

# 數位系統技術



## 4-Bit Adder/Subtractor

Ren-Der Chen (陳仁德)

Department of Computer Science and  
Information Engineering

National Changhua University of Education

E-mail: rdchen@cc.ncue.edu.tw

Spring, 2025

# 實驗內容

---

- 二進位數字之1's及2's complement。
- 4-bit正負數之加/減法運算。
- 使用Verilog設計4-bit adder/subtractor，並以實驗板驗證其功能。

# 2's Complement

---

- 二進位數字B之2's complement為B之負數表示法，其轉換方式為1's complement 再加上1
- B之1's complement轉換方式為B中每個位元1變0，0變1
- Example
  - $B = 0101$  (5), 1's complement = 1010, 2's complement =  $1010 + 1 = 1011$  (-5)
  - $7 - 5 = 7 + (-5) = 0111 + 1011 = 0010 = 2$
  - $1 - 5 = 1 + (-5) = 0001 + 1011 = 1100 = -4$  (1's of 1100 = 0011, 2's of 1100 = 0011+1 = 0100 = 4)

# 4-Bit Signed Number

---

- 4-bit signed number 可表示之值的範圍為 **-8 ~ 7**, 第一個bit 為sign bit, 0為正, 1為負

0000 -> 0

0001 -> 1, 1111 -> **-1**

0010 -> 2, 1110 -> **-2**

0011 -> 3, 1101 -> **-3**

0100 -> 4, 1100 -> **-4**

0101 -> 5, 1011 -> **-5**

0110 -> 6, 1010 -> **-6**

0111 -> 7, 1001 -> **-7**

1000 -> **-8**

# Overflow

---

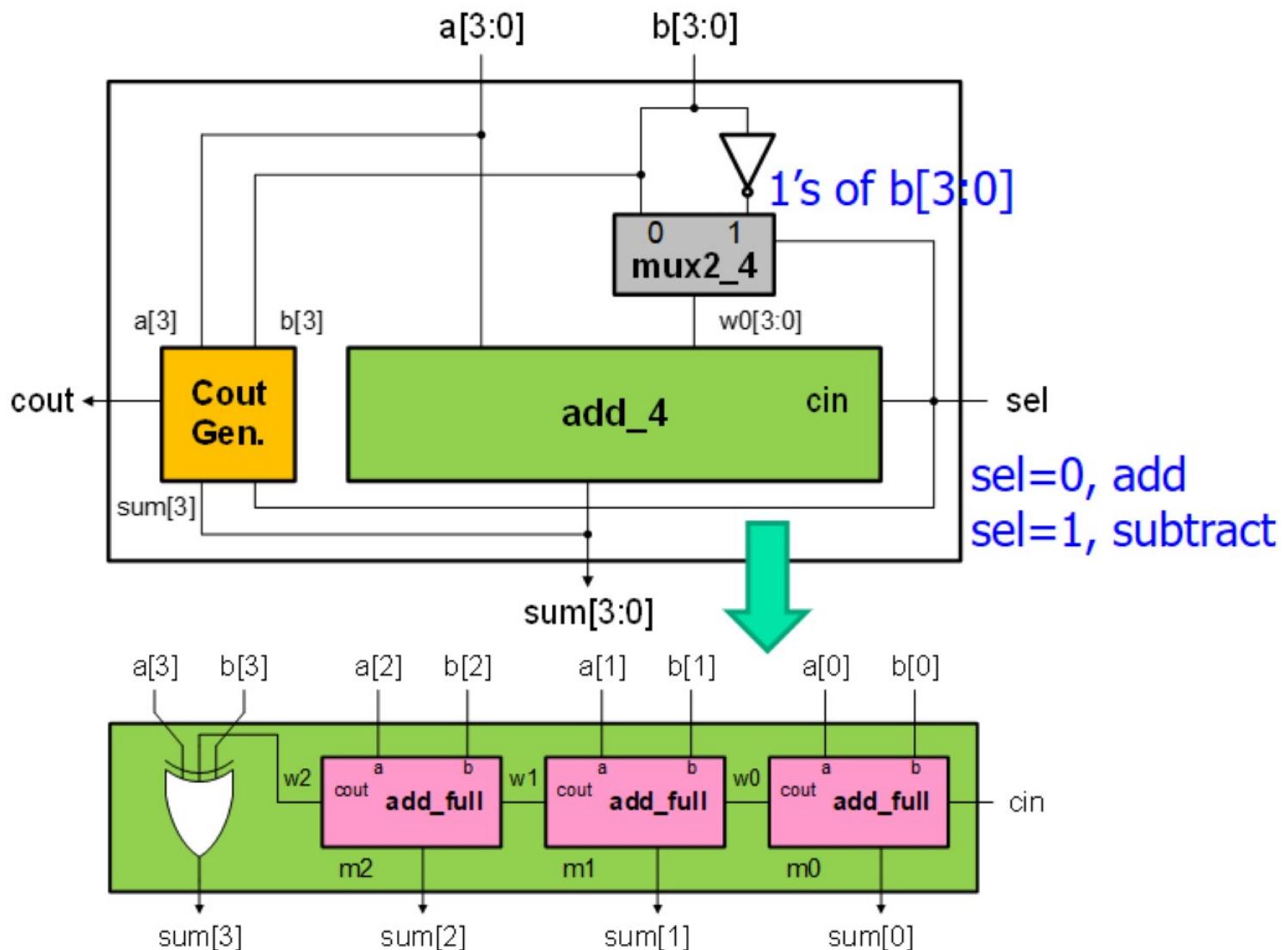
- **Overflow**: 兩個4-bit signed number之運算結果(4-bit)大於7
- Overflow 1:  $A \geq 0, B \geq 0, Y = A + B, Y < 0$   
 $A = 0100$  (4),  $B = 0101$  (5)  
 $Y = A + B = 0100 + 0101 = \underline{1}001 < 0$  (理論上應為9, 但因bit數不夠, 無法由4-bit signed number表示)
- Overflow 2:  $A \geq 0, B < 0, Y = A - B, Y < 0$   
 $A = 0011$  (3),  $B = 1011$  (-5)  
 $Y = A - B = A + (-B) = 0011 + 0101 = \underline{1}000 < 0$  (理論上應為8, 但因bit數不夠, 無法由4-bit signed number表示)

# Underflow

---

- Underflow: 兩個4-bit signed number之運算結果(4-bit)小於 -8
- Underflow 1:  $A < 0, B < 0, Y = A + B, Y \geq 0$   
 $A = 1101 (-3), B = 1010 (-6)$   
 $Y = A + B = 1101 + 1010 = \underline{0}111 > 0$  (理論上應為-9，但因bit數不夠，無法由4-bit signed number表示)
- Underflow 2:  $A < 0, B \geq 0, Y = A - B, Y \geq 0$   
 $A = 1101 (-3), B = 0110 (6)$   
 $Y = A - B = A + (-B) = 1101 + 1010 = \underline{0}111 > 0$  (理論上應為-9，但因bit數不夠，無法由4-bit signed number表示)

# 4-Bit Adder/Subtractor 電路方塊圖



# Design 1: Half Adder (add\_half.v)

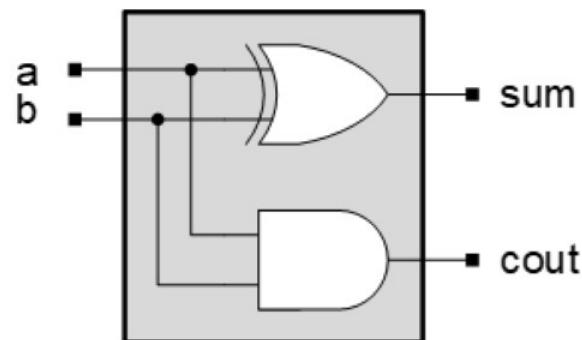
```
module add_half(a, b, cout, sum);  
    input a, b;  
    output cout, sum;  
  
    assign cout = a&b;  
    assign sum =   
endmodule
```



a\b	0	1	a\b	0	1
0	0	1	0	0	0
1	1	0	1	0	1

$$\text{sum} = a \oplus b$$

$$\text{cout} = a \cdot b$$



## Design 2: Full Adder (add\_full.v) (1/2)

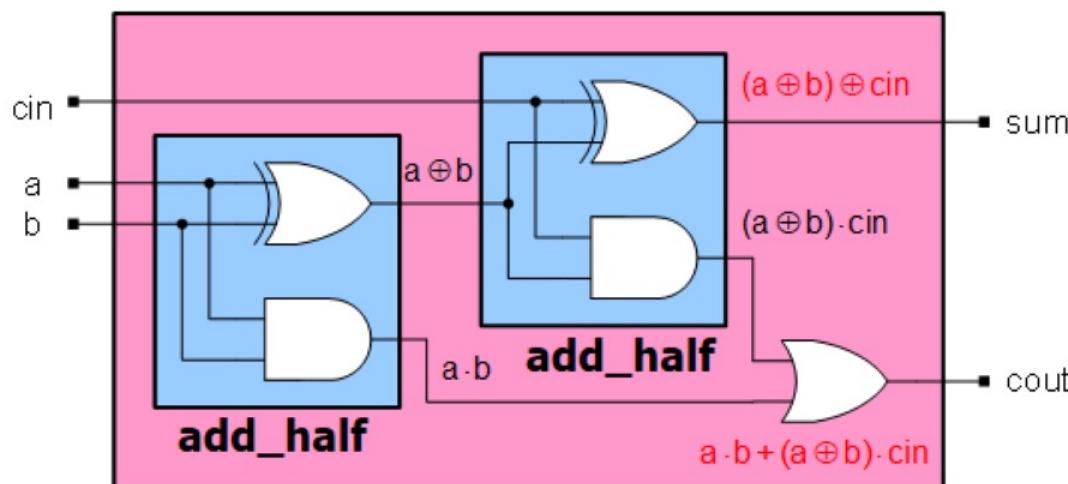


ab\cin	0	1
00	0	1
01	1	0
11	0	1
10	1	0

$$\text{sum} = (a \oplus b) \oplus \text{cin}$$

ab\cin	0	1
00	0	0
01	0	1
11	1	1
10	0	1

$$\text{cout} = a \cdot b + (a \oplus b) \cdot \text{cin}$$

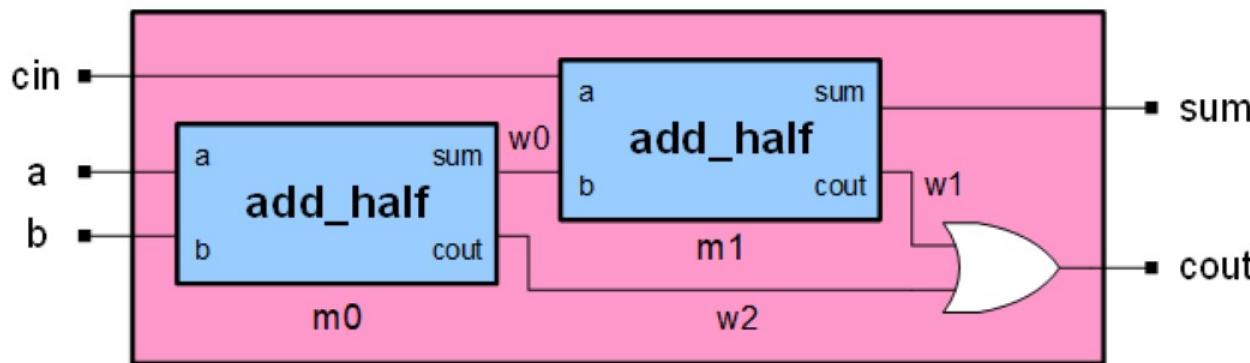


## Design 2: Full Adder (add\_full.v) (2/2)

```
module add_full(a, b, cin, cout, sum);
    input a, b, cin;
    output cout, sum;
    wire w0, w1, w2;

    add_half m0(a, b, w2, w0);
    add_half m1(a, b, w1, w2);

    assign cout = w1;
endmodule
```

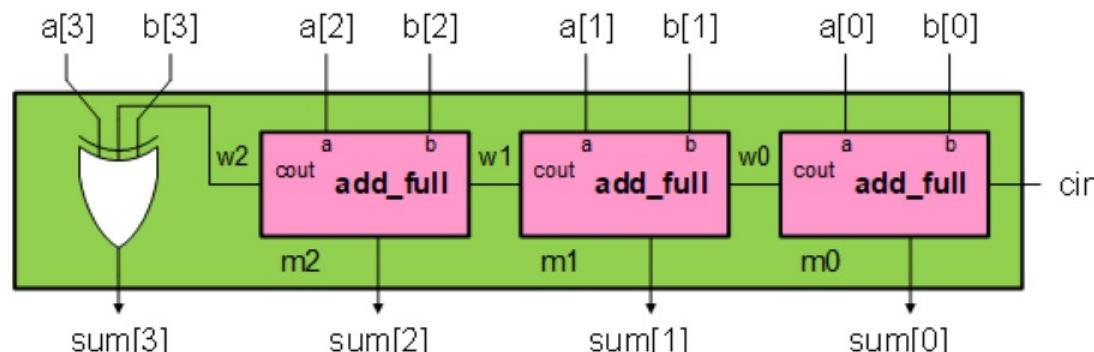


# Design 3: 4-Bit Adder without Carry Out (add\_4.v)

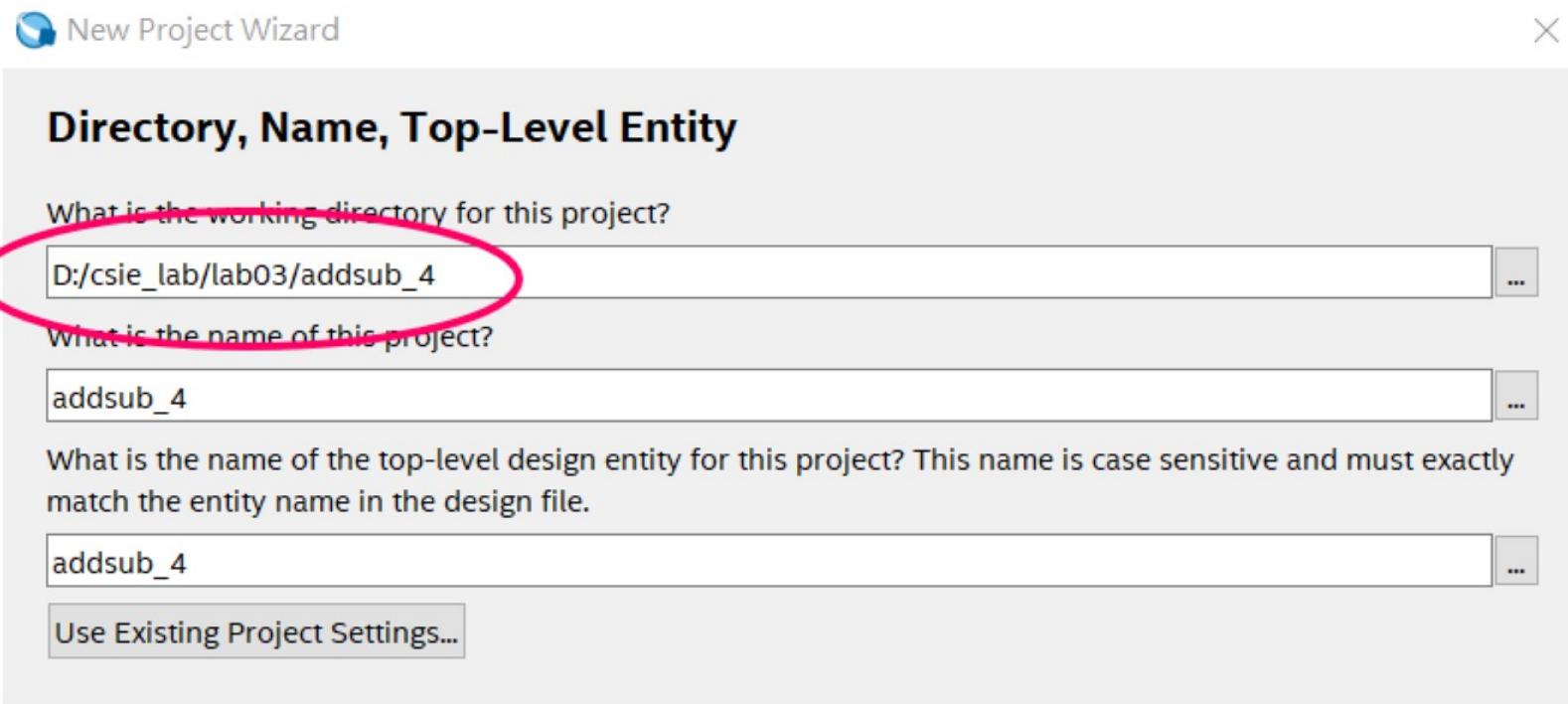
```
module add_4 (a, b, cin, sum);
    input [3:0] a,b;
    input cin;
    output [3:0] sum;

    wire w0, w1, w2;

    add_full m0(a[0], b[0], cin, w0, sum[0]);
    add_full m1(a[1], b[1], w0, w1, sum[1]);
    add_full m2(a[2], b[2], w1, w2, sum[2]);
    assign sum[3] = w2;
endmodule
```



# Create New Project - addsub\_4



# Add Files

New Project Wizard X

## Add Files

Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project.

Note: you can always add design files to the project later.

File name:  ... Add

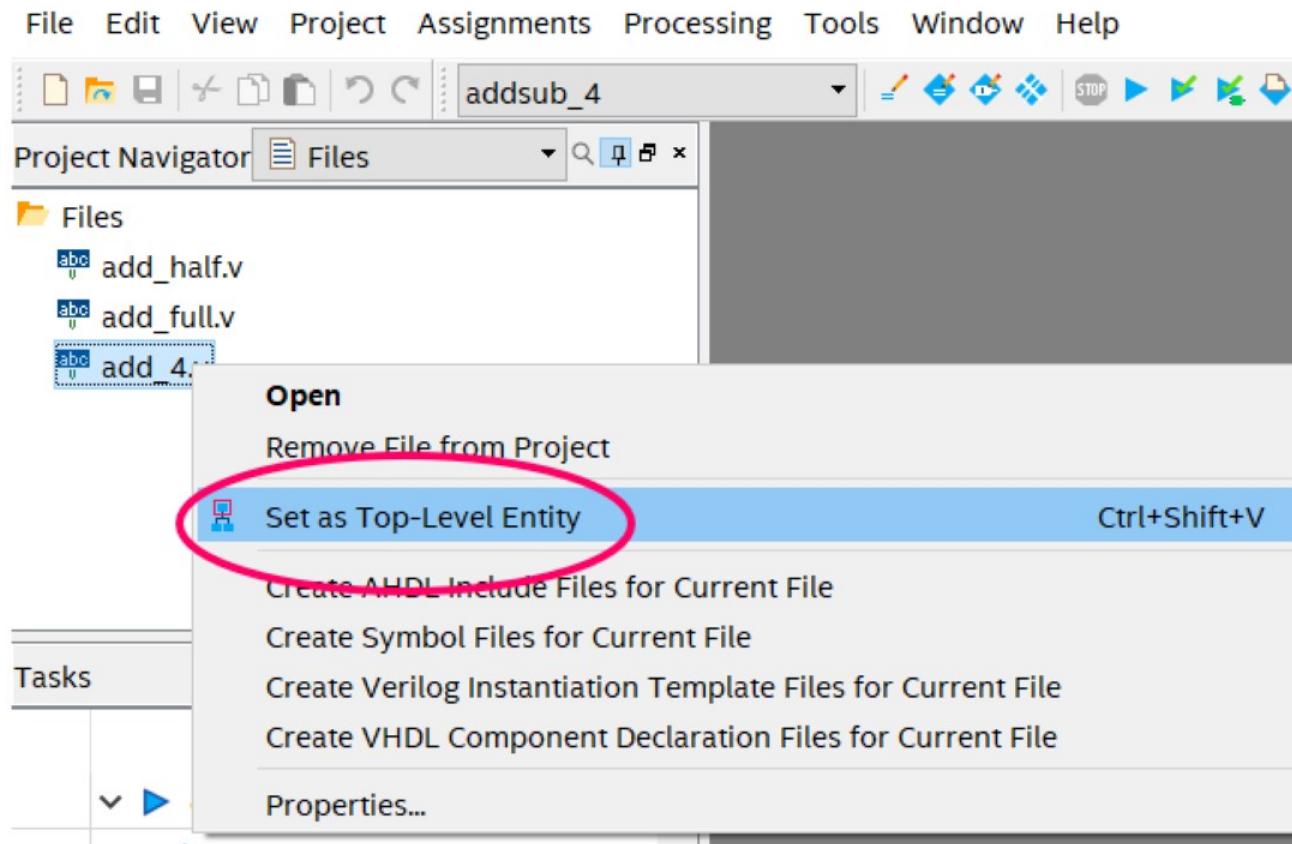
X Add All

File Name	Type	Library	Design Entry/Synthesis Tool	HDL Version	Actions
add_half.v	Ver...			Default	<span style="border: 1px solid #ccc; padding: 2px 10px; border-radius: 5px;">Remove</span> <span style="margin-left: 10px;">Up</span> <span style="margin-left: 10px;">Down</span> <span style="margin-left: 10px;">Properties</span>
add_full.v	Ver...			Default	
add_4.v	Ver...			Default	

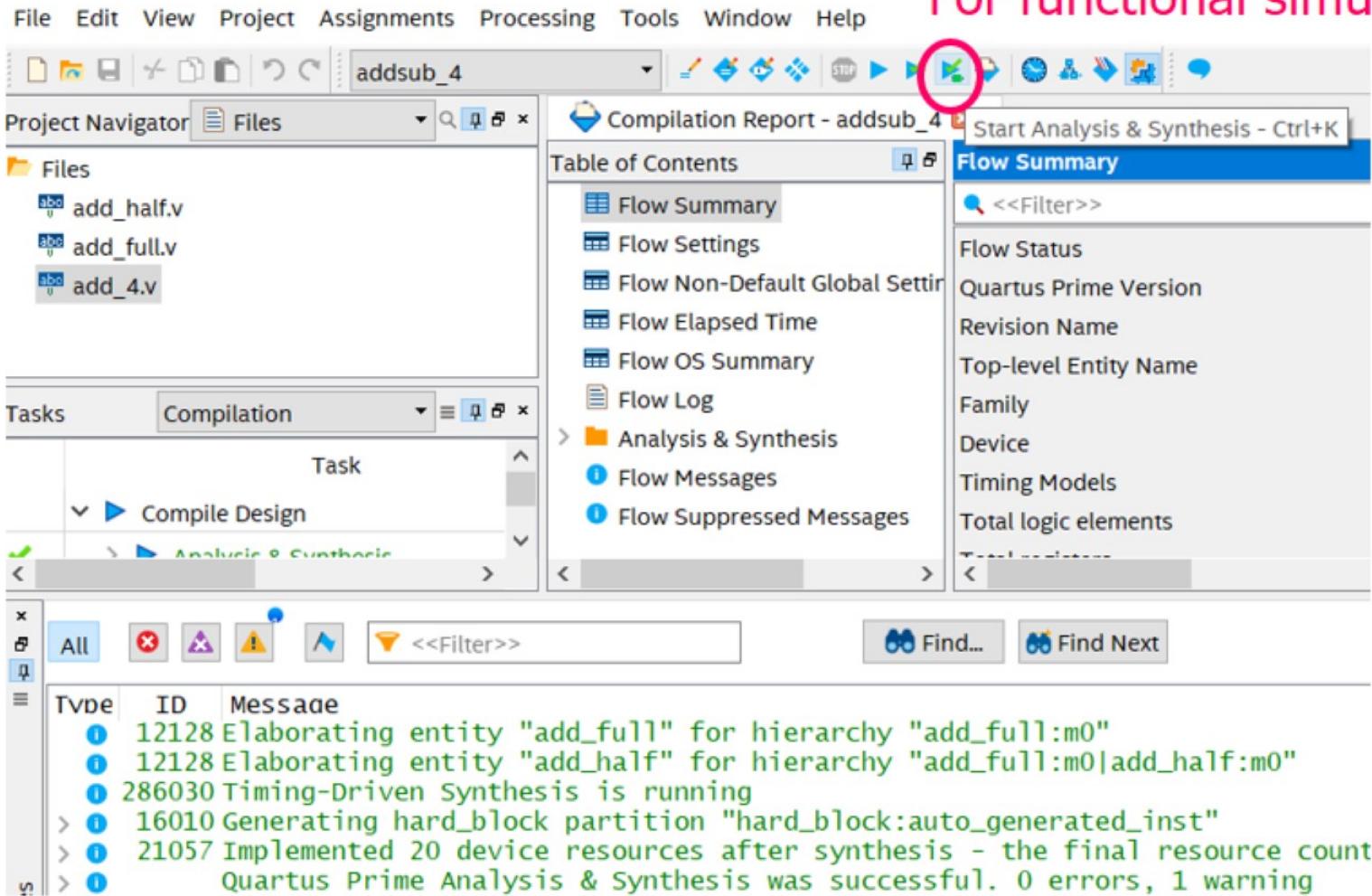
Specify the path names of any non-default libraries. User Libraries...

< Back Next > Finish Cancel Help

# Set “add\_4.v” as Top-Level Entity



# Start Analysis and Synthesis

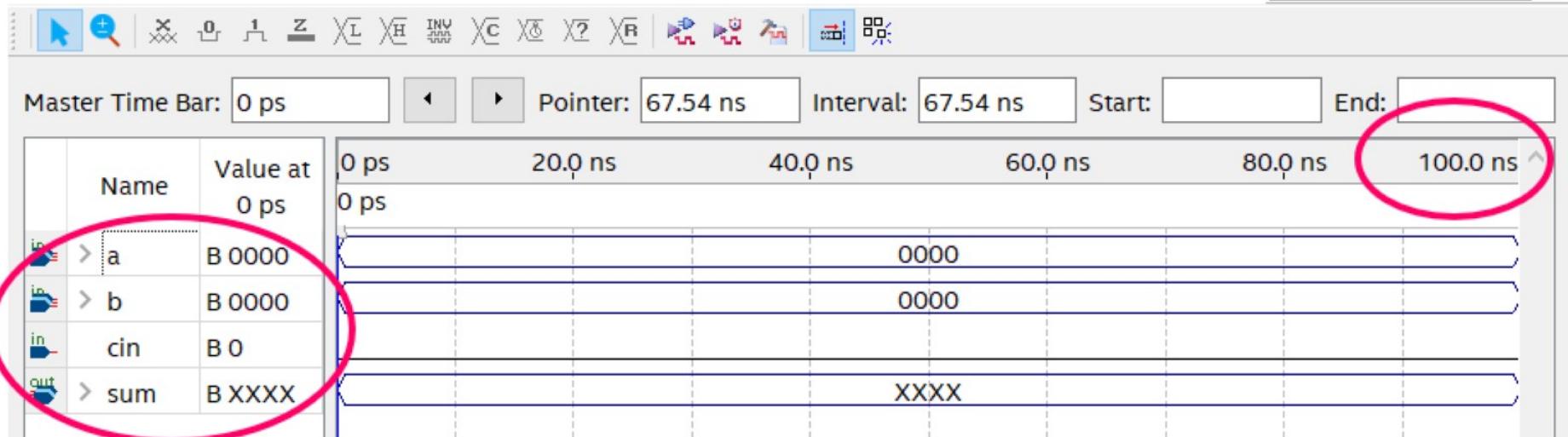


For functional simulation

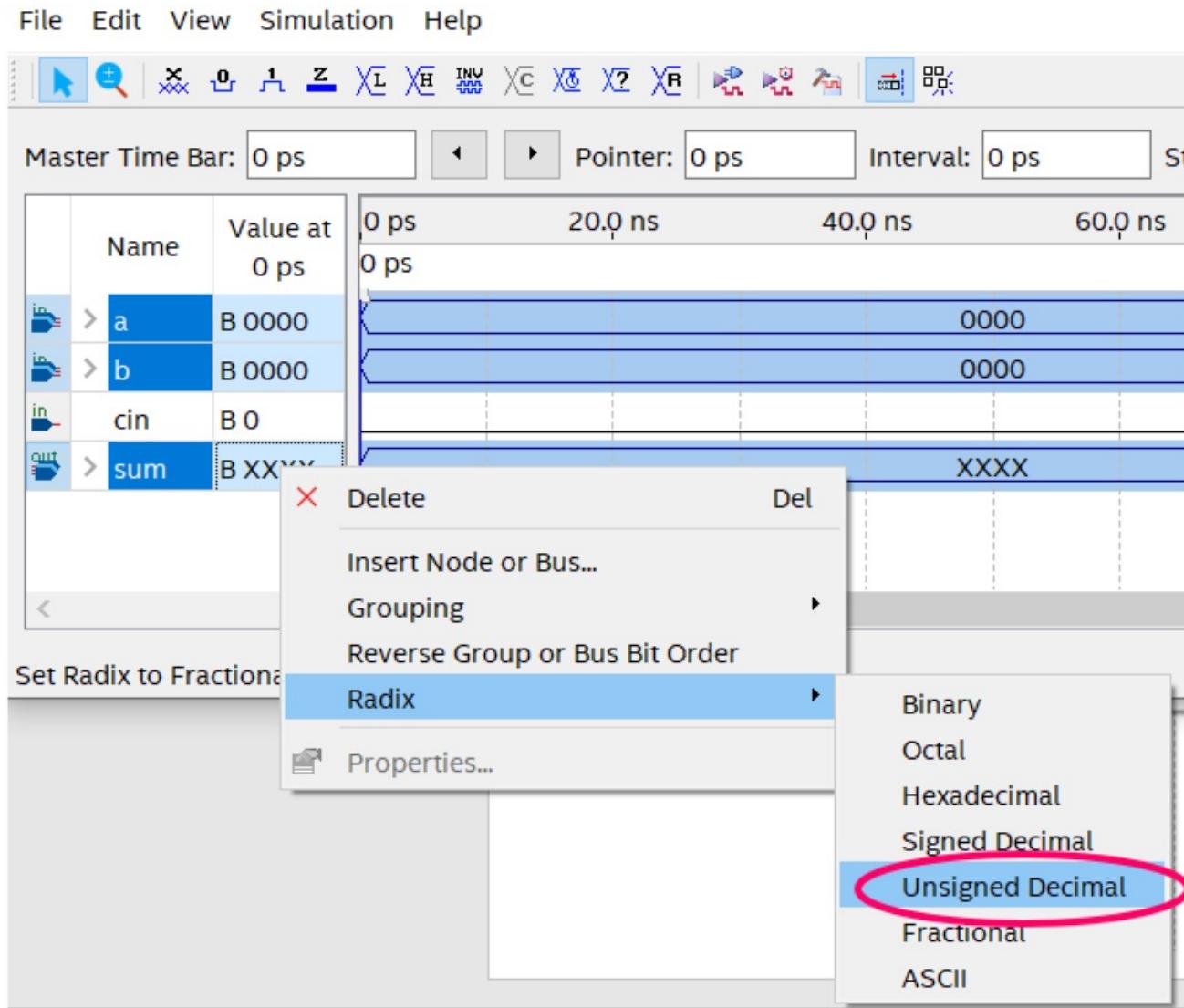
# Create Waveform File

File Edit View Simulation Help

Search altera.com

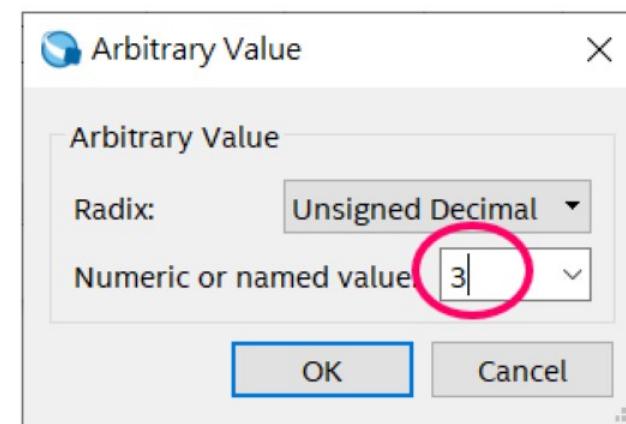
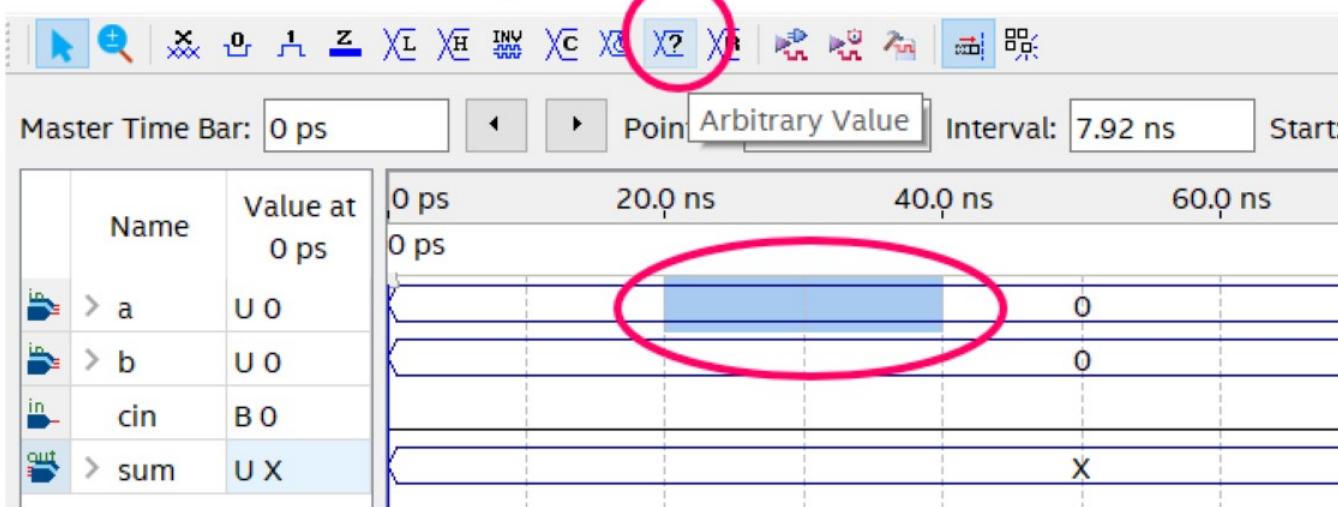


# Assign Node Properties

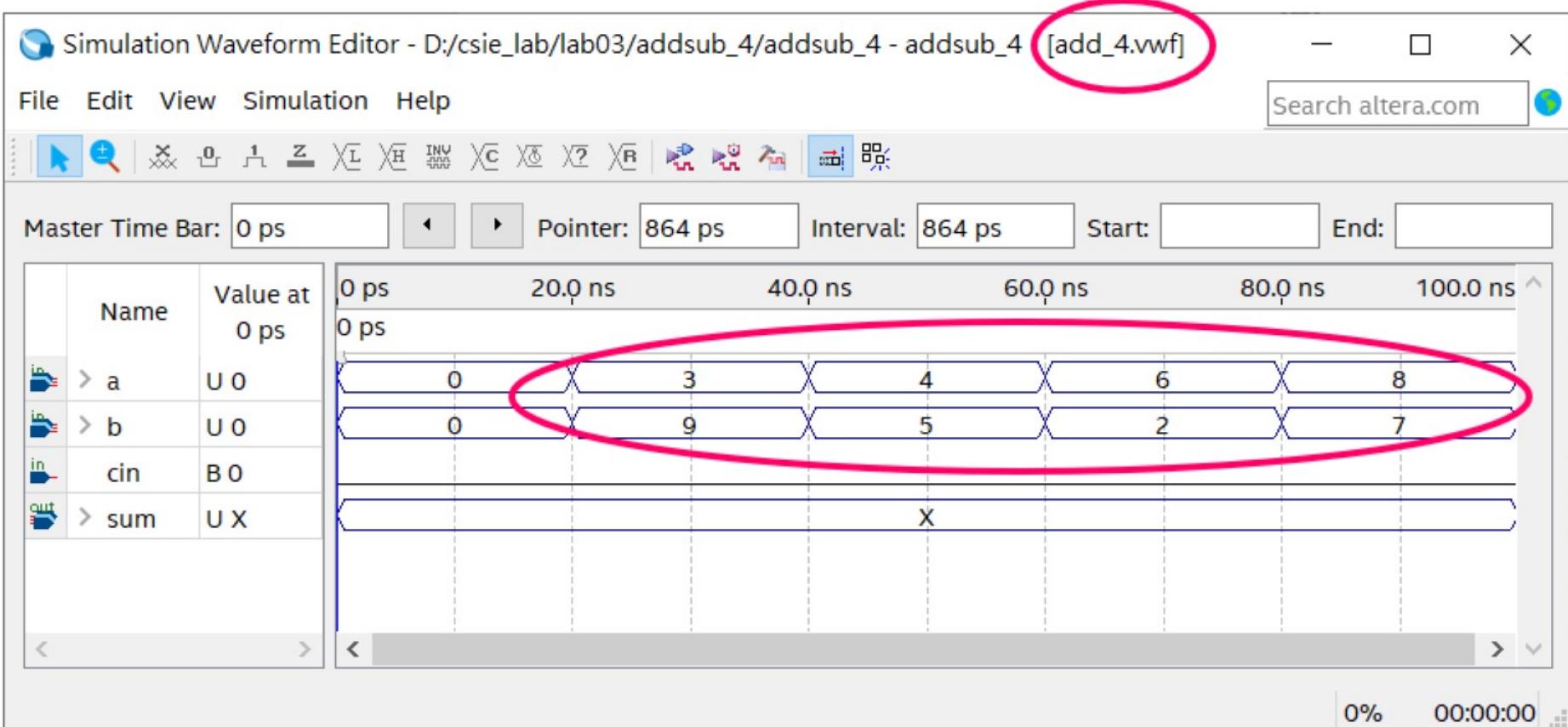


# Set Input Value (1/2)

File Edit View Simulation Help



## Set Input Value (2/2)



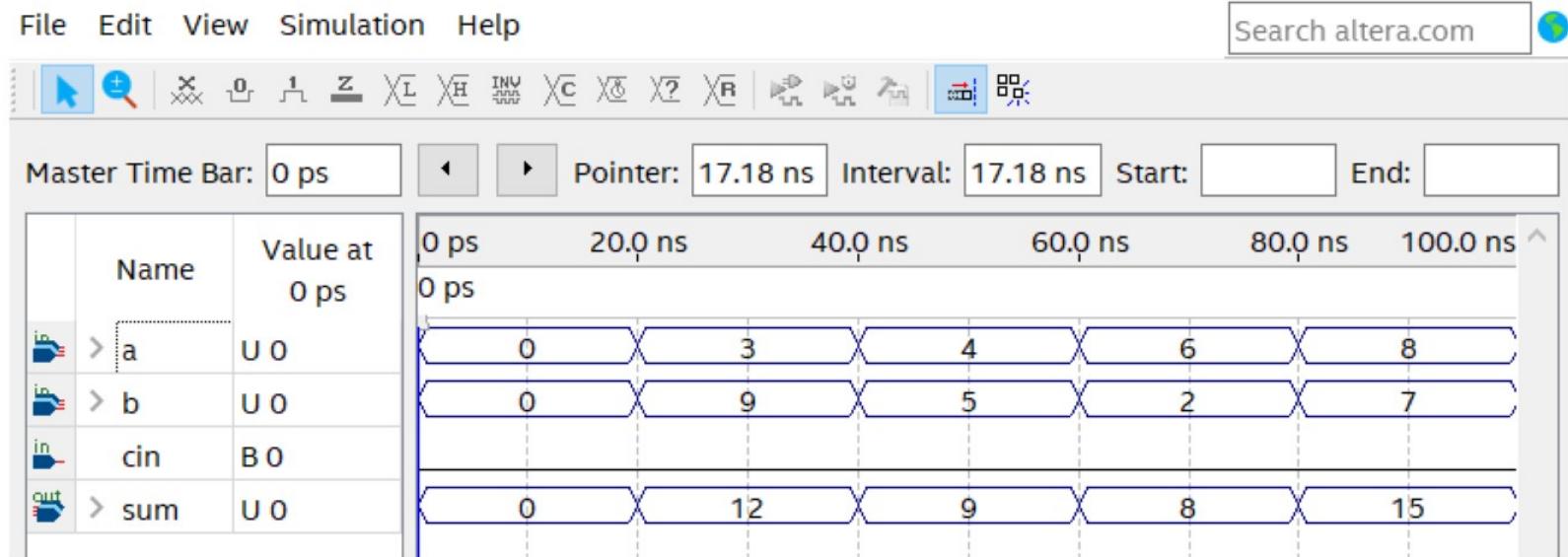
# Set ModelSim Script

Simulation Waveform Editor - D:/csie\_lab/lab03/addsub\_4/addsub\_4 - addsub\_4

The screenshot shows the ModelSim Simulation Waveform Editor interface. The 'Simulation' menu is highlighted with a red oval. The 'Simulation Settings' option is selected. The 'Master Time Bar' shows two signals: 'a' and 'b'. Signal 'a' has values 0, 3, and X. Signal 'b' has values 0, 9, and X. The 'Simulation Options' panel displays a warning: 'Caution: Improperly modifying these settings can c'. It shows the 'HDL Language' set to Verilog (radio button selected). Below are tabs for 'Functional Simulation Settings' and 'Timing Simulation S'. Under 'Testbench Generation Command (Functional Simulation)', the command is 'quartus\_eda --gen\_testbench --tool=modelsim\_oem --'. Under 'Netlist Generation Command (Functional Simulation)', the command is 'quartus\_eda --write\_settings\_files=off --simulation --ft'. Under 'ModelSim Script (Functional Simulation)', the script content is:

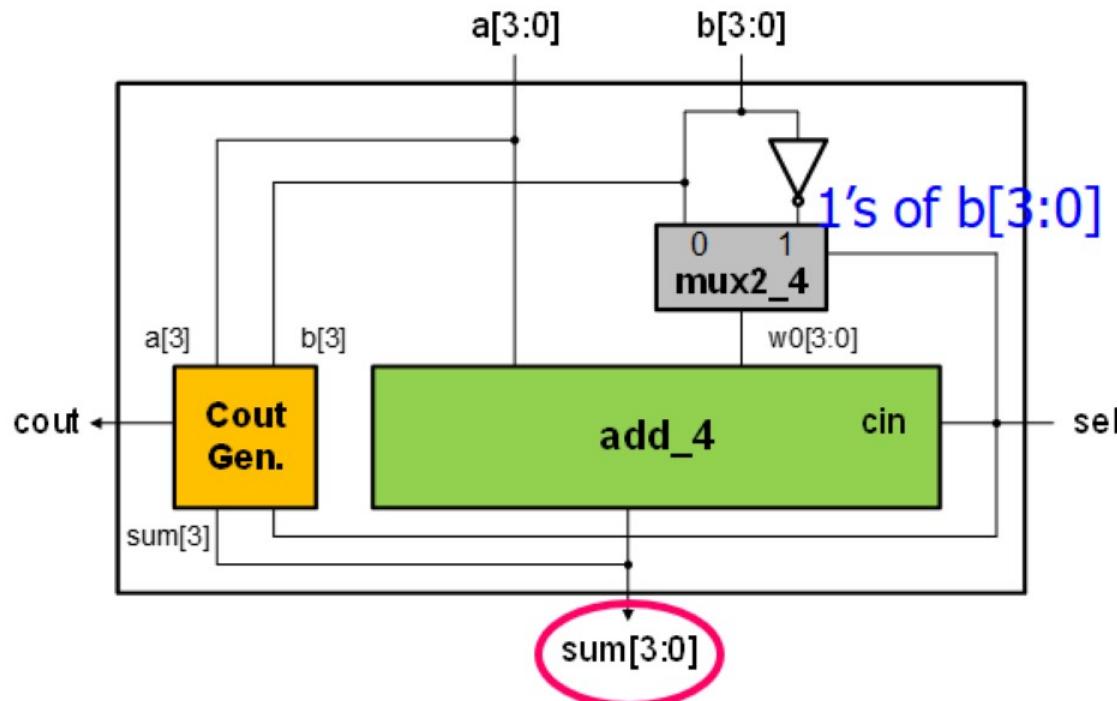
```
onerror {exit -code 1}
vlib work
vlog -work work addsub_4.vo
vlog -work work add_4.vwf.vt
vsim -c -t 1ps -L fiftyfivenm_ver -L altera_ver -L altera
```

# Run Functional Simulation



記錄實驗結果

# Design 4: 4-Bit Adder/Subtractor (addsub\_4.v) (1/5)

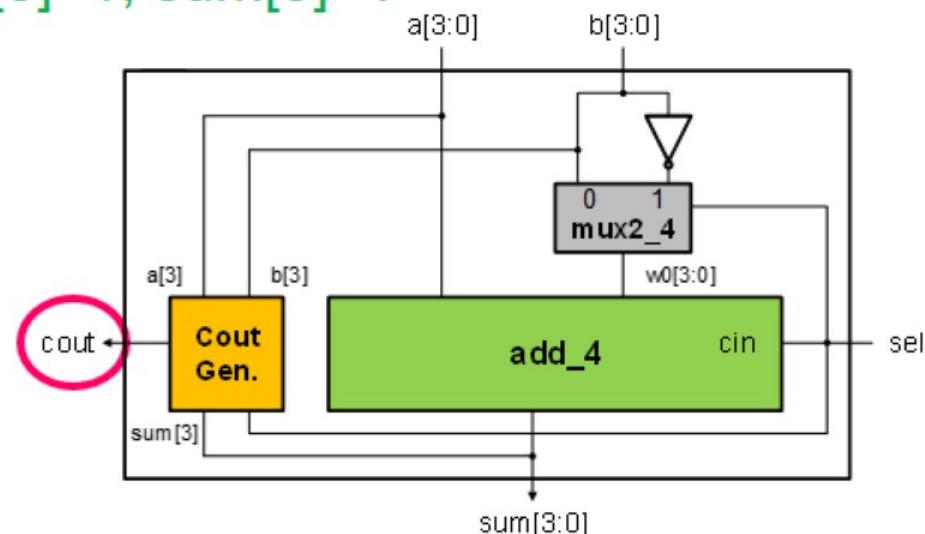


1. if  $sel = 0$ ,  $w0 = b$   
 $sum = a+b$

2. if  $sel = 1$ ,  $w0$  is 1's of  $b$   
 $sum = a-b$

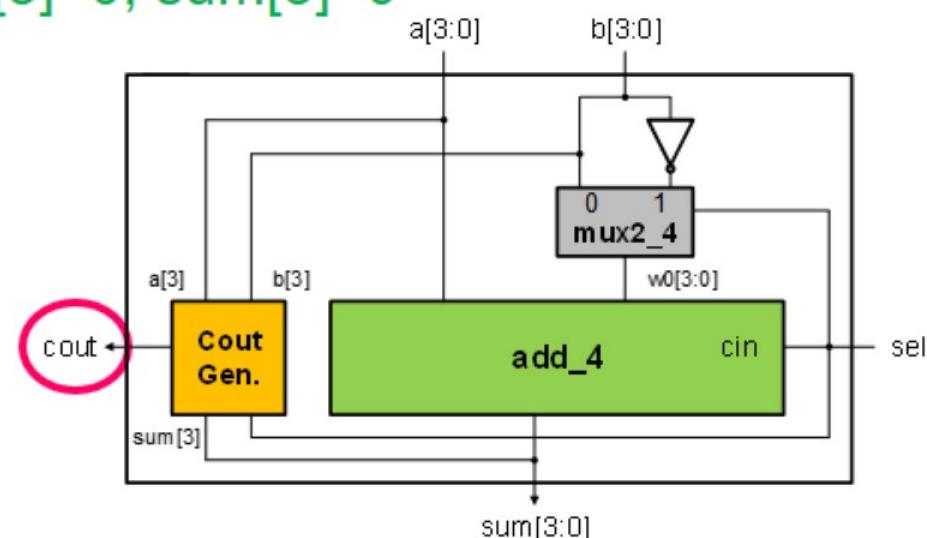
# Design 4: 4-Bit Adder/Subtractor (addsub\_4.v) (2/5)

- Overflow detection: 當sel, a[3], b[3], sum[3]之值滿足下列情況之一時，即發生overflow
- Overflow 1:  $A \geq 0, B \geq 0, Y = A + B, Y < 0$ 
  - sel=0 (add), a[3]=0, b[3]=0, sum[3]=1
- Overflow 2:  $A \geq 0, B < 0, Y = A - B, Y < 0$ 
  - sel=1 (sub), a[3]=0, b[3]=1, sum[3]=1



# Design 4: 4-Bit Adder/Subtractor (addsub\_4.v) (3/5)

- Underflow detection: 當sel, a[3], b[3], sum[3]之值滿足下列情況之一時，即發生underflow
- Underflow 1:  $A < 0, B < 0, Y = A + B, Y \geq 0$ 
  - sel=0 (add), a[3]=1, b[3]=1, sum[3]=0
- Underflow 2:  $A < 0, B \geq 0, Y = A - B, Y \geq 0$ 
  - sel=1 (sub), a[3]=1, b[3]=0, sum[3]=0

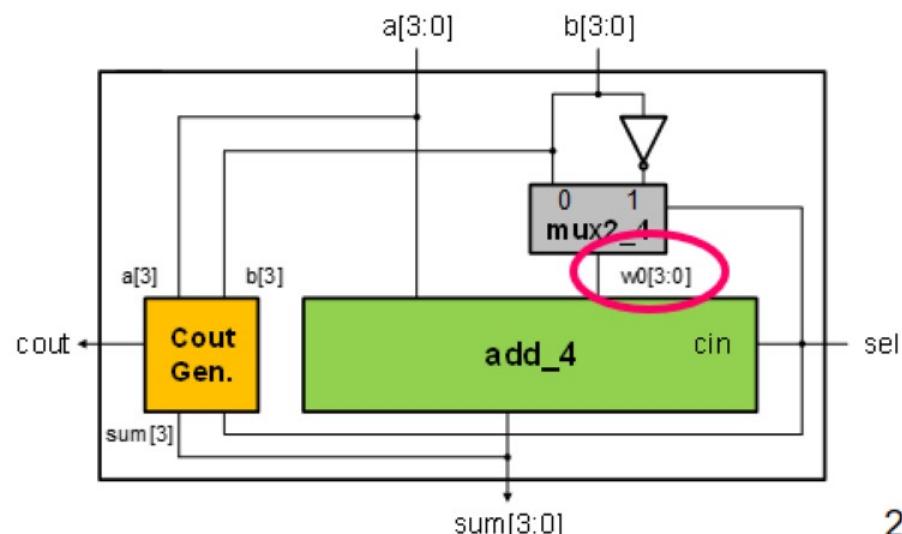


If underflow occurs, cout = 1

# Design 4: 4-Bit Adder/Subtractor (addsub\_4.v) (4/5)

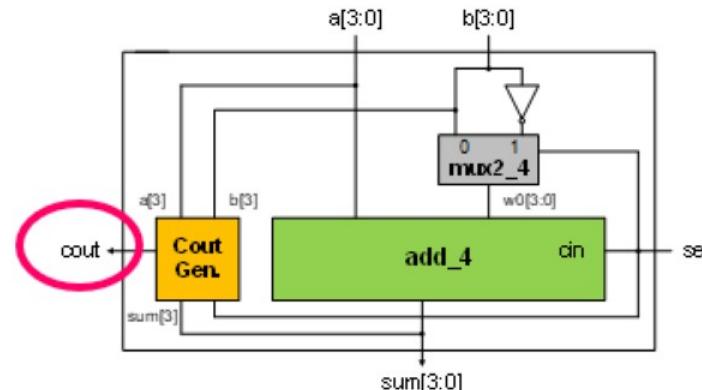
```
module addsub_4(a, b, sel, cout, sum);  
    input [3:0] a,b;  
    input sel;  
    output cout;  
    output [3:0] sum;  
  
    wire [3:0] w0; // ex. assign y = (x==1) ? a : b;  
  
    assign w0 =
```

add\_4 m0



# Design 4: 4-Bit Adder/Subtractor (addsub\_4.v) (5/5)

```
assign cout = (~sel & ~a[3] & ~b[3] & sum[3]) |  
           // overflow 1  
           // sel=0 (add), a[3]=0, b[3]=0, sum[3]=1  
           // overflow 2  
           // sel=1 (sub), a[3]=0, b[3]=1, sum[3]=1  
           // underflow 1  
           // sel=0 (add), a[3]=1, b[3]=1, sum[3]=0  
           // underflow 2  
           // sel=1 (sub), a[3]=1, b[3]=0, sum[3]=0  
endmodule
```



# Add “addsub\_4.v” to Project

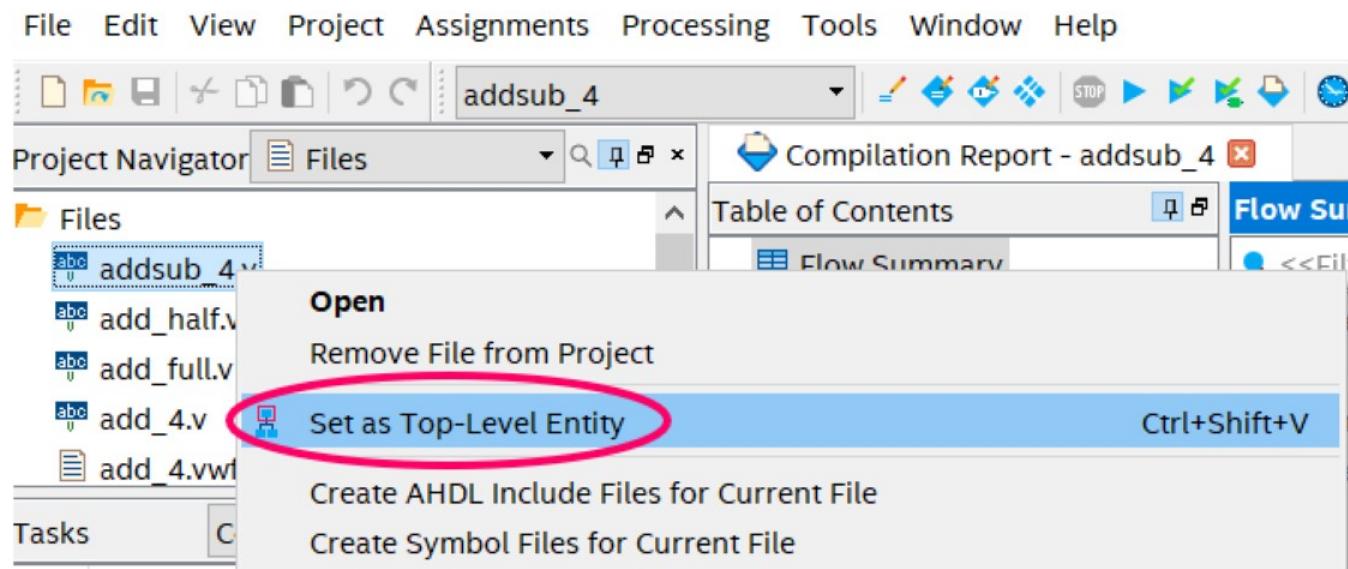
File Edit View Project Assignments Process

The screenshot shows a CAD software interface with the following elements:

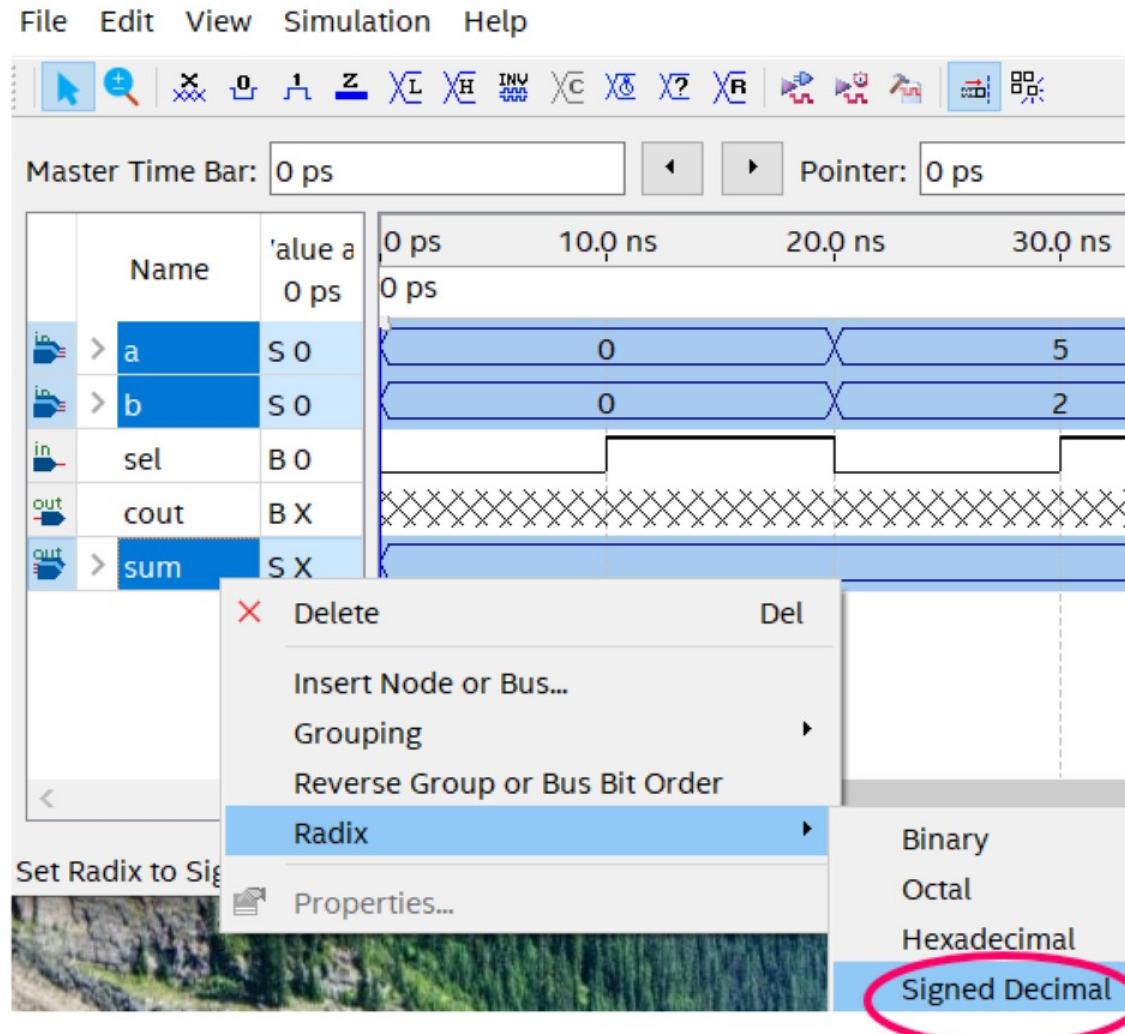
- Toolbar:** Includes icons for file operations like Open, Save, Print, and Undo/Redo.
- Title Bar:** Displays the project name "addsub\_4".
- Project Navigator:** Shows a tree view of files:
  - A folder icon labeled "File..."
  - ABC icons for "add\_full.v", "add\_4.v", and "add\_4.vwf".
- Files Tab:** Active tab, titled "Files". It contains a message: "Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project." Below this is a "File name:" input field and a search bar.
- Table:** A list of files with columns: File Name, Type, Library, and Design Entry/Synth. The "addsub\_4.v" file is highlighted with a red circle and a green arrow points to it from the bottom left.

File Name	Type	Library	Design Entry/Synth
addsub_4.v	Verilog HDL File	<None>	
add_half.v	Verilog HDL File	<None>	
add_full.v	Verilog HDL File	<None>	
add_4.v	Verilog HDL File	<None>	
add_4.vwf	University Program VWF	<None>	

# Set “addsub\_4.v” as Top-Level Entity



# Create Waveform File for Functional Simulation



# Simulation Result

Add

$$5+2=7$$

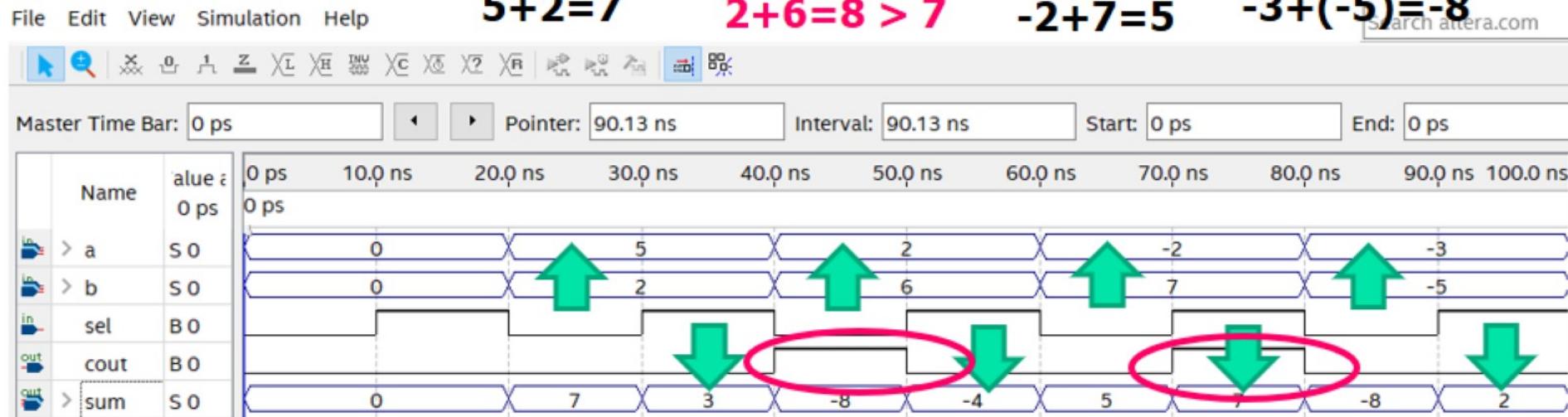
Overflow

$$2+6=8 > 7$$

$$-2+7=5$$

$$-3+(-5)=-8$$

Search altera.com



Subtract

$$5-2=3$$

$$2-6=-4$$

Underflow  
 $-2-7=-9 < -8$

$$-3-(-5)=2$$

記錄實驗結果

## Design 5: Top Module (addsub\_4\_top.v)

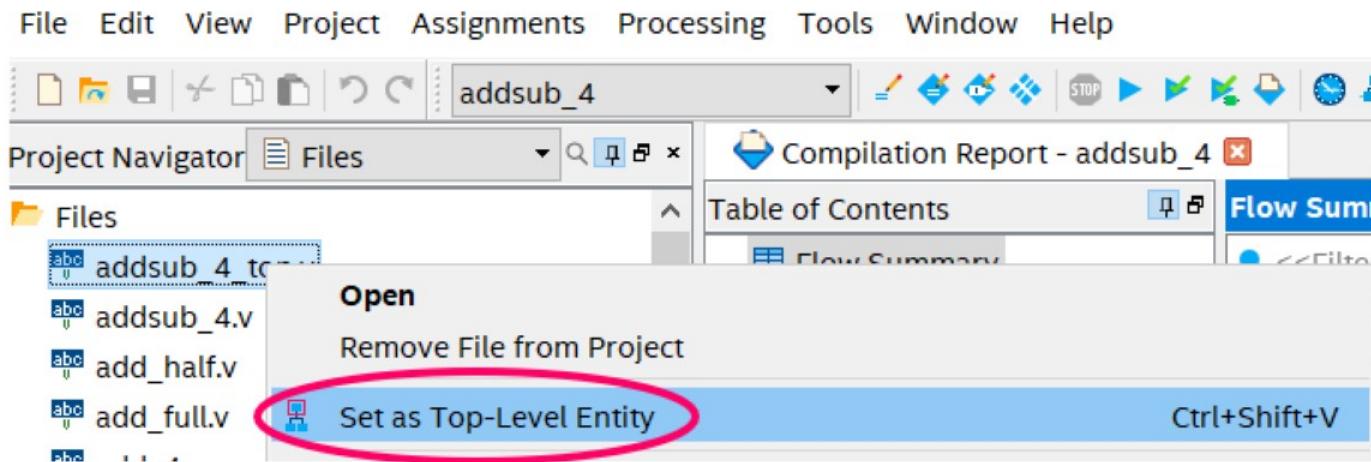
---

```
module addsub_4_top(a, b, sel, cout, sum);
    input [3:0] a,b;
    input sel;
    output cout;
    output [3:0] sum;

    addsub_4 m0
endmodule
```

# Add “addsub\_4\_top.v” to Project and Set as Top-Level Entity

---



# Pin Assignment (1/2)

**cout**

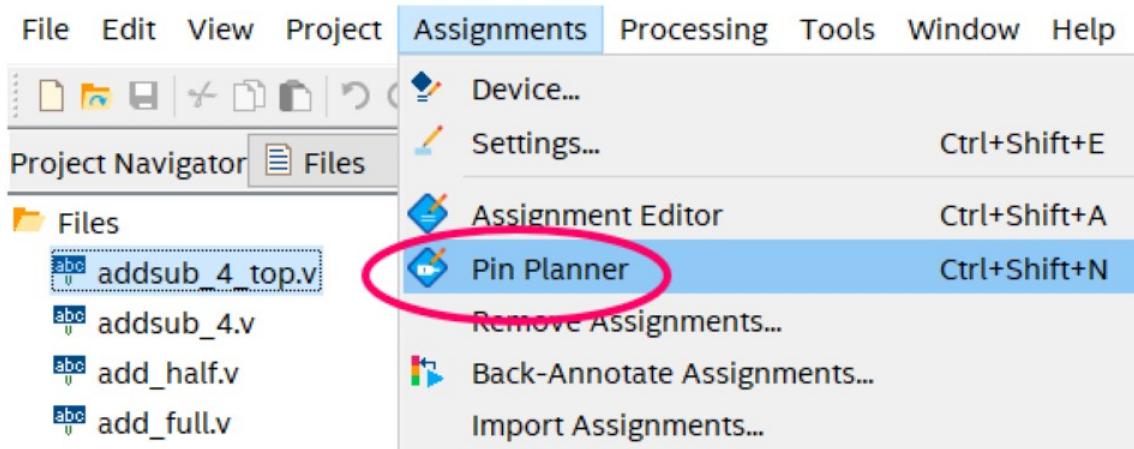


1: LED on  
0: LED off



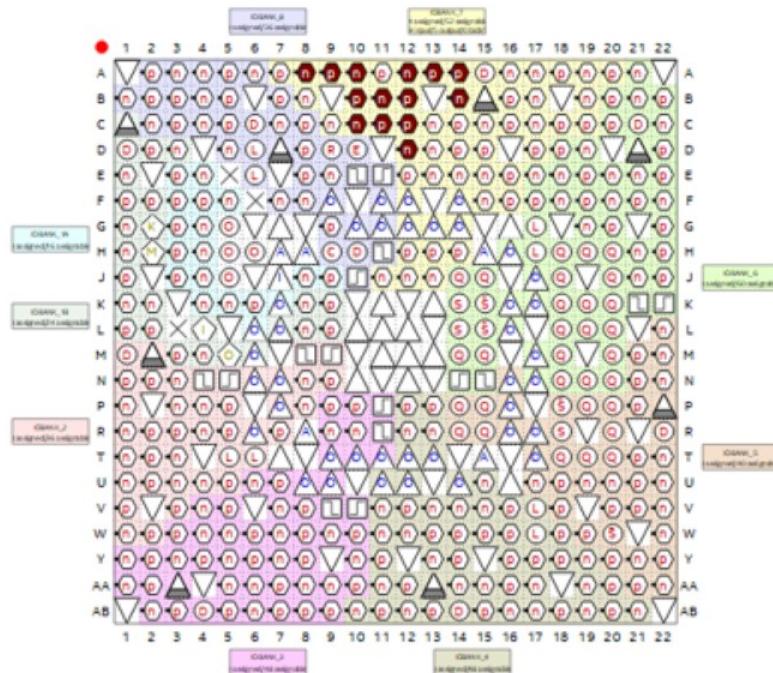
1  
0

**a[3:0]**    **b[3:0]**    **sel** 0: add, 1: subtract



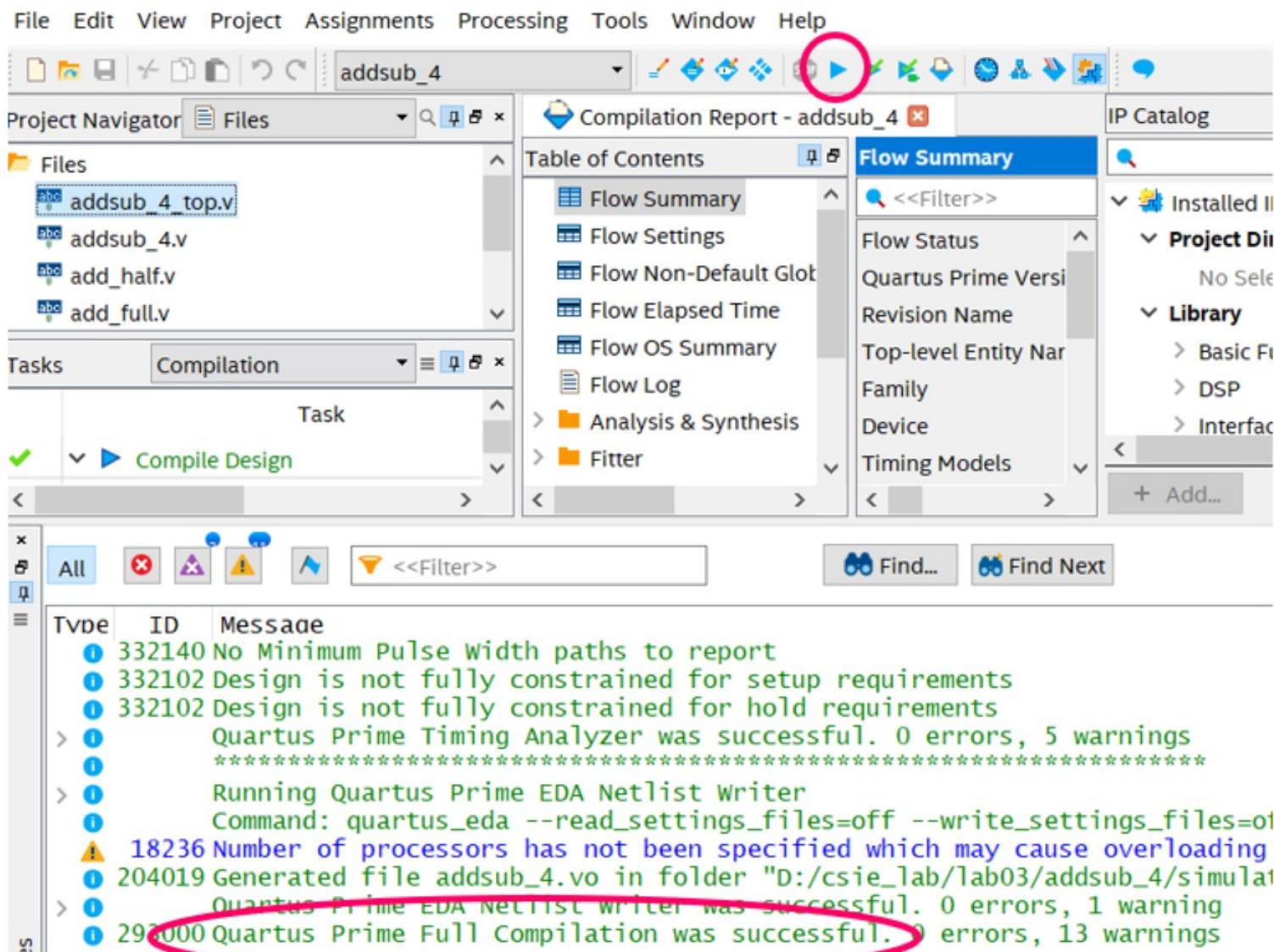
# Pin Assignment (2/2)

Top View - Wire Bond  
MAX 10 - 10M50DAF484C7G

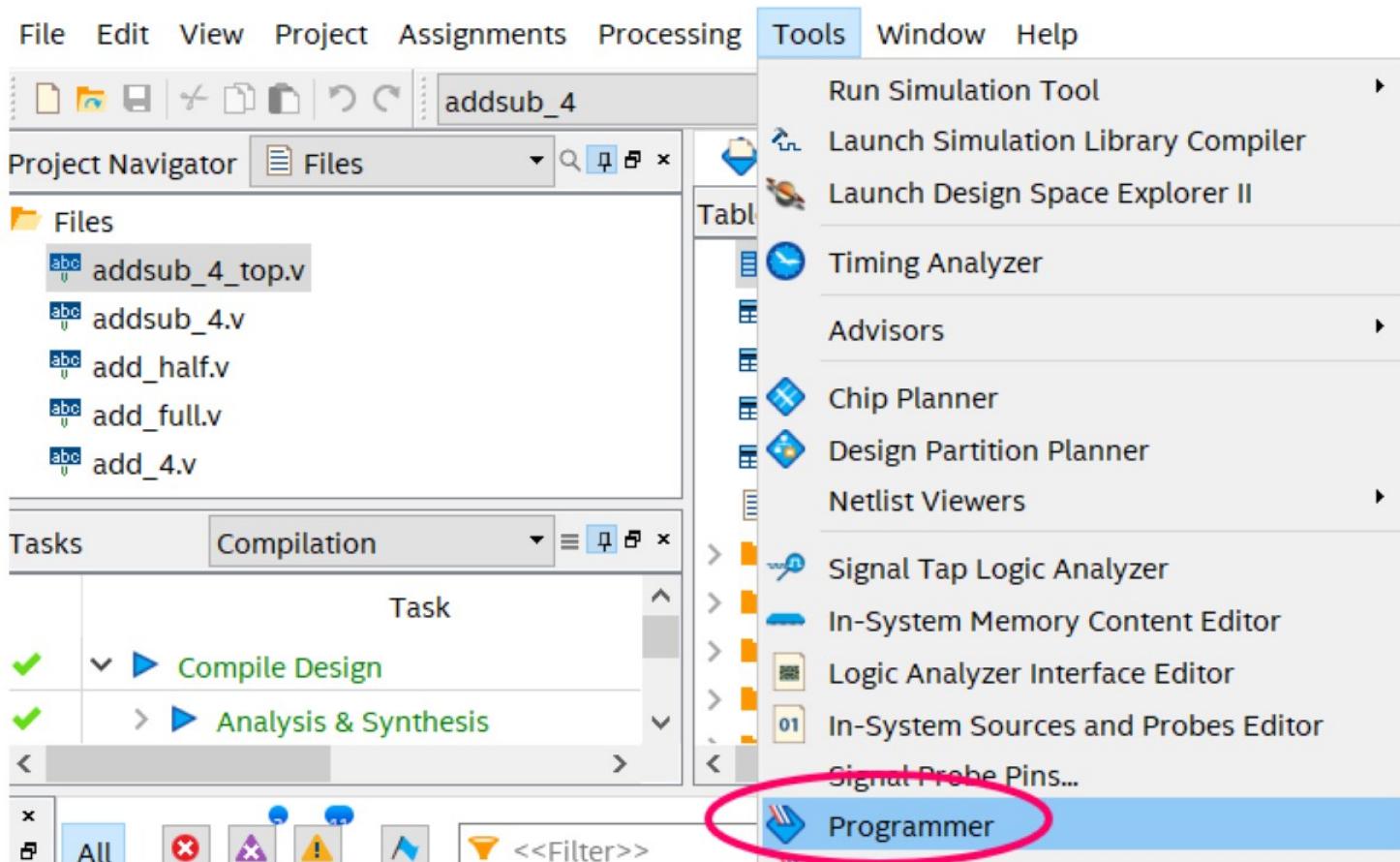


Node Name	Direction	Location
in a[3]	Input	PIN_B14
in a[2]	Input	PIN_A14
in a[1]	Input	PIN_A13
in a[0]	Input	PIN_B12
in b[3]	Input	PIN_A12
in b[2]	Input	PIN_C12
in b[1]	Input	PIN_D12
in b[0]	Input	PIN_C11
out cout	Output	PIN_B11
in sel	Input	PIN_C10
out sum[3]	Output	PIN_B10
out sum[2]	Output	PIN_A10
out sum[1]	Output	PIN_A9
out sum[0]	Output	PIN_A8

# Start Compilation



# Start Programming (1/2)



# Start Programming (2/2)

File Edit View Processing Tools Window Help

Search altera.com

Hardware Setup... **USB-Blaster [USB-0]** Mode: JTAG Progress: % (Success)

Enable real-time ISP to allow background programming when available

**Start** (circled in red)

Stop

Auto Detect

Delete

Add File...

Change File

Save File

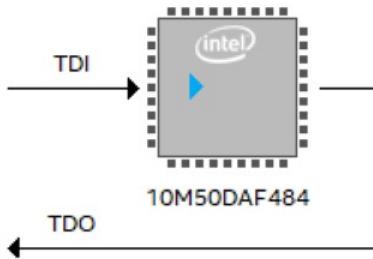
Add Device

Up

Down

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
output_files/adds...	10M50DAF484	002711B3	002711B3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

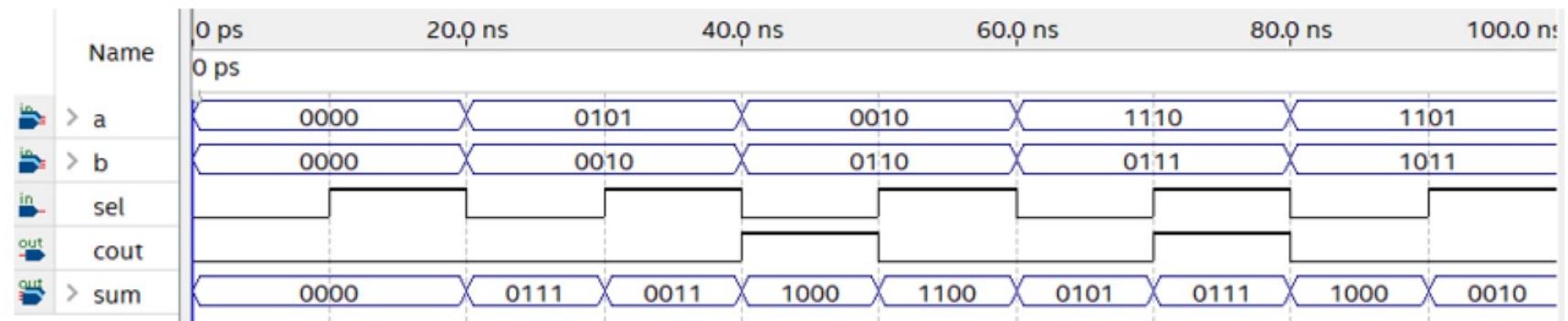
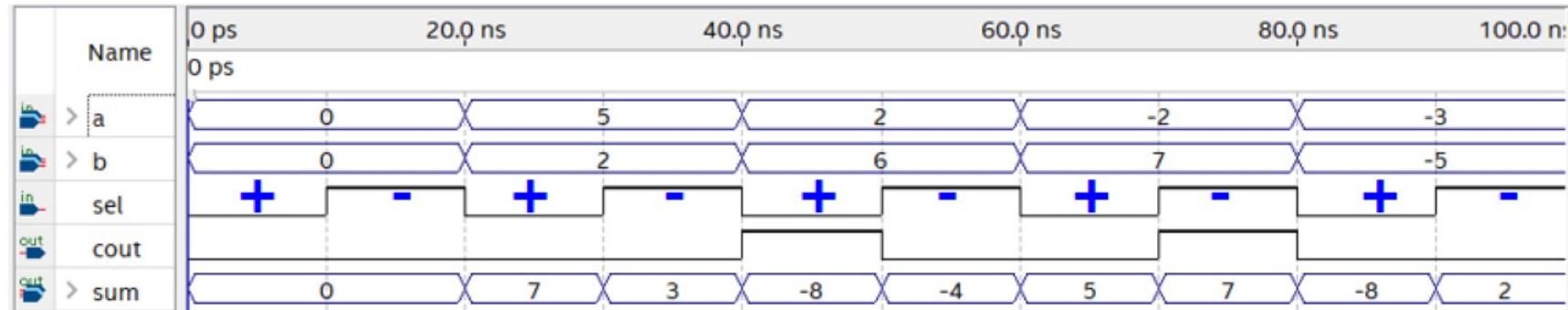
< >



The diagram shows a square Intel chip with a dashed border. An arrow labeled "TDI" points to the top-left pin, and another arrow labeled "TDO" points away from the bottom-left pin.

10M50DAF484

# 實驗板驗證



cout

sum[3:0]



1  
0

1: LED on  
0: LED off

a[3:0]      b[3:0]      Sel  
(0: add, 1: subtract)

# File List

---

## ■ Verilog files

- add\_half.v
- add\_full.v
- add\_4.v
- addsub\_4.v
- addsub\_4\_top.v

## ■ Waveform files

- add\_4.vwf
- addsub\_4.vwf

# 實驗結果驗收

---

- 請老師或助教驗收
  - 1. **add\_4** 波形圖
  - 2. **addsub\_4** 波形圖
  - 3. **addsub\_4\_top** 電路於實驗板之行為