January 2015

Fine-grained Energy and Thermal Management using Real-time Power Sensors

Srikar Bhagavatula
Purdue University

Graduate School Form 30
(Revised 08/14)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Srikar Bhagavatula

Entitled
Fine-grained Energy and Thermal Management using Real-time Power Sensors

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

BYUNGHOO JUNG

DIMITRIOS PEROULIS

KAUSHIK ROY

WILLIAM J. CHAPPELL

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification/Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): BYUNGHOO JUNG

Approved by: Michael R. Melloch 04/29/2015

Head of the Department Graduate Program Date

FINE-GRAINED ENERGY AND THERMAL MANAGEMENT USING REAL-TIME POWER SENSORS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Srikar Bhagavatula

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2015

Purdue University

West Lafayette, Indiana

To the giants on whose shoulders we stand today, to see farther than they ever did.

ACKNOWLEDGMENTS

I would like to thank Prof. Byunghoo Jung for all his help and guidance during my time here at Purdue and for giving me the leeway to make mistakes and learn from them. I am extremely grateful to Prof. Kaushik Roy, Prof. Dimitrios Peroulis and Prof. William Chappell for being part of my advisory committee and for their invaluable inputs in shaping my dissertation. I would like to thank my parents for instilling in me, the importance of patience and perseverance. I would like to thank my friends, who at various points in my life served as pillars to lean on. I am extremely thankful for the opportunities that were provided to me, and am mindful of the many unfortunate ones who would have deserved such opportunities far more than I ever did. I have had many engaging and thoughtful discussions with my fellow students: Wu-Hsin Chen, WingFai Loke, Jangjoon Lee, Jaehyuk Jung, Mohammed Abu Khater, El-kim Roa, Serkan Sayilir, Julia Lu and Mohit Singh which helped shape my research. I would also like to thank all the sports teams (cricket at Purdue, IIT Bombay and other places) that I was part of, for teaching me invaluable lessons in teamwork and leadership. And finally, I would like to thank "Tank" and "Esme", the two canines in my pack, who taught me the meaning of unconditional trust and loyalty, and forced my chaotic schedule to come to order.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

ABBREVIATIONS

ADC	Analog-to-Digital Converter
AVFS	Adaptive Voltage and Frequency Scaling
DAC	Digital-to-Analog Converter
DPM	Dynamic Power Management
DFS	Dynamic Frequency Scaling
DTM	Dynamic Thermal Management
DVS	Dynamic Voltage Scaling
DVFS	Dynamic Voltage and Frequency Scaling
PVT	Process, Voltage and Temperature
PMU	Power Management Unit
TDC	Time-to-Digital Converter
HPC	Hardware Performance Counter
IPC	Instructions Per Cycle
CPI	Cycles per Instruction
HCI	Hot Carrier Injection
HPC	Hardware Performance Counters
NBTI	Negative Bias Temperature Instability
TDDB	Time dependent Dielectric Breakdown
EM	Electro Migration
TC	Thermal Cycling
CPU	Central Processing Unit

ABSTRACT

Bhagavatula, Srikar Ph.D., Purdue University, May 2015. Fine-grained Energy and Thermal Management using Real-time Power Sensors . Major Professor: Byunghoo Jung.

With extensive use of battery powered devices such as smartphones, laptops and tablets energy efficiency has become a critical design criterion in today's System on Chip (SoC) designs. Although shrinking device sizes helped to lower production costs and enabled faster computing, they also resulted in continued rise in power densities. As a result, significant new challenges have appeared in system reliability (due to thermal failures) and feasibility (due to cooling costs).

Techniques such as Dynamic Voltage and Frequency Scaling (DVFS), activity migration, power gating, clock gating and fetch toggling have been proposed to reduce power densities and increase energy efficiency. Such techniques require real-time information such as workload, temperature, power etc. for which thermal sensors and hardware performance counters are deployed.

However, temperature sensors have slow response times and cannot reliably predict future workloads without resorting to computationally intensive algorithms. Hardware performance counters on the other hand, are only proxy measures of dynamic power and cannot account for static power and variations in ambient conditions.

In this dissertation, novel sensors for concurrent and fast estimation of power and temperature, with simple calibration schemes for improved accuracy have been proposed. Occupying less than 0.01mm^2 on-chip area, these sensors consume less than 200μW and provide fast response within 100ns, which is a significant advancement of state-of-the-art in sensors. This sensors is then deployed in multi-core environments

employing DVFS and activity migration to evaluate, and quantify their performance vis-a-vis Hardware Performance counters and temperature sensors.

1. INTRODUCTION

1.1 Energy and Thermal management

Reducing power consumption is one of the most important design goals for electronics. Between 2005 and 2010, electricity consumed by data centers alone grew 36% in USA and 56% worldwide. Computing nodes consume most of this energy in the datacenters, and amongst computing nodes, CPU is the biggest consumer of energy [1]. Rapid growth in the use of mobile platforms for social networking and high definition media sharing has put greater demands on the performance of mobile processors used in devices such as tablets, smartphones and laptops at far higher energy efficiencies Table. 1.1. The quest for longer up-times of such devices has seen significant efforts spent into reduction of power consumption. At the same time, as some of the computing is off-shored to remote servers, cloud storage and data centers also face an ever increasing load.

Higher power dissipations in these servers result in higher temperatures, mandating more aggressive cooling solutions. Power (including cooling costs) has often been cited as the largest contributor to expenditures in the maintenance of data center farms [2]. Therefore, power has emerged as the dominant design criterion, and it is critical to reduce power consumption [3]. [4] suggests that even though efforts at reducing power consumption have been moderately successful, continued increase in device densities resulted in rising power densities and chip temperatures.

Table 1.1: Evolution of Intel® cpus from 2005 to present

	2-core Xeon	Pentium 780M	i5-4300U	Xeon E5-2695v3
Release Date	2005	2005	2013	2014
Core Frequency	3GHz	2.2GHz	1.9-2.9GHz	2.3-3.3 GHz
Technology	90nm	90nm	22nm	22nm
Die Size	$162mm^2$	$87mm^2$	$181mm^2$	$662mm^2$
Cores	2	1	2	14
Transistor count	338 million	144 million	1.3 billion	5.7 billion
Passmark	777	502	3757	21123
TDP	135W	34W	15W	120W
Deployment	Server	Mobile- Laptop	Mobile- Tablet	Server

1.1.1 Reliability

With rising temperatures, more design effort is expended to meet performance goals at higher temperatures. At the same time, rising chip temperatures also offer a serious challenge to system reliability through the following mechanisms

ElectroMigration - Deformation of metal interconnects leading to shorts and disconnects as a result of the transfer of momentum from electrons to the lattice [5]. The mean time to failure (MTTF) for this mechanism is given by

$$MTTF_{EM} = \frac{A_{EM}}{(J - J_c)^n} e^{\frac{E_a}{k_B T}} \tag{1.1}$$

where A_{EM} is an empirical constant, J is current density in the interconnect, J_c is the threshold current density for failure, k_B is the Boltzmann's constant, E_a is the activation energy for electromigration and T is the temperature.

Time dependent Dielectric breakdown - is a wear-out mechanism of the gate dielectric due to electric field and temperature which results in formation of conductive paths through dieletrics [6]. MTTF due to TDDB is described as

$$MTTF_{TDDB} = A_{TDDB} e^{-\gamma E_{ox}} e^{\frac{E_b}{k_B T}} \tag{1.2}$$

where $\gamma$is a field acceleration parameter, E_{ox} is the electric field across the dielectric, A_{TDDB} is an empirical constant and E_b is the activation energy for TDDB.

Thermal cycling - large temporal variations in temperature on a given spatial location are known as thermal cycles. Such cycling can result in plastic deformation that accumulate over time leading to fatigue, cracks, fractures, shorts and other failures between metal and dielectrics [7]. Expected number of thermal cycles to failure is given by

$$N_f = C_o[C_1(T_{max} - T_{min}) - C_2(T_{avg,s} - T_{mold})]^{-q} \tag{1.3}$$

where T_{mold} is the molding temperature of the package process, T_{max}-T_{min} is the amplitude of the thermal cycles, $T_{avg,s}$ is the average temperature. Earlier only the cycles arising out of switching between sleep and active states were considered large enough to result in failure [8], However, with shrinking device sizes, even run-time charactersitics of the workloads can result in large temporal variations in power densities and therefore, thermal cycles with large amplitudes.

The individual and cumulative effects of temperature on failure rate are formulated as the Arrhenius equation

$$T_f = A \cdot e^{\frac{E_A}{k_B \cdot T}} \tag{1.4}$$

where A is an empirical constant, E_A is the activation energy for cumulative stress mechanisms.

On the other hand, leakage current in semiconductor systems can be formulated as [9], [10]

$$I_{leak} \propto T^2 \cdot e^{\frac{\alpha V_{dd}}{T}} \tag{1.5}$$

At lower technology nodes, as the leakage power becomes comparable to the dynamic power, it can contribute to significant increase in chip temperatures, which leads to

a further increase in the leakage current. This phenomenon is known as thermal runaway which can be catastrophic.

Hence, it is imperative that thermal management aims to reduce hotspots, manage spatial skews in temperature and to reduce both the frequency and the amplitude of thermal cycling. Although thermal and energy management policies may, at times, result in different localized directives, reducing the overall power consumption is a common end-goal.

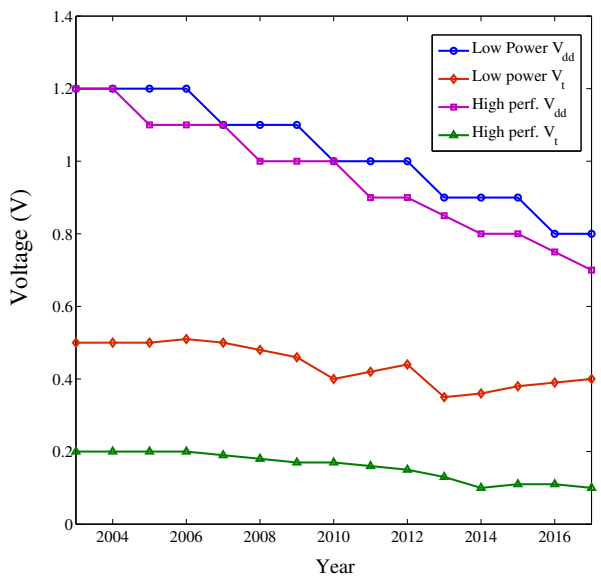


Fig. 1.1.: Projected Supply and threshold voltage scaling (ITRS 2007) [11].

1.2 Power Reduction Techniques

Power consumption in digital circuits can be modelled as [13]

$$P = C \cdot V_{dd}^2 \cdot f + \hat{I}_{sc} \cdot V_{dd} + I_{leak} \cdot V_{dd} \tag{1.6}$$

Improvements in power consumption, therefore, target one of the various parameters appearing in this equation - power supply (V_{dd}), frequency of operation (f), device capacitance (C), leakage currents (I_{leak}), or the average short circuit current ($\hat{I}_{sc}$).

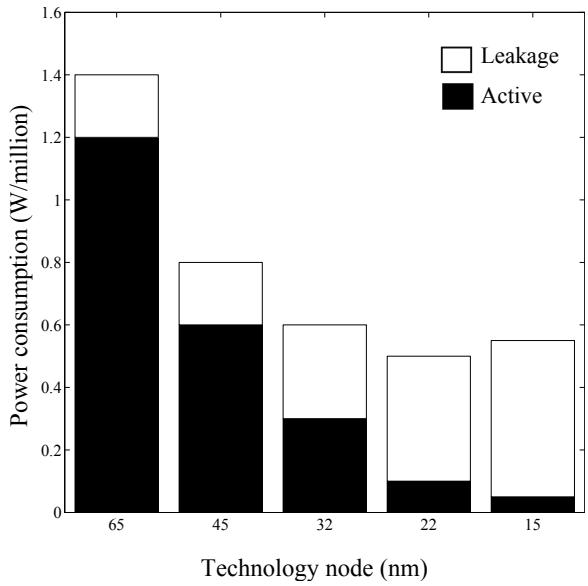
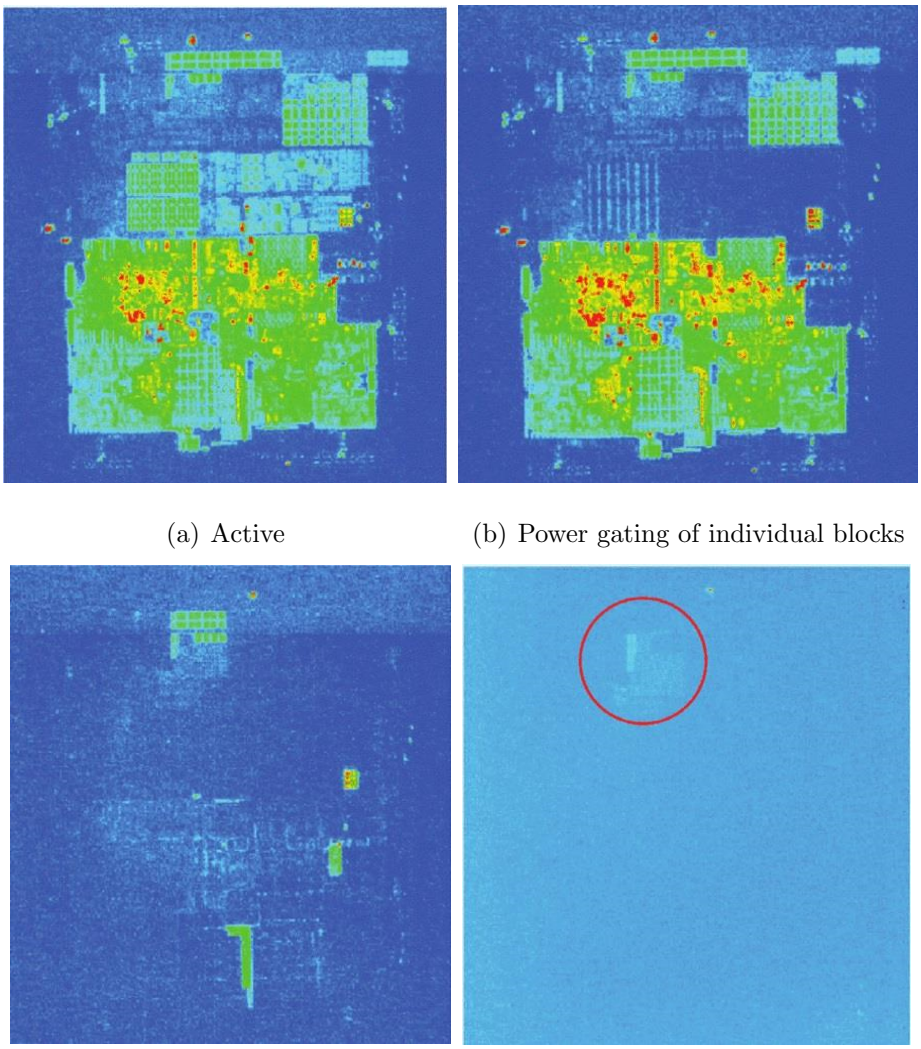


Fig. 1.2.: Trend of leakage and active power in scaled technologies [12].

Device scaling in semiconductor technologies allowed us to reduce power consumption by reducing parasitic capacitances. However, V_{dd} and switching frequency f directly trade-off with performance. Hence, reducing either would result in reduced power consumption only at the cost of performance. Architectural improvements, better pipeling, parallelization etc. add design complexity, but make design at lower frequencies and supply voltages possible without reducing performance. Threshold voltage scaling allows design at lower V_{dd} by increasing overdrive. However, increasing leakage power in sub-90nm technologies (Fig. 1.2) has curtailed Vt scaling resulting in reduced overdrives (Fig. 1.1). Multiple-Vt CMOS devices and identification of critical paths to use low-Vt transistors to improve performance, while using high-Vt transistors to reduce leakage elsewhere are some strategies to counter these issues. Similarly, some system-level power management techniques have evolved to counter these issues and lower power consumption at reduced loss in performance.

Power Gating is a methodology to reconfigure the system on-the-fly and reduce the static current being leaked by idle blocks. This technique relies on sleep transistors i.e very large MOSFETs operating in linear region between the supply rails and the



(a) Active (b) Power gating of individual blocks

(c) Gated clock tree (d) Full chip on stand-by

Fig. 1.3.: Sleep states [14].

circuit blocks, essentially creating virtual power supply nets. However, once power is gated to a circuit block, power-up requires finite time. This is referred to as "wake-up latency". Wake-up latency often trades-off directly with leakage savings. Multiple sleep-states with varying degrees of latencies are available in today's microprocessors [15] (Fig. 1.3).

As switching power is proportional to V^2, reduction of power supply results in quadratic savings in power and cubic reductions in power densities [16], [17]. Reduc-

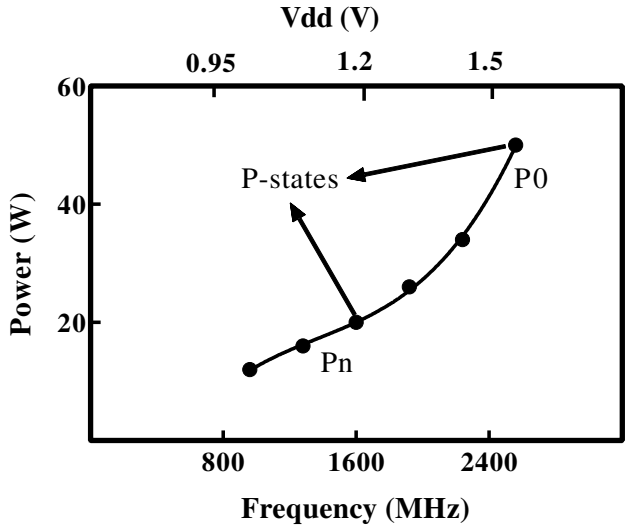


Fig. 1.4.: Dynamic Voltage and Frequency scaling

tion of voltage supply in run-time, known as Dynamic Voltage scaling was proposed as a method for thermal management [7]. In synchronous circuit designs, clock gating is a popular method to reduce dynamic power consumption. Clock to inactive logic circuits is turned off to save on switching power consumed in clock trees [18]. When stopping the entire clock tree is not feasible, fetch gating is used to prevent instruction activity through the pipeline or a more fine-grained version known as "Local toggling" is used in some low power states [19]. On the other hand, Dynamic Frequency Scaling (DFS) tunes the clock frequency according to workload, ensuring lower power consumption at lighter workloads [20]. Circuits can operate with lower supply voltages at slower switching frequencies. Taking advantage of lighter workloads, speed scaling is combined with voltage supply reduction, to obtain cubic power reduction in Dynamic Voltage and Frequency Scaling [21] that either of DVS or DFS alone cannot attain Fig. 1.4.

With the advent of multi-core and many-core processors, activity or thread migration has become one of the most important techniques to manage temperature and workload [22] (Fig. 1.5). Asymmetric microprocessors have introduced different

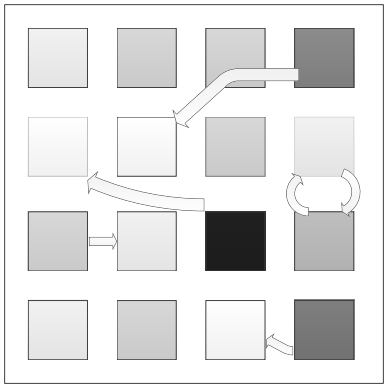


Fig. 1.5.: Activity migration

types of cores, where some are inherently more efficient at execution of a particular class of tasks (e.g float calculations). In such environments, activity migration is also an important tool for reducing the system energy consumption [23].

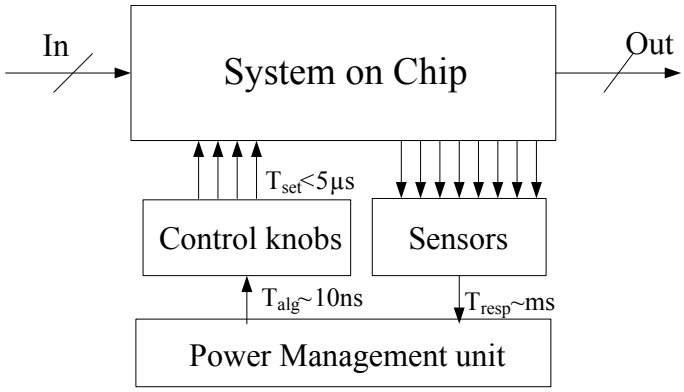


Fig. 1.6.: System-level techniques for reducing power consumption.

Such techniques depend on real-time feedback from sensors for temperature, power and workload data, and tune the system accordingly (Fig. 1.7). As a result, significant improvements in yield can be obtained with a much lower design effort [24]. Control knobs used to tune system performance in real-time include frequency synthesizers, voltage regulators, multiple cores etc. On-chip regulators [25] and Phase-Locked Loops [26] can achieve transition times in the order of tens of nanoseconds. Copying buffers for fine-grained activity migration can also be achieved in tens of nanoseconds

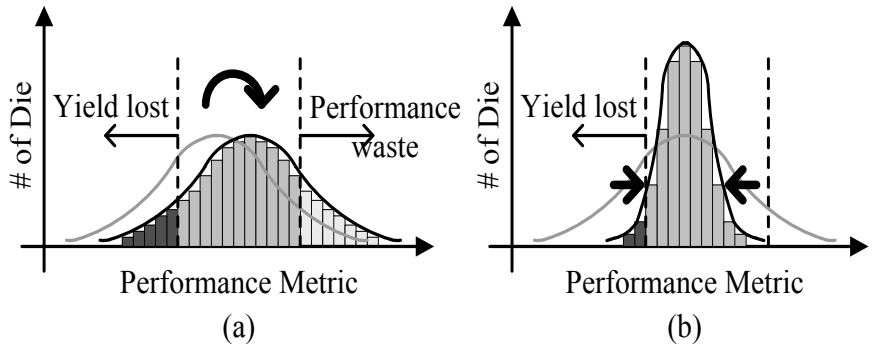


Fig. 1.7.: Online calibration for better yield [24].

[23], and today's scheduling algorithms converge in a few hundreds of cycles. However, even state-of-the-art sensors suffer from slow reponse times in the order of tens of microsecond to a few millisecond. Therefore, this dissertation focuses on designing better sensors with higher accuracy and faster response times.

2. CHALLENGES IN ESTIMATION OF POWER

Some scoping techniques that employ a separate voltage supply for monitors to sample power consumption have been presented [27]. Current supplied through the primary supply voltage is sampled at the rate of a few kHz and estimated accordingly. However, in most microprocessors and digital circuits, the clock rate can vary from hundreds of MHz to a few GHz. A sampling rate of kHz cannot ensure that important events such as occasional spikes in power consumption are captured. On the other hand, increasing the sampling rate to capture such events, results in significant power overheads. Similarly, the need for a separate voltage supply results in area overhead and the inability to obtain block-level power estimates. An on-chip current sensor has also been developed for battery management [28]. In this sensor, a small resistance is introduced in the path of the supply current and the voltage drop across this resistance is sampled by an ADC at the rate of 200S/s. In addition to the slow response time of this current sensor, it also occupies significant area (1mm^2) and therefore cannot be replicated for block-level, fine-grained power management.

Significant efforts have been devoted to modeling power consumption using simulated circuit models. These models can be generated with varying level of detail which trade-off accuracy with computational overhead. Hardware Performance Counters (HPCs) are sets of registers built into the microprocessor to count performance events such as Instructions exected per cycle (IPC), data dependencies, Instruction Cache Misses and Translation lookaside Buffer (TLB) misses. These are fit into linearized, architecture-dependent power models [29]. As various design options can be evaluated without building real hardware, modeling is an extremely useful tool. However, modeling and simulation also have one major drawback: the power consumption from modeling or simulation must be validated using experimental data. In other words, modeling and simulation cannot replace measurement. Estimation

accuracy also depends on the choice of counters, representative benchmarks used to formulate relationships between HPCs and actual power consumption. Moreover, power consumption may vary widely with ambient operating conditions like the supply voltage or on-chip temperature which cannot be expected to remain constant. Hence, simulation data may not display one-to-one correlation with each individual chip at every given ambient operating condition, resulting in the need, once again, for real measurements. Moreover, these HPCs are only accurate in estimating power averaged over 10,000 or more cycles and the errors in estimating dynamic power consumption can be as high as 40% [30], [31]. In addition, on-chip temperatures can cause significant errors in the estimated power values and often an on-chip temperature sensor becomes an essential foil for these performance counters.

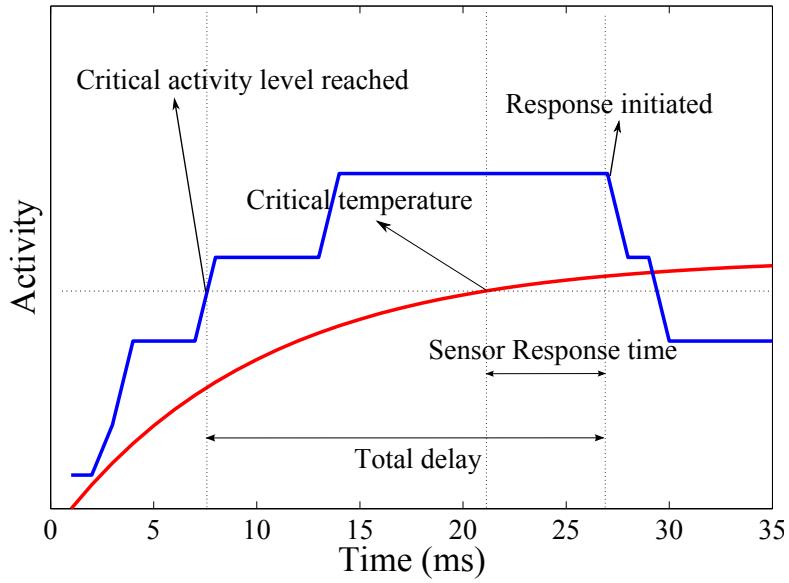


Fig. 2.1.: Inherent lag in temperature sensors based thermal management.

Thermal sensors have widely been deployed in high-performance processors [32]-[33]. But thermal sensors cannot replace power sensors for the following reasons:

1) Rising temperatures are the consequences of power consumption with significant delay (in the order of few milliseconds). Thus, by the time temperatures

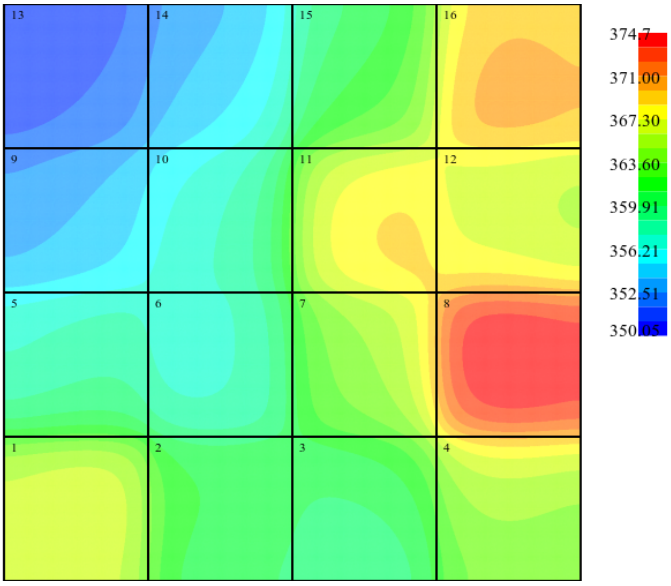


Fig. 2.2.: Diffusion of heat along surface in a multi-core environment.

rise, excessive power has already been consumed. In other words, temperature sensors have poor temporal resolution (Fig. 2.1).

2) Temperatures at thermal sensors depend on ambient temperatures and are affected by cooling. Aggressive cooling may keep sensed temperatures low even though power consumption is high. Similarly, at cold ambient conditions, high power consumption may not trigger power management functions whereas at higher ambient temperatures, it may be triggered earlier.

3) A power-reduction technique may be applied to a subsystem which is sufficiently far away from any thermal sensors and thus show no reduction of temperatures. Similarly, if two blocks are placed close together, heat dissipated from one block may be indistinguishable from the heat dissipated by the other (Fig. 2.2). Hence, thermal sensors have poor spatial resolution.

4) Estimation of actual power consumption from temperature values also suffers from errors due to variable thermal resistances in CMOS processes. As described in [34], [35], conversion of temperature to power is resource intensive; yet, power

based dynamic optimization (even when the estimates come from temperature sensors) outperforms temperature based management [36].

These reasons call for the development of built-in power sensors. However, measuring power consumption is challenging. The challenges can be classified into two categories: overhead and accuracy. To begin with, the Heisenberg Effect: it is impossible to measure anything without perturbing the system being measured. This is because the measurement circuits must consume additional power in order to measure power. We must ensure that the circuits for measurement consume little power compared with the system whose power is being measured. The measurement circuit must not become a hotspot and trigger thermal events that can degrade performance. The measurement circuit must also occupy negligible area. Accuracy is another type of challenges. As power management techniques (such as power gating) are widely adopted, the power consumption of a subsystem can change multiple times within a microsecond. As a result, the measurement circuit must have fast response times. Moreover, as devices become smaller, process variations become a major concern. The measurement circuits must also be able to self-calibrate. Table 2.1 summarizes the challenges. Due to these factors, few successful studies showing how to create low-overhead high-accuracy power sensors have been reported and none of the proposed solutions were integrated on to a single chip thus far.

Table 2.1: Challenges in the design of built-in power sensors

Requirements	Challenges
Low overhead	Power consumption of the sensor circuits
	Area occupied by the sensors
	Performance degradation due to measurement
	Thermal effects due to the sensors
High accuracy	Fast response time to detect rapidly changing power consumption
	Process variation tolerance
	Tolerance to ambient conditions - temperature, noise, power supply

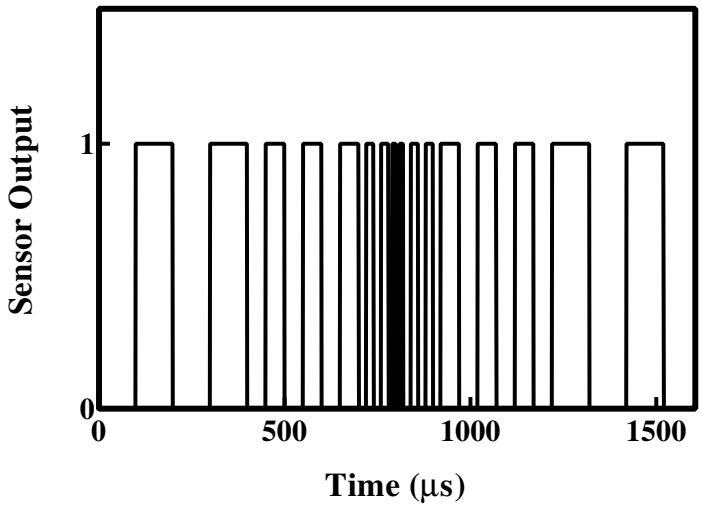
3. IDEA OF A POWER SENSOR

Direct estimation of current requires either addition of a shunt resistance [28], [27] or an expensive hall sensor [37]. As hall sensors have not been fully integrated onto a silicon system, addition of a shunt resistance appears to be the only choice. However, any additional resistance in the power path perturbs the system and changes the power delivered to the system. To minimize this perturbance, there is a limit to the amount and number of shunt resistances that can be added. As a result, block-level power estimation for fine-grained power management becomes difficult.

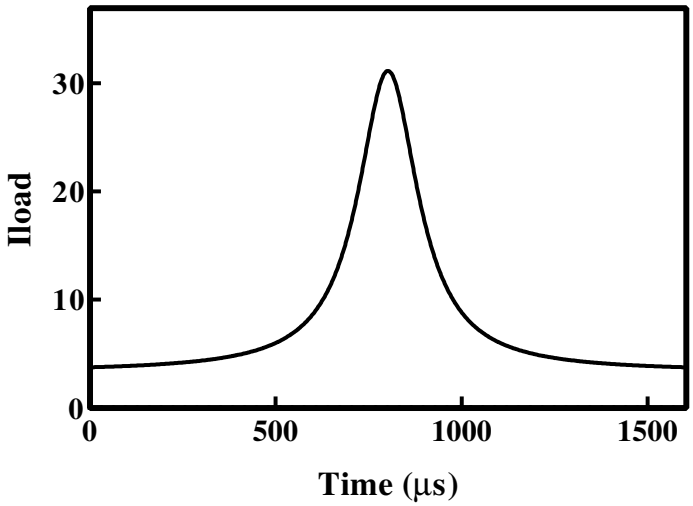
Due to threshold voltage scaling, power gating has become a ubiquitous design choice for digital, mixed signal and increasingly even in low-power analog designs [38]. As the sleep transistors operate in linear region during ON-state, they can be treated as the shunt resistance required to obtain information regarding the load currents. Moreover, the sleep transistor's resistance is already part of the system. Hence, sampling the IR-drop to estimate power gives an inherently more accurate estimate of power delivered to a system.

As sleep transistors are always sized to ensure that the virtual supply is within tens of mV of the real supply rails, input dynamic range of such a sensor remains the same across all levels of hierarchy. This enables easier replication of the sensor with minimal redesign effort for fine-grained, block-wise power management.

Sensed voltage can be converted to a digital count using just an ADC for power management. However, using an ADC would require a very resolution resulting in a high power and area overhead. In addition, fast response times would require high sampling rates in ADC resulting in increased power consumption in the sensor. In order to overcome these challenges, the voltage signal is converted to time domain and the signal can be interfaced with a power management unit using a Time-to-Digital Converter (TDC) like a pulse counter.



(a)

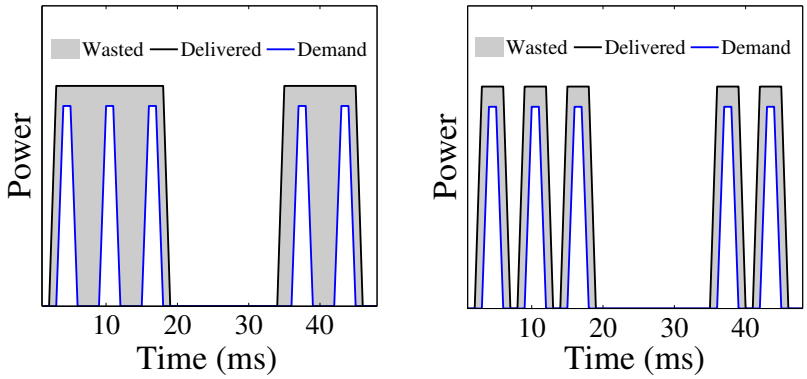


(b)

Fig. 3.1.: Desired output of the sensor with changing current.

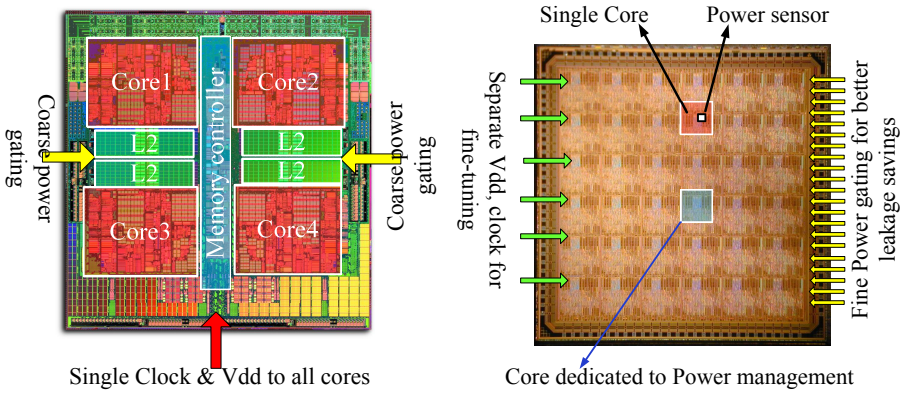
Using an ADC would also result in a constant response time system unless the sampling rate is dynamic, which would increase the overhead and complexity significantly. However, activity level in microprocessors and other SoCs is characterised

by shorter durations of intense activity (peak power) and longer durations of low activity (idle) periods. Therefore, battery powered systems (which have finite energy to supply) see greater depletion in energy resources in times of peak activities and to ensure continued uptime, faster response times are needed in times of these activities whereas slower response times can be tolerated in times of low activity. Therefore, the sensor is designed so that output pulse-rate or frequency of the sensor is proportional to the load current (Fig. 3.1).



(a) Coarse-grain (b) Fine-grain

Fig. 3.2.: Fine-grain power management in time, showing greater savings [39].



(a) Coarse-grain [40] (b) Fine-grain [41]

Fig. 3.3.: Fine-grained power management with per-core DVFS and power gating.

It is our aim to utilize these power sensors to demonstrate fine-grain power management in both temporal (Fig. 3.2(b)) and spatial (Fig. 3.3(b)) is possible as opposed to the coarse-grain management (Fig. 3.2(a), Fig. 3.3(a)) in place today [39].

4. TEMPERATURE CALIBRATED POWER SENSOR

4.1 Architecture

4.1.1 Sleep-transistor based Power Sensor

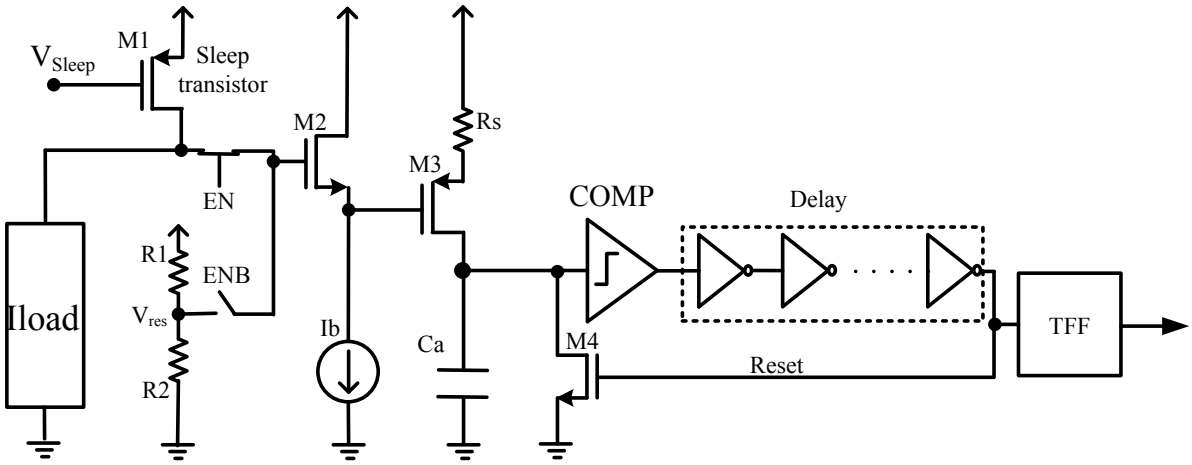


Fig. 4.1.: Circuit schematic of the power and temperature sensor [42].

Fig. 4.1 shows schematic of such a power sensor that provides real-time on-chip estimates [42]. IR-drop V_{DS} is sensed through a source-follower (gain of A_{sf}), amplified and then converted into a current that is proportional to the load current by a common-source FET with a transconductance of G_m. This current is used to charge a capacitor (with a capacitance C_a). When the voltage at this capacitor reaches the threshold voltage of a comparator ($V_{th,COM}$), the capacitor is reset via a delay chain of inverters. By making the discharge time negligible compared to its charging time, an inverse relationship is ensured between the time period of the voltage waveform and the charging current, and by transition, between time period at capacitor and the load current. A T-Flip flop at the end of the delay chain converts this waveform

into a square pulse waveform to be input to a reciprocal pulse counter. As the rate of charging of the capacitor is proportional to load current, the rate of output pulses is proportional to load current.

$$I_{load} = I_0 + \frac{2 \cdot C_a \cdot V_{th,COM} \cdot f_{out}}{R_{sleep} \cdot A_{sf} \cdot G_m} = I_0 + K \cdot f_{out} \tag{4.1}$$

where I_{load} is the load current and I_0 is a constant arising out of charging current at non-zero load current, and fout is the frequency of output pulses From (2), it can be seen that, although the output frequency is proportional to the load current, the proportionality constants are susceptible to process and temperature variations. In order, to obtain a power estimate tolerant to such variations, we integrate a two-point calibration technique, a temperature sensor and a temperature tolerant voltage comparator.

4.1.2 Temperature-tolerant Low Power Comparator

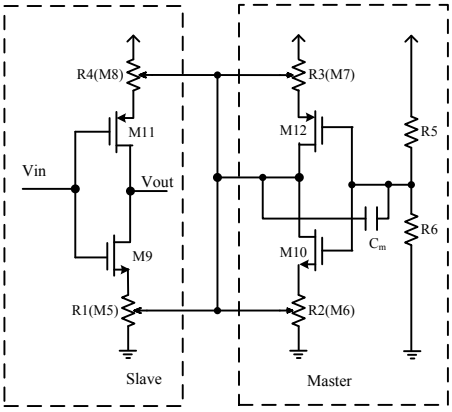


Fig. 4.2.: Schematic of the process and temperature tolerant comparator [42].

A process and temperature tolerant comparator-inverter (presented in [43]) which is used to reset the capacitor(C_a) was incorporated into the design. As shown in Fig. 4.2, this comparator inverter consists of two inverters with voltage controlled resistances at the two supply nodes. One inverter stage acts as the master switch and is fed by a resistor-divider voltage (set to $V_{dd}/2$). Its output controls the resistances

of the four MOSFETs (R1-R4) between the inverters and the supply rails to provide automatic feedback based on ambient conditions. The second slave inverter acts as the actual comparator providing a threshold voltage tolerant to Voltage and temperature variations. As an example, consider the case when this comparator was designed for a threshold voltage of $V_{dd}/2$ at a nominal temperature. Due to temperature variations, if the threshold voltage of this inverter rises to a value greater than $V_{dd}/2$, the input to this switch (resistor-divider voltage) will be lower than the its threshold, hence driving the output slightly higher. Due to increase in this voltage, the resistances R3 and R4 increase, whereas R1 and R2 decrease (compared to nominal case). As a result, the threshold of the inverter switch will be adjusted back, closer to $V_{dd}/2$ compensating for the initial variation due to ambient conditions.

4.1.3 Temperature Sensor

For calibrating the sensor to temperature variations, an estimate of chip temperature is needed. The same sensor can also be used to estimate the temperature by disconnecting the source follower to sleep transistor drain and instead connecting it to a resistor-divider that provides a temperature tolerant voltage input to the sensor. As the threshold voltages of MOSFETs vary linearly with temperature [44], the charging current in this mode of operation can be approximated to increase linearly with temperature and thus, the time period of the output pulse shows a linear variation with respect to temperature.

$$t_{rise} \approx \frac{C_a \cdot V_{th,COM} \cdot (I_c - kT)}{I_c^2} \tag{4.2}$$

where I_c is the charging current at zero Kelvin, and k is a process dependent proportionality constant.

4.2 Temperature Effects

The source follower M2 operating with very low I_b acts as a DC level shifter and hence the effect of temperature on its gain A_{sf} can be neglected.

For the PFET M3, its transconductance, g_{mp}, is given by the following equation

$$g_{mp} = \mu_p \cdot C_{ox} \cdot \frac{W_3}{L_3} \cdot (V_{gs} - V_{th,p}) \tag{4.3}$$

where μ_p is the mobility of holes, C_{ox} is the gate-oxide capacitance, W is the width, L is the channel length, V_{gs} is the gate-to-source bias, and $V_{th,p}$ is the threshold voltage of the PFET. The on-resistance of the sleep transistor (M1), R_{sleep}, is given by

$$R_{sleep} = 1/(\mu_p \cdot C_{ox} \cdot \frac{W_1}{L_1} \cdot |-V_{dd} - V_{th,p}|) \tag{4.4}$$

Comparing (6) with slope-intercept form of a line gives y-intercept, I_0, and a slope, K_1. Combining with (12)-(13), K_1 be understood by the following equation.

$$\begin{aligned} K_1 &= \frac{2C_a \cdot V_{th,COM}}{A_{sf}} \cdot \frac{W_1 \cdot L_3}{W_3 \cdot L_1} \cdot \frac{|-V_{dd} - V_{th,p}|}{V_{gs} - V_{th,p}} \\ &= \alpha \cdot \frac{|-V_{dd} - V_{th,p}|}{V_{gs} - V_{th,p}} \end{aligned} \tag{4.5}$$

where G_m is replaced by g_{mp} in this equation as $g_{mp}R_s << 1$ and all the temperature independent, process dependent parameters are grouped together in the term α.

Dependence of K_1 on temperature is approximated as follows

$$\frac{\partial K_1}{\partial T} = \alpha \cdot \frac{\partial}{\partial T}(\frac{|-V_{dd} - V_{th,p}|}{V_{gs} - V_{th,p}}) \approx \beta \frac{\partial V_{th,p}}{\partial T} \tag{4.6}$$

where β is a process dependent constant that will be calibrated out. From (4.5)-(4.6), it is seen that the slope, K_1, describing I_{load} vs. f_{out} varies linearly with temperature as $V_{th,p}$ is a linear function of temperature [44]. Similarly,

$$I_0 = \frac{1}{A_{sf}} \cdot \frac{I_1}{g_{mp} \cdot R_{sleep}} \approx \frac{1}{A_{sf}} \frac{(V_{gs} - V_{th,p}) \cdot (I_c + kT)}{|-V_{dd} - V_{th,p}|} \tag{4.7}$$

When I_{load} is zero, the overdrive voltage of the common source PFET (M3), V_{ov}, is close to zero, hence I_1 is expected to show a linear dependence on temperature similar

to (4.2). By following the simplifying assumptions made in (4.6), dependence of I_0 on temperature can thus be approximated to the first order as follows:

$$\frac{\partial I_0}{\partial T} \approx \gamma \cdot (\frac{\partial V_{th,p}}{\partial T} + c_1.k) \tag{4.8}$$

where c_1, γ are process dependent constants. Therefore, like K_1, I_0 is also expected to vary linearly with temperature. Thus, we rewrite (6) as following

$$I_{load} = K_1(T) \cdot f_{out} + I_0(T) \tag{4.9}$$

Given the chip temperature, T, the output frequency, f_{out}, can be measured to estimate the load current, I_{load}.

4.3 Calibration

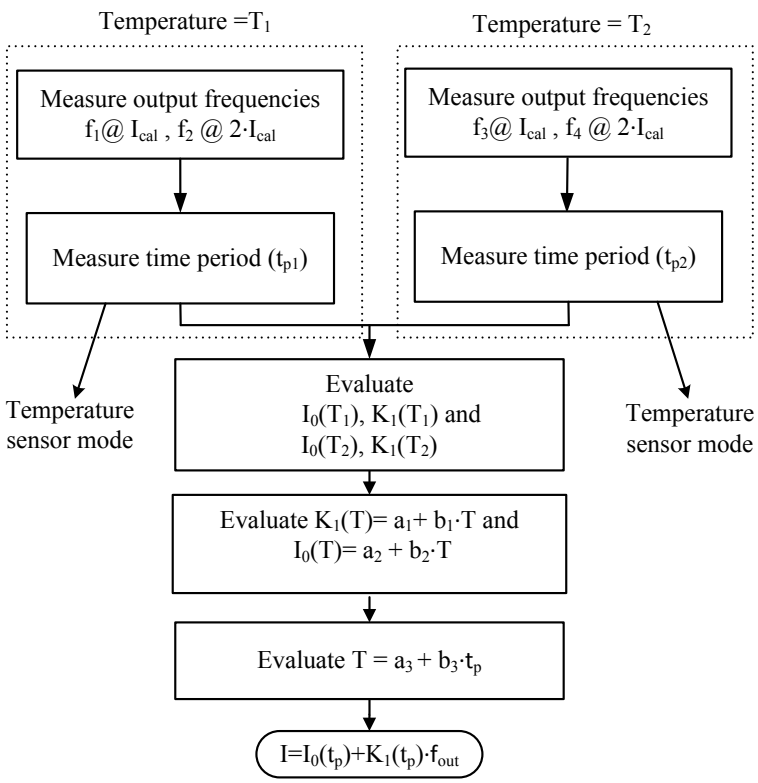


Fig. 4.3.: Calibration flow-chart for the power and temperature sensor.

Calibration algorithm is shown in Fig. refcal. The chip-temperature is a linear function of output pulse-width. Hence, the equation relating temperature(T) as a function of output pulse-width (t_p) can be obtained by measuring output pulse-width (t_{p1}, t_{p2}) at two different test temperatures (T_1 and T_2) as follows:

$$\begin{aligned} T = (T_2 - (\frac{T_2 - T_1}{t_{p2} - t_{p1}}) \cdot t_{p2}) + (\frac{T_2 - T_1}{t_{p2} - t_{p1}}) \cdot t_p \\ = a_1 + b_1 \cdot t_p \end{aligned} \tag{4.10}$$

At each of these two test temperatures (T_i), output frequency is also measured in power sensor mode at two different current loads (I_1 and I_2). Therefore, at each given temperature, the slope K_1 (at T_i) and intercept I_0 at (T_i) for the linear equation between I_{load} and f_{out} are obtained.

As explained earlier, the slope (K_1) and intercept (I_0) also vary linearly with temperature for small ranges in input voltage. Hence, following equations

$$K_1 = a_2 + b_2 \cdot T \tag{4.11}$$

and

$$I_0 = a_3 + b_3 \cdot T \tag{4.12}$$

are obtained where a_1, b_1, a_2, b_2, a_3 and b_3 are process dependent constants.

Thus, at any given ambient condition, the value of load current (I_{load}) is obtained from measured quantities (t_p and f_{out}) as follows:

$$I_{load} = a_2 + b_2 \cdot (a_3 + b_3 \cdot t_p) + (a_1 + b_1 \cdot (a_3 + b_3 \cdot t_p)) \cdot f_{out} \tag{4.13}$$

4.4 Results

This sensor was designed in 45nm SOI process and occupied an on-chip area of 0.0196mm^2 (Fig. 4.4). Monte-carlo simulations showed that in the presence of variations, the estimation error had a mean of 7.5% with a 3-$\sigma_{\max}$ of 15% (Fig. 4.5). With a Vdd of 1.2V, this sensor was tested at various temperatures from 25°C to 85°C for

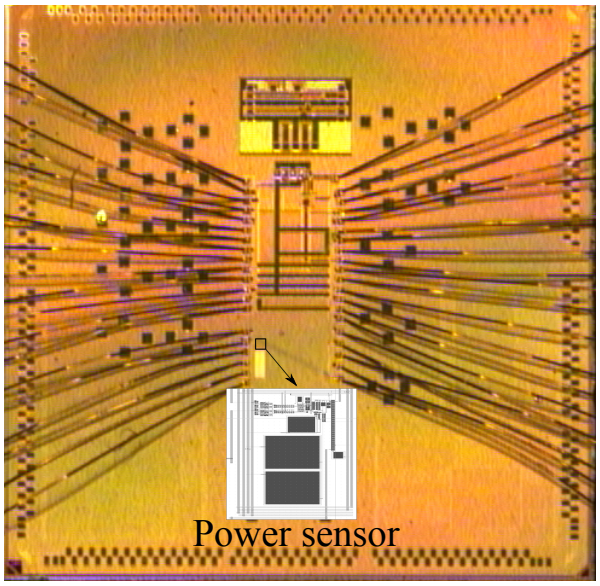


Fig. 4.4.: Microphotograph of the sensor fabricated in 45nm SOI

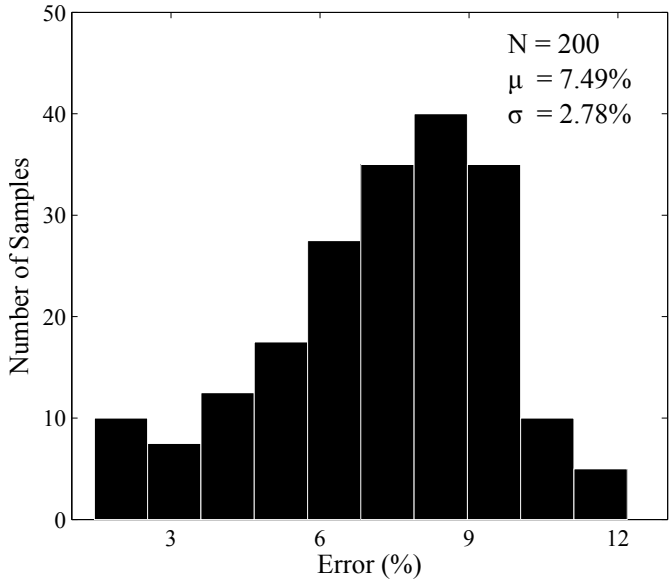


Fig. 4.5.: Monte-Carlo Simulations showing the estimation error under variations.

load currents ranging from 0 to 5mA (Fig. 4.6). For the given sleep transistor design, a current load of 3mA corresponded to a V_{DS} (or IR drop across sleep transistor) of 15mV. For proper functioning of the circuits under test, sleep transistors are typi-

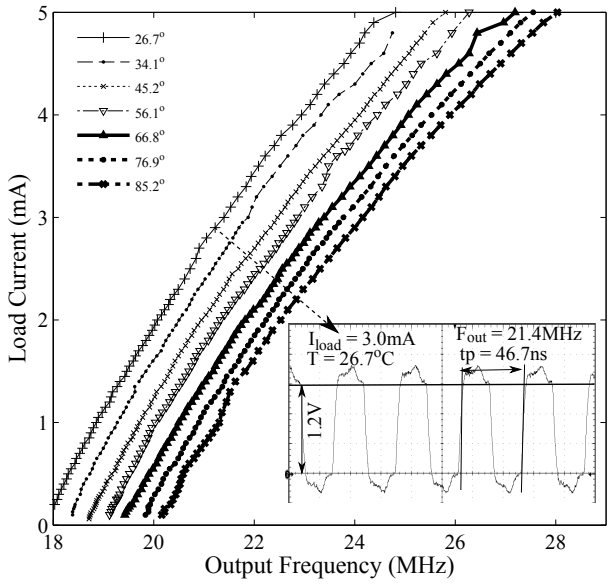


Fig. 4.6.: Measured I_{load} vs. f_{out} at various temperatures in ^{o}C [42].

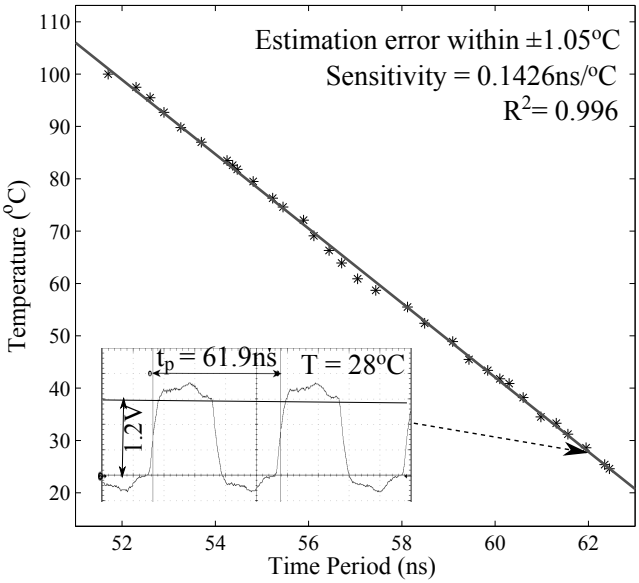


Fig. 4.7.: Measurement as a temperature sensor shows an accuracy of $\pm 1.05^o$C [42].

cally designed to conform to these values of IR-drops [45]. Hence, the power sensor manages to have sufficient dynamic range to monitor average power for most circuits.

The sensor output was monitored by a reciprocal pulse counter implemented using an FPGA running at 500 MHz.

The sensors output pulse width was lower than 54ns under test conditions. So, theoretically, the highest achievable conversion speed would be as high as 18MHz. However, in order to reduce the effect of supply noise and the effect of sampling rate, the output is averaged over a window of 0.5μs (2MHz). With a more accurate, high resolution ($\sim$50ps) on-chip frequency counter [46], response times better than 0.5μs can be achieved. The current overhead of this sensor is 100μA at 1.2V Vdd.

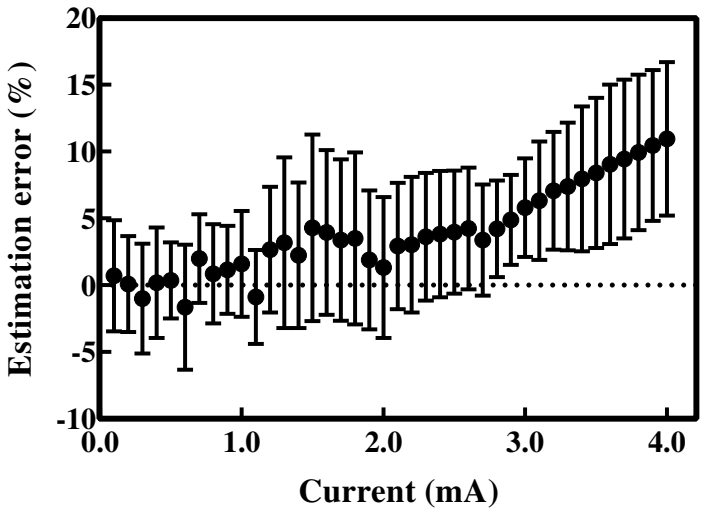


Fig. 4.8.: Estimation errors for loads up to 4mA.

At all the test temperatures, the sensor output showed a linear response with load current (Fig. 3). However, the accuracy of the sensor is limited by linearity in slope (K_1) and the intercept (I_0). Current inaccuracy is estimated as a percentage of the actual load current Hence, the target accuracy of the estimates being within $\pm$10% of the load is limited to current values less than 3.3mA (Fig. 4.8).

In temperature sensor mode of operation, output time period is measured at various temperatures from 22°C to 100°C where the sensor shows linear response with R^2>0.99. The estimation accuracy in this mode was within $\pm$1.05^{o}C of the

on-chip temperature, with a 3-σ error within 4.5^{o}C. This accuracy is also sufficient for thermal management in microprocessors [33].

5. VARIATION TOLERANT POWER SENSOR

Power sensor presented in [42] requires the knowledge of on-chip temperature for accurate power estimates, necessitating a two-point temperature calibration. However, even after calibration, power estimates were susceptible to aging and noise effects. Dynamic range is also limited to a smaller range of input currents, which, although sufficient for average power values, cannot provide accurate estimates of power transients. On the other hand, methods described in [28] and [42] require an external current source for calibration. In this chapter, we present a replica-sleep transistor-based on-chip power sensor with a novel online calibration scheme which shows atleast 5x better resilience to aging effects, 10× better resilience to power supply noise and achieves a wider dynamic range by 10×, while improving the response time by 6×.

5.1 Architecture

5.1.1 Sensor with Replica Structure

Fig. 5.1 shows the circuit schematic of the proposed power sensor which includes a mechanism to sense PVT variations, supply noise and aging degradations. Basic structure of the sensor reported in [42] is retained. Sleep transistors that are used for power gating have a series ON-resistance of R_{ON} when active. A load current I_{load} causes a proportional IR-drop, which is buffered and then amplified by a transconductance stage of gain G_m. Resultant current, I_{chg}, is used to discharge a capacitor C_a from V_{dd}. This node is monitored by an inverter, so that, when the voltage reaches the inveter's threshold($V_{t,inv}$), its output is flipped, which is carried through a delay

line and then used to reset the capacitor to V_{dd}. The output from this delay line is converted to a 50

$$I_{load} = \frac{2 \cdot C_a \cdot V_{t,inv}}{R_{ON} \cdot G_m} \cdot (\frac{1}{tp_m} - \frac{1}{tp_{z0}}) \tag{5.1}$$

where tp_z is the zero error, measured as the output time-period at zero load current. Thus, the output signal has a shorter time-period at higher current loads which can be utilized to obtain a faster response time at a given accuracy, or an improved accuracy with a given response time.

A replica branch is designed to duplicate the gain of the sensor and is used for online calibration. Due to gain compression at larger inputs, systemic errors arising out of calibration are reduced if the input to this replica branch is around mid-range. As the effective input to the sensor is the IR-drop across the sleep transistor, reducing the size of the replica sleep transistor also helps in achieving the same input with a much smaller calibration current.

This circuit is replicated in a second branch with one change: The sleep transistor is scaled down by a factor of "N" to reduce the area and power overhead of the sensor. The ON-resistance of the series resistor in this path now equals N×R_{ON}, therefore, the current required to produce the same IR-drop as the main-branch is reduced by

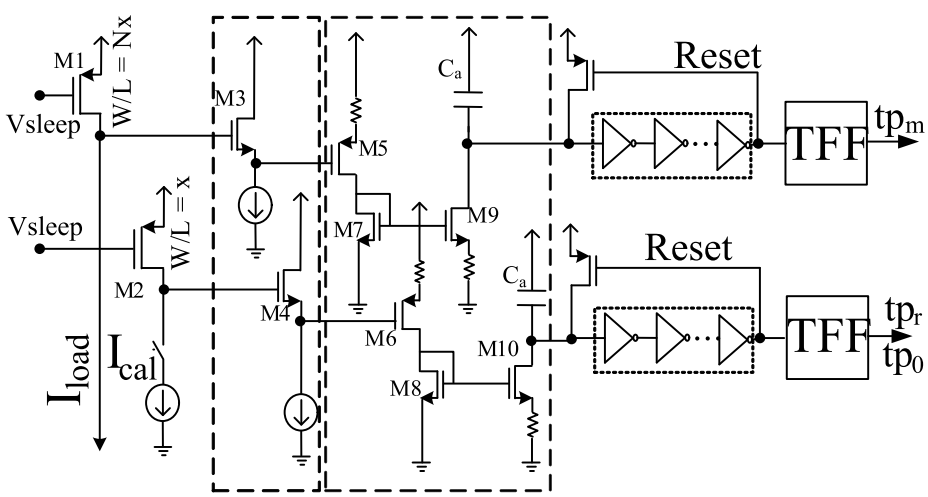


Fig. 5.1.: Architecture of the power sensor

N-times. The output timeperiod of this branch tp_c is related to a calibration current source used to load this branch I_{cal} by the following equation:

$$I_{cal} = \frac{2 \cdot C_b \cdot V_{t,inv}}{N \cdot R_{ON} \cdot G_m} \cdot (\frac{1}{tp_c} - \frac{1}{\cdot tp_z}) \tag{5.2}$$

where C_b is the capacitance which is periodically reset, tp_1 is the zero current output time period of the replica sensor and is related to tp_0 as $tp_{z0} = \eta \cdot tp_z$.

To ensure good matching with the primary branch, all components in the two branches are designed in an interdigital, common-centroid layout. This replica branch can also be used as a temperature sensor in addition to being used for online calibration. As PVT variations, supply noise and aging in the two branches are highly correlated, the ratio of their outputs is tolerant to such effects. Consequently, this sensor can provide variation tolerant estimates of the load current without needing to know the on-chip temperatures as follows

$$\frac{I_{load}}{I_{cal}} = \frac{NC_a}{C_b} \cdot \frac{\frac{1}{tp_m} - \frac{1}{tp_{z0}}}{\frac{1}{tp_c} - \frac{1}{\cdot tp_z}} \tag{5.3}$$

This strategy of measuring current as a ratio of sensor outputs also eases the constraints on the sensor's linearity, enabling a design with wider input dynamic range and higher sensitivity. As noise suppression is vastly improved by online calibration, good noise immunity is achieved even without averaging the sensor output over a long time, improving the overall response time of the sensor. In addition to serving as a means for calibration, the replica branch can also provide an added functionality as a concurrent temperature sensor. When I_{cal} is switched off from the replica branch, I_{chg} depends on the threshold voltages of n and p channel transistors in the gain stages of the sensor [42]. This relationship is approximately linear. As a result, the output pulse rate at zero load, $1/tp_{z0}$, increases linearly with temperature.

5.1.2 Sub-threshold Current reference

A compact, low-power, aging-tolerant current reference, with low temperature coefficient (TC) is therefore needed to make the sensor tolerant to any variations. A few

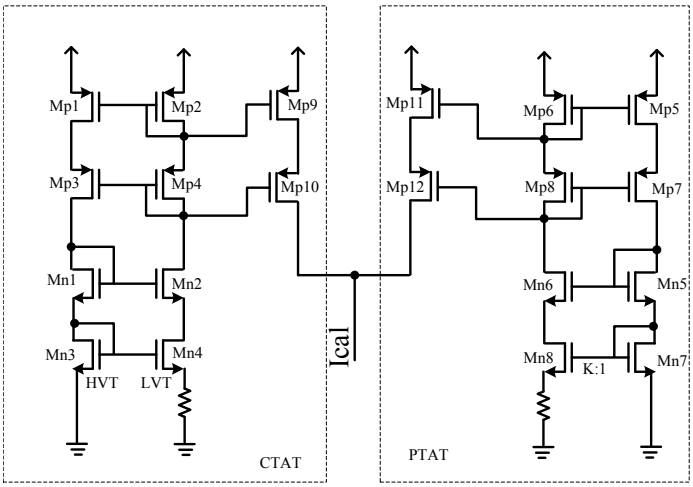


Fig. 5.2.: Schematic of the calibration current source.

current reference circuits for the generation of a temperature-tolerant current source have been presented [47], [48], [49], [50], [51]. Circuit presented in [49] generates a voltage reference from two pnp BJTs and uses this reference to bias an n-MOS at a Zero Temperature coefficient (ZTC) operating point to obtain a reference current. Due to emphasis on digital circuits, today's process technologies may only offer parasitic BJTs which have low current gain β and a wider base-to-emitter voltage spread than those implemented in a BiCMOS processes. [47] uses BJTs to generate a reference current with a very low TC after trim, but trimming requires a total of six measurements at three different temperatures. [50], [48] propose a resistor-less, CMOS only reference circuit that attempts to cancel the variation of threshold voltage with that of carrier mobiliy. However, with footprints in excess of $0.1mm^2$, such designs increase system costs significantly, as multiple instances of local reference currents are preferred to reduce routing overheads.

Fig. 5.2 shows the schematic of an on-chip CMOS calibration current source which overcomes these challenges. This current source consists of four branches biased in weak-inversion, two of which generate a Complementary to Absolute Temperature (CTAT) current, while the other two generate a Proportional to Absolute Temperature (PTAT)current. In both these pairs, the top three transistor pairs (Mp1-Mn2,

Mp5- Mn6) are matched, so that the gate voltages of the bottom n-MOS transistor pairs (Mn3-Mn4, Mn7-Mn8) are also matched. As these transistors are biased in subthreshold region, we can write the following equation

$$V_G - V_S - V_{th} = n\phi_T \cdot ln(\frac{W}{L} \cdot I_{d0}) \tag{5.4}$$

where n is the subthreshold slope factor, ϕ_t is the thermal voltage, W, L are transistor width and length and I_{d0} is a process dependent constant.

As the current in the two branches is matched, by eliminating V_G of the two transistors, the following equation is obtained

$$I = \frac{V_{ta} - V_{tb}}{R} + \frac{n\phi_t}{R} \cdot ln(\frac{W_a/L_a}{W_b/L_b}) \tag{5.5}$$

where I is the current in each branch, V_{ta} and V_{tb} are the threshold voltages and $C_{ox,a}$ and $C_{ox,b}$ are the capacitances of the bottom n-MOS transistors.

In the section generating CTAT current, Mn3 is chosen to have higher than nominal threshold voltage and a low-threshold voltage is chosen for Mn4. In the section generating PTAT current, Mn7 and Mn8 are sized k:1 but have the same threshold voltage. Summation of the currents in these is given by

$$I_{cal} = \frac{V_{t3} - V_{t4}}{R_1} + \frac{n\phi_t}{R_2} \cdot ln(k) \tag{5.6}$$

Therefore, a current source with a low Temperature Coefficient (TC) can be obtained by suitable scaling of k, R_1 and R_2. Although tolerant to variations in ambient conditions, its nominal value may be susceptible to process variations. Therefore, one measurement at any temperature is necessary to calibrate the sensor.

As the reference current is measured post-fabrication, its nominal value itself is not as important as its temperature coefficient. Temperature coefficient of the reference directly impacts the accuracy of the sensor as on-chip temperature is expected to vary across a wide range of values. Measured results of the three samples showed a low TC (<91ppm°C) Fig. 5.3, but significant variance. Monte-carlo simulations were run across 300 samples to estimate the variation in temperature coefficient due to on-chip

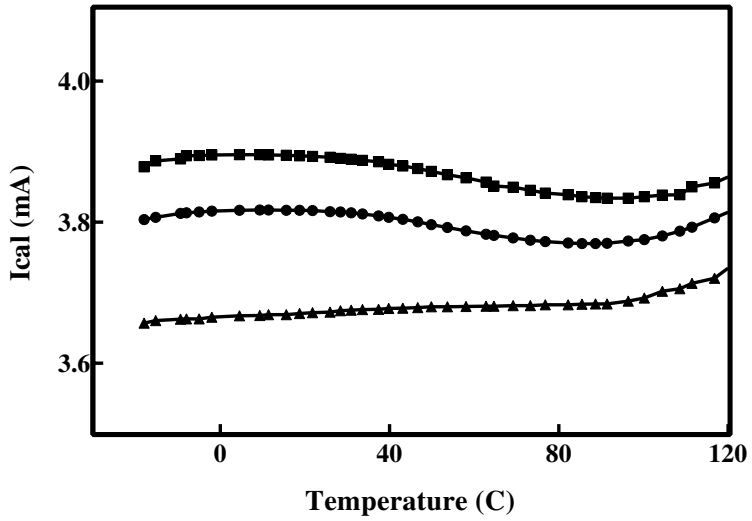


Fig. 5.3.: Measured reference current (-20°C to 120°C, 3 samples).

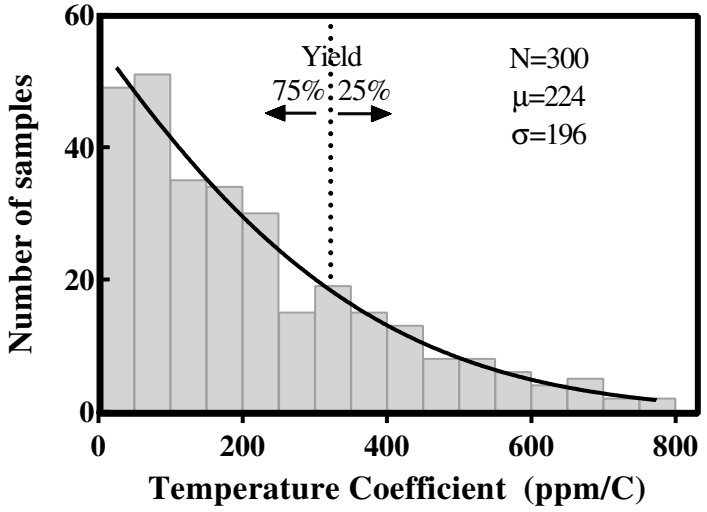


Fig. 5.4.: Variation of TC with mismatch during monte-carlo simulations.

mismatches. Results shown in Fig. 5.4 show a mean of 224ppm/^{o}C and a standard deviation of 196ppm/^{o}C. This distribution implies that for a 75% yield, maximum

value for TC is 322ppm/°C which can result in a decreased accuracy of the sensor by about 3%.

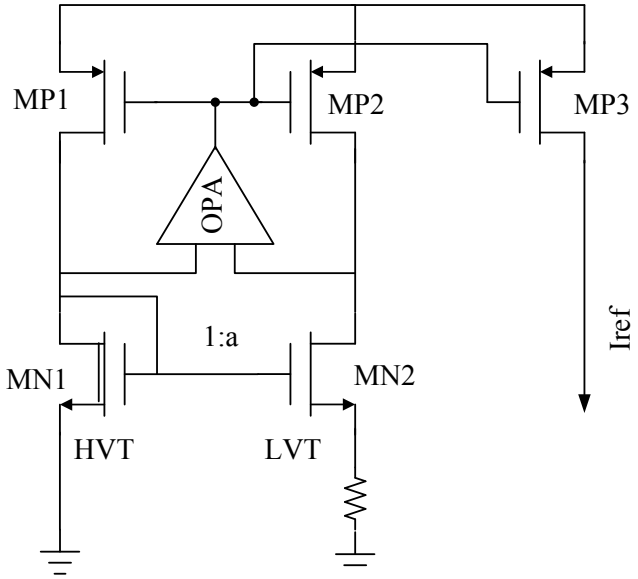


Fig. 5.5.: Circuit schematic of the improved calibration current reference

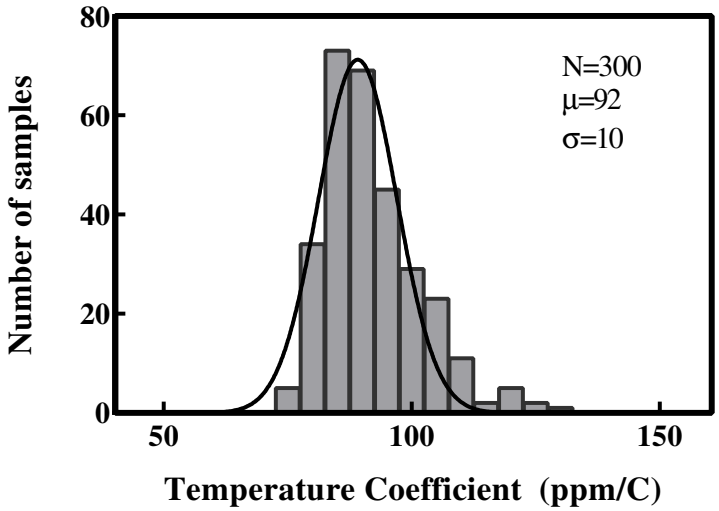


Fig. 5.6.: Monte-carlo simulations showing TC from -20°C to 120°C.

Hence, a reference current design with smaller variation in TC is sought. Fig. 5.5 shows such a design where MN1, a high Vt and MN2, a low Vt n-FETs are sized 1:a and biased in subthreshold. The transistors MP1 and MP2, are biased in saturation by a single-stage opamp with a simple startup circuit. Reference current generated is equal to current in each of the two branches, and is given by

$$I_{cal} = \frac{V_{t1} - V_{t2}}{R} + \frac{n\phi_t}{R} \cdot ln(a) \tag{5.7}$$

As the number of matching elements is reduced from 22 in Fig. 5.2 (20 MOSFETs, 2 resistors) to just 5 in Fig. 5.5, the effect of mismatch is considerably reduced. This can be seen in the Monte-Carlo simulations which show a much smaller spread in the values of Temperature coefficient as seen in Fig. 5.6. The opamp generated bias for p-FETs (MP1,MP2) also improves the line regulation of this circuit to 1%/V (Fig. 5.7) for a supply voltage ranging from 0.85V to 1.5V. However, the nominal value of the reference current is still a strong function of the resistance R1 resulting in significant tolerances of the untrimmed current.

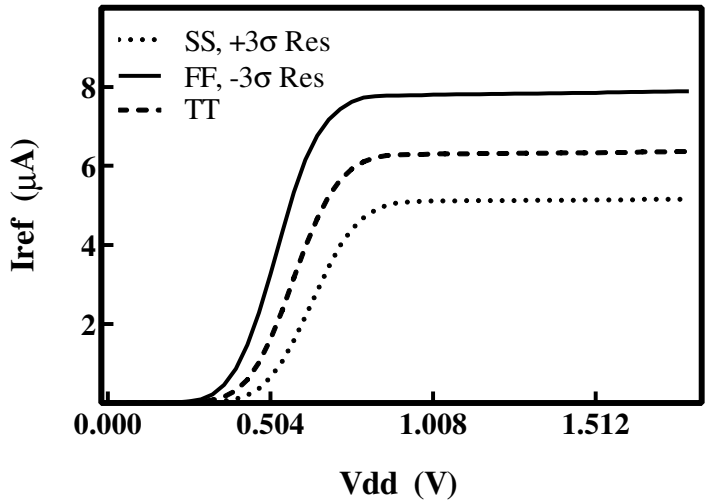


Fig. 5.7.: Line regulation of the current reference at three process corners.

Tab. 5.1 compares this current reference design with the state-of-the-art, stand-alone reference circuits. We find that with a 3-σmaximum of 127 ppm/°C, this circuit

offers a low temperature coefficient reference current with good line-regulation and a very small footprint for sensor calibration.

5.2 Calibration

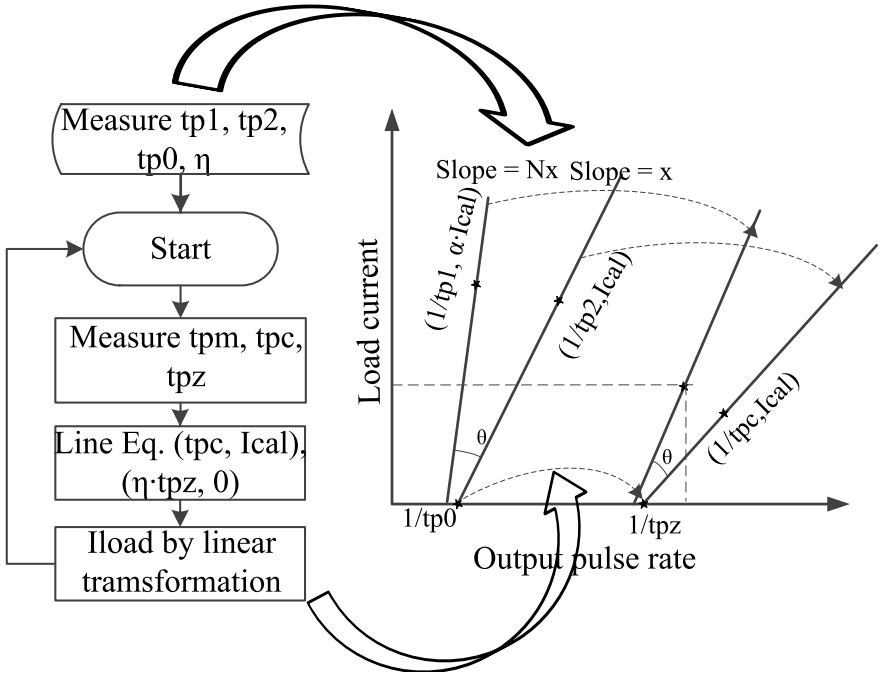


Fig. 5.8.: Online calibration flow chart for replica based power sensor.

From Eq. (5.3), a variation tolerant estimate of I_{load} can be obtained if we can calibrate the scaling ratio, N and know the value of the calibration current, I_{cal}. Calibration of this sensor comprises of two parts as shown in Fig. 5.8. One-time, post-fabrication calibration involves estimating the effective scaling ratio 'N' by loading the main branch with a scaled version of the reference current αI_{cal}. This current is scaled up by a factor of α to ensure that the IR-drop due to this copy in the main branch is large enough to reduce measurement and systematic (linearity) errors. As this current source will be used only once, it does not contribute to the power overhead of the sensor. Any gain mismatches between the two sensor branches are also absorbed into the effective scaling ratio 'N'.

With the αI_{cal} as a load to the primary branch and I_{cal} as the load to the replica branch of the sensor, output pulse rates ($1/tp_1$, $1/tp_2$) are measured. In order to correct for zero error, the output time-periods in the two branches at zero load current (tp_0, tp_{00}) are also measured, with their ratio referred to as η. Scaling ratio N is given by

$$N = (\frac{\alpha I_{cal}}{I_{cal}}) \cdot (\frac{1}{tp_2} - \frac{1}{tp_0})/(\frac{1}{tp_1} - \frac{1}{\eta \cdot tp_0}) \tag{5.8}$$

Thereafter, the value of I_{load} at any given condition is estimated as follows:

$$I_{load} = (\alpha I_{cal}) \cdot \frac{\frac{1}{tp_2} - \frac{1}{tp_0}}{\frac{1}{tp_1} - \frac{1}{\eta \cdot tp_0}} \cdot \frac{\frac{1}{tp_m} - \frac{1}{\eta \cdot tp_z}}{\frac{1}{tp_c} - \frac{1}{tp_z}} \tag{5.9}$$

Thus, a one-time, one-point calibration is sufficient to achieve variation resilient current estimates.

If the sensor needs to be operated as a temperature sensor, a simple, two-point calibration is needed to evaluate the equation of the line relating $1/tp_z$ to temperature, T.

$$T = \frac{T_2 - T_1}{\frac{1}{tp_{01}} - \frac{1}{tp_{02}}} \cdot (\frac{1}{tp_z} - \frac{1}{tp_{01}}) + T_1 \tag{5.10}$$

where tp_{01} and tp_{02} are the output measures of the replica branch at zero load at two different temperatures T_1 and T_2 respectively.

5.3 Non-Idealities

5.3.1 Mismatch

Two components of the mismatch need to be considered.

1. Mismatch within the current reference subcircuit and,

2. Mismatch between the two branches of the sensor.

Mismatches within the current reference circuit that lead to an increase in temperature coefficient were addressed in Sec. 5.1.2. In this section, we focus on the gain

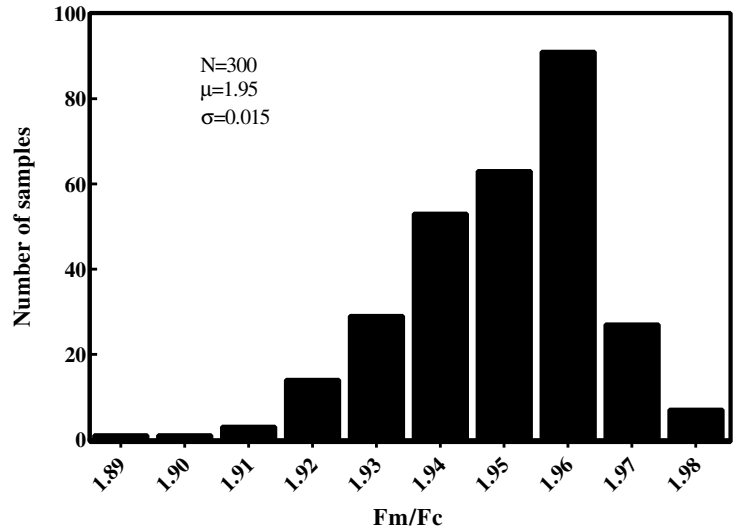


Fig. 5.9.: Monte-carlo simulations to show the effect of sensor gain mismatch.

mismatch between the two branches of the sensor. As the sensor maps the input to output linearly, Eq. (5.1) and Eq. (5.2) can be rewritten respectively as follows

$$I_{load} = A_s \cdot F_m \tag{5.11}$$

and

$$I_{cal} = \frac{A_s}{N} \cdot F_c \tag{5.12}$$

where A_s is the net sensor gain, F_m and F_c are measured quantities from the main branch and the replica branch respectively.

If mismatches in transconductance gain, capacitance, inverter threshold and on-resistance of sleep transistor are absorbed into ΔA_s and ΔN, Eq. (5.3) can be rewritten as

$$\frac{I_{load}}{I_{cal}} = (N + \Delta N) \cdot \frac{A_s + \Delta A_s}{A_s} \cdot \frac{F_m}{F_c} \tag{5.13}$$

As the effective resistance ratio (N) is calibrated by Eq. (5.8), sensor gain mismatches are also absorbed into the same term thereby making first-order gain mismatches irrelevant. However, at high input values, the sensor gain naturally undergoes compression due to nonlinearities in G_m and R_{ON}. In order to estimate the effect of

mismatch on these nonlinearities, Monte-Carlo simulations were run measuring the outputs for midscale and full scale inputs. A ratio of these two outputs is ideally expected to be 2, but, due to gain compression, it will be slightly less than 2. As the input to replica branch falls closer to the mid-scale, whereas the main branch can experience loads from 0 to full-scale, deviation of this ratio from '2' adds to a systematic errors at the two ends. Fig. 5.9 shows that under 3-σmismatches,the minimum value of this ratio is 1.9. Thus, gain compression limits the accuracy of the sensor at full-scale to 95% (3-σ), and 97.5% on average, which can be higher at lower load currents.

5.3.2 Temperature Sensor

From Eq. (5.11) and Eq. (5.12) it is seen that even if the sensor gain of the main branch, denoted by A_s varies with temperature, actual estimate I_{load}, which is a ratio of the outputs of two branches is independant of A_s as follows

$$\frac{\partial I_{load}}{\partial T} = N \cdot \frac{F_m}{F_c} \frac{\partial I_{cal}}{\partial T} \tag{5.14}$$

As F_m and F_c are measured quantities, and N is not expected to vary with temperature, temperature coefficient of the calibration current reference determines the accuracy of the sensor subjected to temperature variations.

To study the effect of temperature on the zero-load output of the sensor, the IR-drop across sleep transistor is assumed to be 0. Therefore, the voltage at the input to Gm stage is $V_{dd} - V_{tn}$. Hence, as described in [42] the current at the output of the G_m stage, I_{chg} varies with temperature as follows

$$\frac{\partial I_{chg}}{\partial T} = c \cdot \frac{\partial (V_{dd} - V_{tn} - V_{tp})}{\partial T} \tag{5.15}$$

where c is a process dependant constant, V_{tn} and V_{tp} are the threshold voltages of p-MOS and n-MOS transistors used in the G_m stage. As threshold voltages of transistors vary linearly with temperature, I_{chg} can be approximated to increase linearly with temperature. Therefore, the output pulse rate at zero-load current $1/tp_z$ is expected

to increase linearly with temperature providing an added functionality as an on-chip temperature sensor.

5.3.3 Supply Voltage

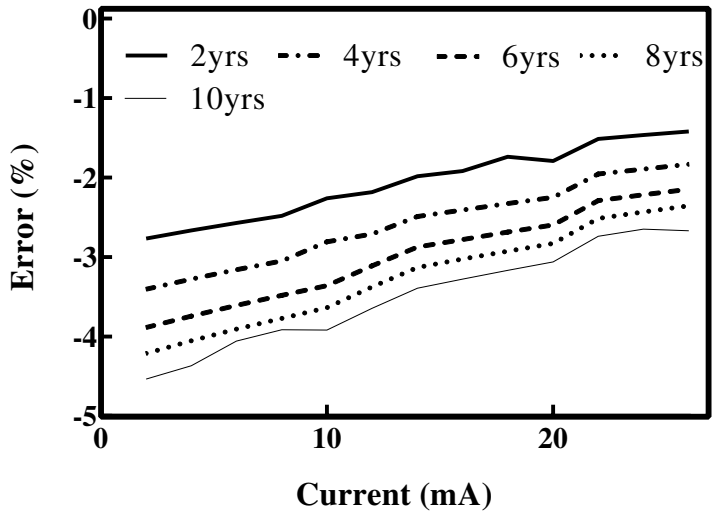
As voltage scaling is a popular power management technique, the sensor needs to be operate at a variety of DC voltage levels. In addition, due to switching activity in the digital and mixed-signal domains, the shared power supply with such circuits is quite noisy. The DC voltage level shift can be modelled as a variation in sensor gain by ΔA_s, and the supply noise is referred to the output as ΔF. From Eq. (5.11) and Eq. (5.12),

$$\begin{aligned} \frac{I_{load}}{I_{cal}} &= N \cdot \frac{A_s + \Delta A_s}{A_s + \Delta A_s} \cdot \frac{F_m + \Delta F}{F_c + \Delta F} \\ &= N \cdot \frac{F_m + \Delta F}{F_c + \Delta F} \end{aligned} \tag{5.16}$$

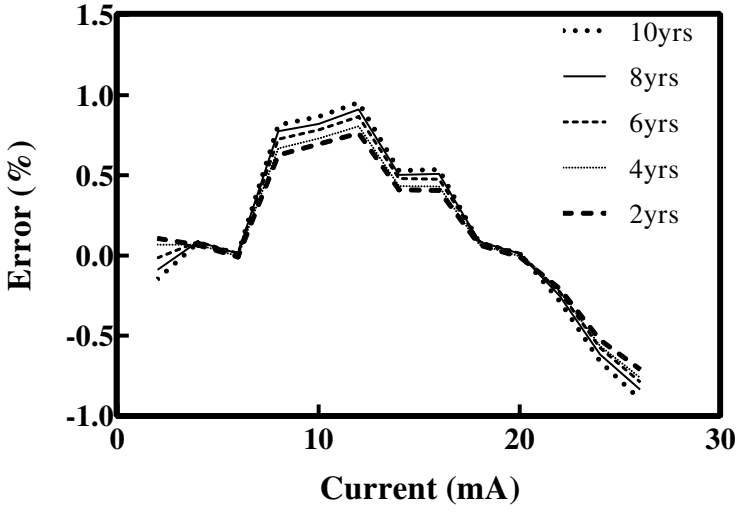
As the two branches have same voltage supply, the effects of supply voltage variation and of supply noise are fully correlated, significantly improving the sensor's power supply rejection.

5.3.4 Aging

Voltage and temperature stresses on devices act through various mechanisms such as Bias Temperature Instability (BTI), Hot carrier Injection (HCI), Electromigration, Time Dependent Dielectric Breakdown (TDDB) to reduce performance. This is typically seen as a shift in the threshold voltage in active devices [54]. Introduction of high-k gate dielectrics and scaled voltages have assuaged the concerns regarding TDDB and HCI respectively [55]. However, BTI related degradations are a significant concern. This mechanism is characterized by an increase in threshold voltage when a device is biased in strong inversion with a small lateral electric field ($V_{DS} \approx 0$) due to breaking of Si-H bonds at the gate dielectric interface.



(a) Error due to aging without calibration



(b) Error due to aging with calibration

Fig. 5.10.: Effect of aging on sensor accuracy

HCI related degradation has a strong dependence on field strength in the channel. For devices of a given length, this is closely related to the drain-source voltage. BTI related degradation is gate voltage depedent. Transistors providing the transconduc-

tance gain in the sensor are biased in saturation and experience similar stresses. The sleep transistor and its replica on the other hand have the same BTI stress (as gate to source voltage of both transistors is either Vdd or 0), but due to different drain to source voltage, undergo slightly different HCI stresses. If aging related degradation is modelled as $\Delta A_{s,a1}$ and $\Delta A_{s,a2}$ for the two sensor branches, output in the presence of aging degradations can be written as

$$\frac{I_{load}}{I_{cal}} = N \cdot \frac{A_s + \Delta A_{s,a1}}{A_s + \Delta A_{s,a2}} \cdot \frac{F_m}{F_c} \tag{5.17}$$

The transistors in the current reference circuit are biased in subthreshold so that they are subjected to very little stress compared to the other devices and hence variations in I_{cal} due to aging are minimal. Due to similarity of stress mechanisms $\Delta A_{s,a1}$ and $\Delta A_{s,a2}$ are highly correlated. In addition, due to similarity of stress levels, their values are also expected to be approximately equal. Fig. 5.10 shows that in the presence of aging effects, the output pulse rate changes by as much as 5% for a given load current within ten years of operation. However, due to replica-based online-calibration, contribution of aging to estimation errors is reduced to less than 1% in the same period of evaluation.

5.4 Results

This sensor was designed and fabricated in 130nm CMOS and occupied an active area of 110μm $\times$ 90μm as shown in Fig. 5.11. The output pulse-rate was measured for current loads from 0 to 25mA at 13 temperatures from -23°C to 100°C for three samples. The time period of the output pulse was averaged over four cycles to improve estimation accuracy and as the longest time period of the output pulse is 20ns, a current estimate is available within 80ns at all times. Fig. 5.12 shows that the pulse-rate varies linearly with load currents with an R^2 >0.99 at all 39 sample points. At room temperature, the sensor was tested for inputs from 0 to 20mA in steps of 10μA and the output was found to be monotonic. This corresponds to a full-scale range of about 11-bits. Fig. 5.13(a) and Fig. 5.13(b) show the equivalent DNL and INL of this

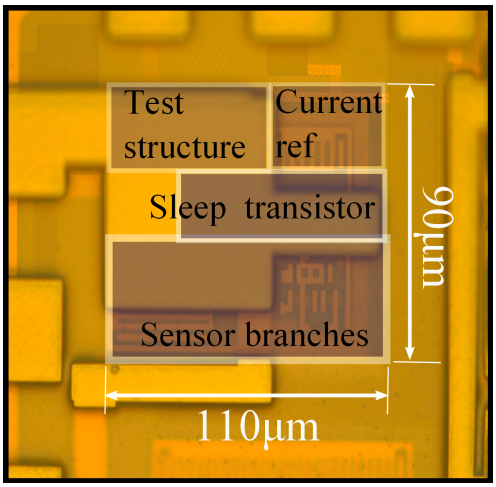


Fig. 5.11.: Die Microphotograph.

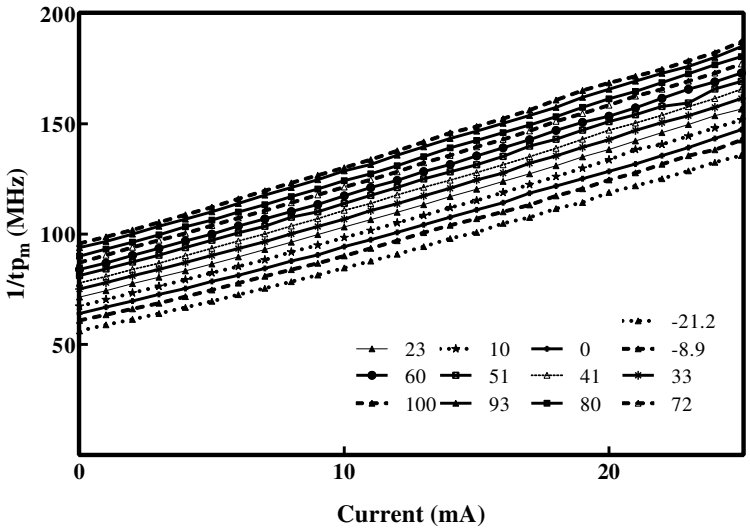
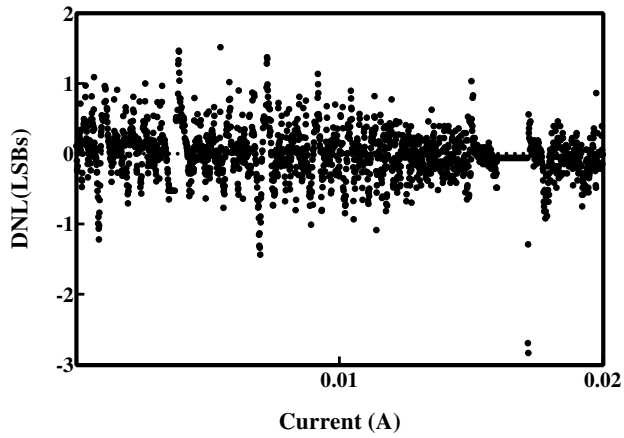
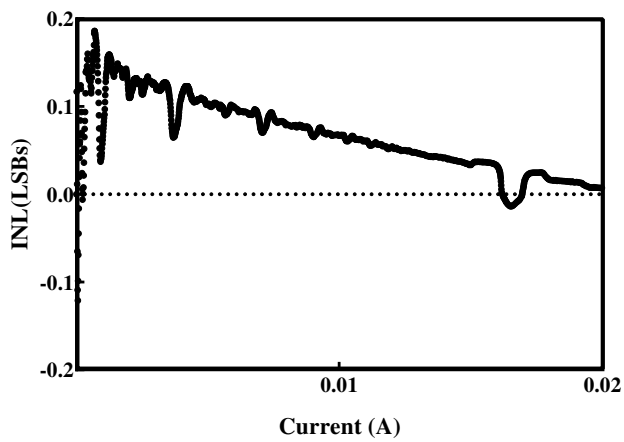


Fig. 5.12.: Measured output of power sensor for loads up to 25mA (-23°C to 100°C).

sensor calculated from this measurement. It can be seen that INL<0.5LSB throughout the range of inputs and DNL<±3LSBs. A load current of 20mA is equivalent to an IR-drop of 100mV across the sleep transistor (at room temperature) which is equivalent to 10% of V_{dd}. For any given circuit block, the sleep transistors are designed such that the virtual supply rail is within tens of millivolt of V_{dd}. Thus, the input dynamic



(a) DNL



(b) INL

Fig. 5.13.: Measured INL and DNL at 23°C.

range for this sensor is from 0 to tens of millivolts, which allows it to be replicated without redesign for blocks across the chip, saving significant design effort. Fig. 5.14 shows that the average error in current estimation across 3 samples and 13 different temperatures was less than $\pm 8.25\%$ with a 3-σ error $\leq \pm 15\%$.

Fig. 5.15 shows the effect of supply noise on estimation accuracy. Single tones of varying amplitudes (10mV-100mV) were superimposed on the voltage supply. Output was sampled for at least two time periods of the superimposed tone or 100ns

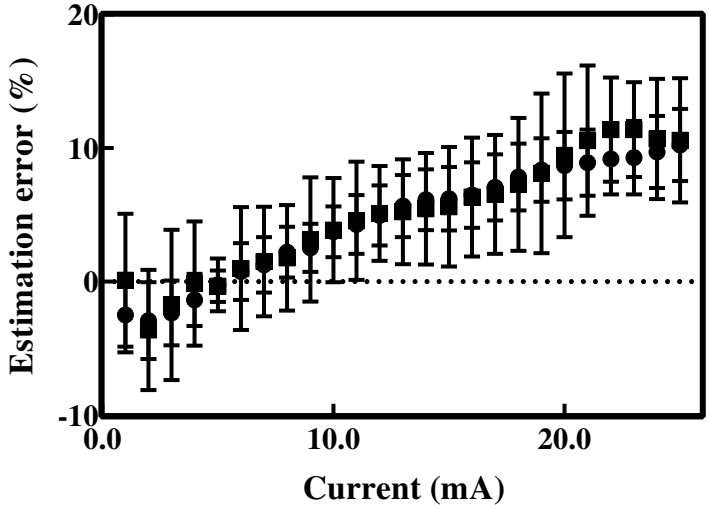


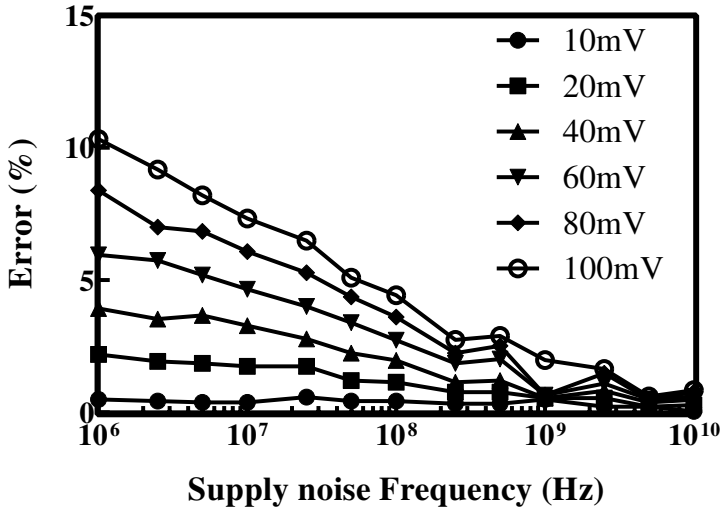
Fig. 5.14.: Distribution of the estimation errors across 40 measurements.

(response time of the sensor), whichever is larger. The maximum deviation in output frequency from the nominal value (without the tone) is reported as error percentage in Fig. 5.15(a). Within this time window, the maximum deviation in the estimated current from the nominal is reported as error in Fig. 5.15(b) and it is seen that due to replica based calibration, the effect of supply noise on output estimate is suppressed by at least 5×.

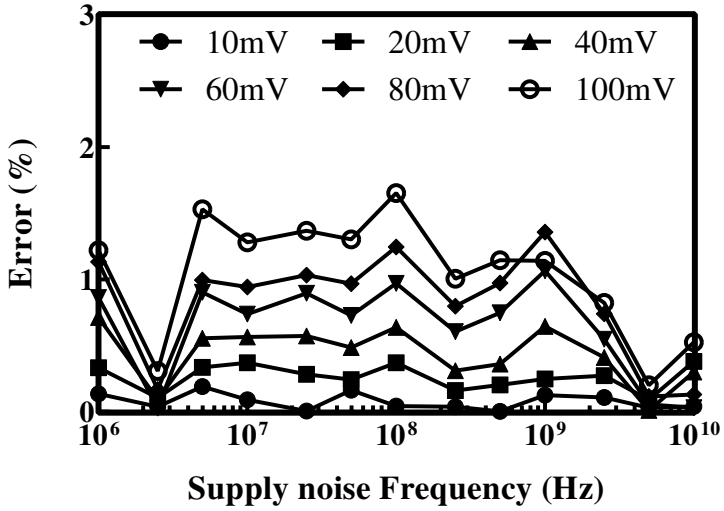
Fig. 5.16 shows that the sensor can provide additional functionality as a temperature sensor, as $1/tp_0$ is varies linearly with temperature. After a two-point calibration, temperature was estimated from the sensor output with an average error of $\pm 0.7^o$C and a 3-σ error$\leq 3^o$C from -20^oC to 120^oC (Fig. 5.17).

5.5 Summary

Energy per Conversion is used as a figure-of-merit to compare the performance of these sensors with other state-of-the art sensors used for power or thermal management. However, this does not account for the time taken to complete the back



(a) Error without replica



(b) Error with replica

Fig. 5.15.: Effect of supply noise on sensor accuracy

annotation of measured parameters into a power estimate and consequently does not show the system-level savings that are possible by utilizing such sensors. Hence, we define FOM_2 on the basis of

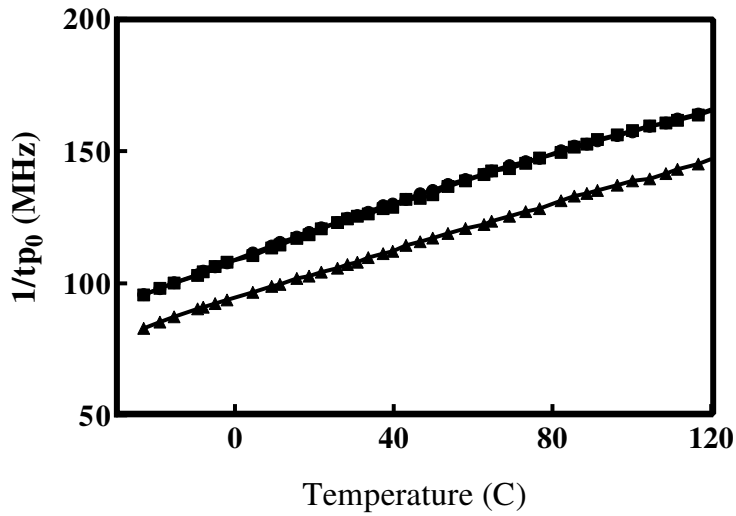


Fig. 5.16.: Linear variation of $1/tp_z$ with tempearture (-20°C to 120°C, 3 samples).

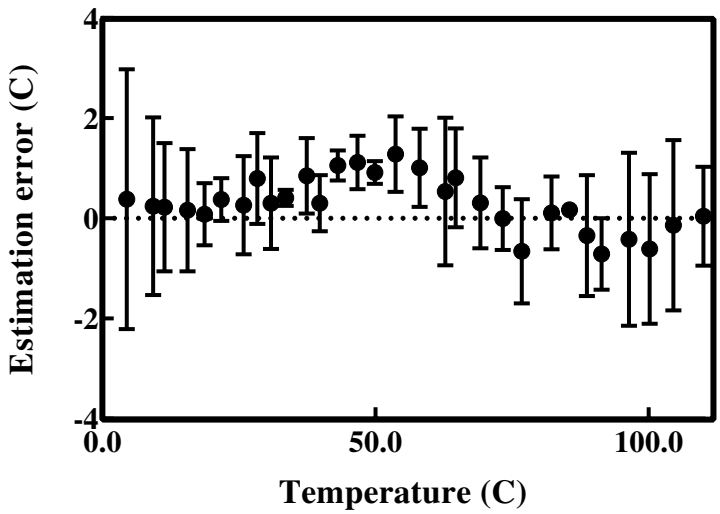


Fig. 5.17.: Distribution of mean and 3-sigma errors for temperature estimation

(a) Response time - from the moment an event took place, to the moment Power Management Unit (PMU) is ready to initiate a response - For thermal sensors, this time includes the time taken for dissipated power to be converted to heat

(which is dependant on thermal resistances in a given technology) and the time taken by the PMU to estimate power. Even if we do not include the (technology-dependent) time taken to convert power to temperature, [34] suggests that in order to estimate power accurately, it can take up to 1.5ms.

(b) Total power overhead - Ideally, it should include the power consumed in PMU. However, as these values are not usually reported, we limit the overhead to power consumed in the sensors.

(c) Inaccuracy - Inaccuracy in estimation can determine the confidence in initiating a response. However, most thermal sensors do not report how inaccuracy in temperature estimate results in inaccuracy in power estimates. [34] reports an average error of around 4% in power estimation based on a combination of simulated models and thermal sensors. We use this value for calculating FOM of all temperature sensors.

With these parameters, FOM_2 is defined as

$$\text{FOM}_2 = \text{Inaccuracy} \times \text{Response Time} \times \text{Power overhead} \tag{5.18}$$

This figure of merit underlines the importance of response time as a faster response time can lead to higher savings by being able to manage workload earlier at the system-level. It can be seen from Table 5.2 that the presented power sensors outperform existing sensors comprehensively.

Table 5.1: Comparison with state-of-the-art reference current circuits

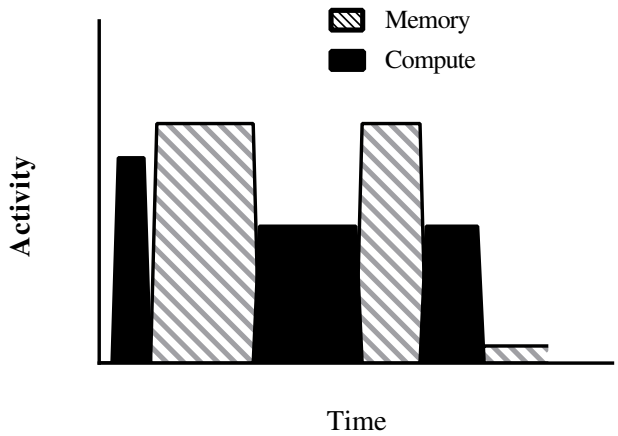
	[52]	[48]	[49]	[51]	[50]	[53]	Improved
Process	0.5µm	0.35 µm	0.18	0.18	0.35 µm	0.13 µm	0.13 µm
Temperature	0-80	-20 - 100	0-100	0-100	0-80	-20-120	-25-125
Vdd	2.3	2.5			1.8-3	1	1
Iref	16-50 µA	92.7 nA	144 µA	7.81 µA	96 nA	5 µA	
TC	130	288	185	24	520	<91	104 (3-σ160)
Power	21 µW	0.81 µW	83 µA	32.7 µA	1 µW	11 µA	
Line regulation	1%	0.1%	-	0.13%	0.2 %	5 %	1 %
Area	0.015	-	-	0.123mm	0.015mm	0.00015mm	0.00009

Table 5.2: Comparison with state-of-the-art in thermal and power sensors

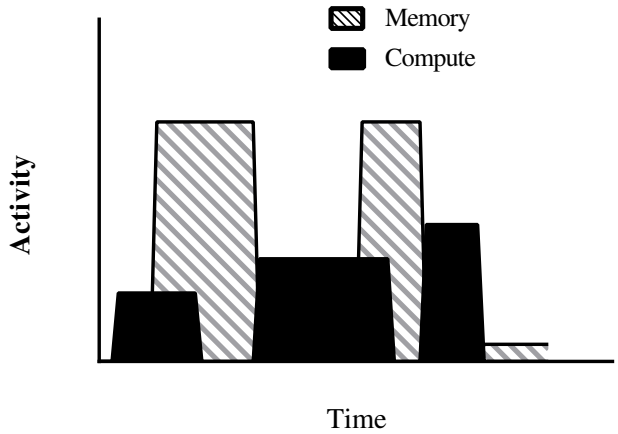
Metric	[32]	[56]	[57]	[58]	[33]	[42]	[28]	This work
Technology	32nm	0.16mm	0.18mm	0.13mm	32nm	45nm	0.13µm	0.13mm
Power overhead	3.78mW	5.1mW	30µW	1.2mW	1.6mW	120µW	82µW	180 mW
Area overhead	0.02mm^2	0.08mm^2	0.18mm^2	0.16mm^2	0.02mm^2	0.02mm^2	1.1mm^2	0.01mm^2
Conversion time	10-100ms	5.3ms	12.5µs	0.2ms	1ms	0.5µs	5ms	0.08µs
Temperature inaccuracy	4.5^{o}C	±0.15^{o}C	±0.5^{o}C	±2.3^{o}C	5°C	±4.05^{o}C	-	±3°C
Current inccuracy	-	-	-	-	-	±10%	±0.03%	±8.25%
Input Dynamic Range	-10-110°C	-55-125°C	0-100°C	0-100°C	-10-110°C	0-0.1V$_{dd}$ (22-85°C)	0-1A	0-0.1V$_{dd}$ (-20-120°C)
Energy per Conversion	37.8nJ	26.5nJ	0.375nJ	240nJ	1.6µJ	0.06nJ	410nJ	0.014nJ
FOM$_2$	228nJ	1.36nJ	1.82nJ	81.6nJ	160nJ	6pJ	12.3pJ	1.19pJ

6. ENERGY AWARE SPEED SCALING

6.1 DVFS Governors



(a)



(b)

Fig. 6.1.: Example showing slack reclamation to reduce energy.

As dynamic power scales by $V^2 \cdot f$, reducing voltage and frequency together offers cubic savings in power while only resulting in a linear order of slowdown. However, as leakage power has now reached ∼50% of the total power consumption [11], slowing down the processor does not guarantee a reduction in energy. On the other hand, increasing the core frequency does not always result in a faster completion of the task due to bottlenecks imposed by realistic memory bandwidths [59]. Hence, the concept of cpu slack sets the upper bound on the energy savings possible for a given application while minimizing throughput penalties. As seen in Fig. 6.1, slack refers to the amount of time the cpu spends in an idle/wait state. Slack can appear at various levels from system level slack (with no active tasks in the pipeline) to instruction-level slack (when the cpu waits for memory access to complete) [60].

Energy aware task scheduling can be classified into static and dynamic scheduling. If 'slack' information is available beforehand, a (static) scheduler can maximize processor utilization while meeting deadlines [61], [62], [63]. In dynamic or real time scheduling, task deadlines are known, but their workload characteristics and execution times are unknown. "Phase" becomes an important variable to solve this dynamic scheduling problem. Phase behavior of an application is characterized by the ratio of (instruction level) slack to compute time βwhich is not only task dependent, but more importantly, is also time varying [64]. Depending on the characteristics of the tasks and the specific region during its execution, the task may either be cpu-bound ($\beta \rightarrow 0$) or memory bound ($\beta \rightarrow 1$) .

Thus, a DVFS governor needs to make three important predictions in order to decide the optimum V-F setting

Workload for the next interval

Time taken or delay to finish the given quantum of work at different V-F settings.

Energy consumed to complete the given quantum of work at different V-F settings.

Workload prediction - [65], [66] and [67] have previously researched phase prediction by keeping track of historical phase values in a look-up table. But, these

studies target power management in quanta of tens of milliseconds which is a coarse-grained approach to power management. In fine grained power management, we target scheduling quanta of <1ms. At this granularity, the size of lookup tables needed to predict phase becomes prohibitively expensive. Therefore, we shall treat the systems as memoryless and use a simple last value predictor, i.e workload in the next time window is assumed to be equal to the workload in the present window. Last-value predictor has been demonstrated to be just as effective as some other phase predictors [68].

Delay estimation - Studies such as [69] and [70] build upon [64] in evaluating the effects of realistic memory access by proposing the concept of 'stall cycles' or 'leading loads'. The idea is that while compute time may scale with core frequency, the time required to access data from memory depends entirely on the memory bandwidth. Hence, stalled time can be treated as a constant latency. If t_{stall} be the amount of time a core is stalled during the current time window of T_{win} while running at F_{cur}, the total time required to complete the same tasks at a different frequency F_{new} is given by

$$T_{new} = T_{stall} + (T_{win} - T_{stall}) \cdot \frac{F_{new}}{F_{cur}} + T_{Lat} \tag{6.1}$$

where T_{Lat} is the transition latency to go from one P-state to another.

Energy estimation- Governors based on stall cycles models and the advanced CRIT models have been presented [71], [72]. However, all previous governors rely on the existence of a number of extensive hardware performance counters to estimate power. [73] uses 12 HPCs to form an offline power model, While [74] and [75] use five different counters to estimate power for various frequencies. [65], [76] and [77] use training to create statistical models based on "architectural signatures" which must be revalidated for newer architectures. In [71], the issue of estimating power accurately is not addressed at all. The knowledge of static and dynamic power consumption at each given interval is assumed to exist. [72] demonstrates green governors which can optimize EDP or ED^2P as required using, Instructions Executed/retired Per Cycle as a proxy for dynamic power. However, observed R^2 for IPC vs dynamic power was

only 0.85 (for Intel architecture) and 0.67 for AMD architecture. Moreover, as the static power is not accounted for, extensive calibration is needed to generate a lookup table of static power values at all possible combinations of V-F. In addition, the effect of temperature on static power consumption is completely ignored.

If the sensor presented in [53] is used instead of the aforementioned performance counter, power can be estimated with an R^2 of 0.99 and at any given conditions. More importantly, the power estimates here include both static and dynamic power, leading to a more accurate DVFS setting.

6.2 Algorithm

Algorithm 1 DVFS governor using proposed sensor

```
Every scheduling window
for first 100 nanoseconds do
    Clock gate
    Read sensor outputs as P_s, T_this
end for
P_d ← P_out - P_s
Read stall time from the Idle counter
for all P-states do
    Estimate Delay_next, P_s,next, P_d,next
    Calculate M, metric to be optimized (EDP, PDP or ED^2P)
    if M_i <0.9· M_min then
        M_min ← M_i
        P_next ← P_i
    end if
end for
return P_next
```

Our sensor provides values of (static + dynamic) power during the normal mode of operation with a fast response time of <100ns. As static and dynamic power scale differently with V-F, in order to estimate power at a different V-F setting, we need to know the breakdown of total power into its static and dynamic components. P_{static} is measured by gating the system clock for a short period of time, equal to sensor response time ($\tilde{1}$00ns) at the beginning of every scheduling window.

For scheduling intervals under consideration (<100µs), we can safely assume that the temperature does not change appreciably [78]. Thus, static power measured in the 100ns window is used as the static power for the next time window. However, if the RC constants in the future become small enough to affect the accuracy, static power estimates can be updated using temperature readings from the available temperature sensor according to the Eq. (1.5) [79]. It must be noted that by the virtue of being updated every 100µs, that these estimates are more accurate than static lookup table based estimates as used in [72].

Assuming that the workload in the next time window is equal to the workload in current window, static and dynamic energies at P-state different from the current are estimated as

$$P_s = (V/Vc)^2 \cdot P_{s,curr} \tag{6.2}$$

and dynamic Power P_{dyn} at other P-states is estimated as

$$P_d = P_{d,curr} \cdot (V/Vc)^2 \cdot f/fc. \tag{6.3}$$

where V, f, P_s and P_d are the voltage, frequency, static power and dynamic power at a different P-state, and Vc, fc, $P_{s,curr}$, $P_{d,curr}$ are the voltage, frequency, static power and dynamic power in the current time window respectively.

To estimate idle time, stall cycle counters are still needed, and delay values at other P-states are estimated from Eq. (6.1). Total power is estimated from Eq. (6.2) and Eq. (6.3) to choose a P-state for the next interval that results in the optimum chosen metric, be it EDP, PDP or ED^2P.

6.3 Experimental Setup

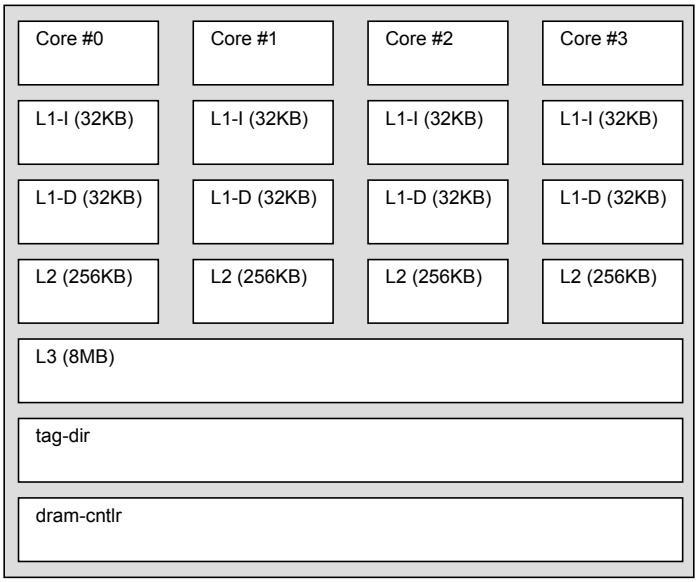


Fig. 6.2.: Topology of the four-core processor modeled in this study.

A 4-core Nehalem based processor system as shown in Fig. 6.2 has been modeled in Snipersim, an interval based simulator [80], which is fully integrated with McPAT (Multicore Power, Area and Timing) modeling suite [81]. Power states that determine the V-F table have been obtained from [82] as is shown in Tab. 6.1 Four controls are considered for baseline

Static optimal For entire duration of the task one static V-F setting is used which yields the lowest metric (PDP, EDP or ED^2P)

Static-worstcase After running the simulation at all static V-F settings, the worst V-F is chosen to highlight the potential loss if wrong V-F pair is chosen

Dynamic-optimal Given task is divided into regions of 10ms, and the best V-F setting for each interval is chosen. This should represent the current theoretical bound on savings, given the coarse grained optimization

Hardware Performance Counters Instructions executed per cycle is used as a proxy for power. i.e For EDP minimization, we use IPC·delay·delay minimization.

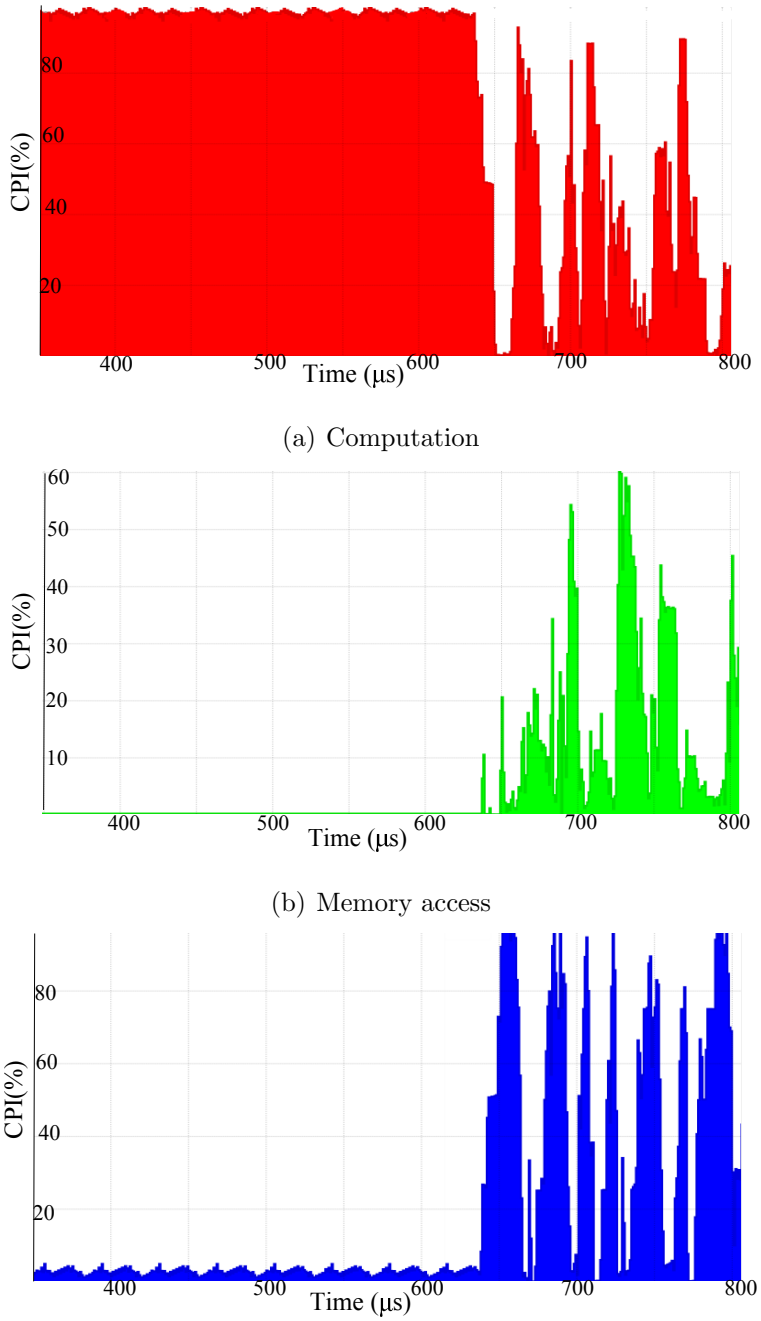
All values are normalized as a percentage of the Static optimal case.

Table 6.1: P-states in a Nehalem processor [82]

P-state	f (GHz)	V_{dd} (V)
P0	1.6	1.484
P1	1.4	1.420
P2	1.2	1.276
P3	1.0	1.164
P4	0.8	1.036
P5	0.6	0.956

Test-benches from SPLASH-2 benchmark suite are run with a pinned scheduler, where a single thread is pinned to one core. Fig. 6.3 shows the normalized CPI stacks for a section of the testbench, *radix* divided into compute, memory access and synchronization branches. Figure. 6.4 shows the effect of grain-length on the DVFS governor in this window. At a scheduling quantum of 1000µs, these phases are missed completely by the scheduler, whereas at 100µs, the scheduler catches only some phases. However, at 10µs, we can see that the scheduler reacts individually to all the fine grained phase behavior exhibited by this workload.

However, it must be noted that using a nave last-value predictor comes with a risk of toggling to a suboptimal V-F. If the phase change interval is comparable to the scheduling interval while changing faster than the controller, the controller will keep switching between wrong P-states for each interval. In Fig. 6.4, this behavior can be observed between 650µs and 700µs, when the frequency toggles between F_{min}



(a) Computation

(b) Memory access

(c) Synchronize

Fig. 6.3.: A section of the CPI stack during the execution of *radix*

to F_{max} only to go back to F_{min}. Hence, it is important to choose the right length of scheduling window to avoid sub-optimal performance.

6.4 Results

In order to quantify the effects of grain length on the DVFS governors, an EDP optimization algorithm was run on all testbenches at three interval lengths 10µs, 100µs and 1ms. As the Nehalem processor is modeled with a transition latency of 2µs, using a finer window than 10µs did not make sense. As can be seen in Fig. 6.5, at all points, power-sensor based DVFS outperformed the HPC based DVFS, sometimes by upto 20%. Based on these results, a 100µs scheduling interval was chosen as an optimum interval length for the given benchmarks while using last-value predictor.

Three different governors were implemented in this system -

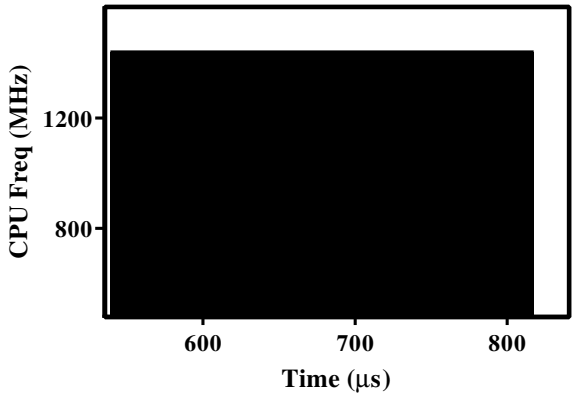
1. EDP optimization
2. ED^2P optimization and
3. PDP optimization

Fig. 6.6 shows the performance of our EDP minimizing governor vis-a-vis our chosen controls. Only in the case of highly cpu-bound benchmarks like fft, fmm and ocean was the sensor unable to perform better than the dynamic optimal case. This may have been due to an overestimation of static power at F_{min}, based on the static power at higher frequencies. As mentioned earlier, we predict static power in a different P-state according to the Eq. 6.2 which leads to an overestimate, perhaps because in treating the processor architecture as a blackbox, we did not explore the details of the clock-gating implemented. This can easily be rectified by storing a lookup table for static power in each of the P-states which is revalidated after regular intervals or with the knowledge of how gating has been implemented which results in a more accurate estimate of static power.

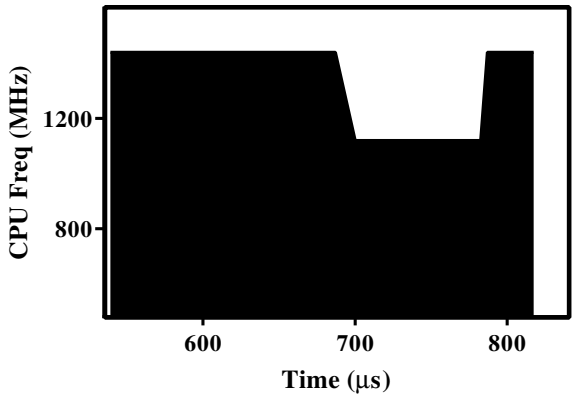
Fig. 6.7 and Fig. 6.8 show the corresponding energy and the time taken (delay) to complete each of these tasks. All values are normalized to the static optimal case. It was observed that utilization of power sensors can lead to a 15% improved EDP on average compared to the same governor running on inputs from hardware performance

counters. Using the wors1t case static P-state as the baseline, the improvement seen was about 37% on average.

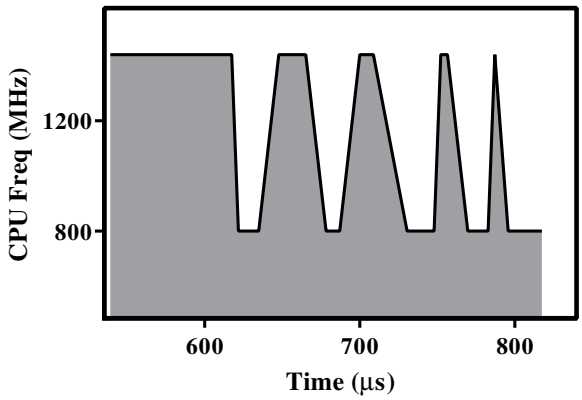
Fig. 6.9 shows the results from ED^2P optimization governor run on the same setup and Fig. 6.10 shows the results from PDP minimizing governors. In each case, due to more accurate, power estimates, using a real power and temperature sensor leads to significant improvements even while running exactly the same optimization algorithm.



(a) Scheduling interval of 1ms



(b) Scheduling interval of 100us



(c) Scheduling Interval of 10 us

Fig. 6.4.: Core frequency within a segment of *radix*.

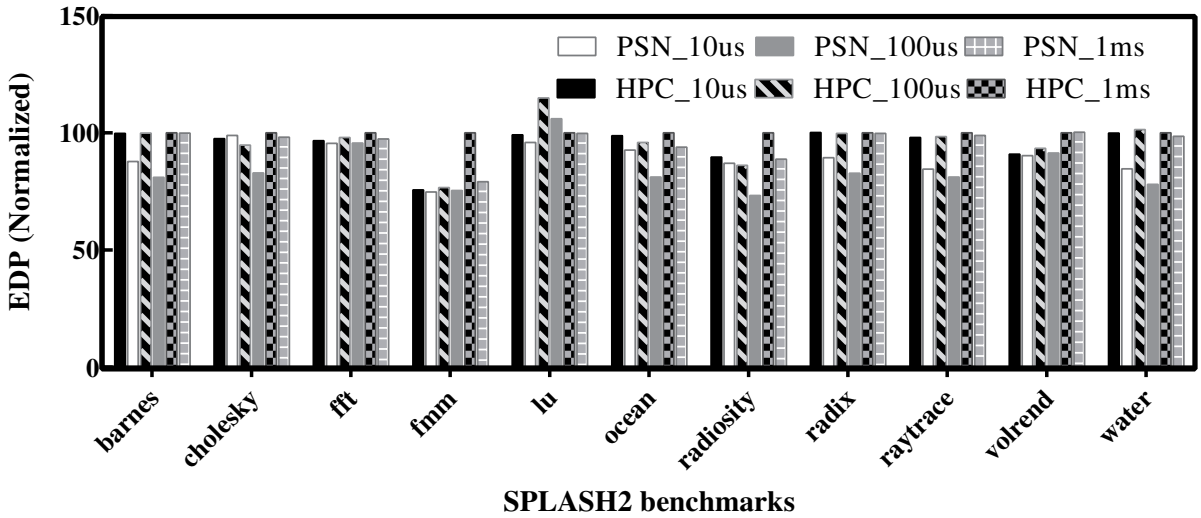


Fig. 6.5.: Performance of EDP minimization governor with HPC and power sensor.

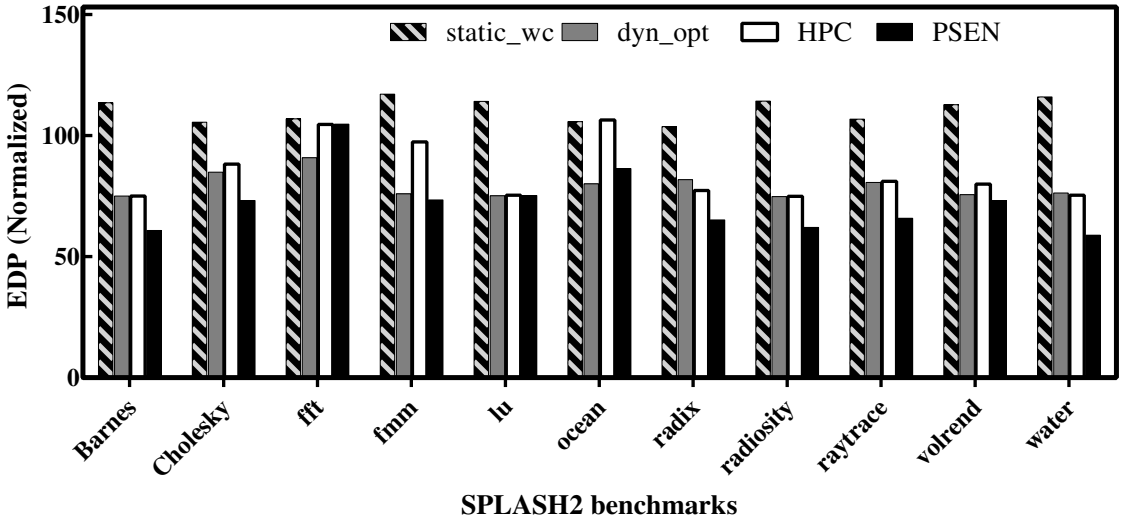


Fig. 6.6.: Performance comparison of EDP minimizing governor.

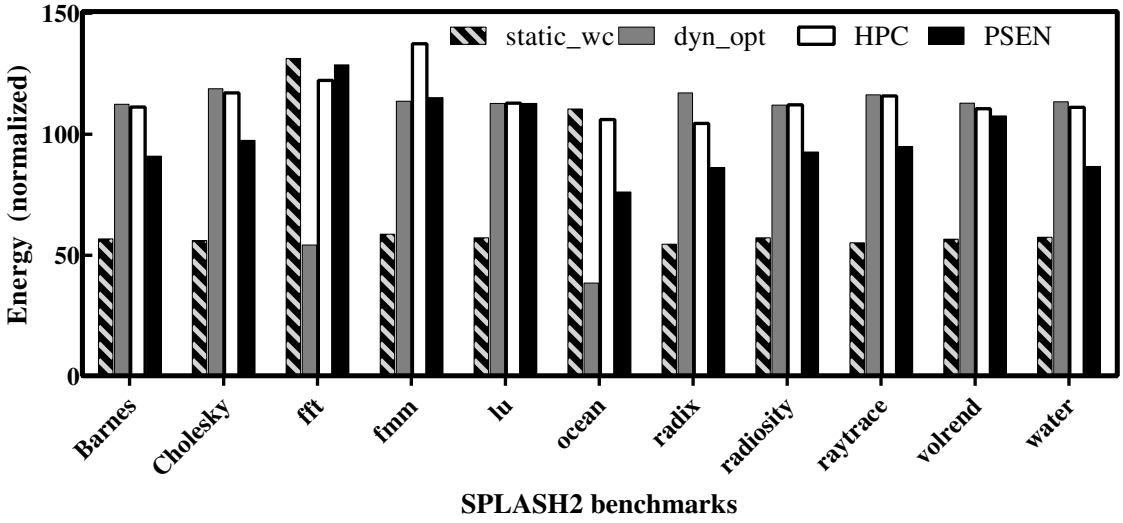


Fig. 6.7.: Comparison of energy consumed by EDP minimizing governor.

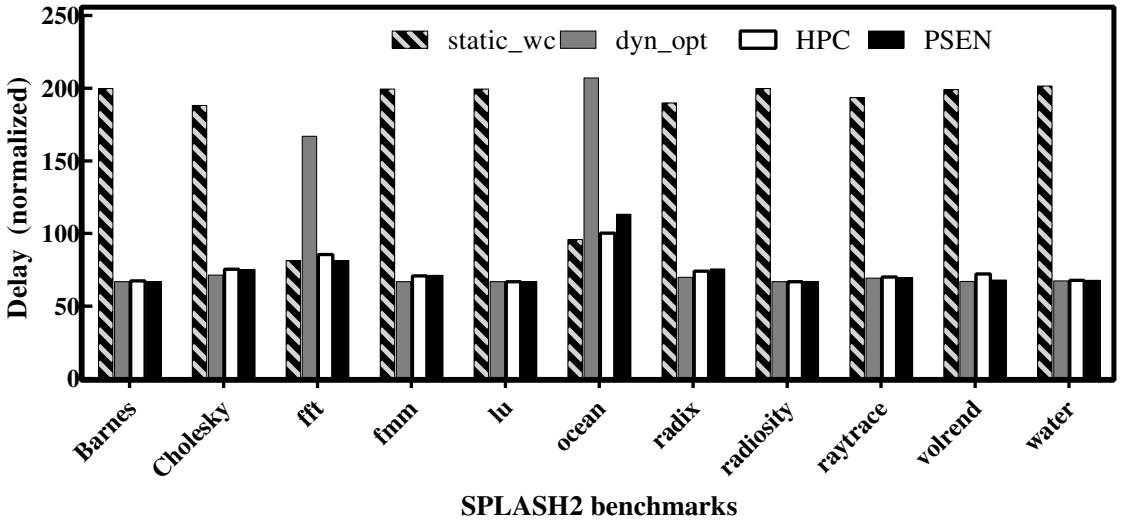


Fig. 6.8.: Comparison of task completion times by EDP minimizing governor.

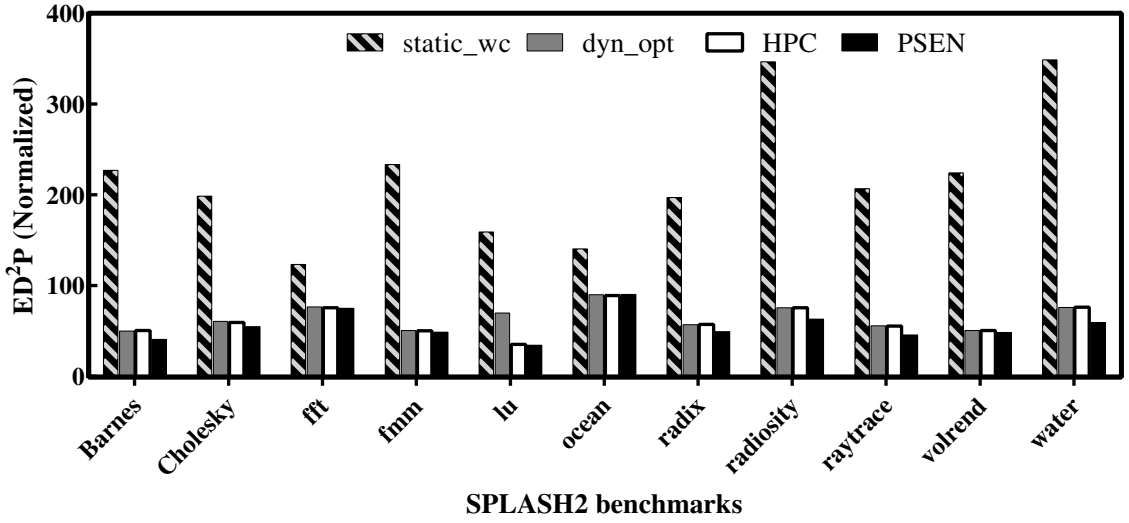


Fig. 6.9.: Comparison of ED^2P minimizing governor for various benchmarks.

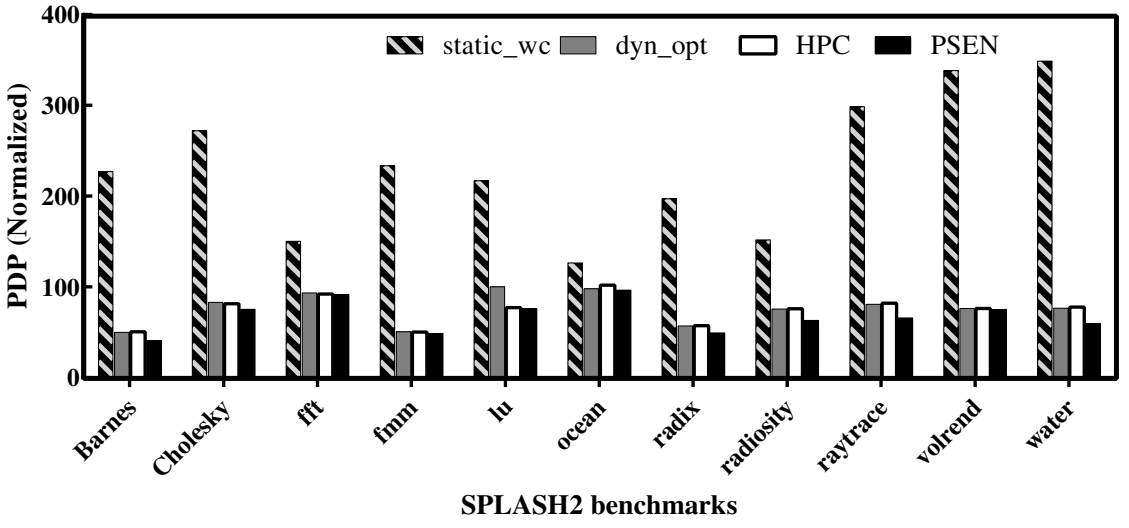


Fig. 6.10.: Comparison of PDP minimzing governor for various benchmarks.

7. DYNAMIC THERMAL MANAGEMENT

7.1 Thermal Management Units

In order to ensure reliable operation in the face of rising power densities, extensive efforts have been made in researching and formulating static as well as dynamic techniques in thermal management over the years. Thermal management techniques can be broadly grouped into

- Scheduling

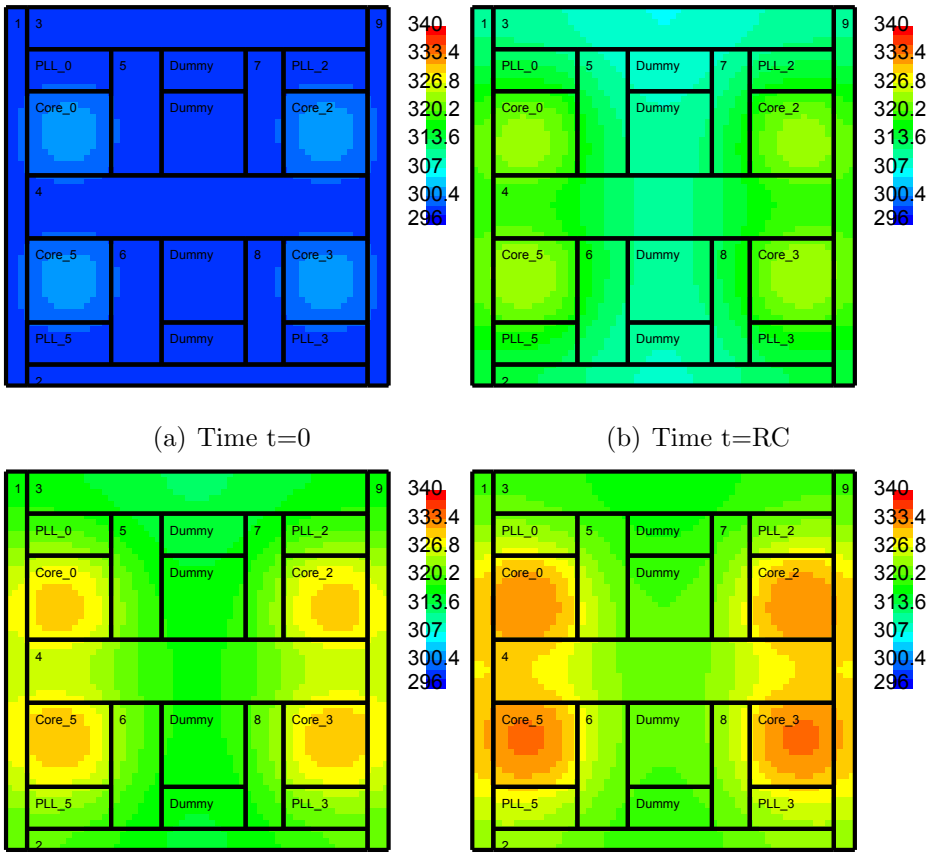
- Voltage and Frequency Scaling and

- Activity migration

Scheduling algorithms have been proposed in [83], [84], [85] to reduce the incidence of hotspots as well as thermal gradients. These algorithms use heuristics to schedule tasks to various cores based on their heat signatures characterized by the steady-state temperatures for execution of these tasks. Similarly, using the knowledge of heat signatures of various tasks, combinations of dynamic voltage scaling, activity migration and clock throttling are used to reduce the incidence of hotspots [86], [87]. [88] and [87] build regression models based on the observed temperature readings. Generation of these models is compute-intensive taking as much as 300ms.

Voltage and Frequency scaling as a reactive and proactive method to reign in heat dissipation has been explored in [89], [90], [91], [92]. However, DVFS to regulate temperature results in significant slowdowns [78] which can be avoided if the use of DVFS is governed by the principle of slack reclamation. Hence, DVFS is better suited to conserving energy.

With the advent of many-core processors, ***activity migration*** as a means of thermal management has become more attractive due to its small transition overheads [23], [93] and lower throughput penalties than those entailed by DVFS policies.



(a) Time t=0

(b) Time t=RC

(c) Time t=2·RC

(d) Time t=3·RC

Fig. 7.1.: Effect of step power input on temperature (Simulated).

Traditional dynamic thermal management schemes implementing core-hopping or activity migration are scheduled at OS-level and at a coarse granularity of 10ms or greater. These policies rely on accurate monitoring of core temperatures and exchange of workloads between a hotter core and a cooler one. While the heating/cooling RC time constants at die level typically lie between 1ms and 10ms, response times of thermal sensors typically vary from hundreds of microsecond to a few milliseconds. Therefore, fine-grained thermal management needs more information than is provided

by thermal sensors alone. Moreover, as power dissipation leads the rise in temperature, ability to predict future power consumption, and, thereby the future thermal profile of a given core is hampered by reliance on thermal sensors alone. As a result, activities that move from high recent activity to low current activity or vice versa will not be properly accounted for.

[87] improves upon this naive algorithm and considers the history of core temperatures to build an online-learning based regression table. In [94], [95], steady-state temperatures of applications are recorded and used to migrate tasks. In [88] assuming stationary workloads, auto regression is used to estimate future temperatures. A distributed algorithm known as MATM is used to determine if exchange of tasks between two cores is thermally efficient However, the conductance matrix requires validation via extensive measurements. Thus, these algorithms rely on apriori knowledge and/or extensive training and retraining of the models. A Power-based prediction algorithm is suggested in [96] which utilizes fourier transform of the power trace to migrate activity amongst various cores by calculating the zero-th moment of temperature as a product of the thermal conductivity matrix and the Power trace. However, this method is inapplicable if the characteristics of the workload (as captured in its fourier transform) are unknown apriori.

In this study, we present a fine-grained predictive dynamic thread migration (PDTM) based on the sensor presented in [53]. Although activity migration was used to demonstrate the advantages of power sensor based predictive thermal management, the same information can also be used for other corrective actions such as fetch or clock toggling or DVFS.

7.2 Algorithm

Heat-flow in a silicon system can be modeled as an RC mesh, where heat sources act as current sources, conducting elements are characterized by their thermal resistance and materials act as thermal capacitances based on their specific heats [97], [98].

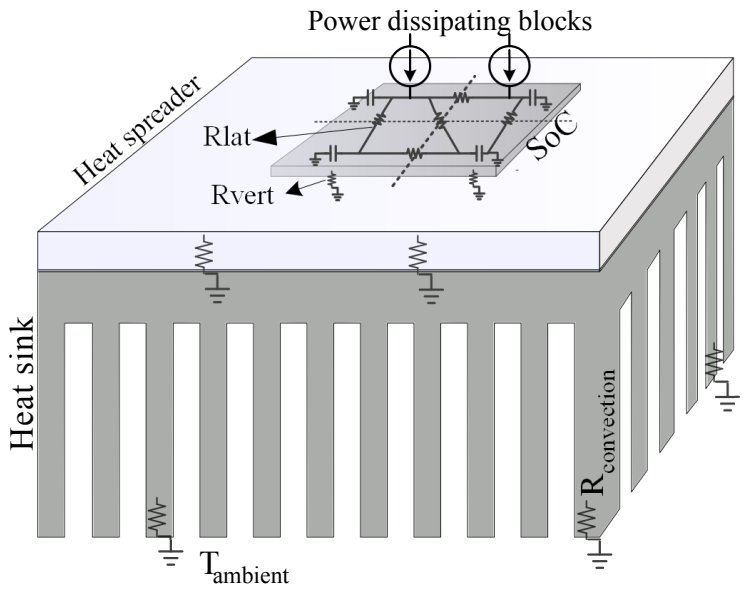


Fig. 7.2.: A typical SoC with its thermal resistances and capacitances.

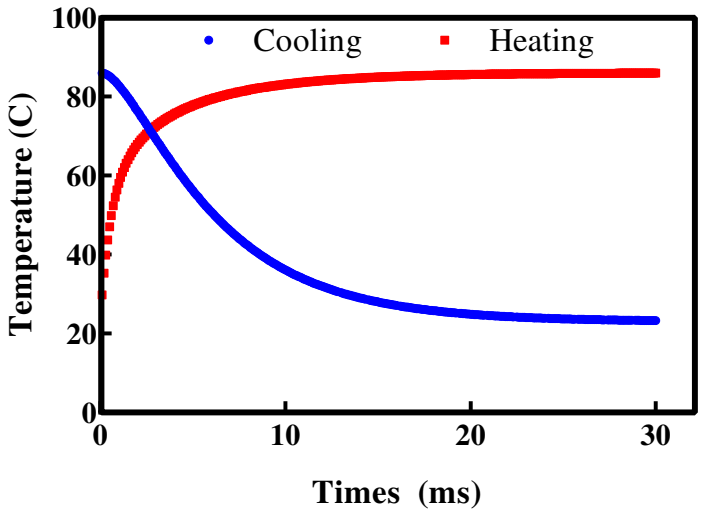


Fig. 7.3.: Measurements showing the heating and cooling profiles on silicon.

Analogous to electrical circuits, the temperature at each node exhibits an exponential time constant, which has been found to be in the order of a few millisecond at die level, and a few seconds at the heat sink (due to high thermal capacitance provided

by the heatsink.) If multiple sources of heat are to be considered (as is the case on a SoC), heat transfer theory gives the relationship between the power consumed and the node temperatures as follows

$$C\frac{dT(t)}{dt} = RT(t) - pU(t) \tag{7.1}$$

where T(t) is the column temperature difference vector = $[T_1\text{-}T_A, T_2\text{-}T_A, \ldots T_n\text{-}T_A]^T$ p is the column power vector = $[p_1\ p_2 \ldots p_n]$ C is the n×n thermal capacitance matrix and R is the n×n thermal resistance matrix.

Converting this to a discrete time, difference equation, this equation can be rewritten as

$$T[i+1] = G \cdot P[i] + T[i] \tag{7.2}$$

or

$$\Delta T[i] = G \cdot P[i] \tag{7.3}$$

where P[i] is the power vector for the i-th interval, T[i+1] is the column temperature vector at the end of ith interval, T[i] is the temperature vector at the beginning of i-th interval so that, ΔT[i] refers to the change in temperature as a result of power consumed during that interval. G is the combined matrix consisting of RC constants for all pairs. This simplification holds true as long as the time intervals are small enough to be able to approximate the continuous time system with difference equations without significant errors.

If ΔT[i] be the temperature difference as a result of the power vector P[i] during the ith interval, following equations hold

$$\begin{aligned} \Delta T[1] &= G \cdot P[1] \\ \Delta T[2] &= G \cdot P[2] \\ &\vdots \\ \Delta T[n] &= G \cdot P[n] \end{aligned} \tag{7.4}$$

where the column vectors can be juxtaposed to form a n×n matrices ΔT_{nxn} = [ΔT[1],ΔT[2] ... ΔT[n]] and P_{nxn}= [P[1], P[2], ... P[n]] and G can be obtained as G =$\Delta T_{nxn} \cdot P_{nxn}^{-1}$

As presented in [53], our sensor can concurrently give estimates of temperature and power. If there are 'n' sources of power dissipation, the first 'n' time intervals are used to "train" the system by evaluating G. Once the G-matrix has been evaluated, the thermal controller takes over. We use a simple last-value predictor. i.e, power in the next scheduling interval for each task (thread) is expected to be equal to power consumed in the current interval. A sensor with fast response time is essential to keep the thread migration overheads small (<2μs)

Even if G-matrix is known and the P-vector for next interval is predicted, assignment of 'n' tasks to 'n' cores is an NP-hard problem for which a complete solution is prohibitively expensive. The complete solution would involve an exhaustive search of all permutations of P-vector to see which distribution yields the most even distribution of temperature (least standard deviation amongst the 'n' core temperatures). This algorithm has a complexity of $O(n!\cdot n^2)$ and is implemented as Alg. 2 as a control to evaluate the theoretical upperbound of performance.

In order to implement a heuristic based algorithm with a much lower order of complexity, we define heat index vector as H= G·P. This vector represents the contribution to rise in temperatures due to power vector P. Values in 'H' are sorted in descending order to be matched with cores sorted in ascending order of their temperatures at the end of i-th interval. Calculation of heat index is $O(n^2)$ and sorting can be achieved in O(nlogn). This heuristic based algorithm shown in Alg. 3 matches hottest cores with tasks having the smallest heat index to achieve a more equitable distribution of temperatures with a complexity of $O(n^2)$.

Algorithm 2 Variation Minimizing Dynamic Thread Migration

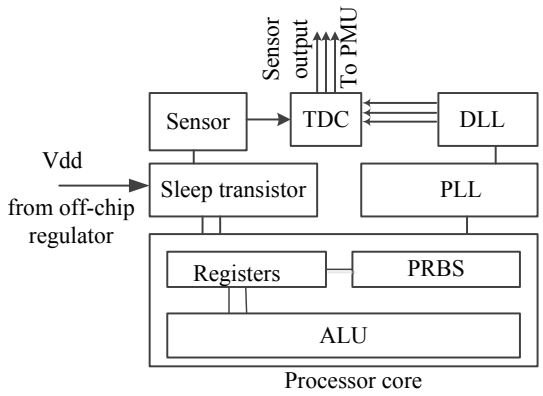
```
Begin task
for first 'n' iterations do
    Read T_i(t) , P_i(t-1)
end for
Calculate G_{n× n}
while tasks in pipeline do
    Predict Pthread_i for i ∈ (1...n) {In this case Pthread_i(n)=Pthread_i(n-1) }
    for all permutations of P_thread to the 'n' cores do
        T_{nextn × 1} ← G_{n × n}·Pthread_{n× 1} + T_{n × 1}
        Find Core ← Pthread such that ΔT_{nextmin}
    end for
end while
```

Algorithm 3 Predictive Dynamic Thread Migration

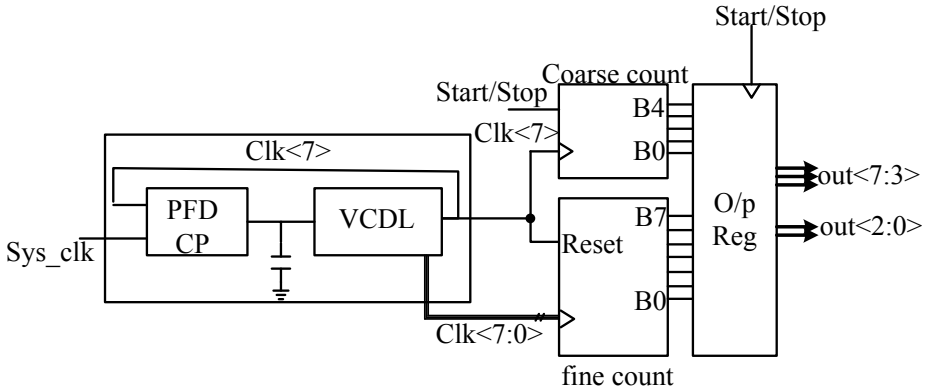
```
Begin task
for first 'n' iterations do
    Read T_i(t) , P_i(t-1)
end for
Calculate G_{n× n}
while tasks in pipeline do
    Predict Pthread_i for i ∈ (1...n) {In this case Pthread_i(n)=Pthread_i(n-1) }
    H_{n×1} ← Pthread_{i,n× 1} · G_{n× n}
    sort H ⇑, sort T_i⇓
    H_(max) → T_i(min)....
end while
```

7.3 System Architecture

In order to evaluate the proposed algorithm, power and temperature sensor presented in [53] is integrated onto a digital "core". This core consists of digital circuits



(a)



(b) TDC

Fig. 7.4.: Block diagram of a single "CORE".

including an ALU, input/output registers and a 32-bit PRBS generator to simulate the load characteristics of a microprocessor core as shown in Fig. 7.4.

Output from the sensor needs to be converted to a digital signal to be interfaced with the thermal management unit. Hence, a Time-to-Digital Converter (TDC) with high time resolution is needed. A counter clocked by a high-fidelity clock source is highly tolerant to PVT variations, but the resolution achieved is 1/Ts where Ts is the sampling period. At a clocking frequency of 2GHz, this yields a resolution of just 500ps. Various Time-to-Digital Converter architectures have been presented in [99], [100], [101] to achieve better resolution. [100] presents a TDC on the principle of a Vernier calipers - two delay lines with differing delays are used to obtain a

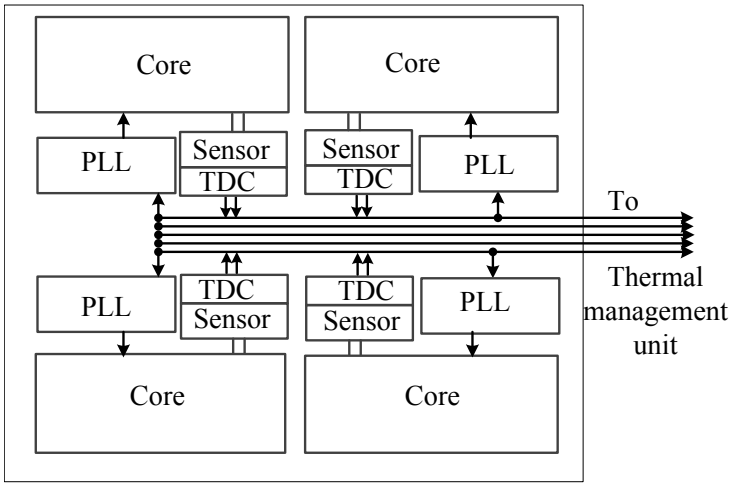


Fig. 7.5.: Block diagram of the 4-core SoC.

resolution equal to the difference of these delays. However, this is prone to PVT variations and needs extensive calibration. [99] presents a time difference amplifying comparators to achieve a resolution of 2.8ps, but occupies an active area of 1350μm^2, which is 10× the core area of the power sensor.

As a solution to achieving a high, PVT tolerant resolution at low overheads, we implment a time-interleaved TDC as shown in Fig. 7.4(b). A DLL generates multiple phases of the given clock signal and these phases are used to run a set of fine counters. The set of fine counters is reset on the edge of phase 0 of each clock cycle, whereas the coarse counters are reset on the edge of the event which is being measured. As a result, all the counters are running at the same frequency F_{clk}, but a resolution of $\frac{1}{(M \cdot F_S)}$ is achieved where M is the number of phases generated by the DLL. Multi-core environment for testing thread migration consists of four such cores each with its own dedicated integer-N PLL, a DLL and TDC as shown in Fig. 7.5.

7.4 Results

Shown in Fig. 7.6 is the block diagram of the test setup with an inset showing the microphotograph of the unpackaged die. Active on-chip area for the four core system

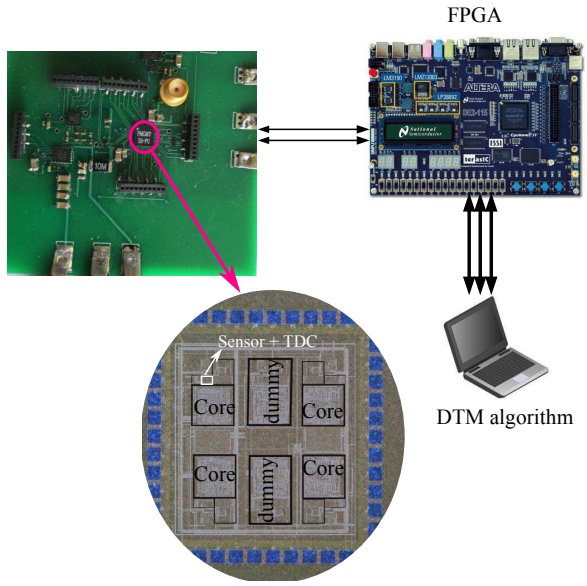


Fig. 7.6.: Experimental setup of the system to evaluate efficacy of PDTM.

is 1.3mm×1.4mm (with two dummy cores), which is limited by the number of pads. This chip has been packaged in QFN with a thermal pad to aid in the heat dissipation. However, no external heat sink is used. It is supplied by an external voltage regulator. Due to a limitation on the number of pads, external test input vectors could not be used. A PRBS input generator is therefore used instead. Power traces for various benchmarks from the SPLASH2 benchmark suite were obtained by simulation in McPAT and repeated until 0.5 second of real,silicon time elapses. These values are quantized into eight frequency settings for the cores. As clock frequency is used as an input, we are limited to utilizing an orthogonal thermal management technique in this case is core-hopping. However, it must be noted that our sensor can be used to perform DVFS or fetch toggling in a real system. The core-hopping algorithms are implemented as a MATLAB code running on a laptop which communicates with the processor via a FPGA.

Worst-case settling time for the PLL is less than 2μs which is higher than even a conservative estimate of the transition penalty imposed due to activity migration [78].

In order to keep this overhead small (<2%), the algorithm reassigns tasks in intervals of 100μs or larger. We use three controls for the experiment

RTM Randomized thread migration

TTM Temperature based migration where tasks between hottest and coolest cores are exchanged and so on

VMDTM The complete power sensor based solution for reducing spatial skews in temperature

Each benchmark has its own distinct power and thermal signature, Hence, we normalize all results (TTM, RTM, VMDTM and PDTM) to the case when the benchmark is run without any thermal management (NODTM).

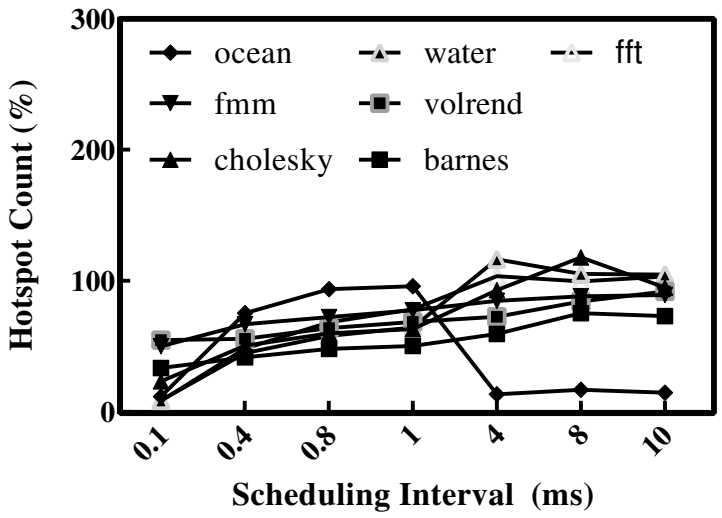


Fig. 7.7.: Incidence of hotspots in a system running RTM.

First set of experiments involved varying the activity reassignment interval- From a minimum interval of 100μs, benchmarks were run for varying interval lengths up to 10ms. In case of RTM, it is seen that the number of hotspots (with temperature >350K) decreases with decreasing interval length. However, the gains taper off.

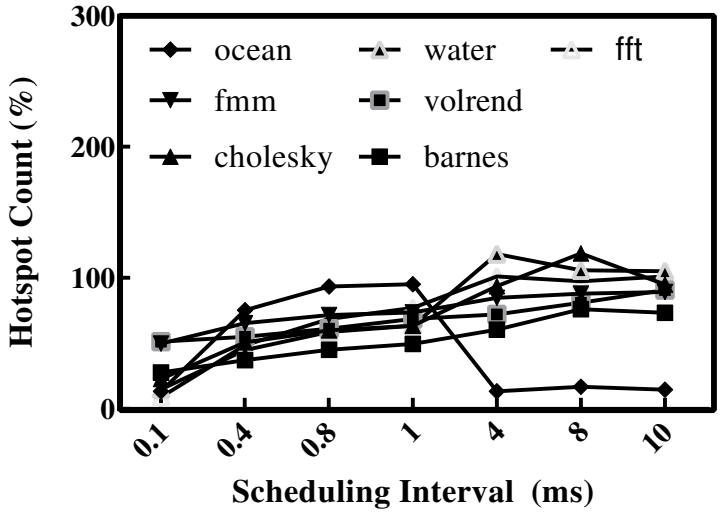


Fig. 7.8.: Incidence of hotspots in a system running TTM.

(Fig. 7.7) In the case of TTM, We observe a similar trend in the reduction of hotspots (Fig. 7.8).

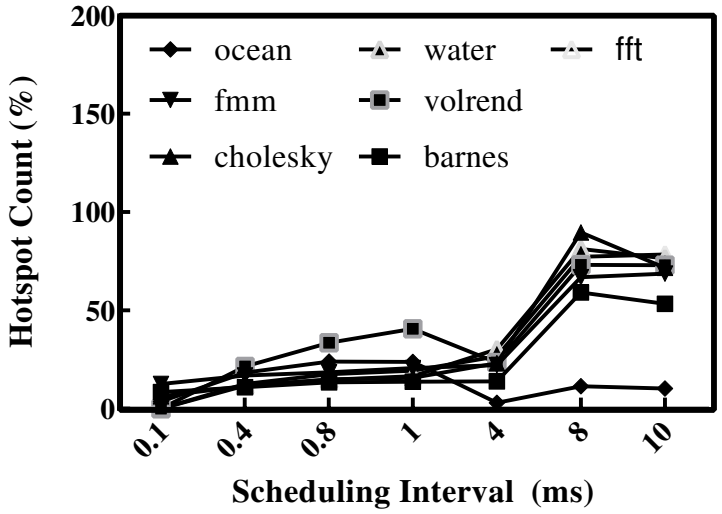


Fig. 7.9.: Incidence of hotspots in system running VMDTM.

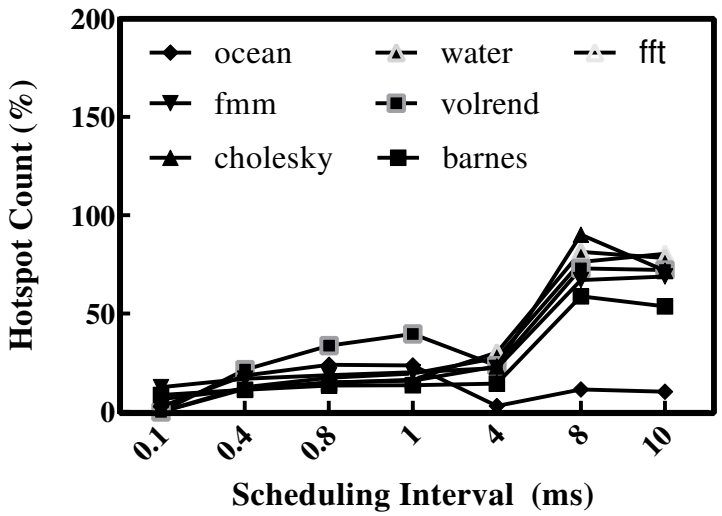


Fig. 7.10.: Incidence of Hotspots when running PDTM.

Fig. 7.9 and Fig. 7.10 show the hotspot count (normalized to NODTM) while running VMDTM and PDTM for various scheduling intervals. The hotspot count decreases as the scheduling interval gets smaller. However, unlike with the temperature based migration, power based task migration continues to show improvements even at fine-grain lengths for task migration. This can be explained by the fact that RC time constants for on-die heating/cooling was observed to be around 9ms (Fig. 7.3). At <0.5ms, as temperature values do not change sufficiently, results of temperature based migration approach that of randomized task migration. However, it must be noted that power-based thread migration performs better than either at all grain-lengths in most of the benchmarks. In addition, the sorting heuristic algorithm works almost as well as the exhaustive search in all cases.

Fig. 7.11 shows the comparison of the best cases of TTM, RTM with a 1ms migration interval for PDTM. It can be seen that PDTM reduces the incidence of hotspots quite effectively. In order to investigate the effects of spatial thermal stress standard deviation was calculated for average temperatures across a 1ms interval. It

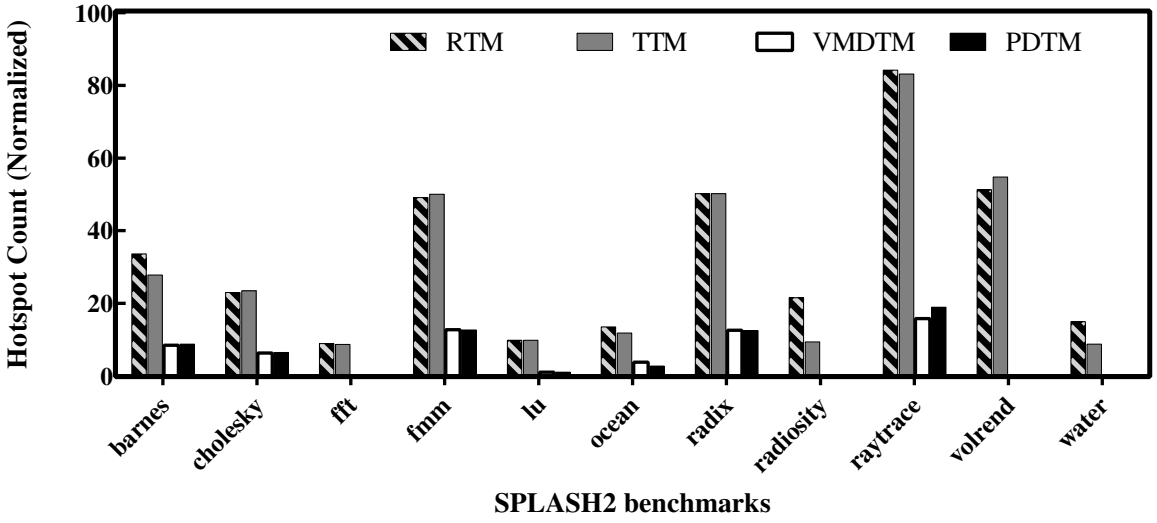


Fig. 7.11.: Comparison of incidence of Hotspots.

can be seen that PDTM is more adept at reducing incidence of thermal stress than TTM and RTM (Fig. 7.12).

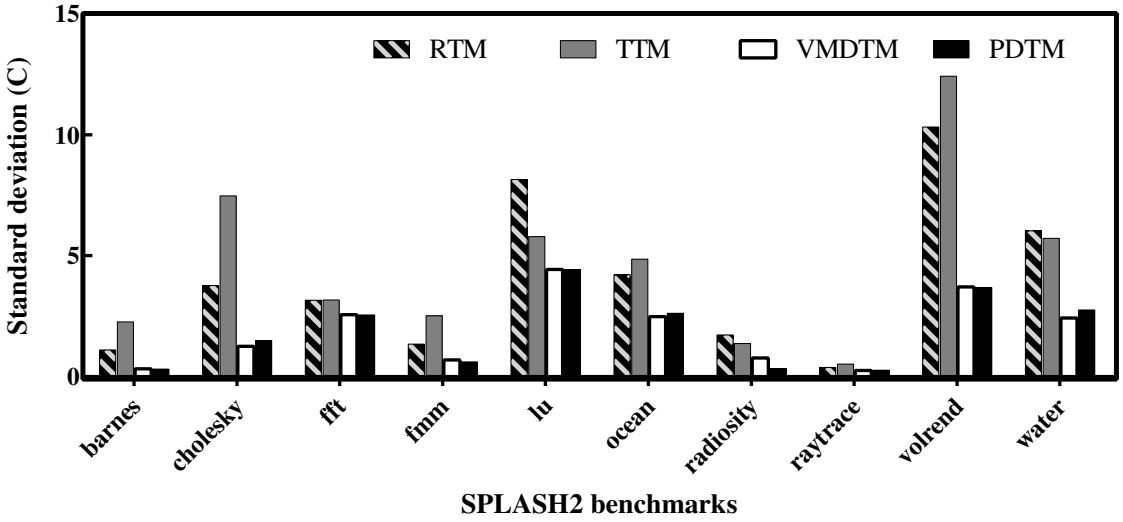


Fig. 7.12.: Comparison of spatial skews in temperature.

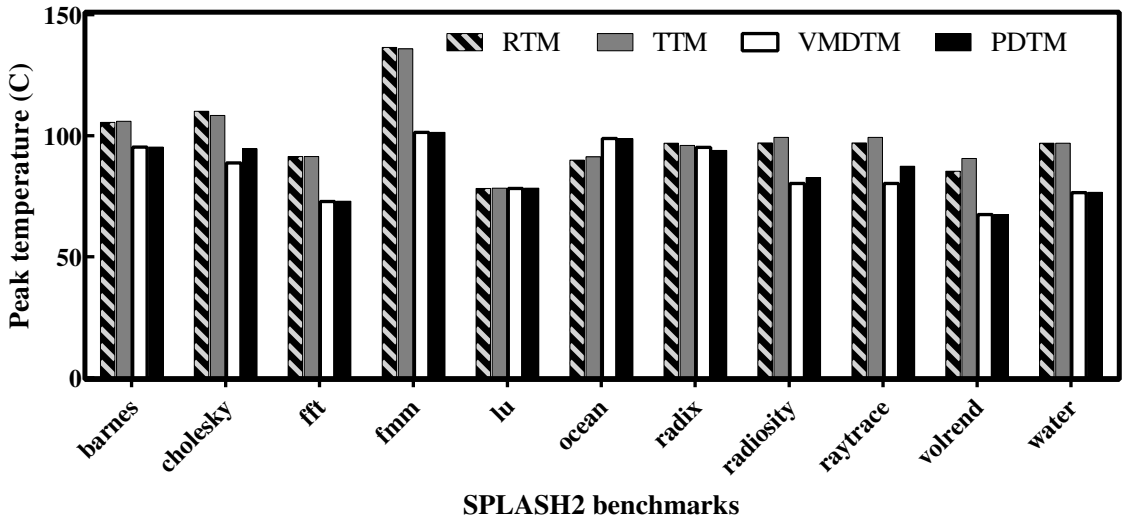


Fig. 7.13.: Comparison of Peak temperatures.

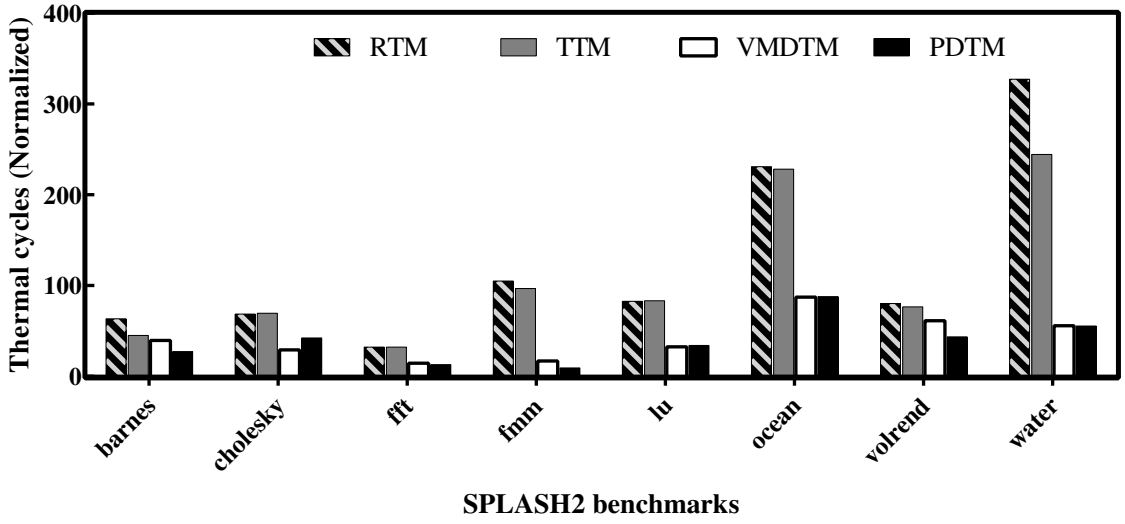


Fig. 7.14.: Comparison of incidence of thermal cycles.

Fig. 7.13 shows the peak temperatures in each cases averaged over 100μs intervals. And an overall reduction of 5°C on average can be seen. Fig. 7.14 shows the incidence of thermal cycles. Fig. 7.15 shows the average amplitude of these thermal cycles in

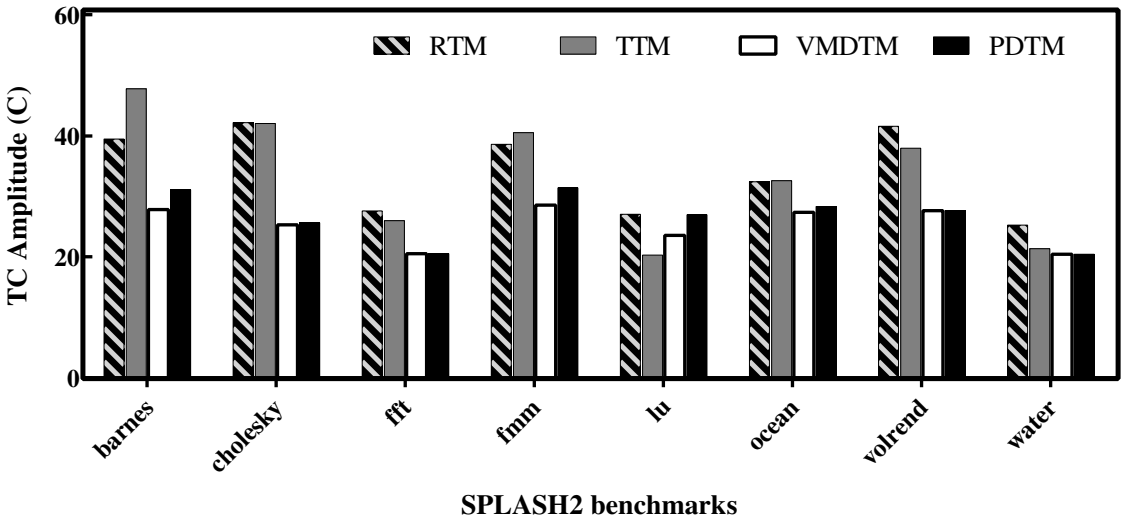


Fig. 7.15.: Average amplitude of thermal cycles.

the case of each benchmark. All results have been presented as a percentage of the case with no activity migration. Therefore, PDTM has shown an average reduction of 2.97°C standard deviation, 13.8°C in peak temperature, 82% in hotspot occurence, and 70% in frequency of thermal cycling compared to TTM and paves the way for finer-grained thermal management of the future.

8. FUTURE OF POWER MANAGEMENT

With the advent of multi-core and many-core processors into mainstream computing, kernel level scheduling algorithms that implement core-stopping, thread-hopping, global and local DVFS techniques to reduce the incidence of hot-spots and improve reliability as well as improve throughput will become more popular [102]. As the number of cores increases, a given core-temperature depends on chip-level heat dissipation and cooling effects rather than just the local (core-level) heat dissipation. As a result, reliability of thermal sensors for power management decreases. At the same time, the number of computations required to back annotate temperature values to local power dissipation increase with the number of sources of power dissipation, which lead to higher power management overheads. Therefore, sensors that rely on true power estimation such as the one presented here are essential in next-generation computing.

Furthermore, there has been growing recognition of the need to define efficiency of algorithms not just in terms of the orders of computational complexity, but also in terms of energy efficiency [103]. Power sensors with quick response times become essential to validate not just the algorithms, but also in order to evaluate the validity of such metrics. In addition, as testing becomes more complicated, sensors needed for on-line testing after the chip has been packaged also become relevant. These sensors enable the implementation of such online test and debug schemes due to the ready availability of output readouts in the form of digital codes.

As more and more electronic systems are connected through networks, saving the power in just one system regardless of its interactions with the other connected systems is insufficient. As an example, most mobile systems (smartphones, tablets, and laptops) are connected to the Internet through wireless networks. When a mobile user watches streaming video, power is consumed on the mobile system, as well as

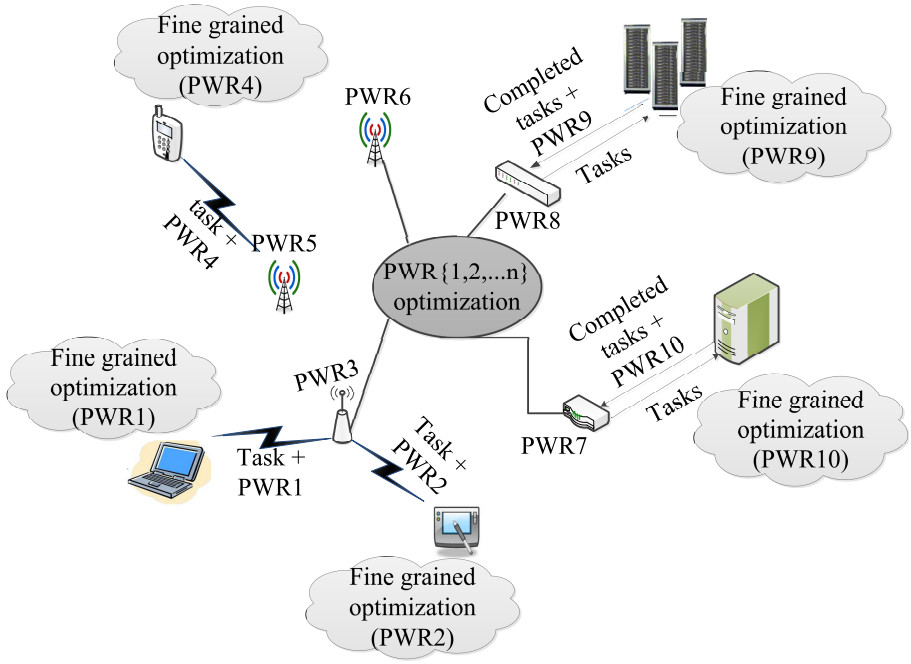


Fig. 8.1.: Envisioned future of power management in a connected world.

wireless access points, network routers, servers, and storage. It is inadequate to separate these connected systems and reduce their power consumption independently. A recent paper [104] proposes the concept of End-to-End Energy Management, suggesting the need to consider multiple connected systems as a whole for power reduction (Fig. 8.1).

Real-time power sensors are essential components for realizing end-to-end energy management because we are able to monitor the power dissipation of multiple systems as they communicate through networks. Moreover, the premise of cloud computing is the ability to autonomously migrate computing to meet performance requirements and resource constraints. The information of real-time power consumption enables researchers and engineers to dynamically adjust power management strategies across systems to ensure better efficiency. Thus, tomorrow's power management strategies require coordination across layers and optimization involves a combination of algo-

rithms at the network, software, kernel and hardware levels (Fig. 8.2). On-chip power sensors that provide real-time power readings with minimal overheads are, therefore, crucial in realizing this future.

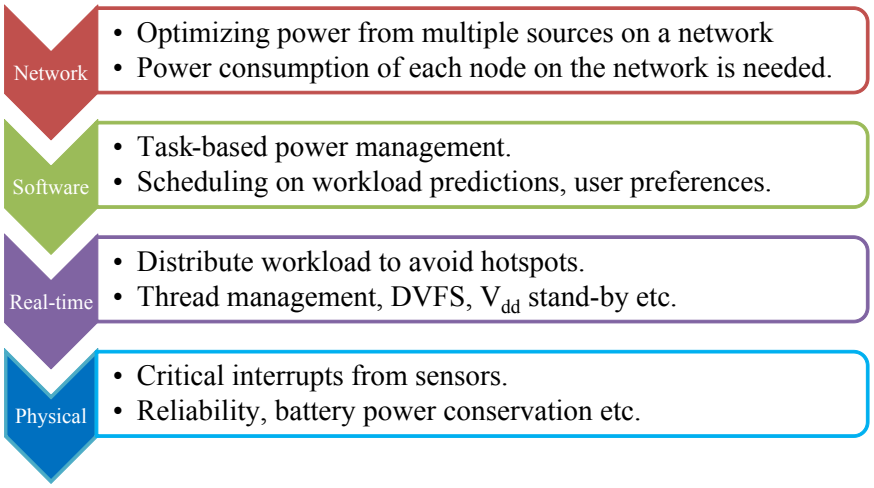


Fig. 8.2.: Cross-layer coordination for smarter power management.

9. CONTRIBUTIONS

Two on-chip power sensors with fast response times have been reported for the first time. Both the sensors can also be used as temperature sensors with same response times which are faster than any others reported thus far. Low area and power overheads of these sensors enable replication at multiple levels on a chip for fine-grained power management. A low area-overhead current sensor with low temperature coefficient has also been reported. Operating in weak-inversion, this sensor is inherently more tolerant to aging related defects and allows for a PVT and aging tolerant power estimate which paves the way for a smarter, fine-grained power management in both spatial and temporal domains for high-density systems like microprocessors.

Power sensor was modeled into a four cour processor environment and a EDP minimizing DVFS governor was demonstrated to have significantly improvements over hardware performance counters. This system was also used to demonstrated improved performances of PDP and ED^2P minimization governers.

A new algorithm was proposed using the readings from the proposed power sensor. In order to demonstrate the efficacy of these sensors and the proposed algorithm, a mulit-core system was fabricated in 45-nm SOI. Predictive thread migration significantly reduced the incidence of hotspots and thermal cycles, and also ensured a more equitable distribution of power dissipation in a multi-core environment, which results in a lower spatial variation of temperatures.

LIST OF REFERENCES

LIST OF REFERENCES

[1] J. Koomey, "Growth in data center electricity use 2005 to 2010", *CA: Analytics Press*, Aug. 2011.

[2] P. Somavat, S. Jadhav, and V. Namboodiri, "Accounting for energy consumption of personal computing including portable devices", *Proc. Int. Conf. on Energy-Efficient Computing and Networking*, April 2010.

[3] D. Singh and V. Tiwar, "Power challenges in the internet world", *Proc. 32nd Int. Microarchitecture Symp.*, pp. 8–15, Dec. 1999.

[4] S. Rusu et al., "ISSCC technology trends, microprocessors, 2011", *ISSCC Dig. Tech.*, Feb. 2011.

[5] H. V. Nguyen, *Multilevel interconnect reliability on the effects of electro-thermomechanical stresses*, PhD thesis, Univ. of Twente, Enschede, March 2004.

[6] R. Degraeve, J.L. Ogier, R. Bellens, P.J. Roussel, G. Groeseneken, and H.E. Maes, "A new model for the field dependence of intrinsic and extrinsic time-dependent dielectric breakdown", *IEEE Trans. Electron Devices*, vol. 45, no. 2, pp. 472–481, 1998.

[7] M. Huang, Z. Suo, Q. Ma, and H. Fujimoto, "Thin Film Cracking and Ratcheting Caused by Temperature Cycling", *J. Material Res.*, vol. 15, no. 06, pp. 1239–1242, 2000.

[8] T. S. Rosing, M. Kresimir, and G.De Micheli, "Power and reliability management of SoCs", *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 15, no. 4, pp. 391–403, 2007.

[9] Y. Zhang and D. Parikh, "Hotleakage: A temperature-aware model of subthreshold and gate leakage for architects", *Univ. of Virginia Dept. Comput. Science*, 2003.

[10] W. Liao, L. He, and K. M. Lepak, "Temperature and supply voltage aware performance and power modeling at microarchitecture level", *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 24, no. 7, pp. 1042–1053, 2005.

[11] Semiconductor Industry Association, "International technology roadmap for semiconductors (itrs), 2007 edition", Dec. 2007.

[12] E. Morifuji et al., "Supply and threshold-voltage trends for scaled logic and sram mosfets", *IEEE Transactions on Electron Devices*, vol. 53(6), pp. 1427–1432, June 2006.

[13] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low power cmos digital design", *Journal of Solid-State Circuits*, vol. 27, pp. 473–484, April 1992.

[14] R. Zahir, M. Ewert, and H. Seshadri, "The medfield smartphone: Intel architecture in a handheld form factor", *IEEE Micro*, vol. 33, no. 6, pp. 38–46, 2013.

[15] S. Kottapalli and J. Baxter, "Nehalem-ex cpu architecture", in *Hot chips*, 2009, vol. 21.

[16] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors", in *Int. Symp. High-Perf. Comput. Archit.* IEEE, 2001, pp. 171–182.

[17] M. Fleischmann, "Crusoe power management, reducing the operating power with longrun", in *Proc. Symp. Hot Chips*, 2000.

[18] J. Oh and M. Pedram, "Power reduction in microprocessor chips by gated clock routing", in *Design Automation Conference 1998. Proceedings of the ASP-DAC '98. Asia and South Pacific*, Feb 1998, pp. 313–318.

[19] K. Skadron, T. Abdelzaher, and M.R. Stan, "Control-theoretic techniques and thermal-rc modeling for accurate and localized dynamic thermal management", in *High-Performance Computer Architecture, 2002. Proceedings. Eighth International Symposium on*, Feb 2002, pp. 17–28.

[20] N. Bansal, T. Kimbrel, and K. Pruhs, "Dynamic speed scaling to manage energy and temperature", *45th Annu. IEEE Symp. Found. Comput. Sci.*, vol. 54, no. 1, 2004.

[21] S. Dahr, D. Maksimovic, and B. Kranzen, "Closed-loop adaptive controller for standard-cell ASICs", *Int. Symp. Low Power Electronics and Design*, pp. 103–107, 2002.

[22] S. Heo, K. Barr, and K. Asanovic, "Reducing power density through activity migration", in *Proc. Int. Symp. Low Power Electron. Des.*, 2003, pp. 217–222.

[23] K. K. Rangan, G. Wei, and D. Brooks, "Thread motion: fine-grained power management for multi-core systems", *ACM SIGARCH Comp. Arch.*, vol. 37, pp. 302–313, 2009.

[24] J. Lee, S. Bhagavatula, B. Jung, and K. Roy, "Self-healing design in deep scaled cmos technologies", *Proc. Midwest Symp. on Circuits and Systems*, pp. 1–4, 2011.

[25] W. Kim, M. S. Gupta, G. Y. Wei, and D. Brooks, "System level analysis of fast, per-core DVFS using on-chip switching regulators", *Proc. Int. Symp. High-Perf. Comput. Archit.*, pp. 123–134, 2008.

[26] S. Hoppner, H. Eisenreich, S. Henker, D. Walter, G. Ellguth, and R. Schuffny, "A compact clock generator for heterogeneous gals mpsocs in 65-nm cmos technology", *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 21, no. 3, pp. 566–570, March 2013.

[27] J. Flinn and M. Satyanarayanan, "Powerscope: A tool for profiling the energy usage of mobile applications", *IEEE Workshop Mobile Comp. Syst. App*, pp. 2–10, 1999.

[28] S. H. Shalmany, D. Draxelmayr, and K. A. Makinwa, "A micropower battery current sensor with $\pm 0.03\%$ (3σ) inaccuracy from -40 to $+85^o$c", *ISSCC Dig. Tech. Papers*, pp. 386–387,387a, Feb. 2013.

[29] K. Singh, M. Bhadauria, and S. A. McKee, "Real time power estimation and thread scheduling via performance counters", *ACM SIGARCH Comput. Archit. News*, vol. 37, pp. 46, 2009.

[30] R. Rodrigues, A. Annamalai, I. Koren, and S. Kundu, "A study on the use of performance counters to estimate power in microprocessors", *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 60, no. 12, pp. 882–886, 2013.

[31] Y. Sun, L. Wanner, and M. Srivastava, "Low-cost estimation of sub-system power", in *Proc. Int. Green Computing Conf.* IEEE, 2012, pp. 1–10.

[32] J. Shor, K. Luria, and D. Zilberman, "Ratiometric BJT-based thermal sensor in 32nm and 22nm technologies", *ISSCC Dig. Tech. Papers*, pp. 210–212, 2012.

[33] Y. W. Li et al., "A 1.05V 1.6mW 0.45^oC 3σ-resolution $\Sigma\Delta$-based temperature sensor with parasitic-resistance compensation in 32nm cmos", *ISSCC Dig. Tech. Papers*, pp. 340–341, 341a, 2009.

[34] H. Wang et al., "Runtime power estimator calibration for high-performance microprocessors", *Design Automation and Test in Europe*, pp. 352–357, 2012.

[35] J. Coburn, S. Ravi, and A. Raghunathan, "Hardware accelerated power estimation", *Design, Automation and Test in Europe*, pp. 528–529, 2009.

[36] D. Oh et al., "Runtime temperature-based power estimation for optimizing throughput of thermal-constrained multi-core processors", *Proc. ASP-DAC*, pp. 593–599, Jan. 2010.

[37] N. Gudino, M. J. Riffe, J. A. Heilman, and M. A. Griswold, "Hall effect current sensor", Feb. 2013, US Patent 8,378,683.

[38] J. Jang, D. F. Berdy, J. Lee, D. Peroulis, and B. Jung, "A wireless condition monitoring system powered by a sub-100/spl mu/w vibration energy harvester", *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 60, no. 4, pp. 1082–1093, 2013.

[39] V. De, "Fine-grained power management", *ISSCC Tech. Forum*, Feb. 2013.

[40] Advanced Micro Devices, "Quad-core opetron processor front die-view [online]".

[41] B. Baas, "Asynchronous array of simple processors [online]".

[42] S. Bhagavatula and B. Jung, "A low power real-time on-chip power sensor in 45nm SOI", *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, pp. 1577–1578, July 2012.

[43] M. T. Tan, J. Chang, and Y. C. Tong, "A process independent threshold voltage inverter-comparator for pulse width modulation applications", in *Int. Conf. on Electronics, Circuits and Systems*, 1999, pp. 1201–1204.

[44] R. Iijima and M. Takayanagi, "Experimental and theoretical analysis of factors causing asymmetrical temperature dependence of V_t in high-k metal gate CMOS with capped high-k techniques", *Int. Electron Devices Meeting*, pp. 1–4, 2008.

[45] K. Shi and D. Howard, "Sleep transistor design and implementation - simple concepts yet challenges to be optimum", *Int. Symp. VLSI Design, Automation and Test*, pp. 1–4, 2006.

[46] K.H. Chenf et al., "A time-to-digital converter using multi-phase-sampling and time amplifier for all digital phase-locked-loop", *Proc. IEEE Design and Diagnostics Electronic Circuits and Systems*, pp. 285–288, Apr. 2010.

[47] B. D. Yang, Y. K. Shin, J. S. Lee, Y. K. Lee, and K. C. Ryu, "An accurate current reference using temperature and process compensation current mirror", *Proc. IEEE Asian Solid-State Circuits Conf.*, pp. 241–244, 2009.

[48] Y. Osaki, T. Hirose, N. Kuroki, and M. Numa, "Nano-ampere CMOS current reference with little temperature dependence using small offset voltage", *Midwest Symp. Circuits Syst.*, , no. 2, pp. 668–671, 2010.

[49] A. Bendali and Y. Audet, "A 1-V CMOS current reference with temperature and process compensation", *IEEE Tran. Circuits and Systems. I Regular Papers*, vol. 54, no. 7, pp. 1424–1429, 2007.

[50] K. Ueno, T. Hirose, T. Asai, and Y. Amemiya, "A 1-μw 600-ppm/^{o}c current reference circuit consisting of subthreshold cmos circuits", *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 57, no. 9, pp. 681–685, 2010.

[51] J. Lee and S. Cho, "A 1.4-μW 24.9-ppm/°C current reference with process-insensitive temperature compensation in 0.18-μm CMOS", *IEEE J. Solid-State Circuits*, vol. 47, no. 10, pp. 2527–2533, 2012.

[52] G. Serrano and P. Hasler, "A precision low-TC wide-range CMOS current reference", *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 558–565, 2008.

[53] S. Bhagavatula and B. Jung, "A power sensor with 80ns response time for power management in microprocessors", *Proc. Custom Integrated Circuits Conf.*, pp. 1–4, Sept. 2013.

[54] B. C. Paul, K. Kang, H. Kufluoglu, M. A. Alam, and K. Roy, "Impact of NBTI on the temporal performance degradation of digital circuits", *Electron Device Letters*, vol. 26, no. 8, pp. 560–562, 2005.

[55] R. Degraeve, M. Aoulaiche, B. Kaczer, P. Roussel, T. Kauerauf, S. A. Sahhaf, and G. Groeseneken, "Review of reliability issues in high-k/metal gate stacks", in *Proc. Int. Symp. Physical and Failure Analysis of Integrated Circuits*. IEEE, 2008, pp. 1–6.

[56] K. Souri, Y. Chae, and K. Makinwa, "A CMOS temperature sensor with a voltage-calibrated inaccuracy of $\pm 0.15^o$C (3σ) from -55 to 125°C", *ISSCC Dig. Tech. Papers*, pp. 208–210, 2012.

[57] C. Wu, W. Chan, and T. Lin, "A 80ks/s 36μw resistor-based temperature sensor using bgr-free sar adc with a unevenly-weighted resistor string in 0.18μm cmos", *Symp. On VLSI Circuits*, pp. 222–223, June 2011.

[58] K. Woo et al., "Dual DLL-based cmos all-digital temperature sensor for microprocessor thermal monitoring", *ISSCC Dig. Tech. Papers*, pp. 68–69, 69a, 2009.

[59] R. Jejurikar, C. Pereira, and R. Gupta, "Leakage aware dynamic voltage scaling for real-time embedded systems", *Proc. Design Automation Conf.*, p. 275, 2004.

[60] S. Kaxiras and M. Martonosi, "Computer architecture techniques for power-efficiency", *Synthesis Lectures on Computer Architecture*, vol. 3, no. 1, pp. 1–207, 2008.

[61] X. Ruan, X. Qin, Z. Zong, K. Bellam, and M. Nijim, "An Energy-Efficient Scheduling Algorithm Using Dynamic Voltage Scaling for Parallel Applications on Clusters", *Proc. Int. Conf. Comput. Commun. Networks*, pp. 1–6, 2007.

[62] J. Zhuo and C.i Chakrabarti, "Energy-efficient dynamic task scheduling algorithms for DVS systems", *ACM Trans. Embed. Comput. Syst.*, vol. 7, no. 2, pp. 1–25, 2008.

[63] M. Bao, A. Andrei, P. Eles, and Z. Peng, "Temperature-aware voltage selection for energy optimization", *Proc. Design, Autom. Test Eur. DATE*, , no. 1, pp. 1083–1086, 2008.

[64] K. Choi, R. Soma, and M. Pedram, "Fine-grained dynamic voltage and frequency scaling for precise energy and performance tradeoff based on the ratio of off-chip access to on-chip computation times", *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 24, pp. 18–28, 2005.

[65] C. Isci, G. Contreras, and M. Martonosi, "Live , Runtime Phase Monitoring and Prediction on Real Systems with Application to Dynamic Power Management tory Table Predictor", *Electr. Eng.*, 2006.

[66] C. Isci and M. Martonosi, "Phase characterization for power: Evaluating control-flow-based and event-counter-based techniques", *Proc. - Int. Symp. High-Performance Comp. Archit.*, vol. 2006, pp. 122–133, 2006.

[67] T. Sherwood, S. Sair, and B. Calder, "Phase tracking and prediction", *Proc. Int. Symp. Comput. Archit.*, vol. 00, no. c, 2003.

[68] F. Vandeputte, L. Eeckhout, and K. D. Bosschere, "A detailed study on phase predictors", *Int. Euro-Par Conf. Parallel Processing*, 2005.

[69] S. Eyerman and L. Eeckhout, "A counter architecture for online DVFS profitability estimation", *IEEE Trans. Comput.*, vol. 59, no. 11, pp. 1576–1583, 2010.

[70] G. Keramidas, V. Spiliopoulos, and S. Kaxiras, "Interval-based models for runtime DVFS orchestration in superscalar processors", *Proc. ACM Int. Conf. Comput. Front*, p. 287, 2010.

[71] R. Miftakhutdinov, E. Ebrahimi, and Y.N. Patt, "Predicting performance impact of dvfs for realistic memory systems", in *Proc. IEEE Int. Symp. Microarchitecture*, Dec 2012, pp. 155–165.

[72] V. Spiliopoulos, S. Kaxiras, and G. Keramidas, "Green governors: A framework for continuously adaptive DVFS", *2011 Int. Green Comput. Conf. Work. IGCC 2011*, 2011.

[73] R. Joseph and M. Martonosi, "Run-time Power Estimation in High Performance Microprocessors", in *Int. Symp. Low Power Electron. Des.*, 2001, pp. 135–140.

[74] G. Contreras and M. Martonosi, "Power prediction for Intel XScale® processors using performance monitoring unit events", *ISLPED '05. Proc. 2005 Int. Symp. Low Power Electron. Des. 2005.*, pp. 0–5, 2005.

[75] K. Rajamani, H. Hanson, J. Rubio, S. Ghiasi, and F. Rawson, "Application-aware power management", *Proc. IEEE Int. Symp. Workload Charact.*, pp. 39–48, 2006.

[76] K. J. Lee and K. Skadron, "Using performance counters for runtime temperature sensing in high-performance processors", *Proc. - 19th IEEE Int. Parallel Distrib. Process. Symp. IPDPS 2005*, vol. 2005, 2005.

[77] B. Goel, S.A McKee, R. Gioiosa, K. Singh, M. Bhadauria, and M. Cesati, "Portable, scalable, per-core power estimation for intelligent resource management", *Int. Conf. Green Comput.*, pp. 135–146, 2010.

[78] P. Chaparro, J. González, G. Magklis, Q. Cai, and A. González, "Understanding the thermal implications of multi-core architectures", *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 8, pp. 1055–1065, 2007.

[79] Y. Ge, *Dynamic Thermal Management for Microprocessors*, PhD thesis, Syracuse University, 2012.

[80] T. E. Carlson, W. Heirman, S. Eyerman, I. Hur, and Lieven Eeckhout, "An evaluation of high-level mechanistic core models", *ACM Trans. Archit. and Code Optimization*, 2014.

[81] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures", in *Proc. Int. Symp. Microarch.* IEEE, 2009, pp. 469–480.

[82] R. Singhal, "Inside intel next generation nehalem microarchitecture", in *Hot Chips*, 2008, vol. 20.

[83] M. D. Powell and T. N. Vijaykumar, "Heat-and-run: leveraging SMT and CMP to manage power density through the operating system", *Archit. Support Program. Lang. Oper. Syst.*, , no. Asplos, pp. 260–270, 2004.

[84] A. Coskun, T. S. Rosing, K.A. Whisnant, and K.C. Gross, "Static and dynamic temperature-aware scheduling for multiprocessor SoCs", *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 16, no. 9, pp. 1127–1140, 2008.

[85] P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: A 32-way multi-threaded sparc processor", *Micro, IEEE*, vol. 25, no. 2, pp. 21–29, 2005.

[86] A. Kumar, L. Shang, L.-S. Peh, and N. K. Jha, "Hybdtm: a coordinated hardware-software approach for dynamic thermal management", in *Proc. Des. Automation Conf.* ACM, 2006, pp. 548–553.

[87] I. Yeo, C. Liu, and E. J. Kim, "Predictive dynamic thermal management for multicore systems", *Proc. Des. Autom. Conf.*, pp. 734 – 739, 2008.

[88] A. K. Coskun, T.S Rosing, and K. C. Gross, "Proactive temperature balancing for low cost thermal management in MPSoCs", *IEEE/ACM Int. Conf. Comput. Des. Dig. Tech. Pap. ICCAD*, pp. 250–257, 2008.

[89] Y Liu, H Yang, R. P. Dick, H. Wang, and L. Shang, "Thermal vs energy optimization for DVFS-enabled processors in embedded systems", *Proc. Int. Symp. Qual. Electron. Des.*, pp. 204–209, 2007.

[90] R. Cochran and S. Reda, "Consistent runtime thermal prediction and control through workload phase detection", *Proc. Des. Autom. Conf.*, pp. 62–67, 2010.

[91] Osman Sarood, Phil Miller, and Ehsan Totoni, " Cool Load Balancing for High Performance Computing Data Centers", pp. 1–14.

[92] A. K. Coskun, T. S. Rosing, and K. C. Gross, "Utilizing predictors for efficient thermal management in multiprocessor SoCs", *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 28, no. 10, pp. 1503–1516, 2009.

[93] J. D. Regehr, *Using hierarchical scheduling to support soft real-time applications in general-purpose operating systems*, PhD thesis, University of Virginia, 2001.

[94] B. Salami, M. Baharani, H. Noori, and F. Mehdipour, "Physical-Aware Task Migration Algorithm for Dynamic Thermal Management of SMT Multi-core Processors", pp. 292–297, 2014.

[95] I. Yeo and E.J. Kim, "Temperature-aware scheduler based on thermal behavior grouping in multicore systems", *Des. Autom. Test Eur. Conf. Exhib.*, 2009.

[96] Z. Liu, T. Xu, S. X. D. Tan, and H. Wang, "Dynamic thermal management for multi-core microprocessors considering transient thermal effects", *Proc. Asia South Pacific Des. Autom. Conf.*, , no. 2, pp. 473–478, 2013.

[97] K. Skadron, T. Abdelzaher, and M. R. Stan, "Control-theoretic techniques and thermal-rc modeling for accurate and localized dynamic thermal management", in *Proc. Int. Symp. High-Performance Comput. Archit.* IEEE, 2002, pp. 17–28.

[98] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture", in *ACM SIGARCH Comput. Archit. News.* ACM, 2003, vol. 31, pp. 2–13.

[99] K. Niitsu, M. Sakurai, N. Harigai, T. J. Yamaguchi, and H. Kobayashi, "CMOS circuits to measure timing jitter using a self-referenced clock and a cascaded time difference amplifier with duty-cycle compensation", *IEEE J. Solid-State Circuits*, vol. 47, pp. 2701–2710, 2012.

[100] T. Hashimoto, H. Yamazaki, A. Muramatsu, T. Sato, and A. Inoue, "Time-to-digital converter with vernier delay mismatch compensation for high resolution on-die clock jitter measurement", *IEEE Symp. VLSI Circuits, Dig. Tech. Pap.*, pp. 156–157, 2008.

[101] S. Tabatabaei and a. Ivanov, "Embedded timing analysis: a soc infrastructure", *IEEE Des. Test Comput.*, vol. 19, pp. 24–36, 2002.

[102] P. Chaparro and J. Gonzalez, "Understanding the thermal implications of multi-core architectures", *IEEE Tran. Parallel and distributed systems*, pp. 1055–1065, Aug. 2007.

[103] K. Kant, "Towards a science of power management", *Computer*, pp. 99–101, Sept. 2009.

[104] Y. H. Lu, Q. Qiu, A. R. Butt, and K. W. Cameron, "End-to-end energy management", *Computer*, pp. 75–77, Nov. 2011.

VITA

VITA

Srikar Bhagavatula graduated from the class of 2006 with a B. Tech in Electrical Engineering from Indian Institute of Technology, Bombay. Upon graduation, he joined Cypress semiconductors, Bangalore, India as a Design Engineer and then moved to IHP Microelectronics, Frankfurt (oder), Germany as a Research Engineer in 2008. He began his Ph.D program in 2009 Spring at Purdue University where his research focused on variation tolerant circuit design, sensing and power management techniques. Since 2014, he has been with Entropic Communications working on high speed ADCs and sensors.