Project Writeup - Tomasulo's Algorithm for FOCO realization

Rajath Shetty (902825666)

Experimentation:

Approach:

First step was to run the code for different values with an aim to achieve saturation for Retired Instructions per Cycle (RPC). However, while using only 3 values per parameter, a saturation point cannot be accurately determined. Therefore, I've run more iterations with different value to arrive at the ideal solution.

Another factor is the compromise between hardware used and RPC. I have used a ratio of Hardware used to RPC as the first level of filtering. The lower the value, the better the hardware utilization.

Problem states that we need to find values as close to the max RPC as possible, therefore I have found the max. RPC value and found the value of least hardware within **75% of the maximum RPC.**

Code: In procsim.cpp enable #define DEBUG for debug traces and #define OUTPUT (already enabled) for output traces.

GCC Trace:

Max value of RPC: 3.777434
This was obtained for all maximum possible parameter values which is
R=128, K0=3, K1=3, K2=3, F=8, M=8,
Hardware Count = 345

Best results considering hardware limitations **(for 75% of max RPC):**

R	K0	K1	K2	F	M	HW	RPC
32	3	2	2	8	8	135	2.834869

Best results considering stricter hardware limitations **(below 75% of max RPC):**

R	K0	K1	K2	F	M	HW	RPC	Cycles
32	3	2	2	4	4	103	2.649007	37750

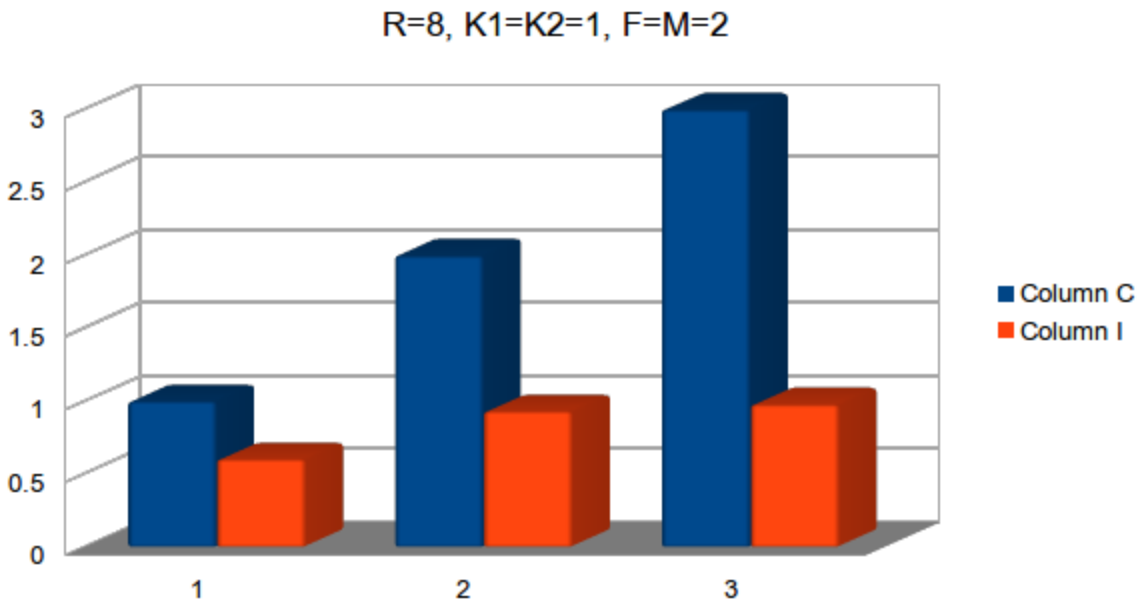
The following shows how these results were obtained:

Project Writeup - Tomasulo's Algorithm for FOCO realization

Rajath Shetty (902825666)

Saturation after individual tests were obtained for the following values:

k0=3, k1=2, k2=2, f=4, m=4, r=32.
Determining K0:
Column C = k0
Column I = RPC



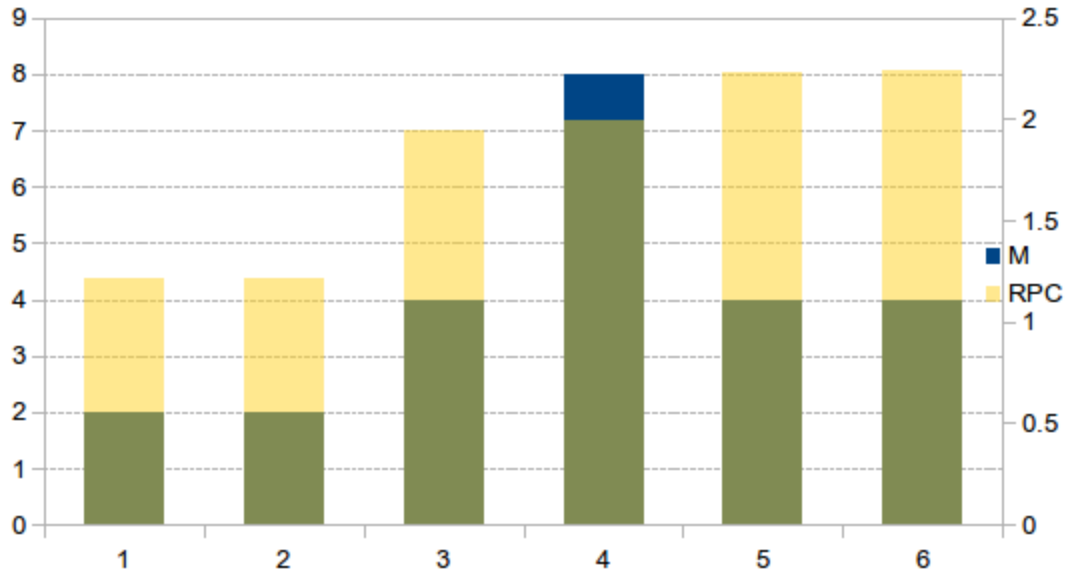
K0 continues to increase until k0=3, therefore we can assume that gcc trace is heavily dependent on k0 being large.
However, increasing value of k1 and k2 didn't seem to have any significant effects. Therefore, maintain k1 and k2 as 1 for now.

We can also reason that increasing the multiplier without increasing ROB size will not give us the desired effects because the schedule queue will only contain as many entries as the ROB holds at any point in time.

Project Writeup - Tomasulo's Algorithm for FOCO realization

Rajath Shetty (902825666)

Therefore, intuitively increase value of R to 32. Along with this we can now start experimenting with values of M
The



The RPC values are aligned with secondary y-axis on the right hand side to give more visible changes. (M and RPC are overlapped, shown as different colors)
The following values were used in the above 6 trials

F	M	HW	RPC
2	2	81	1.212974
4	2	83	1.213003
2	4	91	1.942162
2	8	111	1.998321
4	4	93	2.235986
8	4	97	2.238288

As we can see saturation occurs at m=4 and f=4. In fact increase in fetch rate is effective only for values of m beyond 2.

Therefore the best hardware which delivers RPC within **75% of max RPC** is the following

R	K0	K1	K2	F	M	HW	RPC
32	3	2	2	8	8	135	2.834869

For further hardware optimization we can choose the following which is 70% of max RPC

R	K0	K1	K2	F	M	HW	RPC	Cycles
32	3	2	2	4	4	103	2.649007	37750

The above has the best ratio of HW to RPC for all values of RPC outside of 75% ie among all RPC values from 0 to 75% of max RPC.

Trace GOBMK

Max value of RPC: 3.146732
This was obtained for all maximum possible parameter values which is
R=128, K0=3, K1=3, K2=3, F=8, M=8,
Hardware Count = 345

Best results considering hardware limitations **(for 75% of max RPC):**

R	K0	K1	K2	F	M	HW	RPC
32	3	2	2	8	8	135	2.394751

Best results considering stricter hardware limitations **(below 75% of max RPC):**

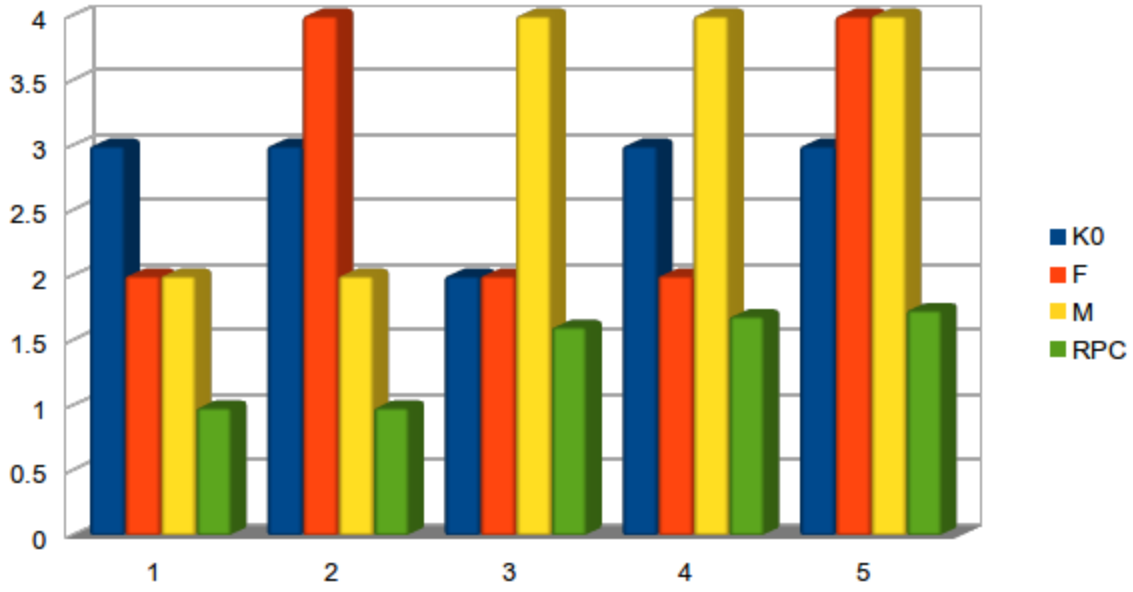
R	K0	K1	K2	F	M	HW	RPC	Cycles
32	3	2	2	4	4	103	2.255096	44344

Project Writeup - Tomasulo's Algorithm for FOCO realization

Rajath Shetty (902825666)

The following shows how these results were obtained:

To determine saturation values, similar tests were conducted as with gcc trace and the following graph shows how the saturation values were determined.



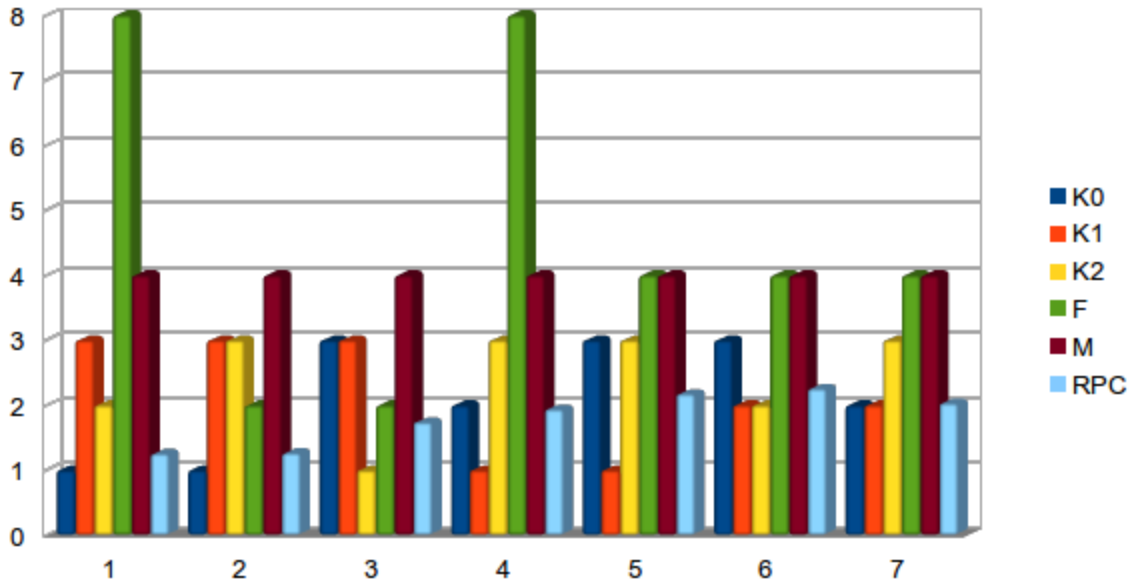
The above graph is a reflection of the following values:

R	K0	K1	K2	F	M	HW	RPC
32	3	1	1	2	2	81	0.987333
32	3	1	1	4	2	83	0.987362
32	2	1	1	2	4	86	1.605703
32	3	1	1	2	4	91	1.689161
32	3	1	1	4	4	93	1.737197

Now to arrive at saturation values of k0, k1, k2, the following graph is useful.

Project Writeup - Tomasulo's Algorithm for FOCO realization

Rajath Shetty (902825666)



The above graph is tabulated using these values:

R	K0	K1	K2	F	M	RPC
32	1	3	2	8	4	1.25097
32	1	3	3	2	4	1.262069
32	3	3	1	2	4	1.734846
32	2	1	3	8	4	1.936671
32	3	1	3	4	4	2.171977
32	3	2	2	4	4	2.255096
32	2	2	3	4	4	2.028151

This table shows that saturation is reached for K0, K1 & K2 at 3, 2 & 2 along with f=4 and m=4.

Beyond this we can increase value of f and m to reach 75% of max RPC value.

Project Writeup - Tomasulo's Algorithm for FOCO realization

Rajath Shetty (902825666)

Trace : HMMER

Max value of RPC: 2.952553
This was obtained for all maximum possible parameter values which is
R=128, K0=3, K1=3, K2=3, F=8, M=8,
Hardware Count = 345

Best results considering hardware limitations **(for 75% of max RPC):**

R	K0	K1	K2	F	M	HW	RPC
32	3	2	3	8	8	144	2.217393

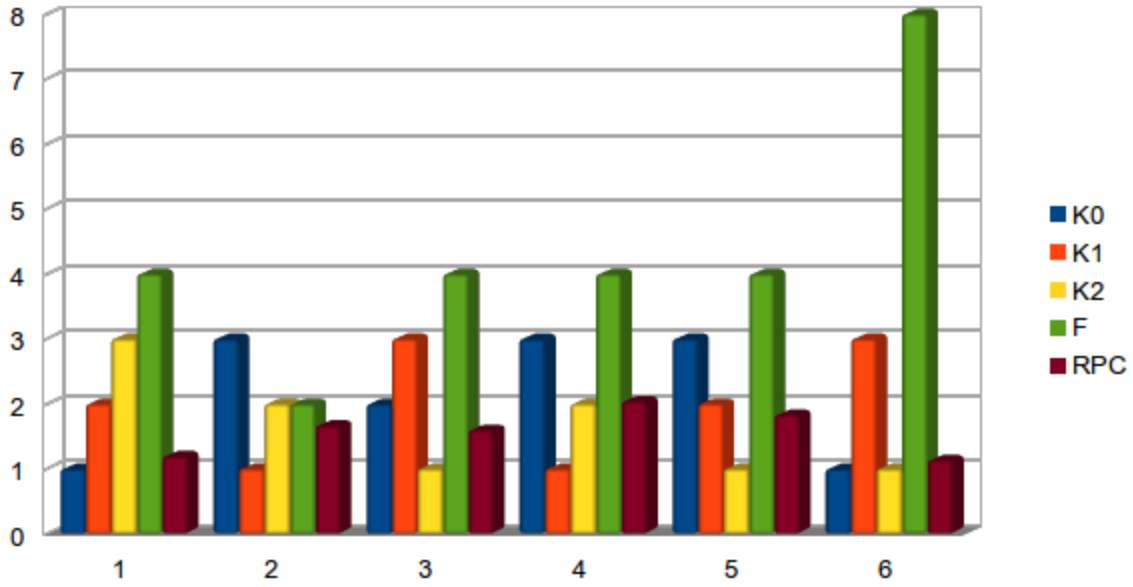
Best results considering stricter hardware limitations **(below 75% of max RPC):**

R	K0	K1	K2	F	M	HW	RPC	Cycles
32	3	1	2	4	4	98	2.040858	48999

The following shows how these results were obtained:

Project Writeup - Tomasulo's Algorithm for FOCO realization

Rajath Shetty (902825666)



The above graph was computed using this table:

R	K0	K1	K2	F	M	HW	RPC
32	1	2	3	4	4	98	1.194715
32	3	1	2	2	4	96	1.666861
32	2	3	1	4	4	98	1.601127
32	3	1	2	4	4	98	2.040858
32	3	2	1	4	4	98	1.833248
32	1	3	1	8	4	97	1.135422

The above values helps in getting close to the saturation values of k0, k1 and k2.

However, this value (highlighted) is not 75% of the max RPC value for this trace. Therefore, we by increasing value of F & M we can obtain a good solution which is closer to 75% of the max RPC.

Project Writeup - Tomasulo's Algorithm for FOCO realization

Rajath Shetty (902825666)

Trace MCF

Max value of RPC: 4.252604
This was obtained for all maximum possible parameter values which is
R=128, K0=3, K1=3, K2=3, F=8, M=8,
Hardware Count = 345

Here, the value of RPC shoots up to very high values for last few iterations of hardware resources. Therefore, I have averaged out the last 15 values of RPC and use this as the average max RPC.
Avg. Max RPC = 3.729

I've reduced the range to 70% given the large values of RPC.

Best results considering hardware limitations **(for 70% of max RPC):**

R	K0	K1	K2	F	M	HW	RPC
32	3	2	3	8	8	144	2.714146

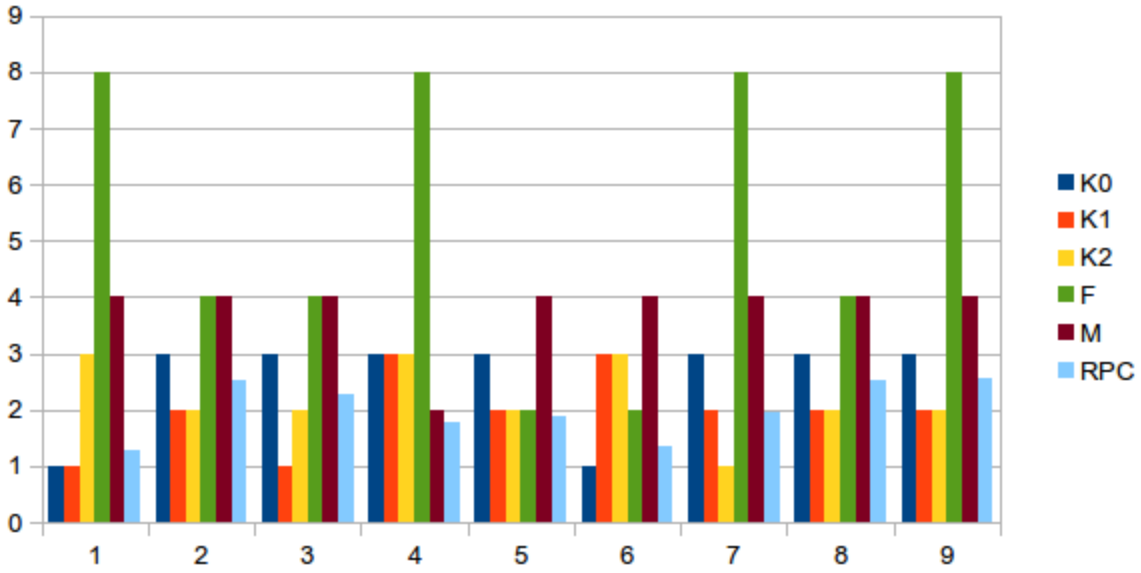
Best results considering stricter hardware limitations **(below 70% of max RPC):**

R	K0	K1	K2	F	M	HW	RPC	Cycles
32	3	2	2	4	4	103	2.518638	39704

The following shows how these results were obtained:

Project Writeup - Tomasulo's Algorithm for FOCO realization

Rajath Shetty (902825666)



The above graph is obtained from the below table:

R	K0	K1	K2	F	M	HW	RPC
32	1	1	3	8	4	97	1.282972
32	3	2	2	4	4	103	2.518638
32	3	1	2	4	4	98	2.284357
32	3	3	3	8	2	99	1.785459
32	3	2	2	2	4	101	1.887825
32	1	3	3	2	4	101	1.344086
32	3	2	1	8	4	102	1.942087
32	3	2	2	4	4	103	2.518638
32	3	2	2	8	4	107	2.541038

From the above table, we can see that saturation occurs around the values which are highlighted above. However, if we are to increase the RPC to a value closer to 70% of max RPC, then we need to iteratively increase the value for each hardware component until we reach a value which can give a good trade off between hardware used and RPC.