

ISE Quick Start Tutorial with VHDL

ก่อนที่จะปฏิบัติตามเอกสารนี้ ท่านจะต้องทำการติดตั้งโปรแกรม ISE WebPACK และ ModelSim ก่อน หากว่ายังไม่ได้ติดตั้งขอติดตั้งโดยดูวิธีการจากเอกสาร “วิธีการติดตั้งโปรแกรม”

ภาพรวมของ Tutorial

Tutorial นี้ดัดแปลงมาจาก ISE Quick Start Tutorial ของบริษัท Xilinx โดยดัดแปลงให้ใช้กับบอร์ดฝึกหัดการใช้งาน CPLD Explorer XC9572 ของบริษัท Apex Instrument

Tutorial นี้มีความมุ่งหมายให้เป็นแบบฝึกหัดอันแรกสำหรับผู้เริ่มต้นการเขียนโปรแกรมเพื่อออกแบบวงจรดิจิทัลด้วยภาษา VHDL ซึ่งเป็นภาษามาตรฐานที่นิยมใช้งานกันมาก Tutorial นี้จะสอนวิธีการสร้างโปรเจกโปรแกรมนับ (counter) ขนาด 4 บิต โดยนับสัญญาณที่กำเนิดจากสวิตช์กดติดปล่อยดับ และมีอีกสวิตช์อีกอันหนึ่งเป็นตัวกำหนดว่าให้นับขึ้นหรือนับลง

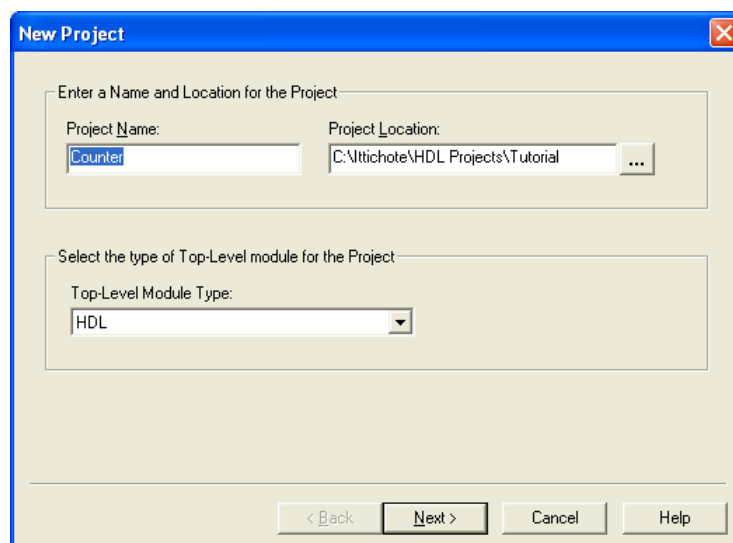
การเริ่มต้นการใช้งาน

ที่ Start Menu ให้เลือก Start → All Programs → Xilinx ISE 7.1i → Project Navigator

การสร้างโปรเจกใหม่ในโปรแกรม ISE

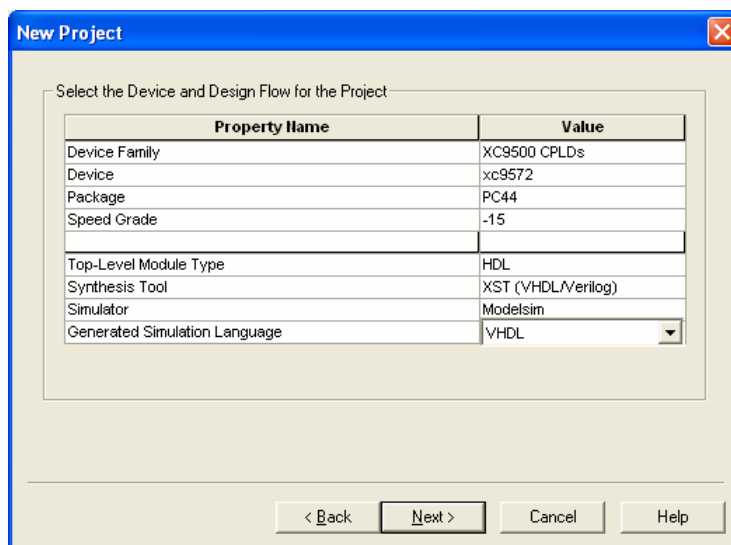
- ที่เมนูบาร์เลือก File > New Project โปรแกรมเปิดหน้าต่าง New Project ขึ้นมา
- หน้าต่างแรกให้ตั้งค่าดังต่อไปนี้ แล้วคลิกปุ่ม Next

- Project Location: (กำหนด Directory สำหรับเก็บโปรเจกที่จะสร้างขึ้น)
- Project Name: Counter (โปรแกรมจะใส่ชื่อ Subdirectory ชื่อ Counter ให้โดยอัตโนมัติ)
- Top-Level Module Type: HDL



3. โปรแกรมจะแสดงหน้าต่างให้ตั้งค่าคุณสมบัติ ให้ตั้งค่าดังต่อไปนี้ แล้วคลิกปุ่ม Next

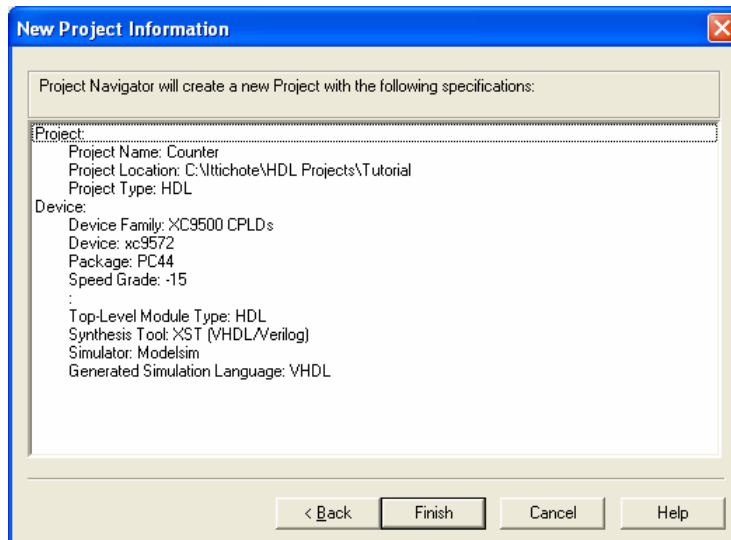
- Device Family: XC9500 CPLDs
- Device: xc9572
- Package: PC44
- Speed Grade: -15
- Top-Level Module Type: HDL
- Synthesis Tool: XST (VHDL/Verilog)
- Simulator: ModelSim
- Generated Simulation Language: VHDL



4. โปรแกรมจะทำการแสดงหน้าต่างเพื่อให้สร้างโปรแกรม (source) ใหม่ แต่เนื่องจากเราจะสร้างใหม่ ภายหลัง ตอนนี้ให้คลิกปุ่ม Next เลย

5. คลิกปุ่ม Next โปรแกรมจะทำการแสดงหน้าต่างให้เลือกโปรแกรม (source) ที่มีอยู่เดิม คลิกปุ่ม Next

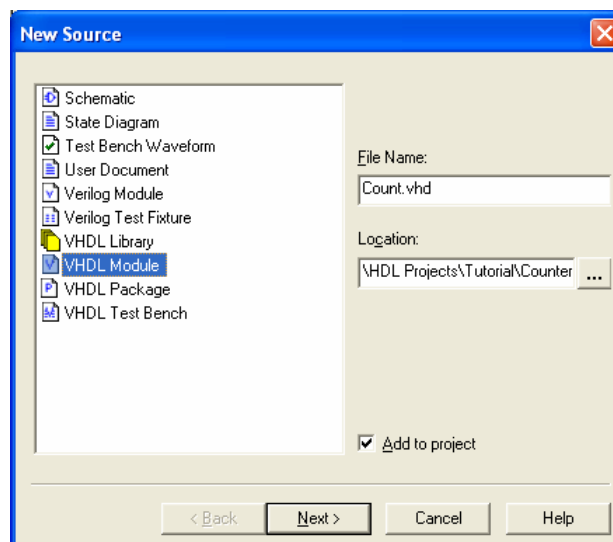
6. โปรแกรมจะแสดงสรุปข้อมูลของโปรเจกใหม่ที่จะสร้างขึ้น คลิกปุ่ม Finish



การสร้าง VHDL Source

โปรแกรมนี้จะมีสองสัญญาณอินพุตได้แก่ CLOCK และ DIRECTION โปรแกรมจะนับขึ้นหรือลงขึ้นอยู่กับสัญญาณ DIRECTION และมีสัญญาณเอาต์พุตเป็นบิตขนาด 4 บิต เรียกว่า COUNT_OUT

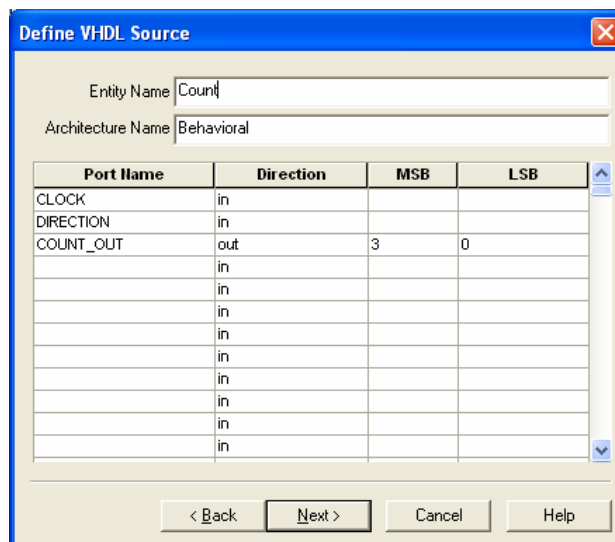
1. จากเมนูบาร์เลือก Project > New Source โปรแกรมจะเปิดหน้าต่าง New Source ขึ้นมา
2. เลือก VHDL Module และใส่ชื่อไฟล์ว่า Count
3. ตรวจสอบว่ามีเครื่องหมายถูกในช่อง Add to project
4. คลิกปุ่ม Next



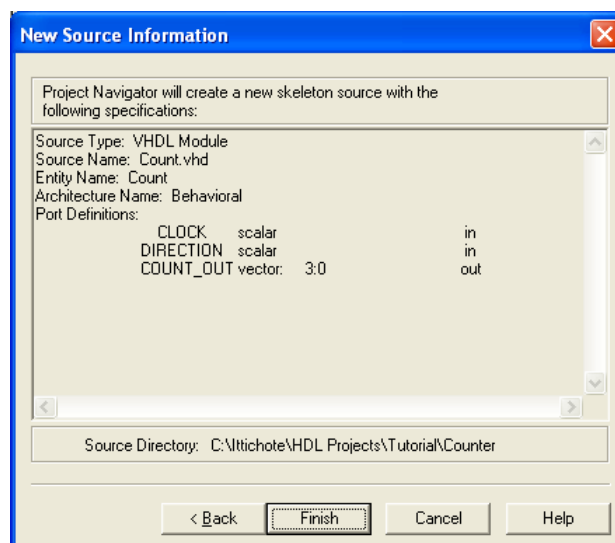
5. กำหนด Port สำหรับ VHDL Source (ดังแสดงในรูป) เมื่อเสร็จให้คลิกปุ่ม Next

- ในคอลัมน์ Port Name ในแต่ละแถวให้พิมพ์ CLOCK, DIRECTION และ COUNT_OUT

- ในคอลัมน์ Direction ให้เลือกแต่ละ Port เป็น input, output หรือ inout สำหรับ CLOCK และ DIRECTION ให้เลือก in และ สำหรับ COUNT_OUT เลือก out
- เพื่อกำหนดว่าพอร์ท COUNT_OUT เป็นบัสขนาด 4 บิต ให้เลือก 3 สำหรับคอลัมน์ MSB (Most Significant Bit) และ 0 ในช่อง LSB (Least Significant Bit)



6. โปรแกรมจะแสดงสรุปข้อมูลของ Source ใหม่ที่จะสร้างขึ้น คลิกปุ่ม Finish



7. โปรแกรมจะทำการเปิดไฟล์ Count.vhd ที่สร้างขึ้นใหม่โดยอัตโนมัติ ถ้าไฟล์ยังไม่ได้ถูกเปิด ให้ดับเบิลคลิกที่ไฟล์ Count.vhd ในหน้าต่าง Sources in Project เพื่อเปิดไฟล์นี้ขึ้นมา

ไฟล์ Count.vhd จะประกอบด้วย

- Header information.
- Library declaration and use statements.
- Entity declaration และ Architecture statement.

8. ในส่วน header ให้ใส่ข้อมูลต่อไปนี้

ผศ.ดร. อธิโชค จักรไพวงศ์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยมหิดล

VHDL Tutorial

Last Updated 11/12/2006

- Design Name: Count.vhd
- Project Name: Counter
- Target Device: XC9572-15PC44
- Description: This is the top level HDL file for an up/down counter.
- Dependencies: None

หมายเหตุ: แม้ว่าการเขียนข้อมูลส่วนนี้จะไม่จำเป็น แต่ก็เป็นข้อควรปฏิบัติในการเขียนโปรแกรม

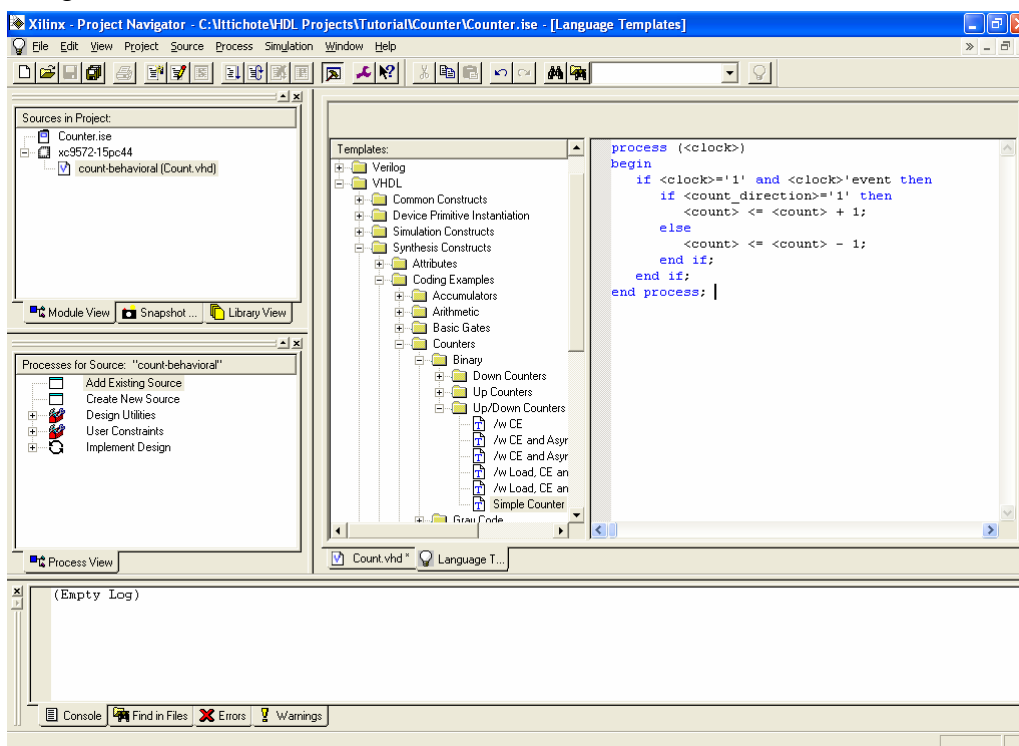
การใช้ Language Templates (VHDL)

โปรแกรม ISE มี Template โปรแกรมในภาษาต่างๆ (ABEL, UCF, Verilog, VHDL) สำหรับวงจรที่ใช้บ่อย ใน Tutorial นี้จะใช้ Template สำหรับวงจร Up/Down Simple Counter ในการจะเรียกใช้ Template ให้ทำดังต่อไปนี้

1. เลือก Edit > Language Templates
2. คลิกที่เครื่องหมาย “+” เพื่อที่จะเปิดโฟลเดอร์ต่อไปนี้

VHDL > Synthesis Constructs > Coding Examples > Counters > Binary > Up/Down Counters > Simple Counter

3. คัดลอกโค้ด Language Template มาจากหน้าต่างด้านขวา แล้ววางลงบนไฟล์ Count.vhd ระหว่างคำสั่ง begin และ end ของ architecture



4. แทนชื่อที่อยู่ในกรอบเครื่องหมาย <> ด้วยชื่อพอร์ทซึ่งได้กำหนดไว้ตอนที่สร้างไฟล์ VHDL แม้ว่าภาษา VHDL จะไม่สนใจตัวอักษรใหญ่เล็ก แต่ก็ขอให้เขียนให้ตรงกันเพื่อความเป็นระเบียบเรียบร้อย ให้ทำการแก้ไขดังต่อไปนี้

แทนคำว่า <clock> ด้วย CLOCK

แทนคำว่า <count direction> ด้วย DIRECTION

แทนคำว่า <count> ด้วย COUNT_OUT

5. ปิดหน้าต่าง Language Template

การแก้ไข VHDL Source เพิ่มเติม

1. ในไฟล์ VHDL โปรแกรมจะไม่ยอมให้อ่านค่าจากพอร์ทที่ถูกกำหนดให้เป็นเอาต์พุทโดยตรง ดังนั้นจึงต้องทำการกำหนดตัวแปรชั่วคราวที่เก็บค่าของเอาต์พุทเอาไว้ เราจะทำการประกาศตัวแปรชื่อ count_int ให้เป็นตัวแปรชั่วคราวนี้ โดยการประกาศดังต่อไปนี้ได้ architecture declaration และเหนือคำสั่ง begin อันแรก

```
signal count_int : std_logic_vector(0 to 3) := "0000";
```

2. ให้แทรกคำสั่งเพื่อเรียกใช้ตัวแปร count_int ที่ตำแหน่งต่อไปนี้

```
if DIRECTION = '1' then
    count_int <= count_int + 1;
else
    count_int <= count_int - 1;
end if;
```

3. ใต้คำสั่ง end process ให้เพิ่มคำสั่งว่า

```
COUNT_OUT <= count_int;
```

4. บันทึกไฟล์โดยเลือก File > Save จากเมนูบาร์

5. เมื่อแก้ไขเสร็จทั้งหมดแล้วให้ตรวจทานว่าไฟล์ Count.vhd ควรมีคำสั่งดังนี้

```
architecture Behavioral of Count is
    signal count_int : std_logic_vector(0 to 3):="0000";

begin

    process (CLOCK)
    begin
        if CLOCK='1' and CLOCK'event then
            if DIRECTION = '1' then
                count_int <= count_int + 1;
            else
                count_int <= count_int - 1;
            end if;
        end if;
    end process;

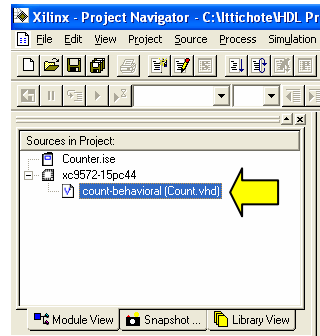
    COUNT_OUT <= count_int;

end Behavioral;
```

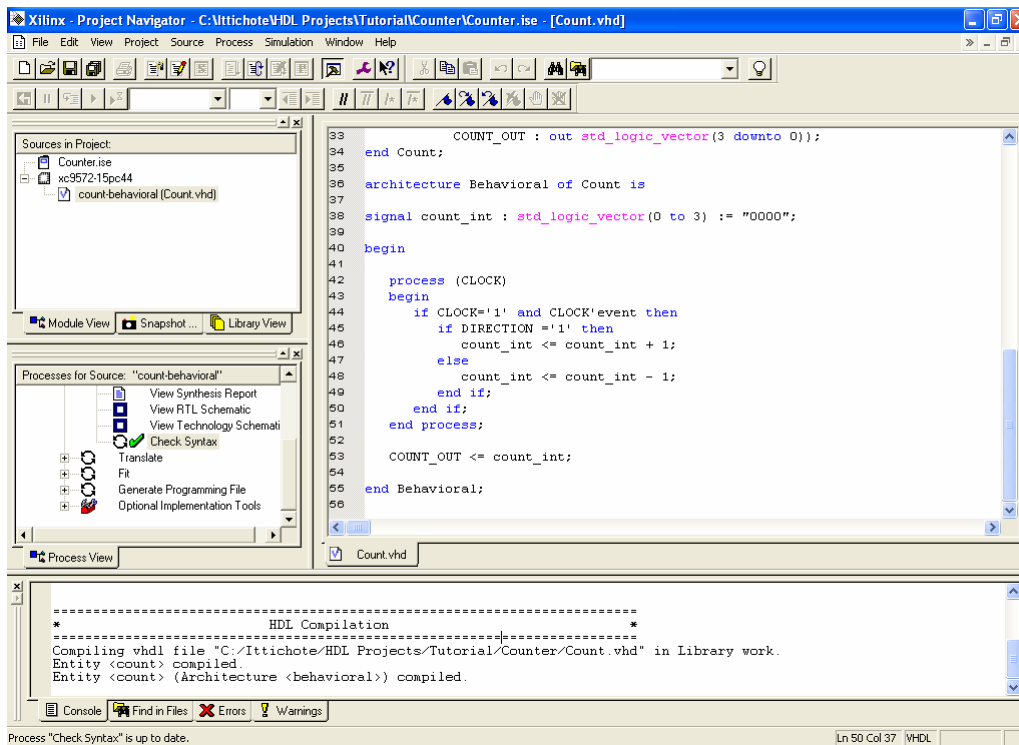
การตรวจสอบความถูกต้องของโปรแกรมที่สร้างขึ้น

เมื่อทำการสร้างไฟล์ VHDL ขึ้นมาเรียบร้อยแล้ว ขั้นต่อไปจะต้องทำการตรวจสอบความถูกต้อง โดยให้ทำตามขั้นตอนต่อไปนี้

1. เลือกไฟล์ Count.vhd ในหน้าต่าง Source in Projects โปรแกรมจะทำการแสดง Process ที่เกี่ยวข้องกับ source นี้ในหน้าต่าง Processes for Source



2. ในหน้าต่าง Processes for Source คลิกเครื่องหมาย “+” หน้าคำว่า Implement Design > Synthesize-XST
3. ดับเบิลคลิกที่ Check Syntax เมื่อการตรวจสอบเสร็จจะมีเครื่องหมายแสดงผลการตรวจสอบแสดงอยู่หน้าชื่อ
 - ถ้ามีเครื่องหมายลูกศรชี้เขียว แสดงว่าโปรแกรมถูกต้อง โดยไม่มีคำเตือน
 - ถ้ามีเครื่องหมายกากบาทสีแดง แสดงว่าโปรแกรมหักข้อผิดพลาด
 - ถ้ามีเครื่องหมายตกใจสีเหลือง แสดงว่าโปรแกรมถูกต้อง แต่มีคำเตือนเกิดขึ้น
 - ถ้ามีเครื่องหมายคำถามสีส้ม แสดงว่าโปรแกรมถูกแก้ไขหลังจากการตรวจสอบครั้งสุดท้าย และควรที่จะต้องทำการตรวจสอบอีกครั้ง
4. มองในแท็บ Console ของหน้าต่าง Transcript ด้านล่าง เพื่อดูผลการตรวจสอบ

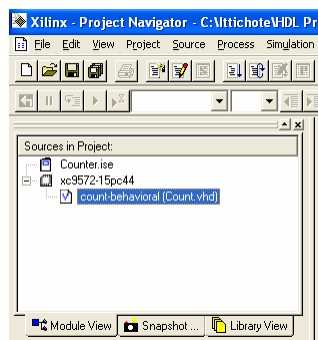


การจำลองการทำงาน (Design Simulation)

การสร้าง Test Bench สำหรับการ Simulation

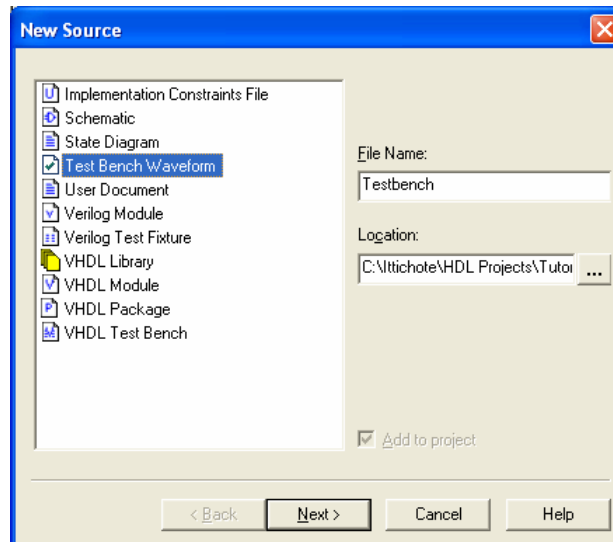
ในส่วนนี้เราจะทำการสร้าง Test bench waveform ซึ่งจะเป็นการจำลองสัญญาณอินพุตเพื่อใช้ในการ Simulate การทำงานของโปรแกรม เราจะต้องใช้ Test bench waveform ร่วมกับ Simulator เพื่อสร้างเอาท์พุทเพื่อจะดูว่าโปรแกรมทำงานตามที่ต้องการหรือไม่ เราจะใช้ Waveform Editor ในการสร้างไฟล์ Test bench waveform (TBW)

1. เลือกไฟล์ Count.vhd ในหน้าต่าง Sources in Project



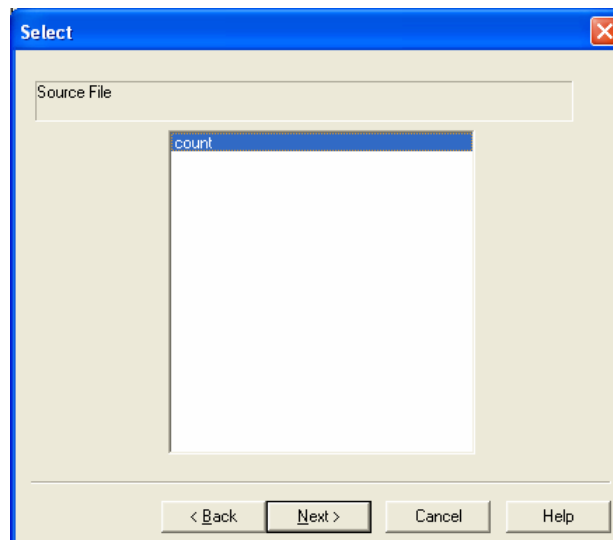
2. สร้าง Source ใหม่ ที่เมนูบาร์เลือก Project > New Source

3. เลือก Test Bench Waveform และพิมพ์ชื่อ Testbench ในช่อง File Name



4. คลิกปุ่ม Next

5. หน้าต่างแสดงเพื่อให้เลือกไฟล์ที่จะใช้ในการทดสอบ ให้เลือก count แล้วคลิกปุ่ม Next



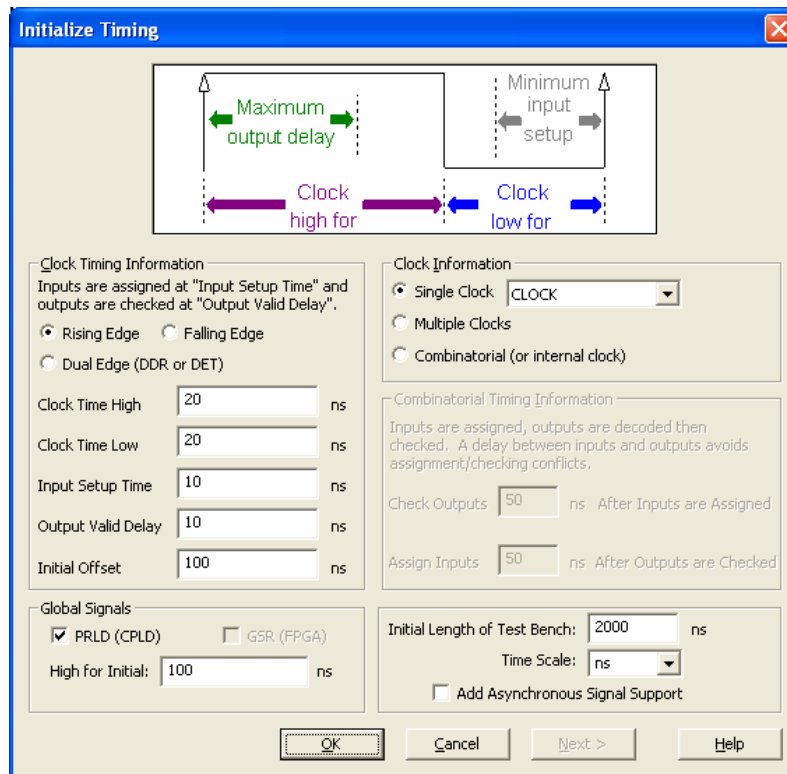
6. คลิกปุ่ม Finish

7. โปรแกรมจะแสดงหน้าต่าง Initialize Timing ซึ่งเราจะต้องกำหนดค่าเริ่มต้นของสัญญาณที่จะใช้ทดสอบ ให้ดังค่าดังต่อไปนี้

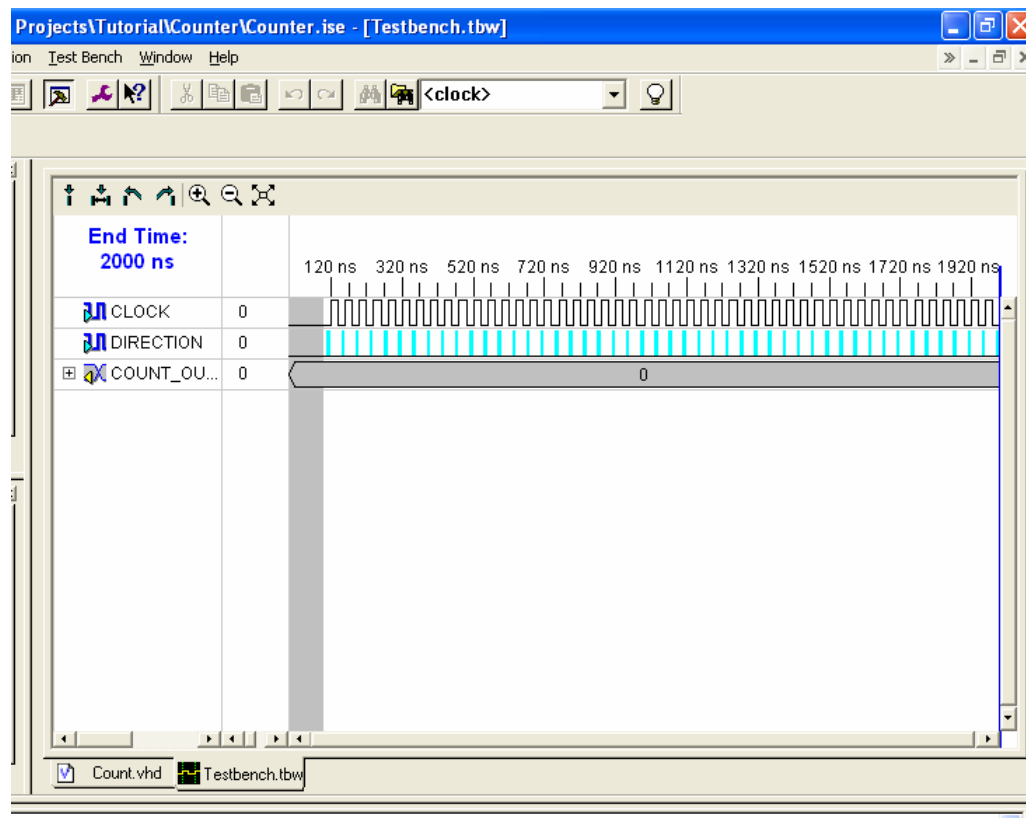
- Clock Time High: 20 ns.
- Clock Time Low: 20 ns.
- Input Setup Time: 10 ns.
- Output Valid Delay: 10 ns.
- Initial Offset: 0 ns
- Global Signals: PRLD (CPLD)

หมายเหตุ: เมื่อคลิกที่ช่อง PRLD (CPLD) ช่อง Initial Offset จะเปลี่ยนเป็น 100 ns โดยอัตโนมัติ

- Initial Length of Test Bench: 2000 ns



8. คลิกปุ่ม OK เพื่อเปิด Waveform Editor

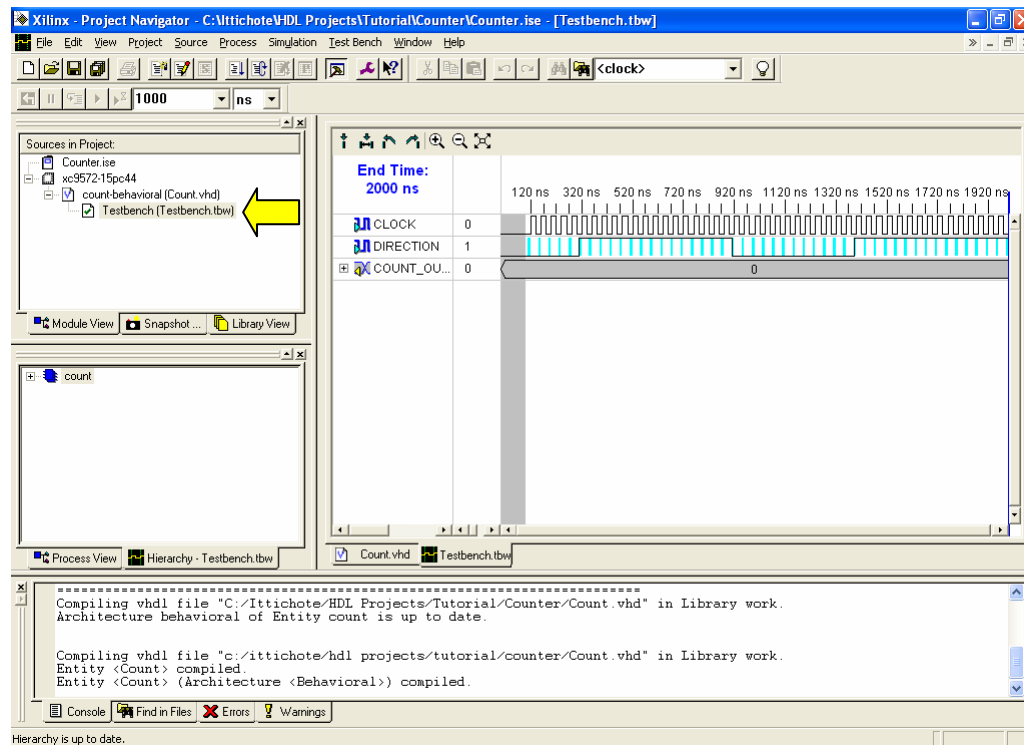


แถบสีฟ้าในแถวของ DIRECTION เป็นตำแหน่งการเปลี่ยนสัญญาณอินพุตซึ่งเป็นไปตาม Input Setup Time ที่ได้กำหนดค่าในหน้าต่าง Initialize Timing ก่อนหน้านี้ ซึ่งในที่นี้ขอให้สังเกตว่า ตำแหน่งสีฟ้านี้จะอยู่ที่ตำแหน่ง Rising edge ของสัญญาณ CLOCK

9. ในที่นี้เรามีสัญญาณที่สามารถเปลี่ยนได้เพียงหนึ่งตัว ก็คือพอร์ท DIRECTION ให้ลองเปลี่ยนสัญญาณของพอร์ท DIRECTION ดังต่อไปนี้

- กดที่แถบสีฟ้าที่ตำแหน่งประมาณ 300 ns ให้สัญญาณเปลี่ยนจากต่ำไปสูง
- กดที่แถบสีฟ้าที่ตำแหน่งประมาณ 900 ns ให้สัญญาณเปลี่ยนจากสูงไปต่ำ
- กดที่แถบสีฟ้าที่ตำแหน่งประมาณ 1400 ns ให้สัญญาณเปลี่ยนจากต่ำไปสูง

10. ที่เมนูบาร์ คลิกที่ File > Save เพื่อจัดเก็บ Waveform ที่เพิ่งสร้างขึ้น ให้สังเกตว่าไฟล์ Testbench.tbw จะเพิ่มขึ้นมาในหน้าต่าง Sources in Project

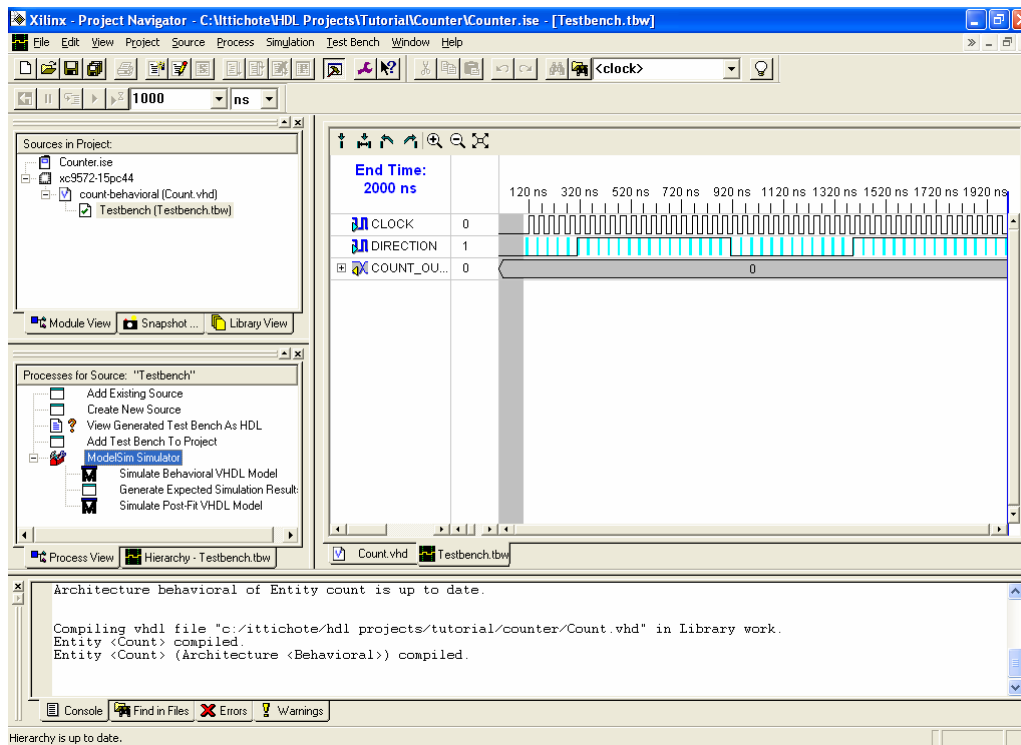


11. เปิดหน้าต่าง Testbench.tbw

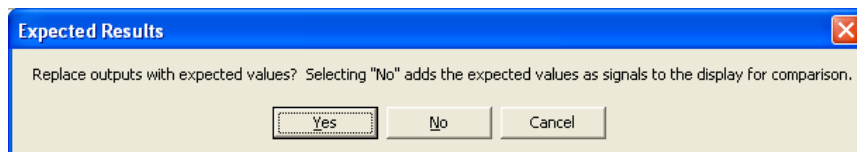
การสร้าง Expected Results ให้กับ Testbench Waveform

ในขั้นตอนนี้เราจะสร้าง Expected Results หรือผลลัพธ์ที่คาดว่าจะได้ให้กับ Testbench เมื่อให้อินพุตตามที่ได้กำหนดไว้ ขั้นตอนนี้เพื่อนำผลที่ได้มาเทียบกับผลที่ได้จากการ Simulate ซึ่งจะมีประโยชน์ในการทำ Simulate Post-Place & Route HDL Model เพื่อเทียบผลว่าโปรแกรมทำงานตามที่ได้คาดไว้หรือไม่ ในการสร้าง Expected Results สามารถทำได้โดยกำหนดเองหรือว่าให้โปรแกรมสร้างให้โดยอัตโนมัติก็ได้ การให้โปรแกรมสร้างให้โดยอัตโนมัติทำได้ดังนี้

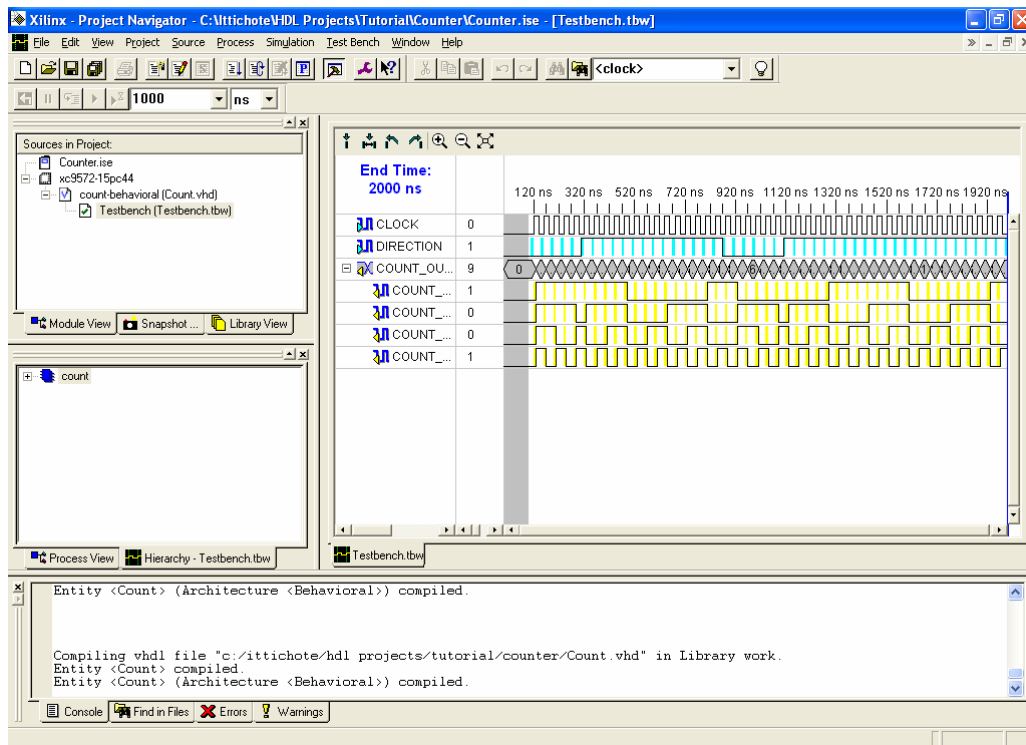
1. ให้เลือกไฟล์ Testbench.tbw ในหน้าต่าง Sources in Project เมื่อเลือกแล้วจะมี Processes ชื่อ ModelSim Simulator ปรากฏอยู่ในหน้าต่าง Processes for Source



2. ในหน้าต่าง Processes for Source ให้ดับเบิ้ลคลิกที่ Generate Expected Simulation Results
3. โปรแกรมจะแสดงหน้าต่าง Expected Results ให้คลิกปุ่ม Yes เพื่อวางผลลัพธ์ใน Waveform editor



4. คลิกที่เครื่องหมาย "+" เพื่อแสดงแต่ละบิตของบัส COUNT_OUT จะได้ Expected Results ดังแสดงในรูป



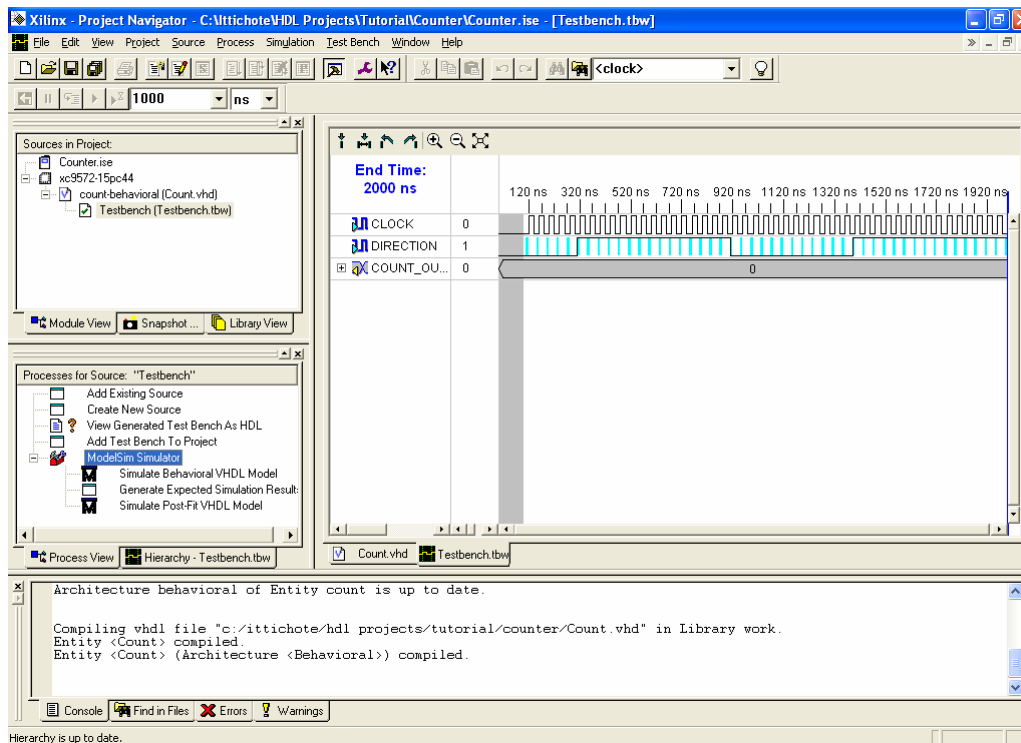
5. ที่เมนูบาร์ คลิก File > Save เพื่อบันทึก Waveform

6. เปิด Waveform Editor

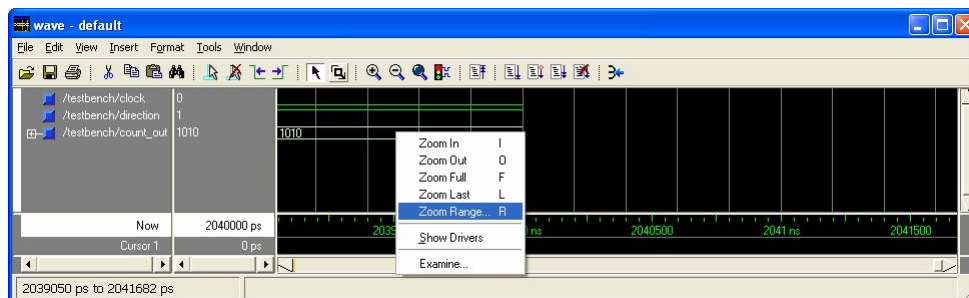
Simulating the Behavioral Model ด้วยโปรแกรม ModelSim

เนื่องจากเราได้ทำการติดตั้งโปรแกรม ModelSim ก่อนหน้านี้แล้ว เราจะสามารถทำการ Simulate การทำงานของโปรแกรมได้โดยการเรียกใช้โปรแกรม ModelSim ผ่าน Processes ในโปรแกรม ISE ให้ปฏิบัติดังต่อไปนี้

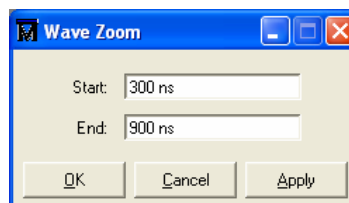
1. ให้เลือกไฟล์ Testbench.tbw ในหน้าต่าง Sources in Project เมื่อเลือกแล้วจะมี Processes ชื่อ ModelSim Simulator ปรากฏอยู่ในหน้าต่าง Processes for Source



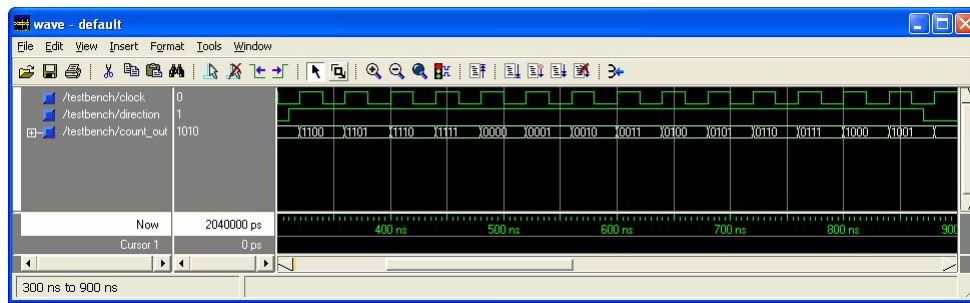
2. ในหน้าต่าง Processes for Source ให้ดับเบิลคลิกที่ Simulate Behavioral VHDL Model โปรแกรม ModelSim จะถูกเปิดขึ้นเพื่อทำการ Simulation เมื่อทำเสร็จเรียบร้อยแล้วจะได้ผลการ Simulation
3. ให้เลือกชมผลการ Simulation โดยคลิกขวาที่พื้นที่กลางหน้าต่าง wave-default เลือก Zoom Range



4. โปรแกรมให้แสดงหน้าต่าง Wave Zoom ให้เลือก Start: 300 ns และ End: 900 ns



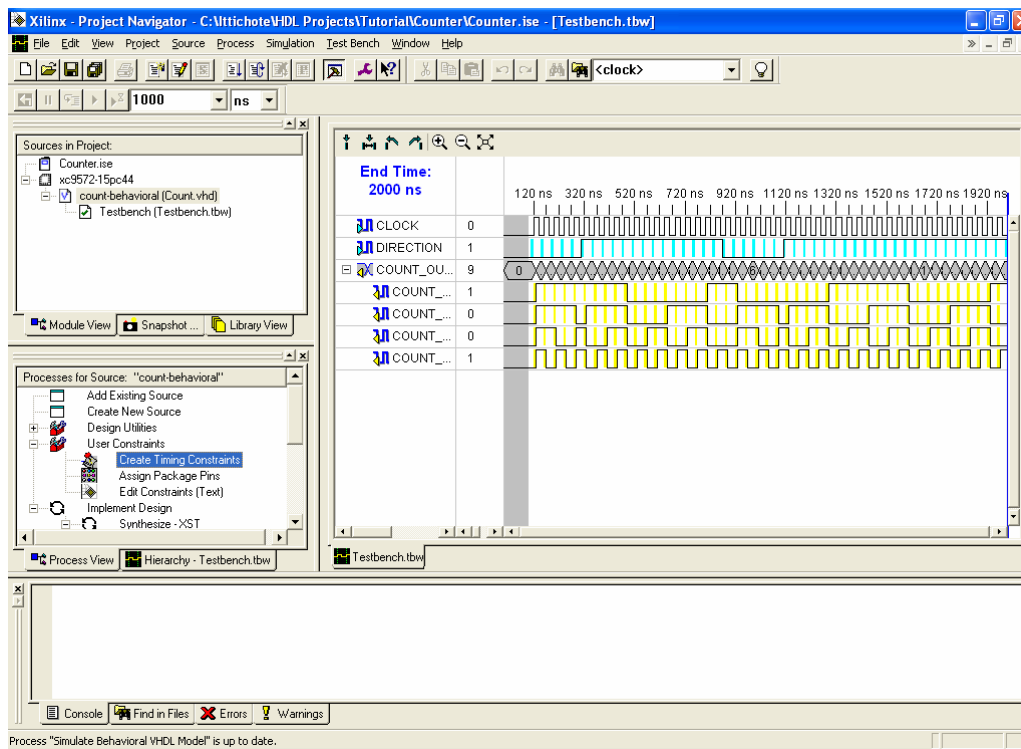
5. โปรแกรมจะแสดงผลการ Simulation ในช่วง 300 ns ถึง 900 ns ดังแสดงในรูป



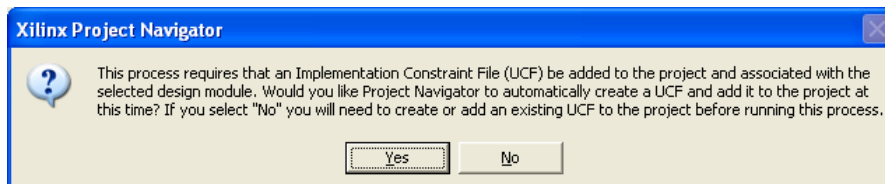
6. ปิดโปรแกรม ModelSim

การสร้างไฟล์ Constraints

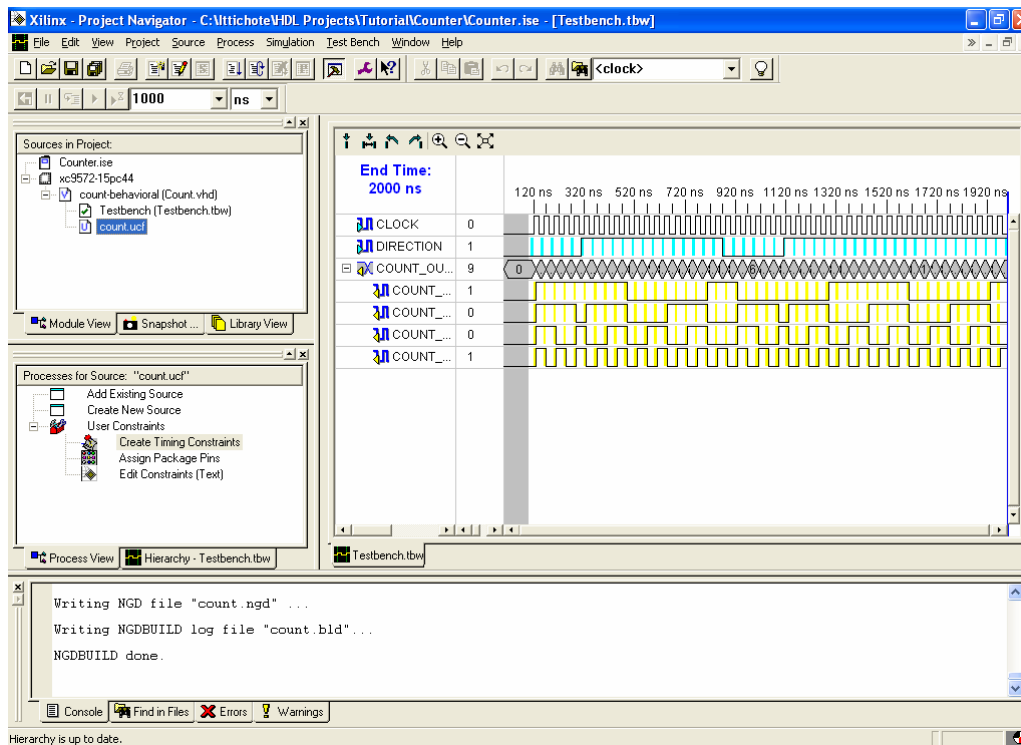
1. เลือกไฟล์ Count.vhd ในหน้าต่าง Sources in Project
2. ในหน้าต่าง Processes for Source คลิกที่เครื่องหมาย "+" หน้า User Constraints เพื่อขยายรายการ
3. ดับเบิลคลิก Create Timing Constraints โปรแกรมจะทำการสร้างไฟล์ UCF – User Constraints File ขึ้นมา



4. โปรแกรมจะถามว่าต้องการให้เชื่อมต่อไฟล์ UCF นี้กับโปรเจกเดิมหรือไม่ คลิกปุ่ม Yes

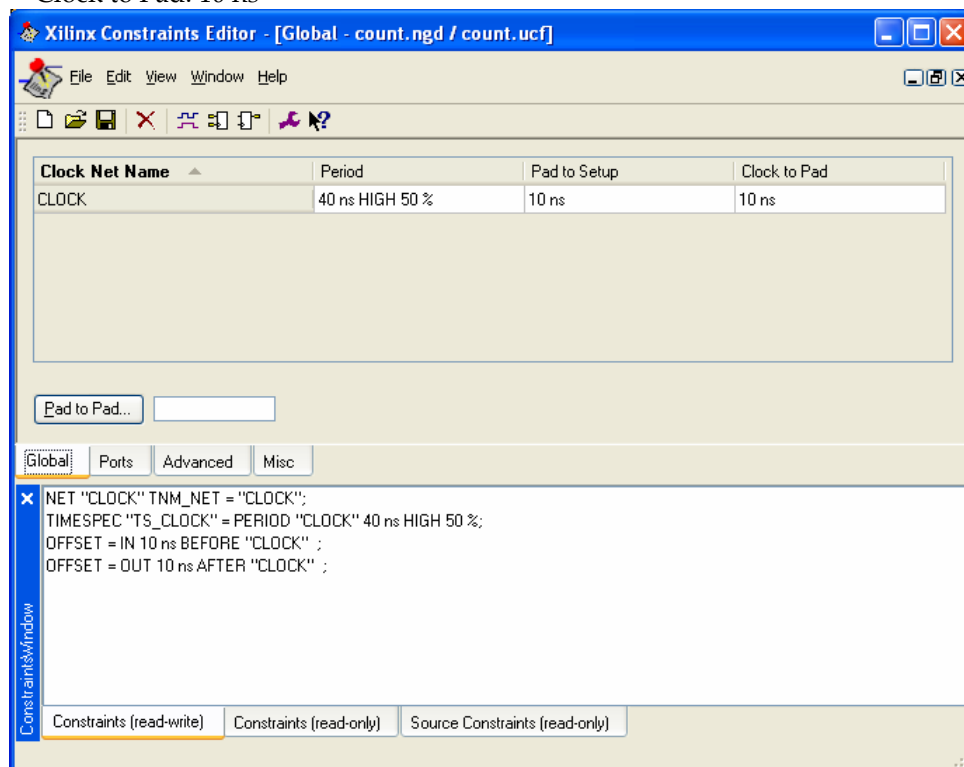


ไฟล์ count.ucf จะถูกสร้างและเพิ่มเข้ามาที่โปรเจกซึ่งสามารถดูได้ที่หน้าต่าง Sources in Project



5. เลือกหน้าต่าง Xilinx Constraints Editor เลือกแท็บ Global ใส่ค่าต่างๆ ต่อไปนี้สำหรับ CLOCK ค่าเหล่านี้ควรจะสอดคล้องกับค่าที่กำหนดไว้ใน Period, Input Setup และ Output Delay ใน Test

- Period: 40 ns
- Pad to Setup: 10 ns
- Clock to Pad: 10 ns



6. ที่เมนูบาร์ คลิก File > Save

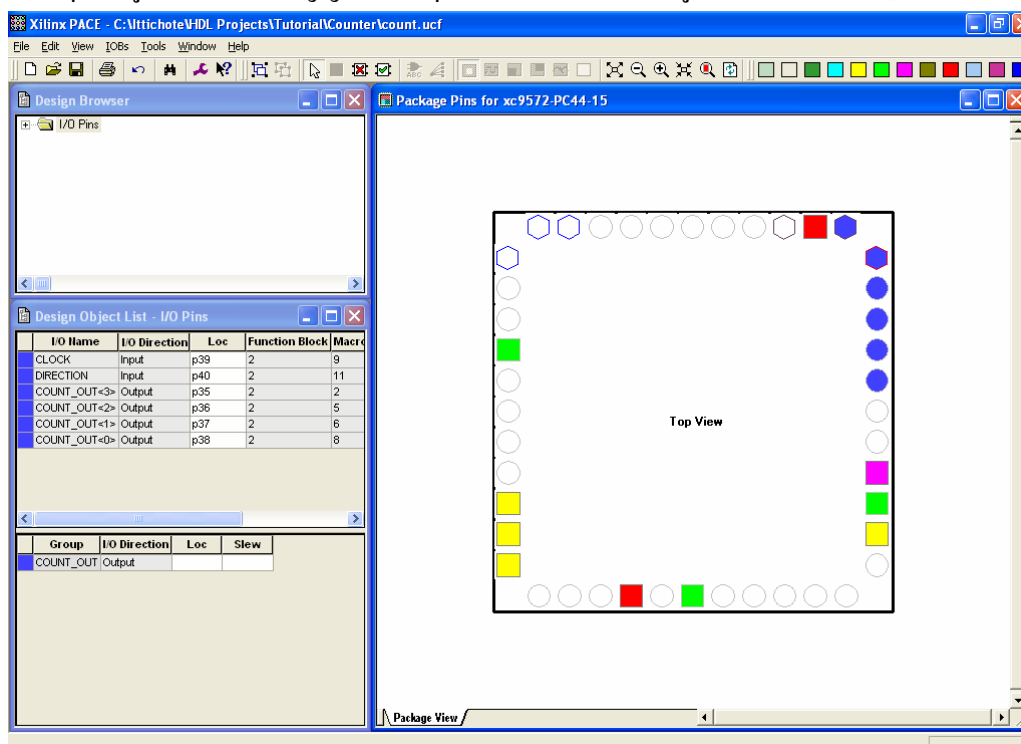
7. ปิดหน้าต่าง Xilinx Constraints Editor

การกำหนดตำแหน่งขา (Pin Location)

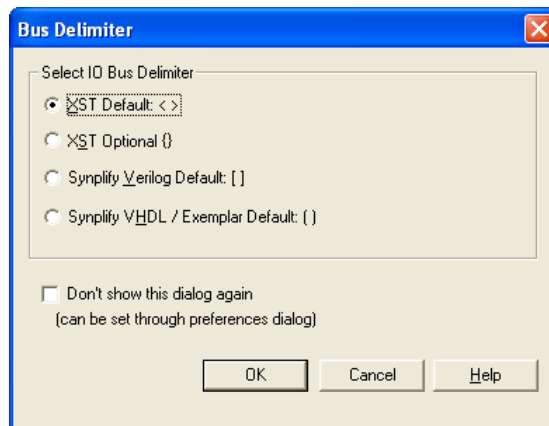
นอกเหนือจาก Timing Constraints แล้ว เราจะต้องเลือกตำแหน่งขาของอุปกรณ์ให้กับอินพุตกับเอาต์พุตที่ได้ออกแบบ โดยให้ปฏิบัติตามดังต่อไปนี้

1. เลือกไฟล์ Count.ucf ในหน้าต่าง Sources in Project
2. ในหน้าต่าง Processes for Source คลิกที่เครื่องหมาย “+” หน้า User Constraints เพื่อขยายรายการ
3. ดับเบิลคลิกที่ Assign Package Pins โปรแกรมจะเปิดหน้าต่าง Xilinx PACE (Pinout and Area Constraints Editor) ขึ้นมา
4. เลือกขาตามรายการต่อไปนี้
 - CLOCK: p39
 - COUNT_OUT<0>: p38
 - COUNT_OUT<1>: p37
 - COUNT_OUT<2>: p36
 - COUNT_OUT<3>: p35
 - DIRECTION: p40

หมายเหตุ: ให้ดูการเชื่อมต่อสัญญาณของชุดคิททดสอบจากจากคู่มือของ CPLD Explorer XC9572



5. ที่เมนูบาร์เลือก File > Save
6. โปรแกรมจะแสดงหน้าต่าง Bus Delimiter เพื่อให้เลือกชนิดของ Bus Delimiter ให้เลือก XST Default <> แล้วคลิกปุ่ม OK



7. ปิดหน้าต่าง Xilinx PACE

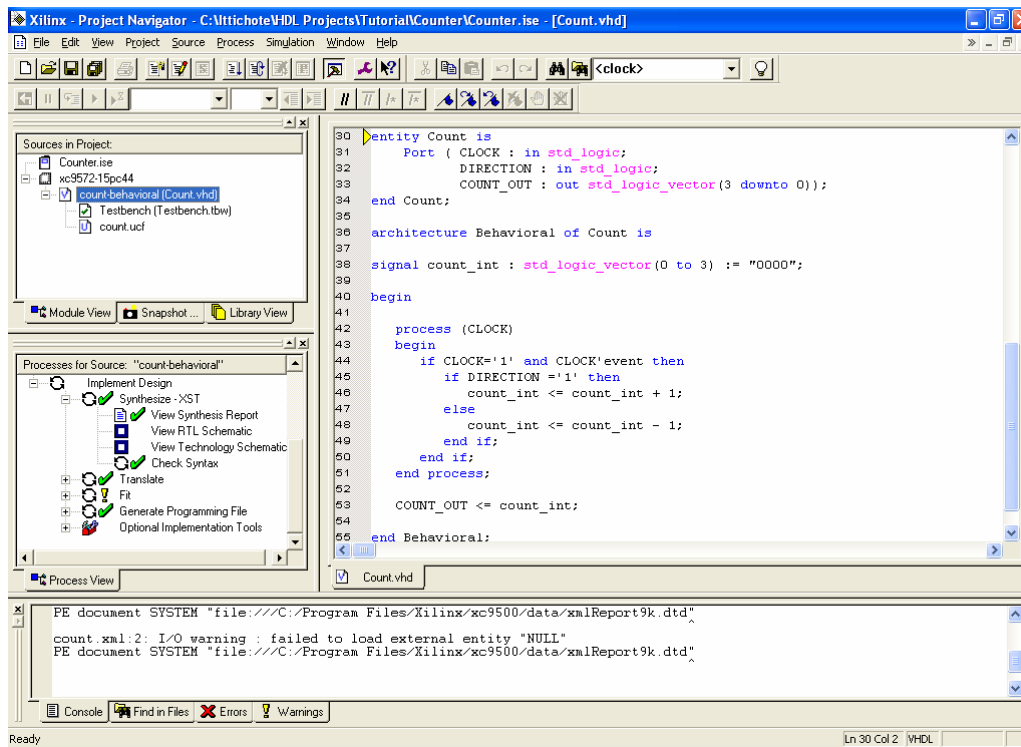
การ Synthesis and Implementation

ขั้นตอนการ Synthesis และ Implementation มีดังต่อไปนี้

- Synthesis with XST
- FPGA Implementation:
- Translate (which runs the NGDBuild program)
- Map
- Place and Route (PAR).

ต่อไปเราจะดำเนินการตามขั้นตอนต่อไปนี้

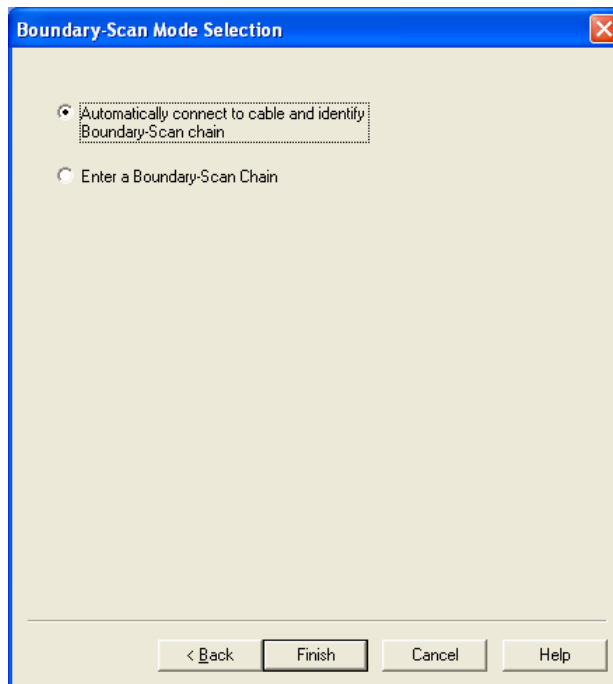
1. เลือกไฟล์ Count.vhd ในหน้าต่าง Sources in Project
2. ในหน้าต่าง Processes for Source คลิกที่เครื่องหมาย “+” หน้า Implement Design เพื่อขยายรายการ
3. ดับเบิ้ลคลิกที่ Implement Design ระบบจะดำเนินการ Translate, Map และ Place & Route ให้ตามลำดับ (ไม่ต้องสนใจค่าเตือนที่ Fit)



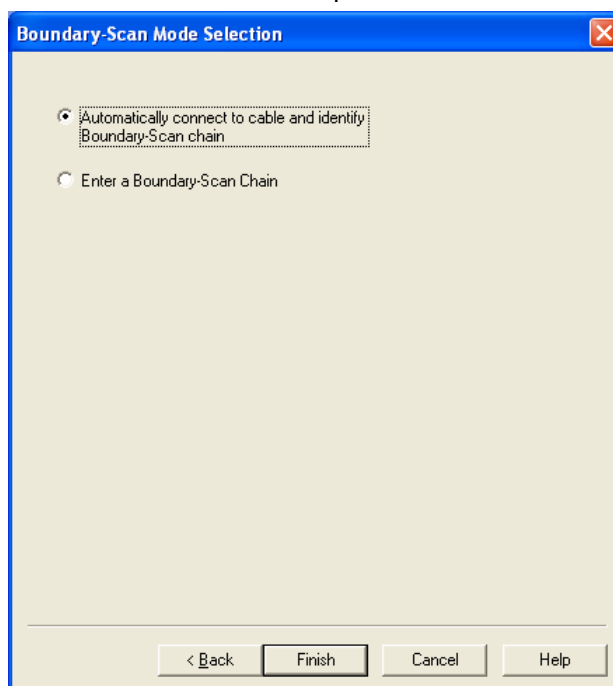
การโปรแกรมข้อมูลวงจรลงชิพ CPLD

เมื่อปรากฏเครื่องหมายถูกหน้า Generate Programming File แสดงว่าเราสามารถที่จะโปรแกรมข้อมูลวงจรลงบนชิพ CPLD ได้แล้ว โดยที่ไฟล์ข้อมูลที่จะดาวน์โหลดลงบนชิพจะเป็นไฟล์ Bitstream หรือไฟล์นามสกุล .bit สำหรับ FPGA และไฟล์ Jedec หรือไฟล์นามสกุล .jed สำหรับ CPLD ดังนั้นในขั้นนี้ให้ตรวจสอบดูใน Directory ของโปรเจกต์ว่าจะต้องพบไฟล์ count.jed การโปรแกรมข้อมูลวงจรลงชิพ CPLD ให้ทำดังต่อไปนี้

1. คลิกที่เครื่องหมาย "+" หน้าคำว่า Generate Programming File
2. ดับเบิ้ลคลิกเมาส์ที่ Configure Device (iMPACT)
3. โปรแกรมจะแสดงหน้าต่างต่าง Configure Devices ให้เลือกอุปกรณ์โปรแกรมเป็นประเภท Boundary-Scan Mode (JTAG) แล้วคลิกปุ่ม Next



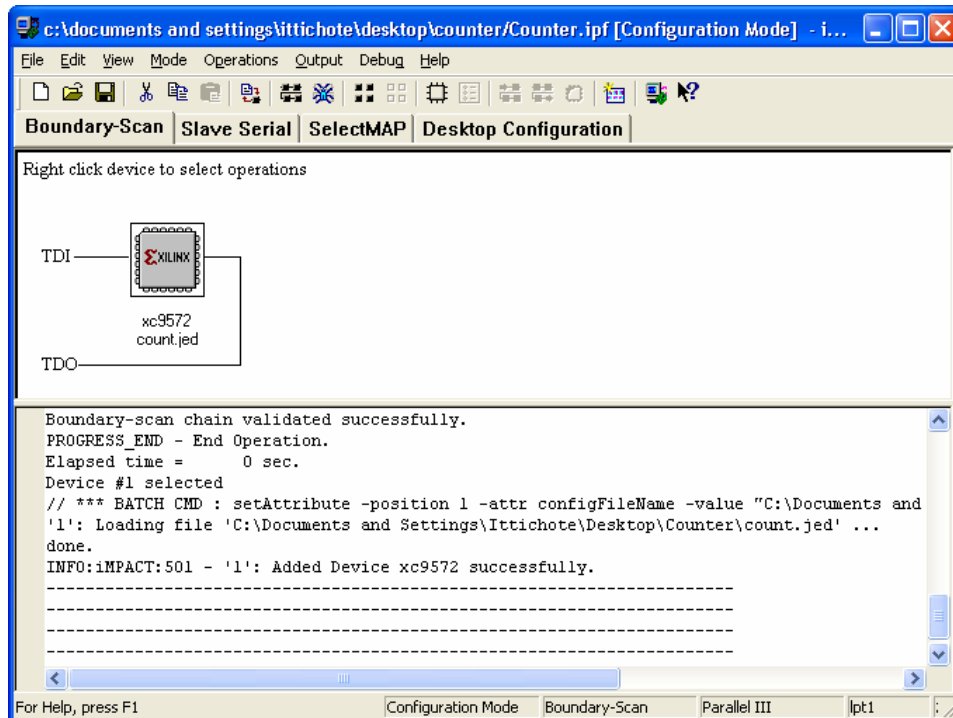
4. โปรแกรมจะแสดงหน้าต่าง Boundary-Scan Mode Selection ให้เลือก Automatically connect to cable and identify Boundary-Scan chain แล้วคลิกปุ่ม Finish



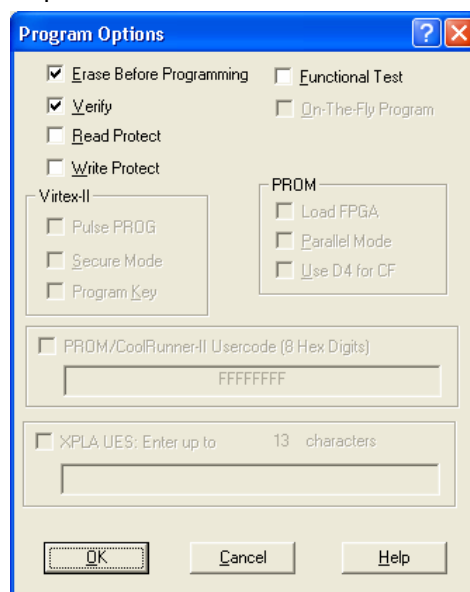
5. โปรแกรมจะทำการตรวจหาชิพที่สาย JTAG เชื่อมต่ออยู่ ถ้าพบชิพโปรแกรมจะบอกโดยแสดงหน้าต่าง Boundary-Scan Chain Contents Summary คลิกปุ่ม OK



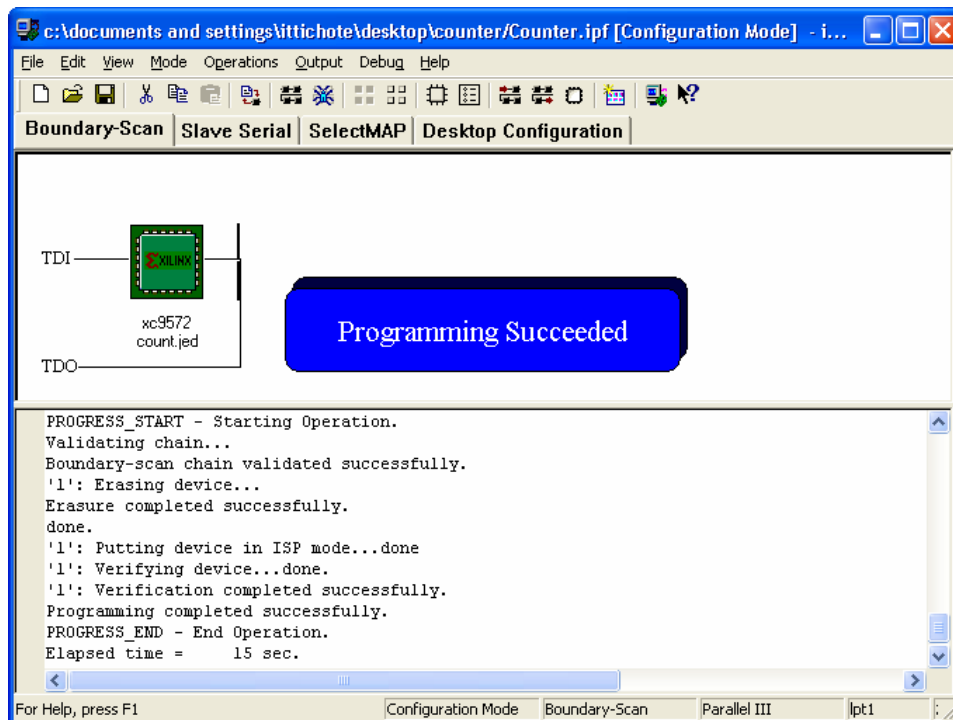
6. เลือกไฟล์ที่โปรแกรมลงบนชิพ ซึ่งในกรณีนี้จะสังเกตว่าเป็น CPLD ดังนั้นจึงต้องใช้ไฟล์ .jed ให้เลือกไฟล์ count.jed แล้วคลิกปุ่ม Open โปรแกรมจะแสดงหน้าต่าง Untitled [Configuration Mode] – iMPACT



7. คลิกขวาที่ตัวไอซี แล้วเลือกคลิกที่คำว่า Program จะแสดงหน้าต่าง Program Options ไม่ต้องเปลี่ยนแปลงตัวเลือกใดๆ คลิกปุ่ม OK



8. เมื่อเริ่มทำการดาวน์โหลด เมื่อดาวน์โหลดเสร็จเรียบร้อยแล้วโปรแกรมจะแสดงข้อความที่หน้าจอว่า Programming Succeeded



ตอนนี้ได้ทำการ โปรแกรมข้อมูลวงจรลงบนชิพเรียบร้อยแล้ว ให้เราทดลองการทำงานของโปรแกรมโดยการกดปุ่ม PB1 แทนสัญญาณนาฬิกา แล้วปุ่ม PB2 แทนทิศทาง จะพบว่ามันมั่วๆ ไม่เป็นตามที่เราคาดไว้ ขอให้ลองทำการแก้ไขโปรแกรมจนใช้ได้ถูกต้อง