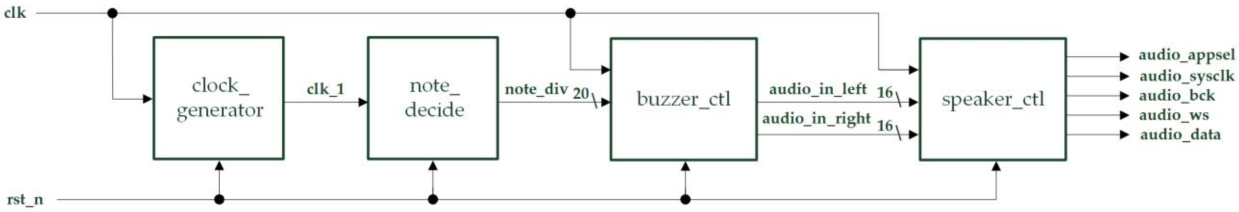
Lab 10: Electronic Organ

103061207 徐安廷 An-Ting Hsu

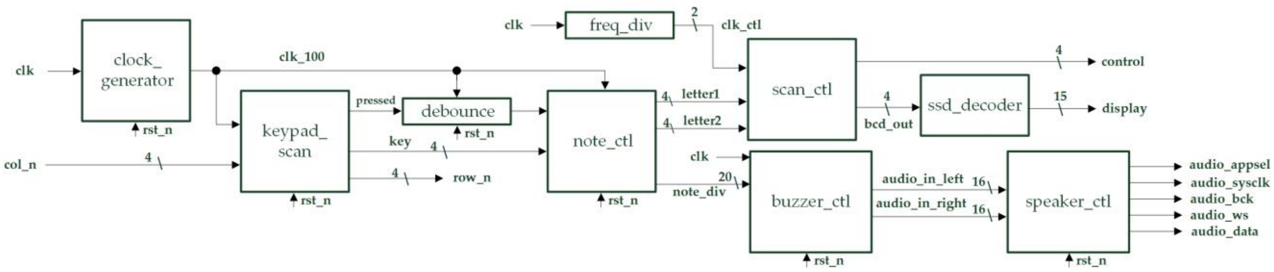
Design Specification

1. **Repeat Speaker**
   - ✓ Experiment Goal:
     Let the board play 16 sounds repeatly. Every sound is played for one second.
   - ✓ Block Diagram:



   - ✓ I/Os:
     Inputs: clk, rst_n.
     Outputs: audio_appsel, audio_sysclk, audio_bcd, audio_ws, audio_data.
   - ✓ Details about some modules:
     - ■ note_decide: It has an 1Hz clock signal input, so it will accord to the clock and change the note signal in every one second.

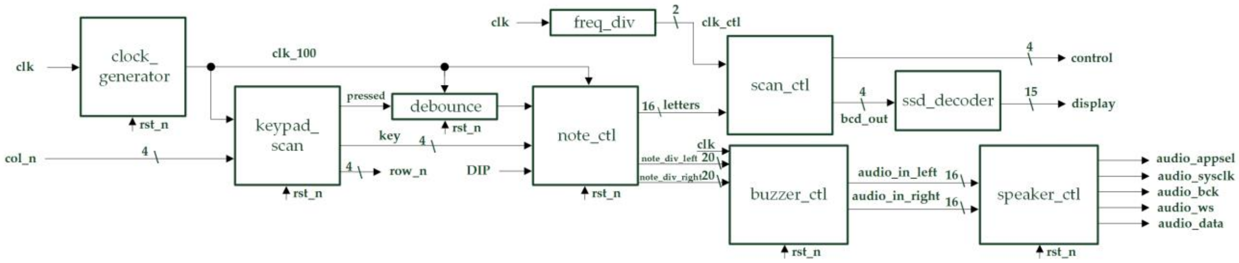
2. **Electronic Organ – Single Tone**
   - ✓ Experiment Goal:
     When the keypad is pressed, the speaker will generate out corresponding sound and the 14SD will show out the sound.
   - ✓ Block Diagram:



   - ✓ I/Os:
     Inputs: clk, rst_n, [3:0] col_n.
     Outputs: audio_appsel, audio_sysclk, audio_bcd, audio_ws, audio_data, [3:0] row_n, [3:0] control, [14:0] display.
   - ✓ Details about some module:
     - ■ note_ctl: Use the keypad outputs as inputs to determine the note signal output to the buzzer.

3. **(Bonus) Electronic Organ – Double Tones**
   - ✓ Experiment Goal:
     When the keypad is pressed and DIP switch is off, the speaker will generate out corresponding single tone sound and the 14SD will show out the sound. Otherwise, if the DIP switch is on, the speaker will generate a double tones sound out.

- ✓ Block Diagram:



- ✓ I/Os:
  Inputs: clk, DIP, rst_n, [3:0] col_n.
  Outputs: audio_appsel, audio_sysclk, audio_bcd, audio_ws, audio_data, [3:0] row_n, [3:0] control, [14:0] display.
- ✓ Details about some module:
  - ■ note_ctl: Use the keypad outputs and DIP switch signal as inputs to determine the note signal output to the buzzer.

Design Implementation

1. Repeat Speaker

- ✓ First, construct a note decision module that can change the note_div output signal in every one second. Second, connect the note decision module to buzzer_ctl & speaker_ctl module.
- ✓ I/O Pin Assignments:

Port Name	Assignment	Function
clk	R10	FPGA board oscillator input
rst_n	T2	Active low reset input button
audio_appsel	H18	Playing mode selection output
audio_sysclk	H17	Control clock for DAC output
audio_bck	K16	Bit clock of audio data output
audio_ws	L15	Left/right parallel to serial control output
audio_data	L16	Serial output audio data

2. Electronic Organ – Single Tone

- ✓ First, copy the keypad scanner module in lab6. Second, build the note control module which the inputs are connect from keypad scanner, so it can generate the right note output. Third, connect the note control module outputs to speaker and 14SD display module.
- ✓ I/O Pin Assignments:

Port Name	Assignment	Function
clk	R10	FPGA board oscillator input
rst_n	T2	Active low reset button input
col_n[0]~col_n[3]	J3, J1, H2, H1	Pressed column index input
audio_appsel	H18	Playing mode selection output
audio_sysclk	H17	Control clock for DAC output

audio_bck	K16	Bit clock of audio data output
audio_ws	L15	Left/right parallel to serial control output
audio_data	L16	Serial output audio data
row_n[0]~row_n[3]	K2, K1, L4, L3	Scanned row index output
control[0]~control[3]	V8, U8, V6, T6	14-SD control signal output
display[0]~display[14]	U5, T7, R7, V7, V4, T4, T3, R5, N5, R3, U7, T5, V5, N4, P6	14-SD light control signal output

3. (Bonus) Electronic Organ – Double Tones

- ✓ First, adjust the note control module to support DIP switch input and double tones output. Second, assign the "audio_appsel" signal in buzzer controller to 1'b0. Third, connect the wires as in experiment 2.
- ✓ I/O Pin Assignments:

Port Name	Assignment	Function
clk	R10	FPGA board oscillator input
DIP	L2	DIP switch input
rst_n	T2	Active low reset button input
col_n[0]~col_n[3]	J3, J1, H2, H1	Pressed column index input
audio_appsel	H18	Playing mode selection output
audio_sysclk	H17	Control clock for DAC output
audio_bck	K16	Bit clock of audio data output
audio_ws	L15	Left/right parallel to serial control output
audio_data	L16	Serial output audio data
row_n[0]~row_n[3]	K2, K1, L4, L3	Scanned row index output
control[0]~control[3]	V8, U8, V6, T6	14-SD control signal output
display[0]~display[14]	U5, T7, R7, V7, V4, T4, T3, R5, N5, R3, U7, T5, V5, N4, P6	14-SD light control signal output

Discussion

- ✓ The only problem I met is when I was doing experiment 2. Because I want the speaker to have sound only when the keypad is pressed, so I connect the "pressed" signal to note controller. The idea seems to be good, but bugs showed up when I programed the code. The speaker sound sounds like a little bit shivery. I debugged for about an hour, and find out that the "pressed" signal from the keypad scanner is not in a clean waveform. It is oscillating in a small amplitude, so I send the signal to a debounce circuit to make it cleaner. And, it works perfectly just after I do this small adjustment.

Conclusion

- ✓ Since we can produce 16 kinds of sound by pressing keypad, we can play some easy music, and I really played it for a long time. I think maybe I can build a module which can play music automatically and repeatly, and used it in the final project to make it more interesting.

References

- ✓ Previous codes in Lab 6 (keypad scanner) and Lab 9.