San Francisco Bay University

CE305 - Computer Organization
2023 Fall Homework #3

Due day: 10/29/2023

Instruction:

1. **Homework answer sheet should contain the original questions and corresponding answers.**
2. **Answer sheet must be in PDF file format with Github links for the programming questions, but MS Word file can't be accepted. As follows is the answer sheet name format.**
   <course_id>_week<week_number>_StudentID_FirstName_LastName.pdf
3. **The program name in Github must follow the format like**
   <course_id>_week<week_number>_q<question_number>_StudentID_FirstName_LastName
4. **Show screenshot of all running results, including the system date/time.**
5. **Only accept homework submission uploaded via Canvas.**
6. **Overdue homework submission can't be accepted.**
3. **Takes academic honesty and integrity seriously (Zero Tolerance of Cheating & Plagiarism)**

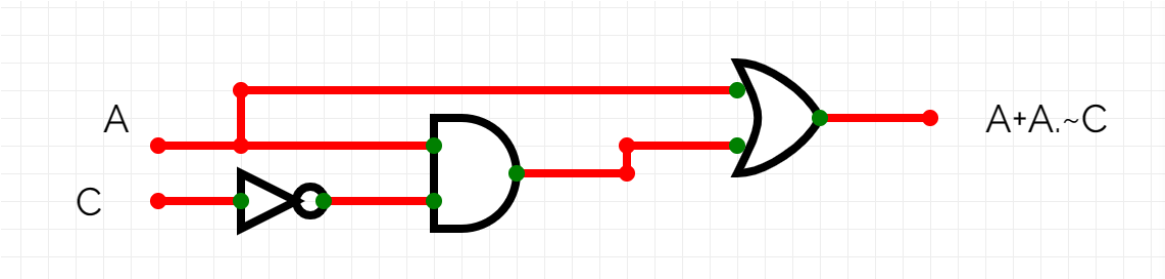
1. Simplify the following Boolean logic functions to the format in sum of product first and then create a truth table in Excel for each as the verification reference of the circuit design, finally implement by online tools at *https://circuitverse.org/simulator*

   a. $f = (A + \bar{C} + \bar{D})(\bar{B} + \bar{C} + D)(A + \bar{B} + \bar{C})$

   → $= (A + \neg C + \neg D)(\neg B + \neg C + D)(A + \neg B + \neg C)$

$= (A + \neg C + \neg D)(\neg B + \neg C + D)(A + \neg B + \neg C)$
$= (A \cdot \neg B \cdot A + A \cdot \neg B \cdot \neg C + A \cdot \neg C \cdot \neg D + \neg C \cdot \neg B \cdot A + \neg C \cdot \neg B \cdot \neg C + \neg C \cdot \neg B \cdot D + \neg D \cdot \neg C \cdot A + \neg D \cdot \neg C \cdot \neg C + \neg D \cdot \neg C \cdot \neg D)$
$= (A + 0 + A \cdot \neg C + 0 + 0 + 0 + 0 + 0 + 0 + 0)$
$= A + A \cdot \neg C$

A	B	C	D	~C	F = A+A.(~C)
0	0	0	0	1	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	1	0	1



b. $f = (Z + X)(\bar{Z} + \bar{Y})(\bar{Y} + X)$

→$(Z + X)(!Z + !Y)(!Y + X)$

= (Z + X)(!Z + !Y)(!Y + X)

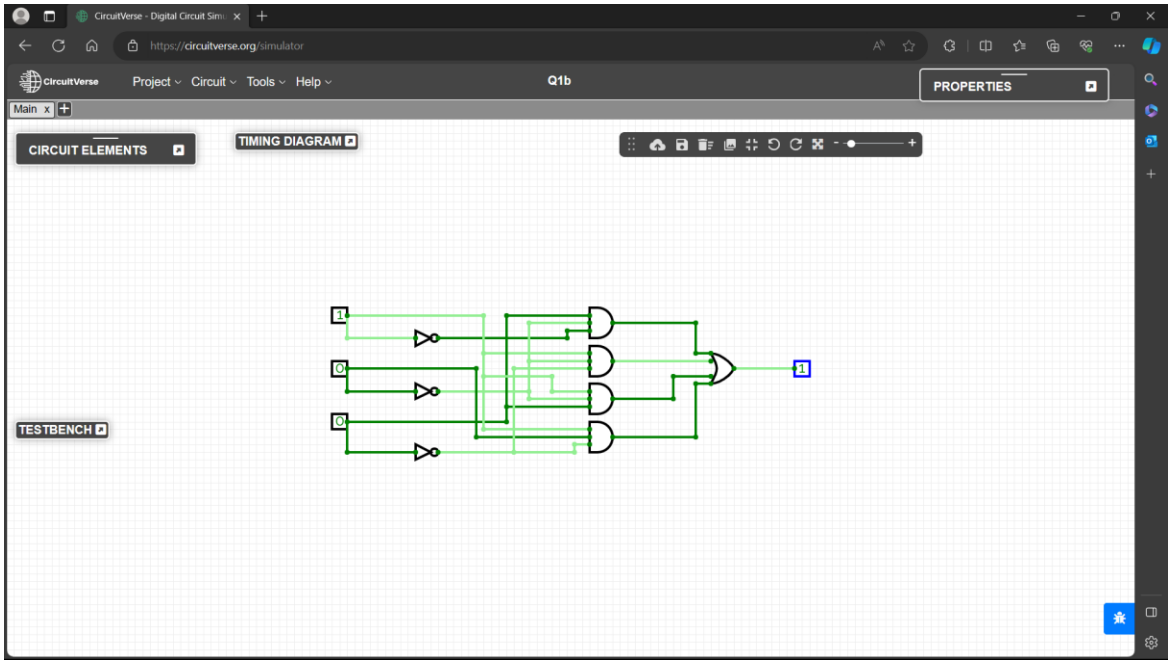
= (Z!Z + Z!Y + X!Z +X!Y)(!Y + X)

= (0 + Z!Y + X!Z +X!Y)(!Y + X)

= Z!Y!Y + X!Z!Y +X!Y!Y + Z!YX + XX!Z +XX!Y

= Z!Y + X!Z!Y + Z!YX + X!Z + X!Y

X	Y	Z	~X	~Y	~Z	Z~Y	X~Z~Y	Z~YX	X~Z	X~Y	f
0	0	0	1	1	1	0	0	0	0	0	0
0	0	1	1	1	0	1	0	0	0	0	1
0	1	0	1	0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	1	0	1	1	1
1	0	1	0	1	0	1	0	1	0	1	1
1	1	0	0	0	1	0	0	0	1	0	1
1	1	1	0	0	0	0	0	0	0	0	0

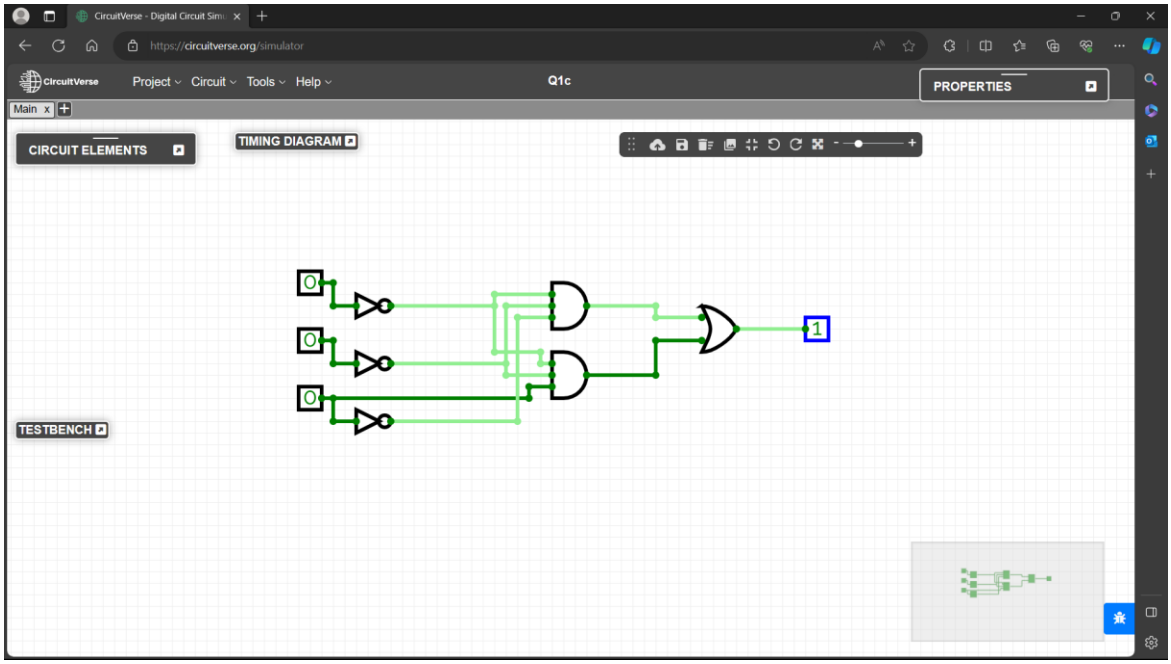


c. $f = \overline{(X+Y)}Z + \bar{X}\bar{Y}\bar{Z}$

$=!(X+Y)Z + !X!Y!Z$

$= !X!YZ + !X!Y!Z$

X	Y	Z	~X	~Y	~Z	~X~YZ	~X~Y~Z	f
0	0	0	TRUE	TRUE	TRUE	FALSE	TRUE	1
0	0	1	TRUE	TRUE	FALSE	TRUE	FALSE	1
0	1	0	TRUE	FALSE	TRUE	FALSE	FALSE	0
0	1	1	TRUE	FALSE	FALSE	FALSE	FALSE	0
1	0	0	FALSE	TRUE	TRUE	FALSE	FALSE	0
1	0	1	FALSE	TRUE	FALSE	FALSE	FALSE	0
1	1	0	FALSE	FALSE	TRUE	FALSE	FALSE	0
1	1	1	FALSE	FALSE	FALSE	FALSE	FALSE	0

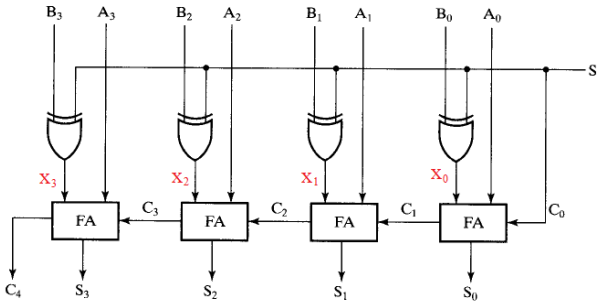


Notes:

- *The simulation tools provide users with a lot of help for circuit designs by clicking "**Help**" button on the tool bar.*
- *Each design should be exported as "**.cv**" file by "**Export as file**" in "**Project**" tab on the tool bar and submitted with the truth table.*

2. Design signal bit full adder based on the truth table and circuit on chapter 3 lecture handouts first and then create 4-bit "Adder-Subtractor circuit" as follows to implement arithmetic addition and subtraction operations.

- If $S = 0$ in the circuit, then $X_3 = B_3, X_2 = B_2, X_1 = B_1$ and $X_0 = B_0$, so the adder circuit simply adds A and B when $C_0 = S = 0$ (carry in = 0).
- If $S = 1$, then $X_3 = \overline{B_3}, X_2 = \overline{B_2}, X_2 = \overline{B_2}$ and $X_0 = \overline{B_0}$. Since $C_0 = S = 1$, the circuit is equivalent to adding the 2's complement of B to A, that is, implementing subtraction operation "A-B".



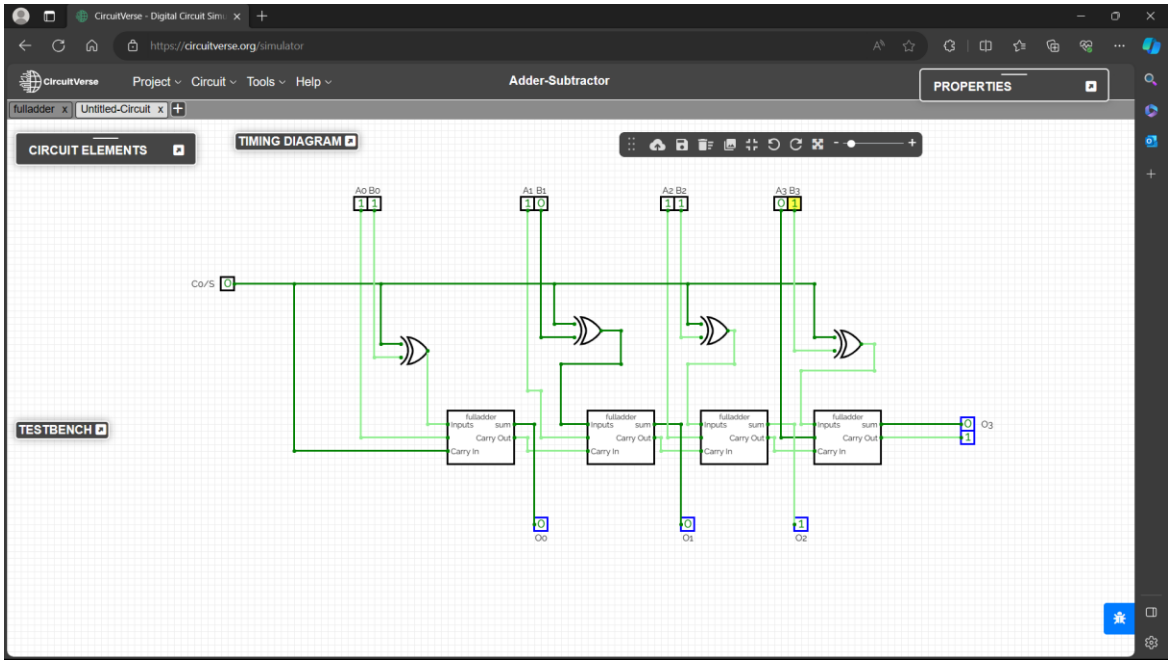
Verify your design with the following testcases by the conversion from decimal to binary as inputs A and B

(a) A + B = 7 + (-3) = 4

(b) 7 = 0111

(c) -3 = 1101

(d) 4 = 0100

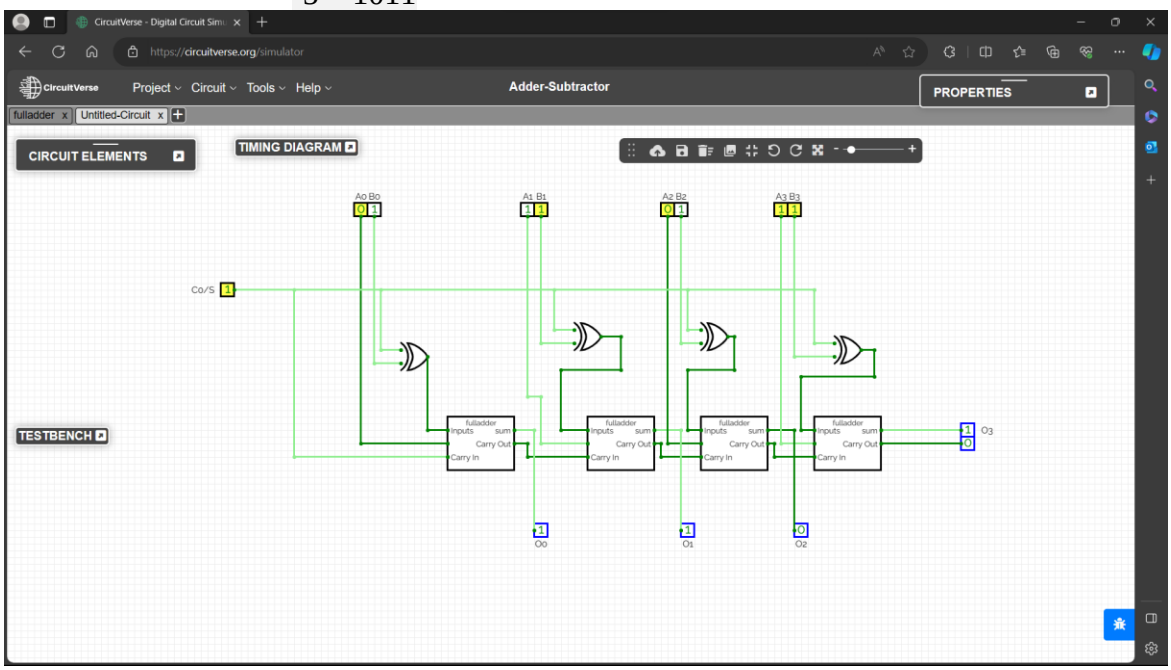


(b) A - B = -6 - (-1) = -5

-6 = 1010

-1 = 1111

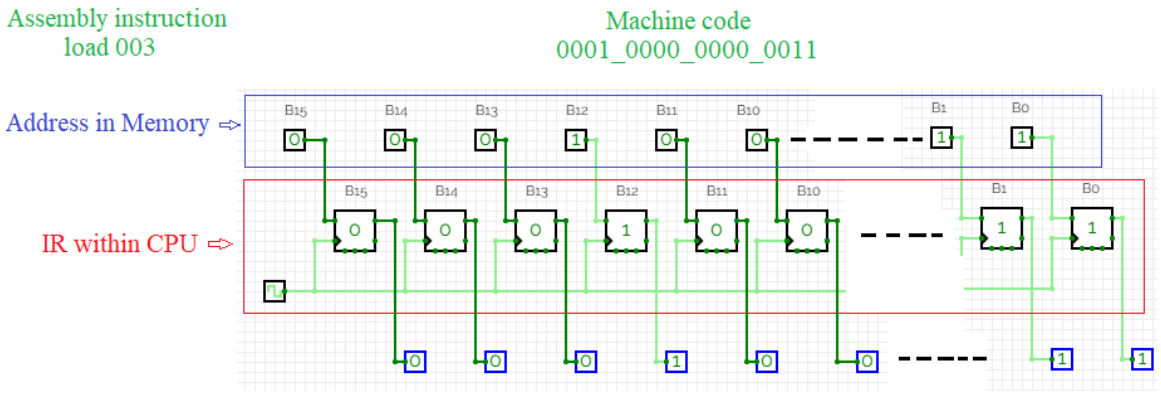
-5 = 1011



Notes:

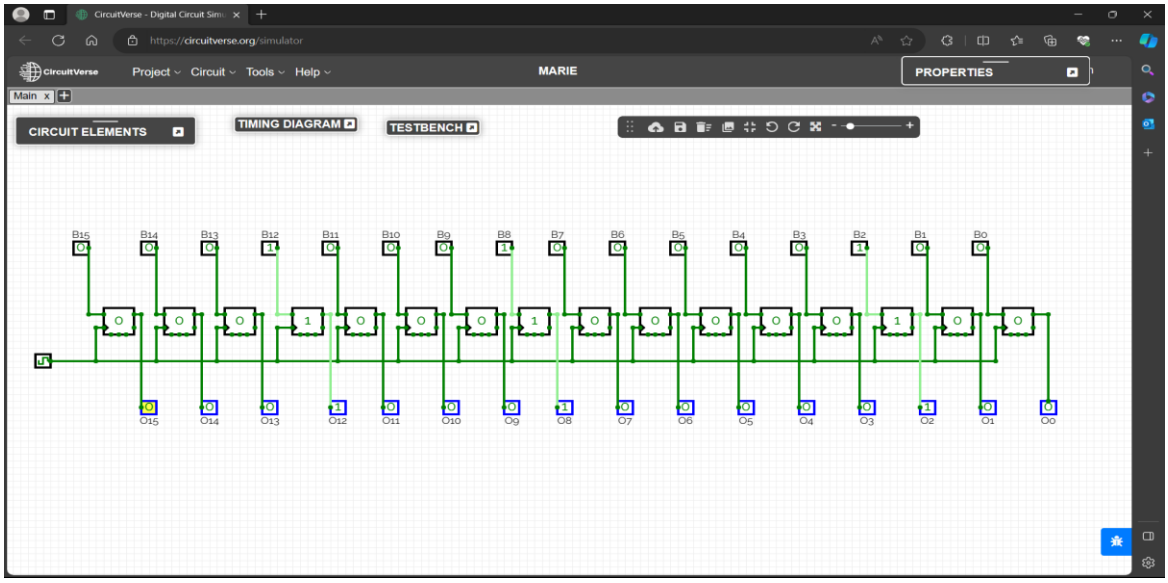
- *The design should be exported as "**.cv**" file by "**Export as file**" in "**Project**" tab on the tool bar and submitted with the truth table.*

3. The following circuit is to simulate the MARIE assembly instruction "load 003", which means that it moves this instruction saved in the certain location of the memory to the instruction register (IR) using 16 D-Flip Flops within CPU.

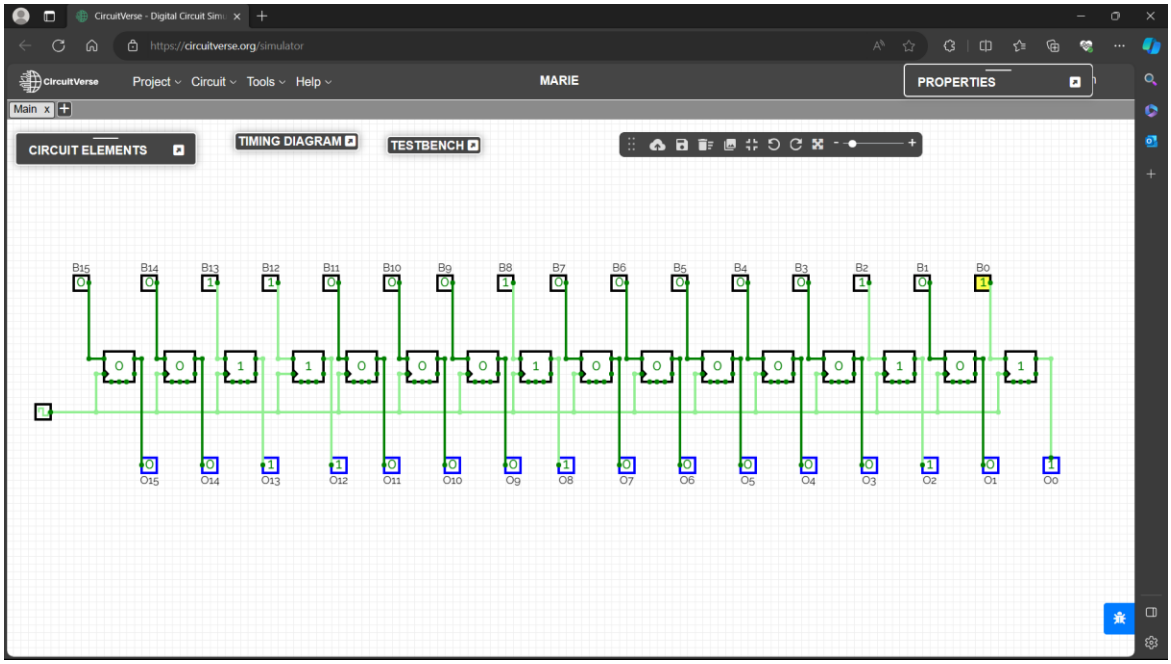


Based on the above circuit, please implement the following MARIE assembly instructions by translating assembly code to machine code depending on the given lookup table in chapter 4 lecture handouts

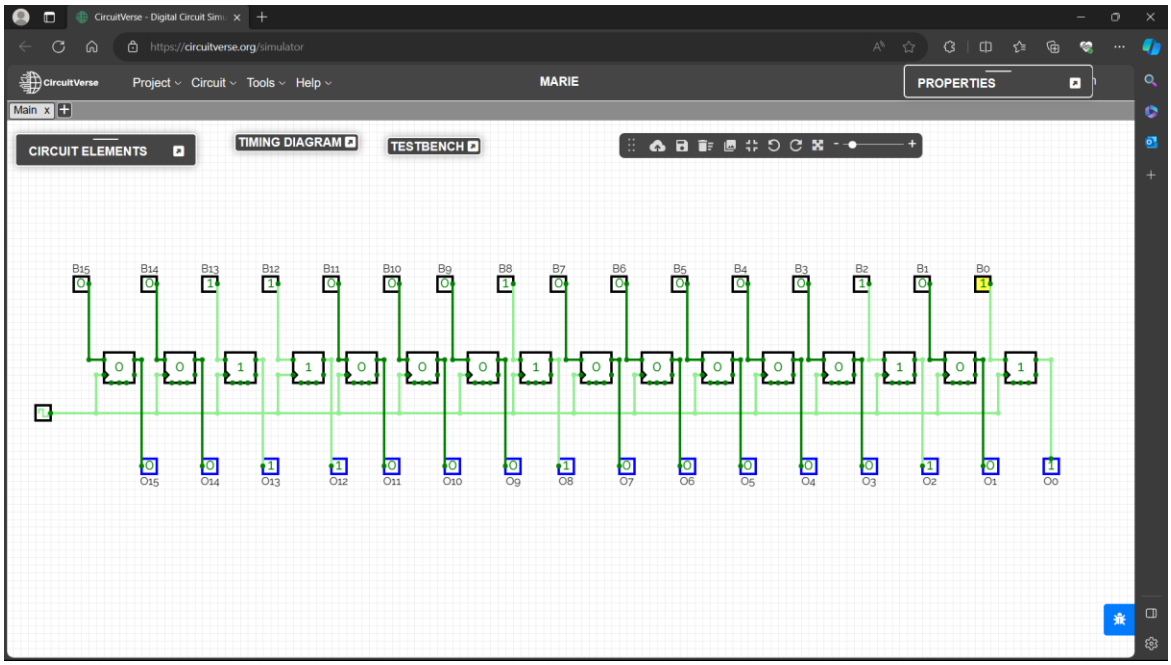
Load 104



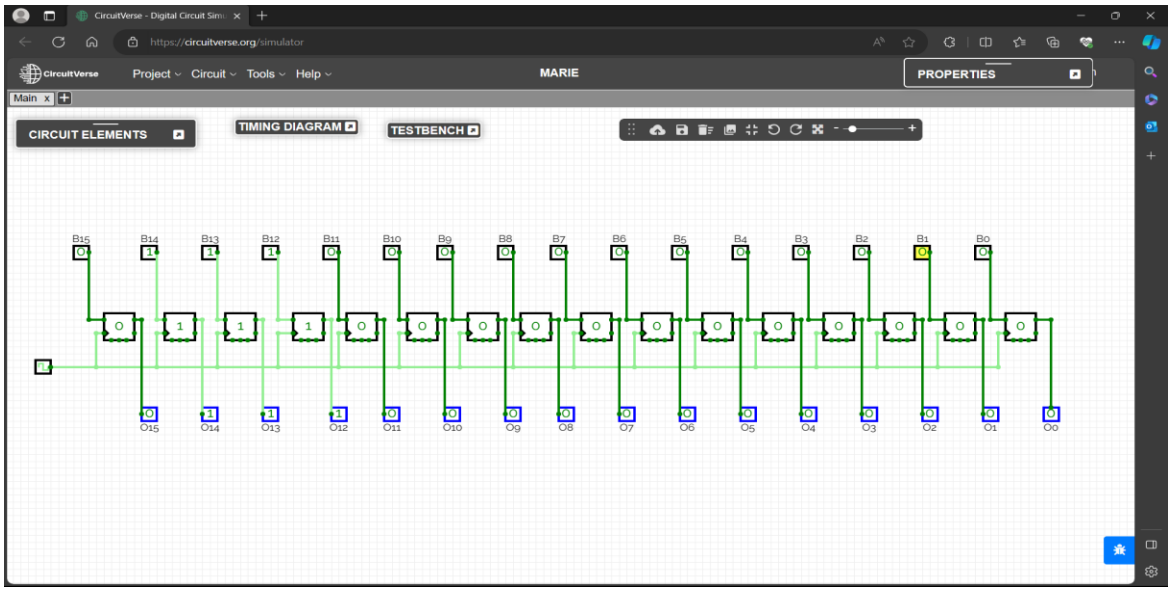
Add 105


CircuitVerse - Digital Circuit Simu
https://circuitverse.org/simulator
CircuitVerse
Project
Circuit
Tools
Help
MARIE
PROPERTIES
Main
CIRCUIT ELEMENTS
TIMING DIAGRAM
TESTBENCH
B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
O15 O14 O13 O12 O11 O10 O9 O8 O7 O6 O5 O4 O3 O2 O1 O0

Store 106


CircuitVerse - Digital Circuit Simu
https://circuitverse.org/simulator
CircuitVerse
Project
Circuit
Tools
Help
MARIE
PROPERTIES
Main
CIRCUIT ELEMENTS
TIMING DIAGRAM
TESTBENCH
B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
O15 O14 O13 O12 O11 O10 O9 O8 O7 O6 O5 O4 O3 O2 O1 O0

Halt



Notes:

- *The design should be exported as "**.cv**" file by "**Export as file**" in "**Project**" tab on the tool bar and submitted with the truth table.*

SWEKCHHA HAMAL, 19700