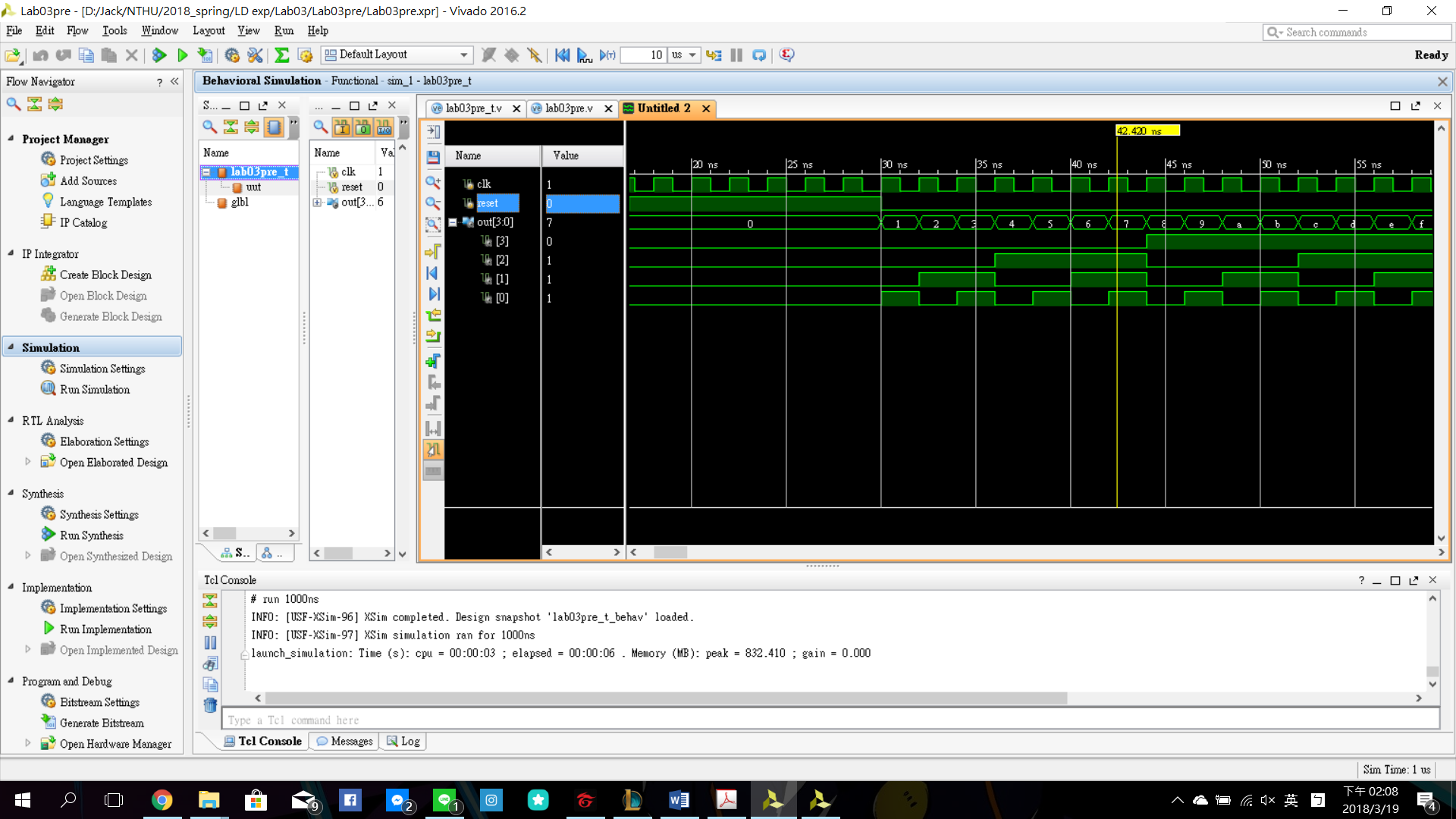
Lab03 counters 106061146 陳兆廷

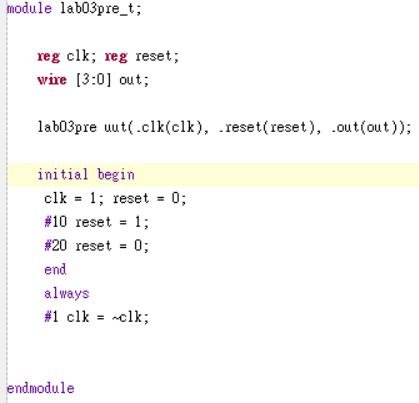
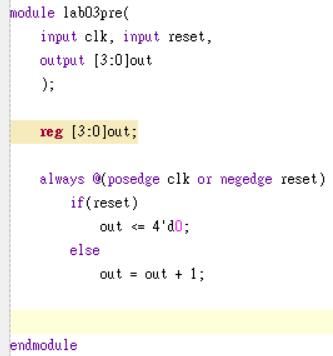
Pre-labs:

1. 4-bit synchronous binary up counter:
   1. 一張含有 文字, 白板 的圖片

      描述是以非常高的可信度產生Logic diagram:
   2. Verilog representation:

Verilog codes:

Experiments:



Module:

Testbench:

out++ when reset!=1, out=0 when reset = 1

1. Frequency divider:
   1. specification: a circuit that takes an input signal of a frequency f, and generates an output signal of a frequency fout = f/n, where n is an integer. The n is 2^27 in this case.
   2. 一張含有 文字, 白板 的圖片

      描述是以非常高的可信度產生Block diagram:
   3. Implementation:

一張含有 螢幕擷取畫面, 文字 的圖片

描述是以高可信度產生

One output signal(clk\_out)

Input clock and reset(clk, rst)

a parameter [26:0]cnt to count to 2^27

temp parameter

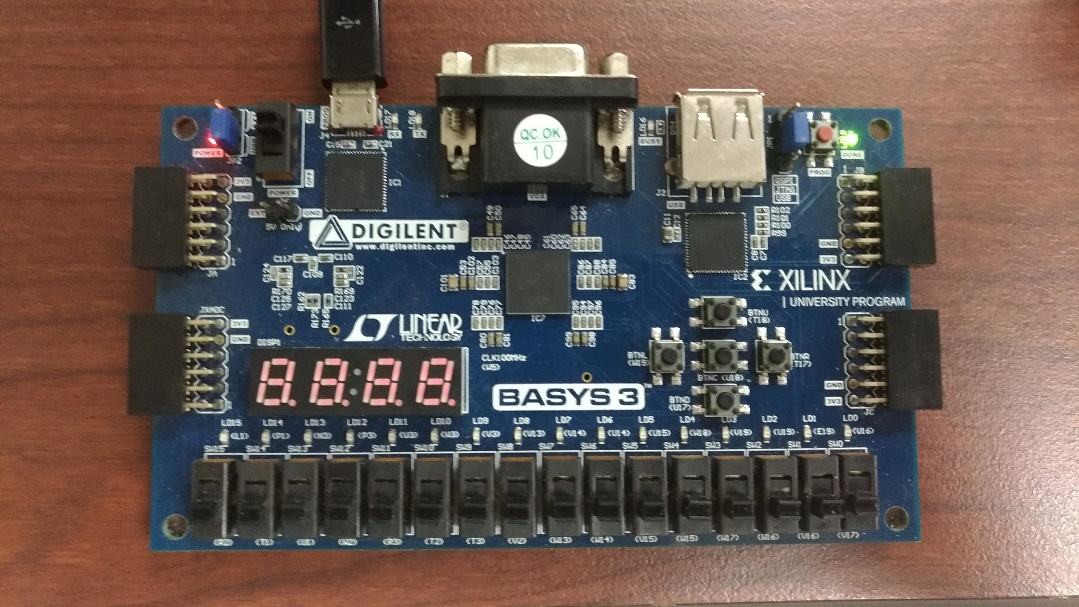
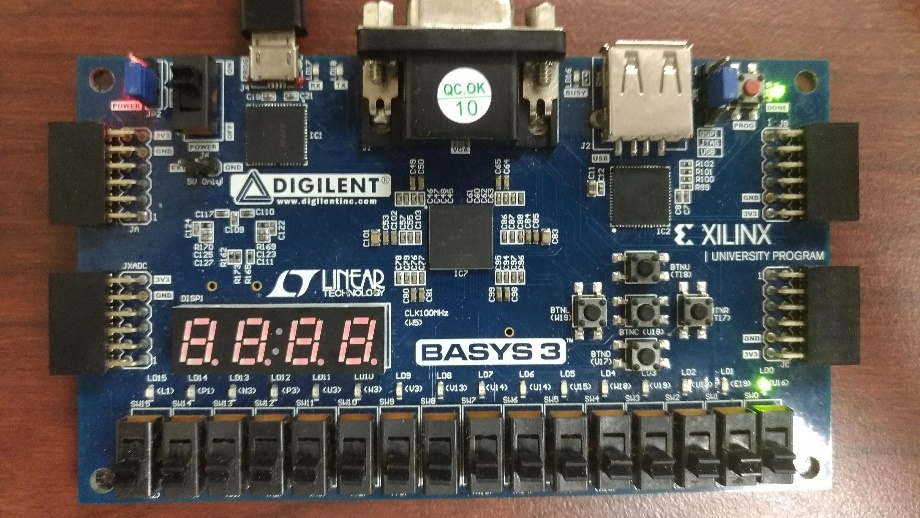
the output signal is the 26th bit of “count”

count keeps ++

if reset == 1, count returns to 0

else count ++

FPGA:



1. Count-for-50M frequency divider:
   1. Specification: a circuit that takes an input signal of a frequency f, and generates an output signal of a frequency: fout = f/n, where n is an integer. The n is 50M in this case.
   2. 一張含有 文字, 白板 的圖片

      描述是以非常高的可信度產生Block diagram:
   3. Implementation:

一張含有 螢幕擷取畫面 的圖片

描述是以非常高的可信度產生

Counter counts to 2^26(67108864>50000000)

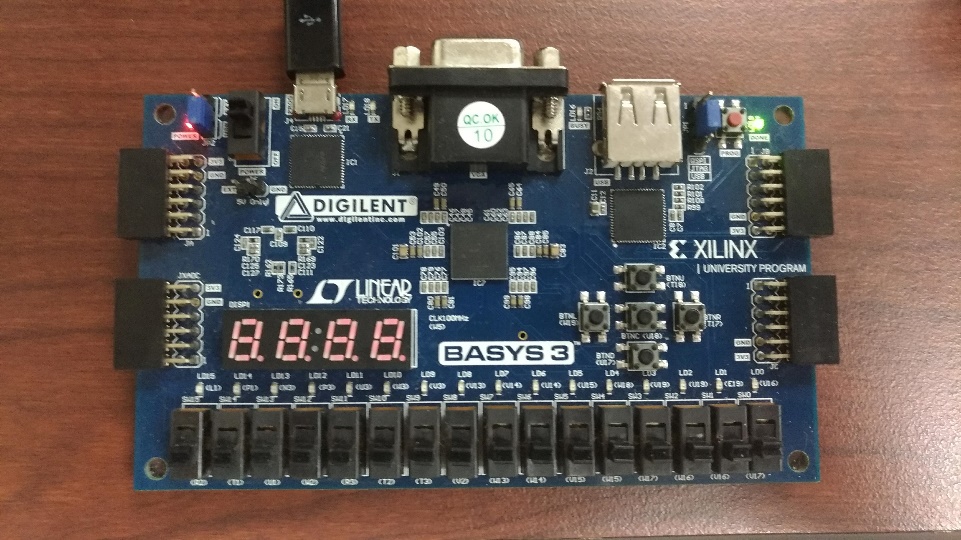
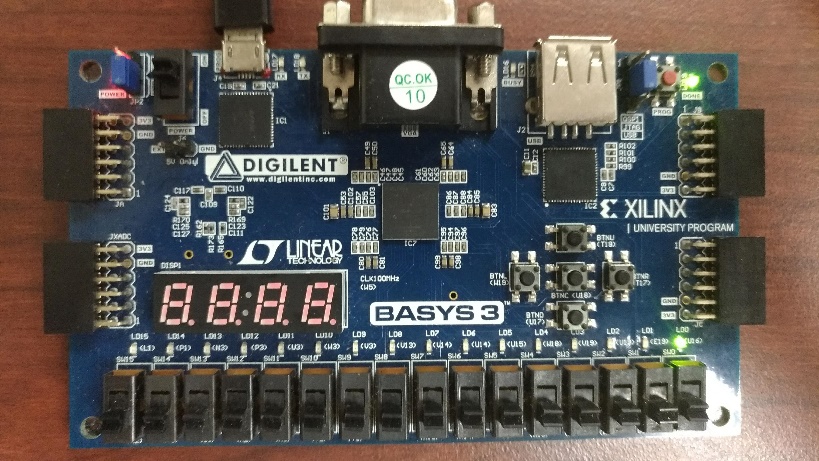
If reset == 1, then counter returns to 0.

If counter counts to 49999999(from 0)

Counter returns to 0.

clk\_out = ~clk\_out

counter ++ (always)



1. 4-bit synchronous binary up counter:

一張含有 螢幕擷取畫面, 文字 的圖片

描述是以非常高的可信度產生

input clock and reset

output a 4-bit b

create a 27-bit counter

if reset is 1

counter return to 0

b return to 0

if reset is 0

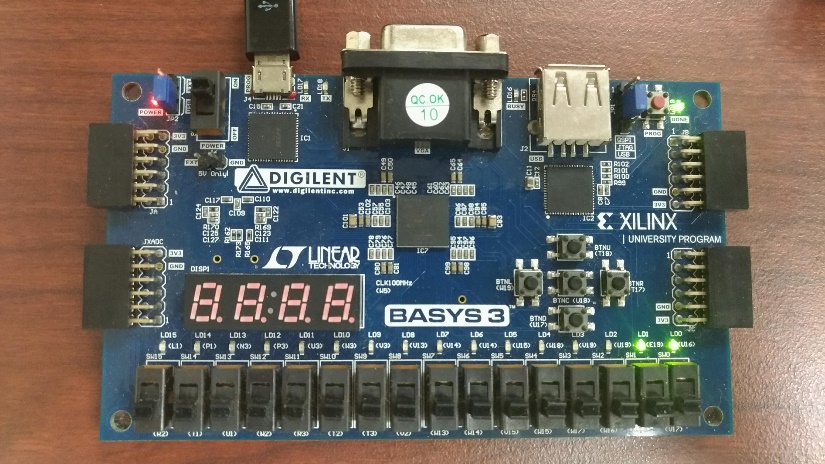
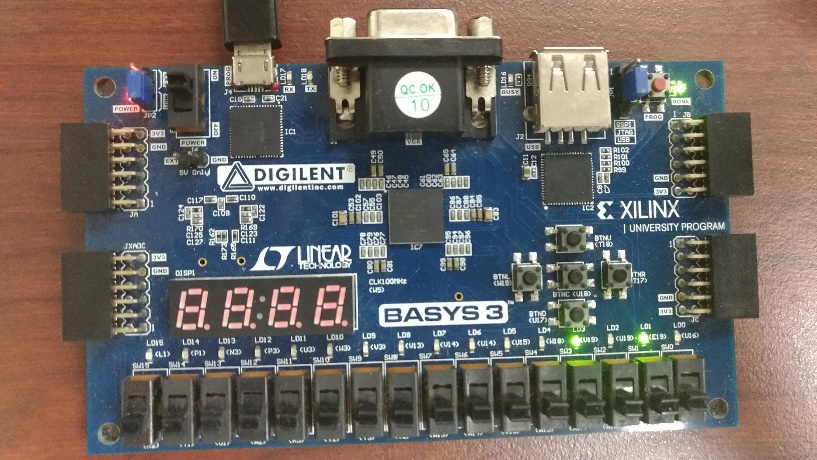
if counter added to 49999999

counter return to 0

b ++

counter ++ (if reset == 0)

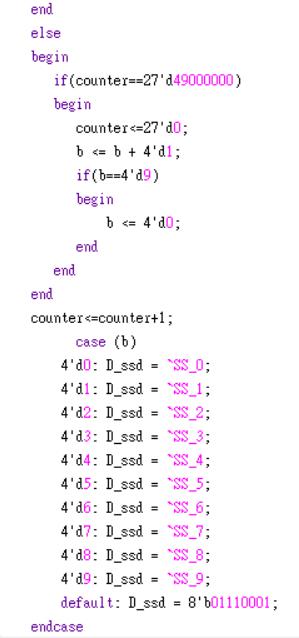
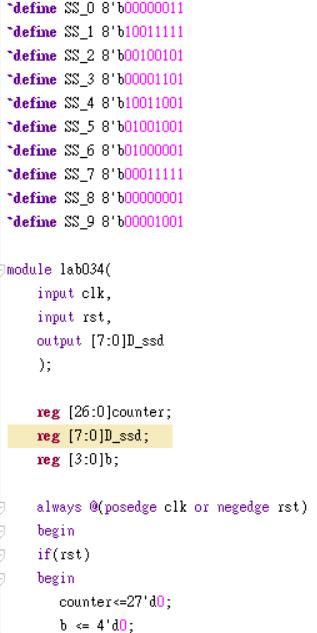
FPGA:



1. 一張含有 螢幕擷取畫面 的圖片

   描述是以非常高的可信度產生Single digit BCD up counter display on seven-segment display

Display module:



Seven-segment display code

An output for 7-seg display

A Counter to count to 49999999

b to calculate 1~10

if rst == 1 then return to 0

If counter == 49999999

( f = 50M)

Counter return to 0

b++

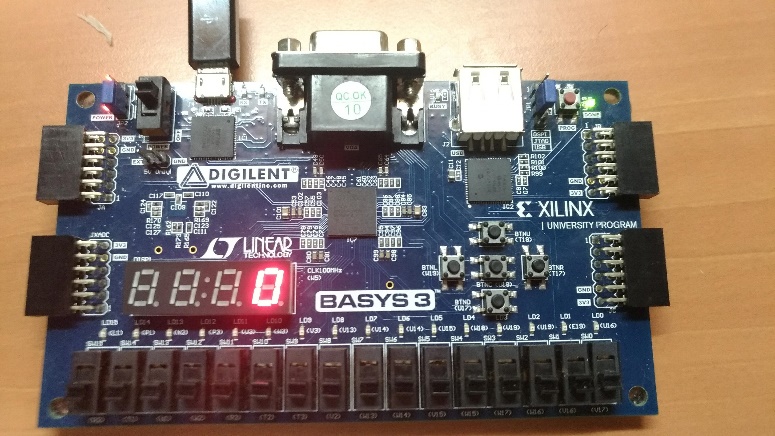
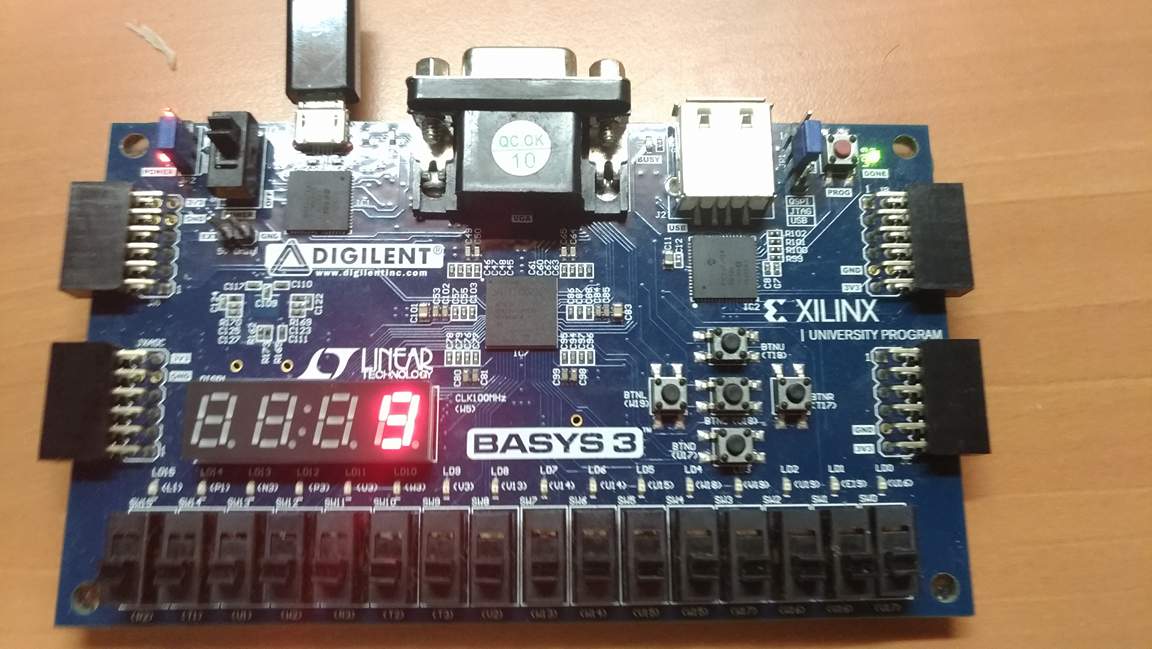
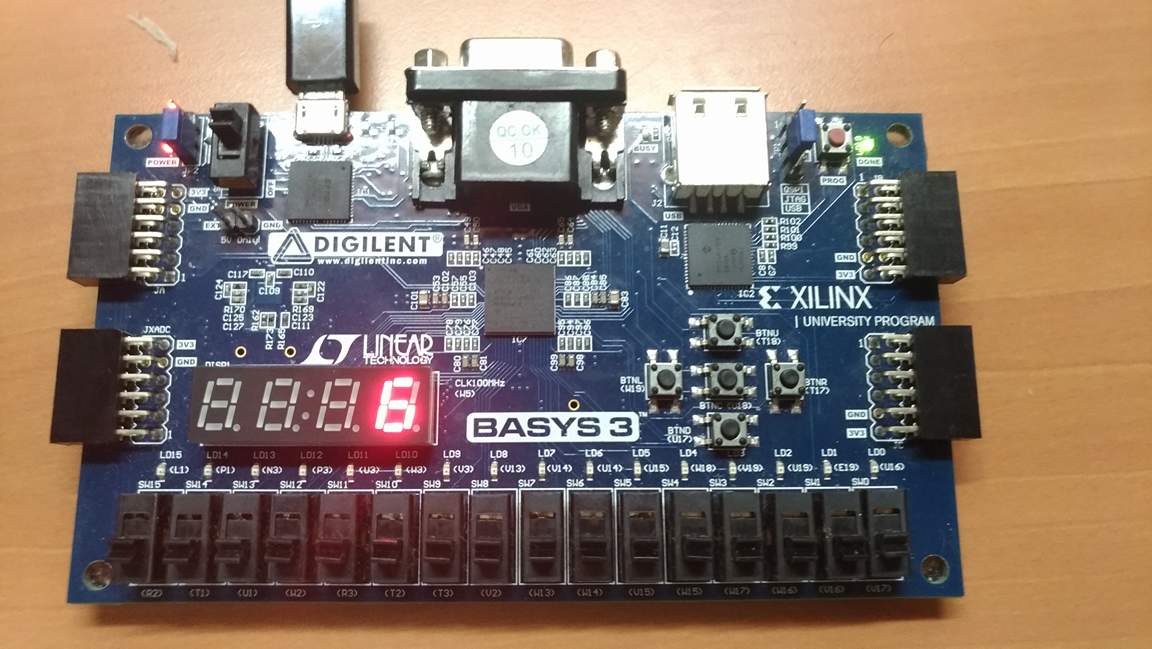
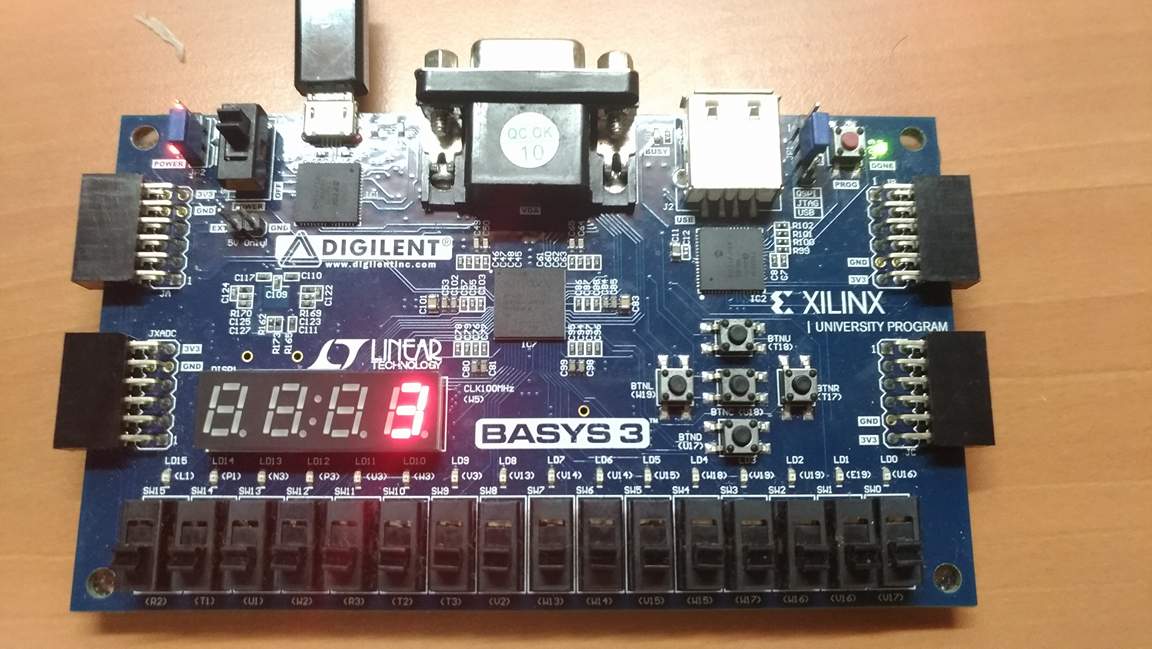
if b == 9

b return to 0

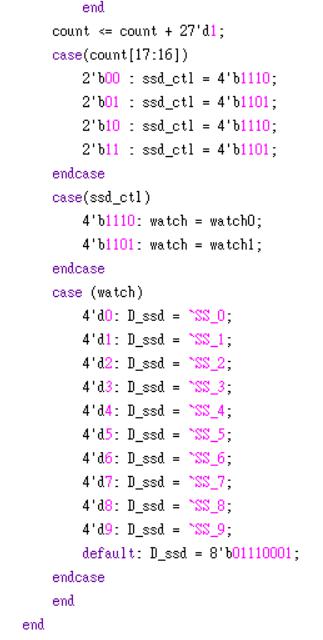
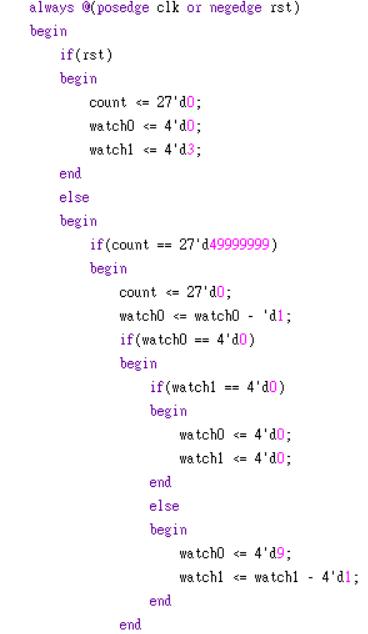
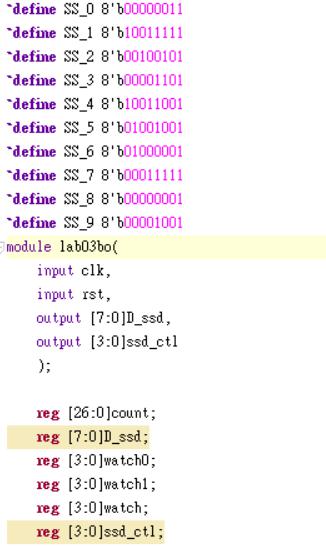
counter ++

b turn to 7-seg display

FPGA:



Bonus: 30-second count-down timer:



Seven-segment display code

An output for 7-seg display

An output for 7-seg control

A Counter to count to 49999999

Watch0 is the first digit

Watch1 is the second digit

If rst == 1

return to 30

If count to 49999999

Count return to 0

Watch0 – 1 (first digit)

If watch0 == 0

Then watch1 – 1 (second digit)

and watch0 return to 9

If all == 0 (00)

Then stops at 00

Count continue ++

Give ssd\_ctl a case of [17:16]

Then repeatedly change the on position on 7-seg display

Display the number on the right position

