Midterm November 5, 2020
Deadline October 6, 2020 by 2pm Pacific Time

Problem 1 (10 points)

How many bits are required to encode a color spectrum capable of supporting 16 million colors using:

a: Decimal digits in BCD
b: Hexadecimal representation

which representation is more efficient and why?

Problem 2 (10 points)

Fill in the missing entries

Radix	Value	Value in decimal
16	(5 1 7)	
8	(5 1 7)	

True or False:
a: A bubble on the output of a logic gate indicates that the output is active HIGH
b: Bubbles on the input lines of logic gates indicate that these inputs are active LOW
c: The Boolean expression for a logic circuit is A’ + B’ = X (A’ is complement A)
This tells us both inputs are active LOW and the output is active HIGH.

Problem 3 (10 points)

Given $E(a, b, c, d) = (ab + c)' (ac + (b' + c' + a'cd)') + a((b + c)(b + d) + c)'$, which of the following represents the same function as $E(a, b, c, d)$? Show all your work.

1. $a+b+c+d'$ 2. $a' + b + c$ 3. $b + c' + d$

4. $a'b'c'd$ 5. $ab'c'$ 6. $b'cd'$

Problem 4: (10 points)

a: Explain Gray Codes and their characteristics and explain conversion of GC to Binary with an example
b: What is X-3 code and explain the self complementary property of it.
c: Subtract $(-25)_8$ from $(25)_{10}$ using 2’s complement and explain the results in octal and decimal

d: realize XOR gate using NOR gates only.

Problem 5: (10 points)

Given the following simplification of a boolean expression, answer the following.

$$
\begin{aligned}
& (ab' + c')'(b' + c)(a + bc') && (1)\\
= \; & (ab')'c'(b' + c)(a + bc') && (2)\\
= \; & (a' + b)c'(b' + c)(a + bc') && (3)\\
= \; & (a' + b)c'(ab + ac + bb'c' + bcc') && (4)\\
= \; & (a' + b)c'(ab + ac) && (5)\\
= \; & (a' + b)(ab + ac)c' && (6)\\
= \; & (aa'b + abb + aa'c + abc)c' && (7)\\
= \; & abc' + abcc' && (8)\\
= \; & abc' && (9)
\end{aligned}
$$

a: (4 points) There is at least one mistake in this simplification. Find all steps that are derived incorrectly from its previous step (for example, write (8)→(9) if equation (9) is derived incorrectly from (8)).

b: (6 points) Show the correct simplification of (1)

Problem 6: (20 points)

a: Express the given expression in a product of maxterms:

F = xy+ x’z

b: simplify the following boolean function and realize it using NOR gates

F = (A,B,C,D) = A’B’D’ + A’CD + A’BC
d(A,B,C,D) = A’BC’D + ACD + AB’D’

c: simplify the Boolean function in SOP and POS form
F = $\sum$ (0,1,2,5,8,9,10)

Problem 7: (20 points)

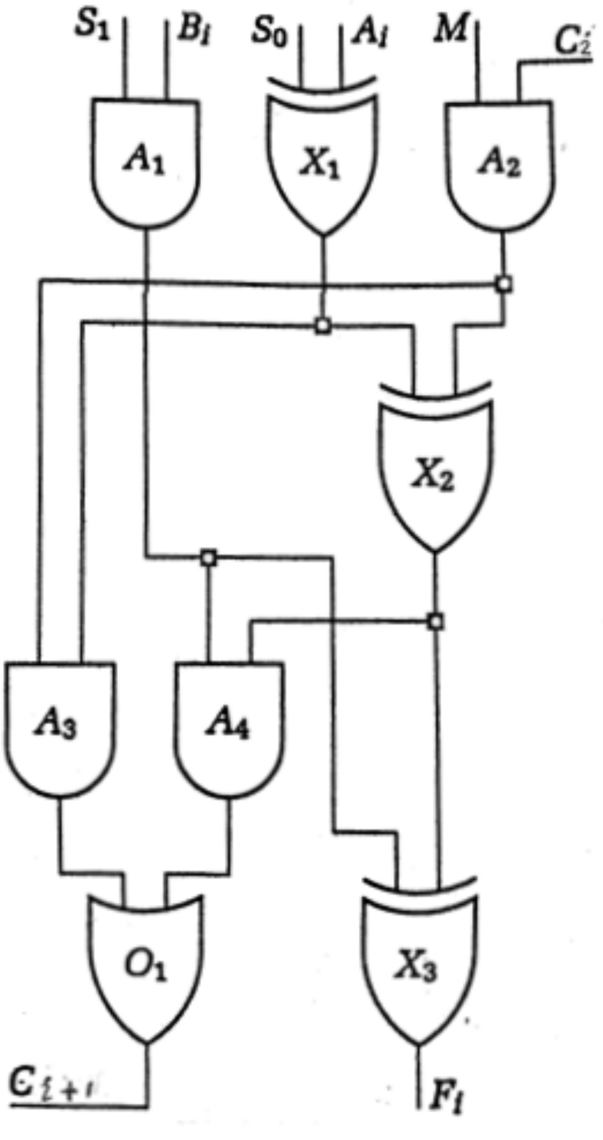
a: Draw the logic diagram for a four-bit comparator and explain the ≥ block.

b: explain the BCD-to-7-segment decoding logic for “g” segment

c: explain Decimal to BCD priority encoder, draw it partially and show the truth table for inputs 3 and 6.

Problem 8: (10 points)

Consider the following gate network:



Determine the switching expressions for the outputs F_i and C_{i+1}. Using AND, OR, NOT, XOR expressions as appropriate fill in the following table:

M	S_1S_0	Expression for F_i	Expression for $CI + 1$
0	00		
0	01		
0	10		
1	10		
1	11		

#1. 16 million colors.

a: Decimal digits in BCD.

↳ needs 4 digits for each decimal digits.

To express 16million colors using BCD,

we need $8 \text{ decimal digits} \times \frac{4 \text{ BCD digits}}{1 \text{ decimal digit}} = \boxed{32 \text{ digits in BCD}}$

b: Hexadecimal representation.

↳ each digit in hexadecimal can represent 16 numbers

to express 16 million colors using hexadecimal,

$16^n \geq 16\times10^6$, n: number of digits in hexadecimal representation

$16^6 = 16{,}777{,}216$
$16^5 = 1{,}048{,}576$) → $\boxed{\text{we need 6 digits in Hexadecimal}}$

⇒ b: Hexadecimal representation is more efficient because we only need 6 digits to represent 16 million colors.

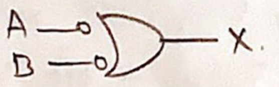
#2.

Radix	Value	Value in decimal
16	(517)	$7\times16^0+1\times16^1+5\times16^2=\boxed{1303}$
8	(517)	$7\times8^0+1\times8^1+5\times8^2=\boxed{335}$

a: False – bubble on the output indicates the output is active low.

b: True

c: True.



#3. $E(a,b,c,d) = \overline{(ab+c)}(ac+(\overline{\bar b+\bar c+\bar a cd})) + a(\overline{(b+c)(b+d)+c})$.

$= \overline{ab}\cdot\bar c\,(ac + b\cdot c\cdot\overline{\bar a cd}) + a(\overline{(b+c)(b+c+d)})$

$= (\bar a+\bar b)\bar c\,(ac+bc(a+\overline{cd})) + a\,(\overline{(b+c)}+\overline{(b+c+d)})$

$= (\bar a\bar c+\bar b\bar c)\,(ac+abc+bc(\bar c+\bar d)) + a(\bar b\bar c+\bar b\bar c\bar d)$

$= (\bar a\bar c+\bar b\bar c)\,(ac+abc+bc\bar d) + a\bar b\bar c + a\bar b\bar c\bar d$

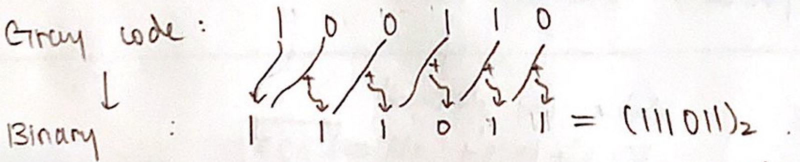
$= \bar a\bar c\cancel{a}c + \bar a\bar c\cancel{a}bc + \bar a\bar c\cancel{b}c\bar d + \bar b\bar c\cancel{a}c + \bar b\bar c\cancel{a}bc + \bar b\bar c\cancel{b}c\bar d + a\bar b\bar c + a\bar b\bar c\bar d$

$= a\bar b\bar c + a\bar b\bar c\bar d$

$= a\bar b\bar c$ $\boxed{5}$

#4. a: Gray code has a characteristic that adjacent numbers have a single digit difference, so it is used as an error correction application.

To convert a gray code to binary, we use a method called "moving sum between Binary bit and Gray bit," and an example follows:



b: X-3 code is a code that we can get by adding 3 to each 4 digits of BCD code that represents a decimal digit.

The self-complement property is that the 1's compliment of an X-3 code equals to the X-3 code of the 9's complement of the corresponding decimal number.

For example, X-3 code of decimal number 5 is 1000 and 1's complement of this is 0111. 9's complement of 5 is 4 and its X-3 code is 0111, which is the same as 1's complement of the excess code 1000.

c. $(25)_{10} - (-25)_8$

(i) using 2's complement.

$$(25)_{10} - (-25)_8 = (\underset{\uparrow}{0}011001)_2 - (-(\underset{\uparrow}{0}010101)_2)$$

sign bit sign bit

$2^4+2^3+2^0$

$$= (\underset{\uparrow}{0}011001)_2 - (\underset{\uparrow}{1}101011)_2$$

$$= (\underset{\uparrow}{0}011001)_2 + (\underset{\uparrow}{0}010101)_2$$

$$= (\underset{\uparrow}{0}101110)_2 = \boxed{(56)_8} \leftarrow \text{octal}$$

$$= 2^5+2^3+2^2+2^1 = \boxed{(46)_{10}.} \leftarrow \text{decimal.}$$

$$\begin{array}{r} 0011001 \\ +\ 0010101 \\ \hline 0101110 \end{array}$$

(ii) In octal, $(25)_{10} - (-25)_8 = (31)_8 - (-25)_8$

$3 \times 8^1 + 8^0$

$$= (31)_8 + (25)_8$$

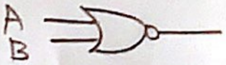
$$= \boxed{(56)_8.}$$

(iii) In decimal $(25)_{10} - (-25)_8 = (25)_{10} - (-21)_{10}$

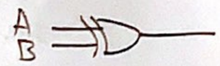
$$= (25)_{10} + (21)_{10}$$

$$= \boxed{(46)_{10}}$$

d. XOR Gates using NOR Gates only

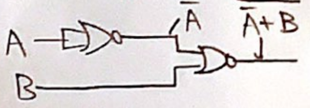


$\overline{A+B}$: NOR Gate.

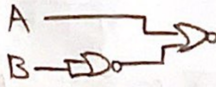


$A \oplus B = \overline{A}B + A\overline{B}$: XOR Gate.

(i) Make $\overline{A}B$ and $A\overline{B}$ separately

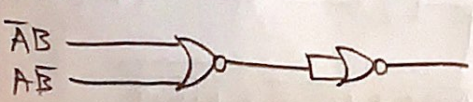


$\overline{\overline{A}+B} = A\overline{B}$



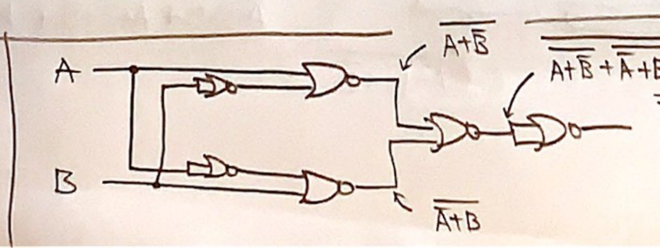
$\overline{A+\overline{B}} = \overline{A}B.$

(ii) $\overline{A}B + A\overline{B}$



$\overline{\overline{\overline{A}B+A\overline{B}}} = \overline{A}B+A\overline{B}$

(iii) XOR Gate


A
B
$\overline{A+\overline{B}}$
$\overline{\overline{A}+B}$
$\overline{\overline{A+\overline{B}}+\overline{\overline{A}+B}}$

$\overline{\overline{\overline{A+\overline{B}}+\overline{\overline{A}+B}}} = \overline{A}B+A\overline{B} = A\oplus B.$

#5 a: (1)→(2): $(ab'+c')'(b'+c)(a+bc') = (ab')'\underline{c''}(b'+c)(a+bc')$

(3)→(4): $(a'+b)c'(b'+c)(a+bc') = (a'+b)c'(\underline{ab'}+ac+bb'c'+bcc')$

b: $\overline{(a\bar{b}+\bar{c})}(\bar{b}+c)(a+b\bar{c}) = \overline{a\bar{b}}\cdot c(\bar{b}+c)(a+b\bar{c})$

$= (\bar{a}+b)c(a\bar{b}+ac+\bar{b}\cancel{b}\bar{c}+b\cancel{\bar{c}}c)$

$= (\bar{a}+b)(a\bar{b}c+ac)$

$= \bar{a}\cancel{a}\bar{b}c+a\cancel{b}\bar{b}c+\bar{a}\cancel{a}c+abc$

$= abc.$

#6 a: $F = xy+\bar{x}z = (xy+\bar{x})(xy+z)$

$= (x+\bar{x})(\bar{x}+y)(x+z)(x+y)$

$= (\bar{x}+y+z\bar{z})(x+y\bar{y}+z)(x+y+z\bar{z})$

$= (\bar{x}+y+z)(\bar{x}+y+\bar{z})(x+y+z)(x+\bar{y}+z)(x+\cancel{y+}z)(x+y+\bar{z})$

$= \boxed{(x+y+z)(\bar{x}+y+z)(x+\bar{y}+z)(x+y+\bar{z})(\bar{x}+y+\bar{z})}$

b: $F(A,B,C,D) = \bar{A}\bar{B}\bar{D}+\bar{A}CD+\bar{A}BC$

$d(A,B,C,D) = \bar{A}B\bar{C}D+ACD+A\bar{B}\bar{D}$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1		1	1
$\bar{A}B$		X_0	1	1
AB			X_0	
$A\bar{B}$	X_1		X_0	X_1

$Y = \bar{A}C+\bar{B}\bar{D}$

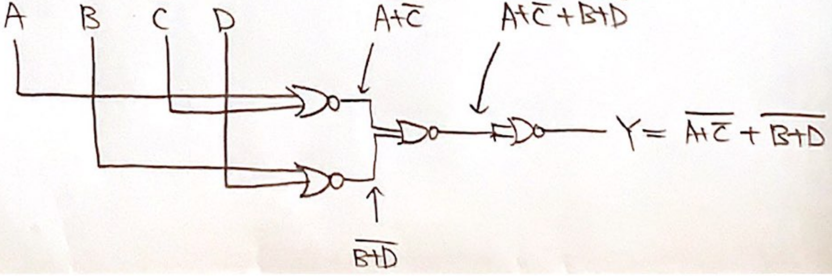
$\bar{Y} = \overline{\bar{A}C+\bar{B}\bar{D}}$

$= \overline{\bar{A}C}\cdot\overline{\bar{B}\bar{D}}$

$= (A+\bar{C})(B+D).$

$Y = \overline{(A+\bar{C})(B+D)}$

$= \overline{A+\bar{C}}+\overline{B+D}$


A
B
$\bar{C}$
D
$\overline{A+\bar{C}}$
$\overline{\overline{A+\bar{C}}+\overline{B+D}}$
$Y = \overline{A+\bar{C}}+\overline{B+D}$
$\overline{B+D}$

c: $F = \Sigma(0,1,2,5,8,9,10)$

(i) In SOP

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	1
$\bar{A}B$	0	1	0	0
AB	0	0	0	0
$A\bar{B}$	1	1	0	1

$$\boxed{F = \bar{B}\bar{D} + \bar{B}\bar{C} + \bar{A}\bar{C}D}$$

(ii) In POS, $\bar{F} = \Pi(3,4,6,7,11,12,13,14,15)$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	1
$\bar{A}B$	0	1	0	0
AB	0	0	0	0
$A\bar{B}$	1	1	0	1

$$\bar{F} = B\bar{D} + AB + CD$$

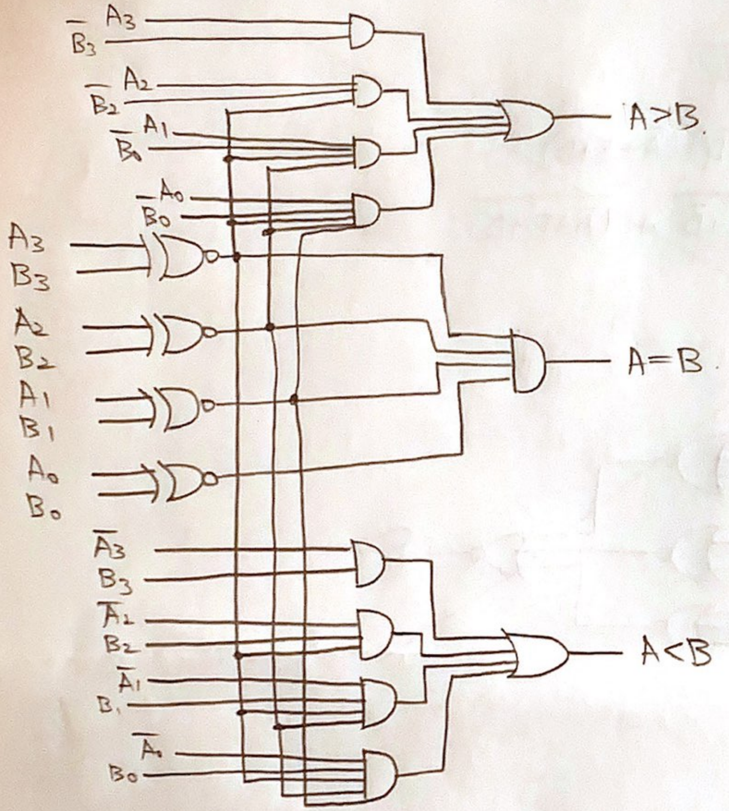
$$F = \overline{B\bar{D} + AB + CD}$$

$$= \overline{B\bar{D}} \cdot \overline{AB} \cdot \overline{CD}$$

$$\boxed{F = (\bar{B} + D)(\bar{A} + \bar{B})(\bar{C} + \bar{D})}$$

#7.

a: 4bit : $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$.

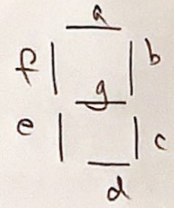


A>B block compares each digit and output 1 if A>B.

This block compares from MSB to LSB and if $A_3>B_3$, rest of the digits don't need to be compared because $A_3>B_3$ means entire A is greater than B.

A=B block outputs 1 if A=B.

Each XNOR gate that compares each digit of A and B outputs 1 if each digit is the same. And AND gate accomodates the outputs from all XNOR gates and outputs 1 if every input to this gate is 1 or 0 for everyother cases b: g segment



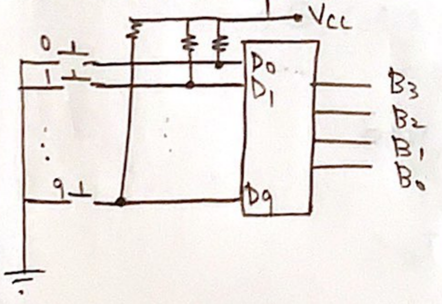
g segment will be activated for decimal numbers: 2, 3, 4, 5, 6, 8, 9.

So with BCD inputs D, B, C, A, segment g can be represented with minterms,

$$g = \overline{D}\,\overline{C}B\overline{A} + \overline{D}\,\overline{C}BA + \overline{D}C\overline{B}\,\overline{A} + \overline{D}C\overline{B}A + \overline{D}CB\overline{A} + D\overline{C}\,\overline{B}\,\overline{A} + D\overline{C}\,\overline{B}A$$
$$\quad \overset{\parallel}{2} \quad \overset{\parallel}{3} \quad \overset{\parallel}{4} \quad \overset{\parallel}{5} \quad \overset{\parallel}{6} \quad \overset{\parallel}{8} \quad \overset{\parallel}{9}.$$

if any of the minterms is 1, the output is 0 (because it's active low output) and g segment will light up.

c: Decimal to BCD priority encoder is an encoder that can prioritize the input when there are multiple inputs. The higher decimal number has priority than lower decimal numbers, so if the encoder gets more than 1 input, it automatically encodes the highest number only and ignore the lower decimal number inputs.



	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	B_3	B_2	B_1	B_0
	⋮										⋮			
Input3	X	X	X	L	H	H	H	H	H	H	H	H	L	L
				⋮							⋮			
input6	X	X	X	X	X	X	L	H	H	H	H	L	L	H
				⋮							⋮			

input 6 don't care the lower numbers.

#8. $F_i = \left[MC_i \oplus (S_0 \oplus A_i) \right] \oplus S_1 B_i$

$C_{i+1} = MC_i \cdot (S_0 \oplus A_i) + S_1 B_i (MC_i \oplus (S_0 \oplus A_i))$

	M	$S_1 S_0$	F_i	C_{i+1}
①	0	00	A_i	0
②	0	01	$\overline{A_i}$	0.
③	0	10	$A_i \oplus B_i$	$A_i B_i$
④	1	10	$(C_i \oplus A_i) \oplus B_i$	$C_i A_i + B_i (C_i \oplus A_i)$
⑤	1	11	$(C_i \oplus \overline{A_i}) \oplus B_i$	$C_i \overline{A_i} + B_i (C_i \oplus \overline{A_i})$

① $F_i = [0 \oplus (0 \oplus A_i)] \oplus 0 = [0 \oplus (0\overline{A_i} + 1 \cdot A_i)] \oplus 0$

$= [0 \oplus A_i] \oplus 0$

$= (0\overline{A_i} + 1 \cdot A_i) \oplus 0$

$= A_i \oplus 0$

$= \overline{A_i} 0 + A_i 1$

$= A_i$

$C_{i+1} = 0 \cdot (0 \oplus A_i) + 0 \cdot (0 \oplus (0 \oplus A_i)) = 0.$

② $F_i = [0 \oplus (1 \oplus A_i)] \oplus 0 = [0 \oplus (1\overline{A_i} + 0 \cdot A_i)] \oplus 0$

$= (0 \oplus \overline{A_i}) \oplus 0$

$= (0 A_i + 1 \cdot \overline{A_i}) \oplus 0$

$= (\overline{A_i} \oplus 0)$

$= A_i \cdot 0 + \overline{A_i} \cdot 1$

$= \overline{A_i}$

$C_{i+1} = 0 \cdot (1 \oplus A_i) + 0 (0 \oplus (1 \oplus A_i)) = 0$

③

M	S_1	S_0
0	1	0

$$F_i = [0 \oplus (0 \oplus A_i)] \oplus B_1$$
$$= [0 \oplus (0\bar{A_i} + 1 A_i)] \oplus B_i$$
$$= (0 \oplus A_i) \oplus B_i$$
$$= (0\bar{A_i} + 1A_i) \oplus B_i$$
$$= A_i \oplus B_i$$

$$C_{i+1} = 0(0 \oplus A_i) + 1 \cdot B_i(0 \oplus (0 \oplus A_i))$$
$$= B_i(0 \oplus A_i)$$
$$= A_i B_i$$

④

M	S_1	S_0
1	1	0

$$F_i = [C_i \oplus (0 \oplus A_i)] \oplus B_i$$
$$= (C_i \oplus A_i) \oplus B_i$$

$$C_{i+1} = C_i(0 \oplus A_i) + B_i(C_i \oplus (0 \oplus A_i))$$
$$= C_i A_i + B(C_i \oplus A_i).$$

⑤

M	S_1	S_0
1	1	1

$$F_i = [C_i \oplus (1 \oplus A_i)] \oplus B_i$$
$$= (C_i \oplus \bar{A_i}) \oplus B_i$$

$$C_{i+1} = C_i(1 \oplus A_i) + B_i(C_i \oplus (1 \oplus A_i))$$
$$= C_i \bar{A_i} + B_i(C_i \oplus \bar{A_i})$$