

#### การทดลองที่ 4 การใช้งานโปรแกรม ISE Xilinx WebPACK

##### วัตถุประสงค์

1. เพื่อให้นักศึกษาติดตั้งและฝึกการใช้งานโปรแกรม ISE Xilinx WebPACK
2. เพื่อให้นักศึกษาเข้าใจขั้นตอนการออกแบบวงจร Combinational logic โดยใช้ FPGA ได้

หมายเหตุ ให้อ่านเอกสารการทดลองและเอกสารประกอบให้ครบก่อนเริ่มทำการทดลอง

##### บทนำ

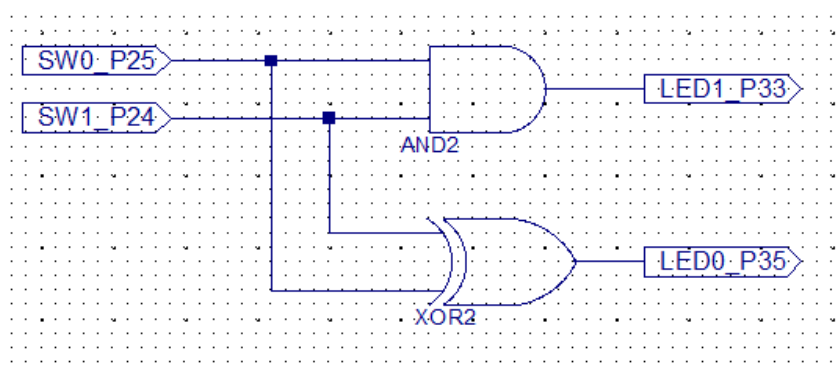
การทดลองนี้แตกต่างจากการทดลองที่ผ่านมา นักศึกษาจะได้รู้จักการออกแบบวงจรด้วยการวาดรูป (Schematic Circuit) และบันทึกโปรแกรมลงในชิป Field Programmable Gate Array (FPGA)

##### โปรแกรม ISE Xilinx WebPACK

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

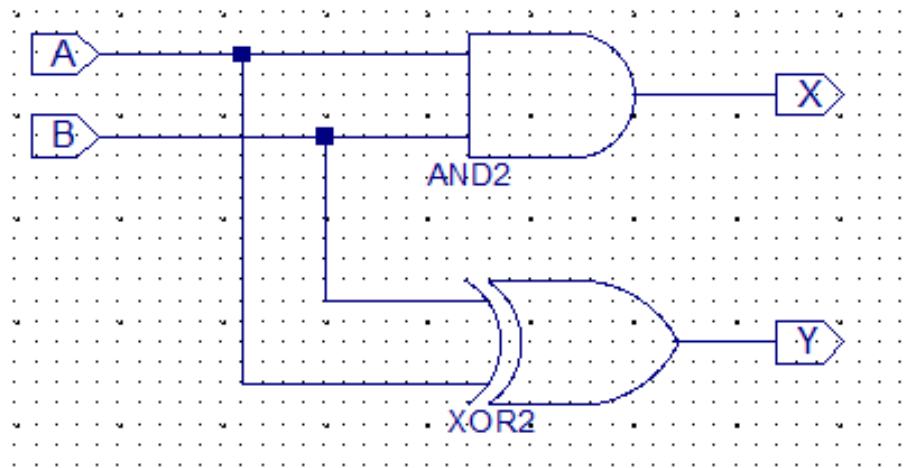
##### การทดลอง

1. ให้นักศึกษาติดตั้งโปรแกรม ISE Xilinx WebPACK และ ทดลองใช้งานโปรแกรม ตามเอกสารการใช้งานโปรแกรม Xilinx จนถึงหน้าที่ 12/61 ส่วนต่อจากนั้นยังไม่ต้องทำเนื่องจากยังไม่มีบอร์ด FPGA
2. เมื่อทำตามข้อที่ 1. เสร็จเรียบร้อยแล้วนักศึกษาจะได้ Schematic ดังรูปที่ 1.1



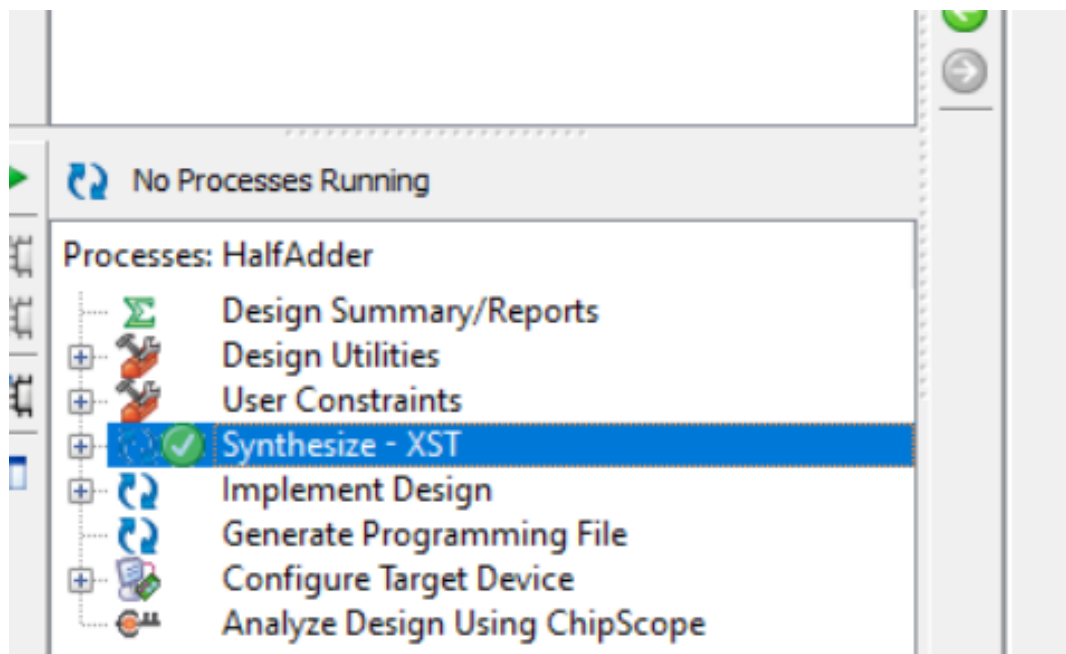
รูปที่ 2.1 แสดง Schematic ที่ได้หลังจากการทำตามขั้นตอนข้อที่ 1.

3. สังเกตว่าชื่อของ Port ต่าง ๆ จะตั้งให้ตรงกับ Port จริง ๆ ที่มีอยู่ใน FPGA แต่เนื่องจากแลปนี้เรายังไม่ได้ใช้บอร์ด FPGA เราจึงจะทำการเปลี่ยนชื่อ Port ใหม่ให้สั้นลงเพื่อให้สะดวกกับการใช้งานในขั้นต่อไปดังรูปที่ 3.1



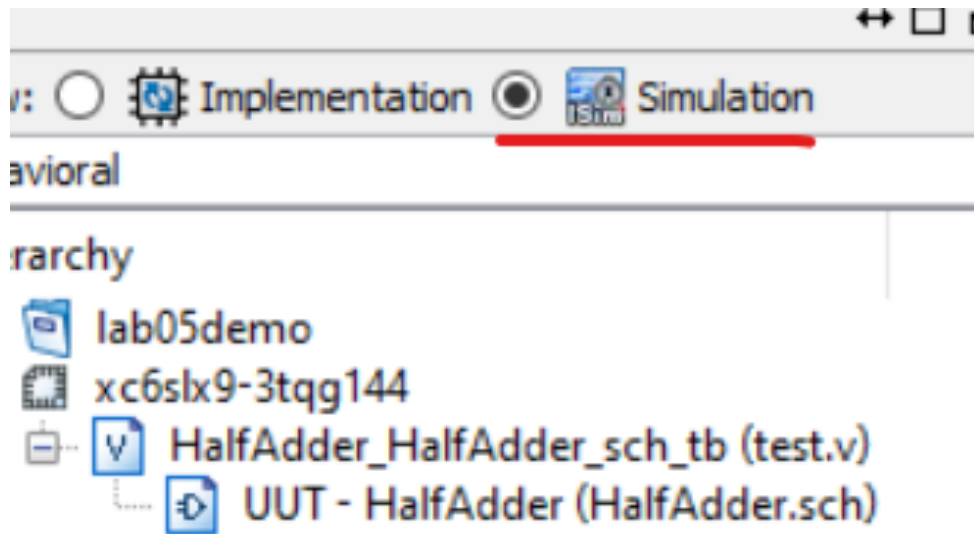
รูปที่ 3.1 แสดง Schematic ที่ได้หลังจากเปลี่ยนชื่อ Port ให้สั้นลง

4. จากนั้น Double-Click คลิกที่ Synthesize – XST เพื่อตรวจสอบการเชื่อมต่อวงจร ดังแสดงในรูปที่ 4.1 และต้องให้แสดงเครื่องหมายถูก หากไม่แสดงให้กลับไปเช็คการเชื่อมต่อวงจรว่ามีจุดไหนที่มีปัญหาให้แก้ไขแล้วกลับมา Double-Click เพื่อ Synthesize ใหม่จนกว่าจะขึ้นเครื่องหมายถูก



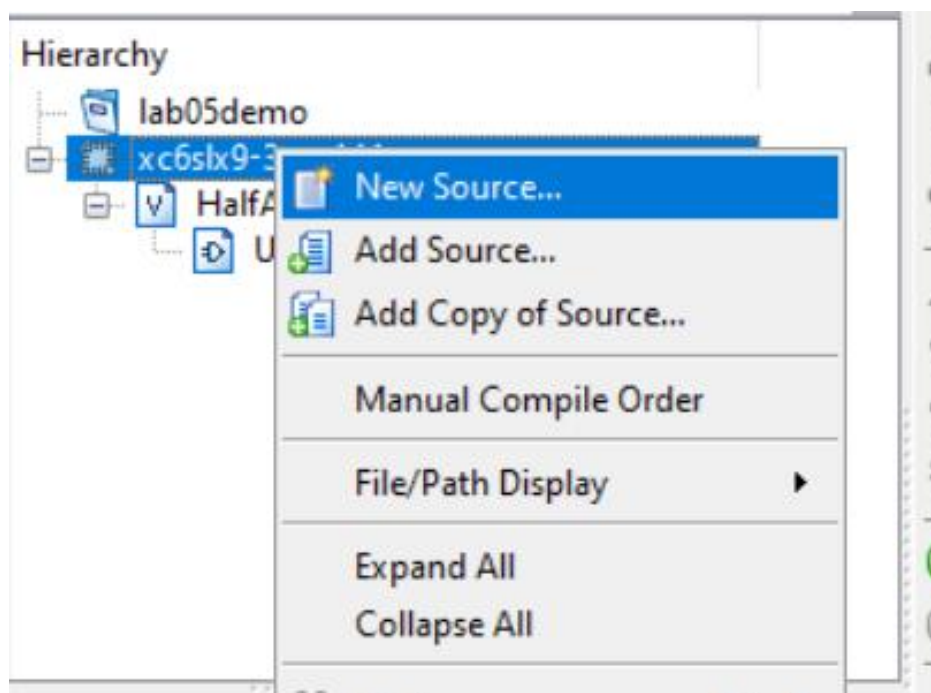
รูปที่ 4.1 แสดง การ Synthesize – XST

5. จากนั้นเราจะทำการทดสอบการทำงานของวงจรในโหมด Simulation โดยให้นักศึกษาลิกละเปลี่ยนโหมดเป็นโหมด Simulation ดังรูปที่ 5.1



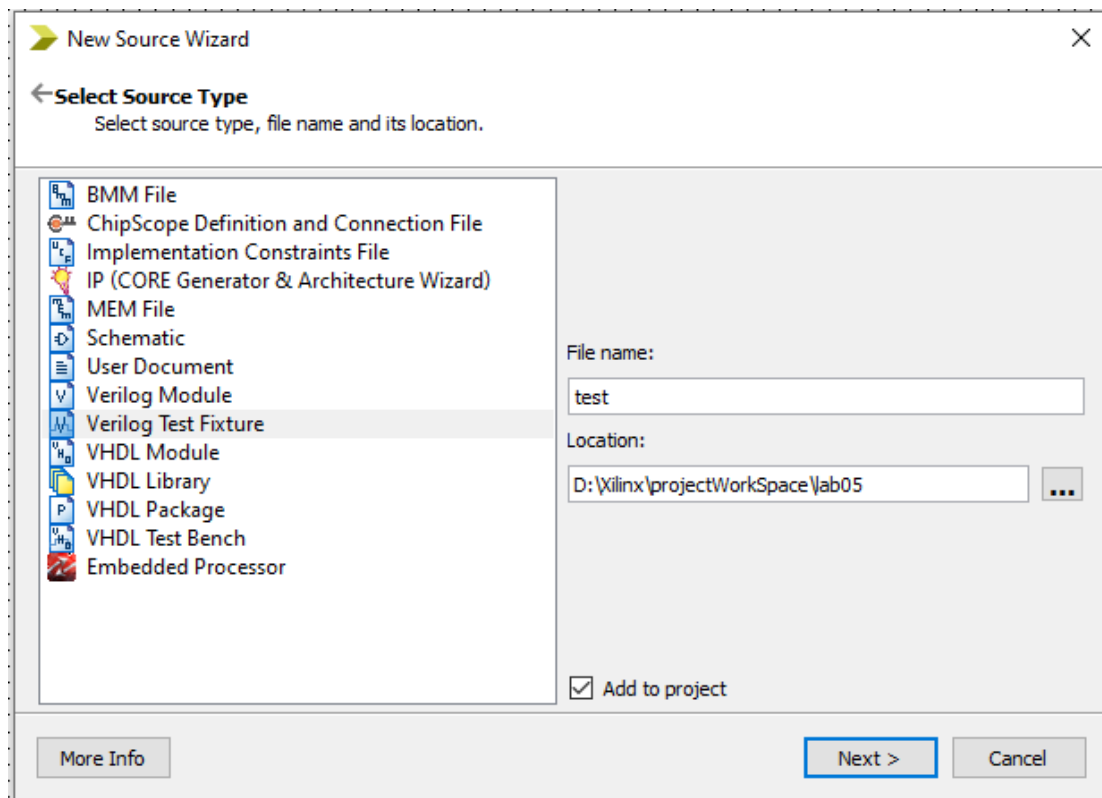
รูปที่ 5.1 แสดง การเปลี่ยนเป็นโหมด Simulation

6. จากนั้นให้นักศึกษาลิกละขวาที่ icon รูปชิปที่ตามด้วยรายละเอียดของชิป FPGA ที่เราเลือกตอนสร้างโปรเจกต์ เช่น xc6slx9-3tqg144 ดังรูปที่ 6.1



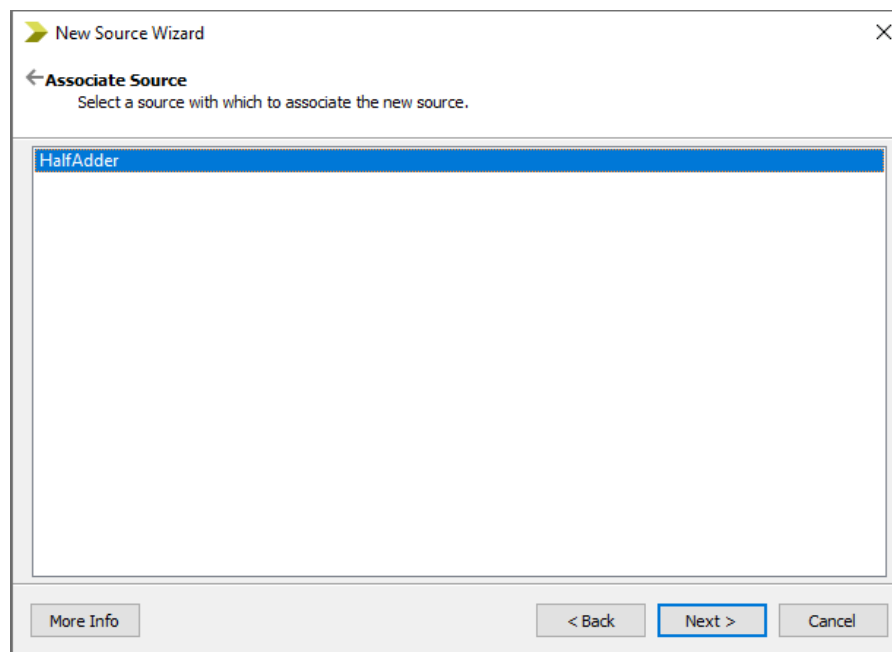
รูปที่ 6.1 แสดง การเพิ่ม Source ใหม่

7. เลือกชนิดไฟล์เป็น Verilog Test Fixture จากนั้นตั้งชื่อไฟล์ แล้วกด Next



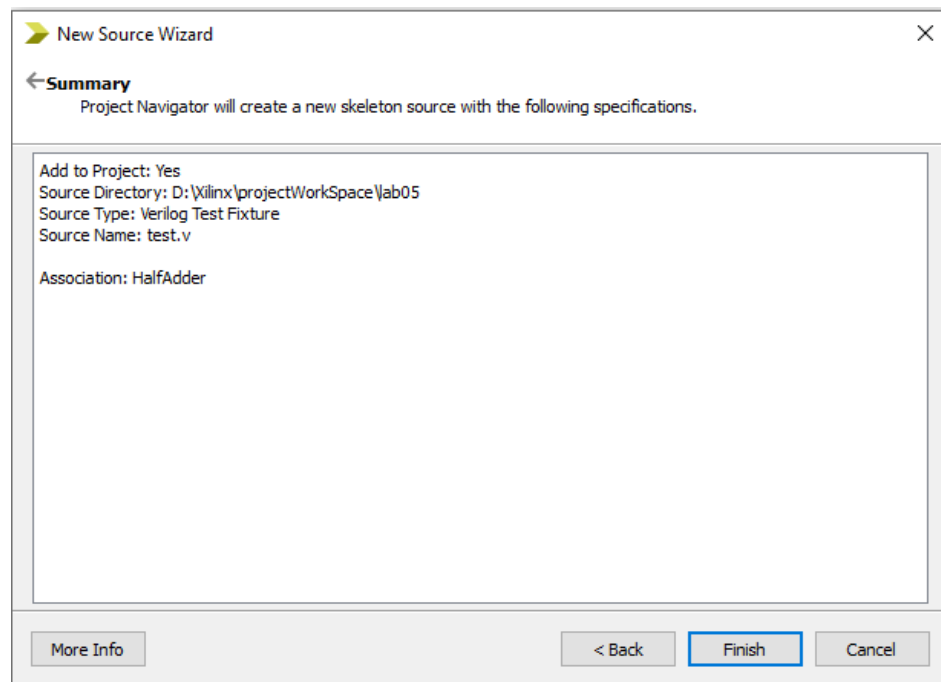
รูปที่ 7.1 แสดงการเลือกชนิดไฟล์ Verilog Test Fixture

8. จากนั้นเลือกไฟล์ Schematic ที่จะทดสอบแล้วกด Next



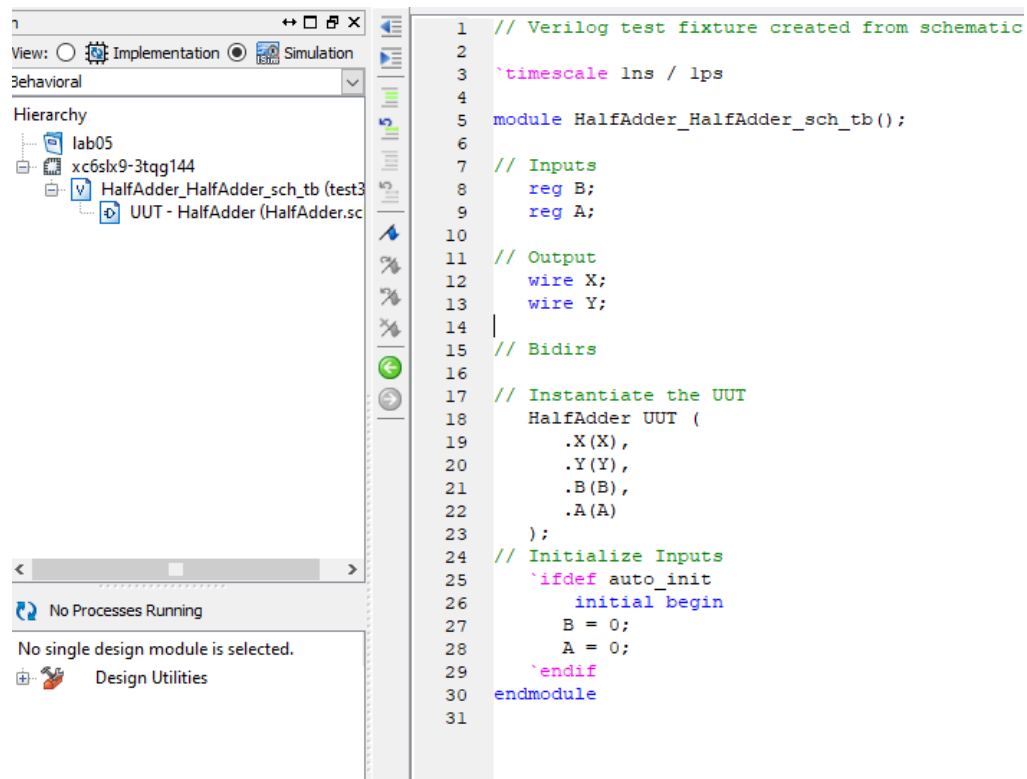
รูปที่ 8.1 แสดงการเลือกไฟล์ Schematic ที่จะทดสอบ

## 9. จากนั้นกด Finish



รูปที่ 9.1 แสดงหน้าต่างการสร้างไฟล์ Verilog Test Fixture เสร็จสิ้น

## 10. จะได้โครงสร้างไฟล์ และ ไฟล์โค้ดภาษา Verilog ดังรูปที่ 10.1



รูปที่ 10.1 โครงสร้างไฟล์ที่ได้ และ ไฟล์โค้ดภาษา Verilog ที่ใช้ในการทดสอบ

11. จากนั้นให้ลบโค้ดในส่วนของบรรทัดที่ 25 – 29 ในโค้ดออก ดังรูปที่ 11.1

```
21     .B(B),  
22     .A(A)  
23 );  
24 // Initialize Inputs  
25 `ifdef auto_init  
26     initial begin  
27         B = 0;  
28         A = 0;  
29     `endif  
30 endmodule  
31
```

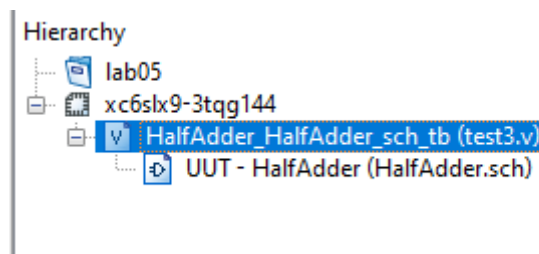
รูปที่ 11.1 แสดงส่วนของโค้ดที่ลบออก

12. จากนั้นใส่โค้ดต่อไปนี้แทน ดังรูปที่ 12.1

```
// Initialize Inputs  
initial begin // เริ่มขอบเขตของการทำงาน  
    A = 0; // กำหนด port A มีค่าเท่ากับ 0  
    B = 0; // กำหนด port B มีค่าเท่ากับ 0  
    #5; // หน่วงเวลา 5 หน่วย(ns)  
    A = 0; // กำหนด port A มีค่าเท่ากับ 0  
    B = 1; // กำหนด port B มีค่าเท่ากับ 1  
    #5; // หน่วงเวลา 5 หน่วย(ns)  
    A = 1; // กำหนด port A มีค่าเท่ากับ 1  
    B = 0; // กำหนด port B มีค่าเท่ากับ 0  
    #5; // หน่วงเวลา 5 หน่วย(ns)  
    A = 1; // กำหนด port A มีค่าเท่ากับ 1  
    B = 1; // กำหนด port B มีค่าเท่ากับ 1  
    #5; // หน่วงเวลา 5 หน่วย(ns)  
    $finish; //จุดสิ้นสุดการทำงาน  
end // ปิดขอบเขตการทำงาน  
initial begin // เริ่มขอบเขตของการทำงาน  
    $monitor("A=%d,B=%d : X=%d,Y=%d",A,B,X,Y); // แสดงผลค่า IO ออกทาง console  
end // ปิดขอบเขตการทำงาน  
endmodule
```

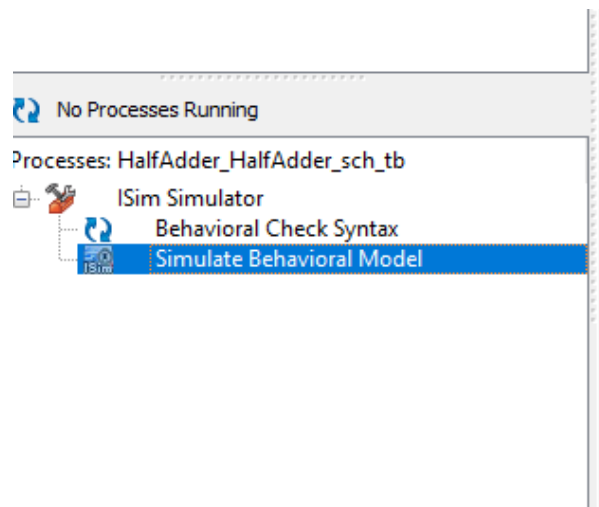
รูปที่ 12.1 แสดงส่วนของโค้ดภาษา Verilog ที่ใช้ทดสอบการทำงานของวงจร

13. จากนั้นให้นักศึกษาคลิกซ้าย 1 ครั้งเลือกไฟล์ทดสอบที่มีนามสกุล .v ดังรูปที่ 13.1



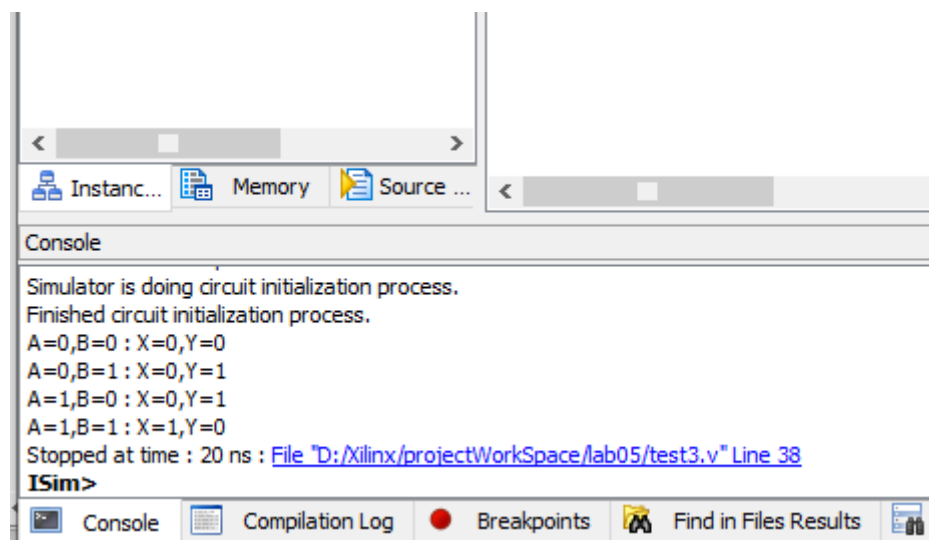
รูปที่ 13.1 การเลือกไฟล์ทดสอบ

14. จากนั้นให้ Double-Click ที่ Simulate Behavioral Model เพื่อเริ่มการทดสอบวงจร ดังรูปที่ 14.1

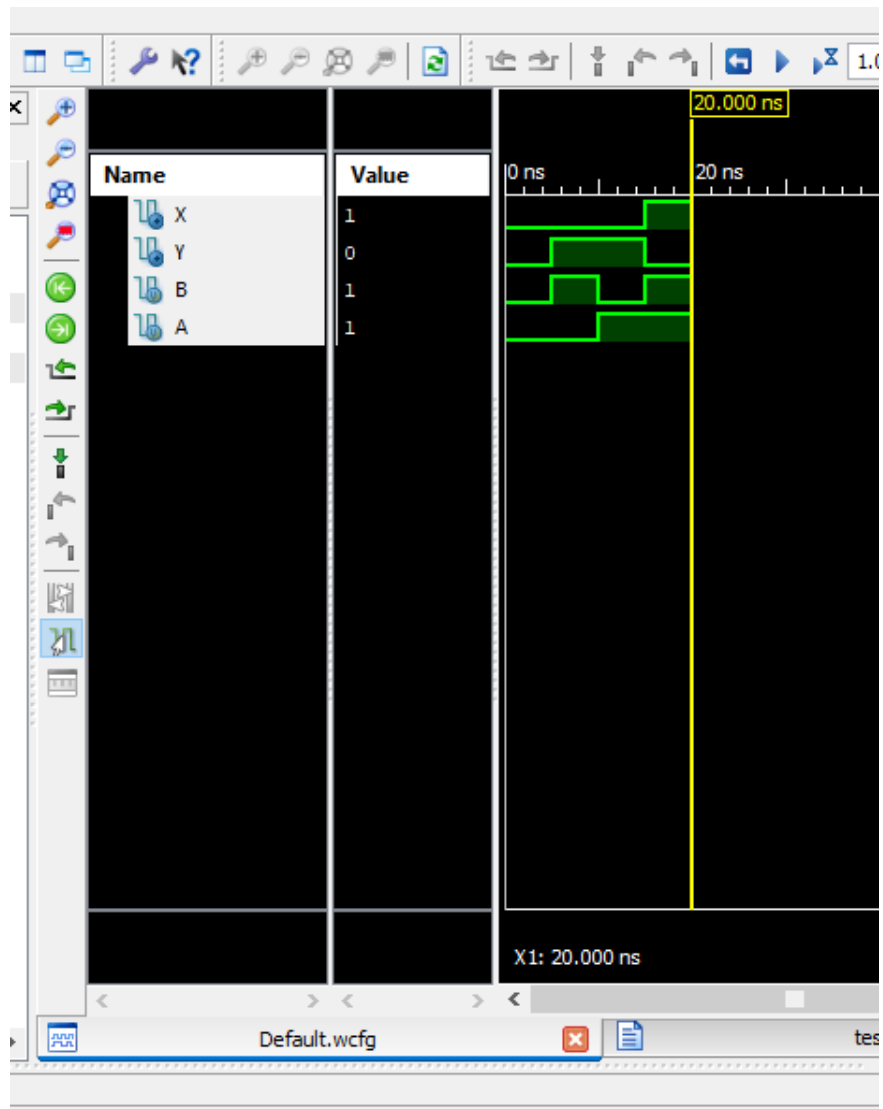


รูปที่ 14.1 แสดงการเริ่มการทดสอบวงจร

15. สุดท้ายจะได้ผลลัพธ์ดังแสดงในรูปที่ 15.1 ซึ่งจะมีข้อความที่แสดงค่า Input Output ของวงจรที่หน้าต่าง Console หรือ สามารถกดดูค่าของ Input และ Output ในรูปของสัญญาณ Digital ได้ที่ไฟล์ Default.wcfg ดังแสดงในรูปที่ 15.2



รูปที่ 15.1 แสดงผลลัพธ์ของการทดสอบการทำงานของวงจร



รูปที่ 15.2 แสดงผลลัพธ์ของการทดสอบการทำงานของวงจรในรูปแบบของสัญญาณ Digital

16. สามารถดูวิดีโอประกอบการอธิบายได้ดังลิงค์ต่อไปนี้

<https://youtu.be/85FeH-dLszM>





17. ให้นักศึกษาออกแบบพร้อมแสดงวิธีการออกแบบวงจรสร้างสัญญาณเตือนในรถยนต์ซึ่งต่ออยู่กับ Buzzer เพื่อเตือนผู้ขับขี่รถยนต์ในกรณีต่างๆ ดังต่อไปนี้

17.1 เมื่อประตูผู้ขับขี่เปิดและมีการติดเครื่องยนต์

17.2 เมื่อมีผู้ขับขี่นั่งในเบาะผู้ขับขี่แต่ไม่มีการคาดเข็มขัดนิรภัยและมีการติดเครื่องยนต์

17.3 เมื่อเปิดไฟหน้าแต่ไม่มีการติดเครื่องยนต์

17.4 เมื่อไม่มีผู้ขับขี่นั่งในเบาะผู้ขับขี่แต่มีการคาดเข็มขัดนิรภัยและประตูผู้ขับขี่ปิด

โดยกำหนดให้เซนเซอร์แต่ละจุดส่งสัญญาณดิจิตอลมาดังนี้

- H = Headlights คือไฟหน้า จะเป็น 1 เมื่อเปิดไฟ และเป็น 0 เมื่อดับไฟ
- S = Seat คือเบาะผู้ขับขี่ จะเป็น 1 เมื่อมีคนนั่ง และเป็น 0 เมื่อไม่มีคนนั่ง
- B = Seatbelt คือเข็มขัดนิรภัย จะเป็น 1 เมื่อคาด และเป็น 0 เมื่อไม่ได้คาด
- D = Door คือประตูผู้ขับขี่ จะเป็น 1 เมื่อปิด และเป็น 0 เมื่อเปิด
- E = Engine คือเครื่องยนต์ จะเป็น 1 เมื่อติดเครื่องยนต์ และเป็น 0 เมื่อดับเครื่องยนต์

และกำหนดให้รับอินพุต H S B D E ทางสวิตช์โยกบนเครื่อง Logic Trainer และต่อเอาต์พุตเข้ากับ Buzzer บน Logic trainer

18. เมื่อต่อวงจรเสร็จแล้วให้ทำการทดสอบการทำงานของวงจรด้วยการจำลองการทำงานในโหมด Simulation พร้อมทั้งแสดงค่า Input Output ออกทาง Console โดยทดสอบ Input ให้ครบตามเงื่อนไขทั้ง 4 ข้อ ในข้อที่ 17.

## ใบตรวจการทดลองที่ 4

วัน/เดือน/ปี \_\_\_\_\_ ☐ กลุ่ม 101 ☐ กลุ่ม 102 ☐ กลุ่ม 103 ☐ กลุ่ม 153

รหัสนักศึกษา \_\_\_\_\_ ชื่อ-นามสกุล \_\_\_\_\_

การตรวจการทดลอง

☐ บันทึกคะแนนแล้ว

การทดลองข้อ 4      ลายเซ็นอาจารย์ \_\_\_\_\_