ECE 4530 Semester Project
"HW-SW Codesign Challenge"

Due: Tuesday **4 December** not later than 11:59 pm

Points: 100

Honor Code: You are reminded that the Hokie Honor Code applies. It is a violation of the Honor Code to discuss procedures for performing the below actions, or discussing the results of this assignment with members who are not in your immediate group.

References:

[a] BIG Cipher Specification, 10/21/2018
[b] Required C source code in Project Folder on CANVAS

Objective: To develop software, hardware, and mixed hardware-software codesigned implementations of a simple cryptographic block cipher. The pure software implementation will be optimized for **latency**, the pure hardware implementation will be optimized for **area**, and the hardware-software codesigned implementation will be optimized for **throughput-to-area ratio**.

Procedure:

1. Generate a "Hello World" Nios 2 configuration, consisting of Nios 2 processor, on-chip memory, and JTAG UART controller; configure per "Hello World" demonstration. Instantiate 8192 (8K) bytes of RAM for the on-chip memory. Ensure to use the "Hello World Small" template.

2. Integrate the same or improved custom timer that you developed for Mini Project 6. It should meet the following specifications:

    a. The timer should count clock cycles from 0 to $2^{32} - 1$.

    b. The user should be able to control the timer using 3 commands:

        (1) Start: Upon a command of "start," the timer commences incrementing.

        (2) Stop: Upon a command of "stop," the timer ceases incrementing.

        (3) Reset: Upon a command of "reset," the timer resets to zero.

    c. The "reset" command should only be accepted when the timer is stopped.

    d. Additionally, there should be a way for the user to read the current value on the timer at any time.

    e. Upon overflow, the timer should roll over to zero; no other action is required.

3. Integrate the timer with the Nios 2 in .qsys as an Avalon Memory-mapped Slave (not as a Nios 2 custom instruction!). Note: You can use one or more Avalon MM slave interfaces.

4. All three implementations of your project should use the exact same source code (included in the project folder), and follow the mandatory conventions for definitions and function prototypes. You will develop three (3) implementations of the BIG cipher as follows:

    a. A purely software implementation. The software version should be optimized for **latency**, i.e., the minimum number of clock cycles required for encryption + decryption.

    b. A purely hardware (HW) implementation. The HW implementation will be called from inside the encrypt() and decrypt() functions, using an interface of your choice. ALL computation must be performed in hardware, and OUTSIDE of the Nios 2 processor. This implementation will be optimized for **area**, expressed in ALMs. Caveats:

        (1) The hardware implementation must achieve at least a 5.0 factor speed-up over the latency in the pure software implementation.
        (2) The use of the Hard Processor System (ARM) is prohibited.
        (3) All computation must be performed on a single DE1 SoC; no off-board computation is permitted.

    c. The final implementation will be a combined HW-SW codesigned implementation. In this version, you will perform transformations that can be efficiently parallelized in HW, while keeping control processes inside the SW. Since the HW area required for the Hello World Nios 2 processor and peripherals, plus the area required for a pure HW implementation is large, this version should have higher throughput than the pure SW implementation, and less HW than the pure HW implementation. You will optimize for **throughput-to-area (TP/A) ratio**, defined in Mbps/ALM. Throughput (TP) is based on the number of encryptions and decryptions per unit time.

5. The Nios 2 processor must be driven from the 50 MHz clock from PIN_AF14 from the DE1.

Deliverables:

Part I Demonstration – (50 points) You will demonstrate the correct output of all 3 implementations, running in the prescribed SW environment on the DE1 SOC, to the Teaching Assistant (TA) during TA office hours. The TA will enter a test vector of his choosing, and verify desired output. Note: The TA will record your performances during the demonstration. If you subsequently achieve better performance, you can revisit the TA for a re-demo, provided that there is sufficient time.

Part II. Submission – (50 points) Package the below files in a .zip file and submit to CANVAS as follows:

1. (10 points) Completed and commented C source codes for all three software implementations.

2. (10 points) Verilog source codes for the pure hardware implementation and HW-SW codesign implementations.

3. (5 points) A block diagram (hand-drawn, or drawn in Microsoft Visio, or .ppt), for the HW implementation. Any other tools require explicit permission from the instructor. Any block-diagram produced by Intel Quartus is not acceptable. Use the guidance in https://cryptography.gmu.edu/athena/CAESAR_HW_API/Reduced_Complexity_Block_Diagrams.pdf, and as shown in Homework 4 solutions.

4. (5 points) A 1 to 2 page summary of your strategy for your decisions on which functions to locate in SW, and which functions to locate in HW, for your HW-SW codesign. Include an analysis of whether the HW-SW codesigned implementation is computationally or communication-constrained.

5. (10 points) Completion of the following table

	Enc Cycles	Dec Cycles	Enc+Dec Cycles	Throughput [1]	.text[2] (bytes)	ALM	Memory[3] bits	TP/A[4] Ratio
Software								
Hardware								
HW/SW Codesign								

Notes: 1 – Based on encryption and decryption, assume long, continuous messages, Mbps; 2- .text section from .elf (in decimal); 3 – Memory bits from Fitter resource summary during compilation; 4 – TP/A Ratio in Mbps/ALM

6. "Competition" Points - A total of 10 out 100 points will be awarded based on competition. For each category, all results will be ranked "most optimal" to "least optimal," and then prorated based on the number of submissions. For example, the most optimal project would receive 3 points in a category, and the least optimal would receive 0. The categories are as follows:

    a. (3) Lowest combined encryption + decryption latency for the pure SW version.
    b. (3) Lowest area in ALMs for the pure HW version.
    c. (3) Highest TP/A ratio for the HW-SW codesigned version.
    d. (1) Most innovative solution (Instructor discretion) (*).

* Only one project will receive this point

Additional grading criteria:

1. Part II deliverables be submitted electronically through CANVAS. The project is due not later than 11:59 pm on 4 December. There will be a 4% reduction per day for the first five late days (i.e., 5 – 9 December), followed by a 20% reduction per day for all remaining days.

2. All team members (up to 4 students) will generally receive the same grade on the project. However, the professor reserves the right to assign differing grades, if determined that project submissions are a result of unequal contributions.