

## **การทดลองที่ 8** วงจรบวกเลขโดยใช้วิธี Schematic บนบอร์ด FPGA

### **วัตถุประสงค์**

1. เพื่อให้นักศึกษาฝึกใช้งานโปรแกรม툴ช่วยการออกแบบวงจรดิจิทัลเป็น
2. เพื่อให้นักศึกษาฝึกการใช้งาน FPGA

### **บทนำ**

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

### **โปรแกรม ISE Xilinx WebPACK**

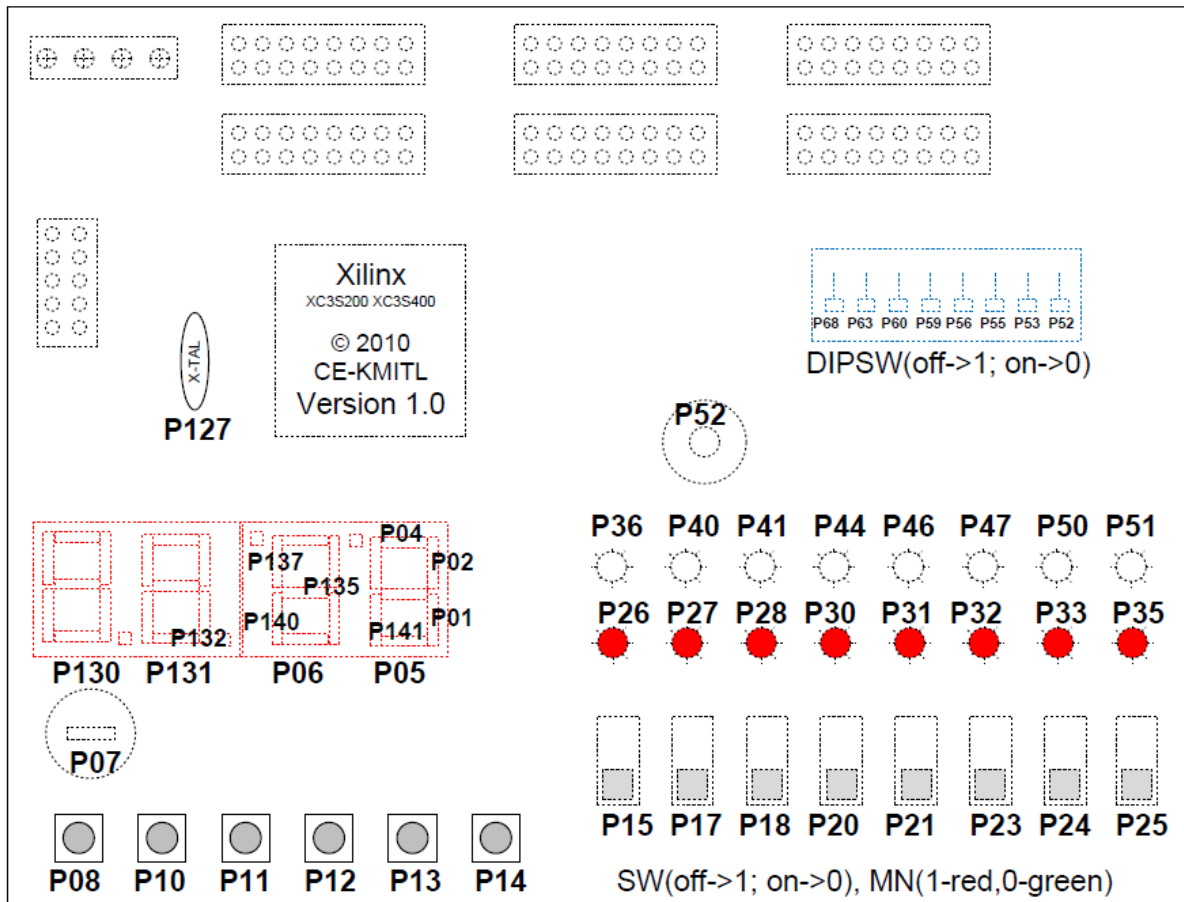
นักศึกษาที่มีหนังสือ “ออกแบบไอซีดิจิทัลด้วย FPGA และ CPLD ภาคปฏิบัติ โดยใช้วิธี Schematic ซอฟต์แวร์ ISE WebPACK”ให้อ่านหนังสือก่อนการทดลองในบทที่ 1 และ 2 ก่อน

### **บอร์ดทดลอง**

วงจรดิจิทัลในปัจจุบันมีความซับซ้อนสูงทำให้การปรับแก้ไขวงจรทำไม่สะดวก การลากสายไฟการติดตั้งไอซีเกดเพิ่มเติมเพื่อปรับเปลี่ยนการทำงานของวงจรทำได้ยาก จึงมีการใช้ไอซีที่สามารถโปรแกรมวงจรได้ ซึ่งมีหลายแบบไม่ว่าจะเป็น Generic Logic Array (GAL), Programmable Array Logic (PAL), Programmable Logic Device (PLD), Complex Programmable Logic Device (CPLD) และ Field Programmable Gate Array (FPGA) ซึ่งล้วนแต่มีลักษณะที่แตกต่างกัน ทั้งขนาดจำนวนเกต ภาวการณ์คงอยู่ของโปรแกรม/ข้อมูล การโปรแกรม โดยการทดลองนี้ใช้งาน FPGA

FPGA มีหลายบริษัทเป็นผู้ผลิต แต่มีสองบริษัทใหญ่คือ Xilinx กับ Altera โดยตระกูลผลิตภัณฑ์ของ Xilinx ที่นิยมกันคือตระกูล Spartan รุ่นเทคโนโลยีขนาด 45nm และตระกูล Virtex รุ่นเทคโนโลยีขนาด 28nm ทางด้าน Altera ตระกูลผลิตภัณฑ์ที่นิยมคือตระกูล Stratix รุ่นเทคโนโลยีขนาด 40nm และ 28nm การทดลองในวิชานี้ใช้ Xilinx Spartan-3 รุ่นเทคโนโลยีขนาด 90nm

ทั้งนี้ Spartan-3 ที่ใช้ในการทดลองคือรุ่น XC3S400 มีความจุขนาด 400,000 เกต จำนวนเกตยิ่งมกยังสามารถสร้างวงจรที่มีขนาดใหญ่และซับซ้อนได้ จุดอ่อนที่ต้องระมัดระวังของ FPGA คือมีอายุการใช้งานจำกัด กล่าวคือสามารถเขียนโปรแกรมลงได้ไม่เกิน 20,000 ครั้ง จากนั้นต้องเปลี่ยนไอซี FPGA ใหม่ การเขียนโปรแกรมลง FPGA ทำได้โดยใช้สายซึ่งใช้มาตรฐานการเชื่อมต่อแบบ Joint Test Action Group (JTAG) และส่งผ่านโปรแกรมด้วย Parallel Port to JTAG (ปัจจุบันนิยมใช้ USB to JTAG มากกว่า)



รูปที่ 1 เลย์เอาต์ของบอร์ด FPGA รุ่น XC3S200 และ XC3S400

บอร์ด FPGA ที่ใช้ในการทดลองดังรูปที่ 1 นั้นหมายเลข Pxx ต่างๆ ระบุถึงขาที่เชื่อมกับตัวชิป FPGA ฉะนั้นจึงไม่ควรสัมผัสตัวไอซี ขา และส่วนประกอบในบอร์ด เนื่องจากไฟฟ้าสถิตจากร่างกายอาจก่อให้เกิดความเสียหายแก่ไอซี อีกทั้งยังทำให้เกิดฮาร์ดแวร์ที่ชำรุดและส่วนประกอบโลหะในวงจร

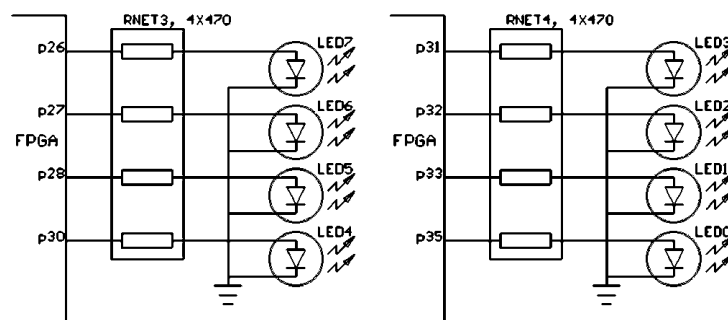
#### วงจรภายในของบอร์ดทดลอง

- **สวิตช์ (SW0 – SW7)** ลักษณะการต่อภายในวงจรเป็นการต่อแบบ R - Pull Up ดังรูปที่ 2 ซึ่งตัวต้านทานต่อเข้า VCC เพื่อรักษาระดับของแรงดันให้คงที่ วงจรจึงอยู่สถานะลอจิก "1" ตลอดเวลา และเมื่อเลื่อนสวิตช์จะให้สถานะลอจิก "0" หรือที่เรียกว่า Active Low เพราะจะทำงาน (ON) เมื่อกระแสไฟฟ้าไหลลงกราวด์ วงจรแบบ Pull Up ได้รับความนิยมมากเพราะกันสัญญาณรบกวนได้ดี ฉะนั้นเมื่อเลื่อน SW = OFF ให้ "1" และ ON ให้ "0"



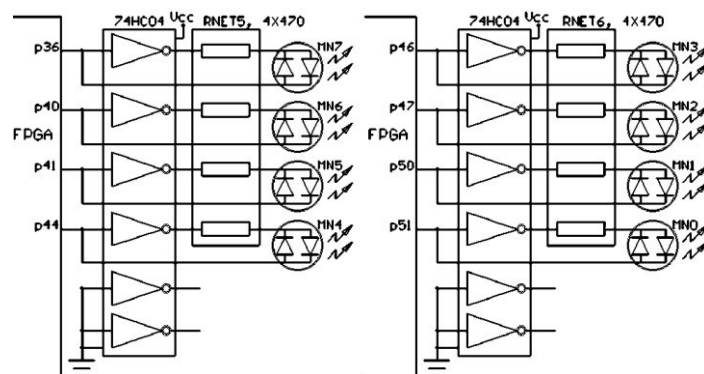
รูปที่ 2 วงจร R-Pull Up (ซ้าย) และวงจร R-Pull Down (ขวา)

- **หลอด LED0 – LED7** ลักษณะวงจรเป็นดังรูปที่ 3 สัญญาณลอจิกจาก FPGA ผ่าน R ความต้านทาน 470Ω เพื่อจำกัดกระแสป้องกัน LED เสียหาย ฉะนั้นเมื่อกำหนดให้ Pxx ดังกล่าวเป็น “1” หลอด LEDx จะสว่างเป็นสีแดง และเมื่อกำหนดให้ Pxx เป็น “0” LED จะไม่สว่าง



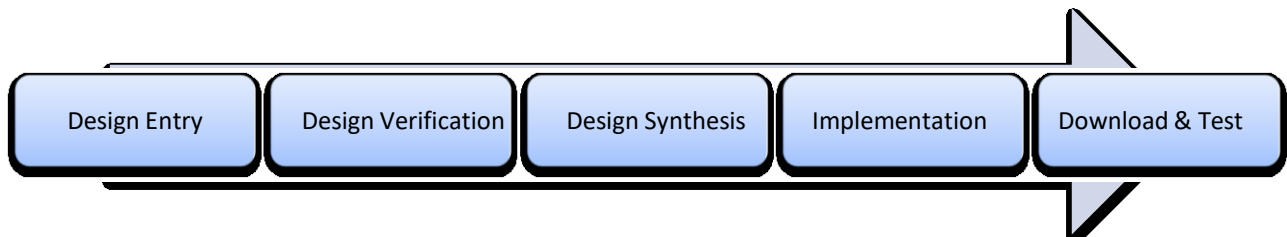
รูปที่ 3 วงจรขับ LED สีแดง

- **หลอด LED สองสี (MN0 – MN7)** ลักษณะวงจรขับแสดงดังรูปที่ 4 ลักษณะวงจรมีอินเวอร์เตอร์ทำหน้าที่กลับขั้วโดยต่ออนุกรมกับ LED กับ R ความต้านทาน 470Ω เพื่อจำกัดกระแสป้องกัน LED เสียหาย หากลอจิกที่ขา P36 เป็น “1” จะได้ผลลัพธ์ไปยัง LED เป็น “0” กระแสย่อมไหลผ่าน LED ดวงซ้ายมือทำให้ติดสว่าง แต่หากขา P36 เป็น “0” กระแสจะมีทิศทางกลับกัน LED ดวงขวามือจึงติดสว่างแทน ฉะนั้นเมื่อกำหนดให้ Pxx ดังกล่าวเป็น “1” หลอด MNx จะสว่างเป็นสีแดง หากเป็น “0” จะเป็นสีเขียว และกำหนดเป็นสัญญาณนาฬิกาจะติดสลับไปมาเป็นสีส้ม



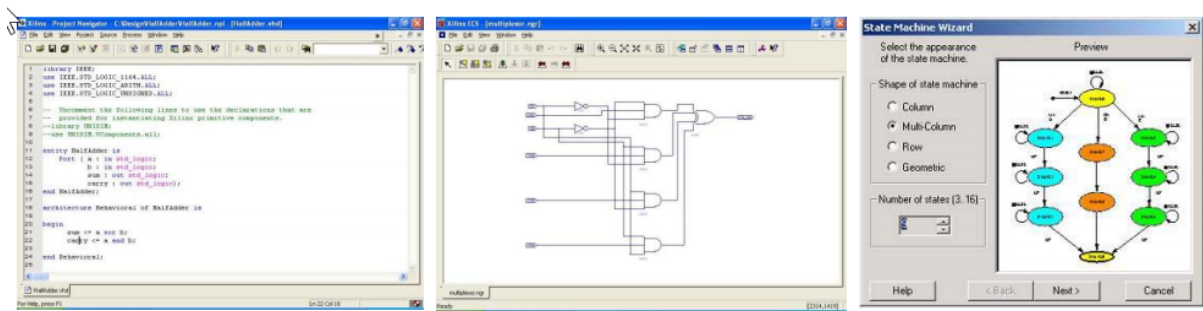
รูปที่ 4 วงจรขับ LED สองสี

ขั้นตอนการออกแบบวงจรอย่างย่อของ version 14.7(ฉบับเต็ม version 8.1i ดูในหนังสือ)  
กระบวนการออกแบบวงจรดิจิทัลด้วย FPGA มีทั้งหมด 5 ขั้นตอนดังนี้



รูปที่ 5 กระบวนการออกแบบ 5 ขั้นตอน

1. **Design Entry** การสร้างวงจรด้วยวิธีการต่างๆ เช่น สร้างด้วยภาษา VHDL หรือ Verilog สร้างด้วยวิธีการออกแบบผังวงจร (Schematic Design) และสร้างด้วยแผนภูมิสถานะ (State diagram) ดังรูปที่ 6 ซึ่งการทดลองครั้งนี้สร้างวงจรด้วยวิธี Schematic Design



รูปที่ 6 ขั้นตอนการสร้างวงจรดิจิทัลทั้ง 3 แบบ

2. **Design Verification** การตรวจสอบวงจรที่ออกแบบเป็นการนำวงจรที่ออกแบบมาตรวจสอบความถูกต้องด้วยโปรแกรมจำลองการทำงาน โดยทั่วไปมีการจำลองการทำงานใน 3 ระดับคือ
  - **Behavioral Simulation** การจำลองเฉพาะพฤติกรรมของวงจรโดยยังไม่คิดถึงโครงสร้างภายใน เพื่อได้แบบจำลองการทำงานเบื้องต้น
  - **Functional Simulation** การจำลองการทำงานของโค้ดในระดับ Register-Transfer-Level (RTL) เพื่อตรวจสอบการทำงานของวงจรถูกนำไป Synthesis
  - **Timing simulation** การจำลองการทำงานที่ใกล้เคียงกับฮาร์ดแวร์มากที่สุดเนื่องจากนำข้อมูลเชิงเวลา (Timing) ที่เกิดขึ้นใน FPGA มาประกอบด้วย มักใช้หาข้อผิดพลาดในกรณีการทำงานจริงบน FPGA ไม่เหมือนผลจาก Functional Simulation
3. **Design Synthesis** การสังเคราะห์วงจร คือ การแปลงโค้ดให้เป็นวงจรในระดับเกต (Gate - level) ซึ่งเป็น Netlist ของเกตต่างๆ ในวงจร ผู้ออกแบบต้องระบุถึงเทคโนโลยี ผู้ผลิต และหมายเลขรุ่นของฮาร์ดแวร์ที่ใช้แก่โปรแกรม Synthesis ซึ่งบางโปรแกรมเปิดโอกาสให้เลือกวิธีการสังเคราะห์ได้ว่าต้องการออปติไมซ์แบบใดระหว่างความเร็วสูงกับประหยัดเนื้อที่
4. **Design Implementation** ขั้นตอนนี้ประกอบด้วย การแบ่งวงจร การวางอุปกรณ์ และการเชื่อมต่อสัญญาณ (Partitioning, Placement and Routing) ตามลำดับ โดยแบ่งวงจรที่ได้จากการสังเคราะห์

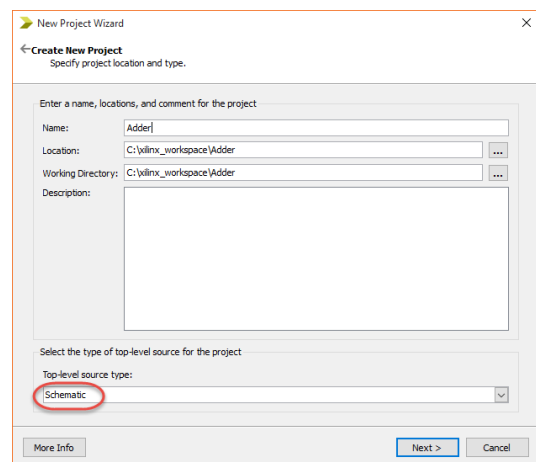
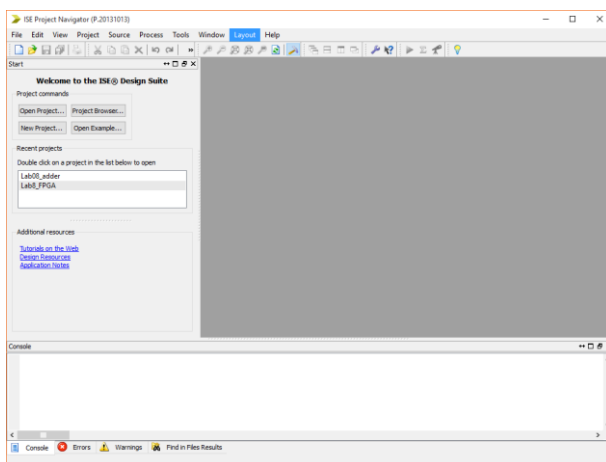
แล้วแยกเป็นวงจรย่อยๆ เพื่อวางลงในโครงสร้างของอุปกรณ์ชนิดต่างๆ ตามชิปที่เลือกเป็นเป้าหมาย จากนั้นจึงวิเคราะห์ความเหมาะสมถึงการวางวงจรย่อยๆ เหล่านั้น ณ ตำแหน่งภายในชิปดังกล่าว ท้ายสุดเป็นขั้นตอนการเชื่อมต่อสัญญาณภายในชิปเข้าด้วยกัน

5. **Download and Test** เป็นการนำข้อมูลวงจรบันทึกลงชิป โดยใช้ไฟล์บิตสตรีม (.JED) ที่ได้จากขั้นตอน Design Implementation ก่อนหน้า

## การใช้งานโปรแกรม

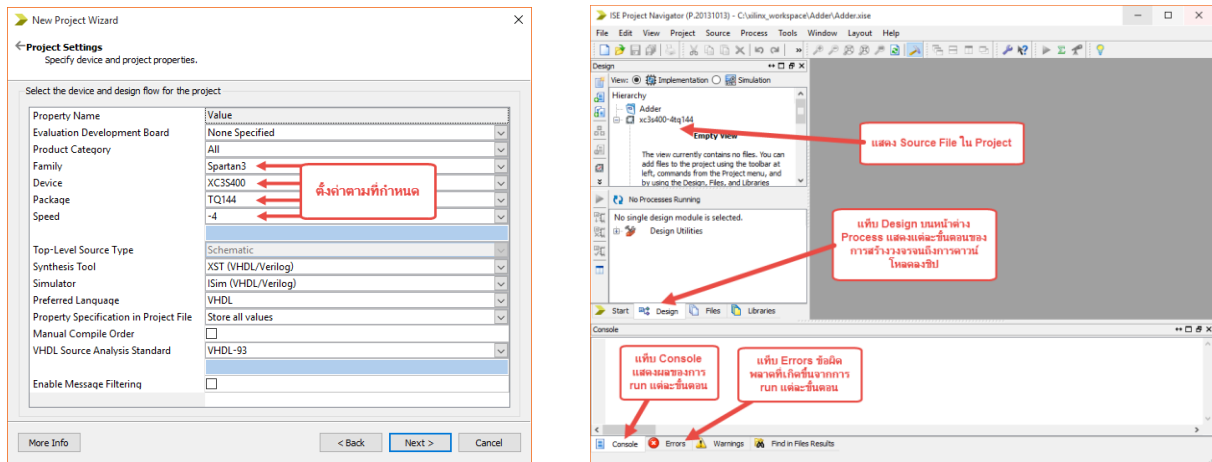
การทดลองนี้ใช้โปรแกรม Xilinx ISE WebPack 14.7 ให้ทดลองตามขั้นตอนดังนี้

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

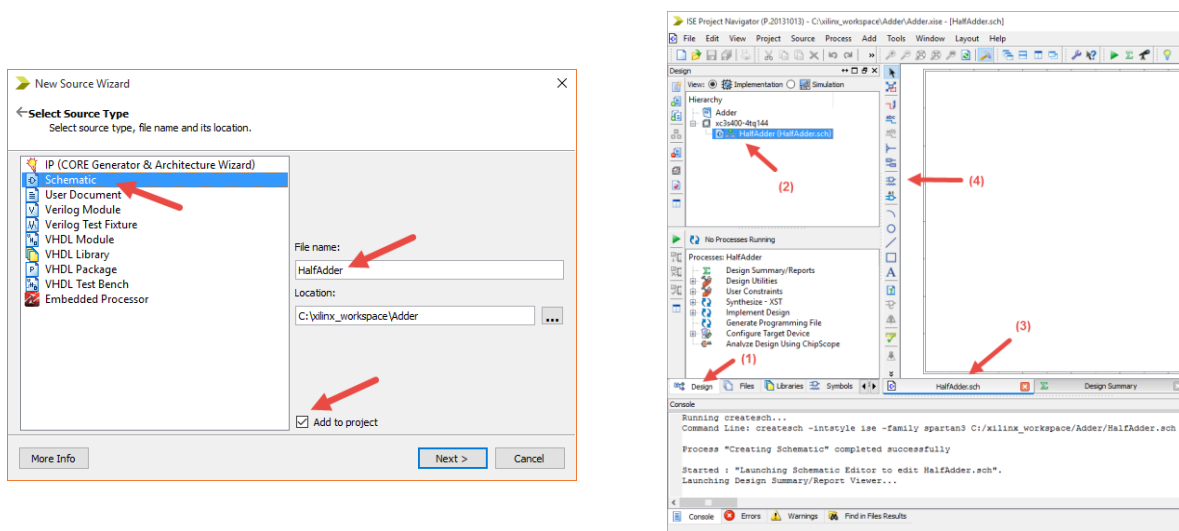


รูปที่ 7 หน้าต่างโปรแกรม Xilinx 8.2 (ซ้าย) และการกำหนดไฟล์ Project (ขวา)

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



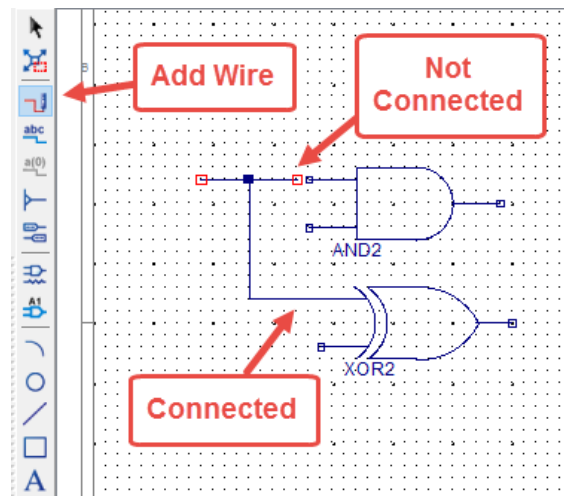
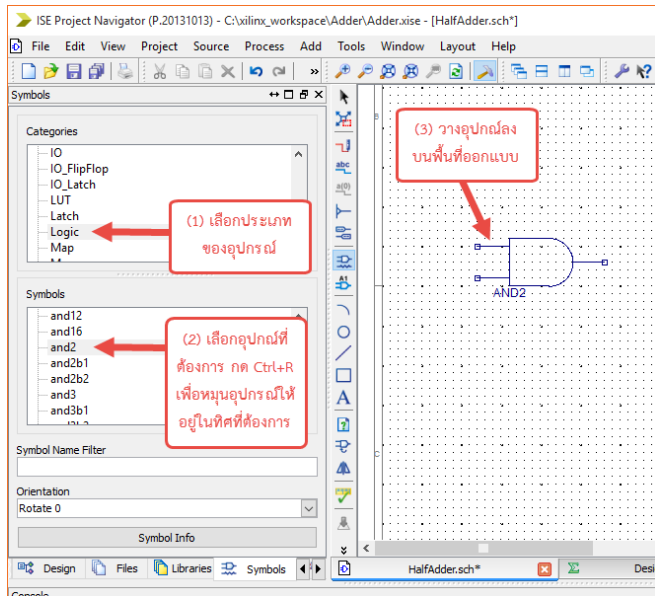
รูปที่ 8 การกำหนดลักษณะของ FPGA ที่ใช้งาน (ซ้าย) ส่วนประกอบหลักของโปรแกรม (ขวา)



รูปที่ 9 สร้าง Schematic (ซ้าย) และส่วนประกอบภายในโปรแกรม (ขวา)

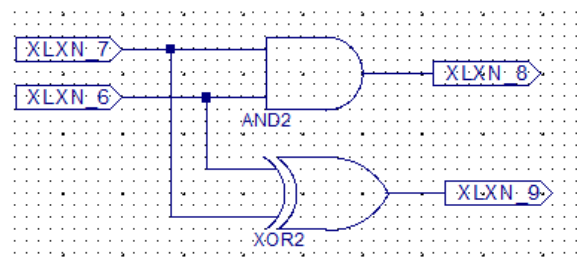
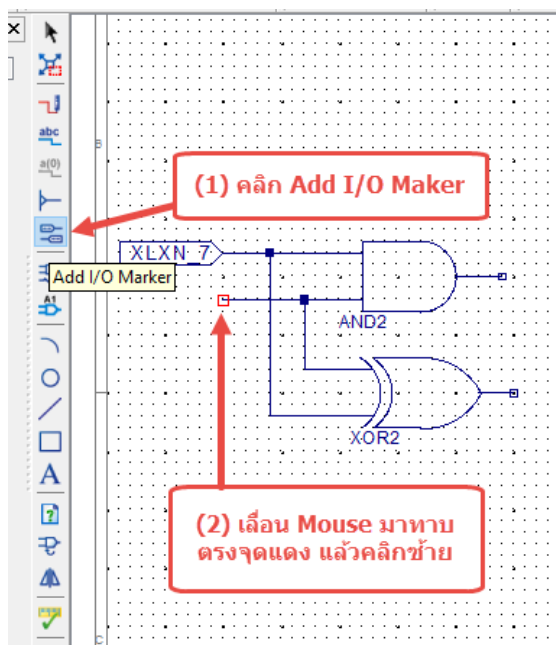
- เลือก Schematic และกำหนดชื่อไฟล์ของวงจรที่จะสร้าง (ต้องขึ้นต้นด้วยตัวอักษร) ดังรูปที่ 9 เลือก Next แล้ว Finish ควรได้ผลลัพธ์ดังรูปที่ 9 (ขวา) โดย (1) เลือกแท็บ Design โดยแสดงเป็น Tree ตัวในสุด (2) แสดงเป็น Source File ที่สร้างขึ้น (3) เลือกแท็บพื้นที่ซึ่งใช้ในการวาดวงจร และ (4) ใช้แถบด้านข้างสำหรับเป็นเครื่องมือวาดวงจร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



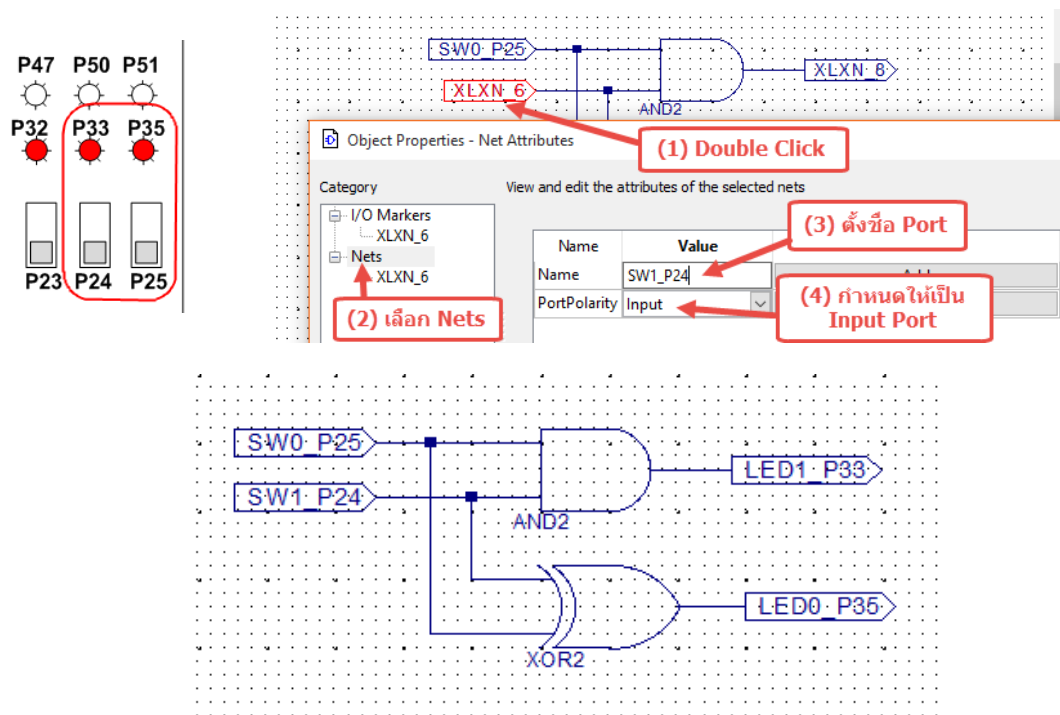
รูปที่ 10 การวางอุปกรณ์ (ซ้าย) และการเชื่อมต่อสาย (ขวา)

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)

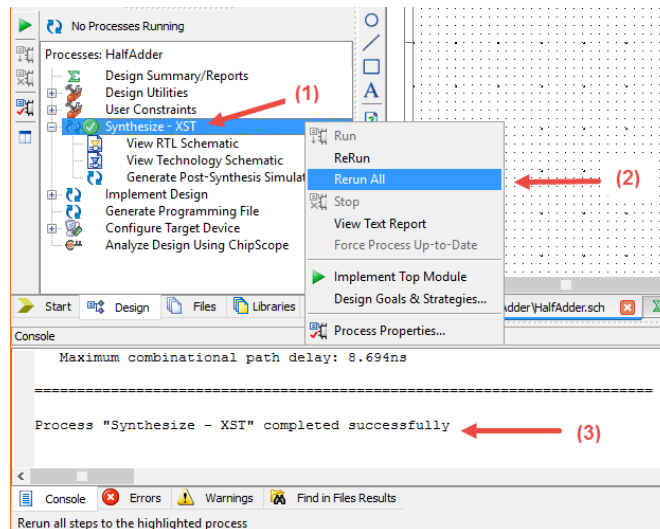
9. เปลี่ยนชื่อตัวรับสัญญาณเข้าและออกใน I/O Marker โดยพิจารณารูปที่ 1 หรือ 12 (บนซ้าย) สำหรับการทดลองนี้ให้เลือกสวิตช์ที่เชื่อมต่อกับขา 24 และ 25 ของ FPGA เป็นตัวป้อนสัญญาณอินพุต และใช้ LED ที่เชื่อมต่อกับขา 33 และ 35 ของ FPGA เป็นตัวแสดงสัญญาณเอาต์พุต โดยกำหนดให้ SW0 และ SW1 เป็นชื่อสัญญาณอินพุต และ LED0 และ LED1 เป็นชื่อสัญญาณเอาต์พุตเพื่อแสดงผลรวมและตัวทดตามลำดับ ทั้งนี้ควรตั้งชื่อในรูปแบบ “ชื่อสัญญาณ\_หมายเลขขา” เช่น SW0\_P25 หมายถึง ใช้สัญญาณ SW0 ต่อกับขา P25 แม้ว่าจะไม่ได้เป็นข้อบังคับ แต่การตั้งชื่อรูปแบบนี้ช่วยให้แก้ไข้ปัญหาได้ง่าย รูปที่ 12 (บนขวา) แสดงหน้าต่างสำหรับการตั้งชื่อ และรูปที่ 12 (ล่าง) แสดงการตั้งชื่อ I/O Marker ที่เสร็จสมบูรณ์



รูปที่ 12 การกำหนดชื่อตัวรับสัญญาณเข้าและออกใน I/O Marker

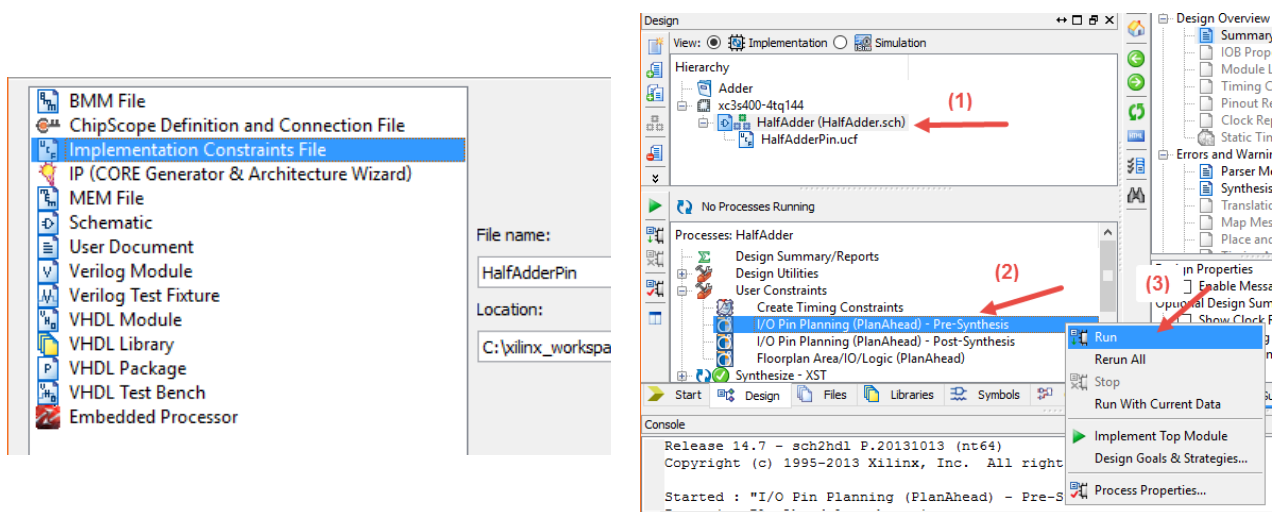
10. ตรวจสอบการทำงานของวงจรที่ออกแบบโดยเลือกแท็บ Design ที่หน้าต่าง Process เลือก Synthesize - XST แล้วดับเบิลคลิกหรือกดคลิกขวาแล้วเลือก Rerun All เพื่อเริ่มตรวจสอบวงจร หากการเดินสายและการวางอุปกรณ์ถูกต้องจะแสดง **Process “Synthesize” completed successfully** ที่หน้าต่าง Console ดังรูปที่ 13 กรณีเกิดข้อผิดพลาดต้องกลับไปตรวจสอบวงจรตามขั้นตอนก่อนหน้า





รูปที่ 13 ขั้นตอนตรวจสอบ Schematic Design

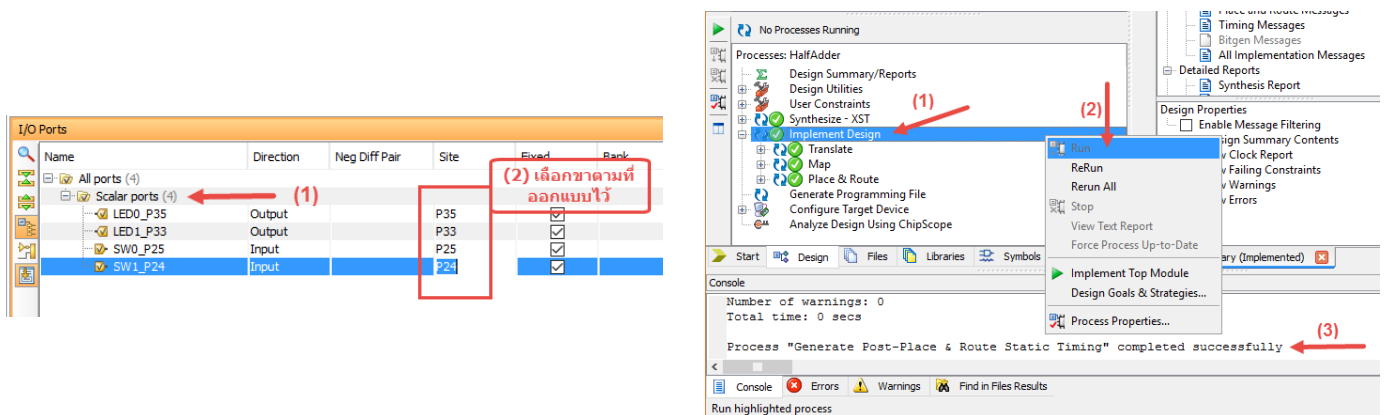
11. จับคู่อุปกรณ์กับหมายเลขพอร์ต โดยเลือก Project -> New Source -> Implementation Constraints File เพื่อจับคู่ระหว่างอุปกรณ์บนบอร์ดกับหมายเลขขาไอซี (Pin Number) ดังรูปที่ 14



รูปที่ 14 การจับคู่อุปกรณ์กับหมายเลขพอร์ต

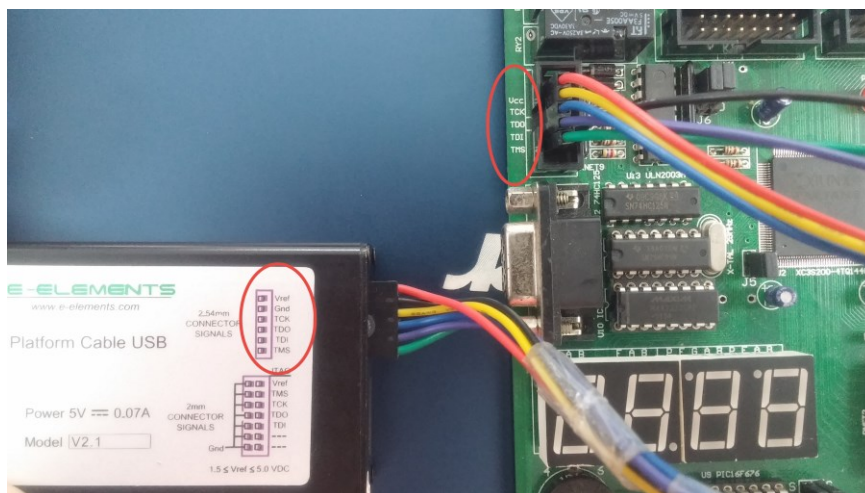
12. เลือกตามขั้นตอนดังรูปที่ 14 (ขวา) เลือกไฟล์ Schematic Source จากนั้นที่หน้าต่าง Process เลือกขั้นตอน User Constraints แล้วคลิกขวาที่ I/O Pin Planning (PlanAhead - Pre-Synthesis) เลือก Run จากนั้นโปรแกรม PlanAhead จะปรากฏขึ้นมา ดังรูปที่ 15

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



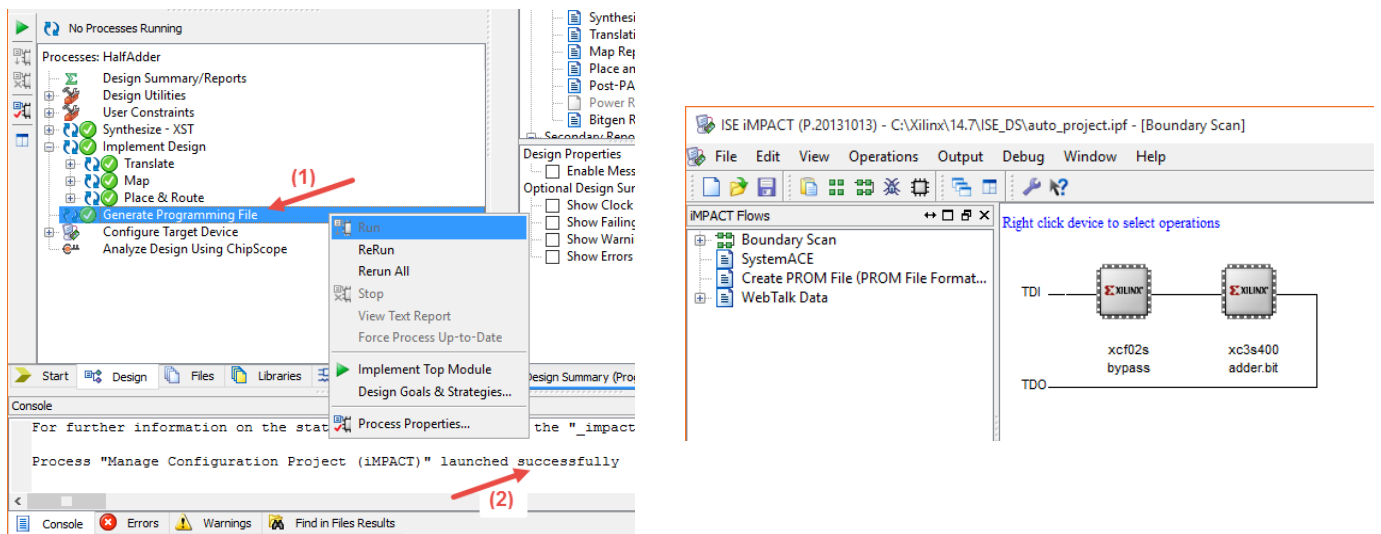
รูปที่ 15 กำหนดเลขขาลงใน Xilinx PlanAhead (ซ้าย) และ Implement Design (ขวา)

14. เริ่มขั้นตอน Implement Design โดยทำตามขั้นตอนในรูปที่ 15 (ขวา) ถ้าไม่มีข้อผิดพลาดจะแสดงเครื่องหมายถูกด้านหน้าหัวข้อ และแสดง **Process "... completed successfully** ที่หน้าต่าง Console กรณีเกิดข้อผิดพลาดต้องกลับไปตรวจสอบการจับคู่อุปกรณ์กับหมายเลขพอร์ตตามขั้นตอนก่อนหน้า
15. เชื่อมต่อ Platform Cable USB เข้ากับเครื่องคอมพิวเตอร์ด้วยสาย USB และเชื่อมต่อ Platform Cable USB เข้ากับบอร์ด FPGA ทางพอร์ต JTAG โดยให้สายสัญญาณทั้งสองฝั่งตรงกัน ดังรูปที่ 16



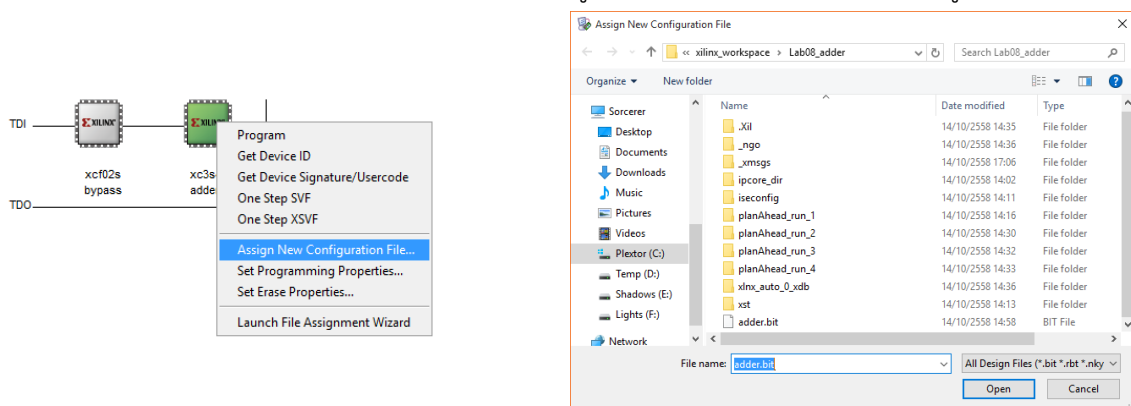
รูปที่ 16 การเชื่อมต่อ Platform Cable USB เข้ากับบอร์ด FPGA ทางพอร์ต JTAG

16. 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 FPGA โดยอัตโนมัติ โดยจะแสดงหน้าต่างถามหาไฟล์ .MCS ให้กด Cancel ข้ามไป และจะถามหาไฟล์ .BIT ก็ให้กด Cancel ข้ามไปเช่นเดียวกัน



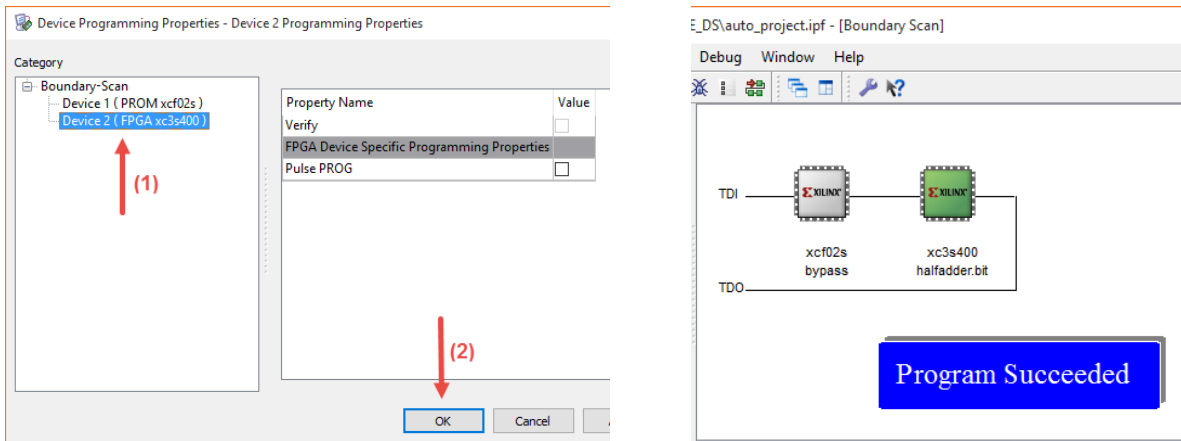
รูปที่ 17 การ Generate Programmable File และโปรแกรม iMPACT

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



รูปที่ 18 การกำหนดไฟล์ .BIT

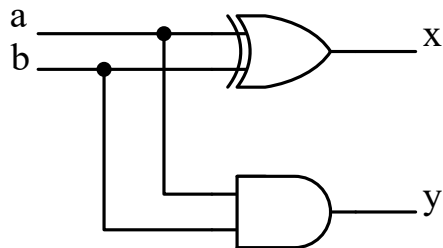
18. เมื่อเลือกไฟล์ .BIT ไว้แล้ว ขั้นสุดท้ายคือบันทึกลงในชิป FPGA ให้คลิกขวาบนรูปชิป XC3S400 สั่ง Program ดังรูปที่ 19 (ซ้าย) หากประสบความสำเร็จจะขึ้นข้อความ **Program Succeeded** ดังรูปที่ 19 (ขวา)



รูปที่ 19 การบันทึกลงในชิป FPGA

### การทดลอง

1. ให้นักศึกษาสร้างวงจร 1-Bit Half Adder ดังรูปข้างล่างนี้ จนเสร็จสมบูรณ์ และทำการจำลองการทำงาน แล้วบันทึกผลการทดลองลงตาราง



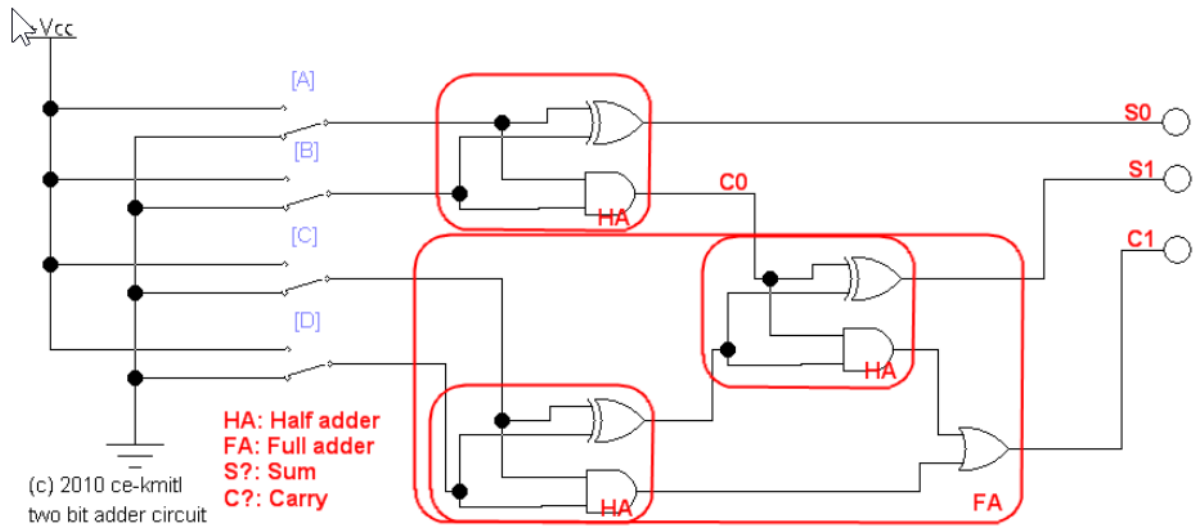
บันทึกผลการทดลอง (สว่าง=1 ดับ=0)

a	b	x	y
0	0		
0	1		
1	0		
1	1		

วิเคราะห์ผลลัพธ์ที่ได้จากตาราง

.....  
.....  
.....  
.....  
.....

2. ให้นักศึกษาร่างวงจรบวกเลขสองบิตดังรูปข้างล่างนี้ โดยเทียบเคียงจากขั้นตอนการใช้งานโปรแกรมตามข้อ 1 แล้วโปรแกรมลงบอร์ด FPGA โดยให้กำหนดอินพุตมาจากสวิทช์เลื่อน และเอาต์พุตแสดงที่ LED บันทึกหมายเลข PIN และบันทึกผลการทดลองลงตาราง



A หมายเลข PIN คือ P.....

S0 หมายเลข PIN คือ P.....

B หมายเลข PIN คือ P.....

S1 หมายเลข PIN คือ P.....

C หมายเลข PIN คือ P.....

C1 หมายเลข PIN คือ P.....

D หมายเลข PIN คือ P.....

บันทึกผลการทดลอง (สว่าง=1 ดับ=0)

SW0 (A)	SW1 (B)	SW2 (C)	SW3 (D)	LED0 (C1)	LED1 (S1)	LED2 (S0)
0	0	0	0			
0	1	1	0			
0	1	1	1			
1	0	1	1			
1	1	0	1			
1	1	1	0			
1	1	1	1			

## ใบตรวจการทดลองที่ 8

วัน/เดือน/ปี \_\_\_\_\_ ☐ กลุ่มเช้า ☐ กลุ่มบ่าย กลุ่มที่ \_\_\_\_\_

1. รหัสนักศึกษา \_\_\_\_\_ ชื่อ-นามสกุล \_\_\_\_\_

2. รหัสนักศึกษา \_\_\_\_\_ ชื่อ-นามสกุล \_\_\_\_\_

การตรวจการทดลอง

☐ บันทึกคะแนนแล้ว

การทดลองข้อ 1      ลายเซ็นอาจารย์ \_\_\_\_\_

การทดลองข้อ 2      ลายเซ็นอาจารย์ \_\_\_\_\_

คำถามท้ายการทดลอง (5 คะแนน) \_\_\_\_\_

1. Symbols ที่ใช้ได้กับชิป FPGA ที่เลือกใช้มีหลายตัว จงแนะนำสัญลักษณ์มาหนึ่งตัวในแต่ละ Categories รวมไม่น้อยกว่า 5 ตัว (รายละเอียดเพิ่มเติมหาได้ที่ Symbol Info)

