Indian Institute of Information Technology, Allahabad

PROJECT REPORT

LOW POWER OPTIMIZATION FOR GCC COMPILER

Project Supervisor - Dr. Bibhas Ghoshal

Declaration by the Candidates

We, hereby declare that the project titled *Low Power Optimization for GCC Compiler* is a record of bonafide project work carried out by us under the guidance of *Dr. Bibhas Ghoshal* in partial fulfillment of the 5th semester Mini-Project work for the B.Tech (IT) Course in Indian Institute of Information Technology, Allahabad.

Nishit Gupta – IIT2014502

Sachin Agarwal – IIT2014501

Saurabh Tanwar – IIT2014140

D. Rajeswar Rao – IIT2014055

Certificate

This is to certify that the project report entitled *Low Power Optimization for GCC Compiler* submitted to Department of Information Technology, Indian Institute of Information Technology, Allahabad in partial fulfillment of the 5th semester Mini-Project work, is a record of bonafide work carried out by :

1. Nishit Gupta – IIT2014502
2. Sachin Agarwal – IIT2014501
3. Saurabh Tanwar – IIT2014140
4. D. Rajeswar Rao – IIT2014055

under my supervision and guidance.

This report has not been submitted anywhere else for any other purpose.

Submission Date : 20/09/2016

Dr.Bibhas Ghoshal
Assistant Professor
Department of Information Technology
Indian Institute of Information Technology
Allahabad - 211012

CONTENTS

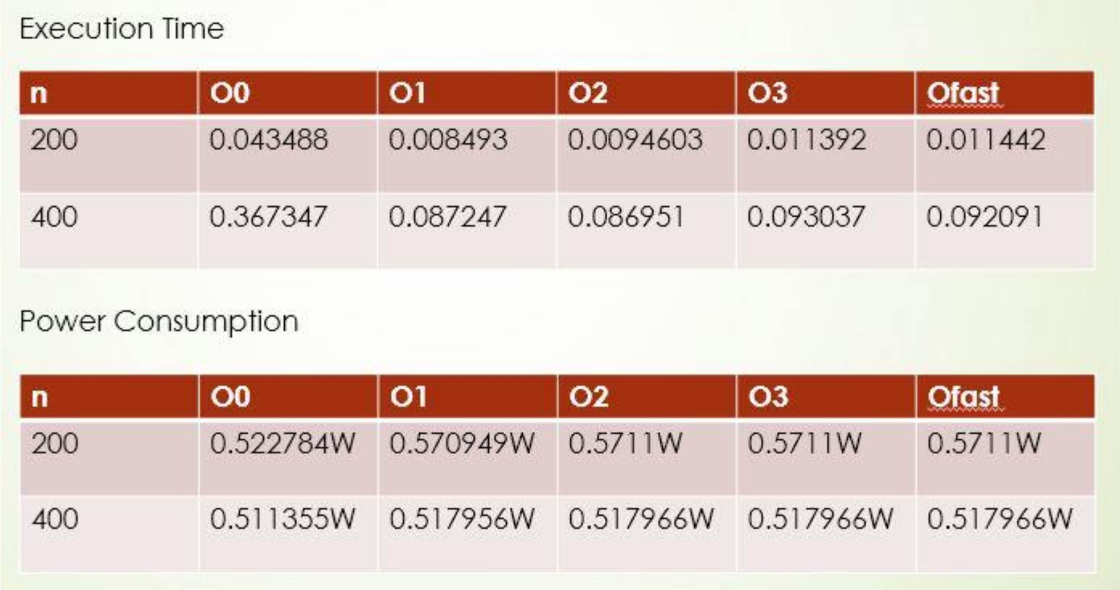
Introduction

A. Background

As suggested from literature survey, compiler optimization techniques have primarily aimed at reducing the execution time (Runtime) of programs and power optimization has been considered as a by-product of it.

B. Motivation

We conducted a few experiments to test the aforestated theory on ARM Architecture using Gem5 Simulator.



Execution Time

n	O0	O1	O2	O3	Ofast
200	0.043488	0.008493	0.0094603	0.011392	0.011442
400	0.367347	0.087247	0.086951	0.093037	0.092091

Power Consumption

n	O0	O1	O2	O3	Ofast
200	0.522784W	0.570949W	0.5711W	0.5711W	0.5711W
400	0.511355W	0.517956W	0.517966W	0.517966W	0.517966W

From above experiment we arrived at the conclusion that power optimization being a by-product of compiler optimization did not hold true.

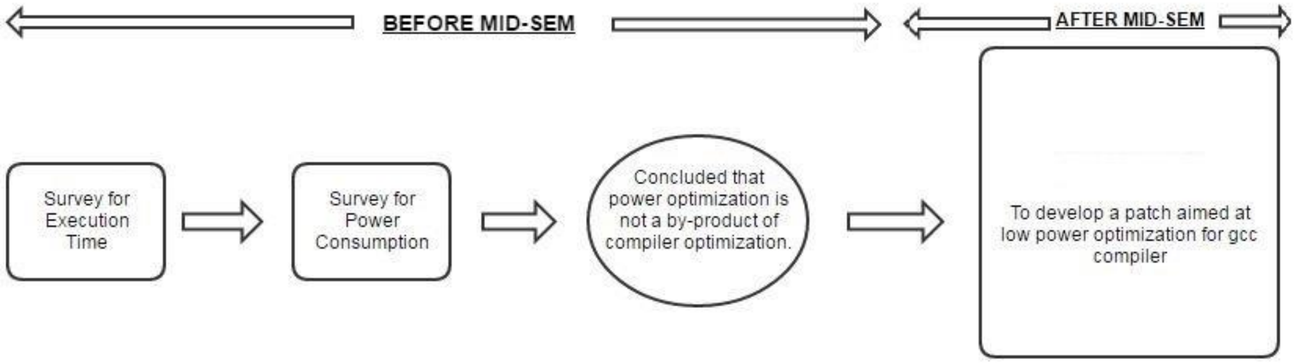
Problem Statement and Objective

1. Exploring various compiler optimization techniques (GCC) and study their effect on power optimization on ARM Architecture.
2. Develop low power compiler techniques as patches for gnu tool chain.

Literature Survey

S No.	**Author**	**Paper Title**	**Year**	**Crux**	**Venue**
1.	David Branco & P.R Henriques	*Impact of GCC Optimization levels in energy consumptions during C/C++ program execution.* [1]	2015	*Presenting experimental setup and method followed to measure and compare resources consumed by a program during execution.*	*2015 IEEE 13th International Scientific Conference on Informatics.*
2.	Vivek Tiwari, Sharad Malik and A. Wolfe	*Compilation techniques for low energy : an overview*[2]	1994	*Used techniques such as Re-ordering instructions to reduce switching and using patterns for Code generations to reduce Power.* *Conclusion: Conducted an experiment which reduced power upto 40%.*	In *Low Power Electronics, 1994, Digest of Technical Papers, IEEE Symposium.*
3.	M Kandemir, N Vijaykrishnan and M.J. Irwin	*Power aware computing*[3]	2002	*Focuses on two power aware low-level techniques: 1) Instruction Scheduling for reducing switching acitivity, and 2) Post-compilation relabeling of Register for reducing Power.*	In *Chapter Compiler Optimizations for Low Power Systems*
4.	M Valluri and Lizy K. John	*Is Compiling for Performance -- Compiling for Power?* [4]	2001	*They present a quantitative study where they examine the effect of the standard optimizations levels −01 to −04 on power and energy of the processor. They also evaluate the effect of four individual optimizations on power/energy and classify them as "low energy" or "low power" optimizations.*	Springer, USA, Boston, MA
5.	U. Kremer	*Low Power/Energy Compiler Optimizations*[5]	2005	*Comparison of Power & Energy and Performance Analysis and concluded that both of them are different strategies and one can not be a by-product of the other.*	*Low-Power Electronics Design, CRC Press, 2005*

Proposed Approach


BEFORE MID-SEM
AFTER MID-SEM
Survey for Execution Time
Survey for Power Consumption
Concluded that power optimization is not a by-product of compiler optimization.
To develop a patch aimed at low power optimization for gcc compiler

BLOCK DIAGRAM (APPROACH)

Software & Hardware Requirements

Software Requirements:

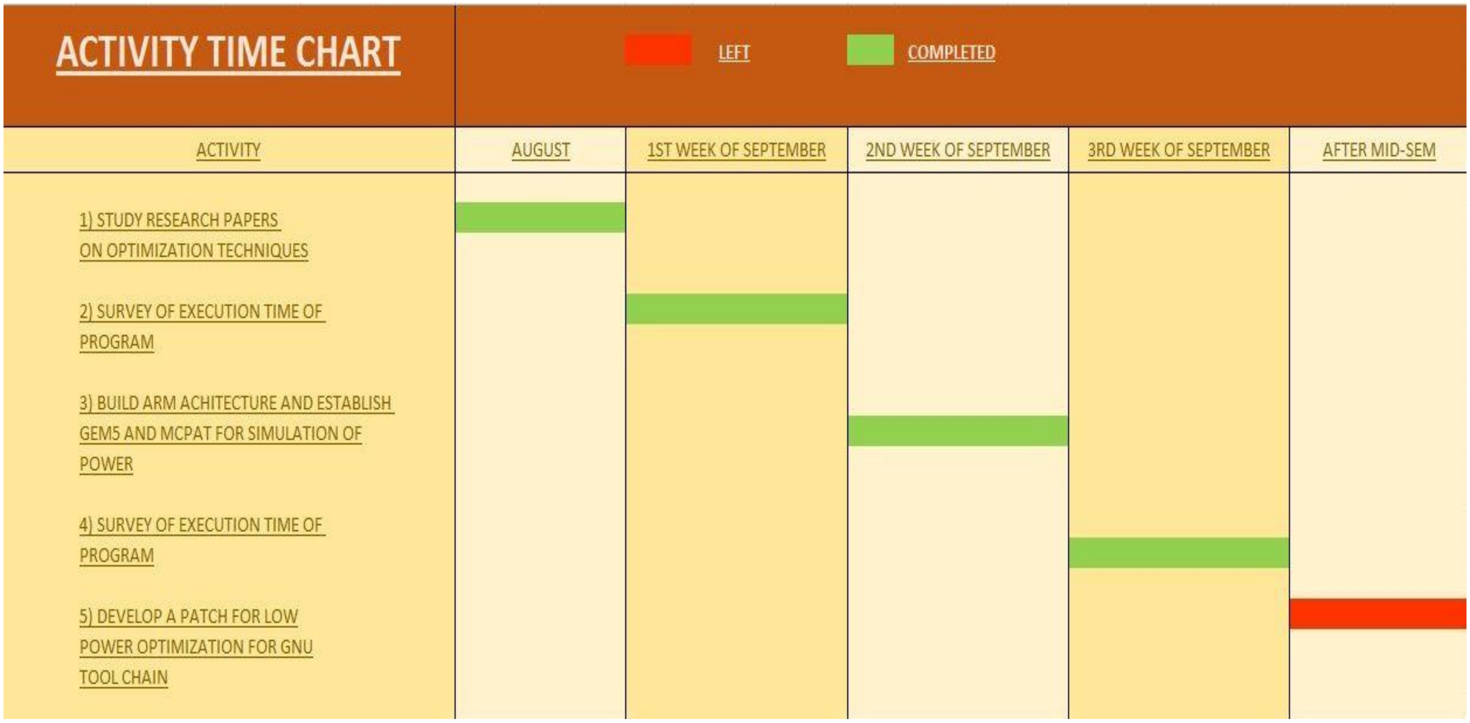
1. Gem5
2. ARM architecture Cross Compiler
3. Gem5ToMcpatConverter[6]
4. McPat[7]
5. GNUPlot
6. GCC (C language for programs)
7. Ubuntu 14.04/ Linux

Hardware Requirements:

System with minimum Requirements:

4GB RAM , Intel Core i3 Processor, 10 GB HD

Activity Time Chart



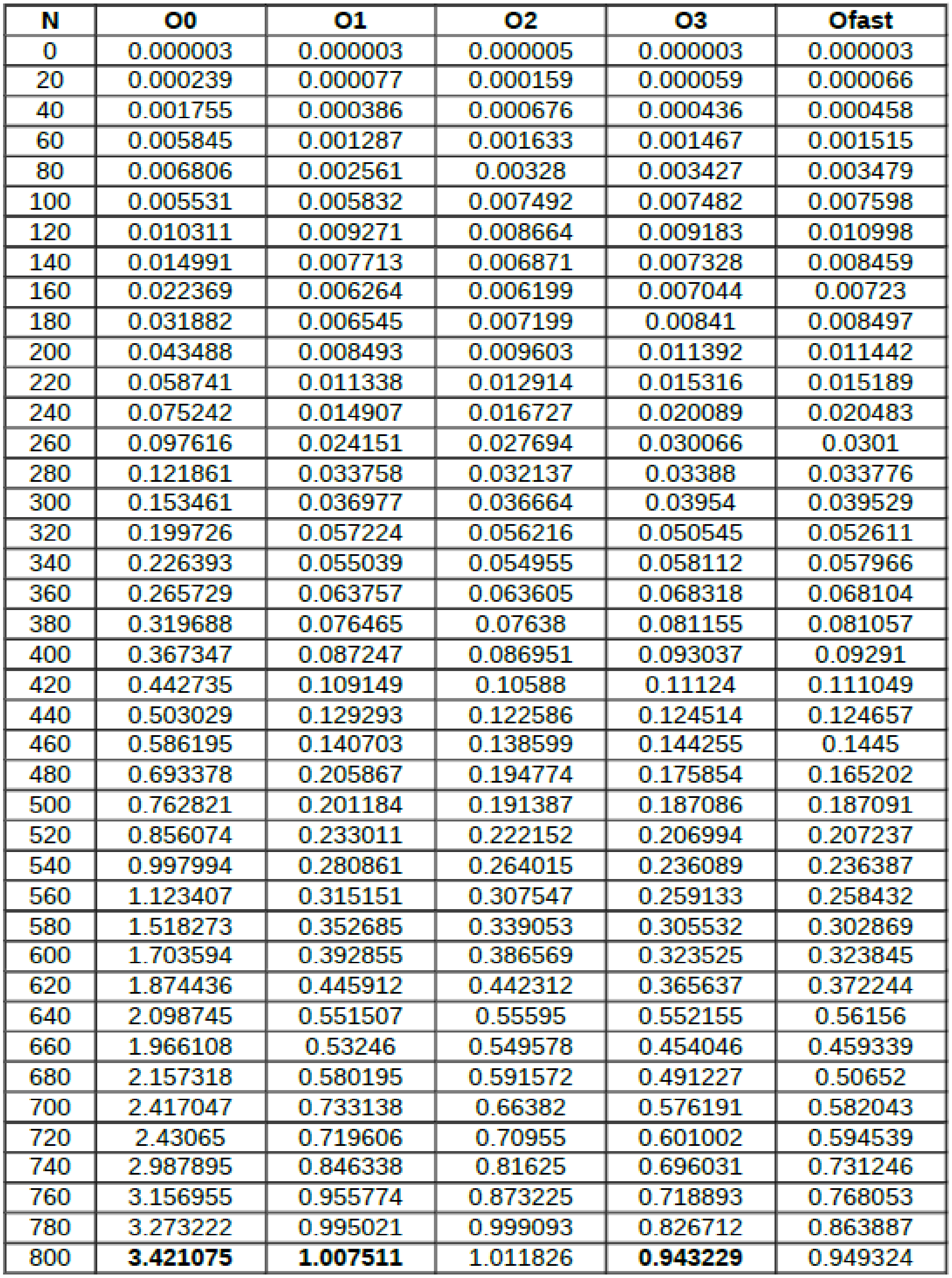
ACTIVITY TIME CHART

🟥 LEFT 🟩 COMPLETED

ACTIVITY	AUGUST	1ST WEEK OF SEPTEMBER	2ND WEEK OF SEPTEMBER	3RD WEEK OF SEPTEMBER	AFTER MID-SEM
1) STUDY RESEARCH PAPERS ON OPTIMIZATION TECHNIQUES	🟩				
2) SURVEY OF EXECUTION TIME OF PROGRAM		🟩			
3) BUILD ARM ACHITECTURE AND ESTABLISH GEM5 AND MCPAT FOR SIMULATION OF POWER			🟩		
4) SURVEY OF EXECUTION TIME OF PROGRAM				🟩	
5) DEVELOP A PATCH FOR LOW POWER OPTIMIZATION FOR GNU TOOL CHAIN					🟥

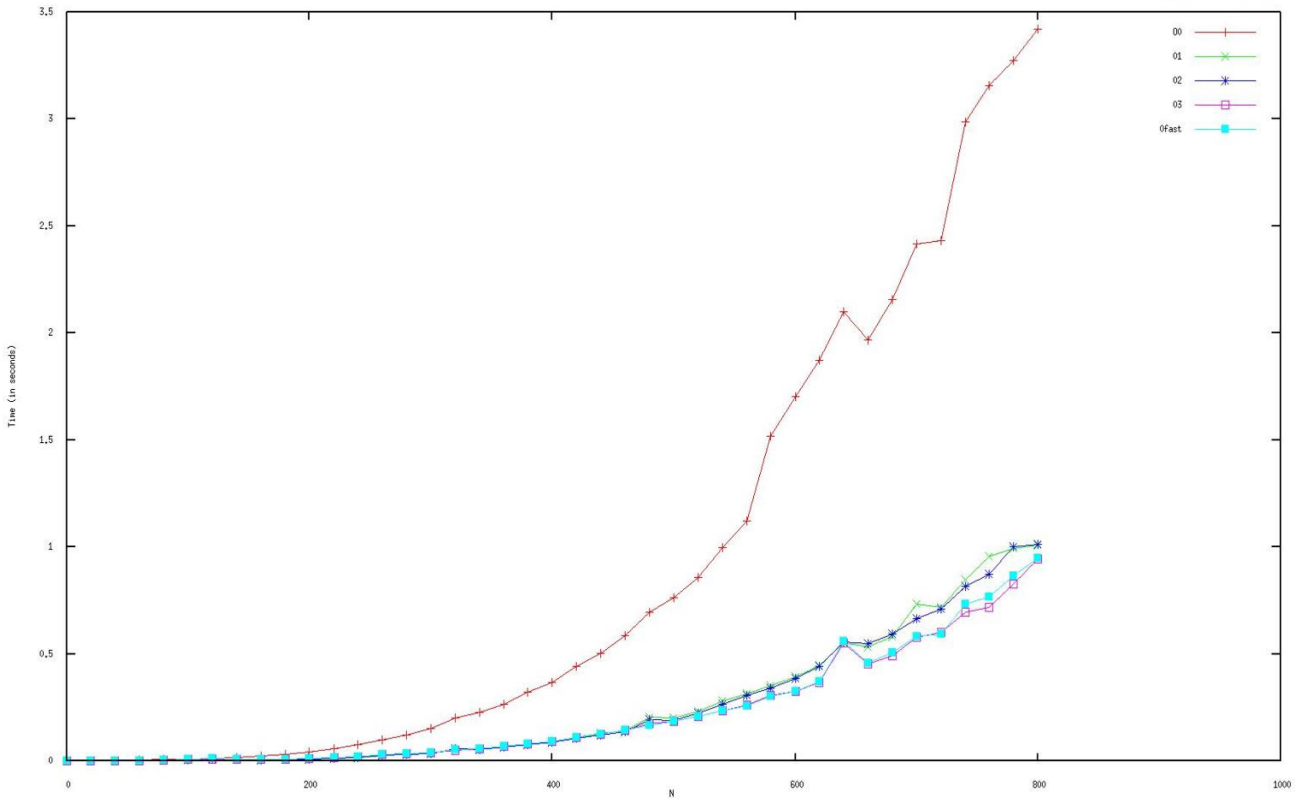
Work completed till Mid-Semester

Matrix Multiplication
Comparison of Execution Time

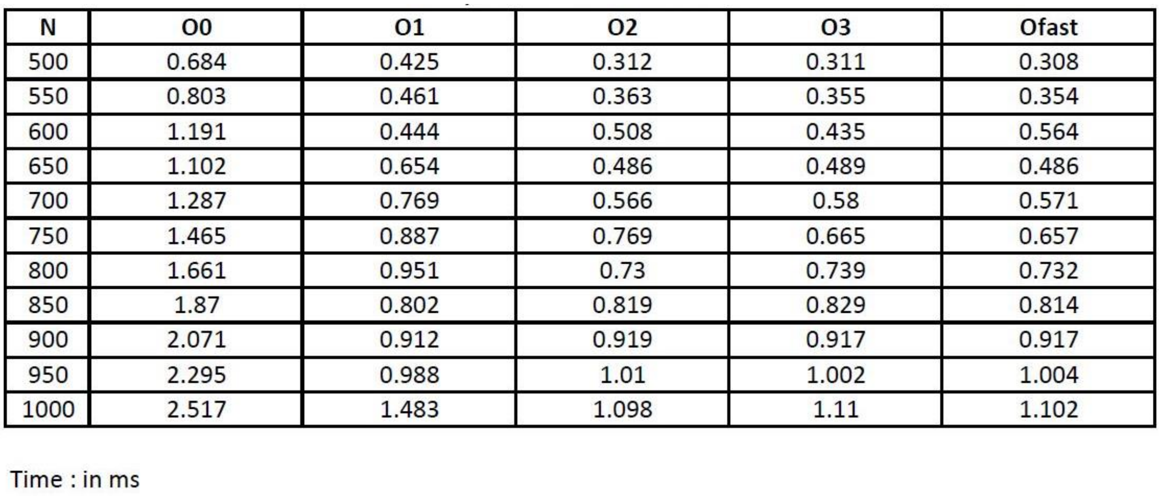


N	O0	O1	O2	O3	Ofast
0	0.000003	0.000003	0.000005	0.000003	0.000003
20	0.000239	0.000077	0.000159	0.000059	0.000066
40	0.001755	0.000386	0.000676	0.000436	0.000458
60	0.005845	0.001287	0.001633	0.001467	0.001515
80	0.006806	0.002561	0.00328	0.003427	0.003479
100	0.005531	0.005832	0.007492	0.007482	0.007598
120	0.010311	0.009271	0.008664	0.009183	0.010998
140	0.014991	0.007713	0.006871	0.007328	0.008459
160	0.022369	0.006264	0.006199	0.007044	0.00723
180	0.031882	0.006545	0.007199	0.00841	0.008497
200	0.043488	0.008493	0.009603	0.011392	0.011442
220	0.058741	0.011338	0.012914	0.015316	0.015189
240	0.075242	0.014907	0.016727	0.020089	0.020483
260	0.097616	0.024151	0.027694	0.030066	0.0301
280	0.121861	0.033758	0.032137	0.03388	0.033776
300	0.153461	0.036977	0.036664	0.03954	0.039529
320	0.199726	0.057224	0.056216	0.050545	0.052611
340	0.226393	0.055039	0.054955	0.058112	0.057966
360	0.265729	0.063757	0.063605	0.068318	0.068104
380	0.319688	0.076465	0.07638	0.081155	0.081057
400	0.367347	0.087247	0.086951	0.093037	0.09291
420	0.442735	0.109149	0.10588	0.11124	0.111049
440	0.503029	0.129293	0.122586	0.124514	0.124657
460	0.586195	0.140703	0.138599	0.144255	0.1445
480	0.693378	0.205867	0.194774	0.175854	0.165202
500	0.762821	0.201184	0.191387	0.187086	0.187091
520	0.856074	0.233011	0.222152	0.206994	0.207237
540	0.997994	0.280861	0.264015	0.236089	0.236387
560	1.123407	0.315151	0.307547	0.259133	0.258432
580	1.518273	0.352685	0.339053	0.305532	0.302869
600	1.703594	0.392855	0.386569	0.323525	0.323845
620	1.874436	0.445912	0.442312	0.365637	0.372244
640	2.098745	0.551507	0.55595	0.552155	0.56156
660	1.966108	0.53246	0.549578	0.454046	0.459339
680	2.157318	0.580195	0.591572	0.491227	0.50652
700	2.417047	0.733138	0.66382	0.576191	0.582043
720	2.43065	0.719606	0.70955	0.601002	0.594539
740	2.987895	0.846338	0.81625	0.696031	0.731246
760	3.156955	0.955774	0.873225	0.718893	0.768053
780	3.273222	0.995021	0.999093	0.826712	0.863887
800	**3.421075**	**1.007511**	1.011826	**0.943229**	0.949324

<u>Matrix Multiplication</u>
<u>Graph of Execution Time</u>



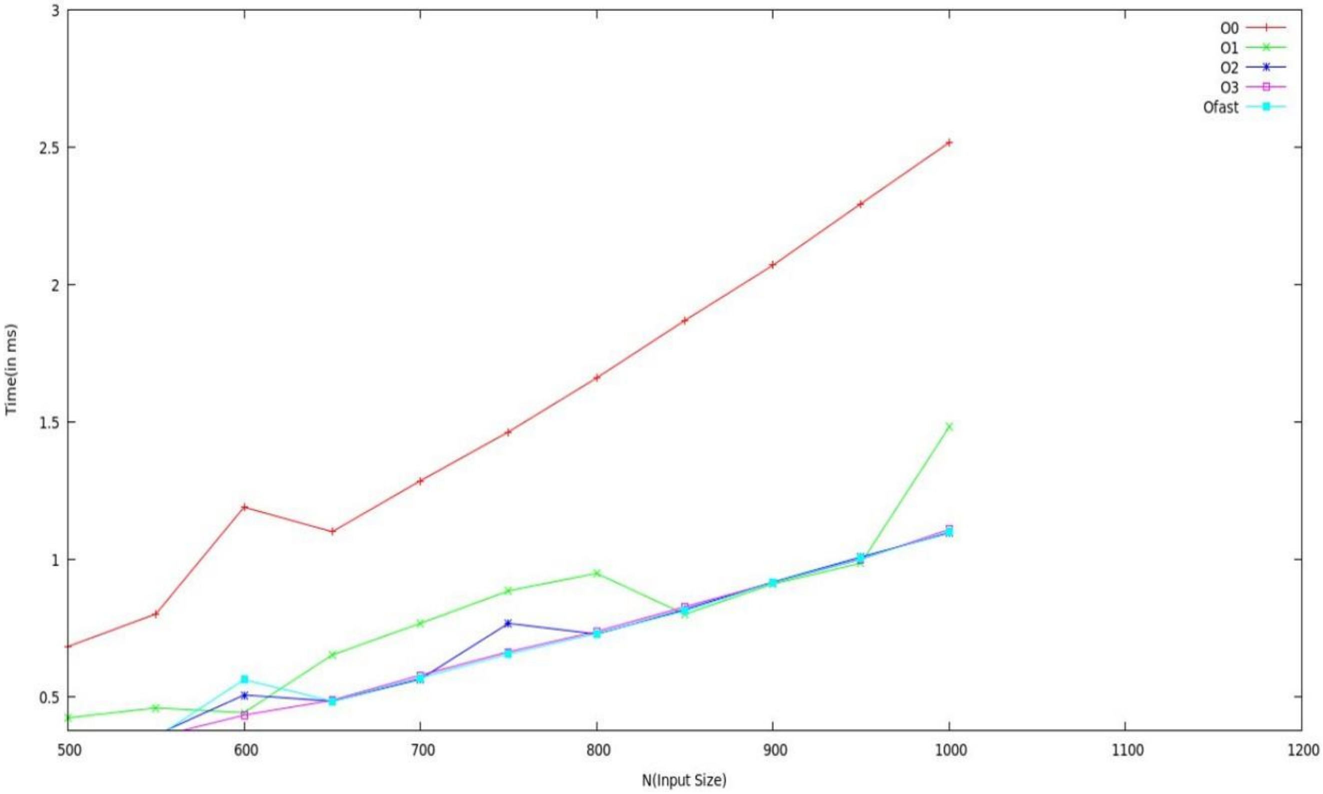
Bubble Sort
Comparison of Execution Time



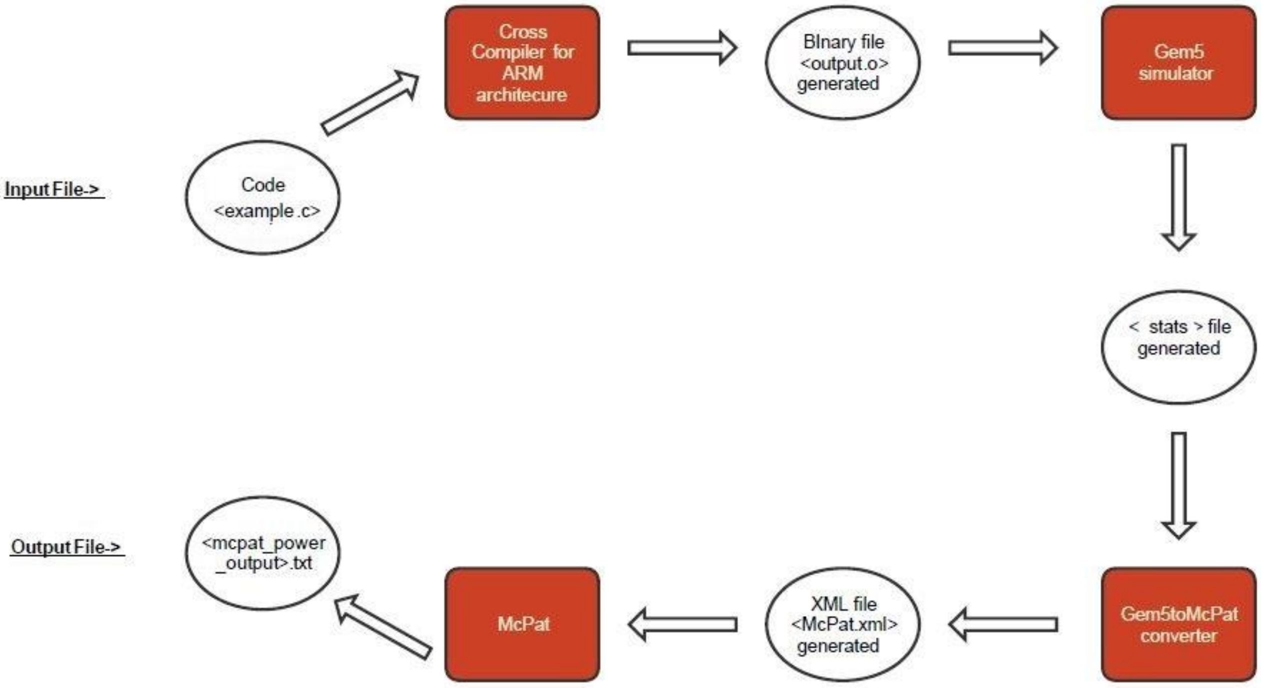
N	O0	O1	O2	O3	Ofast
500	0.684	0.425	0.312	0.311	0.308
550	0.803	0.461	0.363	0.355	0.354
600	1.191	0.444	0.508	0.435	0.564
650	1.102	0.654	0.486	0.489	0.486
700	1.287	0.769	0.566	0.58	0.571
750	1.465	0.887	0.769	0.665	0.657
800	1.661	0.951	0.73	0.739	0.732
850	1.87	0.802	0.819	0.829	0.814
900	2.071	0.912	0.919	0.917	0.917
950	2.295	0.988	1.01	1.002	1.004
1000	2.517	1.483	1.098	1.11	1.102

Time : in ms

Bubble Sort
Graph of Execution Time

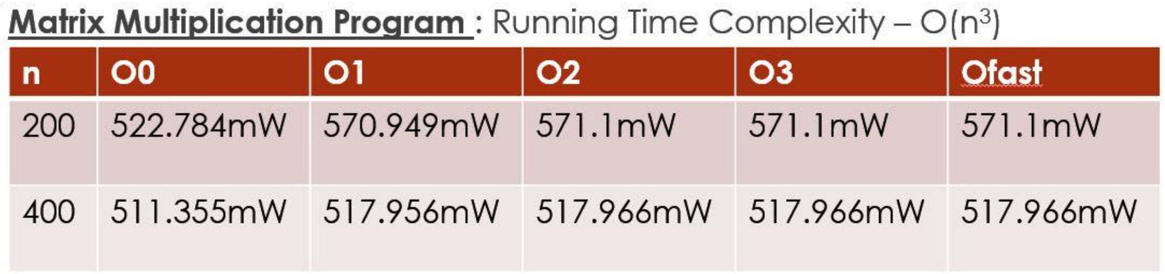


Power Consumption Survey



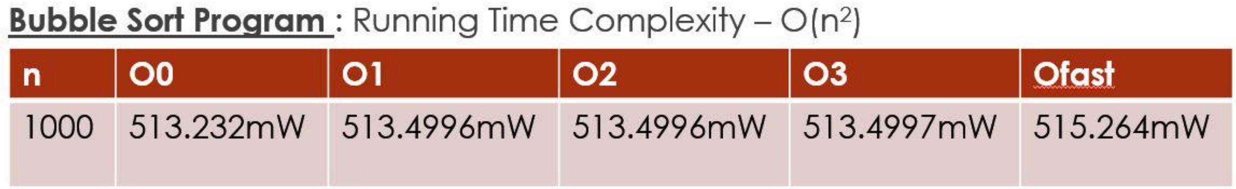
BLOCK DIAGRAM(POWER ESTIMATION)

Comparison of Power Consumption



Matrix Multiplication Program : Running Time Complexity – O(n^3)

n	O0	O1	O2	O3	Ofast
200	522.784mW	570.949mW	571.1mW	571.1mW	571.1mW
400	511.355mW	517.956mW	517.966mW	517.966mW	517.966mW



Bubble Sort Program : Running Time Complexity – O(n^2)

n	O0	O1	O2	O3	Ofast
1000	513.232mW	513.4996mW	513.4996mW	513.4997mW	515.264mW

Conclusion

After Execution Time Analysis on both Matrix Multiplication and Bubble Sort, it was observed that

RUNNING TIME

For Large Values of N:

- reduces considerably (for larger inputs) from O0 to O1,
- then reduces relatively less from O1 to O2,
- and similarly for other levels, with Ofast being the best.

For Small Values of N

- does not reduce considerably from O0 to O1,
- then reduces relatively less from O1 to O2,
- and similarly for other levels, with Ofast being the best.

POWER

- increases from O0 to O1,
- then increases relatively less from O1 to O2,
- then very slow increase (or similar) from O2 to O3
- then either remains stable or increases from O3 to Ofast

From the statistics obtained from our experiment, we come at the conclusion that

1) The existing compilation techniques focus on minimizing Execution Time with various optimization Levels.
2) These optimization techniques however, didn’t take into consideration the Power consumption of the Program as it is clearly seen that Power Consumption increases from O0 to Ofast.

Hence, there is a need to come up with optimization techniques specifically targeting for Power Consumption.

Future Implementation

- To Study Various Power Optimization Techniques.
- To Develop a patch for the GNU tool-chain that will aim at reducing Power at various Optimization levels.
- To ensure/prove that the functionality of the program is conserved after applying optimization techniques.

References

[1] D Branco and P.R Henriques: Impact of GCC Optimizations Levels in Energy Consumption during C or C++ program execution.
2015 IEEE 13th International Scientific Conference on Informatics.

[2] V.Tiwari, S Malik and A. Wolfe : Compilation Techniques for Low Energy, An Overview.
-Low Power Electronics, 1994. Digest of Technical Papers, IEEE Symposium.
Pages 38–39, Oct 1994.

[3] M. Kandemir, N Vijaykrishnan and M. Jane Irwin: Power Aware Computing, Chapter: Compiler Optimizations for Low Power Systems.
-Pages : 191 to 210. Kluwer Academic Publishers, Norwell, MA, USA – 2002

[4] M. Valluri and Lizy K. John: Is Compiling for Performance – Compiling for Power?
Springer, USA, Boston, MA – 2001

[5] U Kremer: Low Power/Energy Compiler Optimizations
In Low-Power Electronics Design, CRC Press, 2005

[6] https://bitbucket.org/dskhudia/gem5tomcpat

[7] http://www.hpl.hp.com/research/mcpat/

Suggestions by Board-4 Members