Microprocessors: A Hot Topic

M.L.Roxburgh

***Abstract*—A computational 2D model was used to determine the temperature of a microprocessor at maximum power output. It was found that with no active cooling, the processor will easily reach melting point. With a heat sink and forced convection, it was able to maintain a temperature below $100°C$.**

I. Introduction

MICROPROCESSORS are central processing units (CPUs) that are embedded onto a single microchip as defined by S.Furber[**1**], and are used in a vast array of electrical devices at differing scales; from household appliances such as a microwave, to the largest supercomputers. Since the initial release of the Intel 4004 in 1971[**1**], microprocessors have increased in computing power and reduced in size, following a trend similar to that stipulated by G. Moore whereby the number of transistors on a microchip doubles every 2 years.

However, we are approaching the physical limits of the size of transistors. As of June 2022, the South Korean chipmaker 'Samsung' started shipping a microprocessor with a gate length of 48nm[**2**] which is only 2 orders of magnitude away from atomic radii. The other major issue with the shrinking of transistor size is the energy dissipated in the form of heat from the processes of the chip.

An Intel i9-14900K draws 144W of power at maximum performance and such microprocessors can have power densities, q on the order of $5 \cdot 10^8$ W/m^3.

This investigation models the temperature distribution and dissipation in a discretised 2D model of a cross-section of a microprocessor. Three systems were modeled; just the microprocessor and its CuW ceramic under natural convection; the same system with an aluminum heat-sink with natural and forced convection. The properties of the heat sink were investigated to determine how the dimensions affected the temperature of the processor.

II. Mathematics, Method and Implementation

The primary equations governing the dynamics of the system in question, relate the temperature (T) as a function of position in an object at steady state, to q by a second-order elliptic differential equation in the form of Poisson's equation (1)[**3**].

$$-k\nabla^2 T(\vec{r}) = q(\vec{r}) \tag{1}$$

The energy flux $\vec{\phi}$ at the surface of a body is dependent on the ambient temperature outside (T_a) of this body and the temperature of the body at the surface (T_s) in $[K]$, as energy moves in the form of convection.

$$\vec{\phi} = h(T_s - T_a) \tag{2}$$

Equation (2) highlights this relationship where h is the heat transfer coefficient $[Wm^{-2}K^{-1}]$ of the interface which has been empirically defined under the two schemes of natural and forced convection as quantified in (3) and (4) respectively.

$$h_{natural} = 1.31(T_s - T_a)^{1/3} \tag{3}$$

$$h_{forced} = 11.4 + 5.4v \tag{4}$$

Where v is the velocity of the air moving around the body in $[m \cdot s^{-1}]$.

To computationally calculate the second derivative of a discrete 2D grid, we may use a finite difference (FD) scheme for small h, coupling each point to the neighboring points in the grid in calculating the second derivative.

$$\nabla^2 T_{i,j} = \frac{T_{i\pm h,j\pm h} - 4T_{i,j}}{h^2} \tag{5}$$

The scheme (5) for calculating the second derivative as per the Taylor expansion of $T(x \pm h, y \pm h)$ is accurate to $O(h^2)$. Note the subscripts $T_{i\pm h,j\pm h}$ implies a sum of 4 elements $T_{i+h,j}$ + $T_{i-h,j}$ + $T_{i,j+h}$ + $T_{i,j-h}$. This system requires the use of a grid value on either side of the point where the second derivative is calculated. To reduce the error to $O(h^4)$, another

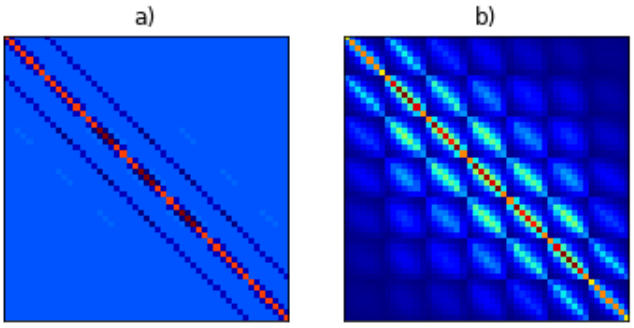

Fig. 1. a) A matrix formed by a grid of (7x7) for a combined finite difference scheme of (5) on the edges and (6) for the center points. b) The inverse of the matrix in a), calculated using Gaussian elimination.

point in each direction $T(x \pm 2h, y \pm 2h)$ must be considered giving scheme (6).

$$\nabla^2 T_{i,j} = \frac{\frac{4}{3}T_{i\pm h,j\pm h} - \frac{1}{12}T_{i\pm 2h,j\pm 2h} - 5T_{i,j}}{h^2} \quad (6)$$

In either of these schemes, it is possible to vectorise the grid and turn it into a matrix equation in the form of (7).

$$M \cdot \vec{\hat{T}} = \vec{b} \quad (7)$$

$\hat{T}$ here is the matrix of T, condensed to a 1D vector where the top row $(i = 0)$ are the first elements in order from left to right. M is the finite difference scheme put into matrix form which can then be inverted as shown in Fig. 1. $\vec{b}$ is a vector that holds the information about the boundary conditions (BCs).

In the system modeled, the BCs are Neuman and an artificial point must be created that satisfies; energy conservation at a boundary of another solid material; and convective equations for the boundary with air. This leads to (8) ,

$$-k_1 \nabla_j T_1 = k_2 \nabla_j T_2 \quad (8)$$

where the subscripts of T and k denote that they are from different materials, and the j subscript implies it is a derivative in the j direction assuming the boundary lies along i=constant. To ensure continuity, the artificial boundary values for each material are set to be equal with the form (9) satisfying (8).

$$T_{shared} = \frac{k_1 T^{above}_{i,j+h} + k_2 T^{bellow}_{i,j-h}}{k_1 + k_2} \quad (9)$$

To solve the equations, an iterative scheme is used whereby an update on the temperature of the system is made followed by an update on the boudnary conditions. The methods employed in this study, segment the different rectangular systems of the overall mesh into the processor, ceramic case, heat-sink body, and each individual fin of the heat-sink. As the system moves towards equilibrium, the temperature change after each iteration becomes smaller. Using this, we can start each iteration above and below our expected value and define convergence as the temperature where the above and below iterations agree to within a specified tolerance. Multiple methods were tested to solve this system, the first of which utilised a Jacobi iterative solver on the system (7). The method for this can be represented using pictorial operators for the $O(h^4)$ FD scheme (10).

$$T^{k+1}_{i,j} = \frac{-1}{5} \cdot (T^0_{i,j} + \begin{bmatrix} 0 & 0 & \frac{-1}{12} & 0 & 0 \\ 0 & 0 & \frac{4}{3} & 0 & 0 \\ \frac{-1}{12} & \frac{4}{3} & 0 & \frac{4}{3} & \frac{-1}{12} \\ 0 & 0 & \frac{4}{3} & 0 & 0 \\ 0 & 0 & \frac{-1}{12} & 0 & 0 \end{bmatrix} T^k_{i,j} + \frac{q_{i,j}}{k_m}) \quad (10)$$

Implementation of each scheme was consistent; do one iteration inside each block; do one iteration updating boundaries at air and any shared boundaries. The Jacobi (JC) method may be implemented using a pictorial operator or using $\hat{T}$ with a matrix that encodes the coupling. The same principle can be applied to a system with faster convergence such as the Gauss-Seidel (GS) Method for solving equations of the form (7).

The method that was used for the majority of this study was using a different method with faster convergence. As the coupling matrix, shown in Fig. 1. does not change as the boundary conditions are encoded into b, it is possible to use the analytical inverse of M for each segment during each iteration inside the boundaries to find the solution to (1). The inverse of each matrix was calculated before solving the entire system using Gaussian elimination which could then be used in the solve-update cycle. Theoretically, this limits the number of points in the grid that is feasible to use as the matrices are square and size nxn where n is the number of points in the grid. Gaussian elimination scales as $O(n^3)$, and by increasing the scaling (scaling is points per mm) n increases by $scaling^2$, as well as matrix multiplication increasing in time with respect to n.

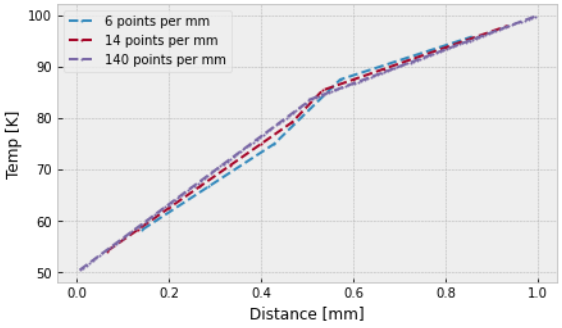

Fig. 2. Shows the effect of using a discrete number of points to represent interactions and its impact on calculating values at an interface. The higher the ppmm, the closer the values are to the analytical solution.

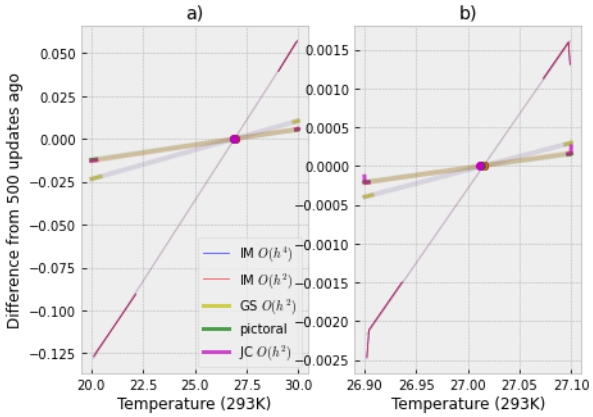

Fig. 3. comparison of 5 iterative solving methods a) far from equilibrium and b) close to equilibrium. The steeper the gradient, the faster the rate of convergence. All methods converge to the same value within 1K. Artifacts in b) are due to the system coming into a stable state. Both the set of IM methods overlap with each other and there is overlap between the set of JC and pictorial methods as each set does the same thing with minor differences in application

III. Method Validation

A. Boundary Interface

To test the boundary interface, a one-dimensional FD scheme on the same order as (5) was set up as the analytical solution for the Laplacian in 1D under Dirichlet BC is trivially a straight line connecting one boundary value to the other. Therefore, for the boundary between two solids, the expected solution is two straight lines with the gradients satisfying (8). As shown in Fig. 2., the expected solution is achieved but there are artifacts of the discretisation of the material. The interface has an artificial shared point (9) which is then used in the solution of the Laplacian. This causes discontinuity at the boundary. In the limit that the distance between points (h) approaches 0, the solution approaches the analytical continuous solution. Fig. 2. highlights that at lower points per mm (ppmm), the gradient is shallower in both sections left and right of the boundary than the solution at higher ppmm. This indicates there will be systematic errors due to each of the interfaces in the system with the gradient error in the test case being 2% and 4% for the higher and lower sections respectively.

B. Order of Magnitude Estimates

It is possible to calculate the order of magnitude for the expected values for the temperature for different systems. Using dimensional analysis, q and $\vec{\phi}$ may be related using the dimensions of the entire system. The processor has a cross-sectional area of $14mm^2$ and each system modeled has a different perimeter. Using these values as well as equations (2) and (3) for $\vec{\phi}$, the general estimates are given in TABLE I. A major assumption that is used is that the system is in equilibrium and the surface temperature is the same everywhere.

TABLE I
ESTIMATES OF SYSTEM TEMPERATURE ASSUMING ENTIRE SYSTEM IN EQUILIBRIUM

System	T estimate (°K)
just processor	8960
processor and case	6590
10 30mm fins with 2mm spacing	1140

C. Methods of Solution

The methods of solution tested include; using an inverse matrix to solve each section and then update boundaries (IM), this was done using the $O(h^4)$ and $O(h^2)$ scheme; GS iterative method where on each iteration updating the temperature values, the boundary values were also updated; and a Jacobi method in both pictorial operator and matrix form (JC) for the $O(h^2)$ scheme.

Testing each method was achieved by running 10^4 iterations above and below the equilibrium position. This is plotted in Fig. 3. a). A straight line is plotted between the above and below runs for each method and all these lines converge at the intercept of the difference axis. As the solution approaches equilibrium, the difference between each update becomes smaller. In Fig. 3. b) the system was set

to values just above and below equilibrium. All methods converge to the same temperature with a tolerance of 1 K. Sharper gradients imply a faster rate of convergence as there are greater value changes between each run. IM method converges much quicker than the other methods as applying the matrix to a section will completely solve the Laplacian or Poisson equation in that section and so information can travel instantaneously across each section whereas, in every other method, information travel is limited in speed via each iteration. in Fig. 3. b) there is a section that is not linear in each method. This artifact is due to the system starting at a uniform temperature and it taking a few iterations to form a stable shape of solution. this is achieved in fewer iterations for IM therefore it is in the linear phase for longer.

6 methods were tested for the time taken to do 10^4 updates on just the processor. The fastest method was the IM method (26.4s) followed by JC $O(h^4)$ and $O(h^2)$ (+5%) and GS method (+10%) and the slowest was using a pictorial operator (+78%). IM is the quickest as it does not require passing the grid of temperatures into the function as it solved using inverse matrix and the BCs. IM outperformed other methods in both run time and convergence with the slowest pictorial method being up to 2100% slower to converge to the same value.

Another optimisation utilised was the symmetry in solutions by halving the number of fins a solution was found for and then mirroring the array to find the other solutions.

D. Scaling

scaling or ppmm was the limiting factor in terms of systematic error. Fig. 4 shows that as you increase the ppmm, the temperature increases significantly with up to 25% change between the scaling of 3 and 9 in both systems. With a lower scaling value, the temperature used to calculate the derivative at the boundary gets closer to the center of the system where it is hottest. This creates an artificially high gradient meaning the calculated boundary value is lower than it should be resulting in equilibrium at a lower overall temperature. The equation modeled (1) implies at every point the second derivative is either zero or negative, which implies that any discretisation will be less than the continuous value and this becomes more accurate the closer the points are.

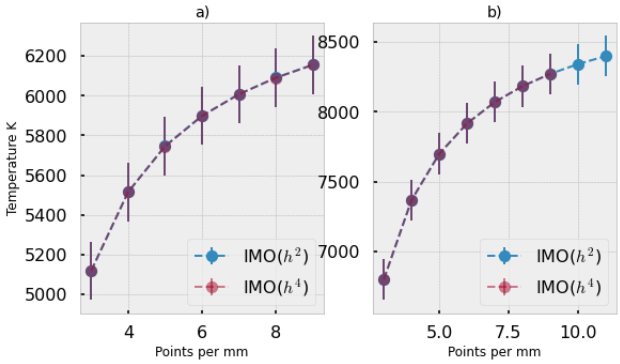

Fig. 4. a) and b) show the relationship between increasing the ppmm and the equilibrium nodeled temperature that the system comes to. a) models the processor and case b) is just the processor. Both follow the same shape and increase in temperature at finer resolutions.

IV. SUMMARY AND DISCUSSION OF RESULTS

For all simulations, IM $O(h^4)$ was used where after $2{\cdot}10^5$ iterations above and below, the difference and temperature were plotted as in Fig. 6. a) and used to find a value close to equilibrium and re-running for $2{\cdot}10^5$ the program about this new value.

A. No heat sink

1) Processor only: With the processor only, the average processor temperature was found using a system at a scaling of 11 and reached a temperature of $8400 \pm 100°K$. There is likely more systematic error from the scaling and the value of $8400{\pm}100°K$ is an underestimate. This already is well above the temperature that silicon would vaporize at and before getting to this, the system would break from overheating.

2) Processor and ceramic case: Fig. 5. a) depicts a heat map of the combined system with the ceramic case on top of the silicon processor. This was run at a scaling of 6. The overall temperature of the system is $> 6000°K$ and is lower than the system of the processor only. This again is well over the melting point of all components. Both with or without the case, this situation is physically impossible without the system breaking.

This does not account for energy lost through black body radiation which increases with temperature, but the equilibrium temperature is so high that even with these effects, it is likely not enough to keep the system at a reasonable temperature.

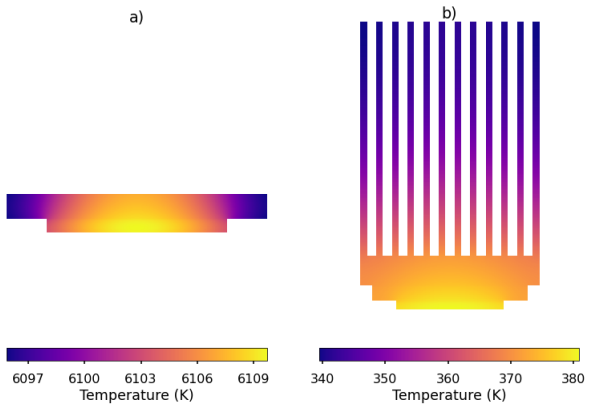

Fig. 5. a) is a heat map of the processor plus case system and b) is a heat map for a forced convection system with 12 30mm fins. The hottest points are in the middle of the processor and as you get further away from the heat source, the system becomes cooler.

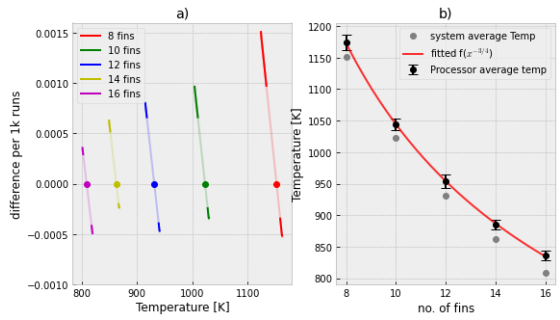

Fig. 6. a) shows after 20k iterations from above and below on different numbers of fins and the finding of the equilibrium value to simulate around next to get the data for b). b) shows the distribution of the system and respective processor temperatures with a fitted distribution (11) through the processor temperatures

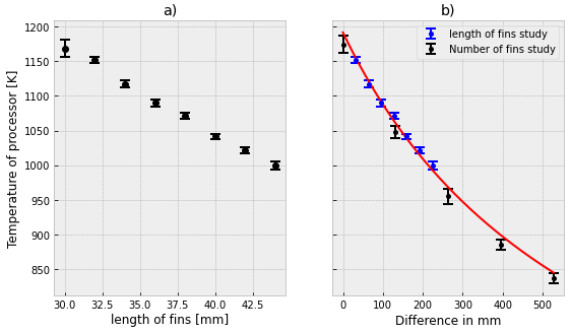

Fig. 7. a) is the data produced for the investigation of fins of differing lengths. b) shows the combined data for the length of fin and number of fin studies with the x-axis indicating the number of mm added to the perimeter from the baseline of 8 30mm fins with a spacing of 2mm. The data follows the same expected trend (11).

B. With heat sink

With the heat sink, the expected temperature of the system with natural convection is considerably lower as the fins allow for a much greater surface area to dissipate heat. An investigation was run into how changing aspects of the heat-sink, such as fin length, fin height and spacing between the fins affected the temperature of the system. All sims were run with scaling 6.

1) Number of fins investigation: It is expected that with an increasing number of fins, keeping the length and the spacing the same, with the increased surface area of the heat-sink the stable temperature would be reduced. The expected distribution given by (2) and (3), for the temperature of the body would be some $f(x^{-3/4})$. A function of the form (11),

$$f(x,a,b) = (\frac{a}{x})^{\frac{3}{4}} + b \tag{11}$$

was fitted to the simulated processor average temperatures as shown in Fig. 6. b). This fit implied that the distribution has an asymptote at $64 \pm 6°C$. Whilst this is a reasonable temperature for a CPU, it is to be at least 6% higher due to the scaling used and the dimensions are not appropriate for a computer. To get to $100°C$, there would have to be 10^6 fins.

2) Length of fins investigation: Fig. 7. highlights the combined data from the length of fins study and the number of fins study, in a plot that compares the temperature with the increase in perimeter of the system by either adding fins or lengthening them from a baseline of 8 fins of 30mm each. This trend again follows the expected distribution (11). The trend is the same as in the previous investigation and a reasonable temperature within the working range of the CPU is not achievable with natural convection alone in a system that would fit inside a desktop computer.

3) Fin separation investigation: Fin separation was tested on a system with 12 30mm fins and a spacing of 1-4mm with 1mm intervals. The difference between each simulation is between $925 \pm 20°K$ with the larger intervals being the coolest. The small increase in cooling efficiency may be explained by the system increasing in perimeter marginally with larger spacing. By increasing spacing, the efficiency of the area used to cool the CPU is decreased. The closer the fins are, the worse our assumption that the temperature of air around the system is a constant $20°$ as the air between the fins is being closely heated from both sides.

C. With heat sink and forced convection

The forced convective equation for $\vec{\phi}$ is linear concerning surface temperature therefore the expected distribution would be $f(x^{-1})$. This means the cooling gained per mm of perimeter added will decrease at a faster rate than with natural convective cooling.

1) Forced convection with heat sinc: Initially simulating a system with 1mm spacing and 12 fins of 30mm, an equilibrium temperature of $105 \pm 1°C$ was achieved with a scaling of 6. Due to the systematic error introduced by scaling, this value is likely higher and when run at a scaling of 10, the equilibrium temperature increases by 2%. The difference between processor temperature and the temperature of the rest of the system is also a bigger factor at lower temperatures, with the processor being up to 6% hotter than the system's average temperature up from 2.4% in the same system under natural convection.

2) Minimum dimensions for below $80°C$: The primary variable that determines the cooling ability of the system is the perimeter in contact with the air. After repeating the number of fins investigation with forced convection and fitting to a function inversely proportional to x, as well as accounting for systematic error due to scaling, it was determined that the minimum perimeter needed to keep the system below $80°$C is 1670$\pm$40mm. It is possible to set up equations to solve the set of solutions for fin length and an integer number of fins that give such a perimeter. This set of values can be used to determine which dimensions suit the needs of the CPU housing. The results for a desktop computer are 20 fins with a length of 39mm and spacing 1mm. This was tested and after adjusting for scaling, the result was $81 \pm 2°C$. These dimensions were chosen as it is close to a square and suitable for a desktop computer, but the more efficient use of the cross-sectional area is longer fin length and fewer fins.

D. Limitations

This study neglects the complexity of the heat generated by the processor as it is not uniform [**4**]. For the study with natural convection, a key assumption is that the ambient temperature around the system is constant at $20°C$ which at high temperature is a poor assumption, but this would only increase the stable temperature and so the conclusion that the system is not feasible still holds. The boundary between surfaces in these systems is also not perfect and thermal paste is often used to increase the efficiency of heat transfer across the boundaries. Imperfect heat transfer could be added to the model used by decreasing the effective conductivity at the boundary between materials. A simplification is that heat transfer due to forced convection is the same everywhere whereas fans are usually next to the heat sink and not directly next to the processor. It would be possible to implement separate functions for heat transfer at the separate boundaries. Due to the simplification made, the temperature will be higher than that modeled.

V. CONCLUSION

In the idealised system modeled, with only natural convection it was determined that it was not feasible to cool the system down to a desired temperature of $100°$C and in a conventional computer it reaches well above the melting point of some components, even with a small heat-sink. The primary variable for decreasing the temperature of the processor is the perimeter of the 2D model which refers to the surface area of the 3D system. This followed a trend proportional to $x^{-\frac{3}{4}}$ for natural convection and x^{-1}. These relationships were used to determine the minimum dimensions being a perimeter of 1670$\pm$40mm to be below $80°$C at max power output relating to a system such as 20, 39mm fins at 1mm spacing. This accounts for systematic error due to the discretisation of the system. The model overlooks many nuances with the actual physical system and is an idealised case and the actual working temperature of the system proposed would be higher.

REFERENCES

[**1**] Furber S. Microprocessors: the engines of the digital age. Proc Math Phys Eng Sci. 2017 Mar

[**2**] Samsung Begins Chip Production Using 3nm Process Technology With GAA Architecture" (Press release). Samsung. June 2022

[**3**] Evans, Lawrence C. Partial Differential Equations. Providence (RI): American Mathematical Society. 1998

[**4**] J. Zhang, "Accurate Power Density Map Estimation for Commercial Multi-Core Microprocessors," Publisher: IEEE, France, 2020