Software-Based On-Chip Thermal Sensor Calibration for DVFS-enabled Many-core Systems

Sami Teräväinen[1], Mohammad-Hashem Haghbayan[1], Amir-Mohammad Rahmani[1,2],
Pasi Liljeberg[1], and Hannu Tenhunen[1,2]
[1]Department of Information Technology, University of Turku, Turku, Finland
[2]Department of Industrial and Medical Electronics, KTH Royal Institute of Technology, Stockholm, Sweden
Email: {satate, mohhag, amirah, pakrli}@utu.fi, hannu@kth.se

***Abstract*—Due to increase in power density and temperature gradient in modern chips, multiple thermal sensors are deployed on the chip area to provide realtime temperature feedback for fine-grained dynamic thermal management (DTM) techniques. Thermal sensor accuracy is extremely prone to intra-die process variation and aging phenomena, and its report gradually drifts from the nominal value. This necessitates efficient calibration techniques to be applied before the sensor values are used. In addition, in modern many-core systems which are often enabled with dynamic voltage and frequency scaling (DVFS), thermal sensors located on cores are sensitive to the core's current voltage-frequency (VF) level, meaning that dedicated calibration is needed for each VF level. In this paper, we propose a general-purpose software-based auto-calibration strategy for thermal sensors without using any hardware infrastructures for DVFS-enabled many-core systems. We adopt a 2-point calibration method for calculating the calibration constants of each thermal sensor at each VF level. We demonstrate the efficiency of the proposed calibration strategy on a many-core platform, Intel's Single-chip Cloud Computer (SCC), covering all voltage and frequency combinations on the platform.**

Keywords*—*Many-Core Systems, Intel Single-chip Cloud Computer, Thermal Calibration.

I. Introduction

By increasing the number of transistors in a single chip, coupled with breakdown of Dennardian scaling and increasing the on-chip power density, temperature and power management is a necessity in the current and future technologies [1]. In addition, different activity rate of functional blocks, non-uniform workload variation, and advanced static and dynamic power management capabilities in recent CMPs result in non-uniform power distribution on the substrate which leads to significant temperature gradient [2]. Large temperature variation across a chip decreases the reliability of the circuits and degrades their performance [3]. Several research studies in the field of dynamic thermal management (DTM) aim at mitigating temperature and power violations at runtime in multi-/manycore systems [4]. An efficient DTM technique necessities accurate on-chip thermal sensors in recent technologies to maximize the performance under a restricted chip temperature. Localized sensors can provide critical information regarding the location of hotspots [5]. Today's multi-/many-core platforms are often equipped with multiple on-chip thermal sensors to monitor the chip's temperature in a fine-grained manner [6], [7], [8].

Due to process variation, on-chip thermal sensors may report temperature values which differs from the nominal values [9]. This can lead to both overestimation as well as underestimation of the real thermal status of the system. For example in [10], the authors show that un-calibrated thermal sensors for IBM25PPC750L processors deviate as much as 33°C and 48°C from their original temperature of 35°C and 95°C, respectively. Therefore, on-chip thermal sensors need to be calibrated initially before being used. However, the cost of infield calibration is too high which requires infrared camera and additional infrastructures [11]. Furthermore, due to device wear out, even though the sensors are well-calibrated before being used, their reports gradually drift away from actual temperature values which demands re-calibration at the time they are being used [11]. Thus, many commodity microchips prefer to use un-calibrated thermal sensors to be available for end-users [12].

There have been some studies on thermal sensor calibration using hardware [13] and software [14] techniques. In [15], the authors extract the relation between the actual temperature value and sensor outputs with the aim of probability distribution of the process variation. This process is the reverse form of regression. In [14], the authors propose a software-based strategy to statistically extract the thermal sensor characterization directly from the power and temperature measurements. They selected Intel's Single-Chip Cloud Computer (SCC) [16] as their platform to show the applicability of their approach. However, the mentioned work provides a calibration technique at a single voltage-frequency (VF) level making it inefficient to be used for modern DVFS-enables manycore systems as a sensor'value located on a core varies by changing the voltage level of each core [13]. For instance, our measurements illustrated in Figure 1 confirm the fluctuation of four different Intel SCC's thermal sensors (*S1* to *S4*) at different voltage levels when the actual temperature in all cases is fixed. As can be seen from the figure, the response drifts by changing the voltage of the region on which the sensor is located. Therefore, as the VF in recent chips dynamically changes at run-time for thermal or power management purposes, for each specific sensor a separate calibration is needed at each VF level. In [13], the authors propose a special fine-grained sensor design (called process-variation sensor) to bring high degree of immunity to the process variation and also the voltage. However, the proposed method demands extra on-chip hardware and is not applicable to every system.

In this paper, we propose an agile general-purpose

978-1-4799-8606-4/15/$31.00 ©2015 IEEE

Authorized licensed use limited to: University of Queensland. Downloaded on May 23,2020 at 12:43:18 UTC from IEEE Xplore. Restrictions apply.

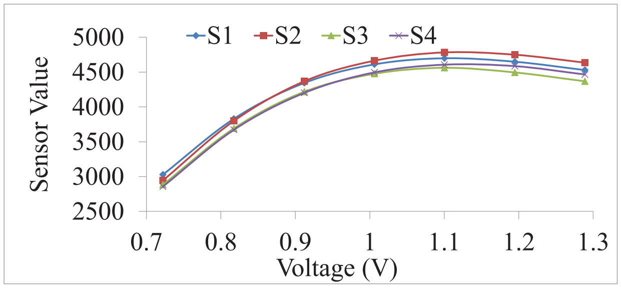


Fig. 1: Drift in sensor response due to changing the voltage level in a constant temperature demands calibration for different voltage levels

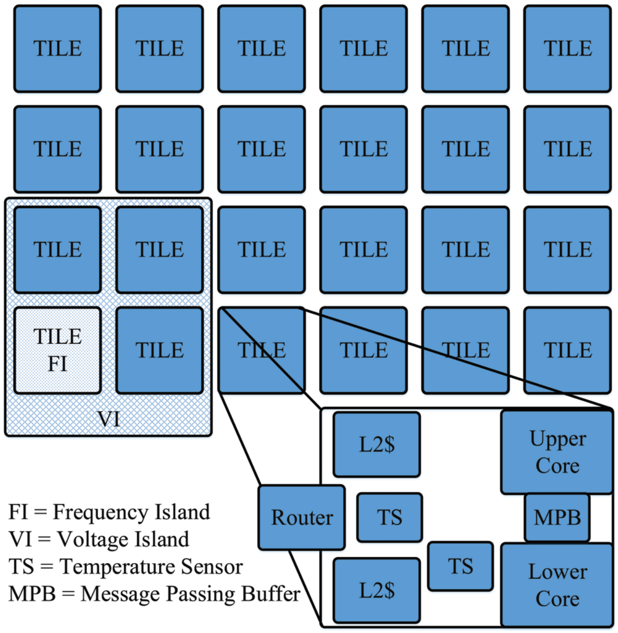


Fig. 2: Cores and Voltage and Frequency Islands

software-based strategy to calibrate multiple thermal sensors on a chip at different voltage/frequency levels. In our proposed method, we first eliminate the effect of VF level from sensor's value. After that, we use a 2-point calibration strategy to calibrate the thermal sensors. We use Intel SCC as a state-of-the-art many- core system to demonstrate our thermal calibration technique.

The remainder of this paper is organized as follows: Section II presents the state-of-the-art framework used in this paper. Section III discusses the proposed strategy for thermal sensor calibration for DVFS-enabled Systems. Section IV shows the efficiency of our method experimentally, and Section V concludes our work.

II. Intel SCC: A State-of-the-Art Many-core System

Intel's Single-chip Cloud Computer (SCC) is an experimental platform to study many-core CPUs [16]. It contains 48 cores placed on a 6×4 mesh network (Figure 2). The network has 24 tiles and each of them embeds two cores. Each tile is connected to the mesh network with a router. Cores'architecture are based on Intel P54C core which supports general x86 instruction set. By default every core has a light-weight Linux operating system running on them. Mesh network is divided into 6 voltage islands and 24 frequency islands. Table I shows all the possible voltage and frequency range of the cores. As it can be seen cores can operate from 0.7V to 1.3V and run on frequencies from 100MHz to 800MHz.

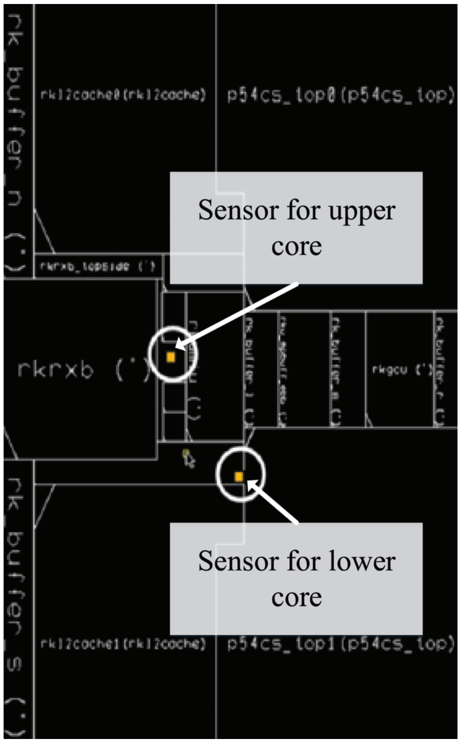


Fig. 3: Location of the two thermal sensors, one near the router and other one near lower core's L1 cache [17]

There are two thermal sensors placed on every tile: near the router and near one core's L1 cache as can be seen from Figure 3. Thermal sensors consist of two ring oscillators that count clocks over user defined time window T_{window}. Output of a thermal sensor is the difference in counts of these two oscillators that we address it as *counter value* (*CV*). The output value of the thermal sensor depends also on the T_{window} which decides for how long clocks are count by the oscillators. Using Equation 1, T_{window} is calculated for certain amount of clock cycles C_{cycles} on a certain frequency F. If the frequency F is changed on a tile, the amount of C_{cycles} needs to be calculated again to keep the T_{window} constant. C_{cycles} is the value that is written to a tile register which defines the number of clock cycles oscillators count.

$$T_{\text{window}} = \frac{C_{\text{cycles}}}{F} \tag{1}$$

Figure 4 shows the change of sensor's counter value by

Authorized licensed use limited to: University of Queensland. Downloaded on May 23,2020 at 12:43:18 UTC from IEEE Xplore. Restrictions apply.

TABLE I: Frequency and voltage levels for SCC

Tile Frequency (MHz)	Frequency Divider	Minimum Voltage (V)
800	2	1.1
533	3	0.8
400	4	0.8
320	5	0.7
266	6	0.7
228	7	0.7
200	8	0.7
178	9	0.7
160	10	0.7
145	11	0.7
133	12	0.7
123	13	0.7
114	14	0.7
106	15	0.7
100	16	0.7

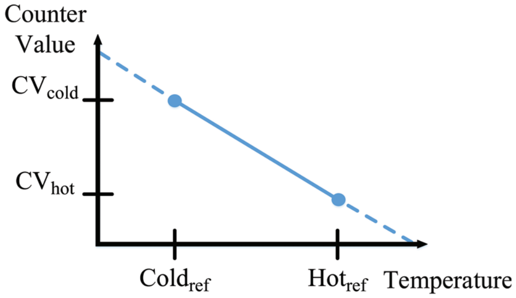


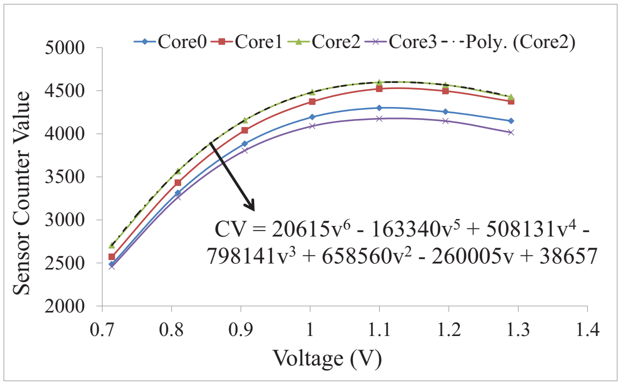
Fig. 4: Sensor behavior under different temperatures (Ambient temperature rises when cores generate heat)

changing the temperature [17]. As can be seen, the change of sensor's counter value is linear with the temperature change with a negative slope. As discussed before, for the same temperature, the values for thermal sensors are not the same as each other and a separate calibration is needed. SCC-platform has an off-chip ambient thermal sensor that shows temperature in Celsius scale. This ambient thermal sensor can be used as a reference for the temperature readings.

III. Calibrating the Thermal Sensors on SCC

When cores are running at the lowest frequency and are completely idle, the ambient temperature reaches its coldest point and then can be assumed that all the cores will share this temperature [17]. We address this temperature value as $Cold_{ref}$ and its corresponding counter value as CV_{cold}. When the cores are stressed, they get hotter and so does the ambient temperature. When the ambient temperature reaches its maximum, it can be used as a reference point for the maximum core temperature. We address this maximum temperature as Hot_{ref} and its corresponding counter value as CV_{hot}. Therefore, the slope of the graph shown in Figure 4 (i.e., s) can be calculated as follows:

$$s = \frac{Hot_{ref} - Cold_{ref}}{CV_{hot} - CV_{cold}} \tag{2}$$



CV = 20615v^6 - 163340v^5 + 508131v^4 - 798141v^3 + 658560v^2 - 260005v + 38657

Fig. 5: Sensor values on various cores under different voltages

The core temperature, T, can be calculated as follows:

$$T = s \times CV + bias \tag{3}$$

Where the constant $bias$ is calculated as follows:

$$bias = Cold_{ref} - CV_{cold} \times \left(\frac{Hot_{ref} - Cold_{ref}}{CV_{hot} - CV_{cold}}\right) \tag{4}$$

In Equation 3, the change in the sensors' values is converted to change in the actual temperature with the calibration value, then a base temperature $bias$ is added to this. However, before extracting the highest and the lowest values from the sensors, the effects that voltage causes to sensor report should be considered.

A. The Effect of Region's Voltage Level on Sensor's Response

To observe how region's voltage level afflicts the sensor reports, we cooled down the system and extracted the sensors' counter value in the minimum temperature, i.e., the *bias* value, for every voltage level between 0.7V and 1.3V with 0.1V step as shown in Figure 5. In Figure 5, for 0.8V supply voltage, the counter value for different sensors varies between 3200 and 3500, and for 1.1V, varies between 4100 and 4600 depending on the sensor's ID. It can be seen that the shape of the curves for sensors' behavior at different voltages for the same temperature are different from each other. Therefore, a separate process for each sensor is needed to eliminate the voltage level effect from sensor's response.

On the other hand, as the change of sensor's counter value is linear with the temperature change, each sensor replicates the *bias* values on an additional constant value while the temperature changes. Figure 6 shows a sensor's response for different temperatures versus different voltage levels. As can be seen from the figure, the trend of changing counter values for different VF levels is the same but with change of an overall constant. Thus, Equation 3 is valid in each separate voltage level and can be rewritten as follows:

$$T = s_v \times CV + bias_v \tag{5}$$

Authorized licensed use limited to: University of Queensland. Downloaded on May 23,2020 at 12:43:18 UTC from IEEE Xplore. Restrictions apply.

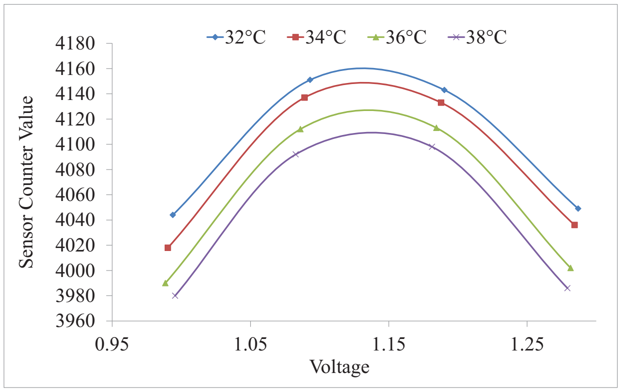


Fig. 6: Replication of sensor's response in different temperatures

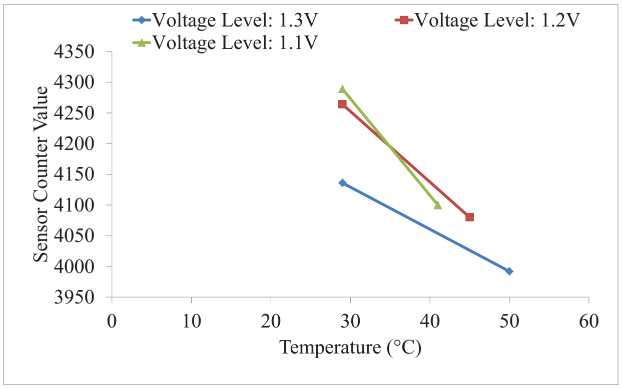


Fig. 7: Sensors temperature versus counter value for different voltage levels

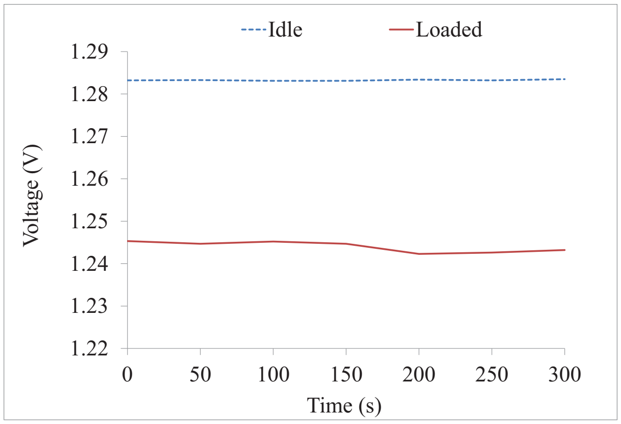


Fig. 8: Voltage drop when cores draw more power compared to the idle mode

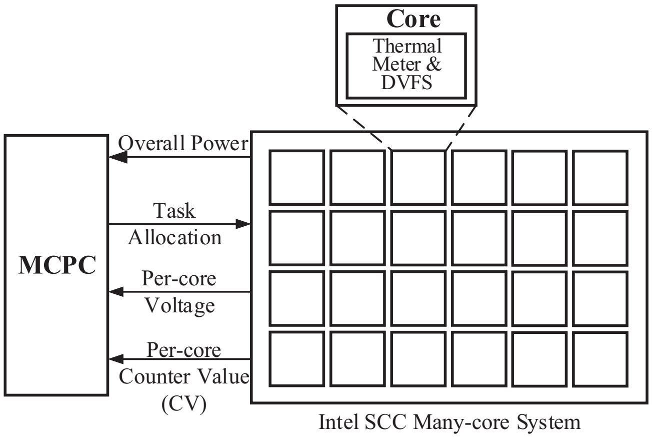


Fig. 9: The system architecture for thermal sensor reading

s_v and $bias_v$ are calculated as follows:

$$s_v = \frac{Hot_{ref} - Cold_{ref}}{CV_{hot}^v - CV_{cold}^v} \tag{6}$$

$$bias_v = Cold_{ref} - CV_{cold}^v \times (\frac{Hot_{ref} - Cold_{ref}}{CV_{hot}^v - CV_{cold}^v}) \tag{7}$$

Where CV_{cold}^v and CV_{hot}^v are the corresponding counter values at voltage v for $Cold_{ref}$ and Hot_{ref}, respectively. Figure 7 shows the graph of temperature versus counter value of a sensor for different voltage levels. As can be seen, the calibration constants for different voltages change while preserving the linearity in each voltage level.

As voltage level changes dynamically at runtime, the information of voltage level in each region is available at the system level to be used for calibration. However, even though SCC generally supports voltages from 0.7V to 1.3V with a step of 0.1V, these voltage levels are not totally fixed. Practically, the voltage region also changes due to workload of the system. When cores are loaded, the cores draw more power and generate heat, which result in voltage drops in voltage islands. Figure 8 shows the voltage of one Intel SCC's core at 533MHz frequency and 1.3V voltage settings. As shown in the figure, the voltage drops when the core is under stress compared to the situation when it is idle. Furthermore, there are some noises from supply voltage source that affects thermal sensors. Even though such noises on supply voltage is not so high, the thermal sensors are very sensitive to this noise. Thus, in order to calibrate thermal sensors that are under affliction of the voltage level, a runtime feedback from the voltage level of the region on which the sensor is located, is needed.

The system architecture for thermal sensor reading process in Intel SCC is presented in Figure 9. The idea is to show how temperature is calculated using the sensors' counters and the current voltage of regions. As can be seen from the figure, based on the calculated calibration values and the feedback of the current voltage of the region, actual temperature of the region can be calculated. It should be noted that in Intel SCC platform, the current voltage of each region is available in a local register and can be sent to the Management Console (MCPC) at runtime.

The other fact that can be concluded from the runtime

Authorized licensed use limited to: University of Queensland. Downloaded on May 23,2020 at 12:43:18 UTC from IEEE Xplore. Restrictions apply.

voltage change in different workloads is that the voltage is not discrete but continuous. Therefore, a continuous calibration function is needed for CV^v_{hot} and CV^v_{cold}. For this purpose, a simple polynomial regression program is used to extract CV^v_{cold} function from the sample values [18]. An estimated 6th degree polynomials curve for one core is shown in Figure 5. Using this function, the corresponding counter value in every voltage level can be calculated.

According to Equation 6 and Equation 7, calibrating the constants in each voltage level requires sensor report in minimum and maximum temperature for different voltage levels, i.e., CV^v_{cold} and CV^v_{hot}. However, as the rate of changing the counter value in terms of voltage at a certain temperature is the same as this rate at other temperatures, CV^v_{hot} can be calculated from CV^v_{cold} by measuring one sample of counter value in the maximum temperature and voltage level v_0, i.e., $CV^{v_0}_{hot}$, as follows:

$$CV^v_{hot} = CV^v_{cold} + (CV^{v_0}_{cold} - CV^{v_0}_{hot}) \tag{8}$$

Using Equation 8, with counter value observation in one voltage level for maximum temperature $CV^{v_0}_{hot}$, counter values for different voltage levels can be calculated for maximum temperature.

IV. Experimental Results

To demonstrate the efficiency of our calibration approach for DVFS-enabled many-core systems, we run the calibration code for 48 thermal sensors in Intel SCC platform. When cores are running with the lowest frequency and completely idle, the ambient temperature gets to its coldest point and then can be assumed that all the cores will share this temperature [17]. We consider this ambient temperature the minimum temperature, i.e., $Cold_{ref}$. To calculate the Hot_{ref}, we run a power virus called *cpuburn* on all the cores in the SCC platform [19] [17]. Running this software causes the core to consume the maximum power consumption which results in the maximum possible temperature, accordingly. We used the data from [17] for the temperature of the cores while running *cpuburn* at 0.8V supply voltage and 533MHz frequency. This value then is used in Equation 8 to extract CV^v_{hot} polynomial curve from CV^v_{cold}.

In our calibration process, the minimum difference between CV_{cold} and CV_{hot} for a single core is 273 unit which corresponds to 28°C temperature change. The coldest achieved ambient temperature $Cold_{ref}$ is 22°C and the maximum temperature Hot_{ref} is 50°C. Figure 10 shows how the coldest core's thermal sensor reacts to the temperature changes. When cores reached their maximum temperature, the overall sensor value dropped approximately by 300. Frequency and voltage of the cores were raised slowly, so we had a stable readings for temperatures between 30°C and 50°C with a step of 5°C in each voltage level for calibration. Now, the sensors have been calibrated for voltage changes.

It is impractical to compare the calibration method with other methods because non of the methods calibrate the thermal sensors at different voltage levels. Using simulators such as Hotspot for comparison is not also practical because the floorplan input and power distribution on the chip in Hotspot is not accurate comparing to the real SCC platform. Furthermore, as the sensitivity of thermal sensors is very high, even a marginal difference in configuration of the tool comparing to SCC framework causes inaccuracy in simulating the real thermal behavior of the system specially while the voltage of different regions on the chip is changing dynamically.

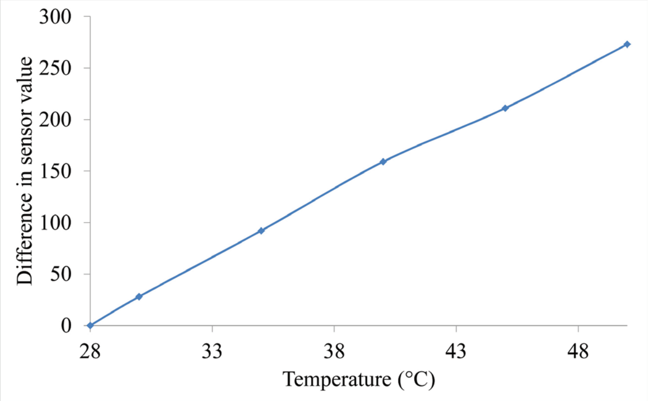


Fig. 10: Differences in Core 0's thermal sensors under various temperatures (Figure shows that thermal sensor works linearly)

We run three scenarios to see how our calibration method works on the SCC platform:

1) Scenario 1: Running *cpuburn* on all cores with 1.3V supply voltage and 800MHz frequency.
2) Scenario 2: Running *cpuburn* on 24 cores on the left side of the chip with 1.3V supply voltage and 800MHz frequency while 24 right-side cores are idle.
3) Scenario 3: Running *cpuburn* on all cores with 0.9V supply voltage and 320MHz frequency.

Fig. 11 shows the temperature distribution on the chip while running Scenario 1. It can be observed that, the temperature is higher in the middle cores whereas cores near to the edge of the die are colder as assumed. Distribution of the chip temperature while running Scenario 2 is shown in Fig. 12. In this case, the cores located at the left side of the chip are hotter than the idle cores located at the right side. In Scenario 3, all the cores are running the *cpuburn* virus with a different setup, where voltage was dropped to 0.9V and frequency to 320MHz. In this case, the cores are not too much stressed as can be seen from Fig. 13.

V. Conclusion

In this paper, we presented a method to calibrate thermal sensors for DVFS-enabled many-core systems. In our study, we used Intel's SCC as the demonstration platform. Thermal sensors are heavily affected by voltage changes dynamically at system level or by voltage supply noise. To calibrate the sensors in different voltage levels, we studied how each thermal sensor acts at different voltage levels. Another issue is converting the sensors readings to a proper temperature scale such as Celsius. To calibrate the sensors, we used the maximum and minimum temperature values as a reference for sensor readings.

Authorized licensed use limited to: University of Queensland. Downloaded on May 23,2020 at 12:43:18 UTC from IEEE Xplore. Restrictions apply.

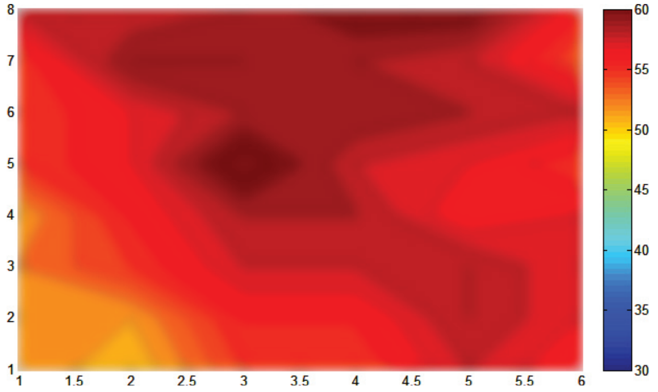


Fig. 11: Cpuburn on all cores. Ambient temperature = 50°C

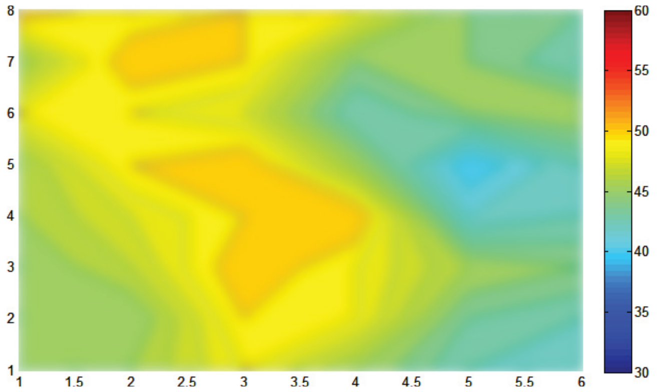


Fig. 12: Cpuburn on cores 0-3, 12-15, 24-27, and 36-39. Ambient temperature = 42°C

Acknowledgment

The authors acknowledge the financial support by the Academy of Finland project entitled "MANAGE: Data Management of 3D Systems for the Dark Silicon Age", University of Turku graduate school (UTUGS), EU COST Actions IC1103: Manufacturable and Dependable Multicore Architectures at Nanoscale (MEDIAN) and IC1202: Timing Analysis on Code-Level (TACLe).

References

[1] W. Lee, Y. Wang, and M. Pedram. Vrcon: Dynamic reconfiguration of voltage regulators in a multicore platform. In *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pages 1–6, 2014.

[2] A.H. Ajami, K. Banerjee, and M. Pedram. Analysis of substrate thermal gradient effects on optimal buffer insertion. In *IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pages 44–48, 2001.

[3] A.K. Coskun, T.S. Rosing, K.A. Whisnant, and K.C. Gross. Static and dynamic temperature-aware scheduling for multiprocessor socs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, (9):1127–1140, 2008.

[4] A.-M. Rahmani, M.-H. Haghbayan, A. Kanduri, A.Y. Weldezion, P. Liljeberg, J. Plosila, A. Jantsch, and H. Tenhunen. Dynamic power management for many-core platforms in the dark silicon era: A multi-objective control approach. In *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2015.

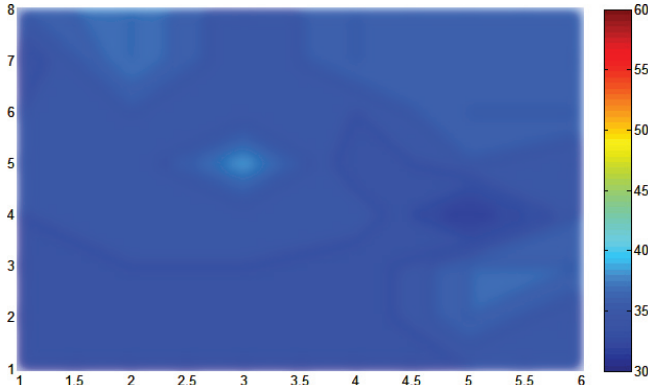


Fig. 13: Cpuburn on all cores, low voltage and frequency. Ambient temperature = 32°C

[5] J. Lee, K. Skadron, and S. Chung. Predictive temperature-aware dvfs. *IEEE Transactions on Computers*, 59(1):127–133, 2010.

[6] M. Sasaki, M. Ikeda, and K. Asada. -1/+0.8 deg;c error, accurate temperature sensor using 90nm 1v cmos for on-line thermal monitoring of vlsi circuits. In *IEEE International Conference on Microelectronic Test Structures*, pages 9–12, 2006.

[7] D. Pham, S. Asano, M. Bolliger, M.N. Day, H.P. Hofstee, C. Johns, J. Kahle, A. Kameyama, J. Keaty, Y. Masubuchi, M. Riley, D. Shippy, D. Stasiak, M. Suzuoki, M. Wang, J. Warnock, S. Weitzel, D. Wendel, T. Yamazaki, and K. Yazawa. The design and implementation of a first-generation cell processor. In *IEEE International Solid-State Circuits Conference (ISSCC)*, pages 184–592 Vol. 1, 2005.

[8] C. Poirier, R. McGowen, C. Bostak, and S. Naffziger. "power and temperature control on a 90nm itanium reg-family processor". In *IEEE International Solid-State Circuits Conference (ISSCC)*, pages 304–305 Vol. 1, 2005.

[9] A. Bartolini, M. Cacciari, A. Tilli, and L. Benini. A distributed and self-calibrating model-predictive controller for energy and thermal management of high-performance multicores. In *Design, Automation and Test in Europe Conference (DATE)*, pages 1–6, 2011.

[10] S. Remarsu and S. Kundu. On process variation tolerant low cost thermal sensor design in 32nm cmos technology. In *ACM Great Lakes Symp*, page 487Ű492, 2009.

[11] S. Remarsu and S. Kundu. On process variation tolerant low cost thermal sensor design in 32nm cmos technology. In *Proceedings of the 19th ACM Great Lakes Symposium on VLSI (GLSVLSI)*, pages 487–492, 2009.

[12] "revision guide for amd npt family 0fh processor". In *AMD Publication #33610*, page 37, 2006.

[13] B. Datta and W. Burleson. Calibration of on-chip thermal sensors using process monitoring circuits. In *11th International Symposium on Quality Electronic Design (ISQED)*, pages 461–467, 2010.

[14] A. Bartolini, M. Sadri, F. Beneventi, M. Cacciari, A. Tilli, and L. Benini. A system level approach to multi-core thermal sensors calibration. In *Power and Timing Modeling, Optimization, and Simulation of Integrated Circuit and System Design*, pages 22–31, 2011.

[15] Y. Zhang and A. Srivastava. "accurate temperature estimation using noisy thermal sensors for gaussian and non-gaussian cases". *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pages 1617–1626, 2011.

[16] Intel. SCC external architecture specification. In *Intel, Tech. Rep.*, 2010.

[17] Intel Labs. Using the sensor registers. In *Revision 1.1*, 2010.

[18] L. Magee. Nonlocal behavior in polynomial regressions. *The American Statistician*, pages 20–22, 1998.

[19] Cpu burn-in homepage. http://www.cpuburnin.com. Accessed: 2015-05-18.

Authorized licensed use limited to: University of Queensland. Downloaded on May 23,2020 at 12:43:18 UTC from IEEE Xplore. Restrictions apply.