

การทดลองที่ 0 การใช้งานโปรแกรม ISE Xilinx WebPACK

วัตถุประสงค์

1. เพื่อให้นักศึกษาติดตั้งและฝึกการใช้งาน FPGA

หมายเหตุ ให้อ่านเอกสารการทดลองและเอกสารประกอบให้ครบก่อนเริ่มทำการทดลอง

บทนำ

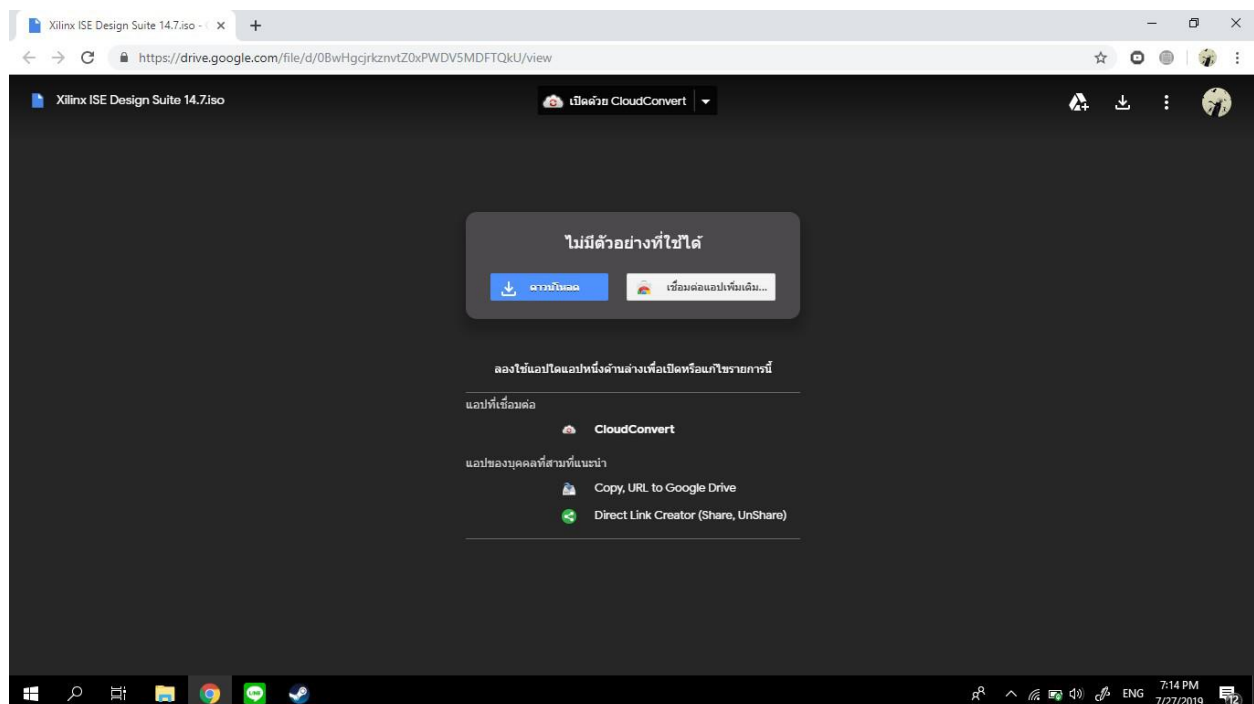
การทดลองนี้ นักศึกษาจะได้รู้จักการออกแบบวงจรด้วยการวาดรูป (Schematic Circuit) และบันทึกโปรแกรมลงในชิป Field Programmable Gate Array (FPGA)

โปรแกรม ISE Xilinx WebPACK

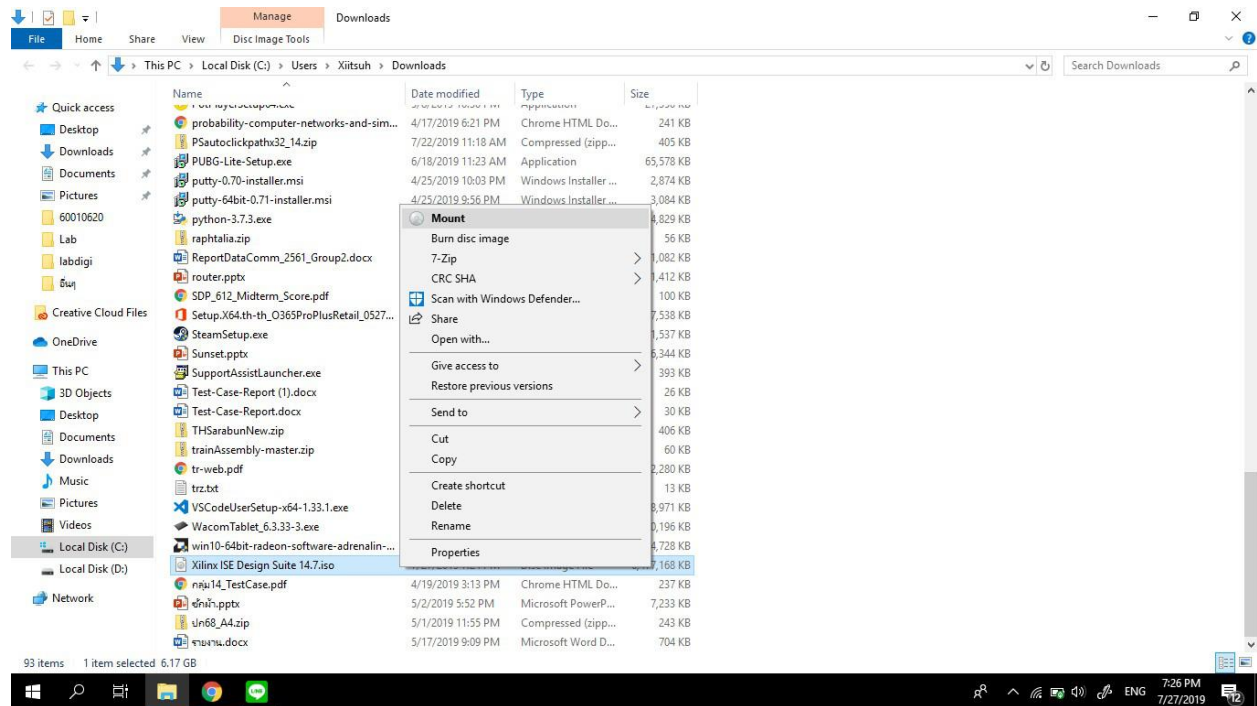
นักศึกษาที่มีหนังสือ “ออกแบบไอซีดิจิทัลด้วย FPGA และ CPLD ภาคปฏิบัติ โดยใช้วิธี Schematic ซอฟต์แวร์ทูล ISE WebPACK” ให้อ่านหนังสือก่อนการทดลองในบทที่ 1 และ 2 ก่อน

การติดตั้งโปรแกรม ISE Xilinx WebPACK

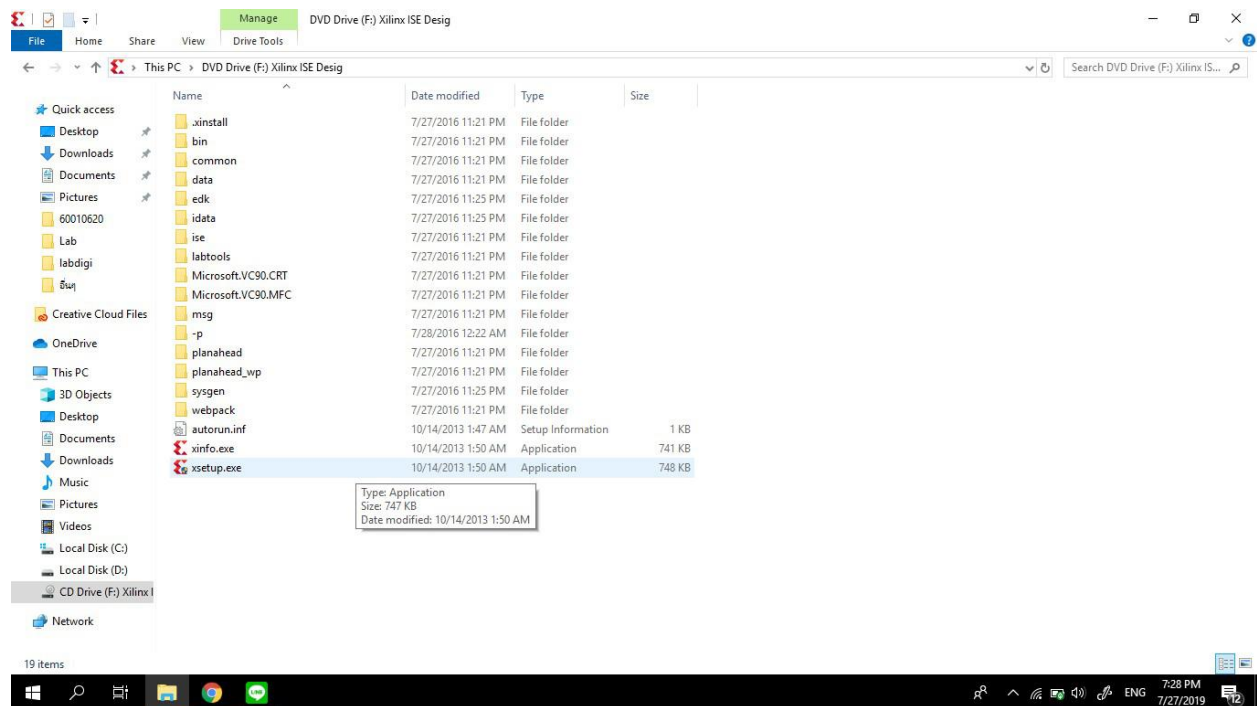
1. download โปรแกรมจาก <https://drive.google.com/file/d/0BwHgcjrkznvtZ0xPWDV5MDFTQkU/view>



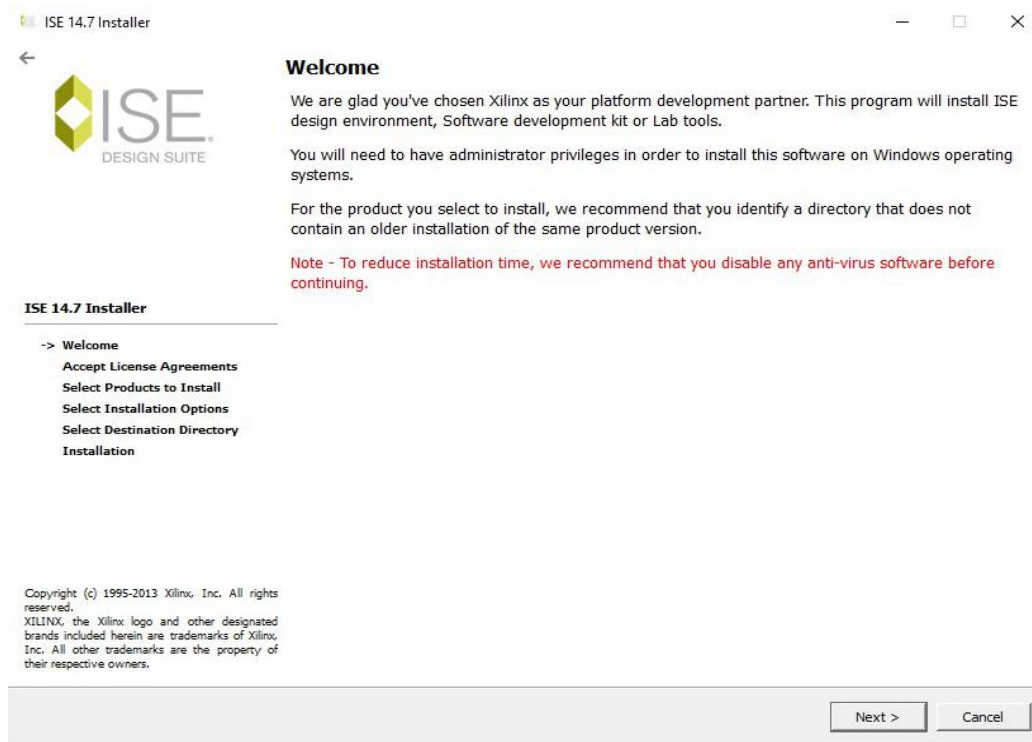
2.คลิกขวาที่ไฟล์ที่download จากนั้นกดคำว่า Mount



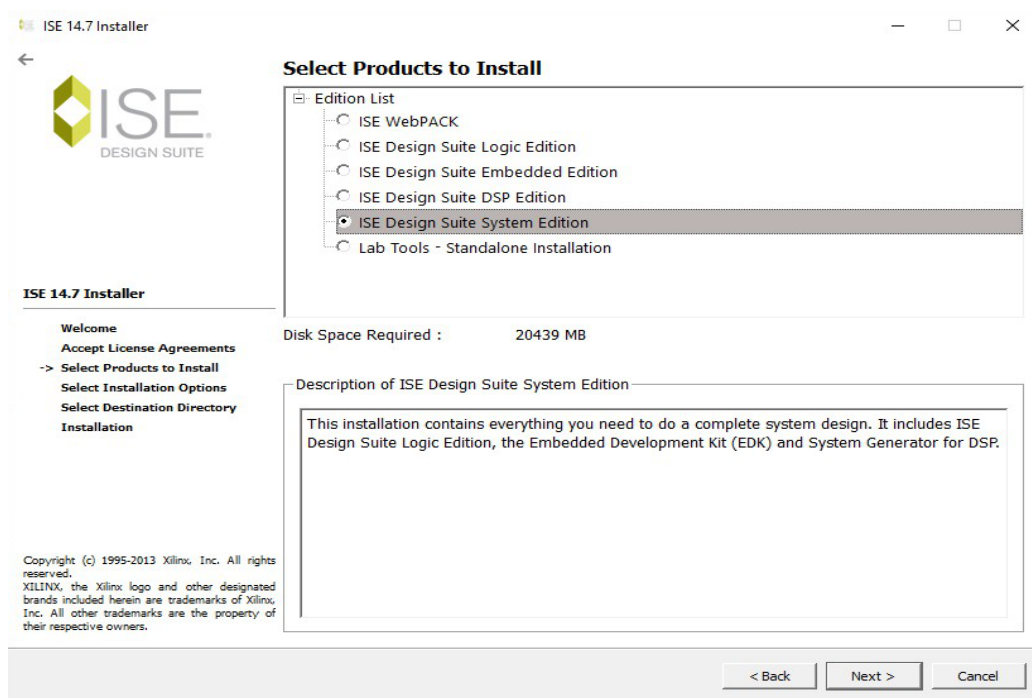
3.เมื่อคลิกMountเสร็จจะขึ้นหน้าขึ้นมาให้กดที่ xsetup.exe



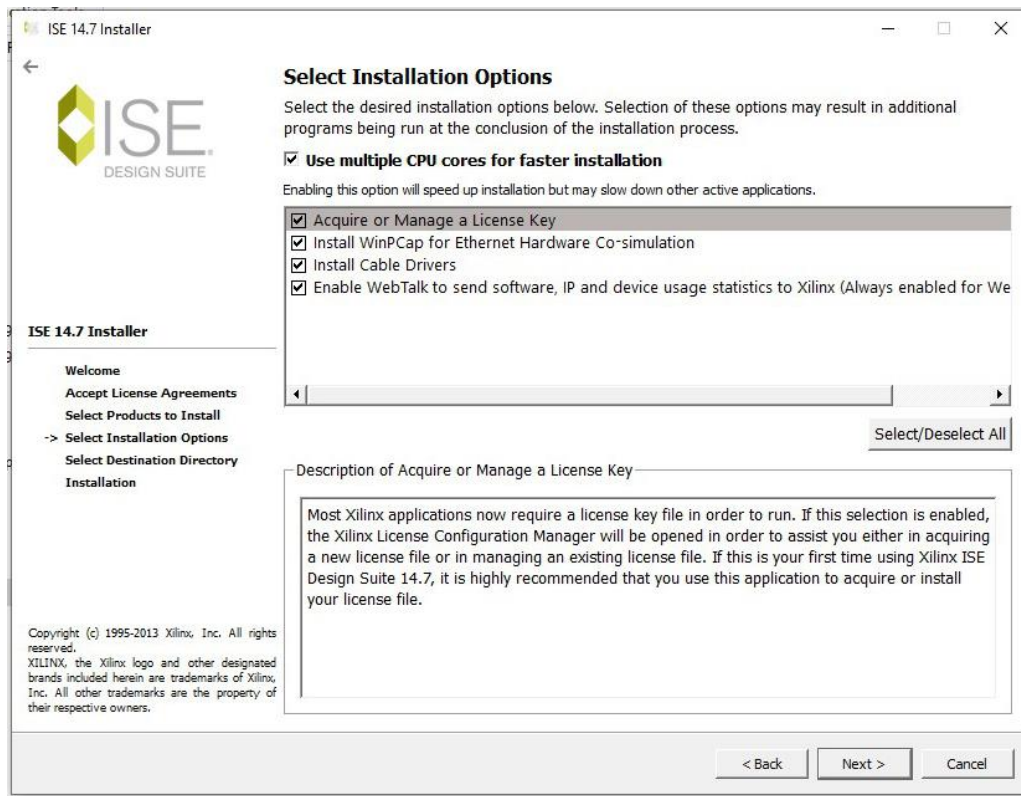
4. ให้กด Next ไปเรื่อยๆ



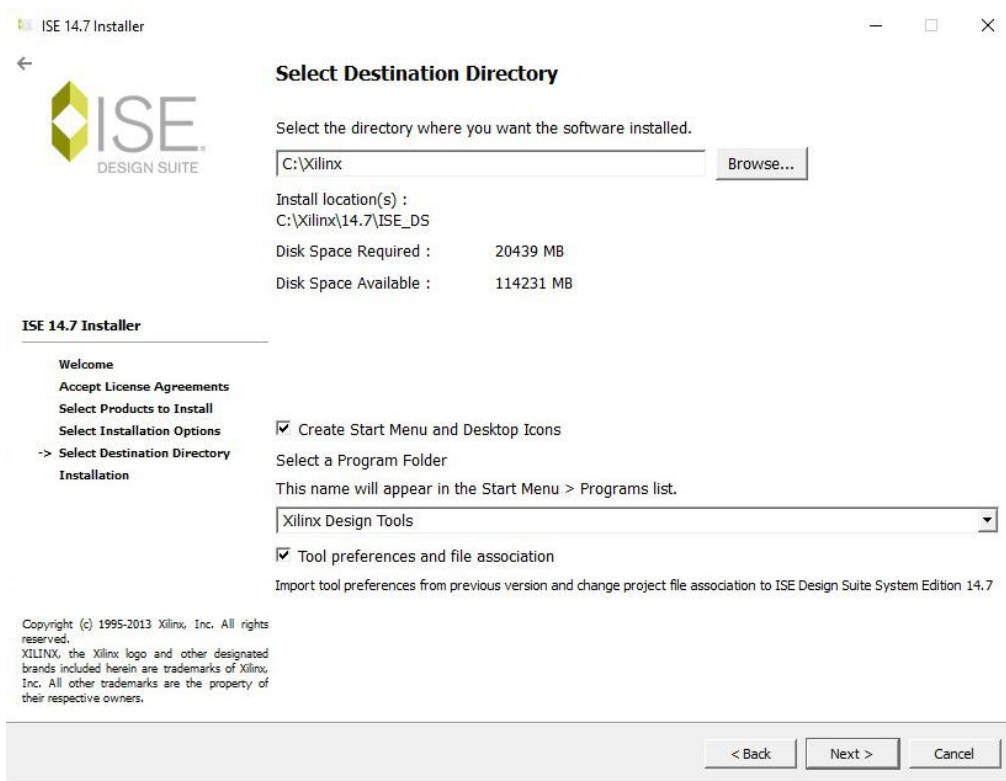
5. เมื่อถึงหน้านี้ให้เลือก ISE Design Suite System Edition จากนั้นกด Next



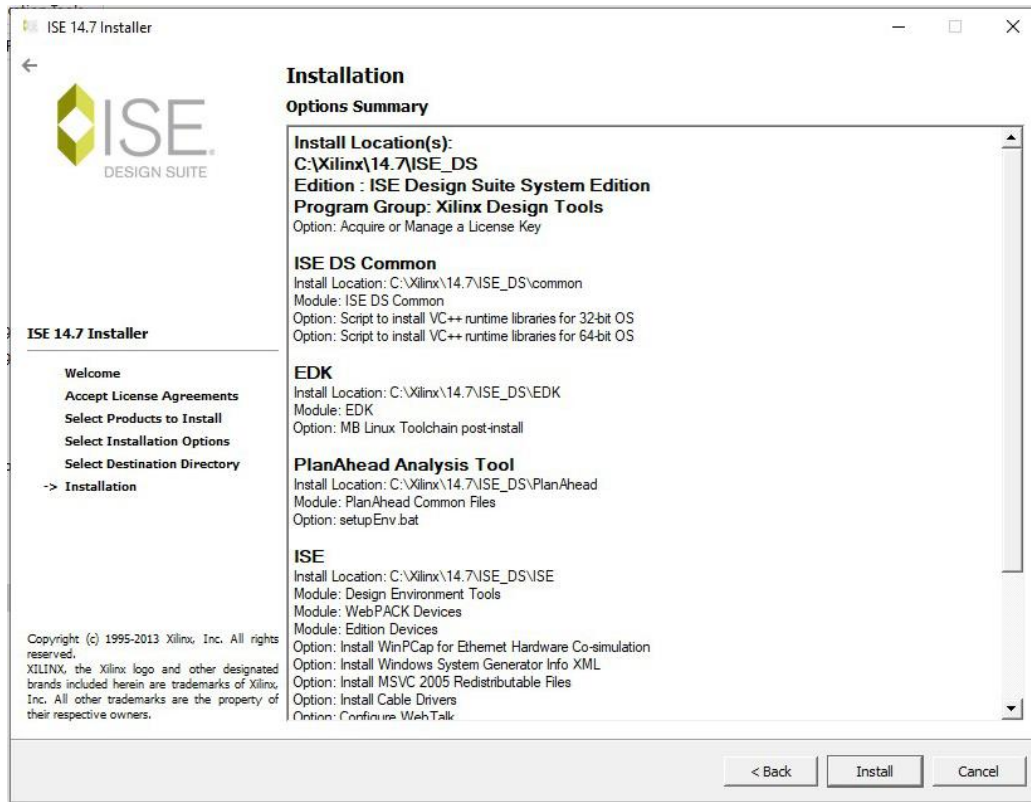
6.เมื่อถึงหน้านี้ให้กดเลือกทั้งหมด จากนั้นกด Next



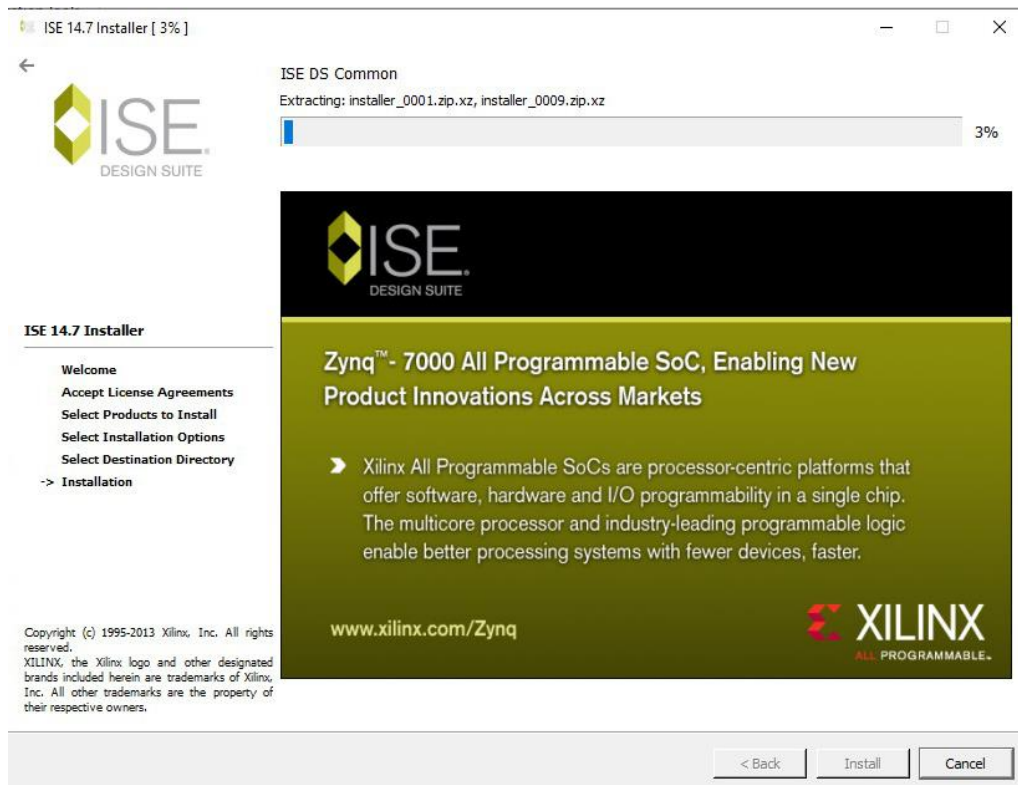
7.เลือกโฟลเดอร์ที่จะติดตั้ง จากนั้นกด Next



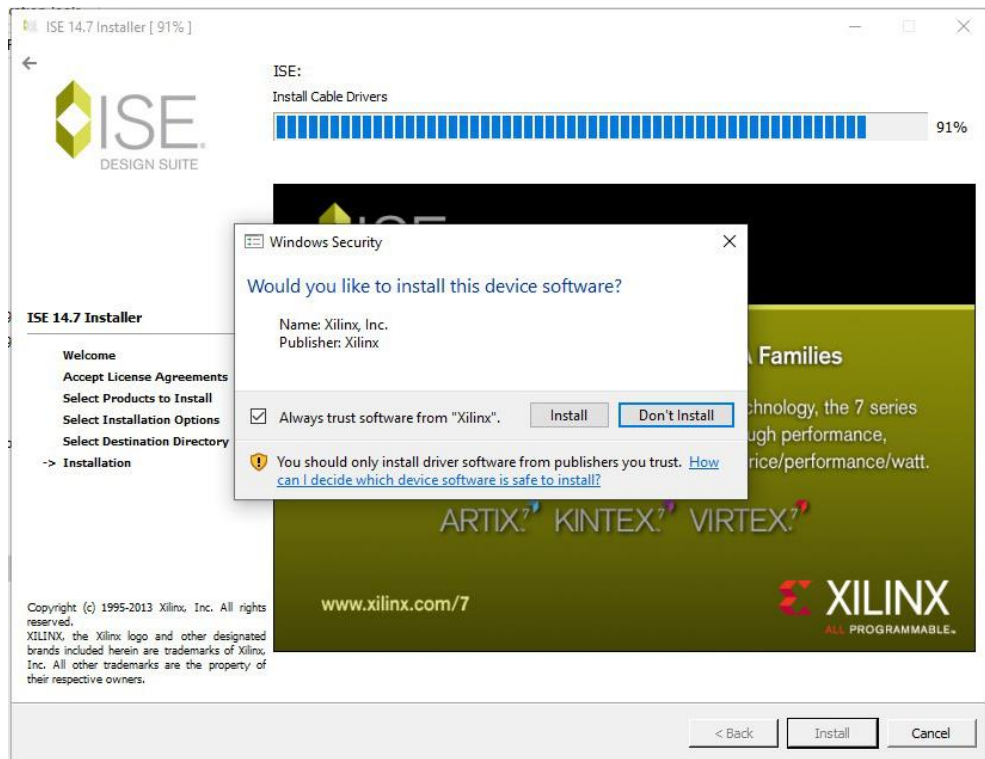
8.เมื่อถึงหน้านี้ให้กด Install



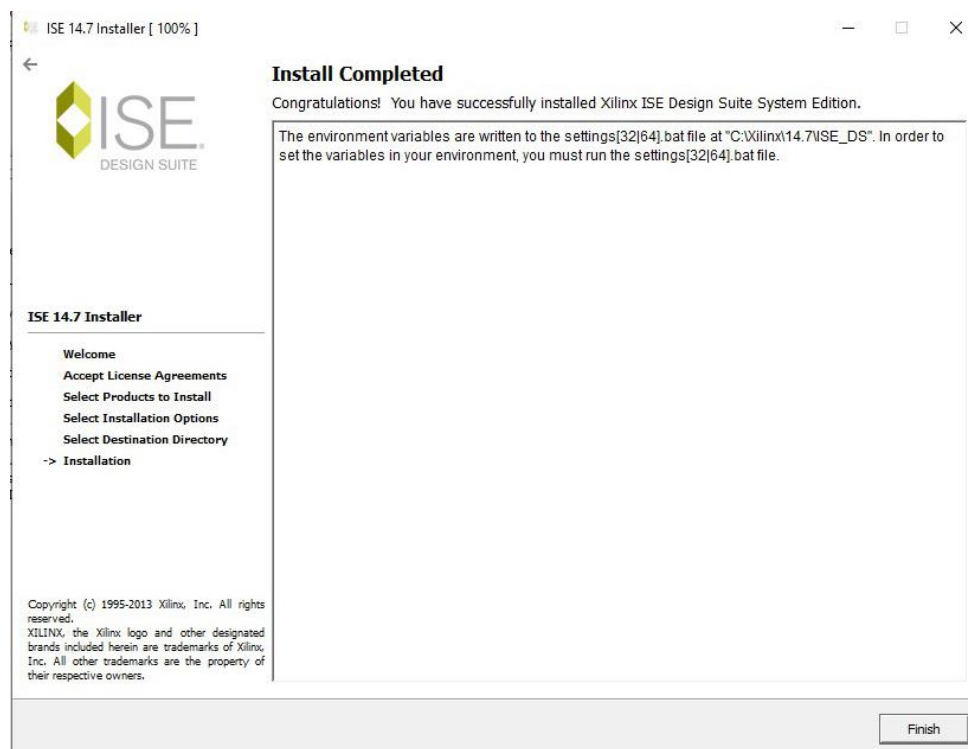
9. เมื่อกด Install แล้วหน้าจอจะแสดงผลดังภาพ ให้รอจนกว่าจะครบ 100%



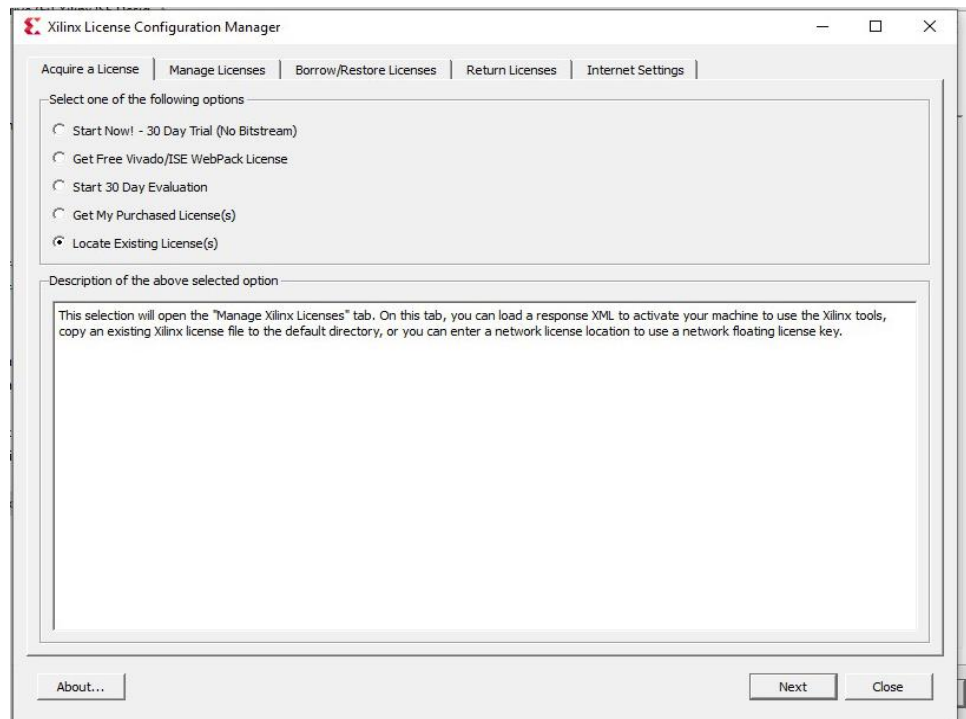
10. หากมีแจ้งเตือนดังรูปให้กด Install



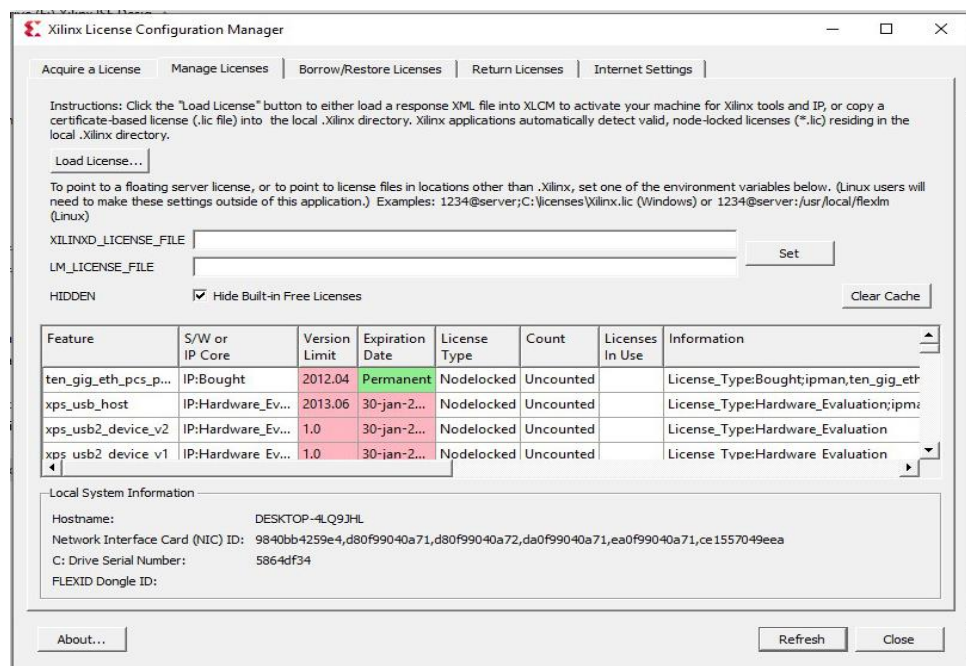
11. เมื่อเสร็จสิ้นจะขึ้นดังรูป



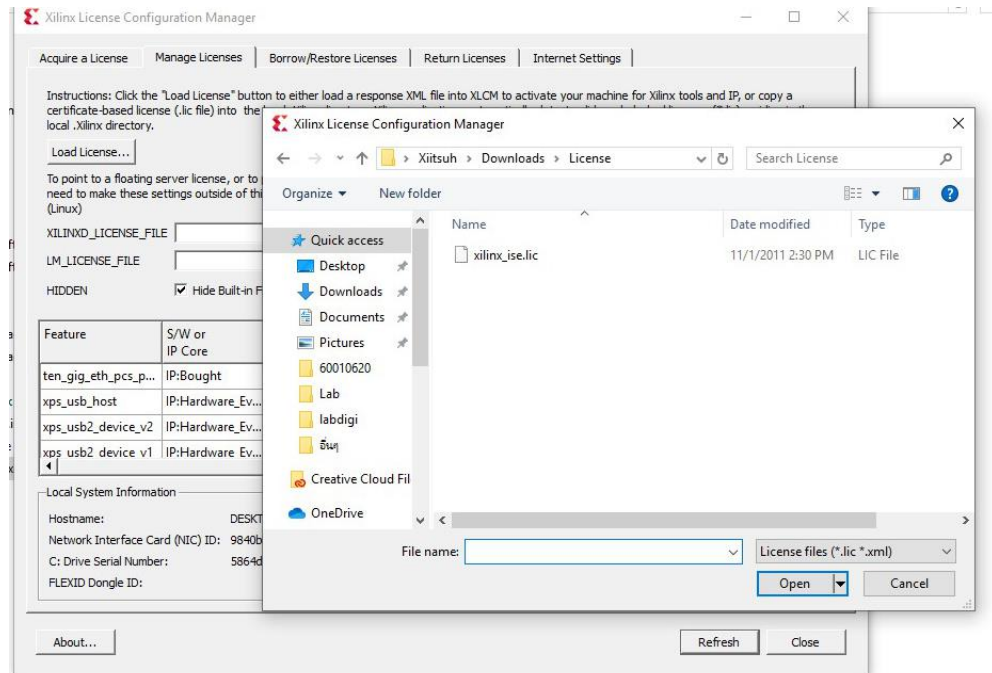
12.เมื่อติดตั้งเสร็จจะแสดงหน้าจอขึ้นมาจากนั้นให้กดเลือก Locate Existing License(s) จากนั้นกด Next



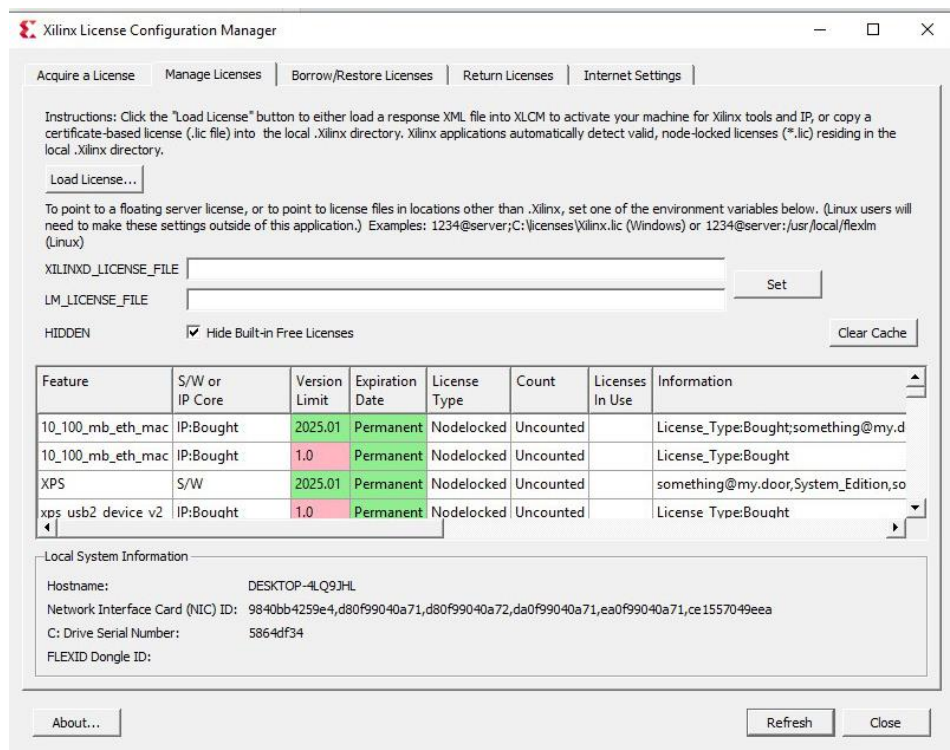
13.ให้ Download Licenseจาก <https://drive.google.com/file/d/0BwHgcirkznvtV2UzTDdWa1p1TjQ/view>
จากนั้นคลิกที่ Load License



14. จากนั้นเลือกไฟล์ License ที่ Download มา



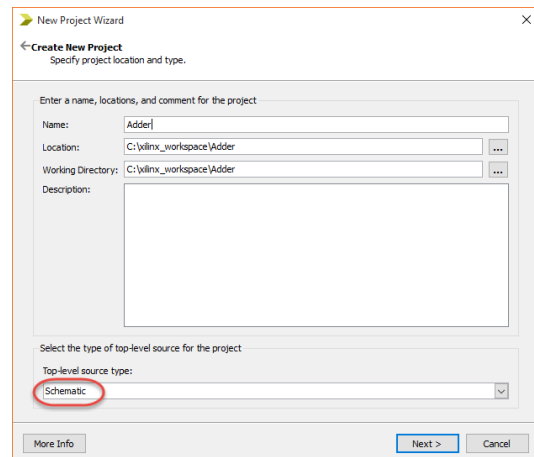
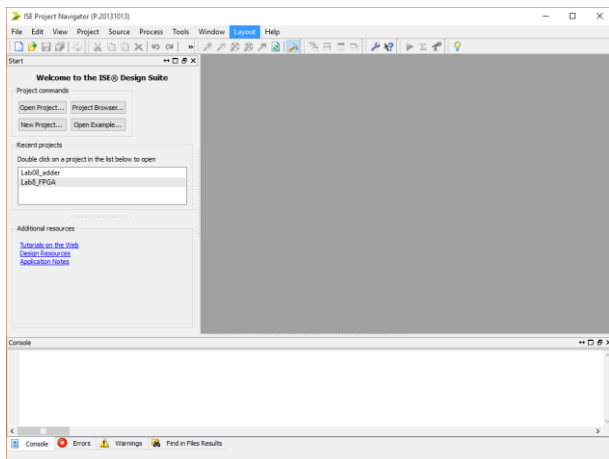
15. เมื่อเสร็จสิ้นจะขึ้นดังรูป



การใช้งานโปรแกรม

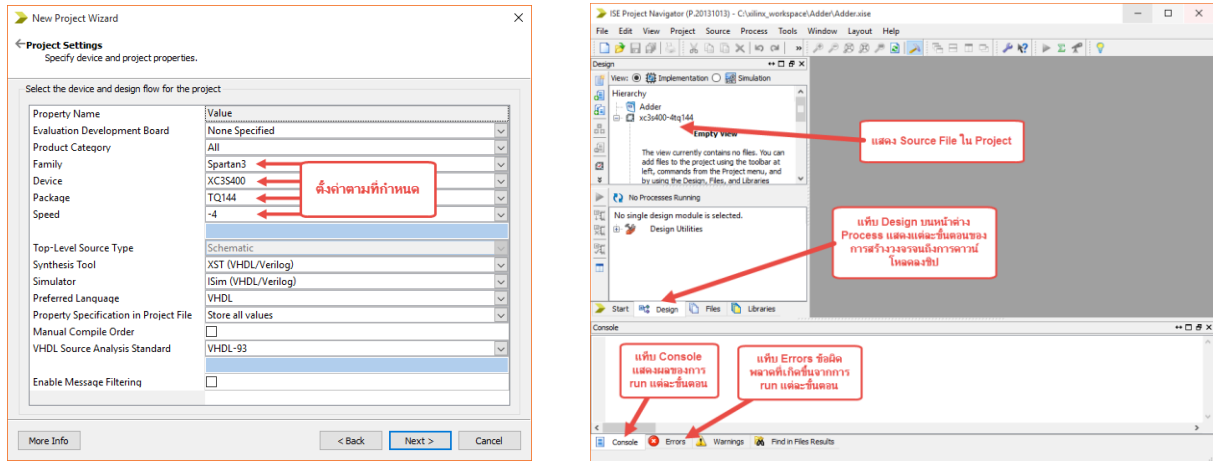
การทดลองนี้ใช้โปรแกรม Xilinx ISE WebPack 14.7 ให้ทดลองตามขั้นตอนดังนี้

1. เปิดโปรแกรม Xilinx ISE Design Suite 14.7 (Run as administrator) แล้วสร้าง Project ขึ้นใหม่ (File -> New Project) ดังรูปที่ 7
2. กำหนด Project Name (ต้องขึ้นต้นด้วยตัวอักษร) กำหนด Project Location ระบุ Top-Level Source Type เป็น Schematic เลือก Next
3. กำหนดลักษณะของชิป FPGA ที่ใช้โดยกำหนดพารามิเตอร์ดังรูปที่ 8 (ซ้าย) จากนั้นเลือก Next ไปเรื่อยๆ จน Finish หน้าจอของโปรแกรมมีลักษณะดังรูปที่ 8 (ขวา)
4. ลักษณะของ Project ในโปรแกรม ประกอบด้วย Source Files หลายตัว ซึ่งอาจสร้างจาก Schematic, VHDL, Verilog, State-Machine เพื่อประกอบเป็น Project ใหญ่ ให้เลือก Project -> New Source ดังรูปที่ 9

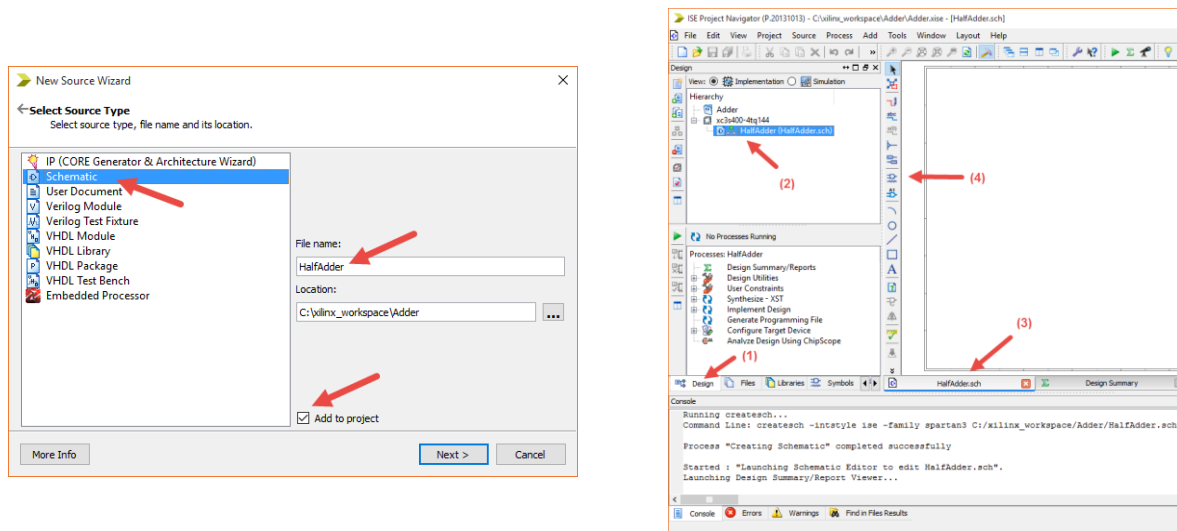


รูปที่ 7 หน้าต่างโปรแกรม Xilinx 8.2 (ซ้าย) และการกำหนดไฟล์ Project (ขวา)

อย่าลืม! เซฟ (Ctrl+S) บ่อย ๆ
 เพราะ Xilinx ISE อาจเกิด Program Error ได้ง่าย
"If anything can go wrong, it will." -- Murphy's Law



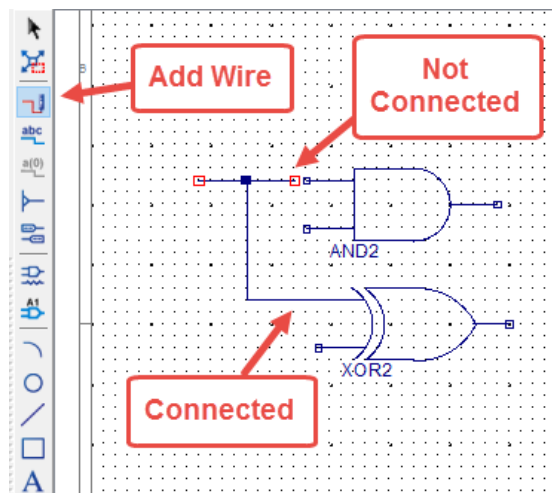
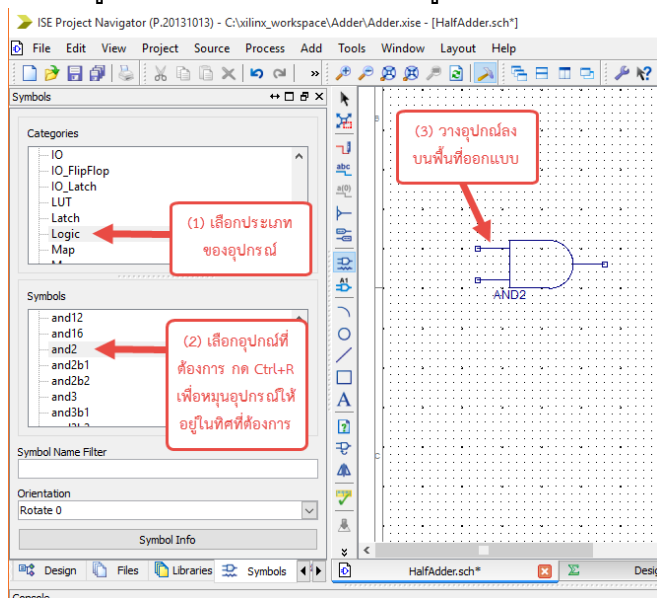
รูปที่ 8 การกำหนดลักษณะของ FPGA ที่ใช้งาน (ซ้าย) ส่วนประกอบหลักของโปรแกรม (ขวา)



รูปที่ 9 สร้าง Schematic (ซ้าย) และส่วนประกอบภายในโปรแกรม (ขวา)

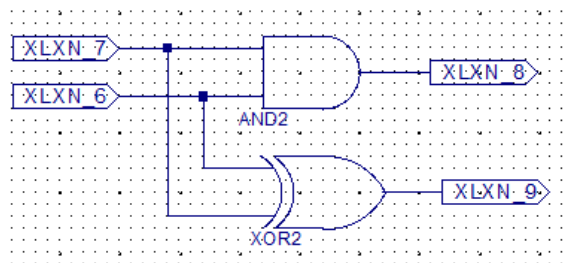
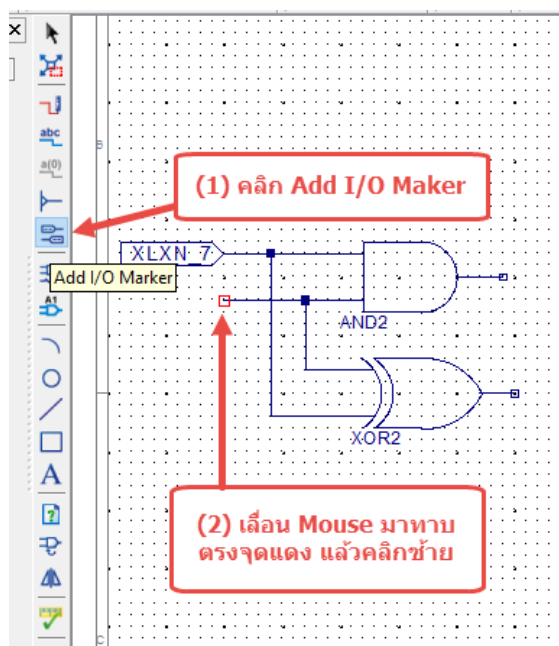
- เลือก Schematic และกำหนดชื่อไฟล์ของวงจรที่จะสร้าง (ต้องขึ้นต้นด้วยตัวอักษร) ดังรูปที่ 9 เลือก Next แล้ว Finish ควรได้ผลลัพธ์ดังรูปที่ 9 (ขวา) โดย (1) เลือกแท็บ Design โดยแสดงเป็น Tree ตัวในสุด (2) แสดงเป็น Source File ที่สร้างขึ้น (3) เลือกแท็บพื้นที่ซึ่งใช้ในการวาดวงจร และ (4) ใช้แถบด้านข้างสำหรับเป็นเครื่องมือวาดวงจร

6. เริ่มวาดวงจรแรกเป็น 1-Bit Half Adder อุปกรณ์ประกอบคือเกต XOR กับ AND โดยหาอุปกรณ์ในแท็บ Symbols ตามชื่อของอุปกรณ์ที่ต้องการ ดังรูปที่ 10 (ซ้าย) วางอุปกรณ์ทั้งสองบนพื้นที่ออกแบบ
7. จากนั้นลากเส้นโดยเลือก Add Wire (**ระวัง! ห้ามใช้ Add Line**) เชื่อมต่อสาย ลักษณะการเชื่อมต่อที่สมบูรณ์(Connected) และไม่สมบูรณ์ (Not Connected) แสดงดังรูปที่ 10 (ขวา)



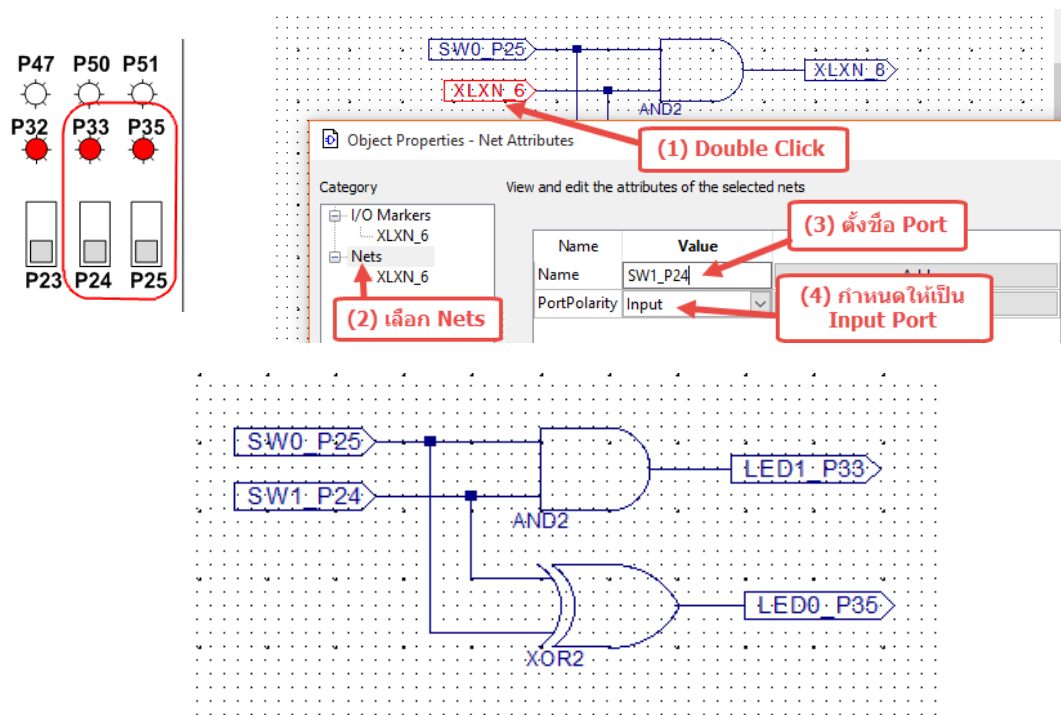
รูปที่ 10 การวางอุปกรณ์ (ซ้าย) และการเชื่อมต่อสาย (ขวา)

8. สร้างจุดสำหรับเชื่อมต่อกับขาของ ชิป FPGA โดยเลือก Add I/O Marker ดังรูปที่ 11 (ซ้าย) วางจุดเชื่อมต่อจนได้ผลลัพธ์ เป็นวงจร 1-Bit Half Adder ดังรูปที่ 11 (ขวา)



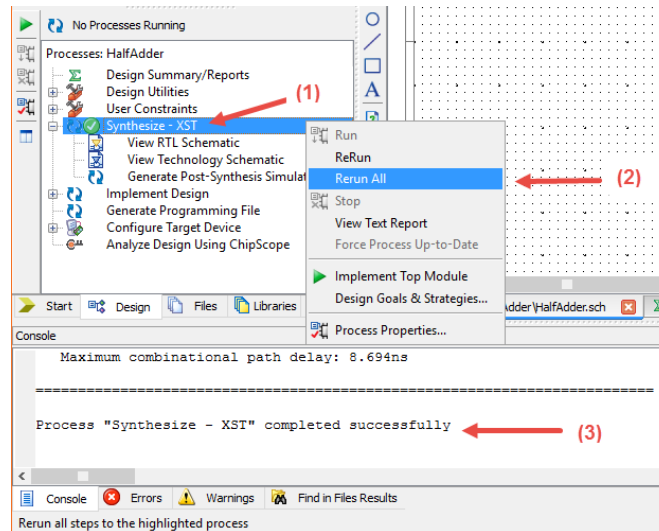
รูปที่ 11 สร้างจุดเชื่อมต่อ (ซ้าย) และวงจรเมื่อวาดเสร็จแล้ว (ขวา)

9. เปลี่ยนชื่อตัวรับสัญญาณเข้าและออกใน I/O Marker โดยพิจารณารูปที่ 1 หรือ 12 (บนซ้าย) สำหรับการทดลองนี้ให้เลือกสวิตช์ที่เชื่อมต่อกับขา 24 และ 25 ของ FPGA เป็นตัวป้อนสัญญาณอินพุต และใช้ LED ที่เชื่อมต่อกับขา 33 และ 35 ของ FPGA เป็นตัวแสดงสัญญาณเอาต์พุต โดยกำหนดให้ SW0 และ SW1 เป็นชื่อสัญญาณอินพุต และ LED0 และ LED1 เป็นชื่อสัญญาณเอาต์พุตเพื่อแสดงผลรวมและตัวทดตามลำดับ ทั้งนี้ควรตั้งชื่อในรูปแบบ “ชื่อสัญญาณ_หมายเลขขา” เช่น SW0_P25 หมายถึง ใช้สัญญาณ SW0 ต่อกับขา P25 แม้ว่าจะไม่ได้เป็นข้อบังคับ แต่การตั้งชื่อรูปแบบนี้ช่วยให้แก้ไขปัญหาดังกล่าวได้ง่าย รูปที่ 12 (บนขวา) แสดงหน้าต่างสำหรับการตั้งชื่อ และรูปที่ 12 (ล่าง) แสดงการตั้งชื่อ I/O Marker ที่เสร็จสมบูรณ์



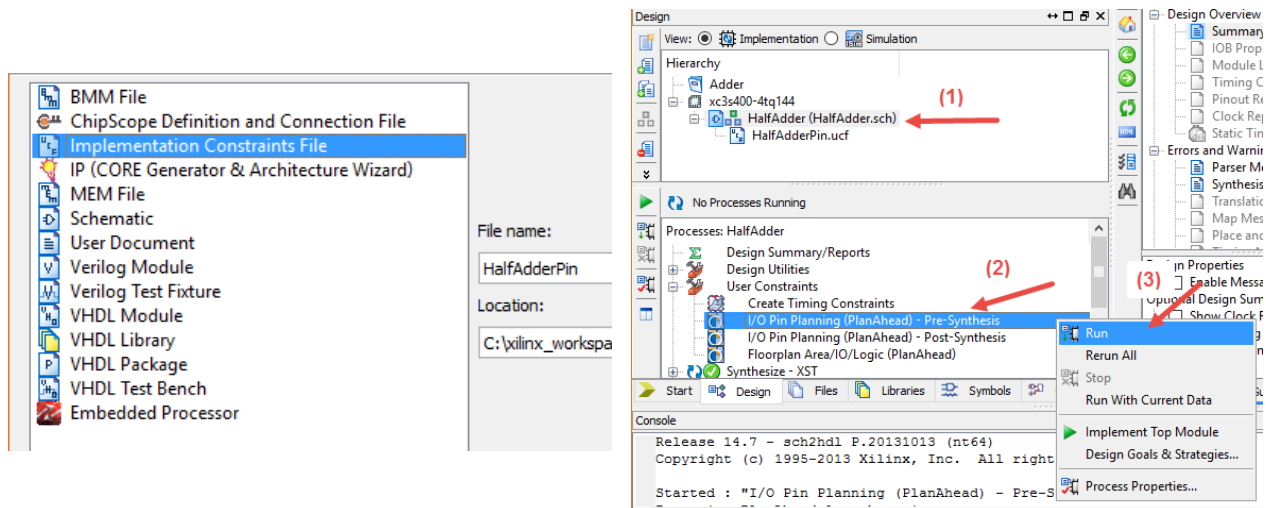
รูปที่ 12 การกำหนดชื่อตัวรับสัญญาณเข้าและออกใน I/O Marker

10. ตรวจสอบการทำงานของวงจรที่ออกแบบโดยเลือกแท็บ Design ที่หน้าต่าง Process เลือก Synthesize - XST แล้วดับเบิลคลิกหรือกดคลิกขวาแล้วเลือก Rerun All เพื่อเริ่มตรวจสอบวงจร หากการเดินสายและการวางอุปกรณ์ถูกต้องจะแสดง **Process “Synthesize” completed successfully** ที่หน้าต่าง Console ดังรูปที่ 13 กรณีเกิดข้อผิดพลาดต้องกลับไปตรวจสอบวงจรตามขั้นตอนก่อนหน้า



รูปที่ 13 ขั้นตอนตรวจสอบ Schematic Design

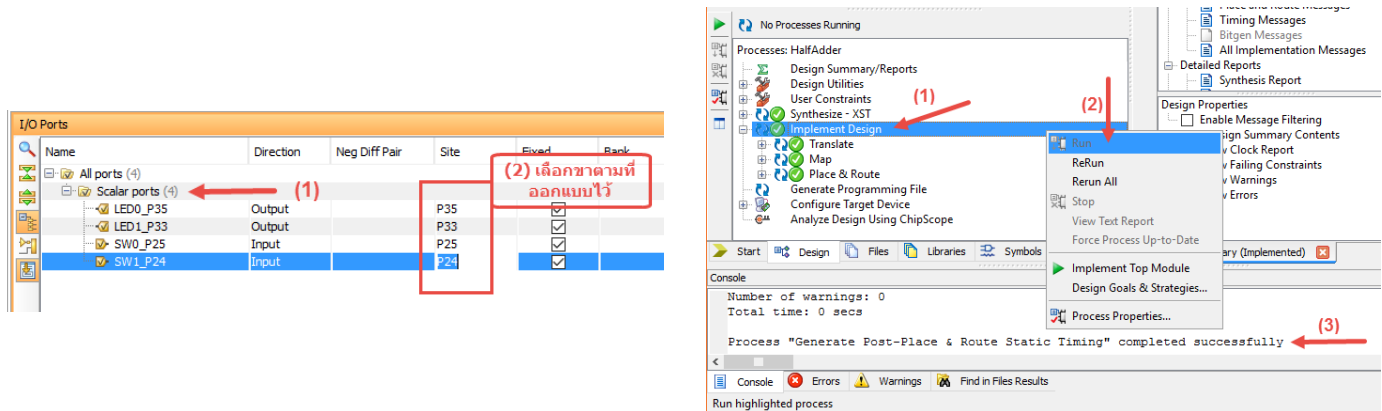
11. จับคู่อุปกรณ์กับหมายเลขพอร์ต โดยเลือก Project -> New Source -> Implementation Constraints File เพื่อจับคู่ระหว่างอุปกรณ์บนบอร์ดกับหมายเลขขาไอซี (Pin Number) ดังรูปที่ 14



รูปที่ 14 การจับคู่อุปกรณ์กับหมายเลขพอร์ต

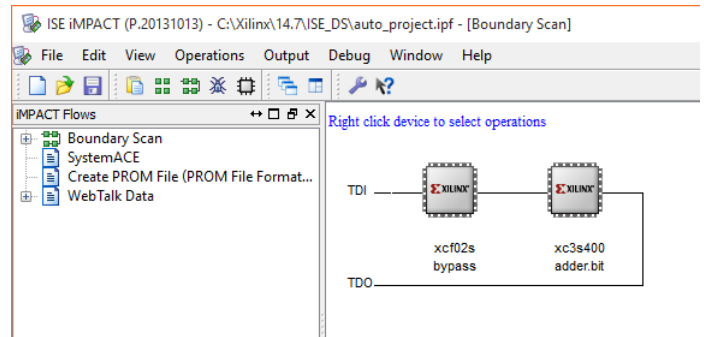
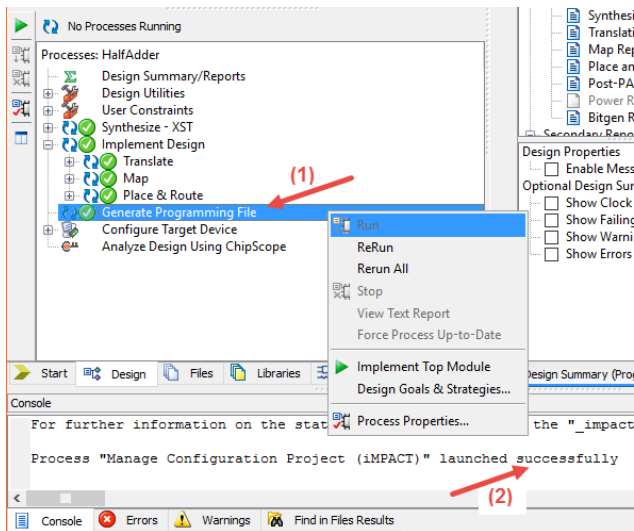
12. เลือกตามขั้นตอนดังรูปที่ 14 (ขวา) เลือกไฟล์ Schematic Source จากนั้นที่หน้าต่าง Process เลือกขั้นตอน User Constraints แล้วคลิกขวาที่ I/O Pin Planning (PlanAhead – Pre-Synthesis) เลือก Run จากนั้นโปรแกรม PlanAhead จะปรากฏขึ้นมา ดังรูปที่ 15

13. กำหนดหมายเลขขาลงในหน้าต่าง I/O Port ซึ่งอยู่ด้านล่างของโปรแกรม PlanAhead แล้วใส่รายละเอียดให้ครบเหมือนดังรูปที่ 15 (ซ้าย) หากไม่สามารถระบุเลขลงไปได้หรือโปรแกรมปฏิเสธแสดงว่ากำหนดหมายเลขขาผิด



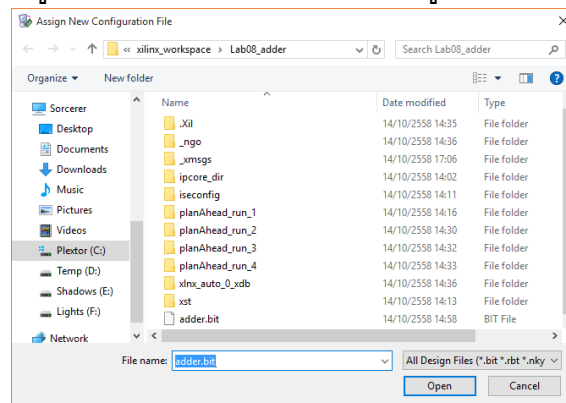
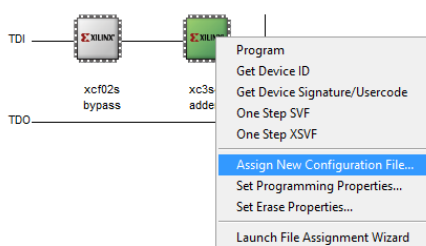
รูปที่ 15 กำหนดเลขขาลงใน Xilinx PlanAhead (ซ้าย) และ Implement Design (ขวา)

14. เริ่มขั้นตอน Implement Design โดยทำตามขั้นตอนในรูปที่ 15 (ขวา) ถ้าไม่มีข้อผิดพลาดจะแสดงเครื่องหมายถูกด้านหน้าหัวข้อ และแสดง **Process "... completed successfully** ที่หน้าต่าง Console กรณีเกิดข้อผิดพลาดต้องกลับไปตรวจสอบการจับคู่อุปกรณ์กับหมายเลขพอร์ตตามขั้นตอนก่อนหน้า
15. บันทึกโปรแกรมลงในชิป FPGA โดยทำตามขั้นตอนในรูปที่ 17 (ซ้าย) เลือก Generate Programming File เลือก Run, Rerun All จากนั้นเปิดโปรแกรม iMPACT (Run as administrator) หากการเชื่อมต่อสมบูรณ์จะปรากฏดังรูปที่ 17 (ขวา) หากไม่สำเร็จให้ทดลองทำ Initialize Chain โดยเลือก File -> Initialize Chain หากยังไม่สำเร็จแสดงว่ามีปัญหาในการเชื่อมต่อสาย USB ให้ลองสำรวจการเชื่อมต่อสาย USB ว่ามีปกติหรือไม่ หรือลองเปลี่ยนไปใช้พอร์ต USB พอร์ตอื่นบนเครื่อง หากการเชื่อมต่อสมบูรณ์ตั้งแต่ตอนเปิดโปรแกรม โปรแกรมจะถามหาไฟล์ที่ใช้ดาวน์โหลดบอร์ด FPAG โดยอัตโนมัติ โดยจะแสดงหน้าต่างถามหาไฟล์ .MCS ให้กด Cancel ข้ามไป และจะถามหาไฟล์ .BIT ก็ให้กด Cancel ข้ามไปเช่นเดียวกัน



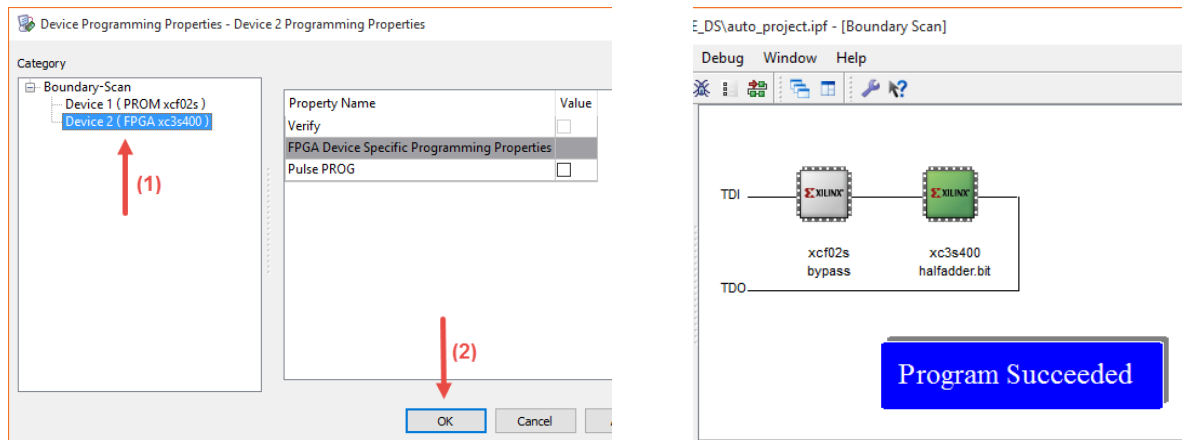
รูปที่ 16 การ Generate Programmable File และโปรแกรม iMPACT

16. จากนั้นเริ่มต้นการโปรแกรมลงบอร์ด FPGA โดยเลือกคำสั่ง Assign New Configuration File ดังรูปที่ 18 (ซ้าย) จากนั้นโปรแกรม iMPACT จะแสดงหน้าต่างถามหาไฟล์นามสกุล .BIT ให้เลือกไฟล์นามสกุลดังกล่าวที่มีชื่อไฟล์เป็นชื่อเดียวกับ Project ซึ่งจะอยู่ภายใต้โฟลเดอร์ของ Project ดังรูปที่ 18 (ขวา)



รูปที่ 17 การกำหนดไฟล์ .BIT

17. เมื่อเลือกไฟล์ .BIT ไว้แล้ว ขั้นตอนสุดท้ายคือบันทึกลงในชิป FPGA ให้คลิกขวานรูปชิป XC3S400 สั่ง Program ดังรูปที่ 19 (ซ้าย) หากประสบความสำเร็จจะขึ้นข้อความ **Program Succeeded** ดังรูปที่ 19 (ขวา)



รูปที่ 18 การบันทึกลงในชิป FPGA

หมายเหตุ

บอร์ดที่ใช้ทดลองเป็นบอร์ดรุ่นใหม่ ขาและรูปแบบอาจจะไม่ตรงกับในหนังสือ รวมถึงวิธีการนำไฟล์ .bit ลง FPGA

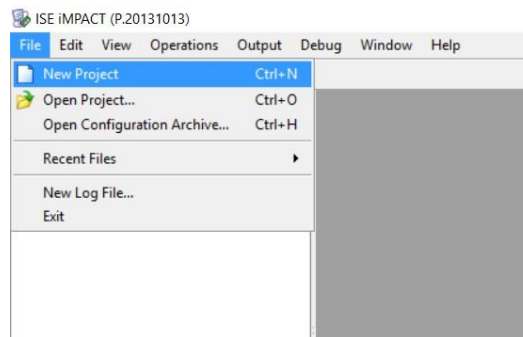
คู่มือการใช้งานโมดูล FT2232H ในการโปรแกรมบอร์ด Xilinx SERVEYOR-6

หมายเหตุ ถ้าหากเสร็จในขั้นการเตรียมไฟล์ SVF สำหรับโปรแกรมลงแฟลชของ FPGA จะสามารถ run ใน UrJTAG ได้ แต่เมื่อถอดปลั๊ก โปรแกรมจะหายไปจาก FPGA หากต้องการให้ FPGA ยังมีโปรแกรมที่เรา run อยู่แม้จะถอดปลั๊กให้ลงโดยใช้ PROM

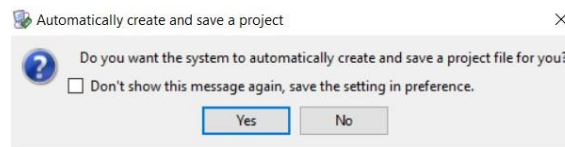
การใช้งาน FT2232

ขั้นตอนการเตรียมไฟล์ SVF สำหรับโปรแกรมลงแฟลชของ FPGA

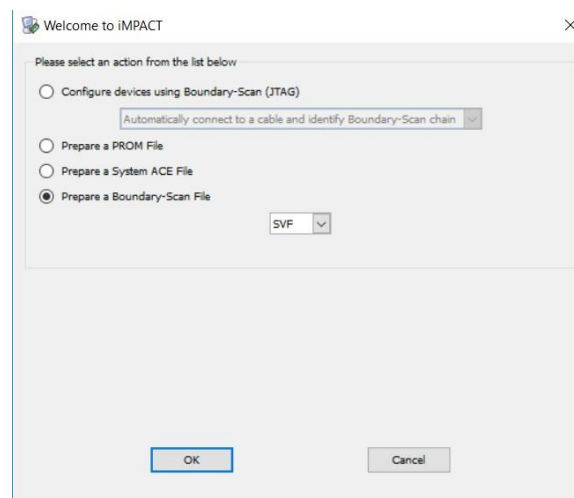
1. เปิดโปรแกรม iMPACT แล้วกดสร้าง New Project



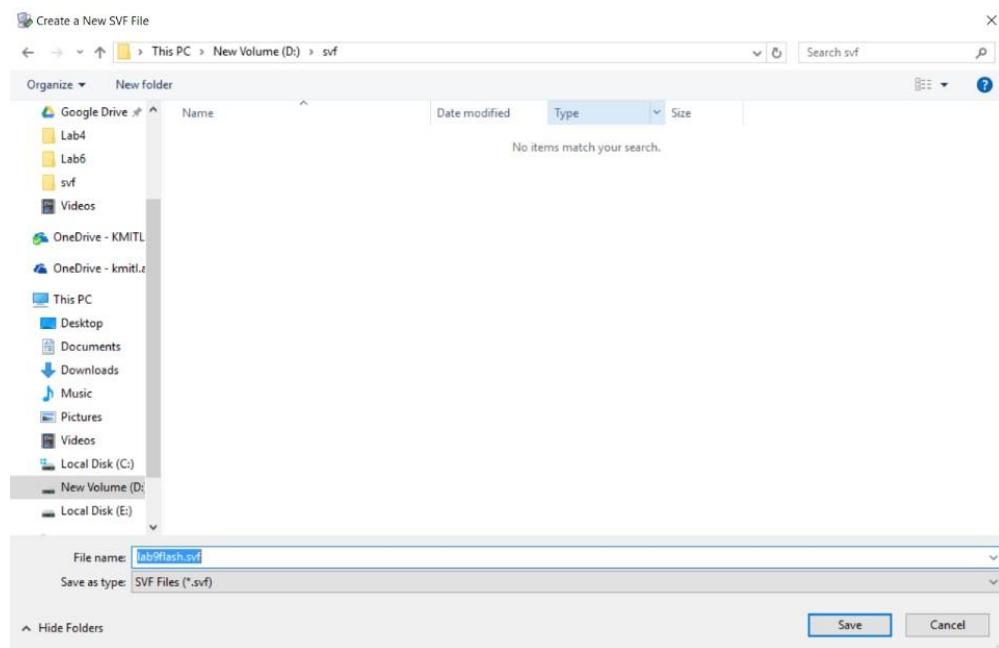
2. กด Yes



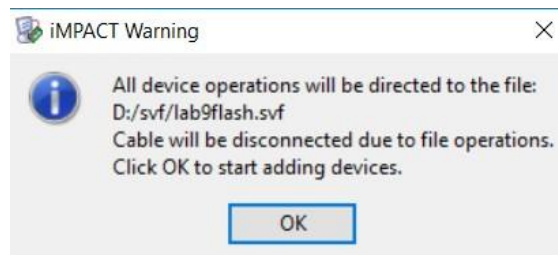
3. เลือก option Prepare a Boundary-Scan File เลือก SVF แล้วกด OK



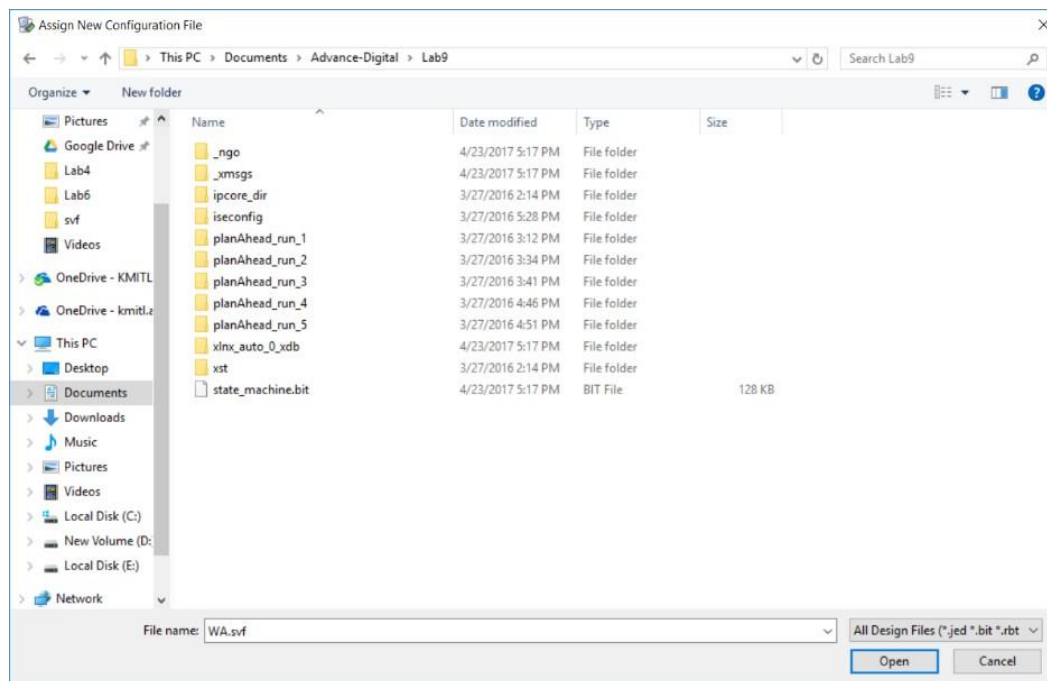
4. เลือกพื้นที่สำหรับ save ไฟล์ SVF ซึ่งเป็นไฟล์ที่ใช้สำหรับโปรแกรมผ่านโมดูล FT2232H โดยในตัวอย่างได้ save ไว้ในโฟลเดอร์ที่สร้างขึ้นมา ชื่อ svf ในไดรฟ์ D



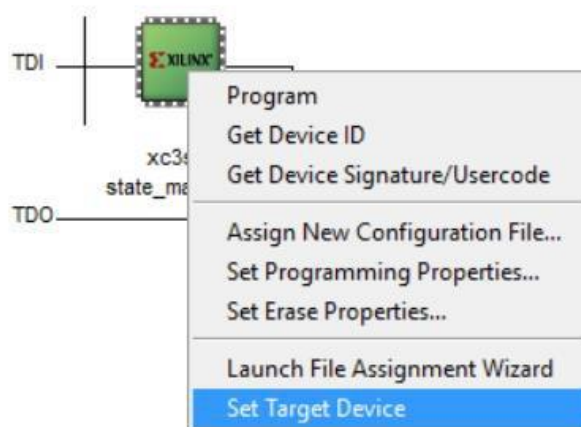
5. กด OK



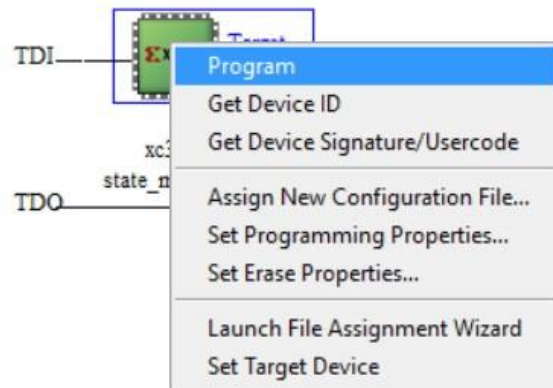
6. ให้เลือกไฟล์นามสกุล .bit จากโฟลเดอร์ของ Project ที่เราต้องการโปรแกรมข้อมูลลง FPGA แล้วกด Open



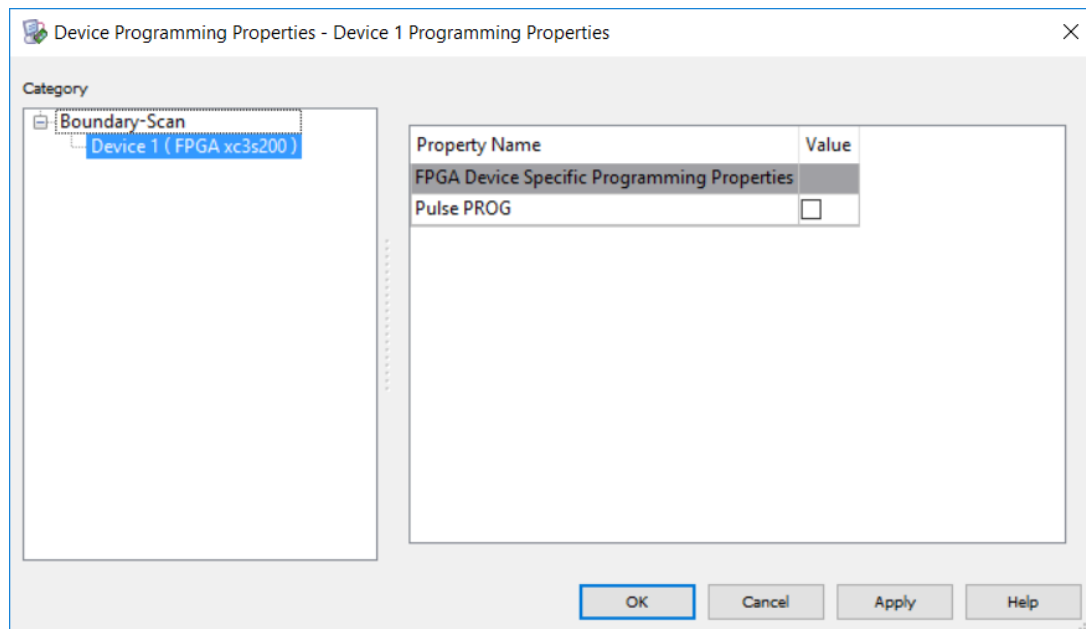
7. จะมีสัญลักษณ์ขึ้นปรากฏขึ้น คลิกขวาแล้วเลือก Set Target Device



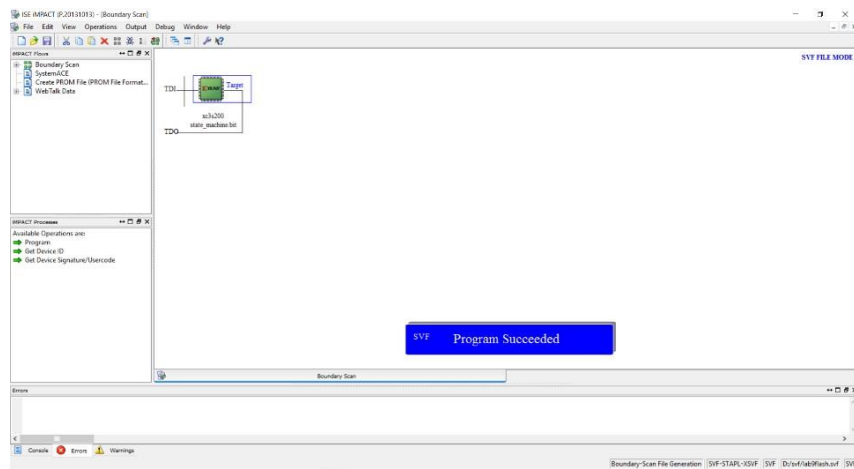
8. คลิกขวาที่ชิปอีกครั้ง แล้วเลือกกด Program



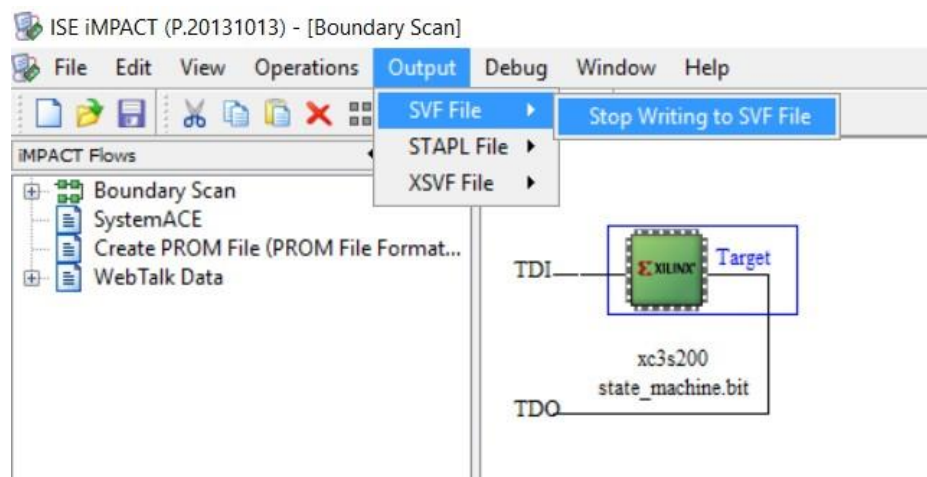
9. กด OK



10. เมื่อโปรแกรมทำงานสำเร็จ จะแสดงข้อความบอกดังรูป

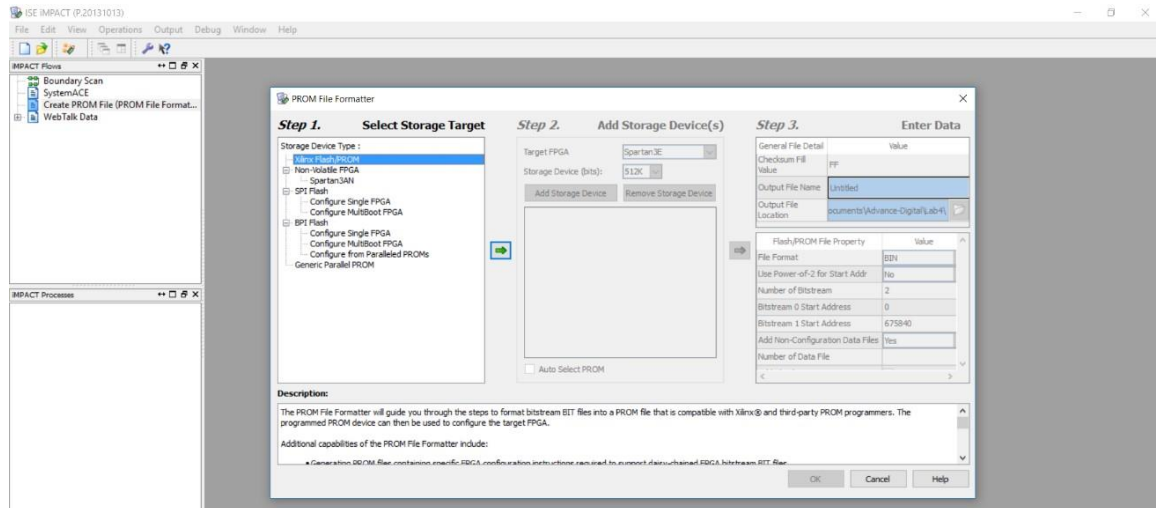


11. ไปที่เมนู Output -> SVF File -> เลือก Stop Writing to SVF File จะเป็นการเสร็จสิ้นการเตรียมไฟล์ SVF สำหรับ โปรแกรมลงเฟิร์มแวร์

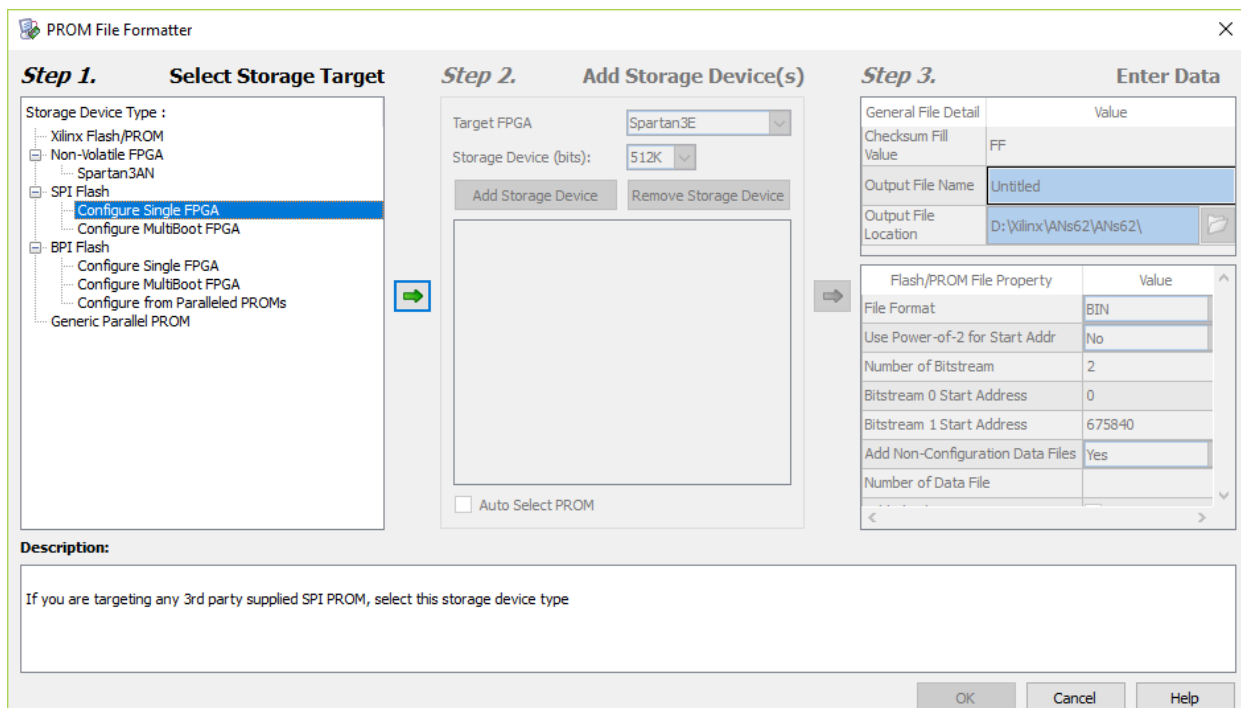


ขั้นตอนการเตรียมไฟล์ SVF สำหรับโปรแกรม PROM ผ่าน SPI

1. เปิดโปรแกรม iMPACT แล้วดับเบิลคลิกที่ Create PROM File(PROM File Formatter) ที่เมนูด้านซ้าย จะปรากฏหน้าต่างตามรูปด้านล่าง



2. Step 1. ในส่วนของ SPI Flash เลือก Configure Single FPGA แล้วกดลูกศรสีเขียว



3. Step 2. เลือกขนาดของ PROM ที่ใช้ โดยในบอร์ด SERVEYOR-6 ดังเป็น 8M ตามขนาดของ M25P80 กด Add Storage Device แล้วกดลูกศร

PROM File Formatter

Step 1. Select Storage Target

Storage Device Type :

- Xilinx Flash/PROM
- Non-Volatile FPGA
 - Spartan3AN
- SPI Flash
 - Configure Single FPGA
 - Configure MultiBoot FPGA
- BPI Flash
 - Configure Single FPGA
 - Configure MultiBoot FPGA
 - Configure from Paralleled PROMs
- Generic Parallel PROM

Step 2. Add Storage Device(s)

Storage Device (bits): 8M

Add Storage Device Remove Storage Device

8M

☐ Auto Select PROM

Step 3. Enter Data

General File Detail	Value
Checksum Fill Value	FF
Output File Name	Untitled
Output File Location	D:\Xilinx\ANS62\ANS62\

Flash/PROM File Property	Value
File Format	BIN
Use Power-of-2 for Start Addr	No
Number of Bitstream	2
Bitstream 0 Start Address	0
Bitstream 1 Start Address	675840
Add Non-Configuration Data Files	Yes
Number of Data File	

Description:

In this step, you will select the appropriate target device.

- Storage Device:** This selection allows you to choose the specific device memory density you are targeting.
- Add Storage Device:** After selecting the memory target, use this button to add the device to the target Storage Device list below.
- Remove Storage Device:** Use this button to delete the target device from the list below. Select the device and click this button to remove it from the list.
- Auto Select PROM:** If you select this option, IMPACT will choose a device density large enough to hold your specified data.

OK Cancel Help

4. Step 3. สามารถตั้งชื่อไฟล์ และที่อยู่ได้ โดยไฟล์ฟอร์แมตต้องเป็น MCS แล้วกด OK

PROM File Formatter

Step 1. Select Storage Target

Storage Device Type :

- Xilinx Flash/PROM
- Non-Volatile FPGA
 - Spartan3AN
- SPI Flash
 - Configure Single FPGA
 - Configure MultiBoot FPGA
- BPI Flash
 - Configure Single FPGA
 - Configure MultiBoot FPGA
 - Configure from Paralleled PROMs
- Generic Parallel PROM

Step 2. Add Storage Device(s)

Storage Device (bits): 8M

Add Storage Device Remove Storage Device

8M

☐ Auto Select PROM

Step 3. Enter Data

General File Detail	Value
Checksum Fill Value	FF
Output File Name	Untitled
Output File Location	D:\Xilinx\ANS62\ANS62\

Flash/PROM File Property	Value
File Format	MCS
Add Non-Configuration Data Files	No

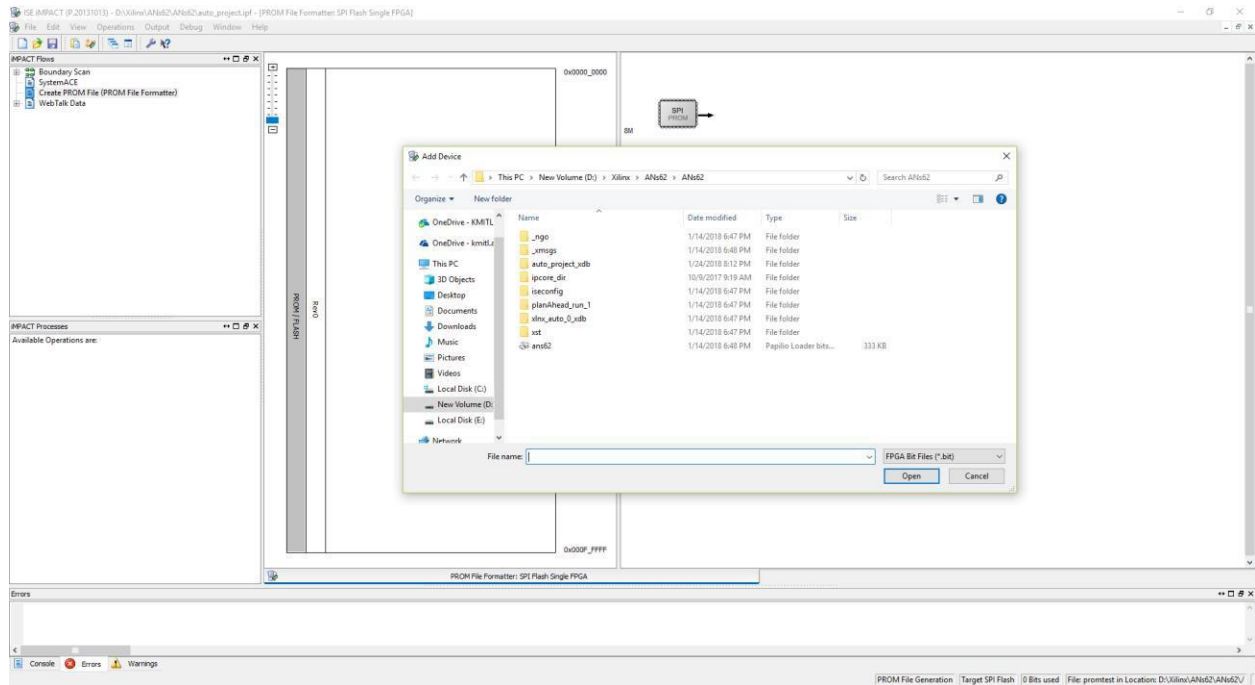
Description:

In this step, you will enter information to assist in setting up and generating a PROM file for the targeted storage device and mode.

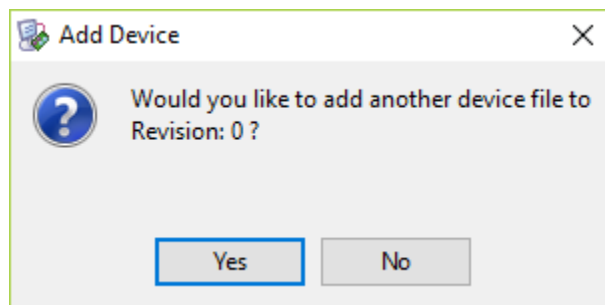
- Checksum Fill Value:** When data is insufficient to fill the entire memory of a PROM, the value specified here is used to calculate the checksum of the unused portions.
- Output File Name:** This allows you to specify the base name of the file to which your PROM data will be written
- Output File Location:** This allows you to specify the directory in which the file named above will be created
- File Format:** PROM files can be generated in any number of industry standard formats. Depending on the PROM file format your PROM programmer uses, you output a MCS

OK Cancel Help

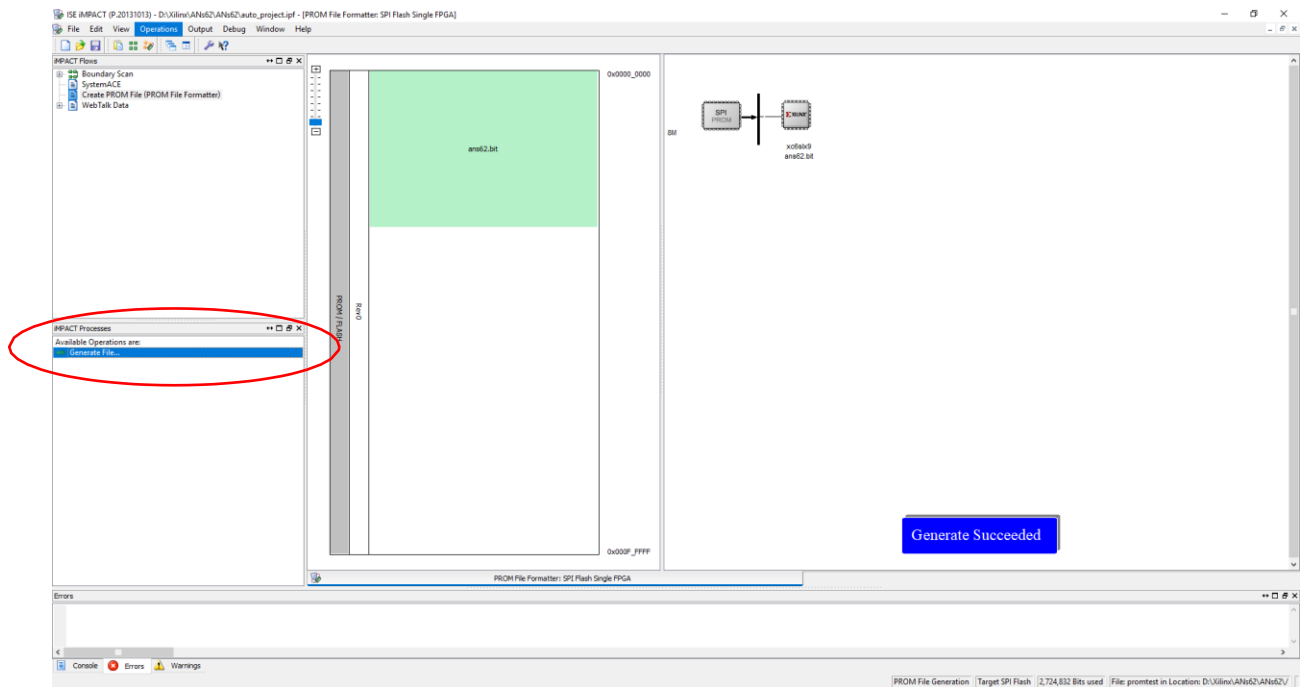
5. ให้เลือกไฟล์นามสกุล .bit ที่ต้องการ กด OK



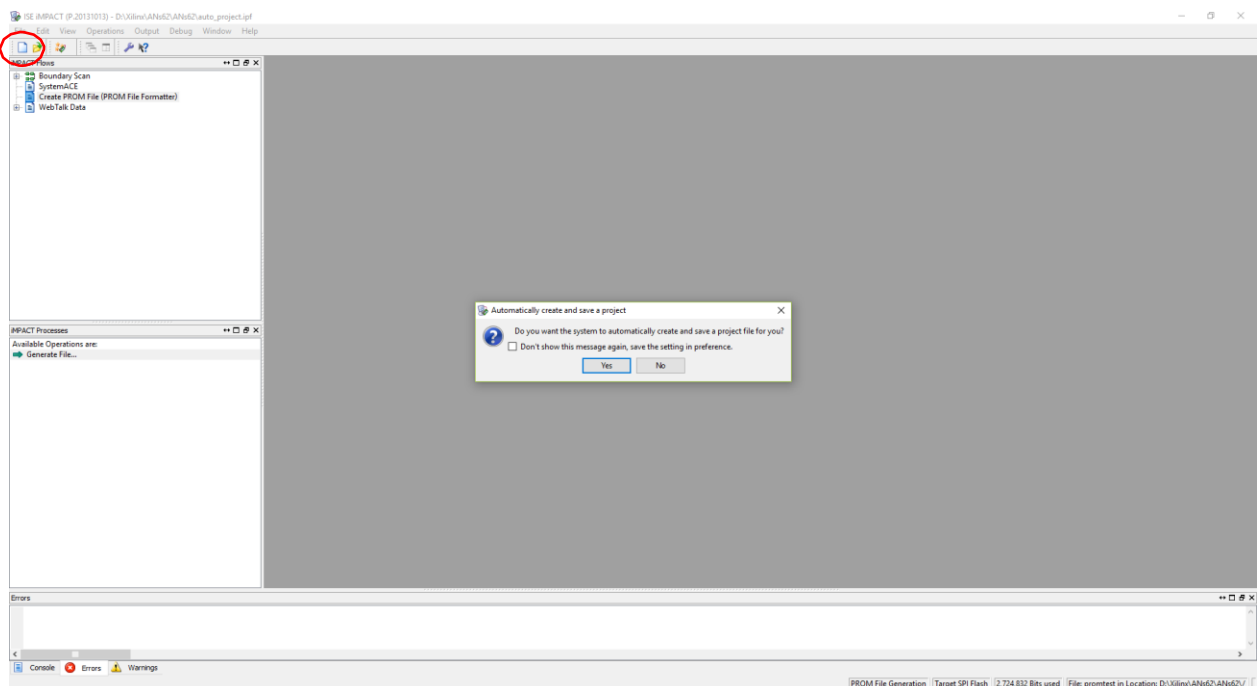
6. เลือก No



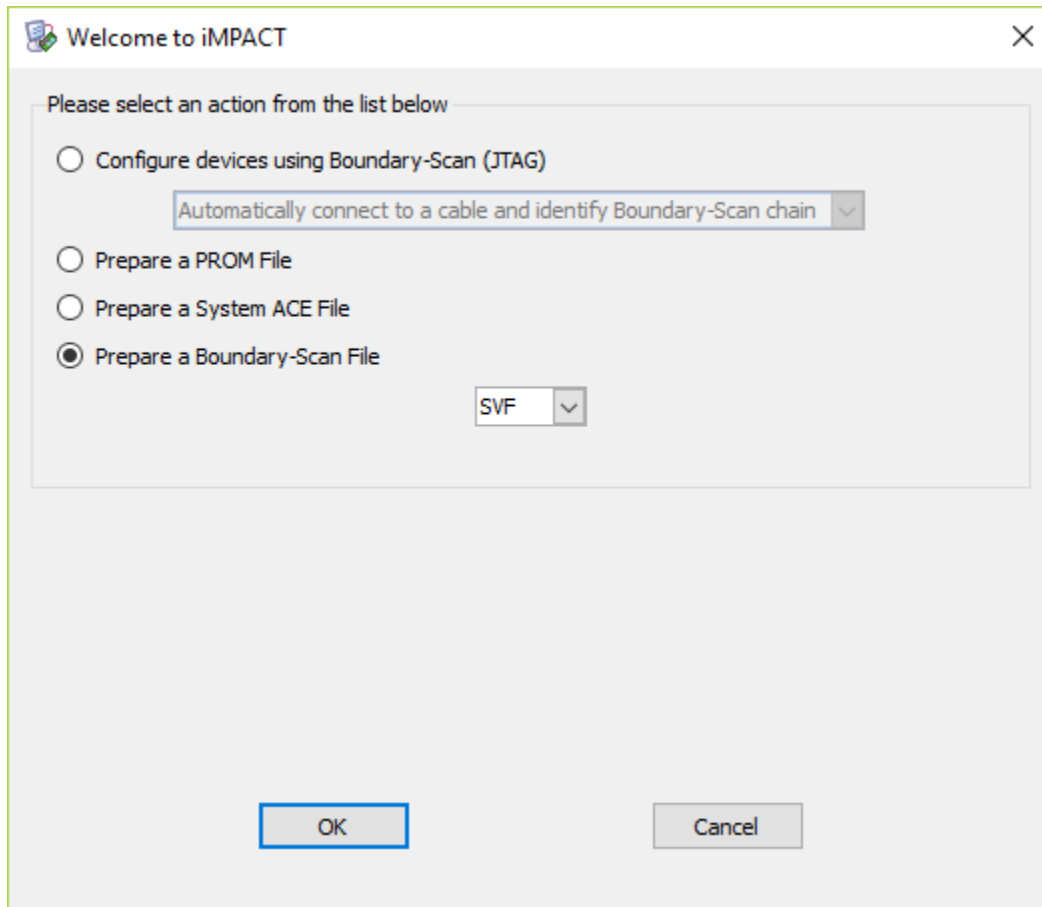
7. กด Generate File... เพื่อสร้างไฟล์ โดยจะขึ้นแสดงคำว่า Generate Succeeded เมื่อสำเร็จ



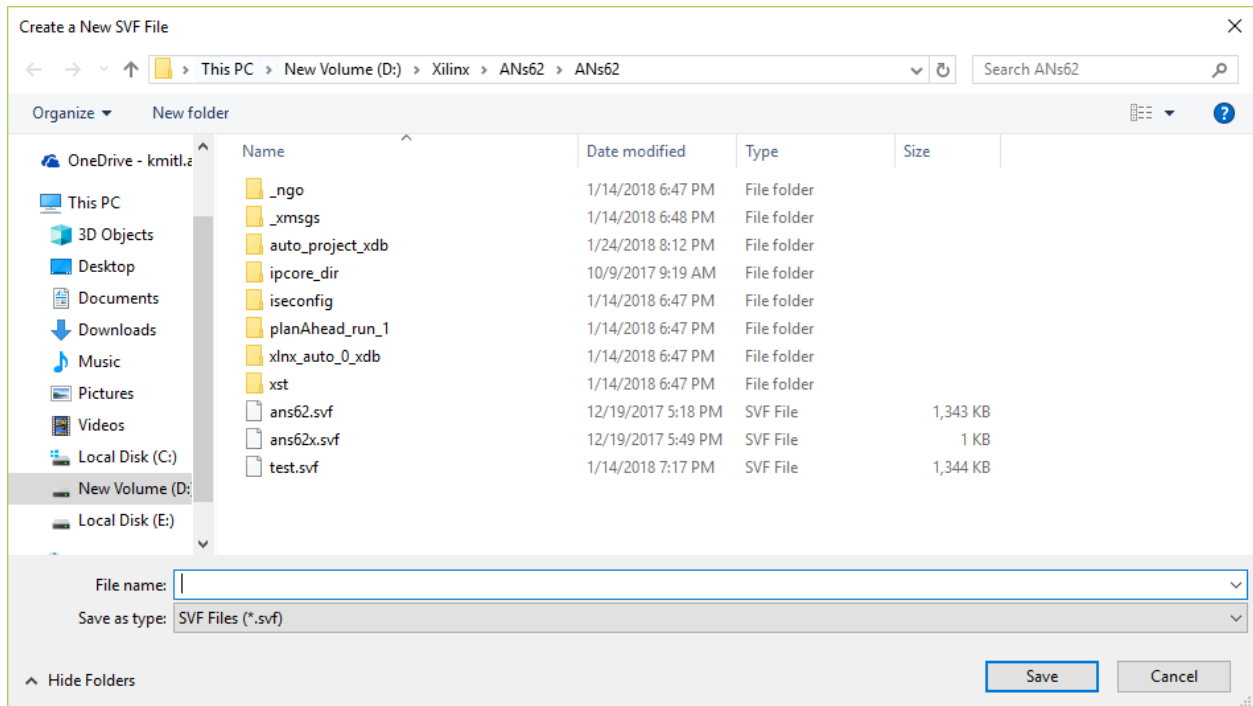
8. เลือกสร้างไฟล์ใหม่ ให้กด Yes ที่ Automatically create and save project



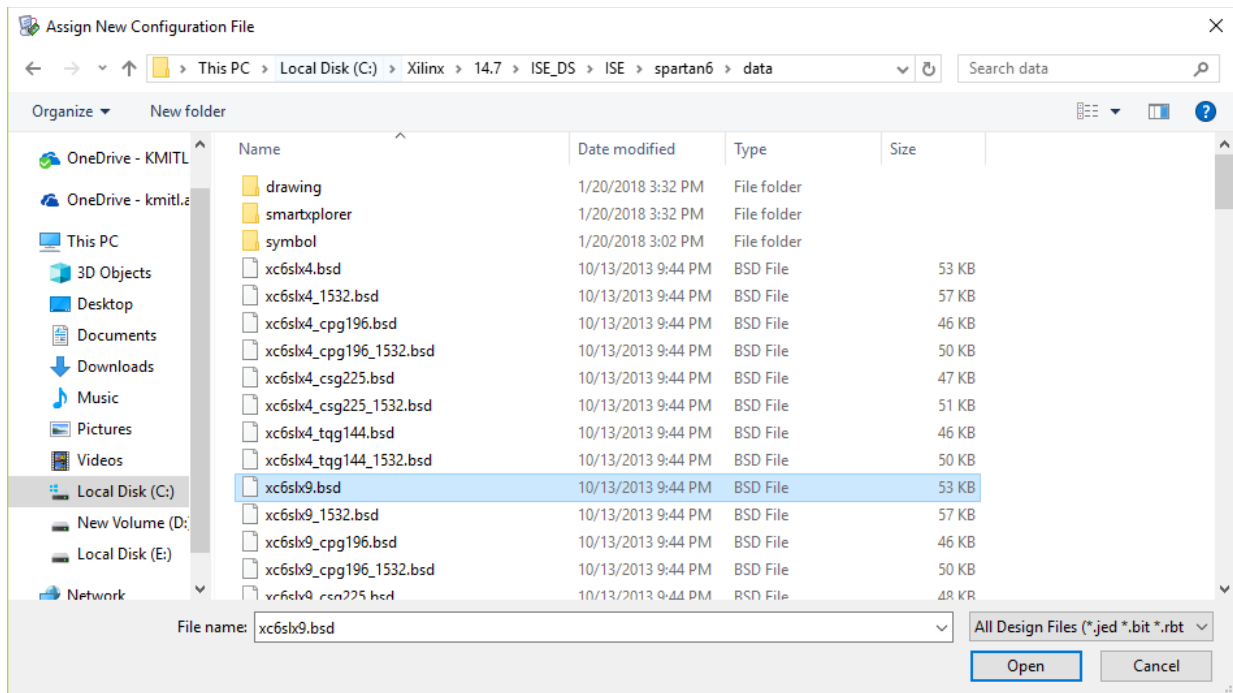
9. จะปรากฏหน้าต่างดังรูป เลือก Prepare a Boundary-Scan File แล้วเลือกประเภท SVF แล้วกด OK



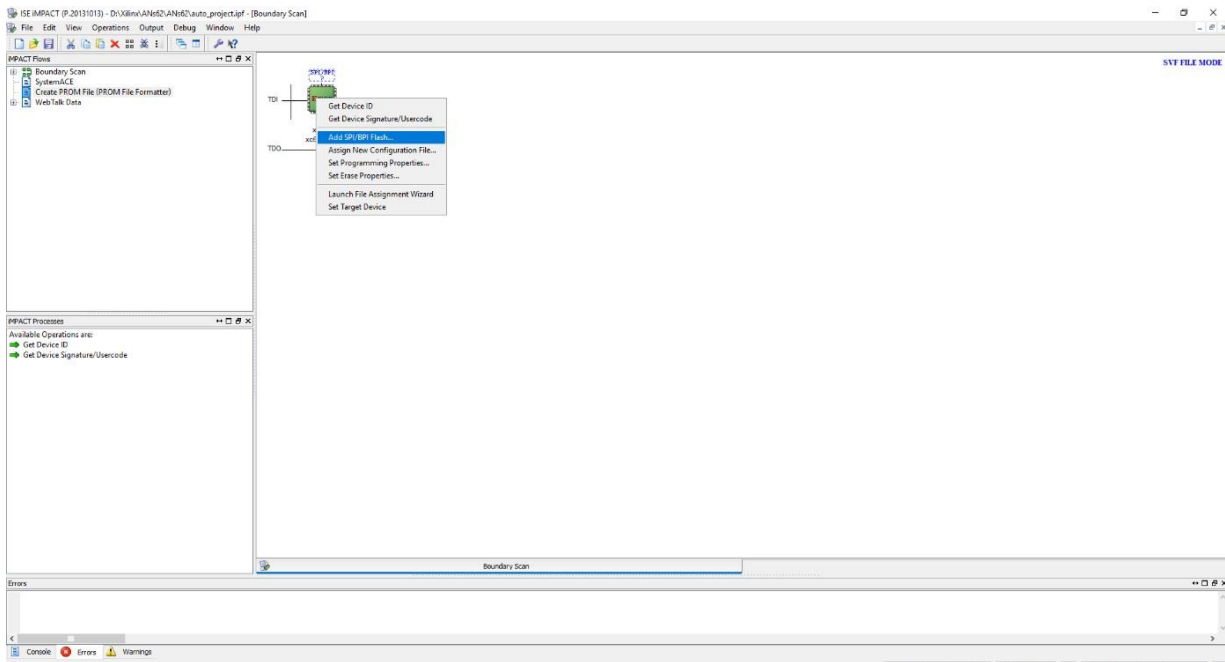
10. โปรแกรมจะให้เลือกพื้นที่สำหรับเซฟไฟล์ SVF



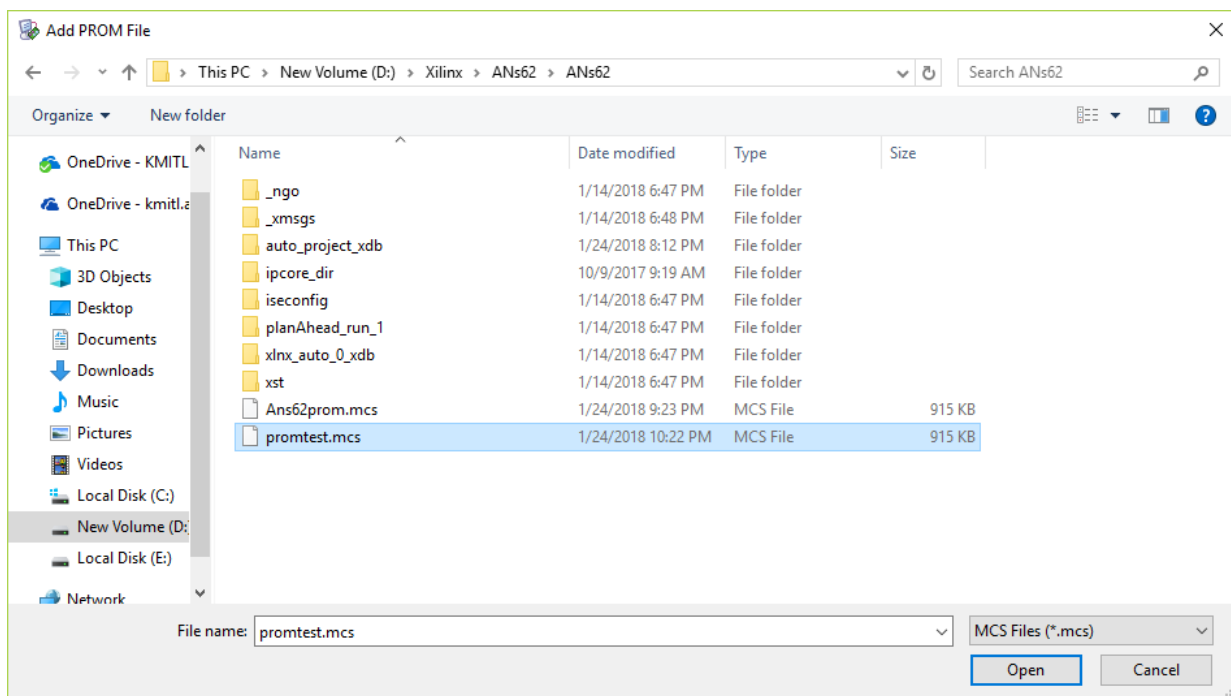
11. โปรแกรมจะถามหา configuration ของ FPGA ที่ใช้ โดยสามารถหาได้จากโฟลเดอร์ที่ชื่อ C:\Xilinx\14.7\ISE_DS\ISE\spartan6\data เลือก xc6slx9.bsd



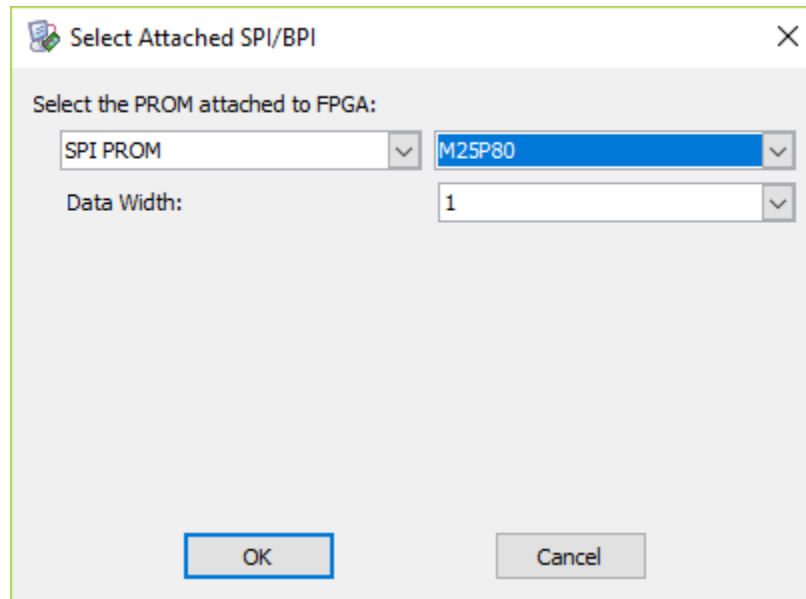
12. คลิกขวาที่รูปชิปแล้วเลือก ADD SPI/BPI Flash...



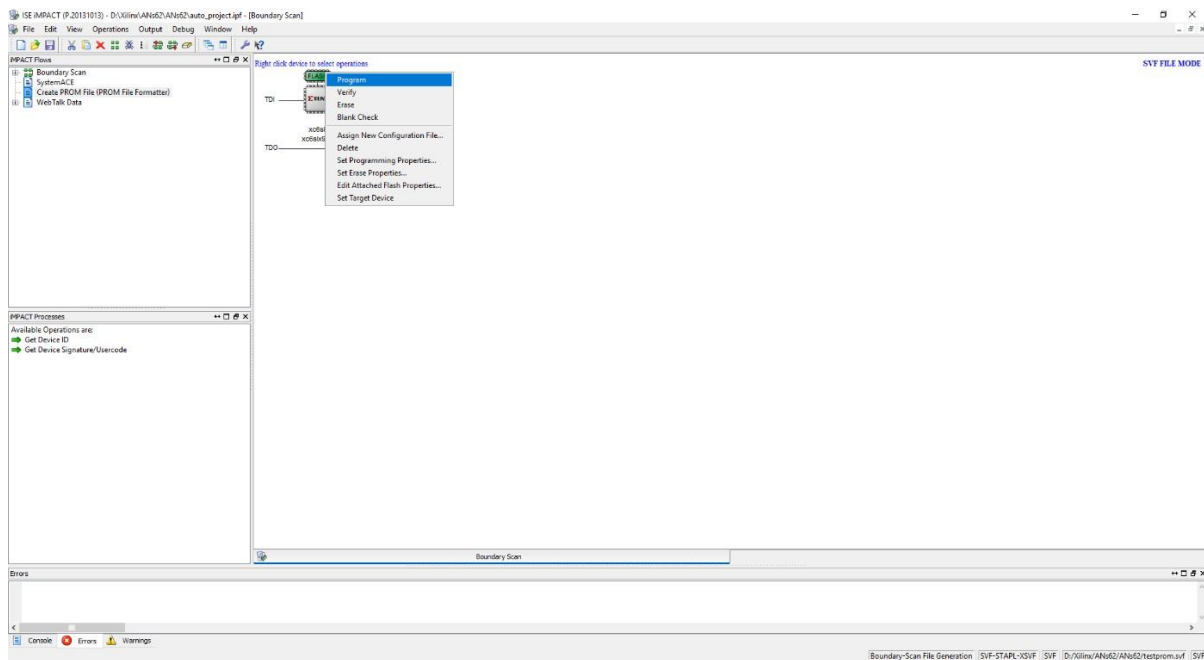
13. โปรแกรมจะถามหา PROM File ให้เลือกไฟล์ .MCS ที่ได้สร้างไว้ก่อนหน้านี้



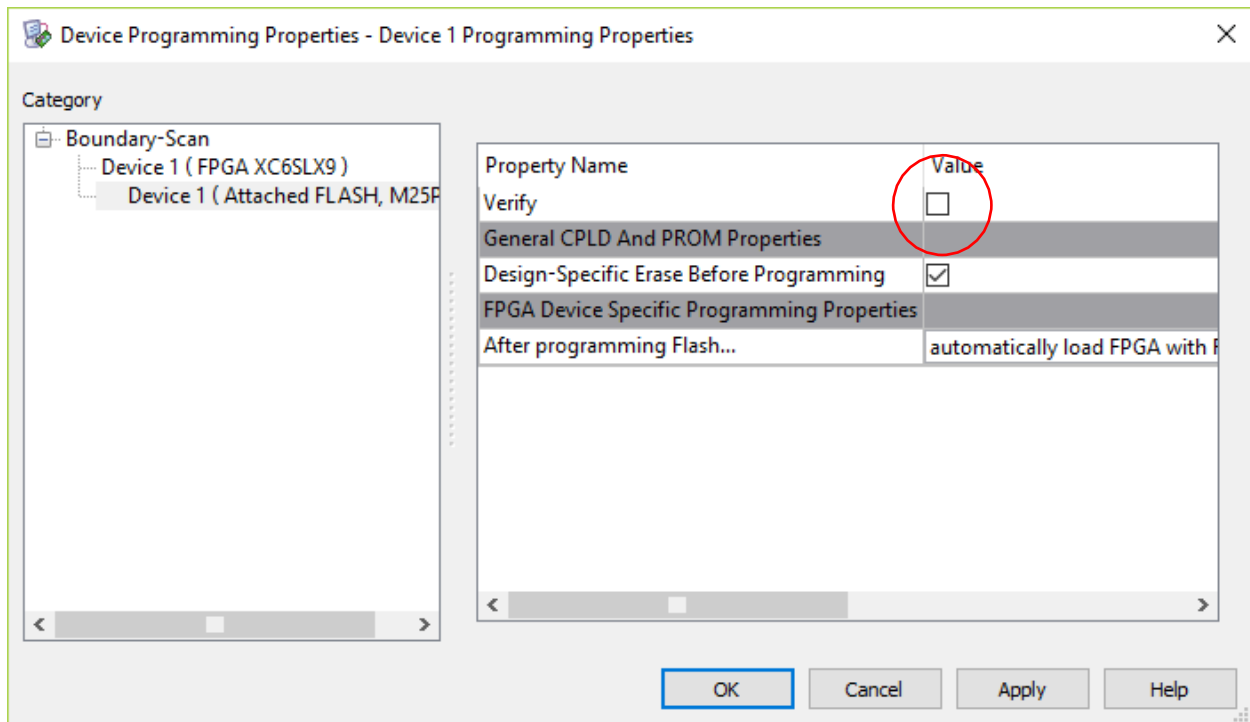
14. โปรแกรมจะถาม PROM ที่ใช้ ให้เลือก M25P80 สำหรับบอร์ด SERVEYOR-6



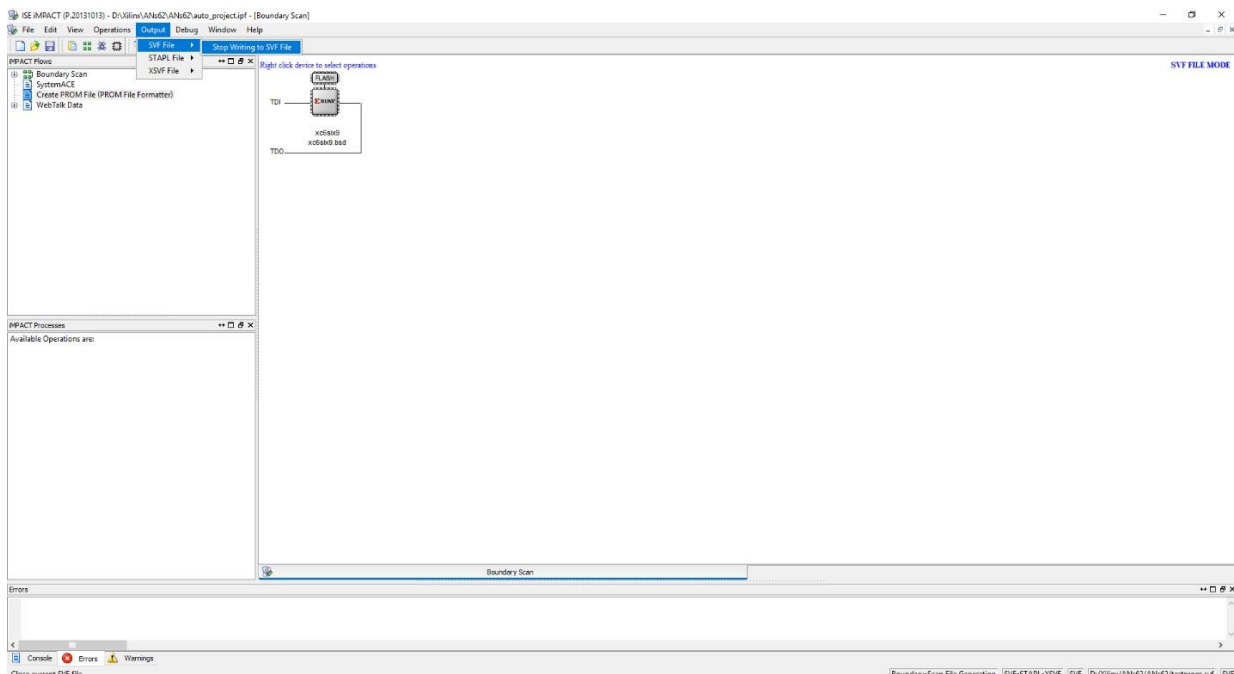
15. คลิกขวาที่สัญลักษณ์ Flash เลือก Program



16. ให้คลิกเครื่องหมายถูกของ Verify ออก แล้วกด OK โปรแกรมจะเริ่มสร้างไฟล์ SVF

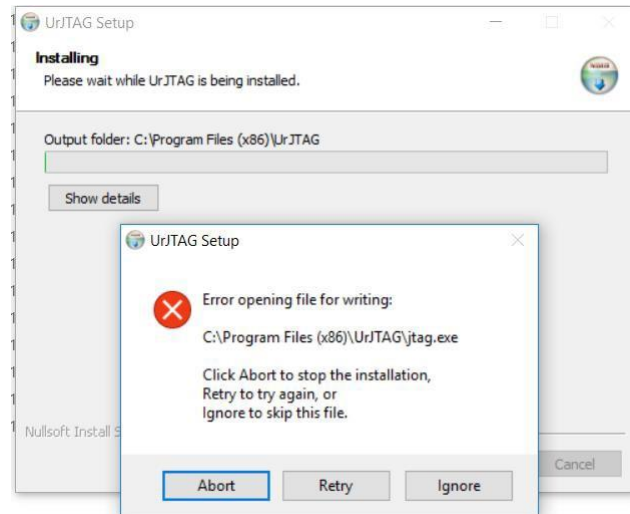


17. เมื่อสร้างไฟล์ SVF เสร็จไปที่ Output -> SVF File -> เลือก Stop Writing to SVF File เสร็จสิ้นการเตรียมไฟล์ SVF สำหรับโปรแกรมลง PROM ผ่าน SPI

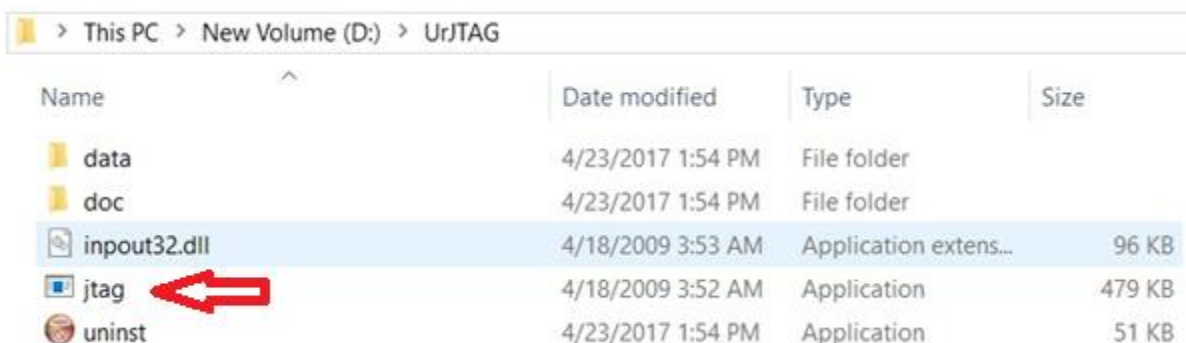


ขั้นตอนการติดตั้ง และใช้งานแรม UrJTAG

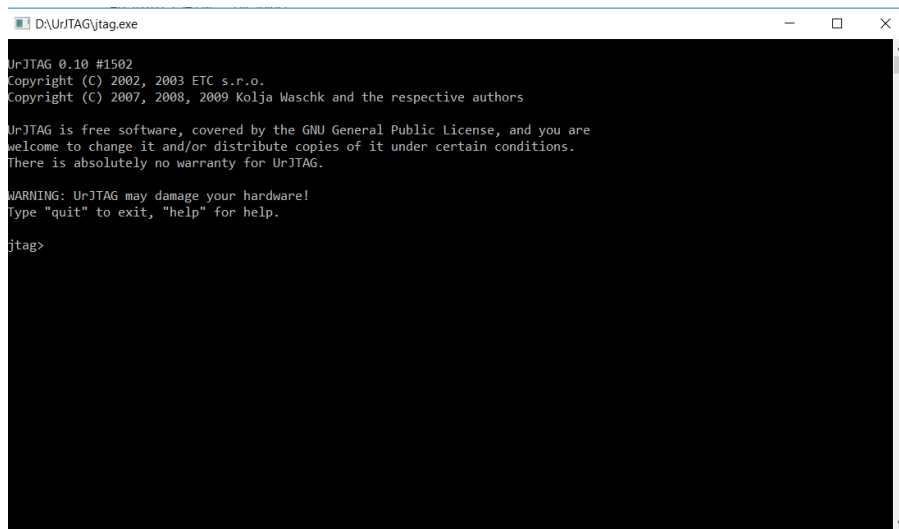
1. ติดตั้งโปรแกรมชื่อ UrJTAG-0.10 โดยเลือกไฟล์ urjtag-0.10.exe ในโฟลเดอร์ที่เก็บไฟล์นี้ไว้
 - ถ้าเกิด Error ดังรูปให้เปลี่ยน Directory ในการติดตั้ง โปรแกรม UrJTAG เช่น เปลี่ยนเป็น D:\ หรือ E:\



2. หลังจากติดตั้ง ให้เข้าไปที่โฟลเดอร์ UrJTAG ที่ติดตั้งจากข้อ 1 จากนั้นเข้าใช้โปรแกรมไฟล์ชื่อ jtag



จะได้หน้าต่างโปรแกรมดังนี้



```
D:\UrJTAG\jtag.exe
UrJTAG 0.10 #1502
Copyright (C) 2002, 2003 ETC s.r.o.
Copyright (C) 2007, 2008, 2009 Kolja Waschk and the respective authors

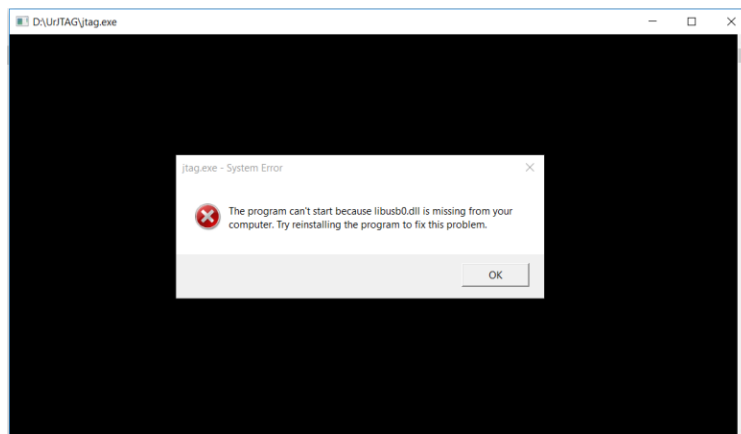
UrJTAG is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
There is absolutely no warranty for UrJTAG.

WARNING: UrJTAG may damage your hardware!
Type "quit" to exit, "help" for help.

jtag>
```

ถ้ามีการแจ้ง Error ดังภาพด้านล่าง ให้ปฏิบัติตามขั้นตอนต่อไปนี้

1. Copy ไฟล์ libusb0.dll ไปวางที่โฟลเดอร์ C:\Windows\SysWOW64 สำหรับ x64 หรือ c:\Windows\system32\ สำหรับ x86 แล้วทำการ restart โปรแกรม UrJTAG ใหม่



2. หากยังมีการแจ้ง Error ให้ Copy ไฟล์ FTD2XX.dll ไปวางที่โฟลเดอร์ C:\Windows\SysWOW64 สำหรับ x64 หรือ c:\Windows\system32\ สำหรับ x86 แล้วทำการ restart โปรแกรม UrJTAG ใหม่
3. เชื่อมต่อโมดูล FT232H กับคอมพิวเตอร์ด้วย USB
4. พิมพ์คำสั่ง cable jtagkey ตามด้วยการกด enter ในโปรแกรม UrJTAG เพื่อให้เชื่อมต่อ กับโมดูล ถ้าเชื่อมต่อได้สมบูรณ์จะแสดงผลพร้อมรูปด้านล่าง

```
D:\UrJTAG\jtag.exe
UrJTAG 0.10 #1502
Copyright (C) 2002, 2003 ETC s.r.o.
Copyright (C) 2007, 2008, 2009 Kolja Waschk and the respective authors

UrJTAG is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
There is absolutely no warranty for UrJTAG.

WARNING: UrJTAG may damage your hardware!
Type "quit" to exit, "help" for help.

jtag> cable jtagkey
Connected to libftd2xx driver.
jtag>
```

5. เชื่อมต่อโมดูล FT2232H กับ บอร์ด Xilinx
6. พิมพ์คำสั่ง detect ตามด้วยการกด enter เพื่อตรวจอุปกรณ์ที่สามารถโปรแกรมได้ ในกรณีเกิด Error กับ xcf01s ตามภาพ

```
D:\UrJTAG\jtag.exe
UrJTAG is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
There is absolutely no warranty for UrJTAG.

WARNING: UrJTAG may damage your hardware!
Type "quit" to exit, "help" for help.

jtag> cable jtagkey
Connected to libftd2xx driver.
jtag> detect
IR length: 14
Chain length: 2
Device Id: 00000001010000010100000010010011 (0x000000001414093)
Manufacturer: Xilinx
Part(0): xc3s200
Stepping: 0
Filename: d:\urjtag\data\xilinx\xc3s200\xc3s200
Device Id: 11110101000001000100000010010011 (0x00000000F5044093)
Manufacturer: Xilinx
Part(1): xcf01s
Unknown stepping!
chain.c(149) Part 1 without active instruction
chain.c(200) Part 1 without active instruction
chain.c(149) Part 1 without active instruction
jtag>
```

ให้แก้ไขดังนี้

- หาไฟล์นามสกุล bsd ชื่อตามอุปกรณ์ที่ต้องการในโฟลเดอร์

C:\Xilinx\14.7\ISE_DS\ISE\???\data\???.bsd

ตัวอย่างเช่นต้องการไฟล์ของ xcf01s จะหาได้ที่โฟลเดอร์

C:\Xilinx\14.7\ISE_DS\ISE\xcf\data\xcf01s.bsd

- copy และนำไฟล์ที่ได้ไปวางไว้ในโฟลเดอร์ที่สร้างขึ้นเอง เช่น โฟลเดอร์ชื่อ BSD ในไดรฟ์ D

> This PC > New Volume (D:) > bsd				
Name	Date modified	Type	Size	
xcf01s.bsd	10/13/2013 9:44 PM	BSD File	10 KB	

- พิมพ์คำสั่ง `bsdl path <Directory>` ตามด้วย enter จากนั้นพิมพ์คำสั่ง `detect` แล้วกด enter จะได้ผลลัพธ์ตามรูปตัวอย่างด้านล่าง
หมายเหตุ <Directory> ให้แทนด้วย ชื่อDirectory ที่เก็บไฟล์ xcf01s.bsd ไว้ โดยในตัวอย่างเก็บไว้ที่ D:\bsd ก็จะมีคำสั่งเป็น `bsdl path d:\bsd`

```
jtag> bsd path d:\bsd
jtag> detect
IR length: 14
Chain length: 2
Device Id: 00000001010000010100000010010011 (0x000000001414093)
  Manufacturer: Xilinx
  Part(0):      xc3s200
  Stepping:     0
  Filename:     d:\urjtag\data\xilinx\xc3s200\xc3s200
Device Id: 11110101000001000100000010010011 (0x00000000F5044093)
  Filename:     d:\bsd\xcf01s.bsd
jtag>
```

7. พิมพ์คำสั่ง `print chain` ตามด้วยการกด enter จะเป็นการแสดง JTAG chain

```
jtag> print chain
(140) String conversion failed!
No.
-----
  0 Xilinx      xc3s200      0      BYPASS      BR
  1             XCF01S             BYPASS      BYPASS
jtag>
```


8. การโปรแกรมลงแฟรช สำหรับบอร์ด SURVEYOR 3 และ SURVEYOR 6

- Download file Test.svf จาก

<https://drive.google.com/drive/folders/1MZyRACJDneNtqHNfVotfy4cPsq7lJq8s>

- พิมพ์คำสั่ง part 0 ตามด้วยการกด enter เพื่อเลือกเป้าหมายเป็นชิป Xilinx สำหรับการโปรแกรมลง Flash
- พิมพ์คำสั่ง svf <directory ไฟล์สกุล svf สำหรับโปรแกรมแฟรช> progress stop ตามด้วยการกด enter เช่น svf d:\svf\Test.svf progress stop
สำหรับสั่งโปรแกรมสู่เป้าหมาย
- พิมพ์คำสั่ง instruction BYPASS ตามด้วยการกด enter สำหรับเตรียม register ในการโปรแกรมครั้งต่อไป

```
jtag> part 0
jtag> svf d:\svf\lab9flash.svf progress stop
Parsing 4390/4397 ( 99%)
Scanned device output matched expected TDO values.
jtag> instruction BYPASS
```

9. การโปรแกรมลง PROM สำหรับบอร์ด SURVEYOR 3

- Download file Test_PROM.svf จาก

<https://drive.google.com/drive/folders/1MZyRACJDneNtqHNfVotfy4cPsq7lJq8s>

- พิมพ์คำสั่ง part 1 ตามด้วยการกด enter เพื่อเลือก xcf01s เป็นเป้าหมาย
- พิมพ์คำสั่ง svf <directory ไฟล์สกุล svf สำหรับโปรแกรมPROM> progress stop ตามด้วยการกด enter สำหรับสั่งโปรแกรมสู่เป้าหมาย
เช่น svf d:\svf\Test_PROM.svf progress stop
- พิมพ์คำสั่ง instruction BYPASS ตามด้วยการกด enter สำหรับเตรียม register ในการโปรแกรมครั้งต่อไป

```
jtag> part 1
jtag> svf d:\svf\lab9prom.svf progress stop
Parsing 17020/17025 ( 99%)
Scanned device output matched expected TDO values.
jtag> instruction BYPASS
jtag>
```

10. การโปรแกรมลง PROM ผ่าน SPI สำหรับบอร์ด Spartan 6

- พิมพ์คำสั่ง svf <directory> ไฟล์สกุล svf สำหรับโปรแกรม PROM> progress stop ตามด้วยการกด enter สำหรับสั่งโปรแกรมสู่เป้าหมาย
- พิมพ์คำสั่ง instruction BYPASS ตามด้วยการกด enter สำหรับเตรียม register ในการโปรแกรมครั้งต่อไป

```
jtag> svf d:\svf\testprom.svf progress stop
Parsing 145720/145727 ( 99%)
Scanned device output matched expected TDO values.
```

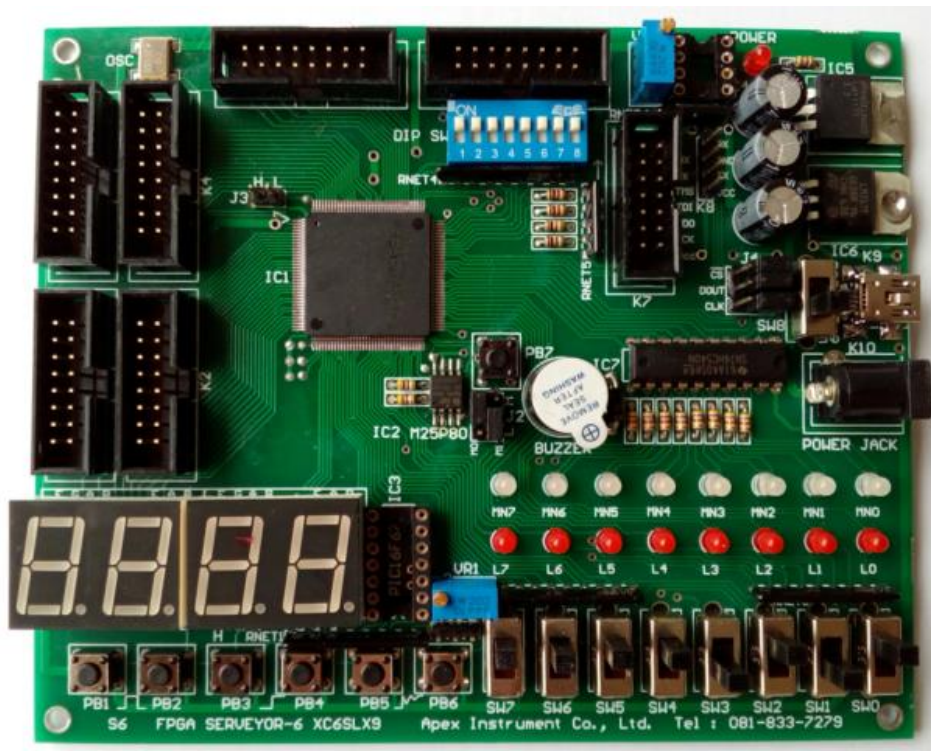
หมายเหตุ การโปรแกรมลง PROM ผ่าน SPI

1. การโปรแกรมใช้เวลานาน
2. ไฟล์ SVF ที่ใช้ต้องสร้างมาจากหัวข้อขั้นตอนการเตรียมไฟล์ SVF สำหรับโปรแกรม PROM ผ่าน SPI เท่านั้น

อ้างอิง

1. แหล่งที่มา http://www.urjtag.org/book/jtag_commands.html
2. แหล่งที่มา <https://www.youtube.com/watch?v=Sp02GEPfooY>

บอร์ด FPGA Surveyor-6 XC6SLX9



ชุดทดลองบอร์ด FPGA ยี่ห้อ เอเพก อินสตรูเมนต์ รุ่น FPGA Surveyor-6 XC6SLX9

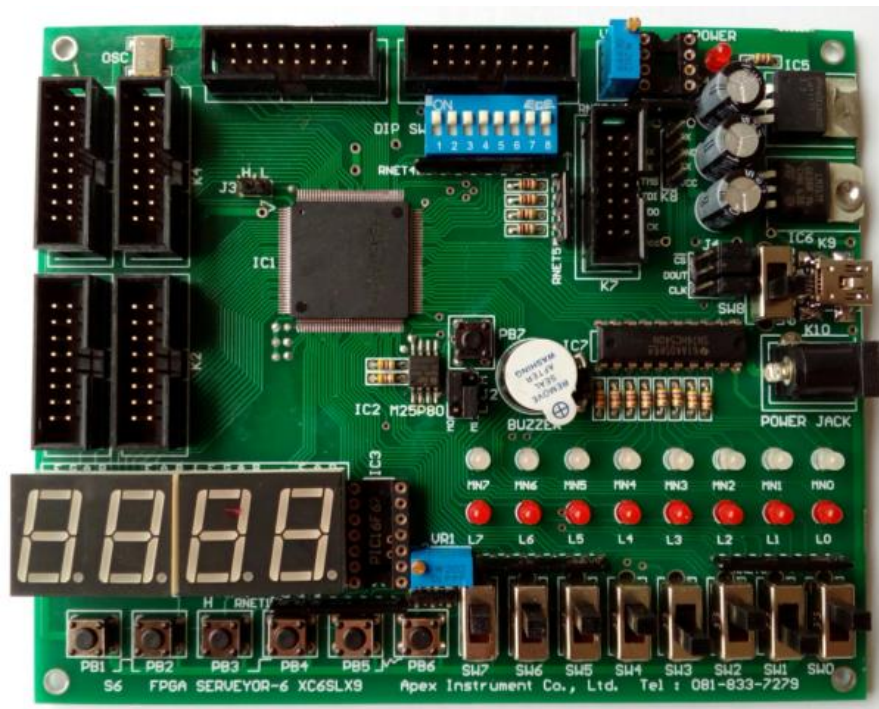
บอร์ด FPGA Surveyor-6 XC6SLX9 พร้อมสายดาวน์โหลด JTAG แบบ USB ประกอบด้วย

- FPGA : Spartan-6 ของ Xilinx เบอร์ XC6SLX9 144pin พร้อม Flash PROM ภายนอก
- 7-Segment 4 หลัก
- LED 3 สถานะ 8 ดวง
- LED 2 สถานะ 8 ดวง
- Logic Switch (Slide Switch) 8 ตัว (มี SW7 ใช้ input ร่วมกับ PB6)
- 6 Expansion port (48 Bits)
- ปุ่มกดแบบ One-Shot, Bounceless และ Bounce อย่างละ 2 ตัว
- DIP Switch 8 ดวง ร่วมกับ Expansion port K6
- Buzzer จำนวน 1 ตัว
- RS-232C 1 Port
- 20 Mhz Oscillator
- Clock generator : 1-100Hz และ 10-1KHz
- สายดาวน์โหลด JTAG แบบ USB
- อะแดปเตอร์ 9VDC output / 220VAC input

คู่มือการใช้งานบอร์ด FPGA Surveyor-6 XC6SLX9 ยี่ห้อ เอเพก อินสตรูเมนต์

บอร์ด FPGA Surveyor-6 XC6SLX9 พร้อมสายดาวน์โหลด JTAG แบบ USB ประกอบด้วย

- FPGA : Spartan-6 ของ Xilinx เบอร์ FPGA เบอร์ XC6SLX9-2TQG144C
- และ PROM เบอร์ M25P80-VMW6TG
- 7-Segment 4 หลัก
- LED 3 สถานะ 8 ดวง
- LED 2 สถานะ 8 ดวง
- Logic Switch (Slide Switch) 8 ตัว (มี SW7 ใช้ input ร่วมกับ PB6)
- 6 Expansion port (48 Bits)
- ปุ่มกดแบบ One-Shot, Bounceless และ Bounce อย่างละ 2 ตัว
- DIP Switch 8 ดวง ร่วมกับ Expansion port K6
- Buzzer จำนวน 1 ตัว
- RS-232C 1 Port
- 20 Mhz Oscillator
- Clock generator : 1–100Hz และ 10-1KHz
- สายดาวน์โหลด JTAG แบบ USB
- อะแดปเตอร์ 9VDC output / 220VAC input



รูปที่ 1 บอร์ด FPGA Surveyor-6 XC6SLX90

FPGA เบอร์ XC6SLX9-2TQG144C

Flash PROM เบอร์ M25P80-VMW6TG

รายการ Pin List

K1 CONNECTOR					
Descriptions	FPGA Pinout	K1 Pinout	K1 Pinout	FPGA	Descriptions
GND	-	16	15	P5	I/O
GND	-	14	13	P7	I/O
GND	-	12	11	P9	I/O
GND	-	10	9	p11	I/O
GND	-	8	7	p14	I/O
GND	-	6	5	P16	I/O
GND	-	4	3	P21	I/O
+3.3V	-	2	1	P23	I/O

K2 CONNECTOR					
Descriptions	FPGA Pinout	K2 Pinout	K2 Pinout	FPGA	Descriptions
GND	-	16	15	P6	I/O
GND	-	14	13	P8	I/O
GND	-	12	11	P10	I/O
GND	-	10	9	p12	I/O
GND	-	8	7	p15	I/O
GND	-	6	5	P17	I/O
GND	-	4	3	P22	I/O
+3.3V	-	2	1	P24	I/O

K3 CONNECTOR					
Descriptions	FPGA Pinout	K3 Pinout	K3 Pinout	FPGA	Descriptions
GND	-	16	15	P124	I/O
GND	-	14	13	P127	I/O
GND	-	12	11	P132	I/O
GND	-	10	9	p134	I/O
GND	-	8	7	p138	I/O
GND	-	6	5	P140	I/O
GND	-	4	3	P142	I/O
+3.3V	-	2	1	P1	I/O

K4 CONNECTOR					
Descriptions	FPGA Pinout	K4 Pinout	K4 Pinout	FPGA	Descriptions
GND	-	16	15	P126	I/O
GND	-	14	13	P131	I/O
GND	-	12	11	P133	I/O
GND	-	10	9	p137	I/O
GND	-	8	7	p139	I/O
GND	-	6	5	P141	I/O
GND	-	4	3	P143	I/O
+3.3V	-	2	1	P2	I/O

K5 CONNECTOR					
Descriptions	FPGA Pinout	K5 Pinout	K5 Pinout	FPGA	Descriptions
GND	-	16	15	P114	I/O
GND	-	14	13	P115	I/O
GND	-	12	11	P116	I/O
GND	-	10	9	p117	I/O
GND	-	8	7	p118	I/O
GND	-	6	5	P119	I/O
GND	-	4	3	P120	I/O
GND	-	2	1	P121	I/O

K6 CONNECTOR					
Descriptions	FPGA Pinout	K6 Pinout	K6 Pinout	FPGA	Descriptions
GND	-	16	15	P99	I/O
GND	-	14	13	P100	I/O
GND	-	12	11	p101	I/O
GND	-	10	9	p102	I/O
GND	-	8	7	P104	I/O
GND	-	6	5	P105	I/O
GND	-	4	3	P111	I/O
GND	-	2	1	P112	I/O

DIP SWITCH	
Descriptions	FPGA Pinout
DIP1	P112
DIP2	P111
DIP3	p105
DIP4	p104
DIP5	P102
DIP6	P101
DIP7	P100
DIP8	P99

OSC	
Descriptions	FPGA Pinout
OSC	P123

RS232	
Descriptions	FPGA Pinout
TX	P98
RX	P97

SLIDE SWITCH	
Descriptions	FPGA Pinout
SW7(PB6)	P55
SW6	P56
SW5	P57
SW4	P58
SW3	P59
SW2	P61
SW1	P62
SW0	P66

BUZZER	
Descriptions	FPGA Pinout
BUZZER	P83

FLASH PROM	
Descriptions	FPGA Pinout
MOSI->D	P64
MISO(DIN)->Q	P65
CSO_B->S'	P38
CCLK->C	P70

7-SEGMENT	
Descriptions	FPGA Pinout
a	P41
b	P40
c	P35
d	P34
e	P32
f	P29
g	P27
p	P26
COMMON3	P30
COMMON2	P33
COMMON1	P43
COMMON0	P44

LED	
Descriptions	FPGA Pinout
L7	P67
L6	P74
L5	P75
L4	P78
L3	P79
L2	P80
L1	P81
L0	P82

LOGIC MONITOR	
Descriptions	FPGA Pinout
MN7	P84
MN6	P85
MN5	P87
MN4	P88
MN3	P92
MN2	P93
MN1	P94
MN0	P95

PUSH BUTTON SWITCH	
Descriptions	FPGA Pinout
PB1	P45
PB2	P46
PB3	P47
PB4	P48
PB5	P51
PB6(SW7)	P55
VRCLK	P50