NC State University

Department of Electrical and Computer Engineering

ECE 463/521: Fall 2015 (Rotenberg)

Project #3: Dynamic Instruction Scheduling

by

PARTH BHOGATE

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

Student's electronic signature: Parth Bhogate

Course number: ECE 521

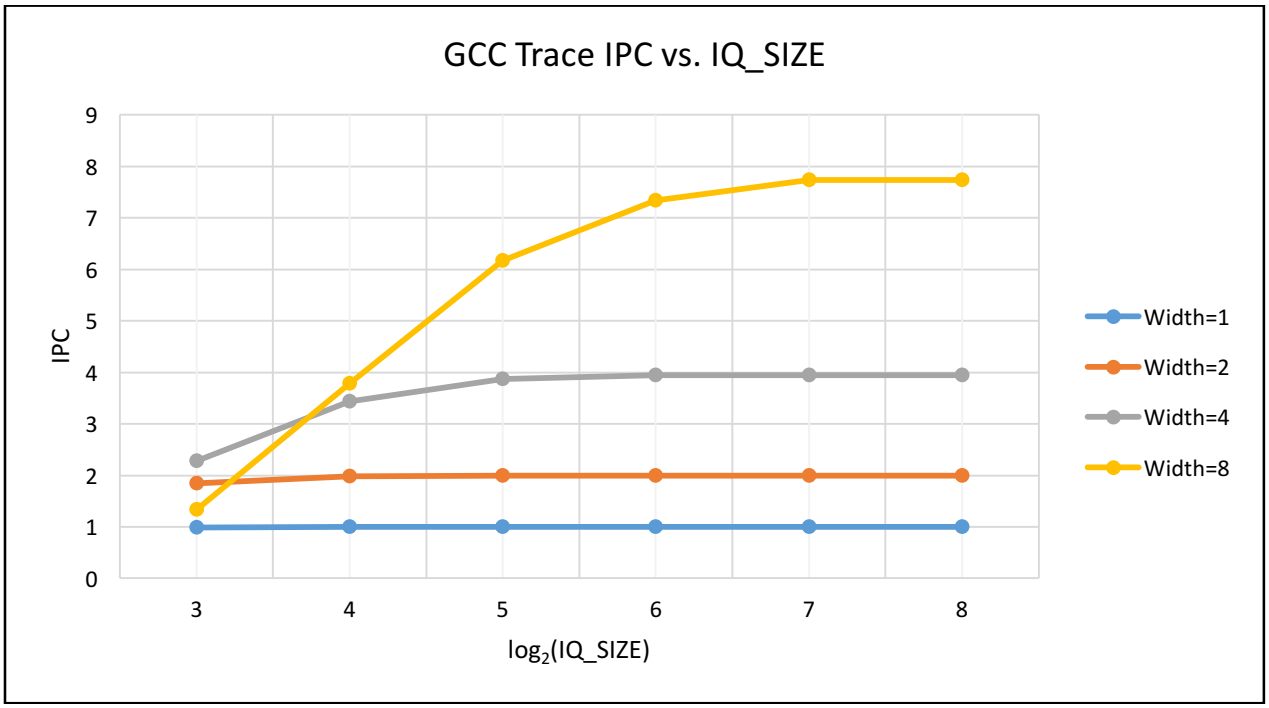
Analysis:

PART A: ROB_SIZE is kept constant at a sufficiently large value (so that it is not the performance bottleneck), IQ_SIZE and WDITH is varied to study the effect on IPC

ROB_SIZE = 512 entries (constant)

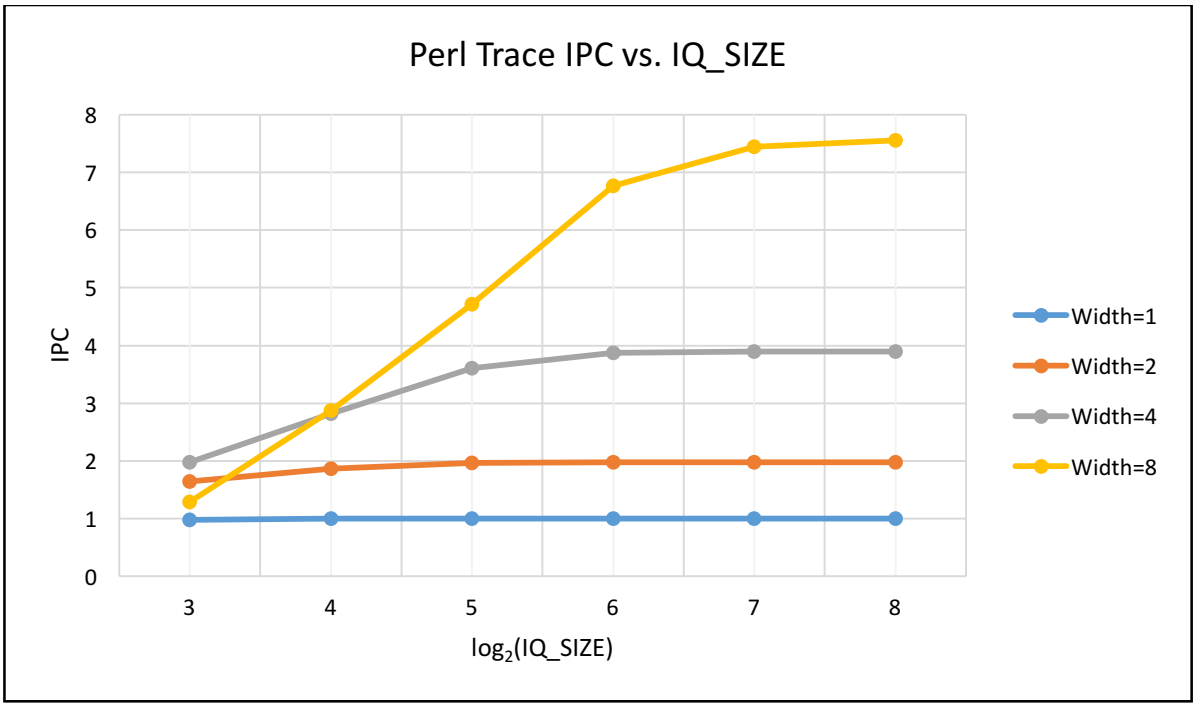
1) GCC TRACE

WIDTH	IQ_SIZE					
	8	16	32	64	128	256
1	0.99	1	1	1	1	1
2	1.85	1.98	1.99	1.99	1.99	1.99
4	2.28	3.43	3.87	3.94	3.94	3.94
8	1.33	3.78	6.17	7.34	7.73	7.73



2) PERL TRACE

WIDTH	IQ_SIZE					
	8	16	32	64	128	256
1	0.98	1	1	1	1	1
2	1.64	1.87	1.97	1.98	1.98	1.98
4	1.98	2.82	3.61	3.87	3.89	3.89
8	1.29	2.88	4.72	6.77	7.44	7.55



Optimal IQ_SIZE for each WIDTH:
Looking at the IPC numbers for each WIDTH configuration, the value of IPC which is within 5% of the best IPC for the highest IQ_SIZE (which is 256) is chosen. IQ_SIZE = 256 gives the highest value for the IPC, since with such a large IQ_SIZE it does not act as a bottleneck to processor performance.

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

Discussion: From the above graphs, we can draw the following conclusions -

1. Constant IQ_SIZE with increasing WIDTH: As observed in the above graphs, for a fixed IQ_SIZE, the IPC increases as the WIDTH is increased. This is quite expected since the number of instructions in-flight in the pipeline in each stage is determined by the WIDTH. As a result, in a processor with greater WIDTH, the number of instructions retiring every cycle is also greater.
The exception to this general observation is when the IQ_SIZE is almost equal to the WIDTH. In this scenario, the Dispatch stage is stalled quite often since it cannot issue WIDTH number of instructions unless IQ has WIDTH number of free entries in it. IQ_SIZE = WIDTH is an extreme example of this case, and the resulting reduction in IPC can be observed in the graph for WIDTH = 8, IQ_SIZE = 8, where the IPC for WIDTH = 8 turns out to be lesser than WIDTH = 2, 4 with IQ_SIZE = 8.

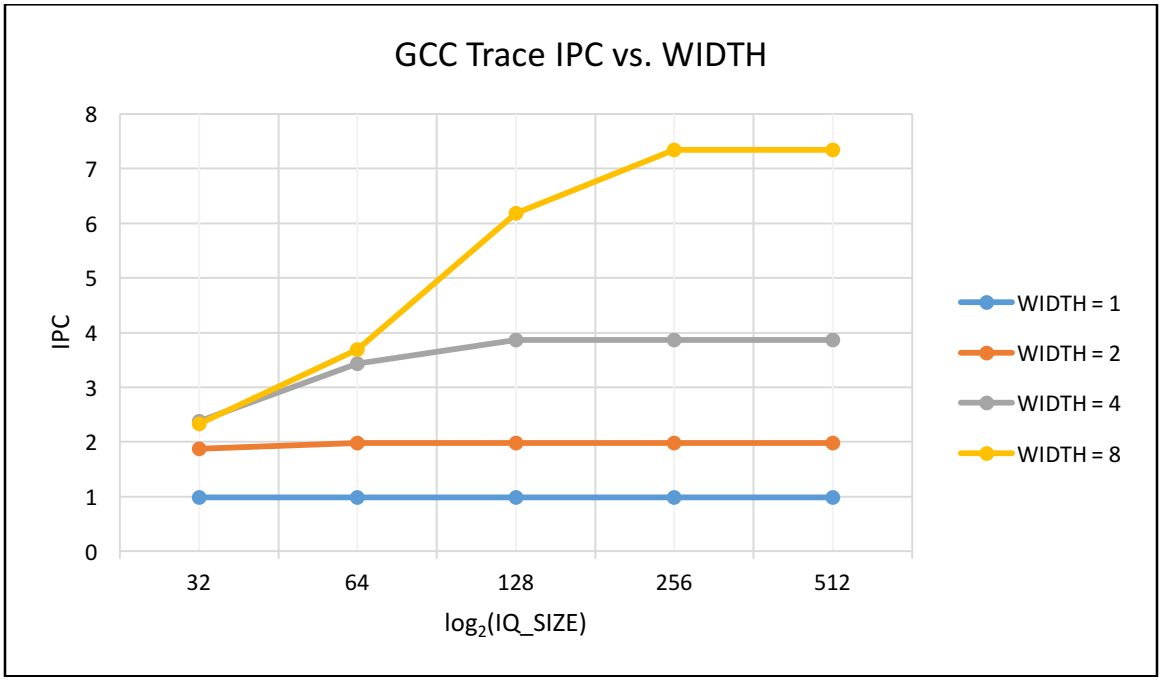
2. Constant WIDTH with varying IQ_SIZE: As we can observe in the graph, increasing the IQ_SIZE for a constant WIDTH leads to an increase in the IPC. With an increase in IQ_SIZE, it does not become a hardware bottleneck and thus leads to fewer stalls in the pipeline. However, with a sufficiently sized IQ, the IPC begins to level off as IQ_SIZE is no longer the performance bottleneck. This leveling off can be observed for IQ sizes greater than 64 entries.

3. Effect of Benchmark on IPC: Comparing the graphs obtained for GCC trace and Perl trace, we can see that they follow similar trends. However, the values of IPC for the GCC trace are a bit higher than those for the Perl trace. The benchmark instruction trace affects the IPC depending upon the number of RAW and WAR hazards in the pipeline. Presence of these hazards in the pipeline lead to a stall, and thus a decrement in the IPC. The most plausible reason for the lower IPC observed in the Perl trace is the presence of more hazards in the instruction sequence compared to the GCC trace. Also, the number of high latency instructions (instructions taking 5 cycles in the EX stage) in the Perl trace might be higher than GCC leading to a lower overall IPC.

PART B: Effect of varying ROB_SIZE, using the optimal IQ_SIZE values for each WIDTH obtained from Part A.

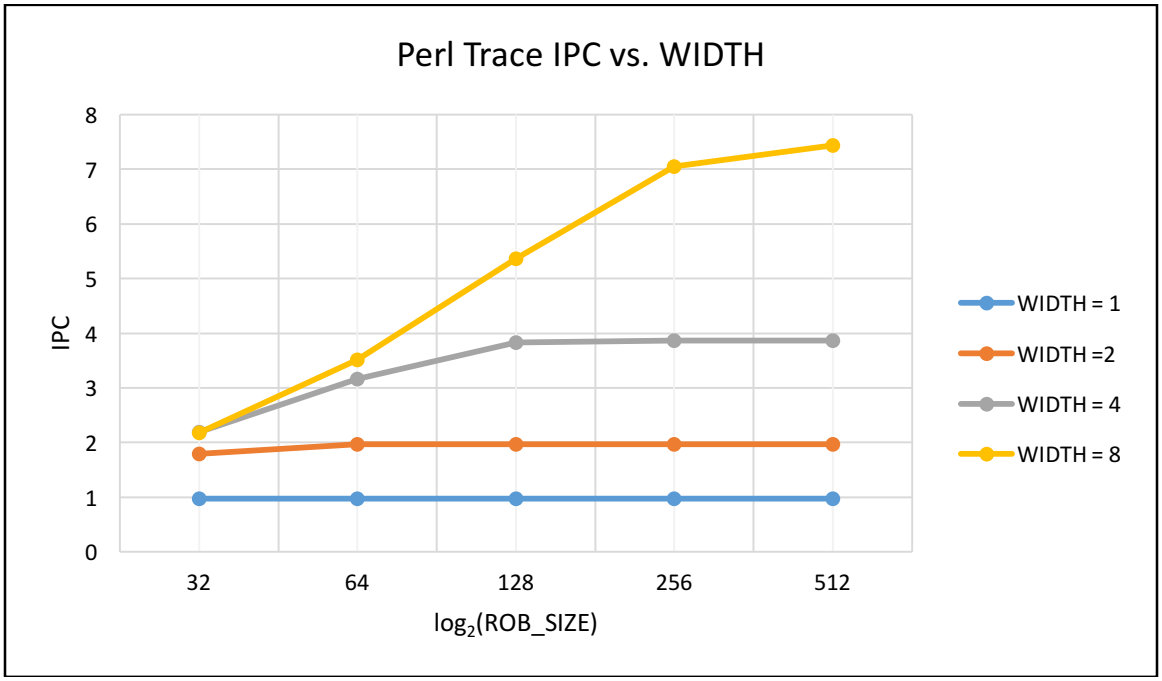
1) GCC TRACE:

ROB_SIZE	WIDTH = 1 IQ_SIZE = 8	WIDTH = 2 IQ_SIZE = 16	WIDTH = 4 IQ_SIZE = 32	WIDTH = 8 IQ_SIZE = 64
32	0.99	1.88	2.38	2.33
64	0.99	1.98	3.43	3.69
128	0.99	1.98	3.87	6.18
256	0.99	1.98	3.87	7.34
512	0.99	1.98	3.87	7.34



2) PERL TRACE:

ROB_SIZE	WIDTH = 1 IQ_SIZE = 8	WIDTH = 2 IQ_SIZE = 32	WIDTH = 4 IQ_SIZE = 64	WIDTH = 8 IQ_SIZE = 128
32	0.98	1.79	2.19	2.18
64	0.98	1.97	3.16	3.51
128	0.98	1.97	3.83	5.36
256	0.98	1.97	3.87	7.05
512	0.98	1.97	3.87	7.44



Discussion: From the graphs plotted for varying ROB_SIZE, the following conclusions can be drawn-

1. Effect of increasing ROB_SIZE with constant WIDTH: As observed in the graph, when the ROB_SIZE is increased keeping the WIDTH constant, the IPC increases. This is due to the fact that with increase in ROB_SIZE, it no longer continues to be a hardware resource bottleneck. With a sufficiently sized ROB, the IPC begins to level off since ROB_SIZE no longer remains a performance determining factor. This leveling off can be observed when ROB size becomes 128, 256 or 512 entries.

2. Effect of increasing WIDTH at constant ROB_SIZE: When the WIDTH is increased at a constant ROB_SIZE, the IPC keeps on increasing. This is due to the fact that a processor with a greater width has more instructions in-flight in each stage of the pipeline.