Gebze Technical University Department of Computer Engineering
# CSE 331 – Homework 3 Report

DİLARA KARAKAŞ
171044010

Data Path Design

      When 32-bit multiplication is done, a 64-bit product is created. Logisim does not support 64 bit, so I divided it into 32-32. On the least significant bit side there is the product on the multiplier most significant bit side.

      I did not use a ready-made arithmetic circuit in my project. I designed it all myself and embedded it in the data path.

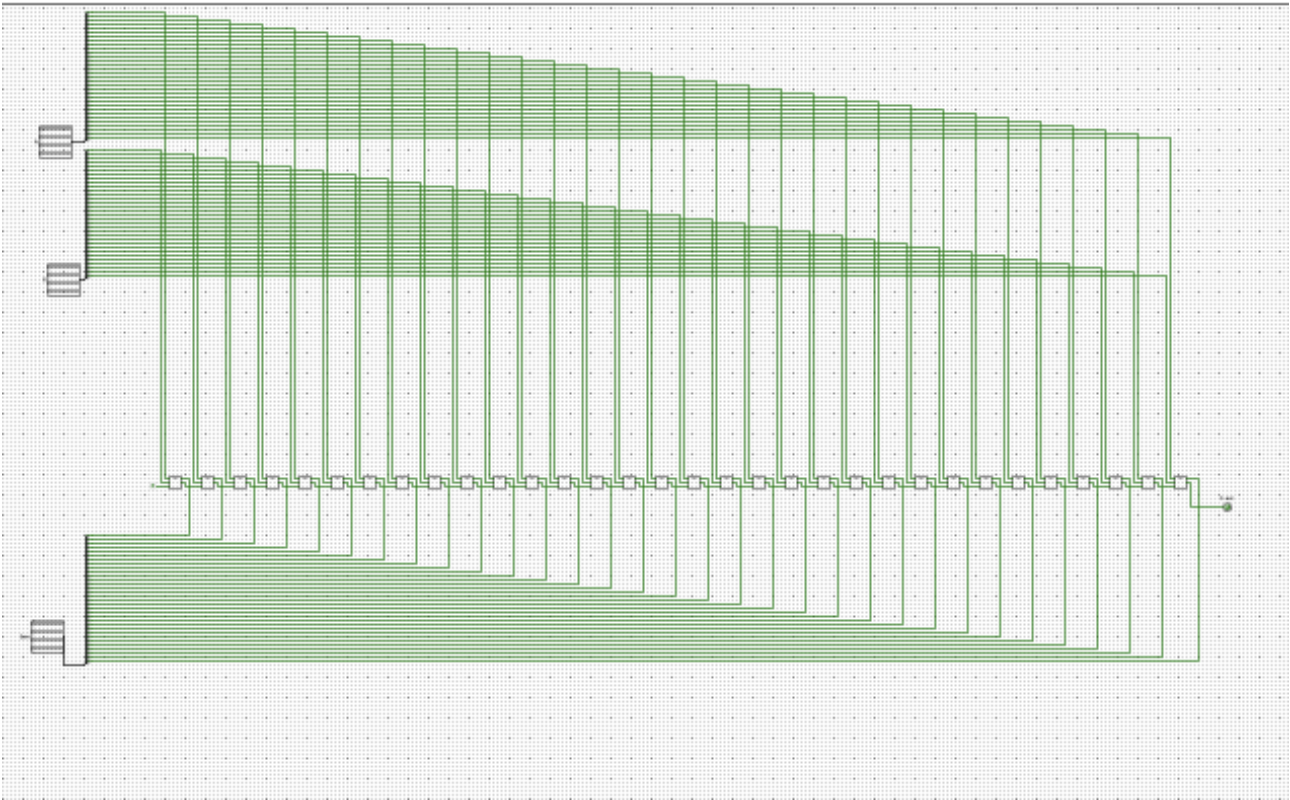## 1. 1-Bit Adder
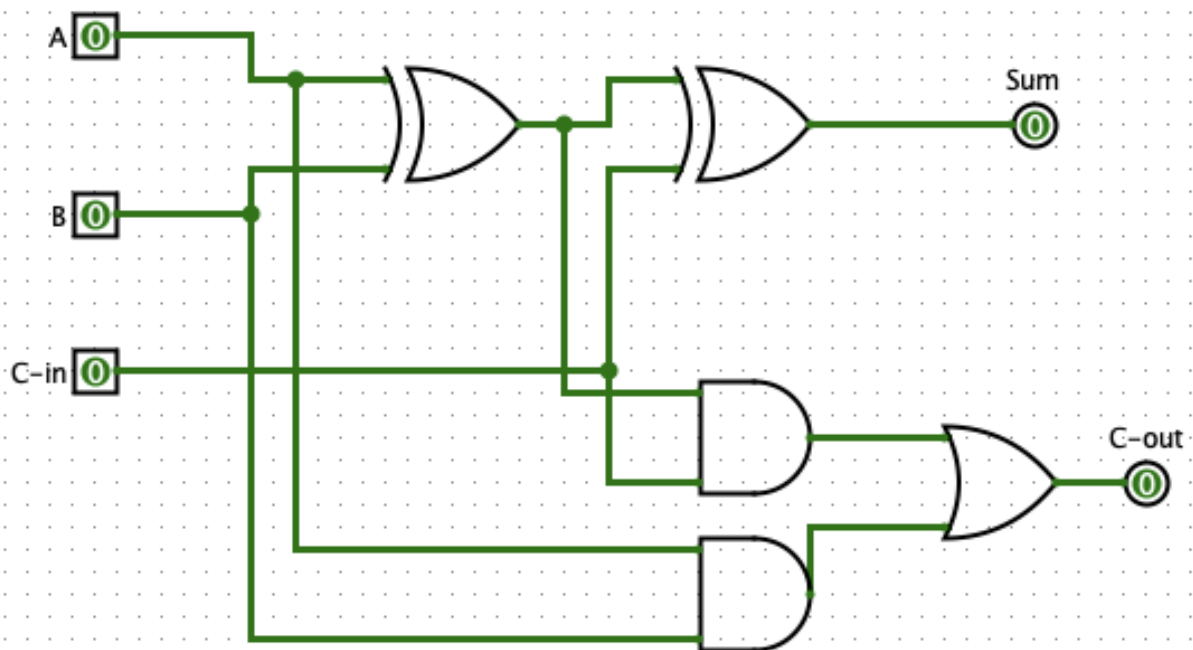
2. 6-Bit Adder
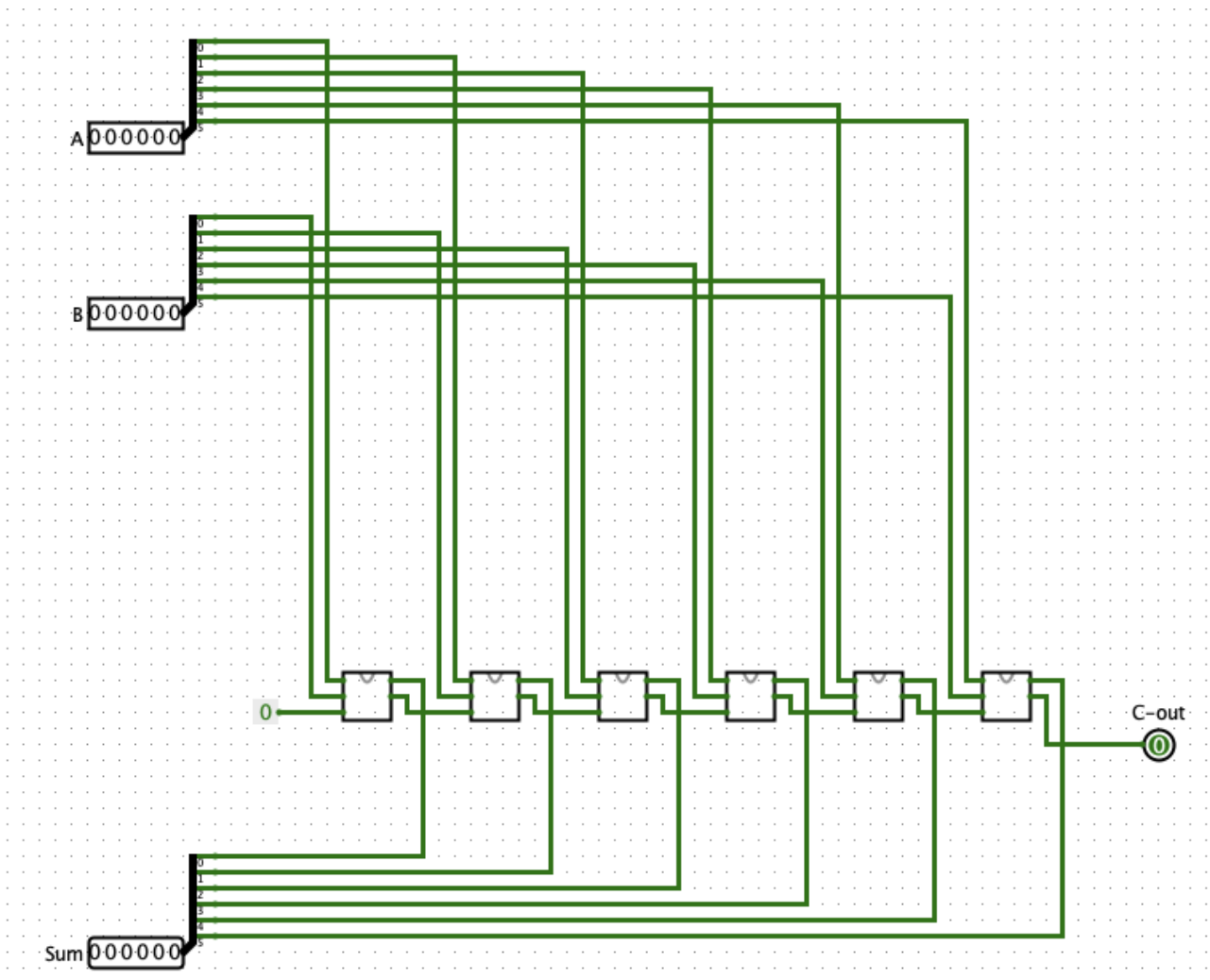
The reason I designed this is because I had to count count values in the program. This count value is counting how many shifts have been made. If he shifts 32 times, he ends the program. It means we found the result.

3. 32-Bit Adder

4. 64-Bit Shifter
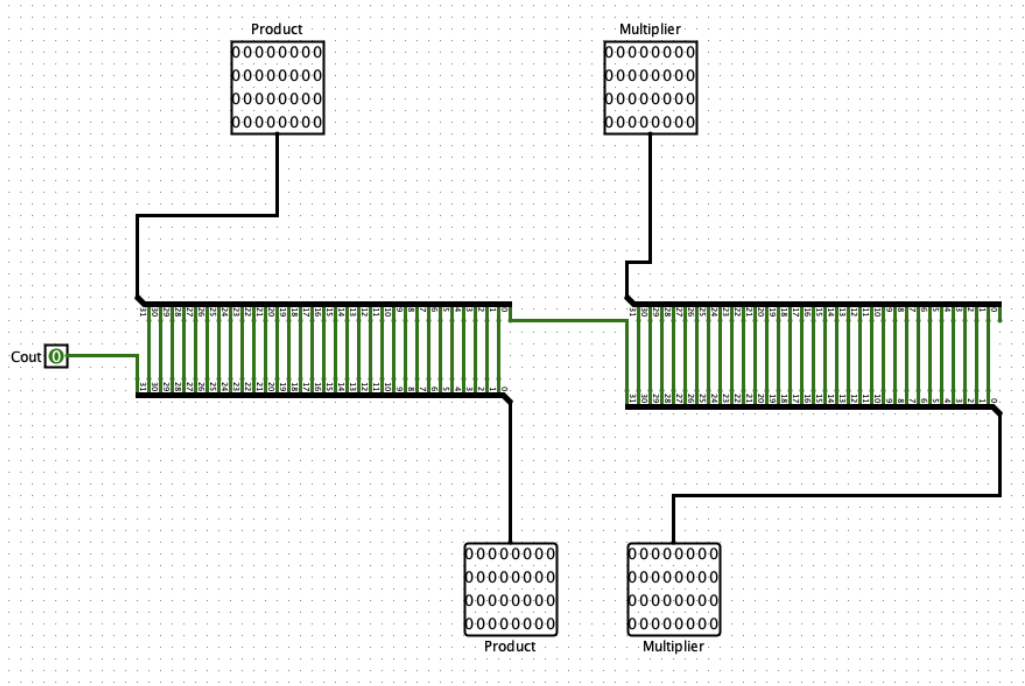
5. Equal (Comparator)
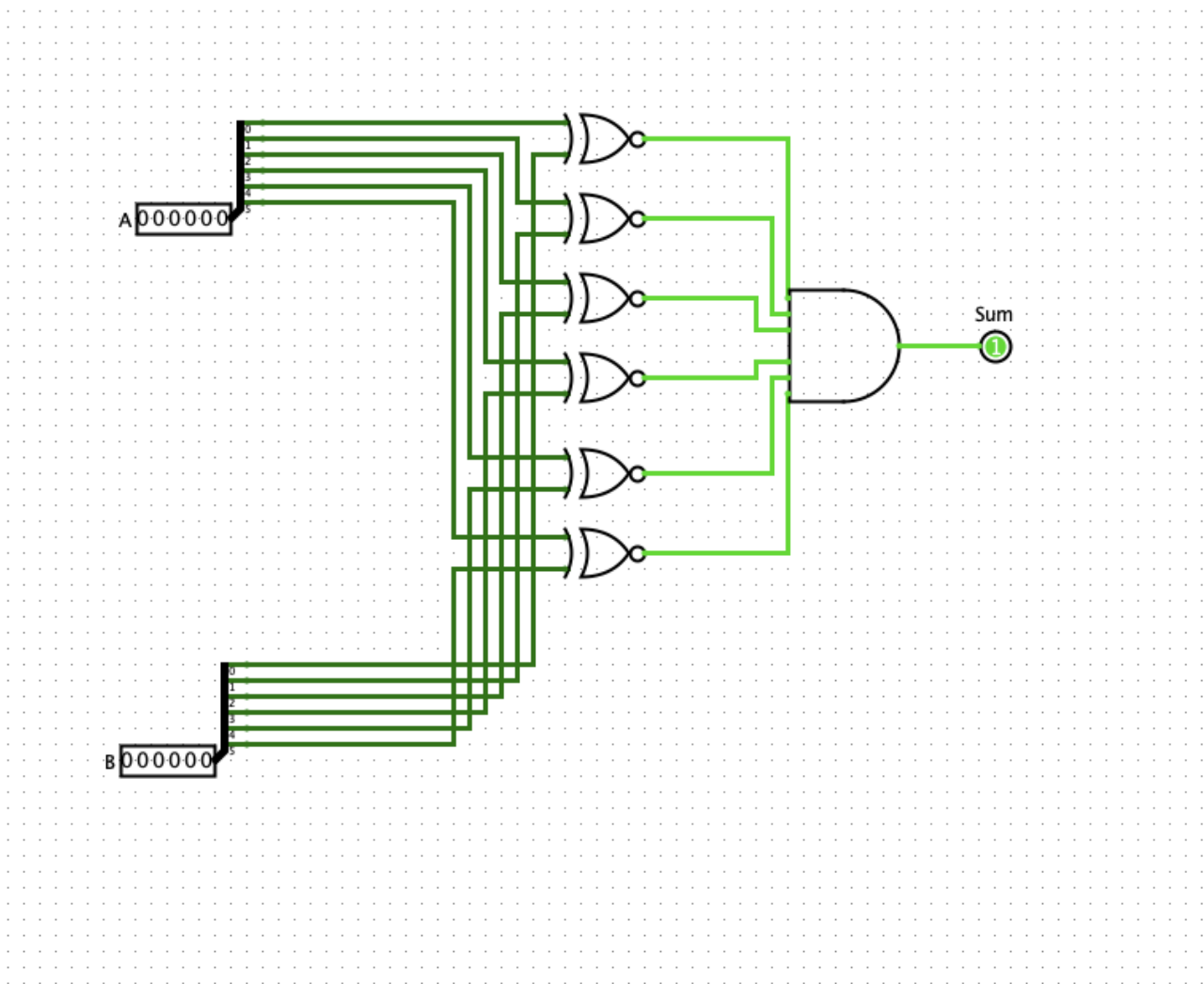
I prepared it to check if the shift number is 32 times.

6. Datapath
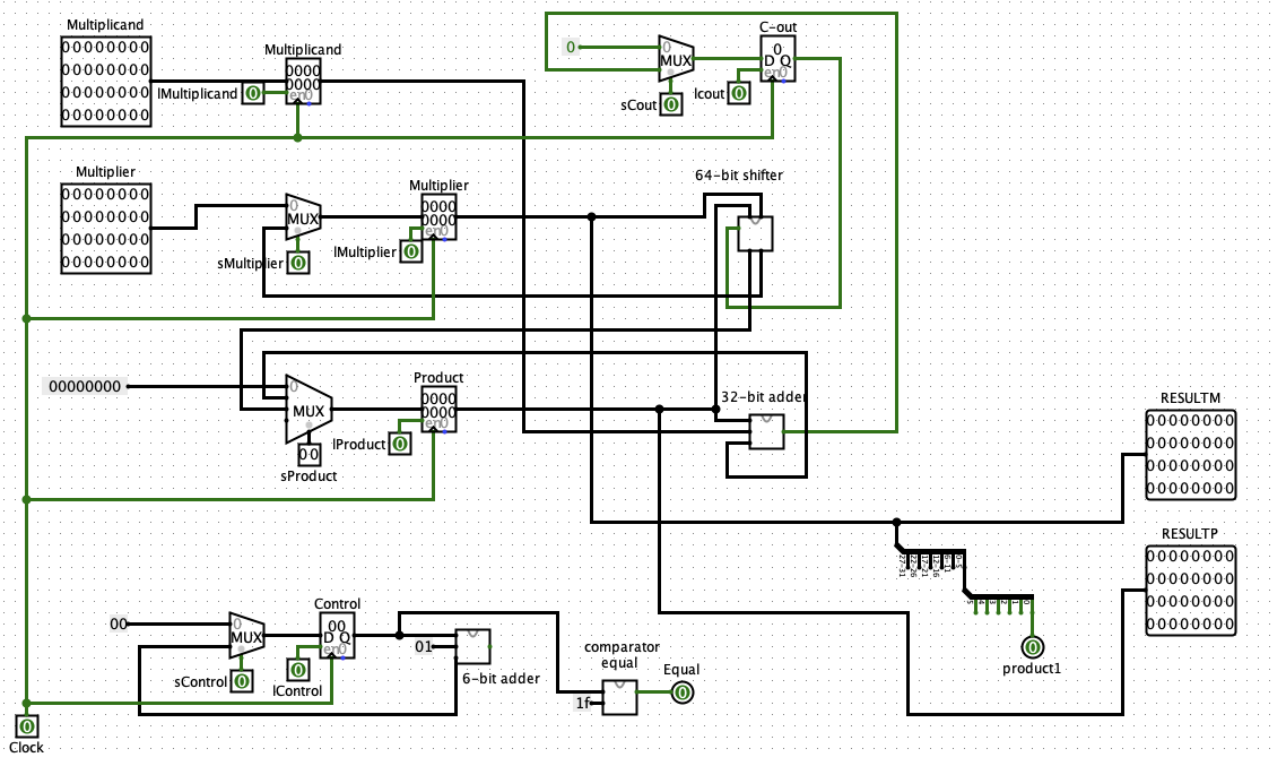
CONTROL UNIT DESIGN

1. STATE DIAGRAM
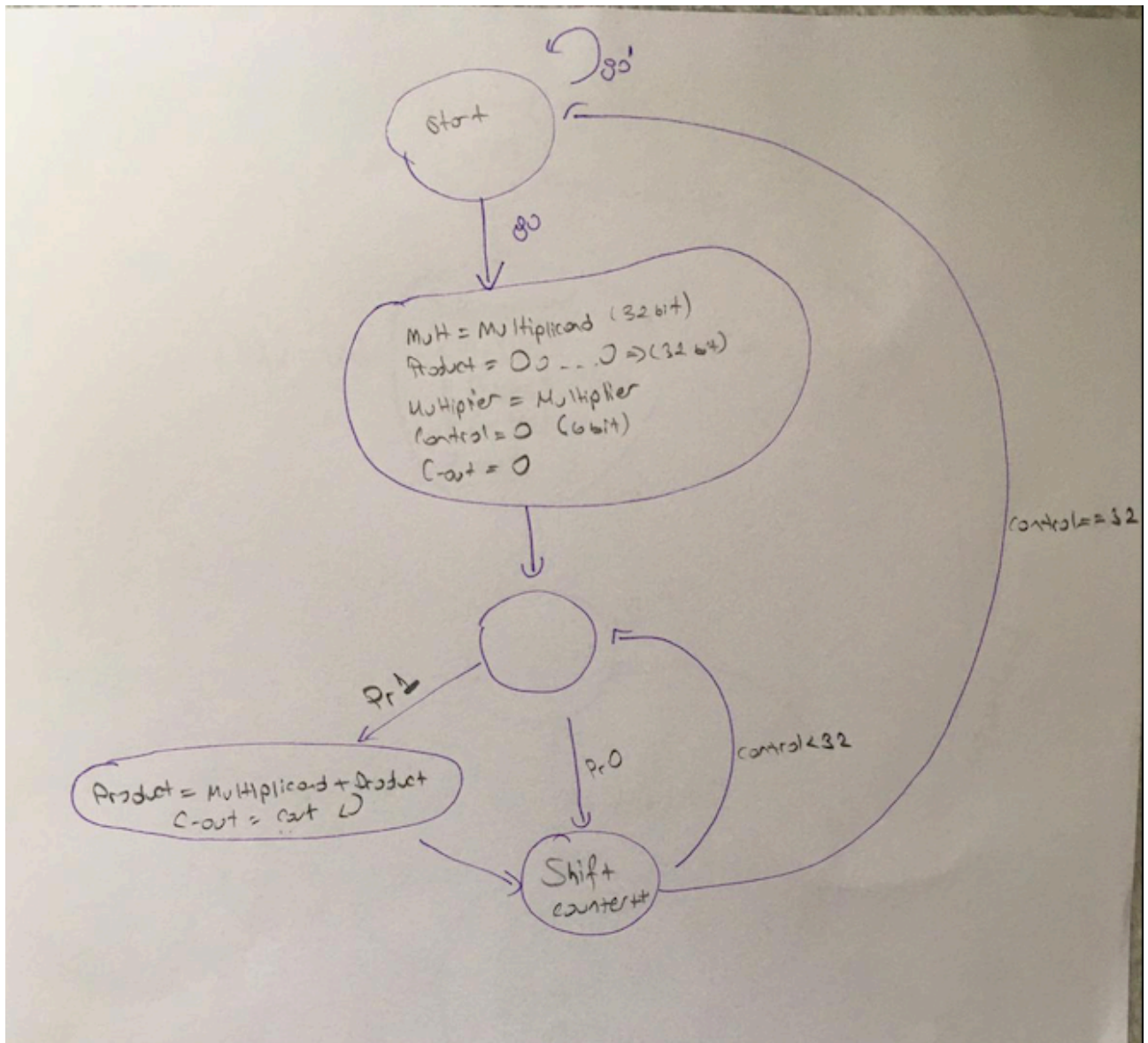
After placing the necessary signals in the data path

000 — start

go' (self-loop on start)

go →

(S1) 001
- ℓMultiplicand
- ℓMultiplier
- ℓProduct
- ℓControl
- ℓcout

010

Product1==1 →

(S3) 011
- sProduct=01
- ℓProduct
- sCout
- ℓCout

Product1=0 →

(S4) 100
- sMultiplier
- ℓMultiplier
- sProduct=10
- ℓProduct
- sControl
- ℓControl

equal1==1

equal1==0

equal1==1
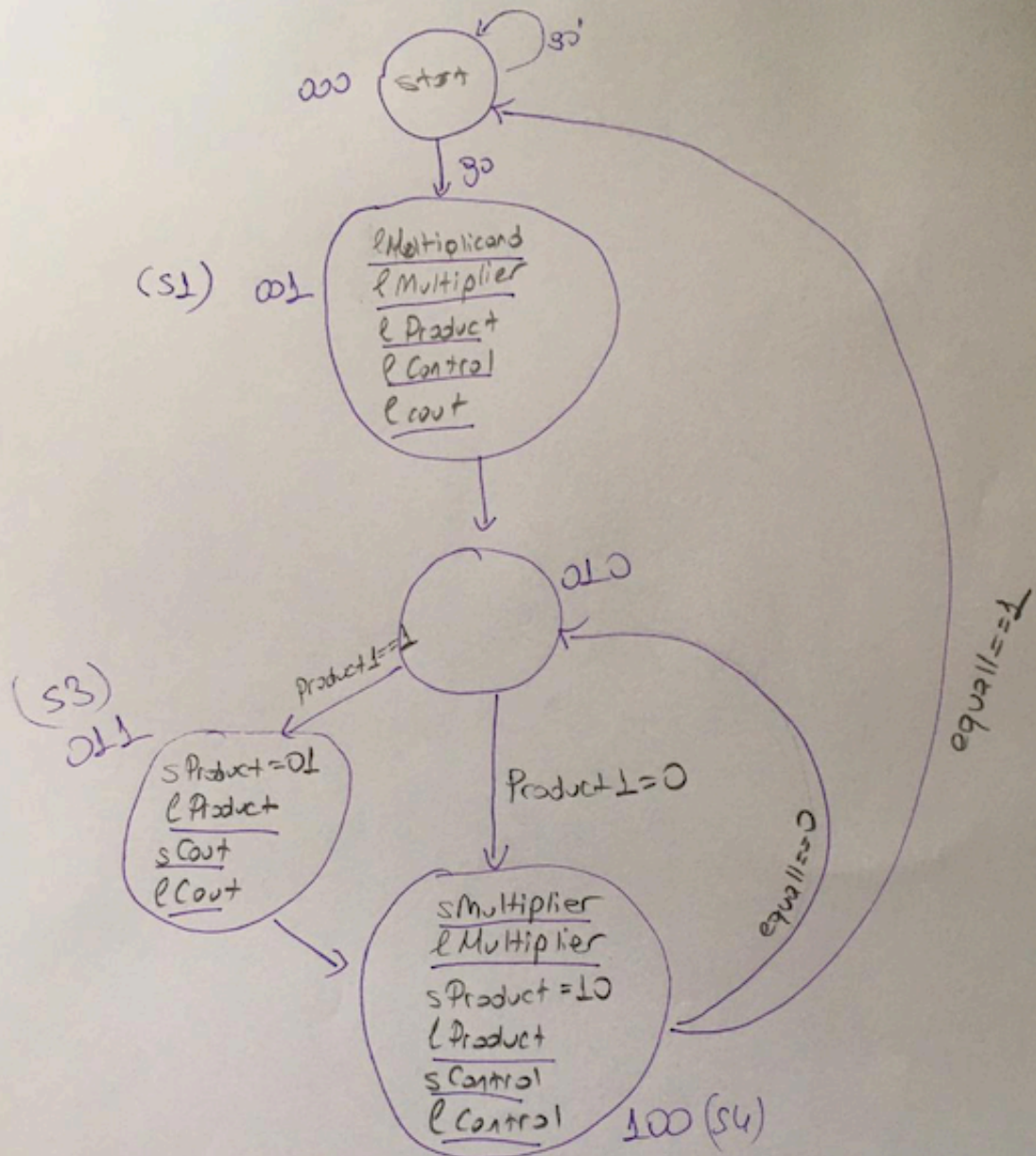
2. Boolean Table

S2 S1 S0	Go	Product1	Equal	N2	N1	N0
0 0 0	0	x	x	0	0	0
0 0 0	1	x	x	0	0	1
0 0 1	x	x	x	0	1	0
0 1 0	x	1	x	0	1	1
0 1 0	x	0	x	1	0	0
0 1 1	x	x	x	1	0	0
1 0 0	x	x	0	0	1	0
1 0 0	x	x	1	0	0	0

N2 = s2's1s0'Product1' + s2's1s0
N1 = s2's1's0 + s2's1s0'Product1 + s2s1's0'Equal'
N0 = s2's1's0'Go + s2's1s0'Product1

3. FSM

TEST RESULTS

Multiplicand
```
00000000
00000000
00000000
00000010
```

Multiplier
```
00000000
00000000
00000000
00000110
```

START

RESULT
```
00000000    00000000
00000000    00000000
00000000    00000000
00000000    00001100
```

Multiplicand
```
00000000
00000000
00001100
00000010
```

Multiplier
```
00000000
00000000
00000000
00110110
```

START

RESULT
```
00000000    00000000
00000000    00000010
00000000    10001000
00000000    01101100
```

Multiplicand
```
00000000
00000000
00000000
00100010
```

Multiplier
```
00000000
00000000
00000000
00010000
```

START **0**

RESULT
```
00000000        00000000
00000000        00000000
00000000        00000010
00000000        00100000
```


Multiplicand
```
00000000
00000000
00001100
00100000
```

Multiplier
```
00000000
00000000
00000010
00010000
```

START **0**

RESULT
```
00000000        00000000
00000000        00011001
00000000        00000010
00000000        00000000
```

Multiplicand
```
11111111
11111111
11111111
11111110
```

Multiplier
```
00000000
00000000
00000000
00000010
```

START 0

RESULT
```
00000000   11111111
00000000   11111111
00000000   11111111
00000001   11111100
```

Multiplicand
```
00000000
00000000
00000000
00000010
```

Multiplier
```
00000000
00000000
00000000
00001110
```

START 0

RESULT
```
00000000   00000000
00000000   00000000
00000000   00000000
00000000   00011100
```