2020 Digital IC Design Homework 1: 4-bit binary adder-subtractor

NAME	Nguyen Vu Le Minh
Student ID	P76087099

Simulation Result

Functional simulation	Pass	Gate-level simulation	Pass	Gate-level simulation time	10852 (ns)

(your pre-sim result)

```
#          502 data is correct
#          503 data is correct
#          504 data is correct
#          505 data is correct
#          506 data is correct
#          507 data is correct
#          508 data is correct
#          509 data is correct
#          510 data is correct
#          511 data is correct
#          512 data is correct
# --------------------PASS--------------------
# All data have been generated successfully!
```

(your post-sim result)

```
#          503 data is correct
#          504 data is correct
#          505 data is correct
#          506 data is correct
#          507 data is correct
#          508 data is correct
#          509 data is correct
#          510 data is correct
#          511 data is correct
#          512 data is correct
# --------------------PASS--------------------
# All data have been generated successfully!
```

Synthesis Result

Total logic elements	10 / 68,416 (< 1 %)
Total memory bit	0 / 1,152,000 (0 %)
Embedded multiplier 9-bit element	0 / 300 (0 %)

(your flow summary)

Flow Summary

Flow Status	Successful - Sat Apr 11 23:10:22 2020
Quartus II Version	10.0 Build 262 08/18/2010 SP 1 SJ Full Version
Revision Name	AS
Top-level Entity Name	AS
Family	Cyclone II
Device	EP2C70F896C8
Timing Models	Final
Met timing requirements	Yes
Total logic elements	10 / 68,416 (< 1 %)
Total combinational functions	10 / 68,416 (< 1 %)
Dedicated logic registers	0 / 68,416 (0 %)
Total registers	0
Total pins	14 / 622 (2 %)
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 %)

Description of your design

The binary adder-subtractor is a device that is capable of adding or subtracting elements. A circuit which is used to add or subtract, depending on the control signal. In my homework, I used dataflow description to design 4-bit binary adder-subtractor.

```
module AS(sel, A, B, S, O);
    input [3:0] A, B;
    input sel;
    output [3:0] S;
    output O;
```

First, my input includes A, B, sel. The input A and B which are 4-bit binary, push from file A.txt and B.txt. The input “sel” that are 1-bit binary, decide to design one of adder-subtractor. Besides, there are 2 output value that are S and C in my design.

```
        Single_State s0( .a( A[0] ), .b(sel^B[0]), .cin( sel ),     .s( S[0]), .cout( ripple0 ) );
        Single_State s1( .a( A[1] ), .b(sel^B[1]), .cin( ripple0 ), .s( S[1]), .cout( ripple1 ) );
        Single_State s2( .a( A[2] ), .b(sel^B[2]), .cin( ripple1 ), .s( S[2]), .cout( ripple2 ) );
        Single_State s3( .a( A[3] ), .b(sel^B[3]), .cin( ripple2 ), .s( S[3]), .cout( ripple3 ) );

endmodule

module Single_State(
    input a,
    input b,
    input cin,
    output  s,
    output cout );

    assign s    = a ^ b ^ cin;
    assign cout = (a & b)  |  (a & cin)  |  (b & cin);
endmodule
```

According to block overview, I separated input A and B into each of bit in order to process. Besides, I have a carry bit that is “cin” - 1 bit binary to memo overflow bit of each stage. Initial of “cin” by “sel”. In processing each bit, I used XOR to calculate “sum” output and (AND, OR) to calculate “cin”. The processing will loop 4 times because input is 4-bit binary. Depending on the “sel”, if sel is 1, the design will be subtractor and other case, it will be adder. Finally, in order to calculate overflow bit, I used XOR between "cin3" and "cin4".

I also edited the Clock Cycle is 2.1 in Test_bench file to decrease gate-level simulation time in post-sim.

```
`timescale 10ns / 1ps
`define CYCLE 2.1
`define A_dat "./A.txt"
`define B_dat "./B.txt"
`define O_dat "./O.txt"
`define SUM_dat "./SUM.txt"
module AS_tb;
```

*Scoring = (Total logic elements + total memory bit + 9*embedded multiplier 9-bit element) × (gate-level simulation time in ns)*