Microprocessor Thermal Analysis using the Finite Element Method

Bhavya Daya

Massachusetts Institute of Technology

Abstract— **The microelectronics industry is pursuing many options to sustain the performance improvement expected every two years. One method for performance improvement is scaling transistor sizes down such that many more transistors can be compacted on chip. The on-chip temperature is a concern because the reliability and performance can be degraded due to hot spots. Thermal modeling of the chip will allow the designer to view the hot spots and adjust the architecture to obtain a reliable chip architecture. In order to meet the performance demands of the current consumer market, the trend towards multicore processors is causing thermal effects to become increasingly important. The method of thermal analysis implementation is evaluated to determine the benefits of the different approaches. The finite element analysis was ultimately chosen and used to perform a case study on a microprocessor, and to evaluate different floorplans for multicore processors. These simple floorplan evaluations are a step towards obtaining a thermally-aware multicore chip platform that can be further evaluated once the detailed layout/floorplan has been established. Heat and hot spots are a challenge in the development of emerging 3D microprocessors. While these challenges are more applicable for high performance applications, designing the 3D chip's floorplan to take heat into account could prevent poor performance in 3D low power processors as well. Tackling thermal analysis for 3D microprocessors and creating thermally-aware 3D chips is the next step in microprocessor thermal analysis and design.**

Keywords— **Thermal Analysis, Finite Element, Microprocessor, Multicore, On-chip temperature**

I. Introduction

The microelectronics industry has been fulfilling Moore's law for 4 decades. To continually have a doubling of speed every 2 years is achieved by reducing the transistor size and increasing the transistor density on the chip. This allows more complex applications to be performed with a higher clock frequency. The current consumer market is constantly demand higher performance and the capability to run more complex applications. The fundamental limit of Moore's law has to occur because the transistor sizes cannot be scaled further than the atomic level.

Current measures to continue the performance improvement is to utilize parallelism. The introduction of the multicore processor is achieving these expectations but the on-chip temperature begins to become a more important concern. With the development on hot spots on the chip, the accuracy and reliability of certain on-chip functional blocks can tremendously suffer. The on chip temperature may contain spikes, but the steady state heat is of interest because the long term high temperature exposure can limit the operability and speed of the processor.

Fast and accurate thermal modeling tools are important for multicore processors or high performance microprocessors. Instead of analyzing the chip architecture after the development process has been completed, during the process chip floorplans can be tested in to determine the best option with the provided temperature constraints.

The thermal analysis of the chip is first described by a partial differential equation (PDE). Then the method of approximating a solution to that PDE is chosen by evaluating a few approaches. The choice of the finite element approach leads into a description of how the finite element method (FEM) works. The method of implementation is explained and the results of the functionality verification and convergence analysis of the FEM are presented. A case study on a commercial microprocessor is performed to view the thermal profile of the chip. The FEM is extended to the evaluation of multicore floorplans during the chip design process.

II. Partial Differential Equation For Thermal Chip Analysis

The hardware of a chip is made of a silicon die wrapped in packaging. Heat flow starts from the die and goes up to the heat spreader and ends at the environment or heat sink, spreaded by convection.

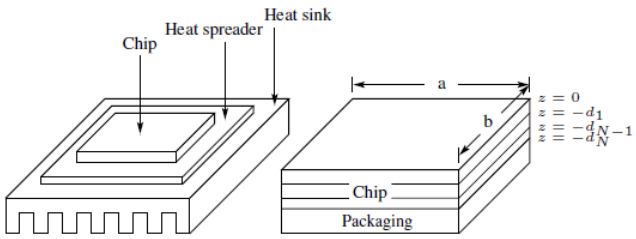

Fig. 1: IC Chip and Packaging [1]

The temperature distribution in a closed structure such as a processor packaged into a chip can be modeled by the heat conduction equation. [1] The general heat conduction equation in Cartesian coordinates is shown below, where U is the temperature, f is the heat generation rate, k is the thermal conductivity and α is the specific heat of the material.

$$\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2}\right) + \frac{f}{k} = \frac{1}{\alpha} * \frac{\partial t}{\partial \theta}$$

The special case when the 2D steady state heat conduction is considered, the equation reduces to Poisson's Equation:

$$\nabla^2 U(r) = -\frac{f(r)}{k(r)}$$

The thermal conductivity varies with temperature in a multilayer structure. Since the goal is to analyze the 2D chip architecture and provide a plot of the temperature on the chip, the multilayer structure is not considered. This leads to the thermal conductivity being a constant equal to the thermal conductivity of silicon.

Figure 1 shows a useful diagram of the heat sink and heat spreader when modeling the chip as a multilayer structure. The single layer approach taken for this project assumes only one silicon layer and doesn't include the bottom and top of the chip packaging. The boundary condition is dirichlet and the value was chosen to be zero along the four edges of the chip. It could have easily been set to an ambient temperature value.

III. Comparison Of Methods of Analysis

Obtaining an approximate solution to the PDE can be accomplished in a multitude of ways. The three methods chosen for comparison are finite differences, finite elements, and green's function. These methods can be grouped into grid-based and spectral methods.

The finite difference and finite elements methods are grid-based methods that have an advantage of modeling detailed chip geometries such as bonding wires, interlayer vias, and buses. The problem with these methods is the number of nodes required to accurately model a specific part of the chip, this leads to large systems to be solved. The large size of the system results in a long run time and limits its applications. If the chip contains different materials for different components, or the distribution of the thermal conductivity varies along the chip, the grid-based methods can tackle quite easily.

The Green's function based approaches is a spectral method that is based on the Fourier transform approach for chip thermal analysis. The Green's function assume that the heat sources considered are in a 2-D rectangular region, so the power-density spectrum can be computed by the fast Fourier transform (FFT). To apply the spectral methods, heat sources must be on the top surface of the chip, and heat transfer is forbidden there. The type of chip packaging dictates where the heat is transferred from. The wire bonding package satisfies the requirements, but the popular flip chip packaging causes heat to dissipate from the bottom and top, therefore the spectral method cannot be applied.

Green's function methods are suitable for computing the temperature distribution incurred by a planar heat source distribution; however, they cannot calculate the temperature distribution incurred by heat sources of arbitrary shapes. Compared to the grid-based methods, the Green's function-based methods are advantageous at the earlier stages of VLSI physical design flow, such as floorplanning and cell placement. The Green's function-based methods do not discretize the chip regions of no heat sources and of no monitored temperatures. Therefore, the Green's function-based methods improve the thermal simulation speed by a few orders of magnitude by not modeling otherwise costly chip regions such as the bulk. A summary of the surface comparison between the approaches is in Table 1. From the comparison, an approach can be chosen to undertake when performing the thermal analysis of a microprocessor.

Table 1: Comparison of Thermal Analysis Methods

Method	Advantages	Disadvantages
Grid Based (Finite Differences, Finite Elements)	Modeling detailed chip geometries such as inter-layer vias, buses, and bonding wires High flexibility in handling different kinds of boundary conditions High Accuracy	Large size causes a long run time and limits it applicability
Spectral (Green's function)	Obtains an approximate solution fast Applicable to floorplanning and cell placement steps of VLSI design flow	Cannot calculate the temperature distribution incurred by heat sources of arbitrary shapes Underestimates the temperature

From this rough comparison, the Green's function method has been eliminated as an approach that will be implemented. The reason is that the applicability of the Green's function method is limited by the package. This method also overlooks some aspects of the 2D chip region to obtain its speed advantage. The finite difference and finite element method have to be compared for one to be chosen.

To solve the steady state or time-independent heat conduction equation using the finite difference method, the Laplacian operator has to be discretized using a second order central finite difference scheme. The chip, whether in a multilayer structure or planar, can be seen as an RC network that consists of a matrix of hexagonal junctions, with resistors to model the heat conductivities, capacitors to model the thermal diffusivities, and current sources to model the power densities. [2]

The thermal domain, on which the steady state heat conduction is of interest, is broken up into many subdomains or elements. In FEM the representation of the differential equation is converted into an integrated form, which is then approximated using a selected set of trial functions defined in the domain. This results in a linear system of equations which could be solved to obtain an approximate solution to the differential equation. The linear system of equations so obtained is normally very large. Hence it requires advance computational algebra to solve this system of equations. In FEM the trial functions are selected in such a way that the system is sparse.

The numerical methods based on finite difference approximation are quite efficient in solving the ordinary differential equations in one and two dimension whereas for solving partial differential equation in more than two-dimensions an advance methodology is required. The finite

elements method (FEM) is an important and efficient numerical analysis tool used in solving partial differential equations in higher dimensions. The sparse linear system that emerges from the FEM method allows FEM to compete with finite differences in terms of computational efficiency.

The finite element method was chosen as the method for thermal analysis due to its applicability for higher dimensional problems (although it is used for 2D purposes in this project), accuracy, and complexity. The understanding of finite elements and how it can be used for thermal analysis led to it being chosen over finite differences. With the understanding and knowledge, microprocessor thermal analysis for 3D chips can be an extension of this work.

IV. Finite Element Analysis

The finite element method is used to find an approximate solution to a differential equation. In FEM, the approximate solution is expressed as a sum of trial functions (ϕ).

$$U(x,y) = \sum a_i \phi_i(x,y)$$

The main steps of the FEM method are to select a convenient set of trial functions and to determine the coefficients that will obtain the closest approximation to a true solution. Galerkin's method is used for finite element analysis and involves the variational form of the differential equation. The variational form is obtained by multiplying the PDE with a test function and integrated over the domain. The equation is simplified using integration by parts. This variational equation has to be satisfied, for each trial function, by the temperature approximation. This allows the most accurate coefficients of the approximation to be determined. A system of linear equations has to be solved to obtain these coefficients. The system that results is usually very large and the computational cost can be enormous when solving this system. Therefore the system is created such that it is sparse and the final linear system of equations is sparse. This is achieved by selecting an appropriate trial function. This motivates the use of hat functions as trial functions.

The hat functions require the domain to be decomposed into smaller parts or elements. Each trial function is associated with a node in the mesh. The trial function is such that it has a value of one on that node and zero everywhere else. This results in the sparse nature of the final linear system, because the trial functions don't interact.

The overview of the steps involved in obtaining the finite element solution is shown. The first step is to write the PDE in its weak form as follows, the PDE is multiplied by test function v and integrated:

$$\int k\left(\frac{\partial \mathrm{u}}{\partial \mathrm{x}} * \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} * \frac{\partial v}{\partial y}\right)\, dxdy = -\int fv\, dxdy$$

The Galerkin method additionally introduces the idea that if the test functions are equal to the trial functions, such that the system that emerges is symmetric, sparse, and positive definite. The region is subdivided into elements for the stiffness matrix to be computed by first computing the element matrix for each element. The stiffness matrix is assembled by taking all the element matrices and assembling them with some overlap between adjacent elements. The inclusion of the boundary conditions is not shown in the flowchart. The boundary conditions come after assembling the stiffness matrix. For boundary nodes, the stiffness matrix and load vector are adjusted by putting an I in K, and the boundary value in F.

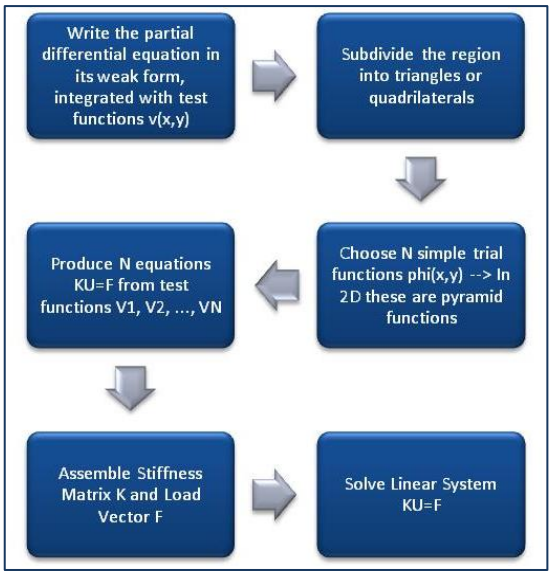

Figure 2: Flowchart of Finite Element Method [3]

A. *Numerical Implementation*

The implementation of the finite element method in MATLAB would follow the general code structure:

1. **Pre-processing:** problem structures by reading input data, finite element discretization, and material properties developed.
2. **Processing:** finite element object computed, boundary conditions enforced, and system is solved.
3. **Post-Processing:** plot the results of the finite element method, and evaluate the results

The FEM deals with integrated form of the differential equation. Thus the computation process involves the evaluation of a large number of integrals. Gaussian Quadrature schemes are good for numerically computing the integrals. A quadrature rule is an approximation of the definite integral of a function, usually stated as a weighted sum of function values at specified points within the domain of integration. Since the element chosen is triangular, a one-point quadrature is needed to evaluate the integral. To perform this gaussian quadrature the quadrature points have to be located for a triangular element. The quadrature points are obtained by finding the roots of Legendre polynomials. [5] Since the element matrices are assembled into the stiffness matrix, the integration is done on an element per element basis. Then a mapping is performed to assemble the stiffness matrix.

The preconditioned conjugate gradient present in Matlab was compared to the backslash operator to determine if the preconditioned conjugate gradient method is faster. The convergence problems encountered limited the preconditioned conjugate gradient use. The flowchart in Figure 3 shows the steps followed for the numerical implementation.

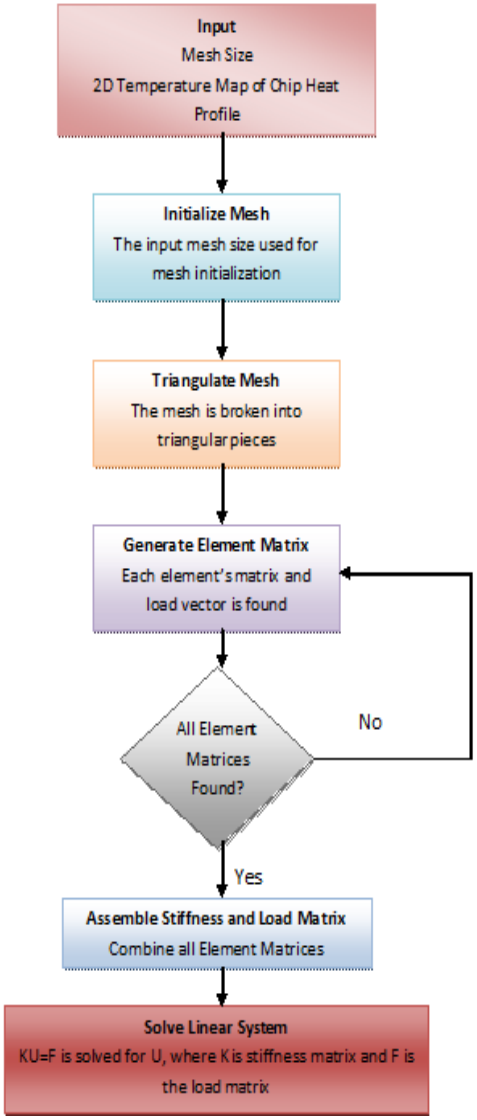

Figure 3: Flowchart showing high level implementation of finite element method for microprocessor thermal analysis

B. *Functionality Verification*

The finite element implementation was tested for functionality by setting the heat generation function to be smooth along the plane and boundaries. The comparison verified that the FEM program was indeed obtaining a close approximate solution. The maximum error of the approximation is $0.5 * 10^{-4}$. The result of the FEM analysis compared to the exact solution to test that the program is working as expected, Figure 4. The next step is to determine the convergence order of this method of thermal analysis.

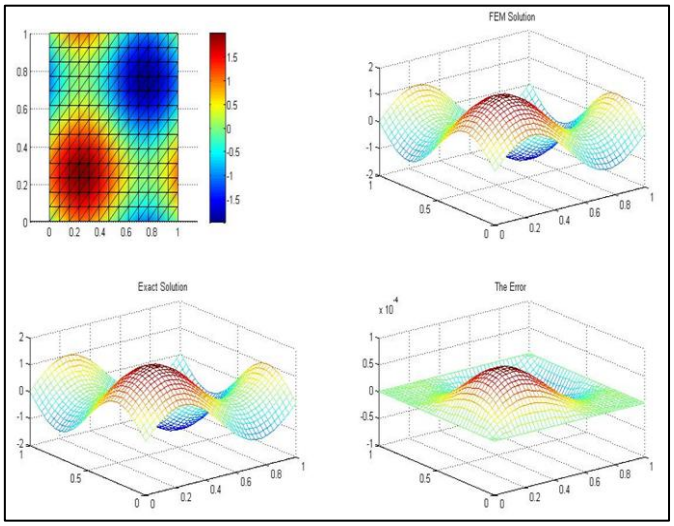

Figure 4: Functionality Verification, upper left is the plot of the top of the chip and the temperature, the error plot is on the bottom right of the graph

C. *Convergence Analysis*

Initially, prior to implementing the FEM program, it was attempted to determine the stability of the finite element method for this PDE. After some research it was concluded that the stability and accuracy of the method can be best evaluated by testing the convergence of the finite element scheme as the mesh is refined.

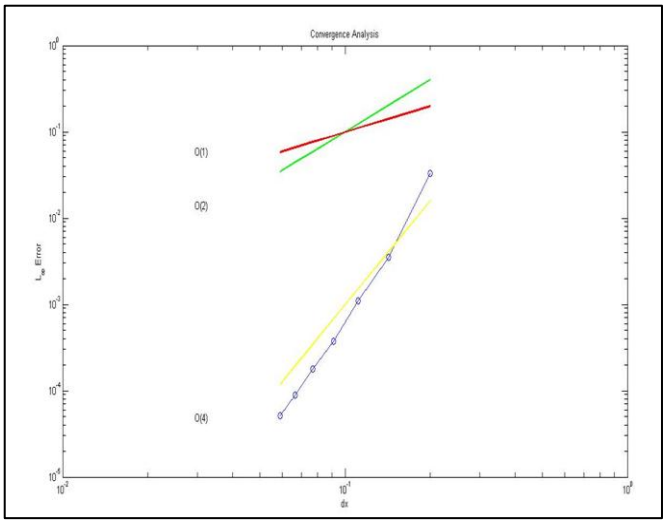

Figure 5: Convergence Analysis of FEM (4th order)

The convergence analysis plot shows the finite element method is fourth order in space because time is not considered in the steady state heat conduction equation.

V. Case Study: Alpha 21364 Processor

The Alpha 21364 chip was used as a case study for thermal analysis. The floorplan of the Alpha 21364 chip is shown below and L2 cache surrounds the floorplan on the left, right, and bottom sides:

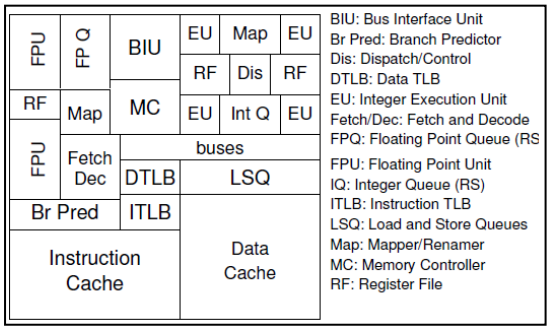

Figure 6: Floorplan of Alpha 21364 Processor [4]

The program was run by having the load function describe the temperature values in the respective locations on the chip floorplan. The temperature values of functional blocks within the chip were obtained from an experimental study performed in [4]. Internal boundary conditions were set within the floorplan such that the core and L2 cache are separated by these internal boundary conditions. The reason for this is that it is clear to see the heat generated by each functional block.

The plot of the top surface of the chip and the temperature is shown below and the mesh plot of the temperature profile is also shown.

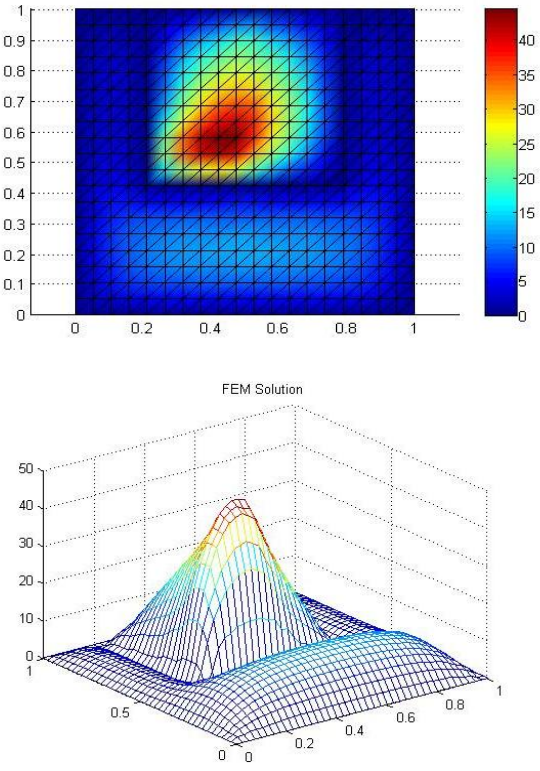

Figure 7: Temperature Profile of the Alpha Chip

VI. Multicore Floorplan Evaluation

When tiling the different cores into a multi/many core processor, the thermal profile of the chip can significantly increase. This increase in temperature can affect the speed and reliability of the processor. The optimal placement of the cores will result in the ideal thermal profile. Different floorplans for tiling the core together will be investigated, to find the one with the best thermal profile. These simple floorplan evaluations are a small step towards obtaining a thermally-aware multi-core chip platform that can be further evaluated once the detailed layout/floorplan has been established. The evaluation of on-chip thermal behavior as the floorplan varies is of interest. Two simple floorplans were compared to determine the better floorplanning option. The first approach is putting the cores on the top and bottom edges and the L2 cache is in the middle. The second approach is putting the cores and L2 caches in a checkered structure. Comparison of these two approaches reveals that the checkered approach is better in terms of the steady state heat profile. When the cores are lined up next to each other, the temperature is significantly higher. There are some instances when the one floorplan is chosen over another despite the temperature differences.

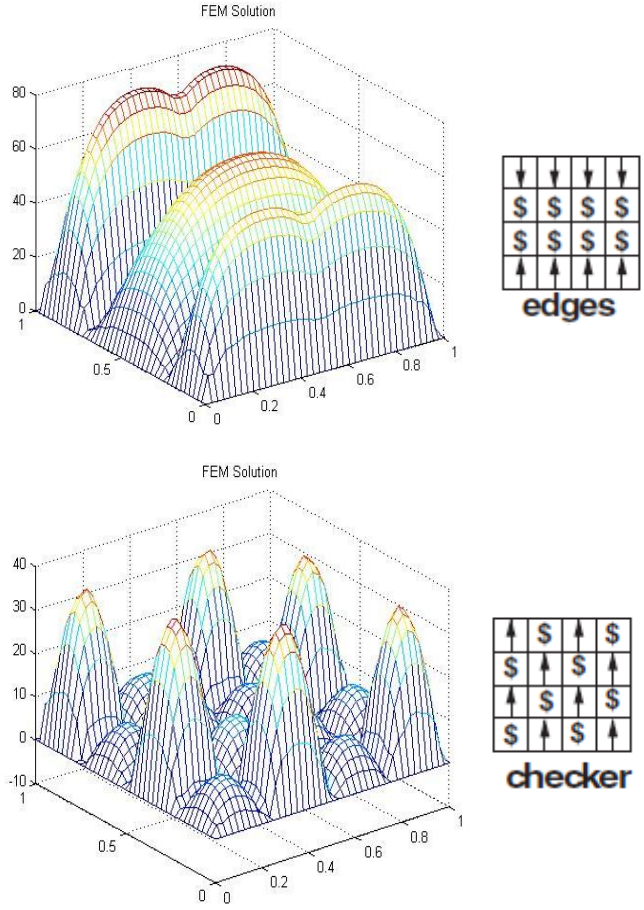

Figure 7: Comparison of multicore edges and checker floorplans

Different multicore floorplans can be compared to determine the most thermally-aware option. Although green's function methods are the most useful for the floorplanning step of the design process, when more complex core geometries and orientations need to be modeled, it may be better to use the finite element method.

VII. Future Work

The next steps are to evaluate more detailed multicore chip floorplans and to reduce the computational cost of the finite element method. A long term goal would be to extend this work to 3D microprocessors.

VIII. Conclusions

The finite element method of analysis was used to model microprocessors and evaluate multicore floorplans in the prelayout stages of development. The finite element analysis can be extended to 3D processors and is useful for accurately modeling the steady state on chip temperature. As the microprocessor industry evolves from multicore to manycore and 3D, thermal modeling will become an essential part of the design process.

References

[1] Y. Zhan, S. Sapatnekar, "High Efficiency Green Function-Based Thermal Simulation Algorithms"

[2] B. Wang, "Chip-Level Thermal Analysis, Modeling, and Optimization Using Multilayer Green's Function"

[3] G. Strang, "Computational Science and Engineering," Wellesley-Cambridge Press, 2007

[4] K. Puttaswamy, G. Loh, "Thermal Analysis of 3D Die-Stacked High Performance Microprocessor"

[5] M. Gockenbach, "Understanding and Implementing the Finite Element Method," Society for Industrial and Applied Mathematics, 2006

[6] M. Monchiero, R. Canal, A Gonzalez, "Design Space Exploration for Multicore Architectures: A Power/Performance/Thermal View"

[7] G. Xu, "Thermal Modeling of Multi-core Processors" SUN Microsystems

[8] M. Healy, H. Lee, G. Loh, S. Lim, "Thermal Optimization in Multi-Granularity Multi-Core Floorplanning"

[9] O. Bokhove, J. van der Vegt, "Introduction to (dis)continuous Galerkin finite element methods," Department of Applied Mathematics, University of Twente