# *LAB* - 05

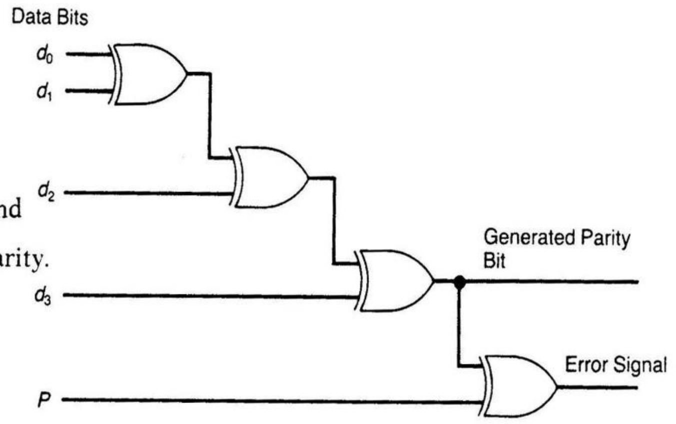
陳培殷

國立成功大學　資訊工程系

## Outline

- Video preview for HDL介紹(Parts I~IV)
- Combinational circuit
- Lab I-- 4-Bit Adder Subtractor
- Lab II-- 8 to 3 encoder
- Appendix

## Combinational Circuit



A 4-bit parity generation and checking circuit for even parity.

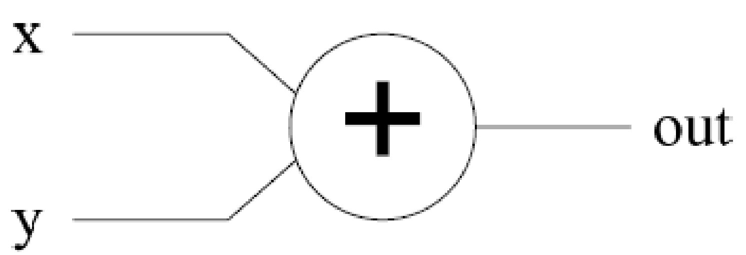
- 組合電路是一種邏輯電路，它任一時刻的輸出值，僅與目前的(present)輸入值有關，而與先前的(previous)輸入值無關
- 在電路結構上，組合電路由各種邏輯閘組成，電路中無記憶元件與反饋線
- 與之相對的則是循序邏輯電路，循序邏輯電路的輸出結果與目前的輸入及先前的輸入都有關係

## Continuous assignment

**The first way to define a combinational circuit.**

```
module  half_adder (x, y, out);

// port declaration
input              x, y;
output  [1:0]  out;

// functionality or structure
assign  out = x + y;

endmodule
```

```
s = x ^ y;
c = x & y;
```

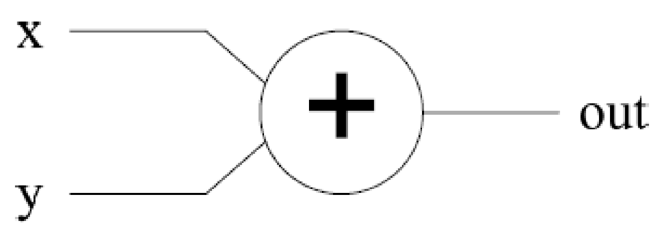


## always block

The second way to define a combinational circuit.

```
module  half_adder (x, y, out);

// port declaration
input              x, y;
output  [1:0]  out;

// I/O type
reg        [1:0]  out;

// functionality or structure
always @ (x or y)
begin
   out = x + y;
end
```

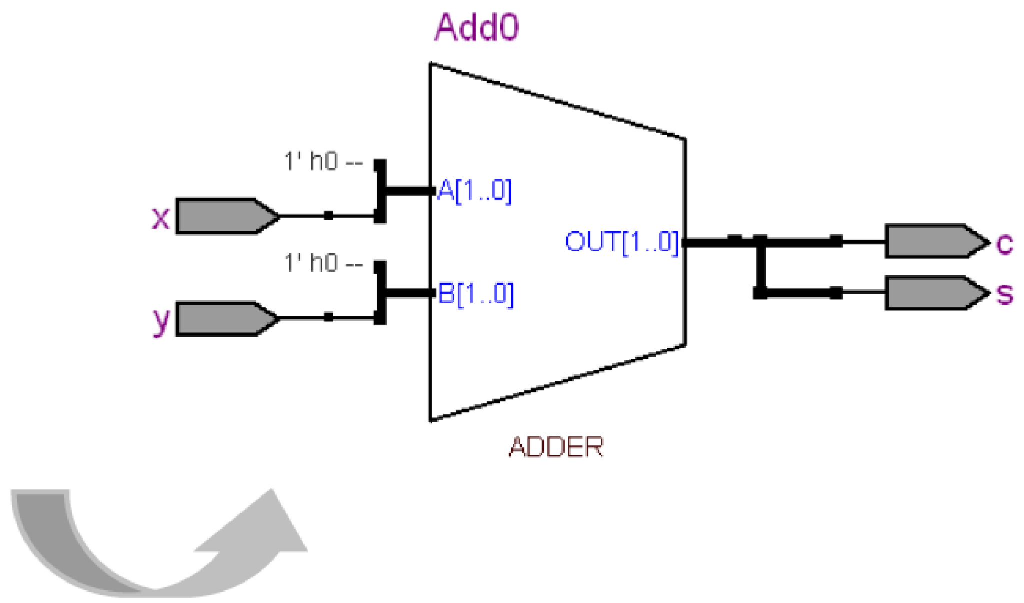
Sensitivity list



# Example: half adder (1/3)

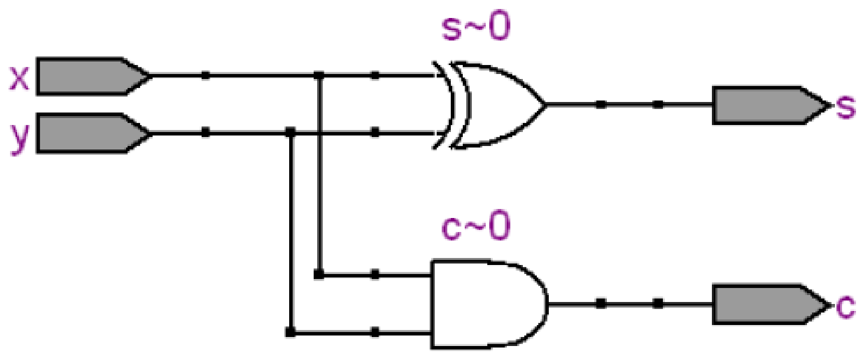
- Data flow description

```
module  half_adder (x, y, c, s);

// port declaration
input     x, y;
output  c, s;

// functionality or structure
assign {c, s} = x + y;

endmodule
```



# Example: half adder (2/3)

```
module  half_adder (x, y, c, s);
// port declaration
input     x, y;
output  c, s;

// I/O type
reg        c, s;

// functionality or structure
always @ (x or y)
begin
   s = x ^ y;
   c = x & y;
end
endmodule
```

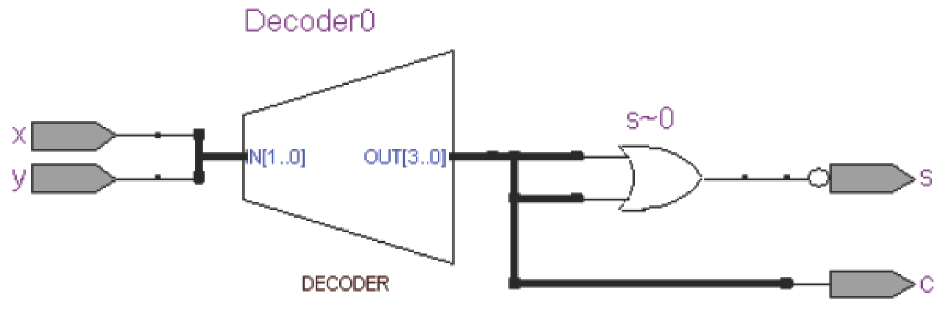


# Example: half adder (3/3)

```
// functionality or structure
always @ (x or y)
begin
   case ({x, y})
      2'b00: begin
         s = 0;
         c = 0;
      end
      2'b01: begin
         s = 1;
         c = 0;
      end
      2'b10: begin
         s = 1;
         c = 0;
      end
      2'b11: begin
         s = 0;
         c = 1;
      end
   endcase
end  //end always block
```

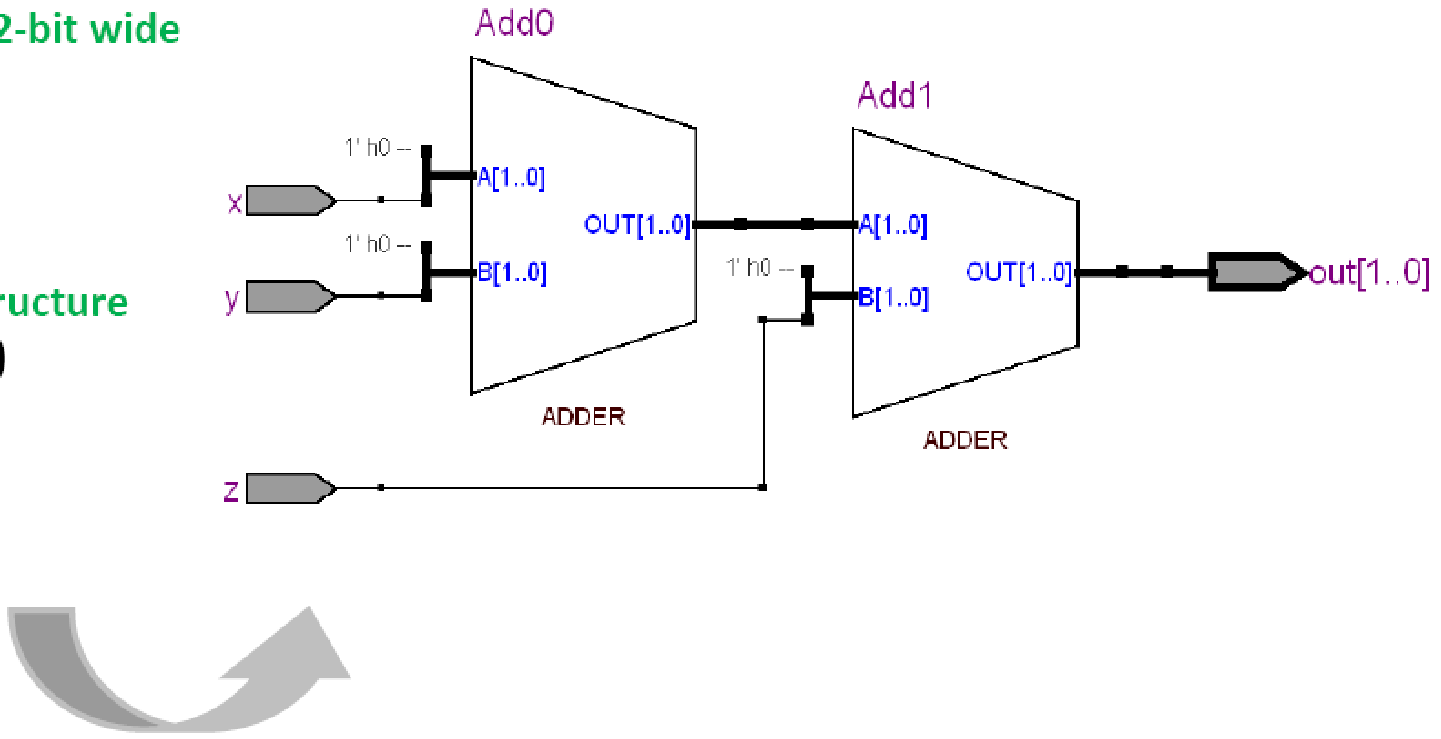
Truth table

| x | y | C | S |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |



## Example: full adder (1/2)

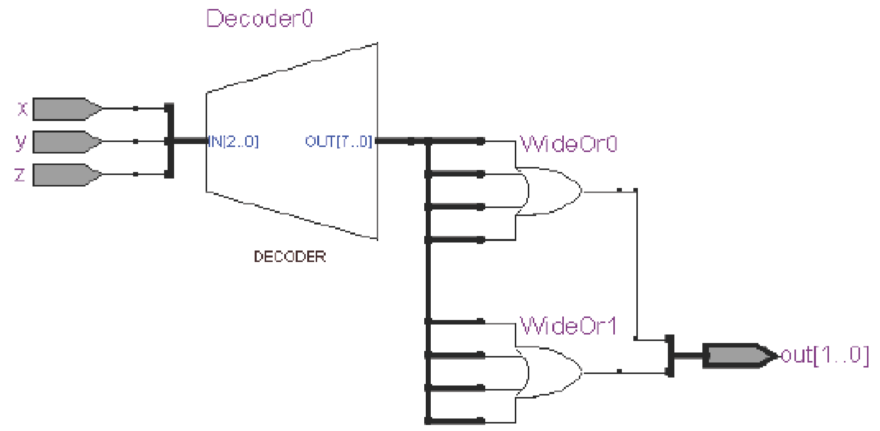
```
module  full_adder (x, y, z, out);
// port declaration
input             x, y, z;
output  [1:0] out; // 2-bit wide

// I/O type
reg        [1:0] out;

// functionality or structure
always @ (x or y or z)
begin
   out = x + y + z;
end
endmodule
```



## Example: full adder (2/2)

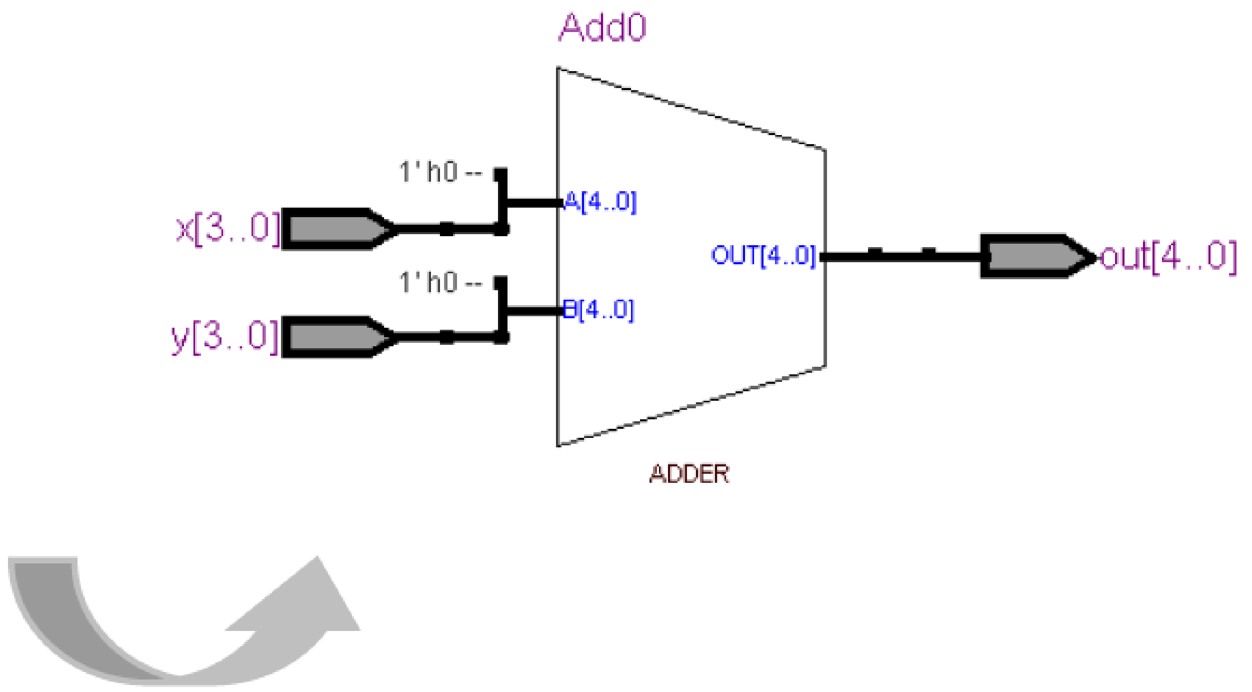
```
module  full_adder (x, y, z, out);
// port declaration
input            x, y, z;
output  [1:0] out; // 2-bit wide

// I/O type
reg        [1:0] out;

// functionality or structure
always @ (x or y or z)
begin
   case ({x, y, z})
      3'b000: out = 2'b00;
      3'b001: out = 2'b01;
      3'b010: out = 2'b01;
      3'b011: out = 2'b10;
      3'b100: out = 2'b01;
      3'b101: out = 2'b10;
      3'b110: out = 2'b10;
      3'b111: out = 2'b11;
   endcase
end
endmodule
```

| x | y | z | C | S |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |



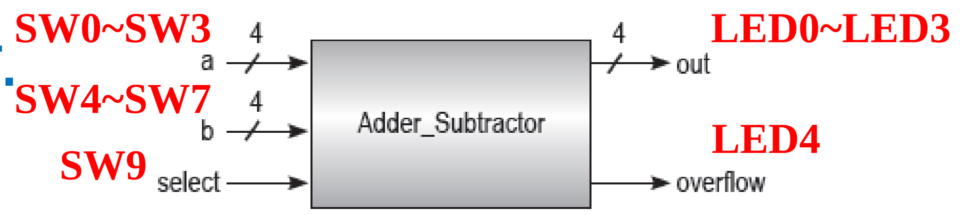
## Example: 4-bit adder

```
module  four_bit_adder (x, y, out);
// port declaration
input     [3:0]  x, y;
output  [4:0] out; // 5-bit wide

// I/O type
reg        [4:0] out;

// functionality or structure
always @ (x or y)
begin
   out = x + y;
end
endmodule
```



# Lab I

- Please design a 4-Bit Adder Subtractor.



無號數加減法器藉由選擇 (select) 訊號決定目前是做加法或減法運算：選擇線為 1 時做加法，選擇線為 0 時做減法，而溢位 (overflow) 訊號則是用來判斷加法有無進位或減法有無借位。

Reference: 1-bit Adder Subtractor

| 輸入 (input) | | | 輸出 (output) | |
|---|---|---|---|---|
| 被加減數 (a) | 加減數 (b) | 選擇 (select) | 和 / 差 (out) | 溢位 (overflow) |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 |

# Notice for Lab I

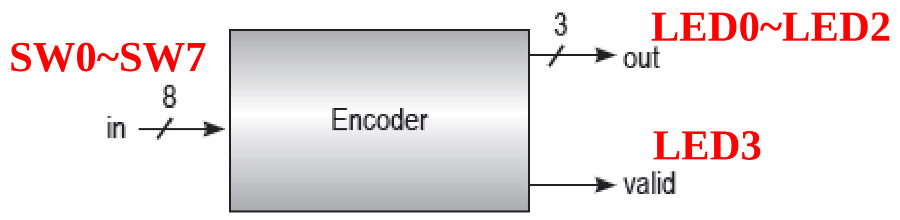
- 加減法器裝置模擬
  - select=1
    - 加法
    - Ex: 9+10=19 (overflow)
    - 9  =1001
    - 10=1010
    - 19(overflow) = 10011
  - select=0
    - 減法
    - Ex: 4-7= 3 (overflow)
    - 4=0100
    - 7=0111
    - 3 (overflow) = 10011

# Lab II

- Please design an 8 to 3 encoder

編碼器可以將 $2^n$ 個輸入訊號轉換成 n 位元輸出訊號。假設有 m 個輸入與 n 個輸出，則稱為 m 對 n 編碼器。

| 輸入 (input) | | | | | | | | 輸出 (output) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| in[7] | in[6] | in[5] | in[4] | in[3] | in[2] | in[1] | in[0] | valid | out[2] | out[1] | out[0] |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 其餘的輸入情況 | | | | | | | | 0 | 0 | 0 | 0 |



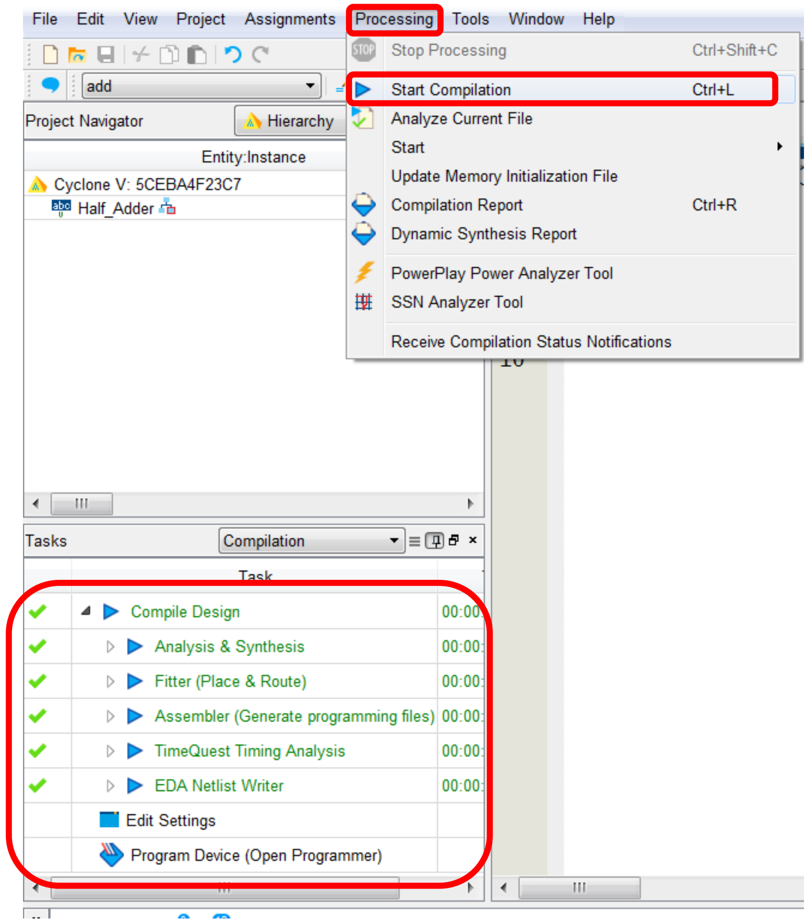
## Notice

- 請勿命名中文資料夾
- Device family 請確認與 FPGA Chip 符合 (5CEBA4F23C7)
- Top module name & Project name 需要一致
- 確認 module … endmodule 為keyword 變成藍色字體

## *Appendix*

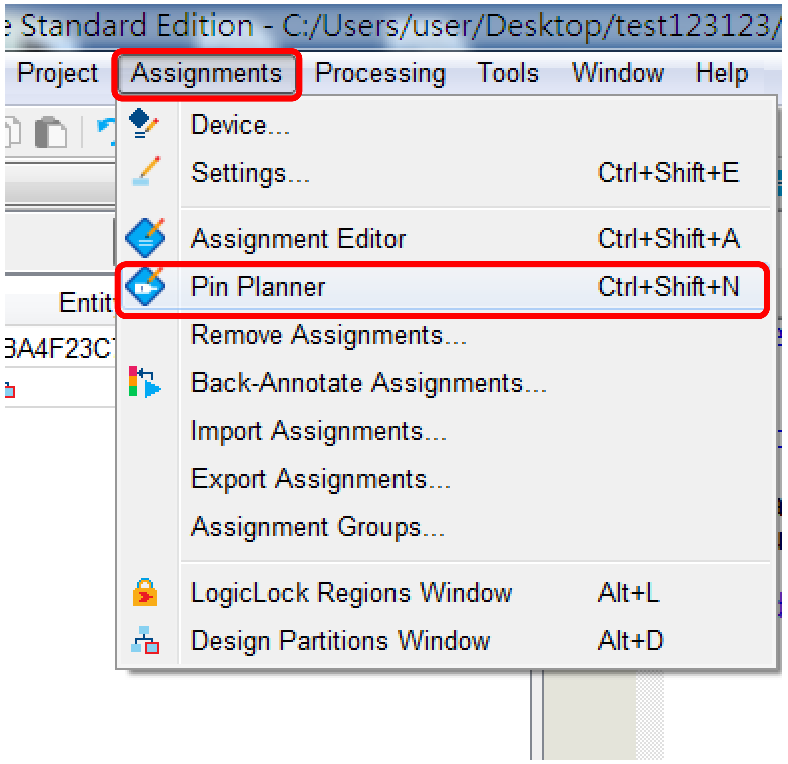
# Programming DE0-CV (1/12)

- Start compilation


File
Edit
View
Project
Assignments
Processing
Tools
Window
Help
add
Project Navigator
Hierarchy
Entity:Instance
Cyclone V: 5CEBA4F23C7
Half_Adder
Stop Processing Ctrl+Shift+C
Start Compilation Ctrl+L
Analyze Current File
Start
Update Memory Initialization File
Compilation Report Ctrl+R
Dynamic Synthesis Report
PowerPlay Power Analyzer Tool
SSN Analyzer Tool
Receive Compilation Status Notifications
Tasks
Compilation
Task
Compile Design 00:00
Analysis & Synthesis 00:00
Fitter (Place & Route) 00:00
Assembler (Generate programming files) 00:00
TimeQuest Timing Analysis 00:00
EDA Netlist Writer 00:00
Edit Settings
Program Device (Open Programmer)

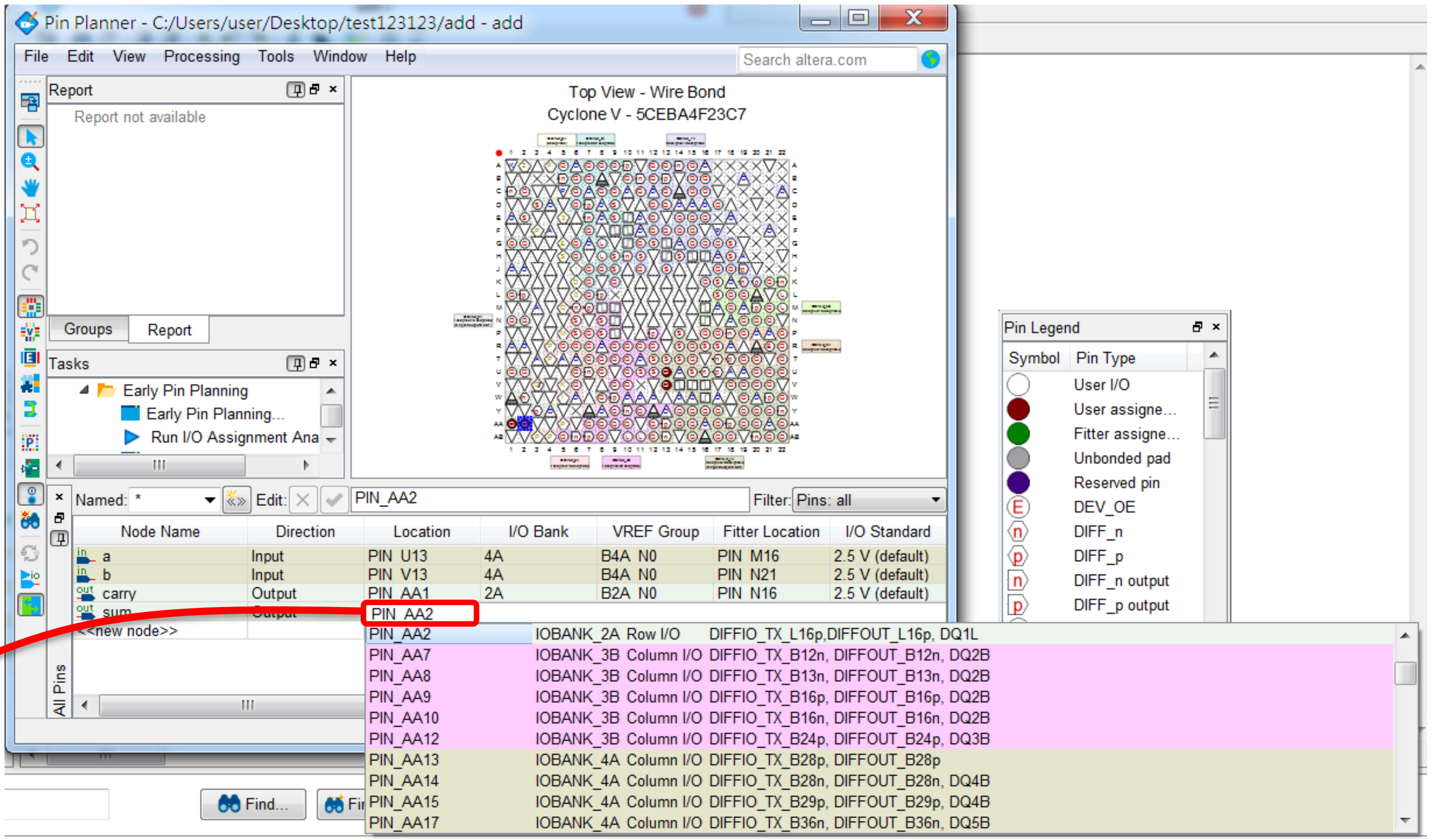
# Programming DE0-CV (2/12)

- Open Pin Planner



# Programming DE0-CV (3/12)

- Pin assignment



| Symbol | Pin Type |
|---|---|
| ○ | User I/O |
| ● | User assigne... |
| 🟢 | Fitter assigne... |
| ● | Unbonded pad |
| ● | Reserved pin |
| Ⓔ | DEV_OE |
| n | DIFF_n |
| p | DIFF_p |
| n | DIFF_n output |
| p | DIFF_p output |

Named: * Edit: PIN_AA2 Filter: Pins: all

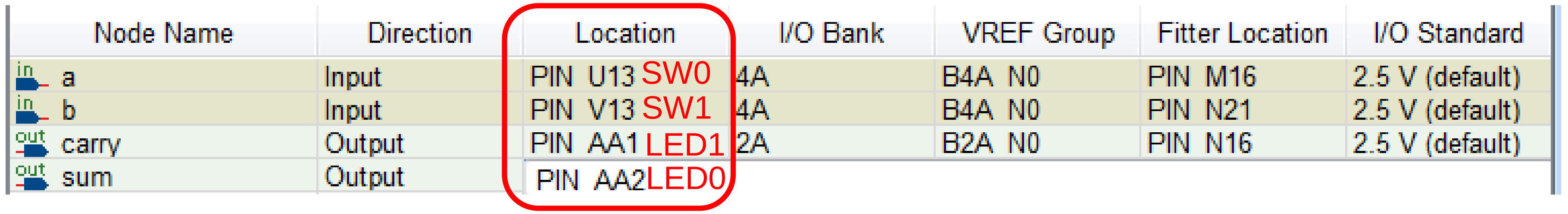
| Node Name | Direction | Location | I/O Bank | VREF Group | Fitter Location | I/O Standard |
|---|---|---|---|---|---|---|
| a | Input | PIN_U13 | 4A | B4A_N0 | PIN_M16 | 2.5 V (default) |
| b | Input | PIN_V13 | 4A | B4A_N0 | PIN_N21 | 2.5 V (default) |
| carry | Output | PIN_AA1 | 2A | B2A_N0 | PIN_N16 | 2.5 V (default) |
| sum | Output | PIN_AA2 | | | | |
| <<new node>> | | | | | | |

| | | |
|---|---|---|
| PIN_AA2 | IOBANK_2A Row I/O | DIFFIO_TX_L16p,DIFFOUT_L16p, DQ1L |
| PIN_AA7 | IOBANK_3B Column I/O | DIFFIO_TX_B12n, DIFFOUT_B12n, DQ2B |
| PIN_AA8 | IOBANK_3B Column I/O | DIFFIO_TX_B13n, DIFFOUT_B13n, DQ2B |
| PIN_AA9 | IOBANK_3B Column I/O | DIFFIO_TX_B16p, DIFFOUT_B16p, DQ2B |
| PIN_AA10 | IOBANK_3B Column I/O | DIFFIO_TX_B16n, DIFFOUT_B16n, DQ2B |
| PIN_AA12 | IOBANK_3B Column I/O | DIFFIO_TX_B24p, DIFFOUT_B24p, DQ3B |
| PIN_AA13 | IOBANK_4A Column I/O | DIFFIO_TX_B28p, DIFFOUT_B28p |
| PIN_AA14 | IOBANK_4A Column I/O | DIFFIO_TX_B28n, DIFFOUT_B28n, DQ4B |
| PIN_AA15 | IOBANK_4A Column I/O | DIFFIO_TX_B29p, DIFFOUT_B29p, DQ4B |
| PIN_AA17 | IOBANK_4A Column I/O | DIFFIO_TX_B36n, DIFFOUT_B36n, DQ5B |

Double click

# Programming DE0-CV (4/12)

- Assign pin location to all inputs and outputs

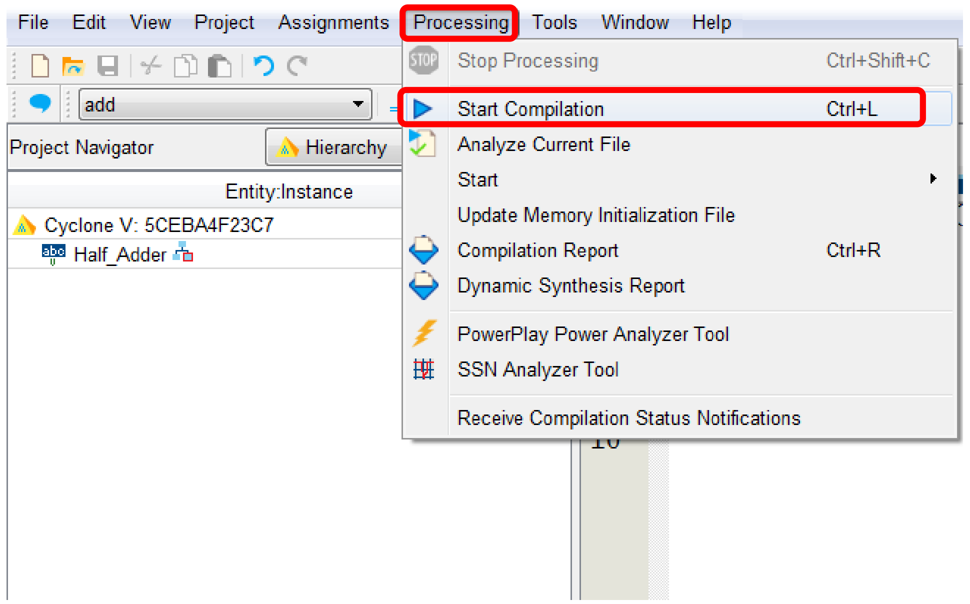


| Node Name | Direction | Location | I/O Bank | VREF Group | Fitter Location | I/O Standard |
|---|---|---|---|---|---|---|
| a | Input | PIN_U13 SW0 | 4A | B4A_N0 | PIN_M16 | 2.5 V (default) |
| b | Input | PIN_V13 SW1 | 4A | B4A_N0 | PIN_N21 | 2.5 V (default) |
| carry | Output | PIN_AA1 LED1 | 2A | B2A_N0 | PIN_N16 | 2.5 V (default) |
| sum | Output | PIN_AA2 LED0 | | | | |

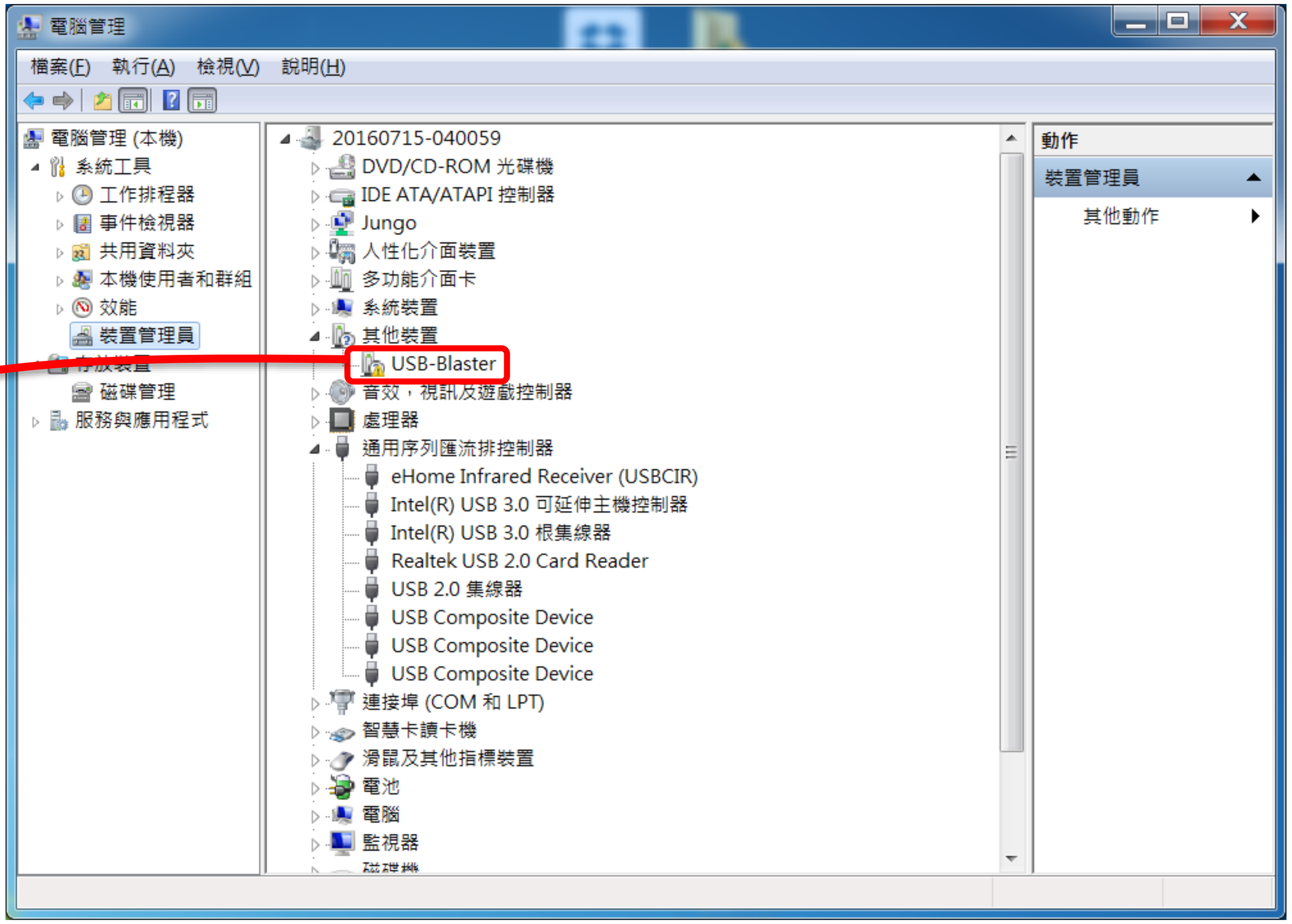
  - Please refer to DE0_pin.xls for pin location assignment

# Programming DE0-CV (5/12)

- Start compilation



# Programming DE0-CV (6/12)



電腦管理

檔案(F) 執行(A) 檢視(V) 說明(H)

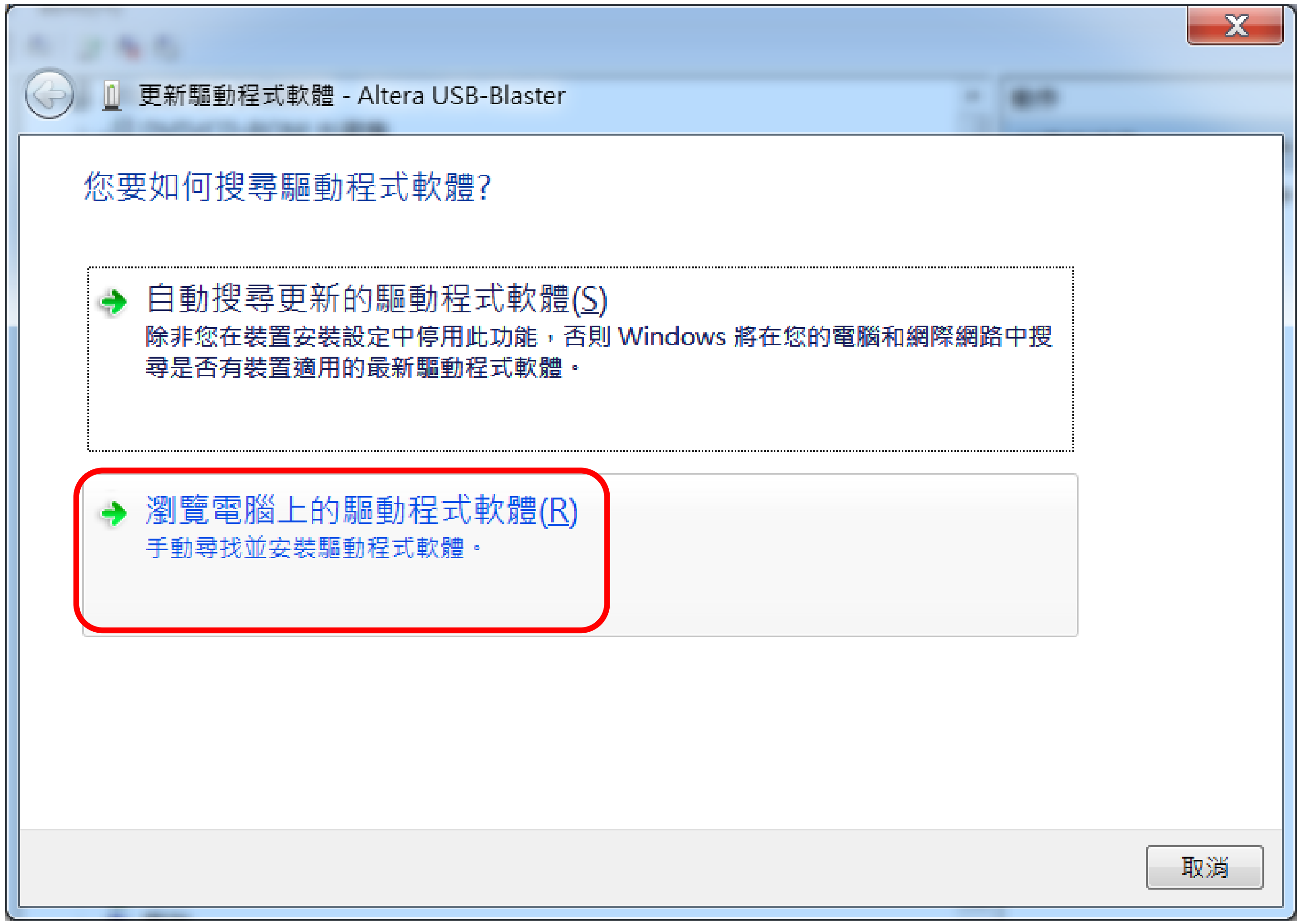
電腦管理 (本機)
- 系統工具
  - 工作排程器
  - 事件檢視器
  - 共用資料夾
  - 本機使用者和群組
  - 效能
  - 裝置管理員
- 存放裝置
  - 磁碟管理
- 服務與應用程式

20160715-040059
- DVD/CD-ROM 光碟機
- IDE ATA/ATAPI 控制器
- Jungo
- 人性化介面裝置
- 多功能介面卡
- 系統裝置
- 其他裝置
  - USB-Blaster
- 音效，視訊及遊戲控制器
- 處理器
- 通用序列匯流排控制器
  - eHome Infrared Receiver (USBCIR)
  - Intel(R) USB 3.0 可延伸主機控制器
  - Intel(R) USB 3.0 根集線器
  - Realtek USB 2.0 Card Reader
  - USB 2.0 集線器
  - USB Composite Device
  - USB Composite Device
  - USB Composite Device
- 連接埠 (COM 和 LPT)
- 智慧卡讀卡機
- 滑鼠及其他指標裝置
- 電池
- 電腦
- 監視器
- 磁碟機

動作
- 裝置管理員
  - 其他動作

右鍵選更新驅動程式軟體

## Programming DE0-CV (7/12)



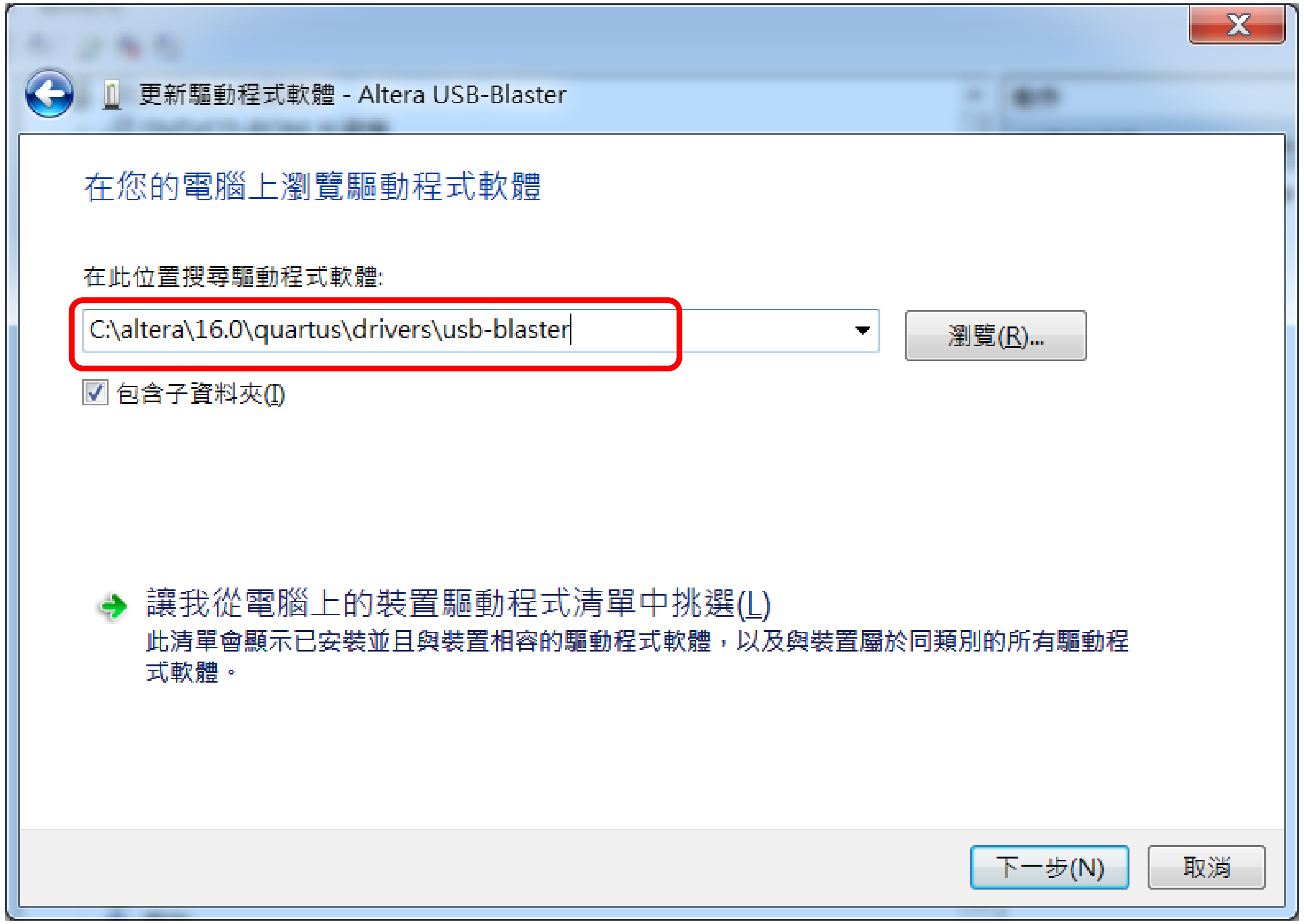
更新驅動程式軟體 - Altera USB-Blaster

您要如何搜尋驅動程式軟體?

自動搜尋更新的驅動程式軟體(S)
除非您在裝置安裝設定中停用此功能，否則 Windows 將在您的電腦和網際網路中搜尋是否有裝置適用的最新驅動程式軟體。

瀏覽電腦上的驅動程式軟體(R)
手動尋找並安裝驅動程式軟體。

# Programming DE0-CV (8/12)



更新驅動程式軟體 - Altera USB-Blaster

在您的電腦上瀏覽驅動程式軟體

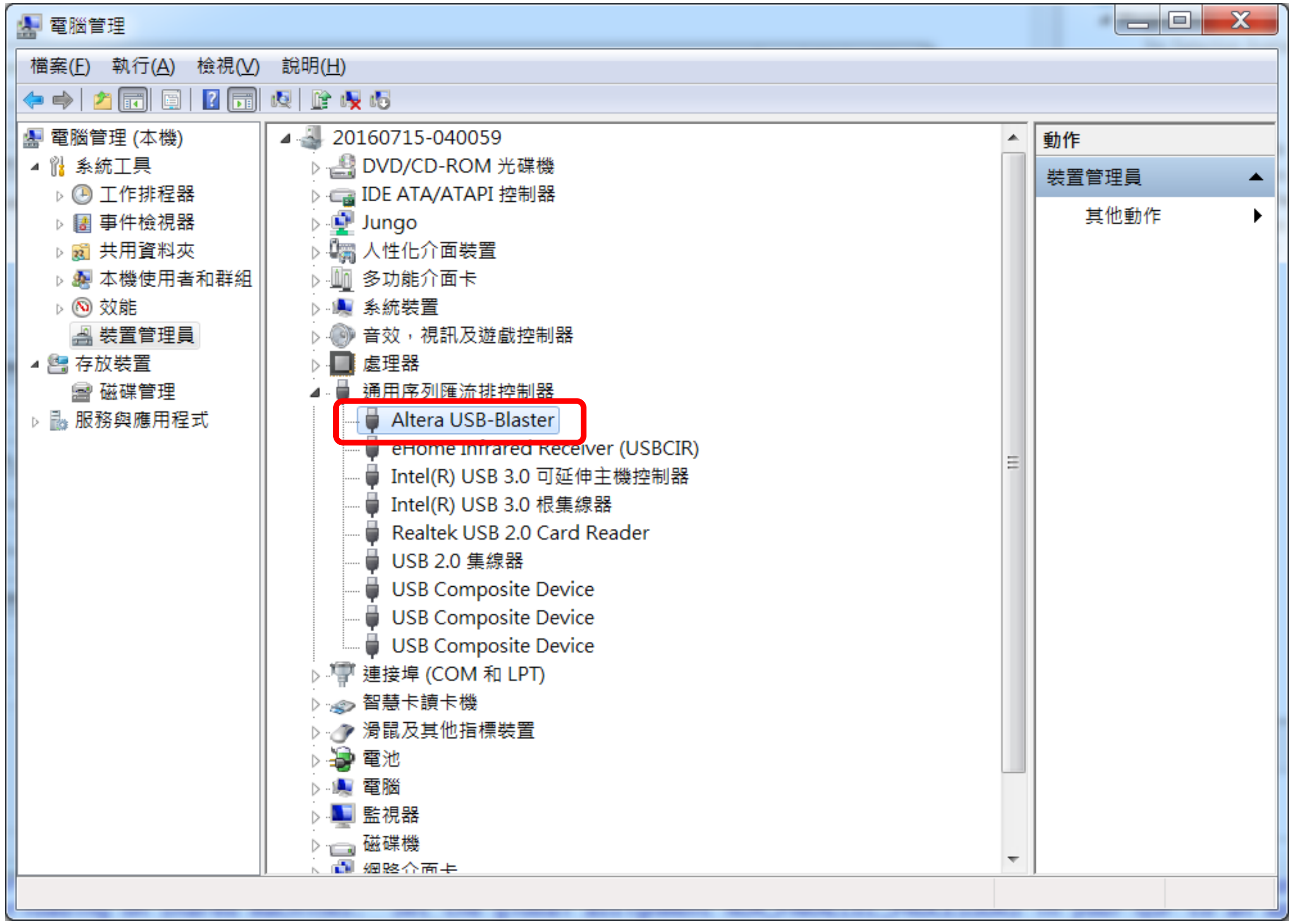
在此位置搜尋驅動程式軟體:

C:\altera\16.0\quartus\drivers\usb-blaster

讓我從電腦上的裝置驅動程式清單中挑選(L)

此清單會顯示已安裝並且與裝置相容的驅動程式軟體，以及與裝置屬於同類別的所有驅動程式軟體。

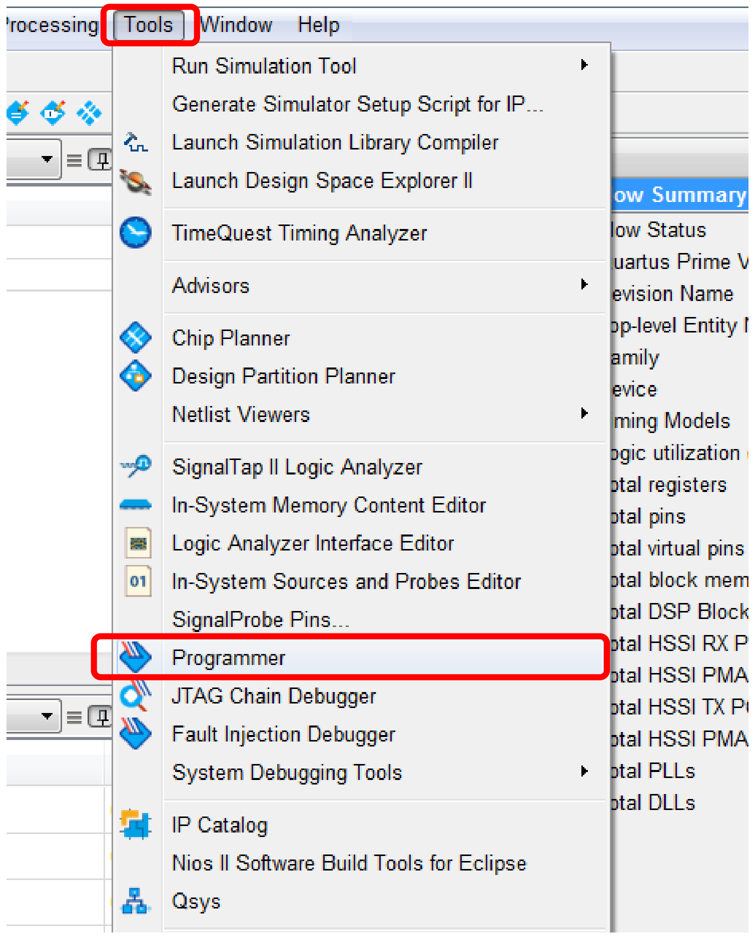
# Programming DE0-CV (9/12)



20160715-040059
- DVD/CD-ROM 光碟機
- IDE ATA/ATAPI 控制器
- Jungo
- 人性化介面裝置
- 多功能介面卡
- 系統裝置
- 音效，視訊及遊戲控制器
- 處理器
- 通用序列匯流排控制器
  - Altera USB-Blaster
  - eHome Infrared Receiver (USBCIR)
  - Intel(R) USB 3.0 可延伸主機控制器
  - Intel(R) USB 3.0 根集線器
  - Realtek USB 2.0 Card Reader
  - USB 2.0 集線器
  - USB Composite Device
  - USB Composite Device
  - USB Composite Device
- 連接埠 (COM 和 LPT)
- 智慧卡讀卡機
- 滑鼠及其他指標裝置
- 電池
- 電腦
- 監視器
- 磁碟機
- 網路介面卡

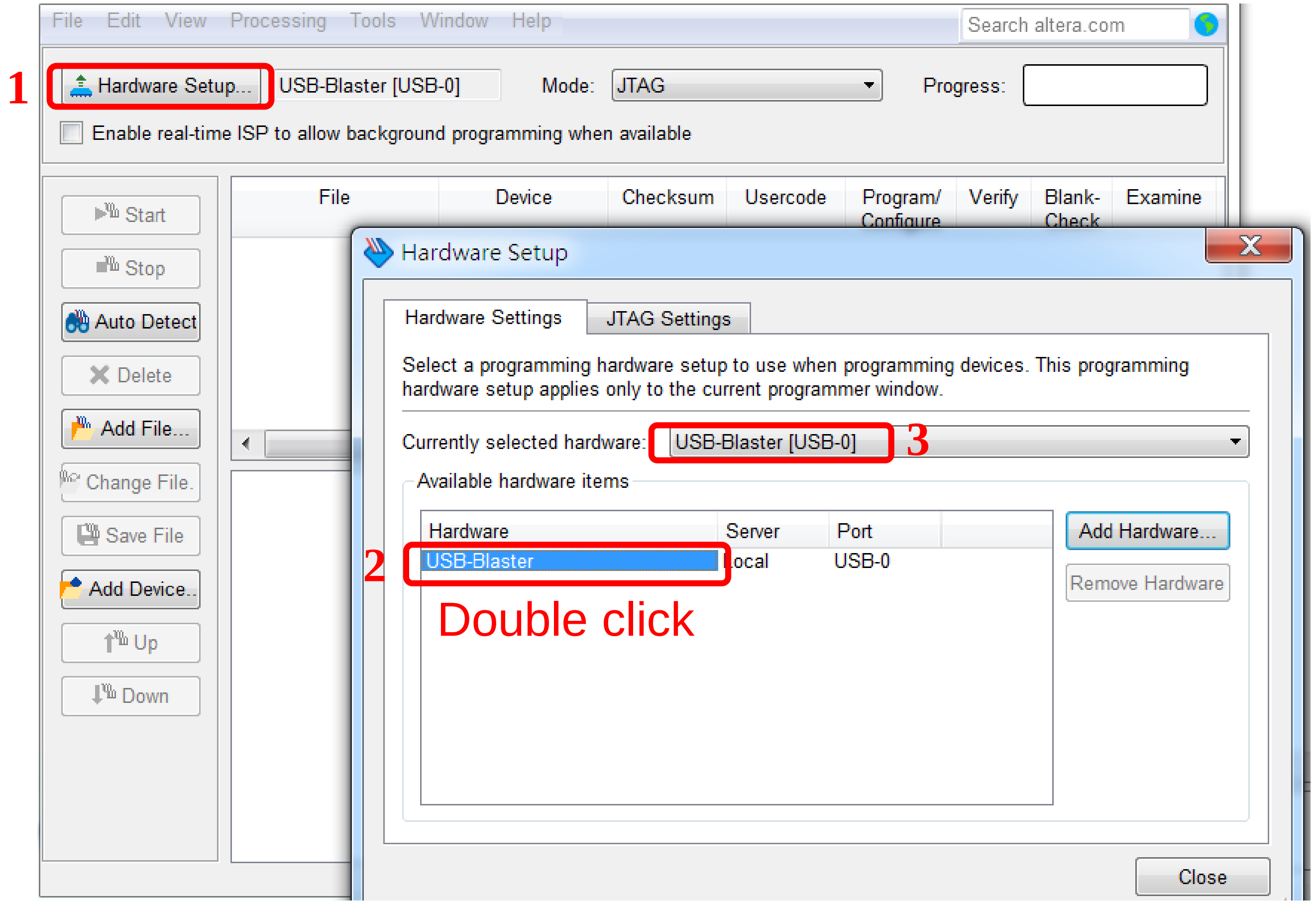
# Programming DE0-CV (10/12)

- Programming device



# Programming DE0-CV (11/12)

- Hardware setup: add USB-Blaster



Enable real-time ISP to allow background programming when available

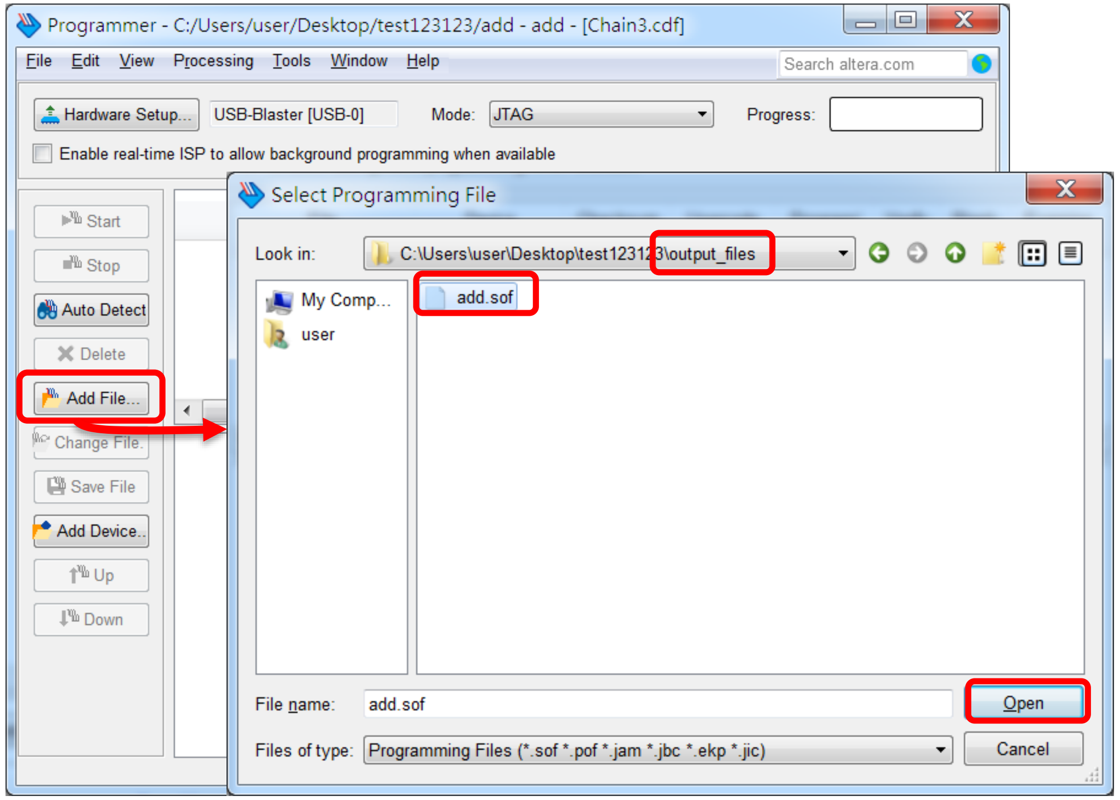
Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.

Currently selected hardware: USB-Blaster [USB-0]

| Hardware | Server | Port |
|---|---|---|
| USB-Blaster | Local | USB-0 |

# Programming DE0-CV (12/12)

- Programming device



Programmer - C:/Users/user/Desktop/test123123/add - add - [Chain3.cdf]

Hardware Setup... USB-Blaster [USB-0]    Mode: JTAG    Progress:

☐ Enable real-time ISP to allow background programming when available

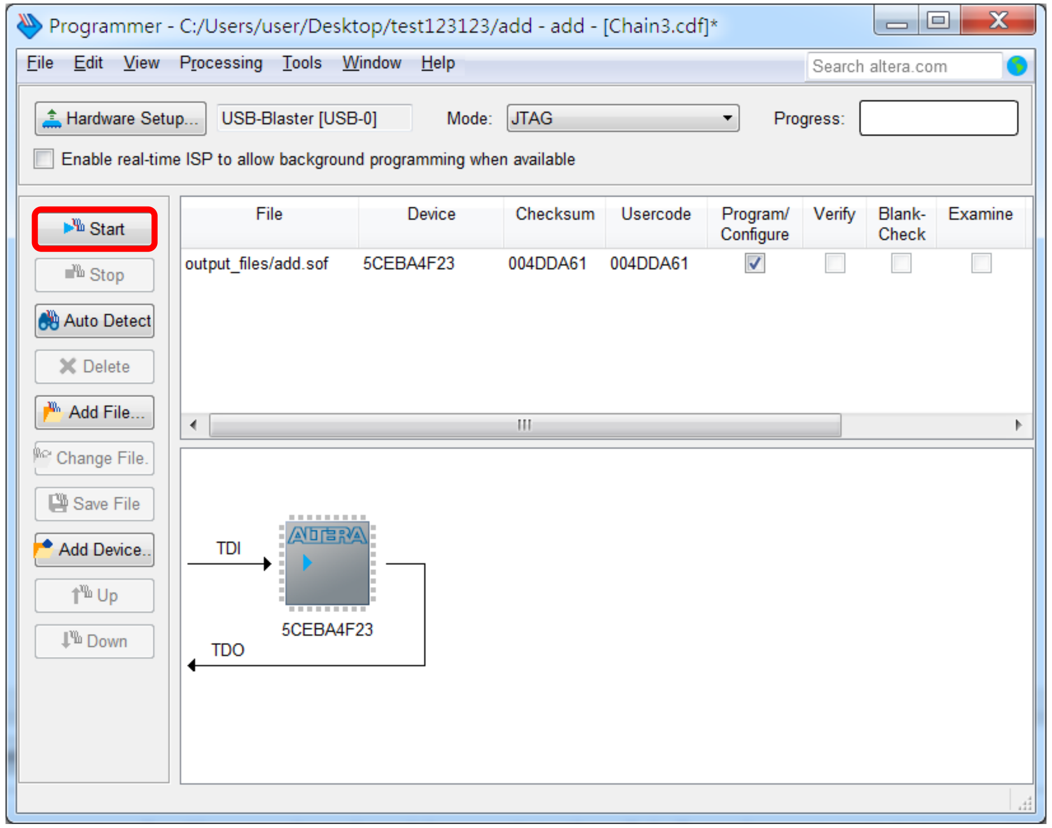
Select Programming File

Look in: C:\Users\user\Desktop\test123123\output_files

add.sof

File name: add.sof

Files of type: Programming Files (*.sof *.pof *.jam *.jbc *.ekp *.jic)



Programmer - C:/Users/user/Desktop/test123123/add - add - [Chain3.cdf]*

Hardware Setup... USB-Blaster [USB-0]    Mode: JTAG    Progress:

☐ Enable real-time ISP to allow background programming when available

| File | Device | Checksum | Usercode | Program/ Configure | Verify | Blank-Check | Examine |
|---|---|---|---|---|---|---|---|
| output_files/add.sof | 5CEBA4F23 | 004DDA61 | 004DDA61 | ✓ | ☐ | ☐ | ☐ |