ECE 545 Project Deliverables

Top-level folder:
`<Circuit_Name>_<Your_First_Name>`

Second-level folders:
`1_assumptions`
`2_block_diagrams`
`3_interface`
`4_ASM_charts`
`5_source_code`
`6_verification`
`7_timing_analysis`
`8_results`
`9_benchmarking` **[BONUS]**
`10_bug_reports` **[BONUS]**
`11_feedback` **[BONUS]**

The recommended content of these folders is described below:

`1_assumptions`

The file `assumptions.pdf`, containing the description of all assumptions (including possible simplifications) that you have made.

`2_block_diagrams`

Third-level folders:
`A_detailed`
`B_simplified`

Re: `A_detailed`

For the detailed block diagram, please follow class examples and a few basic rules given below:

a. name and label all data, status, and control signals in your datapath that appear as signals in your VHDL code
b. do your best to keep your signal names the same or similar to VHDL names and names used in the cipher specification (if necessary, deviate from the names used in the cipher specification, e.g., replace S[2i+1] by S2ip1)
c. label lower level entities (such as "Hash") inside of the respective symbols
d. explicitly label all ports of basic and lower-level entities by placing port names inside of symbols of respective components
e. use arrows to indicate the direction of data flow
f. explicitly label all bus widths, or include a legend that associates thick lines with buses of a particular width, e.g., 128 bits, and all thin lines with single wires
g. use mostly explicit connections using buses and wires; use connections by name only for connecting signals at the same level of hierarchy when they are located far from each other in your block diagram
h. use either “flyovers” to indicate that wires do not touch, or “solder points” (i.e., dots) to show that connections exist

i. use hierarchical block diagrams
j. clearly state the name of each entity at the top of the respective block diagram
k. place all diagrams in one PDF file, or name multiple PDF files, starting with the hierarchy level (1=top), followed by a unique letter identifier, underscore, and the name of an entity (without _Datapath), e.g., 1a_CipherCore, 2a_TEM, 2b_TEA, 3a_Sbox, etc.

Re: `B_simplified`

Please follow

Rules for Reduced Complexity Block Diagrams

posted on the course web page.

Re: both diagrams

A. All diagrams must be submitted in

PDF.

You can draw diagrams using draw.io or MS Visio.

You can also submit scanned versions of handwritten block diagrams.

B. **[BONUS]:** The complete editable versions of block diagrams prepared using draw.io or MS Visio will be rewarded with bonus points. ***Please keep in mind that you still need to submit the PDF versions of all block diagrams.***

`3_interface`

Interface with the division into the Datapath and Controller.

Please clearly mark the width of each input/output bus.

You can draw this diagram using draw.io, MS Visio, or MS Power Point.

`4_ASM_charts`

Comments:

A. If more than one FSM is used, please include a block diagram illustrating communication among all FSMs.

B. As a minimum, please include scans of your handwritten ASM charts and the controller block diagram in PDF.

C. **[BONUS]:** The complete editable versions of ASM charts prepared using draw.io or MS Visio will be rewarded with bonus points. ***Please note that you still need to submit PDF versions of all ASM charts.***

`5_source_code`

a. Please include all synthesizable VHDL files
b. Please include also a file `source_list.txt` listing all files in the order they should be synthesized (bottom-up).

`6_verification`

[THIS PART IS EXTREMELY IMPORTANT AND MAY DETERMINE YOUR FINAL GRADE]

In this folder include:

A. Modified reference software implementation of your algorithm, which you have used to generate test vectors.

B. All test vector files you have used for verification and debugging.

C. All testbenches used to verify your circuit operation at various levels of hierarchy.

D. Report describing:

- Tools used for compiling and running your software implementations and generating test vectors
- Your strategy for verification: order of tests and testbenches used, source and format of test vectors
- Highest level entity verified for functional correctness and the results of its verification

  If the result of any verification was negative (i.e., your circuit did not operate as expected), please describe incorrect behavior and try to explain the possible sources of errors.
- Verification of lower-level entities (optional if the top-level unit was verified to work correctly):
  - Name of an entity
  - Test vectors and their source
  - Testbench used for verification
  - Result of verification, correct or incorrect behavior
  - Possible sources of errors.

`7_timing_analysis`

Comments:
A. In the file
`timing.pdf`
please provide the exact formulas for

a. Key Setup Time, expressed in clock cycles

b. Encryption Time as a function of the number of plaintext blocks, n, and the number of associated data blocks, m, expressed in clock cycles.

c. Decryption Time as a function of the number of ciphertext blocks, n, and the number of associated data blocks, m, expressed in clock cycles.

d. Hash Time as a function of the number of hash message blocks, n, expressed in clock cycles.

e. Time between two consecutive input blocks (in clock cycles), during processing of

    i. associated data blocks
    ii. plaintext blocks
    iii. ciphertext blocks
    iv. hash message blocks (only if your algorithm supports hashing)

f. Throughput for long inputs as a function of the clock period, T (calculated as a ratio of the input block size and the time between inputting two consecutive input blocks) for processing of

    i. associated data blocks
    ii. plaintext blocks
    iii. ciphertext blocks
    iv. hash message blocks (only if your algorithm supports hashing).

B. Please do your best to confirm the execution time of your circuit through simulation.

Please clearly indicate in `timing.pdf`, which formulas in your report have been confirmed through functional simulation, and which circuit was used in these simulations (entire core, datapath, other?).

`8_results`

Please submit the document `results.pdf` containing the results for the highest-level entity verified for functional correctness.

Please use the following family of FPGAs to generate your results:
Xilinx Artix-7.
Preferably, use the following device (if your design fits within this device):
Device: xc7a12t
Package: csg325
Speed: -3
Full name: xc7a12tcsg325-3.

Include in your deliverables at least the following results:
A. resource utilization
    a. LUTs
    b. Slices
    c. Flip-flops
    d. LUT-FF pairs

e. BRAMs
f. DSP units
g. I/O pins

B. numerical values of the following timing parameters
   a. Minimum Clock Period
   b. Maximum Clock Frequency
   c. Maximum Throughput for processing of a large number of
      i. associated data blocks
      ii. plaintext blocks
      iii. ciphertext blocks
      v. hash message blocks (only if your algorithm supports hashing)

   based on the minimum clock period after placing and routing.

C. analysis of the obtained results; your observations and conclusions.

Clearly indicate the name of the entity for which these results were obtained.

`9_benchmarking` **[BONUS]**

- Results optimized using Minerva
  [see https://cryptography.gmu.edu/athena/index.php?id=Minerva]

- Excel graphs and charts summarizing all results

- Analysis of the obtained results; your observations and conclusions.

`10_bug_reports` **[BONUS]**

In this folder, please include the following subfolders

```
1_specification
2_SW
3_Minerva
4_Other
```

Each bug report should be placed in a separate subfolder of these directories, and named:

`bug_<n>_mm_dd`, where
`mm_dd` represents the month and day when a given problem was discovered, and
`<n>` is a unique bug number.

For each bug you report, please include short description.

In
`1_specification`
please describe all mistakes you have found in the specification of the algorithm assigned to you.

These mistakes can include mistakes in the pseudocode, figures, tables, explanations, etc.

In
`2_SW`
please describe any mistakes, inefficiencies, and/or discrepancies (compared to the specification) you have found in the reference implementation of your circuit.

In
`3_Minerva`
please include all information necessary to recreate the same problem (in particular, source files, version of Vivado used, etc.).