





Confidential

	
ICT-2009.3.2-248603-IP	
Modelling, Control and Management of Thermal Effects in Circuits of the Future	
	

	WP no.	Deliverable no.	Lead participant
	WP3	D3.4.3	CEA-LETI
Report on the thermal and energy optimization of a SoC using thermal and energy prediction schemes based on activity monitors, and a thermal-aware clock network featuring body-biasing			
Prepared by	Alexandre VALENTIAN (CEA-LETI)		
Issued by	THERMINATOR Project Office		
Document Number	THERMINATOR/D3.4.3/v1.0		
Dissemination Level	Confidential		
Date			

© Copyright 2010-2013 STMicroelectronics, Intel Mobile Communication, NXP Semiconductors, GRADIENT DESIGN AUTOMATION , MUNEDA, SYNOPSYS , BUDAPESTI MUSZAKI ES GAZDASAGTUDOMANYI EGYETEM , CSEM, FRAUNHOFER , IMEC, CEA-LETI, OFFIS, Politecnico di Torino, ALMA MATER STUDIORUM -Universita' Di Bologna, ST-Polito.

This document and the information contained herein may not be copied, used or disclosed in whole or in part outside of the consortium except with prior written permission of the partners listed above.

Document

Title	Report on the thermal and energy optimization of a SoC using thermal and energy prediction schemes based on activity monitors, and a thermal-aware clock network featuring body-biasing
Type	Report
Ref	D3.4.3
Target version	
Current issue	
Status	released
File	D3.4.3_v2.0.docx
Author(s)	Romain LEMAIRE and Alexandre VALENTIAN (CEA-LETI) Andrea Calimera and Alberto Macii (POLITO)
Reviewer(s)	Domenik HELMS (CSEM), Michele Carrano (ST)
Approver(s)	Giuliana Gangemi (ST)
Approval date	20-08-2012
Release date	20-08-2012

Distribution of the release

Dissemination level	CO
Distribution list	

History

Rev.	DATE	Comment
0.0	25/06/2012	Draft
1.0	27/06/2012	Final version sent to internal reviewers
2.0	30/07/2012	Final version to be shipped to EU
2.0	20/08/2012	The document is released but still need a second internal reviewer feedback

This page was intentionally left blank.

Contents

Document.....	2
Distribution of the release.....	2
1 Introduction	5
2 Thermal and activity monitoring architecture.....	6
2.1 Thermal monitoring	6
2.2 Activity monitoring	8
3 Temperature estimation scenarios	9
3.1 Direct temperature measurement.....	9
3.2 Temperature estimation from energy dissipation.....	10
3.2.1 Thermal modeling	10
3.2.2 Dynamic thermal analysis.....	12
4 Layout Constrained Thermal Clock-Skew Compensation using Body-Biasing.....	15
4.1 Thermal Effects on Clock Distribution Networks	16
4.2 Adaptive Clock Tree.....	17
4.3 ILP Model	19
4.4 Experimental Results	20
4.4.1 Toolchain internals.....	20
4.4.2 Benchmarks and simulation results.....	21
5 Conclusions.....	23
6 References.....	24
7 Annex.....	25
7.1 Thermal sensor registers description	25
7.2 List of signal tracked by activity monitors.....	26

1 Introduction

This deliverable is organized into two parts: In the first part the implementation of a Multi-Processor System-on-Chip (MPSOC), called Genepy, is described. Genepy is fitted with temperature sensors and activity monitors. The temperature sensors will provide a reference temperature, while it will be possible to estimate the energy dissipation from the activity monitors. From those two sets of data, it will be possible to correlate the temperature and the energy consumption, and prove that in the end, temperature can be correctly estimated without using on-chip temperature sensors. Genepy testchip was implemented in Bulk 65nm, with a die area equal to 30mm². It was sent in foundry end of March 2012.

A first-order thermal model of Genepy (in its BGA package) was made. Simulations show that temperature is pretty uniform across the die: it is therefore mandatory that the various parts of the chip communicate with each other, so that each one is aware of the power dissipation of its neighbors.

In the second part, a row-based ILP formulation that provides optimal body bias assignment for thermal clock skew compensation is proposed. Although effective, the physical implementation of this technique is not trivial: While a fine tuning of the clock buffers would require to apply the body bias cell-by-cell, design constraints imposed by semi-custom layout rules impose higher granularities, i.e., clusters of cells. Experiments on a set of realistic benchmarks show that the proposed solution allows a real skew compensation under different thermal profiles, but avoids unfeasible conditions where buffers placed in the same row require different bulk polarizations.

2 Thermal and activity monitoring architecture

The GENEPI SoC is composed of 4 SMEP clusters interconnected by a Network-on-Chip (NoC). The SMEP cluster is based on 2 DSP MEPHISTO cores providing high-performance processing functionalities and a SME (Smart Memory Engine) in charge of data management both internally within the cluster and externally by communicating with other clusters. Finally the cluster is controlled by a MIPS processor.

In order to support advanced power management strategies, a set of sensors has been embedded into the SMEP cluster. Figure 1 presents the SMEP cluster architecture, highlighting the available sensors:

- Thermal sensor: it is sensitive to the temperature in the cluster,
- Activity monitors: they are composed of a set of counters reacting to various events in the SME part and in each MEPHISTO cores (0 and 1).

Both thermal sensors and activity monitors are considered as MIPS peripherals. It can control and access them directly from its data bus (memory-mapped).

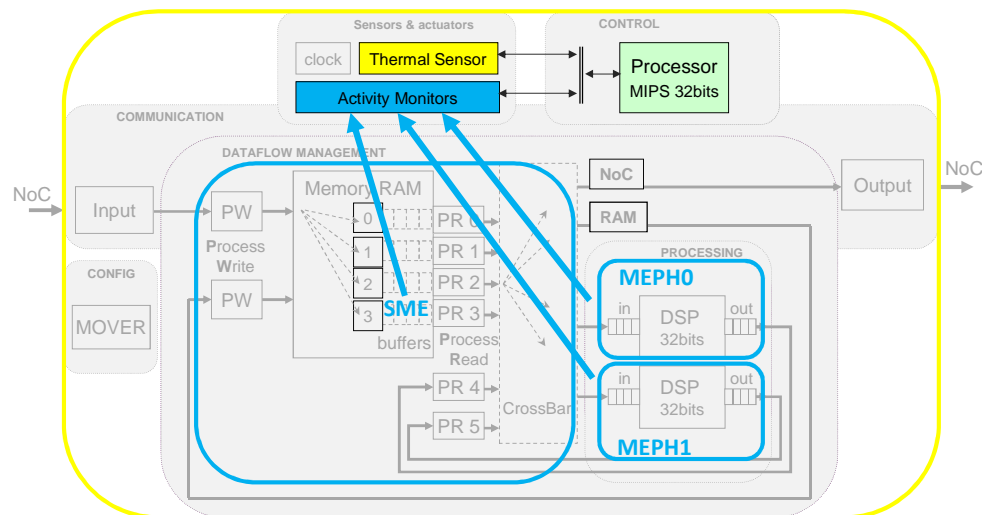


Figure 1: SMEP cluster architecture focused on thermal and activity monitoring

2.1 Thermal monitoring

The Thermal sensor can measure the temperature inside the SMEP cluster. The architecture of the thermal sensor subsystem (Figure 2) is made of two parts:

- A hard macro called MultiProbe which implements 2 ring oscillators, a counter and a set of control registers accessible through a daisy chain.
- A controller that can be configured through address-mapped registers.

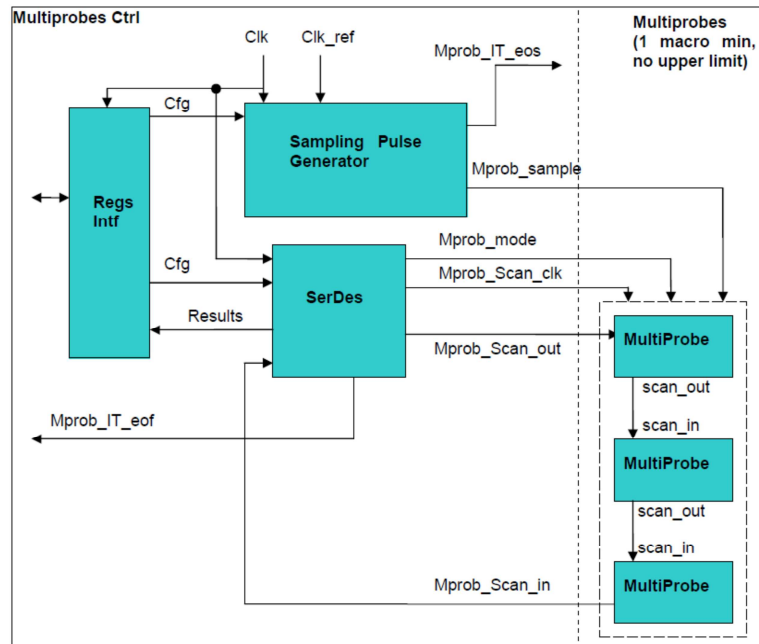


Figure 2: Thermal sensor schematic (only 1 MultiProbe in SMEP cluster)

In the MultiProbe, the counter is used to measure precisely the number of oscillations of a ring oscillator (RO) during a specific period of time. Two different ROs are implemented:

- RO1: made of standard inverter cells (used as a reference);
- RO2: made of custom inverters with increased temperature sensitivity (used in conjunction with RO1 to measure temperature).

The controller is composed of 2 subparts:

- A SER-DES to send/receive data to/from the MultiProbe macro that is accessible through a daisy chain;
- A programmable sampling pulse generator.

The thermal sensor is controlled by the MIPS processor through configuration and status registers. There are 2 64-bit registers as presented in Table 1. A more detailed description is available in Annex 7.1.

Table 1: Thermal sensor configuration and status registers

	63	32	31	0
REG0	serdes_ctrl			serdes_reg
REG1	control			sample_time

The 'serdes_reg' is set before shifting the SER-DES to configure the MultiProbe. Then, it is read after shifting to get the content of the counter.

In addition, the controller also drives two interruption lines directly connected to the MIPS status register:

- IT_eos: end of sampling, generated by the sampling pulse generator;
- IT_eof: end of frame, generated by the SER-DES.

2.2 Activity monitoring

The activity monitoring consists in instrumenting a set of control signals in some specific hardware blocks of the SMEP cluster: MEPHISTO cores and SME subsystem.

At design time, a correlation process between power consumption and signal activity has been developed (see Deliverable D3.4.2) to identify the most significant signals to be used as monitors. As a consequence, at execution time, it is possible to compute an accurate estimation of the power consumption of a block for a given period of monitoring.

For each considered signal, the monitor consists in a 16-bit counter which is incremented each time the signal is active (Polarity can be taken into account to define if the signal is active at low or high level). When a counter reaches the maximum value ($2^{16}-1$), it is automatically blocked (i.e. 0xFFFF value encodes an overflow).

The power estimation process is decomposed into 2 steps:

- A Monitoring step, when counters are activated,
- A Computing step, when the values from counters are collected to estimate the activity and consequently the power consumption, by applying a correlation function.

To control the monitoring step precisely, i.e. the counting duration, the MIPS activates a dedicated timer (32-bit). To activate a timer, the MIPS has to write a 'timer_limit' value. The timer will count clock cycles up to this value. When the value is reached, the timer is automatically disabled and an interruption is raised on the status register.

Monitors can be masked to control their activity and limit the power consumption of counters when they are not used. The monitor is active when mask is set to '1'.

Finally, the monitoring process is working as follows:

1. Set the mask to define which monitors will be used,
2. Start the 'timer_monitor' and wait for timeout,
3. When timeout is reached, the MIPS processor is interrupted and can access all the activity monitors in order to perform the power estimation.

The list of monitored signals is detailed in Annex 7.2

3 Temperature estimation scenarios

In the frame of the Therminator project, the hardware resources available in the SMEP cluster will be used in order to perform on-chip temperature estimation. Two scenarios will be investigated:

- Direct temperature measurement: this will be the reference temperature value based on the thermal sensor.
- Temperature estimation: this scenario will be based on the power estimation extracted from activity monitors associated with a power/temperature correlation modeling.

3.1 Direct temperature measurement

In this scenario, the MIPS core configures the thermal sensor. The sensor does not directly deliver a value expressed in degree but a number of oscillations that is function of the actual temperature where the ring device is positioned.

Depending on the operating condition (external reference clock frequency) and the precision of the measure expected, the MIPS may have to tune some parameter such as the duration of the measurement. In particular, a trade-off has to be found to avoid overflow in the counting process while the full 28-bit dynamic of the counter is exploited.

The MIPS will perform 2 measures on the ring oscillators: one on the reference RO1, the other one on the RO2. The time between the 2 measures is limited enough to consider the temperature as constant. Once the values from the counter are obtained, the MIPS will convert into a temperature value according to the conversion table based on the characteristics of the oscillators as illustrated on Figure 3. The temperature is directly a function of the frequency difference between the 2 ring oscillators.

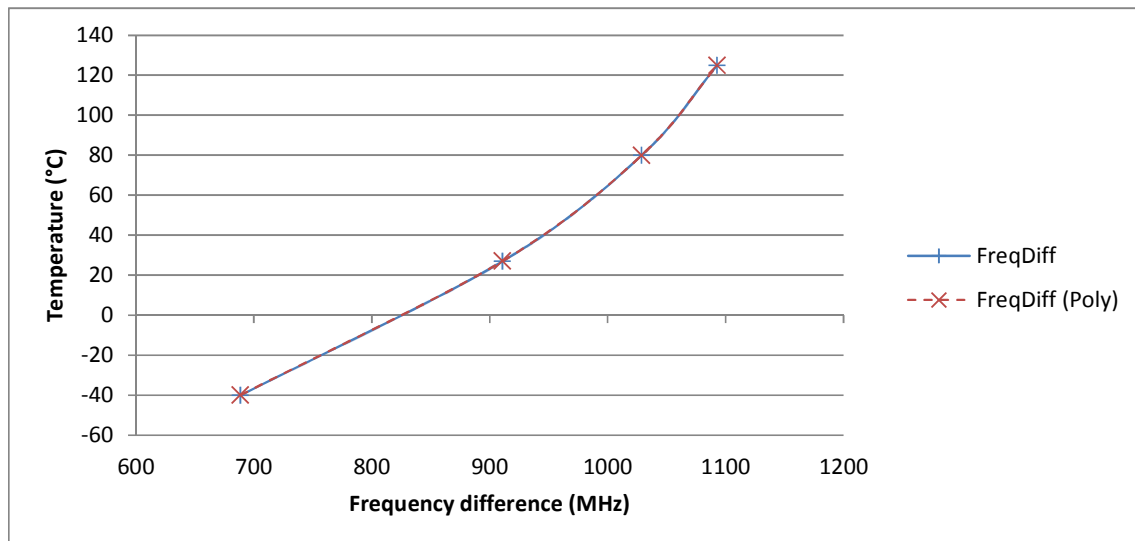


Figure 3: Temperature function of the frequency difference between ring oscillators

A polynomial function (1) has been extracted from SPICE simulations on the electrical model of the thermal sensor.

$$T = 2381 \frac{^{\circ}\text{C}}{\text{GHz}^3} \cdot \Delta f^3 - 5827 \frac{^{\circ}\text{C}}{\text{GHz}} \cdot \Delta f^2 + 5024 \frac{^{\circ}\text{C}}{\text{GHz}} \cdot \Delta f - 1514^{\circ}\text{C} \quad (1)$$

These characteristics will be refined by measurement and calibration on the actual chip back from manufacturing.

3.2 Temperature estimation from energy dissipation

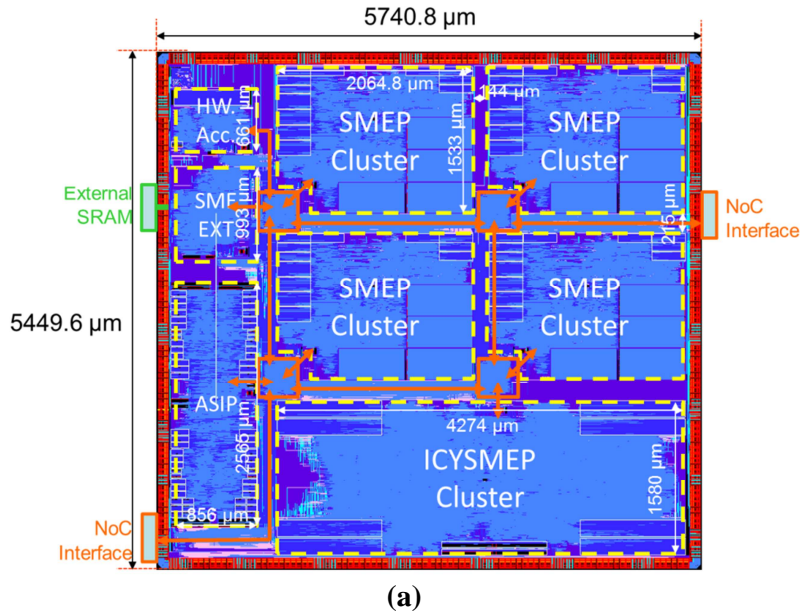
3.2.1 Thermal modeling

The temperature of an integrated circuit (IC) can be estimated from its power dissipation and the thermal properties of its layers, the package, the board and the thermal exchange with the air surrounding the package. A thermal model of an IC is derived by spatially discretizing its constituents into discrete thermal elements: each thermal element is defined by its thermal capacitance and thermal resistances to its nearest neighbors. For estimating the temperature from the power dissipation, the equation to be solved is:

$$C \frac{dT(t)}{dt} + AT(t) = Pu(t) \quad (2)$$

Where C is the thermal capacitance matrix ($[n \times n]$ diagonal matrix), A is the thermal conductivity matrix ($[n \times n]$ matrix), $T(t)$ and P are temperature and power vectors (of n -dimensional Euclidean space R^n) and $u(t)$ is the unit step function, *i.e.* the time step used by the MIPS to periodically read the activity monitors.

For avoiding a too high computational complexity and memory usage, it was decided to map one thermal element per unit of Genepy testchip. Those discrete thermal elements are shown in Figure 4-b and the corresponding floorplan in Figure 4-a.



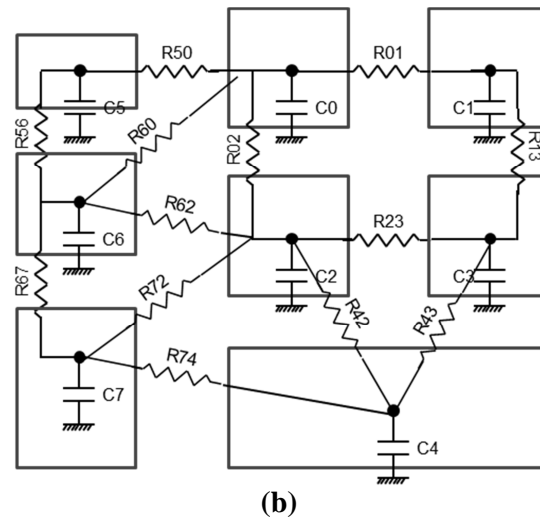


Figure 4 – (a) Floorplan of Genepy; (b) Discretization of Genepy into thermal elements.

Given the thermal properties of silicon, the thermal resistances and capacitances can be calculated in a straightforward way (the impact of the back-end was neglected in this first order model). The values are given in Table 2.

Table 2 – Values of thermal resistances and capacitances of the die layer.

Die layer	
Thermal conductivity Si =	149 W/m.K
Heat capacity Si =	19,8 J/mol.K
Thickness Si =	775 μm
Atomic weight Si =	28,0855
Density Si =	2,329 g/cm ³
Capacitance	
C0 = C1 = C2 = C3	0,00402786 J/K
C4 =	0,00859302 J/K
C5 =	0,000719995 J/K
C6 =	0,001081626 J/K
C7 =	0,002793929 J/K
Resistance	
R01 = R23 =	12,5 K/W
R02 = R13 =	7,3 K/W
R42 = R43 =	9,3 K/W
R50 =	21,7 K/W
R60 =	32,3 K/W
R62 =	33,7 K/W
R72 =	13,1 K/W
R74 =	19,7 K/W
R56 =	20,2 K/W
R67 =	40,8 K/W

A similar discretization was performed for the BT/Glass layer (shown in Figure 5), the Epoxy mold compound layer, the PCB layer, the package-to-board thermal resistance (through the balls) and the package-to-air and PCB-to-air thermal resistances.

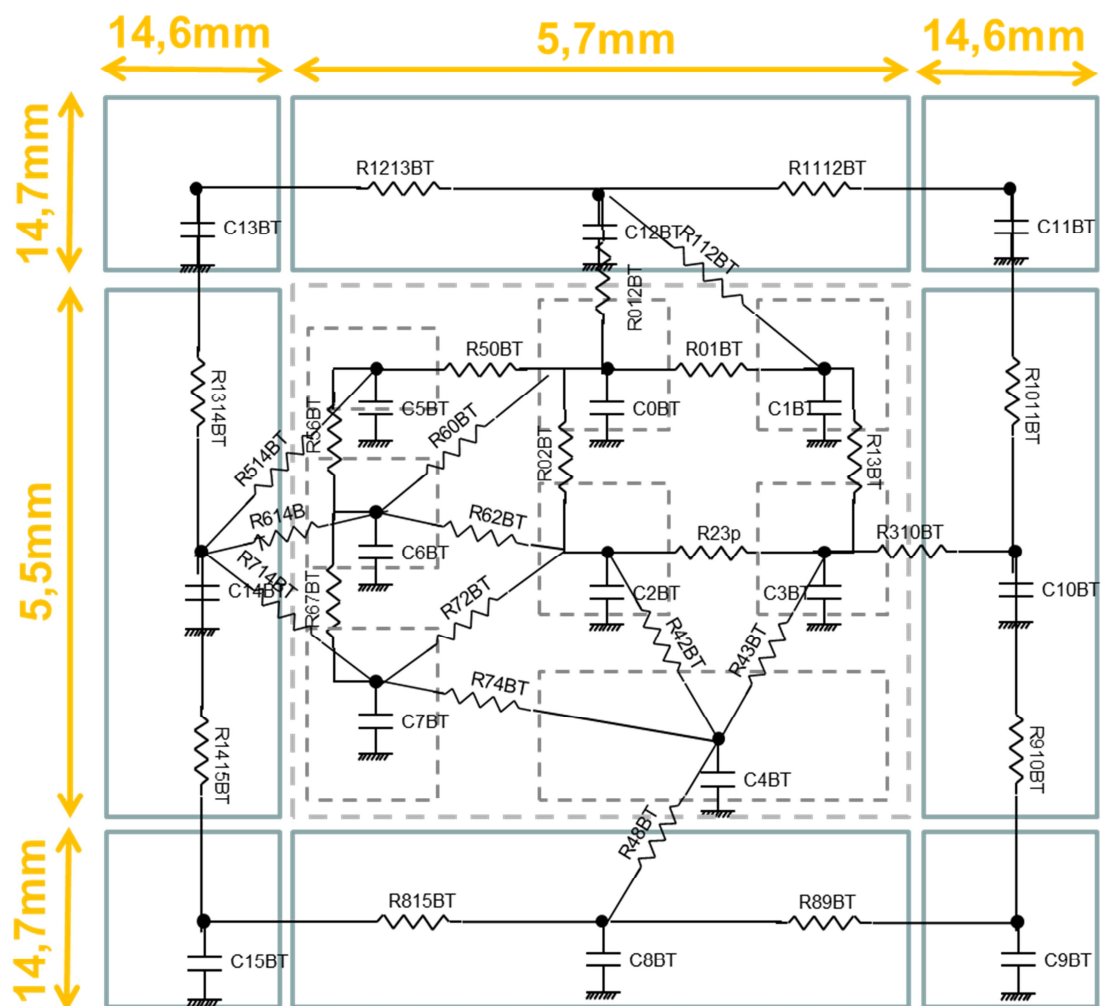


Figure 5 – Discretization of Genepy BGA package into thermal elements.

3.2.2 Dynamic thermal analysis

A Primepower simulation was performed on a SMEP unit, when it performs an FFT computation. The power profile is shown in Figure 6. Two activity phases can be distinguished: an initialisation phase, that lasts $91\mu\text{s}$, and the FFT computation phase which lasts $63\mu\text{s}$ and is repeated many times. During this second phase, the average power is 63mW . This power profile is then injected into the thermal model, in order to have a first order estimation of the die temperature.

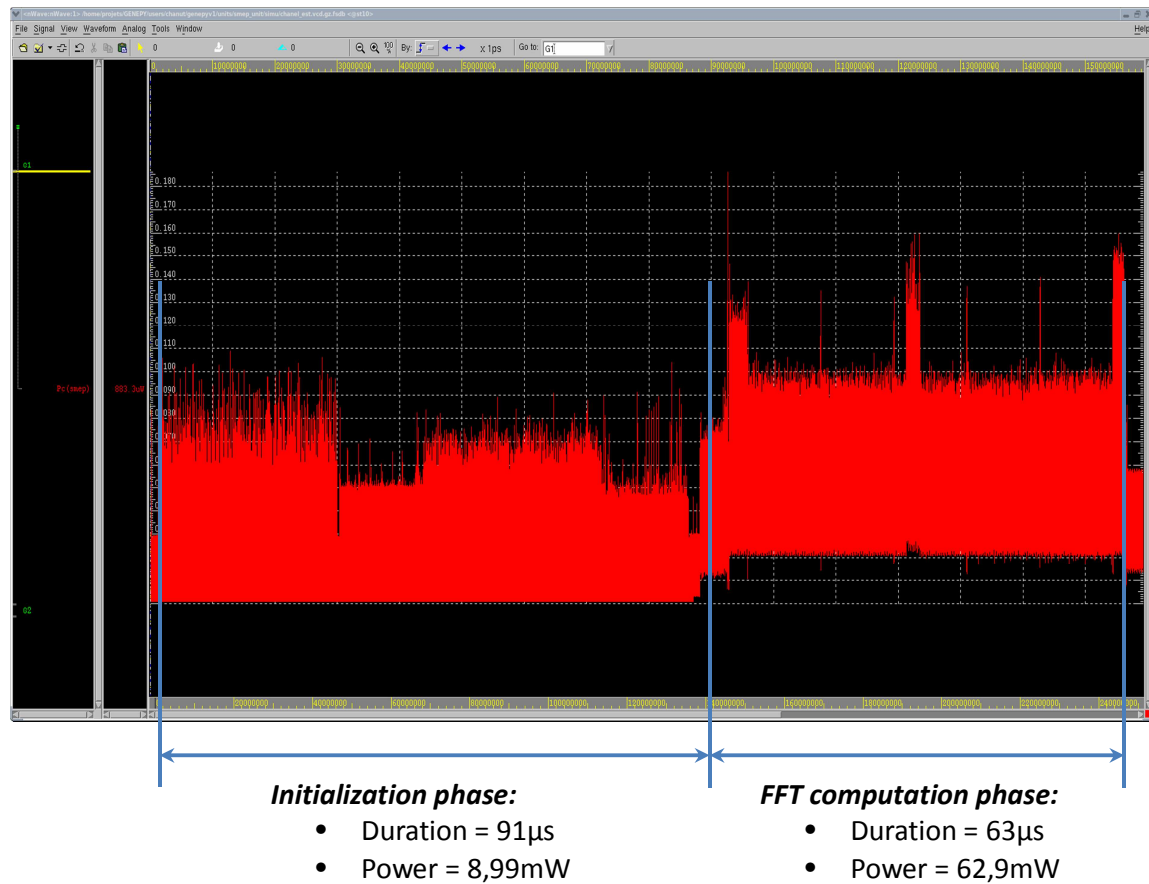


Figure 6 – Power profile of a SMEP unit when performing an FFT computation

In the above thermal models, a thermal resistance is analogous to an electrical resistance and a thermal capacitance is analogous to an electrical capacitance. Using such analogies, power dissipation corresponds to an electrical current and temperature corresponds to voltage. Thus, for characterizing the run-time thermal profile of Genepy, a simple electrical simulator can be used. The simulation results are shown in the figure below. The top set of curves shows the temperature of the 4 SMEP units when injecting a power of 63mW for 100s in SMEP0 unit. It can be seen that the temperature is almost the same for the 4 SMEPs, illustrating the fact that it is pretty uniform across the whole die. This is explained by the fact that the thermal resistance of the package is quite high, i.e. calculated close to 1000 K/W. The temperature of the die is therefore homogenized before heat can be evacuated.

As a consequence, each SMEP unit has to communicate its power dissipation to its neighbors, since this will have a direct impact on the temperature estimation of the other SMEP units. For correct temperature estimation, the power dissipation of all SMEPs has to be summed up at each time step.

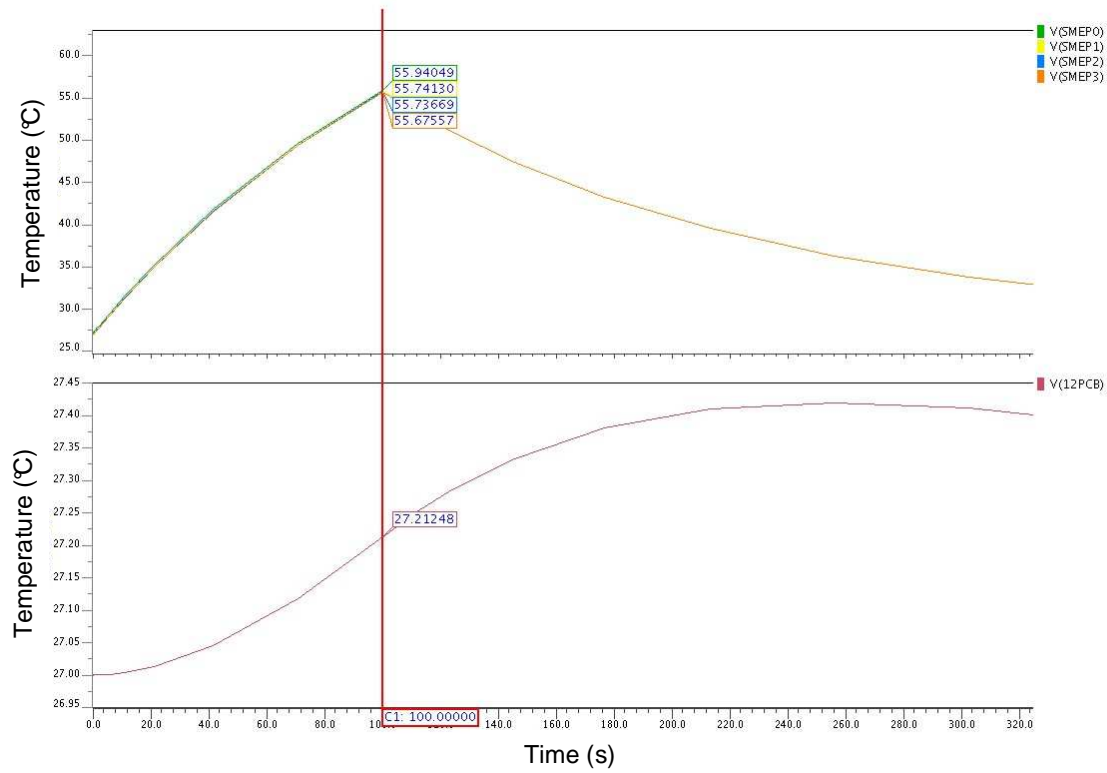


Figure 7 – Thermal profile of Genepy when performing an FFT computation on one of the SMEP units

4 Layout Constrained Thermal Clock-Skew Compensation using Body-Biasing

In the era of uncertain devices [3], the availability of design strategies and tools that consider variability as a direct variable in the design space has become of paramount importance. This is even truer when considering the design of special components that represent a concentrated source of failure, as the case of the Clock Distribution Networks (CDNs). For those components, even a minimal variation on the electrical parameters that govern their behavior can drastically impact the reliability of the entire circuit. In standard ASIC implementations, the CDN is in charge of distributing the clock signal to the sequential elements scattered across the layout. It consists of buffered global interconnects routed in the top metal layers and following a tree-like structure, hereafter the clock-tree. As the main responsible of the synchronization, a clock-tree must be designed to avoid race conditions between paths; more precisely, to enforce that clock edges reach registers approximately at the same time. This qualitative criterion is measured by the clock skew, that is, the difference between the longest and shortest clock tree paths, which must be as small as possible. Traditional clock tree synthesis algorithms, as those presented in [7, 9], allow the implementation of clock structures that minimize total wirelength and power consumption under zero or bounded skew. However, as in most of state-of-the-art commercial tools, they rely on the simple, yet wrong assumption that on-chip thermal profiles are flat and static in time. Unfortunately, the non-uniformity of the power densities, mainly due to the adoption of power-management strategies [15], can lead to uneven temperature distribution, i.e., thermal gradients. Obviously, devices working at different temperatures may show significant performance mismatch¹. The consequence is therefore the presence of branches of the clock-tree that show unbalanced delays; branches crossing hot regions get slower, while those crossing cold regions get faster. This difference have negative impact on global clock-skew [4], which, in the worst case, may cause setup/hold time violations. Clock tree optimization for thermal skew was first studied in [8], where the authors modified the traditional Deferred-Merge Embedding (DME) method proposed in [7] to search for nodes in the vicinity of merging points that can minimize clock skew for both uniform and non-uniform thermal profiles. However, while the above mentioned methods rely on design-time, i.e., static, optimizations, the need to manage time-dependent thermal gradients demanded for new design paradigms. To address this issue, the authors of [6, 10, 11] proposed a new post-silicon solution. The latter, which falls under the class of adaptive strategies, is based on the use of dynamically adjustable delay clock buffers that can be tuned to compensate at run-time the clock-skew. The main differentiation factor among the proposed adaptive techniques is the type of knob used to regulate the delay of the buffers. Among all the possible solutions [16], Adaptive Body Bias (ABB) has been proven to be a key alternative. It does not require any modification of the standard gates, and most technology libraries offers support for it. ABB exploits the body effect of MOS transistors allowing step-wise modulation of the buffer delay. Different embodiments of ABB have been presented by several authors in the recent years, e.g., [11, 17]; only few of them however considered the physical constrain imposed by row-based layout implementation of semi-custom ASICs [14]. Compensating clock skews with ABB requires the capability of selectively applying different bulk polarizations (Reverse or Forward Body Bias) to each individual clock buffer in the tree. As a matter of fact, the physical rules imposed by CMOS technologies below the 65nm do not allow applying BB to the cell granularity, as bulk contacts for each cell are no longer available. Instead, they provide ad-hoc contact cells that

¹ Typically, the higher the temperature, the larger the propagation delay. However this is not true for CMOS devices which show an Inverted Temperature Dependence [5]; not considered in this work.

are uniformly placed through the layout, in each row every 30 to 50 μm . That solution provides uniform and stable polarization for an entire bunch of cells with limited area overhead [14]. Needless to say, these rules constrain ABB to be applied with a larger granularity, thus making most of the proposed strategies unfeasible. In this work we propose a new ILP formulation that operates at the row level to find a proper clustering of cells that enables the application of ABB, thus allowing maximum thermal skew compensation while matching the aforementioned physical constraints. The optimal selection returned by the ILP can be used as reference to fill the LUT of a thermal management unit that will drive the tuning during in-field operations [6]. Experimental results conducted on a selection of large-scale benchmarks mapped into an industrial 40nm technology provided by ST and obtained from industry-strength physical design tools, show that the clock skew induced by realistic thermal profiles can be effectively reduced (6% on average) still maintaining compliance with physical rules.

4.1 Thermal Effects on Clock Distribution Networks

The clock skew is formally defined as the maximum difference between the (source to sink, i.e., register) arrival times $D_i - D_j$ of any pair of nodes (i, j) belonging to the set of sinks S of the tree (Equation 1).

$$\text{Skew} = \max \{D_i - D_j\}, \forall (i, j) \in S, \text{ with } i \neq j \quad (1)$$

The arrival time at each sink is the result of the additive delay contributions introduced by wire segments and buffers that break up the clock-path from the root. Both wires and buffers may vary their behavior significantly depending on the operating temperature. Concerning metal wires, a larger temperature gets a linear increase of the metal resistivity, as described by Equation 2:

$$R_0(x) = R_0(1 + \beta \cdot T(x)) \quad (2)$$

where, R_0 is the resistance at reference temperature (room temperature), β is the temperature coefficient that depends on the type of material (3.9×10^{-3} for Copper), while T represents the local temperature of the wire resulting from the diffusion of heat from the substrate and the self-heating effect [2]). Notice that, for long wires, as the case of clock trees, T may change depending on the actual position along the wire x .

Concerning MOS transistors, one of the main parameters affected by temperature is the carrier mobility μ , which, due to larger temperature induced lattice vibrations, decreases with increasing temperature. The relation governing such dependency is shown in Equation 3, where T is the junction temperature and T_0 is the nominal temperature (about 300K) and m is the temperature coefficient, which is about 1.5 but varies depending on the process.

$$\mu(T) = \mu(T_0) \cdot (T_0/T)^m \quad (3)$$

As described by the alpha power law model [13], the Equation 4, lower mobility reflects in a smaller current capability of the active transistors, which, in turn, affects the speed of the CMOS buffers, that get slower as temperature increases.

$$I_d \propto \mu(T) (V_{dd} - V_{th})^\alpha \quad (4)$$

Even if the clock runs on a symmetric network with both wires and buffers showing a monotonic delay increase due to temperature, the presence of thermal gradients may cause

substantial timing skew; branches of the tree that cross hot regions slow down due to buffers and interconnects, while branches that run over cold regions, show smaller propagation delays. Figure 8 depicts a scenario where two paths, Route-to-Sink1 and Route-to-Sink2, having same length, thus zero clock skew, can result in different delays due to the thermal effects.

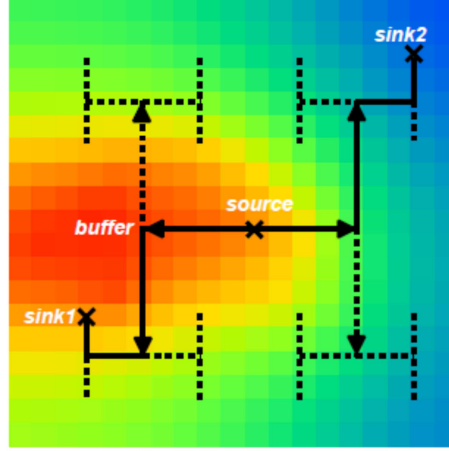


Figure 8 - Clock branches reaching different thermal regions develop unequal delays.

4.2 Adaptive Clock Tree

Recent works [6, 10, 11] have proposed the use of adaptive strategies to compensate, dynamically, the effects of temperature on the clock skew. All of them are based on the same idea, the use of tunable clock buffers through which compensate delay variation along the clock paths. However they substantially differ in terms of i) control knob used to adjust the buffers delay, ii) granularity at which the control knob operates on the buffers, iii) implementation of the control structure. Here we use Adaptive Body Bias (ABB) as main control knob and we introduce a layout constrained row-based scheme where ABB is applied at the row-level, namely, each individual row of the layout has its own independent polarization voltage that is controlled dynamically depending on the thermal profile of the die.

Control Knob: ABB can alter the threshold voltage of the device exploiting the body effect of MOS transistors. The body effect is modeled by Equation 5, which describes the dependency of the threshold voltage V_{th} from the bulk voltage V_b :

$$V_{th} = V_{T0} + \gamma(\sqrt{|V_{sb} + 2\Phi_f|} - \sqrt{2\Phi_f}) \quad (5)$$

with V_{T0} the threshold voltage under zero substrate bias, γ is the body effect parameter, V_{sb} is the voltage between the source and bulk terminals, and Φ_f is the surface potential. A Forward Body Bias FBB, i.e., $V_{bs} > 0$ ($V_{bn} = +V_{FBB}$ for nMOS and $V_{bp} = V_{dd} - V_{FBB}$ with $V_{FBB} > 0$), decreases the V_{th} of the transistor that can drain more current and make the buffers faster. As side effect, FBB exponentially increases the sub-threshold leakage power [12]. On the contrary, a Reverse Body Bias RBB, i.e., $V_{bs} < 0$ ($V_{bn} = -V_{RBB}$ for nMOS and $V_{bp} = V_{dd} + V_{RBB}$ for pMOS with $V_{RBB} > 0$), increases the V_{th} getting slower buffers. Positive effect of RBB is a lower leakage current. The selection of V_{FBB} and V_{RBB} is a technology dependent process.

Depending on the internal characteristics of the transistor, like doping concentration of the source/drain-bulk pn junctions, RBB and FBB polarizations can be defined as to guarantee the best trade off between delay variation, power and reliability. For the adopted technology, a low-power 40nm technology with nominal V_{dd} of 1.1V, we used symmetric values, i.e., $V_{FBB} = V_{RBB} = 0.5V$.

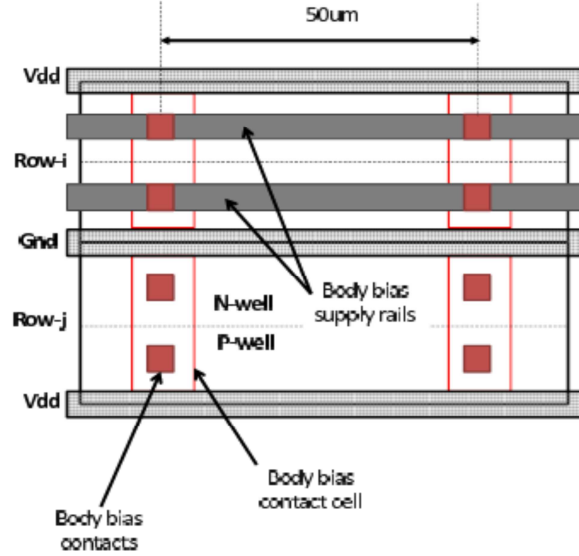


Figure 9 - Abstract view of row-based ABB layout [14]

Granularity: From a conceptual point of view, ABB should be applied, separately and independently to each individual buffer of the clock tree. Unfortunately, physical design rules of modern CMOS technologies do not support cell-level body biasing, as the bulk contacts are no longer available in the cells. On the contrary, silicon vendors provide special cells, the body bias contact cells, which are placed through layout rows and driven by the body bias generator. The design rules require the body bias contact cells to be placed every $50\mu m$ (for technology we have used). Therefore, the minimum grain at which ABB can be applied is a portion of the row, a sub-row. However, since multiple ABB per row seriously complicates the routing of the body bias supply rails, in this work we consider the row as atomic unit, i.e., each row can be polarized with a dedicated voltage (V_{bp} and V_{bn} , as shown in Figure 9).

Architecture: Concerning the control structure, we borrowed the idea proposed in [6], where an embedded hardware mechanism, called the thermal management unit (TMU), is used to translate on chip thermal profiles (provided by dedicated temperature sensors) into the proper bias configuration that minimize the skew. Figure 3 illustrates the proposed architecture. Data collected from the sensors are used to point a specific row in the TMU table; the one that contains the body-bias configurations for compensating the skew generated by the current thermal profile. Conceptually, rows containing buffers that belong to slow paths are Forward Body Biased (FBB), while, rows that contain buffers of fast paths are Reversed Body Biased (RBB). The optimum body bias configuration, which is stored in the TMU table, is carried out by means of an off-line characterization step, that is, for each applied thermal profile, the optimal body bias selection (RBB or FBB) of each row is obtained by solving an ad-hoc ILP model (described in the next section).

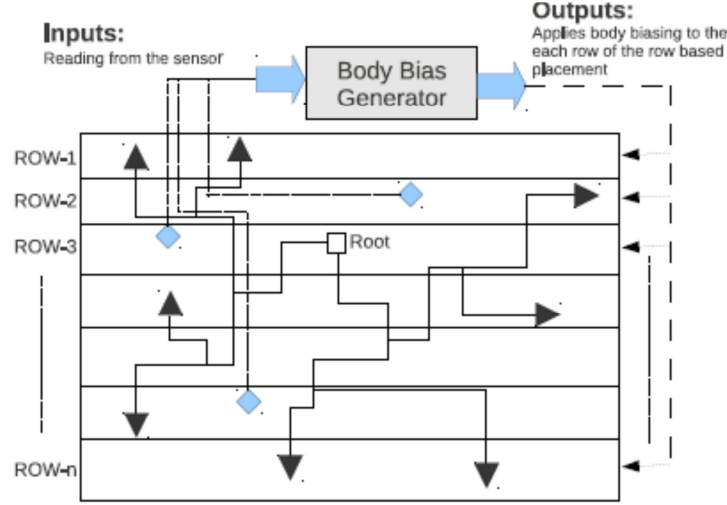


Figure 10 - Revisited TMU architecture [6] for skew compensation using row-based ABB

4.3 ILP Model

In this section we introduce the mathematical formulation through an ILP of the optimization problem we are trying to solve. Specifically, we want to find an optimal row-based ABB configuration that compensates the clock skew for a given thermal map and matching the physical constraints. Let assume that the gates in the design have been placed among N different rows. We assign two 0-1 integer variables to each row i , x_i and y_i , whose meaning is described in Equation 6:

$$\begin{aligned} x_i &= \begin{cases} 0, & \text{zero bias} \\ 1, & \text{if FBB} \end{cases} \\ y_i &= \begin{cases} 0, & \text{zero bias} \\ 1, & \text{if RBB} \end{cases} \end{aligned} \quad (6)$$

The first main constraint concerns the clock skew, which has to be maintained below a user defined bound B_{skew} . This is described in Equation 7, where the difference between the propagation delay D of any pair of paths p_n and p_k belonging to the set of root-to-sink paths Π is forced to be lesser than B_{skew} .

$$0 \leq D_{\pi_k} - D_{\pi_n} \leq B_{skew} \quad \forall \pi_k, \pi_n \in \Pi / k \neq n \quad (7)$$

The delay D_{π_i} of the generic path π is given by the sum of the propagation delays of the clock buffers and wire segments in the path itself:

$$D_{\pi_i} = \sum_{j=0}^l d_j + d_{net,j} \quad \forall j \in \pi \quad (8)$$

Where $d_{net,j}$ is the delay introduced by wire load in the fanout of the buffer, while d_j is the propagation delay of each buffer. As described in the following Equation, each d_j can be further expanded as a function of the body bias voltage applied to the row i that hosts buffer j :

$$D_j = d_0 - x_i \delta_{j,FBB} + y_i \delta_{j,RBB} \quad (9)$$

where d_0 is the nominal propagation delay of the cell under zero biasing, $\delta_{j,FBB}$ represents the delay variation of the buffer in case of FBB, and $\delta_{j,RBB}$ represents the delta delay in case of RBB. To calculate d_j , we make use of LUTs containing pre-characterized delay under different loads, transition time and operating temperatures. Depending on the biasing configuration associated to each row i , namely, the pair (x_i, y_i) , the clock skew can be compensated. Obviously, each row needs a unique bulk polarization, FBB or RBB; this condition is forced by the following constraints:

$$x_i + y_i \leq 1 \quad \forall i \in N \quad (10)$$

The resulting ILP model is then reported in Equation 11, where the cost function to minimize, the Equation 11 represents the total number of rows polarized at a voltage other than zero. Minimizing such metric we reduce the impact of ABB to the other gates in the circuit.

$$\min: \sum_i^N (x_i + y_i) \quad (11)$$

4.4 Experimental Results

4.4.1 Toolchain internals

The design framework we implemented to validate the proposed methodology relies on different commercial tools like Synopsys Design Compiler (DC), used in the logic synthesis phase, Synopsys IC Compiler (ICC) used for the physical design (placement and clock tree routing), Synopsys HSPICE used for transistor level simulations. ILP problems, instead, have been solved using the open source tool Mixed Integer Programming Solver *lp_solve* [1]. The entire flow consists of three main stages:

1. **Synthesis:** Starting from a RTL description, the design goes through standard logic/physical synthesis, which yields a gate-level netlist containing both the logic gates and the clock tree. Once those steps are accomplished an in-house tool extracts a transistor-level netlist of the clock network which includes extracted view of the buffers (taken from an industrial 40nm technology library) and parasitics back-annotated from the SPEF file².
2. **ABB assignment:** The SPICE netlist is simulated along with thermal map (assumed to be given) via HSPICE, in order to obtain path delays and skew information used for the construction of the ILP, as described in Section 4.3. The solver is then fed with the ILP model and run. The returned solution describes the ABB configuration to be applied to each row.
3. **Validation:** The ABB configuration returned from the ILP solver is finally applied to the original netlist and statically analyzed within ICC. For this stage we make use of timing libraries characterized under different bulk polarizations: FBB ($V_{bn} = 0.5V$ and $V_{bp} = 0.5V$), RBB ($V_{bn} = -0.5V$ and $V_{bp} = 1.5V$). Synopsys NCX has been used to collect the timing information.

² SPEF: Standard Parasitic Exchange Format.

4.4.2 Benchmarks and simulation results

Table 3 reports the main characteristics of the three circuits used as benchmarks.

Table 3: Synthesized benchmarks

Design	# Layout Rows	# Buffers	Clock Skew (ps)
Benchmark 1	66	35	72.87
Benchmark 2	177	138	197.52
Benchmark 3	305	182	224.74

To emulate different thermal behaviors, the chip die is divided in eight equally sized layout regions, clockwise labeled from one to eight, starting from the top left. We therefore assume the availability of a dedicated temperature sensor for each region. Table 4 shows the four thermal profiles we considered during simulations.

Table 4: Chip die thermal region temperatures (in °C)

	Reg #1	Reg #2	Reg #3	Reg #4	Reg #5	Reg #6	Reg #7	Reg #8
Map #1	100	75	25	75	100	25	100	50
Map #2	50	100	75	25	25	50	25	100
Map #3	75	50	75	50	100	25	50	25
Map #4	25	100	50	25	75	50	25	100

Table 5 reports, for each benchmark, the amount of clock skew compensation (ΔSkew) we can achieve under different thermal maps. The comparison is made between the physical constrained ILP model presented in this work (column Row-Based) and a standard ILP model (column Cell-Based) that does not take into account any physical constraints, that is, each buffer can be assigned, separately, to any body bias. Such cell-based formulation is very similar to the row-based one, but boolean variables representing ABB configuration, i.e., x_i and y_i (please refer to Section 4.3), are referred to each single buffer in the clock tree rather than rows.

As expected, the cell-based ILP, which has more degrees of freedom, shows a better skew compensation, 7.55% on average, w.r.t. the row-based ILP, 5.18% on average. Nevertheless, the cell-based approach returns a non-feasible solution for most of the cases under test. The column Feasibility reports the number of rows that produce frustrated constraints, namely, rows containing buffers that should be assigned to a certain bulk polarization, but cannot because of the presence of other buffers polarized at different body bias. Only for one case, the cell-based ILP returns a feasible solution, the Benchmark 2 under Map #4, for which 0 rows show constraint violations. On the contrary, the row-based approach is always feasible, as each row is assigned to a single body bias.

One additional comment concerns the potential overheads introduced by the row-based body biasing. Changing the bulk polarization of a row affects not just the clock buffers, but also the other cells placed in that row. This may alter other metrics of the circuit, as the leakage power (FBB reduces the V_{th} increasing the sub-threshold current) and the maximum operating frequency (RBB increases the V_{th} making logic gates slower). However the proposed ILP has been constructed in order to minimize such effects. This has been achieved by minimizing the number rows assigned to any FBB or RBB (maximum number of rows left to zero bias). Experimental results have shown that leakage overhead is always below 5%, for any benchmark and for any thermal profile; larger delay penalties, instead, have been registered, 10% on average. The reason is that the ILP does not consider that RBB on cells belonging to

the critical paths directly affect the overall speed of the circuit. We are working on new formulations and heuristics that consider timing penalties during optimization.

Table 5: Clock skew compensation.

			Cell-Based		Row-Based	
	# Rows	Thermal Map	Δ Skew (%)	Feasibility	Δ Skew (%)	(FBB, RBB)
Benchmark 1	66	Map #1	8.84	14	6.39	(17, 6)
		Map #2	7.68	6	4.36	(13, 4)
		Map #3	7.72	6	5.33	(12, 4)
		Map #4	8.67	12	6.82	(16, 7)
Benchmark 2	177	Map #1	11.74	2	7.98	(2, 2)
		Map #2	11.54	2	4.28	(4, 1)
		Map #3	8.63	2	7.41	(5, 1)
		Map #4	5.54	0	4.34	(4, 1)
Benchmark 3	305	Map #1	5.70	26	4.28	(41, 6)
		Map #2	3.59	28	2.27	(32, 7)
		Map #3	4.69	28	3.75	(28, 4)
		Map #4	6.25	31	4.97	(39, 7)
Avg.			7.55		5.18	

5 Conclusions

In the first part of this deliverable, an MPSOC circuit, called Genepy, was implemented in Bulk 65nm with structures for monitoring its activity and temperature. This will enable to correlate temperature and energy dissipation, thus enabling to do thermal management in a more proactive way than just waiting for temperature to reach a given threshold.

A first-order thermal model of Genepy, encased in its BGA package, was developed. The simulation based on this model shows that the temperature is almost uniform across the die. As a consequence, each and every processor constituting an MPSOC circuit must communicate its power dissipation to all its neighbors, for a correct temperature estimation. The measurement results of Genepy testchip will be performed in the autumn 2012 timeframe and detailed in Deliverable D7.2.1.

In the second part of this document, the basic idea behind the construction of the ILP model and the implementation of an in-house framework for dynamically compensating the clock skew variation due to thermal gradients have been illustrated. It turned out that the proposed ILP-based approach is an effective option to recover clock skew degradation. Results show that compensation of clock skew based on physical placement information is possible; however it will come at expense of worse slack timing among the circuit's critical paths.

6 References

- [1] Introduction to *lp_solve* (2012), <http://lpsolve.sourceforge.net/5.5/>
- [2] Ajami, A., Banerjee, K., Pedram, M.: Modeling and analysis of nonuniform substrate temperature effects on global ulsi interconnects. *Computer-Aided Design of Integrated Circuits and Systems*, IEEE Transactions on 24(6), 849 – 861 (June 2005)
- [3] Borkar, S.: Designing reliable systems from unreliable components: the challenges of transistor variability and degradation. *Micro*, IEEE 25(6), 10–16 (nov 2005)
- [4] Bota, S., Rossello, J., de Benito, C., Keshavarzi, A., Segura, J.: Impact of thermal gradients on clock skew and testing. *Design Test of Computers*, IEEE 23(5), 414–424 (May 2006)
- [5] Calimera, A., Bahar, R., Macii, E., Poncino, M.: Temperature-insensitive dual synthesis for nanometer cmos technologies under inverse temperature dependence. *Very Large Scale Integration (VLSI) Systems*, IEEE Transactions on 18(11), 1608–1620 (Nov. 2010)
- [6] Chakraborty, A., Duraisami, K., Sathanur, A., Sithambaram, P., Benini, L., Macii, A., Macii, E., Poncino, M.: Dynamic thermal clock skew compensation using tunable delay buffers. *Very Large Scale Integration (VLSI) Systems*, IEEE Transactions on 16(6), 639 –649 (June 2008)
- [7] Chao, T.H., Hsu, Y.C., Ho, J.M., Kahng, A.: Zero skew clock routing with minimum wirelength. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 39(11), 799 –814 (Nov 1992)
- [8] Cho, M., Ahmedtt, S., Pan, D.: Taco: temperature aware clock-tree optimization. In: *Proc. of International Conference on Computer-Aided Design*. pp. 582–587 (Nov. 2005)
- [9] Cong, J., Koh, C.K.: Minimum-cost bounded-skew clock routing. In: *Proc. of International Symposium on Circuits and Systems*. vol. 1, pp. 215 –218 (Apr. 1995)
- [10] Long, J., Ku, J.C., Memik, S., Ismail, Y.: Sacta: A self-adjusting clock tree architecture for adapting to thermal-induced delay variation. *Very Large Scale Integration (VLSI) Systems*, IEEE Transactions on 18(9), 1323–1336 (Sept. 2010)
- [11] Ragheb, T., Ricketts, A., Mondal, M., Kirolos, S., Links, G., Narayanan, V., Massoud, Y.: Design of thermally robust clock trees using dynamically adaptive clock buffers. *Circuits and Systems I: Regular Papers*, IEEE Transactions on 56(2), 374–383 (Feb. 2009)
- [12] Roy, K., Mukhopadhyay, S., Mahmoodi-Meimand, H.: Leakage current mechanisms and leakage reduction techniques in deep-submicrometer cmos circuits. *Proceedings of the IEEE* 91(2), 305 – 327 (Feb. 2003)
- [13] Sakurai, T., Newton, A.: Alpha-power law modfet model and its applications to cmos inverter delay and other formulas. *Journal of Solid State Circuits* 25(2), 584–594 (Apr. 1990)
- [14] Sathanur, A., Pullini, A., Benini, L., Micheli, G.D., Macii, E.: Modeling of thermally induced skew variations in clock distribution network. In: *DATE'09: Proceedings of the conf. on Design, Automation and Test in Europe*. pp. 162– 167 (Mar. 2009)
- [15] Skadron, K., Stan, M.R., Sankaranarayanan, K., Huang, W., Velusamy, S., Tarjan, D.: Temperature-aware microarchitecture: Modeling and implementation. *ACM Trans. Archit. Code Optim.* 1(1), 94–125 (Mar. 2004)
- [16] Tawfik, S., Kursun, V.: Dual supply voltages and dual clock frequencies for lower clock power and suppressed temperature-gradient-induced clock skew. *Very Large Scale Integration (VLSI) Systems*, IEEE Transactions on 18(3), 347 –355 (Mar. 2010)
- [17] Wolpert, D., Ampadu, P.: Exploiting programmable temperature compensation devices to manage temperature-induced delay uncertainty. *Circuits and Systems I: Regular Papers*, IEEE Transactions on 59(4), 735 –748 (Apr. 2012)

7 Annex

7.1 Thermal sensor registers description

The following tables present details on the content of the thermal sensor registers.

Table 6: Thermal sensor: serdes_reg register

31	30	29	28	27	26	...	1	0
config			overflow		counter			

TYPE: R/W RESET: 0x0000 0000

config: Select the ring oscillator to be used.
In the SMEP cluster, only values 0x1 or 0x2 are accepted (for respectively RO1 and RO2).

overflow: Set to '1' when overflow of counter happens.

counter: ring oscillator counter.

Table 7: Thermal sensor: serdes_ctrl register

31	16	15	14	13	12	11	10	9	8	7	6	5	3	2	1	0
									status					delay		

TYPE: R/W RESET: 0x0000 0000

status: (read only) Set to '1' while a cycle of shifting serdes_reg is on-going.

delay: slow down the SER-DES interface by a factor of 2^X.

Table 8: Thermal sensor: sample_time register

31	30	29	28	27	26	25	24	23	0
								duration	

TYPE: R/W RESET: 0x0000 0000

duration: number of clk_ref clock cycles for sampling.

Table 9: Thermal sensor: mprob_control register

31	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			test							reset							force	start

TYPE: R/W RESET 0x0000 0000

start: Set to '1' to start sampling, automatically return to '0' when sampling duration is reached.

force: Set to '1' to force sampling

reset: Set to '1' to release the reset of the MultiProbe macro

test: Allow to manually increase the counter in the MultiProbe

7.2 List of signal tracked by activity monitors

The following tables give the list of the signals that has been instrumented by activity monitors in the SMEP cluster.

Table 10: MEPHISTO activity monitors list

Index	Sub-block	Function
0	PC	Start of a loop
1		Execution of the sequencer
2		Read a word to load in DP or PC register
3		Access to firmware RAM
4	DP	Write access in register bank
5		Read access in registers bank
6		Access to RAM 0 bank
7		Access to RAM 1 bank
8		Start operation on MAC1
9		Start operation on MAC1
10		Complex multiplication on MAC 0
11		Complex multiplication on MAC 1

Table 11: SME activity monitors list

Index	Sub-block	Function
0	PR	Write output data FIFO 0
1		Write output data FIFO 1
2		Write output data FIFO 2
3		Write output data FIFO 3
4		Write output data FIFO 4
5		Write output data FIFO 5
6	RAM	Access to RAM 0 bank
7		Access to RAM 1 bank
8		Access to RAM 2 bank
9		Access to RAM 3 bank
10	PW	Read input data FIFO0
11		Read input data FIFO 1

12		Read input data FIFO2
13	PR	Instruction execution in PR0
14		Instruction execution in PR1
15		Instruction execution in PR2
16		Instruction execution in PR3
17		Instruction execution in PR4
18		Instruction execution in PR5