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

ESE532, Fall 2019 — 200Mb/s and Area Milestone — Wednesday, November 13

Due: Friday, November 22, 5:00PM

Group: Achieve target speed and writeup progress (Items 1, 2, 3)

Individual: Area estimate including writeup (separate, individual turn in for Item 4)

1. Achieve 200 Mb/s on deduplication and compression task
   (a) Report throughput achieved. Include details on the throughput supported by each major operation as well as the overall throughput.
   (b) Report current compression status.
   (c) Describe all validation performed on your accelerated implementation.
   (d) Identify where this design is in your design space. Explain additional design-space axes beyond your previous milestone as necessary.
   (e) Describe the techniques you used to achieve the speedup. Be clear where each component runs and the resources it uses.
   (f) Support your description with a performance model.
   (g) Description of who did what.
2. Turn in a tar file for your design above to the designated assignment component in canvas.
3. The tar (or zip) files should package up the `projectdir/debug/sd_card` directory for your encoder implementation.
   - Your encoder should take inputs from the ethernet link.
   - Your encoder should store the encoded result in `/compress.dat` on the SD Card.
4. For your 200 Mb/s milestone design described above, estimate the area of a custom design using the area model on the following page.

Area Model

- Model here is of a custom design (not the area of the programmable logic to hold your design) in a 28 nm CMOS process.
- Use a simple sum of components area model ($A = \sum_i A_{component_i}$).
- Only include components you use (so, for example, if your current solution only uses one ARM core, only count one; if you use more than one, count 2, 3, or 4 as appropriate.)
- Use CACTI for estimating memories [1]. Estimate memories as custom memories of the organization you actually use (so, for example, if you use an 8K×8, single ported RAM in your accelerator, use CACTI to estimate that memory instead of estimating the area as 2 dual-ported 36Kb RAMs (as you would be using in the Zynq Programmable Logic). You can find a version of CACTI on the `eniac` file system in: `/home1/e/ese532/cacti/cacti` (or the source in `/home1/e/ese532/cacti.tar`, if you want to download and build on your own machine). Use the 32nm technology node with ITRS-LOP devices (as illustrated in the sample configuration files in `/home1/e/ese532/cacti_examples`). A sample CACTI run is invoked:
  `/home1/e/ese532/cacti/cacti -infile armca53_l2.cfg > armca53_l2.out`
- Fixed area below is intended to capture area that should be the same in any implementation (unchanging as you change the resources for computation and memory), including: DRAM and FLASH interface, I/O and power pads, clocking, and reset.
- ARM Cortex-A53 area includes L1 caches and neon. Area does not include L2. Model L2 using CACTI (`armca53_l2.cfg` configuration provided).

Unit	Symbol	Area (mm^2)
Fixed Area	A_{fixed}	10
ARM Cortex-A53	A_{arm}	2.0
Logic in one 6-LUT	A_{lut}	3.0×10^{-5}
DSP Block	A_{dsp}	0.01
$n \times m$ Fixed-Point Multiplier	$A_{mpy}(n, m)$	$n \times m \times 10^{-5}$
8-Channel DMA Engine	A_{dma}	0.1
AXI Crossbar with i input and o output 64b ports	$A_{xbar}(i, o)$	$(i + o) \times 10^{-2} + i \times o \times 10^{-3}$

References

[1] Naveen Muralimanohar, Rajeev Balasubramonian, and Norman P. Jouppi. CACTI 6.0: A tool to model large caches. HPL 2009-85, HP Labs, Palo Alto, CA, April 2009. http://www.hpl.hp.com/techreports/2009/HPL-2009-85.html; latest code release for CACTI 6 is 6.5.