

Temperature dependent timing in standard cell designs

Andras Timar^{a,*}, Marta Rencz^{a,1}

^a*Budapest University of Technology and Economics, Department of Electron Devices,
Magyar tudósok körútja 2, Building Q, section B, 3rd floor, Budapest, Hungary H-1117*

Abstract

This paper proposes a methodology to simulate temperature dependent timing in standard cell designs. Temperature dependent timing characteristics are derived from standard delay format (SDF) files that are created by synthesis tools automatically based on SPICE characterizations. In addition, a fast calculation of temperatures using the equivalent Foster RC network is presented. A case study is also presented in this paper where the temperature dependent frequency variation of a ring oscillator is simulated demonstrating the necessity of temperature dependent timing simulations. An adaptively refineable partitioning method for simulating standard cell designs logi-thermally is proposed as well. This paper also introduces recent enhancements in the CellTherm logi-thermal simulator developed in the Department of Electron Devices, BME, Hungary. Finally, the simulation results are compared and verified with the SPICE compatible ELDO analog simulator from Mentor Graphics.

Keywords: temperature dependent delays, grid, logi-thermal, electro-thermal, simulation

1. Introduction and motivation

In today's digital integrated circuits dealing with heating issues became a primary concern. As technology nodes and minimum feature size (MFS) keep shrinking device parameters tend to become more sensitive to process

*Corresponding author. Telephone: +36-1-463-2702, Fax: +36-1-463-2973
Email addresses: timar@eet.bme.hu (Andras Timar), rencz@eet.bme.hu (Marta Rencz)

¹Member, IEEE

variations, supply voltage fluctuations and device temperature, to mention a few. These phenomena inferred the need for a simulation tool which is aware of process, temperature and supply voltage variations and can detect issues caused by such effects. A simulation framework is therefore needed to detect such problems and to allow for taking actions during the design phase, far before manufacture.

In this paper we demonstrate a methodology to constantly monitor device temperature during logic simulation and change cell propagation delays according to the current device temperature. This way we can predict operation of the circuit in a real environment encumbered with self- and ambient heating. Standard cell propagation delays are characterized by SPICE simulations and synthesis tools derive actual delay-temperature functions from cell placement and routing. The resulting sampled delay-temperature functions are stored in a Standard Delay Format (SDF) file.

The paper also shows a fast methodology to calculate cell temperatures using the analogy between the electrical and thermal domain. The methodology shows an easily implementable algorithm that calculates device temperatures using dissipated powers and the equivalent thermal Foster RC network of the structure.

2. Related work

In [1] Pable et al. deals with ultra-low-power signaling challenges caused by process, voltage and temperature (PVT) variations. Exponential dependency of subthreshold drive current on V_{th} and temperature in subthreshold operating region makes process and temperature variations of great interest while designing robust ULP systems. Small variation in the device V_{th} will translate into exponential variation in bias current and hence the device delay and power dissipation. The subthreshold phenomena addressed in the paper are potential sources of failures in a digital circuit. These issues must be addressed by a suitable simulator. Small changes in device V_{th} due to temperature variations can lead to timing problems that must be prepared for.

Rebaud et al. in [2] describe a new monitoring system, allowing failure anticipation in real-time, looking at the timing slack of a pre-defined set of observable flip-flops. They propose adaptive voltage scaling (AVS) and adaptive body biasing (ABB) to compensate PVT variations. Authors of [2]

implemented a monitoring system on silicon to measure and adaptively compensate process variations. The dedicated sensor structures allow for real-time compensation of the mentioned effects. With a PVT variation-aware simulator engine the need for these extra sensor, monitor and compensation circuits might be eliminated as construction could be done in a PVT variation-aware manner.

Lin et al. in [3] introduce a novel 9 transistor SRAM cell where PVT variations are taken into account. The proposed CMOS SRAM cell is PVT tolerant. The authors simulated their proposed 9 transistor SRAM cell with HSPICE and demonstrated excellent process variation tolerance. The PVT tolerance in this circuit is achieved by exploiting circuit techniques.

Kumar and Kursun in [4] attract attention to the fact that temperature-dependent propagation delay characteristics of CMOS integrated circuits will experience a complete reversal in the near future. They demonstrate that the speed of circuits in a 45-nm CMOS technology is enhanced when the temperature is increased at the nominal supply voltage. This is a quite interesting phenomenon contrary to the older technology generations. The inversion of delay-temperature functions in standard cell designs is a very important effect when simulating circuits with temperature-aware logic simulators. The reverse temperature effect described in [4] can be taken into account with the simulator proposed in this paper.

Authors of [5] introduce a sensor circuit with an area of 985 NAND2 equivalent gates that is capable of detecting whether a system is operating in the normal dependence or the reverse dependence region. This area overhead can be very significant especially for smaller designs. Digital standard cell circuits susceptible to temperature variations might be simulated with temperature-aware logic simulators and therefore such extra sensor circuits might be spared or their complexity might be decreased.

In [6] Sánchez-Azqueta et al. introduce a CMOS ring VCO design where PVT variations were taken into account. To overcome the limitation of the PVT variations, a tuning range of about 20% is sufficient for their ring VCO. The VCO mentioned in [6] contains a 4-stage ring oscillator. Compensating for process, temperature and voltage changes is also a main concern in VCO design and the presented demonstration circuit in this paper also utilizes a ring oscillator circuit to demonstrate frequency shifts caused by self-heating.

In [7] the leakage current, active power and delay characterizations of the dynamic dual V_t CMOS circuits in the presence of process, voltage, and temperature (PVT) fluctuations are analyzed based on multiple parameter

Monte Carlo method.

In [8] Winther et al. show that using wirelength as the evaluation metric for floorplanning does not always produce a floorplan with the shortest delay. They propose a temperature dependent wire delay estimation method for thermal aware floorplanning algorithms, which takes into account the thermal effect on wire delay.

[9] presents the temperature influence on energy consumption and propagation time delay in CMOS ASIC circuits with several measurements.

Reviewing the present literature it turns out that there is a need for a simulator that can take process, voltage and temperature variations into account and especially self-heating of circuits during operation. This paper introduces such a simulator tool called CellTherm developed in the Department of Electron Devices, BUTE, Hungary.

In this paper the most recent improvements of the CellTherm [10, 11] simulator engine is introduced. The CellTherm logi-thermal simulator is capable of simulating standard cell integrated circuit designs given with Verilog structural description. The simulator couples a standard compliant logic simulator (e.g. QuestaSim[©], Incisive[©]) and thermal solver engine and calculates device temperatures in function of simulation time. CellTherm also reads the full layout, power and timing data of the design and calculates temperature-dependent delays of the consisting standard cells. This way CellTherm not only can detect hot-spots in the design but can back-annotate device timing and propagation delays during the simulation. CellTherm also can create heating animations of the design and watch out for thermally induced timing violations with the help of the logic simulator.

3. Design for demonstration

Our case study was a $0.516\text{mm} \times 0.353\text{mm}$ standard cell digital circuit with four buffer cells driven with a 4-bit digital counter stimuli and a ring oscillator circuit. The process node was TSMC $0.35\ \mu\text{m}$. The design has been created from a synthesized Verilog description. For automated placement and routing Pyxis was used from Mentor Graphics[©]. This design was created in order to demonstrate the effect of temperature variations and evolving hot-spots on cell propagation delay. Cell delays and power dissipations has been extracted from preceding SPICE simulations of the circuit.

Standard Delay Format (SDF) files were created from the extracted delay data for the temperature dependent timing simulation. To be able to visibly

demonstrate the effect of temperature variation on delays, the propagation delay data in the SDF have been slightly modified manually. In spite of this, the true temperature dependence of cell delays has been simulated with SPICE.

In fig. 1 the schematic layout of the design is shown. The P&R algorithm placed the cells in two rows. The four buffer cells were driven by an external stimulus like a 4-bit counter. The period of the counter was $1\ \mu\text{s}$. This feature has been implemented to demonstrate both self-heating and induced heating of the ring oscillator inverter chain. The inverter chain consists of 10 inverters and one kick-in NAND gate.

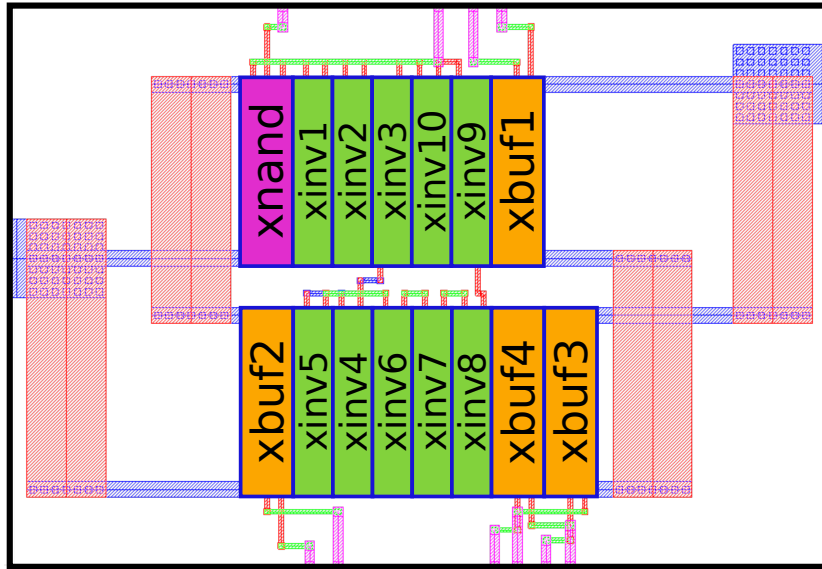


Figure 1: Design layout of the ring oscillator circuit

4. Grid partitioning

In CellTherm the surface of the standard cell IC can be divided into subregions called *partitions* where dissipated powers are accumulated and temperatures are calculated. The partitioning approach speeds up simulation times and initial database parsing in large designs containing more than 1000 standard cells. This approach makes the initial thermal model generation practically insensitive to the number of standard cells. Of course logic simulation can take longer for designs containing high number of cells,

the time taken by the thermal engine to solve equations remains the same because thermal equations are generated for the partitioning grid not the standard cells.

Powers dissipated in partitions and temperatures of cell instances are calculated using the partition area and overlapping cell area ratio. This means that if a standard cell falls partly into a partition, then the cell's power dissipation is taken into account proportionally to the overlap ratio.

Temperature and power values inside the partitions are calculated using bilinear interpolation with a higher resolution. For example, for a 10×10 mesh of partitions, the bilinear interpolation is done for a 100×100 mesh. This way a fine detailed power and temperature distribution can be displayed. The error introduced by the partitioning has been calculated by comparing the partitioned design with a simulation where partitioning was switched off. The resulting error functions are shown in fig. 2. The error in steady-state operation stays below 3%.

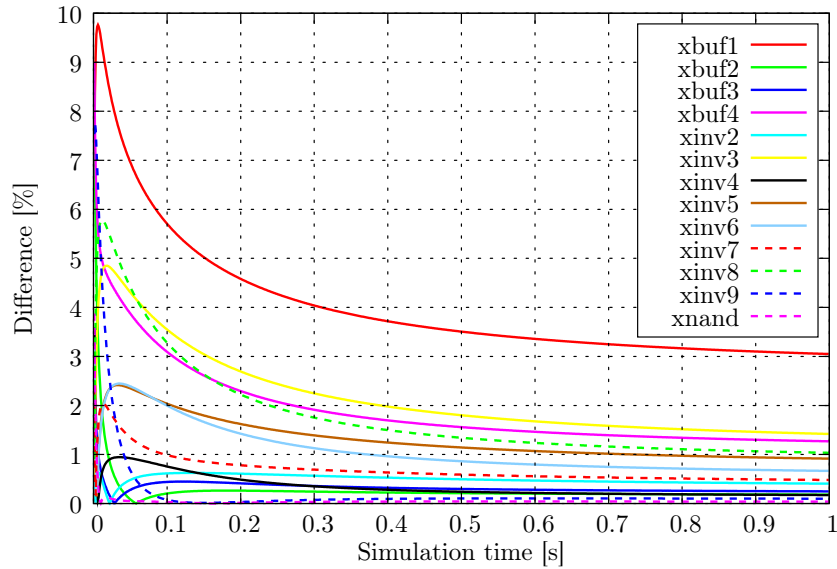


Figure 2: Error introduced by partitioning

When using partitioned calculation in CellTherm, partitions can be subdivided into subpartitions down to an unlimited depth. This way temperature and delay resolution in critical areas of the circuit can be refined. In a basic scenario the partitions can be sliced into 4 equal parts that can further be sliced into another 4 parts as shown in fig. 3

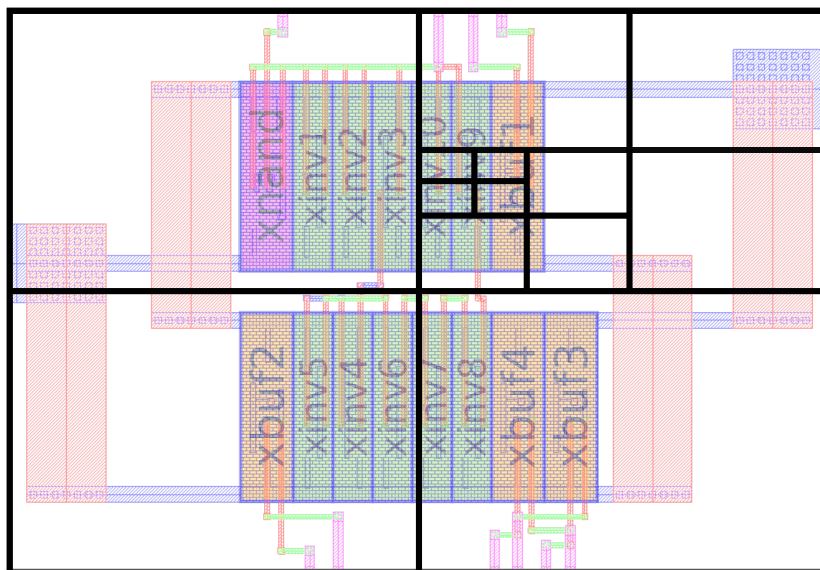


Figure 3: Subpartitioning area of interest

5. Power and temperature distribution

Fig. 4 depicts the power distribution of the design. In our test case, the BUFFER cells are driven in a counter-like pattern, that is, switching activity of the flip-flop chain can be specified with (1),

$$A(\text{xbuf4}) = 2 \cdot A(\text{xbuf3}) = 4 \cdot A(\text{xbuf2}) = 8 \cdot A(\text{xbuf1}) \quad (1)$$

where $A()$ means the switching activity of the cell. The dissipated dynamic power per logic transition is proportional to the switching activity.

In the power map it can be seen that due to the evenly distributed layout, the power dissipation spreads evenly in the proximity of the rows. The resulting temperature map is shown in fig. 5. Smooth transitions in power and temperature distribution figures is achieved by bilinear approximation among partitions. The approximation was run with 100×100 resolution.

6. Data serialization

The initial database for the logi-thermal simulation is generated from the *LEF/DEF* layout files, Liberty *.lib* files, and *SDF* files. In a design with thousands of standard cells this initial database creation can take hours

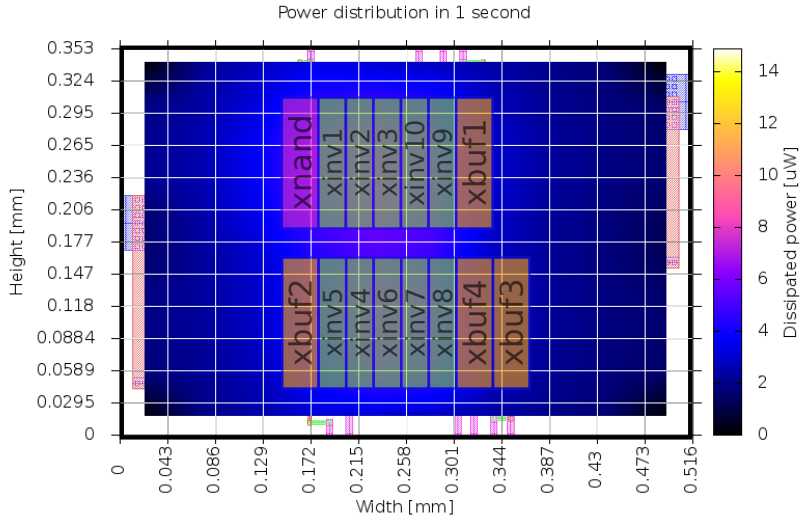


Figure 4: Power map of test design

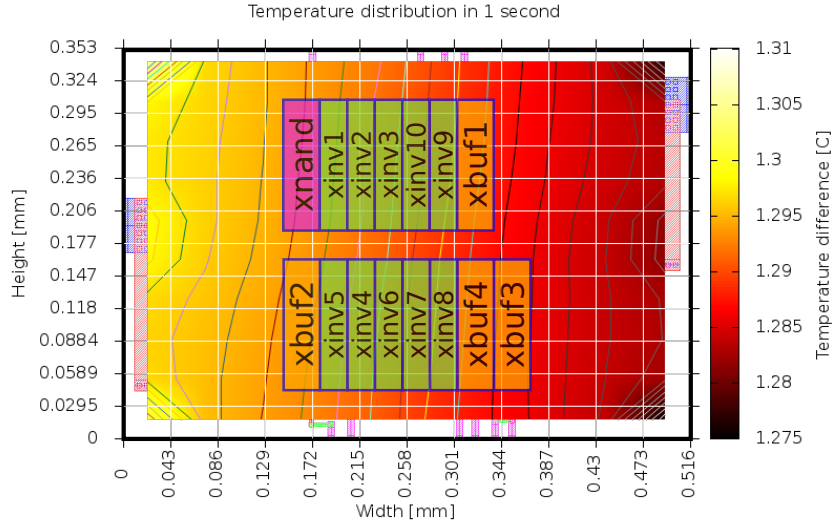


Figure 5: Temperature map of test design

to complete. For example, in a design containing 1490 cells manufactured on a 65nm STMicroelectronics process the initial database generation and

thermal model generation took 56 minutes. This initial process of internal database creation and thermal model generation has to be done only once in the beginning of the simulation, however, when running simulations with different stimuli the database creation has to be done every time a new simulation is started.

In this paper a serialization method is described that speeds up initial loading times of simulations. After the initial database creation in the beginning of the simulation, CellTherm can be asked to serialize the created internal database to disk.

In computer science, in the context of data storage and transmission, serialization is the process of converting a data structure or object state into a format that can be stored (for example, in a file or memory buffer, or transmitted across a network connection link) and "resurrected" later in the same or another computer environment [12].

The serialized data can be read back from disk in fractions of seconds when starting a new simulation run thus the time consuming process of thermal model generation and database creation can be skipped. The deserialized data in memory will result in the same data as if the database creation phase had been run.

A comparison is presented here between the serialized and the pure initial loading time of the simulator. When loading without serialization the load time was 99.654 seconds. The serialized load time resulted to 0.262 second measured by the TCL command *time*. The thermal model generation time depends only on the structure and size of the chip die and packaging which does not change from simulation to simulation.

7. Thermal model generation

In this section we describe the steps needed to create the thermal RC model of microelectronics structures. Fig 6 shows a typical structure of an integrated circuit.

1. In an integrated circuit like depicted in fig. 6, different structure parameters are required for generating the thermal RC model. These parameters are
 - lateral size of the structure (width, height),
 - width of layers,

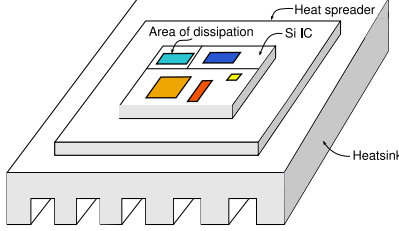


Figure 6: Typical IC structure

- material parameters of the layers (heat transfer coefficient, heat capacity),
- boundary conditions (adiabatic, isothermal, heat transfer),
- location and size of dissipation areas

From these parameters the CellTherm simulator solves the time dependent Laplace equation of heat transfer (2)

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = \frac{c}{\lambda} \frac{\partial T}{\partial t}, \quad (2)$$

where T is the temperature, x , y , z are three dimension coordinates, c is unit heat capacity, λ is heat transfer coefficient and t is the time. The algorithm and solution of the Laplace equation is described in [13, 14]. With the solution of the Laplace equation the power step response ($a(t)$) of the thermal structure can be determined. In a system where multiple dissipation sources are present the step response functions must be determined for all sources. The algorithm takes into account the mutual heating effect of multiple sources. This means that for N sources we get N^2 step response functions.

In microelectronic systems thermal time constants spread over a wide spectrum. There can be several orders of magnitude difference between time constants. In order to be able handle time constants in a wide range, it is practical to represent the $a(t)$ step response function in logarithmic scale, that is, the step response function becomes $a(z)$ with a substitution of $z = \ln t$.

2. Knowing the system's step response functions ($a(z)$) the time constant spectrum ($R(z)$) for one source can be determined by deconvolution of

the response. The deconvolution process of the step response functions is explained in [15, 16]. The time constant spectrum can be calculated either from the frequency domain or the time domain response. The procedure is called *Network Identification with Deconvolution, NID*. Fig. 7 shows a schematic time spectrum function.

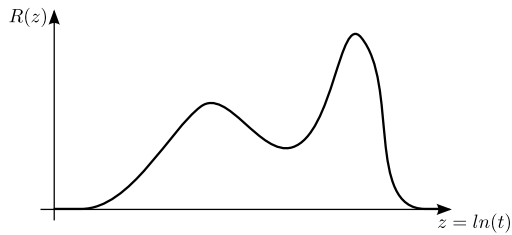


Figure 7: Schematic time constant spectrum

3. The thermal time constant spectra of the system model the thermal behavior of the system efficiently in both the frequency and time domain. The time constant spectra are also providing an opportunity to create compact RC models of the structure. With these RC models the thermal behavior of the structure is represented by an electrical RC network that can be used to simulate thermal transients with an analog SPICE simulator.

In order to generate the compact model the discretization of the time constant spectra is needed because the spectra are continuous functions. In order to be able to investigate the derived model in an analog electrical simulator, the time constant spectrum must be quantized to a finite number of RC terms. To achieve the reduction the continuous time constant spectrum must be divided to N equal sections as fig. 8 shows.

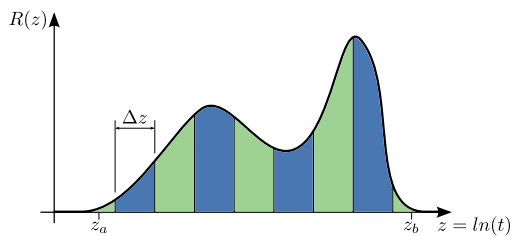


Figure 8: Divided time constant spectrum

The discrete z_i time constant values are calculated from the equally divided spectrum depicted in fig. 9. The z_i values are in the middle of the sections and their height is determined by (3) as the area below the corresponding function section.

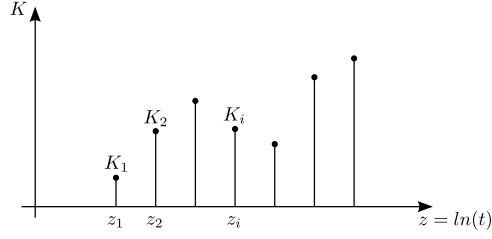


Figure 9: Quantized time constant spectrum

$$K_i = \int_{z_a+(i-1)\Delta z}^{z_a+i\Delta z} R(\zeta)d\zeta \quad (3)$$

4. The compact RC Foster model parameters can be directly calculated from the (z_i, K_i) pairs. After transforming the z_i values back into the linear time domain the (τ_i, K_i) pairs can be used to calculate the capacitances (4). The R_i values are equal to the K_i values.

$$C_i = \frac{\tau_i}{K_i} \quad (4)$$

In our work we observed that a quantization of the time constant spectra to 12 RC terms provides accurate thermal simulation results[17],[18]. Fig. 10 shows an example of the resulting Foster RC model.

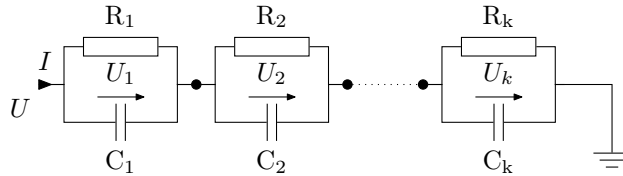


Figure 10: Foster RC ladder

5. Using the compact RC models the thermal transients can be simulated with analog SPICE simulators.

In this work we used the presented methodology to generate and simulate the thermal model of the IC structures. The Therman thermal simulator developed in the Department of Electron Devices, BME has been adapted for EDA compatibility and integrated into the CellTherm logi-thermal simulator.

8. Equivalent thermal RC network calculation

CellTherm determines the power dissipated by the standard cells using the switching activity extracted from the logic simulator, and the Liberty power database. This power heats up the device during operation. In the beginning of the simulation CellTherm builds the equivalent thermal representation of the device in terms of an RC network. It does this from the layout data, the position of the cells on the surface and the layer structure. Due to the fact that thermal resistances and thermal capacitances can be modeled by resistances and capacitances in the electrical domain (using the heat equation and transmission line analogy), the equivalent thermal RC network can be solved by ordinary electrical network theory. Accordingly, the equivalent thermal RC networks can be analyzed and simulated with SPICE simulators. In the thermal domain, the corresponding quantity of *current* is *power*, and the corresponding quantity of *voltage* is *temperature*.

CellTherm generates the thermal time constant spectra of the physical structure using the Fourier solution of the heat equation. From these continuous spectra it creates a network of Foster RC ladders of finite poles by sampling the continuous spectra. The resulting RC network can then be simulated and analyzed with a SPICE simulator and the resulting temperatures can be calculated in function of the input powers. In CellTherm, the smallest building block of the circuit is the standard cell, therefore the voltages on the thermal Foster network's nodes correspond to the cell's temperatures. Every cell has a self-impedance (a Foster ladder) which is liable for calculating the self-heating of the cell, and $n - 1$ cross-impedances which typify the other cell's heating influence to the given cell. n here denotes the number of cells in the design.

Fig. 11 shows that every cell's temperature is derived from the self-heating and the other cell's temperature together. One cell's resulting temperature is comprised of the power (current) "flowing" through the self-impedance

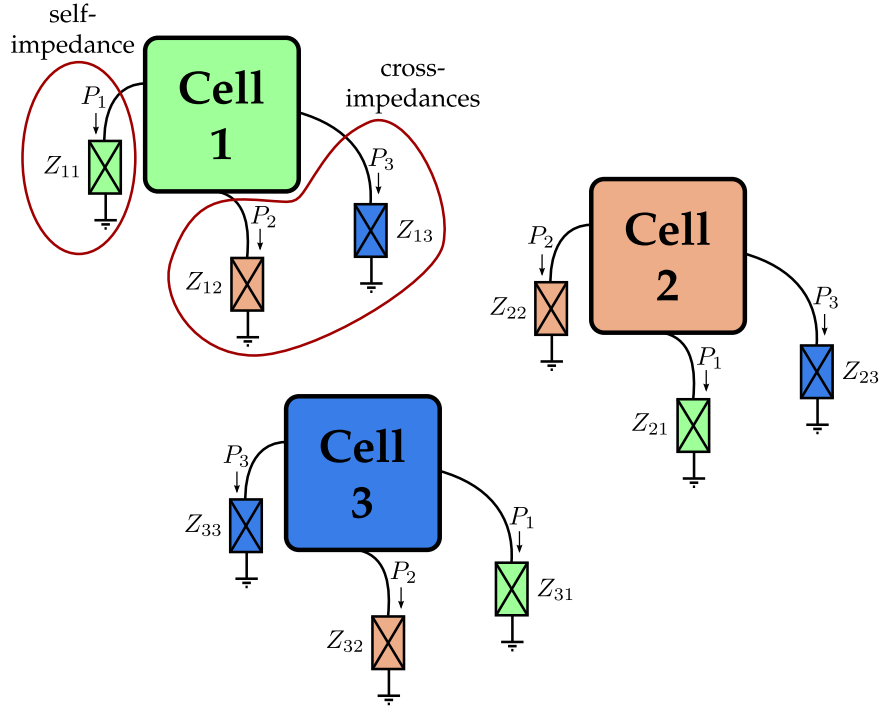


Figure 11: Thermal self- and cross-impedances

and powers flowing through the cross-impedances modeling the other cell's influence. The power flowing through the cross-impedances corresponds to the other cells' dissipated power (see equation (5)).

$$T_1 = P_1 Z_{11} + P_2 Z_{12} + P_3 Z_{13} \quad (5)$$

In a circuit with n cells equation (5) transforms to equation (6).

$$T_1 = P_1 Z_{11} + P_2 Z_{12} + \cdots + P_n Z_{1n} \quad (6)$$

$$T_2 = P_1 Z_{21} + P_2 Z_{22} + \cdots + P_n Z_{2n} \quad (7)$$

⋮

$$T_n = P_1 Z_{n1} + P_2 Z_{n2} + \cdots + P_n Z_{nn} \quad (8)$$

Writing (6)-(8) in a matrix form, we get (9).

$$\begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_n \end{pmatrix} = \begin{pmatrix} Z_{11} & Z_{12} & \cdots & Z_{1n} \\ Z_{21} & Z_{22} & \cdots & Z_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ Z_{n1} & Z_{n2} & \cdots & Z_{nn} \end{pmatrix} \cdot \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{pmatrix} \quad (9)$$

In equations (5)–(9) the Z impedances symbolize the Foster ladders for each node. A typical Foster RC ladder is shown in fig. 12.

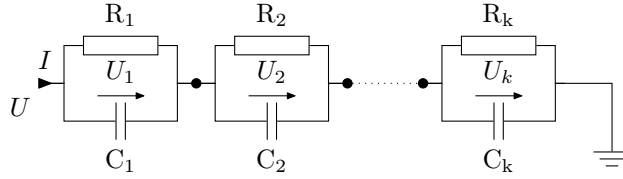


Figure 12: Foster RC ladder

Calculating node voltages in such a simple circuit can be simplified down to a specific solution without the need to implement a linear network solver. Calculation of the node voltages can be done in the following way. Assume we know the Foster RC ladder's input current (the power in the thermal domain). We need to calculate the voltage for node U (the temperature in the thermal domain). This voltage corresponds to one $P_n Z_{nm}$ term in (8). To determine the voltage of the poles in the Foster RC ladder (U_1 – U_k) we can write equation of the currents flowing through the poles (see equations (10)–(12)).

$$I = \frac{U_1}{R_1} + C_1 \cdot \frac{dU_1}{dt} \quad (10)$$

$$I = \frac{U_2}{R_2} + C_2 \cdot \frac{dU_2}{dt} \quad (11)$$

⋮

$$I = \frac{U_k}{R_k} + C_k \cdot \frac{dU_k}{dt} \quad (12)$$

The current flowing through the capacitance can be calculated with the $I_k = C \cdot \frac{dU_k}{dt}$ equation, where the $\frac{dU_k}{dt}$ differential term is approximated with (13).

$$\frac{dU_k}{dt} \approx \frac{U_k(t_i) - U_k(t_{i-1})}{\Delta t} \quad (13)$$

$U_k(t_i)$ means the *actual* value of the voltage and $U_k(t_{i-1})$ means the *previous* value.

$$t_i = t_{i-1} + \Delta t \quad (14)$$

Δt is the time step of how often the simulator calculates a new voltage value. In this case the voltage corresponds to the node's temperature and Δt means the time period for which the accumulated dissipations are averaged. The typical time constants of a digital system (clock period, switching frequency) are much smaller than the typical time constants of a thermal system. To this end, the temperature calculations can be done in larger time intervals. For example, assume that in a digital system the clock frequency is 100 Mhz. This means a 10 ns clock period. A thermal system usually has time constants ranging typically from a few tenths of seconds to several tens of seconds or even minutes (for example a hot cup of coffee can take minutes to cool down to the ambient temperature). Due to the several orders of magnitude difference between the electrical and thermal time constants the averaging of the dissipation can be done in larger intervals (e.g. 10 milliseconds). The average of the dissipations taken place during this interval gives the input powers for the Foster RC ladders (the input current). In this example Δt will be 10 ms.

By transforming (12) we get (15).

$$U_k(t_i) = \frac{I + \frac{C_k}{\Delta t} U_k(t_{i-1})}{\frac{1}{R_k} + \frac{C_k}{\Delta t}} \quad (15)$$

By calculating (15) for every pole of the Foster RC ladder and adding them up we get the voltage (the temperature) on node U .

$$U = U_1(t_i) + U_2(t_i) + \dots + U_k(t_i) \quad (16)$$

$$U = \sum_{k=1}^{\text{poles}} U_k(t_i) \quad (17)$$

As previously mentioned, the calculated voltage on node U (equation (17)) stands for *one* term in the temperature equation for one cell (see equation (8)). The resultant temperature of one cell in the circuit evolve from the self-heating of the cell and the influence of other cells' temperature. By calculating every term in (8) using the previously proposed algorithm and accumulating them we get the cell's resultant temperature in the current time step of the simulation. This calculation shall be done for every cell in the design to get the temperatures of all the cells.

9. Temperature-delay functions

The temperature dependent delays of the standard cells can be calculated using the SPICE simulations of the cells. These measurements provide the most accurate temperature-delay functions but the simulated cells given by transistor-level netlists do not contain layout data. Propagation delay of standard cells depends strongly on placement and wiring.

Temperature-delay functions for the placed and routed design can also be calculated from synthesizer-generated SDF files. SDF files contain timing and delay data for cells in the placed and routed design. Synthesis tools usually generate these SDF files within voltage, process and temperature corners. Timing data are present in the SDF file for the worst case, nominal and best case corners. From these corner cases the corresponding temperatures and thus the related delays can be extracted. By interpolating the extracted temperature-delay corners delays can be calculated for arbitrary temperature values. Listing 1 shows an example SDF file.

In fig. 13–15 the temperature dependent propagation delays of the cells can be seen. In fig. 15 an interesting phenomenon can be observed. The NAND cell's delay decreases with rising temperature that means that the cell performs faster on higher temperatures. This phenomenon is often referred as Reverse Temperature Dependency (RTD) or Inverse Temperature Dependency (ITD). The authors of [4] draw attention to the fact that this

```

(DELAYFILE
(SDFVERSION "3.0")
(DESIGN "dut")
(DATE "2013/06/07_15:35")
(VENDOR "EET_Semiconductor")
(PROGRAM "CellTherm")
(VERSION "0.9")
(DIVIDER /)
(VOLTAGE 4.5:5.0:5.5)
(PROCESS "typical")
(TEMPERATURE 0:25:65)
(TIMESCALE 1ns)
(CELL
  (CELLTYPE "inv01")
  (INSTANCE xinvt)
  (DELAY
    (ABSOLUTE
      (IOPATH A Y (0.02912:0.03140:0.03291) )))
)
(CELL
  (CELLTYPE "nand02")
  (INSTANCE xnand)
  (DELAY
    (ABSOLUTE
      (IOPATH A0 Y (0.03777:0.03813:0.03889) )
      (IOPATH A1 Y (0.03764:0.03817:0.03870) )))
)
)

```

Listing 1: Example SDF file

phenomenon should be accounted for in upcoming electro- and logi-thermal simulators.

When a circuit is operating in low voltage, the propagation delay of a cell may decrease as the temperature increases. The reason behind ITD effect is due to the temperature effect on the threshold voltage, V_{th} . As supply voltage (VDD) scaled, the value of $|V_{GS} - V_{th}|$, the absolute difference between transistor gate to source voltage and threshold voltage, decreases. The smaller $|V_{GS} - V_{th}|$ makes saturation current more sensitive to change in V_{th} , which decreases as the increase of temperature. The smaller V_{th} incurs more current that makes the device switching faster. On the other hand, transition delay is also proportional to the electron mobility, which decreases as the temperature rises. Hence the device performance depends on the racing condition of electron mobility and V_{th} together as temperature rises. Traditionally, timing is signed off at two extreme temperature corners, one representing the best case and the other representing the worst case. With ITD, the highest sign-off temperature can no longer guarantee the worst case, and vice versa. This poses a serious problem to the timing sign-off methodology, i.e. it is possible that the worst-case temperature occurs at some intermediate point and finding this point can be quite difficult[19].

With the SDF approach mentioned above, CellTherm can handle these

inverse temperature dependencies of cells easily because the SDF files contain sampled points of the true temperature-delay functions.

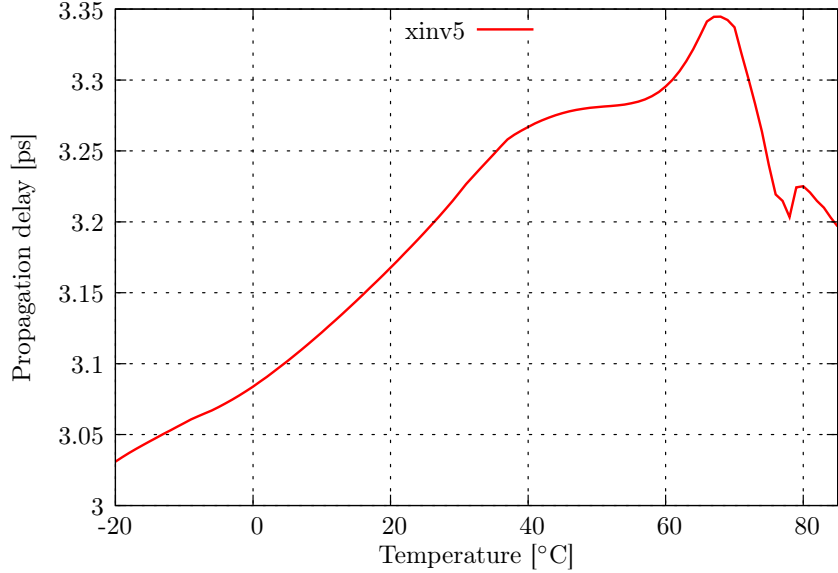


Figure 13: Propagation delay of INVERTER gate

10. Temperature dependent delay simulation

In the test vehicle showed in fig. 1 the temperature-dependent frequency variation of the ring oscillator is demonstrated. By the self-heating of the ring oscillator’s inverters and the dissipation of the buffer cells, the temperature distribution on the IC surface will not be uniform thus each inverter cell in the ring oscillator will have a different propagation delay. The delays are calculated from the standard cell’s current temperature and updated in every simulation timestep. The varying delays of the cells will mistune the frequency of the ring oscillator. The changing frequency over simulation time is monitored constantly by the logi-thermal simulator and displayed to the user in every simulation timestep. The output frequencies of the ring oscillator after different simulation times can be seen in Table 1.

11. Results

Temperature curves versus simulation time for the *xinv5*, *xnand* and *xnand* cells is shown in fig. 16. It can be observed that the circuit reaches

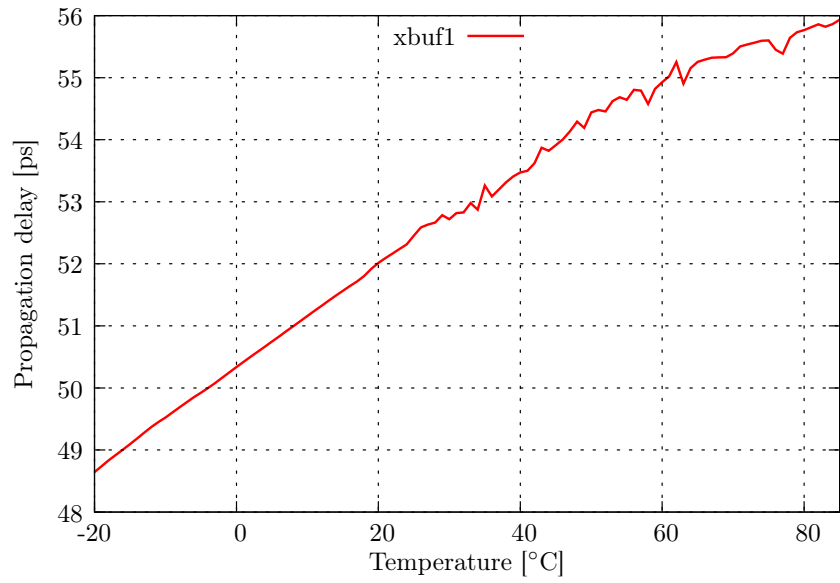


Figure 14: Propagation delay of BUFFER gate

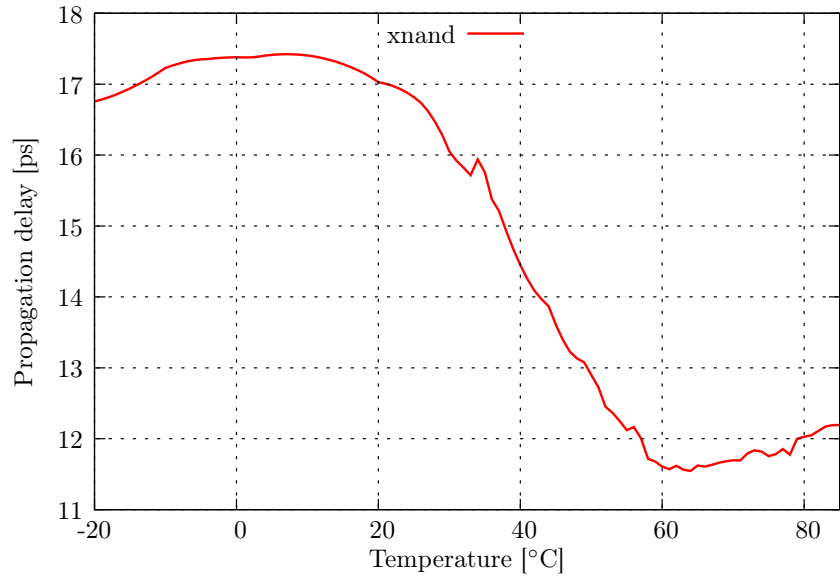


Figure 15: Propagation delay of NAND gate

steady-state temperature near the 4th second. The temperatures throughout this paper are differential not absolute temperature values.

Table 1: Output frequency of ring oscillator depending on device temperature

Simulation time [ms]	Period [ns]	Frequency [MHz]
10	6.576	152.06
250	7.542	132.59
500	7.938	125.97
750	8.136	122.91
1000	8.246	121.27
2000	8.380	119.33
3000	8.402	119.01
3250	8.410	118.90
3500	8.420	118.76
4000	8.422	118.73

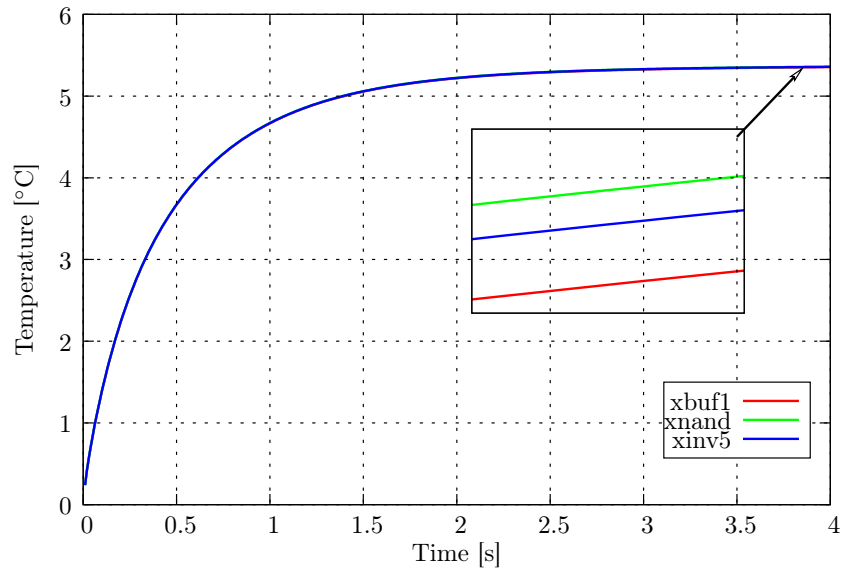


Figure 16: Temperature of some cells

In fig. 17 the period and frequency of the ring oscillator is depicted versus

simulation time. It is clearly visible that with the increasing simulation time the temperature also rises thus the period of the ring oscillator starts to rise also. As the temperature of the functioning circuit rises the period of the ring oscillator also rises according to the temperature-delay functions in fig. 13–15. This increase in the period means a decrease in the oscillation frequency as shown by fig. 17. The frequency of the oscillator drops from the initial 152.06 MHz to 118.73 MHz in the 4th second.

As the thermal steady state is reached near the 4th second, the frequency also reaches a steady state at the 4th second. This phenomenon is visible in fig. 17.

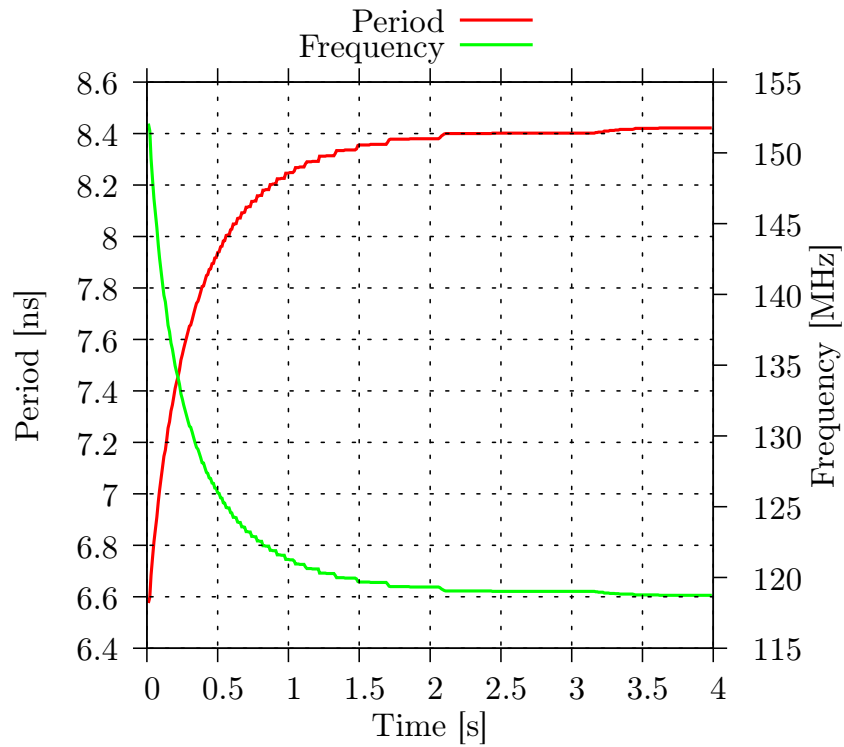


Figure 17: Period and frequency of ring oscillator over time

12. Assessment and validation

The CellTherm simulations has been verified with several approaches. From the SPICE simulation of the transistor-level netlist of the design, the

dissipations and delays of the cells has been extracted. Dissipation and timing data could have been also extracted from a Liberty database of the process. A Liberty file in our process node (TSMC 0.35 μm) was not present, so we needed to extract the parameters from SPICE simulations manually (Liberty databases claim to be inside 2% of accuracy compared to SPICE simulations [20]).

The SPICE simulation time was 64 μs . This was enough to reliably sample power, delay and frequency of the circuit. The average power dissipations were compared to the dissipations measured by CellTherm. As a next step, the equivalent thermal Foster RC networks that represent the structure and layout were transformed to a SPICE compatible netlist. The average cell powers calculated in the previous step has been fed into the Foster RC network that resulted in the temperature functions of the cells. Finally, these functions simulated with SPICE and the CellTherm temperature curves (shown in fig. 16) were compared. The difference between the SPICE and CellTherm temperature curves was less than 0.16%. The calculated difference function can be seen in fig. 18.

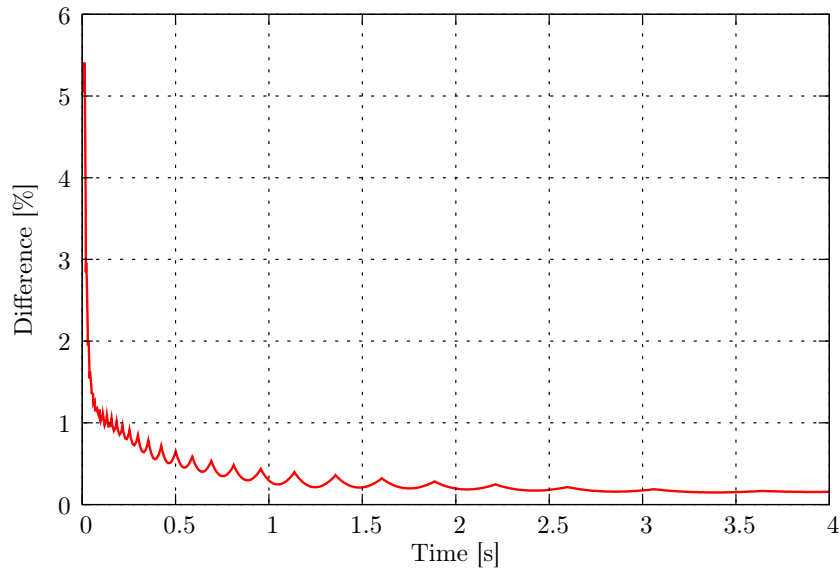


Figure 18: CellTherm vs SPICE simulation difference

Another validation has been done to further verify the correct operation of CellTherm. A simulation has been run using CellTherm on the demonstration circuit until steady-state temperature. The previously mentioned SPICE

netlist simulation was executed on the steady-state temperature measured with CellTherm in the last step. The output frequency of the ring oscillator was measured in both the CellTherm and SPICE results. The SPICE simulation resulted in $f_{osc} = 1.453$ GHz. CellTherm measured $f_{osc} = 1.524$ GHz. The difference is below 4.82%. This difference could be further eliminated by providing delay values for every type of transitions ($0 \rightarrow 1$, $1 \rightarrow 0$, $0 \rightarrow x$, $1 \rightarrow x$, etc.) in the SDF file.

13. Summary and conclusion

In this paper a methodology for simulating temperature dependent propagation delays in a ring oscillator circuit is demonstrated. A special demonstration circuit has been created to spectacularly demonstrate the effect of circuit self-heating on propagation delays and operating frequency.

Temperature dependent delays are calculated from synthesizer-generated SDF files making thermal-aware logic simulations possible.

A grid partitioning method has been introduced where the created thermal model is independent of the number of standard cells in the circuit. This approach speeds up initial thermal model generation and does not scale with increasing standard cell count.

This paper also proposed a method of serialization that can speed up initial database loading for the logi-thermal simulation with CellTherm.

A method for fast and effective calculation of temperatures from the thermal equivalent Forster RC network has been also proposed.

Finally, simulation results are shown to prove the concept introduced in this paper. CellTherm simulation results were compared to SPICE simulations and resulting difference between temperatures was less than 0.16%. Output frequencies were also compared to SPICE simulations conducted on the steady-state temperature and difference below 4.82% was measured.

14. Acknowledgement

This work was partly supported by the IP 248603 THERMINATOR FW7 project of the European Union and by the Hungarian Government through TÁMOP-4.2.1/B-09/1/KMR- 2010-0002. I would like to thank Dr. Vladimir Szekely for his consultation and very useful help as well as for some program parts in the CellTherm application.

References

- [1] S. Pable, M. Hasan, Ultra-low-power signaling challenges for subthreshold global interconnects, *Integration, the VLSI Journal* 45 (2012) 186 – 196.
- [2] B. Rebaud, M. Belleville, E. Beigné, C. Bernard, M. Robert, P. Maurine, N. Azemard, Timing slack monitoring under process and environmental variations: Application to a DSP performance optimization, *Microelectronics Journal* 42 (2011) 718 – 732.
- [3] S. Lin, Y.-B. Kim, F. Lombardi, Design and analysis of a 32nm PVT tolerant CMOS SRAM cell for low leakage and high stability, *Integration, the VLSI Journal* 43 (2010) 176 – 187.
- [4] R. Kumar, V. Kursun, Reversed temperature-dependent propagation delay characteristics in nanometer CMOS circuits, *Circuits and Systems II: Express Briefs, IEEE Transactions on* 53 (2006) 1078 –1082.
- [5] D. Wolpert, P. Ampadu, A sensor to detect normal or reverse temperature dependence in nanoscale CMOS circuits, in: *Defect and Fault Tolerance in VLSI Systems, 2009. DFT '09. 24th IEEE International Symposium on*, pp. 193–201.
- [6] C. Sánchez-Azqueta, S. Celma, F. Aznar, A 0.18 μm CMOS ring VCO for clock and data recovery applications, *Microelectronics Reliability* 51 (2011) 2351 – 2356.
- [7] J. Wang, N. Gong, L. Hou, X. Peng, R. Sridhar, W. Wu, Leakage current, active power, and delay analysis of dynamic dual vt cmos circuits under P–V–T fluctuations, *Microelectronics Reliability* 51 (2011) 1498 – 1502. *Proceedings of the 22th European Symposium on the RELIABILITY OF ELECTRON DEVICES, FAILURE PHYSICS AND ANALYSIS.*
- [8] A. Winther, W. Liu, A. Nannarelli, S. Vrudhula, Temperature dependent wire delay estimation in floorplanning, in: *NORCHIP, 2011*, pp. 1 –4.
- [9] A. Golda, A. Kos, Temperature influence on power consumption and time delay, in: *Digital System Design, 2003. Proceedings. Euromicro Symposium on*, pp. 378 –382.

- [10] A. Timar, G. Bogнар, A. Poppe, M. Rencz, Electro-thermal co-simulation of ICs with runtime back-annotation capability, in: Thermal Investigations of ICs and Systems (THERMINIC), 2010 16th International Workshop on, Barcelona, Spain, pp. 1 –6.
- [11] A. Timar, G. Bogнар, M. Rencz, Improved power modeling in logi-thermal simulation, in: Thermal Investigations of ICs and Systems (THERMINIC), 2011 17th International Workshop on, Paris, France, pp. 1 –6.
- [12] Serialization, <http://en.wikipedia.org/wiki/serialization> (2012).
- [13] I. Lászó, Villamos gépek és eszközök melegedése és hűtése, Műszaki Könyvkiadó, 1982.
- [14] V. Székely, A. Poppe, M. Rencz, M. Rosental, T. Teszéri (????).
- [15] V. Székely, THERMODEL: a tool for compact dynamic thermal model generation, *Microelectronics Journal* 29 (1998) 257 – 267. <ce:title>Thermal Investigations of ICs and Microstructures II</ce:title>.
- [16] V. Székely, M. Rencz, A. Poppe, B. Courtois, THERMODEL: A tool for thermal model generation, and application for MEMS, *Analog Integrated Circuits and Signal Processing* 29 (2001) 49–59.
- [17] A. Timar, M. Rencz, Real-time heating and power characterization of cells in standard cell designs, *MICROELECTRONICS JOURNAL* (2012).
- [18] A. Timar, M. Rencz, Acquiring real-time heating of cells in standard cell designs, in: Test Workshop (LATW), 2012 13th Latin American, pp. 121–125.
- [19] S. H. Wu, A. Tetelbaum, L.-C. Wang, How does inverse temperature dependence affect timing sign-off, in: A. Amara, T. Ea, M. Belleville (Eds.), *Emerging Technologies and Circuits*, volume 2021 of *Lecture Notes in Electrical Engineering*, Springer Netherlands, 2010, pp. 179–189.
- [20] G. Mekhtarian, Composite Current Source (CCS) Modeling Technology Backgrounder, Synopsys, 2005.11 edition, 2005.