 pts Part(b) quite wrong SoP

- 4 pts Part(b): all wrong

- 1.5 pts Part(c): if Not distributive

- 2 pts Pat(c) no 6 SoP

- 1 pts Part(c). For each wrong SoP term

- 4 pts Part(c): all wrong

- 0.5 pts Part(d): Partial wrong reduced expression

- 1 pts Part(d): quite wrong reduced expression

✓ **- 1 pts Part(d): For one missing property**

- 2 pts Part(d): For two missing properties

QUESTION 3

3 Problem #3 18 / 22

- 0 pts Correct. Good Job.

- 1 pts (a) math error

- 2 pts (a) incorrect

- 1 pts (a) (b) or (c) math error

- 2 pts (b) incorrect

- 2 pts (c) incorrect = -105

- 0.5 pts (c) negative error for 2's complement

✓ **- 2 pts (d) incorrect hex**

- 2 pts (e) incorrect BCD

- 1 pts (f) partial credit

- 2 pts (f) incorrect

- 2 pts (g) incorrect bias

- 2 pts (g) incorrect real number

- 1 pts (g) partial credit for error

- 6 pts (h) incorrect

- 2 pts (h) incorrect floating point choice

- 1 pts (h) partially incorrect min bits for mantissa

- 2 pts (h) incorrect min bits for mantissa

- 1 pts (h) partially incorrect min bits for exponent

✓ **- 2 pts (h) incorrect min bits for exponent**

QUESTION 4

4 Problem #5 7.5 / 11

✓ **- 0 pts (a) Correct. Nice Job!**

- 7 pts (a) incorrect

- 5.5 pts (a) partial for ok attempt

- 3 pts (a) Too much excessive logic

- 0 pts (a) No or incorrect Hit logic

- 0.5 pts (a) Some unnecessary logic

- 0 pts (b) Correct. Nice Job!

- 4 pts (b) incorrect

✓ **- 3.5 pts (b) Partial credit for attempt**

- 1 pts (b) incorrect bit weight (hop and adder cnt)

- 1 pts (b) no or incorrect adder count or hop

- **0.5 pts** (b) Good attempt but slightly too many FA
- **1.5 pts** (b) ok attempt but incorrect design
- **3 pts** (b) partial credit for attempt

Midterm Exam

Name (Last, First): [redacted]

Student Id #: [redacted]

Do not start working until instructed to do so.

1. *You must answer in the **space provided** for answers after every question. We will ignore answers written anywhere else in the booklet. **All pages in this booklet must be accounted** for otherwise it will not be graded.*
2. *You are permitted 1 page of notes 8.5x11 (front and back).*
3. *You may not use any electronic device.*

Following table to be filled by course staff only

	Maximum Score	Your Score
Question 1		
Question 2		
Question 3		
Question 4		
Question 5		
TOTAL	**100**	

Question #1

Consider the Boolean function defined by the truth table below where *A, B, C, and D* are inputs, and *Y* is the sole output.

	A	B	C	D	Y	
✓	0	0	0	0	1	0
✓	0	0	0	1	0	1
✓	0	0	1	0	0	2
✓	0	0	1	1	0	3
✓	0	1	0	0	1	4
✓	0	1	0	1	1	5
✓	0	1	1	0	1	6
✓	0	1	1	1	1	7
✓	1	0	0	0	1	8
✓	1	0	0	1	1	9
✓	1	0	1	0	0	10
✓	1	0	1	1	1	11
✓	1	1	0	0	0	12
✓	1	1	0	1	X	13
✓	1	1	1	0	1	14
✓	1	1	1	1	X	15

(a) Complete the following statements

$$Y = \sum m(\ 0,4,5,6,7,8,9,11,14 \qquad)$$

(b) Complete the Karnaugh Map shown below, **circle** the prime implicants.

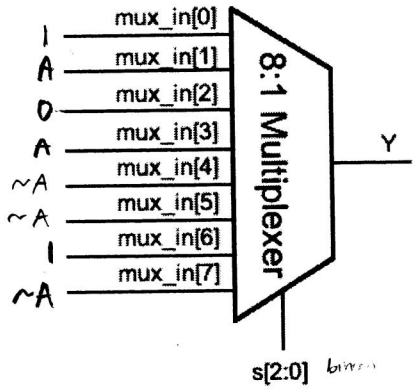
CD \ AB	"00"	"01"	"11"	"10"
"00"	1	1	0	1
"01"	0	1	X	1
"11"	0	1	X	1
"10"	0	1	1	0

How many prime implicants are there? 6

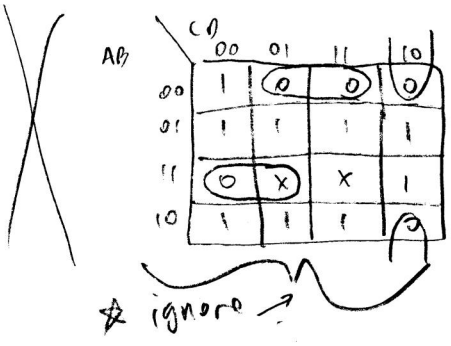
(c) Write the Boolean (sum-of-product) expression for the essential prime implicants (if any).

EssentialPrimeImplicants = none

(d) Implement the function *Y* using an 8-input multiplexer. The select signal is *s[2:0]={B,C,D}* where *s=3'b100* is *B*=1 and *C=D*=0 selecting the input *mux_in*[4]. *A* or *~A* are permissible as inputs, *mux_in*[7:0]. Write the desired inputs on the figure below.



(e) Implement ¬*Y* using the minimum # of NOR gates with fewest # of inputs (minimize literals and terms). PRODUCT OF SUMS



$Y = (\bar{B}\wedge\bar{C}\wedge\bar{D}) \vee (\bar{A}\wedge B) \vee (B\wedge D) \vee (A\wedge D) \vee (B\wedge C)$

$\bar{Y} = (B\vee C\vee D)\wedge(A\vee\bar{B})\wedge(\bar{B}\vee\bar{D})\wedge(\bar{A}\vee\bar{D})\wedge(\bar{B}\vee\bar{C})$

Question #2

$$Y = \neg\left(\neg(a \wedge \neg b) \vee \left(c \wedge \neg(d \vee e)\right)\right)$$

(a) For the above Boolean function, if you were to convert the above expression into a sum-of-product representation, how many times did you have to apply DeMorgan's theorem?

2 11

(b) For part (a), what is the resulting function?

$Y = ((a \wedge \bar{b}) \wedge \overline{(c \wedge (d \vee e))}) = ((a \wedge \bar{b}) \wedge (\bar{c} \vee (d \vee e)) = (a \wedge \bar{b}) \wedge (\bar{c} \vee d \vee e) = (a \wedge \bar{b} \wedge \bar{c}) \vee (a \wedge \bar{b} \wedge d) \vee (a \wedge \bar{b} \wedge e)$

$Y =$ $(a \wedge \bar{b} \wedge \bar{c}) \vee (a \wedge \bar{b} \wedge d) \vee (a \wedge \bar{b} \wedge e)$

(c) The following expression can be written as a 6-term sum-of-product,

$$Y = (a \vee b) \wedge (a \vee \neg b \vee \neg c)$$

What Boolean property do you need to apply to do this?

Distributive property

Without reducing, what are the 6 product terms?

a a

$(a \wedge a) \vee (a \wedge \bar{b}) \vee (a \wedge \bar{c}) \vee (b \wedge a) \vee (b \wedge \bar{b}) \vee (b \wedge \bar{c})$

(d) The 6-term sum-of-product of part (c) can obviously be reduced.

What is the reduced expression?

$a \vee (b \wedge \bar{c})$

What Boolean axioms or properties are needed for the reduction?

Absorption and Combining Properties

$a \vee (a \wedge \bar{b}) \vee (a \wedge \bar{c}) \vee (b \wedge a) \vee (b \wedge \bar{c})$

a a

$a \vee a \vee (a \wedge \bar{c}) \vee (b \wedge \bar{c})$

$a \vee (a \wedge \bar{c}) \vee (b \wedge \bar{c})$

$a \vee (b \wedge \bar{c})$

Question #3

(a) The following 8 bits can be used to represent different numbers depending on the encoding

8b'10010111 $128+16+4+2+1 = 151$

If this was unsigned, what is the corresponding integer? 151

(b) If the 8 bits in (a) was sign magnitude, what is the corresponding integer? −23

$-0010111 = -(16+4+2+1) = -23$

(c) If the 8 bits in (a) was 2's complement, what is the corresponding integer? −105

$01101000 + 1 = 01101001$ $-1(64+32+8+1) = -105$ (-105)

(d) If the 8 bits in (a) was hexadecimal, what is the corresponding hexadecimal? 2685 01265

10010111 $16^7+16^4+16^2+16^1+16^0$

(e) If the 8 bits in (a) was binary coded decimal, what is the corresponding integer? 97

1001_0111 9 (1+2+4) 97

(f) If the 8 bits is fixed point 1001.0111, what is the corresponding number? 9.4375

9.____ $2^{-2}+2^{-3}+2^{-4} =$

(g) If the 8 bits in (a) was a 4E3 floating point number (IEEE format S+EEE+MMMM), $E - b = \text{shift} \rightarrow \text{shift} = -2$

What is the bias? 3 $\text{bias} = \left\lfloor \frac{2^3-1}{2} \right\rfloor = 3$ 1_001_0111 $E = 1$

$1.0111 \times 2^{-2} = .0010111 = .359375$

What is the corresponding real number? −0.359375

(h) Military temperature range is -55°C to +125°C with 1% accuracy. min magnitude = 0.25°C

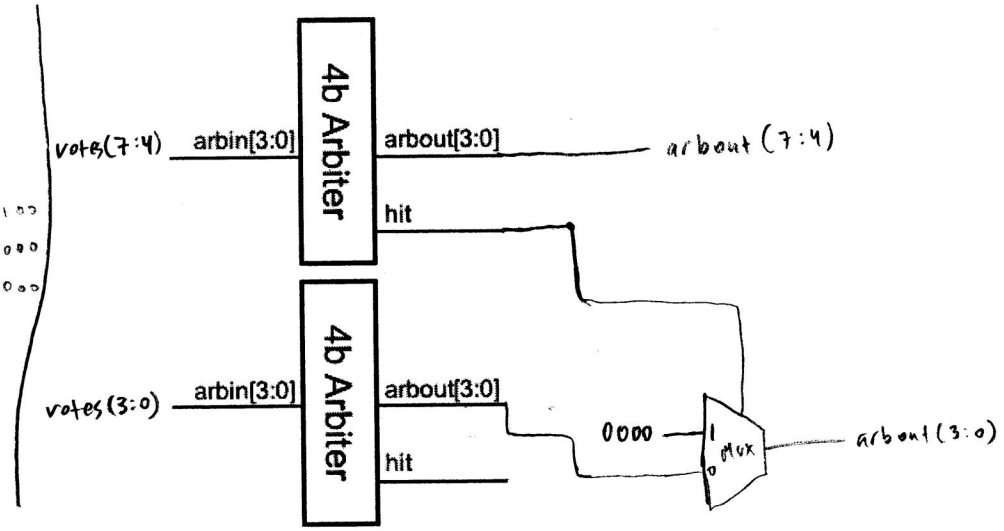
Would you choose floating point or fixed point? floating point

If you are to represent this in floating point, what is the minimum # of bits for mantissa? 6 1.0001×2^{-2}

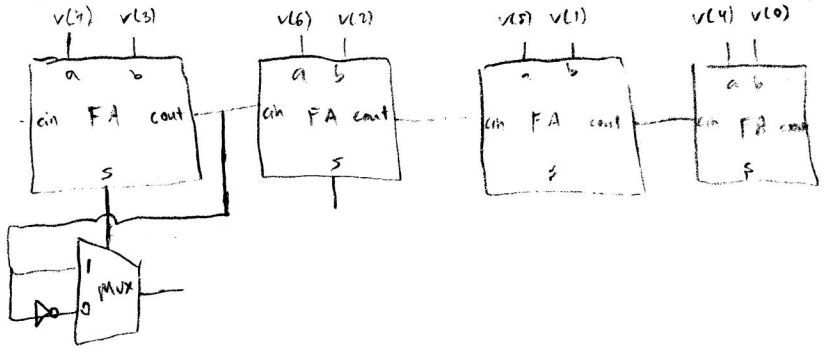
And, what is the minimum # of bits for exponent? 2 $2^{-2} = 0.25$

Question #5

(a) Given 8-bit input, *votes*[7:0], in which any number of the inputs can be a 1'b1. Build an **arbiter** that provides an 8-bit output, *arbout*[7:0], that is 1-hot. The hot signal corresponds to the position with the highest priority. Note that votes[7] has higher priority than votes[6] etc. You have available to you a module ARB that is a 4-bit arbiter already built that you **must** use. ARB accepts as inputs *arbin*[3:0] and outputs *arbout*[3:0] and a *hit* signal to indicate that one or more of the signals is a 1'b1. You also have available to you INV (inverters), and 2-input MUX (multiplexers). Recall that you can implement considerable arbitrary logic with 2-input MUXs.



(b) Now, the *votes*[7:0] need to be counted. You have available Full Adders (FA) as building blocks for implementing a design. If the delay of the logic is determined by the number of hops where each hop is the traversal of a Full-Adder from any input (*a,b,and c*) to any output (*sum, carry*). Design your block to minimize this delay. Note that your design should output 4 bits to indicate the binary count, *cnt*[3:0].



V = votes

How many Full Adders to you need? 4

How many hops is your design? 4

Blank page for scratch work

Blank page for scratch work