NC State University

Department of Electrical and Computer Engineering

ECE 463/563: Fall 2021 (Rotenberg)

Project #3: Dynamic Instruction Scheduling

by

Prajakta Keshavrao Jadhav

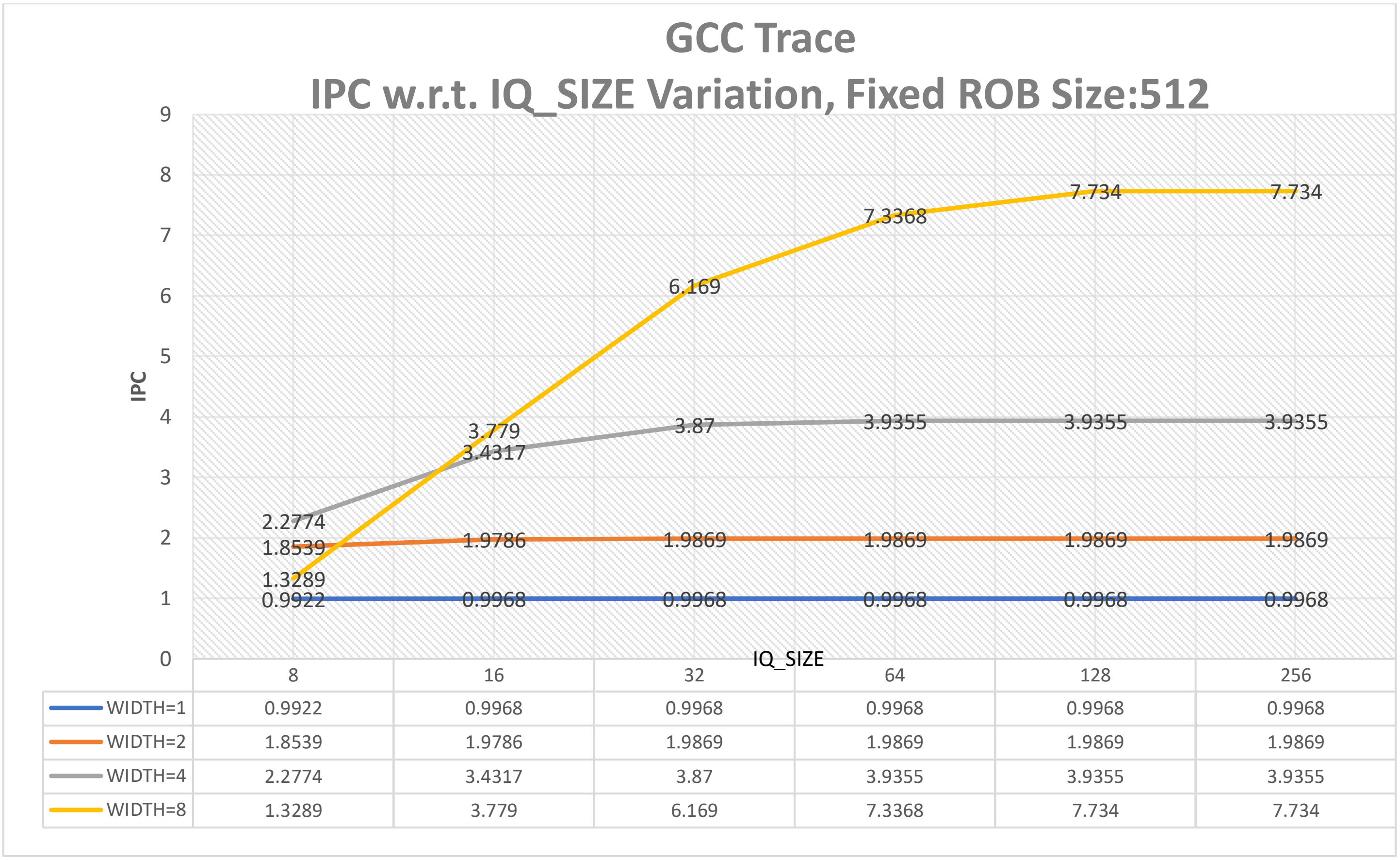
pkjadhav, 200375352

Analysis I:

Effect of IQ_SIZE Variation on IPC with Fixed Large ROB Size

Q1: 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.

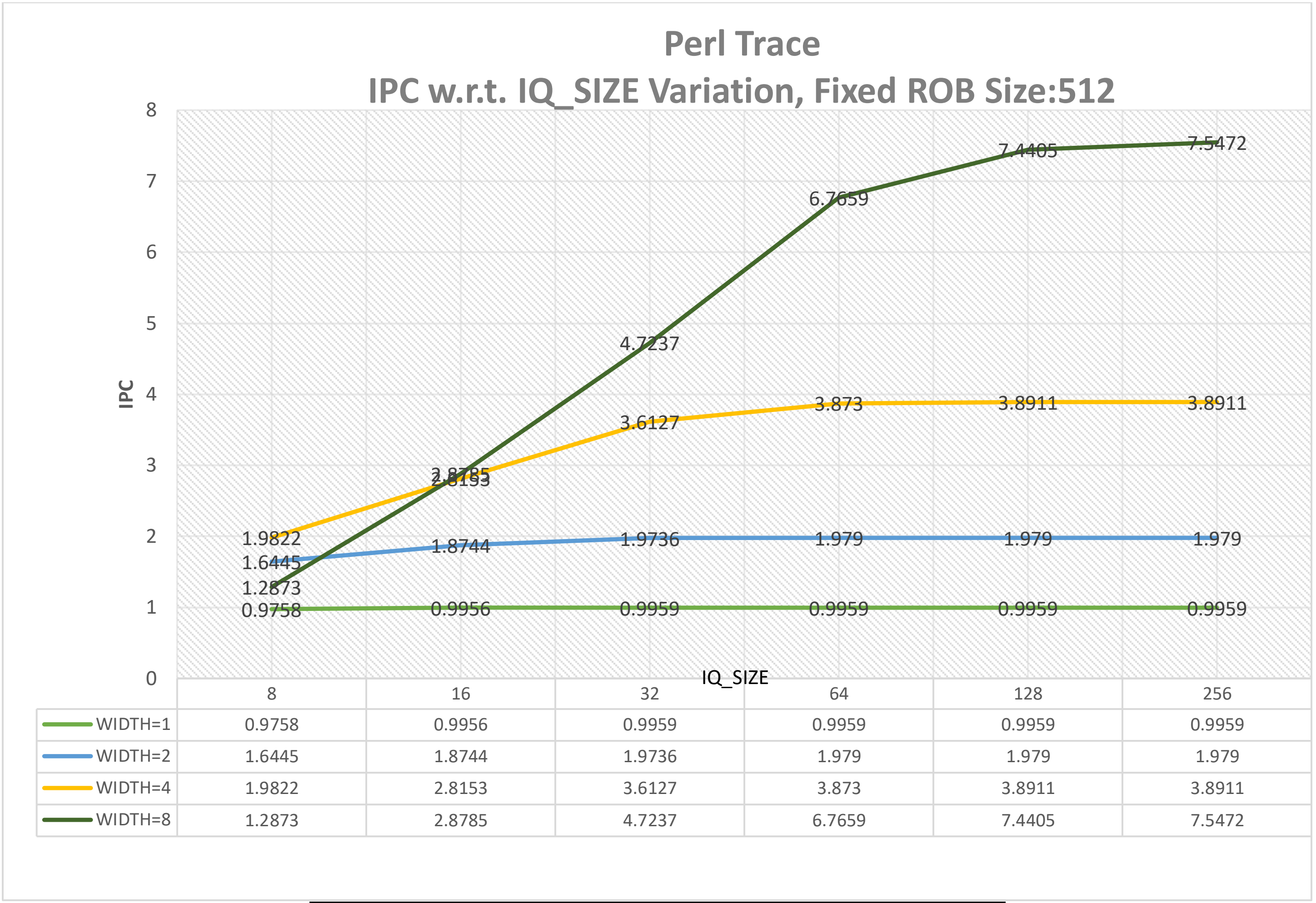
1. **With GCC Trace**



	8	16	32	64	128	256
WIDTH=1	0.9922	0.9968	0.9968	0.9968	0.9968	0.9968
WIDTH=2	1.8539	1.9786	1.9869	1.9869	1.9869	1.9869
WIDTH=4	2.2774	3.4317	3.87	3.9355	3.9355	3.9355
WIDTH=8	1.3289	3.779	6.169	7.3368	7.734	7.734

GCC_Trace				
IQ_SIZE	IPC			
	WIDTH=1	WIDTH=2	WIDTH=4	WIDTH=8
8	0.9922	1.8539	2.2774	1.3289
16	0.9968	1.9786	3.4317	3.779
32	0.9968	1.9869	3.87	6.169
64	0.9968	1.9869	3.9355	7.3368
128	0.9968	1.9869	3.9355	7.734
256	0.9968	1.9869	3.9355	7.734
95% IPC of IQ_Size:256	0.94696	1.88756	3.738725	7.3473

2. With PERL Trace



	8	16	32	64	128	256
WIDTH=1	0.9758	0.9956	0.9959	0.9959	0.9959	0.9959
WIDTH=2	1.6445	1.8744	1.9736	1.979	1.979	1.979
WIDTH=4	1.9822	2.8153	3.6127	3.873	3.8911	3.8911
WIDTH=8	1.2873	2.8785	4.7237	6.7659	7.4405	7.5472

Perl_Trace				
IQ_SIZE	IPC			
	WIDTH=1	WIDTH=2	WIDTH=4	WIDTH=8
8	0.9758	1.6445	1.9822	1.2873
16	0.9956	1.8744	2.8153	2.8785
32	0.9959	1.9736	3.6127	4.7237
64	0.9959	1.979	3.873	6.7659
128	0.9959	1.979	3.8911	7.4405
256	0.9959	1.979	3.8911	7.5472
95% IPC of IQ_Size:256	0.946105	1.88005	3.696545	7.16984

Q2. Using the data in the graph, 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_SIZE's per benchmark, one optimized for each of WIDTH = 1, 2, 4, and 8. Tabulate the results of this exercise as follows:

Fixed ROB Size: 512		
Pipeline WIDTH	"Optimized IQ_SIZE per WIDTH" Minimum IQ_SIZE that still achieves within 5% of the IPC of the largest IQ_SIZE:256	
	GCC TRACE	PERL TRACE
1	8	8
2	16	32
4	32	64
8	64	128

Optimum IQ Size Table

Q3.

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).

3.1) As we increase WIDTH, what trend do you notice regarding the IQ_SIZE needed to achieve this goal (refer to the "Optimized IQ_SIZE per WIDTH" table that you created)? Why is there this trend?

From above table, it is observed that optimum IQ_SIZE increases as the WIDTH of pipeline registers in the superscalar processor is increased.

With increased WIDTH pipeline size, more number of instructions are moved from one pipeline register to the next and then in the IQ as well. When this is done, IQ will get larger instruction bundle which might include both- the truly dependent and independent instructions. If IQ is made bigger enough to accommodate both dependent and independent instructions at the same time, younger independent instructions will not be stalled in dispatch stage i.e. increasing the size of the issue queue will allow the younger independent instructions to be executed without stalling. As result, we get better IPC(i.e. lower CPI) value compare to smaller IQ size processor.

Also note that, instead of increasing IQ size up to 256(maximum), here, similar IPC is achieved using optimum IQ size. This indicates for given particular WIDTH pipeline, further increase in the IQ size yields diminishing returns and only adds to the Hardware's expense. This trend shows, there is a trade-off between the cost per increase in size for a minor performance boost vs. the cost per increase in area.

So, it can be concluded that an issue queue of the largest feasible size gives an idea of how much IPC a system with a specific superscalar width for a fixed reorder buffer size can achieve. Choosing a minimum issue queue size that produces an IPC value that is within 5% of the IPC of the highest Issue queue size is an optimum way to get closer to a processor's theoretical peak IPC while keeping area limits in mind.

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

Yes, given the identical microarchitecture configuration for two traces, PERL trace shows lower IPC values than GCC trace.

1. It's possible that the cause is due to a **reliance between the instructions** in the different trace files. Perl may contain more number of instructions that are truly dependent on each other, such that an instruction's one of the operands (or both) may be dependent on an instruction that is ahead of its time in the issue queue or is in execution waiting for its wake-up call, resulting in a wait period. If this might be the case, in comparison to GCC trace, Perl trace will be causing stalling of more younger independent instructions, hence gives the a lower IPC value for PERL than GCC.
2. Another reason could be **total number of complex instructions** present in the benchmark might differ. If PERL includes most of instructions of 5 cycles execution, it will make independent instructions in IQ to wait for few more cycles before issuing to execution unit, hence IPC value be lower compared to the GCC trace which might be including most of instructions with lower execution cycles(1/2) requirement.

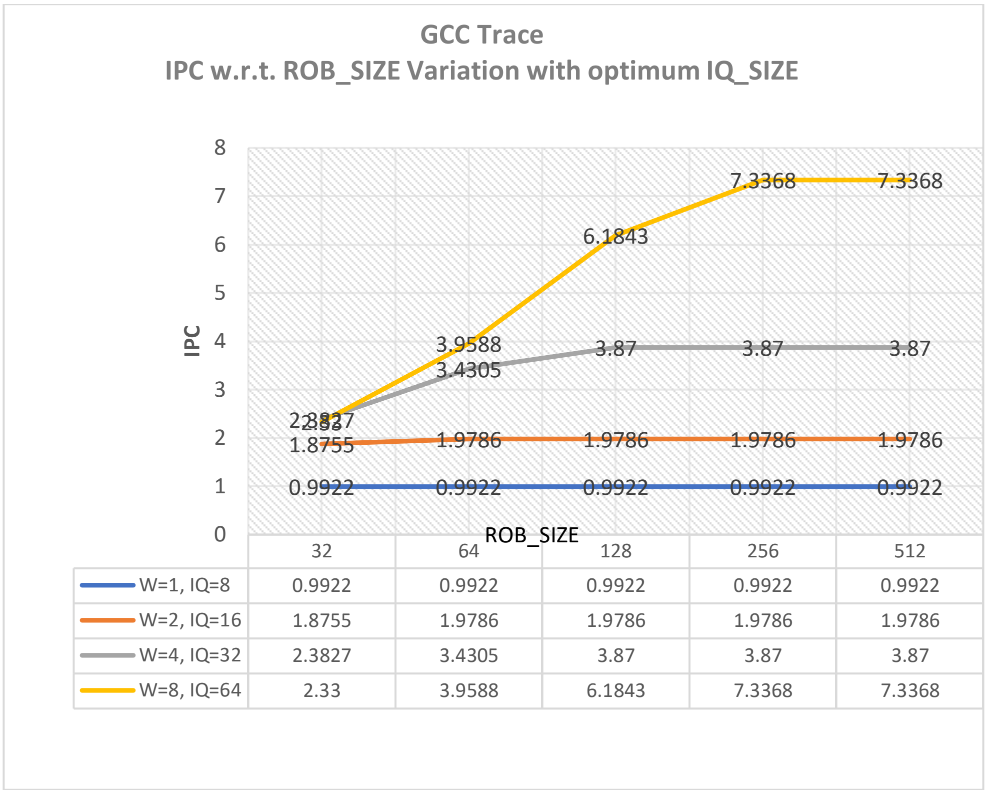
Analysis II:

Effect <u>of ROB_SIZE Variation</u> on IPC with Fixed optimum IQ Size per WIDTH

Q. 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, as obtained from the table in Section 1.

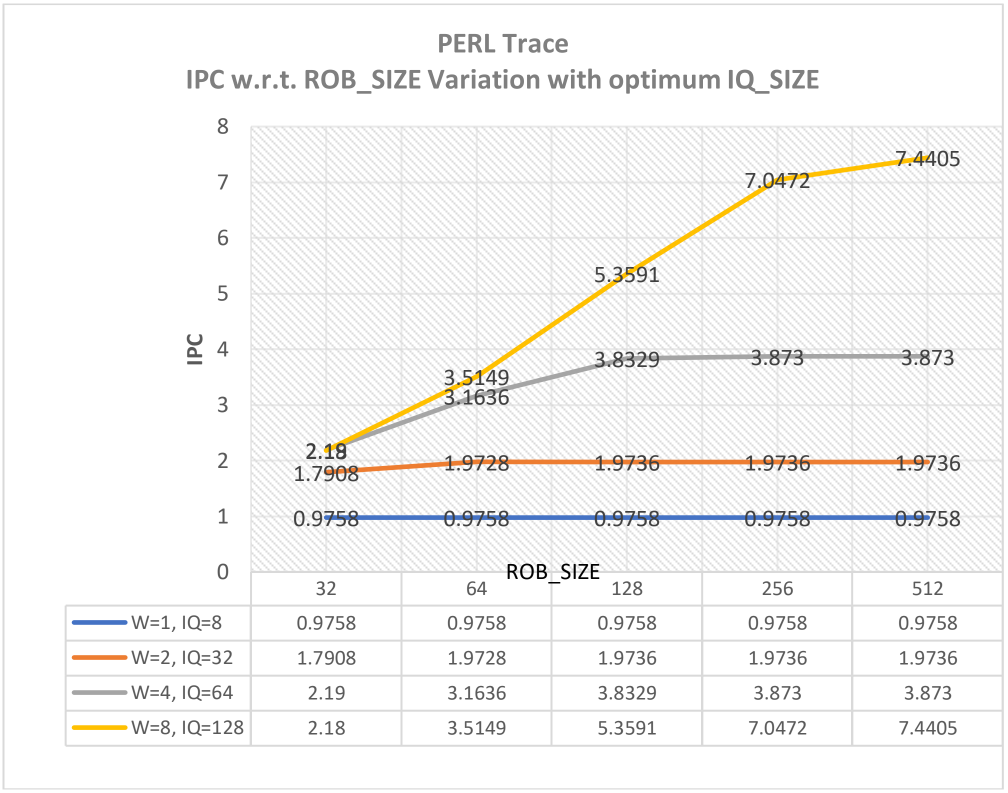
Used respective IQ size per WIDTH from optimum IQ table and varied ROB size:

1. With GCC Trace:



	32	64	128	256	512
▬ W=1, IQ=8	0.9922	0.9922	0.9922	0.9922	0.9922
▬ W=2, IQ=16	1.8755	1.9786	1.9786	1.9786	1.9786
▬ W=4, IQ=32	2.3827	3.4305	3.87	3.87	3.87
▬ W=8, IQ=64	2.33	3.9588	6.1843	7.3368	7.3368

2. With PERL Trace:



	32	64	128	256	512
▬ W=1, IQ=8	0.9758	0.9758	0.9758	0.9758	0.9758
▬ W=2, IQ=32	1.7908	1.9728	1.9736	1.9736	1.9736
▬ W=4, IQ=64	2.19	3.1636	3.8329	3.873	3.873
▬ W=8, IQ=128	2.18	3.5149	5.3591	7.0472	7.4405