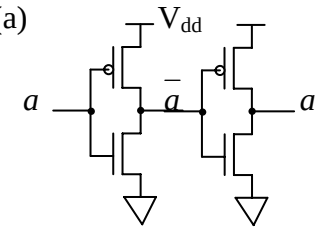
EECS 203 HW3 Solution

3. (10 pts.) CMOS review

(a)



(b) No, there is no current flowing in the circuit.

(c) No. The circuit will be ok. If a is 0, intermediate $\bar{a}$ will be 1. Then NMOS of the 2nd inverter is switched on. Therefore connect output to ground will be safe.

(d) Yes. There will be current from V_{DD} to output and to ground (V_{SS}).

(e) Yes the circuit will be probably destroyed, because there is current flowing from V_{DD} to V_{SS}.

4. (5 pts.)

The XOR function. (Also accepted are the XNOR, Parity, or Odd Functions)

$$f(i_{n-1}, i_{n-2}, ..., i_1, i_0) = i_{n-1} \oplus i_{n-2} \oplus .... \oplus i_1 \oplus i_0$$

Hence, when you have N inputs, you will have the following equation:

$N*2^{N-1} >= 2*2^{N-1} = 2^N$

5. (5 pts.)

Inversely correlated.

6. (5 pts.)

(a) $f(a,b,c,d) = \prod(4,8,10,13)+d(5,14)$

$f(a,b,c,d) = \overline{\sum(4,8,10,13)} + d(5,14)$

From K-map below, $\boxed{f(a,b,c,d) = \bar{a}\bar{b} + \bar{b}d + bc + ab\bar{d}}$

ab \ cd	00	01	11	10
00	1	1	1	1
01	0	X	1	1
11	1	0	1	X
10	0	1	1	0

7. (5 pts.)

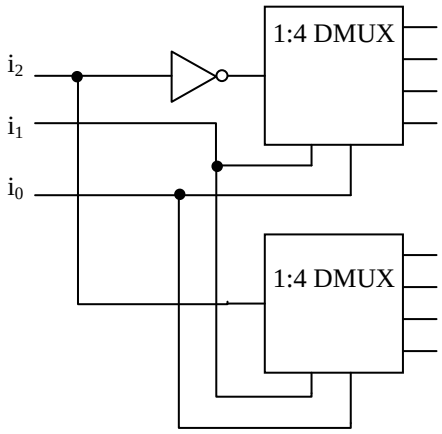
(a) $f(a,b,c,d) = \Sigma(0,3,5,7,8,13)+d(9,10,11,12,15)$

By collecting zeros from the k-map below, we see $\bar{f}(a,b,c,d) = b\bar{d} + c\bar{d} + \bar{b}\bar{c}d$. Therefore,

$\boxed{f(a,b,c,d) = (\bar{b} + d)(\bar{c} + d)(b + c + \bar{d})}$.

ab \ cd	00	01	11	10
00	1	0	1	0
01	0	1	1	0
11	X	1	X	0
10	1	X	X	X

8. (5 pts.)

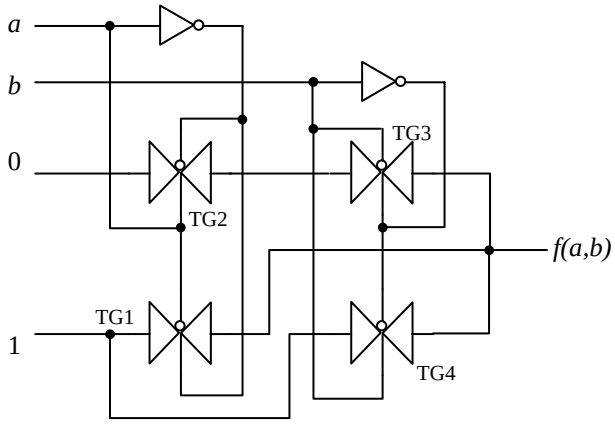


9. (10 pts.)

(a) First simplify function:

$$f(a,b) = \overline{a} + ab$$
$$= (\overline{a} + a)(\overline{a} + b)$$
$$= \overline{a} + b$$

The schematic diagram can be implemented as:



A truth table of *f(a,b)* is,

a	*b*	TG1	TG2	TG3	TG4	*f*
0	0	ON	OFF	ON	OFF	1
0	1	ON	OFF	OFF	ON	1
1	0	OFF	ON	ON	OFF	0
1	1	OFF	ON	OFF	ON	1

(b) $j(a,b) = f(a,b) = \overline{a} + b$

(c) $k(a,b) = \overline{f(a,b)} = a\overline{b}$

(d) $j(a,b)k(a,b) = (\overline{a} + b)(a\overline{b}) = \overline{a}ab + a\overline{b}b = 0$

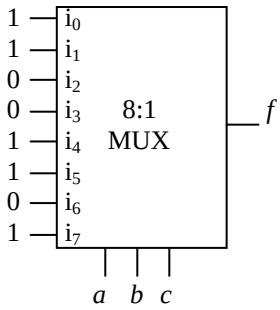
(e) $j(a,b) + k(a,b) = \overline{a} + b + a\overline{b} = \overline{a} + (b + a)(b + \overline{b}) = \overline{a} + b + a = b + 1 = 1$

(f) One PMOS each for TG1 and TG4. One NMOS each for TG2 and TG3. Therefore the total number of transistors necessary is 4 (excluding the transistors for the inverters).

10. (5 pts.)
f (*a*,*b*,*c*) = *ac*+ *b*'
Truth table of function *f(a,b,c)* is:

a	*b*	*c*	*f(a,b,c)*
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

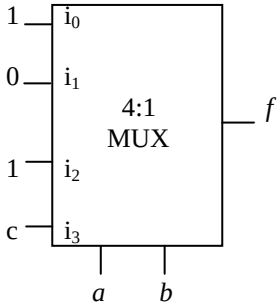
Schematic diagram using 8:1 MUX:



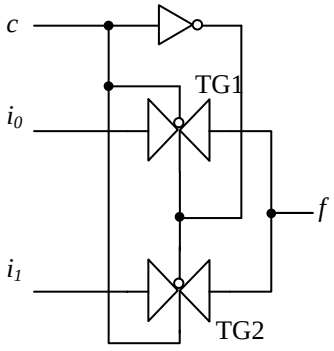
11. (10 pts.)
f (*a*,*b*,*c*) = *ac*+ *b*'
Consider *a*, *b* or *c* as input of the 4:1 MUX.
Then one possible truth table is:

a	*b*	*c*	*f(a,b,c)*
0	0	0	1
0	0	1	
0	1	0	0
0	1	1	
1	0	0	1
1	0	1	
1	1	0	c
1	1	1	

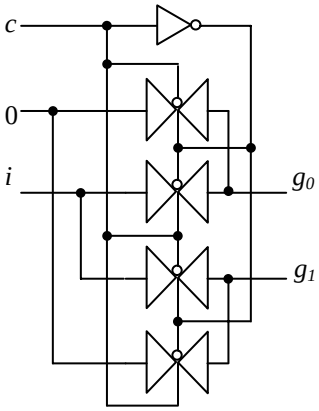
Schematic diagram using 4:1 MUX:



12. (10 pts.) MUX and DMUX review
(a)

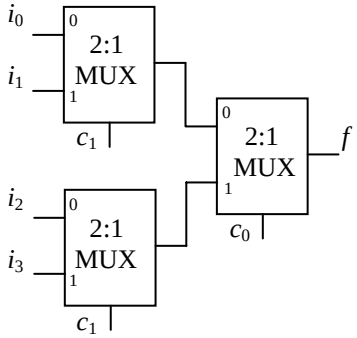


(c)



(b)
1) In some situations the output of TGs might be undefined.
2) It is not so easy to identify whether the output of a TG is high or low, so some times it might be possible that a high is connected to low.

(d) Treating your 2:1 MUX as a component (block in the circuit), build a 4:1 MUX.



13. (10 pts.) DMUX Logic: Use a 1:8 DMUX and OR/NOR gates to implement the following functions. Try to use as few inputs on your OR/NOR gates as possible.

(a)

$\boxed{f_a(a,b,c) = \bar{a}b\bar{c} + ab\bar{c} + abc}$ OR gate

(b)

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

$f_b(a,b,c) = \overline{\overline{(\bar{a} + \bar{b} + c)} + \overline{(a + b + c)}}$

$\boxed{f_b(a,b,c) = \overline{a\bar{b}c + \bar{a}\bar{b}\bar{c}}}$ NOR gate

Truth table for these functions:

abc	f_a	f_b	f_c
000	0	0	1
001	0	1	1
010	1	1	1
011	0	1	1
100	0	1	0
101	0	1	1
110	1	0	0
111	1	1	0

(c)

$f_c(a,b,c) = \bar{a} + \bar{b}c \Rightarrow \overline{f_c(a,b,c)} = \overline{\bar{a} + \bar{b}c}$

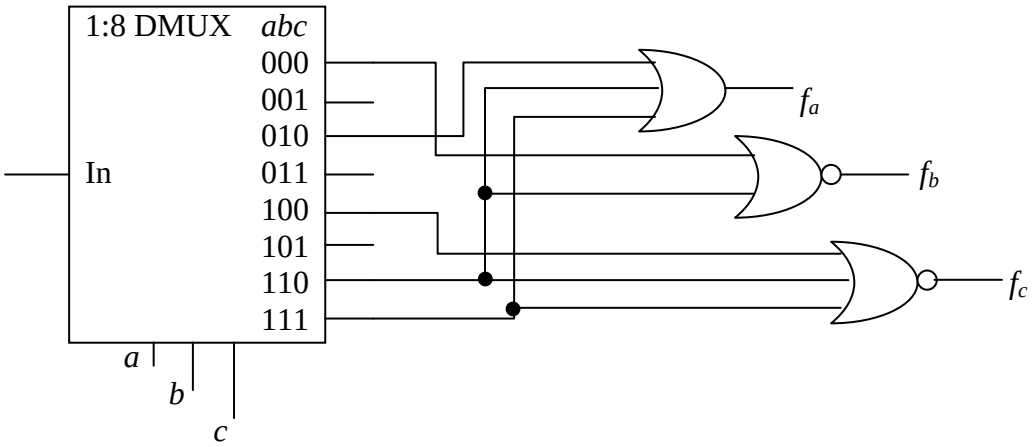
$\overline{f_c(a,b,c)} = \bar{\bar{a}}(\bar{\bar{b}}\bar{c}) = a(b + \bar{c})$

$\overline{f_c(a,b,c)} = ab + a\bar{c} = ab(c + \bar{c}) + a(b + \bar{b})\bar{c}$

$\overline{f_c(a,b,c)} = abc + ab\bar{c} + ab\bar{c} + a\bar{b}\bar{c}$

$\overline{f_c(a,b,c)} = abc + ab\bar{c} + a\bar{b}\bar{c}$

$\boxed{f_c(a,b,c) = \overline{abc + ab\bar{c} + a\bar{b}\bar{c}}}$ NOR gate



14. (10 pts.) Encoder review

Use Karnaugh Maps to design a three-input priority encoder, $f(i_2, i_1, i_0)$ for which i_2 has the highest priority. Reserve one code for $\overline{i_2}\,\overline{i_1}\,\overline{i_0}$.

(a) Write the minimized SOP expression for each encoded signal.

Truth Table:

i_2	i_1	i_0	o_1	o_0
0	0	0	0	0
0	0	1	0	1
0	1	x	1	0
1	x	x	1	1

K-maps:

For o_1:

i_2 \ i_1i_0	00	01	11	10
0	0	0	1	1
1	1	1	1	1

Therefore, $o_1=i_2+i_1$

For o_0:

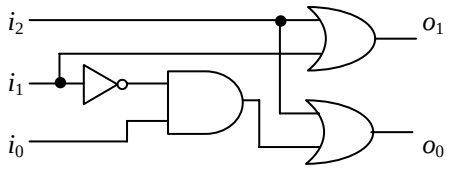
i_2 \ i_1i_0	00	01	11	10
0	0	1	0	0
1	1	1	1	1

Therefore, $o_0 = i_2 + \overline{i_1}\,i_0$

(b) Write the minimized POS expression for each encoded signal. You may start from the same Karnaugh maps used in (a) to save time.

Looking at 0's in o_1 k-map, we get $o_1 = \overline{\overline{i_2}\,\overline{i_1}} = i_2 + i_1$ and $o_0 = \overline{\overline{i_2}\,i_0 + \overline{i_2}\,i_1} = (i_2 + \overline{i_1})(i_2 + i_0)$

(c) Based on the expression with fewer literals, derive a schematic for the encoder.



15. (10 pts.) Number systems

(a) Convert 0xBADF00D to binary. Note: This is easier if you it one base-16 digit at a time instead of considering the whole number at once.

0xBADF00D=1011 1010 1101 1111 0000 0000 1101

B A D F 0 0 D

(b) Convert $(302)_8$ to base-10:

$(302)_8=3x8^2+0x8^1+2x8^0=(194)_{10}$

Convert $(194)_{10}$ to base-8:

Residual	factor#	base#
194	3	8^2
2	0	8^1
2	2	8^0

So $(194)_{10}=(302)_8$

(c) Convert the following ASCII codes to characters (see page 25 of Mano), using base-2 as an intermediate format: 0x48 0x65 0x58 0x21

$0x48=(0100\ 1000)_2=(100\ 1000)_2$ → H
$0x65=(0110\ 0101)_2=(110\ 0101)_2$ → e
$0x58=(0101\ 1000)_2=(101\ 1000)_2$ → X
$0x21=(0010\ 0001)_2=(010\ 0001)_2$ → !

(d) Convert $(744)_{10}$ to base-7:

Residual	factor#	base#
744	2	7^3
58	1	7^2
9	1	7^1
2	2	7^0

So $(744)_{10}=(2112)_7$

Convert $(2112)_7$ to base-10:
$(2112)_7=2x7^3+1x7^2+1x7^1+2x7^0=(744)_{10}$

16. (5 pts.) Using the conversion method in class (which, I hope, is clearer than that described in Mano),
(a) Convert from (01111)2 to Gray code and back again.

To convert $(01111)_2$ to gray code:

Binary	Right-shift	Gray code (XOR of left columns)
01111	00111	01000

To convert $(01000)_{gray}$ to binary:

Gray code	Parity of higher digits	Binary (XOR of left columns)
01000	00111	01111

(b) Convert from $(10000)_2$ to Gray code and back again.:

To convert $(01111)_2$ to gray code:

Binary	Right-shift	Gray code (XOR of left columns)
10000	01000	11000

To convert $(01000)_{gray}$ to binary:

Gray code	Parity of higher digits	Binary (XOR of left columns)
11000	01000	10000