Overview

In order to monitor the effect of global warming, the National Oceanic and Atmospheric Administration (NOAA) plans to deploy a huge number of IoT (Internet of Things) motes (i.e., sensor nodes) throughout the US to monitor the atmospheric temperature. You are asked to design a circuit which takes the temperature readings from a temperature sensor in each mote and calculates the average and the standard deviation of the readings.

Problem Statement and Solution

The problem is to find the **moving average** and the **moving standard deviation** of the last 10 temperature readings from the temperature sensor. Suppose that there are $n$ temperature readings $T_1, T_2, \cdots, T_n$ so far. If $n \geq 10$, then the moving average is

$$T_{\text{avg}} = \frac{1}{10} \sum_{i=n-9}^{n} T_i$$  \hspace{1cm} (1)

and the moving standard deviation is

$$\sigma = \frac{1}{10} \sum_{i=n-9}^{n} (T_i - T_{\text{avg}})^2 = \sqrt{\frac{1}{10} \left( \sum_{i=n-9}^{n} T_i^2 \right) - T_{\text{avg}}^2}$$  \hspace{1cm} (2)

If $n < 10$, then the moving average and standard deviation of all $n$ readings should be computed instead.

Design Specification

Assume the temperature sensor has a sensitivity of $(2^{-5})$ °F and has a range of 0 °F to 100 °F. The output of the temperature sensor is represented as a 12-bit unsigned integer. For example, an output of 011000110100 represents $2^{10} + 2^9 + 2^5 + 2^4 + 2^2) \times 2^{-5} = 49.625$ °F

The inputs and outputs of the circuit are specified in the block diagram below:
The circuit is initialized (i.e., forgetting all previous temperature readings and starting from scratch) when RESET is set to 1. The circuit reads in one temperature reading at a time. When the circuit is ready to read in a new temperature, it will set SAMPLE to 1. The circuit will then read in a new temperature reading TN and a mode of operation MODE at the next clock edge. After that, the circuit will spend one or multiple clock cycles to process the readings. When the processing is done, the circuit will set DONE to 1 and output either the moving average temperature (if MODE=0) or the moving standard deviation (if MODE=1) through AVG/SD.

We need to take the square root of the variance to find the standard deviation in Equation (2). Square root is not easy to implement in hardware. In this project, we will approximate the square root of variance \( V \) by one iteration of the Babylonian method described below:

\[
\sqrt{V} \approx \frac{1}{2} \left( \sigma + \frac{V}{\sigma} \right)
\]  

where \( \sigma \) is a guess of \( \sqrt{V} \). For example, let \( V = 810000 \). To find \( \sqrt{810000} \), suppose we guess that \( \sqrt{810000} \approx 1000 \). Therefore, we set \( \sigma = 1000 \). Then \( \sqrt{810000} \approx \frac{1}{2} \left( 1000 + \frac{810000}{1000} \right) = 905 \), which is quite close to the exact value \( \sqrt{810000} = 900 \). In this project, to compute the current moving standard deviation, we set the value of \( \sigma \) to the moving standard deviation computed when MODE was last set to 1. At the beginning before any moving standard deviation is computed, we will use \( \sigma = 010000000000 \) (i.e., 32 °F).

To reduce the hardware cost, you should minimize the use of division. When division is performed, the result should be rounded to the nearest integer. To reduce the error due to rounding, the expressions should be manipulated so that divisions are performed as late as possible. If divisions are performed too early, the rounding errors will be propagated to and amplified by later computation steps.

You have freedom to choose your design style and even to make changes to the actual design specification (get feedback first on any change you wish to make). We reserve the right to make adjustments to the project description as necessary.

**Team Formation**

Students may work in teams of two or three. For a team with more students, the expectation of the final product will be higher.

**Project Requirements**

The basic requirement of the project is that the design should function correctly. However, if a functional but unoptimized design is submitted, a passing but not high score will be given. Techniques to optimize the timing, area and energy
consumption of the design should be applied in order to get a high score. Each team should decide how much / which optimization will be performed.

Note that the power consumption by itself is not a good objective to optimize. The power of a circuit can be reduced easily by breaking down the computations into more clock cycles or by using a slower clock. However, the circuit does not really become more efficient as a result. Instead, the energy consumption per temperature reading, which equals (power consumption) x (# of clock cycles required per temperature reading) x (clock cycle time), is a much more appropriate optimization objective. The IoT motes will run on battery. The battery life is inversely proportional to the energy consumption per temperature reading. Minimizing the energy consumption per reading can extend the battery life.

All projects should be coded in Verilog, properly verified by ModelSim using the given testbenches (and perhaps some additional testbenches), synthesized using RTL Compiler and laid out using Encounter. Reports on timing, area and power of the final design should be generated and discussed. Please do not change the default switching activity parameter when performing power consumption analysis.

You are advised to carefully study the testbenches to ensure that you understand the requirements of the project before you start designing the circuit.

**Project Demonstration and Report**

A 15-minute project demonstration needs to be performed by each team. The main purpose of the project demo is to explain the design to the instructor.

Besides, one project report needs to be submitted by each team. A formal final report is required. The formal report must be typed neatly and hand-writing is not acceptable. All graphs and figures must be labeled. All references must be properly cited. The following sections must be included in the report:

1. **Cover Page**
   - Date
   - Project title
   - Group Member Names

2. **Introduction**
   - Describe the problem (do not simply take materials from this project description).
   - Briefly summarize what your team has done.

3. **Detailed description of the design**
   - Describe design styles used, how the design works, any special design features that you may have, etc.
   - Include schematics and layouts of the final design.
   - Include the timing, area and power reports by both RTL Compiler and Encounter.
   - **Add a table summarizing the following of your design both after synthesis and after layout:**
• clock period (in ns)
• slack (in ns)
• area (in \( \mu m^2 \))
• average power consumption (in mW)
• average energy consumption per temperature reading
• latency (i.e., the # of clock cycles it takes to generate AVG/SD after SAMPLE is asserted)
• throughput (i.e., the maximum # of temperature readings the circuit can handle in 1 second)

4. Verilog codes of design
   • Print the Verilog codes of all modules in your design.

5. Test results
   • Include testbenches and results.

6. Conclusions and Discussions
   • Report on difficulties and learning experiences
   • Provide feedback on the project experience and any suggestions

**Grading Criteria**

Grading is based on the following criteria:

1. Completeness – does project meet requirements?
2. Functionality – has it been tested thoroughly? does the final design work?
3. Optimality – does the design achieve the best performance using the smallest area, energy, time, etc.?
4. Modularity – was the project designed in small blocks that can be repeatedly used?
5. Design methods – were proper design techniques employed?
6. Report – is the report thorough, organized, and complete?

The above list is a qualitative description of the grading criteria. Your final project score will reflect your effort in meeting the above criteria.

**Important Dates**

Important dates are listed below:

• The team composition must be submitted to the instructor by email *on or before* your lab time in the week of Nov. 5-9, 2018.
• The project report (hardcopy) will be due on Dec. 6, 2018 in class. Please also submit a copy of the project report and all the design files (Verilog codes, test benches, scripts, constraint files, etc.) to Canvas on this day.
• The project demonstration will be done around Dec. 6, 2018. The instructor will talk about the schedule for demonstration time a few days in advance.