CS60003 : High Performance Computer Architecture

Spring 2021

Assignment 1 - Report

Submitted by: Group - $I_{\{eye\}}$

20CS60R52	Yogesh Porwal	porwalyogesh715@gmail.com
20CS60R53	Suprajit Sardar	primesupra220@gmail.com
20CS60R55	Tushika Agrawal	mkagarwal7723@gmail.com
20CS60R56	Vineeth Kumar Balapanuru	vineethkumar6001@gmail.com
20CS60R57	Vaibhav Saxena	vaibhav.vbv121094@gmail.com
20CS60R58	Km Simran Jaiswal	jaiswalsimran153@gmail.com
20CS60R59	Vivek Kumar Arya	vivekarya599@gmail.com
20CS60R60	Ravi Pratap Singh	areweravi@gmail.com
20CS60R61	Murtaza Mustafa Saifee	murtaza.guitarz@gmail.com
20CS60R62	Nikhil Kumar	nikhilk128@gmail.com
20CS60R63	Mukesh Kumar Gautam	mukesh.pmail926@gmail.com
20CS60R64	Sahil Jain	sahiljain1214@gmail.com
20CS60R65	Saurav Koranga	sauravkorangaa@gmail.com

Department of Computer Science & Engineering,

IIT Kharagpur

PROBLEM STATEMENT

To run a benchmark program on different system configurations on gem5 and analyse the output statistics of each of these config combinations to select top 10 combinations.

IMPLEMENTATION

The benchmark program towers.c was run on the Linux System where we have the Gem5 installed.

(a) Based on the CPI values, the top 10 configurations for the benchmark program

Constant parameters :

Sno	Parameter	Value
1	$CPU\ model$:	$Out\ -\ of\ -\ Order\ (DerivO3CPU)$
2	$Caches$:	$L1I,\ L1D,\ L2\ (set\ associative)$
3	$Clock\ Frequency$:	$2GHz$
4	$Memory\ mode$:	$Timing$
5	$Memory\ size$:	$1GB$
6	$Memory\ controller$:	$MemCtrl()$
7	$DRAM\ type$:	$DDR3\ 1600\ 8x8()$
8	$NumberofReorderBuffer$:	1
9	$l1\ tag\ latency$:	2
10	$l1\ data\ latency$:	2
11	$l1\ response\ latency$:	2
12	$l1\ mshrs$:	4
13	$l1\ tgts\ per\ mshr$:	20
14	$l2\ tag\ latency$:	20

15	$l2\ data\ latency$:	20
16	$l2\ response\ latency$:	20
17	$l2\ mshr$:	20
18	$l2\ tgts\ per\ mshr$:	12
19	$No\ of\ cache\ lines$:	64

Variable parameters :

Sno	$l1d\ size(B)$	$l1i\ size(B)$	$l2\ size(B)$	$l1\ assoc$	$l2\ assoc$	$bp\ type$	$LQEntries$	$SQEntries$	$ROBEntries$	$numIQEntries$
1	65536	65536	262144	8	8	TournamentBP	16	64	192	64
2	65536	65536	262144	8	8	TournamentBP	16	64	128	64
3	65536	32768	131072	8	4	TournamentBP	16	64	128	64
4	65536	65536	524288	2	4	TournamentBP	16	64	192	64
5	32768	32768	262144	4	4	TournamentBP	16	64	192	64
6	65536	65536	524288	2	4	TournamentBP	16	64	128	64
7	32768	65536	262144	2	8	TournamentBP	16	64	192	64
8	32768	32768	262144	2	8	TournamentBP	16	64	128	64
9	32768	32768	262144	2	8	TournamentBP	16	64	192	64
10	65536	65536	262144	8	8	TournamentBP	16	32	128	64

(b) Why do these combinations of parameters work best for the given benchmark program ?

Explanation -

The problem source code (towers.h) is a tower of hanoi problem variation with 3 pegs and n=7 discs. The code has the worst case time complexity of $O(2^n)$.

The size of the source code file is 22 KB, this tells that instruction cache size more than 22KB would be sufficient for this and it can also be concluded from the top 10 configurations where 40% are 32KB and rest 60% are 64KB.

One major finding is that L2 associativity of 8 gave best results when the size of L2 cache is taken as 256KB. We can conclude that choosing the maximum associativity for the L2 cache for size 256KB will reduce the cache misses and thus increase accuracy.

Tournament BP: Since Tournament BP combines both gshare and bimodal, it is better than predicting alone therefore all the top configurations used tournament BP as branch predictor.

For SQ, ROB and IQ parameters, the maximum size for each gave the best result for benchmark configurations.

1) **For CPI -**
    a) The cycles per instruction(CPI) decreases due to the fact that there is less miss penalty due to a higher hit-rate.
    b) It can be seen that the L1D hit-rate increases as the L1D cache size increases.
    c) In config 2 and 8, as SQEntries decreases for config 10, CPI is increased.
2) **For Mispredicted branches detected during execution -**
    a) It can be observed in config pair (1,2), (8,9), (6,7) that as we increase ROB entries, the branch misprediction also increases because ?
    b) The difference in the branch misprediction was negligible in (4,5) as can be seen in the graph, although they have different L1 and L2 cache sizes because the array was linear. The L1 cache size was always less than L2 cache size to improve access latency.
3) **For Number of branches that were predicted not taken incorrectly -**
    a) When associativity of either one of the caches is on the lower side could be the reason for a lower number of branches that were predicted not taken incorrectly .

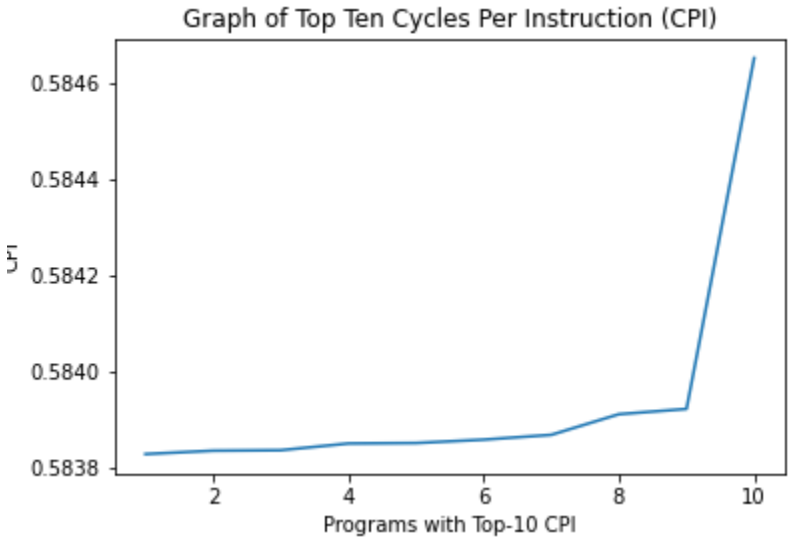
b) With low associativity of l1 cache , high BTB hit percentage was observed.

4) **For Number of branches that were predicted taken incorrectly -**
   a) This parameter was constant so seems like it has no effect on different top 10 configurations .
5) **For IPC -**
   a) The difference in the IPC was negligible as can be seen in the line graph.
6) **BTB hit percentage -**
   a) High BTB hit percentage was observed with low l1 and l2 cache sizes along with their associativity .
   b) With 32KB l1 & l2 cache sizes , high BTB hit was with low number of ROB entries.
7) **Overall miss cycles, miss rate, average overall miss latency -**
   a) Large number of miss cycle was with high associativity and large cache sizes
   b) Low mis rate was with large cache sizes and low associativity
   c) Average overall miss latency was with parameters with large miss cycles .
8) **Number of ROB accesses (read and write both) -**
   a) With less number of ROB entries lesser ROB accesses were seen, the reason for this could be due to large cache sizes in that configuration as compared to others.
9) **Number of times the LSQ has become full, causing a stall-**
   a) Number of times the LSQ has become full was high when associativity of the l1 & l2 cache and number of ROB entries were less.
10) **Number of loads that had data forwarded from stores -**
   a) Since this is the inverse of the previous statement , less number of loads that had forwarded from stores were observed when there was high associativity of the l1 & l2 cache and number of ROB entries were less.
11) **Number of times access to memory failed due to the cache being blocked**
   a) This parameter was constant so seems like it has no effect on different top 10 configurations .

(c) Graph or plot for each of the top 10 combinations :

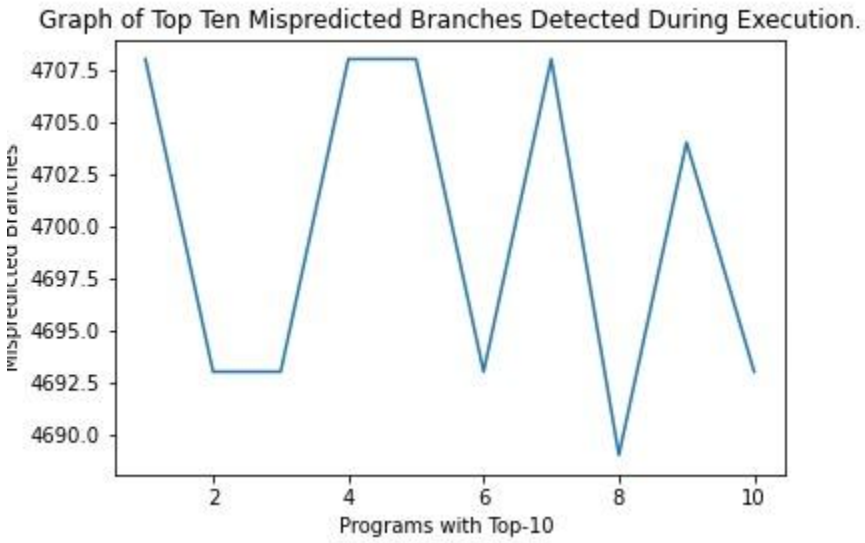
– Cycles Per Instruction (CPI) :

0.583829	0.583836	0.583837	0.58385	0.583852	0.583859	0.583869	0.583912	0.583923	0.584654



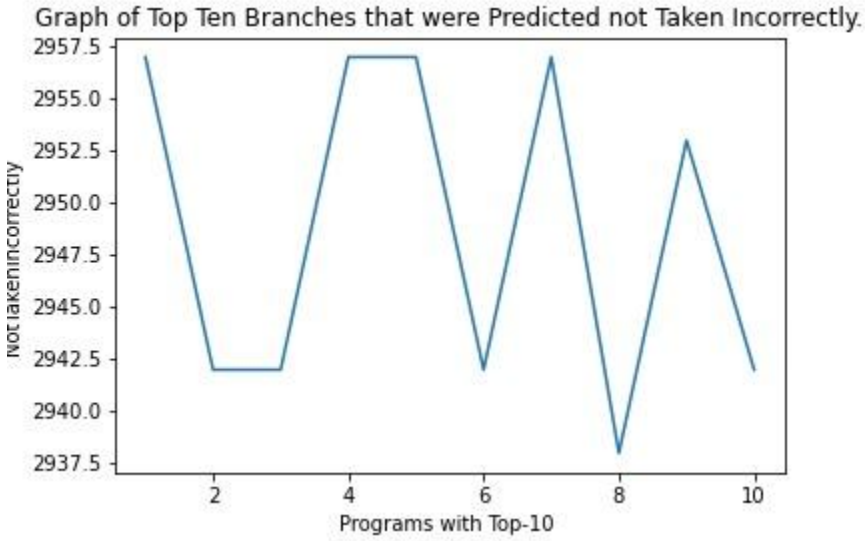
– Mispredicted branches detected during execution.

4708	4693	4693	4708	4708	4693	4708	4689	4704	4693



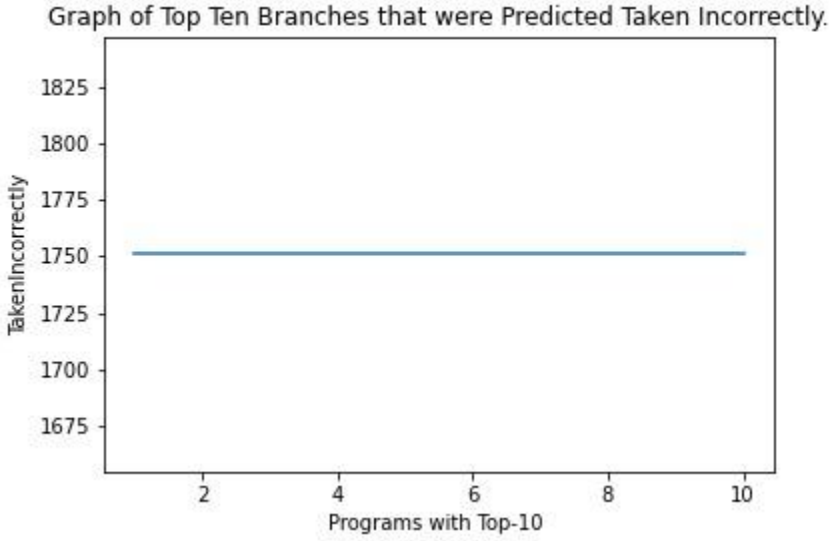
– Number of branches that were predicted not taken incorrectly.

2957	2942	2942	2957	2957	2942	2957	2938	2953	2942



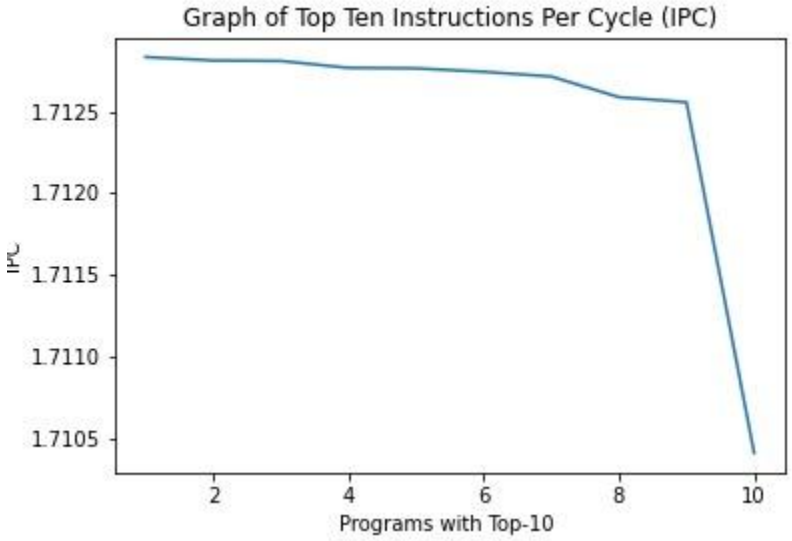
– Number of branches that were predicted taken incorrectly.

1751	1751	1751	1751	1751	1751	1751	1751	1751	1751



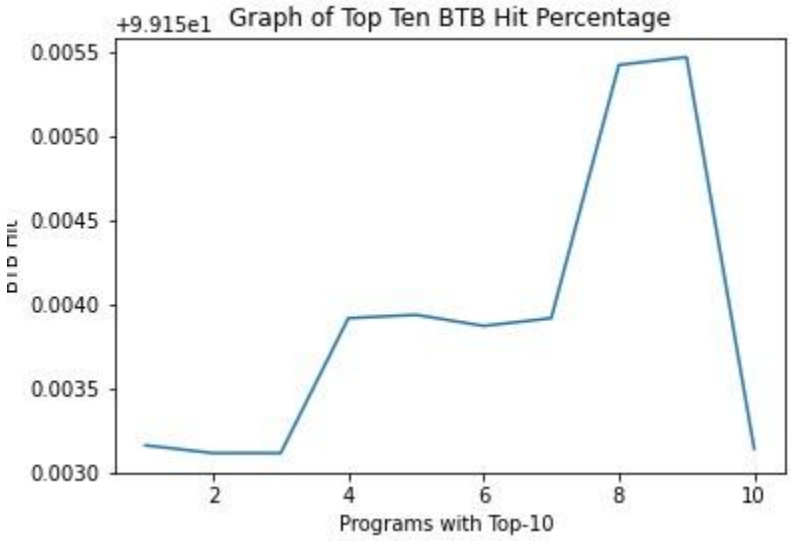
– Instructions Per Cycle (IPC).

1.712832	1.712811	1.712808	1.712766	1.712763	1.712742	1.712713	1.712587	1.712556	1.710413



– Number of BTB hit percentage.

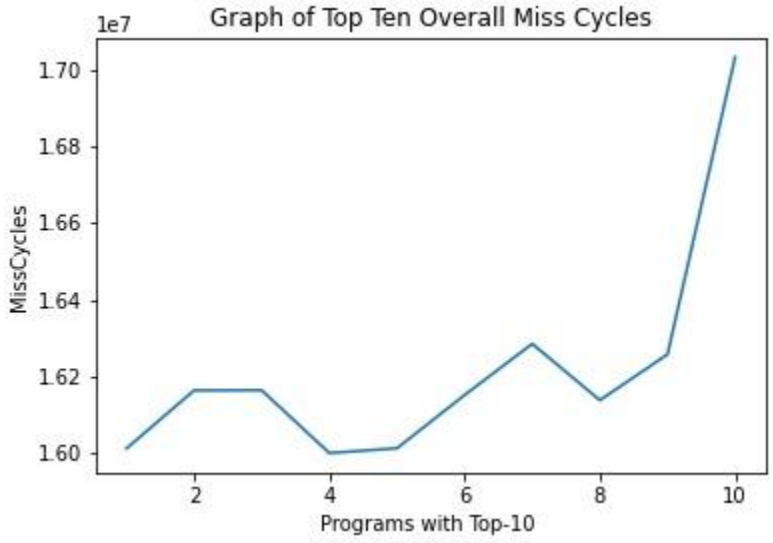
99.153159	99.153113	99.153113	99.153915	99.153935	99.153869	99.153915	99.155421	99.155468	99.153139



– Number of overall miss cycles, miss rate, average overall miss latency.

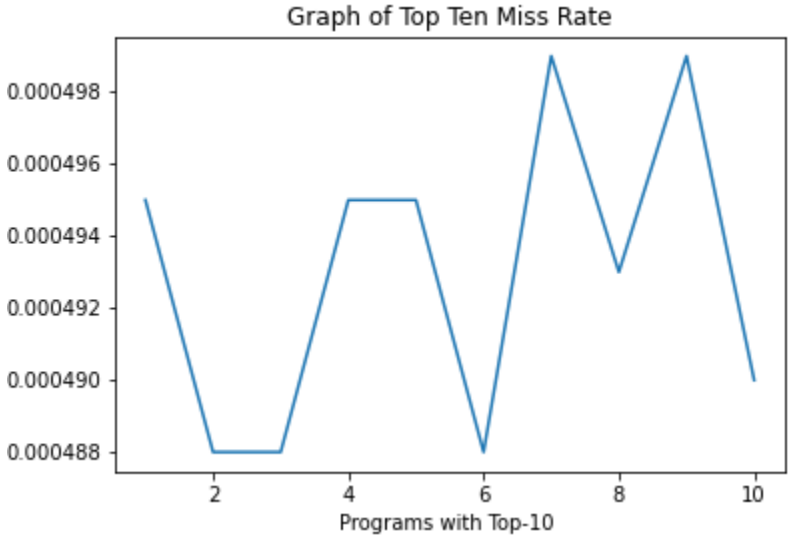
Overall Miss Cycle :

16012999	16163999	16164499	16000999	16013499	16151999	16285499	16138499	16257999	17031999



Overall Miss Rate :

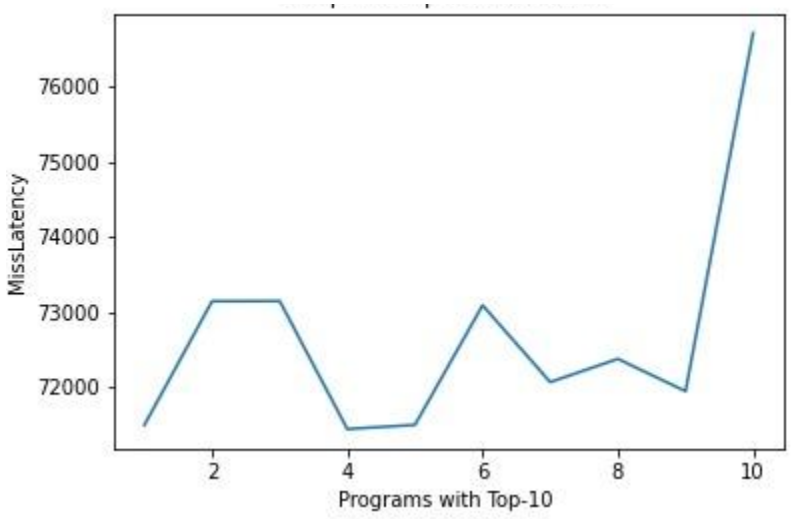
0.000495	0.000488	0.000488	0.000495	0.000495	0.000488	0.000499	0.000493	0.000499	0.00049



Avg. overall Miss Latency :

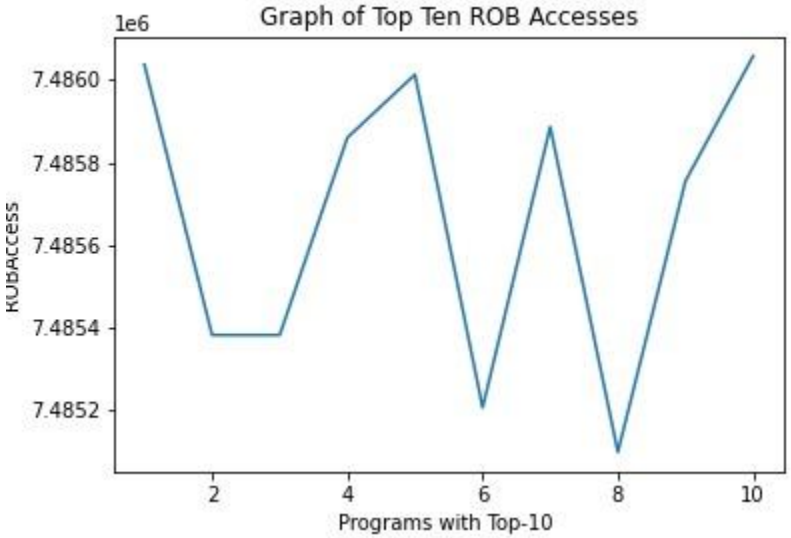
71486.602	73140.266	73142.529	71433.031	71488.834	73085.968	72059.730	72369.950	71938.048	76720.716

Graph of top 10 average overall miss latency



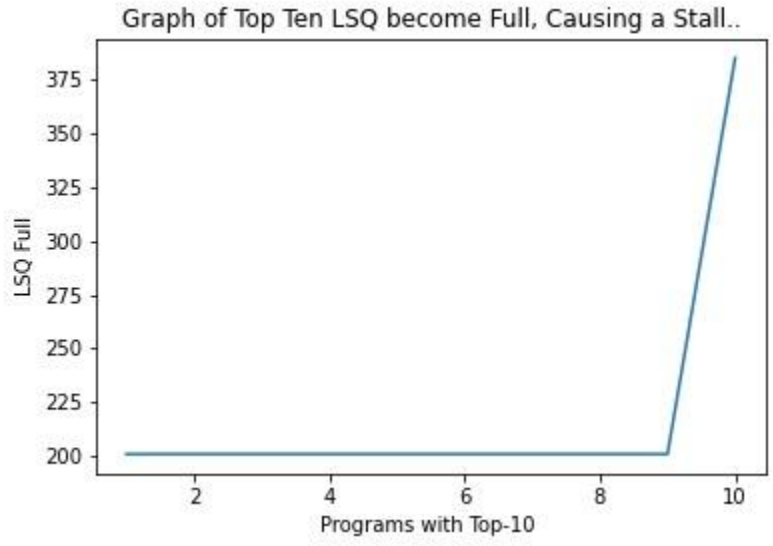
– The number of ROB accesses (read and write both).

7486037	7485381	7485381	7485860	7486013	7485205	7485886	7485097	7485755	7486058



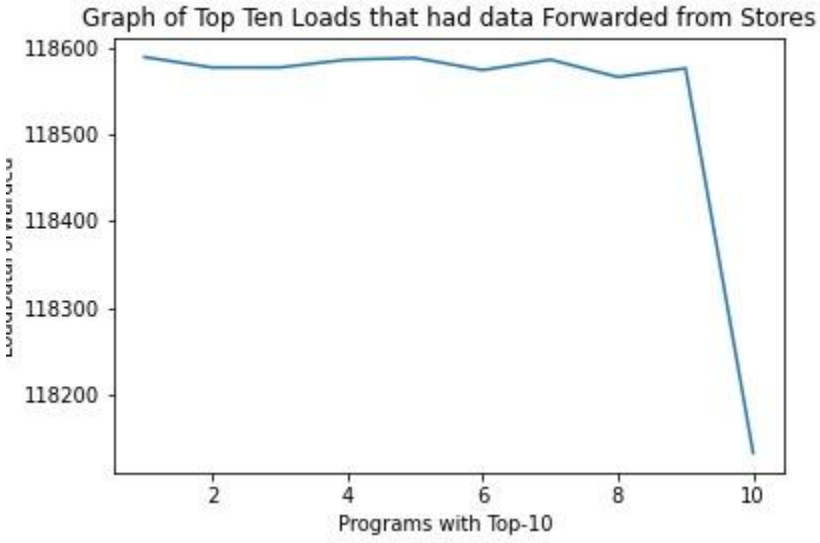
– Number of times the LSQ has become full, causing a stall.

201	201	201	201	201	201	201	201	201	385



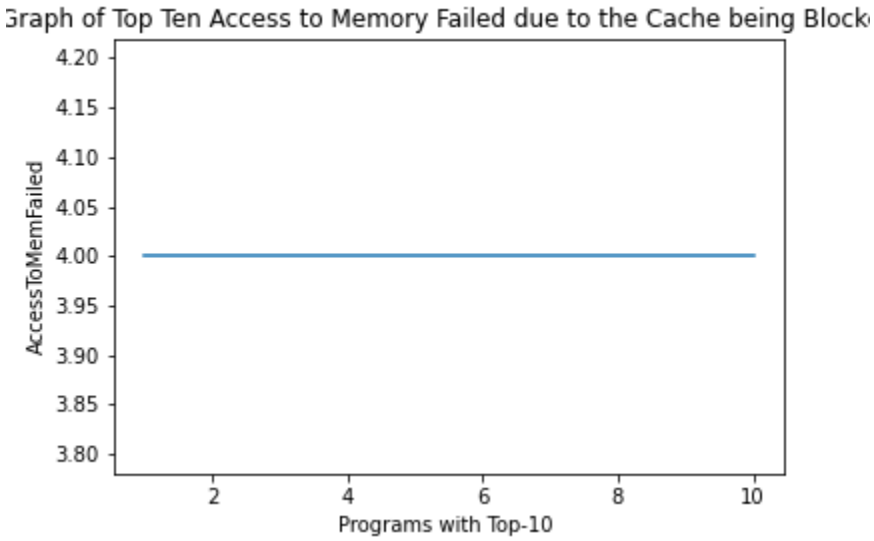
– Number of loads that had data forwarded from stores.

118589	118577	118577	118586	118588	118574	118586	118566	118576	118133



– Number of times access to memory failed due to the cache being blocked

4	4	4	4	4	4	4	4	4	4



Individual contribution of group members to this project :

{

1. Understanding the problem statement.

2. Understanding gem5 parameters & how everything works together and explaining to all members and co-ordinator of each phase.

3. Developing custom configuration script.

4. Provided assistance in installation of gem5 and running simulation(equal division).

5.Anlaysing the stats.txt.

6.Creating Graphs plots

7.Report.

}

Member name	Contribution
Yogesh Porwal	1,2,5
Suprajit Sardar	1,3,6
Tushika Agrawal	1,5,7
Vineeth Kumar Balapanuru	1,6,7
Vaibhav Saxena	1,5,7
Km Simran Jaiswal	1,5,7
Vivek Kumar Arya	1,6,7
Ravi Pratap Singh	1,3,4
Murtaza Mustafa Saifee	1,6,7
Nikhil Kumar	1,3,4
Mukesh Kumar Gautam	1,6,7
Sahil Jain	1,3,4
Saurav Koranga	1,4,7