

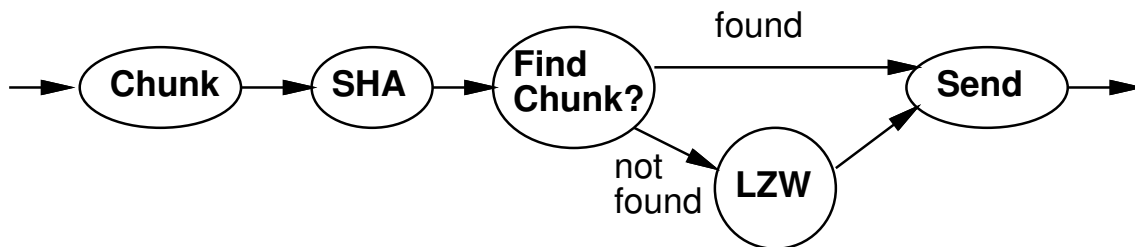
University of Pennsylvania
Department of Electrical and System Engineering
System-on-a-Chip Architecture

ESE532, Fall 2017 Deduplication and Compression Project Wednesday, October 25

Due: Friday, December 8, 5:00PM

1 Goal

Develop a pipeline to compress the data transferred between computers using deduplication and compression that can operate at modern ethernet speeds (1Gb/s, 10Gb/s). Specifically, we'll look at Content-Defined Chunking to break the input into chunks, SHA-256 hashes to screen for duplicate chunks, and LZW compression to compress non-duplicate chunks. Your minimum goal for implementation is to achieve 1Gb/s, but you should try to run as fast as possible, up to 10Gb/s. In any case, you will sketch the solution to scale to 10Gb/s even if you cannot implement it.



This is a 5 week project assignment; the intent is to allow you to plan and execute a significant, open-ended design exploration and mapping. You will not achieve the implementation goal or the course learning goals by trying to do this in one week. We give you milestones to help provide some structure, but the milestones are minimal and doing the minimum to hit the milestone each week will be insufficient to get you where you need to be at the end. We are giving you flexibility in planning and ordering rather than lock-step specifying exactly what you need to do each week.

- Project work is done in teams of 2. You select partners during first week.
- Collaboration between teams is limited as specified on the [course web page](#).
- Individual milestone and final writeups, including individual solution of exercises.
- The spirit of this exercise is to optimize the SoC mapping of the algorithm. As such, explorations of alternate solutions that change the algorithms and generally optimize the solution for hardware and software are out of scope. Explorations that tweak or tune the algorithms slightly to better exploit the SoC hardware are potentially in scope.

2 Final Report

- Describe your single ARM processor mapped design. [1 page]
 - Key parameters in the solution
 - Performance achieved and energy required
 - Characterization and breakdown of time spent in the major components
- Describe your final Zynq XC7Z020 mapped design. [5 pages]
 - Performance achieved and energy required
 - Key design aspects: task decomposition, parallelism, mapping to Zynq resources, include diagrams to support
 - Model to explain performance, area, and energy of design
 - Current bottleneck preventing higher performance
- Describe design that achieves 10Gb/s and estimate chip area and costs. [3 pages]
 - Key design aspects. Build on description in first section, describing differences for this design from the Zynq XC7Z020 design
 - Use or expand model introduced in first section to show how you meet the 10Gb/s encoding problem
 - Custom implementation area estimate with area model support
- Describe how you validated your implementations. [1 page]
- Describe the key lessons you learned from this design experience. [1 page]
- Describe design space explored and show graphs and models to support design selection. [any number of pages as needed]
- Describe who did what. [1 page]
- Submit a tar file with your final design; there will be a separate project assignment link for your code submission.

3 Milestones

We will provide precise requirements for milestones each week. These may include a few exercises to help prepare you for questions that may be on the final in addition to the project specific components. Milestones and feedback feed into the final report. In most cases, the milestones can serve as a first draft of a component of your report, and the feedback we give you will help provide guidance on how to refine it for the report.

- Analysis, parallelism, and teaming [11/3]
- Functional pieces and design space [11/10]
 - (functional should work with provided decoder, but may have dummy or simplified elements for parts of the design.)
- Fully functional version and energy estimate [11/17]
 - (intervening lectures and this milestone specification will clarify what we want for energy estimate)
- 1Gb/s design and custom area estimates for that design [12/1]
 - (intervening lectures and this milestone specifications will specify area model)
- Final Report [12/8]

4 Components

The components we will use are standard enough that the wikipedia pages are useful, and there are several other nice tutorial blog posts out there. Here's a roundup of starting points.

- Content-Defined Chunking (Rabin Fingerprint)
 - <https://moinakg.wordpress.com/tag/rabin-fingerprint/>
 - <https://restic.github.io/blog/2015-09-12/restic-foundation1-cdc>
 - https://en.wikipedia.org/wiki/Rabin_fingerprint
- SHA 256 Hashing
 - <https://tools.ietf.org/html/rfc6234>
 - <https://en.wikipedia.org/wiki/SHA-2>
- Lempel-Ziv-Welch Compression
 - <http://www.geeksforgeeks.org/lzw-lempel-ziv-welch-compression-technique/>
 - <https://en.wikipedia.org/wiki/LempelZivWelch>

5 Some Suggested Parameters

There will be some discretion in picking implementation parameters. From the start we will suggest considering:

- 4KB average chunk size with a 8KB maximum
- Use on-board DRAM for the chunk dictionary and hash fingerprints
- Full, maximum chunk size as the LZW compression window

We may refine our suggestions as we get more experience, and you may want to experiment with some of the parameters when tuning your implementation.

6 Examples of Use

1. Yan Zhang, Nirwan Ansari, Mingquan Wu, and Heather Yu. “On Wide Area Network Optimization.” In *IEEE Communications Surveys & Tutorials*, vol. 4, issue 4, pp. 1090–113, Oct. 2013. <http://ieeexplore.ieee.org/document/6042388/> Sections III A and B survey the role of compression and decompression in optimizing WAN data traffic.
2. Athicha Muthitacharoen, Benjie Chen, and David Mazieres. 2001. A low-bandwidth network file system. In *Proceedings of the eighteenth ACM symposium on Operating systems principles (SOSP '01)*. pp. 174-187. <https://dl.acm.org/citation.cfm?id=502052> Use of deduplication for optimizing a file system operating across a low bandwidth link.