POLITECNICO DI MILANO
Corso di Laurea MAGISTRALE in Ingegneria Dell'Automazione Dipartimento di Elettronica, Informazione e Bioingegneria

CONVECTION COOLING FOR MULTI-CORE PROCESSORS: DESIGN AND SIMULATION

Supervisor: Prof. Francesco Casella

Co-supervisor: Prof. Alberto Leva

A thesis submitted by:

Daniel Jose Penagos Dordevic, Matricola 872202

Academic year 2018/2019

Dediche et similia

To my family and friends who always helped and supported me in this endeavor.

To my girlfriend for its constant support, love and comprehension.

To my professors and their constant encouragement and guidance.

Abstract

Today the biggest limitation for the performance of microprocessors is given by the thermal stress produce by the high power density of the state of the art cores. The cooling solution proposed I this thesis is to inject a cooling fluid directly in the bulk (non-active) silicon of the silicon die. A model to simulate this system has been established along with a design approach that allow to choose the adequate type of fluid that is needed. Considerations on the possible hydrodynamic circuit are presented. Then an analysis of three possible fluid candidates is presented: water, mercury and gallium. It performance is then evaluated through the use of simulations. Finally, a control strategy is presented in order to finalize the feasibility study of the system.

Contents

CONVECTION COOLING FOR MULTI-CORE PROCESSORS: DESIGN AND SIMULATION ..... **I**

DEDICHE ET SIMILIA ..... **I**

ABSTRACT ..... **III**

CONTENTS ..... **V**

LIST OF FIGURES ..... **VII**

TABLE INDEX ..... **VIII**

CHAPTER 1. INTRODUCTION **1**

1.1 MOTIVATIONS AND STATE OF THE ART 1

1.2 CONTRIBUTIONS 4

1.3 THESIS ORGANIZATION 4

CHAPTER 2. THEORETICAL MODEL AND FLUID SELECTION **7**

2.1 SELECTION PROCESS 7

2.1.1 Simple theoretical model *8*

2.1.2 Decision parameters *10*

2.2 CONSIDERED FLUID FAMILIES 10

2.3 FINAL SELECTED CANDIDATES 13

2.3.1 Water *13*

2.3.2 Mercury *13*

2.3.3 Gallium *14*

2.4 FEASIBILITY CONSIDERATIONS 14

CHAPTER 3. THERMAL AND POWER DYNAMIC MODELS **16**

3.1 WHY MODELICA 16

3.2 TYPICAL LAYOUT OF A MICROPROCESSOR 17

3.3 CORE MODELS 18

3.3.1 Point model ... *19*
3.3.2 Parallelepiped model ... *20*
3.4 POWER INJECTION ... 21
3.5 FLUID MODELING ... 22
3.5.1 Liquid metals ... *25*
3.5.2 Water ... *25*
3.5.3 How the tubes are modeled ... *25*
3.6 CONTROL VALUES ... 27
3.7 MODEL VERIFICATION ... 28
3.7.1 Fourier test ... *29*
3.7.2 Exponential test ... *30*
3.7.3 Delay test ... *31*

CHAPTER 4. MODEL ANALYSIS ... 33

4.1 HOTSPOT SELECTION AND CRITICAL CASES ... 36
4.1.1 Core center ... *36*
4.1.2 Middle ... *37*
4.1.3 Sides ... *37*
4.1.4 Corners ... *37*
4.1.5 Comparison ... *37*
4.2 FINAL FLUID SELECTION ... 38
4.2.1 Gallium ... *39*
4.2.2 Water ... *39*
4.2.3 Mercury ... *39*

CHAPTER 5. CONTROL SCHEMES ... 40

5.1 WHY ON/OFF ... 40
5.2 FEEDBACK SELECTION ... 40
5.2.1 Maximum core temperature ... *41*
5.2.2 Average core temperature ... *42*
5.3 DEPENDENCE ON FLOW RATE AND WARNING ON OTHER CONTROL METHODS ... 43

CHAPTER 6. CONCLUSIONS ... 45

6.1 FUTURE RESEARCH ... 46

BIBLIOGRAPHY ... 47

List of Figures

Figure 1-1. Moore's Law graphical representation 2
Figure 3-1. Architecture layout of an Intel® Core™ i7 processor 6700K [21]. 17
Figure 3-2. Thermal model and equivalent RC circuit. 18
Figure 3-3. Electric equivalent circuit [11]. 19
Figure 3-4. Representation in Modelica of the “parallelepiped” model 21
Figure 3-5. Power Input block representation in Modelica 22
Figure 3-6. Core with convection cooling system representation in Modelica 24
Figure 3-7. Grid representation of the tubes 26
Figure 3-8 Control Block representation in Modelica 28
Figure 3-9. Coding example of a simple P controller. 28
Figure 3-10. Equivalent circuit for the Fourier test 29
Figure 3-11. Modelica representation of the Fourier test 29
Figure 3-12. Results of the exponential test 31
Figure 3-13 results of the Delay test 32
Figure 4-1. Simulation benchmark in Modelica 33
Figure 4-2. Tubes distribution 35
Figure 4-3. Core limits and distribution 35
Figure 4-4. Configurations description and names 36
Figure 4-5. Temperature comparison for critical configurations 38
Figure 4-6. Comparison among fluids 38
Figure 4-7. Zoom on the fluid comparison 39
Figure 5-1. Max temperature control result 41
Figure 5-2 Average temperature control results 42
Figure 5-3 Relation between flow and max temperature 43
Figure 5-4. Relationship between flow and pressure 43

Table index

Table 2-1. Water calculated heat flow ($\dot{Q}$), pressure differential (ΔP) and power consumption ($\dot{W}$) 11

Table 2-2. Complex organic fluids calculated heat flow ($\dot{Q}$), pressure differential (ΔP) and power consumption ($\dot{W}$) 11

Table 2-3. Heavy water calculated heat flow ($\dot{Q}$), pressure differential (ΔP) and power consumption ($\dot{W}$) 12

Table 2-4. SAE 15W 40 calculated heat flow ($\dot{Q}$), pressure differential (ΔP) and power consumption ($\dot{W}$) 12

Table 2-5. Liquid metal calculated heat flow ($\dot{Q}$), pressure differential (ΔP) and power consumption ($\dot{W}$) 12

Table 3-1. Resulting ***tubesSpec*** matrix 26

Table 4-1. Geometric constraints of the simulation 34

Chapter 1.
INTRODUCTION

This thesis is part of a bigger research activity whose objective is to design new control strategies for cooling high-power multi-core processors, integrating thermal, power and performance management. As such this thesis builds upon previous models and software previously developed by the project team.

Moore's law highlights the tendency of microprocessors to increase they performance by increasing the number of micro-transistors available per chips as time goes on. This law should always be considered since the goal is to design control systems that endure through time.

In this thesis the control strategy design is to inject liquid coolant inside the chip's bulk using small tubes that had been previously carved. The overall control approach is to add a new way of cooling a multi-core processor by controlling the flow of the fluid inside the tubes. The fluid inside the tube must be carefully selected taking into account the very different thermodynamic properties of each family of fluids and their consequences on the thermal transfer and pressure losses in the circuit. Additionally, this strategy can be used in conjunction with other cooling strategies such as DVFS (Dynamic Voltage and Frequency Scaling) or task reallocation.

Before developing this idea, Chapter 1 will serve as an introduction to topic, including a brief motivation for this work, a general description of the state of the art and the overall organization of the thesis.

1.1 Motivations and state of the art

History of the transistors can be divided thermally in three different eras [1]. Before the Intel Pentium, microprocessors had very low power densities, so low that there was no need to develop any sort of temperature control. This period corresponds to the first era. Later on,

in the second era, thermal sinks provided a more than satisfying solution for the heat dissipation problems. Currently we are on the third era, one in which power density has increased so much that new thermal solutions must be implemented. This is due to the current trend of making transistors smaller and smaller and of packing them substantially closer to each other. This is in broader terms the description of the Moore's Law; a graphical representation of the law can be found in Figure 1-1.

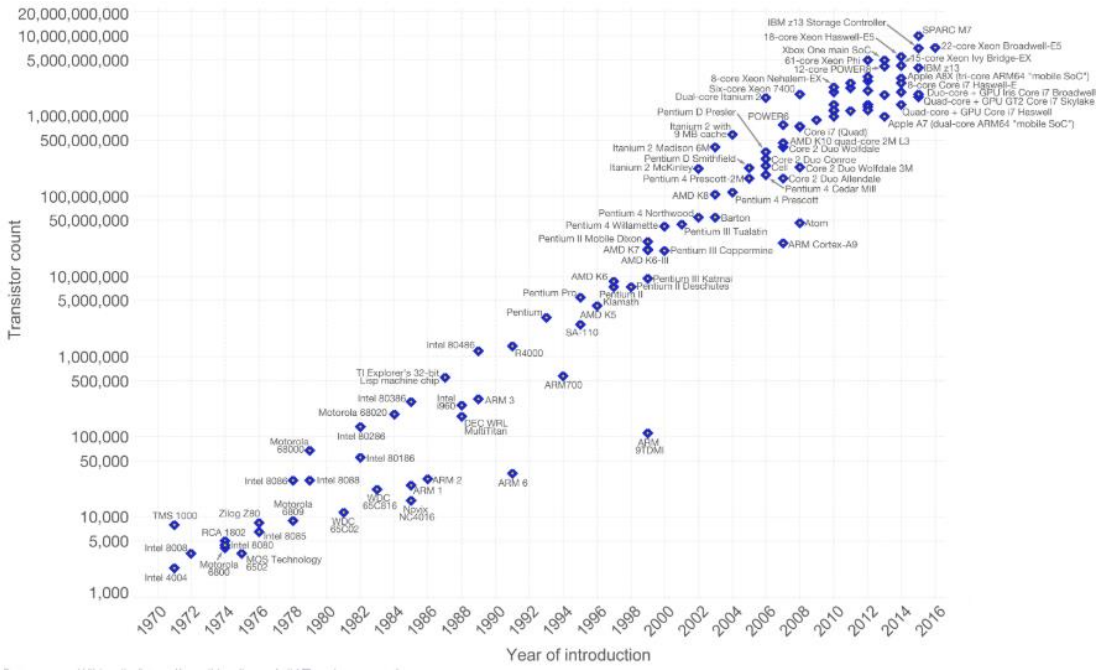

Figure 1-1. Moore's Law graphical representation

While this behavior is desired because it increases the performances of the microprocessor, it also means that temperature-related problems quickly grow in intensity.

High temperatures cause failure in the core due to thermal runaway, usually when cores operate at full power. But while this is the main problem, several other arise from the existence of high temperatures or hot spots inside the chip. Lower performance, accelerated failure rate due to thermal cycling and degradation of the microprocessor life-time are amongst the more common. These problems are created by several phenomena that depend on the temperature: hot carrier injection (HCI) [2], electro-migration [3], negative bias temperature instability (NBTI) [4], and thermal fatigue on the solder joints [5]. All of this make the temperature the key limiting factor in today microprocessor performance.

Another aspect of this problem is the complex dependencies of temperature and load distribution, this aspect prevents the effective use of predictive methods [1]. As such we must assume that the temperature readings are readily available through thermal sensors. All of this shows the need to consider thermal and power management for future microprocessors design.

Today the most commonly used technics used today are based on DVFS. They adjust the supply voltage and the operating frequency. Both actions reduce the processor power consumption, but at the same time reduce the performance of the core. More complex and advanced control techniques have been implemented based on linear quadratic regulator (LQR) [6], convex optimization [7] ,model predictive control (MPC) [8] and task migration [9].

All of this approaches have important drawbacks. The usual reaction time needed for this cooling systems should be around a few milliseconds, and all these strategies are not capable of fully reacting in this timeframe. Also they are not able to maximize the efficiency of the microprocessor, especially since all the methods have non negligible computation effort they may increase the load of the processor in a relevant way.

A novel approach tries to minimize the software consumption and load is to directly inject water through the silicon bulk [10]. The idea is to take advantage of the superior properties of water to cool down the hot spots inside the core. This also gives an additional advantage since making a fluid flow through the bulk allows to transport some of the heat through the use of the velocity of the fluid inside. This method uses convection as the main method of heat dissipation inside the core in contrast to the usual conductivity approach. This work develops this strategy further and tries to find alternative fluids to water which have better suited properties for this approach.

Another problem which has been studied is the capability to simulate this phenomenon. Currently HotSpot is the more used simulator in the literature [11] [12] [13], but it is a single-purpose and is designed to be a design tool for floor plan architecture design and not as a simulation tool for trying different control strategies. As such an alternative is to use a Modelica based object-oriented modelling library that can simulate thermal/power phenomena with more flexibility. A validation of the use of Modelica for this application can be found at [12]. On the other side Modelica can also model thermo-hydraulic objects like the tubes inside the silicon bulk. Successful simulation of this kind of application can be found at [14] using the Modelica library *ThermoPower*.

This problem is increased by the inability to properly understand how temperature actually evolves inside a particular chip, because floor plans and detailed information of the architecture of the chips are protected by patents and copyright laws. This is understandable since, microprocessor companies want to keep these factors a secret, since they represent a competitive advantage against their competition. Some studies like [15] shed some light on the matter, using thermal imagery they were able to determine that hot spots tend to appear in areas of around 1 mm^2. Using this we can estimate how thermal phenomenon evolve inside the silicon bulk.

1.2 Contributions

The first contribution of this work is to carry out a theoretical estimation of the feasibility of the whole convection control strategy. This control process will be evaluated on reduced complexity theoretical models, in which key parameters will be used to evaluate the success or failure of the strategy. Since this model uses convection to cool down the chip, a key component of the feasibility analysis is to select a series of fluid candidates which can be used in the operation and exhibit thermodynamic behaviors which facilitate the removal of heat on the chip. A wide variety of fluid families will be evaluated and the three must ideal candidates will be evaluated in section 4.2.

The second contribution of this work is to develop a library of components which could be used in conjunction with the already exiting Modelica library for thermal/power/performance control of multi-core microprocessors [11]. In particular, this adds the ability to model the tubes inside the chip and some of the control modules that can be used with this new control strategy. These components are designed to be flexible enough to adapt to future ideas such as new liquids, new control modules or different tubes configurations. A test bench was implemented to ensure the accuracy and validity of the models implanted. These tests are carried out in idealized situations where the analytical solution is known, to verify the correct implementation of the components.

The third contribution of this thesis is to generate a simulation that validates the control strategy and use this result to select the best fluid and composition of the tubes. This information will provide the best approximation that can be achieved without manufacturing a prototype.

1.3 Thesis organization

The thesis is structured as follows. Chapter 2 explains the process used to estimate the possible fluids to use in the next stages of the project. A compact theoretical model will be developed for this purpose and different fluids will be tested in this environment to select the ones with better performance.

Chapter 3 elaborates on the different models and components of the Modelica library and explains the way in which the components allow to create a complete simulation. Examples of the components are shown and their parameters are explained.

Chapter 4 shows the models used in the simulation phase and the results that provide the framework for any control strategy implemented. The final comparison between the three fluid candidates is done in this chapter.

Chapter 5 shows the implemented control models and the respective degrees of freedom that can be used to elaborate on other possible control models.

Finally, chapter 6 presents the conclusions of the thesis and proposes future work and improvements.

Chapter 2.
THEORETICAL MODEL AND FLUID SELECTION

Since the main cooling mechanism in the proposed model is convection, a fundamental factor in the success of the model is the selection of the fluid. The properties of the selected fluid influence the rate at which heat can be dissipated and even if the fluid is capable of absorbing the heat.

A lot of fluids can be considered to flow through the tubes inside the chip and as such a model must be developed to select the fluid which better suits our needs. This chapter will explain in detail the process and criteria to select the appropriate fluids.

Three candidates will be selected to proceed to the simulation phase as possible candidates for the ideal fluid for this application.

2.1 Selection process

Building a model which takes into account all the effects and parameters of the cooling of a multi-core microprocessor is extremely taxing and lacks of a simple analytical solution. As such the first approach to this new control strategy is to use a known theoretical approach with similar characteristic to the chip thermal problem. The selected model is the tube exchanging heat with an external fluid at a fixed temperature.

The main difference between this model and our final model is that the external temperature is not fixed through the full length of the tube. This allows us to use this known scenario to get qualitative information on the performance of the fluids considered. This is due to the fact that if a fluid performs better than others in this simplified scenario they can be expected to also perform better when used inside the microprocessor.

Since the amount of fluids to consider is substantial, this approach reduces the use of a lot of computationally intensive and time-consuming simulations.

2.1.1 Simple theoretical model

For a simple tube exchanging heat with an external fluid at a fixed temperature if we consider the temperature difference between the external temperature and the fluid temperature to be constant (in our numerical solution of the model we will always consider a ΔT of 10 K), the formula to calculate the heat dissipated or absorbed through the walls is:

$$Q = \gamma \cdot A \cdot \Delta T, \qquad \text{(Eq. 2.1-1)}$$

where Q is the heat flow through the walls of the tube, γ is the coefficient of heat transfer, A is the area in which the heat exchange occurs and ΔT is the difference between external temperature and fluid temperature. Since we are dealing with a square tube, the area is calculated as in (Eq. 2.1-2)

$$A = 4 \cdot l \cdot L, \qquad \text{(Eq. 2.1-2)}$$

where l is the length of a side of the cross sectional area of the square tube and L is the length of the tube. The analysis of the thermodynamic convection problems has been studied through the use of dimensionless numbers, which let us determine the effects of the convection phenomena through the relationships between this numbers [16]. In the case of convective heat transfer, the relevant numbers are: the Reynolds (Eq. 2.1-3), Nusselt (Eq. 2.1-4) and Prandtl (Eq. 2.1-5) numbers.

$$Re = \frac{\rho \cdot u \cdot D}{\mu}, \qquad \text{(Eq. 2.1-3)}$$

$$Nu = \frac{\gamma \cdot D}{\lambda}, \qquad \text{(Eq. 2.1-4)}$$

$$Pr = \frac{\mu \cdot C_p}{\lambda}, \qquad \text{(Eq. 2.1-5)}$$

where ρ is the density of the fluid, u is the mean velocity of the fluid, D is the hydraulic diameter of the tube, λ is the heat conductivity of the fluid and C_p is the specific heat capacity of the fluid. It is important to note that since we are dealing with square tubes the hydraulic diameter must be calculated as in (Eq. 2.1-6)

$$D = \frac{4A}{P} = \frac{4 \cdot l^2}{4 \cdot l} = l \qquad \text{(Eq. 2.1-6)}$$

This means that the hydraulic diameter is equal to the length of one side of the cross sectional area of the square tube for all the relationships of the dimensionless numbers of

model. This relationship between the dimensionless numbers heavily depends on the type of fluid and the conditions in which it is flowing. The main equation used for this relationship is the Dittus-Boelter equation shown in (Eq. 2.1-7).

$$Nu = 0.023 \cdot Re^{0.8} \cdot Pr^{0.4} \quad \text{(Eq. 2.1-7)}$$

Note that this equation is only applicable if the conditions expressed in equation (Eq. 2.1-8) are true.

$$Re \geq 10\,000; \quad \frac{L}{D} \geq 10; \quad 10 \leq Pr \leq 160 \quad \text{(Eq. 2.1-8)}$$

A different case which is very important for our project is the case of the liquid metals. In this kind of fluid, the heat conductivity is very high and as such conduction takes a great part in the thermodynamic process. The equation used in this cases is equation (Eq. 2.1-9) [16]

$$Nu = 4.82 + 0.0185(Re \cdot Pr)^{0.827} \quad \text{(Eq. 2.1-9)}$$

As with the Dittus-Boelter equation this equation is only applicable if the conditions expressed in equation (Eq. 2.1-10) are true.

$$3\,600 \leq Re \leq 905\,000; \;\; 0.003 \leq Pr \leq 0.05; \quad 100 \leq Pr \cdot Re \leq 100\,000 \quad \text{(Eq. 2.1-10)}$$

Finally, as it will be discussed later, we need to calculate the pressure differential needed to maintain the flow rate and the power a pump needs to provide this differential. Pressure is calculated using equation (Eq. 2.1-11) and power is calculated using equation

$$\Delta P = \frac{u^2 \cdot c_f \cdot \rho \cdot 4 \cdot D \cdot L}{2 \cdot A}, \quad \text{(Eq. 2.1-11)}$$

$$W_{pump} = \frac{\Delta P \cdot u \cdot A}{\eta}, \quad \text{(Eq. 2.1-12)}$$

where c_f is the Fanning's friction coefficient, ΔP is the pressure differential, W_{pump} the power needed by the pump and η is the efficiency of the pump. Since our purpose is to do comparative analysis η will always be the same value for each fluid (80% efficiency). Also since the expected smoothness of the tube is not expected to be very good we will always assume a large value of c_f (0.03) which will guarantee that we consider the worst possible case.

In steady-state this model behaves in space following equation (Eq. 2.1-13) In this equation the term $\frac{\gamma \cdot A_{tot}}{\omega \cdot C_p}$ is known as the dimensionless number NTU (Number of Transfer Units).

$$T(x) = (T - Tamb) \cdot e^{-\frac{x}{L}\left(\frac{\gamma \cdot A_{tot}}{\omega \cdot C_p}\right)} + T_{amb} \quad \text{(Eq. 2.1-13)}$$

NTU lets you know if the exchange is equivalent to a short or long exchangers. In short exchangers behaves almost linearly and in the long exchangers it exponential. An exchanger is considering short if NTU is less than one and long otherwise.

As you can see all of the equations are time independent and this is due to the fact that we are only interested on the steady-state during the design process and the dynamic effects will be considered in the simulation part.

2.1.2 Decision parameters

The first selection criterion for the fluids is that they should be in liquid state in the expected temperature range of 20 to 100 °C. It is very important to avoid going into a vapour state since liquids work better as conductors of heat and are easier to handle.

Subsequently, the basic parameters that must be calculated to evaluate the performance of a fluid are heat dissipation on one hand, pressure and work of the hypothetical pump that will power the flow of the fluid on the other hand. Since we are mainly interested in heat dissipation, the heat produced in the microprocessor is the main constraint and we know that an Intel microprocessor consumes 100W which can be transformed into heat [17]. To cool efficiently, the fluid must be in contact with the largest possible area; this means that the chip it must have around 10 tubes of an expected diameter of 0.5 mm. With 10 tubes, each one must dissipate at least 10 W of heat. This is the more restrictive condition since if a fluid is not capable to take at least this amount of heat the fluid can't be used for this application.

Pressure is also critical because the critical application for a microprocessor is working in a data center. This kind of application cannot have big pressure differentials to avoid dealing with possible ruptures or leaks. We decided to allow a maximum value of 3 bars or 300 kPa. As it can be seen from the equations pressure is directly proportional to the square of the velocity. As such we need to minimize this, while at the same time ensuring turbulent flow. This is done in the equations by using the lowest allowable Reynolds number.

Finally, since the idea is to reduce the impact on the power consumption we will select the fluids that provide the lowest power consumption.

2.2 Considered fluid families

As it has been shown before, the model requires that the values of thermal conductivity, density, viscosity and specific heat are known and as such we must use very well-known materials whose thermodynamic properties can be obtained through known formula or tables. The Modelica Standard Library has a thermal library for over 100 known fluids. Other very popular fluids whose data is readily available from different sources are thermal, SAE and

engine oils. Finally, liquid metals (metals that are in their liquid form at temperatures close to room temperature) were included in this analysis due to their astonishing thermal conductivity which plays a huge role in the convection process.

The first fluid that must be evaluated in any cooling system is water mainly due to its low cost and high availability. The values calculated for water are then taken as the benchmark values for all the other fluids. This values are reported on Table 2-1. On the table the results of a single tube are presented and in regards to our analysis to be approve they must be greater than 10W. Also three different initial temperatures of the fluid are show (50, 70 and 90) to have a grasp of how the fluid behaves at different temperatures.

Table 2-1. Water calculated heat flow ($\dot{Q}$), pressure differential (ΔP) and power consumption ($\dot{W}$)

Fluids	T [C°]	ΔP [atm]	Ẇ [W]	Q̇ [W]
Water	50	2.04	0.72	12.80
	70	1.12	0.29	14.06
	90	0.69	0.14	12.89

After eliminating the fluids that boil under 100°C in the Modelica library, such as simple organic fluids (for example pentane), the next family of fluid to evaluate is the complex organic fluid, specifically from heptane to dodecane. The calculated value from heptane and dodecane are reported on Table 2-2. This values show that these fluids are not capable of dissipating the needed amount of heat.

Table 2-2. Complex organic fluids calculated heat flow ($\dot{Q}$), pressure differential (ΔP) and power consumption ($\dot{W}$)

Fluids	T [C°]	ΔP [atm]	Ẇ [W]	Q̇ [W]
Heptane	50	0.91	0.26	3.63
	70	0.64	0.16	3.35
	90	0.47	0.10	3.08
Dodecane	50	7.76	6.20	5.75
	70	4.71	2.96	5.15
	90	3.06	1.57	4.67

The other fluid that does not boil at 100°C in the thermal library is heavy water. It values are reported on Table 2-3. As it can be seen it fulfills all the constraints and compared to water it has a slightly better heat dissipation but a higher power consumption.

Table 2-3. Heavy water calculated heat flow (Q̇), pressure differential (ΔP) and power consumption (Ẇ)

Fluids	T [C°]	ΔP [atm]	Ẇ [W]	Q̇ [W]
Heavy water	50	2.60	0.98	16.38
	70	1.41	0.39	14.57
	90	0.85	0.19	13.19

Following heavy water, oils were considered. Very different oils family where considered such as SAE oils, thermal oils and engine oil. All of them, due to their high viscosity (over ten times the viscosity of water), require a huge velocity to reach turbulent flow and as such a really high pressure differential is needed. Table 2-4 shows the calculated values for the oil SAE 15W 40, and it clearly shows the excessive amount of pressure needed to bring the fluid to turbulent flow [18].

Table 2-4. SAE 15W 40 calculated heat flow (Q̇), pressure differential (ΔP) and power consumption (Ẇ)

Fluids	T [C°]	ΔP [atm]	Ẇ [W]	Q̇ [W]
SAE 15W 40	50	25507	1073846	29.95
	70	11390	321556	25.67
	90	5590	110934	22.40

The final family of fluid considered are the liquid metals among which only mercury and gallium are in liquid form for most of our temperature range. Due to their high thermal conductivity, they can dissipate heat better than water and thanks to their high density they require relatively low velocity to reach turbulent flow. This can clearly be seen for both Gallium and Mercury at Table 2-5.

Table 2-5. Liquid metal calculated heat flow (Q̇), pressure differential (ΔP) and power consumption (Ẇ)

Fluids	T [C°]	ΔP [atm]	Ẇ [W]	Q̇ [W]
Gallium	50	1.80	0.178	93.58
	70	1.56	0.166	92.74
	90	1.40	0.157	92.06
Mercury	50	1.47	0.015	22.67
	70	1.25	0.013	22.23
	90	1.09	0.012	21.89

2.3 Final selected candidates

Section 2.2 highlighted the four fluids that are usable while it showed a short description on other fluid families that are not usable in our application. As explained on Section 2.1.2 the fluids were sorted by power consumption leaving heavy water at last place. In addition to this heavy water is an expensive fluid this two factors drop this fluid from the following analysis and simulations.

The following sections show the considerations that must be taken for the use of each of the remaining fluids: water, mercury and gallium.

2.3.1 Water

Water as it has been stated before has the very distinct advantage of been a chap very common fluid. This coupled with the fact that it fulfills all the constraints of the model make it a very strong candidate for this application.

However, water does have a small problem in particular with the boiling temperature. Water boils at 100°C and at atmospheric pressure, and multi-core processors allow a working temperature at exactly 100°C and at pressures very close to 1 bar (the circuit could be pressurized at higher values of pressure, though high pressures would make this impractical and possibly dangerous). This means using water may generate the presence of steam inside the tube, which eventually will result in leakages and pressurization inside the tubes, both of them should be avoided especially considering that chips usually work at data centers.

2.3.2 Mercury

Mercury is not a gentle fluid to be working with especially considering its high toxicity and the magnetic effects it can have on circuitry. These problems though van be easily solved by insulating and sealing the tubes so that no leakage is permitted and to avoid the electromagnetic iteration with the active layer of the chip.

On the other hand, mercury is an amazing fluid for this application. It is the fluid with the least amount of pressure needed and with a virtually zero power consumption (less than 0.02% of the chips electrical consumption). Initially its high density may seem like a problem to pump it through the chip but it is in fact this density which allows to pump it at very small velocities.

2.3.3 Gallium

Due to its high conductivity gallium is able to absorb almost all the maximum heat expected from the chip in a single tube. With less pressure requirement and smaller power consumption than water. It is a very good option for this application.

Its main problem is that under 30°C it returns to its solid state. This means that some heat must be generated to melt the gallium in order to pump it inside the chip. While this can be done outside the chip it will still increase the power consumption substantially.

This also means that our cooling fluid will never reach temperatures of under 30°C which is a fairly high value compared to the room temperature (20°C) or lower that can be reached with water and mercury. Furthermore, solidifying processes or solid parts going through the hydrodynamic circuit may block tubes or break small components such as valves or pumps needed for this application.

2.4 Feasibility considerations

With some fluids selected and with a fairly clear operating point, for each of the fluids, some discussion must be done in regards to the feasibility of pumping these fluids inside the chip. Since we are dealing with dimensions in the orders of 0.5 mm we must consider using MEMS technologies either for the pumps or from the valves.

MEMS micro pumps have gone a long way since their original designs. Today they are able to provide the required pressures (some can provide up to 270 kPa). Today the pumps that can provide this amount of pressure don't provide the required flowrate, this could be very easily solved by simple adding more pumps in parallel. The problem with this approach is that space is very limited in a motherboard and adding several MEMS micro pumps per chip decreases the available space for other important components, since current micro pumps are more than 0.5 mm long [19].

A MEMS micro valve though is already available in the market which offers substantially better specifications that the ones required by the theoretical model (the expected flowrate is around 36 to 166 cc/min and some MEMS valves can provide up to 5000 cc/min) [20]. It is very important to note that since we want to avoid leaks the system that couples the valves to the tubes in the chip must be designed very carefully, but this design is outside of the scope of this thesis.

Other important considerations of feasibility are regarding the return fluid circuit. We have only considered pressure and power of the tube inside the chip. However, the fluid needs to be recirculated through a return pipe. Since we can design the geometry of this pipe much more

freely than the tubes inside the chip, we could modify the cross sectional area to reduce the speed to a level in which the return flow contribution to pressure and power are insignificant.

Chapter 3.
THERMAL AND POWER DYNAMIC MODELS

This chapters describes the used object-oriented model that defines the microprocessor thermal and power behavior. The idea is to build upon the previously developed blocks [11] creating new ones that can be used with the existing library. Due to the nature of this control mechanisms we need to use the three-dimensional description of the problem.

Old and new block needed for a successful simulation of the phenomenon are described in this chapter along with instructions to use each block. First a brief description is shown of how the multi core processor is modeled and simulated in Modelica using existing blocks. Then the modifications done to the model to implement the convection cooling control system are explained. And finally, validations of the model with theoretical values are shown to prove the validity of the used blocks.

3.1 Why Modelica

The current state of the art simulator used in the literature is HotSpot [11] [12] [13] but its functionalities are designed mainly for design purposes. It is not flexible enough to analyze different control strategies let alone to model the physical changes done to the chip core to apply the convection cooling strategy.

The Modelica context is very useful for object oriented approaches to models. It allows to create complete and well organized libraries and to create different blocks that can be connected among them to create complex structures and models. The main advantage is that, through the use of a causal physical connector, multi-physical modeling can be achieved, with each block having its own set of equations and/or algorithms. This is especially useful in the

convection model since some part of the chip will be composed of fluid blocks which represent the tubes and other from silicon block which will represent the bulk of the chip.

Furthermore, previous studies have built a Modelica library that has been put to test next to HotSpot, using identical power traces and chip floorplan. By comparing the results of this comparison the Modelica library was validated [12]. The latest version of this is library is called ThermalModel and was a product of the group research [11]. This library was created to implement cooling control strategies such as DVFS but due to its modular design they can be used with the convection cooling strategy. In particular, for this thesis the relevant blocks used are the three dimensional power core and the power injection block.

3.2 Typical layout of a microprocessor

Before explaining the used blocks, a small introduction to microprocessor is needed specially about the material and physical properties of the chip. A microprocessor consists of a silicon die that host the cores and its support structure. The cores are volumes of active silicon that generate heat/thermal power which depend on the processor load and the clock frequency. Both of this parameters are highly time varying and depending of the architecture but this heat generation dependence is applicable to single core and quad-core structures.

Figure 3-1.Figure 3-1 shows the architecture of an Intel® Core™ i7 processor 6700K [21]. This correspond to the typical structure of quad core chips which will be used in this work. As it can be seen from the picture the cores are aligned in a 2x2 table. This configuration will be used on the simulations and Modelica models.

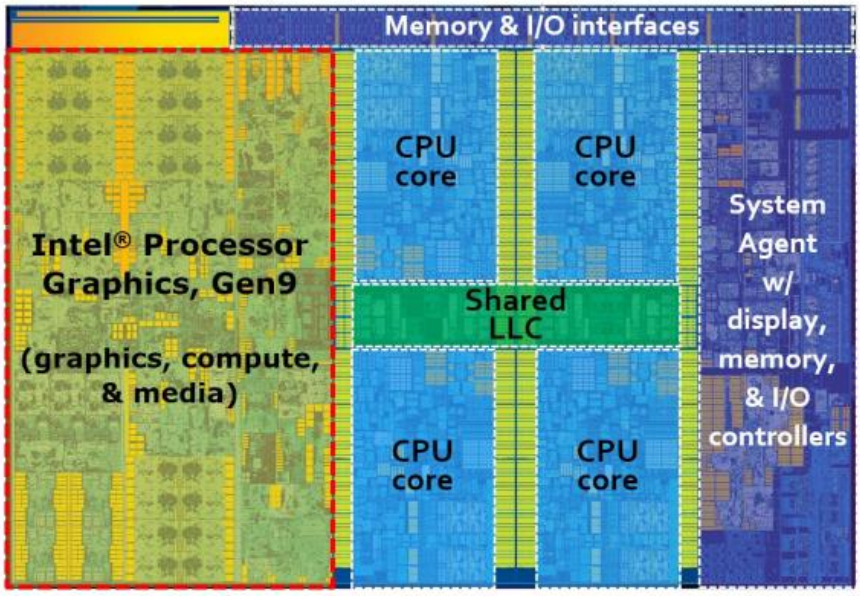

Figure 3-1. Architecture layout of an Intel® Core™ i7 processor 6700K [21].

3.3 Core models

The core is the fundamental element of the thermal analysis since it is the layer were heat is generated and is the part of the model which reaches the higher temperatures.

The thermal and power load starts at this active silicon cores and it is exchanged through the chip-wide silicon bulk. After this the heat is exchanged to a heat spreader (usually metallic) with a heat sink (e.g. a fan) that dissipates the heat to the environment. Usually this exchange is modelled as a RC Network [13] [1]. Due to the material properties (thermal conductivity, volume, etc.) the capacitance of the active silicon is smaller than the capacitance of the bulk which is in turn smaller than the spreader and sink capacitance. Figure 3-2 shows the general model of the whole system.

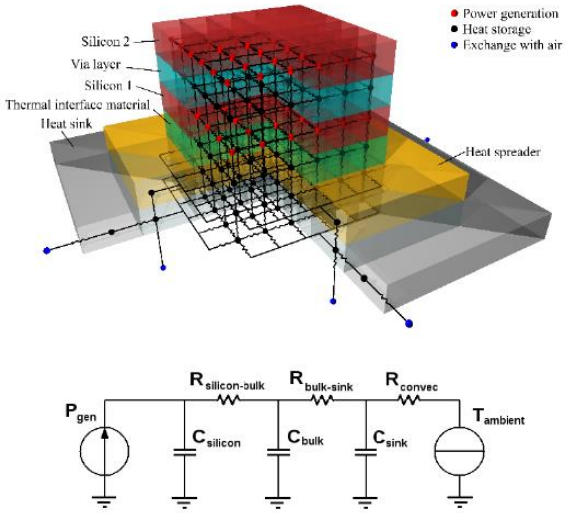

Figure 3-2. Thermal model and equivalent RC circuit.

Since the capacitances are quite different, the whole process works in three different time scales. In this way the system is thermally decoupled in bandwidth. With active silicon effects having the faster dynamical effect and the spreader the slowest one. This means that the first effects to take place are the ones due to the silicon heating, which happen in the first milliseconds of operation, followed by the effects of the bulk which start to be evident arround the 10 milliseconds, and finally after more than 100 milliseconds the effects of the spreader capacitance take dominance of the process.

This model is useful for determining general behavior on usual microprocessor application. However, due to the existence of the tubes inside the bulk we need to develop a finer model.

3.3.1 Point model

To achieve a finer model a finite volume (FV) approach is needed. For the bulk, with uniform density, specific heat and conductivity, a layer of material can be thermally represented by an electrical equivalent circuit [12] [11] [1]. Figure 3-3 shows the equivalent circuit of a single FV. The sub-indexes top, bottom, north, southeast and west where chosen to give a direction and easily identify the boundaries of the core.

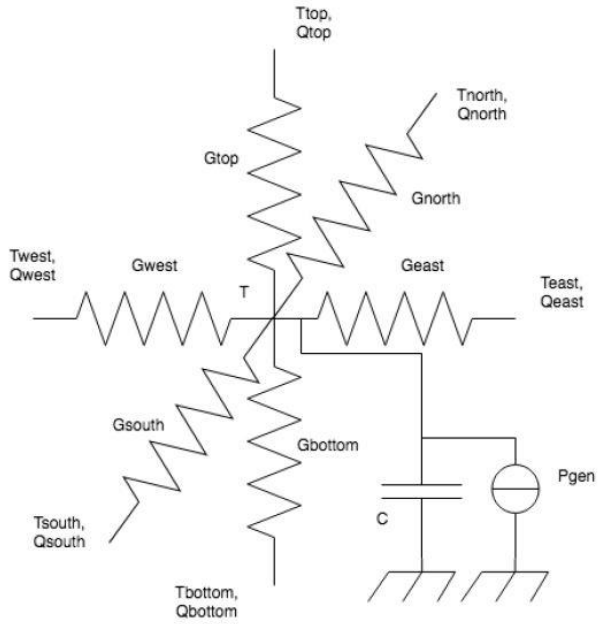

Figure 3-3. Electric equivalent circuit [11].

In this circuit analogy temperature (T) is treated as voltage and the heat flow (Q) as a current. The source ***Pgen*** injects a value of power into the circuit which represents the generated power (and heat) of the active silicon core. The capacitance c models the dynamic effects of the point model. As the model shows, the values of T are dependent on the conductance and capacitance values.

For the sake of simplicity, the "point" model will be explained considering the whole silicon core as a single point model. In this case the values of the capacitance and conductance are dependent only on the silicon properties and the dimensions of the microprocessor core. The equation for capacitance is shown in (Eq. 3.3-1).

$$c = \alpha \cdot C_p \cdot \rho \cdot V \qquad \text{(Eq. 3.3-1)}$$

Where α is a fitting factor for lumped/distributed compensation, C_p is the specific heat of the silicon, ρ is the density of the silicon and V is the volume. The equation for conductance is shown in (Eq. 3.3-2).

$$g_i = \frac{\lambda \cdot A_i}{t_i} \qquad \text{(Eq. 3.3-2)}$$

Where i is the port index (top, bottom, north, south, east, west), λ is the thermal conductivity, A is the cross-sectional area and t is the thickness.

Since this is a simple electrical circuit model, Q can be calculated with equation

$$Q_i = 2g_i(T_i - T) \qquad \text{(Eq. 3.3-3)}$$

The microprocessor as a single point has the following energy balance equation (Eq. 3.3-4).

$$c \cdot \dot{T} = Q_{bottom} + Q_{top} + Q_{north} + Q_{south} + Q_{east} + Q_{west} + P_{gen} \qquad \text{(Eq. 3.3-4)}$$

But since we don't have just one single finite volume, we need to have several point models in line to achieve a fine enough model. This is why a different three-dimensional model is required. This model is known as the "parallelepiped" model.

3.3.2 Parallelepiped model

A first approach may be to simply take the point model and create a 3D model in Modelica. This approach has several problems, in particular a grid built this way is not easy to modify since it grows in complexity very fast the smaller the discretization needs to be. An alternative approach is to build a model whose parameters are the width, length and thickness of the microprocessor and the discretization is determined by the amount of rows, columns and layers. This creates a model in which temperature is stored in a matrix of dimension rows by columns by layers. This means that the temperature of each FV can be taken by taking the value of $T[i, j, k]$.

To adjust to this new model the differential equation (Eq. 3.3-4) must be applied to each individual cube created by the discretization. Resulting in a model of rows-by-columns-by-layers equations to be solved. For example, a model of 10 x 10 x 10 model creates a 1000 equations. The value of the capacitance of each of this equations must be recalculated using equation (Eq. 3.3-5). This adjust the value of the capacitance of the whole bulk to the capacitance of each individual FV by using the volume of each individual FV.

$$c = \frac{\alpha \cdot C_p \cdot \rho \cdot V}{rows \cdot columns \cdot layers} \qquad \text{(Eq. 3.3-5)}$$

This approach can be also applied accordingly to the conductance of the model. By using the appropriate cross-sectional area and thickness of the resulting FV cubes. However, to calculate the heat flow some precaution must be taken. The equations change depending on if the flow is between block or if they are done between a block and a port. Since when the exchange occurs between two blocks the equivalent circuit is a series of two resistances the heat flow is equal to (Eq. 3.3-6).

$$Q_i = g_i(T_i - T_{i\pm1}) \qquad \text{(Eq. 3.3-6)}$$

In this equation i is the dimension in which the heat flow is going to be calculated. For the cases in which the exchange is done to the exterior (through a port) the equation used for the heat exchange is the same as in the "point" model (Eq. 3.3-3). Also average, maximum and minimum temperature are calculated to get insight of the evolution of the thermal behaviour of the silicon chip as a whole and not at each individual FV.

Additional parameters (such as density, specific heat and thermal conductivity) can be changed in the block to model not only the core, but any other rectangular component of the model, of particular interest for this work is the spreader. This existing block is presented in Figure 3-4.

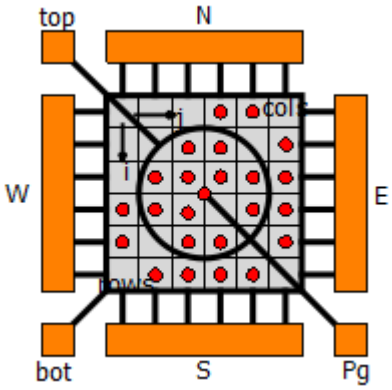

Figure 3-4. Representation in Modelica of the "parallelepiped" model

Keep in mind that this ports are two-dimensional ports since each of them represents a face with all of their FV exchanging heat through that port. A small exception is the ***Pg*** port since, as it will be discussed in the next section, it is a three-dimensional port.

3.4 Power injection

As mentioned in [1] workload prediction are an unreliable estimation for power consumption, because malicious software can use generate unexpected power consumptions that will not be predicted provoking a performance degradation in the temperature control system. Also load or frequency are not used as an estimation of power since the phenomenon

is extremely fast and using these estimations will not heat the core fast enough. The best way is to inject directly the values of power. This is why in our electrical model the heat is generated using a current generator.

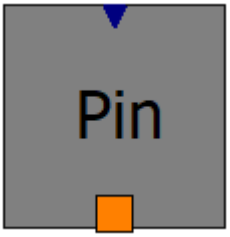

Figure 3-5. Power Input block representation in Modelica

Figure 3-5 shows the existing block used to inject power into the "parallelepiped" model. This block takes a vector with the power profiles and produces an output with a three-dimensional matrix with the generated power in each of the FV cubes. It uses a simple algorithm in which for each profile the power is distributed uniformly on the expected volume in which it is injected. These volumes are retrieved from the parameters. In case the specific FV is not selected the power generated value ***Pgen*** is equal to zero for that particular cube.

3.5 Fluid modeling

Until this point previously developed blocks were used. The next step is to use the knowledge provided by the previous blocks to create one that is capable to model the fluid canals inside the chip. This is done by creating two types of FV block: one corresponding to the silicon parts of the chip and the other for the parts in which the fluid flows.

Both of the types are modeled as a "point" models. In particular, we are using the electrical equivalent circuit. The main differences between the two types are the material parameters and the equation used to calculate the resistances of each FV cube. It is important to point out that in this model we use resistances since, as it has been pointed out before, when the exchange is done between inner blocks the resistances (or conductance) for the heat transfer are in series. It is easier to work with resistances in this case because they are simply added.

This means that the equation for heat transfer needs to change to (Eq. 3.5-1)

$$Q_i = \frac{T_i - T_{i\pm1}}{R_i + R_{i\pm1}} \qquad \text{(Eq. 3.5-1)}$$

For the silicon parts the values of the resistance are calculated simply by using (Eq. 3.5-2)

$$R_i = \frac{1}{g_i} \qquad \text{(Eq. 3.5-2)}$$

Particular attention is required in the equations for the fluid since the resistances are not equal in all directions due to the existence of a flow. Since the flow of the fluid is one directional the block is made considering a flow from north to south. This direction was arbitrarily selected but it can be changed to other directions only by connecting the other blocks appropriately.

The model used to determine these resistances is very similar to the one used in the design phase. The differential equation that models this behavior is

$$c \cdot \dot{T_o} = \omega \cdot C_p \cdot (T_i - T_o) + \dot{Q_{cond}}, \qquad \text{(Eq. 3.5-3)}$$

where T_i is the temperature at the inlet of the tube, $T_o$is the temperature at the outlet of the tube and Q_{cond} is the heat dissipated thanks to conduction. Applying this equation to our FV blocks we can express $T_{i_{i,j,k}}$ to be $T_{o_{i,j,k}}$. By renaming T_o for simplicity as $\hat{T}$ (Eq. 3.5-3) can be rewritten as

$$c \cdot \dot{\hat{T}_{[i,j,h]}} = \omega \cdot C_p \cdot (\hat{T}_{[i,j,h]} - \hat{T}_{[i-1,j,h]}) + \dot{Q_{cond}} \qquad \text{(Eq. 3.5-4)}$$

This particular solution needs that a correction is applied to the temperature matrix inside the flow blocks considering the temperature at the outlets $\hat{T}$. This is shown in (Eq. 3.5-5).

$$T_{[i,j,k]} = \frac{\hat{T}_{[i,j,h]} + \hat{T}_{[i-1,j,h]}}{2} \qquad \text{(Eq. 3.5-5)}$$

In our particular case since each direction exchanges temperature is different Q_{cond} cannot be summarized in a single expression but must be expressed as (Eq. 3.5-6). It should be noted that in this equation the temperature used is not $\hat{T}$ since it must take into account the temperature at the middle of the block..

$$\begin{aligned} \dot{Q_{cond}} = {} & \gamma A_{top}\left(T_{[i,j,h+1]} - T_{[i,j,h]}\right) + \gamma A_{bottom}\left(T_{[i,j,h-1]} - T_{[i,j,h]}\right) \\ & + \gamma A_{east}\left(T_{[i,j+1,h]} - T_{[i,j,h]}\right) + \gamma A_{west}\left(T_{[i,j-1,h]} - T_{[i,j,h]}\right) \end{aligned} \qquad \text{(Eq. 3.5-6)}$$

We now have a reduce version of the design model taking into account the existence of different temperatures between the top, bottom west and east directions. This means that for this directions the value of the resistance is (Eq. 3.5-7).

$$R_i = \frac{1}{\gamma \cdot A_i} \qquad \text{(Eq. 3.5-7)}$$

Where γ is the coefficient of heat transfer and A_i is the area over which the exchange is happening. The remaining needed resistance are north and south and due to them being the resistances in the direction of the flow need to reflect the amount of heat that is transported

thanks to the flow of the fluid. By expanding the term of the equation (Eq. 3.5-4) responsible of the flow we find that the heat in each direction is calculated as in (Eq. 3.5-8) and (Eq. 3.5-9)

$$Q_{north} = \frac{\hat{T}_{[i-1,j,h]}}{r_{north,[i,j,h]}} \quad \text{(Eq. 3.5-8)}$$

$$Q_{south} = \frac{\hat{T}_{[i,j,h]}}{r_{south,[i,j,h]}} \quad \text{(Eq. 3.5-9)}$$

With this we can calculate their respective resistances values as shown in (Eq. 3.5-10) and (Eq. 3.5-11).

$$R_{north} = \frac{1}{\omega \cdot C_p} \quad \text{(Eq. 3.5-10)}$$

$$R_{south} = -\frac{1}{\omega \cdot C_p} \quad \text{(Eq. 3.5-11)}$$

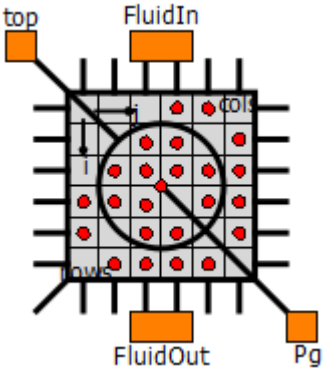

Figure 3-6. Core with convection cooling system representation in Modelica

The resulting block is presented in Figure 3-6, as it can be clearly seen this blocks lack the ports for all the directions except for the top one, this is due to the fact that chips are surrounded by plastic containers that completely insulate them form the environment. And this was hard coded in the module to limit the amount of ports needed since a large amount of ports will complicate the model unnecessarily. The power injection port and the top port are coded equally to the "parallelepiped" model.

The last addition of this model are the fluidIn and fluidOut ports. This ports are configured as an array of dimension equal to the number of tubes. This is because in each port we only carry information regarding the input or output conditions of the tubes. The fluidIn port injects the initial temperature of the fluid at the inlet, the dissipated heat and the mass flow rate of each individual tube. The port fluidOut carries the information of the maximum temperature of the whole silicon bulk (used in the control mechanisms) as well as the

temperature of the fluid at each the outlets, the mass flow rate at the outlet and the dissipated heat.

3.5.1 Liquid metals

It is needed to determine which equation must be used to calculate the value of γ, which is fluid-dependent. As it has been explained before, there are two options for this value using the Dittus-Boelter or the liquid metal equations. The selected value to determine which one to use is the Peclet number (Eq. 3.5-12).

$$Pe = Re \cdot Pr \qquad \text{(Eq. 3.5-12)}$$

If this number is lower than 0.5 it is known that the fluid use had characteristics similar to the liquid metals. As such we use the equations in section 2.1.1 to obtain the following equation (Eq. 3.5-13).

$$\gamma = \left(4.82 + 0.0185 \cdot \left(\frac{\omega \cdot C_p}{D \cdot \lambda}\right)^{0.827}\right) \cdot \frac{\lambda}{D} \qquad \text{(Eq. 3.5-13)}$$

Note that this is due to the fact that in this kinds of fluids the thermal conductivity property is high enough to aid in the cooling even when convection plays an important part in the cooling process. This equation is applicable when using our candidates: mercury and gallium.

3.5.2 Water

On the other hand, when the Peclet number is greater than 0.5 convection effects completely overrun the conduction effects and in this case it can be assumed that the fluid will work according to the Dittus-Boelter equation. Taking this into account, we use the same approach as in the previous section to obtain equation (Eq. 3.5-14).

$$\gamma = 0.023 \cdot \left(\frac{\omega}{D}\right)^{0.8} \left(\frac{C_p}{\mu}\right)^{0.4} \frac{\lambda^{0.6}}{D} \qquad \text{(Eq. 3.5-14)}$$

This equation will be used when evaluating the performance of water. This is the only fluid in our candidates that will not apply the liquid metal considerations.

3.5.3 How the tubes are modeled

With all the equations sorted the following step is to explain how the tubes are modeled in the Modelica library. This is not a trivial challenge since this is how the system choses if the FV in the position [i,j,k] is a fluid block or a silicon block. As such it means that the way to

identify the tubes must contain all the blocks that are part of a tube. It is also important to note that since the tubes flow from north to south this means that the tubes go through all the rows. Tubes should only be present on a layer of the simulation (in our case it will always be the top part). Finally, it is clear that the amount of cubes must be proportional to the amount of tubes.

Taking all of this into account the best way to save the position of the fluid blocks is to build a matrix with a number of columns equal to the number of tubes, with a number of rows equal to the number of blocks per tube and three layers, representing the positions of the block in the temperature matrix. The first layer correspond to the *i* index (row), the second one is the *j* index (column) and in the third layer the *k* index (layer).

In the code this matrix is known as ***tubesSpec***. For each tube, a column is created which contains n rows and three layers. Each of the layer correspond to the positions i, j or k of a fluid block. This process is better illustrated with an example. Take for example Figure 3-7, which shows the top layer in which the tubes are present. They are highlighted in gray while the white blocks are silicon blocks.

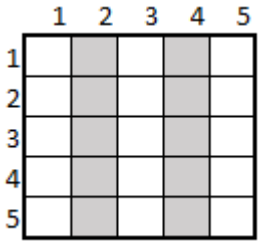

Figure 3-7. Grid representation of the tubes

Assuming that this is the third layer of the discretization, the resulting matrix is reported on Table 3-1. In the table we have shown every layer of the matrix in the same cell, separating each one by commas.

Table 3-1. Resulting ***tubesSpec*** matrix

1,2,3	1,4,3
2,2,3	2,4,3
3,2,3	3,4,3
4,2,3	4,4,3
5,2,3	5,4,3

With this matrix now you can know for example that the second block of the first tube is located in the second row, the second column and in the third layer and its temperature is $T_{[2,2,3]}$.

The utility of this matrix is that after the whole core has been modeled as a silicon blocks, a simple pass over this matrix can be used to replace the silicon blocks of the tubes with the fluid blocks. We simply take every simple position saved in this matrix and replace their properties (density, thermal conductivity, thermal resistances, etc.) with the fluid properties. And with this swap the core is modeling now correctly the tubes with the fluid flowing through them and the silicon bulk around them.

3.6 Control values

A quick glance through the equations and the modeled system show that the only variables which are not a material property or a geometrical constraint are the flow rate inside the tubes and the temperature at which the flow goes inside the tubes. This last one is not a parameter over which we have control because it will depend on the return circuit. Applying a control scheme on the return system is not feasible since adding extra components to a miniaturized system will consume resources mainly space and computation capability. The value of this field can be assumed to be ambient temperature. Again this is due to the fact that we can design the geometry of the return circuit and with a proper design guarantee almost perfect dissipation of the heat taken from the core.

As such we can only fully control the flow rate of the fluid, using micro-valves as explained in section 2.4. While it may be tempting to implement a strategy that modifies the flow through each tube, current chips don't provide localized temperature information and only provide a single temperature measurement. As such the flow rate through all the tubes must be used to control the temperature of the whole system. It will be injected through a block which takes the control temperature readings and sets a value for the flow rate inside the tubes.

Figure 3-8 shows one of the blocks created to control. As it can be seen it has the same two ports ***fluidIn*** and ***fluidOut***. This port names correspond to the name of the ports in the core block where they must be connected (i.e. ***fluidIn*** must be connected to ***fluidIn***). From ***fluidOut*** it gets the control temperature and determines the temperature at which the flow goes inside the core and the flow rate. These two variables are then injected into the core through the ***fluidIn*** port. The control block must be configured to comply with the restrictions previously describe on this section. This is done through the parameters ***wnom*** (which is the value of the nominal flow rate) and ***Tin*** (which is the value of the temperature at the intlet of the tubes)

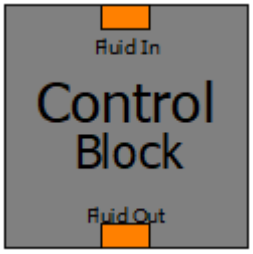

Figure 3-8 Control Block representation in Modelica

The current control block models an ON/OFF controller for reasons that will be explained in Chapter 5. If a different approach is needed this block can be modified to implement a different control law by rewriting the equation of the block under the line //Control Law. As shown in Figure 3-9.

```
model FluidControlONOFF
  extends ThermalModel.Icons.ControlBlock;
  // Tubes Parameters
  parameter Integer numTubes = 2;
  parameter Modelica.SIunits.Temperature Tin(displayUnit = "K") = 66 + 25 + 273.15;
  parameter Modelica.SIunits.MassFlowRate wtot = 8.58e-3;
  parameter Modelica.SIunits.Temperature Tswitch(displayUnit = "K") = 70 + 273.15;
  //Connection
  Connectors.FluidOut fluidOut(numTubes = numTubes) annotation( ...);
  Connectors.FluidIn fluidIn(numTubes = numTubes) annotation( ...);
equation
//Effects
  for t in 1:numTubes loop
//Control Law
    fluidIn.w[t] = 2*wtot*(70-fluidOut.Tcontrol);
    fluidIn.Tin[t] = Tin;
  end for;
end FluidControlONOFF;
```

Figure 3-9. Coding example of a simple P controller.

This new equation must be dependent on T_{max}

3.7 Model verification

While the model has been constructed with utmost care its results must be validated with some kind of test. Since no other proved simulator for this particular system exist we must reduce the model to cases whose theoretical solution is known. At the same time, we must be sure that the tests performed allows us to verify each important feature of the behavior of the model. Therefore, three test are executed: A Fourier test to validate the effect of the conductive part of the model (i.e. the silicon bulk), an exponential test to validate the behavior of the convective part and finally a delay test to test the behavior of the flow inside the tube.

3.7.1 Fourier test

In this test, we are only testing the steady-state thermal behaviour of the silicon bulk. We want to test if the model can work as a simple resistance through which a heat flow of 100W is flowing from an unknown temperature to a known one. This is done by simple generating heat through the whole lower layer equally. The expected value can be calculated by using the equivalent electric circuit shown in Figure 3-10. This circuit can be represented by this formula.

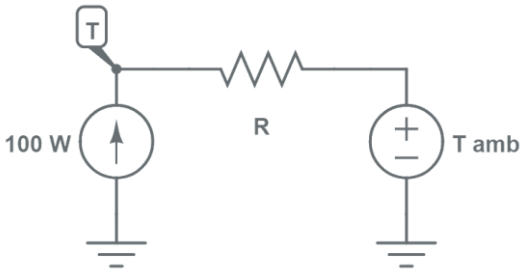

Figure 3-10. Equivalent circuit for the Fourier test

$$T - T_{amb} = \dot{q} \cdot R \qquad \text{(Eq. 3.7-1)}$$

Where $\dot{q}$ is the expected heat flow rate and R is the equivalent resistance of the whole silicon die. In this equation we need to calculate T (Eq. 3.7-2) and compare it to the obtained value of the simulation, where t is the thickness of the die and l is the length of the die

$$T = \frac{\dot{q} \cdot t}{\lambda \cdot l^2} + T_{amb} \qquad \text{(Eq. 3.7-2)}$$

This particular simulation setup is shown in Figure 3-11.

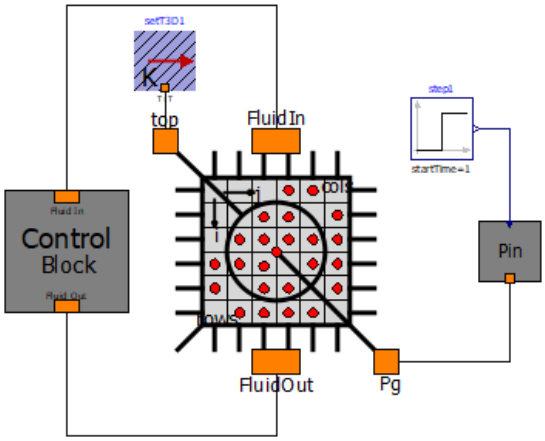

Figure 3-11. Modelica representation of the Fourier test

In this reported validation simulation, the following values of the parameters were used to obtain the result in (Eq. 3.7-4):

$$t = 1.25x10^{-3}\,m \quad l = 0.01\,m \quad \lambda = 149\frac{W}{m\,K}; \qquad \text{(Eq. 3.7-3)}$$

$$\dot{q} = 100\,W \qquad T_{amb} = 318.15\,K$$

$$T = 326.54\,K \qquad \text{(Eq. 3.7-4)}$$

With the simulation result giving T = 326.539 with a discretization of 10x10x3 core discretization. The same result was obtained using a 20x20x3 discretization verifying the convergence of the test. This result verifies the conduction part of the model since the theoretical result is identical to the one obtained in the simulation.

3.7.2 Exponential test

The exponential test is called this way since we are referring to the expected form of the result. This test consists in heavily increasing the thermal conductivity and specific heat of the bulk to guarantee that they will remain at constant ambient temperature for the duration of the simulation. The core block this case will have only one tube in the middle of the die. This transforms our block into a good representation of our design theoretical model discussed in section 2.1.1. We then know the expected profile of the temperatures inside our tubes (Eq. 3.7-5). For our reported test the theoretical curve follows the equation (Eq. 3.7-6).

$$T(x) = (T - Tamb) \cdot e^{-\frac{x \cdot NTU}{L}} + T_{amb} \qquad \text{(Eq. 3.7-5)}$$

$$T(x) = (66) \cdot e^{-189 \cdot x} + 298.15 \qquad \text{(Eq. 3.7-6)}$$

Figure 3-12 shows the simulation results for different discretization with n number of rows. It shows that the results start converge to the expected result the bigger n is.

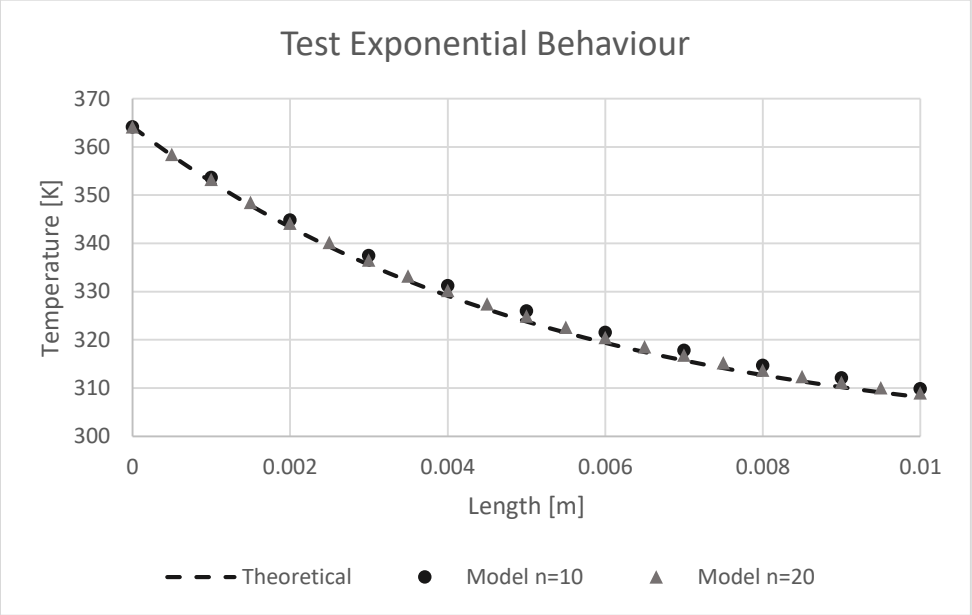

Figure 3-12. Results of the exponential test

3.7.3 Delay test

The final test validates the fluid thermal dynamics of the model. The idea is to reduce the conductivity of the bulk to avoid heat losses inside the tube and then to apply a step change of temperature at the inlet at time t = 0. As such we must expect that at the outlet we see the inlet temperature delayed by the time that the fluid takes to flow through the tube.

Since we are using a FV discretization method there will be some numerical diffusion that smooths out the sharp temperature change so the expected result is that with more steps and the same parameters the outlet temperature graph will converge to the ideal delay. Figure 3-13shows that this is the case.

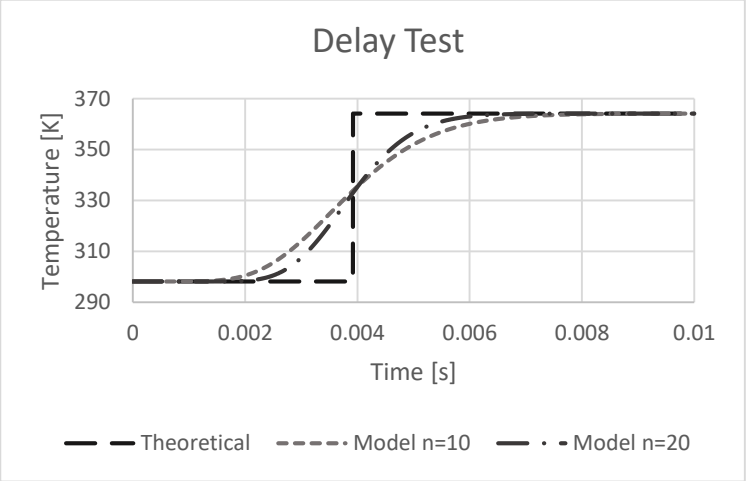

Figure 3-13 results of the Delay test

With this we can be certain that our Modelica blocks are working correctly.

Chapter 4.
MODEL ANALYSIS

With a fully validated model we proceed to simulate possible cases in which the heat produced failure. A standard simulation model must be built that can be use in various different situations. This is done using the model shown in Figure 4-1.

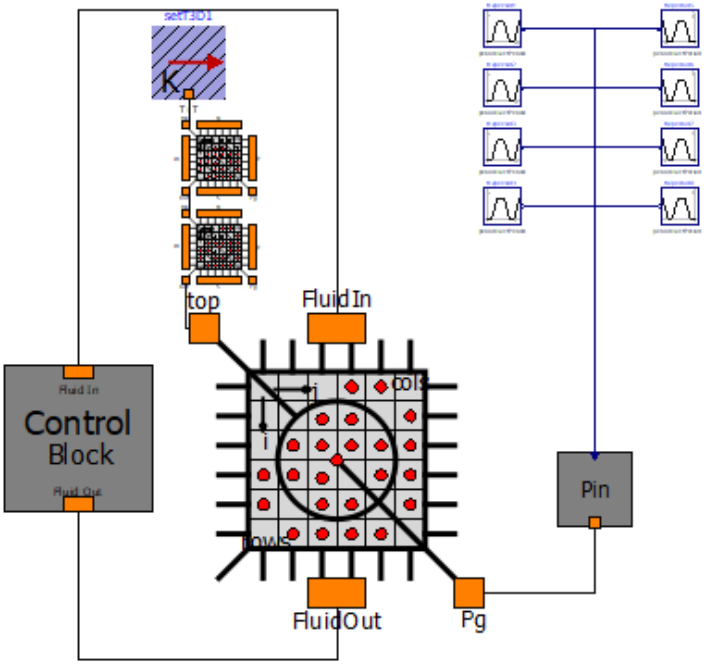

Figure 4-1. Simulation benchmark in Modelica

This model let us create 8 different areas of activation, which follow trapezoidal power profile with controllable period, active time, start time and amplitude. Note that the power injection block and two "parallelepiped" blocks are used to represent the layer of copper and aluminum that are part of the spreader and heat sink. This is needed instead of modeling a single spreader because this changes in the material must be considered independently. Keep

in mind that in real life the spreader is substantially larger than the core, but due to how the core with a cooling fluid must be connected they have been compressed for simplicity. The external temperature must be connected to the top port of the spreader. And finally the control block has been programed so that it is always working by activating at 0°C.

The geometric constraints that are common to all further simulations are shown in

Table 4-1. Geometric constraints of the simulation

Fluid block	
Length	10 mm
Width	10 mm
Thickness	1.5 mm
Rows	20
Columns	20
Levels	3
Tube diameter	0.5 mm
Copper block	
Length	10 mm
Width	10 mm
Thickness	3 mm
Rows	20
Columns	20
Levels	3
Aluminum block	
Length	10 mm
Width	10 mm
Thickness	10 mm
Rows	20
Columns	20
Levels	3

Taking into account that in the worst case the heat will be concentrated in an area of 1mm x 1mm, the layer of the tubes has the distribution shown in Figure 4-2.

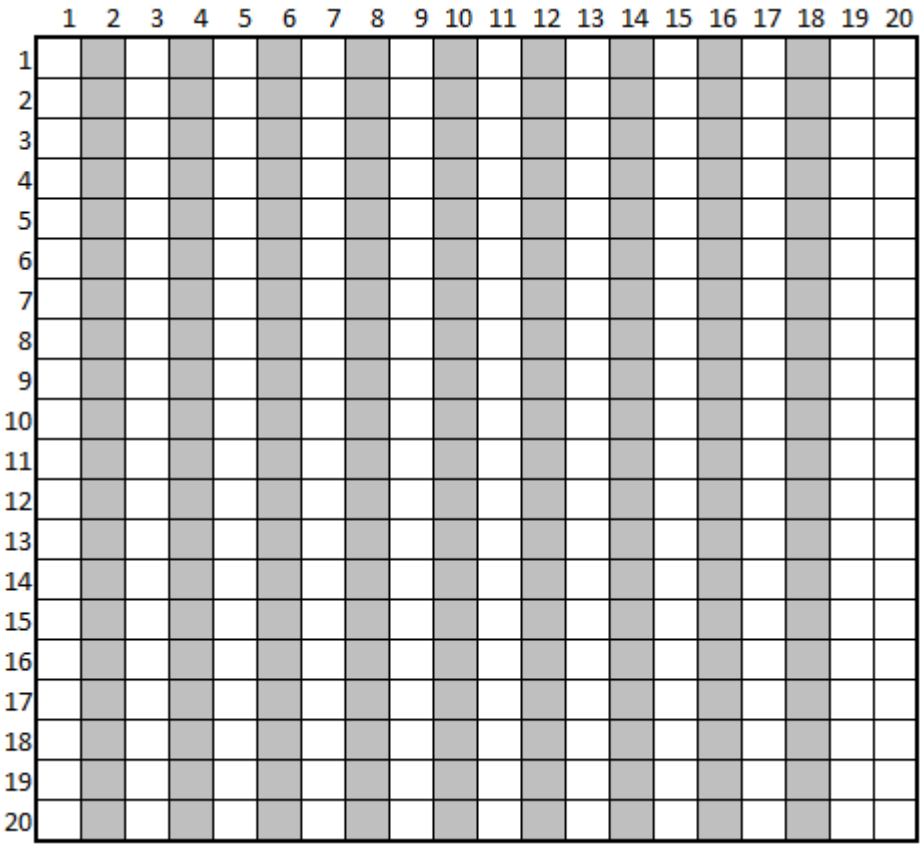

Figure 4-2. Tubes distribution

Finally, the four cores where embedded in a single die simulated in a single core as such the limits for each core are shown in gray in Figure 4-3.

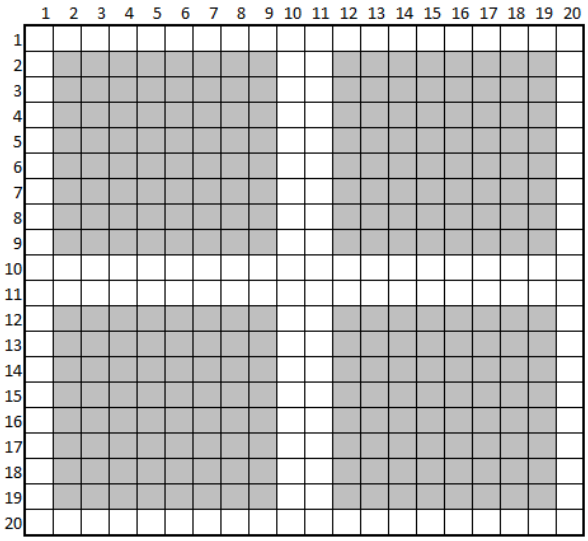

Figure 4-3. Core limits and distribution

4.1 Hotspot selection and critical cases

The next step is to simulate the hotspots which produce the most heat since this is the case that will be critical and this critical situation is the one we must guarantee the convection cooling works. A pattern was found through the use of simulations: the more critical cases are when two hotspots coincide in the same tube or when they are close to the border of the die, while the lower boundary is when the cores heat is injected right in the middle of the cores.

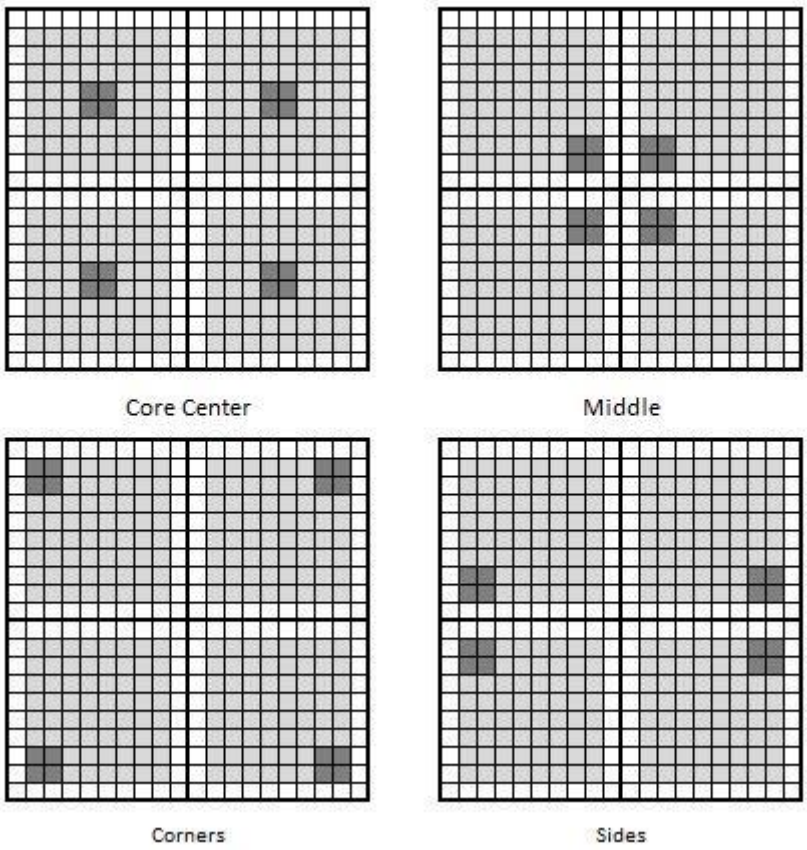

Figure 4-4. Configurations description and names

Based on this observations 4 critical cases were selected and analyzed. They are shown on Figure 4-4 along with their name given for the sake of simplicity.

In Modelica the existence of a hotspot is identified by simply looking at the maximum temperature inside the die.

4.1.1 Core center

This configuration is the one which provides less temperature increase. This is due to the fact that heat has the capability of spreading in all directions and as such is easier to cool down.

This limit provides an achievement limit since this will be the lowest achievable value of the cooling system.

4.1.2 Middle

This configuration occurs when all the hotspots are very close to each other. This configuration of hotspot seems like a critical configuration but since it still leaves a lot of room to dissipate heat around the conglomerate of hotspots.

4.1.3 Sides

This is the most critical situation since two hotspots are really close to each other and there is only room to dissipate in one direction. As it will be shown later this is the configuration in which at long time schedules this reaches the maximum temperature values.

4.1.4 Corners

Dynamically very close to the sides configuration and it is critical since two of the 6 conduction mechanisms are almost useless. Also this is critical because it heats the fluid right before the outlet.

4.1.5 Comparison

Figure 4-5 shows a comparison of the four configuration and the temperatures reached by each one of them. This graph serves the purpose of identifying the lowest and higher temperatures that must be used in the following simulations.

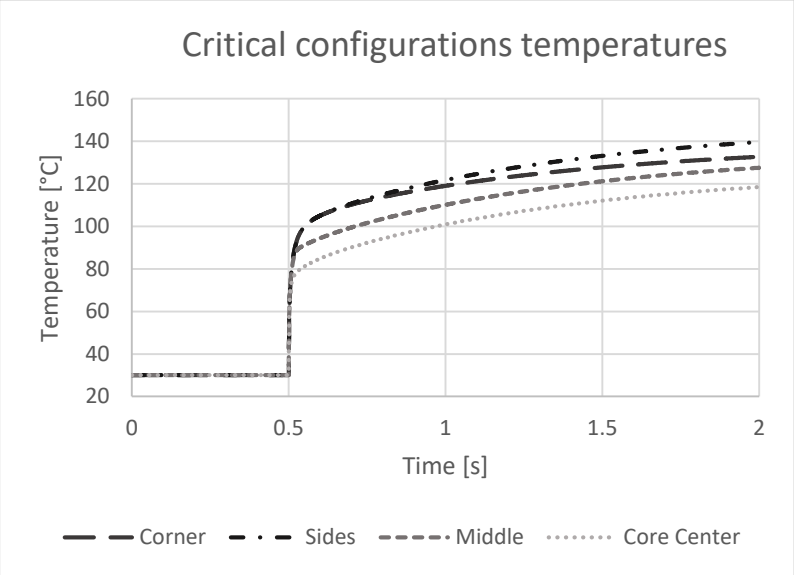

Figure 4-5. Temperature comparison for critical configurations

4.2 Final fluid selection

Based on the previous results the following step is to test the three fluid candidates in a situation in which the model switches continuously between the sides configuration and the core center configuration. This is done for each fluid and the comparison graph is shown in Figure 4-6.

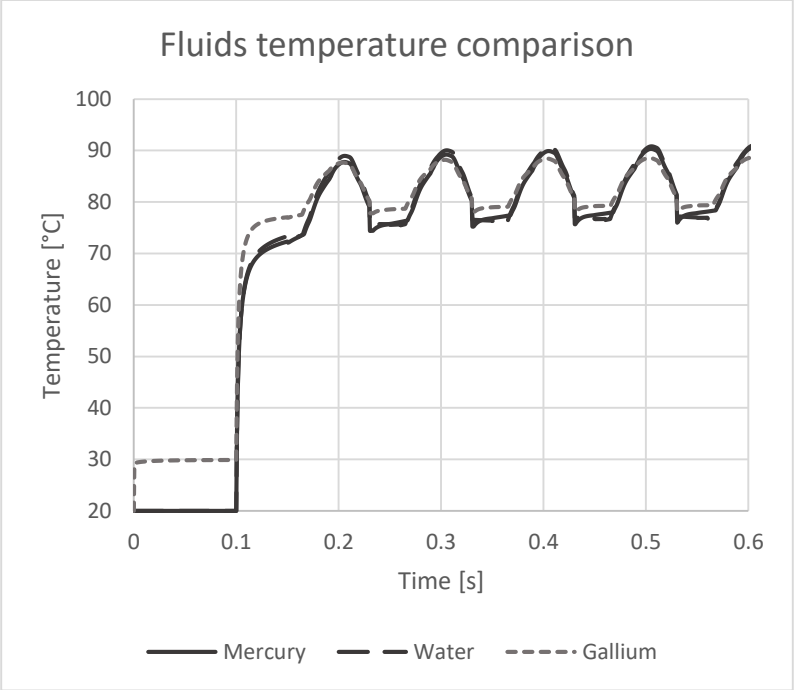

Figure 4-6. Comparison among fluids

Since the behavior is clearly similar a zoom of the graph is presented in Figure 4-7.

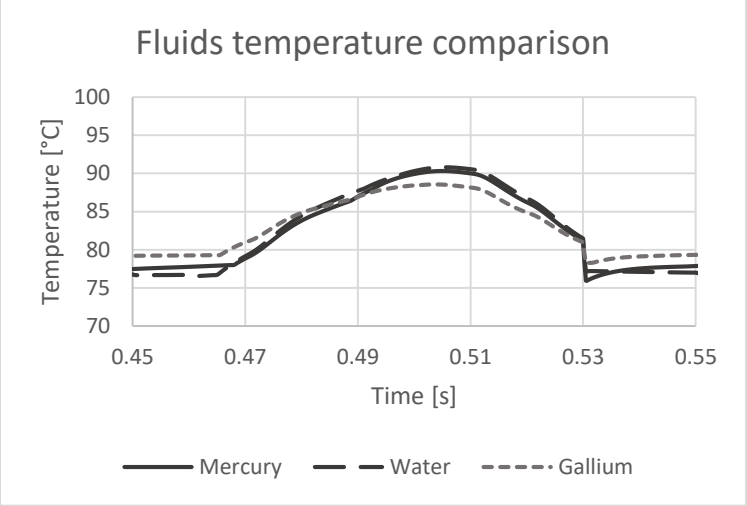

Figure 4-7. Zoom on the fluid comparison

It is important to note that none of the fluids goes over the 100°C threshold meaning that all three can be used for the prototype.

4.2.1 Gallium

Gallium has the problem of starting at a temperature of 30 °C instead of the 20° the other fluids have. This is due to it solidifying temperature just below 30°C. It is the fluid that reaches the lower maximum temperatures. However, it is not noticeably better than mercury although it requires significantly more power to pump as discussed in Chapter 2.

4.2.2 Water

Amongst the selected fluids it seems to be the one that heats the most. It is also the one which requires more pressure. However, its high availability, low price and zero toxicity may be a strong enough reason to select this as the final fluid.

4.2.3 Mercury

Mercury seems to be the more suitable fluid for the job since it is the one that reaches the lowest temperatures. In our previous considerations Mercury is also the fluid with the lowest power consumption. This is a great deal since we are trying to create a system. As such it is mercury the selected fluid to be used in the control simulations.

Chapter 5.
CONTROL SCHEMES

As it has been mentioned in section 1.1 a very important resource to manage in microprocessors is cooling. As such simple control methods are preferred since the usually require low computational processing. You also want to keep in mind that power consumption has a high impact on the desired performance since to reduce cost it should be as small as possible ideally substantially less than the consumed power of the core. While a quick glance at the data in section 2.2 shows that even for 10 tubes the power consumption is under 1%, but for some extreme applications (such as space exploration) this values may be important and should be minimized.

5.1 Why ON/OFF

An ON/OFF controller may be the simplest controller that may be implemented. A threshold temperature is defined. If the temperature reading of the core goes over that threshold the pump system is activated. If the temperature goes under that temperature the pump system turns down. This simple comparison will consume almost no computational resources. Again this is also a very simple control scheme that requires also low simulation times due to its simplicity. Both this factors are extremely appealing in this situation so this is the implemented control.

5.2 Feedback selection

As explained in section 1.1 one major problem in the control system is that the exact way the temperature is measured for thermal control in commercial microprocessors is not known. In this thesis we work with two options: either this temperature reading is the maximum

overall temperature in the core or it is an average of the temperature in the active layer of silicon. These choices change significantly how the control work.

5.2.1 Maximum core temperature

With the maximum core temperature as a reading the ON/OFF controller works pretty straight forward. Since the dynamics of the pump are fast enough the system almost immediately achieves the expect dynamic with the flow running. When setting the threshold temperature at 75°C, the obtained result is shown in Figure 5-1, the graph of pressure and power are not displayed due to its dependence with the flow parameter. This dependence will be explained in section 5.3. As you can see once he core is heated the core will never go under 75 °C.

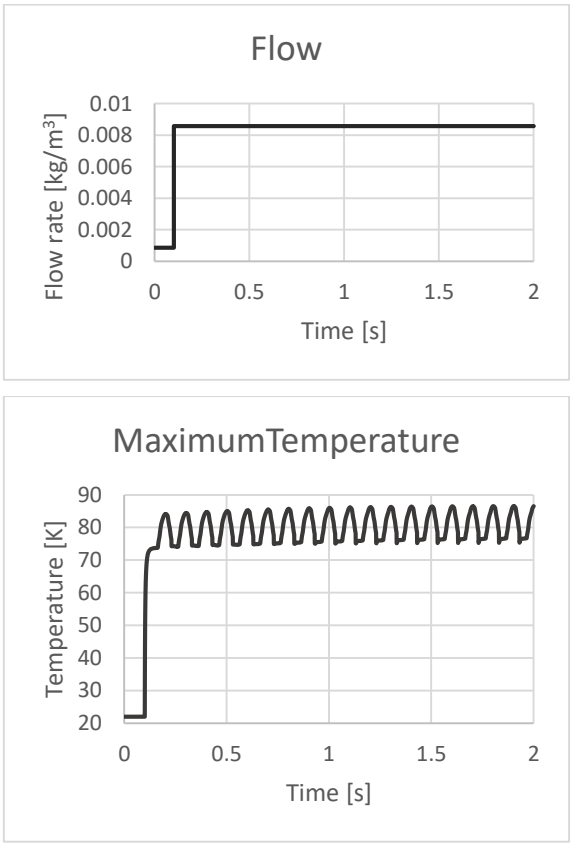

Figure 5-1. Max temperature control result

While this may seem very power intensive you can chose a higher value (always under 90°C to be safe) to decrease the amount of time the system is at nominal flow. This will not change the graph significantly since it will mainly truncate the graph not at 75°C but at your selected point.

5.2.2 Average core temperature

When using the average core temperature, a different problem arises since in this case an overshoot is present at the beginning of the process. This happens because the core needs more time to heat up all the layers than the time in which a hotspot is created. But with proper tuning (threshold temperature at 40°C) the overshoot can be contained under the 100°C. This result is shown in figure

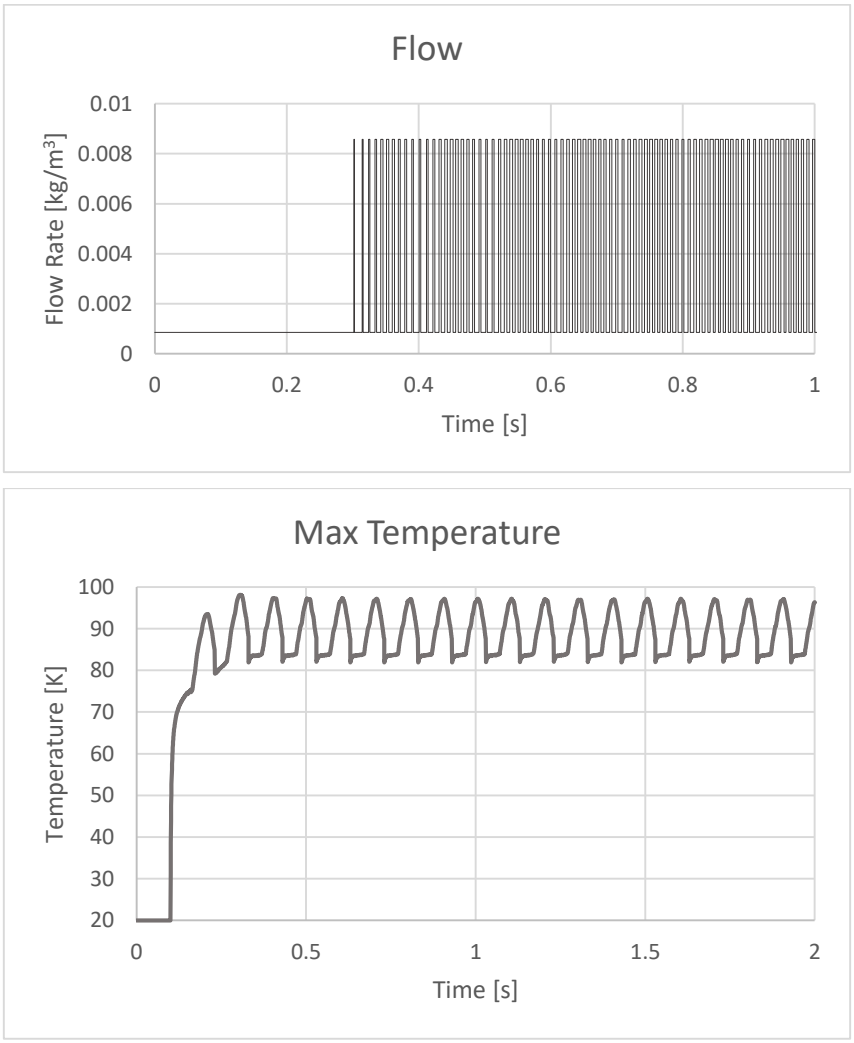

Figure 5-2 Average temperature control results

As you can see from the graph power consumption in this method is heavily reduced but the system now gets closer to the 100°C limit. But even if the maximum temperature is not directly recorded this control system is enough to prevent the core for heating up to a dangerous level.

5.3 Dependence on flow rate and warning on other control methods

Finally, an explanation must be made regarding modulation control strategies, while flow control may be possible thanks to the micro-valves it is not recommended to change the flow to another value other than the nominal one. Serval simulations show the dependence of the temperature compared to flow rate in Figure 5-3.

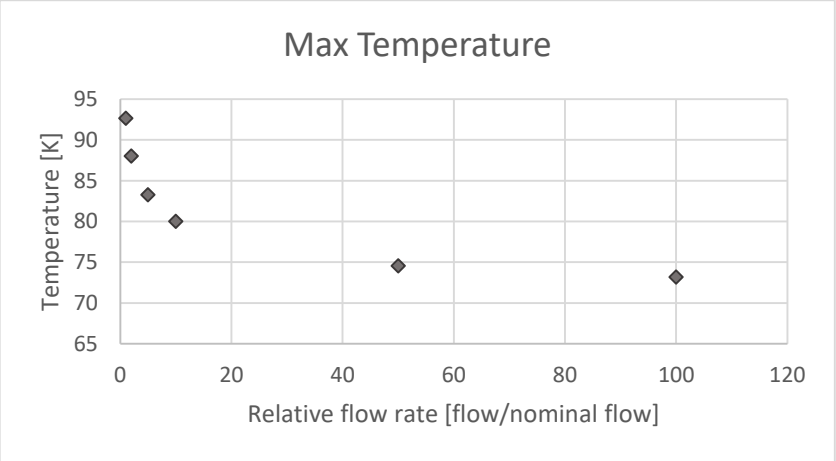

Figure 5-3 Relation between flow and max temperature

As you can see to get a reduction of 10 °C you need an increase the flow by 10 times. And to reduce it by 20 °C you need to increase it by 100 times. While plants with similar relations can be controlled without problem the main value that restricts our control to be modular is the rapidly increase in pressure drop needed.

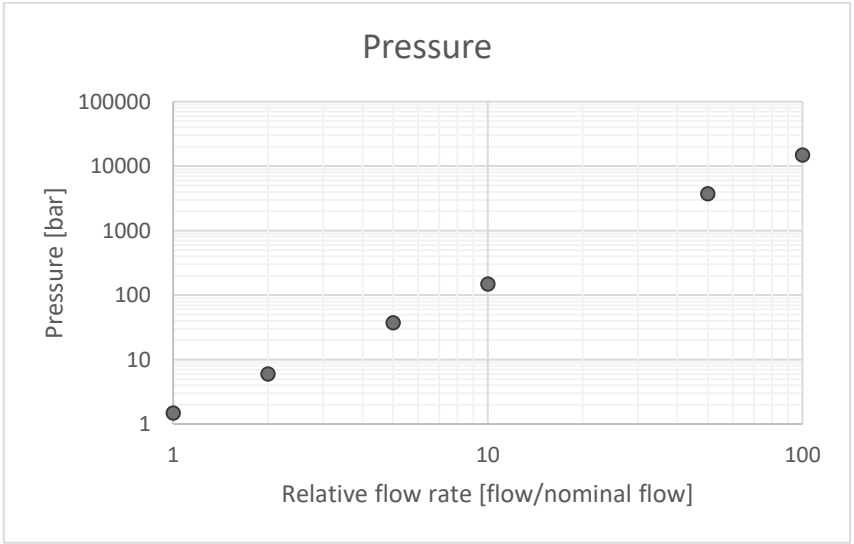

Figure 5-4. Relationship between flow and pressure

Figure 5-4 shows the logarithmic dependence of pressure with flow. The relationship is show using the relative flow rate in comparison with the nominal flow of the fluid. As it can be seen, by simply doubling the flow rate pressure in mercury has gone over our limit of 3 bar. This leaves us with an extremely narrow range of available flows. That can barely produce a change of 5°C. It also should be reminded that going under the nominal flow will result in the loss of turbulence which will rapidly decrease the ability of the system to cool the chip and is extremely dangerous because temperature will quickly rise over the limit. Additionally, power increases on a cubic way instead off in a quadratic way. This means that the increase of power quickly increases beyond the core power of 100W.

With all of this into consideration continuous control strategies are discarded and in fact is the ON/OFF control strategy selected as the most appropriate for this system.

Chapter 6.
CONCLUSIONS

This work has demonstrated the feasibility of the convection cooling model at least at a simulation level. Extending in this way the long term research activity aimed at the development of new techniques for integrated thermal/power/performance management in high-power microprocessors.

The advantages of each fluid have been presented and compared among them to select the best possible candidate for this system. Selecting mercury as the most fitting even when taking into account its toxicity due to its incredible performance and low power consumption.

The proposal of novel fluid into this new control strategy increases the applicability of such systems since it reduces substantially the pressure differential needed allowing for the use of micro-valves in a feasible way. This make the cooling strategy a more realistic product in the future.

This cooling mechanism has been successfully implemented and validated at a simulation level, successfully overcoming all the verification test. This increases the trust in the cooling method allowing it to move into the next implementation phase.

A control strategy has shown that a simple and not computational consuming control law is possible. As it has been explained this is the best control strategy for this system due to the limitations shown. In particular, due to the great increase in pressure that arises from a slight increase in the flow rate.

Taking all of this into consideration this is just an introductory study and a lot of future work is required, before this technique can be implemented fully.

6.1 Future Research

Some detail design will be needed to create a functional prototype. In particular, further analysis is needed in the return flow and the interface between the tubes as mentioned above. This is needed to select the final appropriated pump and valves required.

The next step for this project is to create a prototype of the suggested strategy to validate the simulations with data from an experiment. This is a very important step and can be done initially using water to avoid toxicity problems. The scope of this first prototype is to validate the simulation model and gather data in particular in the temperature read by the real sensors.

Using this data further improvements in the model can be implemented. As well as mixed strategies such as coupling this convection cooling system with other control strategies currently in use DVFS.

Another way of improving the usefulness of this models is to obtain substantially more credible data about the patterns of power injection in a real processor. Also more specific information will be required on the actual load patterns of the processor and on the actual location of temperature sensors. However, this requires an experimental set up that may be extremely time consuming.

Finally, once all of this is validated a mercury prototype can be constructed to analyze the performances of this fluid.

Bibliography

[1] L. Alberto, F. Terraneo, I. Giacomello y W. Fornaciari, «Event-based power/performance-aware thermal management for high-density microprocessors,» *IEEE Transactions on Control Systems Technology ,* vol. 26, nº 2, 2018.

[2] P. Heremans, G. V. den Bosch, R. Bellens, G. Groeseneken and H. E. Maes, "Temperature dependence of the channel hot-carrier degradation of a-channel mosfet's," *IEEE Transactions on Electron Devices,* vol. 37, no. 4, 1990.

[3] M. L. Dreyer, K. Y. Fu and V. C. J., "An electromigration model that includes the effects of microstructure and temperature on mass transport," *Journal of Applied Physics,* 1993.

[4] C. Ma, H. Mattausch, M. Miyake, T. Iizuka, M. Miura-Mattausch, K. Matsuzawa, S. Yamagushi, T. Hoshida, M. Imade, R. Koh, T. Arakawa and J. He, "Compact reliability model for degradation of advanced p-mosfets due to nbti and hot-carrier effects in the circuit simulation," in *IEEE International Reliability Physics Symposium (IRPS)*, 2013.

[5] B. Vandevelde and E. Beyne, "Improved thermal fatigue reliability for flip chip assemblies using redistribution techniques," *IEEE Transactions on Advanced Packaging,* vol. 23, no. 2, 200.

[6] F. Zanini, D. Atienza and G. De Micheli, "A control theory approach for thermal balancing of mpsoc," in *2009 Asia and South Pacific Design Automation Conference*, 2009.

[7] S. Murali, A. Mutapcic, D. Atienza, R. Gupta, S. Boyd, L. Benini and G. De Micheli, "Temperature Control of High-Performance Multi-core Platforms Using Convex Optimization," in *2008 Design, Automation and Test in Europe*, 2008.

[8] F. Zaninit, C. N. Jones, D. Atienza and G. De Micheli, "Multicore thermal management using approximate explicit model predictive control," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, 2010.

[9] D. Cuesta, J. L. Ayala, J. I. Hidalgo, d. Atienza, A. Aquaviva and E. Macii, "Adaptive task migration policies for thermal control in mpsocs," in *2010 IEEE Computer Society Annual Symposium on VLSI*, 2010.

[10] N. Lamaison, J. R. Thome and J. B. Marcinichen, "Dynamic Modeling and Experimental Evaluation of a Controlled Two-Phase On-Chip Cooling System Designed for High Efficiency Datacenters," Lausanne, EPFL, 2014.

[11] M. Damiani, "Event-based power/performance/thermal management with task migration in high-power CPUs," Magister Thesis, 2018.

[12] F. Terraneo, "Thermal and energy management techniques for multi-core and many-core systems," PhD Thesis, 2014.

[13] L. Alberto, F. Terraneo and W. Fornaciari, "Event-based control as an enabler for high power density processors," in *2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, 2016.

[14] F. Casella and A. Leva, "Modelling of Distributed Thermo-Hydraulic Processes Using Modelica," in *Proceedings 4th Mathmod Vienna*, 2003.

[15] F. J. Mesa-Martinez, J. Nayfach-Battilana and J. Renau, "Power Model Validation Through Thermal Measurements," *ACM SIGARCH Computer Architecture News,* vol. 35, no. 2, 2007.

[16] F. P. Incropera, T. L. Bergman, A. S. Lavine and D. P. Dewitt, Fundamentals of heat and mass transfer, Jefferson City: Jonh Wiley & Sons, 2011.

[17] Intel, "INTEL® CORE™ i7-9700K PROCESSOR," [Online]. Available: https://www.intel.com/content/www/us/en/products/processors/core/i7-processors/i7-9700k.html. [Accessed 20 November 2018].

[18] Anton Paar, "Anton Paar," [Online]. Available: https://wiki.anton-paar.com/en/engine-oil/. [Accessed 25 November 2018].

[19] MicroPUMPS, "Innovative MicroPUMPS for infusion and injection biomedical devices," [Online]. Available: http://www.micropumps.fr/. [Accessed 25 November 2018].

[20] DMQ, "MEMS Microvalve Technology," [Online]. Available: http://www.dmq-us.com/mems-microvalve-technology/. [Accessed 26 November 2018].

[21] Intel, "The Compute Architecture of Intel® Processor Graphics Gen9," 2015.