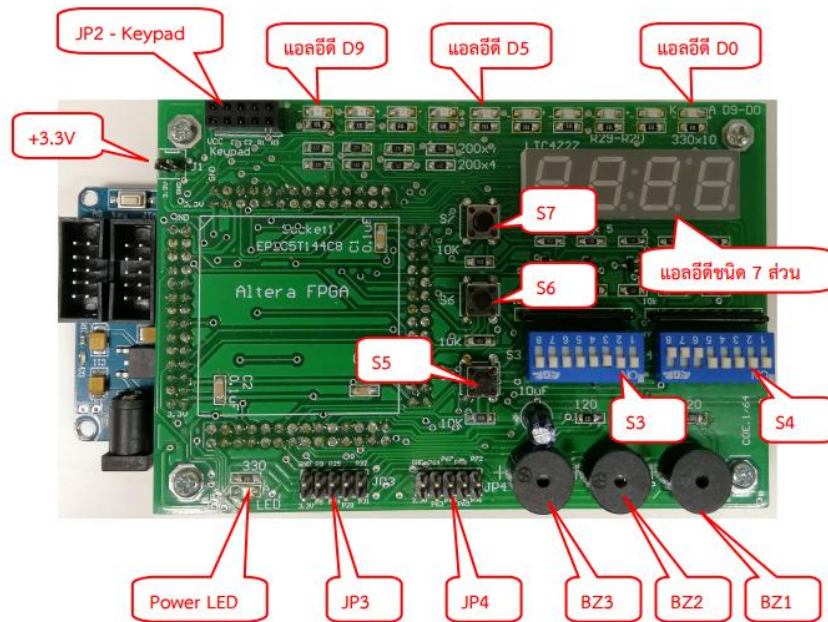


รายงานผลการทดลอง : 3HA01: Introduction to FPGA design



รูปที่ 4 องค์ประกอบของบอร์ด FPGA Shield

รูปที่1 Layout อุปกรณ์ย่อยของบอร์ด

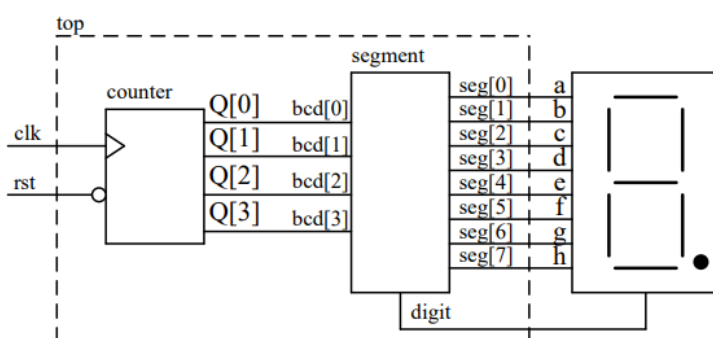
ผลการทดลอง Check Point #1

Node Name	Direction	Location
in clk	Input	PIN_125
out digit	Output	PIN_99
in rst	Input	PIN_60
out seg[7]	Output	PIN_121
out seg[6]	Output	PIN_120
out seg[5]	Output	PIN_119
out seg[4]	Output	PIN_118
out seg[3]	Output	PIN_115
out seg[2]	Output	PIN_114
out seg[1]	Output	PIN_113
out seg[0]	Output	PIN_112

PIN	คือ
125	Switch5 [Clock] input
99	Output ที่ Digit #4
60	S3 ซ้ายสุด
Seg[7:0]	แสดงผล Segmentย่อย

รูปที่2 Pin Planner

*ดูแต่ละPinคืออะไรได้ที่ Document : คู่มือการใช้งานบอร์ด Altera FPGA Shield-v03.doc

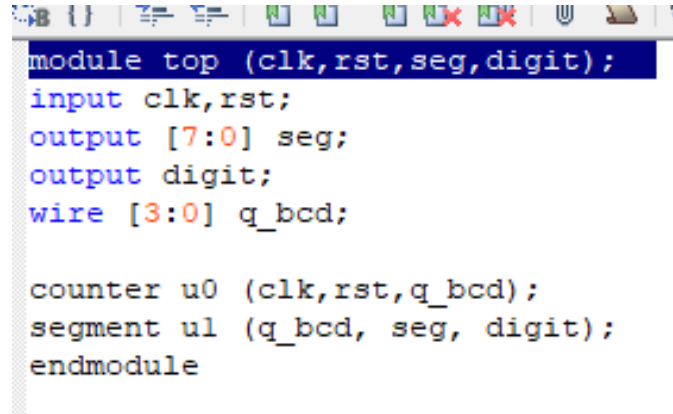


รูป 1 Block Diagram วงจรนับ BCD ออก 7-Segment จำนวน 1 หลัก

รูปที่ 3 Block Diagram

การออกแบบวงจร

1. Module หลัก Top.v



```

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

counter u0 (clk,rst,q_bcd);
segment u1 (q_bcd, seg, digit);
endmodule

```

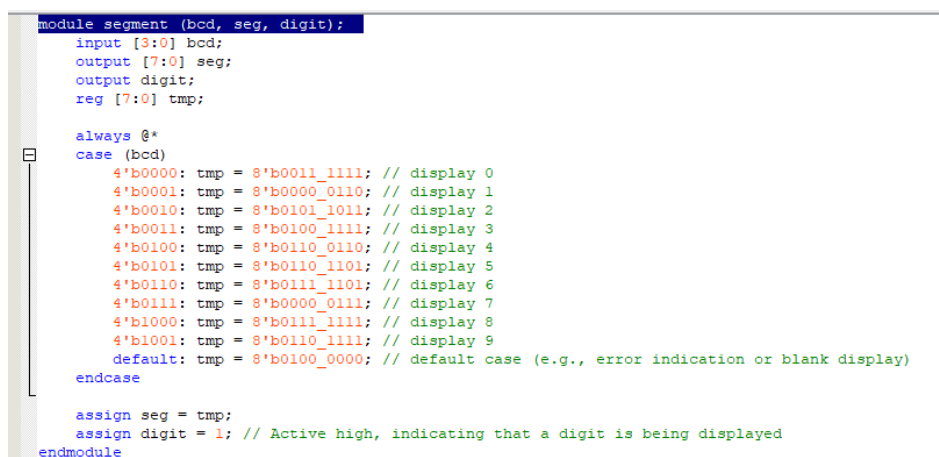
รูปที่ 4 Top.v

- counter u0 (clk,rst,q_bcd); // เชื่อมสัญญาณตาม Block Diagram
- segment u1 (q_bcd,seg,digit); // เชื่อมสัญญาณตาม Block Diagram

2. Module Counter.v : ไม่ต้องทำการแก้ไข

3. Module Segment.v : Segment เป็น Segment แบบ common Cathode

จ่าย 1 เพื่อให้ Segment ติด



```

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 = 8'b0000_0110; // display 1
4'b0010: tmp = 8'b0101_1011; // display 2
4'b0011: tmp = 8'b0100_1111; // display 3
4'b0100: tmp = 8'b0110_0110; // display 4
4'b0101: tmp = 8'b0110_1101; // display 5
4'b0110: tmp = 8'b0111_1101; // display 6
4'b0111: tmp = 8'b0000_0111; // display 7
4'b1000: tmp = 8'b0111_1111; // display 8
4'b1001: tmp = 8'b0110_1111; // display 9
default: tmp = 8'b0100_0000; // default case (e.g., error indication or blank display)
endcase

assign seg = tmp;
assign digit = 1; // Active high, indicating that a digit is being displayed
endmodule

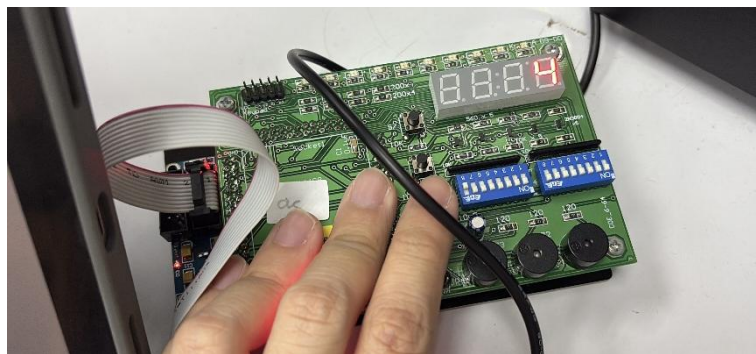
```

รูปที่ 5 Segment.V

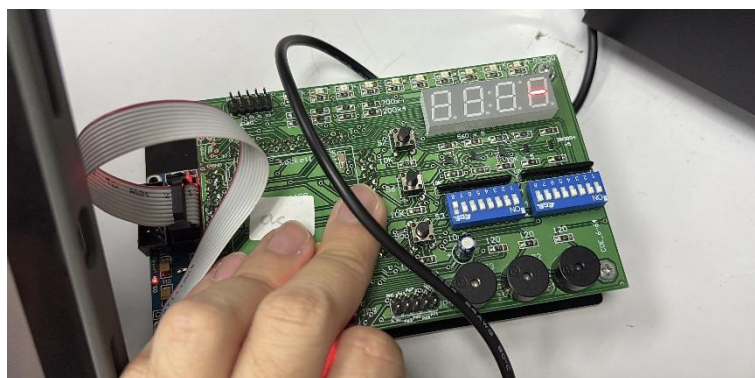
Counter และ 7-Segment Display บน FPGA Board

1. การทดสอบ Counter

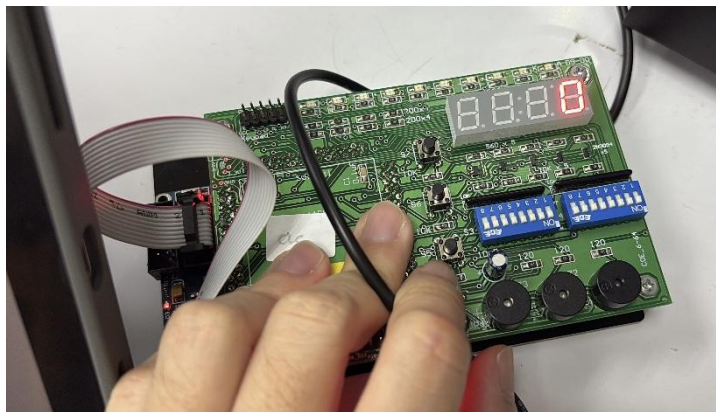
- ขั้นตอนการทดลอง: กดปุ่ม Switch S5 เพื่อให้สัญญาณ clock เข้าสู่วงจร counter
- ผลที่ได้: 7-segment display แสดงตัวเลขเพิ่มขึ้นตามลำดับ ดังนี้
 - เริ่มต้นแสดง "0"
 - กดปุ่ม S5 ครั้งที่ 1 แสดง "1"
 - กดปุ่ม S5 ครั้งที่ 2 แสดง "2"
 - กดปุ่ม S5 ครั้งที่ 3 แสดง "3"
 - ดำเนินการต่อไปจนถึง "9"
 - เมื่อกดต่อไปจะแสดงเฉพาะ segment g ("-") เนื่องจากเกิดการนับเกินค่า 9



รูปที่ 6 แสดงผล เลข4 เมื่อ กด SW5 เป็นครั้งที่ 5 เพื่อเพื่อ Counter



รูปที่ 7 แสดงผล "-" เมื่อ กด SW5 เป็นครั้งที่ 11 เพื่อเพื่อ Counter



รูปที่ 8 แสดงผล “-” เมื่อ กด SW5 เป็นครั้งที่ 5 เพื่อเฟล Counter

2. การทดสอบ Reset Function

- ขั้นตอนการทดลอง:
 1. ให้ counter นับไปยังค่าใดค่าหนึ่ง (เช่น "5" หรือ "8")
 2. โยก DipSwitch ที่เชื่อมต่อกับขา rst
- ผลที่ได้:
 - 7-segment display แสดง "0" ทันที
 - วงจรพร้อมสำหรับการเริ่มนับใหม่
 - เมื่อโยก DipSwitch กลับไปที่ตำแหน่งเดิม สามารถกดปุ่ม S5 เพื่อนับต่อได้ปกติ

3. การทดสอบ 7-Segment Display Pattern

จากโค้ดในส่วน segment module พบว่า:

- แต่ละตัวเลข 0-9 จะแสดงผลตาม pattern ที่กำหนด
- ตัวอย่างการแสดงผล:
 - "0": แสดง segments a,b,c,d,e,f (ไม่แสดง g)
 - "1": แสดง segments b,c
 - "2": แสดง segments a,b,g,e,d
 - "9": แสดง segments a,b,c,d,f,g

สรุปผลการทดลอง

1. **Counter Function:** ทำงานได้อย่างถูกต้อง นับเพิ่มทีละ 1 ทุกครั้งที่กดปุ่ม S5
2. **7-Segment Display:** แสดงผลตัวเลข 0-9 ได้อย่างชัดเจน และแสดง "-" เมื่อนับเกิน
3. **Reset Function:** ทำงานได้อย่างสมบูรณ์ สามารถรีเซ็ตค่าเป็น "0" และเริ่มนับใหม่ได้

Check Point #2: Automatic Counter with 1Hz Clock

Node Name	Direction	Location
in clk	Input	PIN_17
out digit	Output	PIN_99
in rst	Input	PIN_60
out seg[7]	Output	PIN_121
out seg[6]	Output	PIN_120
out seg[5]	Output	PIN_119
out seg[4]	Output	PIN_118
out seg[3]	Output	PIN_115
out seg[2]	Output	PIN_114
out seg[1]	Output	PIN_113
out seg[0]	Output	PIN_112

PIN	คือ
17	50MHz Clock จากบอร์ด
99	Output ที่ Digit #4
60	S3 ซ้ายสุด Reset

รูปที่ 9 Pin Planner

การทำงานของ Clock Divider

- รับสัญญาณนาฬิกา 50 MHz จาก PIN_17
- หาค่าความถี่ลงเหลือประมาณ 1 Hz โดยใช้ parameter $\text{max_count2} = 50000000/2$
- ใช้ 22-bit counter สำหรับนับ
- สร้าง toggle output เมื่อ counter ถึง 0

ผลการทดลอง Check Point #2

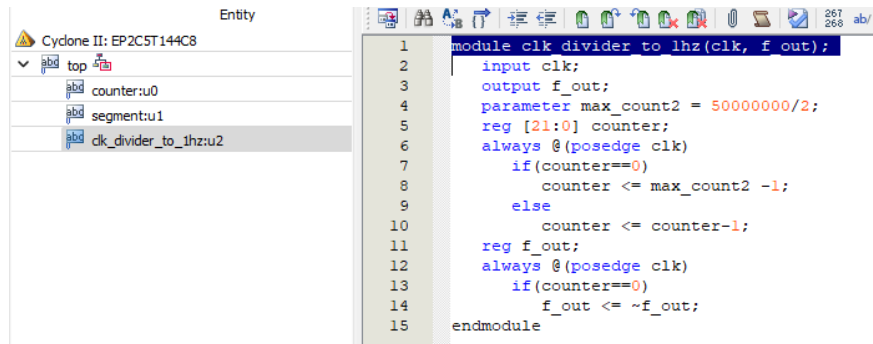
การแสดงผลที่ปรับปรุงแล้ว

- การนับอัตโนมัติ: 7-Segment แสดงตัวเลขเปลี่ยนอัตโนมัติทุกๆ 1 วินาที
- ช่วงการนับ: นับเฉพาะ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 เท่านั้น
- การวนรอบ: เมื่อนับถึง 9 แล้วจะกลับไปนับ 0 ทันที (ไม่แสดงเครื่องหมาย "-")
- ไม่ตัว Switch Bouncing: เนื่องจากใช้สัญญาณนาฬิกาที่เสถียร นับครบทุกเลข

การทำงานของ Reset Function

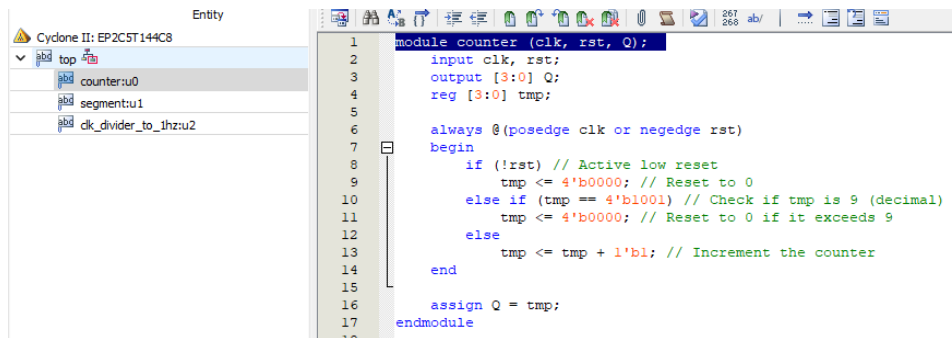
- ฟังก์ชัน Reset ยังคงทำงานได้: เมื่อโยก DIP Switch ที่ต่อกับขา rst
- Reset เป็น 0: 7-Segment จะแสดงเลข 0 ทันที
- กลับมานับต่อได้: เมื่อปล่อย reset วงจรจะเริ่มนับใหม่จาก 0

1. เพิ่ม Clock Divider Module



รูปที่10 Code Clock_divider_1Hz.v

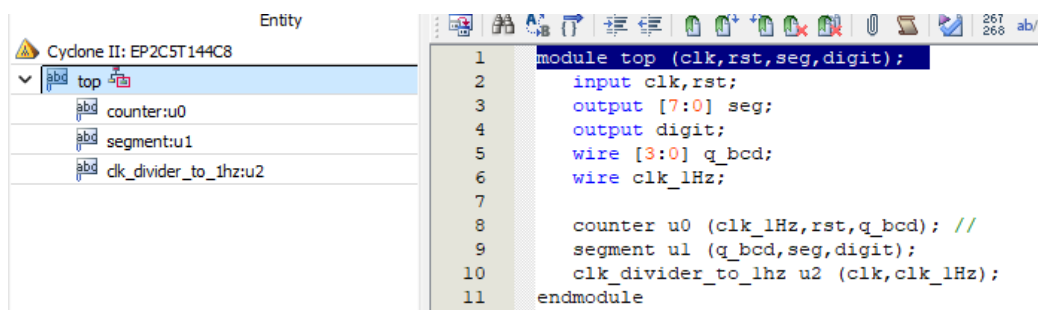
2. ปรับปรุง Counter Module : เพิ่มการตรวจ เมื่อถึง 9 ให้รีเซ็ตที่ 0



รูปที่11 Code Counter.v

3. ปรับปรุง Top Module เพิ่ม

- counter u0 (clk_1Hz,rst,q_bcd); // ใช้ clk_1Hz แทน clk
- clk_divider_to_1hz u2 (clk,clk_1Hz);



รูปที่12 Code Top.v

- Vdo Link ผลการทดลอง Checkpoint #2 : <https://youtu.be/cdxlHFyhgvs>

Check Point #3: Up/Down Counter with Mode Control

	Node Name	Direction	Location
in	clk	Input	PIN_17
out	digit	Output	PIN_99
in	mode	Input	PIN_51
in	rst	Input	PIN_60
out	seg[7]	Output	PIN_121
out	seg[6]	Output	PIN_120
out	seg[5]	Output	PIN_119
out	seg[4]	Output	PIN_118
out	seg[3]	Output	PIN_115
out	seg[2]	Output	PIN_114
out	seg[1]	Output	PIN_113
out	seg[0]	Output	PIN_112

PIN	คือ
17	50MHz Clock จากบอร์ด
99	Output ที่ Digit #4
51	S3 ขวสุด Mode Change
60	S3 ซ้ายสุด Reset

รูปที่13 Pin Planner

ผลการทดลอง Check Point #3**การทดสอบ Count Up Mode**

- ตั้ง DIP switch ที่ต่อกับ PIN_51 เป็น logic high (1)
- ผลลัพธ์: 7-Segment นับขึ้น 0→1→2→3→4→5→6→7→8→9→0→...
- ความเร็ว: เปลี่ยนทุกๆ 1 วินาที (ตาม clock divider)

การทดสอบ Count Down Mode

- ตั้ง DIP switch ที่ต่อกับ PIN_51 เป็น logic low (0)
- ผลลัพธ์: 7-Segment นับลง 9→8→7→6→5→4→3→2→1→0→9→...
- ความเร็ว: เปลี่ยนทุกๆ 1 วินาที (เหมือนกับ count up)

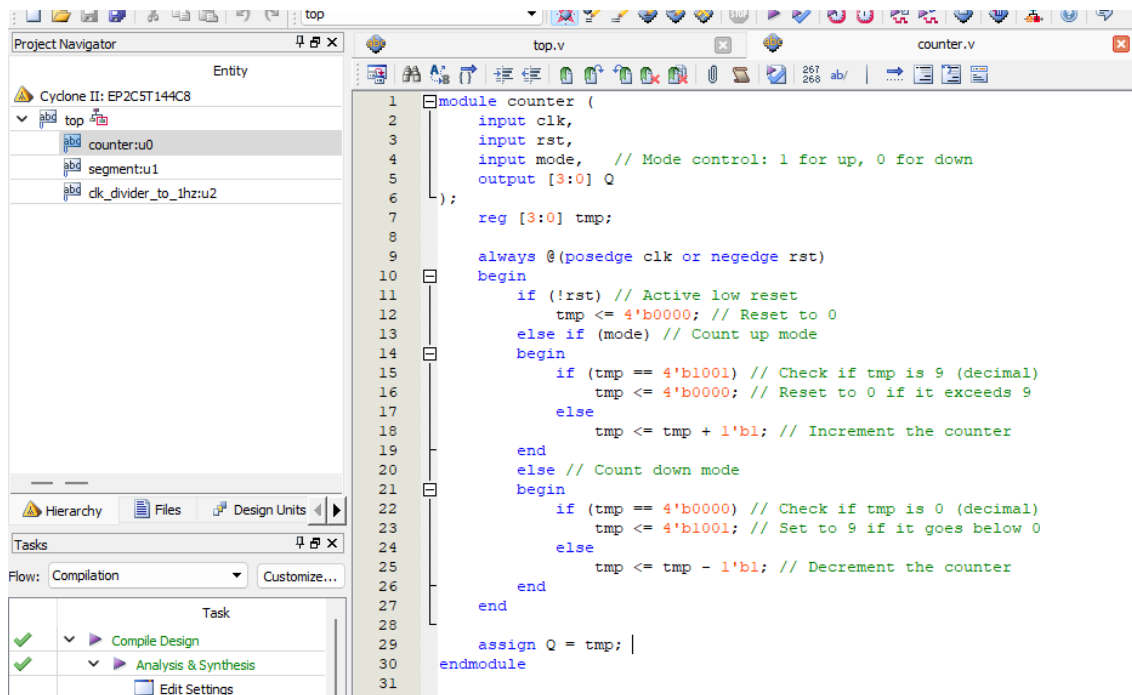
การทดสอบ Reset Function

- Reset ใน Count Up Mode: รีเซ็ตไป 0 แล้วนับขึ้นต่อ
- Reset ใน Count Down Mode: รีเซ็ตไป 0 แล้วนับลงไป 9, 8, 7...

การทดสอบการเปลี่ยน Mode ระหว่างการทำงาน

- เปลี่ยนจาก Up เป็น Down: เปลี่ยนทิศทางการนับทันทีจากตัวเลขปัจจุบัน
- เปลี่ยนจาก Down เป็น Up: เปลี่ยนทิศทางการนับทันทีจากตัวเลขปัจจุบัน

1. ปรับปรุง Counter Module : เพิ่ม mode เพื่อกำหนด นับขึ้น / นับลง



รูปที่14 Code counter

2. ปรับปรุง Top Module : เพิ่ม Mode control input + mode input

```

1 module top (clk,rst,seg,digit,mode);
2     input clk,rst,mode;
3     output [7:0] seg;
4     output digit;
5     wire [3:0] q_bcd;
6     wire clk_1Hz;
7
8     counter u0 (clk_1Hz,rst,mode,q_bcd);
9     segment u1 (q_bcd,seg,digit);
10    clk_divider_to_1hz u2 (clk,clk_1Hz);
11 endmodule

```

รูปที่15 Code counter

3. Segment Module : ไม่ต้องทำการแก้ไข

4. Clock Divider Module : ไม่ต้องทำการแก้ไข

Vdo Link ผลการทดลอง Checkpoint #2 : <https://youtu.be/5XNxqK7RQKA>