7th RTD Framework Program

REALITY

Reliable and Variability Tolerant System-on-a-Chip Design in More-Moore Technologies

Contract No 216537

COOPERATION

Information Society Technologies

Deliverable D2.2

Methods for statistical variability simulation from the transistor to the system level

Editor: Paul Zuber, Miguel Miranda
Co-author / Acknowledgement: Asen Asenov, Yves Laplanche, Davide Pandini, Campbell Millar, Andrew Brown

Status - Version: V1.2
Date: 09/02/2009
Confidentiality Level: Public
ID number: IST-216537-WP2-D2.2-v1.2.doc

The REALITY Consortium consists of:

Interuniversity Microelectronics Centre (IMEC vzw)	Prime Contractor	Belgium
STMicroelectronics S.R.L. (STM)	Contractor	Italy
Universita Di Bologna (UNIBO)	Contractor	Italy
Katholieke Universiteit Leuven (KUL)	Contractor	Belgium
ARM Limited (ARM)	Contractor	United Kingdom
University Of Glasgow (UoG)	Contractor	United Kingdom

1. Disclaimer

The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

2. Acknowledgements

The editors Paul Zuber and Miguel Miranda acknowledges contributions by project partners.

3. Document revision history

Date	Version	Editor/Contributor	*Comments*
13/01/2009	V0.1	Paul Zuber	Framework, ST and IMEC contributions
20/01/2009	V0.2	Miguel Miranda	General Updates, added conclusion, figure, tables, references, etc.
23/01/2009	V0.3	Andrew Brown, Yves Laplanche, Miguel Miranda	Specific updates from UoG, and ARM, format conforming
24/01/2009	V1.0	Davide Pandini, Miguel Miranda	Final Update STM and IMEC
09/02/2009	V1.2	Miguel Miranda, Tom Tassignon	Final version

4. Preface

The scope and objectives of the REALITY project are:

- Development of design techniques, methodologies and methods for real-time guaranteed, energy-efficient, robust and adaptive SoCs, including both digital and analogue macro-blocks“

The Technical Challenges are:

- To deal with increased static variability and static fault rates of devices and interconnects.
- To overcome increased time-dependent dynamic variability and dynamic fault rates.
- To build reliable systems out of unreliable technology while maintaining design productivity.
- To deploy design techniques that allow technology scalable energy efficient SoC systems while guaranteeing real-time performance constraints.

Focus Areas of this project are:

- “Analysis techniques” for exploring the design space, and analysis of the system in terms of performance, power and reliability of manufactured instances across a wide spectrum of operating conditions.

- “Solution techniques” which are design time and/or runtime techniques to mitigate impact of reliability issues of integrated circuits, at component, circuit, architecture and system (application software) design.

The REALITY project has started its activities in January 2008 and is planned to be completed after 30 months. It is led by Dr. Miguel Miranda of IMEC. The Project Coordinator is Mr. Tom Tassignon. Five contractors (STM, ARM, KUL, UoG, UNIBO) participate in the project. The total budget is 2.899 k€.

5. Abstract

The goal of this WP is to develop advanced methodologies and techniques for Statistical Analysis all the way from the device level to the system level. The WP also targets developing and fully characterizing a limited standard cell library (50-100 cells) for synthesis based on restricted design rules for use in WP2, WP3, WP4, and WP5. Novel techniques to percolate variability all the way from the device level to the system level shall be developed to evaluate the impact that intrinsic variability will have on timing, energy, and yield of the complete SoC architecture, including a view on the impact of application-dependent activity. Commercial EDA solutions (e.g., fast circuit simulators, SSTA tools, power analysis tools, etc) shall be reused in the flow wherever possible in combination with Monte Carlo-based simulation techniques. Also considered in this WP is the strategic aspect of the standardization of the interfaces between different abstraction levels to enable the propagation of variability specific information throughout the design flow in order to guarantee the compatibility with existing electronic design simulation/verification tools.

This document is the deliverable D 2.2 comprising method descriptions for Tasks 2.2-2.4, namely Statistical Characterization of Macro-Blocks, Statistical Analysis of Digital Blocks, and Statistical Analysis of SoC Architectures.

6. List of Abbreviations

REALITY	Reliable and Variability tolerant System-on-a-chip Design in More-Moore Technologies
PDF	Probabilistic Density Function
RTL	Register Transfer Level
SoC	System on Chip
EDA	Electronic Design Automation
SSTA	Statistical Static Timing Analysis
IP	Intellectual Property (block)
WID	Within Die Variations
CDF	Cumulative Density Functions
CPU	Central Processing Unit
MOSFET	Metal Oxide Field Effect Transistor
SRAM	Static Random Access Memory

7. List of Tables

Empty

8. List of Figures

9. Table of contents

10. Introduction

Given the increasing importance of intrinsic variability in electronic design in advanced nanometer technologies (i.e., 45nm and 32nm), system designers need to estimate its impact on timing (and eventually energy) parametric yield before tape-out in order to meet the constraints on system performance (and battery lifetime). This can only be achieved with a design framework that percolates the process variability impact on timing and energy parameters from the device level (i.e., device compact model) all the way to the Register Transfer Level (RTL), also taking into account the effects of application-dependent activity if energy is considered. The framework can also be used to study the impact of environmental effects like supply voltages and/or temperature fluctuations on reliability and parametric ageing, and on the switching time and current amplitudes of the transistors.

In this work package (WP 2) we developed methodologies and techniques for Statistical Analysis (see) from a device compact model representative of 32nm technology (resulting from WP1) to the system level. We have explicitly targeted a design framework that can unify existing commercial and academic tools into a holistic analysis/simulation flow. Commercial EDA solutions such as simulators, Statistical Static Timing Analysis (SSTA) tools, power analysis tools, etc., can be reused wherever necessary, eventually in combination with Monte Carlo-based simulation techniques. The figure clearly illustrates that standard cell logic and transistor-level macro blocks require different approaches for variability characterization.

Overall strategy

System architectures in general, comprise IP components and glue logic (typically implemented using standard cells, thus it can be treated in the same way as standard cell-based IP blocks). IP components can include soft and hard macro-blocks, mixed signal or analogue circuits. Each of them is specified in a different manner. Soft IP is usually delivered as synthesizable RTL code, while hard macros or analogue components as layouts. A generic framework for variability propagation should be able to handle all these alternatives.

The level below system architecture differentiates between IP components built using standard cell logic and macro-blocks (e.g. memories, register files, etc.) that exist as transistor net-lists or layouts from which the netlist can either be extracted or generated via specific compilers (e.g., memory compilers). In case of IP macro-blocks, timing and energy are usually characterized using transistor-level simulations and given the increasing complexity of the typical macro-blocks in SoC design, fast simulators are usually necessary. It is worth noticing that the statistical characterization of macro-blocks, such as memories, is significantly more difficult as it is necessary to consider also the parameter spread within the block itself.

In contrast, standard cell logic can be characterized in two steps. The first step is the traditional characterization of the standard cell library both for timing and power. The second step involves characterizing each library component with statistical techniques (such as Monte Carlo) that can capture the impact of variability on timing (and eventually energy). The final step is the merging of the statistical properties of the various library cells into a common format representation of statistical energy and timing. Statistics at the level of cell library needs still to be propagated up for post-synthesis analysis of standard-cell based designs. After both the macro-blocks and the standard cell based design have been statistically characterized, their statistical properties can be integrated into a unified representation at the system level (shown in the top part of). When energy must also be considered, the correlations between timing and energy consumption can be propagated through the different levels of abstraction.

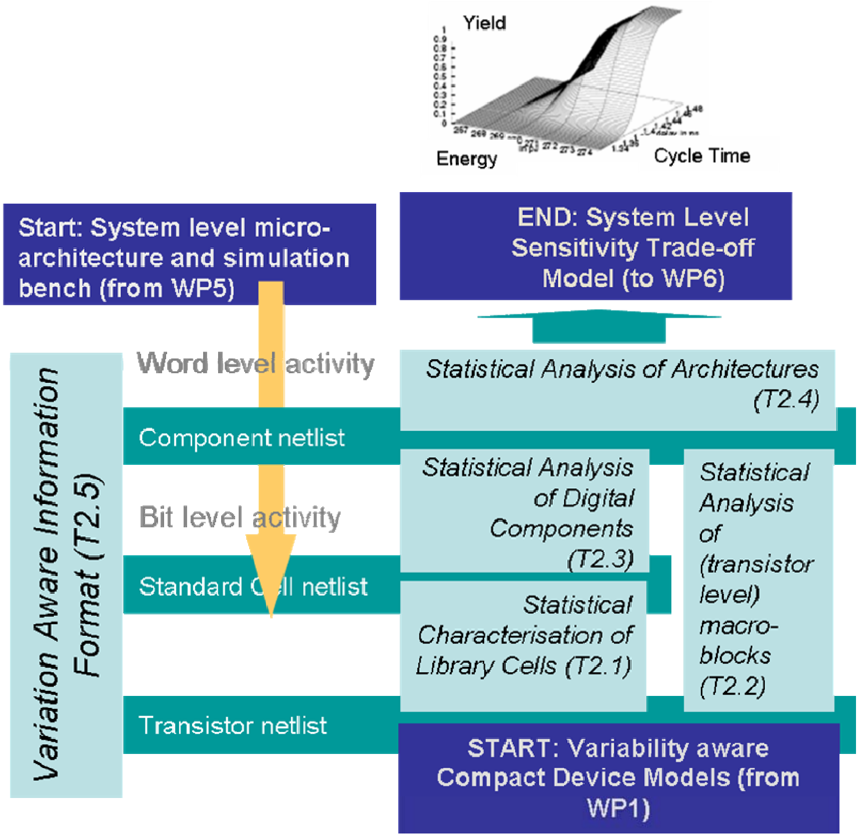
Yield
Energy
Cycle Time
Start: System level micro-architecture and simulation bench (from WP5)
END: System Level Sensitivity Trade-off Model (to WP6)
Variation Aware Information Format (T2.5)
Word level activity
Component netlist
Bit level activity
Standard Cell netlist
Transistor netlist
Statistical Analysis of Architectures (T2.4)
Statistical Analysis of Digital Components (T2.3)
Statistical Characterisation of Library Cells (T2.1)
Statistical Analysis of (transistor level) macro-blocks (T2.2)
START: Variability aware Compact Device Models (from WP1)

Figure 1: Overview of a bottom-up analysis/simulation flow for percolating sensitivities to intrinsic variability from the device level to the system level, across the existing electronic design abstraction levels. The input of the flow is a representation of variability by means of a device compact models resulting from WP1, and the output is a cost model of the complete system for yield versus timing versus energy trade-off analysis.

Particularly important in this WP is standardization of the interfaces to transfer this information through different levels of abstraction. Variability propagation from process technology to analogue simulations can be done in a number of ways and no standard is emerging yet in the industry. Similar considerations apply for the interface between standard cell calibration and digital simulation. Existing SSTA solutions use proprietary standards for the cell library characterized for variability. Therefore, a unified format is developed for the analysis tool, which can capture the IP component energy and timing statistics.

Finally, this WP interacts with other key work packages of this project proposal. Specifically it absorbs inputs from WP1 on variation-aware device compact models, representative of the technology nodes addressed in this project (e.g., 45nm, 32nm), and from WP5 for the system-level micro architecture. On the other hand, it creates outcomes for WP6 for evaluation and benchmarking in terms of a variation-aware sensitivity model for trade-offs between performance, energy and yield of the overall system.

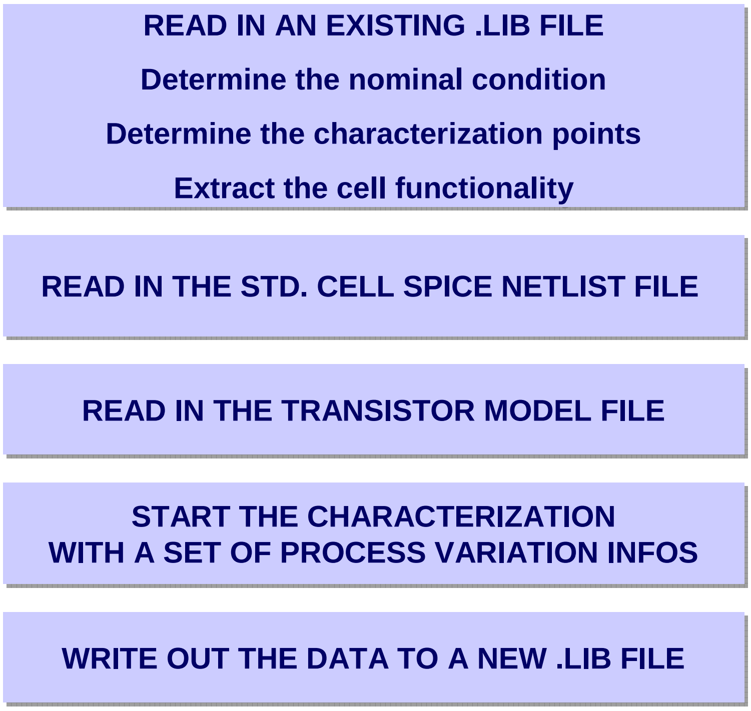
READ IN AN EXISTING .LIB FILE
Determine the nominal condition
Determine the characterization points
Extract the cell functionality
READ IN THE STD. CELL SPICE NETLIST FILE
READ IN THE TRANSISTOR MODEL FILE
START THE CHARACTERIZATION
WITH A SET OF PROCESS VARIATION INFOS
WRITE OUT THE DATA TO A NEW .LIB FILE

Figure 2: Statistical characterization flow for standard cell libraries.

11. Statistical Characterization of Standard Cell Libraries

The objective of this activity was to develop analysis and simulation techniques for statistical characterization of standard cell libraries.

The increasing impact of process variations on design performance, particularly of the within-die (WID) random variations is becoming more and more critical in advanced nanometer technologies, at 65nm and below, as it was demonstrated by silicon characterization and variability analysis performed on specific test structures and during at-speed testing on several fabricated products. Statistical static timing analysis (SSTA) is a promising technique to deal with process variability on large multi-million gate System-on-Chip (SoC) designs, to decrease the pessimistic design margins based on worst-case PVT corners that are progressively reducing the benefits of technology scaling especially at 32nm and below, to improve the performances, and to increase the parametric yield. However, SSTA requires a standard cell model that accounts for the process parameter variations. Therefore, statistical characterization is a critical enabler for SSTA-based variability analysis, and an efficient and accurate statistical characterization methodology for large standard cell libraries must be developed. In particular, the challenges addressed in this Task were:

- Better accuracy to account for global, systematic, and random variations, where transistor-level details must be exposed to the characterization tool;
- Characterization time due to an increasing number of characterization grid points and library cells, larger number of Power Supply/Temperature corners (NBTI effect), and a huge number of circuit simulations;
- Large file sizes that are difficult to handle/transfer.

All timing quantities have been characterized with the process variations based on the linear sensitivity technique (Figure 3), which even if on few timing arcs of some specific library cells may be slightly inaccurate with respect to higher-order modeling techniques such as quadratic delay modeling, nevertheless it is the only practical approach that can be used

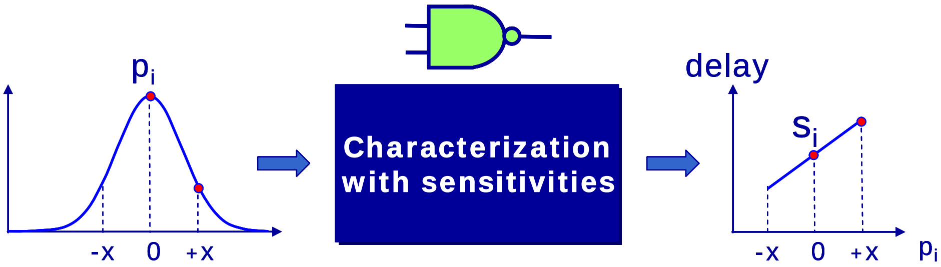

Figure 3: Standard cell statistical characterization with linear sensitivities.

today to statistically characterize large cell libraries. A statistical characterization flow illustrated in Figure 2 has been developed to characterize a complete set of standard cell libraries in both 65nm and 45nm CMOS low-power technology. The flow is currently used in production, and it is important to point out that is fully compatible with the standard digital design flow.

One of the problems addressed in this Task has been the process parameter selection for statistical characterization. Ideally, all the process parameters present in the transistor compact model directly impacting the library cell timing quantities should be used, but with an unaffordable characterization time and file size. We developed a practical yet accurate characterization technique, where only a subset of these parameters are considered, still using the corner values for the other parameters. The parameters that have been statistically characterized are those impacting the library cell propagation delay and input capacitance, such as W, L_{eff}, T_{ox}, β, and V_{th}.

After parameter selection, the statistical device mismatch should be characterized. Since ideally every single transistor within a library cell should be characterized statistically, the characterization time would be too large and speed-up techniques are needed to reduce the run time. Moreover, the transistors impacting the cell performance should also be identified, along with a reduction in the characterization grid size, to evaluate a trade-off between accuracy improvement and characterization effort. After a careful analysis and comparisons against Monte Carlo methods, it was observed that the mismatch impact on performances cancels out on most of the timing critical paths, and it has some impact only on short paths. In fact, with the increase in the number of stages, the σ/μ ratio follows a $1/\sqrt{n}$ relationship, where n is the number of stages along the delay path, as it is illustrated in Figure 4. The σ/μ ratio does not change significantly as the number of stages increase along a path delay extracted from a digital block in 65nm CMOS low-power technology. The green and red plots also show a very good accuracy between statistical timing analysis performed with Monte Carlo transistor-level simulations, and gate-level block-based analysis carried out with an industrial SSTA tool.

The methodology and flow developed in this Task for an efficient and accurate statistical characterization of industrial standard cell libraries have been exploited to characterize a complete set of cell libraries in the relevant Power Supply/Temperature corners in 65nm and 45nm CMOS low-power technology. The cell libraries are available and ready for exploitation, and can be used for the statistical timing analysis of digital blocks.

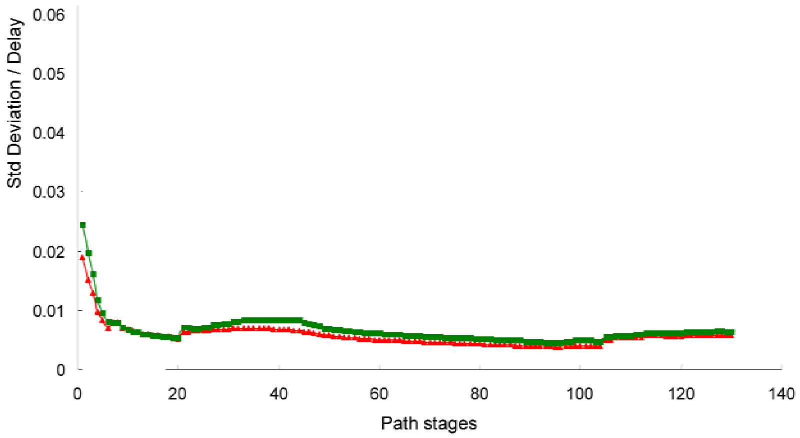

Figure 4: Mismatch impact on path delay.

In order to provide a “Gold Standard” reference for Statistical Standard Cell characterization we have developed a ‘pure’ Monte-Carlo simulation methodology that can be used to accurately simulate standard cells in the presence of variability. The detaied comparison between the fast cell characterization method developed and the golden standard will be subject to evaluation in deliverable D2.3. Underlying, the UoG simulation methodology is a characterization test bed (shown in Figure 5), which is automatically generated for the cell under test, based on a description of the input and output nodes required. Each input of the cell under test is buffered by a chain of 4 input inverters, this ensures that the input signal to the cell is realistically smooth and allows various input slews to be generated by varying the drive strength of the inverters. The cell is loaded with another output inverter, whose sizing and therefore load capacitance can also be varied. Power consumption can be measured through a separated supply to only the circuit under test.

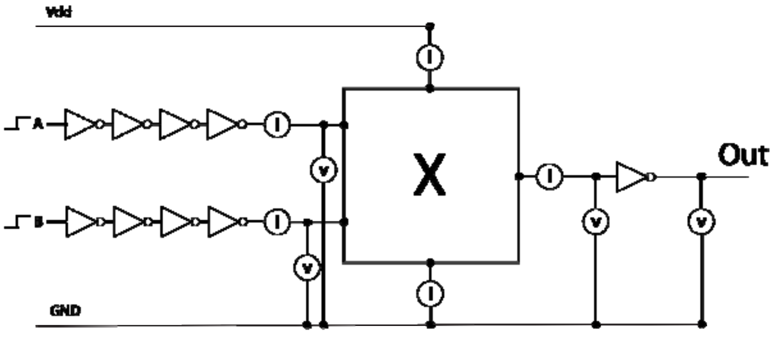

Figure 5: The characterization test bed. Test beds are automatically generated with arbitrary numbers of inputs and outputs, based on a description of the contained cell.

Having auto-generated the test bed for the specific conditions required, the circuit is simulated using large compute cluster resources available at UoG. Typically, 250+ CPUs are utilized, for simulation ensembles containing 10000-50000 instances of circuits composed of devices, randomly selected, from sets of calibrated MOSFET models containing the effects of intrinsic parameter fluctuations. In order to ensure that all capacitances in the simulated system are fully charged and discharged, and that the measured power consumption is an accurate representation of the average behavior of the circuit, all combinations of input transitions are applied to the inputs as can be seen in Figure 6.

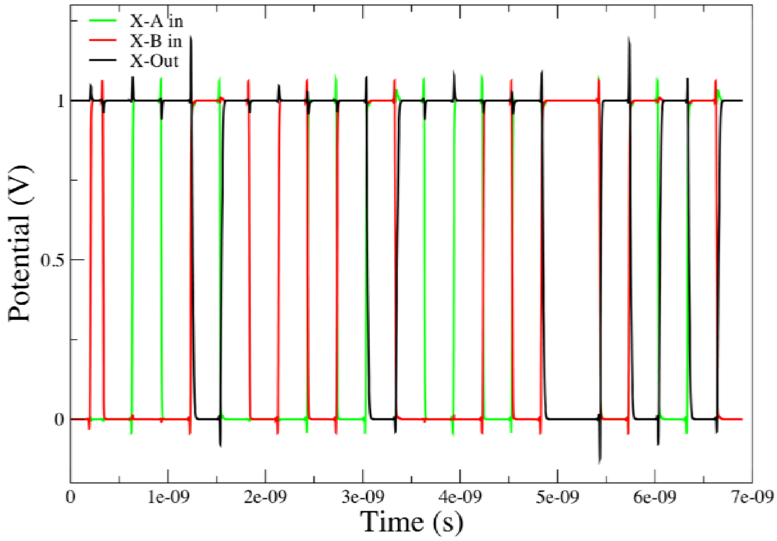

Figure 6: All combinations of input transitions are applied to the input buffers.

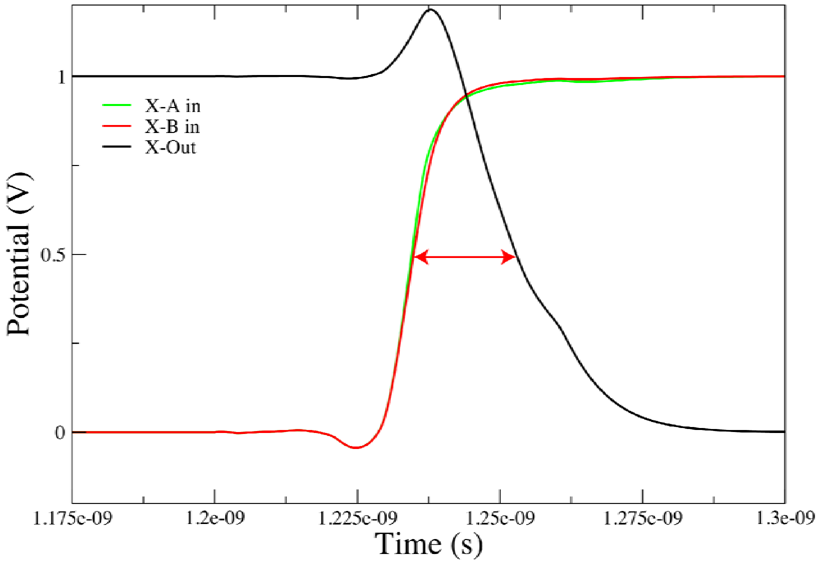

Figure 7: 50-50 Delay Times are extracted from the first switching input to the last switching output.

In this case delay/propagation times are defined as the time between 50/50 points of the first input to switch and the last output to change, as illustrated in Figure 7. Other definitions of delay e.g. 20/80 10/90 can be used. Input switching is used to define an 'event', this may or may not result in a delay measurement since only those input vectors which result in an output transition can be used to measure a propagation delay. However, these events can be used in order to measure the average energy required per input vector transition. Since not all input vectors cause an output transition there is a significant difference in the power consumption of these two types of event. However, both must be taken into account in order to calculate the average energy that is consumed by the circuit when active. Having, defined a switching event, the dynamic energy consumed is calculated by integrating the test circuit supply current over the time where the cell is active, as shown in Figure 8. The quiescent or leakage power can then be calculated for all combinations of input vector states by time averaging the supply current over the points within the simulation which are not contained within switching events.

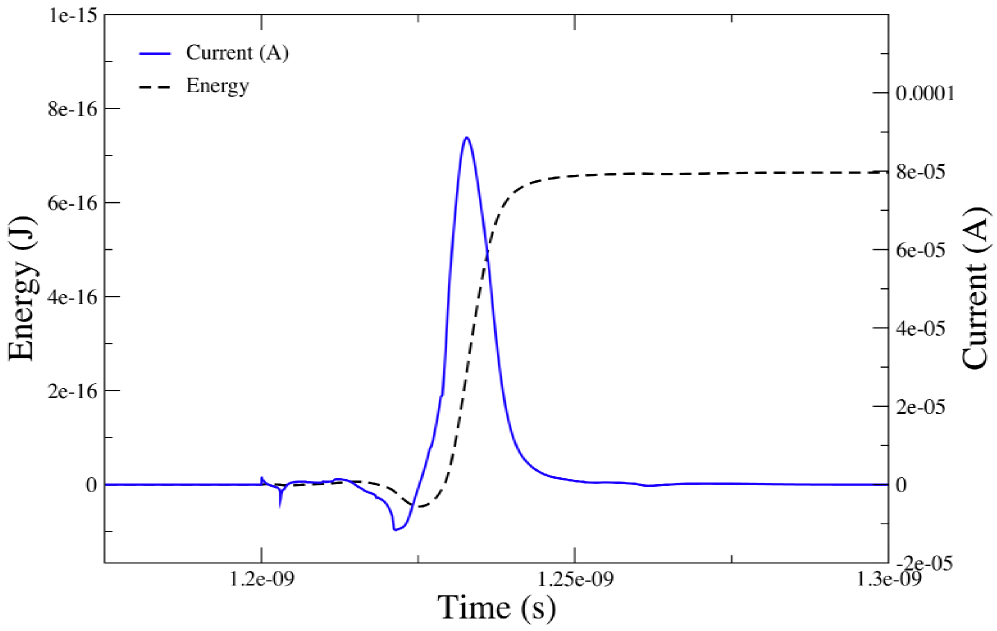

Figure 8: Energy per transition is calculated by integrating over the time that the test circuit is active. Leakage Power is the time averaged power consumption over times when the test circuit is inactive.

For demonstration purposes we have simulated an ensemble of 50,000 NAND gates based on a design published by IMEC. An ensemble of MOSFET models representative of the 65nm technology node, containing variability introduced by Random Distributed Dopants, are used to replace the standard uniform models.

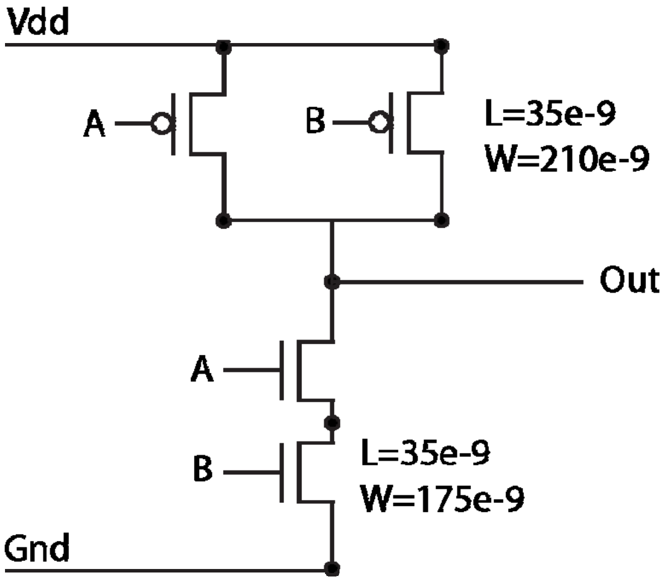

Figure 9: IMEC NAND gate circuit under test. 35nm Bulk MOSFET developed at UoG are used. These models contain intrinsic parameter fluctuations from random dopants.

Figure 10 and Figure 11 show the Energy per transition and leakage power as a function of the propagation delay, calculated from the Monte-Carlo simulation 50,000 cell instances. Figures are presented for the fastest and slowest switching input vectors and for the average delay vs. average power/energy consumption. This generation ensemble required approximately 25000 CPU hours on a large compute cluster and generated ~100Gb of simulation data when compressed.

By utilizing this accurate, but extremely computationally expensive, simulation methodology we aim to provide a standard reference methodology by which other statistical simulation techniques may be judged and accurate estimates of the impact of intrinsic parameter fluctuations on standard cells may be made.

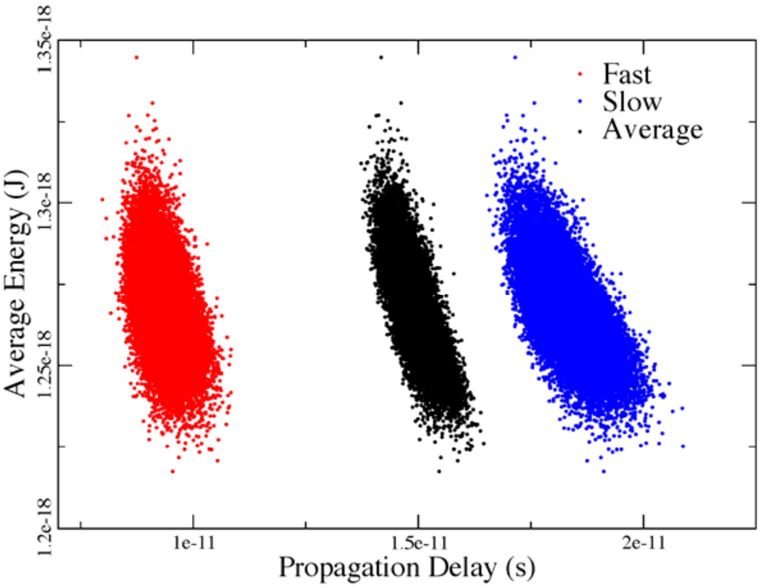

Figure 10: Average Energy Per Transition Vs. Propagation Delay. Distributions for the fastest and slowest switching input vectors are shown as well as the average value for the cell.

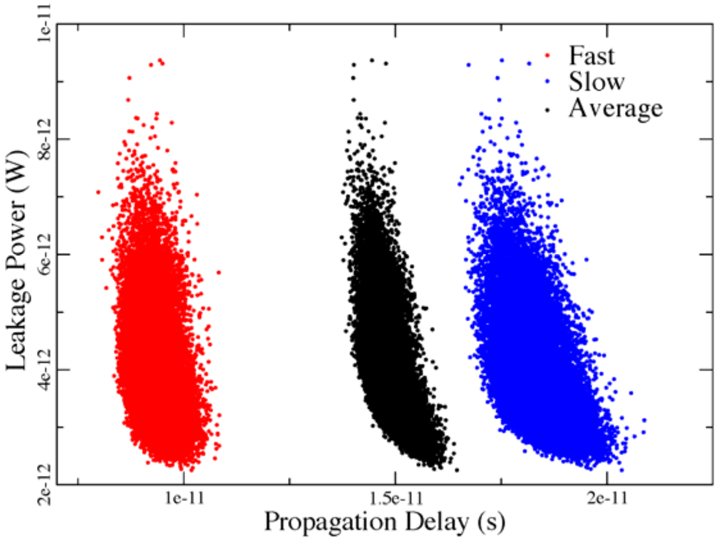

Figure 11: Leakage Power Vs Propagation Delay. Distributions for the fastest and slowest switching input vectors are shown as well as the average value for the cell.

12. Statistical Analysis of Digital Blocks

The objective of this activity has been to develop analysis and simulation techniques for statistical timing analysis of digital logic blocks, taking as input data the circuit gate-level netlist, and the statistical standard cell libraries characterized in Task 2.1.2 (see Section above).

The target digital block was ILP, a mobile processor for imaging applications designed by the Imaging Division of STMicroelectronics, of about 370K nets with the fastest clock running at about 322MHz, implemented in 65nm CMOS low-power technology. It is worth pointing out that ILP is an advanced design included into a real product. As such, it is a very relevant test case to be used in this Task.

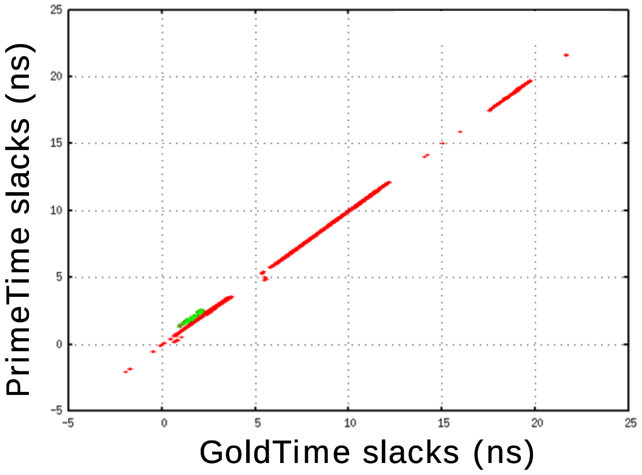

Figure 12: PrimeTime vs. GoldTime

Since the purpose of this activity was to develop and evaluate a methodology for variability analysis of a digital block (and not developing a new SSTA tool), we used the statistical timing analyzer GoldTime developed by Extreme DA. First, we compared the accuracy of GoldTime's delay calculation against Synopsys' PrimeTime, which is the reference tool for static timing analysis and sign-off. We obtained a very good correlation between the delay calculation performed with GoldTime vs. PrimeTime, as it is demonstrated in Figure 12, where the slacks computed with GoldTime show an excellent accuracy with respect to the slacks obtained with PrimeTime.

After this preliminary correlation activity, we performed SSTA on ILP, using the statistical libraries characterized in Task 2.1.2 (Section above), also considering the impact of mismatch and analyzing the potential benefits of including mismatch in statistical timing analysis against the large run time required for mismatch cell library characterization.
The results obtained on ILP demonstrated that by using SSTA it is possible to remove some pessimism with respect to the corner-based analysis (which is the current sign-off methodology).

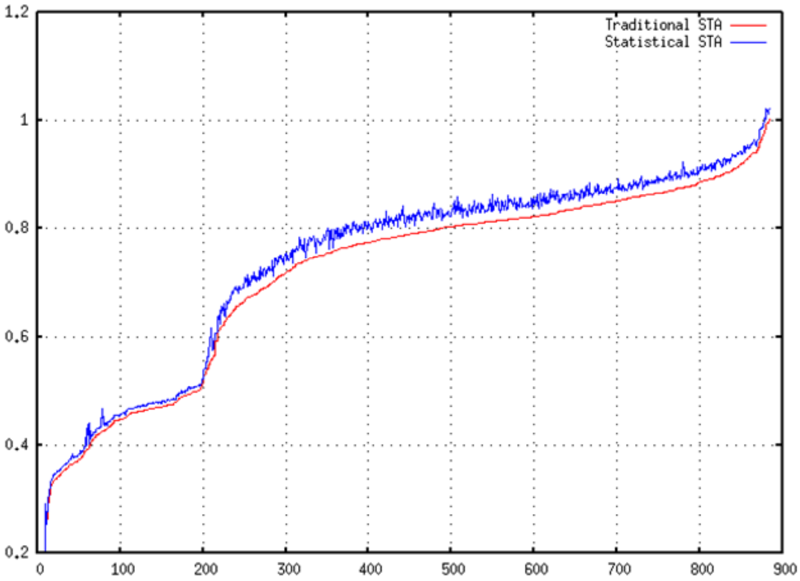

Figure 13: STA vs. SSTA - results on ILP

Figure 13 shows the slack distribution obtained with traditional STA (red plot) against the distribution obtained with SSTA (blue plot). The oscillating behavior of the blue plot confirms a different path ranking generated by SSTA, and it also demonstrates that by using statistical timing analysis it is possible to remove some of the pessimism introduced by corner-case analysis.
This result is even more interesting since it was obtained considering only the lot-to-lot global variations instead of the WID random variations and it clearly demonstrate the process variations impact on performances even at 65nm. By considering the process random variations and their correlations it will be possible to potentially achieve a more significant pessimism removal.

Moreover, we also performed SSTA on ILP considering device mismatch, and as it was discussed previously, this impact was not very relevant, as shown in Figure 14 (green plot). Hence, we have to carefully trade off the small accuracy increase against the huge mismatch characterization time. Since mismatch can be accurately taken into account in SSTA, we may not consider the de-rating factors typically used for bounding the on-chip variations (OCV), and the timing constraints should be modified accordingly.

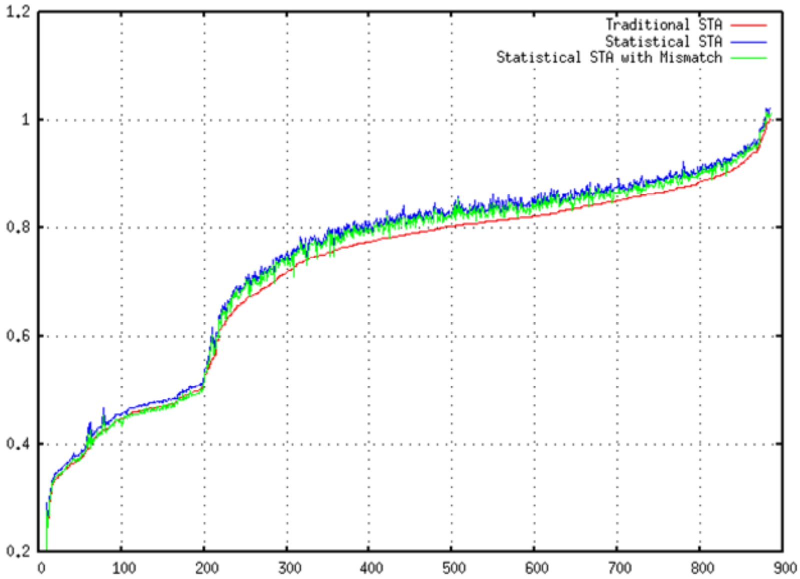

Figure 14: SSTA with device mismatch - results on ILP

By comparing SSTA against the traditional static timing analysis (STA) we obtained different criticality on a few paths. This behavior means that paths are sorted differently when their slacks are computed with SSTA instead of STA, and the SSTA 3σ slacks are generally larger than STA slacks (also when considering the mismatch impact), even if the gain does not seem impressive since we used the same distributions provided in the 65nm SPICE models (capturing the lot-to-lot global variations, but not the WID variations).

A very relevant result obtained with SSTA was that a few timing paths had the 3σ slack value smaller than their corresponding worst-case corner value. In this case, we may have potential timing violations that were not identified by traditional STA, thus causing timing failures during at-speed testing. The undetected timing violations are visible in Figure 15, which shows a zoom-in of the slack distributions in Figure 14.

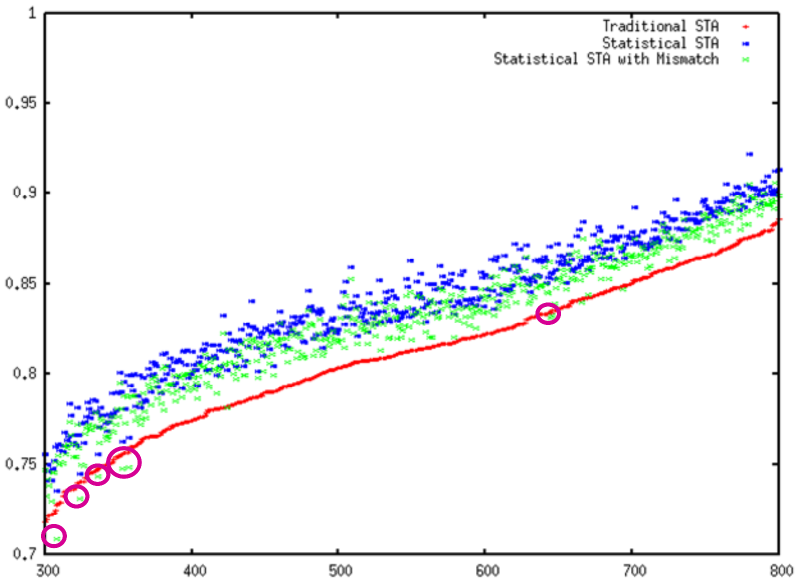

Figure 15: Undetected potential timing violations determined by SSTA

Finally, we started the variation-aware parasitic extraction, in order to include also interconnects into SSTA. We considered 43 interconnect parameters, and we obtained a very good correlation around Worst-, Typical-, and Best-case corners, against Synopsys' StarRCX, the reference tool used for parasitic extraction. The next step will be to include the statistical interconnects into SSTA.

It is worth to point out that our SSTA flow is fully compatible with the standard flow for static timing analysis and sign-off, as it is depicted in Figure 16, where the input data for the SSTA tool are compatible with traditional STA tools, except for the statistical libraries and process variation information.

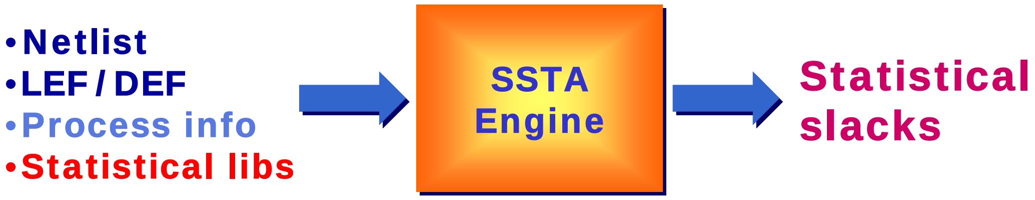

Figure 16: SSTA flow

13. Statistical Characterization of Macro-Blocks

The objective of this activity is to develop analysis and simulation techniques for statistical characterization of macro-blocks (i.e., eSRAMs, Register Files, etc.), having as input data the transistor compact model and the transistor-level netlist of the macro-block. In essence the goal is similar as in Section 11, however given the complexity of these macro-blocks we need to develop special characterization techniques to avoid an explosion in CPU characterization time. For instance, a full blown simulation of the complete transistor level netlist of a SRAM is intractable, much less therefore it is their statistical characterization using such brute force purely-based simulation techniques.

Since the statistical characterization of macro-blocks such as eSRAMs is more complicated than digital standard cell libraries, we considered the tool Extreme DA's ROAD.
First, we evaluate the tool compatibility with the eSRAM flow. ROAD calculates the parameter linear sensitivities by perturbing the parameter to its upper bound and measuring the difference in performances after circuit simulations. It is the sensitivity-based analysis that allows fixing those parameters that are not relevant with respect to performances (it can be done with all parameters, mismatch, design, within-die, operating).
ROAD creates a quadratic response surface model (RSM), and second-order polynomials are generated. The number of required simulations is known in advance and depends on the number of free variables, fixed corners, and accuracy, but not on the number of performances (cost functions or constraints).
In case of strong performance non-linearity against a design variable it is possible to tighten the range of perturbation of the parameter.

However, if the non-linearity depends on process parameters this is not possible and the model is not reliable. Therefore new developments for Monte-Carlo based SRAM macro-blocks statistical analysis were also carried out on 45nm technologies. An in house comprehensive solution to evaluate how the statistical variation impacts the yield is under development. First Yield forecasts have been done on the READ cycle. The aim of this work has been to understand the impact of the variations on our designs and evaluate the needs of new development methods.

In parallel a direct evaluation of the yield of the array towards read and write cycle can be carried out. A simple model of the system needs to be considered. The figure below shows a sketch of the considered system for the read mode.

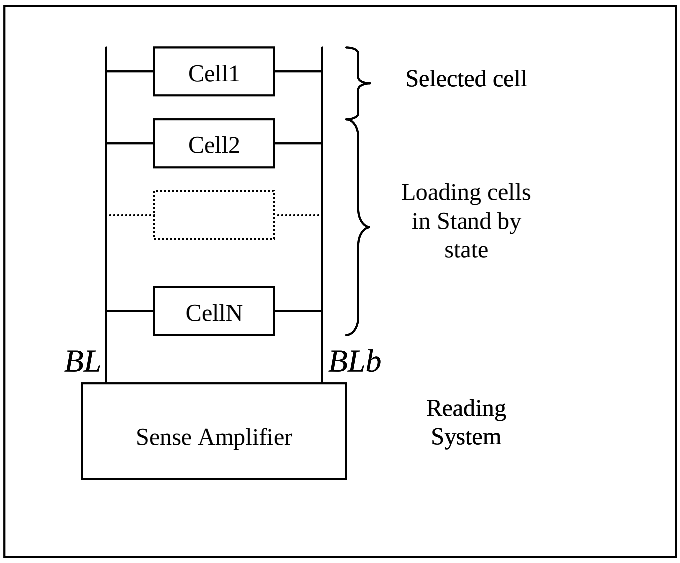

Figure 17 Sketch of a SRAM for read

N cells are linked to one sense amplifier. The sensitivity of the sense amplifier can be evaluated by forcing the differential voltage between BL and BLb and performing Monte Carlo simulations. The failure probability as a function of the differential voltage can be extracted. A Gauss distribution provides a good analytical modeling of the sense amplifier sensitivity (figure below)

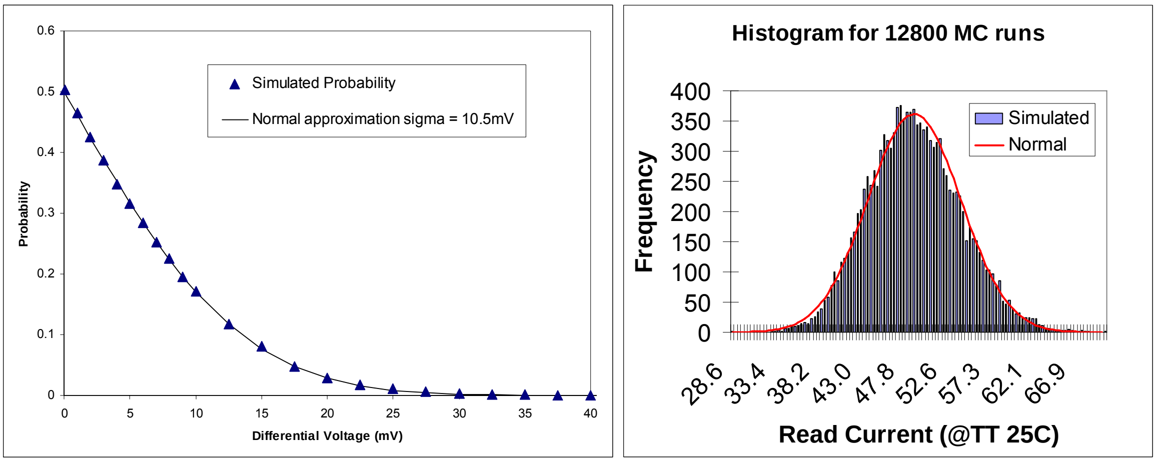

Figure 18: left: sense Amplifier sensitivity, right: read current statistics

The next interesting variable to evaluate when dealing with read margin is the Read current of the bitcells. Monte Carlo simulations allow defining the statistics of this read current in different conditions. A first approximation shows a gauss distribution for the read current. As a consequence, a Gumble distribution can model the worst case current out of N cells. Using this distribution as analytical modeling allows getting the Yield of the SRAM array by the following equation:

$$Yield\left(t_{sense}\right) = 2\cdot\int_{v=0}^{+\infty} p_{SA}\left(SA\_mismatch = v\right)\cdot P_{Ncells}\left(Iread > \frac{v\cdot C_{BL}}{t_{sense}}\right)\cdot dv$$

Where t_{sense} is the time before starting the sense amplifier, p_{SA} is the probability for the sense amplifier to discriminate a voltage v and P_{Ncells} is the cumulative probability for the read current to be higher than a given current I (a Gumble distribution).

We finally obtain the yield as a function of the read timing.

A similar work is now being carried out to define a good analytical model for the write mode of the SRAM arrays.

14. Statistical Analysis of SoC Architectures

A SoC comprises high-level components such as processors, memories, accelerators, etc. The large majority of SoCs contains well defined register boundaries between any of these high-level components. This is not only needed for easy plug&play IP-level and architectural trade-offs but also to alleviate timing closure during the physical design phase. Even when the SoC is optimized across block boundaries during synthesis, these registers remain still in the synthesized netlist.

We can therefore formalize the SoC critical path delay to be the maximum of any the critical path of the combinational logic islands between registers, and this irrespective of their interconnection pattern: parallel, serial, tree-like etc. This assumption is also applicable to each of the multiple voltage islands of a complex SoC even when each is operating each at different frequencies. The slack available in timing for each of the islands refer to the difference between their statistical longest path delay and the cycle-time of the frequency associated.

The SoC power / energy consumption is the sum of the power / energy consumption of the individual components. Here a distinction can be made between leakage power and dynamic power consumption. When treated independently, it is better to use power to measure leakage and energy for dynamic. Dynamic power for the IP blocks is obtained via detailed gate-level simulations starting from a RTL test bench of the system that percolates activity related information (such as toggling count) down to to the inputs of the blocks via RTL simulation; and in a second step to the gates of the IP block via detailed gate level simulation. To capture the indirect impact of timing variations into unwanted

Dynamic energy is independent on the clock frequency, likewise for leakage power which is also independent of the selected clock frequency. Dynamic energy only depends on the operation performed by the circuit namely, the average number of times a particular component is being activated during operation of the SoC for a given task and the energy consumed by the component for each of the operations performed. This energy can be transformed to dynamic power dissipation only when an assumption is made on the operating clock frequency.

Having concluded that the statistical critical timing of the SoC is the maximum of the timing of any of its components, and the statistical energy (for dynamic) and power (for leakage) is the sum of their individual energy/power, we can formalize the statistics of the SoC in base of two stochastic properties: (a) The Cumulative Density Functions (CDF) of the maximum value of two stochastic variables is the product of their CDF; (b) the Probabilistic Density Functions (PDF) of the sum of two stochastic variables is the convolution of their PDF.

Let's assume an SoC *S* with two sub-systems *1, 2* each with a joint longest path delay and energy/power stochastic variables: (T_1, E_1) and (T_2, E_2); with joint CDFs: $F_1(t,e)$ and $F_2(t,e)$; and joint PDFs: $f_1(t,e)$ and $f_2(t,e)$ (where $F(x)=\int f(x)\, \mathrm{d}x$).

We could compute the stochastic longest path delay of the SoC (T_S) as:

$$P\ [\mathrm{Max}\{T_k\} <= t] = f_S(t) = f_1(t) \times F_2(t) + F_1(t) \times f_2(t)$$

We could also compute the stochastic energy/power of the SoC (E_S) defined as:

$$P\ [\mathrm{Sum}\{E_k\} <= e] = f_S\ (e) = \int f_1(e-a)) \times f_2(e)\ \mathrm{d}a$$

However propagating the statistical properties of the SoC sub-systems to the system integration level comprises simultaneously propagating a joint probability density function for both critical path and the energy/power variables. At the SoC level we are mostly interested in trade-offs between the two, hence their correlations must also propagate simultaneously.

This is achieved by simultaneously applying a product-convolution transformation in the two joint statistical variables:

$$\begin{aligned} f_S(t,e) = \int\int & f_1(t,e-a) \times f_2(b,a) \\ & + f_1(b,e-a) \times f_2(t,a)\ \mathrm{d}a\ \mathrm{d}b \end{aligned}$$

The product-convolution transformation is a pair-wise operator. However it has the property of being commutative and associative. Therefore, applying the transformation to a SoC comprising more than two components e.g. three components would simply require to apply it to two of the components first and the result to the third of the components.

We have applied the variability analysis flow mentioned to a wireless protocol processor and characterized at the gate level. We have proceed for the whole processor at once (all 120 Kgates), and for each of its five pipeline stages one at a time. The purpose is to compare the statistics obtained when simulating the whole processor against the ones obtained by characterizing each of the stages and then using the product-convolution to obtain the full processor statistics. For this we have used the joint statistics between longest path delay and dynamic energy and leakage power respectively. Worth to mention that the five stages of the processor have been simulated considering the actual load and driving conditions of each stage as present in the complete processor netlist. This is done by reporting an `.sdc` (Synopsys Design Constraints) file for each of the stages from the hierarchical netlist report.

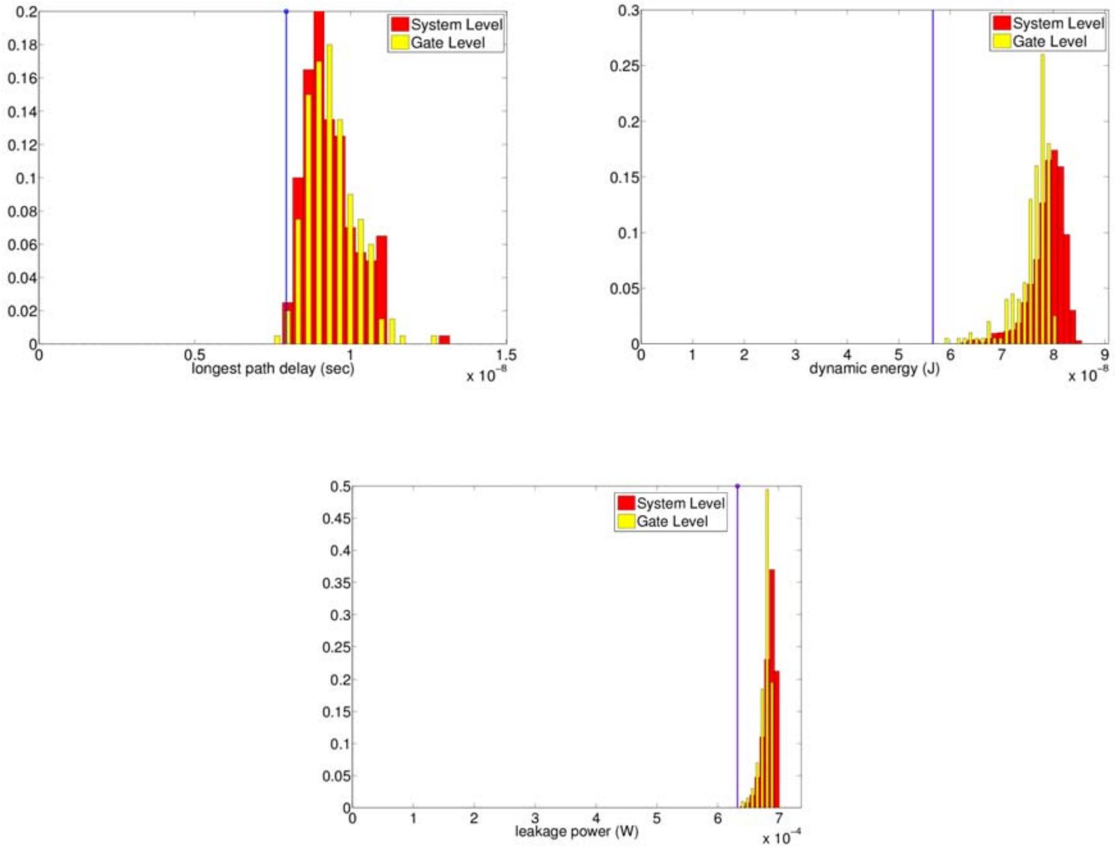

Figure 19: Comparison between full gate-level analysis and SoC-level analysis for the VLIW processor: (a) PDF longest path delay; (b) PDF dynamic energy; (c) PDF leakage power

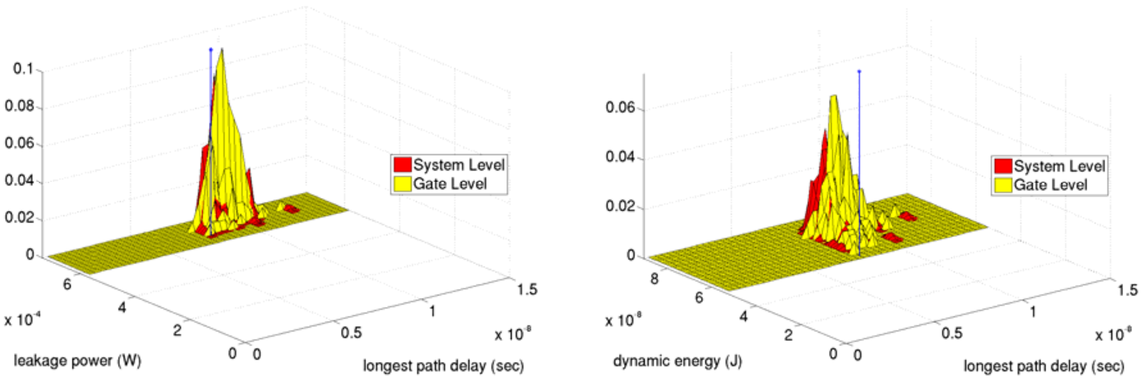

Figure 20 Comparison between full gate-level analysis and SoC-level analysis for joint PDFs: (a) dynamic energy and longest path delay; (b) leakage power and longest path delay.

The results are plotted in Figure 19 and Figure 20. The label *gate-level* refers to the results obtained when analyzing the complete processor at the gate-level while the label *System-level* refers to analyzing each of the processor stages and percolating their statistics using the product-convolution transform to obtain the full processor behavior. The Figures show the good accuracy achieved by the product-convolution technique. This technique "opens the door" for breaking up the complexity bottleneck of gate-level statistical analysis techniques for multi-million gate designs by allowing a divide-and-conquer analysis approach.

The advantage of this approach is that it does not require assumptions on the nature of the statistical distributions involved (e.g., Gaussians, Log-Normal or alike) and it can be applied to any form of statistical data e.g., from close form PDFs to histograms of plain data samples (as in our case).

15. Conclusions

In this deliverable we (STM, IMEC, ARM and UoG) have designed a process variability analysis framework that brings commercial and academic tools into a holistic analysis/simulation flow. Commercial EDA products for process variability analysis such as Statistical Static Timing Analysis (SSTA) have been be reused in combination with Monte-Carlo based simulation techniques there were statistical analysis techniques are missing. This comprises novel techniques to bring correlated statistical timing and power and/or energy metrics from the level of the IP block to the SoC integration level (IMEC). For power and energy estimation a given Register Transfer Level benchmark is used to properly characterize the system. The level below system architecture differentiates between IP components built using standard cell logic and macro-blocks (e.g. memories) that exist as transistor net-lists or layouts from which the netlist can either be extracted or generated via specific compilers (e.g., memory compilers). For standard cell logic existing commercial offerings for statistical timing analysis are used. In case of IP macro-blocks, however, timing and energy are usually characterized using transistor-level simulations and given the increasing complexity of the typical macro-blocks in SoC design, there are no commercial solutions available. Therefore new developments for Monte-Carlo based SRAM macro-blocks statistical analysis were also carried out on 45nm technologies. An in house (ARM) comprehensive solution to evaluate how the statistical variation impacts the yield is under development. First Yield forecasts have been done on the READ cycle. The aim of this work has been to understand the impact of the variations on our designs and evaluate the needs of new development methods. In contrast, standard cell logic follows the classical characterization both for timing and power and each library component is later characterized with statistical techniques, such as Statistical Static Timing Analysis (STM) and/or Monte Carlo (IMEC, UoG) to capture the impact of variability on timing (and eventually power/energy). The final step has been the merging of the statistical properties of the various library cells into a common format representation of statistical energy and timing. Statistics at the level of cell library needs still to be propagated up for post-synthesis analysis of standard-cell based designs. After both the macro-blocks and the standard cell based design have been statistically characterized, their statistical properties can be integrated into a unified representation at the system. When energy must also be considered, the correlations between timing and energy consumption can be propagated through the different levels of abstraction.

16. References

[i] A. Asenov, A. R. Brown, J. H. Davies, S. Kaya and G. Slavcheva, “Simulation of Intrinsic Parameter Fluctuations in Decananometer and Nanometer-Scale MOSFETs”, IEEE Trans. on Electron Devices, Vol.50, No.9, pp.1837-1852 (2003)

[ii] M. Pelgrom et al., ``Matching properties of MOS transistors'', IEEE JSSC, vol. 24, iss. 5, 1989.

[iii] B.Dierickx, et. al ``Propagating Variability from Technology to System Level'', Intl. Wrkshp. on The Physics of Semiconductors, Bombay, 2007

[iv] M.Miranda, et. al, “Variability Aware Modeling of SoCs: from device variations to manufactured system yield”, IEEE International Symposium on Quality Electronics Design, San Jose, March 2009

[V] C. Forzan and D. Pandini, “Statistical Static Timing Analysis: A Survey,” Integration, the VLSI Journal, accepted for publication in press.