University of Pennsylvania
Department of Electrical and System Engineering
System-on-a-Chip Architecture

ESE532, Fall 2018 — Design and Function Milestone — Wednesday, October 31

Due: Friday, Nov. 9, 5:00PM

Group: Develop functional code. Identify design space options.

Individual: Writeup is an individual task.

1. Identify major design space axes that could be explored for your implementation.

   - For this milestone, aim for breadth (quantity of options)
   - Each axis description can be 2–3 sentences. Identify challenge being addressed, basic solution opportunity, and continuum. A single point in the design space is not a continuum; except in rare cases, this should capture a range of potential parameter values.
   - Include a simple equation to illustrate ideal benefit (e.g., running N tasks in parallel reduces runtime by a factor of N; $T(N) = T(1)/N$).
   - Cover all operations that must be accelerated.
   - Aim for at least 6 axes per operation. Identify a few associated with how operations interact with each other.
   - Some of this should build on the parallelism opportunities you identified on the previous milestone.

   Example from FFT design discussed in class.

Axis:	P, number of butterfly units.
Challenge:	Improving the throughput of the FFT
Opportunity:	Implement multiple hardware butterfly datapath units.
Continuum:	This can range from 1 to a fully spatial design with $P = \frac{N}{2}\log(N)$ butterfly units.
Equation for Benefit:	$Throughtput(P) = P \times SingleButterflyThroughput$

2. Develop a functional implementation for the project task that can run on the Zynq ARM and produce a valid compressed output stream that works with the supplied decompressor. This does not need to be integrated with I/O. It can compress from SDCard input to SDCard output.

    - The primary goal for this assignment is functionality. As such, you should focus on a simple design that captures the necessary behavior.
    - As a result, this design need not be efficiently synthesizable to hardware.
    - However, you will eventually be optimizing this design and likely exploring HLS mappings to hardware. So, given a choice, you might want to use design constructs and idioms that you know will be more amenable to HLS hardware mappings.
    - Alternately, you should be prepared to rewrite your code later for efficient hardware mappings.
    - **Hint:** rather than getting all of the operations perfect and then integrating them and testing with our decoder at the end of the week, you might want to put together simpler versions of the operations first. That way, you aren't waiting on the debug of one hard module to start intgeration. You can integrate and debug interface problems earlier in the week, and then replace the inner computations as you get them worked out. For example, you might use addition modulo 2^{64} as a placeholder hash for SHA for initial integration, then replace with the proper SHA-256.

3. Turn in a tar file with your functional code to the designated assignment component in canvas. This should be code for the full project, so all 3 team members can turn in a copy of the same tar file.

4. Document your design. This writeup should be comprehensive across the project group; it should **not** be limited only to the subset that was performed by an individual.

    (a) Code sources (e.g., URLs) for any open-source code you used as a starting point or as a primary reference

    (b) Current compression ratio and breakdown of contribution from deduplication and from LZW compression; current throughput achieved for full task.

    (c) Description of all validation performed on your current functional implementation.

    (d) Description of who did what. How did your team collaborate on the design, implementation, and validation?

5. Identify any challenges your group had in collaboration and design integration this week and how you plan to address them for future weeks.