**NC State University**

**Department of Electrical and Computer Engineering**

**ECE 521: Fall 2015 (Rotenberg)**

# Project #3: Dynamic Instruction Scheduling

**by**

**ARJUN AUGUSTINE**

NCSU Honor Pledge: "I have neither given nor received unauthorized aid on this test or assignment."

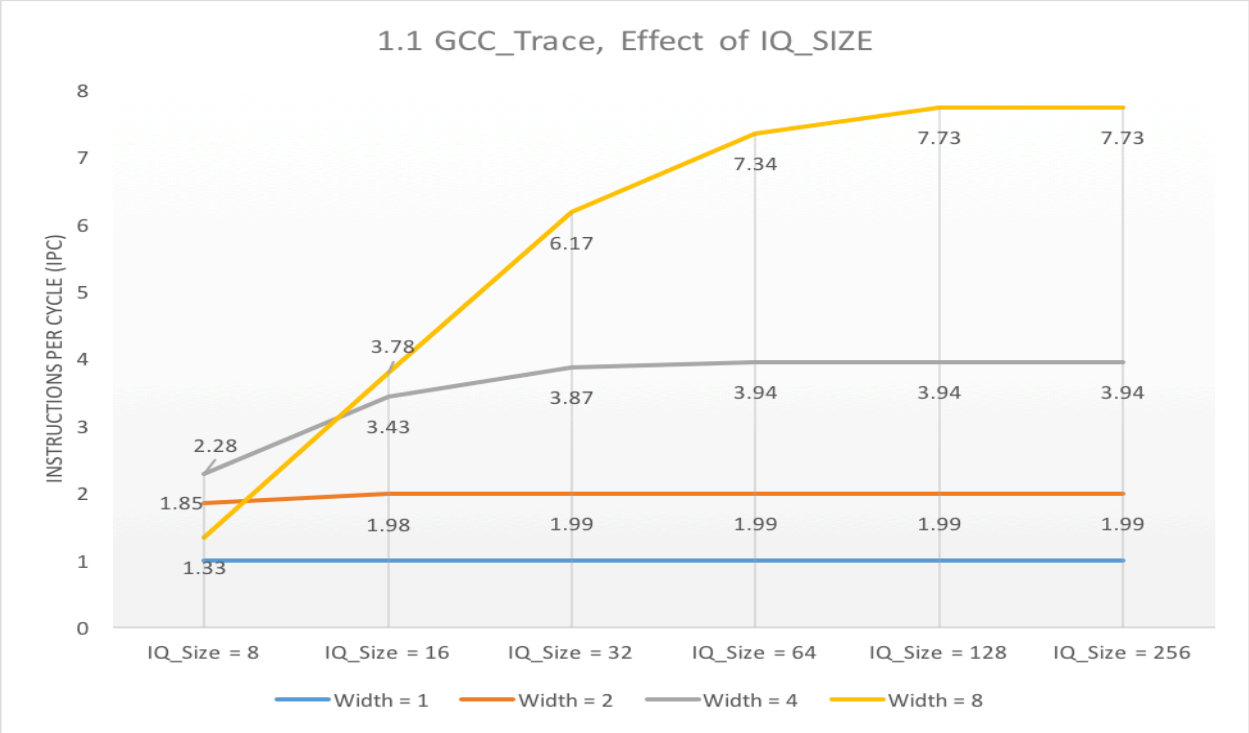
Student's electronic signature: _____Arjun Augustine_____
(sign by typing your name)

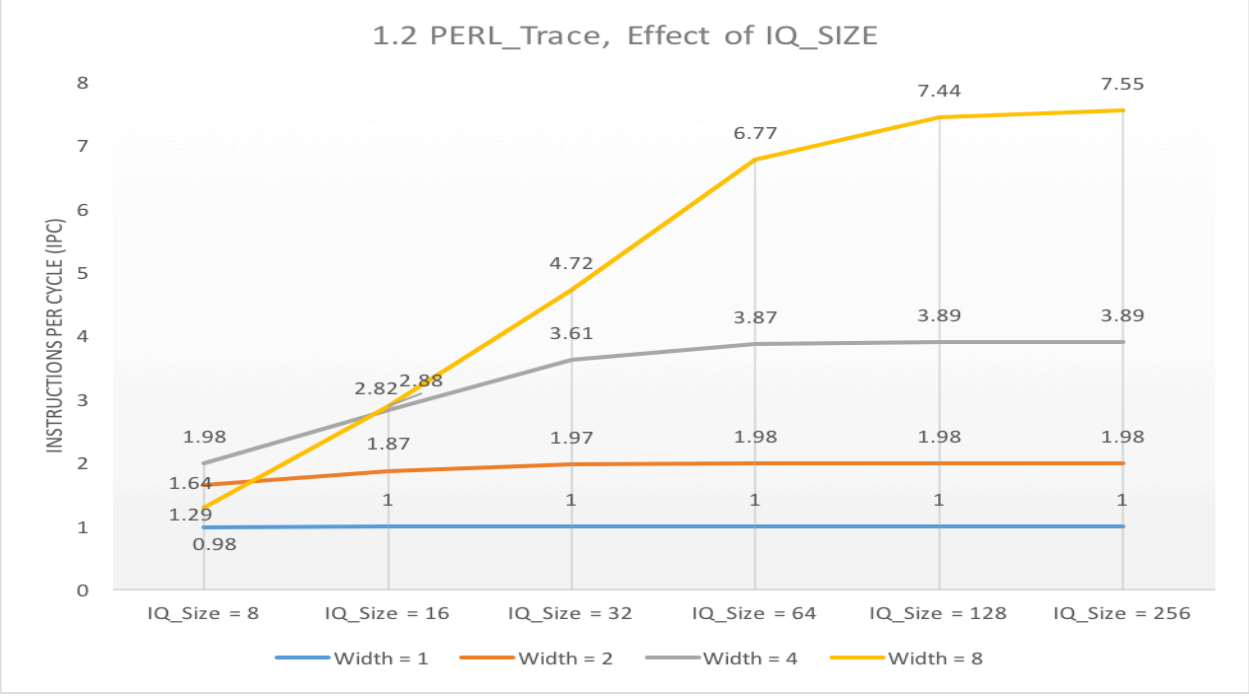
Course number: _____521_____
(463 or 521?)

## 1. Large ROB, Effect of IQ_SIZE

***For each trace*** on the website, use your simulator as follows:

***1.1. [5 points] Graphs:*** *Keep ROB_SIZE fixed at 512 entries so that it is not a resource bottleneck. For each benchmark, make a graph with IPC on the y-axis and IQ_SIZE on the x-axis. Use IQ_SIZE = 8, 16, 32, 64, 128, and 256. Plot 4 different curves (lines) on the graph: one curve for each of WIDTH = 1, 2, 4, and 8.*



***1.2. [5 points] Graphs Analysis:*** *Using the data in the graphs, for each WIDTH (1,2,4 and 8), find the minimum IQ_SIZE that still achieves within 5% of the IPC of the largest IQ_SIZE (256). This exercise should give four optimized IQ_SIZEs per benchmark, one optimized for each of WIDTH = 1,2,4 and 8. Tabulate the results of this exercise.*

| | Minimum IQ_SIZE that lies within 5% of the IPC of the largest IQ_SIZE | |
|---|---|---|
| **Width of Pipeline** | **Gcc trace benchmark** | **Perl trace benchmark** |
| 1 | 8 | 8 |
| 2 | 16 | 32 |
| 4 | 32 | 64 |
| 8 | 128 | 128 |

**1.3. [5 points] Discussion**:

*1.3.1 The goal of a superscalar processor is to achieve an IPC that is close to WIDTH (which is the peak theoretical IPC of the processor). Given this goal, what is the relationship between WIDTH and IQ_SIZE? Explain.*

**ANSWER:** As width increases, the processor needs a larger issue queue size to continue giving proportional performance. This is not surprising since a larger width implies that a lot more instructions get dumped to the issue queue per cycle and a proportional increase in queue size is required to handle this. From the table, we can see that though IQ_SIZE required seems to be roughly 8 times the Width of the pipeline, as we increase Width beyond a certain level, we need more than 8 times Width as our Issue Queue size to handle the traffic.

From the graphs, we can conclude that at lower levels of Issue Queue size, increase in Width gives marginal improvement in IPC up to a certain level, after which IPC goes down drastically. This may be due to the fact that instructions dependent on a source came closer on the heels of the producer due to larger Width and the Issue Queue got clogged up. Also, it can be noted that for a particular pipeline width, increasing Issue Queue size provided good returns initially before the returns started peaking out.

Thus, we need an Issue Queue size proportional to the Width of the pipeline, but increasing this drastically doesn't provide returns as expected.

*1.3.2 Do some benchmarks show higher or lower IPC than other benchmarks, for the same microarchitecture configuration? Why might this be the case?*

**ANSWER:** The perl trace seems to generate IPC levels notably lower than the IPC levels generated by the gcc trace. This may be because there were more dependent consumer instructions or because the consumer instructions that came right behind the producer instructions stalling the pipeline at the issue, and subsequently the higher stages, many times more than the gcc trace, leading to a lower IPC.

## 2. Effect of ROB_SIZE

**2.1** ***[5 points] Graphs:*** *for each benchmark, make a graph with IPC on the y-axis and ROB_SIZE on the x-axis. Use ROB_SIZE = 32, 64, 128, 256, and 512. Plot 4 different curves (lines) on the graph: one curve for each of WIDTH = 1, 2, 4, and 8. For a given WIDTH, use the optimized IQ_SIZE for that WIDTH.*