ECE M16 Final

Lawrence Liu

August 10, 2022

Problem 1

	1	1	0	1
1	11	00	00	10
	1			
101	10	00		
	1	01		
1100		11	00	
		0		
		11	00	10
11001		1	10	01
		1	10	01

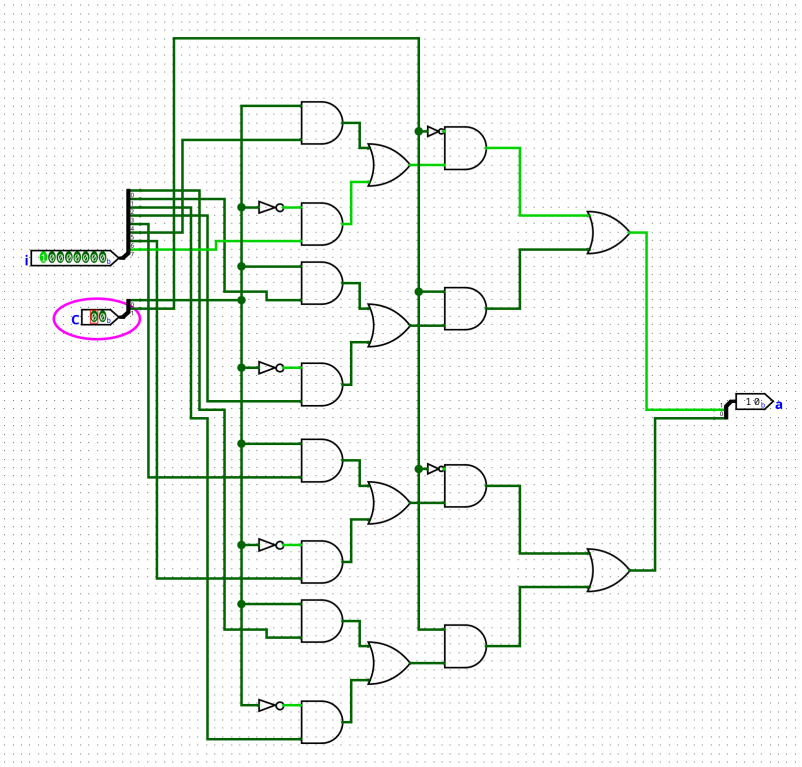
Problem 2

Basing on the assumption that $c[1:0] = 11$ corresponds with $o[1:0] = i[1:0]$ we have that

$$\begin{aligned} a[1] &= \overline{c[1]}.\overline{c[0]}.i[7] + \overline{c[1]}.c[0].i[5] + c[1].\overline{c[0]}.i[3] + c[1].c[0].i[1] \\ &= \overline{c[1]}.(\overline{c[0]}.i[7] + c[0].i[5]) + c[1].(\overline{c[0]}.i[3] + c[0].i[1]) \end{aligned}$$

$$\begin{aligned} a[0] &= \overline{c[1]}.\overline{c[0]}.i[6] + \overline{c[1]}.c[0].i[4] + c[1].\overline{c[0]}.i[2] + c[1].c[0].i[0] \\ &= \overline{c[1]}.(\overline{c[0]}.i[6] + c[0].i[4]) + c[1].(\overline{c[0]}.i[2] + c[0].i[0]) \end{aligned}$$

Which results in a circuit like this



Problem 3

I created the circuit, and it is shown above, and I tested it with the following python checker srcipt

```python
import numpy as np
import pandas as pd
import os
from calendar import monthrange

def RunCircuit(logisim_jar  : str, circuit : str):
    """
    This function runs the logisim simulator and returns the output of the circuit as
    a  pandas dataframe.
    """
    output=os.popen(f"java -jar {logisim_jar} {circuit} -tty table").read()
    output=[o.split() for o in output.split("\n")[:-1]]
    return pd.DataFrame(output[1:],columns=output[0])

def checkQ2(truth_table:pd.DataFrame)->bool:
    """
    This function checks the output of the circuit for the truth table and returns
    weather the output is correct or not.
```

```
    """
    #convert hex to binary
    truth_table['i']=truth_table['i'].apply(lambda x: f'{int(x,16):0>8b}')
    for i,row in truth_table.iterrows():
        c=int(row.C,2)
        i=row.i
        a=row.a
        #calculate a expected
        a_expected=i[c*2:c*2+2]
        #check if a is equal to a_expected
        if a!=a_expected:
            print("error!")
            print(f"at c={row.C}")
            print(f"expected a={a_expected}")
            print(f"got a={a}")
            print(f"i={row.i}")
            return False
    return True

if __name__=="__main__":
    truth_table=RunCircuit("../logisim-evolution.jar","Logisim/FinalQ3.circ")
    if checkQ2(truth_table):
        print("Q2 passed!")
```

This script utilizes Logisim's command line ability. I had the files in the following format

```
ECEM16
 |- .gitignore
 |- Final
 | |- Logisim
 | | |- FinalQ3.circ
 | :
 | :
 | |- checker.py
 |- .gitignore
 |- logisim-evolution.jar
```

Problem 4

Let the output