

PROJECT DESCRIPTION: In this project, you will use a trace driven execution simulator for a superscalar RISC architecture to explore the effect of *superscalar factor*, *reservation stations per function unit*, and the *functional unit duplication factor* on performance. The principle performance metric is instructions/cycle throughput (IPC). However, assessments of performance should consider the implementation parameter's affect on cycle time and system cost. Measuring resource utilization (reservation stations, functional units, buffer utilization) provides important insight into trace execution.

The simulator and associated documentation can be found at <http://www.ece.gatech.edu/~linda/6100/Projects/>. In your analysis, use the **compress** trace file which gives a trace of the instructions completed during the execution of the Spec95 “compress” benchmark program. The files are in ASCII format with each entry containing the instruction address, instruction word (broken into two parts), and assembly language source. The instructions are in a hybrid MIPS/DLX format, which is described in the Trace Format Guide and “Additional instruction descriptions” on the 6100 Projects web page.

SYSTEM DEFINITIONS: The following characteristics should be set upon starting the simulation:

- ? Instruction fetches hit in the cache 96% of the time.
- ? Data fetches hit in the cache 94% of the time.
- ? All cache misses incur a 3 cycle delay.
- ? Branch prediction accuracy is 91%.

The *functional unit duplication factor*,  $N$ , is defined as the  $N$  copies of the four units described below. It is the degree to which they are duplicated. For example, a FU duplication factor of 3 means there are 3 of each type of unit. The latency of each type of unit is shown in the second row of the table.

integer arithmetic	floating point arithmetic	memory accesses	branch computation
1 cycle	3 cycles	2 cycles	1 cycle

PART A: GAINING FAMILIARITY WITH THE SIMULATOR. Experiment with the simulator by single stepping through portions of the trace. Answer the following questions:

1. When are reservation stations freed up? That is, when does an instruction leave a reservation station?
2. Can reservations for non-memory functional units be serviced out of order? Explain how you determined this.

PART B: SIMULATION. For this part, run the simulator on the compress trace using lane sizes of 1, 2, 4, and 6. Simulate each of the four lane sizes with the following FU duplication factors: 1, 2, 3, 4. For each of the 16 pairings of lane size and FU duplication, vary the number of reservation stations/execution unit from some small, initial value until you see less than a 5% reduction in overall execution time (total cycles executed). Fix the size of the renaming buffer and reorder buffer at sizes that are large enough so that they do not cause any stalls by filling up and preventing instructions from dispatching or completing.

1. Summarize your simulation results in a 4x4 table (one entry for each of the 16 pairings of lane size and FU duplication factor). Each entry should contain 2 pieces of data: the execution time (measured in cycles) and the instruction per cycle throughput rate (IPC).
2. Report the renaming buffer and reorder buffer sizes you used and explain how you chose them.
3. Discussion: Explain how the tabulated performance statistics are affected by the number of lanes, the FU duplication factor, and the number of reservation stations.

PART C: OPTIMIZING THE CPU ARCHITECTURE UNDER A CONSTANT CHIP-AREA CONSTRAINT.

In this part of the project, we will assign a chip-area cost for implementing each of the hardware components described in the machine configuration. Parameterized area functions for each hardware

component are given in the cost model in Attachment A. Using these area functions, *your objective is to design a superscalar CPU that achieves the highest performance on the compress program while occupying a chip area of 100 mm<sup>2</sup> or less.*

For your design, you are free to vary the machine parameters in any way you desire with the following restrictions: 1) you must not change the latencies of any of the functional units, 2) you must not vary the cache hit rates, miss penalties, or the branch prediction accuracy, 3) your CPU must include at least one of each type of functional unit. The renaming and reorder buffers may be sized differently.

While the results obtained in Parts A and B will be helpful in guiding your design process, you will need to perform additional simulations if you investigate other design points. You may find it helpful to create a spreadsheet for calculating and comparing chip-area costs for alternative CPU designs.

For this part of the project, you do not have to provide data from every simulation you performed to obtain an optimized design. However, you should outline your strategy for exploring the design space, describe the different simulations you performed, discuss any interesting design trade-offs you encountered, and justify your final choice of machine parameters. You should also provide performance statistics (execution time in cycles and IPC) for your final design, a description of its machine configuration, and a chip-area breakdown for the different hardware components.

**PROJECT REPORT:** The project report is limited to fifteen pages, using a 12 point times roman font, 1.5 line spacing, and one inch margins on all sides. The format of the report is as follows:

1. Title and Author
2. Acknowledgments of others with whom you discussed the project.
3. Abstract (summarize the report's conclusions in 100 words or less)
4. Introduction (state what is being studied; briefly summarize results)
5. Answers to Part A questions.
6. Results and analysis from Part B.
7. Discussion of design exploration and results obtained from final design (parameters chosen, statistics, chip-area cost, etc.).
8. Conclusions (summarize critical results and key insights learned about the effect of the various architectural parameters examined on performance).

Be sure to allow adequate time to run the simulations, analyze the results, and write a thorough, readable report. Your grade on this project will be based primarily on your presentation and analysis of the data. Try to summarize data in a clear manner using tables and/or graphs. Every table or graph you include must be referenced somewhere in your discussion.

**PROJECT DUE DATE: 25 October 2001, in class.** Bring a hardcopy of your report and a diskette storing your results (e.g. in log files or a spreadsheet) and a softcopy of your report. Put the report and your diskette in a manila envelope, clearly labeled with your name and course number. Be sure to keep a backup of all materials you hand in.

*Any project reports slipped under Professor Wills' office/lab door will not be graded!* Arrangements for handing in the project report after the deadline must be made with **Cory Hawkins** ([cory@ece.gatech.edu](mailto:cory@ece.gatech.edu)) before the due date.

**Attachment A: Chip-Area Models for CPU Hardware Structures** (for use in Part C)

Table 1: Definition of Chip-Area Model Parameters

Parameter Name	Corresponding Machine Configuration File Parameter(s)
$N_{RS}$	Number of reservation stations/execution unit
$N_{RN}$	Number of renaming buffer entries
$N_{RB}$	Number of reorder buffer entries
$L$ (# lanes)	Superscalar factor
$U_I$	Number of integer units
$U_{FP}$	Number of floating point units
$U_B$	Number of branch computation units
$U_M$	Number of memory units

Table 2: Chip-Area Cost Functions for CPU Components

CPU Component	Chip-Area Cost Function ( $\text{mm}^2$ )
Reservation Stations	$0.065 * (U_I + U_{FP} + U_B + U_M)N_{RS}$
Renaming Buffer	$0.04 * N_{RN}$
Integer Units	$2.0 * U_I$
FP Arithmetic Units	$2.5 * U_{FP}$
Branch Units	$2.0 * U_B$
Memory Units	$3.0 * U_M$
Reorder Buffer	$0.04 * N_{RB} * L + 0.035 * N_{RB}$