

Introduction to FPGA Design

วัตถุประสงค์

1. นักศึกษาเข้าใจการออกแบบวงจรเบื้องต้นในอุปกรณ์ที่โปรแกรมได้ ชนิด FPGA
2. นักศึกษาสามารถออกแบบวงจรนับและวงจรเชื่อมต่อกับ 7-Segment ด้วยภาษา Verilog
3. นักศึกษาสามารถสร้างวงจรวงจรรนับและวงจรเชื่อมต่อกับ 7-Segmentเบื้องต้น บน FPGA

ให้นักศึกษาเรียนรู้ ศึกษา ติดตั้ง Software และ อุปกรณ์ที่เกี่ยวข้องดังต่อไปนี้

1. ศึกษาการออกแบบวงจรด้วยภาษา Verilog
2. ติดตั้ง โปรแกรม Intel Quartus II (ไฟล์ “คู่มือการลงโปรแกรม Intel Quartus II.pdf”)
3. ติดตั้ง โปรแกรม USB-Blaster (ไฟล์ “คู่มือการลง USB Blaster.pdf”)
4. ทดลองใช้งานบอร์ด FPGA (ไฟล์ “คู่มือการใช้งาน Intel Quartus II.pdf” และ “คู่มือการใช้งานบอร์ด Altera FPGA Shield-v03.pdf”)

การทดลอง

ตอนที่ 1

- 1.1 ให้นักศึกษานำ Verilog Code เหล่านี้ save ไว้ใน Drive ที่เขียนได้ในเครื่องคอมพิวเตอร์ที่นักศึกษาใช้งาน ตามชื่อไฟล์ที่ระบุไว้ (counter.v)
- 1.2 ให้นักศึกษา แก้ไข เพิ่มเติม Verilog Code ในส่วนของไฟล์ “segment.v” เพื่อให้สามารถแสดงผลออกทาง 7-Segment ที่เป็นเลขฐานสิบ 0-9
- 1.3 ให้นักศึกษา แก้ไข เพิ่มเติม Verilog Code ในส่วนของไฟล์ “top.v” เพื่อให้การเชื่อมต่อสัญญาณต่างๆ ที่เหมาะสมและเข้ากันได้ กับ Block Diagram ในรูปที่ 1.
- 1.4 ให้นักศึกษาทำการสังเคราะห์วงจรและ download ลงสู่บอร์ด FPGA ซึ่งขั้นตอนนี้ ควรศึกษา “คู่มือการใช้งาน Intel Quartus II.pdf” ให้เข้าใจ โดยนักศึกษา ต้องเพิ่มไฟล์ที่ชื่อ top.v เข้าสู่ Project เป็นไฟล์แรก แล้วจึงเพิ่มอีก 2 ไฟล์คือ counter.v และ segment.v เข้าไปที่หลัง เมื่อเพิ่มสำเร็จแล้ว จะเห็นว่าในหน้าต่าง Entity มีโมดูลชื่อ top อยู่บนสุด และมี โมดูลชื่อ counter:u0 กับ segment.u1 เป็นโมดูลย่อย เมื่อสั่ง Run “start compilation” หากผลลัพธ์มี error ก็ต้องกลับไปแก้ไข Verilog code จนกระทั่งได้ข้อความว่า “Full Compilation was successful”
- 1.5 ทำตามขั้นตอนที่อยู่ใน “คู่มือการใช้งาน Intel Quartus II.pdf” เมื่อถึงขั้นตอนกำหนดขาที่จะใช้งาน Pin Planner ให้นักศึกษาใช้งาน 7-Segment หลักขวามือสุด จากทั้งหมด 4 หลัก (รายละเอียดขาที่จะใช้ อยู่ใน “คู่มือการใช้งานบอร์ด Altera FPGA Shield-v03.pdf” หน้าสุดท้าย) ในการทดลองนี้ ให้นักศึกษากำหนดขาสัญญาณ clk เป็นสัญญาณที่รับจากการกดปุ่ม switch “S5” (หมายเหตุ การ

กดปุ่ม S5 แต่ครั้งนั้น ผลการนับค่า อาจจะมีการเพิ่มขึ้นมากกว่า 1 ค่าได้ ซึ่งเป็นความผิดพลาดจากหน้าสัมผัสของโลหะในปุ่มกดนั่นเอง ซึ่งเราจะใช้สัญญาณนาฬิกาแทน ใน Check Point ที่ 2) Check Point # 1 ให้นักศึกษาแสดงผลการทำงานของบอร์ด โดยกดปุ่ม Switch S5 ให้ 7-segment แสดงเป็นเลขต่างๆ (บางช่วงจะแสดงเฉพาะ segment g เป็นเครื่องหมาย “-“ เพราะมีการนับค่าเกิน 9) และเมื่อโยก Dip-Switch ที่ต่อกับขา rst ค่าตัวเลขบน 7-segment นั้นต้องแสดงเป็นเลข “ศูนย์” พร้อมสำหรับการเริ่มนับใหม่เมื่อโยก Dip-Switch กลับไปที่เดิม

ชื่อไฟล์ “counter.v”

```
module counter (clk,rst,Q);
input clk,rst;
output [3:0] Q;
reg [3:0] tmp;

always @(posedge clk or negedge rst)
begin
    if(rst == 0)
        tmp <= 4'b0000;
    else if (clk == 1)
        tmp <= tmp + 1'b1;
    end

assign Q = tmp;
endmodule
```

ชื่อไฟล์ “segment.v”

```
module segment (bcd,seg,digit);
input [3:0] bcd;
output [7:0] seg;
output digit;
reg [7:0] tmp;

always @*
case (bcd)
4'b0000: tmp = 8'b0011_1111; // display 0
```

```

4'b0001: tmp = .....; // display 1 ต้องแก้ไขเพิ่มเติม
4'b0010: tmp = .....; // display 2 ต้องแก้ไขเพิ่มเติม
4'b0011: tmp = .....; // display 3 ต้องแก้ไขเพิ่มเติม
4'b0100: tmp = .....; // display 4 ต้องแก้ไขเพิ่มเติม
4'b0101: tmp = .....; // display 5 ต้องแก้ไขเพิ่มเติม
4'b0110: tmp = .....; // display 6 ต้องแก้ไขเพิ่มเติม
4'b0111: tmp = .....; // display 7 ต้องแก้ไขเพิ่มเติม
4'b1000: tmp = .....; // display 8 ต้องแก้ไขเพิ่มเติม
4'b1001: tmp = .....; // display 9 ต้องแก้ไขเพิ่มเติม
default: tmp = 8'b0100_0000;
endcase
assign seg = tmp;
assign digit = 1;
endmodule

```

ชื่อไฟล์ “top.v”

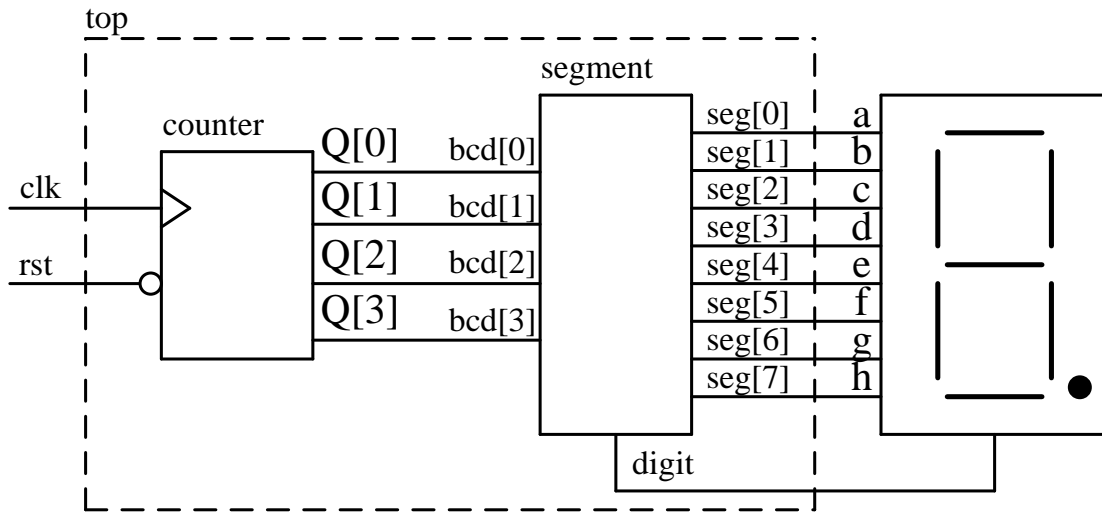
```

module top (clk,rst,seg,digit);
input clk,rst;
output [7:0] seg;
output digit;
wire [3:0] q_bcd;

counter    u0 (.....); // ต้องแก้ไขเพิ่มเติม เพื่อเชื่อมสัญญาณ ตามรูปที่ 1
segment    u1 (.....); // ต้องแก้ไขเพิ่มเติม เพื่อเชื่อมสัญญาณ ตามรูปที่ 1

endmodule

```



รูป 1 Block Diagram วงจรนับ BCD ออก 7-Segment จำนวน 1 หลัก

ตอนที่ 2 ใช้สัญญาณนาฬิกาแทนการกดปุ่ม Switch S5

2.1 ให้นักศึกษา เพิ่มโมดูลหารสัญญาณนาฬิกาจากบอร์ด FPGA ที่มีความถี่ 50 MHz (คือขาสัญญาณ PIN_17) ให้มีความถี่เหลือประมาณ 1 Hz ในที่นี้คือ ไฟล์ clk_divider_to_1hz.v โดยเพิ่มไฟล์เข้าไปใน Project ปัจจุบันที่ทำงานอยู่

2.2 เพิ่มการเชื่อมต่อสัญญาณ ในโมดูล top (เป็นการเชื่อมต่อสัญญาณนาฬิกาของ FPGA เข้ากับ โมดูล clk_divider_to_1hz โดยเอาต์พุตของโมดูลนี้ จะป้อนสัญญาณนาฬิกาความถี่ประมาณ 1 Hz ให้กับโมดูล counter) โดยให้นักศึกษา พิมพ์ Verilog code เพิ่ม 2 บรรทัดที่เป็นอักษรตัวหน้า และแก้ไขการเชื่อมต่อสัญญาณนาฬิกา ใน top.v แล้ว save

2.3 แก้ไข Verilog Code เพื่อให้วงจรสามารถนับค่าและต้องแสดงเฉพาะ เลข 0-9 เท่านั้น ห้ามแสดงค่าที่นับเกิน 9 กล่าวคือ การนับต้องนับจาก 0 ถึง 9 เท่านั้น (ห้ามแสดง “-” ที่ segment g นั่นเอง) โดยยังคงมี Switch ที่ทำหน้าที่ Reset วงจรนับ ให้กลับไปเริ่มที่ค่าศูนย์

Check Point # 2 ให้นักศึกษาแสดงผลการทำงานดังต่อไปนี้ แสดงผลการทำงานบนบอร์ด FPGA ที่มีการนับค่าอัตโนมัติ จาก 0 ถึง 9 ห้ามแสดงค่า “-” กล่าวคือเมื่อนับ 9 แล้วการนับครั้งถัดไปต้องกลับไปนับ 0 และเมื่อมีการโยก Dip-Switch ที่ต่อกับขา rst ค่าตัวเลขบน 7-segment นั้นต้องแสดงเป็นเลข “ศูนย์” พร้อมสำหรับการเริ่มนับใหม่เมื่อโยก Dip-Switch กลับไปที่เดิม

ชื่อไฟล์ “clk_divider_to_1hz.v”

```
module clk_divider_to_1hz(clk, f_out);
input clk;
output f_out;
parameter max_count2 = 50000000/2;
reg [21:0] counter;
always @(posedge clk)
    if(counter==0)
        counter <= max_count2 -1;
    else
        counter <= counter-1;
reg f_out;
always @(posedge clk)
    if(counter==0)
        f_out <= ~f_out;
endmodule
```

ชื่อไฟล์ “top.v”

```
module top (clk,rst,seg,digit);
input clk,rst;
output [7:0] seg;
output digit;
wire [3:0] q_bcd;
wire clk_1Hz;
counter    u0 (.....); // ต้องแก้ไขเพิ่มเติม จาก Check Point ที่ 1
segment    u1 (.....); // ไม่ต้องแก้ไขจาก Check Point ที่ 1
clk_divider_to_1hz u2 (clk,clk_1Hz);
endmodule
```

ตอนที่ 3

3.1 แก้ไข Verilog Code เพื่อให้วงจรสามารถเลือกได้ว่า จะนับขึ้น หรือนับลง โดยอ่านค่าจาก Switch อีก 1 ตัว และค่าที่นับได้ต้องแสดงเฉพาะ เลข 0-9 เท่านั้น โดยยังคงมี Switch ที่ทำหน้าที่ Reset วงจรนับ ให้กลับไปเริ่มที่ค่าศูนย์

3.2 ให้นักศึกษาทำการสังเคราะห์วงจรและ download ลงสู่บอร์ด FPGA
Check Point # 3 ให้นักศึกษาแสดงดังผลลัพธ์ต่อไปนี้ ผลการทำงานบนบอร์ด FPGA ที่สามารถเลือกให้เห็นถึงการนับขึ้น การนับลง ในเลข 0 ถึง 9 และ การ Reset

ข้อกำหนดในการส่งรายงานการทดลอง

1. การส่งไฟล์รายงานการทดลอง จะต้องเป็นไฟล์รูปแบบ PDF เท่านั้น
2. การส่งไฟล์รายงาน ให้ส่งใน Link ที่กำหนดให้ตรงกับวันที่ลง Lab เท่านั้น หากส่งผิด Link อาจารย์ อาจจะได้ไม่ตรวจเพราะสับสนหรือหารายงานไม่พบ คะแนนรายงานจะได้ ศูนย์ คะแนน
3. การส่งไฟล์รายงานการทดลอง จะต้องส่งภายใน 1 สัปดาห์ หลังจากวันที่ลง Lab หากเกินกำหนดเวลา จะ ถูกหักคะแนนการส่งสาย นาทีละ 1 คะแนน แต่หักสูงสุดไม่เกินคะแนนเต็มของ คะแนนรายงาน