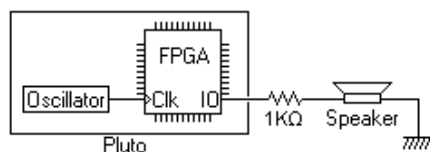


3HA08 : Simple Music Box using FPGA

1. แนะนำ

ใน Lab นี้เป็นการสอนเพื่อให้ทราบว่าใช้งาน FPGA เล่นเสียงหรือดนตรีแบบง่ายได้อย่างไร โดยเป็นการประยุกต์ใช้งานวงจรสร้างความถี่ จากรูปที่ 1 บอร์ด FPGA จะมีตัวกำเนิดสัญญาณนาฬิกา (Oscillator) อยู่ภายใน ความถี่ 25 MHz ตัวกำเนิดสัญญาณนาฬิกานี้จะให้คลื่นความถี่ที่คงที่ให้แก่ FPGA โดยนักศึกษาจะต้องทำการเขียนโปรแกรมให้ตัว FPGA ทำการหารแบ่งความถี่นั้นก่อนที่จะจัดส่งออกทางขาสัญญาณเอาต์พุตที่ต่ออยู่กับ speaker ผ่านตัวต้านทานขนาด 1 กิโลโห์ม การเปลี่ยนความถี่ของขาเอาต์พุตช่วยให้ FPGA สามารถสร้างเสียงที่แตกต่างกันได้



รูปที่ 1 บอร์ด FPGA กับ การสร้างเสียงด้วยการเปลี่ยนความถี่อย่างง่าย

2. วิธีการสร้างเสียงอย่างง่ายออกทางลำโพง

ในการสร้างเสียงอย่างง่ายจะใช้วิธีการสร้างวงจรนับขนาด 16 บิต ดังโปรแกรมในรูปที่ 2 ซึ่งจะเป็นวงจร Counter ทำหน้าที่รับสัญญาณนาฬิกาความถี่ 25 MHz เข้ามา หากพิจารณาที่บิตล่างสุดที่ขา counter[0] จะ toggle บิตที่ความถี่ 12.5 MHz ส่วนบิตถัดมาคือบิต counter[1] จะ toggle บิตที่ความถี่ 6.125 MHz ซึ่งจะเห็นว่าบิตที่สูงขึ้นมา 1 ตำแหน่งจะทำการหารความถี่เพิ่มขึ้นตำแหน่งละ 2 ดังนั้น ที่บิตสูงสุด (MSB) ของ counter[15] จะ toggle ที่ความถี่เท่ากับ $25 \times 10^6 / 65536 = 381 \text{ Hz}$

```
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 2 โปรแกรมสร้างเสียงความถี่ 381 Hz ออกทางลำโพง

3. การสร้างโน้ตดนตรีอย่างง่าย

โปรแกรมตัวอย่างในรูปที่ 3 เป็นการสร้างเสียงโน้ต “La” ออกทางลำโพงโดยใช้วิธีการหารค่าความถี่อินพุต 25 MHz ด้วยค่า 56818 ซึ่งจะได้ค่าความถี่เท่ากับ 440.0014 Hz ซึ่งเป็นค่าความถี่ใกล้เคียงกับเสียงโน้ตตัว “La” ซึ่งมีค่าความถี่เท่ากับ 440 Hz โดยการสร้างเสียงโน้ตใช้วิธีการต่อลำโพงกับขา counter[15] เหมือนเดิม

```
module music(clk, speaker);  
input clk;  
output speaker;  
  
reg [15:0] counter;  
always @(posedge clk) if(counter==56817) counter <= 0; else counter <= counter+1;  
  
assign speaker = counter[15];  
endmodule
```

รูปที่ 3 โปรแกรมสำหรับสร้างเสียงตัวโน้ต “La”

ถึงแม้ว่าโปรแกรมสร้างเสียงตัวโน้ต “La” ในรูปที่ 3 จะสามารถสร้างเสียงออกทางลำโพงได้ค่าความถี่ที่ค่อนข้างเที่ยงตรง แต่ก็มีข้อเสียคือค่า Duty Cycle ของความถี่เอาต์พุตจะไม่เท่ากับ 50 เปอร์เซ็นต์ ทั้งนี้เนื่องจากการสร้างค่าเสียงใช้วิธีการกลับบิตของวงจรรนับบิตสูงสุด หรือที่ขา counter[15] ดังนั้นค่าที่เอาต์พุตขา counter[15] จะเป็นลอจิกต่ำในขณะที่วงจรรนับค่าตั้งแต่ 0-32767 และจะมีค่าเป็นลอจิกสูงเมื่อวงจรรนับค่าตั้งแต่ 32768-56817 หรือพูดอีกนัยหนึ่งก็คือเอาต์พุตจะมีค่า Duty Cycle ประมาณ 42 เปอร์เซ็นต์

โปรแกรมในรูปที่ 4 จะเป็นการแก้ไขข้อบกพร่องของโปรแกรมในรูปที่ 3 โดยจะเป็นการสร้างวงจรรนับลงตั้งแต่ 28408 - 0 และมีการสร้าง state สำหรับการกลับบิตซึ่งต่อกับลำโพงเป็นบิตตรงกันข้าม

```
module music(clk, speaker);  
input clk;  
output speaker;  
parameter clkdivider = 25000000/440/2;  
  
reg [14:0] 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 4 โปรแกรมสำหรับสร้างตัวโน้ต “La” ซึ่งมีค่า Duty Cycle เท่ากับ 50 เปอร์เซ็นต์

ตารางที่ 1 ค่าความถี่ของตัวโน้ตแต่ละตัว

เสียง	สัญลักษณ์	ค่าความถี่โดยประมาณ (Hz)
“Do”	C4	262
“Re”	D4	294
“Mi”	E4	330
“Fa”	F4	349
“Sol”	G4	392
“La”	A4	440
“Si”	B4	494

4. สร้างกล่องดนตรีอย่างง่าย

จงสร้างกล่องดนตรีอย่างง่ายซึ่งรับค่าจาก push button สวิตช์ 3 ตัวบนบอร์ด FPGA เพื่อสร้างเสียง โน้ตดนตรี 3 เสียงด้วย FPGA โดยหากไม่มีการกดสวิตช์จะไม่ได้ยินเสียง แต่ถ้ากดสวิตช์ปุ่มใดก็จะมีเสียงออก ตามที่ได้กำหนดไว้ กำหนดเลือกโน้ตมาจากรหัสเลขประจำตัว 3 ตัวท้ายของสมาชิกในกลุ่ม โดยกำหนดให้แทน ตัวเลข 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 ด้วยเสียง Do, Re, Mi, Fa, Sol, La, Si, Do, Re, Mi ตามลำดับ

```
module musicbox(clk, sw1, sw2, sw3, speaker);

...

endmodule
```

Check-point

	ลายเซ็นต์	วันที่
Check #1: สร้างเสียง 381 Hz ออกทาง buzzer ได้		
Check #2: สร้างโน้ต “La” ออกทาง buzzer ได้		
Check #3: สร้างเสียงโน้ต 3 เสียงที่ต่างกันออกทาง buzzer ด้วยการกดสวิตช์ 3 ปุ่ม		