VLSI Thermal Sensing and Management using Low Power Self-calibrated Delay-line Based Temperature Sensors

by

Shuang Xie

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy
Department of Electrical and Computer Engineering
University of Toronto

© Copyright by Shuang Xie 2014
Abstract

The power density of microprocessor chips continues to rise due to the growing demand on microprocessor performance and technology scaling. The resulting temperature rise and thermal gradient not only increase cooling cost, leakage current/power consumption and failure rate, but also affect device speed. These thermal induced problems could be alleviated by employing dynamic thermal management (DTM) that reduces thermal emergencies while avoiding performance degradation. To enable an effective DTM, accurate monitoring of the on-chip temperatures at a large number of strategic locations becomes necessary. This thesis proposes a fully-digital self-calibration method that could remove the delay-line based temperature sensors’ sensitivities to process and supply voltage variations. The proposed calibration method assigns a unique correction factor, \( N_C \) to each sensor, making all the sensors have the same calibrated outputs at start-up. Only one calibration block is required to calibrate multiple delay-line based temperature sensors sequentially. For each additional sensor, only additional registers for storing \( N_C \) are required. The proposed temperature sensors are demonstrated on an Altera Cyclone IV FPGA based VLSI thermal management system. Four microprocessor cores are mapped onto a Cyclone IV FPGA chip to emulate the VLSI load. Each core has a temperature sensor close by. The runtime thermal profiles for the four microprocessor cores with eight different Dynamic
Thermal Management (DTM) methods are obtained. Experimental results from different DTM techniques are studied. In comparison to a conventional global DFS approach, a proposed hybrid DTM reduces the amount of time that the MPSoC spends at higher temperatures and larger thermal gradients, by 10% and 21%, respectively. In addition, the proposed hybrid DTM offers a 10% improvement in the average processing rate (instructions per second) when compared with the conventional global DFS approach.
Acknowledgments

I would like to thank my supervisor, Professor Wai Tung Ng, for his guidance during my PhD study. This thesis would be impossible without Professor Ng’s patience and kindness. In addition, I would like to express my gratitude to everyone in professor Ng’s Lab.

I would also like to thank the China Scholarship Council for the financial support. The work in this thesis was supported in part by the China Scholarship Council (File No. 2009102021), AMD Canada, and Natural Science and Engineering Research Council of Canada. We would like to acknowledge Canadian Microsystems Corporation for facilitating IC fabrication and design support.

Finally, I would like to thank my all families and friends for their unconditional emotional support.
# Table of Contents

Acknowledgments ...................................................................................................................... iv

Table of Contents ....................................................................................................................... v

List of Publications .................................................................................................................... viii

List of Glossary .......................................................................................................................... ix

List of Tables ............................................................................................................................... x

List of Figures ............................................................................................................................. xi

List of Appendices ..................................................................................................................... xxii

Chapter 1  Introduction ................................................................................................................. 1
  1.1 Motivation .............................................................................................................................. 1
  1.2 Thesis Contributions ............................................................................................................. 5

Chapter 2  Low Power, Small Area, Low Noise Delay-line Based Temperature Sensors ....... 9
  2.1 Literature Review of Digital Temperature Sensors ............................................................. 9
    2.1.1 Bandgap based temperature sensor ................................................................................. 9
    2.1.2 Delay-line based temperature sensor ............................................................................. 12
    2.1.3 Comparison of the state-of-the-art digital temperature sensors, design challenges and considerations ......................................................................................................................... 18
  2.2 Temperature Dependence of Delay-line Based Temperature Sensors ............................ 22
    2.2.1 Deductions of single CMOS inverter propagation delay .............................................. 22
    2.2.2 Temperature dependence of delay-line based temperature sensor ............................... 24
  2.3 Low Power and Small Area Delay-line Based Temperature Sensor .............................. 26
    2.3.1 Power and area saving techniques ................................................................................. 26
    2.3.2 Simulation results on proposed power and area saving techniques ............................ 31
  2.4 Reduction of Digital Outputs Noise in Time Domain ......................................................... 33
    2.4.1 Noise in ring oscillator ................................................................................................. 33
    2.4.2 Time-domain noise reduction for delay-line based temperature sensors ................. 33
2.5 Experimental Results ........................................................................................................ 35
  2.5.1 Temperature dependence of delay-line based temperature sensor ......................... 35
  2.5.2 Low power and small area delay-line based temperature sensor ......................... 37
  2.5.3 Reduction of the digital outputs noise in the time domain ................................... 38
  2.6 Summary ......................................................................................................................... 39

Chapter 3 Self-calibration Methods for Delay-line Based Temperature Sensors .......... 40
  3.1 Literature Review of the State-of-the-Art Calibration Methods ........................................ 40
    3.1.1 State-of-the-art process variation removal methods for delay-line based
          temperature sensors ........................................................................................................ 40
    3.1.2 State-of-the-art voltage variation removal methods for delay-line based
          temperature sensors ........................................................................................................ 50
  3.2 Proposed Self-calibration Step I: Process Variation Sensitivity Removal .......... 58
    3.2.1 Theoretical analysis for proposed process variation removal method for delay-
          line based temperature sensors ..................................................................................... 58
    3.2.2 Circuit architectures and simulation results in ModelSim .................................... 64
  3.3 Proposed Self-calibration Step II: Voltage Variation Sensitivity Removal ........ 67
  3.4 Proposed Self-calibration Step III: Automatic Calibration against Accurate On-chip
       Reference .......................................................................................................................... 69
    3.4.1 Automatic self-calibration against accurate on-chip temperature reference ...... 69
    3.4.2 Accurate on-chip temperature reference ................................................................. 72
    3.4.3 Layout and simulation Results .................................................................................. 85
  3.5 Experimental Results ....................................................................................................... 102
    3.5.1 Removal of sensitivity to process variations .............................................................. 105
    3.5.2 Removal of sensitivity to supply voltage level variations ........................................ 114
    3.5.3 Automatic calibration against accurate temperature reference ......................... 116
  3.6 Summary .......................................................................................................................... 125

Chapter 4 Thermal Management Using the Proposed Delay-line Based Temperature Sensors
.................................................................................................................................................. 128
4.1 Literature Review on the State-of-the-Art Thermal Management Technologies ........ 128

4.2 Thermal Management Demonstrated on a 60nm FPGA using Proposed Delay-line Based Temperature Sensors ................................................................. 133
  4.2.1 Thermal modeling and prediction ......................................................... 133
  4.2.2 Experimental results of thermal management on Cyclone IV FPGA using proposed delay-line based temperature sensors .................................. 141

4.3 Summary .................................................................................................. 153

Chapter 5 Conclusions .................................................................................. 157

  5.1 Thesis Contributions ................................................................................ 157
  5.2 Future Trends .......................................................................................... 158
  5.3 Future Work ............................................................................................ 160

References ..................................................................................................... 162

Appendices .................................................................................................... 167

Copyright Acknowledgements ........................................................................ 173
List of Publications

1) S. Xie and W.T. Ng, “Delay-line based temperature sensors and VLSI thermal management demonstrated on a 60nm FPGA,” accepted for publication in IEEE International Symposium on Circuits and Systems (ISCAS), 4 pages, to be held in Melbourne, Australia, June 1-5, 2014.


7) S. Xie and W.T. Ng, “Implementation of VLSI thermal management systems with on-chip all-digital self-calibrated temperature sensors”, submitted to VLSI, the Integration Journal.


List of Glossary

TDP: thermal design power

MPSoC: multiprocessor system on a chip

DTM: dynamic thermal management

DPM: dynamic power management

DVFS: dynamic voltage frequency scaling

ARMA: auto-regression moving average

OLDL: open loop delay line

DLL: delay locked loop

MSB: most significant bit

LSB: least significant bit

SAR: successive approximate

ZTC: zero temperature coefficient

PCA: principle component analysis

PMU: power management unit
List of Tables

Table 2.1  Comparison of the State-of-the-art Digital Temperature Sensors ......................... 20
Table 2.2  A Comparison of the Bandgap and the Delay-line Based Temperature Sensors.. 21
Table 2.3  Parameters Setting for the Schematic Shown in Figure 2.17 and Simulation Results Plotted in Figure 2.18 .............................................................. 25
Table 2.4  Descriptions and Measurement Results of Four Different Temperature Sensor Architectures (M = 12) ................................................................. 36
Table 3.1  Transistor Size of the Bias and Startup Circuit Shown in Figure 3.29 ............... 86
Table 3.2  Component Sizes for the Circuit Shown in Figure 3.28. ................................. 87
Table 3.3  Component Sizes for the Circuit Shown in Figure 3.27. ................................. 89
Table 3.4  Transistor Sizes for the Circuit Shown in Figure 3.30 ................................. 90
Table 3.5  Parameters for Calculating Mismatches for a Capacitor Sizing 10 μm × 10μm .... 94
Table 3.6  Test Parameters for SAR ADC’s Static and Dynamic Characteristics ............. 98
Table 3.7  Simulated Specifications of the SAR ADC .................................................... 101
Table 3.8  A Comparison of the State-of-the-art Digital Temperature Sensors and Calibration Methods ............................................................... 126
Table 3.9  A Comparison with Previous Publications on Calibration methods ................. 127
Table 4.1  Power Consumption of the FPGA Without DTM With Four Cores Running .... 140
Table 4.2  Performance and Temperature Comparisons for Different DTM Techniques.... 156
List of Figures

Figure 1.1  The increases in clock speed and the power density for CPUs over the past two decades [6]........................................................................................................................................... 2

Figure 1.2  Maximum junction temperature for a quad-core processor as a function of power consumption [4]........................................................................................................................................... 3

Figure 1.3  Multiple on-chip temperature sensors (red squares) for thermal management as a supplement to an existing power management system on a VLSI chip................. 4

Figure 2.1  Digital temperature sensors used in the microprocessor in [9]. ....................... 9

Figure 2.2  Block diagram of the state-of-the-art bandgap based temperature sensors in [29], [33], [34], [35]........................................................................................................................................... 11

Figure 2.3  Operating principle of a substrate PNP based PTAT [34]. ......................... 12

Figure 2.4  A PTAT voltage generator in classical textbook [32]........................................ 12

Figure 2.5  A ring-oscillator based temperature sensor on a Xilinx FPGA proposed in [37]. 13

Figure 2.6  A delay-line based temperature sensor in [38].................................................. 14

Figure 2.7  Timing diagram of the delay-line based temperature sensor in Figure 2.6........ 14

Figure 2.8  Single cell in the temperature-independent delay line in [38]............................... 15

Figure 2.9  Circuit schematic of the TDC in Figure 2.6 in [38]................................................ 15

Figure 2.10 Timing diagram of the TDC in Figure 2.9. .......................................................... 15

Figure 2.11 The time-domain delay-line based temperature sensor in [24]......................... 16

Figure 2.12 Timing diagram of the delay-line based temperature sensor in Figure 2.11.... 16

Figure 2.13 The cyclic delay-line based temperature sensor as an improved version of Figure 2.11........................................................................................................................................... 17
Figure 2.14 The dual delay-line based temperature sensor proposed in [21] .......................... 18
Figure 2.15 Future sensing architecture for thermal management, as predicted in [41]. ........ 22
Figure 2.16 The logic inverter used in the deductions in Section 2.2.1. ................................ 22
Figure 2.17 Proposed delay-line based temperature sensor in Section 2.2.2 ....................... 25
Figure 2.18 Simulated output codes using schematic shown in Figure 2.17 and setup listed in Table 2.3 ................................................................................................................. 26
Figure 2.19 The current-starved delay cell as proposed in [44] and [46]. ......................... 28
Figure 2.20 A comparison of the conventional ring-oscillator based temperature sensor (a) and the proposed power saving architecture with tab decoding (b). ......................... 30
Figure 2.21 Timing diagram of the tab decoding technique proposed in Figure 2.20. ............ 30
Figure 2.22 Timing diagram of the latch and tab decoding shown in Figure 2.20 ............... 31
Figure 2.23 Simulation results of power and energy consumptions using the conventional and the proposed power saving methods. The area includes the total area of the ring oscillator, the counter and the tab decoder (M = 12). ......................................................... 32
Figure 2.24 Simulated power is reduced as the number of bits N in the decoder increases ..... 32
Figure 2.25 Noise reduction of digital output noise for delay-line based temperature sensors in the time domain. (a) Architecture. (b) Timing diagram. ................................. 34
Figure 2.26 Un-calibrated digital codes of the 12 sensors, with 3 for each of the four architectures as listed in Table 2.4, on the Cyclone III FPGA chip. ...................... 36
Figure 2.27 Measurement errors of the digital outputs shown in Figure 2.26, using a linear master curve, after a self-calibration method that removes process variations is performed. .................................................................................................................. 37
Figure 2.28  Experimental results on power and energy consumption using the traditional and proposed power/area saving techniques on Cyclone III FPGA. The area includes the total area of the ring oscillator, the counter and the tab decoder (M = 12). .... 38

Figure 2.29  Experimental Results show that errors caused by timing jitter are reduced by the method proposed in Section 2.4.2. ................................................................. 39

Figure 3.1  A linear curve fitting for the delay-line based temperature sensor measurements. ........................................................................................................................................... 41

Figure 3.2  Two-point calibration method for different temperature sensors.................... 42

Figure 3.3  Illustration of digital outputs before and after the one-point calibration. .......... 43

Figure 3.4  The dual delay-line based temperature sensor proposed in [21]....................... 46

Figure 3.5  Calibration mode (upper) and measurement mode (lower) in [21].................... 46

Figure 3.6  One-point calibration for cyclic delay-line based temperature sensor in [22]...... 48

Figure 3.7  Schematic of the programmable time amplifier in [22]. .................................... 48

Figure 3.8  Timing diagram of the circuit architecture in Figure 3.7..................................... 49

Figure 3.9  Threshold voltage sensing circuit in the process sensor proposed in [53]........... 50

Figure 3.10  Block diagram of the temperature sensor in [55]............................................ 52

Figure 3.11  Capacitor delay determined by the discharging current in [55]........................ 52

Figure 3.12  Bias circuits that generate the positive and the negative delay in [55]........... 53

Figure 3.13  Timing diagram of the design in Figure 3.11 .................................................. 53

Figure 3.14  Subthreshold leakage current based temperature sensor proposed in [56]. ...... 55

Figure 3.15  Process sensor proposed in [56], when VBIASP = 0.6 V, VBIASN = 0.4 V, and VDD = 0.5V to make its delay temperature independent. .............................. 56
Figure 3.16  Schematic of the voltage sensor in [56]. ................................................................. 57

Figure 3.17  MATLAB verification of the self-calibration method proposed in (3.31) using randomly outputs generated following the pattern described in (3.28) .................. 60

Figure 3.18  MATLAB verification of the self-calibration method proposed in (3.31) using measured un-calibrated outputs from Cyclone III FPGA chip shown in Figure 2.26. .............................................................................................................. 61

Figure 3.19  The effect of process variations on the threshold voltage, surface carrier mobility, $H(T_i)$ and the errors resulted from treating $H(T_i)$ as constant despite process variations. The simulations are in three corners: TT, FF, SS. ........................................... 63

Figure 3.20  System level diagram of the proposed self-calibration block that calibrates multiple temperature sensors .................................................................................. 65

Figure 3.21  The SA algorithm and multiplying circuitry. ......................................................... 66

Figure 3.22  The removal of process variations in ModelSim simulation. The un-calibrated outputs are from the measurements of four temperature sensors on a Cyclone III FPGA. .............................................................................................................. 67

Figure 3.23  Output codes versus time on a FPGA based VLSI thermal management system, when DFS takes place, the output code change may be mistaken for a temperature change. .............................................................................................................. 68

Figure 3.24  Power supply level of the FPGA chip, measured at the same time as the experimental results shown in Figure 3.23 .......................................................................... 68

Figure 3.25  ModelSim simulation results for the calibration to true temperature after the removal of process variations. The un-calibrated outputs are from the measurements of four temperature sensors on a Cyclone III FPGA shown in Figure 3.22. .......................................................... 71

Figure 3.26  Block diagram of the on-chip bandgap based temperature sensor reference. ...... 72

Figure 3.27  Schematic of the PTAT in the temperature reference. ................................. 73
Figure 3.28  Schematic of the op amp used in Figure 3.27. ................................................................. 75

Figure 3.29  Startup and biasing circuit for the accurate reference on-chip. ................................. 76

Figure 3.30  The bandgap voltage reference for the SAR ADC. ......................................................... 77

Figure 3.31  The timing circuit for the SAR ADC. .............................................................................. 78

Figure 3.32  Schematic of Driver1 used in Figure 3.31. ................................................................. 78

Figure 3.33  The circuit diagram of the SAR ADC in sampling phase [59]. ................................. 79

Figure 3.34  The circuit diagram of the SAR ADC in bit conversion phase (I) [59]. ............... 80

Figure 3.35  The circuit diagram of the SAR ADC in bit conversion phase (II) $(V_{ip} > V_{in}$ in the first step) [59]. ................................................................. 80

Figure 3.36  SAR ADC in bit conversion phase (II) $(V_{ip} > V_{in}$ in the first step). (a) $V_{ip} > V_{in} + (1/2)(V_{ref} - V_{cm})$. (b) $V_{ip} < V_{in} + (1/2)(V_{ref} - V_{cm})$ [59]. ......................................................... 82

Figure 3.37  The SAR logic circuit [60]. ......................................................................................... 83

Figure 3.38  The rail-to-rail comparator used in the SAR ADC in Figure 3.33 [61]. ............... 84

Figure 3.39  Layout of the accurate on-chip bandgap based temperature sensor using TSMC’s 65nm CMOS technology in Cadence Virtuoso (Capacitor array is not shown). . . . 86

Figure 3.40  Schematic simulation of startup circuit shown in Figure 3.29. ......................... 87

Figure 3.41  Postlayout simulation results of the op amp shown in Figure 3.28, with gain 52 dB, UBW 300MHz, phase margin 45 °. ......................................................... 88

Figure 3.42  Post-layout simulation results of the PTAT voltage architecture in Figure 3.27. . 89

Figure 3.43  Postlayout simulations of VCM and VTOP in Figure 3.30. ............................................. 90

Figure 3.44  1/PSRR of the Bandgap reference in Figure 3.30. ..................................................... 91

Figure 3.45  Schematic simulation results of the SAR timing circuit in Figure 3.31. ............. 92
Figure 3.46  Schematic simulation results of the SAR logic circuit in Figure 3.37. .............. 93

Figure 3.47  Post-layout simulation of the comparator shown in Figure 3.38. ...................... 93

Figure 3.48  Monte Carlo mismatch simulation results on different sizes of the minimum capacitor sizes used in the SAR AD when minimum cap = 19 pF, Mu/sigma = 350. ................................................................................................................. 95

Figure 3.49  Monte Carlo mismatch simulation results on different sizes of the minimum capacitor sizes used in the SAR ADC when minimum capacitor = 76 pF, Mu/sigma = 960. ....................................................................................................................... 95

Figure 3.50  Monte Carlo mismatch simulation results on different sizes of the minimum capacitor sizes used in the SAR ADC when minimum capacitor = 200 pF, Mu/sigma = 1738.3. .......................................................................................................................... 96

Figure 3.51  Schematic simulations of SAR ADC. (a) using VCM, Vref voltage levels from Figure 3.43 and differential input voltage levels as those in PTAT voltage in Figure 3.42. (b) zoom in of (a) at around 350 μs when INN = 779.79 mV and INP = 649.96 mV ............................................................................................................................................. 97

Figure 3.52  Simulated static linearity of the SAR ADC. ......................................................... 99

Figure 3.53  Simulated dynamic linearity of the SAR ADC. .................................................. 100

Figure 3.54  Simulation results of the accurate temperature reference using schematic shown in Figure 3.26. The schematic simulation includes I/Os. The post-layout simulation is performed with extracted RC using Calibre tools in the Cadence Environment. ................................................................................................................................. 102

Figure 3.55  Micrograph of the prototype custom IC implemented using TSMC’s 1V 65nm CMOS technology .............................................................................................................................................. 103

Figure 3.56  Micrograph of the custom IC implemented using IBM’s 0.13 μm CMOS technology. .............................................................................................................................................. 104
Figure 3.57  Floor-plan of the self-calibration algorithm block implemented on a Cyclone III FPGA. The accurate temperature reference is off-chip. The four temperature sensors are of different architectures (sizes), as listed in Table 2.4.  ........................................ 104

Figure 3.58  Floor-plan of the self-calibration algorithm block implemented on a Cyclone IV FPGA chip. .......................................................................................................................... 105

Figure 3.59  Un-calibrated output codes for 4 sensors on each of the three Cyclone III FPGA chips, using the self-calibration method proposed in Section 3.2. Chip I and II have batch number 1108 and Chip III has batch number 0810. ........................................... 106

Figure 3.60  Die-to-die and on-chip process variations are removed for 4 sensors on each of the three Cyclone III FPGA chips, using the self-calibration method proposed in Section 3.2. Chip I and II have batch number 1108 and Chip III has batch number 0810. .......................................................................................................................... 107

Figure 3.61  Un-calibrated outputs for 4 sensors on a Cyclone IV FPGA chip, using the self-calibration method proposed in Section 3.2........................................................................ 108

Figure 3.62  On-chip process variations are removed on a Cyclone IV FPGA chip, using the self-calibration method proposed in Section 3.2................................................................. 109

Figure 3.63  Un-calibrated outputs on 3 custom ICs fabricated using TSMC’s 65 nm CMOS technology, using the self-calibration method proposed in Section 3.2.......... 110

Figure 3.64  Process variations are removed on 3 custom ICs fabricated using TSMC’s 65 nm CMOS technology, using the self-calibration method proposed in Section 3.2. 111

Figure 3.65  Process variations are removed on 3 custom ICs fabricated using IBM’s 0.13 μm CMOS technology, using the self-calibration method proposed in Section 3.2. 112

Figure 3.66  Removal of sensitivity to supply voltage level variations for 12 sensors on 3 custom IC chips fabricated using TSMC’s 65nm CMOS technology. ............... 115
Figure 3.67 Measurement results of the un-calibrated outputs of the accurate temperature sensor shown in Figure 3.26 in Section 3.4.2, on three custom IC chips fabricated using TSMC’s 65nm 1V technology ................................................................. 116

Figure 3.68 Measurement results of the accurate temperature sensor shown in Figure 3.26 in Section 3.4.2, on three custom IC chips fabricated using TSMC’s 65nm 1V technology ................................................................. 117

Figure 3.69 Measurement results of 12 delay-line based temperature sensors on 3 custom ICs, after self-calibration step I and III are performed (cool and warm startups). The measurement results are obtained without supply voltage variations .................. 119

Figure 3.70 Measurement results of 12 delay-line based temperature sensors on 3 custom ICs fabricated using 65nm technology, after self-calibration step I, II and III are performed (all warm startups). The measurement results are obtained when the supply voltage variations are within ±5 % ................................................................. 120

Figure 3.71 Measurement results of 12 delay-line based temperature sensors on 3 custom ICs fabricated using IBM’s 0.13 μm technology, after self-calibration step I and III are performed (cool startup) ........................................................................ 122

Figure 3.72 Measurement results of 12 delay-line temperature sensors on 3 Cyclone III chips, after self-calibration step I and III are performed ......................................................... 123

Figure 3.73 Measurement results of 4 delay-line based temperature sensors on a Cyclone IV chip, after self-calibration step I and III are performed ................................................. 124

Figure 4.1 Illustrations of the DTM control strategy employed by the Core i7 processor compared to a predictive approach [63] ................................................................................. 130

Figure 4.2 Flow chart of the proactive thermal management in [3] ................................................................. 131

Figure 4.3 Flow chart of predictive thermal management in [4] ....................................................................... 131

Figure 4.4 The lumped RC thermal model analogous to an electrical RC network [2] .... 134

Figure 4.5 Block diagram of the 8085 processor used in each core ................................................................. 135
Figure 4.6  Floor plan of the proposed thermal management system on the Cyclone IV FPGA (The calibration block is loosely placed to avoid heating effect and routing problems) ........................................................................................................................................ 136

Figure 4.7  Measurement and modeled temperatures on the FPGA emulating MPSoC shown in Figure 4.6. ........................................................................................................................................ 137

Figure 4.8  Measured vs. predicted temperatures for four cores. Only Core #1’s run control is enabled. ........................................................................................................................................ 139

Figure 4.9  Errors resulted from using the predicted temperatures for thermal estimation. ........................................................................................................................................ 140

Figure 4.10 Errors resulted from using the predicted temperatures for thermal sensing. ........................................................................................................................................ 141

Figure 4.11 Output codes versus time in Core #1 on the Cyclone IV FPGA, with and without the proposed self-calibration proposed in Section 3.3 to remove sensitivity to supply voltage variations. ........................................................................................................................................ 142

Figure 4.12 Power supply level of the FPGA chip, measured at the same time as the experimental results shown in Figure 4.11 ........................................................................................................................................ 142

Figure 4.13 Thermal profiles of all four cores on the Cyclone IV FPGA without DTM. ........................................................................................................................................ 144

Figure 4.14 Thermal profiles of all four cores on the Cyclone IV FPGA with reactive global DFS DTM. ........................................................................................................................................ 144

Figure 4.15 Thermal profiles of all four cores on the Cyclone IV FPGA chip with reactive local DFS DTM. ........................................................................................................................................ 145

Figure 4.16 Thermal profiles of all four cores on the Cyclone IV FPGA chip with reactive thread migration DTM. ........................................................................................................................................ 145

Figure 4.17 Thermal profiles of all four cores on the Cyclone IV FPGA chip with a combined hybrid reactive global DFS and thread migration DTM. ........................................................................................................................................ 146

Figure 4.18 Thermal profiles of all four cores on the Cyclone IV FPGA chip with predictive global DFS DTM. ........................................................................................................................................ 146
Figure 4.19  Thermal profiles of all four cores on the Cyclone IV FPGA chip with predictive local DFS DTM. ................................................................. 147

Figure 4.20  Thermal profiles of all four cores on the Cyclone IV FPGA chip with predictive thread migration DTM. ................................................................. 147

Figure 4.21  Thermal profiles of all four cores on the Cyclone IV FPGA with a combined hybrid predictive global DFS and thread migration DTM. ......................... 148

Figure 4.22  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when without DTM ......................................................... 148

Figure 4.23  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with reactive global DFS DTM ...................... 149

Figure 4.24  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with reactive local DFS DTM ...................... 149

Figure 4.25  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with reactive thread migration DTM .......... 150

Figure 4.26  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with a combined predictive hybrid global DFS and thread migration DTM. ......................................................................................... 150

Figure 4.27  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with predictive global DFS DTM. .......... 151

Figure 4.28  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with predictive local DFS DTM .............. 151

Figure 4.29  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with predictive thread migration DTM ....... 152

Figure 4.30  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with a combined predictive hybrid global DFS and thread migration DTM. ......................................................................................... 152
Figure 4.31  Comparisons for different DTM techniques on higher temperatures.................. 154

Figure 4.32  Comparisons for different DTM techniques on larger spatial thermal gradients.155

Figure 4.33  Comparisons for different DTM techniques on processing rate. ......................... 155
List of Appendices

Appendix A: DTM vs. DPM.................................................................171

Appendix B: Screenshots of verifications in ModelSim of self-calibration methods proposed in Section 3.2 and 3.4..............................................................172

Appendix C: Layout of the accurate bandgap temperature sensor reference proposed in Section 3.4.2.................................................................175
Chapter 1   Introduction

1.1   Motivation

The power density of microprocessor chips continues to rise to accommodate for the growing demand of performance. This increased density is caused by the shrinking device dimension as a result of technology scaling and higher power consumption [1]–[4]. For example, Intel’s i7-980X microprocessor has a thermal design power (TDP) of 130W [5] in each of the four cores, compared with 115 W in the single core Pentium 4. Figure 1.1 (a) and (b) show the increases in clock speed and the power density for CPUs over the past two decades [6]. The relationship between the elevated temperature and increased power consumption is shown in Figure 1.2. This elevated temperature could cause problems for the microprocessor chip in the following aspects: (1) Degraded performance. Device speed has an inverse relationship with temperature. (2) Increased leakage power. Leakage current increases exponentially with temperature. The increased leakage power in turn gives rise to higher temperature. (3) Degraded reliability. A small difference in the operating temperature (i.e., 10–15 °C) can result in up to a 50 % reduction in the lifetime of the devices [3]. In addition, temporal and spatial thermal gradients degrade device reliability even at moderate temperature. These gradients could be created by, for example, fast thermal cycles of powering on and off [3]. These thermal induced problems limit the performance and reliability of modern VLSI chips. This limitation could be alleviated by employing dynamic thermal management (DTM) that reduces thermal emergencies while avoiding degradation in the computational throughput [4].

To alleviate the above mentioned thermal induced problems, cooling solutions are needed. Cooling solutions include package-level, design-time and runtime solutions. An example of a package-level solution is the heat sink that changes the thermal conductance of the thermal network [1]. Design-time thermal management ensures that reliability, performance and leakage constraints are met at design time, either through floor planning or package level techniques. Compared with design-time thermal management that aims for worst case situation, runtime thermal management is more flexible and cost effective [3]. Well-known runtime dynamic thermal management (DTM) techniques fall into two categories: toggling techniques such as
dynamic voltage frequency scaling (DVFS) and clock gating that reduce power consumption; thread migration scheduling that redistributes instead of directly affecting power consumption.

Figure 1.1  The increases in clock speed and the power density for CPUs over the past two decades [6].
To implement an effective DTM system, accurate thermal information is needed. There are basically three ways to capture temperatures on a DTM system [3], [7]: (1) temperature sensors (2) thermal modeling (3) performance counter (that relates temperature to hardware usage, e.g. instructions per cycle [8]) based temperature estimation. Modern microprocessors, such as the Pentium M, Core 2 duo, and Core i7, use thermal sensors to trigger an alert when their temperatures exceed a predefined threshold [7]. In Intel’s Core i7 microprocessor, multiple digital temperature sensors calibrated at the factory are located across the die. They provide thermal information and the speed of the cooling fans is adjusted accordingly [8]. The elevated temperature and thermal gradients also increase the number of hotspots. On modern VLSI chips, the thermal gradient could be larger than 10 °C [4]. This further increases the number of temperature sensors on-chip. An example of the on-chip temperature sensor placement is shown in Figure 1.3. The power and area overhead caused by the temperature sensors should be minimized. One research trend focuses on identifying hotspots by simulation and placing the temperature sensors in the best locations [10], [11]. Another is to up-sample the temperature readings from a limited number of temperature sensors and then re-construct the temperature profile [12], [13]. For example, reference [12] reconstructs the thermal status (detailed thermal

**Figure 1.2** Maximum junction temperature for a quad-core processor as a function of power consumption [4].
mapping) of an integrated circuit during runtime using a minimal number of thermal sensors, based on the Nyquist–Shannon sampling theory. This method applies to both uniform and non-uniform thermal sensor placements, generating a thermal profile with an absolute error of 0.6 % [12]. The most common thermal model is the lumped RC thermal model that is analogous to an electrical RC model. The variations of this model have been used in thermal simulation software such as HotSpot [14] to simulate an MPSoC (microprocessor system on a chip)’s transient and steady state behaviour. Modern MPSoCs are equipped with performance counters that provide information such as access rate, timing, and instructions per cycle. In [15], performance counter information reflecting the thermal contribution of the processing activity is collected to facilitate an estimation of the temperature.

Figure 1.3  Multiple on-chip temperature sensors (red squares) for thermal management as a supplement to an existing power management system on a VLSI chip.

The thermal information collected could be used either in a DTM control loop reactively or to estimate future temperature trend predictively. In reactive thermal management, when a certain thermal level is triggered on a microprocessor, the computer’s performance is throttled back to bring the temperature under control [4]. However, thermal overshoot could occur due to the thermal time constant (delay) before the control technique takes effect [16]. If a thermal management system could anticipate and choose the maximum frequency that does not lead to
future thermal violations, performance penalties could be avoided [4]. In [4], thermal modeling and prediction are based on the workload phase transition information. The thermal prediction analysis is simplified by extracting principal components from the computer performance counter measurements. In [3], based on a moving history window of temperature sensor measurements, temperature increments over a predefined time window ahead are predicted into the future using an auto-regressive moving average (ARMA) model.

An FPGA provides a fast emulation framework for thermal management system in MPSoC [14][17][18]. At the same time, the power dissipation in modern FPGAs increases with their logic densities to the point that managing power dissipation becomes a primary concern for designs under 90nm [19]. In [17], a hardware based MPSoC is mapped onto a FPGA, where runtime information is extracted. The information is then processed by a software unit that evaluates the thermal management strategies on the FPGA emulating MPSoC.

Although both DTM and dynamic power management (DPM) systems employ power saving techniques such as clock gating or DVFS, their differences can be briefly described as follows. On a VLSI chip, some local hotspot temperatures heat up much faster than the chip does as a whole [2]. This local hotspot problem is addressed by DTM to ensure that a certain temperature limit must not be exceeded. In contrast, DPM optimizes the overall energy consumption and is not effective in reducing temperature in the above local hotspots [2]. A detailed comparison of the DTM and DPM systems can be found in Appendix A.

1.2 Thesis Contributions

Currently, most MPSoCs use pn junction diodes as temperature sensors. The diode voltage, which varies with temperature, is measured by an analog to digital converter (ADC). In older microprocessors the ADC is off-chip while in newer ones (e.g. Intel’s i7 microprocessor) the ADC is on-chip [4]. The diode temperature sensor requires a pn junction or a parasitic bipolar transistor in the CMOS process. The ADC is usually a delta-sigma ADC. The MPSoC temperatures in [14] and [18] are monitored using CMOS delay-line based temperature sensors [20]. The CMOS delay-line based temperature sensor relies on the negative temperature coefficient of the logic inverters. It shares the same power grid with digital blocks and can be fully synthesizable. For thermal monitoring on a digital platform such as FPGA, to synthesize
temperature sensors for thermal monitoring, fully digital temperature sensors have to be used, where the delay-line based temperature sensors are the preferred choice.

However, it is reported that the readings of the conventional delay-line based temperature sensors used in [14] and [18] are affected by process and power supply voltage variations, on both FPGA and custom IC implementations [21], [22], [23]. This is due to the fact that the single inverter propagation delay is not only a function of temperature, but also that of process variations and the supply voltage level. Process variations include variations in carrier mobility $\mu$, threshold voltage $V_{TH}$, load capacitors $C$, or $W/L$ ratio, due to doping or etching non-uniformities during the fabrication process. Due to these process variations, two delay-line based temperature sensors either on the same or different chips will have different outputs, at the same temperature and the same power supply level. According to the conventional two-point calibration method, each sensor has to be calibrated individually, as each has a unique digital output versus temperature characteristic (each has a unique gain and a unique offset). As mentioned in Section 1.1, a large number of sensors are needed across the chip. Traditional two-point calibration requires a significant amount of time and effort, making it unsuitable for mass production.

Reference [21] proposes a one-point calibration method that removes the effect of process variations of delay-line based temperature sensors. The one-point calibration method in [21] has a calibration mode where the delay of the temperature-dependent open-loop delay line (OLDL) is adjusted to be the same for all the sensors at the reference/calibration temperature. This is achieved by comparing the delay of theOLDL to that of a delay locked loop (DLL) that is always independent of temperature, in the calibration mode. During the measurement mode, the pulse width of theOLDL is measured by the DLL. Making use of the same calibration principle as in [21], reference [22] proposes a one-point calibration for a cyclic delay-line based temperature sensor design. The sensors’ digital outputs are made to be the same at the calibration temperature (e.g. 50 °C) in [22]. This is done by adjusting the number of cycles of each sensor’s cyclic delay line using an off-chip time-domain phase detection circuit. The details of the calibration circuitries of [21] and [22] will be reviewed in Chapter 3.

One of the limitations of the one-point calibration in [21] and [22] is that each temperature sensor has to be calibrated individually. In other words, mass calibration is still not possible in
Another problem is that after the removal of process variations, the digital outputs of the temperature sensor are still sensitive to power supply variations, as reported in [21]. In this case, output variations caused by the power supply level changes can be mistaken for a temperature change.

The power supply variations occur when workload changes during runtime. Experiments on a Xilinx Virtex-5 FPGA verified that an instantaneous change from 0 to 80 % utilization running at 100 MHz leads to a 73 °C error in the estimated temperature [23]. Reference [21] reports that delay-line based temperature sensors on a 0.13 µm CMOS custom IC have a $\frac{\Delta T}{\Delta V_{dd}}$ (temperature sensing error over supply voltage) sensitivity ratio of 1.6 °C/mV, which translates to a 80 °C error when the supply voltage changes by 50 mV.

To overcome the above problems of sensitivities to process and supply voltage variations, this thesis presents a fully digital self-calibration method that could calibrate multiple delay-line based temperature sensors using only one calibration block. The contributions of this thesis resulted in numerous publications. They are summarized as follows.

- Chapter 2 introduces a power saving technique for delay-line based temperature sensor. The power saving technique alleviates the trade-off between power and area in the traditional designs. A method that reduces time-domain noise in the digital outputs of the delay-line based temperature sensors is also proposed. Experimental verifications are provided. The power saving technique was published in [24].

- Chapter 3 presents a fully digital self-calibration method that removes the temperature sensors’ sensitivities to process and supply voltage variations. The proposed method compensates the effects of power supply voltage and process variations by assigning a unique correction factor, $N_C$ to each sensor, making all the sensors’ calibrated outputs the same at start-up. The correction factor is updated whenever significant supply voltage variations are detected. Only one calibration block is required to calibrate multiple delay-line based temperature sensors sequentially. For each additional sensor, only additional registers for storing $N_C$ are required. The proposed self-calibration method is supported with measurement results. The self-calibration method also correlates the process and voltage variations removed digital outputs to the true temperatures with reference to an
accurate on-chip bandgap based temperature sensor reference. The proposed self-calibration method was published in [25], [26].

- Chapter 4 demonstrates the self-calibrated temperature sensors proposed in Chapter 3 on an Altera Cyclone IV FPGA based VLSI thermal management system. Four microprocessor cores are mapped onto a Cyclone IV FPGA chip to emulate the VLSI load. The runtime thermal profiles for the four microprocessor cores using eight different dynamic thermal management (DTM) methods are obtained. Among the eight different DTM methods, a proposed DTM that combines the global Dynamic Frequency Scaling (DFS) method and thread migration has been shown to be effective. Experimental results from eight different DTM techniques are studied. In comparison to a conventional global DFS approach, the proposed hybrid DTM reduces the amount of time that the MPSoC spends at higher temperatures and larger thermal gradients, by 10 % and 21 %, respectively. In addition, the proposed hybrid DTM offers a 10 % improvement in the average processing rate (instructions per second) when compared with the conventional global DFS approach. Part of this demonstration of the delay-line based temperature sensors on the FPGA based thermal management system was published in [26].
Chapter 2  Low Power, Small Area, Low Noise

Delay-line Based Temperature Sensors

The spatial thermal gradients across the VLSI chip could be as large as 10 °C. On the chip the number of hotspots increases due to higher power consumption and smaller device size. As a result there is a need for a large number of temperature sensors to monitor the thermal profile. This leads to the requirements that the area and power of the temperature sensor must be minimized.

This chapter will focus on delay-line based temperature sensors [21] that are suitable for massive on chip deployment. Section 2.1 gives a literature review of bandgap based and delay-line based temperature sensors. Section 2.2 explores temperature dependence of delay-line based temperature sensors. The power saving technique for delay-line based temperature sensors is proposed in Section 2.3. Reduction of time-domain digital output noise is introduced in Section 2.4, followed by experimental results in Section 2.5.

2.1 Literature Review of Digital Temperature Sensors

2.1.1 Bandgap based temperature sensor

The most common type of digital temperature sensors used in state-of-the-art microprocessors (e.g. Intel’s i7 microprocessor [9]) is based on a pn junction diode. The voltage could be quantized by an ADC, as shown in Figure 2.1. The sensors are calibrated (trimmed) at the factory to remove the voltage offsets in $V_{BE}$ [9].

![Figure 2.1 Digital temperature sensors used in the microprocessor in [9].](image)
In 1986, Meijer et al. proposes an accurate bandgap temperature sensor that generates a voltage proportional to the absolute temperature (PTAT) in [28]. Meijer commented that the pn junction based temperature sensor shown in Figure 2.1 suffers from offset problems, as \( V_{BE} \) varies for different diodes, due to process variations. At ISSCC 2012, a bandgap based temperature sensor with a three sigma accuracy of ±0.15 °C from −55 to 125 °C was presented [29]. It is calibrated by a fast voltage comparison of \( \Delta V_{BE} \) and \( V_{BE} \) with an external voltage. Its analog temperature sensing front-end and the ADC circuitries only consume a power of 3.4 μW and occupy a chip area of 0.08 mm\(^2\). References [30] and [31] implemented bandgap based temperature sensors in 65nm and 32nm CMOS technologies, respectively. Their specifications are reported in Table 2.1.

The block diagram of a state-of-the-art bandgap based temperature sensor is as shown in Figure 2.2. In standard CMOS technology, the PNP s are available as vertical substrate PNP s, the collectors of which are always connected to ground. Figure 2.3 shows the basic operating principle of a substrate PNP based PTAT. The collector current of a PNP transistor is:

\[
I_C = I_S e^{(V_{BE}/V_T)}
\]

where \( I_S \) is the saturation current, and it is related to a Gummel number \( G_B \) that reflects the number of impurities per unit area of the base. Thus, the \( V_{BE} \) and \( \Delta V_{BE} \) are respectively:

\[
V_{BE} = V_T \cdot \ln\left(\frac{I_C}{I_S}\right)
\]

\[
V_T = \frac{kT}{q}
\]

\[
\Delta V_{BE} = V_{BE}(I) - V_{BE}(pI) = \frac{kT}{q} \ln(p)
\]

At room temperature \( V_T = kT/q = 26 \text{ mV} \). Therefore \( \Delta V_{BE} \) is proportional to absolute temperature \( T \) as indicated in (2.4). In a classical textbook [32], the two current sources drawn in Figure 2.3 are implemented using NMOS transistors, as shown in Figure 2.4. However, in the recent publications [29] and [33], the NMOS transistors are replaced with another PNP pair of biasing current sources. In this way, not only the channel length modulation effect in the NMOS transistors is avoided but also the second-order non-linearity in \( \Delta V_{BE} \) is compensated. Precision
circuit techniques such as dynamic element match (DEM) are also used on the PNP pair as shown in Figure 2.3 to further reduce the effect of mismatches [34].

Chopping and auto-zeroing [34] are examples of other precision circuit techniques employed to eliminate the input offset and mismatch of the Op-amps used in the ADC circuitries. The $\Delta V_{BE}$ and $V_{BE}$ are sampled by the ADC shown in Figure 2.2. The temperature independent reference is the sum of two components, $\Delta V_{BE}$ that has a positive and $V_{BE}$ that has a negative temperature coefficient:

$$V_{\text{REF}} = \alpha \cdot \Delta V_{BE} + V_{BE}$$ (2.5)

Exact values for $\alpha$ were obtained through simulations, taking into account the actual front-end biasing and PNP emitter areas [35]. In [33] and [35], second or first order delta-sigma ($\Delta\Sigma$) ADCs are used to convert the analog PTAT signal into digital representations of temperature, respectively. In [29], a SAR (successive approximation) ADC is used for fast (coarse) conversion of MSBs, followed by a second-order delta-sigma ADC for fine conversion. The bitstream measurement from the ADC is:

$$\mu = \frac{\alpha \cdot \Delta V_{BE}}{V_{\text{REF}}} = \frac{\alpha \cdot \Delta V_{BE}}{\alpha \cdot \Delta V_{BE} + V_{BE}}$$ (2.6)

The digital back-end converts the ADC’s outputs into true temperature readings:

$$D_{\text{OUT}} = A \cdot \mu + B$$ (2.7)

where $A \approx 600$ and $B \approx -273$ are the gain and offset coefficients resolved for a direct result in degree Celsius [30]. The digital back-end is often off-chip for flexibility and area saving [29], [33], [34], [35].

![Figure 2.2](image)

**Figure 2.2** Block diagram of the state-of-the-art bandgap based temperature sensors in [29], [33], [34], [35].
2.1.2 Delay-line based temperature sensor

Delay lines, or ring oscillators, are direct ways to measure delay, which is affected by temperature. It was not until the 1990s that publications on the delay-line based temperature sensors first appeared [36], [37]. In 1998, Sergio Lopez-Buedo et al. presented experimental results using ring oscillators for thermal sensing on a Xilinx FPGA [37]. The sensor’s circuit
schematic is as shown in Figure 2.5. The details of the operating principle of a ring-oscillator based temperature sensor will be covered in Section 2.2.2.

![Diagram of a ring-oscillator based temperature sensor](image)

**Figure 2.5** A ring-oscillator based temperature sensor on a Xilinx FPGA proposed in [37].

In 2005, P. Chen *et al.* proposed a delay-line based temperature sensor. The temperature sensor’s schematic and timing diagram are shown in Figure 2.6 and Figure 2.7, respectively [38]. The temperature sensor is comprised of two parts: the thermal sensing part that generates temperature-dependent time pulse $T_P$ and the time-to-digital converter (TDC) that converts the time pulse $T_P$ into digital signal. In the thermal sensing part, there are a temperature-dependent delay line, which is comprised of CMOS logic inverters or buffers, and a temperature-independent delay line. The propagation delay of a single inverter cell has a positive temperature coefficient, mainly due to the negative temperature coefficient of the surface carrier mobility for electrons [39]:

$$t_p = \frac{(L/W)C_L}{\mu C_{in}(V_{DD} - V_{TH})} \left[ \frac{2V_{TH}}{V_{DD} - V_{TH}} \ln \frac{1.5V_{DD} - 2V_{TH}}{0.5V_{DD}} \right]$$  \hfill (2.8)

$$\mu(T) = \mu_0 \left( \frac{T}{T_C} \right)^{-k_\mu}$$  \hfill (2.9)

$$V_{TH}(T) = V_{TH}(T_C) - \alpha_v(T - T_C)$$  \hfill (2.10)

where $k_\mu$ and $\alpha_v$ are treated as positive constants. According to (2.8), the higher the temperature, the slower the temperature-dependent line propagates, and the wider the pulse width $T_P$. The introduction of the temperature-independent delay line alleviates the number of bit requirement for the following TDC, as the delay between the propagation delay of $Reset_R$ and $Reset_D$ is
less than that between Reset and Reset$_D$, as illustrated in Figure 2.7. The single cell in the temperature-independent delay line is shown in Figure 2.8. Except for the inverter drawn in the dashed square, the other transistors in Figure 2.8 have to be properly sized such that the temperature coefficient of the inverter’s propagation delay induced by that of the other transistors’ carrier mobility and threshold voltages are cancelled out. The single cell inverter in the TDC has the same architecture as that shown in Figure 2.8. The circuit diagram of the TDC is shown in Figure 2.9. The TDC has an even number of inverter cells. All of the inverters in the TDC have the same dimension except for one inhomogeneous inverter whose width is several times as much as that of the rest. The inclusion of the inhomogeneous inverter causes the input pulse width to shrink for each number of cycles [38]. The operating principle of the TDC is as shown in Figure 2.10: Between $t_0$ and $t_1$, when Reset is low, $T_P$ is low, and the output $T_{OUT}$ is still at its stable state “0”. After $T_P$ rises to high at $t_2$, the NAND gate driven by the $T_P$ can be seen as a buffer, and the output $T_{OUT}$ is a delayed version of $T_P$. The counter counts the number of the incoming pulse $T_{OUT}$. For each cycle, the pulse width $T_P$ shrinks for a fixed amount (independent of temperature) in the TDC, until it diminishes completely, when the counter stops to count.

Figure 2.6  A delay-line based temperature sensor in [38].

Figure 2.7  Timing diagram of the delay-line based temperature sensor in Figure 2.6.
The authors of [38] pointed out that there are two limitations with their own design. First of all, the minimum propagation delay of the temperature-dependent delay line must be longer than that of the temperature-independent one. This happens at the lowest temperature in the temperature range. Second, the output pulse width of the temperature-to-pulse generator cannot exceed the circulation time of the cyclic TDC, to prevent the cyclic TDC from entering the erroneous stable state (when $T_{OUT}$ is all high) [38]. This requirement has to be met at the lowest temperature in the temperature range, when the circulation time of the cyclic TDC is at its lowest. However, there are two limitations with the above design shown from Figure 2.6 to Figure 2.10 in [38] not mentioned by its authors. First, both the temperature-independent delay line and the TDC need
careful sizing of their transistors to cancel out the temperature coefficients in carrier mobility and threshold voltage, as discussed earlier. However, there are geometry variations due to lithography resolution limitations. Using larger size transistor would eliminate the problem, but doping uncertainties still exist, which could make the TDC’s resolution or the temperature-independent delay line’s delay temperature dependent. Second, the design in Figure 2.8 is a digital-like, but not real digital design, which makes it impossible to be either synthesized or implemented using the standard digital logic cells.

The same authors of [38] proposed a FPGA based time-domain delay-line based temperature sensor in [24]. As shown in Figure 2.11, the design includes a temperature-dependent logic delay line which is implemented using logic buffers on the FPGA, and a counter that quantizes the number of reference clocks (temperature independent) within the temperature-dependent pulse $T_p$. The higher the temperature, the slower the delay line in Figure 2.11 propagates, and the wider the pulse $T_p$ is in Figure 2.12.

![Figure 2.11 The time-domain delay-line based temperature sensor in [24].](image)

![Figure 2.12 Timing diagram of the delay-line based temperature sensor in Figure 2.11.](image)

An improved version of Figure 2.11 is proposed in [22], as shown in Figure 2.13. The physical delay-line length is expanded (multiplied) $P$ times, which is equivalent to the number of $Q$ (a preset and programmable number) in Figure 2.13. The introduction of cyclic delay line has two
purposes. One is for calibration, which will be discussed in Chapter 3. Another is maintaining the same total delay $T_P$ shown in Figure 2.13. The length of the temperature-to-time conversion delay line could be reduced by increasing the number of cycles $P$; or for the same temperature-to-time conversion delay-line length and a larger $P$, the total delay $T_P$ is longer, if a slower reference clock is used. A slower clock means less dynamic power consumed by the counter. In this way, area could be saved at the expense of reduced conversion rate, where the requirement is usually less than 1 kHz.

![Diagram of temperature sensor](image)

**Figure 2.13** The cyclic delay-line based temperature sensor as an improved version of Figure 2.11.

A dual delay-line based temperature sensor is proposed in [21], as shown in Figure 2.14. In Figure 2.14, there are two delay lines: a temperature-dependent delay line and a temperature-independent reference delay line. The latter temperature-independent delay line is made of a temperature-independent delay locked loop (DLL) and it is used to measure the propagation delay of the former temperature-dependent delay line. The single cell in the temperature-dependent delay line is a logic inverter, and its propagation delay has the same characteristic as that in (2.8): the higher the temperature, the slower the delay line propagates. The calibration process that generates digital representations of temperature without being affected by process variations will be discussed in Chapter 3.
Figure 2.14  The dual delay-line based temperature sensor proposed in [21].

2.1.3 Comparison of the state-of-the-art digital temperature sensors, design challenges and considerations

A comparison of the state-of-the-art digital temperature sensors is shown in Table 2.1. As shown in column 2 to 5, the bandgap based temperature sensor does not save area and power proportionally to technology scaling. Comparing references [29], [30] and [33] that are designed by the same group at Delft University of Technology, the area of the 65nm design doesn’t scale proportionally, and its current consumption is higher. Besides, the 65nm design has higher power supply sensitivities and a longer conversion time. The 32 nm bandgap based temperature sensor proposed in [31] consumes a current of 1.6 mA (ADC excluded), compared to 4.6 μA (ADC included) in a 0.16 μm design proposed in [33]. The accuracy of the 32 nm bandgap based temperature sensor is 5 °C (untrimmed), compared to 0.2 °C (trimmed) in a 0.16 μm design [33]. As explained in [31], with device scaling, transistor variations make static measurements difficult as the contributions from voltage offsets and flicker noise dominate the error budget. The larger area overhead is partly due to the extra circuitry required to minimize variations in order to reduce the temperature calibration overhead during high-volume manufacturing. On the
other hand, the delay-line based temperature sensor proposed in 65nm in [40] has a 90 % reduction in its area compared to the delay-line based temperature sensor in 0.13 μm in [21]. Similar to the case for the bandgap based temperature sensor, the delay-line based temperature sensors’ power consumption doesn’t scale proportionally with the technology. One of the explanations is that as technology scales leakage current increases [2], due to thinner gate thickness and higher current density per unit area. Another is that the power supply voltage level doesn’t scale proportionally with the shrinking device size (as listed in Table 2.1), which leads to higher current and power consumption.

A comparison of the bandgap and the delay-line based temperature sensors is shown in Table 2.2. The delay-line based type has smaller area but higher power consumption, compared to its bandgap based counterpart. This is due to the fact that there is always one transistor being charged or discharged in the delay line as the input signal propagates. With a longer delay line or more circulation cycles in the design architect shown in Figure 2.13, the delay-line based temperature sensor could theoretically achieve as a fine resolution as possible. However, the accuracy of the delay-line based temperature sensor could not be improved further, due to the non-linearity of the propagation delay versus temperature characteristics, as indicated in (2.8), which is induced by the non-linearity of the temperature coefficient of the surface carrier mobility in (2.9), as $k_\mu$ could be from 1 to 2 [24]. The delay-line based temperature sensor only generates meaningful digital representations of true temperature after being properly calibrated. This is because the propagation delay of an inverter is not only a function of temperature, but also affected by the process factors and the voltage supply levels. The delay-line based temperature sensors are more sensitive to process and voltage supply variations, compared with their bandgap based counterparts. In contrast, without calibration the bandgap based temperature sensors could achieve accuracies of 0.5 °C in [30] and 5 °C in [31] (32nm). The calibration methods for delay-line based temperature sensors will be discussed in detail in Chapter 3.
## Table 2.1 Comparison of the State-of-the-art Digital Temperature Sensors

![Table 2.1 Comparison of the State-of-the-art Digital Temperature Sensors](image)

<table>
<thead>
<tr>
<th>Sensor Type</th>
<th>Technology</th>
<th>Chip Area (mm$^2$)*</th>
<th>Supply Current*</th>
<th>Supply Voltage (V)</th>
<th>Supply Sensitivity</th>
<th>Temperature Range (°C)</th>
<th>Resolution</th>
<th>Inaccuracy (Trim Method)</th>
<th>Conversion Time (ms)</th>
<th>Power (μW)*</th>
<th>Energy Per Conversion (nJ)</th>
<th>Res.FOM (pJ°C)$^1$</th>
<th>Acc.FOM (nJ%)$^2$</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref Spec</td>
<td>[29]</td>
<td>[33]</td>
<td>[30]</td>
<td>[31]</td>
<td>[21]</td>
<td>[22]</td>
<td>[40]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bandgap</td>
<td>0.16 μm</td>
<td>0.08</td>
<td>3.4 μA</td>
<td>1.5 to 2.0</td>
<td>0.5 °C/V</td>
<td>-55 to 125</td>
<td>0.02</td>
<td>±0.15 °C (1-point)</td>
<td>5.3</td>
<td>6.8</td>
<td>36</td>
<td>11</td>
<td>0.75</td>
<td>2012</td>
</tr>
<tr>
<td>Bandgap</td>
<td>0.16 μm</td>
<td>0.12</td>
<td>4.6 μA</td>
<td>1.6 to 2.0</td>
<td>0.1 °C/V</td>
<td>-30 to 125</td>
<td>0.015</td>
<td>±0.2 °C (1-point)</td>
<td>100</td>
<td>9.2</td>
<td>9200</td>
<td>170</td>
<td>49</td>
<td>2011</td>
</tr>
<tr>
<td>Bandgap</td>
<td>65 nm</td>
<td>0.1</td>
<td>8.3 μA</td>
<td>1.2 to 1.3</td>
<td>0.9 to 1.2°C/V</td>
<td>-70 to 125</td>
<td>0.03</td>
<td>0.2°C (1-point)</td>
<td>454</td>
<td>10.8</td>
<td>4900</td>
<td>4400</td>
<td>196</td>
<td>2010</td>
</tr>
<tr>
<td>CMOS</td>
<td>32 nm</td>
<td>0.02</td>
<td>1.6 mA</td>
<td>1.05</td>
<td>N/A</td>
<td>-10 to 100</td>
<td>0.45</td>
<td>5 °C</td>
<td>1</td>
<td>1680</td>
<td>1680</td>
<td>340</td>
<td>42000</td>
<td>2009</td>
</tr>
<tr>
<td>CMOS</td>
<td>0.13 μm</td>
<td>0.12</td>
<td>1.2 mA</td>
<td>1.2</td>
<td>1.6 °C/mV</td>
<td>0 to 100</td>
<td>0.78</td>
<td>±4 °C</td>
<td>1</td>
<td>1200</td>
<td>240</td>
<td>146</td>
<td>3840</td>
<td>2009</td>
</tr>
<tr>
<td>CMOS</td>
<td>0.18 μm</td>
<td>N/A</td>
<td>80 μA</td>
<td>2.2</td>
<td>N/A</td>
<td>0 to 100</td>
<td>0.133</td>
<td>±0.7 °C</td>
<td>0.2</td>
<td>175</td>
<td>175</td>
<td>3</td>
<td>15</td>
<td>2011</td>
</tr>
<tr>
<td>CMOS</td>
<td>65 nm</td>
<td>0.01</td>
<td>150 μA</td>
<td>1.0</td>
<td>N/A</td>
<td>0 to 60</td>
<td>0.139</td>
<td>± 5 °C</td>
<td>1</td>
<td>150</td>
<td>15</td>
<td>290</td>
<td>375</td>
<td>2011</td>
</tr>
</tbody>
</table>

1 Res.FOM = Energy/Conversion×(Resolution) [29], 2 Acc.FOM = energy/Conversion×(Relative inaccuracy) [29], *for Bandgap type, digital backend power and area are not included, and for [29], ADC is not included.
Table 2.2 A Comparison of the Bandgap and the Delay-line Based Temperature Sensors

<table>
<thead>
<tr>
<th>Type Spec</th>
<th>Area</th>
<th>Power</th>
<th>Resolution</th>
<th>Accuracy</th>
<th>Supply sensitivity</th>
<th>Process variations</th>
<th>Design effort/ported to another technology</th>
<th>Calibration Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandgap</td>
<td>★★★☆</td>
<td>★★★☆</td>
<td>★★★☆</td>
<td>★★★☆</td>
<td>★★★☆</td>
<td>★★★☆</td>
<td>★☆☆☆</td>
<td>★☆☆☆</td>
</tr>
<tr>
<td>Delay-line (CMOS)</td>
<td>★★★☆</td>
<td>★★★☆</td>
<td>★★★☆</td>
<td>★★★☆</td>
<td>★★★☆</td>
<td>★★★☆</td>
<td>★★★☆</td>
<td>★★★☆</td>
</tr>
</tbody>
</table>

The design considerations and future trends for digital temperature sensors are as follows. Due to the large temperature gradients across the modern MPSoC chip, a large number of temperature sensors may be needed. For example, Intel’s 45-nm Dunnington has two temperature sensors located next to each of the six cores [41]. Therefore, the area and power of the temperature sensors should be kept to the minimum. For on-chip thermal management, the temperature sensors must operate in a highly noisy digital environment [41]. The challenges that come with technology scaling are that process variations increase with the scaled geometry and the unscaled noise compared to the scaled current ratio [41]. The future trend is to have multiple small, less accurate (accuracy of 3 to 5 °C), small sized temperature sensors located at as all hotspots of the chip. On the chip there is an accurate sensor that is carefully characterized and trimmed, and the many small sized less accurate temperature sensors are automatically calibrated against this accurate reference [41], as shown in Figure 2.15. An example could be found in [13], where a thermal network is created by an array of accurate bandgap sensors that provide seed temperatures, and the seed temperatures are further up-sampled using temperature readings from the surrounding less accurate delay-line based temperature sensors.
2.2 Temperature Dependence of Delay-line Based Temperature Sensors

2.2.1 Deductions of single CMOS inverter propagation delay

The temperature dependence of the propagation delay of a single inverter cell is affected by the negative temperature coefficient of the surface carrier mobility for electrons and the threshold voltage. The following deductions calculate the single inverter propagation delay using the MOSFET I/V characteristics. Part of the following deductions is in reference to [39].

In Figure 2.16, to discharge the NMOS transistor in the inverter, the inverter output has to drop from $V_{OH}$ to $V_{OL}$:

When $V_o > V_t - V_{TH}$, the NMOS is in saturation region, and its current is:
The time to discharge $V_o$ from $V_{OH}$ to $V_i - V_{TH}$ is:

$$t_{dis1} = \int_{V_i - V_{TH}}^{V_{OH}} \frac{C}{I_{d1}} dV_o$$

$$= \frac{C}{k_n} \int_{V_i - V_{TH}}^{V_{OH}} \frac{1}{[(V_i - V_{TH})V_o - \frac{1}{2}V_o^2]} dV_o$$

$$= \frac{C}{k_n} \cdot \frac{1}{V_i - V_{TH}} \int_{V_i - V_{TH}}^{V_{OH}} \left[ \frac{1}{(V_i - V_{TH} - \frac{1}{2}V_o)} + \frac{2}{V_o} \right] dV_o$$

$$= \frac{C}{k_n} \cdot \frac{1}{V_i - V_{TH}} \left[ 2 \ln \frac{V_o}{V_{OH}} + 2 \ln (V_i - V_{TH} - \frac{1}{2}V_o) \right]$$

$$= \frac{C}{k_n} \cdot \frac{1}{V_i - V_{TH}} \ln \left( \frac{2(V_i - V_{TH} - \frac{1}{2}V_{OL})}{V_{OL}} \right)$$

$$= \frac{C}{k_n} \cdot \frac{1}{V_i - V_{TH}} \ln \left( \frac{2(V_i - V_{TH} - \frac{1}{2}V_{OL})}{V_{OL}} \right)$$

When $V_{OH} = V_{DD}$, $V_{OL} = 0.5V_{DD}$, $V_i = V_{DD}$, the total propagation discharging time of the NMOS is:

$$I_{d1} = \frac{1}{2} k_n (V_i - V_{TH})^2$$

$$k_n = \mu_n C_{OX} \frac{W}{L}$$

(2.11)
\[ t_{ds} = t_{ds1} + t_{ds2} \]
\[
= \frac{C}{k_n (V_{DD} - V_{TH})} \left[ \frac{2V_{TH}}{V_{DD} - V_{TH}} + \ln \frac{1.5V_{DD} - 2V_{TH}}{0.5V_{DD}} \right] \\
= \frac{(L/W)C_L}{\mu C_{ox} (V_{DD} - V_{TH})} \left[ \frac{2V_{TH}}{V_{DD} - V_{TH}} + \ln \frac{1.5V_{DD} - 2V_{TH}}{0.5V_{DD}} \right]
\]

\[
\mu(T) = \mu_0 \left( \frac{T}{T_C} \right)^{-k_\mu}
\]

\[
V_{TH}(T) = V_{TH}(T_C) - \alpha_v (T - T_C)
\]

where \(k_\mu\) and \(\alpha_v\) are treated as positive constants. Part of the above deduction is in reference to [39]. It can be seen from (2.15) that the propagation delay of the delay line is not only a function of temperature, but also affected by process and voltage variations. \(V_{TH}\), \(\mu\), \(W/L\), \(C_L\), \(C_{OX}\) are affected by doping and lithography variations. \(V_{DD}\) is the supply voltage, which is affected by workload conditions during operation.

### 2.2.2 Temperature dependence of delay-line based temperature sensor

In order for the delay-line based temperature sensors to be imported or scaled into another technology effortlessly, and for the temperature sensors to share the same power grid with the digital blocks they are monitoring, they must be all-digital. Therefore, the delay cells used in the delay line must be logic inverters. This thesis proposes a delay-line based temperature sensor as shown in Figure 2.17. This design architecture is in reference to the ring-oscillator based temperature sensor designs published in [37], [43], [44], [45] and [46]. The ring oscillator’s frequency is inversely proportional to the propagation delay of a single inverter cell:

\[
f_0 = \frac{1}{N_{stage} \cdot t_{cell}}
\]

\(t_{cell}\) is the single cell propagation delay. As shown in (2.15), \(t_{cell}\) that is the sum of charging and discharging time has a positive temperature coefficient. Therefore, the ring oscillator’s frequency \(f_0\) has a negative temperature coefficient. The counter quantizes the number of ring oscillator frequency pulses \(f_{out}\) within the fixed reset (gating) time “Reset”, as shown in Figure 2.17.
Figure 2.17 Proposed delay-line based temperature sensor in Section 2.2.2.

Figure 2.18 shows simulation results on the temperature sensor schematic shown in Figure 2.17. The simulation is performed using TSMC’s 65nm CMOS technology in the Cadence Environment, and the simulation setup is listed in Table 2.3. It can be seen from Figure 2.18 that the simulated digital outputs have negative temperature coefficients under corners FF, TT and SS. At different corners, the digital outputs differ at the same temperature, due to the variations in the threshold voltage and the carrier mobility.

Table 2.3 Parameters Setting for the Schematic Shown in Figure 2.17 and Simulation Results Plotted in Figure 2.18

<table>
<thead>
<tr>
<th>Environment</th>
<th>Technology</th>
<th>Voltage</th>
<th>W/L</th>
<th>N\text{stage}</th>
<th>M</th>
<th>Reset period</th>
<th>Reset duty Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cadence Virtuoso Simulation</td>
<td>TSMC 65nm</td>
<td>1V</td>
<td>0.4 μm/1 μm</td>
<td>64</td>
<td>12</td>
<td>400 μs</td>
<td>50%</td>
</tr>
</tbody>
</table>
2.3 Low Power and Small Area Delay-line Based Temperature Sensor

2.3.1 Power and area saving techniques

For CMOS inverters, the majority of the power it consumes is the dynamic power, when its output logic level toggles [42]:

\[ P = C_L \cdot V_{DD}^2 \cdot f \]  \hspace{1cm} (2.19)

And the energy it consumes for one cycle is:

\[ E = C_L \cdot V_{DD}^2 \]  \hspace{1cm} (2.20)

The power given in (2.19) is the total power consumed for an entire charging and discharging cycle [42]. For each cycle, half of the energy supplied by the power source is stored on the inverter’s load capacitor, and the other half is dissipated by the transistors during the charging and the discharging processes [42]. In Figure 2.17, the frequency of the ring oscillator is determined by the single cell propagation and the number of cells \(N_{stage}\) in the ring oscillator:
\[
f = \frac{i_{AVE}}{C_{L,\text{inverter}} \cdot V_{DD} \cdot N} \tag{2.21}
\]

The total power consumed by the number of \( N_{\text{stage}} \) cells in the ring oscillator is:

\[
P_{\text{ring}} = C_{L,\text{ring}} \cdot V_{DD}^2 \cdot \frac{i_{AVE}}{C_{L,\text{inverter}} \cdot V_{DD} \cdot N_{\text{stage}}} \cdot N_{\text{stage}} \tag{2.22}
\]

\[= V_{DD} \cdot i_{AVE} \]

As there is always one transistor, either NMOS or PMOS, being charged or discharged in the delay line, the total power consumed by the number of \( N_{\text{stage}} \) cells in the ring oscillator is actually equivalent to power consumption of a single cell, as indicated in (2.22). In other words, although a ring oscillator comprised of a longer delay line (larger \( N_{\text{stage}} \)) has a lower \( f_{\text{OUT}} \) frequency, its dynamic power remains the same. Another source of the dynamic power comes from the counter in Figure 2.17:

\[
P_{\text{counter}} = V_{DD} \cdot i_{AVE} \cdot \frac{C_{L,\text{counter}}}{N_{\text{stage}} \cdot C_{L,\text{inverter}}} \left( \sum_{i=0}^{M} \frac{1}{2^i} \right) \tag{2.23}
\]

The total power and energy consumed by the temperature sensor are respectively:

\[
P_{\text{conventional}} = V_{DD} \cdot i_{AVE} + V_{DD} \cdot i_{AVE} \cdot \frac{C_{L,\text{counter}}}{N_{\text{stage}} \cdot C_{L,\text{inverter}}} \left( \sum_{i=0}^{M} \frac{1}{2^i} \right) \tag{2.24}
\]

\[
E_{\text{conventional}} = P_{\text{total}} \cdot \frac{1}{f} = \left[ V_{DD} \cdot i_{AVE} + V_{DD} \cdot i_{AVE} \cdot \frac{C_{L,\text{counter}}}{C_{L,\text{inverter}}} \left( \sum_{i=0}^{M} \frac{1}{2^i} \right) \right] \cdot \frac{C_{L,\text{inverter}} \cdot V_{DD} \cdot N_{\text{stage}}}{i_{AVE}} \tag{2.25}
\]

\[= V_{DD}^2 \left[ C_{L,\text{inverter}} \cdot N_{\text{stage}} + C_{L,\text{counter}} \left( \sum_{i=0}^{M} \frac{1}{2^i} \right) \right] \]

From (2.24) and (2.25) it can be seen that increasing the number of stages \( N_{\text{stage}} \) or single inverter cell load capacitance reduces the \( f_{\text{out}} \) frequency. This reduces the dynamic power consumed by the counter at the expense of increased energy per conversion of the ring oscillator. In the meantime, the dynamic power consumption of the ring oscillation remains the same as there is
always one transistor that is being charged or discharged. Reference [46] proposes increasing load capacitance in each single inverter cell. This is done by using a MOS capacitor with its gate connected to the logic inverter output and its drain and source grounded. However, this approach in [46] reduces dynamic power of the counter at the expense of increased energy per conversion of the ring oscillator, as discussed above. Both [44] and [46] propose reducing power consumption using a current-starved ring oscillator, as shown in Figure 2.19. In Figure 2.19, \( V_{BIASP} \) level could be larger than \( GND \), and \( V_{BIASN} \) could be smaller than \( VDD \), so that the transistors consume less current than those in logic inverters. However, the structure as shown in Figure 2.19 cannot be implemented using standard logic cells, as it is not fully-digital. The dynamic power consumed by the counter shown in Figure 2.11 [24] is constant as it is driven by the fixed reference clock (e.g. 50 MHz in [24]). In contrast, the power consumed by the counter in the proposed temperature sensor shown in Figure 2.17 is determined by the oscillation frequency \( f_{out} \) (<10MHz). Therefore, the dynamic power consumed by the counter in the ring-oscillator based design shown in Figure 2.17 is less when compared to the delay-line based counterpart shown in Figure 2.11 in [24].

![Figure 2.19](image)

**Figure 2.19** The current-starved delay cell as proposed in [44] and [46].

This thesis proposes minimizing delay-line based temperature sensor power and area using a tab decoding approach. As shown in Figure 2.20 (b), for a \( M \)-bit temperature sensor, the counter provides the \( (M-N) \) MSBs while the decoder captures the position where the input reset pulse stops within the ring oscillator at the falling edge of “Reset” and generates the remaining \( N \) LSBs. The timing diagrams of the proposed power and area saving techniques and that of the
latch and tab decoding are shown in Figure 2.21 and Figure 2.22, respectively. The power and energy consumed by the proposed architecture shown in Figure 2.20 (b) can be expressed as:

\[
P_{\text{proposed}} = V_{DD} \cdot i_{AVE} + V_{DD} \cdot i_{AVE} \cdot \frac{C_{L,counter}}{N_{stage} \cdot C_{L,inverter}} \left( \sum_{x=0}^{M-N} \frac{1}{2^x} \right) \tag{2.26}
\]

\[
E_{\text{proposed}} = P_{\text{total}} \cdot \frac{1}{f} = \left[ V_{DD} \cdot i_{AVE} + V_{DD} \cdot i_{AVE} \cdot \frac{C_{L,counter}}{C_{L,inverter}} \left( \sum_{x=0}^{M-N} \frac{1}{2^x} \right) \right] \cdot \frac{C_{L,inverter} \cdot V_{DD} \cdot N_{stage}}{i_{AVE}} \tag{2.27}
\]

Both the power and energy expressions of the tab decoding approach shown in (2.26) and (2.27) are less than those of the conventional approach shown in (2.24) and (2.25). This power and energy saving is achieved by employing a counter with a fewer number of bits.

For the tab decoding to work properly, the delay between the neighboring stages (designed to be approximately 140 ns at room temperature) in Figure 2.20 (b) has to be longer than that between the tab and latch.
Figure 2.20 A comparison of the conventional ring-oscillator based temperature sensor (a) and the proposed power saving architecture with tab decoding (b).

Figure 2.21 Timing diagram of the tab decoding technique proposed in Figure 2.20.
2.3.2 Simulation results on proposed power and area saving techniques

Simulation results on power and energy consumptions using conventional and proposed power and area saving techniques are shown in Figure 2.23 (a). The simulation setup is the same as that in Table 2.3. Several observations can be made from Figure 2.23 (a). First, there is a trade-off between power consumption and area for both the conventional and the proposed methods. Second, power decreases with $N_{\text{stage}}$. Third, there is also a trade-off between energy and power consumption, as energy increases with $N_{\text{stage}}$. The theoretical basis for the above observations can be found in the equations (2.25) to (2.27). Figure 2.24 shows that power is reduced as the number of bits $N$ of the decoder increases. The proposed method reduces power without incurring further area, compared with the conventional approach. In other words, for the same area, the proposed method has lower power consumption.
Figure 2.23  Simulation results of power and energy consumptions using the conventional and the proposed power saving methods. The area includes the total area of the ring oscillator, the counter and the tab decoder ($M = 12$).

Figure 2.24  Simulated power is reduced as the number of bits $N$ in the decoder increases.
2.4 Reduction of Digital Outputs Noise in Time Domain

2.4.1 Noise in ring oscillator

Output variations in the time domain are reported in delay-line based temperature sensors [21]. One way to reduce the time-domain variations is by averaging [21]. The time-domain variations are mainly due to the phase noise induced by flicker noise and white noise in the MOSFETs [47]. The total SSB phase noise (Phase noise is a continuous stochastic process indicating random accelerations and decelerations in phase ($\phi$) as an oscillator orbits at a nominally constant frequency ($f_0$) in steady state) resulting from all the inverter cells’ white noise in the ring oscillator is [47]:

$$ L(f) = \frac{1}{4f^2} \left( \frac{f_0}{2MI} \right)^2 M \left( S_{iN}^{W} + S_{iP}^{W} \right) $$

(2.28)

The final SSB phase noise induced by flicker noise is [47]:

$$ L(f) = \frac{1}{16MI^2} \left( S_{iN}^{W} + S_{iP}^{W} \right) \left( \frac{f_0}{f} \right)^2 $$

(2.29)

It can be seen from (2.28) and (2.29) that the SSB phase noise caused by white noise is proportional to the ring oscillator frequency $f_0$, and is inversely proportional to the charging current $I$; the SSB phase noise due to flicker noise is proportional to the ring oscillator frequency $f_0$ and is inversely proportional to the number of cells $M$, charging current $I$ and transistor channel length $L_N$ in the ring oscillator. To suppress time-domain noise in ring oscillator, a longer delay line with larger transistor channel length $L$ and width $W$, could be used.

2.4.2 Time-domain noise reduction for delay-line based temperature sensors

As discussed in the above section, to reduce time-domain noise in the ring-oscillator based temperature sensor, the length of the delay line could be increased or transistors with larger channel length and width could be used. However, increasing the delay line length will increase
chip area and transistor channel length could not be altered in standard cells. Reference [48] indicates that for a ring oscillator with fixed length, the noise caused by timing jitter increases with its reset period $T$ (running time):

$$\Delta t_{\text{jitter}} \propto \sqrt{T}$$  \hspace{1cm} (2.30)

The timing jitter accumulates with the law of $\sqrt{T}$.

Reference [48] proposes reducing time-domain noise by resetting and realigning the oscillator in shorter intervals for a ring oscillator in 90nm. In this thesis, the ring oscillator is reset 8 times during one sampling period (conventionally the ring oscillator is reset once during one sampling period), as shown in Figure 2.25 (a). Compared with the design shown in Figure 2.20, the sensor output number of bits remains the same, as the adder adds up all the 8 outputs from each reset clock period during the sample time, as shown in Figure 2.25 (b). Experimental results on the proposed noise reduction architecture illustrated in Figure 2.25 (a) will be shown in Section 2.5.3.

---

**Figure 2.25** Noise reduction of digital output noise for delay-line based temperature sensors in the time domain. (a) Architecture. (b) Timing diagram.
2.5 Experimental Results

2.5.1 Temperature dependence of delay-line based temperature sensor

Four different temperature sensor architectures: conventional short, conventional long, proposed short, and proposed long are implemented on a 65nm Cyclone III EP3C25F324 FPGA chip. The Cyclone III chip is embedded on a Cyclone III starter development board [49]. The descriptions and measurement results of the four different temperature sensor architectures are listed in Table 2.4. \( N \) refers to the number of bits of the decoder, as labeled in Figure 2.20. Four temperature sensors are implemented on the FPGA at one time, with one conventional long temperature sensor and another three sensors of the same type. To prevent the inverters that made up delay lines from being minimized during compilation in the Quartus™ II software, the connecting wires have to be indicated as (*keep = 1*) [24]. The FPGA sensors are tested at 5 °C increment from 20 °C to 75 °C in an ESPEC™ ETC-3 temperature chamber. Measurement is carried out with a minimum time interval of 15 minutes. A commercial temperature sensor is used as the external accurate temperature reference [50]. The measured digital outputs of the 12 sensors are shown in Figure 2.26. The measurement errors shown in Figure 2.27 are obtained after compared to a linear curve fitting, after calibrations (will be introduced in Chapter 3 in detail) that removes process variations are applied to the measurement results shown in Figure 2.26. The delay-line based temperature sensor resolution is 0.36 °C on the Cyclone III FPGA. The power reported in Table 2.4 is obtained by measuring the total power of 32 sensors of the same prototype on the FPGA chip.

The measurement errors shown in Figure 2.27 resulted from four sources. The first is that the thermal gradient between the temperature of the temperature sensor under test and that of the reference temperature sensor. The second is the nonlinearity of the delay line’s temperature characteristic (non-linearity in the digital outputs versus temperature relationship, as indicated in (2.15)). It could be seen that the errors among sensors of the same prototype are closer to each other than those among sensors of different prototypes. The third comes from the self-calibration procedure (will be discussed in Chapter 3), and this error is less than the sensor resolution (0.36 °C). The fourth is the time-domain digital output noise caused by phase noise, and it could be observed as the inconsistency among measurement errors of sensors of the same prototype.
Table 2.4 Descriptions and Measurement Results of Four Different Temperature Sensor Architectures (M = 12)

<table>
<thead>
<tr>
<th>Prototype no.</th>
<th>Total LE</th>
<th>RO LE</th>
<th>Power (µW)</th>
<th>Errors (°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional short</td>
<td>72</td>
<td>32</td>
<td>8.73</td>
<td>±2.5</td>
</tr>
<tr>
<td>Conventional long</td>
<td>295</td>
<td>256</td>
<td>8.56</td>
<td>±1.5</td>
</tr>
<tr>
<td>Proposed short (N = 2)</td>
<td>60</td>
<td>32</td>
<td>7.88</td>
<td>±3.0</td>
</tr>
<tr>
<td>Proposed long (N = 5)</td>
<td>380</td>
<td>256</td>
<td>2.91</td>
<td>±2.0</td>
</tr>
</tbody>
</table>

Figure 2.26 Un-calibrated digital codes of the 12 sensors, with 3 for each of the four architectures as listed in Table 2.4, on the Cyclone III FPGA chip.
Figure 2.27 Measurement errors of the digital outputs shown in Figure 2.26, using a linear master curve, after a self-calibration method that removes process variations is performed.

2.5.2 Low power and small area delay-line based temperature sensor

Power consumptions are measured on sensors implemented with conventional and with the proposed power saving techniques, respectively. The experimental setup is the same as that in Section 2.5.1. The measurement results are shown in Figure 2.28. The number of logic elements is determined by the number of inverters in the delay line $N_{\text{stage}}$ and that in the counter, as labelled in Figure 2.17. It can be observed from Figure 2.28 that: (1) For both conventional and proposed temperature sensors, the larger the $N_{\text{stage}}$, the less power they consume. (2) The energy per conversion increases with $N_{\text{stage}}$. (3) The proposed tab decoding method as shown in Figure 2.20 reduces both power and energy consumptions for the same number of LE, compared to the conventional sensors. In other words, the proposed method reduces power without incurring further area. The above observations are in correspondence with the calculations given in Section 2.3.1 and the simulation results shown in Figure 2.23. As listed in Table 2.4, compared to the conventional short sensor that uses 72 LE and consumes a power of 8.73 µW, the proposed short one uses 60 LE and consumes a power of 7.88 µW. The proposed long delay-line based
temperature sensor consumes a power 2.91 µW, which is a 70% reduction compared to that consumed by its conventional counterpart.

![Measured Power vs. Area](image)

**Figure 2.28** Experimental results on power and energy consumption using the traditional and proposed power/area saving techniques on Cyclone III FPGA. The area includes the total area of the ring oscillator, the counter and the tab decoder (M = 12).

2.5.3 Reduction of the digital output noise in the time domain

The time domain temperature errors are obtained using 1250 samples from three different architectures listed in Table 2.4 in Section 2.4. Both situations with and without the proposed noise reduction technique are shown in Figure 2.29. There are 8 stages in each sensor, and the x axis represents the number of inverters in each delay cell (stages). For example, when the x axis shows the number of inverters per delay cell is 4, there are $4 \times 8$ (stages) = 32 inverters in the ring oscillator. Observations that could be made from Figure 2.29 are as follows. (1) For all architectures, the measured errors due to time-domain jitter decrease as the length of the delay line increases, as predicted in Section 2.4.1. (2) The proposed power saving technique in Section 2.3 (tab decoding) incurs noise by using additional tabs along the delay line. (3) The proposed noise reduction method in Section 2.4 can effectively reduce the measurement errors due to the
time-domain noise, for all architectures. Table 2.4 shows that for the proposed long architect ($N = 5$), the measurement errors are within 2.0 °C, compared to 1.5 °C for the conventional long. For the proposed short architect (proposed $N = 2$), the measurement errors are within 3.0 °C, compared to 2.5 °C for the conventional short design architect.

**Figure 2.29** Experimental Results show that errors caused by timing jitter are reduced by the method proposed in Section 2.4.2.

2.6 Summary

This chapter presents power saving and time-domain noise reduction techniques for ring-oscillator based delay-line temperature sensors. The power of a delay-line based temperature sensor is minimized using a tab decoding approach. In the proposed approach, for a $M$-bit temperature sensor, the counter provides the $(M-N)$ MSBs while the decoder captures the position where the input reset pulse stops within the ring oscillator at the falling edge of reset signal and generates the remaining $N$ LSBs. Therefore, power is saved by having a counter of fewer bits. Experimental results shown in section 2.5 on a 65nm Cyclone III FPGA verified that the proposed power saving technique saves power up to 70% without incurring additional area, compared with the conventional technique that only uses a counter for decoding in [20].
Chapter 3 Self-calibration Methods for Delay-line Based Temperature Sensors

As discussed in Section 2.2.1, the propagation delay of the delay-line based temperature sensor is not only a function of temperature, but also that of process and supply voltage level variations. This effect from process and supply level variations causes the digital outputs of different temperature sensors to deviate at the same temperature. The above phenomenon is supported by the simulations results as shown in Figure 2.18 and the measurement results as shown in Figure 2.26. The phenomenon has caused the process or voltage variations to be mistaken for a temperature change and this problem could be solved using the self-calibration method that will be proposed in this Chapter. Section 3.1 gives a literature review of the state-of-the-art calibration methods for delay-line based temperature sensors. Sections 3.2, 3.3 and 3.4 propose a self-calibration method that removes the temperature sensors’ sensitivities to supply voltage and process variations. In the proposed self-calibration method, only one calibration block is required to calibrate multiple delay-line based temperature sensors sequentially. For each additional sensor, only additional registers for storing the individual 12-bit correction code, NC, are required. Sections 3.2, 3.3 and 3.4 introduce (I) the process variation removal (II) the voltage supply sensitivity removal and (III) the automatic calibration against an on-chip accurate temperature reference, respectively. The proposed self-calibration method in Sections 3.2, 3.3 and 3.4 is verified by experimental results shown in Section 3.5.

3.1 Literature Review of the State-of-the-Art Calibration Methods

3.1.1 State-of-the-art process variation removal methods for delay-line based temperature sensors

The measured digital outputs, N, are a series of data points plotted as a function of the true temperature as shown in Figure 3.1. If time allows, as many data points as possible should be measured over the temperature range of interest. A first, second or third order master curve function such as \( T = f(N) \) could be use to fit the group data of \( N \). For example, a linear curve fitting function could be expressed as:
\[ T(x) = T_{OFFSET} + \alpha \cdot N(x) \quad (3.1) \]

\( \alpha \) and \( T_{OFFSET} \) are the resolution (°C) and offset (°C) of the temperature sensor. Using the relationship in (3.1), a measurement \( N(x) \) at an unknown temperature could be related to the true temperature. However, as revealed in measurement results shown in Figure 2.26, the digital outputs of different temperature sensors deviate at the same temperature, due to process variations. Therefore each sensor needs to be calibrated individually. To save the calibration time and effort, it is most common to perform a two-point calibration [51] for each temperature sensor, as illustrated in Figure 3.2.

![Figure 3.1 A linear curve fitting for the delay-line based temperature sensor measurements.](image)

Each group of the three temperature sensor’s outputs as illustrated in Figure 3.2 has a different gain and offset:

\[
\begin{align*}
T(x) &= T_{OFFSET} + \alpha \cdot N(x) \\
T'(x) &= T'_{OFFSET} + \alpha' \cdot N'(x) \\
T''(x) &= T''_{OFFSET} + \alpha'' \cdot N''(x)
\end{align*}
\quad (3.2)
\]
Figure 3.2  Two-point calibration method for different temperature sensors.

\( \alpha \) and \( T_{\text{OFFSET}} \) are obtained by solving:

\[
\begin{align*}
T_{c1} &= T_{\text{OFFSET}} + \alpha \cdot N_{c1} \\
T_{c2} &= T_{\text{OFFSET}} + \alpha \cdot N_{c2}
\end{align*}
\]  \hspace{1cm} (3.3)

Then \( \alpha \) and \( T_{\text{OFFSET}} \) are solved as:

\[
\alpha = \frac{T_{c1} - T_{c2}}{N_{c1} - N_{c2}} \tag{3.4}
\]

\[
T_{\text{OFFSET}} = \frac{N_{c1} T_{c2} - N_{c2} T_{c1}}{N_{c1} - N_{c2}} \tag{3.5}
\]

Substituting (3.4) and (3.5) into (3.2) and using a similar approach for \( \alpha' \), \( \alpha'' \), \( T_{\text{OFFSET}}' \) and \( T_{\text{OFFSET}}'' \) we obtain:

\[
\begin{align*}
T(x) &= \frac{N_{c1} T_{c2} - N_{c2} T_{c1}}{N_{c1} - N_{c2}} + \frac{T_{c1} - T_{c2}}{N_{c1} - N_{c2}} \cdot N(x) \\
T'(x) &= \frac{N'_{c1} T_{c2} - N'_{c2} T_{c1}}{N'_{c1} - N'_{c2}} + \frac{T_{c1} - T_{c2}}{N'_{c1} - N'_{c2}} \cdot N'(x) \\
T''(x) &= \frac{N''_{c1} T_{c2} - N''_{c2} T_{c1}}{N''_{c1} - N''_{c2}} + \frac{T_{c1} - T_{c2}}{N''_{c1} - N''_{c2}} \cdot N''(x)
\end{align*}
\]  \hspace{1cm} (3.6)
Each temperature sensor needs to be calibrated individually as $N_{CI}$, $N'_{CI}$ and $N''_{CI}$ in (3.6) are different. If all sensors can be aligned to have the same gain and offset $\alpha$ and $T_{OFFSET}$, one set of $\alpha$ and $T_{OFFSET}$ can be applied to multiple sensors and in this way the calibration effort can be greatly saved (there is no need to calibrate each sensor individually). Figure 3.3 illustrates the above scenario. It shows the temperature sensor outputs before and after the one-point calibration, when the process variations (inconsistencies) among the temperature sensors’ outputs are removed. The labels $\alpha$, $T_{OFFSET}$, $N_{CI}$, $N'_{CI}$, $N''_{CI}$ in Figure 3.3 are the same as those in Figure 3.2 and in (3.2), (3.3), (3.4) and (3.5). The removal of the above inconsistencies could be achieved by the one-point calibration method proposed in [21]. After the removal of process variations, all the temperature sensors have the same gain $\alpha$ and offset $T_{OFFSET}$.

Figure 3.3 Illustration of digital outputs before and after the one-point calibration.
The one-point calibration method proposed in [21] is based on dual delay lines. The dual delay-line architecture is shown in Figure 3.4. As discussed at the beginning of this section, the propagation delay of a single delay cell is a function of temperature, process factors and power supply level. The one-point calibration in [21] is based on the assumption that the temperature sensor propagation delay could be separated into two parts: the temperature-dependent part and the temperature-independent part (affected by process factors and supply voltage level):

\[ D(T, P, V) = T^{-a} \cdot G(P, V) \]  

(3.7)

\( G(P, V) \) is affected by process and voltage supply variations. For the same temperature sensor, \( G(P, V) \) is assumed to be the same over the temperature range, although in fact it is a weak function of temperature [21]. Thus a normalized delay that is only dependent on temperature can be obtained as:

\[ D_{\text{norm}}(T) = D(T, P, V) / D(T_c, P, V) = (T / T_c)^{-a} \]  

(3.8)

where \( D_{\text{norm}}(T) \) depends only on temperature (independent of process variations), as \( G(P, V) \) is assumed to be the same for the same sensor over the entire temperature range. The one-point calibration method proposed in [21] has a calibration mode where the delay of the temperature-dependent open-loop delay line (OLDL) is adjusted to be the same for all the sensors at the calibration temperature \( T_c \). This is achieved by comparing the delay of the OLDL to that of a delay locked loop (DLL). This DLL delay is designed to be independent of temperature, in the calibration mode. At the beginning of the calibration sequence, the delay \( D_{\text{DLL}} \) of the reference line shown in Figure 3.4 is fixed by making \( N = N_C \) in MUX-1:

\[ D_{\text{DLL}} = \Delta_0 \cdot N_C \]  

(3.9)

Then at the calibration temperature (50 °C) \( M \) in MUX-2 is increased until the delay of the OLDL is equivalent to \( D_{\text{DLL}} \) (delay of the reference line) in (3.9) when \( M = M_C \):

\[ T_c^{-a} \cdot G(P, V) \cdot M_c = D_{\text{DLL}} = \Delta_0 \cdot N_C \]  

(3.10)

In (3.10), \( \Delta_0 \) (the single cell propagation delay in the DLL, as shown in Figure 3.4) and \( N_C \) are both constants despite process variations and temperature changes. \( G(P, V) \) is a unique factor of
each sensor, but is assumed to be constant over the temperature range. So $M_C$ that is inversely proportional to $G(P, V)$ is a unique number associated with each individual sensor. During the measurement mode, the pulse width of the OLDL is measured by the DLL. $N$ in MUX-1 is increased until the delay of the reference line is equivalent to that of the OLDL when $N = N_M$:

$$T^{-\alpha} \cdot G(P, V) \cdot M_C = D_{DLL} = \Delta_0 \cdot N_M$$  \hspace{1cm} (3.11)

Eliminating the common terms $G(P, V)$, $\Delta_0$, and $M_C$ in (3.10) and (3.11), the relationship between $N_C$ and $N_M$ is:

$$\left( \frac{T}{T_C} \right)^{-\alpha} = \frac{N_M}{N_C}$$  \hspace{1cm} (3.12)

$$N_M = N_C \left( \frac{T}{T_C} \right)^{-\alpha}$$  \hspace{1cm} (3.13)

where $N_M$ is a function of temperature only, as $N_C$ is a constant. A graphical illustration of the calibration and measurement modes in [21] is as shown in Figure 3.5. $M_C$ is configured by a hard-wired setting once the calibration is done. The temperature sensor in [21] has a resolution of 0.78 °C and it consumes a power of 1.2 mW, occupying an area of 0.12 mm$^2$ when implemented using a 0.13 µm CMOS technology.
Figure 3.4  The dual delay-line based temperature sensor proposed in [21].

Figure 3.5  Calibration mode (upper) and measurement mode (lower) in [21].
The post process variation removal codes in Figure 3.3 need to be correlated with the true temperature by a master curve. The master curve can be the same for the temperature sensors fabricated in the same batch or using the same technology. It is obtained by curve fitting of the temperature sensors’ output code measurements, e.g. at every 10 °C from 0 to 100 °C in [21]. The master curve could be linear, second order [22] or third order [21]. In [21], a third-order master curve is used, and the deviation of the measurement results from the master curve is within ±4 °C.

Making use of the same calibration principle as proposed in (3.8) that the propagation delay could be separated into temperature-dependent and temperature-independent components [21], Chen et al. proposed a one-point calibration method for a cyclic delay-line temperature sensor architecture, as shown in Figure 3.6 [22]. The cyclic delay-line based temperature sensor shown in Figure 3.6 was previously discussed in Section 2.1.2. The propagation delay $T_{OSC}$ of the delay-line is expanded (multiplied) by the number of cycles $N$. This expansion of the delay $T_{OSC}$ is achieved by the circuit (a programmable time amplifier) shown in Figure 3.7. The operating principle of Figure 3.7 is described as follows. First of all, the START signal resets the down counter’s outputs to all logic high and the two D flip-flops’ outputs to logic low. During the number of $N$’s clock period, the down counter is clocked by the oscillatory delay $T_{OSC}$, and the $NAND$ gate’s output is logic low. When the counter counts to zero, its outputs are all at logic low, and the $NAND$ output rises to logic high. $DFF1$ is then driven by a positive clock edge and its output $Q$ goes from logic low to logic high. $DFF2$ is driven by a positive clock edge and its output $Qb$ goes from logic high to logic low. Figure 3.8 shows that the expanded delay $T_A$ is generated from an $AND$ gate driven by its inputs $RESET$ and $EOC$ (the $DFF2 Qb$ output). $DFF1$ is there for deglitching. To make expanded delay $T_A$ the same for all the sensors the calibration temperature (e.g. 50 °C), the number of circulation cycles $N$ for the cyclic delay line is adjusted such that:

$$D_{out} = \frac{T_{A,T_6}}{t_{ref}} = \frac{N \cdot T_{OSC,T_6}}{t_{ref}} = D_{out,0}$$

(3.14)

$D_{out,0}$ is the same for all temperature sensors, and $t_{ref}$ is the reference clock period shown in Figure 3.6. The procedure in (3.14) is performed using an off-chip calibration circuit in the gray dotted square in Figure 3.6. The number $N$ is increased based on a successive approximation
pattern such that when the digital outputs $D_{out}$ are equivalent to a constant value $D_{out,0}$ (the same for all the sensors), the associated $N$ is stored in an on-chip register ($P=N$) to keep number of the cycles and the off-chip calibration circuit is not in use any more. As in (3.14) $T_{OSC,T0}$ may be different from sensor to sensor due to process variations, and $N$ is a unique number for each individual sensor.

**Figure 3.6**  One-point calibration for cyclic delay-line based temperature sensor in [22].

**Figure 3.7**  Schematic of the programmable time amplifier in [22].
Then at any temperature $T$, the digital outputs $D_{out}$ are:

$$D_{out} = \frac{N \cdot T_{OSC,T}}{t_{ref}}$$  \hspace{1cm} (3.15)

Eliminating the common terms in (3.14) and (3.15):

$$\frac{D_{out}}{D_{out,0}} = \frac{T_{OSC,T}}{T_{OSC,T_0}}$$ \hspace{1cm} (3.16)

$T_{OSC}$ can be expressed as:

$$T_{OSC} = t_{cell} \cdot N_{stage}$$ \hspace{1cm} (3.17)

where $t_{cell}$ is the single cell propagation delay and $N_{stage}$ is the number of cells in the ring oscillator, as shown in Figure 3.6. Substituting (3.17) into (3.16), (3.16) can be re-expressed as:

$$\frac{D_{out}}{D_{out,0}} = \frac{T_{OSC,T}}{T_{OSC,T_0}} = \frac{t_{cell,T} \cdot N_{stage}}{t_{cell,0} \cdot N_{stage}} = \frac{t_{cell,T}}{t_{cell,0}}$$ \hspace{1cm} (3.18)

Considering in (3.7) that the single cell propagation delay could be expressed as the product of a temperature-dependent part and a temperature-independent part, (3.18) can be re-expressed as:

$$\frac{D_{out}}{D_{out,0}} = \frac{t_{cell,T}}{t_{cell,0}} = \left(\frac{T}{T_0}\right)^{-\alpha}$$ \hspace{1cm} (3.19)

Then $D_{out}$ only depends on temperature, as $D_{out,0}$ and $T_0$ are constants.
A conventional way to detect process variations is using process sensors. Process sensors in [52], [53] and [54] detect threshold variations by the change or the absolute value in the output voltages in the circuits analogous to that in Figure 3.9. The change in sense voltage $V_X$ is related to the threshold voltage of the DUT in Figure 3.9. The sense voltage could be amplified by a common amplifier [53] or digitalized by a VCO (voltage controlled oscillator) [54]. However, the process sensors as shown in Figure 3.9 are not suitable for calibrating temperature sensors, for the following reasons. First, in addition to the threshold voltage $V_{TH}$, there are many other parameters affecting the propagation delay that are subject to process variations. These include $\mu$, $W/L$, $C_L$, and $C_{OX}$, which could be affected by doping and lithography variations. Second, for delay-line based sensors made of a large number of inverters and for VLSI chips where a large amount of sensors are integrated, each transistor needs to be calibrated. In this case, both the calibration time and effort are.

![Threshold voltage sensing circuit in the process sensor proposed in [53].](image)

3.1.2 State-of-the-art voltage variation removal methods for delay-line based temperature sensors

After the removal of process variations, the digital outputs of the temperature sensor in (3.13) [21] and (3.19) [22] are still sensitive to power supply level changes [21]. The same observations are made in (2.15) in Section 2.2.1, where the temperature sensor outputs are affected by voltage supply level. In this case, after the process variations are removed as proposed in Section 3.1.1, two temperature sensors only have the same outputs at the same temperature and power supply level. But when power supply voltage levels are different, they will have different output codes, and the power supply level change may be mistaken for a temperature change. The power supply variations occur if the workload changes during runtime. Experiments on a Xilinx Virtex-5
FPGA verify that an instantaneous change from 0 to 80 % utilization running at 100 MHz leads to a 73 °C error in the estimated temperature [23]. Reference [21] reports that delay-line based temperature sensors on a 0.13 µm custom IC have a $\Delta T/\Delta V_{DD}$ (temperature sensing error to supply voltage) sensitivity ratio of 1.6 °C/mV, which translates to a 80 °C error when the supply voltage changes by 50 mV.

Reference [55] proposes reducing power supply sensitivity by using differential capacitor delays. As single inverter propagation delay could be expressed as [55]:

$$t_d = \frac{C_L \cdot V_{DD}}{2I}$$  \hspace{1cm} (3.20)

$I$ is the average charging/discharging current. Therefore if $I$ is proportional to $V_{DD}$, a voltage-supply insensitive delay line could be obtained. Figure 3.10 shows the block diagram of the power supply insensitive temperature sensor proposed in [55]. In Figure 3.10, the upper capacitor array is charged by current with negative temperature coefficient and generates delay having positive temperature coefficient. The lower capacitor is charged by current with positive temperature coefficient and generates delay having negative temperature coefficient. Both are controlled by the same $START$ signal. The output delays of the two capacitor units are XORed to generate the final time-domain output. Figure 3.11 shows the circuit that generates the temperature dependent delay. When $START$ is at logic low, the capacitor voltage is at logic high. When $START$ rises to logic high, the capacitor voltage is discharged by the current in the transistor labelled $(W/L)_2$. The time for the capacitor voltage to transit from logic high to logic low is inversely proportional to its discharging current. If the discharging current $I_2$ has a negative temperature coefficient, then the capacitor output delay has a positive temperature coefficient and vice versa.
Figure 3.10  Block diagram of the temperature sensor in [55].

Figure 3.11  Capacitor delay determined by the discharging current in [55].
In Figure 3.10 [55], the current in the upper delay line $I_{bias1}$ and the current in the lower delay line $I_{bias2}$ have negative and positive temperature coefficient, respectively. Both of the biasing circuits that generate $I_{bias1}$ and $I_{bias2}$ are shown in Figure 3.12. For certain value range of $V_g$, $I_{bias1}$ has negative temperature coefficient:

$$I_{bias1} = \mu C_{OX} \frac{W}{L} [(V_g - V_{TH})V_x - \frac{1}{2}V_x^2]$$  \hspace{1cm} (3.21)
The value of $V_g$ is selected so that the current expressed in (3.21) has negative temperature coefficient. The capacitor generated delay is inversely proportional to its biasing current and it has positive temperature coefficient. In Figure 3.12, $V_X$ within the blue square (generating negative delay) can be expressed as:

$$V_X \approx V_b - \frac{V_{TH}}{A}$$  \hspace{1cm} (3.22)

It is well known that $V_{TH}$ is affected by negative temperature coefficient, as discussed in (2.10) in Section 2.1.2. Substituting (2.10) into (3.22):

$$V_X \approx V_b - \frac{V_{TH0} + \alpha_v (T - T_0)}{A}$$  \hspace{1cm} (3.23)

Then $I_{bias2}$ is:

$$I_{bias2} = \frac{V_X}{R} \approx \frac{V_b - \frac{V_{TH0} + \alpha_v (T - T_0)}{A}}{R}$$  \hspace{1cm} (3.24)

where $I_{bias2}$ has positive temperature coefficient. Both $I_{bias1}$ and $I_{bias2}$ are proportional to $V_b$, and $V_b$ is proportional to the supply voltage level. As shown in Figure 3.12 that $V_b$ is generated from a voltage divider that is made of an array of diode connected transistors. The two capacitor delays are XORed to generate $DXOR$ that eventually has negative temperature coefficient. The delay $DXOR$ (when $V_X \approx V_b$ by using large Op amp gain $A$) is:

$$t_{DXOR} = \frac{C_L V_{DD}}{2} \left( \frac{1}{I_{bias2}} - \frac{1}{I_{bias1}} \right)$$

$$\approx \frac{C_L V_{DD}}{2} \left[ \frac{R}{V_b - \frac{V_{TH0}}{A} + \frac{\alpha_v (T - T_0)}{A}} - \frac{1}{\mu_n C_{OX} \frac{W}{L} [(V_g - V_{TH0} + \alpha_v (T - T_0))V_b - \frac{1}{2} V_b^2]} \right]$$  \hspace{1cm} (3.25)

Since $V_b \propto V_{DD}$, then $t_{DXOR}$ is approximately insensitive to voltage supply level. However, in (3.25), there are temperature dependent factors ($T$) and second order factors ($V_b^2$), which makes $t_{DXOR}$ [55] slightly voltage supply level sensitive. The reported temperature sensing error to supply voltage sensitivity is less than 1 °C/100mV (10 °C/1 V), according to the simulation
results in 0.35 µm standard technology in [55]. The design in [55], albeit effective, is not suitable for implementing with standard cells as it is not a fully digital design.

Reference [56] proposes a PVT (process, voltage and thermal) sensing system. There are three sensors in the system: a process sensor, a voltage sensor, and a temperature sensor. The temperature sensor in [56] is based on MOSFETs working in subthreshold region (weak inversion). The current of a MOSFET working in weak inversion has an exponential relationship with temperature, and in this way the MOSFET is similar to a bipolar transistor [57]. The subthreshold leakage based temperature sensor proposed in [56] is shown in Figure 3.14. The current in the resistor $R$ and transistor $N2 P2$ could be expressed as [56]:

\[
I_{PTAT} = \frac{mV_T}{R} \ln \left( \frac{W_{p1}W_{N2}/L_{p1}L_{N2}}{W_{p2}W_{N1}/L_{p2}L_{N1}} \right)
\]

(3.26)

where $m$ depends on process. The ring oscillator current is controlled by the current in (3.26) and its frequency has a positive temperature coefficient. This temperature dependent frequency is quantized by a counter and a reference clock, as shown in Figure 3.14.

In [56], the architecture of the process sensor is analogous to that of the temperature sensor except the gate voltage levels are different in the current starved inverters in the ring oscillator. In the process sensor shown in Figure 3.15, the gate voltage levels are equal to the simulated values to make the delay of the ring oscillator temperature independent.

![Figure 3.14](image)

**Figure 3.14** Subthreshold leakage current based temperature sensor proposed in [56].
The temperature coefficient of the delay in the current-starved cell shown in Figure 3.15 is determined by the combined temperature coefficients in the surface carrier mobility $\mu$ and the threshold voltage $V_{TH}$. Since the threshold voltage $V_{TH}$ has a positive temperature coefficient while the carrier mobility $\mu$ has a negative temperature coefficient, a different gate voltage ($VBIASP$ or $VBIASN$) could change the contributing portion of their temperature coefficients, generating either positive ([46], 4)) or negative temperature coefficient ([21] [22]) delays. Within the voltage range $(0, V_{DD})$, the higher the gate voltage of a NMOS, the more positive is the temperature coefficient of the delay. When the gate voltage increases from 0 to $V_{DD}$, the delay’s temperature coefficient changes gradually from negative, to zero temperature point (ZTC), and then to positive. ZTC occurs for certain combinational values of $VBIASP$ and $VBIASN$. The $VBIASP$ and the $VBIASN$ values are found through simulations, and in [56] they are 0.4 V and 0.6 V for NMOS and PMOS respectively, when the supply voltage level is 0.5 V. The circuit in Figure 3.15 converts the process dependent delay into digital representations.

The voltage sensor in [56] has a single cell structure as shown in Figure 3.16. The discharging current of the inverter made of M1 and M2 is proportional to $Vin$. The inverter delay is inversely proportional to its discharging current. The delay speed is converted into a pulse position along the delay line, and the pulse position is quantized using a decoding and latch circuit shown in Figure 3.16.
Figure 3.16  Schematic of the voltage sensor in [56].

Reference [56] doesn’t mention how the process, temperature and voltage sensors compensate each other. However, there are several limitations with the design in [56]. The voltage sensor is affected by temperature and process variations. The relationships between outputs of the three sensors could be expressed by:

\[
\begin{align*}
D_T &= \alpha \cdot f(T) \\
D_G &= \beta \cdot g(G) \\
D_V &= \gamma \cdot f(T) \cdot g(G) \cdot h(V)
\end{align*}
\]  \( (3.27) \)

where \( D_T, D_G, D_V \) are the outputs of the temperature, process and voltage sensors, respectively. One of the limitations is that each sensor needs to be calibrated at one point (for example, the voltage sensor needs to be calibrated at a known voltage level) to obtain factors \( \alpha, \beta, \gamma \) in (3.27). Another limitation is that the process, temperature and voltage conditions might not be the same for all the three types of sensors. In this case they cannot fully compensate for each other’s variations. For example, in larger chips, the thermal gradient might be larger than 10 °C [4]. The voltage levels may differ as well. The process factors also vary across the chip due to doping and etching non-uniformities, and the problem increases as technology scales.
3.2 Proposed Self-calibration Step I: Process Variation Sensitivity Removal

3.2.1 Theoretical analysis for proposed process variation removal method for delay-line based temperature sensors

This section proposes a self-calibration method for multiple delay-line based temperature sensors intended for VLSI thermal management applications. The method compensates the effect of process variations by assigning a unique correction factor, $N_C$ to each sensor, making all the sensors have the same outputs at start-up. This correction factor $N_C$ is computed automatically by an on-chip low power successive approximation (SA) algorithm circuit.

Based on the assumption that the temperature sensor propagation delay could be divided into two parts: the temperature-dependent part and the temperature-independent part (affected by process variations), as indicated in (3.7)[21]:

$$D(T, P, V) = T^{-\alpha} \cdot G(P, V)$$

(3.28)

For the same temperature sensor, $G(P, V)$ is assumed to be the same over the temperature range, although in fact it is a weak function of temperature [21]. Further analysis on this weak dependence will be discussed later in this section.

To calibrate all the sensors’ codes to be the same at start-up, a correction factor $N_C$ for each delay-line sensor is computed. This correction factor $N_C$ is the quotient of a stored constant value $C(T_C)$ and the coarse (un-calibrated) codes $D(T_C, P, V)$:

$$N_C = \frac{C(T_C)}{D(T_C, P, V)}$$

(3.29)

Due to process variations, the un-calibrated outputs $D(T_C, P, V)$ differ from sensor to sensor. These variations could be observed in the experimental results shown in Figure 2.18 and Figure 2.26. Therefore $N_C$ is a unique number for each sensor. Each sensor’s $N_C$ is then multiplied with its un-calibrated outputs $D(T_1, P, V)$ to become the calibrated outputs. Based on (3.29), at an unknown temperature $T_1$, the calibrated outputs become:
\[ C(T_i) = D(T_i, P, V) \cdot N_C \]
\[ = C(T_C) \frac{D(T_i, P, V)}{D(T_C, P, V)} \]  
(3.30)

Assuming for the same temperature sensor, \( G(P, V) \) is the same for the outputs \( D(T_i, P, V) \) and \( D(T_C, P, V) \) in (3.28), (3.30) can be expressed as:

\[ C(T_i) = D(T_i, P, V) \cdot N_C \]
\[ = C(T_C) \frac{D(T_i, P, V)}{D(T_C, P, V)} \]
\[ = C(T_C) \left( \frac{T_i}{T_C} \right)^{-a} \]  
(3.31)

As \( C(T_C) \) is the same constant number for each sensor, \( C(T_i) \) is only dependent on temperature. The effectiveness of the proposed self-calibration method as described in (3.31) is verified in MATLAB mathematically. Random temperature readings are generated in MATLAB as un-calibrated codes following the pattern as shown in (3.28), at every 5 °C from 0 to 100 °C. 32 groups of readings are generated. The simulated digital outputs of 32 temperature sensors before and after the process variation removal are as shown in Figure 3.17. The self-calibration method proposed in (3.31) is also verified using measured un-calibrated outputs from a Cyclone III FPGA chip in Figure 2.26 as an input file, and the simulated outputs before and after calibration in MATLAB is shown in Figure 3.18. Despite the case that the process variations are totally removed using mathematically generated inputs in Figure 3.17, there are still inconsistencies in Figure 3.18, when real measurement data from the Cyclone III FPGA are used as the un-calibrated input. One of the reasons is that the assumption made earlier that \( G(P, V) \) is a constant over the temperature range for each individual sensor is invalid.
Figure 3.17  MATLAB verification of the self-calibration method proposed in (3.31) using randomly outputs generated following the pattern described in (3.28).
Figure 3.18 MATLAB verification of the self-calibration method proposed in (3.31) using measured un-calibrated outputs from Cyclone III FPGA chip shown in Figure 2.26.

To explore the variations of $G(P,V)$ over temperature and process variations, the expression of the inverters’ propagation delay given in (2.15) is further analyzed. The ratio of the un-calibrated outputs $D(T_1, P)$ at an unknown temperature $T_1$ and $D(T_C, P)$ at a reference temperature $T_C$ is (when the ratio is approximately linear over the temperature range):

$$\frac{D(T_1, P)}{D(T_C, P)} = 1 - (T_1 - T_C) \cdot H(T_1)$$

(3.32)

$$H(T_C) = \frac{k_\mu}{T_C} - \alpha_s \cdot F(T_C)$$

(3.33)

$$F(T_C) = \left\{ \frac{V_{DD} + V_{TH}(T_C)}{[V_{DD} - V_{TH}(T_C)]V_{TH}(T_C)} - \frac{2}{V_{DD}^2} [V_{DD} - V_{TH}(T_C)] \right\}$$

(3.34)

$$\mu(T_1) = \mu_0 \left( \frac{T_1}{T_C} \right)^{-k_\mu}$$

(3.35)
\[ V_{\text{TH}}(T_i) = V_{\text{TH}}(T_C) - \alpha_c (T_i - T_C) \]  

(3.36)

where \( k_\mu \) and \( \alpha_c \) are the temperature coefficient of the electron surface carrier mobility and that of the threshold voltage, respectively. Both coefficients are positive in this analysis. \( H(T_i) \) (unit: \( ^\circ\text{C}^{-1} \)) is proportional to the temperature sensor gain. Assuming that \( H(T_i) \) is process independent, the ratio \( D(T_1, P)/D(T_C, P) \) is only dependent on temperature.

The mathematical analysis from (3.32) to (3.36) is supported with the following simulations in Figure 3.19 using the setup listed in Table 2.3 in Section 2.2.2, to further explore the temperature and process dependences of \( H(T) \) that determine the temperature sensor gain. The simulations in Figure 3.19 are performed using TSMC’s 65nm CMOS technology in the Cadence Environment. TT, FF and SS corners are represented by red, green and blue solid lines respectively. Figure 3.19 (a) and (b) show the temperature and process dependences on the threshold voltage \( V_{\text{TH}} \) and the surface carrier mobility \( \mu \), respectively. Figure 3.19(c) shows that the process dependence of the \( F(T) \) (3.3 \( \text{V}^{-1} \) to 3.6 \( \text{V}^{-1} \)) is much less than that of the threshold voltage (0.33 V to 0.49 V). Or, \( F(T) \) is a weak function of the threshold voltage. In Figure 3.19 (d) the three solid lines are \( H(T_i) \) in three corners, showing slight dependence on temperature and on the process corners. The dashed line represents the constant ratio \( H(T_C) \) used in (3.34). The errors resulted from treating \( H(T_i) \) as a constant, \( H(T_C) \), are plotted in Figure 3.19 (e) when the self-calibration is performed at 50 \( ^\circ\text{C} \). Even though \( H(T_i) \) varies with both process and temperature, as shown in Figure 3.19 (d), the errors from treating it as a constant \( H(T_C) \) are within \( \pm 7^\circ\text{C} \), as demonstrated in Figure 3.19 (e).
Figure 3.19  The effect of process variations on the threshold voltage, surface carrier mobility, $H(T_c)$ and the errors resulted from treating $H(T_c)$ as constant despite process variations. The simulations are in three corners: TT, FF, SS.
3.2.2 Circuit architectures and simulation results in ModelSim

Figure 3.20 shows the circuit architecture of the self-calibration method proposed in Section 3.2.1. The architecture consists of two parts: the four temperature sensors as shown in Figure 3.20 (a) and the self-calibration circuit as shown in the middle of Figure 3.20 (b). The temperature sensor architecture is the same as that in Figure 2.17, which has been discussed and simulated in Section 2.2.2 and experimentally verified in 2.5.1. For the proposed self-calibration method, there are three steps: (1) the removal of process variations (as shown in the red dotted square in Figure 3.20 (b), and will be discussed in this section); (2) the removal of voltage variations (will be discussed in Section 3.3); (3) the continuous calibration with an accurate reference to generate the true temperature (will be discussed in Section 3.4). The calibration procedures of the first step (removal of process variations) are as follows. Firstly the uncalibrated outputs of one of the sensors are selected as $D_T(T)$ at a time, as the input for the correction factor $N_C$ solving circuit. All the sensors’ $N_C$ are then computed sequentially during start up. Once each $N_C$ is solved, it is stored in a register. The multiplying circuit also works sequentially to generate the calibrated outputs $C(T)$. The dividing algorithm that solves $N_C$ as in (3.29) is not directly synthesizable using Verilog. It is computed using a Successive Approximation (SA) algorithm as shown in Figure 3.21.

The procedures of solving $N_C$ as calculated in (3.29) are shown in Figure 3.21 and are described as follows. In order to divide $C(T_C)$ by $Cal_codes\ D(T_C)$ that are both of the same order of magnitude, $C(T_C)$ is shifted by $M$ digits to left ($M = 11$ in this design) and becomes $C'(T_C)$. This will serve as the dividend in the SA dividing circuit. In this way, $C(T_C)$ is magnified by the times of $2^M$. Therefore, a reasonable resolution for the quotient, $N_C$ can be obtained. A larger $M$ results in a finer resolution of $N_C$ and a smaller $M$ reduces the computation power and logic element usage of the SA dividing circuit. The SA division process begins as follows. $Q[11:0]$ is a trial quotient, and at the beginning, its MSB is 1 and its remaining bits are all 0. Initially, the product of the trial quotient, $Q$ and $Coarse_codes$ is compared with $C'(T_C)$. If $C'(T_C)$ is the smaller of the two, it means that $N_C$ should be smaller than the current value of $Q$. Therefore, $N_C$’s MSB should be logic 0, otherwise its MSB should be logic 1. For the former case, in the next step, $C'(T_C)$ equals to itself, and for the latter, it is equivalent to itself minus the product of $Q$ and $Coarse_codes$. In the next step, $Q$’s second MSB is 1 and its remaining lower bits are all 0. The same comparison as in the previous step repeats until $Q$’s LSB is 1 and $n$ shown in Fig. 3 is 0.
This SA dividing process computes $N_C$ only once at start-up. $N_C$ is then multiplied with $D(T)$ and moved $M$ bits to the right (divided by $2^M$) to generate the calibrated codes $Cal_codes C(T_i)$. As explained above and indicated in (3.31), at start-up $Cal_codes$ of each individual temperature sensor is equal to the stored value $C(T_C)$ and the calibrated $Cal_codes C(T_i)$ at any other temperature contains no process variations. The SA circuit could calibrate as many sensors’ $N_C$ as required, with the expense of additional $N_C$ registers.

![Temperature sensor schematic](image)

(a) Temperature sensor schematic

![Block diagram](image)

(b) Block diagram

Figure 3.20 System level diagram of the proposed self-calibration block that calibrates multiple temperature sensors.
Figure 3.21  The SA algorithm and multiplying circuitry.

The self-calibration method (including the SA dividing algorithm) is implemented using Verilog code and is verified by ModelSim software simulation, as shown in Figure 3.22. The un-calibrated outputs are from four temperature sensors implemented on a Cyclone III FPGA. The temperature sensor schematic is the same as that shown in Figure 3.20 (a). The un-calibrated outputs are written as verilog codes as a test bench source file. The un-calibrated outputs’ change over temperature is converted into their change over time in the ModelSim test bench. Screenshots of the ModelSim simulation can be found in Appendix B.
3.3 Proposed Self-calibration Step II: Voltage Variation Sensitivity Removal

After the removal of process variations, the calibrated outputs as shown in (3.31) are still sensitive to power supply level changes, as $G(P, V)$ is a function of voltage level. In other words, after the process variations are removed in Section 3.2, two temperature sensors will have the same outputs at the same temperature and power supply level. But when their power supply voltage levels are different, they will have different output codes. In this case, the power supply level change is mistaken for a temperature change. The power supply variations could occur if the workload changes during runtime. Experiments on a Xilinx Virtex-5 FPGA verify that an instantaneous change from 0 to 80% utilization running at 100 MHz leads to a 73 °C error in the estimated temperature [23]. Reference [21] reports that delay-line based temperature sensors on a 0.13 µm custom IC have a $\Delta T/\Delta V_{DD}$ (temperature sensing error to supply voltage) sensitivity ratio of 1.6 °C/mV, which translates to a 80 °C error when the supply voltage changes by 50 mV. In Section 3.2.1 $C(T_1)$ in (3.31) is shown to be vulnerable to power supply level changes:

Figure 3.22 The removal of process variations in ModelSim simulation. The un-calibrated outputs are from the measurements of four temperature sensors on a Cyclone III FPGA.
\[ C'(T_i) = C(T_i) \frac{D(T_i, P, V_{DD,M})}{D(T_i, P, V_{DD,C})} \]

(3.37)

where \( D(T_c, P, V_{DD,C}) \) and \( D(T_c, P, V_{DD,M}) \) are the un-calibrated codes before and after the supply level changes, respectively. Equation (3.37) indicates that when calibration and measurement are not performed at the same voltage level, measurement errors will occur. For example, when Dynamic Voltage Frequency Scaling (DVFS) takes place on a VLSI thermal management, the load condition changes and the voltage supply level varies, as shown in Figure 3.23 and Figure 3.24.

Figure 3.23  Output codes versus time on a FPGA based VLSI thermal management system, when DFS takes place, the output code change may be mistaken for a temperature change.

Figure 3.24  Power supply level of the FPGA chip, measured at the same time as the experimental results shown in Figure 3.23.
To minimize the measurement errors caused by the sensitivity to power supply variations, it is necessary to detect this error without using a voltage sensor. The supply level change is assumed whenever \(|C(T_i) - C'(T_i)|\) is larger than a pre-determined maximum possible temperature change, \(\Delta T\) between consecutive sampling time, \(\Delta t\) [7]:

\[
\left|C(T_i) - C'(T_i)\right| > \Delta T \cdot GS
\]

\[
\Delta T = \frac{P_{CORE} \cdot \Delta t}{m \cdot C_p}
\]

where \(GS\) is temperature sensor gain (°C\(^{-1}\)); \(m\) is the mass of core that dissipates a power of \(P_{CORE}\); \(C_p\) is its heat capacity.

Then after the process variations are removed as in (3.31), the correction factor \(N_c\) is updated whenever a supply level change is detected, at an unknown temperature \(T_1\):

\[
N_c = \frac{C(T_i)}{D(T_1, P, V_{DD,M})}
\]

Therefore the updated calibrated outputs \(C(T_i)\) remain the same despite power supply level changes. In Chapter 4 the correction factor \(N_c\) is updated when a DTM technique (e.g. DFS) is about to take place.

### 3.4 Proposed Self-calibration Step III: Automatic Calibration against Accurate On-chip Reference

#### 3.4.1 Automatic self-calibration against accurate on-chip temperature reference

The process variation removal self-calibration method proposed in Section 3.2 eliminates the inconsistencies among temperature sensors’ outputs at the same temperature, but post process variation removal codes \(C(T)\) in (3.31) needs further calibration to generate the true temperature. For example, in [21] and [22], the post process variation removal codes are further processed by software based second or third order master curve fittings. This section further enhances the process variation removal feature proposed in Section 3.2 and Section 3.3 with the additional capability of automatic self-calibration against an accurate on-chip bandgap based temperature
sensor reference. This section proposes a bandgap based temperature sensor, digitalized by a SAR ADC, as the accurate on-chip temperature reference. The accurate on-chip temperature sensor is trimmed to be accurate with an external reference in the first place. Then the multiple delay-line based temperatures proposed in Section 2.2.2 are calibrated automatically against the on-chip bandgap based temperature sensor reference, after the process variation and voltage supply level sensitivity are removed (as described in Section 3.2 and 3.3). This calibration procedure can be accomplished by calibrating only one delay-line based temperature sensor and applying the same offset $OFFSET$ and gain $GS$ to the remaining sensors, as after the process variation removal all delay-line based temperature sensors share the same gain and offset. At start-up and after the process variation removal, the delay-line based temperature sensor’s offset is computed using the accurate on-chip bandgap based temperature sensor reference output $R(T_c)$:

$$OFFSET = C(T_c) + \frac{R(T_c) \cdot GS}{GR}$$  \hspace{1cm} (3.41)

where $GR$ and $GS$ are the gain ($^\circ C^{-1}$) of the accurate on-chip bandgap based temperature sensor reference and delay-line based temperature sensor respectively. Along with the sensor gain, the true temperature in centigrade can be solved. The gain could be updated using the process variations removal codes $C(T_1)$ and $C(T_c)$:

$$GS' = GR \cdot \frac{C(T_1) - C(T_c)}{R(T_1) - R(T_c)}$$  \hspace{1cm} (3.42)

The calculation in (3.42) is performed whenever:

$$|C(T_1) - C(T_c)| > \Delta T_{\text{min}} \cdot GS$$  \hspace{1cm} (3.43)

where $\Delta T_{\text{min}}$ is the minimum temperature change to trigger the calculation in (3.42). The values of $GS'$ and $OFFSET$ are applied to all sensors.
Figure 3.25  ModelSim simulation results for the calibration to true temperature after the removal of process variations. The un-calibrated outputs are from the measurements of four temperature sensors on a Cyclone III FPGA shown in Figure 3.22.
The self-calibration method described in (3.41), (3.42) and (3.43) is verified using Verilog codes in ModelSim software. Figure 3.25 shows the simulation results of calibrating the post process variation removal outputs (in Figure 3.22) to the true temperature. The temperature change of the accurate reference temperature sensor is converted into the output code changes over time in the ModelSim test bench source file. A screenshot of the ModelSim software can be found in the Appendix B.

3.4.2 Accurate on-chip bandgap based temperature sensor reference

Figure 3.26 shows the block diagram of the on-chip bandgap based temperature sensor reference. The temperature reference is designed using TSMC’s 65nm 1V technology in the Cadence Environment. It contains four main blocks: the PTAT block that provides differential PTAT voltage, the VREF block that generates temperature independent reference of the SAR (successive approximation) ADC, the SAR timing block that provides switching timing reference of the SAR ADC, and the SAR ADC.

![Block diagram of the on-chip bandgap based temperature sensor reference.](image-url)
Figure 3.27  Schematic of the PTAT in the temperature reference.

The schematic of the PTAT reference is shown in Figure 3.27, in reference to [34] and [35]. $Vn$ and $Vm$ are made approximately equal by being connected to the differential inputs of an op amp, whose output is connected to $Vout$. M0, M1, Q0, Q1, R0, R3 comprise the bias circuit. Suppose the current in R0 is represented by $I_0$, $Vm = Vn$, $R0 = m \cdot R3$, $(W/L)_0 = m \cdot (W/L)_1$ is:

$$V_{BE,Q0} + I_0 R0 = V_{BE,Q1} + mI_0 \frac{1}{\beta_{F,Q1} + 1} \frac{R0}{m}$$  \hspace{1cm} (3.44)$$

Then $I_0$ is solved as:

$$I_0 = \frac{1 + \beta_{F,Q1}}{\beta_{F,Q1}} \frac{V_{BE,Q1} - V_{BE,Q0}}{R0}$$  \hspace{1cm} (3.45)$$
\[ V_{BE,Q1} - V_{BE,Q0} = V_T \ln(m) \] (3.46)

As \((W/L)_0 = (W/L)_3\), then the \(V_{BE2}\) can be expressed as:

\[ V_{BE2} = V_T \cdot \ln \left( \frac{I_0}{I_s} \frac{\beta_{F,Q1}}{1 + \beta_{F,Q1}} \right) \] (3.47)

Substituting (3.46) into (3.47), then

\[ V_{BE2} = V_T \cdot \ln \left( \frac{1 + \beta_{F,Q1} \frac{V_T \ln(m)}{R_0}}{I_s \beta_{F,Q1}} \frac{I_s \beta_{F,Q1}}{1 + \beta_{F,Q1}} \right) \] (3.48)

\[ = V_T \ln \left( \frac{V_T \ln(m)}{I_s R_0} \right) \]

\[ V_T = \frac{kT}{q} \] (3.49)

Therefore \(V_{BE2}\) is a PTAT voltage independent of any PNP gain \(\beta\). The differential voltage \(V_{BE1} - V_{BE2}\) is used as the PTAT voltage at the SAR ADC’s differential inputs.

Figure 3.28 shows the schematic of the two-stage op amp I2 used in Figure 3.27. The op amp’s frequency response is compensated by the pole splitting miller capacitance \(C_0\). The gain and the unity gain bandwidth of the op amp are respectively:

\[ \frac{\partial V_o}{\partial V_i} = g_{m2} \left( r_{o1} \parallel r_{o2} \right) \cdot g_{m6} \left( r_{o5} \parallel r_{o6} \right) \] (3.50)

\[ GBW = \frac{g_{m3}}{2\pi C_0} \] (3.51)

In order to have higher gain, larger width size in M2, M3 and M6, and longer length size in M1 and M0 have to be used as \(r_o\) is inversely proportional to \(\lambda\) that is inversely proportional to transistor length.
The op amp circuit shown in Figure 3.28 needs a bias voltage, which is generated by the circuit shown in Figure 3.29. The bias current in Figure 3.29 is:

\[ I_{\text{bias}} = \frac{V_{\text{GS4}}}{R0} \]  

(3.52)

Then \( I_{\text{bias}} \) is mirrored to M0, M6 and M7 in Figure 3.29. M7 provides the \( V_{\text{bias}} \) shown in Figure 3.28. As discussed in [58], the bias circuit in Figure 3.29 has two equilibrium points: (1) when its output is \( V_{\text{bias}} \) and (2) when its output is at the origin zero. In order to make the bias circuit work at the desired point to provide \( V_{\text{bias}} \), a startup circuit is needed. The startup circuit is shown in Figure 3.29, which is comprised of R1, M2 and M3. In Figure 3.29 when the circuit powers up, net18 rises due to current in R1 and M2. The same current in M2 flows into M4 and causes net17 to rise, while the current in M2 decreases. After the bias circuit enters the desired operating point that provides \( V_{\text{bias}} \), the startup circuit is not in use anymore.
The schematic of bandgap reference for the SAR ADC is shown in Figure 3.30. Conventional bandgap reference architectures, such as that proposed in [32], are not suitable to be used in conditions those power supply levels are less than 1.21 V. The technology used for this work is a 65nm 1.0 V CMOS technology. In Figure 3.30 M0, M1, Q0, Q1, R0 and feedback op amp I11 constitute a typical PTAT current reference. The current in R0 is:

\[
I_{PTAT} = \frac{V_z \ln(m)}{R0}
\]  (3.53)

where \( m = \frac{A_{Q1}}{A_{Q2}} \), where A is the emitter area size. \( V_z (V_{BE4}) \) has a negative temperature coefficient:

\[
V_z = V_T \cdot \ln\left(\frac{I_z}{I_S}\right)
\]  (3.54)
Figure 3.30  The bandgap voltage reference for the SAR ADC.

$V_1$ is made to be approximately the same with $V_2$ by the feedback op amp I14. The current in R7 has a negative temperature coefficient:

$$I_{CTAT} = \frac{V_2}{R7} = \frac{V_T}{R7} \cdot \ln \left( \frac{I_{M7}}{I_S} \right)$$  \hspace{1cm} (3.55)

The summing current in R4, R5 and R6 is:

$$I_{sum} = I_{CTAT} \left( \frac{W}{L} \right)_3 + I_{PTAT} \left( \frac{W}{L} \right)_4 \left( \frac{W}{L} \right)_0$$

$$= \frac{V_T}{R7} \cdot \ln \left( \frac{I_{M7}}{I_S} \right) \left( \frac{W}{L} \right)_3 + \frac{V_T}{R0} \ln \left( \frac{m}{R0} \right) \left( \frac{W}{L} \right)_4 \left( \frac{W}{L} \right)_0$$  \hspace{1cm} (3.56)

where
\[ I_s = bT^{4+m} \exp\left(\frac{E_g}{kT}\right) \]  

(3.57)

Therefore by proper sizing \((W/L)_3/(W/L)_2\) and \((W/L)_4/(W/L)_0\), temperature independent current \(I_{sum}\) can be obtained. The voltage reference \(V_{CM}\) and \(V_{TOP}\) has the same shape as \(I_{sum}\) and is temperature independent.

**Figure 3.31** The timing circuit for the SAR ADC.

**Figure 3.32** Schematic of Driver1 used in Figure 3.31.
The timing circuit for the SAR ADC is as shown in Figure 3.31. It provides switching control signals S1, S2, S3 and reset for the SAR ADC. The differential SAR ADC circuit diagram is as shown in Figure 3.33, in reference to [59].

![Circuit Diagram](image)

**Figure 3.33** The circuit diagram of the SAR ADC in sampling phase [59].

In the differential SAR ADC, for one conversion (reset) cycle, initially (in the sampling phase) the bottom plates of the all capacitors are connected to $V_{ip}$ ($V_{in}$) and the top plates are all connected to $V_{cm}$, as shown in Figure 3.33. Therefore the charges stored in the differential capacitor arrays are respectively:

\[
Q_{total,p} = 2^N C_{10} (V_{cm} - V_{ip})
\]
\[
Q_{total,n} = 2^N C_{10} (V_{cm} - V_{in})
\]

(3.58)

Then in the first step of the bit conversion phase, for the $V_{ip}$ side, the bottom plate of C1 is connected to $V_{ref}$ and those of the remaining capacitors are all connected to $V_{cm}$, as shown in Figure 3.34. While at the $V_{in}$ side, the bottom plate of C1 is connected to $V_{cm}$ and those of the remaining capacitors are all connected to $V_{ref}$. For charge balancing on both capacitor arrays, the voltages at the two inputs of the comparator ($V_+$ and $V_-$) are respectively:
\[ V_+ = \frac{3}{2} V_{cm} - V_{ip} + \frac{V_{ref}}{2} \]
\[ V_- = \frac{3}{2} V_{cm} - V_{in} + \frac{V_{ref}}{2} \]

(3.59)

Figure 3.34  The circuit diagram of the SAR ADC in bit conversion phase (I) [59].

If \( V_{ip} > V_{in} \), then the MSB (B[9]) should be 1. In this case, the bottom plate of C1 at the \( V_{ip} \) side will still be connected to \( V_{ref} \) and that of the C1 at the other side will still be connected to \( V_{cm} \). Otherwise if \( V_{ip} < V_{in} \) the MSB should be 0. In this case, the bottom plate of C1 at the \( V_{ip} \) side will be switched back to \( V_{cm} \) and that of the C1 at the other side will still be switched back to \( V_{ref} \).

Figure 3.35  The circuit diagram of the SAR ADC in bit conversion phase (II) \( (V_{ip} > V_{in} \) in the first step) [59].
In the second step, the bottom plate of C2 is switched to $V_{ref}$ at the $V_{ip}$ side and that of C2 is switched to $V_{cm}$ at the $V_{in}$ side, as shown in Figure 3.40 (if $V_{ip}>V_{in}$ in the first step). Then the voltages at the two inputs of the comparator are respectively:

$$
V_+ = \frac{5}{4}V_{cm} - \frac{3}{4}V_{ip} + \frac{1}{4}V_{ref}
$$

$$
V_- = \frac{7}{4}V_{cm} - \frac{1}{4}V_{in} + \frac{1}{4}V_{ref}
$$

(3.60)

According to (3.60), if $V_{ip}>V_{in}+(1/2)(V_{ref}-V_{cm})$, then the B[8] should be 1 and the bottom plate of C2 at the $V_{ip}$ side will still be connected to $V_{ref}$ and that of C2 at the other side will still be connected to $V_{cm}$. Otherwise if $V_{ip}<V_{in}+(1/2)(V_{ref}-V_{cm})$ the B[8] should be 0 and the bottom plate of C2 at the $V_{ip}$ side will be switched back to $V_{cm}$ and that of C2 at the other side will still be switched back to $V_{ref}$. Both situations are as illustrated in Figure 3.36. Comparison procedures of the remaining bits are analogous to the above description of the second step.

The SAR logic is shown in Figure 3.37, in reference to [60]. The logic outputs $L[9:0]$ turn on sequentially to conduct the SAR capacitor array’s operations as described from Figure 3.33 to Figure 3.36. For the $DFCSND1$ in the standard cell library, when $SDN$ (set) is low, its output $Q$ is high despite $D$’s state; when $CND$ (reset) is low, its output is low despite $D$’s state. When the input $RST$ is low: $Q[10]$ is high; $Q[9:0]$ is low; $L[9:0]$ is low and $L_\text{dummy}$ is high. When $RST$ rises to high, the high logic state of $Q[10]$ is transferred from $Q[9]$ to $Q[0]$ sequentially by I16. Then the low state of the complementary $QN[10:0]$ is transferred to I17 and generates logic high $L[10:0]$ sequentially. Finally the state of $L[9:0]$ is determined by the comparator outputs at I17’s input $D$. For example, the final state of $L[10]$ is determined by the comparison results at the comparator output when $L[9]$ rises from low to high (positive edge).
Figure 3.36  SAR ADC in bit conversion phase (II) ($V_{ip}$>$V_{in}$ in the first step). (a) $V_{ip}$>$V_{in}$+(1/2)($V_{ref}$-$V_{cm}$). (b) $V_{ip}$<$V_{in}$+(1/2)($V_{ref}$-$V_{cm}$) [59].
The SAR ADC shown from Figure 3.33 to Figure 3.36 needs a comparator. As the SAR ADC is used for the on-chip bandgap based temperature sensor reference, the comparator’s speed requirement is low while its accuracy should be as fine as possible. One challenge lies in the gate current leakage at the differential inputs of the comparator. For example, for a NMOS with size $W/L = 2 \mu m / 0.5 \mu m$, its gate leakage current is $115 \, nA$. As its gate is directly connected to the DAC capacitor array shown in Figure 3.33 to Figure 3.36, the leakage current causes the comparator input voltage to change gradually during each clock period by charging the capacitor array by (if the clock period is $1 \, \mu s$):

$$ I_{\text{leakage}} / \text{Carray} \times \text{clock \_ period} = 115 \, nA / 64 \, \text{pF} \times 1 \, \mu s \approx 2 \, mV $$

As a comparison, the differential PTAT voltage at the SAR input is $120 \, mV$, and the ADC resolution is designed to be $200 \, mV/512 = 0.4 \, mV$. In order to get rid of the gate leakage current’s effect on the ADC, a rail-to-rail pre-amplifier is used so that the leakage currents at both the gates of the NMOS and PMOS input differential pairs compensates each other with approximately the same value but opposite direction.
Figure 3.38  The rail-to-rail comparator used in the SAR ADC in Figure 3.33 [61].

The rail-to-rail comparator is shown in Figure 3.38. The preamplifier on the left is in reference to [61]. As discussed in [62], the comparator shown in Figure 3.38 has one stage of preamplification (left) followed by a track-and-latch stage (right). The pre-amplification stage is used to obtain higher resolution and to minimize the effects of kickback. The preamplifier typically has gain less than 10 (sometimes it is a unity-gain buffer), and the higher its gain the slower the speed. Another usage of the preamplifier is to eliminate the kickback. The kickback is the charge transferred either into or out of the inputs when the track-and-latch stage goes from track mode to latch mode. This charge transfer is caused by the charge needed to turn on the transistors in the positive-feedback circuitry and is also caused by the charge that must be removed to turn off the transistors in the tracking circuitry. The preamplifier works as follows. Taking the case when the inputs INP>INN for example, in the NMOS input pair more current flows in the diode connected load M8 than that does in M9 and therefore the voltage drop of M8 is larger than that of M9 and the voltage OUTP is larger than the voltage OUTN. For the PMOS input pair, more current flows in the M5, and this causes the voltage of net143 to be larger than
that of net161. Therefore more current is in M7 than that in M6. This adds up to even larger current in M8 than M9 and more voltage drop on the former. The voltage difference in OUTP-OUTN is further amplified by the track-and-latch stage in a positive feedback, and this feedback regenerate analog signal into a digital signal. The track-and-latch stage works as follows. In the reset state, VITCH is at logic high, M23 and M24 are off and the voltages COMPPP and COMN are reset to the logic low (ground) level. In the comparison state, VITCH drops to logic low, and M23 and M24 are on. Continuing with the above deduction earlier in the paragraph that OUTP is larger than OUTN, the gate voltage of M10 is larger than that of M11, which makes the former a larger Ron initially and the voltage of net181 lower than that of net76 eventually. As M23 and M24 are on, the voltage of COMPPP is lower than that of COMN. The positive feedback formed by M18 and M16 further causes M16’s drain current to go down (due to its lower gate voltage) and M16’s drain voltage (gate voltage of M18) to go up. Finally the COMN voltage goes up to digital high, being followed and converted to fully logic high by a driver made of two inverters M36–M39.

3.4.3 Layout and simulation results

The layout of the accurate on-chip bandgap based temperature sensor (SAR capacitor array excluded) using TSMC’s 65nm 1V technology in Cadence Virtuoso is shown in Figure 3.39. A layout including the SAR capacitor array is shown in the Appendix C. The area occupied by each block is labeled.
Figure 3.39 Layout of the accurate on-chip bandgap based temperature sensor using TSMC’s 65nm CMOS technology in Cadence Virtuoso (Capacitor array is not shown).

The schematic and the post-layout simulations of all the sub-blocks in the accurate on-chip bandgap based temperature sensor designed in Section 3.4.2 are shown from Figure 3.40 to Figure 3.47, using TSMC’s 65nm technology in the Cadence Environment. Figure 3.40 shows the simulation results of the bias and startup circuit illustrated in Figure 3.29. The transistor sizes are given in Table 3.1. It can be seen that the bias voltage $V_{\text{bias}}$ rises to the proper voltage level when the circuit powers up (when $V_{DD}$ begins to rise from 0 to 1V at 0 second).

Table 3.1 Transistor Size of the Bias and Startup Circuit Shown in Figure 3.29

<table>
<thead>
<tr>
<th>MOS Transistors</th>
<th>M0, M1, M5, M6</th>
<th>M4, M7</th>
<th>M2, M3</th>
</tr>
</thead>
<tbody>
<tr>
<td>W/L (µm / µm)</td>
<td>2 / 0.5</td>
<td>2 / 0.5</td>
<td>2 / 0.5</td>
</tr>
<tr>
<td>Number of Fingers</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Resistor</td>
<td>R1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W/L (µm / µm)</td>
<td>0.4 / 28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value (Ohm)</td>
<td>1k</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 3.40  Schematic simulation of startup circuit shown in Figure 3.29.

Figure 3.41 shows the post-layout simulation results of the op amp shown in Figure 3.28. Its gain, UBW and phase margin are simulated to be 52 dB, 300 MHz and 45 °, respectively. Its component sizes are given in Table 3.2.

Table 3.2 Component Sizes for the Circuit Shown in Figure 3.28

<table>
<thead>
<tr>
<th>MOS Transistors</th>
<th>M0, M1</th>
<th>M2, M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
</tr>
</thead>
<tbody>
<tr>
<td>W/L (µm / µm)</td>
<td>2 / 0.5</td>
<td>2 / 0.5</td>
<td>2 / 0.5</td>
<td>2 / 0.5</td>
<td>2 / 0.5</td>
</tr>
<tr>
<td>Number of Fingers</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>6</td>
<td>80</td>
</tr>
<tr>
<td>Miller Capacitor</td>
<td>C0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W/L (µm / µm)</td>
<td></td>
<td></td>
<td>20 / 30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value (pF)</td>
<td>1.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 3.41  Postlayout simulation results of the op amp shown in Figure 3.28, with gain 52 dB, UBW 300MHz, phase margin 45°.

Figure 3.42 shows the post-layout simulation results of the PTAT voltage reference illustrated in Figure 3.27. It could be seen that the resolution of the input differential voltage of the SAR ADC (VBE1-VBE2) is 0.44 mV/°C. In Figure 3.27, the ratio of the resistances of R0 and R3 is not as calculated in (3.44) to be m (m = 5), due to parasitic resistance at the PNPs’ base terminals.
Table 3.3 Component Sizes for the Circuit Shown in Figure 3.27

<table>
<thead>
<tr>
<th>Component</th>
<th>M0</th>
<th>M1</th>
<th>M3</th>
<th>M4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOS Transistors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W/L (µm / µm)</td>
<td>2 / 1</td>
<td>2 / 1</td>
<td>2 / 1</td>
<td>2 / 1</td>
</tr>
<tr>
<td>Number of Fingers</td>
<td>100</td>
<td>20</td>
<td>20</td>
<td>200</td>
</tr>
<tr>
<td>PNPQs</td>
<td>Q0</td>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
</tr>
<tr>
<td>W/L (µm / µm)</td>
<td>10 / 10</td>
<td>10 / 10</td>
<td>10 / 10</td>
<td>10 / 10</td>
</tr>
<tr>
<td>Multiplier/ value</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>Resistor</td>
<td>R0</td>
<td>R3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W/L (µm / µm)</td>
<td>300 / 0.4</td>
<td>20 / 0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value (Ohm)</td>
<td>10 k</td>
<td>713</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.42 Post-layout simulation results of the PTAT voltage architecture in Figure 3.27.
Figure 3.43 shows the post-layout simulation of the reference voltage designed in Figure 3.30. The reference voltage’s transistor sizes are as listed in Table 3.4. Its simulated PSRR is shown in Figure 3.44.

Table 3.4 Transistor Sizes for the Circuit Shown in Figure 3.30

<table>
<thead>
<tr>
<th>MOS Transistors</th>
<th>M0, M3, M6</th>
<th>M7</th>
<th>M4</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>W/L (µm / µm)</td>
<td>2 / 1</td>
<td>2 / 1</td>
<td>2 / 1</td>
<td>2 / 1</td>
</tr>
<tr>
<td>Number of Fingers</td>
<td>10</td>
<td>8</td>
<td>29</td>
<td>2</td>
</tr>
<tr>
<td>PNPs</td>
<td>Q0</td>
<td>Q1</td>
<td>Q4</td>
<td></td>
</tr>
<tr>
<td>W/L</td>
<td>10 / 10</td>
<td>10 / 10</td>
<td>10 / 10</td>
<td></td>
</tr>
<tr>
<td>Multiplier</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Resistors</td>
<td>R0</td>
<td>R4</td>
<td>R5</td>
<td>R6</td>
</tr>
<tr>
<td>W/L (µm / µm)</td>
<td>0.4 / 27</td>
<td>0.4 / 4</td>
<td>0.4 / 16</td>
<td>0.4 / 40</td>
</tr>
<tr>
<td>Values (Ohm)</td>
<td>1k</td>
<td>150</td>
<td>598</td>
<td>150</td>
</tr>
</tbody>
</table>

Figure 3.43  Postlayout simulations of VCM and VTOP in Figure 3.30.
The temperature coefficient of the bandgap reference simulated in Figure 3.43 is calculated as (in ppm):

$$TC = \frac{V_{max} - V_{min}}{V_{nominal}(T_{max} - T_{min})} \cdot 10^6 \text{ ppm}$$

$$TC_{Vtop} = \frac{694.15 - 692.35}{693(100)} \cdot 10^6 = 25.9 \text{ ppm}$$  (3.62)

$$TC_{Vcm} = \frac{495.8 - 494.6}{495(100)} \cdot 10^6 = 24.2 \text{ ppm}$$

![AC Response](image)

**Figure 3.44** 1/PSRR of the Bandgap reference in Figure 3.30.

Figure 3.45 shows the schematic simulation results of the timing circuit shown in Figure 3.31. In Figure 3.33 S1 controls the transmission gate that switches between $V_{ip}$ ($V_{in}$) and $V_{ref}$. S2 controls $V_{cm}$. S3 controls complementary SAR logic.
Figure 3.45  Schematic simulation results of the SAR timing circuit in Figure 3.31.

Figure 3.46 shows the schematic simulation results of the SAR logic shown in Figure 3.37. The comparator output is assumed to be logic low for the simulation. It could be seen that L[9] to L[0] turns on sequentially. As the comparator output is set to be logic low in the test bench, the final states of L[9:0] are all logic low.

Figure 3.47 shows the post-layout simulation results of the comparator shown in Figure 3.38. The comparator layout must be ensured to be symmetrical. One input of the comparator INN is swept to be from 0 to 1V and the other is kept to be constant 0.5V.
Monte Carlo mismatch simulations are performed using different cap sizes, in order to find the minimum capacitor size that could enable the SAR with a minimum resolution of 9 bit. The simulation starts with the minimum capacitor size (2 μm × 2 μm (19 fF)). Table 3.5 shows the
parameters used for calculating mismatches for a capacitor with size 10 \( \mu m \times 10\mu m \), when mismatches is calculated as:

\[
Factmis = mis \cdot geo\_fact = 0.0061 \times 0.1 = 0.0061 = 0.061% 
\] (3.63)

In order to have enough accuracy for a 9 bit ADC when both INL and DNL is smaller than 0.5 LSB, the minimum capacitor area for the least significant bit should be:

\[
Area_{min} = Area_{10\,\mu m \times 10\mu m} \cdot Factmis \cdot 2^{Number\_of\_bits} 
\]
\[
= 10\mu m \times 10\mu m \times 0.0061 \times 2^9 
\]
\[
= 31\mu m^2 
\] (3.64)

To achieve the minimum area shown in (3.64), the capacitor width and length are \( W = 5 \mu m \), \( L = 6 \mu m \) and its capacitance is 64 fF. Generally speaking, if the number of bit requirement increases by 1, both \( W \) and \( L \) should increase to be 1.4 times of their original values. Meantime the unit capacitor area increases to be twice of the original and the total area is four times as large. Monte Carlo mismatch simulation results on different sizes of the minimum capacitor size used in the SAR ADC are shown in Figure 3.48 to Figure 3.50.

**Table 3.5 Parameters for Calculating Mismatches for a 10 \( \mu m \times 10\mu m \) Capacitor**

<table>
<thead>
<tr>
<th>mis (( \mu m ))</th>
<th>mismatchflag</th>
<th>geo_fact</th>
<th>Capacitor W/L (( \mu m / \mu m ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0061</td>
<td>3</td>
<td>0.1</td>
<td>10 / 10</td>
</tr>
</tbody>
</table>

Figure 3.51 shows schematic simulations of the SAR ADC. The SAR ADC’s reference voltages are at the same as voltage levels as \( V_{CM} \) and \( V_{REF} \) shown in Figure 3.43. The SAR ADC’s differential input voltage levels are the same as those in PTAT voltage shown in Figure 3.42. The \( IN_{IN} \), \( IN_{P} \) are connected to \( V_{in} \) and \( V_{ip} \) as shown in Figure 3.33. \( V_{in\_plus} \) and \( V_{in\_minus} \) refer to the positive and negative inputs of the comparator. As shown in Figure 3.51 (b), \( V_{in\_plus} \) and \( V_{in\_minus} \) gradually come close to each other during the SAR comparison process.
Figure 3.48  Monte Carlo mismatch simulation results on different sizes of the minimum capacitor sizes used in the SAR AD when minimum cap = 19 fF, Mu/sigma = 350.

Figure 3.49  Monte Carlo mismatch simulation results on different sizes of the minimum capacitor sizes used in the SAR ADC when minimum capacitor = 76 fF. Mu/sigma = 960.
Figure 3.50  Monte Carlo mismatch simulation results on different sizes of the minimum capacitor sizes used in the SAR ADC when minimum capacitor = 200 fF, \( \text{Mu/sigma} = 1738.3 \).
Figure 3.51  Schematic simulations of SAR ADC. (a) using VCM, Vref voltage levels from Figure 3.43 and differential input voltage levels as those in PTAT voltage in Figure 3.42. (b) zoom in of (a) at around 350 μs when INN = 779.79 mV and INP = 649.96 mV.
The SAR ADC’s static and dynamic characteristics are simulated in Cadence using the setup shown in Table 2.3 (in TSMC’s 65nm technology), and test parameters are listed in Table 3.6. The simulated INL and DNL results are shown in Figure 3.52. The simulated dynamic linearity is shown in Figure 3.53.

Table 3.6 Test Parameters for SAR ADC’s Static and Dynamic Characteristics

<table>
<thead>
<tr>
<th>Sampling clock frequency (Hz)</th>
<th>Input clock frequency (Hz)</th>
<th>Record length</th>
</tr>
</thead>
<tbody>
<tr>
<td>52.63 k (period:19 μs)</td>
<td>25.7 (38.912 ms =19 μs × 2048)</td>
<td>2048</td>
</tr>
</tbody>
</table>
Figure 3.52  Simulated static linearity of the SAR ADC.
Figure 3.53  Simulated dynamic linearity of the SAR ADC.
A summary of the SAR ADC’s specifications is listed in Table 3.7.

**Table 3.7 Simulated Specifications of the SAR ADC**

<table>
<thead>
<tr>
<th></th>
<th>Target</th>
<th>Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of bits</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Sampling rate</td>
<td>1 kHz</td>
<td>53.63 kHz</td>
</tr>
<tr>
<td>Input voltage range</td>
<td>0 to 200 mV</td>
<td>2 mV to 198 mV</td>
</tr>
<tr>
<td>Power supply level</td>
<td>1 V</td>
<td>1 V</td>
</tr>
<tr>
<td>Current consumption</td>
<td>200 µA</td>
<td>104 µA</td>
</tr>
<tr>
<td>resolution</td>
<td>0.4 mV</td>
<td>0.383 mV</td>
</tr>
<tr>
<td>DNL</td>
<td>±0.5 LSB</td>
<td>±0.5 LSB</td>
</tr>
<tr>
<td>INL</td>
<td>±0.5 LSB</td>
<td>±0.4 LSB</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.4 mV</td>
<td>0.1 mV</td>
</tr>
<tr>
<td>SINAD</td>
<td>56.84</td>
<td>54.98</td>
</tr>
<tr>
<td>ENOB</td>
<td>9</td>
<td>8.9</td>
</tr>
</tbody>
</table>

Figure 3.54 shows the final system level simulation results of the accurate on-chip bandgap based temperature sensor reference. It shows both simulation results with I/O and post-layout simulation results at three temperatures: 20 °C, 50 °C and 80 °C.
Figure 3.54 Simulation results of the accurate temperature reference using schematic shown in Figure 3.26. The schematic simulation includes I/Os. The post-layout simulation is performed with extracted RC using Calibre tools in the Cadence Environment.

3.5 Experimental Results

The self-calibration method block was initially implemented on Cyclone III and Cyclone IV FPGAs, respectively. It contains blocks on (I) process variation removal, (II) supply level variation removal and (III) automatic calibration against accurate on-chip reference. The FPGA based self-calibration block was used to calibrate delay-line based temperature sensors (as proposed in Chapter II) implemented on the same FPGA or on a custom IC. The self-calibration method block was also synthesized using IBM 0.13 μm CMOS technology’s ARM cell, to calibrate the delay-line based temperature sensors implemented on the same chip.

A micrograph of the prototype custom IC implemented using TSMC’s 65nm 1V CMOS technology is shown in Figure 3.55. A block diagram of the prototype could be found in Figure 3.20. The accurate bandgap based temperature reference is implemented in the middle of the chip. The four delay-line based temperature sensors are located at each of the four corners to exhibit on-chip process variations. One of the delay-line based temperature sensors is located
next to the accurate on-chip bandgap based temperature sensor reference for automatic gain and offset calibration as described in Section 3.4.1.

![Diagram]

**Figure 3.55** Micrograph of the prototype custom IC implemented using TSMC’s 1V 65nm CMOS technology.

Figure 3.56 shows the micrograph of the custom IC implemented using IBM’s 0.13 μm 1.2V CMOS technology.

Figure 3.57 and Figure 3.58 show floor-plan of the self-calibration block and the four delay-line based temperature sensors on Cyclone III and Cyclone IV FPGA chips, respectively.
Figure 3.56  Micrograph of the custom IC implemented using IBM’s 0.13 μm CMOS technology.

Figure 3.57  Floor-plan of the self-calibration algorithm block implemented on a Cyclone III FPGA. The accurate temperature reference is off-chip. The four temperature sensors are of different architectures (sizes), as listed in Table 2.4.
Same as in Section 2.5, all the temperature sensors are tested at 5 °C increment from 20°C to 80°C in an ESPEC™ ETC-3 temperature chamber. A commercial temperature sensor in [50] is used as the external (off-chip) accurate temperature reference.

3.5.1 Removal of sensitivity to process variations

The process variation removal experiments are performed using the method proposed in Section 3.2 on delay-line based temperature sensors implemented on various platforms: 3 Cyclone III and a Cyclone IV FPGAs, 3 custom ICs fabricated using TSMC’s 65nm CMOS technology and 3 custom ICs fabricated using IBM’s 0.13 μm CMOS technology. The self-calibration method that removes process variations is implemented using Verilog code on Cyclone III or Cyclone IV FPGA to calibrate temperature sensors implemented on the same FPGA and to calibrate those fabricated on the TSMC’s 65nm CMOS ICs. The self-calibration method is also synthesized
using ARM cells using IBM’s 0.13 μm 1.2 V CMOS technology when calibrating temperature sensors implemented on the same chip.

Figure 3.59 and Figure 3.60 show the experimental results before and after the process variations are removed in 12 sensors on 3 different Cyclone III EP3C25F324 FPGA chips. It can be seen that both “on-chip” and “die to die” process variations are removed by the proposed self-calibration method on the three Cyclone III chips. Chip I and II have batch number 1108 and Chip III has batch number 0810.

Figure 3.61 and Figure 3.62 show the experimental results before and after the process variations removal on a Cyclone IV EP4CE115F29C8N FPGA chip.

![Figure 3.59](image)

**Figure 3.59** Un-calibrated output codes for 4 sensors on each of the three Cyclone III FPGA chips, using the self-calibration method proposed in Section 3.2. Chip I and II have batch number 1108 and Chip III has batch number 0810.
Die-to-die and on-chip process variations are removed for 4 sensors on each of the three Cyclone III FPGA chips, using the self-calibration method proposed in Section 3.2. Chip I and II have batch number 1108 and Chip III has batch number 0810.
Figure 3.61  Un-calibrated outputs for 4 sensors on a Cyclone IV FPGA chip, using the self-calibration method proposed in Section 3.2.
On-chip process variations are removed on a Cyclone IV FPGA chip, using the self-calibration method proposed in Section 3.2.
Figure 3.66 and Figure 3.64 shows the experimental results before and after the process variations removal of 12 sensors on 3 custom IC chips fabricated using TSMC’s 65nm 1V CMOS technology.

![Graph showing un-calibrated outputs on 3 custom ICs](image)

**Figure 3.63** Un-calibrated outputs on 3 custom ICs fabricated using TSMC’s 65 nm CMOS technology, using the self-calibration method proposed in Section 3.2.
Figure 3.64  Process variations are removed on 3 custom ICs fabricated using TSMC’s 65 nm CMOS technology, using the self-calibration method proposed in Section 3.2.
Figure 3.65  Process variations are removed on 3 custom ICs fabricated using IBM’s 0.13 μm CMOS technology, using the self-calibration method proposed in Section 3.2.
Figure 3.65 shows the experimental results before and after the process variations removal of 12 sensors on 3 custom IC chips fabricated using IBM’s 0.13 μm 1.2 V CMOS technology. In this case, the self-calibration method is also synthesized using ARM cells when calibrating temperature sensors implemented on the same chip.

Figure 3.60 to Figure 3.65 show that using the self-calibration method proposed in Section 3.2, the process variations are removed for delay-line based temperature sensors implemented on 3 Cyclone III, a Cyclone IV FPGA and on 3 custom ICs fabricated using 65nm and 0.13 μm technology respectively. The remaining code deviations among calibrated outputs shown in Figure 3.60 (b) and (c) (or Figure 3.64 (b) and (c)) are mainly caused by the following three sources. The first is the thermal gradient between the temperature sensor under test and the external reference temperature sensor. In particular, the thermal gradient between the external reference [50] and the delay-line based temperature sensors can be different among the three chips’ measurements, due to the differences in their physical locations. This assumption is supported by the fact that as shown in Figure 3.60 (b) and (c), there are larger die-to-die than on-chip inconsistency. The second comes from the self-calibration procedure (as $N_C$ must be an integer). The resulted error is less than the sensor resolution (e.g. 0.36 °C for Cyclone III). The third is the gain (as $H(T)$ in Section 3.2.1) differences among different chips (as shown in Figure 3.19 the gain $H(T)$ is a function of both process and temperature).

It could be observed from Figure 3.64 (a) that on the 65nm custom IC there are more on-chip process variations due to different physical location, than chip-to-chip process variations at the same location. As the un-calibrated outputs of delay-line temperature sensors at the same location but on different chips are closer than those of the sensors on the same chip but on different locations (corners). A possible explanation is that the effects from doping variations across the chip due to different neighboring blocks are larger than those from the geometrical variations.

It can also be seen that when the self-calibration is performed at different startup temperatures, the output codes differ at the same measurement temperature, as shown in Figure 3.60 (b) and (c). Solution of the above problem will be covered in Section 3.5.3.
3.5.2 Removal of sensitivity to supply voltage level variations

As the supply voltage level variations are best observed and measured on custom ICs, the experiments are performed on the TSMC’s 65nm 1V CMOS custom ICs.

Prior to remove the sensitivity to supply voltage level variations, the calibration is performed at 1 V and measured at 1 V, 1.05 V and 0.95 V, respectively, as shown in Figure 3.66 (a). All the measurements use the same setup and the same chips as those in Figure 3.64. For example, when all 12 sensors are calibrated at 1 V and measured at 1.05 V, the inconsistencies among themselves are no larger than those seen in Figure 3.64. However, there are inconsistencies between digital outputs measured at 1.05 V and 1 V respectively, when the calibrations are done at 1 V for both situations. This is because that the delay in the delay-line based temperature sensor is affected by voltage supply level. The higher the voltage, the faster the delay propagates, and the more the output codes are. As discussed in (3.37) and as shown in Figure 3.66 (a), if the temperature sensor is calibrated at 1 V, and its output codes are read at 0.95 V (this may happen in real application due to workload variations during operation), the voltage supply change may be mistaken for a temperature change. Without the voltage supply variation removal, the resulting error could be as much as 25 °C, or 0.6 °C/mV (the power supply sensitivity). This is estimated using the changes in the output codes at different supply voltages shown in Figure 3.66 (a) for the same 5 % change in the supply voltage.

The above sensitivity to power supply level variations is removed, as shown in Figure 3.66 (b), using the self-calibration method step II introduced in Section 3.3. It can be seen that there are gain differences between outputs measured at different voltages. This is because the temperature sensor gain is a function of power supply level, as indicated in (3.33) and (3.34). These gain differences, however, translate to less than 4 °C errors for a ±0.05 V temperature change (or, 0.06 °C/mV, approximately ten times smaller than the case without the proposed self-calibration shown in Figure 3.66 (a)). These errors will be shown in Section 3.5.3.
(a) Without removal of sensitivity to supply voltage variation, when 12 sensors on 3 chips are calibrated at 1 V and measured at 1 V, 0.95 V and 1.05 V respectively.

(b) With removal of sensitivity to supply voltage variation

Figure 3.66 Removal of sensitivity to supply voltage level variations for 12 sensors on 3 custom IC chips fabricated using TSMC’s 65nm CMOS technology.
3.5.3 Automatic calibration against accurate temperature reference

The procedures to prepare the custom IC for on-chip automatic self-calibration are as follows. First of all, the on-chip accurate bandgap temperature sensor designed in Section 3.4.2 and 3.4.3 is calibrated (trimmed) against an accurate external commercial temperature sensor [50]. After this step the external commercial sensor is no longer in use. Secondly, the post process variation and voltage supply variation removal output codes of delay-line based temperature sensors are further automatically calibrated against the on-chip accurate bandgap temperature sensor, using the automatic self-calibration method proposed in Section 3.4.1.

![Graph showing measurement results of the un-calibrated outputs of the accurate temperature sensor](image)

**Figure 3.67** Measurement results of the un-calibrated outputs of the accurate temperature sensor shown in Figure 3.26 in Section 3.4.2, on three custom IC chips fabricated using TSMC’s 65nm 1V technology.
(a) Calibrated outputs of accurate bandgap based temperature sensor

(b) Measurement errors of (a)

Figure 3.68  Measurement results of the accurate temperature sensor shown in Figure 3.26 in Section 3.4.2, on three custom IC chips fabricated using TSMC’s 65nm 1V technology.
The measurement results of the accurate temperature sensor described in Section 3.4.2 and Section 3.4.3 are shown in Figure 3.68. It can be seen that there are inconsistencies among the un-calibrated digital outputs of the accurate bandgap temperature sensor designed in Section 3.4.2, being affected by process variations. The un-calibrated digital outputs are then trimmed with an accurate external commercial temperature reference in [50]. After each temperature sensor is trimmed by a linear curve fitting (a different fitting for each sensor), its 3 sigma measurement errors are kept within ±2 °C, on three custom IC chips fabricated using TSMC’s 65nm 1V CMOS technology, as shown in Figure 3.68 (c).

The post process variation removal outputs of 12 delay-line based temperature sensors on 3 custom IC chips (as shown in Figure 3.64) are further automatically calibrated against the on-chip accurate bandgap based temperature sensor (measurement results as shown in Figure 3.68), using the automatic self-calibration method proposed in Section 3.4.1. The measurement results are as shown in Figure 3.69. The process variation removal (step I in Section 3.2) and automatic calibration against on-chip accurate temperature reference (step III in Section 3.4) are both automatic on-chip. That is, by powering on the chips, the digital outputs shown in Figure 3.69 (a) are generated from the FPGA based self-calibration blocks automatically. In Figure 3.69 (a), the digital outputs represent true temperatures. For example, at 20 °C, output codes 81 are equivalent to 81/4 = 20.25 °C. The same output code versus temperature scheme applies to the all the measurement results in this section.

The post process variation and supply voltage variation removal outputs of 12 sensors on 3 custom IC chips fabricated using TSMC’s 65nm CMOS 1V technology (as shown in Figure 3.66) are further automatically calibrated against the on-chip accurate bandgap temperature sensor, using the automatic self-calibration method proposed in Section 3.4.1. The measurement results are shown in Figure 3.69. The process variation removal (step I in Section 3.2), supply voltage variation removal (step II in Section 3.3) and calibration against on-chip accurate temperature reference (step III in Section 3.4) are all automatic on-chip.
(a) Calibrated outputs after being calibrated to true temperature

(b) Measurement errors of (a)

Figure 3.69  Measurement results of 12 delay-line based temperature sensors on 3 custom ICs, after self-calibration step I and III are performed (cool and warm startups). The measurement results are obtained without supply voltage variations.
(a) Calibrated outputs after process and supply voltage variations are removed and are calibrated to true temperature.

(b) Measurement errors of (a). The dotted line shows 3 sigma errors.

Figure 3.70 Measurement results of 12 delay-line based temperature sensors on 3 custom ICs fabricated using 65nm technology, after self-calibration step I, II and III are performed (all warm startups). The measurement results are obtained when the supply voltage variations are within ±5 %.
The post process variation removal outputs of 12 sensors on 3 custom IC chips using IBM’s 0.13 μm 1.2 V CMOS technology (as shown in Figure 3.65) are further automatically calibrated against the on-chip accurate bandgap temperature sensor, using the automatic self-calibration method proposed in Section 3.4.1. The self-calibration method is synthesized using ARM cell on the same chip using the same IBM’s 0.13 μm 1.2 V CMOS technology. The measurement results are shown in Figure 3.71.

The post process variation removal measurement results of 12 sensors on 3 Cyclone III FPGA chips shown in Figure 3.60 in Section 3.5.1 are further automatically calibrated against an accurate external temperature reference in [50], using the automatic self-calibration method proposed in Section 3.4.1. As the temperature sensors are measured in a temperature chamber, the external accurate temperature reference is there to mimic an on-chip bandgap based temperature sensor reference. The measurement results of 3 cyclone III chips are shown in Figure 3.72.

The post process variation removal measurement results of 4 temperature sensors on a Cyclone IV FPGA chip are further automatically calibrated against an accurate external temperature reference in [50], using the automatic self-calibration method proposed in Section 3.4.1. The measurement results are shown in Figure 3.73.
(a) Calibrated outputs after being calibrated to true temperature.

(b) Measurement errors of (a). The dotted line shows 3 sigma errors.

Figure 3.71 Measurement results of 12 delay-line based temperature sensors on 3 custom ICs fabricated using IBM’s 0.13 μm technology, after self-calibration step I and III are performed (cool startup).
Figure 3.72  Measurement results of 12 delay-line temperature sensors on 3 Cyclone III chips, after self-calibration step I and III are performed.
Figure 3.73 Measurement results of 4 delay-line based temperature sensors on a Cyclone IV chip, after self-calibration step I and III are performed.
3.6 Summary

Comparisons with the state-of-the-art digital temperature sensors are shown in Table 3.8 and Table 3.9. As shown in Table 3.8, the proposed delay-line based temperature sensor that is implemented in Cyclone III FPGA has a power of 9 µW and an area of 72 LEs (logic element). The same design with longer delay-line length implemented in Cyclone IV FPGA consumes a power of 14 µW and an area of 256 LEs. The same temperature sensor on 65nm custom ICs consumes a power of 115 µW and an area of 40µm ×40µm (0.016mm²). The SA block (for self-calibration method step I) shown in Figure 3.21 occupies 333 LE (consuming 300 µW), while the rest of the self-calibration block (step II and III) occupies another 427 LE (consuming 370 µW). The synthesized self-calibration block occupies 300 µm ×500 µm using IBM’s 0.13 µm technology. The conversion time is 0.23 µs for the continuous self-calibration on Cyclone III FPGA chip. The digital outputs from Cyclone III and Cyclone IV FPGAs are accumulated and averaged over 128 cycles. The custom IC’s outputs are not averaged.

As shown in Table 3.9, compared with previous publications, the proposed work features: (1) automatic removal of process variations; (2) calibration of multiple sensors on the same chip with only one calibration block; (3) no additional circuitry on the delay-line based temperature sensor is required; (4) reduced sensitivity on power supply variations; and (5) automatic calibration to true temperature with an accurate on-chip bandgap based temperature sensor reference.
### Table 3.8 A Comparison of the State-of-the-art Digital Temperature Sensors and Calibration Methods

<table>
<thead>
<tr>
<th>Sensor</th>
<th>[33]</th>
<th>[21]</th>
<th>[22]</th>
<th>[40]</th>
<th>This Work (Cyclone III)</th>
<th>This Work (Cyclone IV)</th>
<th>This Work (65nm custom IC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Type</td>
<td>Bandgap</td>
<td>CMOS</td>
<td>CMOS</td>
<td>CMOS</td>
<td>CMOS</td>
<td>CMOS</td>
<td>CMOS</td>
</tr>
<tr>
<td>CMOS Technology</td>
<td>0.16µm</td>
<td>0.13µm</td>
<td>0.22/0.18µm</td>
<td>65nm</td>
<td>65nm</td>
<td>60nm</td>
<td>65nm</td>
</tr>
<tr>
<td>Chip Area (mm²)</td>
<td>0.12</td>
<td>0.12</td>
<td>140LE¹</td>
<td>0.01</td>
<td>72LE²</td>
<td>333LE³</td>
<td>256 LE</td>
</tr>
<tr>
<td>Resolution (°C)</td>
<td>0.015</td>
<td>0.78</td>
<td>0.133</td>
<td>0.139</td>
<td>0.36</td>
<td>0.58</td>
<td>0.43</td>
</tr>
<tr>
<td>Error (°C)</td>
<td>±0.2</td>
<td>-1.8~2.3</td>
<td>-0.7~0.6</td>
<td>-5.1~3.4</td>
<td>±2.5</td>
<td>±3.0</td>
<td>±2.5*</td>
</tr>
<tr>
<td>Conversion time (ms)</td>
<td>100</td>
<td>0.2</td>
<td>1</td>
<td>0.1</td>
<td>0.328</td>
<td>0.328</td>
<td>0.4</td>
</tr>
<tr>
<td>Power (µW)</td>
<td>9.2</td>
<td>1200</td>
<td>175</td>
<td>150</td>
<td>300³ (self-calibration)</td>
<td>14</td>
<td>115</td>
</tr>
<tr>
<td>Energy (nJ)</td>
<td>9200</td>
<td>240</td>
<td>175</td>
<td>15</td>
<td>5</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>Res.FOM (pJ°C)¹</td>
<td>170</td>
<td>146</td>
<td>3</td>
<td>290</td>
<td>388</td>
<td>1.68</td>
<td>8.5</td>
</tr>
<tr>
<td>Acc.FOM (nJ%)²</td>
<td>49</td>
<td>3840</td>
<td>20</td>
<td>375</td>
<td>19</td>
<td>45</td>
<td>288</td>
</tr>
<tr>
<td>Calibration</td>
<td>One-point</td>
<td>One-point</td>
<td>One-point</td>
<td>Auto-calibration</td>
<td>Self-calibration</td>
<td>Self-calibration</td>
<td>Self-calibration</td>
</tr>
<tr>
<td>Temperature range(°C)</td>
<td>-30~125</td>
<td>0~100</td>
<td>0~100</td>
<td>0~60</td>
<td>20~80</td>
<td>20~80</td>
<td>20~80</td>
</tr>
</tbody>
</table>

¹Res.FOM = Energy/Conversion×(Resolution) [33], ²Acc.FOM = Energy/Conversion×(Relative inaccuracy) [33], for Bandgap type, digital backend power and area are not included [33]. ³delay-line based temperature sensors (1 LE =0.005 mm² on Cyclone III FPGA, if implemented using standard cells on a custom IC of similar technology node, its area is around 2×2 µm²); ⁴self-calibration (process variation removal) algorithm block; ⁵cool start-up
<table>
<thead>
<tr>
<th>Publications</th>
<th>Reference [21]</th>
<th>Reference [22]</th>
<th>This Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calibration block</td>
<td>On-chip</td>
<td>Off-chip</td>
<td>On-chip</td>
</tr>
<tr>
<td>Requiring additional circuit on sensing devices</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Calibrates how much sensors</td>
<td>One sensor one multiplexer</td>
<td>One sensor one time amplifier</td>
<td>One calibration block calibrates as many sensors, requiring only additional register for storing the correction factor $N_C$</td>
</tr>
<tr>
<td>Prone to power supply voltage variations</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Automatic calibration to true temperature</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Chapter 4  Thermal Management Using the Proposed Delay-line Based Temperature Sensors

Elevated temperature as a result of increased power density gives rise to problems such as degraded device speed and performance, increased failure rate and cooling cost, etc. Power density increase is caused by device scaling and performance demand. Less than ideal scaling of the supply voltages and the threshold voltages have created a situation in which leakage and dynamic power are not keeping pace with geometric scaling [63]. MPSoC (multi-processor systems on chip) architectures lead to localized temperature hot spots that could severely limit the overall system performance, as the speed of transistors is negatively affected by temperature. All circuit breakdown phenomena (e.g., electro migration) are highly temperature dependent. Temperature has become a true limiter to the performance and reliability of computing systems [63].

In this chapter, the temperature sensors designed and experimentally verified in Chapter 2 and Chapter 3 are utilized to study different DTM techniques on an Altera Cyclone IV FPGA based MPSoC test bench. Section 4.1 provides a literature review on the state-of-the-art thermal management technologies. Section 4.2 presents the experimental results of the thermal management system implemented on a Cyclone IV FPGA using the proposed self-calibrated delay-line based temperatures sensors. Four microprocessor cores are mapped onto a Cyclone IV FPGA chip to emulate the VLSI load.

4.1 Literature Review on the State-of-the-Art Thermal Management Technologies

Section 1.1 gives an overview of different cooling solutions and DTM techniques. This section further illustrates the reactive and predictive thermal management methods shown in Figure 4.1.
A typical control scheme has a series of thermal thresholds associated with progressively harsher performance penalties approaching the thermal emergency level. The lowest threshold corresponds to a temperature set-point, or soft threshold, under which the processor will try to maintain its temperature using fine-grained control changes, such as selecting the optimal DVFS (dynamic voltage frequency scaling) setting. Violations of the set-point incur only incremental performance penalties. Each threshold above the set-point, or hard thresholds, is associated with progressively harsher performance penalties as the thermal risk grows. For instance, many commercial processors use clock gating in addition to the lowest DVFS setting when the temperature exceeds a hard limit, which translates into a very harsh performance penalty. If the penalties associated with the hard temperature limits fail to constrain the temperature, it eventually reaches the emergency threshold and the system shuts down in response. This distance between the set-point (soft limit) and the emergency level constitutes the thermal guard band. The upper portion of Figure 4.1 illustrates a sample scheme employed by many popular commercial processors, where the processor incurs harsh clock-gating penalties when the temperature exceeds the hard limit [63].

It can be seen from Figure 4.1 that the predictive DTM strategy has less thermal overshoot and overhead, compared to its reactive alternative. Reference [3] proposes a proactive (predictive) thermal management method based on an autoregressive moving average (ARMA) model, and its flow chart is as shown in Figure 4.2. The predictive DTM shown in Figure 4.2 works as follows [3]. Firstly, temperature measurements are obtained from on-chip temperature sensors. Secondly, based on a moving history window of temperature measurements from previous steps, certain time frame $t_n$’s temperature increments are predicted into the future using an ARMA model. Finally, the threads are reallocated to different cores on the MPSoC to balance the temperature distribution across the die. Experimental results on a real computer microprocessor (UltraSPARC T1) verified that the method proposed in [3] not only reduces the peak temperature but also the thermal gradient by over 60%.

Reference [4] proposes a predictive thermal management. The flow chart of predictive thermal management in [4] is shown in Figure 4.3. For the off-line analysis, principle component analysis (PCA) is performed using computer performance counter information. The principle components are extracted to save calculation effort for later analysis. $k$-means clustering method is used to define the global phase locations that closely approximate the $n$ observations in the
representative workload data. In [4], a phase is a stage of execution in which a workload exhibits nearly identical power, temperature, and performance characteristics. There are 15 different phases defined in [4]. For the runtime analysis shown in Figure 4.3, the current performance condition is firstly identified as one of the 15 phases, using the runtime performance counter readings. Then the predictive thermal manager predicts the future processor temperature, which is determined by the current phase (different frequency assignment scenarios according to the phase-dependent state-space model). In [4], a core’s temperature is predicted as a linear function of the temperature of its own and those of its neighbouring cores, and their activities (phases):

$$T_i[k+1] = \sum_{j=1}^{N} a_{ij} T_j[k] + b(f, p)$$  \hspace{1cm} (4.1)

where \( b(f, p) \) is a function of the current phase \( p \) and frequency \( f \). \( a_{ij} \) is the thermal coupling between cores and it a function of their physical proximity. Then thermal management unit chooses the highest frequency that does not lead to a future thermal violation.

**Figure 4.1** Illustrations of the DTM control strategy employed by the Core i7 processor compared to a predictive approach [63].
Figure 4.2  Flow chart of the proactive thermal management in [3].

Figure 4.3  Flow chart of predictive thermal management in [4].

FPGAs provide fast emulation frameworks for thermal management systems in MPSoCs [14][17][18]. Power dissipation in modern FPGAs increases with their logic densities to the point
that managing power dissipation becomes a primary concern for designs beyond the 90nm technology node [19]. In [17], a hardware based MPSoC is mapped onto a FPGA, where runtime information is extracted. The information is then used to interact in real-time with a software thermal model running on a host computer via an Ethernet port. The software evaluates the real-time thermal behavior of the MPSoC design and returns its feedback to the FPGA emulating MPSoC. The experimental results show that [17] saves simulation time by three orders of magnitude compared to the conventional MPSoC simulators. The MPSoC temperatures in [14] and [18] are monitored using CMOS delay-line based temperature sensors proposed in [20]. In [14], an MPSoC is implemented on a Xilinx Virtex-2 Pro (XC2VP7) FPGA. The temperatures on the FPGA based MPSoC are monitored using the CMOS delay-line based temperature sensors in [20]. The readings from the temperature sensors are compared with simulation results from the same setup in HotSpot simulation software designed by the same authors in [14]. Different floor-plans are enabled for the above comparison. The experiments in [14] show spatial temperature gradient can be as much as 7 °C on the Xilinx Virtex-2 Pro FPGA. The explanations are that blocks such as the register files and the issue queues in a superscalar processor are small in area, but are accessed frequently. This results in local hotspots across the chip. Reference [18] proposes a power/thermal management system on a FPGA emulating MPSoC comprised of four Cyclone III FPGAs connected by an Ethernet to mimic a multi-core microprocessor. A Nios II Embedded Evaluation Kit (NEEK) test bench is used to evaluate power/thermal management methods. Each core has its own clock and temperature sensor. The temperature sensor is as proposed in [24]. DFS (dynamic frequency scaling) is used in [18] as a DTM/DPM technique. Linear performance improvements are observed as the number of cores increases in [18]. However, as the temperature sensors in [18] are not calibrated, their readings only provide relative temperature trends instead of true temperature readings.

Dynamic power management (DPM) is often compared with DTM in their effectiveness in optimizing thermal profiles and performances of VLSI chips. A detailed comparison can be found in Appendix A.
4.2 Thermal Management Demonstrated on a 60nm FPGA using Proposed Delay-line based temperature sensors

In [14] and [18], the MPSoC temperature is monitored using conventional delay-line based temperature sensors [20]. However, it is reported in [23] that the conventional delay-line based temperature sensor’s readings are affected by power supply voltage variations. Experiments on a Xilinx Virtex-5 FPGA verified that an instantaneous change from 0 to 80 % utilization running at 100 MHz leads to a 73 °C error in the estimated temperature [23]. This problem of sensitivity to power supply level changes is solved in Chapter 3’s Section 3.3, which presents a fully digital self-calibration method that removes sensitivities to power supply voltage and process variations for multiple all-digital delay-line based temperature sensors. This section studies different VLSI thermal management techniques implemented with on-chip all-digital self-calibrated delay-line based temperature sensors proposed in Chapter 3. Four microprocessor cores are mapped onto a Cyclone IV FPGA chip and their runtime thermal profiles are monitored by four on-chip all-digital temperature sensors located close by. The runtime thermal profiles of the four microprocessor cores are plotted, for eight different DTM techniques: reactive global DFS, reactive local DFS, reactive thread migration, reactive hybrid DFS, predictive global DFS, predictive local DFS, predictive thread migration and predictive hybrid DFS. This paper introduces a hybrid DTM technique that combines the benefits of global Dynamic Frequency Scaling (DFS) and thread migration. The hybrid DTM could predict thermal violations and avoid performance penalty ahead of time. Performance parameters such as the percentages of time that the MPSoC spends in higher temperatures (>50 °C) and larger thermal gradients (>5 °C) and the MPSoC processing rate are compared for the above-mentioned eight different DTM techniques. The proposed predictive hybrid DTM is found to be effective in optimizing the above parameters.

4.2.1 Thermal modeling and prediction

Figure 4.4 shows a common lumped RC thermal model [2]. The lumped RC thermal model is analogous to an electrical RC network [2]. \( C_T \) is the thermal capacitance (heat capability, \( J/°C \)) and \( R_T \) is the thermal resistance (\( Θ, °C/W \)). In reference to [7], this paper assumes that the temperature increment in each of the cores is affected by the temperature differences between the current and ambient or neighboring cores’ plus its own power dissipation:
\[ C_i \frac{dT_i(t)}{dt} = \sum_{j=1}^{N} \left( \frac{T_j - T_i}{R_{j-i}} \right) + \frac{T_A - T_i}{R_{i-A}} + P_{i,CORE} \]  \hspace{1cm} (4.2) 

\( C_i \) is the thermal capacitance of core \( i \). \( R_{j-i} \) and \( R_{i-A} \) are the thermal conductance between core \( j \) and core \( i \), core \( i \) and ambient, respectively. \( P_{i,CORE} \) is the power consumption of core \( i \).

\[ C \frac{dV}{dt} = \frac{V_0 - V}{R} + i \]  \hspace{1cm} \text{Electrical}

\[ C_i \frac{dT_i}{dt} = \frac{T_0 - T}{R_T} + P \]  \hspace{1cm} \text{Thermal}

**Figure 4.4** The lumped RC thermal model analogous to an electrical RC network [2].

The \( R_{i-A} \) obtained using Altera datasheet is 18.9 °C/W (with still air flow). The silicon thermal resistivity of silicon used in [14] is 13.9 °C/W. The thermal capacitance \( C_i \) of each core is assumed to be identical and it is found through empirical curve fitting. The thermal management system test bench contains four microprocessor cores. Each core has eight 8085 processors, and the block diagram of an 8085 is shown in Figure 4.5 (in reference to Terasic DE2-115 development board kits). Each core is located on each of the four corners of the Cyclone IV FPGA, as shown in Figure 4.6. The 8085 processor is implemented using verilog code (in reference to Altera DE2-115 development board kits) and utilizes 6000 LE. A delay-line based temperature sensor is placed next to each core for thermal sensing. The self-calibration
algorithm, the thermal management system and the timing (four PLLs) blocks implemented in the middle of the chip use 3000 LE (as shown in Figure 4.6).

Figure 4.5  Block diagram of the 8085 processor used in each core.
The procedures to obtain $C_i$ are as follows. First of all, the run control for one core (e.g. core #1) is enabled while those for the remaining are disabled. Then the temperatures of the four cores are modeled using (4.2) as:

\[
\begin{align*}
C_1 \frac{dT_1(t)}{dt} &= \sum_{j \neq 1}^\infty \left( \frac{T_j - T_1}{R_{j-1}} \right) + \frac{T_A - T_1}{R_{1-A}} + P_{1,\text{CORE}} \\
C_2 \frac{dT_2(t)}{dt} &= \sum_{j \neq 2}^\infty \left( \frac{T_j - T_2}{R_{j-2}} \right) + \frac{T_A - T_2}{R_{2-A}} + P_{2,\text{CORE}} \\
C_3 \frac{dT_3(t)}{dt} &= \sum_{j \neq 3}^\infty \left( \frac{T_j - T_3}{R_{j-3}} \right) + \frac{T_A - T_3}{R_{3-A}} + P_{3,\text{CORE}} \\
C_4 \frac{dT_4(t)}{dt} &= \sum_{j \neq 4}^\infty \left( \frac{T_j - T_4}{R_{j-4}} \right) + \frac{T_A - T_4}{R_{4-A}} + P_{4,\text{CORE}} \\
\end{align*}
\]

(4.3)

\[
\begin{align*}
C_1 &= C_2 = C_3 = C_4 \\
R_{1-A} &= R_{2-A} = R_{3-A} = R_{4-A} \\
R_{1-2} &= R_{1-3} = R_{2-3} = R_{3-4} = \frac{1}{\sqrt{2}} R_{1-4} = \frac{1}{\sqrt{2}} R_{2-3} \\
T_A &= \text{room temperature}
\end{align*}
\]

(4.4)
where $R_{t-4} = 12.6 \, ^\circ C/W$. The power consumption of each core is:

\[
\begin{align*}
    P_{1,\text{CORE}} &= 1.2 \, W \\
    P_{2,\text{CORE}} &= P_{3,\text{CORE}} = P_{4,\text{CORE}} = 0
\end{align*}
\] (4.5)

The assumptions made in (4.5) does not account for leakage power in other cores than Core #1. $P_{1,\text{CORE}}$ is obtained by measuring the power on the FPGA. In (4.3), all the parameters are known except $C_i$. An initial $C_i$ value is used for (4.3), and based on the initial $C_i$ value the temperatures of four cores are solved using the ODE45 function in MATLAB. Then the temperatures of four cores are compared against those in real measurement, until $C_i$ value is increased gradually for a better fitting. The temperatures of four cores solved using (4.3) with the final $C_i$ value ($C_1 = C_2 = C_3 = C_4 = 6.4 \, J/\, ^\circ C$), being compared with the measurement results, are shown in Figure 4.7. The above procedures of solving $C_1$ through $C_4$ were performed before the experiments on thermal management system in Section 4.2.2 are carried out.

![Measurement vs Modeling Temperature Profiles](image)

**Figure 4.7** Measurement and modeled temperatures on the FPGA emulating MPSoc shown in Figure 4.6.
According to (3.39), the maximum rate of temperature change is +0.3 °C/s for the Cyclone IV FPGA test bench. The errors allowed for DTM thermal monitoring is 5 °C [41]. Therefore for a temperature change of 5 °C to take place, the minimum time is 18 seconds. In this experiment, the temperature is predicted up to 18 seconds ahead of time using (4.3). The predicted temperature is compared against the measurement results in MATLAB, as shown in Figure 4.8. Figure 4.9 shows the errors between the predicted temperature and measured temperature for 18 seconds later. The temperature measurements used to estimate the errors in Figure 4.9 are smoothed (averaged) for every 18 seconds using the measurements shown in Figure 4.8.

Thermal prediction for the DTM test bench setup that will appear in Section 4.2.2 is given as follows. The prediction makes use of (4.3) and the measured power dissipation of each core. The measured power dissipation is shown in Table 4.1. The prediction is programmed using verilog language on the Cyclone IV FPGA chip for real time thermal prediction. In the test bench, Core #1, 2, 3 are assigned to perform addition, subtraction and moving functions, respectively. Core #4 is idling. The predicted temperature is compared against the measured temperature on the same FPGA. Both temperatures are read out from the FPGA using the SignalTap function [64]. Figure 4.10 shows the errors between predicted temperature and measured temperature. The estimation errors using the prediction model are less than those using current temperature.
Figure 4.8 Measured vs. predicted temperatures for four cores. Only Core #1’s run control is enabled.
Figure 4.9 Errors resulted from using the predicted temperatures for thermal estimation.

Table 4.1 Power Consumption of the FPGA Without DTM With Four Cores Running

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Voltage drop on $R_{\text{sense}}$ (mV)</th>
<th>Current (mA)</th>
<th>Resistance (Ohm)</th>
<th>Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before configure</td>
<td>1.2158</td>
<td>0.317</td>
<td>5.28</td>
<td>0.003</td>
</tr>
<tr>
<td>800 MHz</td>
<td>1.21</td>
<td>6.3</td>
<td>105</td>
<td>0.003</td>
</tr>
<tr>
<td>400MHz</td>
<td>1.2115</td>
<td>4.15</td>
<td>69.1</td>
<td>0.003</td>
</tr>
<tr>
<td>200Hz</td>
<td>1.2151</td>
<td>1.958</td>
<td>16</td>
<td>0.003</td>
</tr>
</tbody>
</table>
Figure 4.10  Errors resulted from using the predicted temperatures for thermal sensing.

4.2.2  Experimental results of thermal management on Cyclone IV FPGA using proposed delay-line based temperature sensors

This section uses the floor plan shown in Figure 4.6 and presents runtime thermal profiles of the four cores on the Cyclone IV based MPSoC, when different DTM techniques are used. The digital outputs shown in this section are all read out from the FPGA using the SignalTap function [64].

When a runtime DTM technique (e.g. global DFS) is used, the workload condition changes. This leads to temperature reading errors in the delay-line based temperature sensors, as reported in [23]. In this situation, the workload change is mistaken for a temperature change. The above problem can be alleviated by utilizing the self-calibration method that removes the sensitivity to supply voltage variation proposed in Section 3.3. Figure 4.11 shows the experimental results of the temperature sensor outputs with and without the self-calibration method proposed in Section 3.3. It shows that without the proposed self-calibration, the sensor mistakenly reads a –26 °C
change within 0.7 second (at around $t = 400$ s). However, according to (4.2), the rate of temperature change should be no larger than $+0.3 \, ^\circ C/s$ for the FPGA emulating MPSoC under test. This mistake is corrected when the self-calibration proposed in Section 3.3 is performed.

![Figure 4.11](image1)

**Figure 4.11**  Output codes versus time in Core #1 on the Cyclone IV FPGA, with and without the proposed self-calibration proposed in Section 3.3 to remove sensitivity to supply voltage variations.

![Figure 4.12](image2)

**Figure 4.12**  Power supply level of the FPGA chip, measured at the same time as the experimental results shown in Figure 4.11.

On the Cyclone IV FPGA emulating MPSoC, Core #1, 2, 3 are assigned to perform addition, subtraction and move functions, respectively. Core #4 is idling. The addition function has a higher toggling frequency than the subtraction function. Subtraction function has higher toggling
frequency than the move function. Since Core #1 performs addition and it is close to two other non-idling cores, it has the highest temperature among the four. Since Core #4 is idling and its distance to the hottest Core #1 is the longest, it has the lowest temperature.

Figure 4.13 shows the runtime profiles of the four cores on the Cyclone IV FPGA without dynamic thermal management (DTM). Figure 4.14 to Figure 4.21 show the thermal profiles of four different cores when different DTM techniques are used. The different DTM techniques include: reactive global DFS, reactive local DFS, reactive thread migration, reactive hybrid DFS, predictive global DFS, predictive local DFS, predictive thread migration and predictive hybrid DFS, as listed in Table 4.2. The details of the DTM strategies are described as follows. The initial clock frequency is 800 MHz, and when DFS is in use, it is reduced by half when the temperature exceeds 45 °C and by another half when the temperature exceeds 50 °C. Clock gating is applied to a particular core when local DFS is in use, if the temperature exceeds a predetermined limit (55 °C in this case). This temperature threshold is conservatively set to be lower than the typical case temperature limit of 85 °C. For global DFS, all the cores share one clock, and for local DFS each core has its own clock. Clock gating is applied to all the cores when global DFS is in use, if the temperature exceeds 55 °C in at least one core. Glitch free DFS is achieved using Altera’s ALTPLL_RECONFIG mega function [65]. Thread migration assigns the job from the hottest core’s to the idle core (Core #4) when their thermal gradient reaches a threshold of 5 °C.

The percentage of time that each of the four cores spends in a particular temperature range when using different DTM techniques as shown in Figure 4.22 to Figure 4.30.
Figure 4.13  Thermal profiles of all four cores on the Cyclone IV FPGA without DTM.

Figure 4.14  Thermal profiles of all four cores on the Cyclone IV FPGA with reactive global DFS DTM.
Figure 4.15  Thermal profiles of all four cores on the Cyclone IV FPGA chip with reactive local DFS DTM.

Figure 4.16  Thermal profiles of all four cores on the Cyclone IV FPGA chip with reactive thread migration DTM.
Figure 4.17  Thermal profiles of all four cores on the Cyclone IV FPGA chip with a combined hybrid reactive global DFS and thread migration DTM.

Figure 4.18  Thermal profiles of all four cores on the Cyclone IV FPGA chip with predictive global DFS DTM.
Figure 4.19  Thermal profiles of all four cores on the Cyclone IV FPGA chip with predictive local DFS DTM.

Figure 4.20  Thermal profiles of all four cores on the Cyclone IV FPGA chip with predictive thread migration DTM.
Figure 4.21  Thermal profiles of all four cores on the Cyclone IV FPGA with a combined hybrid predictive global DFS and thread migration DTM.

Figure 4.22  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when without DTM.
Figure 4.23  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with reactive global DFS DTM.

Figure 4.24  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with reactive local DFS DTM.
Figure 4.25  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with reactive thread migration DTM.

Figure 4.26  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with a combined predictive hybrid global DFS and thread migration DTM.
Figure 4.27  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with predictive global DFS DTM.

Figure 4.28  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with predictive local DFS DTM.
Figure 4.29  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with predictive thread migration DTM.

Figure 4.30  The percentage of time each of the four cores spends in each temperature range on the Cyclone IV FPGA chip, when with a combined predictive hybrid e global DFS and thread migration DTM.
4.3 Summary

Figure 4.33, Figure 4.32, Figure 4.33 and Table 4.2 compare different DTM techniques on their capabilities to optimize the VLSI test bench’s thermal profiles, thermal gradients and keeping the performance stable. The experimental data used for comparison is those shown from Figure 4.13 to Figure 4.21.

All different DTM techniques are tested using the same test bench and floor plan. The eight different DTM techniques are enabled by different combinations of switch positions on the DE2-115 development board while being configured by the same sof file. There are several observations that could be made from the comparisons in Figure 4.33 and Table 4.2. First of all, both power saving techniques such as global or local DFS and task re-distribution strategies such as thread migration are effective in reducing the peak temperature for the core. Secondly, comparing global and local DFS, the former is more effective in reducing peak temperature while the latter is more effective in reducing thermal gradient. As for local DFS each core has its own clock, and each core’s temperature is adjusted according to its own temperature profile, therefore the spatial thermal gradients among the cores are lower than those found in core when using the global DFS. Instead, with global DFS, all the cores share one clock and all the cores’ temperatures are reduced at the same time so the spatial thermal gradients remain. Thirdly, there are less spatial thermal gradient when the predictive method is used, compared with its reactive alternatives. Fourthly, for thread migration the total temperature profiles are improved for the four cores without degradation in the total performance. For example, it can be observed that in Figure 4.16 (thermal profile when purely thread migration DTM is used) that the temperature of Core #4 rises after the thread is migrated to itself. In contrast, in Figure 4.13 (no DTM) Core #4’s temperature is the lowest. In Figure 4.16, after Core #4’s temperature rises, Core #1’s temperature is lower than that in Figure 4.13 where no DTM is used. Finally, in the proposed hybrid DTM, thread migration is introduced to minimize thermal gradient while the global DFS is used to reduce peak temperature. In comparison to a conventional global DFS approach, the proposed hybrid DTM reduces the amount of time that the MPSoC spends at higher temperatures and larger thermal gradients, by 10% and 21%, respectively. In addition, the proposed hybrid DTM offers a 10% improvement in the average processing rate (instructions per second) when compared with the conventional global DFS approach.
In Figure 4.16, Figure 4.17, Figure 4.20 and Figure 4.21 where thread migration are applied, the thermal gradient still exists. The explanations for the above observation are as follows. (1) In Figure 4.16 and Figure 4.20 where only thread migration is used, the maximum spatial thermal gradient are between Core #1 and Core #3, after the thread on Core #1 is migrated to Core #4. (2) Thermal diffusion causes the heat to transfer between cores. The effectiveness of thread migration, compared with the above diffusion phenomenon, is not very obvious. In the experiments without DTM, the steady state is reached earlier in the time scale than those with DTM. Therefore for the former, the spatial thermal gradient is lower. (3) The current temperature is a function of past temperature and the temperature differences between the current and neighboring cores, as indicated in (4.2). That is to say, since Core #1’s temperature is higher in the past, even it doesn’t dissipate dynamic power any longer after its thread is migrated to Core #4, its temperature may still rise as power dissipated by nearby cores is being diffused to itself. E.g. For example, the close proximity of Core #1 and Core #2 will always lead to similar temperatures. (4) Due to process variations, even with the same activity, different cores may have different temperatures [7]. Core #1 might have a higher leakage power. (5) Other factors, including power dissipated by the connecting wires (reference [66] reports that interconnecting cells are responsible for 1/3 of power on FPGA), affect power and thermal behaviors.

![Chart comparing different DTM techniques on higher temperatures.](image)

**Figure 4.31** Comparisons for different DTM techniques on higher temperatures.
Figure 4.32 Comparisons for different DTM techniques on larger spatial thermal gradients.

Figure 4.33 Comparisons for different DTM techniques on processing rate.
<table>
<thead>
<tr>
<th>Thermal Estimation</th>
<th>DTM Method</th>
<th>Percentage of time when temperature &gt; 50 °C (%)</th>
<th>Percentage of time when thermal gradient &gt; 5 °C (%)</th>
<th>Processing rate compared to case without DTM (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>No DTM</td>
<td>51</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>Reactive</td>
<td>Global DFS</td>
<td>20</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>Reactive</td>
<td>Local DFS</td>
<td>49</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Reactive</td>
<td>Thread Migration</td>
<td>44</td>
<td>67</td>
</tr>
<tr>
<td>4</td>
<td>Reactive</td>
<td>hybrid</td>
<td>10</td>
<td>59</td>
</tr>
<tr>
<td>5</td>
<td>Predictive</td>
<td>Global DFS</td>
<td>41</td>
<td>82</td>
</tr>
<tr>
<td>6</td>
<td>Predictive</td>
<td>Local DFS</td>
<td>26</td>
<td>49</td>
</tr>
<tr>
<td>7</td>
<td>Predictive</td>
<td>Thread Migration</td>
<td>47</td>
<td>52</td>
</tr>
<tr>
<td>8</td>
<td>Predictive</td>
<td>hybrid</td>
<td>23</td>
<td>0</td>
</tr>
</tbody>
</table>
Chapter 5  Conclusions

5.1  Thesis Contributions

The motivation of this thesis is to overcome thermal induced problems on VLSI chips due to the increased power density as a result of the growing performance demand and technology scaling. To overcome the above problems, this thesis presents a low power self-calibrated temperature sensor for thermal monitoring to facilitate a thermal management system. The proposed temperature sensor is demonstrated on a FPGA emulating MPSoC based DTM system. The motivation and the contribution of the thesis are discussed in detail in Chapter 1.

As a large number of temperature sensors are needed on-chip for thermal sensing, Chapter 2 presents a power saving technique for delay-line based temperature sensors. The power of a delay-line based temperature sensor is minimized using a tab decoding approach. In the approach, for a $M$-bit ring-oscillator based temperature sensor, the counter provides the $(M-N)$ MSBs while the decoder captures and interprets the position where the input reset pulse stops within the ring oscillator at the falling edge of reset signal and generates the remaining $N$ LSBs. Therefore the power is saved by having a counter of fewer bits (having a fewer number of toggling gates). Experimental results on a 65nm Cyclone III FPGA verified that the proposed power saving technique saves power up to 70% without incurring additional area, compared with the conventional technique that only uses a counter for decoding in [20].

Due to the large amount of sensors on-chip, the calibration procedure for delay-line based temperature sensors has to be time and cost efficient. Chapter 3 introduces a self-calibration method for multiple delay-line based temperature sensors. The method compensates the effect of process variations by assigning a unique correction factor, $N_C$ to each sensor, making all the sensors’ calibrated outputs to be the same at start-up. The self-calibration method is further expanded with capabilities to remove the temperature sensors’ sensitivity to power supply variation and to automatically calibrate the delay-line based sensors against an accurate on-chip bandgap based temperature sensor reference. For the automatic self-calibration feature, an
accurate bandgap based temperature sensor digitalized by a SAR ADC is implemented on-chip. Experimental results on three 65nm custom ICs, three 0.13 µm custom ICs, three Cyclone III and one Cyclone IV FPGA verify that the measurement errors resulted from using the proposed self-calibration method are kept within ±2.5 °C (without power supply variations) and ±4 °C (with power supply variations) despite different startup (calibration) temperatures. Compared with previous publications, the method proposed in this thesis features: (1) automatic removal of process variations; (2) calibration of multiple sensors on the same chip using only one calibration block; (3) no additional circuitry on the delay-line based temperature sensor is required so that the entire calibration block could be placed at a non-critical location on-chip or off-chip; (4) reduced sensitivity on power supply variations; and (5) automatic calibration to the true temperatures with an accurate on-chip bandgap based temperature sensor reference.

Chapter 4 studies different VLSI thermal management techniques implemented with on-chip all-digital self-calibrated delay-line based temperature sensors proposed in Chapter 2 and Chapter 3. Four microprocessor cores are mapped onto a Cyclone IV FPGA chip and their runtime thermal profiles are monitored by four on-chip all-digital temperature sensors located close by. The runtime thermal profiles of the four microprocessor cores are plotted, for eight different DTM techniques: reactive global DFS, reactive local DFS, reactive thread migration, reactive hybrid DFS, predictive global DFS, predictive local DFS, predictive thread migration and predictive hybrid DFS. This thesis introduces a hybrid DTM technique that combines the benefits of global Dynamic Frequency Scaling (DFS) and thread migration. The hybrid DTM could predict thermal violations and avoid performance penalties ahead of time. Performance parameters such as the percentages of time that the MPSoC spends in higher temperatures (>50 °C) and larger thermal gradients (>5 °C) and the MPSoC processing rate are compared for the above-mentioned eight different DTM techniques. The proposed predictive hybrid DTM is found to be effective in optimizing the above parameters.

5.2 Future Trends

One of the future trends is to keep the area and power incurred by the temperature sensor to a minimum. This is because on the VLSI chips the temperature sensors play the auxiliary role of thermal sensing rather than the main function of computation, and an accuracy of 3 to 5 °C is acceptable [41]. The area and power could be further reduced at the expenses of reduced but still
acceptable accuracy and resolution. Another way to minimize power and area is to locate the thermal sensing circuits close to the hotspots and to assign other circuits (ADCs, digital back-end blocks) to non-critical locations or off-chip, such as in [31]. Another stream of minimizing area and power is to use a limited number of temperature sensors and then to up-sample to reconstruct the temperature profile [12], [13]. E.g. Reference [12] fully reconstructs the thermal status of an integrated circuit during runtime using a minimal number of thermal sensors. Based on the Nyquist–Shannon sampling theory, this method applies to both uniform and non-uniform thermal sensor placements, generating a thermal profile with absolute error of 0.6 %. In sum, for area and power minimization, either the number or the circuit complexity of the temperature sensors need to be reduced, being compensated by more advanced digital back-end calibration approaches. The digital back-end calibration circuits could be either located at non-critical locations, or off-chip implemented using software approach.

Another future trend in thermal sensing is for the temperature sensors to share the same power grid with the digital logic blocks they are monitoring. It is best that the digital temperature sensors are immune to substrate noise [41]. Or the noise could be reduced using digital back-end calibration blocks.

The calibration techniques have to be more advanced in the future. As explained by the authors in [31], with device scaling, transistor geometry variations make static measurements difficult as the contributions from voltage offsets and flicker noise dominate the error budget. Either larger area overhead has to incur due to the extra technique circuitries to minimize variations, or the temperature calibration overhead has to take place during high-volume manufacturing [31]. It is desirable that the calibration method could be intended for batch or technology level. For example, one calibration circuit could calibrate as many temperature sensors and the calibration circuit is all-digital and independent from the sensing circuitries so that it could be located at non-critical locations.

One of the future directions for thermal management is to relate a reduction in heat dissipation to the energy savings. The DTM could be combined with DPM for optimal power saving and minimum degradation in the chip throughput performances [3][4][7].

For thermal management, accurate prediction of temperature remains a critical research issue, especially in a dynamic environment where the complexity of solving thermal models is very
high. As the processors scale to include an increasing numbers of cores, the complexity of thermal model driven scheduling will become quite high and novel optimization approaches are required. At the same time, multi-core processors are expected to become more heterogeneous, calling for even greater challenges in managing on-chip resources [7].

5.3 Future Work

Based on the thesis contributions and future trends discussed in Section 5.1 and 5.2, this thesis’s work could be further expanded in the future in the following aspects:

- As both the proposed self-calibrated temperature and its self-calibration algorithm are fully synthesizable, they can be incorporated onto different digital technology platforms to obtain a detailed real-time thermal map. This is analogous to the thermal image collected by an infrared camera. This thermal map obtained using the temperature sensors provides runtime information to facilitate the thermal management systems. Up-sampling techniques [12], [13], when necessary, can be used. This extended work could help gain more insight into the overall thermal profile of the MPSoC during its runtime. It would also lead to, for example, identification of hotspot locations.

- Further improvement on thermal behavior can be achieved if a more efficient thread migration algorithm [67] or Dynamic Voltage Scaling (DVS) could be used. Chapter 4 proposes migrating threads between Core #1 and Core #4. However, after the hottest thread is migrated from Core #1 to Core #4, the coolest core is Core #3. At the moment, the thread could be swapped between Core #4 and Core #3 to achieve a lower spatial thermal gradient. DVS is not studied in this paper, as there is no provision to change the power supply level of the Cyclone IV FPGA chip on the development printed circuit board.

- Heterogeneous cores, instead of identical ones, could be used as the MPSoC cores in the extend work. As a programmable tool, the FPGA could be used to emulate different types of MPSoCs. In Chapter 4, all four cores are identical, but are assigned different tasks. In the future work, the four cores could have different architectures. The FPGA could be configured into a PowerPC microprocessor, analogous to the experiments performed in [14] and [17].
The temperature sensor could be used as part of a PVT (process, voltage and temperature) sensor. As a result of technology scaling, the heterogeneity among process, voltage and temperature among different chips as well as those on the same chip increase. Therefore there is a need for a larger number of PVT sensors. For example, on a VLSI chip, there may be several voltage islands, on which there are multiple PVT sensors (An example is shown in Figure 1.3). In the future work, the thermal management system demonstrated in Chapter 4 could comprise part of a DPM/DTM system. The process, voltage and temperature information could be collected by a DPM control unit to facilitate decision in adjusting the power supply voltage level of a PMU. In the meantime, the DPM control unit takes runtime performance requirement into consideration.
References


[29] K. Souri, C. Youngcheol, and K. Makinwa, "A CMOS temperature sensor with a voltage-calibrated inaccuracy of \( \pm 0.15 \, ^\circ C(3\sigma) \) from \(-55 \, ^\circ C \) to \( 125 \, ^\circ C \),” in Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International, 2012, pp. 208-210.


[33] K. Souri and K. A. A. Makinwa, " A 0.12 mm\(^2\) 7.4 \( \mu \)W micropower temperature sensor with an inaccuracy of \( 0.2 \, ^\circ C (3\sigma) \) from \(-30 \, ^\circ C \) to \( 125 \, ^\circ C \), " Solid-State Circuits, IEEE Journal of, vol. 46, pp. 1693-1700, 2011.


165


Appendix A: DTM vs. DPM

Dynamic power management (DPM) is often compared with DTM in their effectiveness in optimizing thermal profiles and performances of VLSI chips. DPM is a feature of the runtime environment of a power-managed circuit (PMC) that adaptively reconfigures itself to provide the requested services and performance levels with a minimum number of active components or a minimum activity level on such components [2]. DPM encompasses a set of techniques that achieve energy-efficient computation by selectively turning off (or reducing the performance of) circuit components when they are idle (or partially unexploited) [2]. The fundamental premise of the applicability of DPM is that the circuits (and its functional blocks) experience non-uniform workloads during the operation time. Such an assumption is valid for many circuits. A second assumption of the DPM is that it is possible to predict, with a certain degree of confidence, the fluctuations of workload [2]. The key differences between DPM and DTM can be stated as follows. Localized heating occurs much faster than chip-wide heating. Additionally, power dissipation is spatially non-uniform across the chip, resulting in the emergence of hot spots and spatial temperature gradients that can cause timing errors or even physical damage. These effects evolve over time scales of hundreds of microseconds, which imply that the power management techniques must directly target the spatial temperature behavior of the chip temperature. In fact, many DPM techniques have little or no effect on substrate temperature, because they do not reduce the power density in hot spots, or reduce the power dissipation with fine timing granularity. In fact they do not attempt to reduce power dissipation when no positive timing slack is present. Furthermore, DPM techniques attempt to lower the total energy consumed over the entire application run, while DTM techniques must ensure that a thermal limit is not exceeded. The power control algorithm tracks the power consumption of the entire chip as a whole, while the temperature control algorithm concentrates on the power consumption of specific localized structures on chip. Finally, DPM algorithms seek to minimize energy while meeting a task completion deadline whereas with DTM algorithms, there is no minimal performance target other than not exceeding a temperature threshold. In other words DPM is a constrained optimization problem whereas DTM is often formulated as an unconstrained optimization problem [2].
Appendix B: Screenshots of verifications in ModelSim of self-calibration methods proposed in Section 3.2 and 3.4

Figure B.1 and Figure B.2 show the simulation results in ModelSim, to verify the self-calibration method proposed in Section 3.2 that removes process variations among multiple sensors. In Figure B.1, the un-calibrated codes M1 (/ToFPGA_tb/To1/M1) decreases as a function of time, which represents its output decrease versus temperature. For example, in Figure B.1 M1’s output at 20 °C is 3485, at 25 °C is 3458, and at 75 °C is 3335. There are deviations among the un-calibrated outputs M1, M2, M3 and M4, which are from real measurements of four temperature sensors on a Cyclone III FPGA chip. The correction factors Nc1, Nc2, Nc3, Nc4 differ for the four sensors. For example, M2 whose outputs codes are the largest has the smallest correction factor among the four as the product of un-calibrated output codes and the correction factor is a constant, as indicated in (3.29).
Figure B.1 ModelSim verification of self-calibration method that removes process variations proposed in Section 3.2.

Figure B.2 that is a zoom in of Figure B.1 around the yellow line shows the calibrated outputs $M_{\text{tr}}$ (as shown in Figure 3.22 (b)) for four temperature sensors (at 45 °C). The “shift” codes: 00, 01, 10, or 11 signals the time each sensor’s calibrated codes shows up in $M_{\text{tr}}$. For example, when “shift” is 00, Sensor #1’s outputs are those in $M_{\text{tr}1}$. The deviations among the calibrated outputs ($M_{\text{tr}}$) of different temperature sensors are much less than those of the un-calibrated codes ($M_{1}$~$M_{4}$). The deviations of the former translate to around 1 °C in Figure B.2.
Figure B.2  ModelSim verification of self-calibration method that removes process variations proposed in Section 3.2 (zoom in of Figure B.1).

Figure B.3 shows the ModelSim simulation of the self-calibration method proposed in Section 3.4. This step III in the self-calibration procedure automatically calibrates the delay-line based temperature sensor outputs to an accurate temperature sensor reading on-chip. In Figure B.3, the “Mref” is the temperature reference’s readings (in absolute temperature). The “shift” codes: 0, 1, 2, or 3 signals the time each sensor’s calibrated codes shows up in D_cal3. For example, when “shift” is 1, Sensor #1’s outputs are those in D_cal3. “D_cal3” is calibrated output that has be correlated to true temperature with an accurate temperature reference, as shown in Figure 3.25 (c). When “Mref” is 343, the D_cal3 of four temperature sensors are between 341 and 343, which has less than 2 °C errors when compared to true temperature.
Figure B.3 ModelSim verification of self-calibration method that automatic calibrating the delay-line based temperature sensor outputs to an accurate temperature sensor reading on-chip, as proposed in Section 3.4.
Appendix C: Layout of the accurate bandgap temperature sensor reference proposed in Section 3.4.2

A layout of the accurate temperature sensor reference appears in Section 3.4.2 is as shown in Figure C. 1. The layout that includes the SAR capacitor array excluded is done using TSMC’s 65nm 1V technology in Cadence Virtuoso tools. The area occupied by each block is as labeled.

Figure C.1 Layout of the accurate on-chip temperature sensor using TSMC’s 65nm technology in Cadence Virtuoso (SAR capacitor array included).
Copyright Acknowledgements

Part of Chapter 2 is a reprint from the following publications:

Part of Chapter 3 is a reprint from the following publication:

Part of Chapter 4 is a reprint from the following publication: