Saaif Ahmed 11/18/19
661925946 ahmeds7

CoCo HW #11

Problem 1:

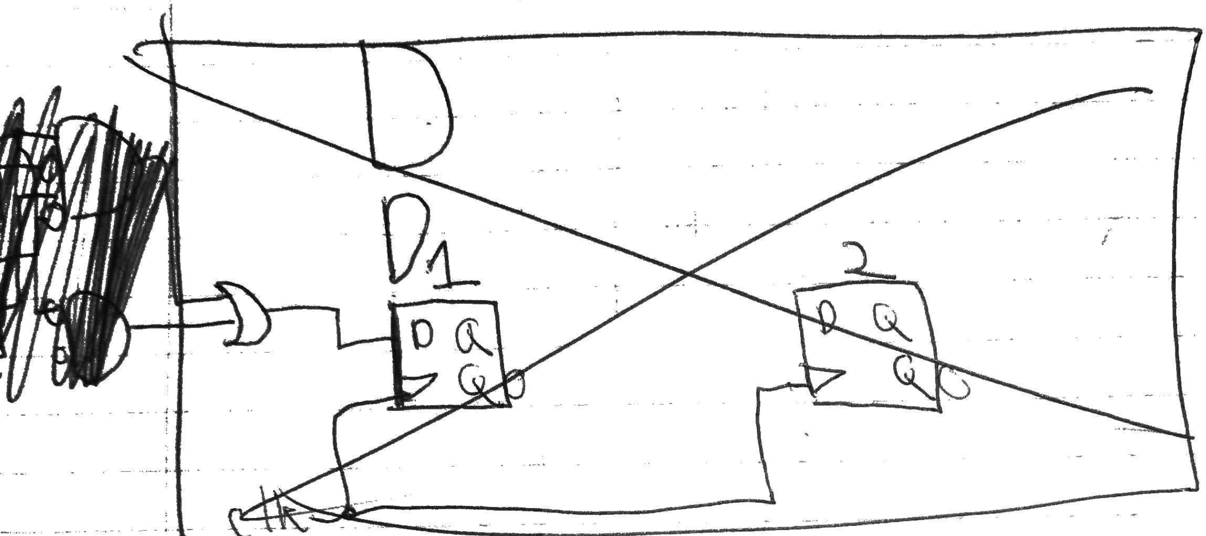
State:

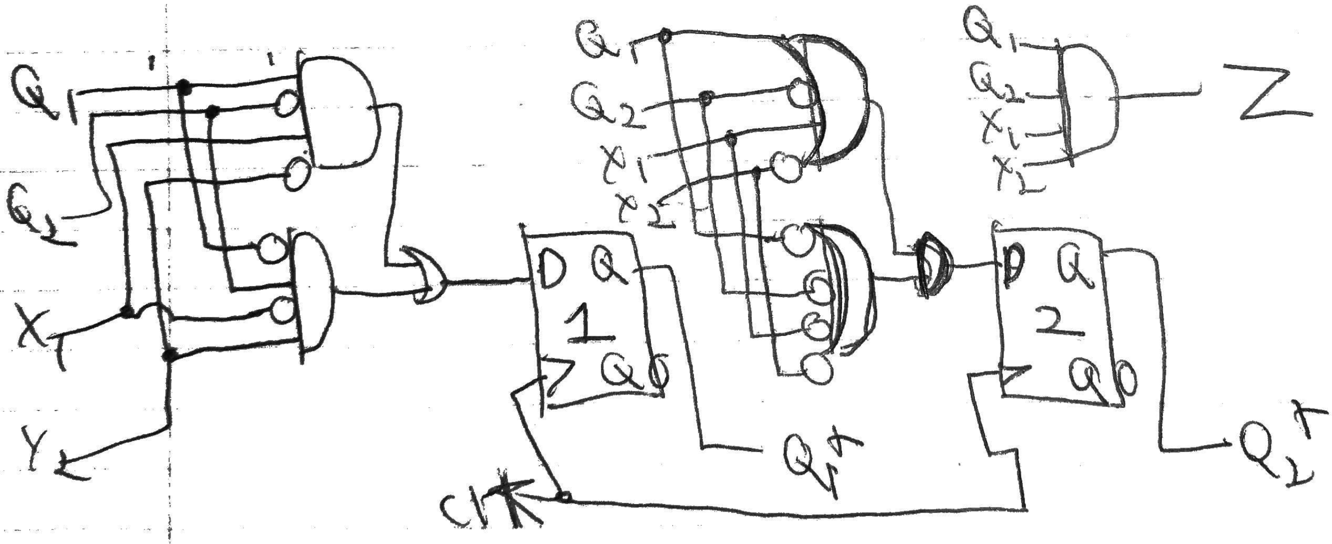
~~Start 000~~	Got X_1	00	A
~~Got X_1 001~~	Got X_1X_2	01	B
~~Got X_1X_2 010~~	Got $X_1X_2X_2$	10	C
~~Got $X_1X_2X_2$ 011~~	Got $X_1X_2X_2X_1$	11	D
~~Got $X_1X_2X_2X_1$ 100~~			

State	Input	Next State	Unlock
00	00	00	0
	01	01	0
	10	00	0
	11	00	0
01	00	00	0
	01	10	0
	10	00	0
	11	00	0
10	00	00	0
	01	00	0
	10	11	0
	11	00	0
11	00	00	0
	01	01	0
	10	00	0
	11	00	1
$Q_1 \mid Q_2$	$X_1 \mid X_2$	$Q_1^+ \mid Q_2^+$	

From Espresso

$$Q_1^+ = Q_1\overline{Q_2}X_1\overline{X_2} + \overline{Q_1}Q_2\overline{X_1}\cancel{}X_2$$
$$Q_2^+ = (Q_1+Q_2+\cancel{X_1}+\overline{X_2})(\overline{Q_1}+\overline{Q_2}+\cancel{X_1}+\overline{X_2})$$
$$Z = Q_1Q_2\cancel{X_1}X_2$$





Assumptions: Every state I made was an assumption. I also invalidated double button presses. To compensate for this I assumed that everyone would at least start with X_1. Pressing X_1 brings you back to the OG state so it ends up working out.

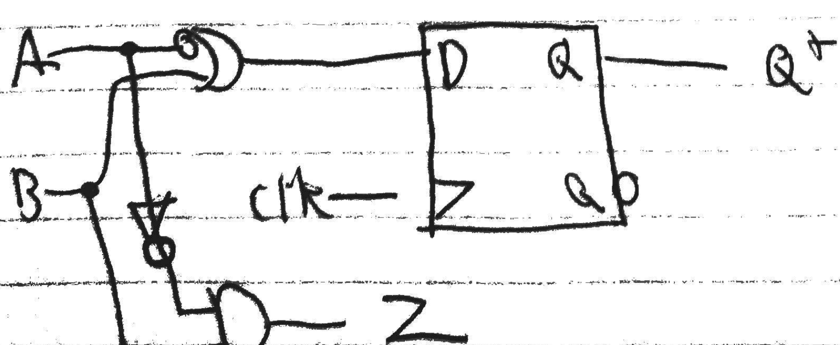
Problem 2:

~~States~~ States
~~Floor 1~~ Floor 1 or Floor 2 : Q
Inputs: button pressed for each floor : A
door open / door close : B 1 = door open 0 = door close
Output: "moving" : Z

State	Input		Next State	Z
0	0	0	0	0
	0	1	0	0
	1	0	1	1
	1	1	0	0
1	0	0	1	0
	0	1	1	0
	1	0	0	1
	1	1	1	0
Q	A	B	Q^+	

Use D flip flops

From espresso
$Q^+ = (\bar{A} + B)$
$Z = \bar{A}\ B$



```
#Hw 11 Problem 1 Input
.i 4
.o 3
0000 000
0001 010
0010 000
0011 000
0100 000
0101 100
0110 000
0111 000
1000 000
1001 000
1010 110
1011 000
1100 000
1101 010
1110 000
1111 001
```

```
#Hw 11 Problem 1 Output
.i 4
.o 3
.p 5
1010 110
0101 100
0001 010
1111 001
1101 010
.e
```

```
#Hw 11 Problem 2 Input
.i 3
.o 2
000 00
001 00
010 11
011 00
100 10
101 10
110 01
111 10
```

```
#Hw 11 Problem 2 Output
.i 3
.o 2
.p 4
010 10
-10 01
1-1 10
10- 10
.e
```

HW #11 Re Sub

State:

A : 000 : Start

B : X_1 : 001

C : $X_1 X_2$: 010

D : $X_1 X_2 X_2$: 011

E : $X_1 X_2 X_2 X_1$: 100

Table:

			$X_1 X_2$				
Q_1	Q_2	Q_3	00	01	10	11	Unlock
0	0	0	000	000	001	000	0
0	0	1	000	010	000	000	0
0	1	0	000	011	000	000	0
0	1	1	000	000	100	000	0
1	0	0	000	000	001	000	1

$\underbrace{\hspace{6em}}_{Q_1^+ Q_2^+ Q_3^+}$

Boolean expressions for next states! Used D Flip Flops.

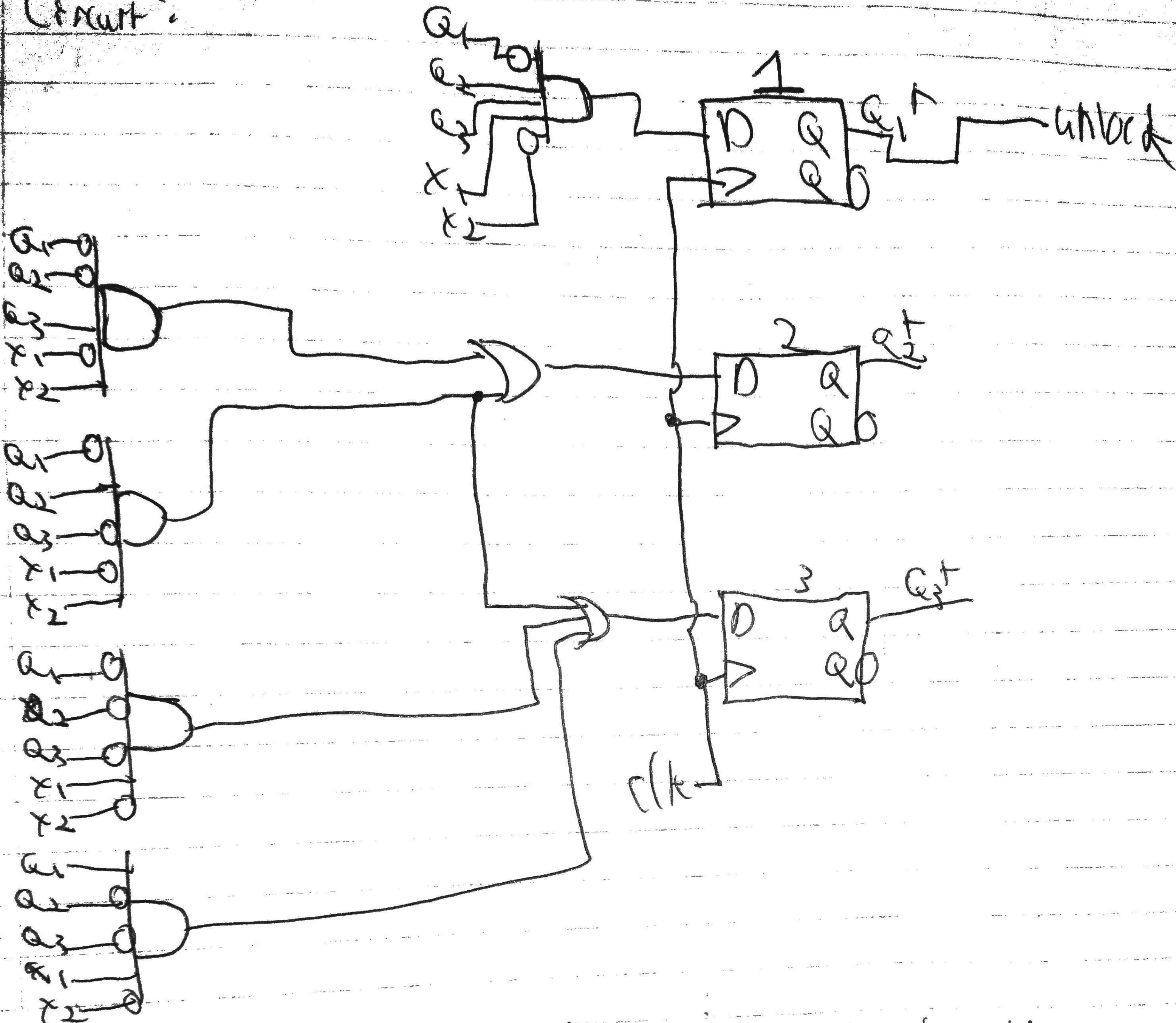
Characteristic equation $D = Q^+$

$$Q_1^+ = \overline{Q_1} Q_2 Q_3 X_1 \overline{X_2}$$

$$Q_2^+ = \overline{Q_1}\,\overline{Q_2} Q_3 \overline{X_1} X_2 + \overline{Q_1} Q_2 \overline{Q_3}\,\overline{X_1} X_2$$

$$Q_3^+ = \overline{Q_1}\,\overline{Q_2}\,\overline{Q_3} X_1 \overline{X_2} + \overline{Q_1} Q_2 \overline{Q_3}\,\overline{X_1} X_2 + Q_1 Q_2 Q_3 X_1 \overline{X_2}$$

Circuit:



* I believe my design was correct for the four states I created I just had to add in one more state and then develop the logic for that. I understand the design method for this FSM now.

Problem 22

States: 0 = elevator at floor 1
1 = elevator at floor 2

Inputs: A: up button
B: down button
C: door open
D: door close

~~Z = output = moving~~

Transition Diagram / excitation table

Inputs: ABCD

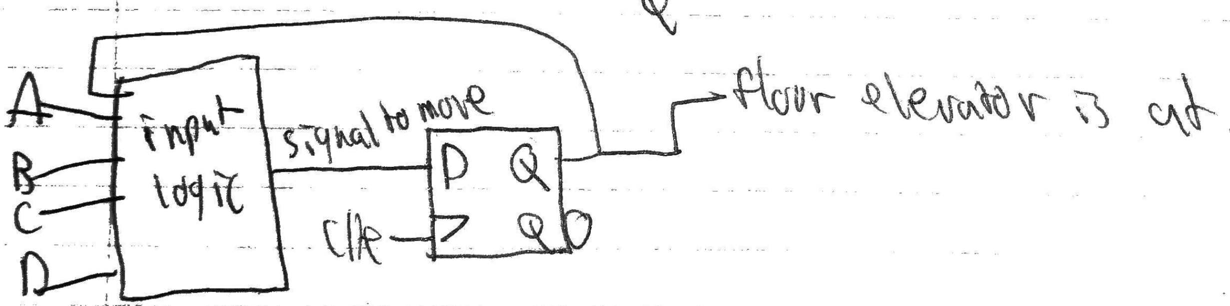
States	0000	0001	0010	0011	0100	0101	0110	0111
0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	1	0

Inputs: ABCD

States	1000	1001	1010	1011	1100	1101	1110	1111
0	0	1	0	1	0	1	0	1
1	1	1	1	1	1	0	1	0

Q

Q^+



* In order to move the elevator the correct relative button must be pressed and the door must be closed. This is shown through the combinational logic. The elevator will then move to the respective floor as represented by the D flip flop. I made the assumption that pressing both elevator buttons and both door open and close is valid. The reason is, elevators in real life can handle that input.