Nolan Tremelling
Department of Electrical Engineering
Columbia University
EE E3082. Digital Electronics Laboratory

Laboratory 5: Electric piano using Xilinx FPGAs, Part I

2.1 User Interface

The notes are encoded using an 8-bit binary system in which the sequential bits represent a different note. The first note, for example, being C3, is represented by the 8-bit binary number 00000001. Then, D3 is represented by 00000011, and so on. It's important to note that flat and sharp notes are sequentially marked, so C3# or D3b are both represented as 00000010. Control for the sharps and flats are defined within the pb_in function where PB(1) represents a flat, PB(2) represents a sharp, and PB(3) represents a raised octave. These combinations allow for a variety of notes.

2.2 Clock Divider

Div $50 = \frac{\text{input clock}}{\text{input frequency}} \cdot 0.5$

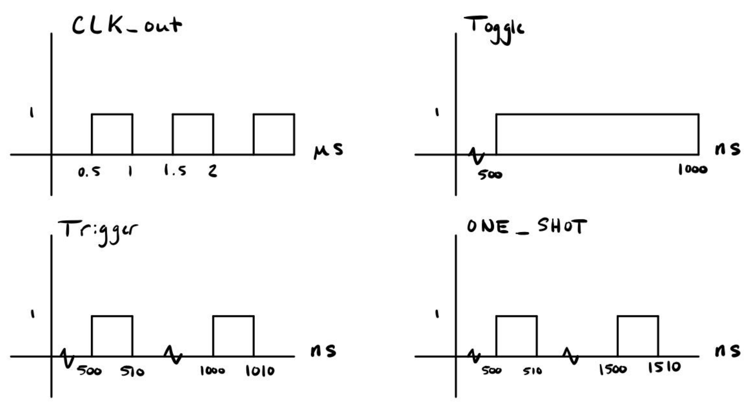
$50 = \frac{100\text{ MHz}}{\text{output}} \cdot 0.5$

$100 = \frac{100 \times 10^6}{\text{output}}$

$10^6\text{ Hz} = 1\text{ MHz}$

Period $\frac{1}{1\text{ MHz}} = 1\ \mu s$

Input Period $\frac{1}{100\text{ MHz}} = 10\text{ ns}$



2.3 Note Decoder

The output frequency as a function of divider value for a 1MHz input is as follows: $f(x) = \frac{5\times10^5}{divider\ value}$.

Note	Hexadecimal Value	Decimal Value	Output Frequency	True Frequency	Error
C3	0EEE	3822	130.8215594	130.81	-0.0088368 %
C3#	0E18	3608	138.5809313	138.59	0.0065436 %
D3#	0C8E	3214	155.5693839	155.56	-0.0060324 %
E3	0BDA	3034	164.7989453	164.81	0.0067076 %

G3	09F7	2551	196.001568	196.00	-0.0008000 %
A3b	0968	2408	207.641196	207.65	0.0042398 %
A4b	04B4	1204	415.282392	415.30	0.0042398 %
E4	05ED	1517	329.5978906	329.63	0.0097411%
F4#	0547	1351	370.096225	369.99	-0.0287102 %
A4#	0431	1073	465.9832246	466.16	0.0379216 %

2.4 7-Segment Display State Machine

The use of the 7-segment display allows for us to display the note being played by the piano at a given time. This is done through a multiplexed 7-segment display, scanned at a rate that allows for a visual persistence that closely matches that of a television—ultimately producing a visually "smooth" output. Then using the 7-segment display—which utilizes a number of inputs that act as cathodes for each of the 7 segments—we can display combinations to form alphanumeric representations. In the seven_seg.vhl file we see that a 9-bit value is assigned to each note. These bits follow the following pattern:

- The first four bits, i.e. "1100", that are read from NOTE_IN are used to indicate the letter of the note. In this case, "C".
- The next four bits, i.e. "0011", are used to indicate the octave of the note. In this case, "3".
- The last bit is used to indicate whether a note is sharp or not, so if the last bit is "1", the note is sharp.
- Putting together all this information, the seven_seg.vhl file will interpret "110000111" as a C3#.

Then, using the point_now function, this newfound value is mapped into the seven segment display.

3.3 Simulating your Design

The following code, which includes varied test stimuli for different values of pb_in and switch_in, can be seen below.

```
-- *** Test Bench - User Defined Section ***
   tb : PROCESS
   BEGIN

            -- System Reset
            pb_in(0) <= '1';
            wait for 50 ns;
            pb_in(0) <= '0';

            wait for 1 us;

            -- Test note for C3 output 0eee
```

```
        switch_in <= "10000000"; pb_in <= "0000";
        wait for 2us;
        -- Test note for E3 output 0bda
        switch_in <= "00100000"; pb_in <= "0000";
        wait for 2us;
        -- Test note for E#3 output 0b30
        switch_in <= "00100000"; pb_in <= "0100";
        wait for 2us;
        -- Test note for A#4 output 0431
        switch_in <= "00000100"; pb_in <= "1100";
        wait for 2us;
        -- Test note for Gb3 output 0a8e
        switch_in <= "00001000"; pb_in <= "0010";
        wait for 2us;
        -- Test note for E4 output 05ed
        switch_in <= "00100000"; pb_in <= "1000";
        wait for 2us;

    END PROCESS;
-- *** End Test Bench - User Defined Section ***
```

When simulated, this code shows that the expected notes can be observed as seen in the following image.