EE 109 Homework 3

Redekopp

Name: ___Solutions_____

Due: **Score: _____**

Show work to get full credit. Remember, use on only one side of the paper and staple them together. Only use a calculator to CHECK your work, not to DO your work.

1.a F = (A'+B')(C)

ABC	A'	B'	A' + B'	C	F
000	1	1	1	0	0
001	1	1	1	1	1
010	1	0	1	0	0
011	1	0	1	1	1
100	0	1	1	0	0
101	0	1	1	1	1
110	0	0	0	0	0
111	0	0	0	1	0

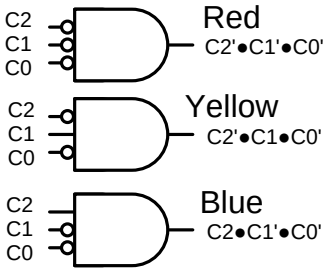
1.b G = WX' + XY' + W'

WXY	X'	W • X'	Y'	X • Y'	W'	G
000	1	0	1	0	1	1
001	1	0	0	0	1	1
010	0	0	1	1	1	1
011	0	0	0	0	1	1
100	1	1	1	0	0	1
101	1	1	0	0	0	1
110	0	0	1	1	0	1
111	0	0	0	0	0	0

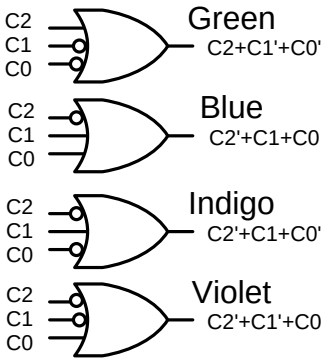
2. **[BB]** (14 pts.) Implement the following functions

	Color Bit Combinations		
Meaning	**C2**	**C1**	**C0**
Red	0	0	0
Orange	0	0	1
Yellow	0	1	0
Green	0	1	1
Blue	1	0	0
Indigo	1	0	1
Violet	1	1	0
Unused	1	1	1

a. Build appropriate single gate (+ inverters) checker/decoders that outputs a logic '1' for each of the three primary colors: **Red, Blue, Yellow** (i.e. one decoder to check for each color)

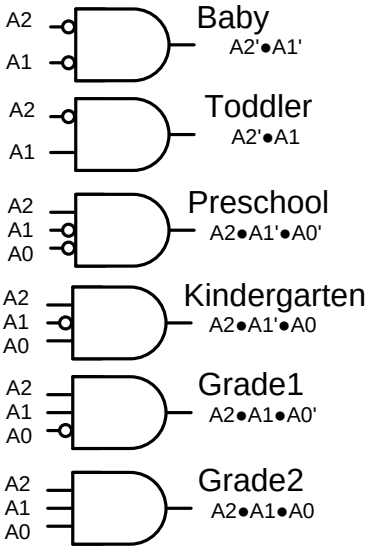


b. Build an appropriate single gate (+inverters) checker/decoder that outputs a logic '0' for each color: **Green, Blue, Indigo, Violet**.



3. **[BB]** (12 pts.) Given a child's age [0-7 years old], design logic to produce the following outputs: {**Baby, Toddler, Preschool, Kinder, Grade1, Grade2**} which have the following age correspondence/mapping. Use only 1 AND gate (and any number of inverters) per output.

	Age Combinations		
Meaning	**A2**	**A1**	**A0**
Baby	0	0	0
	0	0	1
Toddler	0	1	0
	0	1	1
Preschool	1	0	0
Kindergarten	1	0	1
Grade1	1	1	0
Grade2	1	1	1



4. **$H = \Sigma_{A[3:0]}(0,3,6,9,12,15) =$**
   A3'A2'A1'A0' + A3'A2'A1A0 + A3'A2A1A0' + A3A2'A1'A0 + A3A2A1'A0' + A3A2A1A0

5. **$G = \Sigma_{ABCD}(2,3,6,13,15)$**
   = A'B'CD' + A'B'CD + A'BCD' + ABC'D + ABCD

6. **G = A'B'CD' + A'B'CD + A'BCD' + ABC'D + ABCD = m2 + m3 + m6 + m13 + m15**
   Notice m2 and m3 differ by 1 variable, m2 and m6 differ by 1 variable, and m13 and m15 differ by 1 variable…We can replicate m2 (because T3: X+X = X) to use m2 to help us with m3 and m6

= **(m2+m3) + (m2+m6) + (m13+m15)**
= **(A'B'CD' + A'B'CD) + (A'B'CD' + A'BCD') + (ABC'D + ABCD)**
= **A'B'C(D'+D) + A'CD'(B'+B) + ABD(C'+C)**
= **A'B'C + A'CD' + ABD**

7. a.

F = X+[(W'Y'Z)(W+(X'(Y+Z)))]
= X+[W'Y'ZW + W'Y'ZX'(Y+Z)] T8 Distributing
= X+[W'Y'ZW + W'Y'ZX'Y + W'Y'ZX'Z] T8 Distributing
= X+[0•Y'Z + W'ZX'•0 + W'Y'ZX'] T5' (W'W = 0 and Y'Y = 0) & T3'
= X+[0 + 0 + W'X'Y'Z] T2'
= X+[W'X'Y'Z] T1
= (X+W')(X+X')(X+Y')(X+Z) T8' (Distributing + over •)
= (X+W')(X+Y')(X+Z) T5, T1'

b

$G = \overline{(w+x+\bar{y})} + w\bar{x}y + (w\bar{z}+x\bar{y})\cdot\overline{(x(\bar{w}+z))}$

$= \bar{w}\cdot\overline{(x+\bar{y})} + w\bar{x}y + (w\bar{z}+x\bar{y})\cdot(\bar{x}+\overline{(\bar{w}+z)})$ -- DeMorgan's theorem

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

$= \bar{w}\bar{x}y + w\bar{x}y + w\bar{x}\bar{z} + x\bar{x}\bar{y} + ww\bar{z}\bar{z} + wx\bar{y}\bar{z}$

$= (w+\bar{w})\bar{x}y + w\bar{x}\bar{z} + 0 + w\bar{z} + wx\bar{y}\bar{z}$ -- $x\bar{x} = 0$

$= \bar{x}y + w\bar{x}\bar{z} + w\bar{z} + wx\bar{y}\bar{z}$

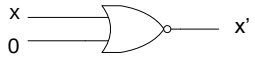
$= \bar{x}y + w\bar{z}(\bar{x}+1) + wx\bar{y}\bar{z}$ -- $1+\bar{x} = 1$

$= \bar{x}y + w\bar{z} + wx\bar{y}\bar{z}$

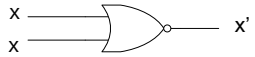
$= \bar{x}y + w\bar{z}(1+x\bar{y})$ -- $1+x\bar{y} = 1$

$= \bar{x}y + w\bar{z}$

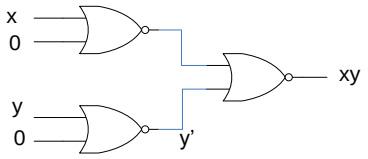
8.a x’ = x NOR 0



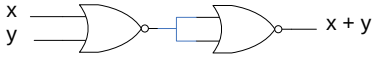
or inversion can be achieved as



8.b According to DeMorgan’s theorem, $x \cdot y = \overline{(\bar{x} + \bar{y})}$

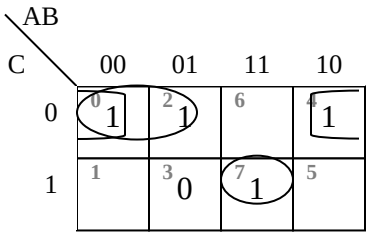


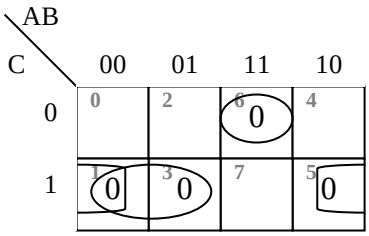
8.c $x + y = \overline{\overline{(x + y)}}$



8.d Yes. 8.a, 8.b, and 8.c can all be accomplished only using NAND gates in a similar fashion.

9.a

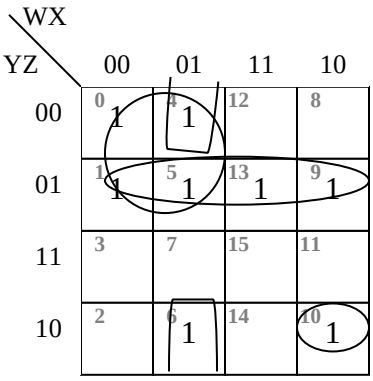




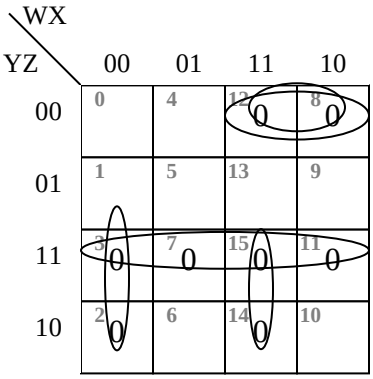
SOP = B’C’ + A’C’ + ABC

POS = (B+C’)(A+C’)(A’+B’+C)

9.b

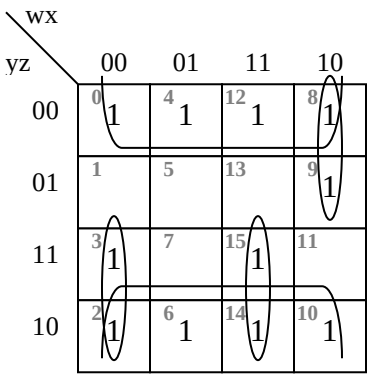


SOP = $\overline{Y}Z + \overline{W}\overline{Y} + \overline{W}X\overline{Z} + W\overline{X}Y\overline{Z}$

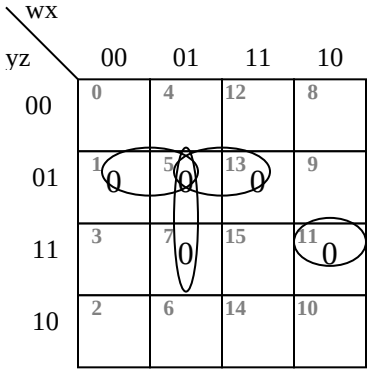


POS = $(\overline{Y} + \overline{Z})(W + X + \overline{Y})(\overline{W} + \overline{X} + \overline{Y})(\overline{W} + Y + Z)$

9.c



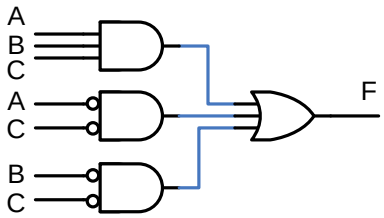
SOP = $\bar{z} + \bar{w}\bar{x}y + wxy + w\bar{x}\bar{y}$



POS = $(W + Y + \bar{Z})(\bar{X} + Y + \bar{Z})(W + \bar{X} + \bar{Z})(\bar{W} + X + \bar{Y} + \bar{Z})$

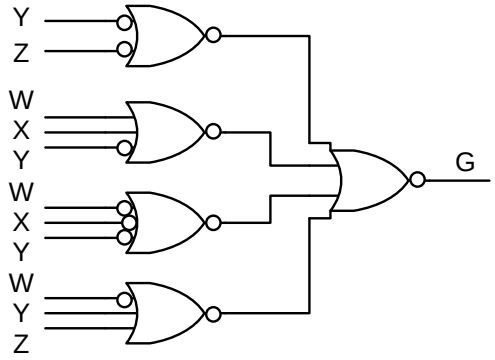
10.a (a) SOP = $\overline{A}B + \overline{B}\overline{C} + BC$ shown, but alternately: B'C' + A'C' + BC

- AND-OR implementation



10b.

- NOR-NOR implementation



10c.

- NAND-NAND implementation