2020 Digital IC Design Homework 3: Approximate Average

NAME	Nguyen Vu Le Minh				
Student ID	P76087099				
Simulation Result					
Functional simulation	Pass	Gate-level simulation	Pass	Gate-level simulation time	196392 (ns)

(your pre-sim result)

```
VSIM 3> run -all
# ----------------------------------------------
#
# ----------------------------------------------
#
# All data have been generated successfully!
#
# -------------------PASS-------------------
#
# ----------------------------------------------
#
# ** Note: $finish    : P:/2. Study/Semester 2/Digital IC DESIG
#    Time: 200400 ns  Iteration: 2  Instance: /test
```

(your post-sim result)

```
# All data have been generated successfully!
#
# -------------------PASS-------------------
#
# ----------------------------------------------
#
# ** Note: $finish    : P:/2. Study/Semester 2/Digital IC
#    Time: 196392 ns  Iteration: 2  Instance: /test
# 1
```

Synthesis Result

Total logic elements	713 / 68,416 (1 %)
Total memory bit	0 / 1,152,000 (0 %)
Embedded multiplier 9-bit element	0 / 300 (0 %)

Flow Status	Successful - Tue May 05 17:25:27 2020
Quartus II Version	10.0 Build 262 08/18/2010 SP 1 SJ Full Version
Revision Name	CS
Top-level Entity Name	CS
Family	Cyclone II
Device	EP2C70F896C8
Timing Models	Final
Met timing requirements	No
Total logic elements	713 / 68,416 (1 %)
Total combinational functions	713 / 68,416 (1 %)
Dedicated logic registers	84 / 68,416 (< 1 %)
Total registers	84
Total pins	20 / 622 (3 %)
Total virtual pins	0
Total memory bits	0 / 1,152,000 (0 %)
Embedded Multiplier 9-bit elements	0 / 300 (0 %)
Total PLLs	0 / 4 (0 %)

Description of your design

The average is simply the sum of the numbers in a given problem, divided by the number of numbers added together. For example, if four number are added together their sum is divided by four to find the average or arithmetic mean.

$$Xavg_j = \left\lfloor \frac{\sum_{i=j}^{j+n-1} X_i}{n} \right\rfloor \quad \text{(1)}$$

where X_i is the value of the ith input data and $j \geq 1$.

The approximate average is the one which is one of the last n input data whose value is smaller than and closest to the integral part of the real average.

$$Xappr_j = \begin{cases} Xappr_j = Xavg_j \\ \quad \text{if } Xavg_j \in XS \\ X_i \,\middle|\, (X_i \in XS) \textit{ and } (X_i < Xavg_j) \text{ and } (Xavg_j - X_i \text{ is minimal}) \\ \quad \text{if } Xavg_j \notin XS \end{cases} \quad \text{.................} (3)$$

where $Xappr_j$ is the value of the jth approximate average.

The output will be created by formula:

$$Y_j = \left\lfloor \frac{\sum_{i=j}^{j+n-1} (X_i + Xappr_j)}{n-1} \right\rfloor \quad \text{.................} (4)$$

where Y_j is the value of the jth output data

Description dataflow of the circuit

```
reg     [71:0]  ValX;
reg     [11:0]  add_sum;
```

<u>Step1:</u>

In this step, there are 2 registers that are ValX and add_sum. The ValX contains input values of X and the bit-size of ValX is 72 bit corresponding to 9 values of X. The add_sum is a register to remember the sum of the X values.

```
always @(posedge clk) begin
    if (reset == 1) begin
        ValX        <= 0;
        add_sum     <=0;
    end
    else begin
        ValX        <= ValX << 8;
        ValX[7:0]   <= X[7:0];
        add_sum     <= sum;

    end
end
```

In this picture, showing the computational system is reset by asserting reset signal for 2 periods. And the registers will begin to save data in the positive edges of the clock.

Step2:

```
assign sum  = add_sum + {4'b0, X} - {4'b0, ValX[71:64]} ;
assign Xavg = add_sum/9;
```

After the edge of clock is positive and reset has negative value, I designed an addition to execute sum all values in ValX, from that I use directly divide to calculate average value.

Step3:

```
always @(*) begin
    Xapp        = 0;
    Last_X      = ValX;
    for(i=0; i<9; i = i+1) begin
        if  (i) Last_X = Last_X >> 8;
        if  (Xavg >= Last_X[7:0])
            Xapp_compare1   = Xavg  - Last_X[7:0];
            Xapp_compare2   = Xavg  - Xapp;
            Xapp            = (Xapp_compare1 < Xapp_compare2) ? Last_X[7:0]:Xapp;
    end
end
```

To calculate the approximate average value, I also add more another block for calculating. In this block, I use a temporary variable to compare the prior value to look for the value approximate average.

Step4:

When we have the approximate average value, we calculate the output value by the following formula:

```
assign Y    = (add_sum + 9*Xapp)/8;
```

I also edit the Clock Cycle is 98 in Test_bench file to decrease gate-level simulation time in post-sim.

```
`timescale 1ns/10ps
`define CYCLE  98         // Modify your clock period here
```

*Scoring = (Total logic elements + total memory bit + 9*embedded multiplier 9-bit element) × (gate-level simulation time in ns)*