IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 5, Issue 3 (Mar. - Apr. 2013), PP 01-07
www.iosrjournals.org

Supply Insensitivity Temperature Sensor for Microprocessor Thermal Monitoring Using Adc-Sar

N.Elakkiya[1], Ms. S.Sindhu Meenakshi [2]
[1]PG Scholar., Dept. of Electronics and Communication Engineering, Velalar College of Engineering and Technology., [2]Asst.Professor., Dept. of Electronics and Communication Engineering., Velalar College of Engineering and Technology.

Abstract: *In this project, temperature sensing in microprocessors and other digital integrated circuits is essential to prevent thermal overrun and guarantee processor performance. Increased attention to energy-efficient computing is pushing power management systems to become dynamic and localized to processor sub-systems, to balance power and performance requirements. In this paper we analyze CMOS temperature sensors that work by measuring temperature-dependent delays in CMOS inverters with two new features distinguish this work from the prior delay-based temperature sensors. First, our sensor operates with simple, low-cost one-point calibration. Second, it uses delay-locked loops (DLLs) to convert inverter delays to digital temperature outputs: the use of DLLs enables low energy (0.24 J/sample) and high bandwidth (5 kilo-samples/s), facilitating fast thermal monitoring. After calibration, measurement errors for 15 chips fabricated in digital CMOS 0.13 m fall -4.0^{o}C to 4.0^{o}C within C in a temperature range of 0-100^{o}C, where the temperature chamber used has a control uncertainty of+1.1^{o}C or -1.1^{o}C. ADC-SAR is used for fast thermal monitoring. Simulations show the sensor to have a resolution of approximately +.5℃ with each conversion taking approximately a minimum of 550 µs and a maximum of 1.05ms*

Keywords - *CMOS, delay locked loops (DLLs), microprocessors, temperature sensor, ADC-SAR.*

I. Introduction

Since the advent of the very first microprocessor, the transistors that make them have continued to shrink in size exponentially. This scaling has consistently kept up with or even surpassed the rate of growth predicted by Gordon Moore's empirical law stated in 1965. In comparison with the early microprocessors, today's processors are faster and cheaper by several orders of magnitude owing to this exponential growth in transistor density. However, as process technology scales into the nanometer region, several hindrances to its continuation emerge. Low-level effects which were previously considered second-order and have traditionally been largely invisible to computer architects have surfaced to become primary concerns. Processor temperature is one such concern that has arguably become one of the hardest obstacles to continued technology scaling. Most of the energy consumed by a microprocessor is dissipated as heat due to the resistive behavior of the processor circuits. The temperature of a chip, which is a measure of the amount of heat energy stored in it, is directly related to the power density (*i.e.*, the power consumed per unit area of the chip). In order to see the impact of scaling on temperature, it is illuminating to study the relationship between scaling and power density. Scaling theory provides a simple way to reason about such a relationship between the device parameters (such as transistor length, supply voltage *etc.*) of successive technology generations. Every successive process generation shrinks the length of the transistor by a constant fraction of the previous length. This fraction is by a called the scaling factor (say k) and is typically $\approx 1/\sqrt{2}$. Hence, the area of the transistor scales proportional to $k2$, *i.e.*, approximately halving every successive generation.

In order to maintain the performance under increasing leakage, it has not been able to scale at all! This has resulted in processor power almost remaining constant in moving to a newer process generation. More interestingly, the power density increases approximately by a factor of $k2$ every generation! This exponential increase in power density, if left unmanaged, would result in an exponential increase of temperature every successive generation. Since that cannot be allowed (otherwise, the chip would melt!), huge efforts have been put into the removal of heat away from the die-surface of a microprocessor.

This has led to expensive packages (heat sinks, heat pipes *etc.*) and as a corollary, exponential increase in the cost of such cooling solutions. Increase in die temperature is undesirable for a variety of reasons. Catastrophic failure such as the melting of the chip is a possibility, albeit a distant one. In reality, more pertinent reasons are in the form of increased leakage power and accelerated aging that reduces lifetime reliability. Transistors consume power even when they are idle and not switching. This is called static power or leakage power. Even under nominal conditions, it can be a significant fraction (> 30%) of the total power consumption at current feature sizes. It varies exponentially with respect to temperature. As temperature itself depends on the power consumption, there is a circular dependence between them. In extreme cases, this can result in a self-

reinforcing positive feedback loop that leads to thermal runaway. Temperature also has an exponentially adverse effect on the expected lifetime of a microprocessor The Mean Time to Failure (MTTF) of a chip can be empirically described using the Arrhenius
Equation given by:

$$\text{MTTF} = Ae^{-Ea/KT}$$

Here, *T* is the temperature, *A* is an empirical constant and *Ea* is the activation energy of the failure mechanism. It is to be noted that this equation does not capture all the effects of temperature (like thermal cycling, thermal gradients *etc.*) on reliability. However, it is a useful expression for first-order estimation. It actually comes from Chemistry where the rate of a chemical reaction is expressed as an exponential function of the temperature of operation. The rate of reaction or equivalently, the rate of failure in case of silicon, is highly accelerated at higher temperatures.

II. Previous Work

A new type of integrated temperature sensors that exploit the temperature dependence of time delays of CMOS inverters. A time-to-digital converter (TDC) compares the temperature-dependent delay of an inverter chain to a temperature-independent delay reference (e.g., crystal oscillator) to produce a digital temperature output.

The inverter delay, delay reference, and TDC are analogous to the base-emitter voltage, band gap reference, and ADC of the conventional sensor. Needing improvements in various aspects, whether the delay based sensors can be a viable practical option remains to be seen, subject to further study. They can be potentially well suited for microprocessor thermal profiling, as they are more digital and do not need parasitic bipolar transistors. Here we report a CMOS delay-based temperature sensor intended for the microprocessor thermal profiling. It builds up from the original delay-based CMOS temperature sensor, with the following two new key features.

1) Since the inverter delay varies not only with temperature but also with process variation, the temperature dependence of the delay differs from sensor to sensor. Therefore, each sensor needs calibration according to its process variation. The prior delay-based sensors measure delays at two known temperature points to calibrate out process variations. By contrast, our sensor operates with simple one-point calibration, which is the first key feature of our work.

The one-point calibration may not produce as accurate temperature measurements as the two-point calibration, but is simpler, thus, reduces high-volume production cost. This choice of trade-off is meaningful in the microprocessor application, where high precision is not required but lower calibration cost is desirable. The latter is because temperature sensors play auxiliary, albeit important, roles without taking part in main computing activity. The one-point calibration is enabled by our observation that the inverter delay can be separated into a function of temperature only and a function of process only.

2) The prior delay-based temperature sensors measure inverter delays using a counter-based cyclic TDC and a single temperature-independent delay reference. The second key feature of our work is the architectural modification by using two delay-locked loops (DLLs). One DLL synthesizes multiple temperature-independent delay references; the other DLL serves as a TDC, and compares temperature-dependent inverter delays to the multiple delay references synthesized by the first DLL. Crystal oscillators required by DLLs are readily available in microprocessor environment. The use of multiple delay references via DLLs leads to a high bandwidth (5 kilo-samples/s) at 7-bit resolution, which can facilitate tracking of fast thermal transients. This work was partially reported in [25].

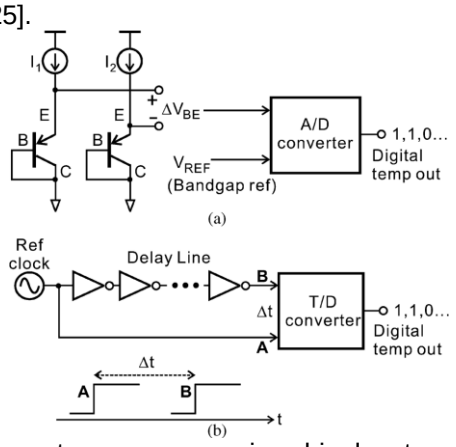


Fig. 1. (a) Conventional temperature sensor using bipolar transistors.(b) Inverter-delay-based temperature sensor

This paper is a significant expansion with new substantial experiments with an increased number of sensor chips (5 chips versus 15 chips)and new analyses. Sections II and III describe in details the two new features of our sensor, thus, its operating principles. Section IV presents design implementation. Section V reports measurements of fabricated CMOS sensors to validate the principle, and compares the performance of our sensor to that of the prior delay-based temperature sensors.

Delay-based CMOS temperature sensors are prone to high sensitivity to static supply shift [18]–[21]. Our design here also does not overcome this issue. We explicitly measure and quantify the sensitivity of our sensor to static supply shift to motivate further studies. This is presented in Section VI.

III. Proposed Method

A. OPERATING PRINCIPLE (1)—ONE-POINT CALIBRATION

1) Inverter Delay and Separation of Variables

Separation of the CMOS inverter delay into a function of only temperature and a function of only process parameters is a key to our one-point calibration. Here we establish this separation of variables. Consider a CMOS inverter with equivalent pMOS and nMOS strengths. The propagation delay, through the CMOS inverter may be expressed as [26]

$$D = \frac{L}{W}\frac{C_L}{C_{\text{ox}}} \cdot \frac{1}{\mu} \cdot \frac{\ln\{3 - 4V_{\text{th}}/V_{\text{dd}}\}}{V_{\text{dd}}(1 - V_{\text{th}}/V_{\text{dd}})}.$$

C_L is the load capacitance and V_{DD} is the supply. L and W are the gate length and width, μ is the C_L is the load capacitance and V_{DD} is the supply. L and W are the gate length and width, μ is the mobility, C_{OX} is the gate oxide capacitance per unit area, and V_{th} is the threshold voltage, all for the nMOS transistor.

2) Principle of One-Point Calibration

Since the inverter delay depends on both temperature and process variation, the effect of the process variation should be

calibrated out, in order to extract temperature information from a measured delay. The separation of variables in the inverter delay enables calibration with a delay measurement only at on known temperature (one-point calibration). We first set temperature at $T=T_C$ (subscript " c" denotes "calibration") and measure the inverter delay $D(T_c,P)$

$$D(T_c,P) = F(T_c,P).G(P).$$

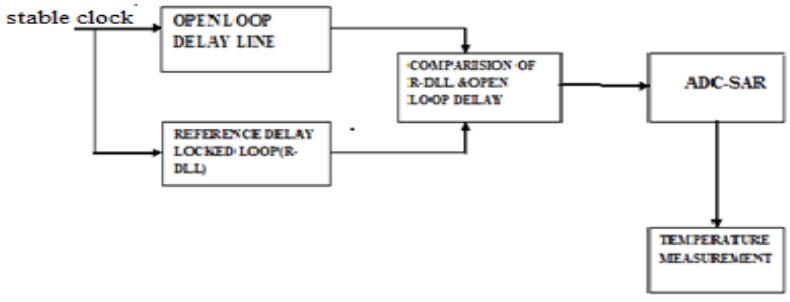


Fig 2: CMOS temperature sensor using ADC- SAR

C. OPERATING PRINCIPLE (2)—DUAL-DLL-BASED DELAY MEASUREMENT

A DLL is essentially a nonlinear negative feedback system. However, it is a common practice to characterize a DLL by linear analysis. Although linear analysis is not able to produce a very accurate result, it can still serve as a reasonable first-order approximation and can lead to some useful insights into a DLL.s operation. In a DLL, the input clock signal propagates through the VCDL and develops phase shift (or time delay) at every delay stage of the VCDL. The phase shift of each delay stage is controlled by the voltage of a loop filter. The output is taken from one of the delay stages.

The phase of the output signal is compared with the phase of the input clock in the PD. The phase error information generated by the PD (usually in the form of a voltage or a current) is then transferred to the CP. The CP uses the phase error information to adjust the voltage of the loop filter and thus to change the delay of the VCDL. Owing to such a negative feedback mechanism, the phase error is gradually reduced until it finally becomes zero. At that time, the delay of the whole VCDL line becomes equal to one clock period, and the voltage of the loop filter is stabilized, which indicates that a locked state has been established.

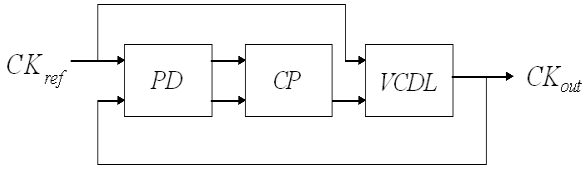


Figure 3. Simplified block diagram of a DLL

D. DESIGN AND IMPLEMENTATION

The implemented sensor is schematically shown in Fig. 5, which is a detailed version of Fig. 3. The R-DLL, shown in the lower portion, consists of a voltage-controlled delay line (VCDL), a phase detector, and a charge-pump in a closed loop. The VCDL is an even-numbered cascade of current-starved inverters. The open-loop delay line, at top of Fig. 5, is an even numbered cascade of CMOS inverters. The R-DLL is locked to a 30 MHz stable clock x(t) , which also drives the open-loop delay line. The period P_0 of x(t) is $P_0 \approx 33.3$ ns. In the R-DLL (see Fig. 5), the VCDL contains a total of 200 delay buffers. 32 of them (164^{th}~ 195th) are connected to MUX-1. The input and output phases of a buffer selected by MUX-1 serve as two input phases for the tristate-inverter-based phase interpolator [34], [35] consisting of 16 inverters. One of the two phases from MUX-1 is tapped to the inputs of 8 inverters; the other phase from MUX-1 is tapped to the inputs of the remaining 8 inverters. The 16 inverters are all tied at output. A 16-bit control input (an 8-bit enable signal and its negation) turns on only 8 inverters proper. By varying the combination of the 8 inverters that are turned on, the output phase can be interpolated at 7 different positions between the two input phases. In this way, the interpolator generates 7 additional phases between its two input phases. This arrangement is to achieve a measurement range of C and a sub- measurement.

E. DLL FOR BIST APPLICATIONS

The test of digital integrated circuits often involves expensive test equipment in order to obtain the capability of precise timing control. To avoid the cost of using expensive test equipment, we propose to use a DLL-based BIST circuit to generate the necessary delays on chip. For example, in the testing of setup and hold time, a reference clock and a series of digital test data are supplied to the chip I/O pins concurrently. The reference clock drives the DLL to generate two new clock signals on chip, which are the delayed versions of the reference clock. With the help of the DLL, the delays are set to be aligned with the setup and hold time specifications. Specifically, the setup time test clock is produced by delaying the input clock by the setup time specification. The hold time test clock is generated by delaying the input clock by one clock cycle minus the specified hold time. The newly generated clock signals and the input test data become the inputs to the input registers. The outputs of the input registers are read out and compared with the input test data. If the output data matches the input test data, the register is considered to function correctly and the chip is considered to pass the setup or hold time test. Otherwise, the chip is considered to fail the setup or hold time test.

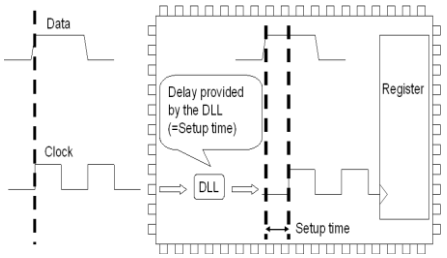


Figure 4. Proposed Test Of Setup Time

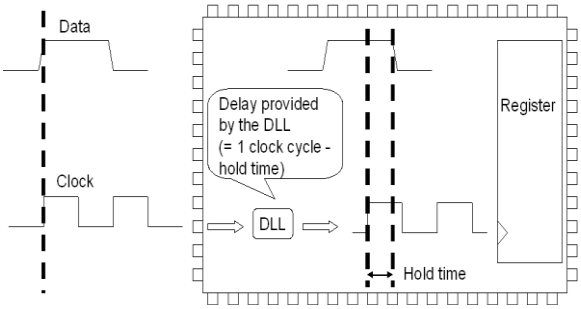


Figure.5. proposed test of Hold Time

F. DLL ARCHITECTURE DESIGN

The first design decision that must be made is the architecture of the DLL. As discussed before, there are two major options: single-loop architecture and dual-loop architecture. Although the dual-loop structures provide some advantages over single loop DLLs, they inevitably introduce more power and area consumption. More importantly, dual-loop DLLs bring in more design complexity. On the other hand, single-loop DLLs can still provide acceptable jitter performance, and by redesigning the individual components, they can also achieve wide lock range and equal cell delays. Based on the above considerations, single-loop architecture is chosen for the proposed DLL.

G. DLL COMPONENT DESIGN

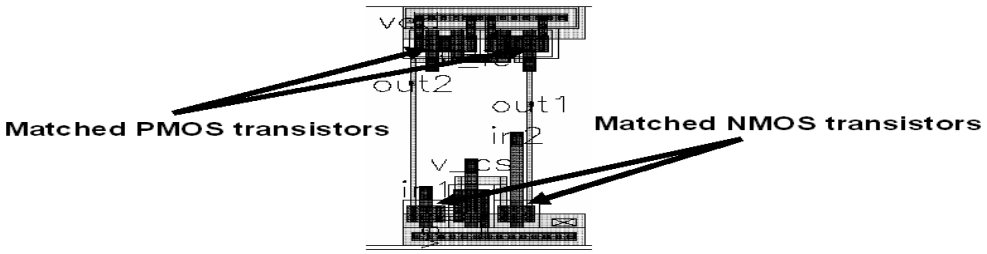
Major redesigns have been done to improve the performance of the DLL. For example, the proposed DLL uses a combined PD and CP circuit for increased speed and reduced jitter. The proposed DLL also employs an eight-stage shift averaging VCDL to improve the matching between delay stages and thus to equalize the delay of each individual stage. With the help of the above techniques, the proposed DLL is able to achieve lock as long as the minimum VCDL delay (min) *VCDL D* is less than one reference clock cycle *REF T* . This is the *largest possible* lock range that can be realized by the DLL, since it would be impossible for the DLL to achieve lock if 3 8 (min) *VCDL D* were larger than *REF T* . In the next section, a detailed discussion of the design of the BIST circuit will be presented.

H. LAYOUT OF DLL

The layout of an integrated circuit defines the geometries that appear on the masks used in fabrication. Careful layout helps to achieve the design goals. In this section, we will discuss some of the techniques we used in the layout of the prototype chip.

I. IMPROVING MATCHING

Matching is an extremely important issue for analog integrated circuits to achieve the desired performance. Significant mismatch can be very harmful for some sensitive circuit blocks such as differential pairs or current mirrors. In the proposed DLL, the output clocks are directly taken from the delay stages of the VCDL and each delay stage is based on a differential pair. Thus, the matching between the VCDL delay stages and the matching of the transistors within each delay stage are crucial for the proper operation of the DLL. In the layout process, all possible precautions have been taken to minimize mismatch. Generally, mismatch can be divided into two categories: random mismatch and systematic mismatch. Random mismatch stems from microscopic fluctuations in dimensions, doping levels, oxide thicknesses, and other parameters that influence component values. Random mismatch cannot be completely eliminated as it is intrinsic to the fabrication process. However, its impact on circuit performance can be greatly reduced through proper selection of the component values and device dimensions. In contrast, systematic mismatch originates from process biases, contact resistances, no uniform current flow, diffusion interactions, mechanical stresses, temperature gradients, and a host of other causes. In the layout of analog circuits, maximum design efforts must be made so that critical circuit components are not sensitive to systematic mismatch. Typically, transistors of different widths and lengths do not match very well. Therefore, a uniform channel length the minimum channel length is used for all the transistors in the layout. Moreover, transistors that need be matched in each delay stage are divided into multiple fingers with all fingers being of the same width and length. Identical finger length and width can reduce systematic mismatch of a specific fabrication run. A transistor with multiple fingers also shows reduced junction area and gate resistance



.

Figure 6. Layout of a single delay stage

Symmetry is another important factor that affects matching. In the layout of the prototype chip, fully differential signaling is used to reduce the impact of common-mode noise and even-order nonlinearity. The circuit blocks that involve differential signals are placed in a symmetrical position to improve matching.

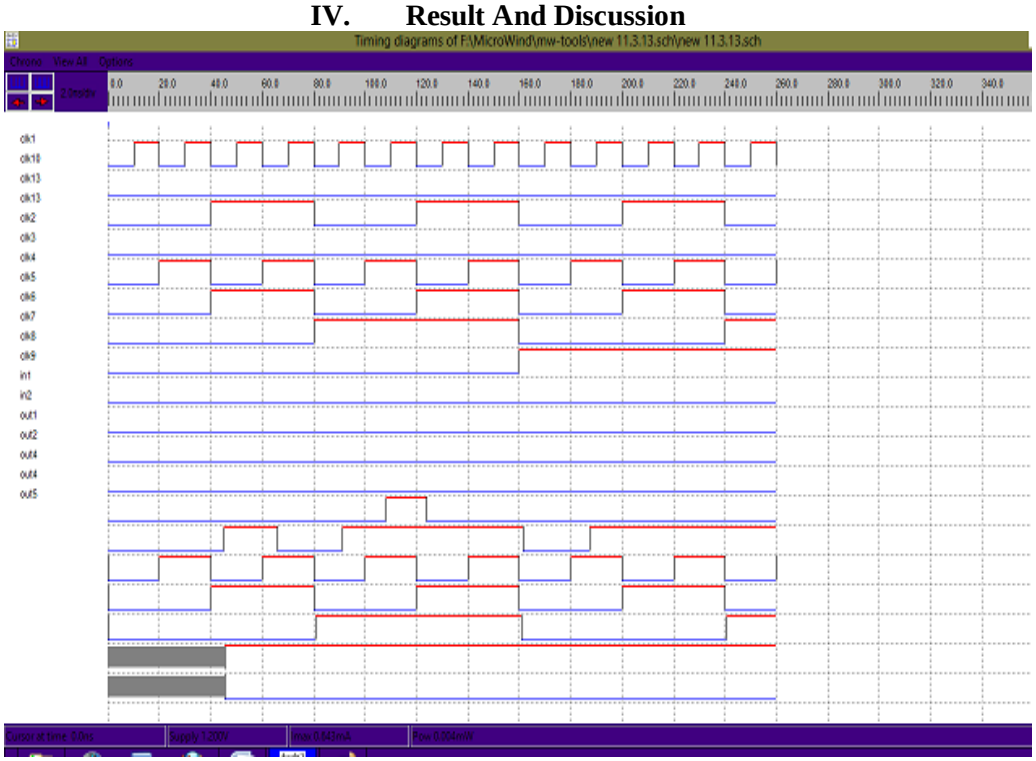
J. PERFORMANCE COMPARISON TO PRIOR DELAY-BASED SENSORS

Table I compares this work to delay-based temperature sensors with two-point calibration [18]–[22]. For completeness, the table also includes delay-based temperature sensors with one point calibration [23], [24], which appeared after the present work was originally reported in [25]. First, our sensor tends towards a larger error than other works, e.g., [18]–[22] with two-point calibration, which is due to the one-point calibration we use. Nonetheless, our accuracy is deemed sufficient for microprocessor thermal profiling [5], [6], [31]–[33]. Additionally, the one-point calibration has the advantage of simplicity and low-cost, which is a desirable feature in the microprocessor application (see Sections I and II). Second, our sensor using the dual-DLL architecture with multiple delay references achieve a high bandwidth of 5 kilo-samples/s, as we compare the effective sampling rates of the sensors at which errors were actually measured.

TABLE:1
COMPARISON TO OTHER DELAY-BASED TEMPERATURE SENSORS

Authors' Name	**Method Proposed**	**Advantages**	**Drawbacks**
P. Chen and W.-F. Lu	Two point calibration	High accuracy	Increased cost, energy consumption is high.
T.-K. Chen, and S.-W. Chen	Two point calibration	High accuracy	Cost is high, increased area.

IV. Result And Discussion



Compared to prior based temperature sensor the delay has been reduced and fast thermal monitoring has been obtained, but the static supply shift in cmos temperature sensor is the major problem that could be reduced in future study.

.

References

[1] R. McGowen, C. Poirier, C. Bostak, J. Ignowski, M. Millican, W. H.Parks, and S. Naffziger, "Power and temperature control on a 90 nmItanium-family processor," *IEEE J. Solid-State Circuits*, vol. 41, no. 1,pp. 229–237, Jan. 2006.

[2] J. Clabes, J. Friedrich, M. Sweet, J. Dilullo, S. Chu, D. Plass, J.Dawson, P. Muench, L. Powell, M. Floyd, B.Sinharoy, M. Lee, M.Goulet, J. Wagoner, N. Schwartz, S. Runyon, G. Gorman, P. Restle, R. Kalla, J. McGill, and S. Dodson, "Design and implementation of the Power5 microprocessor," in *IEEE Int. Solid-State Circuits Conf.Dig. Tech. Papers*, 2004, pp. 56–57.

[3] E. Rotem, A. Naveh, M. Moffie, and A. Mendelson, "Analysis of thermal monitor features of the Intel Pentium processor," in *Proc.Workshop Temperature-Aware Comput. Syst.*, 2004, pp. 29–35.S. O. Memik, R. Mukherjee, M. Ni, and J. Long, "Optimizing thermal sensor allocation for microprocessors," *IEEE Trans.Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 3, pp.516–527, Mar. 2008.

[4] Y. Li, H. Lakdawala, A. Raychowdhury, G. Taylor, and K.Soumyanath, "A 1.05 V 1.6 mW 0.45 C -resolution –based temperature sensor with parasitic-resistance compensation in 32 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*,2009, pp. 340–341.E.

[5] H. F. Hamann, A. Weger, J. A. Lacey, Z. Hu, P. Bose, E. Cohen, and J. Wakil, "Hotspot-limited microprocessors: Direct temperature and power distribution measurements," *IEEE J. Solid-State Circuits*, vol.42, no. 1, pp. 50–65, Feb. 2007.

[6] R. C. Dobkin, "Monolithic temperature transducer," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 1974, pp. 126–127.

[7] T. Verster, " junction as an ultralinear calculable thermometer,"*Electron. Lett.*, vol. 4, no. 9, pp. 175–176, May 1968.

[8] M. A. P. Pertijs,K.A.A.Makinwa, and J.H.Huijsing, "ACMOS smart temperature sensor with a inaccuracy of from C to 125 C," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2805–2815,Dec. 2005.

[9] Bakker, "CMOS smart temperature sensors—An overview," *Proc. IEEE Sensors*, vol. 2, pp. 1423–1427, Jun. 2002.

[10] M. A. P. Pertijs, A. Niederkorn, X.Ma, B.McKillop, A. Bakker, and J.H. Huijsing, "A CMOS smart temperature sensor with a inaccuracy of C from C to 120 C," *IEEE J. Solid-State Circ.*, vol.40, no. 2, pp. 454–461,Feb.2005.

[11] M. A. P. Pertijs, C. M. Meijer, and J. H. Huijsing, "Precision temperature measurement using CMOS substrate PNP transistors," *IEEE SensorsJ.*, vol. 4, no. 3, pp. 294–300, Jun. 2004.

[12] Bakker and J. H. Huijsing, "Micropower CMOS temperature sensor with digital output," *IEEE J. Solid-State Circuits*, vol. 31, no. 7, pp. 933–937, Jul. 1996.

[13] M. A. P. Pertijs, A. Bakker, and J. H. Huijsing, "A high-accuracy temperature sensor with second-order curvature correcton and digital bus interface," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2001, pp.368–371.

[14] M. A. P. Pertijs, C. M. Meijer, and J. H. Huijsing, "Precision temperature measurement using CMOS substrate PNP transistors," IEEE SensorsJ., vol. 4, no. 3, pp. 294–300, Jun.2004.

[15] Bakker and J. H. Huijsing, "Micropower CMOS temperature sensor with digital output," IEEE J. Solid-State Circuits, vol. 31, no. 7, pp. 933–937, Jul. 1996.

[16] M. A. P. Pertijs, A. Bakker, and J. H. Huijsing, "A high-accuracy temperature sensor with second-order curvature correcton and digital bus interface," in Proc. IEEE Int. Symp. Circuits Syst., 2001, pp. 368–371.

[17] D. E. Duarte, G. Geannopoulous, U. Mughal, K. Wong, and G.Taylor, "Temperature sensor design in a high volume manufacturing 65 nm CMOS digital process," in Proc. IEEE CICC, 2007, pp. 221–224.

[18] M. Tuthill, "A switched-current, switched-capapcitor temperature sensor in 0.6- m CMOS," IEEE J. Solid-State Circuits, vol. 33, no. 7, pp. 1117–1122, Jul. 1998.