

**การทดลองที่ 0 การใช้งานบอร์ด FPGA การใช้งาน Logic Trainer การใช้งาน Logic probe และการใช้งาน Multimeter  
วัตถุประสงค์**

1. เพื่อให้นักศึกษาติดตั้งและฝึกการใช้งาน FPGA
2. เพื่อให้นักศึกษาทำความเข้าใจและฝึกใช้งาน Logic Trainer
3. เพื่อให้นักศึกษาทำความเข้าใจและฝึกใช้งาน Logic probe
4. เพื่อให้นักศึกษาทำความเข้าใจและฝึกใช้งาน Multimeter

หมายเหตุ ให้อ่านเอกสารการทดลองและเอกสารประกอบให้ครบก่อนเริ่มทำการทดลอง

**การใช้งานบอร์ด FPGA**

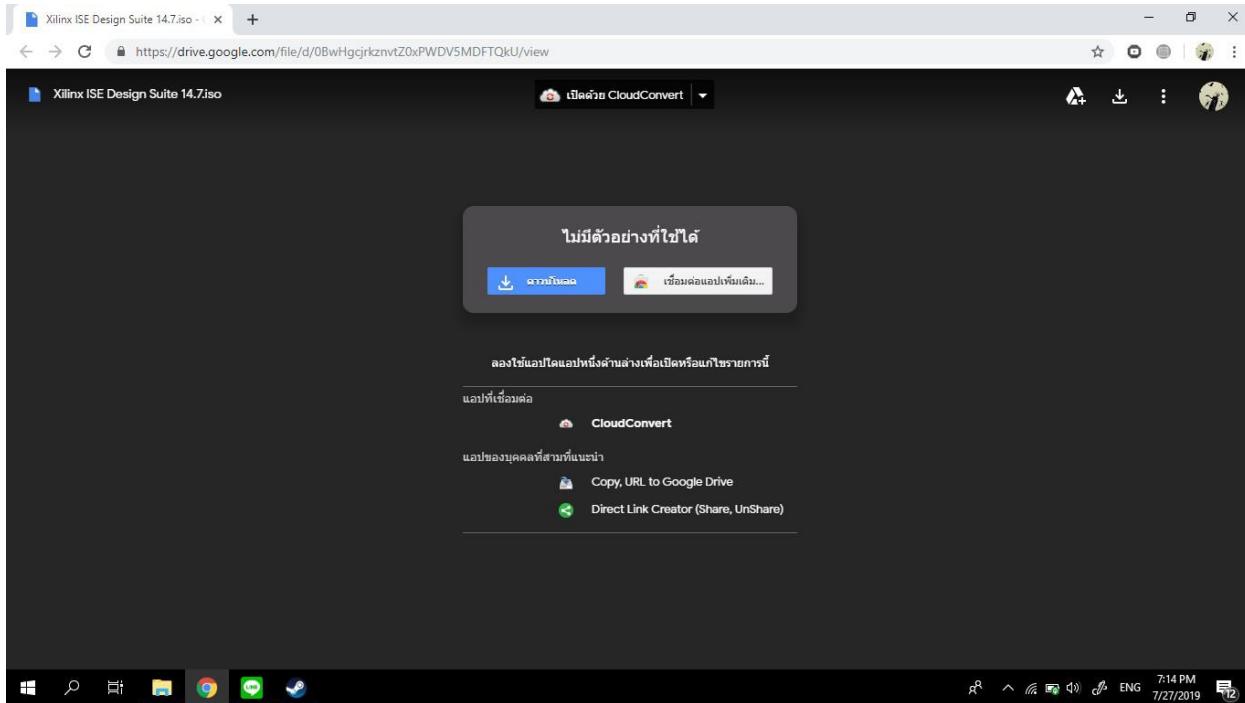
การทดลองนี้ นักศึกษาจะได้รู้จักการออกแบบวงจรด้วยการวาดรูป (Schematic Circuit) และบันทึกโปรแกรมลงในชิป Field Programmable Gate Array (FPGA)

**โปรแกรม ISE Xilinx WebPACK**

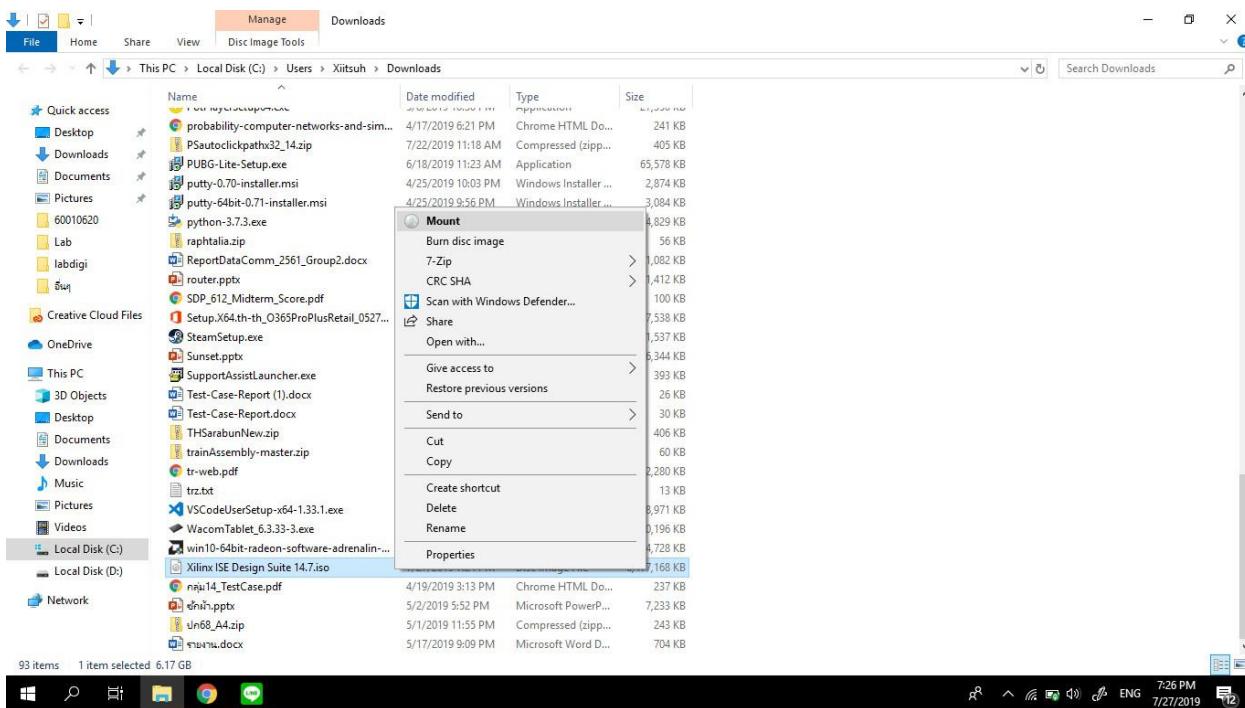
นักศึกษาที่มีหนังสือ “ออกแบบไอซีดิจิตอลด้วย FPGA และ CPLD ภาคปฏิบัติ โดยใช้วิธี Schematic ซอฟต์แวร์ทุล ISE WebPACK” ให้อ่านหนังสือก่อนการทดลองในบทที่ 1 และ 2 ก่อน

# การติดตั้งโปรแกรม ISE Xilinx WebPACK

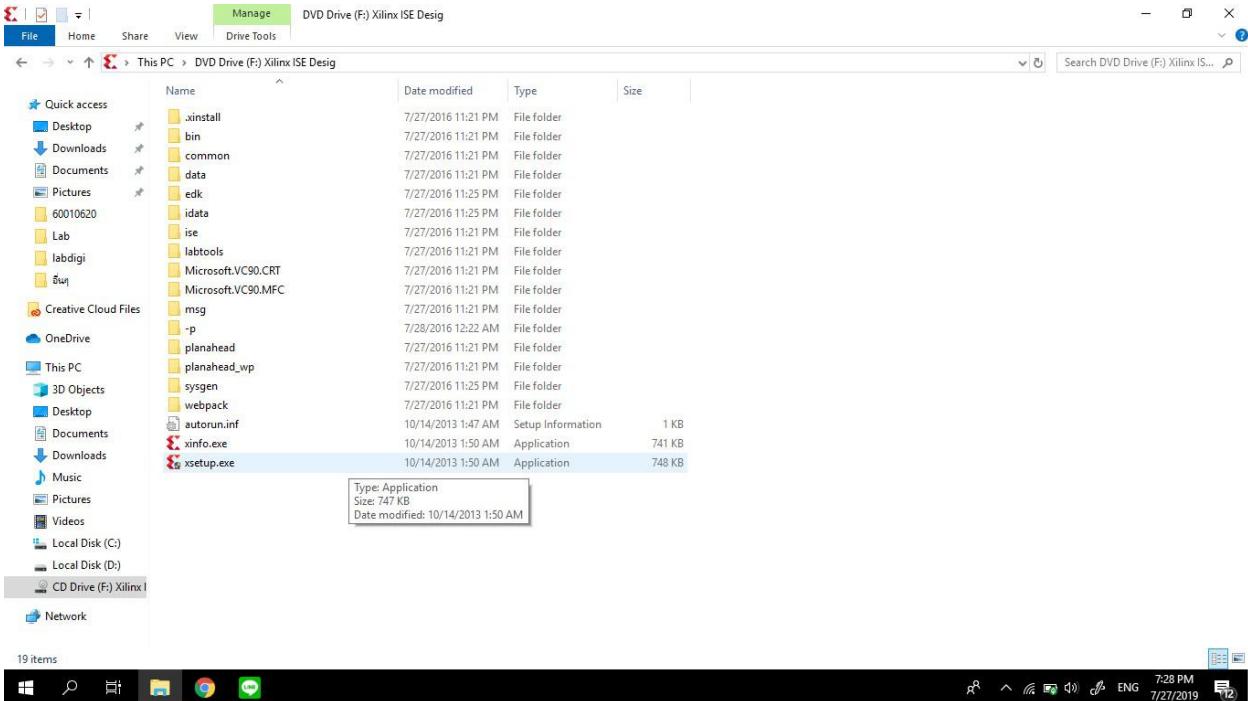
1.download โปรแกรมจาก <https://drive.google.com/file/d/0BwHgcrkznvtZ0xPWDV5MDFTQkU/view>



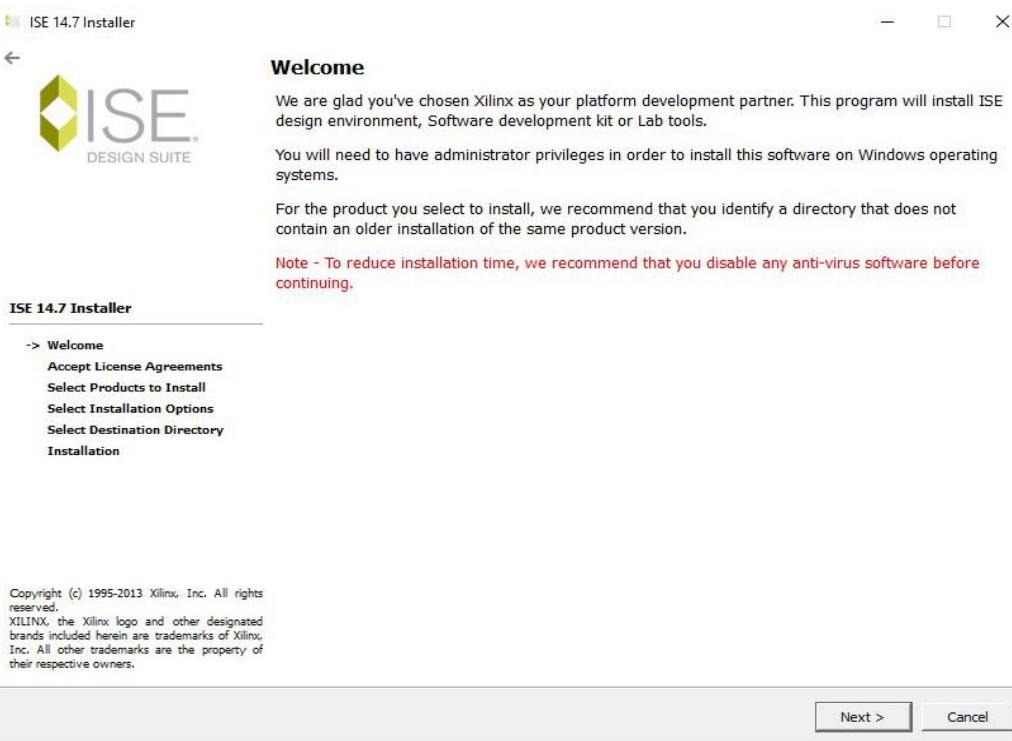
2.คลิกขวาที่ไฟล์ที่download จากนั้นกดคำว่า Mount



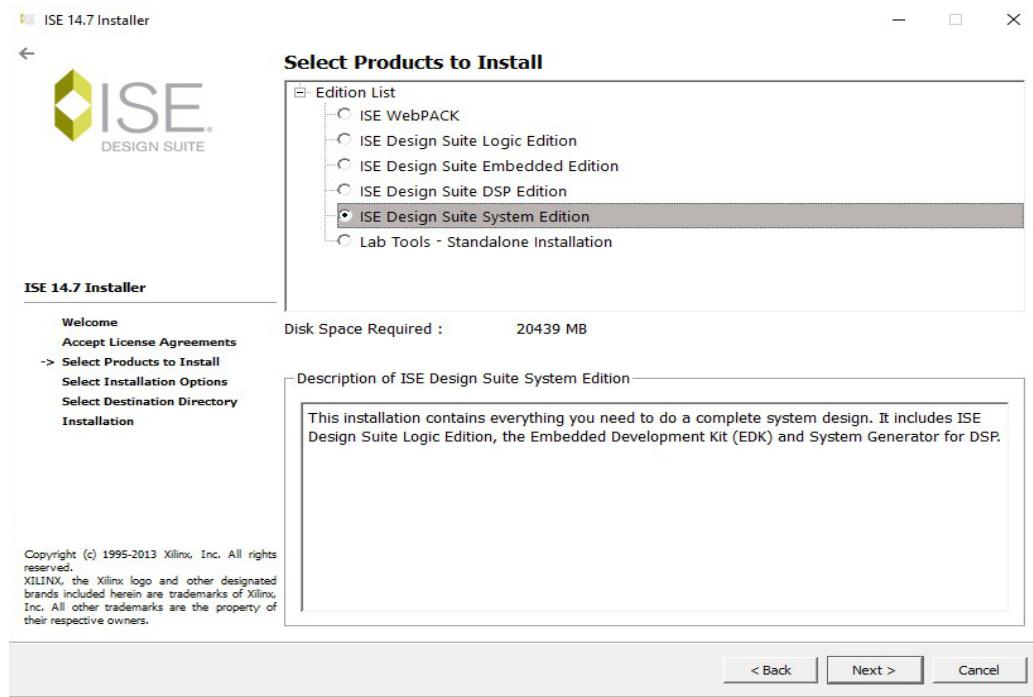
### 3. เมื่อคลิก Mount เสร็จจะขึ้นหน้าหน้าไปทั้งหมดที่ xsetup.exe



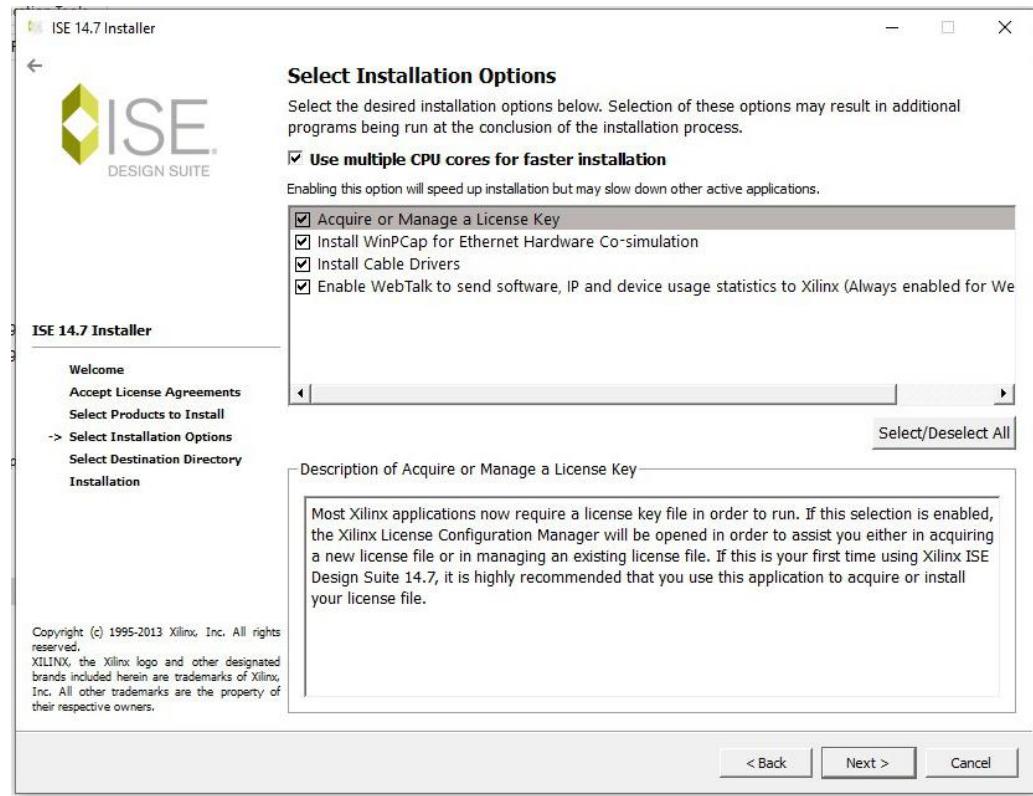
### 4. ให้กด Next ไปเรื่อยๆ



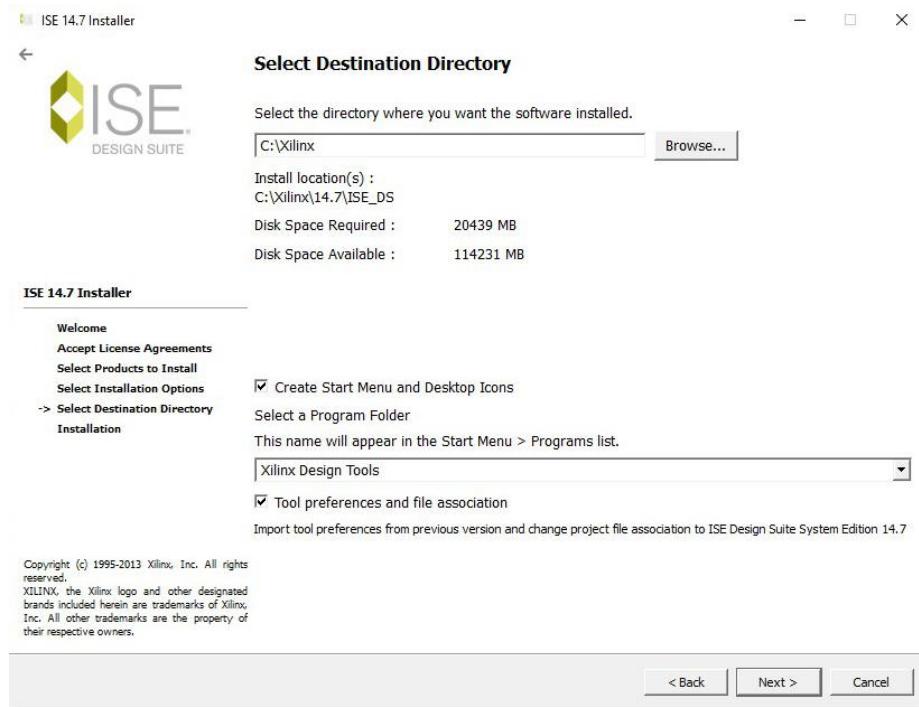
5. เมื่อถึงหน้านี้ให้เลือก ISE Design Suite System Edition จากนั้นกด Next



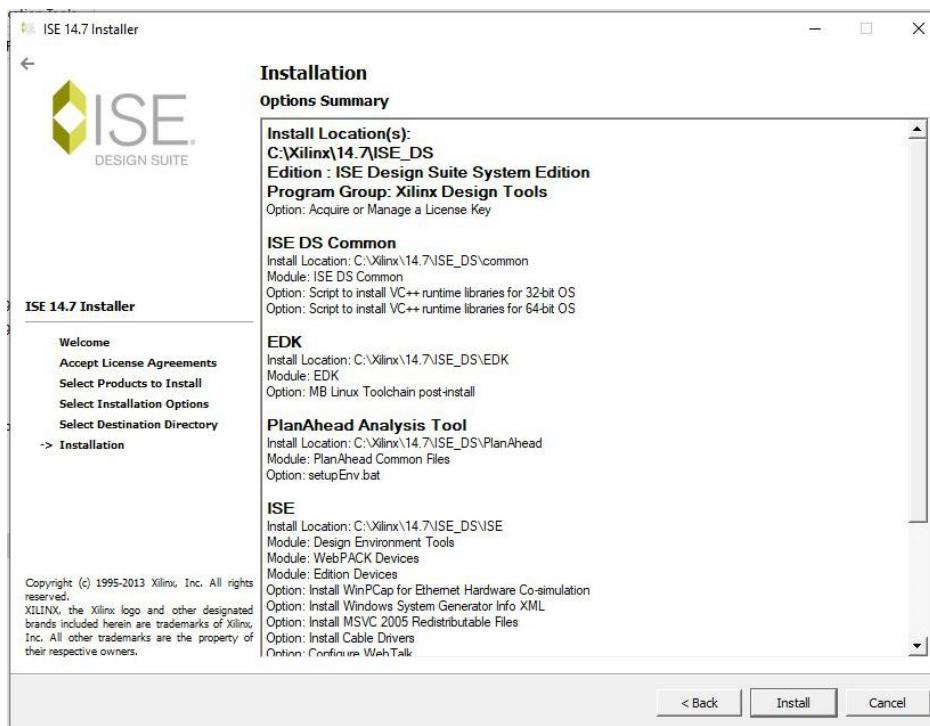
6. เมื่อถึงหน้านี้ให้กดเลือกทั้งหมด จากนั้นกด Next



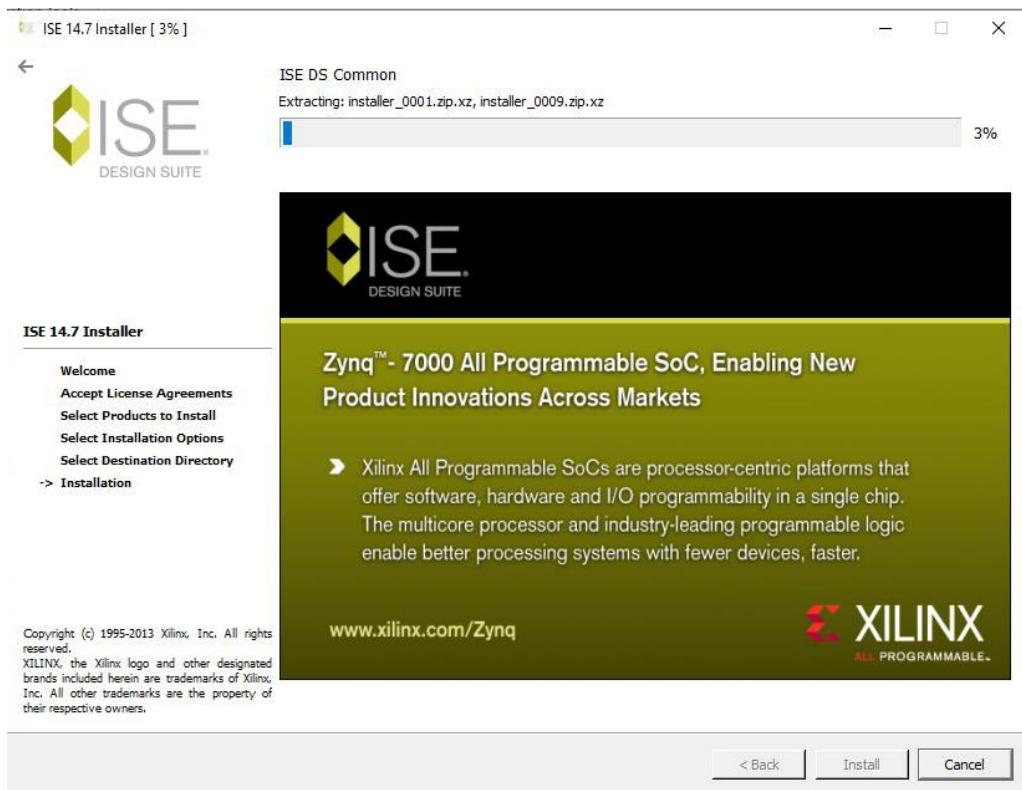
## 7.เลือกโฟลเดอร์ที่จะติดตั้ง จากนั้นกด Next



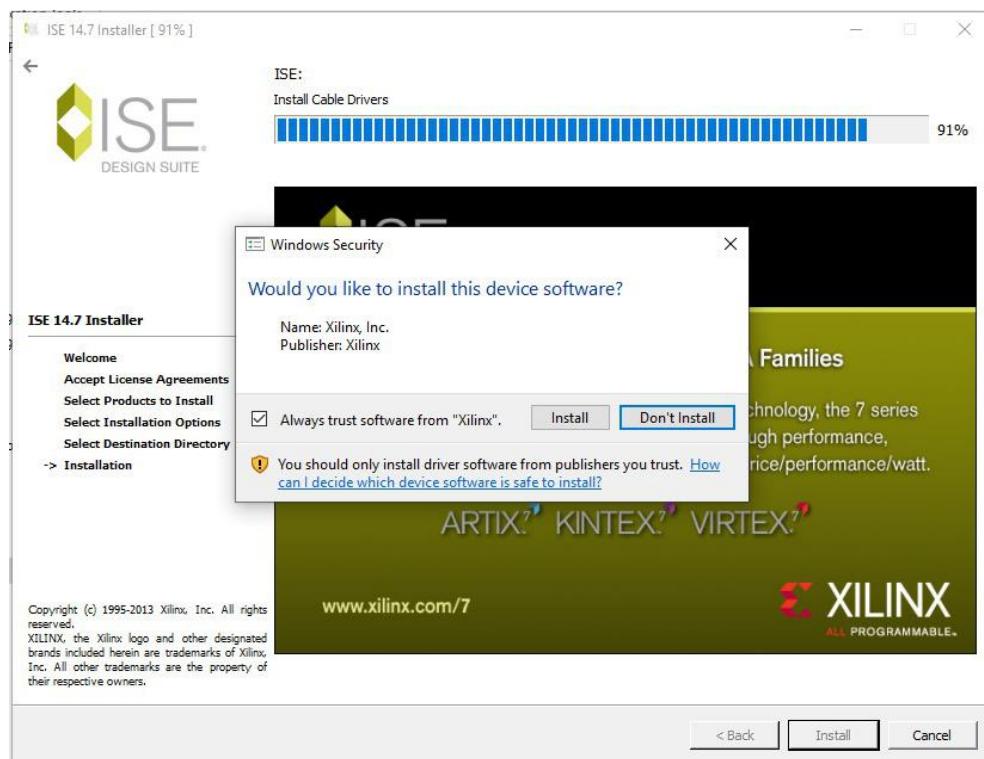
## 8.เมื่อถึงหน้านี้ให้กด Install



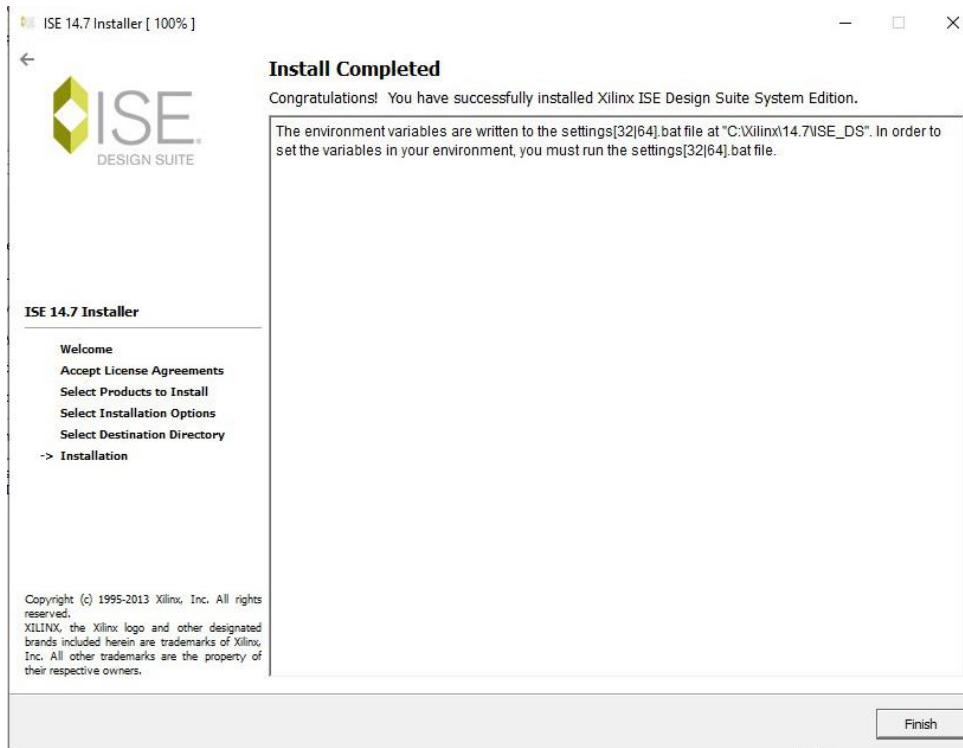
## 9. เมื่อกด Install แล้วหน้าจอจะแสดงผลตั้งภาพ ให้รอจนกว่าจะครบ 100%



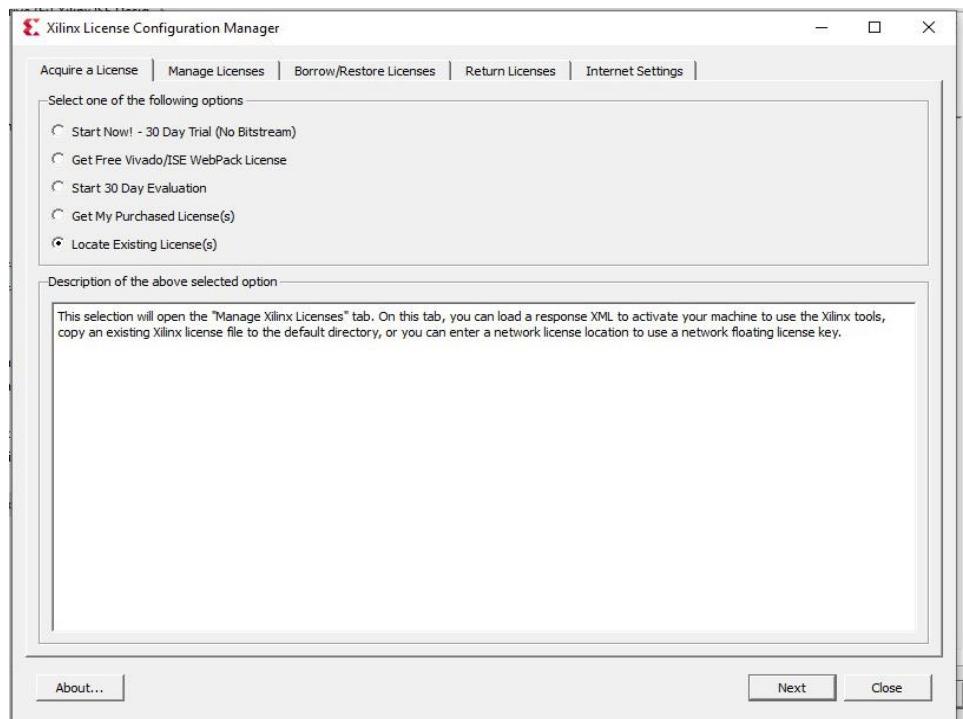
## 10. หากมีแจ้งเตือนดังรูปให้กด Install



## 11. เมื่อเสร็จสิ้นจะขึ้นดังรูป

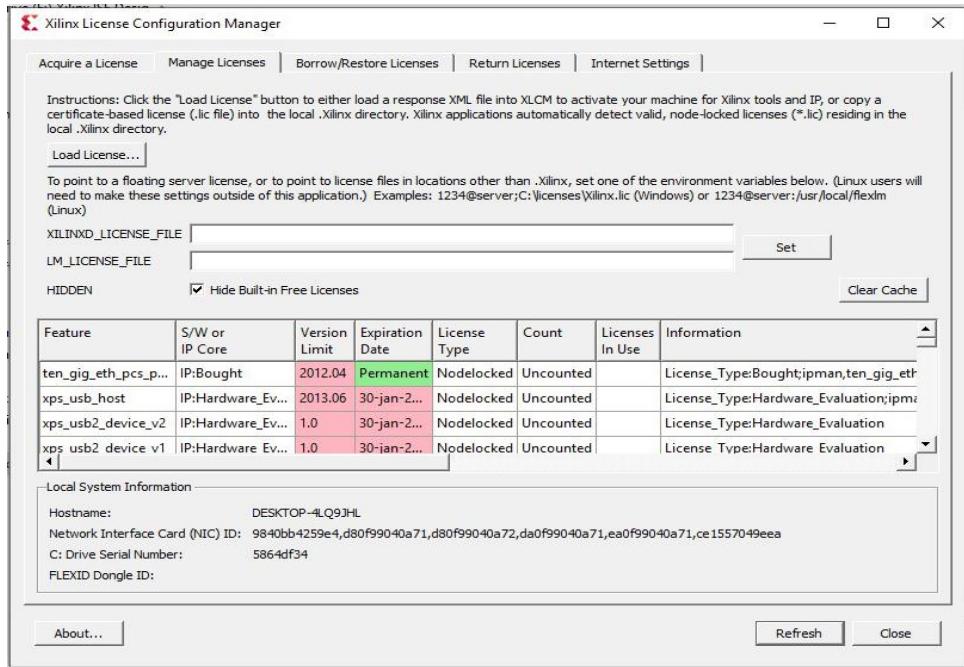


## 12. เมื่อติดตั้งเสร็จจะแสดงหน้านี้ขึ้นมาจากนั้นให้กดเลือก Locate Existing License(s) จากนั้นกด Next

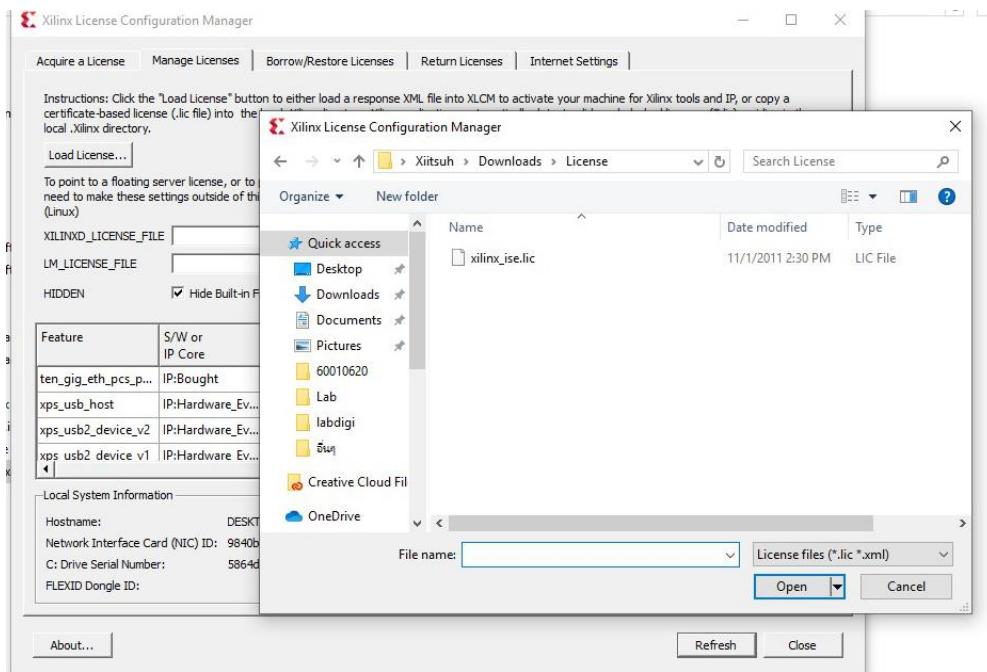


13.ให้ Download License จาก <https://drive.google.com/file/d/0BwHgcrkzntV2UzTDdWa1p1TjQ/view>

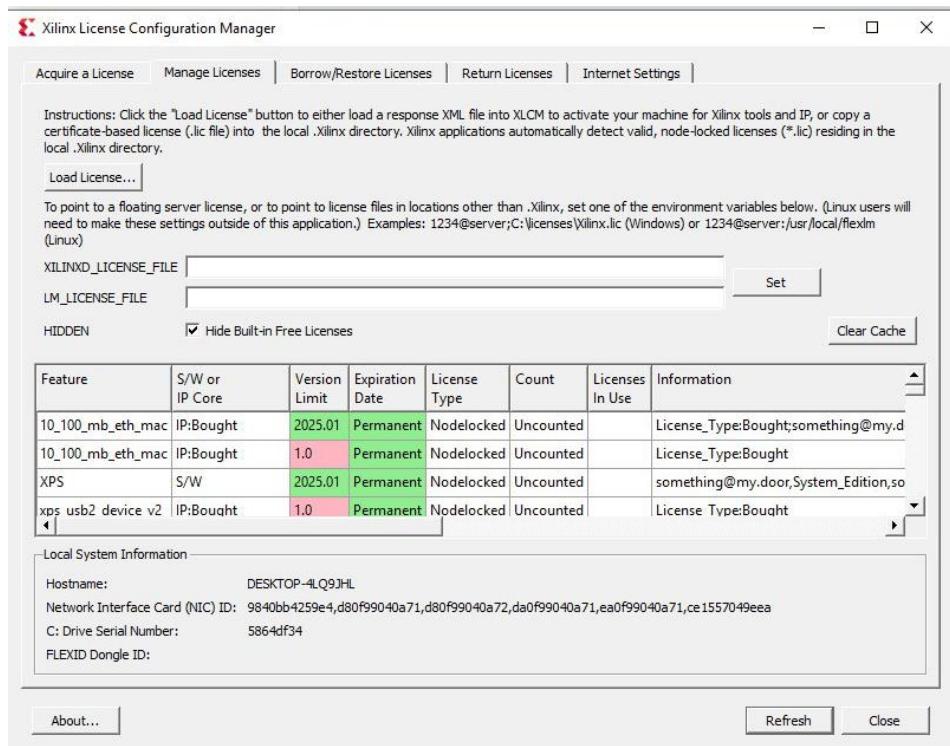
จากนั้นคลิกที่ Load License



14.จากนั้นเลือกไฟล์ License ที่ Download มา



## 15. เมื่อเสร็จสิ้นจะขึ้นดังรูป



## ปัญหาที่อาจ(ต้อง)พบ

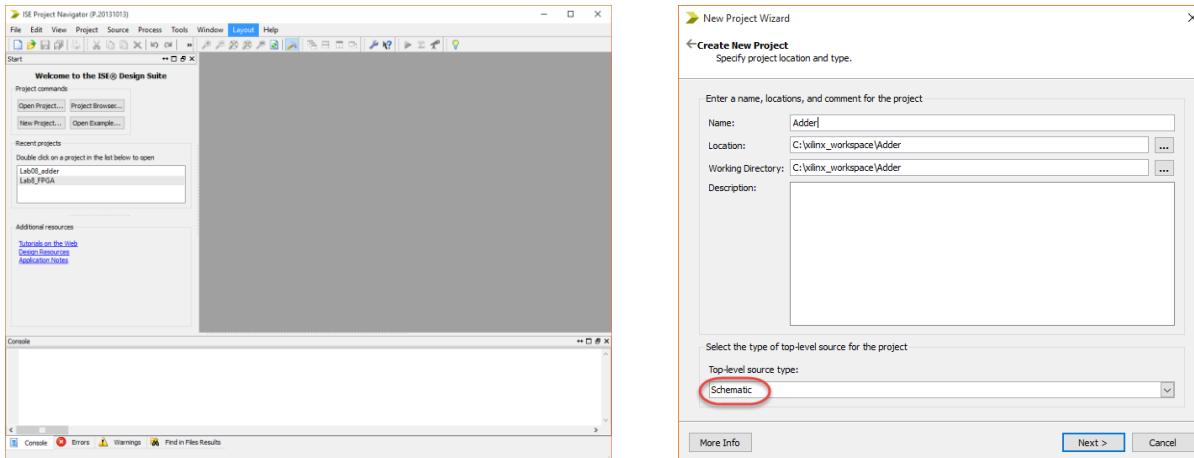
- เมื่อติดตั้งเซิร์ฟ ไม่แสดงหน้าต่างสำหรับลง license หรือ ไม่สามารถเปิด program impact หรือ create svf ไฟล์ได้เนื่องจากมีปัญหาที่โปรแกรม Impact , PlanAhead สำหรับระบบปฏิบัติการ Windows 10 โดยสามารถแก้ไขได้ตามขั้นตอนดังนี้
  - เปิด ไดเรกทอรี่ C:\Xilinx\14.7\ISE\_DS\ISE\lib\nt64
  - หาไฟล์ที่มีชื่อ libPortability.dll และเปลี่ยนชื่อให้เป็น libPortability.dll.orig
  - ทำการ copy ไฟล์ libPortabilityNOSH.dll และวัดความที่ไดเรกทอรีนี้ เสร็จแล้วให้ทำการเปลี่ยนชื่อเป็น libPortability.dll
  - Copy libPortabilityNOSH.dll อีกครั้ง, เปลี่ยนไดเรกทอรีเป็น C:\Xilinx\14.7\ISE\_DS\common\lib\nt64 และวางไฟล์ที่ไดเรกทอรีนี้
  - ในไดเรกทอรีนี้ C:\Xilinx\14.7\ISE\_DS\common\lib\nt64 หาไฟล์ที่มีชื่อ libPortability.dll และเปลี่ยนชื่อให้เป็น libPortability.dll.orig
  - เปลี่ยนชื่อ libPortabilityNOSH.dll เป็น libPortability.dll
  - เปิดไดเรกทอรี่ C:\Xilinx\14.7\ISE\_DS\PlanAhead\bin และเปลี่ยนชื่อ rdiArgs.bat เป็น rdiArgs.bat.orig

8. Download ไฟล์ rdiArgs.bat จากลิงค์ [https://drive.google.com/file/d/1skq6DJ0tEJNyU0fooqGex\\_iP7cDlf0/view?usp=sharing](https://drive.google.com/file/d/1skq6DJ0tEJNyU0fooqGex_iP7cDlf0/view?usp=sharing)
9. ทำการแตกไฟล์ จากนั้น copy ไฟล์ rdiArgs.bat แล้วทำการวางในไดเรกทอรี่ C:\Xilinx\14.7\ISE\_DS\PlanAhead\bin
10. ศึกษาเพิ่มเติมได้ที่ <https://www.eevblog.com/forum/fpga/guide-getting-xilinx-ise-to-work-with-windows-8-64-bit/msg479087/?PHPSESSID=f071046e7039d4d001072688ff9deb3f#msg479087>

## การใช้งานโปรแกรม

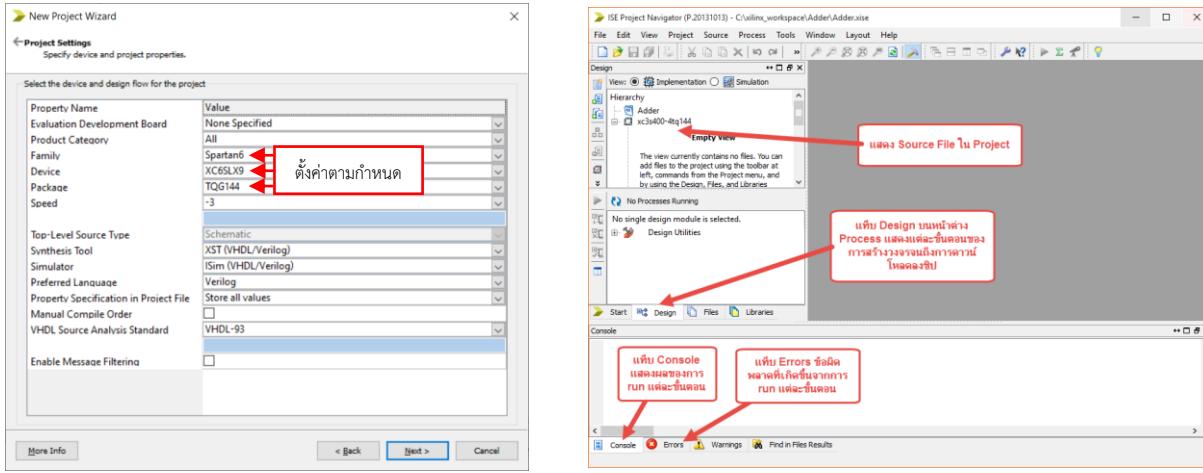
การทดลองนี้ใช้โปรแกรม Xilinx ISE WebPack 14.7 ให้ทดลองตามขั้นตอนดังนี้

1. เปิดโปรแกรม Xilinx ISE Design Suite 14.7 (Run as administrator) และสร้าง 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

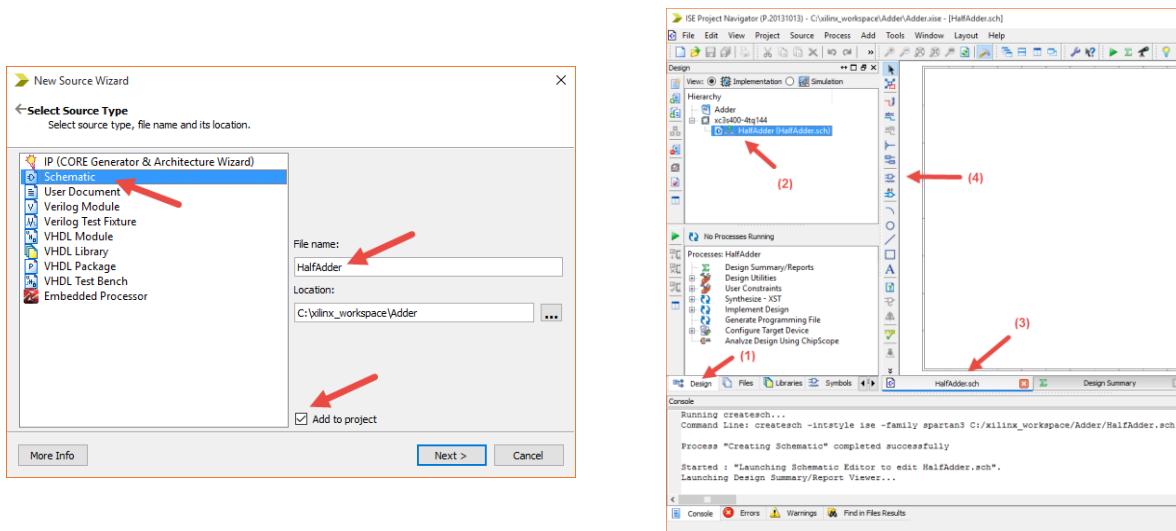


รูปที่ 7 หน้าต่างโปรแกรม Xilinx 8.2 (ซ้าย) และการกำหนดไฟล์ Project (ขวา)

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

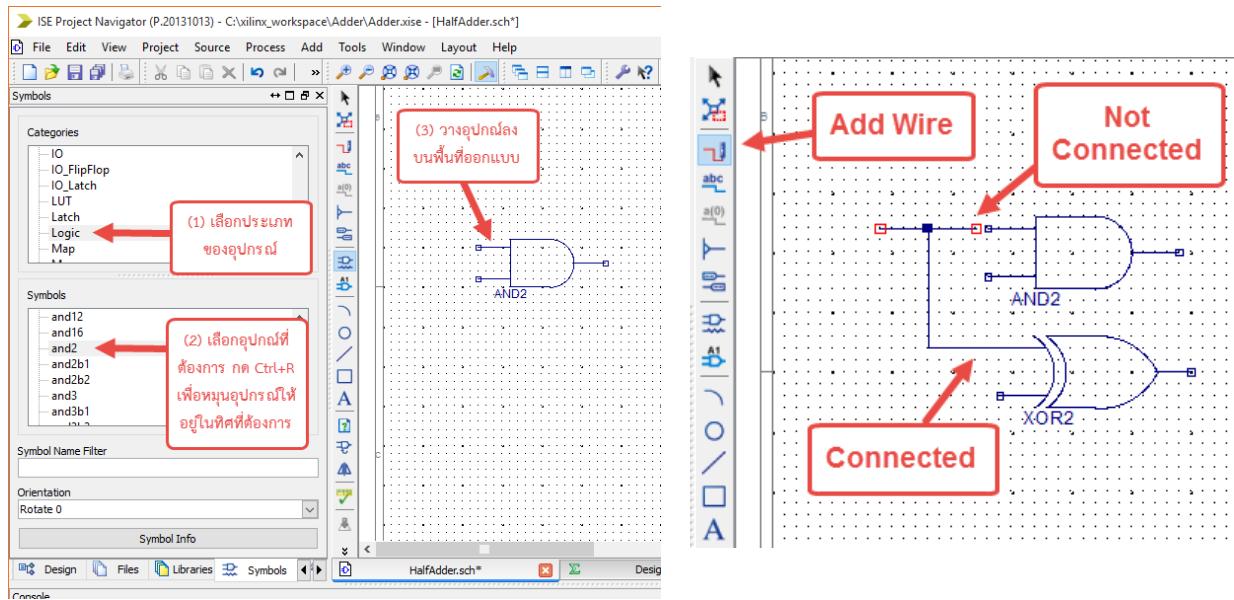


รูปที่ 8 การกำหนดลักษณะของ FPGA ที่ใช้งาน (ซ้าย) ส่วนประกอบหลักของโปรแกรม (ขวา)



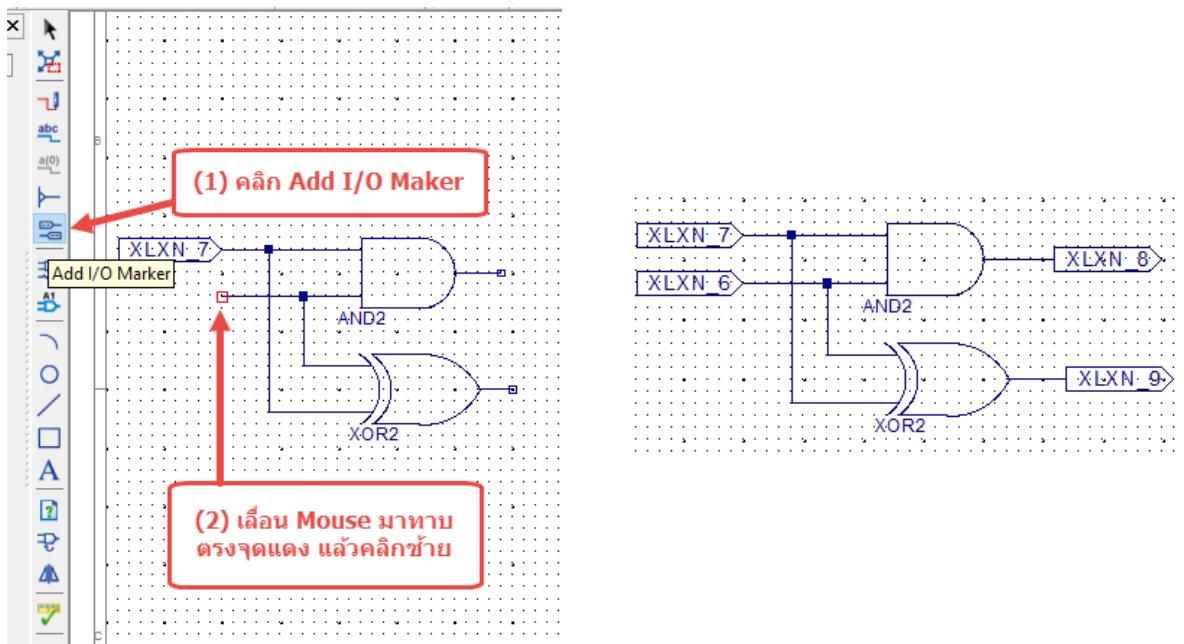
รูปที่ 9 สร้าง Schematic (ซ้าย) และส่วนประกอบภายในโปรแกรม (ขวา)

5. 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) เลือกแท็บพินที่ใช้ในการวัดวงจร และ (4) ใช้แบบด้านข้างสำหรับเป็นเครื่องมือวัดวงจร
6. เริ่มวัดวงจรแรกเป็น 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 การวางแผน (ซ้าย) และการเชื่อมต่อสาย (ขวา)

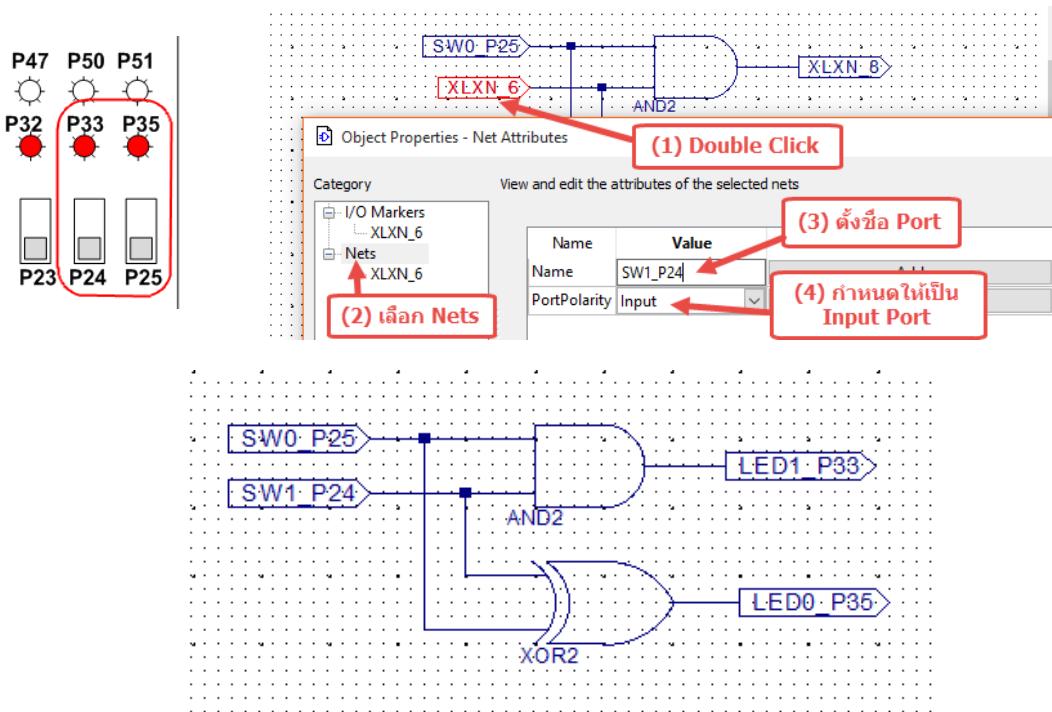
8. สร้างจุดสำหรับเชื่อมต่อ กับขาของ ชิป FPGA โดยเลือก Add I/O Marker ดังรูปที่ 11 (ซ้าย) วางแผนเชื่อมต่อจนได้ผลลัพธ์ เป็นวงจร 1-Bit Half Adder ดังรูปที่ 11 (ขวา)



รูปที่ 11 สร้างจุดเชื่อมต่อ (ซ้าย) และวงจรเมื่อวัดเสร็จแล้ว (ขวา)

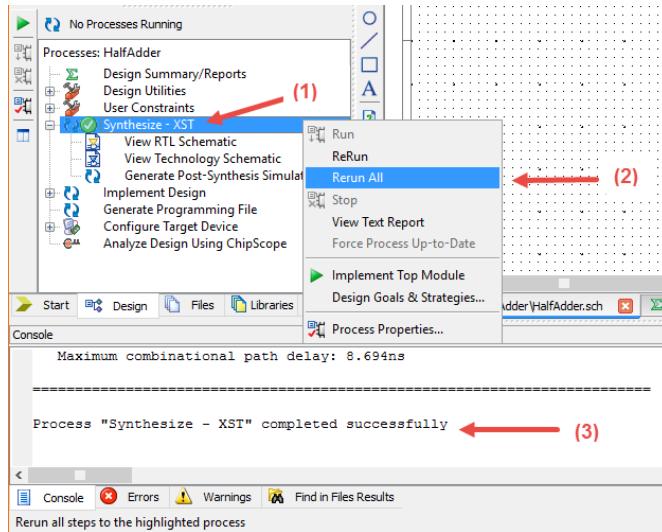
9. เปลี่ยนชื่อตัวรับสัญญาณเข้าและออกใน I/O Marker โดยพิจารณา Ruth ที่ 1 หรือ 12 (บนซ้าย) สำหรับการทดลองนี้ให้เลือกสวิตช์ที่เชื่อมต่อกับขา 24 และ 25 ของ FPGA เป็นตัวป้อนสัญญาโนินพุต และใช้ LED ที่เชื่อมต่อกับขา 33 และ 35 ของ FPGA เป็นตัวแสดงสัญญาโนเอาต์พุต โดยกำหนดให้ SW0 และ SW1 เป็นชื่อสัญญาโนินพุต และ LED0 และ LED1 เป็นชื่อสัญญาโนเอาต์พุตเพื่อแสดงผลรวมและตัวทดตามลำดับ ทั้งนี้ควรตั้งชื่อในรูปแบบ “ชื่อสัญญาณ\_หมายเลข”

ขา” เช่น SW0\_P25 หมายถึง ใช้สัญญาณ SW0 ต่อกับขา P25 เมื่่าว่าจะไม่ได้เป็นข้อบังคับ แต่การตั้งชื่อรูปแบบนี้ช่วยให้แก้ไขปัญหาได้ง่าย รูปที่ 12 (บนขวา) แสดงหน้าต่างสำหรับการตั้งค่า และรูปที่ 12 (ล่าง) แสดงการตั้งค่า I/O Marker ที่เสร็จสมบูรณ์



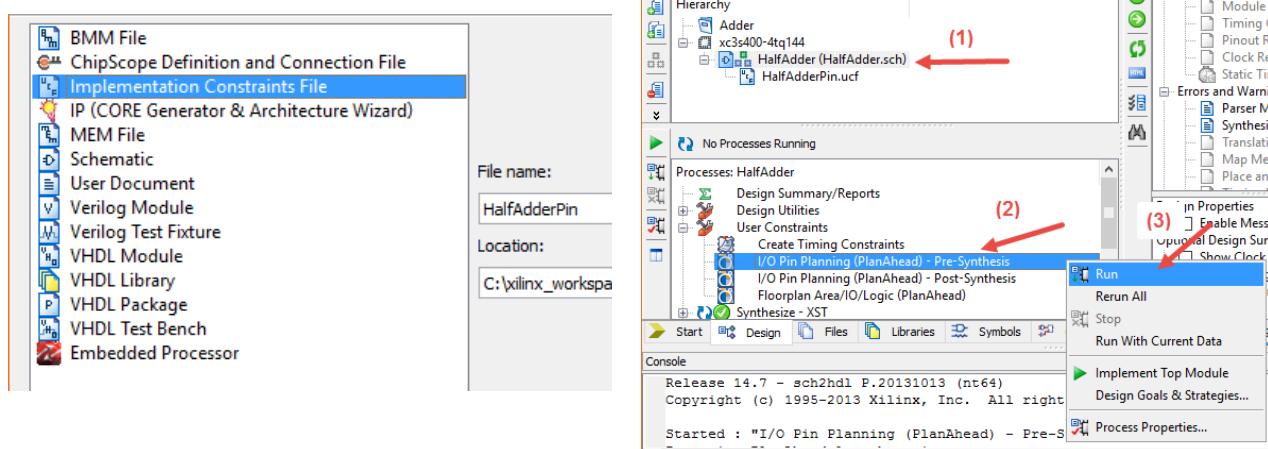
รูปที่ 12 การกำหนดชื่อตัวรับสัญญาณเข้าและออกใน I/O Marker

10. ตรวจสอบการทำงานของวงจรที่ออกแบบโดยเลือกแท็บ Design ที่หน้าต่าง Process เลือก Synthesize - XST แล้ว ดับเบิลคลิกหรือกดคุณลักษณะแล้วเลือก Rerun All เพื่อเริ่มตรวจสอบวงจร หากการเดินสายและการวางอุปกรณ์ถูกต้อง จะแสดง Process “Synthesize” completed successfully ที่หน้าต่าง Console ดังรูปที่ 13 กรณีเกิดข้อผิดพลาดต้องกลับไปตรวจสอบborgตามขั้นตอนก่อนหน้า



รูปที่ 13 ขั้นตอนตรวจสอบ Schematic Design

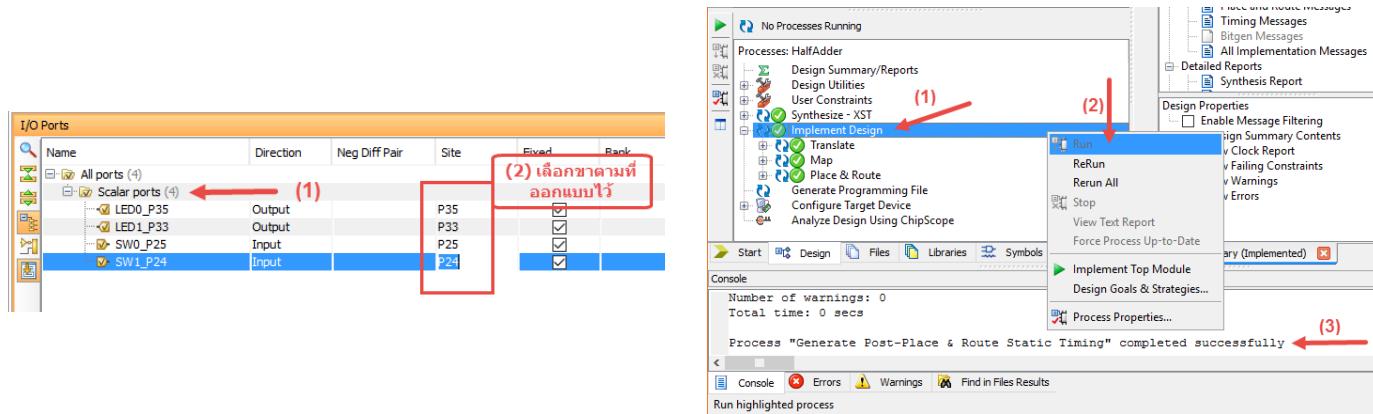
11. จับคู่อุปกรณ์กับหมายเลขพอร์ต โดยเลือก Project -> New Source -> Implementation Constraints File เพื่อจับคู่ระหว่างอุปกรณ์บนบอร์ดกับหมายเลขขาไอซี (Pin Number) ดังรูปที่ 14



รูปที่ 14 การจับคู่อุปกรณ์กับหมายเลขพอร์ต

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

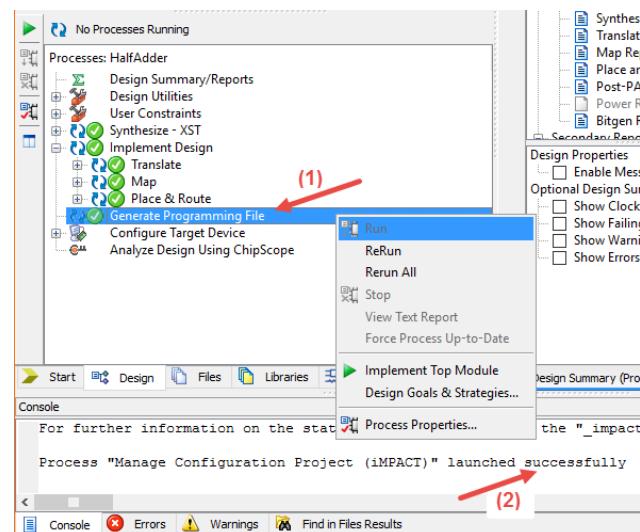
13. กำหนดหมายเลขขาลงในหน้าต่าง I/O Port ซึ่งอยู่ด้านล่างของโปรแกรม PlanAhead และใส่รายละเอียดให้ครบเหมือนดังรูปที่ 15 (ซ้าย) หากไม่สามารถระบุเลขลงไปได้หรือโปรแกรมปฏิเสธแสดงว่ากำหนดหมายเลขขาผิด



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 16 (ซ้าย) เลือก Generate Programming File เลือก Run, Rerun All



รูปที่ 16 การ Generate Programmable File

คู่มือการใช้งานโปรแกรม

FPGA Flasher

ในการโปรแกรมลงบอร์ด FPGA

ผ่านโมดูล FT2233HL

หมายเหตุ ถ้าหากเลือกโปรแกรมแบบ Flash เมื่อถอดปลั๊ก โปรแกรมจะหายไปจาก FPGA หากต้องการให้ FPGA ยังมีโปรแกรมที่เรา run อยู่แม้จะถอดปลั๊กให้ลงโดยเลือกแบบ PROM โดยการโปรแกรมแบบ PROM จำนวนครั้งที่โปรแกรมได้จะมีอยู่อย่างจำกัด ให้ใช้อย่างคุ้มค่า

## วิธีการอัพโหลดโปรแกรมลงบอร์ด FPGA ด้วยโปรแกรม FPGA Flasher

1) ดูคลิปด้านล่างนี้ และทำตามขั้นตอนดังกล่าว



<https://youtu.be/PWsH0virRqc>

2) ถ้าจะอัพโหลดโปรแกรมใหม่ ให้ปิดทุกอย่างแล้วเริ่มทำการขั้นตอนตั้งแต่แรกอีกครั้ง

3) ไฟล์ที่ซับใช้แคปไฟล์ .bit

5) ลิงค์สำหรับดาวน์โหลด FPGA Flasher



<https://github.com/ouoam/FPGA-Flasher>

การใช้งานบอร์ด FPGA จะเสร็จสิ้นที่ส่วนนี้ ต่อไปจะเป็นการโปรแกรมในรูปแบบอื่น ๆ ซึ่งนักศึกษาสามารถข้ามส่วนหลังจากนี้ไปได้

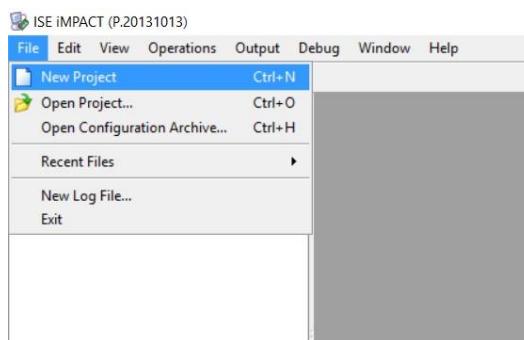
# คู่มือการใช้งานโมดูล FT2232H ในการโปรแกรมบอร์ด Xilinx SURVEYOR-6

หมายเหตุ ถ้าหากเสร็จในขั้นการเตรียมไฟล์ SVF สำหรับโปรแกรมลงแฟร์ชของ FPGA จะสามารถ run ใน XC3SPROG ได้ แต่เมื่อถอดปลั๊ก โปรแกรมจะหายไปจาก FPGA หากต้องการให้ FPGA ยังมี โปรแกรมที่เรา run อยู่แม้จะถอดปลั๊กให้ลงโดยใช้ PROM

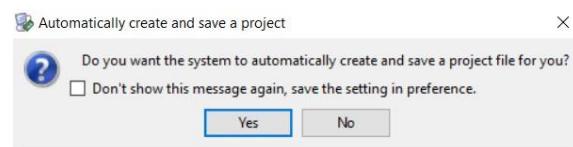
## การใช้งาน FT2232

### ขั้นตอนการเตรียมไฟล์ SVF สำหรับโปรแกรมลงเฟรชของ FPGA

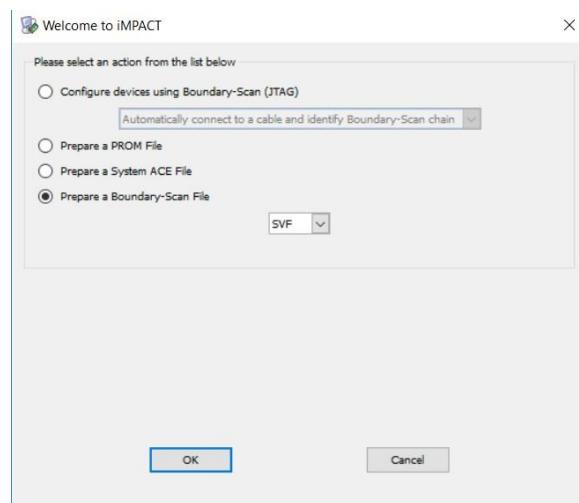
1. เปิดโปรแกรม iMPACT และกดสร้าง New Project



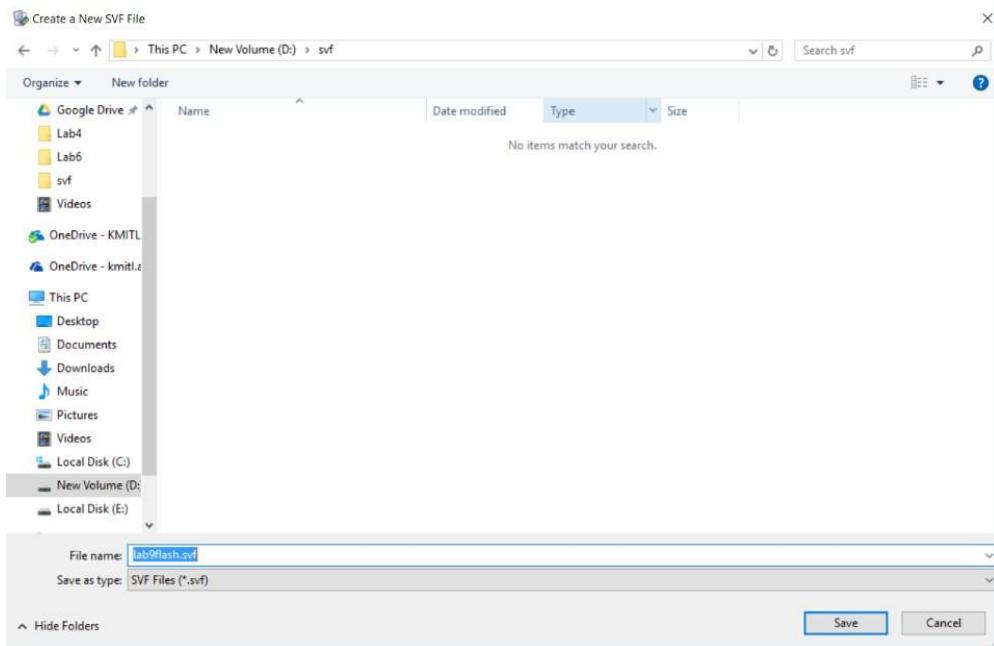
2. กด Yes



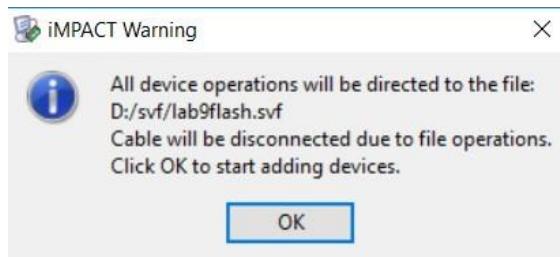
3. เลือก option Prepare a Boundary-Scan File เลือก SVF และกด OK



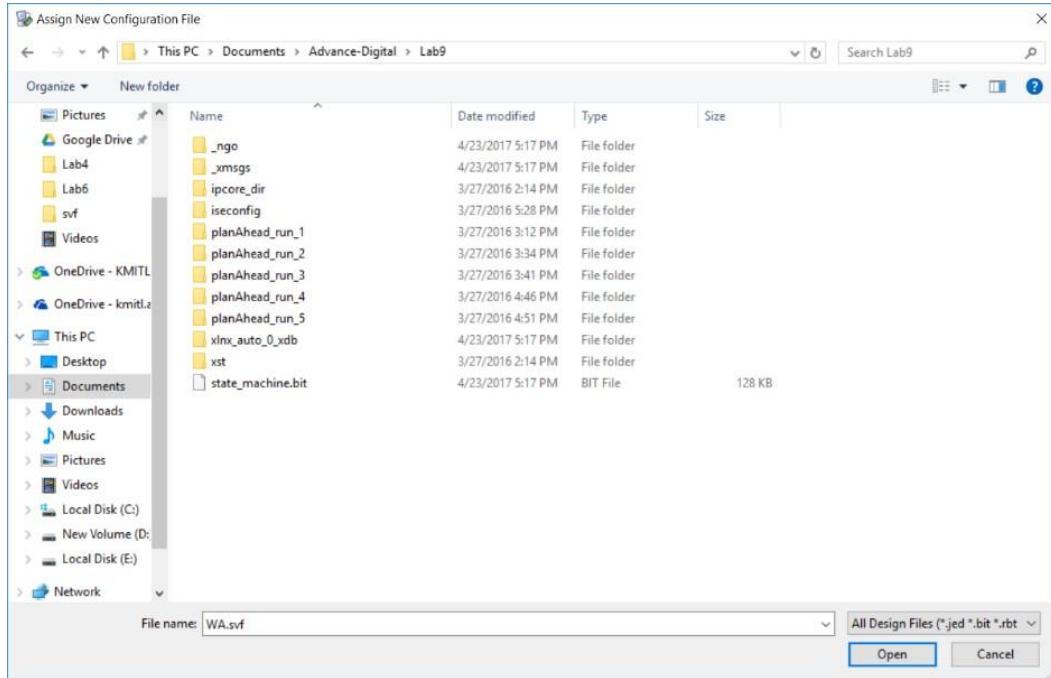
4. เลือกพื้นที่สำหรับ save ไฟล์ SVF ซึ่งเป็นไฟล์ที่ใช้สำหรับโปรแกรมผ่านโมดูล FT2232H โดยในตัวอย่างได้ save ไว้ในโฟลเดอร์ที่สร้างขึ้นมา ชื่อ svf ในไดร์ฟ D



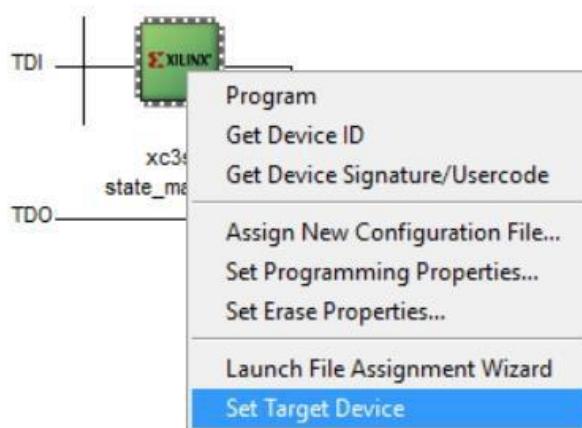
5. กด OK



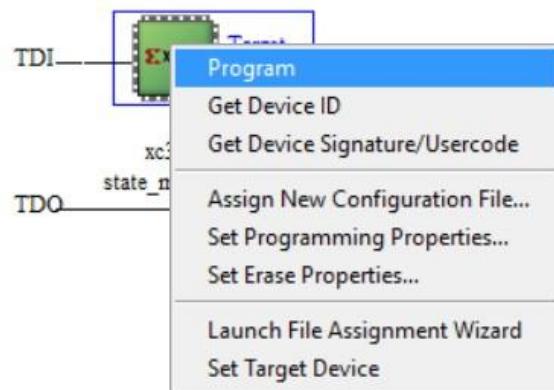
6. ให้เลือกไฟล์นามสกุล .bit จากโฟลเดอร์ของ Project ที่เราต้องการโปรแกรมข้อมูลลง FPGA แล้วกด Open



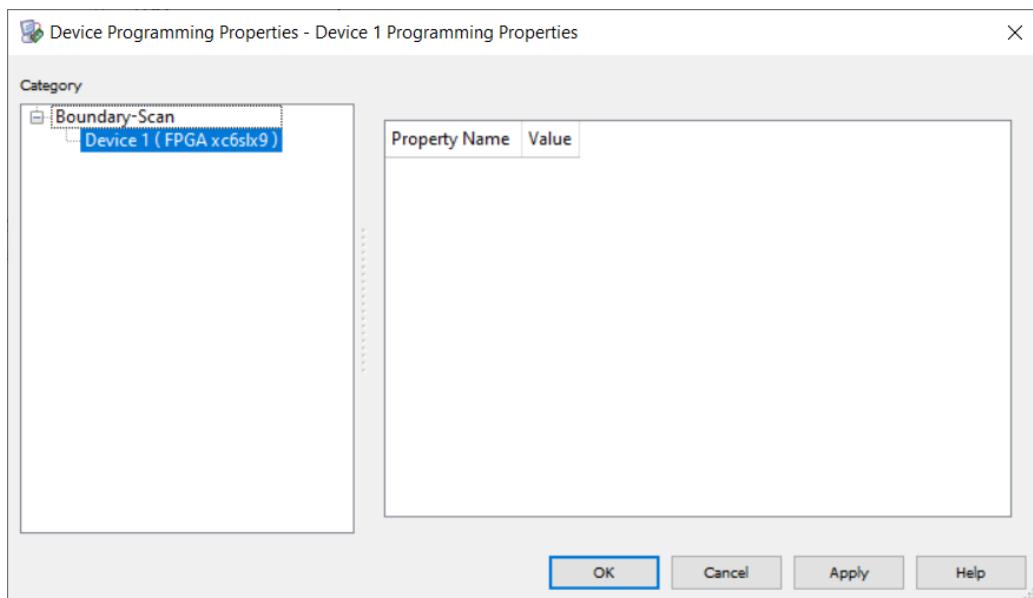
7. จะมีสัญลักษณ์ชิปประกายขึ้น คลิกขวาแล้วเลือก Set Target Device



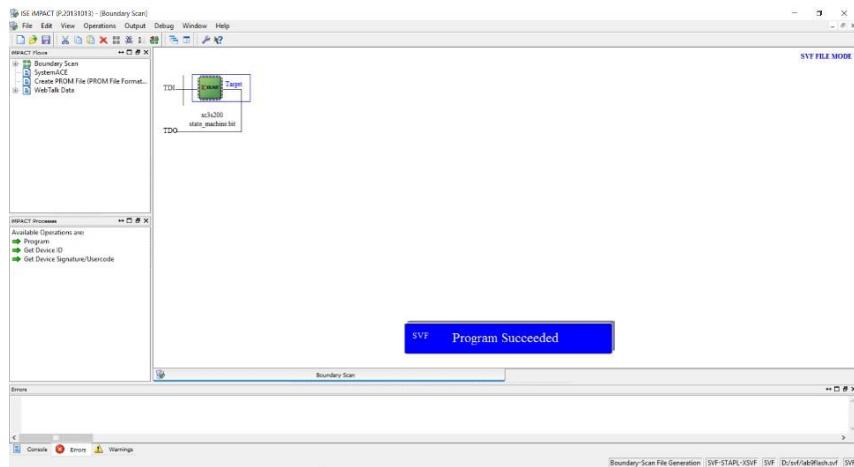
8. คลิกขวาที่ชิปอิกครั้ง แล้วเลือกกด Program



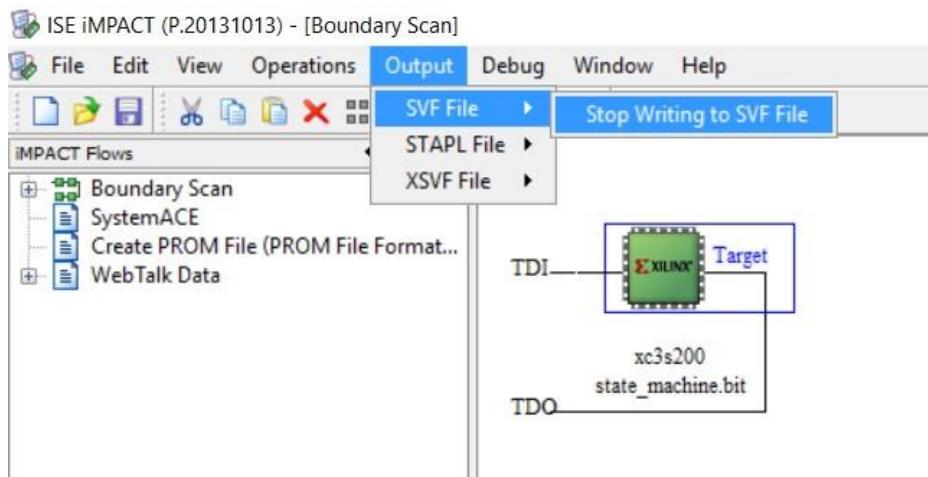
9. กด OK



## 10. เมื่อโปรแกรมทำงานสำเร็จ จะแสดงข้อความบอกดังรูป

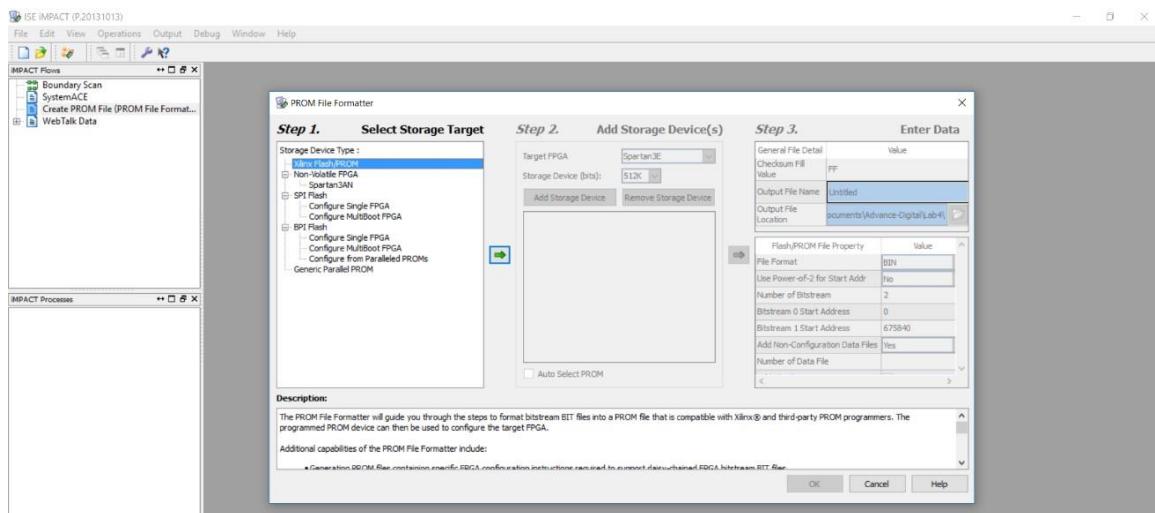


## 11. ไปที่เมนู Output -> SVF File -> เลือก Stop Writing to SVF File จะเป็นการเสร็จสิ้นการเตรียมไฟล์ SVF สำหรับ โปรแกรมลงแฟร์ช

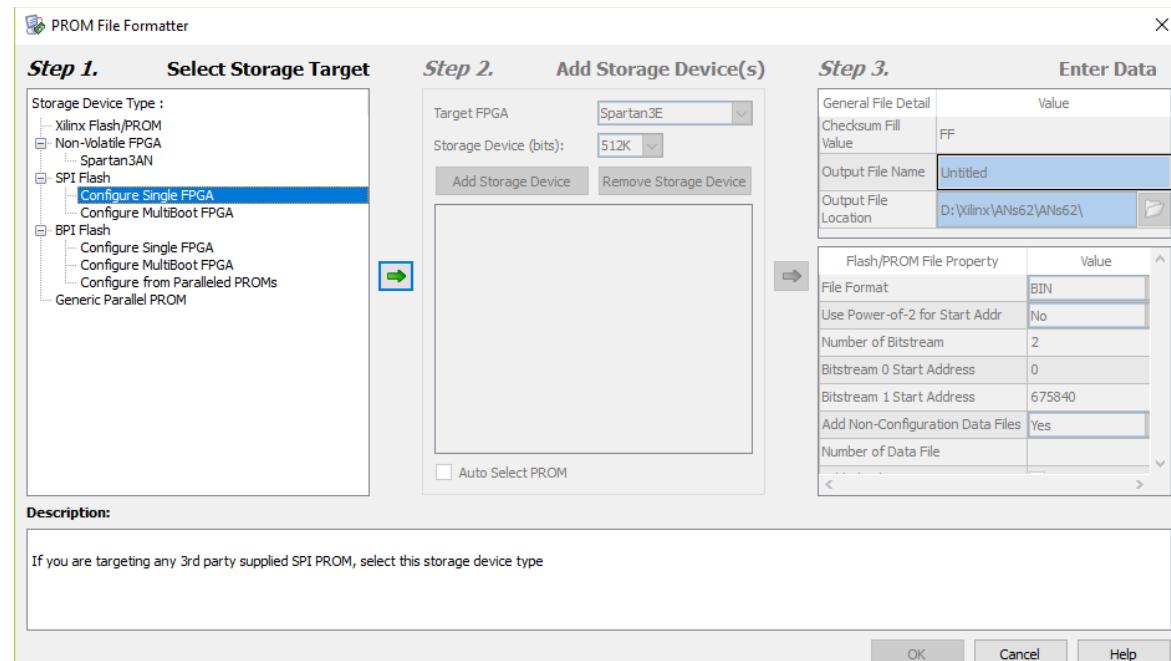


## ขั้นตอนการเตรียมไฟล์ SVF สำหรับโปรแกรม PROM ผ่าน SPI

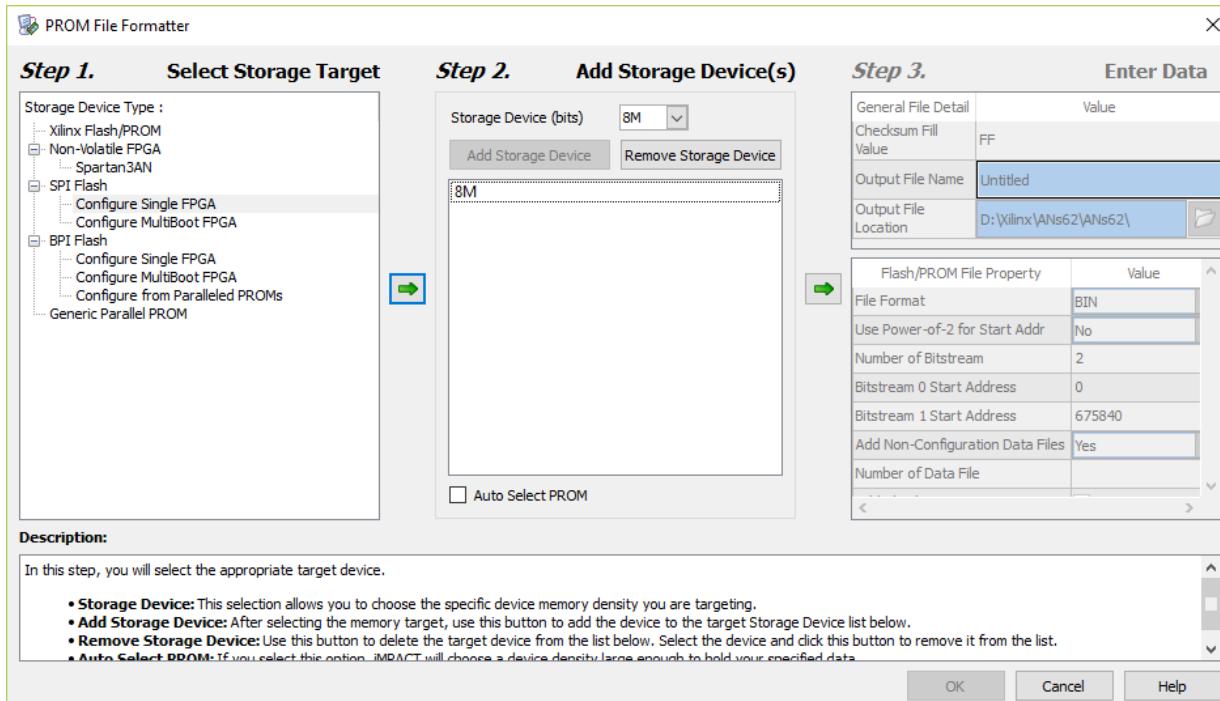
- เปิดโปรแกรม iMPACT แล้วดับเบิลคลิกที่ Create PROM File(PROM File Formatter) ที่เมนูด้านซ้าย จะปรากฏหน้าต่างตามรูปด้านล่าง



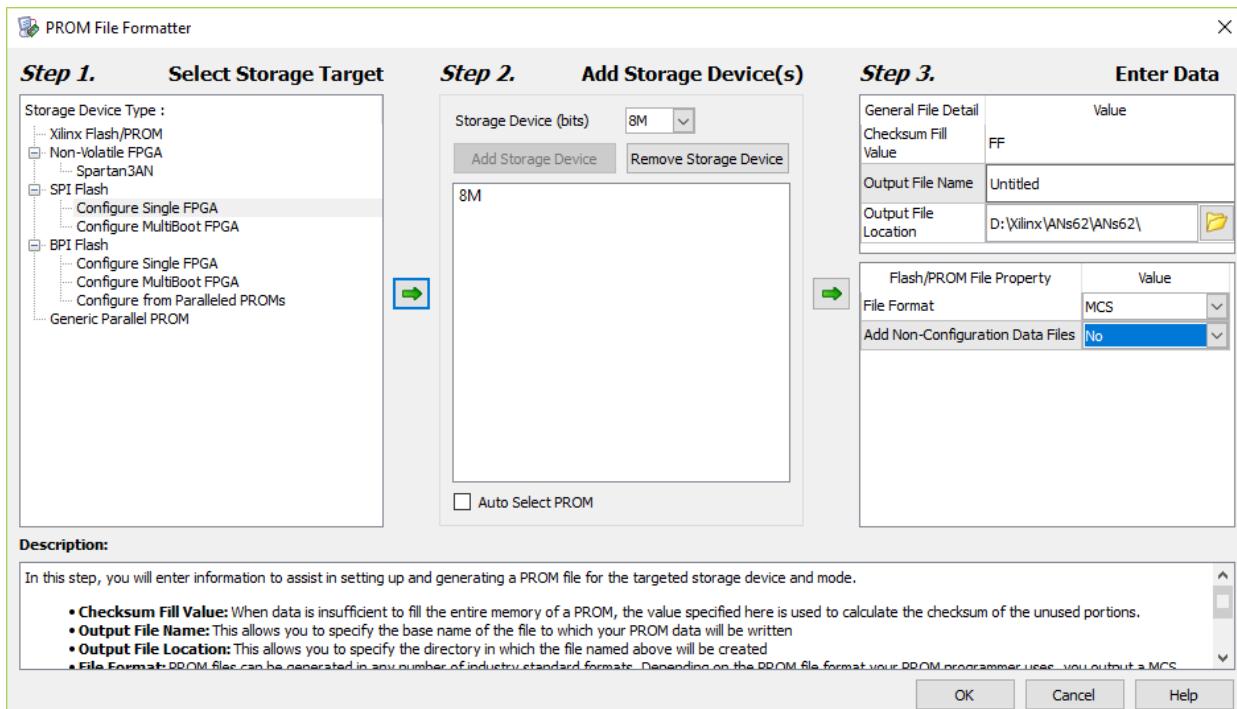
- Step 1. ในส่วนของ SPI Flash เลือก Configure Single FPGA และกดลูกศรสีเขียว



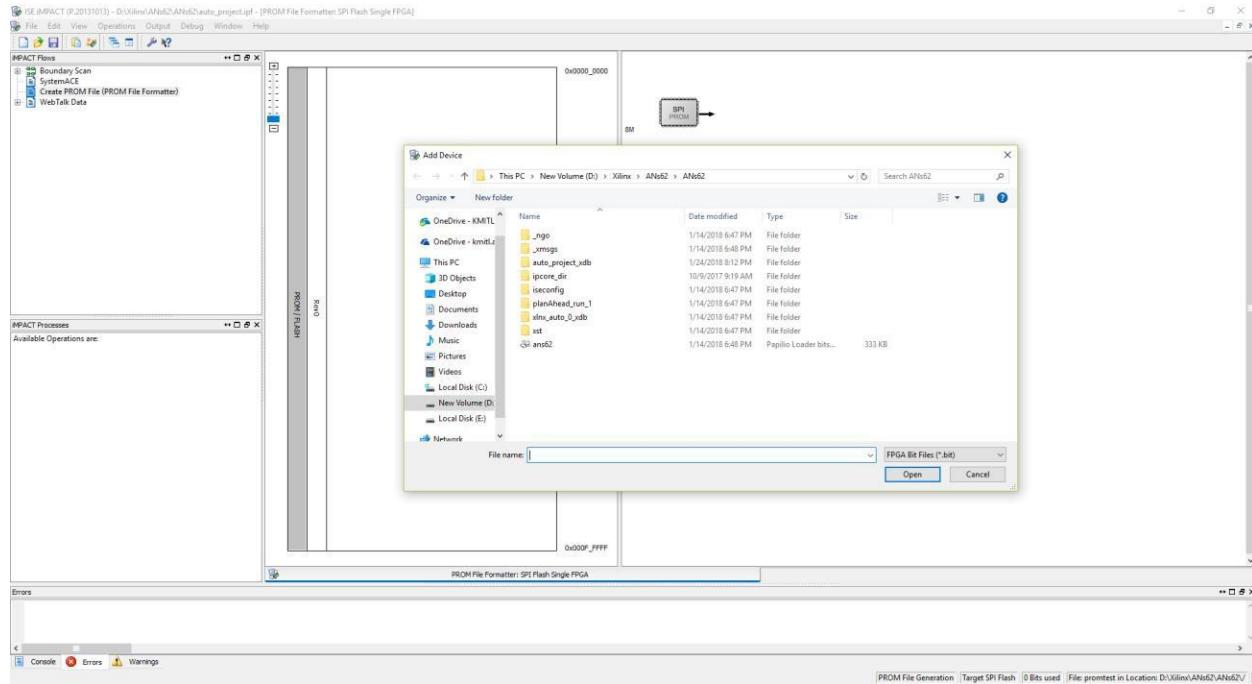
3. Step 2. เลือกขนาดของ PROM ที่ใช้ โดยในบอร์ด SURVEYOR-6 ดังเป็น 8M ตามขนาดของ M25P80 กด Add Storage Device และกดลูกศร



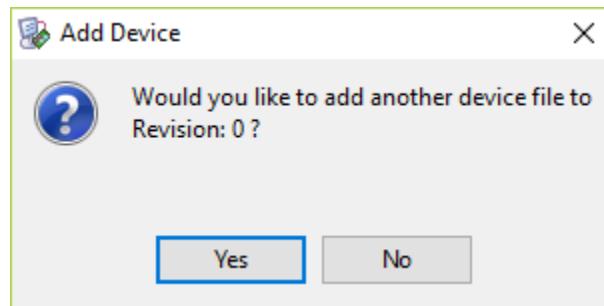
4. Step 3. สามารถตั้งชื่อไฟล์ และที่อยู่ได้ โดยไฟล์ฟอร์แมตต้องเป็น MCS และกด OK



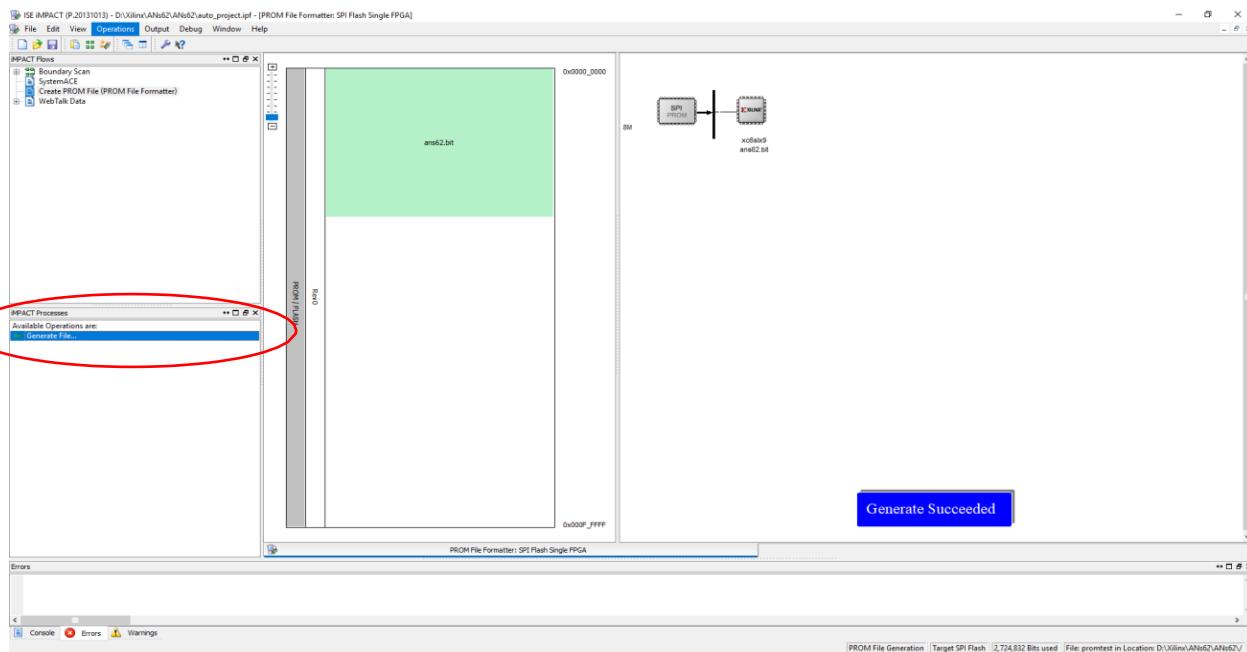
5. ให้เลือกไฟล์นามสกุล .bit ที่ต้องการ กด OK



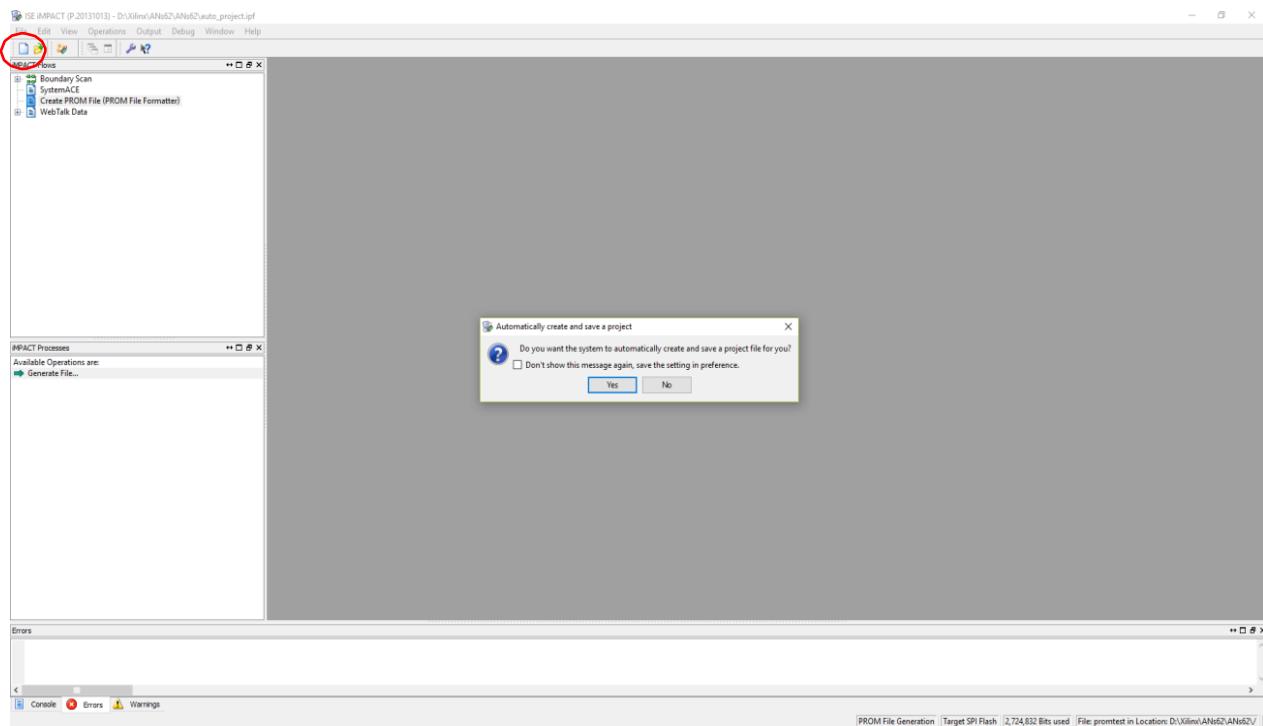
6. เลือก No



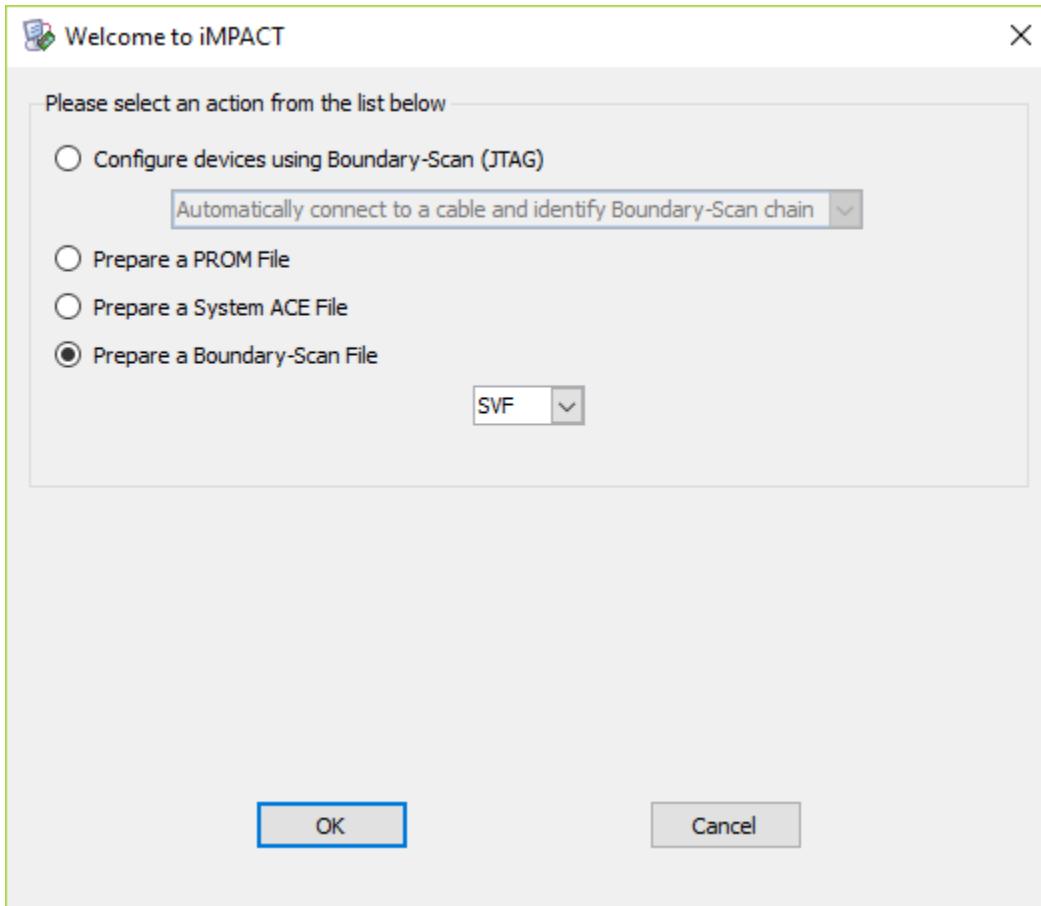
7. กด Generate File... เพื่อสร้างไฟล์ โดยจะขึ้นแสดงคำว่า Generate Succeeded เมื่อสำเร็จ



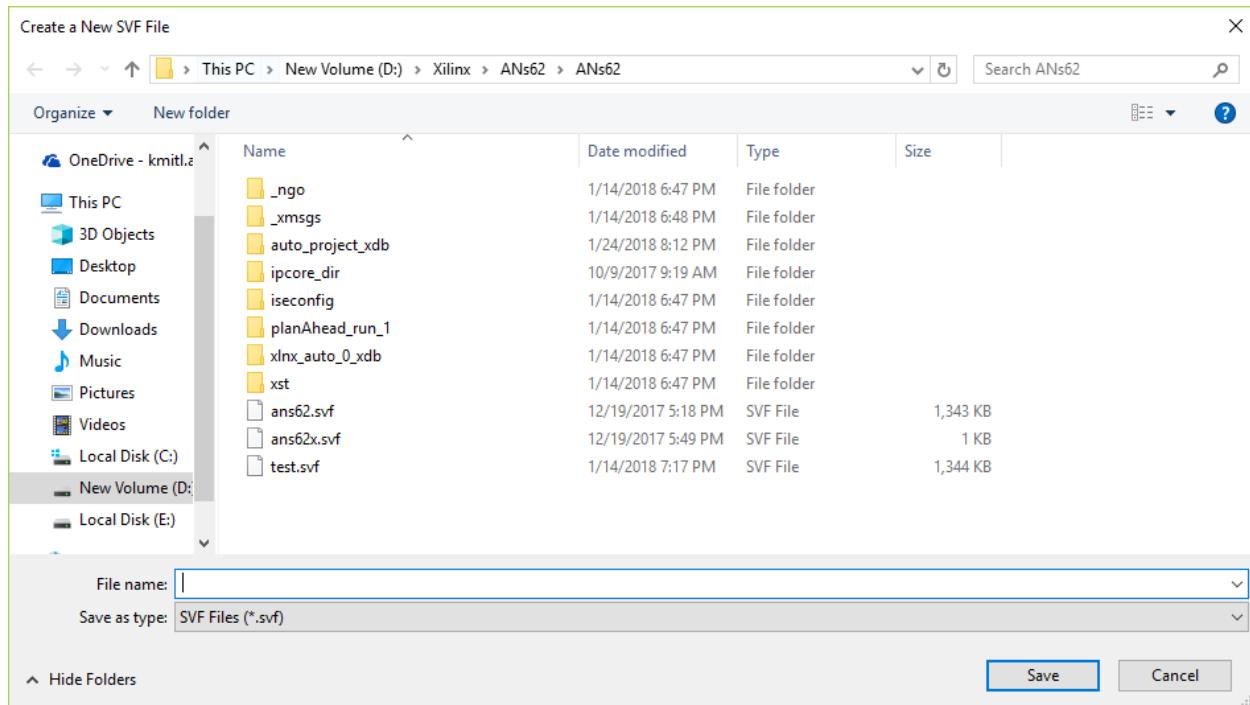
8. เลือกสร้างไฟล์ใหม่ ให้กด Yes ที่ Automatically create and save project



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

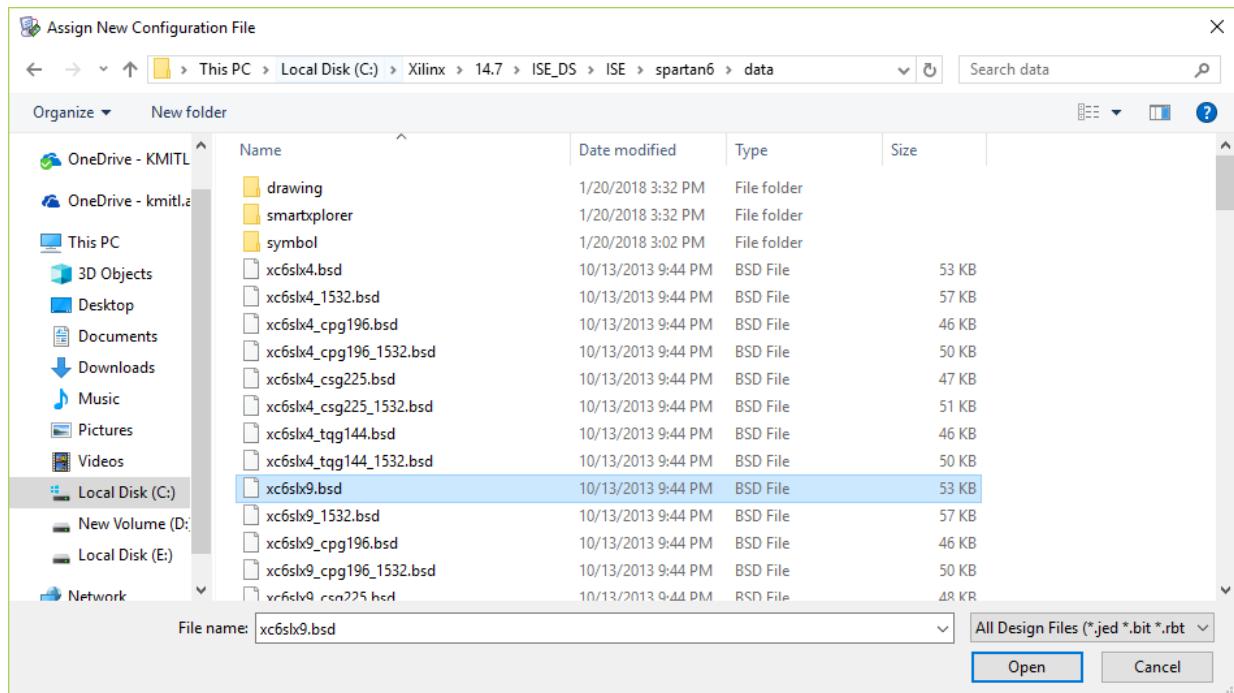


## 10. โปรแกรมจะให้เลือกพื้นที่สำหรับเซฟไฟล์ SVF

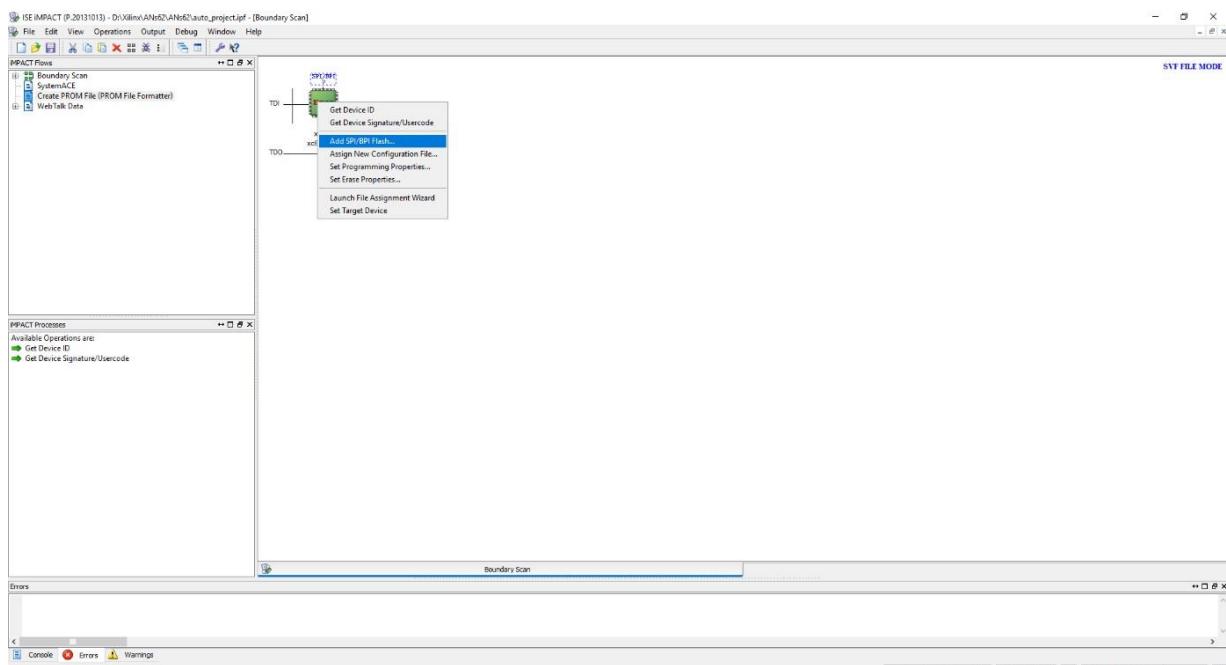


## 11. โปรแกรมจะถามหา configuration ของ FPGA ที่ใช้ โดยสามารถหาได้จากโฟลเดอร์ที่ชื่อ

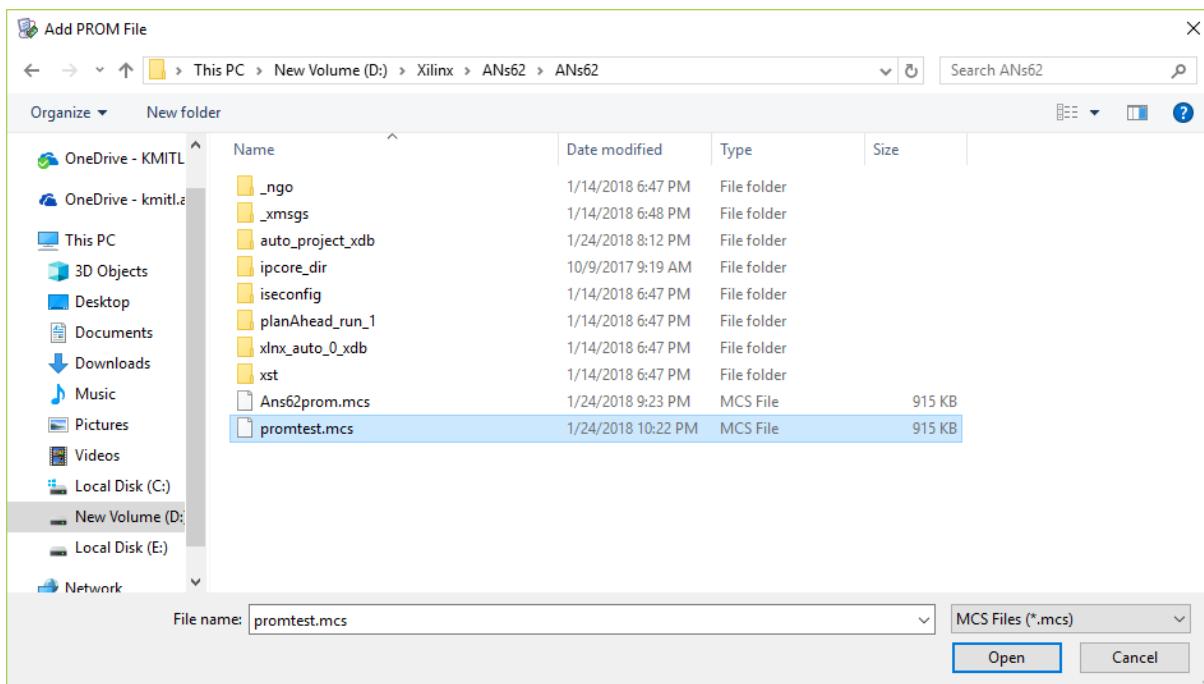
C:\Xilinx\14.7\ISE\_DS\spartan6\data เลือก xc6slx9.bsd



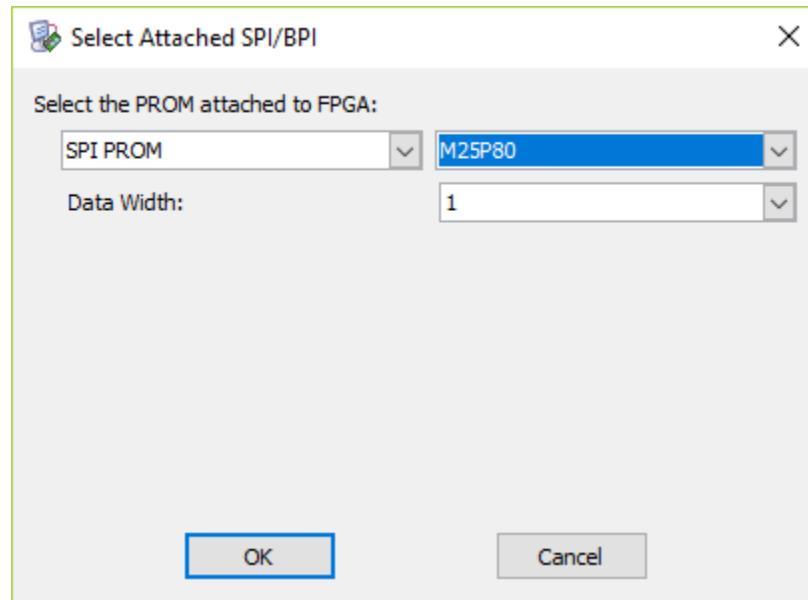
## 12. คลิกขวาที่รูปชิปแล้วเลือก ADD SPI/BPI Flash...



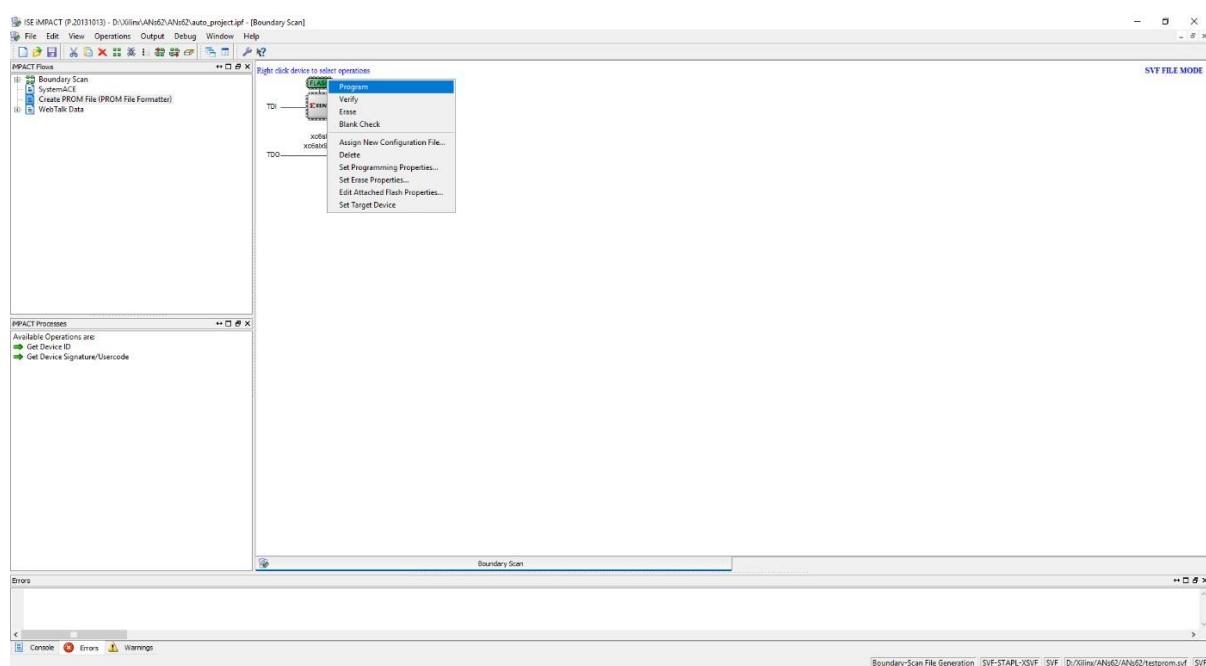
## 13. โปรแกรมจะถ่านหา PROM File ให้เลือกไฟล์ .MCS ที่ได้สร้างไว้ก่อนหน้านี้



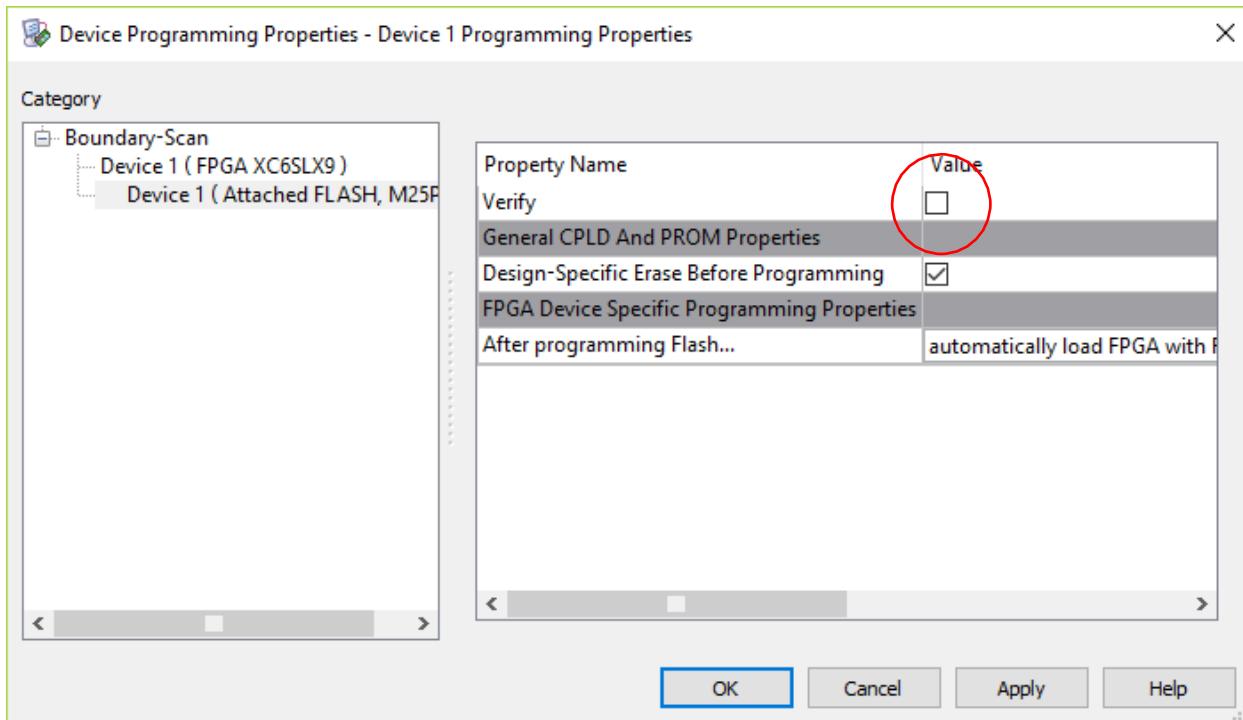
14. โปรแกรมจะถาม PROM ที่ใช้ ให้เลือก M25P80 สำหรับอร์ด SURVEYOR-6



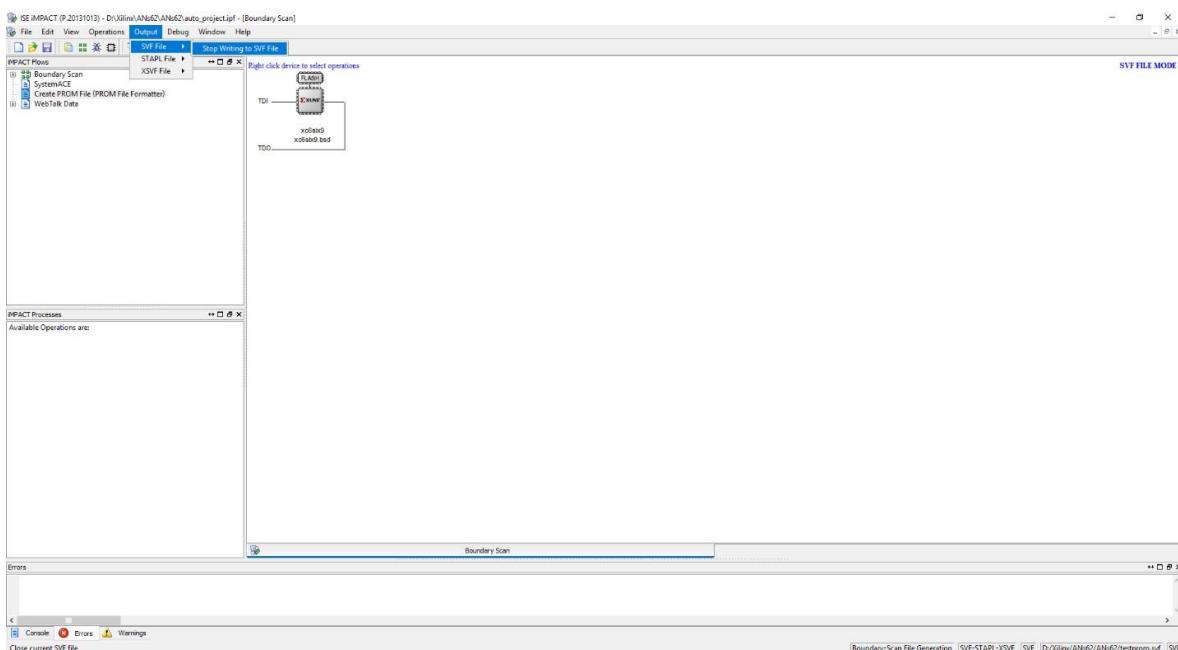
15. คลิกขวาที่สัญลักษณ์ Flash เลือก Program



16. ให้คลิกเครื่องหมายถูกของ Verify ออก แล้วกด OK โปรแกรมจะเริ่มสร้างไฟล์ SVF



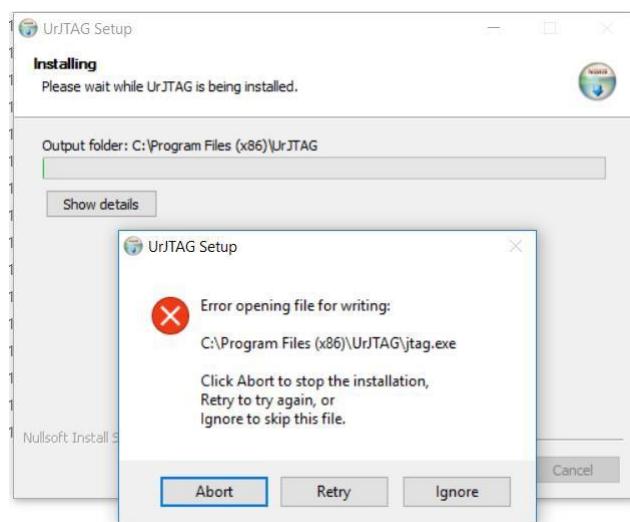
17. เมื่อสร้างไฟล์ SVF เสร็จไปที่ Output -> SVF File -> เลือก Stop Writing to SVF File เสร็จสิ้นการเตรียมไฟล์ SVF สำหรับโปรแกรมลง PROM ผ่าน SPI



# คู่มือการใช้งาน UrJtag ในการโปรแกรมลงบอร์ด FPGA

## ขั้นตอนการติดตั้ง และใช้งานโปรแกรม UrJTAG

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



- หลังจากติดตั้ง ให้เข้าไปที่โฟลเดอร์ UrJTAG ที่ติดตั้งจากข้อ 1 จนนั้นเข้าใช้โปรแกรมไฟล์ชื่อ jtag

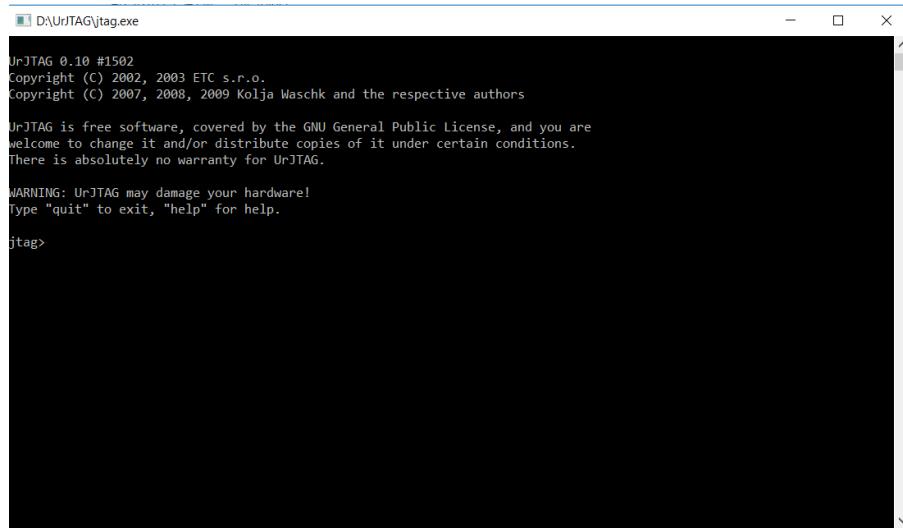
A screenshot of a Windows File Explorer window showing the contents of the UrJTAG folder on drive D. The folder structure is as follows:

- This PC > New Volume (D:) > UrJTAG

Name	Date modified	Type	Size
data	4/23/2017 1:54 PM	File folder	
doc	4/23/2017 1:54 PM	File folder	
input32.dll	4/18/2009 3:53 AM	Application extens...	96 KB
jtag	4/18/2009 3:52 AM	Application	479 KB
uninst	4/23/2017 1:54 PM	Application	51 KB

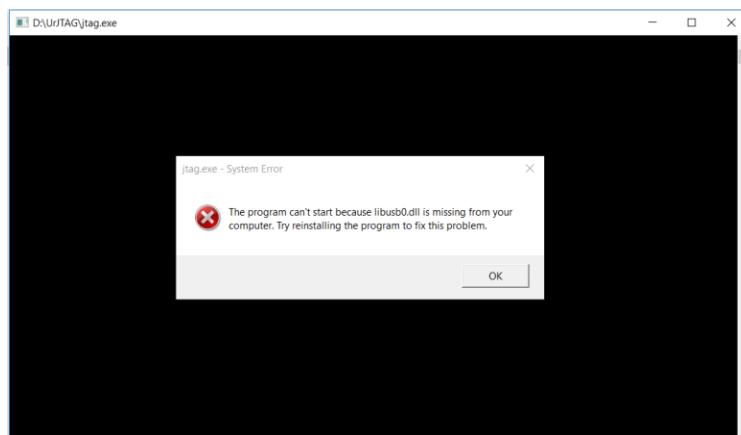
A red arrow points to the "jtag" file in the list.

จะได้หน้าต่างโปรแกรมดังนี้



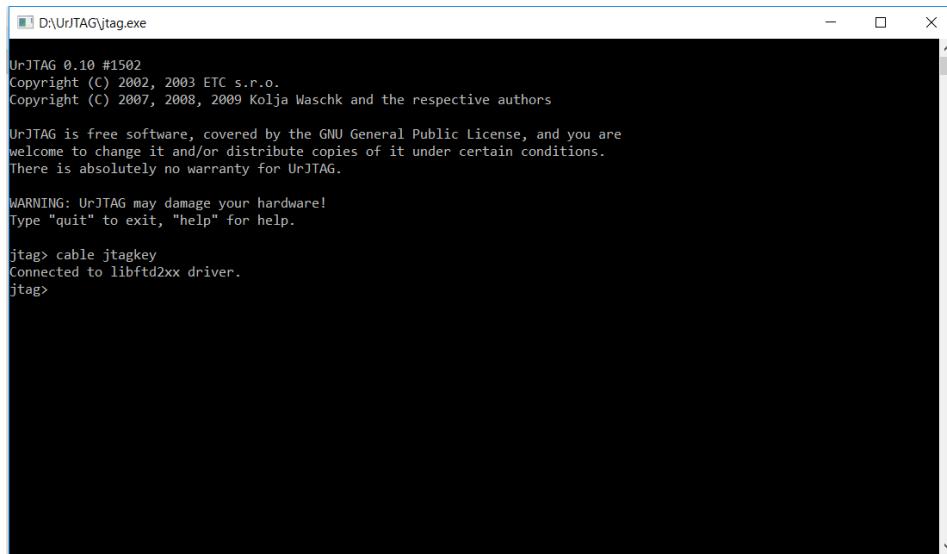
ถ้ามีการแจ้ง Error ดังภาพด้านล่าง ให้ปฏิบัติตามขั้นตอนด่อไปนี้

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



2. หากยังมีการแจ้ง Error ให้ Copy ไฟล์ FTD2XX.dll ไปวางที่โฟลเดอร์ C:\Windows\SysWOW64 สำหรับ x64 หรือ c:\Windows\system32\ สำหรับ x86 และทำการ restart โปรแกรม UrJTAG ใหม่
3. เชื่อมต่อโมดูล FT2232H กับคอมพิวเตอร์ด้วย USB

- พิมพ์คำสั่ง cable jtagkey ตามด้วยการกด enter ในโปรแกรม UrJTAG เพื่อให้เชื่อมต่อ กับโมดูล ถ้าเชื่อมต่อได้สมบูรณ์จะแสดงผลลัพธ์ตามรูปด้านล่าง



```
D:\UrJTAG\jtag.exe

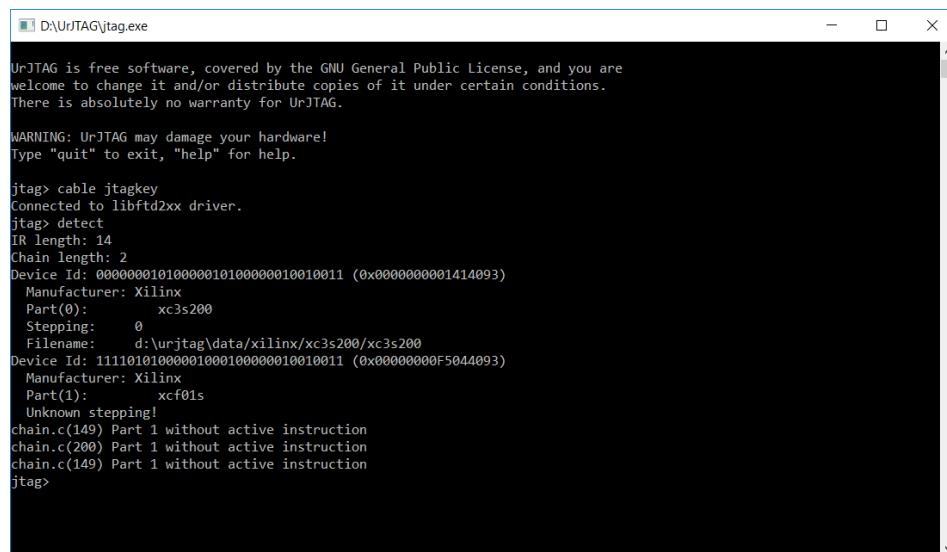
UrJTAG 0.10 #1502
Copyright (C) 2002, 2003 ETC s.r.o.
Copyright (C) 2007, 2008, 2009 Kolja Waschk and the respective authors

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>
```

- เชื่อมต่อโมดูล FT2232H กับ บอร์ด Xilinx
- พิมพ์คำสั่ง detect ตามด้วยการกด enter เพื่อตรวจสอบที่สามารถโปรแกรมได้ในกรณีเกิด 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 (0x0000000001414093)
  Manufacturer: Xilinx
  Part(0): xc3s200
  Stepping: 0
  Filename: d:\urjtag\data\xilinx\xc3s200\xc3s200
Device Id: 11101010000010001000000010010011 (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 ซึ่งตามอุปกรณ์ที่ต้องการในโฟลเดอร์

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

ตัวอย่างเช่นต้องการไฟล์ของ xc01s จะหาได้ที่โฟลเดอร์

C:\Xilinx\14.7\ISE\_DS\ISE\xcf\data\xc01s.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 และกด enter จะได้ผลลัพธ์ตามรูปตัวอย่างด้านล่าง  
หมายเหตุ <Directory> ให้แทนด้วย ชื่อDirectory ที่เก็บไฟล์ xc01s.bsd ไว้ โดยในตัวอย่างเก็บไว้ที่ D:\bsd ก็จะพิมพ์คำสั่งเป็น bsdl path d:\bsd

```
jtag> bsdl 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\xc01s.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. การโปรแกรมลงแฟร์ช สำหรับบอร์ด SURVEYOR 3 และ SURVEYOR 6

- Download file Test.svf จาก

<https://drive.google.com/drive/folders/1MZYrACJDneNtqHNfVotfy4cPsq7Uq8s>

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

```
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/1MZYrACJDneNtqHNfVotfy4cPsq7Uq8s>

- 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 ในการโปรแกรมครั้ง ต่อไป

```
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 สำหรับอร์ด Spartan 6

- พิมพ์คำสั่ง svf <directory> ไฟล์สกุล svf สำหรับโปรแกรมPROM> progress stop ตามด้วยการ กด enter สำหรับสั่งโปรแกรมสู่เป้าหมาย
- พิมพ์คำสั่ง instruction BYPASS ตามด้วยการกด enter สำหรับเตรียม register ในการโปรแกรมครั้ง ต่อไป

```
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

คู่มือการใช้งานโปรแกรม

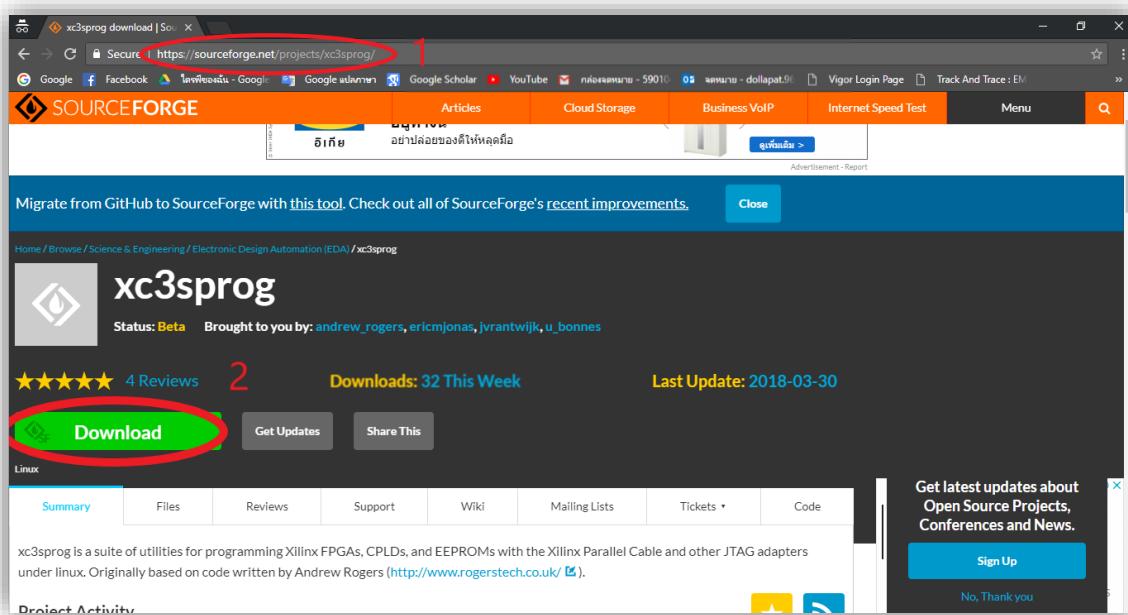
XC3SPROG

ในการโปรแกรมลงบอร์ด FPGA

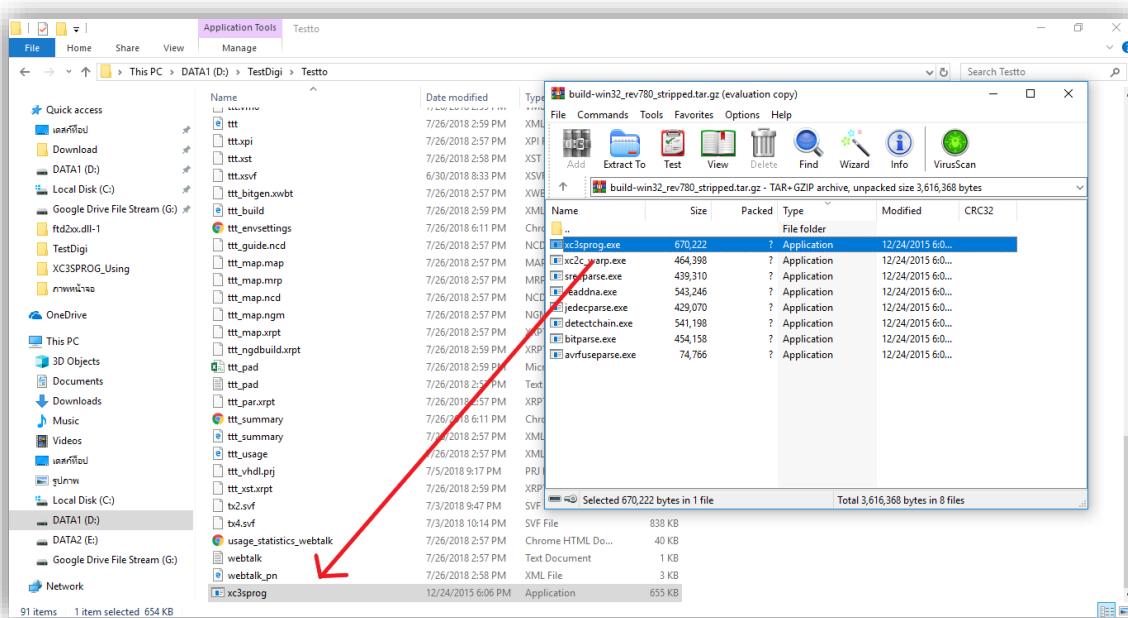
ผ่านโมดูล FT2233HL

## ขั้นตอนการเตรียมโปรแกรม xc3sprog

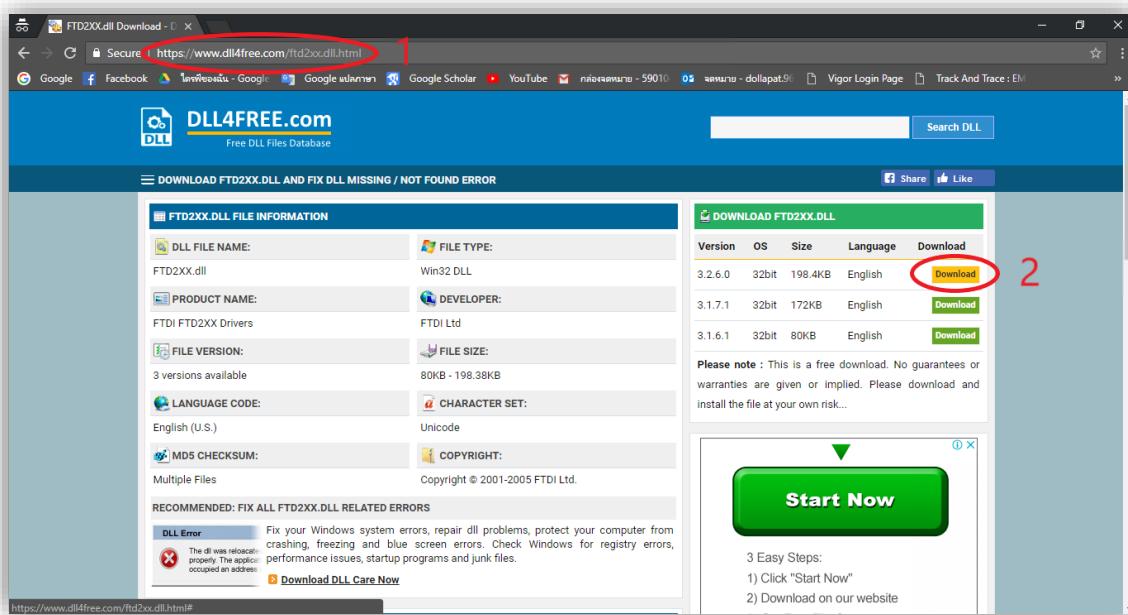
- ดาวน์โหลดโปรแกรม xc3sprog จาก <http://sourceforge.net/projects/xc3sprog/>



- เปิดไฟล์ที่ดาวน์โหลดและนำเข้าไฟล์ xc3sprog วางลงในไฟล์เดอร์ที่มีไฟล์นามสกุล .bit (สำหรับ FPGA) ที่ต้องการโปรแกรมลงบอร์ด

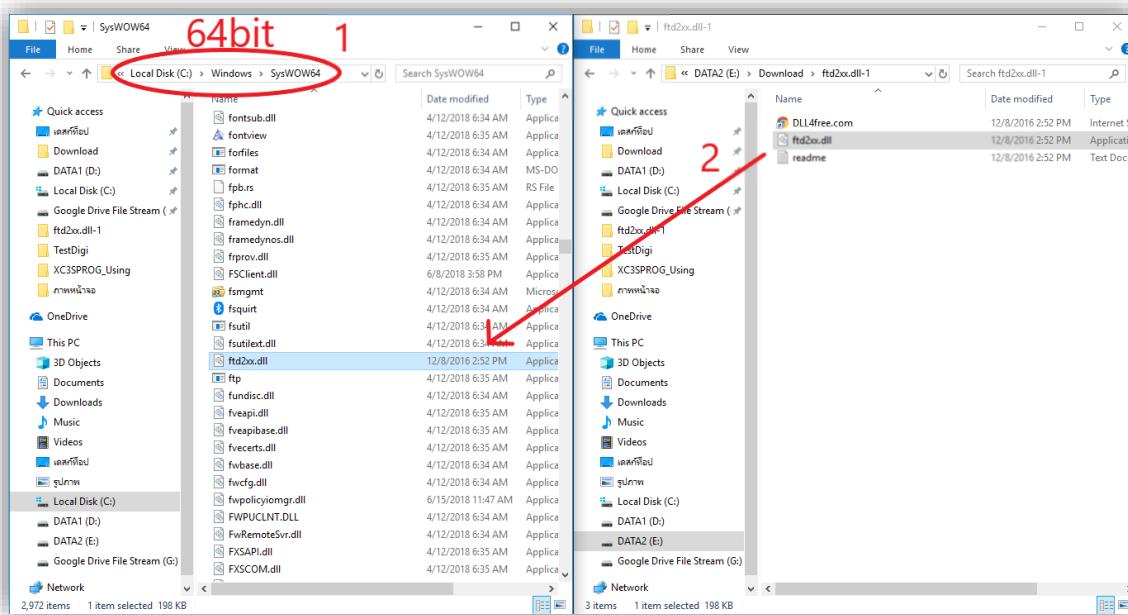


3. ดาวน์โหลด ftd2xx.dll จาก <https://www.dll4free.com/ftd2xx.dll.html>

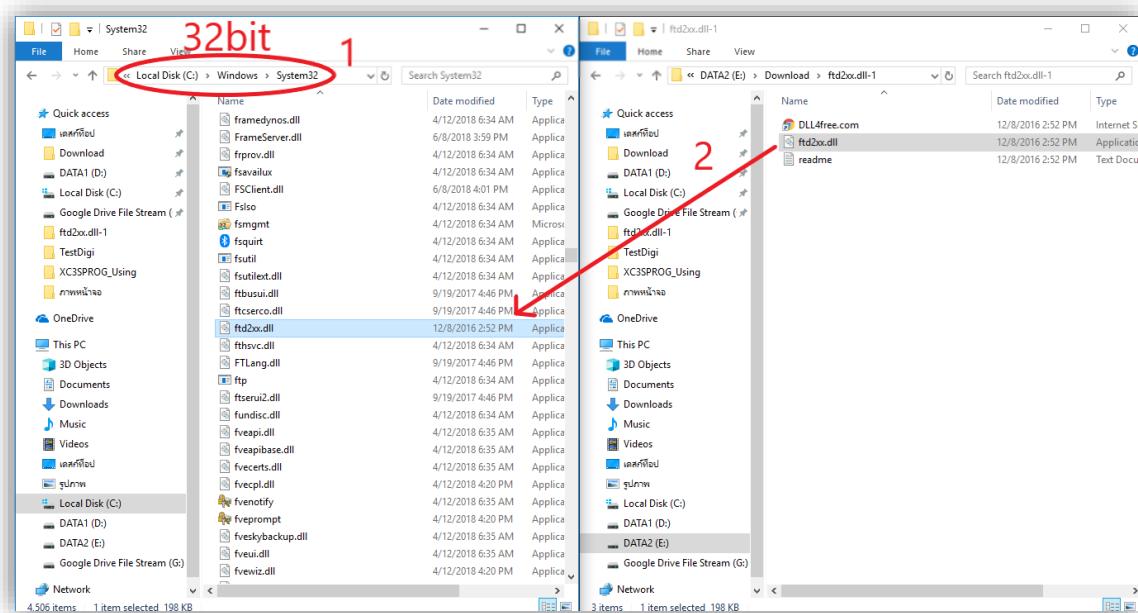


4. เปิดไฟล์ที่ดาวน์โหลดและนำเฉพาะไฟล์ ftd2xx.dll ไปวางในตำแหน่งดังนี้

- สำหรับ 64 bit นำไปยัง "C:\Windows\SysWOW64"



- สำหรับ 32 bit นำไปปั้น "C:\Windows\System32\"



\*หากต้องการโปรแกรมลง PROM ในบอร์ด FPGA Spartan6 XC6SLX9 TQG144 ให้ทำต่อข้อ 5. ถึง 6.

5. ดาวน์โหลด spiflasherLX9.bit จาก <https://www.scarabhardware.com/forums/topic/boot-from-on-board-spi-flash/>

January 6, 2015 at 3:05 pm

Bitfiles needed to write data to SPI flash:

For LX25:  
[spiflasherLX25.bit](#)

For LX9 (please note that I don't know if it works as I have LX25 board):  
[spiflasherLX9.bit](#) 2

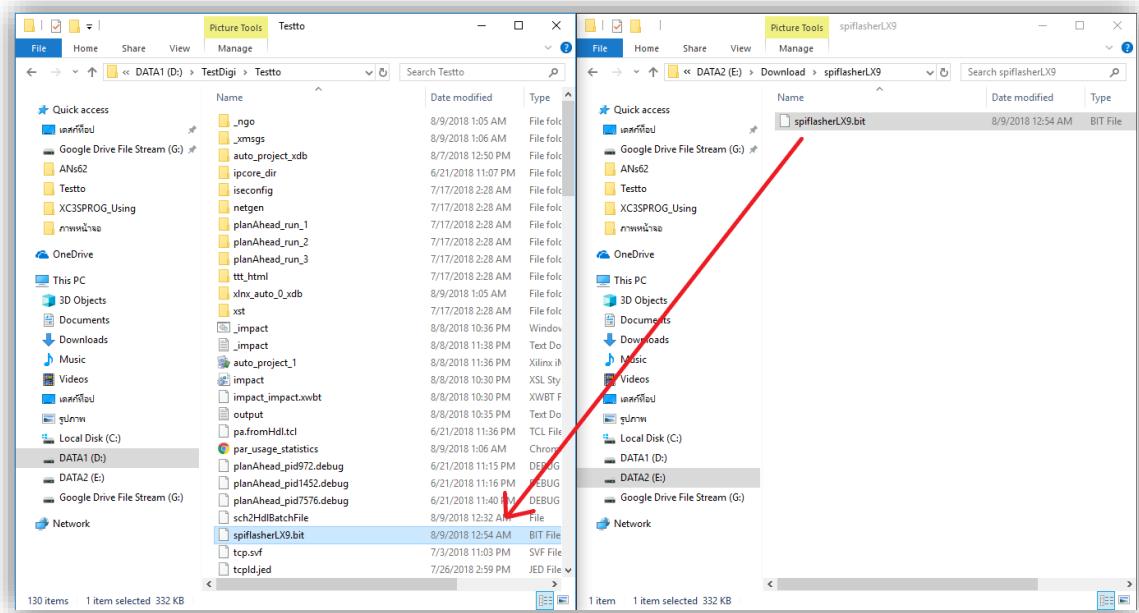
As you can see I've also created a fork of miniProg with additional button which can be used to program SPI flash. You need to have spiflasher.bit file placed in current directory, just copy and rename LX25 or LX9 version depending on your board. This is rather temporary solution until someone will add this functionality to original miniProg.

January 6, 2015 at 3:43 pm

Thank you very much for your support, it works now!

Cheers,

6. นำ spiflasherLX9.bit ที่ดาวน์โหลด วางลงในโฟลเดอร์ที่มีไฟล์นามสกุล .bit (สำหรับ FPGA) ที่ต้องการโปรแกรมลง PROM

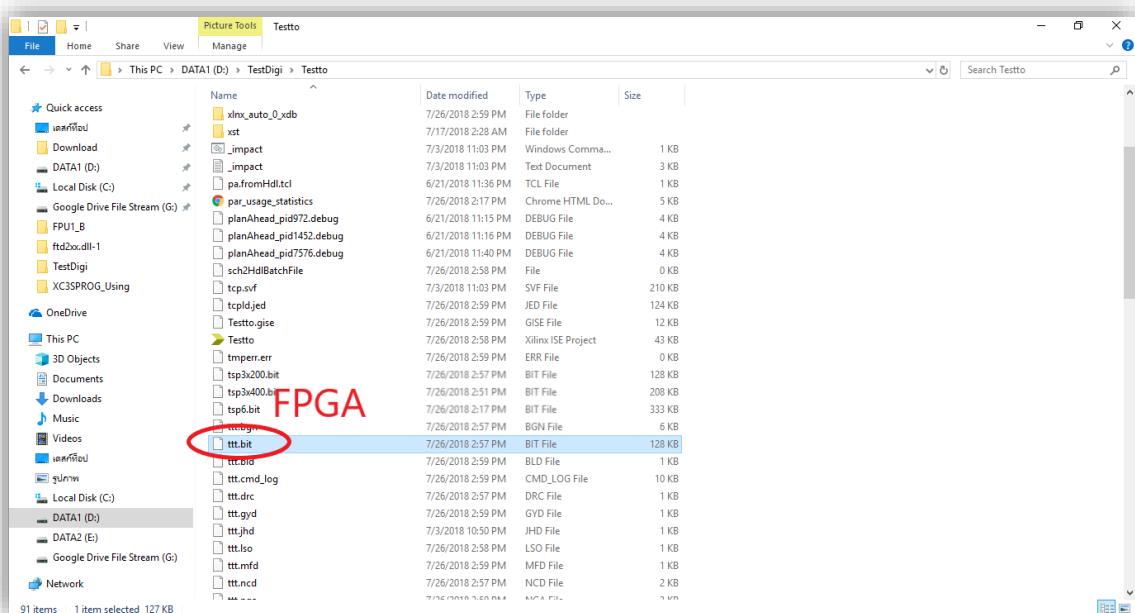


## ขั้นตอนการโปรแกรมลง flash สำหรับบอร์ด FPGA

- ต้องมีไฟล์ที่ต้องการโปรแกรมลงบอร์ด โดย

- สำหรับ FPGA เป็นไฟล์นามสกุล .bit

หมายเหตุ เตรียมไฟล์ svf ให้เรียบร้อยก่อนทำการ flash program ลงบอร์ด



- เตรียม xc3sprog ตามขั้นตอนการเตรียมโปรแกรมข้อ 1. ถึง 4. และทำการเปิดโปรแกรม Notepad

หรือโปรแกรมใด ๆ เพื่อสร้าง batch ไฟล์

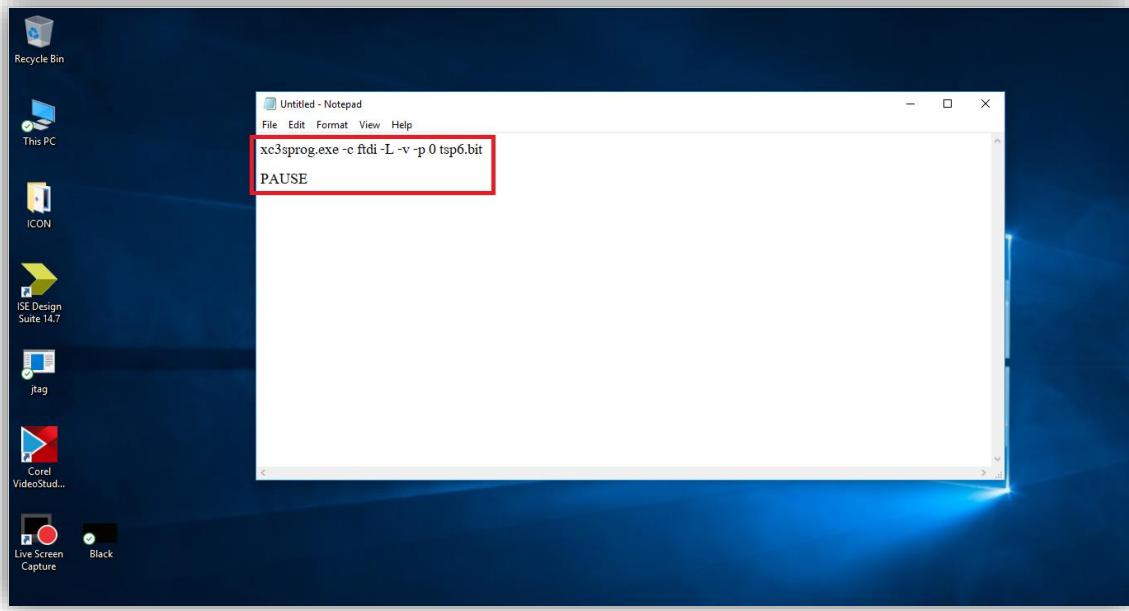
- ในไฟล์ดังกล่าวให้ทำการเลือกใช้คำสั่งตามชนิดบอร์ด ดังนี้

➤ FPGA Sparten6 XC6SLX9 TQG144 : “xc3sprog.exe -c ftdi -L -v -p 0 **file\_name.bit**”

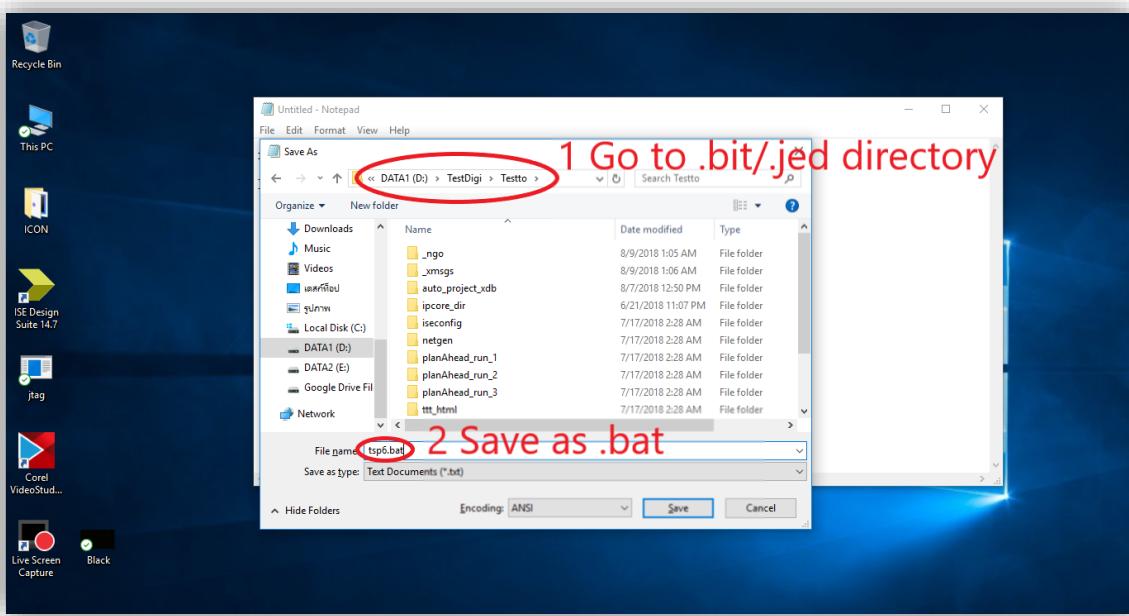
โดย **file\_name** คือชื่อไฟล์ .bit ที่จะโปรแกรมลงบอร์ด

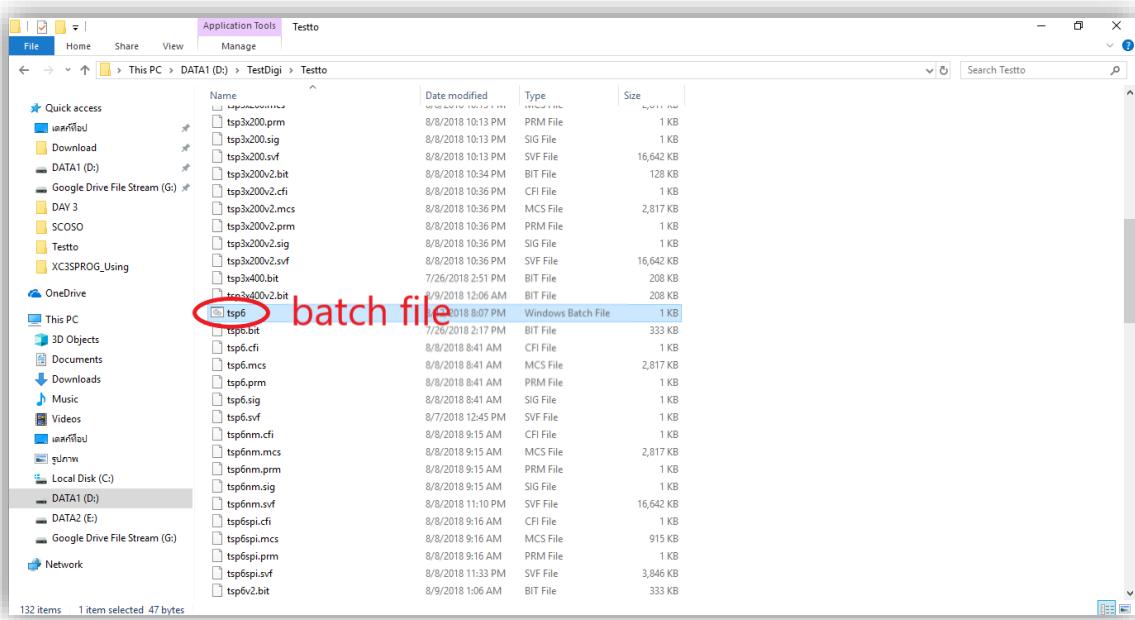
และตามด้วยคำสั่ง “PAUSE” ในบรรทัดถัดมา เพื่อให้หน้าจอโปรแกรมยังแสดงอยู่หลังจากรันเสร็จ

ตัวอย่างการโปรแกรมไฟล์ชื่อ tsp6.bit ลง flash ในบอร์ด FPGA Spartan6 XC6SLX9 TQG144



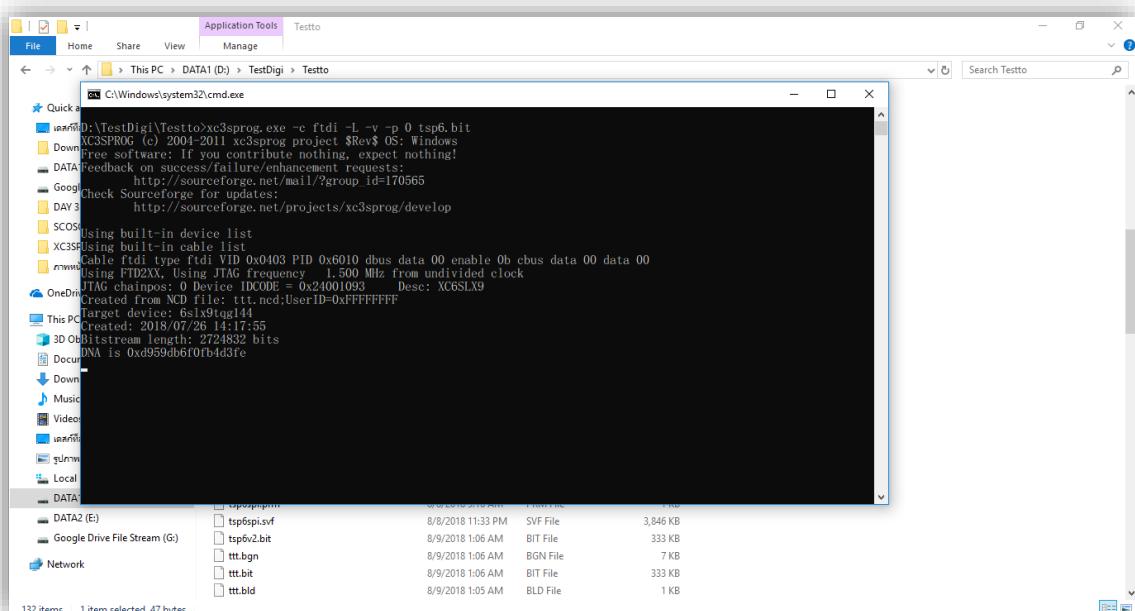
4. บันทึกไฟล์ในไฟล์เดอร์ที่มีไฟล์นามสกุล .bit โดยต้องบันทึกเป็นนามสกุล .bat

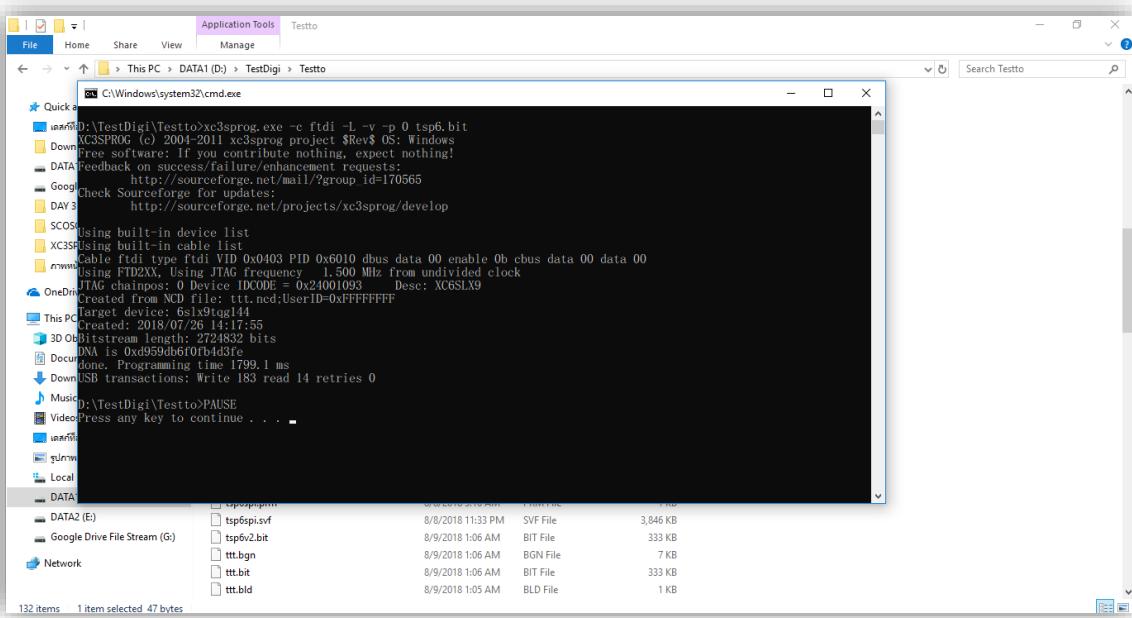




5. ตรวจสอบให้แน่ใจว่าเสียบสาย USB ของบอร์ดเข้ากับคอมพิวเตอร์แล้ว (พบรหัสของบอร์ด)

6. รัน batch file ที่สร้างไว้ เพื่อโปรแกรม flash





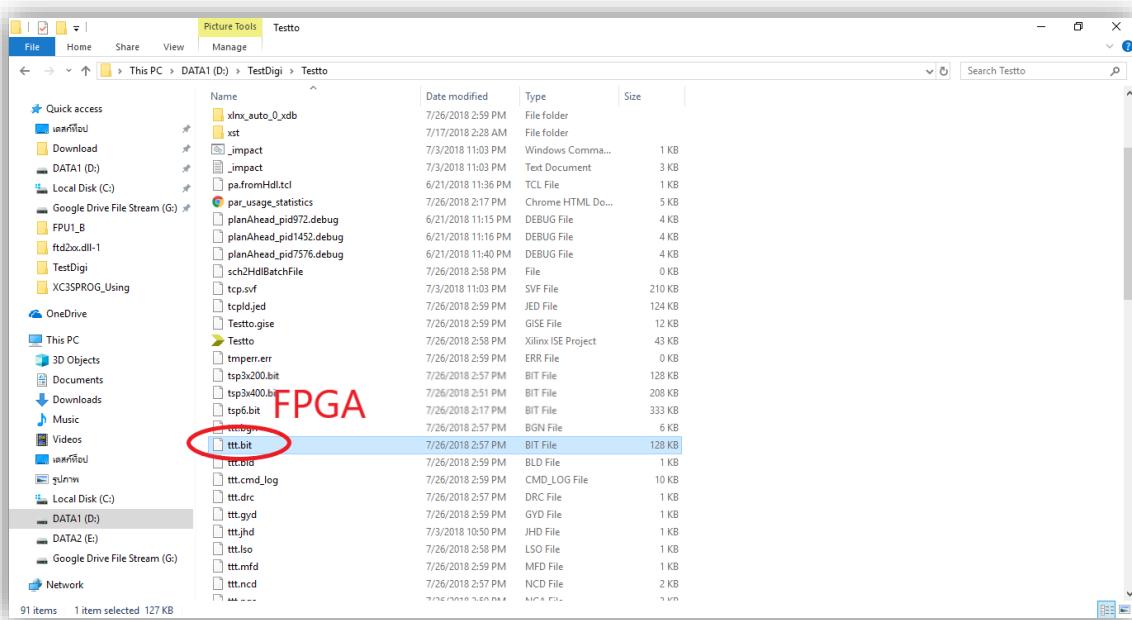
\*หากต้องการโปรแกรมอีกครั้งในกรณีแก้ไขโปรแกรม (ซื้อไฟล์ .bit คงเดิม) ให้ทำเฉพาะข้อ 5. ถึง 6.

## ขั้นตอนการโปรแกรมลง PROM สำหรับบอร์ด FPGA

- ต้องมีไฟล์ที่ต้องการอัปโหลดขึ้นบอร์ด โดย

- สำหรับ FPGA เป็นไฟล์นามสกุล .bit

หมายเหตุ เตรียมไฟล์ svf,mcs ให้เรียบร้อยก่อนทำการ prom program ลงบอร์ด



- เตรียม xc3sprog ตามขั้นตอนการเตรียมโปรแกรม

- ในบอร์ด FPGA Spatan3 เตรียมตามข้อ 1. ถึง 4.
- ในบอร์ด FPGA Spatan6 เตรียมตามข้อ 1. ถึง 6.

และทำการเปิดโปรแกรม Notepad หรือโปรแกรมใด ๆ เพื่อสร้าง batch ไฟล์

- โดยในไฟล์ดังกล่าวให้ทำการเลือกใช้คำสั่งตามชนิดบอร์ด ดังนี้

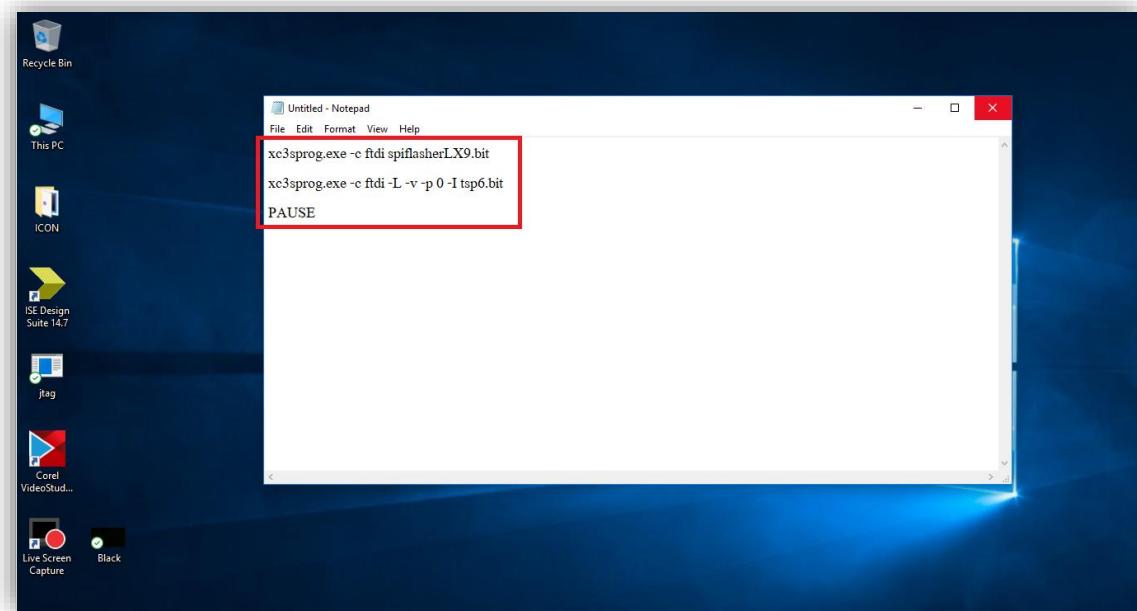
➤ FPGA Spatan6 XC6SLX9 TQG144 : “xc3sprog.exe -c ftdi spiflasherLX9.bit” และ

“xc3sprog.exe -c ftdi -L -v -p 0 -l **file\_name.bit**”

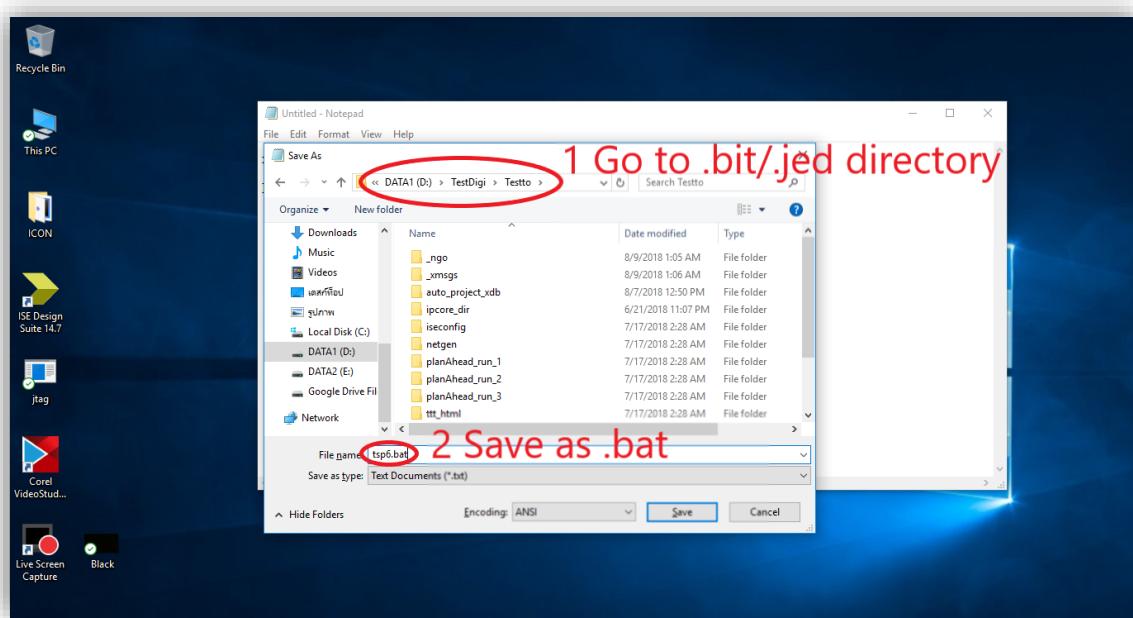
โดย **file\_name** คือชื่อไฟล์ .bit ที่จะโปรแกรมลงบอร์ด

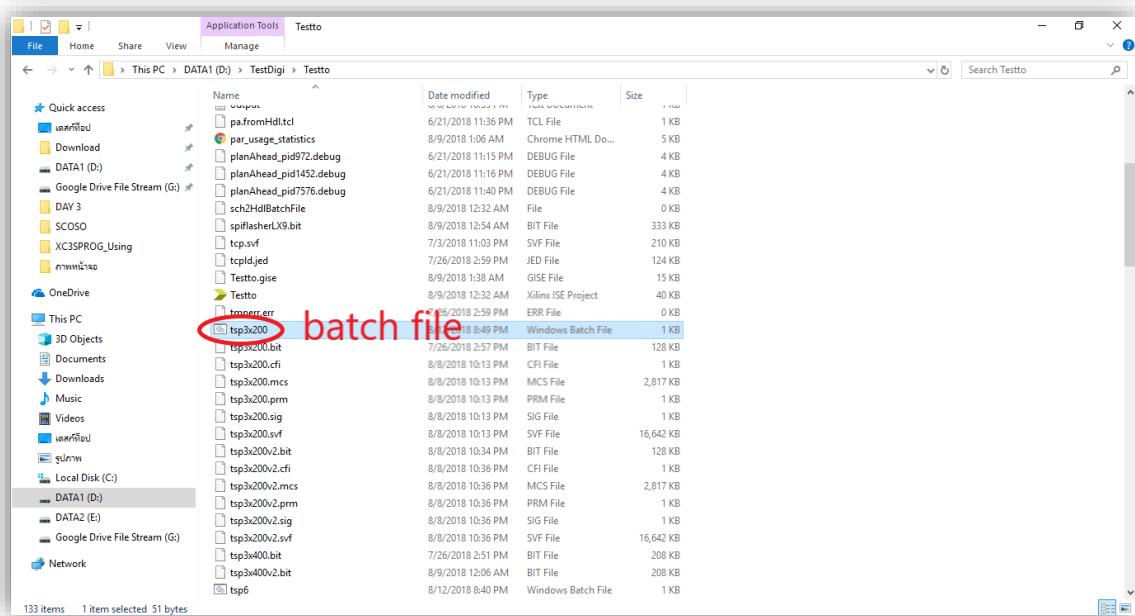
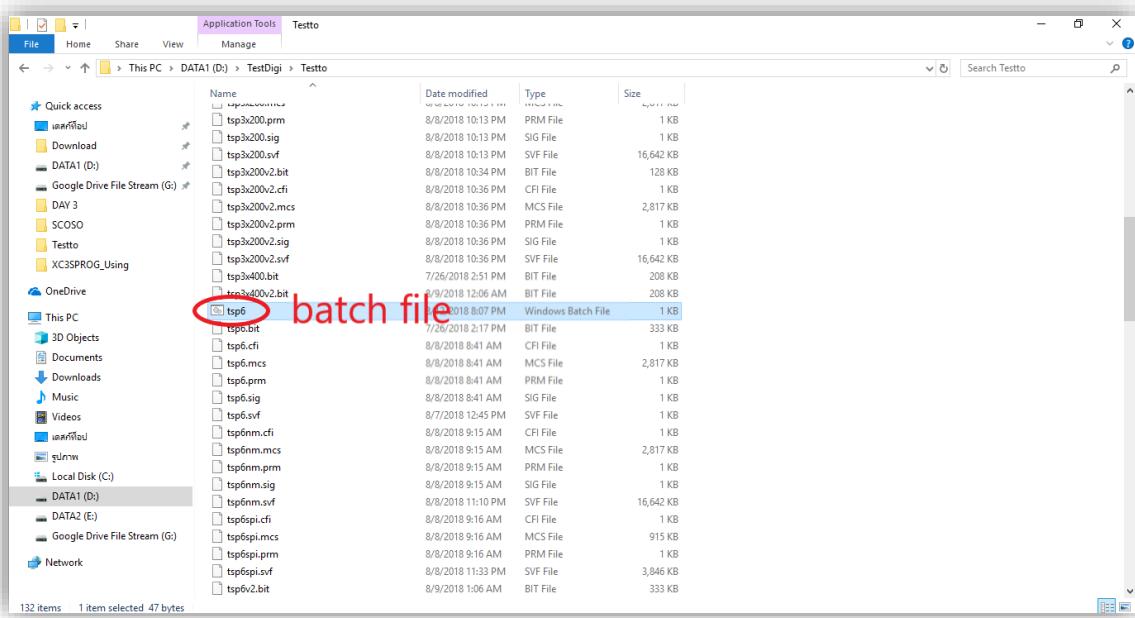
และตามด้วยคำสั่ง “PAUSE” ในบรรทัดถัดมา เพื่อให้หน้าจอโปรแกรมยังแสดงอยู่หลังจากรันเสร็จ

ตัวอย่างการโปรแกรมไฟล์ชื่อ tsp6.bit ลง PROM ในบอร์ด FPGA Sparten6 XC6SLX9 TQG144



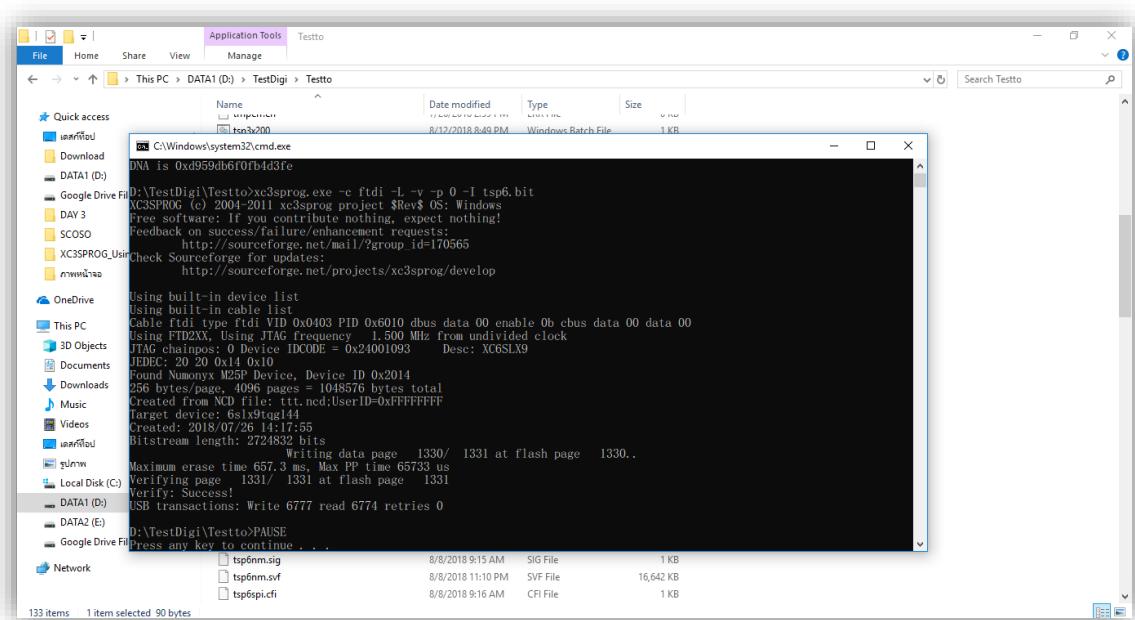
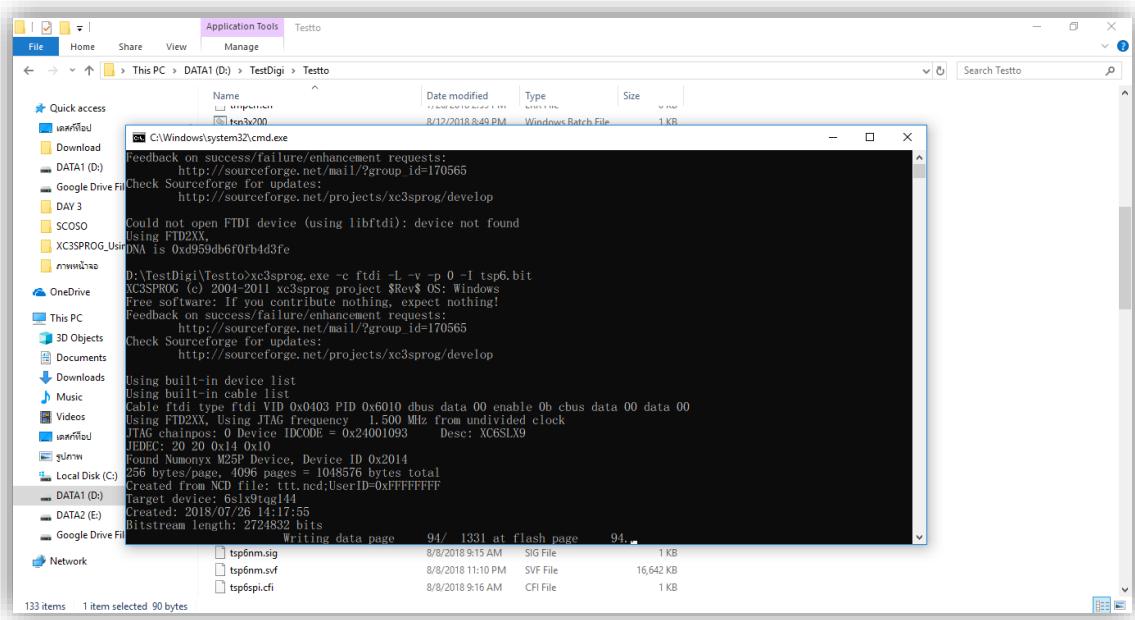
4. บันทึกไฟล์ในไฟล์เดอร์ที่มีไฟล์นามสกุล .bit โดยต้องบันทึกเป็นนามสกุล .bat





5. ตรวจสอบให้แน่ใจว่าเสียบสาย USB ของบอร์ดเข้ากับคอมพิวเตอร์แล้ว (พบพอร์ตของบอร์ด)

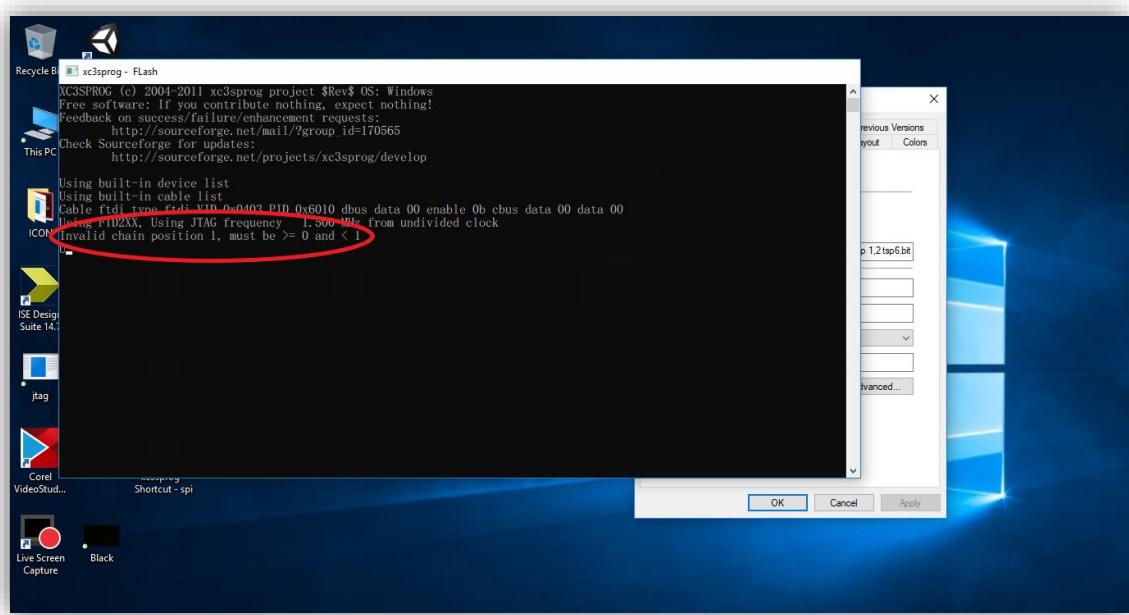
6. รัน batch ไฟล์ที่สร้างไว้ เพื่อโปรแกรม PROM และทำการรีเซ็ตบอร์ด 1 รอบ



\*หากต้องการโปรแกรมอิกครั้งในกรณีแก้ไขโปรแกรม (ซื้อไฟล์ .bit/.jed คงเดิม) ให้ทำเฉพาะข้อ 5. ถึง 6.

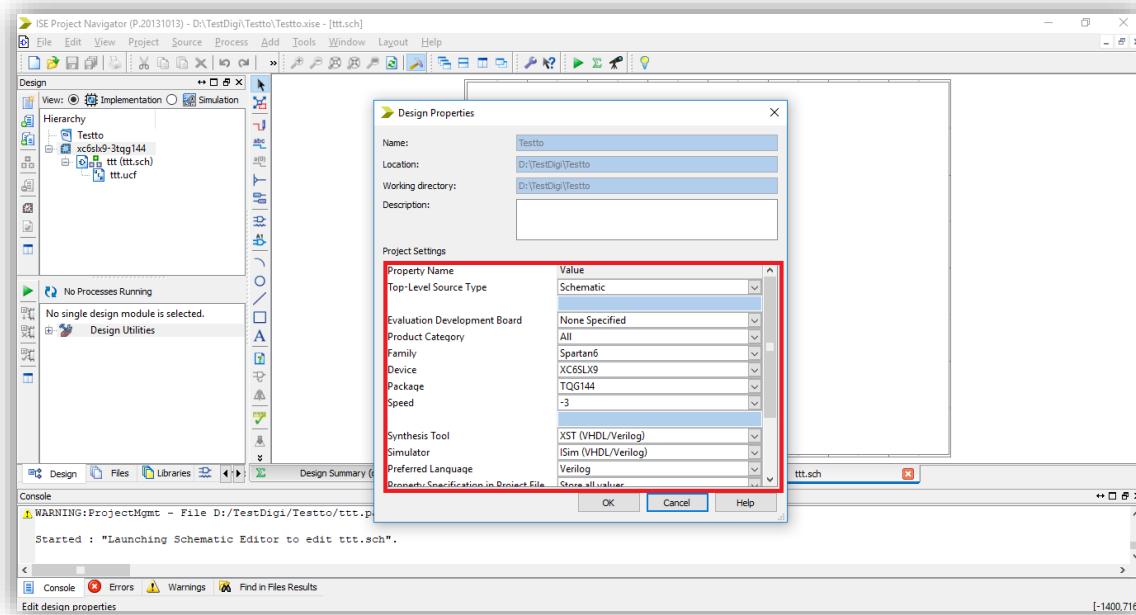
## ปัญหาที่อาจพบและข้อควรระวัง

- สำหรับบอร์ด CPLD จะใช้ไฟล์ .jed และไฟล์ .bit เช่นเดียวกัน
  - การใช้ -p ในบอร์ดชนิดต่าง ๆ หรือในการโปรแกรมแต่ละแบบ (flash / PROM) ต้องใช้ไฟล์ชนิดเดียวกัน
- \*ตัวเลขหลัง -p คือ เลขของอุปกรณ์ที่ต้องการอัพโหลดโปรแกรม หากเลือกไม่ถูกต้องจะพบปัญหาดังภาพ

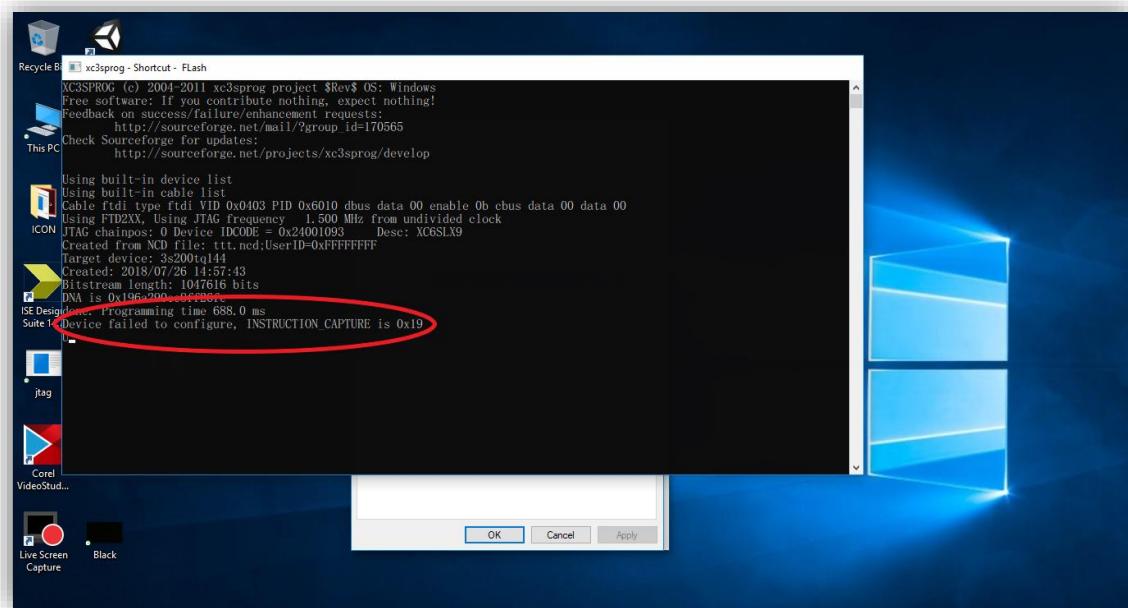


- ในการสร้างไฟล์ .bit หรือ .jed ต้องแน่ใจว่าตั้งค่า Design Properties ตรงกับชนิดบอร์ดที่ใช้

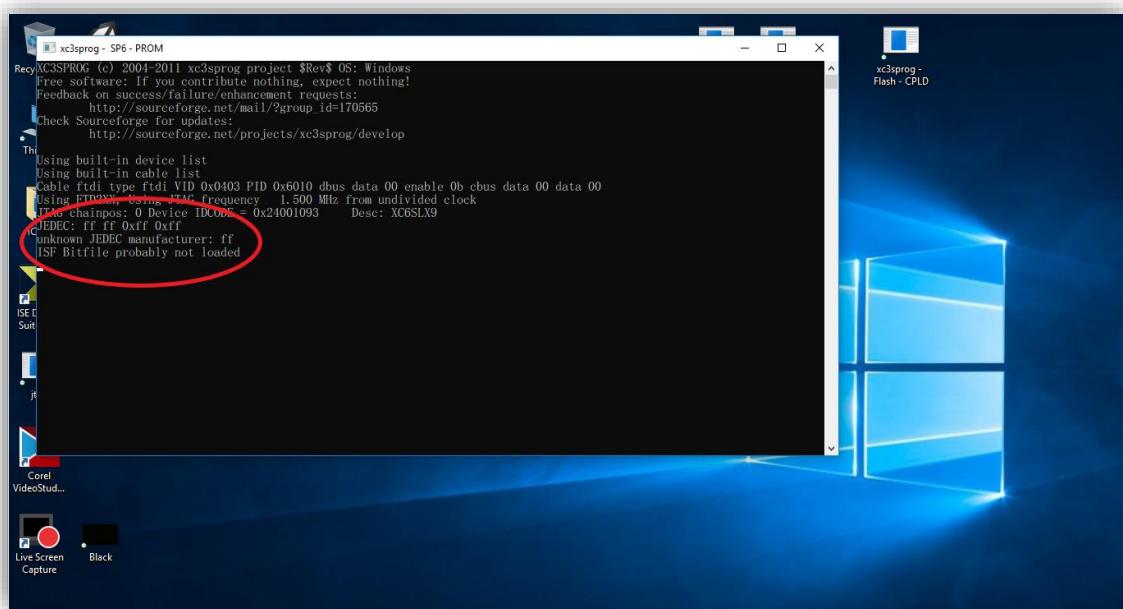
## ตัวอย่างการตั้งค่า Design Properties เพื่อโปรแกรมลงบอร์ด FPGA Spartan6 XC6SLX9 TQG144



\*การสร้างไฟล์ .bit หรือ .jed หากตั้งค่า Design Properties ไม่ตรงกับชนิดบอร์ดที่ใช้ จะพบปัญหาดังภาพ



- ในการอัพโหลดโปรแกรมเข้า PROM ในบอร์ด FPGA Spartan6 XC6SLX9 TQG144 หากไม่ใส่คำสั่ง “xc3sprog -c ftdi spiflasherLX9.bit” จะเกิดปัญหาดังภาพ



- หากไม่ใส่คำสั่ง “PAUSE” ในตอนสร้าง batch ไฟล์ โปรแกรมจะปิดลงทันทีหลังรันเสร็จ

## อ้างอิง

<http://pldkit.com/fpu1>

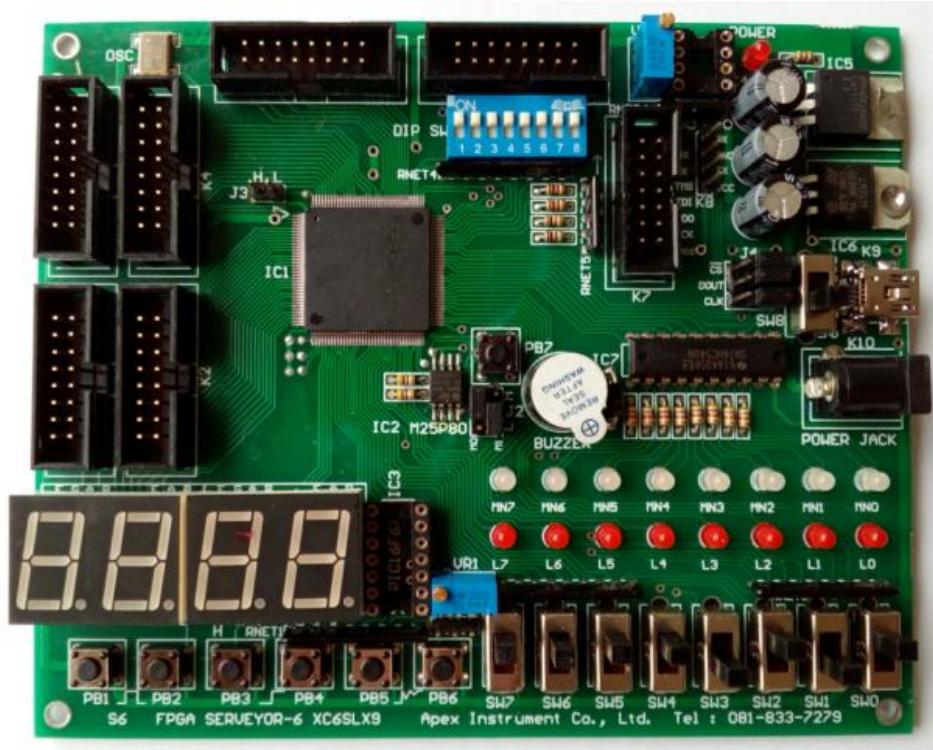
<http://xc3sprog.sourceforge.net/>

<http://xc3sprog.sourceforge.net/manpage.php>

<https://sourceforge.net/p/xc3sprog/bugs/15/>

<https://www.scarabhardware.com/forums/topic/boot-from-on-board-spi-flash/>

## ໂປຣໜ້ວ່າ FPGA Surveyor-6 XC6SLX9



ບຸດທດລອງບອໍດ FPGA ຍື້ຫ້ອ ເອເກີ ອິນສຕຽມເນັດ ຮູ່ນ FPGA Surveyor-6 XC6SLX9

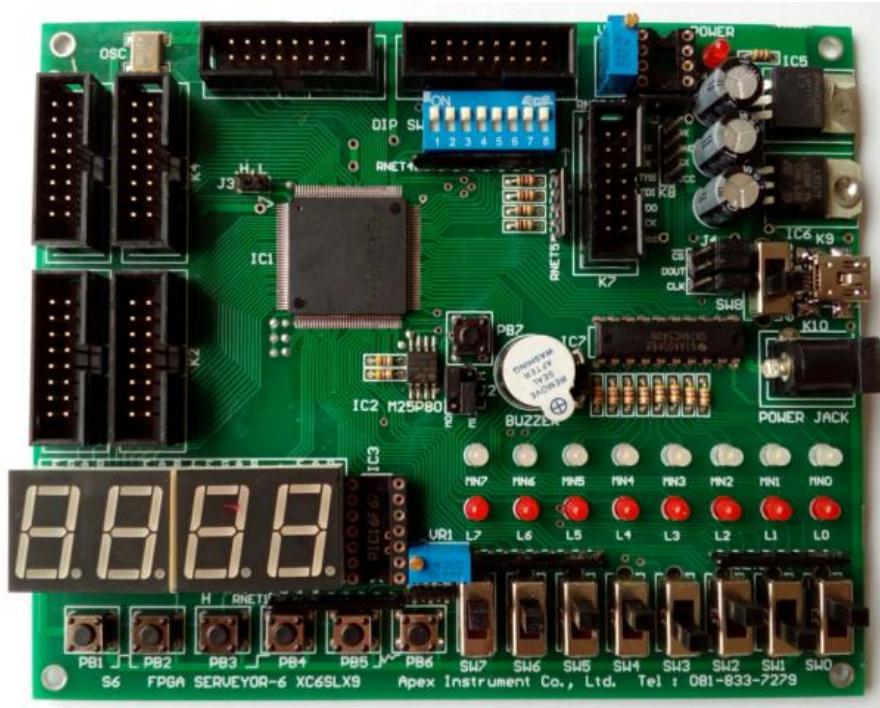
ບອໍດ FPGA Surveyor-6 XC6SLX9 ພ້ອມສາຍດາວນີ້ໂຫດ JTAG ແບບ USB ປະກອບດ້ວຍ

- FPGA : Spartan-6 ຂອງ Xilinx ເບື້ນ XC6SLX9 144pin ພ້ອມ Flash PROM ກາຍນອກ
- 7-Segment 4 ທັກ
- LED 3 ສຖານະ 8 ດວງ
- LED 2 ສຖານະ 8 ດວງ
- Logic Switch (Slide Switch) 8 ຕ້າ (ມີ SW7 ໃຊ້ input ຮ່ວມກັບ PB6)
- 6 Expansion port (48 Bits)
- ປຸ່ມກົດແບບ One-Shot, Bounceless ແລະ Bounce ອຍ່າງລະ 2 ຕ້າ
- DIP Switch 8 ດວງ ຮ່ວມກັບ Expansion port K6
- Buzzer ຈຳນວນ 1 ຕ້າ
- RS-232C 1 Port
- 20 Mhz Oscillator
- Clock generator :1-100Hz ແລະ 10-1KHz
- ສາຍດາວນີ້ໂຫດ JTAG ແບບ USB
- ອະແດພເຕອ້ນ 9VDC output / 220VAC input

## คู่มือการใช้งานบอร์ด FPGA Surveyor-6 XC6SLX9 ยี่ห้อ เอเพก อินสตรูเม้นต์

บอร์ด FPGA Surveyor-6 XC6SLX9 พร้อมสายดาวน์โหลด JTAG และ 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 และ USB
- อะแดปเตอร์ 9VDC output / 220VAC input



รูปที่ 1 บอร์ด FPGA Surveyor-6 XC6SLX9

FPGA เบอร์ XC6SLX9-2TQG144C

Flash PROM เบอร์ M25P80-VMW6TG

รายการ Pin List

K1 CONNECTOR					
Descriptions	FPGA Pinout	K1 Pinout	K1 Pinout	FPGA Pinout	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 Pinout	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 Pinout	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 Pinout	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 Pinout	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 Pinout	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

## การใช้งาน Logic Trainer

นักศึกษาจะได้ฝึกการใช้งานเครื่องมือและรู้จักกับอุปกรณ์ไอซีโลจิกเกตเบื้องต้น โดยเครื่องมือที่ใช้คือ ลอจิกเทรนเนอร์(Logic Trainer) สำหรับໄອซีโลจิกเกตที่ใช้ทดลองเป็นชนิด AND, OR, NOT(Inverter), XOR และ NAND



รูปที่ 1 ลอจิกเทรนเนอร์

## Logic Trainer

1. **Power Supply** เป็นส่วนจ่ายแรงดันให้กับอุปกรณ์ที่ใช้ในการทดลอง แรงดันที่จ่ายมี 4 ระดับคือ  $+5V$ ,  $-5V$ ,  $+12V$  และ  $-12V$  ส่วน  $0V$  คือ Ground (GND) สำหรับในการทดลองนี้เราใช้แรงดัน  $+5V$  เท่านั้น หากในวงจรที่นักศึกษาทำลังต่อเกิดการลัดวงจรจะรบกวนการทำงาน ดวงไฟโอเวอร์โหลด (Overload) จะสว่างขึ้น นักศึกษาต้องรีบปลดสายจากวงจรที่เชื่อมต่อกับ Power Supply และกดปุ่มรีเซ็ต (Reset) หรือปิดเครื่องแล้วเปิดใหม่ แล้วตรวจหาสาเหตุที่ทำให้เกิดการลัดวงจร
2. **Logic Switch** เป็นส่วนที่ใช้ป้อนอินพุตให้กับวงจรโลจิก ประกอบด้วยสวิตซ์โยกและดวงไฟแสดงสถานะจำนวน 8 หลัก จาก 0 ถึง 7 โยกสวิตซ์ไปที่ ON เพื่อป้อนอินพุตโลจิก “1” (แรงดัน 5V) ให้กับวงจร โดยไฟแสดงสถานะสี แดงจะสว่าง โยกสวิตซ์ไปที่ OFF เพื่อป้อนอินพุตโลจิก “0” (แรงดัน 0V) ให้กับวงจร โดยไฟแสดงสถานะ สีเขียวจะสว่าง
3. **Logic Monitor** เป็นส่วนที่ใช้ตรวจสอบค่าโลจิก โดยใช้หลอดไดโอดเปล่งแสง (LED) จำนวน 8 หลอด สำหรับแสดงผล หาก LED สว่างเป็นสีแดง ผลลัพธ์คือโลจิก “1”  
หาก LED สว่างเป็นสีเขียว ผลลัพธ์คือโลจิก “0”  
หาก LED ไม่ติด หมายถึงไม่มีแรงดัน

## ไอซีลوجิกเกต

ภายในไอซีลوجิกเกตจะบรรจุเกตต่างๆ ได้แก่ AND, OR, NOT และ XOR เป็นต้น โดยไอซีลوجิกเกตมีมากมายหลายชนิดและหลากหลายแบบ ในการทดลองนี้จะให้นักศึกษารู้จักไอซีลوجิก 6 ชนิดคือ

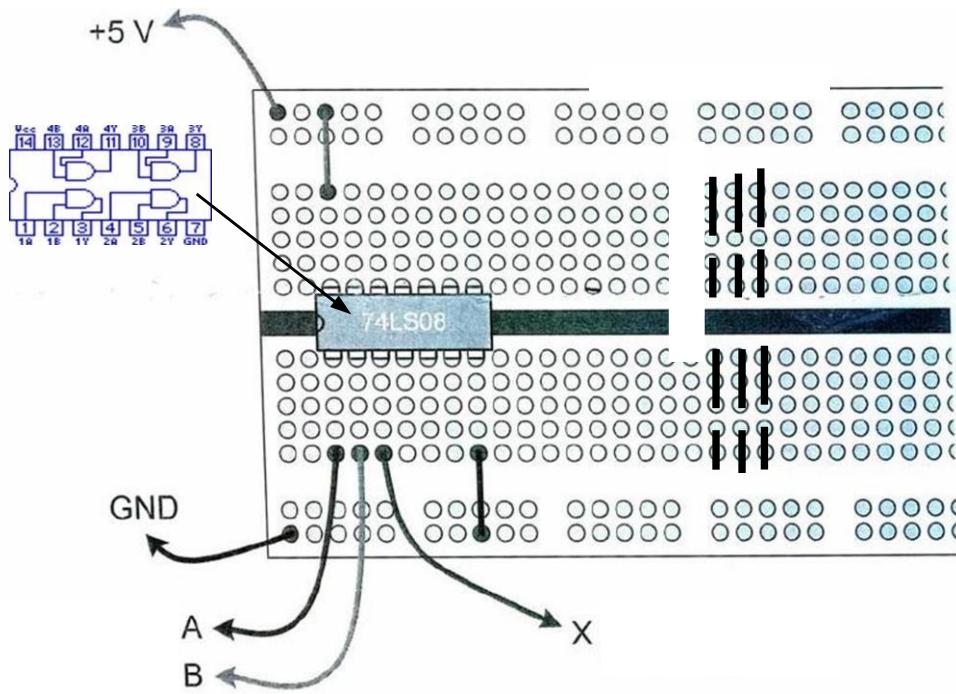
ลำดับ	สัญลักษณ์/เบอร์ไอซี	Truth Table	รายละเอียด
1	 	$\begin{array}{l l l} \text{a} & \text{b} & \text{y} \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{array}$	AND gate จากรูปมีสองอินพุต หนึ่งเอาต์พุต ลักษณะของเอาต์พุตมีค่าเป็น “1” ก็ต่อเมื่ออินพุต ทั้งหมดเป็น “1” เท่านั้น กรณีอื่นๆ ค่าเอาต์พุต เป็น “0”
2	 	$\begin{array}{l l l} \text{a} & \text{b} & \text{y} \\ \hline 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{array}$	OR gate จากรูปมีสองอินพุต หนึ่งเอาต์พุต ลักษณะของเอาต์พุต มีค่าเป็น “0” ก็ต่อเมื่อ <sup>อินพุตทั้งหมดเป็น “0”</sup> เท่านั้น กรณีอื่นๆ ค่า <sup>เอาต์พุตเป็น “1”</sup>

ลำดับ	สัญลักษณ์/เบอร์ไอซี	Truth Table	รายละเอียด															
3	  <b>7404</b>	<table border="1"> <tr> <td>a</td> <td>y</td> </tr> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </table>	a	y	0	1	1	0	NOT gate หรือ Inverter มีหนึ่งอินพุต หนึ่งเอ้าต์พุต ผลลัพธ์ของเอ้าต์พุตเป็นส่วนกลับจากอินพุต									
a	y																	
0	1																	
1	0																	
4	  <b>7400</b>	<table border="1"> <tr> <td>a</td> <td>b</td> <td>y</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	a	b	y	0	0	1	0	1	1	1	0	1	1	1	0	NAND gate ลักษณะของสัญลักษณ์คล้ายกับ AND gate แต่ทางด้านเอ้าต์พุตเสมอ 0 เมื่ออินพุตทั้งสองตัวเป็น 1 และเท่ากับ 1 เมื่ออินพุตทั้งสองตัวเป็น 0
a	b	y																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
5	  <b>7402</b>	<table border="1"> <tr> <td>a</td> <td>b</td> <td>y</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	a	b	y	0	0	1	0	1	0	1	0	0	1	1	0	NOR gate ลักษณะของสัญลักษณ์คล้ายกับ OR gate แต่ทางด้านเอ้าต์พุตเสมอ 1 เมื่ออินพุตทั้งสองตัวเป็น 0 และเท่ากับ 0 เมื่ออินพุตทั้งสองตัวเป็น 1
a	b	y																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
6	  <b>7486</b>	<table border="1"> <tr> <td>a</td> <td>b</td> <td>y</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	a	b	y	0	0	0	0	1	1	1	0	1	1	1	0	XOR gate ย่อมาจาก Exclusive-OR gate ในรูป มีสองอินพุต หนึ่งเอ้าต์พุต ลักษณะของเอ้าต์พุตมีค่าเป็น “1” ก็ต่อเมื่ออินพุตไม่เข้าพวก และค่าเอ้าต์พุตเป็น “0” เมื่ออินพุตทุกดัวเป็น “0” ทั้งหมด หรือ อินพุตทุกดัวเป็น “1” ทั้งหมด
a	b	y																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

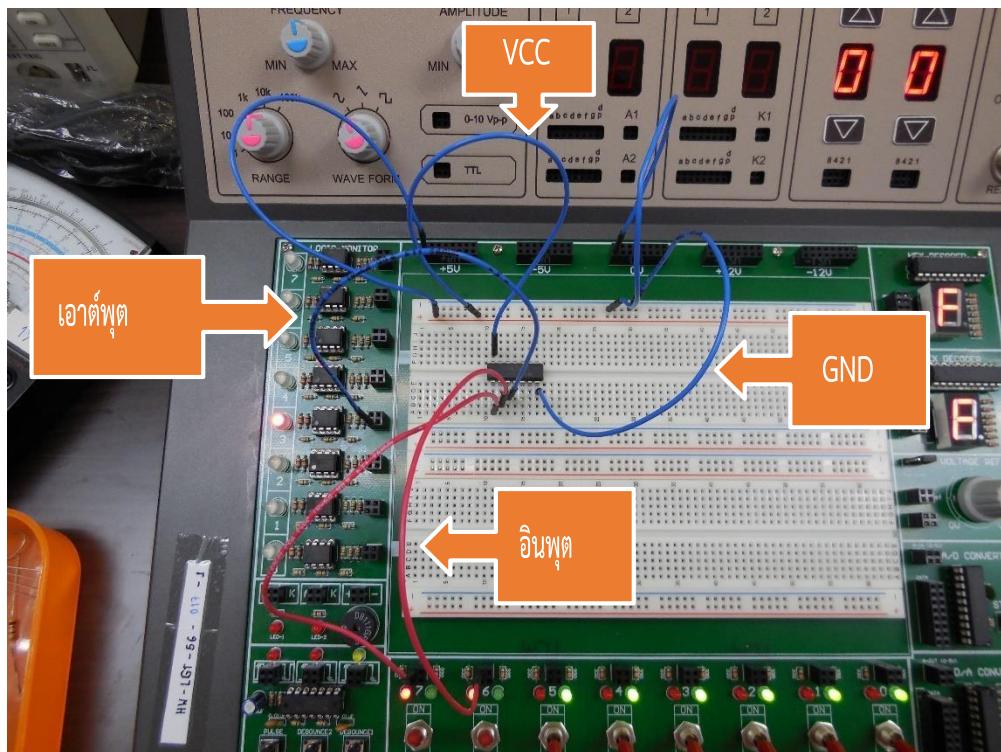
### ข้อควรปฏิบัติ

- ก่อนลงมือทดลอง ต้องตรวจสอบความสมบูรณ์ด้านความปลอดภัย และการทำงานของอุปกรณ์และไอซีเสมอ!
- ขา VCC รับแรงดันที่ป้อนให้แก่ไอซีขนาด +5V ส่วนขา GND เป็นขาราوترต่อกับ 0V หากต่อสลับข้าว ไอซีอาจพังเสียหาย

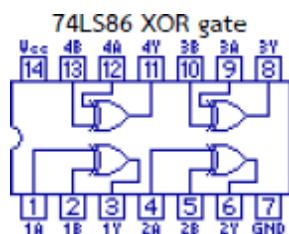
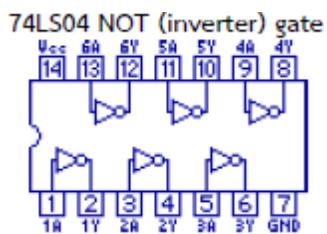
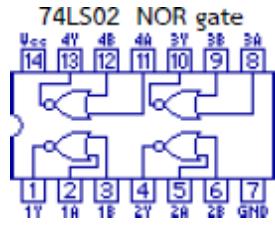
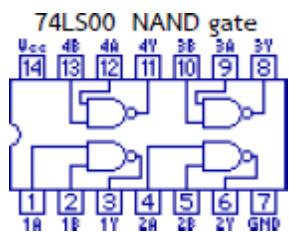
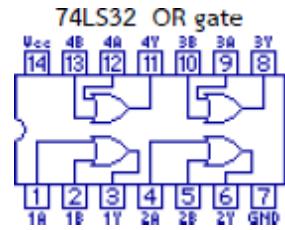
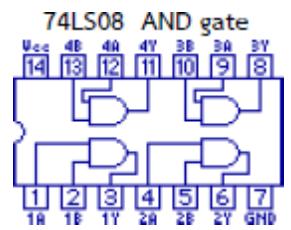
3. ระดับโลจิก “0” (Low) มีแรงดันช่วง 0 - 0.5 V และระดับโลจิก “1” (Hi) มีแรงดันช่วง 2.5 – 5 V
4. การเขื่อมต่อวงจรบนพร็อตบอร์ดมีแนวการเขื่อมต่อในแนวตั้งกับแนวนอน สังเกตจากรูปที่ 2
5. ก่อนการต่อสายต้องตรวจสอบให้แน่ใจว่า ไอซีล็อกจิกเกตที่ใช้เป็นชนิดใด ขนาดนิ่งอยู่ทิศทางใดโดยตรวจสอบได้จากรูปที่ 2 รูปที่ 3 และรูปที่ 4
6. การถอดไอซีออกจากพร็อตบอร์ดให้ใช้ไขควงจัดด้านข้างของไอซีอย่างระมัดระวัง เพื่อป้องกันขาไอซีชำรุดและอุบัติเหตุบาดเจ็บจากขาไอซีทิมแทง



รูปที่ 2 ไดอะแกรมแสดงตัวอย่างการต่อวงจร



รูปที่ 3 แสดงตัวอย่างการต่อวงจร



รูปที่ 4 โครงสร้างภายในของไอซีลوجิกเกต

ระบุหมายเลขไอซีที่ใช้งานของล็อกจิกเกตดังต่อไปนี้ พิริมาณาดสัญลักษณ์ล็อกจิกเกตประกอบ

OR gate หมายเลขไอซี .....	NOR gate หมายเลขไอซี.....
NAND gate หมายเลขไอซี.....	XOR gate หมายเลขไอซี.....

## การใช้งาน Logic probe

### การเชื่อมต่อ Logic probe

1. คลิปประจำเข็มกั๊สสำหรับต่อ กับกราวเดล์ลบ
2. คลิปประจำเข็มกั๊สเดง: เชื่อมต่อกับ supply (logic supply +5 volts หรือ CMOS บางตัวที่สูงถึง 15 volts โดยศึกษาจาก datasheet ของอุปกรณ์ที่ใช้)
3. Probe: ส่วนที่เป็นโลหะใช้สำหรับตรวจสอบวงจร ระมัดระวังว่าหัววัดจะไม่ทำให้เกิดไฟฟ้าลัดวงจรซึ่งอาจทำให้วงจรเสียหายได้ในระหว่างการทดสอบ

### การตั้งค่าเริ่มต้นสำหรับการใช้ logic probe

1. TTL / CMOS: เนื่องจากการแบ่งส่วนกำหนดสถานะ high กับ low ของสองตัวนี้ต่างกัน จึงต้องเลือก logic family ให้ตรงกับอุปกรณ์ที่จะทดสอบ เพื่อให้ได้ผลที่ถูกต้อง
2. MEM / PULSE : การเลือก operational mode โดย ‘Pulse’ ใช้ตรวจสอบสัญญาณที่เป็นลอจิก “0” , “1” หรือสัญญาณพัลส์ ‘MEM’ หรือ memory ใช้จับสัญญาณพัลส์ เช่นตรวจสอบว่ามีสัญญาณพัลส์ เกิดขึ้นหรือไม่

### วิธีการใช้ logic probe

1. เชื่อมต่อคลิปสีดำกับกราวเดล์ หรือ common line ของวงจรที่ต้องการจะทดสอบ
2. เชื่อมต่อคลิปสีแดงกับแหล่งจ่ายที่เป็นบวกของวงจร
3. เลือก logic family ว่าเป็น CMOS หรือ TTL (TTL มากใช้ 5V supply, CMOS ใช้ 5 - 15V)
4. ใช้ probe ต่อไปที่จุดที่ต้องการทดสอบ และสังเกต LED ที่อยู่บนอุปกรณ์

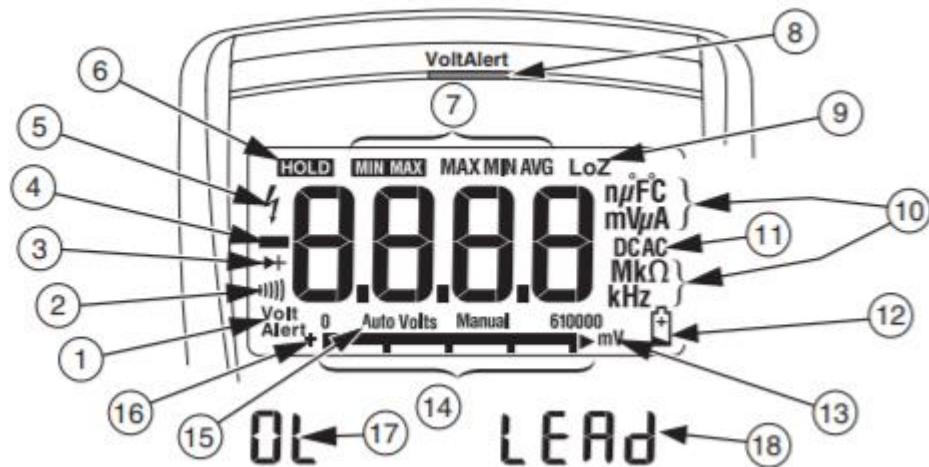
## การใช้งาน Multimeter



Fluke 115 Field Service Technicians Multimeter

## จอแสดงผล

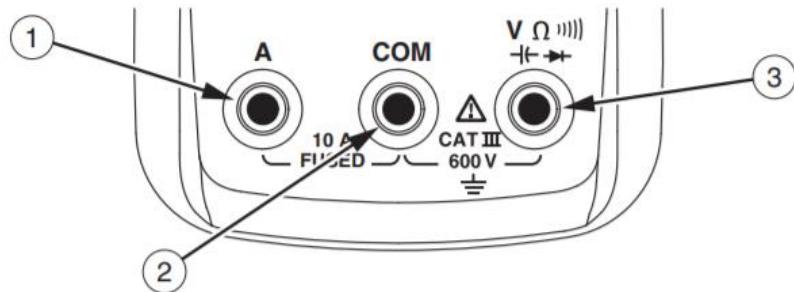
ตารางรายการคุณสมบัติของจอแสดงผลของรุ่น 115



หมายเลข	สัญลักษณ์	ความหมาย
2	⚡	ฟังก์ชันการทำงานของมิเตอร์ถูกตั้งให้เป็นการวัดความต่อเนื่อง
3	➡	ฟังก์ชันการทำงานของมิเตอร์ถูกตั้งให้เป็นการทดสอบค่าไดโอด
4	-	อินพุทมีค่าติดลบ
5	⚡	แรงดันไฟฟ้าขนาดที่ไม่ปลอดภัย วัดค่าแรงดันไฟฟ้าอินพุทได้มากกว่า เท่ากับ 30 โวลต์ หรือเกิดสภาพว่าค่าแรงดันไฟฟ้าเกินกำลัง (OL)
6	<b>HOLD</b>	เปิดการใช้งานฟังก์ชันค้างหน้าจอแสดงผล จะจะค้างเพื่อแสดงค่าปัจจุบัน ที่อ่านได้
7	<b>MIN MAX MAX MIN AVG</b>	เปิดการใช้งานโหมดบันทึกค่า MIN MAX AVG และแสดงค่าสูงสุด ค่าต่ำสุด ค่าเฉลี่ยหรือค่าปัจจุบันที่อ่านได้
9	<b>LoZ</b>	มิเตอร์กำลังวัดค่าแรงดันไฟฟ้าหรือค่าประจุไฟฟ้าด้วยอิมพิเดนซ์อินพุทที่ มีค่าต่ำ
10	<b>nF mV μA MkW kHz</b>	หน่วยการวัด
11	<b>DC AC</b>	กระแสตรงหรือกระแสสลับ

12		การเตือนแบตเตอรี่อ่อน
13		แสดงย่านการวัดของมิเตอร์ที่ถูกเลือกใช้
14	(กราฟแท่ง)	การแสดงผลแบบนาลอก
15	Auto	ย่านการวัดอัตโนมัติ มิเตอร์จะเลือกย่านการวัดให้โดยอัตโนมัติ เพื่อให้ได้ความละเอียดในการวัดที่ดีที่สุด
	Manual	ย่านการวัดด้วยตัวเอง ผู้ใช้ต้องเลือกตั้งค่า y ย่านการวัดด้วยตนเอง
16	+	ขั้วกรางแท่ง
17		อินพุทมีค่าสูงเกินไฟสำหรับย่านการวัดที่ถูกเลือกใช้
18		แจ้งเตือนสายวัดทดสอบ จะปรากฏขึ้นบนหน้าจอแสดงผลในระยะเวลาสั้น ๆ เมื่อได้กีตามที่หมุนสวิตซ์ฟังก์ชันของมิเตอร์ไปยังหรือจากตำแหน่งใด ๆ

## ขั้วต่อสาย



หมายเลข	คำอธิบาย
①	ขั้วต่ออินพุทสำหรับการวัดกระแสสลับและกระแสแสดงได้ถึง 10 A
②	ขั้วต่อสายร่วม (ขากลับ) สำหรับการวัดห้องหมด
③	ขั้วต่ออินพุทสำหรับวัดค่าแรงดันไฟฟ้า, ความต่อเนื่อง, ความต้านทาน, ประจุไฟฟ้า, ความถี่, อุณหภูมิ, ไมโครแอมป์และไดโอดทดสอบ

## ข้อความเกี่ยวกับข้อผิดพลาด

ข้อความเกี่ยวกับข้อผิดพลาด	
EEET	เปลี่ยนแบตเตอรี่ก่อนนำมีเตอร์ไปใช้งาน
EEL Err	ต้องทำการตรวจสอบมาตรฐาน ทำการตรวจสอบมาตรฐานมีเตอร์ก่อนนำมีเตอร์ไปใช้งาน
EEPr Err	เกิดข้อผิดพลาดภายนอก ต้องทำการซ่อมมีเตอร์ก่อนนำมีเตอร์ไปใช้งาน
F11- Err	เกิดข้อผิดพลาดภายนอก ต้องทำการซ่อมมีเตอร์ก่อนนำมีเตอร์ไปใช้งาน

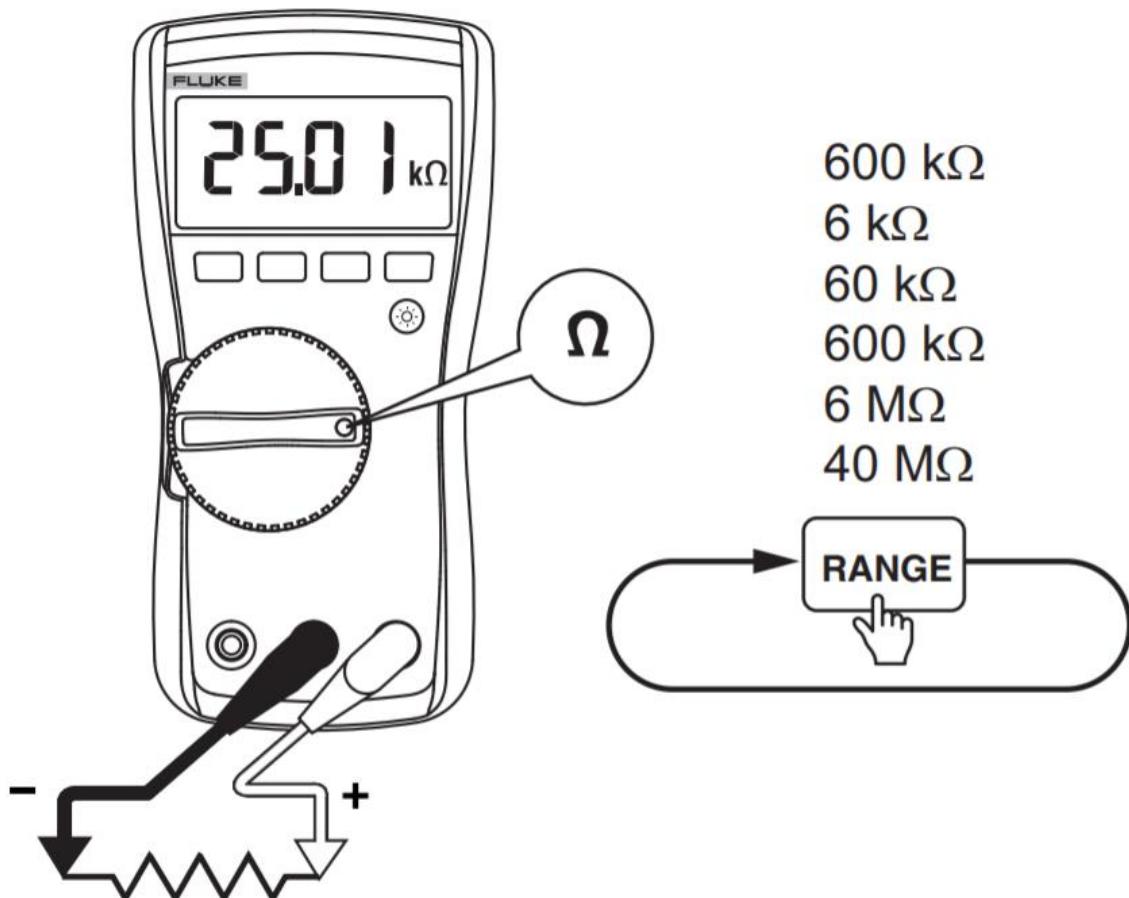
## การวัดพื้นฐาน

ในขณะที่ทำการต่อเข็มสายนำการทดสอบเข้ากับวงจรหรืออุปกรณ์ ให้ต่อสายนำการทดสอบร่วม (COM) ก่อนที่จะต่อเข็มสายนำที่มีกระแสไฟ เมื่อทำการปลดสายนำการทดสอบ ให้ปลดสายนำที่มีกระแสไฟ ก่อนที่จะปลดสายนำ การทดสอบร่วม

คำเตือน !!

ปลดการเชื่อมต่อวงจรพลังงานและคายประจุตัวเก็บประจุแรงดันไฟฟ้าสูงทั้งหมดก่อนที่จะทดสอบความต้านทาน ความต่อเนื่อง ไดโอด และความจุไฟฟ้า ทั้งนี้ เพื่อป้องกันไม่ให้เกิดไฟฟ้าช็อตหรือการบาดเจ็บ หรือป้องกันไม่ให้มิเตอร์เสียหาย

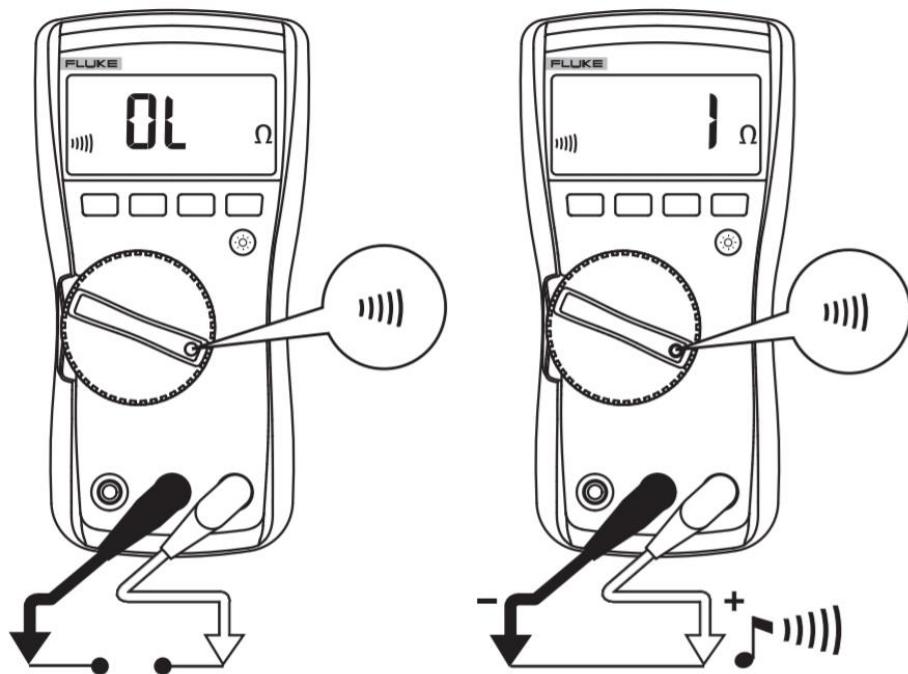
## การวัดค่าความต้านทาน



## การทดสอบความต่อเนื่อง

ເປົ້າ

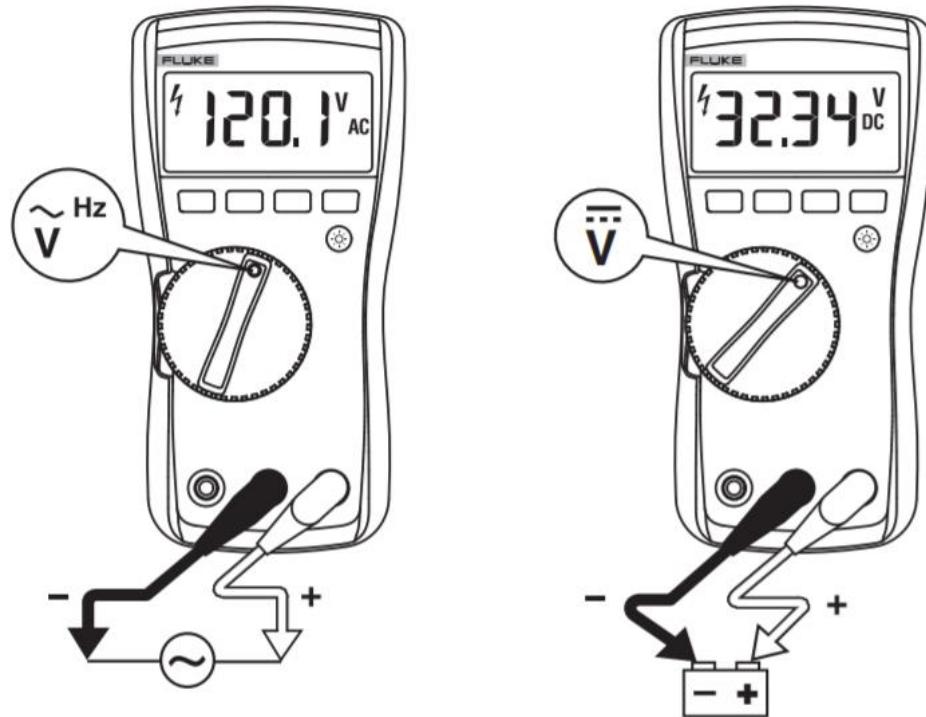
ปี๑



## การวัดแรงดันไฟฟ้ากระแสสลับและกระแสตรง

ໄວລຕົກຮະແສສລັບ

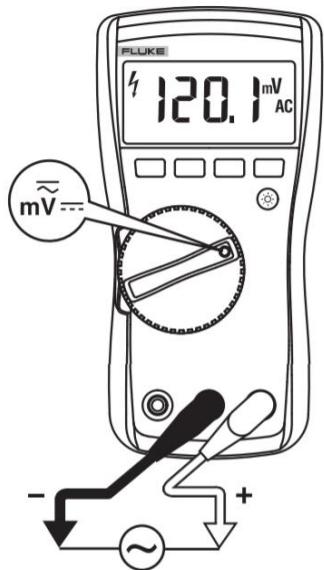
ໄວລຕັກຮະແສດງ



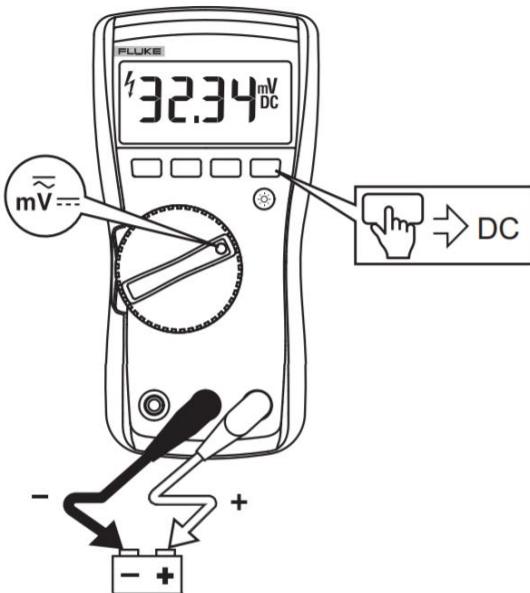
## การวัดมิลลิโวลต์กระแสสลับและกระแสตรง

เมื่อฟังก์ชันอยู่ในตำแหน่ง  $\text{mV} \sim$  มิเตอร์จะวัดมิลลิโวลต์กระแสสลับ รวมทั้งกระแสตรง กด  เพื่อตั้งมิเตอร์ไปยังมิลลิโวลต์กระแสตรง

มิลลิโวลต์กระแสสลับ



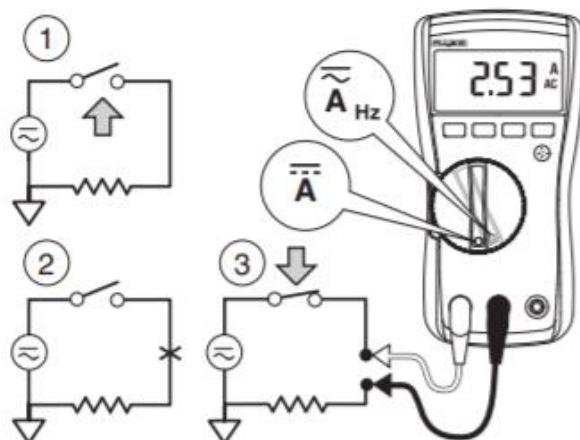
มิลลิโวลต์กระแสตรง



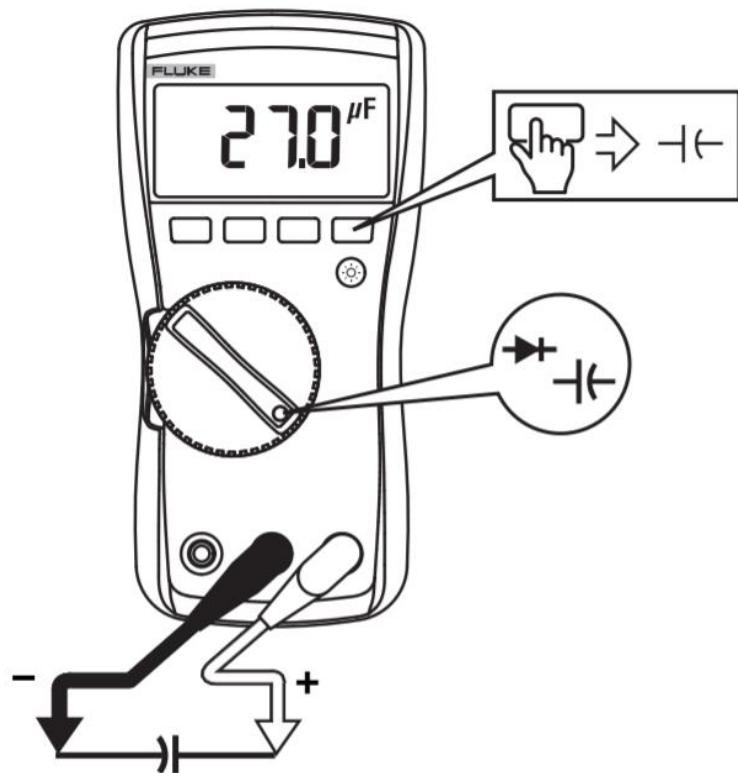
## การวัดกระแสไฟฟ้ากระแสสลับและไฟฟ้ากระแสตรง

ในการวัดกระแสไฟฟ้า

1. ปิดวงจรพลังงาน
2. ตัดวงจร
3. สอดมิเตอร์เป็นอนุกรมกับวงจร แล้วเปิดใช้งานวงจร

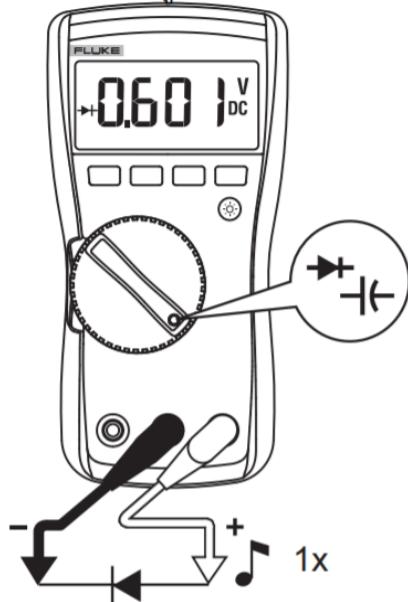


### การวัดความจุไฟฟ้า



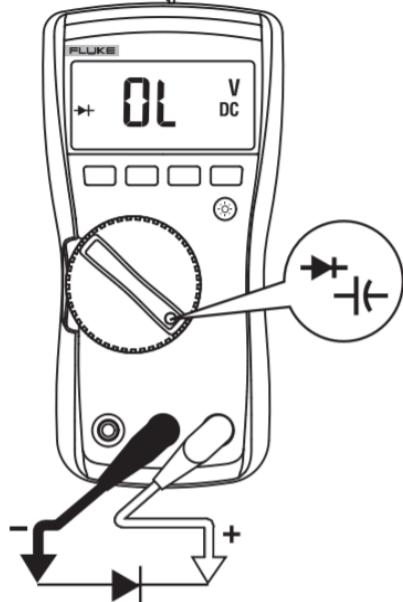
### การทดสอบไดโอด

'ไดโอดที่อยู่ในสภาพดี



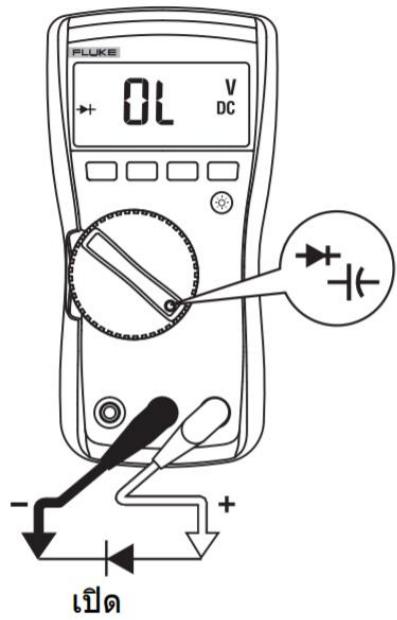
'ใบแอลอสส่งต่อ

'ไดโอดที่อยู่ในสภาพดี



'ใบแอลอสย้อนกลับ

ไดโอดที่อยู่ในสภาพไม่ดี



ไดโอดที่อยู่ในสภาพไม่ดี

