<div align="center">

Nolan Tremelling
Department of Electrical Engineering
Columbia University
EE E3082. Digital Electronics Laboratory
Laboratory 6: Electric piano using Xilinx FPGAs, Part II

</div>

Using the Basys 3 board, VHDL code can be written to play a song on the electric piano. This could be done in a number of ways; however , by utilizing the clock cycles on the board, one can create a simple loop that plays a given note after a certain amount of time. This process was used to play the piano introduction to Kanye West's *Runaway*. A video of the completed project can be seen here.

The Notes and Display
The introduction to *Runaway*, which is a rather simple song consisting of fifteen notes each played individually—each note is played for one second, with a small break of about a quarter of a second between notes.

<div align="center">

E4 E4 E3 D#4 D#4 D#4 D#3 C#4 C#4 C#4 C#3 A3 A3 G#3 E4

</div>

The values for each of the notes were defined in the note_gen.vhd file. This file, which uses 5-bits to name each note, assigns the correct frequency output to the speaker. Similarly, the values for each of the displays were defined in the seven_seg.vhd file. Each of the notes consist of a letter, a number denoting the octave, as well as one bit to define whether or not the note is raised or lowered a half step. The alphanumeric values are displayed as such, and the sharp/flat notation is a single "dot" on the bottom right of the display.

The Code
The following code was made in the piano.vhd file, and were the only changes necessary for this project. The code is split into parts for readability and explanation, but is one process.

```
note_in <= note_next;
process (CLK,RST)
variable cnt : integer range 0 to 2147483647; -- Creates a count variable
with highest int range possible
begin
```

A variable cnt is declared to be an integer that can range from 0 to 2147483648—the highest possible value for an integer.

```
   if (RST = '1') then
      note_next <= (others => '0');
   elsif (CLK'event and CLK = '1') then
      if (switch(0) = '1') then
         cnt := cnt + 1; -- Increases cnt variable each clock cycle
   end if;
```

The cnt variable is then defined to increase by 1 each clock cycle when the 0 switch is in the on position.

```vhdl
    if (cnt > 1 and cnt < 100000000) then    -- Clock runs on 100MHz
 oscillator resulting in 100000000 cycles per second
        note_next <= "10101"; -- E4
        elsif(cnt > 100000000 and cnt < 125000000) then
            note_next <= "00000";    -- Wait
        elsif (cnt > 125000000 and cnt < 225000000) then
            note_next <= "10101";    -- E4
        elsif (cnt > 250000000 and cnt < 275000000) then
            note_next <= "00000";    -- Wait
        elsif (cnt > 275000000 and cnt < 375000000) then
            note_next <= "00101";    -- E3
        elsif (cnt > 375000000 and cnt < 400000000) then
            note_next <= "00000";    -- Wait
        elsif (cnt > 400000000 and cnt < 500000000) then
            note_next <= "10100";    -- D#4
        elsif (cnt > 500000000 and cnt < 525000000) then
            note_next <= "00000";    -- Wait
        elsif (cnt > 525000000 and cnt < 625000000) then
            note_next <= "10100";    -- D#4
        elsif (cnt > 625000000 and cnt < 650000000) then
            note_next <= "00000";    -- Wait
        elsif (cnt > 650000000 and cnt < 750000000) then
            note_next <= "10100";    -- D#4
        elsif (cnt > 750000000 and cnt < 775000000) then
            note_next <= "00000";    -- Wait
        elsif (cnt > 775000000 and cnt < 875000000) then
            note_next <= "00100";    -- D#3
        elsif (cnt > (second*17)/2 and cnt < (second*18)/2) then
            note_next <= "00000";    -- Wait
        elsif (cnt > 875000000 and cnt < 975000000) then
            note_next <= "10010";    -- C#4
        elsif (cnt > 975000000 and cnt < 1000000000) then
            note_next <= "00000";    -- Wait
        elsif (cnt > 1000000000 and cnt < 1100000000) then
            note_next <= "10010";    -- C#4
        elsif (cnt > 1100000000 and cnt < 1125000000) then
            note_next <= "00000";    -- Wait
        elsif (cnt > 1125000000 and cnt < 1225000000) then
            note_next <= "10010";    -- C#4
        elsif (cnt > 1225000000 and cnt < 1250000000) then
            note_next <= "00000";    -- Wait
```

```vhdl
        elsif (cnt > 1250000000 and cnt < 1350000000) then
            note_next <= "00010";   -- C#3
        elsif (cnt > 1350000000 and cnt < 1375000000) then
            note_next <= "00000";   -- Wait
        elsif (cnt > 1375000000 and cnt < 1475000000) then
            note_next <= "01010";   -- A3
        elsif (cnt > 1475000000 and cnt < 1500000000) then
            note_next <= "00000";   -- Wait
        elsif (cnt > 1500000000 and cnt < 1600000000) then
            note_next <= "01010";   -- A3
        elsif (cnt > 1600000000 and cnt < 1625000000) then
            note_next <= "00000";   -- Wait
        elsif (cnt > 1625000000 and cnt < 1750000000) then
            note_next <= "01001";   -- G#3
        elsif (cnt > 1750000000 and cnt < 1775000000) then
            note_next <= "00000";   -- Wait
        elsif (cnt > 1775000000 and cnt < 1875000000) then
            note_next <= "10101";   -- E4
        end if;
    end if;
 end process;
```

Comparative if statements then output a given note, note_next for a range of the cnt variable. Each note is a 5-bit number defined in the note_gen.vhd file. Since the clock cycle of the Basys 3 is 100MHz, there are 100000000 cycles per second. Each note is played for 1 second, or 100000000 clock cycles and there is a .25 second or 2500000 clock cycle pause between each note. Once the song has been played, the process ends.