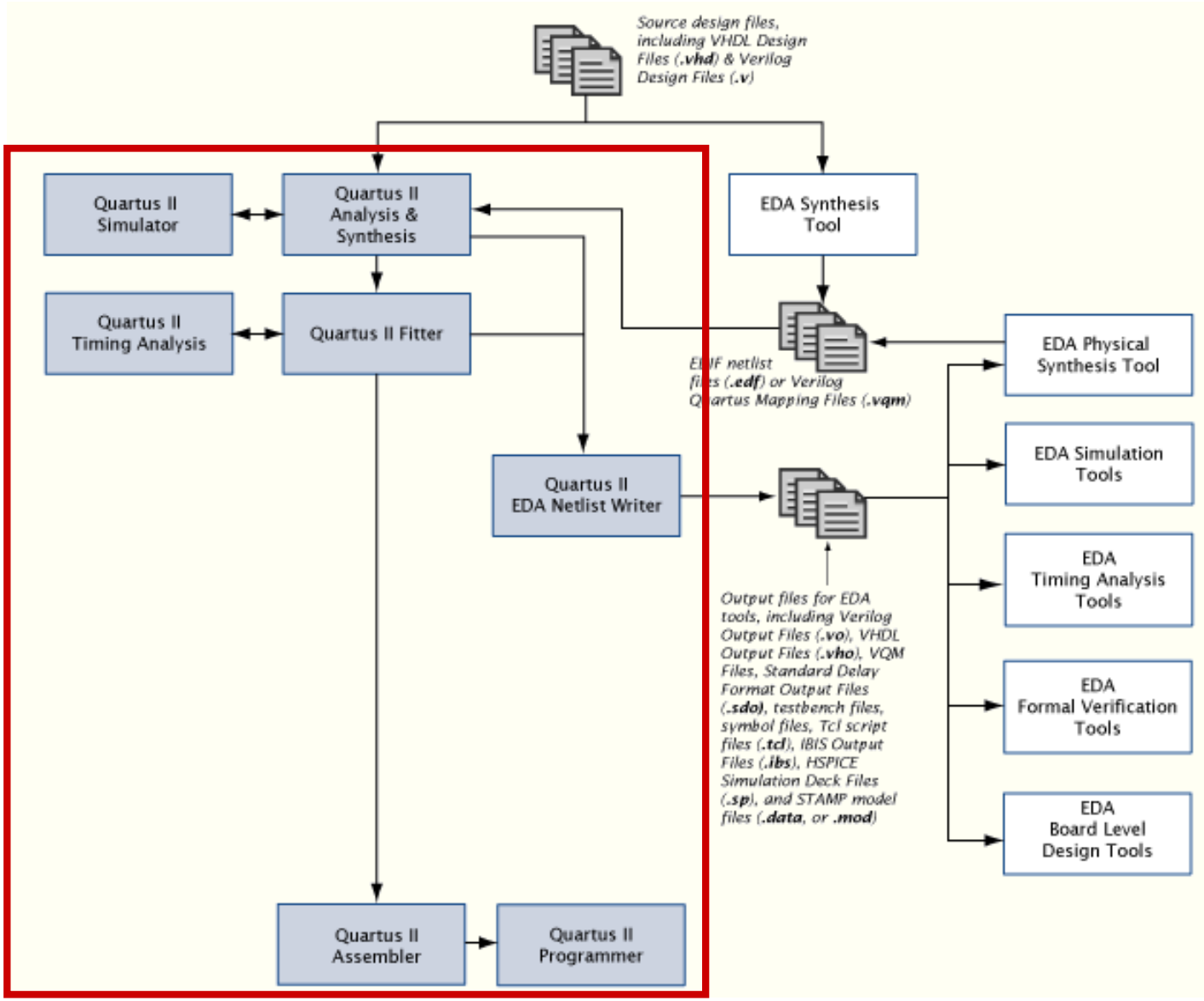
QuartusⅡ & DE2-115

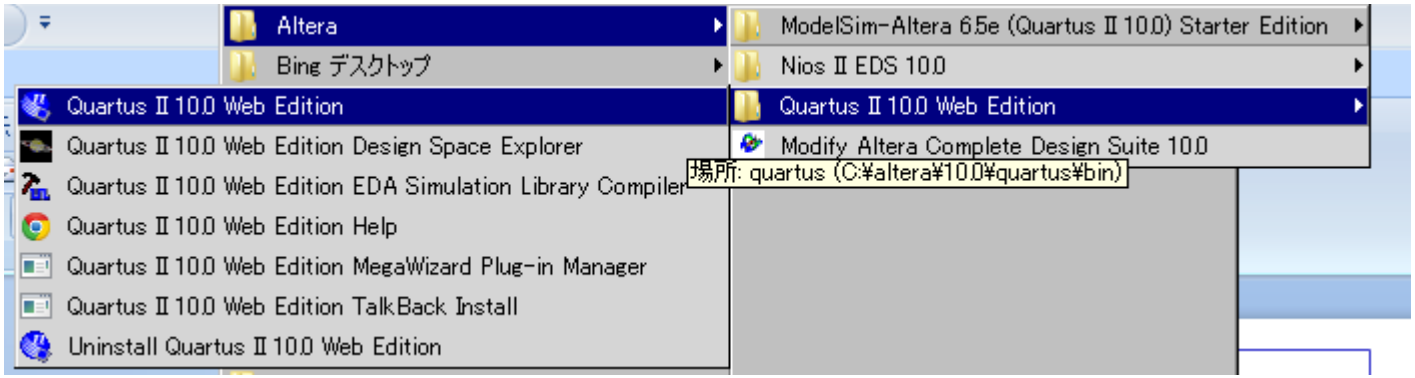
K.Mashiko

EDA Tool Flow



Setup of QuartusII : Start-up

From Start Menu: Start> Program> Altera> Quartus II10.0 Web Edition
> Quartus II10.0 Web Edition

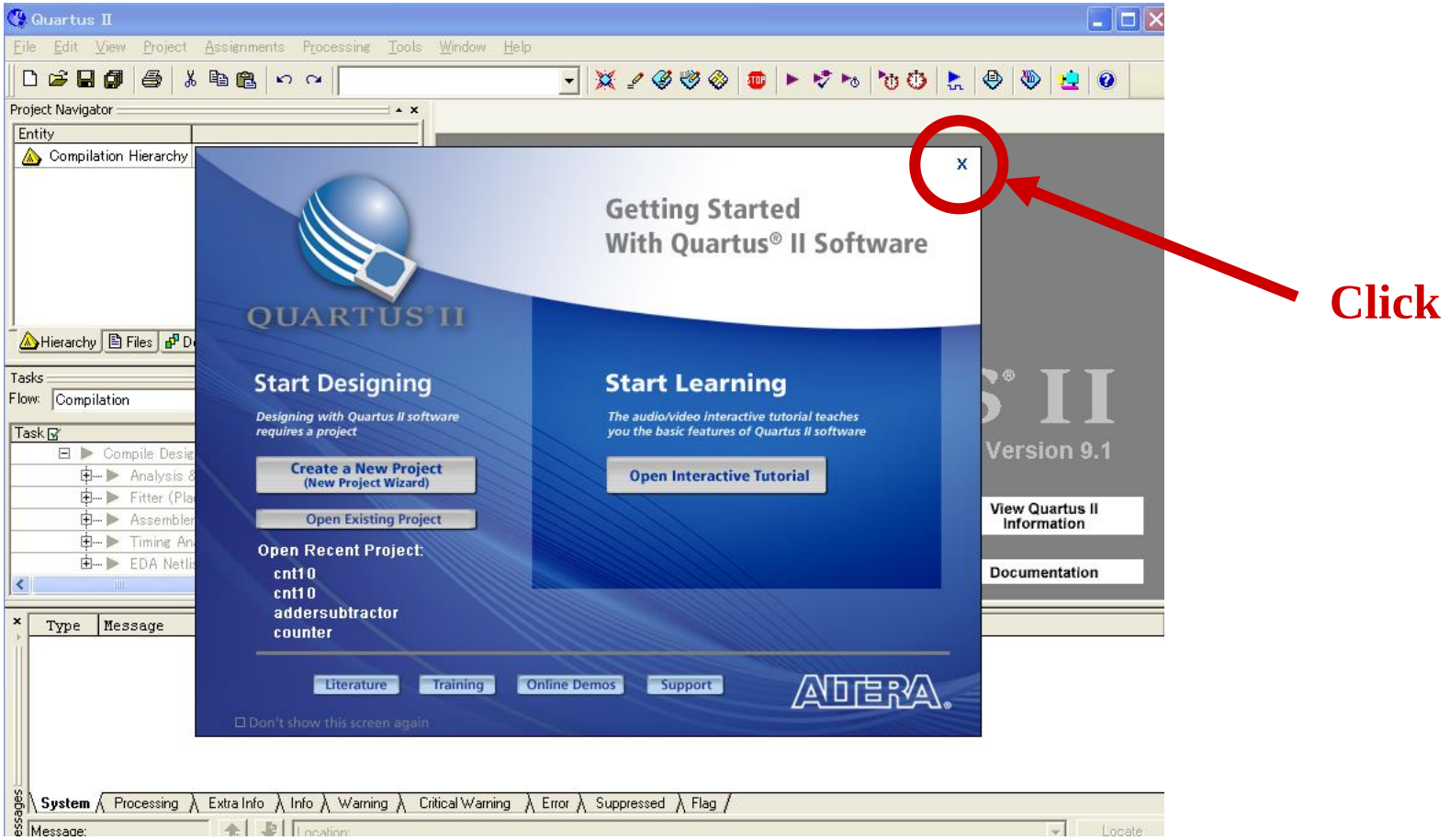


Or
from Desktop Icon

Quartus II: IDE (Integratd Design Environment) provided by Altera (FPGA vendor)
Friendly GUI and automated data conversion
Free Web Edition and Expensive Subscription Edition

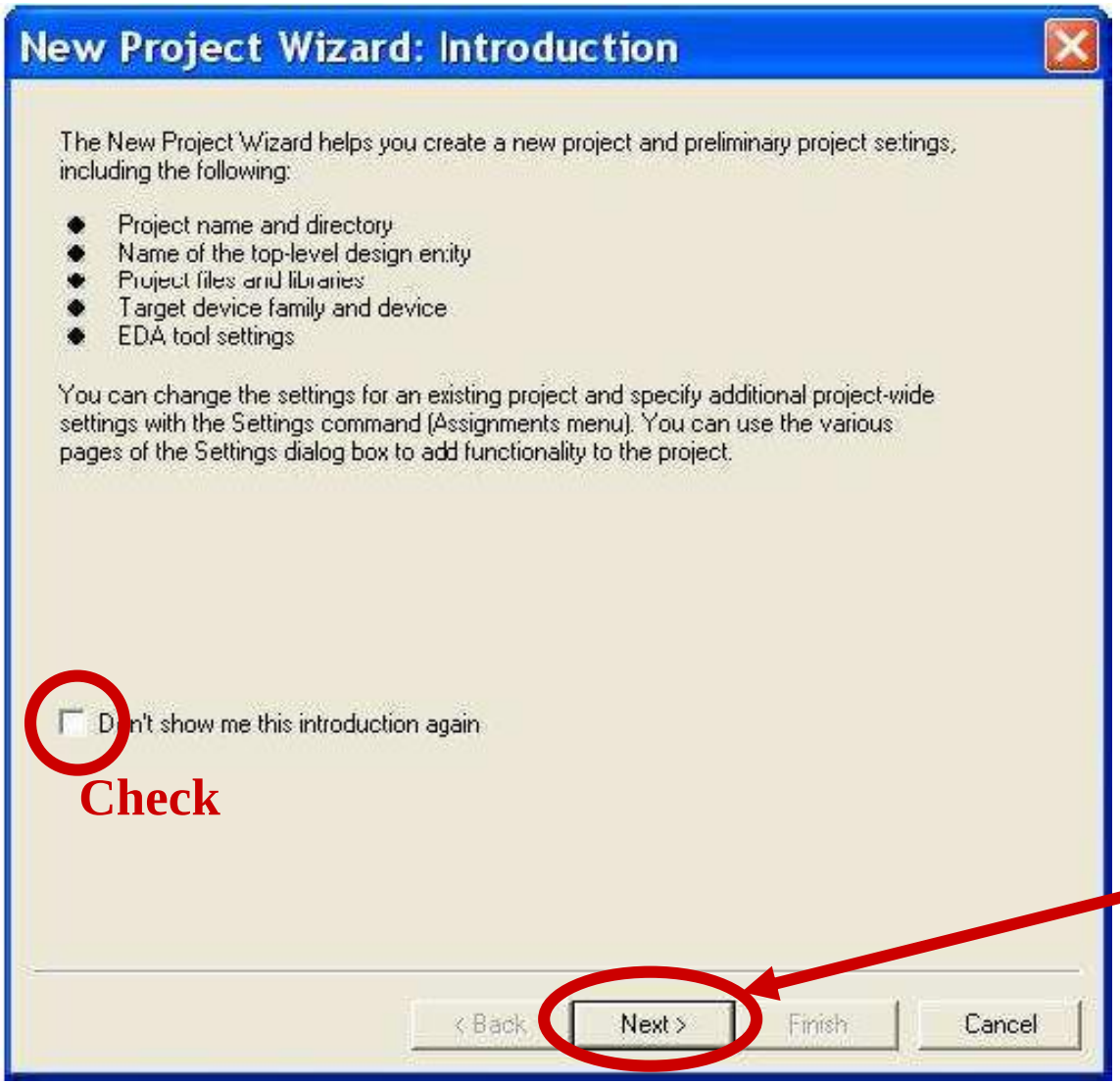
Xilinx (competitor of Altera) has a similar IDE: ISE

Starting Window



When you have time, you can obtain lots of information and tutorial from this Window.

New Project wizard



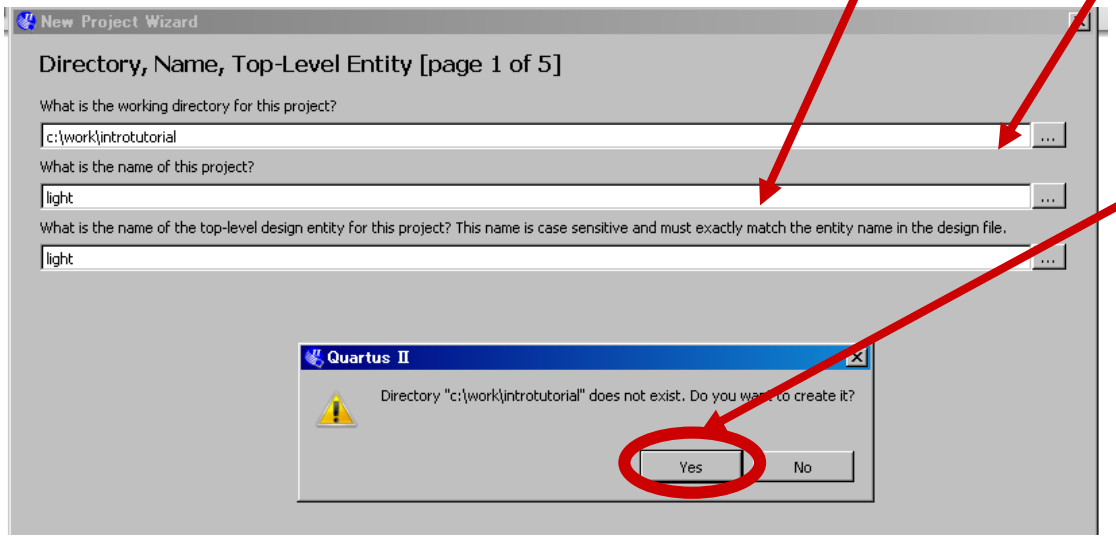
Check

Click

Project Starting-1 (New Project Wizard Window)

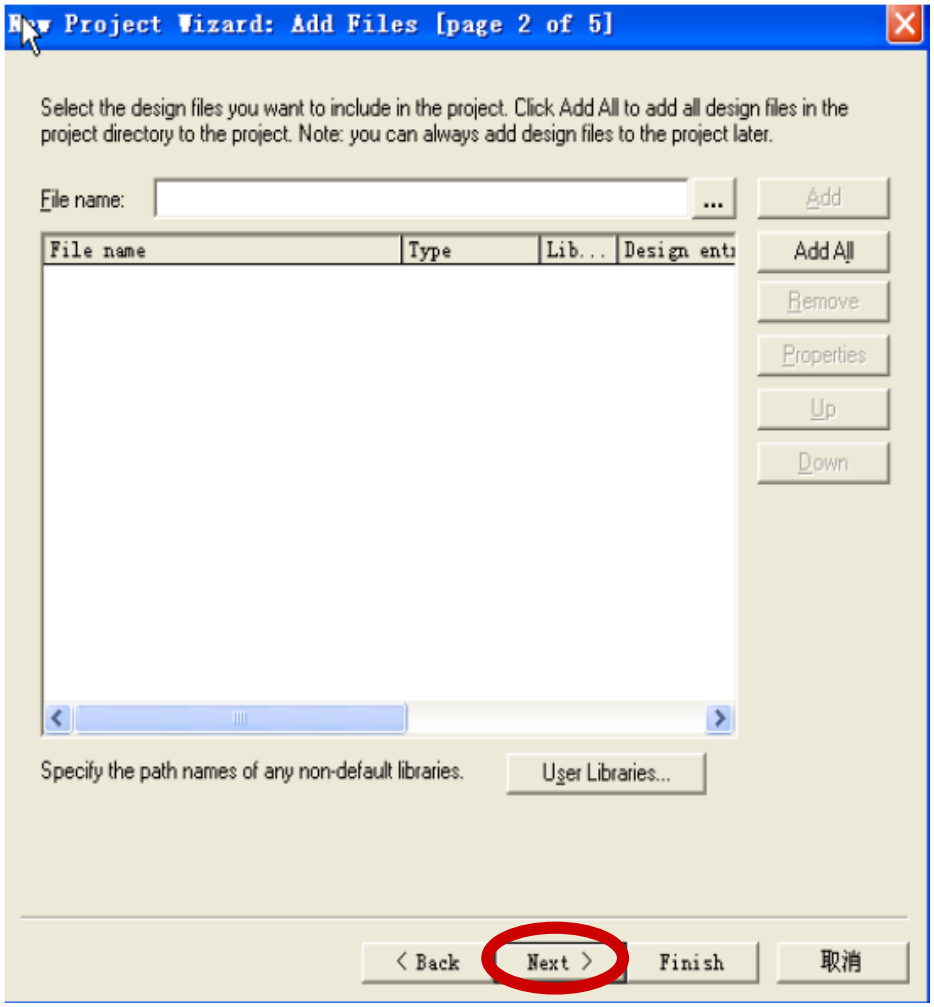
Kick-off of Project

1. Assign the name of Working Directry for this project: "C:¥project¥light" in "What is the working directory for this project?" box.
2. Assign the Project Name, in this project "light" in "What is the name of this project?" box. Then the top module name (top level design entity) is automatically assigned.
3. Click "Next"
4. When there is no Working Directry, Quartus asks you to create, please click "Yes".



Project Name and Top Module Name better be the same so as to simplify the Project management.

Project Starting- 2



Since this time you make a new design file, click "Next" in this window.

（If you want to reuse the existing design files or libraries, browse the directory or input file names and click ADD.

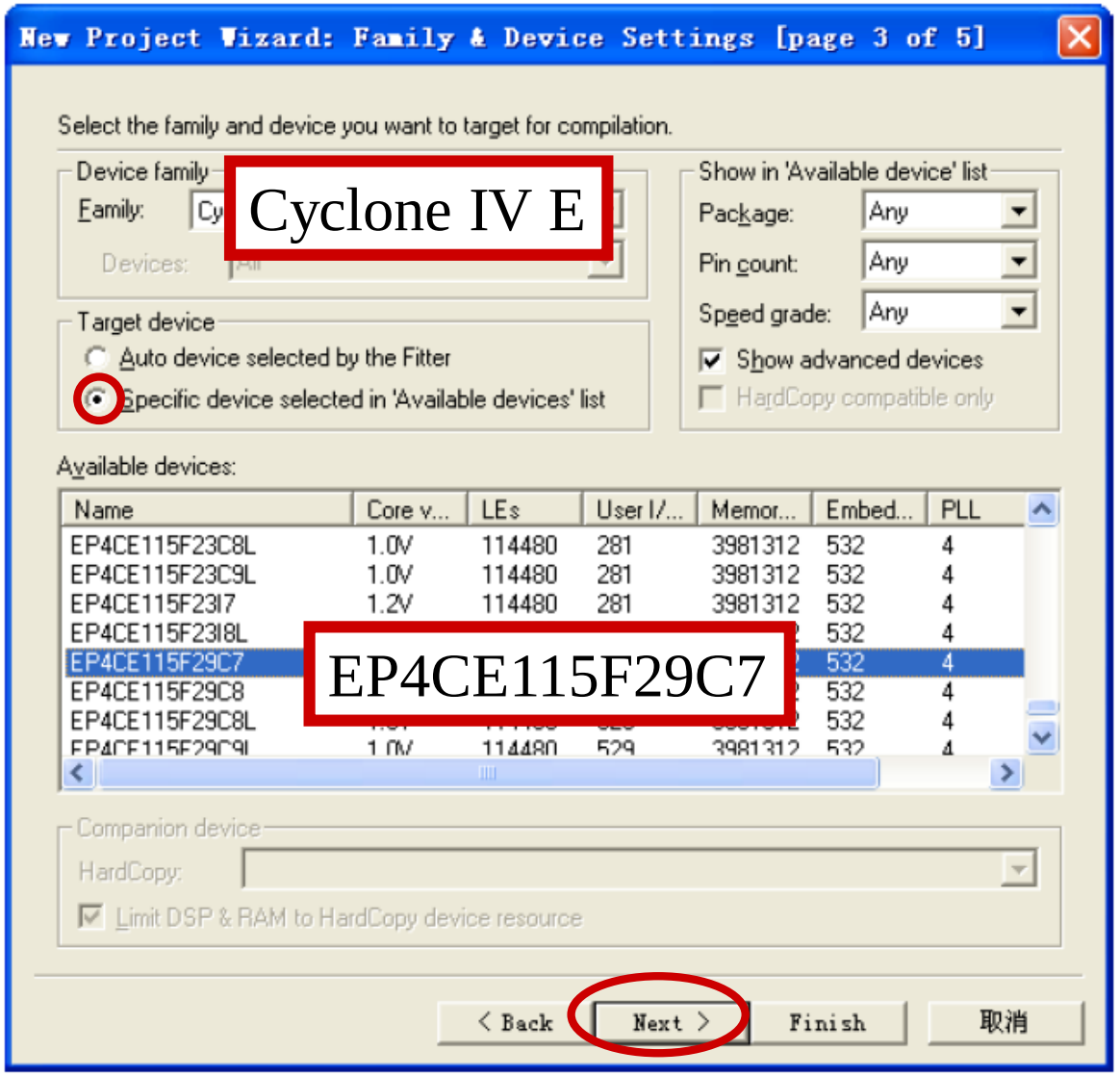
Project Starting -3 Assign the Device (FPGA):

Assign Cyclone IV E as the Target device family

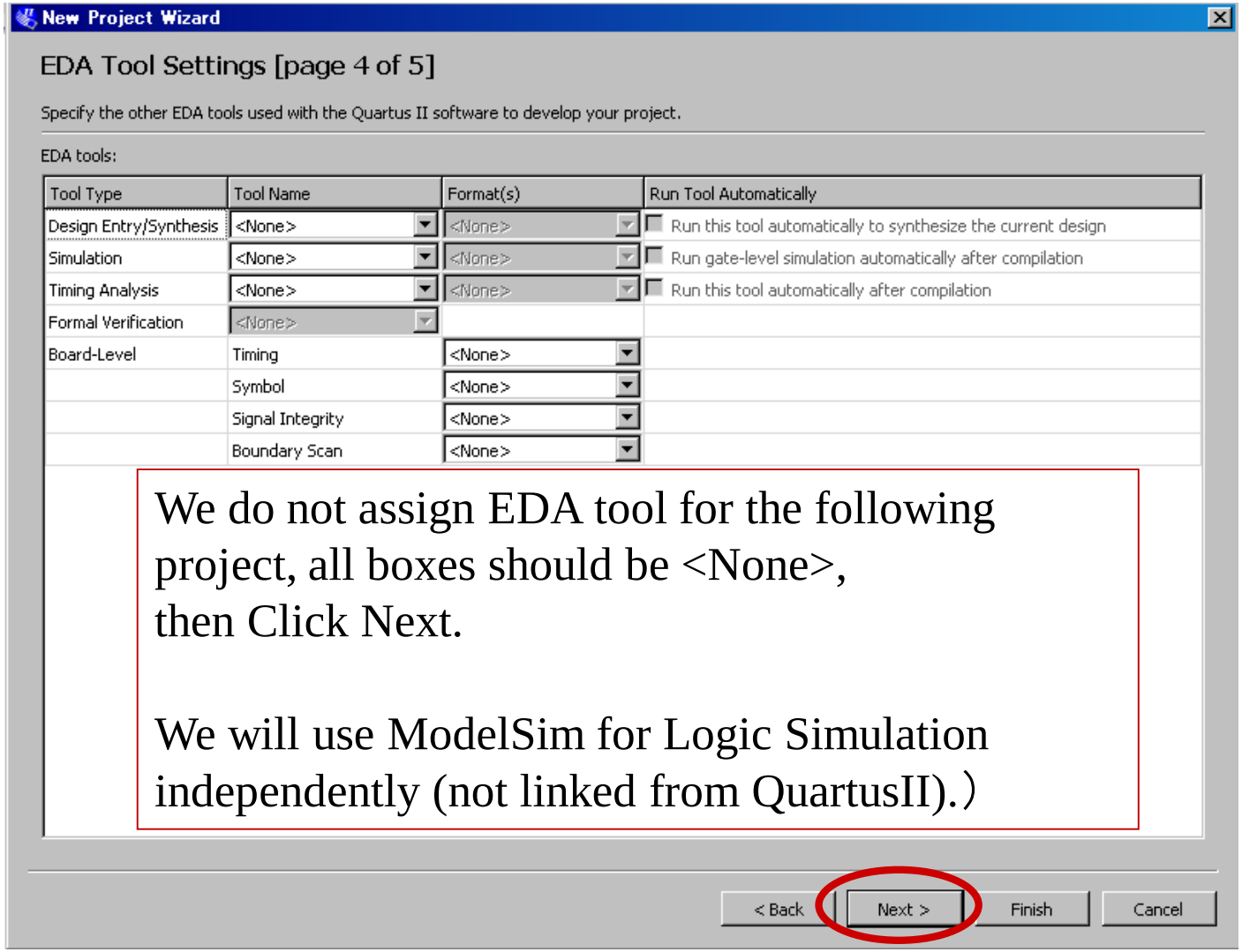
Click " Specific device selected in 'Available device' list.

FPGA device used in Altera's DE2-115 board is EP4CE115F29C7.

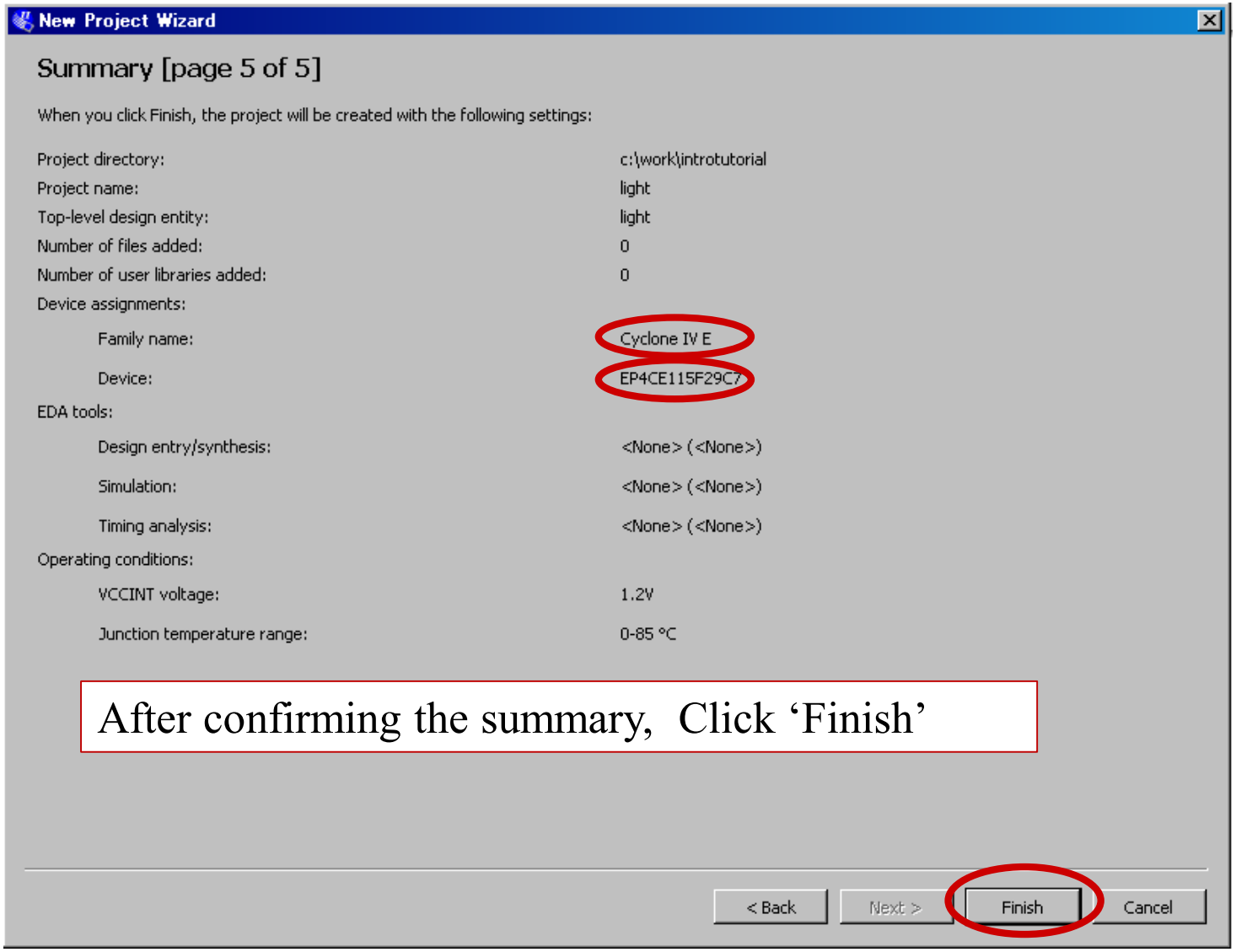
After assigning RED boxes/Circles, Click 'Next'.



Project Starting -4 Assigning EDA tool

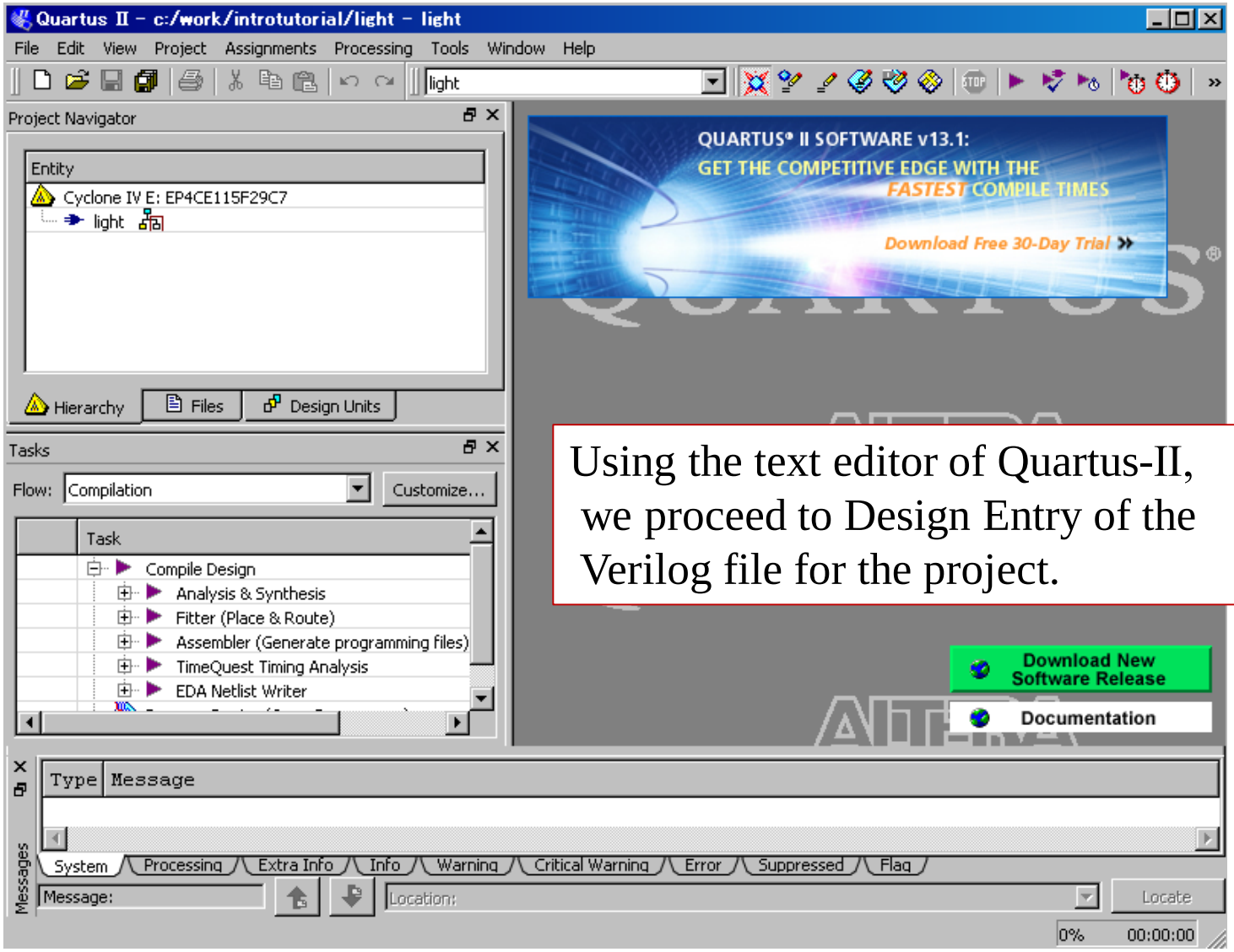


Project Starting -5 Confirm the setting



After confirming the summary, Click 'Finish'

After the Click of 'Finish' of New Project Wizard, the display would be like this.

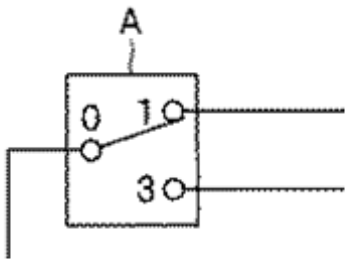


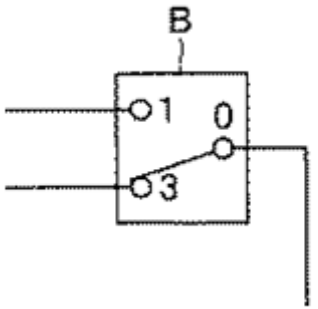
Using the text editor of Quartus-II, we proceed to Design Entry of the Verilog file for the project.

Example 1 （light controller circuit） #1

Before programming this circuit, a small brain teasing…

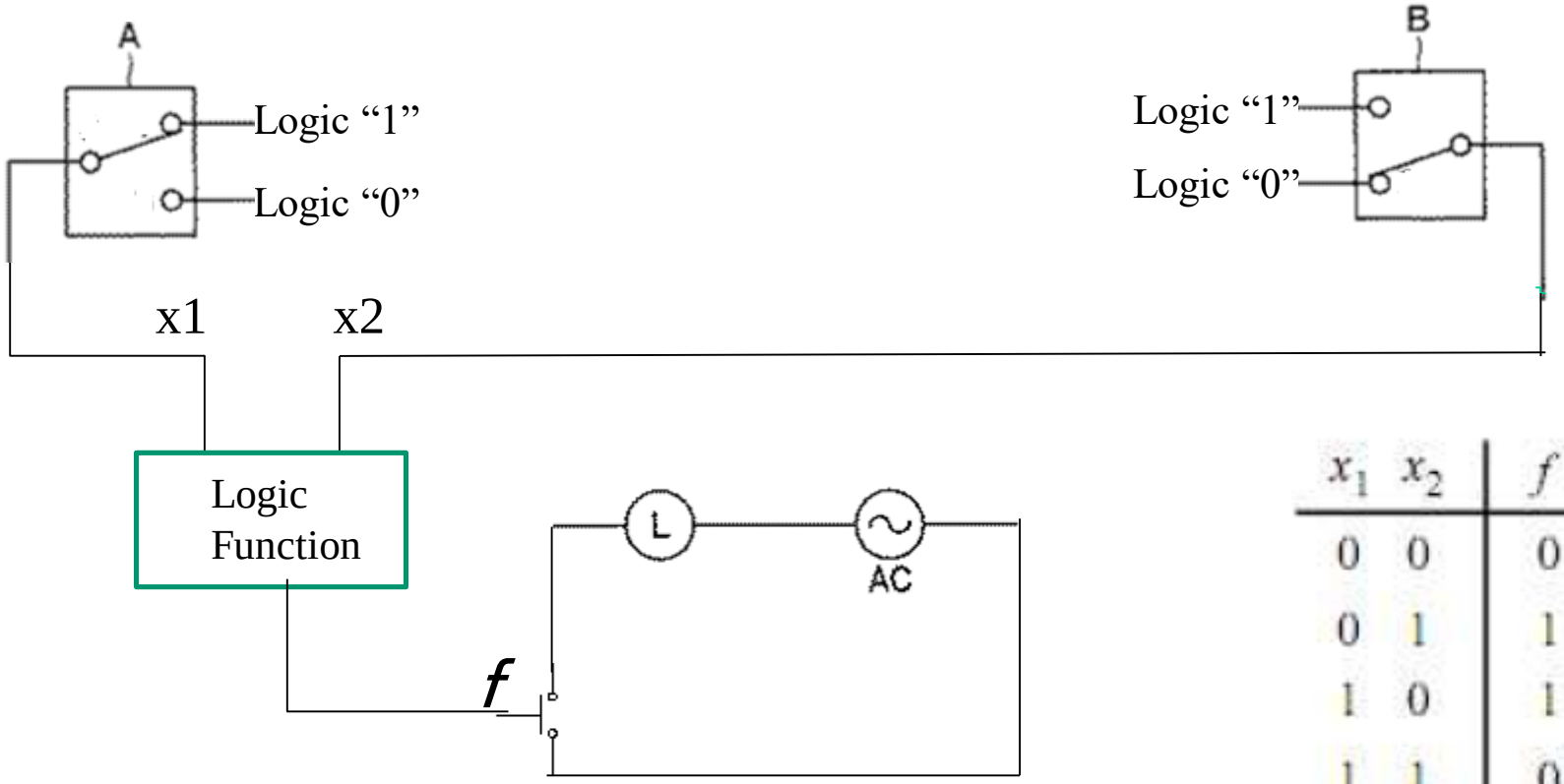
Problem: How to control the light in the middle of long corridor from either end ?





Long Corridor

Example 1 （light controller circuit） #2 Solution 1

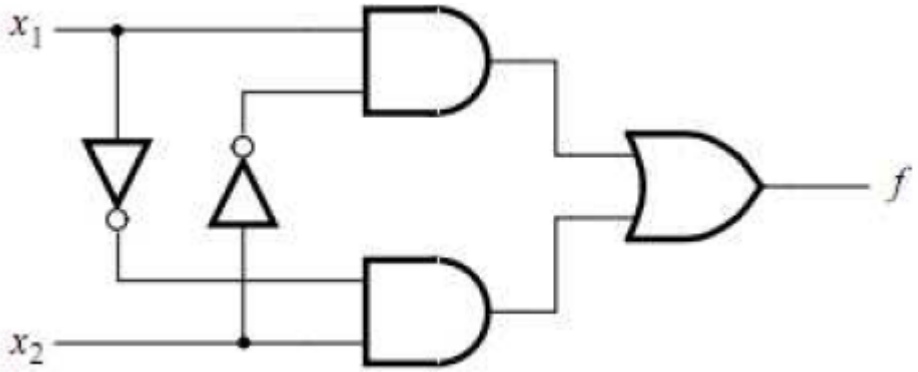


x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	0

Question：
What is the logic function of this circuit ?

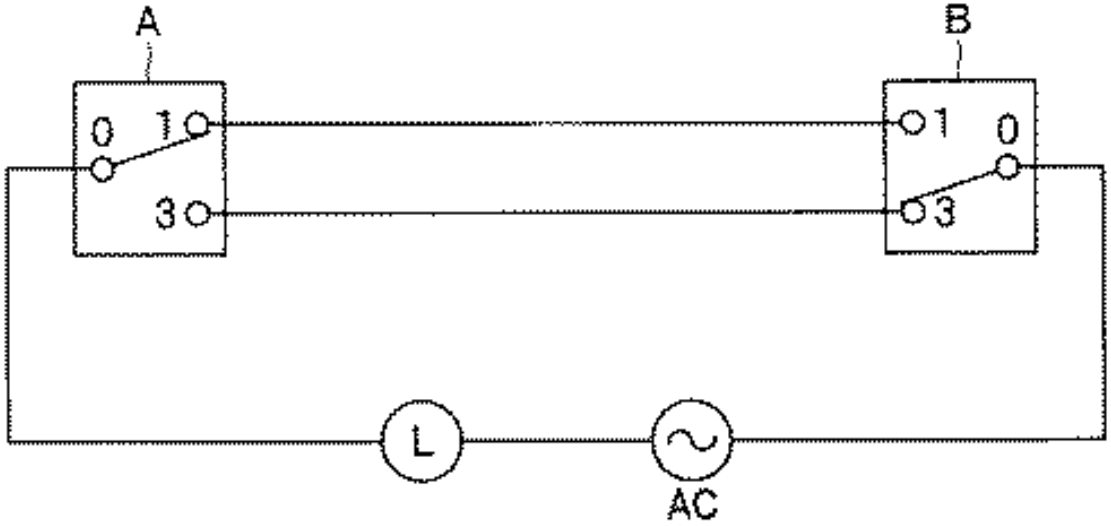
Example 1 （light controller circuit） #3

Schematic of the logic function



x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	0

Solution 2



Some rules for the file name to be used in projects

Top Module name : Assign when starting projects = light
File name : Recommend to be the same as Top Module = light
extension to indicate Verilog = . v

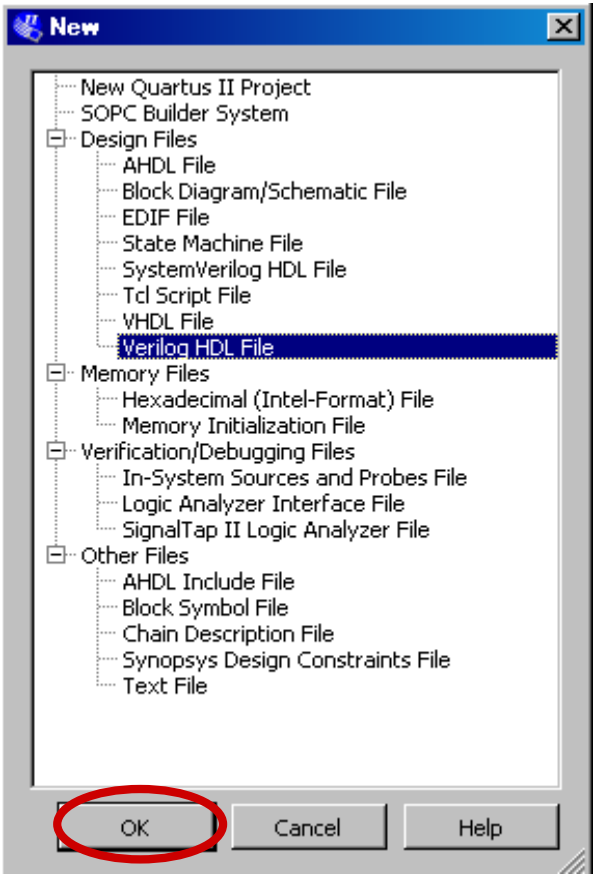
→ light.v

light.v

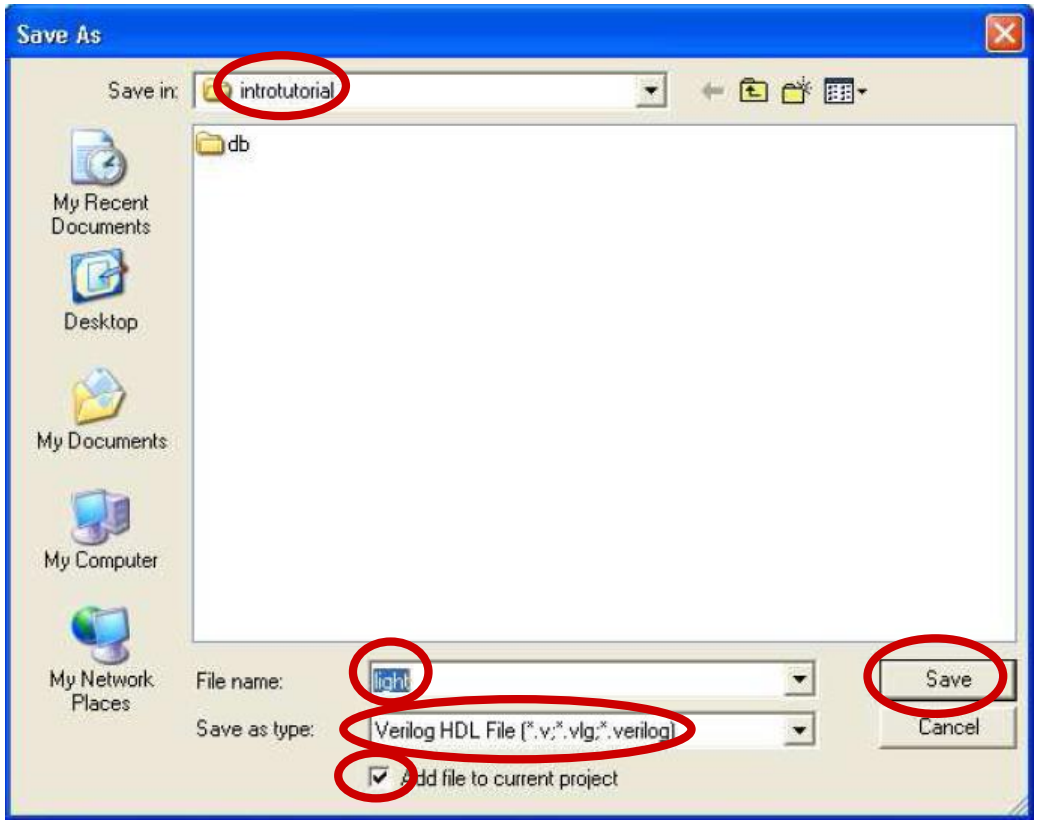
```
module light (x1, x2, f);
input x1, x2;
output f;
assign f = (x1 & ~x2) | (~x1 & x2);
endmodule
```

How to use the text editor of Quartus-II

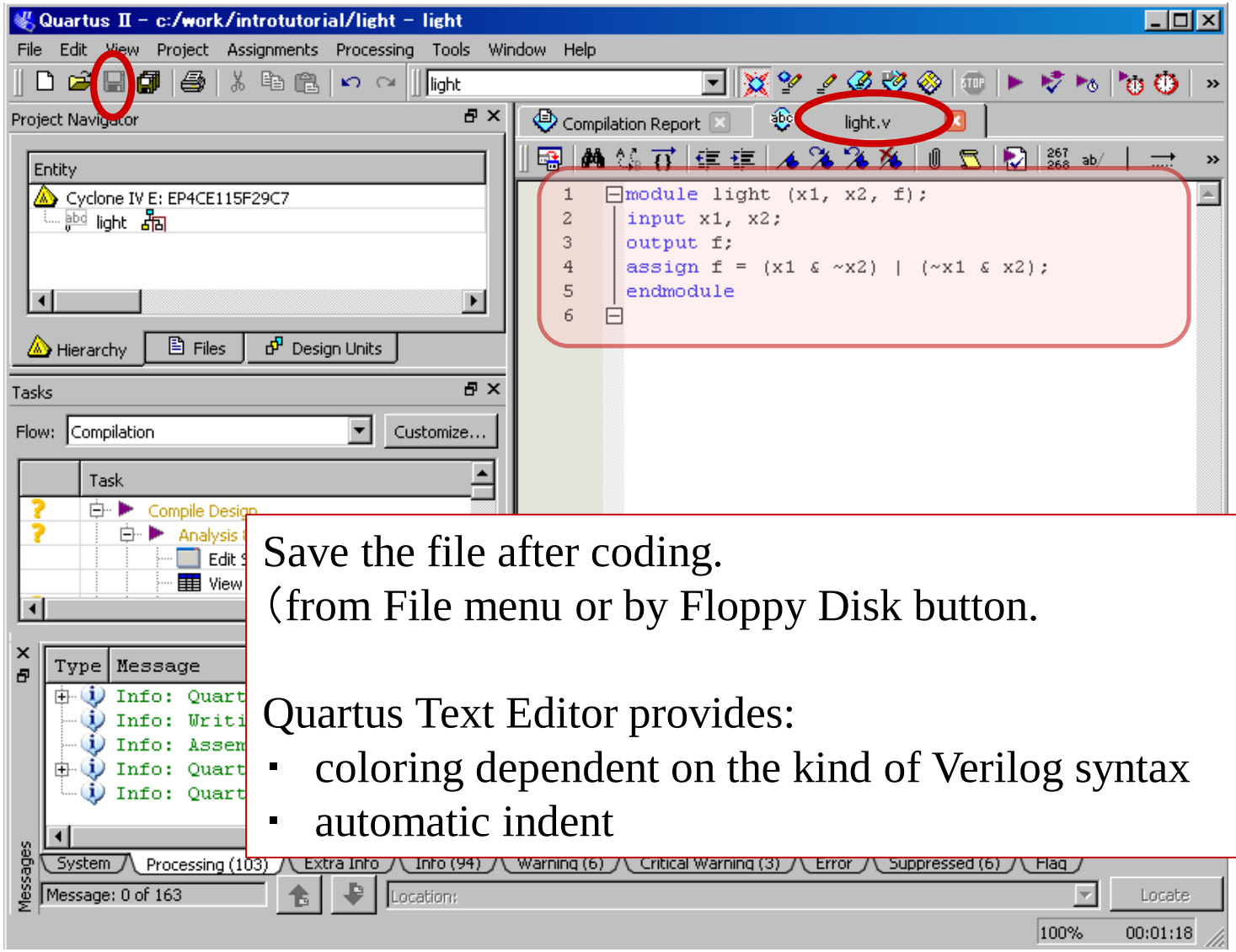
File > New opens the box:
Design Files >Verilog HDL File
Click OK



File > Save As opens the box:
Input the File name and File Type.
Check 'Add file to current project' and
Click 'Save'



Verilog Coding by the Text Editor



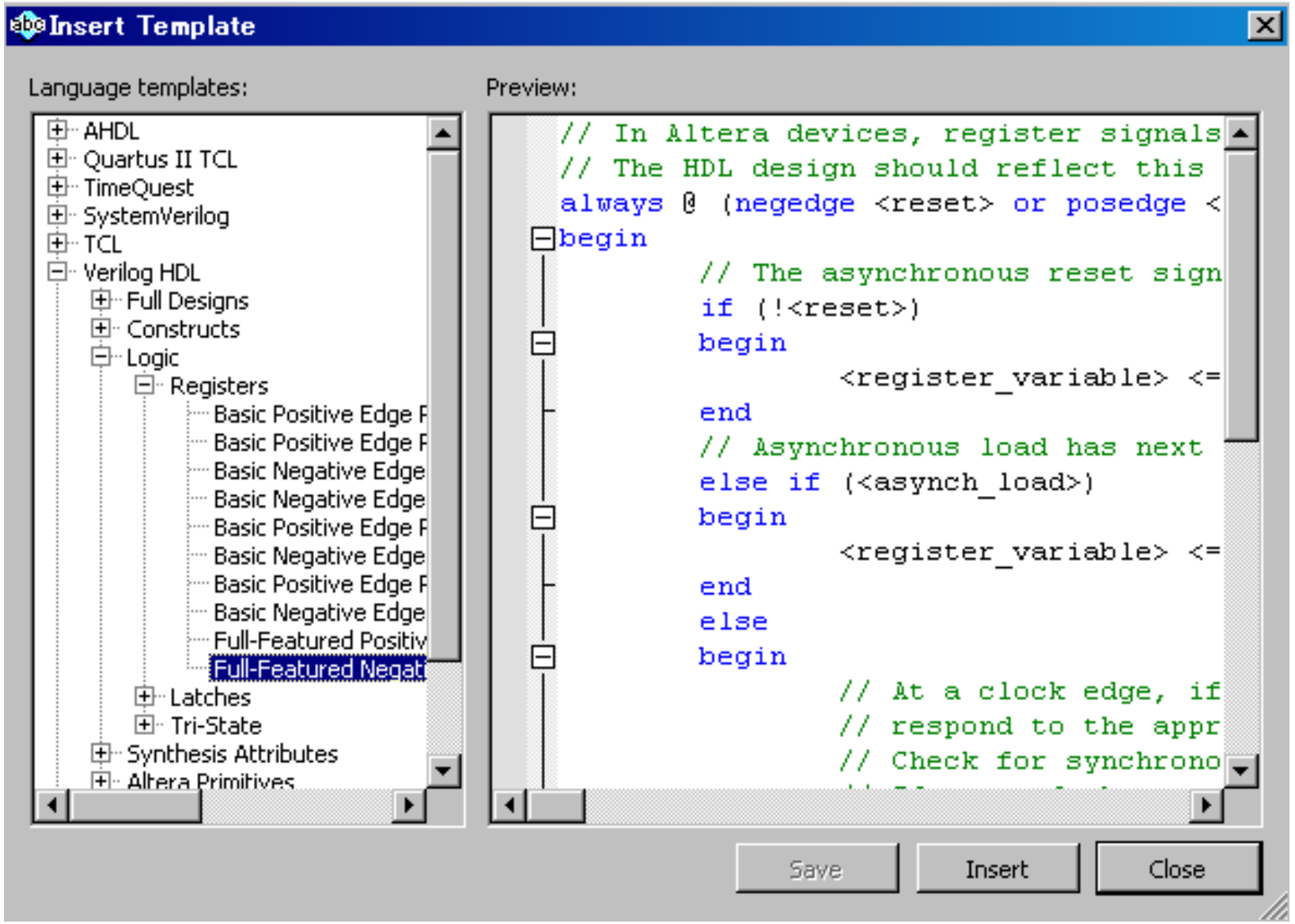
Save the file after coding.
（from File menu or by Floppy Disk button.

Quartus Text Editor provides:

- coloring dependent on the kind of Verilog syntax
- automatic indent

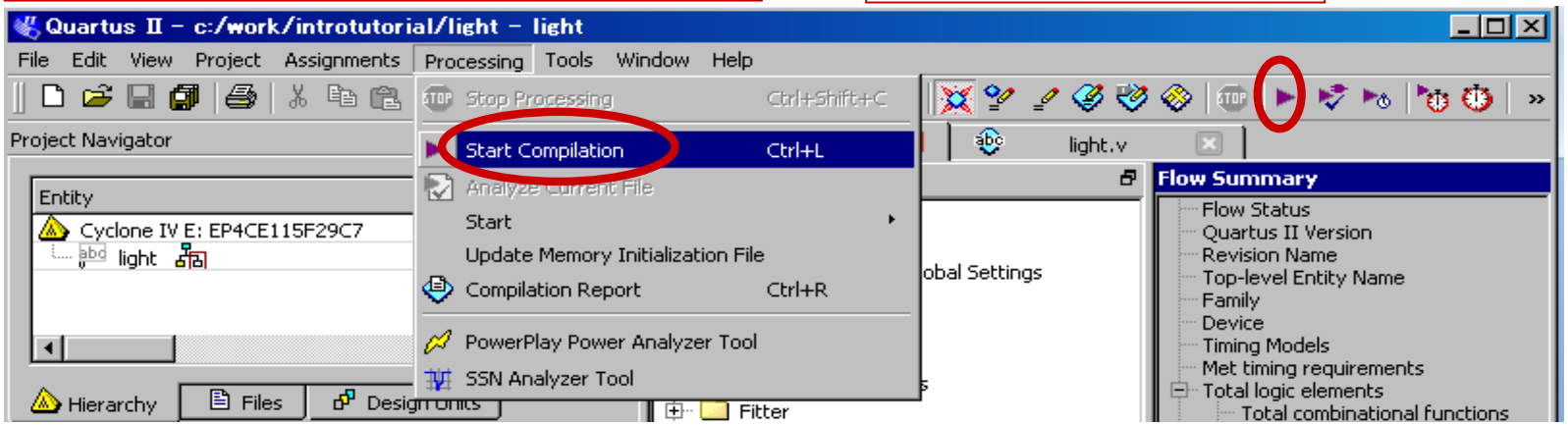
For use in the future:

Edit > Insert Template > Verilog HDL is convenient and helpful

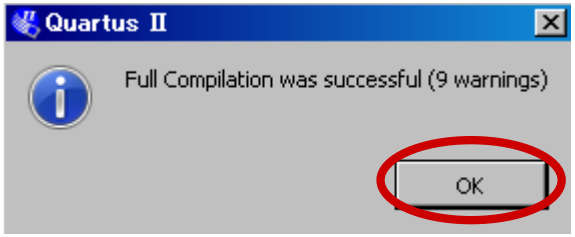


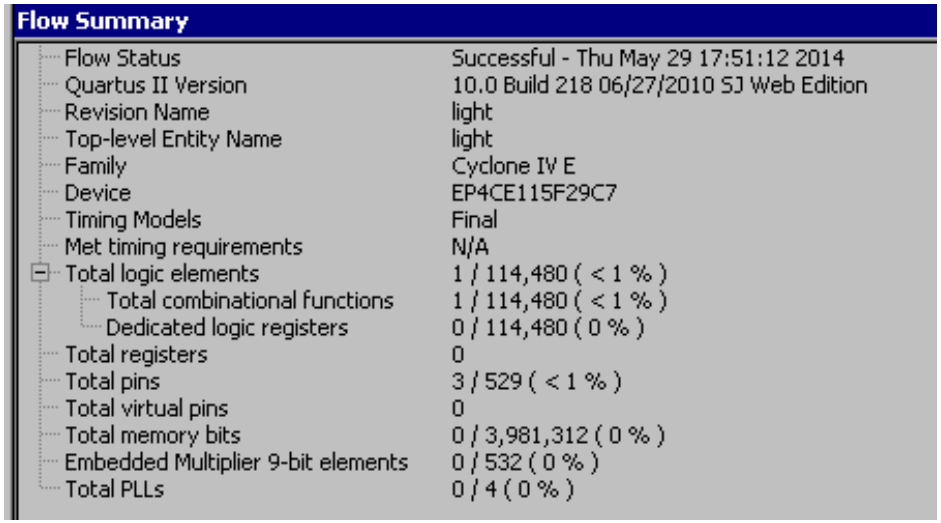
Verilog File is saved, then Compile !

Processing > Start Compilation or toolbar icon Starts compilation



When Compile finished,
Message box of ‘Successful / Unsuccessful’ is displayed.
Click ‘OK’, and the summary is shown.

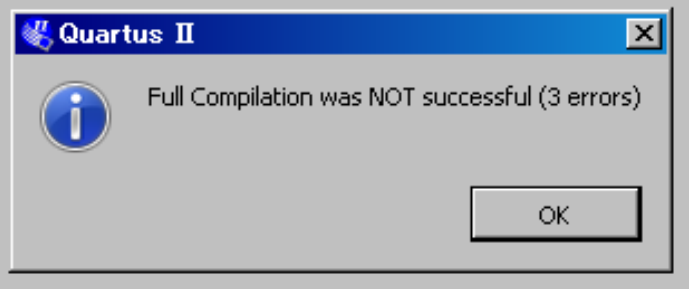




If Compilation unsuccessful･･･ （some mistakes in Verilog Code）

NOT Successful

Revision Name	light
Top-level Entity Name	light
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Met timing requirements	N/A
Total logic elements	N/A until Partition Merge
Total combinational functions	N/A until Partition Merge
Dedicated logic registers	N/A until Partition Merge
Total registers	N/A until Partition Merge
Total pins	N/A until Partition Merge
Total virtual pins	N/A until Partition Merge
Total memory bits	N/A until Partition Merge
Embedded Multiplier 9-bit elements	N/A until Partition Merge
Total PLLs	N/A until Partition Merge

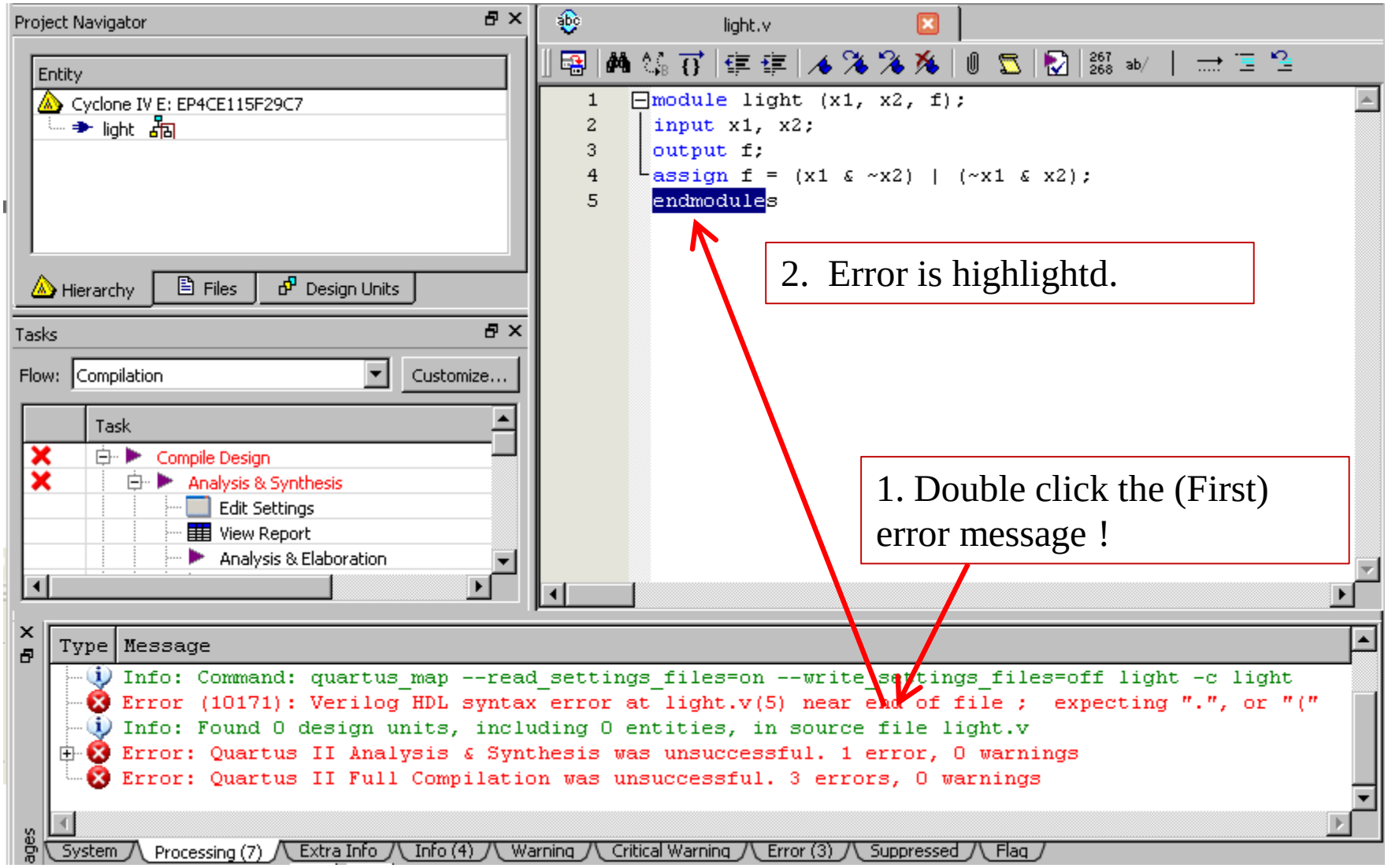


Flow Summary

Flow Status	Successful - Thu May 29 17:51:12 2014
Quartus II Version	10.0 Build 218 06/27/2010 SJ Web Edition
Revision Name	light
Top-level Entity Name	light
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Met timing requirements	N/A
Total logic elements	1 / 114,480 (< 1 %)
Total combinational functions	1 / 114,480 (< 1 %)
Dedicated logic registers	0 / 114,480 (0 %)
Total registers	0
Total pins	3 / 529 (< 1 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

Successful (same as previous page)

Locating the Error !

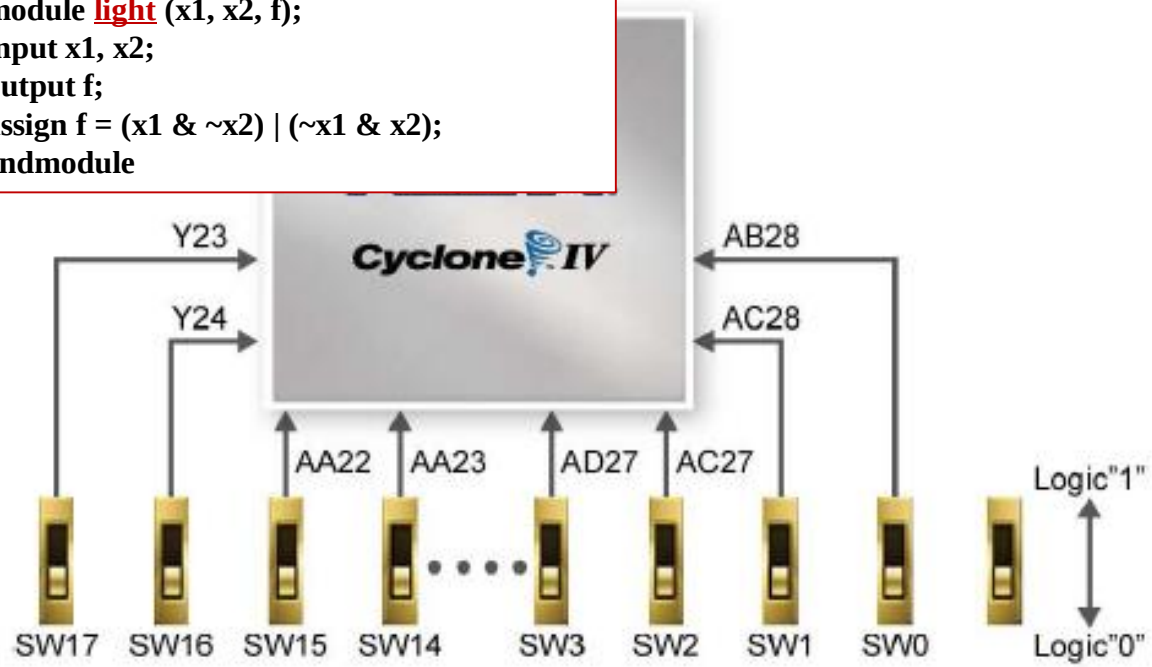


Pin Assignment

On FPGA board like DE2-115, every FPGA pins are connected to the devices or terminals of the board with fixed wires.
So you have to assign the signals in the module (Verilog Code) to FPGA Pins.

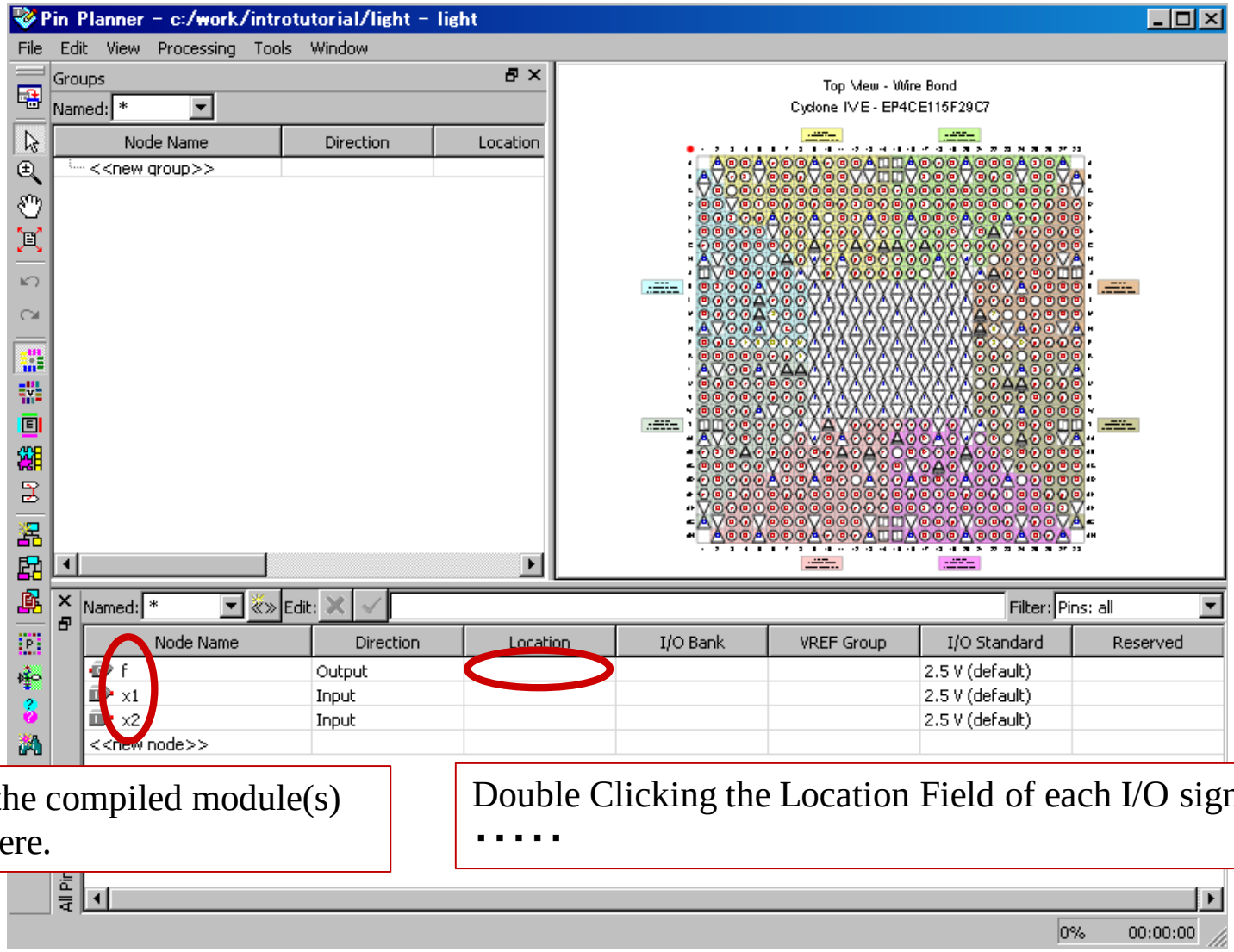
Here, we use the slide swithes labeled SW0 and SW1 to the input signals of the light.v module.
On the board, SW0 and SW1 are connected to AB28 pin and AC28 pin of FPGA respectively.
Further output f shall be connected to the LED labeled LEDG0.
LEDG0 is connected to E21 pin of FPGA.。

```
module light (x1, x2, f);
input x1, x2;
output f;
assign f = (x1 & ~x2) | (~x1 & x2);
endmodule
```



Pin Assignment with Quartus-II

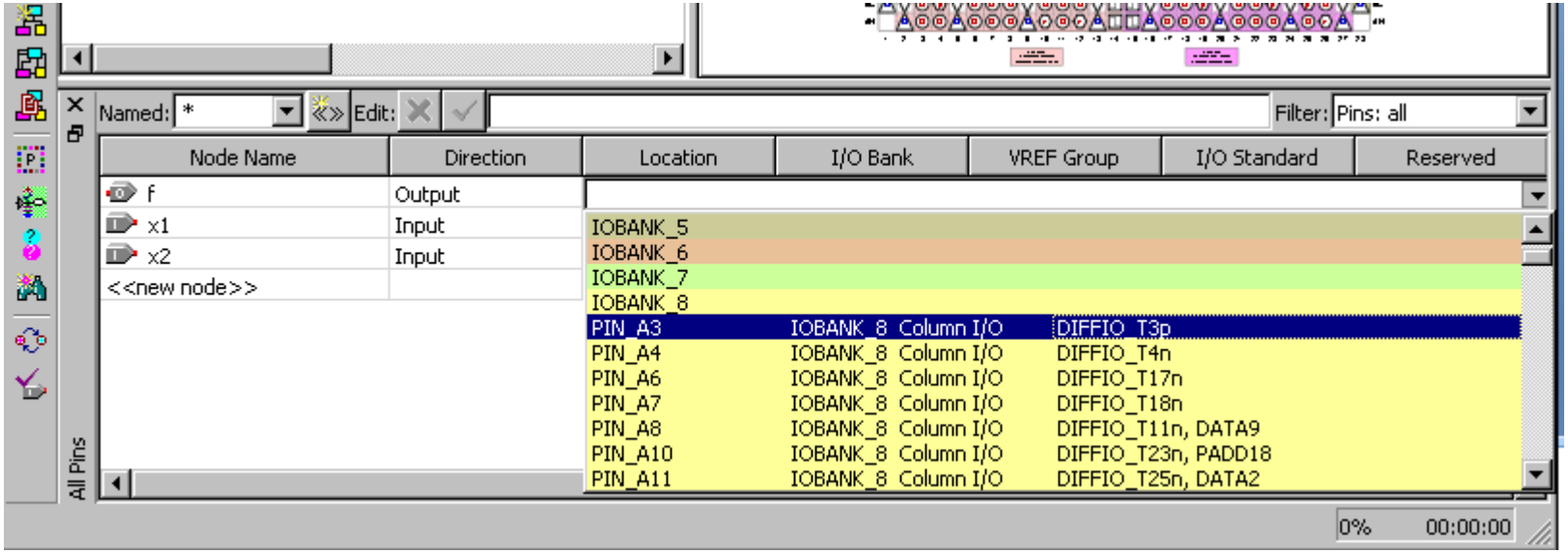
Assignments > Pin Planner opens Pin Planner Window



I/O signals of the compiled module(s) are extracted here.

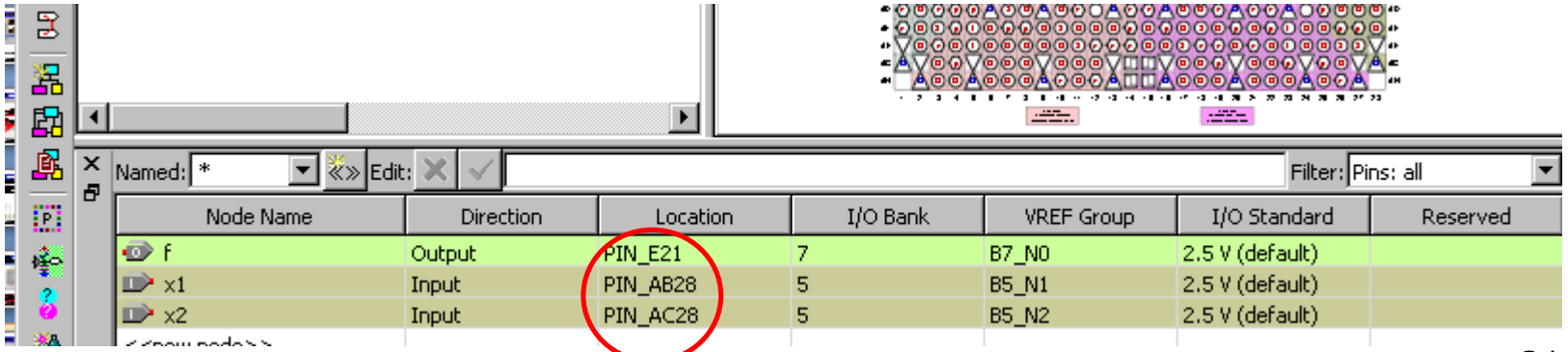
Double Clicking the Location Field of each I/O signals, ...

Double Clicking the Location Field,
you can scroll the list of FPGA pins in the Location Field.



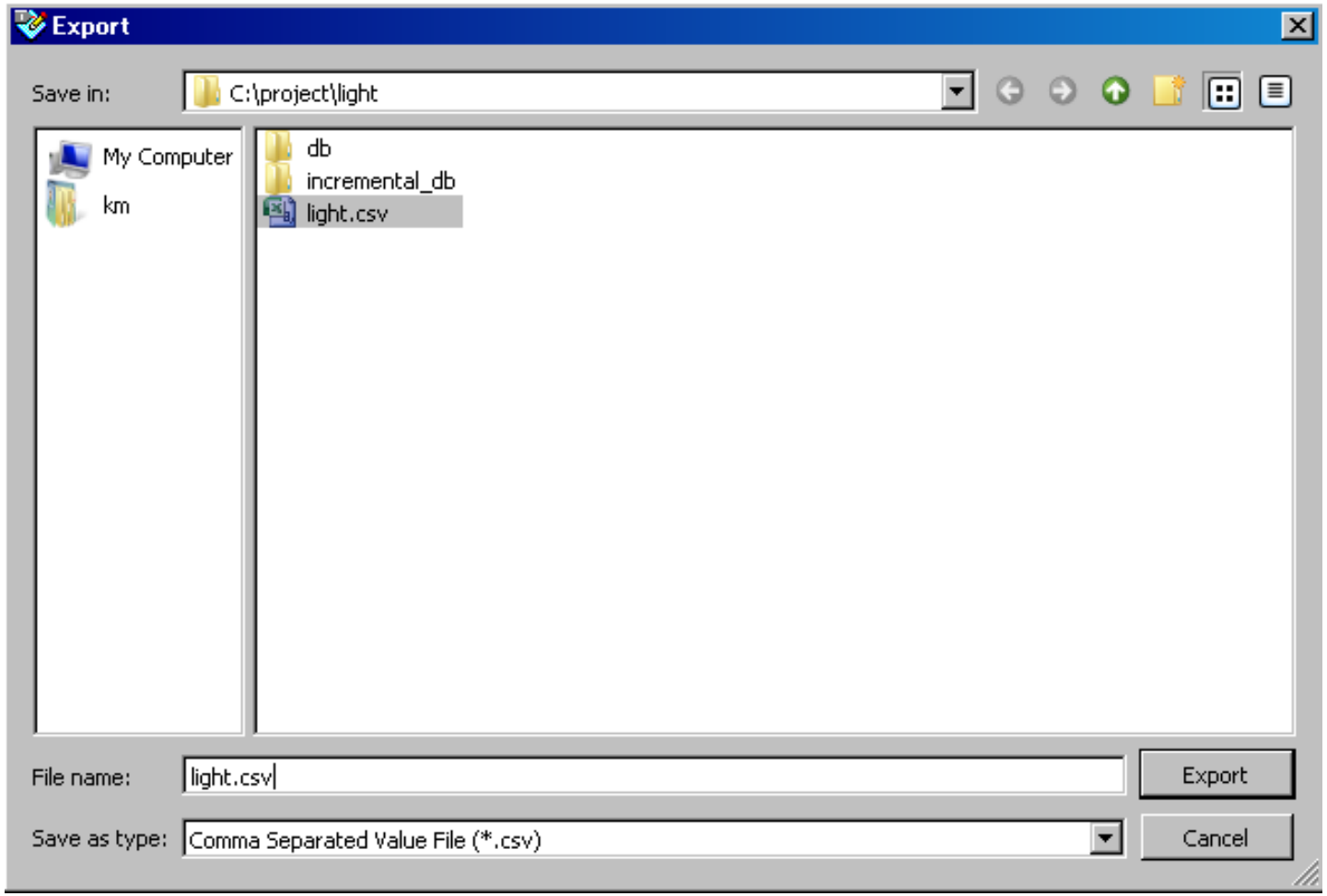
Assign Output f to pin E21, input x1 to AB28, input x2 to AC28.
(instead of scrolling you can input pin names with the keyboard)
Close the Pin Planner Window after pin assignment finished.

The Location filed will be filled like this:



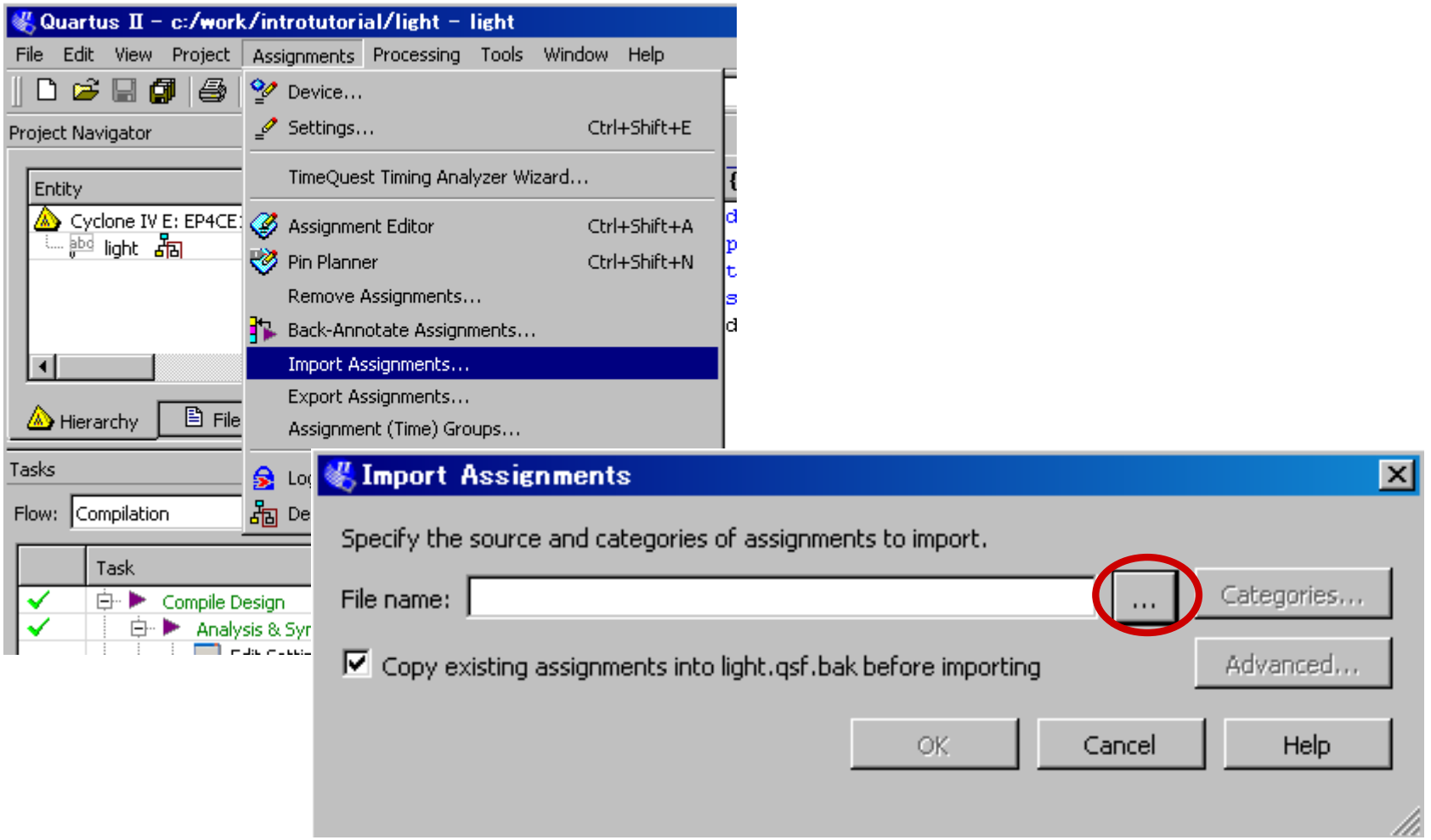
The Pin Assignment information can be re-used！ -1

In Pin Planner : File>Export you can save the CSV file in any directory.

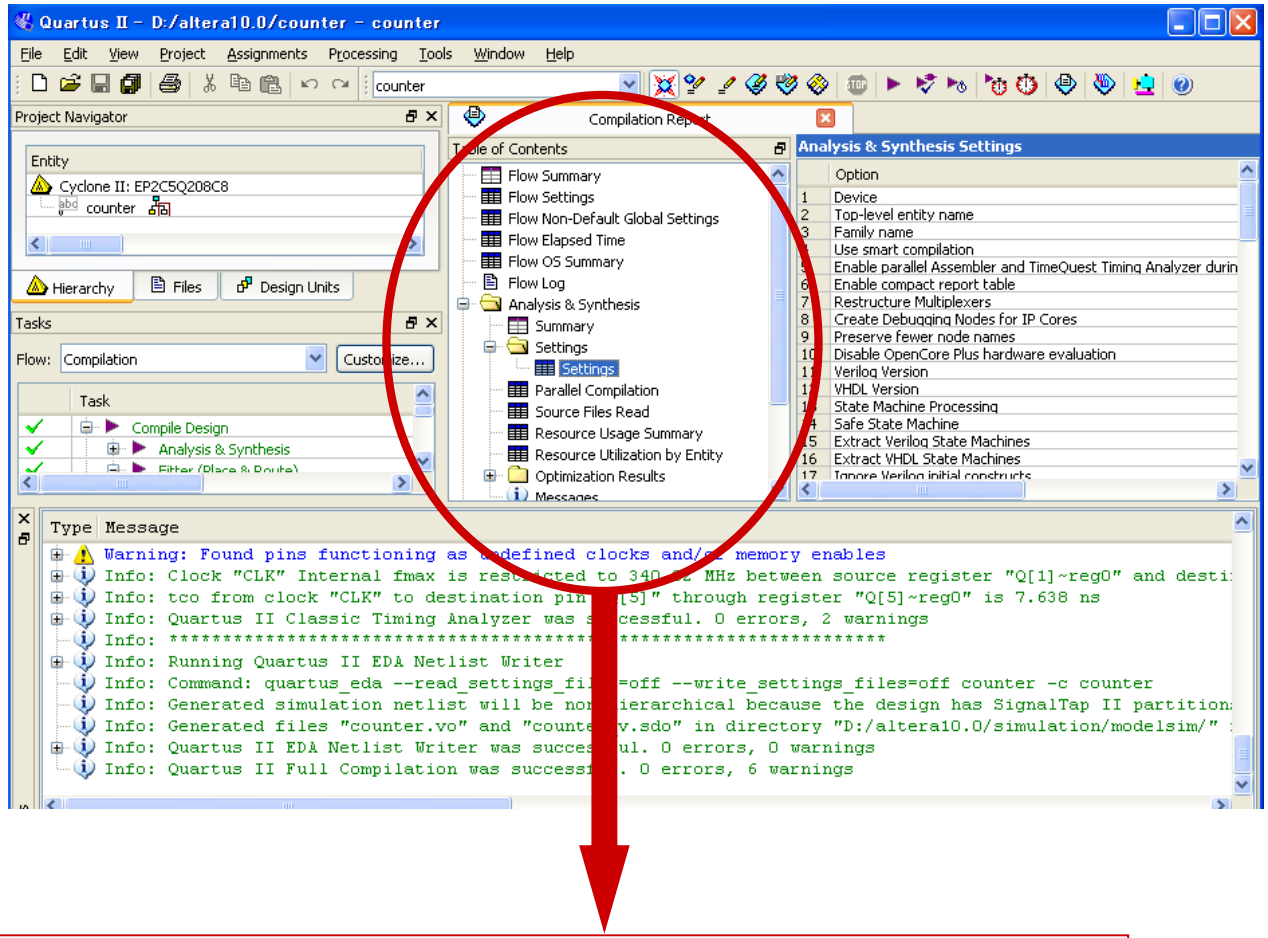


The Pin Assignment information can be re-used！ -2

Quartus-II’s Assignments>Import Assignments Tab, you can import the existing CSV files.



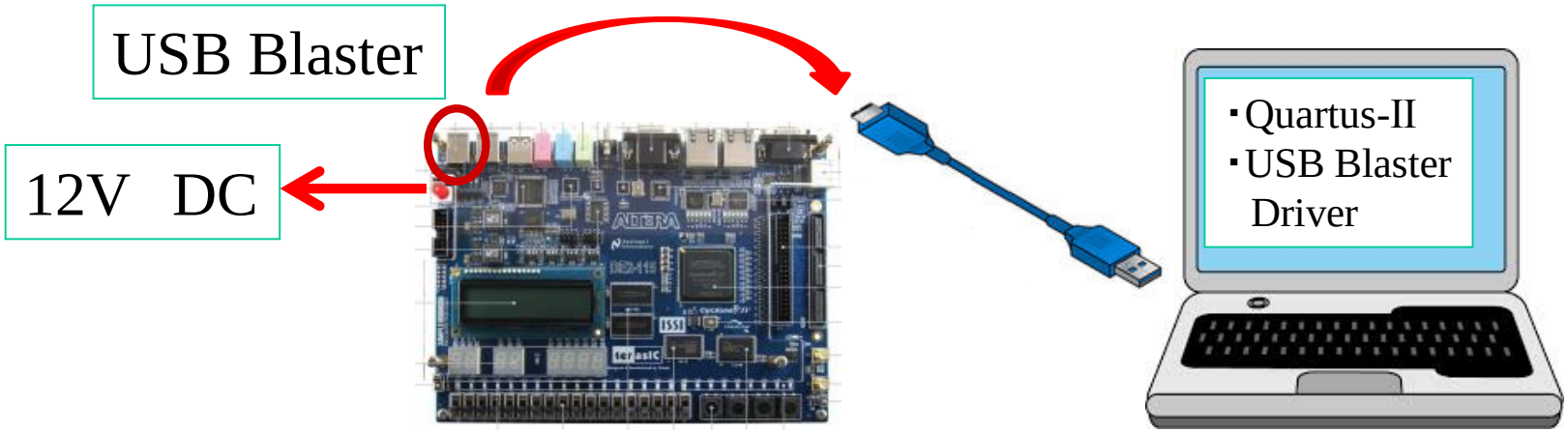
Compilation Report （Processing > Compilation Report）



Navigating this directory, you can get lots of informations.

You can get the schematic view of the logic compilation with RTL Viewer.

If Compile is successful, then Program the FPGA



There are Two Programming Modes

- JTAG Mode
- Active Serial (AS) Mode

We are employing

JTAG（Joint Test Action Group）Mode

JTAG Programming: Prepartion for PC

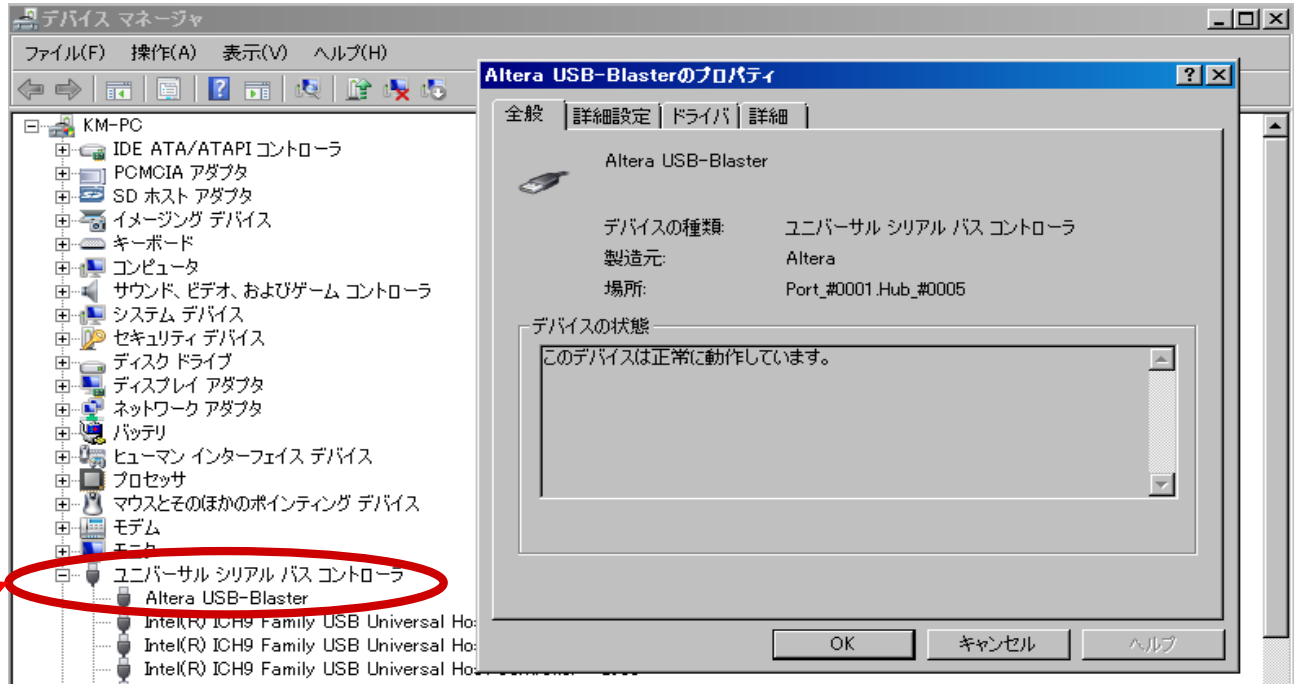
When PC does not recognize USB-Blaster:
Check the Device Manager where Driver is installed.

Connect USB Cable, Power on DE2-115.
If PC detects new Device, Specify the source of Driver as follows:
(Do not specify x32 or x64 below ¥ysb-blaster directory)

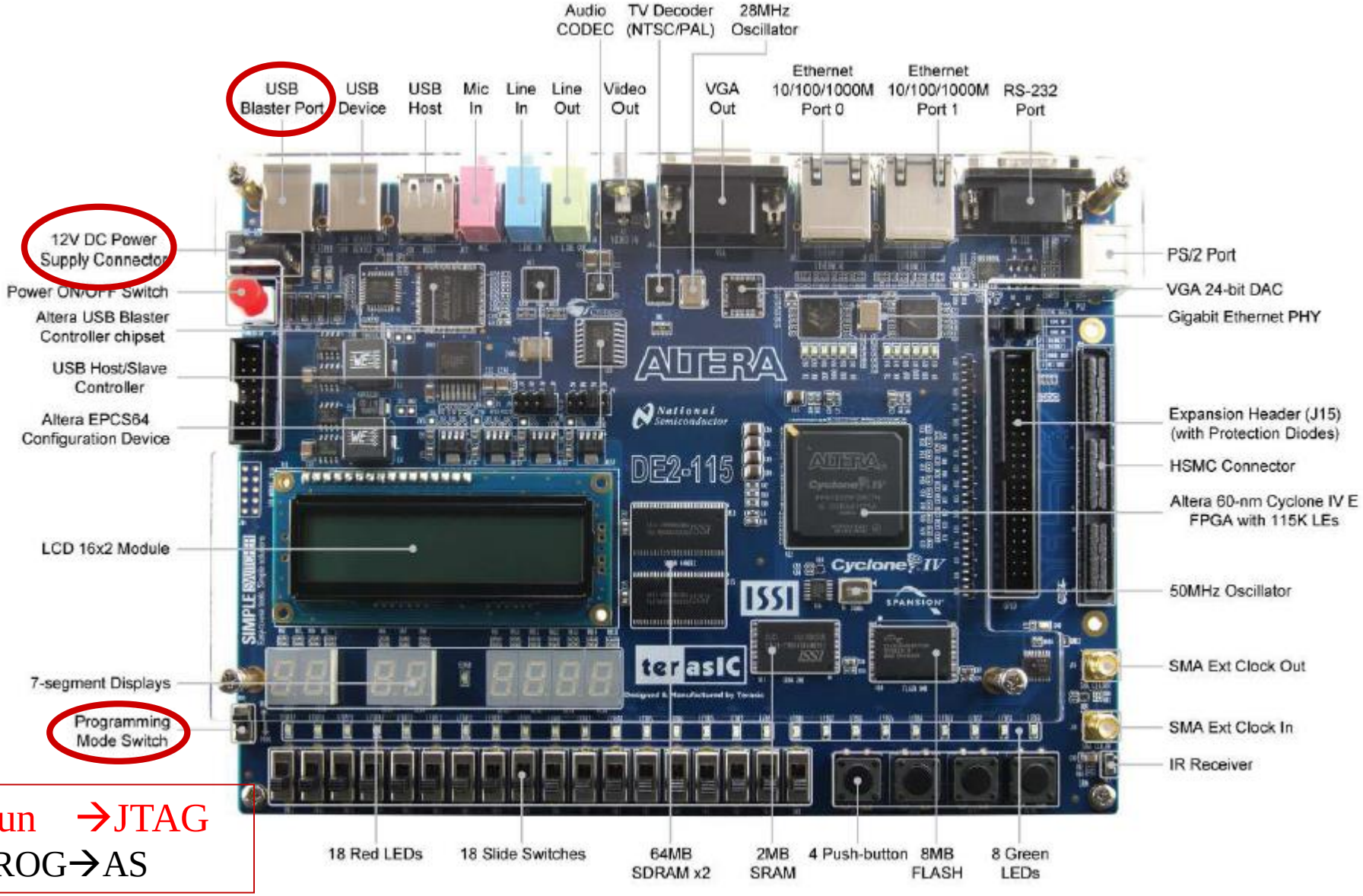
C:¥altera¥10.0¥quartus¥drivers¥usb-blaster

Confirm

USB controller



JTAG Programming: Preparaion for DE2-115側準
(Specify the Programming Mode)



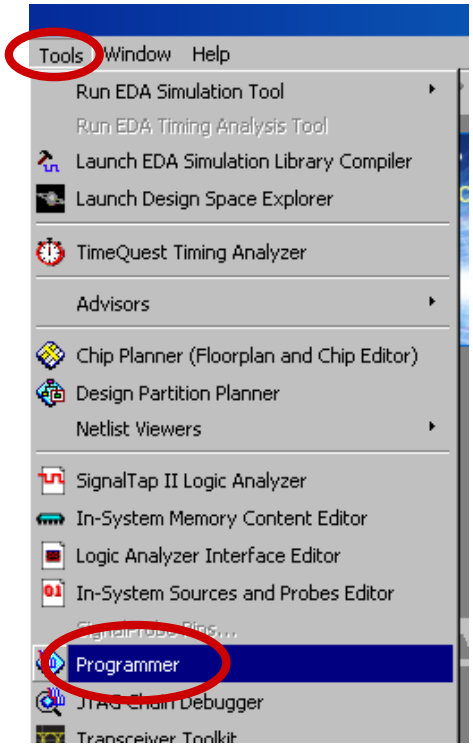
Run →JTAG
PROG→AS

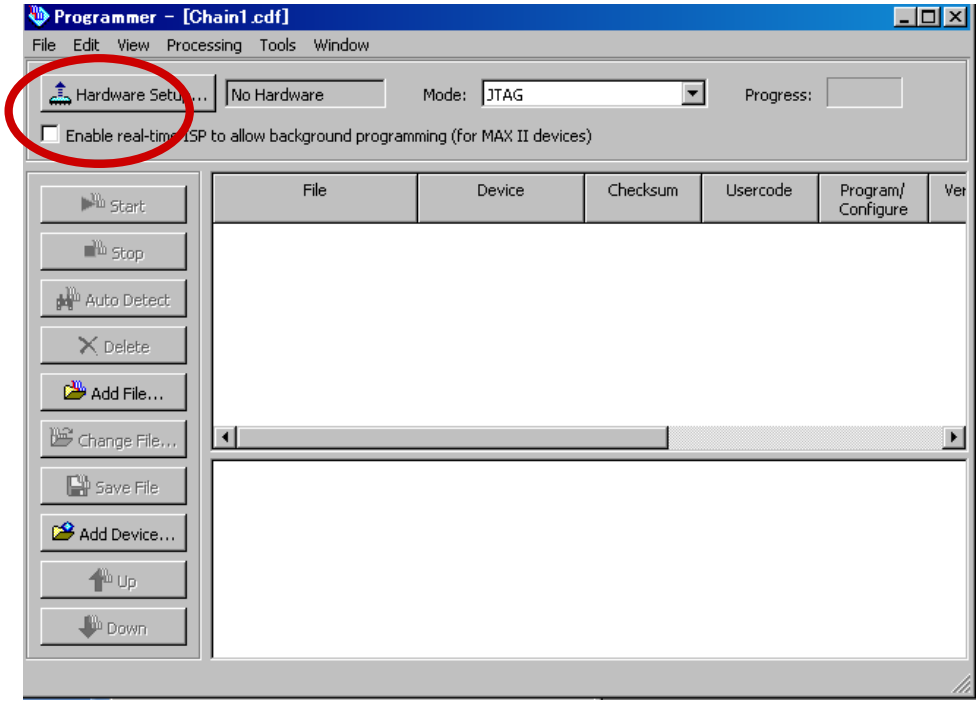
JTAG Programming

After preparation finished,

- Connect the Power Cable of DE2-115 and
- Connect the USB-Blaster Port and USB Port of PC with USB cable provided.

1. Power On DE2-115 (No error message on PC Screen, then DE2-115 is recognized successfully by PC)
2. From QuartusII's Tab, select Tools > Programmer
3. Programmer Window opens as below.



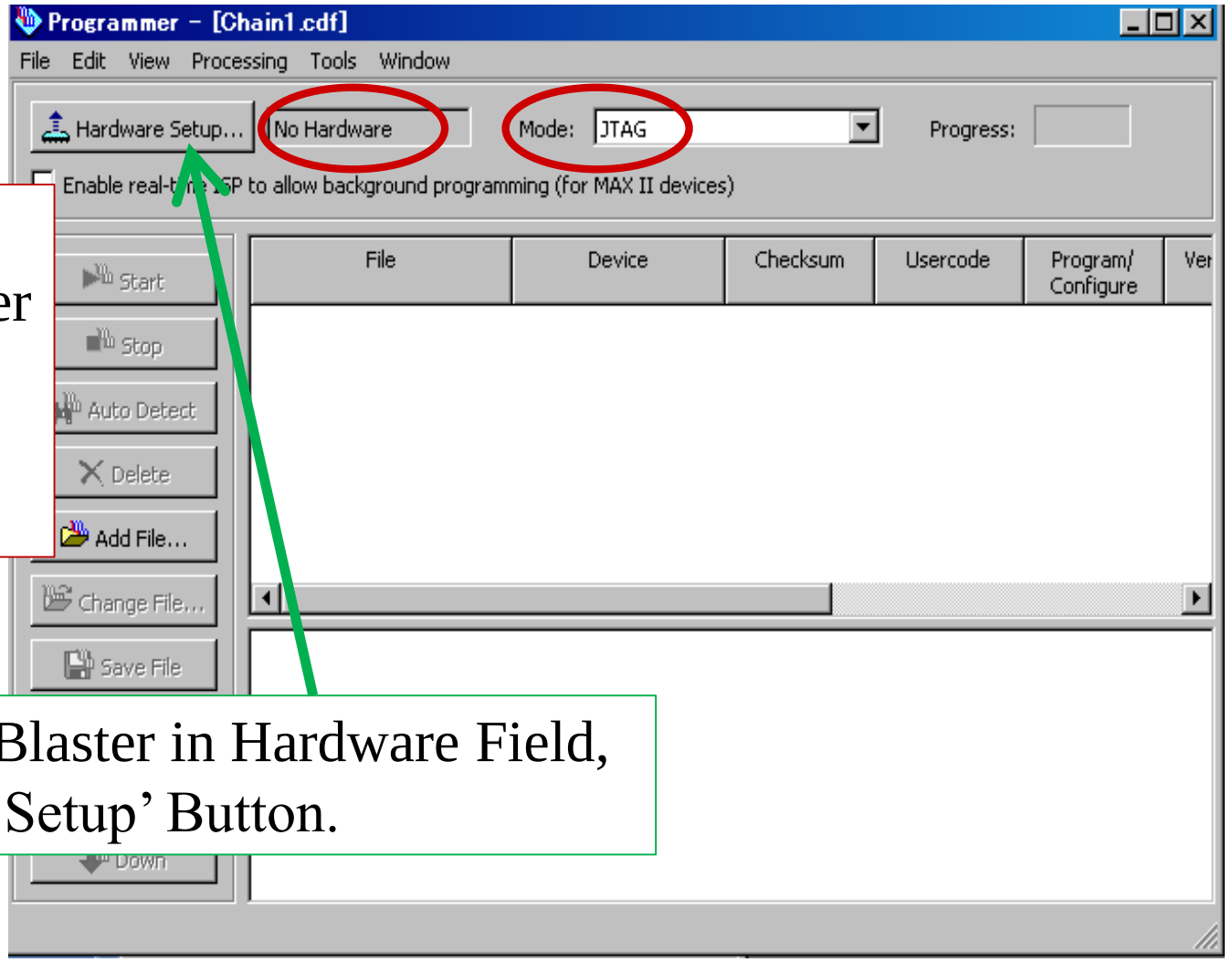


Set-up of Programmer Window

Confirmation

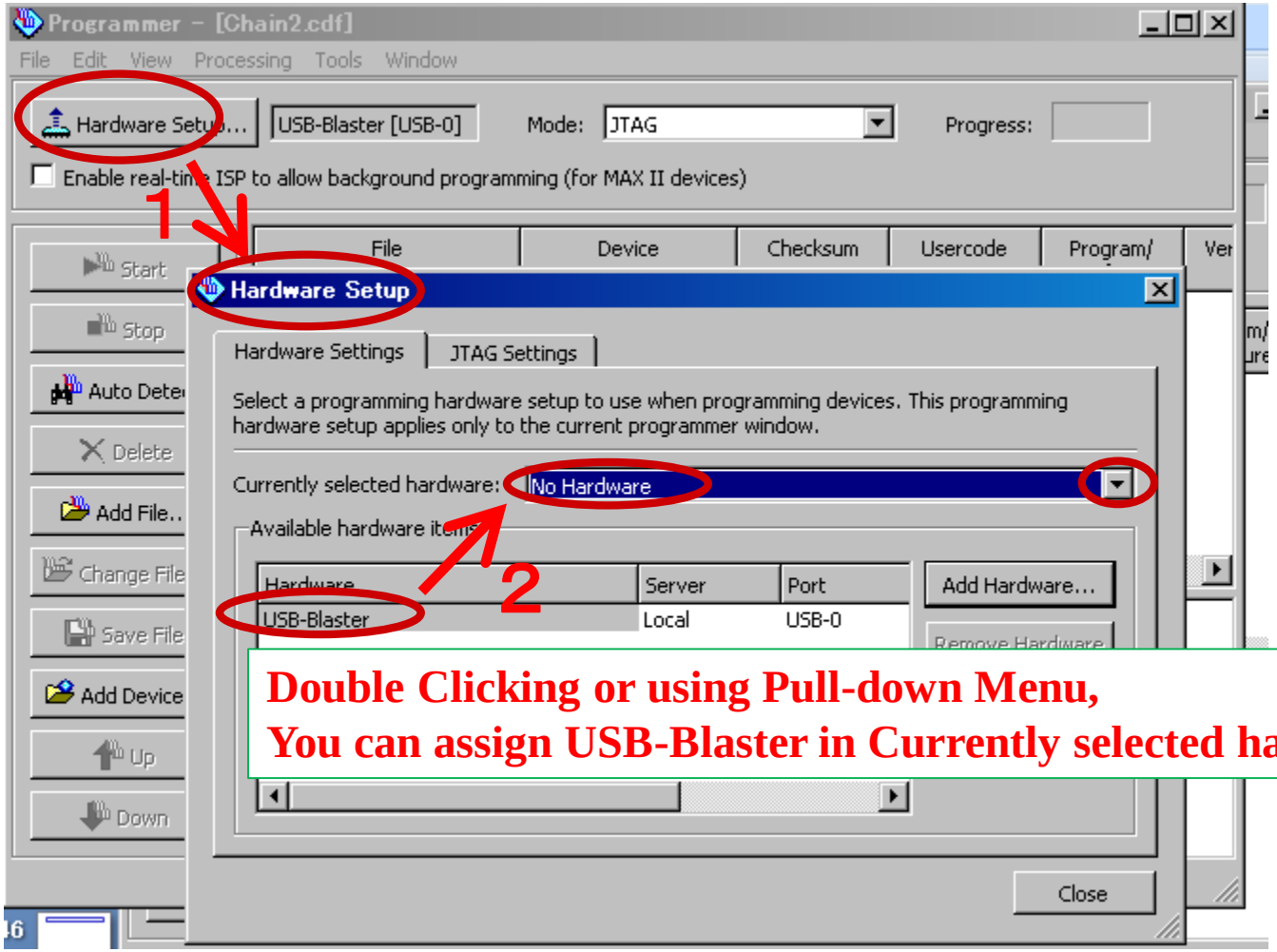
Hardware
→ USB-Blaster

Mode
→ JTAG



To specify USB-Blaster in Hardware Field,
Click ‘Hardware Setup’ Button.

Hardware Setup

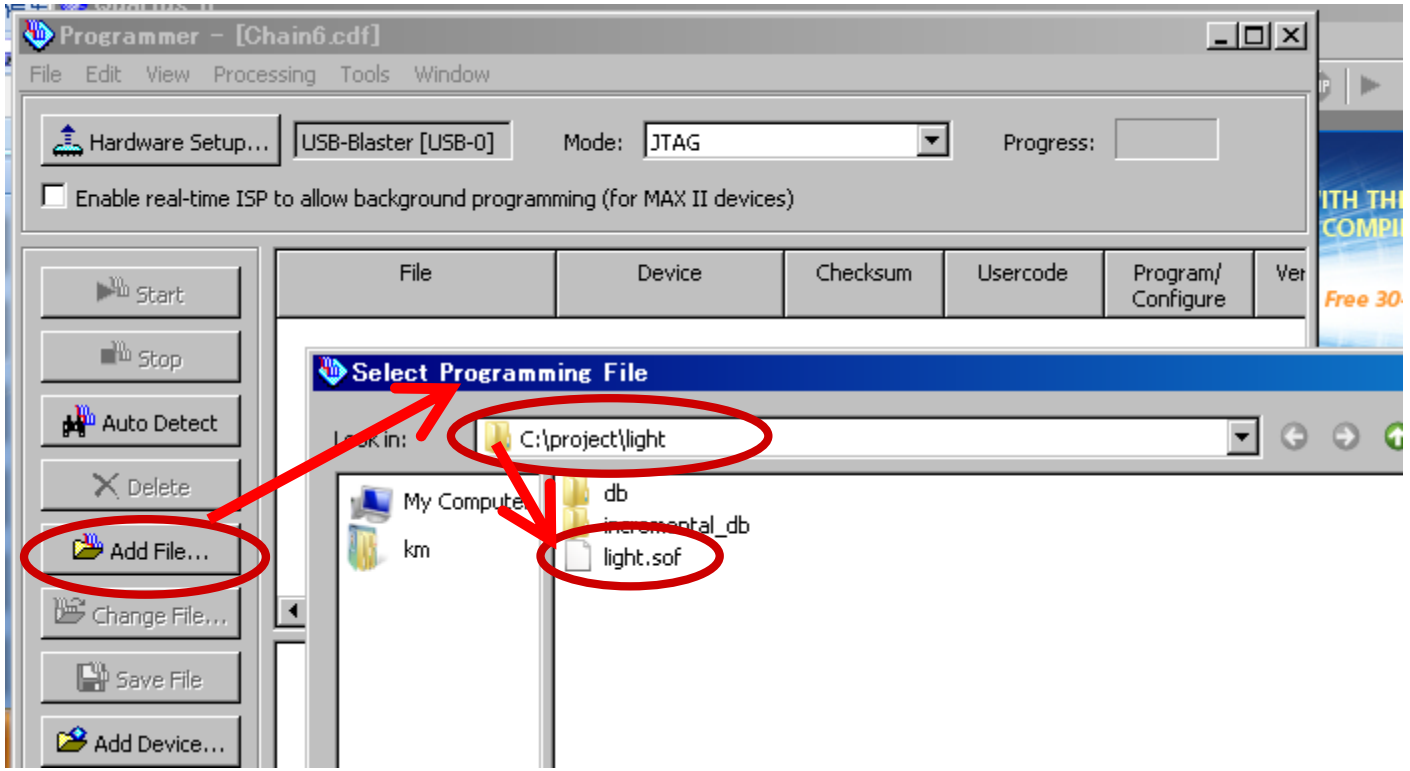


Double Clicking or using Pull-down Menu,
You can assign USB-Blaster in Currently selected hardware

3 **When 'Available hardware items' does not indicate USB-Blaster,**
Check ▪USB cable, ▪DE2-115 Power switch or ▪USB Driver

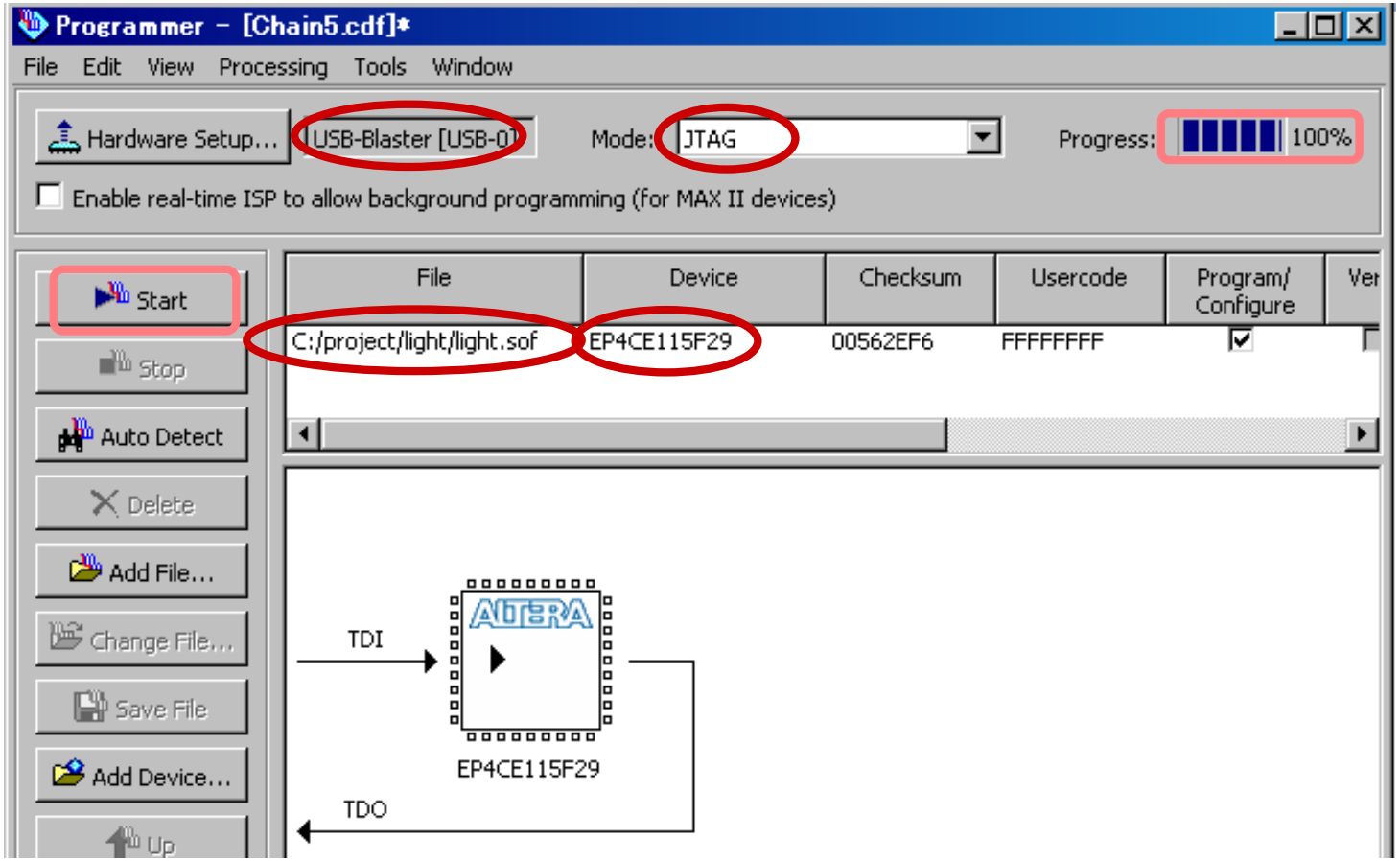
Specify the Configuration File (.sof) to be programmed onto FPGA.

1. **In Programmer Window, click 'Add File…' Button to open 'Select Programming File' Window.**
2. **Into Look in Box, Specify the Project Directory (assigned in Project Wizard of Quartus-II)**
3. **Select the file with the extension (sof), which was generated by the compiler of Quartus-II. (In this case, light.sof)**

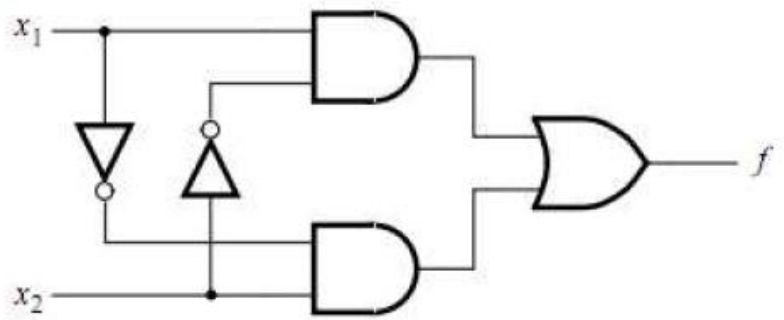


After checking 4 Points in Programmer Window, **Start !**

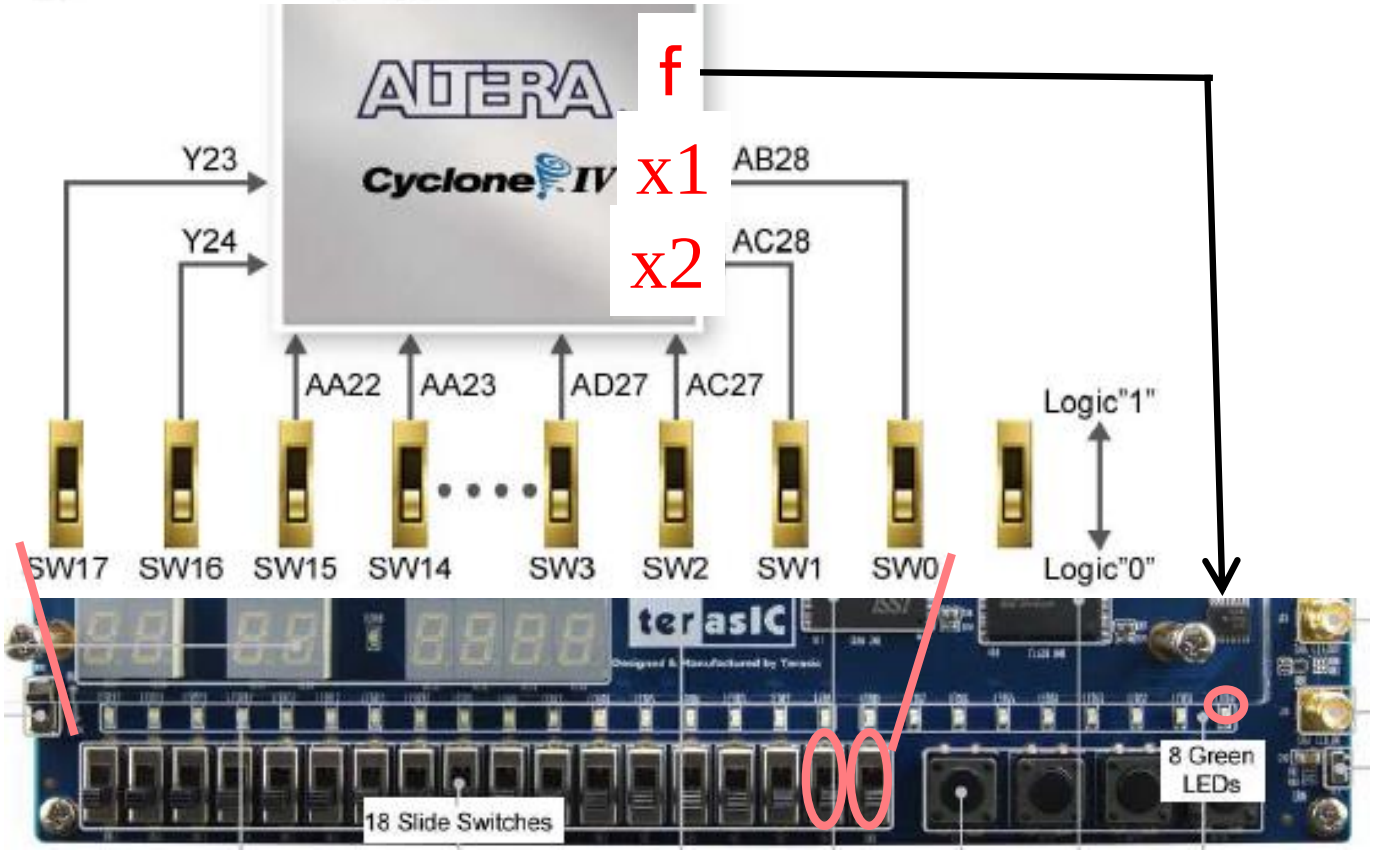
Progress Box in the upper-right corner shows the progress.
No error is displayed, the programming is successful.



Testing the FPGA board, any combination of SW0 and SW1,
What will become of LEDG0 ?



x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	0



Exercise : Modification of Project ‘light’

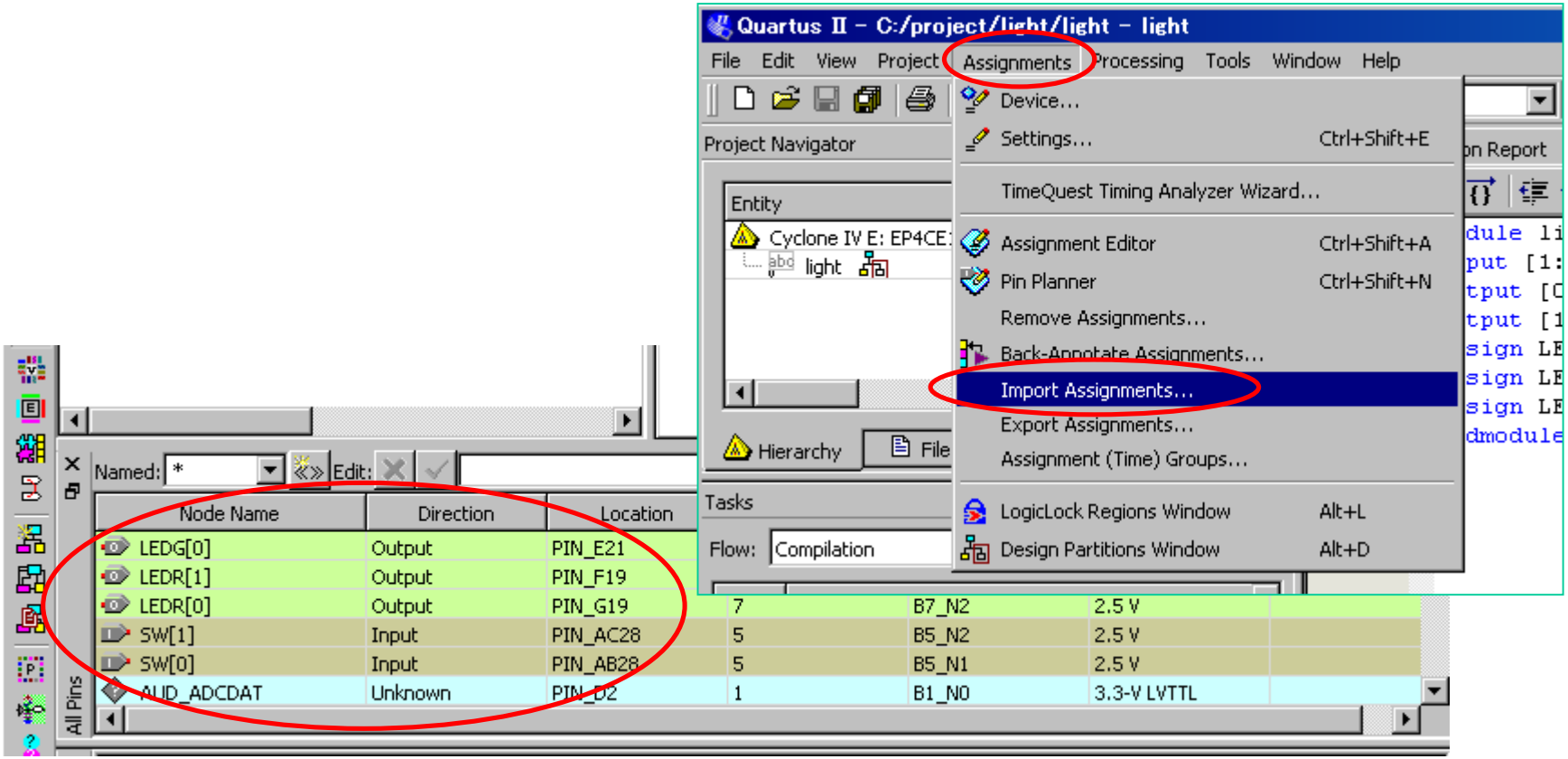
Original light.v

```
module light (x1, x2, f);
input x1, x2;
output f;
assign f = (x1 & ~x2) | (~x1 & x2);
endmodule
```

1. **Display the status of SW[0] and SW[1] (On/Off) on LEDR[0] and LEDR[1], respectively.**
2. **Change the logic of exclusiveOR to other logic such as OR or AND.**

After modification, compile again in Quartus-II

1. Using ‘Assignments > Import Assignments’ ,
   Import ‘DE2_115_pin_assignments’.
1. After Compile, confirm the Pin Assignment.



Re-use of Pin Assignment Information

When pin number is very large, Pin-Assignment becomes a tough job.

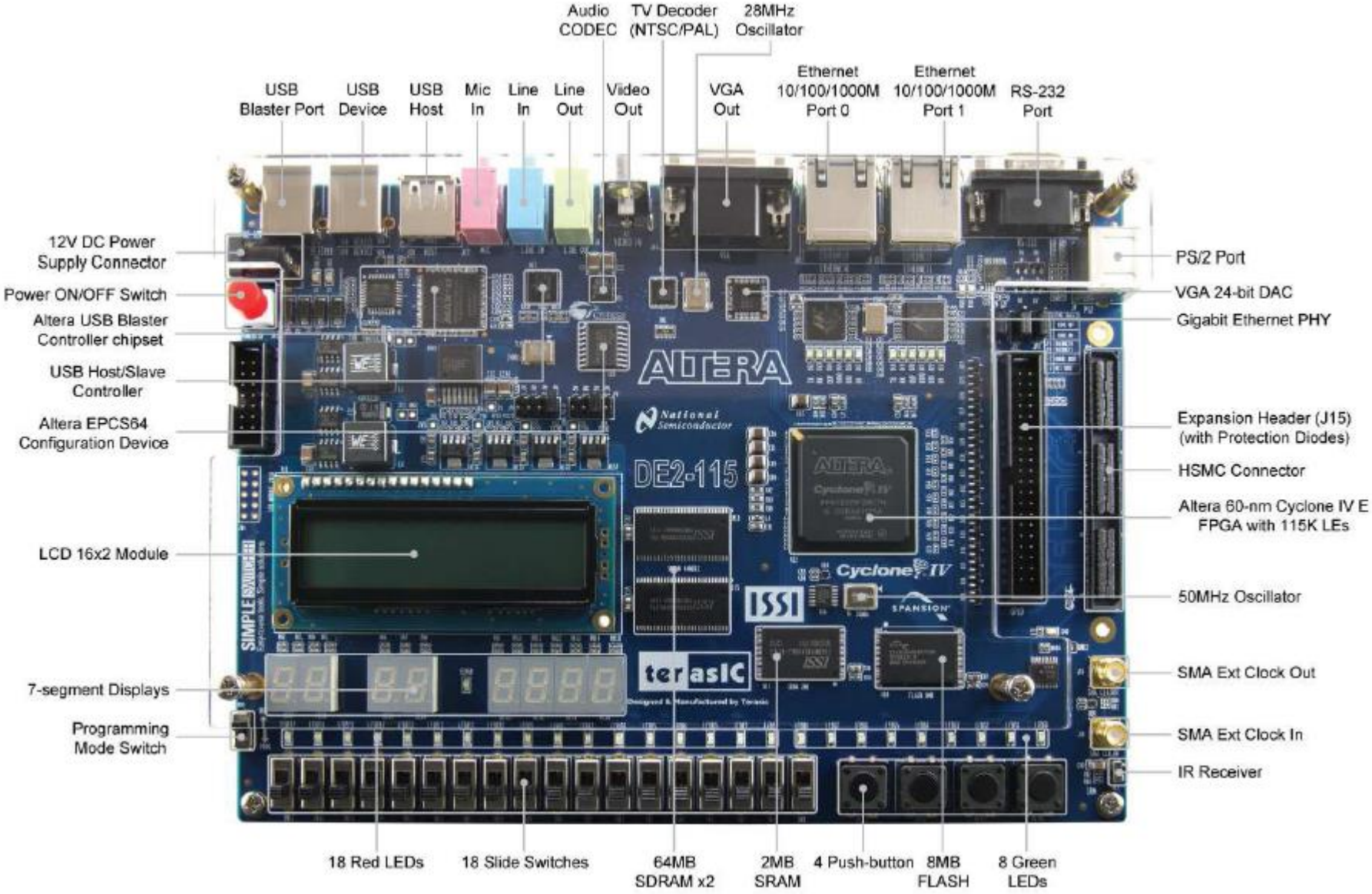
Fortunately, DE2-115 provides CSV file that describes the connection of
- all FPGA pins and
- all signals on the board.

File Name: DE2-115_pin_assignments.csv
Directory: in the system CDROM of DE2-115:
¥DE2-115_tutorials¥design_files
or web page of Altera DE2-115
You can import the assignment thru Quartus-IIusing 'Assignments>Import Assignments'.

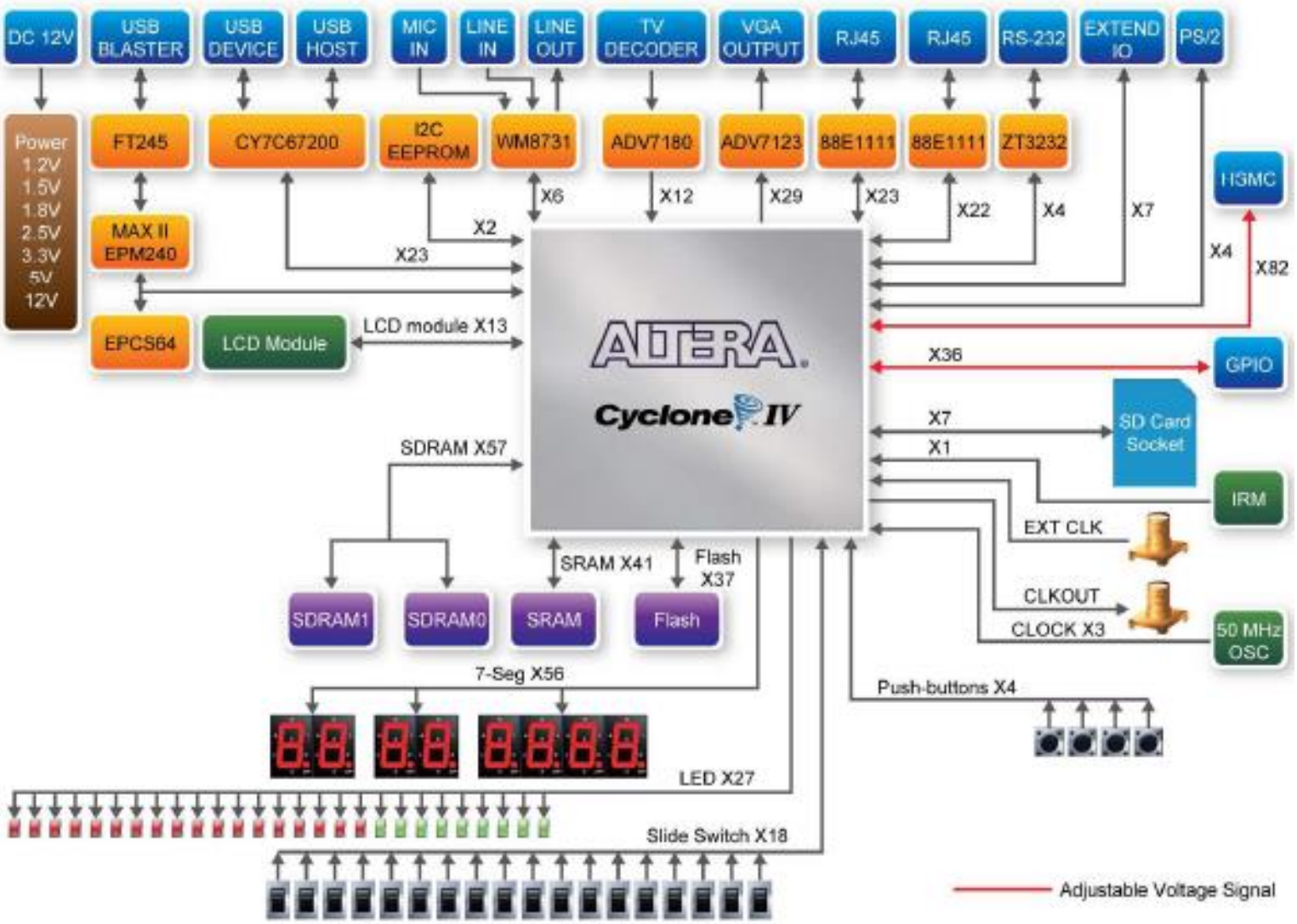
The important thing is to make the signal names agree between port names of design files and those used in DE2-115 User Manual.

In the following, the board design and pin characteristics of DE2-115 User Manual are briefly explained.

Brief introduction of DE2-115 board

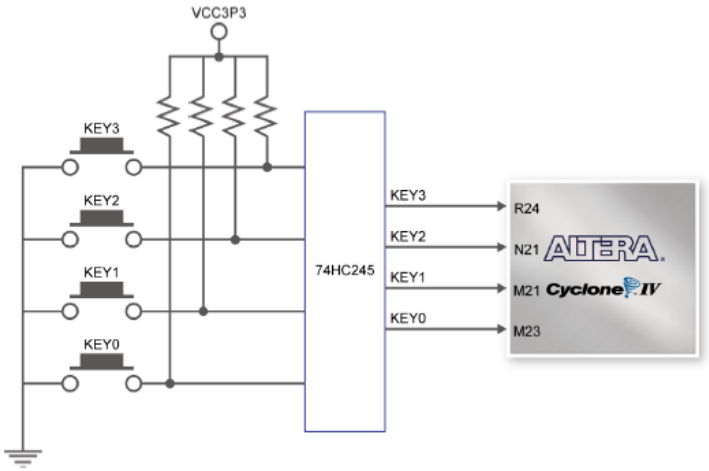


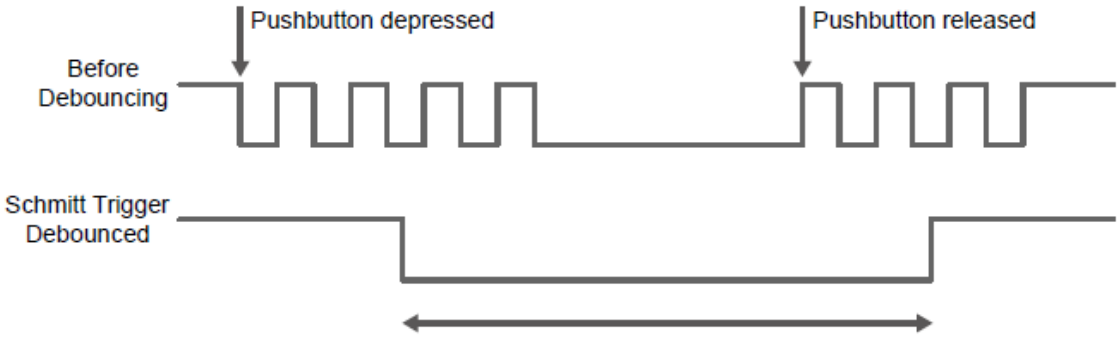
Block diagram of DE2-115



Push Switch

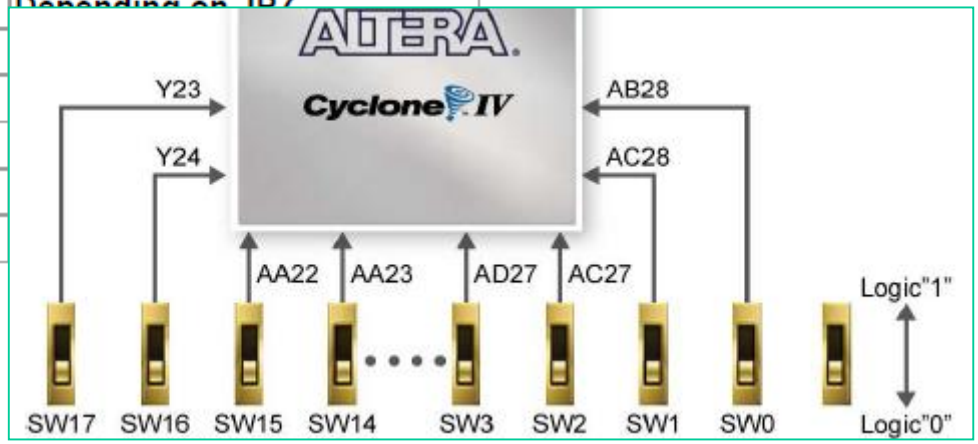
Signal Name	FPGA Pin No.	Description	I/O Standard
KEY[0]	PIN_M23	Push-button[0]	Depending on JP7
KEY[1]	PIN_M21	Push-button[1]	Depending on JP7
KEY[2]	PIN_N21	Push-button[2]	Depending on JP7
KEY[3]	PIN_R24	Push-button[3]	Depending on JP7





Slide Switch

Signal Name	FPGA Pin No.	Description	I/O Standard
SW[0]	PIN_AB28	Slide Switch[0]	Depending on JP7
SW[1]	PIN_AC28	Slide Switch[1]	Depending on JP7
SW[2]	PIN_AC27	Slide Switch[2]	Depending on JP7
SW[3]	PIN_AD27	Slide Switch[3]	Depending on JP7
SW[4]	PIN_AB27	Slide Switch[4]	Depending on JP7
SW[5]	PIN_AC26	Slide Switch[5]	Depending on JP7
SW[6]	PIN_AD26	Slide Switch[6]	Depending on JP7
SW[7]	PIN_AB26	Slide Switch[7]	Depending on JP7
SW[8]	PIN_AC25	Slide Switch[8]	Depending on JP7
SW[9]	PIN_AB25	Slide Switch[9]	Depending on JP7
SW[10]	PIN_AC24	Slide Switch[10]	Depending on JP7
SW[11]	PIN_AB24	Slide Switch[11]	Depending on JP7
SW[12]	PIN_AB23	Slide Switch[12]	Depending on JP7
SW[13]	PIN_AA24	Slide Switch[13]	
SW[14]	PIN_AA23	Slide Switch[14]	
SW[15]	PIN_AA22	Slide Switch[15]	
SW[16]	PIN_Y24	Slide Switch[16]	
SW[17]	PIN_Y23	Slide Switch[17]	



LEDs (Green & Red)

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
LEDG[0]	PIN_E21	LED Green[0]	2.5V
LEDG[1]	PIN_E22	LED Green[1]	2.5V
LEDG[2]	PIN_E25	LED Green[2]	2.5V
LEDG[3]	PIN_E24	LED Green[3]	2.5V
LEDG[4]	PIN_H21	LED Green[4]	2.5V
LEDG[5]	PIN_G20	LED Green[5]	2.5V
LEDG[6]	PIN_G22	LED Green[6]	2.5V
LEDG[7]	PIN_G21	LED Green[7]	2.5V
LEDG[8]	PIN_F17	LED Green[8]	2.5V

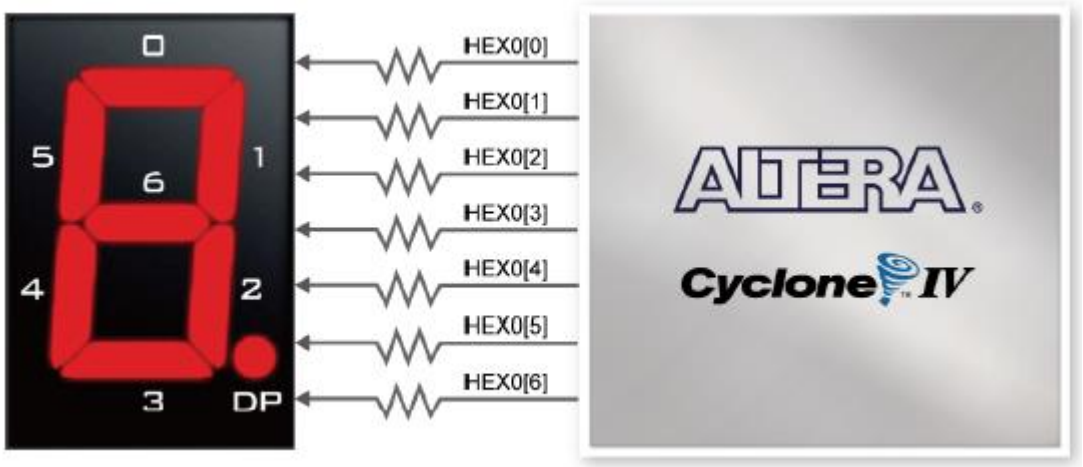
LEDs (Green & Red)

Signal Name	FPGA Pin No.	Description	I/O Standard
LEDR[0]	PIN_G19	LED Red[0]	2.5V
LEDR[1]	PIN_F19	LED Red[1]	2.5V
LEDR[2]	PIN_E19	LED Red[2]	2.5V
LEDR[3]	PIN_F21	LED Red[3]	2.5V
LEDR[4]	PIN_F18	LED Red[4]	2.5V
LEDR[5]	PIN_E18	LED Red[5]	2.5V
LEDR[6]	PIN_J19	LED Red[6]	2.5V
LEDR[7]	PIN_H19	LED Red[7]	2.5V
LEDR[8]	PIN_J17	LED Red[8]	2.5V
LEDR[9]	PIN_G17	LED Red[9]	2.5V
LEDR[10]	PIN_J15	LED Red[10]	2.5V
LEDR[11]	PIN_H16	LED Red[11]	2.5V
LEDR[12]	PIN_J16	LED Red[12]	2.5V
LEDR[13]	PIN_H17	LED Red[13]	2.5V
LEDR[14]	PIN_F15	LED Red[14]	2.5V
LEDR[15]	PIN_G15	LED Red[15]	2.5V
LEDR[16]	PIN_G16	LED Red[16]	2.5V
LEDR[17]	PIN_H15	LED Red[17]	2.5V

7-segment Displays-1

8 letters（HEX0[0:6]~HEX7[0:6]）of 7-segment

Glows with logic ”0”, turn off with logic “1”.



Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
HEX0[0]	PIN_G18	Seven Segment Digit 0[0]	2.5V
HEX0[1]	PIN_F22	Seven Segment Digit 0[1]	2.5V
HEX0[2]	PIN_E17	Seven Segment Digit 0[2]	2.5V
HEX0[3]	PIN_L26	Seven Segment Digit 0[3]	Depending on JP7
HEX0[4]	PIN_L25	Seven Segment Digit 0[4]	Depending on JP7
HEX0[5]	PIN_J22	Seven Segment Digit 0[5]	Depending on JP7
HEX0[6]	PIN_H22	Seven Segment Digit 0[6]	Depending on JP7

7-segment Displays-2

HEX1[0]	PIN_M24	Seven Segment Digit 1[0]	Depending on JP7
HEX1[1]	PIN_Y22	Seven Segment Digit 1[1]	Depending on JP7
HEX1[2]	PIN_W21	Seven Segment Digit 1[2]	Depending on JP7
HEX1[3]	PIN_W22	Seven Segment Digit 1[3]	Depending on JP7
HEX1[4]	PIN_W25	Seven Segment Digit 1[4]	Depending on JP7
HEX1[5]	PIN_U23	Seven Segment Digit 1[5]	Depending on JP7
HEX1[6]	PIN_U24	Seven Segment Digit 1[6]	Depending on JP7
HEX2[0]	PIN_AA25	Seven Segment Digit 2[0]	Depending on JP7
HEX2[1]	PIN_AA26	Seven Segment Digit 2[1]	Depending on JP7
HEX2[2]	PIN_Y25	Seven Segment Digit 2[2]	Depending on JP7
HEX2[3]	PIN_W26	Seven Segment Digit 2[3]	Depending on JP7
HEX2[4]	PIN_Y26	Seven Segment Digit 2[4]	Depending on JP7
HEX2[5]	PIN_W27	Seven Segment Digit 2[5]	Depending on JP7
HEX2[6]	PIN_W28	Seven Segment Digit 2[6]	Depending on JP7
HEX3[0]	PIN_V21	Seven Segment Digit 3[0]	Depending on JP7
HEX3[1]	PIN_U21	Seven Segment Digit 3[1]	Depending on JP7
HEX3[2]	PIN_AB20	Seven Segment Digit 3[2]	Depending on JP6
HEX3[3]	PIN_AA21	Seven Segment Digit 3[3]	Depending on JP6
HEX3[4]	PIN_AD24	Seven Segment Digit 3[4]	Depending on JP6
HEX3[5]	PIN_AF23	Seven Segment Digit 3[5]	Depending on JP6
HEX3[6]	PIN_Y19	Seven Segment Digit 3[6]	Depending on JP6
HEX4[0]	PIN_AB19	Seven Segment Digit 4[0]	Depending on JP6
HEX4[1]	PIN_AA19	Seven Segment Digit 4[1]	Depending on JP6
HEX4[2]	PIN_AG21	Seven Segment Digit 4[2]	Depending on JP6
HEX4[3]	PIN_AH21	Seven Segment Digit 4[3]	Depending on JP6

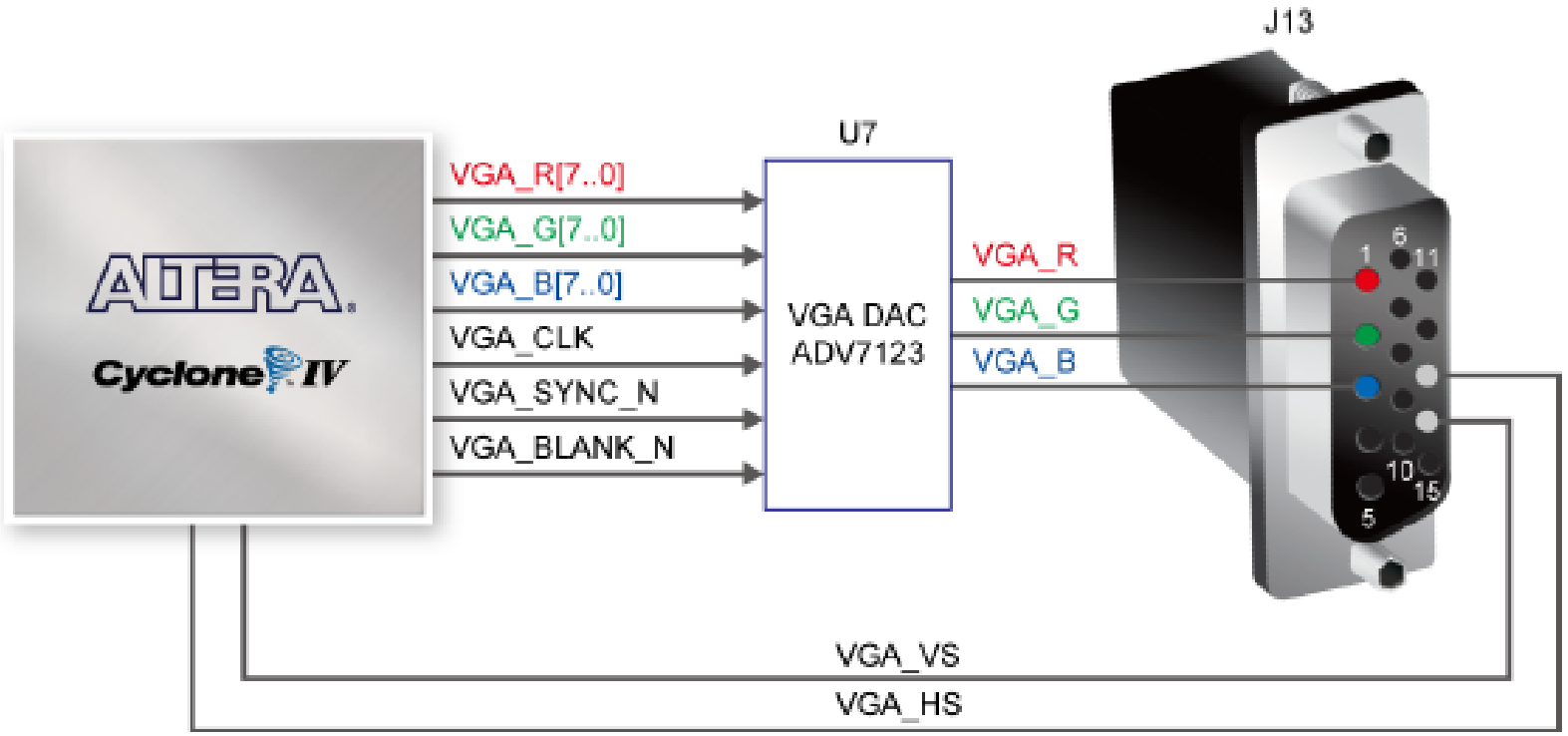
7-segment Displays-3

HEX4[4]	PIN_AE19	Seven Segment Digit 4[4]	Depending on JP6
HEX4[5]	PIN_AF19	Seven Segment Digit 4[5]	Depending on JP6
HEX4[6]	PIN_AE18	Seven Segment Digit 4[6]	Depending on JP6
HEX5[0]	PIN_AD18	Seven Segment Digit 5[0]	Depending on JP6
HEX5[1]	PIN_AC18	Seven Segment Digit 5[1]	Depending on JP6
HEX5[2]	PIN_AB18	Seven Segment Digit 5[2]	Depending on JP6
HEX5[3]	PIN_AH19	Seven Segment Digit 5[3]	Depending on JP6
HEX5[4]	PIN_AG19	Seven Segment Digit 5[4]	Depending on JP6
HEX5[5]	PIN_AF18	Seven Segment Digit 5[5]	Depending on JP6
HEX5[6]	PIN_AH18	Seven Segment Digit 5[6]	Depending on JP6
HEX6[0]	PIN_AA17	Seven Segment Digit 6[0]	Depending on JP6
HEX6[1]	PIN_AB16	Seven Segment Digit 6[1]	Depending on JP6
HEX6[2]	PIN_AA16	Seven Segment Digit 6[2]	Depending on JP6
HEX6[3]	PIN_AB17	Seven Segment Digit 6[3]	Depending on JP6
HEX6[4]	PIN_AB15	Seven Segment Digit 6[4]	Depending on JP6
HEX6[5]	PIN_AA15	Seven Segment Digit 6[5]	Depending on JP6
HEX6[6]	PIN_AC17	Seven Segment Digit 6[6]	Depending on JP6
HEX7[0]	PIN_AD17	Seven Segment Digit 7[0]	Depending on JP6
HEX7[1]	PIN_AE17	Seven Segment Digit 7[1]	Depending on JP6
HEX7[2]	PIN_AG17	Seven Segment Digit 7[2]	Depending on JP6
HEX7[3]	PIN_AH17	Seven Segment Digit 7[3]	Depending on JP6
HEX7[4]	PIN_AF17	Seven Segment Digit 7[4]	Depending on JP6
HEX7[5]	PIN_AG18	Seven Segment Digit 7[5]	Depending on JP6
HEX7[6]	PIN_AA14	Seven Segment Digit 7[6]	3.3V

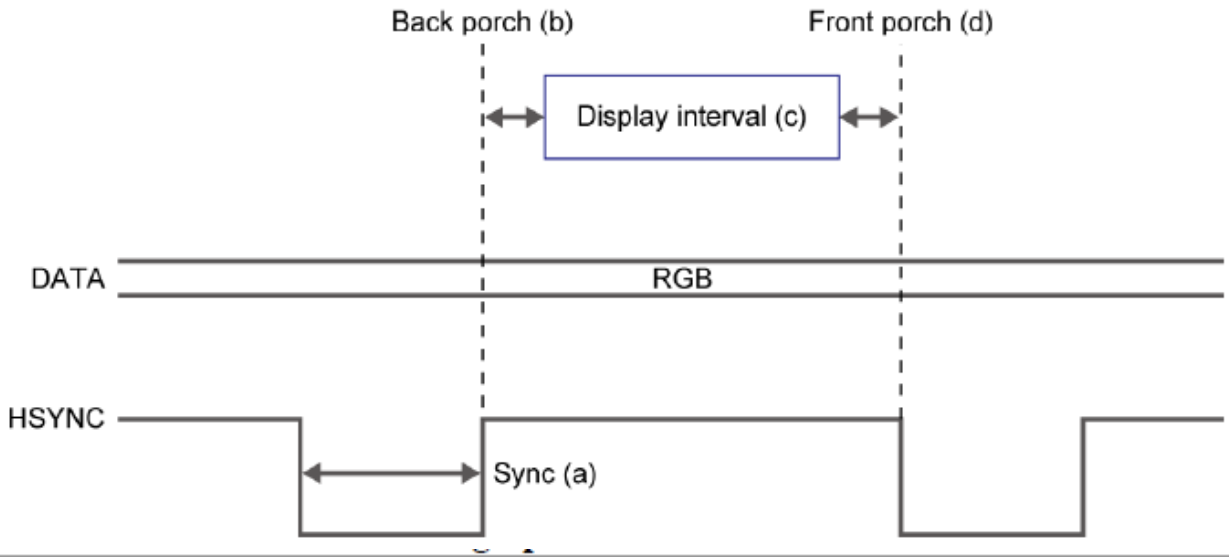
Clock Inputs/ Output

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
CLOCK_50	PIN_Y2	50 MHz clock input	3.3V
CLOCK2_50	PIN_AG14	50 MHz clock input	3.3V
CLOCK3_50	PIN_AG15	50 MHz clock input	Depending on JP6
SMA_CLKOUT	PIN_AE23	External (SMA) clock output	Depending on JP6
SMA_CLKIN	PIN_AH14	External (SMA) clock input	3.3V

VGA

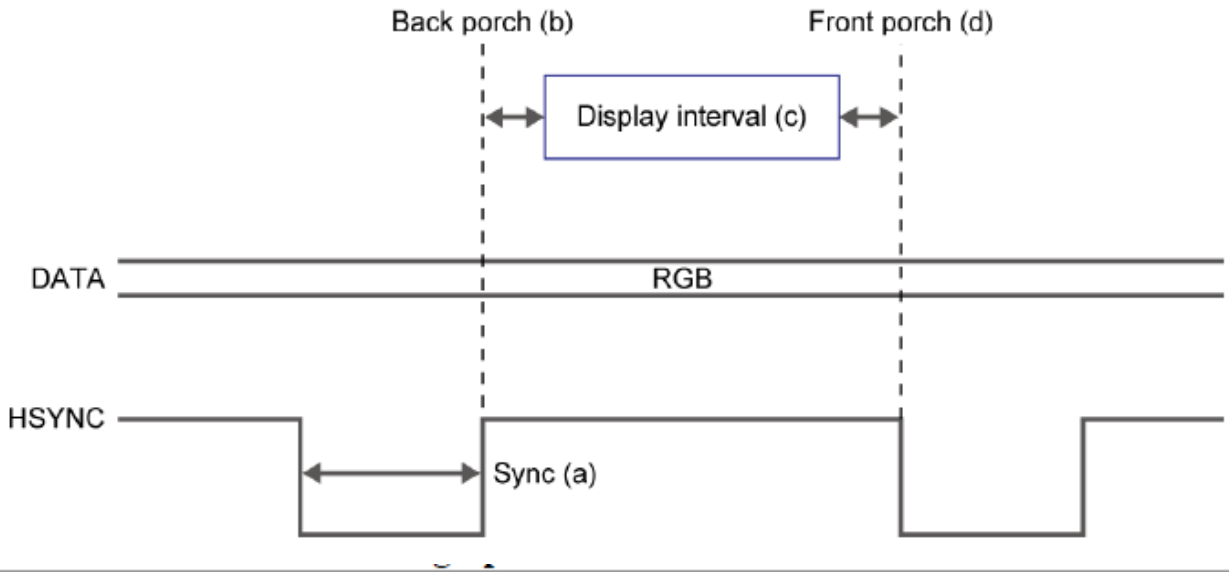

J13
U7
VGA_R[7..0]
VGA_G[7..0]
VGA_B[7..0]
VGA_CLK
VGA_SYNC_N
VGA_BLANK_N
VGA DAC
ADV7123
VGA_R
VGA_G
VGA_B
1
6
11
5
10
15
VGA_VS
VGA_HS

VGA Horizontal timing



VGA mode		*Horizontal Timing Spec*				
Configuration	*Resolution(HxV)*	*a(us)*	*b(us)*	*c(us)*	*d(us)*	*Pixel clock(MHz)*
VGA(60Hz)	640x480	3.8	1.9	25.4	0.6	25
VGA(85Hz)	640x480	1.6	2.2	17.8	1.6	36
SVGA(60Hz)	800x600	3.2	2.2	20	1	40
SVGA(75Hz)	800x600	1.6	3.2	16.2	0.3	49
SVGA(85Hz)	800x600	1.1	2.7	14.2	0.6	56
XGA(60Hz)	1024x768	2.1	2.5	15.8	0.4	65
XGA(70Hz)	1024x768	1.8	1.9	13.7	0.3	75
XGA(85Hz)	1024x768	1.0	2.2	10.8	0.5	95
1280x1024(60Hz)	1280x1024	1.0	2.3	11.9	0.4	108

VGA Vertical Timing

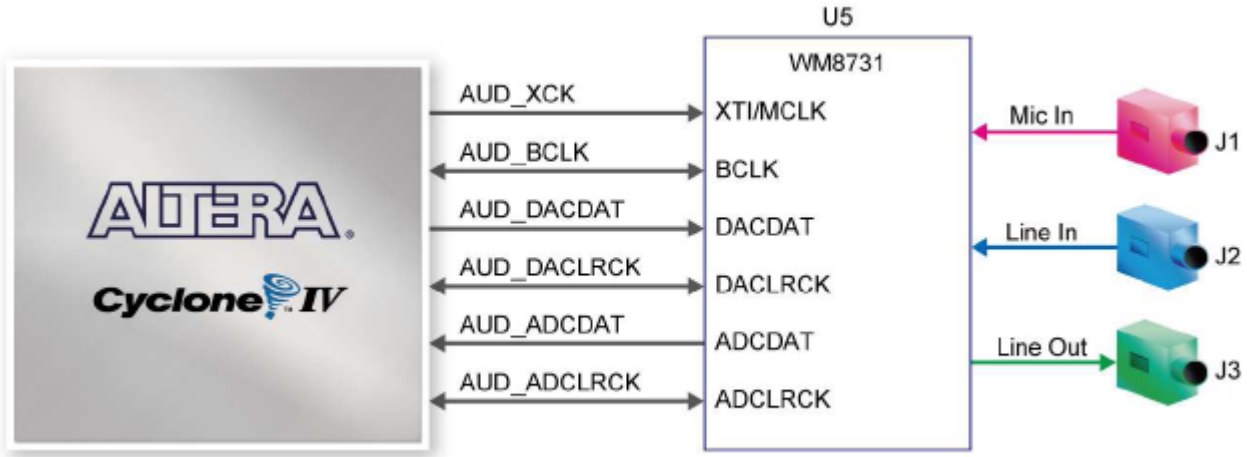


VGA mode		Vertical Timing Spec				
Configuration	Resolution(HxV)	a(lines)	b(lines)	c(lines)	d(lines)	Pixel clock(MHz)
VGA(60Hz)	640x480	2	33	480	10	25
VGA(85Hz)	640x480	3	25	480	1	36
SVGA(60Hz)	800x600	4	23	600	1	40
SVGA(75Hz)	800x600	3	21	600	1	49
SVGA(85Hz)	800x600	3	27	600	1	56
XGA(60Hz)	1024x768	6	29	768	3	65
XGA(70Hz)	1024x768	6	29	768	3	75
XGA(85Hz)	1024x768	3	36	768	1	95
1280x1024(60Hz)	1280x1024	3	38	1024	1	108

Pin Assignment for VGA

Signal Name	FPGA Pin No.	Description	I/O Standard
VGA_R[0]	PIN_E12	VGA Red[0]	3.3V
VGA_R[1]	PIN_E11	VGA Red[1]	3.3V
VGA_R[2]	PIN_D10	VGA Red[2]	3.3V
VGA_R[3]	PIN_F12	VGA Red[3]	3.3V
VGA_R[4]	PIN_G10	VGA Red[4]	3.3V
VGA_R[5]	PIN_J12	VGA Red[5]	3.3V
VGA_R[6]	PIN_H8	VGA Red[6]	3.3V
VGA_R[7]	PIN_H10	VGA Red[7]	3.3V
VGA_G[0]	PIN_G8	VGA Green[0]	3.3V
VGA_G[1]	PIN_G11	VGA Green[1]	3.3V
VGA_G[2]	PIN_F8	VGA Green[2]	3.3V
VGA_G[3]	PIN_H12	VGA Green[3]	3.3V
VGA_G[4]	PIN_C8	VGA Green[4]	3.3V
VGA_G[5]	PIN_B8	VGA Green[5]	3.3V
VGA_G[6]	PIN_F10	VGA Green[6]	3.3V
VGA_G[7]	PIN_C9	VGA Green[7]	3.3V
VGA_B[0]	PIN_B10	VGA Blue[0]	3.3V
VGA_B[1]	PIN_A10	VGA Blue[1]	3.3V
VGA_B[2]	PIN_C11	VGA Blue[2]	3.3V
VGA_B[3]	PIN_B11	VGA Blue[3]	3.3V
VGA_B[4]	PIN_A11	VGA Blue[4]	3.3V
VGA_B[5]	PIN_C12	VGA Blue[5]	3.3V
VGA_B[6]	PIN_D11	VGA Blue[6]	3.3V
VGA_B[7]	PIN_D12	VGA Blue[7]	3.3V
VGA_CLK	PIN_A12	VGA Clock	3.3V
VGA_BLANK_N	PIN_F11	VGA BLANK	3.3V
VGA_HS	PIN_G13	VGA H_SYNC	3.3V
VGA_VS	PIN_C13	VGA V_SYNC	3.3V
VGA_SYNC_N	PIN_C10	VGA SYNC	3.3V

24bit Audio CODEC



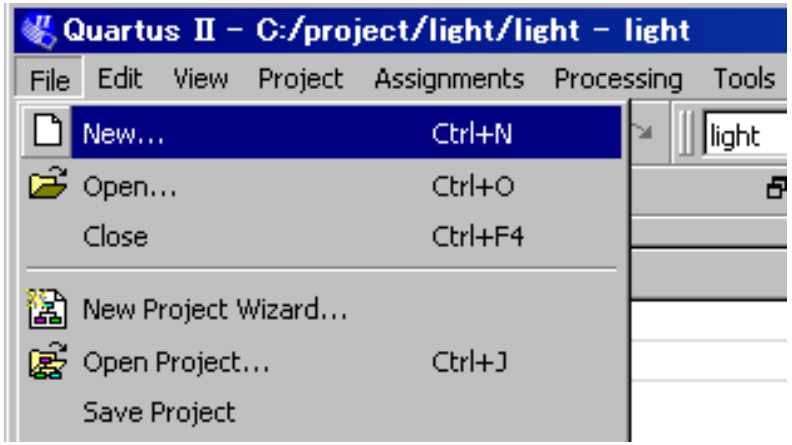
Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
AUD_ADCLRCK	PIN_C2	Audio CODEC ADC LR Clock	3.3V
AUD_ADCDAT	PIN_D2	Audio CODEC ADC Data	3.3V
AUD_DACLRCK	PIN_E3	Audio CODEC DAC LR Clock	3.3V
AUD_DACDAT	PIN_D1	Audio CODEC DAC Data	3.3V
AUD_XCK	PIN_E1	Audio CODEC Chip Clock	3.3V
AUD_BCLK	PIN_F2	Audio CODEC Bit-Stream Clock	3.3V
I2C_SCLK	PIN_B7	I2C Clock	3.3V
I2C_SDAT	PIN_A8	I2C Data	3.3V

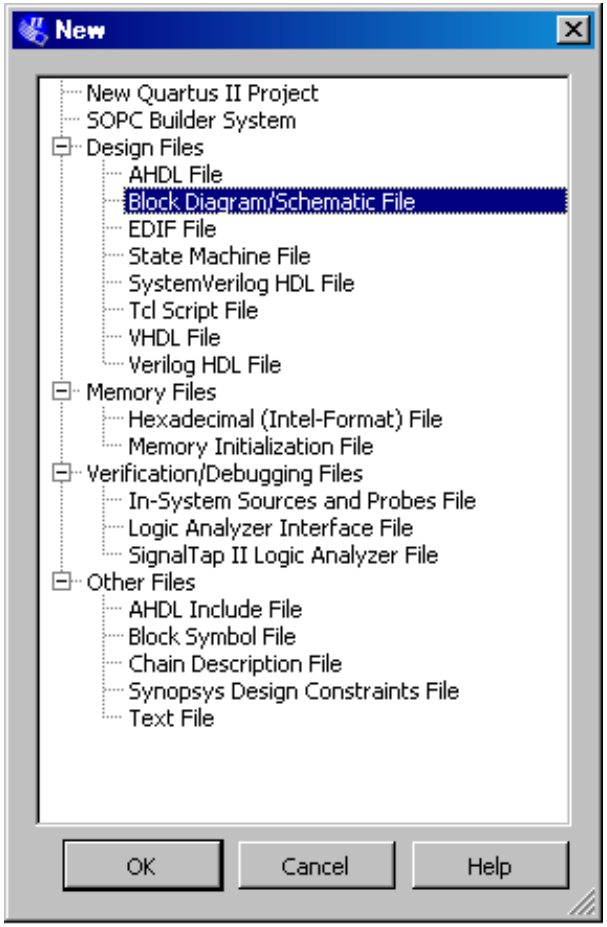
Experience here the traditional way of logic design

- Schematic entry of the project ‘lignht’ using graphic editor function of Quartus-II.

- Kick-off the new project of “old” , just like the project “light”.
- ‘File > New’ opens the pop-up window shown in the right-hand.
- Then select ’Design Files > Block Diagram/Schematic File’ and click OK.

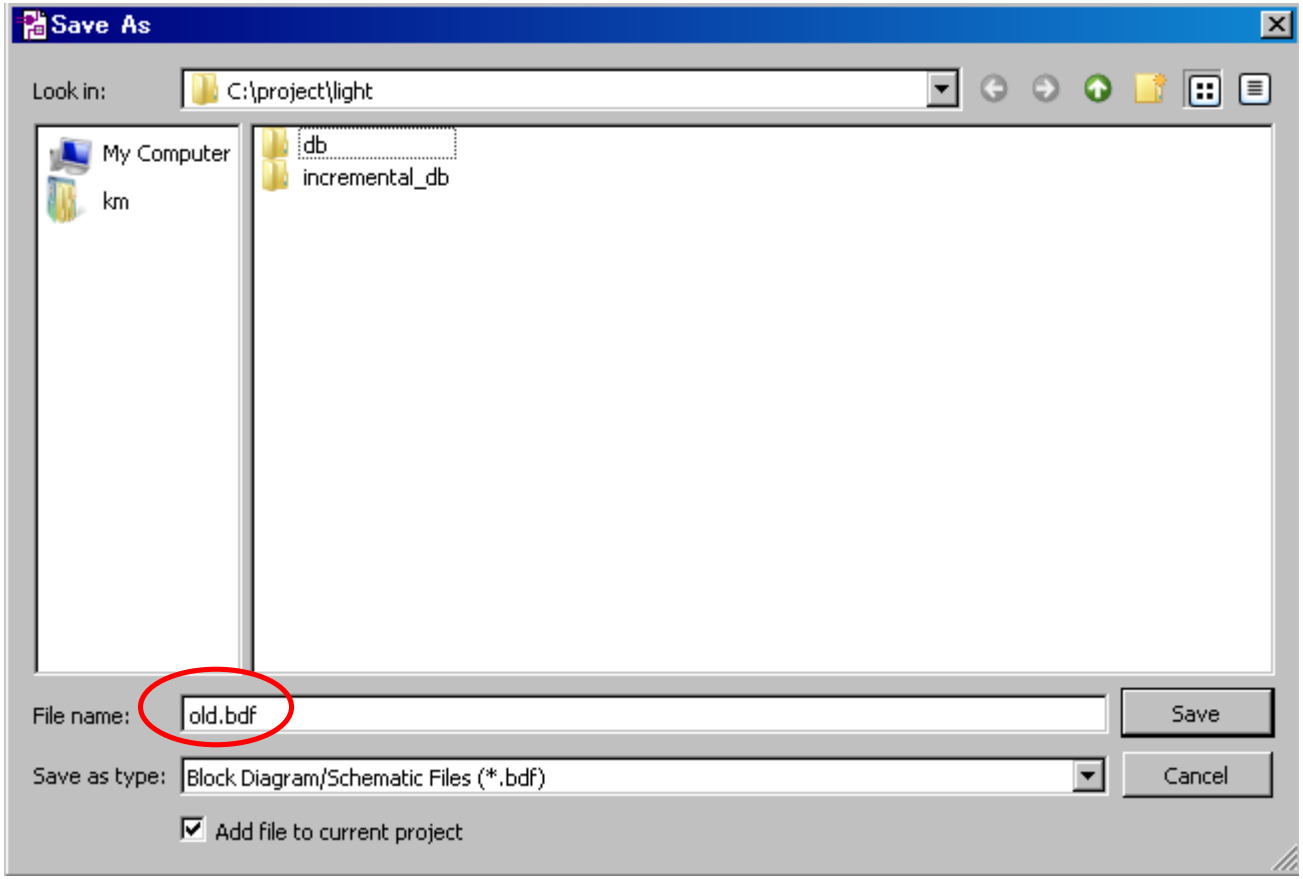
In the project “light”, we chose Verilog HDL File instead of Block Diagram/Schematic File.



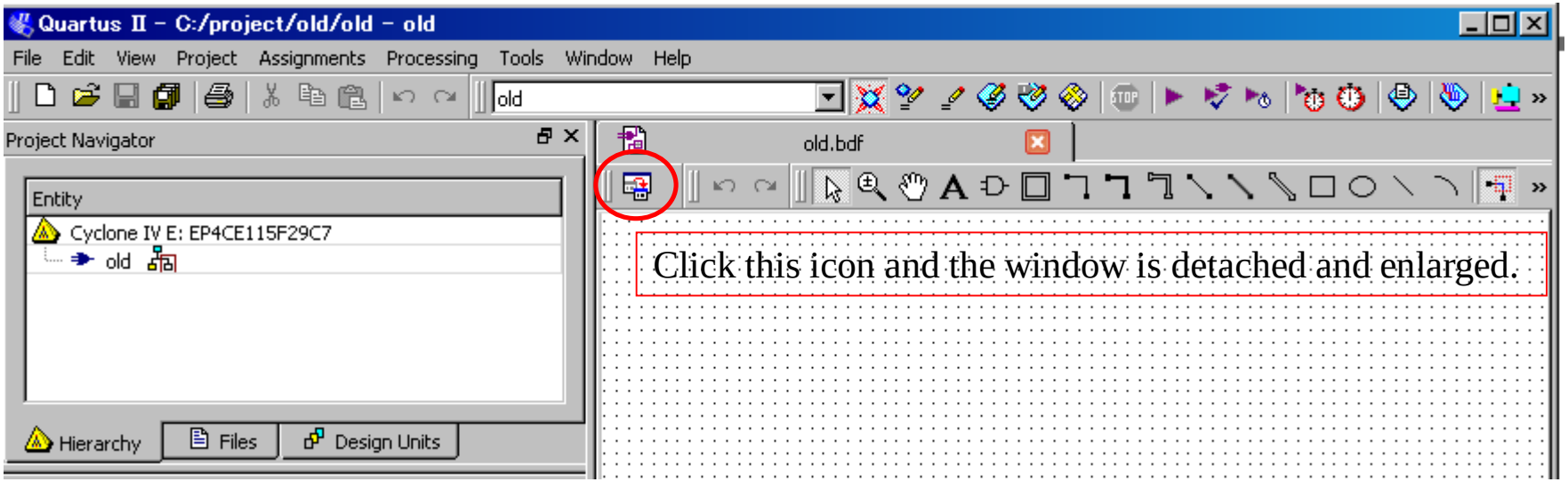


Make an empty bdf（Block Diagram/Schematic File)

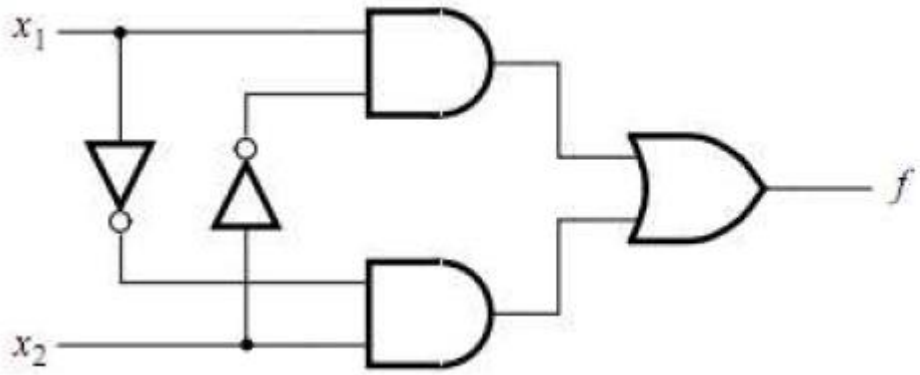
- In the Graphic Editor Window, Specify the File Name and Save.
- The File Name should be the same as Project Name with extension (.bdf) in this case: old.bdf



the Graphic Editor

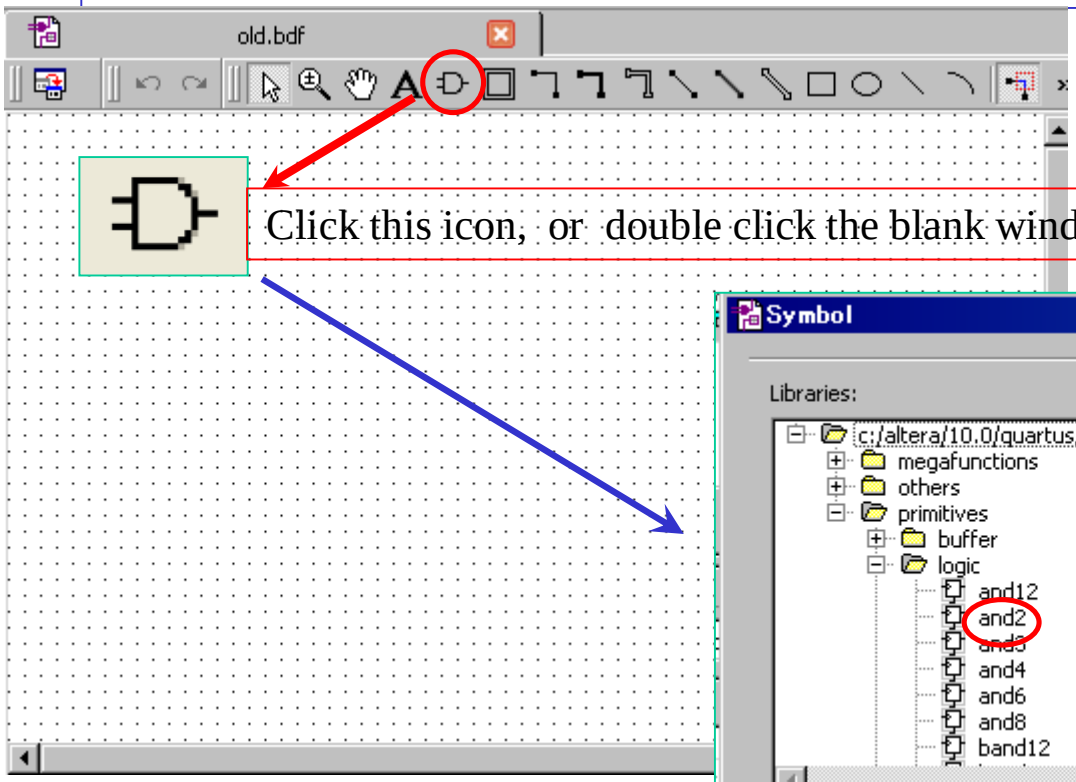


In this window we do schematic entry of the exclusive OR circuit below.

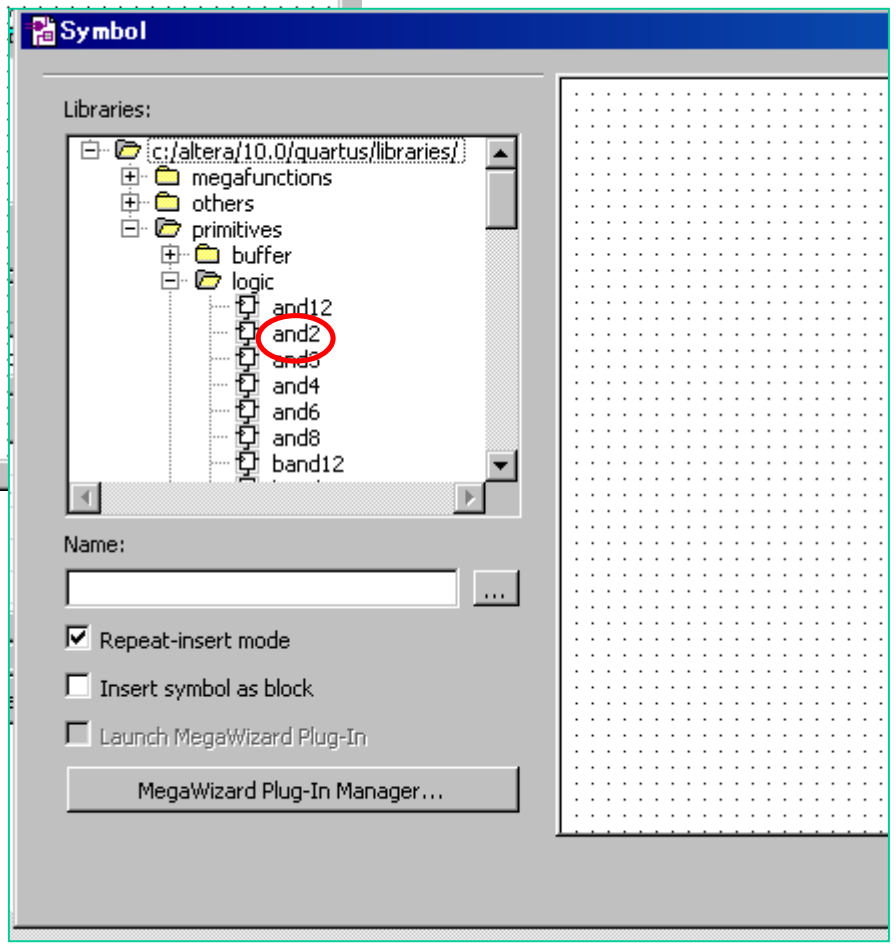


x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	0

Placement of Logic-Gate Symbol



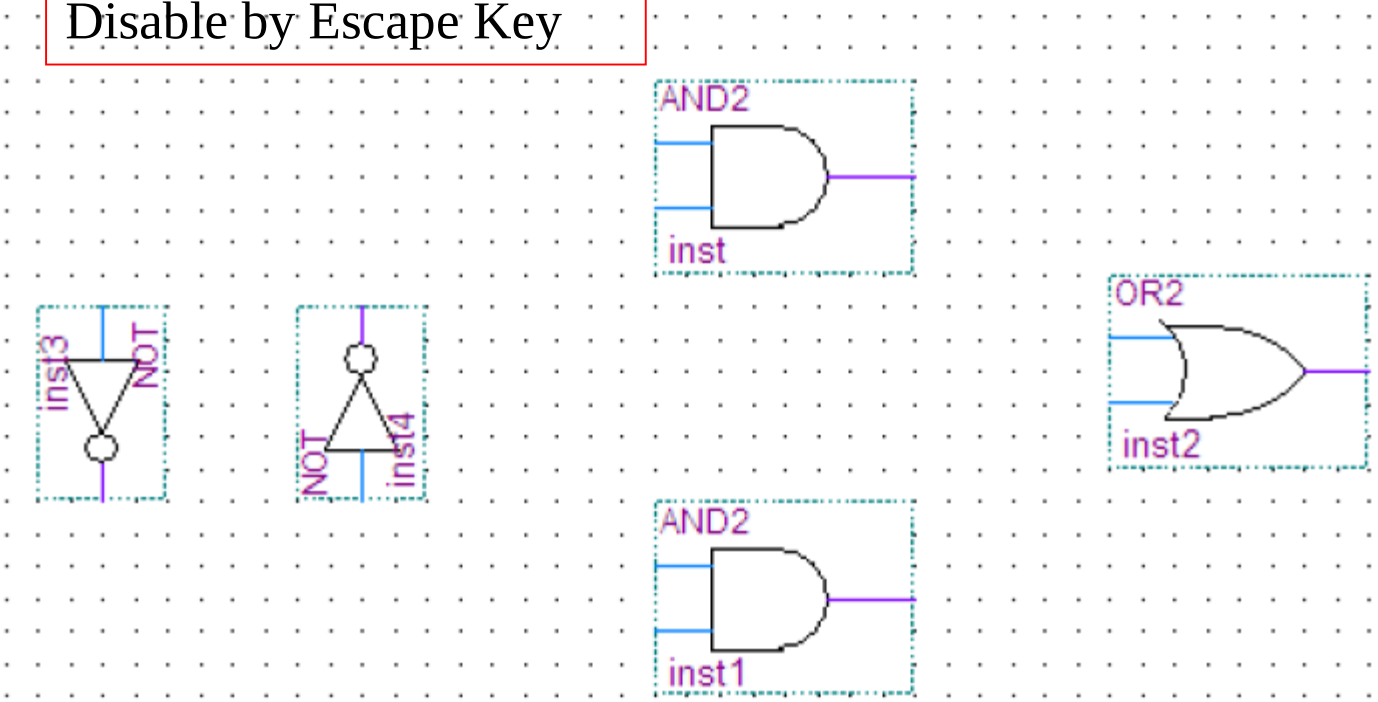
Click this icon, or double click the blank window, and you get the directory of library.



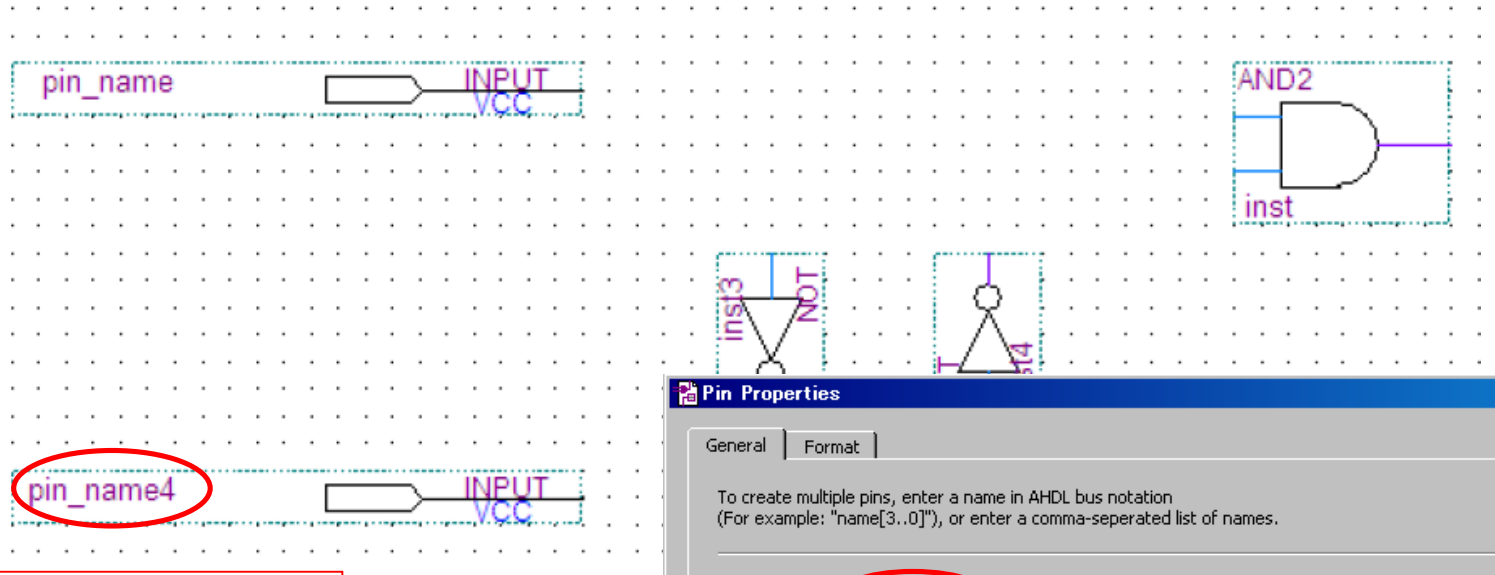
Pick the necessary gates form Primitives > Logic

2 input AND Gate : and2
2 input OR Gate : or2
NOT Gate : not

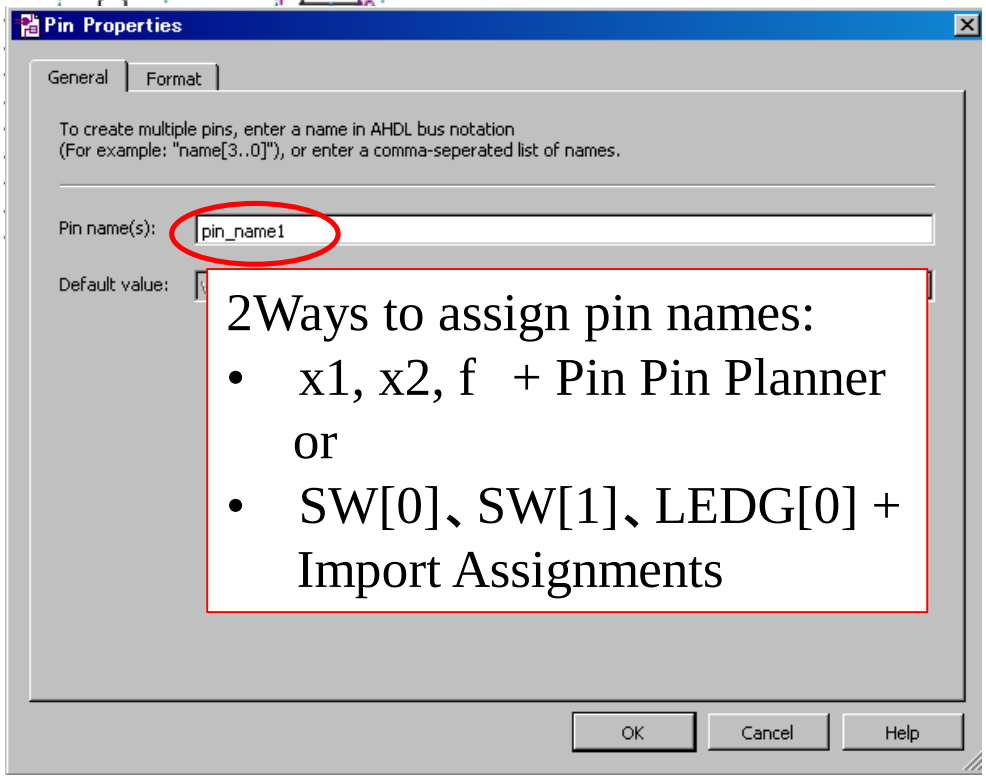
Disable by Escape Key



Select and assign I/O symbols from Primitives>Pin



Double Click !

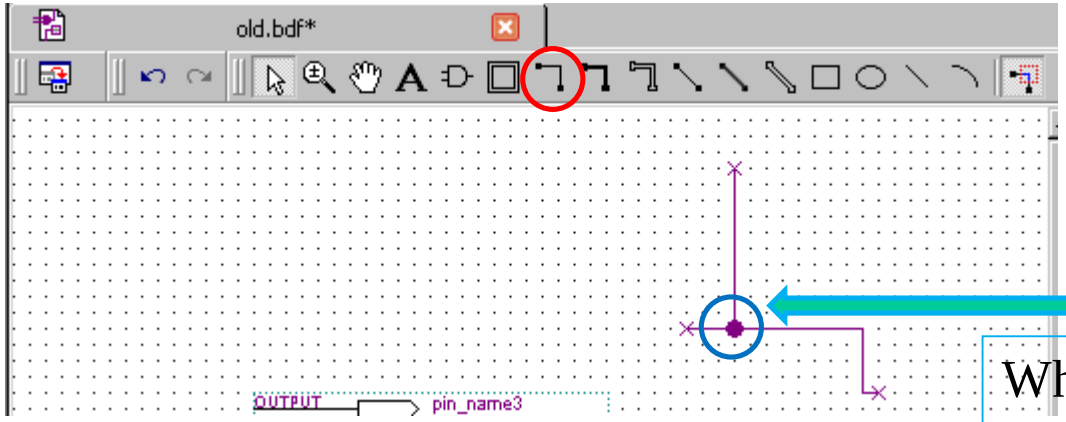


2Ways to assign pin names:

- x1, x2, f + Pin Pin Planner

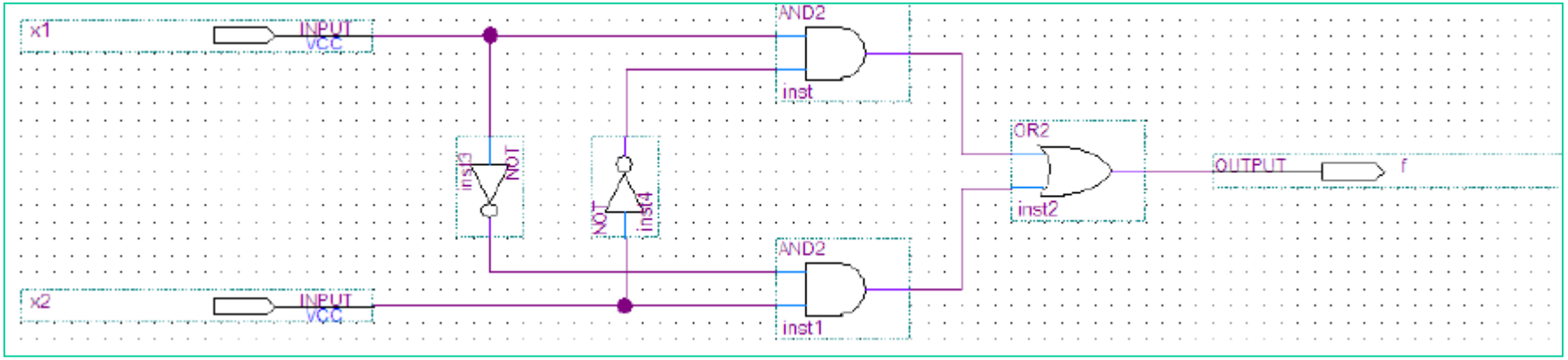
  or
- SW[0]、SW[1]、LEDG[0] + Import Assignments

Connect the devices and I/O → Operate DE2-115 !



Click  In the Tool Bar !

When wires are connected, dot is formed.



When schematic is finished, like project 'light' , assign pins (if not yet), compile, and program onto DE2-115, and verify the operation !!