Digital System Design

Submission Date: 23rd Nov, 2017

Question 1:

a. Add two fixed-point ***signed*** numbers give the equivalent floating point value of the operand and the sum,

$$a = 111\_10\ in\ Q3.2\ and$$
$$b = 111\_111\ in\ Q3.3\ format$$

b. Perform $signed\ by\ unsigned, signed\ by\ signed\ and\ unsigned\ by\ signed$ multiplication of the numbers given in part a, give equivalent floating point value of the product.
   - For each multiplication show the numbers (a, b and out) being used in VERILOG and their equivalent real numbers.
   - Afterwards apply round before truncate to get final format of Q6.3

c. Convert the numbers ($a, b\ and\ sum$) in IEEE floating point format.

d. Calculate the following using 16 bit floating point format.

$$w = x + 2 * y$$

$$where\ x = 10.782, y = -90.252$$

Hint: Let N be 6

e. Draw the internal architecture of a floating point format multiplier and figuratively show the working (signal values and their direction) by multiplying the numbers given in part d.

Question 2:

a. Convert the following C code to **its equivalent fixed point listing in C.** Consider $x, b, a, and\ out$ are in 16 bit format (Q1.15) and acc in Q8.32 format. Check corner cases and saturate the result if overflow occurs.

b. Calculate the rms error between floating point and fixed point implementation (calculate it on paper).

```
float example (float x[], float y[], float b[], float a[])
{
        float acc, out;
        int i;
        acc = 0;
        for ( i = 0; i < 2; i++)
        {
                acc += b[i] * x[i] + a[i] * y[i];
        }
        out = acc;
        return out;
}
```

The Constants are:

$$float\ b[] = \{0.258, -0.309, -0.704\};$$

$$float\ a[] = \{-0.123, -0.51, 0.223\};$$
$$float\ x[] = \{-0.19813, -0.76291, 0.57407);$$
$$float\ y[] = \{0.123, -0.213, -0.912\}$$

Question 3:

Improve the code learnt in last lab session to implement digital clock on FPGA such that the 4 7-segments on the given board must be used to display seconds and minutes or minutes and hours which will be controlled via a button. The display must be continuous in such a manner that all 7-segments display separate values [***Mins Ten:Mins Unit:Secs Tens:Secs Unit*** or ***HRS Tens:HRS Unit: Mins Ten:Mins Unit***]. This can be achieved by controlling what is to be displayed on the 7-segment using mux, the select line (0, 1, 2 and 3) of the mux is to be self-generated within the code. This select line must update its value (0->1, 1->2 and so on) after every 1msec, in doing so you will observe that all the segments ***seem*** to operate in parallel (but in reality they are working in sequence).

Furthermore, add a reset button that restarts the clock from 00:00:00.

Additional marks:

FPGA is a volatile device which resets itself whenever power is turned off, we can make use of available EPROM on FPGA to turn it into a non-volatile device. Additional marks will be awarded to people who can learn and make your design non-volatile meaning that whenever power is turned on your code starts.