My First FPGA for Altera DE2-115 Board

Digital Circuit Lab

TA: Po-Chen Wu

Outline

- Complete Your Verilog Design
- Assign The Device
- Add a PLL Megafunction
- Assign the Pins
- Create a Default TimeQuest SDC File
- Compile and Verify Your Design
- Configuring the Cyclone IV E FPGA

Complete Your Verilog Design

exp1_traffic.v (1/5)

It is a 10 seconds countdown system.

```
module exp1_traffic (
    clk,
    rst_n,
    pause,
    HEX0
);

//==== parameter definition ================================
    // for finite state machine
    parameter S_NORMAL = 1'd0;
    parameter S_PAUSE  = 1'd1;

    // for countdown
    parameter C_PERIOD = 4'd9;

//==== in/out declaration ==================================
    //-------- input --------------------------
    input clk;
    input rst_n; // reset signal (button)
    input pause; // pause signal (switch)

    //-------- output -------------------------------------
    output [6:0] HEX0;
```

module name = file name

The countdown system can be paused by turning on the switch.

exp1_traffic.v (2/5)

```
//==== reg/wire declaration =================================
    //-------- output ----------------------------------------
    reg [6:0] HEX0;

    //-------- wires -----------------------------------------
    wire clk_16; // 16MHz clock signal
    wire [23:0] next_clks;
    reg         next_state;
    reg   [3:0] next_countdown;
    reg   [6:0] next_HEX0;

    //-------- flip-flops ------------------------------------
    reg [23:0] clks;
    reg        state;
    reg  [3:0] countdown;

//==== combinational part ===================================

    // clock signal
    clksrc clksrc1 (clk, clk_16);
    assign next_clks = (state==S_PAUSE)? clks: clks+24'd1;
```

Output should be register.
(Critical path issue)

PLL
(input: clk, 50MHz)
(output: clk_16, 16MHz)

exp1_traffic.v (3/5)

```
// finite state machine (state)
always@(*) begin
    case(state)
        S_NORMAL: begin
            if(pause==1) next_state = S_PAUSE;
            else next_state = S_NORMAL;
        end
        S_PAUSE: begin
            if(pause==1) next_state = S_PAUSE;
            else next_state = S_NORMAL;
        end
    endcase
end

// countdown
always@(*) begin
    if(clks[23]==1'b1 && next_clks[23]==1'b0)
        next_countdown = (countdown==0)? C_PERIOD: countdown-4'd1;
    else
        next_countdown = countdown;
end
```

Cover every possible branch of every if or case to avoid latches.

Be careful!!

1Hz

Cover every possible branch.

exp1_traffic.v (4/5)

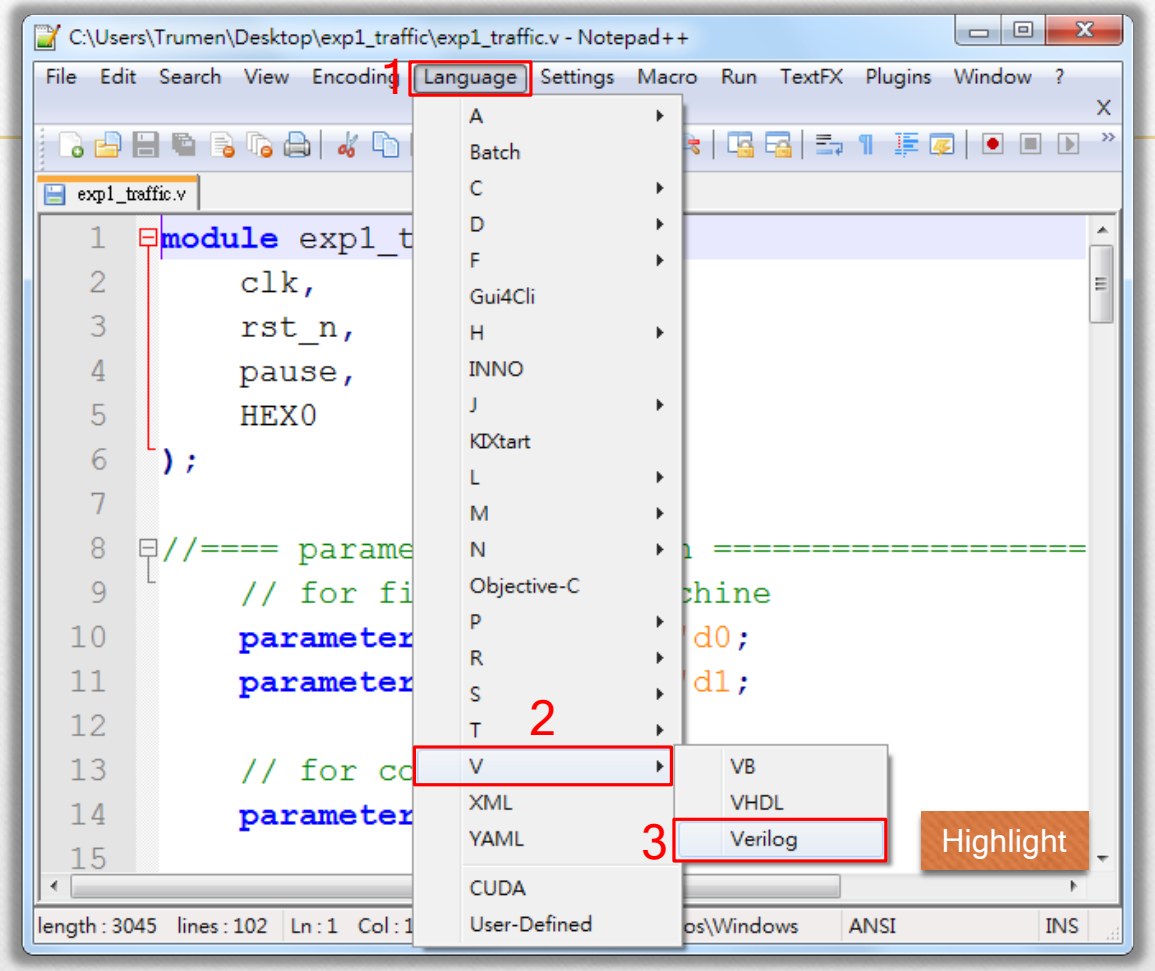
```
// 7-segment Displays
always@(*) begin
    case(countdown)
        7'd0: next_HEX0 = 7'b1000000;
        7'd1: next_HEX0 = 7'b1111001;
        7'd2: next_HEX0 = 7'b0100100;
        7'd3: next_HEX0 = 7'b0110000;
        7'd4: next_HEX0 = 7'b0011001;
        7'd5: next_HEX0 = 7'b0010010;
        7'd6: next_HEX0 = 7'b0000010;
        7'd7: next_HEX0 = 7'b1111000;
        7'd8: next_HEX0 = 7'b0000000;
        7'd9: next_HEX0 = 7'b0010000;
        default: next_HEX0 = 7'b1111111;
    endcase
end
```

Cover every possible branch.

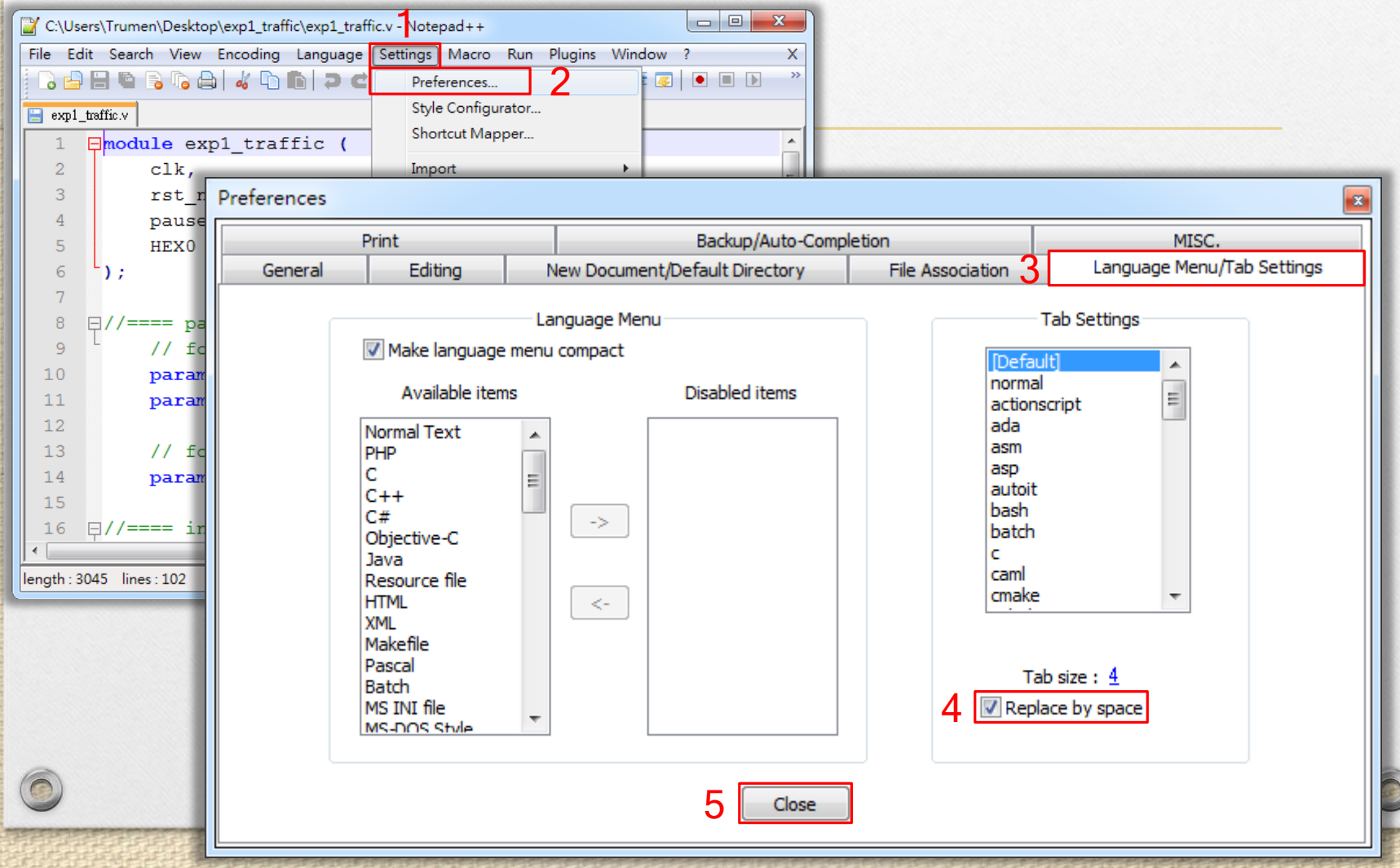
exp1_traffic.v (5/5)

```
//==== sequential part ====================================
    always@( posedge clk_16 or negedge rst_n ) begin
        if( rst_n==0 ) begin
            clks      <= 24'd0;
            state     <= S_NORMAL;
            countdown <= C_PERIOD;
            HEX0      <= 7'h7f;
        end
        else begin
            clks      <= next_clks;
            state     <= next_state;
            countdown <= next_countdown;
            HEX0      <= next_HEX0;
        end
    end
endmodule
```

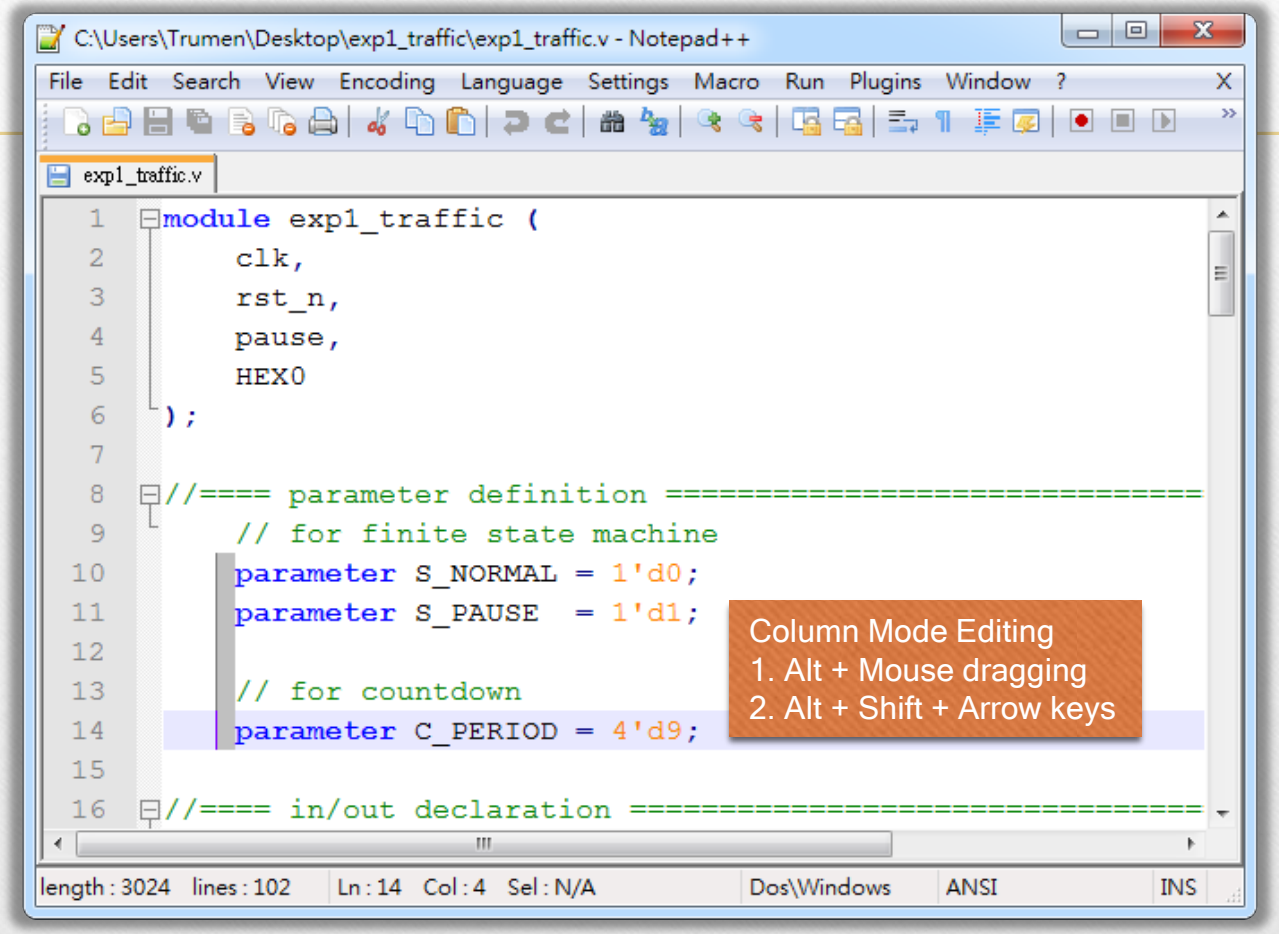
Notepad++ (1/5)


C:\Users\Trumen\Desktop\exp1_traffic\exp1_traffic.v - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
exp1_traffic.v
1
2
3
Highlight
A
Batch
C
D
F
Gui4Cli
H
INNO
J
KIXtart
L
M
N
Objective-C
P
R
S
T
V
XML
YAML
CUDA
User-Defined
VB
VHDL
Verilog
length : 3045 lines : 102 Ln : 1 Col : 1
ANSI
INS

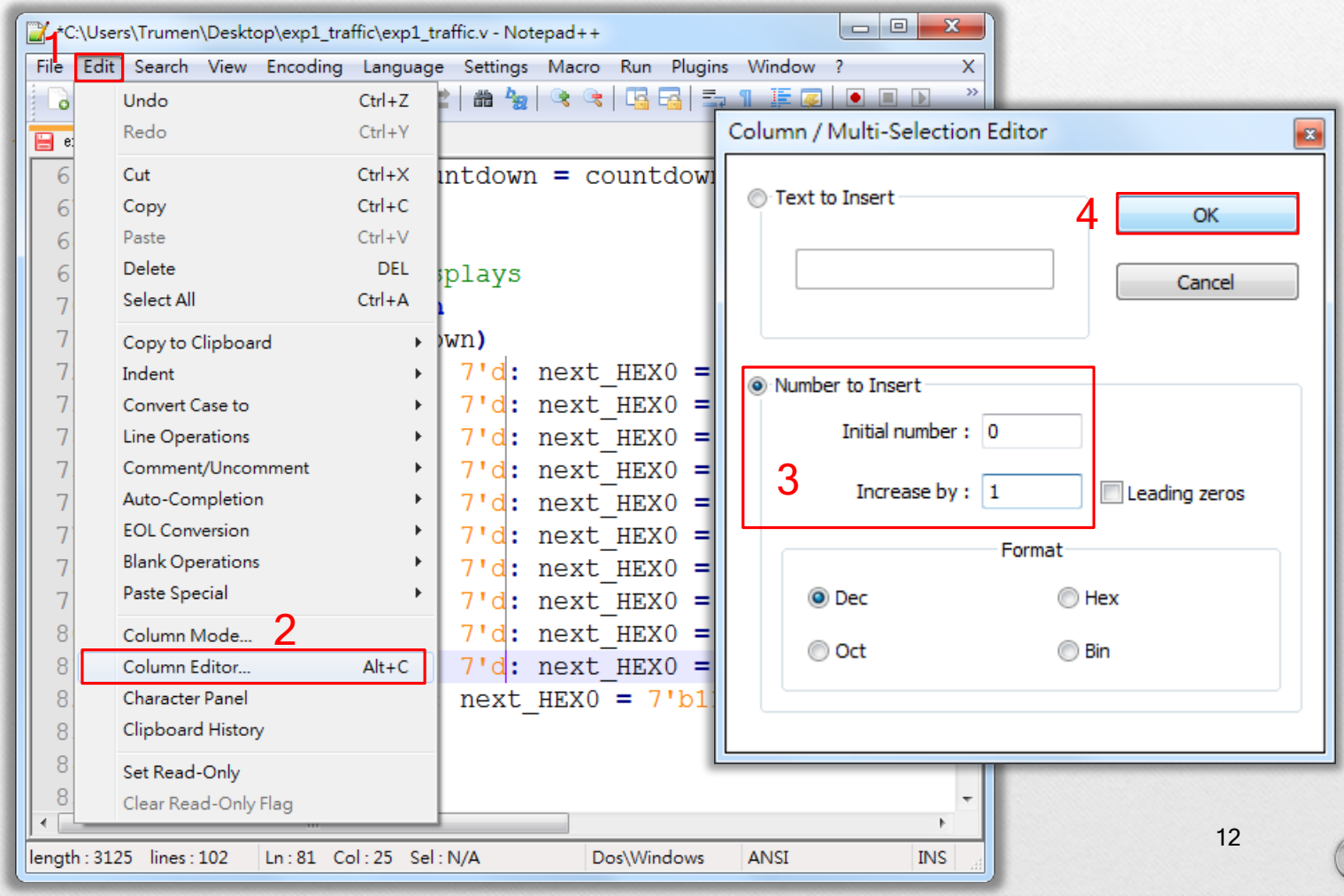
Notepad++ (2/5)


C:\Users\Trumen\Desktop\exp1_traffic\exp1_traffic.v - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
1
2
Preferences...
Style Configurator...
Shortcut Mapper...
Import
exp1_traffic.v
module exp1_traffic (
clk,
rst_n
pause
HEX0
);
length : 3045 lines : 102
Preferences
Print
Backup/Auto-Completion
MISC.
General
Editing
New Document/Default Directory
File Association
3
Language Menu/Tab Settings
Language Menu
Make language menu compact
Available items
Disabled items
Normal Text
PHP
C
C++
C#
Objective-C
Java
Resource file
HTML
XML
Makefile
Pascal
Batch
MS INI file
->
<-
Tab Settings
[Default]
normal
actionscript
ada
asm
asp
autoit
bash
batch
c
caml
cmake
Tab size : 4
4
Replace by space
5
Close

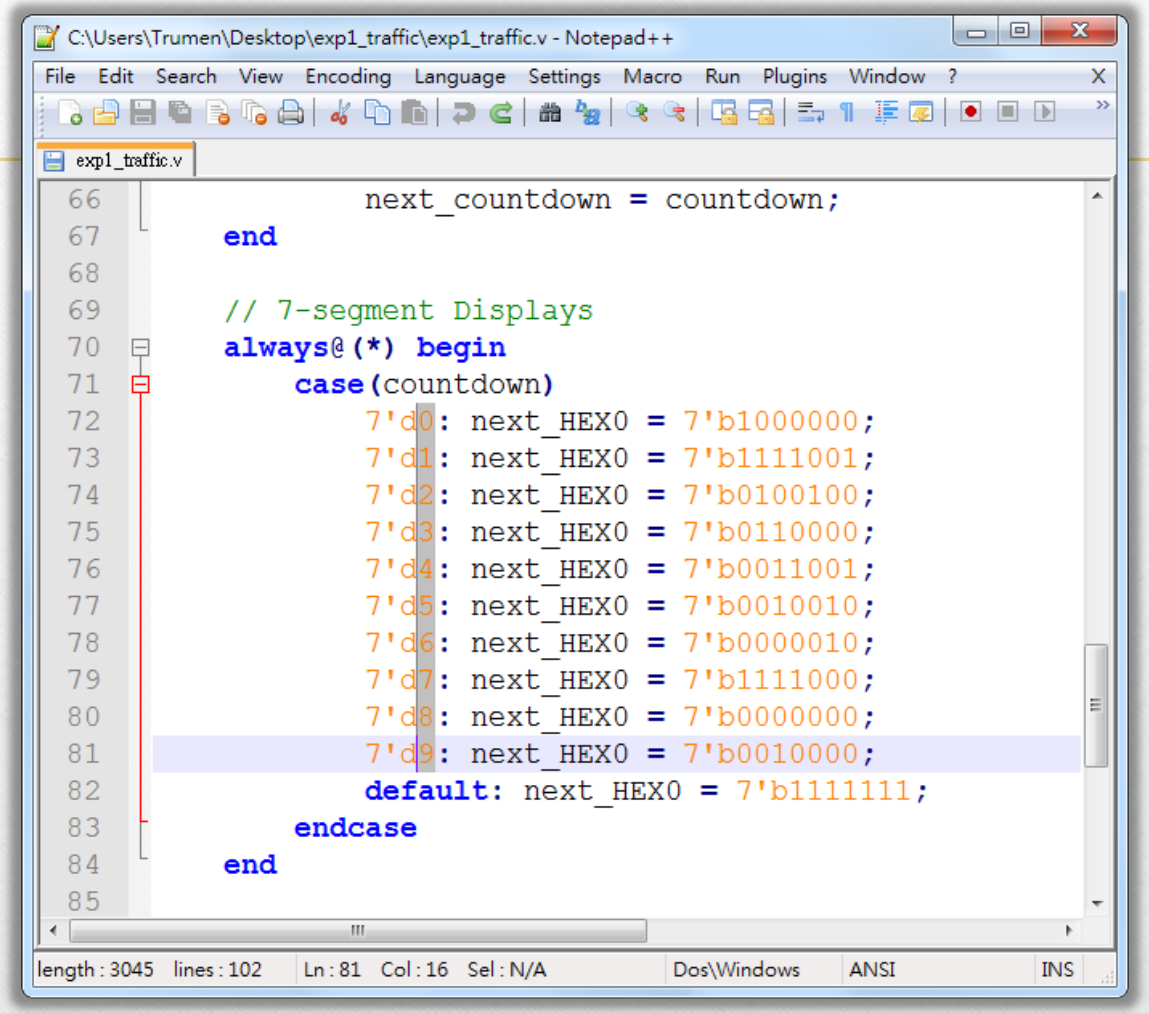
Notepad++ (3/5)


C:\Users\Trumen\Desktop\exp1_traffic\exp1_traffic.v - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
exp1_traffic.v
1 module exp1_traffic (
2 clk,
3 rst_n,
4 pause,
5 HEX0
6);
7
8 //==== parameter definition ==========
9 // for finite state machine
10 parameter S_NORMAL = 1'd0;
11 parameter S_PAUSE = 1'd1;
12
13 // for countdown
14 parameter C_PERIOD = 4'd9;
15
16 //==== in/out declaration ==========
Column Mode Editing
1. Alt + Mouse dragging
2. Alt + Shift + Arrow keys
length : 3024 lines : 102 Ln : 14 Col : 4 Sel : N/A Dos\Windows ANSI INS

Notepad++ (4/5)


*C:\Users\Trumen\Desktop\exp1_traffic\exp1_traffic.v - Notepad++
File
Edit
Search
View
Encoding
Language
Settings
Macro
Run
Plugins
Window
?
1
Undo Ctrl+Z
Redo Ctrl+Y
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete DEL
Select All Ctrl+A
Copy to Clipboard
Indent
Convert Case to
Line Operations
Comment/Uncomment
Auto-Completion
EOL Conversion
Blank Operations
Paste Special
Column Mode...
2
Column Editor... Alt+C
Character Panel
Clipboard History
Set Read-Only
Clear Read-Only Flag
length : 3125 lines : 102 Ln : 81 Col : 25 Sel : N/A Dos\Windows ANSI INS
Column / Multi-Selection Editor
Text to Insert
4
OK
Cancel
Number to Insert
3
Initial number : 0
Increase by : 1
Leading zeros
Format
Dec
Hex
Oct
Bin

Notepad++ (5/5)


C:\Users\Trumen\Desktop\exp1_traffic\exp1_traffic.v - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
exp1_traffic.v
66 next_countdown = countdown;
67 end
68
69 // 7-segment Displays
70 always@(*) begin
71 case(countdown)
72 7'd0: next_HEX0 = 7'b1000000;
73 7'd1: next_HEX0 = 7'b1111001;
74 7'd2: next_HEX0 = 7'b0100100;
75 7'd3: next_HEX0 = 7'b0110000;
76 7'd4: next_HEX0 = 7'b0011001;
77 7'd5: next_HEX0 = 7'b0010010;
78 7'd6: next_HEX0 = 7'b0000010;
79 7'd7: next_HEX0 = 7'b1111000;
80 7'd8: next_HEX0 = 7'b0000000;
81 7'd9: next_HEX0 = 7'b0010000;
82 default: next_HEX0 = 7'b1111111;
83 endcase
84 end
85
length : 3045 lines : 102 Ln : 81 Col : 16 Sel : N/A Dos\Windows ANSI INS

Assign The Device

Introduction to FPGA (1/3)

- A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a designer after manufacturing.
  - An electronic device is said to be field-programmable if it can be modified "in the field".

Introduction to FPGA (2/3)

- FPGAs contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together".
- FPGAs can be used to implement any logical function that an ASIC could perform.

Introduction to FPGA (3/3)

- Xilinx and Altera are the current FPGA market leaders and long-time industry rivals.
  - Both Xilinx and Altera provide free Windows and Linux design software (ISE and Quartus)

Altera's Main FPGA Products

- Stratix series FPGAs are the largest, highest bandwidth devices, with up to 1.1 million logic elements.
- Cyclone series FPGAs and are the company's lowest cost, lowest power FPGAs.
- Arria series FPGAs are between the two device families above.

Altera® Development Kits

http://www.altera.com/products/devkits/kit-dev_platforms.jsp

- Development kits include software, reference designs, cables, and programming hardware (development board).

Installed The USB-Blaster driver

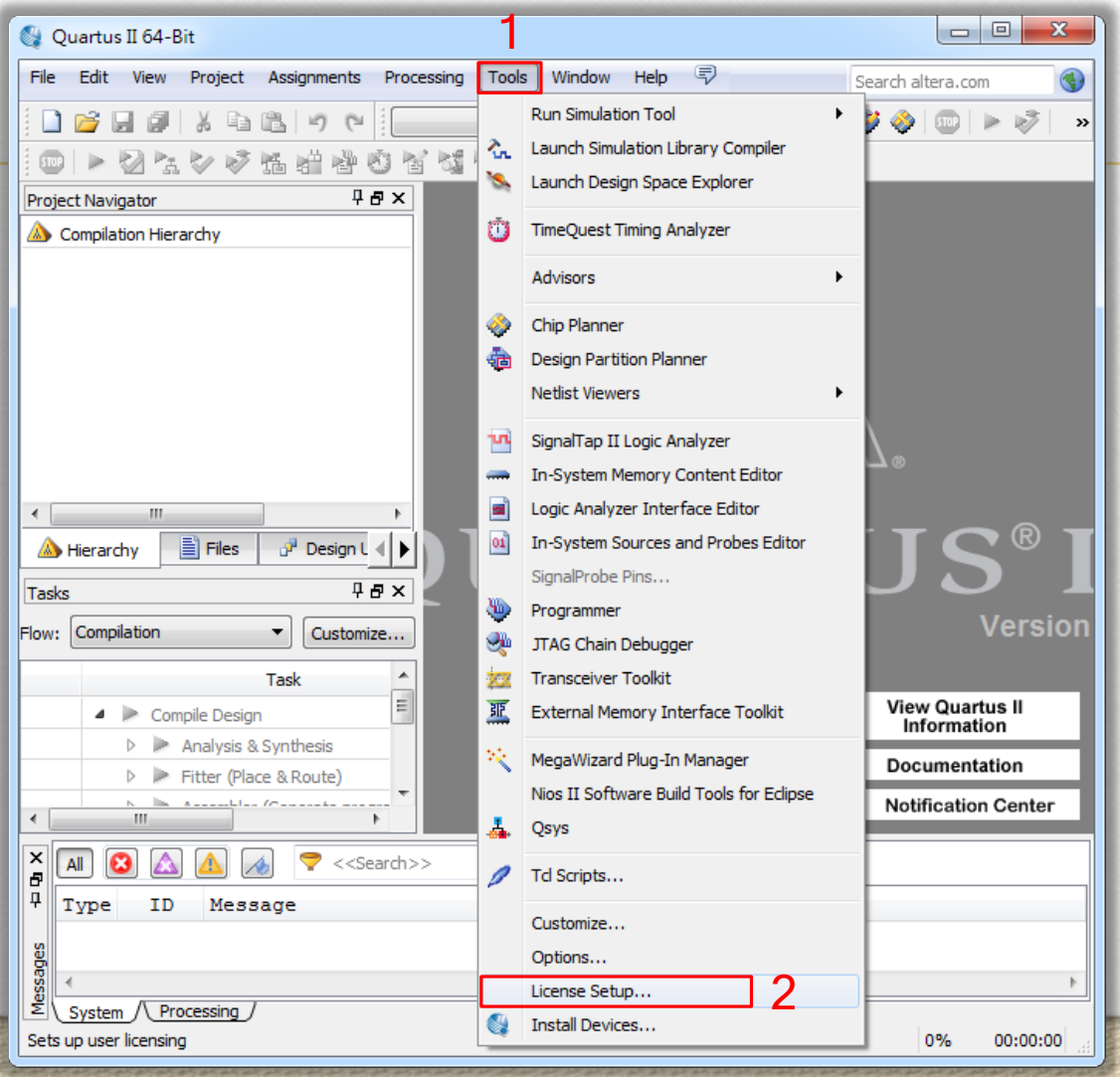
(1/2)

- Plug in the 12-volt adapter to provide power to the board.
- Use the USB cable to connect the leftmost USB connector (the one closest to the power switch) on the DE2-115 board to a USB port on a computer that runs the Quartus II software.
- Turn on the power switch on the DE2-115 board.

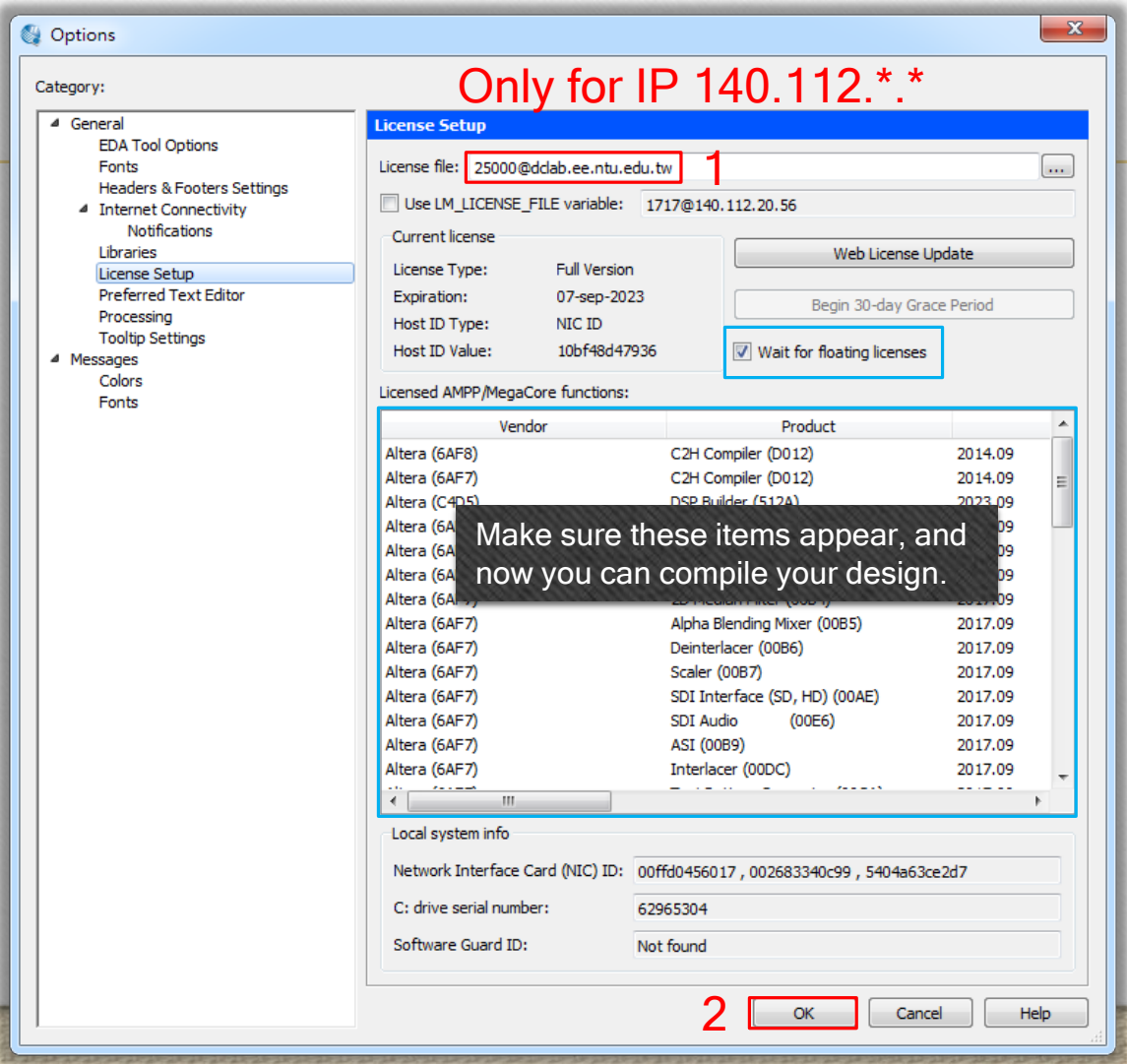
Installed The USB-Blaster driver
(2/2)

- The computer will recognize the new hardware connected to its USB port.
  - But it will be unable to proceed if it does not have the required driver already installed.
  - The DE2-115 board is programmed by using Altera USB-Blaster mechanism. If the USB-Blaster driver is not already installed, the New Hardware Wizard will appear.
    - Next →Next →… →OK!

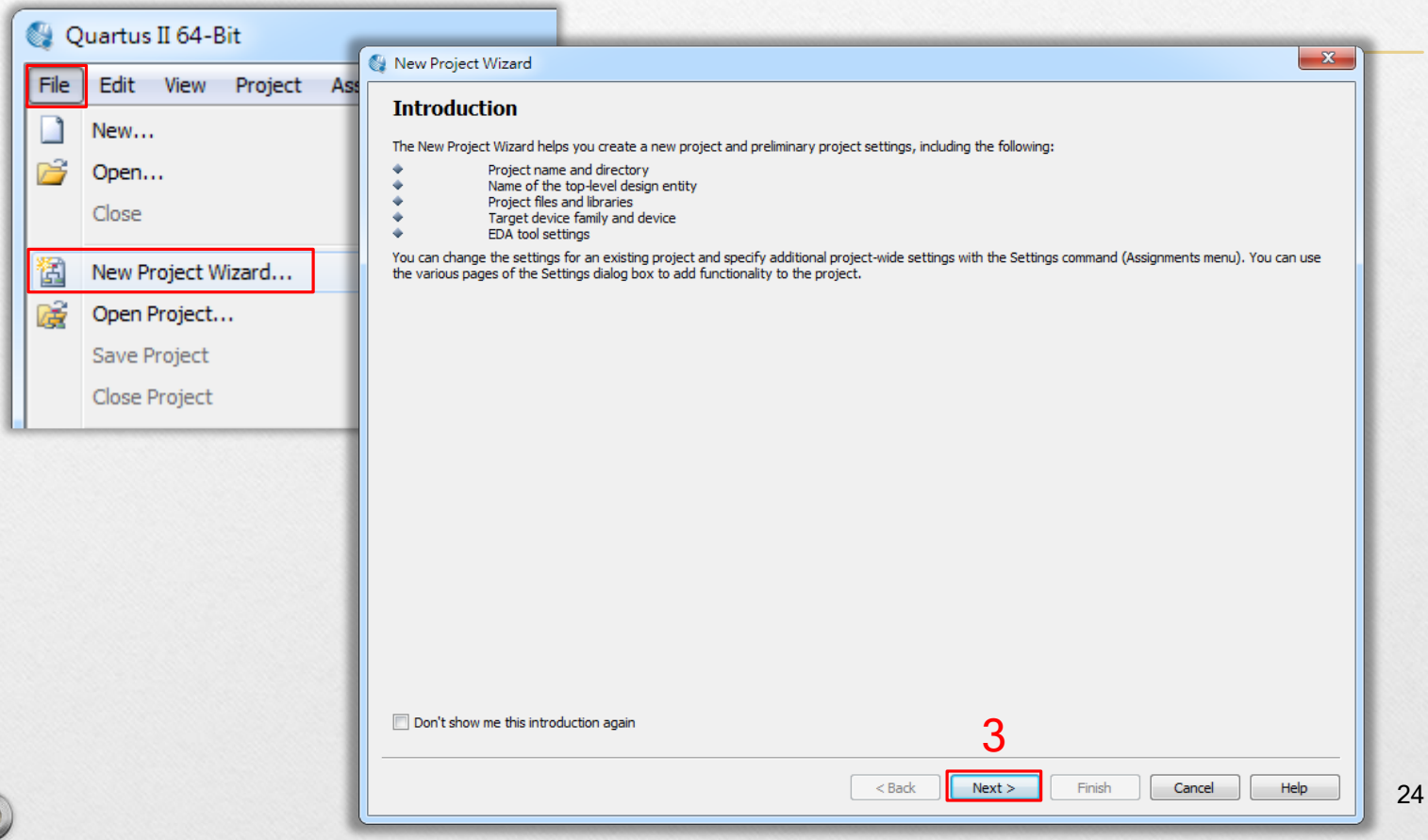
Setup Licensing (1/2)

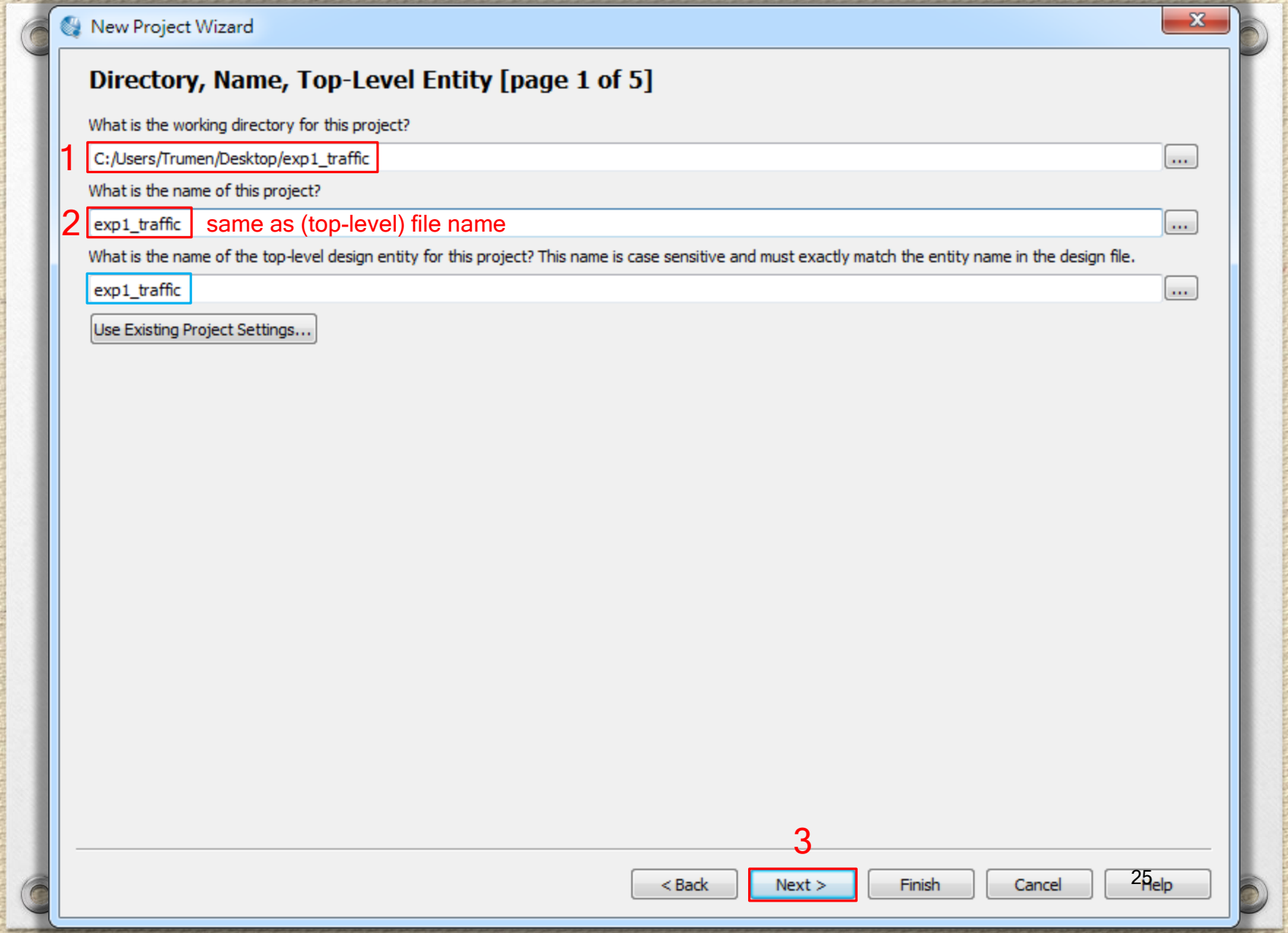


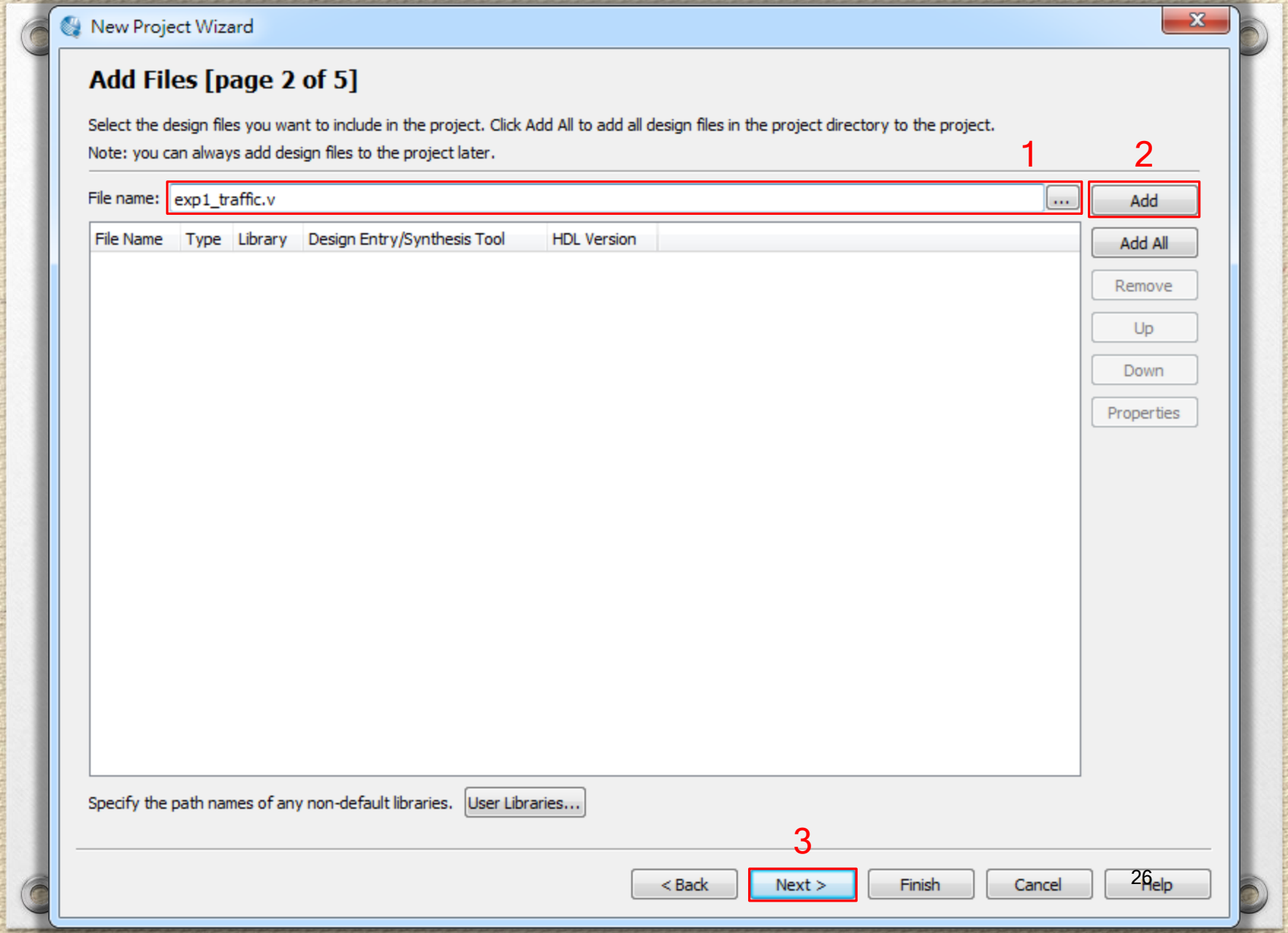
Setup Licensing (2/2)

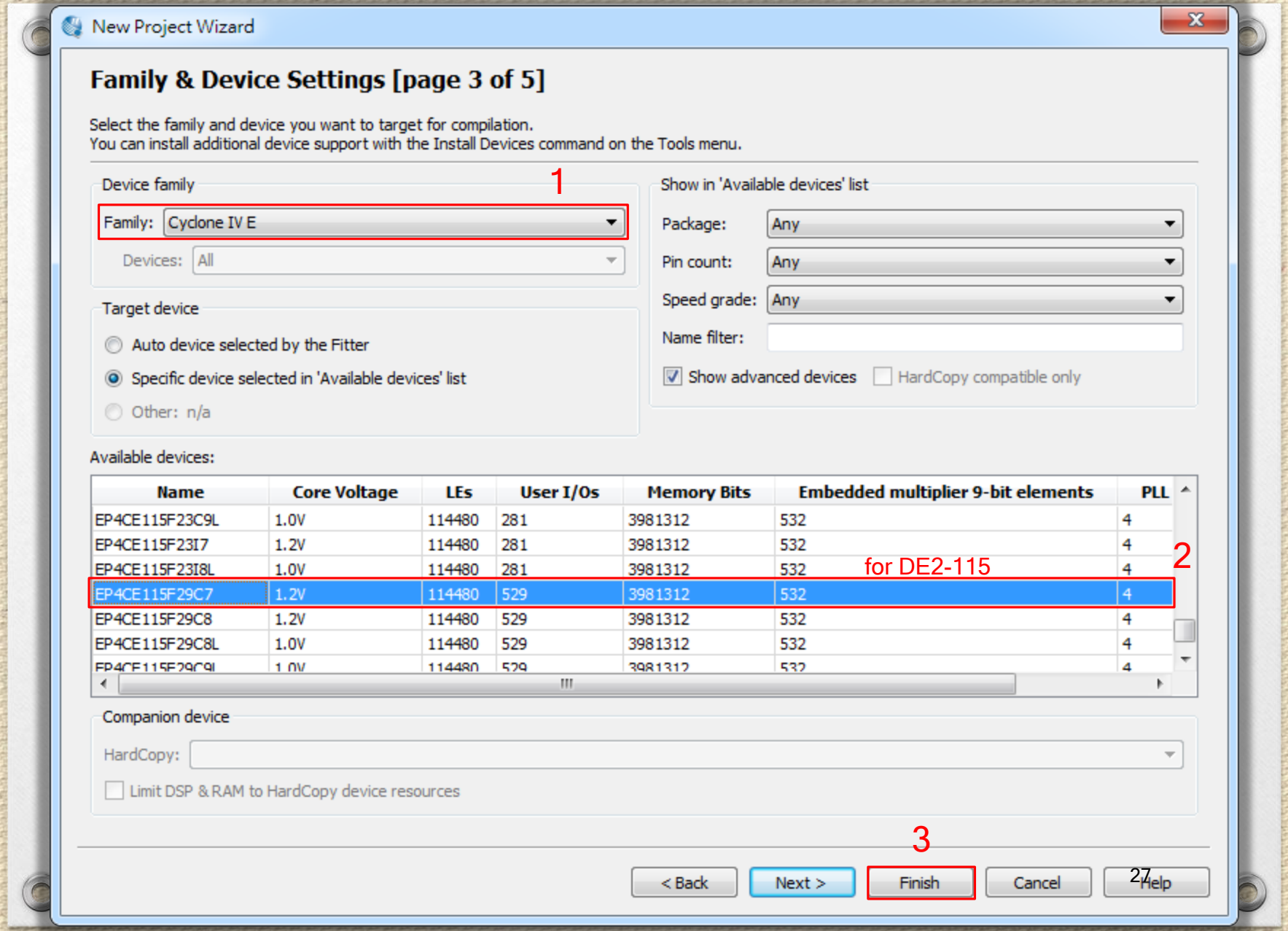


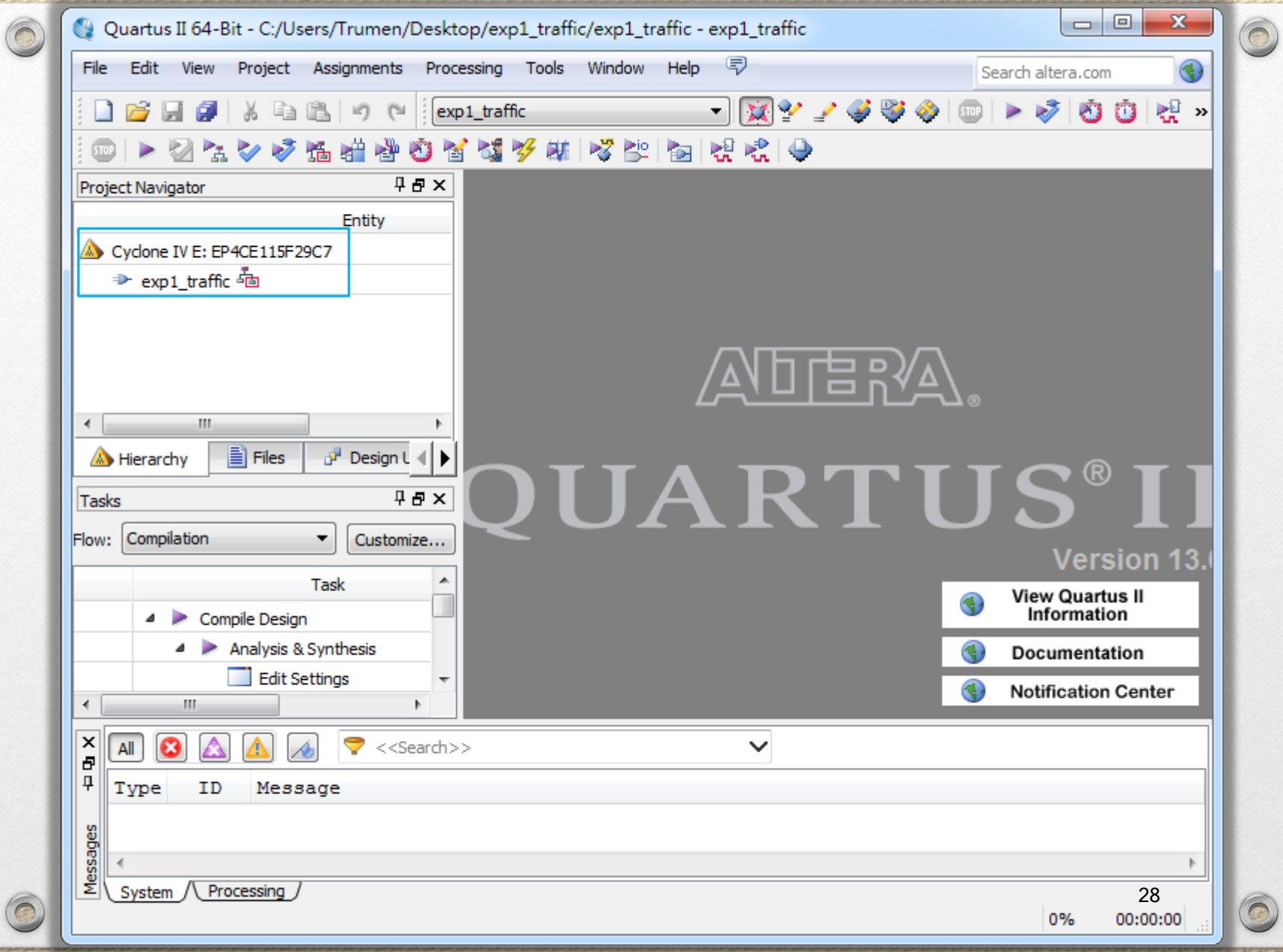
Create a New Project




New Project Wizard
Directory, Name, Top-Level Entity [page 1 of 5]
What is the working directory for this project?
1
C:/Users/Trumen/Desktop/exp1_traffic
...
What is the name of this project?
2
exp1_traffic
same as (top-level) file name
...
What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.
exp1_traffic
...
Use Existing Project Settings...
3
< Back
Next >
Finish
Cancel
Help


New Project Wizard
Add Files [page 2 of 5]
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.
1
2
File name:
exp1_traffic.v
...
Add
File Name
Type
Library
Design Entry/Synthesis Tool
HDL Version
Add All
Remove
Up
Down
Properties
Specify the path names of any non-default libraries.
User Libraries...
3
< Back
Next >
Finish
Cancel
Help

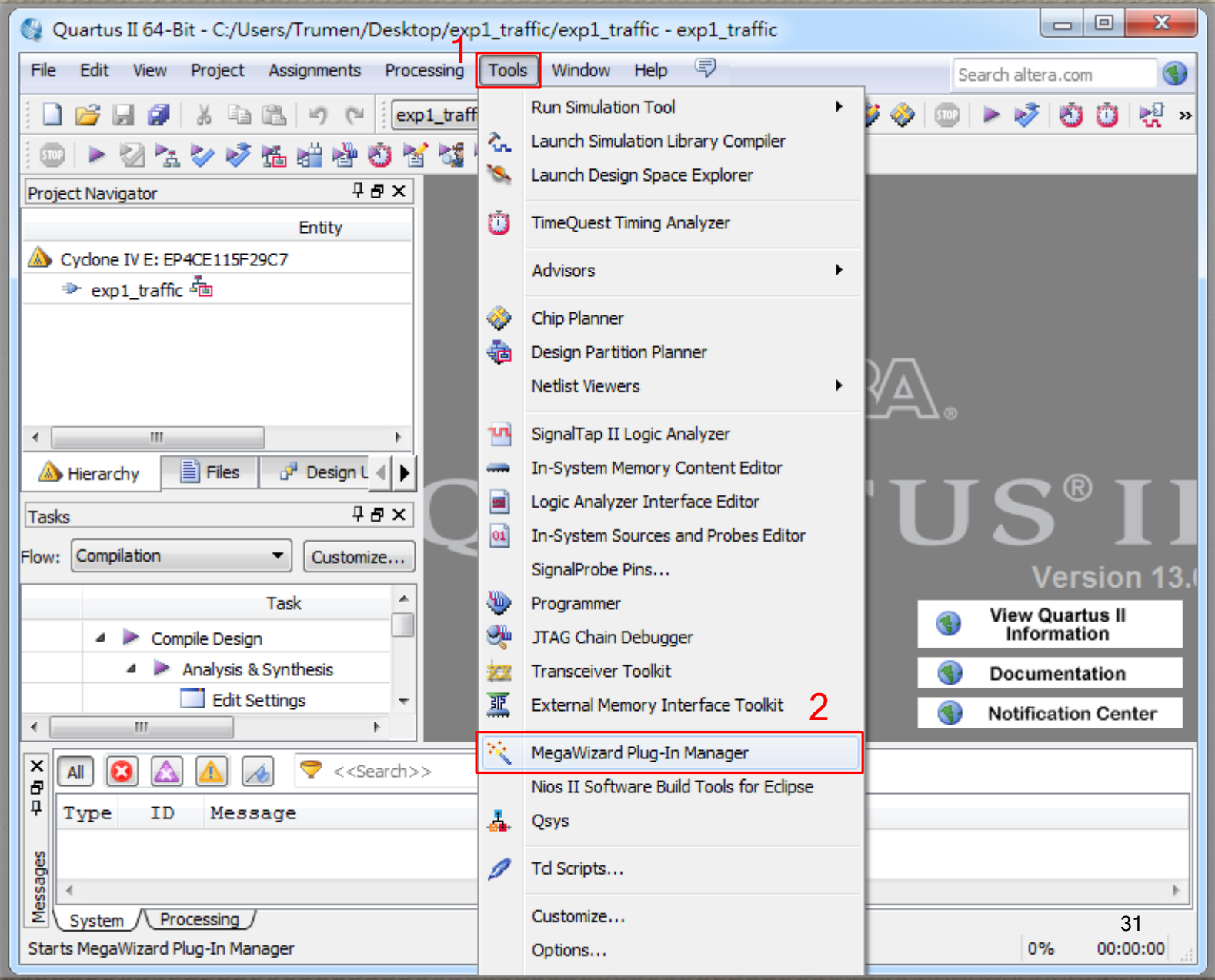

New Project Wizard
Family & Device Settings [page 3 of 5]
Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.
1
Device family
Family: Cyclone IV E
Devices: All
Target device
Auto device selected by the Fitter
Specific device selected in 'Available devices' list
Other: n/a
Show in 'Available devices' list
Package: Any
Pin count: Any
Speed grade: Any
Name filter:
Show advanced devices
HardCopy compatible only
Available devices:
Name | Core Voltage | LEs | User I/Os | Memory Bits | Embedded multiplier 9-bit elements | PLL
EP4CE115F23C9L | 1.0V | 114480 | 281 | 3981312 | 532 | 4
EP4CE115F23I7 | 1.2V | 114480 | 281 | 3981312 | 532 | 4
EP4CE115F23I8L | 1.0V | 114480 | 281 | 3981312 | 532 | 4
EP4CE115F29C7 | 1.2V | 114480 | 529 | 3981312 | 532 | 4
EP4CE115F29C8 | 1.2V | 114480 | 529 | 3981312 | 532 | 4
EP4CE115F29C8L | 1.0V | 114480 | 529 | 3981312 | 532 | 4
for DE2-115
2
Companion device
HardCopy:
Limit DSP & RAM to HardCopy device resources
3
< Back
Next >
Finish
Cancel
Help

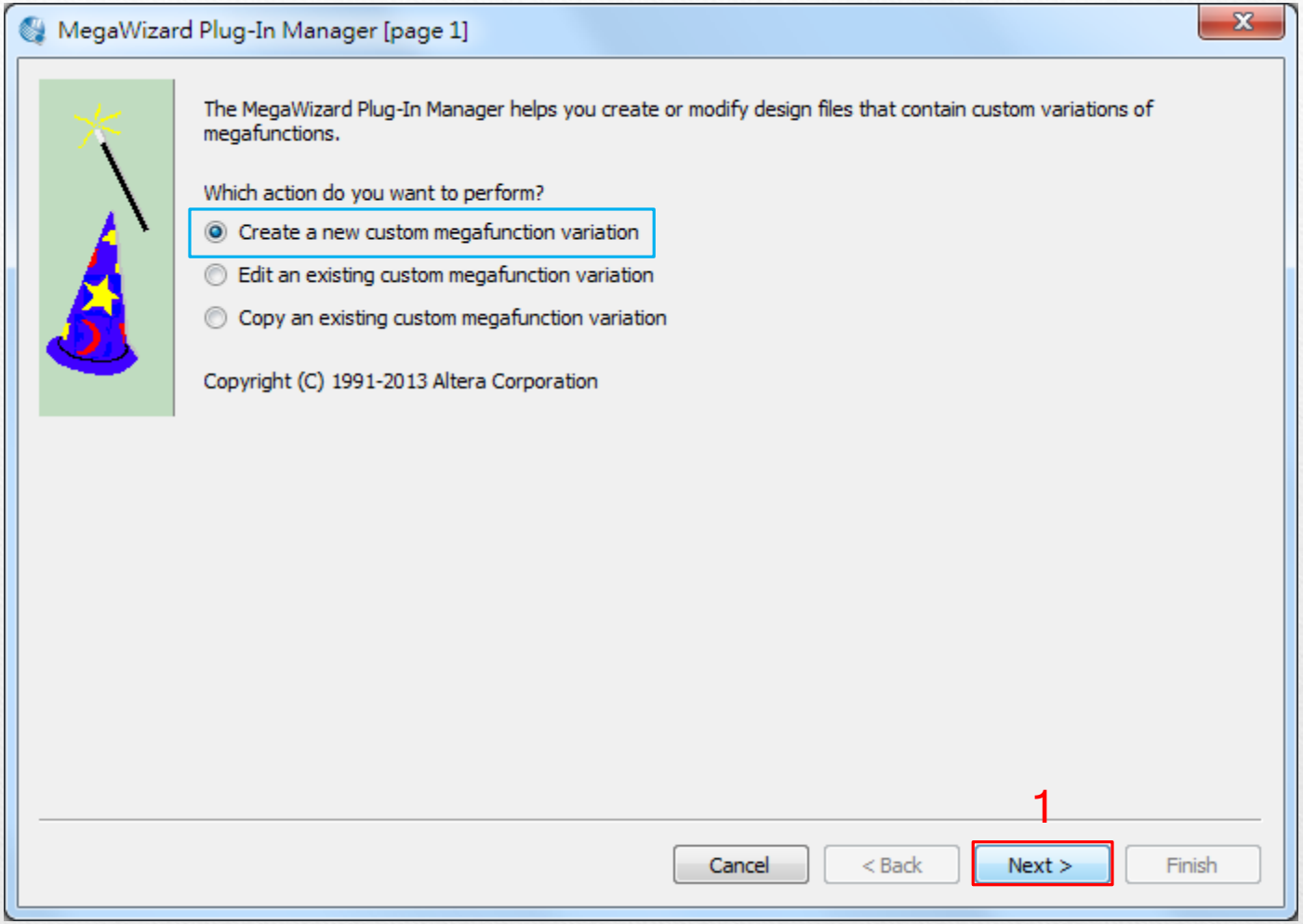

Quartus II 64-Bit - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - exp1_traffic
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
exp1_traffic
Project Navigator
Entity
Cyclone IV E: EP4CE115F29C7
exp1_traffic
Hierarchy
Files
Design U
Tasks
Flow: Compilation
Customize...
Task
Compile Design
Analysis & Synthesis
Edit Settings
ALTERA
QUARTUS® II
Version 13.
View Quartus II Information
Documentation
Notification Center
All
<<Search>>
Type ID Message
Messages
System
Processing
0%
00:00:00

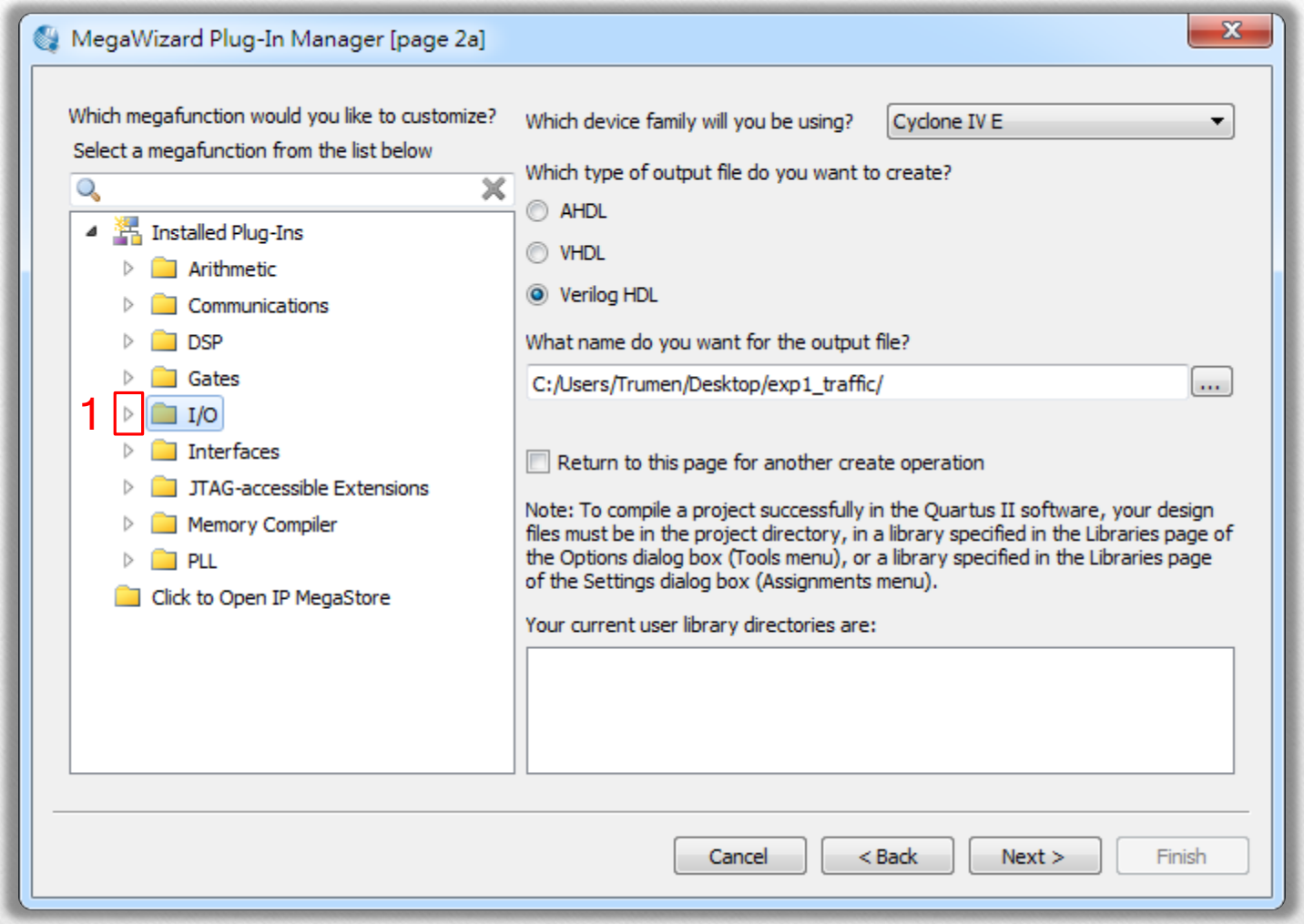
Add a PLL Megafunction

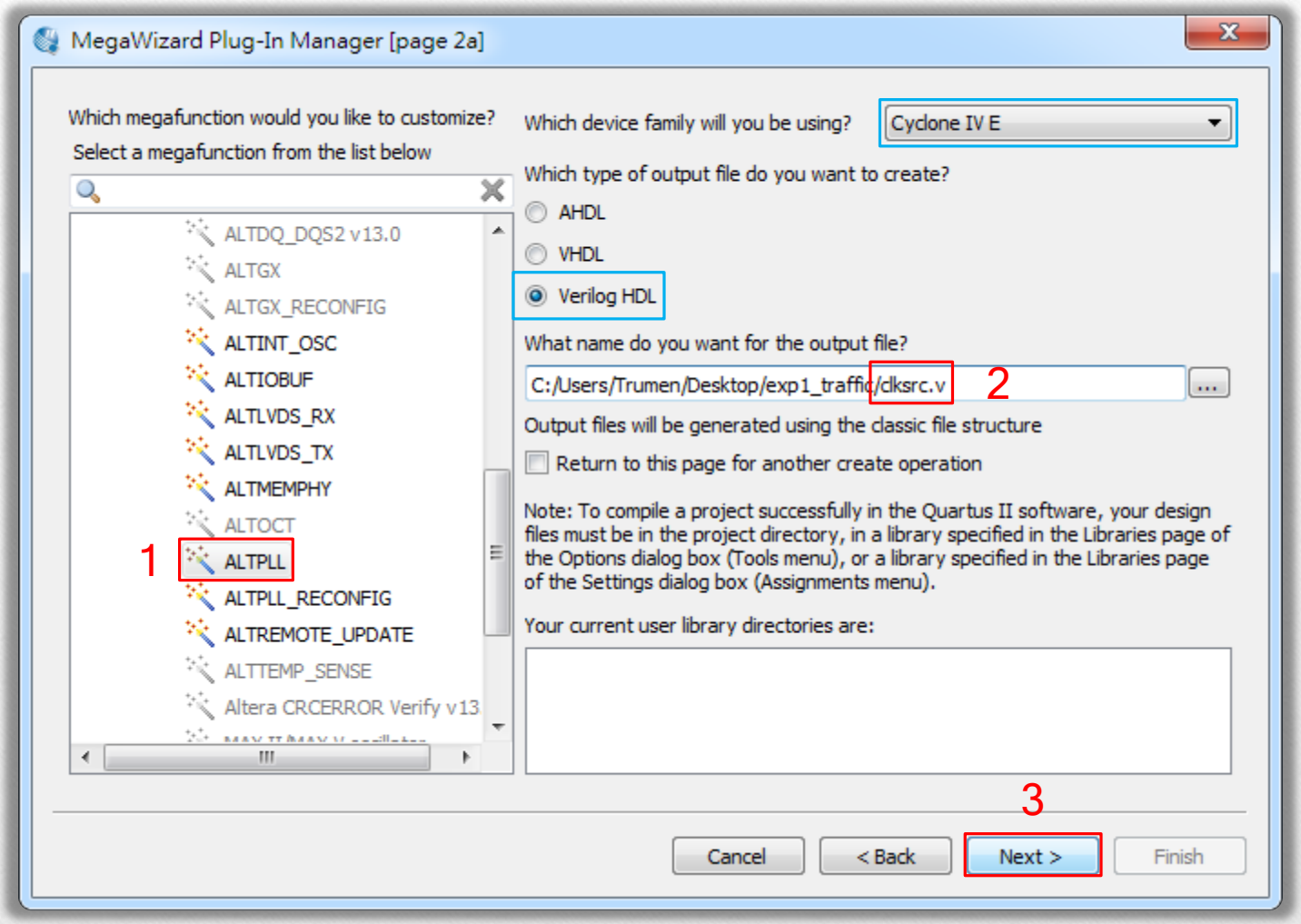
Using Quartus Add a PLL Megafunction

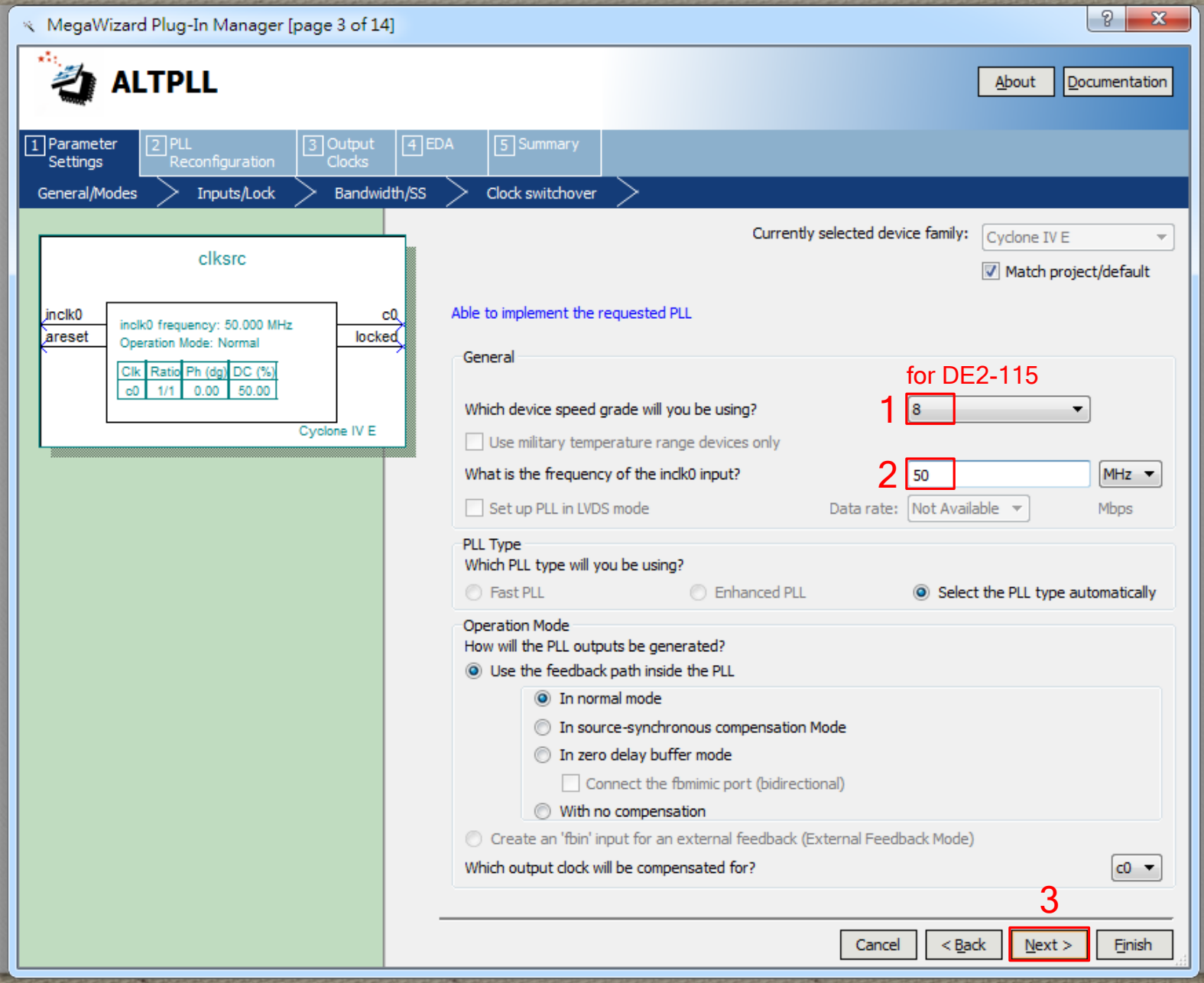
- A PLL uses the on-board oscillator (50 MHz for DE2-115 Board) to create a constant clock frequency as the input to the counter.
- To create the clock source, you will add a pre-built library of parameterized modules (LPM) megafunction named ALTPLL.

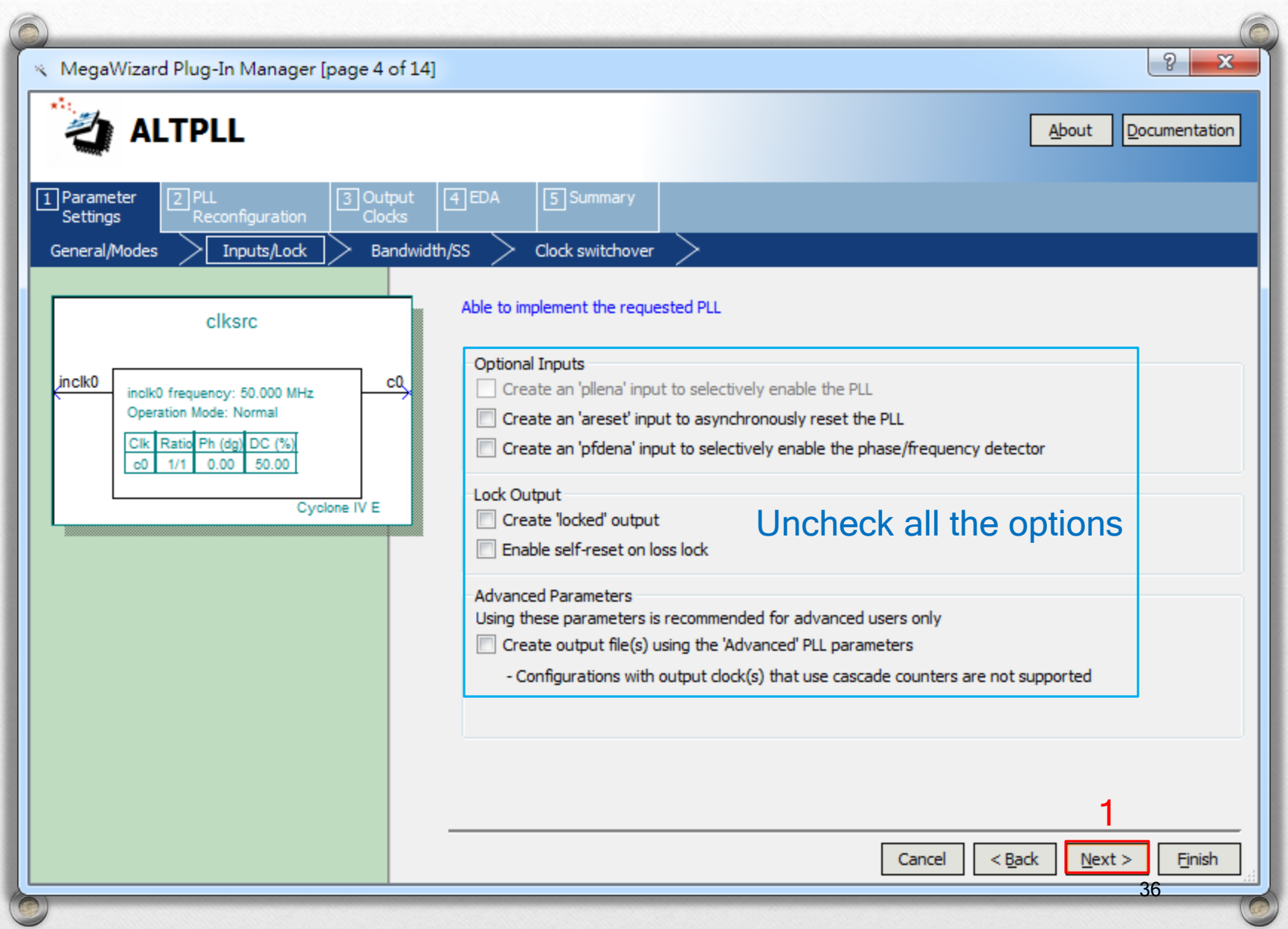

Quartus II 64-Bit - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - exp1_traffic
File
Edit
View
Project
Assignments
Processing
Tools
Window
Help
Search altera.com
1
Run Simulation Tool
Launch Simulation Library Compiler
Launch Design Space Explorer
TimeQuest Timing Analyzer
Advisors
Chip Planner
Design Partition Planner
Netlist Viewers
SignalTap II Logic Analyzer
In-System Memory Content Editor
Logic Analyzer Interface Editor
In-System Sources and Probes Editor
SignalProbe Pins...
Programmer
JTAG Chain Debugger
Transceiver Toolkit
External Memory Interface Toolkit
2
MegaWizard Plug-In Manager
Nios II Software Build Tools for Eclipse
Qsys
Tcl Scripts...
Customize...
Options...
Project Navigator
Entity
Cyclone IV E: EP4CE115F29C7
exp1_traffic
Hierarchy
Files
Design U
Tasks
Flow:
Compilation
Customize...
Task
Compile Design
Analysis & Synthesis
Edit Settings
Version 13.
View Quartus II Information
Documentation
Notification Center
All
<<Search>>
Type
ID
Message
Messages
System
Processing
Starts MegaWizard Plug-In Manager
31
0%
00:00:00

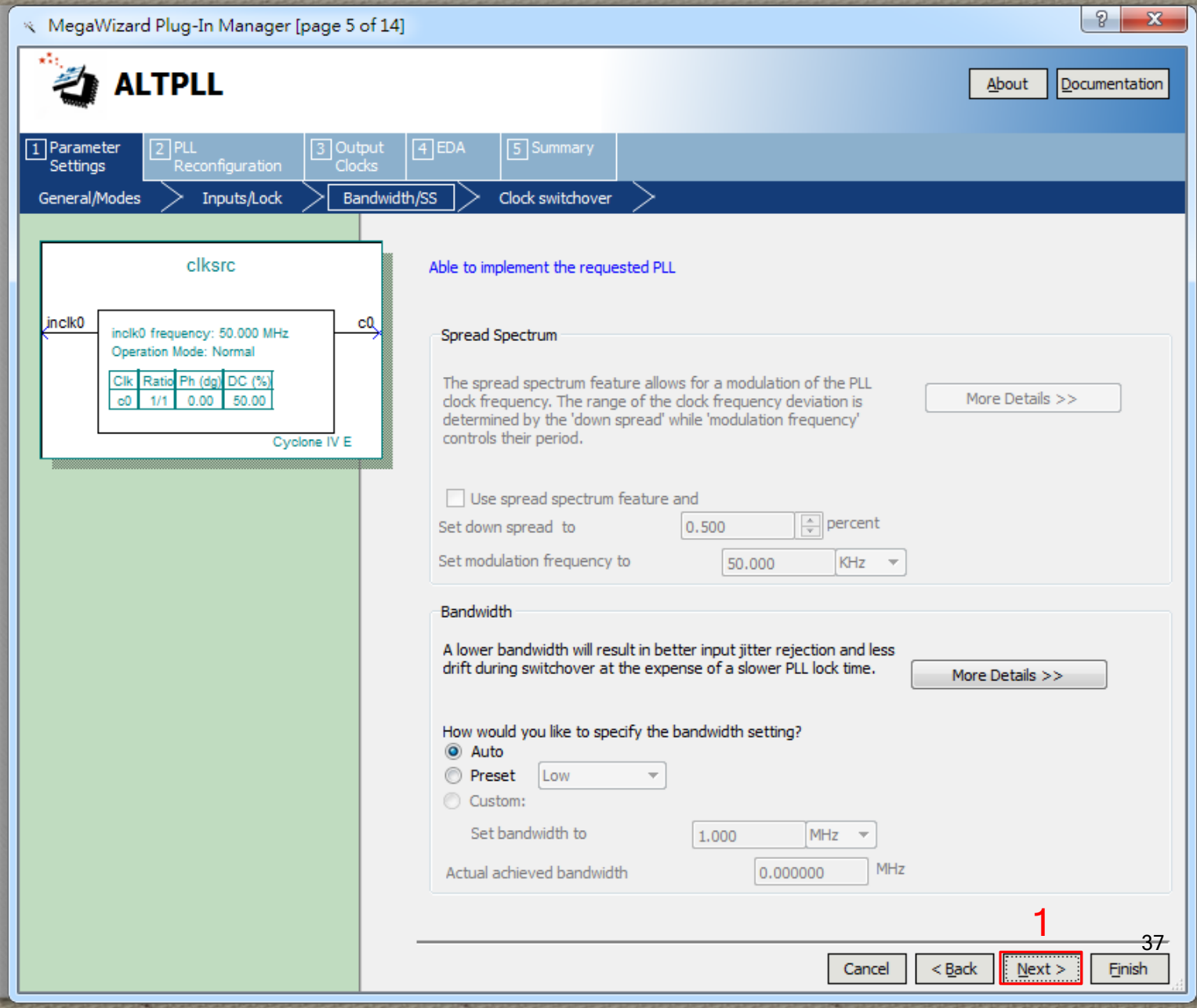

MegaWizard Plug-In Manager [page 1]
The MegaWizard Plug-In Manager helps you create or modify design files that contain custom variations of megafunctions.
Which action do you want to perform?
Create a new custom megafunction variation
Edit an existing custom megafunction variation
Copy an existing custom megafunction variation
Copyright (C) 1991-2013 Altera Corporation
1
Cancel
< Back
Next >
Finish

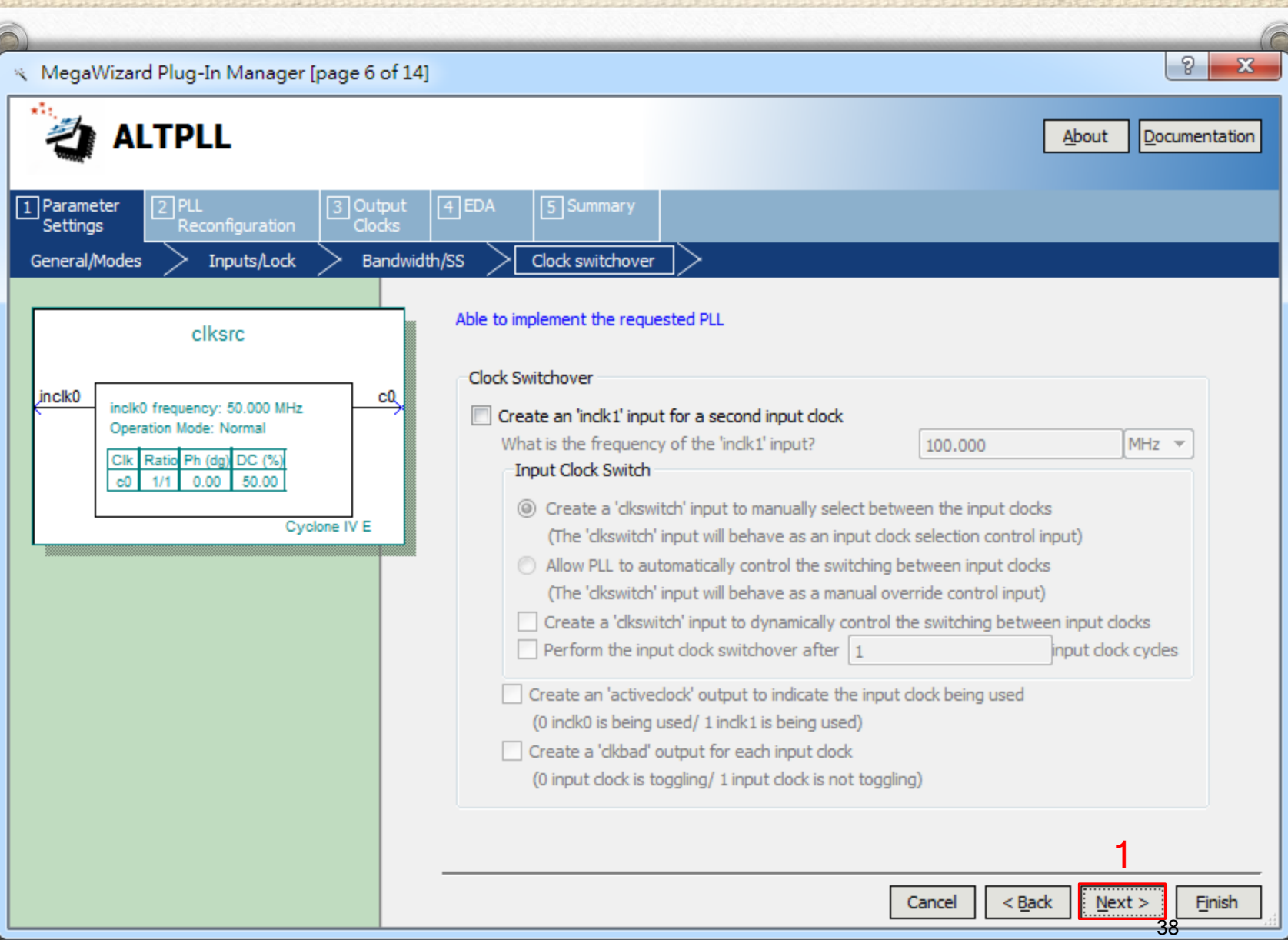

MegaWizard Plug-In Manager [page 2a]
Which megafunction would you like to customize?
Select a megafunction from the list below
Installed Plug-Ins
Arithmetic
Communications
DSP
Gates
1
I/O
Interfaces
JTAG-accessible Extensions
Memory Compiler
PLL
Click to Open IP MegaStore
Which device family will you be using?
Cyclone IV E
Which type of output file do you want to create?
AHDL
VHDL
Verilog HDL
What name do you want for the output file?
C:/Users/Trumen/Desktop/exp1_traffic/
Return to this page for another create operation
Note: To compile a project successfully in the Quartus II software, your design files must be in the project directory, in a library specified in the Libraries page of the Options dialog box (Tools menu), or a library specified in the Libraries page of the Settings dialog box (Assignments menu).
Your current user library directories are:
Cancel
< Back
Next >
Finish

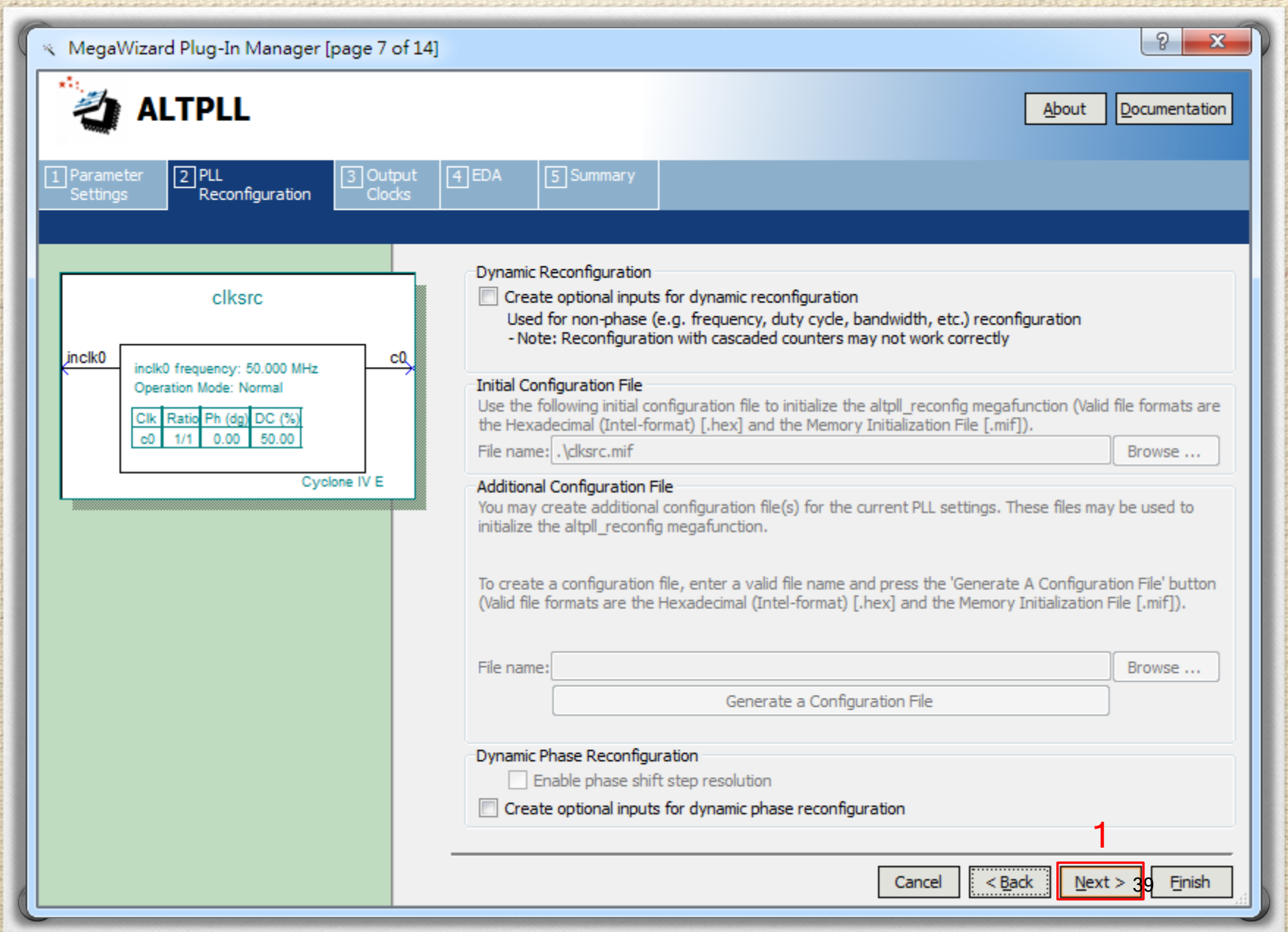

MegaWizard Plug-In Manager [page 2a]
Which megafunction would you like to customize?
Select a megafunction from the list below
ALTDQ_DQS2 v13.0
ALTGX
ALTGX_RECONFIG
ALTINT_OSC
ALTIOBUF
ALTLVDS_RX
ALTLVDS_TX
ALTMEMPHY
ALTOCT
ALTPLL
ALTPLL_RECONFIG
ALTREMOTE_UPDATE
ALTTEMP_SENSE
Altera CRCERROR Verify v13
1
Which device family will you be using?
Cyclone IV E
Which type of output file do you want to create?
AHDL
VHDL
Verilog HDL
What name do you want for the output file?
C:/Users/Trumen/Desktop/exp1_traffic/clksrc.v
2
Output files will be generated using the classic file structure
Return to this page for another create operation
Note: To compile a project successfully in the Quartus II software, your design files must be in the project directory, in a library specified in the Libraries page of the Options dialog box (Tools menu), or a library specified in the Libraries page of the Settings dialog box (Assignments menu).
Your current user library directories are:
3
Cancel
< Back
Next >
Finish

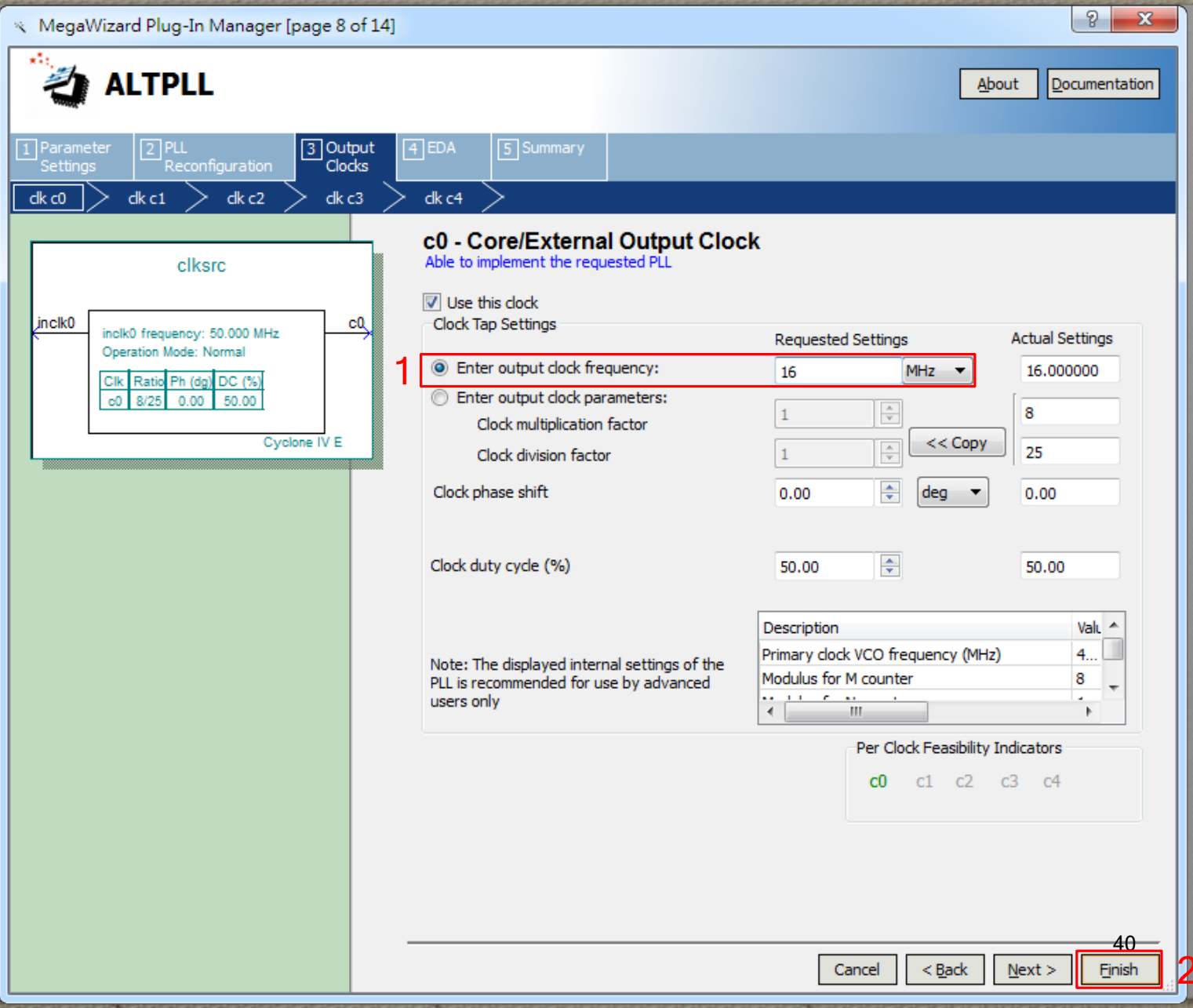

MegaWizard Plug-In Manager [page 3 of 14]
ALTPLL
About
Documentation
1 Parameter Settings
2 PLL Reconfiguration
3 Output Clocks
4 EDA
5 Summary
General/Modes
Inputs/Lock
Bandwidth/SS
Clock switchover
clksrc
inclk0
areset
inclk0 frequency: 50.000 MHz
Operation Mode: Normal
Clk
Ratio
Ph (dg)
DC (%)
c0
1/1
0.00
50.00
c0
locked
Cyclone IV E
Currently selected device family:
Cyclone IV E
Match project/default
Able to implement the requested PLL
General
for DE2-115
Which device speed grade will you be using?
1
8
Use military temperature range devices only
What is the frequency of the inclk0 input?
2
50
MHz
Set up PLL in LVDS mode
Data rate:
Not Available
Mbps
PLL Type
Which PLL type will you be using?
Fast PLL
Enhanced PLL
Select the PLL type automatically
Operation Mode
How will the PLL outputs be generated?
Use the feedback path inside the PLL
In normal mode
In source-synchronous compensation Mode
In zero delay buffer mode
Connect the fbmimic port (bidirectional)
With no compensation
Create an 'fbin' input for an external feedback (External Feedback Mode)
Which output clock will be compensated for?
c0
3
Cancel
< Back
Next >
Finish

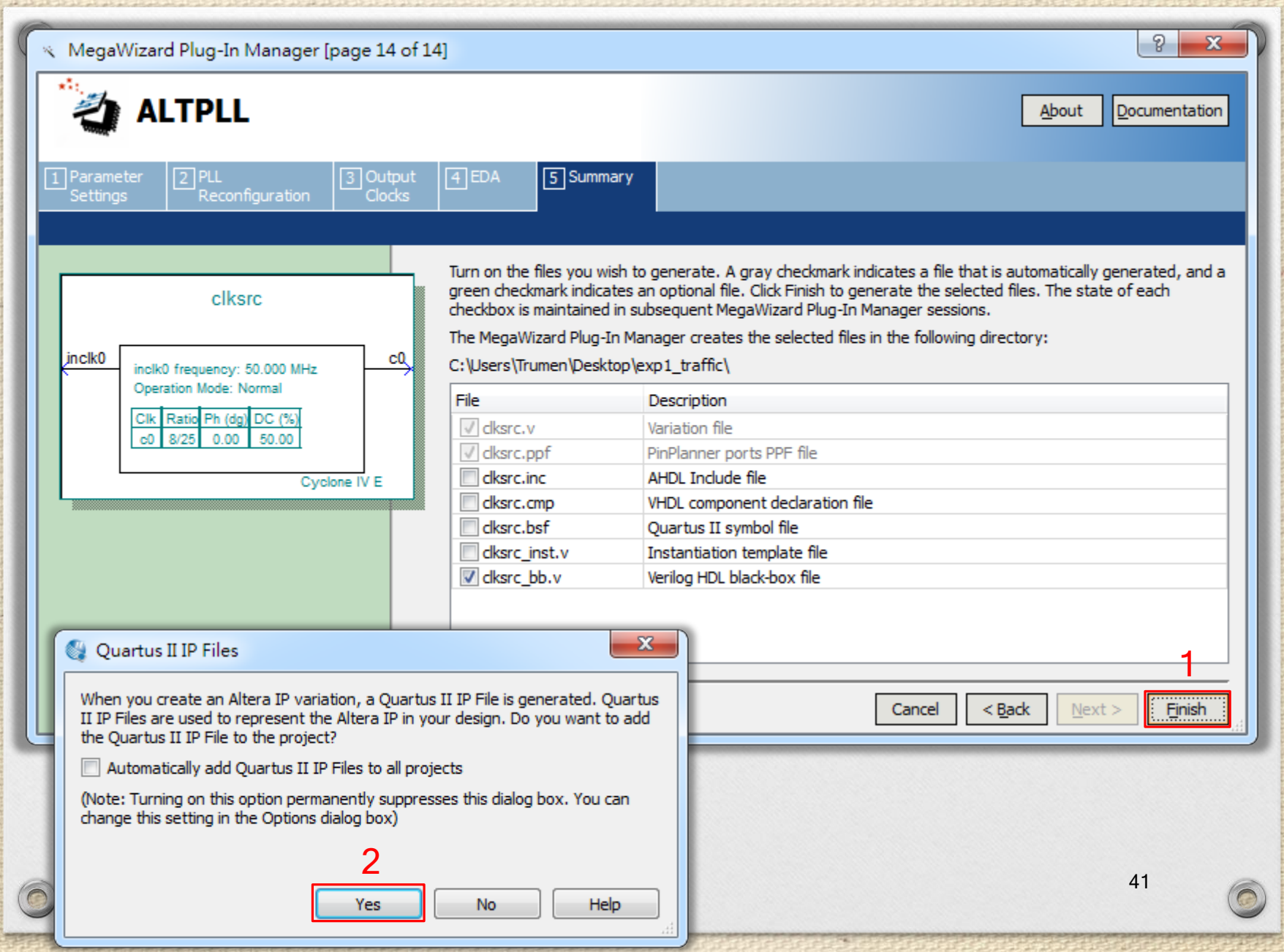

MegaWizard Plug-In Manager [page 4 of 14]
ALTPLL
About
Documentation
1 Parameter Settings
2 PLL Reconfiguration
3 Output Clocks
4 EDA
5 Summary
General/Modes
Inputs/Lock
Bandwidth/SS
Clock switchover
clksrc
inclk0
c0
inclk0 frequency: 50.000 MHz
Operation Mode: Normal
Clk
Ratio
Ph (dg)
DC (%)
c0
1/1
0.00
50.00
Cyclone IV E
Able to implement the requested PLL
Optional Inputs
Create an 'pllena' input to selectively enable the PLL
Create an 'areset' input to asynchronously reset the PLL
Create an 'pfdena' input to selectively enable the phase/frequency detector
Lock Output
Create 'locked' output
Enable self-reset on loss lock
Uncheck all the options
Advanced Parameters
Using these parameters is recommended for advanced users only
Create output file(s) using the 'Advanced' PLL parameters
- Configurations with output clock(s) that use cascade counters are not supported
1
Cancel
< Back
Next >
Finish


MegaWizard Plug-In Manager [page 5 of 14]
ALTPLL
About
Documentation
1 Parameter Settings
2 PLL Reconfiguration
3 Output Clocks
4 EDA
5 Summary
General/Modes
Inputs/Lock
Bandwidth/SS
Clock switchover
clksrc
inclk0
c0
inclk0 frequency: 50.000 MHz
Operation Mode: Normal
Clk
Ratio
Ph (dg)
DC (%)
c0
1/1
0.00
50.00
Cyclone IV E
Able to implement the requested PLL
Spread Spectrum
The spread spectrum feature allows for a modulation of the PLL clock frequency. The range of the clock frequency deviation is determined by the 'down spread' while 'modulation frequency' controls their period.
More Details >>
Use spread spectrum feature and
Set down spread to
0.500
percent
Set modulation frequency to
50.000
KHz
Bandwidth
A lower bandwidth will result in better input jitter rejection and less drift during switchover at the expense of a slower PLL lock time.
More Details >>
How would you like to specify the bandwidth setting?
Auto
Preset
Low
Custom:
Set bandwidth to
1.000
MHz
Actual achieved bandwidth
0.000000
MHz
1
37
Cancel
< Back
Next >
Finish


MegaWizard Plug-In Manager [page 6 of 14]
ALTPLL
About
Documentation
1 Parameter Settings
2 PLL Reconfiguration
3 Output Clocks
4 EDA
5 Summary
General/Modes
Inputs/Lock
Bandwidth/SS
Clock switchover
clksrc
inclk0
c0
inclk0 frequency: 50.000 MHz
Operation Mode: Normal
Clk
Ratio
Ph (dg)
DC (%)
c0
1/1
0.00
50.00
Cyclone IV E
Able to implement the requested PLL
Clock Switchover
Create an 'inclk1' input for a second input clock
What is the frequency of the 'inclk1' input?
100.000
MHz
Input Clock Switch
Create a 'clkswitch' input to manually select between the input clocks
(The 'clkswitch' input will behave as an input clock selection control input)
Allow PLL to automatically control the switching between input clocks
(The 'clkswitch' input will behave as a manual override control input)
Create a 'clkswitch' input to dynamically control the switching between input clocks
Perform the input clock switchover after
1
input clock cycles
Create an 'activeclock' output to indicate the input clock being used
(0 inclk0 is being used/ 1 inclk1 is being used)
Create a 'clkbad' output for each input clock
(0 input clock is toggling/ 1 input clock is not toggling)
1
Cancel
< Back
Next >
Finish


MegaWizard Plug-In Manager [page 7 of 14]
ALTPLL
About
Documentation
1 Parameter Settings
2 PLL Reconfiguration
3 Output Clocks
4 EDA
5 Summary
clksrc
inclk0
inclk0 frequency: 50.000 MHz
Operation Mode: Normal
Clk
Ratio
Ph (dg)
DC (%)
c0
1/1
0.00
50.00
c0
Cyclone IV E
Dynamic Reconfiguration
Create optional inputs for dynamic reconfiguration
Used for non-phase (e.g. frequency, duty cycle, bandwidth, etc.) reconfiguration
- Note: Reconfiguration with cascaded counters may not work correctly
Initial Configuration File
Use the following initial configuration file to initialize the altpll_reconfig megafunction (Valid file formats are the Hexadecimal (Intel-format) [.hex] and the Memory Initialization File [.mif]).
File name: .\clksrc.mif
Browse ...
Additional Configuration File
You may create additional configuration file(s) for the current PLL settings. These files may be used to initialize the altpll_reconfig megafunction.
To create a configuration file, enter a valid file name and press the 'Generate A Configuration File' button (Valid file formats are the Hexadecimal (Intel-format) [.hex] and the Memory Initialization File [.mif]).
File name:
Browse ...
Generate a Configuration File
Dynamic Phase Reconfiguration
Enable phase shift step resolution
Create optional inputs for dynamic phase reconfiguration
1
Cancel
< Back
Next > 39
Finish

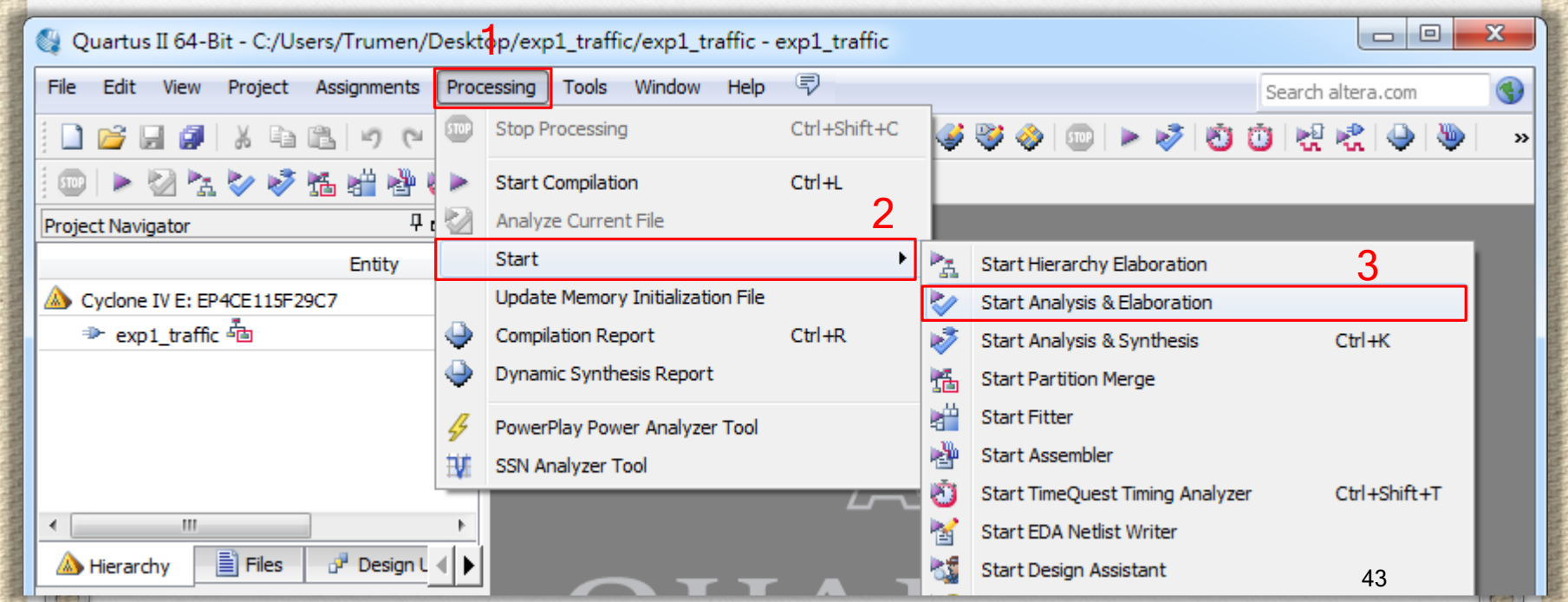

MegaWizard Plug-In Manager [page 8 of 14]
ALTPLL
About
Documentation
1 Parameter Settings
2 PLL Reconfiguration
3 Output Clocks
4 EDA
5 Summary
clk c0
clk c1
clk c2
clk c3
clk c4
clksrc
inclk0
inclk0 frequency: 50.000 MHz
Operation Mode: Normal
Clk
Ratio
Ph (dg)
DC (%)
c0
8/25
0.00
50.00
c0
Cyclone IV E
c0 - Core/External Output Clock
Able to implement the requested PLL
Use this clock
Clock Tap Settings
Requested Settings
Actual Settings
1
Enter output clock frequency:
16
MHz
16.000000
Enter output clock parameters:
Clock multiplication factor
1
8
Clock division factor
1
<< Copy
25
Clock phase shift
0.00
deg
0.00
Clock duty cycle (%)
50.00
50.00
Description
Value
Primary clock VCO frequency (MHz)
4...
Modulus for M counter
8
Note: The displayed internal settings of the PLL is recommended for use by advanced users only
Per Clock Feasibility Indicators
c0
c1
c2
c3
c4
40
Cancel
< Back
Next >
Finish
2

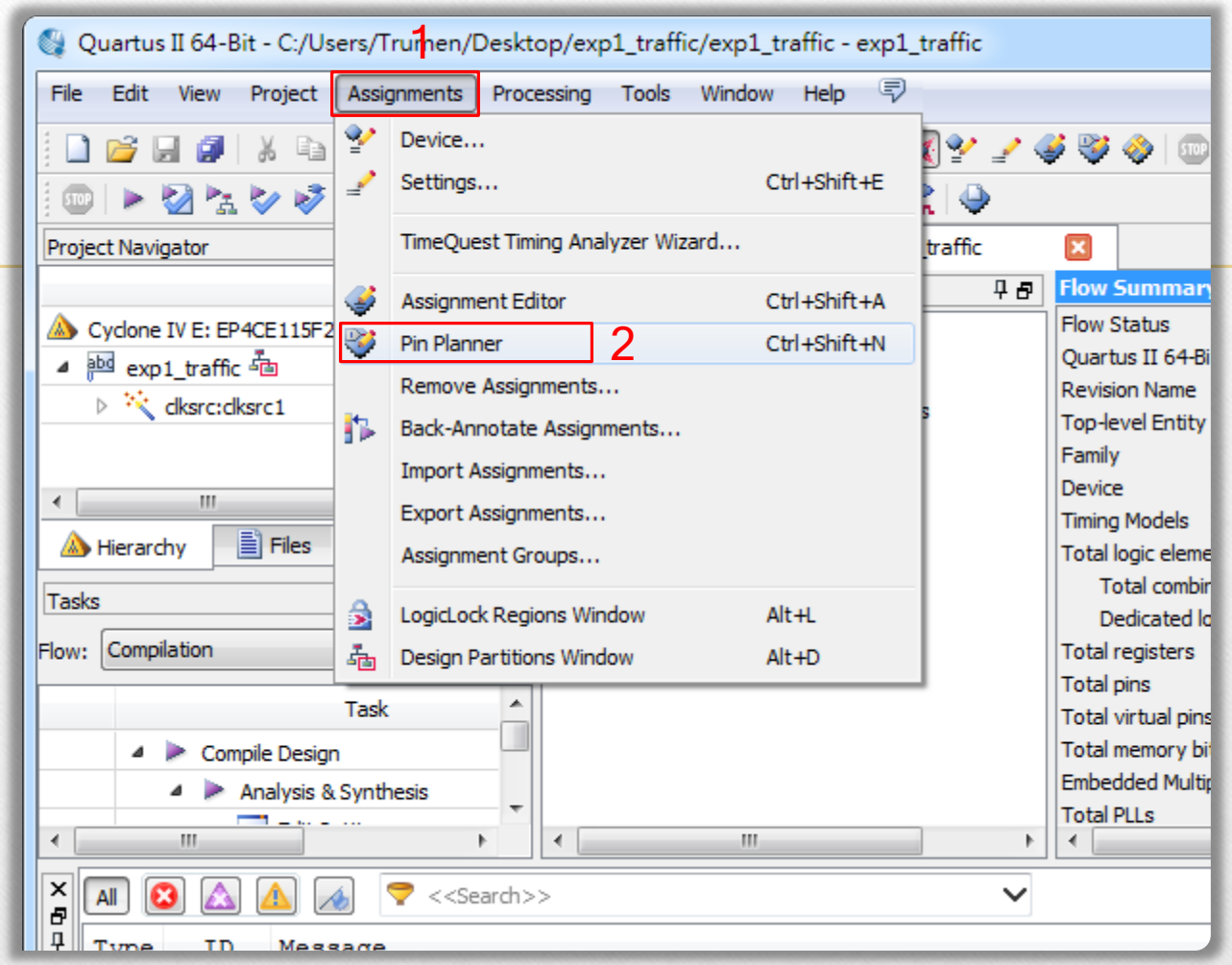

MegaWizard Plug-In Manager [page 14 of 14]
ALTPLL
About
Documentation
1 Parameter Settings
2 PLL Reconfiguration
3 Output Clocks
4 EDA
5 Summary
clksrc
inclk0
c0
inclk0 frequency: 50.000 MHz
Operation Mode: Normal
Clk
Ratio
Ph (dg)
DC (%)
c0
8/25
0.00
50.00
Cyclone IV E
Turn on the files you wish to generate. A gray checkmark indicates a file that is automatically generated, and a green checkmark indicates an optional file. Click Finish to generate the selected files. The state of each checkbox is maintained in subsequent MegaWizard Plug-In Manager sessions.
The MegaWizard Plug-In Manager creates the selected files in the following directory:
C:\Users\Trumen\Desktop\exp1_traffic\
File
Description
clksrc.v
Variation file
clksrc.ppf
PinPlanner ports PPF file
clksrc.inc
AHDL Include file
clksrc.cmp
VHDL component declaration file
clksrc.bsf
Quartus II symbol file
clksrc_inst.v
Instantiation template file
clksrc_bb.v
Verilog HDL black-box file
1
Cancel
< Back
Next >
Finish
Quartus II IP Files
When you create an Altera IP variation, a Quartus II IP File is generated. Quartus II IP Files are used to represent the Altera IP in your design. Do you want to add the Quartus II IP File to the project?
Automatically add Quartus II IP Files to all projects
(Note: Turning on this option permanently suppresses this dialog box. You can change this setting in the Options dialog box)
2
Yes
No
Help

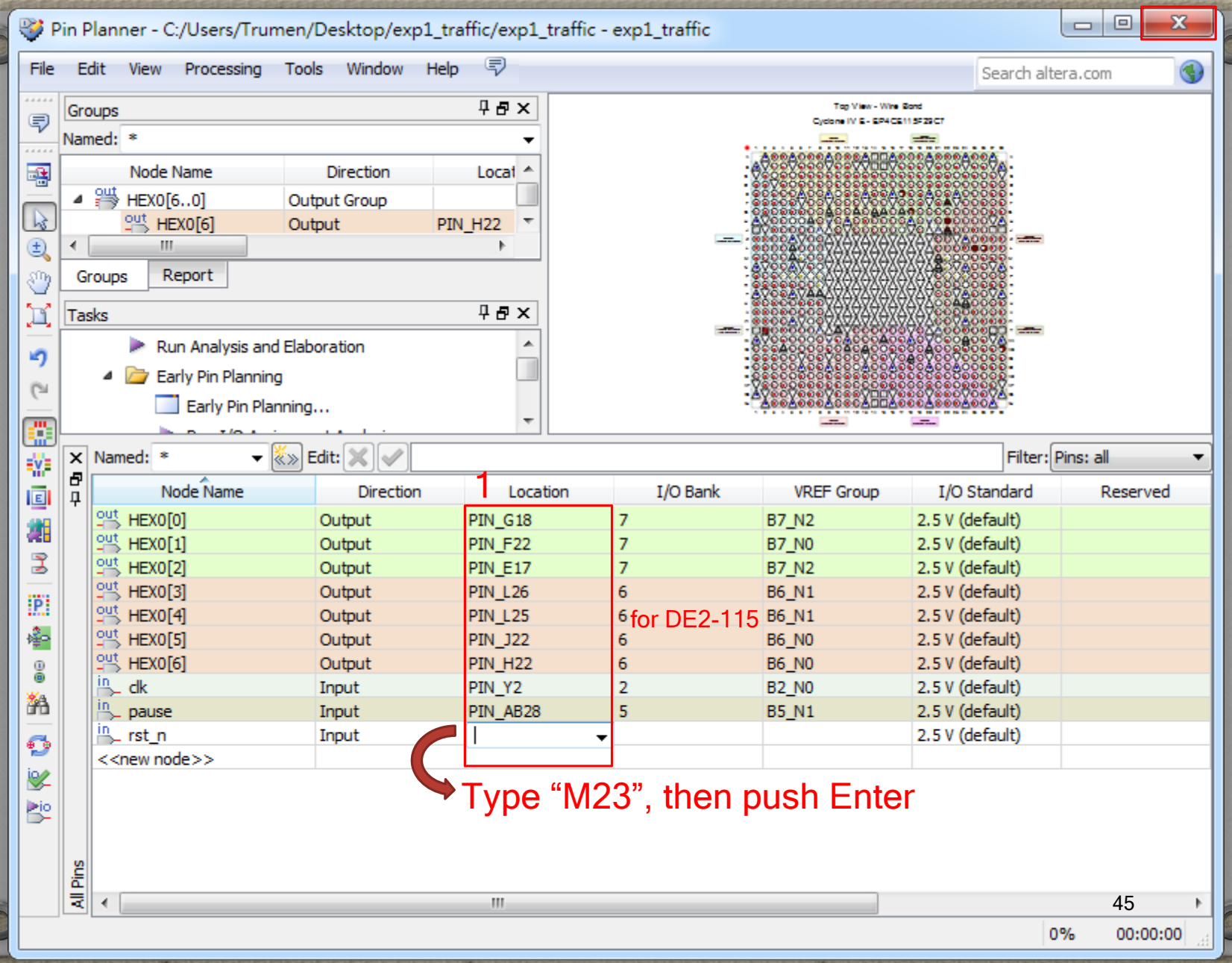
Assign the Pins

Assign the Pins

- Before making pin assignments…




1
Quartus II 64-Bit - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - exp1_traffic
File
Edit
View
Project
Assignments
Processing
Tools
Window
Help
Device...
Settings... Ctrl+Shift+E
TimeQuest Timing Analyzer Wizard...
Assignment Editor Ctrl+Shift+A
Pin Planner Ctrl+Shift+N
2
Remove Assignments...
Back-Annotate Assignments...
Import Assignments...
Export Assignments...
Assignment Groups...
LogicLock Regions Window Alt+L
Design Partitions Window Alt+D
Project Navigator
Cyclone IV E: EP4CE115F2
exp1_traffic
clksrc:clksrc1
Hierarchy
Files
Tasks
Flow: Compilation
Task
Compile Design
Analysis & Synthesis
Flow Summary
Flow Status
Quartus II 64-Bi
Revision Name
Top-level Entity
Family
Device
Timing Models
Total logic eleme
Total combin
Dedicated lo
Total registers
Total pins
Total virtual pins
Total memory bi
Embedded Multip
Total PLLs
All
<<Search>>
Type
ID
Message

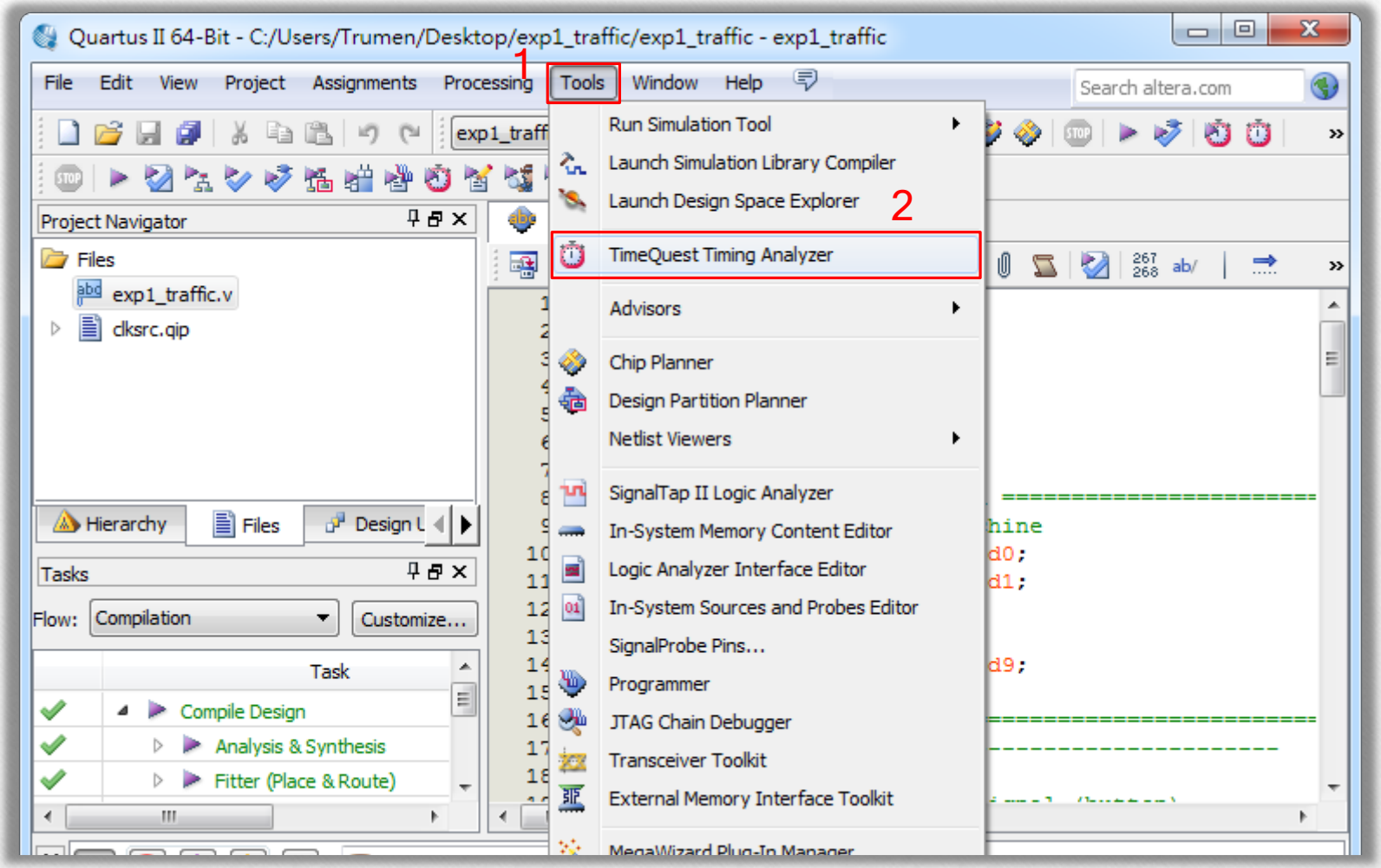


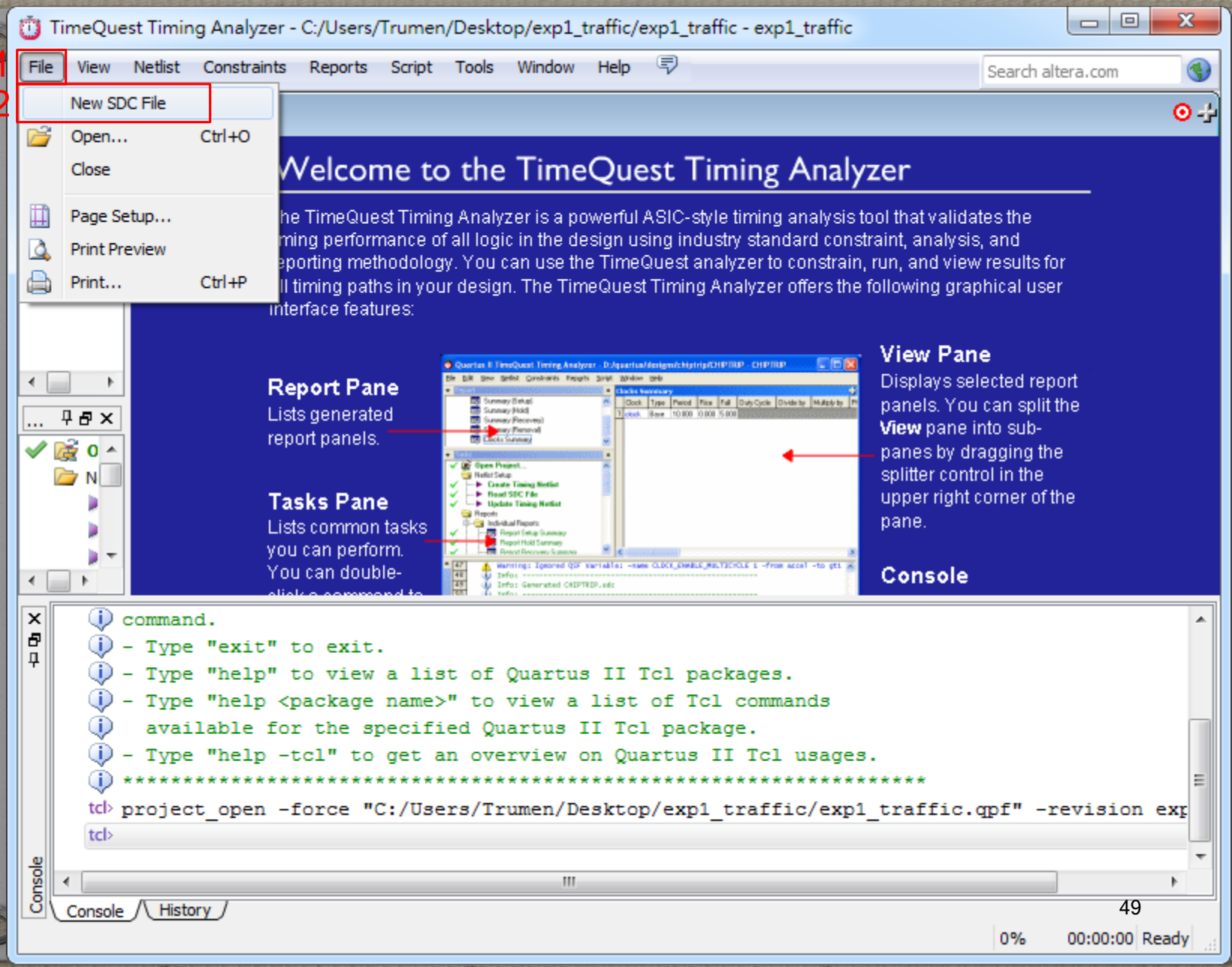
Now, you are finished creating your Quartus II design!

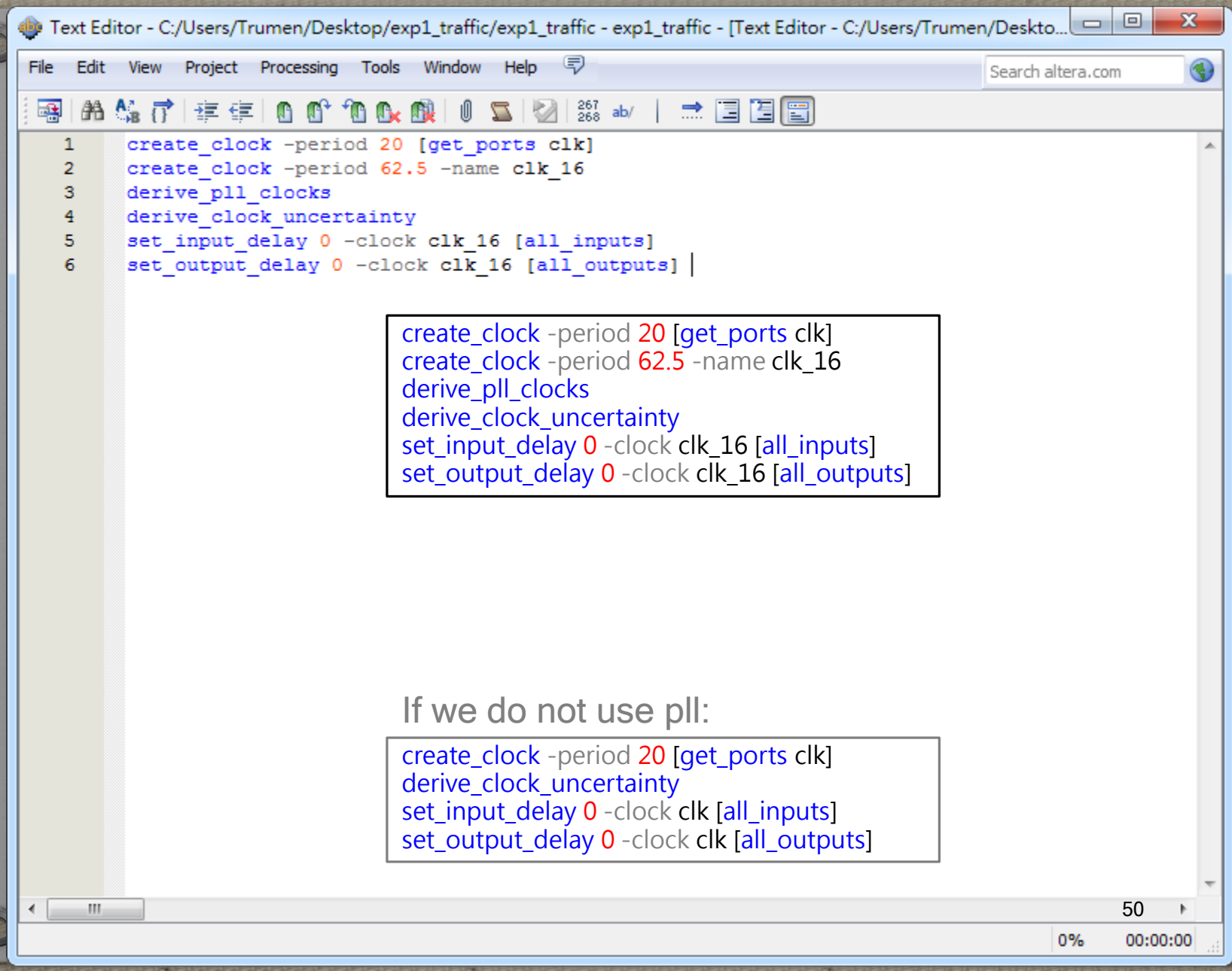
Create a Default TimeQuest SDC File

Create a Default TimeQuest SDC File

- Timing settings are critically important for a successful design.
- For this tutorial you will create a basic Synopsys Design Constraints File (.sdc) that the Quartus II TimeQuest Timing Analyzer uses during design compilation.
- For more complex designs, you will need to consider the timing requirements more carefully.


Quartus II 64-Bit - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - exp1_traffic
File
Edit
View
Project
Assignments
Processing
Tools
Window
Help
1
Search altera.com
Run Simulation Tool
Launch Simulation Library Compiler
Launch Design Space Explorer
2
TimeQuest Timing Analyzer
Advisors
Chip Planner
Design Partition Planner
Netlist Viewers
SignalTap II Logic Analyzer
In-System Memory Content Editor
Logic Analyzer Interface Editor
In-System Sources and Probes Editor
SignalProbe Pins...
Programmer
JTAG Chain Debugger
Transceiver Toolkit
External Memory Interface Toolkit
Project Navigator
Files
exp1_traffic.v
clksrc.qip
Hierarchy
Files
Tasks
Flow:
Compilation
Customize...
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)

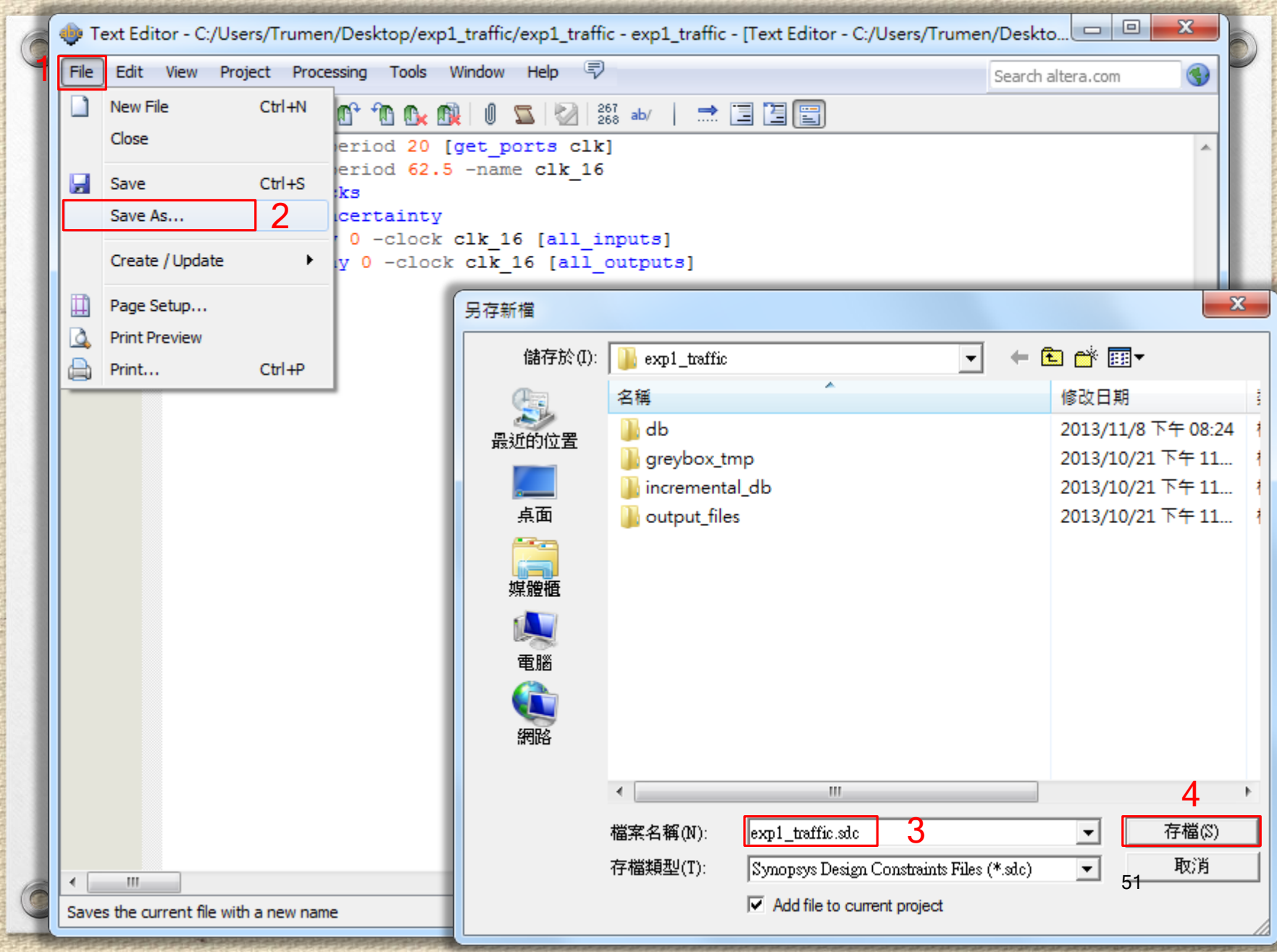

TimeQuest Timing Analyzer - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - exp1_traffic
1
File
View
Netlist
Constraints
Reports
Script
Tools
Window
Help
Search altera.com
2
New SDC File
Open... Ctrl+O
Close
Page Setup...
Print Preview
Print... Ctrl+P
Welcome to the TimeQuest Timing Analyzer
he TimeQuest Timing Analyzer is a powerful ASIC-style timing analysis tool that validates the
ming performance of all logic in the design using industry standard constraint, analysis, and
porting methodology. You can use the TimeQuest analyzer to constrain, run, and view results for
l timing paths in your design. The TimeQuest Timing Analyzer offers the following graphical user
interface features:
Report Pane
Lists generated report panels.
Tasks Pane
Lists common tasks you can perform. You can double-
View Pane
Displays selected report panels. You can split the View pane into sub-panes by dragging the splitter control in the upper right corner of the pane.
Console
command.
- Type "exit" to exit.
- Type "help" to view a list of Quartus II Tcl packages.
- Type "help <package name>" to view a list of Tcl commands
available for the specified Quartus II Tcl package.
- Type "help -tcl" to get an overview on Quartus II Tcl usages.
tcl> project_open -force "C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic.qpf" -revision exp
tcl>
Console
Console
History
0%
00:00:00
Ready




```
create_clock -period 20 [get_ports clk]
create_clock -period 62.5 -name clk_16
derive_pll_clocks
derive_clock_uncertainty
set_input_delay 0 -clock clk_16 [all_inputs]
set_output_delay 0 -clock clk_16 [all_outputs]
```

If we do not use pll:

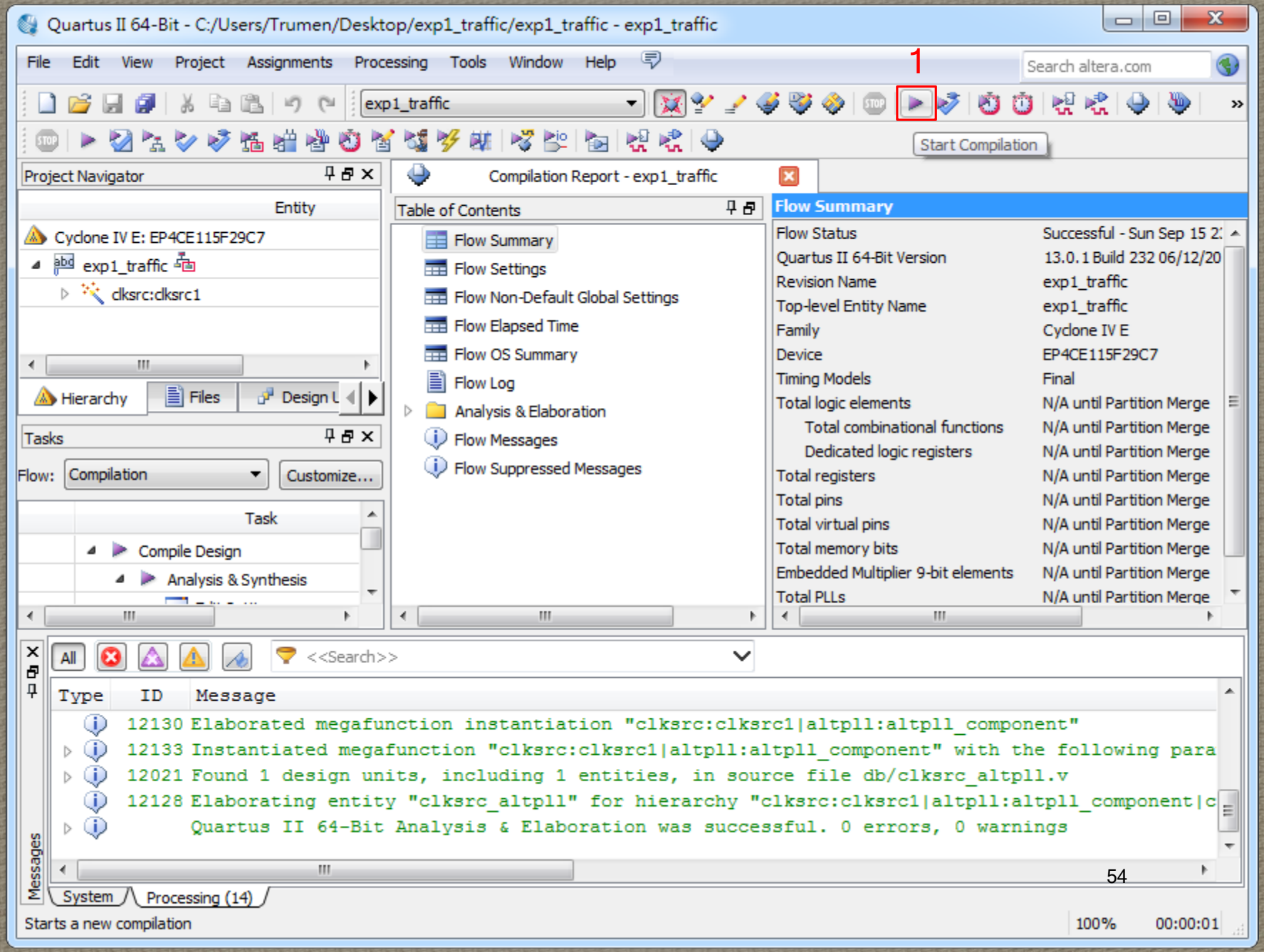
```
create_clock -period 20 [get_ports clk]
derive_clock_uncertainty
set_input_delay 0 -clock clk [all_inputs]
set_output_delay 0 -clock clk [all_outputs]
```


Text Editor - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - exp1_traffic - [Text Editor - C:/Users/Trumen/Deskto...
1
File
Edit
View
Project
Processing
Tools
Window
Help
Search altera.com
New File Ctrl+N
Close
Save Ctrl+S
Save As...
2
Create / Update
Page Setup...
Print Preview
Print... Ctrl+P
eriod 20 [get_ports clk]
eriod 62.5 -name clk_16
ks
certainty
0 -clock clk_16 [all_inputs]
y 0 -clock clk_16 [all_outputs]
Saves the current file with a new name
另存新檔
儲存於(I):
exp1_traffic
名稱
修改日期
db
2013/11/8 下午 08:24
greybox_tmp
2013/10/21 下午 11...
incremental_db
2013/10/21 下午 11...
output_files
2013/10/21 下午 11...
最近的位置
桌面
媒體櫃
電腦
網路
4
檔案名稱(N):
exp1_traffic.sdc
3
存檔(S)
存檔類型(T):
Synopsys Design Constraints Files (*.sdc)
取消
Add file to current project

Compile and Verify Your Design

Compile Your Design

- After creating your design you must compile it.
- Compilation converts the design into a bitstream that can be downloaded into the FPGA.
- The most important output of compilation is an SRAM Object File (.sof), which you use to program the device.


Quartus II 64-Bit - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - exp1_traffic
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
1
exp1_traffic
Start Compilation
Project Navigator
Entity
Cyclone IV E: EP4CE115F29C7
exp1_traffic
clksrc:clksrc1
Hierarchy Files Design U
Tasks
Flow: Compilation Customize...
Task
Compile Design
Analysis & Synthesis
Compilation Report - exp1_traffic
Table of Contents
Flow Summary
Flow Settings
Flow Non-Default Global Settings
Flow Elapsed Time
Flow OS Summary
Flow Log
Analysis & Elaboration
Flow Messages
Flow Suppressed Messages
Flow Summary
Flow Status Successful - Sun Sep 15 2
Quartus II 64-Bit Version 13.0.1 Build 232 06/12/20
Revision Name exp1_traffic
Top-level Entity Name exp1_traffic
Family Cyclone IV E
Device EP4CE115F29C7
Timing Models Final
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
All
<<Search>>
Type ID Message
12130 Elaborated megafunction instantiation "clksrc:clksrc1|altpll:altpll_component"
12133 Instantiated megafunction "clksrc:clksrc1|altpll:altpll_component" with the following para
12021 Found 1 design units, including 1 entities, in source file db/clksrc_altpll.v
12128 Elaborating entity "clksrc_altpll" for hierarchy "clksrc:clksrc1|altpll:altpll_component|c
Quartus II 64-Bit Analysis & Elaboration was successful. 0 errors, 0 warnings
Messages
System Processing (14)
54
Starts a new compilation
100% 00:00:01

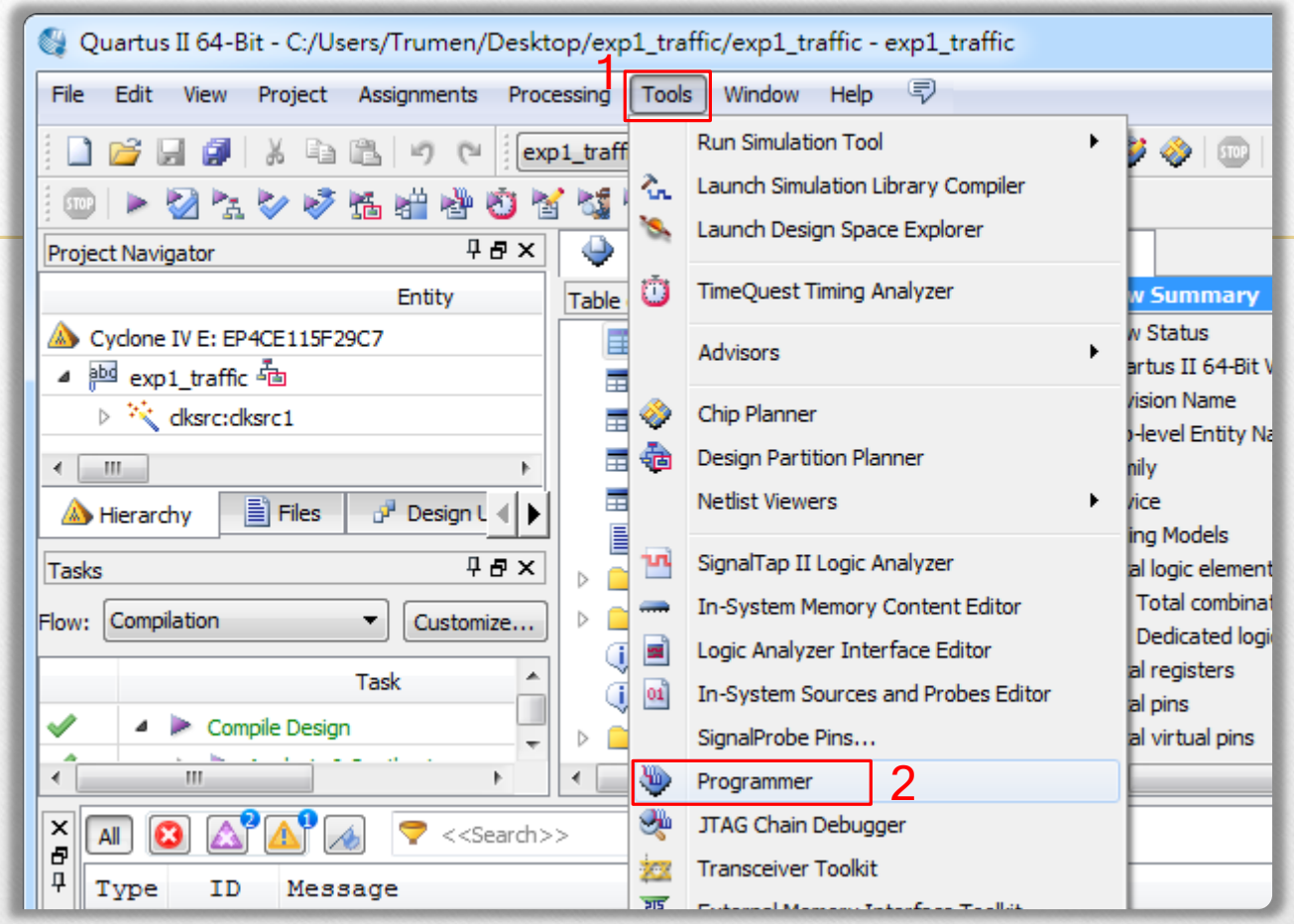
Compilation Report

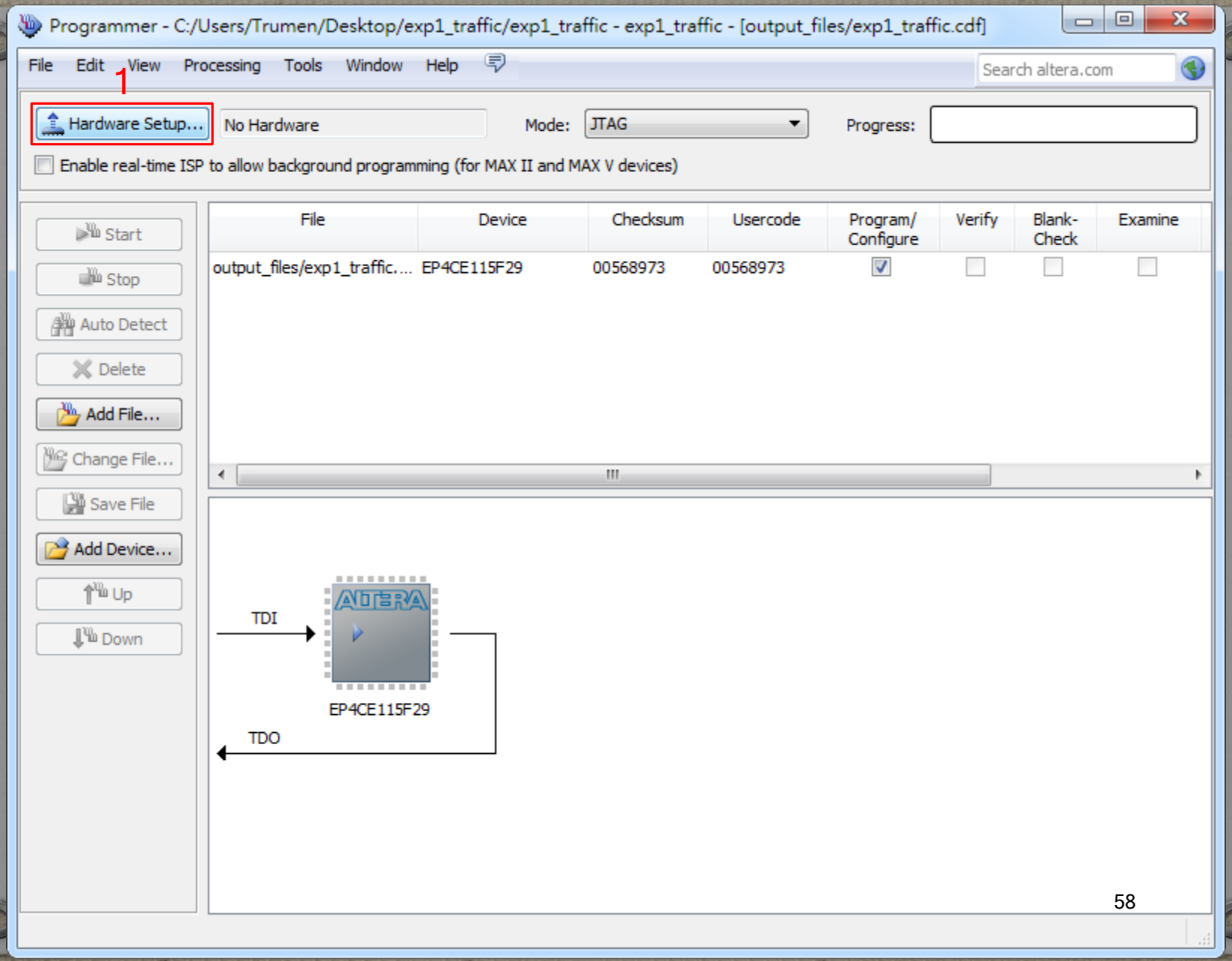
- Make sure there is no error.

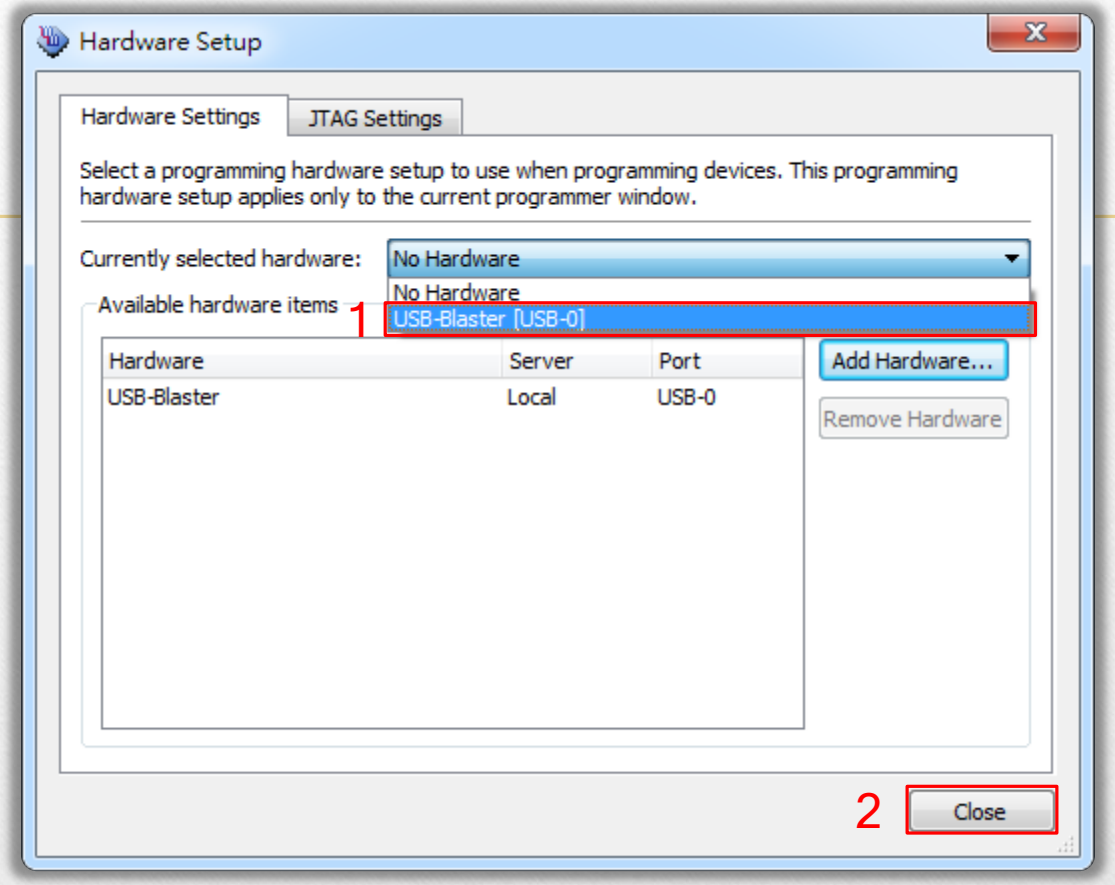
Flow Summary	
Flow Status	Successful - Mon Sep 16 14:59:00 2013
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	exp1_traffic
Top-level Entity Name	exp1_traffic
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	37 / 114,480 (< 1 %)
Total combinational functions	37 / 114,480 (< 1 %)
Dedicated logic registers	36 / 114,480 (< 1 %)
Total registers	36
Total pins	10 / 529 (2 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	1 / 4 (25 %)

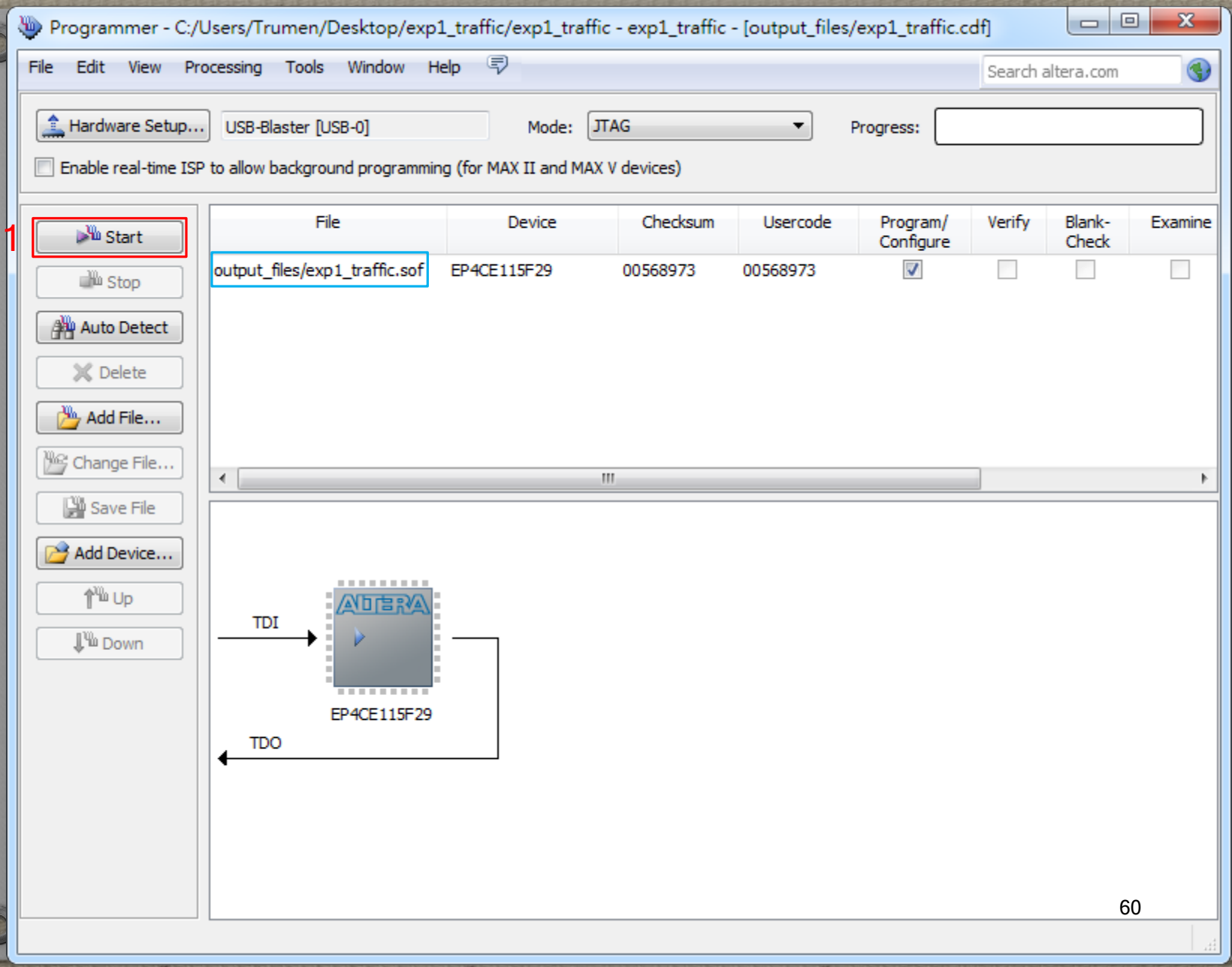
Program the FPGA Device

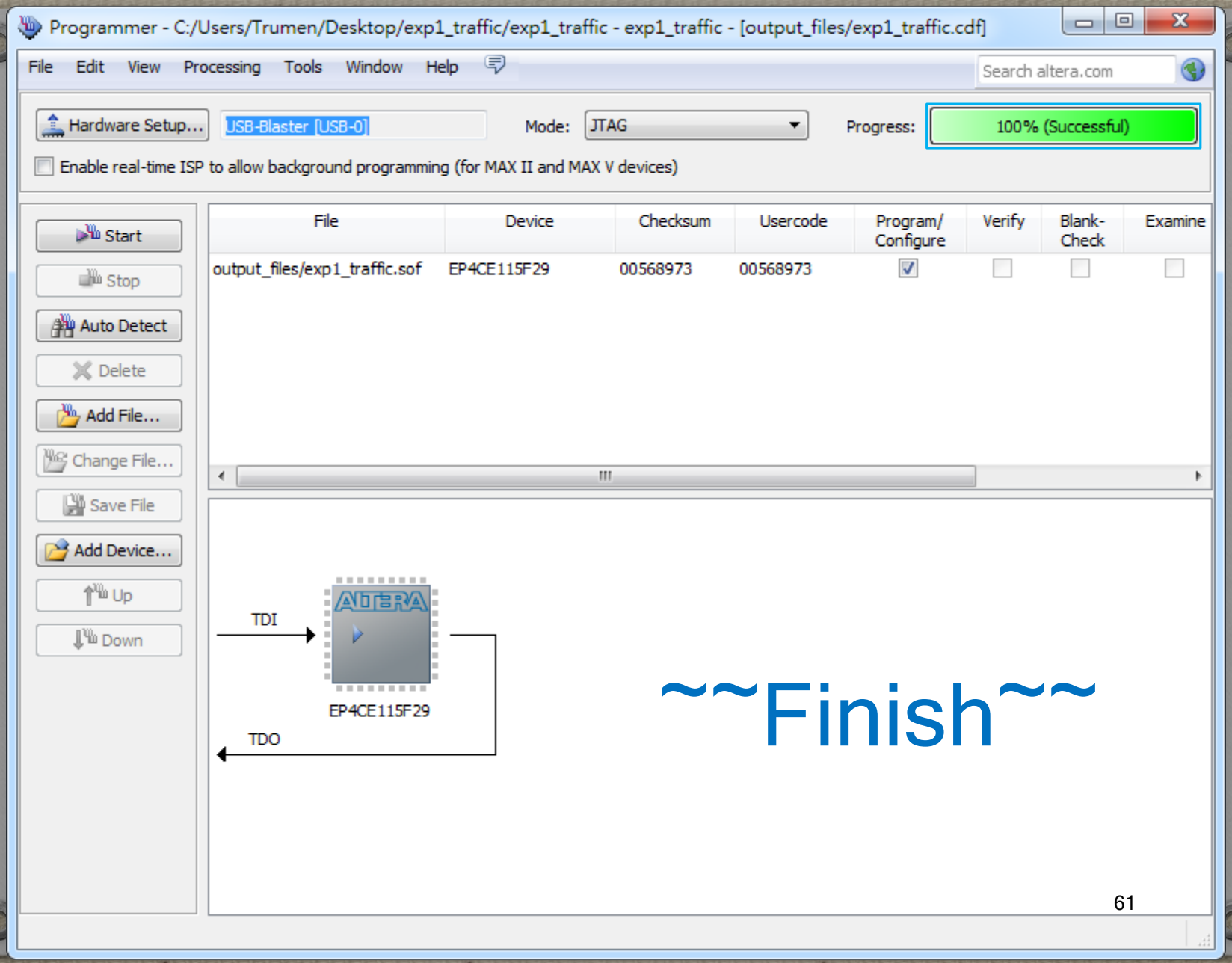
- After compiling and verifying your design you are ready to program the FPGA on the development board.
- You download the SOF you just created into the FPGA using the USB-Blaster circuitry on the board.


Quartus II 64-Bit - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - exp1_traffic
File
Edit
View
Project
Assignments
Processing
Tools
Window
Help
1
Run Simulation Tool
Launch Simulation Library Compiler
Launch Design Space Explorer
TimeQuest Timing Analyzer
Advisors
Chip Planner
Design Partition Planner
Netlist Viewers
SignalTap II Logic Analyzer
In-System Memory Content Editor
Logic Analyzer Interface Editor
In-System Sources and Probes Editor
SignalProbe Pins...
Programmer
2
JTAG Chain Debugger
Transceiver Toolkit
Project Navigator
Entity
Cyclone IV E: EP4CE115F29C7
exp1_traffic
clksrc:clksrc1
Hierarchy
Files
Tasks
Flow:
Compilation
Customize...
Task
Compile Design
Table
All
<<Search>>
Type
ID
Message


Programmer - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - exp1_traffic - [output_files/exp1_traffic.cdf]
File
Edit
View
Processing
Tools
Window
Help
Search altera.com
1
Hardware Setup...
No Hardware
Mode:
JTAG
Progress:
Enable real-time ISP to allow background programming (for MAX II and MAX V devices)
Start
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/exp1_traffic....
EP4CE115F29
00568973
00568973
TDI
TDO
EP4CE115F29

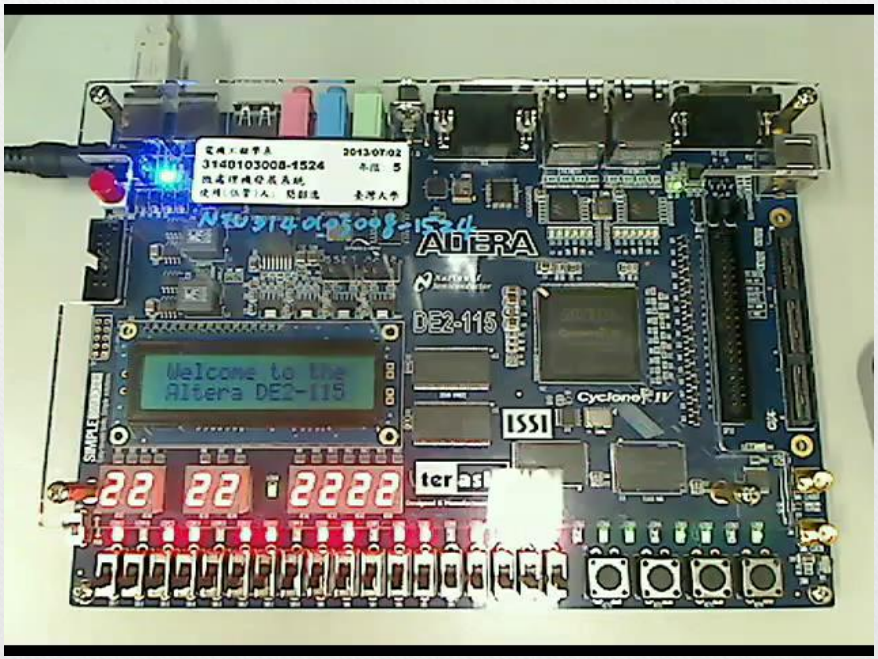

Hardware Setup
Hardware Settings
JTAG Settings
Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.
Currently selected hardware:
No Hardware
No Hardware
USB-Blaster [USB-0]
1
Available hardware items
Hardware
Server
Port
USB-Blaster
Local
USB-0
Add Hardware...
Remove Hardware
2
Close


Programmer - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - exp1_traffic - [output_files/exp1_traffic.cdf]
File Edit View Processing Tools Window Help
Search altera.com
Hardware Setup...
USB-Blaster [USB-0]
Mode: JTAG
Progress:
Enable real-time ISP to allow background programming (for MAX II and MAX V devices)
1
Start
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/exp1_traffic.sof
EP4CE115F29
00568973
00568973
TDI
TDO
EP4CE115F29


Programmer - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - exp1_traffic - [output_files/exp1_traffic.cdf]
File Edit View Processing Tools Window Help
Search altera.com
Hardware Setup...
USB-Blaster [USB-0]
Mode: JTAG
Progress: 100% (Successful)
Enable real-time ISP to allow background programming (for MAX II and MAX V devices)
Start
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/exp1_traffic.sof EP4CE115F29 00568973 00568973
TDI
TDO
EP4CE115F29
~~Finish~~

Demo Video

- 10 seconds countdown system



Configuring the Cyclone IV E FPGA

Configuring the FPGA

- JTAG programming

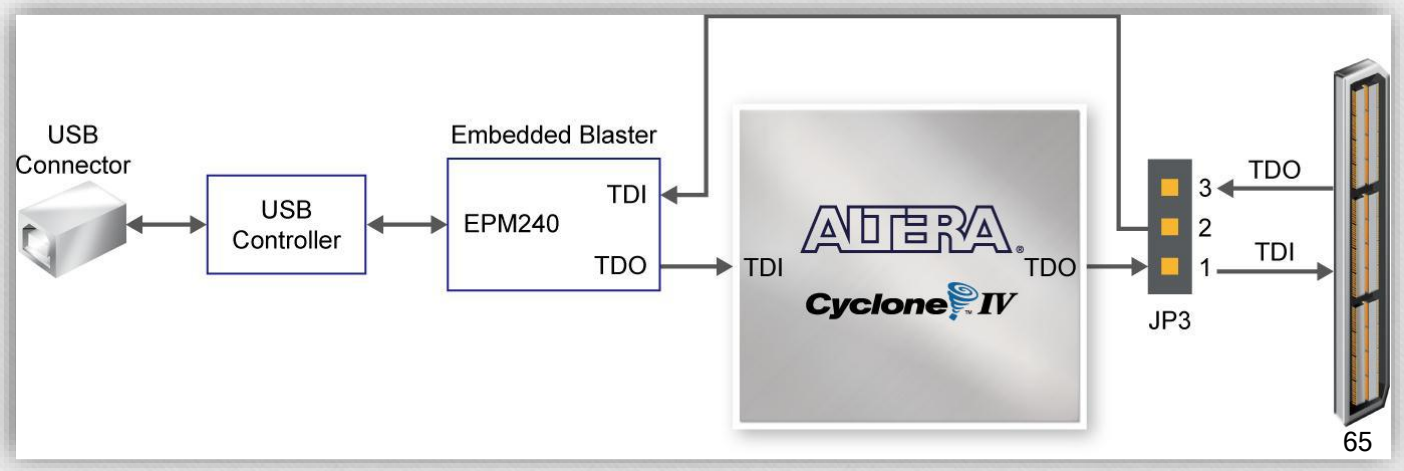
  In this method of programming, named after the IEEE standards Joint Test Action Group, the configuration bit stream is downloaded directly into the Cyclone IV E FPGA. The FPGA will retain this configuration as long as power is applied to the board; the configuration information will be lost when the power is turned off.

- AS programming

  In this method, called Active Serial programming, the configuration bit stream is downloaded into the Altera EPCS64 serial configuration device. It provides non-volatile storage of the bit stream, so that the information is retained even when the power supply to the DE2-115 board is turned off. When the board's power is turned on, the configuration data in the EPCS64 device is automatically loaded into the Cyclone IV E FPGA.

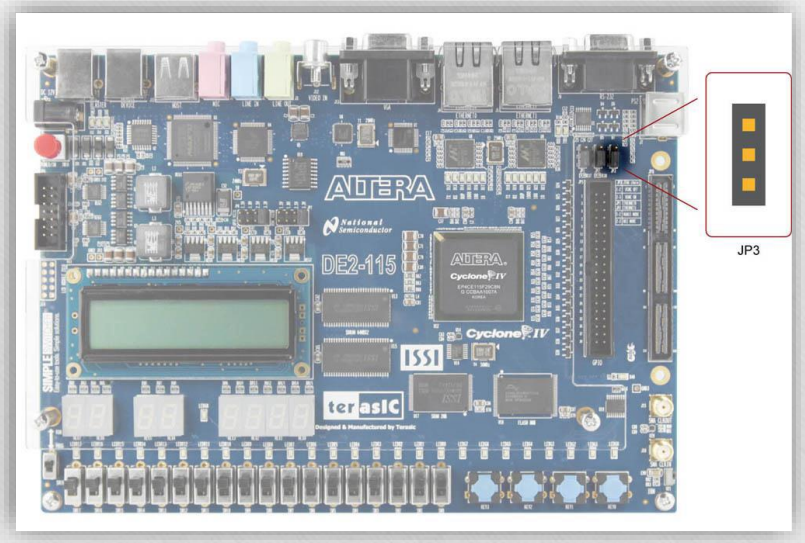
JTAG Chain (1/2)

- To use JTAG interface for configuring FPGA device, the JTAG chain on DE2-115 must form a close loop that allows Quartus II programmer to detect FPGA device.



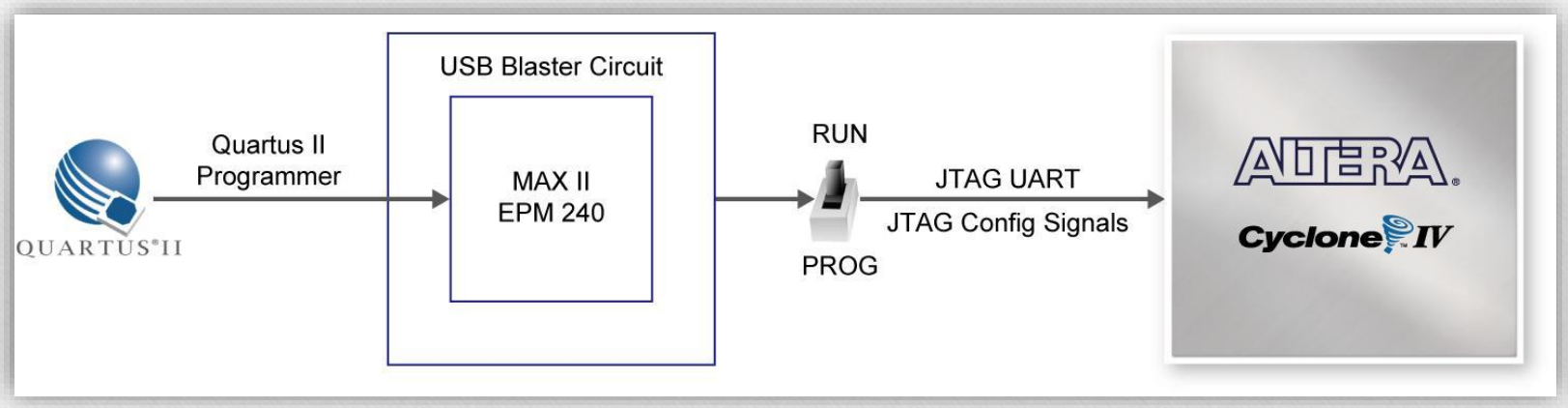
JTAG Chain (2/2)

- Shorting pin1 and pin2 on JP3 can disable the JTAG signals on HSMC connector that will form a close JTAG loop chain on DE2-115 board. Thus, only the on board FPGA device (Cyclone IV E) will be detected by Quartus II programmer.



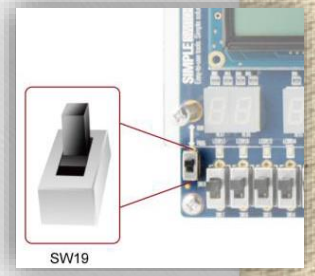
Configuring the FPGA in JTAG Mode (1/2)

- This figure illustrates the JTAG configuration setup.



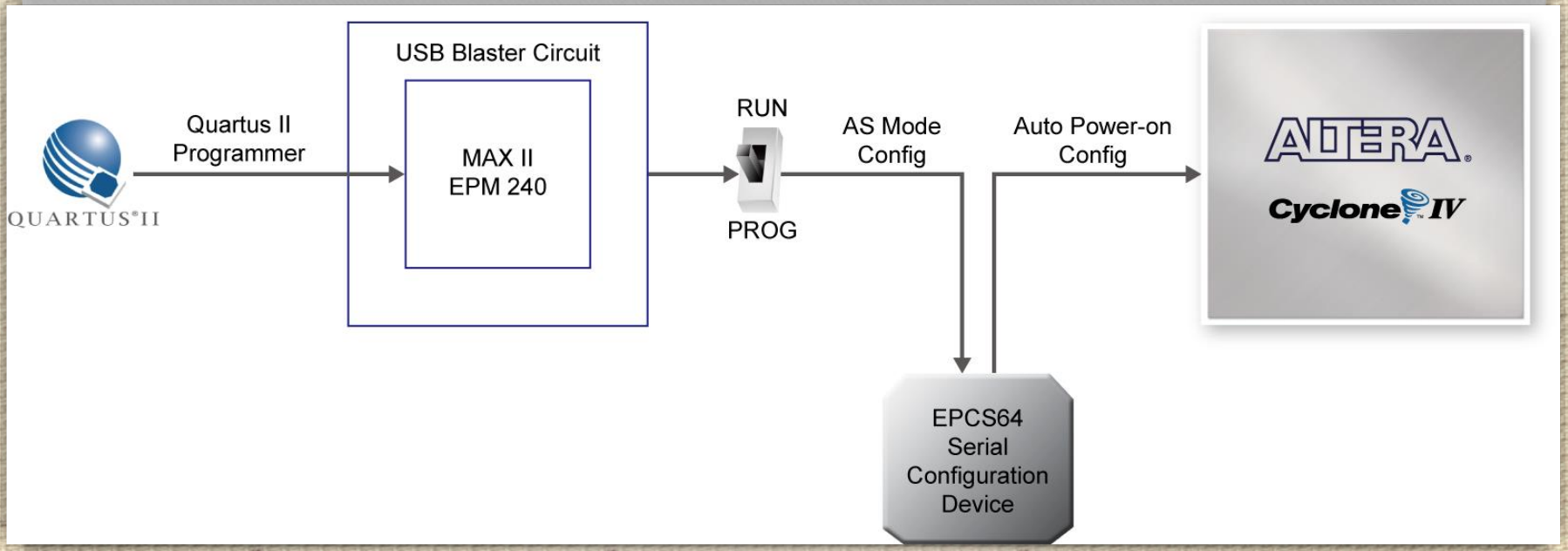
Configuring the FPGA in JTAG Mode (2/2)

1. Ensure that power is applied to the DE2-115 board.
2. Configure the JTAG programming circuit by setting the RUN/PROG slide switch (SW19) to the RUN position.
3. Connect the supplied USB cable to the USB Blaster port on the DE2-115 board.
4. The FPGA can now be programmed by using the Quartus II Programmer to select a configuration bit stream file with the .sof filename extension.



Configuring the EPCS64 in AS Mode (1/2)

- This figure illustrates the AS configuration setup.

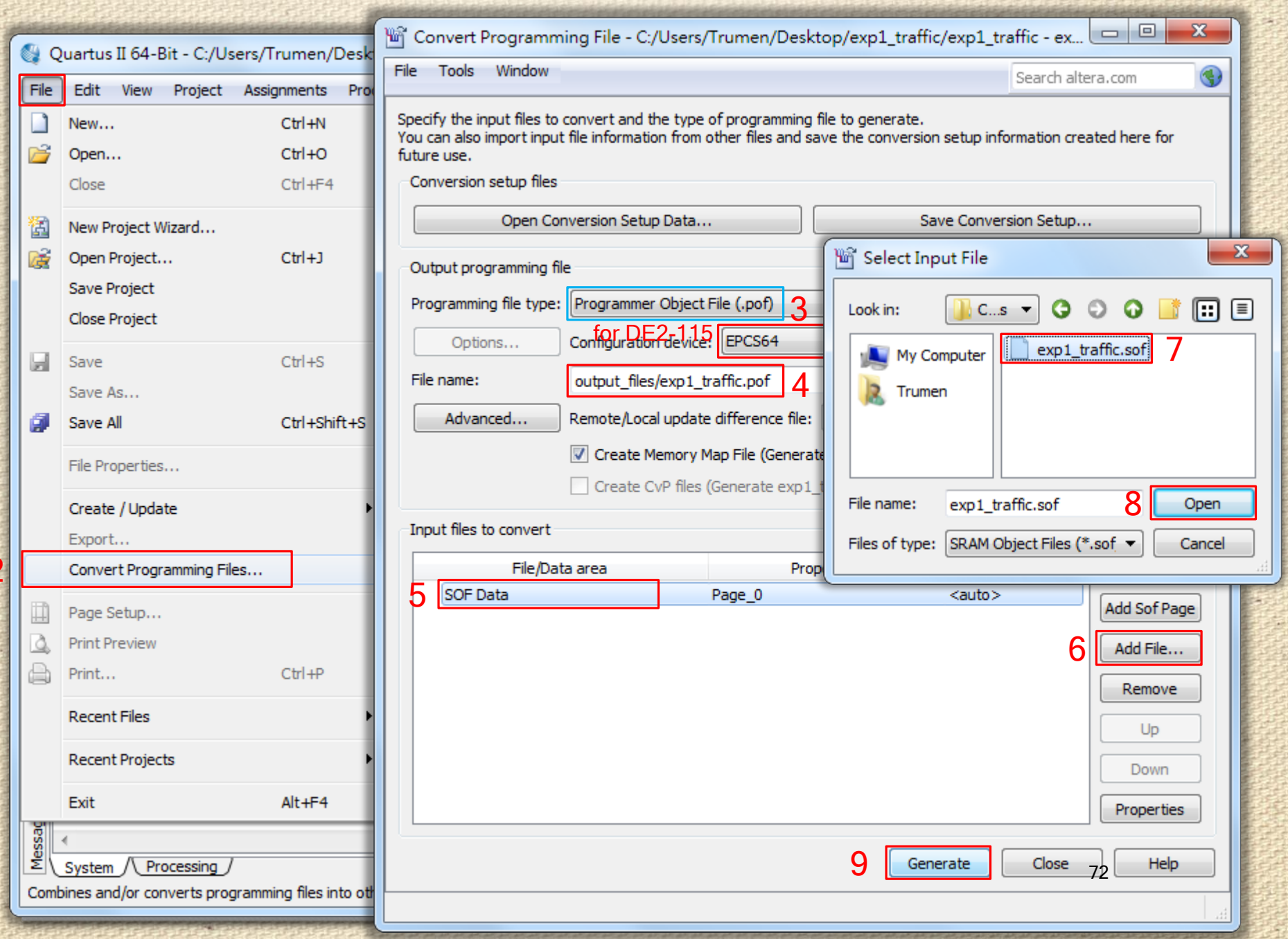


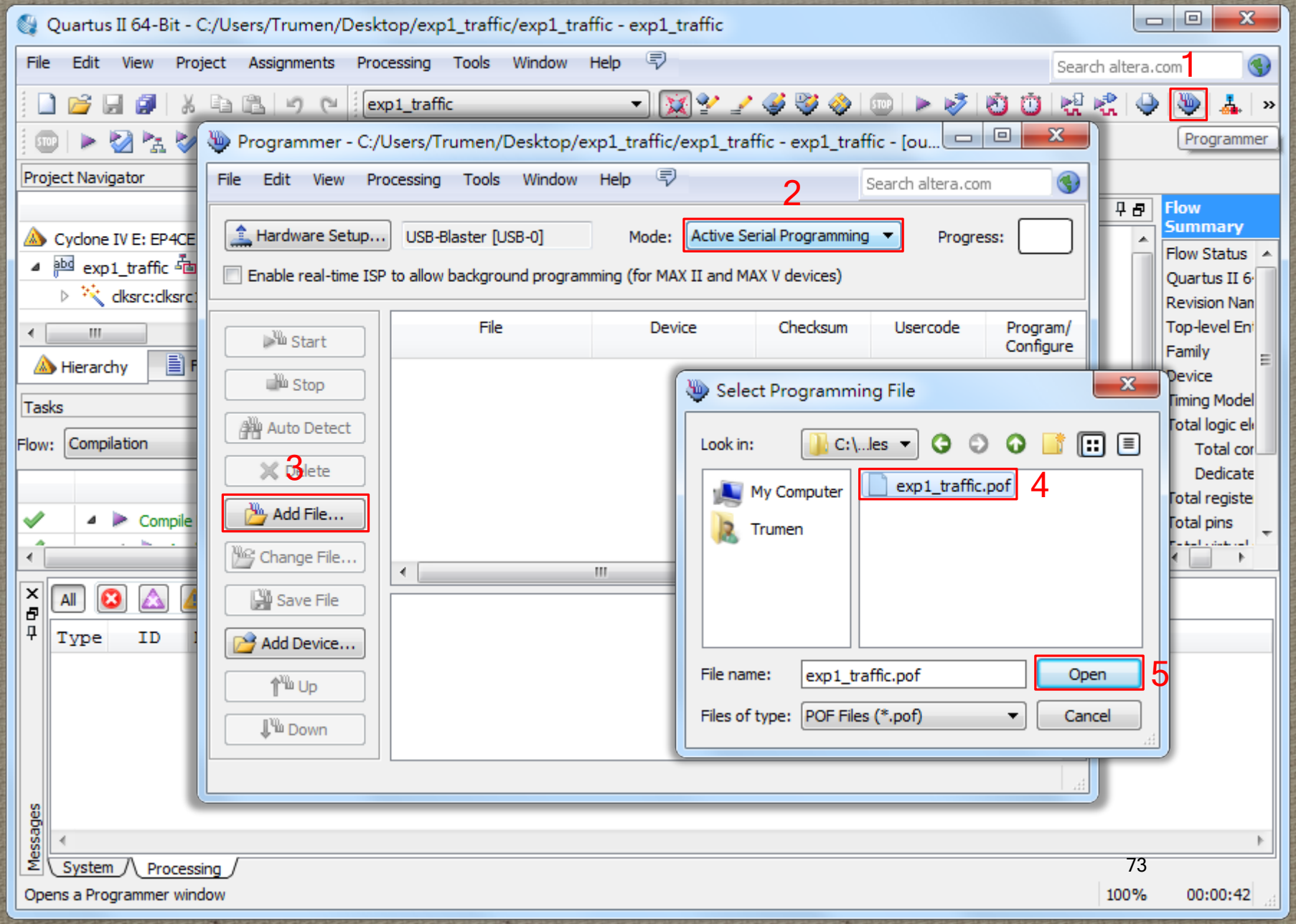
Configuring the EPCS64 in AS Mode (2/2)

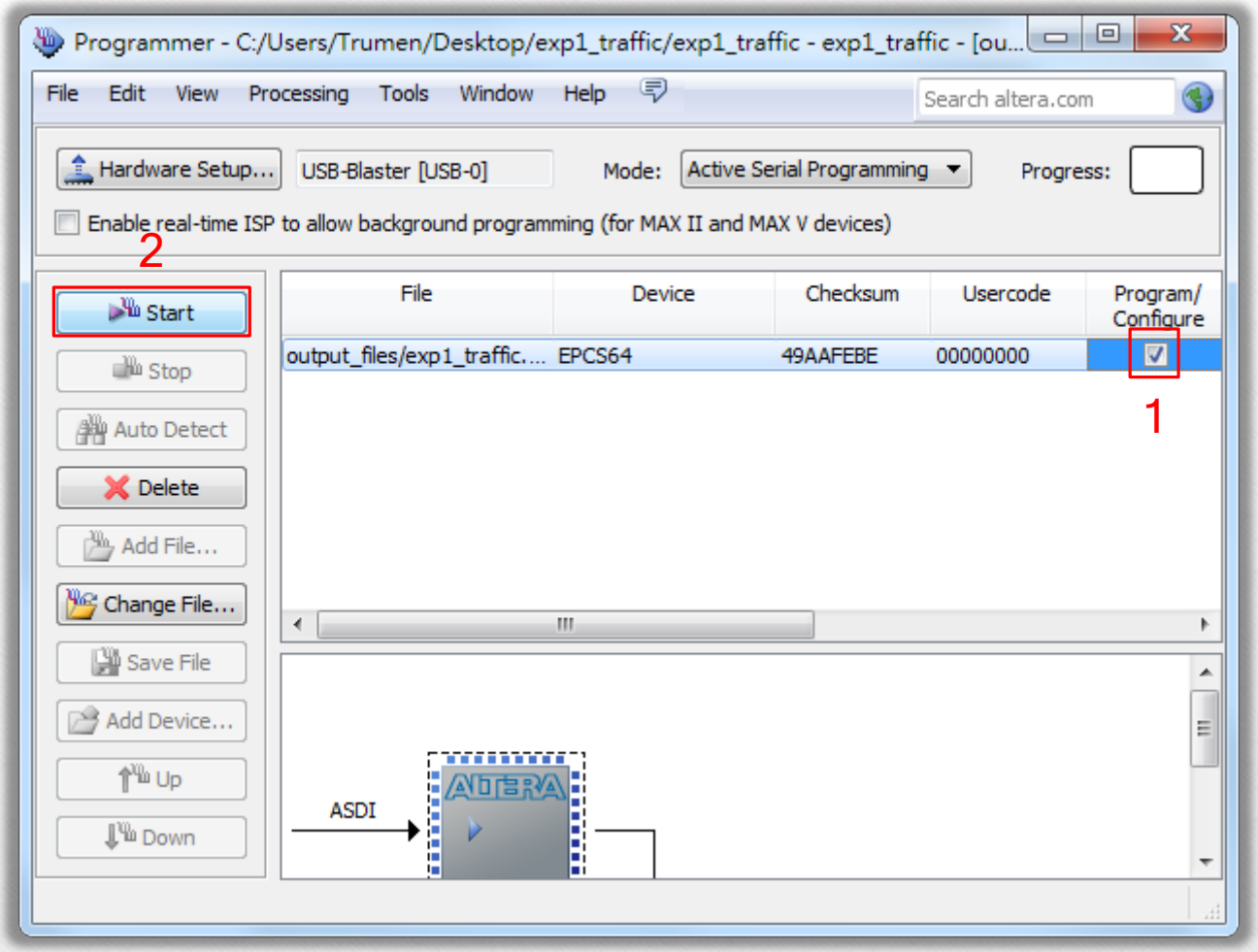
1. Ensure that power is applied to the DE2-115 board.
2. Connect the supplied USB cable to the USB Blaster port on the DE2-115 board.
3. Configure the JTAG programming circuit by setting the RUN/PROG slide switch (SW19) to the PROG position.
4. The EPCS64 chip can now be programmed by using the Quartus II Programmer to select a configuration bit stream file with the .pof filename extension.
5. Once the programming operation is finished, set the RUN/PROG slide switch back to the RUN position and then reset the board by turning the power switch off and back on; this action causes the new configuration data in the EPCS64 device to be loaded into the FPGA chip.

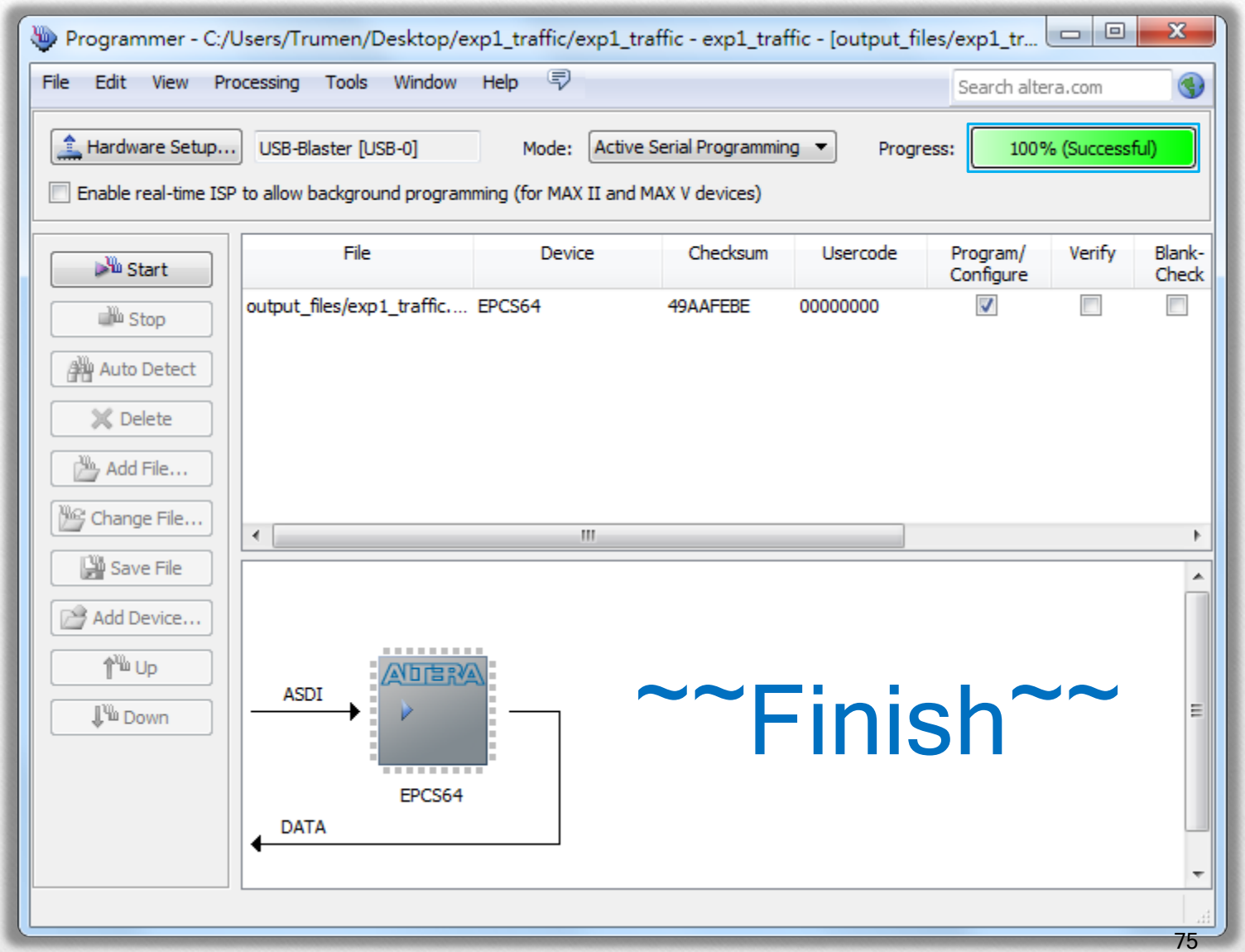
Programmer Object File

- Programmer Object File is a binary file (with the extension .pof) containing the data for programming a configuration device.
- A Programmer Object File for a configuration device can be generated by the Convert Programming Files command (File menu).


1
File
Edit
View
Project
Assignments
New... Ctrl+N
Open... Ctrl+O
Close Ctrl+F4
New Project Wizard...
Open Project... Ctrl+J
Save Project
Close Project
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
File Properties...
Create / Update
Export...
2
Convert Programming Files...
Page Setup...
Print Preview
Print... Ctrl+P
Recent Files
Recent Projects
Exit Alt+F4
System
Processing
Combines and/or converts programming files into
Convert Programming File - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - ex...
File
Tools
Window
Search altera.com
Specify the input files to convert and the type of programming file to generate.
You can also import input file information from other files and save the conversion setup information created here for future use.
Conversion setup files
Open Conversion Setup Data...
Save Conversion Setup...
Output programming file
Programming file type:
Programmer Object File (.pof)
3
for DE2-115
Options...
Configuration device:
EPCS64
File name:
output_files/exp1_traffic.pof
4
Advanced...
Remote/Local update difference file:
Create Memory Map File (Generate
Create CvP files (Generate exp1_t
Input files to convert
File/Data area
Prop
5
SOF Data
Page_0
<auto>
Add Sof Page
6
Add File...
Remove
Up
Down
Properties
9
Generate
Close
Help
Select Input File
Look in:
C...s
My Computer
Trumen
exp1_traffic.sof
7
File name:
exp1_traffic.sof
8
Open
Files of type:
SRAM Object Files (*.sof
Cancel


Quartus II 64-Bit - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - exp1_traffic
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
1
exp1_traffic
Programmer
Project Navigator
Cyclone IV E: EP4CE
exp1_traffic
Hierarchy
Tasks
Flow: Compilation
Compile
Programmer - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - exp1_traffic - [ou...
File Edit View Processing Tools Window Help
Search altera.com
2
Hardware Setup...
USB-Blaster [USB-0]
Mode: Active Serial Programming
Progress:
Enable real-time ISP to allow background programming (for MAX II and MAX V devices)
Start
Stop
Auto Detect
3
Delete
Add File...
Change File...
Save File
Add Device...
Up
Down
File
Device
Checksum
Usercode
Program/ Configure
Select Programming File
Look in: C:\...les
My Computer
Trumen
exp1_traffic.pof
4
File name: exp1_traffic.pof
Open
5
Files of type: POF Files (*.pof)
Cancel
Flow Summary
Flow Status
Quartus II 6
Revision Nan
Top-level En
Family
Device
Timing Model
Total logic el
Total cor
Dedicate
Total registe
Total pins
All
Type
ID
Messages
System
Processing
Opens a Programmer window
100%
00:00:42


Programmer - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - exp1_traffic - [ou...
File
Edit
View
Processing
Tools
Window
Help
Search altera.com
Hardware Setup...
USB-Blaster [USB-0]
Mode:
Active Serial Programming
Progress:
Enable real-time ISP to allow background programming (for MAX II and MAX V devices)
2
Start
Stop
Auto Detect
Delete
Add File...
Change File...
Save File
Add Device...
Up
Down
File
Device
Checksum
Usercode
Program/ Configure
output_files/exp1_traffic....
EPCS64
49AAFEBE
00000000
1
ASDI
ALTERA


Programmer - C:/Users/Trumen/Desktop/exp1_traffic/exp1_traffic - exp1_traffic - [output_files/exp1_tr...
File Edit View Processing Tools Window Help
Search altera.com
Hardware Setup...
USB-Blaster [USB-0]
Mode: Active Serial Programming
Progress: 100% (Successful)
Enable real-time ISP to allow background programming (for MAX II and MAX V devices)
Start
Stop
Auto Detect
Delete
Add File...
Change File...
Save File
Add Device...
Up
Down
File
Device
Checksum
Usercode
Program/ Configure
Verify
Blank-Check
output_files/exp1_traffic.... EPCS64 49AAFEBE 00000000
ASDI
EPCS64
DATA
~~Finish~~

The End.

Any question?

Reference

1. http://en.wikipedia.org/wiki/Field-programmable_gate_array
2. "My First FPGA for Altera DE2-115 Board" by Terasic Technologies Inc.
3. "DE2-115 User Manual" by Terasic Technologies Inc.