EFFICIENT ARCHITECTURE BY EFFECTIVE FLOORPLANNING OF CORES IN PROCESSORS

Thesis submitted in partial fulfillment of the requirements for the degree of

Bachelor in Technology

in
Computer Science and Technology

by

APARUP BEHERA

(Roll: 110CS0117)

Under the guidance of

Prof. Ashok Kumar Turuk

NIT Rourkela

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769008, Odisha, India

Department of Computer Science and Engineering, National Institute of Technology Rourkela, Rourkela-769008, Odisha, India.

5th May, 2014

CERTIFICATE

This is to certify that the work entitled *Efficient Architecture by Effective Floorplanning of cores in processors* by *Aparup Behera* is a record of an original research work carried out under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Prof. Ashok Kumar Turuk
Dept. of Computer Science and Engineering
NIT Rourkela

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my guide Prof. Ashok Kumar Turuk for his exemplary guidance and professional help. His profound insights helped me to reach my goal in this project. His support led to the encouragement to pursue this research.

I am highly indebted to Prof. Santanu Kumar Rath, Head of Department of Computer Science Department, for giving the opportunity and help required every time. I would also like to thank Prof. S.K. Jena, Prof. B. Majhi and all the faculty staff for their support.

I am thankful to all my friends whose constant motivation compelled me to finish this work.

AUTHOR'S DECLARATION

I hereby declare that all the work contained in this report is my own particular work unless otherwise recognized. Also, all of my work has not been previously submitted for any scholarly degree. All sources of quoted information have been recognized by method of appropriate references

Aparup Behera
110CS0117
Dept. of Computer Science and Engineering

ABSTRACT

Hotspots in cores are becoming more prominent all thanks to the increasing clock speed in the market. Core size cannot be increased further. Better technology means reducing the heat or distributing the heat evenly to keep the peak temperature low. One has to find a trade-off between area and cost to make the cores and overall architecture cost effective and heat controlled. Generating floorplans and evaluating them to make the best core is underway. For the current market a trade-off has to be made between area and peak temperature. The floorplan accepted can be built into the desired architecture to gain efficient results. Processors like Alpha ev6 and Pentium have become obsolete paving way for more advanced processors like i3, i5, i7 and the recent Haswell processors have greater efficiency and speed, so temperature management should be next point to ponder on after efficiency and speed. Better the temperature management, more is the cost effectiveness of the single core or the multiple cores in the processor.

CONTENTS

LIST OF FIGURES

Chapter 1
INTRODUCTION

Processors are the computing resources of the Central Processing Unit. The Integrated Circuits used are very sophisticated and designed accordingly to cater the needs of different users. They are the smallest embedded systems inside the computational devices that do all the computation. Different architectures are now available in the market. These architectures are very different from one another depending on the number of cores or on L-caches. They give different running times for different programs and tend to heat up differently for different purposes. So designing of processors is also a hefty process. Also the cores, which do work like storing in registers (e.g. IntReg), have higher temperatures. Such parts are called hotspots [1]. Since components of the core heat up in a different proportion, the peak temperature becomes the point to consider. The solution is to adjust. The temperature of a particular block depends on its power wattage. The heat from adjacent blocks also causes a particular block to heat up more than it is expected to do. This leads to rise in temperature evenly and reduction in temperature in hotspots [1]. If processor heats up quickly the longevity of cores decreases and so does the life of the processors. This leads to repetitive replacement of processors.

1.1   Current Scenario

Nowadays new processors are being made with more speed i.e. more cycles per second. So, the processors and their cores tend to heat up more quickly. Increasing the area is not a viable alternative as more and

more components are being inserted into the motherboard creating less space. So final solution is to reorder the components of the cores and run it to find which gives minimum peak temperature for a given arrangement. After finding the optimum structure one can insert the core in one of many architectures to find out the best architecture which gives minimum running time. This can be done by multi2sim simulator [9].

1.2   Choosing the best architecture

Best architecture can be chosen only by comparison of running time, cycles taken while running different serial and parallel programs. This is done by changing the number of cores to compare the results. This can be done by the multi2sim simulator [9] which gives details of running different programs. Choosing the best architecture can be difficult as structures with different cores can be at par with structures with different connected components. Increasing the core number should be restricted at certain point because some cores may not be used and adding extra cores is redundant. There may be switches that allow different memory hierarchies to function and give results for threaded programs.

1.3 Choosing the best core

'Floorplanning' term refers to dividing the given space into divisions (like planning rooms for construction in a house that is to be built). Floorplanning [4] in cores means arranging the components [1] like IntReg, FPAdd, FPReg etc. to find an arrangement that gives either gives

better efficiency or better temperature management or both . Power density of different components are different. The temperature of each component does not only depend on the power densities of the concerned blocks but also the blocks adjacent. In processors like Alpha and Pentium Pro Processors [1] the heat may rise up to high values like 70°C which is way too high and may lead to thermal hazard. Planning efficiently can reduce these high temperatures. The areas where temperatures are maximum are called "hotspots" [1]. These hotspots have to be evenly divided and surrounded by low temperature zones for minimum peak temperatures.
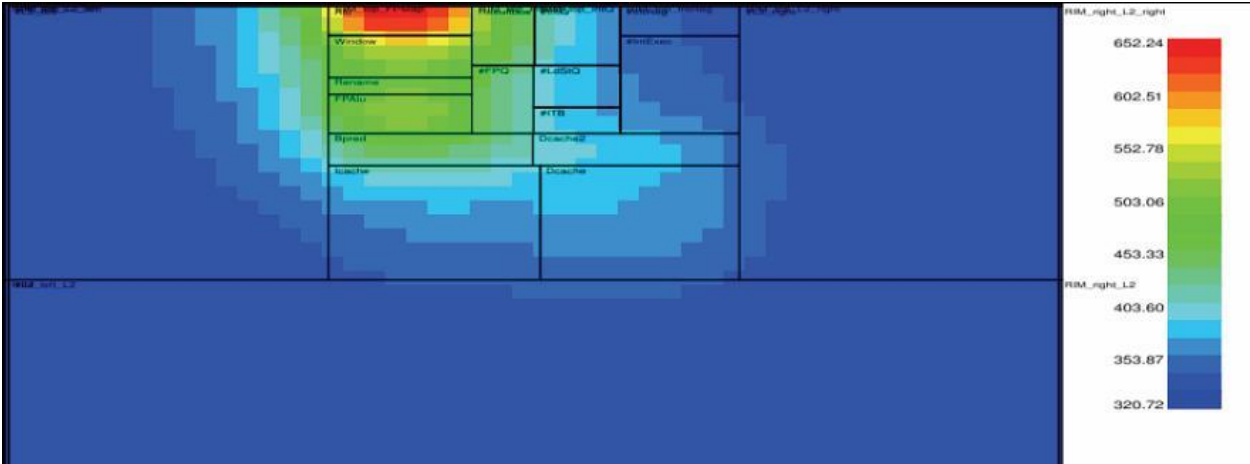

Fig 1.1: Alpha ev6 Processor with temperature variation

1.4 Problem Statement

The objective of this project is to use the existing methods to create and effective core that can be put in the best architecture to provide the best system available from the given limited choices by comparison. This comparison method can be used in large scale in industries whenever required.

Chapter 2
Literature Review

2.1 Memory Architectures

Memory Architecture defines a structure which combines different cores, different cache memories and different bus structures. The architecture may be serial or parallel. The architecture may be 8-bit, 16-bit, 32-bit or 64-bit. The architecture may be SMP (Symmetric Multi-Processing) or AMP (Asymmetric Multi Processing). Either each may handle one thread of each process or each core may handle all threads of a single project.

2.2 Clock Rate

The Clock Rate [6] refers to the frequency in which the processor is running. The clock rate is normally determined by frequency of a crystal oscillator. The faster the clock the more operations it does in a unit time. Clock Rate is measured in Mega Hertz (MHz) or Giga Hertz (GHz). It is a common notion that processors with better clock speed perform better. An IBM PC with an Intel 80486 CPU running at 50 MHz will be about twice as fast as one with the same CPU and memory running at 25 MHz but even if two processors have the same clock rate they may take different number of cycles to perform the same operation. The frequency with which the instructions are being executed per unit time is increasing every year.
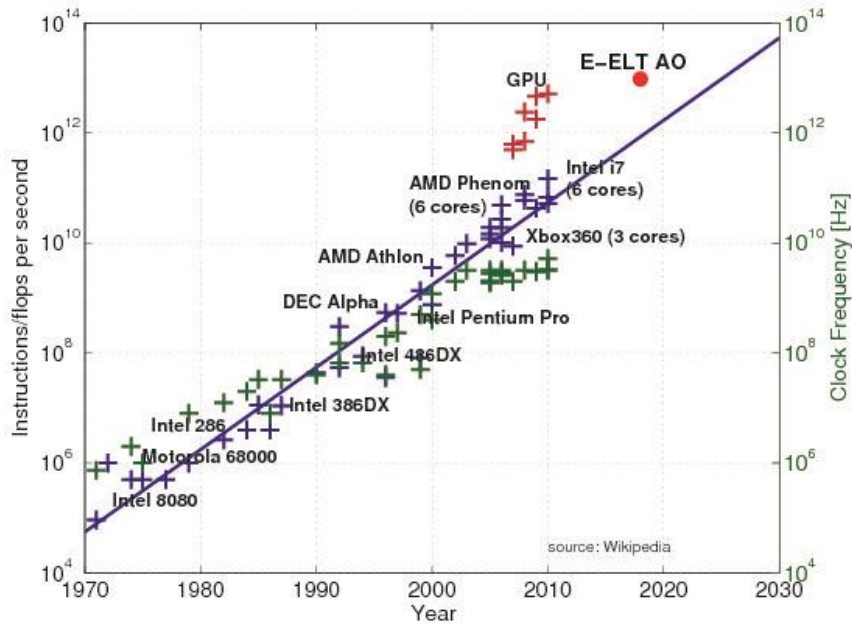
Fig 2.1 Clock frequency estimation year-wise

2.3 Heat due to Clock Rate

Processors in mobiles and small gadgets require processors of low power. However, the processors in personal computers require high power components for more complex operations. Higher clock rate [6] leads to greater temperature. This temperature differs in different systems. The total power P is divided into:

$P = P_d + P_s + P_{leak}$, where P_d refers to dynamic power, P_s refers to short-circuit power. More clock rate means more instructions executed per unit time. This leads to heating up of different components like registers in cores tend to heat up quickly than other components like arithmetic resolving components. Cache memory which do not heat up quickly also help to dissipate heat.

2.4 Hotspots

Hotspots [1] are those parts of the processor that tend to heat up more quickly than their other counterparts. Hotspots have to controlled in such a way that the wire length overhead is minimized and the temperature is effectively diffused keeping the performance same. E.g In an Alpha Processor parts like IntReg, FPReg, Bpred etc. heat up more, so are known as hotpots of Alpha Processor.

2.5 Heat Dissipation

Heat produced in different components is proportional to the power densities of the components. Also the temperature depends on the adjacent blocks. Suppose in an original Alpha ev6 core, power densities of *IntQ* and *FPReg* are 0.137 Watt/mm^2 and 0.623 Watt/mm^2 respectively. But, temperature of IntQ in original floorplan is higher than FPReg because IntQ is near blocks like IntReg, IntExec etc. which are of higher power densities [1,3]. Better floorplanning can be done using trying out random floorplans in QUILT simulator [10] by placing the one with greater power density in between components with lower power densities. Also floorplans can made by generating different plans and selecting the better one among them.

2.6. Simulated Annealing

Certain optimization problems that use combinatorial methods face difficulty because the number of objects to find is too high and will take time. Basically NP Problems like Travelling Salesman Problem can use

methods like Simulated Annealing [8] to find a better if not an optimum one. For these problems, simulated annealing is very effective. Simulated Annealing has the condition for accepting a calculated value: if it satisfies $e^{-\Delta D/T} > R(0,1)$, where ΔD is the change of distance implied by the trade (negative for a better trade; positive for a worse trade. T is a "synthetic temperature," and $R(0,1)$ is a random number in the interval $(0,1)$. D is called a "cost function," and corresponds to the free energy in the case of annealing a metal (in which case the temperature parameter would actually be the kT, where k is Boltzmann's Constant and T is the physical temperature, in the Kelvin absolute temperature scale). If T is large, many values are accepted, and a large part of solution space is accessed. Objects to be traded are generally chosen randomly, although there are better methods like genetic algorithms. The temperature is reduced every loop by a factor until it reaches a certain point to get a good result.

2.6.1 Advantages of Simulated Annealing

(i) Plan may not be an optimal one but a balanced one with appropriate area and peak temperature.

(ii) It does not go through all the possible solutions, so it takes less computational time.

(iii) It may not ignore some local maximum.

2.6.2 Disadvantages of Simulated Annealing

(i)   The plan proposed may be having gaps inside which does not allow it to be an optimal one.

2.7 Core Orientation

Core Orientation [7] is beneficial usually in evaluating multicore architecture. Cores can be accumulated together with different orientation. Cache memories are positioned in different parts of the processor. Cores can be rotated and positioned in the center with L2 Caches wrapped around or in the corners. Without compromising the reusability of the core, a temperature efficient floorplan with more than cores can be sketched to give good results. L2 Banks can be distributed throughout the processor to distribute the heat. In Alpha Processor, cores can be arranged in checkerboard format sending out the cores to four corners. Alternatively, different blocks of the cores can be scattered around the processor.

Chapter 3
Proposed Work

Studies conducted in University of Virginia deal in using HotSpot Simulator [2] to evaluate generated floorplans of cores and cache memories. The HotSpot Simulator is made by University of Virginia. To select the appropriate memory architecture, one has to select the one that gives lowest running time for a sample program and the threaded version of the same program. This has to be selected from a random group of architectures. However to find the best core that does not heat up very quickly, one has to manually build the core or generate it using any algorithm. The generated floorplan is tested against HotSpot 5.02 [2], that gives results to decide if the floorplan is to be accepted or not.

i)   Floorplans are proposed (can be done either manually using QUILT simulator [10] or by algorithms involving simulated annealing [8] and genetic algorithms) which give same comparable performance with respect to the older ones.

ii)   HotSpot Simulator [2] is used to compare the floorplans generated. The floorplans may generate temperatures as high as 70^oC. Heat diffusion [1,3] measure is taken to reduce the peak temperature. Heat diffusion is possible if high power and high functioning blocks are surrounded by ones with lower corresponding values.

Chapter 4
Simulation Environment and Settings

4.1 Tools Used:

4.1.1 QUILT 1.0

QUILT stands for Quick Utility for Integrated circuit Layout and Temperature Modelling [10].

4.1.1.1 Features of QUILT 1.0
i) Coverlet grants clients to quickly assemble floorplans of incorporated circuits, giving both a visual help and additionally an information to the Hotspot 5.02 test system [2].
ii) The tool gives various characteristics to assessing circuit execution, for example, interconnect delay, and for creating graphical pictures for distributions.
iii) QUILT is based on Java; so it is platform-independent.

4.1.1.2 Advantages of QUILT 1.0
i) GUI Interface makes it easy to shift blocks of core.
ii) One can adjust the protruding parts.
iii) Insertion of extra L2 Caches is possible.

4.1.1.3 Disadvantages of QUILT 1.0

i) Aspect ratio cannot be changed directly.
ii) Multi-core manipulation is not allowed.

4.1.2 HotSpot 5.02

HotSpot 5.02 [2] is an accurate and fast thermal model suitable for use in architectural studies. It is focused around a comparable circuit of warm resistances and capacitances that relate to microarchitecture pieces and vital parts of the warm bundle. The model has been accepted utilizing limited component recreation. HotSpot Simulator has a simple set of interfaces and hence can be integrated with most power-performance simulators like Wattch [6]. The boss playing point of Hotspot is that it is good with the sorts of force/execution models utilized as a part of the workstation-structural planning group, obliging no nitty gritty outline or combination portrayal. HotSpot Simulator makes it possible to study thermal evolution over long periods of time over all types of applications and functions. HotSpot 5.02 allows one to get both transient temperatures and steady temperatures.

Features of HotSpot 5.02
  i)   It saves time by giving the steady final temperature immediately.
  ii)   One can use grid method of HotSpot to generate temperature of each unit area (like in a grid).
  iii)   One can use it for 3D Cores i.e. core on top of another core.

4.2 Platform Used: Ubuntu 12.04

4.3 Programming Language Used: C

Chapter 5
Implementation

5.1 Model Used

The Model used here is the Alpha ev6 processor. The components of Alpha ev6 processor are shifted and have their aspect ratio changed to generate a new model. The entire project focuses on Alpha ev6 core [1].

5.2 Using QUIT 1.0 [10]

i)   The application can be opened by the command "java –jar quilt.jar " .

ii)   It has three editing modes that can be used namely:

   a) Moving units.

   b) Resizing units keeping area constant.

   c) Resizing units without keeping area constant.

iii)   Existing floorplans are rearranged or new parts are added or parts are deleted using QUILT Graphical User Interface (GUI).

5.3 Floorplans by Simulated Annealing [8]

The following algorithm generates the steps to make a floorplan:

   1) Set T, Limit,factor_to_reduce

   Loop 1:

   2) count=0

   Loop 2:

3) If count=6 exit Loop 1

4) Generate random new Plan Sequence from previous accepted plan by changing once at a place.

5) Calculate cost of new Plan

6) Calculate difference = cost (new Plan) – cost (Previous Plan)

7) If difference<0, accept the plan and go to Step 3

8) Else If difference>0 and $e^{-difference/T}$> Random (0, 1), count++, accept the plan and go to Step 3

9) Else discard the plan, go to Step 3

End Loop 2

10) T=T*(factor_to_reduce)

11) If T>=Limit go to Step 2

12) Else exit Loop 1

End Loop 1

The above algorithm gives the sequence of blocks and the orientation (either "horizontal" or "vertical"). The proposed plan is built using QUILT Simulator [10] to produce the floorplans.

5.4 Using HotSpot 5.02 [2]

i)   To install HotSpot 5.02, one has to implement the "make" command inside the directory.

ii)   To evaluate a floorplan one needs a .flp file and .ptrace file inside the HotSpot folder. The .flp file contains the height, width, x and y coordinates.

iii) To get steady temperatures of all the blocks one has run the command:

```
./hotspot -c hotspot.config -f <plan_name>.flp -p
<benchmark_name>.ptrace -o <benchmark_name>.ttrace -
steady_file <benchmark_name>.steady
```

5.5 Benchmarks

The Benchmarks used belong to the SPEC2000 Suite [5]. The Standard Performance Evaluation Corporation (SPEC) is an American non-profit organization that aims to produce, establish, maintain and endorse a standardized set of performance benchmarks for computers. SPEC2000 helps to compare the performance based on different aspects. SPEC2000 is a retired SPEC CPU Benchmark Suite [5]. The Benchmarks of SPEC2000 [5] are : amp, applu, apsi, art, bzip2, crafty, eon, equake, facerec, fma3d, galgel, gap, gcc, gzip, lucas, mcf, mesa, mgrid, parser, perlbmk, sixtrack, swim, twolf, vortex, vpr and wupwise. These benchmarks are compiled using the "intel-linux" configuration and the SPEC "base" tuning option [5]. Benchmarks give a wide variety of workload and functions to show the results for.

5.6 Assumptions
    i) The core may or may not have some empty space.
    ii) The area of each block does not change
    iii) The orientation of L2 caches should be around the core so as to cover it.
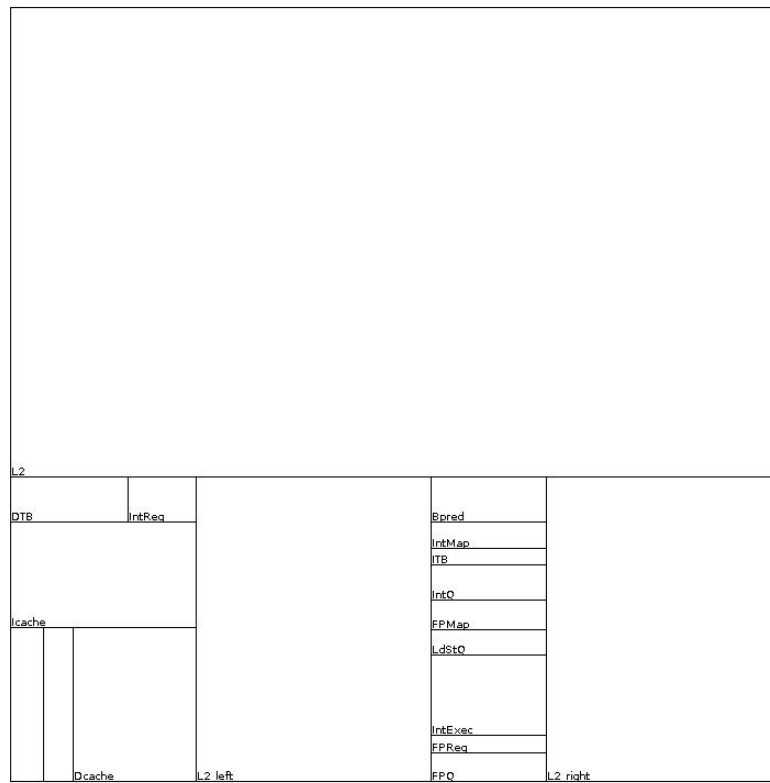   iv) The power of each block remains the same.

5.7 Screenshots

Fig 5.1: Screenshot of a Plan Generated by QUILT Simulator

Chapter 6
Results and Analysis

This section shows simulation results from HotSpot 5.02 [2].

6.1 Six Plans designed manually and run with HotSpot 5.02:

Floorplan 1: Original Alpha ev6 floorplan

Floorplan 2: Random floorplan with area same as Floorplan 1

Floorplan 3: With Rotated Core

Floorplan 4: With Low Wire Overhead

Floorplan 5: With Low Wire Overhead

Floorplan 6: Insertion of L2 Cache into middle of core

6.2 Effect of Power and Orientation
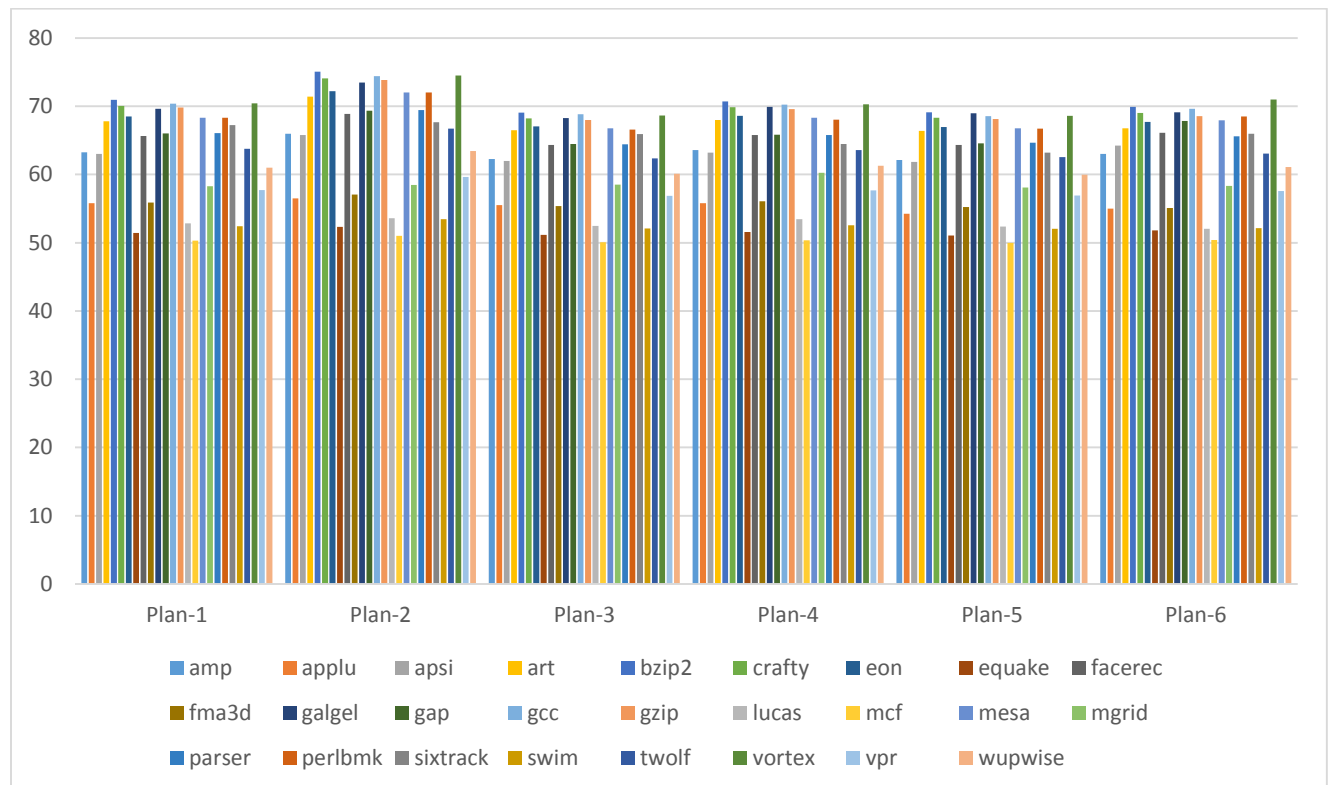
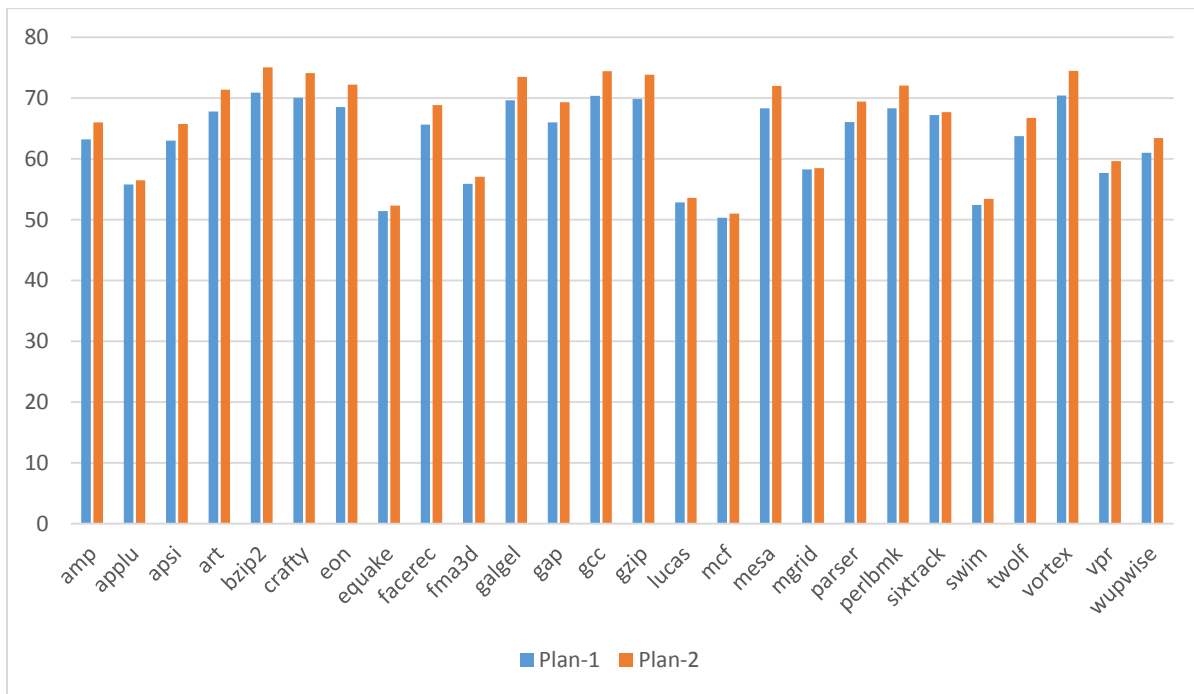Fig. 6.1: Graph of Temperature (in ^{o}C) of all benchmarks for each plan

Fig 6.2: Difference between Floorplan-1 and Floorplan-2 w.r.t all benchmarks
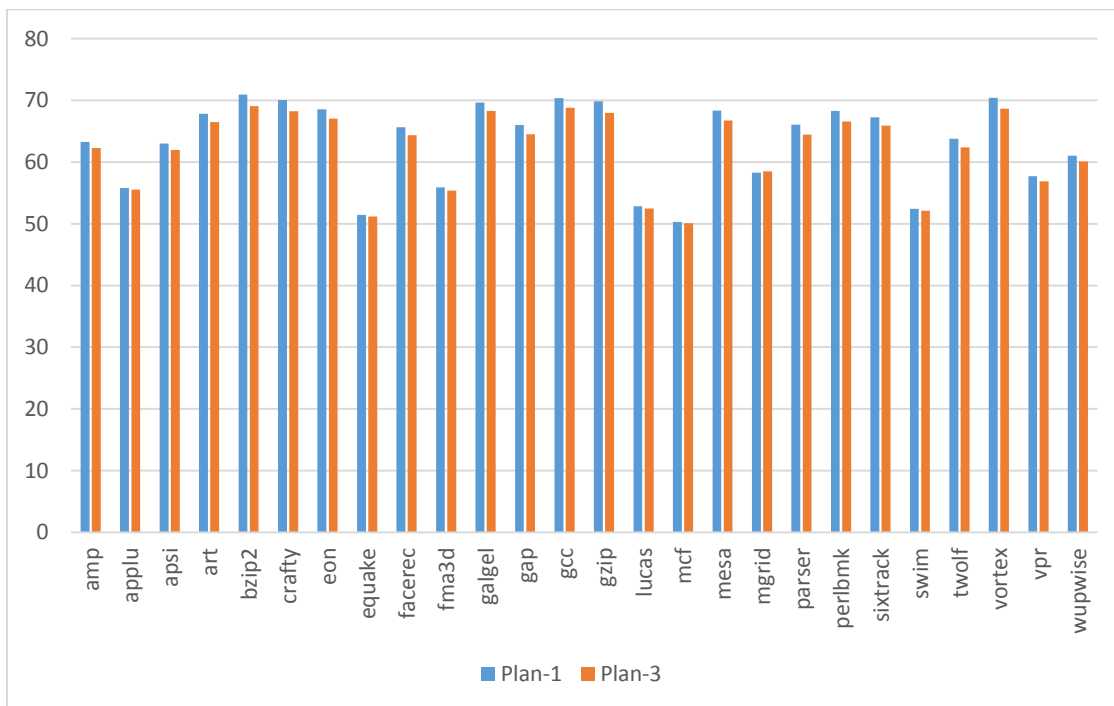

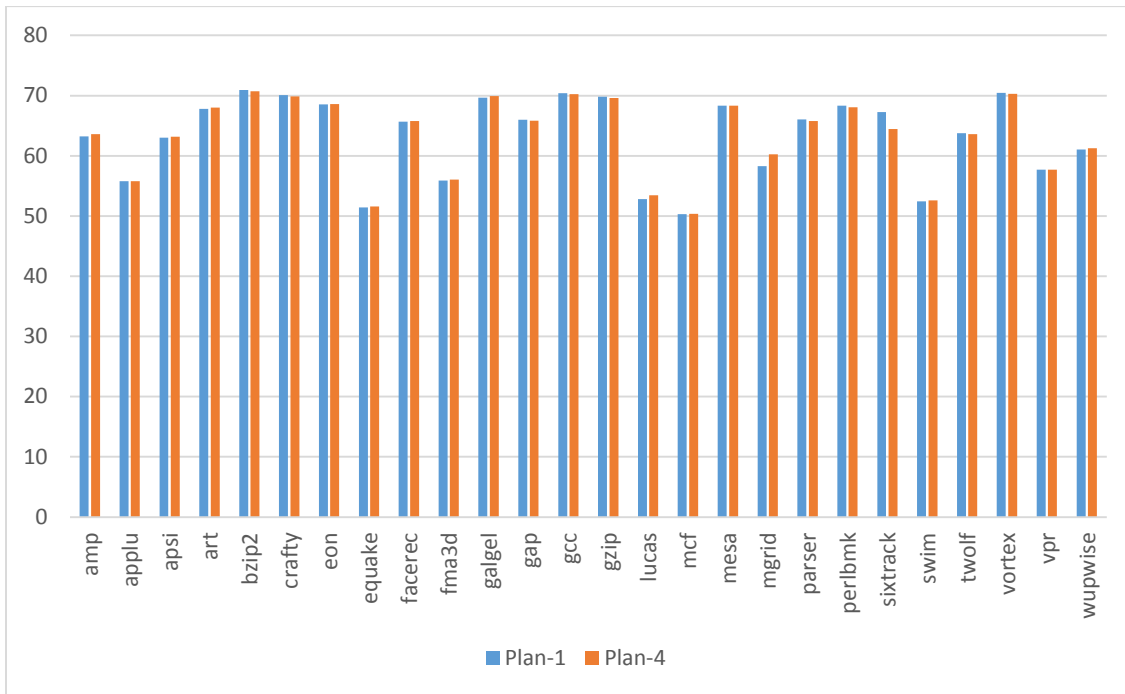Fig 6.3: Difference between Floorplan-1 and Floorplan-3 w.r.t all benchmarks

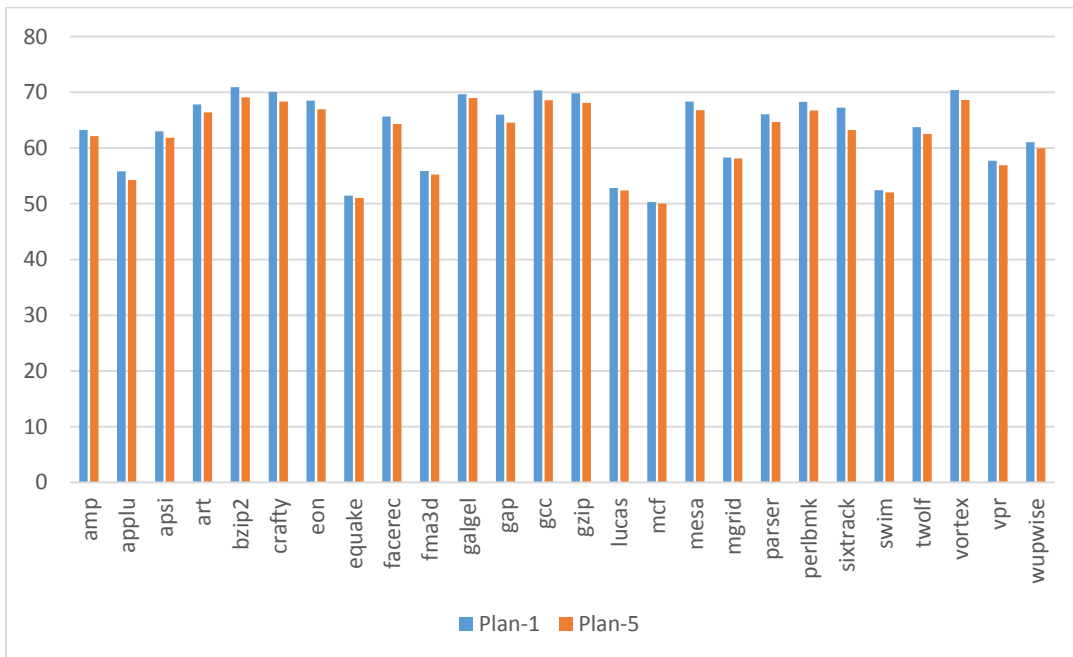Fig 6.4: Difference between Floorplan-1 and Floorplan-4 w.r.t all benchmarks

Fig 6.5: Difference between Floorplan-1 and Floorplan-5 w.r.t all benchmarks
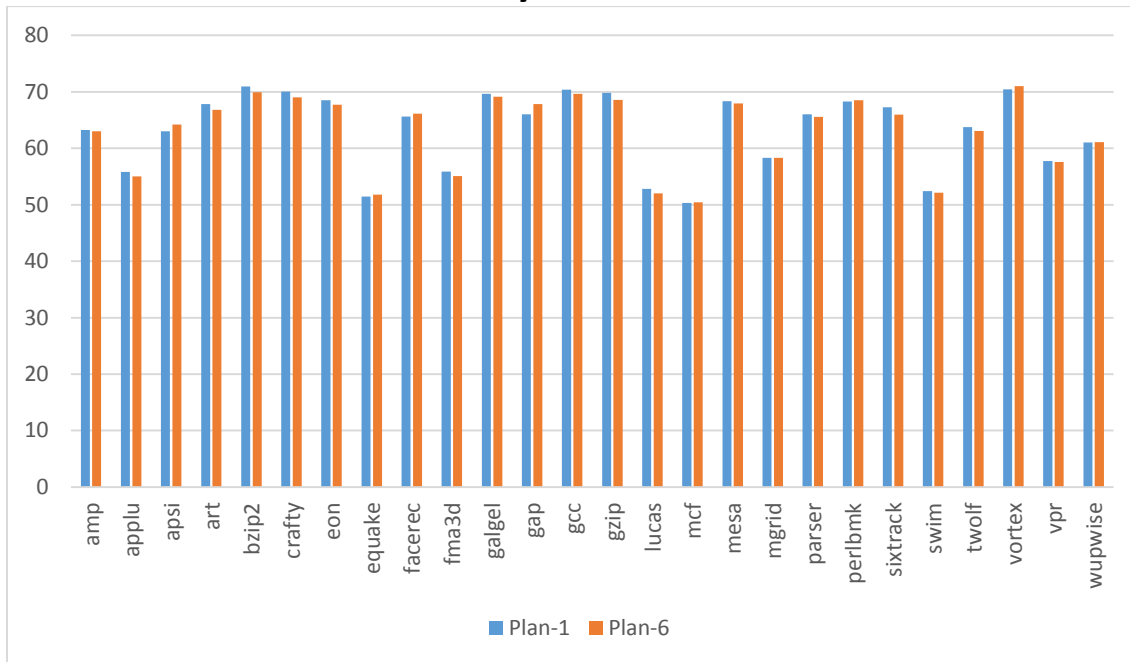
6.3 Effect of L2 Power Density

Fig 6.6: Difference between Floorplan-1 and Floorplan-6 w.r.t all benchmarks

Fig. 6.1 explains effect of power and orientation of blocks of generated floorplans with respect to all benchmarks The left axis shows Peak Temperatures (in ^{o}C). Next figures Fig. 6.2, Fig. 6.3, Fig. 6.4, Fig. 6.5 and Fig. 6.6 show comparison of all generated floorplans with the original one. Of all the graphs only Floorplan-5 (in Fig. 6.5) shows reduced temperatures with respect to all the benchmarks. So Plan-5 is the best among all.

6.4 Effect of Area

Using Floorplans from 1st to 5th and running them in Hotspot Simulator using only gcc benchmark to compare.
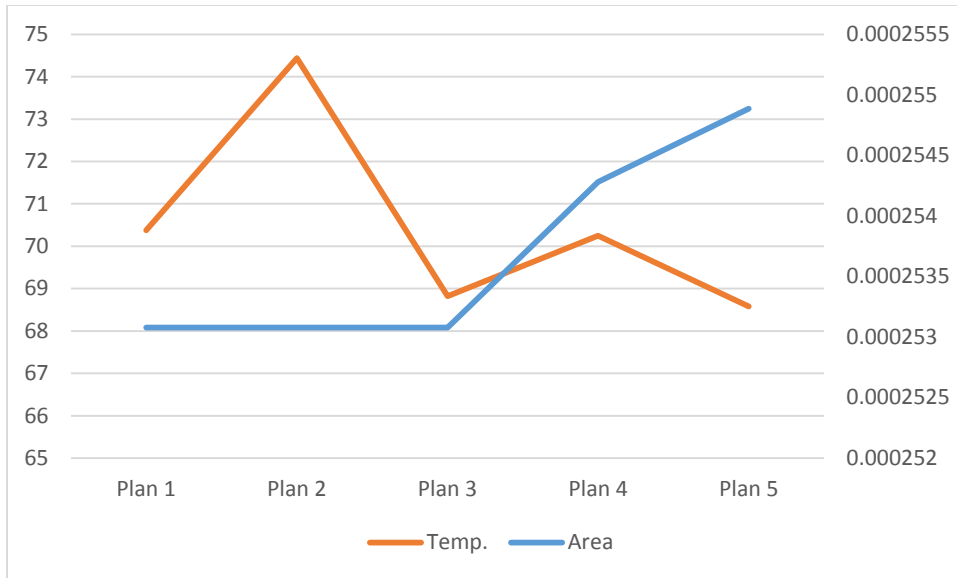
Fig 6.7 Effect of area on peak temperature (of gcc benchmark).

Fig. 6.7 shows the peak temperature in primary y-axis (in ^{o}C) and area is represented in y-axis secondary (in m^2).

6.5 Effect of Area and Power based on Simulated Annealing

Four Floorplans have been generated by Simulated Annealing [8] and QUILT [10] Simulator.
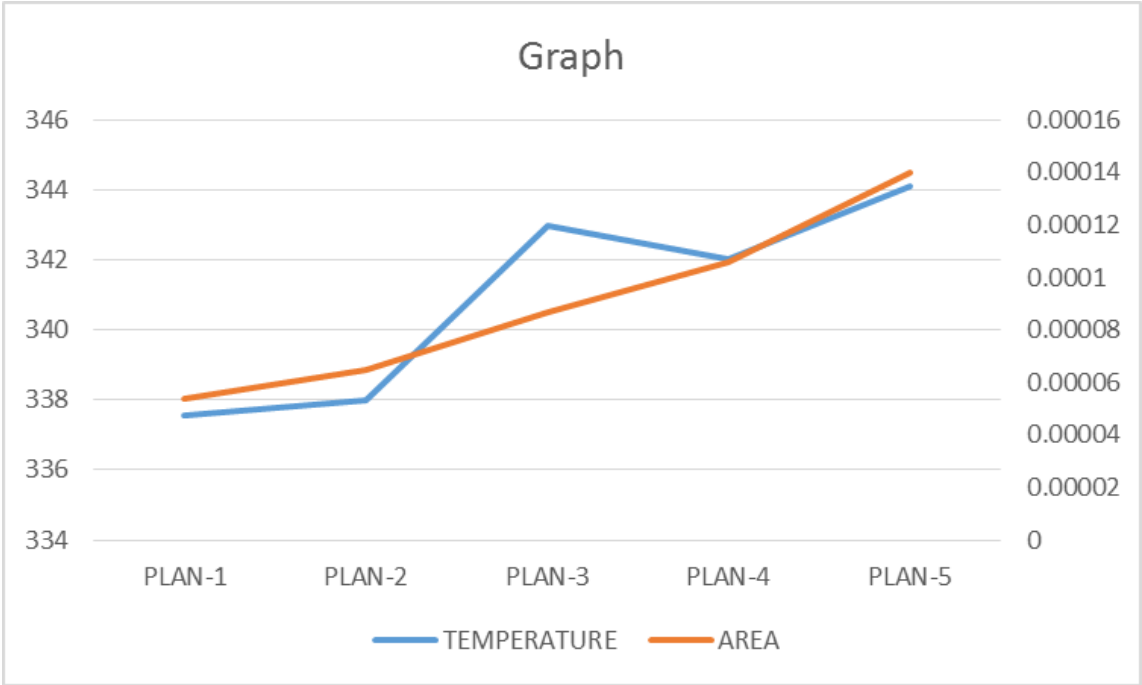

Fig 6.8: Graph of Temperature(in K) and Area trade-off (Simulated Annealing)

Fig. 6.8 shows Temperature(in K) on left (primary) y-axis and area (in m2) in right (secondary) y-axis.

Floorplan 1 has minimum area and low peak temperature at the same time. So, it is a balanced plan good in both factors. So it is the best of all.

After accepting the proper core, one can use the core in any architecture and run multi2sim [9] using any sample program to find the most effective architecture.

Chapter 7
Conclusion

As one can see from the results the ideal floorplans are spotted out in each case. There must be a floorplan with a balance between area and peak temperature. Hotspots [1] are distributed evenly along the floorplan and temperatures of different blocks show us where to make the adjustments. Since every year shows us a new processor with greater speed and greater efficiency, these methods are primitive but very useful when developed further.

With HotSpot 5.02 [2], divisions are more and unnecessary. So, using resources from HotSpot 2.0 helped to show the heat signatures of all the basic blocks. Quilt 1.0 does not support HotSpot 5.02. Using .flp file from HotSpot 2.0 and running it in HotSpot 5.02 bore results.

REFERENCES

1) Temperature Aware Floorplanning**,** Yongkui Han, Israel Koren and Csaba Andras Moritz**,** *Department of Electrical and Computer Engineering University of Massachusetts,Amherst, MA 010032)Po-Hsun and Tsung-Yi Ho*, "Thermal- Aware Bus-Driven Floorplanning" in Low Power Electronics and Design (ISLPED) 2011 International Symposium.

2) http://lava.cs.virginia.edu/hotspot

3) Heat Spreading Aware Floorplan for Chip Multicore Processor1 Liqiang He and Cha Narisu*, College of Computer Science, Inner Mongolia University, Hohhot, P. R. China*

4) A Fast Algorithm for Area Minimization of Slicing Floorplans Weiping Shi*, Member, IEEE*

5) http://www.spec.org

6) Mitigating Memory Wall Effects in High-Clock-Rate and Multicore CMOS 3-D Processor Memory Stacks; P Jacob*, Center for Integrated Electron., Rensselaer Polytech. Inst.,* Troy, NY , A Zia, O Erdogan , P.M. Belemjian

7) Interconnections in Multi-core Architectures: Understanding Mechanisms, Overheads and Scaling Rakesh Kumar⊠, Victor Zyuban_, Dean M. Tullsen*, Dept. of Computer Science and Engineering, University of California*

8) The Theory and Practice of Simulated Annealing, Darrall Henderson, *Department of Mathematical Sciences*, *United States Military Academy*

9) Multi2Sim: a simulation framework for CPU-GPU computing, Rafael Ubal, Perhaad Mistry, Dana Schaa, David Kaeli, *Northeastern University, Boston*

10) http://www.ece.rochester.edu/projects/acal/quilt/faq.html