Project Part 2

Emon Sarker

ID: 1931461642

CSE231.4

Introduction

In phase 2 of our CSE231 project we have to include a sequential circuit to the previously created combinational circuit. The main purpose is to create a circuit that will sequentially print out the entire code given to us without requiring any ~~an~~ user input. I have decided to use JK Flip-Flops and my previously built "Simplified Sum of Products" combinational circuit. I will be explaining my reasoning in the next segment

Cost - Analysis

The reason I chose JK-Flip Flop and simplified SOP is because the total circuit becomes the cheapest. All permutations have a couple of fixed costs: Such as ~~an~~ a clock input, 7-segment display and an IC 7404 Hex Inverter (NOT gate)

~~Calc~~ Calculation: IC 7404 → 1 × Th. 25.59

7-segment display → 1 × Th. 9.85

Total = Th. 35.44

The rest of the circuit consists of 3 JK-Flip-Flops which we can get from 2 × IC 4027. ~~We I~~ I needed 12, 2-input AND gates which I got from 3 × IC 7408 (Quad 2-input AND gates). I also needed 8, 2-input OR gates which I got from 2 × IC 7432 (Quad 2-input OR gates)

Calculation

2 × IC 4027 → 2 × Th. 20.90
3 × IC 7408 → 3 × Th. 23.59
2 × IC 7432 → 2 × Th. 27.59

Total = Th. 167.75

Therefore, the total cost of this circuit is (167.75 + 35.44) = Th. 203.19. This is an estimated price based on component prices taken ~~on~~ from "robodocbd.com"

This is the cheapest one possible. The second cheapest one would be with a NOR gate circuit for the combinational part at around Th. 241.35.

These is the cost analysis, next part is how I built it.

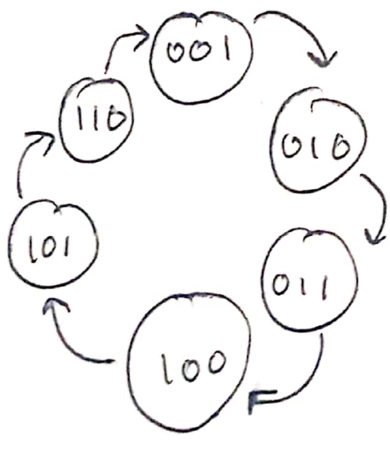
State Table

I constructed a State Table where my states where A, B, C. ~~The output Q of my~~

Current State			Next State			JK Flip Flop input function					
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
x	x	x	x	x	x	x	x	x	x	x	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
1	0	0	1	0	1	x	0	0	x	1	x
1	0	1	1	1	0	x	0	1	x	x	1
1	1	0	0	0	1	x	1	x	1	1	x
x	x	x	x	x	x	x	x	x	x	x	x

State Diagram

A visual of the state table



(As there is no input/output the arrows are not labeled)

K-maps of the state table

$J_A = BC$

	C'	C
$A'B'$	X	0
$A'B$	0	1
AB	X	X
AB'	X	X

$K_A = B$

	C'	C
$A'B'$	X	X
$A'B$	X	X
AB	1	X
AB'	0	0

$J_B = C$

	C'	C
$A'B'$	X	1
$A'B$	X	X
AB	X	X
AB'	0	1

$K_B = A + C$

	C'	C
$A'B'$	X	X
$A'B$	0	1
AB	1	X
AB'	X	X

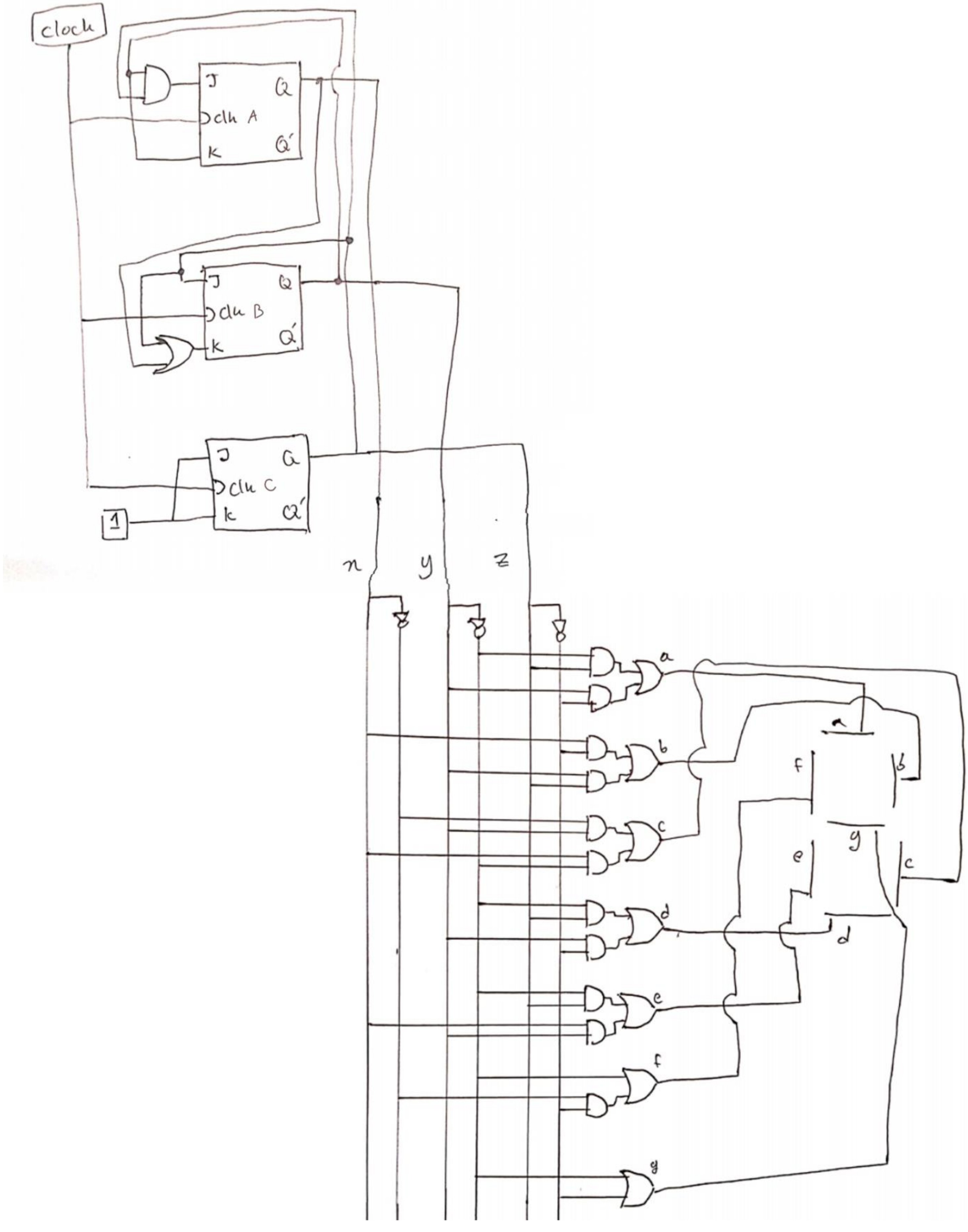
$J_C = 1$

	C'	C
$A'B'$	X	X
$A'B$	1	X
AB	1	X
AB'	1	X

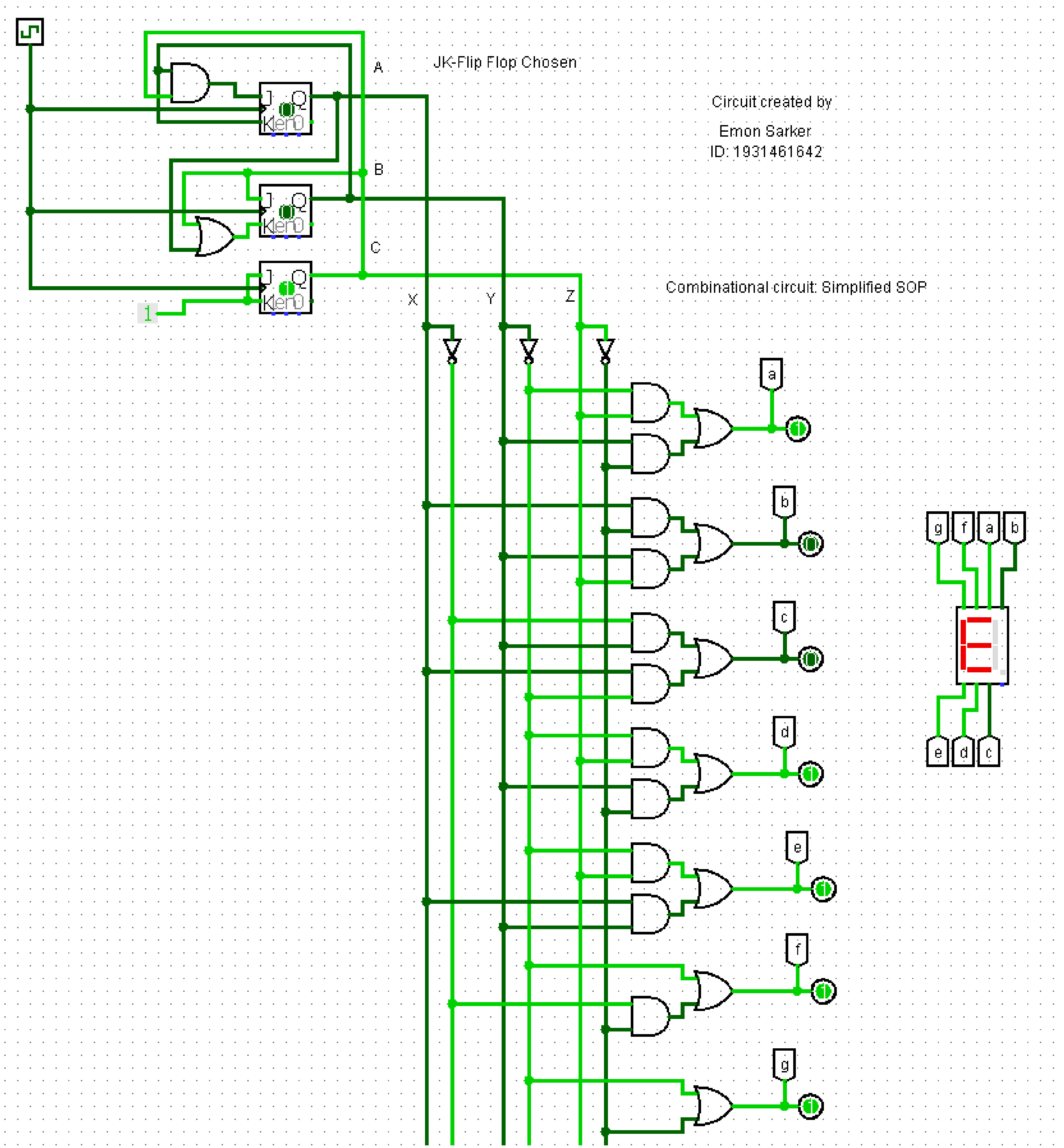
$K_C = 1$

	C'	C
$A'B'$	X	1
$A'B$	X	1
AB	X	X
AB'	X	1

Hand drawn



Logisim Circuit



Discussion & conclusion

In this project, we were given a unique code to print using a 7-segment display. In part one of the project we built a circuit that did so and we called it a combinational circuit.

In the discussion and conclusion ~~of~~ included in my project phase 1, I go in-depth about 7-segment displays; how to construct various combinational circuits using truth tables, k-maps, Boolean algebra and so on. Therefore, for ~~this~~ the discussion and conclusion about the project phase 2, I'll only talk about the newer additions to the project.

Combinational circuits cannot hold information as it does not have any memory. This also means there is no state. ~~or in~~ However, sequential circuits do have memory, therefore they can interact with states based on inputs. ~~and s.~~ This is done by utilizing previous input, clock and a memory element.

Scanned with CamScanner

Flip-Flops are the memory elements we learned about in this course. There are three types of flip-flops: JK flip-flop, T flip-flop, D flip flops. These flip-flops take in a clock pulse, input and gives out an output and a inverted output.

These flip-flops cycle through states of 0 and 1. We can make them hold many information by using a lot of flip flops. For example: 3 flipflops can store 3-6.7 info.

We cycle through states by giving a specific input function to the flip flops. The minimum inputs that are necessary to generate a "next state" when the correct state is known are plotted down in an excitation table.

I used a JK-flipflop for my memory element and wanted to cycle through 7's states in this order: (001, 010, 011, 100, 101, 110) which I was able to by ~~uti~~ creating a state table and utilizing JK Flipflop excitation table.

Excitation table for ~~J-K~~ JK FlipFlop

Q	next Q	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

After that I found out the input for JK Flip Flop functions by utilizing K-maps.

Afterwards a circuit was created, which cycled through states as mentioned (also shown in the state diagram).

Then I attached the output of the Flip Flop to my previously built combinational circuit in this fashion: C → z, B → y, A → ~~z~~ x (shown in diagram)

Finally, when the simulation is turned on. It automatically ~~cy~~ cycles through the code "ES1462" without user input.