Homework 2

1. $f_1(a,b,c) = \Sigma m(0,4,5,6)$

a)

c \ ab	00	01	11	10
0	1 (0)	0 (2)	1 (6)	1 (4)
1	0 (1)	0 (3)	0 (7)	1 (5)

b) Pairs: m0 & m4 ($b'c'$), m4 & m5 (ab'), m4 & m6 (ac')

c) Non-Pairs: $\emptyset$

d) $f_1(a,b,c) = b'c' + ab' + ac'$

Cost: Three 2-input AND, One 3-input OR (4 Gates total)

9 inputs

e) All essential, same cost

$f_2(a,b,c) = \Pi M(0,1,5,6)$

a)

c \ ab	00	01	11	10
0	0 (0)	1 (2)	0 (6)	1 (4)
1	0 (1)	1 (3)	1 (7)	0 (5)

b) Pairs: m2 & m3 ($a'b$), m3 & m7 (bc)

c) Non Pairs: m4 ($ab'c'$)

d) $f_2(a,b,c) = a'b + bc + ab'c'$

cost: Two 2-input AND, One 3-input AND, One 3 input OR (4 Gates total)

10 inputs

e) All essential, same cost

$f_3(a,b,c) = a'b' + a'c + abc + ab'c'$

a)

c \ ab	00	01	11	10
0	1 (0)	0 (2)	0 (6)	1 (4)
1	1 (1)	1 (3)	1 (7)	0 (5)

b) Pairs: m0 & m1 ($a'b'$), m0 & m4 ($b'c'$), m1 & m3 ($a'c$), m3 & m7 (bc)

c) Non-Pairs: $\emptyset$

d) $f_3(a,b,c) = a'b' + b'c' + a'c + bc$

cost: Four 2-input AND, one 4-input OR (5 Gates total)

12 inputs

e) Essential: m0 & m4 ($b'c'$), m3 & m7 (bc)

Optional: m0 & m1 ($a'b'$), m1 & m3 ($a'c$)

new minimal: $f_3(a,b,c) = b'c' + a'b' + bc$ (using m0 & m1)

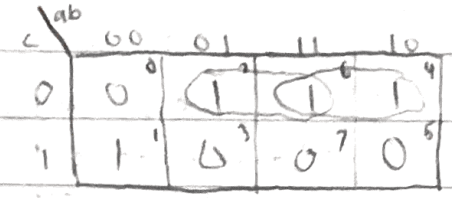
new cost: Three 2-input AND, One 3-input OR (4 Gates total)

9 inputs

* One less 2-input AND, 3-input OR instead of one 4-input OR and 3 less inputs

$f_4(a,b,c) = (b'+c')(a'+b+c')(a+b+c)$

a)



b) Pairs: m2 & m6 (bc'), m4 & m6 (ac')

c) Non-Pairs: m1 (a'b'c)

d) $f_4(a,b,c) = a'b'c + bc' + ac'$

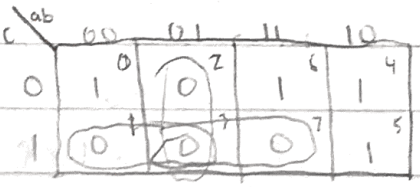
Cost: One 3-input AND, Two 2-input AND, One 3-input OR, (4 Gates total)
10 inputs

e) All essential: m2&m6 (bc'), m4&m6 (ac'), m1 (a'b'c)

Same cost

2. $f_1(a,b,c) = \Sigma m(0,4,5,6)$

a)



b) Pairs: M1 & M3 (a+c')
M2 & M3 (a+b')
M3 & M7 (b'+c')

c) Non-Pairs: ∅

d) $f_1(a,b,c) = (a+c')(a+b')(b'+c')$

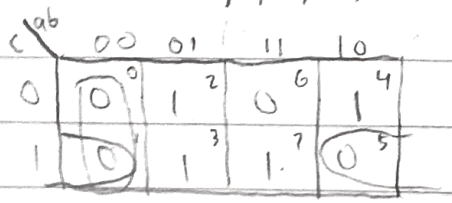
Cost: three 2-input OR, one 3-input
(Four Gates Total)
9 Inputs

e) All essential: M1&M3 (a+c'), M2&M3 (a+b'), M3 & M7 (b'+c')

same cost

$f_2(a,b,c) = \Pi M(0,1,5,6)$

a)



b) Pairs: M0 & M1 (a+b), M1 & M5 (b+c')

c) Non-Pairs: M6 (a+b+c')

d) $f_2(a,b,c) = (a+b)(b+c')(a+b+c')$

Cost: Two 2-input OR, One 3-input OR, One 3-input AND, (4 Gates Total)

10 Inputs

e) All essential: M0 & M1 (a+b), M1 & M5 (b+c'), M6 (a+b+c')

Same cost

$f_3(a,b,c) = a'b' + a'c + abc + ab'c'$

a)

c \ ab	00	01	11	10
0	1 (0)	0 (2)	0 (6)	1 (4)
1	1 (1)	1 (3)	1 (7)	0 (5)

b) Pairs: M2 & M6 ($b'+c$)

c) Non-Pairs: M5 ($a'+b+c'$)

d) $f_3(a,b,c) = (b'+c)(a'+b+c')$

Cost: One 2-input OR, one 3-input OR, One 2-input AND (3 Gates Total)

7 Inputs

e) All essential: M2 & M6 ($b'+c$), M5 ($a'+b+c'$)

Same cost

$f_4(a,b,c) = (b'+c')(a'+b+c')(a+b+c)$

a)

c \ ab	00	01	11	10
0	0 (0)	1 (2)	1 (6)	1 (4)
1	1 (1)	0 (3)	0 (7)	0 (5)

b) Pairs: M3 & M7 ($b'+c'$), M5 & M7 ($a'+c'$)

c) Non-Pairs: M0 ($a+b+c$)

d) $f_4(a,b,c) = (a+b+c)(b'+c')(a'+c')$

Costs: Two 2-input OR, One 3-input OR, One 3-input AND (4 Gates Total)

10 Inputs

e) All essential: M3 & M7 ($b'+c'$), M5 & M7 ($a'+c'$), M0 ($a+b+c$)

Same Cost

3. a) $f(a,b,c) = a \oplus b$

c \ ab	00	01	11	10
0	0	1	0	1
1	0	1	0	1

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

b) $f(a,b,c) = a \oplus c$

c \ ab	00	01	11	10
0	0	0	1	1
1	1	1	0	0

$f(a,b,c) = a'c + ac'$

c) $f(a,b,c) = b \oplus c$

c \ ab	00	01	11	10
0	0	1	1	0
1	1	0	0	1

$f(a,b,c) = bc' + b'c$

a	b	c	a⊕b	a⊕c
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	0	1

d) $f(a,b,c) = (a \oplus b) \oplus (a \oplus c)$

c \ ab	00	01	11	10
0	0	1	1	0
1	1	0	0	1

$f(a,b,c) = bc' + b'c$

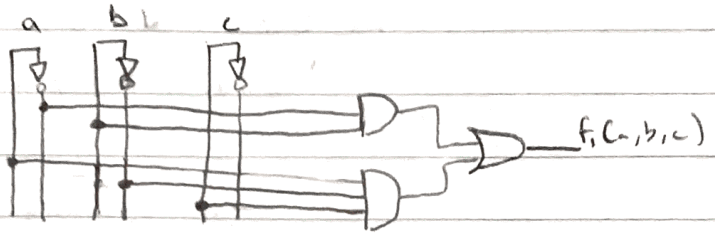
e) $f(a,b,c) = (a \oplus b) \oplus (c)$

c \ ab	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$f(a,b,c) = a'b'c + a'bc' + abc + ab'c'$

C and D have the same K-map and Algebraic Expression

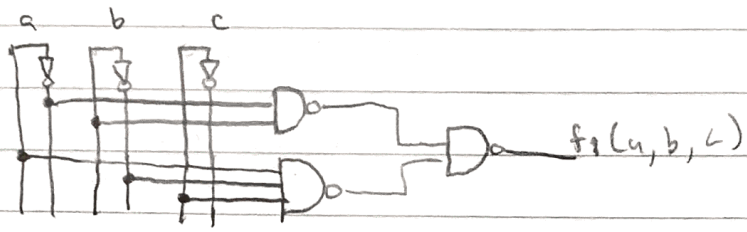
4a. $f_1(a,b,c) = a'b + ab'c$



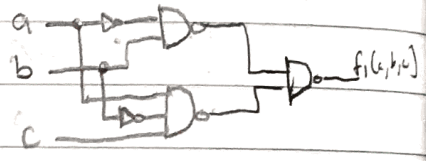
$(a'b)'' + (ab'c)''$

$((a'b)')' + ((ab'c)')'$

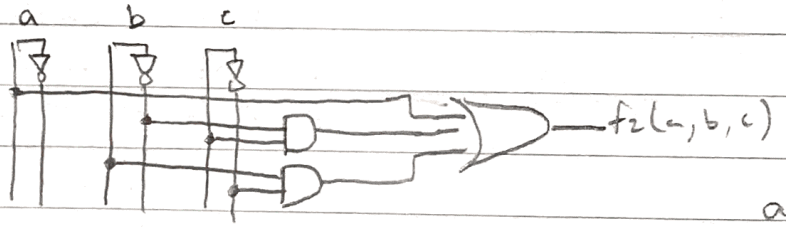
$((a'b)' \cdot (ab'c)')'$



or



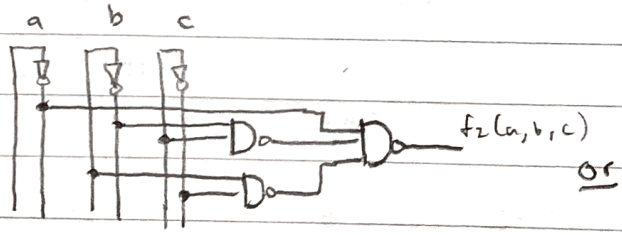
4b. $f_2(a,b,c) = a + b'c + bc'$



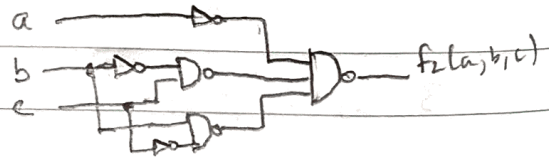
$a'' + (b'c)'' + (bc')''$

$(a')' + ((b'c)')' + ((bc')')'$

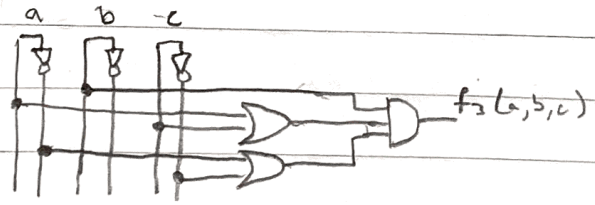
$(a' \cdot (b'c)' \cdot (bc')')'$



or



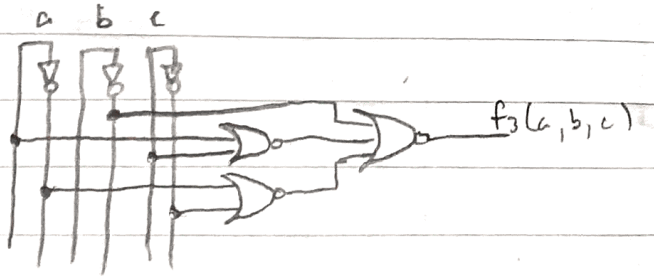
4c. $f_3(a,b,c) = b \cdot (a+c) \cdot (a'+c')$



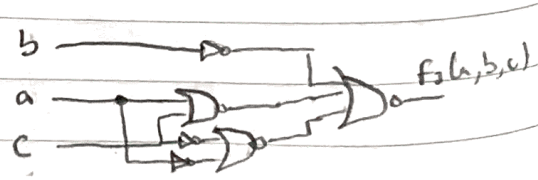
$b'' \cdot (a+c)'' \cdot (a'+c'')''$

$(b')' \cdot ((a+c)')' \cdot ((a'+c')')'$

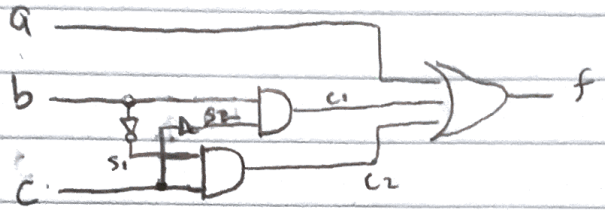
$(b' + (a+c)' + (a'+c')')'$



or



5. $f_2(a,b,c) = a + b'c + bc'$



Verilog Module

```
wire s1, s2, c1, c2;
module full_adder_structural(a,b,c,f);
    input  a, b, c;
    output f;

    not  not_b(s1,b);
    not  not_c(s2,c);
    and  and_c1(c1,b,s2);
    and  and_c2(c2,s1,c);
    or   or_f(f,a,c1,c2);
endmodule
```