University of California at Berkeley
Department of Electrical Engineering and Computer Science
Reconfigurable Computing Seminar

CS294-7, Spring 1997 — Project 1: Assignment — Tuesday, February 4

Project 1: Implement a computational function in space on a conventional FPGA.

Dates

Assigned:	February 4
Proposals:	February 13 (description due by class time)
Due:	March 6 (HTML report pointer to `amd@cs` by class time)
Talks:	March 11

Goals

- **Primary Goal:** Gain experience with spatial implementation of computations
- **Secondary Goals:**
  - Gain experience designing/using FPGAs with traditional design flow
  - Expand datapoints for FPGA/spatial implementation

Scope

This is primarily an exercise to gain implementation experience. We don't expect you to come up with new algorithms or architectures, we but do expect you to look into a spatial implementation scheme for some problem and realize it in an FPGA.

Pick a medium-to-small computational task/kernel. This should be bigger than a simple, linear datapath operator (*e.g.* comparator, adder) but smaller than an FFT. A systolic cell which combines a few primitive operators and control such that it can be cascaded into a larger array is probably about the right level. The implementation should be something you can do in a week once you've decided what to do and become accustomed to the CAD tools. The SPLASH sequence comparator which we will see in lecture 6 (Chapter 8 of [BAK96]) or Leiserson's priority queue [Lei79] (which we'll see in lecture 9) are probably about the right level of complexity. It is understood that you won't have time to try out several different implementation styles for your task—pick one and do the best you can with it. We've included several suggestions for tasks at the bottom of this handout, but are not limiting you to this group. You may have encountered particular computational kernels in your previous work which you would like to try implementing spatially, and we encourage this.

Anticipated schedule:

2/4-11	think about potential project, read couple of papers
2/11-13	Pick task and implementation style, write proposal
2/13-20	Familiarize self with FPGA tools
2/20-28	Build implementation
3/1-6	slippage time (writeup report)

We do expect you to make an effort to take advantage of the spatial parallelism available. In the spirit of the assignment, you should try to use the FPGA device efficiently – minimizing cycle time, maximizing the useful work performed by the device per cycle, and minimizing the resources required by the task. We encourage you to think about the spatial aspects of your design and perhaps do key floorplanning manually.

The project should be simple enough that synthesis is not necessary. If you do choose to use synthesis tools in the preparation of your design, we recommend you use them more as a design advisor rather than as a substitute for dealing with the mapped logic. *i.e.* Look at and critique the mapped logic, learning from it if the mapping is superior to your own and replacing it with hand mapped logic if it is shortsighted and inefficient. In any case, you are responsible for understanding where delays are going in your design and how it should be floorplanned so you cannot ignore the details of the mapped computation.

Target FPGAs

The XC4000E series devices, as discussed in lecture 4, will be your implementation target. We do not require that you run your design on an actual device, so the choice of a particular family member is less important. For concrete mapping you should probably use the smallest device which will conveniently fit your design (or a reasonable number of instances of your design). Evaluation boards with XC4005E components are available should you desire to run your design on a real component.

Project Proposal

Your project proposal should let us know what task you have chosen to implement and the basic implementation strategy. This should be a couple of paragraphs long and include the references for your chosen design.

We will review the proposals to give you early feedback on your chosen task (*e.g.* too trivial, too ambitious, additional/alternative approaches and references).

Report

For your report, prepare an HTML page describing your project. This should be about 1000-1500 words and include a few figures to illustrate the design. Following is a recommended composition:

- Describe the problem in one paragraph.
- Provide an implementation overview in 1-2 paragraphs and a few figures. Included references to the papers used for basic scheme or inspiration.
- Give a succinct highlight of the interesting aspects of your implementation in 1-2 paragraphs and a figure or two. (What was interesting? difficult? exhibited cleverness?)
- Characterize your design by the resources used (CLBs), cycle time achieved, task throughput and latency (based on estimates provided by CAD tools) [DeH97]. Show the floorplan of your routed design and account for the delays composing the critical path.
- If you can comment on how your implementation would scale (area-latency, area-throughput curve), that would be nice, but is not required.

In Class Project Talks

We have set aside class time on March 11th for in-class descriptions of the project. We will select half a dozen projects for short presentations on that day.

Sample Selection of Computational Tasks

- Priority Queue [Lei79]
- Sort [Lei92] [TSK87]
- Systolic GCD [BK83] [Jeb94]
- Median filter (more generally, RVF) [KOA90] [RST93]
- Division [LE94]
- Square root [TVG96]
- Radix conversion [Par92]

References

[BAK96] Duncan Buell, Jeffrey Arnold, and Walter Kleinfelder. *Splash 2: FPGAs in a Custom Computing Machine.* IEEE Computer Society Press, 10662 Los Vasqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1264, 1996.

[BK83] R. P. Brent and H. T. Kung. Systolic VLSI Arrays for Linear-Time Computation. In F. Anceau and E. J. Aas, editors, *VLSI'83.* Elsevier Science Publishers, 1983.

[DeH97] André DeHon. Comparing Computing Machines. preprint contact `amd@cs.berkeley.edu`, January 1997.

[Jeb94] T. Jebelean. Implementing GCD systolic arrays on FPGA. In R. W. Hartenstein and M. Z. Servit, editors, *Field-Programmable Logic Architectures, Synthesis and Applications. 4th International Workshop on Field-Programmable Logic and Applications, FPL'94*, number 849 in Lecture Notes in Compuer Science, pages 132–4. Springer-Verlag, 1994.

[KOA90] Mustafa Karaman, Levent Onural, and Abdullah Atalar. Design and Implementation of a General-Purpose Median Filter Unit in CMOS VLSI. *IEEE Journal of Solid-Sate Circuits*, 25(2):505–513, April 1990.

[LE94] Marianne Louie and Milos Ercegovac. Implementing Division with Field Programmable Gate Arrays. *Journal of VLSI Signal Processing*, 7:271–285, 1994.

[Lei79] Charles Leiserson. Systolic Priority Queues. CMU-CS-TR 115, Carnegie-Mellon University, Pittsbugh, Pennsylvania 15213, April 1979.

[Lei92] F. Thomson Leighton. *Introduction to Parallel Algorithsm and Architectures: Arrays, Trees, Hypercubes.* Morgan Kaufmann Publishers, Inc., 1992.

[Par92] B. Parhami. Systolic Number Radix Converters. *Computer Journal*, 35(4):405–9, August 1992.

[RST93] Roberto Roncella, Roberto Saletti, and Pierangelo Terreni. 70-MHz 2μm CMOS Bit-Level Systolic Array Median Filter. *IEEE Journal of Solid-Sate Circuits*, 28(5):530–536, May 1993.

[TSK87] Nobuo Tsuda, Tetsuji Satoh, and Tadamichi Kawada. A Pipeline Sorting Chip. In *1987 IEEE International Solid-State Circuits Conference, Digst of Technical Papers*, pages 270–271. IEEE, February 1987.

[TVG96] V. Tchoumatchenko, T. Vassileva, and P. Gurov. A FPGA based square-root coprocessor. In *Proceedings of the 22nd EUROMICRO Conference*, pages 520–525. IEEE, IEEE Computer Society Press, September 1996.