QUILT: A GUI-based Integrated Circuit Floorplanning Environment for Computer Architecture Research and Education*

Gregory J. Briggs, Edwin J. Tan, Nicholas A. Nelson
Electrical and Computer Engineering
University of Rochester
Rochester, NY 14627
{*grbriggs*, *etan*, *ninelson*}*@ece.rochester.edu*

David H. Albonesi
Computer Systems Laboratory
Cornell University
Ithaca, NY 14853
albonesi@csl.cornell.edu

Abstract

*In this paper, we describe a graphic editing tool called QUILT (**Q**uick **U**tility for **I**ntegrated circuit **L**ayout and **T**emperature modeling). QUILT permits users to rapidly build floorplans of integrated circuits, providing both a visual aid as well as an input to the HotSpot simulator. The tool provides numerous features for estimating circuit performance, such as interconnect delay, and for generating graphical images for publications. As a graphical and easy to use tool, QUILT is well suited for both research and coursework purposes.*

1. Introduction

An essential element of computer architecture education, whether at the level of an undergraduate homework assignment or doctoral-level research, is investigating the tradeoffs among multiple design criteria, such as performance, cost, and power dissipation. The most hands-on, real world, avenue for exploring computer engineering tradeoffs is designing, testing, and fabricating different integrated circuits and comparing their characteristics. However, the prohibitively high time and monetary costs of these activities make them useful only for small circuits or long term projects. Hardware emulation via FPGAs provides more rapid turnaround time yet suffers from two major limitations. First, the density of FPGAs significantly lags that of full-custom CMOS designs, making it necessary to span several FPGAs for large, microprocessor-level emulations. Second, the internal hardware structure of FPGAs differs considerably from a full-custom design making it difficult to correlate performance and power measurements from the FPGA to that of the full-custom design.

For these reasons, the use of software simulation for exploring design tradeoffs is very popular in computer architecture research and education, permitting large system investigations to be performed very rapidly. Although not as accurate as real hardware, simulators produce results for large scale systems in a matter of minutes or hours. Thus, many tools, both proprietary and public-domain, have been developed to study the performance, power, and temperature aspects of different architectures. The reasonable accuracy and rapid turnaround times of these tools makes them highly appealing.

However, most popular simulation toolsets are text-based and command-line driven. When used for tasks such as floorplanning, such interfaces are tedious and lead to frequent and hard-to-detect errors. For instance, the input to the HotSpot temperature modeling tool [7, 12] is a text file containing a listing of x,y coordinates and sizes of the functional units. In the past, researchers modified this file using text editors and manually computed each coordinate, a tedious and error-prone approach. Thus, the prime motivation for designing the QUILT tool described in this paper was to provide a more productive means to utilize HotSpot. However, in the course of development, it became apparent that QUILT would also be useful in estimating IC transistor counts, rough floorplanning for very large scale integration (VLSI) layout, producing graphics which can be used in reports and presentations, and as a general educational tool. Research has shown that the use of graphical user interfaces (GUIs) can increase productivity and also help to reinforce concepts learned in the classroom [1, 9].

The rest of this paper discusses the QUILT tool and is organized as follows. Section 2 describes the operation of QUILT in detail. Section 3 gives a brief overview of the technical aspects of the software. Examples of how QUILT can be used in an academic environment are given in Section 4. New features that can be added onto the current version of the tool are discussed in Section 5, and conclusions are provided in Section 6.

*This research was supported in part by National Science Foundation grant CCR-0304574.

2. Detailed Description of the QUILT Tool

QUILT allows one to easily build a floorplan as an input for various simulation tools. The current version is optimized for use with HotSpot. Users can quickly adjust their designs by simply "pushing polygons", and running the HotSpot simulation again. This removes the tedium of manually computing functional unit coordinates and allows users to focus on exploring design issues. Functional units can be easily moved and resized. On-chip interconnects are also simulated in detail.

2.1. General Structure

QUILT is a standalone Java application. The main function of the tool is to generate an input text file for a simulator while viewing or editing a graphical representation of the IC floorplan in a GUI. The tool was tested with a simulator based on SimpleScalar 3.0b [3] with Wattch [2] and HotSpot [7, 12] extensions.

QUILT reads and writes HotSpot floorplan coordinate text files, and also colorizes the floorplan based on power or temperature trace files. The tool can be started from the command line and an existing floorplan file can be specified as a command line option, or it can be given a shortcut icon and associated with `.flp` files similar to most GUI programs.

In addition to a text-based coordinate file, QUILT leverages Java's capability to generate JPEG files to produce layout images that can be used in documents.

QUILT requires various parameters for the technology node of interest. For our research with QUILT, we obtained these from the 2003 ITRS Roadmap [10]. Changing to a different node is a simple matter of replacing some constants with the desired node's "Roadmap" values.

2.2. User Interface

When the tool is started, a GUI window is displayed which has the look and feel of a typical drawing editor. The drop-down menus are located on the top of the window. If an input file was specified at start up, the editing area will display a floorplan of the circuit layout. Figure 1 shows QUILT displaying a sample processor, in this case a modified version of the Alpha 21264 floorplan.

2.3. Floorplan Generation

A floorplan can be created from scratch if desired. The drop-down menus Edit, Mode, Zoom, Select and Generate are used to draw and edit a basic floorplan.

New units can be created by specifying their name and dimensions. SRAMs can be created by choosing a memory size. Level 2 cache can automatically be laid out to surround the core. The Generate menu is shown in Figure 2.

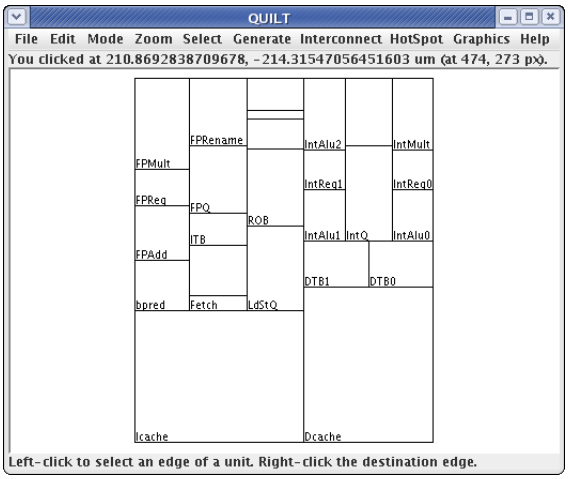

Figure 1. QUILT displaying a sample processor (modified Alpha 21264) floorplan

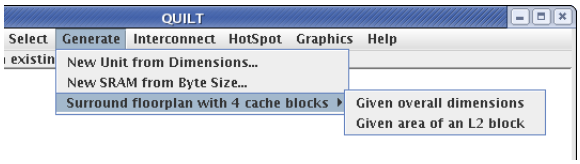

Figure 2. QUILT's Generate drop-down menu

Once units have been created, they can be resized and moved using three possible editing modes. The first mode, Move, simply allows a unit to be translated to new coordinates. The second mode, Resize (constant area), is useful in that one can adjust the dimensions of a unit, while still retaining the original area and thus the same functional capability (for example, to keep the number of bytes of an SRAM constant). Finally, the third mode allows the dimensions to be changed without constraint.

QUILT can compute the transistor count for a particular functional unit and technology node projected by the ITRS Roadmap. Figure 3 shows the pop-up window, obtained by selecting the function in the File menu, displaying this information. This operation is useful in estimating the total number of transistors used in a design.

The Zoom and Select menus make it easy to zoom in to, or select a certain part of, a unit, respectively.

Lastly, the Edit menu (Figure 4) covers typical edit operations, as well as a few extra operations that have proven useful. The Join very close edges operation is especially useful when importing HotSpot floorplans that had been made by hand. These floorplans often contain small calculation errors. Another function is Show overlapping and underlapping points which is useful in verifying that there

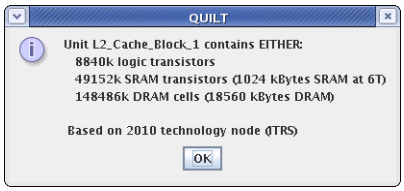

Figure 3. Window displaying functional unit transistor count

are no spaces in the floorplan. A common cause of HotSpot floorplan errors are gaps between unit edges, which act as insulators during temperature simulation.

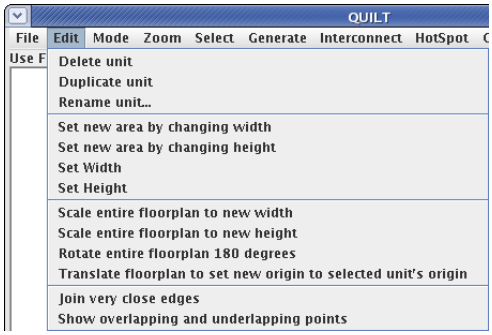

Figure 4. QUILT's Edit **drop-down menu**

To recapitulate information regarding a particular unit, an option called Show Unit Info in the File menu displays a window listing the unit's width, height, area and x,y coordinates.

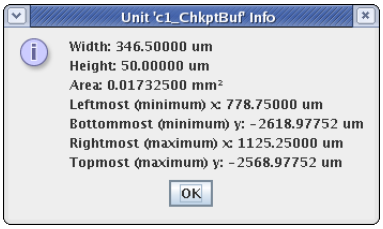

Figure 5. Details of a functional unit in a pop-up window

2.4. Interconnect

Interconnect delay is of growing importance for computer architects. QUILT models multiple types of interconnects, and can be easily extended to other approaches.

Conventional metal interconnects as well as optical interconnects are currently modeled, based on estimates

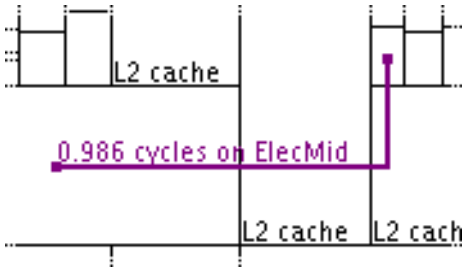

Figure 6. QUILT interconnect delay estimator

from [4]. After two endpoints have been selected for communication, the program presents a list of the estimated delays using each type of interconnect. The user can select the desired type. Electrical connections are automatically routed in a simple Manhattan style. Optical interconnects are modeled as point-to-point links. The area consumed by the interconnect is also depicted (visible for interconnects that are many bits wide). Finally, the connection delay is expressed in terms of clock cycles, for ease of comparison. This is shown in Figure 6.

2.5. HotSpot Usage

The primary file format of QUILT is the `.flp` (floorplan) file used with HotSpot. However, one can do much more with QUILT than just save files for use with HotSpot.

A single menu function takes care of saving the floorplan file, running HotSpot, and depicting the results within the editor. The floorplan is automatically colored to indicate cool (blue) and hot (red) functional units. HotSpot supports running from a power trace file, which means that one does not have to wait for a SimpleScalar simulation to finish. Rather, the power trace file lists the power dissipation in each unit and HotSpot just needs to recompute the thermal interactions.

Once a user has compiled HotSpot, QUILT can quickly run a HotSpot simulation and immediately color the floorplan according to temperature. The user can rearrange functional units according to thermal constraints and re-simulate instantly. This can be very useful for design space exploration.

To use QUILT in this way, one must first produce a power trace file which is a listing of power consumed by the processor units. The simulator in HotSpot called `sim-template` generates this trace file as part of its output. The time it takes for results to be produced is on the order of seconds. An example using the demonstration files included with HotSpot is shown in Figure 7. The cool caches are colored with blue hues and the hot integer units are indicated with red hues.

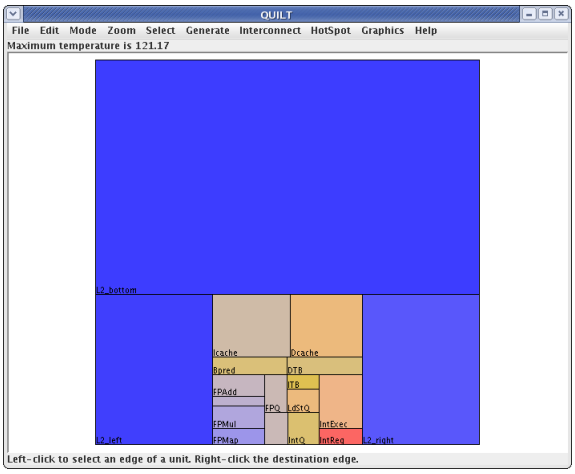

Figure 7. QUILT displaying a temperature-colored floorplan

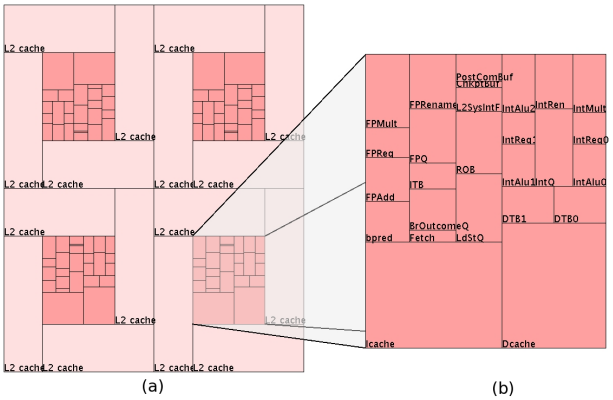

Figure 9. The "zoom effect" for a chip multiprocessor image: (a) floorplan for a proposed 4-core multiprocessing fault-tolerant processor, and (b) closeup of one core [11]

2.6. Graphic Image Generation

Graphics are frequently needed to clarify ideas and depict results for reporting purposes. In the past, producing these graphics involved manually adjusting, resizing and recoloring a large number of functional units. QUILT largely automates these tasks. Single menu commands, shown in Figure 8, permit the floorplan to be recolored, fonts resized, and labels changed.

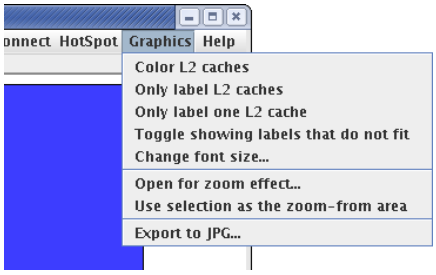

Figure 8. QUILT's Graphics drop-down menu

The coloring may reflect either temperature gradients or IC functionality such as caches. Additionally, for multi-core processors, there is a "zoom effect" generator. This creates a graphic containing a picture of the entire processor with "zoom" lines leading to an image of a single core, as shown in Figure 9.

3. Implementation

QUILT was developed using the Java Virtual Machine environment which makes the software portable across many computing platforms, a common need in academic settings. In its current implementation, the tool comes as a single 70kB file which is easily distributable and does not require tremendous computing resources to run.

This single file is a Java Archive (JAR) file, and in many operating systems, can be executed simply by double clicking on it. Since the JAR file is actually a compressed archive, a user who wishes to modify QUILT can uncompress it to obtain the complete source code.

QUILT takes full advantage of the Java object model. Each functional unit displayed is actually an instance of the `Unit` class, and interconnects are members of the `InterconnectLine` class. The technology node is also encapsulated in a separate class, as are many other components of the software.

Sun's `javax.swing` package was used to render the graphical interface. The actual floorplan editing area was made by extending the `JComponent` class. By selecting Sun's standard graphical interface library, QUILT's source code should be easier to understand and extend.

4. Teaching and Research using QUILT

Simulators are widely used in computer architecture education, as they permit designs to be analyzed relatively quickly and cheaply. However, text-based simulators are not intuitive and they are prone to errors that can be corrected only with careful scrutiny. Although QUILT itself is not a full simulator in the strictest sense (its main task is to provide a front-end for, and a graphical representation of, the data generated for or used by text-based simulators), it enables students to comprehend results more effectively. Studies conducted by Felder and Brent [5] using a questionnaire developed by Felder and Soloman [6] indicated that

82% and 63% of engineering students are visual and sensing learners, respectively. The researchers defined sensing as oriented towards facts and hands-on methods and visual learning pertains to information presented in pictures and diagrams.

A simple exercise using QUILT can be organized in three steps: configuration, simulation, and analysis. The first step requires the student to modify an existing floorplan of a IC chip by using the drawing tools as seen in Figures 2 and 4. An example is shown in Figure 1. If thermal simulation is to be done, the functional unit names should correspond to the power outputs listed by Wattch. The simulation step is actually not performed in QUILT but through HotSpot and/or SimpleScalar. The student can monitor and analyze data visually (Figure 7) while the SimpleScalar simulation is in progress or when it is completed.

There are many other exercises that are possible. For instance, students can compare layouts for temperature versus interconnect delay, or examine the thermal impact of adding newly proposed units, splitting units, etc.

In most computer engineering curricula, computer architecture is taught at two levels: an introductory level course targeted towards undergraduates, and a more advanced course designed for upperclass and graduate students. Due to the complexity of simulators such as SimpleScalar, Wattch and HotSpot, exercises involving their use and modification are usually carried out only in advanced courses, even though they are excellent teaching tools. A graphical based tool such as QUILT permits the instructor to introduce architectural concepts and simulation skills early in a student's education.

Using QUILT as a research tool is not much different from a classroom exercise. A further step would probably involve producing graphics such as the temperature-colored (Figure 7) or multicore floorplan (Figure 9) required as part of the documentation following the research. The authors have used QUILT to generate results for two papers [8, 11].

5. Future Work

Although the authors have used QUILT for their own work, the tool still has room for improvement. For example, when making presentation graphics, XFig and PostScript outputs would be useful. Other areas of improvement are in ease of use, modeling, and new functional unit generation.

From an educational standpoint, ease of use is important. An on-line help system could be added. Additionally, the editor should support "drag-and-drop" of unit placement, similar to other vector graphics editors.

The supported models could also be improved. Interconnect models could be more detailed and more types of interconnects could be added. Delay uncertainty based on temperature could be calculated. Also, more technology node specifications could be added to the system; an easy way to scale a floorplan to a different feature size would also be useful. The transistor count window currently shows three different numbers and a correct interpretation requires the user to decide if the functional unit is of a logic, SRAM or DRAM type. QUILT could be improved to automatically determine the unit's function type and display the appropriate transistor count.

When generating new units, it would be useful to be able to create items like queues and register files based on parameters such as number of ports and byte size. An example would be for QUILT to read SimpleScalar `.cfg` configuration files to automatically generate functional units. ALUs could be pre-defined given an integer width. Such items are impossible to produce exactly, but can be estimated based on current processor designs.

Finally, the program has been designed with modularity and ease of extension in mind. The community is invited to implement any new features they desire and to share them so that all may benefit.

6. Conclusion

In the past, computer architecture simulators tend to be text-based which makes debugging and analysis an inconvenient process. This distracts computer architects from focusing on the main task of designing and verifying new designs. Previous research on human learning and cognition has shown that visual activity enhances the pedagogical experience.

QUILT acts as an interface between raw text data and the user. It can run on a variety of computing platforms which makes it accessible to many users. Using QUILT enables users to make changes to IC layout quickly and to evaluate and analyze the results of their modifications. One of the features not seen in other tools is the ability to generate graphics for hard copies or for use in presentations and documentation.

Finally, QUILT addresses the issues of temperature and interconnect. These are two areas of growing importance for future microprocessors, and need increased emphasis in the classroom. This tool provides interactive visualizations which are effective in helping to meet that need.

7. Acknowledgments

We wish to acknowledge Joseph Toscano for his references to the visual studies performed on human learning.

References

[1] D. Bodemer et al. The active integration of information during learning with dynamic and interactive visualisations. *Learning and Instruction*, 14(3):325–341, 2004.

[2] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A Framework for Architectural-Level Power Analysis and Optimizations. In *Proceedings of the 27th International Symposium on Computer Architecture*, pages 83–94, Vancouver, Canada, Jun 2000.

[3] D. Burger and T. Austin. The SimpleScalar Toolset, Version 2.0. Technical Report TR-97-1342, University of Wisconsin-Madison Computer Sciences Department, Jun 1997.

[4] G. Chen et al. Electrical and optical on-chip interconnects in future microprocessors. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005.

[5] R. M. Felder and R. Brent. Understanding Student Differences. *Journal of Engineering Education*, 94(1):57–72, Jan 2005.

[6] R. M. Felder and B. A. Soloman. Index of Learning Styles. World Wide Web, `http://www.ncsu.edu/felder-public/ILSpage.html`.

[7] W. Huang et al. Compact Thermal Modeling for Temperature-Aware Design. In *Proceedings of the 41st Design Automation Conference*, pages 878–883, San Diego, CA, Jun 2004.

[8] N. A. Nelson et al. Allevating Thermal Constraints while Maintaining Performance via Silicon-Based On-Chip Optical Interconnects. In *Workshop on Unique Chips and Systems*, pages 45–52, Austin, TX, Mar 2005.

[9] O. K. Park and R. Hopkins. Instructional conditions for using dynamic visual displays: A review. *Instructional Science*, 21:427–449, 1993.

[10] Process, Integration, Device and Structures. *The International Technology Roadmap for Semiconductors*. World Wide Web, `http://public.itrs.net/Files/2003PIDS/PIDS2003.pdf`, 2003 edition, 2003.

[11] M. W. Rashid et al. Exploiting Coarse-Grain Verification Parallelism for Power-Efficient Fault Tolerance. In *Proceedings of the 14th International Conference on Parallel Architectures and Compilation Techniques*, Sep 2005.

[12] K. Skadron et al. Temperature-Aware Microarchitecture. In *Proceedings of the 30th International Symposium on Computer Architecture*, pages 2–13, San Diego, CA, Jun 2003.