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

NAME	崔濟鵬
Student ID	F74056069

Simulation Result

Functional simulation	Pass	Gate-level simulation	Pass	Gate-level simulation time	25700 (ns)

```
#          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!
# ** Note: $finish    : D:/Code/git/DIGITAL-IC-DESIG
#    Time: 25700 ns  Iteration: 0  Instance: /AS_tb
# 1
# Break in Module AS_tb at D:/Code/git/DIGITAL-IC-DE
```

```
#          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!
# ** Note: $finish    : D:/Code/git/DIGITAL-IC-DESI
#    Time: 25700 ns  Iteration: 0  Instance: /AS_tb
# 1
# Break in Module AS_tb at D:/Code/git/DIGITAL-IC-D
```

Synthesis Result

Total logic elements	8 / 68,416 (< 1 %)
Total memory bit	0 / 1,152,000 (0 %)
Embedded multiplier 9-bit element	0 / 300 (0 %)

Flow Status	Successful - Tue Apr 07 14:33:20 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	8 / 68,416 (< 1 %)
Total combinational functions	8 / 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

用了三個 wire 來儲存 B xor sel, A xor B, C，因為每個 FA 前都會將 B xor sel 後，當成 FA 的 B 輸入，故先用 4 個 xor()來運算給 4 個 FA 的 B，而在 FA 中會重複利用到 A xor B 運算 S 跟 C，故一樣用 4 個 xor()來運算 FA 中的 A xor B，最終使用 assign 來算 4 組 S 跟 C，O 為 C[2] xor C[3]

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