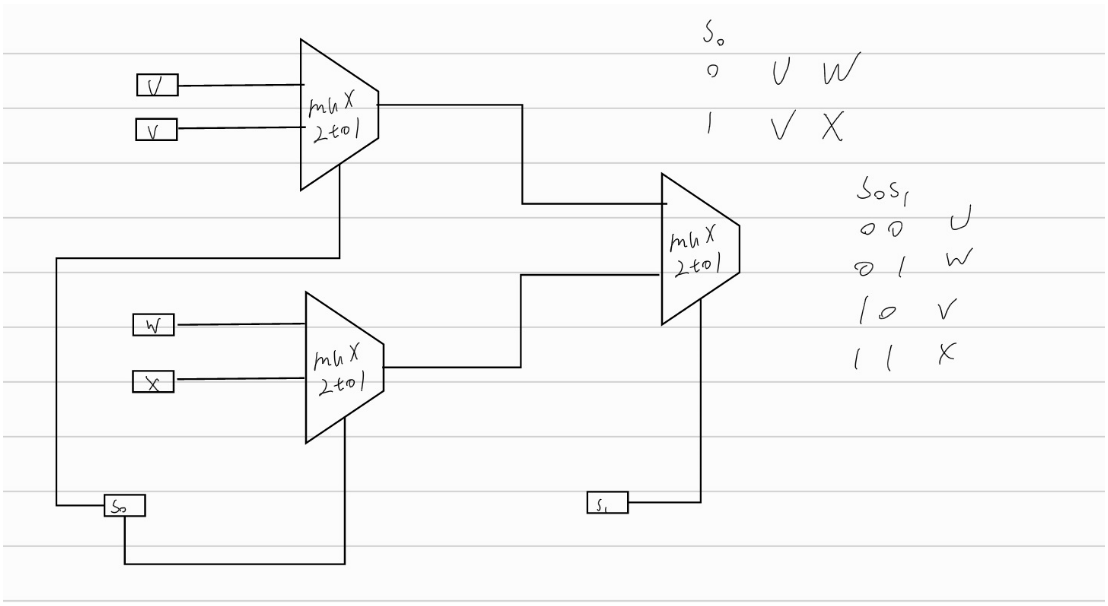
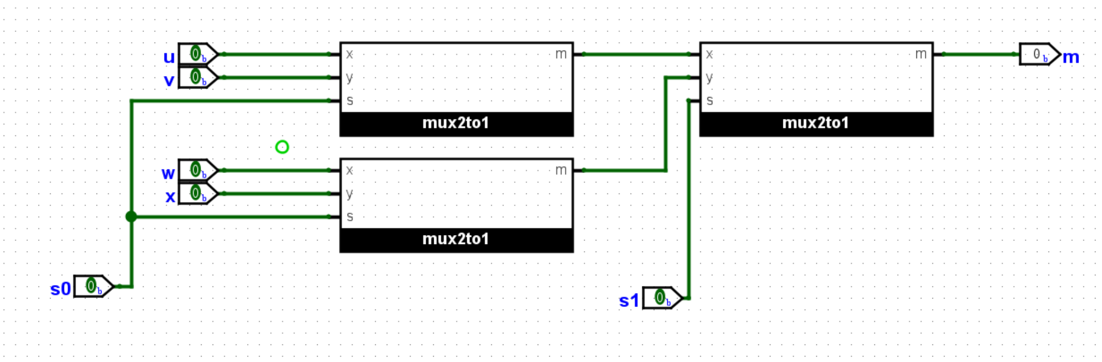
CSC258 Lab Report

Tianle Wang 1006337028

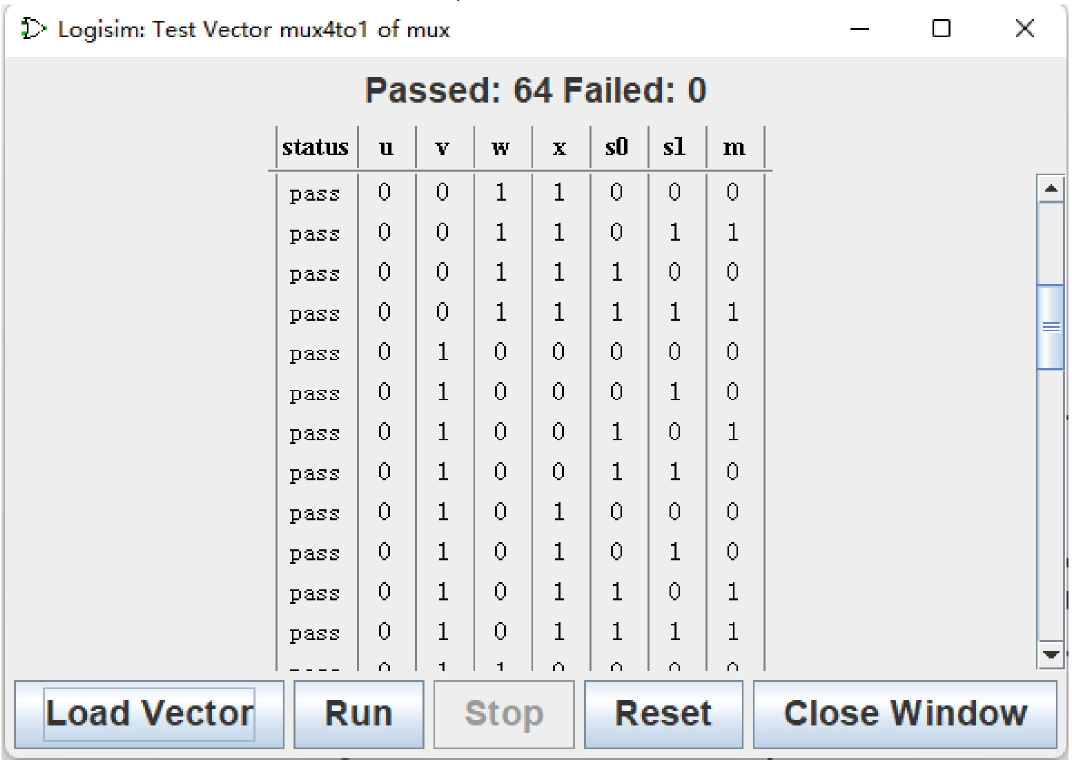
Part II

1. Answer the following question: if the truth table in Table 1 was given in full, how many rows would it have?
   - Since we know the truth value of a variable only has value of `1` or `0`, the combination of multiple variables is just 2^k where `k` is the number of variables. Here there are 6 variables so that $2^6 = 64$
2. Draw a schematic (not in Logisim) showing how you will connect the `mux2to1` modules to build the 4-to-1 multiplexer. Be prepared to explain it to the TA as part of your prelab. The schematic should reflect how you are going to create your Logisim circuit.
   - 

3. Build your circuit in Logisim
   - 

4. Test your circuit with different values of s, u, v, w and x. Do enough testing to convince yourself that the circuit is working. You must show these to the TA as part of your prelab.

- write down full 64 combination tests and pass the test,

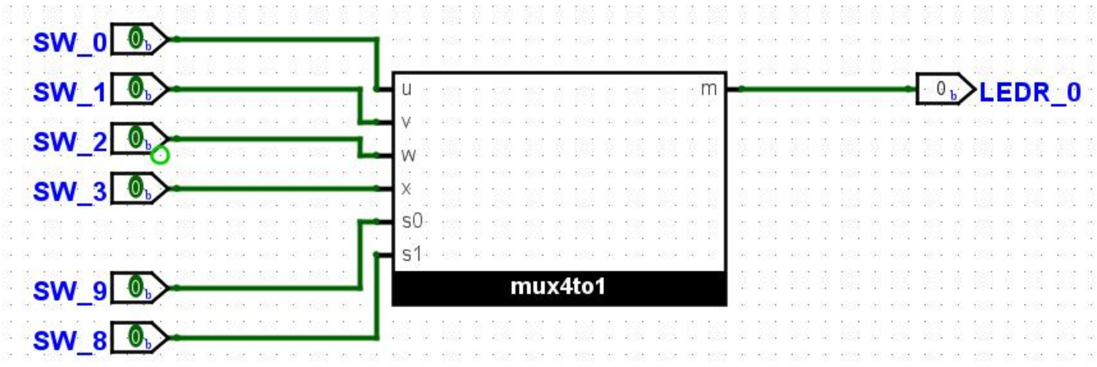


Logisim: Test Vector mux4to1 of mux

Passed: 64 Failed: 0

status	u	v	w	x	s0	s1	m
pass	0	0	1	1	0	0	0
pass	0	0	1	1	0	1	1
pass	0	0	1	1	1	0	0
pass	0	0	1	1	1	1	1
pass	0	1	0	0	0	0	0
pass	0	1	0	0	0	1	0
pass	0	1	0	0	1	0	1
pass	0	1	0	0	1	1	0
pass	0	1	0	1	0	0	0
pass	0	1	0	1	0	1	0
pass	0	1	0	1	1	0	1
pass	0	1	0	1	1	1	1

5. Map your Logisim design to the **DE1-SoC** board inputs and outputs. Use switches SW_{9-8} as the 2-bits input, and switches SW_{0-3} as the data inputs (labeled as u,v,w,x in Figure 3). Connect the output m to $LEDR_0$

   - 

Part III

1. Write the expressions for seven Boolean functions, one for each segment of the 7-segment decoder. You must use Karnaugh maps for optimization. You should be able to explain to the TA how you generated these expressions by showing them the truth-tables you wrote, and Karnaugh maps you used to optimize your circuits.

- HEX_0 : The segment 0 display when 0,2,3,5,6,7,8,9,A,C,E,F

$c_3c_2c_1c_0$	**Character**	**Truth Value**
0000	0	1
0001	1	0
0010	2	1
0011	3	1
0100	4	0
0101	5	1
0110	6	1
0111	7	1
1000	8	1
1001	9	1
1010	A	1
1011	b	0
1100	C	1
1101	d	0
1110	E	1
1111	F	1

According to the truth table, we can get the K-map

	$c_1'c_0'$	$c_1'c_0$	c_1c_0	c_1c_0'
$c_3'c_2'$	1	0	1	1
$c_3'c_2$	0	1	1	1
c_3c_2	1	0	1	1
c_3c_2'	1	1	0	1

Then we can get function
$HEX_0 = c_2'c_0' + c_3c_0' + c_3'c_1 + c_2c_1 + c_3c_2'c_1' + c_3'c_2c_0 = c_0'(c_2' + c_3) + c_1(c_3' + c_2) + c_3c_2'c_1' + c_3'c_2c_0$

- HEX_1: The segment 1 display when 0,1,2,3,4,7,8,9,A,d

$c_3c_2c_1c_0$	**Character**	**Truth Value**
0000	0	1
0001	1	1
0010	2	1
0011	3	1
0100	4	1
0101	5	0
0110	6	0
0111	7	1
1000	8	1
1001	9	1
1010	A	1
1011	b	0
1100	C	0
1101	d	1
1110	E	0
1111	F	0

According to the truth table, we can get the K-map

	$c_1'c_0'$	$c_1'c_0$	c_1c_0	c_1c_0'
$c_3'c_2'$	1	1	1	1
$c_3'c_2$	1	0	1	0
c_3c_2	0	1	0	0
c_3c_2'	1	1	0	1

Then we can get function
$HEX_1 = c_2'c_0' + c_3'c_2' + c_3'c_1'c_0' + c_3'c_1c_0 + c_3c_1'c_0 = c_2'(c_0' + c_3') + c_3'(c_1'c_0' + c_1c_0) + c_3c_1'c_0$

- HEX_2: The segment 2 display when 0,1,3,4,5,6,7,8,9,A,b,d,

$c_3c_2c_1c_0$	**Character**	**Truth Value**
0000	0	1
0001	1	1
0010	2	0
0011	3	1
0100	4	1
0101	5	1
0110	6	1
0111	7	1
1000	8	1
1001	9	1
1010	A	1
1011	b	1
1100	C	0
1101	d	1
1110	E	0
1111	F	0

According to the truth table, we can get the K-map

	$c_1'c_0'$	$c_1'c_0$	c_1c_0	c_1c_0'
$c_3'c_2'$	1	1	1	0
$c_3'c_2$	1	1	1	1
c_3c_2	0	1	0	0
c_3c_2'	1	1	1	1

Then we can get function
$HEX_2 = c_3'c_2 + c_3c_2' + c_1'c_0 + c_3'c_2'c_1' + c_3'c_2'c_0 = c_3'c_2 + c_3c_2' + c_1'c_0 + c_3'c_2'(c_1' + c_0)$

- HEX_3: The segment 3 display when 0,2,3,5,6,8,9,b,C,d,E

$c_3c_2c_1c_0$	**Character**	**Truth Value**
0000	0	1
0001	1	0
0010	2	1
0011	3	1
0100	4	0
0101	5	1
0110	6	1
0111	7	0
1000	8	1
1001	9	1
1010	A	0
1011	b	1
1100	C	1
1101	d	1
1110	E	1
1111	F	0

According to the truth table, we can get the K-map

	$c_1'c_0'$	$c_1'c_0$	c_1c_0	c_1c_0'
$c_3'c_2'$	1	0	1	1
$c_3'c_2$	0	1	0	1
c_3c_2	1	1	0	1
c_3c_2'	1	1	1	0

Then we can get function
$HEX_3 = c_3c_1' + c_3'c_2'c_0' + c_3'c_2'c_1 + c_2c_1'c_0 + c_2c_1c_0' + c_3c_2'c_0 = c_3c_1' + c_3'c_2'(c_0' + c_1) + c_2(c_1'c_0 + c_1c_0') + c_3c_2'c_0$

- HEX_4: The segment 4 display when 0,2,6,8,A,b,C,d,E,F

$c_3c_2c_1c_0$	**Character**	**Truth Value**
0000	0	1
0001	1	0
0010	2	1
0011	3	0
0100	4	0
0101	5	0
0110	6	1
0111	7	0
1000	8	1
1001	9	0
1010	A	1
1011	b	1
1100	C	1
1101	d	1
1110	E	1
1111	F	1

According to the truth table, we can get the K-map

	$c_1'c_0'$	$c_1'c_0$	c_1c_0	c_1c_0'
$c_3'c_2'$	1	0	0	1
$c_3'c_2$	0	0	0	1
c_3c_2	1	1	1	1
c_3c_2'	1	0	1	1

Then we can get function $HEX_4 = c_3c_2 + c_1c_0' + c_2'c_0' + c_3c_1 = c_3(c_2 + c_1) + c_0'(c_1 + c_2')$

- HEX_5: The segment 5 display when 0,4,5,6,8,9,A,b,C,E,F

$c_3c_2c_1c_0$	**Character**	**Truth Value**
0000	0	1
0001	1	0
0010	2	0
0011	3	0
0100	4	1
0101	5	1
0110	6	1
0111	7	0
1000	8	1
1001	9	1
1010	A	1
1011	b	1
1100	C	1
1101	d	0
1110	E	1
1111	F	1

According to the truth table, we can get the K-map

	$c_1'c_0'$	$c_1'c_0$	c_1c_0	c_1c_0'
$c_3'c_2'$	1	0	0	0
$c_3'c_2$	1	1	0	1
c_3c_2	1	0	1	1
c_3c_2'	1	1	1	1

Then we can get function
$HEX_5 = c_1'c_0' + c_3c_2' + c_2c_0' + c_3c_1 + c_3'c_2c_1' = c_0'(c_1' + c_2) + c_3(c_2' + c_1) + c_3'c_2c_1'$

- HEX_6: The segment 6 display when 2,3,4,5,6,8,9,A,b,d,E,F

$c_3c_2c_1c_0$	**Character**	**Truth Value**
0000	0	0
0001	1	0
0010	2	1
0011	3	1
0100	4	1
0101	5	1
0110	6	1
0111	7	0
1000	8	1
1001	9	1
1010	A	1
1011	b	1
1100	C	0
1101	d	1
1110	E	1
1111	F	1

According to the truth table, we can get the K-map

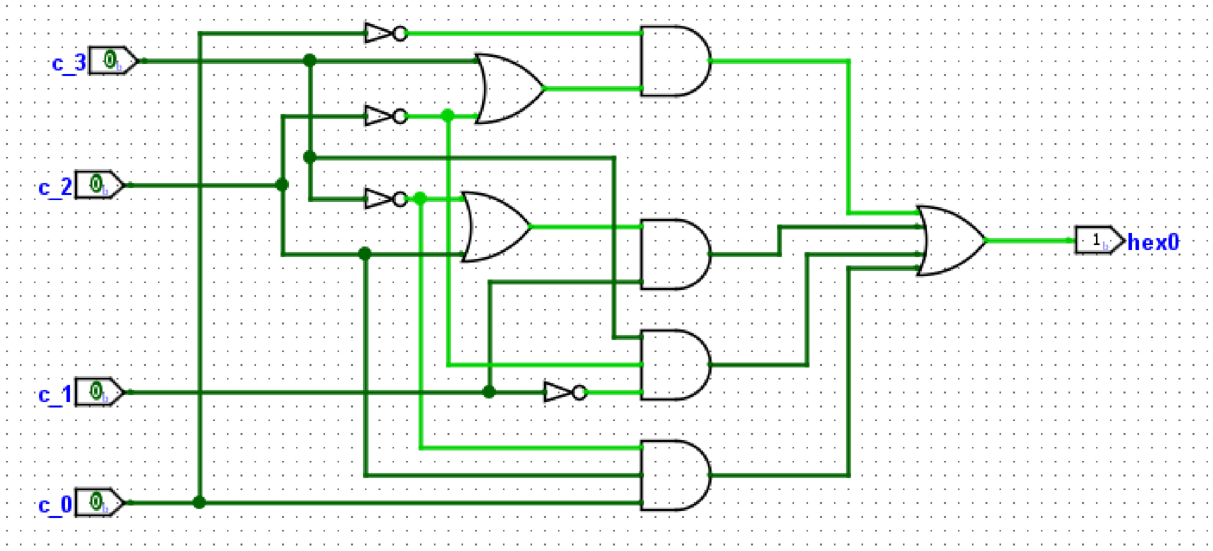
	$c_1'c_0'$	$c_1'c_0$	c_1c_0	c_1c_0'
$c_3'c_2'$	0	0	1	1
$c_3'c_2$	1	1	0	1
c_3c_2	0	1	1	1
c_3c_2'	1	1	1	1

Then we can get function
$HEX_6 = c_3c_2' + c_1c_0' + c_3c_0 + c_2'c_1 + c_3'c_2c_1' = c_3(c_2' + c_0) + c_1(c_0' + c_2') + c_3'c_2c_1'$

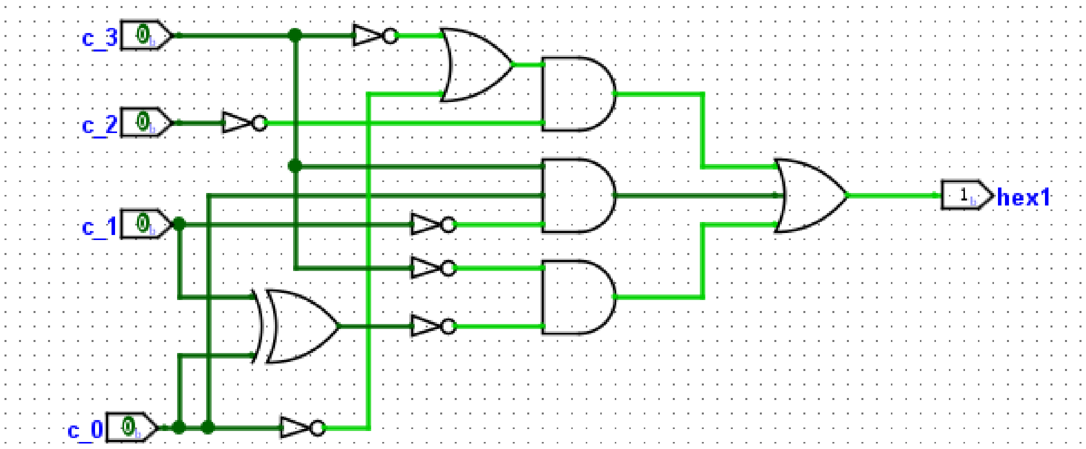
---

2. Build the circuit for the 7-segment decoder in Logisim taking advantage of the aforementioned expressions. HINT: you can change the number of inputs for each gate in by selecting it and go to Properties. Also it might be helpful to separate the circuits for each bit into different modules.

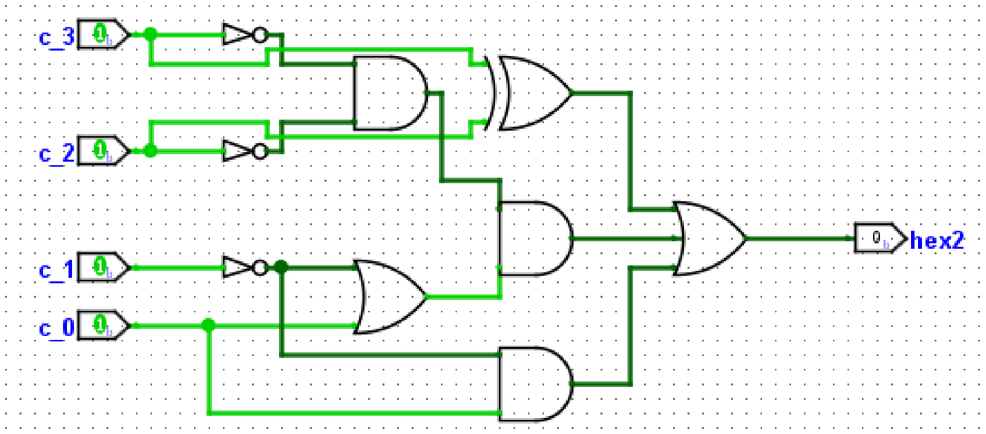
HEX0:



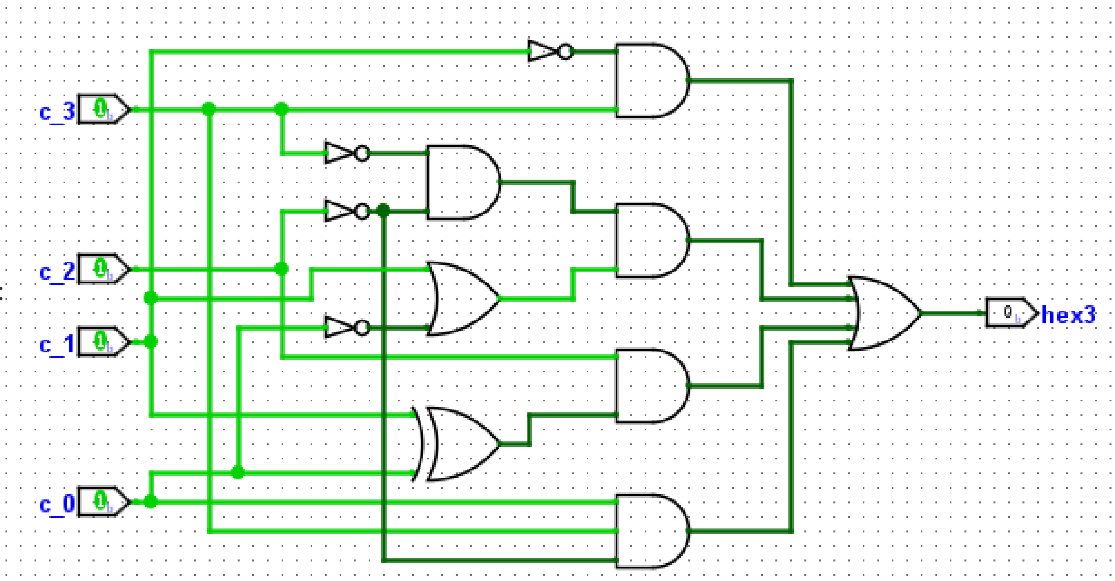
HEX1:



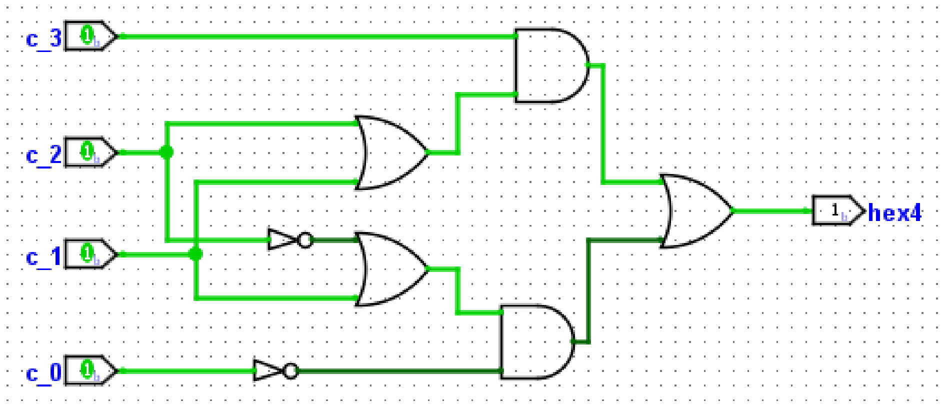
HEX2:



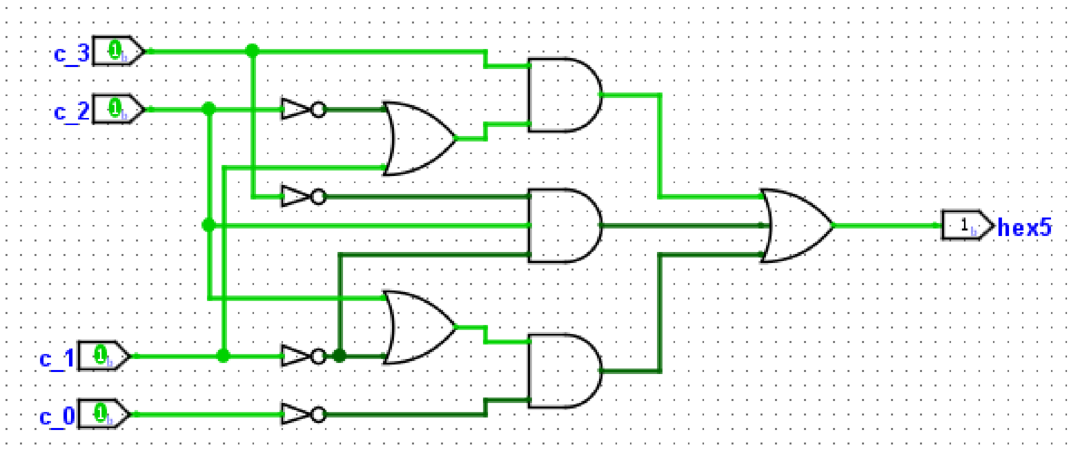
HEX3:



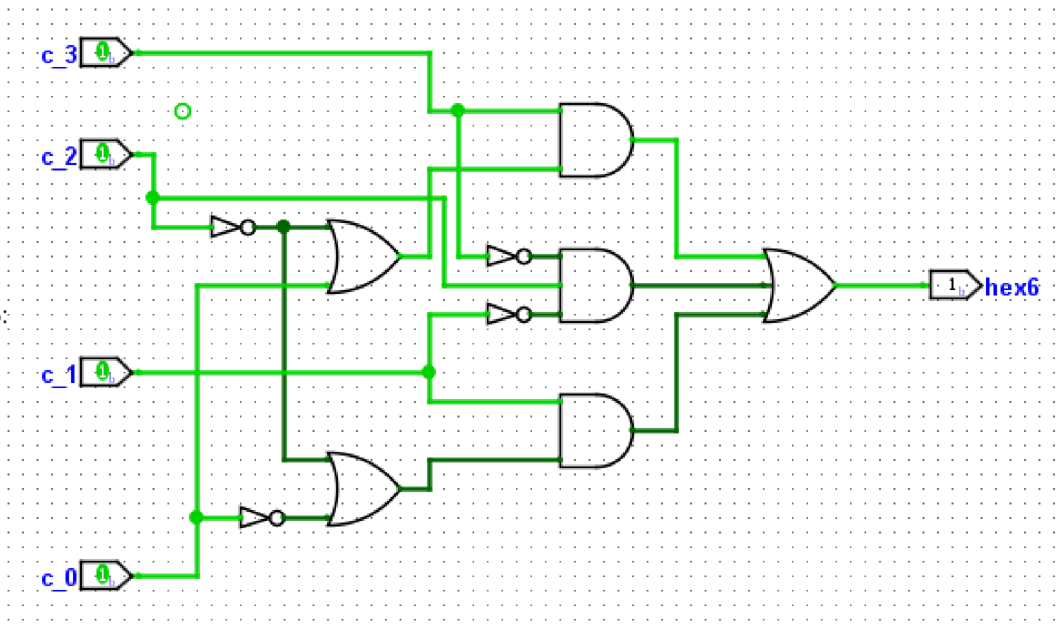
HEX4:



HEX5:

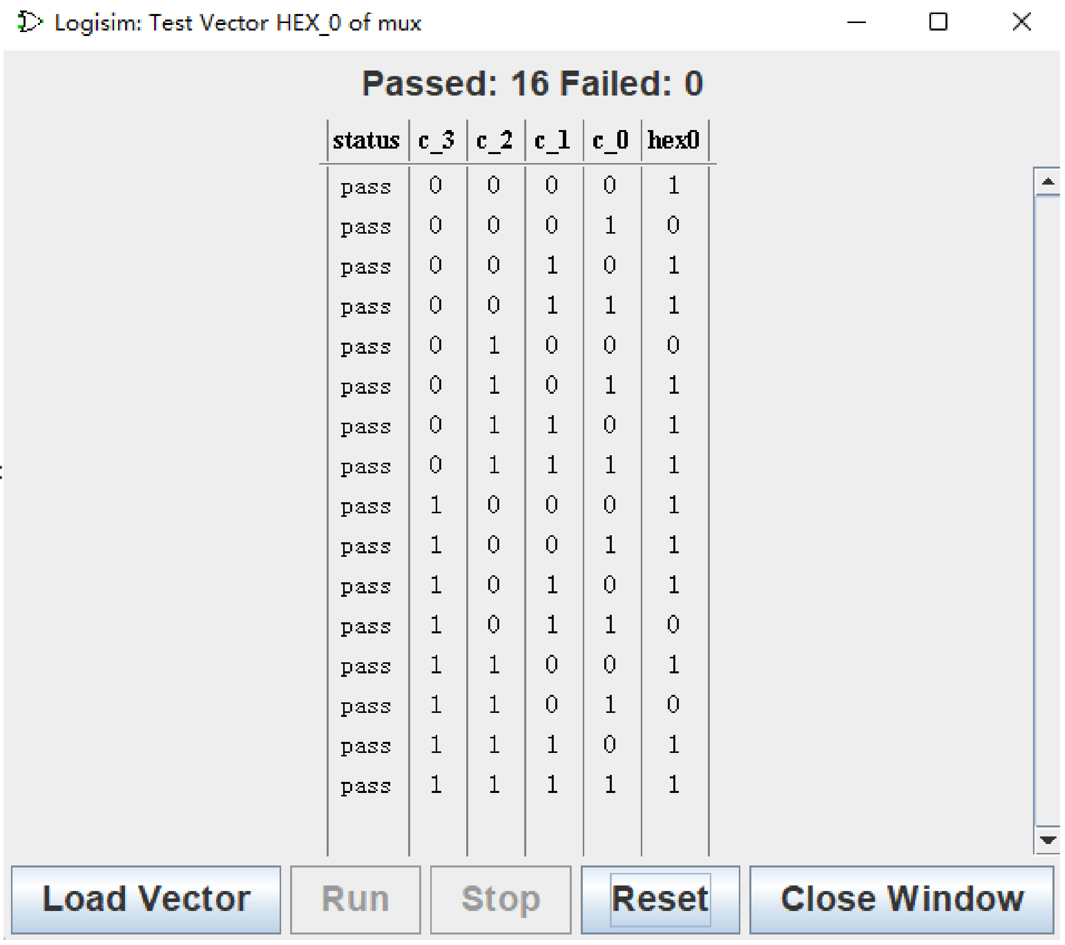


HEX6:



3. Check whether your implementation is correct. To verify the correctness of each segment, you need to create a test vector file for each segment's truth table. To test whether your circuits for each segment connect well with the 7-Segment Display in Logisim, use 'Poke' on the tool bar. Show the screenshots of your simulation to your TA as part of the prelab.

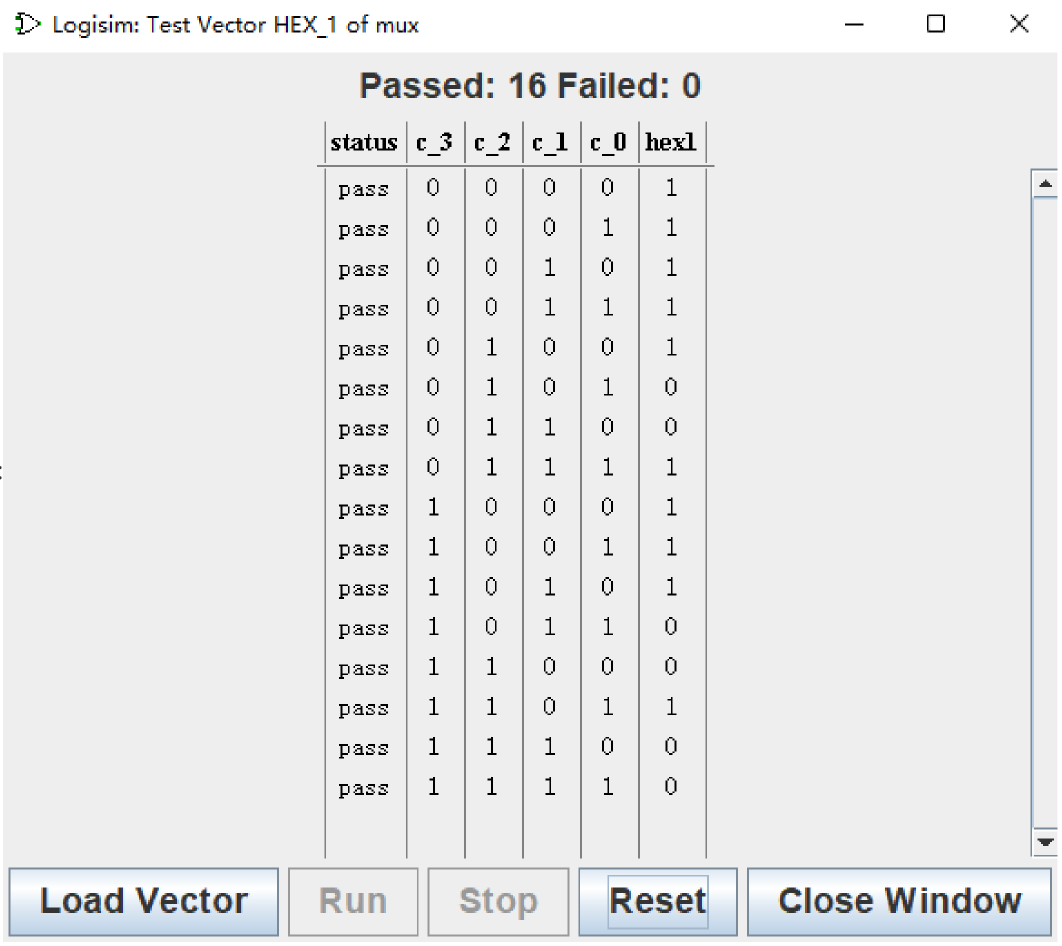
test for HEX0:



Passed: 16 Failed: 0

status	c_3	c_2	c_1	c_0	hex0
pass	0	0	0	0	1
pass	0	0	0	1	0
pass	0	0	1	0	1
pass	0	0	1	1	1
pass	0	1	0	0	0
pass	0	1	0	1	1
pass	0	1	1	0	1
pass	0	1	1	1	1
pass	1	0	0	0	1
pass	1	0	0	1	1
pass	1	0	1	0	1
pass	1	0	1	1	0
pass	1	1	0	0	1
pass	1	1	0	1	0
pass	1	1	1	0	1
pass	1	1	1	1	1

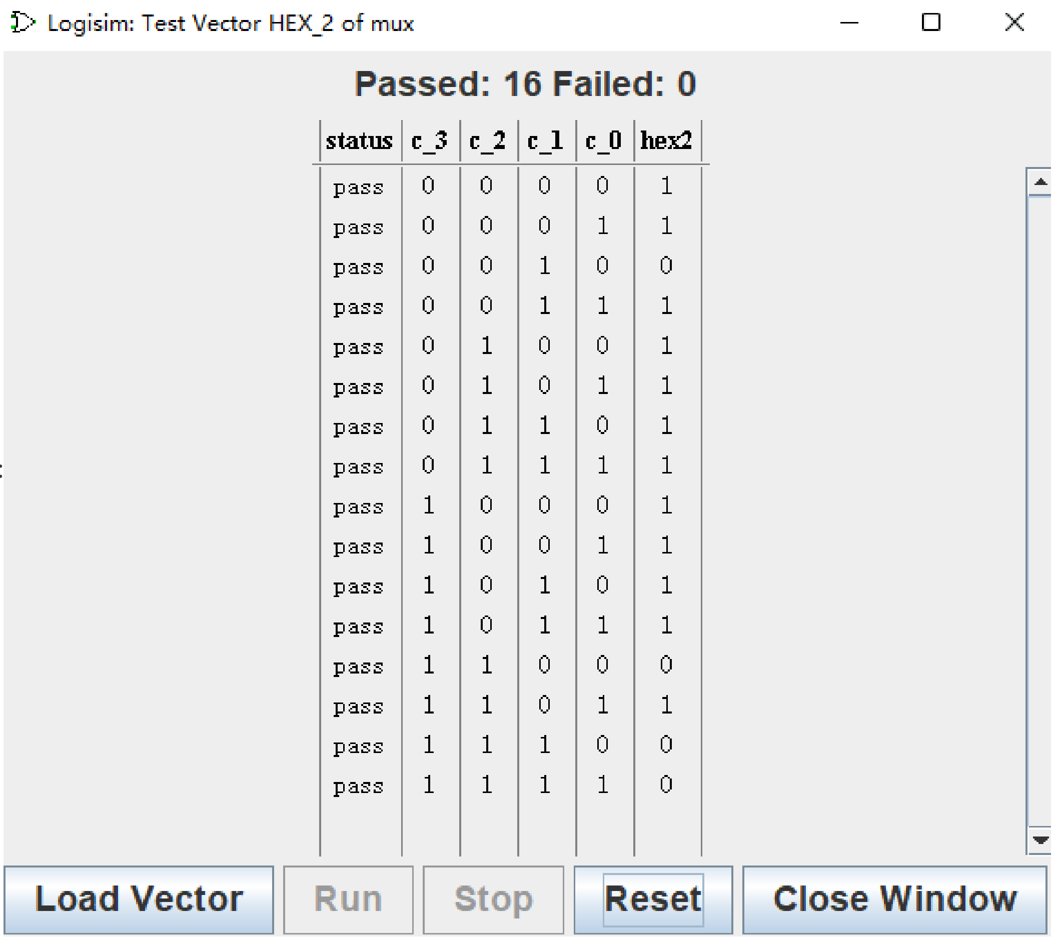
test for HEX1:



Passed: 16 Failed: 0

status	c_3	c_2	c_1	c_0	hex1
pass	0	0	0	0	1
pass	0	0	0	1	1
pass	0	0	1	0	1
pass	0	0	1	1	1
pass	0	1	0	0	1
pass	0	1	0	1	0
pass	0	1	1	0	0
pass	0	1	1	1	1
pass	1	0	0	0	1
pass	1	0	0	1	1
pass	1	0	1	0	1
pass	1	0	1	1	0
pass	1	1	0	0	0
pass	1	1	0	1	1
pass	1	1	1	0	0
pass	1	1	1	1	0

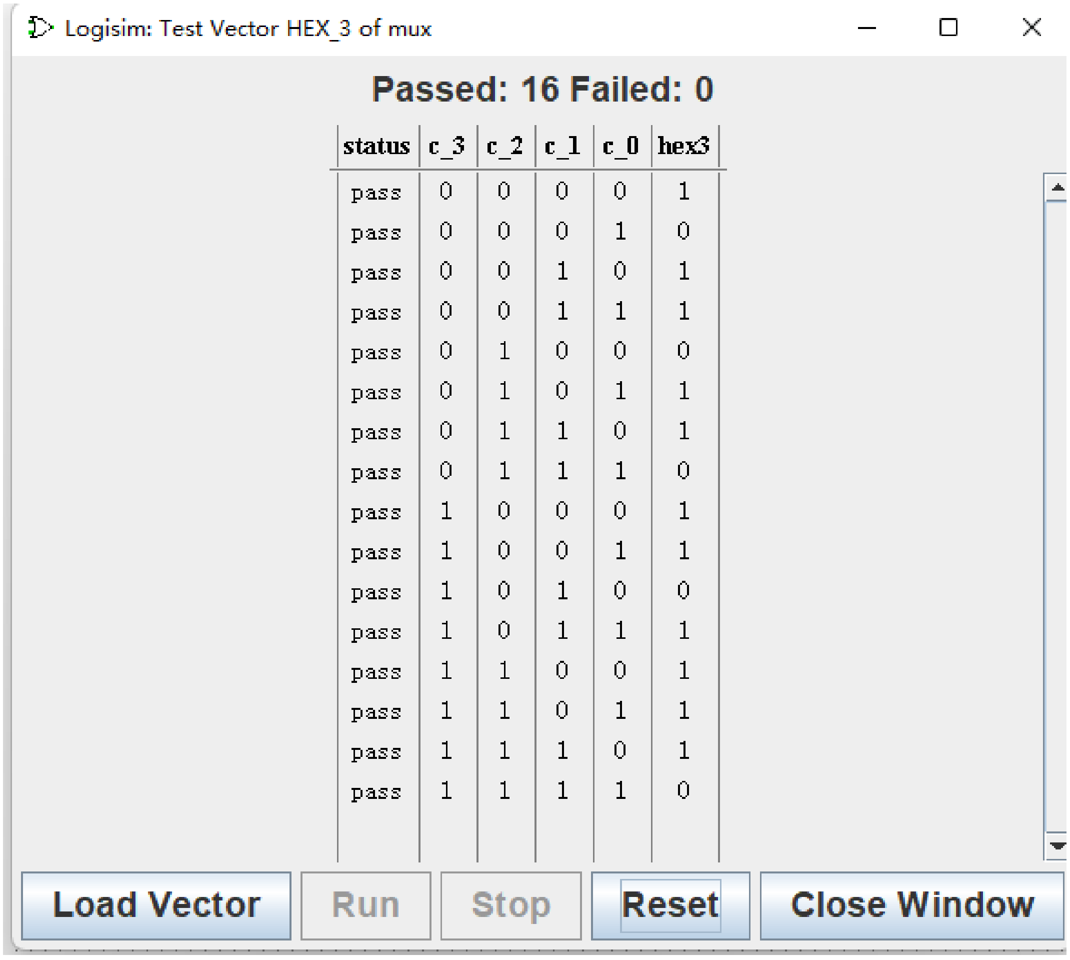
test for HEX2:



Passed: 16 Failed: 0

status	c_3	c_2	c_1	c_0	hex2
pass	0	0	0	0	1
pass	0	0	0	1	1
pass	0	0	1	0	0
pass	0	0	1	1	1
pass	0	1	0	0	1
pass	0	1	0	1	1
pass	0	1	1	0	1
pass	0	1	1	1	1
pass	1	0	0	0	1
pass	1	0	0	1	1
pass	1	0	1	0	1
pass	1	0	1	1	1
pass	1	1	0	0	0
pass	1	1	0	1	1
pass	1	1	1	0	0
pass	1	1	1	1	0

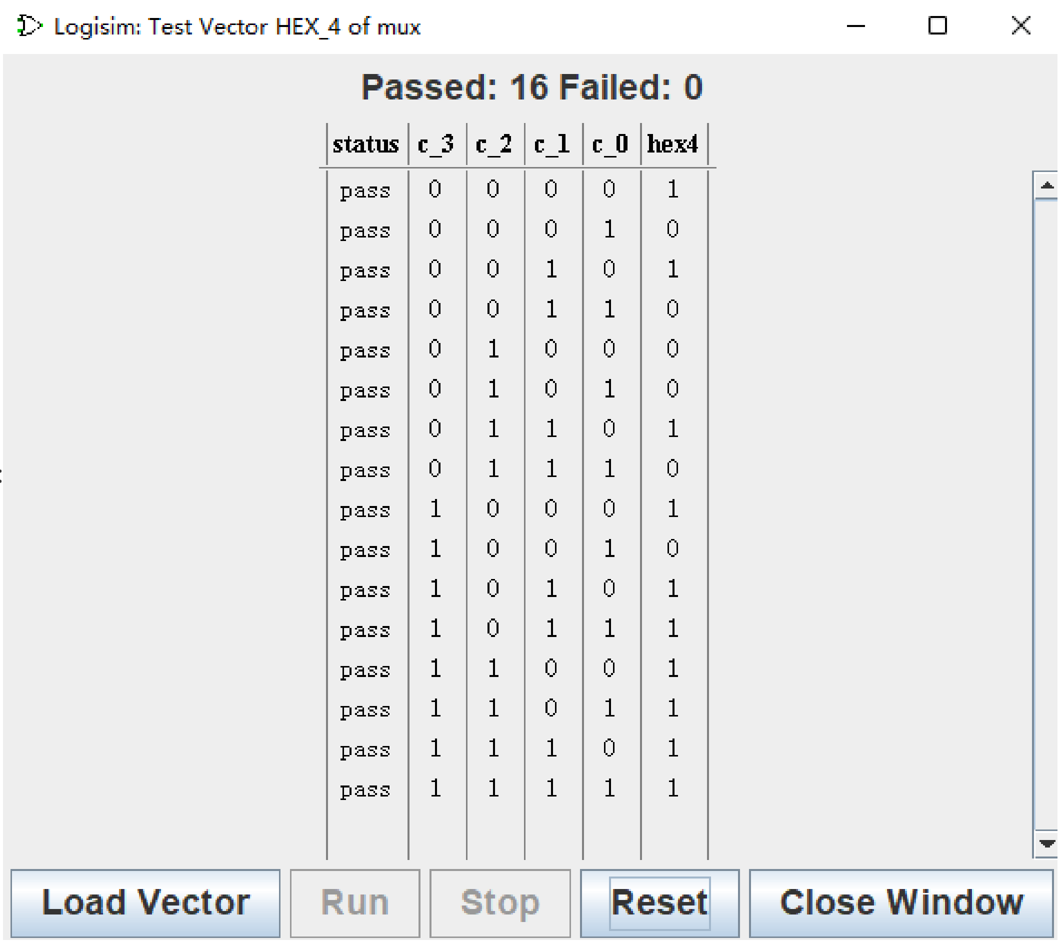
test for HEX3:



Logisim: Test Vector HEX_3 of mux

Passed: 16 Failed: 0

status	c_3	c_2	c_1	c_0	hex3
pass	0	0	0	0	1
pass	0	0	0	1	0
pass	0	0	1	0	1
pass	0	0	1	1	1
pass	0	1	0	0	0
pass	0	1	0	1	1
pass	0	1	1	0	1
pass	0	1	1	1	0
pass	1	0	0	0	1
pass	1	0	0	1	1
pass	1	0	1	0	0
pass	1	0	1	1	1
pass	1	1	0	0	1
pass	1	1	0	1	1
pass	1	1	1	0	1
pass	1	1	1	1	0



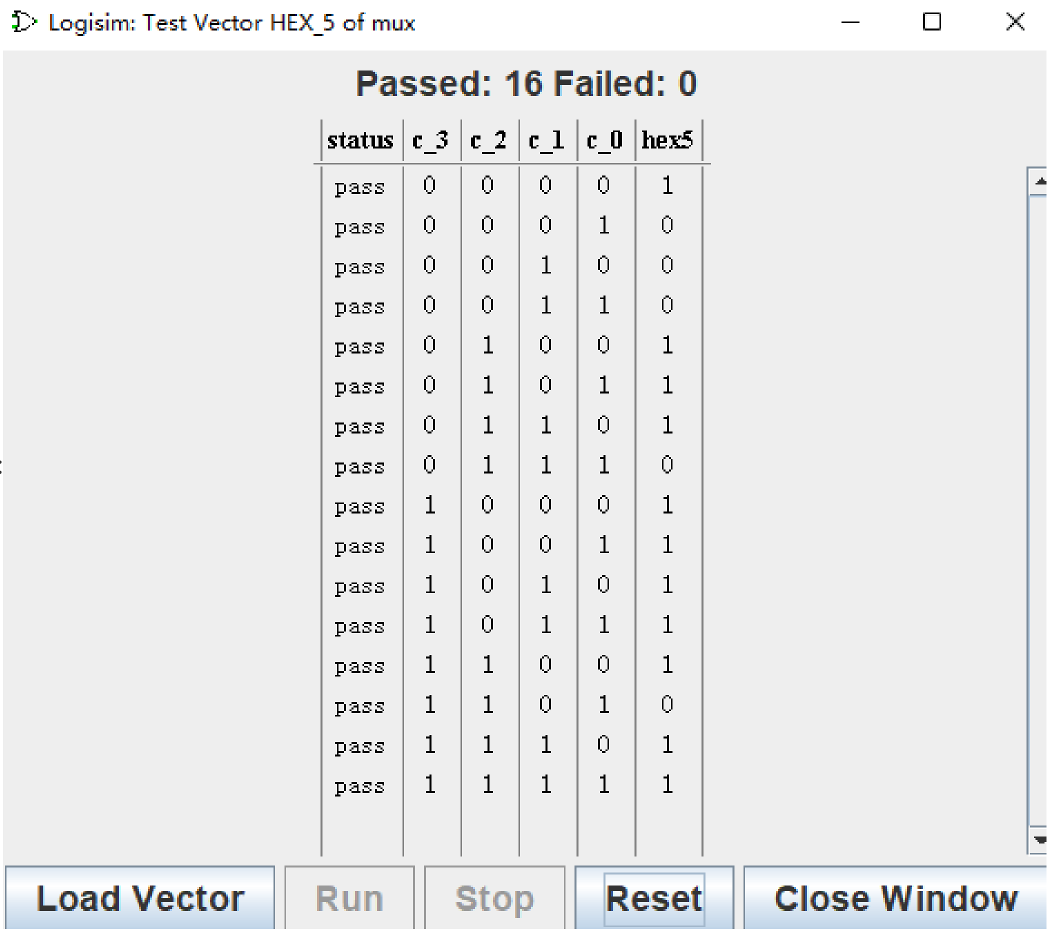
Logisim: Test Vector HEX_4 of mux

Passed: 16 Failed: 0

status	c_3	c_2	c_1	c_0	hex4
pass	0	0	0	0	1
pass	0	0	0	1	0
pass	0	0	1	0	1
pass	0	0	1	1	0
pass	0	1	0	0	0
pass	0	1	0	1	0
pass	0	1	1	0	1
pass	0	1	1	1	0
pass	1	0	0	0	1
pass	1	0	0	1	0
pass	1	0	1	0	1
pass	1	0	1	1	1
pass	1	1	0	0	1
pass	1	1	0	1	1
pass	1	1	1	0	1
pass	1	1	1	1	1

test for HEX4:

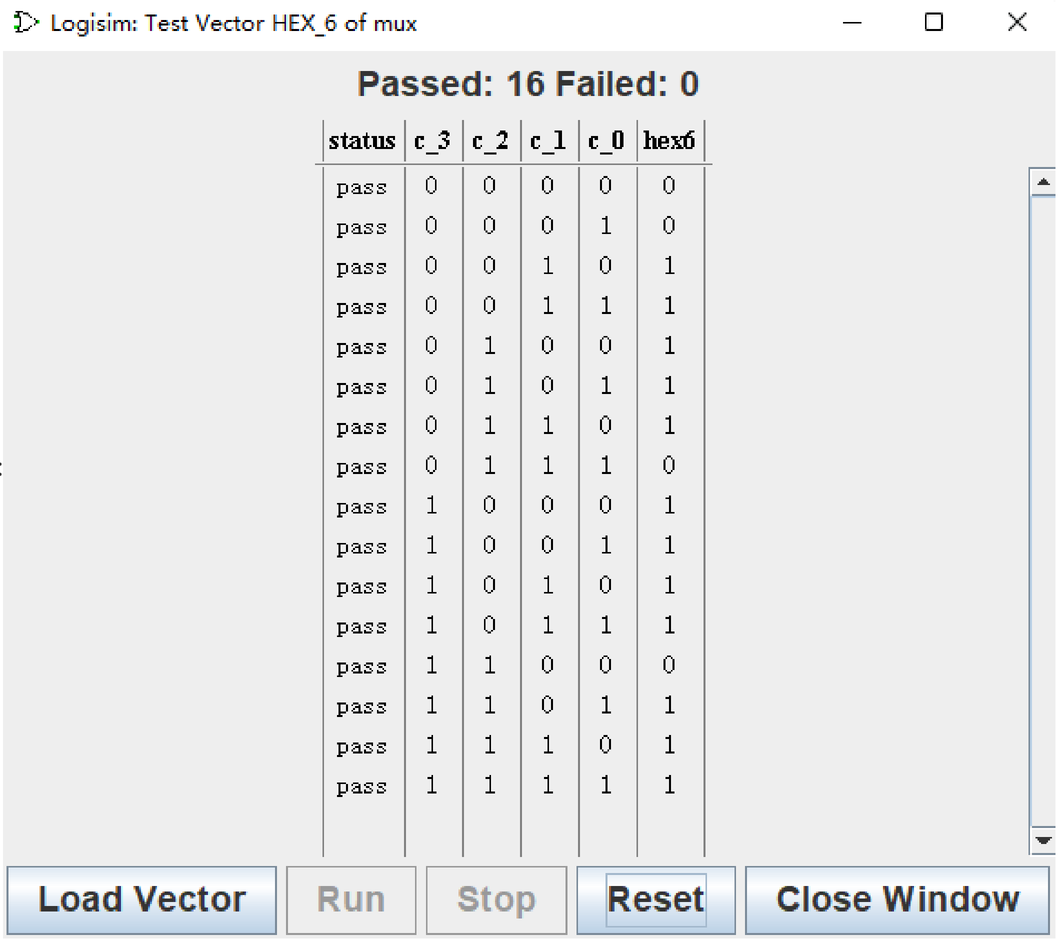
test for HEX5:



Passed: 16 Failed: 0

status	c_3	c_2	c_1	c_0	hex5
pass	0	0	0	0	1
pass	0	0	0	1	0
pass	0	0	1	0	0
pass	0	0	1	1	0
pass	0	1	0	0	1
pass	0	1	0	1	1
pass	0	1	1	0	1
pass	0	1	1	1	0
pass	1	0	0	0	1
pass	1	0	0	1	1
pass	1	0	1	0	1
pass	1	0	1	1	1
pass	1	1	0	0	1
pass	1	1	0	1	0
pass	1	1	1	0	1
pass	1	1	1	1	1

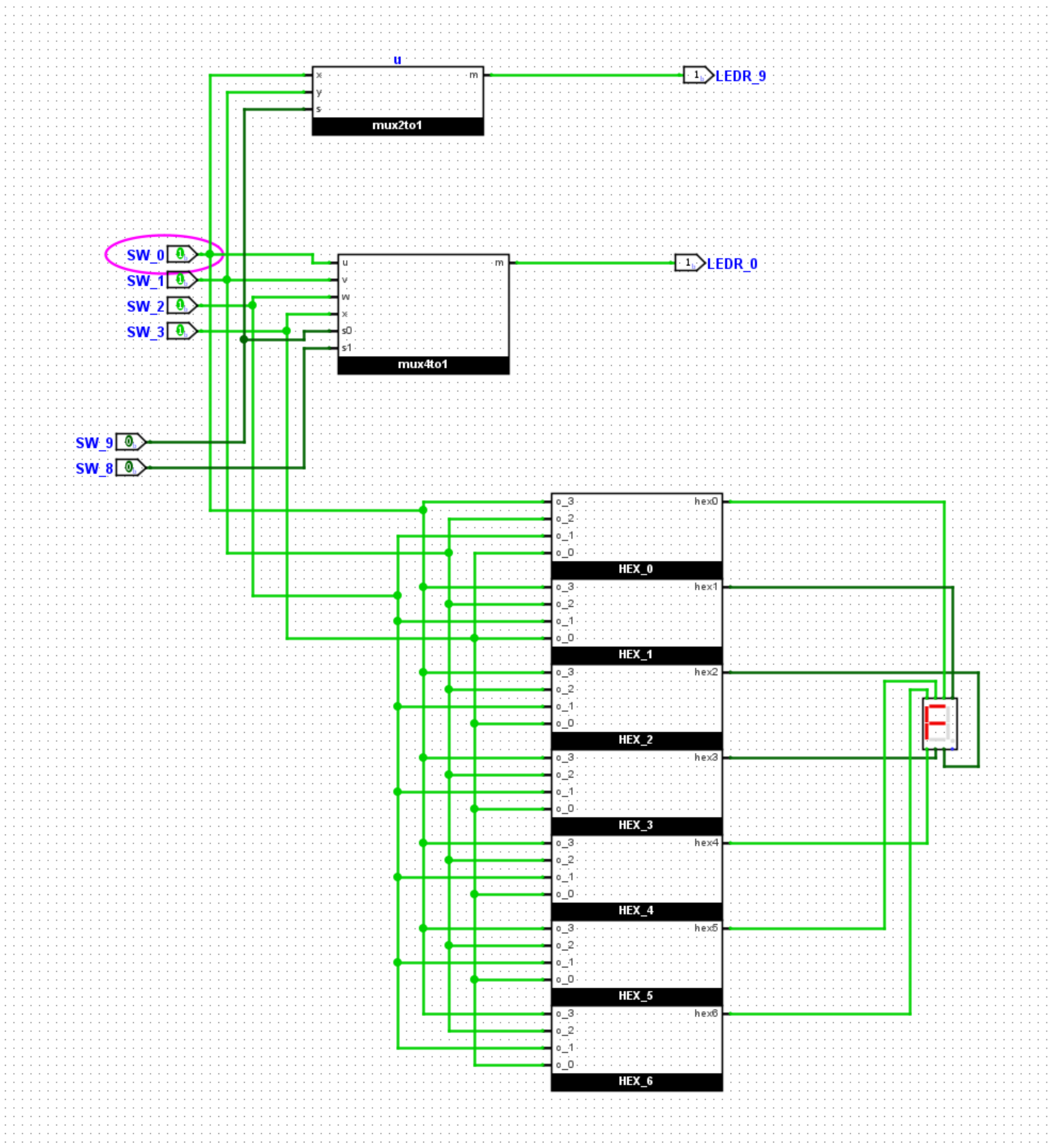
test for HEX6:



Passed: 16 Failed: 0

status	c_3	c_2	c_1	c_0	hex6
pass	0	0	0	0	0
pass	0	0	0	1	0
pass	0	0	1	0	1
pass	0	0	1	1	1
pass	0	1	0	0	1
pass	0	1	0	1	1
pass	0	1	1	0	1
pass	0	1	1	1	0
pass	1	0	0	0	1
pass	1	0	0	1	1
pass	1	0	1	0	1
pass	1	0	1	1	1
pass	1	1	0	0	0
pass	1	1	0	1	1
pass	1	1	1	0	1
pass	1	1	1	1	1

Final Result:


u
mux2to1
mux4to1
LEDR_9
LEDR_0
SW_0
SW_1
SW_2
SW_3
SW_9
SW_8
HEX_0
HEX_1
HEX_2
HEX_3
HEX_4
HEX_5
HEX_6