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	0	0
0	0	0	1	X	1
0	0	1	0	1	2
0	0	1	1	0	3
0	1	0	0	1	4
0	1	0	1	1	5
0	1	1	0	0	6
0	1	1	1	0	7
1	0	0	0	0	8
1	0	0	1	X	9
1	0	1	0	1	10
1	0	1	1	1	11
1	1	0	0	1	12
1	1	0	1	X	13
1	1	1	0	1	14
1	1	1	1	1	15

(a) Complete the following statements

$$\neg Y = \sum m(2, 4, 5, 10, 11, 12, 14, 15\quad)$$

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

AB

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

1111

Essential
$AC, \bar{B}C\bar{D}, B\bar{C}$

How many prime implicants are there? 4

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

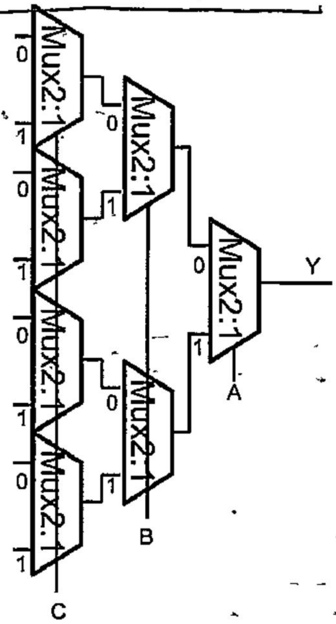
~~[illegible] OR's~~ ORING of AND's

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

product of sums

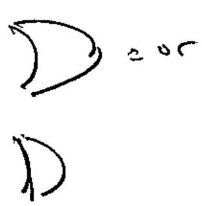
(d) Implement the function Y (not ~Y) using a tree of 2 input multiplexers as shown. The select signals are *A*, *B*, and *C*. The top input of the multiplexer is selected when the select-signal=0 and vice versa. Write the desired inputs on the figure below. You may use *D* or ~*D* as input but avoid using them as much as possible.

Y=



(e) Write Boolean expression for Y as a *minimal* product-of-sum. Then implement by using only NOR gates (each gate can have multiple inputs). Use the minimum number of NORs with fewest # of total inputs (minimize literals and terms). You may assume true and complement inputs are available.

$Y =$ ________________________________



m(2) or m(4) or m(5) or m(10) or m(11) or m(12) or m(14) or m(15)

Question #2

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

(a) For the above Boolean function, convert the above expression into a minimal product-of-sum.

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

$(a \vee \bar{d}) \vee [(\bar{e} \wedge \bar{c}) \vee (\bar{e} \wedge \bar{b})]$

$\left(a \vee [(\bar{e} \wedge \bar{c}) \vee (\bar{e} \wedge \bar{b})]\right) \vee \left(\bar{d} \vee [(\bar{e} \wedge \bar{c}) \vee (\bar{e} \wedge \bar{b})]\right)$

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

$a \vee \bar{e}$

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

(b) If you need to represent the equation in (a) in a Fully **Conjunctive** Normal Form, how many terms, and write the expression.

AB

CDE	00	01	11	10
000	1	1	1	1
001	1	1	1	1
011	0	0	1	1
010	1	1	1	1
100	1	1	1	1
101	1	1	1	1
111	0	0	1	1
110	1	0	1	1

CDE

(A), CD, $C\bar{D}$, $\bar{C}DE$,

BE

00110

00000

00000

10110

ABCDE	
00000	1
00001	1
00010	1
00011	0
00100	1
00101	1
00110	1
00111	0
01000	1
01001	1
01010	1
01011	0
01100	1
01101	1
01110	0
01111	0
10000	1
10001	1
10010	1
10011	1
10100	1
10101	1
10110	1
10111	1
11000	1
11001	1
11010	1
11011	1
11100	1
11101	1
11111	1

$\overline{ABC\,DE}$

$\overline{A}\overline{B}CDE$

$\overline{A}\overline{B}\overline{C}DE$

$\overline{A}BCD\overline{E}$

$\overline{A}BCDE$

29

Number of sum terms: 5

$Y = (\bar{A} \vee B \vee C \vee \bar{D} \vee \bar{E}) \wedge (A \vee B \vee \bar{C} \vee \bar{D} \vee \bar{E})$ 29

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

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

$(A \vee \bar{B} \vee \bar{C} \vee \bar{D} \vee \bar{E})$

Question #3

The following 10-b word can be used to represent different numbers depending on the encoding

10b'11_1001_0001 $2^0+2^4+2^7+2^8+2^9=913$

(a) If the word is 2's complement, what is the corresponding integer? 913

(b) If the word is unsigned fixed point 5.5, what is the corresponding number?

28.17/32 = 5 11100.10001 $=2^4+2^3+2^2+2^{-1}+2^{-5}=$

What is the absolute accuracy of this representation? .015625 Interval/2 = absolute accuracy Interval = 2^{-5}

(c) What is this word in Binary Coded Decimal? 391 $\underbrace{2^1+2^0}_{3}$ $\underbrace{2^3+2^0}_{9}$ $\underbrace{2^0}_{1}$ $\underbrace{11}_{3}$ $\underbrace{1001}_{9}$ $\underbrace{0001}_{1}$

(d) If the word is 4E5 floating point number (IEEE format S+EEEEE+MMMM),

What is the bias? 15 $2^{(5-1)}-1=2^4-1=15$

E = 25 − 15

E = Exp − Bias

What is the corresponding real number? −1088 $1\ \underbrace{11001}_{Exp}\ \underbrace{0001}_{Mantissa}$ val × 2^{10} = 1.0001 × 2^{10} $=(1+2^{-4})\times 2^{10}$ = 1088

What is the relative accuracy as a % of this representation? 2.94117647%

(interval/2) / value = relative accuracy Interval = 64 (Interval/2) / Value = relative accuracy

(e) In the IEEE 754 floating point representation, what is the advantage of the implied 1 for the mantissa? It allows the floating point to represent larger numbers - due to the implied 1

What is the advantage of using a Bias in the exponent?

One can compensate for negative numbers represented in floating point; you can use the bias to calculate the E exponent when converting from floating pt to decimal

(f) In a binary 2's complement system, how do you determine if a number is negative?

If the first bit [Most Significant Bit] is a 1, it is negative

What is the logic for determining an underflow during an subtraction between 2 input words of n-bits, a[n-1:0] and b[n-1:0]?

If the carry bit resulting from a[n-1:0] − b[n-1:0] is a 1, then it is underflow, otherwise if it is a 0, it isn't an underflow.

(g) In base-5 system and using 3 "quinits" (base-5 digits) in 5's complement: 3 digits

How would one represent a base-10 integer -55? 240

−55

55 in 5's comp

210 = 55 in 5's comp

What is the most positive value in base-10 integer that can be represented?

62

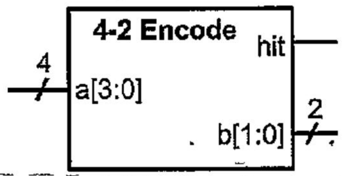
$5\times5\times5=\frac{125}{2}$ 62

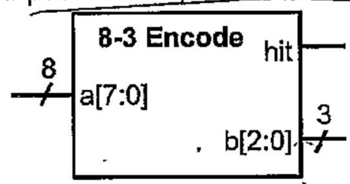
$5\times5=\frac{25}{2}=12.5$

444
−210
234 + 1 = 240 = −55

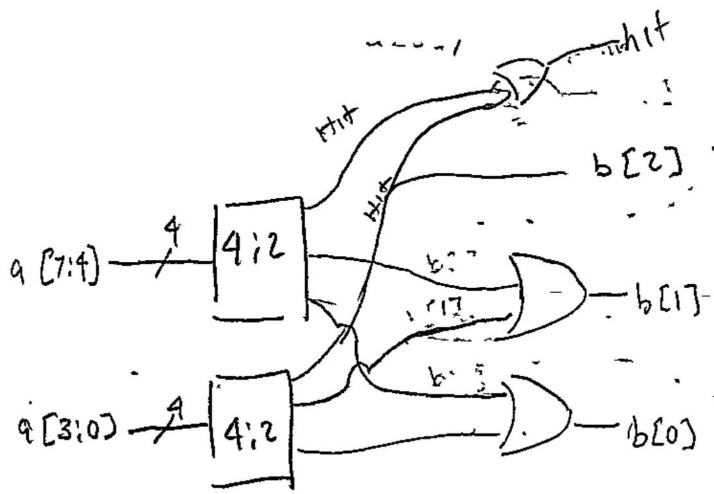
Question #4

A 4-to-2 encoder (from one-hot to binary) is shown as a block below. The inputs are 4 one-hot inputs, *a[3:0]*. The output are 2 binary bits, *b[1:0]*, indicating the position of the "hot" input bit. If the input has a hot bit, then *hit*=1'b1, otherwise, when all inputs are 0's, then *hit*=1'b0.





(a) Use instances of this 4-to-2 encoder block to design an 8-to-3 encoder. On-hot inputs are a[7:0], binary outputs are b[2:0], and a *hit* indicator. You may use additional OR, AND, or Inverters as needed but use as few as possible. - 2:1 Encoder

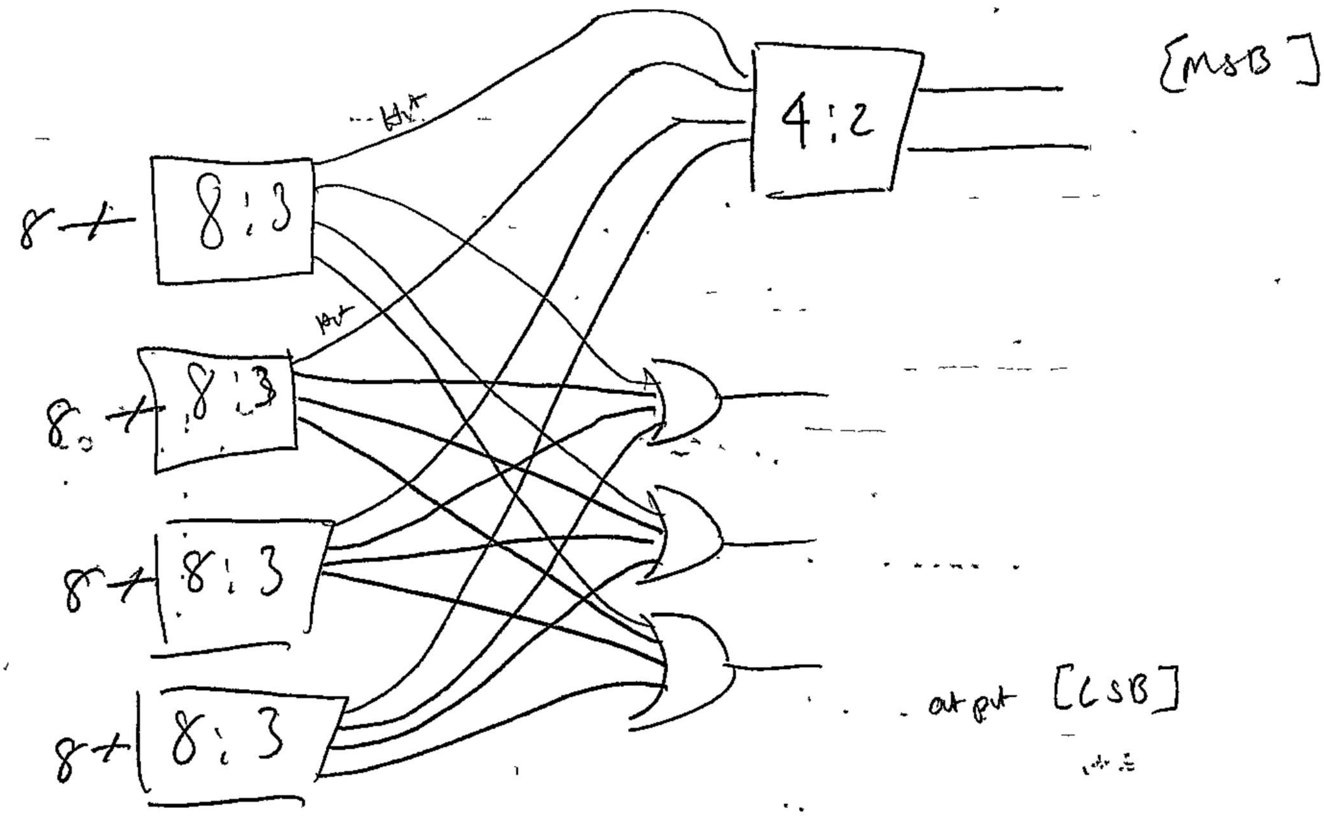


00000001	000
00000010	001
00000ϕ00	010
00001000	011
00010000	ϕ00
00100000	101
01000000	110
10000000	111

0001	00
0010	01
0100	10
1000	11

00
01
10
11

(b) Use 8-to-3 encoders and 4-to-2 encoders as modules and design a 32-5 encoder with as few modules and additional logic as possible. Again, you may use additional OR, AND, or Inverters as needed.



(c) Instead of an encoder that converts to binary, consider a converter that outputs thermometer code instead. For instance, an 8-bit one-hot input, *a[7:0]*, would convert to 7-bit thermometer output, *t[6:0]*. When *a[7:0]* = 8'0100_0000, *t[6:0]* = 7'b011_1111, and when *a[7:0]* = 8'b0000_0001, *t[6:0]* = 7'b000_0000. First write the Boolean expression for *t[3]* as a function of the a[7:0] inputs. Then design, using a **bit-cell approach**, the logic for each bit position. *Denote the inputs, outputs and signals passing between bit-positions clearly as well as the connections for the MSB and LSB positions.* Note that since the output has one fewer bit than the input, one of the bit-slices may have logic that is not used to produce an output. You may only use OR, AND, or Inverters as needed but use as few as possible.

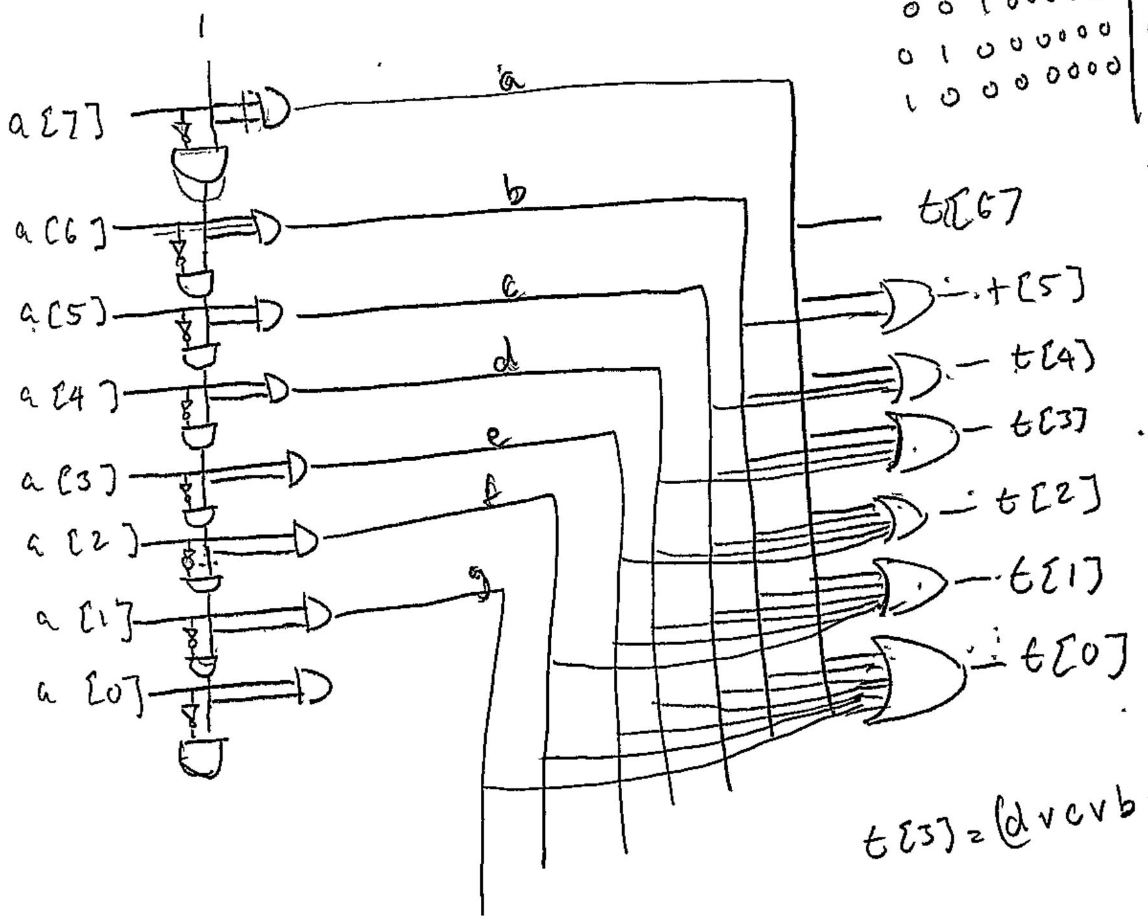
t[3] = $(a \vee b \vee c \vee d)$

$= (a[7]) \vee (a[6] \wedge \overline{a[7]}) \vee$

$((\overline{a[6] \wedge a[7]}) \wedge a[5]) \vee$

$(((a[6] \wedge a[7]) \overline{\wedge a[5]}) \wedge a[4])$

a[7] … a[0]	t[6] … t[0]
00000001	0000000
00000010	0000001
00000100	0000011
00001000	0000111
00010000	0001111
00100000	0011111
01000000	0111111
10000000	1111111



t[3] = $(d \vee c \vee b \vee a)$

Key:

$b = a[6] \wedge \overline{a[7]}$

$c = (\overline{a[6] \wedge a[7]}) \wedge a[5]$

$d = ((a[6] \wedge a[7]) \overline{\wedge a[5]}) \wedge a[4]$

$e = (((\overline{a[6] \wedge a[7]}) \wedge a[5]) \overline{\wedge a[4]}) \wedge (a[3]))$

Blank page for scratch work

Blank page for scratch work