Course: CSC258F

Professor: Steve Engels

Experimenter: Yuhao Yang

Student ID: 1005808057

Lab2

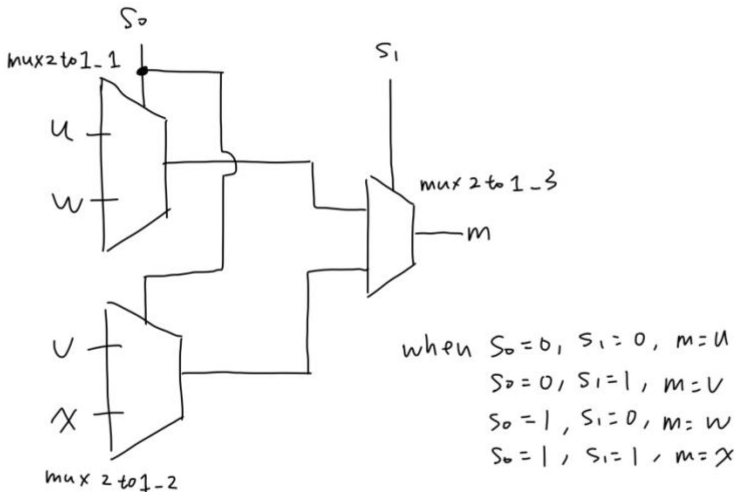
Pre-Lab Report

Part2

Q1: How many rows the truth table need if the truth table is given full?

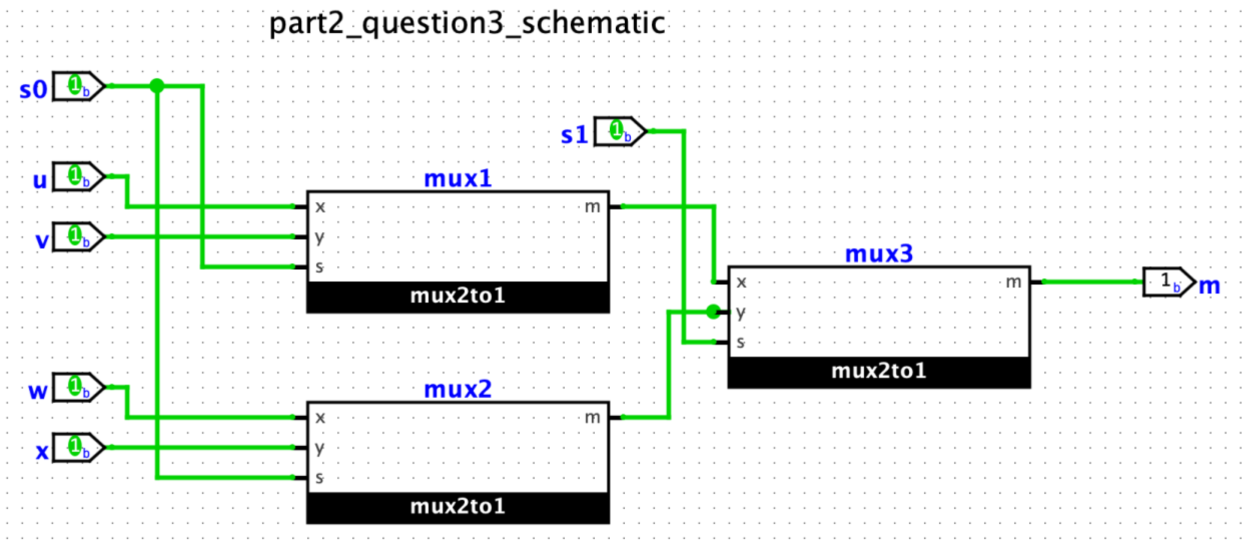
If there is truth table given in full, the truth table should have 2^6 = 64 rows in total.

2. Draw a schematic to show the 4-to-1 multiplexer by connecting 2-to-1 multiplexers.



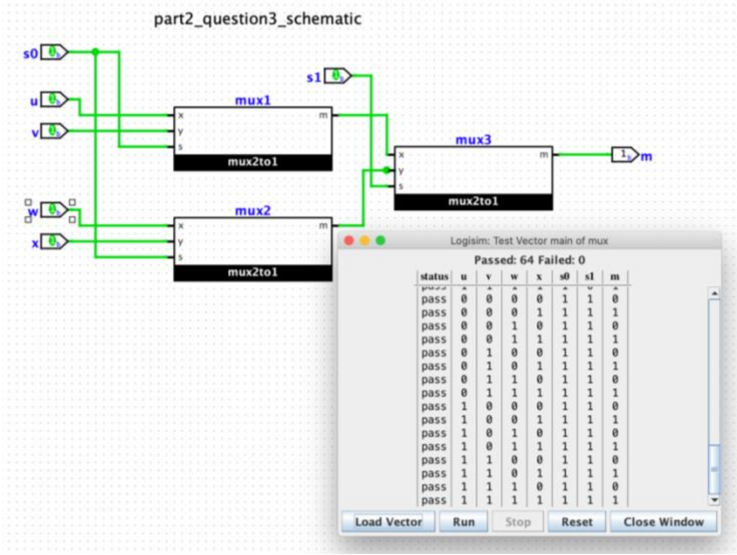
when $S_0=0$, $S_1=0$, $m=u$
$S_0=0$, $S_1=1$, $m=v$
$S_0=1$, $S_1=0$, $m=w$
$S_0=1$, $S_1=1$, $m=x$

3. Build the circuit from previous question in Logisim.



4. Do the test to the circuit to make sure everything is correct.

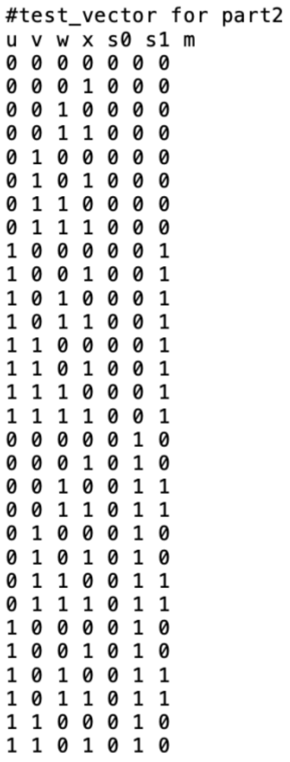
We can just simply list all possibilities in truth table and make a test vector to test the circuit.



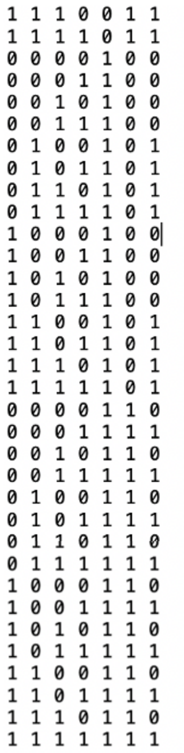
Logisim: Test Vector main of mux

Passed: 64 Failed: 0

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

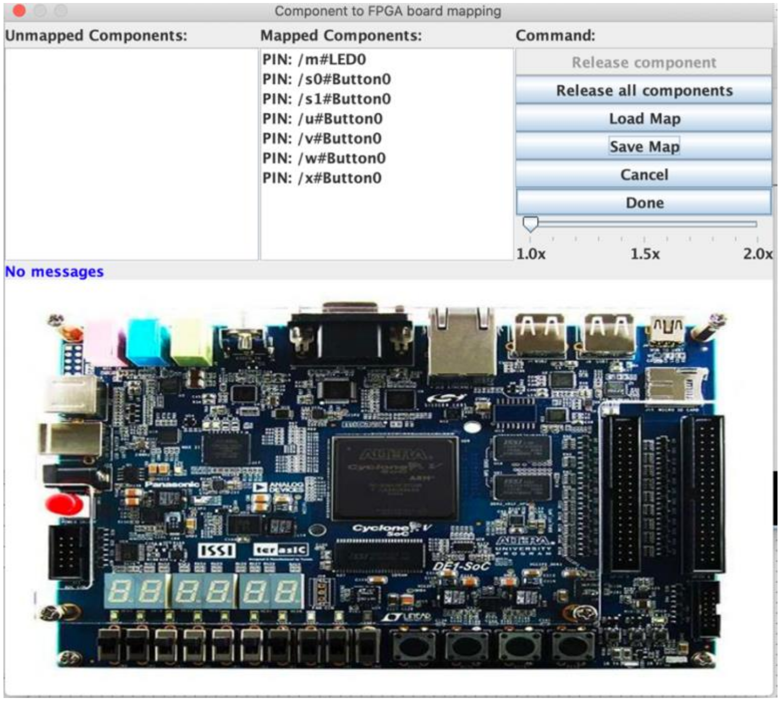


```
#test_vector for part2
u v w x s0 s1 m
0 0 0 0 0 0 0
0 0 0 1 0 0 0
0 0 1 0 0 0 0
0 0 1 1 0 0 0
0 1 0 0 0 0 0
0 1 0 1 0 0 0
0 1 1 0 0 0 0
0 1 1 1 0 0 0
1 0 0 0 0 0 1
1 0 0 1 0 0 1
1 0 1 0 0 0 1
1 0 1 1 0 0 1
1 1 0 0 0 0 1
1 1 0 1 0 0 1
1 1 1 0 0 0 1
1 1 1 1 0 0 1
0 0 0 0 0 1 0
0 0 0 1 0 1 0
0 0 1 0 0 1 1
0 0 1 1 0 1 1
0 1 0 0 0 1 0
0 1 0 1 0 1 0
0 1 1 0 0 1 1
0 1 1 1 0 1 1
1 0 0 0 0 1 0
1 0 0 1 0 1 0
1 0 1 0 0 1 1
1 0 1 1 0 1 1
1 1 0 0 0 1 0
1 1 0 1 0 1 0
```



```
1 1 1 0 0 1 1
1 1 1 1 0 1 1
0 0 0 0 1 0 0
0 0 0 1 1 0 0
0 0 1 0 1 0 0
0 0 1 1 1 0 0
0 1 0 0 1 0 1
0 1 0 1 1 0 1
0 1 1 0 1 0 1
0 1 1 1 1 0 1
1 0 0 0 1 0 0
1 0 0 1 1 0 0
1 0 1 0 1 0 0
1 0 1 1 1 0 0
1 1 0 0 1 0 1
1 1 0 1 1 0 1
1 1 1 0 1 0 1
1 1 1 1 1 0 1
0 0 0 0 1 1 0
0 0 0 1 1 1 1
0 0 1 0 1 1 0
0 0 1 1 1 1 1
0 1 0 0 1 1 0
0 1 0 1 1 1 1
0 1 1 0 1 1 0
0 1 1 1 1 1 1
1 0 0 0 1 1 0
1 0 0 1 1 1 1
1 0 1 0 1 1 0
1 0 1 1 1 1 1
1 1 0 0 1 1 0
1 1 0 1 1 1 1
1 1 1 0 1 1 0
1 1 1 1 1 1 1
```

5. Map the Logisim design to the DE1-SoC board inputs and outputs.



Component to FPGA board mapping

Unmapped Components:

Mapped Components:

PIN: /m#LED0
PIN: /s0#Button0
PIN: /s1#Button0
PIN: /u#Button0
PIN: /v#Button0
PIN: /w#Button0
PIN: /x#Button0

Command:

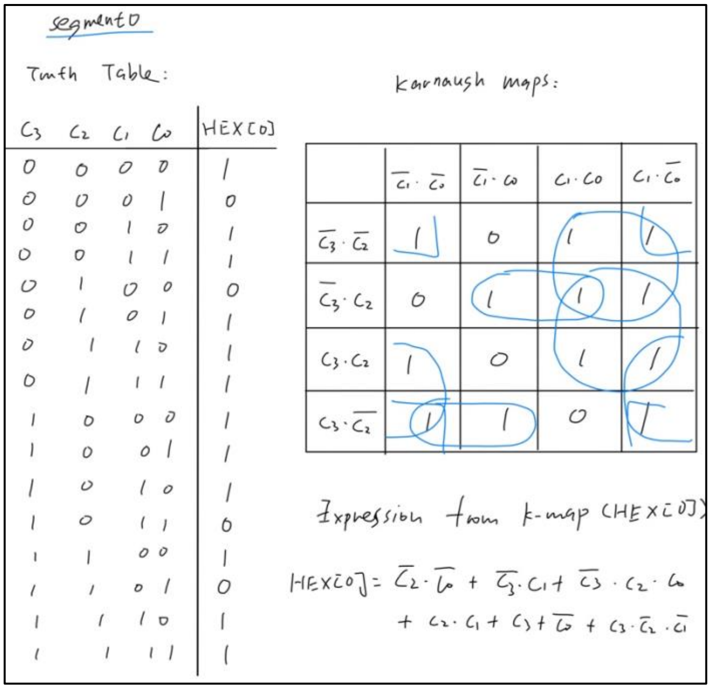
No messages

s0 mapped SW_9
s1 mapped SW_8
u mapped SW_0
v mapped SW_1
w mapped SW_2
x mapped SW_3
m apped $LEDR_0$

Part3

1. Write the expressions of each segment of the 7-segment decoder by optimizing the k-map of each segment.

HEX[0]:



segment0

Truth Table:

C_3	C_2	C_1	C_0	HEX[0]
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

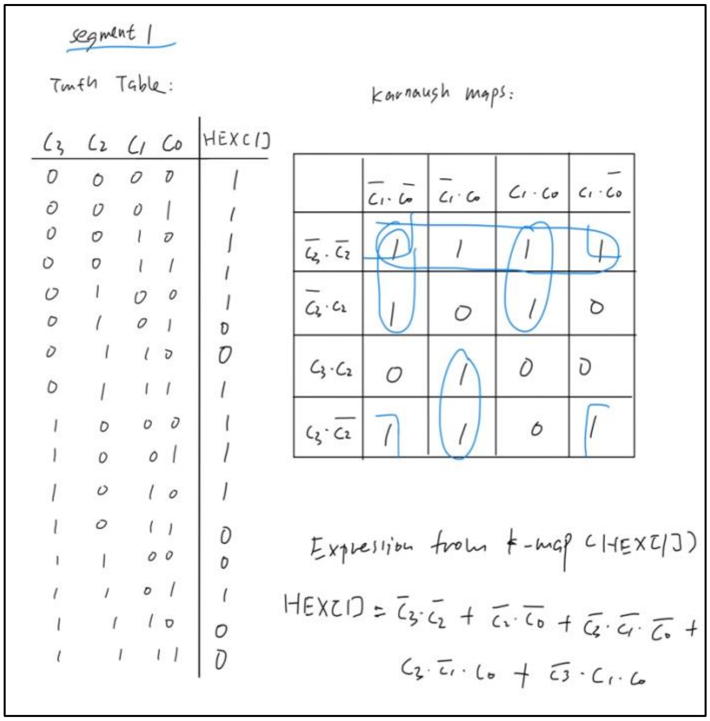
Karnaugh maps:

	$\overline{C_1}\cdot\overline{C_0}$	$\overline{C_1}\cdot C_0$	$C_1\cdot C_0$	$C_1\cdot\overline{C_0}$
$\overline{C_3}\cdot\overline{C_2}$	1	0	1	1
$\overline{C_3}\cdot C_2$	0	1	1	1
$C_3\cdot C_2$	1	0	1	1
$C_3\cdot\overline{C_2}$	1	1	0	1

Expression from k-map (HEX[0])

$$HEX[0] = \overline{C_2}\cdot\overline{C_0} + \overline{C_3}\cdot C_1 + \overline{C_3}\cdot C_2\cdot C_0 + C_2\cdot C_1 + C_3 + \overline{C_0} + C_3\cdot\overline{C_2}\cdot\overline{C_1}$$

HEX[1]:



segment 1

Truth Table:

C_3	C_2	C_1	C_0	HEX[1]
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

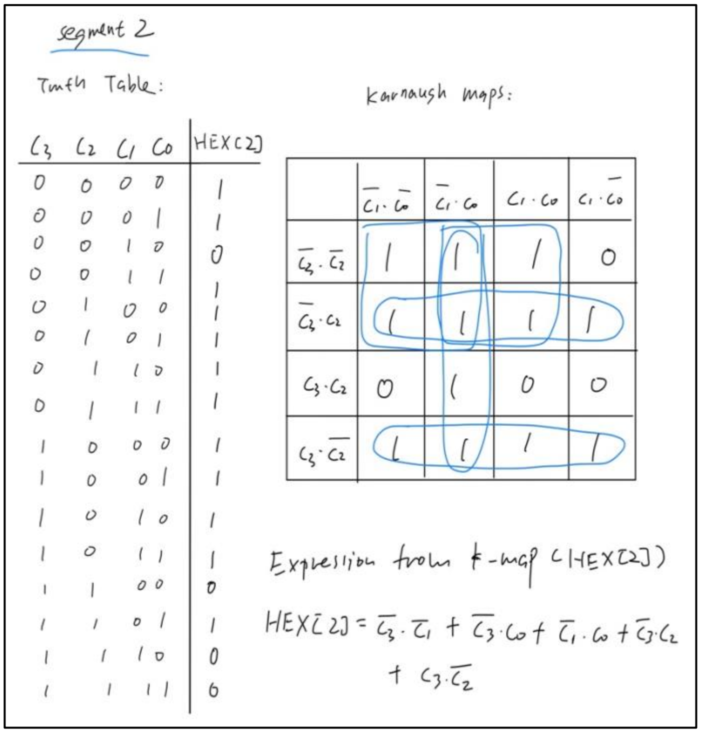
Karnaugh maps:

	$\overline{C_1}\cdot\overline{C_0}$	$\overline{C_1}\cdot C_0$	$C_1\cdot C_0$	$C_1\cdot\overline{C_0}$
$\overline{C_3}\cdot\overline{C_2}$	1	1	1	1
$\overline{C_3}\cdot C_2$	1	0	1	0
$C_3\cdot C_2$	0	1	0	0
$C_3\cdot\overline{C_2}$	1	1	0	1

Expression from k-map (HEX[1])

$$HEX[1] = \overline{C_3}\cdot\overline{C_2} + \overline{C_2}\cdot\overline{C_0} + \overline{C_3}\cdot\overline{C_1}\cdot\overline{C_0} + C_3\cdot\overline{C_1}\cdot C_0 + \overline{C_3}\cdot C_1\cdot C_0$$

HEX[2]:



segment 2

Truth Table:

C_3	C_2	C_1	C_0	HEX[2]
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

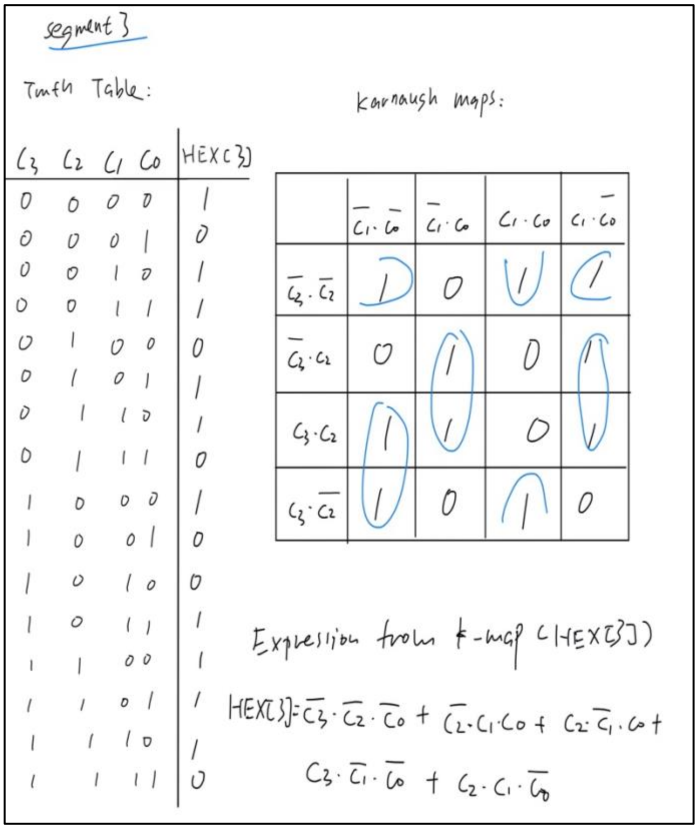
Karnaugh maps:

	$\overline{C_1}\cdot\overline{C_0}$	$\overline{C_1}\cdot C_0$	$C_1\cdot C_0$	$C_1\cdot\overline{C_0}$
$\overline{C_3}\cdot\overline{C_2}$	1	1	1	0
$\overline{C_3}\cdot C_2$	1	1	1	1
$C_3\cdot C_2$	0	1	0	0
$C_3\cdot\overline{C_2}$	1	1	1	1

Expression from K-map (HEX[2])

$$HEX[2] = \overline{C_3}\cdot\overline{C_1} + \overline{C_3}\cdot C_0 + \overline{C_1}\cdot C_0 + \overline{C_3}\cdot C_2 + C_3\cdot\overline{C_2}$$

HEX[3]:



segment 3

Truth Table:

C_3	C_2	C_1	C_0	HEX[3]
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

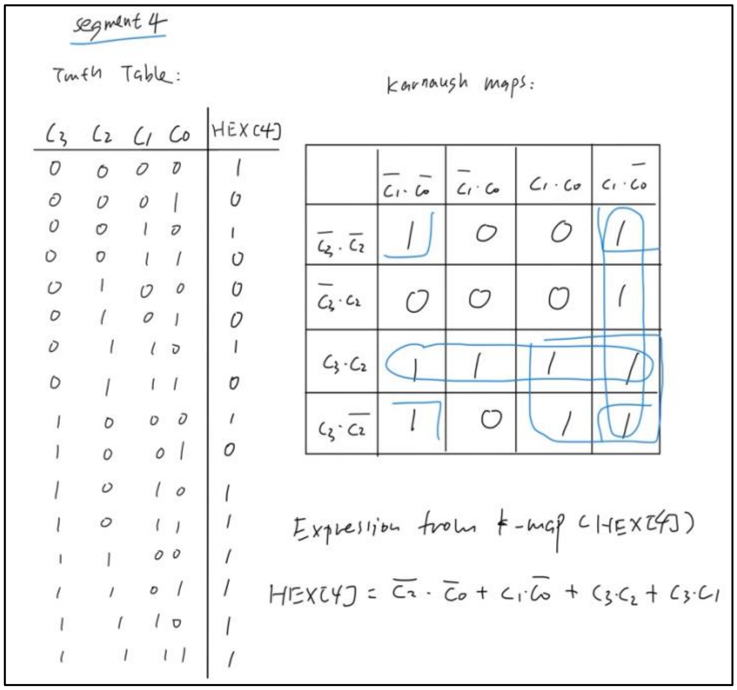
Karnaugh maps:

	$\overline{C_1}\cdot\overline{C_0}$	$\overline{C_1}\cdot C_0$	$C_1\cdot C_0$	$C_1\cdot\overline{C_0}$
$\overline{C_3}\cdot\overline{C_2}$	1	0	1	1
$\overline{C_3}\cdot C_2$	0	1	0	1
$C_3\cdot C_2$	1	1	0	1
$C_3\cdot\overline{C_2}$	1	0	1	0

Expression from K-map (HEX[3])

$$HEX[3] = \overline{C_3}\cdot\overline{C_2}\cdot\overline{C_0} + \overline{C_2}\cdot C_1\cdot C_0 + C_2\cdot\overline{C_1}\cdot C_0 + C_3\cdot\overline{C_1}\cdot\overline{C_0} + C_2\cdot C_1\cdot\overline{C_0}$$

HEX[4]:



segment 4

Truth Table:

C_3	C_2	C_1	C_0	HEX[4]
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

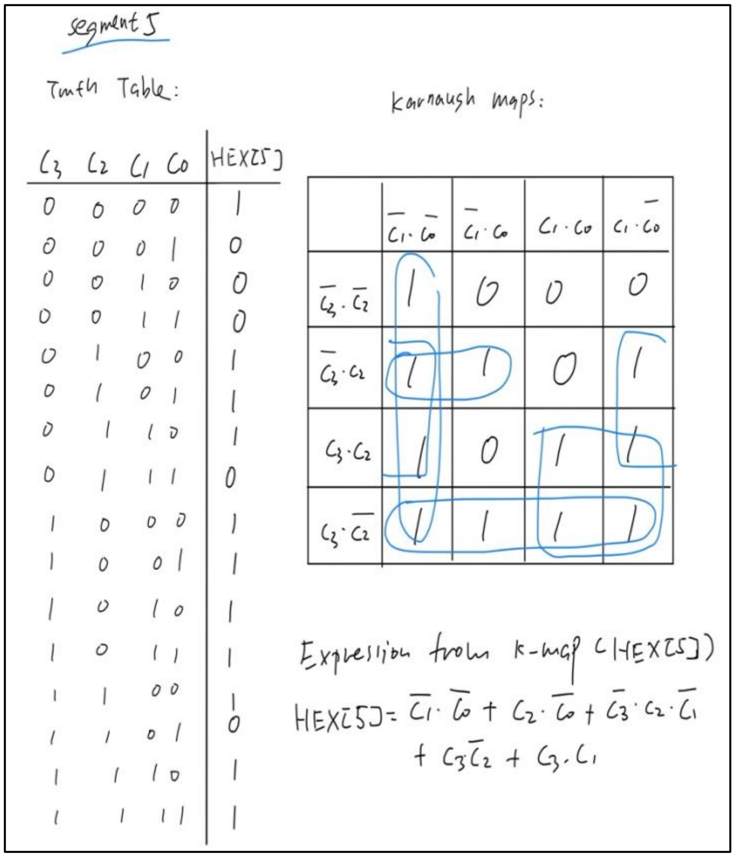
karnaugh maps:

	$\overline{C_1}\cdot\overline{C_0}$	$\overline{C_1}\cdot C_0$	$C_1\cdot C_0$	$C_1\cdot\overline{C_0}$
$\overline{C_3}\cdot\overline{C_2}$	1	0	0	1
$\overline{C_3}\cdot C_2$	0	0	0	1
$C_3\cdot C_2$	1	1	1	1
$C_3\cdot\overline{C_2}$	1	0	1	1

Expression from k-map (HEX[4])

$$HEX[4] = \overline{C_2}\cdot\overline{C_0} + C_1\cdot\overline{C_0} + C_3\cdot C_2 + C_3\cdot C_1$$

HEX[5]:



segment 5

Truth Table:

C_3	C_2	C_1	C_0	HEX[5]
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

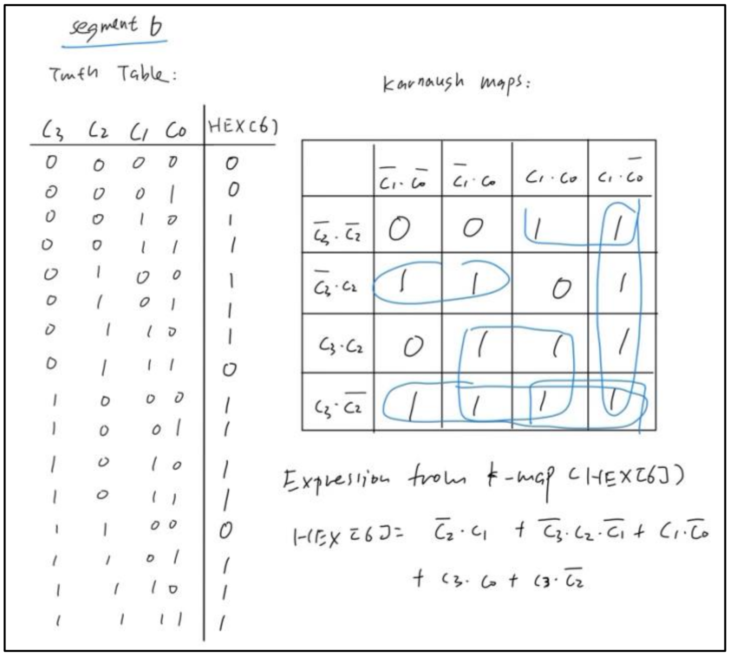
karnaugh maps:

	$\overline{C_1}\cdot\overline{C_0}$	$\overline{C_1}\cdot C_0$	$C_1\cdot C_0$	$C_1\cdot\overline{C_0}$
$\overline{C_3}\cdot\overline{C_2}$	1	0	0	0
$\overline{C_3}\cdot C_2$	1	1	0	1
$C_3\cdot C_2$	1	0	1	1
$C_3\cdot\overline{C_2}$	1	1	1	1

Expression from k-map (HEX[5])

$$HEX[5] = \overline{C_1}\cdot\overline{C_0} + C_2\cdot\overline{C_0} + \overline{C_3}\cdot C_2\cdot\overline{C_1}$$
$$+ C_3\overline{C_2} + C_3\cdot C_1$$

HEX[6]:



segment 6

Truth Table:

C_3	C_2	C_1	C_0	HEX[6]
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Karnaugh maps:

	$\overline{C_1}\cdot\overline{C_0}$	$\overline{C_1}\cdot C_0$	$C_1\cdot C_0$	$C_1\cdot\overline{C_0}$
$\overline{C_3}\cdot\overline{C_2}$	0	0	1	1
$\overline{C_3}\cdot C_2$	1	1	0	1
$C_3\cdot C_2$	0	1	1	1
$C_3\cdot\overline{C_2}$	1	1	1	1

Expression from K-map (HEX[6])

$$HEX[6] = \overline{C_2}\cdot C_1 + \overline{C_3}\cdot C_2\cdot\overline{C_1} + C_1\cdot\overline{C_0} + C_3\cdot C_0 + C_3\cdot\overline{C_2}$$

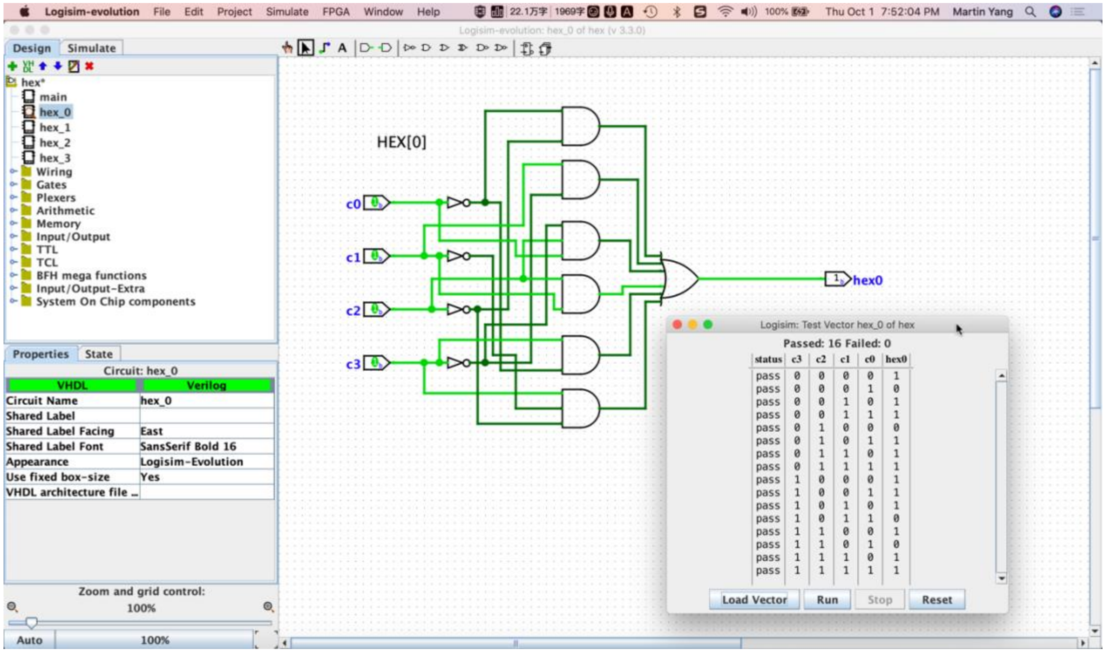
The six pictures above are all staff for question1.

2. Build the circuit for the 7-segment decoder in Logisim taking advantage of the aforementioned expressions.

All circuit are in file part3_circuit.circ. Please check it in file.

3. Show the correctness of 7-segment implementation.

HEX[0]



Circuit: hex_0

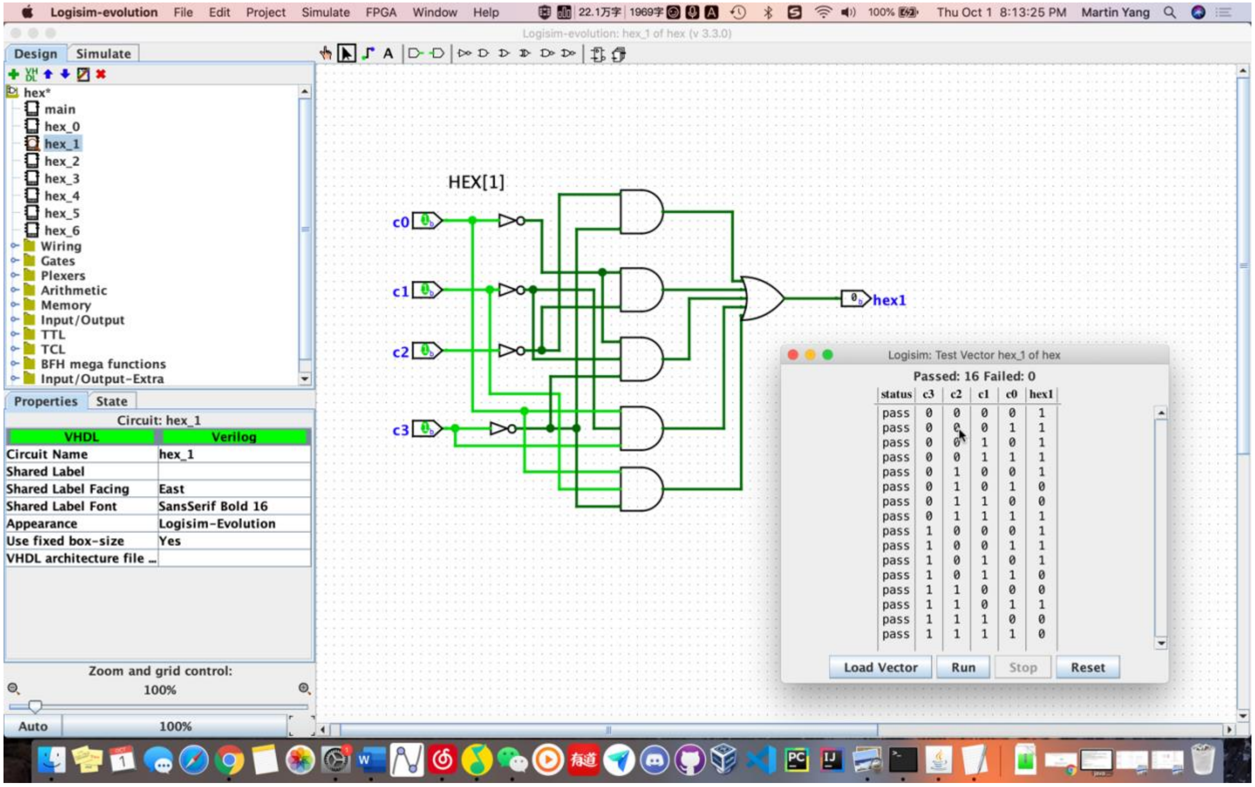
VHDL	Verilog
Circuit Name	hex_0
Shared Label	
Shared Label Facing	East
Shared Label Font	SansSerif Bold 16
Appearance	Logisim-Evolution
Use fixed box-size	Yes
VHDL architecture file ...	

Logisim: Test Vector hex_0 of hex

Passed: 16 Failed: 0

status	c3	c2	c1	c0	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

HEX[1]



Circuit: hex_1

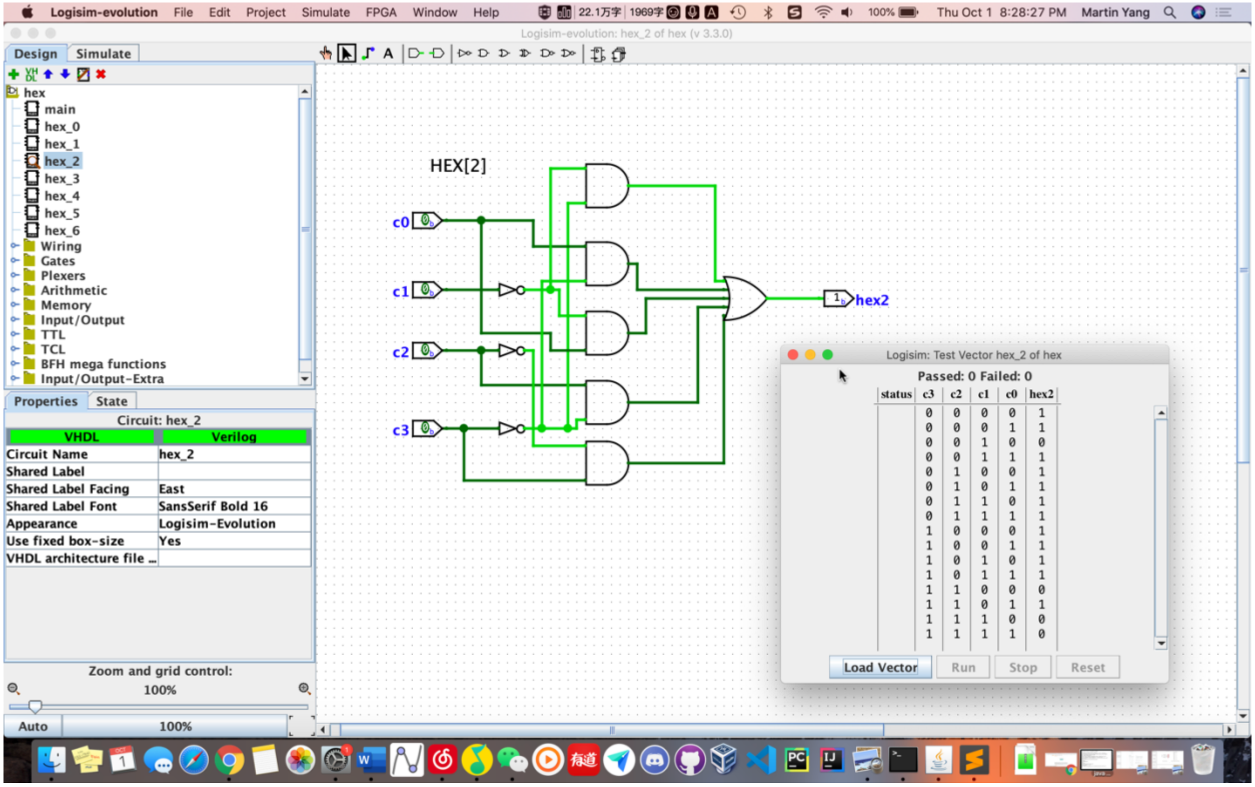
VHDL	Verilog
Circuit Name	hex_1
Shared Label	
Shared Label Facing	East
Shared Label Font	SansSerif Bold 16
Appearance	Logisim-Evolution
Use fixed box-size	Yes
VHDL architecture file ...	

Logisim: Test Vector hex_1 of hex

Passed: 16 Failed: 0

status	c3	c2	c1	c0	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

HEX[2]



Circuit: hex_2

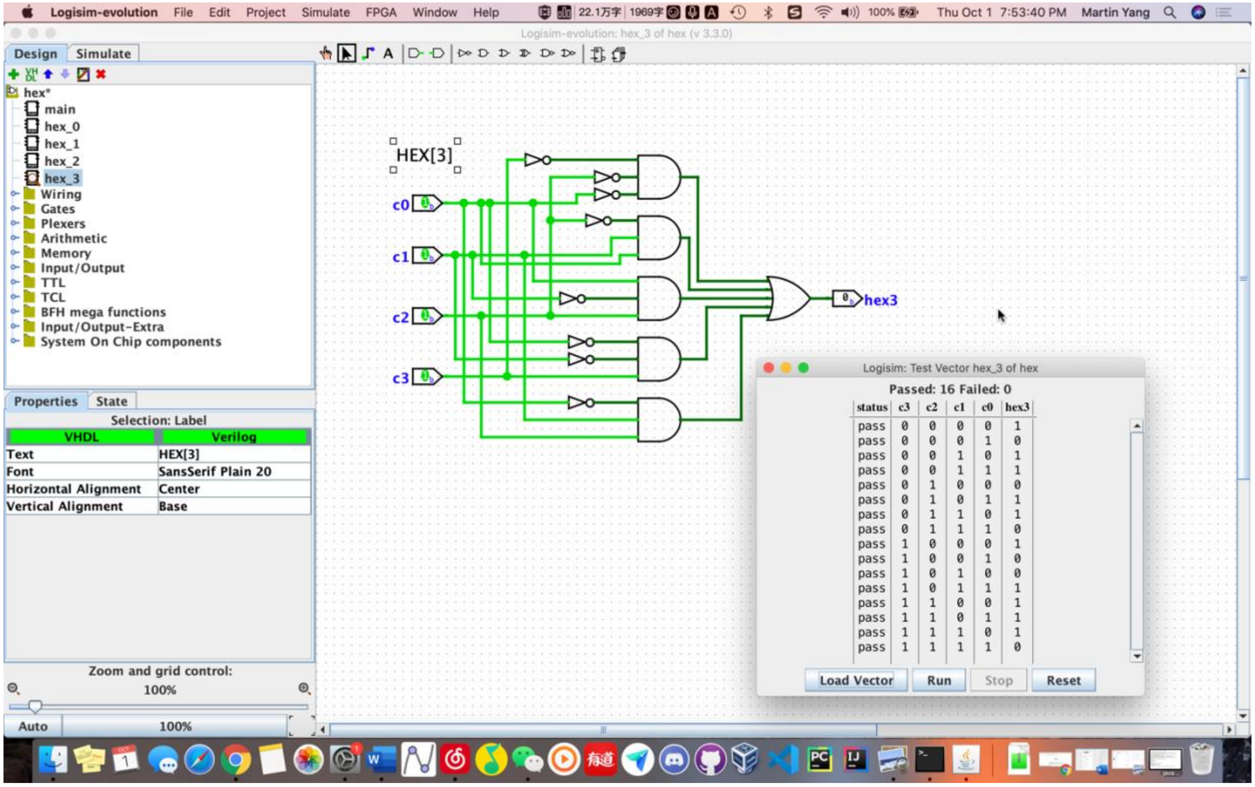
VHDL	Verilog
Circuit Name	hex_2
Shared Label	
Shared Label Facing	East
Shared Label Font	SansSerif Bold 16
Appearance	Logisim-Evolution
Use fixed box-size	Yes
VHDL architecture file ...	

Logisim: Test Vector hex_2 of hex

Passed: 0 Failed: 0

status	c3	c2	c1	c0	hex2
	0	0	0	0	1
	0	0	0	1	1
	0	0	1	0	0
	0	0	1	1	1
	0	1	0	0	1
	0	1	0	1	1
	0	1	1	0	1
	0	1	1	1	1
	1	0	0	0	1
	1	0	0	1	1
	1	0	1	0	1
	1	0	1	1	1
	1	1	0	0	0
	1	1	0	1	1
	1	1	1	0	0
	1	1	1	1	0

HEX[3]



Selection: Label

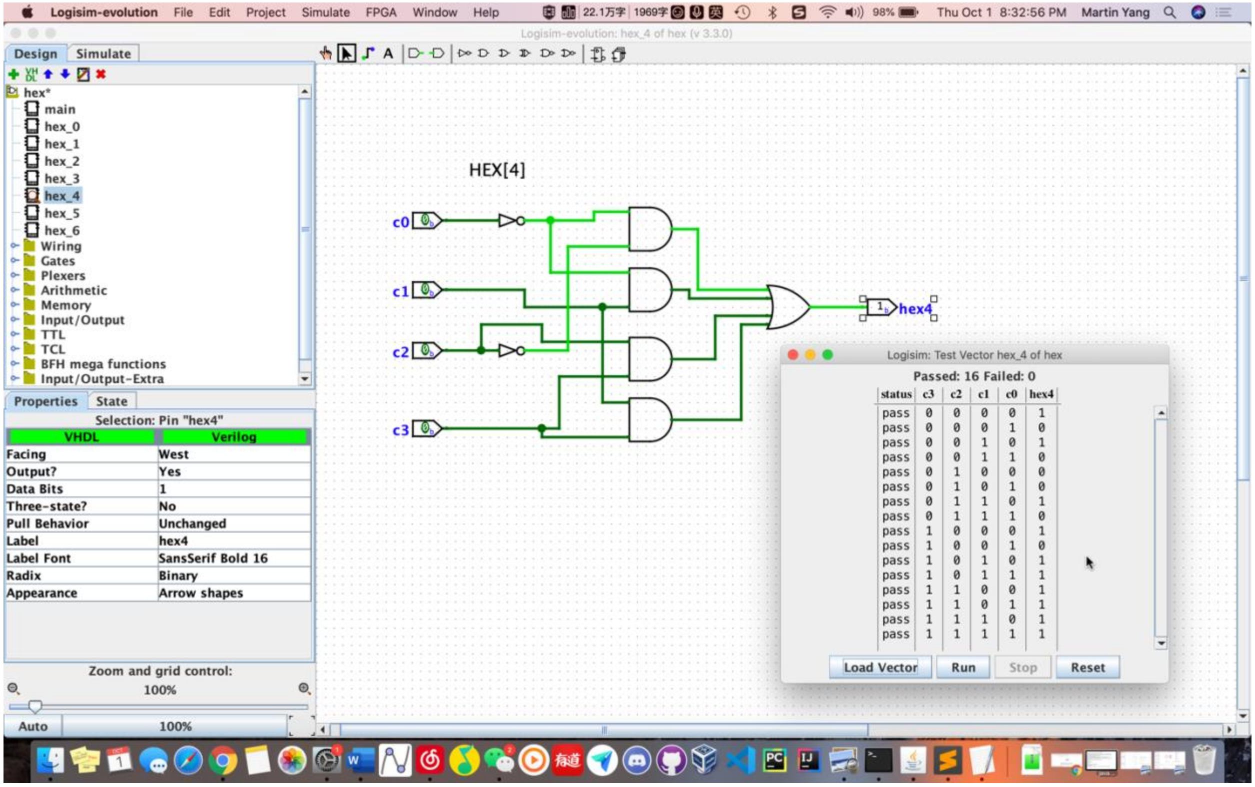
VHDL	Verilog
Text	HEX[3]
Font	SansSerif Plain 20
Horizontal Alignment	Center
Vertical Alignment	Base

Logisim: Test Vector hex_3 of hex

Passed: 16 Failed: 0

status	c3	c2	c1	c0	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	0
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

HEX[4]



Selection: Pin "hex4"

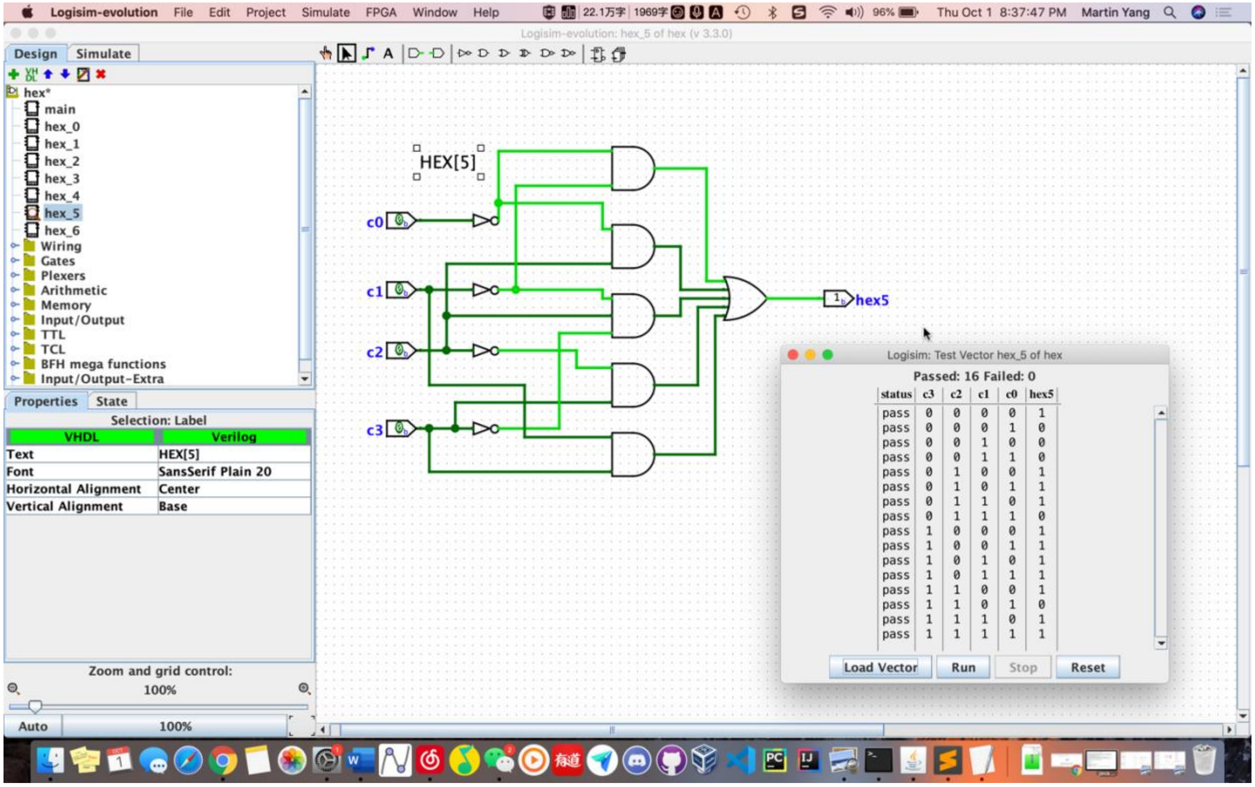
VHDL	Verilog
Facing	West
Output?	Yes
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	hex4
Label Font	SansSerif Bold 16
Radix	Binary
Appearance	Arrow shapes

Logisim: Test Vector hex_4 of hex

Passed: 16 Failed: 0

status	c3	c2	c1	c0	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

HEX[5]



Selection: Label

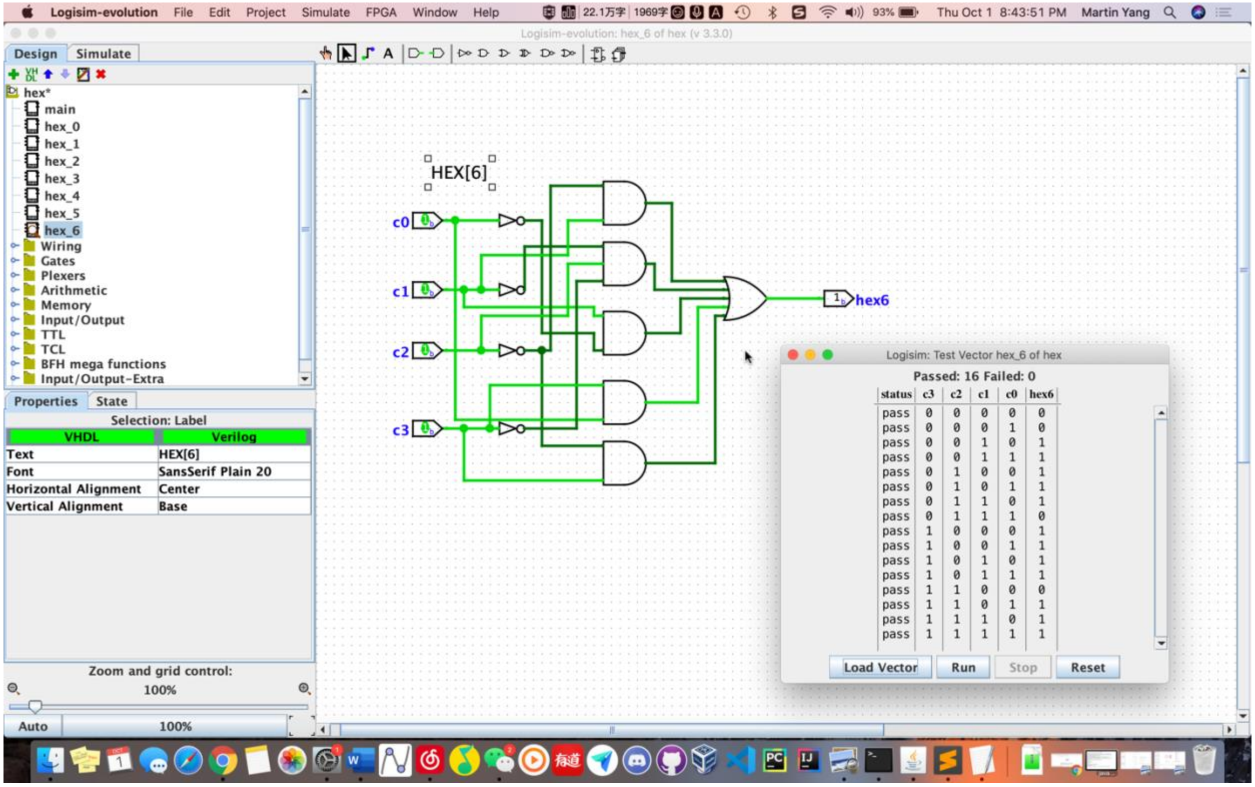
VHDL	Verilog
Text	HEX[5]
Font	SansSerif Plain 20
Horizontal Alignment	Center
Vertical Alignment	Base

Logisim: Test Vector hex_5 of hex

Passed: 16 Failed: 0

status	c3	c2	c1	c0	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

HEX[6]



Selection: Label

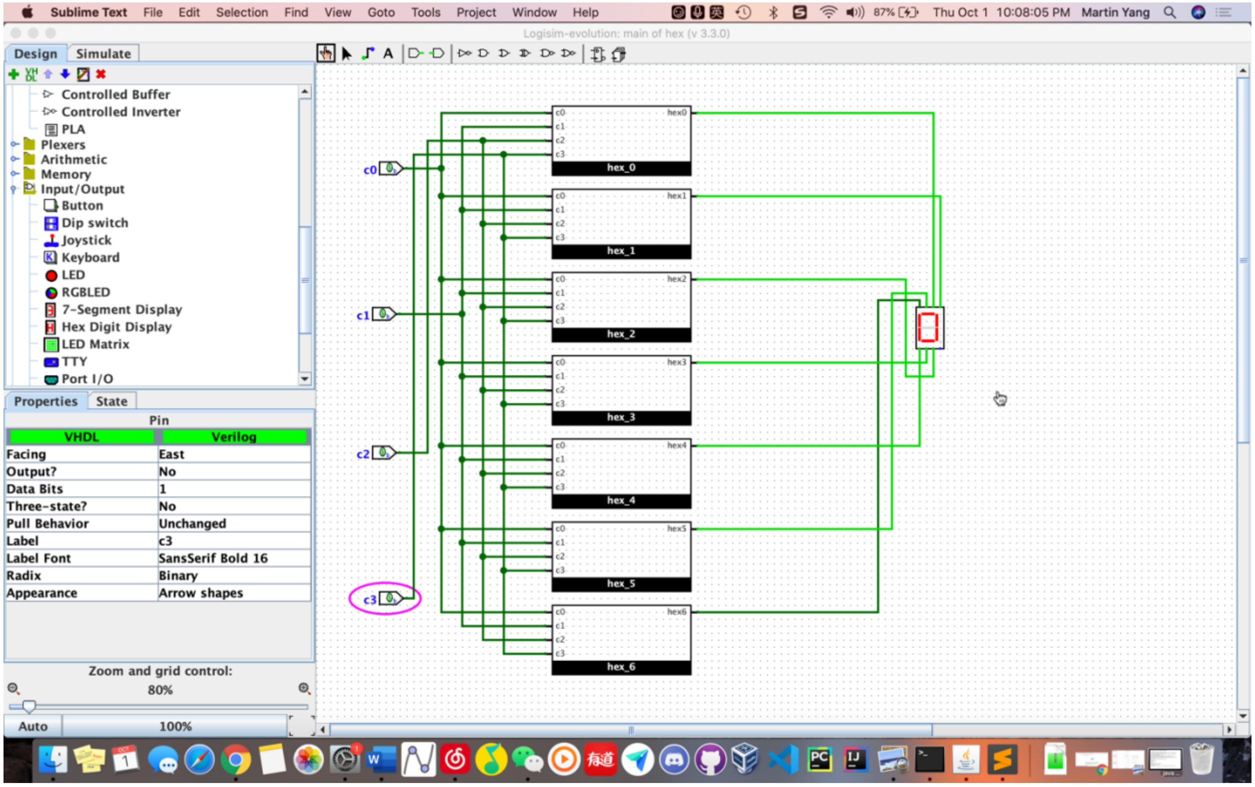
VHDL	Verilog
Text	HEX[6]
Font	SansSerif Plain 20
Horizontal Alignment	Center
Vertical Alignment	Base

Logisim: Test Vector hex_6 of hex

Passed: 16 Failed: 0

status	c3	c2	c1	c0	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

Last Result



Pin	
VHDL	Verilog
Facing	East
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	c3
Label Font	SansSerif Bold 16
Radix	Binary
Appearance	Arrow shapes