ECE 341 Digital Systems Design
Fall 2015
(draft)

Project Description.

In order to protect the security and privacy of information. Encryption ordinarily requires an encryption key or keys that are used to encrypt and decrypt the data. Encrypted data is generated by an encryption algorithm that takes a key and "scrambles" the original information using an encryption algorithm to make it extremely difficult to infer the original information from the encrypted information. The encryption algorithms are mathematical processes that, in many cases, can be shown to have mathematically provable security.

The encryption scheme considered this semester is the Advanced Encryption System (AES) with 256 bit keys using cyber feedback mode. More information can be found on this algorithm on its Wikipedia page and more information will be provided in class. AES is the encryption standard adopted by the US government.

Input and Output

The input to the AES can be any stream of data, whether text, audio, or video. Each independent stream has its own dedicated encryption key, inputs, and outputs. AES must operate as quickly as possible, trying to minimize the latency between when inputs are supplied and the encrypted information is output. You must also support decryption.

Project Requirements

For your project, initially develop a behavioral model to serve as your Gold model. You should also design your Gold model to have the same interface that you anticipate using when you deploy your system. Once you have implemented and tested your Gold model, develop first the partitioned/ dataflow model and then the structural level model, comparing each to the Gold model.
More specific project requirements are as follows:

1. AES has been around for a while and implementations do exist. For this project, we will build the encryption system from basic principals and not based on an existing implementation.
2. Your design must support both encryption and decryption.
3. Your encryption scheme must implement 256 bit keys.
4. Your encryption scheme must be able to stream data using cypher feedback mode (CFB)
5. Your system should handle as many encryption streams as possible. Imagine you have a high performance system that generates and streams many data streams.
6. The top level entity for behavioral, dataflow, and structural models must all be the same and use signals of the **`std_logic`**/**`std_logic_vector`** or related types (e.g. **`unsigned`** and **`signed`** from the **`numeric_std`** library).
7. You may choose one of two technologies to implement your fully structural implementation. Your selection must be unilateral and you cannot selectively use one or the other in different circumstances. The choices are are
   A. The configurable logic block (CLB) that includes LUTs, flip-flops and multiplexers. The precise structure, operation, and performance of the CLB will be presented in class.
   B. Extending the gate library you have been creating this semester. If you choose this option, your gates must be revised to reflect propagation delays and timing specifications described below.

8. You have available memory available in units of 2,048 locations by 8 bits. Use the memory model in Figure 8-15 from your textbook but modify it for the memory size required here
9. The technology supports a minimum clock rate of 20 MHz and a maximum clock rate of 250 MHz.
10. You should design the system to minimize power consumption and cost while achieving the highest possible performance. Use the following metrics:
    $P = F * 0.015 + 12$ watts where F is the clock frequency in MHz.
    $C = 3 * Y + L * 0.002 + M$ is the cost in dollars where Y is the number of inputs plus outputs and is the number of CLBs, and M is the number of memory units.
11. The maximum combined number of inputs and outputs is 80.
12. The maximum number of CLBs available is 1,000,000 or the number of logic gates/flip-flops that are available are 5,000,000.
13. The maximum number of memory units available is 256.

The project requirements are:

1. You must work in teams of 2 or 3. If you work alone, you will forfeit the team assessment portion of your project Deadline to form groups: November 4.
2. Each team must work independently.
3. Provide the interface specifications necessary to meet the requirements
4. Each team will present their behavioral model to the instructor, TA, and class (Nov. 16).
5. Each team will submit their partitioned/dataflow model for a code review. A brief report must also accompany your VHDL code with the format TBD (Nov. 24).
6. Your final submission must include behavioral, partitioned/dataflow, and structural models.
7. Your final submission must include your testbenches and any test data that you used to verify the workings of your models.
8. Your behavioral model will be your Gold model. Your partitioned/dataflow and structural models must be compared against the Gold model in your simulations. You may update your Gold model to reflect a correction and/or expansion of your behavioral model.
9. Provide a performance assessment and implementation complexity (e.g. CLB count, performance, power consumption) for your structural model. The design will be evaluated in part on the performance of your structural model.

Grading:

Teammate assessment	10%
Behavioral model team presentation	10%
Dataflow model code review	10%
Report	40%
Design evaluation	30%

Rubrics will be provided for all project milestones and submissions. In addition, each member of the group must provide an assessment for other team members performance. The criteria will be based on how team members work with others, the extent of the contribution, reliability, etc. You will receive the teammate evaluation forms at least week the project is due.

The main part of your report should describe the design process and results (including timing diagrams)

must be no longer than 15 pages. In the report, include your design and also the timing diagrams that demonstrate the working of your project. You must submit your VHDL workspace either electronically.

Tentative schedule:

Presentations:	November 16
Dataflow model code review:	November 24
Final project submission:	December 4