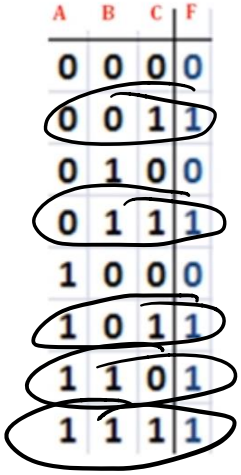
Lab #2 (Boolean Arithmetic)

Name: Raul Aguilar

Date: Sep 11, 2020

1. Given the following Truth Table



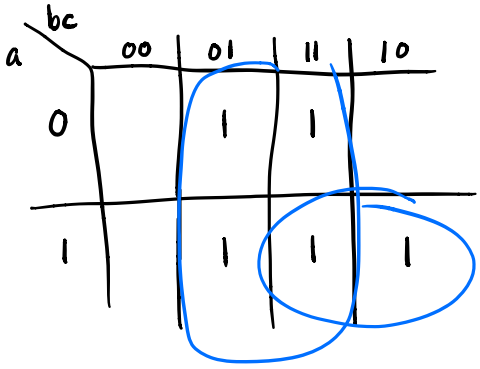
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

a) Write the function in its canonical form

$f(a,b,c) = \bar{a}\bar{b}c + \bar{a}bc + a\bar{b}c + ab\bar{c} + abc$

19 gates

b) Draw the K-Map of the function



c) Write the function in its simplified form

$f(a,b,c) = c + ab$

2 gates

d) What is the difference in the number of gates used between canonical and simplified forms?

17 gates

2. Given the following Truth Table

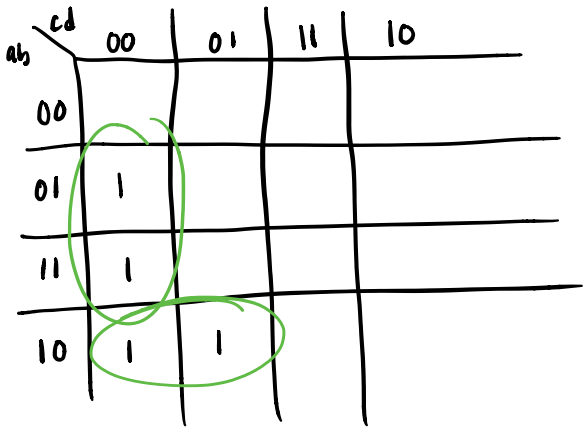
A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

a) Write the function in its canonical form

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

25 gates

b) Draw the K-Map of the function



$!b \,\&\&\, !d \;=\!=\; !(b \,\&\&\, d)$

$!b \;||\; !d$

c) Write the function in its simplified form

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

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

9 gates 7 gates

d) What is the difference in the number of gates used between canonical and simplified forms?

18 gates

3. Given the following Truth Table

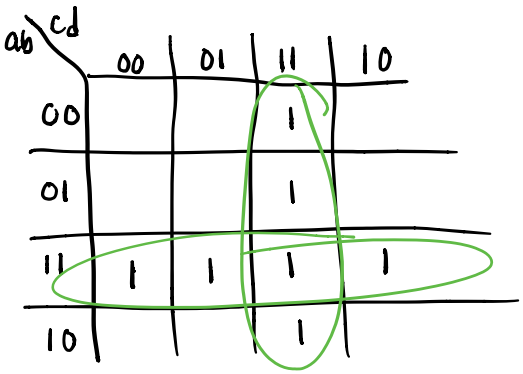
a	b	c	d	z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

a) Write the function in its canonical form

$f(a,b,c,d) = \bar{a}\bar{b}cd + \bar{a}bcd + a\bar{b}cd + ab\bar{c}\bar{d} + ab\bar{c}d + abc\bar{d} + abcd$

35 gates

b) Draw the K-Map of the function



c) Write the function in its simplified form

$f(a,b,c,d) = ab + cd$

3 gates

d) What is the difference in the number of gates used between canonical and simplified forms?

32 gates

Converting between Binary and Decimal Numbers

4. Convert **1110001**$_2$ to decimal (base 10)
   Use sum of expansion of products (don't skip steps!)

1110001

$2^6 + 2^5 + 2^4 + 2^0$

$64 + 32 + 16 + 1 = 113_{10}$

5. Convert **11011100**$_2$ to decimal (base 10)
   Use sum of expansion of products (don't skip steps!)

$2^7 + 2^6 + 2^4 + 2^3 + 2^2$

$128 + 64 + 16 + 8 + 4 = 220_{10}$

Converting between Decimal and Binary Numbers

6. Convert **35**$_{10}$ to binary (base 2)
   Use the Double-Dabble method of successive division (don't skip steps!)

35/2 17 r1
17/2 8 r1
8/2 4 r0
4/2 2 r0
2/2 1 r0
1/2 0 r1

100011_2

7. Convert $\mathbf{111}_{10}$ to binary (base 2)
   Use the Double-Dabble method of successive divsion (don't skip steps!)

111/2 55 r 1
55/2 27 r 1
27/2 13 r 1
13/2 6 r 1
6/2 3 r 0
3/2 1 r 1
1/2 0 r 1

1101111_2

Adding Unsigned Binary Numbers

8. Add 7 + 5 in binary.
   First convert to binary, then compute the sum.

```
 111
 0111
 0101
------
 1100
```

1100_2

Adding Signed Binary Numbers (with Negatives)

9. Add 7 + (-5) in binary. Same as subtraction.
   First convert to binary, then compute the sum.

7_{10} 0111_2
5_{10} 0101_2
1010
-5_{10} 1011_2

$$\begin{array}{r} {\scriptstyle 111} \\ 0111_2 \\ +\ 1011_2 \\ \hline 0010 \end{array}$$

overflow

$0010_2 = 2_{10}$

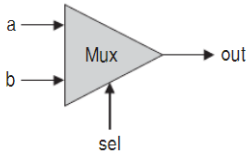
Multiplexor (Mux) Design

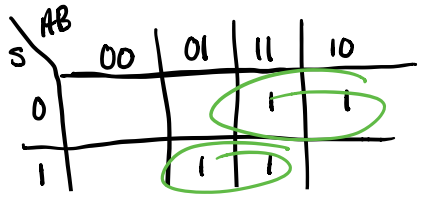
10. Write the Boolean function for the output (out). Use K-maps if needed.
    Then write the HDL code.

out(a,b,sel) = $sb + \bar{s}a$ 4 gates

a	b	sel	out
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

sel	out
0	a
1	b





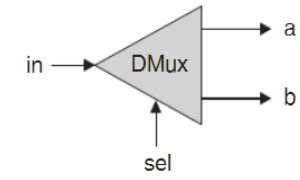

```
CHIP Mux {
   IN a, b, sel;
   OUT out;

   PARTS:
      Not(in = sel, out = nsel)
      And(a = a, b = nsel, out = sela)
      And(a = b, b = sel, out = selb)
      Or (a = sela, b = selb, out = out)
}
```

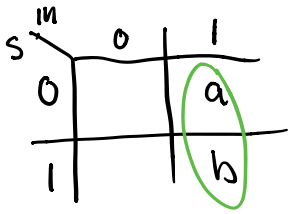
$f(a,b,s) = sb + \bar{s}a$

Demultiplexor (DMux) Design

11. Write the Boolean function for the output (out). Use K-maps if needed. Then write the HDL code.

sel	a	b
0	in	0
1	0	in





$in(\bar{s} + s)$

a(in, sel) = $in(\overline{sel})$

b(in, sel) = $in(sel)$

```
CHIP DMux {
   IN in, sel;
   OUT a, b;

   PARTS:
      Not(in = sel, out = nsel)
      And(a = in, b = nsel, out = a)
      And(a = in, b = sel, out = b)

}
```