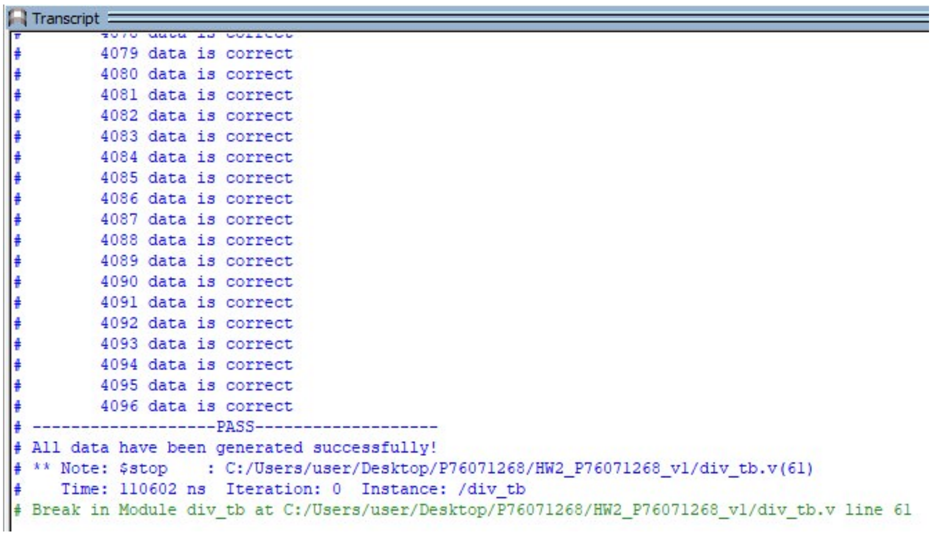
HOMEWORK 2

資工所 碩一 田少谷 P76071268

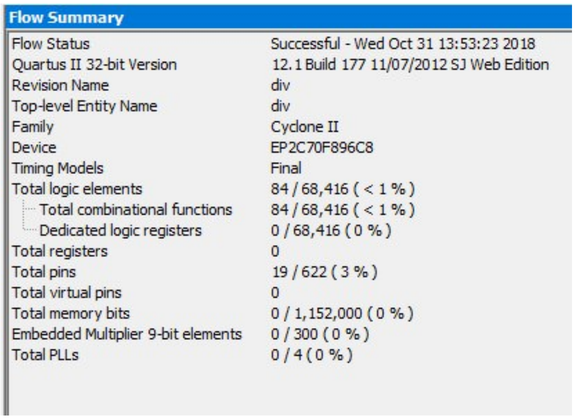
Functional simulation



Transcript

```
#       4079 data is correct
#       4080 data is correct
#       4081 data is correct
#       4082 data is correct
#       4083 data is correct
#       4084 data is correct
#       4085 data is correct
#       4086 data is correct
#       4087 data is correct
#       4088 data is correct
#       4089 data is correct
#       4090 data is correct
#       4091 data is correct
#       4092 data is correct
#       4093 data is correct
#       4094 data is correct
#       4095 data is correct
#       4096 data is correct
# -------------------PASS-------------------
# All data have been generated successfully!
# ** Note: $stop    : C:/Users/user/Desktop/P76071268/HW2_P76071268_v1/div_tb.v(61)
#    Time: 110602 ns  Iteration: 0  Instance: /div_tb
# Break in Module div_tb at C:/Users/user/Desktop/P76071268/HW2_P76071268_v1/div_tb.v line 61
```

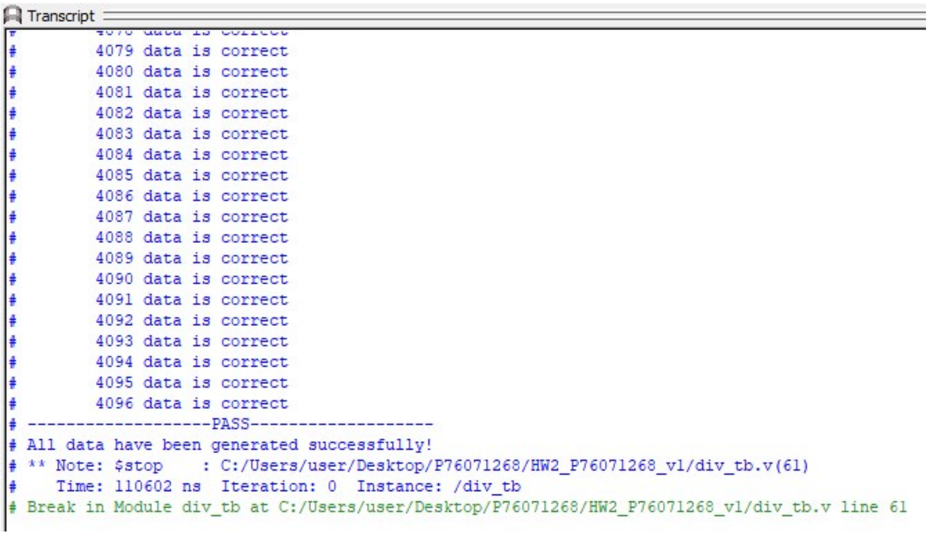
Synthesis by Quartus



Flow Summary

Flow Status	Successful - Wed Oct 31 13:53:23 2018
Quartus II 32-bit Version	12.1 Build 177 11/07/2012 SJ Web Edition
Revision Name	div
Top-level Entity Name	div
Family	Cyclone II
Device	EP2C70F896C8
Timing Models	Final
Total logic elements	84 / 68,416 (< 1 %)
Total combinational functions	84 / 68,416 (< 1 %)
Dedicated logic registers	0 / 68,416 (0 %)
Total registers	0
Total pins	19 / 622 (3 %)
Total virtual pins	0
Total memory bits	0 / 1,152,000 (0 %)
Embedded Multiplier 9-bit elements	0 / 300 (0 %)
Total PLLs	0 / 4 (0 %)

Gate-Level Simulation (cycle= 27)



Transcript

```
#       4079 data is correct
#       4080 data is correct
#       4081 data is correct
#       4082 data is correct
#       4083 data is correct
#       4084 data is correct
#       4085 data is correct
#       4086 data is correct
#       4087 data is correct
#       4088 data is correct
#       4089 data is correct
#       4090 data is correct
#       4091 data is correct
#       4092 data is correct
#       4093 data is correct
#       4094 data is correct
#       4095 data is correct
#       4096 data is correct
# -------------------PASS-------------------
# All data have been generated successfully!
# ** Note: $stop    : C:/Users/user/Desktop/P76071268/HW2_P76071268_v1/div_tb.v(61)
#    Time: 110602 ns  Iteration: 0  Instance: /div_tb
# Break in Module div_tb at C:/Users/user/Desktop/P76071268/HW2_P76071268_v1/div_tb.v line 61
```

Explanation

My method of implemeneting Dividor is to use “carry”.
First I put the “input in1” into a “reg tem[0:11]'s [5:0] position”, then by each iteration shift left “tem” by one bit and compare its [11:6] value to “input in2”.
If it’s bigger than in2, substract in2 from in1[11:6], and add 1 to “temre[5:0]'s highest position(ex: [5] in first iteration)”; else, assign 0 to temre’s corresponding position and pass current value of tem to the next run.
Repete this process for 6 turns(length of in1, in2).

tags: `DIC` `dividor` `no clk`