

CSE 291E and ECE 290C
Spring 2002
Course Project
4/11/02

1 Overview of Project

The goal of this project is to create a custom accelerator for an embedded application. The project consists of picking an important application that you find interesting. The next step is to determine where most of the time is spent during execution for that program, since this is where you need to concentrate your optimizations. You will then design a custom accelerator to reduce the time it takes to perform an important part of the program's execution. Before implementing the design you will need to first estimate its performance and the amount of potential improvement possible from using the accelerator. The next part of the project is to build the accelerator in Verilog to obtain design timing, area and power estimates. Finally, the timing analysis will be used in conjunction with a performance simulator to compare the performance of a processor with the custom accelerator to one without. The Table below summarizes the milestones of the project and the due dates.

Date	Description	What to turn in
April 18th	Intro to Verilog	Verilog Fibonacci Implementation
April 25th	Groups and Application Area	Group members and application area
April 30th	Intro to SimpleScalar	TBD
May 2nd	Select Application and Project Plan	Source code for application and project plan
May 9th	Application Performance Analysis	Performance numbers, and area of application to optimize
May 21st	High Level Accelerator Design and Performance Estimation	Accelerator design and estimated performance
June 4th, 6th	Final Project Presentation	PowerPoint shows
June 7th	Final project Due	Verilog implementation, and performance with and without accelerator

The following sections describe in more detail what is expected for each part of the project.

2 Introduction to Verilog and SimpleScalar

The Verilog and SimpleScalar assignments are to bring you up to speed with the tools you will be using in this project. Verilog will be used to implement your custom accelerator. Synopsys will then be used to evaluate the timing, area and power for your design.

SimpleScalar is a cycle accurate simulator used to examine the performance of your application, before and after using the accelerator. It can also be used to help guide the design of the accelerator, by examining

the detailed pipeline behavior of the program. The SimpleScalar assignment will make you familiar with using SimpleScalar and modifying it to add in an accelerator.

The write ups for these two assignments should be no more than 1-2 pages.

3 April 25th - Form Groups and Pick Application Area

Groups need to be finalized and an application area needs to be decided upon.

4 May 2nd - Select Application and Project Plan

Pick an interesting application and find or create code to implement the application or its important kernels. You need to turn in the source code. This code will be used to guide the design of the accelerator and to examine its performance.

At this stage we also require you to turn in a plan for your project (no more than 2 pages). The plan describes the actions you'll take to complete the rest of the milestones. This includes how you are going to evaluate the performance of the application. This write up includes the list of tools you will be using to perform each step, and the inputs used to run the application.

Creative ideas for application areas and designs are encouraged, and the more time spent on finding interesting applications and related source code, the easier we will grade your implementation. If you take a simple, well documented, already in production application accelerator you will be expected to have a better design than if you pick something that has not been done before. Please discuss your ideas with us as soon as possible so that we can carve out an interesting problem together.

Some of the applications may be too complicated to model their performance using SimpleScalar. If this is the case, you'll need to get approval from Brad Calder for how you plan to evaluate and model the performance of your application.

5 May 9th - Application Performance Analysis

For this milestone you need to have examined in detail the execution performance of your application. This entails finding the procedures, loops, and instructions where a significant amount of execution time is spent. This is one of the most important parts of the project, since it will be used to help guide the design of the custom accelerator. You will need to turn in a detailed analysis of the application and its source code. Details for what to turn in will be specified later.

Many different tools may be used to perform this analysis. These range from SimpleScalar (a detailed cycle-level simulator) to Gprof (a high level profiler provided on most unix systems). Gprof is a profiler that identifies the procedures where most of the time is spent in executing the application.

6 May 21st - High Level Accelerator Design and Performance Estimation

The application performance analysis will have pinpointed the important parts of the application. These are the parts to optimize with a custom accelerator. As part of this milestone, you will turn in the high level design of your accelerator.

Before building the accelerator in hardware (Verilog), it is important to first provide an analysis of the potential benefits of your optimization. For example, this performance estimation could be done in SimpleScalar assuming you can perform your accelerated hardware in N cycles. This provides a rough

estimation as to the benefits of your accelerator and is used to help tune and guide the development of your accelerator.

In order to perform this analysis in SimpleScalar, you would need to modify your program or simulator to replace the instructions that would be executed in the accelerator to take only Y cycles of latency, instead of N in the baseline architecture, where Y should be less than N . Comparing the execution time of these two runs of the program provides the estimated performance improvement.

At this stage, you'll turn in the high level accelerator design, what lines of code the design will be used to optimize for the application, and the estimated performance improvement from it.

7 June 7th - Final Project Due

From the high level design you will implement the accelerator in Verilog and use Synopsys to evaluate the design timing, area and power. The accelerator will be tuned and its design will be optimized to achieve reasonable timing, area and power results. These metrics will then be combined with SimpleScalar simulation results to compare a processor with your custom accelerator to one without.

A final write up of the architecture, accelerator and performance results will be turned in, along with the Verilog implementation of the accelerator. The write up should be around 6 pages. This needs to include a minimum of 3 pages of single spaced "text" describing the performance of the application, the application's important places to optimize, the architecture of the accelerator, the methodology used, and performance results. The final write up should also include figures and graphs to help describe and understand the accelerator and its performance.