Control of Multi-Core CPU with Thermal Constraints

by

Eng. Mohamed Adel Mohamed Elsawaf

A thesis submitted to the Faculty of Engineering at Cairo University in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
In

ELECTRICAL POWER AND MACHINES

Under the supervision of

Prof. Dr. Abdel Latif Mohamed Elshafei

Electrical Power and Machines Department, Cairo University, Egypt

Associate Prof. Dr. Hossam Aly Hassan Fahmy

Engineering Electronics and Communication Department,

Cairo University, Egypt

FACULTY OF ENGINEERING
CAIRO UNIVERSITY
GIZA, EGYPT
September 2010

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ASMP	Asymmetric Multi-Processing
AVS	Adaptive Voltage Scaling
BXU	Branch Execution Unit
CMP	Chip Multiprocessors
CPU	Central Processing Unit
C	Physical Capacitance
°C	Celsius
DPS	Distributed Parameter System
DTM	Dynamic Thermal Management
DVFS	Dynamic Voltage And Frequency Scaling
FL	Fuzzy Logic
FC	Fuzzy Logic Controller
FPU	Floating Point Unit
FXU	Fixed Point Integer Unit
f	Clock Frequency
f_o	Open Loop Frequency At Time n
f_n	Operating Frequency At Time n
f_e	Frequency Deviation
$f_{\Delta e}$	Delta Deviation Frequency
$f_{\max}$	Maximum Frequency
HPC	High Performance Computing
GND	Ground Voltage
IFU	Instruction Fetch Unit
IDU	Instruction Decode Unit
IHS	Integrated Heat Spreader
ILP	Instruction-Level Parallelism
ISU	Instruction Sequencing Unit
ITRS	International Technology Roadmap For Semiconductors
I_{SC}	Short-Circuit Current
LSU	Load Store Unit
L2	Level 2 Cache
L3	Level 3 Cache
MC	Memory Controller
MCM	Multi-Chip Module
MSF	Membership Function
MSMV	Multi-Supply Multi-Voltage
MTCMOS	Multi-Threshold CMOS
MULTI-VT	Multiple Switching Threshold
P	Proportional Controller
PI	Proportional Integrator Controller

P_{CMOS}	Total CMOS Power
$P_{Dynamic}$	CMOS Dynamic Power
P_{Static}	CMOS Static Power
$P_{Shortcircuit}$	CMOS Short-Circuit Power
PSO	Power Shut-Off
Q	Heat conduction
RISC	Reduced Instruction Set Computer
SMP	Symmetric Multi-Processing
SMT	Simultaneous Multi-Threading
SOI	Silicon-On-Insulator
TDP	Thermal Design Power
TLP	Thread-Level Parallelism
TM	Thread Migration
TSC	Thermal Spare Cores
TTV	Thermal Test Vehicle
T_o	Open Loop Temperature
T_{ss}	Steady State Temperature
T_{tsc}	Temperature That Triggers TSC Process
T_{th}	CPU Throttling Temperature
T_e	Temperature Deviation
$T_{\max}$	Maximum Temperature
$T_{Control}$	Temperature That Triggers Controller
$T_{\Delta e}$	Delta Deviation Temperature
T_n	Temperature At Time n
t_{DTM}	Time Required For DTM Controller Response
t_{omx}	Time Required To Reach Open Loop Maximum Temperature
t_{oc}	Time Required To Reach Open Loop Control Temperature
$U_{\max}$	CPU Maximum Utilization
U_{safe}	CPU Safe Utilization
VTCMOS	Voltages Threshold CMOS
VB	Bias Voltages
VT	Multiple Threshold Voltages
V_{dd}	Supply Voltage
V_{th}	Threshold Voltage
V_b	Bias Voltages
α	CMOS Switching Activity Factor
η	Number Of Transistors In The Design
K	The Feature Size Scaling Per Technology Generation
β	The gain factor $\mu A / V^2$ of an MOS Transistor
σ	Thermal conductivity
τ	CMOS Gate Delay
κ	The Characteristic of An Average Device
i_{leak}	Is A Technology Parameter Describing The Per-Device Sub Threshold Leakage

ACKNOWLEDGMENTS

I would like to thank all the people who have helped me to finish my master studies. Namely, I thank my supervisors for many interesting and investigating suggestions they gave me, and for the fruitful (and often long) discussions we had.

Thanks to lava lab team, university of Virginia USA for their efforts and support of “Hotspot” thermal simulator.

Finally, I thank my wife and my children for their nearly unlimited patience.

ABSTRACT

On-chip thermal analysis calculates and reports thermal gradients or variations in operating temperature across a design. This analysis is increasingly important for the advanced digital integrated circuits (ICs). At today's 65nm and 45nm technologies, adding cores to CPU chip increases its power density and leads to thermal throttling. Advanced control techniques give a solution to the central processing unit (CPU) thermal throttling problem. The air cooling limitations is affecting the portable devices. Advanced control techniques offer solutions for the CPU dynamic thermal management (DTM). This thesis objective is to minimize thermal CPU throttling effect and ensure stable CPU utilization using fuzzy logic control. The thesis focuses on the design of a DTM controller based on fuzzy logic control. This approach reduces the problem design time as it is independent of the CPU chip and its cooling system transfer functions. Towards this objective, a thermal model similar to a real IBM CPU chip containing 8 cores is built. This thermal model is integrated to a semiconductor thermal simulator. The open loop response of the CPU chip is extracted. This CPU chip thermal profile illustrates the CPU thermal throttling. The proposed DTM controller design is based on 3D fuzzy logic. There are many cores within CPU chip, each of them is a heat source. The correlation between these cores temperatures and their operating frequencies improves the DTM response. The 3D fuzzy controller takes into consideration these correlations. The traditional 2D fuzzy suffers from the rule explosion phenomena. The thesis introduces a new DTM technique called "Thermal Spare Core" algorithm (TSC). Thermal Spare Core (TSC) is a completely new DTM algorithm. The thermal spare cores (TSC) is based on the reservation of cores during low CPU utilization and activate them during thermal crises. The reservation of some cores as (TSC) doesn't impact CPU over all utilization. These cores are not activate simultaneously due to limitations. The semiconductor technology permits more cores to be added to CPU chip. That means there is no chip area wasting in case of TSC. The TSC is a solution of the Multi-Core CPU thermal throttling problem.

PUBLISHED PAPER

M. A. elsawaf, H. A. Fahmy, A. L. Elshafei, "CPU Dynamic Thermal Management via Thermal Spare Cores", IEEE Semiconductor Thermal Measurement and Management Symposium, Semi-Therm 25th, 2009, Pages 139-145.

http://IEEExplore.IEEE.org/xpl/freeabs_all.jsp?isnumber=4810723&arnumber=4810755&count=65&index=28

Chapter 1

Introduction

1.1 Motivation

We live in a computer controlled epoch. We do not even realize how often our lives depend on machines and their programming. For example, mobile handsets, portable electronic devices, laptops, medical instruments, and many other devices all depend on digital processors in our everyday lives. There is no doubt that the size and the weight of these portable equipments is affecting their utilization. Unfortunately, there are many factors affecting the portability of electronic systems. The power consumption is affecting battery. Efficient cooling of portable electronic devices is becoming a problem due to air cooling limitations.

Other electronic devices cooling techniques like liquid cooling; but it is not valid for portable devices. The semiconductor technology is permitting higher speed and smaller size devices. Increasing the central-processing unit (CPU) speed of portable devices is limited by the air cooling capacity [1]. The CPU reaches the maximum allowed temperature frequently. The CPU cooling system is unable to reduce the temperature to the safe limits. It is mandatory to reduce the CPU operating frequency to avoid the chip hardware damage. This phenomenon is “the CPU thermal throttling” [2]. On-chip thermal analysis is becoming mandatory. The advanced control techniques can give a solution to the CPU thermal throttling problem. On-chip thermal analysis calculates and reports thermal gradients or variations in operating temperature across a design. This analysis is increasingly important for the advanced digital integrated circuits (ICs) created at today’s 65nm and 45nm technologies. Increasing power density, multi-function designs, and the use of advanced low-power design techniques all lead to increased on-chip temperature gradients [3].

On-chip temperature gradient is a design challenge. Many technology factors affect the chip temperature gradients. In terms of the technology factors, power density (power per unit area) is increasing with each new technology node [4]. After all, smaller geometries enable more functionality to be fit within the same area of a die. This aspect enables design teams to commit to larger and larger designs. At the same time, it significantly increases heat generation, which can result in high thermal gradients [3]. As shown in Figure 1.1, adding more cores to the CPU chip increase the total power consumption [5]. Figure 1.2 illustrates the maximum number of cores per chip and their maximum operating frequencies [5]. It is common to think that more functionality simply equates to greater power consumption.

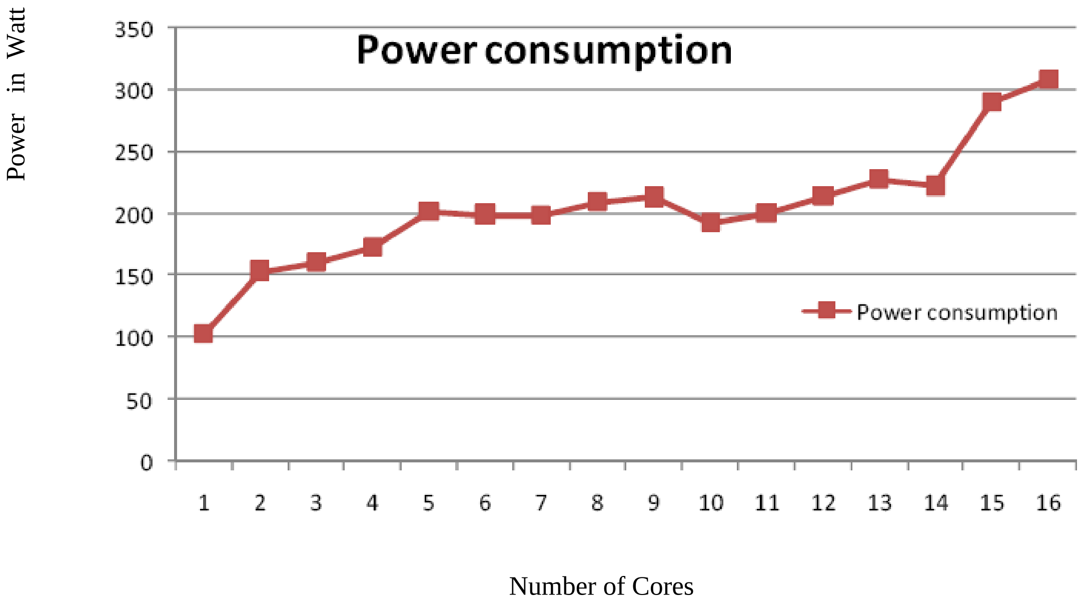


Figure 1.1 Number of cores per CPU vs. the power consumption [5].

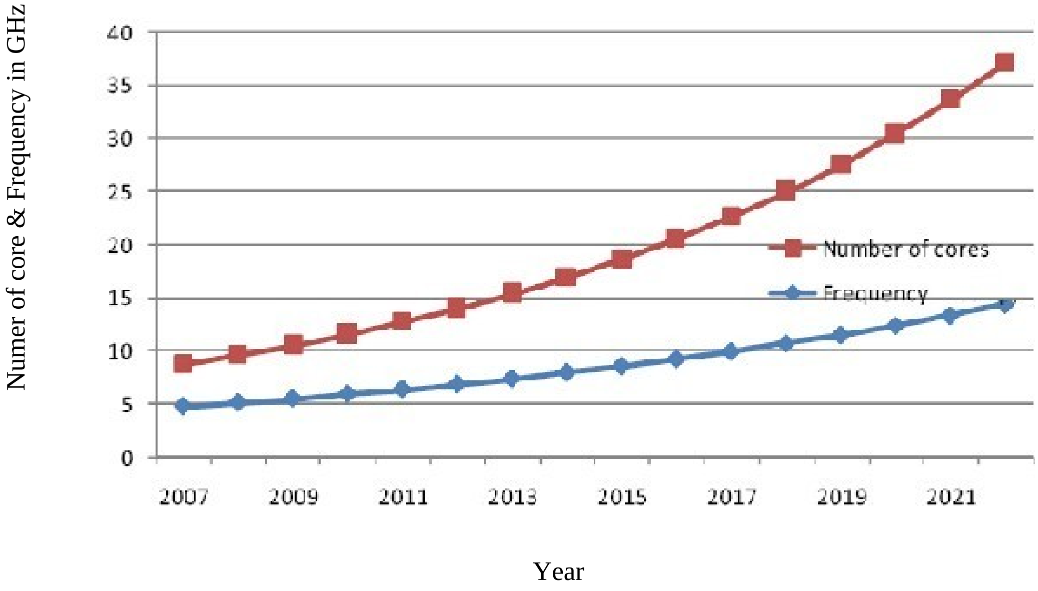


Figure 1.2 - Multi-Core CPU evolutions [5].

CPU cores run relatively hot while on-chip memory tends to run relatively cold. The result is an ever-varying mish-mash of “hot” and “cold” spots that depend on the mode of operation. A cell phone is a good example of this type of design. The act of creating a text message will exercise certain functionality, which creates a specific thermal profile. But the act of transmitting this message will exercise different functionality, which results in a different profile. The same can be said for using the cell phone to make a voice call, play an mp3 file, take a picture, and so forth. The resulting temperature variation across a chip is typically around 10° to 15°C. If this temperature distribution is not managed; then temperature variation will be as high as 30° to 40°C [6].

The CPU power dissipation comes from a combination of dynamic power and leakage power. Dynamic power is a function of logic toggle rates, buffer strengths, and parasitic loading. The leakage power is function of the technology and device characteristics. Thermal-analysis solutions must account for both causes of power. In Figure 1.3 the thermal profile of a CPU chip is showing the temperature variation across the chip surface.

This phenomenon is due to the variation of the power density according to each function block design. This power density distribution generates "hotspots" and “coldspots” areas across the CPU chip surface [3]. The high CPU operating temperature increases leakage current degrades transistor performance, decreases electro migration limits, and increases interconnect resistivity [6]. In addition, leakage current increases the power consumption.

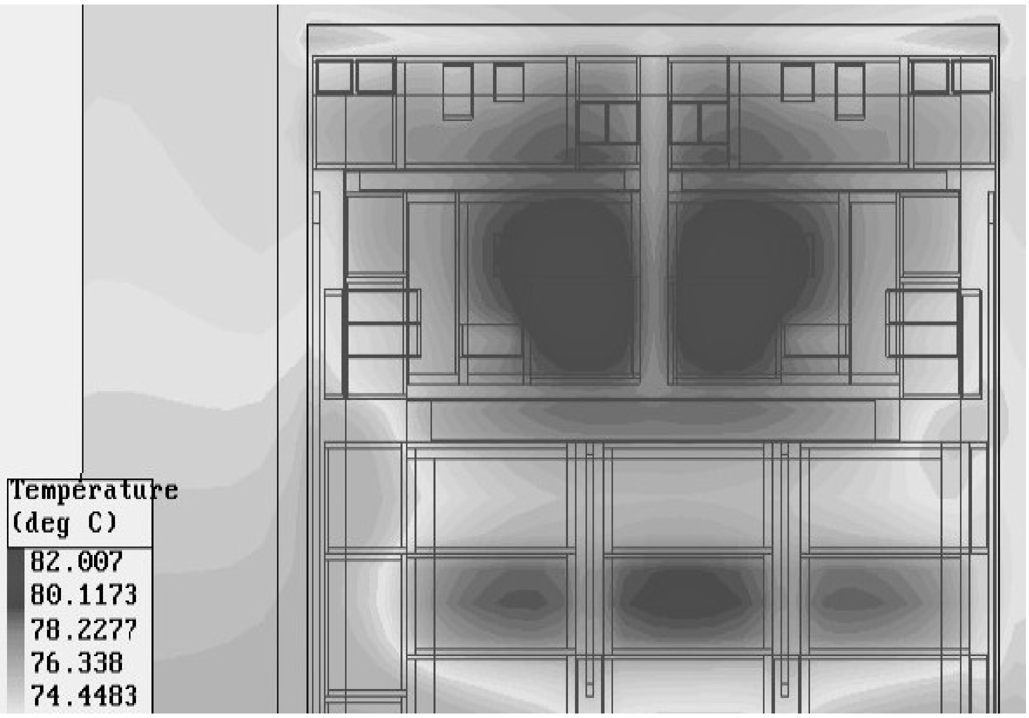


Figure 1.3 - CPU chip thermal profile and its "hotspots" and “coldspots” [94].

The high-performance chips designers are seeking innovative solutions to dissipate the heat out from chip areas. Real time dynamic thermal management (DTM) technique is providing control solution for thermal problem. The thermal problem solution lies in two areas: the design tools and the design engineers. The thermal constraints are the control theory limits. These thermal constrains include architectural decisions, the use of multiple operating modes, the physical implementation of the design, and the CPU cooling system characteristics. All control decisions are based on accurate thermal impact of the electrical analysis. The current DTM controllers are based on traditional promotional (P) controller or proportional integrator (PI) controller [6].

1.2 Thesis Objective

The objective of this thesis is to use fuzzy logic to design a DTM controller for the central processing unit (CPU). The DTM controller is managing Multi-Core CPU. The DTM controller selects each core operating frequency. The DTM controller reduces the total CPU degradation due to thermal limitations "the CPU thermal throttling". The thesis discusses the current DTM control techniques as well as introducing the new DTM thermal spare core (TSC) algorithm.

1.3 Literature Review

A wafer is a thin slice of semiconductor material, such as a silicon crystal, is used in the fabrication of integrated circuits and other micro devices. In CMOS (Complementary Metal-Oxide Semiconductor) technology, both N-type and P-type transistors are used to realize logic functions. Today, CMOS technology is the dominant semiconductor technology for microprocessors, memories and integrated circuits (ICs). The main advantage of CMOS over NMOS and bipolar technology is the smaller power dissipation. A semiconductor manufacturing technology represents the smaller feature of the chip. The CMOS Power density in high-performance processors continues to increase with technology generations as scaling of current, clock speed, and device density outpaces the downscaling of supply voltage and thermal ability of packages to dissipate heat. Power density is characterized by localized chip hot spots that reaches critical temperatures and cause thermal problems. The multicore architecture is introduced to increase the CPU performance without modifying its running frequency; this approach has more benefits like better performance, better power management and better cooling as the multicore processors running at a lower speed dissipates less heat than one single processor with the same chip area.

Thermal enhancements are divided into 4 main categories [7]:

First “packaging and cooling technique design” the cooling system consists of huge heat-sink, heat-pipes, and cooling fan.

Second “device research” the CMOS technology research invents low power devices.

Third “design-time thermal management” the design techniques improve the power and thermal characteristics of integrated circuits.

Fourth “run-time thermal management” the dynamic power management techniques are mandatory for today ICs.

The rest of the thesis focuses on run-time thermal management. The thesis discusses the current techniques improvement. In the year 2008, the heat-sink air cooling reached its maximum limits 198 Watt [1].

Thermal definitions:

A. The CPU thermal throttling

The CPU thermal throttling is CPU degradation due to CPU over heating. The CPU is forced to run with lower frequency. So the cooling system reduces the CPU temperature to low value [2].

B. Chip hotspots and coldspots

The temperature is not uniformly distributed over the chip area. There are some “hotspot” areas having higher temperature. The hotspot function block has higher power density than the “coldspot” function block [3].

DTM techniques:

A. Dynamic Voltage and Frequency Scaling (DVFS)

The dynamic voltage and frequency scaling is a DTM technique that changes the operating frequency of a core at run time [8], [9].

B. Clock Gating (CG)

Clock gating or stop-go [9] technique involves freezing all dynamic operations. CG turns off the clock signals to freeze progress until the thermal emergency is over. When dynamic operations are frozen, processor state including registers, branch predictor tables, and local caches are maintained. So less dynamic power consumed during the wait period. GC is more like suspend or sleep switch rather than an off-switch [8].

C. Threads Migration (TM)

Thread migration (also known as core hopping) is a real time OS based DTM technique. TM reduces the CPU temperature by migrating core tasks “threads” from an overheated core to another core with lower temperature [10], [11], [9].

The thermal sensors

There are 2 types of temperature sensors [1]:

A. Real temperature sensor

The real temperature sensor is a thermo-couple placed within core blocks. But there are fabrication limitations to place the sensors near the hotspots [8].

B. Virtual temperature sensors

The virtual temperature sensors is an OS based sensors. This “soft sensors” estimates the core temperature based on the number of instructions and power consumed per instruction [8]

The thermal controller:

A. Traditional Thermal Control

The current DTM controller uses proportional (P controller) or proportional-integral (PI controller) or proportional-integral-derivative (PID controller) to perform DVFS [8], [12], [13].

B. The Fuzzy Logic Control

The fuzzy logic is introduced by Lotfi A. Zadeh in 1965 [14]. The fuzzy control provides a convenient method for constructing nonlinear controllers via the use of heuristic information or mathematical modeling. Such heuristic information may come from an operator who has acted as a “human-in-the-loop” controller for a process. The fuzzy control design methodology is to write down a set of rules on how to control the process. Then incorporate these rules into a fuzzy controller that emulates the decision-making. Regardless of where the control knowledge comes from, the fuzzy control provides a user-friendly and high-performance control [15].

The traditional fuzzy set is two-dimensional (2D) with one dimension for the universe of discourse of the variable and the other for its membership degree. This 2D fuzzy logic controller (FC) is able to handle a non linear system without identification of the system transfer function. But this 2D fuzzy set is not able to handle a system with a spatially distributed parameter. While a three-dimensional (3D) fuzzy set consists of a traditional fuzzy set and an extra dimension for spatial information [16]. Different to the traditional 2D FC, the 3D FC uses multiple sensors to provide 3D fuzzy inputs. The 3D FC possesses the 3D information and fuses these inputs into “spatial membership function”. The 3D rules are the same as 2D Fuzzy rules. The number of rules is independent on the number of spatial sensors. The computation of this 3D FC is suitable for real world applications [16].

1.4 The Thesis Contributions

The thesis contributes to the problem of CPU thermal throttling by building a thermal model of a real Multi-Core CPU chip. The thermal model is mandatory for the simulation tests. The selection of the CPU chip is based on three factors: the published data regarding the real CPU chip floor plan; the real CPU chip power maps and the thermal profile of the real CPU chip. The published data regarding these three key factors are available only for “IBM POWER4 MCM” CPU chip containing 8 cores. Using reverse engineering, the chip floor plan is extracted from the published CPU floor plan pictures. The floor plan is validated using MATLAB. Then it is integrated to “hotspot 4.1” thermal simulator. The open loop response of the thermal model is extracted. Chapter 3 discusses in detail the reverse engineering process, integration to hotspot 4.1 simulator, and the open loop response curve extraction.

Different DTM controllers are designed based on fuzzy logic control design. The 2D fuzzy controller is used to improve the DTM response and decrease the thermal throttling effect. From the control point of view, the CPU cores are the heat sources. The correlations between these cores temperatures and operating frequencies improve DTM response. The 2D fuzzy is not able to handle these correlations as the number of rules increases as a function of the number of cores. Thus the 2D fuzzy is not feasible. 3D fuzzy controller is taking into consideration these correlations. The 3D fuzzy controller is also suitable for the real word application. The thesis introduces a new DTM controller evaluation index in order to compare between different DTM controllers responses. Also the thesis introduces a new DTM technique called “thermal spare core” algorithm (TSC).

1.5 The Thesis Structure

The rest of the thesis is organized as follows:

Chapter 2 presents the background of semiconductor thermal problem including the current used control techniques. This Chapter discusses the chip technology & integration, the dynamic thermal management techniques, the Multi-Core advantages over single core processors, the CPU parallel vs. serial operations, the core temperature measurements, the available simulators, and finally the chip floor planning.

Chapter 3 illustrates the building of CPU thermal model based on real Multi-Core processor chip floor plan and its cooling system in order to do realistic thermal simulations results. The suitable simulator is selected from the available chip thermal simulators. The real Multi-Core chip is integrated to the Hotspot 4.1 thermal simulator. Then the real chip open loop thermal profile curve is extracted.

Chapter 4 discusses a new DTM controller design approach. A mediation script is used for bypassing the data between the thermal model and the control system. The basic DTM control system consists of a P controller. The advanced DTM controller consists of a P controller followed by a fuzzy controller.

Chapter 5 gives simulations test results, evaluations, and analysis of all suggested DTM controller techniques. Finally, a brief summary of the presented work, conclusions, and major ideas for future research is given in Chapter 6.

Chapter 2

CPU Thermal Throttling Background

2.1 Introduction

This chapter discusses the semiconductor thermal problem historical background. The difference between the chip static and dynamic power consumptions. The dynamic power is the main factor in the dynamic thermal management. The effect of static power consumption can not be neglected any more [17]; as is the case for any CMOS digital circuits using low-threshold devices. The relative importance of this static power depends on the duty factor defined as the average percentage of time the gates are in transition. The CPU chip manufacturing is affected by the growth of both static and dynamic power consumptions. The traditional CPU air cooling system is unable to handle more than 198 W [4]. The semiconductor technology permits the designers to add more cores to CPU chip. The serial portion of the CPU instruction limits the speedup improvements. Thus the CPU cores are not all fully utilized and have different power consumption profiles. This leads to different temperature distribution across the CPU floor plan. The CPU chip floor plan design enhances its thermal profile.

2.2 Chip Technology & Integration

The Silicon-on-insulator (SOI) technology is the dominant silicon chips fabrication technology [18]. SOI wafers have a nanoscale thin surface film of single crystal silicon on an insulating film with a regular silicon wafer. By making this silicon film only as thick as necessary and insulating it from the bulk of the silicon wafer, circuits can run faster and cooler [19]. SOI technology can produce all industry-standard wafer sizes from 100 to 300 mm [19]. Using SOI instead of silicon results in higher speeds, suppressed cross-talk inside the chips, elimination of latch-up (an electrical fault), and increased radiation hardness of the chips [19]. Radiation hardening is a method of designing and testing electronic components and systems to make them resistant to damage or malfunctions caused by ionizing radiation such as particle radiation and high-energy electromagnetic radiation.

2.2.1Moore's law

Since the invention of the integrated circuit (IC), the number of transistors that can be placed on an integrated circuit has increased exponentially, doubling approximately every two years [20] as shown in Figure 2.1. The trend was first observed by Intel co-founder Gordon E. Moore in a 1965 paper. Moore's law has continued for almost half a century! It is not a coincidence that Moore was discussing the heat problem in 1965: "will it be possible to remove the heat generated by tens of thousands of components in a single silicon chip?" [20]. The static power consumption in the IC was neglected compared to the dynamic power for CMOS technology. The static power is now a design problem [21]. The millions of transistors in the CPU chip exhaust more heat than before. The CPU cooling system capacity limits the number of cores within the CPU chip [22].

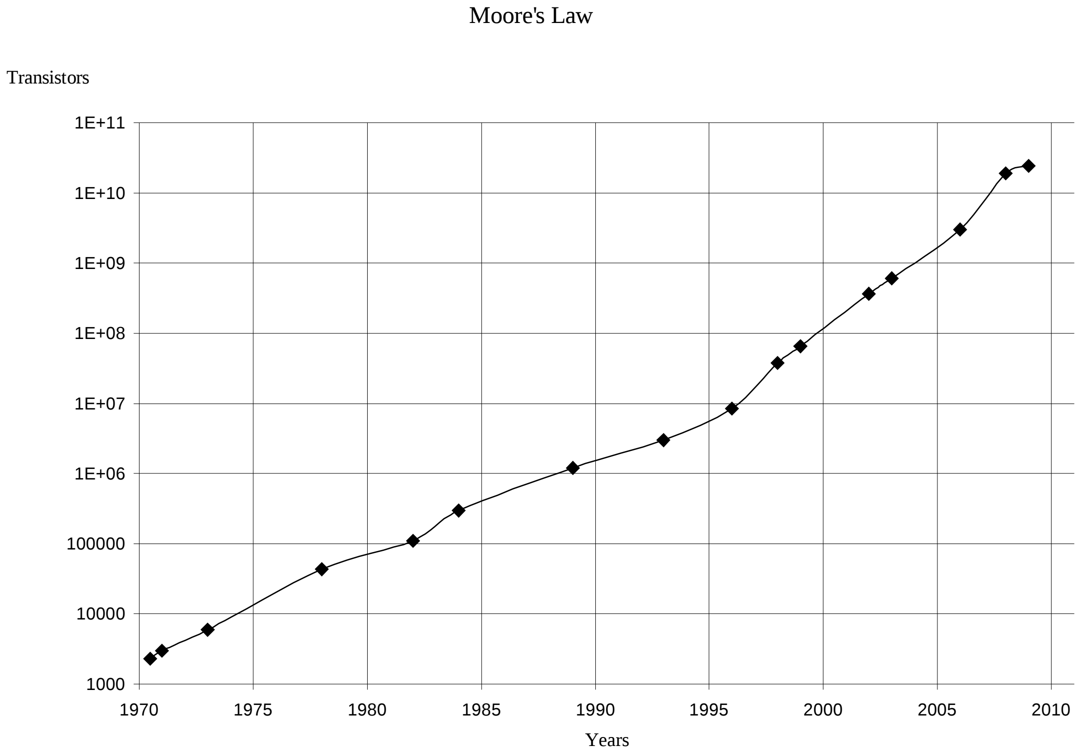


Figure 2.1 - "Moore's law" predicts the future of integrated circuit [96].

2.2.2 Air Cooling Limitation

The International Technology Roadmap for Semiconductors (ITRS) is a set of documents produced by a group of semiconductor industry experts. ITRS specifies the high-performance heat-sink air cooling maximum limits; which is 198 Watt [4]. The chip power consumption design is limited by cooling system level capacity. We already reached the air cooling limitation in 2008 as shown in Figure 2.2.

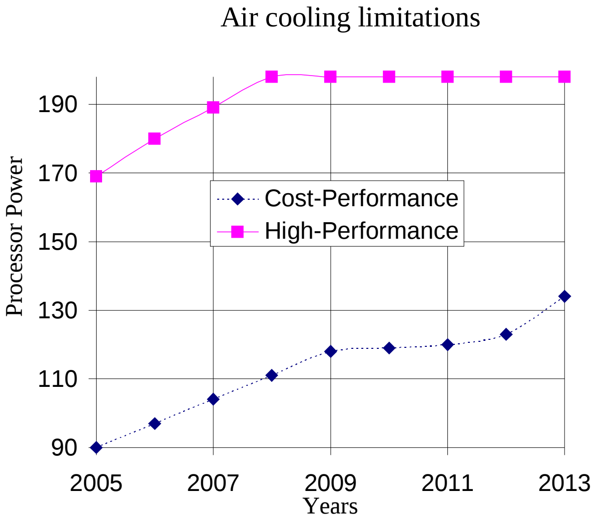


Figure 2.2 - Heat-sink air cooling limitations [1].

2.2.3 Cooling System Effect

Conventional air cooling heat-sink systems are limited by 198 W as shown in Figure 2.2. Thus exploring new cooling techniques such as liquid cooling, ionic wind technology has higher importance. Portable platforms pose a challenge as there is limited space for cooling systems. Moreover, the chips become faster the amount of heat that needs to be removed increases, this increases the challenge to cool down a laptop and its surfaces.

A- Ionic Air Cooling

The idea behind ionic cooling is to generate a constant flow of positively charged particles (ions) over the microchip. As shown in Figure 2.3, the traditional fans that provide a strong flow of air have problems in removing molecules close to CPU chip. The ion cooling system electrodes send a flow of ionized air over the surface of a silicon chip. An ionic air cooling system keeps the CPU chip 25 ºC cooler than traditional air cooling. The ionized air flow has the ability to remove heat from the molecules close to CPU chip better than the unionized air. This leads to improve heat exchange between the CPU chip and the cooling system [23].

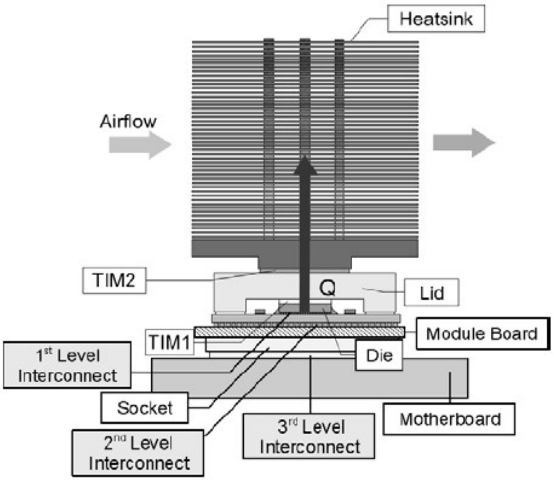


Figure 2.3 - Traditional chips cooling & interconnect [95].

B- Liquid Cooling

Water has a higher specific heat capacity as well as better thermal conductivity relative to that of air. By pumping cold water around a processor, it's possible to remove a large amount of heat in a short time. The main advantage of water cooling is its ability to cool down the CPU faster than the traditional air cooling [24]. However, water cooling is not suitable for portable devices.

2.2.4Semiconductor Power Wall

CMOS power dissipation increases due to power density increase. The power dissipation increased four times every three years until the early 1990's, due to a constant voltage scaling [22]. Recently, a constant field scaling is applied to reduce power dissipation; where the power density is increased proportional to the 0.7th power of scaling factor, resulting in power increase by twice every 6.5 years [22]. It is considered that the power dissipation of CMOS chips will steadily be increased as a natural result of device scaling. Technology scaling will become difficult due to the power wall. On the other hand, future computer and communications technology will require further reduction in power dissipation. Since no new energy efficient device technology is on the horizon, low power CMOS design should be challenged [22].

A - Power, Energy, and Energy Delay

The interest in lowering the power of a processor has grown. The processors' power, energy and performance are correlated [25]. The chips use a constant DC voltage supply; so the power is directly proportional to the current. The total energy drawn from the source to fulfill a function is the running cost the user pays to get that function done. A lower energy is better since it means that the user is paying less. However, the one having less delay may be preferred since it allows the user to start other functions earlier. In some applications, a user may even accept a slightly higher energy to achieve a much lower time delay. In that case, the designer is weighing the cost of the energy and the cost of delay [25]. The presented DTM techniques decrease the processors average energy consumption at runtime depending on the applications and the limit of the supply voltage V_{dd} [17]. As shown in Figure 2.4, different amount of energy consumed to perform the same task, thus the amount of power is varying. The required task is performed before the dead line time in all cases. From power point of view a > b > c while from energy point of view b > a > c

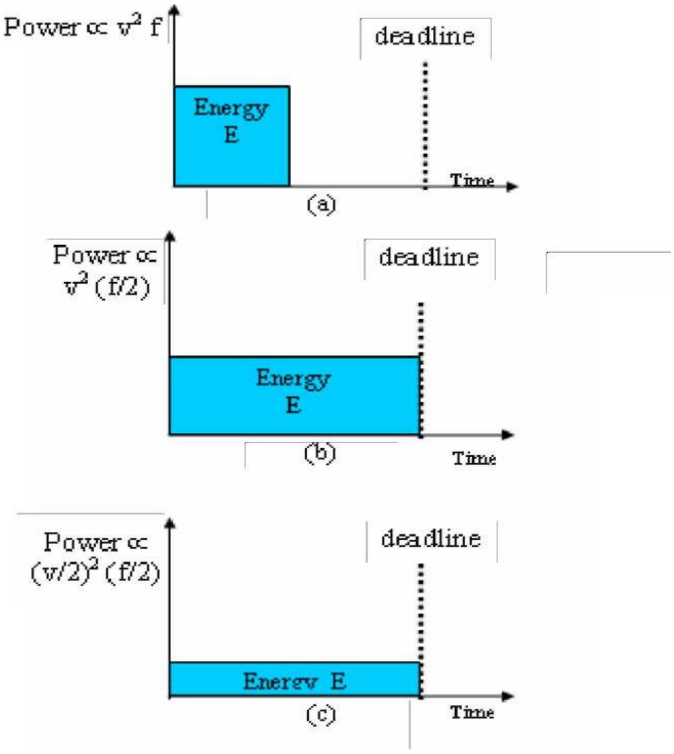


Figure 2.4 - the CPU energy consumption vs. power consumption [17].

The total CMOS power (2.1) is the summation of the dynamic power, the static power and the short-circuit power

$$P_{CMOS} = P_{Dynamic} + P_{Static} + P_{Shortcircuit} \qquad (2.1)$$

B -The Dynamic Power

The CMOS dynamic power are the switching power, and the short-circuit power. The dynamic power equation as a function of swing voltage is:

$$P_{Dynamic} = \alpha f \, C \, V_{dd}^{\ 2} \qquad (2.2)$$

Where $P_{Dynamic}$ is the dynamic power depending on four parameters, supply voltage (V_{dd}), clock frequency (f), physical capacitance (C) and an activity factor (α) that relates to how many 0→1 or 1→0 transitions occur in a chip [26].

$P_{Shortcircuit}$ the short-circuit power (on/off power). When the CMOS inverter input is around $\frac{V_{dd}}{2}$ during the turn-on and turn-off switching transients; both the PFET and the NFET are on. Thus a short circuit current I_{SC} flows from V_{dd} to ground producing short-circuit power [27].

$$P_{Shortcircuit} = \frac{\beta}{12} (V_{dd} - 2V_T)^3 f \, \tau \qquad (2.3)$$

Where τ the CMOS gate delay, V_T the threshold voltage and β the gain factor $\mu A / V^2$ of an MOS Transistor [26]

C - The Static Power

The chip static power still depends on temperature. Off-state leakage is static power consumption. As shown in Figure 2.5, the static current leaks through transistors even when they are turned off [29]. Until recently, only dynamic power was the significant source of power consumption. But the CPU static power consumption increases by adding more transistors to the chip. The Static power is today a design problem. The static power reaches 40% of the total power in 2005 [30]. In 2009, the chip power consumption exceeds the power efficiency requirement as per ITRS updates [82].

$$P_{Static} = V_{dd}\,\eta\,\kappa\,I_{leak} \qquad (2.4)$$

Where V_{dd} the supply voltage, η the number of transistors in the design, κ design the characteristic of an average device, and I_{leak} a technology parameter describing the per-device sub threshold leakage [Butts and Sohi 2000] [30].

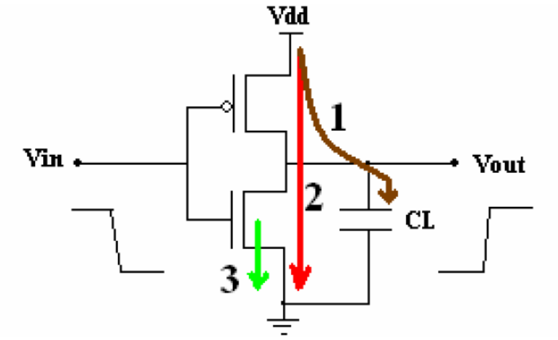


Figure 2.5 - CMOS power dissipation $P_{Dynamic}$ 1. P_{Static} 2. $P_{Shortcircuit}$ 3 [17].

2.2.5 The Chip Area & Power

Low power and low-voltage designs are mandatory with the static power growth. The chip power is proportional to the chip area [28]. As per Pollack's rule [31], the processor performance is proportional to the square root of the chip area [32].

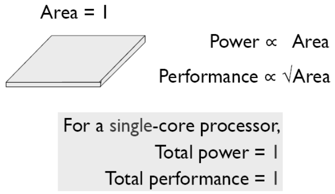
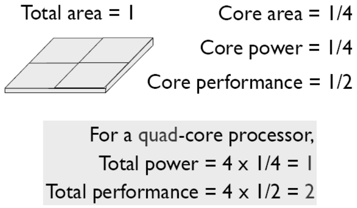


Figure 2.6 - power and area of single core processor [28].



Total area = 1
Core area = 1/4
Core power = 1/4
Core performance = 1/2
For a quad-core processor,
Total power = 4 x 1/4 = 1
Total performance = 4 x 1/2 = 2

Figure 2.7 - power and area of Multi-Core processor [28].

The advantages of the Multi-Core CPU design are shown in Figure 2.6 and Figure 2.7. The distribution of the CPU area into multiple cores instead of one single big core processor leads to higher CPU performance and constant power consumption [28].

2.2.6 The Real CPU Chips

A - AMD CPU chips

Advanced Micro Devices (AMD) founded in 1969 is one of the leading processor manufacturing companies. AMD contributes in Multi-Core computer architecture. As shown in Figure 2.8; AMD processors evolution is starting from dual cores up to 6 cores processors. AMD processor' support different operating frequencies and different thermal design power (TDP) (sometimes called thermal design point). The TDP represents the maximum amount of power the cooling system requires to dissipate. The TDP is not the maximum power that the processor can dissipate.

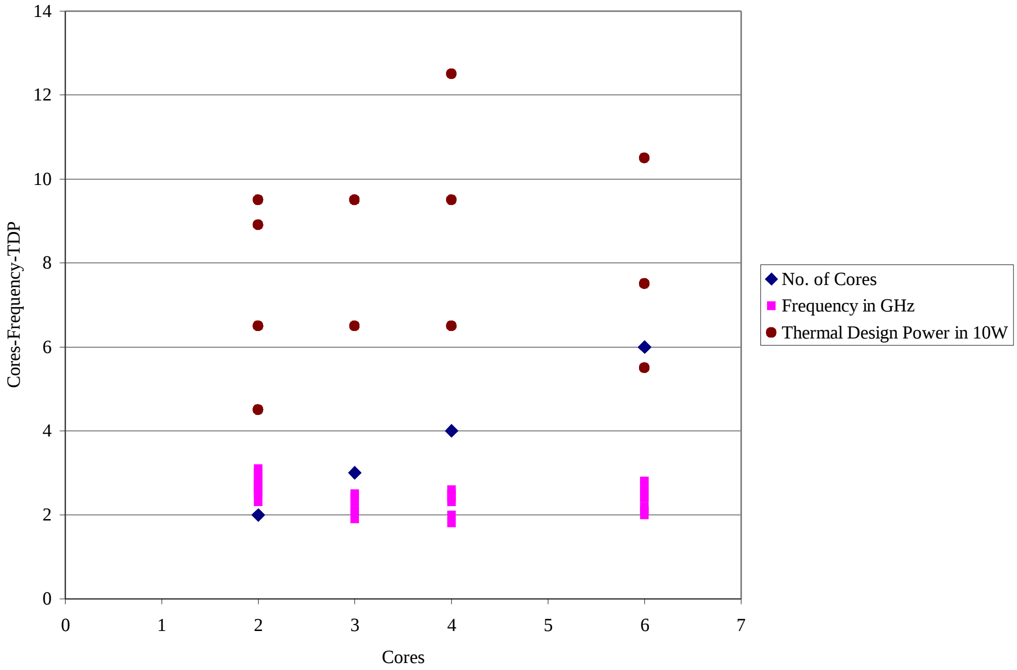


Figure 2.8 - AMD processor evolutions [33].

B - Intel CPU chips

Integrated Electronics Corporation (Intel) founded in 1968 is the world's largest semiconductor chip maker, based on revenue [33]. Intel is the inventor of the x86 series of microprocessors, the processors found in most personal computers. Intel contributes in Multi-Core computer architecture; as shown in Figure 2.9. The Intel processor evolution starts from dual cores up to quad cores processors. Figure 2.10 shows the number of transistors per processor chip and die size evolution.

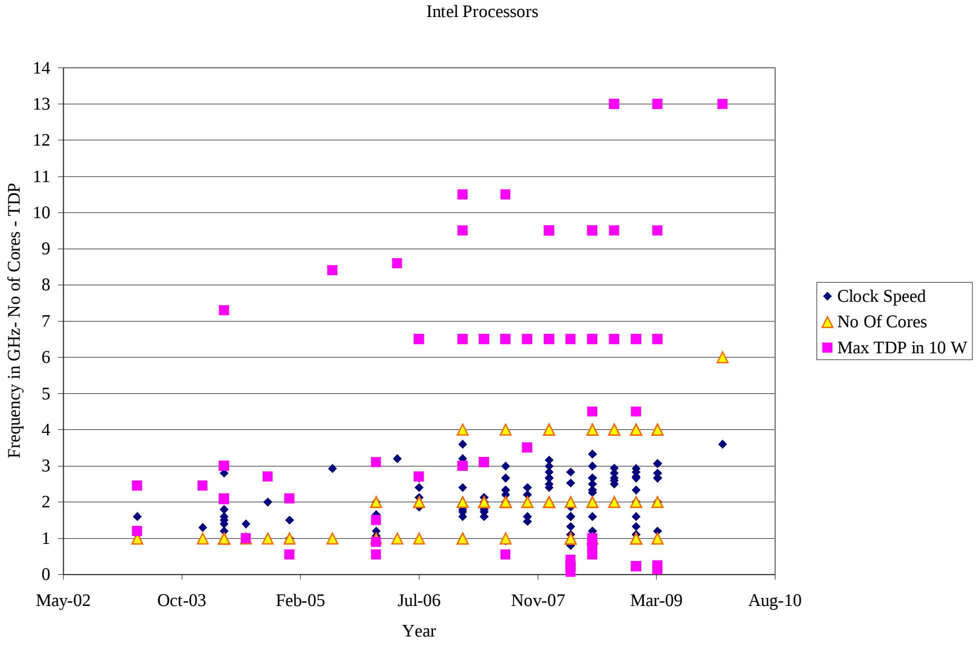


Figure 2.9 - Intel CPU chip evolutions [96].

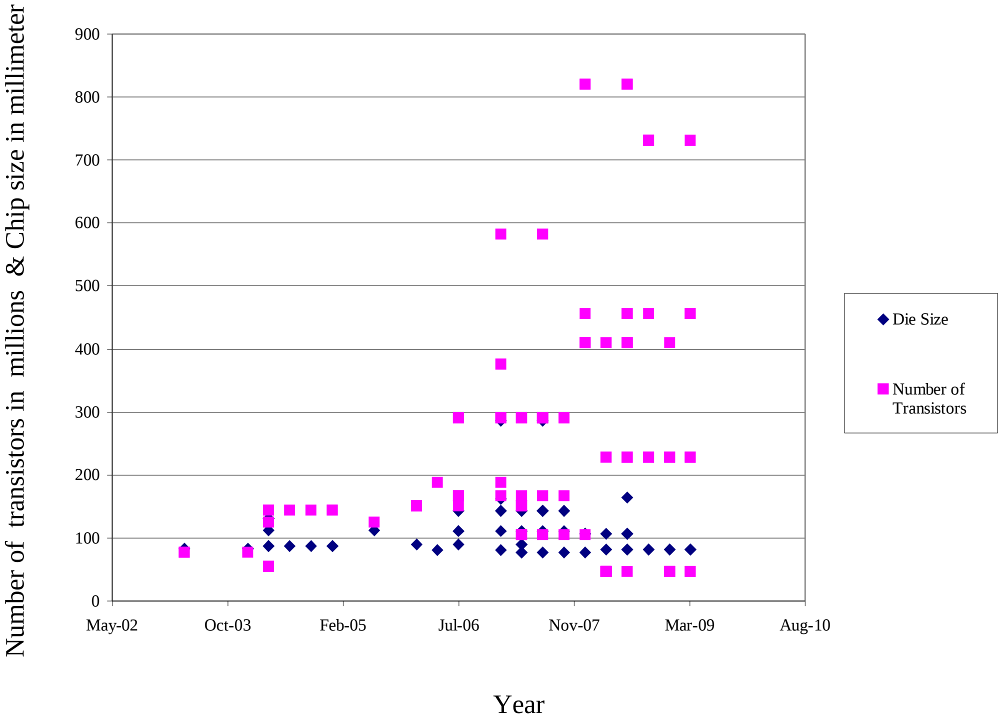


Figure 2.10 - Intel CPU chip size and the number of transistor [96].

C - Nvidia

The graphics processing unit (GPU) computing model is to use the CPU and GPU together in a heterogeneous computing model. The sequential part of the application runs on the CPU and the computationally-intensive part runs on the GPU. From the user’s perspective, the application just runs faster because it is using the high-performance of the GPU to boost graphical performance [34]. The application developer has to modify their application to take the compute-intensive kernels and map them to the GPU. The rest of the application remains on the CPU. The GPU computing is enabled by parallel architecture of Nvidia’s GPUs called “CUDA” architecture. The CUDA architecture consists of 100 processor cores operating together to crunch through the data set in the application [34].

D - SUN CPU chips

SUN Microsystems produces computer servers and workstations based on its own SPARC processors. In the early 1990’s SUN extended its product line to include large-scale symmetric Multi-processing RISC architecture servers. SPARC server 600MP is 4-processor. SPARC server 1000 is 8-processor. SPARC center 2000 is 20-processor [35]. SUN releases the SPARC enterprise server products, as join design by SUN and Fujitsu based on Fujitsu SPARC64 VI processor. SUN UltraSPARC IV+ processor operating frequency is 1.95 GHz - 2.1 GHz. This processor is based on 90nm process technology [35]. In November 2005 SUN launched the Ultra SPARC T1; the first microprocessor supports both Multi-Core and Multi-Thread. This processor has the ability to concurrently run 32 threads of execution on 8 cores processor. In October 2007, SUN release the UltraSPARC T2 microprocessor.

T2 is also Multi-Core and Multi-Thread CPU. T2 extends the number of threads per core from 4 to 8. In April 2008, SUN released servers with UltraSPARC T2 Plus, which is a symmetric multi-processing (SMP) capable version of UltraSPARC T2. SUN UltraSPARC T series is up to 8 cores at 900 MHz - 1.4GHz based on 45nm technology [35]. SUN has alliance with AMD to produce x86/x64 servers based on AMD's Opteron processor. SUN fire x64 family servers are designed to address heat and power consumption issues commonly faced in data centers. SUN has alliance with Intel to produce SUN blade x6250 server based on Intel Xeon processor [35].

E - IBM CPU chips

International Business Machines Corporation (IBM) founded on 1896 as the tabulating machine company. IBM invents the following microprocessors: In 1980 IBM invents pioneering prototype RISC processor; then IBM ROMP RISC processor “032 processor”. IBM POWER processor family is based on RISC architecture (POWER1 - POWER2 - POWER3 - POWER4 - POWER4+ - POWER5 - POWER5+ - POWER6 - POWER7 – POWER8) [36]. POWER4 is the first Dual-Core processor RISC architecture. Its operating frequency starts from 1.1GHz - 1.3 GHz. Its chip has up to 174 million transistors, 115 W TDP based on 90nm technology [37]. POWER5 is a Dual-Core processor support for simultaneous multi-threading (SMT). POWER5 operating frequency starts from 1.5 GHz to 2.2 GHz. Its chips area is 389 mm² and contains up to 276 million transistors [37]. POWER6 is a Dual-Core processor. Its operating frequency starts from 3.5 GHz to 5.0 GHz. Its chips area is 341 mm² and contains up to 790 million transistors based on 65nm technology [38]. POWER7 has have up to 8 cores Its chips area is 567 mm² and contains up to 1.2 billion transistors based on 45nm technology [39]. POWER8 is in research and development phase.

F - PicoChip

PicoChip is English founded in 2000. PicoChip developed a multi-core digital signal processor, the picoArray. This integrates 250-300 individual DSP cores onto a single die. These small cores are a 16-bit processor each. The company has four products currently available (PC102 and PC202 / 203 / 205) [38].

2.3 The Dynamic Thermal Management

2.3.1 The Importance of DTM

Smaller transistor fabrication improves the efficiency, time, costs, design migration, and capacity of the ICs production. The technology node refers to the average half-pitch of a memory cell which is measured in nm. At today's 65nm and 45nm technologies nodes the CPU faces thermal air cooling limitations. The 32nm technology permits the addition of more cores to the CPU chip. Adding more cores to a CPU chip increases the power density. This leads to thermal throttling due to cooling limitations. Due to air cooling limitations, there are many thermal constrains adding more cores to CPU chip. The DTM techniques are required in order to have maximum CPU resources utilization. Also for portable devices the DTM doesn’t only avoid thermal throttling but also preserves the battery consumption. As shown in Figure 2.11, the basic DTM controller is a closed loop feedback controller. The DTM controller measure the CPU cores temperatures and according selects the speed “operating frequency” of each core [28]. The power consumed is a function of operating frequency and temperature. The change in temperature is a function of temperature and the dissipated power.

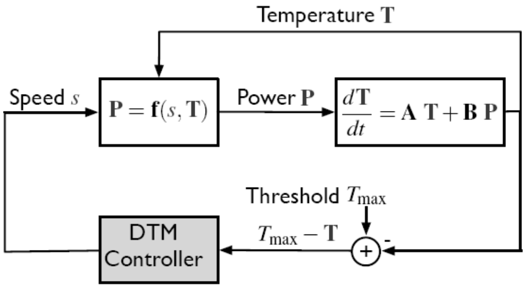

Temperature $\mathbf{T}$
Speed s
$\mathbf{P} = \mathbf{f}(s, \mathbf{T})$
Power $\mathbf{P}$
$\frac{d\mathbf{T}}{dt} = \mathbf{A}\ \mathbf{T} + \mathbf{B}\ \mathbf{P}$
Threshold $T_{\max}$
$T_{\max} - \mathbf{T}$
+
-
DTM Controller

Figure 2.11 Basic DTM controller block diagram [28].

2.3.2 Thread Migration

Thread Migration (TM) is a DTM technique based on power deregulation compensate by leveraging co-operative hardware–software management [10]. Multi-Thread uses the software parallelism technique. The current mobile systems, energy source parameters (including voltage levels) change relatively slowly with respect to processor utilization level (operating frequency), allowing operating systems and processors to adapt to evolving battery behavior [10]. In today's Multi-Core designs provide an effective way of overcoming instruction-level parallelism (ILP) limitations by exploiting thread-level parallelism (TLP). TM requires a mechanism to transfer architectural state from one core to another [9].

2.3.3 On/Off Control "Clock Gating"

Clock gating or stop-go technique is a basic form of dynamic voltage frequency scaling (DVFS) [9]. This involves freezing all dynamic operations and turning off clock signals to freeze the processor progress. When dynamic operations are frozen, processor state including registers, branch predictor tables, and local caches are maintained. So less dynamic power is wasted during the wait period. Thus stop-go is more like suspend or sleep switch rather than an off-switch [8].

2.3.4 Dynamic Voltage and Frequency Scaling

The dynamic voltage frequency scaling (DVFS) is a DTM technique that reduces power and energy consumption of microprocessors. The relation between $P_{Dynamic}$, f and V_{dd} is as given in equation (2.2). Lowering only the operating frequency f can reduce the power consumption. But the energy consumption remains the same because the computation needs more time to finish. Lowering the supply voltage reduces a significant amount of energy [17].

Dynamic tuning the supply voltage V_{dd} and the operating frequency f reduces both power and energy consumption. Figure 2.12 shows the power saving achievable by using variable V_{dd} and f. Using variable supply voltage V_{dd} when the clock frequency f is reduced by half, this lowers the processor's power consumption and still allows the task to complete by deadline. But the energy consumption remains the same. Reducing the voltage level V_{dd} by half reduces the power level further without any corresponding increase in execution time. As a result the energy consumption is reduced significantly, but the appropriate performance could remain the same as shown Figure 2.2 [17]. But the voltage V_{dd} level reduction leads to current flow reduction. This means slower switching in transistor level. Thus voltage V_{dd} level reduction leads to slower microprocessor performance depending on the gate delay.

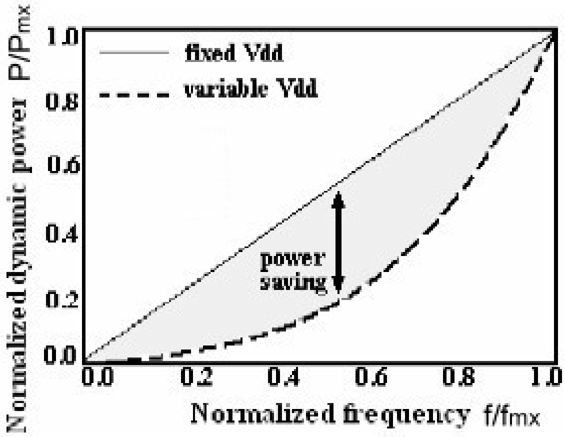


Figure 2.12 - Power saving achievable using variable voltage [17].

The DVFS policy involves more of a continuous adaptive scheme. Power consumption reduction is done by enabling a continuous range of frequency and voltage combinations. Thus DVFS policy is not as simple as the stop-go mechanism, but depends on a control design. A set point slightly is used below the thermal threshold. P controller or PI controller is used to adapt the frequency and voltage levels to towards this target threshold [8].

2.3.5 Multi-Supply Multi-Voltage (MSMV)

In the case of MSMV, non performance critical functional blocks are run at a lower voltage and/or frequency to conserve power. As a result, they run cooler than surrounding functions which are running at higher voltages and/or frequencies [6].

2.3.6 Power Shut-Off (PSO)

The chip function blocks can be completely shut down to conserve power when they’re inactive. Obviously, the thermal characteristics for such blocks depend on their state. Given an increase in the number of low-power designs featuring multiple operating modes, it’s necessary to calculate and account for these mode-specific thermal profiles. In the context of its thermal impact, the PSO is the most extreme low-power technique [6].

2.3.7 Substrate Biasing

CMOS substrate biasing for threshold voltage control mechanism is based on the fact that a semiconductor device includes a PMOS transistor and an NMOS transistor. The back gate of the NMOS device is biased at 0 volts for the maximum V_{th} and is biased at +1 threshold for the minimum V_{th} of the device. Only the back gate of the PMOS device is biased at V for the maximum V_{th} of the device and is biased at 1 V_{th} below V for the minimum V_{th} of the device.

Therefore, the voltage between the source and substrate of the PMOS and NMOS transistors becomes 0 volts. By driving the back gates in opposite direction and in phase with the input to the receiver circuit, the threshold voltage of the receiver is moved away from ground (GND) when the input is at a logical "0" and way from V when the input is at a logical "1". Thus almost no leakage current flows between the source and substrate [40].

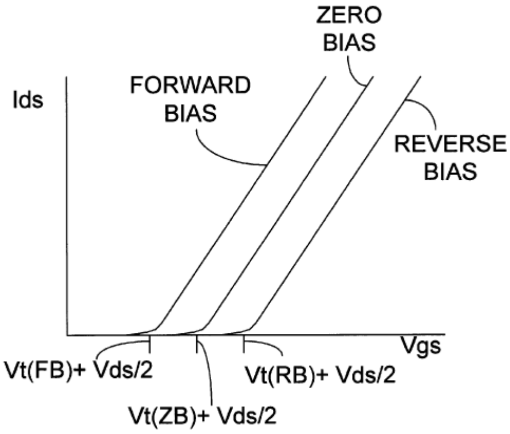


Figure 2.13 - BIAS Voltages [40]

2.3.8 Multi-Threshold CMOS “Multi-Vt”

Multi-Threshold CMOS (MTCMOS) uses transistors with multiple threshold voltages (V_{th}) to optimize delay or power. Lower threshold voltage devices are used on critical delay paths to minimize clock periods. Higher threshold voltage devices are used on non-critical paths to reduce static leakage power without incurring a delay penalty. Typical high threshold voltage devices reduce static leakage by 10 times compared with low voltage devices. One method of creating devices with multiple threshold voltages is to apply different bias voltages (V_b) to the based or bulk terminal of transistors, the other way can be "gate engineering". In MOSFET devices, lower bias voltages increases delay, and reduce static leakage thus the temperature is reduced [41], [42].

2.4 The Core Temperature

The CPU temperature is not uniformly distributed over the chip surface. There is variation from coldspots and hotspots depending on the functionality and power density of each block within chip.

2.4.1Temperature Distributions and Hotspots

The temperature distribution across chip surface is pure design issue. The power density levels across a chip are varying. Units like registers have small area, but are accessed frequently. Thus generates local hotspots across the chip [43]. The local hotspots heat transfer modes (conduction, radiation and convection) are the design key factors [44]. Local hotspots effect is reduced by spreading the high power density units over the chip. Small hotspots can be mitigated if they are split and spread among other cold units [43].

2.4.2 Ambient Temperature

The air density, viscosity, and heat capacity change with temperature. Thus the CPU temperature changes with ambient temperature [45]. The reliability of a device degrades if the device operates above the recommended operational temperature. The CPU lifetime is reduced if it operates continually at maximum operating temperature.

2.4.3 Chip Floor Plan

The chip floor plan is pure design issue. The floor plan has an important effect on the CPU behavior. The cores distribution across a die change power density distribution. This creates a multi-dimensional tradeoff space among core power, floor plan, magnitude of cross-chip variation, and cooling cost [46]. The geometric relationship among function blocks affects the floor planning design process. The Multi-Core distributed floor plan layout is better than Multi-Core dense floor plan layout from thermal point of view [47].

2.4.4 Low Power VLSI

Low power design emerges principal theme in today’s Electronics industry. The primary driver for low power VLSI is the remarkable success and growth of the personal computing devices (portable desktops, audio-video-based multimedia products) and wireless communications systems (Mobiles handsets, personal digital assistants and personal communicators). These portable devices demand high-speed computation and complex functionality with low power consumption. In the past, the major concerns of the VLSI designer are area, performance, cost and reliability. The power dissipation is now important as performance and area [48]. The MTCMOS (some time known as VTCMOS) technology provides low power circuit designs [21] and [49].

2.5 The CPU speed up

This section discusses the CPU speed up limitations and the different CPU architect.

2.5.1 Multi-Core and Multi-Processing Designs

Computing symmetric multi-processing (SMP) involves multi-processor computer architecture where two or more identical processors can connect to a single shared main memory. In the case of Multi-Core processors, the SMP architecture applies to the cores, treating them as separate processors. SMP systems allow any processor to work on any task no matter where the data for that task are located in memory. With proper operating system support, SMP systems easily move tasks between processors to balance the workload efficiently [50]. Asymmetric multi-processors and asymmetric multi-processing (ASMP) were pioneered in 1970 by the Massachusetts institute of technology (MIT) and Digital Equipment Corporation (DEC). ASMP allows applications to run specific subtasks on processors separate from the "master" processor. ASMP computers are comprised of multiple physical processors that are unique, and thus not symmetrical. These processors are defined as either master or slave: master processors are more capable than slaves and are given full control over what the slave processors do. Intel started work on SMP in 1981 which marked the demise of ASMP in the consumer and corporate market [51]. The CPU processor chip manufacturing and desgin having at least three camps in the computer architects community [52]: The Multi-Core CPU: the CPU chip contains small number of "big cores" (Intel, AMD, IBM, etc). The Many-Core CPU: the CPU chip contains a larage number of "small cores" (SUN, NVIDIA). The asymmetric-Core CPU: the CPU chip contains a combination of small number of large cores together with large number of small cores (IBM cell architecture).

2.5.2 Amdahl’s law

Gene Amdahl’s law: “parallel speedups limited by serial portions” [53]. It is often used in parallel computing to predict the theoretical maximum speedup using multiple processors. As long as enough parallelism exists; it is always more efficient to double the number of cores rather than the frequency in order to achieve the same performance. Which means that: adding more cores to CPU will speedups under the serial portion limits.

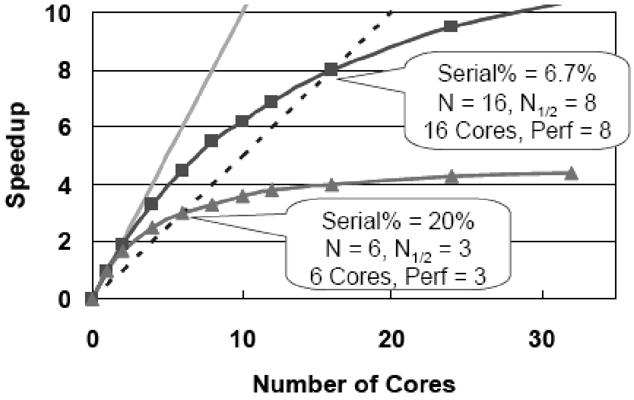


Figure 2.14 - Amdahl's Law limits parallel speedup [31].

Although it is true that the many symmetric core system delivers higher compute throughput than the multi symmetric core system for the same die size and in the same power envelope, it may be difficult to harvest the performance. If the serial percentage in a program is large, then parallel speedup saturates with small number of cores. Figure 2.14 [31] illustrates impact of serial percentage of code on parallel speedup.

2.5.3 Gustafson's Law

Gustafson's law “any sufficiently large problem can be efficiently parallelized” (also known as Gustafson-Barsis' law) [83]. This law states that, with increasing data size, the speedup obtained through parallelization increases, because the parallel work increases with data size. Gustafson’s law is closely related to Amdahl's law, which gives a limit to the degree to which a program can be speed up due to parallelization. Amdahl's law is related to fixed computation load while Gustafson's law removes fixed computation load. Instead, Gustafson proposed a fixed time concept which leads to scaled speed up for larger data sizes.

2.6 Conclusion

The Moore’s Law continues with technology scaling, improving transistor performance to increase frequency, increasing transistor integration capacity to realize complex architectures, and reducing energy consumed per logic operation to keep power dissipation within limit. The technology provides integration capacity of billions of transistors; however, with several fundamental barriers. The power consumption, the energy level, energy delay, power density, and floor planning are design challenges. The Multi-Core CPU design increases the CPU performance and maintains the power dissipation level for the same chip area. The CPU cores are not fully utilized if parallelism doesn't exist. Low cost portable cooling techniques exploration has more importance everyday as air cooling reaches its limits “198 Watt”. The next Chapter presents the available chip simulator techniques and how to integrate a Multi-Core chip to a thermal simulator.

Chapter 3

Building the Multi-Core CPU Model

3.1 Introduction

This Chapter presents the available semiconductor simulators and simulation techniques. There are many academic and commercial simulators. There are different simulator types. Some semiconductors simulator focus on the chip thermal profile. Other simulators focus on the performance estimation. Some simulators focus on the power consumption estimation. The chip floor designer and floor planning techniques are also presented. The chapter presents some chip fabrication cost estimation techniques. Also it disuses the real Multi-Core CPU chip reverse engineering steps. How to build up its thermal model based on 45nm technology. The CPU chip integration steps to the thermal simulator are presented. A real CPU chip open loop thermal profile is extracted using the thermal simulator.

3.2 Semiconductor Chip Simulators

This section presents the available software simulators and design tools.

3.2.1 Thermal Simulators

Thermal simulations play an important role in the design of many engineering applications. Chip thermal analysis is used to calculate the cooling system capacity and the estimated chip life times. Thermal simulators extracts the CPU internal blocks temperatures, and the chip thermal profile.

A - ANSYS

ANSYS is a thermal analysis simulator. It calculates the temperature distribution and the thermal quantities [54]. Typical thermal quantities are: the temperature distributions, the amount of heat lost or gained thermal gradients, and thermal fluxes. The basis for thermal analysis in ANSYS is a heat balance equation obtained from the principle of conservation of energy. The ANSYS program handles all three primary modes of heat transfer: conduction, convection, and radiation. ANSYS supports steady-state and transient thermal analysis [54].

B - Flotherm

Flotherm is 3D simulation software for thermal design of electronic components and systems [55]. It enables engineers to create virtual models of electronic equipment, perform thermal analysis. The Flotherm program uses computational fluid dynamics (CFD) techniques to predict airflow, temperature and heat transfer in components, ICs, boards and complete systems [55].

C - Thermal Test Vehicle

Thermal test vehicle (TTV) is a prototype used for component qualification and test system products for unreleased processors [56]. TTV is available earlier than real processor, similar package construction to real processor. Power may be varied to simulate entire real processor. TTV is the best choice for early component and system testing in characterizing heat-sink performance Figure 3.1

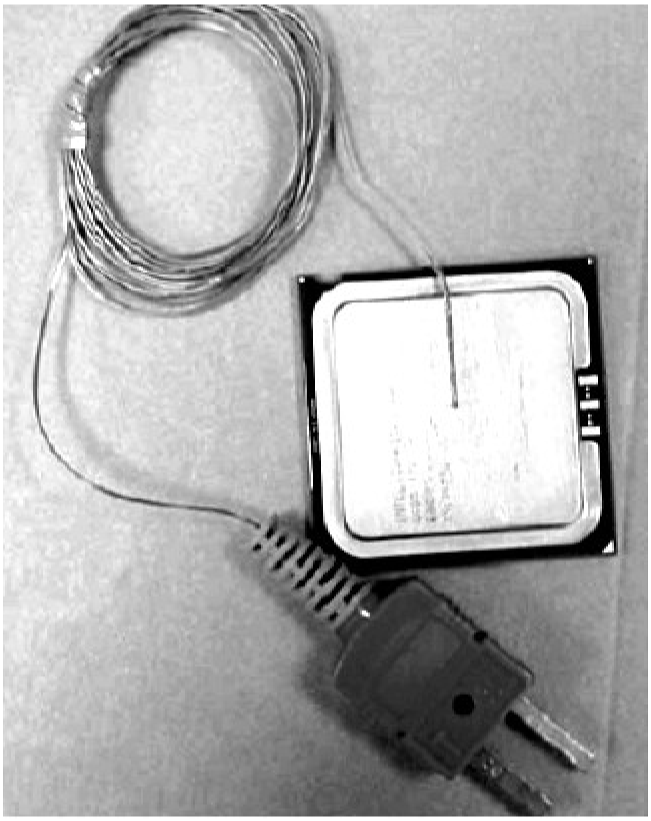


Figure 3.1 thermal test vehicle [56].

D - Virginia Hotspot Simulator

Hotspot is an accurate and fast thermal model suitable for use in architectural studies. It is based on an equivalent circuit of thermal resistances and capacitances. The equivalent thermal RC circuit corresponds to micro architecture blocks and essential aspects of the thermal package. Hotspot takes power numbers as input, and returns temperature numbers Hotspot has a simple set of interfaces and hence can be integrated with most power-performance simulators. The advantage of Hotspot is its compatibility with the power and performance models. Hotspot makes it possible to study thermal evolution over long periods of real applications [57].

E - ATMI Simulator

ATMI is an analytical model of temperature in microprocessors. ATMI takes power numbers as input, and returns temperature numbers. The user must provide power numbers. ATMI does not model a particular packaging and heat-sink [58], [59]. The main advantages of ATMI are its computation speed and ease of use. To obtain these advantages, it requires detailed inputs regarding the physical model. The accurate temperature modeling requires solving a boundary-value problem. First the physical system is modeled (object geometry, material properties, boundary conditions,). Then the heat equation is solved with mathematical means [59].

3.2.2 Power Simulators

A - Virginia Hotleakage Simulator

Hotleakage is a tool for modeling semiconductor leakage current. It is built upon Wattch power performance simulator. Hotleakage accommodates with other technology models. It models leakage in a variety of structures. Temperature effects are important, because leakage current depends exponentially on temperature. Hotleakage includes the un-modeled effects of supply voltage, gate leakage, and parameter variations [60]. Hotleakage has circuit-level accuracy because the parameters are derived from transistor-level simulation. Its simplicity is maintained by deriving the necessary circuit-level model for individual cells, like memory cells or decoder circuits.

3.2.3 Performance Simulators

A - Simple Scalar Simulator

“Simple scalar” toolset provides an infrastructure for simulation and architectural modeling. This simulator supports Multi-Core parallelization methodology. Simple Scalar uses an execution-driven simulation technique that reproduces a device’s internal operation. Execution-driven simulation provides access to all data produced and consumed during program execution. These values are crucial to the study of optimizations such as value prediction, compressed memory systems, and dynamic power analysis [61].

B - PTscalar

“PTscalar” is a micro architecture-level performance and power simulator for the pipeline structure “SuperScalar architectures”. PTscalar reads the user specified power parameters and system configuration, then computes performance and power statistics for each functional unit every clock cycle. PTscalar can be used to evaluate software and compiler optimization, micro architecture innovation, and software and hardware co-design/tradeoff for performance and power optimization. PTsclar includes the temperature dependent leakage power model and built-in thermal model. PTscalar is capable of thermal and transient-current estimations. PTscalar takes into consideration the dependence between leakage power and temperature [62].

C - WATTCH Simulator

"WATTCH" is an architectural simulator that estimates CPU power and performance tradeoff. The power estimates are based on a suite of parametrical power models for different hardware structures [63]. A modified version of simple scalar is used to WATTCH collect results. This simple scalar provides a simulation environment for modern out-of-order processors with 5-stage pipelines: fetch, decode issue, write back and commit. Speculative execution is also supported. WATTCH provides power oriented modifications that track each processor unit per cycle. WATTCH compute the power values associated with these units accordingly [63].

D - CMP-sim simulator

"CMP-sim" is a Multi-Core architectural instruction execution simulation. CMP-sim extends the simple scalar toolset with models of the pipeline structures [64], [65].

3.2.4 Chip Floor Planning

A - Quilt simulator

Quilt stands for quick utility for integrated circuit layout and temperature modeling. Quilt is a graphical tool that permits users to build floor plans of integrated circuits. Quilt provides both a visual aid as well as an input to the Hotspot simulator. The tool provides numerous features for estimating circuit performance, such as interconnect delay. Quilt generates graphical images for publications [66]. It runs on a variety of computing platforms. Quilt enables users to make changes to IC layout quickly, evaluate and analyze the results of their modifications. Quilt addresses temperature and interconnects issues [67].

3.2.5 Processor Hardware Cost Estimation

The objective of the cost model is to better plan for IC production. IC cost estimation accuracy depends on the detailed information such as layout, fabrication cost, and design cost [68]. Cost estimating tools include a microcircuit parametric estimating model. The parametric cost estimating process is not only estimates product cost but also provides an effective program cost management [69]. The processor cost estimation depends on its hardware resources characteristics “area, energy, and delay” [70]. Each type of hardware resource, i.e., function units, register files, and interconnections, are characterized by a specific set of desired properties. These properties are cost dependant. Thus the processor is breakdown into desired functionalities that reflect cost. The cost, functionality tradeoff is calculated.

3.3 Selected CPU chip

3.3.1 IBM CPU chips

The CPU chip selection is based on the on the amount of published information. Thus IBM POWER processor family is selected. As shown in Table 3.1 IBM POWER family chips published information include floor plan, thermal design power (TDP), technology, chip area, and operating frequencies. The detailed floor plan, power density maps, and thermal profile information are published only for POWER4. POWER4 is the first Dual-Core processor RISC architecture. Its operating frequency starts from 1.1GHz - 1.3 GHz. Its chip has up to 174 million transistors, 115 W TDP based on 90nm technology [37]. IBM published its detailed floor plan Figure 3.2, its power map Figure 3.3 and its thermal profile as shown in Figure 3.4. The POWER4 floor plan consists of the following functions blocks: the Branch Execution Unit (BXU), the Floating Point Unit (FPU), the Fixed Point Integer Unit (FXU), Instruction Fetch Unit (IFU), the Instruction Decode Unit (IDU), the Instruction Sequencing Unit (ISU), the Load Store Unit (LSU), the Level 2 Cache (the Floating Point Unit (FPU), the Fixed Point Integer Unit (FXU), Instruction Fetch Unit (IFU), the Instruction Decode Unit (IDU), the Instruction Sequencing Unit (ISU), the Load Store Unit (LSU), the Level 2 Cache (L2), the Level 3 Cache (L3), and the Memory Controller (MC). IBM invents multi-chip module (MCM) having four POWER4 processors [71]. In total the MCM POWER4 chip has 8 cores shown in Figure 3.5. Thus IBM POWER4 MCM chip is selected chip. The floor plans of the POWER4 processor and the MCM are published as pictures.

Table 3.1 IBM POWER family chips

	POWER4	POWER5	POWER 6	POWER 7
Technology	90nm	65nm	65nm	45nm
Area in mm²	427	389	341	Up to 567
Transistors in million	174	276	790	Up to 1.2
Number of cores	2	2	2	4 or 8
TDP in Watt	Up to 115	Up to 130	Up to 180	NA
Operating frequency GHz	1.1 to 1.3	1.5 to 2.2	3.5 to 5.0	5 and Up
Floor plan	Detailed	Overview	Overview	Overview
Power density maps	Detailed	NA	NA	NA
Thermal profile	Detailed	Overview	Overview	Overview
References	[37], [72], [71], [94], [14]	[37], [73]	[38]	[39]

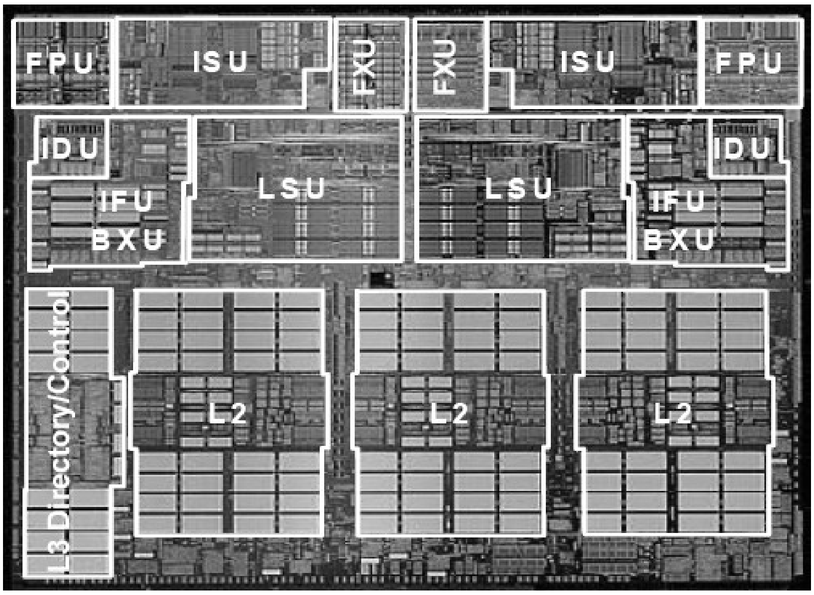


Figure 3.2 POWER4 detailed floor plan [94]

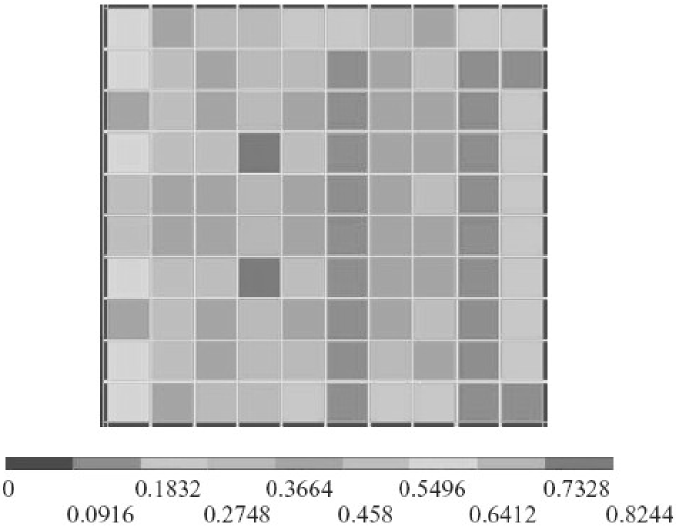


Figure 3.3 POWER4 power density distributions [72]

The entire processor manufacturers consider the CPU floor plan and its power density map as confidential data. Thus there is major difficulty to build a thermal model based on real CPU chip information. Only old CPU chip thermal data is published. The MCM POWER4 floor plan and power density map are published. The only way to build up a CPU thermal model is the reverse engineering of IBM MCM POWER4 chip.

The reverse engineering process took a lot of time and efforts as shown in section 3.3.2. The extracted MCM POWER4 chip is scaled into 45nm technology as POWER4 chip is built on the old 90nm technology [37]. Section 3.3.5 discusses the scaling of the MCM chip from the thermal point of view.

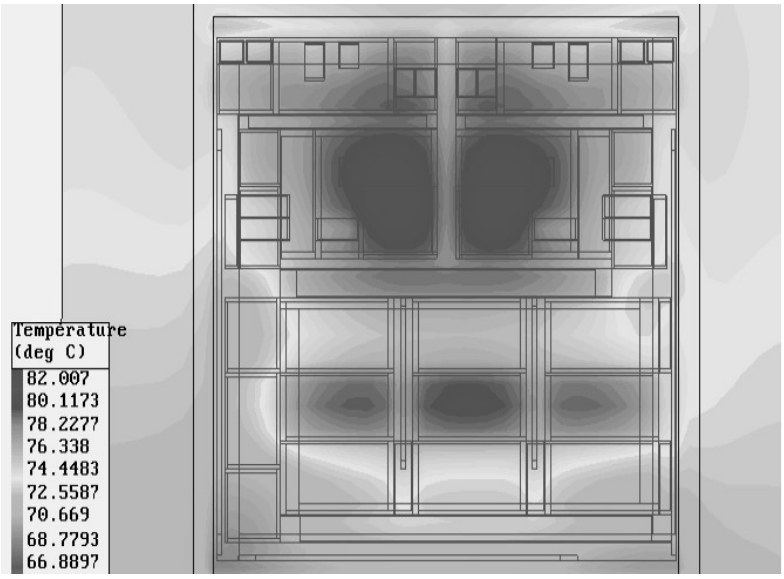


Figure 3.4 - POWER4 detailed floor plan and thermal profile [94]

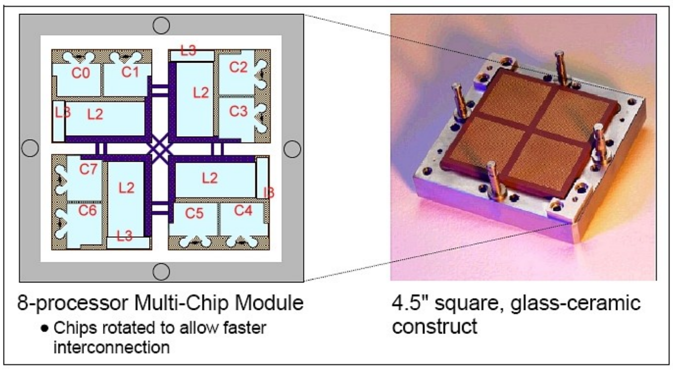

L3
C0
C1
C2
L2
L3
L2
C3
C7
L2
L2
L3
C6
L3
C5
C4
8-processor Multi-Chip Module
• Chips rotated to allow faster interconnection
4.5" square, glass-ceramic construct

Figure 3.5 - MCM POWER4 with 8 cores floor plan [14].

3.3.2 Build the Thermal Model

The reverse engineering of the IBM MCM POWER4 90nm chip floor plan is as follow:

1- The POWER4 chip detailed floor plan picture is shown in Figure 3.4 [94]. The picture pixels are used as the dimension unites. Each block geometric dimensions in pixels are extracted from POWER4 picture.

2- The MCM chip floor plan picture is shown in Figure 3.5 [14]. The four IBM POWER4 processors geometric dimensions in pixels are extracted from the MCM picture. The extracted MCM geometry is shown in Appendix Table 1. Calculate the POWER4 area in pixels from MCM POWER4 floor plan geometry.

3- Resize the extracted POWER4 floor plan dimension to fit inside their positions within the MCM POWER floor plan as shown in Figure 3.6. Thus a new detailed MCM function blocks picture is extracted.

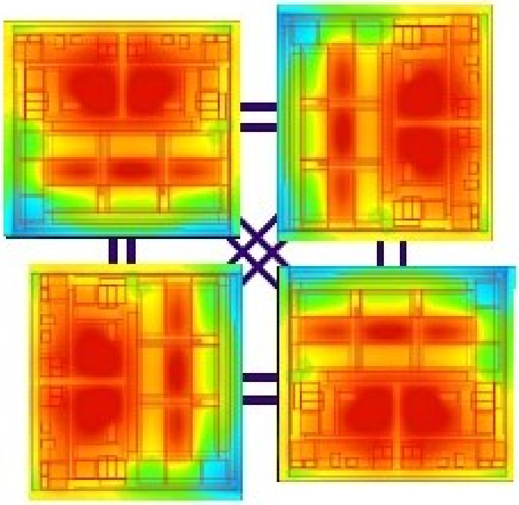


Figure 3.6 - POWER4 fit in side its MCM.

4- Extract all MCM POWER4 blocks dimensions (X , Y , width, height) in pixels. The extracted geometric dimensions are shown in Appendix Table 2. The MCM POWER4 detailed floor plan verification is done by re-plotting all blocks details from extracted data as absolute unites measurements [14], [94].

5- Calculate the required scale factor for transforming absolute unites measurements into meter. The known data are the MCM area A_1 in pixels^2, the POWER4 area A_2 in pixels^2, the real MCM area A_3 in cm^2, the real POWER4 area A_4 in cm^2, MCM width X_1 in pixels, MCM height Y_1 in pixels, POWER4 width X_2 in pixels, POWER4 height Y_2 in pixels.

$$\Lambda_3 = s_1^2 \Lambda_1 = s_1^2 X_1 Y_1 \tag{3.2}$$

$$\Lambda_4 = s_1^2 \Lambda_2 = s_1^2 X_2 Y_2 \tag{3.3}$$

Where s_1 is MCM scale factor from pixel to cm. s_1 is calculated from the equation (3.2) then equation (3.3) validates the values by calculating the POWER4 area in cm^2.

3.3.3 Validate POWER4 MCM Model

As shown in Figure 3.7, the floor plan extracted “section 3.3.2” is validated using MATLAB. The first POWER4 processor is plot at (0,0,0) in the upper left corner. The second POWER4 processor is shifted 112 units from the first processor in the X direction then it is rotated 90 degree in Z direction with respect to its new center. The third POWER4 Processor is shifted 112 units in the X direction and 103 units in the Y direction from the first processor then it is rotated 180 degree in Z direction with respect to its new center. The fourth POWER4 processor is shifted 103 units in the Y direction from the first processor then it is rotated 270 degree in Z direction with respect to its center. The extracted floor plan Figure 3.7 resembles to the real chip Figure 3.4. The cores relative locations are the same in both figures. All function block sizes maintain the same relative size ratio.

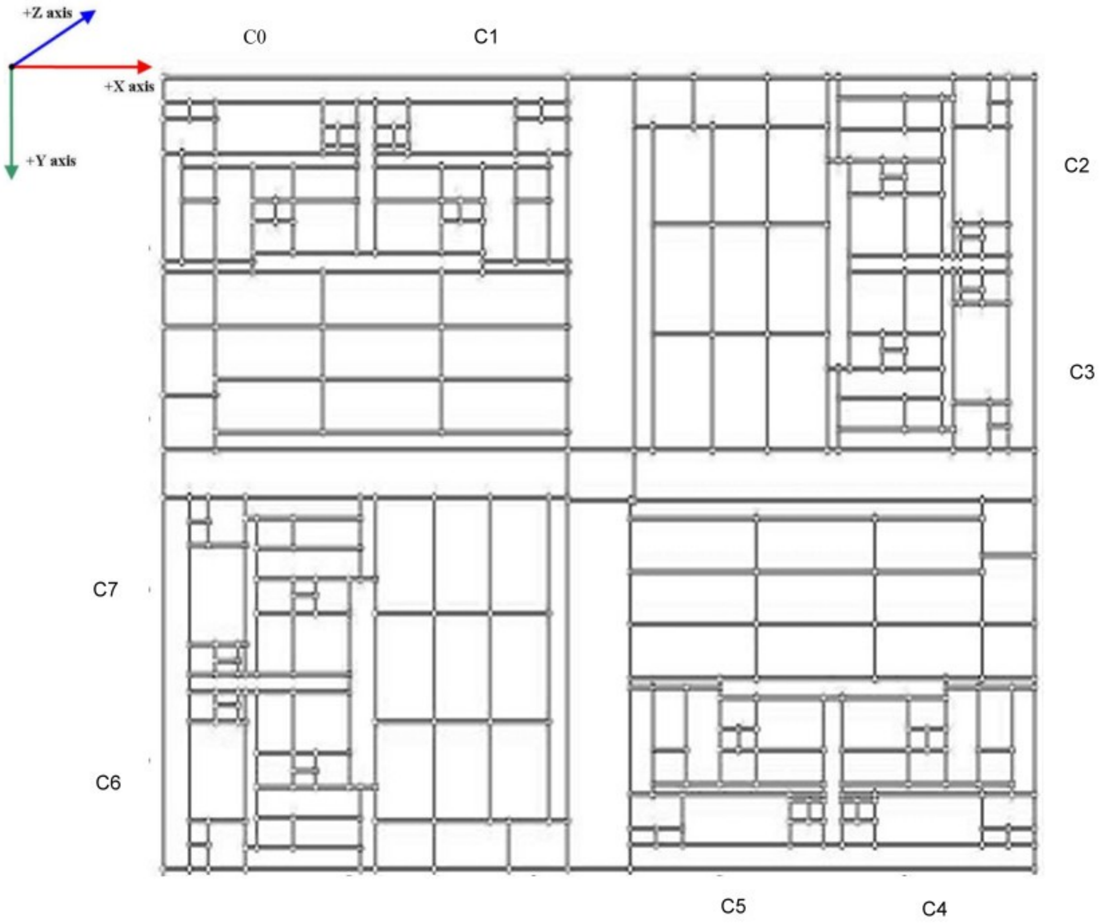


Figure 3.7 - The extracted MCM floor plan.

.

3.3.4 Scaling MCM Floor Plan into 45nm Technology

An optical shrink of the IBM POWER4 90nm floor plan is made as if it was built using 45nm. It is known that scaling 90nm into 45nm technology is not just optical zoom out. The thermal analysis is based on the equivalent circuit of thermal resistances and capacitances. The thermal resistance and capacitance depends on the function block geometry rather than the transistor geometry [74], [75], [76].

$$\Lambda_5 = s_2^2 \Lambda_3 = s_1^2 s_2^2 \Lambda_1 = s_1^2 s_2^2 X_1 Y_1 \qquad (3.4)$$

Where A_5 the MCM area in 45nm technology, s_2 is shrink scale “optical zoom out” $s_2 = 45/90$. MCM POWER4 detailed block floor plan in 45nm technology as shown in Appendix Table 3.

3.3.5 Scaling MCM POWER4 power density into 45nm technology

A semiconductor manufacturing technology node represents the smaller feature of the chip. The main reward for introducing a new semiconductor fabrication technology node is the reduction of the chip area. If the reduction is 70% of previous line width then this will lead to ~50% reduction in the chip area, i.e. 0.7 x 0.7= 0.49. The International Technology Roadmap for Semiconductors (ITRS) is a set of documents produced by a group of semiconductor industry experts. These experts are representative of the sponsoring organizations which include the Semiconductor Industry Associations of the USA, Europe, Japan, Korea and Taiwan. Starting from early 1990’s ITRS projects the voltage (V) almost flat lines; the scaling is only about 2.5% per generation. Nearly twice of the number of circuits can be fabricated on the same wafer size just by using a new technology node. Thus the cost per circuit is reduced and the ICs cost is also reduced [77]. Assuming K is the feature size scaling per technology generation. For the same circuit fabricated at the 90nm technology is scaled to the 45nm ; the area went down by K^2, the capacitance went down by K and the frequency went up by K (ideally, in practice probably less). According to the dynamic power equation, $P_{Dynamic} = f \; C \; V_{dd}^{\;2}$ and neglecting leakage power; the dynamic power also went down by K^2 thus power density is maintained constant. But in reality, if it is assumed that K = 0.7 [77], the capacitance (C) is scaled down by only 0.8, the frequency (f) went up by 1.1, and the voltage (V_{dd}) went down only 2.5%; thus the dynamic power $(P_{Dynamic})$ only went down by 0.84 (0.8 * 0.975^2 * 1.1).

As per the ITRS manufacturing design guide lines and under these assumptions above; the power density scaling is 0.84/0.5 = ~1.7 per generation. The need to control the leakage currant is the main factor restricting the reductions in voltage (V_{dd}). The assumptions can vary a lot, and these calculations assume that the leakage currant stays a constant fraction. This choice of 1.1 for the frequency (f) is also arbitrary. This rough calculation assumes a constant core micro architecture, which isn't necessarily realistic. The power density seems unlikely to scale much more than 2x per generation. It is also possible to come up with reasonable scenarios where power density scales much less. The scaling from 90nm to 65nm then from 65nm to 45nm should be K^2, where K is the scaling factor. The dynamic power density scaling is assumed to be equal to 1.7 per generation under condition that the leakage currant stays almost constant fraction.

3.4 Integrate the model to thermal simulator

This section discusses the selection for the thermal simulator. ATMI and Hotspot share some similarities. They are based on completely different methods. Table 3.2 is a comparison between the two freeware academic thermal simulator software. Virginia Hotspot simulator is selected based on simulator features and on line support provided by Hotspot team at Virginia University. The Hotspot 4.1 simulator uses the duality between RC circuits and thermal systems to model heat transfer in silicon. The Hotspot 4.1 simulator uses a Runge-Kutta (4th order) numerical approximation to solve the differential equations that govern the thermal RC circuit's operation [57], [78]. For an integrated circuit at the die level, heat conduction to the package and heat-sink, and convection from the heat-sink to ambient is the dominant mechanisms that determine temperatures on the die. The horizontal heat transfer path can account for up to 30% of heat transfer [3].The equivalent-circuit model is dependent on the floor plan and packaging of the processor.

Table 3.2 Comparison between Hotspot and ATMI thermal models

Item	Virginia Hotspot	ATMI
Latest software version	2009	2009
Thermal model	The equivalent circuit of thermal resistances and capacitances	Heat equation
Accuracy	Deviations less than 5%.	Unable to mode the chip edges
Advantages	Easy to use and to integrate with other tools	Block relative position
Limitations	Block position from the origin – number of nodes limitation - the revealing of package design details – not detailed modeling approach	Large number of nodes is necessary - heat-sink fins are not modeled explicitly - assuming a constant and uniform ambient temperature.
Flexibility	Very flexible	Not flexible
Simulator technical support	Fast response	NA
Configuration	Text files	C functions

The floor plan and the calculation of values for thermal resistance and capacitance are dependent on the areas of the functional blocks. Hotspot inputs are the chip thickness, the chip floor plan, the heat-sink parameters, the ambient temperature, the initial chip temperature, the sampling interval, and the power consumed per block per cycle. Hotspot outputs are the block temperature per cycle, and the estimated steady state temperature. The Introduction of cooling limitation is needed in order to secure the thermal throttling effect. As shown in Table3.3, the best practice thermal properties [79

] are modified in order to simulate the air cooling limitation. Thermal model cooling system is unable to maintain the CPU open loop thermal profile under the maximum operating temperature. The floor plan shown in Figure 3.7 is integrated to the Hotspot 4.1 simulator

Table 3.3 The thermal simulation setup

Item	Best practice thermal properties [79]	Simulation thermal properties
Chip thickness	0.5 mm	0.5 mm
Convection capacitance	140.4 J/K	140.4 J/k
Convection resistance	0.1 K/W	2 K/W
Heat sink side	60 mm	120 mm
Heat sink thickness	6.9 mm	6.9 mm
Heat spreader side	30 mm	30 mm
Heat spreader thickness	1 mm	1 mm
interface-material thickness	0.075 mm	0.075 mm
Ambient temperature	NA	298 K
Initial chip temperature	NA	298 K
Simulation calling interval	3.333e-06	3.333e-02

Table 3.4 POWER4 Core Power Consumption

Block Name	**P100**	**P0**
FPU_B1	1.2842	0
FPU_B2	1.2842	0
FPU_B3	1.2842	0
ISU_AL	1.2842	0
FXU_R1	0.916	0
FXU_B1	0.916	0
FXU_B2	0.916	0
FXU_B3	0.7328	0
IN_LFT	0.9192	0
IDU_AL	0.7328	0

IFU_B0	0.7328	0
IFU_B1	0.5496	0
IFU_B2	0.5496	0
IFU_B3	0.5496	0
LSU_B0	0.916	0
LSU_CR	1.6488	0
LSU_B1	1.4656	0
LSU_B2	1.4656	0
LSU_B3	0.5496	0
LSU_CH	0.916	0
IN_VER	0.5496	0
IN_HOR	0.5496	0

3.4.1 Open Loop System

The open loop thermal profile of POWER4 core is extracted using the Hotspot 4.1 simulator. The Hotspot 4.1 simulator executed about 493,153 millions Runge-Kutta (4th order) numerical approximation. It took about 650 hours running on a personal computer 3GHz single processor and 1 GB RAM under Linux Redhat Enterprise 4 over a virtual machine. The simulation is done in two simulations steps: Warming, as the core consumes the maximum power “P100” per cycle as shown in Table 3.4. The core temperature increases while the CPU cooling system is functioning normally. Cooling, the core not consuming any power “P0” per cycle as shown in Table 3.4. The core temperature is decreases as the CPU cooling system takes the heat out of the CPU chip. As shown in Figure 3.8, the extracted curve is compliant with the general CPU thermal profile curve [95], [80]. The curve in Figure 3.8 consists of 2 sub-curves the CPU heating curve to the left and the CPU cooling curve to the right as the curve extraction is done within 2 simulations steps.

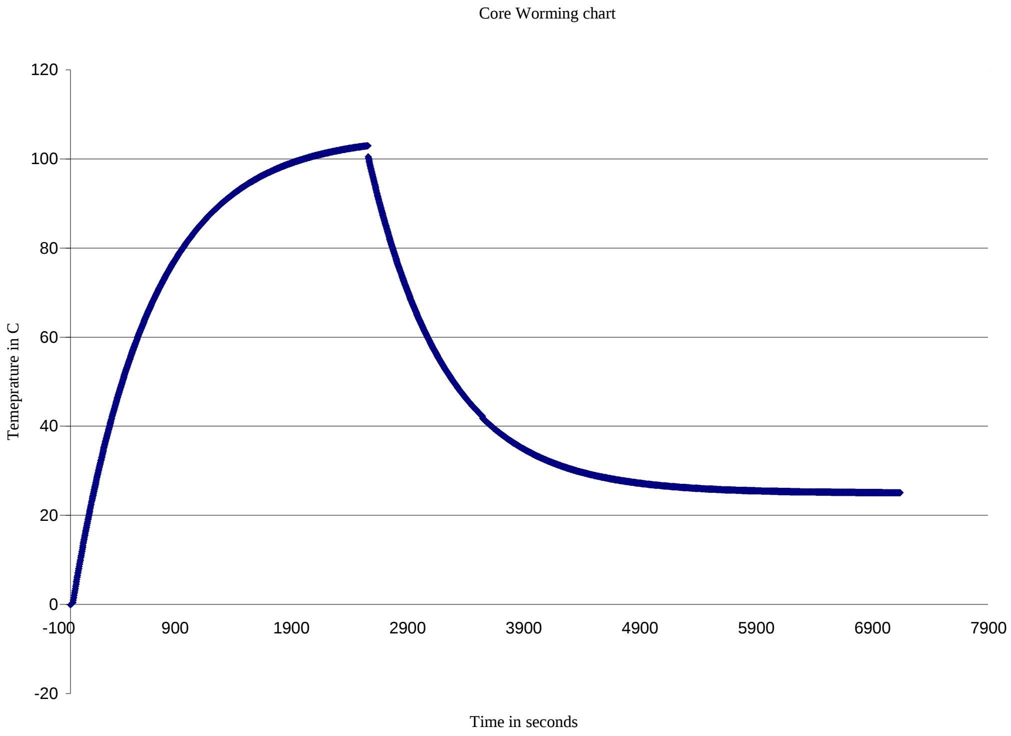


Figure 3.8 - the thermal model open loop temperature curve.

3.5 Conclusion

This Chapter discusses the available semiconductor chip simulators and analysis techniques. The Chapter describes the building of the Multi-Core CPU chip thermal model. The floor plan is similar to the IBM MCM POWER4 chip scaled to 45nm technology. The floor plan is integrated to the Hotspot 4.1 thermal simulator. The CPU open loop thermal profile curve is extracted. The next Chapter discuses the implement of the different DTM controllers based on the tradition P controller, and the different fuzzy logic techniques.

Chapter 4

Multi-Core CPU control design

4.1 Introduction

This chapter discusses three main subjects: the CPU DTM problem, the controller design, and the controller implementation. Section 4.2 presents the CPU DTM problem. Also it presents a new DTM technique. Section 4.3 discusses the maximum DTM controller response time, and presents the proposed comparison technique between different DTM controllers. Section 4.4 presents different DTM controller implementations based on P controller, 2D fuzzy controller and 3D fuzzy controller. The 2D fuzzy DTM controller is verified using MATLAB.

4.2 The CPU DTM problem

As shown in Figure 4.1; the CPU reaches the maximum operational temperature $T_{\max}$ after certain time due to maximum CPU utilization $u_{\max}$. Thus the CPU utilization is reduced to the safe utilization u_{safe} in order not to exceed $T_{\max}$. This phenomenon is called CPU thermal throttling. Figure 4.2 shows the comparison between the ideal case “no thermal constrains”, “low power consumption with thermal constraints” case and “high power consumption with thermal constraints” case. The addition of more cores to the CPU chip doesn’t increase the CPU utilization. The curve drifts to lower CPU utilization due to the CPU thermal limitation in case of low power consumption. In case of high power consumption; the CPU utilization decreases by adding more cores to the CPU chip. Thus the CPU utilization improvement is not proportional to its number of cores.

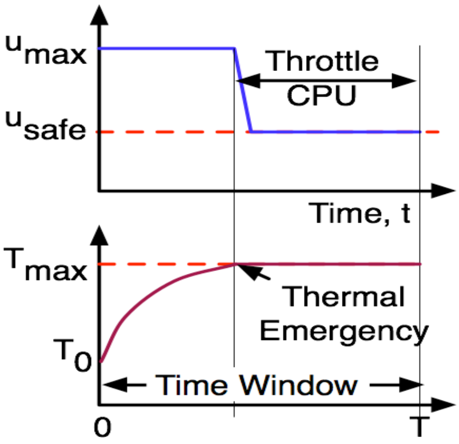


Figure 4.1 - thermal throttling [29].

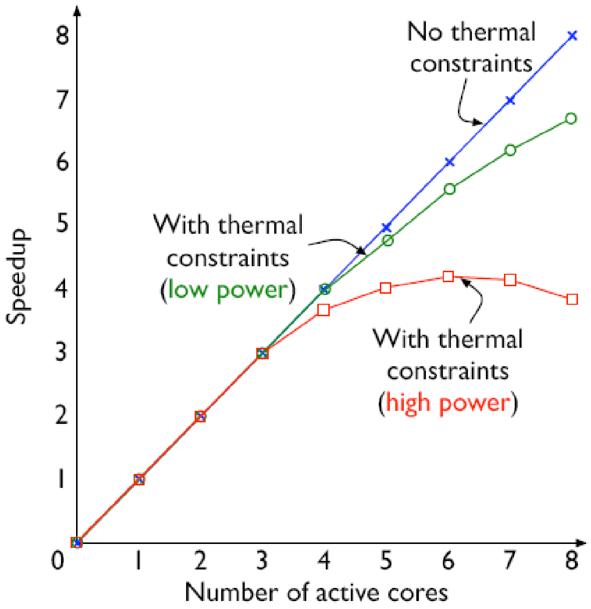


Figure 4.2 - CPU Thermal throttling [29].

4.2.1 Thermal Spare Core

As a CPU is not 100% utilized all time, thus some of the CPU cores could be reserved for thermal crises. Consider Figure 4.3, when a core reaches the steady state temperature T_1, the cooling system is able to dissipate the exhausted heat outside the chip. However, if this core is overheated, the cooling system is not able to exhaust the heat outside the chip. Thus the core temperature increases until it reaches the thermal throttling temperature T_3 [2].

The same thermal phenomena, as shown in Figure 4.3, occur due to faults in the cooling system [81]. The semiconductor technology permits more cores to be added to CPU chip. While the total chip area overhead is up to 27.9 % as per ITRS [82]. That means there is no chip area wasting in case of TSC. So reserving cores as thermal spare core (TSC) doesn’t impact CPU over all utilization. These cores are not activated simultaneously due to thermal limitations. According to Amdahl’s law: “parallel speedups limited by serial portions” [83]. So adding more cores to CPU chip doesn’t speedup due to the serial portion limits. Thus not all cores are fully loaded or even some of them are not even utilized if parallelism doesn't exist. The TSC concept uses the already existing chip space due to semiconductor technology. From the thermal point of view; the horizontal heat transfer path has for up to 30% of CPU chip heat transfer [84]. The TSC is a big coldspot within the CPU area that handles the horizontal heat transfer path. The cold TSC reduces the static power as the TSC core is turned off. Also the TSC is used simultaneous with other DTM technique. The equation 4.1 calculates number of TSCs cores. The selection of TSC cores number is dependant on the number of cores per chip and maximum power consumed per core as follow:

$$N_{TSC} = \left|\left\{\left(P_{mx} N_{C} - 198\right)/198\right\}\right| \qquad (4.1)$$

where N_{TSC} : minimum number of TSCs, abs : absolute value, P_{mx} : maximum power consumed per core, N_{C} : total number of cores, 198 Watts is the thermal limitation of the air cooling system. Figure 4.3 shows core profile where lower curve is normal thermal behavior. The upper curve is the overheated core, T_1 is the steady state temperature, T_1 corresponds to the temperature at t_1. t_2 is required time for a thermal spare core to takeover threads from the overheated core, T_2 corresponds to the temperature at t_2. T_3 is the throttling temperature, and corresponds to the temperature at t_3.

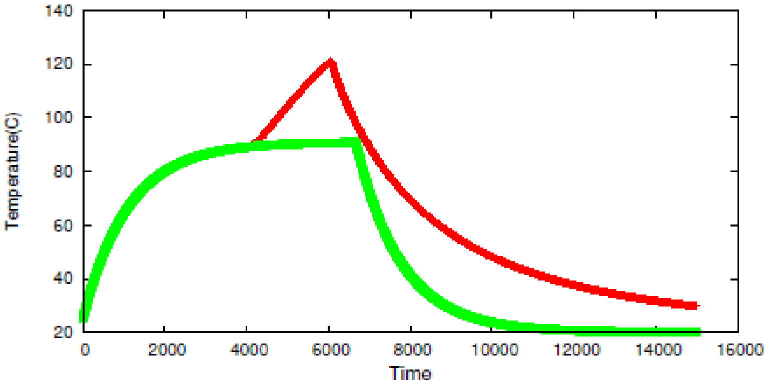


Figure 4.3 - Core thermal throttling “upper” curve [81].

TSC technique uses the already existing cores within CPU chip to avoid CPU thermal throttling as follow:

Hot TSC: is a core within the CPU powered on but its clock is stopped. It only consumes static power. It is a fast replacement core. However, it is still a heat source. Cold TSC: is a core within the CPU chip powered off (no dynamic or static power consumed). It is not a heat source, but it is a slow replacement core. Its activation needs more time than hot TSC. But the cold TSC reduces the static power dissipation. Also cold TSC generates cold spot with relative big area that helps exhausting the horizontal heat transfer path out of the chip.

Defining T_{tsc} as the TSC activation temperature as follow:

$$T_{ss} \leq T_{tsc} \leq T_{th} \tag{4.2}$$

[illegible] (4.3)

Where: T_{ss} : core steady state temperature. T_{tsc} : The temperature that triggers TSC process. T_{th} : CPU throttling temperature. t_{tsc} : The time of activating TSC. t_{th} : The time required to reach thermal throttling. t_{CT} : The estimated time required for completing the current tasks within the over heated core. This information is not always accurate at run time. t_{TM} : Time required migrating threads from over heated core to TSC. If any core reaches T_{tsc} then the DTM controller will inform the OS to stop assigning new tasks to this overheated core. Thus the OS doesn't assign any new task to the overheated core. Therefore, T_{tsc} is not predefined constant temperature but variable temperature between T_{ss} and T_{th} . The DTM selects T_{tsc} depending on the minimum time required to evacuate the over heated core.

4.2.2 TSC illustration

This section illustrates the thermal spare cores (TSC) technique

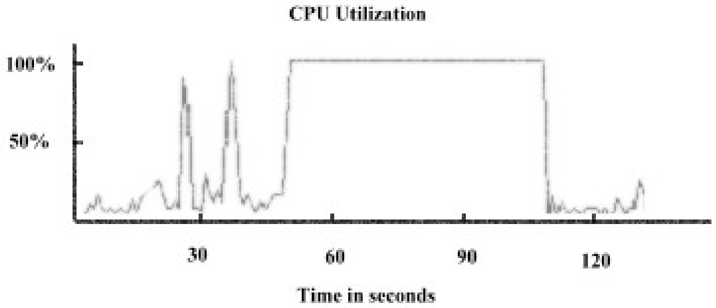


Figure 4.4 - The CPU congestion due to thermal limitations

As shown in Figure 4.4, the CPU is 100% utilized for duration about 50 seconds. The OS realizes that the CPU congestion. The CPU executes its tasks slowly. In fact the CPU suffers from thermal throttling. This CPU utilization curve shows CPU congestion from OS point of view due to thermal limitations.

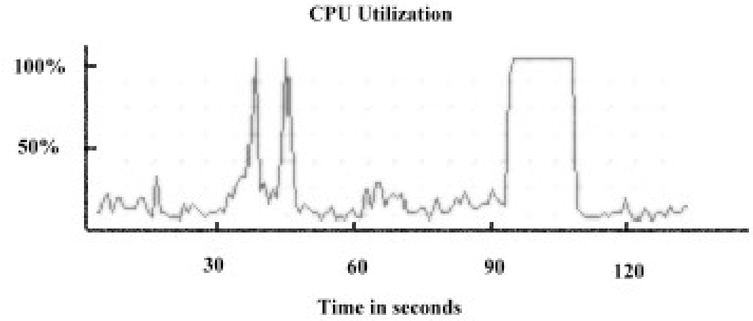


Figure 4.5 - Activating TSC during the CPU thermal crises

As shown in Figure 4.5, The DTM controller detected the CPU high temperature. Thus the DTM controller executes the TSC algorithm. At 40 seconds time line, a TSC core replaces a hot core. The handover between the hot core the TSC core lead to a CPU peak. But The CPU improves its speed after that peak; as the TSC is still cold relatively and operates at higher frequency. At 86 seconds, the CPU reaches thermal throttling again. Thus the CPU reaches congestion again. So the activation of a TSC core during the CPU thermal crises decreases the duration of the CPU degradation from 50 seconds to 15 seconds duration.

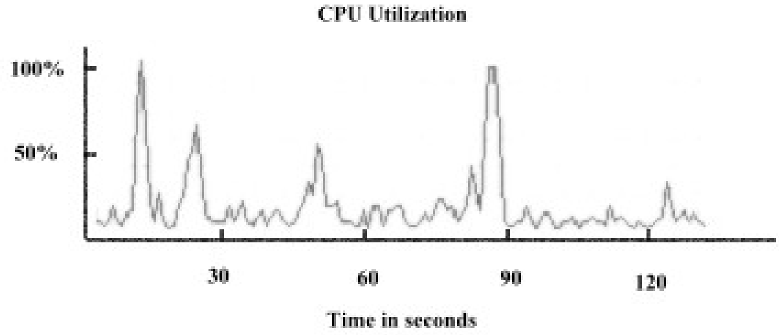


Figure 4.6 - Activating many TSC during the CPU thermal crises

As shown in Figure 4.6, the activation of 3 TSC cores during the thermal crises at 25 seconds, 45 seconds and 85 seconds time lines respectively increases the CPU utilization. The CPU executes its tasks normally without congestion rather than some CPU peaks. AS this CPU chip has many spare cores; the DTM controller activates the required TSC during the CPU thermal crises. So the CPU avoids the thermal throttling theoretically.

4.3 DTM Controller Design

This thesis scope covers the following DTM controller design specifications: symmetric Multi-Core CPU, cooling system limitation, Dynamic Voltage and Frequency Scaling (DVFS) and the CPU resources utilization. The controller design like all engineering designs involves tradeoffs by a satisfactory compromise among the design goals [85].The DTM controller design involves the core operating frequency, the core temperature, the power consumed per core and the total CPU utilization. From feedback control point of view, the CPU cores are heat sources. The DTM controller goal is to decrease their thermal effect (inner control loop in Figure 4.7). The cores are performance improvement tools. It is required to utilize them as much as possible (outer control loop in Figure 4.7). The enhanced feedback control DTM system has an inner loop for local temperature control and outer loop for the CPU utilization [8]. As shown in Figure4.7, the sets of four bold lines represent the sets of data from each core. The DTM controller selects the maximum operating frequency depending on the thermal constrains. If the core temperature is high; then the DTM controller will perform a distributed DVFS policy for each core based on P or PI control [8]. DVFS reduces the effect of hotspots. This increases cooling system efficiency. In case of thermal emergency, the DTM controller performs global DVFS policy or even "stop-go" for all cores as a last resort.

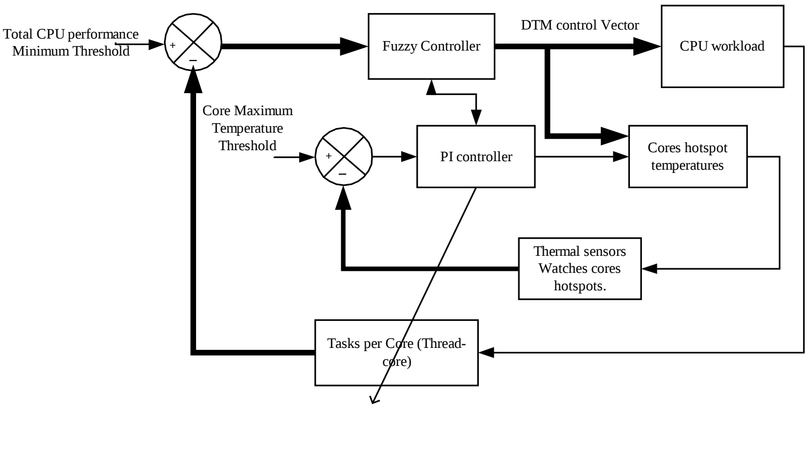

Total CPU performance Minimum Threshold
+
−
Fuzzy Controller
DTM control Vector
CPU workload
Core Maximum Temperature Threshold
+
−
PI controller
Cores hotspot temperatures
Thermal sensors Watches cores hotspots.
Tasks per Core (Thread-core)

Figure 4.7 - The DTM control system [8].

When the CPU temperature reaches the maximum operating temperature $T_{\max}$, the DTM controller performs failsafe strategy to avoid the CPU hardware damage. As shown in Figure 4.8, the enhanced DTM system takes into consideration the cooling system, the cores temperatures, and the cores operating frequencies.

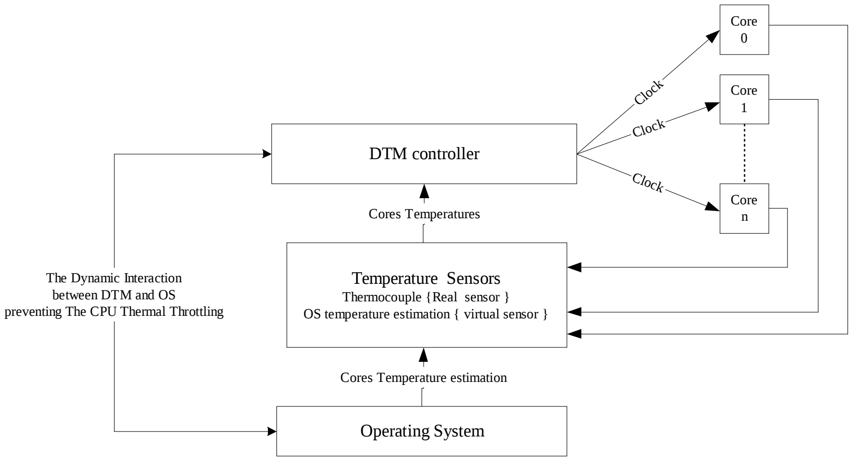


Figure 4.8 - Enhanced DTM systems

The overall temperature of Multi-Core CPU is highly correlated with temperature of each core. Hence, the thermal model for single processor environments can not be directly applied in Multi-Core CPU due to the potential heterogeneity. The DTM schemes considering the thermal correlation effect among the neighboring cores enhances the Multi-Core CPU thermal behaviors [86].

In order to reach the required high CPU utilization under thermal constraints, fuzzy DTM controller interact with the traditional P controller in the feedback control systems as shown in Figure 4.8. The traditional fuzzy set is two-dimensional (2D) with one dimension for the universe of discourse of the variable and the other for its membership degree. The fuzzy control handles the CPU thermal process without knowing its transfer function. This simplifies the DTM controller design and reduces design time [87].

This 2D fuzzy set is not able to handle the spatial information. The traditional fuzzy logic controller (FC) developed from this 2D fuzzy set is not able to control the distributed parameter system. A three-dimensional (3D) fuzzy set is defined to be made of a traditional fuzzy set and an extra dimension for spatial information. Based on concept of the 3D fuzzy set, a new fuzzy control methodology is used to control the distributed parameter system. Similar to the traditional FC, it still consists of fuzzification, rule inference, and defuzzification operations. Different to the traditional FC, it uses multiple sensors to provide 3D fuzzy inputs and possesses the inference mechanism with 3D nature that can fuse these inputs into a so called "spatial membership function". Thus, a simple 2D rule based is used for two obvious advantages. One is that rules will not increase as sensors increase for the spatial measurement; the other is that computation of this 3D fuzzy inference is reduced for real world applications [16]. The 3D fuzzy DTM controller could handle the thermal correlation effect among neighboring cores. The thermal correlation is based on "hotspot distance" between the each core. The hotspot point per core is selected as the highest temperature point within this core. The hotspot distance is calculated as the distance between these hotspot points. Figure 4.9 illustrates the calculation of the hotspot distances between core "C0" and the other cores "C1 till C7". Thus the thermal correlations are calculated as function of the hotspot distances between the different cores. The hotspot distances changes depending on the hotspot location per core. The 3D fuzzy DTM controller is a solution of the online thermal dependencies. The operating system (OS) hosts the fuzzy DTM controller. The DTM controller receives the cores temperatures and operating frequencies. Then it selects the appropriate core clock frequency depending on the correlation between all cores. If any core reaches the maximum operating temperature $T_{\max}$, the DTM fuzzy controller informs OS to stop assigning new tasks. The DTM fuzzy controller uses the advanced configuration & power interface (ACPI). As the ACPI is an industry-standard interface enabling OS-directed configuration, power management, and thermal management of different CPU architectures. The ACPI interaction is already applied for example with "Intel® core TM duo processor" [88]. The ACPI defines the core utilization states relative to DVFS. So the fuzzy DTM controller based on the OS level selects the power state for each core.

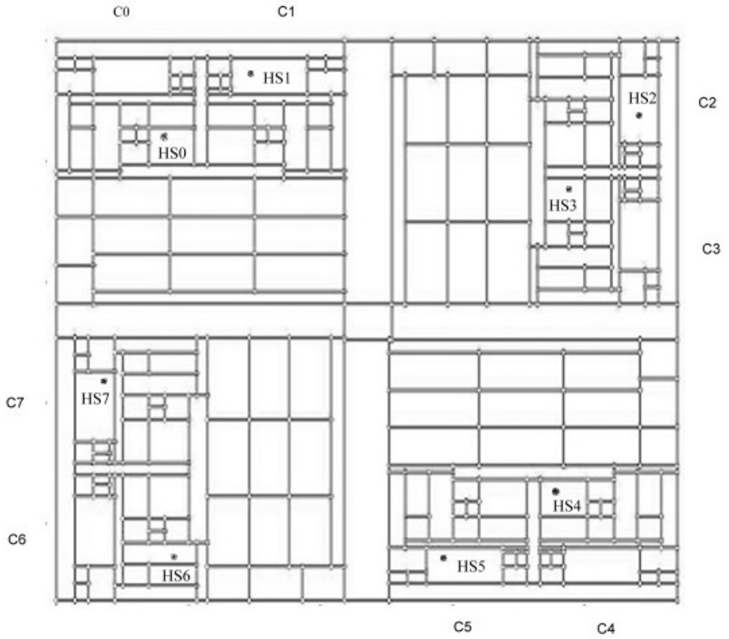


Figure 4.9 - The core thermal correlations hotspot distances "HS".

4.3.1 The Maximum DTM Controller Delay

The required DTM controller speed to adjust the CPU thermal profile is calculated as follow: $t_{DTM} < (t_{omx} - t_{oc})$

(4.4)

Where t_{DTM} : the maximum DTM controller delay time, t_{omx} : the time required to reach open loop maximum operational temperature, t_{oc} : the time required to reach open loop control temperature. Figure 4.10 is an example for the selection of the maximum operational temperature "85 °C"; assuming that the maximum open loop control temperature is 84 °C. t_{omx} = 1135 seconds and t_{oc} = 1125 seconds thus t_{DTM} < 10 seconds which is a feasible value for the real–time applications.

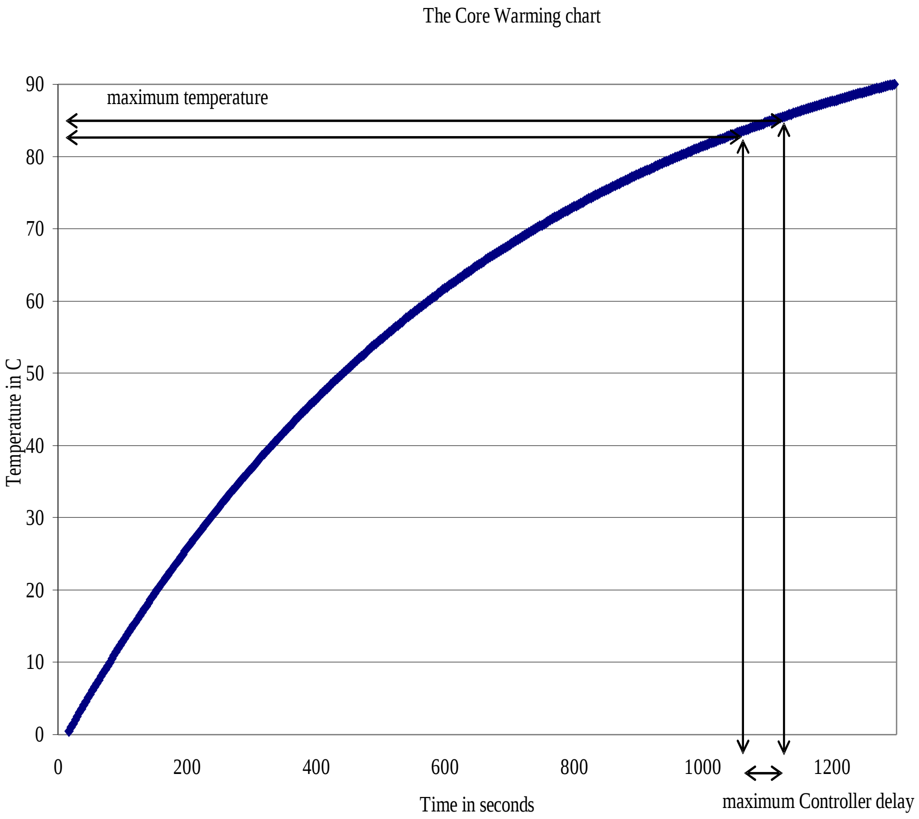


Figure 4.10 - Example of the maximum Controller delay time calculation

4.3.2 DTM Evaluation Index

An evaluation index for the DTM controller outputs is required. As per the thermal throttling definition in section 4.2, “the operating frequency is reduced in order not to exceed the maximum temperature”. Both frequency and temperature changes are monitored as there is a non linear relation between the CPU frequency and temperature. One of the DTM objectives is to minimize the frequency changes. The core theoretically should work at open loop frequency for higher utilization. But due to the CPU thermal constrains the core frequency is decreased depending on core hotspot temperature.

The second DTM objective is to decrease the CPU temperature as much as possible without affecting the CPU utilization. A multi-parameters evaluation index ζ_t is proposed. It consists of the summation of each parameter evaluation during normalized time period. This index is based on the weighted sum method [97]. The objective of multi-parameters evaluation index shows the different parameters effect on the CPU response. Thus the designer selects the suitable DTM controller that fulfils his requirements. The multi-parameters evaluation index permits the selection of DTM design that provides the best frequency parameter value without leading to the worst temperature parameter value.

The DTM evaluation index ζ_t calculation consists of 5 phases:

1- Identify the required parameters

2- Identify the design parameters ranges

3- Identify the desired parameters values of each range $\sigma_{ij}^{\mathrm{Desired}}$

4- Identify the actual parameters values of each range $\sigma_{\mathrm{ij}}^{\mathrm{Actual}}$

5- Evaluate each parameter and the over all multi- parameter evaluation index

$$\zeta_t = \sum_{i=1}^{l} \lambda_i \tag{4.5}$$

The parameter λ_i value during the evalaution time period is the summation of the evaluation ranges divided by the number of ranges m_i.

$$\lambda_i = \frac{1}{m_i} \sum_{j=1}^{m_i} \sigma_{ij} \tag{4.6}$$

Each evaluation range σ_{ij} is evaluated over a normalized time period

$$\sigma_{ij} = \left(\frac{\sigma_{\mathrm{ij}}^{\mathrm{Actual}}}{\sigma_{ij}^{\mathrm{Desired}}} \right) \tag{4.7}$$

$\sigma_{\mathrm{ij}}^{\mathrm{Actual}}$ is the actual percentage of time the CPU runs at that range

$\sigma_{ij}^{\mathrm{Desired}}$ is the desired percentage of time the CPU runs at that range

The λ value should be 1 or near 1. If $\lambda < 1$ then the CPU runs less time than the desired within this range. If $\lambda > 1$ then the CPU runs more time than the desired within this range. Thus the multi-parameters evaluation index equation is:

$$\zeta_t = \sum_{i=1}^{l} \frac{1}{m_i} \sum_{j=1}^{m_i} \left(\frac{\sigma_{\mathrm{ij}}^{\mathrm{Actual}}}{\sigma_{ij}^{\mathrm{Desired}}} \right) \tag{4.8}$$

The DTM controller evaluation index desired value should be $\zeta_t = l$ or near l, where l is the number of parameters. The Multi-parameters evaluation index permit the designer to evaluate each rang independent on the other ranges and also evaluate the over all DTM controller response.

The multi-parameters evaluation index is flexible and accepts to add more evaluation parameters. This permits the DTM controller designer to add or remover any parameter without changing the evaluations algorithm. Figure 4.11 shows an example of the parameter λcalculation. In this example the parameter λ is the temperature. The temperature curve is divided into 3 ranges: High (H) – Medium (m) – Low (L), these ranges are selected as follow: High “greater than78 °C”, Medium “between 74 °C and 78 °C”, and Low “lower than 72 °C”. The actual parameters values of each range $\sigma_{\mathrm{ij}}^{\mathrm{Actual}}$ is calculated as follow: $\sigma_{\mathrm{i\,High}}^{\mathrm{Actual}}$ = 20.5%, $\sigma_{\mathrm{i\,Medium}}^{\mathrm{Actual}}$ = 76%, and $\sigma_{\mathrm{i\,Low}}^{\mathrm{Actual}}$ =3.5%

Chapter 5 discusses the implementation of the DTM evaluation index in details.

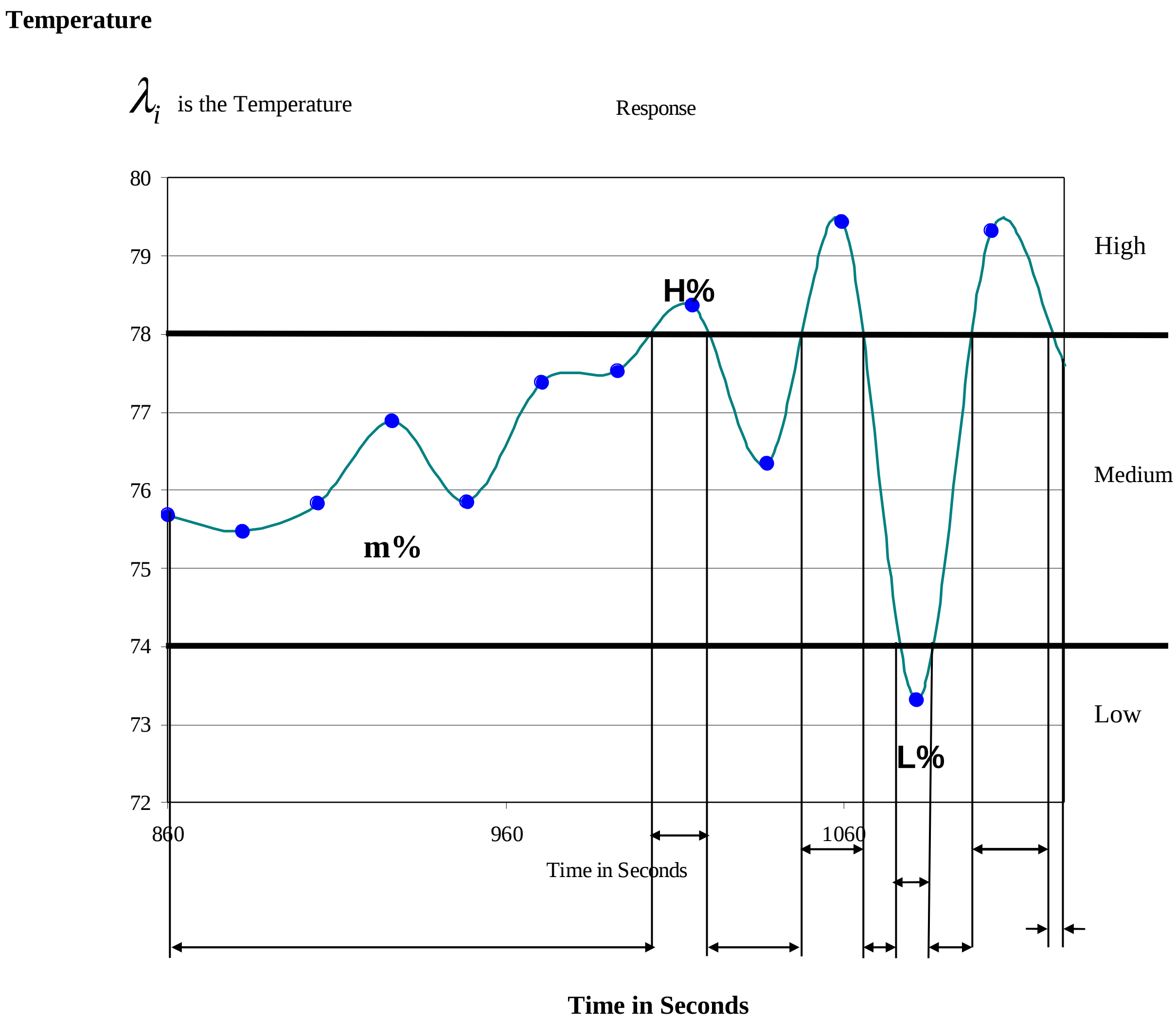


Figure 4.11 - Example of actual parameter value λcalculation

4.4 The DTM Controller Implementation

This section discusses the different control techniques use for implementing the DTM controller.

4.4.1 The Proportional DTM Controller

The proportional controller performs DVFS for each core. As described in section 2.3.4 the DVFS policy involves more of a continuous adjust scheme. A set point is used below the thermal threshold. P controller adapts the frequency and voltage levels towards this target threshold [8]. The integration of the P controller to the Hotspot 4.1 thermal simulator automates the temperature control. A shell script called “Mediator” is the interface between the P controller and the thermal simulator. The mediator script calls the P controller subroutine for each core. It is functioning as if there are 8 SISO P controllers without correlation. The P controller gain tuning is done manually until reaching the optimal utilization. The DTM P controller block diagram is shown in Figure 4.12. The threshold $T_{\max}$ equals 80 °C, T_e is a safety margin of 5 °C, and the feedback control selects the appropriate power profile proportional to the temperature deviation. The DTM P controller objective is not to make the Core work at $T_{\max}$; while its objective is to select the operating frequency proportional to $T_{\max}$. The P controller select maximum frequency if T_e is maximum and vice versa.

The power profile is proportional to the operating frequency. With high temperature deviation from the maximum threshold, the core runs at high operating frequency and vise versa. When the temperature deviation reaches zero, the core fails safe as it reaches the maximum operating temperature.

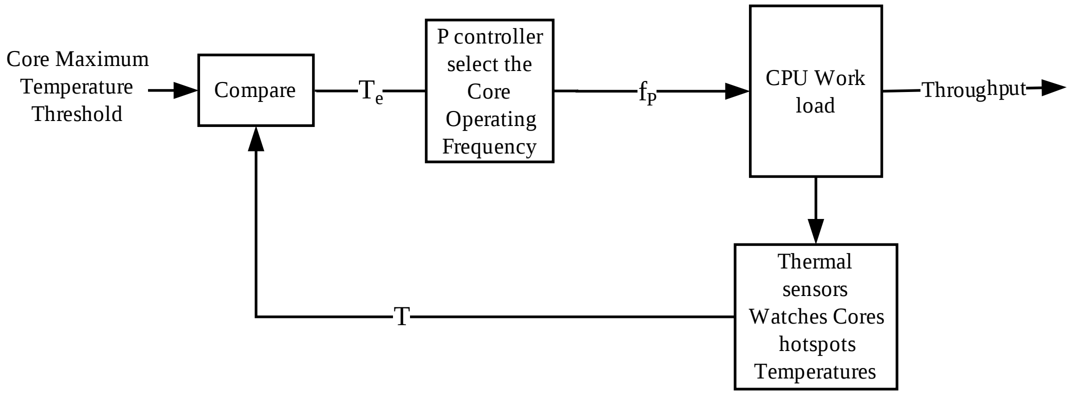

Core Maximum Temperature Threshold
Compare
T_e
P controller select the Core Operating Frequency
f_p
CPU Work load
Throughput
Thermal sensors Watches Cores hotspots Temperatures
T

Figure 4.12 - the DTM P controller block diagram.

4.4.2 2D Fuzzy DTM Controller Implementation

The 2D fuzzy DTM controller in Figure 4.13 is based on fuzzy logic control. The block diagram consists of P controller adjusting core operating frequency at time n depending on the core local hotspot temperature at time n-1. Assuming there is 1 sample delay due to the thermal simulator behavior; the same P controller described in section 4.4.1 is used to perform DVFS. The P controller generates the output control vector as the operating frequencies of each core. A fuzzy controller (FC) follows the P controller. This fuzzy controller adjusts the output frequency control vector depending on rules based design. As shown in Figure 4.13 both the P controller and the fuzzy controller collect the feedback temperature deviations. The P controller selects the core operating frequency at time n. The P controller output is compared to CPU maximum operating frequency. The fuzzy controller inputs are the frequency deviation, the frequency deviation change, the temperature deviation, and the temperature deviation change at time n. Thus the fuzzy controller output depends on the CPU utilization and the core hotspot temperature. The 2D fuzzy DTM controller doesn’t take into consideration the correlation between the cores operating frequencies and their hotspot temperatures. The fuzzy controller output is calculated using the fuzzy center of area [87].

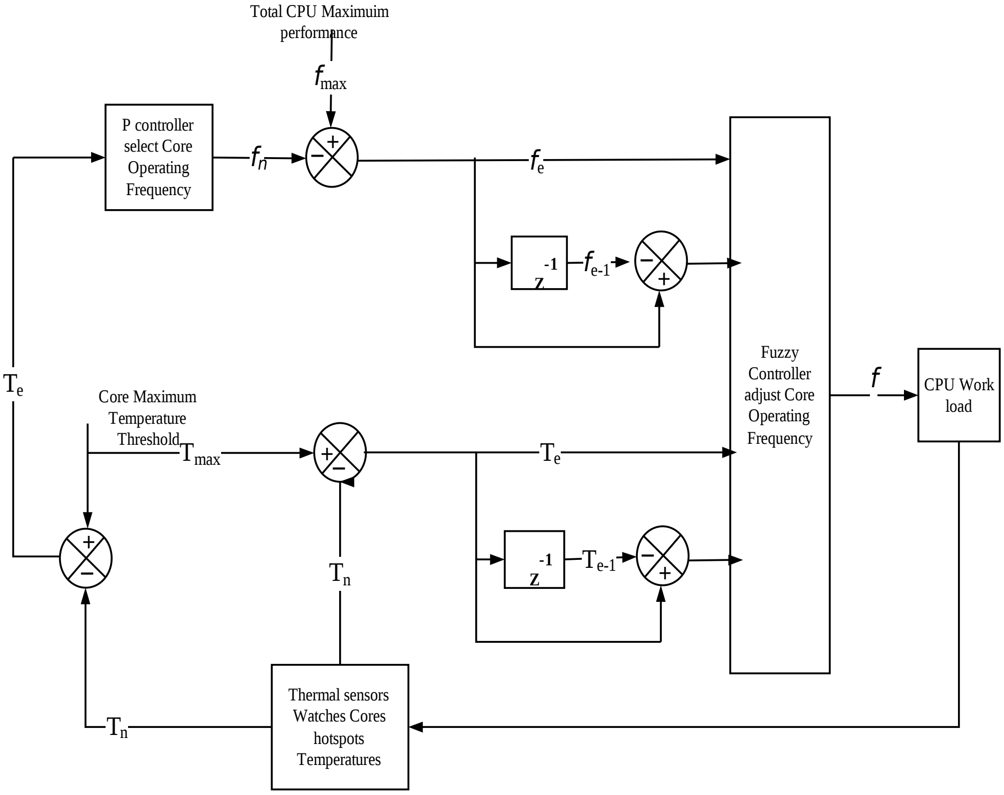

Total CPU Maximuim performance
$f_{\max}$
P controller select Core Operating Frequency
f_n
f_e
z^{-1}
f_{e-1}
Fuzzy Controller adjust Core Operating Frequency
f
CPU Work load
T_e
Core Maximum Temperature Threshold
$T_{\max}$
T_e
z^{-1}
T_{e-1}
T_n
Thermal sensors Watches Cores hotspots Temperatures
T_n

Figure 4.13 - the general fuzzy DTM controller block diagram

The 2D-FC is designed as follow [89]:

1- Define the control objectives:

The DTM controller is controlling the total CPU throughput under thermal constrains. The FC selects each core operating frequency. It is required to have the maximum possible CPU throughput under thermal constrains. The CPU maximum temperature $T_{\max}$ doesn't exceed $T_{\max}$ 80 °C for each core. Where the maximum frequency $f_{\max} = 100$ and $T_{control}$ = 70 °C

2- Determine the input and output relationships

The inputs: for each core at step n the FC is receiving the temperature deviation, the temperature deviation change, the frequency deviation, and the frequency deviation change. The outputs: each core operating frequency at step n+1. The relationships: at step n CPU throughput is proportional to cores operating frequency.

$$f_e = f_{\max} - f_n \qquad (4.9)$$

Where f_e is the core operating frequency deviation at time n, f_n is the core operating frequency at time n, and $f_{\max}$ is the maximum operating frequency.

$$T_e = T_{\max} - T_n \qquad (4.10)$$

T_e is the core temperature deviation at time n, T_n is the core temperature at time n, and $T_{\max}$ is the maximum temperature at time n.

$$f_{\Delta e} = f_e - f_{e-1} \qquad (4.11)$$

Where $f_{\Delta e}$ is the core operating frequency deviation change at time n, and f_e is the operating frequency deviation at time n,

$$T_{\Delta e} = T_e - T_{e-1} \qquad (4.12)$$

$T_{\Delta e}$ is The core operating temperature deviation change at time n, and T_e is the operating temperature deviation at time n.

The fuzzy outputs: Maximum Operating Frequency " $f_{\max}$ ", Dynamic Voltage & Frequency Scaling "DVFS" output response, Thermal Spare Core "starts TSC" output response and Fail Safely "FS".

The rules space contains 120 rules as per Appendix Table 4 while there are only 78 valid fuzzy rules. The 78 fuzzy rules are implemented according to the Meta decisions rules below:

Rule1: if the core temperature deviation is a very large negative “VLN” then the core operates at the maximum operating frequency independent on the other fuzzy variables.

Rule2: if the core temperature deviation is in within the zero range “ZZ” then the core fails safe independent on the other fuzzy variables.

Rule3: if the core temperature deviation is a very small negative “VSN” then the TSC technique is triggered and the core is offloaded independent on the other fuzzy variables.

Rule4: if the core temperature deviation is a small negative “SN” and the core frequency deviation is in one of these states: a zero range “ZZ” or a zero negative range “ZN” or a very small negative “VSN” then the DVFS techniques is triggered.

Rule5: if the core temperature deviation is a large negative “LN” and the core frequency deviation is in one of these states: a large negative “LN” or a small negative range “SN” or a very large negative “VLN” then the TSC techniques is triggered.

Rule6: if the core temperature deviation is a small negative “SN” then the DVFS techniques is triggered independent on the core temperature change deviation or the core frequency change deviation.

Rule7: if the core frequency deviation is a very large negative “VLN” or a large negative “LN” then the core operate at the maximum operating frequency independent on the core frequency change deviation.

Rule8: if the core frequency deviation is a small negative “SN” and the core frequency change deviation is a positive “PP” or the core temperature change deviation is a zero negative range “ZN” then the core operates at the maximum operating frequency.

Rule9: if the core frequency deviation is a small negative “SN” and the core frequency change deviation is a negative range “ZN” or the core temperature change deviation is a positive “PP” then the DVFS techniques is triggered.

Rule10: if the core frequency deviation is a very small negative “VSN” and the core frequency change deviation is a zero negative range “ZN” or the core temperature change deviation is a positive “PP” then the DVFS techniques is triggered.

Rule11: if the core frequency deviation is a very small negative “VSN” and the core frequency change deviation is a zero negative range “ZN” or the core temperature change deviation is a positive “PP” then the TSC techniques is triggered.

Rule12: if the core frequency deviation is within the zero negative range “ZN” then the TSC techniques is triggered independent on the other fuzzy variables.

Rule13: if the core frequency deviation is in the zero range “ZZ” and the core frequency change deviation is with in the zero negative range “ZN” or the core temperature change deviation is with in the zero negative range “ZN” then the DVFS techniques is triggered.

Rule14: if the core frequency deviation is within the zero range “ZZ” and the core frequency deviation change is a positive “PP” or the core temperature deviation change is a positive “PP” then the TSC techniques is triggered. Figure 4.14 shows the 2D fuzzy DTM controller implementation

Membership functions:

Table 4.1 summarizes the input fuzzy variables and their associate membership functions. All membership functions have Normal distribution in the following proof of concept design implementation. Table 4.2 presents the mean and the standard deviation of the input membership functions (MSF). The temperature deviation is represented by 5 input MSF m1 to m5. The temperature deviation change is represented by 2 input MSF m6 to m7. The frequency deviation is represented by 6 input MSF m8 to m13. The frequency deviation change is represented by 2 input MSF m14 to m15.

In total, this design includes 15 input membership functions. The output membership functions are tuned per DTM controller. Chapter 5 discusses these MSF in details. The output MSF tuning has great effect on fuzzy DTM controller behavior. In general we have four outputs MSF Max - DVFS - TSC MSF - FS. The “center of area" algorithm is used for the defuzzification process [87].

Table 4.1 Fuzzy inputs

Frequency deviation f_e inputs		Temperature deviation T_e inputs	
ZZ	zero	ZZ	zero
ZN	zero or negative	VSN	very small negative
VSN	very small negative	SN	small negative
SN	small negative	LN	large negative
LN	large negative	VLN	very large negative
VLN	very large negative		
Frequency deviation change $f_{\Delta e}$		Temperature deviation change $T_{\Delta e}$	
PP	positive	PP	positive
ZN	zero or negative	ZN	zero or negative

Table 4.2 Input MSF Normal Distribution Configurations.

The Input MSF	Mean	Standard Deviation
m1	0	0.5

m2	-2.5	-0.75
m3	-5	1
m4	-8	0.75
m5	-10	0.5
m6	-5	2
m7	5	2
m8	-100	5
m9	-80	75
m10	-60	10
m11	-40	75
m12	-20	10
m13	0	5
m14	-50	20
m15	50	20

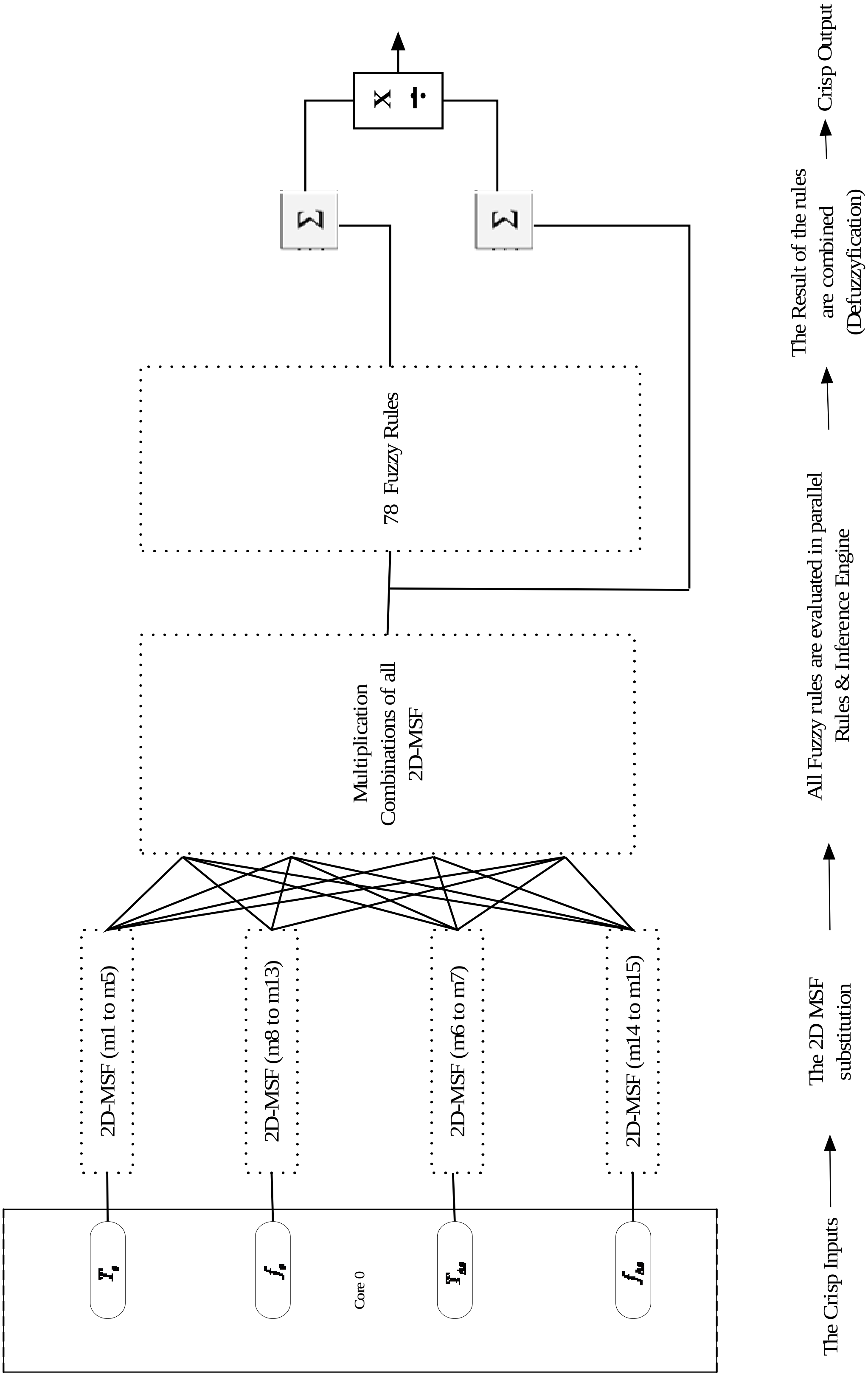

Core 0
2D-MSF (m1 to m5)
2D-MSF (m8 to m13)
2D-MSF (m6 to m7)
2D-MSF (m14 to m15)
Multiplication Combinations of all 2D-MSF
78 Fuzzy Rules
Σ
Σ
X ÷
The Crisp Inputs
The 2D MSF substitution
All Fuzzy rules are evaluated in parallel Rules & Inference Engine
The Result of the rules are combined (Defuzzyfication)
Crisp Output

Figure 4.14 2D-fuzzy DTM of a single Core “C0”

4.4.3 3D Fuzzy Problem Formulation

The 3D fuzzy control is able to handle the correlation between the different variable parameters of a distributed parameter system [16]. Thus the 3D fuzzy logic is able to process the Multi-Core CPU correlation information. The 3D fuzzy control demonstrates its potential to a wide range of engineering applications. The 3D fuzzy control is feasible for real-time world applications [16]. The thermal management process is a distributed parameter systems [90]. The thermal management process is represented by the nonlinear partial differential equations [91][92].

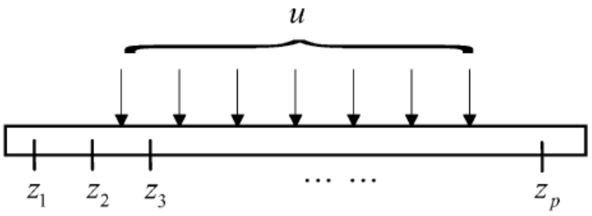


Figure 4.15 Actuator u and the measurement sensors at p point [16].

Figure 4.15 presents a nonlinear distributed parameter system with one actuator ($\gamma = 1$). Where p point measurement sensors are located at $z_1, z_2, \ldots\ldots, z_p$ in the one-dimensional space domain respectively and an actuator u with some distribution acts on the distributed process. Inputs are measurement information from sensors at different spatial locations. i.e., deviations $e_1, e_2, \ldots\ldots, e_p$ and deviations change $\Delta e_1, \Delta e_2, \ldots\ldots, \Delta e_p$ where $e_1 = y_d(z_i) - y(z_i, n)$, $\Delta e_i = e_i(n) - e_i(n-1)$

$y_d(z_i)$ denotes the measurement value from location z_i, $n, n-1$ denote the n and $n-1$ sample time input. The output relationship is described by fuzzy rules extracted from knowledge. Since p sensors are used to provide $2p$ inputs. The rule based is $2p$ dimensional with the following structure [93]:

$R^{j}_{miso}:\ \ if\ \ d_{11}\ \ is\ \ E^{j}_{11}\ \ and.....and\ \ d_{1p}\ \ E^{j}_{1p}\ \ and\ \ d_{21}\ \ is\ \ F^{j}_{21}\ \ and...and\ \ d_{2p}\ is\ \ F$

$Then\ \ u\ \ is\ \ K^{j}$ Where $d_{1i}=e_{i}$ and $d_{2i}=\Delta e_{i}$, $i=1,2,....,p$, R^{j}_{miso} denotes the j^{th} rule $(j=1,2......,M)$ K_{j} , E^{j}_{ti} , F^{j}_{ti} $(t=1,2\ and\ i=1,2,...p)$ denote traditional fuzzy set, u denotes the control action $u\in U\subset IR$ under that rule structure, assuming that for each input e_{i} and Δe_{i} , M labels are designed respectively. As more sensors are used in space domain, then more spatial information is processed. So the fuzzy rules expand exponentially.

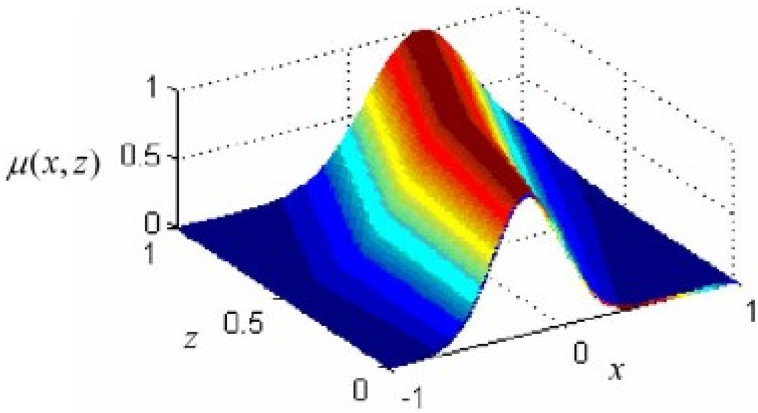


Figure 4.16 - 3D fuzzy set [16]

The 3D fuzzy control system is able to capture and process the spatial domain information defined as the 3D FC. One of the essential elements of this type of fuzzy system is the 3D fuzzy set used for modeling the 3D uncertainty. A 3D fuzzy set is introduced in Figure 4.16 by developing a third dimension for spatial information from the traditional fuzzy set. The 3D fuzzy set defined on the universe of discourse X and on the one-dimensional space is given by:

$$\bar{V} = \{((x,z),\mu_{\bar{V}}(x,z)) \mid \forall x \in X, \forall z \in Z\}$$ And $$0 \le \{(x,z),\mu_{\bar{V}}(x,z)\} \le 1 \quad (4.13)$$

When X and Z are discrete, $\bar{V}$ is commonly written as

$\bar{V} = \sum_{z \in Z} \sum_{x \in X} \mu_{\bar{V}}(x,z)/(x,z)$ Where $\sum \sum$ denotes union over all admissible x and z .

Using this 3D fuzzy set, a 3D fuzzy membership function (3D MSF) is developed to describe a relationship between input x and the spatial variable z with the fuzzy grade u .

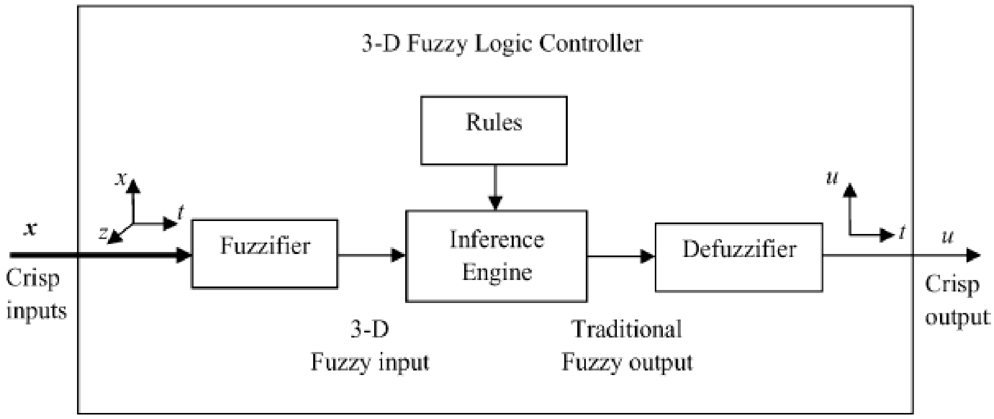


Figure 4.17 - 3D fuzzy system [16].

Theoretically, the 3D fuzzy set or 3D global fuzzy MSF is the assembly of 2D traditional fuzzy sets at every spatial location [16]. However, the complexity of this global 3D nature may cause difficulty in developing the FC. Practically, this 3D fuzzy MSF is approximately constructed by 2D fuzzy MSF at each sensing location [16]. Thus, a centralized rule based is more appropriate, which avoid the exponential explosion of rules when sensors increase. The new FC has the same basic structure as the traditional one. The 3D FC is composed of fuzzification, rule inference and defuzzification as shown in Figure 4.17. Due to its unique 3D nature, some detailed operations of this new FC are different from the traditional one. Crisp inputs from the space domain are first transformed into one 3D fuzzy input via the 3D global fuzzy MSF. This 3D fuzzy input goes through the spatial information fusion and dimension reduction to become a traditional 2D fuzzy input. After that, a traditional fuzzy inference is carried out with a crisp output produced from the traditional defuzzification operation. Similar to the traditional 2D FC, there are two different fuzzifications: singleton fuzzifier and non-singleton.

A singleton fuzzifier is selected as follows: Let $\overline{A}$ be a 3D fuzzy set, x is a crisp input, $x \in X$ and z is a point $z \in Z$ in one-dimensional space Z. The singleton fuzzifier maps x into $\overline{A}$ in X at location z then $\overline{A}$ s a fuzzy singleton with support x' if $\mu_{\overline{A}}(x,z)=1$ for $x = x'$, $z = z'$ and $\mu_{\overline{A}}(x,z)=0$ for all other $x \in X$, $z \in Z$ with $x \neq x'$, $z \neq z'$ if finite sensors are used. This 3D fuzzification is considered as the assembly of the traditional 2D fuzzification at each sensing location. Therefore, for p discrete measurement sensors located at $z_1, z_2, \ldots\ldots, z_p$ shown in Figure 4.15, $x_z = [x_1(z), x_2(z), \ldots, x_j(z)]$ is defined as J crisp spatial input variables in space domain $Z = \{z_1, z_2, \ldots\ldots, z_p\}$ where $x_j(z_i) \in X_j \subset IR(j = 1,2,\ldots,J)$ denotes the crisp input at the measurement location $z = z_i$ for the spatial input variable $x_j(z)$, X_j denotes the domain of $x_j(z_i)$. The variable $x_j(z)$ is marked by " z " to distinguish from the ordinary input variable, indicating that it is a spatial input variable. The fuzzification for each crisp spatial input variable $x_j(z)$ is uniformly expressed as one 3D fuzzy input $\overline{A}_{xj}$ in the discrete form as follows:

$$\overline{A}_{X1} = \sum_{z \in Z} \sum_{x_1(z) \in X_1} \mu_{X1}(x_1(z), z)/(x_1(z), z)$$

:

$$\overline{A}_{Xj} = \sum_{z \in Z} \sum_{x_j(z) \in X_j} \mu_{Xj}(x_j(z), z)/(x_j(z), z)$$

:

$$\overline{A}_{XJ} = \sum_{z \in Z} \sum_{x_J(z) \in X_J} \mu_{XJ}(x_J(z), z)/(x_J(z), z) \qquad (4.14)$$

Then, the fuzzification result of J crisp inputs x_z can be represented by

[illegible]

(4.15)

Where * denotes the triangular norm; t-norm (for short) is a binary operation. The t-norm operation is equivalent to logical AND. Also it has been assumed that the membership function $\mu_{\overline{A}x}$ is separable [92].

Using the 3D fuzzy set, the γ^{th} rule in the rule based is expressed as follows:

$$\overline{R}^{\gamma} : if \ x_1(z) \, is \ \overline{C}_1^{\gamma} and .... and \ x_J(z) \, is \ C_J^{\gamma} then \, u \ is \ G^{\gamma}$$

(4.16)

Where $\bar{R}^{\gamma}$denotes the γ^{th} rule $\gamma = (1,2,....,N)$ $x_j(z),(j = 1,2,...,J)$ denotes spatial input variable C_J^{γ} denotes 3D fuzzy set, u denotes the control action $u \in U \subset R$, $G\gamma$ denotes a traditional fuzzy set N is the number of fuzzy rules, the inference engine of the 3D FC is expected to transform a 3D fuzzy input into a traditional fuzzy output. Thus, the inference engine has the ability to cope with spatial information. The 3D fuzzy DTM controller is designed to have three operations: spatial information fusion, dimension reduction, and traditional inference operation as shown in Figure 4.18. The inference process is about the operation of 3D fuzzy set including union, intersection and complement operation. Considering the fuzzy rule expressed as the (4.16), the rule presents a fuzzy relation $\bar{R}^{\gamma} : \bar{C}_1^{\gamma} \times ......\times C_J^{\gamma} \longrightarrow G\gamma$ $\gamma = (1,2,....,N)$ thus, a traditional fuzzy set is generated via combining the 3D fuzzy input and the fuzzy relation is represented by rules.

The spatial information fusion is this first operation in the inference to transform the 3D fuzzy input $\bar{A}_X$ into a 3D set $W\gamma$ appearing as a 2D fuzzy spatial distribution at each input x_z . $W\gamma$ is defined by an extended sup-star composition on the input set and antecedent set. Figure 4.19 gives a demonstration of spatial information fusion in the case of two crisp inputs from the space domain Z , $x_z = [x_1(z), x_2(z),..., x_j(z)]$.

This spatial 3D MSF, is produced by the extended sup-star operation on two input sets from singleton fuzzification and two antecedent sets in a discrete space Z at each input value x_z . An extended sup-star [93] composition employed on the input set and antecedent sets of the rule, is denoted by

$$W_{\gamma\,\bar{A}_{X}\circ(\bar{C}_1^{\gamma}\times...\times\bar{C}_J^{\gamma})} : \bar{A}_{X}\circ\left(\bar{C}_1^{\gamma}\times...\times\bar{C}_J^{\gamma}\right) \tag{4.17}$$

The grade of the 3D MSF derived as $\mu_{\overline{W^{\gamma}}}(z) = \mu_{\overline{A}X^{o}(\overline{C}_1^{\gamma} \times .. \times \overline{C}_J^{\gamma})}(x_z, z)$ (4.18)

$\mu_{\overline{W_{\gamma}}}(z) = \sup_{x_1(z)\in X_1,......,x_J(z)\in X_J}[\mu_{\overline{A}X}(x_z, z) * \mu_{\overline{C}_1^{\gamma}} \times ... \times \mu_{\overline{C}_J^{\gamma}}(x_z, z)]$ Where $z \in Z$ and

* denotes the t-norm operation.

$$\mu_{\overline{W^{\gamma}}}(z) = \sup_{x_1(z)\in X_1,......,x_J(z)\in X_J}[\mu_{\overline{A}X1}(x_1(z), z) * ...... * \mu_{\overline{A}XJ}(x_J(z), z) * \mu_{\overline{C}_1^{\gamma}}(x_1(z), z) * ...$$
$$.. * \mu_{\overline{C}_1^{\gamma}}(x_1(z), z) * .... * \mu_{\overline{C}_J^{\gamma}}(x_J(z), z)]$$

$\mu_{\overline{W}^{\gamma}}(z) = \{\sup_{x_1(z)\in X_1}[\mu_{\overline{AX1}}(x_1(z), z) * \mu_{\overline{C}_1^{\gamma}}(x_1(z), z)]\} * ...... * \{\sup_{x_J(z)\in X_J}[\mu_{\overline{AXJ}}(x_J(z), z) * \mu_{\overline{C}_J^{\gamma}}(x_J(z), z)]\}$ The dimension reduction operation is to compress the spatial distribution information (x_z, μ, z) into 2D information (x_z, μ) as shown in Figure 4.19. The set $W\gamma$ shows an approximate fuzzy spatial distribution for each input x_z in which contains the physical information. The 3D set $W\gamma$ is simply regarded as a 2D spatial MSF on the plane (μ, z) for each input x_z .

Thus, the option to compress this 3D set $W\gamma$ into a 2D set $\phi\gamma$ is approximately described as the overall impact of the spatial distribution with respect to the input x_z .The traditional inference operation is the last operation in the inference. Where implication and rules' combination are similar to those in the traditional inference engine.

$$\mu_V{}^{\gamma}(u) = \phi\gamma * \mu_{G\gamma}(u), u \in U \qquad (4.19)$$

Where * stands for a t-norm, $\mu_{G\gamma}(u)$ is the membership grade of the consequent set of the fired rule $\overline{R}^{\gamma}$

Finally, the inference engine combines all the fired rules (4.20) .Where $V_\gamma$the output is fuzzy set of the fired rule $\overline{R}^\gamma$, N' denotes the number of fired rules and V denotes the composite output fuzzy set.

$$V = \square_{\gamma=1}^{N'} V_\gamma \tag{4.20}$$

The traditional defuzzification is used to produce a crisp output. The center of area (COA) is chosen as the defuzzifier due to its simple computation [87].

$$u = \frac{\sum_{\gamma=1}^{N'} C_\gamma \mu\phi_\gamma}{\sum_{\gamma=1}^{N'} \mu\phi_\gamma} \tag{4.21}$$

Where $C_\gamma \in U$ is the centroid of the consequent set of the fired rule $\overline{R}^\gamma$ $\gamma = (1,2,\ldots, N')$ which represents the consequent set G_γ in (4.19), N' is the number of fire rules $N' \leq N$

For Multi-Core CPU system; each core is considered as heat source. The heat condiction Q pation is inverse propositional to the distance between the heat sources (4.22) [89]. The nearest hotspot has the highest effect on core temperature increase. Also the far hotspot has the lowest effect on core temperature increase.

$$Q = \frac{\sigma \quad A \quad \Delta T}{d} \tag{4.22}$$

Where Q is the heat conducted, σ the thermal conductivity, A the cross-section area of of heat path (costant value), ΔT the temperature difference at the hotspots locations, d the length of heat path (the distance between the heat sources)

Thus the 3D FC is taking into considerations the correlation between other cores operating frequency and temperature as follow:

The 3D MSF gain G_{ij} is selected as the inverse the distance between 2 cores hotspots locations

$$MSF_{3D} = \sum MSF_{2D}\, G_{ij} \tag{4.23}$$

Where MSF_{2D} the 2D MSF, G_{ij} the correlation gains between core i and core j. G_{ij} is not a constant value as the hotspots locations are changing during the run time. The maximum gain = 1 in case of calculating the correlation gain locally G_{ii} .

The 3D FC is based on 32 variables as follow [87]:

The inputs 3D fuzzy variable at step n for each core are: 8 frequency deviation variables calculate as per (4.7), 8 temperature deviation variables calculate as per (4.10), 8 frequency deviation change variables calculate as per (4.11), and 8 temperature deviation change variables calculate as per (4.12). The output: for each core, the output is the core operating frequency at step n+1. The relationships: at step n CPU throughput is proportional to cores operating frequency. The core operating frequency is also proportional to the power consumption. The maximum power consumption leads to the maximum temperature increase.

In order to compare between the 2D FC and the 3D FC responses, the same configuration describe in section 4.4.2 as reused with the 3D FC as follow: the same the control objectives in section 4.4.2. The same fuzzy inputs as per Table 4.1, the same Meta decisions rules section 4.4.2, the same rule space , the same input 2D MSF Normal distribution configurations as per Table 4.2.

Also The output membership functions are tuned per DTM controller. Chapter 5 discusses these outputs MSF in details. In general we have four outputs MSF: Max - DVFS - TSC MSF - FS. Thus the only design different between the 2D FC and the 3D FC that the 3D FC DTM takes into consideration the surrounding core hotspot temperatures and their operating frequencies. Figure 4.20 shows the 3D fuzzy DTM controller implementation.

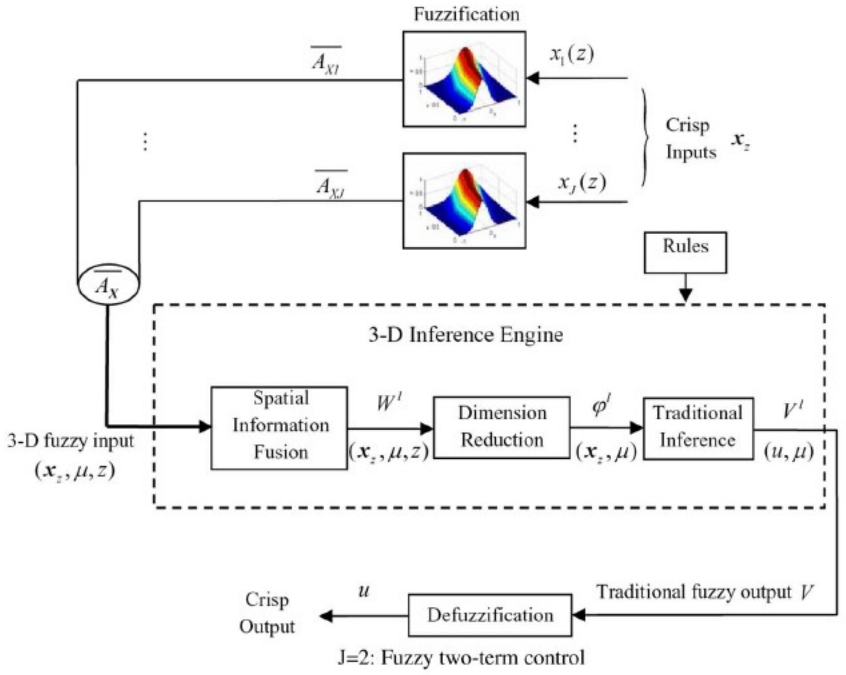


Figure 4.18 - the 3D-FC operation details [16]

Example

As shown in Figure 4.18; the number of p sensors = 5 ; the sensors are located at $z_1, z_2, \ldots\ldots, z_5$ Two crisp input, $x \in X$ and Z is a point $z \in Z$ in one-dimensional space . For $p = 5$ discrete measurement sensors located at $z_1, z_2, \ldots\ldots, z_5$ shown in Figure 4.19, $x_z = [x_1(z), x_2(z)]$ is defined as J is two crisp spatial input variables in space domain $Z = \{z_1, z_2, \ldots\ldots, z_5\}$ where $x_j(z_i) \in X_j \subset IR(j = 1,2)$. The fuzzification for each crisp spatial input variable $x_j(z)$ is uniformly expressed as the 3D fuzzy inputs are $\overline{A}_{x1}$ and $\overline{A}_{x2}$ in the discrete form. As shown in Figure 4.19; μ_1 values are the local substitutions of $x_1(z)$ in each 2D MSF at each z location. μ_2 values are the local substitutions of $x_2(z)$ in each 2D MSF at each z location . μ_{w^1} values are the sup-star composition of μ_1 and μ_2 at each z location as shown in The Table 4.3. The sup-star composition in the fuzzy inference engine becomes a sup-minimum composition.

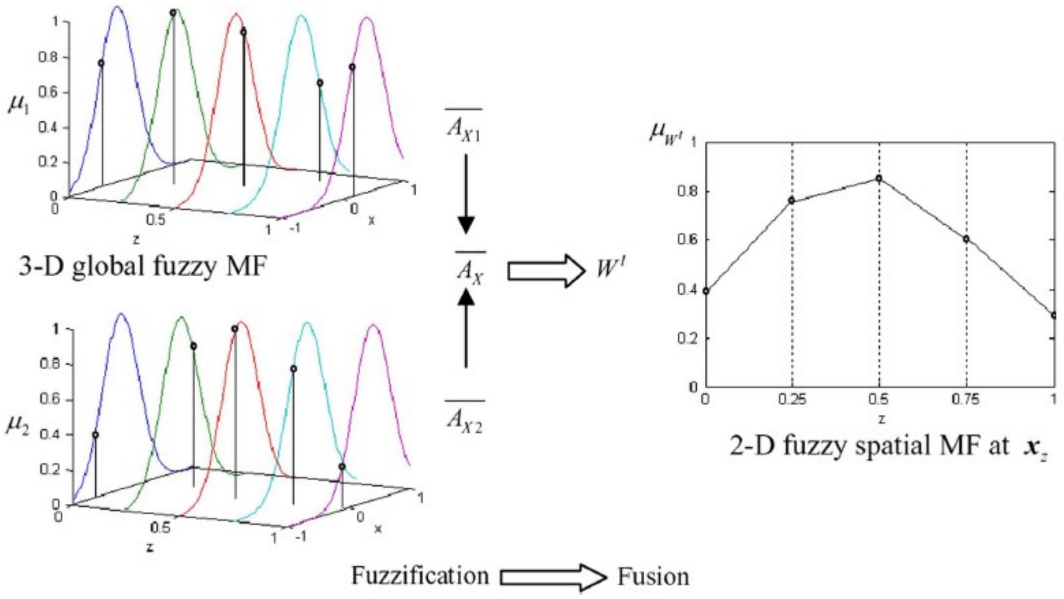


Figure 4.19 - Spatial information fusion at each crisp input x_z [16]

Table 4.3 - 3D Fuzzy with Two crips input example

$x_1(z)$	$x_2(z)$	z	μ_1	μ_2	μ_{w^1}
- 0.5	- 0.6	0.0	0.8	**0.4**	0.4
0.0	0.2	0.5	**0.8**	0.9	0.8
0.3	0.1	0.25	**0.9**	1	0.9
0.7	0	0.75	**0.6**	0.7	0.6
0.2	-0.1	1	0.8	**0.3**	0.3

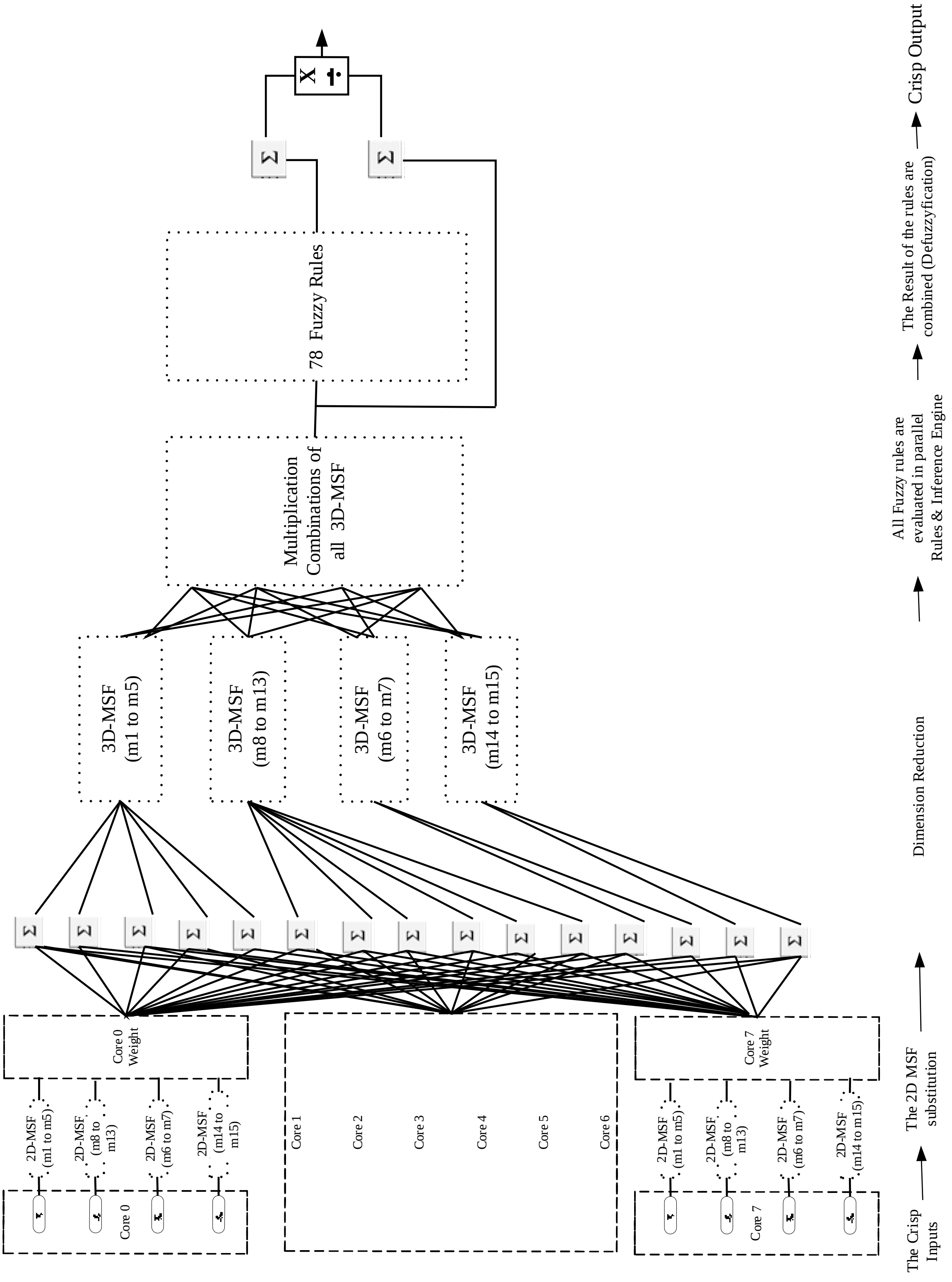
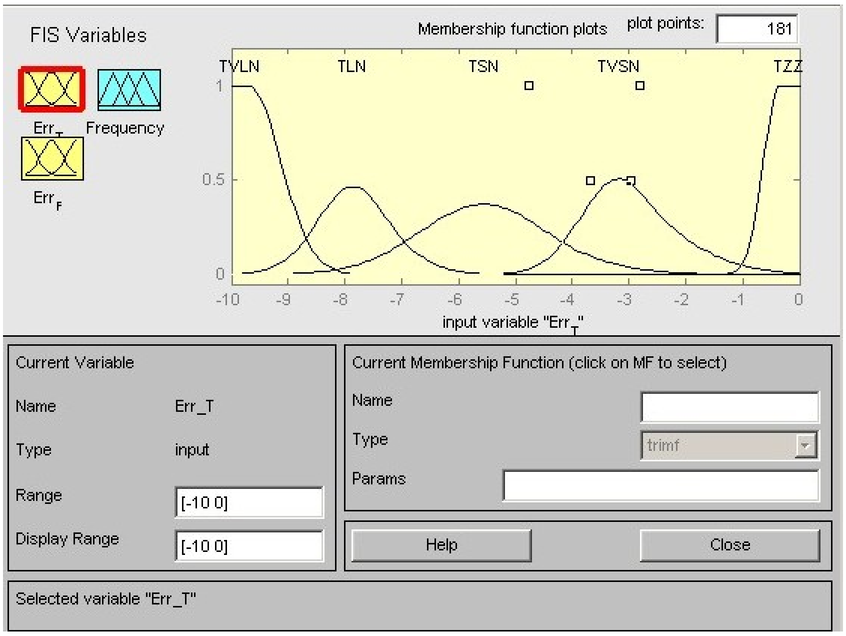


Figure 4.20 3D-Fuzzy DTM controller of a single Core “C0”

4.4.4 The Validation of the Fuzzy DTM Rules

The MATLAB fuzzy tool box is used for validating the designed fuzzy DTM controller before the integration with the thermal model. The fuzzy rules are the same as the Meta decisions rules discussed in section 4.4.2. For simplicity; the temperature deviation and the frequency deviation rules are tested by MATLAB without correlation between the cores. The fuzzy logic (FL) engine inputs are the core operating frequency deviation (4.7) and the core operating temperature deviation (4.10). The fuzzy membership functions (MSF) are shown in Figure 4.21, Figure 4.22, and Figure 4.23. All fuzzy DTM membership functions have Normal distribution the same as section 4.4.2. The temperature deviation is represented by 5 MSF. The frequency deviation is represented by 6 MSF. The output deviation is represented by 4 MSF. The centroid algorithm is used for defuzzification.



plot points: 181

Current Variable

Name Err_T

Type input

Range [-10 0]

Display Range [-10 0]

Current Membership Function (click on MF to select)

Name

Type trimf

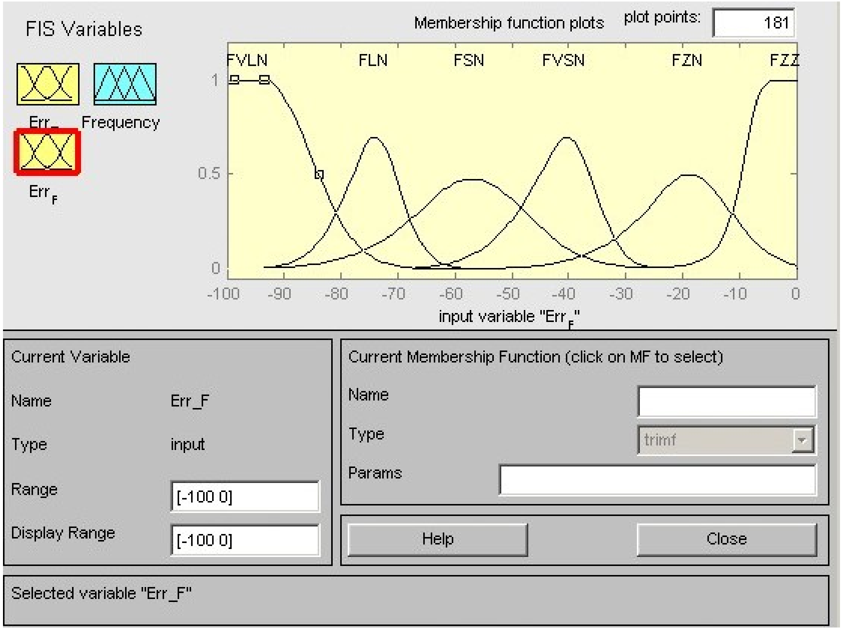
Params

Selected variable "Err_T"

Figure 4.21 - the temperature deviation membership functions.

As shown in Figure 4.24 the temperature deviation variable T_e and the frequency deviation f_e variables are generated using MATLAB random Gaussian generator. The fuzzy DTM controller output is shown in Figure 4.25. The lower graph shows random frequency deviation f_e. The middle graph shows the temperature deviation T_e. The upper graph shows the fuzzy DTM controller response. If the value of DTM controller output is lower than 4 the core runs at the maximum operating frequency. But if the DTM controller output is between 4 and 8 then DTM applies the DVFS technique. Thus the core runs at variable operating frequency. If the DTM controller output is between 8 and 9 the TSC technique is triggered. This core is offloaded and its tasks are moved to the other cores. If the DTM controller output is higher than 9; the core fails safe “FS” to avoid hardware fault.

The DTM controller output is following the Meta fuzzy rules. At the core low temperatures (the temperature deviation T_e between -10 and -8); the core runs at the maximum operating frequency independent on the frequency deviation. At the maximum core operating temperature (the temperature deviation higher than -1); the core fails safe to avoid hardware fault independent on frequency deviation. At the core temperature deviation between -8 and -1; the DTM controller selects the DVFS or the TSC depending on the core frequency deviation values. In case of the core frequency deviation between -20 and -60; the TSC technique is triggered. But if the core frequency deviation lower than -60; the DVFS technique is triggered as shown in Figure 4.25. Thus the fuzzy controller output is according to the desired design of DTM controller objectives.



Current Variable

Name Err_F

Type input

Range [-100 0]

Display Range [-100 0]

Current Membership Function (click on MF to select)

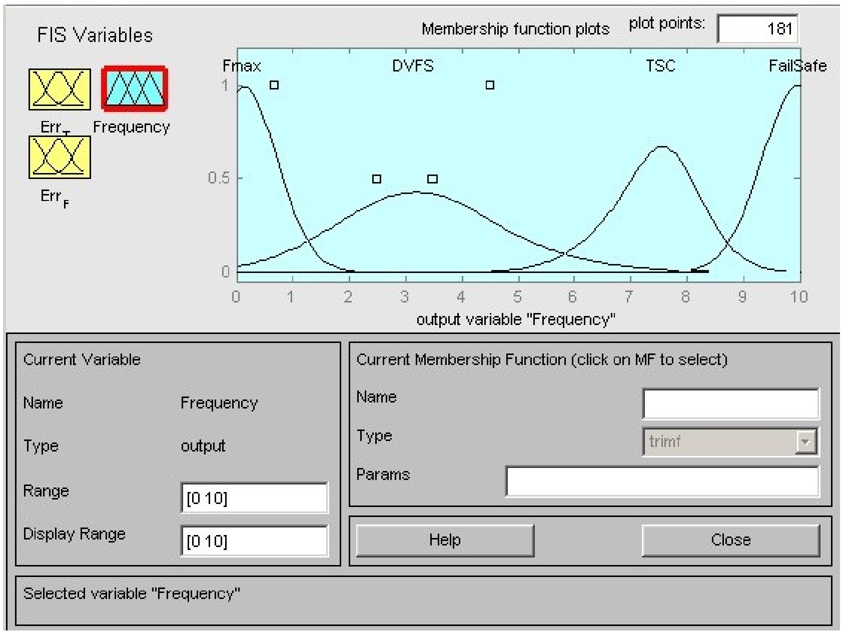
Name

Type trimf

Params

Selected variable "Err_F"

Figure 4.22 - the frequency deviation membership functions.



Current Variable

Name Frequency

Type output

Range [0 10]

Display Range [0 10]

Current Membership Function (click on MF to select)

Name

Type trimf

Params

Selected variable "Frequency"

Figure 4.23 - the output membership functions.

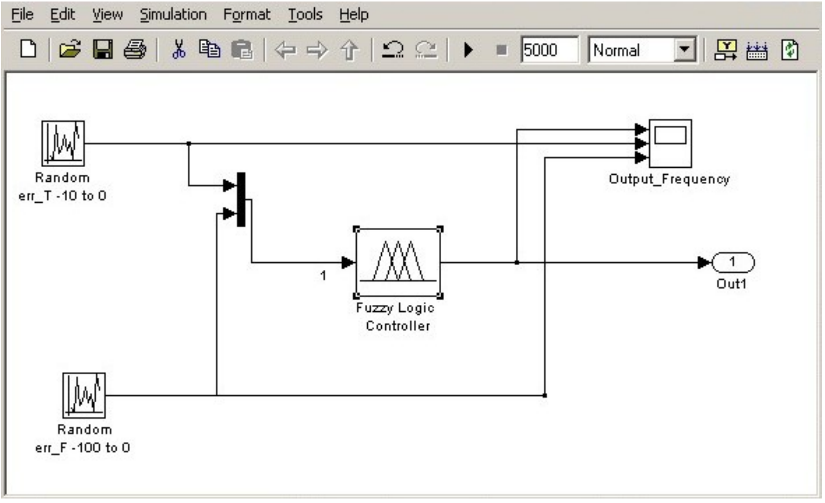


Figure 4.24 - MATLAB validation of the fuzzy DTM controller.

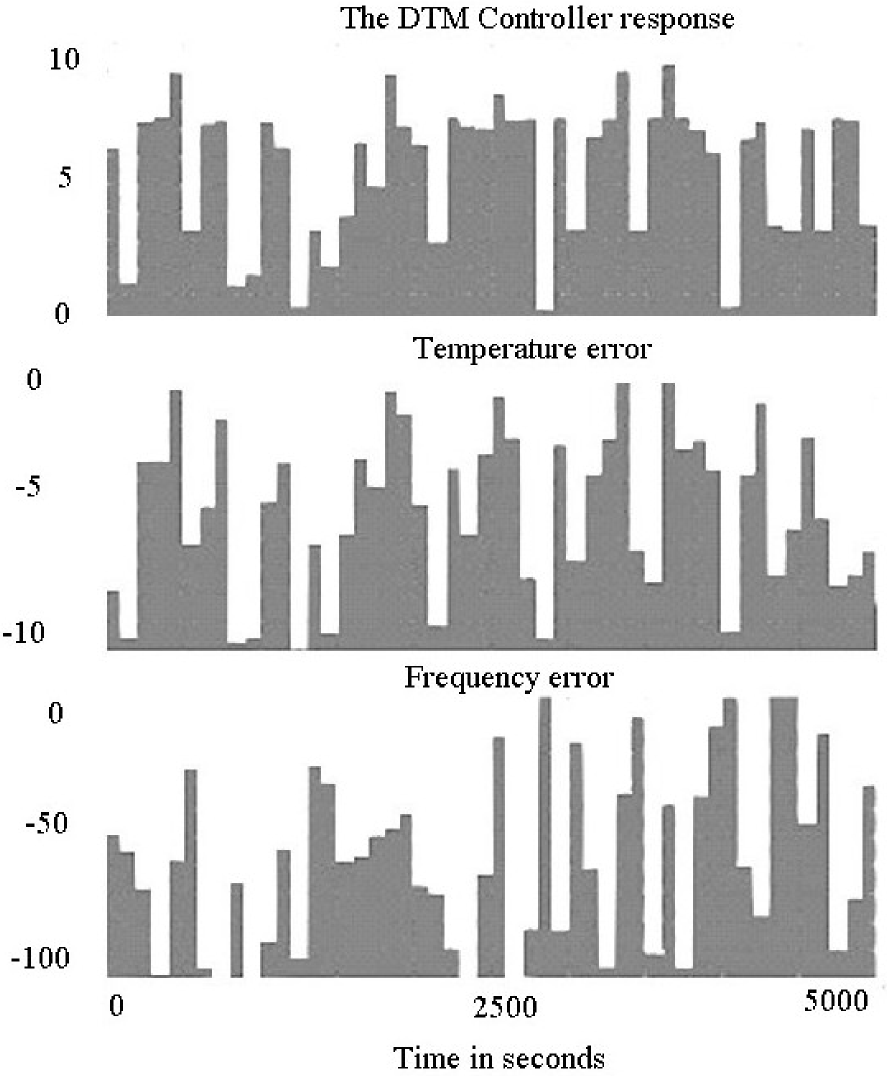


Figure 4.25 - the fuzzy DTM controller response

4.5 Conclusion

At least portable devices are still using the air cooling. The air cooling system already reaches its maximum limitation “198 Watt”. The advanced dynamic thermal management techniques are mandatory to avoid the CPU thermal throttling. According to Amdahl’s law: “parallel speedups limited by serial portions” [83]; thus the CPU is not 100% utilized all time. The thermal spare cores (TSC) is based on reserving cores during low CPU utilization and activate them during thermal crises. TSC is a novel dynamic thermal management (DTM) tool that requires more examinations. The reservation of some cores as (TSC) doesn’t impact CPU over all utilization. As these cores are not activate simultaneously due to limitations. The semiconductor technology permits more cores to be added to CPU chip. While the total chip area overhead is up to 27.9 % as per ITRS [82]. That means there is no chip area wasting in case of TSC. From the thermal point of view; the horizontal heat transfer path has up to 30% of CPU chip heat transfer [84]. The TSC is a big coldspot within the CPU area that handles the horizontal heat transfer path. The cold TSC also handles the static power as the TSC core is turned off. The TSC is used simultaneous with other DTM technique. Fuzzy logic can improve the DTM controller response. Fuzzy control handles the CPU thermal process without known its transfer function. This simplifies the DTM controller design and reduces design time. The 3D fuzzy DTM takes into consideration the surrounding core hotspot temperatures and operating frequencies. The 3D fuzzy DTM avoids the complexity and maintains the correlations. The next Chapter presents the integration details between the Multi-Core CPU thermal model and the fuzzy DTM controller. Also the simulation results and analysis are presented in Chapter 5.

Chapter 5

The Implementation of DTM Fuzzy Control

5.1 Introduction

This chapter discusses the integration between the thermal simulator “Hotspot 4.1” and the DTM controller. As shown in chapter 4, different DTM control designs are presented. These fuzzy controllers (FCs) are evaluated using simulation. The chapter describes the integration between the thermal model and the DTM controllers. Many simulation tests are presented. The comparison between these DTM designs is made using the DTM evaluation index, the CPU temperature profile and the frequency change profile. Finally the analysis of these simulation test results is presented.

5.2 The Thermal Simulator Integration with the Controller

5.2.1 The Mediator Script

This section describes the “Mediator” shell script. This script integrates both the Hotspot 4.1 simulator and the DTM controller together within the same simulation loop. The Mediator script manages the information exchange between both the Hotspot software and the DTM controller. The Mediator script gathers the temperature from the Hotspot simulator. Then it provides this temperature information to the DTM controller as if there is temperature sensors placed on the CPU chip. The DTM controller selects the operating frequency of each core.

The DTM controller has a quantization subroutine. So the DTM selects discreet operating frequency. The DTM control sends the vector output to the Mediator. As shown in Figure 5.1, the Mediator script converts the core operating frequency into power consumed per function block. Then the mediator sends the power information to the Hotspot thermal simulator. The Hotspot sends the temperature to the Mediator so the simulator loop continues without manual intervention. There are 10 operating frequencies available for each core and 2 core switch off modes. The Mediator script manages the simulation as follow: first the Hotspot simulates from step 0 to step n. Then the Mediator applies control vector at n+1 for a period m by preparing the corresponding Hotspot power inputs. After that the Mediator triggers the simulation from step 0 to step n+m and so on. The mediator script supports predefine core initial temperature to minimize simulation time.

5.2.2 DTM Controllers

The basic DTM controller is a proportional controller. The basic DTM control flowchart Figure 5.2 shows simple control subroutine. The basic DTM controller performs DVFS as output control vector. As discussed in chapter 4, the enhanced DTM controller shown in Figure 5.3 is based on a P controller followed by a fuzzy controller. The fuzzy controller adjusts the P controller output control vector. The fuzzy controller output responses are tuned by the modification of the output membership functions (MSFs). Some 2D and 3D fuzzy controllers shares the same output MSFs i.e. FC1 & 3D-FC1, FC2 & 3D-FC2, FC3 & 3D-FC3. But some other 3D fuzzy have their own output MSFs i.e. FC4, 3D-FC4, 3D-FC5, and 3D-FC6. In total there are 11 different DTM responses extracted by tuning the DTM controllers output MSFs.

The simulation tests study the effect of performing the TSC technique. As some 2D and 3D fuzzy DTM perform TSC and DVFS i.e FC1, FC2, FC3, 3D-FC1, 3D-FC2, and 3D-FC3. The other fuzzy DTM perform DVFS only i.e. FC4, 3D-FC4, 3D-FC5, and 3D-FC6. In case of the 2D fuzzy DTM controller there is no correlation between the cores operating frequencies and the hotspot temperatures. But in case of the 3D fuzzy DTM controller, the output control vector takes into consideration the correlation between all cores operating frequencies and all cores hotspot temperatures. The 3D fuzzy controller flowchart Figure 5.4 presents the software implementation of the block diagram Figure 5.3. In general the fuzzy controller output is calculated using the fuzzy center of area [87].

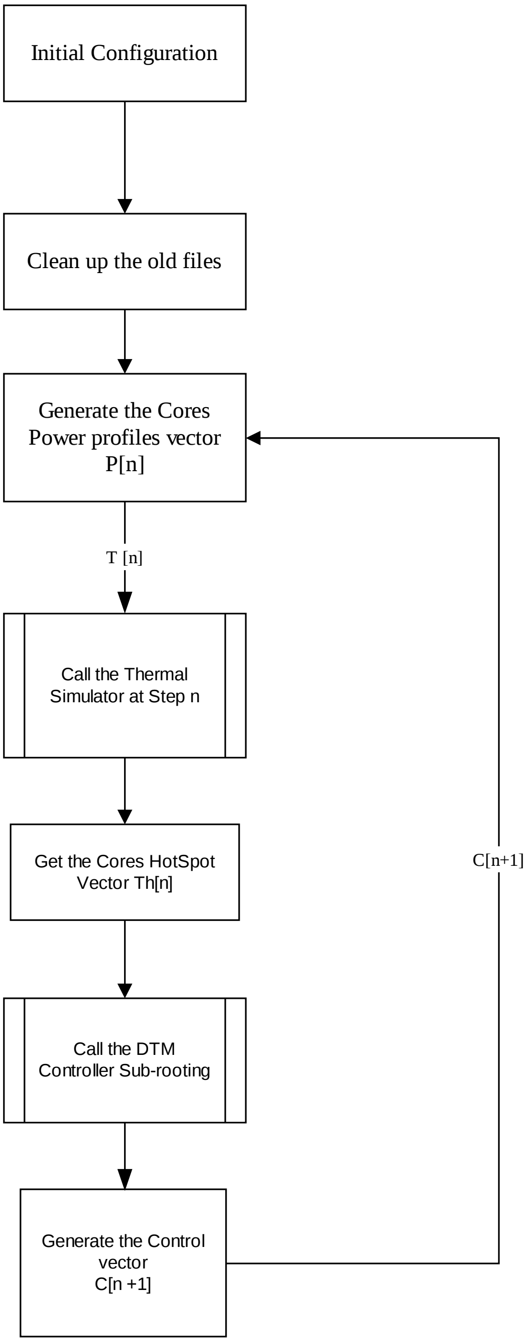

Initial Configuration
Clean up the old files
Generate the Cores Power profiles vector P[n]
T [n]
Call the Thermal Simulator at Step n
Get the Cores HotSpot Vector Th[n]
Call the DTM Controller Sub-rooting
Generate the Control vector C[n +1]
C[n+1]

Figure 5.1 - the Mediator script flowchart.

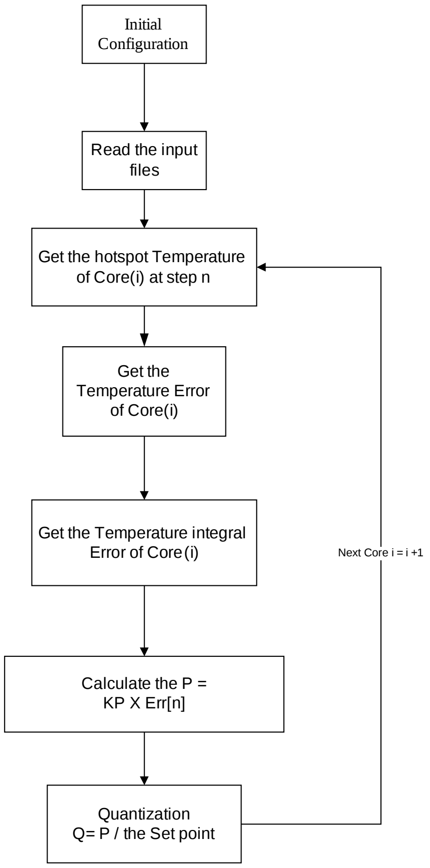

Initial Configuration
Read the input files
Get the hotspot Temperature of Core(i) at step n
Get the Temperature Error of Core(i)
Get the Temperature integral Error of Core (i)
Calculate the P = KP X Err[n]
Quantization Q= P / the Set point
Next Core i = i +1

Figure 5.2 - Basic DTM control flowchart.

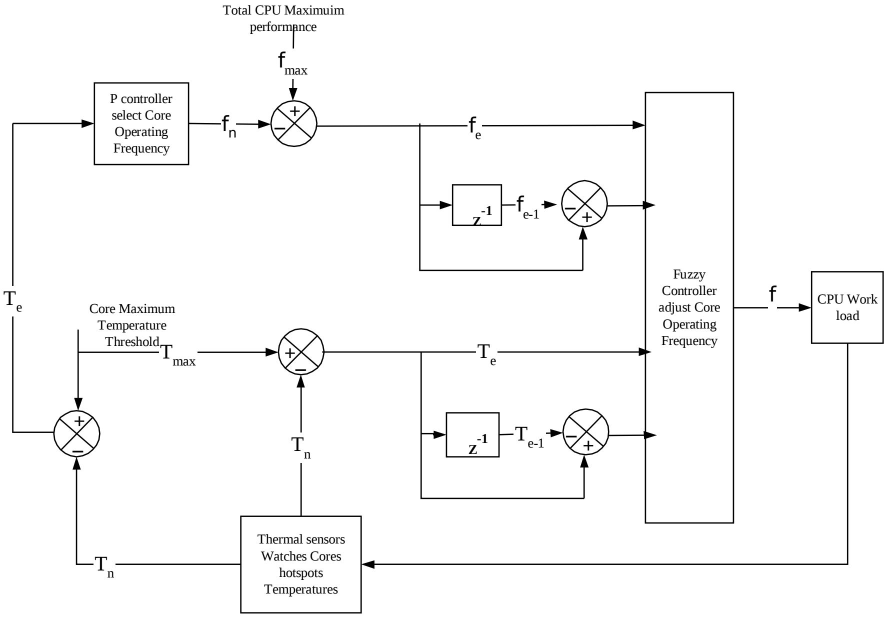

Total CPU Maximuim performance
f max
P controller select Core Operating Frequency
f n
+
–
f e
z -1
f e-1
–
+
T e
Core Maximum Temperature Threshold
T max
+
–
T e
z -1
T e-1
–
+
+
–
T n
Thermal sensors Watches Cores hotspots Temperatures
T n
Fuzzy Controller adjust Core Operating Frequency
f
CPU Work load

Figure 5.3 - General DTM fuzzy controller block diagram.

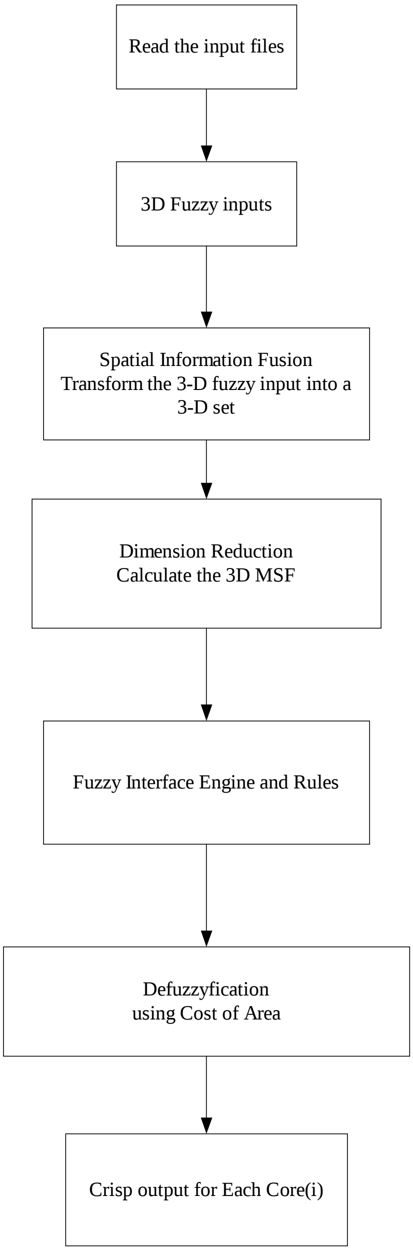

Read the input files
3D Fuzzy inputs
Spatial Information Fusion
Transform the 3-D fuzzy input into a 3-D set
Dimension Reduction
Calculate the 3D MSF
Fuzzy Interface Engine and Rules
Defuzzyfication
using Cost of Area
Crisp output for Each Core(i)

Figure 5.4 - 3D fuzzy controllers flowchart.

5.3 The Simulations Tests

The DTM controller design should fulfill the desired CPU behavior. The Fuzzy output membership functions selection should reflect the desired parameters values for each range $\sigma_{ij}^{\text{Desired}}$ equation (4.7). The thesis covers 2 different DTM design requirements and 11 DTM simulation tests. All these DTM simulation tests are evaluated against these 2 DTM designs as a proof of concept. The 11 DTM controller implementations show that no unique DTM controller design can fit into all DTM requirements. The first test covers the basic DTM P controller. The remaining tests cover the enhanced fuzzy DTM controllers. There are four 2D fuzzy tests and sixth 3D fuzzy tests as shown in Table 5.1. The same P controller is reused in all fuzzy DTM simulations. The same fuzzy rules discussed in chapter 4 are reused in the simulation tests. The 2D fuzzy has only 4 input variables. Temperature deviation is represented by 5 input membership functions [m1 to m5]. The temperature deviation change is represented by 2 input membership functions [m6 to m7]. The frequency deviation is represented by 6 input membership functions [m8 to m13]. The frequency deviation change is represented by 2 input membership functions [m14 to m15] as shown in Table 5.2. The 3D fuzzy has 32 input variables. There are 8 temperature deviation variables [m1 to m5], 8 temperature deviation change variables [m6 to m7], 8 frequency deviation variables [m8 to m13], and 8 frequency deviation change variables [m14 to m15]. The same input membership functions are reused with the 3D fuzzy as shown in Table 5.2. The output membership functions (MSF) selection is critical. The deviation from the accurate output MSF limits affects the DTM controller behavior. The selected Fuzzy output membership functions are normal “Gaussian” distribution as shown in Table 5.3.

The TSC technique is tested 6 times with the DVFS technique (2D & 3D FC1, 2D & 3D FC2, 2D & 3D FC3). The DVFS is evaluated alone in 4 tests (2D & 3D FC4, FC5, and FC6). The 2D and the 3D Fuzzy DTM controller implementation from MATLAB point of view is shown in Figure 5.5 and Figure 5.6 respectively. The main difference between Figure 5.5 and Figure 5.6 is the correlation calculation step beside the number of input variables. The 3D fuzzy gains are dynamic. The hotspot locations change due to temperature distributions change. Thus the 3D-FC calculates these gains at each simulation step. The Fuzzy output membership function design consists of 4 phases:

1- Identify the required parameters

Only the CPU frequency and the CPU temperature are the selected parameters due to the thermal simulator specifications. The hotspot thermal simulator considers the CPU consumed power is proportional to the CPU operating frequency.

2- Identify the design parameters ranges

The CPU frequency design parameter is divided into 4 ranges:

Maximum (M) – High (H) – Medium (m) – Low (L), these ranges are selected as follow: Maximum 100%, High “greater than 70% of the Maximum”, Medium “between 70% and 40% of the Maximum”, and Low “lower than 40% of the Maximum”.

The CPU temperature design parameter is divided into 3 ranges:

High (H)– Medium (m) – Low (L), these ranges are selected as follow: High “greater than78 °C”, Medium “between 74 °C and 78 °C”, and Low “lower than 72 °C”.

3- Identify the desired parameters values for each range $\sigma_{ij}^{\text{Desired}}$

The output membership functions are selected to reflect the desired parameters values for each range $\sigma_{ij}^{\text{Desired}}$

Table 5.1 Fuzzy controller and output MSF mapping

The Controller Name	The output MSF
FC1	MAX1 - DVFS1 - TSC1 - FS1
3D-FC1	MAX1 - DVFS1 - TSC1 - FS1
FC2	MAX2 - DVFS2 - TSC2 - FS2
3D-FC2	MAX2 - DVFS2 - TSC2 - FS2
FC3	MAX3 - DVFS3 - TSC3 - FS3
3D-FC3	MAX3 - DVFS3 - TSC3 - FS3
FC4	MAX4 - DVFS4 - FS4
3D-FC4	MAX4 - DVFS4 - FS4
3D-FC5	MAX5 - DVFS5 - FS5
3D-FC6	MAX6 - DVFS6 - FS6

Table 5.2 Input MSF Normal Distribution Configurations

The Input MSF	Mean	Standard Deviation
m1	0	0.5
m2	-2.5	-0.75

m3	-5	1
m4	-8	0.75
m5	-10	0.5
m6	-5	2
m7	5	2
m8	-100	5
m9	-80	75
m10	-60	10
m11	-40	75
m12	-20	10
m13	0	5
m14	-50	20
m15	50	20

The membership functions in Table 5.2 and Table 5.3 are selected as normal distributed functions based on the current DTM design as a proof of concept only. The simulation tests are performed after the selection of the output membership functions. The DTM controller responses are compared using the DTM evaluation index ζ_t as follow:

1- Identify the actual parameters values for each range σ_{ij}^{Actual}

2- Evaluate each parameter and the over all multi- parameter evaluation index ζ_t equation (4.8)

Table 5.3 Output Membership Functions

MAX		
The output MSF	Mean	Standard Deviation
MAX1 = MAX2 = MAX4 = MAX5 = MAX6	0	0.05
MAX3	0.2	0.1
DVFS		
The output MSF	Mean	Standard Deviation

DVFS1	0.35	0.1
DVFS2	0.35	0.1
DVFS3	0.6	0.09
DVFS4	0.5	0.2
DVFS5	0.6	0.3
DVFS6	0.4	0.3
TSC		
The output MSF	Mean	Standard Deviation
TSC1	0.75	0.075
TSC2	0.7	0.06
TSC3	0.78	0.06
Fail Safely		
The output MSF	Mean	Standard Deviation
FS1 = FS2 = FS3 = FS4 = FS5 = FS6	1	0.05

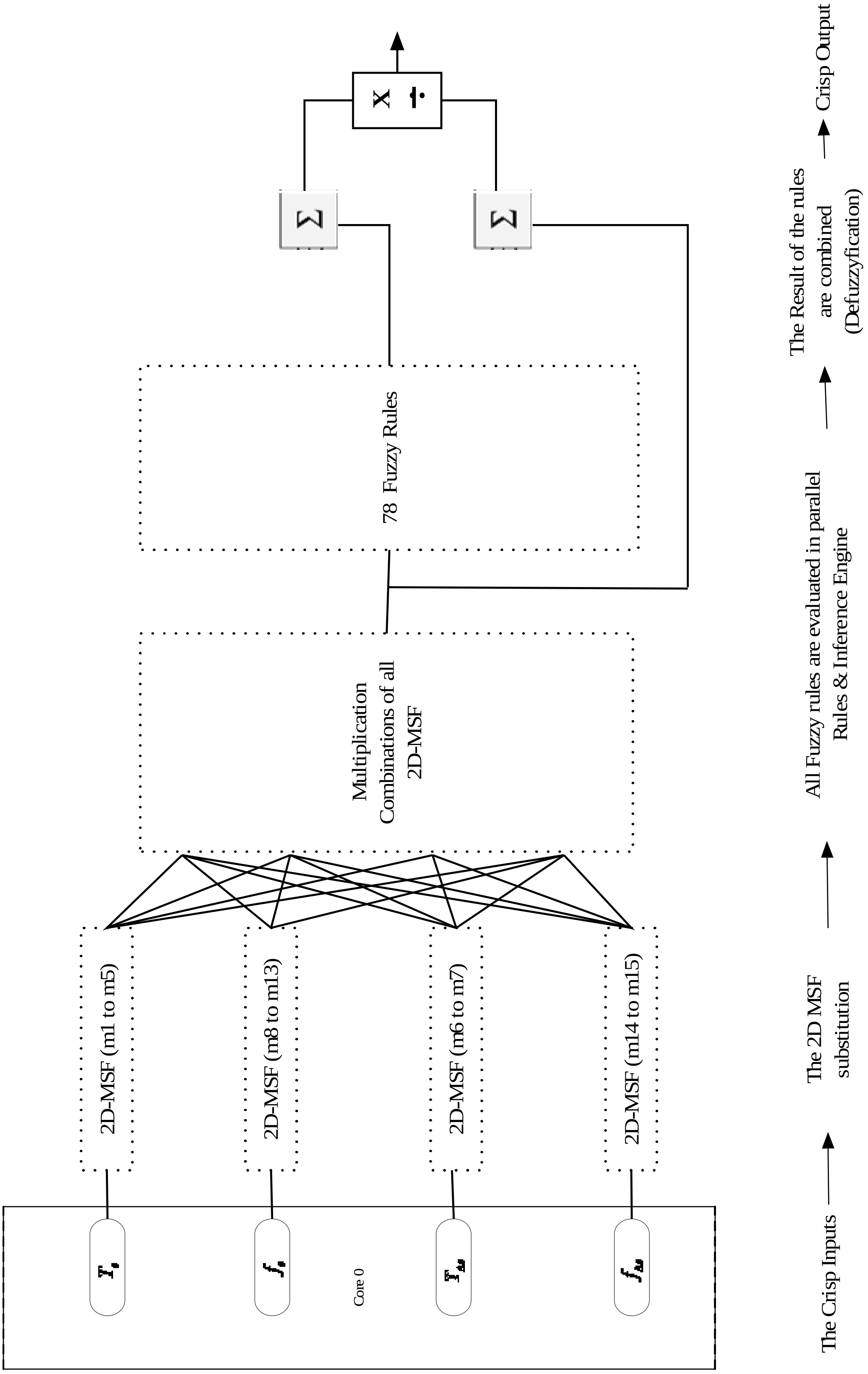

T
f
Core 0
T
f
2D-MSF (m1 to m5)
2D-MSF (m8 to m13)
2D-MSF (m6 to m7)
2D-MSF (m14 to m15)
Multiplication Combinations of all 2D-MSF
78 Fuzzy Rules
Σ
Σ
X
÷
The Crisp Inputs
The 2D MSF substitution
All Fuzzy rules are evaluated in parallel Rules & Inference Engine
The Result of the rules are combined (Defuzzyfication)
Crisp Output

Figure 5.5 2D-fuzzy DTM of a single Core “C0”

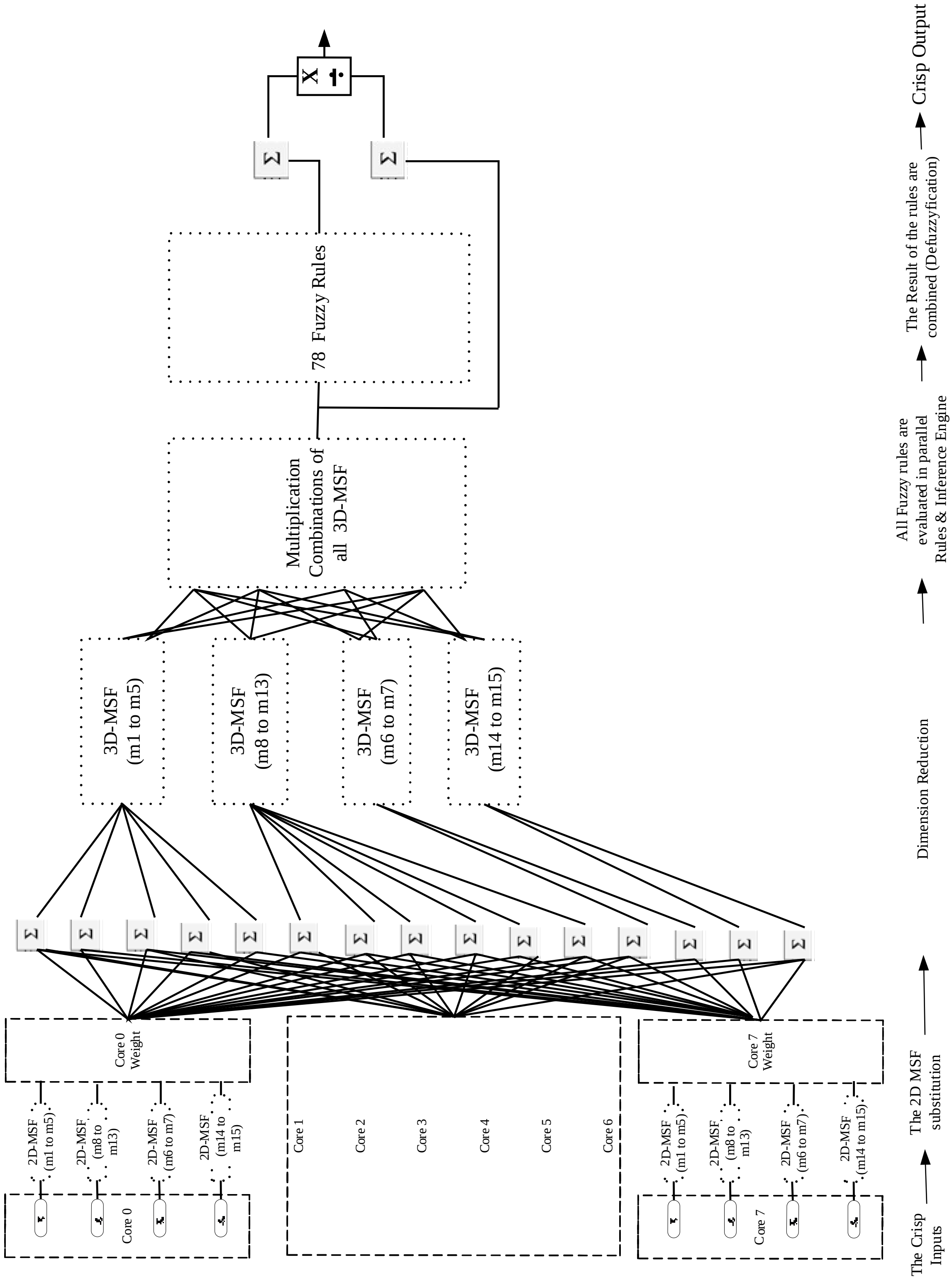


Figure 5.6 3D-Fuzzy DTM controller of a single Core “C0”

0

5.4 The Simulation Analysis

All simulations starts from 860 seconds as the CPU thermal model required 860 seconds to reach $T_{Control}$ 70 °C. Assuming that the CPU output response follows the open loop curve until it reaches 70 °C. At $T_{Control}$, the DTM controller output selects the cores operating frequency. Then each core temperature changes according to its operating frequency. All DTM fuzzy designs tuning are based on their output membership functions (MSF) tuning without changing the fuzzy rules. The DTM evaluation index covers the simulation times between 860 seconds to 1060 seconds. Theses simulation tests 3D-FC1, FC1, 3D-FC2, FC2, 3D-FC3 and FC3 perform both DVFS and TSC together. But these tests FC4, 3D-FC4, 3D-FC5, and 3D-FC6 perform DVFS only.The DTM controller evaluation index (4.8) has only two parameters $l = 2$, the frequency and the temperature. Its desired value is $\zeta_t = 2$ or near 2. Each evaluation parameter λ is evaluated over a normalized time period. The λ value should be 1 or near 1. If $\lambda < 1$ then the CPU runs less time within this range than the desired. If $\lambda > 1$ then the CPU runs more time within this range than the desired. The DTM evaluation index ζ_t calculation consists of 5 phases: 1- Identify the required parameters (frequency and temperature). 2- Identify the design parameters ranges. 3- Identify the desired parameters values for each range $\sigma_{ij}^{\text{Desired}}$. 4- Identify the actual parameters values for each range $\sigma_{\text{ij}}^{\text{Actual}}$. 5- Evaluate each parameter and the over all multi- parameter evaluation index (4.8)

$$\zeta_t = \sum_{i=1}^{l} \frac{1}{m_i} \sum_{j=1}^{m_i} \left(\frac{\sigma_{\text{ij}}^{\text{Actual}}}{\sigma_{ij}^{\text{Desired}}} \right)$$

There are two DTM evaluation index implementations presented in this section. The first DTM implementation assumed that the CPU is required to run 20% of its time at the maximum frequency, 50% of its time at high frequency, 20% of its time at medium frequency and 10% of it is time at low frequency. Also the CPU is required to 30% of its time at high temperature, 40% at medium temperature, and 30% of its time at low temperature. This first DTM requirement evaluation against the DTM controller designs are as follow:

Table 5.4 The frequency comparisons of the first implementation

Controller Name	**Frequency Ranges %** $\sigma_{1j}^{\text{Actual}}$				**Frequency Ranges Values** σ_{1j}				λ_1
	(M) j=1	**(H)** j=2	**(m)** j=3	**(L)** j=4	**(M)** j=1	**(H)** j=2	**(m)** j=3	**(L)** j=4	
$\sigma_{1j}^{\text{Desired}}$	*20%*	*50%*	*20%*	*10%*	*1.0*	*1.0*	*1.0*	*1.0*	*1.00*
Switch	0%	100%	0%	0%	0	2	0%	0	0.500
P	56%	0%	22%	22%	2.7	0.0	1	2	1.528
FC1	12%	22%	44%	22%	0.5	0.4	2	2.2	1.361
3D-FC1	0%	56%	33%	11%	0.0	1.1	1.7	1.1	**0.972**
FC2	0%	100%	0%	0%	0.0	2.0	0.0	0.0	0.500
3D-FC2	0%	89%	11%	0%	0.0	1.8	0.6	0.0	0.583
FC3	22%	22%	56%	0%	1.1	0.4	2.8	0.0	1.083
3D-FC3	0%	78%	22%	0%	0.0	1.6	1.1	0.0	0.667
FC4	0%	66%	33%	0%	0.0	1.3	1.7	0.0	0.750
3D-FC4	22%	56%	22%	0%	1.1	1.1	1.1	0.0	0.833
3D-FC5	0%	56%	33%	11%	0.0	1.1	1.7	1.1	**0.972**
3D-FC6	0%	78%	0%	22%	0.0	1.6	0.0	2.2	**0.944**

Table 5.5 The temperature comparisons of the first implementation

Controller Name	**Temperature Ranges %** $\sigma_{2j}^{\text{Actual}}$			**Temperature Ranges Values** σ_{2j}			λ_2
	(H) j=1	**(m) j=2**	**(L) j=3**	**(H) j=1**	**(m) j=2**	**(L) j=3**	
$\sigma_{2j}^{\text{Desired}}$	*30%*	*40%*	*30%*	*1.0*	*1.0*	*1.0*	*1.00*
Switch	0.0%	100%	0.0%	0.0	2.5	0.0	0.83
P	78%	0%	22%	2.6	0.0	0.7	1.11
FC1	11%	89%	0%	0.4	2.2	0.0	0.86
3D-FC1	22%	78%	0%	0.7	1.9	0.0	0.90
FC2	67%	33%	0%	2.2	0.8	0.0	1.02
3D-FC2	56%	44%	0%	1.8	1.1	0.0	**0.99**
FC3	67%	33%	0%	2.2	0.8	0.0	1.02
3D-FC3	33%	67%	0%	1.1	1.7	0.0	0.93
FC4	44%	56%	0%	1.5	1.4	0.0	**0.96**
3D-FC4	33%	67%	0%	1.1	1.7	0.0	0.93
3D-FC5	0%	100%	0%	0.0	2.5	0.0	0.83
3D-FC6	33%	56%	11%	1.1	1.4	0.4	**0.96**

The Table 5.4 shows the percentage of time when the CPU operates at each frequency ranges. The Table 5.5 shows the percentage of time of the CPU operates at each temperature ranges. The best results are highlighted in bold. The DTM evaluation index selected FC3 and 3D-FC6 as the best DTM controller designs as shown in Table 5.6. The best results are highlighted in bold. Only FC3 and 3D-FC6 controllers have high results in both frequency, and temperature evaluation indexes. As shown in Figure 5.7, both DTM controllers' frequency change responses oscillate all times. The 3D-FC6 controller has less number of frequency oscillation and smaller amplitudes. The FC3 controller operates at maximum frequency then it is switched off between 1060 and 1100 seconds. The 3D-FC6 controller is never switched off and operates at high frequency ranges but not on the maximum frequency. From the temperature point of view; both controllers temperatures are oscillating. 3D-FC6 controller has minimum temperature amplitudes at 970 and 1070 seconds as shown in Figure 5.8. The 3D-FC6 is always operating on lower temperature than the FC3 controller. Thus the 3D-FC6 controller is better then the FC3 controller.

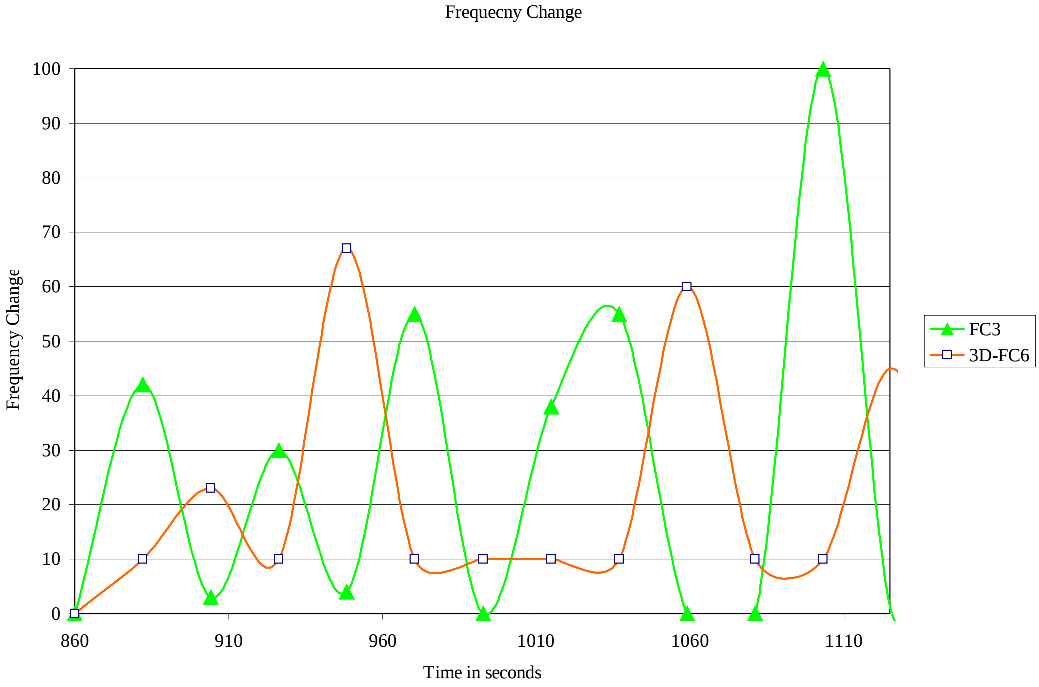


Figure 5.7 - the frequency comparisons of FC3 and 3D-FC6

Table 5.6 The DTM evaluation index of the first implementation

Controller Name	**Frequency Index** λ_1	**Temperature Index** λ_2	**The Evaluation Index** ζ_t
Desired	*1.00*	*1.00*	*2.00*
Switch	0.500	0.83	1.33
P	1.528	1.11	2.64
FC1	1.361	0.86	2.23
3D-FC1	**0.972**	**0.90**	1.87
FC2	0.500	1.02	1.52
3D-FC2	0.583	**0.99**	1.57
FC3	**1.083**	**1.02**	**2.10**
3D-FC3	0.667	**0.93**	1.59
FC4	0.750	**0.96**	1.71
3D-FC4	0.833	**0.93**	1.76
3D-FC5	**0.972**	0.83	**1.81**
3D-FC6	**0.944**	**0.96**	**1.90**

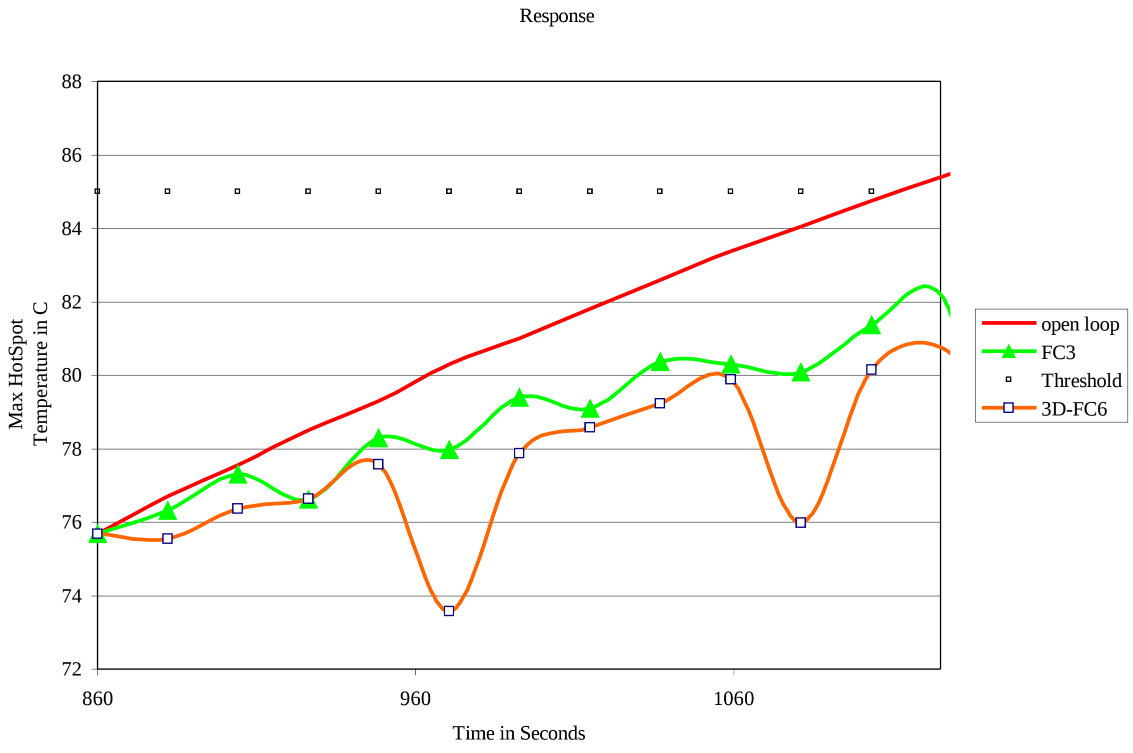


Figure 5.8 - the temperature comparisons of FC3 and 3D-FC6

As shown in Table 5.9. Only FC4, 3D-FC3 and 3D-FC6 controllers have high results in both frequency, and temperature evaluation indexes. As shown in Figure 5.9 and Figure 5.10, all DTM controllers' frequency change responses oscillate all times. The 3D-FC6 controller has the lowest number of frequency oscillation. The 3D-FC3 controller has smallest frequency changes amplitudes. The 3D-FC3 controller operates at high frequency ranges but not on the maximum frequency. From the temperature point of view; all controller temperature are increasing as shown in Figure 5.11 and 5.12. The 3D-FC6 temperature is oscillating and has minimum temperature amplitudes at 970 and 1070 seconds. There is no large advantage of any controllers over the others from temperature point of view. Thus the 3D-FC3 is better then the FC4 controller, and the 3D-FC6 controller as the 3D-FC3 controller operates at higher frequency ranges and almost the same temperature ranges.

Table 5.7 The frequency comparisons of the second implementation

Controller Name	**Frequency Ranges %** $\sigma_{1j}^{\text{Actual}}$				**Frequency Ranges Values** σ_{1j}				λ_1
	(M) j=1	**(H)** j=2	**(m)** j=3	**(L)** j=4	**(M)** j=1	**(H)** j=2	**(m)** j=3	**(L)** j=4	
$\sigma_{1j}^{\text{Desired}}$	*10%*	*70%*	*10%*	*10%*	*1.0*	*1.0*	*1.0*	*1.0*	*1.00*
Switch	0%	100%	0%	0%	0.0	1.4	0.0	0.0	0.357
P	56%	0%	22%	22%	5.6	0.0	2.2	2.2	2.500
FC1	12	22%	44%	22%	1.1	0.3	4.4	2.2	2.024
3D-FC1	0%	56%	33%	11%	0.0	0.8	3.3	1.1	1.309
FC2	0%	100%	0%	0%	0.0	1.4	0.0	0.0	0.357
3D-FC2	0%	89%	11%	0%	0.0	1.3	1.1	0.0	0.595
FC3	22%	22%	56%	0%	2.2	0.3	5.6	0.0	2.024
3D-FC3	0%	78%	22%	0%	0.0	1.1	2.2	0.0	**0.833**
FC4	0%	67%	33%	0%	0.0	0.9	3.3	0.0	**1.071**
3D-FC4	22%	56%	22%	0%	2.2	0.8	2.2	0.0	1.309
3D-FC5	0%	56%	33%	11%	0.0	0.8	3.3	1.1	1.309
3D-FC6	0%	78%	0%	22%	0.0	1.1	0.0	2.2	**0.833**

There are 3 observations extracted from Table 5.1, Table 5.3, Table 5.6, and Table 5.9 related to these two DTM evaluation index implementations as follow:

3D-FC5 vs. 3D-FC6: In the first implementation the DTM evaluation index of both controllers are almost the same from the frequency point of view. The standard deviation of the DVFS membership function (MSF) is the same but the mean is shifted by 0.2. This shift leads to insignificant frequency objective change but also leads to less CPU temperature. In the second implementation the DTM evaluation index values are totally different. So the similarity between any 2 DTM controller responses for a specific DTM design objective is not maintain for other DTM design objective.

2D Fuzzy vs. 3D Fuzzy: These DTM controllers share the same input and output membership functions. The correlation between the CPU cores has significant effect i.e. (FC1 vs. 3D-FC1) and (FC3 vs. 3D-FC3). But for (FC2 vs. 3D-FC2) there is almost no correlation effect in both DTM evaluation index implementations. This means that the selection of non proper membership functions could ignore the correlation effect between the CPU cores.

(TSC+DVFS) vs. (DVFS alone): the DTM temperature design objectives could be fulfilled by TSC+DVFS or by DVFS alone i.e. 3D-FC3 vs. 3D-FC4. The driver for using TSC with DVFS is the CPU thermal throttling limits. So if DVFS can fulfill alone the temperature DTM design objective then there is no need for combining both TSC with DVFS.

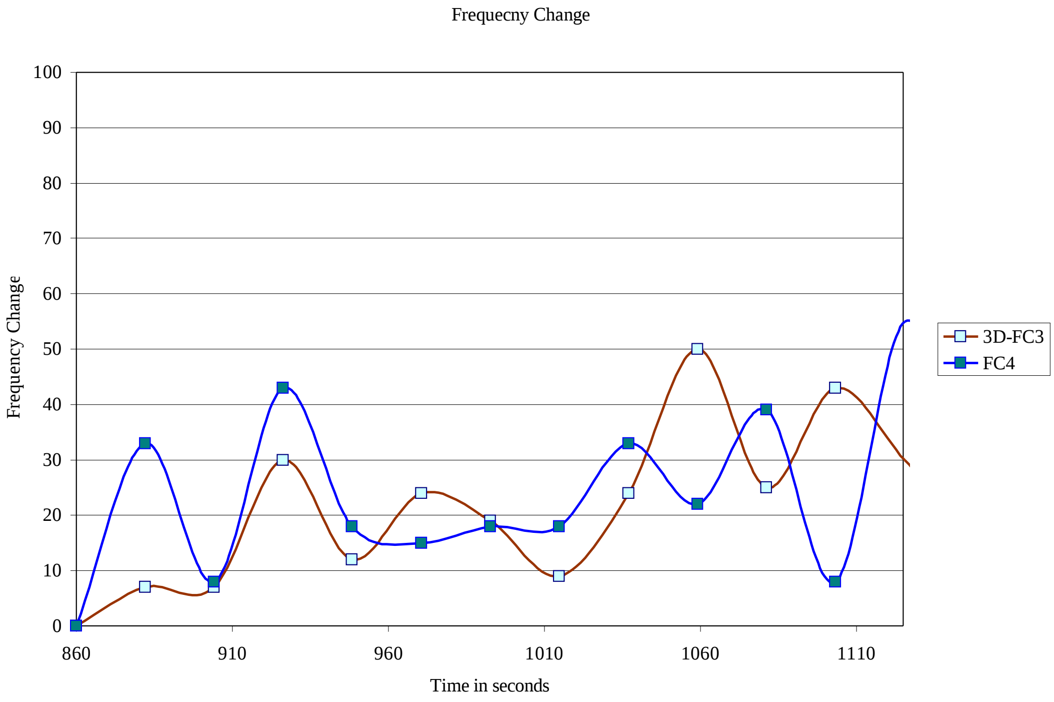


Figure 5.9 - the frequency comparisons of FC4 and 3D-FC3

Table 5.8 The temperature comparisons of the second implementation

Controller Name	**Temperature Ranges %** $\sigma_{2j}^{\text{Actual}}$			**Temperature Ranges Values** σ_{2j}			λ_2
	(H) j=1	**(m) j=2**	**(L) j=3**	**(H) j=1**	**(m) j=2**	**(L) j=3**	
$\sigma_{2j}^{\text{Desired}}$	*20%*	*50%*	*30%*	*1.0*	*1.0*	*1.0*	*1.00*
Switch	0%	100%	0%	0.0	2.0	0.0	0.67
P	78%	0%	22%	3.9	0.0	0.7	1.54
FC1	111%	89%	0%	0.6	1.8	0.0	0.78
3D-FC1	22%	78%	0%	1.1	1.6	0.0	0.89
FC2	67%	33%	0%	3.3	0.7	0.0	1.33
3D-FC2	56%	44%	0%	2.8	0.9	0.0	1.22
FC3	67%	33%	0%	3.3	0.7	0.0	1.33
3D-FC3	33%	67%	0%	1.7	1.3	0.0	**1.00**
FC4	44%	56%	0%	2.2	1.1	0.0	**1.11**
3D-FC4	33%	67%	0%	1.7	1.3	0.0	**1.00**
3D-FC5	0%	100%	0%	0.0	2.0	0.0	0.67
3D-FC6	33%	56%	11%	1.7	1.1	0.4	**1.05**

Table 5.9 The DTM evaluation index of the second implementation

Controller Name	**Frequency Index** λ_1	**Temperature Index** λ_2	**The Evaluation Index** ς_t
Desired	*1.00*	*1.00*	*2.00*
Switch	0.357	0.67	1.02
P	2.500	1.54	4.04
FC1	2.024	0.78	2.80
3D-FC1	1.309	0.89	**2.20**
FC2	0.357	1.33	1.69
3D-FC2	0.595	1.22	1.82
FC3	2.024	1.33	3.36
3D-FC3	**0.833**	**1.00**	**1.83**
FC4	**1.071**	**1.11**	**2.18**
3D-FC4	1.309	**1.00**	2.31
3D-FC5	1.309	0.67	**1.98**

3D-FC6	**0.833**	**1.05**	**1.88**

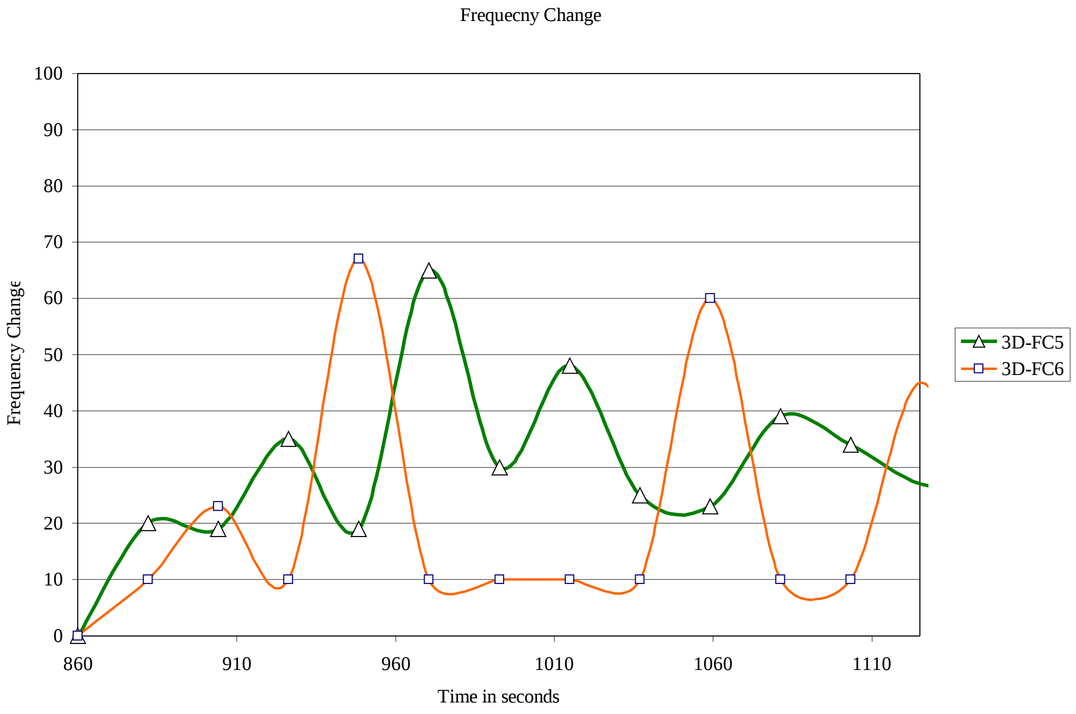


Figure 5.10 - the frequency comparisons of 3-FC5 and 3D-FC6

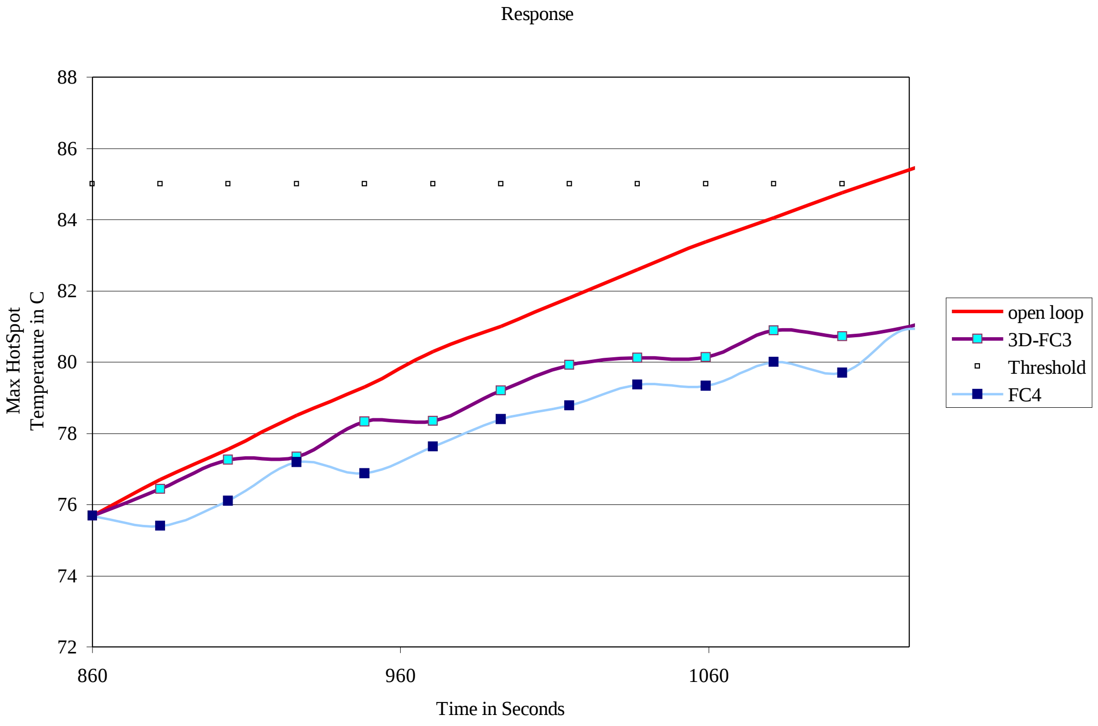


Figure 5.11 - the temperature comparisons of FC4 and 3D-FC3

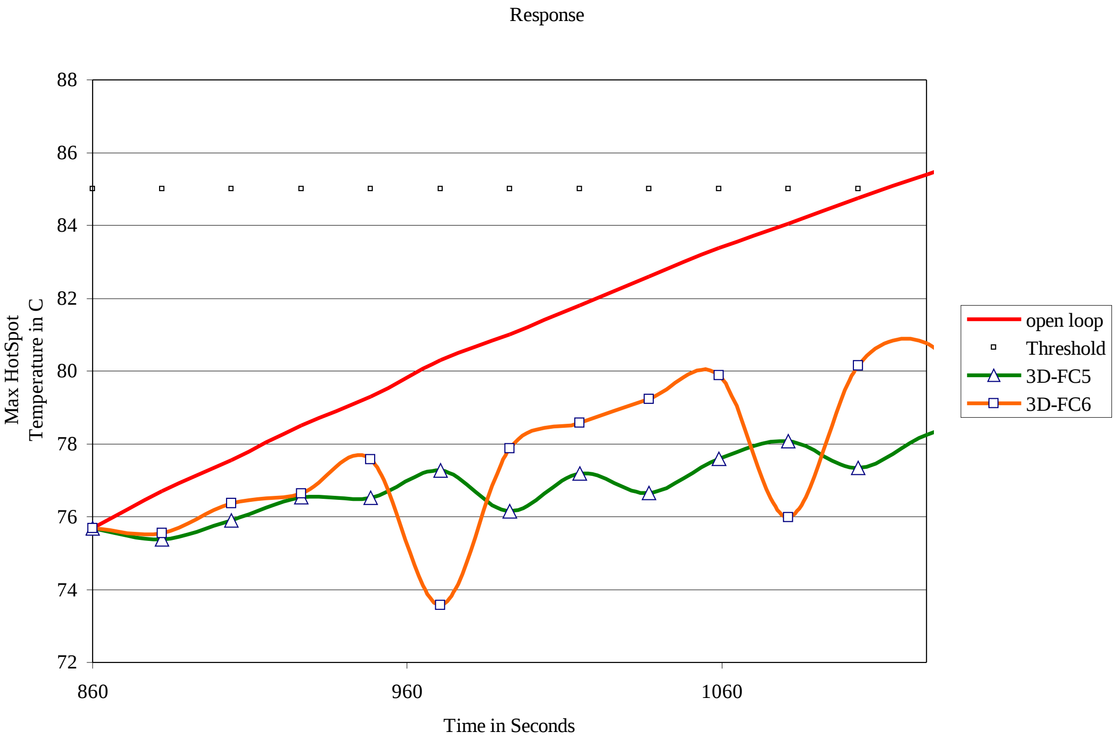


Figure 5.12 - the temperature comparisons of 3D-FC5 and 3D-FC6

5.5 Conclusion

The Fuzzy DTM controller has better response than the traditional DTM P controller. For the same input rules and the same output membership functions (MSF), the 3D Fuzzy logic reduces the CPU temperature better than the 2D Fuzzy logic. The 3D Fuzzy controller takes into consideration multiple temperatures readings distributed over the CPU chip floor plan. The Fuzzy control permits the designers to select the appropriate CPU temperature and frequency responses. For the same CPU chip, the DTM response depends on the Fuzzy controller design. As the 3D fuzzy permits the preservation of portable device battery but this affects the CPU utilization. Or it permits the high performance computing (HPC). But due to cooling limitation this DTM design is not suitable for the portable devices. The fuzzy output MSF is a critical DTM design parameter. The small deviation from the appropriate output membership function affects the DTM controller behavior. From the CPU temperature point of view; the TSC looks like a large coldspot. The cold TSC absorb the horizontal heat path as if it is a heatsink pipe. The CPU cooling system behavior depends on the combinations of the operating frequencies and temperatures. The objective of multi-parameters evaluation index is to show the different parameters effect on the CPU response. Thus the designer selects the suitable DTM controller that fulfils his requirements. The multi-parameters evaluation index permits the selection of DTM design that provides the best frequency parameter value without leading to the worst temperature parameter value.

Chapter 6

Conclusion and Future Work

6.1 Introduction

In Chapter 1, a number of design challenges and technology factors present the concerns regarding the chip temperature gradients. In terms of the technology factors, power density is increasing with each new technology. The static power consumption has significant effect and can not be neglected any more [17]. The dynamic power is the main factor in the dynamic thermal management (DTM). In Chapter 2, the CPU chip manufacturing is affected by the growth of both static and dynamic power consumptions. But the serial portion of the CPU instruction limits the speedup improvements. The CPU cores are not all fully utilized and have different power consumption profiles. In Chapter 3, the thesis discusses the CPU DTM problem, the controller design, and the controller implementation. So, in order to study the CPU thermal problem, a real Multi-Core CPU thermal model based on 45nm technology is built. This thermal model is integrated to the Hotspot 4.1 thermal simulator. In Chapter 4, different DTM controllers are integrated to this simulation setup. In Chapter 5, many simulation tests are presented and compared. The analysis of these simulation tests results is presented. This Chapter discusses the research findings and conclusion. And finally the DTM future work is highlighted.

6.2 The Multi-Core CPU Control Problem

Moore’s Law continues with technology scaling, improving transistor performance to increase frequency, increasing transistor integration capacity to realize complex architectures, and reducing energy consumed per logic operation to keep power dissipation within limit. The technology provides integration capacity of billions of transistors; however, with several fundamental barriers. The power consumption, the energy level, energy delay, power density, and floor planning are design challenges. The Multi-Core CPU design increases the CPU performance and maintains the power dissipation level for the same chip area. The CPU cores are not fully utilized if parallelism doesn't exist. Low cost portable cooling techniques exploration has more importance everyday as air cooling reaches its limits “198 Watt”. In order to study the Multi-Core CPU thermal problem a thermal model is built. The thermal model floor plan is similar to the IBM MCM POWER4 chip scaled to 45nm technology. This floor plan is integrated to the Hotspot 4.1 thermal simulator. The CPU open loop thermal profile curve is extracted. The advanced dynamic thermal management (DTM) techniques are mandatory to avoid the CPU thermal throttling. As the CPU is not 100% utilized all time, the thermal spare cores (TSC) technique is proposed. The TSC technique is based on the reservation of cores during low CPU utilization. These cores are not activate simultaneously due to limitations. During thermal crises, these reserved cores are activated to enhance the CPU utilization. The semiconductor technology permits more cores to be added to CPU chip. But the total chip area overhead is up to 27.9 % as per ITRS [82]. That means there is no chip area wasting in case of TSC. From the thermal point of view; the horizontal heat transfer path has up to 30% of CPU chip heat transfer [84]. The TSC is a big coldspot within the CPU area that handles the horizontal heat transfer path.

The cold TSC also handles the static power as the TSC core is turned off. The TSC is used simultaneous with other DTM technique. From the CPU utilization point of view, the TSC activation is equivalent to the CPU cores DVFS for a low operating frequency range.Fuzzy logic improves the DTM controller response. Fuzzy control handles the CPU thermal process without knowing its transfer function. This simplifies the DTM controller design and reduces design time. The fuzzy control permits the designers to select the appropriate CPU temperature and frequency responses. For the same CPU chip, the DTM response depends on the DTM fuzzy controller design. As the 3D fuzzy permits the preservation of portable device battery but this affects the CPU utilization. Or it permits the high performance computing (HPC). But due to cooling limitation this DTM design is not suitable for the portable devices. The 3D-FC is successfully implemented to the CPU DTM problem. Different DTM techniques are compared using simulation tests. The results demonstrate the effectiveness of the 3D fuzzy DTM controller to the nonlinear Multi-Core CPU thermal problem. The 3D fuzzy DTM takes into consideration the surrounding core hotspot temperatures and operating frequencies. The 3D fuzzy DTM avoids the complexity and maintains the correlations. As the 3D fuzzy DTM controller calculates the correlation between local core hotspot and the surrounding cores hotspots. Then it selects the appropriate local core operating frequency. The Fuzzy DTM controller has better response than the traditional DTM P controller. For the same input rules and the same output membership functions (MSF), the 3D fuzzy logic reduces the CPU temperature better than the 2D fuzzy logic. The fuzzy output MSF is a critical DTM design parameter. The small deviation from the appropriate output membership function affects the DTM controller behavior.

6.3 Future Work

6.3.1 The DTM Thread Migration

The thread migration (TM) DTM technique is orthogonal to DVFS and TSC. Thus it is recommended to use TM with DVFS and TSC. TM requires interaction and handshakes between the operating system and DTM controller.

6.3.2 Software vs Hardware DTM controller

The DTM controller location is a DTM design challenge. The software fuzzy controller embedded with in the operating system increases the flexibility. Also the software fuzzy controller handles different CPU chip without DTM design change. While the hardware DTM controller is CPU chip specific rather than the generic software DTM controller.

6.3.3 The Optimal Floor Plan

The chip floor plan is a critical thermal parameter. The CPU over thermal profile is enhanced by surrounding the hotspots with coldspot function blocks. The chip floor plan thermal optimization is a challenge.

6.3.4 Tuning the Fuzzy Controller

The fuzzy rules tuning improves the DTM controller response. Also the fuzzy input and output membership function (MSF) is critical DTM design parameters. The tuning process is done off line during the DTM design phase. The Genetic Algorithm or the Particle Swarm optimization techniques are suitable for optimizing the DTM fuzzy controller response. The optimization design required high performer computer as the simulation execution takes long time.

Appendix

Appendix Table 1

Normalized MCM POWER4 Floor plan as a picture in pixels

Name	Width	Height	Left-X	Bottom-Y
c0_p1	45	34	11	28
c1_p1	45	34	63	28
L2_p1	82	34	24	68
L3_dc_p1	12	44	11	67
conn1_p1	85	8	24	102
conn2_p1	42	8	105	62
conn3_p1	5	24	109	41
c2_p2	34	45	181	19
c3_p2	34	45	181	64
L2_p2	34	82	138	28
L3_dc_p2	44	12	132	15
conn1_p2	8	85	130	28
conn2_p2	42	8	139	103
conn3_p2	24	5	182	105
c4_p3	45	34	182	170
c5_p3	45	34	135	170
L2_p3	82	34	138	129
L3_dc_p3	12	44	220	124
conn1_p3	8	85	130	122
conn2_p3	8	42	130	129
conn3_p3	5	24	130	170
c6_p4	45	34	26	170
c7_p4	45	34	26	128
L2_p4	82	34	68	128
L3_dc_p4	44	12	68	205
conn1_p4	24	5	41	121
conn2_p4	42	8	62	121
conn3_p4	8	85	105	129
cross1	17	205	113	14
cross2	11	224	11	110

Appendix Table 2

Normalized MCM POWER4 blocks Floor plan as a picture in pixels

Name	Width	Height	Left-X	Bottom-Y
fpu_b1_c0_p1	6	4	12	29
fpu_b2_c0_p1	6	4	18	29
fpu_rg_c0_p1	12	9	12	33
isu_al_c0_p1	26	13	24	29
fxu_r1_c0_p1	8	6	50	29
fxu_b1_c0_p1	4	5	50	35
fxu_b2_c0_p1	4	5	54	35
fxu_r2_c0_p1	8	2	50	40
in_lft_c0_p1	41	3	17	42
idu_al_c0_p1	8	9	17	45
ifu_ch_c0_p1	8	25	25	45
ifu_b1_c0_p1	10	4	17	54
ifu_b2_c0_p1	10	4	17	58
ifu_b3_c0_p1	8	8	17	62
lsu_nt_c0_p1	9	9	33	45
lsu_cr_c0_p1	4	4	38	54
lsu_b1_c0_p1	16	9	42	45
lsu_b2_c0_p1	16	13	42	54
lsu_b3_c0_p1	5	4	33	54
lsu_ch_c0_p1	9	8	33	59
in_ver_c0_p1	5	28	12	42
in_hor_c0_p1	21	3	12	70
inter_ver_p1	4	38	58	29
inter_hor_p1	54	6	33	67
fpu_b1_c1_p1	6	4	102	29
fpu_b2_c1_p1	6	4	96	29
fpu_rg_c1_p1	12	9	96	33
isu_al_c1_p1	26	13	70	29
fxu_r1_c1_p1	8	6	62	29
fxu_b1_c1_p1	4	5	62	35
fxu_b2_c1_p1	4	5	66	35
fxu_r2_c1_p1	8	2	62	40
in_rit_c1_p1	41	3	62	42
idu_al_c1_p1	8	9	95	45
ifu_ch_c1_p1	8	25	87	45
ifu_b1_c1_p1	10	4	93	54
ifu_b2_c0_p1	10	4	93	58
ifu_b3_c1_p1	8	8	95	62

Appendix Table 2 Continued

Normalized MCM POWER4 blocks Floor plan as a picture in pixels

Name	Width	Height	Left-X	Bottom-Y
lsu_nt_c1_p1	9	9	78	45
lsu_cr_c1_p1	4	4	78	54
lsu_b1_c1_p1	16	9	62	45
lsu_b2_c1_p1	16	13	62	54
lsu_b3_c1_p1	5	4	82	54
lsu_ch_c1_p1	9	8	78	59
in_ver_c1_p1	5	28	103	42
in_hor_c1_p1	21	3	87	70
L3_con_b1_p1	12	14	12	73
L3_con_b2_p1	12	18	12	87
L3_con_b3_p1	12	14	12	105
L2_aaa_b1_p1	26	14	24	73
L2_aaa_b2_p1	26	13	24	87
L2_aaa_b3_p1	26	14	24	100
L2_bbb_b1_p1	28	14	50	73
L2_bbb_b2_p1	28	13	50	87
L2_bbb_b3_p1	28	14	50	100
L2_ccc_b1_p1	30	14	78	73
L2_ccc_b2_p1	30	13	78	87
L2_ccc_b3_p1	30	14	78	100
inter_bot_p1	84	5	24	114
fpu_b1_c2_p2	4	6	210	23
fpu_b2_c2_p2	4	6	210	29
fpu_rg_c2_p2	9	12	201	23
isu_al_c2_p2	13	26	201	35
fxu_r1_c2_p2	6	8	208	61
fxu_b1_c2_p2	5	4	203	61
fxu_b2_c2_p2	5	4	203	65
fxu_r2_c2_p2	2	8	201	61
in_lft_c2_p2	3	41	198	28
idu_al_c2_p2	9	8	189	28
ifu_ch_c2_p2	25	8	173	36
ifu_b1_c2_p2	4	10	185	28
ifu_b2_c2_p2	4	10	181	28
ifu_b3_c2_p2	8	8	173	28
lsu_nt_c2_p2	9	9	189	44

Appendix Table 2

Normalized MCM POWER4 blocks Floor plan as a picture in pixels Continued

Name	Width	Height	Left-X	Bottom-Y
lsu_cr_c2_p2	4	4	184	49
lsu_b1_c2_p2	9	16	189	53
lsu_b2_c2_p2	13	16	176	53
lsu_b3_c2_p2	4	5	184	44
lsu_ch_c2_p2	8	9	176	44
in_ver_c2_p2	28	5	173	23
in_hor_c2_p2	3	21	170	23
inter_ver_p2	38	4	176	69
inter_hor_p2	6	54	170	44
fpu_b1_c3_p2	4	6	210	113
fpu_b2_c3_p2	4	6	210	107
fpu_rg_c3_p2	9	12	201	107
isu_al_c3_p2	13	26	201	81
fxu_r1_c3_p2	6	8	208	73
fxu_b1_c3_p2	5	4	203	77
fxu_b2_c3_p2	5	4	203	73
fxu_r2_c3_p2	2	8	201	73
in_rit_c3_p2	3	41	198	73
idu_al_c3_p2	9	8	189	106
ifu_ch_c3_p2	25	8	173	98
ifu_b1_c3_p2	4	10	185	104
ifu_b2_c3_p2	4	10	181	104
ifu_b3_c3_p2	8	8	173	106
lsu_nt_c3_p2	9	9	189	89
lsu_cr_c3_p2	4	4	184	89
lsu_b1_c3_p2	9	16	189	73
lsu_b2_c3_p2	13	16	176	73
lsu_b3_c3_p2	4	5	184	93
lsu_ch_c3_p2	8	9	176	89
in_ver_c3_p2	28	5	173	114
in_hor_c3_p2	3	21	170	98
L3_con_b1_p2	14	12	156	23
L3_con_b2_p2	18	12	138	23
L3_con_b3_p2	14	12	124	23
L2_aaa_b1_p2	14	26	156	35
L2_aaa_b2_p2	13	26	143	35
L2_aaa_b3_p2	14	26	129	35
L2_bbb_b1_p2	14	28	156	61

Appendix Table 2

Normalized MCM POWER4 blocks Floor plan as a picture in pixels

Continued

Name	**Width**	**Height**	**Left-X**	**Bottom-Y**
L2_bbb_b2_p2	13	28	143	61
L2_bbb_b3_p2	14	28	129	61
L2_ccc_b1_p2	14	30	156	89
L2_ccc_b2_p2	13	30	143	89
L2_ccc_b3_p2	14	30	129	89
inter_bot_p2	5	84	124	35
fpu_b1_c4_p3	6	4	214	218
fpu_b2_c4_p3	6	4	208	218
fpu_rg_c4_p3	12	9	208	209
isu_al_c4_p3	26	13	182	209
fxu_r1_c4_p3	8	6	174	216
fxu_b1_c4_p3	4	5	174	211
fxu_b2_c4_p3	4	5	178	211
fxu_r2_c4_p3	8	2	174	209
in_lft_c4_p3	41	3	174	206
idu_al_c4_p3	8	9	207	197
ifu_ch_c4_p3	8	25	199	181
ifu_b1_c4_p3	10	4	205	193
ifu_b2_c4_p3	10	4	205	189
ifu_b3_c4_p3	8	8	207	181
lsu_nt_c4_p3	9	9	190	197
lsu_cr_c4_p3	4	4	190	192
lsu_b1_c4_p3	16	9	174	197
lsu_b2_c4_p3	16	13	174	184
lsu_b3_c4_p3	5	4	194	192
lsu_ch_c4_p3	9	8	190	184
in_ver_c4_p3	5	28	215	181
in_hor_c4_p3	21	3	199	178
inter_ver_p3	4	38	170	184
inter_hor_p3	54	6	145	178
fpu_b1_c5_p3	6	4	124	218
fpu_b2_c5_p3	6	4	130	218
fpu_rg_c5_p3	12	9	124	209
isu_al_c5_p3	26	13	136	209
fxu_r1_c5_p3	8	6	162	216
fxu_b1_c5_p3	4	5	162	211
fxu_b2_c5_p3	4	5	166	211

Appendix Table 2

Normalized MCM POWER4 blocks Floor plan as a picture in pixels

Continued

Name	Width	Height	Left-X	Bottom-Y
fxu_r2_c5_p3	8	2	162	209
in_rit_c5_p3	41	3	129	206
idu_al_c5_p3	8	9	129	197
ifu_ch_c5_p3	8	25	137	181
ifu_b1_c5_p3	10	4	129	193
ifu_b2_c5_p3	10	4	129	189
ifu_b3_c5_p3	8	8	129	181
lsu_nt_c5_p3	9	9	145	197
lsu_cr_c5_p3	4	4	150	192
lsu_b1_c5_p3	16	9	154	197
lsu_b2_c5_p3	16	13	154	184
lsu_b3_c5_p3	5	4	145	192
lsu_ch_c5_p3	9	8	145	184
in_ver_c5_p3	5	28	124	181
in_hor_c5_p3	21	3	124	178
L3_con_b1_p3	12	14	208	164
L3_con_b2_p3	12	18	208	146
L3_con_b3_p3	12	14	208	132
L2_aaa_b1_p3	26	14	182	164
L2_aaa_b2_p3	26	13	182	151
L2_aaa_b3_p3	26	14	182	137
L2_bbb_b1_p3	28	14	154	164
L2_bbb_b2_p3	28	13	154	151
L2_bbb_b3_p3	28	14	154	137
L2_ccc_b1_p3	30	14	124	164
L2_ccc_b2_p3	30	13	124	151
L2_ccc_b3_p3	30	14	124	137
inter_bot_p3	84	5	124	132
fpu_b1_c6_p4	4	6	18	222
fpu_b2_c6_p4	4	6	18	216
fpu_rg_c6_p4	9	12	22	216
isu_al_c6_p4	13	26	18	190
fxu_r1_c6_p4	6	8	18	182
fxu_b1_c6_p4	5	4	24	186
fxu_b2_c6_p4	5	4	24	182
fxu_r2_c6_p4	2	8	29	182

Appendix Table 2

Normalized MCM POWER4 blocks Floor plan as a picture in pixels

Name	Width	Height	Left-X	Bottom-Y
in_lft_c6_p4	3	41	31	182
idu_al_c6_p4	9	8	34	215
ifu_ch_c6_p4	25	8	34	207
ifu_b1_c6_p4	4	10	43	213
ifu_b2_c6_p4	4	10	47	213
ifu_b3_c6_p4	8	8	51	215
lsu_nt_c6_p4	9	9	34	198
lsu_cr_c6_p4	4	4	43	198
lsu_b1_c6_p4	9	16	34	182
lsu_b2_c6_p4	13	16	43	182
lsu_b3_c6_p4	4	5	43	202
lsu_ch_c6_p4	8	9	48	198
in_ver_c6_p4	28	5	31	223
in_hor_c6_p4	3	21	59	207
inter_ver_p4	38	4	18	178
inter_hor_p4	6	54	56	153
fpu_b1_c7_p4	4	6	18	132
fpu_b2_c7_p4	4	6	18	138
fpu_rg_c7_p4	9	12	22	132
isu_al_c7_p4	13	26	18	144
fxu_r1_c7_p4	6	8	18	170
fxu_b1_c7_p4	5	4	24	170
fxu_b2_c7_p4	5	4	24	174
fxu_r2_c7_p4	2	8	29	170
in_rit_c7_p4	3	41	31	137
idu_al_c7_p4	9	8	34	137
ifu_ch_c7_p4	25	8	34	145
ifu_b1_c7_p4	4	10	43	137
ifu_b2_c7_p4	4	10	47	137
ifu_b3_c7_p4	8	8	51	137
lsu_nt_c7_p4	9	9	34	153
lsu_cr_c7_p4	4	4	43	158
lsu_b1_c7_p4	9	16	34	162
lsu_b2_c7_p4	13	16	43	162
lsu_b3_c7_p4	4	5	43	153
lsu_ch_c7_p4	8	9	48	153
in_ver_c7_p4	28	5	31	132
in_hor_c7_p4	3	21	59	132
L3_con_b1_p4	14	12	62	216

Continued

Appendix Table 2

Normalized MCM POWER4 blocks Floor plan as a picture in pixels

Name	Width	Height	Left-X	Bottom-Y
L3_con_b2_p4	18	12	76	216
L3_con_b3_p4	14	12	94	216
L2_aaa_b1_p4	14	26	62	190
L2_aaa_b2_p4	13	26	76	190
L2_aaa_b3_p4	14	26	89	190
L2_bbb_b1_p4	14	28	62	162
L2_bbb_b2_p4	13	28	76	162
L2_bbb_b3_p4	14	28	89	162
L2_ccc_b1_p4	14	30	62	132
L2_ccc_b2_p4	13	30	76	132
L2_ccc_b3_p4	14	30	89	132
inter_bot_p4	5	84	103	132
inter_cross1	16	205	108	23
inter_cross2	208	13	12	119

Continued

Appendix Table 3

MCM POWER4 detailed block Floor plan 45nm

Name	Width	Height	Left-X	Bottom-Y
fpu_b1_c0_p1	0.00067526	0.00045017	0.00135051	0.00326374
fpu_b2_c0_p1	0.00067526	0.00045017	0.00202577	0.00326374
fpu_rg_c0_p1	0.00135051	0.00101288	0.00135051	0.00371391
isu_al_c0_p1	0.00292611	0.00146306	0.00270103	0.00326374
fxu_r1_c0_p1	0.00090034	0.00067526	0.00562714	0.00326374
fxu_b1_c0_p1	0.00045017	0.00056271	0.00562714	0.003939
fxu_b2_c0_p1	0.00045017	0.00056271	0.00607731	0.003939

Appendix Table 3

MCM POWER4 detailed block Floor plan 45nm

Name	Width	Height	Left-X	Bottom-Y
fxu_r2_c0_p1	0.00090034	0.00022509	0.00562714	0.00450171
in_lft_c0_p1	0.00461425	0.00033763	0.00191323	0.0047268
idu_al_c0_p1	0.00090034	0.00101288	0.00191323	0.00506442
ifu_ch_c0_p1	0.00090034	0.00281357	0.00281357	0.00506442
ifu_b1_c0_p1	0.00112543	0.00045017	0.00191323	0.00607731
ifu_b2_c0_p1	0.00112543	0.00045017	0.00191323	0.00652748
ifu_b3_c0_p1	0.00090034	0.00090034	0.00191323	0.00697765
lsu_nt_c0_p1	0.00101288	0.00101288	0.00371391	0.00506442
lsu_cr_c0_p1	0.00045017	0.00045017	0.00427662	0.00607731
lsu_b1_c0_p1	0.00180068	0.00101288	0.0047268	0.00506442
lsu_b2_c0_p1	0.00180068	0.00146306	0.0047268	0.00607731
lsu_b3_c0_p1	0.00056271	0.00045017	0.00371391	0.00607731
lsu_ch_c0_p1	0.00101288	0.00090034	0.00371391	0.00664002
in_ver_c0_p1	0.00056271	0.0031512	0.00135051	0.0047268
in_hor_c0_p1	0.0023634	0.00033763	0.00135051	0.00787799
inter_ver_p1	0.00067526	0.00045017	0.01147936	0.00326374
inter_hor_p1	0.00067526	0.00045017	0.0108041	0.00326374
fpu_b1_c1_p1	0.00135051	0.00101288	0.0108041	0.00371391
fpu_b2_c1_p1	0.00292611	0.00146306	0.00787799	0.00326374
fpu_rg_c1_p1	0.00090034	0.00067526	0.00697765	0.00326374
isu_al_c1_p1	0.0004501	0.0005627	0.0069776	0.003939

Continued

Appendix Table 3

MCM POWER4 detailed block Floor plan 45nm

Name	Width	Height	Left-X	Bottom-Y
	7	1	5	
fxu_r1_c1_p1	0.00045017	0.00056271	0.00742782	0.003939
fxu_b1_c1_p1	0.00090034	0.00022509	0.00697765	0.00450171
fxu_b2_c1_p1	0.00461425	0.00033763	0.00697765	0.0047268
fxu_r2_c1_p1	0.00090034	0.00101288	0.01069156	0.00506442
in_rit_c1_p1	0.00090034	0.00281357	0.00979122	0.00506442
idu_al_c1_p1	0.00112543	0.00045017	0.01046648	0.00607731
ifu_ch_c1_p1	0.00112543	0.00045017	0.01046648	0.00652748
ifu_b1_c1_p1	0.00090034	0.00090034	0.01069156	0.00697765
ifu_b2_c0_p1	0.00101288	0.00101288	0.00877833	0.00506442
ifu_b3_c1_p1	0.00045017	0.00045017	0.00877833	0.00607731
lsu_nt_c1_p1	0.00180068	0.00101288	0.00697765	0.00506442
lsu_cr_c1_p1	0.00180068	0.00146306	0.00697765	0.00607731
lsu_b1_c1_p1	0.00056271	0.00045017	0.00922851	0.00607731
lsu_b2_c1_p1	0.00101288	0.00090034	0.00877833	0.00664002
lsu_b3_c1_p1	0.00056271	0.0031512	0.0115919	0.0047268
lsu_ch_c1_p1	0.0023634	0.00033763	0.00979122	0.00787799
in_ver_c1_p1	0.00045017	0.00067526	0.02363398	0.00258848
in_hor_c1_p1	0.00045017	0.00067526	0.02363398	0.00326374
L3_con_b1_p1	0.00101288	0.00135051	0.02262109	0.00258848
L3_con_b2_p1	0.00146306	0.00292611	0.02262109	0.003939

Appendix Table 3

MCM POWER4 detailed block Floor plan 45nm

Name	Width	Height	Left-X	Bottom-Y
L3_con_b3_p1	0.00067526	0.00090034	0.02340889	0.00686511
L2_aaa_b1_p1	0.00056271	0.00045017	0.02284618	0.00686511
L2_aaa_b2_p1	0.00056271	0.00045017	0.02284618	0.00731528
L2_aaa_b3_p1	0.00022509	0.00090034	0.02262109	0.00686511
L2_bbb_b1_p1	0.00033763	0.00461425	0.02228346	0.0031512
L2_bbb_b2_p1	0.00101288	0.00090034	0.02127058	0.0031512
L2_bbb_b3_p1	0.00281357	0.00090034	0.0194699	0.00405154
L2_ccc_b1_p1	0.00045017	0.00112543	0.02082041	0.0031512
L2_ccc_b2_p1	0.00045017	0.00112543	0.02037024	0.0031512
L2_ccc_b3_p1	0.00090034	0.00090034	0.0194699	0.0031512
inter_bot_p1	0.00101288	0.00101288	0.02127058	0.00495188
fpu_b1_c2_p2	0.00045017	0.00045017	0.02070787	0.00551459
fpu_b2_c2_p2	0.00101288	0.00180068	0.02127058	0.00596477
fpu_rg_c2_p2	0.00146306	0.00180068	0.01980752	0.00596477
isu_al_c2_p2	0.00045017	0.00056271	0.02070787	0.00495188
fxu_r1_c2_p2	0.00090034	0.00101288	0.01980752	0.00495188
fxu_b1_c2_p2	0.0031512	0.00056271	0.0194699	0.00258848
fxu_b2_c2_p2	0.00033763	0.0023634	0.01913227	0.00258848
fxu_r2_c2_p2	0.00045017	0.00067526	0.02363398	0.01271733
in_lft_c2_p2	0.00045017	0.00067526	0.02363398	0.01204207
idu_al_c2_p2	0.0010128	0.0013505	0.0226210	0.01204207

Continued

Appendix Table 3

MCM POWER4 detailed block Floor plan 45nm

Name	Width	Height	Left-X	Bottom-Y
	8	1	9	
ifu_ch_c2_p2	0.00146306	0.00292611	0.02262109	0.00911596
ifu_b1_c2_p2	0.00067526	0.00090034	0.02340889	0.00821562
ifu_b2_c2_p2	0.00056271	0.00045017	0.02284618	0.00866579
ifu_b3_c2_p2	0.00056271	0.00045017	0.02284618	0.00821562
lsu_nt_c2_p2	0.00022509	0.00090034	0.02262109	0.00821562
lsu_cr_c2_p2	0.00033763	0.00461425	0.02228346	0.00821562
lsu_b1_c2_p2	0.00101288	0.00090034	0.02127058	0.01192953
lsu_b2_c2_p2	0.00281357	0.00090034	0.0194699	0.01102919
lsu_b3_c2_p2	0.00045017	0.00112543	0.02082041	0.01170445
lsu_ch_c2_p2	0.00045017	0.00112543	0.02037024	0.01170445
in_ver_c2_p2	0.00090034	0.00090034	0.0194699	0.01192953
in_hor_c2_p2	0.00101288	0.00101288	0.02127058	0.0100163
inter_ver_p2	0.00045017	0.00045017	0.02070787	0.0100163
inter_hor_p2	0.00101288	0.00180068	0.02127058	0.00821562
fpu_b1_c3_p2	0.00146306	0.00180068	0.01980752	0.00821562
fpu_b2_c3_p2	0.00045017	0.00056271	0.02070787	0.01046648
fpu_rg_c3_p2	0.00090034	0.00101288	0.01980752	0.0100163
isu_al_c3_p2	0.0031512	0.00056271	0.0194699	0.01282987
fxu_r1_c3_p2	0.00033763	0.0023634	0.01913227	0.01102919
fxu_b1_c3_p2	0.00067526	0.00045017	0.02408415	0.02453432

Appendix Table 3

MCM POWER4 detailed block Floor plan 45nm

Name	Width	Height	Left-X	Bottom-Y
fxu_b2_c3_p2	0.00067526	0.00045017	0.02340889	0.02453432
fxu_r2_c3_p2	0.00135051	0.00101288	0.02340889	0.02352143
in_rit_c3_p2	0.00292611	0.00146306	0.02048278	0.02352143
idu_al_c3_p2	0.00090034	0.00067526	0.01958244	0.02430923
ifu_ch_c3_p2	0.00045017	0.00056271	0.01958244	0.02374652
ifu_b1_c3_p2	0.00045017	0.00056271	0.02003261	0.02374652
ifu_b2_c3_p2	0.00090034	0.00022509	0.01958244	0.02352143
ifu_b3_c3_p2	0.00461425	0.00033763	0.01958244	0.02318381
lsu_nt_c3_p2	0.00090034	0.00101288	0.02329635	0.02217092
lsu_cr_c3_p2	0.00090034	0.00281357	0.02239601	0.02037024
lsu_b1_c3_p2	0.00112543	0.00045017	0.02307126	0.02172075
lsu_b2_c3_p2	0.00112543	0.00045017	0.02307126	0.02127058
lsu_b3_c3_p2	0.00090034	0.00090034	0.02329635	0.02037024
lsu_ch_c3_p2	0.00101288	0.00101288	0.02138312	0.02217092
in_ver_c3_p2	0.00045017	0.00045017	0.02138312	0.02160821
in_hor_c3_p2	0.00180068	0.00101288	0.01958244	0.02217092
L3_con_b1_p2	0.00180068	0.00146306	0.01958244	0.02070787
L3_con_b2_p2	0.00056271	0.00045017	0.02183329	0.02160821
L3_con_b3_p2	0.00101288	0.00090034	0.02138312	0.02070787
L2_aaa_b1_p2	0.00056271	0.0031512	0.02419669	0.02037024
L2_aaa_b2_p2	0.0023634	0.0003376	0.0223960	0.02003261

Continued

Appendix Table 3

MCM POWER4 detailed block Floor plan 45nm

Name	Width	Height	Left-X	Bottom-Y
		3	1	
L2_aaa_b3_p2	0.00067526	0.00045017	0.0139553	0.02453432
L2_bbb_b1_p2	0.00067526	0.00045017	0.01463056	0.02453432
L2_bbb_b2_p2	0.00135051	0.00101288	0.0139553	0.02352143
L2_bbb_b3_p2	0.00292611	0.00146306	0.01530581	0.02352143
L2_ccc_b1_p2	0.00090034	0.00067526	0.01823192	0.02430923
L2_ccc_b2_p2	0.00045017	0.00056271	0.01823192	0.02374652
L2_ccc_b3_p2	0.00045017	0.00056271	0.0186821	0.02374652
inter_bot_p2	0.00090034	0.00022509	0.01823192	0.02352143
fpu_b1_c4_p3	0.00461425	0.00033763	0.01451801	0.02318381
fpu_b2_c4_p3	0.00090034	0.00101288	0.01451801	0.02217092
fpu_rg_c4_p3	0.00090034	0.00281357	0.01541836	0.02037024
isu_al_c4_p3	0.00112543	0.00045017	0.01451801	0.02172075
fxu_r1_c4_p3	0.00112543	0.00045017	0.01451801	0.02127058
fxu_b1_c4_p3	0.00090034	0.00090034	0.01451801	0.02037024
fxu_b2_c4_p3	0.00101288	0.00101288	0.0163187	0.02217092
fxu_r2_c4_p3	0.00045017	0.00045017	0.01688141	0.02160821
in_lft_c4_p3	0.00180068	0.00101288	0.01733158	0.02217092
idu_al_c4_p3	0.00180068	0.00146306	0.01733158	0.02070787
ifu_ch_c4_p3	0.00056271	0.00045017	0.0163187	0.02160821
ifu_b1_c4_p3	0.00101288	0.00090034	0.0163187	0.02070787

Appendix Table 3

MCM POWER4 detailed block Floor plan 45nm

Name	Width	Height	Left-X	Bottom-Y
ifu_b2_c4_p3	0.00056271	0.0031512	0.0139553	0.02037024
ifu_b3_c4_p3	0.0023634	0.00033763	0.0139553	0.02003261
lsu_nt_c4_p3	0.00045017	0.00067526	0.00202577	0.02498449
lsu_cr_c4_p3	0.00045017	0.00067526	0.00202577	0.02430923
lsu_b1_c4_p3	0.00101288	0.00135051	0.00247594	0.02430923
lsu_b2_c4_p3	0.00146306	0.00292611	0.00202577	0.02138312
lsu_b3_c4_p3	0.00067526	0.00090034	0.00202577	0.02048278
lsu_ch_c4_p3	0.00056271	0.00045017	0.00270103	0.02093295
in_ver_c4_p3	0.00056271	0.00045017	0.00270103	0.02048278
in_hor_c4_p3	0.00022509	0.00090034	0.00326374	0.02048278
inter_ver_p3	0.00033763	0.00461425	0.00348883	0.02048278
inter_hor_p3	0.00101288	0.00090034	0.00382645	0.02419669
fpu_b1_c5_p3	0.00281357	0.00090034	0.00382645	0.02329635
fpu_b2_c5_p3	0.00045017	0.00112543	0.00483934	0.0239716
fpu_rg_c5_p3	0.00045017	0.00112543	0.00528951	0.0239716
isu_al_c5_p3	0.00090034	0.00090034	0.00573968	0.02419669
fxu_r1_c5_p3	0.00101288	0.00101288	0.00382645	0.02228346
fxu_b1_c5_p3	0.00045017	0.00045017	0.00483934	0.02228346
fxu_b2_c5_p3	0.00101288	0.00180068	0.00382645	0.02048278
fxu_r2_c5_p3	0.00146306	0.00180068	0.00483934	0.02048278
in_rit_c5_p3	0.0004501	0.0005627	0.0048393	0.02273363

Continued

Appendix Table 3

MCM POWER4 detailed block Floor plan 45nm

Name	Width	Height	Left-X	Bottom-Y
	7	1	4	
idu_al_c5_p3	0.00090034	0.00101288	0.00540205	0.02228346
ifu_ch_c5_p3	0.0031512	0.00056271	0.00348883	0.02509703
ifu_b1_c5_p3	0.00033763	0.0023634	0.00664002	0.02329635
ifu_b2_c5_p3	0.00045017	0.00067526	0.00202577	0.01485564
ifu_b3_c5_p3	0.00045017	0.00067526	0.00202577	0.0155309
lsu_nt_c5_p3	0.00101288	0.00135051	0.00247594	0.01485564
lsu_cr_c5_p3	0.00146306	0.00292611	0.00202577	0.01620616
lsu_b1_c5_p3	0.00067526	0.00090034	0.00202577	0.01913227
lsu_b2_c5_p3	0.00056271	0.00045017	0.00270103	0.01913227
lsu_b3_c5_p3	0.00056271	0.00045017	0.00270103	0.01958244
lsu_ch_c5_p3	0.00022509	0.00090034	0.00326374	0.01913227
in_ver_c5_p3	0.00033763	0.00461425	0.00348883	0.01541836
in_hor_c5_p3	0.00101288	0.00090034	0.00382645	0.01541836
L3_con_b1_p3	0.00281357	0.00090034	0.00382645	0.0163187
L3_con_b2_p3	0.00045017	0.00112543	0.00483934	0.01541836
L3_con_b3_p3	0.00045017	0.00112543	0.00528951	0.01541836
L2_aaa_b1_p3	0.00090034	0.00090034	0.00573968	0.01541836
L2_aaa_b2_p3	0.00101288	0.00101288	0.00382645	0.01721904
L2_aaa_b3_p3	0.00045017	0.00045017	0.00483934	0.01778175
L2_bbb_b1_p3	0.00101288	0.00180068	0.00382645	0.01823192

Appendix Table 3

MCM POWER4 detailed block Floor plan 45nm

Name	Width	Height	Left-X	Bottom-Y
L2_bbb_b2_p3	0.00146306	0.00180068	0.00483934	0.01823192
L2_bbb_b3_p3	0.00045017	0.00056271	0.00483934	0.01721904
L2_ccc_b1_p3	0.00090034	0.00101288	0.00540205	0.01721904
L2_ccc_b2_p3	0.0031512	0.00056271	0.00348883	0.01485564
L2_ccc_b3_p3	0.00033763	0.0023634	0.00664002	0.01485564
inter_bot_p3	0.00045017	0.00427662	0.00652748	0.00326374
fpu_b1_c6_p4	0.00607731	0.00067526	0.00371391	0.00754036
fpu_b2_c6_p4	0.00135051	0.0015756	0.00135051	0.00821562
fpu_rg_c6_p4	0.00135051	0.00202577	0.00135051	0.00979122
isu_al_c6_p4	0.00135051	0.0015756	0.00135051	0.01181699
fxu_r1_c6_p4	0.00292611	0.0015756	0.00270103	0.00821562
fxu_b1_c6_p4	0.00292611	0.00146306	0.00270103	0.00979122
fxu_b2_c6_p4	0.00292611	0.0015756	0.00270103	0.01125427
fxu_r2_c6_p4	0.0031512	0.0015756	0.00562714	0.00821562
in_lft_c6_p4	0.0031512	0.00146306	0.00562714	0.00979122
idu_al_c6_p4	0.0031512	0.0015756	0.00562714	0.01125427
ifu_ch_c6_p4	0.00337628	0.0015756	0.00877833	0.00821562
ifu_b1_c6_p4	0.00337628	0.00146306	0.00877833	0.00979122
ifu_b2_c6_p4	0.00337628	0.0015756	0.00877833	0.01125427
ifu_b3_c6_p4	0.00945359	0.00056271	0.00270103	0.01282987
lsu_nt_c6_p4	0.0042766	0.0004501	0.0198075	0.00776545

Continued

Appendix Table 3

MCM POWER4 detailed block Floor plan 45nm

Name	Width	Height	Left-X	Bottom-Y
	2	7	2	
lsu_cr_c6_p4	0.00067526	0.00607731	0.01913227	0.00495188
lsu_b1_c6_p4	0.0015756	0.00135051	0.01755667	0.00258848
lsu_b2_c6_p4	0.00202577	0.00135051	0.0155309	0.00258848
lsu_b3_c6_p4	0.0015756	0.00135051	0.0139553	0.00258848
lsu_ch_c6_p4	0.0015756	0.00292611	0.01755667	0.003939
in_ver_c6_p4	0.00146306	0.00292611	0.01609361	0.003939
in_hor_c6_p4	0.0015756	0.00292611	0.01451801	0.003939
inter_ver_p4	0.0015756	0.0031512	0.01755667	0.00686511
inter_hor_p4	0.00146306	0.0031512	0.01609361	0.00686511
fpu_b1_c7_p4	0.0015756	0.0031512	0.01451801	0.00686511
fpu_b2_c7_p4	0.0015756	0.00337628	0.01755667	0.0100163
fpu_rg_c7_p4	0.00146306	0.00337628	0.01609361	0.0100163
isu_al_c7_p4	0.0015756	0.00337628	0.01451801	0.0100163
fxu_r1_c7_p4	0.00056271	0.00945359	0.0139553	0.003939
fxu_b1_c7_p4	0.00045017	0.00427662	0.01913227	0.02070787
fxu_b2_c7_p4	0.00607731	0.00067526	0.0163187	0.02003261
fxu_r2_c7_p4	0.00135051	0.0015756	0.02340889	0.01845701
in_rit_c7_p4	0.00135051	0.00202577	0.02340889	0.01643124
idu_al_c7_p4	0.00135051	0.0015756	0.02340889	0.01485564
ifu_ch_c7_p4	0.00292611	0.0015756	0.02048278	0.01845701

Appendix Table 3

MCM POWER4 detailed block Floor plan 45nm

Name	Width	Height	Left-X	Bottom-Y
ifu_b1_c7_p4	0.00292611	0.00146306	0.02048278	0.01699395
ifu_b2_c7_p4	0.00292611	0.0015756	0.02048278	0.01541836
ifu_b3_c7_p4	0.0031512	0.0015756	0.01733158	0.01845701
lsu_nt_c7_p4	0.0031512	0.00146306	0.01733158	0.01699395
lsu_cr_c7_p4	0.0031512	0.0015756	0.01733158	0.01541836
lsu_b1_c7_p4	0.00337628	0.0015756	0.0139553	0.01845701
lsu_b2_c7_p4	0.00337628	0.00146306	0.0139553	0.01699395
lsu_b3_c7_p4	0.00337628	0.0015756	0.0139553	0.01541836
lsu_ch_c7_p4	0.00945359	0.00056271	0.0139553	0.01485564
in_ver_c7_p4	0.00427662	0.00045017	0.00202577	0.02003261
in_hor_c7_p4	0.00067526	0.00607731	0.00630239	0.01721904
L3_con_b1_p4	0.0015756	0.00135051	0.00697765	0.02430923
L3_con_b2_p4	0.00202577	0.00135051	0.00855325	0.02430923
L3_con_b3_p4	0.0015756	0.00135051	0.01057902	0.02430923
L2_aaa_b1_p4	0.0015756	0.00292611	0.00697765	0.02138312
L2_aaa_b2_p4	0.00146306	0.00292611	0.00855325	0.02138312
L2_aaa_b3_p4	0.0015756	0.00292611	0.0100163	0.02138312
L2_bbb_b1_p4	0.0015756	0.0031512	0.00697765	0.01823192
L2_bbb_b2_p4	0.00146306	0.0031512	0.00855325	0.01823192
L2_bbb_b3_p4	0.0015756	0.0031512	0.0100163	0.01823192
L2_ccc_b1_p4	0.0015756	0.0033762	0.0069776	0.01485564

Continued

Appendix Table 3

MCM POWER4 detailed block Floor plan 45nm

Name	Width	Height	Left-X	Bottom-Y
		8	5	
L2_ccc_b2_p4	0.00146306	0.00337628	0.00855325	0.01485564
L2_ccc_b3_p4	0.0015756	0.00337628	0.0100163	0.01485564
inter_bot_p4	0.00056271	0.00945359	0.0115919	0.01485564
inter_cross1	0.00180068	0.02307126	0.01215462	0.00258848
inter_cross2	0.02340889	0.00146306	0.00135051	0.01339259

Appendix Table 4

Fuzzy rules space contains 120 rules

No	If	Variable A	And	Variable B	Then	Delta U
1	if	temperature is maximum 80c	and	temperature deviation is increasing	then	VLP
2	if	temperature is maximum 80c	and	temperature deviation is decreasing	then	LP
3	if	temperature is maximum 80c	and	core frequency is minimum	then	VLP
4	if	temperature is maximum 80c	and	core frequency is very low	then	LP
5	if	temperature is maximum 80c	and	core frequency is low	then	LP
6	if	temperature is maximum 80c	and	core frequency is high	then	LP
7	if	temperature is maximum 80c	and	core frequency is very high	then	LP
8	if	temperature is maximum 80c	and	core frequency is maximum	then	VLP
9	if	temperature is maximum 80c	and	core frequency deviation is increasing	then	VLP
10	if	temperature is maximum 80c	and	core frequency deviation is decreasing	then	LP
11	if	temperature is very high	and	temperature deviation is increasing	then	VLP
12	if	temperature is very high	and	temperature deviation is decreasing	then	LP
13	if	temperature is very high	and	core frequency is minimum	then	LP
14	if	temperature is very high	and	core frequency is very low	then	LP
15	if	temperature is very high	and	core frequency is low	then	SP
16	if	temperature is very high	and	core frequency is high	then	SP
17	if	temperature is very high	and	core frequency is very high	then	LP

Continued

Appendix Table 4

Fuzzy rules space contains 120 rules

No	If	Variable A	And	Variable B	Then	Delta U
18	if	temperature is very high	and	core frequency is maximum	then	VLP
19	if	temperature is very high	and	core frequency deviation is increasing	then	VLP
20	if	temperature is very high	and	core frequency deviation is decreasing	then	LP
21	if	temperature is high	and	temperature deviation is increasing	then	LP
22	if	temperature is high	and	temperature deviation is decreasing	then	SP
23	if	temperature is high	and	core frequency is minimum	then	LP
24	if	temperature is high	and	core frequency is very low	then	SP
25	if	temperature is high	and	core frequency is low	then	SP
26	if	temperature is high	and	core frequency is high	then	SP
27	if	temperature is high	and	core frequency is very high	then	SP
28	if	temperature is high	and	core frequency is maximum	then	LP
29	if	temperature is high	and	core frequency deviation is increasing	then	LP
30	if	temperature is high	and	core frequency deviation is decreasing	then	SP
31	if	temperature is low	and	temperature deviation is increasing	then	VSP
32	if	temperature is low	and	temperature deviation is decreasing	then	VSP
33	if	temperature is low	and	core frequency is minimum	then	VSP
34	if	temperature is low	and	core frequency is very low	then	VSP
35	if	temperature is low	and	core frequency is low	then	VSP
36	if	temperature is low	and	core frequency is high	then	VSP
37	if	temperature is low	and	core frequency is very high	then	VSP
38	if	temperature is low	and	core frequency is maximum	then	VSP
39	if	temperature is low	and	core frequency deviation is increasing	then	VSP
40	if	temperature is low	and	core frequency deviation is decreasing	then	VSP
41	if	temperature is very low	and	temperature deviation is increasing	then	VSP
42	if	temperature is very low	and	temperature deviation is decreasing	then	VSP
43	if	temperature is very low	and	core frequency is minimum	then	VSP
44	if	temperature is very low	and	core frequency is very low	then	VSP
45	if	temperature is very low	and	core frequency is low	then	VSP
46	if	temperature is very low	and	core frequency is high	then	VSP
47	if	temperature is very low	and	core frequency is very high	then	VSP
48	if	temperature is very low	and	core frequency is maximum	then	VSP
49	if	temperature is very low	and	core frequency deviation is increasing	then	VSP

Continued

Appendix Table 4

Fuzzy rules space contains 120 rules

No	If	Variable A	And	Variable B	Then	Delta U
50	if	temperature is very low	and	core frequency deviation is decreasing	then	VSP
51	if	temperature deviation is increasing	and	core frequency is minimum	then	LP
52	if	temperature deviation is increasing	and	core frequency is very low	then	LP
53	if	temperature deviation is increasing	and	core frequency is low	then	SP
54	if	temperature deviation is increasing	and	core frequency is high	then	SP
55	if	temperature deviation is increasing	and	core frequency is very high	then	SP
56	if	temperature deviation is increasing	and	core frequency is maximum	then	LP
57	if	temperature deviation is increasing	and	core frequency deviation is increasing	then	LP
58	if	temperature deviation is increasing	and	core frequency deviation is decreasing	then	SP
59	if	temperature deviation is decreasing	and	core frequency is minimum	then	SP
60	if	temperature deviation is decreasing	and	core frequency is very low	then	SP
61	if	temperature deviation is decreasing	and	core frequency is low	then	SP
62	if	temperature deviation is decreasing	and	core frequency is high	then	SP
63	if	temperature deviation is decreasing	and	core frequency is very high	then	SP
64	if	temperature deviation is decreasing	and	core frequency is maximum	then	SP
65	if	temperature deviation is decreasing	and	core frequency deviation is increasing	then	SP
66	if	temperature deviation is decreasing	and	core frequency deviation is decreasing	then	SP
67	if	core frequency is minimum	and	core frequency deviation is increasing	then	VSP
68	if	core frequency is minimum	and	core frequency deviation is decreasing	then	SP
69	if	core frequency is very low	and	core frequency deviation is increasing	then	VSP
70	if	core frequency is very low	and	core frequency deviation is decreasing	then	SP
71	if	core frequency is low	and	core frequency deviation is increasing	then	VSP
72	if	core frequency is low	and	core frequency deviation is decreasing	then	SP
73	if	core frequency is high	and	core frequency deviation is increasing	then	VSP
74	if	core frequency is high	and	core frequency deviation is decreasing	then	VSP
75	if	core frequency is very high	and	core frequency deviation is increasing	then	VSP
76	if	core frequency is very high	and	core frequency deviation is decreasing	then	VSP
77	if	core frequency is maximum	and	core frequency deviation is increasing	then	VSP
78	if	core frequency is maximum	and	core frequency deviation is decreasing	then	VSP

Continued

Appendix Table 4

Fuzzy rules space contains 120 rules

No	If	Variable A	And	Variable B	Then	Delta U
79	if	core frequency is minimum	and	temperature is maximum 80c	then	DUPLICATED RULE
80	if	core frequency is minimum	and	temperature is very high	then	DUPLICATED RULE
81	if	core frequency is minimum	and	temperature is high	then	DUPLICATED RULE
82	if	core frequency is minimum	and	temperature is low	then	DUPLICATED RULE
83	if	core frequency is minimum	and	temperature is very low	then	DUPLICATED RULE
84	if	core frequency is minimum	and	temperature deviation is increasing	then	DUPLICATED RULE
85	if	core frequency is minimum	and	temperature deviation is decreasing	then	DUPLICATED RULE
86	if	core frequency is very low	and	temperature is maximum 80c	then	DUPLICATED RULE
87	if	core frequency is very low	and	temperature is very high	then	DUPLICATED RULE
88	if	core frequency is very low	and	temperature is high	then	DUPLICATED RULE
89	if	core frequency is very low	and	temperature is low	then	DUPLICATED RULE
90	if	core frequency is very low	and	temperature is very low	then	DUPLICATED RULE
91	if	core frequency is very low	and	temperature deviation is increasing	then	DUPLICATED RULE
92	if	core frequency is very low	and	temperature deviation is decreasing	then	DUPLICATED RULE
93	if	core frequency is low	and	temperature is maximum 80c	then	DUPLICATED RULE
94	if	core frequency is low	and	temperature is very high	then	DUPLICATED RULE
95	if	core frequency is low	and	temperature is high	then	DUPLICATED RULE
96	if	core frequency is low	and	temperature is low	then	DUPLICATED RULE
97	if	core frequency is low	and	temperature is very low	then	DUPLICATED RULE
98	if	core frequency is low	and	temperature deviation is increasing	then	DUPLICATED RULE
99	if	core frequency is low	and	temperature deviation is decreasing	then	DUPLICATED RULE
100	if	core frequency is high	and	temperature is maximum 80c	then	DUPLICATED RULE
101	if	core frequency is high	and	temperature is very high	then	DUPLICATED RULE
102	if	core frequency is high	and	temperature is high	then	DUPLICATED RULE
103	if	core frequency is high	and	temperature is low	then	DUPLICATED RULE
104	if	core frequency is high	and	temperature is very low	then	DUPLICATED RULE
105	if	core frequency is high	and	temperature deviation is increasing	then	DUPLICATED RULE
106	if	core frequency is high	and	temperature deviation is decreasing	then	DUPLICATED RULE
107	if	core frequency is very high	and	temperature is maximum 80c	then	DUPLICATED RULE

Continued

Appendix Table 4

Fuzzy rules space contains 120 rules

No	If	Variable A	And	Variable B	Then	Delta U
108	if	core frequency is very high	and	temperature is very high	then	DUPLICATED RULE
109	if	core frequency is very high	and	temperature is high	then	DUPLICATED RULE
110	if	core frequency is very high	and	temperature is low	then	DUPLICATED RULE
111	if	core frequency is very high	and	temperature is very low	then	DUPLICATED RULE
112	if	core frequency is very high	and	temperature deviation is increasing	then	DUPLICATED RULE
113	if	core frequency is very high	and	temperature deviation is decreasing	then	DUPLICATED RULE
114	if	core frequency is maximum	and	temperature is maximum 80c	then	DUPLICATED RULE
115	if	core frequency is maximum	and	temperature is very high	then	DUPLICATED RULE
116	if	core frequency is maximum	and	temperature is high	then	DUPLICATED RULE
117	if	core frequency is maximum	and	temperature is low	then	DUPLICATED RULE
118	if	core frequency is maximum	and	temperature is very low	then	DUPLICATED RULE
119	if	core frequency is maximum	and	temperature deviation is increasing	then	DUPLICATED RULE
120	if	core frequency is maximum	and	temperature deviation is decreasing	then	DUPLICATED RULE
1	if	temperature is maximum 80c	and	temperature deviation is increasing	then	VLP
2	if	temperature is maximum 80c	and	temperature deviation is decreasing	then	LP
3	if	temperature is maximum 80c	and	core frequency is minimum	then	VLP
4	if	temperature is maximum 80c	and	core frequency is very low	then	LP
5	if	temperature is maximum 80c	and	core frequency is low	then	LP
6	if	temperature is maximum 80c	and	core frequency is high	then	LP
7	if	temperature is maximum 80c	and	core frequency is very high	then	LP
8	if	temperature is maximum 80c	and	core frequency is maximum	then	VLP
9	if	temperature is maximum 80c	and	core frequency deviation is increasing	then	VLP
10	if	temperature is maximum 80c	and	core frequency deviation is decreasing	then	LP
11	if	temperature is very high	and	temperature deviation is increasing	then	VLP
12	if	temperature is very high	and	temperature deviation is decreasing	then	LP
13	if	temperature is very high	and	core frequency is minimum	then	LP
14	if	temperature is very high	and	core frequency is very low	then	LP
15	if	temperature is very high	and	core frequency is low	then	SP
16	if	temperature is very high	and	core frequency is high	then	SP
17	if	temperature is very high	and	core frequency is very high	then	LP

Continued

Appendix Table 4

Fuzzy rules space contains 120 rules

No	If	Variable A	And	Variable B	Then	Delta U
18	if	temperature is very high	and	core frequency is maximum	then	VLP
19	if	temperature is very high	and	core frequency deviation is increasing	then	VLP
20	if	temperature is very high	and	core frequency deviation is decreasing	then	LP
21	if	temperature is high	and	temperature deviation is increasing	then	LP
22	if	temperature is high	and	temperature deviation is decreasing	then	SP
23	if	temperature is high	and	core frequency is minimum	then	LP
24	if	temperature is high	and	core frequency is very low	then	SP
25	if	temperature is high	and	core frequency is low	then	SP
26	if	temperature is high	and	core frequency is high	then	SP
27	if	temperature is high	and	core frequency is very high	then	SP
28	if	temperature is high	and	core frequency is maximum	then	LP
29	if	temperature is high	and	core frequency deviation is increasing	then	LP
30	if	temperature is high	and	core frequency deviation is decreasing	then	SP
31	if	temperature is low	and	temperature deviation is increasing	then	VSP
32	if	temperature is low	and	temperature deviation is decreasing	then	VSP
33	if	temperature is low	and	core frequency is minimum	then	VSP
34	if	temperature is low	and	core frequency is very low	then	VSP
35	if	temperature is low	and	core frequency is low	then	VSP
36	if	temperature is low	and	core frequency is high	then	VSP
37	if	temperature is low	and	core frequency is very high	then	VSP
38	if	temperature is low	and	core frequency is maximum	then	VSP
39	if	temperature is low	and	core frequency deviation is increasing	then	VSP
40	if	temperature is low	and	core frequency deviation is decreasing	then	VSP
41	if	temperature is very low	and	temperature deviation is increasing	then	VSP
42	if	temperature is very low	and	temperature deviation is decreasing	then	VSP
43	if	temperature is very low	and	core frequency is minimum	then	VSP
44	if	temperature is very low	and	core frequency is very low	then	VSP
45	if	temperature is very low	and	core frequency is low	then	VSP
46	if	temperature is very low	and	core frequency is high	then	VSP
47	if	temperature is very low	and	core frequency is very high	then	VSP
48	if	temperature is very low	and	core frequency is maximum	then	VSP
49	if	temperature is very low	and	core frequency deviation is increasing	then	VSP

Continued

Appendix Table 4

Fuzzy rules space contains 120 rules

No	If	Variable A	And	Variable B	Then	Delta U
50	if	temperature is very low	and	core frequency deviation is decreasing	then	VSP
51	if	temperature deviation is increasing	and	core frequency is minimum	then	LP
52	if	temperature deviation is increasing	and	core frequency is very low	then	LP
53	if	temperature deviation is increasing	and	core frequency is low	then	SP
54	if	temperature deviation is increasing	and	core frequency is high	then	SP
55	if	temperature deviation is increasing	and	core frequency is very high	then	SP
56	if	temperature deviation is increasing	and	core frequency is maximum	then	LP
57	if	temperature deviation is increasing	and	core frequency deviation is increasing	then	LP
58	if	temperature deviation is increasing	and	core frequency deviation is decreasing	then	SP
59	if	temperature deviation is decreasing	and	core frequency is minimum	then	SP
60	if	temperature deviation is decreasing	and	core frequency is very low	then	SP
61	if	temperature deviation is decreasing	and	core frequency is low	then	SP
62	if	temperature deviation is decreasing	and	core frequency is high	then	SP
63	if	temperature deviation is decreasing	and	core frequency is very high	then	SP
64	if	temperature deviation is decreasing	and	core frequency is maximum	then	SP
65	if	temperature deviation is decreasing	and	core frequency deviation is increasing	then	SP
66	if	temperature deviation is decreasing	and	core frequency deviation is decreasing	then	SP
67	if	core frequency is minimum	and	core frequency deviation is increasing	then	VSP
68	if	core frequency is minimum	and	core frequency deviation is decreasing	then	SP
69	if	core frequency is very low	and	core frequency deviation is increasing	then	VSP
70	if	core frequency is very low	and	core frequency deviation is decreasing	then	SP
71	if	core frequency is low	and	core frequency deviation is increasing	then	VSP
72	if	core frequency is low	and	core frequency deviation is decreasing	then	SP
73	if	core frequency is high	and	core frequency deviation is increasing	then	VSP
74	if	core frequency is high	and	core frequency deviation is decreasing	then	VSP
75	if	core frequency is very high	and	core frequency deviation is increasing	then	VSP
76	if	core frequency is very high	and	core frequency deviation is decreasing	then	VSP
77	if	core frequency is maximum	and	core frequency deviation is increasing	then	VSP
78	if	core frequency is	and	core frequency	then	VSP

Continued

Appendix Table 4

Fuzzy rules space contains 120 rules

No	If	Variable A	And	Variable B	Then	Delta U
		maximum		deviation is decreasing		
79	if	core frequency is minimum	and	temperature is maximum 80c	then	DUPLICATED RULE
80	if	core frequency is minimum	and	temperature is very high	then	DUPLICATED RULE
81	if	core frequency is minimum	and	temperature is high	then	DUPLICATED RULE
82	if	core frequency is minimum	and	temperature is low	then	DUPLICATED RULE
83	if	core frequency is minimum	and	temperature is very low	then	DUPLICATED RULE
84	if	core frequency is minimum	and	temperature deviation is increasing	then	DUPLICATED RULE
85	if	core frequency is minimum	and	temperature deviation is decreasing	then	DUPLICATED RULE
86	if	core frequency is very low	and	temperature is maximum 80c	then	DUPLICATED RULE
87	if	core frequency is very low	and	temperature is very high	then	DUPLICATED RULE
88	if	core frequency is very low	and	temperature is high	then	DUPLICATED RULE
89	if	core frequency is very low	and	temperature is low	then	DUPLICATED RULE
90	if	core frequency is very low	and	temperature is very low	then	DUPLICATED RULE
91	if	core frequency is very low	and	temperature deviation is increasing	then	DUPLICATED RULE
92	if	core frequency is very low	and	temperature deviation is decreasing	then	DUPLICATED RULE
93	if	core frequency is low	and	temperature is maximum 80c	then	DUPLICATED RULE
94	if	core frequency is low	and	temperature is very high	then	DUPLICATED RULE
95	if	core frequency is low	and	temperature is high	then	DUPLICATED RULE
96	if	core frequency is low	and	temperature is low	then	DUPLICATED RULE
97	if	core frequency is low	and	temperature is very low	then	DUPLICATED RULE
98	if	core frequency is low	and	temperature deviation is increasing	then	DUPLICATED RULE
99	if	core frequency is low	and	temperature deviation is decreasing	then	DUPLICATED RULE
100	if	core frequency is high	and	temperature is maximum 80c	then	DUPLICATED RULE
101	if	core frequency is high	and	temperature is very high	then	DUPLICATED RULE
102	if	core frequency is high	and	temperature is high	then	DUPLICATED RULE
103	if	core frequency is high	and	temperature is low	then	DUPLICATED RULE
104	if	core frequency is high	and	temperature is very low	then	DUPLICATED RULE
105	if	core frequency is high	and	temperature deviation is increasing	then	DUPLICATED RULE

Appendix Table 5

3D Fuzzy Correlation Distance

The Correlation Distance label	**Value**
hotspot distance inverse between c0 and c0	1
hotspot distance inverse between c0 and c1	40
hotspot distance inverse between c0 and c2	146
hotspot distance inverse between c0 and c3	150
hotspot distance inverse between c0 and c4	205
hotspot distance inverse between c0 and c5	177
hotspot distance inverse between c0 and c6	144
hotspot distance inverse between c0 and c7	121
hotspot distance inverse between c1 and c0	40
hotspot distance inverse between c1 and c1	1
hotspot distance inverse between c1 and c2	106
hotspot distance inverse between c1 and c3	111
hotspot distance inverse between c1 and c4	177
hotspot distance inverse between c1 and c5	155
hotspot distance inverse between c1 and c6	148
hotspot distance inverse between c1 and c7	131
hotspot distance inverse between c2 and c0	146
hotspot distance inverse between c2 and c1	106
hotspot distance inverse between c2 and c2	1
hotspot distance inverse between c2 and c3	40
hotspot distance inverse between c2 and c4	143
hotspot distance inverse between c2 and c5	147
hotspot distance inverse between c2 and c6	205
hotspot distance inverse between c2 and c7	203
hotspot distance inverse between c3 and c0	150
hotspot distance inverse between c3 and c1	111
hotspot distance inverse between c3 and c2	40
hotspot distance inverse between c3 and c3	1
hotspot distance inverse between c3 and c4	103
hotspot distance inverse between c3 and c5	108
hotspot distance inverse between c3 and c6	178
hotspot distance inverse between c3 and c7	181
hotspot distance inverse between c4 and c0	205
hotspot distance inverse between c4 and c1	177
hotspot distance inverse between c4 and c2	143
hotspot distance inverse between c4 and c3	103
hotspot distance inverse between c4 and c4	1
hotspot distance inverse between c4 and c5	40
hotspot distance inverse between c4 and c6	147

Continued

Appendix Table 5

3D Fuzzy Correlation Distance

The Correlation Distance label	**Value**
hotspot distance inverse between c4 and c7	167
hotspot distance inverse between c5 and c0	177
hotspot distance inverse between c5 and c1	155
hotspot distance inverse between c5 and c2	146
hotspot distance inverse between c5 and c3	108
hotspot distance inverse between c5 and c4	40
hotspot distance inverse between c5 and c5	1
hotspot distance inverse between c5 and c6	107
hotspot distance inverse between c5 and c7	127
hotspot distance inverse between c6 and c0	144
hotspot distance inverse between c6 and c1	148
hotspot distance inverse between c6 and c2	205
hotspot distance inverse between c6 and c3	178
hotspot distance inverse between c6 and c4	147
hotspot distance inverse between c6 and c5	107
hotspot distance inverse between c6 and c6	1
hotspot distance inverse between c6 and c7	40
hotspot distance inverse between c7 and c0	121
hotspot distance inverse between c7 and c1	131
hotspot distance inverse between c7 and c2	203
hotspot distance inverse between c7 and c3	181
hotspot distance inverse between c7 and c4	166
hotspot distance inverse between c7 and c5	127
hotspot distance inverse between c7 and c6	40
hotspot distance inverse between c7 and c7	1

References:

[1] S. W. Chung and K. Skadron, "Using on-chip event counters for high-resolution, real-time temperature measurements", International Conference For Scientific and Engineering Exploration Of Thermal, Thermomechanical and Emerging Technology, IEEE ITHERM06, 2006, Pages 114-120.

[2] R. Rao and S. Vrudhula, "Performance Optimal Processor Throttling Under Thermal Constraints", International Conference On Compilers, Architecture, and Synthesis For Embedded Systems, CASES’07, 2007, Pages 257-266.

[3] W. Huangy, M. R. Stany, K. Skadronz, K. Sankaranarayananz, S. Ghoshyz, and S VelUSAmyz, "Hotspot: A Compact Thermal Modeling Methodology For Early-Stage Vlsi Design", Very Large Scale Integration Systems, IEEE Transactions, 2006, Volume 5, Pages 501-513.

[4] ITRS 2006 Executive Summary. http://www.ITRS.Net/Links/2006update/Finaltopost/00_Execsum2006update.pdf

[5] D.D. Kim, J. Kim, C. Cho, J.O. Plouchart, and R. Trzcinski "65nm SOI CMOS SoC Technology for Low-Power mmWave and RF Platform", Silicon Monolithic Integrated Circuits in RF Systems, 2008, Pages 46-49.

[6] P. Mccrorie, "On-Chip Thermal Analysis Is Becoming Mandatory", 2008, Chip Design Magazine.

[7] L. Shang, and R.P. Dick, "Thermal Crisis: Challenges And Potential Solutions", IEEE Potentials, 2006, Volume 25, Number 5, Pages 31-35.

[8] J. Donald and M. Martonosi, "Techniques For Multicore Thermal Management Classification And New Exploration", International Symposium on Computer Architecture, IEEE ISCA’06, 2006, Pages 78-88.

[9] P. Chaparro, J. G. Lez, G. M., Q. Cai, and A. G. Lez, "Understanding

The Thermal Implications Of Multicore Architectures", IEEE Transactions, 2007, Volume 18, Number 8, Pages 1055-1065.

[10] S. Kim, R. P. Dick, and R. Joseph, “Power Deregulation: Eliminating Off-Chip Voltage Regulation Circuitry From Embedded Systems”, The International Conference on Hardware-Software Codesign and System Synthesis, IEEE/ACM (CODES+ISSS), 2007, Pages 105-110.

[11] A. Trabelsi, F. Lafont, M. Kamoun, and G. Enea, "Identification Of Nonlinear Multivariable Systems By Adaptive Fuzzy Takagi-Sugeno Model", International Journal of Computational Cognition, 2004, Volume 2, Number 3, Pages 137-153.

[12] U.Y. Ogras et Al. “Variation-Adaptive Feedback Control for Networks-on-Chip with Multiple Clock Domains”, International Conference on Design Automation Conference, IEEE DAC, 2008, Pages 614-619.

[13] Q. Wu et Al. “Formal online methods for voltage/frequency control in multiple clock domain microprocessors”, International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS, 2004, Volume 32, Number 5, Pages 248-259.

[14] "IBM E-Server Pseries 670 And Pseries 690 System Handbook", IBM Press, May 2003.

[15] M.J. Patyra, J.L. Grantner, and K. Koster, "Digital Fuzzy Logic Controller: Design And Implementation", Fuzzy Systems, IEEE Transactions, 1996, Volume 4, Issue 4, Pages 439-459.

[16] H. Li, X. Zhang, and S. Li, "A Three-Dimensional Fuzzy Control Methodology For A Class Of Distributed Parameter Systems", Fuzzy Systems, IEEE Transactions, 2007, Volume 15, Issue 3, Pages 470-481.

[17] D. Suleiman, M. Ibrahim, and I. Hamarash,"Dynamic Voltage Frequency Scaling (DVFS) For Microprocessors Power and Energy Reduction", International Conference on Electrical and Electronics Engineering, ELECO'05, 2005, Bursa, Turkey.

[18] S. M. F. Bin Syed Adrus, M. H. Bin Abdullah, and M. Rusop, "Electrical analysis of 65 nm PMOS based on SOI technology", International Conference on Nanoscience and Nanotechnology, AIP, 2009, Volume 1136, Pgaes 775-780.

[19] I. Ionica, L. Montès, S. Ferraton, J. Zimmermann, V. Bouchiat, and L.

Saminadayar, "Silicon Nanostructures Patterned on SOI by AFM Lithography", Nano Science and Technology Institute Nanotechnology Conference, NSTI Nanotech, 2004, Volume 3, Page 165.

[20] G. E. Moore, "Cramming More Components Onto Integrated Circuits", IEEE Electronics, 19 April 1965, Volume 38, Number 8 Pages114. This Paper Appears Again In IEEE Solid-State Circuits Newsletter, 2006, Volume 20, Issue 3, Pages 33-35.

[21] T. Kuroda, "CMOS Design Challenges To Power Wall", Microprocesses and Nanotechnology Conference, IEEE MNC, 2001, Shimane, Japan Pages 6-7.

[22] ITRS 2008 update http://www.itrs.net/Links/2008ITRS/Update/2008_Update.pdf

[23] N.E.Jewell-Larsen, H. Ran, Y. Zhang, M.K. Schwiebert, K.A.H. Tessera, and A.V. Mamishev,"Electrohydrodynamic (EHD) cooled laptop", Semiconductor Thermal Management, IEEE SEMI-THERM, 2009, Pages 261-266.

[24] M. Stern,D. Copeland, M. Vogel, J. Dunn, D. Kearns, S. Lindquist, and C.A. Santa Clara,"Preliminary specification for a closed loop liquid cooling system product reliability test plan", Semiconductor Thermal Management, IEEE SEMI-THERM, 2009, Pages 159-163.

[25] R. Gonzalez, and M. Horowitz, "Energy Dissipation In General Purpose Microprocessors", IEEE Journal of Solid-State Circuits,1996, Volume 31, Number 9, Pages 1277-1284.

[26] H. J. M. Veendrick,"Short-Circuit Dissipation of Static Cmos Circuitry And Its Impact On The Design of Buffer Circuits", IEEE Journal of Solid-State Circuits, 1984, Volume19, Numver. 4, Pages 468-473

[27] D. Liu, and C. Svensson, “Trading Speed For Low Power By Choice of Supply and Threshold Voltages”, IEEE Journal of Solid-State Circuits, January 1993, Volume 28, Number 1, Pages 10-17.

[28] R. Rao, S. Vrudhula, and C. Chakrabarti, "Throughput Of Multicore Processors Under Thermal Constraints" Presentation, International Symposium on Low Power Electronics and Design, ISLPED’07, 2007,

Portland, Oregon, USA.

[29] K. M. Passino, and S. Yurkovich, "Fuzzy Control" Text Book, Publish By Addison Wesley Longman, 1998.

[30] V. Venkatachalam, "Self-Calibrating Processor Speed: A New Feedback Loop For Dynamic Voltage Scaling Control", Doctor of philosophy in Information and Computer Science, 2007, Pages 7-20, University of California, USA.

[31] Intel Corporation, “Pollack's Rule”
www.intel.com/research/mrl/Library/pactkeynote.pdf
Available on line at
http://www.itrs.net/Links/2001ITRS/Links/design/pollack.html

[32] S. Borkar, “Thousand Core Chips: A Technology Perspective”, Proceedings of the 44th ACM/IEEE Design Automation Conference DAC, 2007, Pages 746-749.

[33] AMD Corporation, "AMD Completes ATI Acquisition and Creates Processing Powerhouse". NewsWire. 25 October 2006 article,
Available on line at
http://www10.dcccafe.com/nbc/articles/view_article.php?
section=CorpNews&articleid=317343

[34] http://www.nvidia.com/

[35] http://www.sun.com/

[36] B. McCredie, "POWER Roadmap", IBM Systems &Technology Group, Development IBM Fellow, http://www.IBM.com/

[37] B. Sinharoy, R. N. Kalla, J. M. Tendler, and R. J. Eickemeyer, "POWER5 System Microarchitecture", IBM J. Res. & Dev. Volume 49 No. 4/5 July/September 2005.

[38] http://en.Wikipedia.rg/Wiki/

[39] IBM Corporation, "POWER7 IBM's next generation server processor", Power Architecture Conference, 2009 ASIA PAC, 19 October 2009 - Hsinchu, Taiwan.
http://www.IBM.com/

[40] Yamauchi, Tadaaki , Arimoto, and Kazutami “CMOS substrate biasing for threshold voltage control” , United States Patent 5838047
http://www.Freepatentsonline.com/5838047.html

[41] B. Amelifard, F. Fallah, and M. Pedarm,"Low-Power Fanout

Optimization Using Mtcmos And Multi-Vt Techniques", Low Power Electronics and Design, ISLPED'06, 2006, Pages 334-337.

[42] http://en.Wikipedia.Org/Wiki/Multi-Threshold_CMOS

[43] S. Velusamy, W. Huang, J. Lach , M. Stan, and K. Skadron, "Monitoring Temperature In FPGA Based SOCS", International Conference on Computer Design, IEEE ICCD, 2005, Pages 634- 637.

[44] S. Oktay ,"Parametric Study of Temperature Profiles In Chips Joined By Controlled Collapse Techniques", IBM j. Res. Develop., Electronic Components Conference, IEEE (G-PMP) and Electronic Industries Association, May 1969, Volume 13, Pages 272-285.

[45] D. Edwards, "IC Package Thermal Metrics", Texas Instruments, Application Report SPRA953A, June 2007.

[46] E. Humenay, D. Tarjan, and K. Skadron,"Impact Of Process Variations On Multicore Performance Symmetry", Design, Automation & Test In Europe Conference & Exhibition, DATE '07, 2007, Pages 1-6.

[47] Y.Chang,"Recent Development In VLSI floorplan Representations", Course at The National Taiwan University, Taipei, Taiwan, 2001.

[48] M. Pedram, "Design Technologies For Low-Power VLSI", In Encyclopedia of Computer Science And Technol., Volume 36, 1997, Pages 73-96, Marcel Dekker, New York.

[49] H. Im, T. Inukai, H. Gomyo, T. Hiramoto, and T. Sakurai, "VTCMOS characteristics and its optimum conditions predicted by a compact analytical model", Very Large Scale Integration (VLSI) Systems, IEEE Transactions, October 2003, Pages 755- 761.

[50] http://en.wikipedia.org/wiki/symmetric_multiprocessing

[51] http://en.wikipedia.org/wiki/asymmetric_multiprocessing

[52] A. Mendelson, "How Many Core Are Too Many?" Presentation, Intel Mobile Processors Architecture Groupe, HIPEAC, 2007, avialable online at http://www.haifa.ibm.com/workshops/compiler2007/present/avi_mendelson.pdf

[53] G. M. Amdahl. "Validity Of The Single-Processor Approach To Achieving Large Scale Computing Capabilities", American Federation

of Information Processing Societies, AFIPS, 1967, Pages 483–485.

[54] http://www.ansys.com

[55] http://www.mentor.com/products/mechanical/products/flotherm

[56] S. Narasimhan, and A. Yusuf, "Thermal Analysis and Modeling Techniques For Optimizing CPU Cooling Capability at The Component Level" Presentation, Intel Developer Forum, 2006, Session PTMS006.

[57] http://lava.cs.virginia.edu/hotspot/

[58] http://www.irisa.fr/caps/projects/atmi/atmimanual.pdf

[59] P. Michaud, and Y. Sazeides, "ATMI: Analytical Model Of Temperature In Microprocessors", Third Annual Workshop On Modeling, Benchmarking and Simulation, June 2007, available online at http://www.irisa.fr/caps/people/michaud/atmi_mobs.pdf.

[60] http://lava.cs.virginia.edu/hotleakage/

[61] T. Austin, E. Larson, and D. Ernst, “SimpleScalar: an infrastructure for computer system modeling,” IEEE Computer, 2002, Volume 35, Pages 59-67.

[62] W. Liao, L. He, and K. Lepak, “Temperature and supply Voltage aware performance and power modeling at microarchitecture level,” Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions, 2005, Volume 24, Pages 1042-1053.

[63] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework For Architectural-Level Power Analysis And Optimizations", International Symposium Computer Architecture, IEEE/ACM ISCA, 2000, Pages 83-94.

[64] S. Baldawa, and R. Sangireddy, "CMP-Sim: A Flexible CMP Architectural Simulation", High Performance Dependable Computing Laboratory, Department Of Electrical & Computer Engineering , University Of Texas At Dallas.

[65] http://www.utdallas.edu/~rama.sangireddy/cmp-sim

[66] http://www.ece.rochester.edu/research/acal/quilt/

[67] G. J. Briggs, E. J. Tan, N. A. Nelson, and D. H. Albonesi, "Quilt: A Gui-Based Integrated Circuit Floor planning Environment For Computer Architecture Research and Education", Workshop On Computer Architecture Education, WCAE '05, 2005, Number 5, Jun 2005, Pages 26-31, Madison, Wisconsin, Article.

[68] T. Chen, K. Von-Kyoung, and M. Tegethoff, "IC manufacturing test cost estimation at early stages of the design cycle", Microelectronics Journal, Volume 30, Number 8, August 1999 , Pages 733-738.

[69] R.W. Farrington,"Parametric cost estimating: a practical independent method of estimating the manufacturing cost of chips to modules in the Peoples Republic of China", Electronic Packaging Technology, 2005 6th International Conference, Pages 336- 337.

[70] T. Pitkanen, T. Rantanen, A. Cilio, and J. Takala, "Hardware Cost Estimation For Application-Specific Processor Design", Embedded Computer Systems: Architectures, Modeling, And Simulation, 5th International Workshop, SAMOS 2005, Greece, Pages 212-221.

[71] K. Diefendorff, “Power4 Focuses on Memory Bandwidth”, Microprocessor Report, October 1999, Volume 13, Number 13, Pages 11-17.

[72] J.U. Knickerbocker et al., "An Advanced Multichip Module (MCM) For High- Performance UNIX Servers", IBM Journal of Research and Development 2002, Volume 46, Number 6, Pages 779-804.

[73] "POWER5 System Microarchitecture", IBM Journal of Research and Development 2005, Volume 49, Number 4/5, Pages 505-521.

[74] E. Pop, R. Dutton, and K. Goodson,"Thermal Analysis Of Ultra-Thin Body Device Scaling", IEEE International Electron Devices Meeting, 2003, IEDM '03 Technical Digest, Pages 36.6.1- 36.6.4

[75] B. Swahn, and S. Hassoun, "Thermal and Sensitivity Analysis of Multi-Fin Devices", 12th International Workshop on Thermal investigations of ICs, THERMINIC 2006, Pages 229-234.

[76] E. Pop, and K. E. Goodson, "Thermal Phenomena In Nanoscale Transistors", The 19th Intersociety Conference of Thermal and Thermomechanical Phenomena in Electronic Systems, 2004, ITHERM '04, Volume1, Pages 1- 7.

[77] T. Ghani, M. Armstrong, C. Auth, M. Bost, P. Charvat, G. Glass, T. Hoffmann, K. Johnson, C. Kenyon, J. Klaus, B. McIntyre, K. Mistry, A. Murthy, J. Sandford, M. Silberstein, S. Sivakumar, P. Smith, K.

Zawadzki, S. Thompson, and M. Bohr, “A 90nm High Volume Manufacturing Logic Technology Featuring Novel 45nm Gate Length Strained Silicon CMOS Transistors”, IEDM Technical Digest, 2003, Pages 978-980.

[78] M. Kadin, and S. Reda, "Frequency And Voltage Planning For Multi-Core Processors Under Thermal Constraints", IEEE Computer Design, ICCD 2008, Pages 463-470.

[79] http://lava.cs.virginia.edu/hotspot/faq.htm

[80] I. Yeo, C. C. Liu, and E. J. Kim, "Temperature-Aware Scheduler Based On Thermal Behavior Grouping In Multicore Systems", Design Automation & Test In Europe, DATE 2009, Pages 946-951.

[81] A. P. Ferreira, D. Moss, and J. C. Oh, "Thermal Faults Modeling using an RC model with an Application to Web Farms", The 19th Euromicro Conference on Real-Time Systems, 2007, Italy, Pages 113-124.

[82] ITRS updates 2009
http://www.itrs.net/links/2009itrs/2009tables_ortc.xls and
http://public.itrs.net/Links/2009ITRS/2009Chapters_2009Tables/
2009_Sys Drivers.pdf

[83] J. L. Gustafson, “Re-Evaluating Amdahl’s Law”, ACM Communications, May 1988, Volume 31, Number 5, Pages 532-533.

[84] M. R. Stan, K. Skadron, M. Barcella, W. H. K. Sankaranarayanan, and S. Velusamy, "Hotspot: A Compact Thermal Modeling Methodology For Early-Stage VLSI Design", IEEE Transactions, May 2006, Volume 14, Issue 5, Pages 501-513.

[85] S. Boyd, and C. Barratt "Linear Controller Design: Limits Of Performance", Text Book, Prentice-Hall 1991.

[86] http://faculty.cs.tamu.edu/ejkim/hpc_web/dtm.html

[87] R. Yager, and D. Filev, "Essential Of Fuzzy Modeling And Control", Text Book, Wiley, New York 1994, Chapter 4 , Page 121.

[88] A. Naveh, E. Rotem, A. Mendelson, S. Gochman, R. Chabukswar, K. Krishnan, and A. Kumar, "Power And Thermal Management In The Intel Coretm Duo Processor", Intel Technology Journal, Volume 10, Issue 02, May, 2006.

[89] B. Vermeerschand and G. De Mey, “Non-fourier Thermal Conduction in Nano-scaled Electronic Devices”, 13th International Conference on

Mixed Design of Integrated Circuits & Systems, June 2006, Gdynia, Poland.

[90] C. C. Doumanidis, and N. Fourligkas, “Temperature Distribution Control In Scanned Thermal Processing Of Thin Circular Parts,” IEEE Transaction Control System Technolgy, May 2001, Volume 9, Number 5, Pages 708–717.

[91] P. D. Christofides, “Control of Nonlinear Distributed Process Systems: Recent Developments And Challenges”, AIChE Journal, Volume 47, Number 3, Pages 514–518.

[92] P. D. Christofides, “Nonlinear And Robust Control Of Partial Differential Equation Systems: Methods And Applications To Transport-Reaction Processes”, Systems and Control, Foundations and applications series, Page272, Birkhäuser, Boston, 2001.

[93] L. X. Wang, “A Course In Fuzzy Systems And Control”, Text Book, Prentice-Hall, 1997.

[94] A. Devgan, "Emerging Power And Timing Issues In High Performance ICs", Micron Technology Foundation, VLSI Seminar, 2003.

[95] S. Ankireddi, and D. Copeland, "Thermo-Mechanical Reliability Considerations With Dynamic Voltage Frequency Scaling In Microprocessor Applications", IEEE Semiconductor Thermal Measurement and Management Symposium, Semi-Therm 25th, 2009, Pages 134-138.

[96] www.intel.com

[97] I.Y.Kim, and O.L.de Weck, “Adaptive weighted-sum method for bi-objective optimization: Pareto front generation”, Structural and Multidisciplinary Optimization, February 2005, Volume29, Pages 149-158.