**FPGA2-POST**

**PART 1 - PS2:**

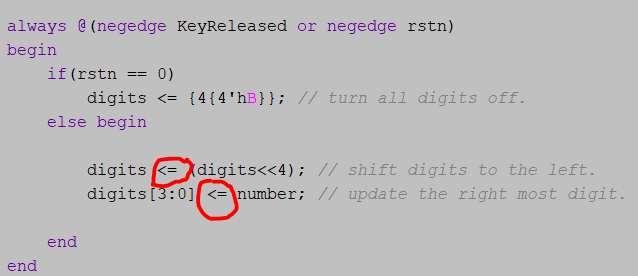
During our first sessions, our implementation didn’t respond at all. Meaning the LEDs and the 7-seg display didn’t react to key presses.

Further investigation led to a problem originating from the ‘rstn’ signal.

The ‘rstn’ was treated as the ‘reset’ input, We fixed this issue by negating this signal in the conditions it appeared in.

In our second lab session, the LEDs and reset button responded as expected, but the display had a bug when every keypress caused two digits to appear: the left digit was random while the second was correct.

The error originated from the following block of code:

 The problem was fixed by replacing ‘<=’ with ‘=’.

This solved the issue because we don’t want these two lines to occur simultaneously, we want the shift to occur prior to the left-most digit update so the use of ‘=’ is needed instead of ‘<=’.

Essentially, the bits were being shifted to the left and new data was inserted from the right at the same time, making the second digit get “random” values.

Another issue we dealt with was a warning about an[3:2] being constant in the synthesis. This is obviously not intended as only 2 out of the 4 digits will change under that constraint. Turns out, the ‘s’ wire wasn’t defined properly and consisted of a single bit bus instead of 2. Thus, indexing only 2 out of the 4 bits of ‘an’. Redefining ‘s’ got all bits of ‘an’ changeable.

**Part 2 – VGA display:**

When first testing the VGA task on the board, the screen presented a black image. Our ‘drawer’ module consists of a path of ‘if’ and ‘else if’ eventually leading every pixel to it’s spot and determining its color. The various conditions divide the pixel into slots whether it's in retrace-area / piano top panel / black keys area etc. The bug was caused by a default ‘else’ option at the bottom of the code that painted a pixel black if it didn’t meet any criteria of the ‘drawer.v’.

The original intention of set condition was for pixels that are in the retrace area. The defaulted ‘else’ was added in order to remove a warning about the possibility of ‘pixel\_color’ remaining undefined when not meeting any of the ‘if’ / ‘else if’ conditions of the drawer.

Eventually changing that default ‘else’ got the image to present properly (as shown below).

Painting a piano was an easier choice due to all the lines being horizontal or vertical.

