

Homework # 3

Due: March 24

Multi-FPGA Systems

In this assignment you will have an opportunity to provide brief summaries of important topics regarding multi-FPGA systems and some additional experimentation to validate routability prediction using VPR. If you prefer, you can substitute experiments with commercial FPGA software for the additional VPR experiment.

Ex 1: Rent's Rule

Please read pages 416-421 in [2] and Section 4.2 in [1]. Based on the references, summarize, in your own words, the Rent's Rule equation, its parameters and its implications from a system standpoint. Your description should include a discussion of approaches to overcome pin limitations. What types of circuits are most susceptible to Rent limitations? Is this problem likely to get better or worse as devices scale? Include in your discussion values taken from Table 9.5 in [2] and Figure 9 in [1] to support your arguments. Please limit your discussion to one page at most.

Ex 2: Virtual Wires

Please read the remainder of [1] and summarize the basics of the virtual wires approach. How does this emulation system differ from existing systems? This discussion should cover the following areas: new software steps added to support virtual wires, impact on multi-FPGA partitioning, description of phase-based scheduling, and summary of results reported in the paper. Please limit your discussion to one page at most.

Ex 3: Transmogripher-4 versus BEE-2

Please read [3] and look over [5]. In less than a page, contrast the hardware architectures and programming environments of these systems. How are they similar? How do they differ? You may wish to summarize your results in a table.

Ex 4: Routability Analysis

Please complete either this exercise or the next one. You do not have to complete both. In class we discussed the potential usefulness of FPGA routability prediction prior to FPGA routing. One approach to routability prediction can be found in Section 2.3 of [4]. Using VPR, determine the placement of five different designs for both $N = 1, I = 4$ and $N = 4, I = 10$. Then, determine $W_{estimate}$ for these 8 placements using Equation (4) in [4]. Finally, perform routing of the designs to determine W_{min} . How accurate are your estimates? Provide a brief analysis of your results and include a table of results.

A few suggestions:

- Additional blif files for designs can be found in the problem set 1 software distribution. You should be able to use copies of the makefiles from earlier problem sets to

perform needed compilation for new designs.

- Make sure T-VPack, placement, and routine are all performed using routability-driven mode (not timing-driven). Note that all tools used for problem set 2 used routability-driven mode.
- The placement wire length can be determined from the placement cost.
- VPR default values for place and route parameters can be used for this experimentation. If the VPR router replacement files from problem set 2 are used, *astar_fac* should be set to 0.

Ex 5: Experimentation with Commercial FPGA Software

Please complete either this exercise or the previous one. You do not have to complete both. The benefits of reconfigurable computing are best seen when comparing the performance of an application mapped to both a microprocessor and an FPGA. In this exercise, please map a simple application (e.g. an FIR filter, small sorting unit, crossbar circuit) to both an FPGA using an RTL implementation and to a microprocessor using a high-level (e.g. C) representation. Please be sure the implementations represent the exact same application (e.g. performs the same function, same data bit width, etc). How does the performance of these implementations compare for an example input data set? Please provide the source for both implementations and an FPGA report which indicates the performance. You can determine microprocessor run time using the **time** command, as in problem set 2, or another suitable approach. FPGA run time can be determined via simulation. If possible, please note the silicon process technology (e.g. 90 nm) of both the microprocessor and FPGA you target. How does this affect your results?

A few suggestions:

- You can use any commercial FPGA software targeted to any device family you wish. Note that Altera offers a free web version of their Quartus II software tool. (https://www.altera.com/support/software/download/altera_design/quartus_wc/dnl-quartus_wc.jsp). I suggest using a PC or workstation as your microprocessor target. You can choose any microprocessor and compiler.
- Compile your FPGA design for the peak clock frequency possible. Often, this can be achieved by setting the desired clock frequency to 1 GHz. Although the FPGA software will be unable to achieve this clock speed, it will try its hardest to achieve the best possible design performance. You should also use microprocessor compiler optimizations, if possible.

Given limited course staffing (e.g. one person), you will be responsible for FPGA and microprocessor tool support if you choose this exercise.

References

- [1] J. Babb, R. Tessier, and A. Agarwal. Virtual Wires: Overcoming Pin Limitations in FPGA-based Logic Emulators. In *Proceedings, IEEE Workshop on FPGA-based Custom Computing Machines*, Napa, Ca, Apr. 1993.
- [2] H. B. Bakoglu. *Circuits, Interconnections, and Packaging for VLSI*. Addison Wesley, Reading, Ma, 1990.
- [3] C. Chang, J. Wawrzynck, and R. Broderson. BEE2: A high-end reconfigurable computing system. *IEEE Design and Test of Computers*, Apr. 2005.
- [4] J. Swartz, V. Betz, and J. Rose. A Fast Routability-Driven Router for FPGAs. In *6th International Workshop on Field-Programmable Gate Arrays*, Monterey, Ca, Feb. 1998.
- [5] University of Toronto. *The Transmogripher-4 Project*, 2005. <http://www.cccg.toronto.edu/~tm4/>.