

การทดลองที่ 0 การใช้งานบอร์ด FPGA และ โปรแกรม ISE Xilinx WebPACK

วัตถุประสงค์

1. เพื่อให้นักศึกษาติดตั้งและฝึกการใช้งาน FPGA
2. เพื่อให้นักศึกษาฝึกการใช้งานโปรแกรม ISE Xilinx WebPACK

**** หมายเหตุ ให้อ่านเอกสารการทดลองและเอกสารประกอบให้ครบก่อนเริ่มทำการทดลอง ****

การใช้งานบอร์ด FPGA

การทดลองนี้ นักศึกษาจะได้รู้จักการออกแบบวงจรด้วยการวาดรูป (Schematic Circuit) และบันทึก

โปรแกรม ลงในชิป Field Programmable Gate Array (FPGA)

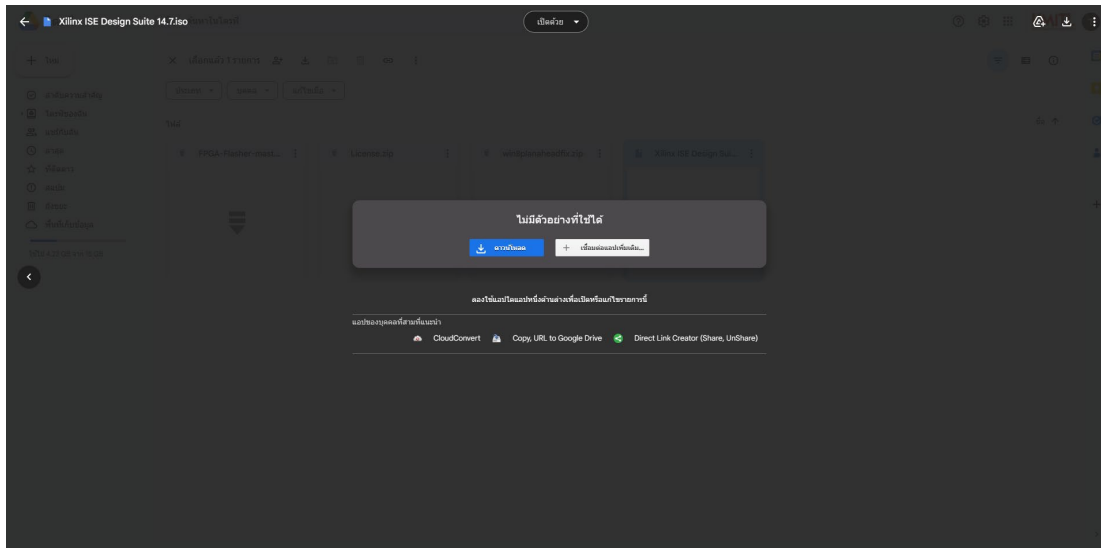
โปรแกรม ISE Xilinx WebPACK

นักศึกษาที่มีหนังสือ “ออกแบบไอซีดิจิทัลด้วย FPGA และ CPLD ภาคปฏิบัติ โดยใช้วิธี Schematic

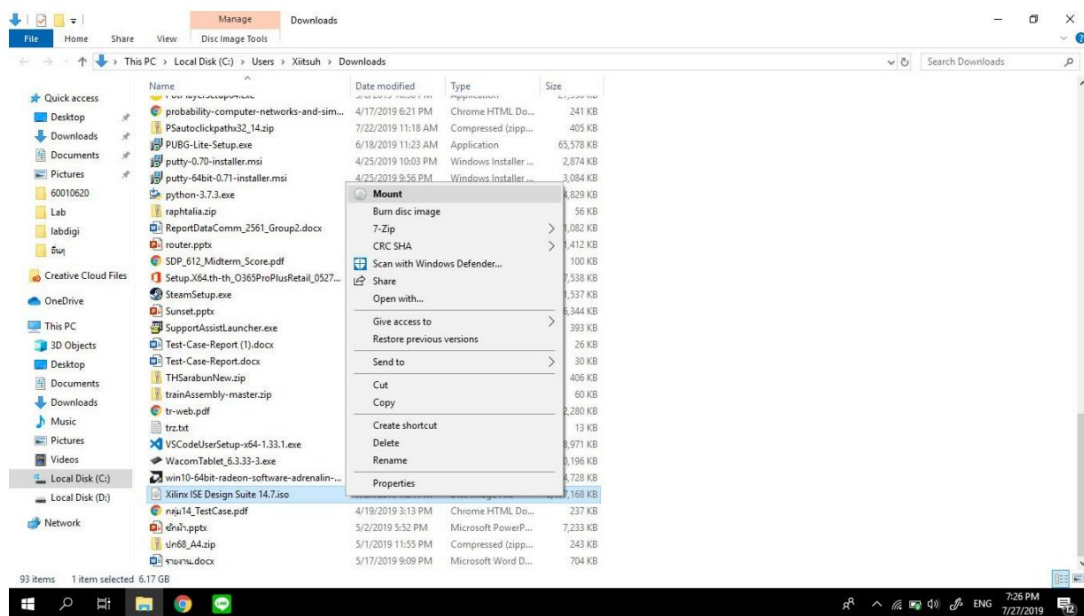
ซอฟต์แวร์ ISE WebPACK” ให้อ่านหนังสือก่อนการทดลองในบทที่ 1 และ 2 ก่อน

การติดตั้งโปรแกรม ISE Xilinx WebPACK

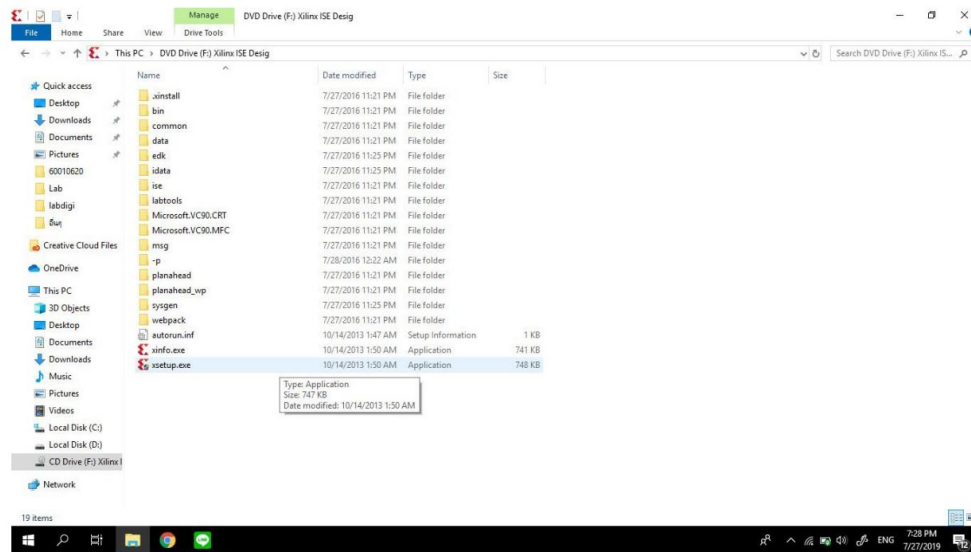
1. download โปรแกรมจาก <https://bit.ly/3Kh6Qhw>



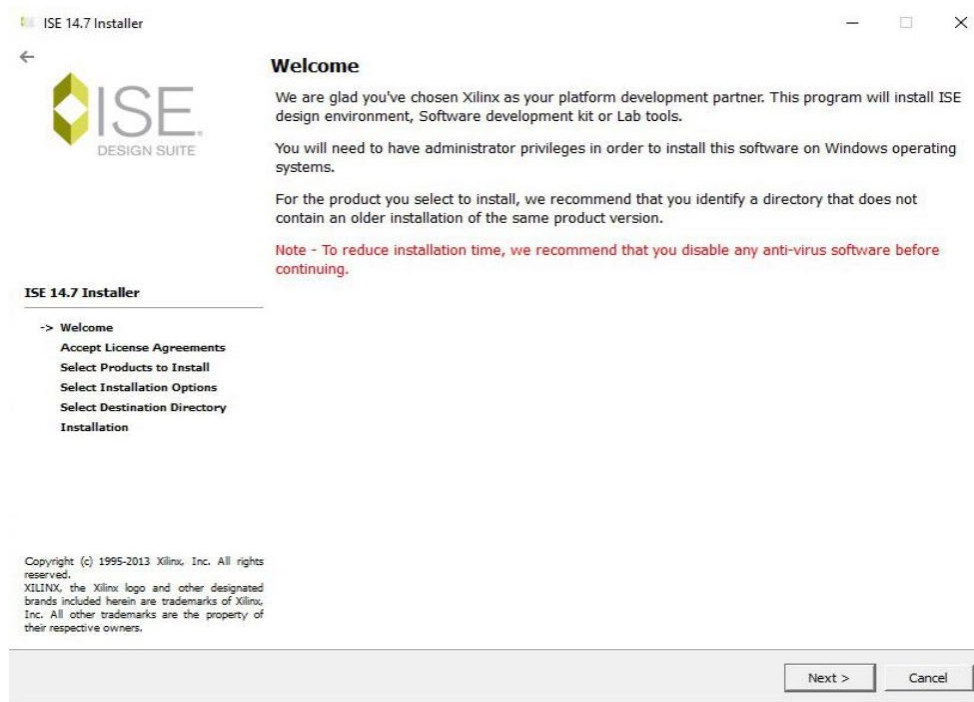
2. คลิกขวาที่ไฟล์ที่ download จากนั้นกดคำว่า Mount



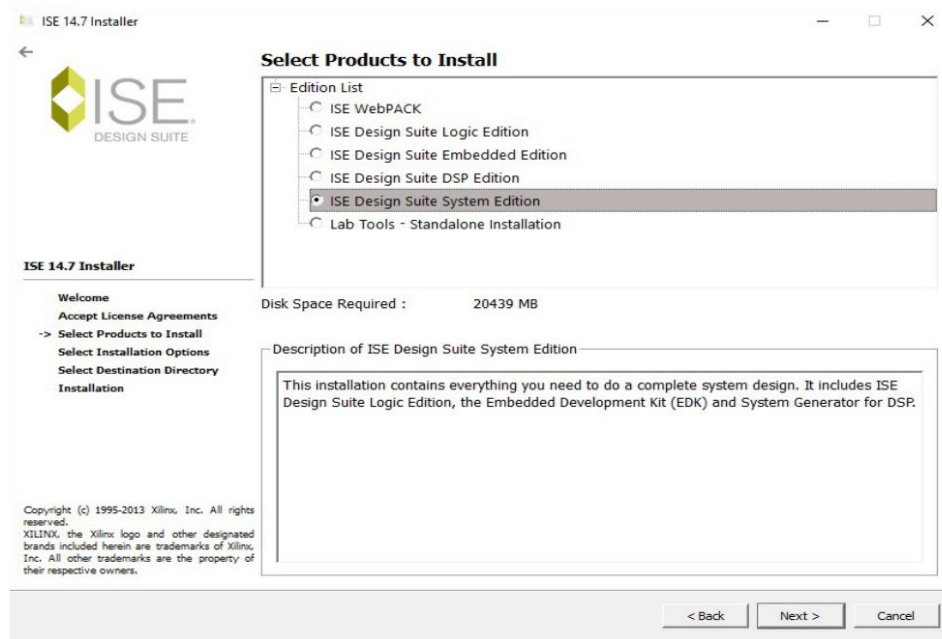
3. เมื่อคลิกMountเสร็จจะขึ้นหน้านี้มาให้กดที่ xsetup.exe



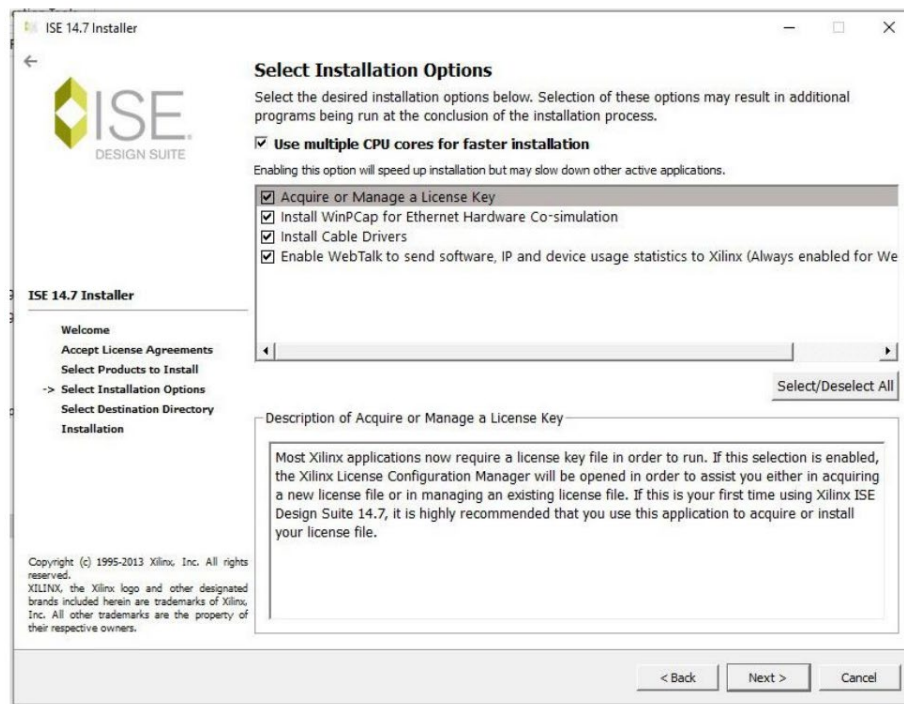
4. ให้กด Next ไปเรื่อย ๆ



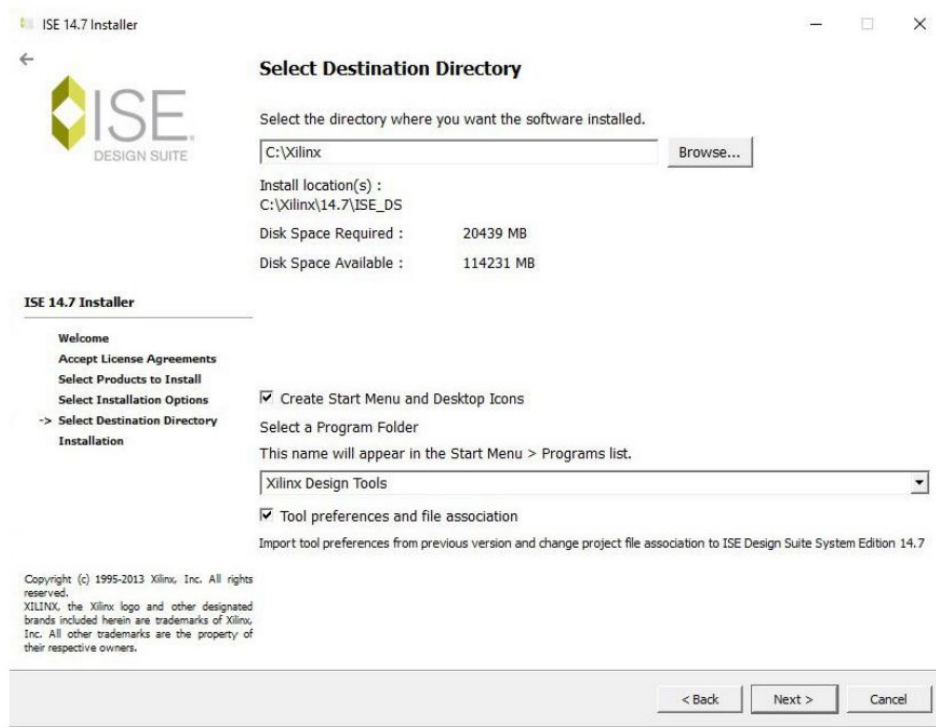
5. เมื่อถึงหน้านี้ให้เลือก ISE Design Suite System Edition จากนั้นกด Next



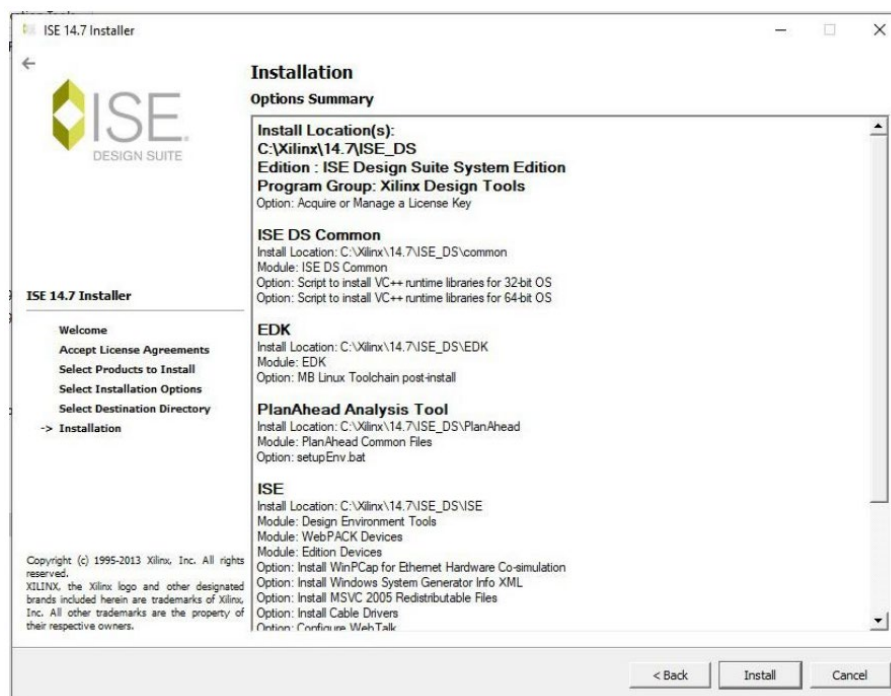
6. เมื่อถึงหน้านี้ให้กดเลือกทั้งหมด จากนั้นกด Next



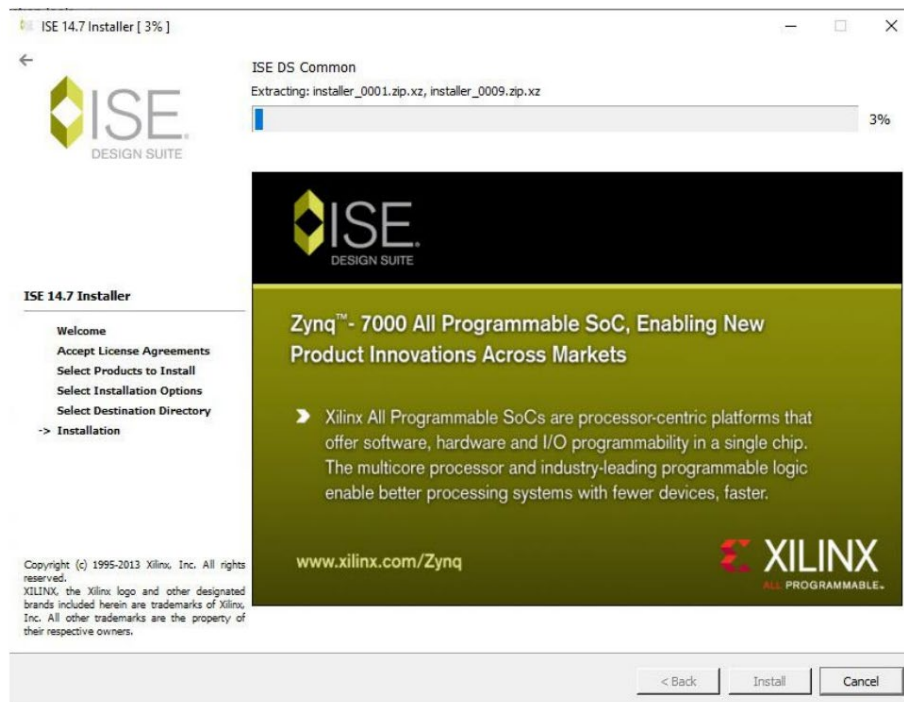
7. เลือกโฟลเดอร์ที่จะติดตั้ง จากนั้นกด Next



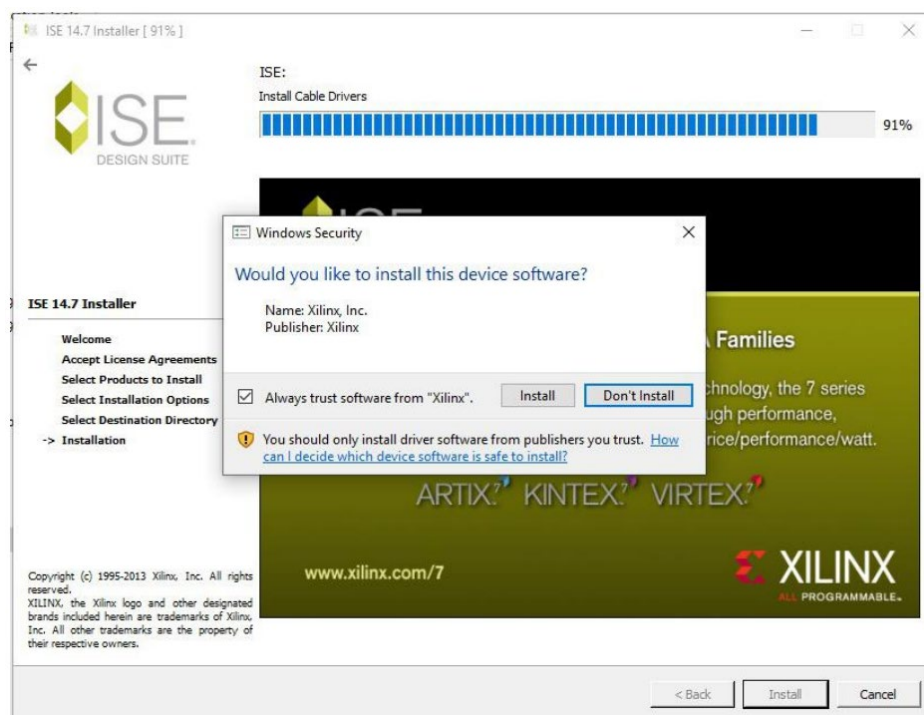
8. เมื่อถึงหน้าให้กด Install



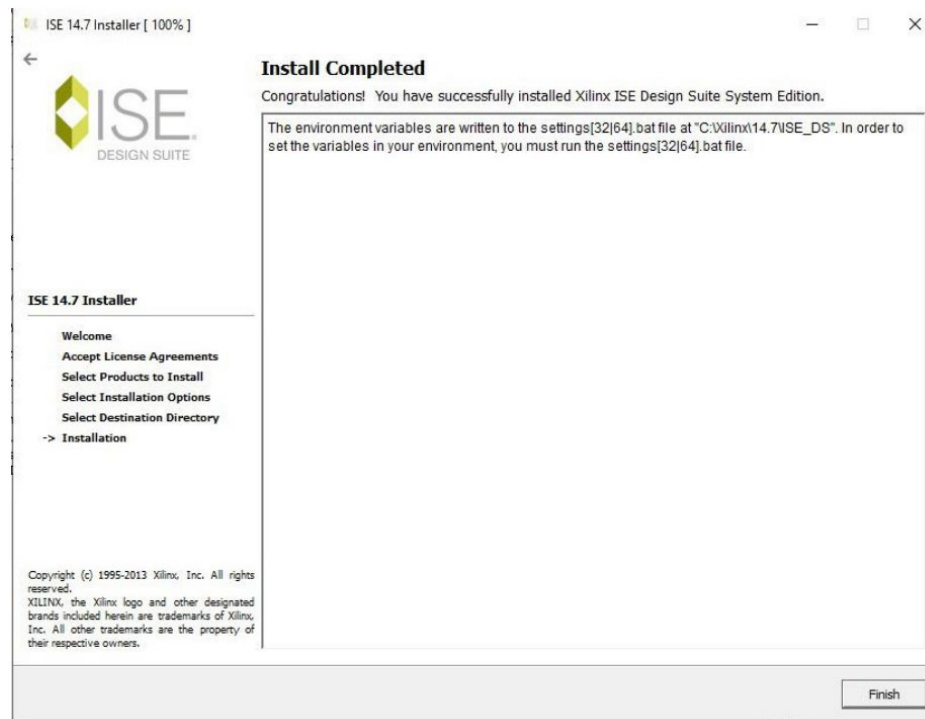
9. เมื่อกด Install แล้วหน้าจอจะแสดงผลดังภาพ ให้รอนกว่าจะครบ 100%



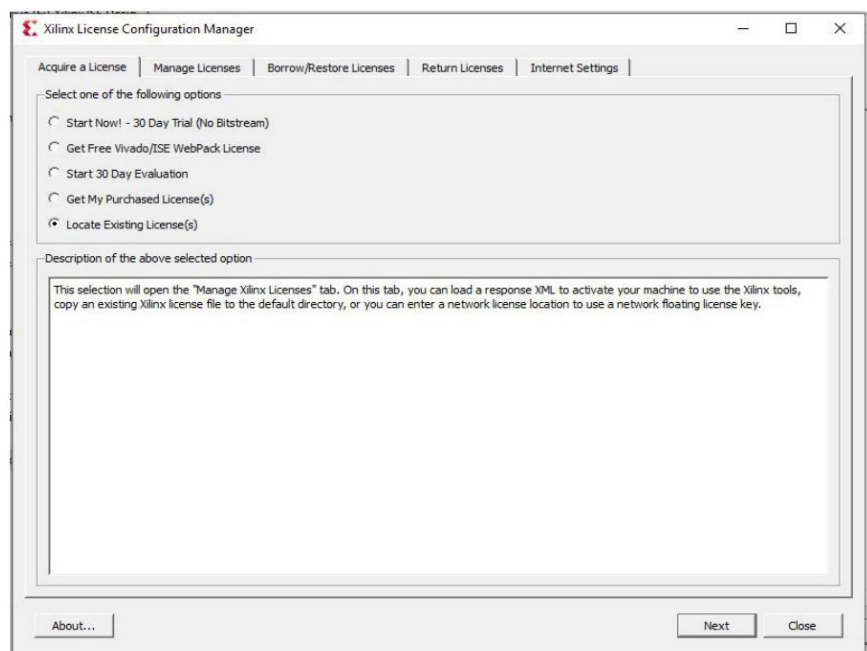
10. หากมีแจ้งเตือนดังรูปให้กด Install



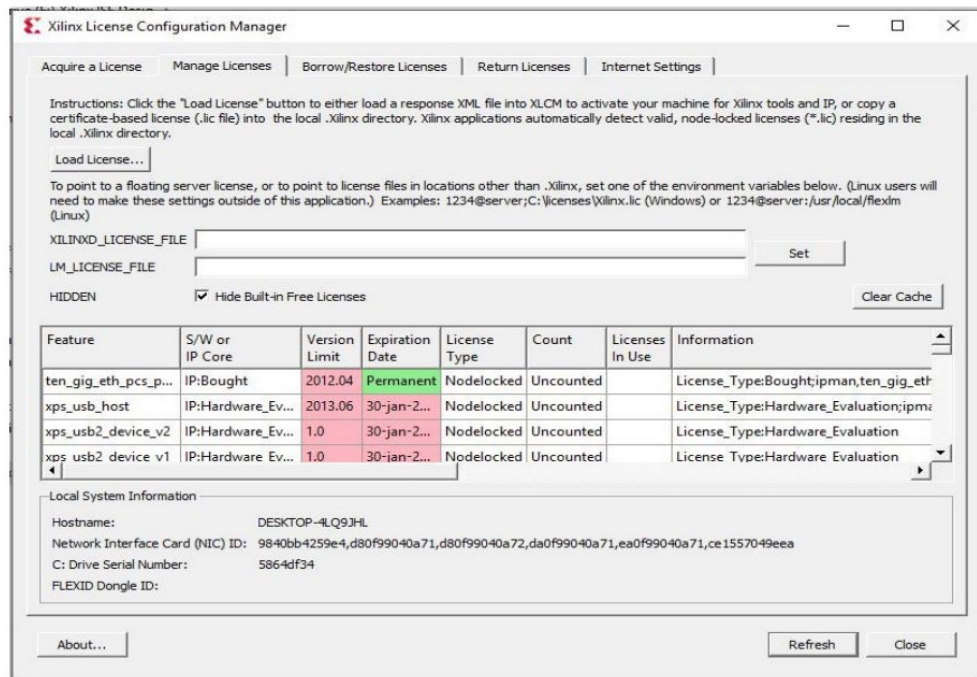
11. เมื่อเสร็จสิ้นจะขึ้นดังรูป



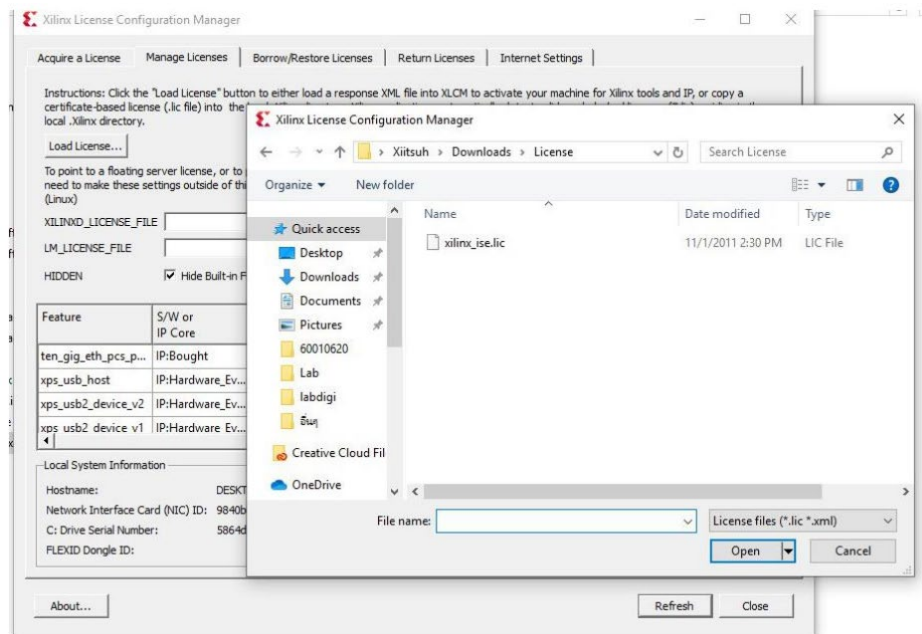
12. เมื่อติดตั้งเสร็จจะแสดงหน้าจอขึ้นมาจากนั้นให้กดเลือก Locate Existing License(s) จากนั้นกด Next



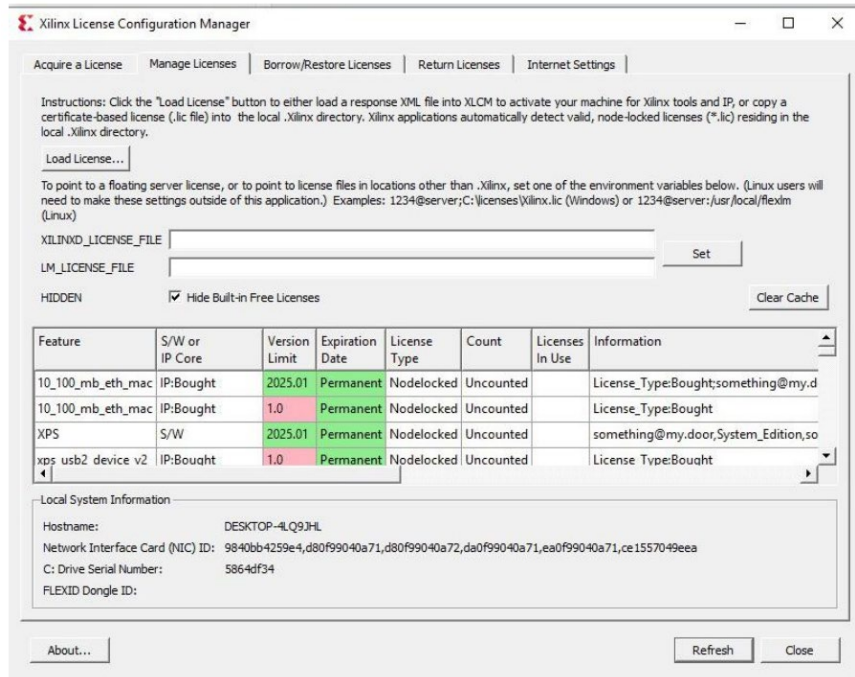
13. ให้ Download License จาก <https://bit.ly/44JUTt2> จากนั้นคลิกที่ Load License



14. จากนั้นเลือกไฟล์ License ที่ Download มา



15. เมื่อเสร็จสิ้นจะขึ้นดังรูป



ปัญหาที่อาจ(ต้อง)พบ

1. เมื่อติดตั้งเสร็จ ไม่แสดงหน้าต่างสำหรับลง license หรือ ไม่สามารถเปิด program impact หรือ create svf ไฟล์ได้ เนื่องจากมีปัญหาที่โปรแกรม Impact , PlanAhead สำหรับระบบปฏิบัติการ Windows 10 โดยสามารถ แก้ไขได้ตามขั้นตอนดังนี้

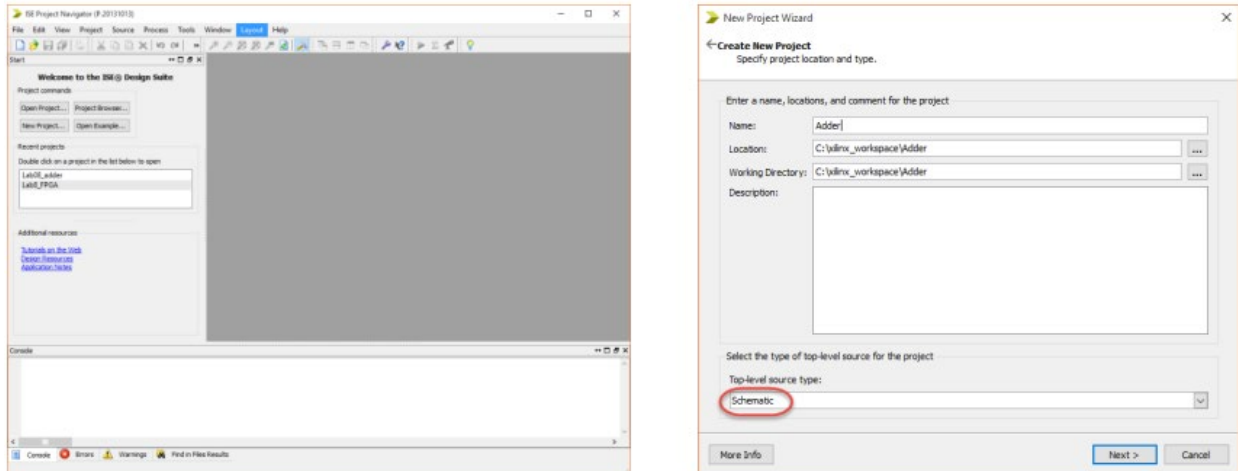
1. เปิด ไดเรกทอรี C:\Xilinx\14.7\ISE_DS\ISE\lib\nt64
2. หาไฟล์ที่มีชื่อ libPortability.dll แล้วเปลี่ยนชื่อให้เป็น libPortability.dll.orig
3. ทำการ copy ไฟล์ libPortabilityNOSH.dll แล้วกดวางที่ไดเรกทอรีนี้ เสร็จแล้วให้ทำการเปลี่ยนชื่อเป็น libPortability.dll
4. Copy libPortabilityNOSH.dll อีกครั้ง, เปลี่ยนไดเรกทอรีเป็น C:\Xilinx\14.7\ISE_DS\common\lib\nt64 และวางไฟล์ที่ไดเรกทอรีนี้
5. ในไดเรกทอรีนี้ C:\Xilinx\14.7\ISE_DS\common\lib\nt64 หาไฟล์ที่มีชื่อ libPortability.dll แล้วเปลี่ยน ชื่อให้เป็น libPortability.dll.orig

6. เปลี่ยนชื่อ libPortabilityNOSH.dll เป็น libPortability.dll
7. เปิดไดเรกทอรี C:\Xilinx\14.7\ISE_DS\PlanAhead\bin และเปลี่ยนชื่อ rdiArgs.bat เป็น rdiArgs.bat.orig
8. Download ไฟล์ rdiArgs.bat จากลิงค์ <https://bit.ly/3OyrhZS>
9. ทำการแตกไฟล์ จากนั้น copy ไฟล์ rdiArgs.bat แล้วทำการวางในไดเรกทอรี C:\Xilinx\14.7\ISE_DS\PlanAhead\bin
10. ศึกษาเพิ่มเติมได้ที่ <https://bit.ly/3QcKTnJ>

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 ไปเรื่อย ๆ 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

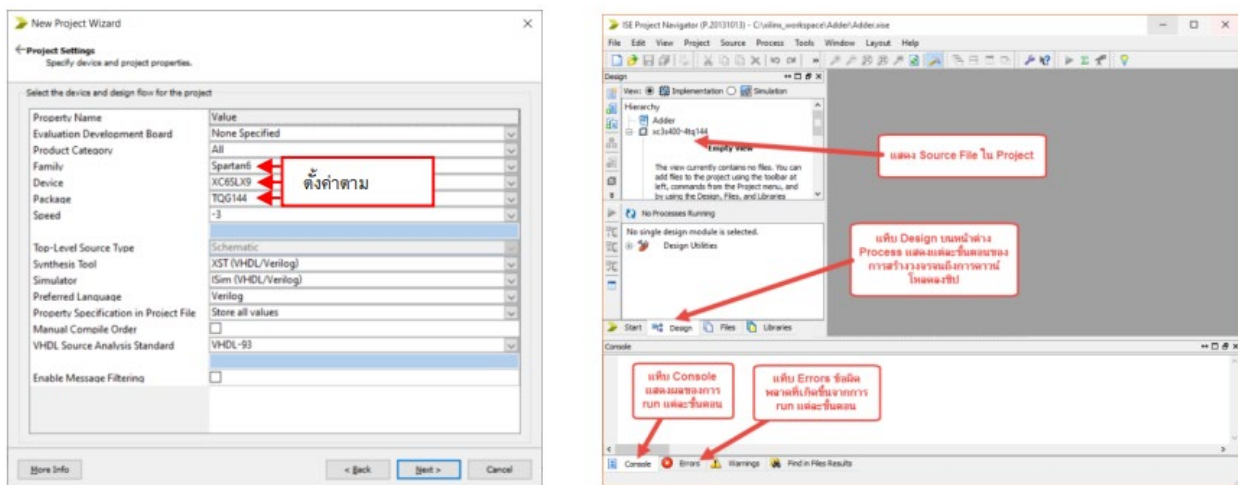


รูปที่ 7 หน้าต่างโปรแกรม Xilinx 8.2 (ซ้าย) และการกำหนดไฟล์ Project (ขวา)

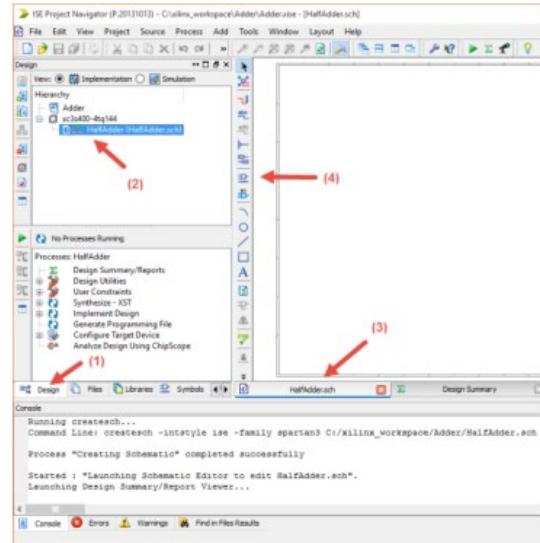
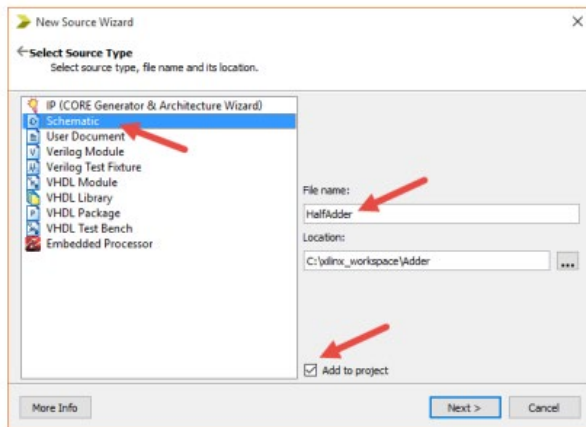
อย่าลืม! เซฟ (Ctrl+S) บ่อยๆ

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)

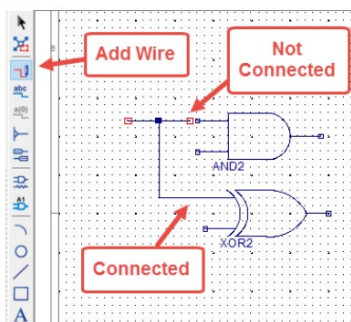
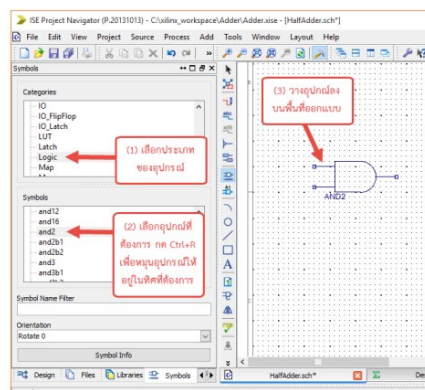


รูปที่ 9 สร้าง Schematic (ซ้าย) และส่วนประกอบภายในโปรแกรม (ขวา)

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) เลือกแท็บพื้นที่ซึ่งใช้ในการวาดวงจร และ (4) ใช้แถบด้านข้างสำหรับเป็นเครื่องมือวาดวงจร

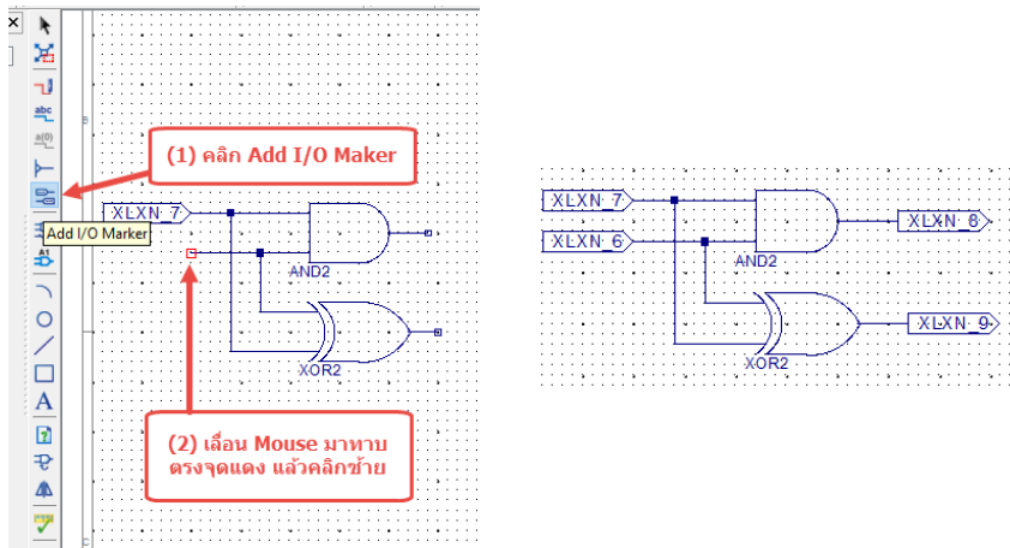
6. เริ่มวาดวงจรแรกเป็น 1-Bit Half Adder อุปกรณ์ประกอบคือเกต XOR กับ AND โดยหาอุปกรณ์ในแท็บ Symbols ตามชื่อของอุปกรณ์ที่ต้องการ ดังรูปที่ 10 (ซ้าย) วางอุปกรณ์ทั้งสองบนพื้นที่ออกแบบ

7. จากนั้นลากเส้นโดยเลือก Add Wire (ระวัง! ห้ามใช้ Add Line) เชื่อมต่อสาย ลักษณะการเชื่อมต่อที่สมบูรณ์ (Connected) และไม่สมบูรณ์ (Not Connected) แสดงดังรูปที่ 10 (ขวา)



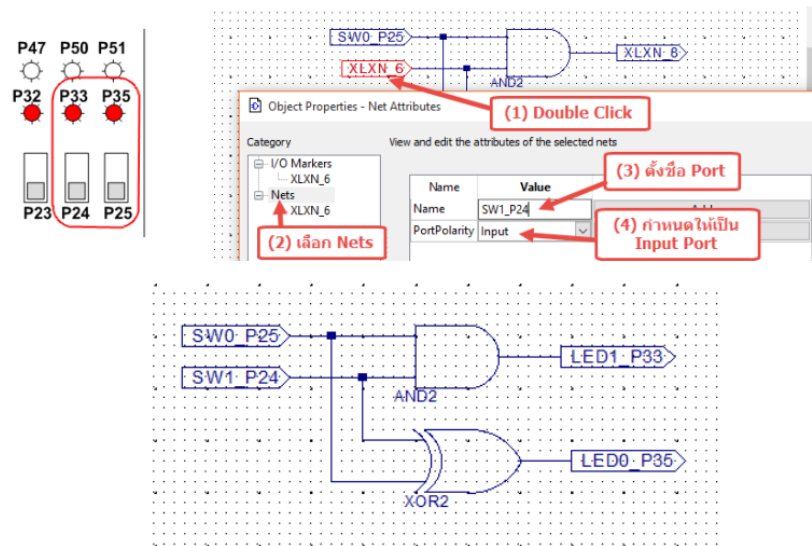
รูปที่ 10 การวางอุปกรณ์ (ซ้าย) และการเชื่อมต่อสาย (ขวา)

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



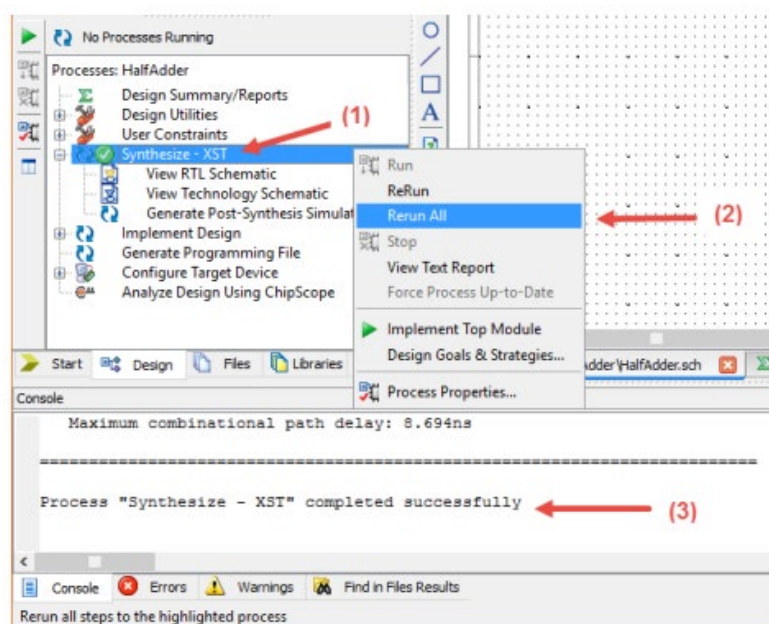
รูปที่ 11 สร้างจุดเชื่อมต่อ (ซ้าย) และวงจรเมื่อวาดเสร็จแล้ว (ขวา)

9. เปลี่ยนชื่อตัวรับสัญญาณเข้าและออกใน I/O Marker โดยพิจารณารูปที่ 1 หรือ 12 (บนซ้าย) สำหรับการทดลองนี้ ให้เลือกสวิตช์ที่เชื่อมต่อกับขา 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 แต่การตั้งชื่อรูปแบบนี้ช่วยให้แก้ไขปัญหาง่ายๆ รูปที่ 12 (บนขวา) แสดงหน้าต่างสำหรับการตั้งชื่อ และรูปที่ 12 (ล่าง) แสดงการตั้งชื่อ I/O Marker ที่เสร็จสมบูรณ์



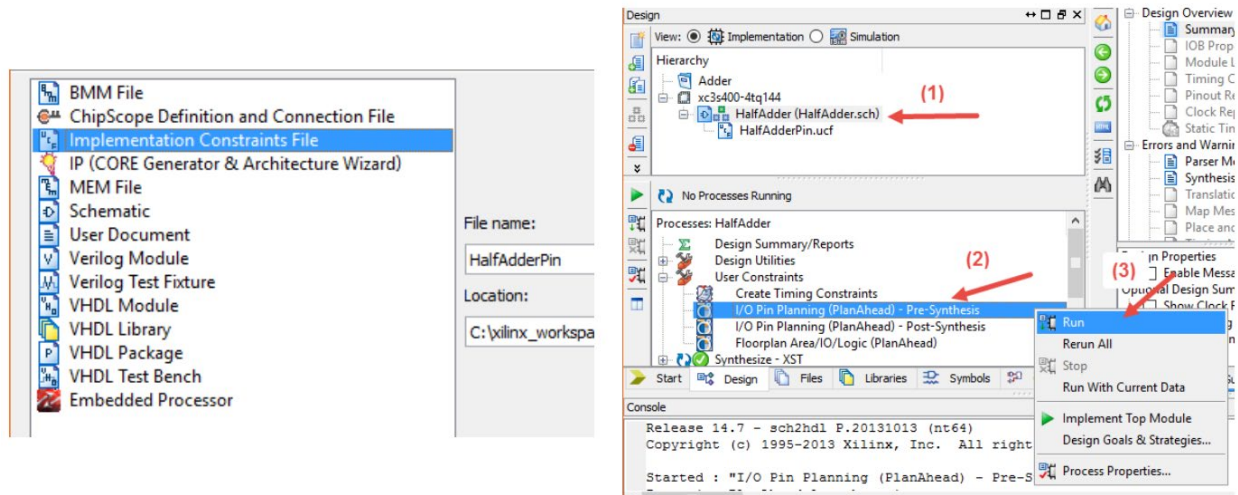
รูปที่ 12 การกำหนดชื่อตัวรับสัญญาณเข้าและออกใน I/O Marker

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

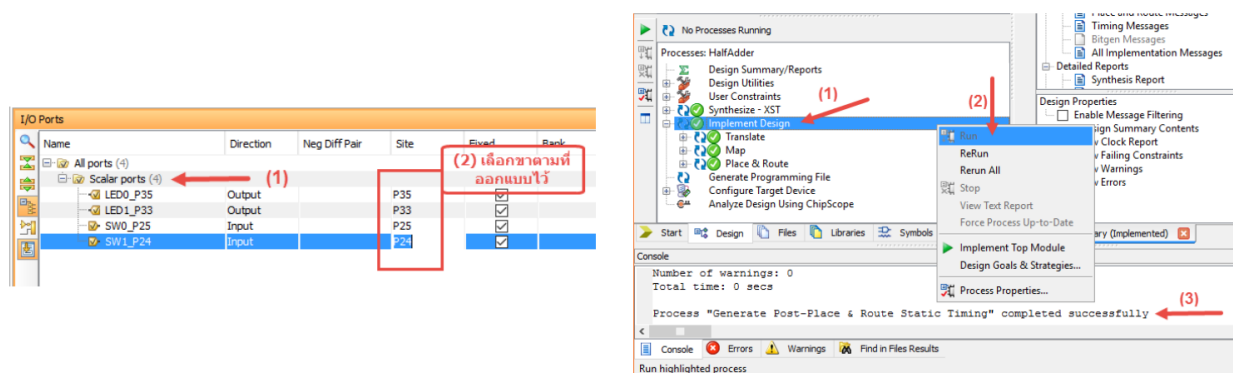
11. จับคู่อุปกรณ์กับหมายเลขพอร์ต โดยเลือก Project -> New Source -> Implementation Constraints File เพื่อจับคู่ระหว่างอุปกรณ์บนบอร์ดกับหมายเลขขาไอซี (Pin Number) ดังรูปที่ 14



รูปที่ 14 การจับคู่อุปกรณ์กับหมายเลขพอร์ต

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

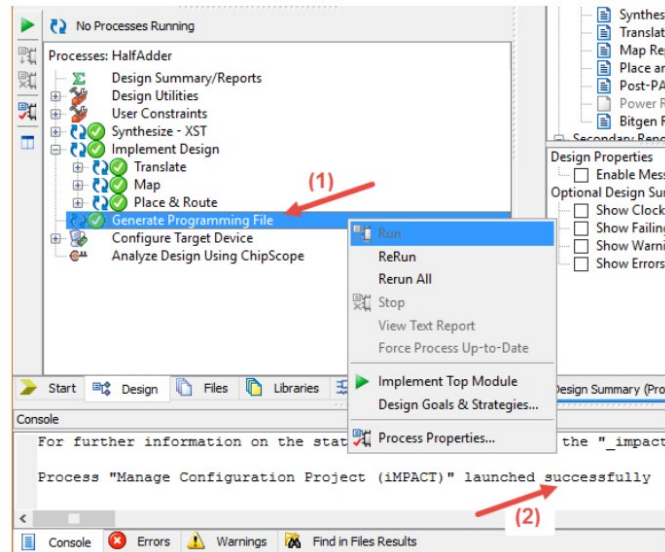
13. กำหนดหมายเลขขาในหน้าต่าง I/O Port ซึ่งอยู่ด้านล่างของโปรแกรม PlanAhead แล้วใส่รายละเอียดให้ครบเหมือนดังรูปที่ 15 (ซ้าย) หากไม่สามารถระบุเลขลงไปได้หรือโปรแกรมปฏิเสธแสดงว่ากำหนดหมายเลขขาผิด



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

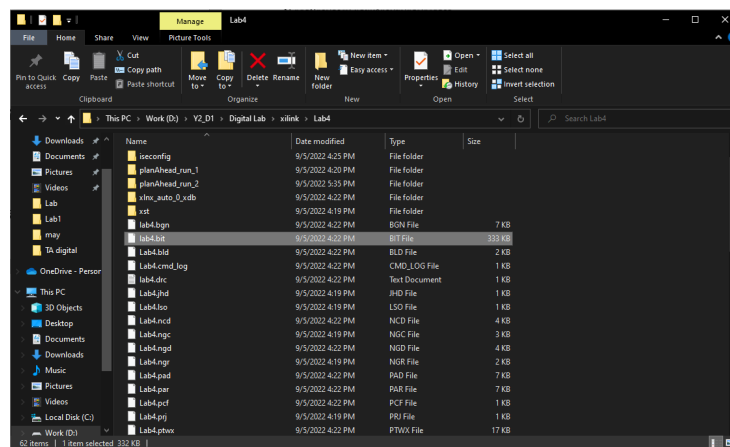
14. เริ่มขั้นตอน Implement Design โดยทำตามขั้นตอนในรูปที่ 15 (ขวา) ถ้าไม่มีข้อผิดพลาดจะแสดงเครื่องหมายถูกด้านหน้าหัวข้อ และแสดง Process “...” completed successfully ที่หน้าต่าง Console กรณีเกิดข้อผิดพลาดต้องกลับไปตรวจสอบการจับคู่อุปกรณ์กับหมายเลขพอร์ตตามขั้นตอนก่อนหน้า

15. บันทึกโปรแกรมลงในชิป FPGA โดยทำตามขั้นตอนในรูปที่ 16 (ซ้าย) เลือก Generate Programming File เลือก Run, Rerun All



รูปที่ 16 การ Generate Programmable File

16. เมื่อ Generate Programming File ทำงานเสร็จสิ้น สิ่งที่จะได้รับคือ “ชื่อโปรเจกต์.bit” ซึ่งต้องนำไป flash ลงบอร์ดต่อไป เพื่อทดลองใช้งานโปรแกรมที่เขียนบนบอร์ดต่อไป



รูปที่ 17 ไฟล์ bit ที่ได้หลังจาก Generate Programmable File

คู่มือการใช้งานโปรแกรม
FPGA Flasher
ในการโปรแกรมลงบอร์ด FPGA
ผ่านโมดูล FT2233HL

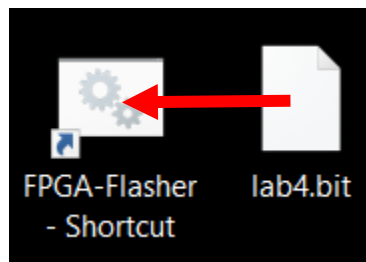
วิธีการอัปโหลดโปรแกรมลงบอร์ด FPGA ด้วยโปรแกรม FPGA Flasher

1. ลิงค์สำหรับดาวน์โหลด FPGA Flasher

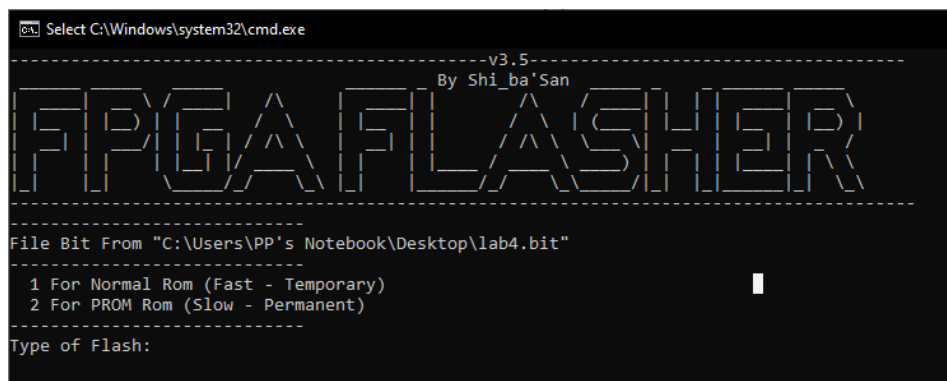


<https://github.com/ouoam/FPGA-Flasher> หรือ <https://bit.ly/44KREBI>

2. เมื่อได้ไฟล์ .bit มาจากการ Generate Programming File ให้ลากไฟล์ .bit ไปใส่ใน FPGA Flasher



รูปที่ 18 การลากไฟล์ .bit ไปใส่ใน FPGA Flasher (ในกรณีนี้คือ lab4.bit)



รูปที่ 19 หน้าตาโปรแกรม flasher เมื่อลากไฟล์ .bit มาใส่ (ในกรณีนี้คือ lab4.bit)

หมายเหตุ ประเภทของการ flash มีสองประเภท คือ For Normal Rom และ For PROM Rom

```
-----  
1 For Normal Rom (Fast - Temporary)  
2 For PROM Rom (Slow - Permanent)  
-----  
Type of Flash:
```

1. ถ้าหากเลือกโปรแกรมแบบ For Normal Rom เมื่อถอดปลั๊ก โปรแกรมจะหายไปจาก FPGA
2. หากต้องการให้ FPGA ยังมีโปรแกรมที่เรา run อยู่แม้จะถอดปลั๊กให้ลงโดยเลือกแบบ For PROM Rom โดยการโปรแกรมแบบ PROM จำนวนครั้งที่โปรแกรมได้จะมีอยู่อย่างจำกัด ให้ใช้อย่างคุ้มค่า
3. เลือกโหมดการ flash ที่ต้องการ แล้วกด enter แล้วรอให้โปรแกรม flash ลงบอร์ดจนเสร็จ
4. ถ้าจะอัปเดตโปรแกรมใหม่ ให้ปิดทุกอย่างแล้วเริ่มทำตามขั้นตอนตั้งแต่แรกอีกครั้ง
5. ไฟล์ที่ใช้ ใช้แค่ไฟล์ .bit

หมายเหตุ ถ้ามี Error ขึ้นว่า “No FTDI device found (using FTD2XX)”

```
Using built-in device list  
Using built-in cable list  
Cable ftdi type ftdi VID 0x0403 PID 0x6010 dbus data 00 enable 0b cbus data 00 data 00  
No FTDI device found (using FTD2XX)  
Press any key to continue . . .
```

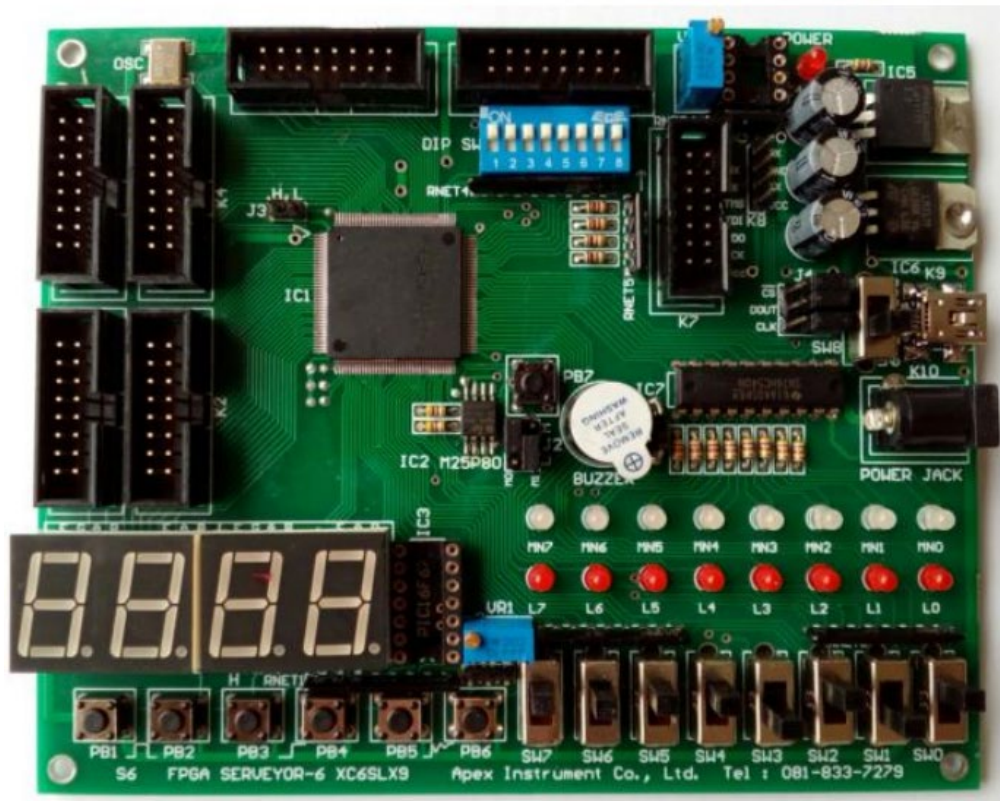
ให้ลง driver FTD2XX ให้เรียบร้อยก่อน แล้วจึงเริ่ม flash ใหม่ (ตามลิงค์ข้างล่าง)

https://ftdichip.com/wp-content/uploads/2021/08/CDM212364_Setup.zip

โปรแกรม และ คู่มือการใช้งานบอร์ด FPGA

Surveyor-6 XC6SLX9

ยี่ห้อ เอเพก อินสตรูเมนต์



รูปที่ 1 บอร์ด FPGA Surveyor-6 XC6SLX90

ชุดทดลองบอร์ด FPGA ยี่ห้อ เอเพก อินสตรูเมนต์ รุ่น FPGA Surveur-6 XC6SLX9

บอร์ด FPGA Surveur-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

รายการ Pin List

FPGA เบอร์ XC6SLX9-2TQG144C

Flash PROM เบอร์ M25P80-VMW6TG

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	K1 Pinout	K1 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	K1 Pinout	K1 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	K1 Pinout	K1 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	K1 Pinout	K1 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	K1 Pinout	K1 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

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

OSC	
Descriptions	FPGA Pinout
OSC	P123

BUZZER	
Descriptions	FPGA Pinout
BUZZER	P83

RS232	
Descriptions	FPGA Pinout
TX	P98
RX	P97

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

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

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