

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

ESE532, Spring 2017

4× Speedup and Area Milestone

Friday, March 31

Due: Friday, April 7, 5:00PM

Group: Achieve speedup, identify components for area model.

Individual: Calculate area and writeup.

1. Achieve 4× speedup

- Report runtime and speedup achieved. Include details on the speedup achieved per component as well as overall speedup.
- Identify where this design is in your design space. Explain additional design-space axes beyond your previous milestone as necessary.
- Describe the techniques you used to achieve the speedup.
- Support your description with a performance model.

2. For your 4× 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 both, count two.)
- 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 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 armc9_12.cfg > armc9_12.out
```
- For floating-point units and multipliers, if possible, model as custom units using the areas below rather than by LUT and DSP usage.
- 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-A9 area includes L1 caches and neon. Area does not include L2. Model L2 using CACTI (`armc9_12.cfg` configuration provided).

	Unit	Symbol	Area (mm ²)
	Fixed Area	A_{fixed}	10
	ARM Cortex-A9	A_{arm}	1.0
	Logic in one 6-LUT	A_{lut}	5.0×10^{-5}
	DSP Block	A_{dsp}	0.01
	Double-Precision Floating-Point Unit	A_{dpfpu}	0.032
	Single-Precision Floating-Point Unit	A_{spfpu}	0.018
	$n \times m$ 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.