

Sequential Circuits and MegaFunction

2016/03/04 YZU CSE

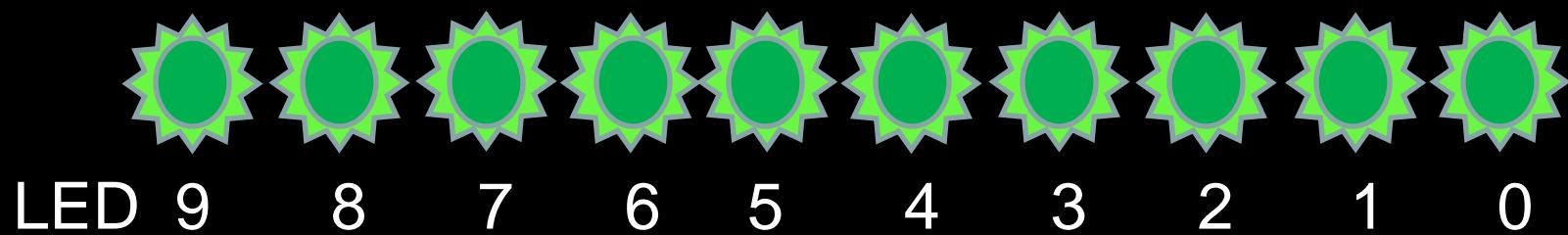
Outline

- **Sequential Circuits**
 - Lab1 Lightning LED
 - Lab2 Lightning LED with Megafunction
- **More Megafuctions**
 - Lab3 MAC (Multiplier-accumulator)

Knight Rider



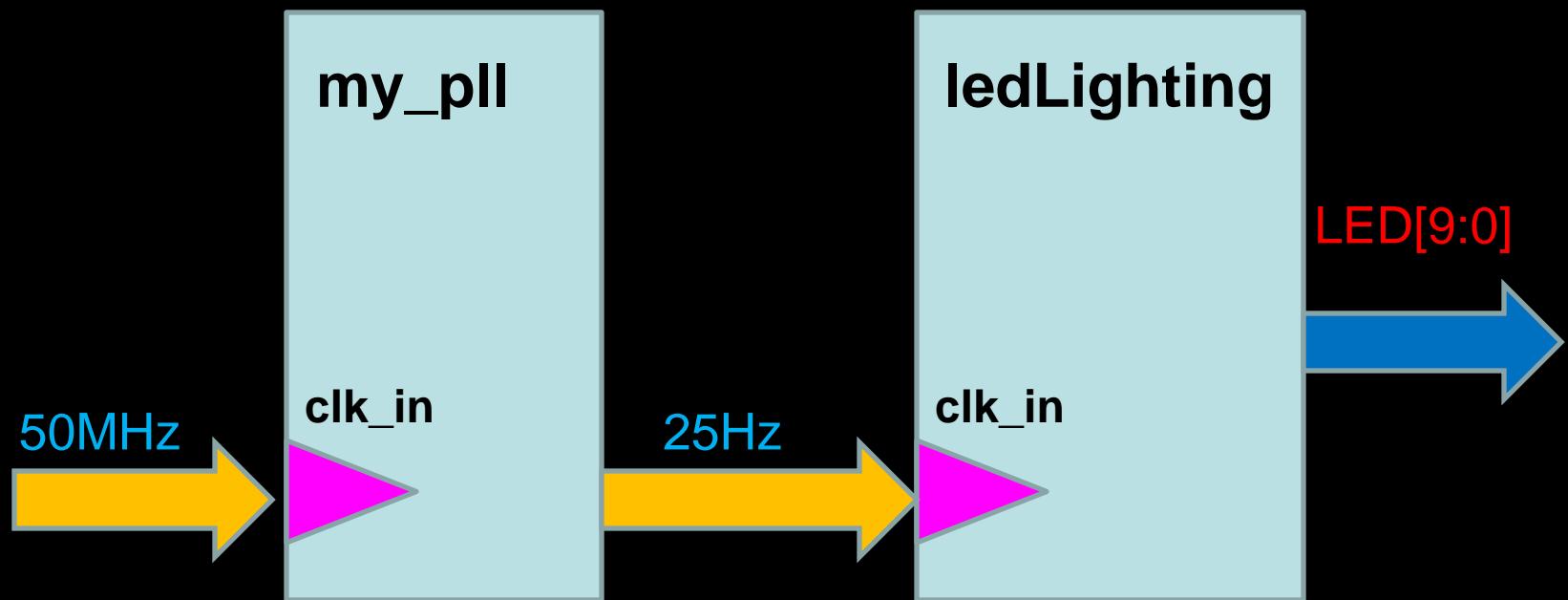
Design: Lightning LED



Outline

- **Sequential Circuits**
 - Lab1 Lightning LED
 - Lab2 Lightning LED with Megafunction
- **More Megafuctions**
 - Lab3 MAC (Multiplier-accumulator)

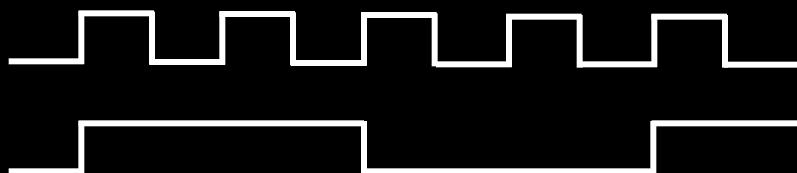
Lab1: Lightning LED



Lab1 Design Flow

- **Create a new project “Lab1”**
- **Create a new Verilog file “my_pll.v”**
- **Create a new Verilog file “ledLightning.v”**
- **Create two new symbols from my_pll.v and ledLightning.v**
- **Schematic Design**
- **Pin Assignment**
- **Full compilation**

Module1 my_pll



New_clock = 1/2n * Original_clock

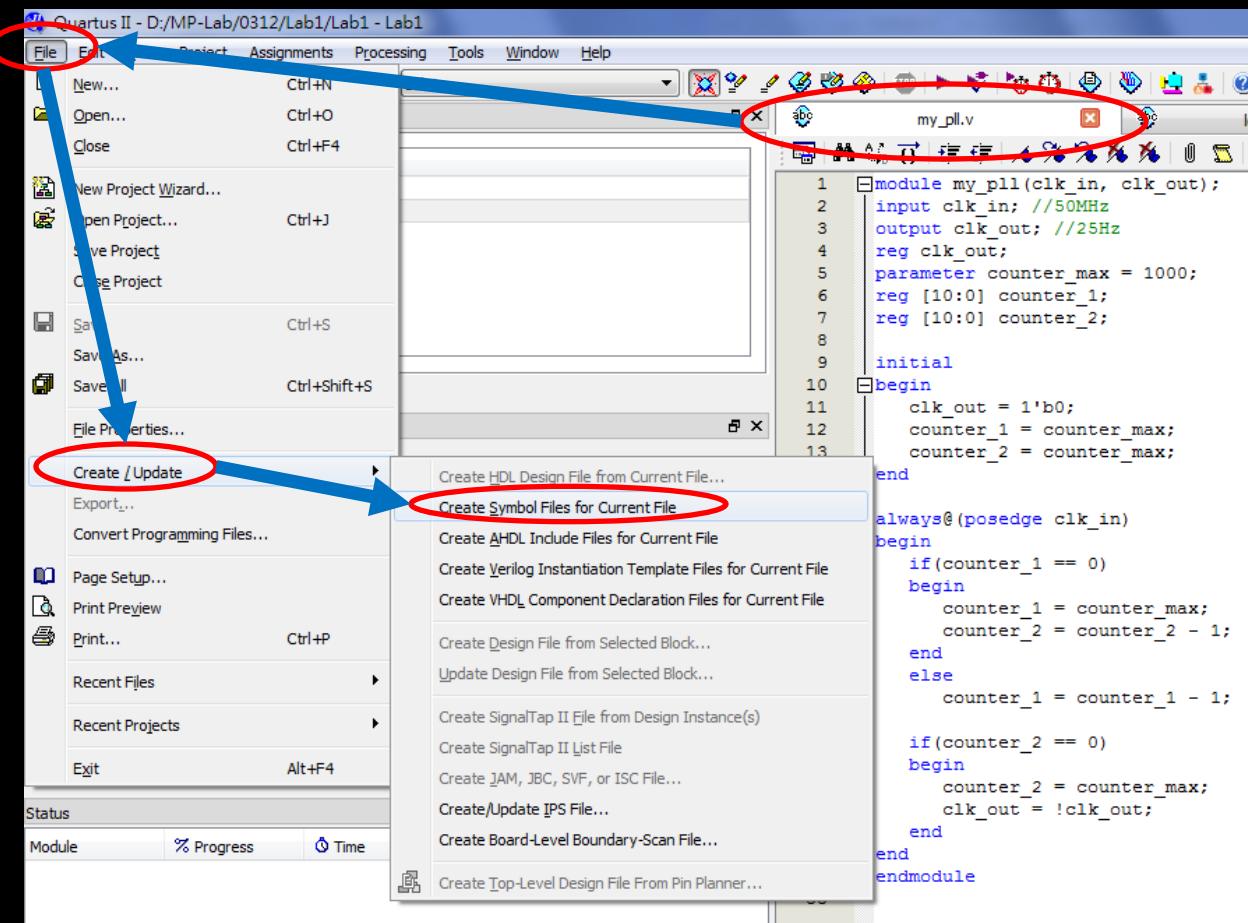
```
1  module my_pll(clk_in, clk_out);
2    input clk_in; //50MHz
3    output clk_out; //25Hz
4    reg clk_out;
5    parameter counter_max = 1000000;
6    reg [19:0] counter_1;
7
8    initial
9    begin
10       clk_out = 1'b0;
11       counter_1 = counter_max;
12   end
13
14   always@ (posedge clk_in)
15   begin
16     if(counter_1 == 0)
17     begin
18       counter_1 = counter_max;
19       clk_out = !clk_out;
20     end
21     else
22       counter_1 = counter_1 - 1;
23
24   end
25 endmodule
26
```

Module2 ledLightning

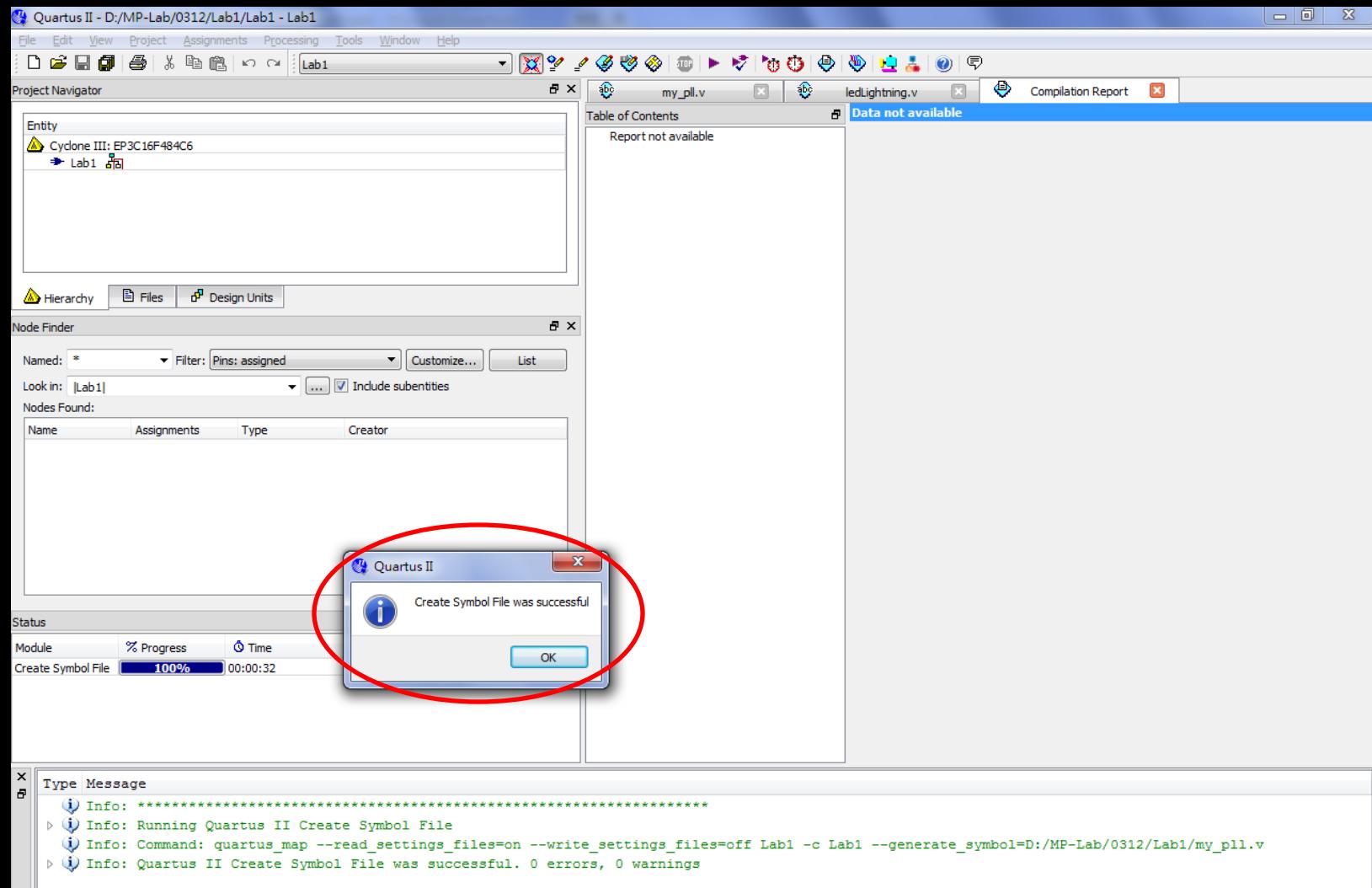
```
1  module ledLightning(clk, Led);
2    input clk;
3    output [9:0] Led;
4    reg [9:0]Led;
5    reg direction;
6    initial
7    begin
8      Led[9:1]=9'b0;
9      Led[0]=1'b1;
10     direction=1'b0;
11   end
12
13  always@ (posedge clk)
14  begin
15    if(Led[0] || Led[9])
16      direction = !direction;
17
18    if(direction)
19    begin
20      Led[0]<=Led[9];
21      Led[1]<=Led[0];
22      Led[2]<=Led[1];
23      Led[3]<=Led[2];
24      Led[4]<=Led[3];
25      Led[5]<=Led[4];
26      Led[6]<=Led[5];
27      Led[7]<=Led[6];
28      Led[8]<=Led[7];
29      Led[9]<=Led[8];
30    end
31  else begin
32      Led[8]<=Led[9];
33      Led[7]<=Led[8];
34      Led[6]<=Led[7];
35      Led[5]<=Led[6];
36      Led[4]<=Led[5];
37      Led[3]<=Led[4];
38      Led[2]<=Led[3];
39      Led[1]<=Led[2];
40      Led[0]<=Led[1];
41      Led[9]<=Led[0];
42    end
43  end
44 endmodule
```

Create a Symbol File from Verilog-1

- File -> Create / Update -> Create Symbol Files for Current File

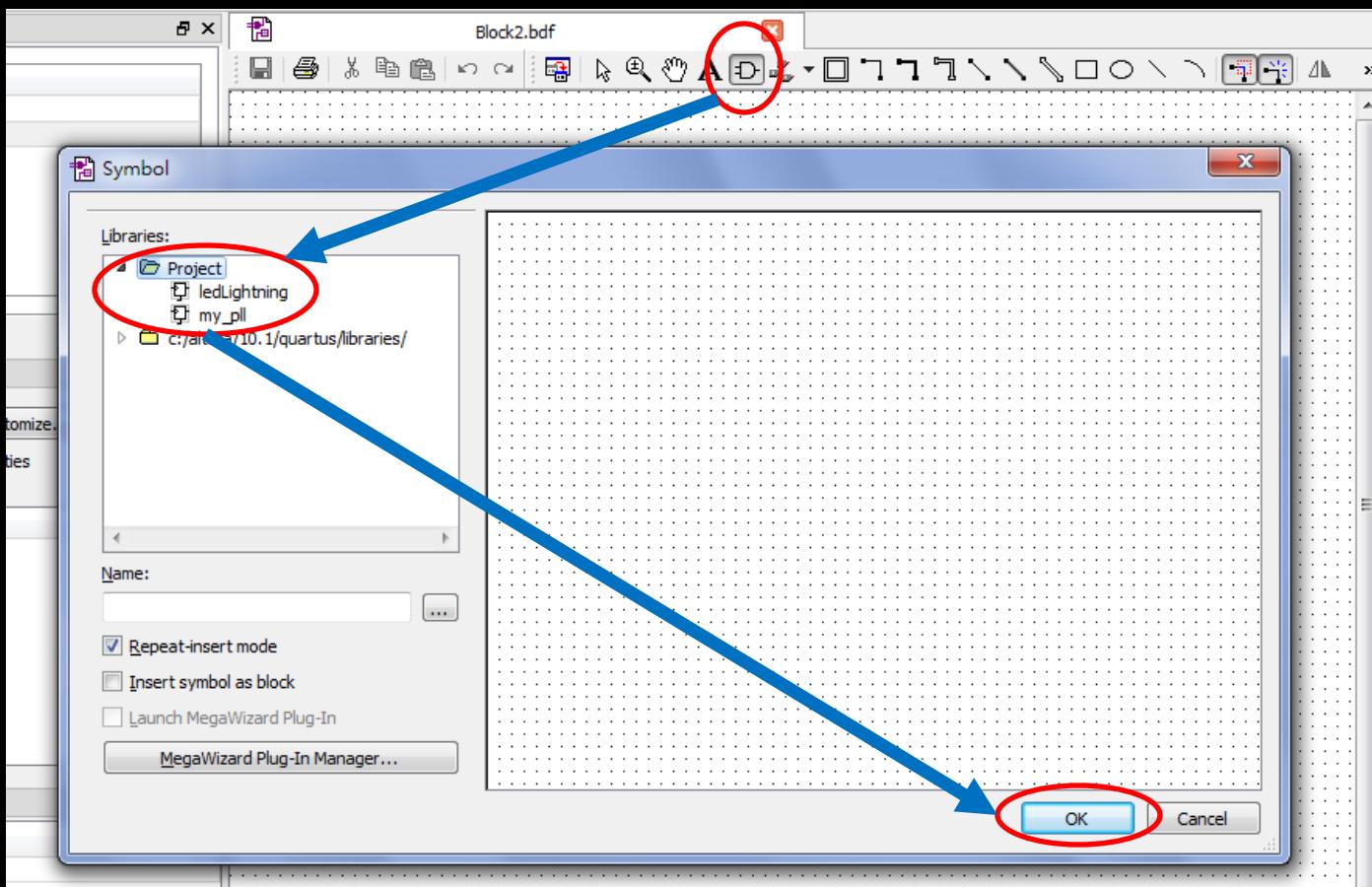


Create a Symbol File from Verilog-2

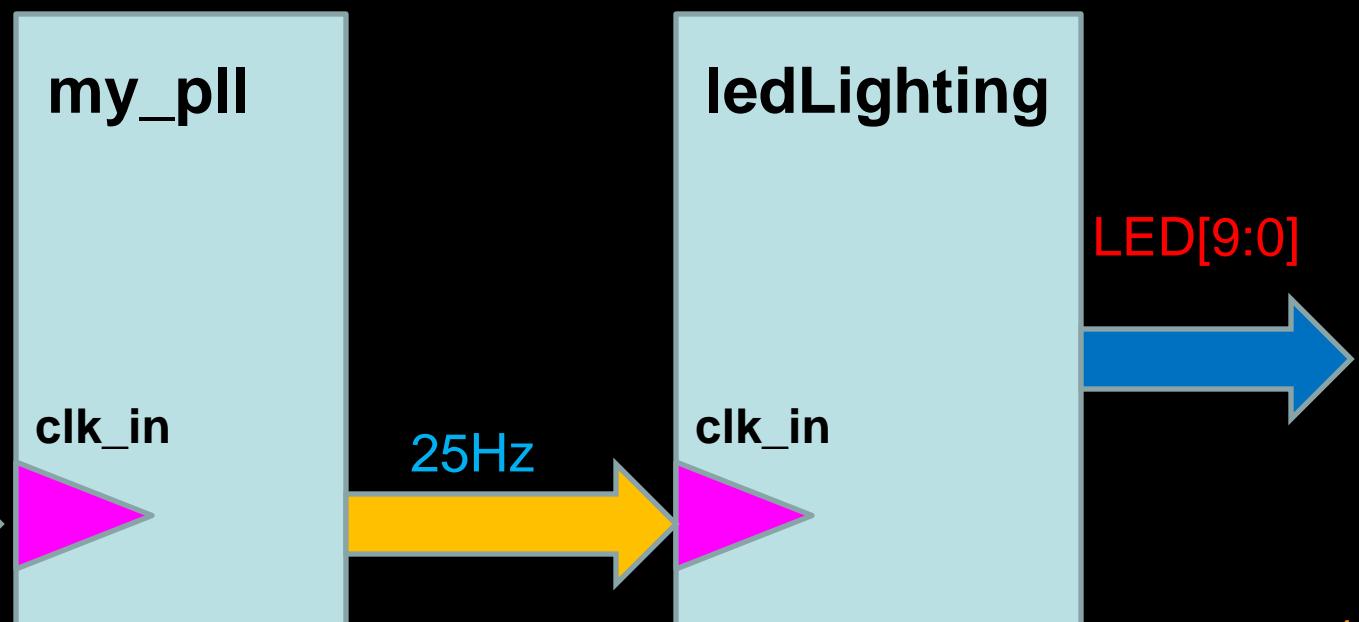
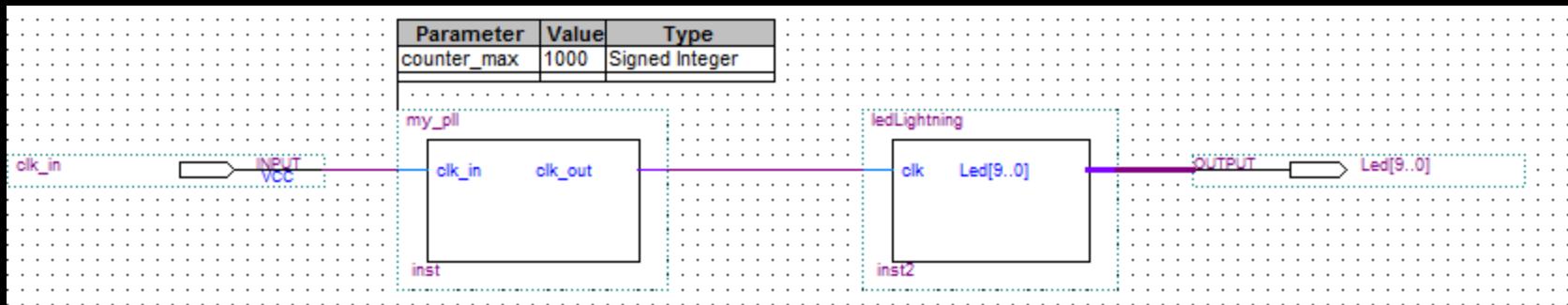


Schematic Design-1

- File -> New -> Block Diagram / Schematic File



Schematic Design-2



Pin Assignment

Pin Planner - D:/MP-Lab/0312/Lab1/Lab1 - Lab1

File Edit View Processing Tools Window

Groups
Named: *

Node Name	Direction	Location
Led[9..0]	Output Group	
<<new group>>		

Top View - Wire Bond
Cyclone III - EP3C16F484C6

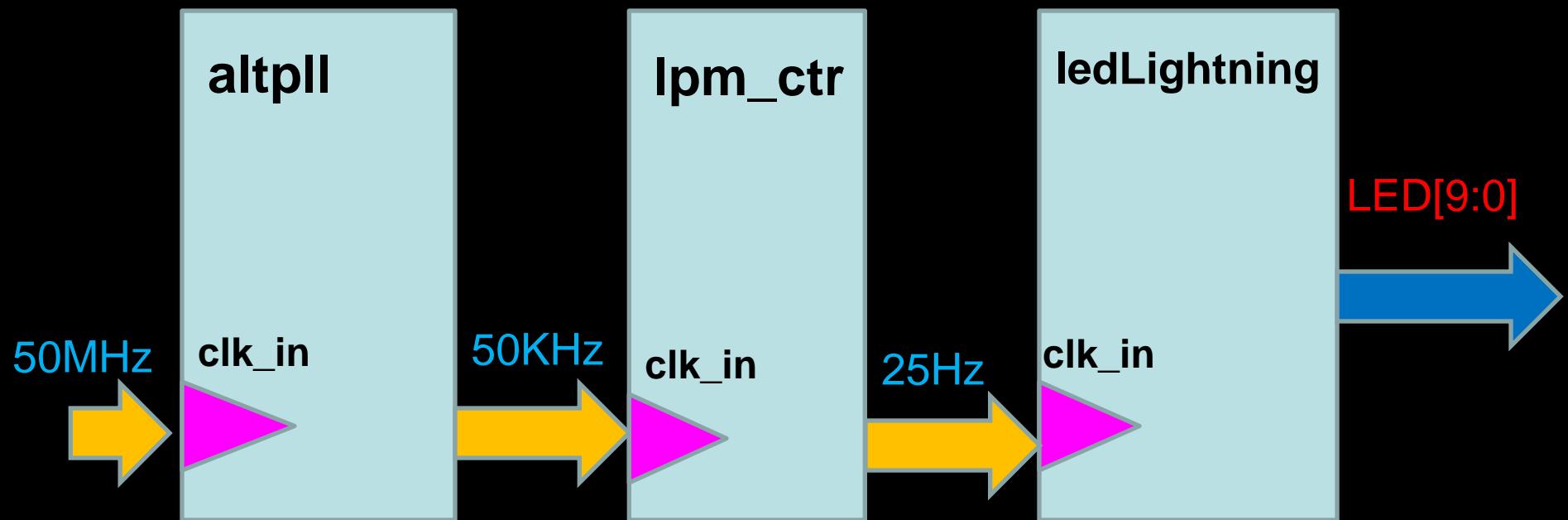
The diagram shows the top view of a Cyclone III EP3C16F484C6 FPGA package. It features 22 pins around the perimeter and a central array of logic cells. The package is organized into four columns of pins labeled A through H and two rows labeled 1 through 22. The central array is color-coded into several regions: yellow (top), green (middle), orange (right), purple (bottom), pink (left), and grey (center). Each pin is represented by a symbol indicating its function, such as a triangle for output or a circle for input.

Node Name	Direction	Location	I/O Standard	Reserved
clk_in	Input	PIN_G21	2.5 V (default)	
Led[9]	Output	PIN_B1	2.5 V (default)	
Led[8]	Output	PIN_B2	2.5 V (default)	
Led[7]	Output	PIN_C2	2.5 V (default)	
Led[6]	Output	PIN_C1	2.5 V (default)	
Led[5]	Output	PIN_E1	2.5 V (default)	
Led[4]	Output	PIN_F2	2.5 V (default)	
Led[3]	Output	PIN_H1	2.5 V (default)	
Led[2]	Output	PIN_J3	2.5 V (default)	
Led[1]	Output	PIN_J2	2.5 V (default)	
Led[0]	Output	PIN_J1	2.5 V (default)	
<<new node>>				

Outline

- **Sequential Circuits**
 - Lab1 Lightning LED
 - Lab2 Lightning LED with Megafunction
- **More Megafuctions**
 - Lab3 MAC (Multiplier-accumulator)

Lab2: Lightning LED with Megafunction



Lab2 Design Flow

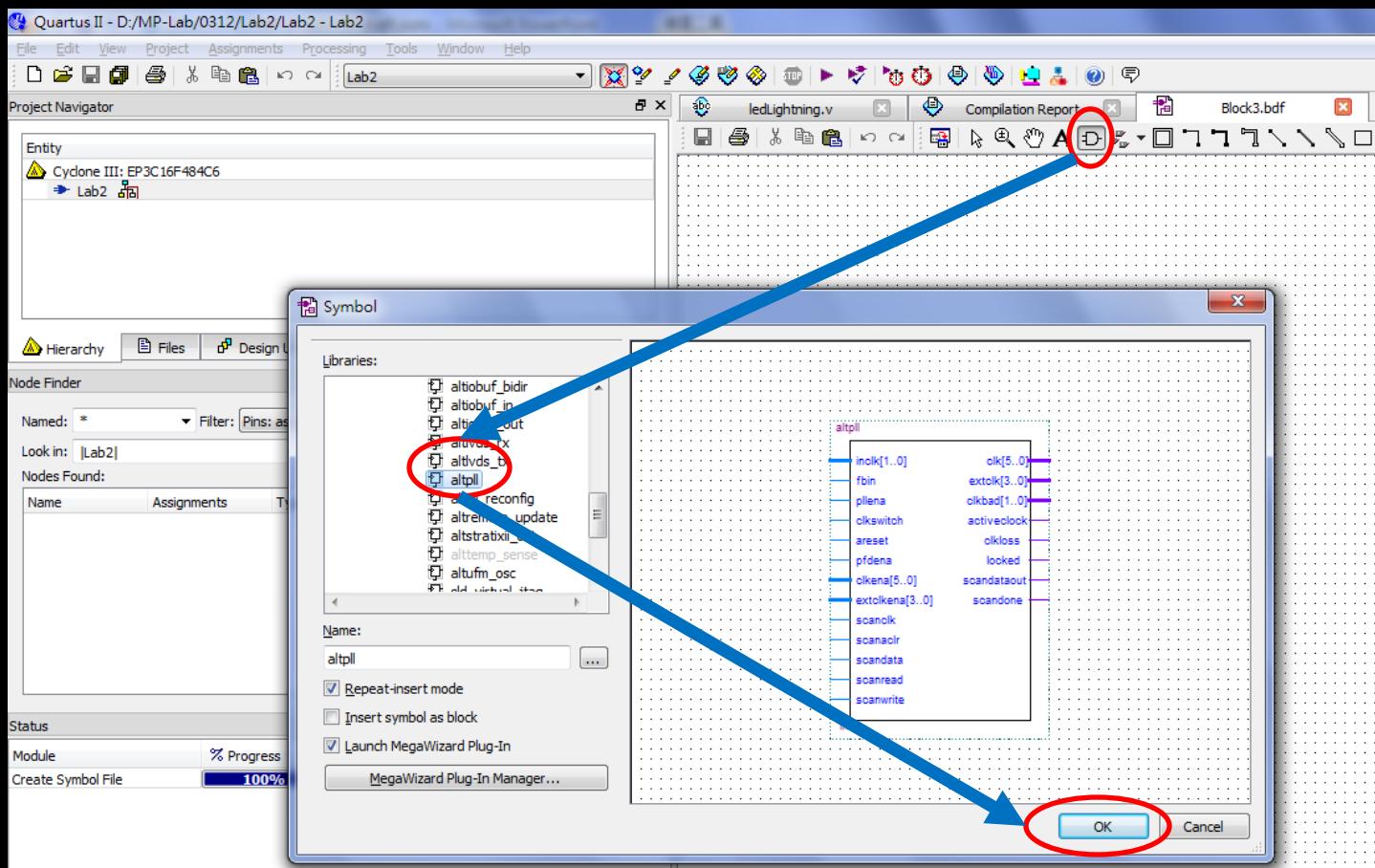
- **Create a new project “Lab2”**
- **Create a new Verilog file “ledLightning.v”**
- **Create a new symbol from ledLightning.v**
- **Schematic Design**
 - Megafunction “ALTPLL”
 - Megafunction “LPM_COUNTER”
- **Pin Assignment**
- **Full compilation**

Module1 ledLightning

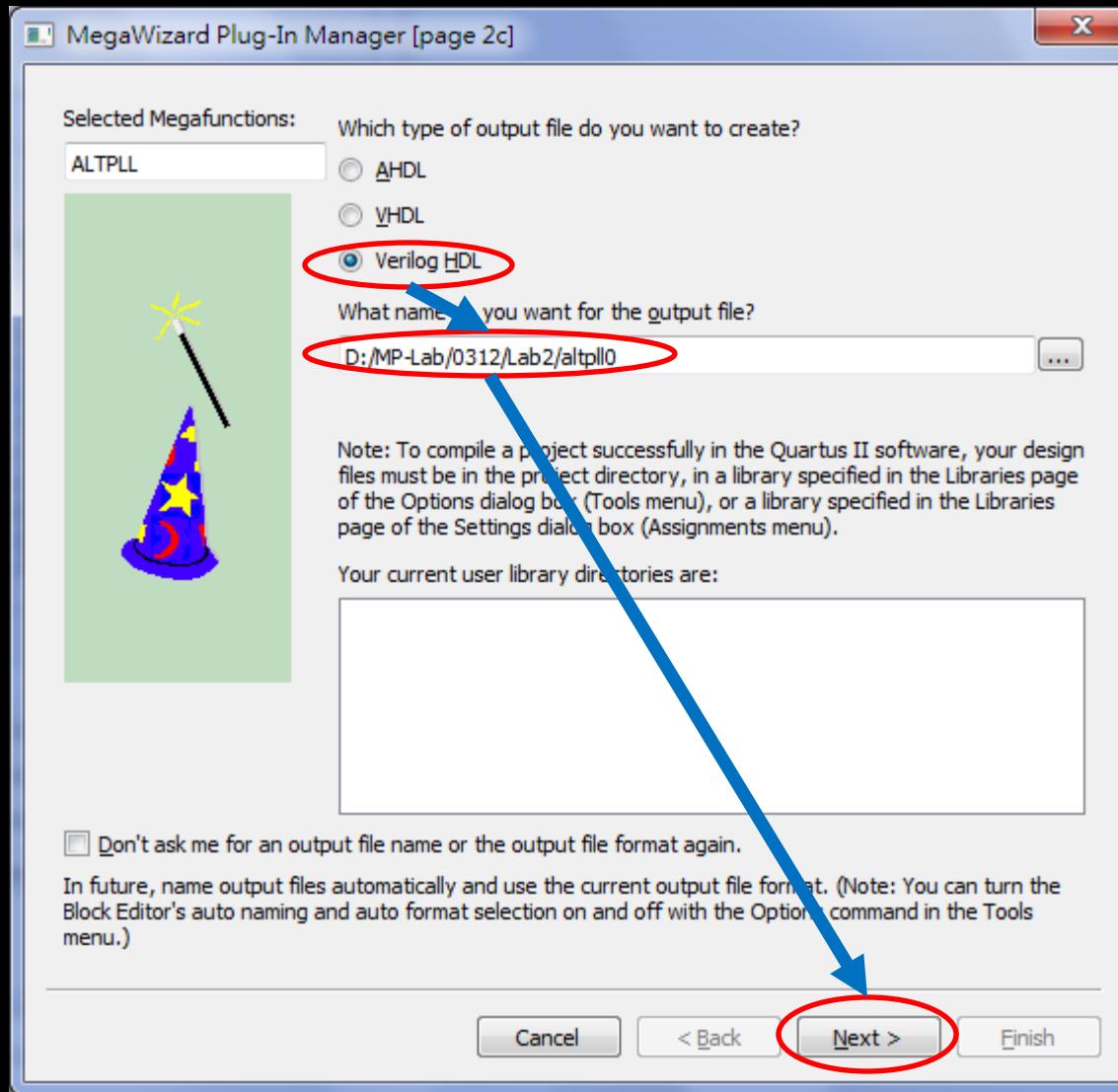
```
1  module ledLightning(clk, Led);
2    input clk;
3    output [9:0] Led;
4    reg [9:0]Led;
5    reg direction;
6    initial
7    begin
8      Led[9:1]=9'b0;
9      Led[0]=1'b1;
10     direction=1'b0;
11   end
12
13  always@ (posedge clk)
14  begin
15    if(Led[0] || Led[9])
16      direction = !direction;
17
18    if(direction)
19    begin
20      Led[0]<=Led[9];
21      Led[1]<=Led[0];
22      Led[2]<=Led[1];
23      Led[3]<=Led[2];
24      Led[4]<=Led[3];
25      Led[5]<=Led[4];
26      Led[6]<=Led[5];
27      Led[7]<=Led[6];
28      Led[8]<=Led[7];
29      Led[9]<=Led[8];
30    end
31  else begin
32      Led[8]<=Led[9];
33      Led[7]<=Led[8];
34      Led[6]<=Led[7];
35      Led[5]<=Led[6];
36      Led[4]<=Led[5];
37      Led[3]<=Led[4];
38      Led[2]<=Led[3];
39      Led[1]<=Led[2];
40      Led[0]<=Led[1];
41      Led[9]<=Led[0];
42    end
43  end
44 endmodule
```

Schematic Design – ALTPPLL – 1

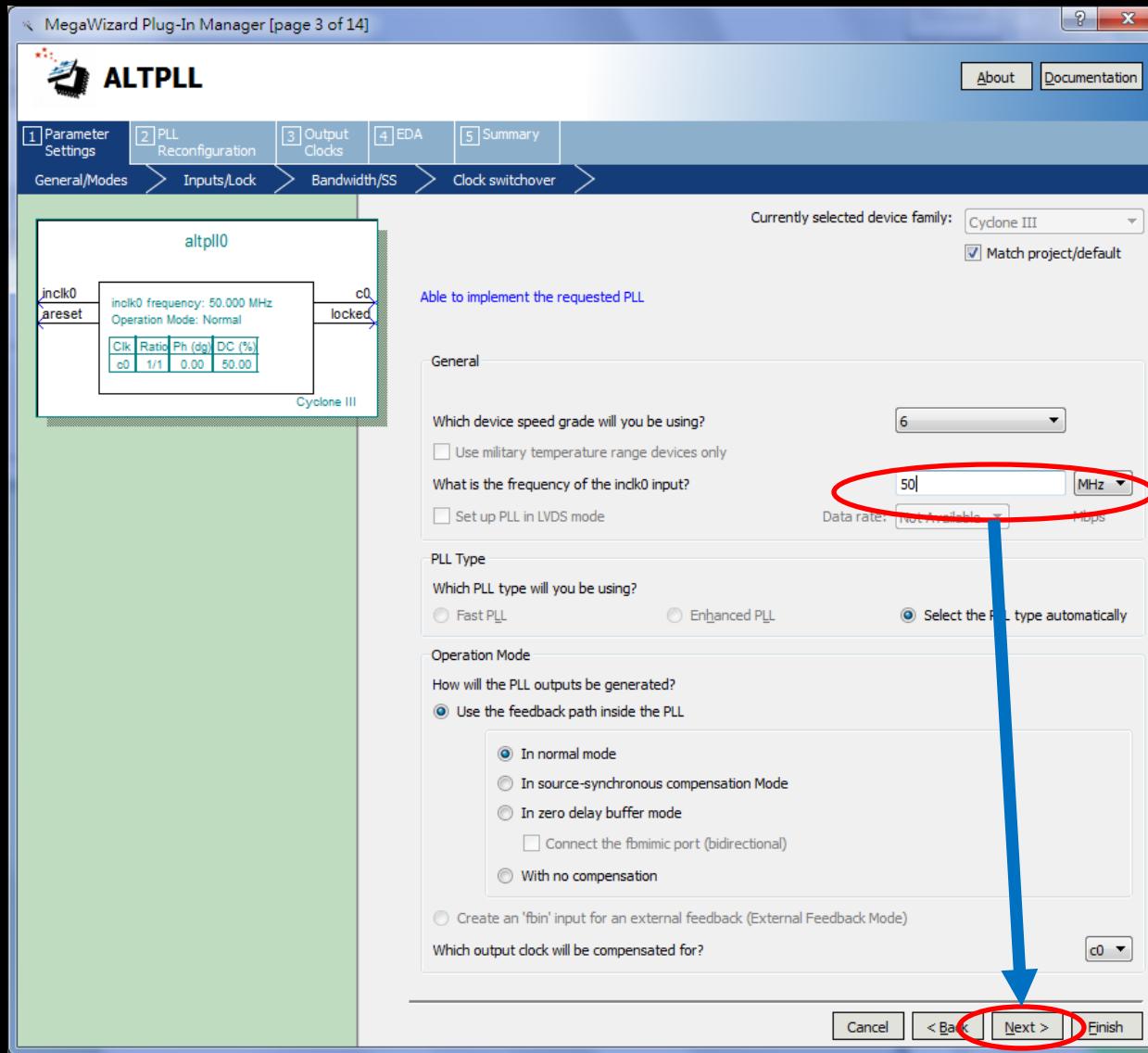
- Symbol Tool -> .../quartus/libraries -> megafuctions
-> IO ->altpll



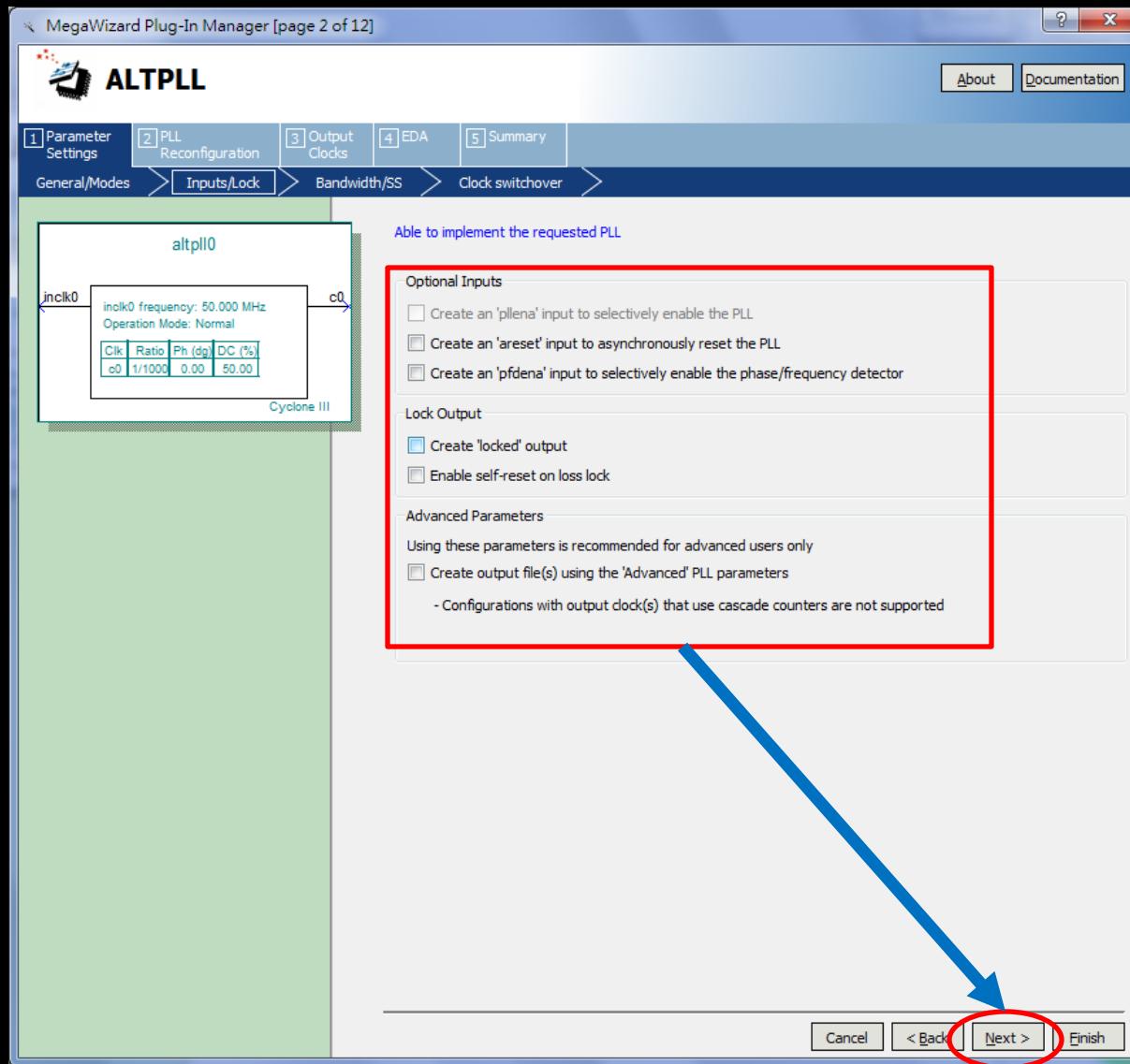
Schematic Design – ALTPLL – 2



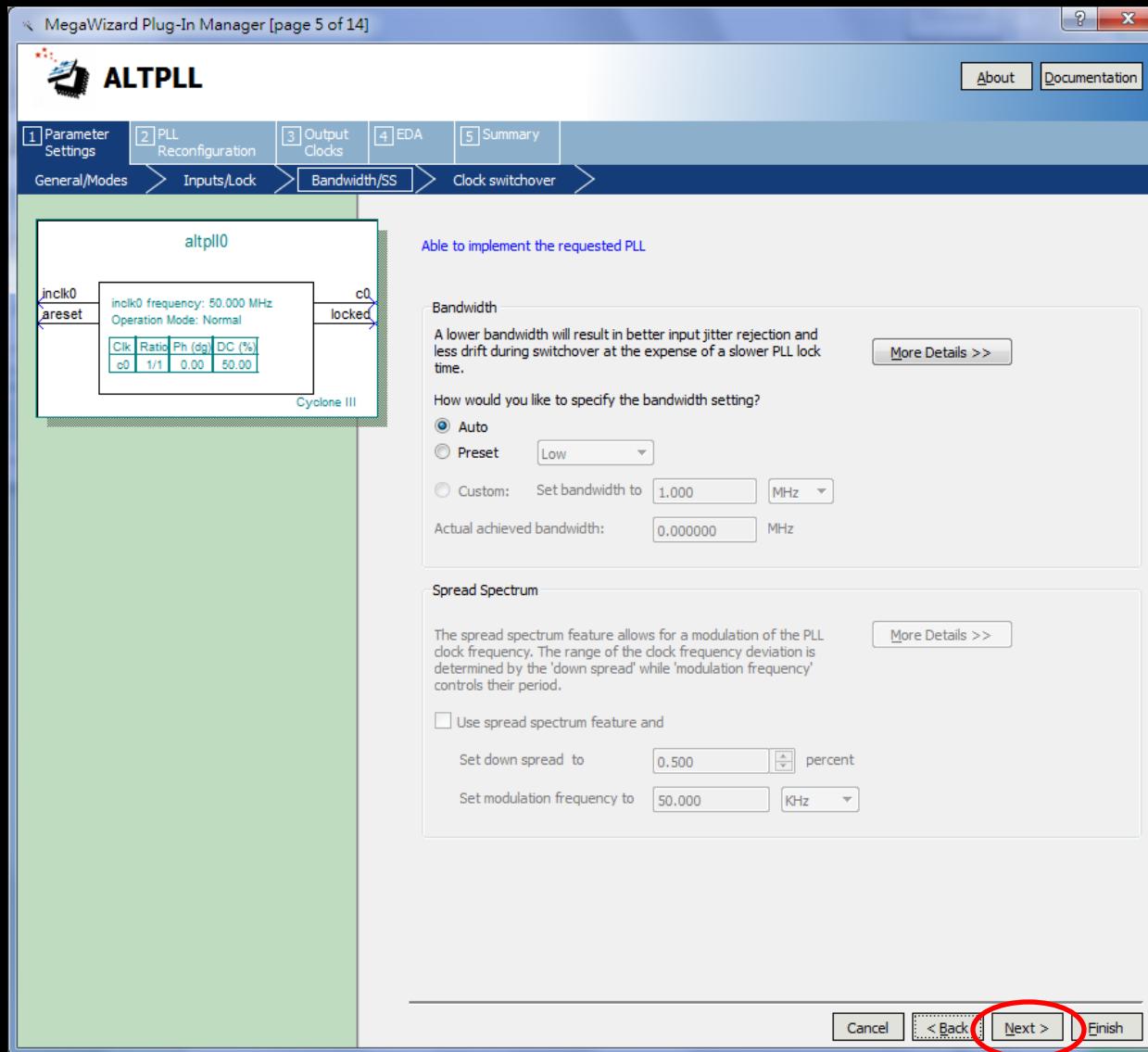
Schematic Design – ALTPLL – 3



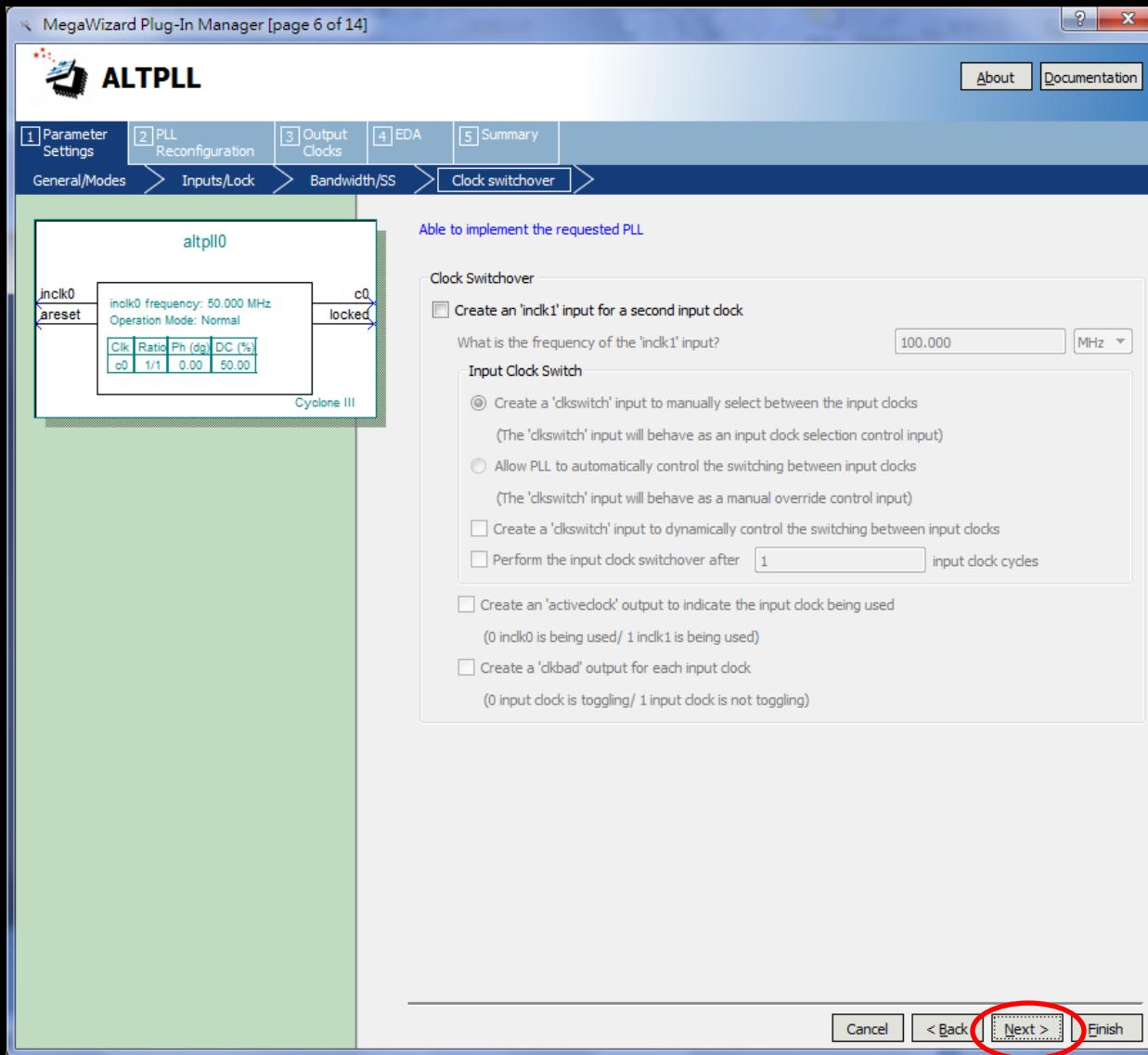
Schematic Design – ALTPLL – 4



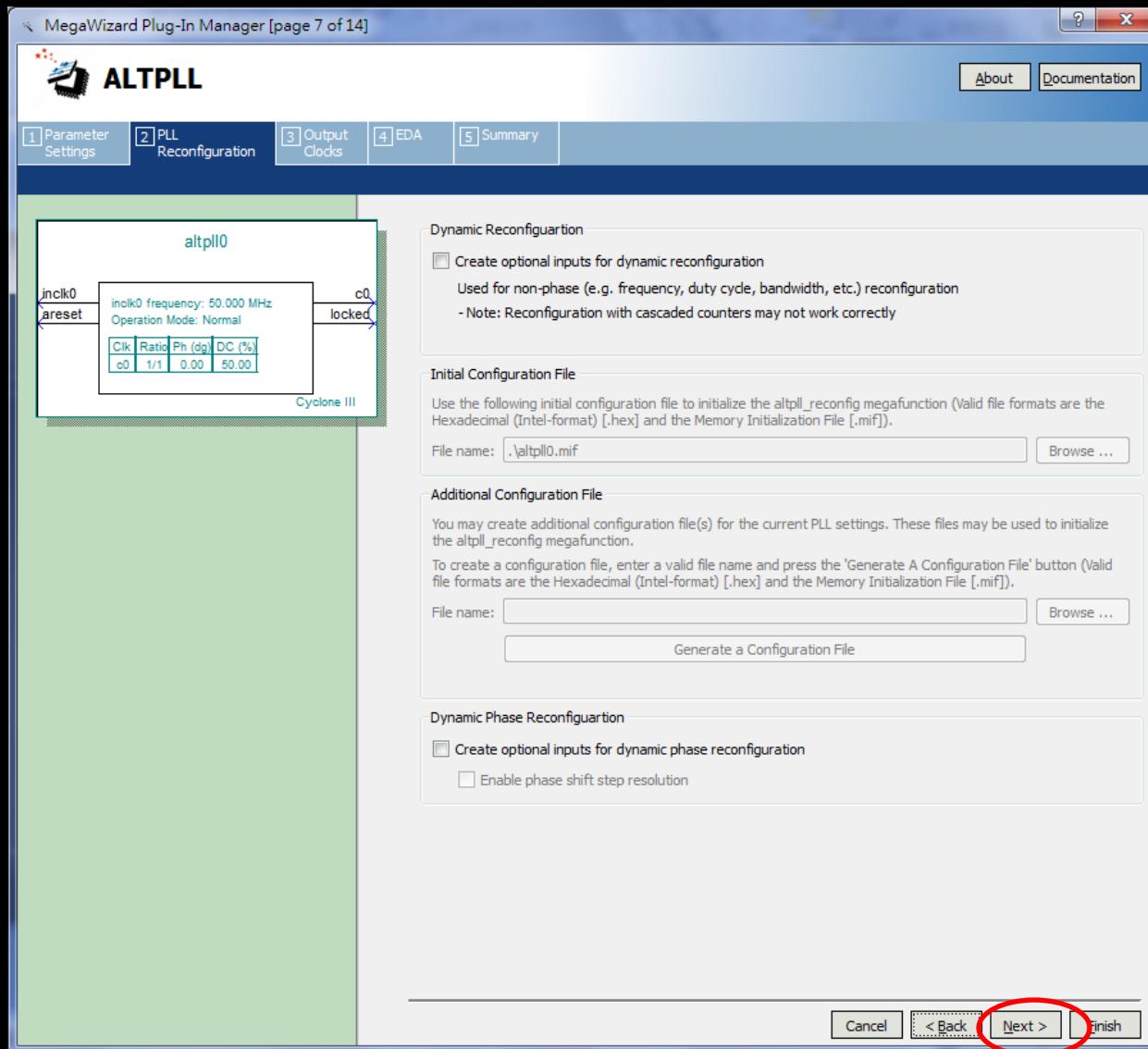
Schematic Design – ALTPLL – 5



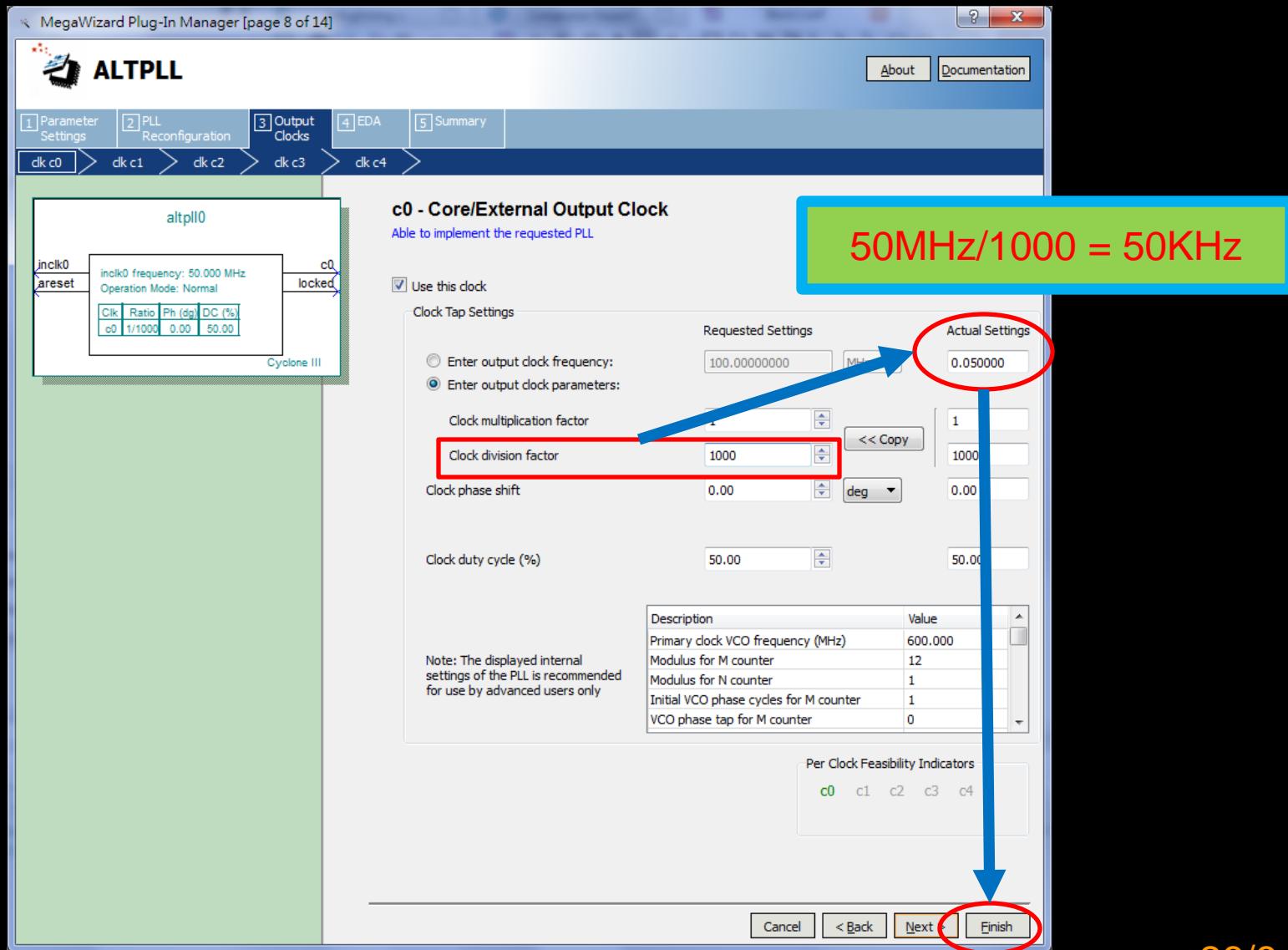
Schematic Design – ALTPLL – 6



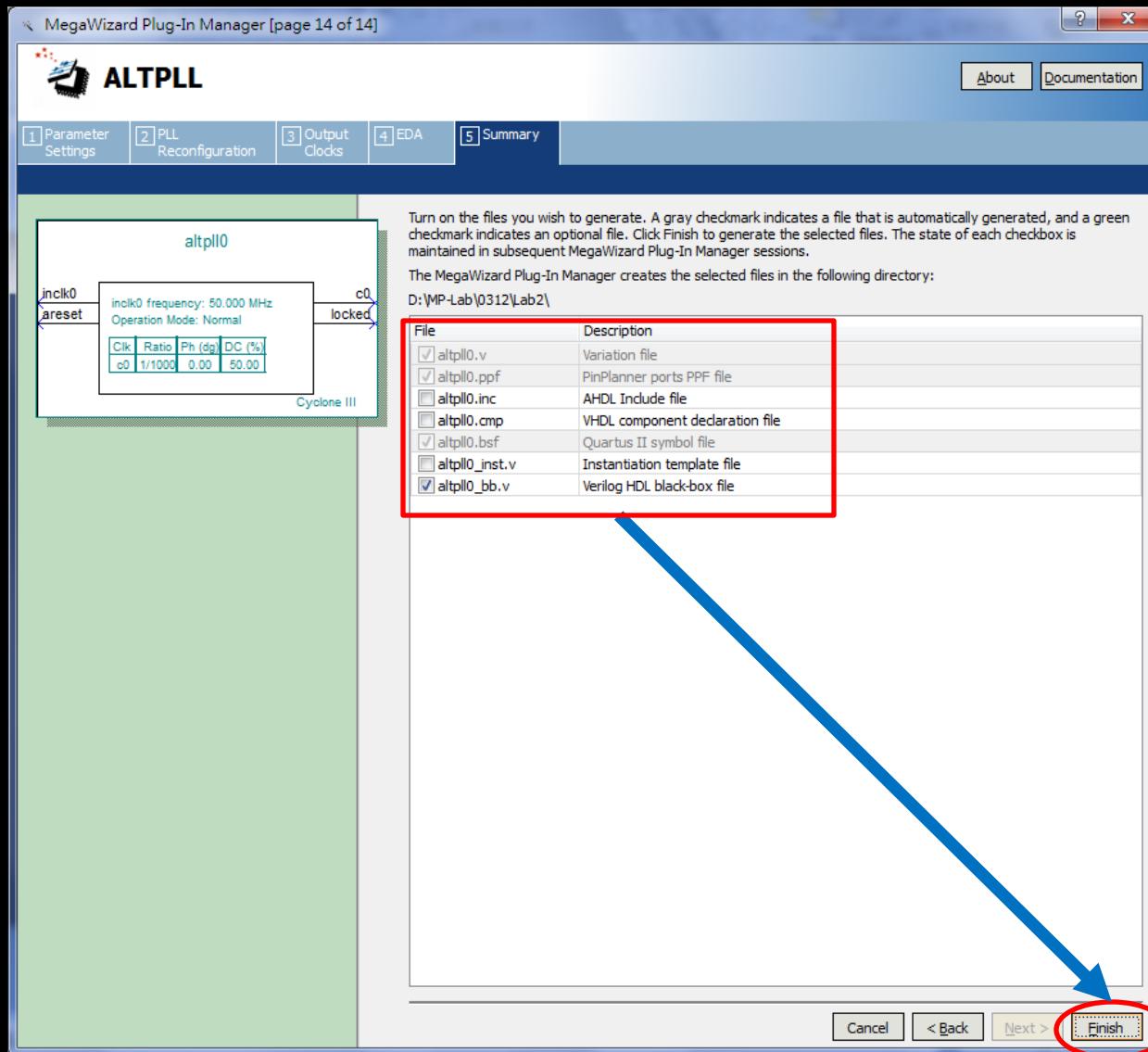
Schematic Design – ALTPLL – 7



Schematic Design – ALTPLL – 8

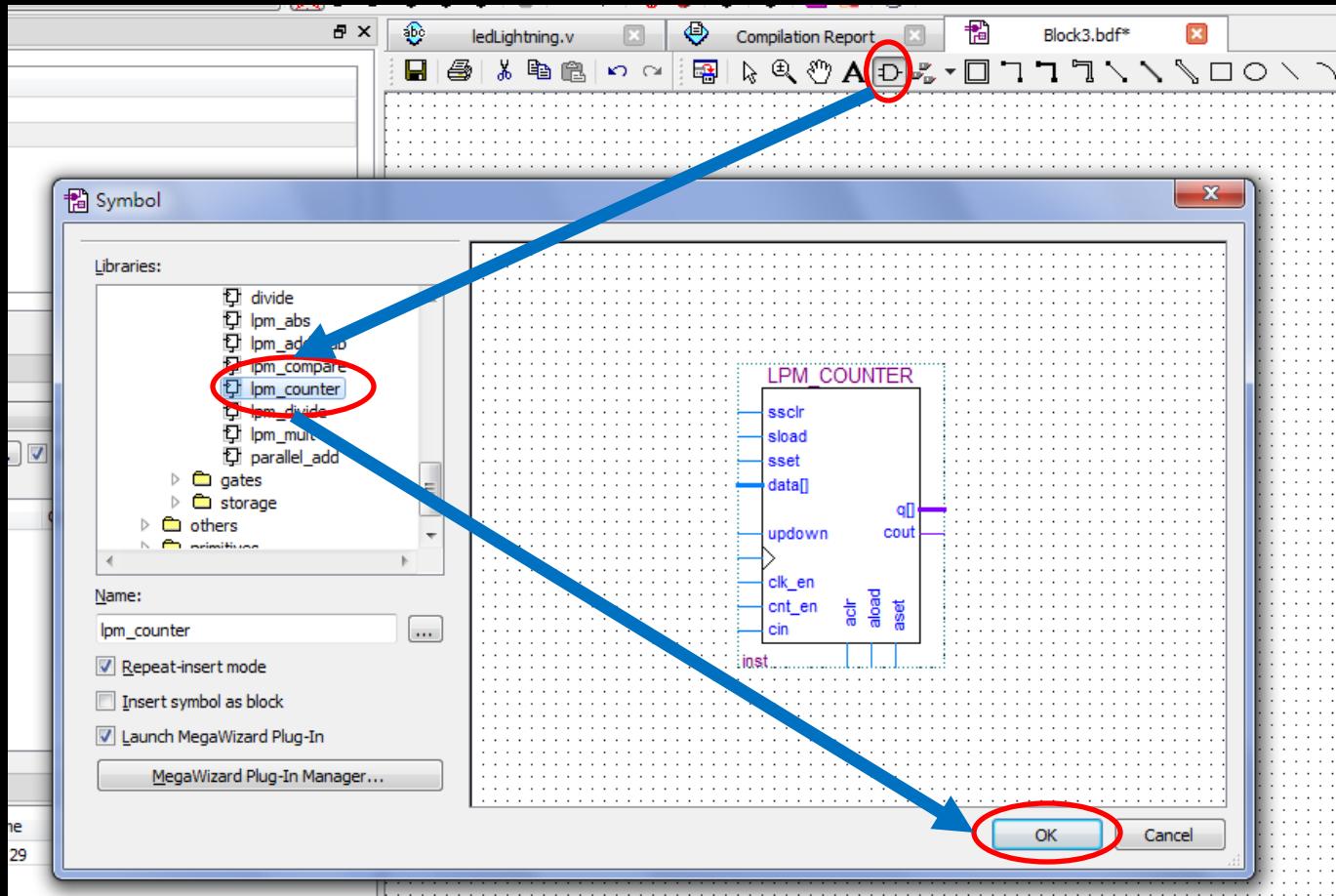


Schematic Design – ALTPLL – 9

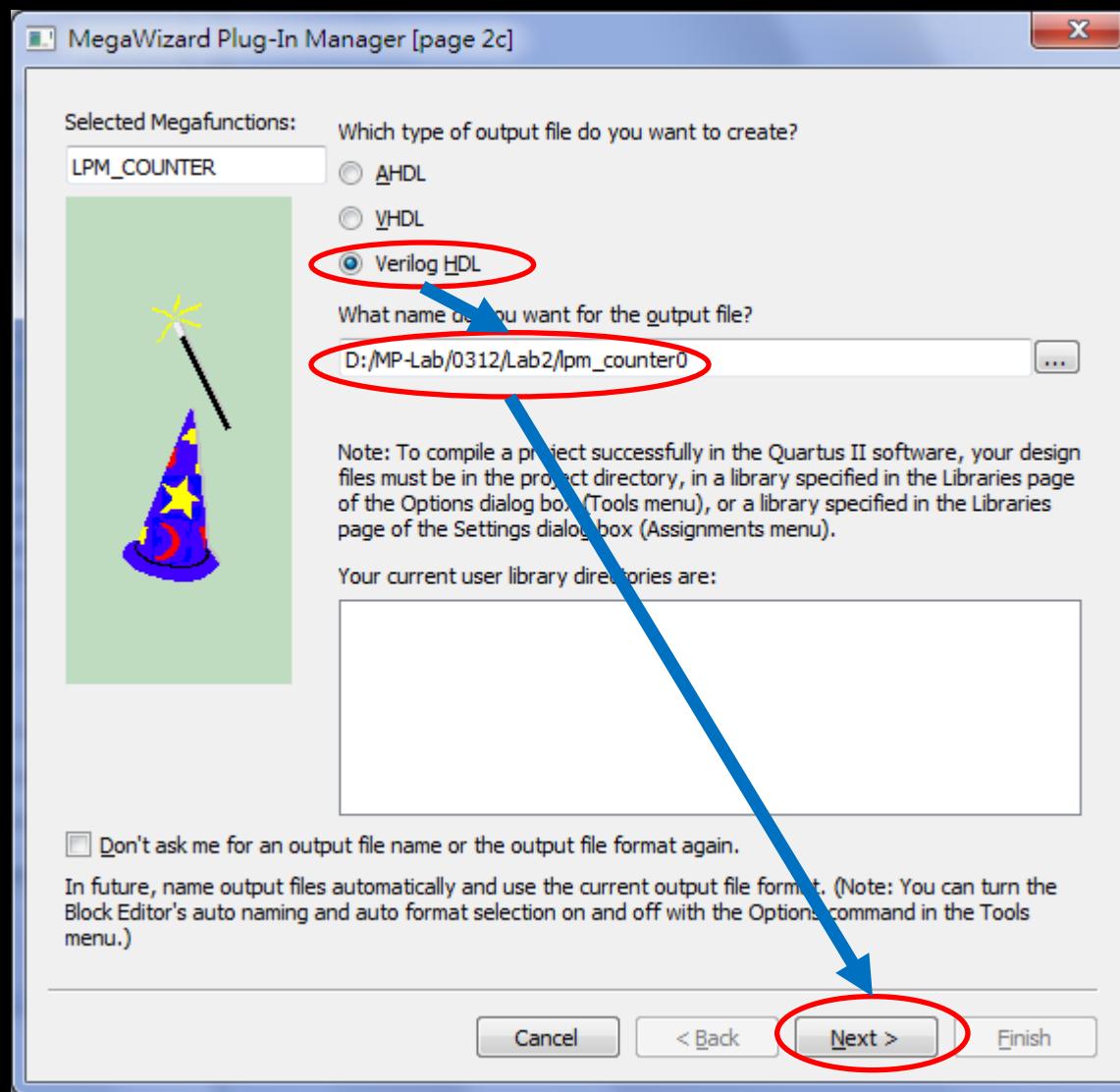


Schematic Design – LPM_COUNTER – 1

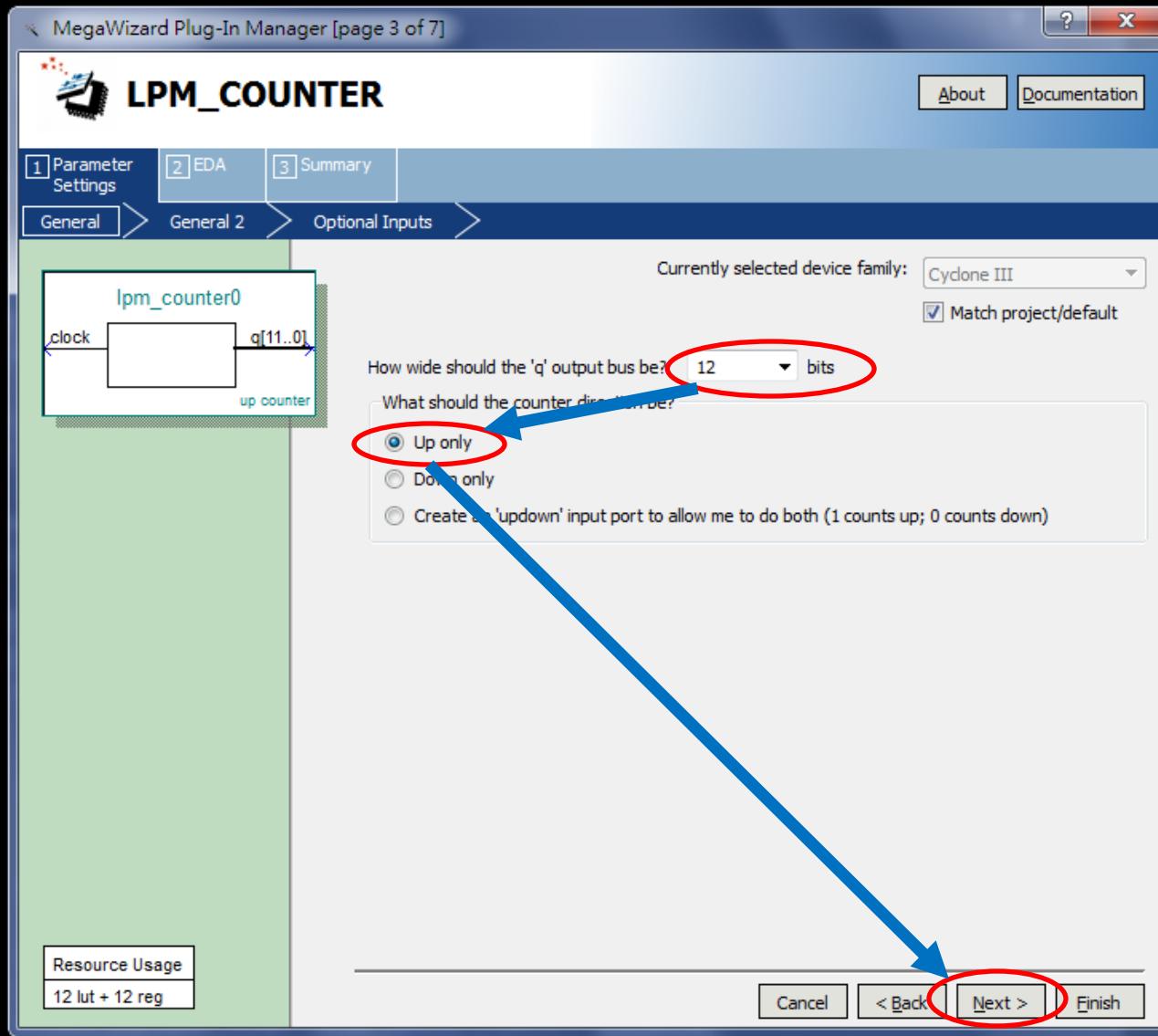
- Symbol Tool -> .../quartus/libraries -> megafuctions -> arithmetic ->lpm_counter



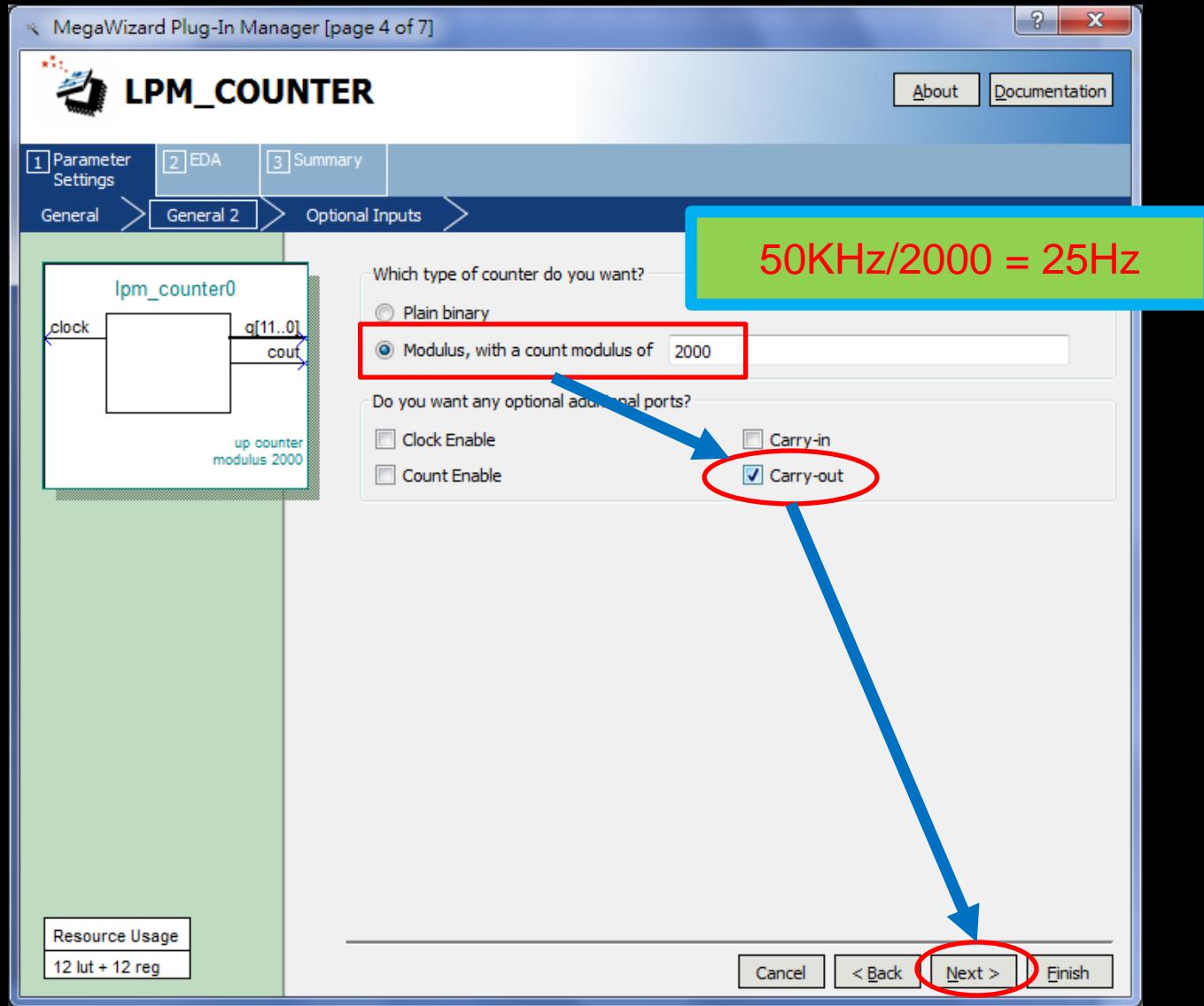
Schematic Design – LPM_COUNTER – 2



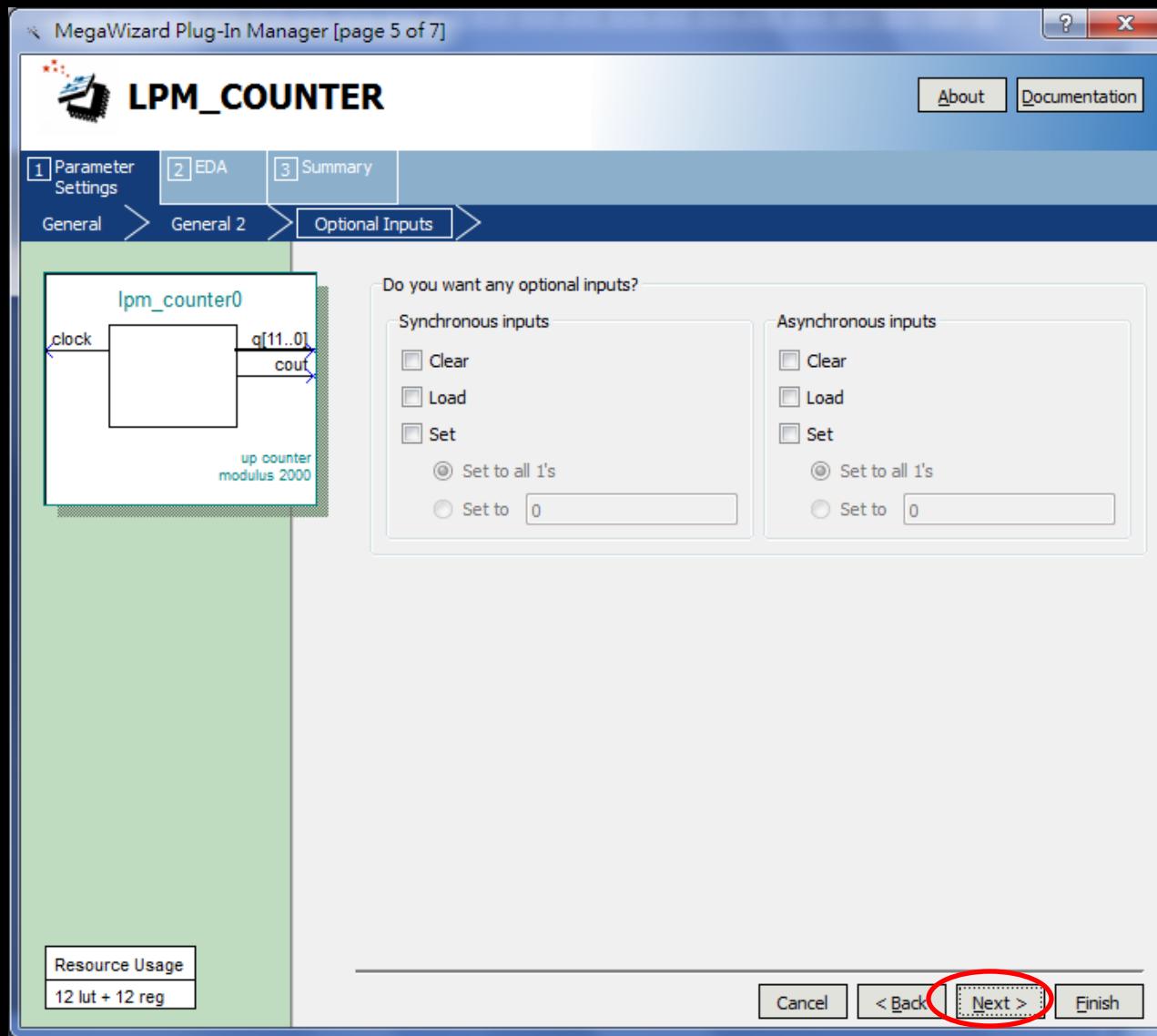
Schematic Design – LPM_COUNTER – 3



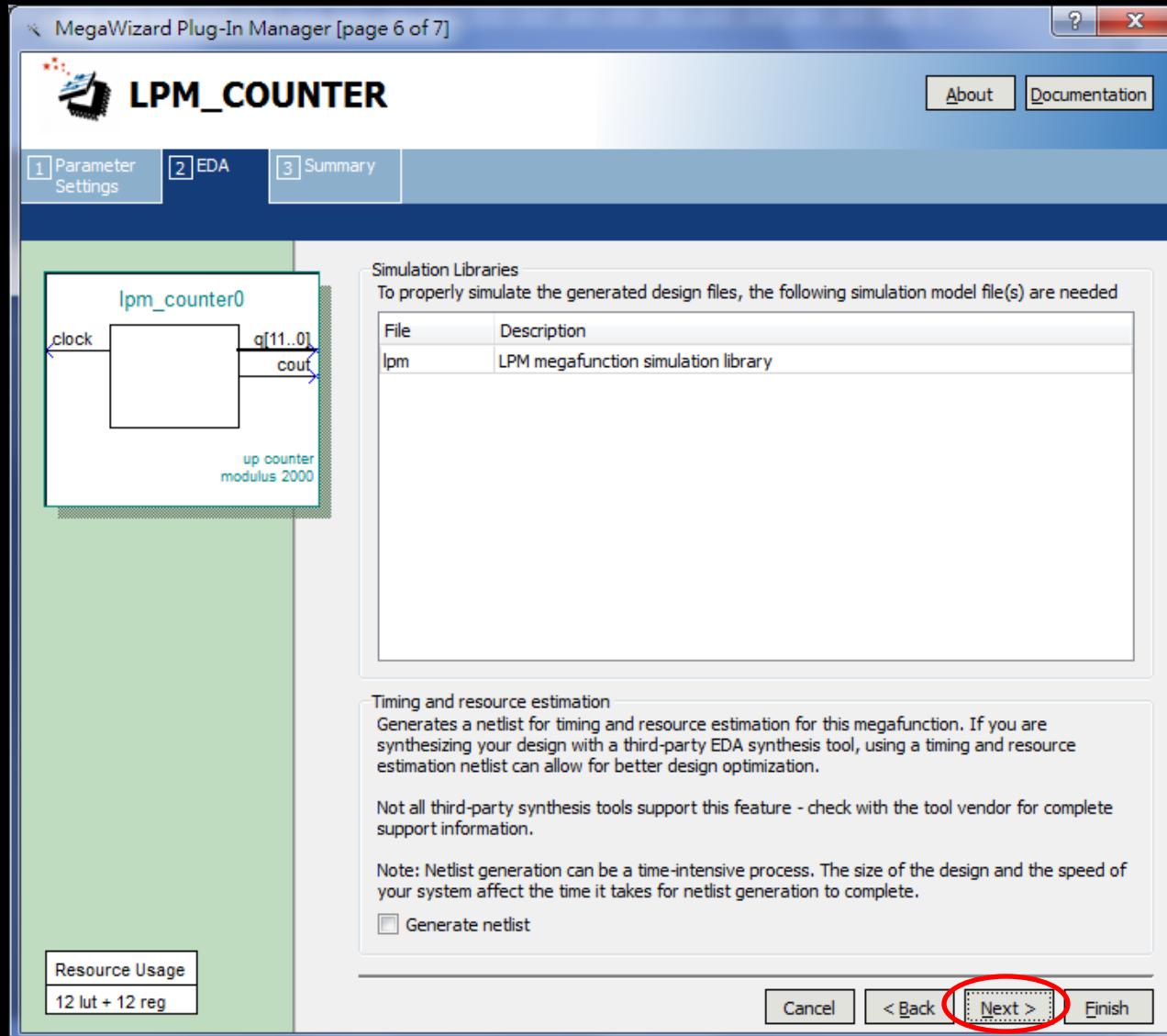
Schematic Design – LPM_COUNTER – 4



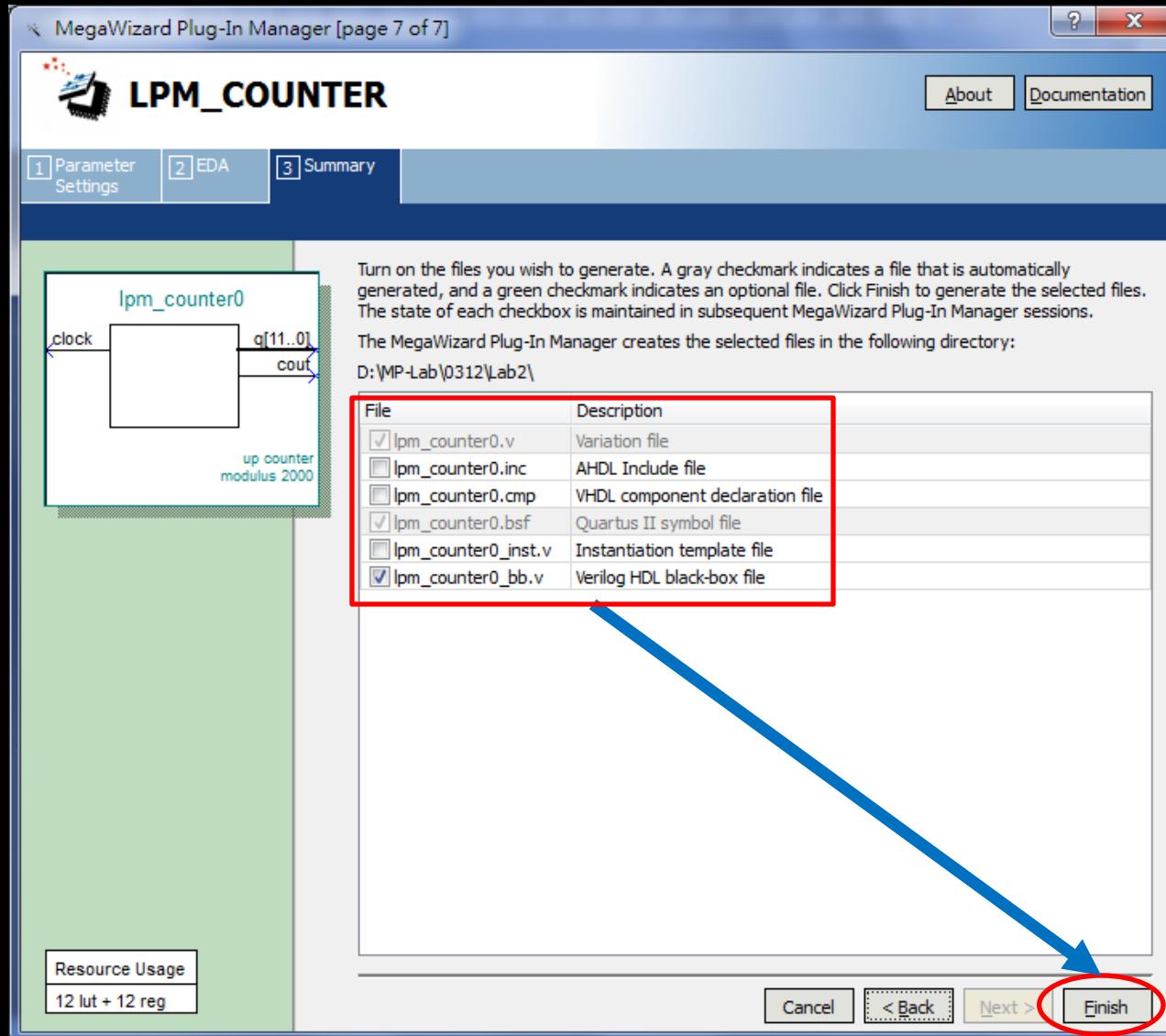
Schematic Design – LPM_COUNTER – 5



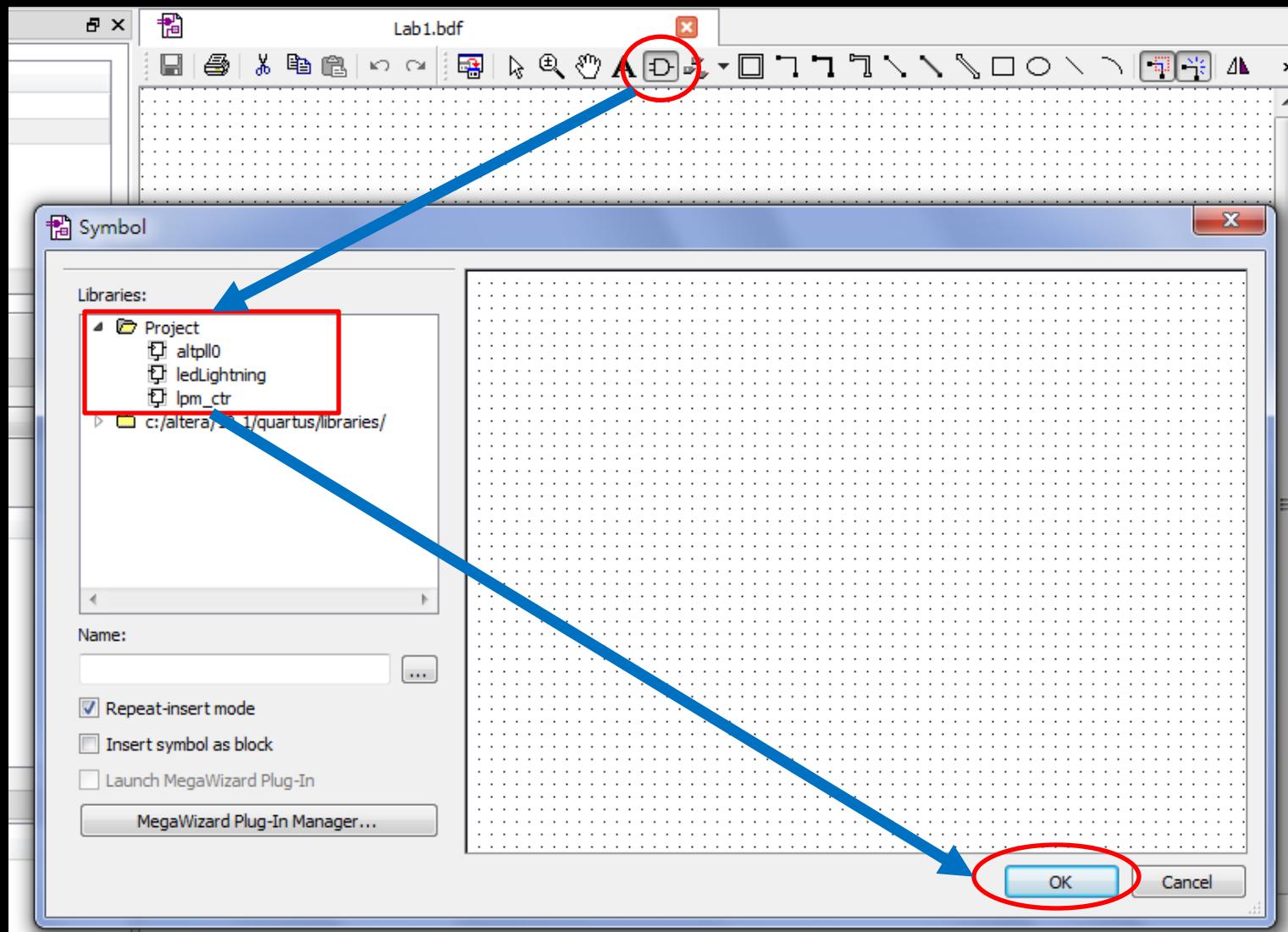
Schematic Design – LPM_COUNTER – 6



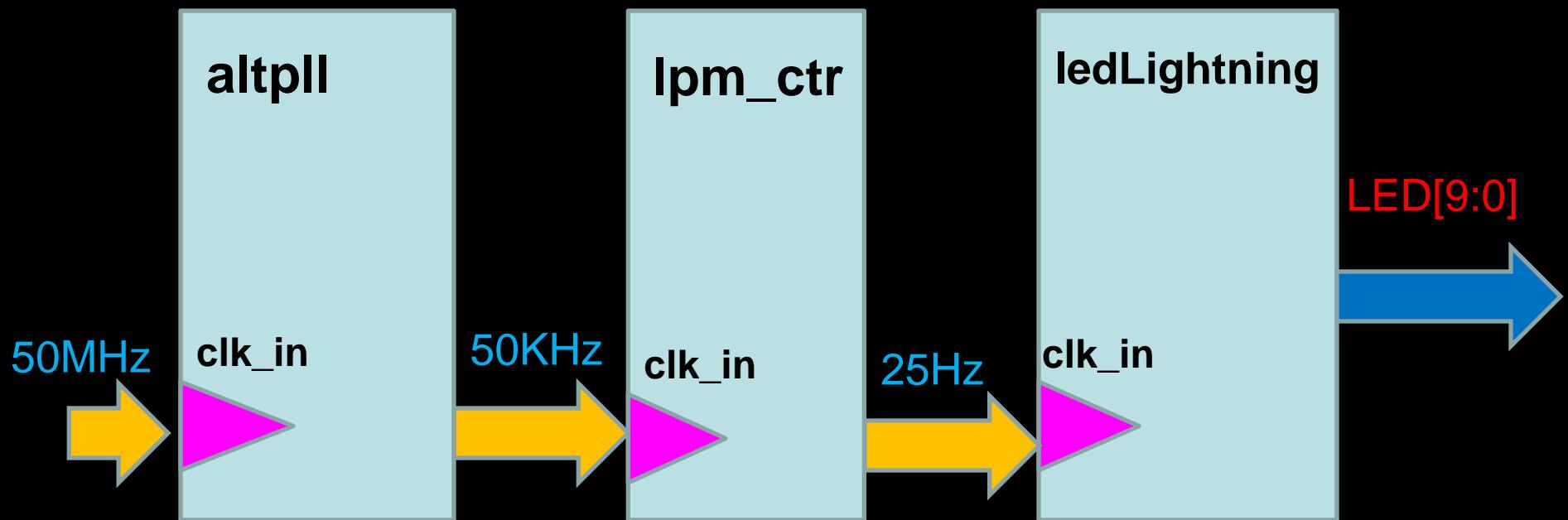
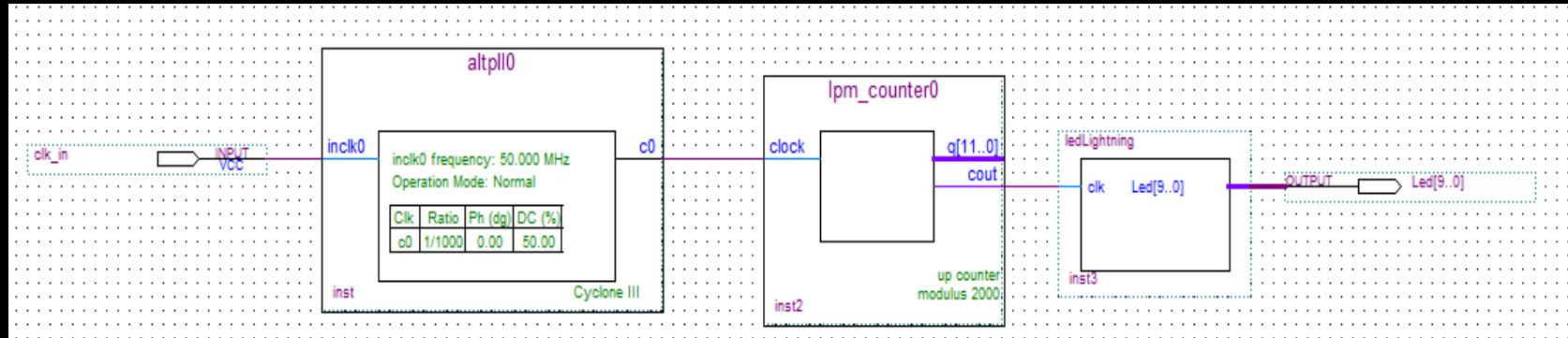
Schematic Design – LPM_COUNTER – 7



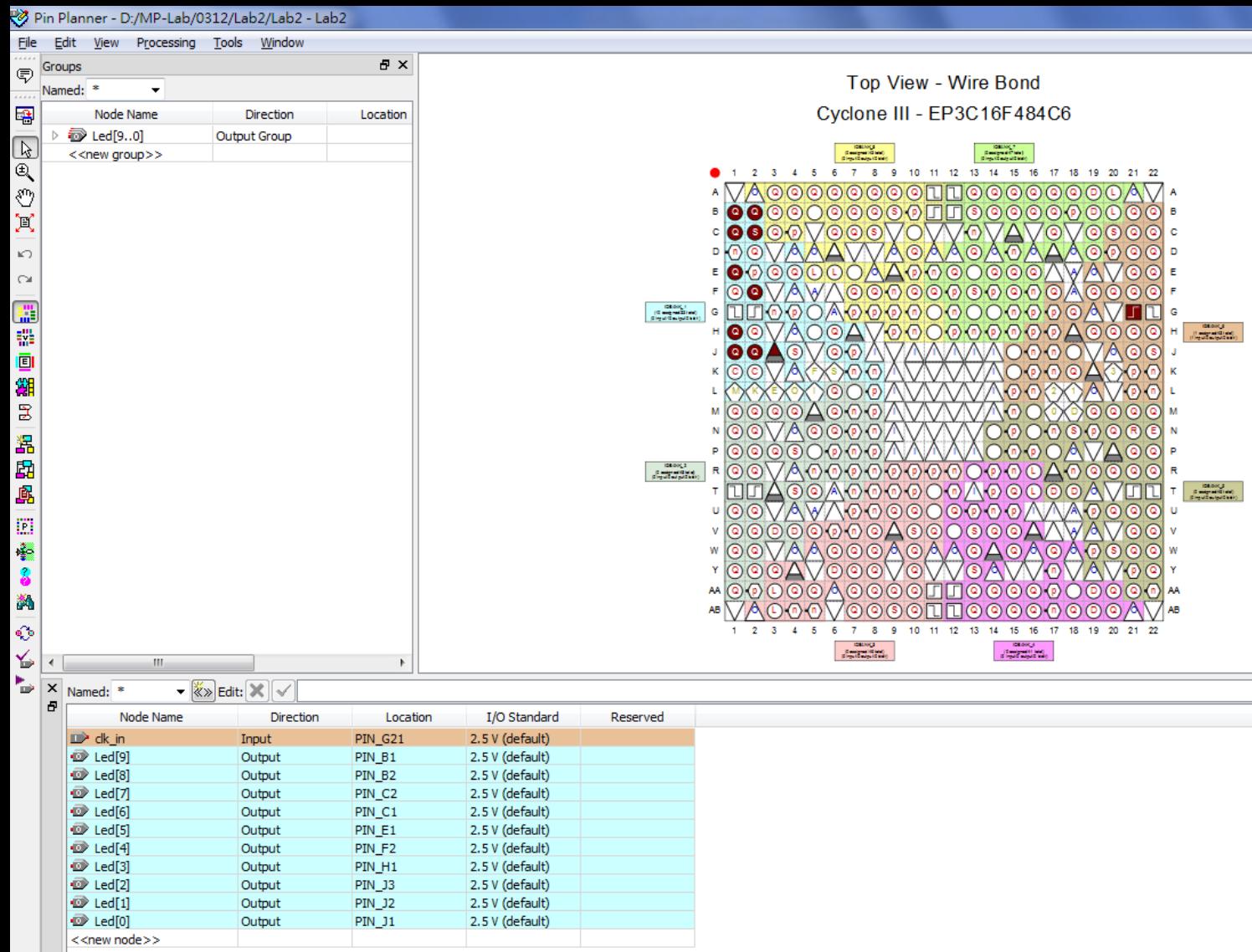
Schematic Design-1



Schematic Design-2



Pin Assignment

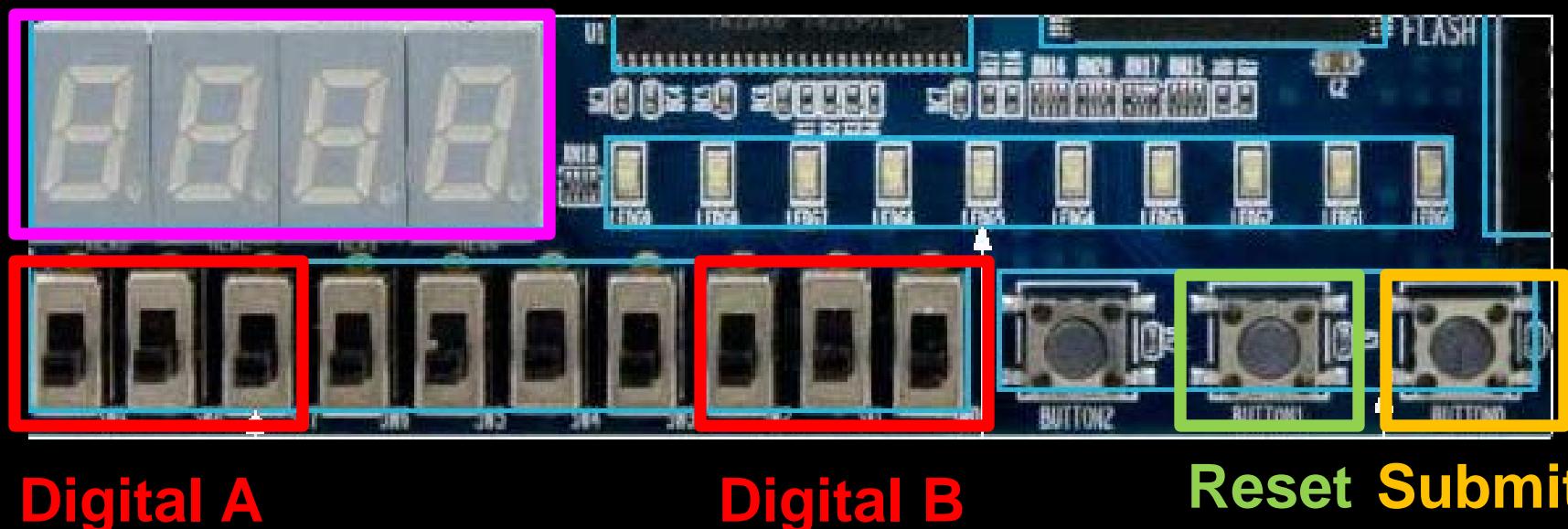


Outline

- **Sequential Circuits**
 - Lab1 Lightning LED
 - Lab2 Lightning LED with Megafunction
- **More Megafuctions**
 - Lab3 MAC (Multiplier-accumulator)

Multiplier-accumulator

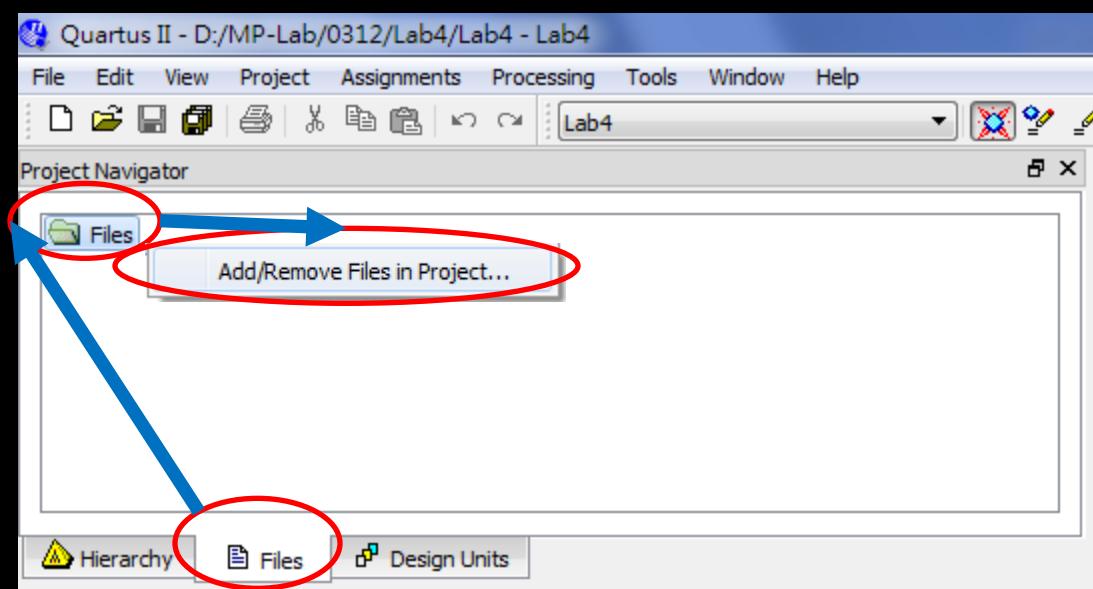
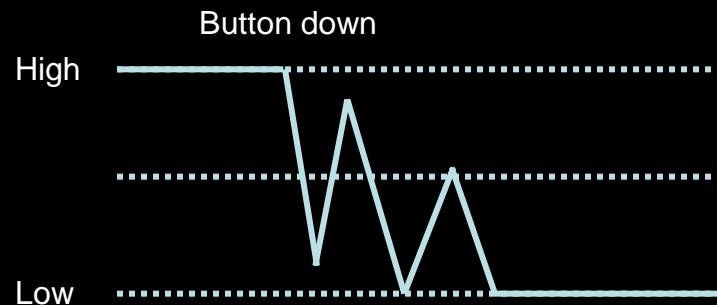
$\text{SUM} = \text{A} * \text{B} + \text{SUM}$



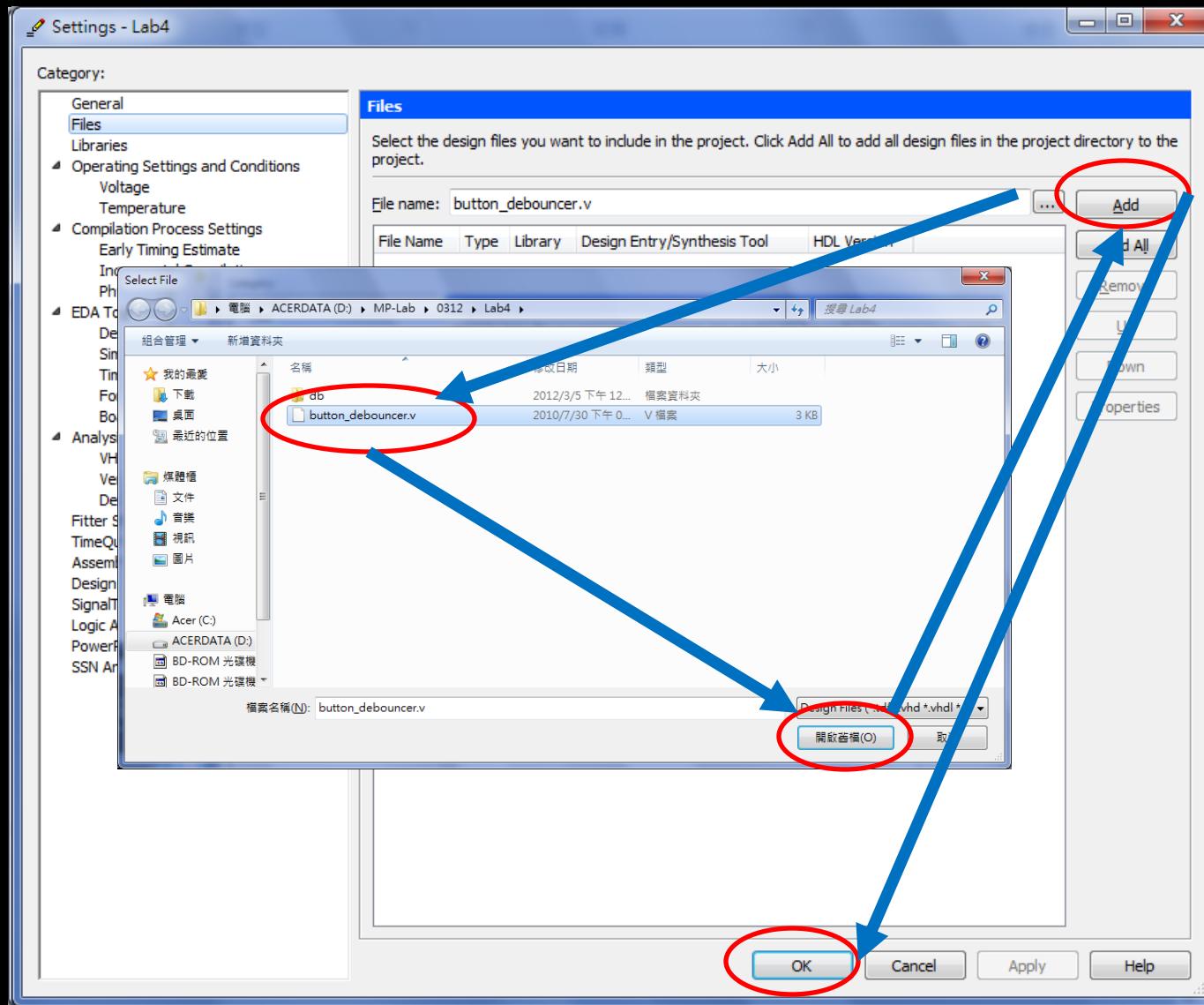
Lab3 Design Flow

- **Create a new project “Lab3”**
- **Create import a Verilog file “button_debounce.v”**
- **Create a new Verilog file “Bin2Dec.v”**
- **Create a new Verilog file “Overflow_Seg7.v”**
- **Create three new symbols from button_debounce.v, Bin2Dec.v and Overflow_Seg7.v**
- **Schematic Design**
 - Megafunction ALTMULT_ACCUM(MAC)
 - Megafunction LPM_MUX
- **Pin Assignment**
- **Compile Full Design**

Module1 button_debounce - 1



Module1 button_debounce - 2



Module2 Bin2Dec

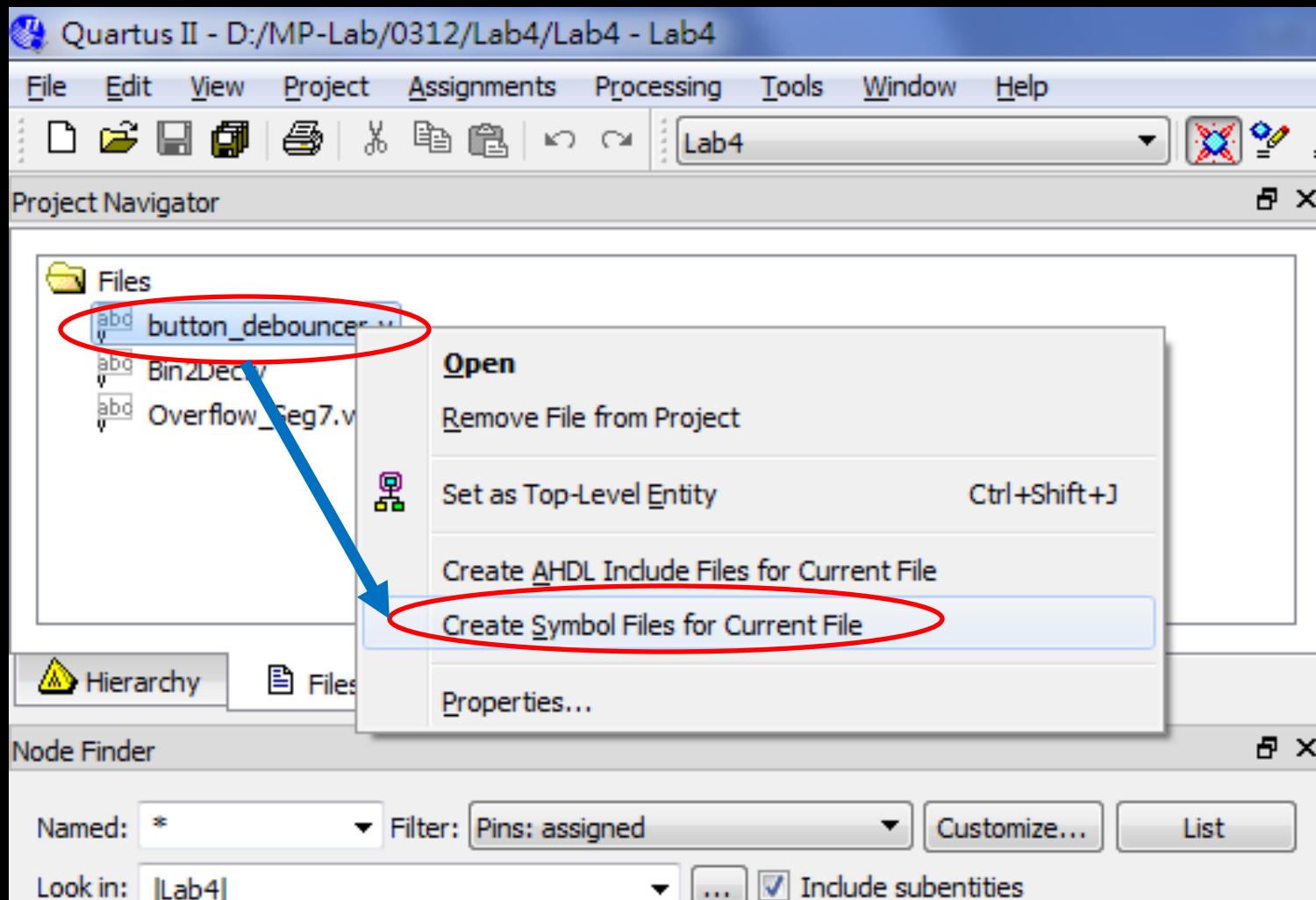
```
1  module Bin2Dec(Sw,Seg7);
2  |  input [8:0] Sw;
3  |  output [31:0] Seg7;
4  |  reg [31:0] Seg7;
5  |  always@(Sw)
6  |  begin
7  |     Seg7 = cal_Bin2Dec(Sw);
8  |  end
9  |
10 | function [31:0] cal_Bin2Dec;
11 |   input [8:0] Sw;
12 | begin
13 |   cal_Bin2Dec = numDecode(Sw);
14 | end
15 | endfunction
16 |
17 | function [31:0] numDecode;
18 |   input [8:0] Sw;
19 |   integer num;
20 |   integer n0, n1,n2,n3;
21 | begin
22 |   num = Sw;
23 |   n0 = num/1000;
24 |   n1 = (num%1000)/100;
25 |   n2 = (num%100)/10;
26 |   n3 = (num%10);
27 |   numDecode[31:24] = seg7Decode(n0);
28 |   numDecode[23:16] = seg7Decode(n1);
29 |   numDecode[15:8] = seg7Decode(n2);
30 |   numDecode[7:0] =seg7Decode(n3);
31 | end
32 | endfunction
33 |
```

```
34 | function [7:0] seg7Decode;
35 |   input [31:0] num;
36 | begin
37 |   case(num)
38 |     0 : seg7Decode = 8'b11000000; // 0
39 |     1 : seg7Decode = 8'b11111001; // 1
40 |     2 : seg7Decode = 8'b10100100; // 2
41 |     3 : seg7Decode = 8'b10110000; // 3
42 |     4 : seg7Decode = 8'b10011001; // 4
43 |     5 : seg7Decode = 8'b10010010; // 5
44 |     6 : seg7Decode = 8'b10000010; // 6
45 |     7 : seg7Decode = 8'b11011000; // 7
46 |     8 : seg7Decode = 8'b10000000; // 8
47 |     9 : seg7Decode = 8'b10010000; // 9
48 |   default :
49 |     seg7Decode = 8'b11111111;
50 |   endcase
51 | end
52 | endfunction
53 |
54 | endmodule
55 |
```

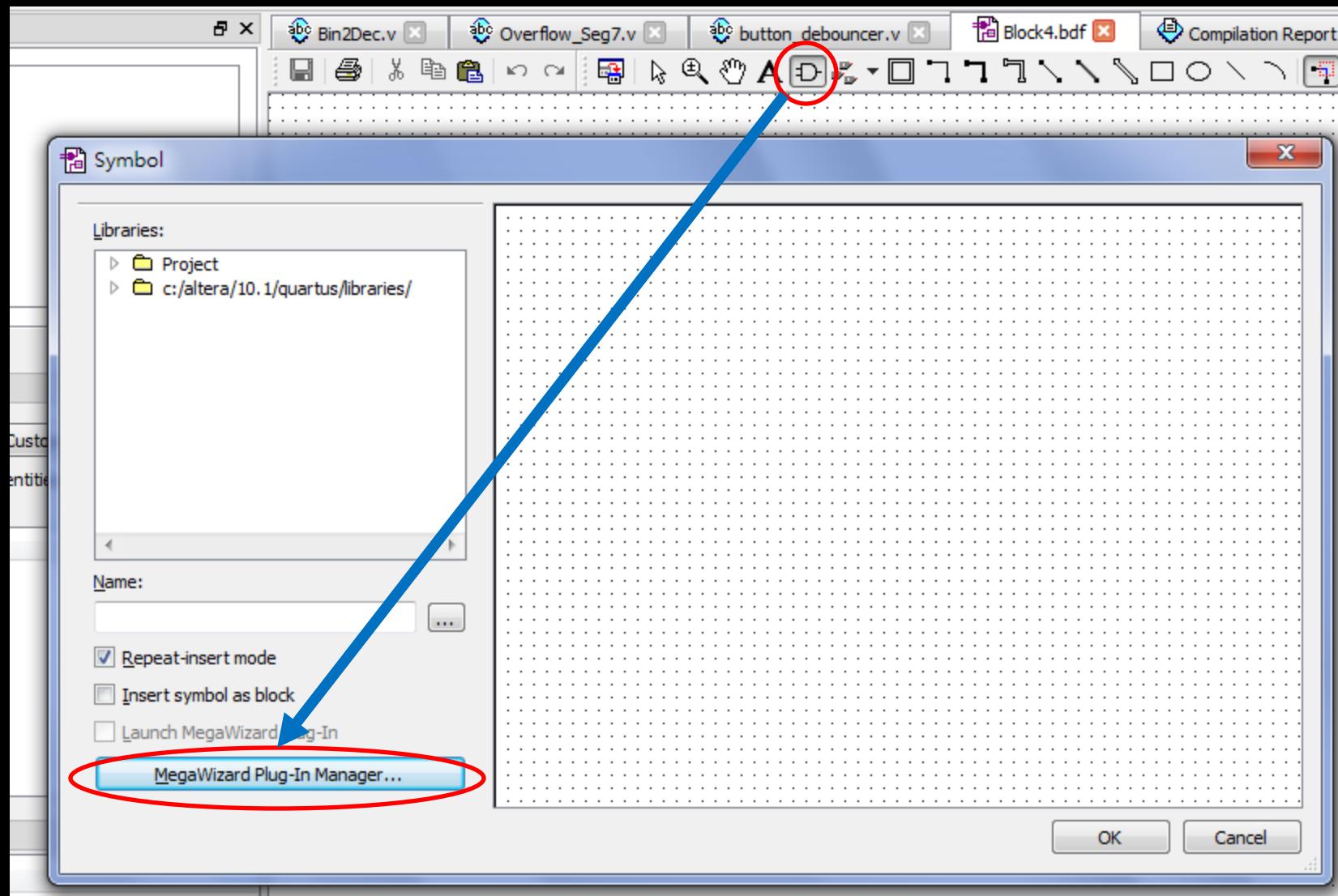
Module3 Overflow_Seg7

```
1  module Overflow_Seg7(Seg7);
2      output [31:0] Seg7;
3      reg [31:0] Seg7;
4      initial
5          begin
6              Seg7[31:24] = 8'b10001110; //F
7              Seg7[23:16] = 8'b10001110; //F
8              Seg7[15:8] = 8'b10001110; //F
9              Seg7[7:0] = 8'b10001110; //F
10         end
11     endmodule
12
```

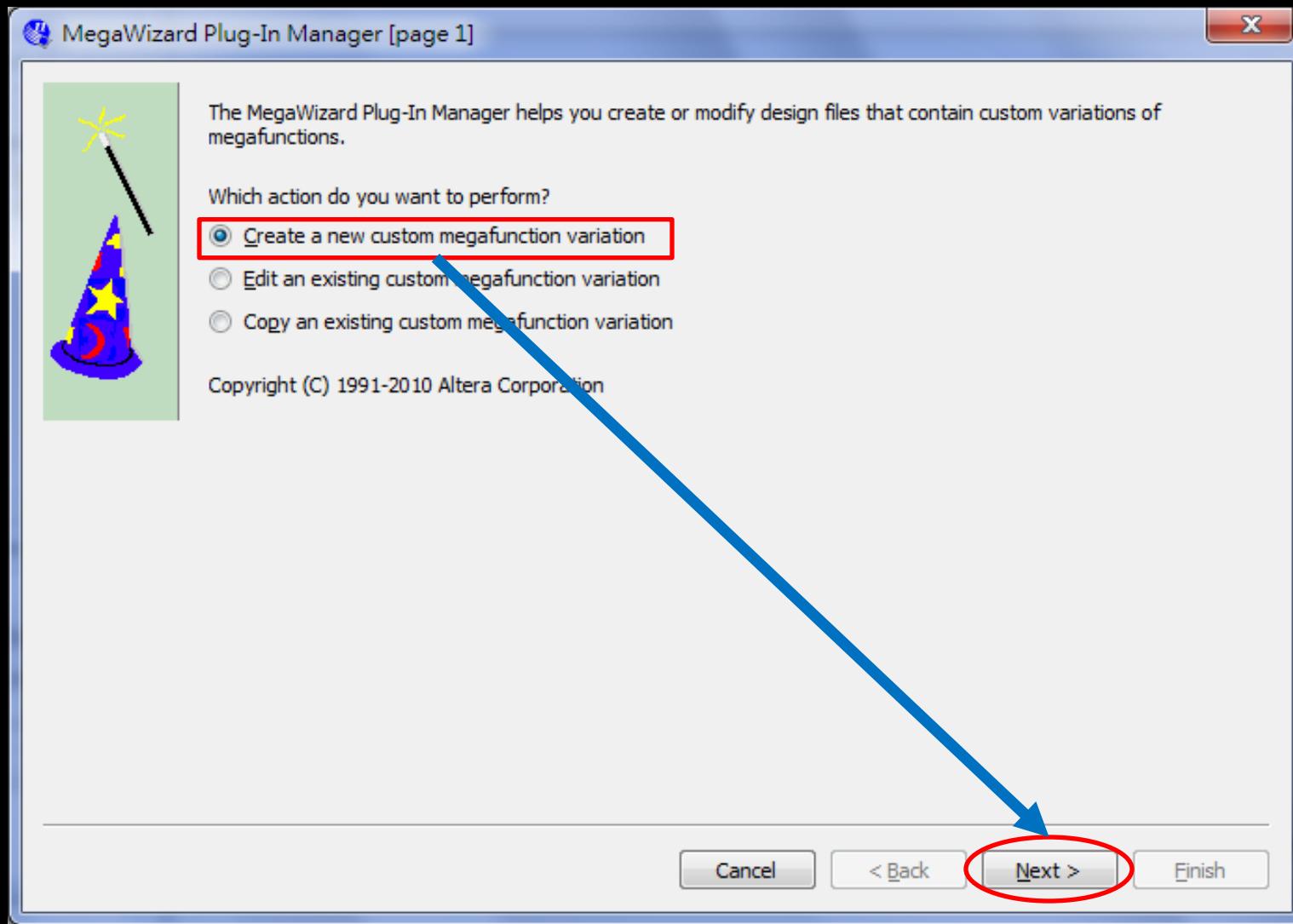
Create New Symbol



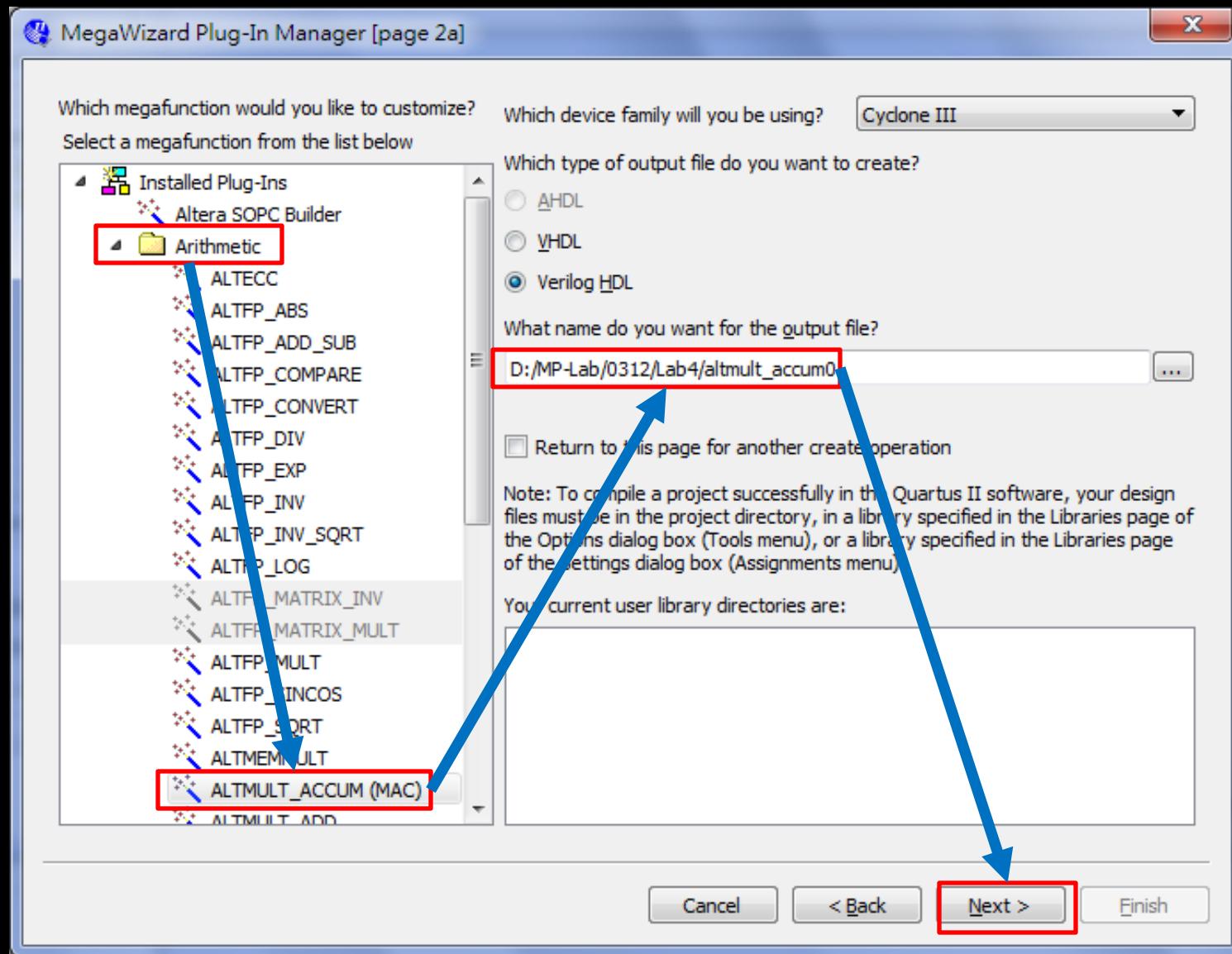
Schematic Design – MAC - 1



Schematic Design – MAC - 2



Schematic Design – MAC - 3



Schematic Design – MAC - 4

MegaWizard Plug-In Manager [page 3 of 8]

ALTMULT_ACCUM

Parameter Settings EDA Summary

General > Extra Modes > Multipliers > Accumulator >

Currently selected device family: Cyclone III
Match project/default

General

What is the number of multipliers? 1 multipliers

All multipliers have similar configurations

How wide should the A input buses be? 3 bits

How wide should the B input buses be? 3 bits

How wide should the 'result' output bus be? 9 bits

Create a 4th asynchronous clear input
This forces all registers to have an associated asynchronous clear input

Create an associated clock enable for each clock

Input Representation

What is the representation format for A inputs? Unsigned
More Options...

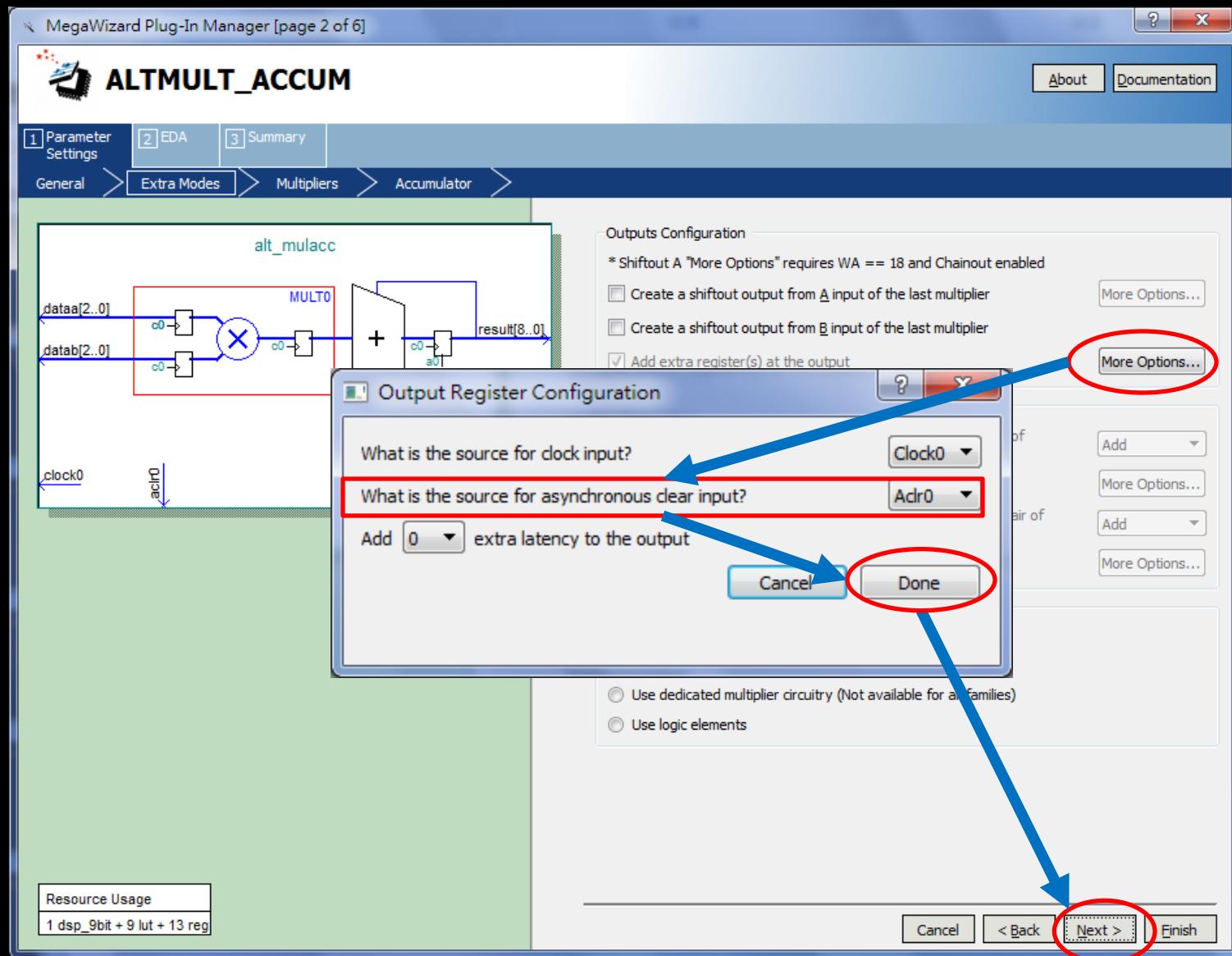
What is the representation format for B inputs? Unsigned
More Options...

Resource Usage
1 dsp_9bit + 9 lut + 9 reg

Cancel < Back Next > Finish

A red oval highlights the width settings for the A, B, and result buses. A blue arrow points from the bottom right towards the 'Next >' button.

Schematic Design – MAC - 5



Schematic Design – MAC - 6

MegaWizard Plug-In Manager [page 5 of 8]

ALTMULT_ACCUM

1 Parameter Settings 2 EDA 3 Summary

General > Extra Modes > Multipliers > Accumulator >

altnmult_accum0

```
graph LR; A[dataaa[2..0]] --> B(( )); B --> C(( )); C --> D[dataab[2..0]]; D --> E(( )); E --> F(( )); F --> G[+]; G --> H[result[8..0]]; G --> I(clock0);
```

dataaa[2..0] dataab[2..0] result[8..0] clock0

dataaa: Unsigned dataab: Unsigned

ALL MULTIPLIERS

Input Configuration

Register input A of the multiplier More Options...

Register input B of the multiplier More Options...

What is the input A of the multiplier connected to? Multiplier input ▾

Use the same signal to control the source for input A of all multipliers

What is the input B of the multiplier connected to? Multiplier input ▾

Use the same signal to control the source for input B of all multipliers

Output Configuration

Register output of the multiplier More Options...

Resource Usage

1 dsp_9bit + 9 lut + 9 reg

Cancel < Back **Next >** Finish

Schematic Design – MAC - 7

MegaWizard Plug-In Manager [page 6 of 8]

ALTMULT_ACCUM

1 Parameter Settings 2 EDA 3 Summary

General > Extra Modes > Multipliers > Accumulator

altmult_accum0

Accumulator

Create an 'accum_sload' input port
 Create an 'accum_sload_upper_data' input port of
 Create an 'overflow' output port
More Options...
1 bits
More Options...

Add 0 extra latency to the multiplier output

Resource Usage
1 dsp_9bit + 9 lut + 13 reg

Cancel < Back **Next >** Finish

A red circle highlights the "Create an 'overflow' output port" checkbox. A large blue arrow points from this checkbox down to the "Next >" button at the bottom right.

Schematic Design – MAC - 8

MegaWizard Plug-In Manager [page 7 of 8]

ALTMULT_ACCUM

1 Parameter Settings 2 EDA 3 Summary

Simulation Libraries
To properly simulate the generated design files, the following simulation model file(s) are needed

File	Description
altera_mf	Altera megafunction simulation library

altmult_accum0

overflow

clock0

Resource Usage
1 dsp_9bit + 9 lut + 13 reg

Timing and resource estimation
Generates a netlist for timing and resource estimation for this megafunction. If you are synthesizing your design with a third-party EDA synthesis tool, using a timing and resource estimation netlist can allow for better design optimization.

Not all third-party synthesis tools support this feature - check with the tool vendor for complete support information.

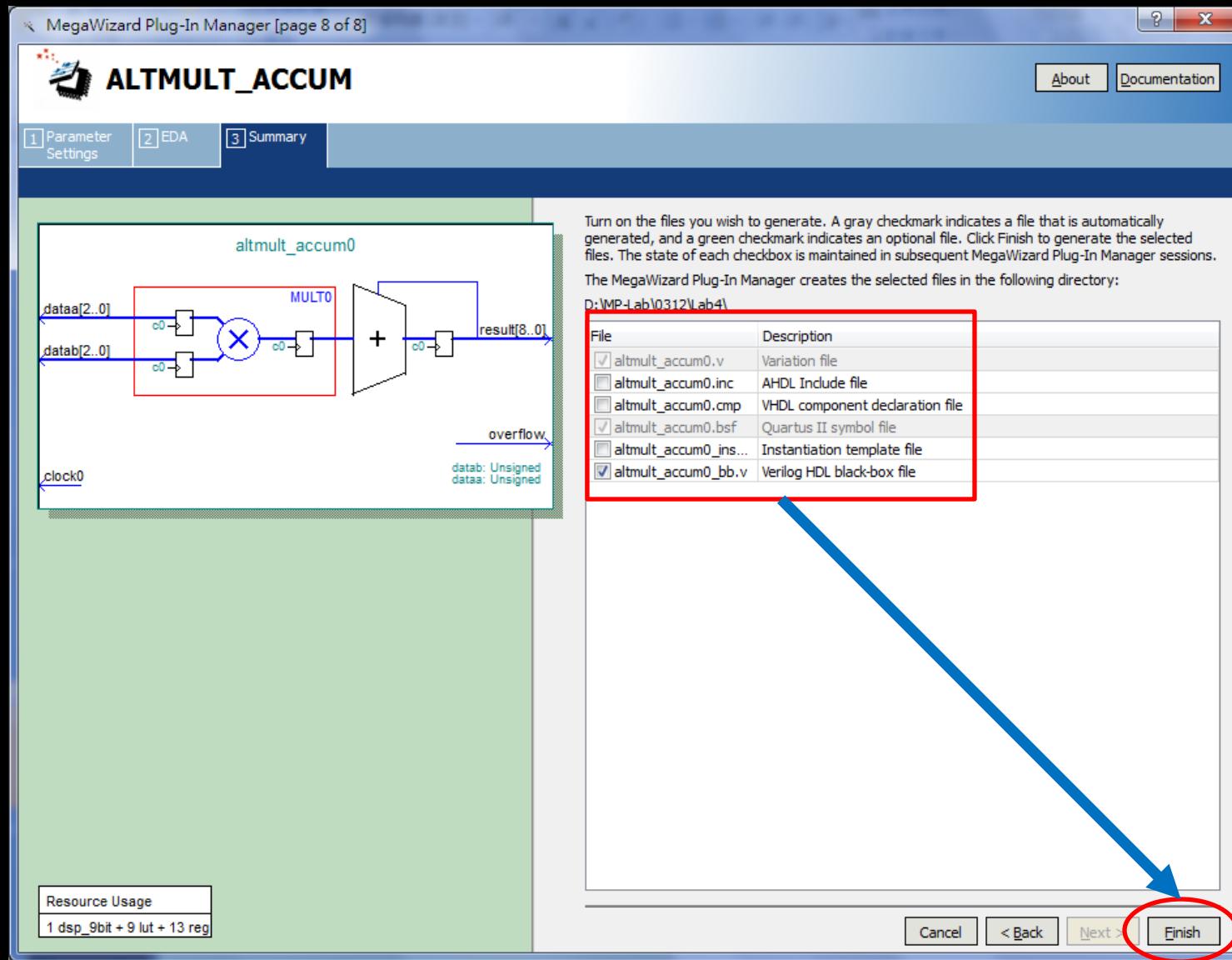
Note: Netlist generation can be a time-intensive process. The size of the design and the speed of your system affect the time it takes for netlist generation to complete.

Generate netlist

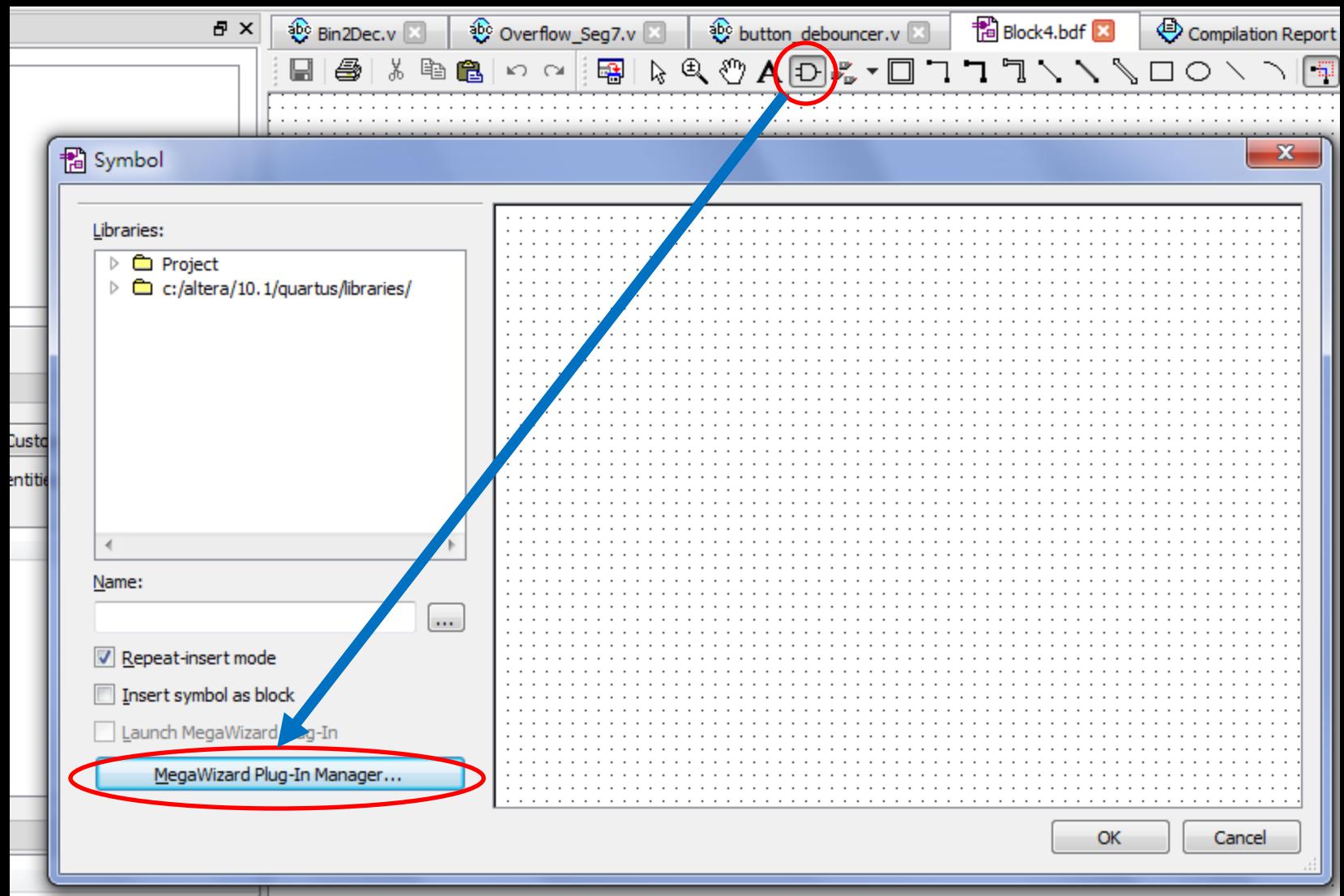
Cancel < Back **Next >** Finish

53/64

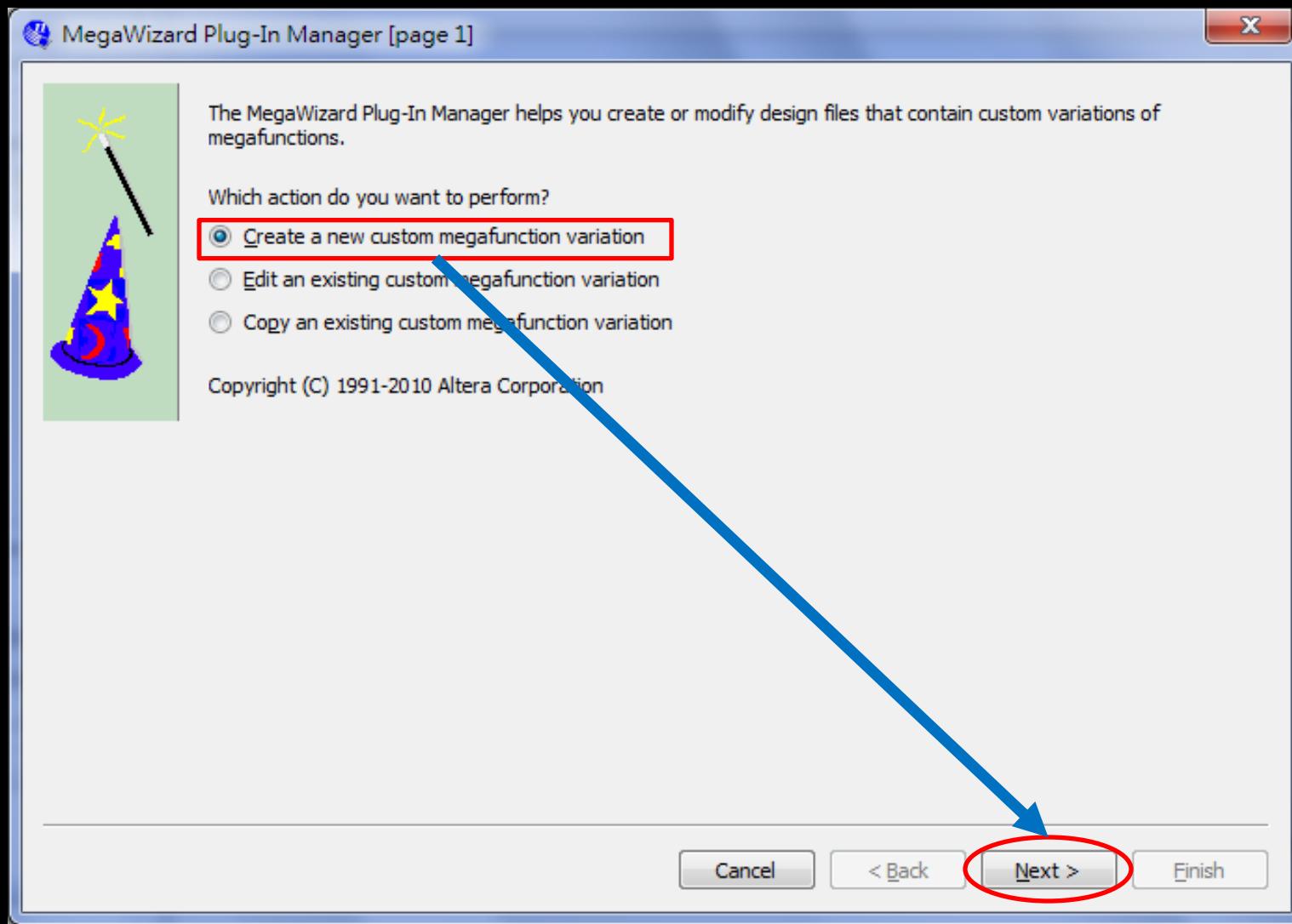
Schematic Design – MAC - 9



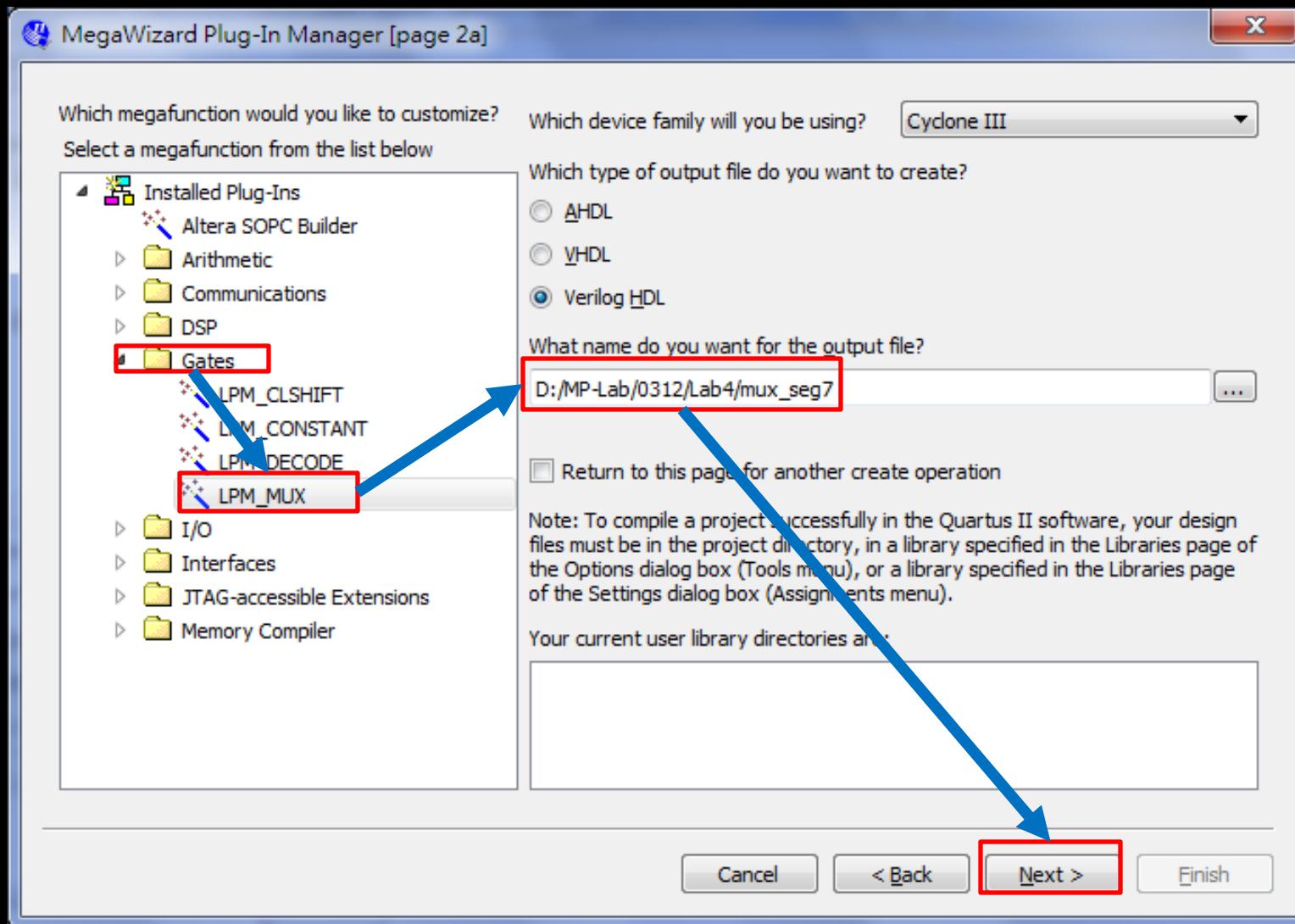
Schematic Design – LPM_MUX - 1



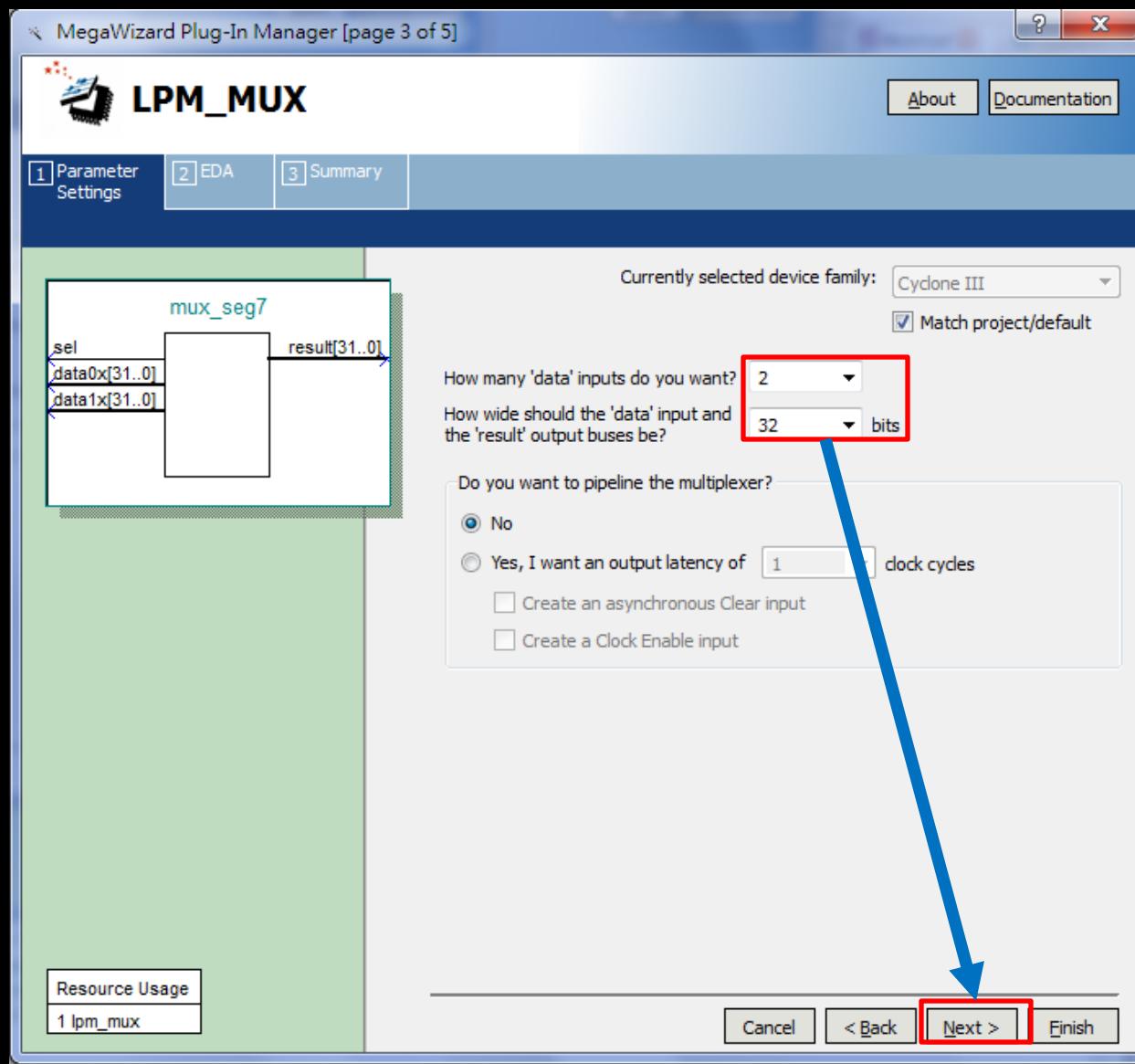
Schematic Design – LPM_MUX - 2



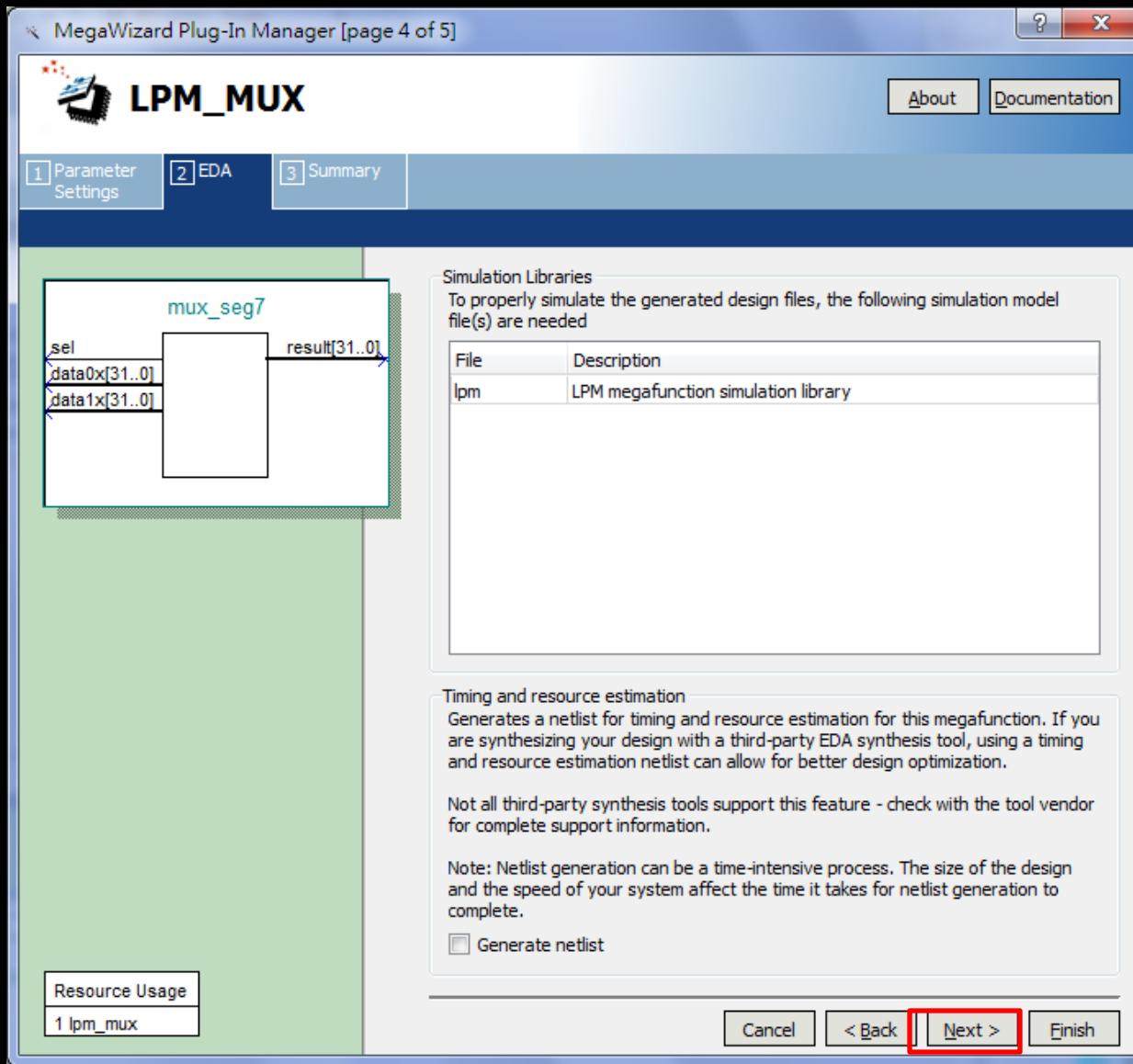
Schematic Design – LPM_MUX - 3



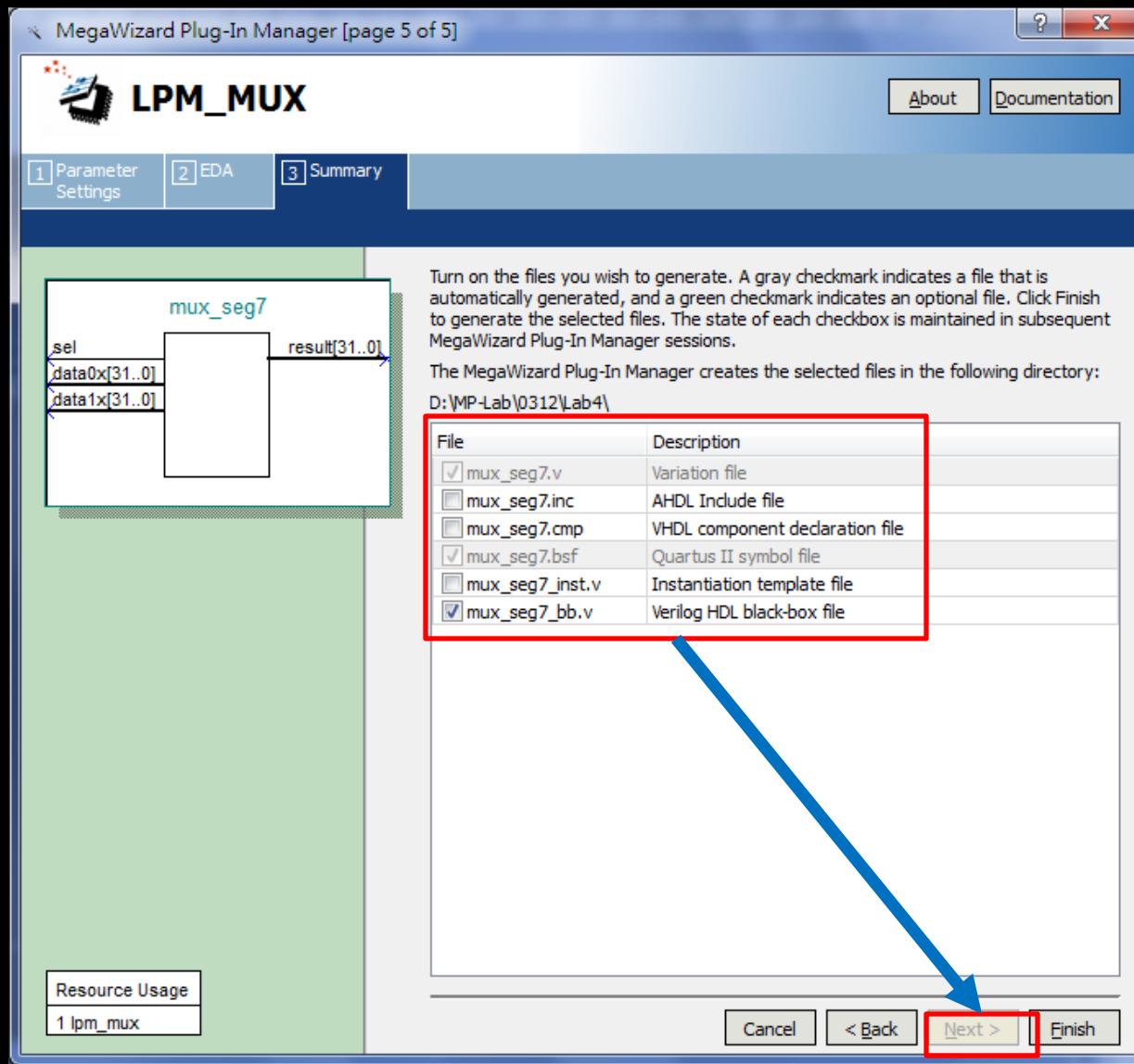
Schematic Design – LPM_MUX - 4



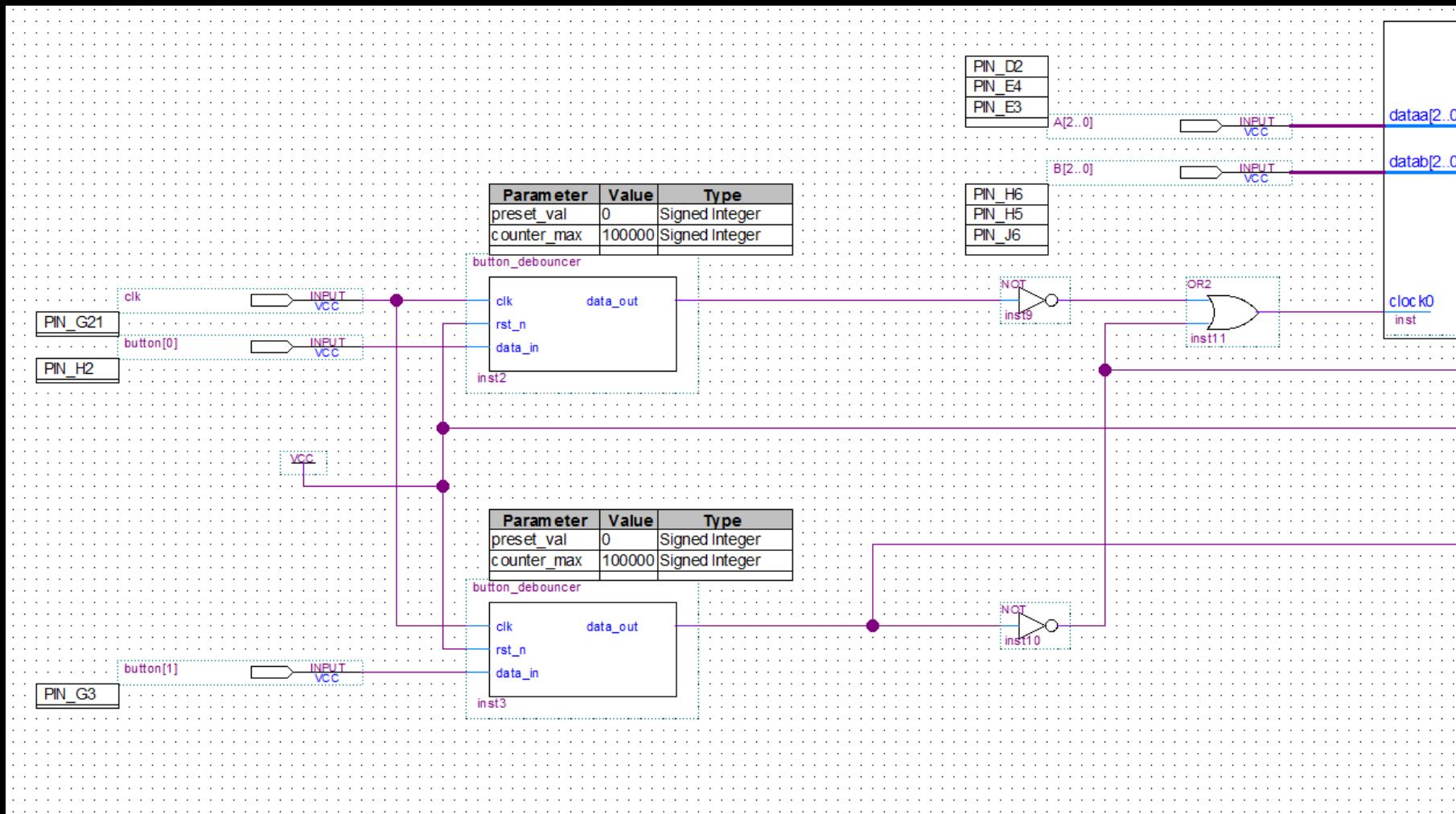
Schematic Design – LPM_MUX - 5



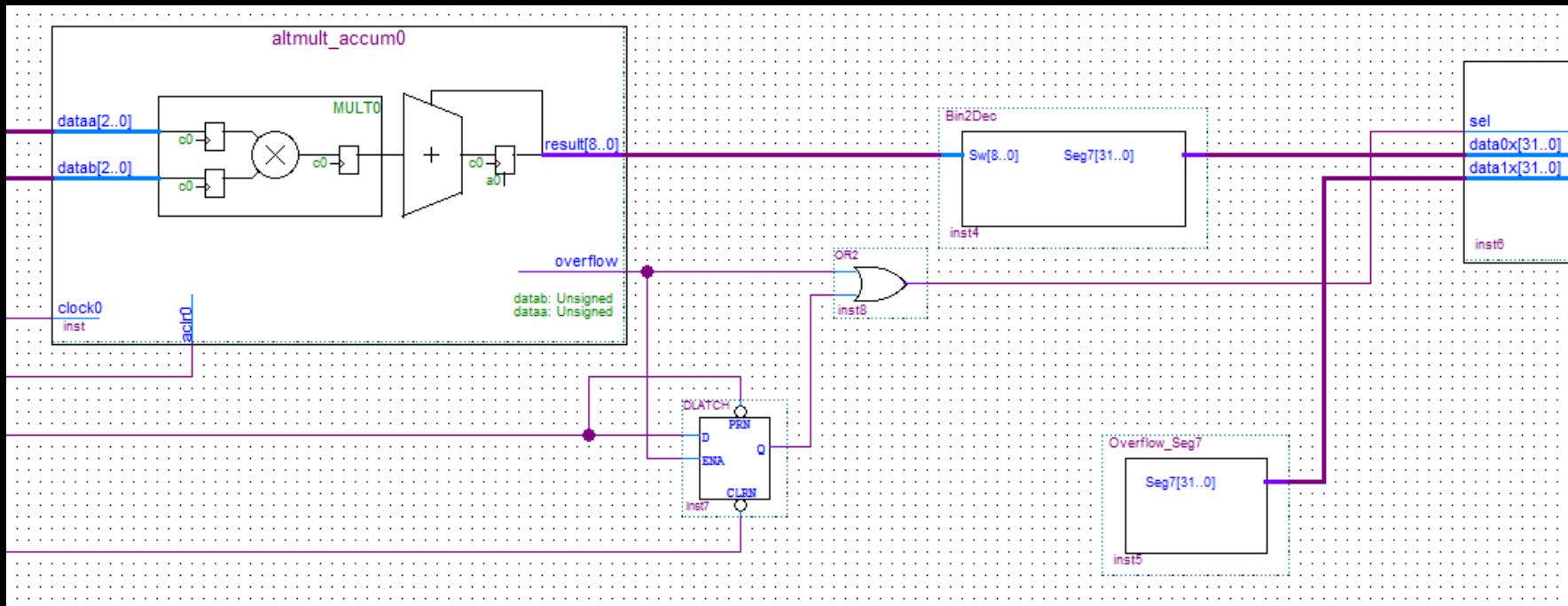
Schematic Design – LPM_MUX - 6



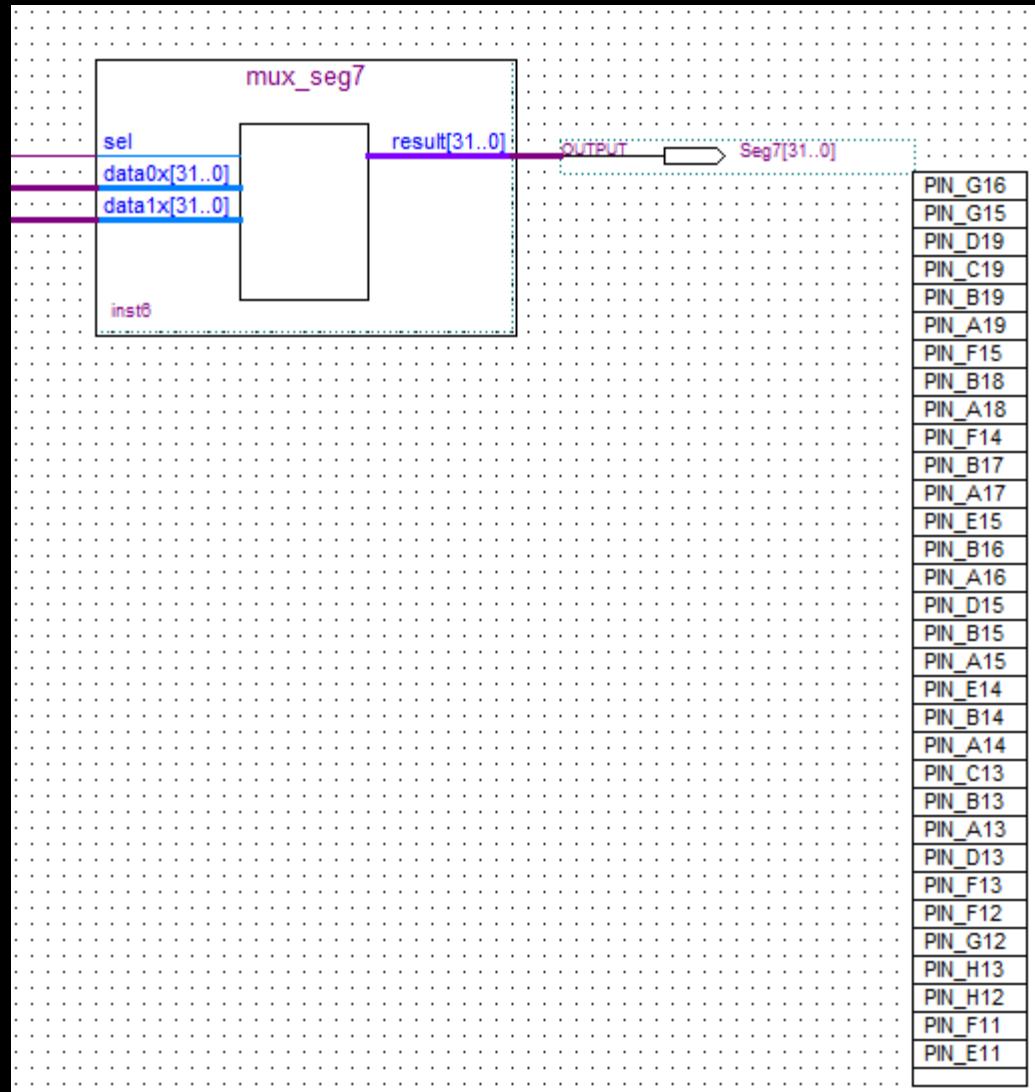
Schematic Design-1



Schematic Design-2



Schematic Design-3



Pin Assignment

Pin Planner - D:/MP-Lab/0312/Lab4/Lab4 - Lab4

File Edit View Processing Tools Window

Groups

Node Name	Direction	Location
A[2..0]	Input Group	
B[2..0]	Input Group	
button[1..0]	Input Group	
Seg7[31..0]	Output Group	
<<new group>>		

Top View - Wire Bond
Cyclone III - EP2C12F484C8

Named: * Edit: X ✓

Node Name	Direction	Location	I/O Standard	Reserved
A[2]	Input	PIN_D2	2.5 V (default)	
A[1]	Input	PIN_E4	2.5 V (default)	
A[0]	Input	PIN_E3	2.5 V (default)	
B[2]	Input	PIN_H6	2.5 V (default)	
B[1]	Input	PIN_H5	2.5 V (default)	
B[0]	Input	PIN_J6	2.5 V (default)	
button[1]	Input	PIN_G3	2.5 V (default)	
button[0]	Input	PIN_H2	2.5 V (default)	
clk	Input	PIN_G21	2.5 V (default)	
Seg7[31]	Output	PIN_G16	2.5 V (default)	
Seg7[30]	Output	PIN_G15	2.5 V (default)	
Seg7[29]	Output	PIN_D19	2.5 V (default)	
Seg7[28]	Output	PIN_C19	2.5 V (default)	
Seg7[27]	Output	PIN_B19	2.5 V (default)	
Seg7[26]	Output	PIN_A19	2.5 V (default)	
Seg7[25]	Output	PIN_F15	2.5 V (default)	
Seg7[24]	Output	PIN_B18	2.5 V (default)	
Seg7[23]	Output	PIN_A18	2.5 V (default)	
Seg7[22]	Output	PIN_F14	2.5 V (default)	
Seg7[21]	Output	PIN_B17	2.5 V (default)	
Seg7[20]	Output	PIN_A17	2.5 V (default)	
Seg7[19]	Output	PIN_E15	2.5 V (default)	
Seg7[18]	Output	PIN_B16	2.5 V (default)	
Seg7[17]	Output	PIN_A16	2.5 V (default)	
Seg7[16]	Output	PIN_D15	2.5 V (default)	
Seg7[15]	Output	PIN_B15	2.5 V (default)	
Seg7[14]	Output	PIN_A15	2.5 V (default)	
Seg7[13]	Output	PIN_E14	2.5 V (default)	
Seg7[12]	Output	PIN_B14	2.5 V (default)	
Seg7[11]	Output	PIN_A14	2.5 V (default)	
Seg7[10]	Output	PIN_C13	2.5 V (default)	
Seg7[9]	Output	PIN_B13	2.5 V (default)	
Seg7[8]	Output	PIN_A13	2.5 V (default)	
Seg7[7]	Output	PIN_D13	2.5 V (default)	
Seg7[6]	Output	PIN_F13	2.5 V (default)	
Seg7[5]	Output	PIN_F12	2.5 V (default)	
Seg7[4]	Output	PIN_G12	2.5 V (default)	
Seg7[3]	Output	PIN_H13	2.5 V (default)	
Seg7[2]	Output	PIN_H12	2.5 V (default)	
Seg7[1]	Output	PIN_F11	2.5 V (default)	
Seg7[0]	Output	PIN_E11	2.5 V (default)	

All Pins <<new node>>

Filter: Pins: all

0% 00:00:00

64/64