MODERN

MOdeling and DEsign of Reliable, process variation-aware Nanoelectronic devices, circuits and systems

Reference: ENIAC-120003 MODERN

Work Package: 3.2
Deliverable: D3.2.2

Standardized PV-aware tools for simulation of digital blocks, AMS&RF blocks, and NVM circuit blocks

Editor: Igor Loi
Co-author(s) / Acknowledgement: Luca Benini
Status: Approved
Date: 22-April-2011
Confidentiality Level: Public
ID number: MODERN-2011-IL01

Revision history:

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.0</td>
<td>22-April-2011</td>
<td>Luca Benini</td>
<td>Approved</td>
</tr>
<tr>
<td>V0.1</td>
<td>01-April-2011</td>
<td>Luca Benini</td>
<td>First Draft</td>
</tr>
</tbody>
</table>

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.
Contents

1. Abstract ........................................................................................................................................... 3
2. Introduction ....................................................................................................................................... 4
3. (UNBO) Heuristic FBB Allocation Algorithm ....................................................................................... 9
3.1. Case Study .................................................................................................................................... 12
3.1.1. Experimental setup ................................................................................................................... 13
3.1.2. Fine-grained FBB vs. Full FBB .................................................................................................. 13
3.1.3. OCV Timing analysis ............................................................................................................... 14
4. (UNGL-NMX) General Introduction for UNGL and NMX section ...................................................... 16
4.1. UNGL ......................................................................................................................................... 18
4.1.1. Introduction ............................................................................................................................. 18
4.1.2. RandomSpice .......................................................................................................................... 18
4.1.3. Test Bed Device ....................................................................................................................... 19
4.1.4. Compact Modelling Strategy ..................................................................................................... 20
4.1.5. Variability Simulations ............................................................................................................ 22
4.1.6. Block Level Variability ............................................................................................................ 24
4.1.7. Yield Analysis .......................................................................................................................... 26
4.1.8. DIFF0 Analysis ....................................................................................................................... 28
4.2. NMX ........................................................................................................................................... 30
4.3. References ................................................................................................................................... 32
5. (UNCA) Comparative Analysis of Yield Optimized Pulsed Flip-Flops ................................................... 34
5.1. Pulsed Flip-Flop Topologies and Simulation Methodology ............................................................... 34
5.2. Circuit-Level Techniques to Improve Parametric Yield ..................................................................... 37
5.3. Post-layout comparative analysis .................................................................................................. 41
6. Self-Repairing SRAM Architecture to mitigate the Inter-die Process variations at 65nm Technology .... 44
6.1. The Implemented Mitigation technique ......................................................................................... 44
6.2. Results and Discussion .................................................................................................................. 46
6.3. References ................................................................................................................................... 48
7. NXP: Introduction ............................................................................................................................ 51
7.1. References ................................................................................................................................... 52
8. Investigation: .................................................................................................................................... 53
8.1. Design Database: .......................................................................................................................... 54
8.2. VCD File Generation: .................................................................................................................. 54
8.3. Flow of the Activity & Coordinate Analysis .................................................................................. 55
8.4. Input Formats ............................................................................................................................... 59
8.4.1. Path Delay Test Patterns: ......................................................................................................... 59
8.4.2. Stuck-At Test Patterns: ............................................................................................................... 60
8.4.3. Application Mode Input Patterns: ............................................................................................... 61
8.5. References ................................................................................................................................... 62
9. Results ............................................................................................................................................... 63
9.1. Activity Profile in time ................................................................................................................... 63
9.1.1. Path Delay Test Pattern, Low Activity File (scl1) ....................................................................... 63
9.1.2. Stuck-At Test Vectors ............................................................................................................... 67
9.2. Comparison of Activity Profiles with Redhawk IR-Drop Profiles: .................................................... 70
9.3. Useless versus Useful Transitions: ................................................................................................ 72
9.4. Activity Modulation in Space ....................................................................................................... 74
10. An Application of the Flow to Simulation Silicon Timing Correlation ................................................ 77
10.1. Simulation Silicon Timing Mismatches: ....................................................................................... 77
10.2. Simple Noise Index Model: ....................................................................................................... 78
10.3. Improved Noise Index Model: .................................................................................................... 83
10.4. References ................................................................................................................................... 85
11. (UNRM) A derivative-free algorithm for circuit design ........................................................................ 87
11.1. References ................................................................................................................................... 91
12. Conclusion and Future Work ........................................................................................................... 92
13. Abbreviations ................................................................................................................................... 93
1. **Abstract**

This report describes some circuit techniques and speed-up algorithms for Process Variation (PV) aware circuit. The aim of these techniques is to tackle post silicon issues due process variability that induce statistical timing variations in some critical paths/cells, while the purpose of the speed-up algorithms is the optimization of circuit design using mathematical optimization techniques.

The Adaptive Body Biasing (ABB) technique has been proposed to reduce the performance impact of Vth variations by playing with body bias voltage regulation. ABB is considered the most promising solution as it introduces limited area and leakage energy overhead compared to other techniques (like adaptive voltage scaling) for a given performance compensation level. A detailed analysis of the influence of random process variations on speed and energy consumption of various classes of flip-flops and SRAM cells is presented. This analysis is performed through extensive Monte Carlo simulations and exploiting the commercial STMicroelectronics 65 nm CMOS technology.

The analysis of speed-up algorithms has been followed in two lines of activities. The first one deals with the optimization of circuit design using mathematical optimization techniques that do not require the evaluation of derivatives, as it is the case in circuit design. Preliminary numerical results have been obtained that are promising of further improvements. The second one deals with the training of learning machines to be used to give surrogate, or behavioural, models of electronic circuits; models to be used in the optimization procedure. A framework has been defined for suitable machine learning in circuit design.
2. Introduction

As a consequence of the scaling of device dimensions to nanometer size, we face a multitude of challenges in designing complex giga-scale systems. Major challenges come from parametric variation such as intra and inter die variation of channel length, oxide thickness, doping concentration [1]. These phenomena have a direct and profound impact on system performance resulting in parametric yield loss and reduced system lifetime. Numerous approaches have been proposed to address these effects and to increase parametric yield as well as tolerance to aging-induced variability. Circuit design optimization techniques target to solve some design problems by describing them in terms of decision variables (continuous or integer), constraints on the variables (physical relationships, budget limitations, performance requirements and so on), and one or more objective functions to be maximized or minimized (yield is an example). In this way the design process is modeled by a complex and large scale optimization problem, often integrated with a simulation tool.

Statistical optimization approaches [2] target process variations as they select optimum design-time parameters (such as Vth) to maximize timing yield. Although effective, these approaches tend to conservatively constrain the design and rely on the accuracy of the statistical characterization of fabrication processes, which may not be known with high precision. As argued in [3], post-silicon tuning can complement and sometimes outperform pre-silicon statistical optimization, as it provides opportunity to tune individual dies to meet timing and other performance constraints. Post-silicon tuning can also address time-dependent variations, such as temperature-induced timing failures [4] NBTI-Induced transistor aging [5].

Common techniques to compensate process variation induced performance deviations during post-silicon testing are: i) Dynamic adaptation of processor voltage; ii) Reverse/Forward body biasing. The first one exploits voltage supply increase to boost the circuit performance, while the second achieves the same effect by reducing the threshold voltage. Applying these techniques in an unconditioned fashion would introduce an unacceptable energy overhead. As a consequence, adaptive strategies have been developed, namely Adaptive Voltage Scaling (AVS) [9] and Adaptive Body Bias (ABB) [10], aimed at applying voltage (either Vdd or Vth) adaptation only when on-line monitoring circuits signal that speed-up is required for maintaining correct functionality at the required performance levels. Moreover, adaptive techniques can react to time and temperature dependent performance variations.

While it has been shown in [8] that adaptive voltage scaling (AVS) requires a much smaller change (percentage-wise) in supply voltage than adaptive body biasing (ABB) requires in threshold voltage to achieve a target frequency boost of a processor core. For this reason AVS in principle has a much milder impact on leakage and is a more power-efficient and thermally compatible solution than ABB, the area and power overhead introduced by the required level shifters between the multiple voltage domains make AVS less attractive than ABB in real implementations.

Indeed, ABB imposes limited overhead for body-bias generation and control [7], thus it is frequently deployed to combat the above mentioned effects which cause circuit timing failure [4, 5, 6] ABB works on the principle of applying reverse bias voltage (RBB)/forward bias voltage (FBB) to increase/decrease the threshold voltage of devices. This results in increasing gate delay while reducing leakage in case of RBB and vice versa in case of FBB. One can trade-off delay with leakage to achieve low leakage or high speed.
Sensing circuits are a challenging block in NVM technology since their requirements in terms of analogue performance are very demanding. Among them: good matching performance in small area for mosfets in the input stage, properties of good device matching not only at minimum spacing but also at long distance, excellent dynamic performance.

Designing complex devices and processes is a very hard task. In this context, using optimization techniques may result in a competitive advantage, as already proved in many industrial activities, see e.g. [14-18]. Many design problems can be described in terms of decision variables (continuous or integer), constraints on the variables (physical relationships, budget limitations, performance requirements and so on), and one or more objective functions to be maximized or minimized (yield is an example). In this way the design process is modeled by a complex and large scale optimization problem, often integrated with a simulation tool.

In this project, UNBO applied ABB technique on digital block (provided by ST) to quantify the potential benefits of the Adapting Body Biasing against the slow down of digital circuits due process variation.

UNGL and NMX, in order to study the effect of PV in sensing circuits for NVM technology, they created an ideal test case by porting a simplified version of a sense amplifier into the 35nm node by keeping the critical mosfets at minimum gate length. This solution can appear quite strange for an analogue design perspective where mismatch is reduced by increasing device size and by avoiding minimum channel length. Nevertheless it is helpful in this study where the statistical model cards are generated by many TCAD simulations to include all possible local variations due to RDD, LER etc. All these local variations can be assumed (at a first order) as independent among the different mosfets so that they can be considered mismatch sources.

Therefore by simulating the sense circuit it is possible, in principle, to study the following aspects:
- compare different methodologies to map the PV into the compact model looking at their results in the sense circuit behavior and verify the accuracy-simplicity trade-off.
- analyze the PV response of the blocks composing the sensing circuit weighted by the entire sense circuit transfer function
- ranking the most critical mosfets which requires particular attention to PV
- Perform a sensitivity of the circuit to the physically based parameters (like threshold voltage, etc) to identify the ones that play the major role to allow process or design solutions.

UNCA evaluated two different circuit-level techniques to mitigate the impact of process variations on pulsed FFs: the transistor reordering [1] and the usage of dual threshold voltage (DVT) transistors [1]. Both these approaches can be applied at design-time without requiring any extra device and architectural modifications, thus they can be easily used also in conjunction with other optimization techniques (such as that proposed in [2]). Experiments, performed on four state-of-the-art pulsed FF topologies, designed using the STMicroelectronics 45-nm 1V CMOS technology, demonstrate that the conjunct usage of the transistor reordering and the DVT techniques is a very effective way to concurrently improve the timing and energy yields. Furthermore, UNCA performed a comparative analysis considering FFs optimized for EDP yield. The impact of layout parasitics on speed, energy and EDP variability was explicitly taken into account for all the analyzed circuits. Moreover, the combined effects of process and environmental variations (power supply voltage and temperature variations) were also comparatively analyzed, with the main objective of identifying the more robust pulsed FF structure.
Moreover, UNCA has evaluated a zone based self-repairing technique, based on adaptive body biasing, which helps in the mitigation of the impact of inter die process variations on SRAM cells. Simulation study has been carried out, exploiting extensive Monte Carlo simulations, on 65 nm 1V CMOS STMicroelectronics technology. Results have been evaluated considering operating temperature spreading from 27°C to 125°C.

NXP analyzed a statistical timing flow based on reduced parameter lib cell models and measurement results from WP5. The achieved results are used as reference for the tool and flow validation.

The activity developed by UNRM within WP 3.2 mainly deals with the optimization of circuit design by means of mathematical programming techniques that do not require the evaluation of derivatives. This is mainly due to the fact that these derivative are not available when using simulation tools or surrogate model for circuit design.

In this Framework, UNRM aim is to develop the best suited optimization procedures for the circuit design activities carried out within WP3, and to provide efficient algorithms for the solution of the corresponding nonlinear programs [11-13].
References:


3. **UNBO) Heuristic FBB Allocation Algorithm**

In this section, we propose a linear heuristic to solve the FBB clustering and allocation problem. Our row-selection algorithm is a linear-time heuristic which solves the allocation problem in a greedy way. The algorithm has two main phases. In the next subsections we will describe each phase in more details.

### A. Phase 1

We have a list of FBB libraries Liblist = \{lib_1, lib_2, lib_3, ..., lib_p\} with different characterization of delay and power for each body bias voltage. The Liblist is sorted from slowest to the fastest library. Also, we have a list of rows: Rowlist = \{r_1, r_2, ..., r_N\}. In a loop we fetch Library lib_j from the Liblist and swap all cells in all rows with the cells from the Library lib_j. Then, we check the timing of all paths to see if the desired speed-up is met or not. For any given design U, and any speed-up factor of \( \beta \) (we define speed-up as the reduction percentage in the critical path delay), if \( D_{\text{crit}} \) is the delay of the critical path in U, the delay after speed-up must be less than \( D_s = D_{\text{crit}} \times (1 - \beta) \). After replacing all cells with cells from Library lib_j, we calculate the delay of all paths and check if all the delays are less than \( D_s \) or not. If so, we have met the desired speed-up and Pass 1 is stopped, otherwise it means we can not obtain the desired speed-up with Library lib_j and we will continue Pass 1 with the next library from the Liblist until we find the suitable library. Note that, each library corresponds to a body bias voltage.

At the end of Phase 1, if we can find a library which meets the speed-up we will continue with Phase 2, otherwise we stop the algorithm since the desired speed-up can not be obtained with the set of FBB libraries Liblist.

If we can find the appropriate Library lib_j in Phase 1, it means we can achieve the desired speed-up and all rows use the cells from Library lib_j. In Phase 2 of the algorithm we try to find those rows in such a way that swapping their cells with the slower libraries i.e. \{lib_0, lib_1, ..., lib_{j-1}\} does not violate the timing but can decrease the leakage power. Therefore, in Phase 2 we will create a few clusters each of which contains a set of rows using the same library (body bias voltage). The number of allowed clusters is a constraint given to the algorithm. After Phase 2 at-least one of the rows uses lib_j and the design still meets timing with the desired speed-up (like outcome of Phase 1), but it has less leakage power than that of created in Phase 1. In the following subsection we will describe Phase 2 of the algorithm in more details.

### B. Phase 2

Phase 2 is based on the fact that there are rows in the design which have gates in the non-critical paths and hence do not need any body biasing or can tolerate a lower vbs voltage (slower library) resulting in reduction of the leakage power. Therefore, at the beginning of Phase 2, we prioritize rows based on their timing criticality. Figure 3-1 shows the row ranking algorithm. To prioritize rows, we examine how many cells each row has in the design timing critical paths. Let \( \Pi \) be a set containing all critical paths whose delays are more than \( D_s = D_{\text{crit}} \times (1 - \beta) \) (lines 1-7), and \( Q_{i,k} \) be the number of cells on row \( i \) and on path \( k \). Then, the timing criticality co-efficient (TC) of row \( i \) is calculated as follows (lines 8-19):

\[ \text{TC}_i = \frac{\sum_{k \in \Pi} Q_{i,k}}{\text{total number of cells on row } i} \]
\[ TC_i = \sum_{k \in \Pi} Delay_k \cdot Q_{i,k} \]

where \( Delay_k \) is the delay of path \( k \).
After calculating timing criticality coefficients for all the rows, we perform row ranking based desktop configuration for large designs with hundreds of rows and thousands of timing paths.

After row sorting, starting from the least critical row, we examine each row to see if it is possible to use a slower library (lower bias voltage) for that row. Assume that all rows after Phase 1 use Library $\text{lib}_j$, we fetch Row $r_i$ from the sorted row list and replace all cells in that row with the cells from Library $\text{lib}_{j-1}$ which is slower than $\text{lib}_j$, update timing and check the timing like we performed in Phase 1. If the result of check timing was true, we fix Library $\text{lib}_{j-1}$ for Row $r_i$ and will continue to examine the next Library ($\text{lib}_{j-2}$) for that row. We examine all libraries below $\text{lib}_j$ for Row $r_i$ until the check timing fails. In that point, we fix all cells of Row $r_i$ to the previous Library which had passed the timing check, update the cluster list and will continue to next row in the sorted row list. Note that if we reach the number of allowed clusters which is given to the algorithm as a constraint, we do not add new clusters but check existing clusters for the remaining rows. After checking all rows, Phase 2 of the algorithm is finished and we have a set of clusters corresponding to each library and each cluster has a set of rows. This section of the algorithm is shown in Figure 3-2.

The computational cost of the heuristic algorithm is $\text{Totalcost} = \text{Max}(C_{\text{phase-one}}, C_{\text{phase-two}})$. The complexity of first phase of the algorithm is $O(\Pi P N W)$, where $P$ is the number of body bias libraries (size of Liblist), $N$ is the number of rows and $W$ is the maximum number of cells in a row. The cost of second phase of the algorithm is equal to $O(N \Pi W) + O(P P N W)$, where $M$ is the number of critical paths (size of $\Pi$). Therefore the overall complexity of the algorithm is $O(M N W)$ which is linear in the number of rows and the number of critical paths in the design. Therefore, the complexity depends on the critical path count, which can lead to high execution time for large balanced designs.
3.1. **Case Study**

In this section we describe the strategy used to experimentally verify our fine-grained FBB.
3.1.1. Experimental setup

To evaluate the effectiveness of our fine-grained FBB methodology, we applied it to a Test-case provided by STMicroelectronic. The design has two main path groups which are related to two different clocks: high frequency clock which is “clock_hf” and low frequency clock which is “clock_lf”. We used 65nm technology provided by ST. For this technology, we built body bias libraries using the library characterization tool (ETS) from Cadence. For each of the cells in the library, we characterized its delay and power for four body bias voltages (vbs) in the range of [0V, 0.150V, 0.300V, 0.450V]. We characterized all four libraries at V_{dd} = 1.0V and normal temperature (25°C).

We synthesized the design in Synopsys DC Compiler and placed them using Cadence SoC Encounter. After placement of the design we dump a set of files which are loaded into Synopsys PrimeTime. We load the design and the list of rows dumped in the placement phase into the PrimeTime. The fine-grained FBB algorithm is implemented using PrimeTime scripting. All experiments are performed on a machine with Intel(R) Core(TM)2 Duo CPU at 2 GHz with 4MB cache and 4GB of memory.

Figure 3.3 shows the placed design in Cadence SoC Encounter.

![Placed design in SoC Encounter](image)

Figure 3.3: Placed design in SoC Encounter

3.1.2. Fine-grained FBB vs. Full FBB

To evaluate the effectiveness of our fine-grained FBB approach, we compared timing and power results with those of single and full FBB while putting the entire design at one body bias voltage to obtain the desired speed-up. We performed experiments for different speed-up targets in a range from 5% to 12%. Speed-up is defined as the percentage reduction in the critical path delay and is less than 100%. Note that, the results of this table are obtained with maximum 2 clusters.
Figure 3-4: Results of fine-grained FBB technique applied on the case study in 65nm

The leakage overhead for three different speed-up is shown in Figure 4 for two different clock groups: “clock_lf” and “clock_hf”. The leakage overhead is calculated with respect to the original designs with no body biasing. As shown in this figure, the leakage overhead of our fine-grained FBB approach is much less than that of full and single FBB. Moreover, the leakage overhead of the full FBB for a range of speed-up factors is the same, because all rows are connected to the same body bias voltage vbs. However, for fine-grained FBB the leakage overhead depends on the speed-up value. For instance, the leakage overheads for 8%, and 12% speed-ups on the paths related to “clock_lf” are equal in the full FBB approach (note that, the difference between leakage overhead of full FBB between 5% and 8% speed-up is due to changing in the VBB voltage of Full FBB); while, in fine-grained FBB the leakage overhead is different for each speed-up factor and depends on the number of rows connected to the higher FBB.

In summary, the results show that our fine-grained FBB provides the same speed-up as the traditional full FBB techniques, while having much lower leakage overhead.

3.1.3. OCV Timing analysis

Finally, on-chip-variation (OCV) analysis is performed on the test-case using PrimeTime. We
calculated the slack distribution of the critical paths in the designs with OCV. We also calculated the slack distribution after applying FBB. Figure 3-5 shows the related plots for the paths related to “clock lf”. As seen, 30% derate on the cell timing delay (0.7 for early derate and 1.3 for late derate) leads to negative slacks for some of the paths. As shown, after applying our FBB technique with 15% speed-up, the slack distribution of all the paths becomes positive.

Figure 3-5: Slack distribution in test-case (paths of Clock_LF) using OCV analysis with and without FBB
4. **(UNGL-NMX) General Introduction for UNGL and NMX section**

In this section the joint activities of UNGL and NMX in T3.2 are highlighted.

Sensing circuits are a challenging block in NVM technology since their requirements in terms of analogue performance are very demanding. Among them: good matching performance in small area for mosfets in the input stage, properties of good device matching not only at minimum spacing but also at long distance, excellent dynamic performance.

In order to study the effect of PV in such a circuit an ideal test case has been created by porting a simplified version of a sense amplifier into the 35nm node by keeping the critical mosfets at minimum gate length. This solution can appear quite strange for an analogue design perspective where mismatch is reduced by increasing device size and by avoiding minimum channel length. Nevertheless it is helpful in this study where the statistical model cards are generated by many TCAD simulations to include all possible local variations due to RDD, LER etc. All these local variations can be assumed (at a first order) as independent among the different mosfets so that they can be considered mismatch sources.

Therefore by simulating the sense circuit it is possible, in principle, to study the following aspects:
- compare different methodologies to map the PV into the compact model looking at their results in the sense circuit behavior and verify the accuracy-simplicity trade-off.
- analyze the PV response of the blocks composing the sensing circuit weighted by the entire sense circuit transfer function.
- ranking the most critical mosfets which requires particular attention to PV.
- Perform a sensitivity of the circuit to the physically based parameters (like threshold voltage, etc) to identify the ones that play the major role to allow process or design solutions.

Some of these aspects are quite well known in analogue design, e.g. threshold mismatch needs to be reduced in differential input stage, however an in depth analysis of PV impact in the circuit performance and in the methods to map PV in compact models allows to set-up useful guidelines even considering this ideal test case.

The block diagram of the sense amplifier if shown in Figure 4-1, it is composed by two identical OPAs (one of the two is used as reference) followed by a comparator [1].

---

![Figure 4-1: Block level schematic of the sense amplifier.](image)
As anticipated, the differential pairs of the OPAs, the current mirrors etc are kept at minimum lengths, while W are in the range of 100nm-600nm for proper circuit operation. An option in which one element of the differential pair is kept at 3W (with a reduced injected current) is also studied and called M3 case.

In this deliverable the main focus is on:
- studying three methods of mapping PV into statistical models, in particular two methodologies by UNGL and a third one, used for matching, by NMX will be compared
- analysis of the different blocks impact including, by NMX, the additional comparison between the case with different OPA, M3 and M1 cases.

To study PV effect in the circuit Vflip has been chosen: it is defined as the output voltage at which the input voltage is half VDD. Vflip represents the spread w.r.t. the nominal value of the input signal which causes the output to switch (this spread is due to the PV of the mosfets weighted by the circuit transfer function).
4.1. UNGL

4.1.1. Introduction

In this work, we evaluate tools and methodologies for simulating and analyzing the impact of statistical and process variability on nano-scale circuits. Simulations are done using the GSS tool-chain, which includes the “atomistic” simulator GARAND, statistical compact model extractor Mystic and statistical circuit simulator RandomSpice [2].

In this work, we employ the GSS tool-chain to study the impact of statistical variability in a template 35nm device developed by the University of Glasgow Device Modelling Group on a sense amplifier circuit provided by NMX. We investigate the impact of variability on different blocks of the sense amplifier and how variability impacts the performance and yield of the circuit.

In Section 4.1.2, we give an overview of RandomSpice. Section 0 describes the test bed device and the compact modeling strategy is outlined in Section 4.1.4 and results of the variability simulations are presented in the following sections (4.1.5-4.1.8).

4.1.2. RandomSpice

RandomSpice is a Monte Carlo circuit simulation engine previously developed by GSS, which enables large-scale simulations of statistical and process variability, data harvesting and statistical analysis of the simulation results. It supports the GSS advanced statistical compact model parameter extraction and generation strategies, and can harness the power of cluster technology. RandomSpice provides enables accurate statistical circuit and standard cell characterisation, and provides power performance and yield (PPY) analyses. The operational flow of RandomSpice is shown in Figure 4-2.

RandomSpice supports multiple backend simulators, including several industrial versions of SPICE, such as Synopsys HSpice, Cadence Spectre or the open source SPICE simulator ngspice.

The database output and advanced statistical analysis capabilities in RandomSpice go beyond the functionality in traditional SPICE simulators, allowing complex, multi-criteria measurements; statistical calculations; and data mining. This is coupled to powerful statistical enhancement strategies that allow significant savings in computational time, and enable accurate predictions of statistical behaviour far into the tails of statistical distributions.

Integration with GSS’s ‘push button’ cluster technology allows RandomSpice to take advantage of massively parallel task-farming to reduce the time taken to characterise circuits and cells.
4.1.3. **Test Bed Device**

The 35 nm gate length template MOSFET used as the test bed in this study was designed to match closely the performance of a high performance transistor, typical for the 45 nm technology generation [3]. The starting point in the TCAD based design was the doping profile of a real 35nm gate length n-channel MOSFET reported by Toshiba [4]. The transistor features retrograde channel doping and pocket/halo implants to suppress short channel effects (SCE). The device structure, illustrated in Figure 4-3 (NMOS) and Figure 4-4 (PMOS), was further updated to closely match the performance of contemporary 45nm technology devices featuring 1.0nm EOT. Strain-induced mobility enhancement is achieved by tensile contact etch stop cap layer in the front-end process, delivering 21% enhancement in the drive current. The electrical characteristics are illustrated in Figure 4-5 and Figure 4-6, with the NMOS device achieving 1.27A/µm drive current with an off-current of approximately 100nA/µm.

![Flowchart for RandomSpice](image)

**Figure 4-2:** Flowchart for RandomSpice.

![Net doping profile for NMOS device](image)

**Figure 4-3:** Net doping profile for the NMOS device.

![Net doping profile for PMOS device](image)

**Figure 4-4:** Net doping profile for the PMOS device.
The TCAD design not only focused on achieving high performance but also on the reduction of statistical variability. Low doping concentration near the interface, delivered by a retrograde channel profile, reduces RDD-induced threshold voltage variability, which is strongly correlated with the doping concentration close to the interface [5]. Well-controlled short-channel effects and gradual threshold voltage roll-off characteristics are critically important to control the LER induced threshold voltage variability [6].

In addition to the test bed device used here, statistical compact model libraries of scaled bulk, fully depleted (FD) SOI and FinFETs are available for RandomSpice. Statistical variability (SV) sources in these devices include random discrete dopants, line edge roughness, poly-silicon or metal gate granularity. Statistical compact models of devices at different stages of NBTI/PBTI degradation are also available.

4.1.4. Compact Modelling Strategy

The compact models used in this study are extracted from 200 physical drift/diffusion simulations of statistical variability in the 35nm test bed device using the GSS tool GARAND [7]. Each extracted model represents a different simulated characteristic, thus there are 200 statistical compact model cards.

The uniform (or base) model is extracted from TCAD simulations of the testbed device with continuous doping. To extract this model, we combined a group-extraction and local-optimization strategy to obtain the complete set of parameters. The uniform model is used as a base for the statistical extraction stage, and statistical compact models are obtained by re-extracting a subset of the uniform model parameters that capture statistical variations in the transistor characteristics. The statistical parameter set is obtained through physical insight and sensitivity analysis of the compact model parameters. The sensitivity of the chosen parameters is shown in Figure 4-7.

For this work, the statistical compact model parameter set is as follows:

- $V_{th0}$, to account for threshold voltage variation.
- $N_{factor}$ and $V_{off}$ are sub-threshold parameters, accounting for sub-threshold behaviour variation introduced by SV.
- $U_0$ is a mobility parameter, accounting for transport variation introduced by SV.
- $D_{sub}$ is a DIBL parameter, accounting for DIBL variation introduced by SV.
- $M_{inv}$ is a moderate inversion parameter, accounting for moderate inversion variation introduced by SV.
- $R_{ds}$ is a source/drain series resistance parameter, and it accounts for the influence of SV in the source/drain region.
As stated above, 200 “full” statistical compact models were extracted from physical simulations using the GSS drift/diffusion simulator and compact model extractor. Although this approach is more accurate in terms of individual transistor characteristics, 200 models provide limited resolution of the true statistical distribution, particularly in the tails. For this reason, some simulations in this work were also carried out using compact models with parameters generated from continuous, uncorrelated Gaussians, similar to the approach employed by NMX. While providing better resolution, this approach fails to capture the essential correlations between compact model parameters, as well as the non-Gaussian nature of parameter distributions [8]. Although the full model and uncorrelated Gaussian approaches are the focus of this work, more advanced approaches using principal component analysis (PCA) and non-linear power methods (NPM) are under active development by GSS [13Error! Bookmark not defined.].

The mean and standard deviation values used for the uncorrelated Gaussians are given in Table 4-1 for NMOS models and in Table 4-2 for PMOS models.

### Table 4-1: Mean and standard deviation of the statistical compact model parameter set for NMOS devices.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>Sigma</th>
</tr>
</thead>
<tbody>
<tr>
<td>vth0</td>
<td>0.1253</td>
<td>0.03491</td>
</tr>
<tr>
<td>dsib</td>
<td>0.01745</td>
<td>0.002423</td>
</tr>
<tr>
<td>u0</td>
<td>111.6434</td>
<td>24.0178</td>
</tr>
<tr>
<td>voff</td>
<td>-0.0904</td>
<td>0.01726</td>
</tr>
<tr>
<td>minv</td>
<td>2.2135</td>
<td>0.5973</td>
</tr>
<tr>
<td>rdsw</td>
<td>149.28</td>
<td>17.633</td>
</tr>
<tr>
<td>nfactor</td>
<td>1.5663</td>
<td>0.2525</td>
</tr>
</tbody>
</table>

### Table 4-2: Mean and standard deviation of the statistical compact model parameter set for PMOS devices.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>Sigma</th>
</tr>
</thead>
<tbody>
<tr>
<td>vth0</td>
<td>-0.1510</td>
<td>0.04558</td>
</tr>
<tr>
<td>dsib</td>
<td>0.01087</td>
<td>0.001182</td>
</tr>
<tr>
<td>u0</td>
<td>68.663</td>
<td>16.929</td>
</tr>
<tr>
<td>voff</td>
<td>-0.04801</td>
<td>0.02447</td>
</tr>
<tr>
<td>minv</td>
<td>-0.5015</td>
<td>0.08154</td>
</tr>
<tr>
<td>rdsw</td>
<td>272.68</td>
<td>70.826</td>
</tr>
<tr>
<td>nfactor</td>
<td>1.3294</td>
<td>0.2666</td>
</tr>
</tbody>
</table>
4.1.5. **Variability Simulations**

**Whole Sense Amplifier**

Initial simulations focused on applying variability to the whole sense amplifier and investigating the flip voltage. In applying variability to the sense amplifier, only transistors with gate lengths of 35nm have variability applied. The larger gate length transistors are assumed to have no variability and thus use the uniform TCAD model.

Further simulations were run, where statistical variability (SV) was applied to different components in the circuit. For reference, the schematic of the sense amplifier is reproduced in Figure 4-8. In total, simulations were performed with SV applied to the whole sense amplifier; to the comparator only; to both differential amplifiers and to each differential amplifier in turn. For each of these, simulations were performed with full compact models extracted from TCAD, and with uncorrelated Gaussians, based on mean and standard deviation values extracted for seven statistical compact model parameters. For each scenario, 50,000 simulations were performed, using the GSS tool RandomSpice™ [9].

![Figure 4-8: Block level schematic of the sense amplifier.](image)

Initial simulations yielded a distribution that showed truncation in the tails. This was found to be due to the minimum/maximum input ramp voltages being insufficient to switch the comparator in cases where the mismatch was extreme. Allowing the input voltage to ramp from 350mV to 650mV was found to be sufficient to switch the sense in all scenarios, and the complete distribution is shown in Figure 4-9. Note that $V_{\text{flip}}$ is defined as the input voltage at which the output voltage is 0.5V.
As discussed in Section 4.1.4, the limited resolution available using 200 models results in statistical instances that are "inaccessible" from the point of view that not all combinations of models that give rise to a particular value are available. For this reason, simulations were also carried out using parameter generation with uncorrelated Gaussians.

Values for the first four moments of the distributions of $V_{\text{flip}}$, obtained from simulations with variability in the whole sense amplifier and using both compact model approaches are given in Table 4-3. The bias and standard error of these values can also be estimated using a technique known as bootstrapping [10]. Bootstrapping is a non-parametric technique used for estimating the bias and variance of a test statistic. This technique resamples with replacement from the original data in order to obtain distributions that may also have occurred given the same experiment. The test statistic is then calculated for each of the resampled distributions, building up a distribution of values for, in this case, the moments of the original distribution.

Bias in the moments of the original sample can be detected if the mean value of the distribution of the test statistic differs significantly from the value of the test statistic for the original sample. This can occur, for example, due to extreme outliers and can improve the estimate of a test statistic if it is not robust. The standard deviation of the distribution of the test statistic is the standard error of the statistic and can be used, for example, to determine if the value of skewness is statistically significant.

The bias and standard error of the moments of both distributions, as calculated using bootstrapping with 1,000 replicates, are given in Table 4-4.

<table>
<thead>
<tr>
<th></th>
<th>Mean (mV)</th>
<th>St. Dev. (mV)</th>
<th>Skewness</th>
<th>Kurtosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Models</td>
<td>508.1</td>
<td>26.0</td>
<td>-2.9×10&lt;sup&gt;-2&lt;/sup&gt;</td>
<td>2.99</td>
</tr>
<tr>
<td>Gaussians</td>
<td>508.9</td>
<td>28.9</td>
<td>3.4×10&lt;sup&gt;-3&lt;/sup&gt;</td>
<td>3.06</td>
</tr>
</tbody>
</table>

Table 4-3: Moments of the two simulated distributions of $V_{\text{flip}}$.

<table>
<thead>
<tr>
<th></th>
<th>Bias</th>
<th>Std. Err.</th>
<th>Bias</th>
<th>Std. Err</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean (mV)</strong></td>
<td>-4.8×10&lt;sup&gt;-3&lt;/sup&gt;</td>
<td>0.11</td>
<td>-8.4×10&lt;sup&gt;-4&lt;/sup&gt;</td>
<td>0.13</td>
</tr>
<tr>
<td><strong>St. Dev. (mV)</strong></td>
<td>-3.9×10&lt;sup&gt;-3&lt;/sup&gt;</td>
<td>8.3×10&lt;sup&gt;-2&lt;/sup&gt;</td>
<td>-6.2×10&lt;sup&gt;-3&lt;/sup&gt;</td>
<td>9.6×10&lt;sup&gt;-2&lt;/sup&gt;</td>
</tr>
<tr>
<td><strong>Skewness</strong></td>
<td>-9.6×10&lt;sup&gt;-5&lt;/sup&gt;</td>
<td>1.1×10&lt;sup&gt;-2&lt;/sup&gt;</td>
<td>2.3×10&lt;sup&gt;-4&lt;/sup&gt;</td>
<td>1.2×10&lt;sup&gt;-2&lt;/sup&gt;</td>
</tr>
<tr>
<td><strong>Kurtosis</strong></td>
<td>-3.1×10&lt;sup&gt;-4&lt;/sup&gt;</td>
<td>2.1×10&lt;sup&gt;-2&lt;/sup&gt;</td>
<td>-1.7×10&lt;sup&gt;-3&lt;/sup&gt;</td>
<td>2.5×10&lt;sup&gt;-2&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

Table 4-4: Bootstrapped estimates of the bias and standard error of the moments of the distributions.
It is evident from Table 4-3 that using uncorrelated Gaussians to generate compact model parameters is overly pessimistic, with >10% error in both the offset of the mean from 0.5V and in the standard deviation of $V_{\text{flip}}$. The discrepancy between the two distributions can be clearly illustrated by plotting the two distributions on a Quantile-Quantile (QQ) plot [11]. A QQ plot is a graphical method for assessing whether two samples are drawn from the same underlying distribution. In this case, the reference distribution is a Gaussian with the same mean and standard deviation as the data, which is shown as a straight line on the QQ plot. Figure 4-10 presents the data in Figure 4-9 as a QQ plot. It can be seen from this that both distributions of $V_{\text{flip}}$ are very close to Gaussian (as expected from the data in Table 4-3) but that there is clear divergence between the two distributions further into the tails and indicates the problem with using uncorrelated Gaussian generated compact model parameters to estimate 3σ (or higher) design margins.

![QQ plot of the distribution of $V_{\text{flip}}$ for variability applied to the whole sense amplifier.](image)

**Figure 4-10**: QQ plot of the distribution of $V_{\text{flip}}$ for variability applied to the whole sense amplifier.

### 4.1.6. Block Level Variability

To examine the impact of variability in individual blocks in the sense amplifier, statistical variability was applied to the comparator only, both differential amps and each differential amp in turn. Figure 4-11 and Figure 4-12 compare the distributions of $V_{\text{flip}}$ for simulations using full compact models and uncorrelated Gaussian parameter generation, for each of the variability scenarios.

![Distributions of $V_{\text{flip}}$ for statistical variability applied to different](image)

**Figure 4-11**: Distributions of $V_{\text{flip}}$ for statistical variability applied to different
blocks of the sense amplifier, using full compact models.

Figure 4-12: Distributions of $V_{\text{flip}}$ for statistical variability applied to different blocks of the sense amplifier, using uncorrelated Gaussians.

From this, it is evident that the majority of the variability in the sense amplifier is due to mismatch in DIFF0, the input differential amplifier. This agrees with simulations performed by Numonyx/Micron, which also indicate that DIFF0 is worst affected by variability.

The first four moments of the distributions of $V_{\text{flip}}$ for each variability scenario are given in Table 4-5 for simulations with full compact models and in Table 4-6 for simulations using uncorrelated Gaussians.

The distributions of $V_{\text{flip}}$ for each block using the two approaches are shown as QQ plots in Figure 4-13 to Figure 4-16, where the discrepancy between the two approaches is clearer.

<table>
<thead>
<tr>
<th>Block</th>
<th>Mean (mV)</th>
<th>St. Dev. (mV)</th>
<th>Skewness</th>
<th>Kurtosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole Sense</td>
<td>508.1</td>
<td>26.0</td>
<td>-0.029</td>
<td>2.99</td>
</tr>
<tr>
<td>Comparator Only</td>
<td>508.8</td>
<td>9.4</td>
<td>-0.096</td>
<td>2.99</td>
</tr>
<tr>
<td>DIFF0 and DIFF1</td>
<td>508.4</td>
<td>23.3</td>
<td>0.023</td>
<td>2.98</td>
</tr>
<tr>
<td>DIFF0</td>
<td>508.1</td>
<td>22.5</td>
<td>0.014</td>
<td>3.01</td>
</tr>
<tr>
<td>DIFF1</td>
<td>509.3</td>
<td>5.5</td>
<td>-0.049</td>
<td>3.01</td>
</tr>
</tbody>
</table>

Table 4-5: Moments of the distribution of $V_{\text{flip}}$ for different variability scenarios, using full compact models.

<table>
<thead>
<tr>
<th>Block</th>
<th>Mean (mV)</th>
<th>St. Dev. (mV)</th>
<th>Skewness</th>
<th>Kurtosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole Sense</td>
<td>509.0</td>
<td>29.0</td>
<td>3.4×10^{-3}</td>
<td>3.06</td>
</tr>
<tr>
<td>Comparator Only</td>
<td>508.8</td>
<td>9.9</td>
<td>-0.11</td>
<td>3.08</td>
</tr>
<tr>
<td>DIFF0 and DIFF1</td>
<td>509.0</td>
<td>27.3</td>
<td>-3.710^{-3}</td>
<td>3.04</td>
</tr>
<tr>
<td>DIFF0</td>
<td>509.0</td>
<td>26.2</td>
<td>0.013</td>
<td>3.04</td>
</tr>
<tr>
<td>DIFF1</td>
<td>508.9</td>
<td>6.6</td>
<td>-0.1</td>
<td>3.02</td>
</tr>
</tbody>
</table>

Table 4-6: Moments of the distribution of $V_{\text{flip}}$ for different variability scenarios, using uncorrelated Gaussians.
4.1.7. Yield Analysis

As well as investigating the effect of statistical variability on flip voltage, the impact on power consumption can also be analysed using RandomSpice. This allows the estimation of power/performance/yield curves to be made (as lower flip voltages translate to smaller delays). For this case study, simulations using full compact models and with SV applied to the whole sense amplifier were used. As above, the flip voltage is defined as the voltage at which the output is 0.5V. The power is quantified in terms of the total energy consumed by the sense amplifier over the simulation. This figure thus includes both leakage/static energy and dynamic power.
Figure 4-17: Scatterplot of flip voltage vs. energy used for simulations of whole sense variability using full compact models.

Figure 4-17 shows a scatterplot of flip voltage against total energy consumed over the simulation window. In digital circuits, it is generally expected that faster circuits (those with lower flip voltages) consume more power. In this instance, the simulations show that faster circuits instead consume less power. Further investigation of the dynamic behaviour of the sense amplifier indicates that this is due to significant leakage current when the input voltage to the sense amplifier is low. The leakage is reduced at higher voltages, once the output of the comparator has switched. The net effect of this behaviour is that circuits that have a lower flip voltages switch to a lower leakage state sooner, thus consuming less energy overall. This is illustrated in Figure 4-18, which shows the current traces for 15 random circuits. Two outliers are highlighted in black, illustrating that slower circuits leak more and for longer. Flip voltage and total energy values for the two outliers are given in Table 4-7.

Figure 4-18: Current traces for 15 random simulations with whole sense variability. The input is swept from 0 to 1V over a simulation window of 1s.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Flip Voltage (V)</th>
<th>Total Energy (J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slow Outlier</td>
<td>0.5427</td>
<td>$8.28 \times 10^{-5}$</td>
</tr>
<tr>
<td>Fast Outlier</td>
<td>0.4792</td>
<td>$7.75 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

Table 4-7: Flip voltage and total energy values for the two outliers shown in Figure 4-18.
In order to estimate the yield of the design given the particular level of variability introduced by the UNGL compact models, the power and flip voltage distributions can be integrated to produce a 2D CDF, as detailed in [12]. The equi-probability contours of this CDF correspond to a particular level of yield, for example, the 0.99 contour corresponds to 99% yield. Figure 4-19 shows the yield estimates and design limits for the sense amplifier with statistical variability applied to the whole circuit.

![Figure 4-19: Distribution of flip voltage vs power for the sense amplifier, with the 2D CDF and equi-yield contours superimposed.](image)

The yield contours indicate the performance criteria that the circuit must meet in order to obtain a given level of yield in the presence of statistical variability. In this case, the maximum flip voltage allowed for 99% yield is ~0.575V and the maximum energy is ~2.75×10^{-7} J.

4.1.8. DIFF0 Analysis

As seen above, DIFF0 contributes the largest fraction of the variability in this circuit for these inputs. Further analysis was carried out on DIFF0 to investigate the effect of key individual transistors on the variability of the differential amplifier. Figure 4-20 shows the distributions of $V_{\text{flip}}$ due to variability in transistors MDIFF2N1 and MDIFF2N2, which form the differential input; and in M3 and M4, which form the current mirror. These simulations are carried out using 200 full compact models. It should be noted that this approach offers only a finite sampling of the underlying distribution of transistor parameters, since there are only 200 possible models that can be substituted into the circuit for a given transistor. This produces binning in distributions of circuit parameters, as can be seen in Figure 4-21, which shows the tail of the distribution of $V_{\text{flip}}$ due to variability in MDIFF2N1. In this case, multiple square transistors are substituted in order to make up the correct device width, which reduces the effect of binning in the center of the distribution, but it is still clearly visible in the tails. Parameter generation methods, such as principal component analysis [13] are necessary to alleviate this problem.
Within these limitations, it can be seen that the differential input devices have the larger contribution to the total variability in the differential amplifier. The moments of the four distributions are given in Table 4-8.

![Figure 4-20: Distributions of $V_{flip}$ for variability applied to individual transistors in DIFF0.](image)

![Figure 4-21: Magnified view of the tail of the distribution of $V_{flip}$ due to variability in MDIFF2N1.](image)

<table>
<thead>
<tr>
<th>MOSFET</th>
<th>Mean (mV)</th>
<th>St. Dev. (mV)</th>
<th>Skew</th>
<th>Kurtosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDIFF2N1</td>
<td>492.9</td>
<td>12.3</td>
<td>-8.85×10⁻³</td>
<td>2.92</td>
</tr>
<tr>
<td>MDIFF2N2</td>
<td>527.9</td>
<td>13.2</td>
<td>5.76×10⁻²</td>
<td>2.99</td>
</tr>
<tr>
<td>M3</td>
<td>515.2</td>
<td>6.2</td>
<td>5.57×10⁻²</td>
<td>3.00</td>
</tr>
<tr>
<td>M4</td>
<td>500.2</td>
<td>8.9</td>
<td>-4.1×10⁻²</td>
<td>2.94</td>
</tr>
</tbody>
</table>

Table 4-8: Moments of the distributions of $V_{flip}$ due to variability in different MOSFETs in DIFF0.
4.2. NMX

NMX studies on the sense test case are based on adaptation of its internal flow for the MC mismatch simulation. The adaptation is related to the fact that the local statistical variation (supply by UNGL within the Modern cooperation framework) is already available for the single device and not measured from the matched pair. In this report all the 7 std are applied to the related compact model parameters (see UNGL section).

In the next deliverable, different levels of accuracy and a sensitivity analysis will also be investigated by applying or not the 7 available sigmas to the related parameters. This is important in the industrial perspective as a cost for complexity vs. accuracy benefit trade off in particular considering the practical aspects to distinguish different variability sources from mismatch electrical measurements on device pairs. This study is therefore an interesting benchmark of how and which local variability sources may affect an important analysis of the circuit performances. Prerequisite for this sensitivity analysis is the verification of the methodology and the preliminary analysis of the circuit behaviour.

Two different criteria have been analyzed to check for possible differences in the impact of PV on sense operation. The first, already anticipated in the general introduction and in the UNGL section is Vflip. The second, called Ipeak is based on the input voltage corresponding to the circuit consumption current peak that is, in turn, related to the sense switching. The two criteria shown identical results if the input swing and the MC simulation sample is large enough to avoid distribution cut or tail deformation, see Table 4-9, Figure 4-22 and Figure 4-23. For the two criteria the simulation is performed in the transient domain by applying a slow Vinput ramp since the objective of this study is the PV impact. Other circuit performances are outside the scope of this deliverable and will possibly be covered in the next one.

<table>
<thead>
<tr>
<th>Criterium</th>
<th>Vinput range (mV)</th>
<th>MC run</th>
<th>min (mV)</th>
<th>MAX (mV)</th>
<th>Average (mV)</th>
<th>Std (mV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ipeak</td>
<td>450-550</td>
<td>1000</td>
<td>450,00</td>
<td>550,00</td>
<td>501,93</td>
<td>27,262</td>
</tr>
<tr>
<td>Vflip</td>
<td>450-550</td>
<td>1000</td>
<td>451,12</td>
<td>549,94</td>
<td>506,77</td>
<td>22,787</td>
</tr>
<tr>
<td>Ipeak</td>
<td>350-650</td>
<td>1000</td>
<td>422,55</td>
<td>600,18</td>
<td>508,99</td>
<td>28,299</td>
</tr>
<tr>
<td>Vflip</td>
<td>350-650</td>
<td>1000</td>
<td>422,54</td>
<td>600,18</td>
<td>508,99</td>
<td>28,299</td>
</tr>
<tr>
<td>Ipeak</td>
<td>450-550</td>
<td>10000</td>
<td>450,00</td>
<td>550,00</td>
<td>501,92</td>
<td>26,894</td>
</tr>
<tr>
<td>Vflip</td>
<td>450-550</td>
<td>10000</td>
<td>450,14</td>
<td>549,97</td>
<td>505,99</td>
<td>22,658</td>
</tr>
<tr>
<td>Ipeak</td>
<td>350-650</td>
<td>10000</td>
<td>389,55</td>
<td>617,66</td>
<td>508,68</td>
<td>27,910</td>
</tr>
<tr>
<td>Vflip</td>
<td>350-650</td>
<td>10000</td>
<td>389,57</td>
<td>617,68</td>
<td>508,68</td>
<td>27,910</td>
</tr>
</tbody>
</table>

Table 4-9: Min, MAX, Average and Std of the sense response with different MC run and input voltage span. The effect of mc run sample size and input voltage swing on circuit performance can be summarized as follow: 1k run are not sufficient to capture rare events in spite the small number of transistor involved, 10k is fine. The input ramp in the range of 350-650mV is the best compromise to cover possible input variation and dense voltage step for good accuracy with reasonable simulation efforts.
Figure 4-22: Impact of MC run and span on sense distribution for the Ipeak criteria: 1000 run and 450-550mV.

Figure 4-23: Impact of MC run and span on sense distribution for the Ipeak criteria: 10000 run and 350-650mV.

The analysis of the impact of each block in the whole sense response is given in Table 4-10. It has been obtained by applying PVs only to the selected block. The major impact is given by the input differential stage (DIFF0); the reference differential amplifier (DIFF1) has a reduced impact even if it is identical to DIFF0 since the PV is weighted by the entire sense transfer function. For the same reason it is not possible to simply sum the PVs of each block to obtain the global one considering the blocks as independent. This point will be possibly studied more in detail in the next deliverable at device level. In fact, it is of great interest in preliminary circuit dimensioning to understand the trend of the relative impact of each device local PV (and therefore at first level independent) on the whole circuit response without performing extensive simulations.

Table 4-10: Impact of selected sense block PV on the global sense response.

<table>
<thead>
<tr>
<th>Blocks with applied variation</th>
<th>Std (mV) of Vflip</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL blocks</td>
<td>27,910</td>
</tr>
<tr>
<td>Comparator</td>
<td>10,270</td>
</tr>
<tr>
<td>DIFF0 and DIFF1</td>
<td>25,164</td>
</tr>
<tr>
<td>DIFF0</td>
<td>24,811</td>
</tr>
<tr>
<td>DIFF1</td>
<td>6,2659</td>
</tr>
</tbody>
</table>

Different design styles (for speed reasons) are also verified here. In addition to the standard case called M=1, a second version of the circuit called M=3 has been simulated; in the M=3 design one element of the input differential pair is 3x the other element (i.e. transistor width is tripled but driven with a reduced current for keeping the same switching condition as the standard case).
Table 4-11 shows that increasing multiplicity means to increase the device size and thus to reduce PV since the Pelgrum rule [14] has been implemented in the statistical model for the channel width. This “PV averaging” is valid even if it is applied to an element of the differential input pair only.

<table>
<thead>
<tr>
<th>Sense version</th>
<th>Std (mV) of Vflip</th>
</tr>
</thead>
<tbody>
<tr>
<td>M=1</td>
<td>27.91</td>
</tr>
<tr>
<td>M=3</td>
<td>22.45</td>
</tr>
</tbody>
</table>

4.3. References

5. **(UNCA) Comparative Analysis of Yield Optimized Pulsed Flip-Flops**

UNCA has analyzed the influence of random process variations on pulsed flip-flops. Monte Carlo simulation results demonstrate that using transistor reordering and dual threshold voltage transistors timing, energy and energy-delay-product yields of more than 1.98, 1.62 and 1.99 times higher are obtained, without requiring architectural modifications and without increasing silicon area requirement. Experimental results show that among the compared circuits the Conditional Precharge flip-flop achieves the highest delay, energy and energy-delay-product yields.

![Pulsed flip flop: (a) block diagram; (b) timing analysis](image)

**Figure 5-1: Pulsed flip flop: (a) block diagram; (b) timing analysis**

### 5.1. Pulsed Flip-Flop Topologies and Simulation Methodology

In this study, UNCA considered pulsed FF topologies widely used in high-performance digital designs. Typically a pulsed FF is structured as shown in Figure 5-1a. A pulse generator circuit derives the *Pulse* signal from the clock edge (Fig.1b) usually exploiting the skew between the clock and its inverted version, thus producing a narrow transparency window for the latch. Note that the *Data* signal can arrive even later than the clock edge. This corresponds to a negative setup time which helps to reduce the data-to-output delay [4].
The four representative pulsed FF topologies shown in Figure 5-2 were selected as case study. In all the chosen circuits the clock signal and its inverted version feed the gates of series connected transistors, thus their skew ensures a properly wide transparency window. Figure 5-2a shows the Hybrid-Latch Flip-Flop (HLFF), used in AMD K6 and K7 processors. This hybrid circuit is particularly fast. However, owing to its pre-charged structure, this design is associated with considerable power consumption [4]. An improved design is the Conditional Precharge Flip-Flop (CPFF), depicted in Figure 5-2b. This circuit overcomes the problem of the glitches at the output, thus reducing dynamic power consumption. This is accomplished by appropriate insertion of keeper elements and introducing a conditional precharge technique to prevent unnecessary transitions [5]. Another interesting hybrid design is the Semi-Dynamic Flip-Flop (SDFF) which is shown in Figure 5-2c. This circuit achieves very high speed at the expense of considerable energy consumption, mainly owing to the switching activity of the clock pulse generator and to the highly loaded dynamic internal node. A more advanced semi-dynamic Flip-Flop implementation is represented by the UltraSPARC Semi-Dynamic Flip-Flop (USDFF), shown in Figure 5-2d. The improvement with respect to the SDFF topology mainly consists in the use of a conditional keeper on the dynamic internal node. It was demonstrated that this allows the energy consumption to be significantly reduced [6]. All the FFs were designed using the STMicroelectronics 45-nm 1V CMOS technology; in Figure 5-2 transistor sizes are reported.

![Figure 5-2: Analyzed Flip-Flops: (a) HLFF; (b) CPFF; (c) SDFF; (d) USDFF.](image)

Figure 5-3 shows the used simulation setup. Input buffers are placed between ideal voltage sources and data/clock inputs to provide realistic input signals. The data input buffer is symmetrically sized to take into account the actual input data capacitance, whereas the clock input buffer is symmetrically sized to keep a constant clock slope equal to \( FO2 \)\(^1\) [7], as it is usually adopted in real designs. The output of a given FF is loaded with 15 minimum sized inverters (i.e. \( \approx4\text{fF} \)). This choice allows realistic running conditions to be examined [8].

\(^1\) FO2 is the slope of the output waveform of an inverter loaded by 2 inverters of the same size [11].
Various optimization criteria can be used for sizing transistors of the analyzed circuits [7]. As an example, transistors can be sized to minimize the EDP, the ED^2P (more effort is provided in reducing delay) or the E^2DP (more effort is provided in reducing energy). In a first phase of this work, all the analyzed FF circuits were sized for optimal EDP, considering energy and delay equally important. The nominal 1V power supply voltage, a temperature T of 100°C, a clock frequency of 1 GHz and pseudorandom input data with a 25% activity rate (i.e. a_{avg} = 0.25) [4] have been used in the experiments.

Since the number of transistors of a single topology varies from 22 to 26 transistors, proper circuit simplifications were introduced to manage the transistor sizing optimization. Transistors that do not significantly affect the FF performance (shown as * in Figure 5-3) were minimum sized to limit the energy consumption. The remaining devices were sized by iterative simulations imposing equal width for series-connected transistors [9]. The iterations were performed until the optimum EDP was obtained.

The impact of process variations was evaluated through Monte Carlo (MC) simulations performed on 1000 samples. In this analysis both inter-die (i.e. fluctuations in device parameters impacting on all devices within the same die in the same way) and intra-die (i.e. random variations in electrical characteristics impacting on different devices in the same die in a different manner) were considered. It is worth noting that, both driving circuits and output loads are not influenced by random process variations in order to isolate process variability effects on circuits under test.

The FF delay considered in this study is the data-to-output delay (TDQ_b) [10] which includes both the worst clock-to-output delay (TCQ_b) and the setup time (T_{setup}). The latter is usually defined as the data-to-clock offset that corresponds to a 10% increase in the clock-to-output delay [10]. Since the setup time can be deeply influenced by process variations, special attention was paid to the determination of the data-to-clock offset to be used in MC analysis. To this purpose, the mean value and the standard deviation of the setup time were evaluated through appropriate parametric MC simulations. Then, the data-to-clock offset was set to the 3-sigma setup time (i.e. (\mu+3\sigma)_b) in the subsequent MC simulations used for evaluating the FF delay. In this way a setup-time margin is introduced, which assures that more than 99.7% of the performed MC runs satisfy the constraint of having less than 10% increasing in the clock-to-output delay.

**Figure 5-3: The simulation setup**

Various optimization criteria can be used for sizing transistors of the analyzed circuits [7]. As an example, transistors can be sized to minimize the EDP, the ED^2P (more effort is provided in reducing delay) or the E^2DP (more effort is provided in reducing energy). In a first phase of this work, all the analyzed FF circuits were sized for optimal EDP, considering energy and delay equally important. The nominal 1V power supply voltage, a temperature T of 100°C, a clock frequency of 1 GHz and pseudorandom input data with a 25% activity rate (i.e. a_{avg} = 0.25) [4] have been used in the experiments.

Since the number of transistors of a single topology varies from 22 to 26 transistors, proper circuit simplifications were introduced to manage the transistor sizing optimization. Transistors that do not significantly affect the FF performance (shown as * in Figure 5-3) were minimum sized to limit the energy consumption. The remaining devices were sized by iterative simulations imposing equal width for series-connected transistors [9]. The iterations were performed until the optimum EDP was obtained.

The impact of process variations was evaluated through Monte Carlo (MC) simulations performed on 1000 samples. In this analysis both inter-die (i.e. fluctuations in device parameters impacting on all devices within the same die in the same way) and intra-die (i.e. random variations in electrical characteristics impacting on different devices in the same die in a different manner) were considered. It is worth noting that, both driving circuits and output loads are not influenced by random process variations in order to isolate process variability effects on circuits under test.

The FF delay considered in this study is the data-to-output delay (TDQ_b) [10] which includes both the worst clock-to-output delay (TCQ_b) and the setup time (T_{setup}). The latter is usually defined as the data-to-clock offset that corresponds to a 10% increase in the clock-to-output delay [10]. Since the setup time can be deeply influenced by process variations, special attention was paid to the determination of the data-to-clock offset to be used in MC analysis. To this purpose, the mean value and the standard deviation of the setup time were evaluated through appropriate parametric MC simulations. Then, the data-to-clock offset was set to the 3-sigma setup time (i.e. (\mu+3\sigma)_b) in the subsequent MC simulations used for evaluating the FF delay. In this way a setup-time margin is introduced, which assures that more than 99.7% of the performed MC runs satisfy the constraint of having less than 10% increasing in the clock-to-output delay.
5.2. Circuit-Level Techniques to Improve Parametric Yield

Transistor reordering is a well known technique that can be used to optimize circuit delay and power dissipation. As an example, placing the critical-path transistor (i.e. the transistor driven by last signal of all inputs which assumes a stable value) closer to the output of the gate can result in reduced delay [12]. It has been demonstrated that this approach improves also the timing yield of basic logic gates [13], but it was not investigated before for sequential circuits especially under process variation conditions. Moreover, signal probability based transistor reordering minimizes the switching activity at internal nodes of the logic gate, thus reducing the dynamic power consumption [14].

Table 5-1: PDN transistor ordering

<table>
<thead>
<tr>
<th>PDN transistor ordering</th>
<th>SDFF-USDFF</th>
<th>HLFF-CPFF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>First stage</td>
<td>Second stage</td>
</tr>
<tr>
<td>Configuration1(C1)</td>
<td>MCLK-Mf</td>
<td>Mf-MCLK</td>
</tr>
<tr>
<td>Configuration2(C2)</td>
<td>Mf-MCLK-M0</td>
<td>M0-MCLK</td>
</tr>
<tr>
<td>Configuration3(C3)</td>
<td>M0-MCLK-MCLK</td>
<td>MCLK-M0</td>
</tr>
<tr>
<td>Configuration4(C4)</td>
<td>M0-MCLK-MCLK</td>
<td>MCLK-M0</td>
</tr>
<tr>
<td>Configuration5(C5)</td>
<td>MCLK-MCLK-M0</td>
<td>M0-MCLK</td>
</tr>
<tr>
<td>Configuration6(C6)</td>
<td>MCLK-M0-MCLK</td>
<td>M0-MCLK</td>
</tr>
</tbody>
</table>
As expected, transistor reordering significantly influences the transistor ordering of the second stage which is always done according to the order of the first one. Table 5-1 presents obtained results in terms of mean values ($\mu_D$, $\mu_E$, $\mu_{E_{DP}}$, $\mu_{Leak}$) and standard deviations ($\sigma_D$, $\sigma_E$, $\sigma_{E_{DP}}$, $\sigma_{Leak}$) of delay, energy, EDP and leakage current. As expected, transistor reordering significantly influences the ($\mu+3\sigma)_E$. Note that all the considered FFs can have a negative mean setup time, depending on the PDN transistor configurations. However, in most of the cases, the ($\mu+3\sigma)_s$ is positive due to the setup time spread (i.e. $3\sigma_s$). Comparing the worst and the best delays of the analyzed configurations, it can be seen that, with a proper transistor arrangement, the improvement achieved on $\mu_D$ ranges between 20% (for the USdff) and 28% (for the CPFF). At the same time, an average variation of about 30% in terms of mean energy can be observed, except for the SDFF which shows a $\mu_E$ variation of about 18%. The lower $\mu_E$ variation of the SDFF is mainly due to the presence of two cross-coupled inverters at the node X of the circuit (see Fig.2c). The two inverter always switch when node X is discharged. Note that this mechanism, which is responsible for a significant portion of the overall dissipated energy, is not significantly influenced by transistor reordering. It is worth noting that, transistor reordering does not significantly influence leakage current since it is not impacted by the particular position of the transistors within the stack but it strongly depends on the number of stacked transistors.

Table 5-2 also shows that most favorable configurations from the delay point of view are those in which the data related signals (i.e. D for the first stage and X for the second stage) drive transistors closest to the output node. Those configurations (i.e. C2 for the SDFF/USDFF and C5 for the HLFF/CPFF) also allow the minimum $\sigma_D$ to be achieved. On the contrary, when data related signals drive transistors more distant from the output node the worst $\mu_D$ and $\sigma_D$ are obtained. This can be explained observing that the amount of charge that needs to be discharged through the data related transistors is smaller when they are closer to the output node, considering other devices in the PDN to be in the 'on-state'.

It is worth noting that for the analyzed FF circuits the transistor arrangement leading to the best $\mu_E$ and $\sigma_E$ (i.e. C2 for the HLFF/CPFF, C1 for the USdff and C6 for the SDFF), is that in which the input signals with the highest probability of being at the high logic state (i.e. CLK for the HLFF/CPFF/USDFF and ICLK for the SDFF) are positioned far from the output node. This is due to the minimization of the switching activity of internal nodes [15].

<table>
<thead>
<tr>
<th>Ordering</th>
<th>($\mu+3\sigma)_s$ [ps]</th>
<th>$\mu_D$ [ps]</th>
<th>$\sigma_D$ [ps]</th>
<th>$\mu_E$ [fJ]</th>
<th>$\sigma_E$ [fJ]</th>
<th>$\mu_{E_{DP}}$ [e-27]</th>
<th>$\sigma_{E_{DP}}$ [e-27]</th>
<th>$\mu_{Leak}$ [uA]</th>
<th>$\sigma_{Leak}$ [uA]</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>1.41</td>
<td>44.89</td>
<td>2.61</td>
<td>22.56</td>
<td>1.97</td>
<td>1012.7</td>
<td>85.1</td>
<td>1.32</td>
<td>0.543</td>
</tr>
<tr>
<td>C2</td>
<td>-3.23</td>
<td>38.99</td>
<td>2.32</td>
<td>23.19</td>
<td>1.99</td>
<td>904.2</td>
<td>72.75</td>
<td>1.36</td>
<td>0.542</td>
</tr>
<tr>
<td>C3</td>
<td>4.73</td>
<td>49.01</td>
<td>2.88</td>
<td>24.42</td>
<td>2</td>
<td>1196.8</td>
<td>98.4</td>
<td>1.35</td>
<td>0.535</td>
</tr>
<tr>
<td>C4</td>
<td>10.84</td>
<td>49.84</td>
<td>2.8</td>
<td>21.45</td>
<td>1.78</td>
<td>1069.1</td>
<td>85.25</td>
<td>1.4</td>
<td>0.541</td>
</tr>
<tr>
<td>C5</td>
<td>-1.52</td>
<td>39.43</td>
<td>2.35</td>
<td>21.01</td>
<td>1.95</td>
<td>828.4</td>
<td>72.1</td>
<td>1.38</td>
<td>0.540</td>
</tr>
<tr>
<td>C6</td>
<td>7.75</td>
<td>47.1</td>
<td>2.65</td>
<td>20.08</td>
<td>1.84</td>
<td>945.8</td>
<td>82.15</td>
<td>1.36</td>
<td>0.539</td>
</tr>
</tbody>
</table>
As a subsequent experiment, UNCA exploited the DVT approach in conjunction with the transistor reordering. More precisely, the lower-Vth devices were used in the critical paths to optimize the performance, while the higher-Vth devices were used in non critical paths to reduce leakage power [16]. Since the exploited 45-nm STM General Purpose transistors library includes devices with standard (SVT) and high threshold (HVT) voltages we used SVT transistors to implement delay-critical PDNs, the PUP of the second stage and the output inverter of each FF; whereas, HVT transistors were exploited in the remaining portions of the circuit which are not critical from the delay point of view. In this case the transistor sizes were maintained unchanged. Obtained results are shown in Table 3. The setup time margins are not significantly influenced by the use of the DVT technique, thus their values are not reported in Table 3.

Comparing the results given in Table 5-2 and Table 5-3, it can be observed that, as expected, the exploited DVT strategy has a minor impact on \( \mu_0 \) and \( \sigma_0 \). On the contrary the use of HVT devices to replace SVT transistors which are not critical for the delay has a beneficial effect to reduce both mean and standard deviation of the leakage current, thus leading to a significant decrease of \( \sigma_{Leak} \) depending on the PDN transistor configurations. More precisely, comparing the best and the worst PDN configurations in terms of energy consumption it can be seen that the improvement obtained on \( \sigma_E \) ranges between 10.9% (for the SDFF) and 15.8% (for the CPFF). Moreover, the mean energy consumption is slightly reduced. As highlighted in Table 5-3, for each Flip-Flop topology the best transistor arrangements in terms of performance or energy consumption are the same as those shown in Table 5-2.

**Table 5-3: Results obtained using transistor reordering and DVT.**

<table>
<thead>
<tr>
<th>SDFF</th>
<th>Ordering</th>
<th>( \mu_0 ) [ps]</th>
<th>( \sigma_0 ) [ps]</th>
<th>( \mu_E ) [fJ]</th>
<th>( \sigma_E ) [fJ]</th>
<th>( \mu_{EOP} ) [e-27]</th>
<th>( \sigma_{EOP} ) [e-27]</th>
<th>( \mu_{Leak} ) [uA]</th>
<th>( \sigma_{Leak} ) [uA]</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1+DVT</td>
<td>45.12</td>
<td>2.74</td>
<td>22.18</td>
<td>1.78</td>
<td>1000.8</td>
<td>67</td>
<td>1.23</td>
<td>0.47</td>
<td></td>
</tr>
<tr>
<td>C2+DVT</td>
<td>39.1</td>
<td>2.49</td>
<td>22.67</td>
<td>1.72</td>
<td>886.4</td>
<td>65.75</td>
<td>1.24</td>
<td>0.47</td>
<td></td>
</tr>
<tr>
<td>C3+DVT</td>
<td>49.08</td>
<td>3</td>
<td>23.95</td>
<td>1.68</td>
<td>1175.5</td>
<td>80.15</td>
<td>1.23</td>
<td>0.47</td>
<td></td>
</tr>
<tr>
<td>C4+DVT</td>
<td>49.94</td>
<td>2.94</td>
<td>20.61</td>
<td>1.7</td>
<td>1029.3</td>
<td>79.3</td>
<td>1.24</td>
<td>0.48</td>
<td></td>
</tr>
<tr>
<td>C5+DVT</td>
<td>39.5</td>
<td>2.44</td>
<td>20.88</td>
<td>1.71</td>
<td>816.9</td>
<td>66.3</td>
<td>1.26</td>
<td>0.48</td>
<td></td>
</tr>
<tr>
<td>C6+DVT</td>
<td>47.21</td>
<td>2.79</td>
<td>19.72</td>
<td>1.64</td>
<td>931</td>
<td>77.25</td>
<td>1.26</td>
<td>0.48</td>
<td></td>
</tr>
</tbody>
</table>
Figure 5-4 shows the effects of the analyzed techniques on the SDFF topology. Results demonstrate that the conjunct use of transistor reordering and DVT technique considerably improves timing and energy yields of fabricated circuits, concurrently. More precisely, comparing the transistor stack arrangements C5 and C1, delay, energy and EDP yields are 1.98, 1.62 and 1.99 times higher, respectively. The yield is here referred to the µ value of the C1 configuration. From data given in Table 5-3 it can be seen that similar improvements are achieved for the other analyzed FF topologies. It is worth noting that such improvements are obtained without requiring any architectural modifications in the design and without increasing the die area.
5.3. Post-layout comparative analysis

Differently from previous technology generations, where the circuit performance variability was dominated mainly by variations at the transistor and the gate levels, in recent technologies larger fluctuations in on chip interconnect parameters can occur [16]. Thus, for the sake of completeness the physical design of the four referenced FF circuits was carried out, using geometrical rules conventionally adopted in standard cell design [9]. In order to perform a fair post-layout comparison, for each analyzed FF topology the solution which minimizes the cost function $CF(C)$ defined in [17] and reported in (5.1) has been selected, where $C$ is the generic transistor configuration.

$$CF(C) = [\mu_{EDP}(C) \times \sigma_{EDP}(C)]$$  \hspace{1cm} (5.1)

The function in (5.1) is not used as an optimization criterion, but it was exploited to choose the configurations to be laid-out. Table 3 shows that the configuration C2 minimizes the above metric for all the compared FFs, the SDFF except. For the latter the minimum value of $CF(C)$ is obtained with the configuration C5.

Post-layout MC simulation results are provided in Table 5-4.

<table>
<thead>
<tr>
<th></th>
<th>$\mu_D$ [ps]</th>
<th>$\sigma_D$ [ps]</th>
<th>$\mu_E$ [fJ]</th>
<th>$\sigma_E$ [fJ]</th>
<th>$\mu_{EDP}$ [e-27]</th>
<th>$\sigma_{EDP}$ [e-27]</th>
<th>$(3\sigma/\mu)_D$</th>
<th>$(3\sigma/\mu)_E$</th>
<th>$(\mu+3\sigma)_D$ [fJ]</th>
<th>$(\mu+3\sigma)_E$ [fJ]</th>
<th>$(3\sigma/\mu)_{EDP}$</th>
<th>$(\mu+3\sigma)_{EDP}$ [e-27]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDFF</td>
<td>44.7</td>
<td>2.58</td>
<td>22.06</td>
<td>1.82</td>
<td>986.08</td>
<td>71.98</td>
<td>17.32%</td>
<td>52.44</td>
<td>24.75%</td>
<td>27.52</td>
<td>21.9%</td>
<td>1202.02</td>
</tr>
<tr>
<td>HLFF</td>
<td>42.64</td>
<td>2.15</td>
<td>27.31</td>
<td>1.61</td>
<td>1164.23</td>
<td>30.11</td>
<td>15.13%</td>
<td>49.09</td>
<td>17.69%</td>
<td>32.14</td>
<td>7.76%</td>
<td>1254.56</td>
</tr>
<tr>
<td>USDFF</td>
<td>48.19</td>
<td>2.26</td>
<td>18.66</td>
<td>1.04</td>
<td>861.9</td>
<td>36.19</td>
<td>14.68%</td>
<td>52.97</td>
<td>18.72%</td>
<td>21.78</td>
<td>12.6%</td>
<td>970.47</td>
</tr>
<tr>
<td>CPFF</td>
<td>42.11</td>
<td>2.25</td>
<td>18.07</td>
<td>1.23</td>
<td>760.93</td>
<td>20.26</td>
<td>16.03%</td>
<td>48.86</td>
<td>20.42%</td>
<td>21.76</td>
<td>7.99%</td>
<td>821.71</td>
</tr>
</tbody>
</table>

The ratio between the maximum spread $3\sigma$ and the mean value $\mu$ (i.e. $3\sigma/\mu$), usually indicated as the parameter variability, was considered as a measure of the uncertainty of a particular parameter under random process variations. As can be easily observed, all the FF topologies show similar results in terms of delay variability (it ranges from 14.68% for the USDFF to 17.32% for the SDFF). A more differentiated susceptibility to process variations can be observed in terms of energy dissipation. The SDFF has the highest energy variability (~24.7%). Again, this is due to the pair of cross-coupled inverters driven by node X. As a consequence of process variability, the discharging speed of node X can significantly vary. This leads to considerable variations of the short circuit energy of the second stage, thus impacting the variability of the total energy consumption.

The USDFF is the circuit with the lowest energy uncertainty (~16.7%). It is worth pointing out that the CPFF circuit exhibits the lowest mean values for all the performance parameters (delay, energy and EDP). Its highest speed comes from a reduced load capacitance at the node X that has two discharging path. Furthermore, due to its conditional precharge technique that prevents unnecessary transitions [18], the CPFF achieves the lowest energy consumption. As an additional advantage, it exhibits a reduced EDP variability.
The 3-sigma value (defined as $\mu+3\sigma$) is also provided in Table 5-4. Such parameter gives more practical information to evaluate the achievable post fabrication yield. As illustrated in Figure 5-5, the 99.87% of fabricated circuits based on CPFF topology would have worst case delay lower than 48.86ps and energy dissipation lower than 21.76 fJ. The 99.81%, 94.65% and 88.12% of fabricated HLFF, SDFF and USDFF circuits would reach a speed performance similar to that obtained for the CPFF structure. At the 3-sigma energy value of the CPFF, the USDFF and the SDFF achieve an energy yield of 99.84% and 43.45% respectively, whereas the HLFF presents an energy yield near to zero. As expected, the CPFF shows also the lowest 3-sigma value in terms of EDP, thus it is the best solution between the four analyzed circuits. At the CPFF 3-sigma value the USDFF and the SDFF show an EDP yield of 13.35% and 1.12% respectively, whereas the HLFF presents an EDP yield near to zero.

In order to explicitly evaluate the impact of process variability due to local interconnections, post-layout MC simulation results are compared to pre-layout results in Figure 5-6. As expected, the mean values of delay, energy and EDP increase as layout parasitics are taken into account. Differences in increasing percentage depend on the layout complexity of the different FF circuits. Also the standard deviation values of delay, energy and EDP increase when layout parasitics are taken into account, thus increasing the uncertainty spread around the mean values. More precisely, the increase of $\sigma_D$ ranges between 5.7% (for the SDFF) and 7.6% (for the CPFF). Similarly, the increase of $\sigma_E$ is between 6.4% (for the SDFF) and 28.1% (for the CPFF). Also $\sigma_{EDP}$ values show an increasing between the 8.6% (for the SDFF) and the 9.8% (for the USDFF). Note that the layout parasitics variability particularly impacts on $\sigma_D$, $\sigma_E$ and $\sigma_{EDP}$ of the USDFF and CPFF circuits. This is very likely due to their longer interconnections.
Figure 5-6: Interconnection impact on FF characteristics.
6. Self-Repairing SRAM Architecture to mitigate the Inter-die Process variations at 65nm Technology

UNCA analyzed a zone based self-repairing technique, based on adaptive body biasing, which helps in the mitigation of the impact of inter die process variations on SRAM cells. Simulation study has been carried out, exploiting extensive Monte Carlo simulations, on 65 nm 1V CMOS STMicroelectronics technology. Results have been evaluated considering operating temperature spreading from 27°C to 125°C.

6.1. The Implemented Mitigation technique

In order to implement the analyzed approach, UNCA distinguishes between dies belonging to low-$V_{TH}$ process corners, dies belonging to standard-$V_{TH}$ process corners and those from high-$V_{TH}$ process corners, on the basis of the leakage current measured from a circuit under test.

As shown in Figure 6-1, differently from [19], we defined three different zones on the basis of the measured leakage current: Zone ‘H’, Zone ‘S’ and Zone ‘L’. Zone ‘H’ corresponds to high leakage (low-$V_{TH}$ process corners). Zone ‘S’ corresponds to standard leakage (standard-$V_{TH}$ process corners) and Zone ‘L’ corresponds to low leakage (high-$V_{TH}$ process corners).

![Figure 6-1: Three zones defined to distinguish between Low-$V_{TH}$, Standard and High-$V_{TH}$ process corners.](image)

![Figure 6-2: Block Diagram of the technique implemented. Tech2 involves temperature monitor whereas Tech1 does not.](image)
The Block diagram corresponding to the implemented mitigation approach is shown in Figure 6-2. Two circuit variations, defined as Tech1 and Tech2 in this report, were designed. The only difference between Tech1 and Tech2 consists of the use of a Temperature Monitor circuit.

The Leakage Monitor produces an output voltage Vin, which is proportional to the leakage current, measured in the SRAM cell under test. Since the leakage current of a CMOS circuit is strongly dependent on the operating temperature, the Vin voltage level can be opportunistically scaled by the Temperature Monitor circuit as in Tech2. Then, Vin acts as an input signal to both the Voltage Comparators, where it is compared to the two reference voltage levels Vref1 = 1.1 x Vnom and Vref2 = 0.9 x Vnom, where Vnom corresponds to Vin when all nominal V_{TH} transistors are considered.

The Voltage Comparators generate Vout1 and Vout2 as output signals, which identify the Zone in which the Vin belongs. Thus it determines whether the die under test lies in Zone ‘H’, Zone ‘S’ or Zone ‘L’. The generic Voltage Comparator [20] is shown in Figure 6-3. When Clk is at the logic level ‘1’, a and b become ‘0’ and q1 and q2 become ‘1’, thus making Vout = ‘0’. The basic functionality of the Voltage Comparator is based on which among ‘a’ or ‘b’ is charged to ‘1’ faster after Clk is made ‘0’. If ‘b’ is charged to ‘1’ faster than ‘a’, then q2 is discharged to ‘0’ and q1 remains at ‘1’. Thus Vout becomes ‘1’. Hence if Vin is greater than Vref, Vout=’1’. On the contrary, when Vin is less than Vref, Vout=’0’.

![Figure 6-3: Circuit diagram of the voltage comparator [20].](image)

Output voltages Vout1 and Vout2, produced by the two Voltage Comparators, act as input signals to the Body Bias Generator which produces the bias voltages (i.e. n bias for NMOS transistors and p bias for PMOS transistors) to be applied to the SRAM cells.

The Body Bias Generator provides reverse body bias for the dies in Zone ‘H’, forward body bias for the dies in Zone ‘L’ and conventional bias otherwise. Reverse body biasing [21]-[23] is an effective technique for reducing the leakage power of a design by raising the voltage of the pMOS N-wells with respect to the supply voltage, or by lowering the voltage of the nMOS P-wells relative to the ground. Similarly, forward body biasing [24]-[26] has been used to increase the speed of a design, although this increases the leakage power as well. In addition, forward body biasing has the desirable result of mitigating the short-channel effects of a transistor, thus reducing sensitivity to critical-dimension variations [27].

The circuit level diagram of the Body Bias Generator is shown in Figure 6-3. If Vout1=1 and Vout2=1, this implies that Vin lies in Zone ‘H’. In this case the values of biasing will be p bias= 1.2V and n bias= -0.2V. If Vout1=0 and Vout2=1, this implies that Vin lies in Zone ‘S’. In this case the values of biasing will be p bias= 1V and n bias= 0V. Finally, if Vout1=0 and Vout2=0, this implies that Vin lies in Zone ‘L’. In this case the values of biasing will be p bias= 0.8V and n bias= 0.2V.
6.2. Results and Discussion

1000 runs of Monte Carlo (MC) simulations were carried out considering two different cases: without temperature monitoring (Tech1) and with temperature monitoring (Tech2).

In the Tech1 case, the voltage references \( V_{\text{ref1}} \) and \( V_{\text{ref2}} \) were chosen considering an operating temperature of 75°C.

MC results obtained for conventional 6T SRAM cell in the case of Tech2 (with temperature monitor) and Tech1 (without temperature monitor) are compared with respect to conventional body biasing in Figure 6-4, Figure 6-5 and Figure 6-7. Evaluated figure of merits are Read Delay, SNM and Leakage power consumption. The parameters were evaluated for 5 different temperature values: 27°C, 50°C, 75°C, 100°C, 125°C. The graphs display the inter die process variations of each parameter around its mean value.

In case of Tech1, variability factor \( \frac{3\sigma}{\mu} \) of leakage power observes a reduction of up to 31.8% and Read Delay as well as SNM observes a reduction of above 9%. As expected, the best results are corresponding to the temperature 75°C for Tech1, and the effects of the applied technique are less evident as we go far from 75°C. Thus, in order to achieve better results for all the temperatures; a Temperature Monitor is also considered in Tech2.

Results given in Figure 6-4, Figure 6-5 and Figure 6-7 are summarized in Table 6-1. DM and VF imply the percentage improvement in the Design Margin \( (\mu+3\sigma) \) and Variability Factor \( (3\sigma/\mu) \). Negative sign indicates degradation in that case.

Two main observations can be made from the Table 6-1. Firstly, all the values corresponding to the Tech2 are positive, which clearly indicates that we get an improvement in every case in comparison to the conventional body biasing of the 6T SRAM cell. Second observation is that each value corresponding to the Tech 2 is equal to or greater than the value corresponding to that of Tech 1. This implies that Tech2 gives better performance for all the considered operating temperatures in comparison to Tech1.
Figure 6-5: Inter Die Read Delay Variations

Figure 6-6: Inter Die SNM Variations
Table 6-1: Percentage Improvements with Implementation of the Techniques (Tech1 and Tech2) in Comparison to the Conventional 6T SRAM

<table>
<thead>
<tr>
<th></th>
<th>27°C</th>
<th>50°C</th>
<th>75°C</th>
<th>100°C</th>
<th>125°C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tech1</td>
<td>Tech2</td>
<td>Tech1</td>
<td>Tech2</td>
<td>Tech1</td>
</tr>
<tr>
<td>RD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VF</td>
<td>-4.4%</td>
<td>6.2%</td>
<td>1.4%</td>
<td>8.6%</td>
<td>10%</td>
</tr>
<tr>
<td>DM</td>
<td>1.1%</td>
<td>1.6%</td>
<td>1.6%</td>
<td>1.8%</td>
<td>1.7%</td>
</tr>
<tr>
<td>SNM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VF</td>
<td>5.7%</td>
<td>7%</td>
<td>5.8%</td>
<td>8.2%</td>
<td>9.2%</td>
</tr>
<tr>
<td>DM</td>
<td>0.33%</td>
<td>0.63%</td>
<td>0.31%</td>
<td>0.82%</td>
<td>0.97%</td>
</tr>
<tr>
<td>Leakage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VF</td>
<td>15.7%</td>
<td>27.1%</td>
<td>29.5%</td>
<td>30.2%</td>
<td>31.8%</td>
</tr>
<tr>
<td>DM</td>
<td>-15%</td>
<td>20.3%</td>
<td>0.7%</td>
<td>21.4%</td>
<td>22%</td>
</tr>
</tbody>
</table>

6.3. References


7. **NXP: Introduction**

With the current advances in VLSI technology the sensitivity of today's chip to power supply noise (PSN) is increasing. In addition to the technology scaling the increase in functional density and the frequency of a chip resulted in significant reliability, test and design challenges due to IR drop, ground bounce and $L_{di/dt}$ effects. The parameters and factors causing these challenges have complex interdependencies and therefore their effects to design performance are hard to model. Dynamic power consumption and IR-drop due to excessive switching activity is one of the most significant of these challenges and has been studied both by industry and academia with respect to design and test aspects. The increase in power dissipation inside the chip causes high current densities in the power distribution network. High current densities coupled with reduced power supply voltages causes an increase power supply noise which adversely affect the reliability and the performance of the circuit. Accurate estimation of supply current is an essential task for today's IC technology in order to design a reliable power distribution network. Power distribution networks suffer from voltage fluctuations due to the rapid changes in the supply current caused by large switching activities inside the design. IR drop and ground bounce are caused by the resistive effects of power supply network. The voltage inside the chip is reduced locally due current traveling from power pads to the core areas. Similarly resistive ground network will experience a voltage increase as current travels through it.

IR drop will reduce voltage difference between VDD and VSS pins of the standard cells leading to a reduction in standard cell's performance and unpredictable timing behavior of a chip [1]. The performance of the chip will be affected if that particular cell is on the critical path. IR drop can also increase the delay of the chip by degrading the current driving capabilities of the transistors leading to a slower design that can fail to meet timing requirements [2]. The task of the designer is to maintain the effects of IR-drop and ground bounce within a tolerable limit. This investigation targets to have a better understanding in local dynamic IR drop effects with the help of detailed switching activity analysis.

Current vector-less statistical IR-drop analyses are performed with the assumption of uniform probability of net toggling since they do not have the knowledge of the states of the nets in the chip. In reality switching activity profile is not uniform inside the design; the activity profile of the chip differs depending on the applied input patterns and depending on the clock cycles of interest. Analysis of the switching activity for every clock cycle and for all possible input patterns is not feasible to perform due to time and space limitations. One of the other aims of the project is to compare the vector-less activity analysis with the activity analysis that is performed by application of different input patterns. Depending on the comparison of the two analyses one can decide how much accurate are the vector-less approaches when they try estimate the activity profile and problematic hot spot areas of the design.

The decoupling capacitances are an efficient way to alleviate power supply noise problem. The role of the decoupling capacitances is to buffer switching activities by supplying part of the current demand. Decoupling capacitances can be placed in the vicinity of switching instances to reduce dynamic local IR drop caused by localized switching activity in certain parts of the design. Accurate calculation for the placement and sizing of decoupling capacitances is necessary to overcome switching activity-generated issues. The project targets to give a better feedback to the designer for decoupling capacitance placement and sizing with the spatial analysis of the switching activity. The project studies the switching activity distribution of the design in space and identifies the problematic areas with high switching activity in the early design stages. The designer can use this information for an accurate placement of the decoupling capacitances.
Another challenge of today’s digital designs is accurate prediction of the silicon behavior with the pre-silicon modeling and simulation. With the shrink in feature sizes and interconnects the performance of the digital designs becomes more sensitive to deep submicron effects. A very important problem arises when the delay of circuit of circuit paths on actual silicon is predicted through the use of simulation tools. Understanding those delay mismatches between silicon measurements and pre-silicon simulation is one of the other aims of this project. A simple noise index model, NIM, is developed to study one of the reasons of the path delay mismatches between measurements and simulation. The NIM model calculates switching activity in the neighbourhood of a path of interest. For certain number of randomly selected different paths; a strong correlation is achieved between the noise index of the paths and the delay difference of the paths.

The remaining part of the report will consider the details of the investigation and the design database. The details of the developed prototype tools, input formats and parameters to these tools will be described in Chapter 8. Chapter 9 will discuss the switching activity profile and hazard analysis of the design in time and in space. Chapter 10 will describe a noise index model, NIM, which is developed in order to study the reasons of the delay mismatches between pre-silicon simulation stage and silicon measurement stage.

7.1. References


8. Investigation:

One of the main objectives of the Hazardous Switching Activity (HSA) project is to have a detailed understanding of switching activity of a design over time and space for test and application modes of operation. At the end of the investigation we would like to understand the following issues:

- How is activity distributed over time?
- How is activity distributed spatially? Is the activity located mostly in certain areas or is it spread uniformly?
- What is the effect of different input vectors (different test patterns and functional input) to the activity profile?
- How much of the activity is generated by functional transitions and how much of activity is generated by non-functional transitions?
8.1. Design Database:

The investigation of the switching activity started with a proper selection of design database to analyze. An NXP digital block TM3271 TriMedia processor is selected for the analysis. The TM3271 is a high-speed 32-bit media processor with an operating frequency of 320MHz. The TM3271 is developed with standard, fully synthesizable ASIC methodology and implemented in a 65nm technology [3].

8.2. VCD File Generation:

The analysis for the switching activity is done with two Perl programs called “activity_analysis” and “coordinate_analysis” which both require VCD and DEF files as inputs. VCD is a very useful ASCII file format with the information of hierarchy, modules, instances and the nets of the design. In addition to the design hierarchy the VCD file also contains information about design activity by reporting the signal value changes in the simulation run. The VCD file can be generated for different input vectors of interest. The input vectors are usually provided in the form of Verilog/VHDL test benches. The required commands to create the VCD file is written in Verilog test benches but if the test bench is written in VHDL the commands can also be written in Verilog netlist. The VCD file can be created with the following commands.

```
$dumpfile("filename.vcd");
$dumpvars(0,DUT);
```

With the `dumpfile` command the name of the VCD file is provided, and with the `dumpvars` command all information on signals that are defined in design under test (DUT) will be reported.

Figure 8.1 shows the flow for the VCD file generation. For a comprehensive activity analysis timing information of the design is also incorporated to the flow with the help of Static Timing Analysis Tool. The STA tools takes layout netlist, LIB library files describing timing characteristics of the cells and SPEF file which is a parasitic extraction file as inputs and generates the SDF as outputs. The netlist can then be simulated with the SDF file using a VHDL/Verilog simulator under the application of different input patterns. The input patterns can be either ATPG generated test patterns or it can application mode input vectors depending on the purpose of the analysis. At the end of the simulation a VCD file which contains switching information of the instances in the design is generated.

![Figure 8-1: VCD file generation flow](image-url)
8.3. Flow of the Activity & Coordinate Analysis

The switching activity analysis tool is written in Perl with the syntax of:

```
activity_analysis   --i <vcd_filename> --o <output_filename> -d <def_filename>
                    -s <start_time>   -e <end_time>
```

Figure 8-2, shows a brief overview of the structure of activity analysis program. With the use of the `activity_analysis` program, distribution of switching activity for different types of nets can be analyzed over time. The program reads two input files called VCD and DEF with two user defined parameters start and end times. For different types of analysis the user can change the range of time that he/she is observing. Since one of the main objectives of the project is to compare switching activity for application and test modes of operations; a VCD file can be generated for different test and application modes. The program reads the VCD file in order to calculate the switching activity of different types of nets for a certain time range. After the switching activity information of the nets is found the program outputs the data in a histogram format. This histogram will tell the number of switching instances for every corresponding simulation time. An example of the histogram data will look like:

<table>
<thead>
<tr>
<th>Simulation Time [ps]</th>
<th>Number of Switching Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>749014400</td>
<td>527</td>
</tr>
<tr>
<td>749014411</td>
<td>3</td>
</tr>
<tr>
<td>749014420</td>
<td>221</td>
</tr>
</tbody>
</table>

The histogram data can then be plotted, for example with Excel program, to see the activity distribution of the design over time. The graph A in Figure 8-2 is an example plot of the activity distribution of a design for one clock cycle. For a more detailed activity analysis the program called `hazard_analysis` can differentiate between functional versus non-functional transitions in the defined time interval. Graph B in Figure 8-2 is an example graph showing the ratio of the functional versus non-functional transitions. Depending on the outcome of the hazard analysis if the ratio of the non-functional transitions is very high either the designer can decide on certain special choices in the synthesis engine or the test engineer can select certain different options in the ATPG engine so that the percentage of non-functional transitions will be reduced during the application of the inputs. Activity analysis is improved with incorporating the drive strength information of the gates into the analysis. Standard cells with larger drive strengths will drive their loads faster with negative effects of more area and more power consumption. The effect of switching between two inverters with different drive strengths 3 versus 20 is not same in terms of power consumption therefore activity analysis is improved by considering the effects of different drive strengths. The programs reads the names of the cells from the DEF file which gives an indication for the drive strengths of the cell’s output stage and provides an estimation of the current consumption. Activity analysis including the drive strength information is called weighted switching activity analysis where switching of a particular instance’s output is multiplied with the drive strength of the output of that instance. With the weighted switching activity analysis more accurate models can be developed to estimate switching activity-generated power consumption and IR drop. Finally switching activity is also analyzed in space. The spatial analysis of the activity is performed with coordinate_analysis tool which reads the exact same inputs and parameters as the activity_analysis tool.
The input syntax of the coordinate analysis is:

```
coordinate_analysis --i <vcd_filename> -o <output_filename> -d <def_filename>
-s <start_time> -e <end_time>
```

The program reads the VCD file to find the switching nets in the design. In order to find the location of the switching instances; the coordinates of the instances are parsed from the DEF file. The program determines the hierarchical instance name for the corresponding net from VCD file so that the X and Y coordinates of the switching instances can be found in the DEF file. The program reports the X and Y coordinates of the switching instances for corresponding simulation times as an output file that is read by a MATLAB code to visualize the activity density over the entire design. A sample output file created with the coordinate_analysis tool is given in Figure 8-3.
The commands to run the MATLAB code are:

```
load (output_filename)
x = (output_filename:2)
y = (output_filename:3)
scattercloud (x, y, n, l ,clm, cmap)
```

Graph C in Figure 8-2, is an example plot generated by the MATLAB program showing the switching activity density of the applied input patterns.

Another useful way of analyzing the location of the switching instances is to use the Cadence First Encounter tool in order to visualize the switching instances. By using Cadence First Encounter the switching instances can be selected to be viewed in a different color than the non-switching instances. A specific area with a lot of switching activity can be zoomed in to see the empty space in the vicinity of the switching instances. The visualization of the switching instances with using Cadence is a very supportive feedback to the designer for decoupling capacitance placement. In order to view the switching instances in a different color the `coordinate_analysis` program is modified such that in addition to the coordinates and simulation time the name of the switching instances will be reported in the output file as well. The output file format needed for Cadence Viewer is given in Figure 8-4.

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>122000</td>
<td></td>
</tr>
<tr>
<td>1515600</td>
<td></td>
</tr>
<tr>
<td>1505600</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 8-4: Output file format generated by coordinate_analysis program**

The output file generated with `coordinate_analysis` program can be modified such that First Encounter will use the file as a command file to change the color of the switching instances. The `cadence` command to select the instance is

```
selectInst <name of the instance>
```

All the switching instances can be selected with the command file in Cadence First Encounter. In the viewer the selected instances will be displayed in white color as in Figure 8-5.
The color of the selected instances can be changed with GUI options to a different color. The usefulness of the Cadence First Encounter is the space around switching instances can be seen by zooming into the switching area as shown in Figure 8-6. Figure 8-6 shows a particular part of the design with highly switching instances selected to be viewed in yellow. The name of the instances can also be seen in the viewer. The information is useful for providing feedback to the designer because the empty areas can be played for optimum decoupling capacitance size and place calculation.

**Figure 8-5: Cadence First Encounter to view the switching instances**
8.4. Input Formats

8.4.1. Path Delay Test Patterns:

The activity profile of the TM3271 is first analyzed for the path delay test patterns. The analysis has been performed on a specific path under high and low activity circumstances. An in-house ATPG tool generated the path delay patterns for TM3271. The path number 2 which is the slowest path according to the silicon measurements is selected for the investigation. In order to detect a transition fault for a certain path the test pattern needs to set only certain states of the flip flops to control the required transition along the path. The other flip flop states can be set to any value depending on the purpose, therefore the don’t care states of these flip flops can be played with to create high and low activity conditions. Originally the AT PG tool creates the patterns with high amount of X’s. Activity modulation can then be performed with different 0-fill of these X’s to create high or low switching activity. For the path 2, 92 patterns are generated in order to create high and low switching activity conditions with different 0-fill of the don’t care states. Out of these 92 patterns 4 different patterns called scl1, scl11, scl46 and scl67 with different activity levels are selected. Table 8-1 summarizes the differences between four activity modulation patterns. Notice that the highest activity is acquired for the last case scl67 when 70% of the X’s are filled with logic 1. It is an unexpected outcome because one would expect that highest activity will occur for case 3 scl46 when half of the X’s are filled with logic 1 and the other half is filled with logic 0. One of the reasons for this could be the structure of the design which will end up with more switching activity due to more 1-fill to the flip flops.

<table>
<thead>
<tr>
<th>Name</th>
<th>% of 0’s in X’s</th>
<th>% of 1’s in X’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>scl1</td>
<td>100%</td>
<td>0</td>
</tr>
<tr>
<td>scl11</td>
<td>80%</td>
<td>20%</td>
</tr>
<tr>
<td>scl46</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>scl67</td>
<td>30%</td>
<td>70%</td>
</tr>
</tbody>
</table>

Table 8-1: Percentage of 1 & 0’s in X’s
A path will start with a launch flip flop, travel through a combinational logic and end up with a capture flip flop. The clocking scheme of path delay test is shown in Figure 8-7. In general, transition delay tests are implemented with the application of two test vectors. Test stimuli are loaded to the scan chain during the shift-in cycles of the scan test, and likewise the test responses are unloaded from scan chains during the shift-out cycles. (Note that the shift-in and shift-out cycles of adjacent patterns overlap). The shift-in and shift-out phases of the scan test are performed at a slower test clock. During the shift-in cycles of the test the states of the flip flops are set to logical 1/0 values with a test clock with a frequency of 10 MHz.

After the last cycle of scan-in operation circuit is clocked with a fast clock in order to launch the transition at the launch flip flop. The circuit is clocked one more time with a fast clock in order to capture the transition at the capture flip flop. The activity analysis is performed for launch and capture cycles because to speed up the analysis time. Shift operations take large number of clock cycles because the number of flip flop states that needs to be set to a logical value for full-scan test can be quite high depending on the size of the circuit. The analysis for the shift-in and shift-out cycles is not less interesting but performing the activity analysis for shift cycles will require a lot of time hence the investigation is limited to the launch and capture cycles of the path delay test.

8.4.2. **Stuck-At Test Patterns:**

Stuck-at test vectors are the second type of the input patterns that is analyzed in this work. The analysis is performed for three randomly selected stuck-at test patterns which are generated with the same in-house ATPG tool. The clocking scheme for the stuck-at test is shown in Figure 8-8. Unlike the path delay test the normal cycle after scan-in is performed with slow test clock. The activity analysis is only performed for the normal cycle of the stuck-at test to reduce the analysis time of the investigation.

![Figure 8-7: Path Delay Test Clocking Scheme](image)
8.4.3. **Application Mode Input Patterns:**

Running functional input patterns on Trimedia processor is the most challenging part of the project. In order to run a representative application on TM3271 the performance benchmark “Motion Compensated Up-Conversion with Halo Reduction” is selected. The benchmark is a very common technique used in modern high-end televisions to increase perceived video quality. The application has to be run on the full Energizer II environment. A detailed understanding of the EII database is mandatory to be able to run the application. An overview of the EII chip is shown in Figure 8-9. To be able to run an application on the TM3271 first the ARM-core has to configure the Clock Generation Unit (CGU), so that every domain in the EII chip will run with its own frequency. The crystal clock frequency is 27 MHz. After the ARM-core configures the CGU clock domain of the Trimedia becomes 312 MHz. After the CGU configuration the ARM-core has to boot the Trimedia and then the application can be run on the TM3271. The required commands to compile and simulate these applications on EII are:

```
  gmake hw                  this will compile the Verilog/VHDL designs in the database
  gmake sw APP = cgu_test   this will compile both the CGU and BOOTING configurations
  gmake sw TMAPP = halo_reduction
  gmake sim APP=cgu_test TMAPP=halo_reduction
```

The simulation is successfully performed in RT-Level and RTL VCD has been generated for the halo_reduction application. RTL VCD only reports the logical value changes of the inputs, outputs and registers that are defined at RT-Level of the circuit. The value changes on the combinational part of the circuit can’t be captured with RTL simulation. Therefore the creation of the RTL VCD for the detailed switching activity investigation was not useful. The simulation has encountered logical X values on some nets when TM3271 RTL description is replaced with gate level description of TM3271. A variety of approaches with the design experts of Trimedia has been applied to get rid of the X values in the simulation but none of the approaches has succeed the perform gate level netlist simulation.
8.5. References


Figure 8-9: Energizer II
9. Results

9.1. Activity Profile in time

9.1.1. Path Delay Test Pattern, Low Activity File (scl1)

The activity analysis is performed on the combinational gate outputs, flip flop outputs and clock
nets. Conceptually a gate is switching only when its output is switching. Internal nets within gates are excluded from the analysis to reduce the complexity of the investigation whereas outputs of combinational logic cells (with type of Z in the VCD), the outputs of registers (with type of Q in the VCD) and the clock pins of the registers (with type of CP in the VCD) are analyzed for path delay and stuck-at test vectors. The VCD file reports the transitions on a net based analysis but this investigation requires instance based switching activity analysis, hence activity_analysis program makes a pre-processing of the header part of the VCD file in order to match the nets with their corresponding instances. Figure 9-1 shows a simple circuit to understand the concept of the switching activity investigation. The transition happening at the output of gate C is reported in the VCD file as a net switching with its corresponding switching time. The activity_analysis performs a pre-processing from the header section of the VCD file to match the nets with types of CP, Z or Q to their corresponding instances and then performs the activity analysis on an instance based switching count. So the transition happening at the output of gate C in Figure 9-1 is counted as one combinational gate (gate C) switching. Although the inputs of gates D, E and F are switching they are not considered as switching instances because the transition at the inputs of these gates can be blocked depending on the value of the side inputs of these gates.

![Figure 9-1: Example circuit](image)

The program generates an output file in the form of a histogram. For every time frame the number of switching Z, Q and CP values are reported separately. The time versus switching count number of the nets is then plotted with Excel to see activity distribution within a clock cycle. Error! Reference source not found. shows the activity distribution of clocks for scl1 (lowest activity case) at launch cycle. Three additional clocks to the main functional clock are identified in Error! Reference source not found.. The clocks differ from each other with 80-90ps of phase shift. Moreover there is a large difference in switching activity that these clocks control. Clock number 3 is the main functional clock of the Trimedia chip and has the largest amount of switching count compared to other three clocks. The difference in the amount of the switching count of these clocks indicates the amount of the flip flops that got controlled by these clocks. When the spatial analysis is performed, the different location of these flip flops can be observed. The spatial activity analysis results will be discussed more in detail in section 3.4 of the report. However figure 26 from spatial activity analysis section shows the different locations of the flip flops that got controlled by these four clocks. Clock number 3 with the highest switching count (refer to Error! Reference source not found.) is observed in figure 26 with more red area than the other clocks. The red area in the
spatial switching activity analysis' plots indicates the switching instances for a certain clock cycle. Trimedia design has one main functional clock which runs at 320 MHz, one test clock which is inactive during its normal mode of operation and active during test mode and two other redundant clocks. When ATPG is performed to generate the test vectors it does not know anything about the functionality of the clocks, so the redundant clocks are also set to logical values during test pattern generation. As a result of it the three additional clocks are observed for the test cases. Figure 9-3 shows the activity distribution at the outputs of flip flops for scl1, the low activity pattern file, at launch cycle. Compared to the activity distribution of clock nets; flip flop outputs do not experience as much switching. The dominant activity at register outputs occurs after a very small delay after the rising clock edge of the launch cycle. Once the clock pin of the flip flop changes its state the state of the flip flop may also change its content after a small delay. The use of negative edge triggered flip flops and flip flops with inverted clock signals are the reasons for the activity on flip flop outputs on the falling edge of the clock 3. Moving further away from clock edges flip flop outputs are quite.

![Figure 9-2: Activity Distribution of CP's at launch cycle, scl1](image)

All the activity plots shown in this section of the report correspond to the not weighted activity analysis where the drive strengths of the instances are not incorporated to the switching activity investigation. The weighted switching activity analysis is only performed for the flip flop outputs and the combinational gate outputs. The difference in the amount of switching activity count for the cases of weighted and not weighted analyses can be seen from amplitudes of the plots for the two cases of analysis.
Outputs of combinational logic gates are the last type of the nets that are analyzed for activity investigation. Figure 9-4 shows activity distribution of combinational gate outputs for the launch clock cycle. Unlike from the former two cases the switching activity of combinational cells is more distributed in time. Peaks can still be observed in the vicinity of the rising and falling clock edge. The similar type of analysis is done for the capture cycle as well. Similar behaviour of switching activity can be observed for the capture cycle although less activity is observed.
The activity investigation is performed with another path delay test pattern where activity modulation is not performed and the X states of the don’t care flip flop are filled without any control. Due to space limitations of the report; results for activity modulation over time are given only for launch cycle of low activity path delay test pattern, scl1. In order to show the differences in activity levels of path delay test patterns switching count statistics are shown in Figure 9-5 and Figure 9-6 for launch and capture cycles respectively. The percentages of $Z$, $Q$- and CP- switching are calculated with equations (1), (2) and (3) correspondingly.

\[
\begin{align*}
Z_{\text{switch-fraction}} &= \frac{\sum Z_{\text{switching}}}{\sum Z_{\text{all}}} \\
Q_{\text{switch-fraction}} &= \frac{\sum Q_{\text{switching}}}{\sum Q_{\text{all}}} \\
CP_{\text{switch-fraction}} &= \frac{\sum CP_{\text{switching}}}{\sum CP_{\text{all}}}
\end{align*}
\]
Figure 9-5 and Figure 9-6 show the increase in activity level of four delay patterns on different types of nets for launch and capture cycles respectively. The reason for having more than 100% of switching activity for clock nets in scl46 and scl67 cases in Figure 9-5 is due to the fact that the activity_analysis tool counts both of the low-to-high transitions and high-to-low transitions during one clock cycle. So theoretically total switching which can happen on the clock nets in one clock cycle is 200%. Comparing the results of the activity profile for four different patterns it can be seen that moving from scl1 to scl67 there is an activity increase on flip flop outputs, on combinational gate outputs and on clock nets. The increase in combinational gate outputs can trigger the increase in clock nets because of the high granularity of the local clock gating used inside the Trimedia. Therefore the activity increase in combinational logic causes more clock nets to be switched. In this investigation of the switching activity, the switching activity is more a concern for the launch cycles than the capture cycles because all the path delay measurements on silicon are performed for the launch clock cycle. The investigation also showed that in this case the capture cycles experience much less activity than launch cycles for path delay test vectors of the Trimedia.

## 9.1.2. Stuck-At Test Vectors

The activity analysis is performed for three randomly selected stuck-at test vectors which are generated with an ATPG tool. Compared to the path delay test, the activity level of stuck-at test vectors is much higher. Figure 16 shows the activity distribution of the clocks for stuck-at test at the normal cycle. It should be noted that the normal cycle of the stuck at test runs at 10 MHz. Compared to the path delay test clock at launch and capture cycles the stuck-at clock is a much slower clock. The four different clock domains can be observed by zooming into the plot 16. Figure 17 shows the plot obtained by zooming into the rising clock edge of figure 16 and figure 18 is obtained by zooming into the falling clock edge of figure 17. With the zoomed versions of the switching activity graph, the four distinct clocks can be identified. Figure 19 shows the activity...
profile at register outputs. Similar to the path delay case activity can be observed in the vicinity of rising and falling clock edges of the normal clock cycle. The activity profile of the combinational gate outputs is shown in figure 20. The combinational activity for stuck-at test is much higher than the path delay test because the goal of the stuck-at test is to detect as much as faults as possible. So in order to be able to increase the fortuitous detection of the other stuck-at faults the test pattern tries to observe as much states as possible. Because of the fact of more state observation, more local clock gates will be active, so the switching count numbers for clock nets are much higher for the stuck-at test when it is compared to the path delay test.
Figure 9H10: Activity Distribution of Q's at normal cycle of stuck at test pattern

Figure 9H11: Activity Distribution of Z's at normal cycle of stuck at test pattern
Finally the activity profile of three stuck at patterns are compared in figure 21. Compared to the path delay test case stuck at test results in much more combinational activity except the high activity pattern scl67. As stated before stuck-at test vectors try to detect as much faults as possible at the same time, hence stuck-at test vectors attempt to observe as much states as possible which enable more clocks to be active in the design. Overall the activity profile of the design for the stuck-at case will be much higher than the path delay test case. For the path delay test, a well defined transition at the launch flip flop and the propagation of that transition through a combinational logic to the capture flip flop is required. Hence few local clock gates will be active in the vicinity of the specific path which will lead to a less activity for the path delay test.

9.2. Comparison of Activity Profiles with Redhawk IR-Drop Profiles:

This section of the report compares switching activity of high activity path delay pattern file .scl67 results that are collected with two different techniques. This section of the report gives a good confidence about the correctness of the activity_analysis tool. The commercial tool Redhawk is used to collect the distribution of the switching activity in time [4]. The same VCD files are used and read by Redhawk. A scenario file is generated from the VCD files. This is a special type of a file which is generated by Redhawk. The scenario file is processed with a script to generate a histogram where number of switching instances for certain time periods is reported. Figure 22 shows the activity profile graph of high activity file scl67 for launch and capture cycles. In the figure the type of nets are differentiated as switching and data_switching nets. Data_switching nets belong to the combinational switching nets in the Trimedia, on the other hand switching nets contain both combinational and non-combinational switching nets. Peaks are more appealing in switching nets because they contain the clock nets, on the other hand the activity profile of the data_switching nets are more uniform because they refer to the combinational switching logic. Figure 23 shows activity modulation graph for the same case obtained with the activity_analysis tool which directly process VCD file and generates the histogram data.
The plots from both analyses show a very similar behavior in the activity distribution in time. Looking to the plots more carefully, the peaks of the both plots can be observed at the same time with almost the same amplitude. Some small differences in the amplitude of the peaks can be observed because of the additional activity on the surrounding blocks around the Trimedia core. The VCD files are generated on chiplet level of the TriMedia core with some other additional wrapper blocks around the core. When Redhawk analysis is performed on the VCD files, the switching activity on the surrounding blocks is included to the analysis. On the other hand, the program activity_analysis is performing some additional filtering on the VCD files such that any switching activity outside the Trimedia core would be excluded from the investigation. So the amplitudes of switching counts of the nodes in the Redhawk analysis will be more than the amplitudes of switching counts of the nodes in the activity_analysis.

Another difference between two plots is due to the binning of the histogram data. The bin size of the Redhawk analysis is 10 ps. On the contrary activity_analysis uses a bin size of 1 ps leading to a more detailed activity distribution in time. The difference in bin size reflects itself in the plots between the times of 2000 ps and 3000 ps.
9.3. **Useless versus Useful Transitions:**
Total switching activity analysis of the investigation is improved with hazard analysis. Hazards are defined as unwanted switching at gate outputs. Figure 24 shows a simple example to understand the procedure to differentiate useless and useful transitions. A very simple algorithm can give the percentage of hazards for a signal in a clock cycle. The algorithm first counts the number of events occurring in one clock cycle, if the event count is even then all the transitions are useless transitions, on the other hand if the number is odd the last event is a useful transition but all the other events happening before are useless transitions. With this detailed activity investigation percentage of useless versus useless transitions out of all transitions can be determined.
Figure 25 shows the percentage of useless versus useful transitions occurring on the combinational gate outputs for launch cycle for path delay test patterns. For low activity file scl1, the percentage of combinational gate outputs’ switching is approximately 3%. The percentage of the switching combinational gate output is the fraction of the switching combinational gate outputs to the all combinational gate outputs of the Trimedia. Out of these 3% of switching outputs almost half if they are caused by useless unwanted transitions. Almost 50% percent of the transitions at the combinational gate outputs are caused by useless transitions. This is a high amount of unwanted switching occurring at the combinational gate outputs, but since the overall switching activity is very low for the low activity pattern scl1 the affect of the unwanted useless transitions to the IR-drop is negligible.

The high activity pattern scl67 has total of 40% combinational gate outputs’ switching where 10% of these total transitions are due to unwanted useless transitions. The data becomes less interesting for capture cycle since the overall switching activity at combinational gate outputs is considerably less than the launch cycle switching activity. Hence the contribution of the useless transitions to the IR-drop will be again negligible. The hazard analysis is performed on 32 additional different paths for the uncontrolled X-fill case.
9.4. Activity Modulation in Space

Understanding the nature of switching activity in the Trimedia requires the visualization of the switching instances spatially. In general the criticality of an instance can be characterized in terms of different parameters. An instance can be critical because it is part of a timing critical path, or it can be located far from power supply connections and experience more IR drop or it can be located in a highly switching environment. With the coordinate analysis program the location of the switching instances can be seen in activity density graphs. In the activity density graphs the color blue will represent the area with no or very low amount of activity whereas the color red will represent the areas with high amount of activity. A particular area can be red either because an instance is switching more than one time or instances that are located very close to each other can switch in the same clock cycle. The activity analysis program output X and Y coordinates of the instances that are switching in specific time. Then these coordinates are read by a MATLAB program which will produce a density scatter plot. Figure 26 shows the switching combinational cells for the launch cycle of the low activity pattern scl1.
From the activity density plots highly switching areas can be identified. The color toolbar on the right hand side of the plot represents the strength of the switching on the density plots. The four different clock domains can also be seen on the activity density plots. As stated in Section 3.1 there are four distinct flip flop clusters that got clocked by different clocks which have 80ps phase difference from each other. From figure 11 the peaks at different times can be observed meaning that four different flip flop cluster got clocked at different types. For each peak in time the name of the instances can be saved and their coordinates can be graphed on activity density plots. Figure 27 shows the location of the four different flip flop clusters. It can be seen that each flip flop cluster is not randomly distributed in the chip instead they are all localized in certain areas. The numbers indicate the clock domain number as it is marked in the figure 11. Clock number 3 is the main functional clock and controlled most of the activity as it is plotted in figure 27. The difference in the color scaling which is used in the spatial activity analysis plots should be taken into account when the switching activity is investigated. In figure 27, the clock number 3 controls a larger area than the other three clocks. In addition to that, its color scaling has the highest number when it is compared with the other color scaling of the remaining other clocks.

Figure 9-17: Location of switching Z’s, launch cycle, scl1
Figure 9H18: Location of four clocks, launch cycle.
10. An Application of the Flow to Simulation Silicon Timing Correlation

10.1. Simulation Silicon Timing Mismatches:

With the shrink in feature sizes and interconnects the performance of the design becomes sensitive to deep submicron effects which makes pre-silicon modeling and simulation insufficient to accurately predict silicon behavior [5]. A related problem involves the difficulty of accurately predicting the delay of circuit of circuit paths on actual silicon through the use of simulation tools. The static timing analysis (STA) tools that are used by the designer to predict the speed-limiting paths are insufficient to estimate the delays of these paths. Mismatches between silicon measurements and simulation are problematic for variety of reasons. It makes it harder to predictably design the circuits so that they would work with expected behavior at first-silicon as well as it makes the silicon debug a more arduous process. The problem is handled nowadays by adding safety margins typically at the cost of more area. As a result multiple researchers have studied such delay mismatches as well as attempted to design complex tools to take into account factors causing the delay mismatches and improve pre-silicon delay predictions. However, these complex solutions suffer from practicality and scalability issues.

As an alternative to creating tools that insert this complexity into the initial analysis, an alternative approach is the development of a simple correction factor that can take the output of an already developed tool and correct the path delay prediction for previously identified delay influences. TM3271 is used for the analysis of simulation and measurement delay correlation. Path delay test vectors are created with an in house tool AMSAL and used for measurements of the paths on silicon. Pre-silicon simulations are performed with Spice simulations while considering side-inputs. [6] The measurements are performed under typical conditions, at 25C with 1.2V of supply voltage. Same settings are also used in the Spice simulations. The effect of IR-drop is not included to the Spice simulations.

The simulation versus measurement correlation graph for TM3271 is shown in figure 28.
Faster on silicon

Slower on silicon

Figure 10.1: Timing Correlation between simulation and measurements for TM3271

Every dot in Figure 10.1 corresponds to a different path on the Trimedia core. Any path locating on the lower side of the line has a real delay which is more than the expected delay and therefore the design will be slower on silicon. There are various factors such as crosstalk and local dynamic IR-drop affecting the silicon simulation timing mismatches.

10.2. Simple Noise Index Model:

Among all of the parameters leading to the timing discrepancy between silicon and simulation, IR drop is considered one of the main reasons. With some minor modifications in the activity analysis tool developed for the HSA project; a correction model called noise index model is developed to understand the timing discrepancy between expected delay obtained with pre-silicon simulations and actual delay measured on silicon. The noise index model calculates switching activity in the neighborhood of a path of interest. Switching activity information for different paths is obtained with the coordinate_analysis program. The flow of the program is shown in figure Figure 10.2. The coordinates of the instances belonging to a specific path is calculated from the timing analysis reports generated by PrimeTime. A path starts with a launch flip flop, goes through combinational logic and ends with a capture flip flop. All the instance names are specified in Synopsys PrimeTime timing report files. The names of the instances corresponding to the path of interest are mapped with the associated names in a DEF file so that the location of the instances is specified. With this information the path location and its relation to the global switching activity of the design is determined.
Figure 10.2: Modified Coordinate Analysis Tool, to determine the location of the paths

Figure 10.3: Switching Activity of Path575 and Path 472

Number of paths with different delay mismatches are selected from Figure 9-18 and simulated with the noise index tool. Switching activity distribution of two paths, Path575 and Path472 is shown in Figure 10.3 in order to look to the relation of the switching activity and delay mismatch of the paths in details. Path 472 is selected from the set of paths that are slower in silicon and it is located in a noisy environment in terms of switching activity. On the other hand path 575 which is selected from the set of paths that are faster in silicon is located in a quieter neighbourhood.
In order to consider the effects of dynamic IR-drop to the delay in a design, the noise index model has developed for the selected paths. The basic idea behind the noise index calculation is given in Figure 10.4. The dotted black ellipses around every instance indicate the region that is used for the noise index calculation. Basically the noise index model is counting the weighted switching activity of the neighbor cells for that defined elliptical region around every instance belonging to a specific path for the launch cycle. It will give the noise indexes of every instance on the path; the sum of those noise indexes will give the noise index of the path. The elliptical shape is selected because of the orientation of the power rails in the design. When a cell is switching the supply will be transferred to the transistors from the power rails which are located as horizontal lines in the design. The radius of the ellipse is selected as 100um along the x-axis and 50um along the y-axis. The activity profile inside the ellipse can be different depending on relative the location of the switching instances to location of the path instance. To illustrate the concept two different activity profiles in same elliptical shape are depicted in Figure 10.5. Essentially Figure 10.5 represents two different cases of switching activity around a specific instance belonging to a certain path. In case 1, switching instances around the path is located very close to the instance which belongs to the path. On the other hand in case 2, switching instances are located far from the instances belonging to the path.

![Diagram of a path and surrounding switching instances](image)

**Figure 10.4:** The basic representation of a path and surrounding switching instances
Considering the differences between the activity profiles around the path instances, three other ellipses are formed around the path instance. The effect of the switching cells falling into the blue ellipse in Figure 10.5 will be counted by 100%, the effect of the switching cells falling into the green ellipse will be counted by 75% etc.

The correlation of the noise index model to delay difference is shown in Figure 10.6.
The y-axis of the graph shows the delay difference between simulation delay and measured actual silicon delay values, so negative values along the y-axis correspond to the paths that are slower in silicon and positive values correspond to the paths that are faster in silicon. The correlation between noise index and delay difference is not perfect but the trend of the relation between the delay difference and noise index can still be observed. One should not expect a perfect correlation between noise index model and the delay difference between measurements and simulation. This noise index model only takes into account the effect dynamic IR-drop caused by the surrounding switching instances. The other factors like cross-talk is not considered in the noise index model.
10.3. Improved Noise Index Model:

The trend between the noise index and delay difference initiated some thoughts on improving the noise index model with putting more physical background into it in order to achieve a better correlation. The decision on the elliptical shape is done considering the physical location of the power rails and the standard cells of the design. The strength of the power rails is not as strong compared to the strength of power layers which are very thick M6 and M7 layers and providing actually supply voltage to the instances. 10.7 shows a part of the TM3271 layout with only power layers, power rails and standard cells.

![Figure 10.7: Power layers, power rails and TM3271 instances](image)

The power layers of TM3271 are the horizontal thick red M7 metal layers and vertical white M6 layers. The thin blue lines are the M1 power rails and the remaining red polysilicon corresponds to the standard cells of the design. The effect of the M1 power rails on the switching instances is negligible they are just contacts between the top power layers and the cells. The power supply is provided from power layers. The decision of the right shape for the noise index model calculation should consider the metal coverage of M6 and M7 layers. The metal coverage of a power layer is defined as the ratio of the metal layer’s width to the metal layer’s pitch. The pitch and the width of the M6 and M7 layers are shown in 10.7. An arbitrary radius choice of 100um and 50um for the basic noise index
model is improved by calculating the actual decoupling distance \( r_d \) which is shown in equation 4 [5].

\[
r_d = \sqrt{\frac{T_s}{R_{sq} * C_u}}
\]  

(4)

\( T_s \) is the switch duration time of the cells and taken 300ps as an average, \( C_u \) is the capacitance per unit area and taken 4fF/um\(^2\) and finally \( R_{sq} \) is the resistance of the metal layer which is given in equation 5.

\[
R_{sq} = 4 \frac{R_{sq,miz}}{\eta_{miz}}
\]  

(5)

\( R_{sq,miz} \) is the thick metal sheet resistance of the ohmic metal layers and taken 0.020 \( \Omega \) and \( \eta_{miz} \) is the metal coverage of the thick metal layers, which is used for power and ground layers. Table 10.1 shows the corresponding values for M6 and M7 layers.

<table>
<thead>
<tr>
<th></th>
<th>( \eta_{miz} )</th>
<th>( R_{sq} )</th>
<th>( T_s )</th>
<th>( C_u )</th>
<th>( r_d )</th>
</tr>
</thead>
<tbody>
<tr>
<td>M6</td>
<td>36%</td>
<td>0.22 ( \Omega )</td>
<td>300ps</td>
<td>4fF/um(^2)</td>
<td>584um</td>
</tr>
<tr>
<td>M7</td>
<td>68%</td>
<td>0.11 ( \Omega )</td>
<td>300ps</td>
<td>4fF/um(^2)</td>
<td>825um</td>
</tr>
</tbody>
</table>

The calculated actual decoupling distances are far bigger than our initial guess of the radius. The construction of the inner ellipses to determine the 100%, 75%, 50% and 25% effect boundaries is performed with considering the voltage drop profile along the distance. The schematic of the voltage drop versus decoupling distance is shown in Figure 10.8. The blue area in the graph represents the boundaries for the 100% effect; the green area shows the area of 75% effect and so on.
The effects of the improvements can be seen on the correlation graph shown in Figure 10.9.

![Correlation Graph](image)

With the accurate calculation of the ellipse radiiuses the noise index model is improved for the actual delay prediction on Si. The noise index model takes the effect of the switching activity of the surrounding instances on a specific path delay. When the delay of a certain path is measured on the actual silicon, the effect of the local IR-drop on the actual path delay is inevitable, on the other hand when the Spice simulation is performed to measure the delay of the path during the pre-silicon stage, the effects of the IR-drop is too expensive to model. The noise index model tries to validate the effect of IR-drop on the actual path delay by calculating the weighted switching activity of the instances around the path.

10.4. References


11. (UNRM) A derivative-free algorithm for circuit design

The optimal design of a circuit can be formulated as a particular global optimization problem.

The structure of such optimization problem can be described as follows:

\[
\min f(x) \\
\text{s.t. } l \leq x \leq u
\]  

where \( x \in \mathbb{R}^n \) is a set of decision variables, \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is an objective function (performance index) and \( l, u \in \mathbb{R}^n \) and lower and upper bounds on the decision variables.

However, it is worth noting that a circuit design problem has the following distinguishing features:

(i) an explicit mathematical representation of the objective function in terms of the variables vector \( x \) is not available, therefore the first-order derivatives of \( f \) cannot be explicitly calculated or approximated;

(ii) the calculation of the objective function requires high computational cost;

(iii) the objective function may be not available in analytic form.

We now define the distributed derivative-free algorithm for bound constrained mixed-integer problems (see [1] for further details).

---

**Algorithm DFL**

_**Data.**_ \( \theta \in (0, 1) \), \( \xi_0 > 0 \), \( x_0 \in X \cap Z \), \( \alpha_0 > 0 \), \( i \in I_c \), \( \tilde{\alpha}_0 = 1 \), \( i \in I_z \), and set \( d_0^i = \nu^i \) for \( i = 1, \ldots, n \).

For \( k = 0, 1, \ldots \)

- Set \( y_k^i = x_k \).
- For \( i = 1, \ldots, n \)
  - If \( i \in I_c \) then compute \( \alpha \) by the _Continuous search_ \((\tilde{\alpha}_k^i, y_k^i, d_k^i; \alpha)\)
    - If \( \alpha = 0 \) then set \( \alpha_k^i = 0 \) and \( \tilde{\alpha}_k^{i+1} = \theta \tilde{\alpha}_k^i \).
    - Else set \( \alpha_k^i = \alpha \), \( \tilde{\alpha}_k^{i+1} = \alpha \).
  - Else compute \( \alpha \) by the _Discrete Search_ \((\tilde{\alpha}_k^i, y_k^i, d_k^i; \alpha)\)
    - If \( \alpha = 0 \) then set \( \alpha_k^i = 0 \) and \( \tilde{\alpha}_k^{i+1} = \max\{1, [\tilde{\alpha}_k^i / 2]\} \).
    - Else set \( \alpha_k^i = \alpha \), \( \tilde{\alpha}_k^{i+1} = \alpha \).
- Set \( y_k^{i+1} = y_k^i + \alpha_k^i d_k^i \) and \( d_k^{i+1} = d_k^i \).

End For

- If \( (y_k^{n+1})_z = (x_k)_z \) and \( \tilde{\alpha}_k = 1 \), \( i \in I_z \) then set \( \xi_{k+1} = \theta \xi_k \) else set \( \xi_{k+1} = \xi_k \).
- Find \( x_{k+1} \in X \cap Z \) such that \( f(x_{k+1}) \leq f(y_k^{n+1}) \).

End For
The basic ingredients of the method are the Continuous search and Discrete search procedures. They are needed to explore the coordinate directions associated with, respectively, continuous and discrete variables. The current point is updated as soon as a sufficient reduction of the objective function is achieved by one of the procedures. The Continuous search procedure is quite standard in a derivative-free context [2].

<table>
<thead>
<tr>
<th>Continuous search ((\tilde{a}, y, d; \alpha)).</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data.</strong> (\gamma &gt; 0, \delta \in (0, 1)).</td>
</tr>
<tr>
<td><strong>Step 1.</strong> Compute the largest (\tilde{a}) such that (y + \tilde{a}d \in X \cap \mathcal{Z}). Set (\alpha = \min{\tilde{a}, \alpha}).</td>
</tr>
<tr>
<td><strong>Step 2.</strong> If (\alpha &gt; 0) and (f(y + \alpha d) \leq f(y) - \gamma \alpha^2) then go to Step 6.</td>
</tr>
<tr>
<td><strong>Step 3.</strong> Compute the largest (\tilde{a}) such that (y - \tilde{a}d \in X \cap \mathcal{Z}). Set (\alpha = \min{\tilde{a}, \alpha}).</td>
</tr>
<tr>
<td><strong>Step 4.</strong> If (\alpha &gt; 0) and (f(y - \alpha d) \leq f(y) - \gamma \alpha^2) then set (d \leftarrow -d) and go to Step 6.</td>
</tr>
<tr>
<td><strong>Step 5.</strong> Set (\alpha = 0) and return.</td>
</tr>
<tr>
<td><strong>Step 6.</strong> While (\left(\alpha &lt; \tilde{a}\right. \text{ and } \left. f\left(y + \frac{\alpha}{\delta} d\right) \leq f(y) - \frac{\gamma \alpha^2}{\delta^2}\right))</td>
</tr>
<tr>
<td>(\alpha \leftarrow \alpha/\delta).</td>
</tr>
<tr>
<td><strong>Step 7.</strong> Set (\alpha \leftarrow \min{\alpha, \alpha}) and return.</td>
</tr>
</tbody>
</table>

The Discrete search procedure is similar to the Continuous search but the sufficient reduction is governed by a control parameter which is reduced during the optimization process. In particular, the control parameter is reduced whenever no discrete variable has been updated by the Discrete search procedure and the tentative steps along the discrete variables are equal to one. This choice guarantees that the algorithm is convergent towards stationary points of the optimization problem we want to solve.
We use a set of test problems for local optimization. On these problems we compare the performances of the proposed algorithm DFL and of the state-of-the-art solver for derivative-free mixed integer nonlinear optimization NOMAD release 3.4.1. DFL has been implemented in Fortran90. All the codes have been run on an Intel Core2 quad CPU 2.66GHz with 4GB main memory.

In the following table we report, besides the problem name, number of variables (n), objective function value on the starting point (f₀), and, for every code, the attained function value (f*) and the number of function evaluations required to satisfy the stopping condition (nF).

From Table 11.1, we can notice that Algorithm DFL outperforms NOMAD in terms of function evaluations. Furthermore, in terms of final objective function values NOMAD and DFL are comparable.
<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>n</th>
<th>$f_0$</th>
<th>DFL</th>
<th>NOMAD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$f^*$</td>
<td>$n_F^*$</td>
</tr>
<tr>
<td>helval</td>
<td>3</td>
<td>5.06E+01</td>
<td>1.113E+00</td>
<td>66</td>
</tr>
<tr>
<td>biggsExp6</td>
<td>6</td>
<td>3.105E+00</td>
<td>5.741E-03</td>
<td>88</td>
</tr>
<tr>
<td>gaussian</td>
<td>3</td>
<td>1.146E+00</td>
<td>2.595E-02</td>
<td>36</td>
</tr>
<tr>
<td>powbad</td>
<td>2</td>
<td>8.100E+03</td>
<td>9.99E+07</td>
<td>17</td>
</tr>
<tr>
<td>box</td>
<td>3</td>
<td>1.958E+00</td>
<td>4.974E-02</td>
<td>36</td>
</tr>
<tr>
<td>vardim</td>
<td>10</td>
<td>1.187E+06</td>
<td>9.454E+02</td>
<td>81</td>
</tr>
<tr>
<td>watson</td>
<td>9</td>
<td>1.225E+03</td>
<td>2.842E+01</td>
<td>154</td>
</tr>
<tr>
<td>watson12</td>
<td>12</td>
<td>4.924E+02</td>
<td>1.434E+01</td>
<td>316</td>
</tr>
<tr>
<td>penalty1</td>
<td>10</td>
<td>3.022E+01</td>
<td>1.733E-01</td>
<td>81</td>
</tr>
<tr>
<td>penalty2</td>
<td>4</td>
<td>4.965E+01</td>
<td>4.593E-01</td>
<td>33</td>
</tr>
<tr>
<td>penalty10</td>
<td>10</td>
<td>6.828E+02</td>
<td>2.582E+00</td>
<td>83</td>
</tr>
<tr>
<td>brownbad</td>
<td>2</td>
<td>1.282E+12</td>
<td>2.21E+10</td>
<td>58</td>
</tr>
<tr>
<td>brownaden</td>
<td>4</td>
<td>7.946E+06</td>
<td>1.679E+05</td>
<td>45</td>
</tr>
<tr>
<td>gulf</td>
<td>3</td>
<td>2.891E+01</td>
<td>5.925E-01</td>
<td>100</td>
</tr>
<tr>
<td>trig</td>
<td>10</td>
<td>4.834E+01</td>
<td>2.432E-03</td>
<td>196</td>
</tr>
<tr>
<td>banex</td>
<td>2</td>
<td>9.352E-01</td>
<td>2.000E-02</td>
<td>22</td>
</tr>
<tr>
<td>powelex</td>
<td>4</td>
<td>4.368E+01</td>
<td>1.210E+00</td>
<td>37</td>
</tr>
<tr>
<td>beale</td>
<td>2</td>
<td>2.706E+01</td>
<td>2.240E-02</td>
<td>34</td>
</tr>
<tr>
<td>wood</td>
<td>4</td>
<td>1.002E+02</td>
<td>1.639E+00</td>
<td>39</td>
</tr>
<tr>
<td>hs1</td>
<td>2</td>
<td>3.031E+03</td>
<td>1.257E+00</td>
<td>45</td>
</tr>
<tr>
<td>hs2</td>
<td>2</td>
<td>6.340E+02</td>
<td>1.257E+00</td>
<td>42</td>
</tr>
<tr>
<td>hs3</td>
<td>2</td>
<td>1.000E-03</td>
<td>0.000E+00</td>
<td>30</td>
</tr>
<tr>
<td>hs4</td>
<td>2</td>
<td>3.199E+00</td>
<td>2.667E+00</td>
<td>10</td>
</tr>
<tr>
<td>hs5</td>
<td>2</td>
<td>5.125E+02</td>
<td>7.711E-01</td>
<td>48</td>
</tr>
<tr>
<td>hs25</td>
<td>3</td>
<td>3.283E+01</td>
<td>2.250E-03</td>
<td>454</td>
</tr>
<tr>
<td>hs38</td>
<td>4</td>
<td>4.200E-01</td>
<td>4.154E+01</td>
<td>80</td>
</tr>
<tr>
<td>hs45</td>
<td>5</td>
<td>1.509E+00</td>
<td>1.000E+00</td>
<td>50</td>
</tr>
<tr>
<td>hs220</td>
<td>2</td>
<td>1.000E+00</td>
<td>7.711E-01</td>
<td>33</td>
</tr>
<tr>
<td>hs242</td>
<td>3</td>
<td>7.660E+01</td>
<td>0.000E+00</td>
<td>45</td>
</tr>
<tr>
<td>hs247</td>
<td>3</td>
<td>8.100E+01</td>
<td>4.930E-30</td>
<td>55</td>
</tr>
<tr>
<td>hs257</td>
<td>4</td>
<td>4.200E+01</td>
<td>3.571E+01</td>
<td>90</td>
</tr>
<tr>
<td>hs307</td>
<td>2</td>
<td>2.567E+02</td>
<td>1.935E+02</td>
<td>29</td>
</tr>
<tr>
<td>hs328</td>
<td>2</td>
<td>1.770E+00</td>
<td>1.741E+00</td>
<td>28</td>
</tr>
<tr>
<td>hs357</td>
<td>4</td>
<td>1.000E+20</td>
<td>5.700E+02</td>
<td>388</td>
</tr>
<tr>
<td>hs358</td>
<td>5</td>
<td>1.227E+01</td>
<td>2.832E-01</td>
<td>183</td>
</tr>
<tr>
<td>hs368</td>
<td>8</td>
<td>0.000E+00</td>
<td>-1.000E+00</td>
<td>63</td>
</tr>
</tbody>
</table>

**Table 11.1.** Comparison between DFL and NOMAD.
11.1. References


Conclusion and Future Work

In this document we have shown standardized PV-aware tools for simulation of digital blocks, AMS&RF blocks, and NVM arrays.

UNBO implemented an Adaptive Body Bias technique to reduce the performance impact of Vth variations by playing with body bias voltage regulation, and investigated in depth the cost/benefits of this technique. The technique has been applied on a digital block provided by ST-I (WP5) and the achieved results show a speed-up improvement from 12% to 15%.

UNRM have reported first results of a derivative-free algorithm for bound constrained mixed-integer problems. The results show the good behavior of the proposed algorithm. Forthcoming activities will be addressed to use the derivative-free approach in a circuit design context, with reference to design problems provided by ST-I.

The comparison of NMX and UNGL (see Table 4-6 and Table 4-10) methodologies show a very good agreement, thus reinforcing the confidence on the simulation industrial flow internally developed by NMX. The perspective for the next deliverable is to further investigate NMX methodology by leveraging the cooperation with UNGL and their tool infrastructure.

NXP presents the approach and the methods that are developed for the investigation of the switching activity on Trimedia block. The detailed understanding of switching activity of a design and its relation with respect to time and space is important for both designers and test engineers. Statistical vector-less approaches to investigate switching activity can be misleading because they consider a uniform toggling on every net of a design and try to identify the hot spots by looking to the structure of the design. However the activity profile of a design can be very different depending on the inputs applied. For example, this investigation showed the big difference on the amount of the switching instances for two different test vectors. The activity profile of the design was much higher during the application of the stuck-at test vectors. The activity, coordinate and hazard analysis tools are developed to investigate switching activity of a design. The run time of all the three programs is in the order of minutes (approximately 4-5 minutes) for the TM3271 block. However to be able to run these programs a VCD file is required as an input. The VCD files can be generated with using commercial logic simulator tools.

With the developed prototype tools, one can study the activity distribution in time and in space; one can decide the contribution of the hazards to the overall total transitions. The investigation is performed for a variety of path delay test vectors and stuck at test vectors. The analysis of the real mode application is done at RT-Level. However RTL description of a design does not have any information about the gates. As a result, the activity analysis on a RTL VCD will only give the switching information on the pins.

In addition to the switching activity investigation of the TM3271, the noise index model is developed in order to understand the timing discrepancy between the silicon measurements and the pre-silicon simulation stage. The delays of different paths are measured on Si under typical conditions and the delays of the same paths are predicted with Spice level timing analysis during pre-silicon stage under same conditions. The predicted and actual path delays show some differences, certain numbers of paths are slower than expected and certain numbers of paths are faster than expected. The noise index model tries to consider the effect of the switching cells on the actual path delays of the circuit. The noise index model calculates the weighted switching activity around a specific path and gives to the path a specific noise index. Experiments are performed on a variety of paths which have been selected randomly from the faster and slower than expected paths. For the selected paths their noise index values are calculated. A strong correlation is reported between the noise index of the paths and the delay difference of these paths.
# 12. Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODERN</td>
<td>MOdeling and DEsign of Reliable, process variation-aware</td>
</tr>
<tr>
<td>CMOS</td>
<td>Complementary Metal Oxide Semiconductor technology</td>
</tr>
<tr>
<td>ABB</td>
<td>Adaptive body bias</td>
</tr>
<tr>
<td>FBB</td>
<td>Forwarding body bias</td>
</tr>
<tr>
<td>RBB</td>
<td>Reverse Body Bias</td>
</tr>
<tr>
<td>NBB</td>
<td>No body bias</td>
</tr>
<tr>
<td>AVS</td>
<td>Adaptive voltage scaling</td>
</tr>
<tr>
<td>ILP</td>
<td>Integer Linear</td>
</tr>
<tr>
<td>PSG</td>
<td>PolySilicon Granularity</td>
</tr>
<tr>
<td>HKG</td>
<td>High-k Granularity</td>
</tr>
<tr>
<td>ICWB</td>
<td>IC Workbench (Synopsys)</td>
</tr>
<tr>
<td>DD</td>
<td>Drift Diffusion</td>
</tr>
<tr>
<td>MC</td>
<td>Monte Carlo</td>
</tr>
</tbody>
</table>