Master of Engineering Preliminary Thesis Proposal
For 6.191 Prototyping Research Results

December 5, 2002

Cemal Akcaba
Massachusetts Institute of Technology
Cambridge, MA 02139.

Thesis Advisor: Prof. Agarwal

1. Introduction

Since the introduction of the first microprocessor, the computation needs of the world and the resources available to microprocessor designers have changed vastly. In the 1970s, the challenge was to pack as much functioning in to a piece of silicon die as possible, since silicon area was precious.

A decade later silicon area was no longer expensive or precious, and computer designers experimented with pipelining, caching, out of order execution and other techniques they borrowed from super computer architectures. Today, most processors rely on pipelining and superscalar techniques to exploit instruction-level parallelism (ILP). Pipelined processors overlap instructions in time on common execution resources. Superscalar processors overlap execution in space on separate resources. Performance gain that can be obtained from parallelism obtained using pipelines and superscalars is hitting its limits. For example, a four stage pipeline overlaps the execution of four instructions, but it falls short of a 4x performance boost. The problem arises when the pipeline stalls. Stalls are results of data hazards (data dependencies), control hazards (changes in the program flow) and structural hazards (hardware resource conflicts) all of which decrease pipeline efficiency. Hardware scheduled superscalars are already reaching their performance and complexity limits and cannot scale indefinitely.

With the advances in manufacturing technologies it is possible to obtain faster computers using different architectures. The Raw Architecture Workstation (Raw) is a simple, wire-efficient architecture that scales with increasing VLSI gate densities. The project's goal is to achieve performance that at worst equals the performance that can be obtained by scaling an existing architecture, but that can achieve significantly (orders of magnitude) better performance for some applications in which the compiler can discover and statically schedule fine-grain parallelism.

The project aims to achieve its goals by implementing a highly parallel VLSI architecture. This approach aims to fully allocate resources for each application by exposing the low-level

details of the hardware architecture to the compiler so that the compiler can determine the best allocation of resources.

A raw processor is a chip containing a 2-D mesh of identical tiles. The tiles are connected to their nearest neighbors by the dynamic and static networks. To program the Raw processor one programs each of the tiles. Each tile has a processor, a static switch processor and a dynamic router. The tile processor has a 32-bit MIPS instruction set with slight modifications, whereas the switch processor has a MIPS-like instruction set that contains only branches, moves and jumps. Each switch processor instruction also has a route component, which specifies the transfer of values on the static network between that switch and its neighboring switches. The dynamic router runs independently and is under direct control of the user.

The Raw Chip was implemented in collaboration with IBM using IBM's SA-27E process. This process features 6 layers of Cu interconnect, and a 0.12 micron effective channel width.

An evaluation board for the raw chip has been designed, that contains the raw chip, 4 FPGA (Field Programmable Gate Arrays), some external memory and a PCI bus. The evaluation boards are being manufactured and are expected to be completed in early January 2003.

2. Project Description

Using the evaluation board built, I will be designing and implementing a low-level ethernet controller and interface module of a one-gigabit ethernet packet processor (router). The rest of the project, which involves packet forwarding and networking algorithms, will be implemented by Oskar E. Bruening, who will also be working for Prof. Agarwal.

A one-gigabit ethernet packet processor is a very suitable application to test the performance of the raw processor and the evaluation board as the input data stream is high

bandwidth, and will be primarily used as a demo, a proof of concept showing the capabilities of the new evaluation board and the raw processor.

My part is to build an ethernet controller that can send and receive ethernet packets using the raw evaluation board as a basis. The ethernet controller is a piece of hardware that allows the interfacing of some hardware unit to an ethernet network. The ethernet controller will interface the part built by Oskar Bruening to complete the router, therefore it will interface heavily with the raw chip and surrounding FPGAs. The ethernet controller module needs to include a media access control (MAC) module, a centralized controller, a 10 Base T transreceiver and Manchester encoder/decoder module, and a PCI interface module to provide basic functionality of sending, receiving IP (Internet protocol) or UDP packets. The ethernet controller unit could be controlled through its interface to the PCI bus. The detailed functionality of the modules of the ethernet controller are as follows.

The MAC module handles all aspects of ethernet frame transmission and reception, including collision generation, preamble generation and detection, CRC (cyclic redundancy check) generation and test.

The centralized controller module controls all of the other modules of the ethernet controller. It accepts commands through the PCI bus and acts accordingly. The controller can send, receive or discard the packages it receives. The centralized controller unit also interfaces with the main memory of the module storing necessary state information.

The transceiver and Manchester encoder/decoder module will provide filtering, preprocessing of the input bit stream (from ethernet jack RJ45) before passing it to the centralized controller unit. The preprocessing involves Manchester decoding of the input data stream if the controller is receiving a packet and encoding if the controller is sending packet over the ethernet.

During the design and implementation of these modules described above, the FPGAs on the Raw evaluation board and one or more of the tiles on the raw processor will be used in order to exploit the implicit parallelism in the task of processing packets. During the design of the module,

especially the 10BASET transceiver, some external off the shelf hardware might be incorporated. These separate parts might be gathered on a printed-circuited board that mounts on the PCI bus of the evaluation.

With the use of the Raw processor and evaluation board, the ethernet controller to be designed will hopefully show an improved performance.

3. Project Milestones

It is the wisest strategy to design hardware systems for iteration. Therefore I will tend to use this strategy, and design a preliminary system before the end of Spring 2003. This preliminary design will probably not implemented in full scale but will be implemented partially and/or simulated in its entirety. This is the period in which the actual thesis proposal will be produced.

Then during the summer of 2003, I will perform tests on my initial design and iterate on it in order to finalize my design. In this period, it will also be wise to order any piece of hardware I will need for the completion of the project. This might involve sending a PCB layout to a production plant in addition to ordering regular parts from hardware vendors.

In Fall 2003, I will implement the integration of hardware and software, and perform basic testing of the system. In IAP of 2003, I will fix any errors that were brought to daylight through testing (hopefully, there will not be too many errors to fix.)

In Spring 2004, the system integration will be carried out in order to produce a fully functional end product, a one-gigabit ethernet router. I will also complete my thesis during Spring term 2004.

4. Resources Required

The most important resource that I will need is the Raw processor evaluation board. As these boards are still manufactured, I might need to use a simulator at early stages until the board is available. Also, since the evaluation boards will be shared with other people who are working on other projects, the time on evaluation boards is another crucial resource that I will need.

Apart from that, I might need to order some hardware which might not be readily available. The other resources I will need, are knowledge about the evaluation board, the raw processor instruction set and any other skills that I will need to learn in order to complete the project.

5. Technical Risks

There are not many technical risks in this project. There is an Error #42 though, the completion of the project depends on successful implementation of the evaluation board. Luckily, the evaluation boards are scheduled to be completed very early in the plan (in January 2003). This will probably give me enough time to do changes should the evaluation board have some problems.

6. What's next

I will start getting more acquainted with the research group and the research area. I will start working for the project as a UROP student in IAP, and I will continue the UROP work throughout Spring 2003. During IAP and Spring 2003, I will experiment with the evaluation board or its simulator and try to come up with a preliminary design or at least a good idea of how things should work before I write my thesis proposal at the end of the Spring 2003. During the summer

of 2003, I plan to keep working as a UROP student and formalize the design of the hardware and order any parts, PCBs that will be needed.