2. Katz and Boriello 3.18

Implement Hazard free circuits for each boolean function

a). $F(A,B,C) = B\bar{C} + \bar{A}C$

K-Map

C \ AB	00	01	11	10
0	0	1	1	0
1	1	1	0	0

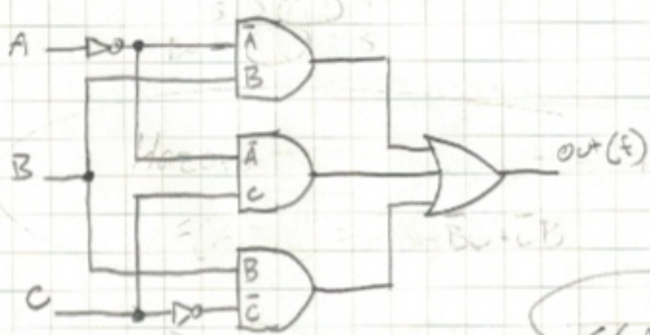
Hazard

Truth table

A	B	C	Out
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

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

Static 1-Hazard eliminated
Static 0 Hazard cannot exist in POS form



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

2 Continued, Katz and Boriello 3.18

B) $F(A,B,C,D) = \sum m(0,4,5,6,7,9,11,13,14)$

K-Map

CD \ AB	00	01	11	10
00	1	1	0	0
01	0	1	1	1
11	0	1	0	1
10	0	1	1	0

static 1-Hazard $+B\bar{C}D$

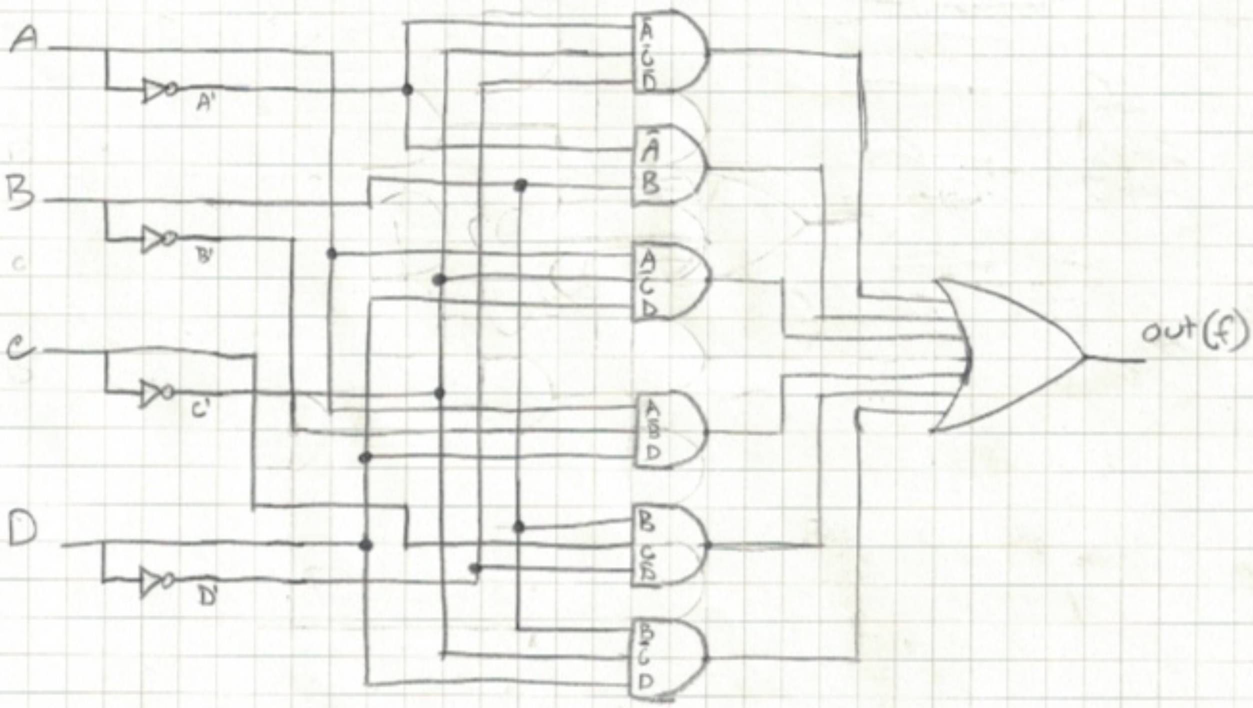
truth table

num	A	B	C	D	out
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

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

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

static 1 hazard eliminated



2. Continued, Katz and Boriello 3.18

c) $F(A,B,C) = (A+B)(\bar{B}+C)$

truth table

A	B	C	Out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

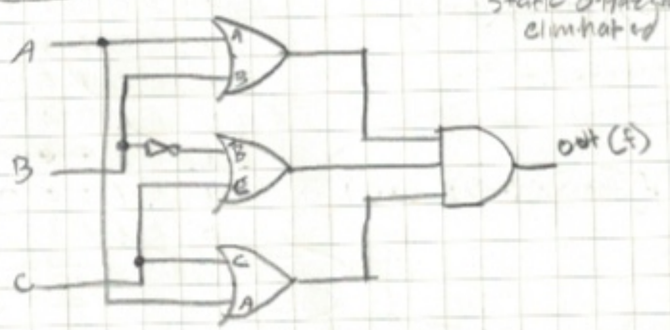
K-Map

C \ AB	00	01	11	10
0	0	0	0	1
1	0	1	1	1

static 0-Hazard

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

static 0-Hazard eliminated



2 Continued, Katz and Borriello 3.18

D). $F(A,B,C,D) = \Pi M(0,1,3,5,7,8,9,13,15)$

K map

CD \ AB	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	0	0	0	1
10	1	1	1	1

static 0-Hazard $(\bar{C}+D)$

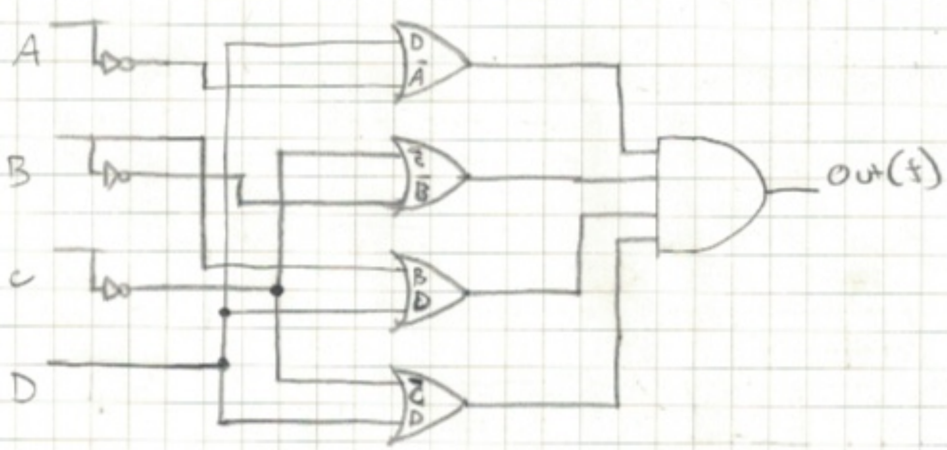
truth table

	A	B	C	D	out
m0	0	0	0	0	0
m1	0	0	0	1	0
m2	0	0	1	0	1
m3	0	0	1	1	0
m4	0	1	0	0	1
m5	0	1	0	1	0
m6	0	1	1	0	1
m7	0	1	1	1	0
m8	1	0	0	0	0
m9	1	0	0	1	0
m10	1	0	1	0	1
m11	1	0	1	1	1
m12	1	1	0	0	1
m13	1	1	0	1	0
m14	1	1	1	0	1
m15	1	1	1	1	0

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

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

static 0-hazard eliminated



2 Continued, Katz and Borriello 3.18

E) $F(A,B,C,D,E) = \sum m(0,1,3,4,7,11,12,15,16,17,20,28)$

layer 1
A=0

DE \ BC	00	01	11	10
00	1	1	1	0
01	1	0	0	0
11	1	1	1	1
10	0	0	0	0

CD'E'
B'C'D'
A'DE

Hazards

layer 2
A=1

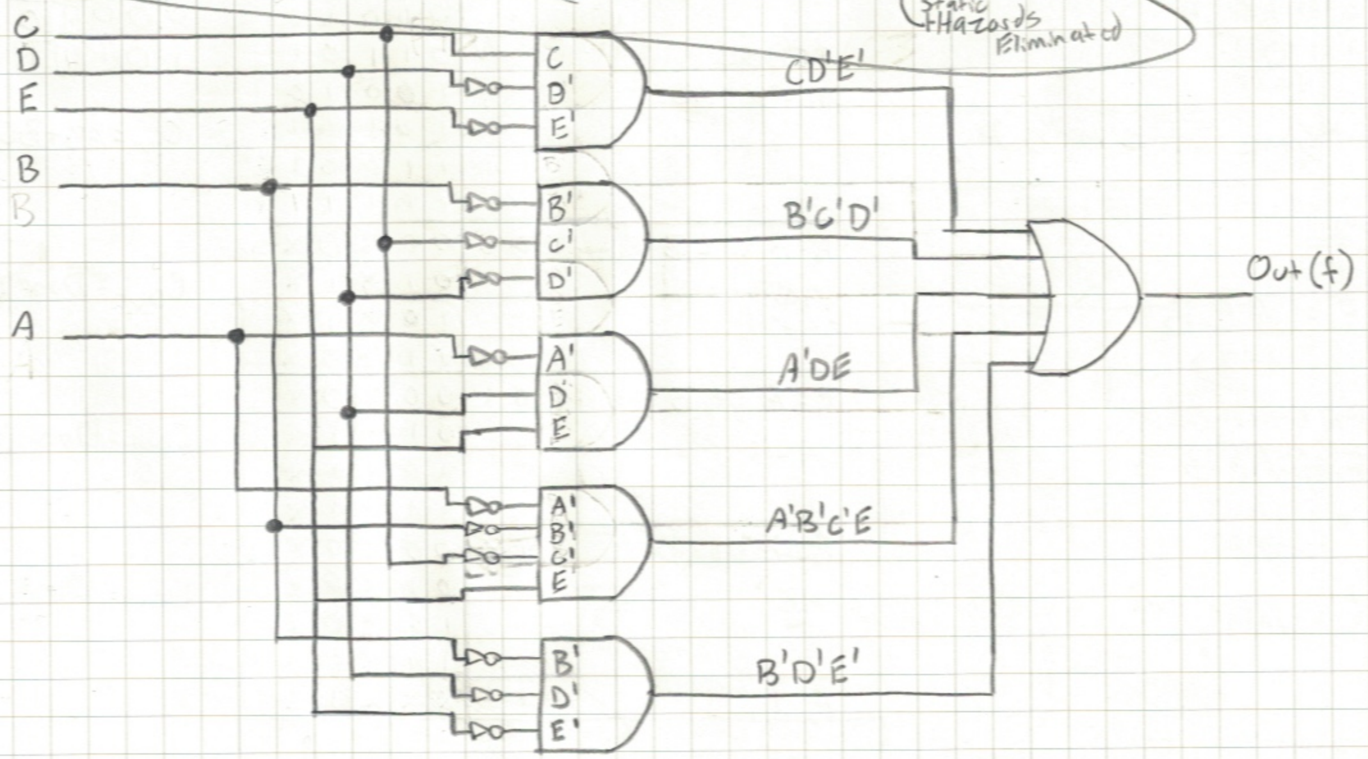
DE \ BC	00	01	11	10
00	1	1	1	0
01	1	0	0	0
11	0	0	0	0
10	0	0	0	0

Hazard Fixes
A'B'C'E
B'D'E'

truth table

m	A	B	C	D	E	Out	m	A	B	C	D	E	Out
0	0	0	0	0	0	1	16	1	0	0	0	0	1
1	0	0	0	0	1	1	17	1	0	0	0	1	1
2	0	0	0	1	0	0	18	1	0	0	1	0	0
3	0	0	0	1	1	1	19	1	0	0	1	1	0
4	0	0	1	0	0	1	20	1	0	1	0	0	1
5	0	0	1	0	1	0	21	1	0	1	0	1	0
6	0	0	1	1	0	0	22	1	0	1	1	0	0
7	0	0	1	1	1	1	23	1	0	1	1	1	0
8	0	1	0	0	0	0	24	1	1	0	0	0	0
9	0	1	0	0	1	0	25	1	1	0	0	1	0
10	0	1	0	1	0	0	26	1	1	0	1	0	0
11	0	1	0	1	1	1	27	1	1	0	1	1	0
12	0	1	1	0	0	1	28	1	1	1	0	0	1
13	0	1	1	0	1	0	29	1	1	1	0	1	0
14	0	1	1	1	0	0	30	1	1	1	1	0	0
15	0	1	1	1	1	1	31	1	1	1	1	1	0

$F(A,B,C,D,E) = CD'E' + B'C'D' + A'DE + \underbrace{A'B'C'E + B'D'E'}_{\text{Static Hazards Eliminated}}$



3) Katz and Borriello 5.12

Implement the ALU in question 5.12

Fout K Maps

$x \oplus y = xy' + x'y$

$(x \oplus y)' = xy + x'y'$

C \ AB	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Distr. $F_1 = AB'C' + A'BC' + A'B'C + ABC$

XOR $C'(AB' + A'B) + C(A'B' + AB)$

XOR' $C'(A \oplus B) + C(A'B' + AB)$

XOR $C'(A \oplus B) + C(A \oplus B)'$

$C \oplus A \oplus B$

C \ AB	00	01	11	10
0	0	1	0	1
1	1	0	1	0

*same as B minus A

C \ AB	00	01	11	10
0	0	1	0	1
1	1	0	1	0

*same as B minus A

$F_1 = A \oplus B$

$F_1 = A + B$

$F_1 = AB$

Truth table

ALU operation	S_2	S_1	S_0	A	B	Cin	F	Cout
$F_1 = 0$	0	0	0	X	X	X	0	X
$F_1 = B$ minus A	0	0	1	0	0	0	0	0
				0	0	1	1	1
$A \oplus B \oplus C$				0	1	0	1	0
$AB' + AC + B'C$				0	1	1	0	0
				1	0	0	1	1
				1	0	1	0	1
				1	1	0	0	0
				1	1	1	1	1
$F_1 = A$ minus B	0	1	0	0	0	0	0	0
				0	0	1	1	1
				0	1	0	1	1
$A \oplus B \oplus C$				0	1	1	0	1
$A'B + A'C + BC$				1	0	0	1	0
				1	0	1	0	0
				1	1	0	0	0
				1	1	1	1	1
$F_1 = A$ plus B	0	1	1	0	0	0	0	0
				0	0	1	1	0
				0	1	0	1	0
$A \oplus B \oplus C$				0	1	1	0	1
$AB + BC + AC$				1	0	0	1	0
				1	0	1	0	1
				1	1	0	0	1
				1	1	1	1	1
$F_1 = A$ XOR B	1	0	0	0	0	X	0	X
$A \oplus B$				0	1	X	1	X
				1	0	X	1	X
				1	1	X	0	X
$F_1 = A$ or B	1	0	1	0	0	X	0	X
$A + B$				0	1	X	1	X
				1	0	X	1	X
				1	1	X	1	X
$F_1 = A$ and B	1	1	0	0	0	X	0	X
AB				0	1	X	0	X
				1	0	X	0	X
				1	1	X	1	X
$F_1 = 1$	1	1	1	X	X	X	1	X

Cout Kmaps

C \ AB	00	01	11	10
0	0	0	0	1
1	1	0	1	1

$C_{out} = AB' + AC + B'C$

C \ AB	00	01	11	10
0	0	1	0	0
1	1	1	1	0

$C_{out} = A'B + A'C + BC$

C \ AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

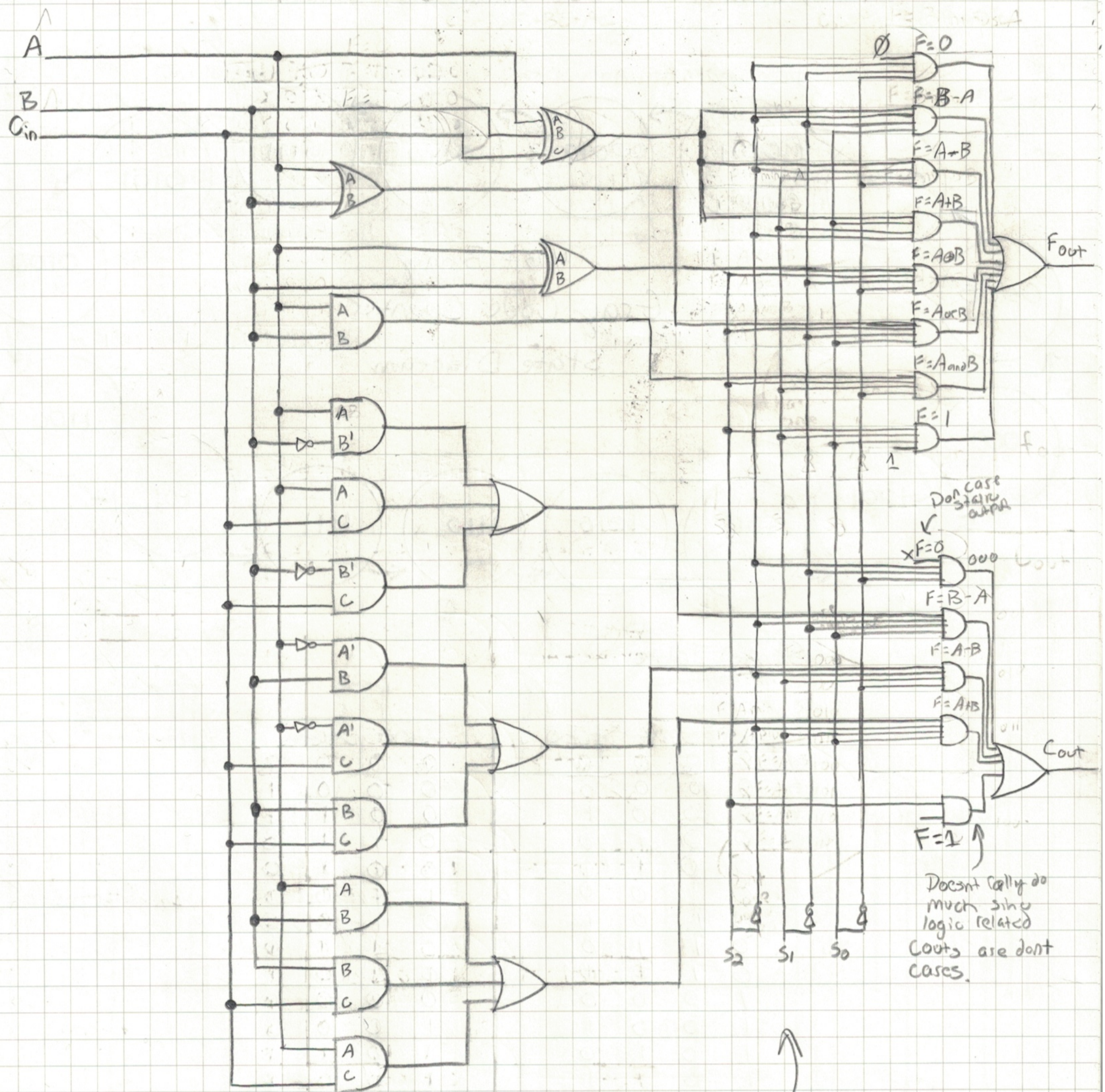
$C_{out} = AB + BC + AC$

C_{out} = Dont Care

C_{out} = Dont Care

C_{out} = Dont Care

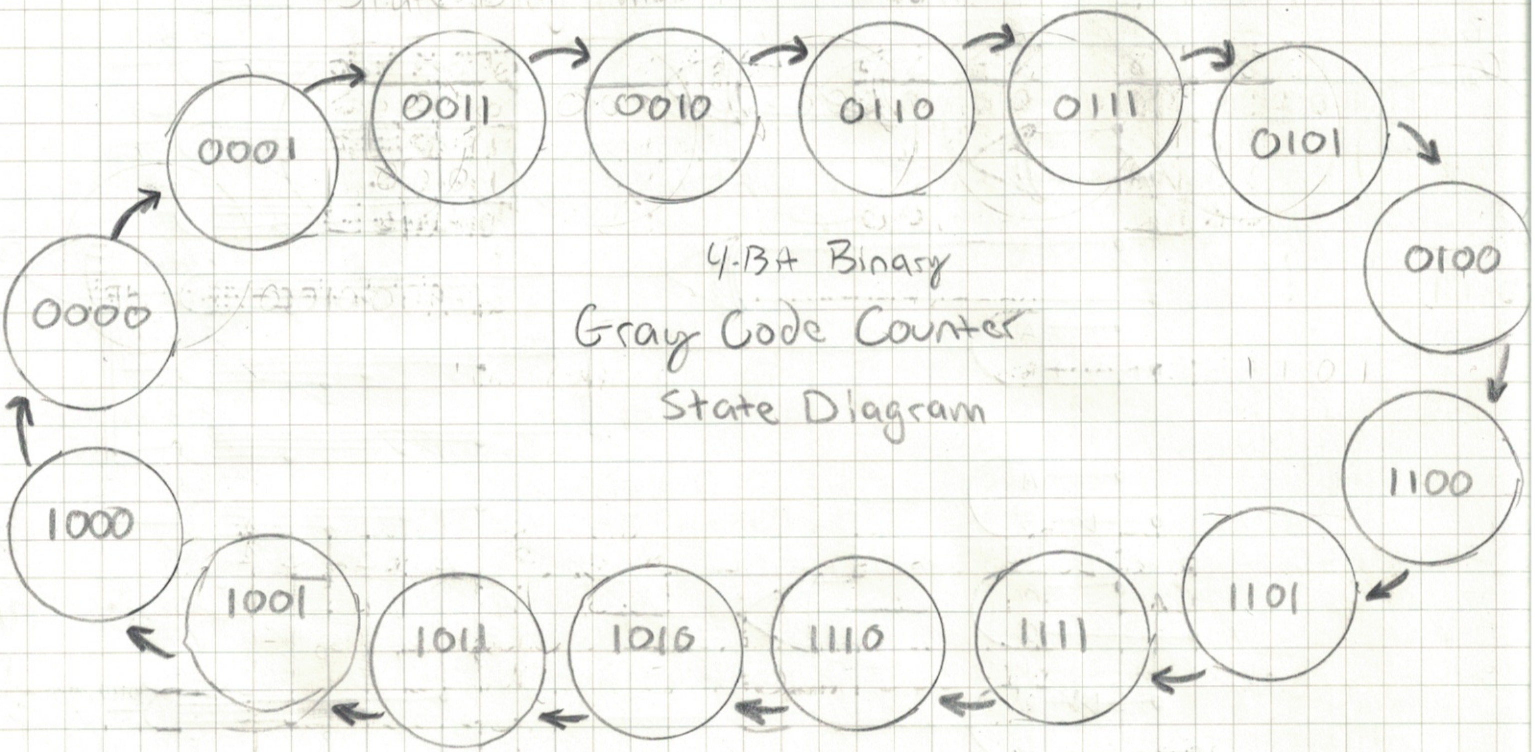
3) Continued, Katz and Borriello 5.12



Sorry if the Mux Gates are messy, i saw last minute that our muxs had to be drawn out at the gate level

4) Katz and Borriello 7.4

a) draw a state diagram and next state table



State Table

Present State				Next State			
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	1
0	0	1	0	0	1	1	0
0	0	1	1	0	0	1	0
0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	1	1
0	1	1	1	0	1	0	1
1	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	1	1
1	0	1	1	1	0	0	1
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	1
1	1	1	0	1	0	1	0
1	1	1	1	1	1	1	0

4) Continued Katz and Boriello 7.4

B) Implement the Counter using D-flip flops

Q_3 K-Map

CD \ AB	00	01	11	10
00	0	1	1	0
01	0	0	1	1
11	0	0	1	1
10	0	0	1	1

$Q_3 = BC'D' + AD + AC$

Q_2 K-Map

CD \ AB	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	0	1	1	0
10	1	1	0	0

$Q_2 = BC' + BD + A'CD'$

Q_1 K-Map

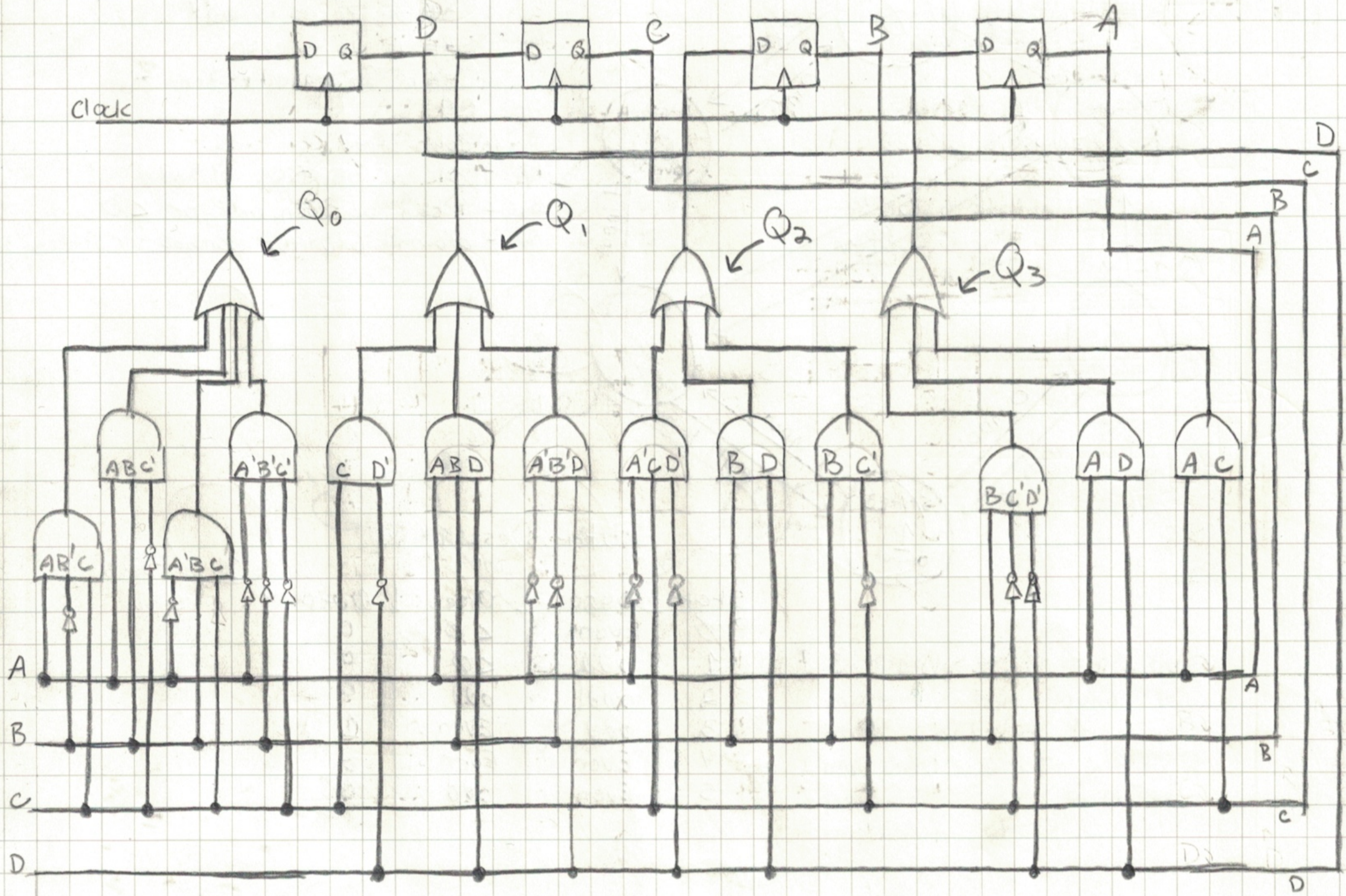
CD \ AB	00	01	11	10
00	0	0	0	0
01	1	0	1	0
11	1	0	1	0
10	1	1	1	1

$Q_1 = A'B'D + ABD + CD'$

Q_0 K-Map

CD \ AB	00	01	11	10
00	1	0	1	0
01	1	0	1	0
11	0	1	0	1
10	0	1	0	1

$Q_0 = A'B'C' + A'BC + ABC' + AB'C$

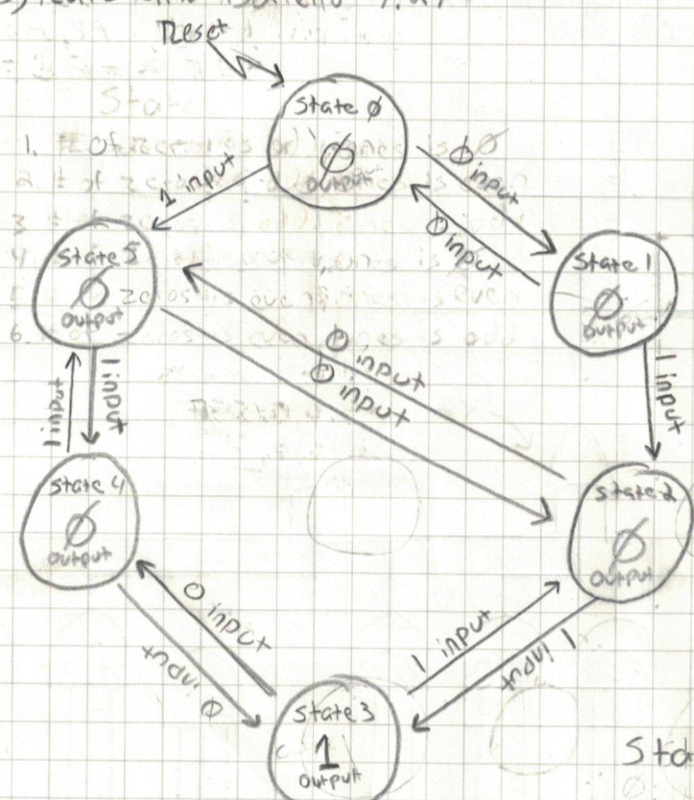


4) Continued Katz and Borriello 7.4

c) Do you have to worry about self starting? why or why not?

A.) Self starting shouldn't be a worry with this counter since it is always already self starting. We know this simply because all possible states are accounted for in the sequence. Also, its fortunate that all possible states are actually in the count sequence, so it will take Ø additional transitions the be sure the counter is in the count sequence

5) Katz and Borriello 7.27



Notes
- one input
- one output
- should be 1 if...
  - # of zeros is odd
  - # of 1s is even
  - # of 1s greater than zero

- in each state...
  - Ø can be even or odd
  - 1 can be even, odd, or < Ø
  - allowing 6 unique combos for states

State Descriptions

State	Zeros	Ones	Output
Ø	Even	< Ø	0
1	odd	< Ø	0
2	odd	odd	0
3	odd	Even	1
4	Even	Even	0
5	Even	odd	0

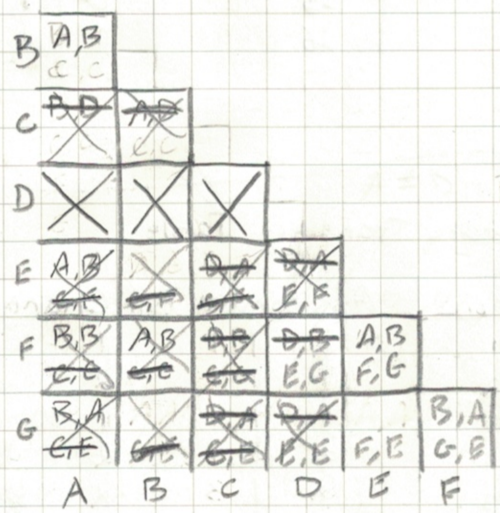
6) Katz and Borello 8.2 State Reduction

State table

Present state	Next state x=0	x=1	Output	Reduced next state x=0	x=1
A	B	C	0	A	C
B	A	C	0	~~A~~	~~C~~
C	D	C	0	D	C
D	D	E	1	D	E
E	A	F	0	A	E
F	B	G	0	~~A~~	~~E~~
G	A	E	0	~~A~~	~~E~~

State Assignment

A: 000 E: 100
B: 001 F: 101
C: 010 G: 110
D: 011



1st pass: A ≠ D, B ≠ D, C ≠ D
2nd Pass: A ≠ C, B ≠ C, C ≠ E, C ≠ F, C ≠ G
D ≠ E, D ≠ F, D ≠ G
3rd Pass: A ≠ E, A ≠ F, A ≠ G
B ≠ E, B ≠ F, B ≠ G

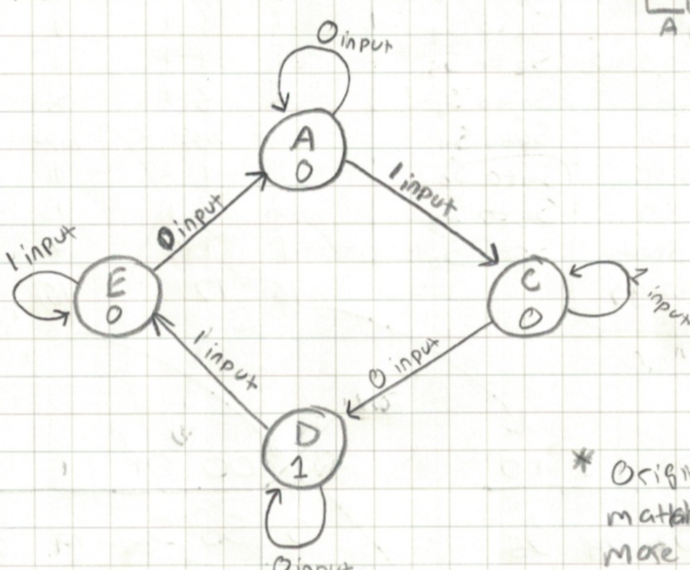
Equivalent state List

A, B → AB

F, G
G, E
F, E
} FG, GE, EF → FGE

All Reduced States

AB, C, D, EFG
A, C, D, E



* Originally tried to use row matching but found it more confusing than implication chart although probably faster, so chose implication chart

7) Katz and Borriello 8.7 (State Assignment)

A) Minimum bit chage heuristic

State Map

Q_2 \ Q_1Q_0	00	01	11	10
0	S_0	S_1	S_3	S_5
1	S_2	S_4	S_6	X

State assignment table

State Name	Assignment Q_2	Q_1	Q_0
S0	0	0	0
S1	0	0	1
S2	1	0	0
S3	0	1	1
S4	1	0	1
S5	0	1	0
S6	1	1	1

Since single bit changes are adjacent due to the nature of K-maps, and Gray code, and the state diagram followed a sort of two path flow, i chose to start by guessing and checking the two main state transition flows (S0,2,4,6 & S0,1,3,5) with each state adjacent to the next in rows similar to the diagram paths. This created many single bit chages but state changes out of "order" (eg. S_1 to S_3) proved difficut, for which I ended up guessing and checking 4 variations below and choosing the one with fewest bits chaged in state transitions overall. S_0 always started at 000.

State Transition Table

Transition	Assignment Bit Changes			
S_0 to S_1	2	1	1	1
S_0 to S_2	1	1	2	1
S_1 to S_3	1	1	1	1
S_1 to S_4	2	3	2	1
S_2 to S_3	2	1	2	3
S_2 to S_4	1	1	1	1
S_3 to S_5	1	1	1	1
S_3 to S_6	2	3	2	1
S_4 to S_4	N/A	N/A	N/A	N/A
S_4 to S_6	1	1	1	1
S_5 to S_0	2	3	1	1
S_6 to S_0	1	1	2	3
	16	17	16	15

1st

Q_2 \ Q_1Q_0	00	01	11	10
0	S_0	S_2	S_4	S_6
1	X	S_1	S_3	S_5

2nd

Q_2 \ Q_1Q_0	00	01	11	10
0	S_0	S_2	S_4	S_6
1		S_1	S_3	S_5	X

3rd

0	S_0	S_1	S_3	S_5
1	X	S_2	S_4	S_6

4th

0	S_0	S_1	S_3	S_5
1	S_2	S_4	S_6	X

7.) Katz and Boriello 8.7 (state Assignment)

B) State Assignmen Guidelines

State assignments

S_0 = 000
X 001
S6 010
S5 011
S3 110
S4 111
S1 100
S2 101

Highest Priority: S_1, S_2 $\quad S_1, S_2, S_4$
S_3, S_4 $\quad (S_5, S_6) \times 2$

Medium Priority: S_1, S_2 $\quad (S_3, S_4) \times 2$
S_4, S_6 $\quad S_5, S_6$

Lowest Priority: S_0, S_1, S_4, S_3, S_2 - 1/0
S_0, S_4, S_5, S_6 - 0/0
S_1, S_2, S_3 = 0/1
S_3, S_6 = 1/1

Q_2 \ Q_1Q_0	00	01	11	10
0	S_0	S_6	S_3	S_1
1		S_5	S_4	S_2

can be flipped to allow (S_4, S_6 adjacency)

S_0=000
X=001
S_5=010
S_6=011
S_3=110
S_4=111
S_1=100
S_2=101

Explination

For the first iteation of the state map i was able to easily accomplish the top 3 adjacencies occuring in the high and mid priority lists. (S_1, S_2 - S_3, S_4 - S_5, S_6). I was unsure of exactly where they should go but i knew they should stay together. I noticed the S_5, S_6 adjacency could be flipped to also support ~~high~~ mid priority S_4, S_6, so i did that. Of course, S_0 started at 000 and S_0 had no high or mid priorities so 001 was left empty

State Map

Q_2 \ Q_1Q_0	00	01	11	10
0	S_0	S_5	S_3	S_1
1	X	S_6	S_4	S_2

State Assignment Table

State Name	Assignment Q_0	Q_1	Q_2
S_0	0	0	0
S_1	1	0	0
S_2	1	0	1
S_3	1	1	0
S_4	1	1	1
S_5	0	1	0
S_6	0	1	1