Performance Evaluation of Computer Memory Hierarchy

Anirban Dutta Choudhury
duttacha@alari.ch
Alie El-Din Mady
madya@alari.ch
Advanced Learning and Research Institute

ALaRI - USI
Lugano, Switzerland
Final version

Jan. 27, 2008

Contents

List of Figures

Chapter 1

Introduction to Computer Memory Hierarchy

1.1 General Purpose Computer Memory Architecture Model

In most cases, the Computer Memory Hierarchy for General Purpose Computers may be summarized as given below [5, 6]:

1. **CPU Registers:** In computer architecture, a processor register is a small amount of storage available on the CPU whose contents can be accessed more quickly than storage available elsewhere. Most, but not all, modern computer architectures operate on the principle of moving data from main memory into registers, operating on them, then moving the result back into main memory.

2. **Level 1 Cache:** A CPU cache may be defined as a volatile memory used by the central processing unit of a computer to reduce the average time to access memory. The cache is a smaller, faster memory which stores copies of the data from the most frequently used main memory locations. As long as most memory accesses are to cached memory locations, the average latency of memory accesses will be closer to the cache latency than to the latency of main memory. L1 cache is the nearest to the processor (most of the times on the same chip) and it is also second fastest memory after the CPU registers.

3. **Level 2 Cache:** L2 cache is slower and bigger cache.

4. **Main Memory (RAM):** This is the Random Access Memory and this is the biggest volatile memory devices, since it loses its data when the power supply is removed.

5. **Hard Disk:** This is nonvolatile memory or non-volatile storage, the slowest and biggest in the computer memory hierarchy that can retain the stored information even when not powered.

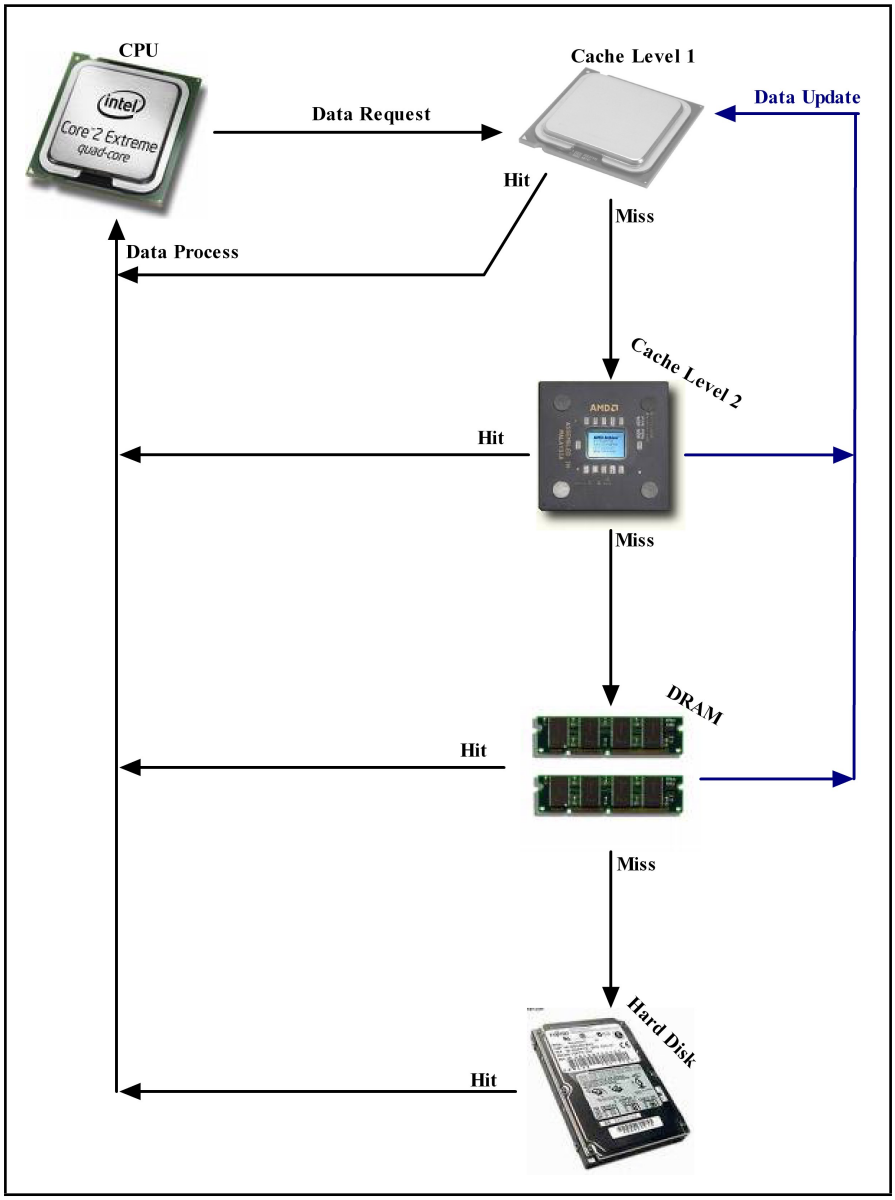

Figure 1.1: General Purpose Computer Architecture Hierarchy

We explained the aforementioned memory structure in the figure 1.1. Nowadays, thanks to top level micro & nano electronics researches and implementation, in general purpose computers we can see different improvisations, like:

1. A large sized L1 and L2 on-chip cache

2. Multi-Processor Personal computers.

1.2 Network-on-Chip Memory Architecture Model

During the last decade, Network on Chip has become a dense area of research in Embedded Systems Research field. Due to tremendous advancement in VLSI, several functional blocks in a single chip has become a reality. Network-on-Chip (NoC) is an emerging paradigm for communications within large VLSI systems implemented on a single silicon chip. In a NoC system, modules such as processor cores, memories and specialized IP blocks exchange data using a network as a *public transportation sub-system* for the information traffic. A NoC is constructed from multiple point-to-point data links interconnected by switches (or routers), such that messages can be relayed from any source module to any destination module over several links, by making routing decisions at the switches.

Chapter 2

Modeling Approaches

2.1 MATLAB Modeling

In our mathematical model, we used the miss rates of different memory components as the input to the system. The output function, also called cost function is a nonlinear function of all the input parameters. The basic form of cost function in our project is taken from the CACTI Tool manual [1].

We used a wide range of possible values of miss rates (0.02 to 0.3 in steps of 0.01) in our project.

```
Ml1 = (0.02:0.01:0.3); % L1 miss rate
Ml2 = (0.02:0.01:0.3); % L2 miss rate
MD  = (0.02:0.01:0.3); % DRAM miss rate
```

As mentioned below, The standard values of power and time costs are taken from different research publications [3, 4].

```
Pl1 = 3; % L1 Power in nJ
Pl2 = 200; % L2 Power in nJ
PD  = 3000; % DRAM Power in nJ
PHD = 1500000000; % disk Power in nJ


Tl1 = 2.2; % L1 access Time in ns
Tl2 = 100; % L2 access Time in ns
TD  = 2000; % DRAM access Time in ns
THD = 2000000; % disk access Time in ns

Al1 = 2.*exp(3 - (Ml1.*100)); % L1 Area in MB
Al2 = Al1 .* (2.*exp(7));     % L2 Area in MB
AD  = Al1 .* (2.*exp(8));     % disk Area in MB
```

2.1.1 Model not considering Hard disk

Following is the power and time cost function defined for such systems. These equations along with other following equations (equations having functions of the parameters Ml1, Ml2, & MD) are directly influenced by the widely known cache hit-miss formulas [5].

$$Power\ Cost = \frac{(Pl1 + Ml1 \times Pl2 + Ml1 \times Ml2 \times PD)}{(Pl1 + Pl2 + PD)} \tag{2.1}$$

where,
Pl1 = Cache Level 1 Power Consumption,
Pl2 = Cache Level 2 Power Consumption,
PD = DRAM Power Consumption,

$$Time\ Cost = \frac{(Tl1 + Ml1 \times Tl2 + Ml1 \times Ml2 \times TD)}{(Tl1 + Tl2 + TD)} \tag{2.2}$$

where,
Tl1 = Cache Level 1 Time Consumption,
Tl2 = Cache Level 2 Time Consumption,
TD = DRAM Time Consumption,

Following is the Area Cost function used in our project[2]. As shown in the following equation, all the different components of Area Cost *except Hard disk* are considered.

$$Area\ Cost = \frac{1}{Memory\ Area\ Efficiency} = \frac{Al1 + Al2 + AD}{IPC} \tag{2.3}$$

where,
Al1 = Cache Level 1 Time Consumption,
Al2 = Cache Level 2 Time Consumption,
AD = DRAM Time Consumption,
IPC = Instruction per Cycle

We derived an empirical formula from the standard results published in research papers for modeling IPC using miss rate parameters. IPC is defined as:

$$IPC = (1 - Ml1) \times 1.45 \tag{2.4}$$

2.1.2 Model considering Hard disk

Now, we will define the individual components of the aforementioned Cost Function.

$$Power\ Cost = \frac{(Pl1 + Ml1 \times Pl2 + Ml1 \times Ml2 \times PD + Ml1 \times Ml2 \times MD \times PHD)}{(Pl1 + Pl2 + PD + PHD)} \tag{2.5}$$

where,
Pl1 = Cache Level 1 Power Consumption,
Pl2 = Cache Level 2 Power Consumption,
PD = DRAM Power Consumption,
PHD = Hard disk Power Consumption

$$Time\ Cost = \frac{(Tl1 + Ml1 \times Tl2 + Ml1 \times Ml2 \times TD + Ml1 \times Ml2 \times MD \times THD)}{(Tl1 + Tl2 + TD + THD)} \quad (2.6)$$

where,
Tl1 = Cache Level 1 Time Consumption,
Tl2 = Cache Level 2 Time Consumption,
TD = DRAM Time Consumption,
THD = Hard disk Time Consumptio

The Area Cost function in this model is same as defined in the model not considering Hard disk (vide equation 2.3).

2.1.3 Final Cost Function

In the final cost function, we normalized each component to make sure each of them affects the final value in a similar way. So the cost function doesn't have any unit of different components mentioned earlier (i.e. nJ, ns and MB) and the *final cost function* is defined as following:

$$Cost\ Function = \frac{Power\ Cost}{Max.\ Power\ Cost} + \frac{Time\ Cost}{Max.\ Time\ Cost} + \frac{Area\ Cost}{Max.\ Area\ Cost} \quad (2.7)$$

In our project we considered two models and compared the cost functions. One model considers the presence of nonvolatile memory(i.e. Hard disk) in the architecture while the other model doesn't take into account Hard disk. In the following two sections, we will represent the different components of the cost functions for both the cases.

2.2 Queuing Network Modeling

2.2.1 Introduction

Queuing Network Modeling Approach is a particularly applicable for the kind of system we are diskussing. We used a special modeling tool named *JMT version 0.7.3* developed in *Politechnico the Milano* [8]. The Java Modeling Tools (JMT) is a free open source suite for performance evaluation, capacity planning and modeling of computer and communication systems. The suite implements numerous state-of-the-art algorithms for the exact, asymptotic and simulative analysis of queueing network models, either with or without product-form solution.

2.2.2 General Purpose Computer Memory Architecture Model

Here, we started with the model in 2.1. In this model, the miss rates (constant) are :
Ml1 = 0.05
Ml2 = 0.05
MD = 0.1

The model is a closed model with instruction buffer queue length = 8.

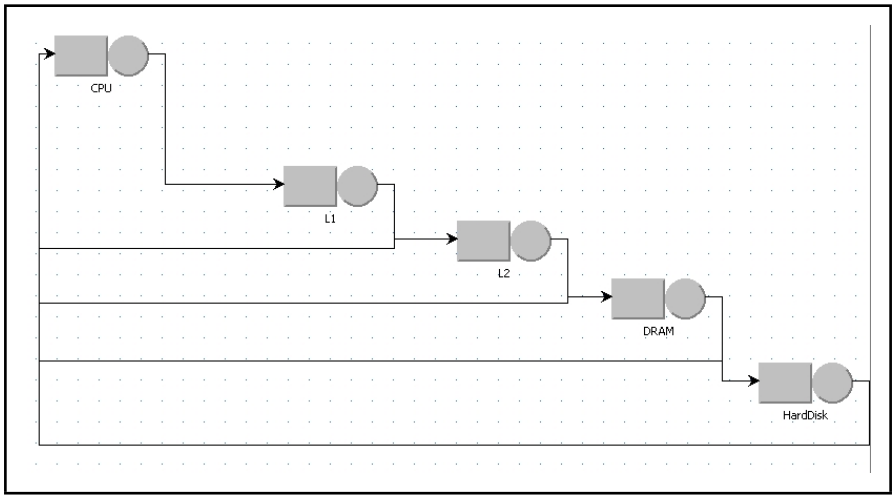

Figure 2.1: General Purpose Computer Architecture Queuing Network Model

Result

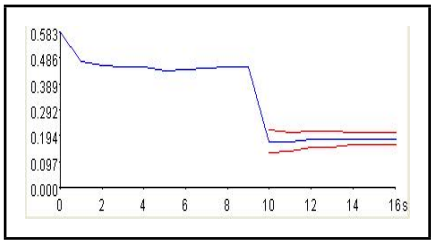

Figure 2.2: QL of CPU

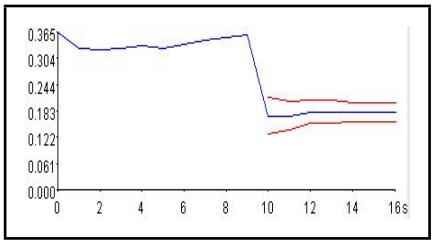

Figure 2.3: QL of L1

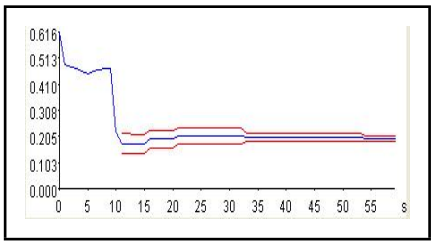

Figure 2.4: QL of L2

Figure 2.5: QL of DRAM

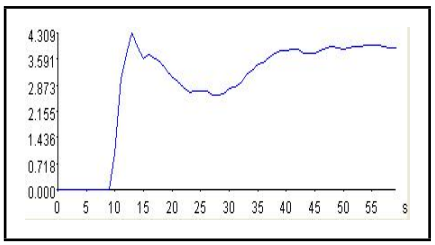

Figure 2.6: QL of HD

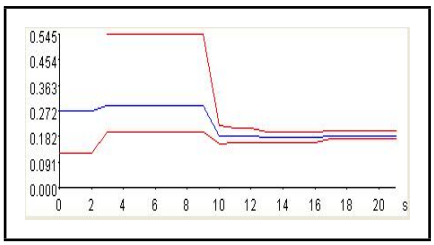

Figure 2.7: TP of CPU

To reflect the variations of the nature of the big (and slow) memory components (e.g. Hard disk), we also checked the same model with Hard disk having a normal distribution with a mean value 2×10^4 ns (same as the previous model) and a standard deviation of 100 ns (refer to figure 2.13 and 2.14).

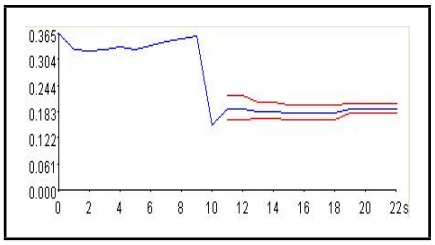

Figure 2.8: TP of L1

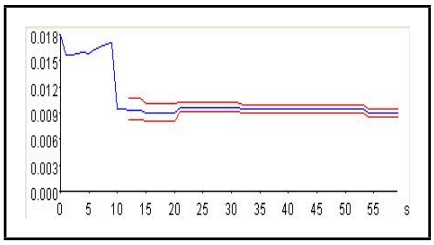

Figure 2.9: TP of L2

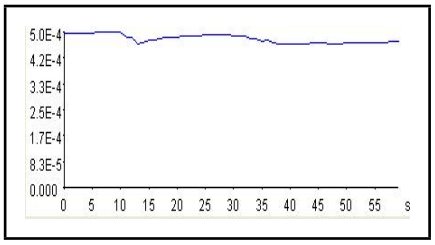

Figure 2.10: TP of Dram

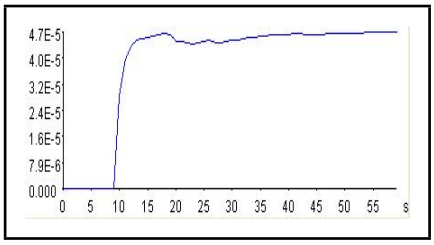

Figure 2.11: TP of HD

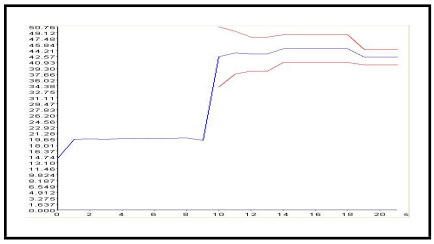

Figure 2.12: System Response Time

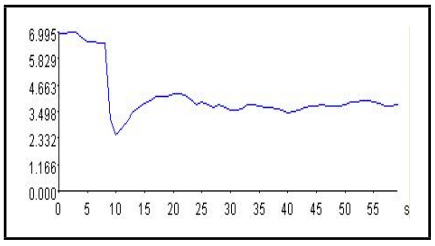

Figure 2.13: QL of DRAM (Normal Distribution)

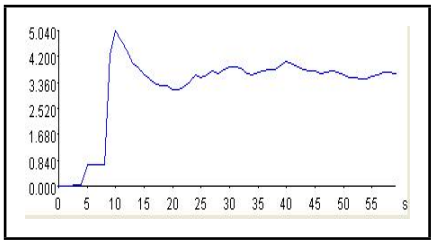

Figure 2.14: QL of HD (Normal Distribution)

In this part of the implementation, we are modeling General Purpose Computers with only one processor and consequent memory structure dedicated to that processor. As shown in the figure 2.15, this system is explained as following. If there is a hit after a cache or other memory search, the value goes back to CPU and L1 cache values are updated (Fork 0 and Fork 1). In case of a miss, the search proceeds to the next level of memory. Processor and L1 cache have more than one inputs. And we use joins (Join 0 and Join 1) to constrain the exponential increment of the number of tokens in the system.Otherwise, it would become an unstable system with 2 forks increasing the number of jobs exponentially.

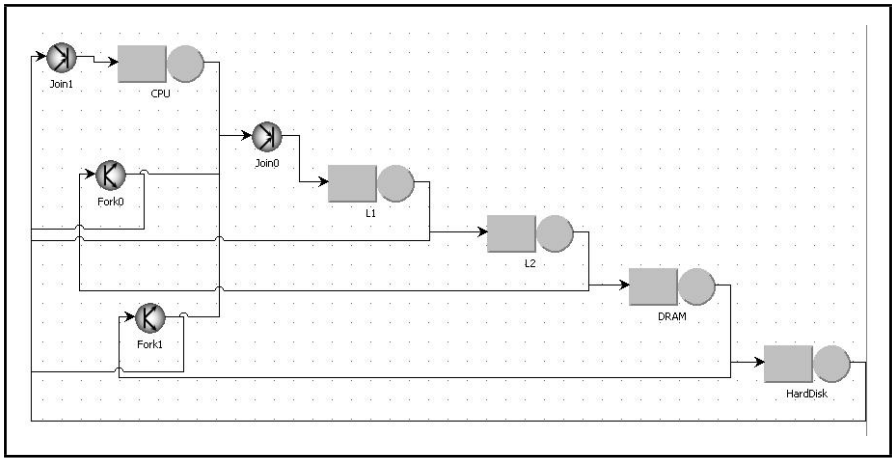

Figure 2.15: General Purpose Computer Architecture Queuing Network Model (with Fork & Join)

Here when we tried to run the system with instruction buffer length = 8, the JMT result window was blank (figure 2.16).

The initial number of instructions (number of jobs in the model) was 10000. The model was run with 2 different set of parameters, namely:

1. **Modeling with High Miss Rate:** L1 miss rate = 0.1, L2 miss rate = 0.05 and DRAM miss rate = 0.01.

2. **Modeling with Low Miss Rate:** L1 miss rate = 0.05, L2 miss rate = 0.005 and DRAM miss rate = 0.001.

Figure 2.16: General Purpose Computer Architecture Queuing Network Model (NOT Working !)

Result (with N = 10000)

1. **Low Miss Rate**

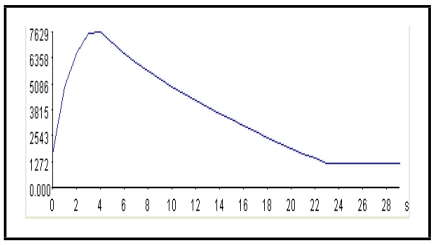

Figure 2.17: Processor Queue Length

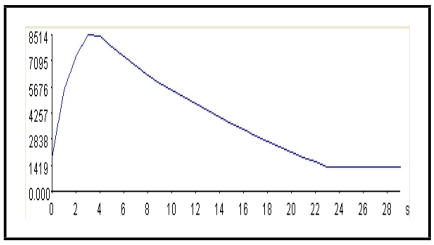

Figure 2.18: Processor Queue Time

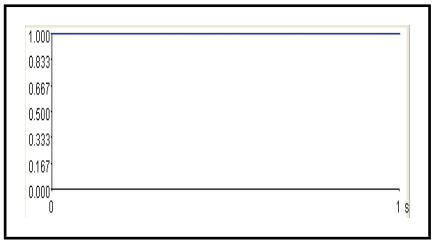

Figure 2.19: Processor Throughput

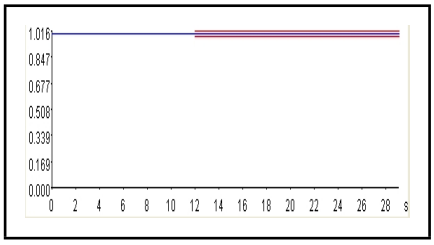

Figure 2.20: L1 Queue Length

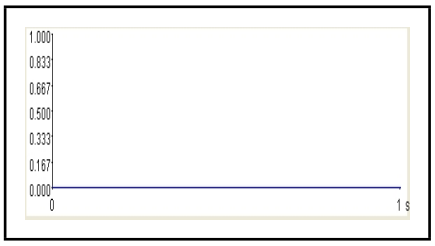

Figure 2.21: L1 Queue Time

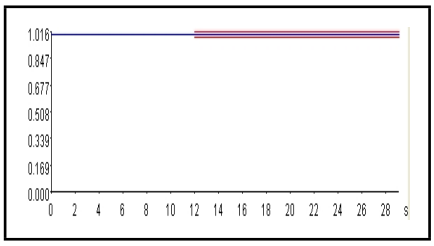

Figure 2.22: L1 Throughput

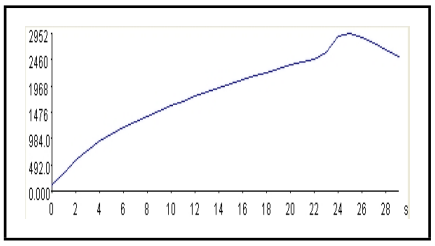

Figure 2.23: L2 Queue Length

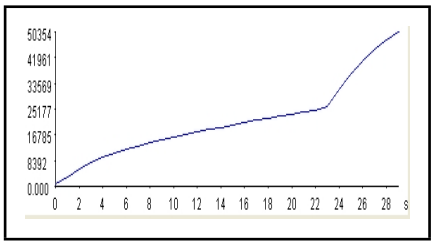

Figure 2.24: L2 Queue Time

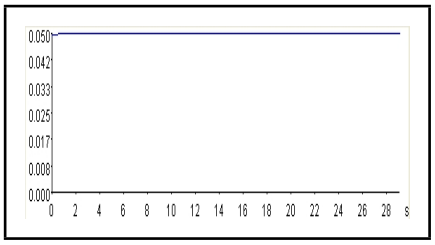

Figure 2.25: L2 Throughput

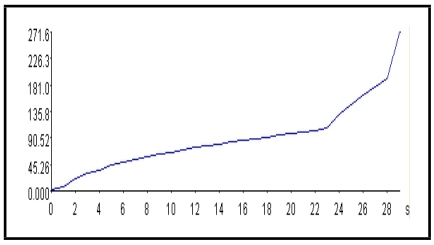

Figure 2.26: DRAM Queue Length

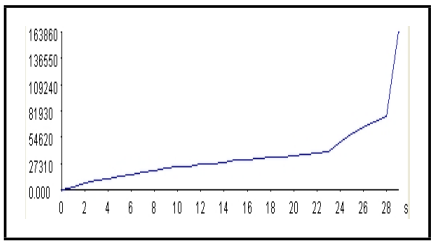

Figure 2.27: DRAM Queue Time

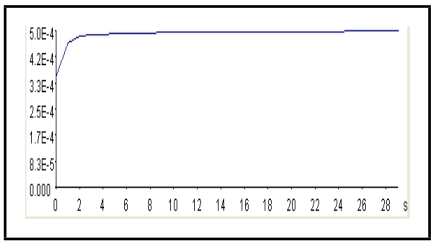

Figure 2.28: DRAM Throughput

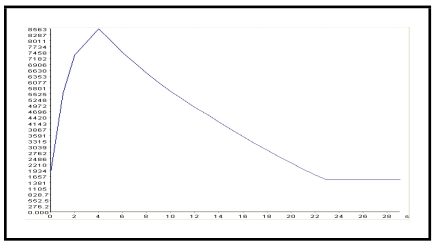

Figure 2.29: System Response Time

2. **High Miss Rate**

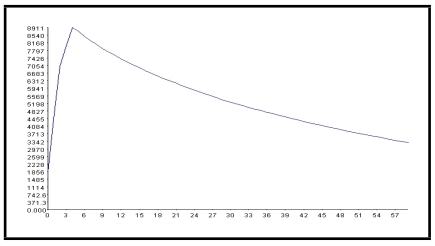

Figure 2.30: Processor Queue Length

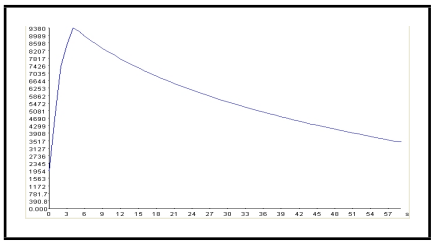

Figure 2.31: Processor Queue Time

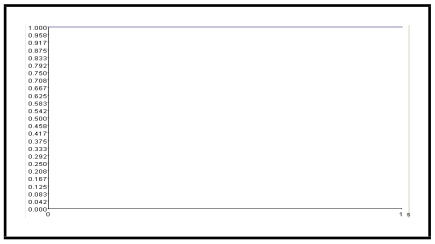

Figure 2.32: Processor Throughput

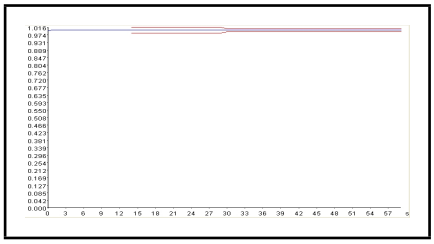

Figure 2.33: L1 Queue Length

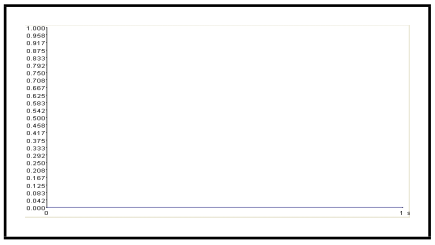

Figure 2.34: L1 Queue Time

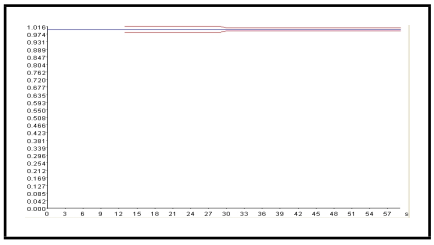

Figure 2.35: L1 Throughput

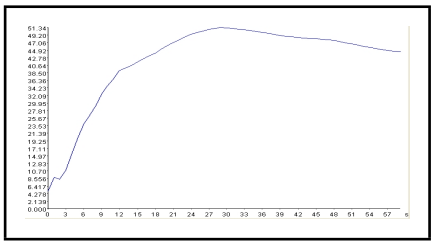

Figure 2.36: L2 Queue Length

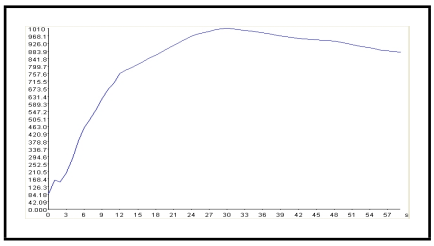

Figure 2.37: L2 Queue Time

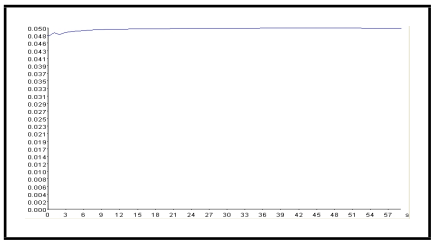

Figure 2.38: L2 Throughput

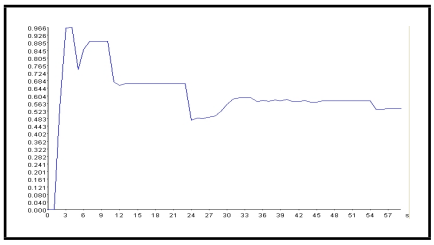

Figure 2.39: DRAM Queue Length

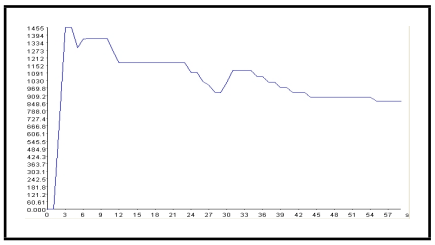

Figure 2.40: DRAM Queue Time

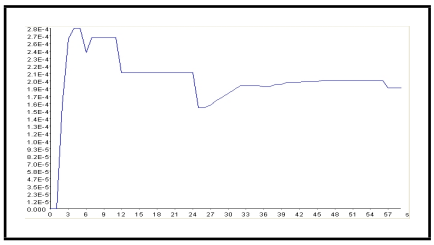

Figure 2.41: DRAM Throughput

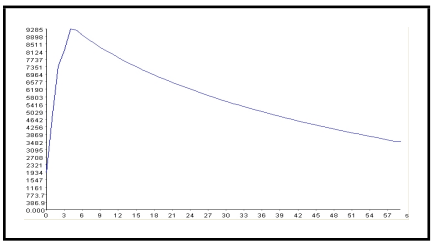

Figure 2.42: System Response Time

2.2.3 Network-on-Chip Memory Architecture Model

In the Network-on-Chip Architecture environment, the systems have typically several resources for the same job(like more than one processor in a single system). As shown in 2.43, we have two processors having dedicated L1 cache memory but sharing L2 cache and main memory. This is an open source model with 2 sources (CP schedulers) and one sink (jobs reaching this point are served, in other words the memory execution is finished).

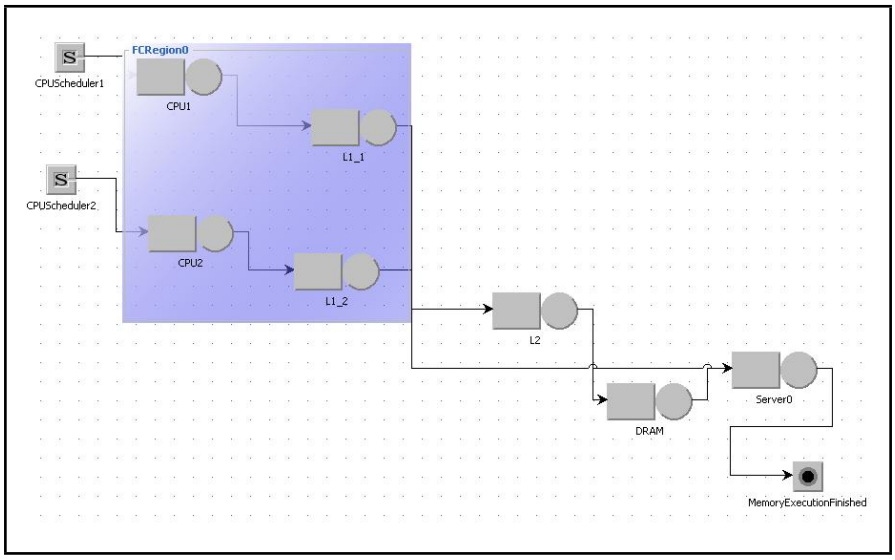

Figure 2.43: NoC Queuing Network Model

In NoC architecture, we used the following parameters - L1 miss rate = 0.05 and L2 miss rate = 0.3.

Result : NoC

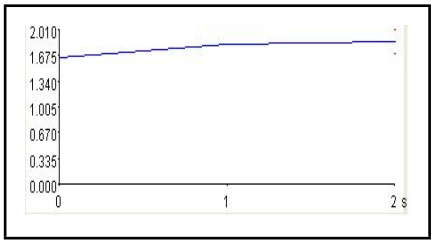

Figure 2.44: NoC QL of CPU1

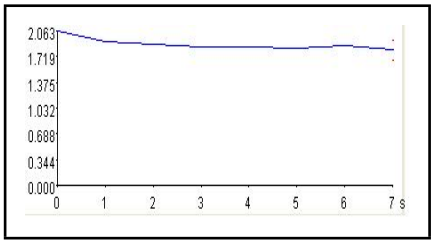

Figure 2.45: NoC QL of CPU2

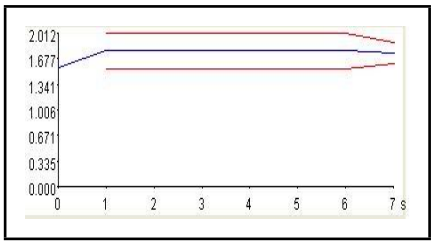

Figure 2.46: NoC QL of 1st L1

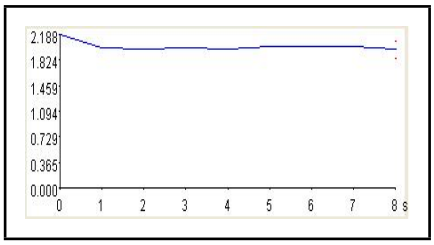

Figure 2.47: NoC QL of 2nd L1

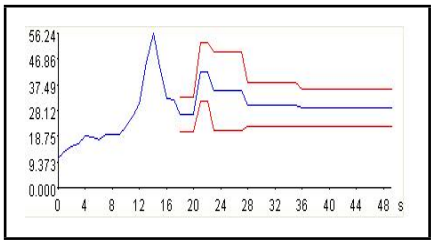

Figure 2.48: NoC QL of L2

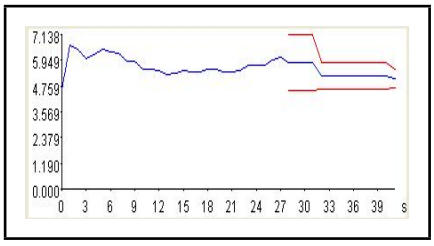

Figure 2.49: NoC QL of DRAM

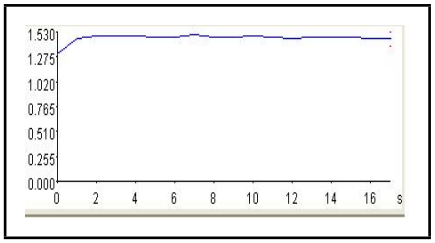

Figure 2.50: NoC QT of CPU1

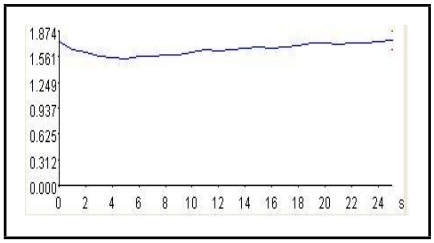

Figure 2.51: NoC QT of CPU2

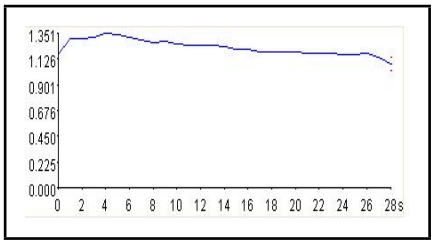

Figure 2.52: NoC QL of L1 of CPU1

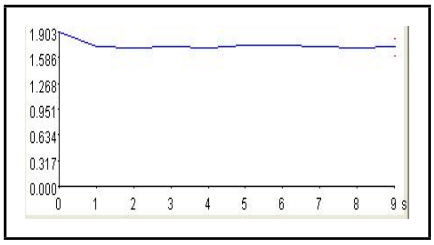

Figure 2.53: NoC QL of L1 of CPU2

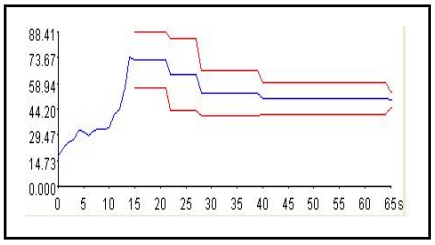

Figure 2.54: NoC QT of L2

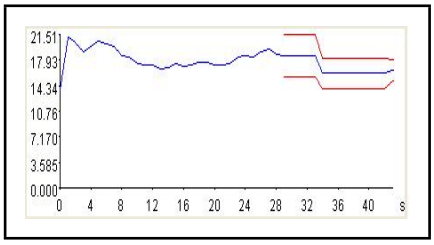

Figure 2.55: NoC QT of DRAM

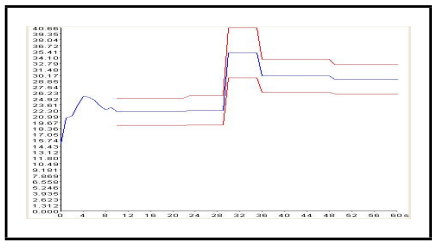

Figure 2.56: NoC System Response Time

Chapter 3

Results

3.1 MATLAB Simulation Result

As shown in chapter 2.7, all the individual components of the final cost function are normalized before adding up. In figure 3.1, we can see the variation of each component over a set of design space points of Ml1, Ml2, MD and MHD. And then in the 4th segment of 3.1, we can see the normalized components added up to generate the *Final Cost Function*.

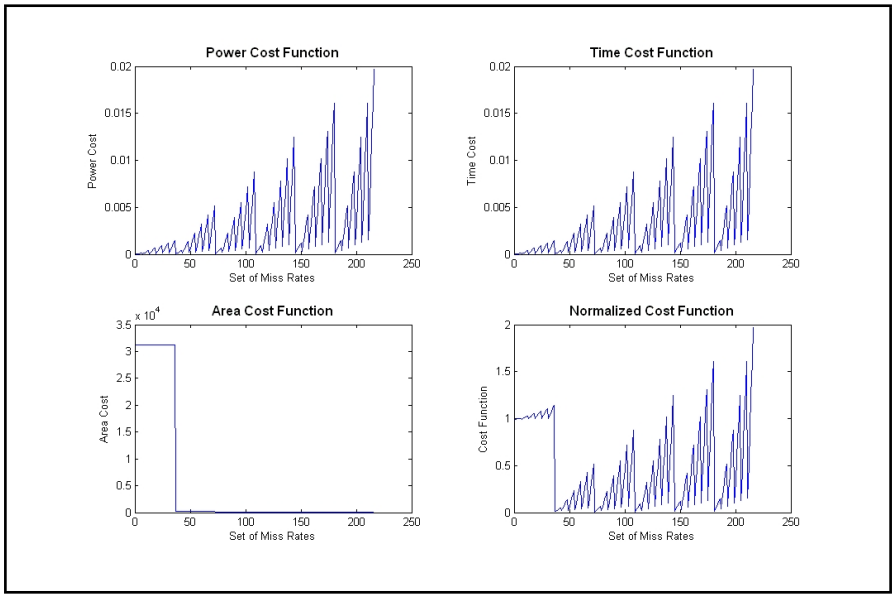

Figure 3.1: Normalized Cost Function

3.1.1 Simulation Not Considering Hard disk

Here in 3.2 we can see the results of MATLAB simulation not considering Hard disk in the system. The color of the graph changes with the height of the 3-D plane. As expected from the basic equations diskussed, Power and Time cost functions are increasing with L1 and L2 miss rate. On the other hand, Area cost function is decreasing abruptly with increasing miss rates. The *Final Cost Function* reflects the nature of all the graphs. The design points with the lowest heights are the optimum combination for our system.

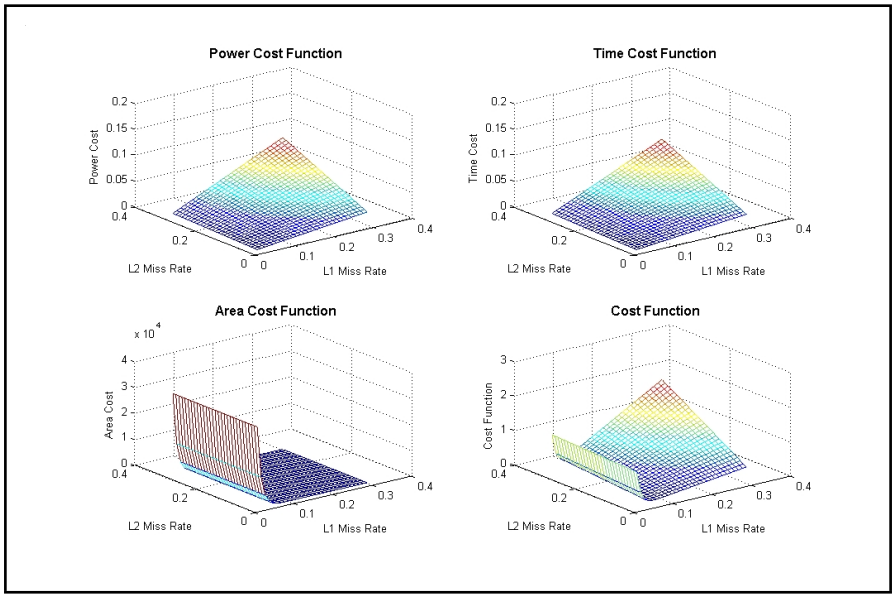

Figure 3.2: Cost function NOT Considering Hard disk

3.1.2 Simulation Considering Hard disk

In 3.3, we have 4 dimensions, namely Ml1, Ml2, MD and MHD. So we have drawn slices along axes of the miss rates to show the *Final Cost Function*. The red end of the colors shown in the graph have a higher value of final cost function, while the black end of the spectrum have a low final cost function value. The intensity of the color changes with the value. The trade off is clear as the lighter region in Area Cost Graph is just opposite of Power and Time cost graph.

As a result, we can say the most optimum points are somewhere middle in the 3-D region (Neither in the extreme left as in Power/Time Cost graph, nor in the extreme right as in Area Cost graph).

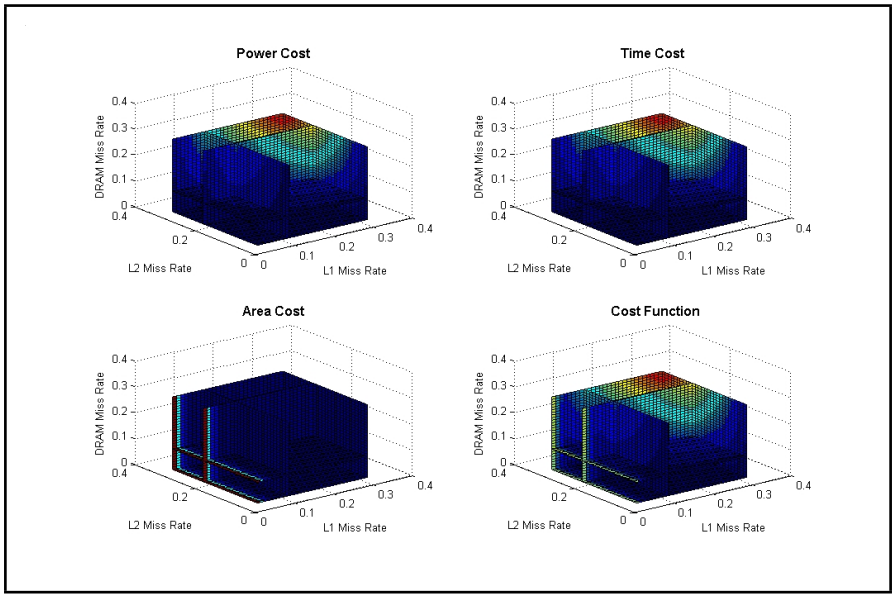

Figure 3.3: Cost function Considering Hard disk

3.1.3 Conclusion

The findings and explanations from this queuing network model is given below:

1. In General Purpose Computer Memory Architecture,
   (a) In low miss rate systems, the CPU queue length and time is staying a bit in the maximum value and then coming back. While in the hight miss rate system there is a sharp maximum point and then the curve is coming down with a steep slope. This can be explained as following. The low miss rate system have hit hit rates. As a result the jobs are driven to CPU more fast and the system is holding the maximum value for a longer time.
   (b) In our system, the level 1 cache in considered on-chip and as fast as CPU, so as expected, the L1 queue time is constant at 0, as there will no queue formed at all. Following the same logic, L1 throughput is also 1 all the time.
   (c) L2 and DRAM having a an exponentially large service time, the queue time and queue length of both the severs are going up.
2. In Network-on-Chip Memory Architecture,
   (a) In NoC, there are L1 cache dedicated to 2 processors. In both of them, the queue length is stable around 2.
   (b) One very important observation is that DRAM is not following any specific pattern. This is because in our simulation, L2 is modeled with a high miss rate. So a good percent of jobs (which are again mixed bunch of jobs) are reaching DRAM. And while DRAM is working on that some miss happens in L2 and that again makes an increment of the queue length of DRAM.
   (c) System Response Time is having a stable value initially, but then it goes up reflecting the queues in L1,L2 and DRAM. And when the queue lengths of different components becomes stable, the System Response Time decrease again to a stable value.

Bibliography

[1] Premkishore Shivakumar and Norman P. Jouppi , "CACTI 3.0: An Integrated Cache Timing, Power, and Area Model"

[2] J. Huh, D. Burger, Stephen W. Keckler, "Maximizing Area Efficiency for Single-Chip Server Processors"

[3] Smail MAR, Samy MEFTALI, Jean-Luc DEKEYSER, "Power Consumption Awareness in CacheMemory Design with SystemC,". *The 16th International Conference on Microelectronics, 2004. ICM 2004 Proceedings.*

[4] Giovanni De Micheli, Yung-Hsiang L, "Adaptive Hard Disk Power Management on Personal Computers," *IEEE Great Lakes Symposium on VLSI, 1999*

[5] John L. Hennessy, David A. Patterson, "Computer Architecture: A Quantitative Approach, Third Edition," *The Morgan Kaufmann Series in Computer Architecture and Design*

[6] www.wikipedia.org

[7] ALaRI Classnote, Prof. G. Serazzi

[8] JMT 0.7.3 Manual