

01204224 ปฏิบัติการวงจรตรรก

ปฏิบัติการที่ 10 การออกแบบฮาร์ดแวร์ด้วย Verilog HDL เบื้องต้น

กำหนดส่งในชั่วโมงหรือก่อนเที่ยงคืนวันพุธที่ 2 พฤษภาคม (23:59 น.)

กติกา:

- ส่งงานผ่าน Google's Classroom รหัสวิชาคือ 5dkjt2u
- ในการส่งงานให้รวมไฟล์ทั้งหมดเหล่านี้ไว้ในไฟล์เดียวชื่อว่า lab10_StudentID.zip แล้วส่งมาที่ Google's Classroom ของวิชานี้ โดย Student ID คือรหัสนิสิต
- ขาดเรียน ส่งไม่ตรงเวลา หรือมีการลางานกันได้ 0 คะแนน

ให้นิสิตดาวน์โหลดและลงโปรแกรม Icarus Verilog (iverilog) และ gtkwave จากไฟล์ที่แนบมาให้ สำหรับนิสิตที่ใช้วินโดวส์ 64 บิต

นิสิตที่ใช้เครื่อง Mac อาจติดตั้ง iverilog และ gtkwave จากคำสั่งต่อไปนี้

```
brew install icarus-verilog  
brew install Caskroom/cask/gtkwave
```

ต่อไปนี้จะกล่าวถึง iverilog ที่รันบนวินโดวส์ เมื่อ install iverilog เรียบร้อยแล้ว นิสิตจะต้องไปตั้งค่า path โดยเพิ่ม C:\iverilog\bin และ C:\iverilog\gtkwave\bin ลงไปในค่า path เดิมเพื่อที่จะเรียกใช้ iverilog ได้ที่ directory ที่นิสิตเซตไว้สำหรับงานนี้ ในการตั้งค่า path ข้างต้น ไปที่ Control Panel -> System -> Advanced system settings -> Environment Variables แล้วเลือก edit PATH เมื่อเสร็จขั้นตอนการ install และตั้งค่า path แล้ว

งานที่ต้องนำส่ง: ให้เขียนโมดูล addsub และเซพในไฟล์ addsub.v โดยใช้ Verilog HDL ที่สามารถทำการบวกและลบ integers ขนาด 32 บิต โดยห้ามนิสิตใช้เครื่องหมาย + หรือ - ในโมดูลที่เขียนขึ้น โดยโมดูลนี้มี format ดังนี้

```
addsub(a, b, result, control);
```

ค่า a b และ result เป็นค่า 32 bits และ control เป็นสัญญาณควบคุมโดยให้ 1 เลือกเป็นการบวกและ 0 เลือกเป็นการลบ เสร็จแล้วให้นำไฟล์ addsub_tb.v มาทดสอบการทำงานของ addsub โมดูล

นิสิตสามารถดูแนวทางการเขียนโปรแกรมได้ใน 2 หน้าสุดท้ายของเอกสารนี้ เทคนิคการออกแบบเหมือนปฏิบัติการที่นิสิตได้เคยทำมาแล้ว ส่วนการเขียน Verilog HDL ให้นิสิตดูจากตัวอย่างและดูแนวทางการเขียนจากคู่มือออนไลน์ทั่วไป

เพื่อให้เข้าใจขั้นตอนการใช้งานโปรแกรมทั้ง iverilog และ gtkwave อาจารย์ได้ให้ไฟล์ addsub_toy.v ไว้เพื่อทดลอง โดยให้นิสิต เริ่มต้นจากการใช้คำสั่งต่อไปนี้บนวินโดวส์ command prompt

```
iverilog -o addsub addsub_toy.v addsub_tb.v
```

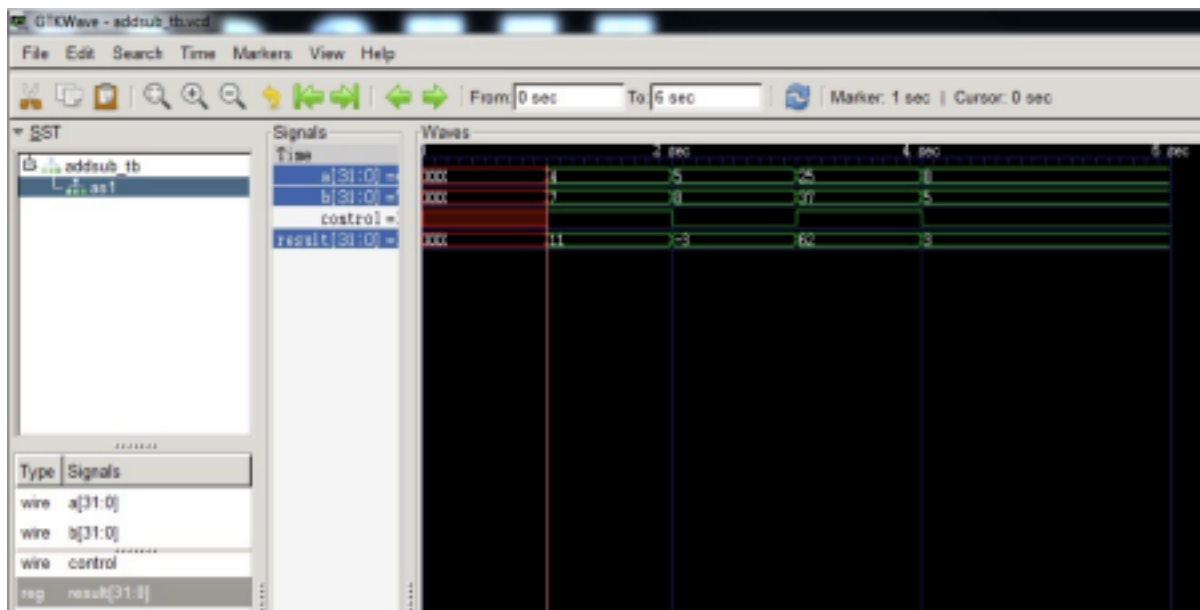
จากนั้นรันด้วย

```
vvp addsub
```

และ

```
gtkwave addsub_tb.vcd
```

ผลลัพธ์จากการรันจะเป็นไปตามภาพที่แสดงด้านล่าง



เมื่อเข้าใจขั้นตอนและการทำงานของโปรแกรมแล้ว ให้กลับไปเขียนโค้ด verilog ลงในไฟล์ addsub.v แล้วทดสอบวงจรที่สร้างขึ้นด้วย กรณีทดสอบที่เขียนใน addsub_tb.v นอกเหนือจากกรณีทดสอบที่ให้มาแล้ว ให้นิสิตเพิ่มกรณีทดสอบในไฟล์ทดสอบนี้อีก 10 กรณี โดยเลือกกรณีที่มีความหมายสำหรับการทดสอบ ตรวจสอบผลลัพธ์ที่แสดงออกที่ command prompt และที่แสดงเป็น กราฟฟิคใน gtkwave

ให้นักศึกษาลงภาพหน้าจอที่ gtkwave แสดงผลในกรณีทดสอบที่สร้างใหม่ 10 กรณีเพื่อนำส่ง เมื่อทำปฏิบัติการเสร็จสิ้นแล้ว ให้รวมไฟล์ที่เกี่ยวข้องทั้งหมดไว้ใน directory lab10_StudentID

- ไฟล์ addsub.v
- ไฟล์ addsub_tb.v ที่มีกรณีทดสอบที่เขียนเพิ่มเติม
- ไฟล์ result.png ที่เก็บภาพหน้าจอแสดงผลการทำงานของวงจร addsub

แล้ว zip มาในรูป lab9_StudentID.zip ไฟล์ แล้วส่งมาที่ Google Classroom ของวิชา

```
module FullAdder1(op1,op2,cin,cout,result);  
    // fill me in  
endmodule  
  
module FullAdder2(op1,op2,cin,cout,result);  
    // fill me in  
endmodule  
  
module FullAdder4(op1,op2,cin,cout,result);  
    // fill me in  
endmodule  
  
module FullAdder8(op1,op2,cin,cout,result);  
    // fill me in  
endmodule  
  
module FullAdder16(op1,op2,cin,cout,result);  
    // fill me in  
endmodule  
  
module FullAdder32(op1,op2,cin,cout,result);  
    // fill me in  
endmodule
```

```

module addsub(a,b,result,control);
    input[31:0] a,b;
    input control;
    output[31:0] result;
    wire[31:0] result;
    reg[31:0] tmp;
    reg cin;
    wire cout;

    FullAdder32 FA32(a, tmp, cin, cout, result);

    always@(b or control)
        begin
            if (control == 0) // for subtract case

                // fill me in; set tmp and cin to the correct values

            else // for add case

                // fill me in; set tmp and cin to the correct values

        end
endmodule

```