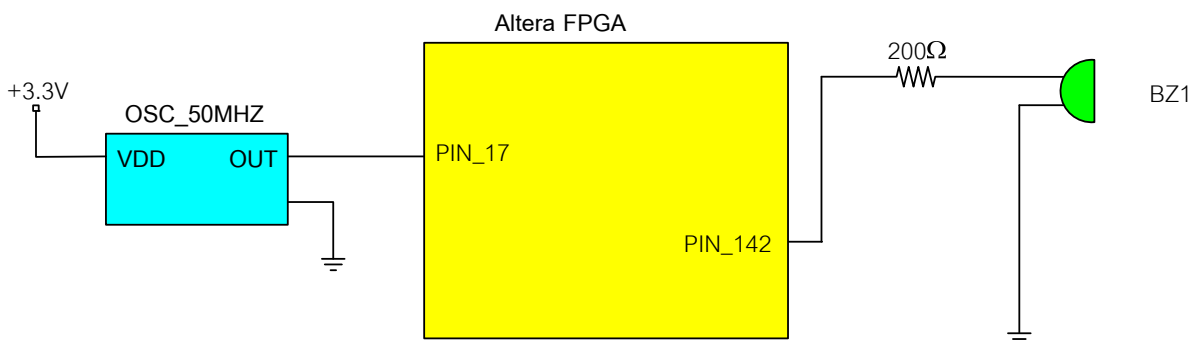


3HB04 : Simple Music Box using FPGA ปีการศึกษา 2567

1. แนะนำ

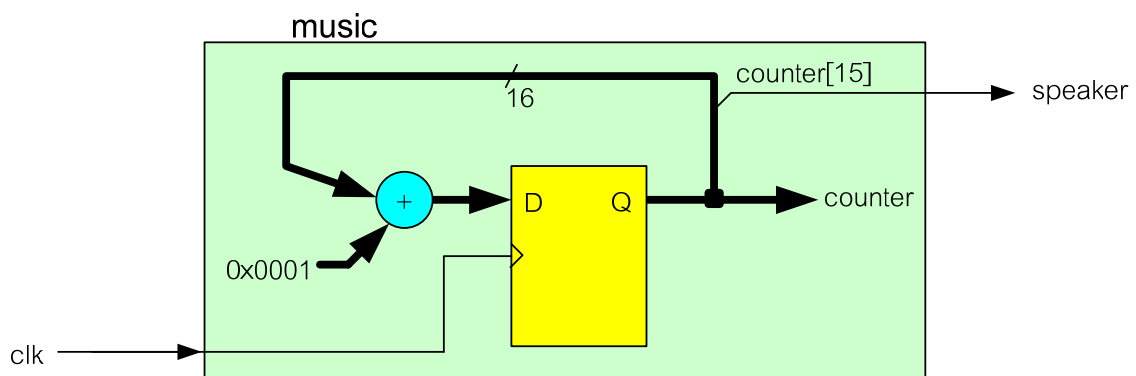
ใน Lab นี้เป็นการสอนเพื่อให้ทราบว่า จะใช้งาน FPGA เล่นเสียงหรือดนตรีแบบง่ายได้อย่างไร โดยเป็นการประยุกต์ใช้งานวงจรสร้างคลื่น จากรูปที่ 1 บอร์ด FPGA จะมีตัวกำเนิดสัญญาณนาฬิกา (Oscillator) อยู่ภายใน ความถี่ 50 MHz ตัวกำเนิดสัญญาณนาฬิกานี้จะให้คลื่นความถี่ที่คงที่ให้แก่ FPGA โดยนักศึกษาจะต้องทำการเขียนโปรแกรมให้ตัว FPGA ทำการหารความถี่นั้นก่อนที่จะจัดส่งออกทางขาสัญญาณเอาต์พุตที่ต่ออยู่กับ Buzzer ผ่านตัวต้านทานขนาด 200 โอห์ม การปรับเปลี่ยนความถี่ของขาเอาต์พุตช่วยให้ FPGA สามารถสร้างเสียงที่แตกต่างกันได้



รูปที่ 1 บอร์ด FPGA กับการสร้างเสียงด้วยวิธีการเปลี่ยนความถี่อย่างง่าย

2. วิธีการสร้างเสียงอย่างง่ายออกทางลำโพง

ในการสร้างเสียงอย่างง่ายจะใช้วิธีการสร้างวงจรนับขนาด 16 บิต ดังวงจรในรูปที่ 2 ซึ่งสามารถสร้างได้โดยใช้ภาษา Verilog ดังโปรแกรมในรูปที่ 3 ซึ่งจะเป็นวงจรนับทำหน้าที่รับสัญญาณนาฬิกาความถี่ 50 MHz เข้ามา หากพิจารณาที่บิตล่างสุดที่ขา counter[0] จะ toggle บิตระหว่างค่าตรรกะสูงและตรรกะต่ำได้เป็นค่าความถี่ 25 MHz ส่วนบิตถัดมาคือบิต counter[1] จะ toggle บิตได้ความถี่ 12.5 MHz ซึ่งจะเห็นว่าบิตที่สูงขึ้นมา 1 ตำแหน่งจะทำการหารความถี่เพิ่มขึ้นตำแหน่งละ 2 ดังนั้น ที่บิตสูงสุด (MSB) ของ counter[15] จะ toggle ที่ความถี่เท่ากับ $50 \times 10^6 / 65536 \approx 763 \text{ Hz}$



รูปที่ 2 โมดูลสำหรับการหารค่าความถี่โดยใช้หลักการของวงจรนับ

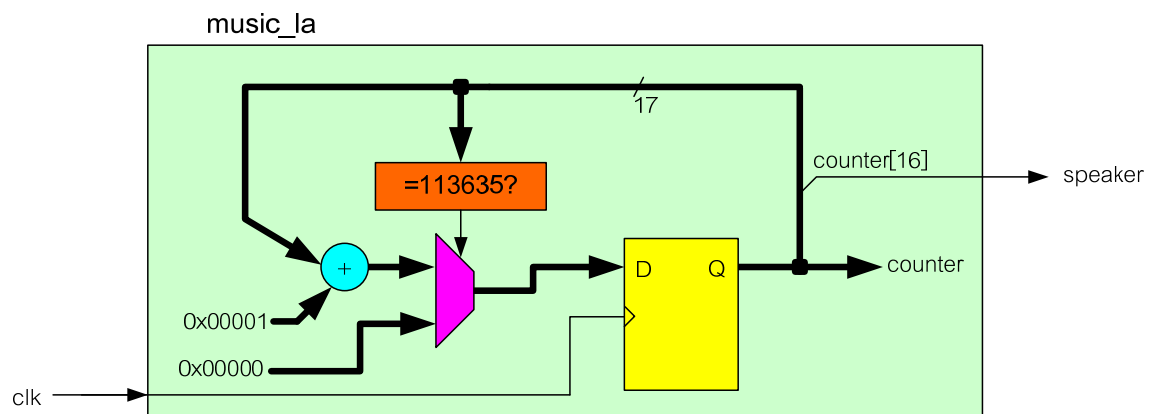
```
module music(clk, speaker);
input clk;
output speaker;
// Binary counter, 16-bits wide
reg [15:0] counter;
always @(posedge clk) counter <= counter+1;

// Use the highest bit of the counter (MSB) to drive the speaker
assign speaker = counter[15];
endmodule
```

รูปที่ 3 โปรแกรมสร้างเสียงความถี่ 763 Hz ออกทางลำโพง

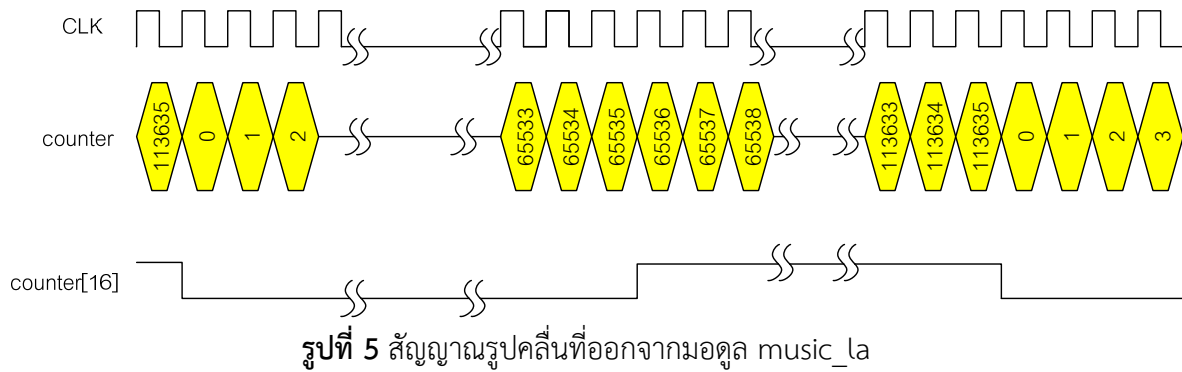
3. การสร้างโน้ตดนตรีอย่างง่าย

โปรแกรมตัวอย่างในรูปที่ 4 เป็นการสร้างเสียงโน้ต “La” ออกทางลำโพงโดยใช้วงจรนับขนาด 17 บิต ซึ่งรับสัญญาณความถี่อินพุต 50 MHz นำมาหารด้วยค่า 113,636 ซึ่งจะได้ค่าความถี่เท่ากับ 440.0014 Hz ซึ่งเป็นค่าความถี่ใกล้เคียงกับเสียงโน้ตตัว “La” ซึ่งมีค่าความถี่เท่ากับ 440 Hz โดยการสร้างเสียงโน้ตใช้วิธีการต่อลำโพงกับขา counter[16] ซึ่งเป็นบิตสูงสุดของวงจรนับ



รูปที่ 4 โมดูลสำหรับใช้ในการสร้างความถี่เสียงโน้ต "La" โดยใช้หลักการของวงจรนับ

จากรูปที่ 4 จะเห็นว่าโมดูลสร้างเสียงโน้ต "La" มีการนำวงจรเปรียบเทียบ (Comparator) มาใช้ในการตรวจสอบว่าค่าที่นับเท่ากับ 113,635 หรือไม่ หากพบว่าเท่าจะสั่งให้มัลติเพล็กซ์เลือกค่า 0x000000 ส่งให้ฟลิปฟล็อป แต่หากไม่เท่าจะสั่งให้มัลติเพล็กซ์ส่งผลลัพธ์จากวงจรบวกไปให้กับฟลิปฟล็อปแทน ดังนั้นวงจรนับนี้จะมีการนับ 0-113635 เท่านั้น (รวมมีการนับเท่ากับ 113636 ค่า) หากมีการเชื่อมต่อสัญญาณที่บิตสูงสุด (บิต 16) ออกสู่ลำโพงจะส่งผลให้ได้ค่าความถี่ 440 เฮิรตซ์ออก ดังแสดงในรูปที่ 5



รูปที่ 5 แสดงสัญญาณรูปคลื่นที่ออกมาจากขา counter หมายเลข 16 จะเห็นว่าสัญญาณนี้มีค่าตรรกะต่ำเมื่อวงจรทำการนับตั้งแต่ค่า 0-65,535 และสัญญาณนี้จะเป็นค่าตรรกะสูงเมื่อวงจรนับตัวนี้ทำการนับตั้งแต่ค่า 65,536-113,635 เราสามารถนำวงจร music_la ในรูปที่ 4 มาสร้างด้วยภาษา Verilog ได้ดังรูปที่ 6

```
module music_la(clk, speaker);
input clk;
output speaker;

reg [16:0] counter;
always @(posedge clk) if(counter==113635) counter <= 0; else counter <= counter+1;

assign speaker = counter[16];
endmodule
```

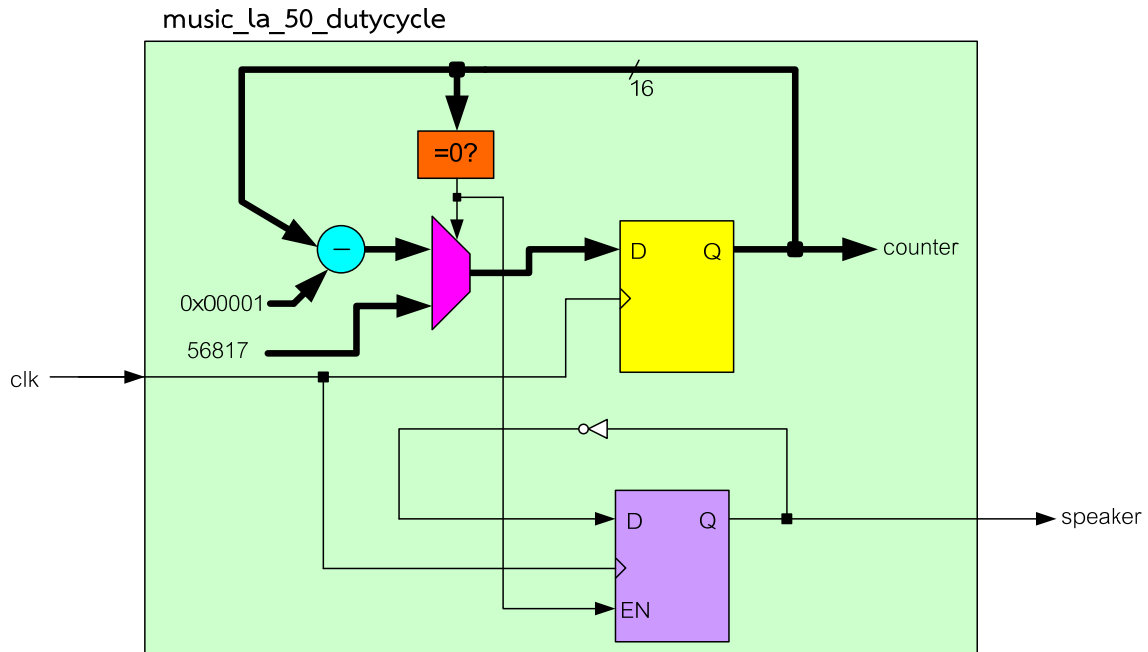
รูปที่ 6 โปรแกรมอย่างง่ายสำหรับสร้างเสียงตัวโน้ต “La”

ถึงแม้ว่าโปรแกรมสร้างเสียงตัวโน้ต “La” ในรูปที่ 6 จะสามารถสร้างเสียงออกทางลำโพงได้ค่าความถี่ที่ค่อนข้างเที่ยงตรง แต่ก็มีข้อเสียคือค่า Duty Cycle ของความถี่เอาต์พุตจะไม่เท่ากับ 50 เปอร์เซ็นต์ ทั้งนี้เนื่องจากการสร้างความถี่เสียงใช้วิธีกลับบิตของวงจรมานับสูงสุด หรือที่ขา counter[16] ดังนั้นค่าที่เอาต์พุตขา counter[16] จะเป็นตรรกะต่ำในขณะที่วงจรมานับค่าตั้งแต่ 0-65535 และจะมีค่าเป็นตรรกะสูงเมื่อวงจรมานับค่าตั้งแต่ 65536-113635 (เป็นตรรกะสูงจำนวน 48100 ค่า) ซึ่งจะทำให้เราสามารถคำนวณค่า Duty Cycle ของสัญญาณเอาต์พุตได้เท่ากับ $48100/113636 \times 100 = 42.3$ เปอร์เซ็นต์

เพื่อแก้ไขปัญหาวงจรสร้างเสียงโน้ต “La” ดังกล่าวให้สามารถสร้างความถี่เอาต์พุตซึ่งมีค่า Duty Cycle ได้เท่ากับ 50 เปอร์เซ็นต์ และยังคงได้ค่าความถี่ 440 เฮิรตซ์เท่าเดิม สามารถทำได้โดยการดัดแปลงโครงสร้างใหม่ให้มีการเชื่อมต่อดังแสดงในรูปที่ 7 ซึ่งจะทำให้มีการปรับเปลี่ยนเป็นวงจรมานับลง และมีการตรวจสอบว่าหากนับลงจนถึงค่า 0 ก็จะมีการตั้งค่าใหม่ให้กับวงจรมานับเท่ากับ 56817 ส่งผลให้วงจรมานับ

ลงในย่าน [56817..0] และมีวงจรตรวจสอบเปรียบเทียบค่าว่าค่าใน counter เท่ากับศูนย์หรือไม่ หากเท่ากับศูนย์ จะมีการสั่งให้ฟลิปฟlopตัวล่างขนาด 1 บิตทำการ Toggle กลับบิตของขาเอาต์พุตซึ่งต่ออยู่กับลำโพงเป็นค่า ตรงกันข้าม โดยจะเห็นว่าในการนับทุก ๆ 56818 สัญญาณนาฬิกาจะมีการกลับบิตที่ขา speaker จำนวน 1 ครั้ง ซึ่งส่งผลให้ได้ความถี่เอาต์พุตค่าประมาณ 440 เฮิรตซ์ออกสู่ลำโพง

วงจรในรูปที่ 7 สามารถสร้างได้โดยการเขียนภาษา Verilog ดังแสดงในโค้ดของรูปที่ 8



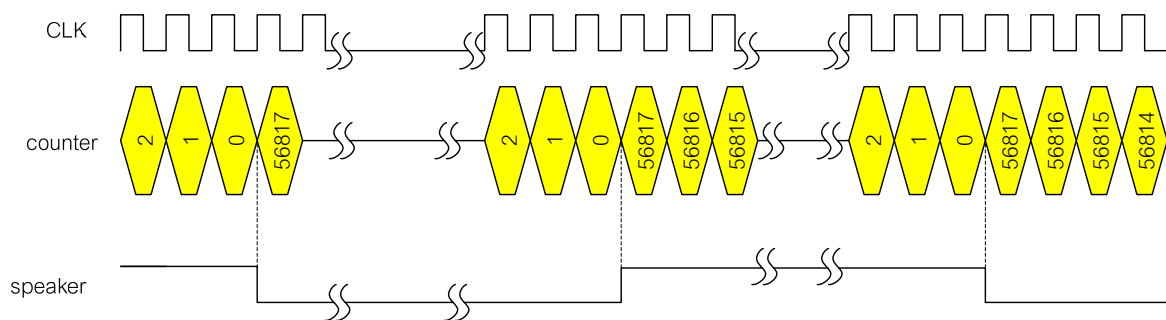
รูปที่ 7 โครงสร้างของวงจรสร้างโน้ต "La" ซึ่งมีค่า Duty Cycle เท่ากับ 50 เปอร์เซ็นต์

```
module music_la_50_dutycycle(clk, speaker);
input clk;
output speaker;
parameter clkdivider = 50000000/440/2;

reg [15:0] counter;
always @(posedge clk)
if(counter==0) counter <= clkdivider-1; else counter <= counter-1;

reg speaker;
always @(posedge clk) if(counter==0) speaker <= ~speaker;
endmodule
```

รูปที่ 8 โปรแกรมสำหรับสร้างตัวโน้ต "La" ซึ่งมีค่า Duty Cycle เท่ากับ 50 เปอร์เซ็นต์



รูปที่ 9 ความถี่ของโน้ตซึ่งได้จากวงจร music_la_50_dutycycle

ตารางที่ 1 ค่าความถี่ของตัวโน้ตแต่ละตัว

เสียง	สัญลักษณ์	ความถี่	สัญลักษณ์	ความถี่	สัญลักษณ์	ความถี่
“Do”	C4	261.63	C5	523.25	C6	1046.50
“Re”	D4	293.66	D5	587.33	D6	1174.66
“Mi”	E4	329.63	E5	659.25	E6	1318.50
“Fa”	F4	349.23	F5	698.46	F6	1396.92
“Sol”	G4	392.00	G5	783.99	G6	1567.98
“La”	A4	440.00	A5	880.00	A6	1760.00
“Si”	B4	493.88	B5	987.77	B6	1975.54

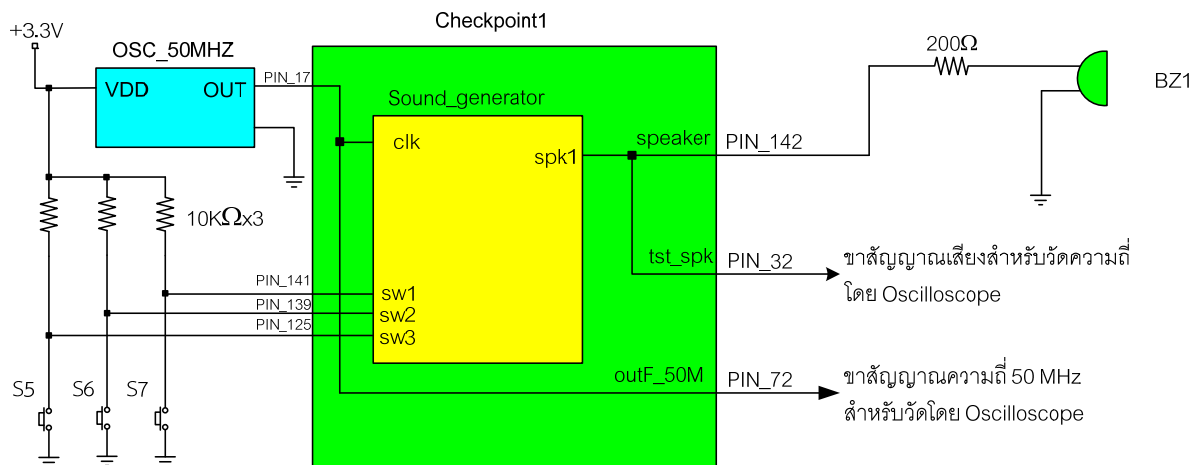
4. สร้างกล่องดนตรีอย่างง่าย

Checkpoint #1

จงสร้างกล่องดนตรีอย่างง่ายซึ่งรับค่าจาก push button สวิตช์ 3 ตัวบนบอร์ด FPGA เพื่อสร้างเสียงโน้ตดนตรี 3 เสียงด้วย FPGA ออกที่บัชเซอร์ BZ1 เพียงตัวเดียว ดังวงจรในรูปที่ 11 โดยหากไม่มีการกดสวิตช์จะไม่ได้ยินเสียง แต่ถ้ากดสวิตช์ปุ่มใดก็จะมีเสียงออกตามที่ได้กำหนดไว้ กำหนดเลือกโน้ตมาจากรหัสเลขประจำตัว 3 ตัวท้ายของนักศึกษาแต่ละคน โดยกำหนดให้แทนตัวเลข 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 ด้วยเสียง C4, D4, E4, F4, G4, A4, B4, C5, D5, E5 ตามลำดับ กำหนดให้ออกแบบวงจรสร้างเสียงใช้โครงสร้างพอร์ตดังรูปที่ 10 และรูปที่ 11 ในการส่งงาน ให้ใช้ออสซิลโลสโคปวัดค่าความถี่ขาเข้าที่ขา 72 (อยู่ใน JP4) ของ FPGA เสียก่อน ทำการบันทึกผลค่าความถี่ขาเข้า จากนั้นให้กดปุ่มที่สวิตช์แต่ละตัวและวัดค่าความถี่ขาออกโดยใช้ออสซิลโลสโคปที่ขา 32 ซึ่งอยู่ใน JP3

```
module musicbox(clk, sw1, sw2, sw3, spk1);
...
...
...
endmodule
```

รูปที่ 10 โครงสร้างพอร์ตของวงจรใน Checkpoint #1



รูปที่ 11 วงจรสำหรับ Checkpoint #1

บันทึกผลการทดลอง checkpoint 1

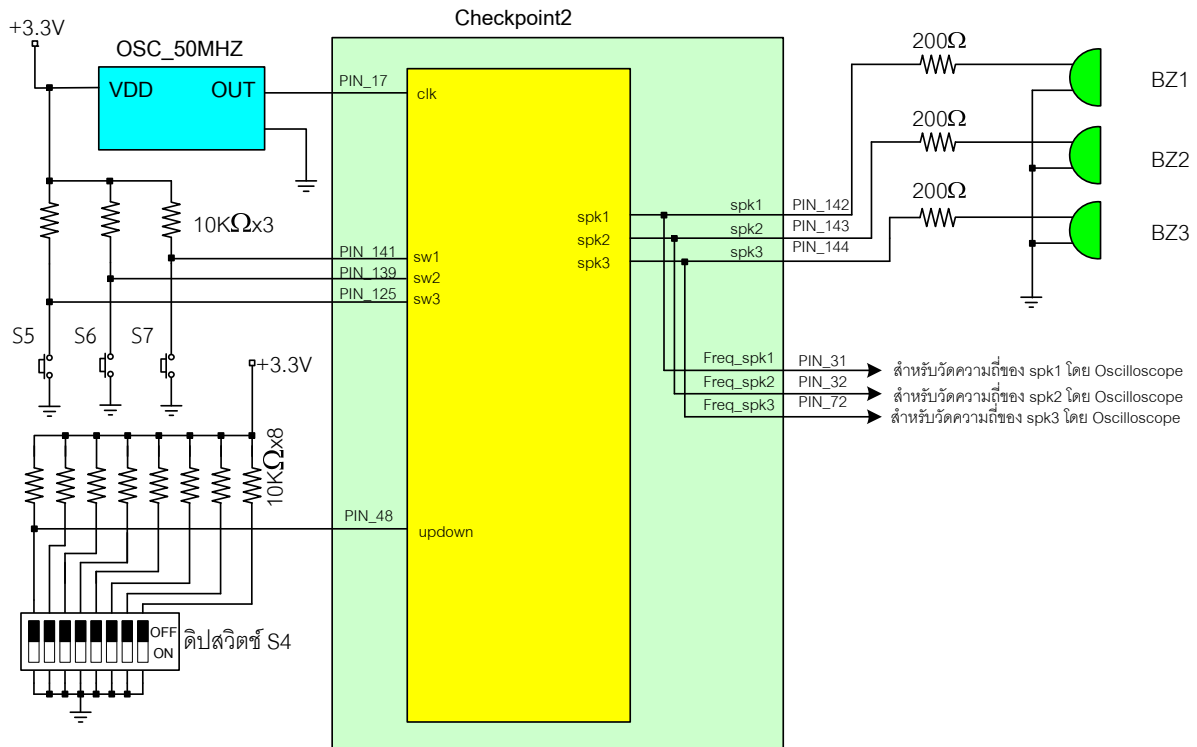
- ค่าความถี่ที่อ่านได้จาก PIN72.....Hz (ให้ถ่ายภาพเก็บไว้ใส่ในรายงาน)
- ค่าความถี่ที่อ่านได้จาก PIN32 เมื่อกดปุ่ม S5.....Hz (ให้ถ่ายภาพเก็บไว้ใส่ในรายงาน)
- ค่าความถี่ที่อ่านได้จาก PIN32 เมื่อกดปุ่ม S6.....Hz (ให้ถ่ายภาพเก็บไว้ใส่ในรายงาน)
- ค่าความถี่ที่อ่านได้จาก PIN32 เมื่อกดปุ่ม S7.....Hz (ให้ถ่ายภาพเก็บไว้ใส่ในรายงาน)

Checkpoint #2

จงสร้างกล่องดนตรีอย่างง่ายซึ่งรับค่าจาก push button สวิตช์ 3 ตัวบนบอร์ด FPGA เพื่อสร้างเสียงโน้ตดนตรี 3 เสียงด้วย FPGA ออกที่บัสเซอร์ BZ1, BZ2, BZ3 โดยกำหนดให้สวิตช์กดติดปล่อยดับแต่ละตัวควบคุมการส่งเสียงโน้ตออกบัสเซอร์ 1 ตัว ดังวงจรในรูปที่ 12 โดยหากไม่มีการกดสวิตช์จะไม่ได้ยินเสียง กำหนดให้ติบสวิตช์จำนวน 1 บิตซึ่งต่อกับ PIN_48 ทำหน้าที่เลือกระดับเสียง หากบิตนี้มีค่าตรรกะต่ำ ให้เสียงที่ออกจากลำโพงเป็นเสียงโน้ตที่ลงท้ายด้วยเลข 4 หรือ 5 (เช่น C4 หรือ C5) เป็นต้น แต่หากบิตนี้มีค่าตรรกะสูง ให้เสียงที่ออกจากลำโพงเป็นเสียงเดียวกัน แต่สูงขึ้นมาอีก 1 ระดับ (เช่น C5 หรือ C6)

ให้นักศึกษาแต่ละคนเลือกโน้ตมาจากรหัสเลขท้ายของนักศึกษา มาจากรหัสเลขประจำตัว 3 ตัวท้ายของนักศึกษาแต่ละคน โดยกำหนดให้แทนตัวเลข 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 ด้วยเสียง C4, D4, E4, F4, G4, A4, B4, C5, D5, E5 ตามลำดับ หากตัวเลขซ้ำกัน 2 ตัว ให้เลือกโน้ตที่สูงขึ้นมาน้อย 1 ระดับที่ไม่ซ้ำกับเลขที่เหลือแทน ยกตัวอย่างเช่น หากนักศึกษารหัสเลขท้าย คือ 322

- ให้เลือกโน้ต E4 สำหรับ เลข 3
- และเลือก D4 สำหรับเลข 2
- และเลือก F4 สำหรับเลข 2 ที่ซ้ำ



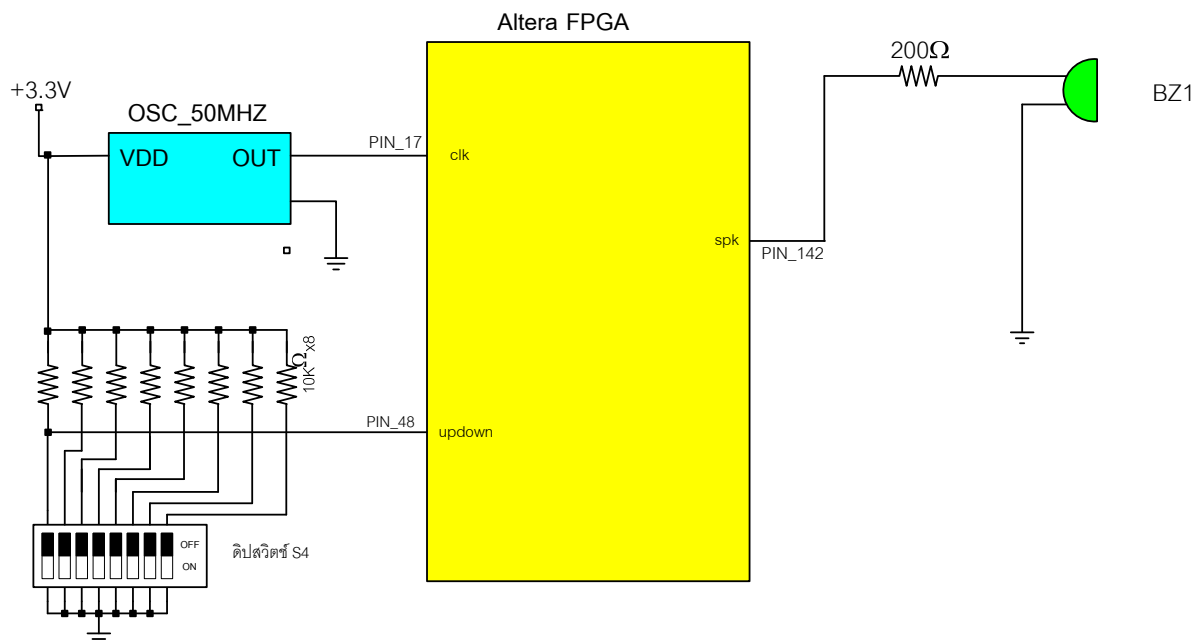
รูปที่ 12 วงจรสำหรับ Checkpoint #2

บันทึกผลการทดลอง checkpoint 2

- กรณีศึกษา updown=0 ค่าความถี่ที่อ่านได้จาก PIN31 เมื่อกดปุ่ม S5.....Hz (ถ่ายภาพเก็บไว้ใส่ในรายงาน)
- กรณีศึกษา updown=1 ค่าความถี่ที่อ่านได้จาก PIN31 เมื่อกดปุ่ม S5.....Hz (ถ่ายภาพเก็บไว้ใส่ในรายงาน)
- กรณีศึกษา updown=0 ค่าความถี่ที่อ่านได้จาก PIN32 เมื่อกดปุ่ม S6.....Hz (ถ่ายภาพเก็บไว้ใส่ในรายงาน)
- กรณีศึกษา updown=1 ค่าความถี่ที่อ่านได้จาก PIN32 เมื่อกดปุ่ม S6.....Hz (ถ่ายภาพเก็บไว้ใส่ในรายงาน)
- กรณีศึกษา updown=0 ค่าความถี่ที่อ่านได้จาก PIN72 เมื่อกดปุ่ม S7.....Hz (ถ่ายภาพเก็บไว้ใส่ในรายงาน)
- กรณีศึกษา updown=1 ค่าความถี่ที่อ่านได้จาก PIN72 เมื่อกดปุ่ม S7.....Hz (ถ่ายภาพเก็บไว้ใส่ในรายงาน)

Checkpoint #3

ส่วนเป็น Checkpoint พิเศษที่นักศึกษาจะทำหรือไม่ก็ได้ แต่ถ้าใครทำจะมีคะแนนพิเศษให้ โดยกำหนดให้วงจรที่สร้างเสียงเพลงซึ่งประกอบด้วยโน้ตดนตรีเล่นต่อเนื่องกันเป็นเพลงความยาวไม่ต่ำกว่า 10 วินาที โดยใช้วงจรดังรูปที่ 13 ขา updown จะเป็นตัวกำหนดคีย์ของเพลงที่เล่น หากบิตที่อ่านได้จากขานี้มีค่าลอจิกต่ำ ก็จะใช้โน้ตเสียงกลางในการเล่น หากบิตนี้มีค่าตรรกะสูง ให้เสียงที่ออกจากลำโพงเป็นเสียงโน้ตเดิม แต่มีระดับสูงขึ้นมาอีก 1 ระดับ (เช่น เดิมเล่นตัวโน้ต C4 แต่หากบิตของขา updown มีค่าลอจิกสูงก็จะเปลี่ยนมาเล่นตัวโน้ต C5 แทน)



รูปที่ 13 วงจรสำหรับ Checkpoint #3

5. วิธีการคิดคะแนน และการส่ง Checkpoint

ในแต่ละ Checkpoint ให้ใช้ออสซิลโลสโคปวัดความถี่สัญญาณเอาต์พุตที่ขาของ Buzzer และถ่ายรูป หรือ capture ภาพผลการวัดความถี่ของสัญญาณที่ได้แนบมาด้วยอย่างน้อย checkpoint ละ 2 รูป

คะแนนของการทดลองนี้แบ่งออกเป็น 3 ส่วน ได้แก่

- | | | |
|-------------------------|----|-------|
| - ส่วนของ Checkpoint #1 | 25 | คะแนน |
| - ส่วนของ Checkpoint #2 | 25 | คะแนน |
| - ส่วนของการสอบภาคทฤษฎี | 50 | คะแนน |

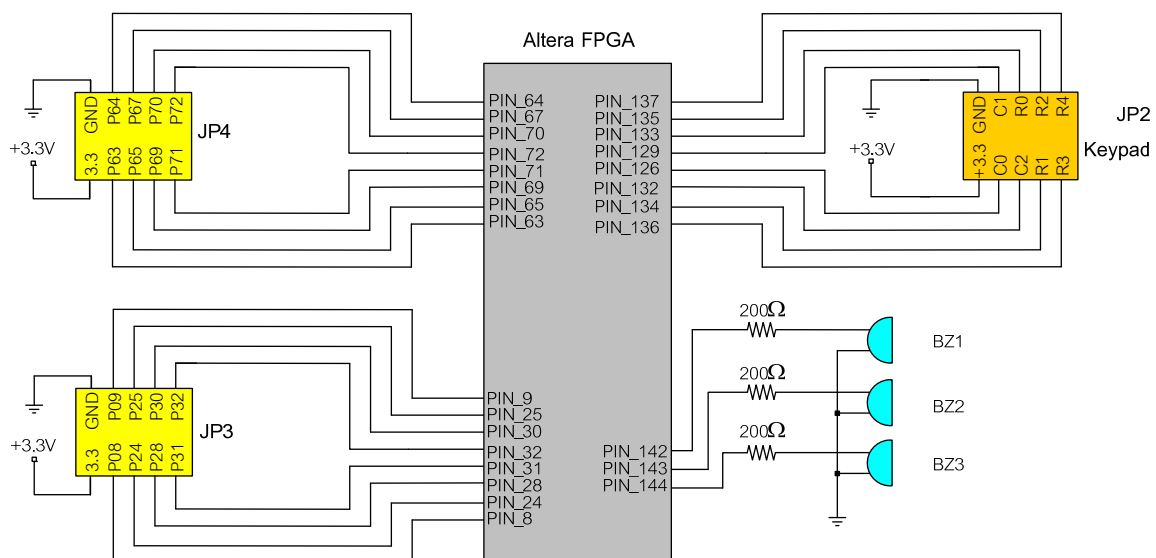
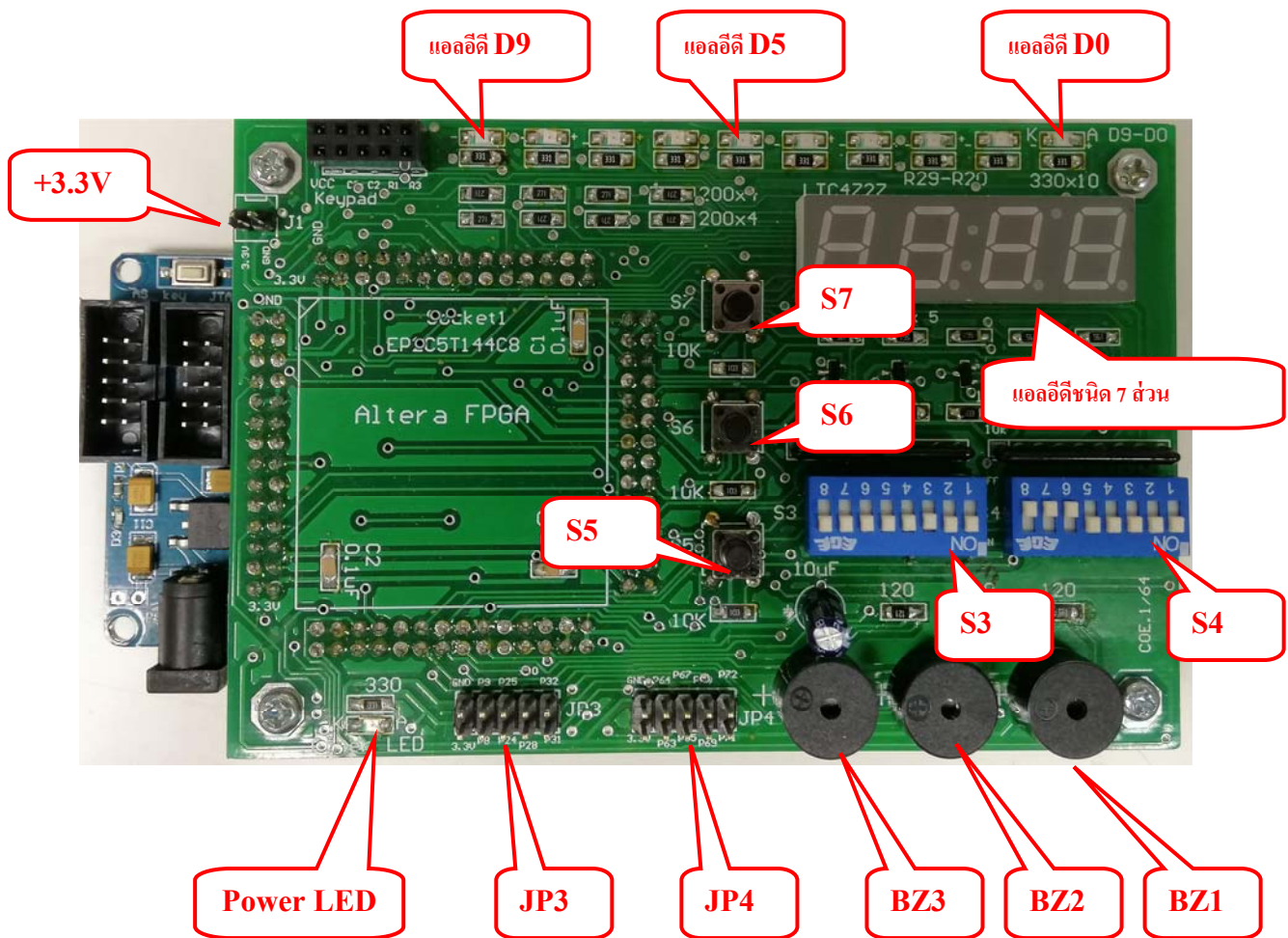
การทดลองนี้ประกอบไปด้วย 2 Checkpoint โดยแบ่งคะแนนออกเป็น Checkpoint ละ 25 คะแนน การให้คะแนนของแต่ละ Checkpoint นั้นพิจารณาจาก

- | | | |
|--|----|---|
| - การทำงานของโค้ดของแต่ละ Checkpoint | 30 | % |
| - การเขียนอธิบายการทำงานของโค้ดของแต่ละ Checkpoint | 30 | % |
| - การส่งงาน | | |

ให้นักศึกษาทุก Section ส่งออนไลน์ใน LMS2 ก่อนเวลา 16.30 น. ของวันที่กำหนด สำหรับในส่วน ของตัวรายงานนั้น นักศึกษาสามารถพิมพ์และสามารถ Capture ภาพจากหน้าจอแปลงในตัวรายงานได้ จากนั้นให้แปลงเป็นไฟล์ .pdf ส่งใน LMS2 เพียงไฟล์เดียว ห้ามส่งไฟล์ฟอร์แมตอื่น

หมายเหตุ

สำหรับคนที่ทำ checkpoint3 ให้ใส่โค้ดและเขียนอธิบายรายละเอียดของโค้ดลงในรายงานด้วย



วงจรของ FPGA Shield ในส่วนของขาเชื่อมต่อคีย์แพด บัสเซอร์ และขาเชื่อมต่อขนาด 8 บิต JP3 และ JP4

----- END -----