NC State University

Department of Electrical and Computer Engineering

ECE 463/563: Fall 2021 (Rotenberg)

Project #3: Dynamic Instruction Scheduling

by

RAMACHANDRAN SEKANIPURAM SRIKANTHAN

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

Student's electronic signature: Ramachandran Sekanipuram Srikanthan

Course number: 563

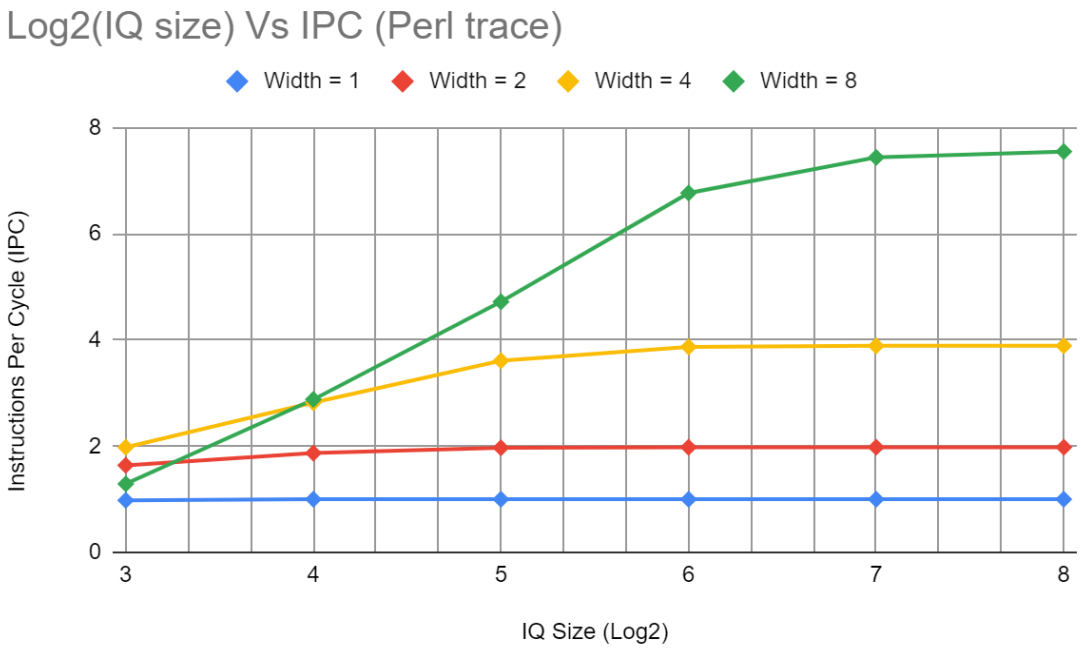
Analysis 1: Large ROB Size, Effect of IQ size

IPC_GCC					
IQ Size	IQ Size log2	Width = 1	Width = 2	Width = 4	Width = 8
8	3	0.99	1.85	2.28	1.33
16	4	1	1.98	3.43	3.78
32	5	1	1.99	3.87	6.17
64	6	1	1.99	3.94	7.34
128	7	1	1.99	3.94	7.73
256	8	1	1.99	3.94	7.73

The graph is plotted for Issue Queue Size in x axis and Instructions Per Cycle in y axis for fixed Reorder buffer size of 512 entries. The IQ size is varied from 8, 16, 32, 64, 128 to 256 for each of the widths of pipeline from 1,2, 4 to 8. From the graph, we can see there is an increase in the number of instructions per cycle as we increase the size of the Issue queue for superscalar Out of Order pipeline of higher width. The reason being that for pipeline width of 1, even though we increase the size of issue queue, a single instruction is being executed, although multiple instructions are ready in the issue queue. As we increase the width of the superscalar pipe line, we can have more ready instructions in issue queue which can be simultaneously executed, depending on width of the pipeline, each cycle which further increases the performance as in number of instructions per cycle while we increase the issue queue size. We can also see that the IPC gets saturated after a threshold point for a particular width. The threshold point is the

point at which the width of pipeline is same as issue queue size. After that we can see that, there isn't significant increase in IPC.

IPC_Perl					
IQ Size	IQ Size log2	Width = 1	Width = 2	Width = 4	Width = 8
8	3	0.98	1.64	1.98	1.29
16	4	1	1.87	2.82	2.88
32	5	1	1.97	3.61	4.72
64	6	1	1.98	3.87	6.77
128	7	1	1.98	3.89	7.44
256	8	1	1.98	3.89	7.55



The graph is plotted with IQ size vs IPC for Perl Trace dataset. From the graph, we can see that, the IPC count increases as we increase in IQ size for larger pipeline width. For smaller width of the pipe line, we can see that increase in Issue queue size doesn't have any significant impact in IPC count. Also, we can see that as we increase the size of issue queue, after a threshold value the IPC gets saturated. The reason being that even though having higher size in issue queue with many ready instructions the width of the pipeline restricts executing all the ready instructions.

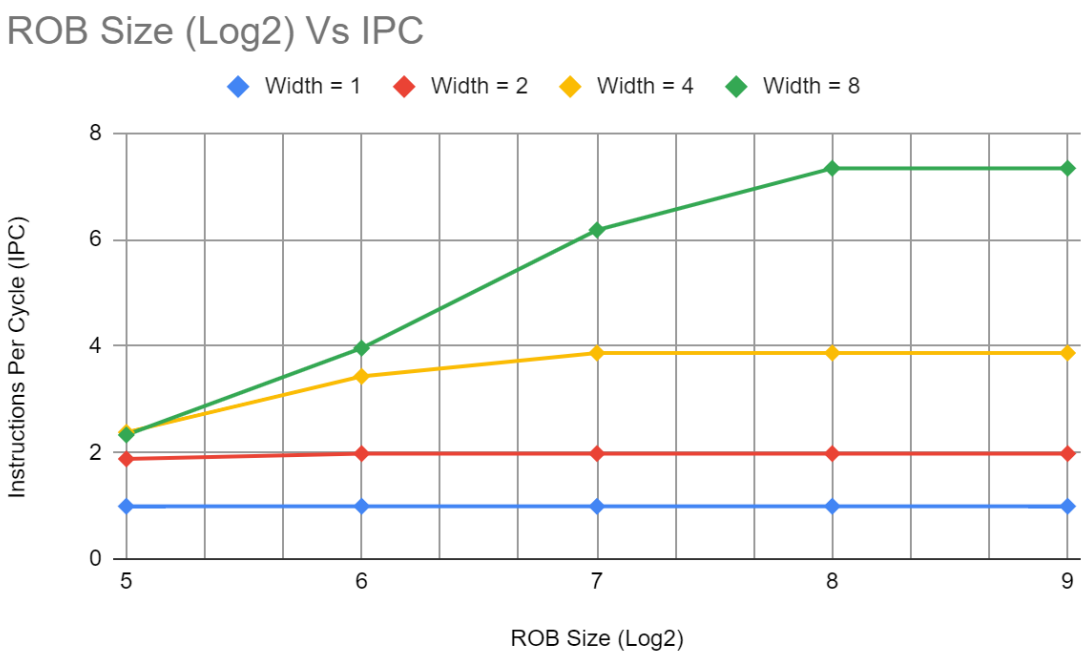
Optimized IQ_SIZE per WIDTH		
	GCC_Trace	PERL_Trace
WIDTH = 1	8	8
WIDTH = 2	16	32
WIDTH = 4	32	64
WIDTH = 8	64	128

We can see from the table that as we increase the width for a particular trace the IQ size required to get 5 % of the performance obtained by maximum IQ size of 256 gets increased for a fixed ROB Size. The reason being that as we increase the width of the pipe line, we are taking more instructions to the next stage which in turn requires bigger issue queue to hold more instructions. Thus, having a bigger issue queue requires higher width of the pipeline in order to utilize the larger issue queue with many independent instructions effectively.

From the graphs, we can identify that Perl_trace benchmark shows a lesser IPC count when compared to Gcc_trace benchmark for the same microarchitecture configuration. The reason for this behavior is that, Perl_trace contains many dependent instructions which stalls the pipeline causing a reduction in the IPC. The Gcc_trace does not contain as much dependent instructions as like perl_trace benchmark and so it shows a higher IPC value compared to perl_trace.

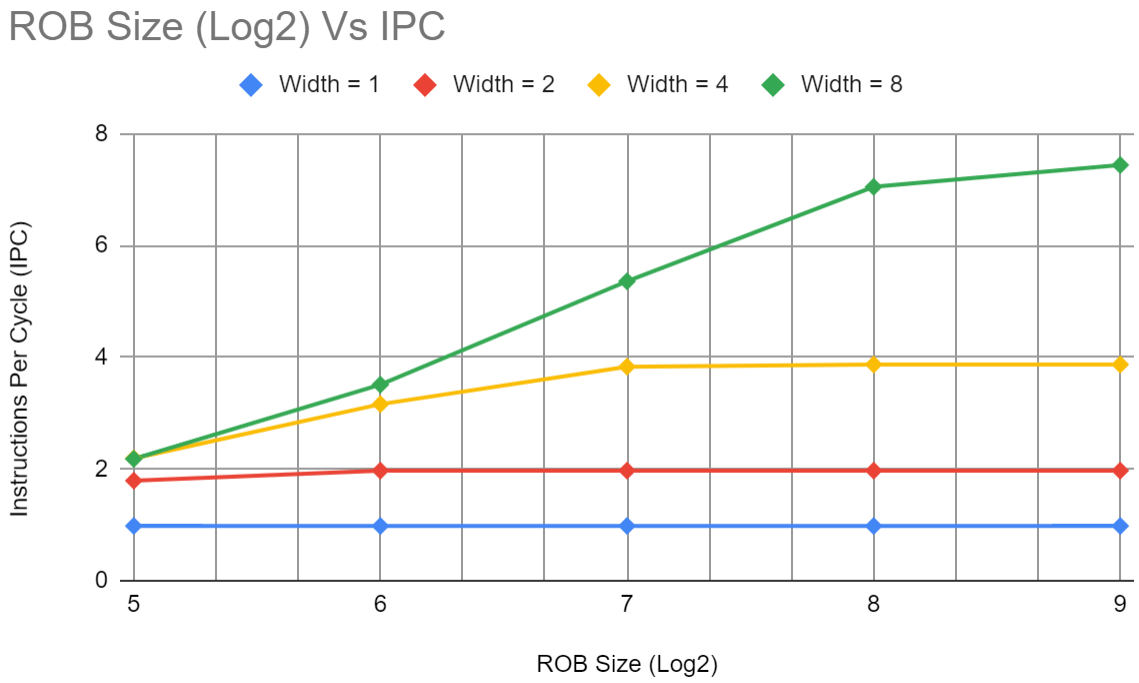
Analysis 2: Effect of ROB_Size

IPC_GCC					
ROB Size	ROB Size (Log2)	Width = 1 (IQ size = 8)	Width = 2 (IQ size = 16)	Width = 4 (IQ size = 32)	Width = 8 (IQ size = 64)
32	5	0.99	1.88	2.38	2.33
64	6	0.99	1.98	3.43	3.96
128	7	0.99	1.98	3.87	6.18
256	8	0.99	1.98	3.87	7.34
512	9	0.99	1.98	3.87	7.34



The graph is plotted for Reorder buffer size value in x axis and IPC along y axis for a fixed Issue queue size which is the optimal value obtained for each width from the previous graph. Having a larger width pipeline require a large reorder buffer size as a bundle of size width, independent instructions are being executed simultaneously. For smaller width of pipe line even though increasing the ROB size has no impact as the number of instructions being executed simultaneously is less and the IPC does not change.

IPC_Perl					
ROB Size	ROB Size (Log2)	Width = 1 (IQ size = 8)	Width = 2 (IQ size = 32)	Width = 4 (IQ size = 64)	Width = 8 (IQ size = 128)
32	5	0.98	1.79	2.19	2.18
64	6	0.98	1.97	3.16	3.51
128	7	0.98	1.97	3.83	5.36
256	8	0.98	1.97	3.87	7.05
512	9	0.98	1.97	3.87	7.44



From the graph, we can see that as we increase the ROB size for a fixed issue queue size (Determined from the optimal width), we can see an increase in IPC. The reason being that as the width of pipeline is higher, we need a larger ROB size to have multiple instruction's meta data, so that they can be run simultaneously. For smaller width of pipeline even though we increase the size of ROB it has less significant impact in IPC as in each cycle we take width number of instructions to the next stage where width is small.