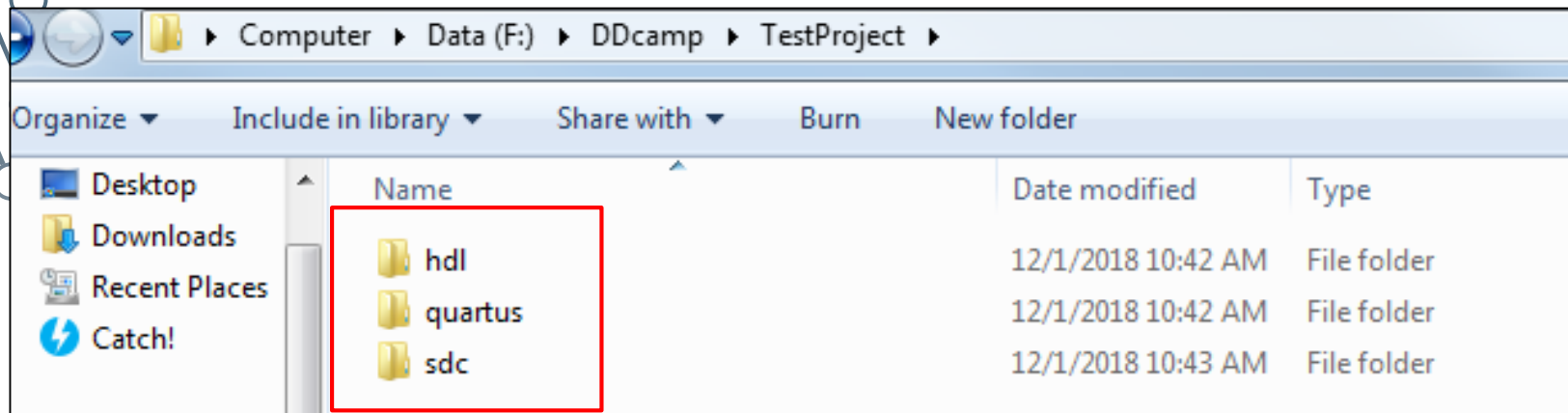


DIGITAL DESIGN WITH FPGA CAMP

QUARTUSII PROJECT SETUP

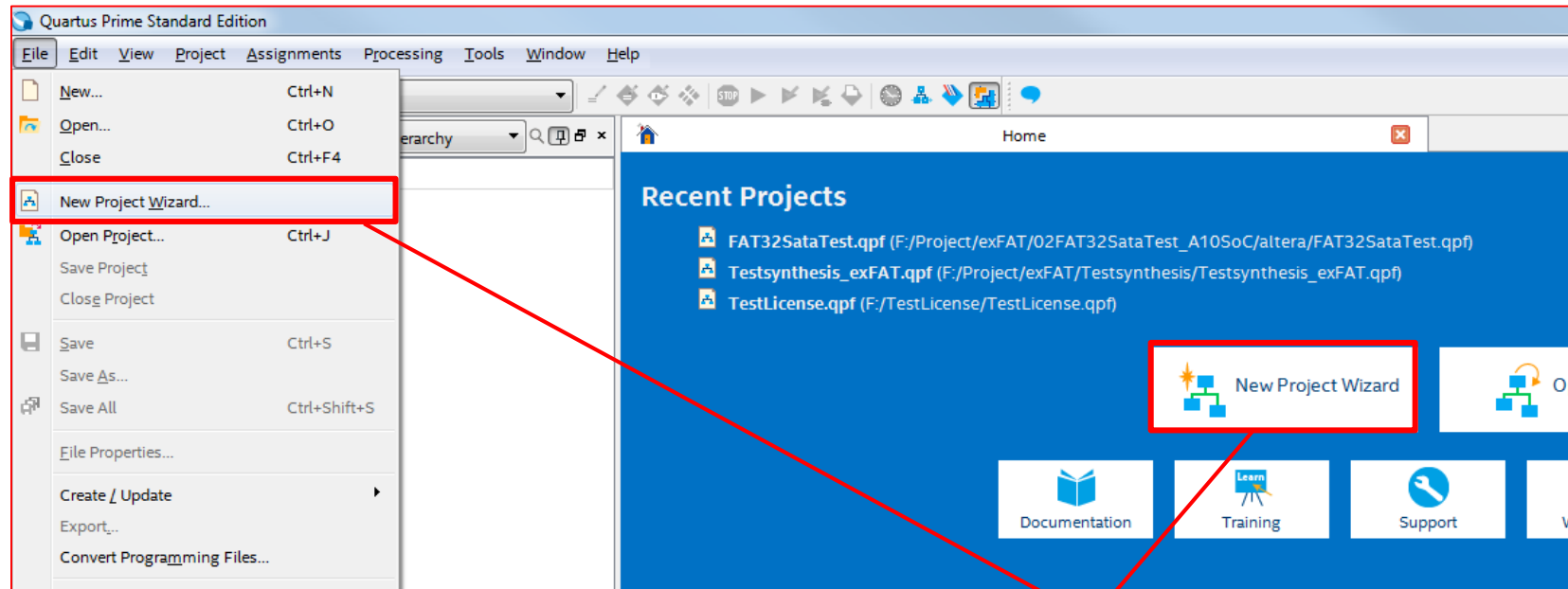
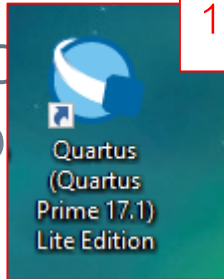
STEP1 : CREATE FOLDER



- การเก็บ Project แต่ละตัว ควรแยกกันเก็บอย่างเด็ดขาด ไม่เก็บรวมกัน
- ตัวอย่าง จะมีโปรเจกใหม่ชื่อ TestProject ภายในจะสร้าง folder เปล่าไว้ทั้งหมด 3 folder คือ hdl quartus และ sdc
- hdl folder ใช้เก็บ HDC code ที่เราออกแบบไว้
- quartus เป็น working directory ของ quartusII
- sdc ใช้เก็บ sdc file ซึ่งเป็นไฟล์ที่ระบุเรื่อง timing constraint ของ design เรา เช่น บอกว่าวงจรที่เราออกแบบจะทำงานที่ clock ความถี่เท่าไร

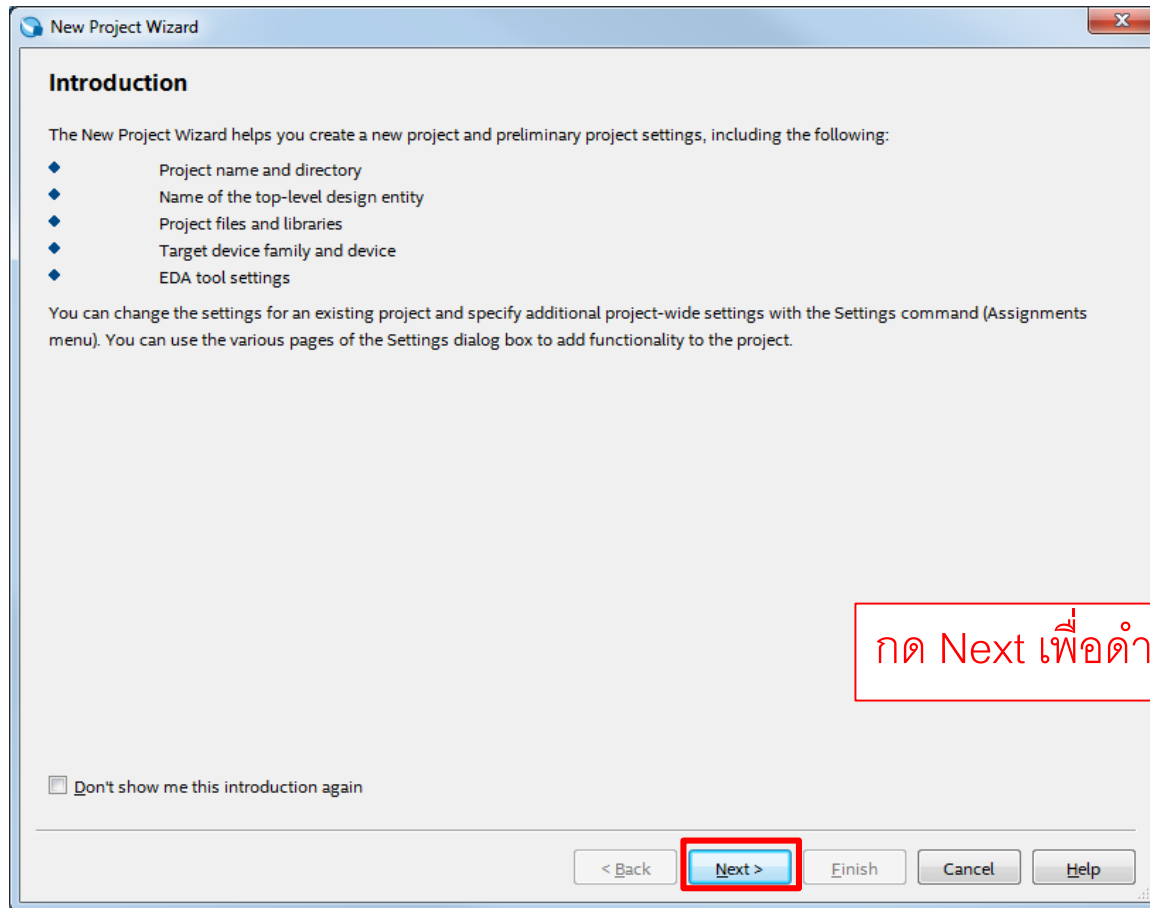
STEP2.1 : CREATE QUARTUSII PROJECT

1. กดเปิดโปรแกรม QuartusII



2. สร้าง QuartusII Project โดยกดเลือกที่รูป New Project Wizard หรือกดที่เมนู File -> New Project Wizard

STEP2.2 : CREATE QUARTUSII PROJECT



กด Next เพื่อดำเนินการขั้นตอนต่อไป

STEP2.3 : CREATE QUARTUSII PROJECT

Directory, Name, Top-Level Entity

What is the working directory for this project?

F:/DDcamp/TestProject/quartus

What is the name of this project?

TestProject

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.

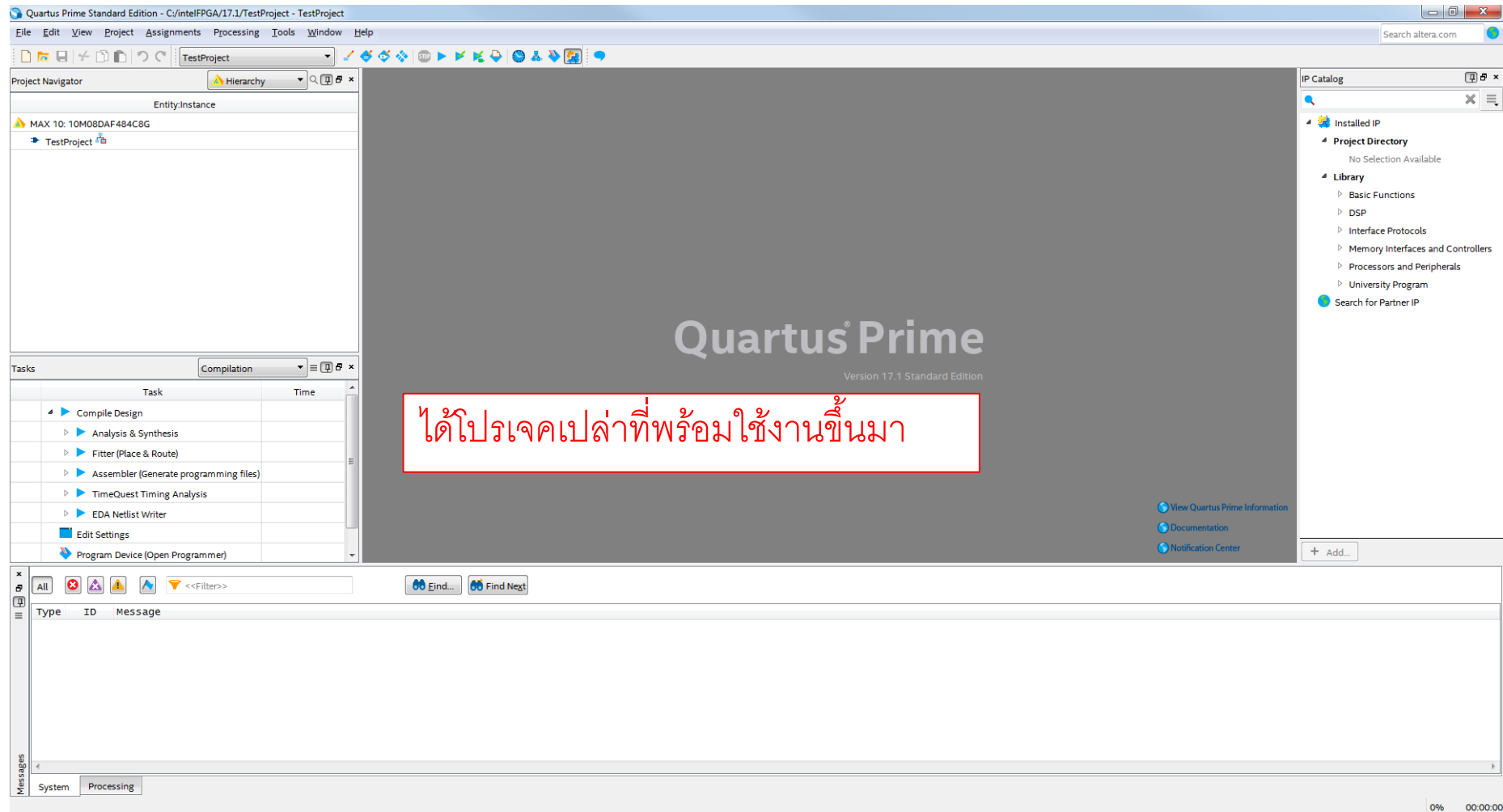
TestProject

Use Existing Project Settings...

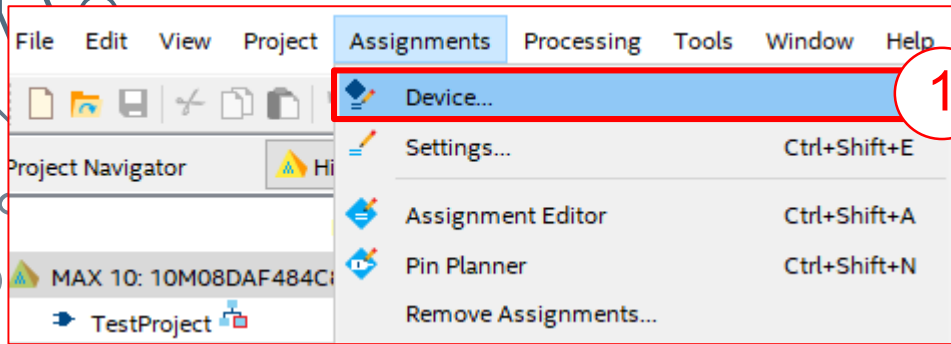
< Back Next > Finish Cancel Help

1. เลือก working directory ไปยัง quartus folder ที่เราสร้างไว้
2. ตั้งชื่อโปรเจกต์และชื่อ Entity ของ HDL ที่เป็น Top module ของระบบเรา (ถ้ามี Entity เดียวทั้งระบบ ให้ระบุชื่อ Entity เดียวของเรานั้นลงไป)
3. กด Finish เพื่อจบการสร้าง Project

STEP2.4 : CREATE QUARTUSII PROJECT

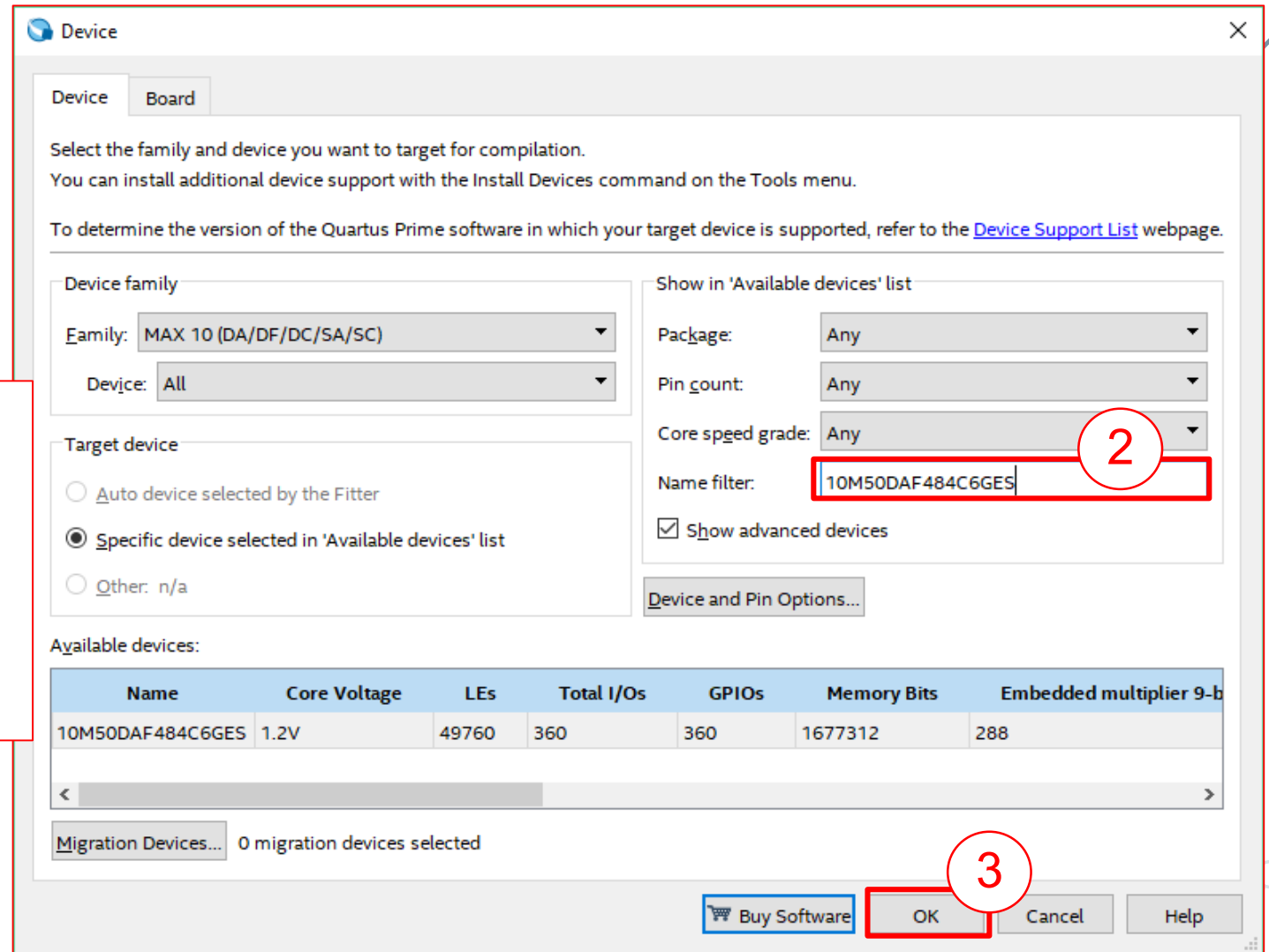


STEP3 : DEVICE SETTING

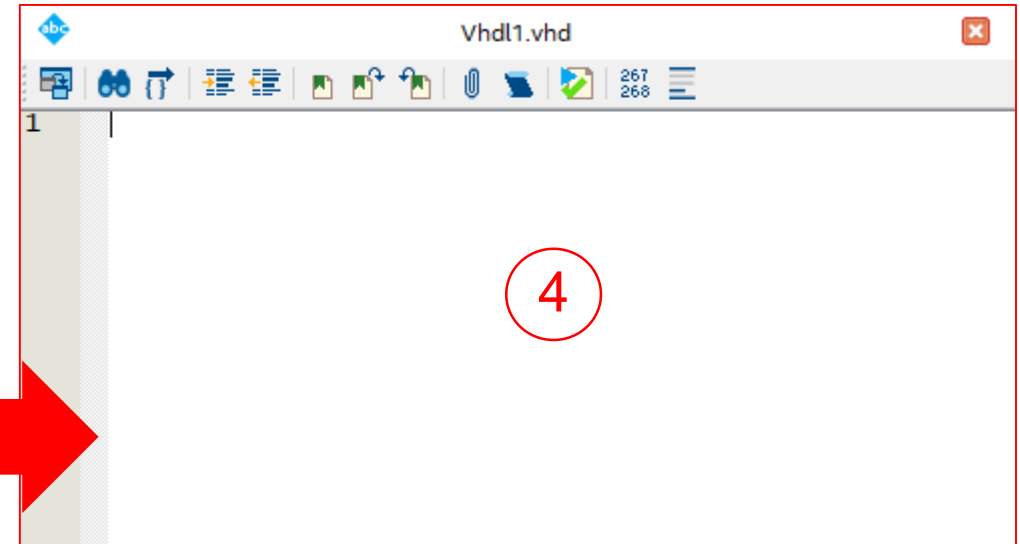
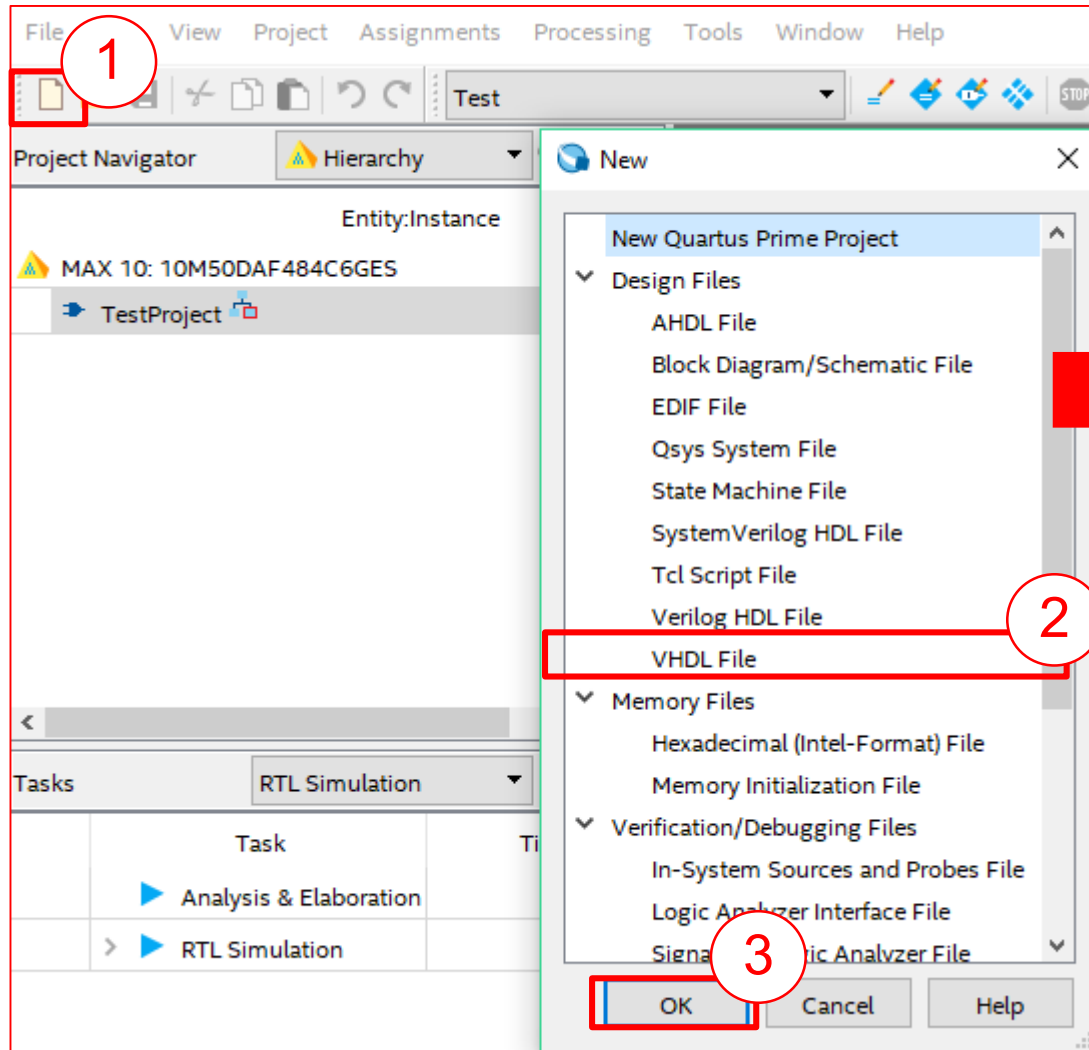


แก้ไข Device ให้ตรงกับบอร์ดเรา โดยมีขั้นตอนดังนี้

1. เลือก Assignments -> Device
2. ที่ช่อง Name filter พิมพ์ชื่อ Device บนบอร์ดเราลงไป คือ 10M50DAF484C6GES
3. กดปุ่ม OK เพื่อบันทึกการกำหนด Device



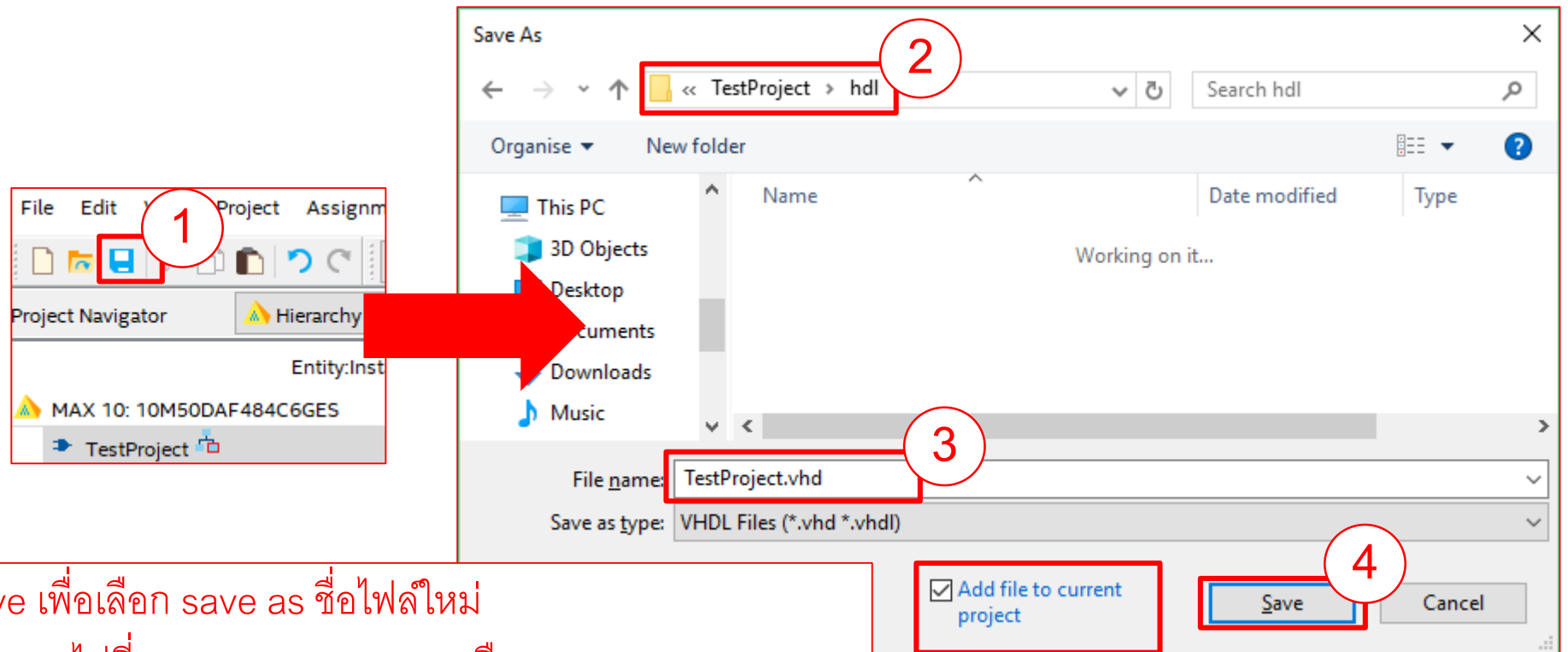
STEP4.1 : CREATE NEW VHDL FILE



กตสร้าง VHDL File ใน Quartus project

1. กดที่ icon รูปไฟล์เปล่าที่มุมบนซ้ายของ Tool เพื่อเลือกสร้างไฟล์ใหม่
2. เลือกชนิดไฟล์เป็น VHDL File
3. กดปุ่ม OK เพื่อจบการสร้างไฟล์
4. บนหน้าต่างของ Tool จะปรากฏไฟล์เปล่าขึ้นมา ซึ่งสามารถเขียน code ลงไปเพิ่มเติมได้

STEP4.2 : RENAME HDL CODE



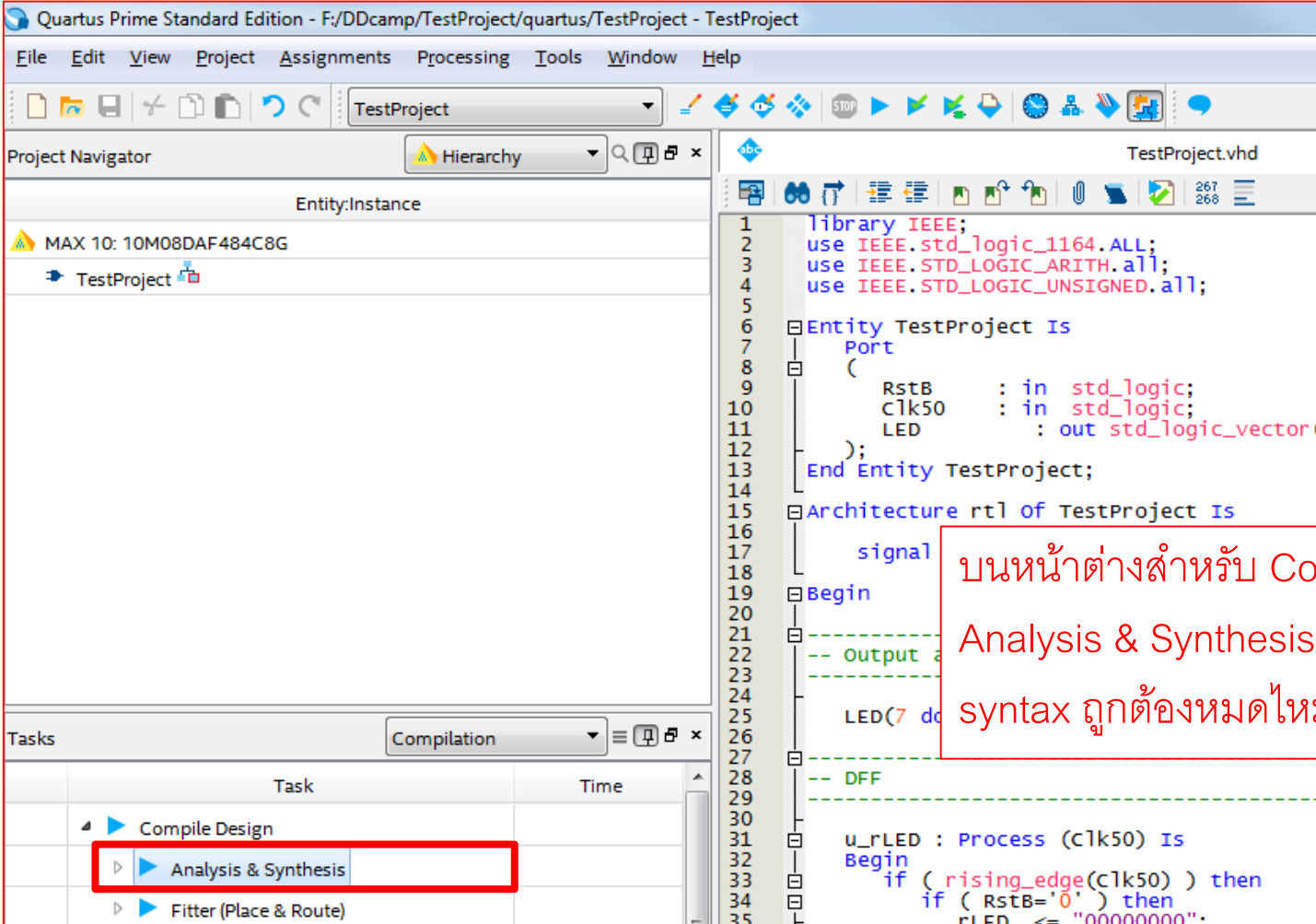
1. กดที่ icon รูป save เพื่อเลือก save as ชื่อไฟล์ใหม่
2. เลือกตำแหน่งของ file ไปที่ project folder ของเราคือ ../TestProject/hdl
3. ตั้งชื่อไฟล์ใหม่เป็น TestProject.vhd
4. กด save ชื่อไฟล์ใหม่ และไฟล์นี้จะถูก add เข้า Project เป็น default setting

STEP4.3 : TYPE HDL CODE TO NEW VHDL FILE

```
1  library IEEE;
2  use IEEE.std_logic_1164.ALL;
3  use IEEE.STD_LOGIC_ARITH.all;
4  use IEEE.STD_LOGIC_UNSIGNED.all;
5
6  Entity TestProject Is
7  Port
8  (
9      RstB      : in std_logic;           -- use push
10     Clk50     : in std_logic;          -- clock in
11     LED       : out std_logic_vector( 7 downto 0 ) -- ad
12 );
13 End Entity TestProject;
14
15 Architecture rtl of BlinkLED Is
16 -- signal declaration
17
18
19
20     signal rLED      : std_logic_vector( 7 downto 0 );
21
22 Begin
23
24     LED(7 downto 0)      <= rLED(7 downto 0);
25
26     u_rLED : Process (Clk50) Is
27     Begin
28         if ( rising_edge(Clk50) ) then
29             if ( RstB='0' ) then
30                 rLED(7 downto 0) <= "11111111";
31             else
32                 rLED(7 downto 0) <= "00000000";
33             end if;
34         end if;
35     End Process u_rLED;
36 End Architecture rtl;
```

พิมพ์ HDL Code ที่เราต้องการออกแบบลงไป ใน HDL File

STEP5.1: ANALYSIS & SYNTHESIS



The screenshot displays the Quartus Prime Standard Edition interface. The top menu bar includes File, Edit, View, Project, Assignments, Processing, Tools, Window, and Help. The Project Navigator on the left shows the hierarchy of the TestProject, with the Entity:Instance section expanded to show MAX 10: 10M08DAF484C8G and TestProject. The main editor window shows the TestProject.vhd file, which contains VHDL code for an entity TestProject and its architecture rtl. The Tasks pane at the bottom left shows a list of tasks: Compile Design, Analysis & Synthesis (highlighted with a red box), and Fitter (Place & Route). The Analysis & Synthesis task is currently selected.

```
1  library IEEE;
2  use IEEE.std_logic_1164.ALL;
3  use IEEE.STD_LOGIC_ARITH.all;
4  use IEEE.STD_LOGIC_UNSIGNED.all;
5
6  Entity TestProject Is
7  Port
8  (
9      RstB      : in std_logic;
10     Clk50     : in std_logic;
11     LED       : out std_logic_vector(
12         );
13 End Entity TestProject;
14
15 Architecture rtl of TestProject Is
16
17     signal
18
19 Begin
20
21
22     -- output a
23
24     LED(7 d
25
26
27
28     -- DFF
29
30
31     u_rLED : Process (Clk50) Is
32     Begin
33         if ( rising_edge(Clk50) ) then
34             if ( RstB='0' ) then
35                 LED <= "00000000";
```

บนหน้าต่างสำหรับ Compile Project มีเมนู Compile Design -> Analysis & Synthesis เพื่อลอง Compile HDL code ของเราว่ามี syntax ถูกต้องหมดไหม และสามารถสังเคราะห์เป็นวงจรได้หรือไม่

STEP5.2 : CHECK RESOURCE AND WARNING MESSAGE

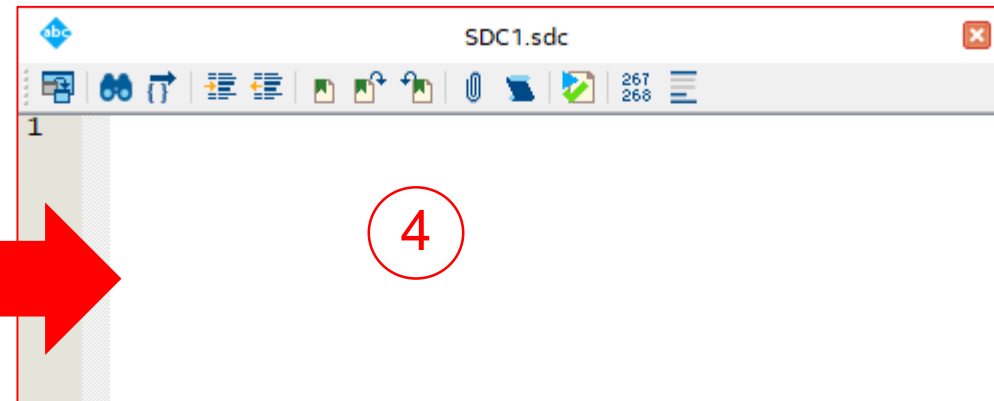
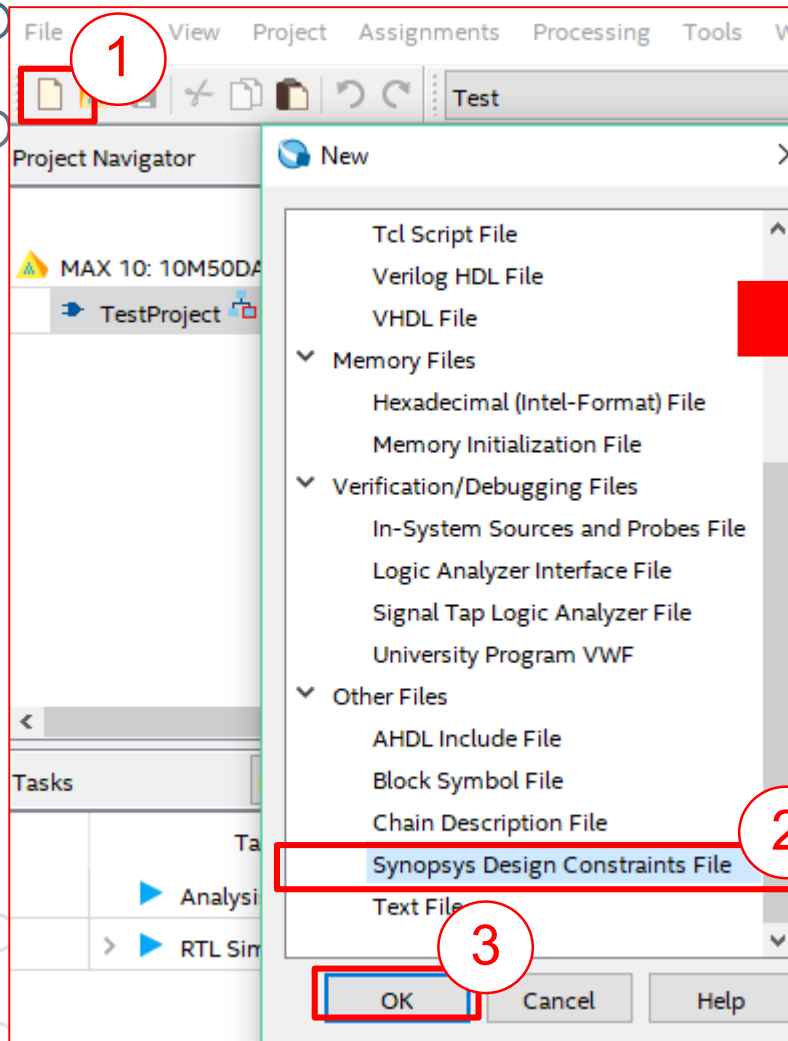
The screenshot displays the Quartus Prime Standard Edition interface. The 'Tasks' window (1) shows the 'Analysis & Synthesis' task completed successfully. The 'Flow Summary' window (2) shows the compilation status as 'Successful - Sat Dec 01 11:00:38 2018'. The 'Messages' window (3) shows a warning message: '18236 Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM_PARALLEL_PROCESSORS in your QSF to 20030 Parallel compilation is enabled and will use 4 of the 4 processors detected'.

| Type | ID | Message |
|---------|--------|--|
| Warning | 18236 | Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM_PARALLEL_PROCESSORS in your QSF to 20030 Parallel compilation is enabled and will use 4 of the 4 processors detected |
| Info | 12021 | Found 2 design units, including 1 entities, in source file /ddcamp/testproject/hdl/testproject.vhd |
| Info | 12127 | Elaborating entity "TestProject" for the top level hierarchy |
| Info | 286030 | Timing-Driven Synthesis is running |
| Info | 16010 | Generating hard_block partition "hard_block:auto_generated_inst" |
| Info | 21057 | Implemented 18 device resources after synthesis - the final resource count might be different |
| Info | | Quartus Prime Analysis & Synthesis was successful. 0 errors, 1 warning |

ตรวจสอบวงจร

1. ถ้า code สามารถ compile ได้จนเสร็จ จะขึ้นเครื่องหมายถูกสีเขียวขึ้น
2. ตรวจสอบ resource ที่ใช้งานบนหน้าจอ ถ้า code ถูกต้องหมด ควรจะมีการสร้าง logic โดยใช้ Logic Elements ไปบ้าง (หากลืม assign output ตัว Logic Elements จะมีค่าเป็น 0)
- 3.ให้อ่าน warning message ให้ครบถ้วน (ตัวหนังสือสีฟ้า) ว่ามีอะไรที่ผิดปกติหรือไม่ ถ้ามีให้กลับไปแก้ไขให้เรียบร้อยก่อน

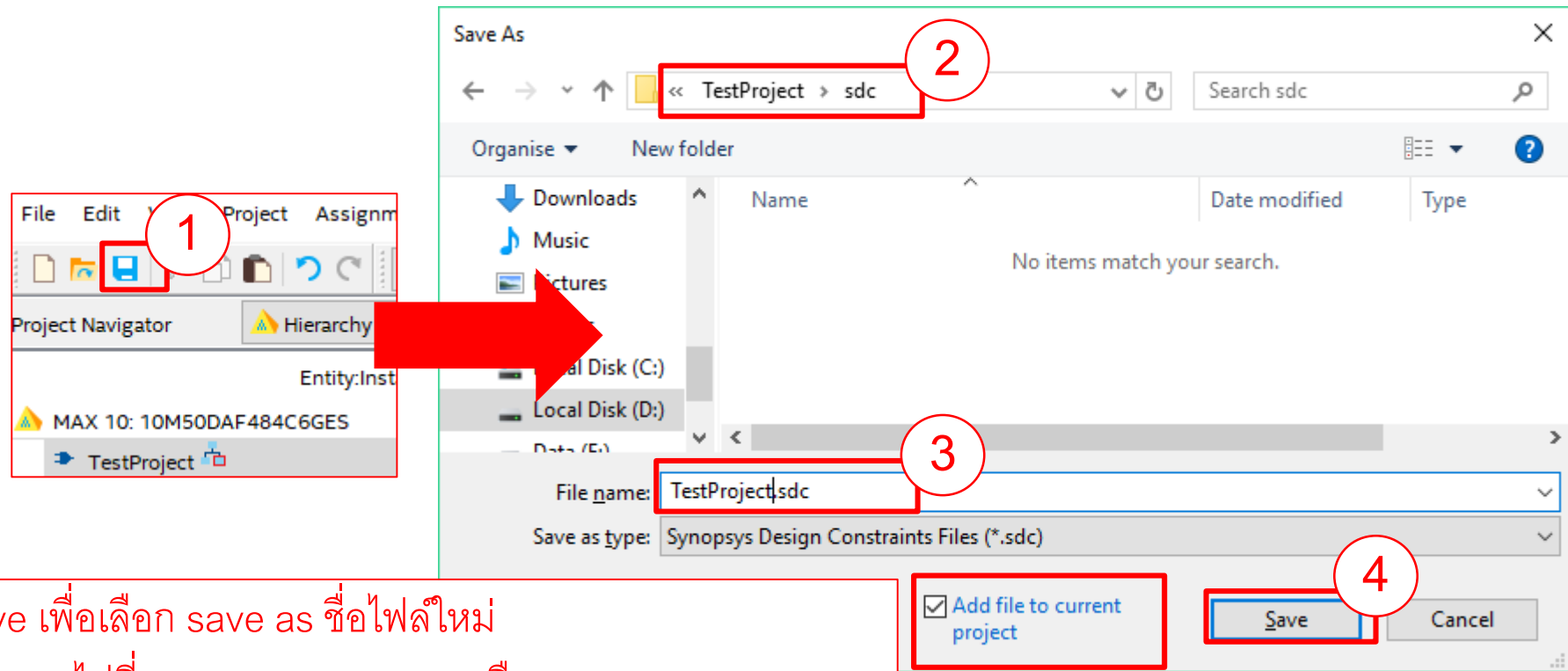
STEP6.1 : CREATE NEW SDC FILE (TIMING CONSTRAINT FILE)



กตสร้าง SDC File ใน Quartus project

1. กดที่ icon รูปไฟล์เปล่าที่มุมบนซ้ายของ Tool เพื่อเลือกสร้างไฟล์ใหม่
2. เลือกชนิดไฟล์เป็น Synopsys Design Constraints File
3. กดปุ่ม OK เพื่อจบการสร้างไฟล์
4. บนหน้าต่างของ Tool จะปรากฏไฟล์เปล่าขึ้นมา ซึ่งสามารถเขียนคำสั่งลงไปเพิ่มเติมได้

STEP6.2 : RENAME SDC FILE



1. กดที่ icon รูป save เพื่อเลือก save as ชื่อไฟล์ใหม่
2. เลือกตำแหน่งของ file ไปที่ project folder ของเราคือ ../TestProject/sdc
3. ตั้งชื่อไฟล์ใหม่เป็น TestProject.sdc
4. กด save ชื่อไฟล์ใหม่ และไฟล์นี้จะถูก add เข้า Project เป็น default setting

STEP6.3 : TYPE COMMAND TO NEW SDC FILE

พิมพ์คำสั่งลงใน SDC File

```
1
2 *****
3 # Create Clock
4 *****
5
6 create_clock -period 20 -name Clk50 [get_ports {Clk50}]
7
8 *****
9 # Create Generated Clock
10 *****
11 derive_pll_clocks
12
13 *****
14 # Set Clock Uncertainty
15 *****
16 derive_clock_uncertainty
```

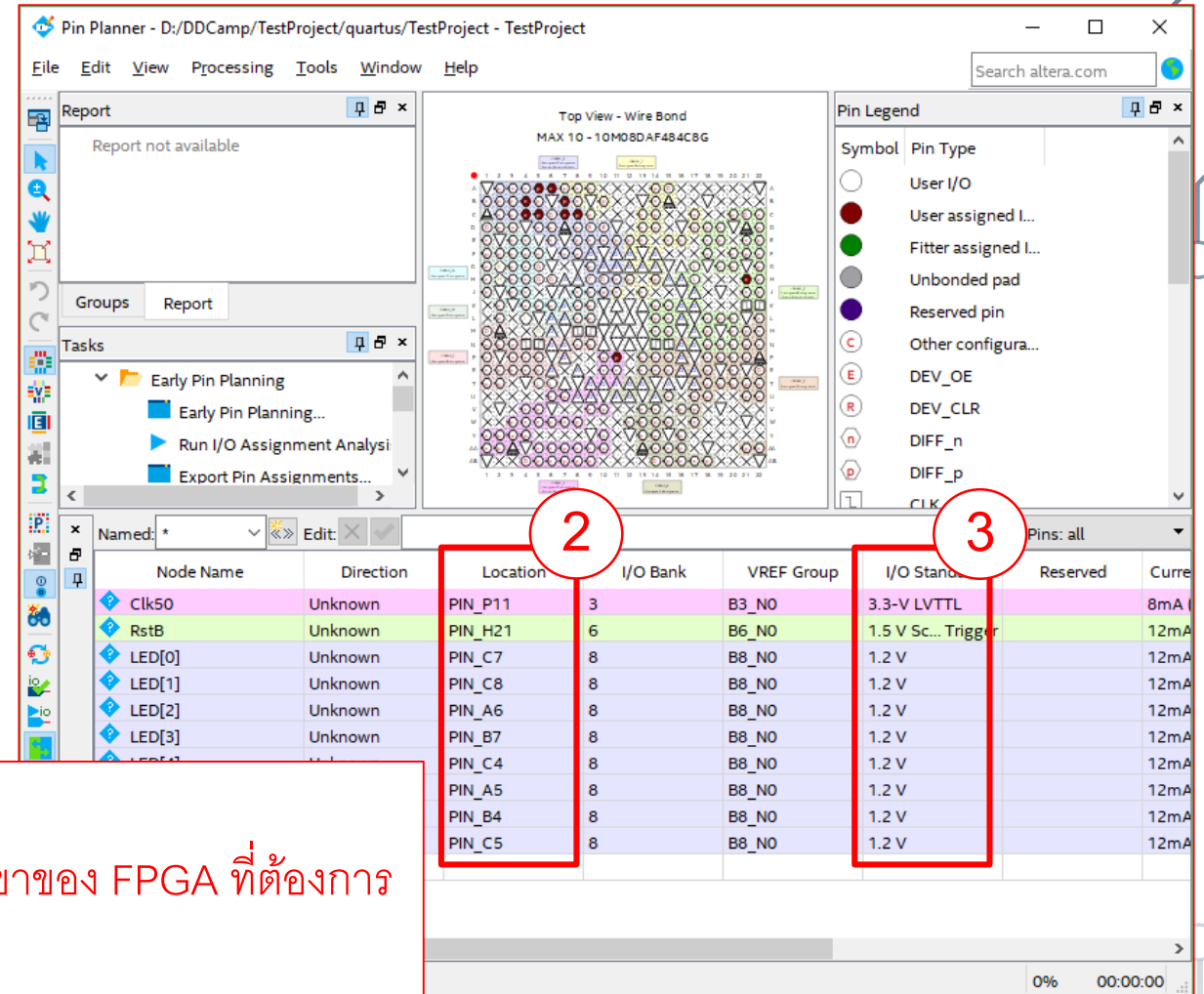
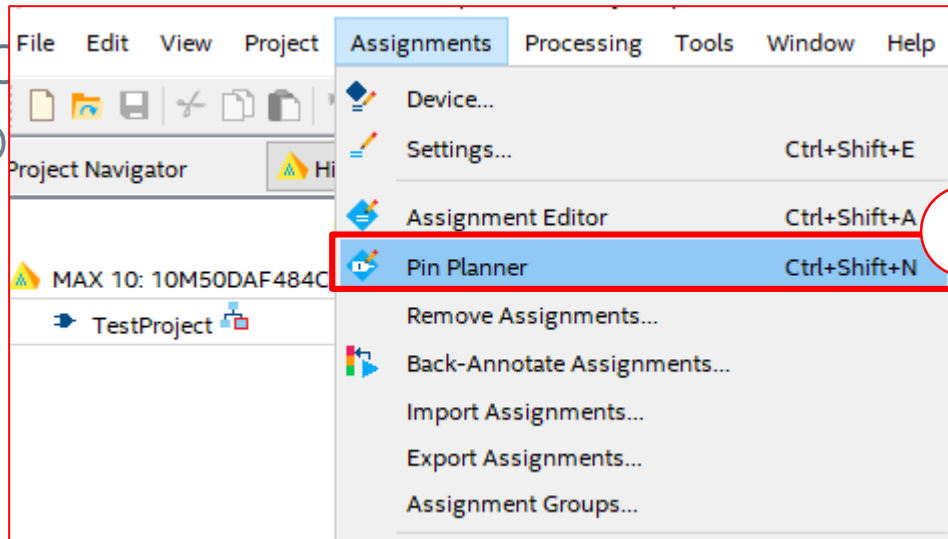
1

1. ระบุความถี่ของสัญญาณนาฬิกาที่ใช้ใน design ของเรา โดยกำหนดเป็นคาบเวลาหน่วย ns ในที่นี้ $20 \text{ ns} = 50 \text{ MHz}$

2

2. เป็นคำสั่งเพิ่มเติม เพื่อให้ tool นั้นสร้างและตรวจสอบ clock เพิ่มเติม เมื่อใน code ของเรามีการเรียกใช้ PLL เพื่อสร้าง clock ความถี่อื่น ๆ เพิ่มเติม

STEP 7.1 : SET PIN ASSIGNMENT

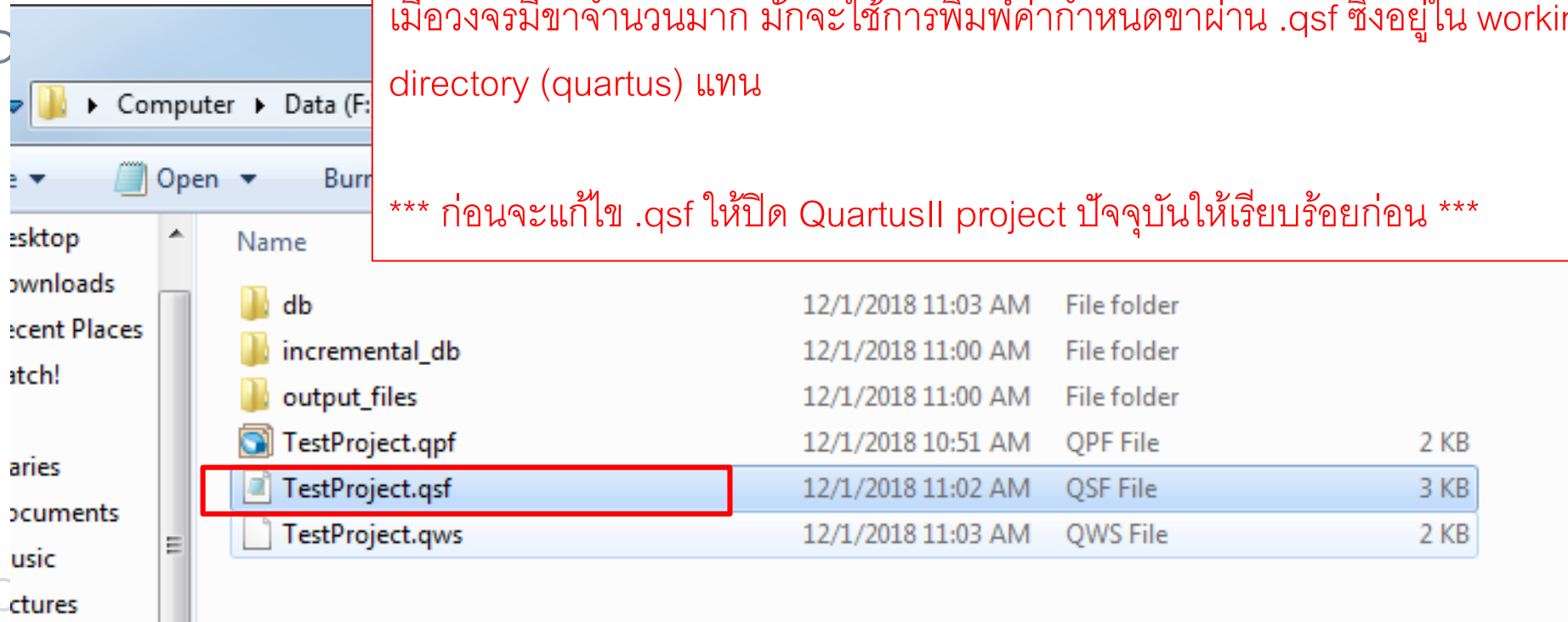


1. กดที่เมนู Assignments -> Pin Planner
2. หน้าต่าง Pin Planner จะปรากฏขึ้น ให้กำหนดหมายเลขขาของ FPGA ที่ต้องการ จะใช้งานสำหรับแต่ละสัญญาณ
3. กำหนดค่า I/O Standard ว่าแต่ละขาที่เราใช้งานนั้น ต่อกับระดับไฟเท่าไรบ้าง
4. กด save เมื่อกำหนดค่าทั้งหมดเสร็จ

STEP7.1 : SET PIN ASSIGNMENT (OPTIONAL)

สำหรับวงจรใหญ่ ๆ การใช้ GUI เพื่อ set ค่าที่ละขานี้จะทำให้เสียเวลา
เมื่อวงจรมีขาจำนวนมาก มักจะใช้การพิมพ์ค่ากำหนดขาผ่าน .qsf ซึ่งอยู่ใน working
directory (quartus) แทน

*** ก่อนจะแก้ไข .qsf ให้ปิด QuartusII project ปัจจุบันให้เรียบร้อยก่อน ***



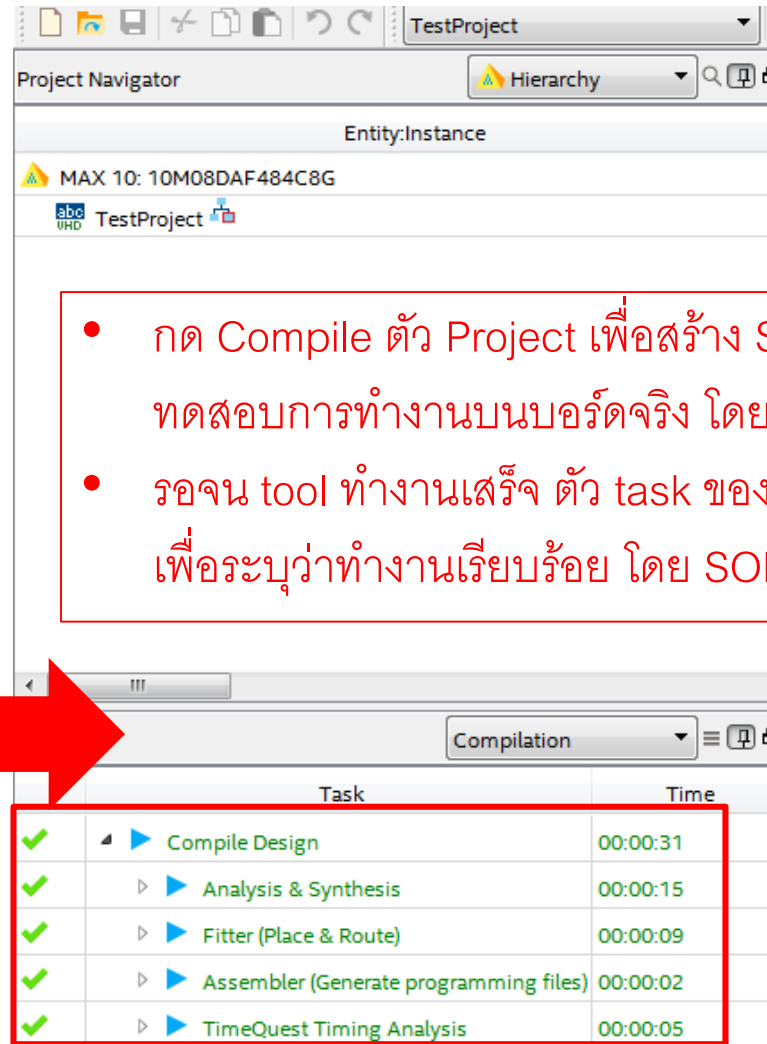
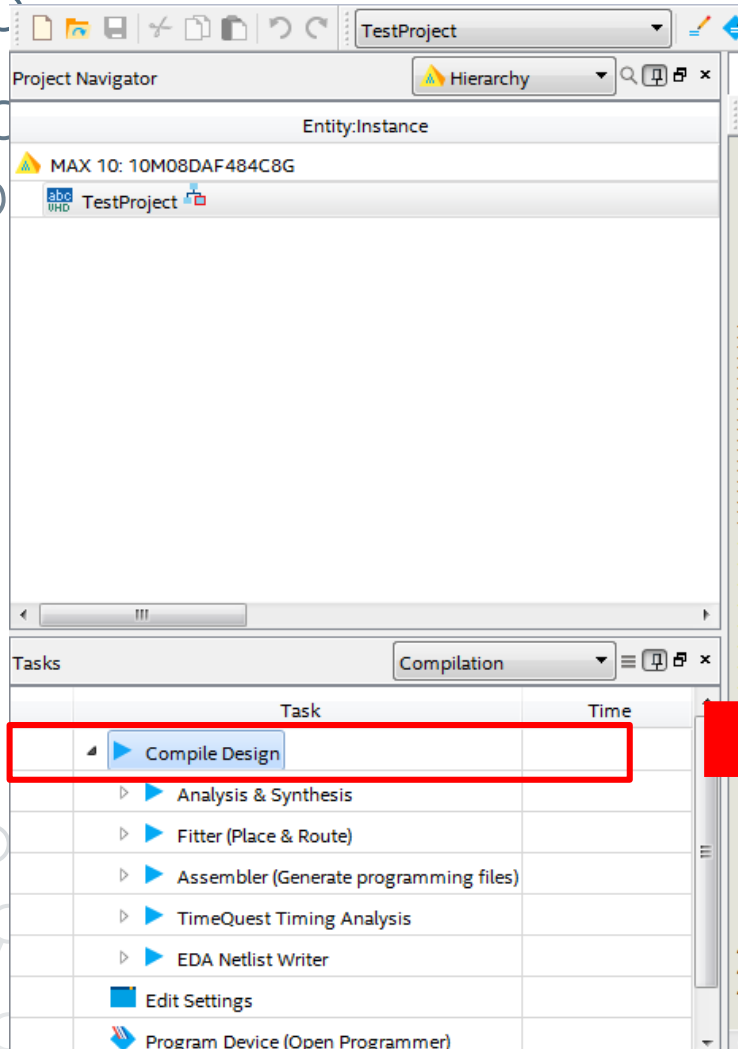
STEP7.1 : SET PIN ASSIGNMENT (OPTIONAL)

```
64
65 #=====
66 # CLOCK
67 #=====
68 set_location_assignment PIN_P11 -to Clk50
69 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to Clk50
70
71 #=====
72 # RESET
73 #=====
74 set_location_assignment PIN_H21 -to RstB
75 set_instance_assignment -name IO_STANDARD "1.5 V Schmitt Trigger" -to RstB
76
77 #=====
78 # LED
79 #=====
80 set_location_assignment PIN_C7 -to LED[0]
81 set_instance_assignment -name IO_STANDARD "1.2 V" -to LED[0]
82 set_location_assignment PIN_C8 -to LED[1]
83 set_instance_assignment -name IO_STANDARD "1.2 V" -to LED[1]
84 set_location_assignment PIN_A6 -to LED[2]
85 set_instance_assignment -name IO_STANDARD "1.2 V" -to LED[2]
86 set_location_assignment PIN_B7 -to LED[3]
87 set_instance_assignment -name IO_STANDARD "1.2 V" -to LED[3]
88 set_location_assignment PIN_C4 -to LED[4]
89 set_instance_assignment -name IO_STANDARD "1.2 V" -to LED[4]
90 set_location_assignment PIN_A5 -to LED[5]
91 set_instance_assignment -name IO_STANDARD "1.2 V" -to LED[5]
92 set_location_assignment PIN_B4 -to LED[6]
93 set_instance_assignment -name IO_STANDARD "1.2 V" -to LED[6]
94 set_location_assignment PIN_C5 -to LED[7]
95 set_instance_assignment -name IO_STANDARD "1.2 V" -to LED[7]
96
```

เปิดไฟล์ .qsf ด้วย text editor เช่น Notepad++
แล้วพิมพ์ข้อความเพื่อกำหนดขาและระดับไฟของขาให้กับแต่ละสัญญาณ

Note: การกำหนดค่าโดยใช้ text editor นั้นแม้จะเร็ว แต่ต้องอาศัยความ
ชำนาญของผู้ออกแบบอยู่ เพราะอาจจะผิดพลาดได้ง่าย
**แนะนำให้หลังจากพิมพ์ qsf เสร็จ ให้ลองกลับไปเปิด Pin Planner เพื่อ
ตรวจสอบว่าทุกขาได้กำหนดค่าเรียบร้อยแล้ว**

STEP8.1 : COMPILE PROJECT



- กด Compile ตัว Project เพื่อสร้าง SOF File สำหรับโปรแกรมลงบอร์ดเพื่อทดสอบการทำงานบนบอร์ดจริง โดยเลือกกดที่เมนู Compile Design
- รอจน tool ทำงานเสร็จ ตัว task ของ step ต่าง ๆ จะเปลี่ยนสถานะเป็นสีเขียว เพื่อระบุว่าทำงานเรียบร้อยแล้ว โดย SOF จะอยู่ใน output_files

STEP8.2 : COMPILATION REPORT

Quartus Prime Standard Edition - F:/DDcamp/TestProject/quartus/TestProject - TestProject

File Edit View Project Assignments Processing Tools Window Help

TestProject

Project Navigator Hierarchy

Entity:Instance

MAX 10: 10M50DAF484C6GES

TestProject

Table of Contents

Flow Summary

Flow Settings

Flow Non-Default Global Set

Flow Elapsed Time

Flow OS Summary

Flow Log

Analysis & Synthesis

Fitter

Flow Messages

Flow Suppressed Messages

Assembler

TimeQuest Timing Analyzer

Flow Summary

<<Filter>>

Flow Status Successful - Tue Dec 11 19:05:48 2018

Quartus Prime Version 17.1.0 Build 590 10/25/2017 SJ Standard Edition

Revision Name TestProject

Top-level Entity Name TestProject

Family MAX 10

Device 10M50DAF484C6GES

Timing Models Preliminary

Total logic elements 9 / 49,760 (< 1 %)

Total registers 8

Total pins 10 / 360 (3 %)

Total virtual pins 0

Total memory bits 0 / 1,677,312 (0 %)

Embedded Multiplier 9-bit elements 0 / 288 (0 %)

Total PLLs 0 / 4 (0 %)

UFM blocks 0 / 1 (0 %)

ADC blocks 0 / 2 (0 %)

Tasks

Compilation

Task

Compile Design

Analysis & Synthesis

Fitter (Place & Route)

Assembler (Generate program

TimeQuest Timing Analysis

EDA Netlist Writer

Edit Settings

Program Device (Open Programme

เมื่อ Compile เสร็จแล้ว เราสามารถดู Resource ที่ใช้ได้นบนหน้า Report

เมื่อ Compile เสร็จแล้ว จะได้ SOF File ซึ่งใช้โปรแกรมลงบอร์ด FPGA เพื่อทดสอบทำงานจริง (ไฟล์ SOF เป็นไฟล์ที่ใช้โปรแกรม FPGA โดยตรง ไม่ผ่าน Flash memory ดังนั้นจะใช้งานได้เฉพาะช่วงที่เปิดไฟอยู่ เมื่อปิดไฟบอร์ด โปรแกรมจะหายไปทันที)

STEP9.1 : OPEN PROGRAMMER

ทดสอบโปรแกรมบนบอร์ดจริง (เปิดไฟบอร์ด และเชื่อมต่อบอร์ดเข้ากับ PC ผ่าน USB Port ให้เรียบร้อยก่อนจะเริ่มขั้นตอนต่อไป)

1. กด Program Device เพื่อเปิด Programmer

2. หน้าต่าง Programmer จะแสดงขึ้นมาพร้อมทั้งเลือกไฟล์ SOF จากโปรเจกต์ปัจจุบันให้เรียบร้อย

STEP9.2 : JTAG SETUP

1. กด Hardware Setup

2. เลือก Device ที่ต้องการลงโปรแกรม (DECA board)

3. กด Close เพื่อ save setting

STEP9.3 : PROGRAM DEVICE

1. กดเลือก ✓ ที่ช่อง Program/Configure

2. กด Start เพื่อโปรแกรม FPGA

3. ตรวจสอบแถบ Progress จนกระทั่งโปรแกรมเรียบร้อยแล้ว (100%)

<https://www.facebook.com/DigitalDesignThailand/>



<https://forfpgadesign.wordpress.com/>

