SSTA Design Methodology for Low Voltage Operation

by

Rahul Rithe

B. Tech. (Honors), Indian Institute of Technology Kharagpur (2008)

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering

at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2010

© Massachusetts Institute of Technology 2010. All rights reserved.

Author ................................................

Department of Electrical Engineering and Computer Science

May 13, 2010

Certified by ...........................................

Anantha P. Chandrakasan
Professor of Electrical Engineering
Thesis Supervisor

Certified by ...........................................

Dennis Buss
Chief Scientist, Texas Instruments
Thesis Supervisor

Accepted by ...........................................

Terry P. Orlando
Chairman, Department Committee on Graduate Theses
Abstract

Statistical process variations have long been an important design issue. But until recently, process variations have been global process variations, i.e., transistor parameters may vary from die to die but are constant within a die. With transistor geometries shrinking below 65nm, however, a new kind of statistical variation, known as Local or Intra-die variation, has become important for logic and memory.

Local variations are primarily the result of variations in the number of dopant atoms in the channel of CMOS transistors. To achieve ultra-low power, ICs are being designed for $V_{DD} \leq 0.5V$. At these voltages, the stochastic delay resulting from local variations has standard deviation comparable to the nominal delay.

In order to predict the statistical impact of local variations on circuit performance, it is necessary to develop the statistical models that accurately reflect local variations and to develop a computationally efficient algorithm for performing SSTA using these models. At low voltage ($V_{DD} \leq 0.5V$), circuit delay is a non-linear function of the transistor random variables. This greatly complicates the statistical analysis because the PDF of the circuit delay is non-Gaussian. Most of the current SSTA approaches that can handle non-Gaussian PDFs, have high computational complexities.

In this work, a complete SSTA design methodology for local variations in logic timing at low voltage operation is presented. The approach can handle non-linear delays with non-Gaussian delay PDFs in a computationally efficient manner. The approach has been implemented using commercial CAD tools and integrated into commercially used IC design flow. Comparison with Monte-Carlo analysis demonstrates high accuracy of the approach.

Thesis Supervisor: Anantha P. Chandrakasan
Title: Professor of Electrical Engineering
Thesis Supervisor: Dennis Buss
Title: Chief Scientist, Texas Instruments
Acknowledgments

First, I would like to thank my supervisor, Professor Anantha Chandrakasan for being a great advisor and a role model. Researching and writing this thesis under his supervision has been an invaluable experience in the pursuit of academic research.

This project would not be what it is without the constant guidance and encouragement of Dr. Dennis Buss. I am deeply thankful to Dennis for everything from the numerous discussions about every aspect of the project to reviewing this thesis.

The summer internship at Texas Instruments was a great experience for me. I am grateful to Alice Wang for supervising my work at TI. I would also like to thank Jie Gu, Satyendra Datla and Gordon Gammie from TI, for their valuable suggestions that helped shape the project.

I would like to thank Texas Instruments for providing fabrication services for the testchip as well as their technical help in the design. I would also like acknowledge the MIT Presidential Fellowship for awarding me the Irwin Mark Jacobs and Joan Klein Jacobs Presidential Fellowship during my first year at MIT.

I am extremely grateful to all the members of Ananthagroup for creating one of the best work environments and for always being ready to help. I must also thank Saurav Bandyopadhyay and Rishabh Singh for being great friends and apartment mates throughout the last two years.

Finally, it is too trivial and inconsequential to say thank you to my parents and my sister Bhagyashree, for everything they have done for me. Without their support through all my endeavours and encouragement to follow my dreams, none of this would ever be possible.

Rahul Rithe
Cambridge, MA
01 MAY 2010
Contents

1 Introduction .................................................. 15
   1.1 Local Random Variations .............................. 16
   1.2 Previous SSTA Work .................................. 17
   1.3 Problem Statement .................................. 20
   1.4 Contributions of this work ......................... 21
   1.5 Thesis organization ................................ 23

2 Cell Characterization ...................................... 25
   2.1 NLOPALV Theory for Cell Characterization ....... 26
       2.1.1 Dimensionality Reduction ..................... 27
       2.1.2 Operating Point ............................... 28
   2.2 Cell Characterization Flow ......................... 32
   2.3 Run Time Analysis ................................ 33
   2.4 Results ............................................... 36
   2.5 Characterization Trade-off: Sigma Spacing vs. Accuracy .......................... 38
   2.6 Cell Delay PDF: Trends .............................. 39
       2.6.1 Dependance on $V_{DD}$ ......................... 39
       2.6.2 Dependance on Cell Size ...................... 41
       2.6.3 Dependance on Drive Strength ................. 41
   2.7 Summary ............................................... 42

3 Timing Path Analysis ...................................... 45
   3.1 Introduction ......................................... 45
3.2 NLOPALV Theory for Timing Path Analysis .................. 47
  3.2.1 Linear - Gaussian Theory .......................... 47
  3.2.2 Non-linear - Non-Gaussian Theory .................... 52
3.3 Multi-Stage Timing Path Analysis .......................... 56
  3.3.1 Correlation due to Slew Propagation ................... 56
  3.3.2 NLOPALV Algorithm for TP Analysis .................. 59
  3.3.3 Integration with the CAD Flow ....................... 60
  3.3.4 Results ........................................... 60
  3.3.5 Trade-off: Number of Iterations vs. Accuracy ........ 63
3.4 Timing Path Setup and Hold Analysis ........................ 65
3.5 Summary .............................................. 68

4 Timing Closure Flow for ICs ................................. 71
  4.1 Reducing Number of Critical Paths ....................... 71
    4.1.1 Step - 1: Elimination by Overly Pessimistic Estimate .. 72
    4.1.2 Step - 2: Elimination by Pessimism Reduction .......... 75
  4.2 Timing Closure for ICs ................................ 78
  4.3 Summary .............................................. 79

5 Case Study: Reconfigurable Transform for Video Coding .......... 81
  5.1 Transform Coding ..................................... 82
    5.1.1 H.264/AVC Transform: Key Features .................. 82
    5.1.2 VC-1 Transform: Key Features ....................... 83
    5.1.3 H.264/AVC Integer Transform ....................... 84
    5.1.4 VC-1 Integer Transform ............................ 86
  5.2 Reconfigurable Transform Design ......................... 87
    5.2.1 Structural Similarity of the Transforms ............. 87
    5.2.2 Symmetry of the Transform Matrix ................... 88
  5.3 Low Voltage Transform ................................ 89
    5.3.1 Motivation for Low Voltage operation ............... 89
    5.3.2 Transform IC Implementation ....................... 90
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4 Timing Analysis using SSTA Methodology</td>
<td>90</td>
</tr>
<tr>
<td>5.5 Summary</td>
<td>93</td>
</tr>
<tr>
<td>6 Conclusions</td>
<td>95</td>
</tr>
<tr>
<td>6.1 Key Features</td>
<td>95</td>
</tr>
<tr>
<td>6.2 Summary of Results</td>
<td>96</td>
</tr>
<tr>
<td>6.3 Limitations</td>
<td>98</td>
</tr>
<tr>
<td>6.4 Future Work</td>
<td>98</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>A Cell/Arc for one of the transistors of a 2-input NAND gate</td>
<td>27</td>
</tr>
<tr>
<td>2-2</td>
<td>Gaussian mapping of non-Gaussian PDF through CADF</td>
<td>29</td>
</tr>
<tr>
<td>2-3</td>
<td>Typical sensitivity curves for an Inverter</td>
<td>30</td>
</tr>
<tr>
<td>2-4</td>
<td>Cell Operating Point</td>
<td>32</td>
</tr>
<tr>
<td>2-5</td>
<td>Cell Characterization Flow</td>
<td>34</td>
</tr>
<tr>
<td>2-6</td>
<td>Cell Library Format (.lib)</td>
<td>35</td>
</tr>
<tr>
<td>2-7</td>
<td>Cell characterization run-time</td>
<td>36</td>
</tr>
<tr>
<td>2-8</td>
<td>Typical cell delay PDF</td>
<td>36</td>
</tr>
<tr>
<td>2-9</td>
<td>Cell characterization accuracy: NLOPALV vs. Monte-Carlo</td>
<td>37</td>
</tr>
<tr>
<td>2-10</td>
<td>Characterization trade-off: sigma spacing vs. accuracy</td>
<td>39</td>
</tr>
<tr>
<td>2-11</td>
<td>Variation of standard deviation (1-sigma) of the stochastic delay with</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>$V_{DD}$</td>
<td></td>
</tr>
<tr>
<td>2-12</td>
<td>Variations in the PDFs of stochastic delays normalized by the corresponding</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>standard deviations with $V_{DD}$</td>
<td></td>
</tr>
<tr>
<td>2-13</td>
<td>Variation of stochastic delay PDF with cell size</td>
<td>41</td>
</tr>
<tr>
<td>2-14</td>
<td>Variation of stochastic delay PDF with cell drive strength</td>
<td>42</td>
</tr>
<tr>
<td>3-1</td>
<td>Defining $f$-sigma in the non-Gaussian context</td>
<td>48</td>
</tr>
<tr>
<td>3-2</td>
<td>Gaussian cell delays with linear TP delay</td>
<td>49</td>
</tr>
<tr>
<td>3-3</td>
<td>Non-Gaussian cell delays with non-linear TP delay</td>
<td>53</td>
</tr>
<tr>
<td>3-4</td>
<td>TP delay curve in $\xi$-space</td>
<td>54</td>
</tr>
<tr>
<td>3-5</td>
<td>CAD Flow for NLOPALV Timing Path Analysis</td>
<td>61</td>
</tr>
<tr>
<td>Figure No.</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3-6</td>
<td>Performance comparison for NLOPALV vs. Monte-Carlo</td>
<td>62</td>
</tr>
<tr>
<td>3-7</td>
<td>Delay PDF for a TP from the 28nm DSP at $V_{DD} = 0.5V$: The zero-sigma delay is the nominal delay. The Gaussian approximation is chosen such that the standard deviation for the Gaussian is same as the 1-sigma delay for NLOPALV</td>
<td>62</td>
</tr>
<tr>
<td>3-8</td>
<td>Variation of Operating Point for a 5-stage TP across iterations at $V_{DD} = 0.5V$</td>
<td>64</td>
</tr>
<tr>
<td>3-9</td>
<td>Accuracy of the 3-sigma TP delay across number of iterations at $V_{DD} = 0.5V$</td>
<td>65</td>
</tr>
<tr>
<td>3-10</td>
<td>Typical Timing Path for Setup/Hold Analysis</td>
<td>65</td>
</tr>
<tr>
<td>3-11</td>
<td>Setup/Hold Analysis Flow</td>
<td>67</td>
</tr>
<tr>
<td>3-12</td>
<td>Setup/Hold Analysis: NLOPALV vs. Monte-Carlo at $V_{DD} = 0.5V$</td>
<td>68</td>
</tr>
<tr>
<td>4-1</td>
<td>Typical Timing Path</td>
<td>72</td>
</tr>
<tr>
<td>4-2</td>
<td>$f$-sigma TP delay as a combination of individual cell delays at the operating point. Considering $f$-sigma delay for each cell leads to overly pessimistic TP delay estimate</td>
<td>73</td>
</tr>
<tr>
<td>4-3</td>
<td>Timing Closure Flow: Step -1</td>
<td>74</td>
</tr>
<tr>
<td>4-4</td>
<td>Start-End pairs with multiple logic paths</td>
<td>76</td>
</tr>
<tr>
<td>4-5</td>
<td>Timing Closure Flow: Step -2</td>
<td>77</td>
</tr>
<tr>
<td>4-6</td>
<td>SSTA Design Methodology for Low Voltage: Timing Closure Flow for an IC</td>
<td>80</td>
</tr>
<tr>
<td>5-1</td>
<td>Delay / Energy variations for the Transform module as a function of $V_{DD}$</td>
<td>90</td>
</tr>
<tr>
<td>5-2</td>
<td>Layout of the shared, reconfigurable Transform chip</td>
<td>91</td>
</tr>
<tr>
<td>5-3</td>
<td>Timing Path Delay PDF for one of the critical paths from the Transform IC at $V_{DD} = 0.5V$</td>
<td>92</td>
</tr>
</tbody>
</table>
List of Tables

1.1 Comparison of Gaussian Approximation vs. Monte-Carlo Analysis . . 19
1.2 Summary of Previous SSTA Works ................................. 20
2.1 Comparison of NLOPALV, Monte-Carlo and Gaussian SSTA ...... 38
3.1 NLOPALV: Cell Characterization vs. Timing Path Analysis ....... 46
3.2 Performance Comparison of NLOPALV vs. Monte-Carlo and Gaussian Approximation at $V_{DD} = 0.5V$ ............................. 63
3.3 Setup/Hold Analysis: NLOPALV vs. Monte-Carlo at $V_{DD} = 0.5V$ . 68
4.1 Timing Closure Flow: Step - 1 Analysis Results at $V_{DD} = 0.5V$ . . 75
4.2 Timing Closure Flow: Complete Analysis Results at $V_{DD} = 0.5V$ . 78
5.1 Timing Closure Flow on the Transform IC at $V_{DD} = 0.5V$ ......... 92
5.2 Detailed Timing Path Analysis on Transform IC at $V_{DD} = 0.5V$ . . 93
Chapter 1

Introduction

Statistical process variations have long been an important design issue. But until recently, process variations have been global process variations, i.e., transistor parameters may vary from die to die but are constant within a die [1, 14]. With transistor geometries shrinking below 65nm, however, a new kind of statistical variation has become important for logic [4], called local statistical variation. There are three categories of process variations that are important in design of modern CMOS logic [16].

1. Global random variation in gate length, gate width, flatband voltage, oxide thickness and channel doping. Global random variations are assumed to be random from lot to lot, water to wafer, and die to die, but they are assumed to be the same for all transistors within a die.

2. Systematic or predictable variations, such as variations in litho or etch or Chemical Mechanical Polishing (CMP), as a result of variations in the local environment. This category also includes non-random variations due to Channel Hot Carrier (CHC) and Negative Bias Temperature Instability (NBTI).

There are many sources of systematic variations:

- *Optical proximity effects*: There are variations in the gate length of devices
in close proximity, caused by diffraction light during lithography, because the wavelength of light is larger than the device feature size. However, these variations are predictable from the layout.

- **Etch loading effects**: They cause etch rates to be a function of pattern density, and affect transistor parameters in a systematic and predictable manner.

- **Mechanical stress**: Stress from one transistor from neighboring transistors change the transistor parameters in a way that depends on the local environment. These effects also are predictable.

- **CMP effects**: Cu CMP depends on line density and line width, with the result that Cu thickness depends on layout. These effects can be predicted and incorporated into post-layout parasitic extraction.

- **Edge effects**: Devices near the edges tend to have different gate lengths than those in the center of the die. However, this variation is also systematic.

The SSTA methodology presented here does not address systematic variations. Conventional methods for dealing with systematic variation need to be used in conjunction with our approach in order to achieve accurate timing simulation.

3. Local random variations in transistor parameters. Local random variations are assumed to be random from one transistor to another within a die. This work deals with the effect that local variation in CMOS transistor parameters has on logic timing at low voltage ($V_{DD} \leq 0.5V$).

### 1.1 Local Random Variations

Local variations are primarily the result of variations in the number of dopant atoms in the channel of CMOS transistors [3, 2]. Based on the physics for the implant
process, local variations have the following characteristics:

1. MOS $V_T$ is the primary transistor random variable

2. PDFs of $V_T$ are Gaussian and statistically independent from one transistor to another

3. The standard deviation of the $V_T$ PDF follows Pelgroms Law [18] $\sigma V_t = \frac{M}{\sqrt{LW}}$, where L & W are the transistor length and width and M is a constant that depends on technology, but not on transistor dimensions.

Local variations have long been known in analog design and in SRAM design [5]. In analog design, local variations are called mismatch because of the mismatch in the $V_T$ of adjacent transistors. But they have not generally been a problem for logic because the parameter variations in different transistors are statistically independent and they add approximately as $\sqrt{N}$, where N is the number of transistors in a circuit. This means that the variation in Timing Path (TP) delay, as a percentage of nominal delay goes like $\frac{1}{\sqrt{N}}$. However, two trends are converging that will make local variations increasingly important for logic:

1. Transistor geometries are shrinking, with the result that $\sigma V_t$ is increasing. (Pelgroms Law) $\sigma V_t$ of about 25-50 mV is not uncommon for modern CMOS.

2. To achieve ultra-low power for applications like portable multimedia devices and distributed sensor networks, ICs are being designed for $V_{DD} \leq 0.5V$. At these voltages, the stochastic delay resulting from local variations has standard deviation comparable to the nominal delay.

### 1.2 Previous SSTA Work

In order to predict the statistical impact of local variations on circuit performance, it is necessary to develop the statistical models that accurately reflect local variations and to develop a computationally efficient algorithm for performing SSTA using these
models. At nominal voltage, it is usually accurate to assume that circuit performance (propagation delay) is linear in transistor variation [2]. In this case, the circuit delay is Gaussian, and the standard deviation can be readily calculated from the standard deviations of the transistor parameters. However, at low voltage ($V_{DD}$ at or below 0.5V), circuit delay is a non-linear function of the transistor random variables. This greatly complicates the statistical analysis because the PDF of the circuit delay is no longer Gaussian [8].

The traditional corner based analysis is not enough to correctly determine the performance of the integrated circuits. This has necessitated the development of new SSTA design techniques to address local variations. Several approaches for SSTA have been proposed ranging from numerical integration techniques [10] to Monte-Carlo based techniques [21, 23] to those based on probabilistic analysis. In [11], an approach to determine the exact distribution of the combinational circuits, given the PDFs of gate delays and wires is proposed. This approach also handles false paths in the design to avoid unnecessary pessimism. The approach first determines the lengths of possible critical paths and then performs Monte-Carlo analysis on these possible critical paths.

Though the methods based on numerical integration and Monte-Carlo analysis can provide a high level of accuracy, practical use of these methods is prohibitive because of the very high computational costs involved in these methods. This has led to the majority of research on SSTA being focused on probabilistic analysis based techniques [8, 6, 7, 13, 27, 15, 20, 24, 26]. However, many of the approaches consider the delay PDF to be Gaussian and the delay to be a linear function of the variation sources [7, 15, 24]. This approximations works well when the circuits are operated at nominal $V_{DD}$. However, with continuous scaling of technology and desired ultra-low voltage operation of the circuits for low-power applications, this assumption can no longer be justified. Particularly, for circuits operated at $V_{DD} = 0.5V$ or below, Gaussian approximation can lead to high errors in the timing analysis. Table 1.1 shows a
Table 1.1: Comparison of Gaussian Approximation vs. Monte-Carlo Analysis

<table>
<thead>
<tr>
<th></th>
<th>Monte-Carlo</th>
<th>Gaussian Approx.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$V_{DD} = 1V$</td>
<td>$V_{DD} = 0.5V$</td>
</tr>
<tr>
<td><strong>NAND2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of SPICE Sims.</td>
<td>10,000</td>
<td>20</td>
</tr>
<tr>
<td>%Error</td>
<td>0% (Reference)</td>
<td>7% 90%</td>
</tr>
<tr>
<td><strong>Flip-Flop</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of SPICE Sims.</td>
<td>100,000</td>
<td>2,000</td>
</tr>
<tr>
<td>% Error</td>
<td>0% (Reference)</td>
<td>9% 85%</td>
</tr>
</tbody>
</table>

comparison between the Monte-Carlo approach and the Gaussian approximation\(^1\) approach in terms of run-time and accuracy at $V_{DD} = 1V$ and $V_{DD} = 0.5V$ for cells designed using commercial 28nm CMOS technology.

As the comparison shows, Gaussian approximation works well at $V_{DD} = 1V$, but it completely fails to capture the non-linearities at $V_{DD} = 0.5V$.

This comparison clearly suggests the need for an approach that is significantly faster than the Monte-Carlo analysis and significantly more accurate compared to the Gaussian approximation. More recently, attempts have been made to address this issue of non-linear delay variations resulting in non-Gaussian PDFs in global variations. Most of these methods rely on Taylor series expansion based polynomial representations to model the cell and timing path delays. The major problem in such representations is that of high computational complexity in performing the MAX operation.

A summary of some of these approaches is presented in Table 1.2.

\(^1\)Gaussian approximation is made by characterizing the cell at zero sigma and one sigma points and then making a linear approximation

\(^2\)Flip-Flop characterization requires almost 10X SPICE simulations compared to a combinational gate because of the setup/hold constraint arcs.
Table 1.2: Summary of Previous SSTA Works

<table>
<thead>
<tr>
<th>Publication</th>
<th>Contributions</th>
<th>Open Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>[11]</td>
<td>Exact PDF, Handles False Paths, Reduces number of paths</td>
<td>Requires Monte-Carlo on a subset of original paths</td>
</tr>
<tr>
<td>[20]</td>
<td>Handles non-Gaussian PDFs using principal component analysis</td>
<td>Delay is considered a linear function of variation parameters</td>
</tr>
<tr>
<td>[27]</td>
<td>Handles non-linear delays, Quadratic delay model</td>
<td>MAX operations assumes inputs to be Gaussian</td>
</tr>
<tr>
<td>[26]</td>
<td>Handles non-Gaussian delay PDFs, Quadratic delay model</td>
<td>MAX operation requires numerical integration</td>
</tr>
<tr>
<td>[22]</td>
<td>Standard cell characterization for local variations, Accounts for switching and non-switching devices</td>
<td>Gives upper bounds. Delay is assumed linear</td>
</tr>
</tbody>
</table>

1.3 Problem Statement

The effect of local variations is very different from the effect of Global variations. Whereas global variations in delay add linearly, local variations do not. Let us assume, for a moment that the PDF of each cell is Gaussian. If we have a number of such cells having local stochastic delay characterized by standard deviation of \(\sigma_i\), the delays add in quadrature, meaning that the standard deviation of the TP delay is given by:

\[
\sigma_{TP} = \sqrt{\sum_{i=1}^{N} \sigma_i^2}
\] (1.1)
If we add the $\sigma_i$ linearly, we would get an overly pessimistic result.

Local (or intra-die or within-die) variations in transistor $V_T$ contribute stochastic variation in logic delay that is a large percentage of the nominal delay. Moreover, when circuits are operated at low voltage ($V_{DD} \leq 0.5V$), the standard deviation of gate delay becomes comparable to nominal delay, because as $V_{DD}$ moves closer and closer to the threshold voltage, small variations in the random variables result in large variations in the device behavior. Furthermore, the Probability Density Function (PDF) of the gate delay is highly non-Gaussian at such low voltages. For CMOS feature size of 65 nm and below, accurately predicting the performance of an integrated circuit in the presence of local variations motivates further research in developing a computationally efficient approach for SSTA. Such an approach should be able to handle non-linear delays resulting in non-Gaussian delay PDFs. It should also be able to integrate into the existing CAD flow for it to be effectively used in the design process.

1.4 Contributions of this work

This work is aimed at developing a computationally efficient algorithm that can perform accurate Path-based SSTA in the regime where delay is a highly non-linear function of the random variables and/or the PDFs of the random variables are non-Gaussian.

- While performing timing closure on a chip, the statistical circuit performance is generally evaluated in terms of the 3-sigma delay for the most critical path. This is to ensure that the design meets all setup/hold constraints with a probability of more than 99.865%. So, generally we are not interested in the entire PDF of the timing path delay. In this work, we take advantage of this fact that we are only interested in computing a specific 3-sigma (or in general, $f$-sigma) delay value and not the entire delay PDF, to achieve major computational savings.
The concept of Operating Point is introduced, which has the advantage that any $f$-sigma stochastic delay of a timing path can be determined by a single calculation of the timing path delay in which the delay $D_i$ if each cell is set to the $f$-sigma operating point. We call this approach the Non-Linear Operating Point Analysis for Local Variations in Logic Timing (NLOPALV) [19].

The NLOPALV approach is implemented using commercial CAD tools and integrated into commercially used IC design flow.

The approach is verified on the logic paths taken from a Digital Signal Processor (DSP) implemented using commercial 28nm CMOS technology and operating at 0.5V.

The approach is used in the design of an IC implementing a shared, reconfigurable Transform Engine for H.264 and VC-1 video coding standards. The design is implemented using commercial 28nm CMOS technology and analyzed for setup/hold constraints to achieve timing closure at $V_{DD} = 0.5V$.

This approach has led to the development of a complete SSTA design methodology to handle local variations in logic timing at low voltage operation. The methodology consists of following steps:

1. Standard Cell Library Characterization

   - The probability density function (PDF) of stochastic delay and the most probable output slew associated with that delay, for each arc of each cell in the standard cell library is characterized. Each timing arc is defined by the input slew and output load.

   - The basics of cell characterization approach were developed in [9]. In this work, the approach has been extended to setup/hold constraint arcs and different run-time vs. accuracy trade-offs have been analyzed. Significant reduction in cell characterization run time has been achieved by algorithmic optimizations.
2. Timing Path Analysis

- *Reducing number of Critical Paths to be analyzed:* Any practical IC has a huge number of timing paths. In order to apply the path-based SSTA analysis to perform timing closure on an entire chip, it is essential that we reduce the number of timing paths that need to be analyzed. It can be shown that the majority of these timing paths have negligible probability of becoming critical even in presence of variations. The concepts of NLOPALV approach are extended to eliminate such paths and thus reduce the number of potentially critical timing paths.

- *Timing Closure:* The potentially critical paths are analyzed in detail using the NLOPALV approach to determine the $f$-sigma setup and hold time in a computationally efficient manner.

### 1.5 Thesis organization

The rest of the thesis is organized as follows:

- *Chapter 2:* The theory underlying the Non-linear Operating Point Analysis as it applies to the cell characterization is described. The algorithmic implementation of the NLOPALV approach for cell characterization is outlined and the results of cell characterization for a standard cell library implemented using commercial 28nm CMOS technology are presented. Run-time vs. accuracy trade-offs are analyzed.

- *Chapter 3:* The theory underlying the Non-linear Operating Point Analysis as it applies to the timing path analysis is described. The algorithmic implementation of the NLOPALV approach for timing path analysis is outlined. Integration of NLOPALV approach in the commercial IC design flow is described. The results for timing paths taken from a DSP implemented using commercial 28nm CMOS technology are presented.
• Chapter 4: Use of NLOPALV approach for reducing the number of critical paths in the design that need to be analyzed and timing closure for an IC is described.

• Chapter 5: A case study of shared, reconfigurable transform for H.264/AVC and VC-1 video coding standards is considered. The SSTA design methodology is used in the design flow for timing closure of the IC in a commercial 28nm CMOS technology at $V_{DD} = 0.5V$.

• Chapter 6: Key features and limitations of the SSTA design methodology are outlined, results are summarized and possible future directions are proposed.
Chapter 2

Cell Characterization

The first step in calculating the effect of local variations on logic timing is the characterization of standard logic cells for local variation. This chapter aims to introduce the reader to the theory underlying the NLOPALV approach for cell characterization, algorithmic implementation of the approach, trade-offs involved in characterization and the trends shown by cell delay PDFs depending on factors such as cell size, drive strength and $V_{DD}$.

The SSTA approach for standard cell characterization using operating point analysis was first introduced in [9]. This work presented a qualitative analysis of the operating point theory for cell characterization. An experimental setup was developed to perform cell characterization for combinational logic gates. This setup made use of SPICE for characterization and the flow was scripted in the C programming language. The results of comparison with the Monte-Carlo analysis were presented and showed good accuracy of the operating point approach for cell characterization. In this work, it was noted that the transistor stacking in standard cells degrades the accuracy of the operating point approach, whereas higher gate drive strength improves the accuracy.

In the work presented here, we extend the operating point approach for cell characterization to sequential logic cells. We automate the cell characterization flow and
optimize the implementation to reduce the characterization run time. Compared to [9], we eliminate the redundancy by reordering the computations. For example, to characterize the cell at different sigma values, a new operating point needs to be determined at each characterization point. However, the sensitivity curves for each transistor, from which the operating points are determined, are independent of the characterization point and need to be computed only once. Repeated computations at the same point are avoided by performing all measurements (such as, rising arc delay/slew, falling arc delay/slew) at that point in a single SPICE simulation. SPICE constructs such as .modify are used to evaluate the delays and slews at different values of random variables, instead of running new SPICE simulations at each point.

In this work, we also take a detailed look at the factors that affect the accuracy of the cell characterization.

2.1 NLOPALV Theory for Cell Characterization

The NLOPALV approach for cell characterization builds on two basic ideas:

1. Dimensionality Reduction: The stochastic delays for individual cells depend on all the random variables in the individual cells. Considering that each transistor has two random variables, the dimensionality of the problem can be enormously high. Gaussian mapping of a non-Gaussian function, described in section 2.1.1, will allow us to significantly reduce the dimensionality of the problem.

2. Operating Point: The concept of Operating Point, introduced in section 2.1.2, is the key concept behind the NLOPALV approach. This concept will allow us to deal with non-linearities in an accurate and computationally efficient manner.

This section considers each of these ideas in detail.

For any given cell, an arc is defined by the input trigger edge (either rising or falling), input slew rate, and output capacitive load. Each cell/arc has an associated stochastic delay and stochastic output slew, which both result from random variations in
transistor parameters relative to their nominal values at the global corner. Fig. 2-1 shows one such cell/arc for a 2-input NAND gate. A similar cell/arc is defined for all the other transistors in the cell as well.

![A Cell/Arc for one of the transistors of a 2-input NAND gate](image)

Figure 2-1: A Cell/Arc for one of the transistors of a 2-input NAND gate

The goal of cell characterization is to determine the probability density function (PDF) of the stochastic cell delay for each arc of each cell (each cell/arc) in the library, as well as the most probable output slew associated with every value of cell delay.

### 2.1.1 Dimensionality Reduction

Variations in transistor mismatch parameters are specified in a SPICE model as independent zero-mean Gaussians, where there are \( N_{rv} \) variables per transistor. In this study \( N_{rv} = 2 \), though theoretically \( N_{rv} \) can be any arbitrary number depending on the transistor model. For a cell consisting of \( N_{tr} \) transistors, there are \( N = N_{rv}N_{tr} \) transistor random variables which we designate as \( x_i \), for \( i = 1, 2, \ldots N \). These random parameters of variation are assumed to be Gaussian and statistically independent.
with standard deviation $\sigma$. It is common to assume that the physical parameters follow Gaussian distribution, because Central Limit Theorem [17] suggests that the sum of arbitrarily distributed independent random variables asymptotically converges to a Gaussian distribution as the number of variables increases. In reality convergence occurs for 10-15 components. This justifies describing the variation of many physical process parameters encountered in IC manufacturing as Gaussian [16].

The goal of cell characterization is to determine the PDF of the delay for each arc of each cell. This is equivalent to determining the cell-arc delay function (CADF), as shown in Fig. 2-2.

Let the PDF of the delay be $P_D(D)$. For the $P_D(D)$ of each cell/arc, we can define a Cell/Arc Delay Function (CADF) $D(\xi)$ that uniquely defines $P_D(D)$ and maps it onto a zero-mean, normalized ($\sigma = 1$) Gaussian parameter $\xi$ (Figure 2-2). The corresponding Cell/Arc Slew Function (CASF) $S(\xi)$ uniquely defines the output slew at each value of the cell delay. This unique mapping makes the delay/slew a function of the Gaussian parameter $\xi$ only.

The functions $D(\xi)$ and $S(\xi)$ are the outputs of cell characterization. In cell characterization, we typically characterize $D(\xi)$ and $S(\xi)$ over the range $-3 \leq \xi \leq 3$. For computational efficiency, it is necessary to characterize $D(\xi)$ and $S(\xi)$ as piecewise linear curves with a limited number of linear segments. The trade-off between accuracy and number of characterization points has been explored and presented in section 2.5.

### 2.1.2 Operating Point

Let us consider the simple case where $D(\xi)$ is a linear function of transistor random variables. At $V_{DD}$ near nominal and for sufficiently small $\sigma$, the CADF is essentially linear. Let $D$ be the stochastic delay and $x_i$ be a transistor random variable with a zero-mean Gaussian distribution. Because $D$ is linearly related to $x_i$, it can be
2.1 NLOPALV Theory for Cell Characterization

Figure 2-2: Gaussian mapping of non-Gaussian PDF through CADF

expressed as:

\[ D = \sum_{i=1}^{N} \frac{dD}{dx_i} x_i = \sum_{i=1}^{N} \frac{dD}{d\zeta_i} \zeta_i = \sum_{i=1}^{N} \alpha_i \zeta_i \] (2.1)

where we define the normalized transistor random variable as: \( \zeta_i = \frac{x_i}{\sigma_i} \) and \( \alpha_i \) is defined as: \( \alpha_i = \frac{dD}{d\zeta_i} \).

\( \alpha_i \) are computed from SPICE simulations using eq. 2.2.

\[ \alpha_i = \frac{D \left( \zeta_i + \frac{\Delta \zeta_i}{2} \right) - D \left( \zeta_i + \frac{\Delta \zeta_i}{2} \right)}{\Delta \zeta_i} \] (2.2)
Since $x_i$ and $\zeta_i$ are statistically independent, the variance of $D$ can be written as:

$$Var(D) = \sigma_D^2 = \sum_{i=1}^{N} \alpha_i^2$$  \hspace{1cm} (2.3)

The resulting stochastic delay is a Gaussian random variable with zero mean and standard deviation $\sigma_D$. In the linear case, the CADF is $D(\xi) = \sigma_D \xi$ and the CASF is $S(\xi) = \sigma_S \xi$. Each cell/arc is completely characterized by the values $\sigma_D$ and $\sigma_S$.

However, at low voltage, stochastic delay is highly non-linear in the transistor random variables $x_i$. As a result, the PDF for delay is highly non-Gaussian. This situation is illustrated in Fig. 2-2.

The goal of cell characterization is to compute the non-linear CADF and CASF for each cell/arc. The first step in cell characterization is to compute the sensitivity curves for each of the $N$ random variables in each of the cell/arcs. A sensitivity curve for a cell with respect to a transistor random variable $x_i$ is defined as the cell delay variation as a function of that random variable, when all the other random variables are set to their nominal values. Fig. 2-3 shows typical sensitivity curves for an inverter with respect to the different transistor random variables.

![Typical sensitivity curves for an Inverter](image)

Figure 2-3: Typical sensitivity curves for an Inverter
2.1 NLOPALV Theory for Cell Characterization

We now introduce the concept of operating point in \( \xi \)-space. There is a different operating point for each value of \( \xi \), and we refer to it as the \( \xi \)-sigma operating point. Recall that \( \xi \) is generally restricted to the range \(-3 \leq \xi \leq 3\). For each arc/cell, characterization requires the evaluation of \( D(\xi) \) at a selected number of \( \xi \) values. For each value of \( \xi \), there is an operating point in \( \xi \)-space, which is the point at which the joint PDF of the \( \zeta_i \) is maximum, and which satisfies the condition, \( D = D_{\xi_0} \).

The fundamental assumption of this analysis is that, although \( D(\zeta_1, \zeta_2, \ldots, \zeta_N) \) is a non-linear function, it can be approximately linearized about any point in \( \xi \)-space, and, in particular, it can be linearized about the operating point. We define \( \zeta_{i_0}^{op} \) as the \( \xi \)-sigma operating point and \( \delta \zeta_i = \zeta_i - \zeta_{i_0}^{op} \) as the incremental variation in \( \zeta_i \) about this operating point. Then we can write:

\[
D(\xi_1, \xi_2, \ldots, \xi_N) = D_{\xi_0} + \sum_{i=1}^{N} \left( \frac{dD}{d\xi_i} \right) \delta \xi_i = D_{\xi_0} + \sum_{i=1}^{N} \alpha_{i_0}^{op} \delta \xi_i \tag{2.4}
\]

Moreover, since delay can be approximated as a linear function of the transistor random variables in the vicinity of the operating point, the PDF of delay is approximately the convolution of the PDFs of the individual \( \zeta_i \). The integrand of the convolution integral peaks at the operating point and falls off sharply in all directions. As a result, in the region of \( \zeta_i \) space that makes the largest contribution to the integral, the linear approximation is valid. Fig. 2-4 illustrates this idea.

We now make use of the Linear-Gaussian theory, which will be explained in more detail in section 3.2.1, which is valid when the function of random variables is linear and the PDFs of the random variables are Gaussian. This theory shows that the \( \xi \)-sigma operating point can be determined as the point of tangency of the hyper-plane \( \sum_{i=1}^{N} \alpha_i^{op} \delta \xi_i = 0 \) with the hyper-sphere \( \sum_{i=1}^{N} \zeta_i^{op} = \xi^2 \). The operating point is evaluated as:

\[
\zeta_{i_0}^{op} = \frac{\xi \alpha_{i_0}^{op}}{\sqrt{\sum_{j=0}^{N} (\alpha_{j_0}^{op})^2}} \tag{2.5}
\]

The self-consistent solution to eq. 2.5 gives the \( \xi \)-sigma operating point.
For each value of $\xi$, once the operating point is determined, the delay $D(\xi)$ and slew $S(\xi)$ are calculated by a single SPICE run at the operating point. The following section describes the implementation of this approach.

2.2 Cell Characterization Flow

Based on the theory described in the previous section, the implementation of NLOPALV approach for cell characterization requires iterative computation of eq. 2.5. This is done in an automated manner using scripts written in PERL and C programming language.

The basic inputs for this process are:

1. RC extracted SPICE netlist for the cell

2. Mismatch file describing global/local mismatch parameters for all the transistors
in the cell

3. SPICE testbench for delay/slew measurement

Both the mismatch file and SPICE testbench are generated by the PREL script. The script then automatically calls the C program to iteratively compute the $\xi$-sigma operating point for the cell. Once the operating point is computed, the same program calls SPICE to compute the delay $D(\xi)$ and slew $S(\xi)$. The script preforms this operation for all the characterization points and outputs the CADF and CASF for each arc for each cell.

After the characterization is completed, it is essential to represent the characterization output in a standard library format so that it can be used by commercial CAD tools in the design flow. We use the Liberty library format (.lib) (used by commercial STA timers like Synopsys PrimeTime) to represent the characterization output. The PERL script collects characterization outputs for all the cells and converts them into the standard liberty library format.

The cell characterization flow is summarized in Fig. 2-5.

The .lib format represents the cell delay and output slew as a function of input slew and output load only. However, we also require the delay and output slew to be represented as a function of the characterization point. In order to achieve this, we create multiple instances of the same cell, each instance corresponding to one characterization point. Fig. 2-6 shows the format of the .lib generated.

As will be described in the following chapter, the STA timer can swap different instances of the cell to compute the timing path operating point.

2.3 Run Time Analysis

Implementation of the algorithm is optimized for characterization run-time minimization. The number of SPICE simulations required for the overall cell characterization
process for each arc can be given by:

\[ N_{spice} = [N_{rv}N_{tr} + N_{char} + 1]N_{iter} \]  

(2.6)

where, \( N_{rv} \) is the number of parameters of random variations in each transistor, \( N_{tr} \) is the number of transistors in the cell, \( N_{char} \) is the number of points at which characterization is done and \( N_{iter} \) is the number of iterations required to compute the operating point.

\( N_{char} \) determines the number of line segments used in the piecewise linear approximation of the CADF/CASF and thus affects the accuracy of the non-Gaussian delay/slew PDF. Section 2.5 explores the trade-off between \( N_{char} \) and accuracy of characteriza-
2.3 Run Time Analysis

The actual processing in SPICE has been further optimized to reduce the run time. This is done by reordering the SPICE simulations to avoid repeated computations and using SPICE features like .modify. The overall characterization run time can be given by:

\[ T_{spice} = [\alpha N_{rv} N_{tr} + \beta N_{char} + 1] N_{iter} \]  \hspace{1cm} (2.7)

where, \(\alpha = 0.28\) and \(\beta = 0.23\).

Significant run-time reduction has been achieved over the previous implementation [9]. Fig. 2-7 shows the variation of run-time with number of transistors in the cell and compares it with the implementation in [9].
2.4 Results

Fig. 2-8 shows typical PDF of the total cell delay, obtained from cell characterization for an Adder operating at $V_{DD}$ of 0.5V. SPICE based Monte-Carlo analysis with 10000 samples is used for comparison.

Fig. 2-9 describes the accuracy of the NLOPALV cell characterization approach with
2.4 Results

respect to SPICE based Monte-Carlo analysis for some of the basic. There is an overall good agreement between the results of NLOPALV cell characterization approach and that of Monte-Carlo. In this work, we characterized a standard cell library consisting of 130 different cells designed using commercial 28nm CMOS technology. The % error in stochastic delay at $V_{DD} = 0.5V$ is observed to be less than 5% for both input rise and fall arcs.

![Figure 2-9: Cell characterization accuracy: NLOPALV vs. Monte-Carlo](image)

The NLOPALV approach achieves very good accuracy compared to the SPICE based Monte-Carlo analysis in a computationally efficient manner. It combines the advantages of both Monte-Carlo analysis and the Gaussian SSTA. Table 2.1 shows a comparison of these approaches in terms of accuracy and the number of SPICE simulations required to characterize 3-sigma delay at $V_{DD} = 0.5V$.

Characterization of setup/hold time requires about 10 times more SPICE simulations compared to a combinational cell with same number of transistors, because setup/hold time can not be directly measured from a SPICE simulation. It requires an indirect characterization approach involving binary search or a similar algorithm where the change in CLK-to-Q delay as a function of setup/hold time is used to characterize the setup/hold time.
### Table 2.1: Comparison of NLOPALV, Monte-Carlo and Gaussian SSTA

<table>
<thead>
<tr>
<th></th>
<th>Monte-Carlo</th>
<th>NLOPALV</th>
<th>Gaussian Approx.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAND2</td>
<td>No. of SPICE Sims.</td>
<td>10,000</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>%Error</td>
<td>0% (Reference)</td>
<td>5%</td>
</tr>
<tr>
<td>Flip-Flop</td>
<td>No. of SPICE Sims.</td>
<td>100,000</td>
<td>3,000</td>
</tr>
<tr>
<td></td>
<td>% Error</td>
<td>0% (Reference)</td>
<td>5%</td>
</tr>
</tbody>
</table>

#### 2.5 Characterization Trade-off: Sigma Spacing vs. Accuracy

The cells are typically characterized over the range $-3 \leq \xi \leq 3$. For computational efficiency, it is necessary to characterize the PDF as a piecewise linear function with a finite number of characterization points. In this section, we explore the trade-off between the sigma spacing (spacing between two adjacent characterization points) and the accuracy of characterization.

We start with the sigma spacing of 0.25 to evaluate the inherent accuracy of the NLOPALV algorithm. The delay PDF obtained in this case is considered to be the reference. The PDFs obtained using higher values of sigma spacing are compared to this reference PDF to determine their relative accuracy. Fig. 2-10 shows the relative accuracy in terms of % error for some of the cells.

As expected, the characterization accuracy decreases as the sigma spacing is increased. However, it should be noted that the degradation in accuracy is cell specific and bigger cells tend to have less degradation in accuracy compared to smaller cells of similar kind.

However, from Fig. 2-10, it is clear that the sigma spacing of 0.5 can be used to reduce the computations (eq. 2.6) and run time (eq. 2.7) without significantly affecting the accuracy. For all the analysis presented in this paper, sigma spacing of 0.5 has been
2.6 Cell Delay PDF: Trends

A number of trends are observed in the delay PDFs for different cells depending on the cell size, supply voltage, drive strength, input slew, etc. This section describes some of these trends in the cell delay PDF and how they affect the cell characterization.

2.6.1 Dependance on $V_{DD}$

The delay variation with respect to the transistor random variables becomes more and more linear as $V_{DD}$ approaches the nominal voltage of operation and as described in section 2.1, linear delay results into a Gaussian delay PDF. Fig. 2-11 shows the variation of standard deviation (1-sigma) of the stochastic delay for a 2-input NAND gate, for $V_{DD}$ ranging from 0.5V to 1V.

It should be noted from Fig. 2-11 that the standard deviation of stochastic delay falls exponentially as the $V_{DD}$ is increased. The results in Fig. 2-11 are for a 2-input NAND gate, but this trend appears in all the cells we analyzed. This is because,
Figure 2-11: Variation of standard deviation (1-sigma) of the stochastic delay with $V_{DD}$

as the $V_{DD}$ increases, relative variation in the threshold voltage as a result of local
variations goes down, leading to less effect on the cell delay.

Fig. 2-12 shows the PDFs of stochastic delays normalized by the respective standard deviations for the same range of $V_{DD}$s.

Figure 2-12: Variations in the PDFs of stochastic delays normalized by the corresponding standard deviations with $V_{DD}$

It can be clearly seen from Fig. 2-12 that the delay PDF for $V_{DD} = 0.5V$ is very non-Gaussian whereas the PDF for $V_{DD} = 1V$ is almost perfectly Gaussian. This
validates the argument made in section 2.1 that for nominal $V_{DD}$, the delay varies linearly with local variations resulting in Gaussian delay PDF, however as $V_{DD}$ is decreased, the delay variation becomes more and more non-linear resulting in highly non-Gaussian delay PDFs.

2.6.2 Dependence on Cell Size

As the cell size (number of transistors in the cell) increases the cell delay PDF becomes more Gaussian and the sigma reduces, making the PDF narrower. This follows directly from the Central Limit Theorem. As the number of random variables increases, the resulting probability distribution tends to become more Gaussian. Fig 2-13 shows stochastic delay PDFs for two cells, a NAND gate and an ADDER, with different transistor counts.

![Figure 2-13: Variation of stochastic delay PDF with cell size](image)

2.6.3 Dependence on Drive Strength

As the drive strength of the cell increases, relative variation in the transistor random variables decreases, leading to a relative decrease in the stochastic delay compared to the nominal cell delay.
Higher drive strength transistors are normally implemented as multiple fingers in the layout. All the fingers need to be treated as individual transistors for local variations. As a result, a higher drive strength cell can be viewed as a cell with more number of transistors. This explains the observation that the delay PDF becomes more Gaussian as the drive strength increases. A comparison of stochastic delay PDFs for an inverter with different drive strengths are shown in Fig. 2-14.

![Figure 2-14: Variation of stochastic delay PDF with cell drive strength](image)

**2.7 Summary**

In this chapter we outlined the NLOPALV approach as it applies to the standard cell library characterization. This is the first step in the SSTA design methodology for low voltage operation.

- In section 2.1 we described the theory underlying the NLOPALV approach for cell characterization. We introduced two key ideas, the Gaussian mapping of non-Gaussian functions and the operating point, that form the basis of NLOPALV approach. The concept of operating point will also be instrumental in timing path analysis as we will see in the next chapter.
Cell characterization flow, as described in section 2.2, starts with the parasitic extracted SPICE netlist for the standard cell. A PERL script generates the necessary SPICE testbench and a mismatch file enlisting all the parameters of random variations for each transistor in the cell. The same PERL script then invokes a C-program to calculate the sensitivity curves and the operating point for the cell. Once the operating point is determined, the cell stochastic delay and output slew is computed by a single SPICE run at the operating point. The PERL script collects the output CADF and CASF for each cell and converts it into standard liberty library format (.lib) so that it can be used by commercial STA tools in the IC design flow.

A standard cell library implemented using a commercial 28nm CMOS technology is characterized. Results presented in section 2.4 not only validate the NLOPALV approach for cell characterization but also highlight the high computational efficiency and accuracy of the approach.

We concluded the discussion about cell characterization by taking a look at different trends shown by the cell delay PDFs in section 2.6. It is critical to take these trends into account while developing a design methodology. These trends have significant impact on the accuracy of the approach in predicting the statistical circuit performance, especially while operating at very low voltages.

In the next chapter, we will look into how the NLOPALV approach can be extended to timing path analysis.
Chapter 3

Timing Path Analysis

In the previous chapter we outlined the complete approach for standard cell characterization. In cell characterization, the transistor random variables $x_i$ are Gaussian. But the cell delay is a non-linear function of transistor random variables. As a result, the Linear-Gaussian theory (sec 3.2.1) can not be used. However, NLOPALV approach provides an acceptable approximation to the non-Gaussian PDF of cell delay. In this chapter, we will look into how the NLOPALV approach can be extended to perform statistical timing analysis on individual logic paths.

3.1 Introduction

It is instructive to compare the NLOPALV approach as it applies to cell characterization and timing path analysis. Table 3.1 presents a comparison of the similarities and differences between these two applications of NLOPALV.

In cell characterization, the transistor random variables, $x_i$ are Gaussian whereas in timing path analysis, the individual cell delays $D_i$ are non-Gaussian variables. The CADF for each cell is mapped to the Gaussian parameter $\xi$ in cell characterization, i.e. we determine the $\xi$-sigma delay for each cell-arc. The $\xi_i$ for each of the individual cells becomes the Gaussian variable for timing path analysis and we determine the
Table 3.1: NLOPALV: Cell Characterization vs. Timing Path Analysis

<table>
<thead>
<tr>
<th>Cell Characterization</th>
<th>Timing Path Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input random variables</td>
<td>Transistor random variables $x_i$</td>
</tr>
<tr>
<td>PDF of input random variables</td>
<td>Gaussian</td>
</tr>
<tr>
<td>Output random variables</td>
<td>Cell delay $D$</td>
</tr>
<tr>
<td>Function relating input variables to output variables</td>
<td>$D(x_1, x_2, \ldots x_N)$</td>
</tr>
<tr>
<td>Linearity of the function</td>
<td>Non-linear</td>
</tr>
<tr>
<td>Unit-Gaussian input variables</td>
<td>$\xi_i = \frac{x_i}{\sigma_i}$</td>
</tr>
<tr>
<td>Unit-Gaussian output variables</td>
<td>$\xi$</td>
</tr>
</tbody>
</table>

$f$-sigma delay for the timing path. Cell delay is a non-linear function of the random variables $x_i$. Timing path delay, as we will see later in this chapter, is a linear function of the individual cell delays if we ignore the effect of correlations coming due to the slew propagation. However, for accuracy of the approach, it is necessary to take into account the effect of these correlations, which makes the timing path delay a non-linear function.

We will now describe timing path analysis in more detail. Timing path (TP) analysis starts with pre-characterized logic cells. In this analysis we will assume that for every arc of every cell there exist PDFs for the stochastic delay and output slew, represented in the form of cell arc delay function (CADF) and cell arc slew function.
Computational efficiency of the NLOPALV approach to timing path analysis results from the fact that the entire PDF of the timing path is not usually required. In timing path analysis, we are usually interested in a specific \( f \)-sigma delay where \( f \) could be +3 for setup time and -3 for hold time.

The goal of timing path analysis is to determine the 3-sigma timing path delay (or in general the \( f \)-sigma timing path delay) without incurring the computational expense of computing the entire PDF of the timing path delay. As we saw in the previous chapter, in general, the PDFs of the logic cell delays are highly non-Gaussian. However, even in the non-Gaussian case, the concept of \( f \)-sigma delay has meaning. As shown in Fig. 3-1, the 1-sigma point is the 84.1345 percentile, the 3-sigma point is the 99.8650 percentile etc. However, in the non-Gaussian case, the sigma defined by percentile is unrelated to the standard deviation of the non-Gaussian variable.

### 3.2 NLOPALV Theory for Timing Path Analysis

In order to clearly explain the concept, let us start with the simple case of a timing path consisting of two logic cells. Each of the logic cells are characterized by the CADF \( D(\xi) \) and CASF \( S(\xi) \). Our objective is to determine the \( f \)-sigma delay for this timing path.

#### 3.2.1 Linear - Gaussian Theory

The timing path delay depends on the variations in both the cells in this timing path. Let us represent the timing path delay as: \( D^{TP}(D_1, D_2) \). For simplicity, let us assume that the cell stochastic delays in the timing path, \( D_i \) are statistically independent. (We will see later in this section that this simplifying assumption is not true, and we will address this complication.)
Figure 3-1: Defining $f$-sigma in the non-Gaussian context

For simplicity, let us first consider the case where individual delay PDFs $P_1(D)$ and $P_2(D)$ are Gaussian. The $f$-sigma timing path delay can then be represented as shown in Fig. 3-2.

Any combination of delays $D_1$ and $D_2$ on the line $D_{f,\sigma}^{TP} = D_1 + D_2$ give $f$-sigma delay for the sum. Whereas, if we combine the $f$-sigma delay for each individual cell, it leads to an overly pessimistic estimate for the timing path delay.

The timing path delay PDF, $P_{TP}(f)$ is the convolution of the individual delay PDFs $P_1(D)$ and $P_2(D)$, computed at every point on the curve $D_{TP}^{TP}(D_1, D_2) = D_{f,\sigma}^{TP}$.

$$P_{TP}(D_{f,\sigma}^{TP}) = \int_{-\infty}^{\infty} P_1(D)P_2(D_{f,\sigma}^{TP} - D)dD$$  \hspace{1cm} (3.1)
Because of the Gaussian assumption for the individual cell delay PDFs, the convolution integral can then be written as:

\[ P_{TP}(D_{TP}) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi} \sigma_1} \exp \left( -\frac{D^2}{2\sigma_1^2} \right) \frac{1}{\sqrt{2\pi} \sigma_2} \exp \left( -\frac{(D_{TP} - D)^2}{2\sigma_2^2} \right) dD \]  

(3.2)

\[ = k_1 \exp \left( \frac{-(D_{TP})^2}{2(\sigma_1^2 + \sigma_2^2)} \right) \int_{-\infty}^{\infty} \exp \left( -\frac{(\sigma_1^2 + \sigma_2^2)}{2(\sigma_1^2 + \sigma_2^2)} (D - \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} D_{TP})^2 \right) dD \]

(3.3)

\[ = k_2 \exp \left( \frac{-(D_{TP})^2}{2(\sigma_1^2 + \sigma_2^2)} \right) \]  

(3.4)

where, \( k_1 \) and \( k_2 \) are constants.

Eq. 3.4 shows that the PDF of the timing path delay, \( P_{TP}(D_{TP}) \) is Gaussian with
the standard deviation $\sigma_{TP} = \sqrt{\sigma_1^2 + \sigma_2^2}$. This is expected since the individual cell delay PDFs, $P_1(D)$ and $P_2(D)$, are Gaussian and the timing path delay is a linear function of the individual cell delays.

Another very important result can be derived from eq. 3.3. Let us examine the integrand of the convolution integral in eq. 3.3 more closely. It can be clearly seen that the integrand peaks at the point \{\(D_{\text{op}}^1, D_{\text{op}}^2\}\}, where:

$$D_{\text{op}}^1 = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} D_{TP}$$
and
$$D_{\text{op}}^2 = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} D_{TP}$$

(3.5)

and falls off with the standard deviation:

$$\sigma_{\text{integ}} = \frac{\sigma_1 \sigma_2}{\sqrt{\sigma_1^2 + \sigma_2^2}}$$

(3.6)

This means that even though every point on the line $D_{TP} = D_1 + D_2$ gives \(f\)-sigma delay for the timing path, some points on this line are more probable than others. The maxima of the joint PDF of $D_1$ and $D_2$, i.e. the maxima of the convolution integrand occurs at:

$$D_{\text{op}}^1 = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} D_{TP}$$
and
$$D_{\text{op}}^2 = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} D_{TP}$$

(3.7)

We call this point as the Operating Point.

In the normalized random variable space, $\xi_i = \frac{D_i}{\sigma_i}$ the operating point can be mapped from $D$-space to $\xi$-space as:

$$\xi_{\text{op}}^1 = \frac{D_{\text{op}}^1}{\sigma_1}$$
and
$$\xi_{\text{op}}^2 = \frac{D_{\text{op}}^2}{\sigma_2}$$

(3.8)

Also, we know from eq. 3.4, that the standard deviation of $D_{TP}$ is $\sigma_{TP} = \sqrt{\sigma_1^2 + \sigma_2^2}$. 
So, the \( f \)-sigma timing path delay can be transformed as:

\[
D_{T\sigma}^{TP} = f \sigma_{TP} = f \sqrt{\sigma_1^2 + \sigma_2^2} \tag{3.9}
\]

Combining eq. 3.7, eq. 3.8 and eq. 3.9, we get:

\[
\xi_1^{op} = \frac{\sigma_1}{\sigma_1^2 + \sigma_2^2} D_{T\sigma}^{TP} \tag{3.10}
\]

\[
= \frac{\sigma_1}{\sigma_1^2 + \sigma_2^2} f \sqrt{\sigma_1^2 + \sigma_2^2} \tag{3.11}
\]

\[
= \frac{f \sigma_1}{\sqrt{\sigma_1^2 + \sigma_2^2}} \tag{3.12}
\]

and

\[
\xi_2^{op} = \frac{\sigma_2}{\sigma_1^2 + \sigma_2^2} D_{T\sigma}^{TP} \tag{3.13}
\]

\[
= \frac{\sigma_2}{\sigma_1^2 + \sigma_2^2} f \sqrt{\sigma_1^2 + \sigma_2^2} \tag{3.14}
\]

\[
= \frac{f \sigma_2}{\sqrt{\sigma_1^2 + \sigma_2^2}} \tag{3.15}
\]

The \( f \)-sigma operating point in the \( \xi \)-space can then be given as:

\[
\xi_1^{op} = \frac{f \sigma_1}{\sqrt{\sigma_1^2 + \sigma_2^2}} \quad \text{and} \quad \xi_2^{op} = \frac{f \sigma_2}{\sqrt{\sigma_1^2 + \sigma_2^2}} \tag{3.16}
\]

From the above discussion, three key observations about the linear-Gaussian theory can be made:

1. The integrand of the convolution integral has a maxima at the operating point, given by eq. 3.16. The integrand falls-off sharply with the standard deviation given by eq. 3.6. Because of this, only a small number of points on the line \( D_{T\sigma}^{TP} = D_1 + D_2 \), that lie in the vicinity of the operating point, contribute significantly towards the convolution.

2. The operating point lies on the line \( D^{TP}(\xi_1, \xi_2) = D_{T\sigma}^{TP} \) as well as on the circle
\( \xi_1^2 + \xi_2^2 = f^2 \). In other words, the points that contribute significantly towards the convolution integral lie on a line that is tangent to the circle of radius \( f \) and passes through the operating point.

3. The \( f \)-sigma timing path delay is given by evaluating the timing path delay at the operating point:

\[
D_{f\sigma}^{TP} = D_{f\sigma}^{TP}(\xi_1^{op}, \xi_2^{op}) = f\sqrt{\sigma_1^2 + \sigma_2^2} \quad (3.17)
\]

### 3.2.2 Non-linear - Non-Gaussian Theory

Now, let us generalize this idea by considering the case where the individual cell delay PDFs are non-Gaussian and the timing path delay is a non-linear function of \( D_1 \) and \( D_2 \). In this case, \( D^{TP}(D_1, D_2) = D_{f\sigma}^{TP} \) could be a highly non-linear curve. Fig. 3-3 illustrates this case.

In this case, unlike eq. 3.4, the timing path delay PDF will not be Gaussian. However, the results that some points on the curve \( D^{TP}(D_1, D_2) = D_{f\sigma}^{TP} \) are more probable than others and that the integrand of the convolution integral has a sharp peak still hold. So, there exists a point on the curve \( D^{TP}(D_1, D_2) = D_{f\sigma}^{TP} \) where the joint probability of \( D_1 \) and \( D_2 \) is maximum i.e. the operating point exists.

Let us now transform the timing path delay PDF into the \( \xi \)-space as well, by a mapping through the CADF, as shown in Fig. 3-1.

This transformation is based on the idea that:

\[
\int_{-\infty}^{f} G(\xi)d\xi = \int_{-\infty}^{D_{f\sigma}} P(D)dD \quad (3.18)
\]

If \( P(D) \) is Gaussian with standard deviation \( \sigma \), \( D(\xi) = \sigma \xi \). If \( P(D) \) is non-Gaussian, \( \frac{\partial P}{\partial \xi} |_{\xi_0} \) is the effective sigma of a Gaussian PDF that is valid in the region around \( \xi_0 \).
After the transformation, the timing path delay curve, $D_{TP}^{TP}(\xi_1, \xi_2) = D_{f\sigma}^{TP}$, can be drawn in the $\xi$-space as shown in Fig. 3-4.

A major advantage of transforming the timing path delay PDF in $\xi$-space is that the lines of constant joint probability are circles given by:

$$\xi_1^2 + \xi_2^2 = f^2 \quad (3.19)$$

In the vicinity of the operating point, we can represent the timing path delay as a
Taylor series expansion, as given by eq. 3.20 and shown in Fig. 3-4.

\[
D^{TP}_{\xi_1, \xi_2} = D^{TP}_{\xi_\sigma} + \left( \frac{dD^{TP}}{d\xi_1} \right)_{op} \delta\xi_1 + \left( \frac{dD^{TP}}{d\xi_2} \right)_{op} \delta\xi_2 
\]

(3.20)

\[
= D^{TP}_{\xi_\sigma} + \left( \frac{\partial D^{TP}}{\partial D_1} \right)_{op} \left( \frac{\partial D_1}{\partial \xi_1} \right)_{op} \delta\xi_1 + \left( \frac{\partial D^{TP}}{\partial D_2} \right)_{op} \left( \frac{\partial D_2}{\partial \xi_2} \right)_{op} \delta\xi_2 
\]

(3.21)

\[
= D^{TP}_{\xi_\sigma} + \alpha_1^{op} \delta\xi_1 + \alpha_2^{op} \delta\xi_2 
\]

(3.22)

where, we define:

\[
\alpha_i^{op} = \left( \frac{dD^{TP}}{d\xi_i} \right)_{op} = \left( \frac{\partial D^{TP}}{\partial D_i} \right)_{op} \left( \frac{\partial D_i}{\partial \xi_i} \right)_{op} 
\]

(3.23)

and

\[
\delta\xi_i = \xi_i^{op} - \xi_i 
\]

(3.24)

The term \( \left( \frac{\partial D^{TP}}{\partial D_i} \right)_{op} \) represents linearization of the non-linear timing path delay function in the vicinity of the operating point and the term \( \left( \frac{\partial D_i}{\partial \xi_i} \right)_{op} \) represents effective sigma of the Gaussian PDF that approximates the non-Gaussian cell delay PDF in the vicinity of the operating point.
At this point in the analysis, the operating point is unknown. But we know that in the vicinity of the operating point, the timing path delay curve can be linearized and the cell delay PDFs can be considered to be Gaussian with the effective sigma. This allows us to make use of the linear-Gaussian theory to determine the $f$-sigma operating point as the point of tangency of the curve $D_TP(\xi_1, \xi_2) = D_{f\sigma}^T$ and the circle $\xi_1^2 + \xi_2^2 = f^2$, which is given as:

$$\xi_{i,op} = \frac{f \alpha_{i,op}}{\sqrt{(\alpha_{1,op})^2 + (\alpha_{2,op})^2}}$$  \hspace{1cm} (3.25)

The self-consistent solution to the transcendental equation 3.25 defines the operating point and the $f$-sigma timing path delay is obtained by evaluating $D_TP(\xi_1, \xi_2)$ at the operating point.

Let us now extend this theory from the two stage timing path to an $N$-stage timing path. We can generalize the concepts described above, from 2-dimensional space to $N$-dimensional space. So, the operating point for an $N$-stage timing path can be determined as the point of tangency of the hyper-plane $D_TP(\xi_1, \xi_2, \ldots, \xi_N) = D_{f\sigma}^T$ and the hyper-sphere $\sum_{i=1}^N \xi_i^2 = f^2$. Similar to eq. 3.26, the operating point for an $N$-stage timing path can be determined by finding the self-consistent solution to the equation:

$$\xi_{i,op} = \frac{f \alpha_{i,op}}{\sqrt{\sum_{j=1}^N (\alpha_{j,op})^2}}$$  \hspace{1cm} (3.26)

Notice that the timing path operating point in $\xi$-space, given by eq. 3.26, is very similar to the operating point we defined for cell characterization in $\zeta$-space, given by eq. 2.5. This is because in both cases, the resulting delay is a non-linear function of the individual parameters and in both cases the convolution integrand has a sharp peak, which allows the linearization of the delay curve about the point of maxima, i.e. the operating point. The only difference is that, unlike cell characterization, where the random variables are Gaussian, the random variables are non-Gaussian in timing path analysis. This difference is highlighted in the way $\alpha$ is defined for cell
characterization (eq. 2.2) and timing path analysis (eq. 3.23).

Once the operating point is determined, the $f$-sigma timing path delay can be obtained as a linear sum of the individual cell delays computed at the operating point.

$$D_{f\sigma}^{TP} = \sum_{i=1}^{N} D_i(\xi_i)$$  \hspace{1cm} (3.27)

3.3 Multi-Stage Timing Path Analysis

In the previous section, we developed the theory underlying the NLOPALV analysis. Now we describe how this theory applies to real timing paths consisting of multiple standard cells.

3.3.1 Correlation due to Slew Propagation

Recollect that in developing the theory we made a simplifying assumption that the individual cell stochastic delays are statistically independent. This assumption makes the timing path delay a linear function of the individual cell delays, as given by eq. 3.27.

However, for a given cell in the timing path, the delay and output slew are highly interdependent. Variation in transistor parameters affects the delay as well as output slew in a generally non-linear manner. In addition, the output slew of $i$-th cell is the input slew to the $(i+1)$-th cell, thus making the delay and output slew of a stage dependent on the variation of transistor parameters in all the previous stages. This not only necessitates combined consideration of delay and output slew for each stage, but also makes the stochastic delays of all the stages in the timing path correlated.

The effect of correlation can be taken into account by introducing an input slew correction. The input slew correction accounts for the fact that the stochastic delay of a given cell is correlated to the stochastic delay of the preceding cells. The stochastic
delay of a given cell, with correlation, is written as the sum of:

1. Stochastic delay resulting from transistor random variables within the cell. This is given by $D_i$ whose PDF is calculated during cell characterization with fixed input slews.

2. Stochastic delay which results from variations in the input slew. This stochastic delay is calculated in terms of $D_i$ for the preceding cells.

The result is that the correlation among individual stages is calculated explicitly and the timing path delay is written as a non-linear function of the statistically independent $D_i$.

The effect of correlation due to slew propagation can be incorporated in the timing path analysis by making two modifications in the NLOPALV theory:

1. The operating point, given by eq. 3.26, needs to be modified to take into account the effect of slew propagation.

2. The timing path delay, given by eq. 3.27 needs to be modified, which makes it a complex function of individual cell delays and output slews.

To further illustrate this idea, consider the case of a two-stage timing path. $D_1(\xi_{op}^1)$ is the stochastic delay of stage-1 and $D_2(\xi_{op}^1,\xi_{op}^2)$ is the stochastic delay of stage-2. The stochastic delay of stage-2 is determined not only by its own operating point, $\xi_2$, but also by the operating point of the first stage, $\xi_1$, because the input slew to stage 2 depends on the variations of transistor random variables in stage 1. The timing path delay can then be expressed as:

$$D_{f\sigma}^{TP}(\xi_1,\xi_2) = D_1(\xi_{op}^1) + D_2(\xi_{op}^1,\xi_{op}^2)$$

(3.28)
Taylor series expansion of the timing path delay around the operating point gives:

\[ D_{T_f}^P = D_{f_1} + \left( \frac{dD_{T_f}^P}{d\xi_1} \right)_{op} \delta \xi_1 + \left( \frac{dD_{T_f}^P}{d\xi_2} \right)_{op} \delta \xi_2 \]

\[ = D_{f_1} + \left( \frac{\partial D_{T_f}^P}{\partial D_1} \right)_{op} \left( \frac{\partial D_1}{\partial \xi_1} \right)_{op} \delta \xi_1 + \left( \frac{\partial D_{T_f}^P}{\partial D_2} \right)_{op} \left( \frac{\partial D_2}{\partial \xi_1} \right)_{op} \delta \xi_2 
+ \left( \frac{\partial D_{T_f}^P}{\partial D_2} \right)_{op} \left( \frac{\partial D_2}{\partial \xi_2} \right)_{op} \delta \xi_2 \]

\[ = D_{f_1} + (\alpha_1^{op} + \eta_1^{op}) \delta \xi_1 + \alpha_2^{op} \delta \xi_2 \]

where, we define:

\[ \alpha_i^{op} = \left( \frac{\partial D_{T_f}^P}{\partial D_i} \right)_{op} \left( \frac{\partial D_i}{\partial \xi_i} \right)_{op} \]

and

\[ \eta_i^{op} = \left( \frac{\partial D_{T_f}^P}{\partial D_{i+1}} \right)_{op} \left( \frac{\partial D_{i+1}}{\partial \xi_i} \right)_{op} = \left( \frac{\partial D_{T_f}^P}{\partial D_i} \right)_{op} \left( \frac{\partial D_i}{\partial \xi_i} \right)_{op} \left( \frac{\partial S_i}{\partial \xi_i} \right)_{op} \]

So, the \( f \)-sigma operating point for the two-stage timing path needs to be modified as:

\[ \xi_1^{op} = \frac{f(\alpha_1^{op} + \eta_1^{op})}{\sqrt{(\alpha_1^{op} + \eta_1^{op})^2 + (\alpha_2^{op})^2}} \quad \text{and} \quad \xi_2^{op} = \frac{f\alpha_2^{op}}{\sqrt{(\alpha_1^{op} + \eta_1^{op})^2 + (\alpha_2^{op})^2}} \]

The timing path delay can then be determined as:

\[ D_{f_1}^P = D_1(\xi_1^{op}) + D_2(\xi_2^{op}) + \left( \frac{\partial D_2}{\partial S_1} \right)_{op} S_1(\xi_1^{op}) \]

This can be extended to an \( N \)-stage timing path by modifying the operating point as:

\[ \xi_i^{op} = \frac{f(\alpha_i^{op} + \eta_i^{op} + \lambda_i^{op})}{\sqrt{\sum_{i=1}^{N}(\alpha_i^{op} + \eta_i^{op} + \lambda_i^{op})^2}} \]
where, we define:

\[
\lambda_i^{op} = \left( \frac{\partial D_{TP}^{i}}{\partial D_{i+2}^{i+2}} \right)_{op} \left( \frac{\partial D_{i+2}^{i+2}}{\partial \xi_i^{op}} \right)_{op} = \left( \frac{\partial D_{TP}^{i}}{\partial D_{i+2}^{i+2}} \right)_{op} \left( \frac{\partial D_{i+2}^{i+2}}{\partial S_{i+1}^{i+1}} \right)_{op} \left( \frac{\partial S_{i+1}^{i+1}}{\partial S_i^{i}} \right)_{op} \left( \frac{\partial S_i^{i}}{\partial \xi_i^{op}} \right)_{op}
\]

(3.37)  
(3.38)

Notice that we only consider the effect of correlation for two previous stages. This is because, experiments show that the effect of correlation falls exponentially and beyond the previous two stages, this effect is hardly significant.

The timing path delay for the \( N \)-stage timing path can then be given as:

\[
D_{TP}^{n} = \sum_{i=1}^{N} D_i(\xi_i^{op}) + \sum_{i=1}^{N-1} \left( \frac{\partial D_{i+1}^{i+1}}{\partial S_i^{i}} \right)_{op} S_i(\xi_i^{op}) + \sum_{i=1}^{N-2} \left( \frac{\partial D_{i+2}^{i+2}}{\partial S_{i+1}^{i+1}} \right)_{op} \left( \frac{\partial S_{i+1}^{i+1}}{\partial S_i^{i}} \right)_{op} S_i(\xi_i^{op})
\]

(3.39)

**3.3.2 NLOPALV Algorithm for TP Analysis**

The NLOPALV approach leads us to an iterative algorithm for determining the operating point and thereon computing \( D_{TP}^{n} \) that can be summarized as follows:

1. Determine the nominal delays, \( D_{i}^{nom} \) for each stage in the timing path.

2. Make the initial estimate of the operating point as:

\[
\xi_i^{op} = \frac{f D_{i}^{nom}}{\sqrt{\sum_{i=1}^{N}(D_{i}^{nom})^2}}
\]

(3.40)

3. At the estimated operating point, calculate \( \alpha_i, \eta_i \) and \( \lambda_i \) for \( 1 \leq i \leq N \) as defined by eq. 3.32, eq. 3.33 and eq. 3.38 respectively.

4. Compute new estimate of the operating point using eq. 3.36.

5. Repeat steps 3 and 4 until the operating point converges to a constant value.

6. Determine the timing path delay as given by eq. 3.39.
3.3.3 Integration with the CAD Flow

In order to effectively use SSTA in the design process, it is important to integrate the NLOPALV algorithm with existing CAD flow. The NLOPALV algorithm presented above is integrated with a commercial STA Timer. Fig. 3-5 describes the process in the form of a flow-chart.

3.3.4 Results

To validate the NLOPALV approach developed here, we tested it on logic paths taken from a commercial Digital Signal Processor implemented in 28nm technology, operating at $V_{DD} = 0.5V$. We used the NLOPALV algorithm and the corresponding CAD flow to determine 3-sigma delay for the logic timing path and compared the results with those obtained from SPICE based Monte-Carlo analysis. Fig. 3-6 shows the comparison results for different timing paths.

The 3-sigma delay obtained from NLOPALV analysis is within 5% accuracy compared to 10,000 points SPICE based Monte-Carlo analysis. Theoretical analysis shows that the NLOPALV approach always underestimates the stochastic delay compared to Monte-Carlo analysis and this is validated by Fig. 3-6.

Fig. 3-7 shows comparison between the NLOPALV approach and Monte-Carlo for a typical timing path taken from the 28nm DSP, at $V_{DD} = 0.5V$. It shows excellent agreement between the NLOPALV approach and Monte Carlo and also illustrates the inadequacy of the Gaussian approximation at low voltage. It is informative to note that:

1. The PDF of the TP delay peaks at a point in time that is less than the nominal delay (zero-sigma delay)

2. The mean of the non-Gaussian PDF lies 1.6ns to the right of the nominal delay, whereas in the Gaussian approximation, the stochastic delay has zero mean.
3.3 Multi-Stage Timing Path Analysis

For each cell \( j = 1 \ldots N \) assume the nominal delay at \( \xi_j^{\text{OP}} = 0 \).

Determine initial estimate of f-sigma operating points: \( \xi_j^{\text{OP}} \)

If \( a_i \leq \xi_j^{\text{OP}} \leq a_{i+1} \)

Swap cell \( j \) by its instance corresponding to \( \xi_j^{\text{OP}} = a_i \)

Swap cell \( j \) by its instance corresponding to \( \xi_j^{\text{OP}} = a_{i+1} \)

Run PrimeTime

Run PrimeTime

Calculate

\[ D_{f\sigma} = \sum_j D_j(\xi_j^{\text{OP}}) \]

Calculate

\[ D_j(\xi_j^{\text{OP}}) = D_j(a_i) + a_j(\xi_j^{\text{OP}} - a_i) \]

Determine new estimate of f-sigma operating points \( \xi_j^{\text{OP}} \)

Operating points converged?

YES

NO

3. The 3-sigma stochastic delay is delay is 13.41ns, compared to a nominal delay of 9.17ns. This shows that the variation at \( V_{DD} = 0.5V \) can be much higher than the nominal delay itself.

4. The 3-sigma stochastic delay calculated using the Gaussian approximation is
Timing Path Analysis

Figure 3-6: Performance comparison for NLOPALV vs. Monte-Carlo

![Graph showing performance comparison for NLOPALV vs. Monte-Carlo.](image)

Figure 3-7: Delay PDF for a TP from the 28nm DSP at $V_{DD} = 0.5V$: The zero-sigma delay is the nominal delay. The Gaussian approximation is chosen such that the standard deviation for the Gaussian is same as the 1-sigma delay for NLOPALV:

8.53 ns compared to the actual 3-sigma stochastic delay of 13.41 ns. This shows that the Gaussian approximation is highly optimistic.

The NLOPALV analysis was performed on different paths of different lengths, taken from the 28nm Digital Signal Processor. Table 3.2 summarizes additional results for a few of these paths at $V_{DD} = 0.5V$. The 3-sigma delay (nominal + 3-sigma stochastic delay) is compared to other methods.
3.3 Multi-Stage Timing Path Analysis

delay) computed using NLOPALV shows excellent agreement with Monte-Carlo. This contrasts with the large errors that result when the delay is assumed to be Gaussian.

Table 3.2: Performance Comparison of NLOPALV vs. Monte-Carlo and Gaussian Approximation at $V_{DD} = 0.5V$

<table>
<thead>
<tr>
<th>TP</th>
<th>Nominal Delay (ns)</th>
<th>3-Sigma Stochastic Delay (ns)</th>
<th>NLOPALV</th>
<th>Monte-Carlo</th>
<th>% Error</th>
<th>Gaussian Approx.</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.81</td>
<td>4.38</td>
<td></td>
<td>4.61</td>
<td>4.98%</td>
<td>2.38</td>
<td>45.6%</td>
</tr>
<tr>
<td>2</td>
<td>4.07</td>
<td>4.31</td>
<td></td>
<td>4.59</td>
<td>6.10%</td>
<td>2.68</td>
<td>37.8%</td>
</tr>
<tr>
<td>3</td>
<td>6.88</td>
<td>7.89</td>
<td></td>
<td>8.33</td>
<td>5.28%</td>
<td>4.19</td>
<td>45.5%</td>
</tr>
<tr>
<td>4</td>
<td>9.76</td>
<td>13.15</td>
<td></td>
<td>13.86</td>
<td>5.12%</td>
<td>6.22</td>
<td>52.7%</td>
</tr>
<tr>
<td>5</td>
<td>14.33</td>
<td>16.30</td>
<td></td>
<td>17.34</td>
<td>5.99%</td>
<td>7.88</td>
<td>50.7%</td>
</tr>
<tr>
<td>6</td>
<td>22.59</td>
<td>25.95</td>
<td></td>
<td>27.55</td>
<td>5.80%</td>
<td>13.27</td>
<td>48.9%</td>
</tr>
<tr>
<td>7</td>
<td>27.12</td>
<td>30.16</td>
<td></td>
<td>32.18</td>
<td>6.21%</td>
<td>14.80</td>
<td>50.9%</td>
</tr>
<tr>
<td>8</td>
<td>32.71</td>
<td>29.85</td>
<td></td>
<td>31.75</td>
<td>5.98%</td>
<td>14.00</td>
<td>53.1%</td>
</tr>
</tbody>
</table>

3.3.5 Trade-off: Number of Iterations vs. Accuracy

In section 2.5, we looked at the trade-off between number of characterization points and accuracy of characterization for standard cells. We saw that the number of characterization points can be reduced to half with less than 2% change in the accuracy. Now, let us take a look at a trade-off for the timing path analysis: number of iterations vs. accuracy.

As we discussed in section 3.3.2, the operating point needs to be determined in an iterative manner. The iterations are carried out till the operating points settles. In
general, this takes about 6-7 iterations. Fig. 3-8 shows how the 3-sigma operating point for a 5-stage timing path from the 28nm DSP, varies across iterations at 0.5V $V_{DD}$.

![Graph showing variation of operating point across iterations.](image)

Figure 3-8: Variation of Operating Point for a 5-stage TP across iterations at $V_{DD} = 0.5V$

Notice that it takes six iterations to settle to the final operating point, but after three iterations, the estimated operating point gets close to the final value. In order to determine the impact of the change in operating point on accuracy of the timing path delay, we calculated the 3-sigma timing path delay at each of the estimated operating points for this path at 0.5V. Fig. 3-9 shows the % error compared to 3-sigma delay from the Monte-Carlo analysis, at each of these points.

From Fig. 3-8 and Fig. 3-9, we can see that it is possible to stop at an earlier point in the iterations at the cost of some degradation in the accuracy. However, it should be noted that the operating point as well as the error in the TP delay do not vary monotonically. This puts some limitations on how soon the iterations can be stopped.
3.4 Timing Path Setup and Hold Analysis

After verifying the approach on single logic paths, we now extend it to perform setup and hold time analysis on timing paths including clock paths. Fig. 3-10 shows a typical timing path for setup/hold analysis.

Figure 3-10: Typical Timing Path for Setup/Hold Analysis
Consider the hold time constraint:

\[ D_1 > D_2 + T_{\text{hold}} \] (3.41)

In presence of statistical local variations, we need to make sure that:

\[(D_1 - D_2)_{-\sigma} - T_{\text{hold}}^{f\sigma} > 0 \] (3.42)

Let, \( D_1 = D_{\text{comm}} + D'_1 \) and \( D_2 = D_{\text{comm}} + D'_2 \) (3.43)

The hold constraint can then be defined as:

\[(D'_1 + D_{\text{comm}} - D'_2 - D_{\text{comm}})_{-\sigma} - T_{\text{hold}}^{f\sigma} > 0 \] (3.44)

which reduces to:

\[(D'_1 - D'_2)_{-\sigma} - T_{\text{hold}}^{f\sigma} > 0 \] (3.45)

Similarly, the setup constraint can be defined as:

\[(D'_1 - D'_2)_{\sigma} + T_{\text{setup}}^{f\sigma} < T_{\text{CLK}} \] (3.46)

Where, \( T_{\text{CLK}} \) is the clock period for the design. The PDFs for setup/hold time for the registers are obtained from cell characterization. Notice that in eq. 3.45 as well as eq. 3.46, the common path delay does not affect the setup or hold constraint. However the output slew at the output of the common path affects both \( D'_1 \) and \( D'_2 \) such that both \( D'_1 \) and \( D'_2 \) increase with increasing slew, but \( (D'_1 - D'_2) \) could be an increasing or decreasing function of the slew. So, to determine \( (D'_1 - D'_2)_{-\sigma} \) or \( (D'_1 - D'_2)_{\sigma} \) we need to consider both \( \pm f\sigma \) slew at the output of the common path. Fig. 3-11 describes the analysis flow.

Considering \( \pm f\sigma \) slew for the common path introduces some pessimism in the calculation of \( (D'_1 - D'_2)_{-\sigma} \) and \( (D'_1 - D'_2)_{\sigma} \). But the pessimism is not significant since
3.4 Timing Path Setup and Hold Analysis

Consider logic path with clock tree

Determine the common clock path

Perform NLOPALV on common path to get 3-sigma delay/slew

Set the common path cells to their 3 sigma operating points

Determine +/-3 sigma delay for (D1 - D2) using NLOPALV, say, S1/H1

Perform NLOPALV on common path to get -3 sigma delay/slew

Set the common path cells to their -3 sigma operating points

Determine +/-3 sigma delay for (D1 - D2) using NLOPALV, say, S2/H2

Is S1 > S2 or H1 < H2?

\[
(D_1 - D_2)_{3\sigma} = S1 \\
(D_1 - D_2)_{3\sigma} = H1
\]

\[
(D_1 - D_2)_{-3\sigma} = S2 \\
(D_1 - D_2)_{-3\sigma} = H2
\]

Figure 3-11: Setup/Hold Analysis Flow

the common path delay does not affect the computation.

This approach is verified by considering timing paths along with the corresponding clock paths from the 28nm Digital Signal Processor at \( V_{DD} = 0.5V \). The 3\( \sigma \) setup/hold slack is computed using the SSTA approach described above and the results are compared with those obtained from SPICE based Monte-Carlo simulations. The results for one of the timing paths are shown in Table 3.3.

Similar analysis is performed on some more paths taken from the same DSP and the
Table 3.3: Setup/Hold Analysis: NLOPALV vs. Monte-Carlo at $V_{DD} = 0.5V$

<table>
<thead>
<tr>
<th>Delay Type</th>
<th>NLOPALV</th>
<th>Monte-Carlo</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$(D_1 - D_2)$ at 3 sigma slew</td>
<td>$(D_1 - D_2)$ at -3 sigma slew</td>
<td>$(D_1 - D_2)$</td>
</tr>
<tr>
<td>Setup</td>
<td>10.91ns</td>
<td>9.21ns</td>
<td>10.05ns</td>
</tr>
<tr>
<td>Hold</td>
<td>7.25ns</td>
<td>5.28ns</td>
<td>6.49ns</td>
</tr>
</tbody>
</table>

results are summarized in Fig. 3-12 in the form of % error compared to Monte-Carlo analysis at $V_{DD} = 0.5V$. As seen from Fig. 3-12, the max % error for both setup and hold analysis is observed to be less than 8% compared to Monte-Carlo analysis.

3.5 Summary

In this chapter, we described the NLOPALV approach for Timing Path Analysis.

- In section 3.2, we described the NLOPALV theory as it applies to the timing path analysis. We started by describing the basic concepts using the Linear-
Gaussian Theory and derived the expression for $f$-sigma operating point for the timing path.

- We extended this theory in section 3.3 to real timing paths. In section 3.3.1, we outlined the effect of correlation, introduced by slew propagation between individual stages and described the modified expressions for operating point and timing path delay. We then outlined the NLOPALV algorithm for timing path analysis in section 3.3.2 and described the integration of this algorithm with the commercial CAD flow in sec 3.3.3.

- Results presented in sec 3.3.4 validate the NLOPALV approach for timing path analysis. Comparison with SPICE based Monte-Carlo analysis for timing paths taken from a Digital Signal Processor implemented using commercial 28nm CMOS technology and operating at $V_{DD} = 0.5V$, shows excellent accuracy of the approach. The results also highlight the inadequacy of the Gaussian approximation in accurately predicting the $f$-sigma stochastic delays for low-voltage operation.

- In section 3.4, we described how the NLOPALV approach works for setup and hold time analysis. The approach is validated by comparison with Monte-Carlo analysis at $V_{DD} = 0.5V$.

In the next chapter, we will describe reducing the number of critical paths in a design, that need to be analyzed: another key step towards timing closure of entire ICs.
Chapter 4

Timing Closure Flow for ICs

So far, we have looked at two key steps in the SSTA design methodology: standard cell library characterization and individual timing path analysis. In this chapter, we will describe another key step in this SSTA design methodology, i.e. reducing the number of timing paths that need to be analyzed. This step will allow us to use NLOPALS analysis to perform timing closure for the entire IC.

4.1 Reducing Number of Critical Paths

Any practical IC has a huge number of timing paths. With all the optimizations and algorithmic implementations described in the previous chapter, we can complete the timing path analysis and obtain the $f$-sigma timing path delay for a path in about 10 sec. However, even with the most optimized algorithm and implementation, it is impossible to analyze each timing path in the design in practically acceptable run time. For example, a typical digital signal processor has more than 10 million timing paths. At the rate of 10 seconds per timing path, it would take more than three years to analyze the entire design.

This necessitates a reduction in the number of timing paths that need to be analyzed, for the technique to be useful for analyzing complete designs and for timing closure
of the ICs. Fortunately, a large number of timing paths in any design have negligible probability of becoming critical despite all the variations. These paths need not be considered for analysis while determining the \( f \)-sigma delay for the chip. The challenge is, how to identify these paths and eliminate them in a time efficient manner so that a detailed analysis can be performed on the potentially critical paths in practically acceptable run time. The concepts of NLOPALV prove to be of great significance in identification and elimination of these non-critical paths in a time-efficient manner.

We now describe a two-step approach using the basic concepts of NLOPALV for Timing Closure Flow in ICs.

### 4.1.1 Step - 1: Elimination by Overly Pessimistic Estimate

Consider a typical timing path, as shown in Fig. 4-1.

![Figure 4-1: Typical Timing Path](image)

We know from our discussion so far, that the \( f \)-sigma timing path delay is a combination of individual cell delays computed at the operating point. We also know that if we take the \( f \)-sigma delay for every cell in the timing path, it leads to an overly pessimistic delay estimate for the timing path, as shown in Fig. 4-2.

This also means that if we consider the \( f \)-sigma delay for each cell in the timing path
4.1 Reducing Number of Critical Paths

Overly Pessimistic

Figure 4-2: $f$-sigma TP delay as a combination of individual cell delays at the operating point. Considering $f$-sigma delay for each cell leads to overly pessimistic TP delay estimate

and determine the pessimistic estimate, the timing paths that pass the setup/hold timing requirement in this overly pessimistic situation are certain to satisfy the actual $f$-sigma setup/hold timing requirement. Such timing paths can be considered non-critical for $f$-sigma delay estimation of the IC and can be eliminated from further analysis.

This is the first step in the Timing Closure Flow. We use the cell instance corresponding to $f$-sigma for each cell, from the standard cell library that we characterized. Then we perform a single STA analysis using a commercial STA timer on the entire design. Once this simulation is complete, we assign the paths that satisfied the setup/hold constraint as non-critical and eliminate them from further analysis. The paths that failed the setup/hold constraint are assigned as potentially critical and considered for
Step-2 analysis. Fig. 4-3 outlines this step as a flow-chart.

Consider all Start-End pairs in the design with clock trees

Swap all the cells in the design with their instances at -3 sigma

Run PrimeTime

Determine Data Arrival/Required Time for each Start-End pair

Swap all the cells in the design with their instances at 3 sigma

Run PrimeTime

Determine Data Required/Arrival Time for each Start-End pair

Determine the Hold/Setup Slack

Slack < 0 ?

YES

Potentially critical Start-End pair for hold/setup violation

NO

Start-End pair safe from hold/setup violation

Figure 4-3: Timing Closure Flow: Step -1

This step is hugely significant in overall critical part reduction since a large number of timing paths fall in the non-critical category and can be eliminated. This step is validated by performing the above analysis on a module from the digital signal processor. Table 4.1 shows the results of this analysis at $V_{DD} = 0.5V$.

As the results show, we are able to eliminate more than 90% of the timing paths from the design using the overly pessimistic delay estimate and this analysis reduces the
4.1 Reducing Number of Critical Paths

Table 4.1: Timing Closure Flow: Step - 1 Analysis Results at $V_{DD} = 0.5V$

<table>
<thead>
<tr>
<th>Setup Constraint</th>
<th>Number of Start-End Pairs</th>
<th>Before Step-1 Analysis</th>
<th>After Step-1 Analysis</th>
<th>% Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7458</td>
<td>104</td>
<td>98%</td>
<td></td>
</tr>
<tr>
<td>Run Time</td>
<td>5 days</td>
<td>10 mins + Run time for further analysis</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Hold Constraint</td>
<td>Number of Start-End Pairs</td>
<td>7458</td>
<td>622</td>
<td>91%</td>
</tr>
<tr>
<td>Run Time</td>
<td>5 days</td>
<td>10 mins + Run time for further analysis</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

run time by a significant amount.

4.1.2 Step - 2: Elimination by Pessimism Reduction

In the second step, we only consider the paths that are considered potentially critical from the step-1 analysis. Recollect that the delay estimate in step-1 is overly pessimistic. We can further eliminate the timing paths by reducing the pessimism. We do this by estimating real $f$-sigma clock skew, but we still use the $f$-sigma cell instances for the logic paths. This is because there are several logic paths associated with each Start-End pair, as shown in Fig. 4-4.

The setup time constraint can be given as:

$$T_{\text{clk-skew}} + T_{\text{logic}} + T_{\text{setup}} < T_{ULK}$$ (4.1)
and the hold time constraint can be given as:

\[ T_{\text{clk-skew}} + T_{\text{logic}} > T_{\text{hold}} \]  

(4.2)

Determining real \( f \)-sigma clock skew is much more time efficient compared to analyzing all the potentially critical paths in detail, but it still significantly reduces the pessimism. At the same time, it ensures that the paths that satisfy setup/hold constraint in this analysis are certain to satisfy the constraints for real \( f \)-sigma delay estimate.

In this step, we first consider only the clock paths corresponding to each start-end pair and perform the NLOPALV timing path analysis on these clock paths to determine the real \( f \)-sigma clock skew for each start-end pair. Then we compute a pessimistic estimate of the logic path delay corresponding to each of these start-end pairs by using the \( f \)-sigma cell instances from the standard cell library for the cells in the logic paths. This is done by a single run of the STA timer on the entire design. Then we combine the real \( f \)-sigma clock skew with the pessimistic logic path delay estimate and determine the if a given timing path satisfies the setup/hold constraints given by eq. 4.1 and eq. 4.2 respectively. Timing paths that satisfy these constraints can be considered non-critical for \( f \)-sigma delay estimate for the IC and those that fail the setup/hold constraints are considered critical. The critical paths are then analyzed.
in detail using the NLOPALV timing path analysis approach. Fig. 4-5 outlines this step as a flow-chart.

The potentially critical paths obtained from step-1, for the module from the DSP, are analyzed using step-2. Table 4.2 combines the results of step-1 analysis and step-2 analysis and presents the overall improvement achieved by reducing the number of critical paths in the design at $V_{DD} = 0.5V$.

\[3\text{The additional 50 mins of run time for setup constraint and 4 hours for the hold constraint is the time required to analyze the remaining critical timing paths using the detailed NLOPALV analysis after Step-2 is completed.}\]
Table 4.2: Timing Closure Flow: Complete Analysis Results at $V_{DD} = 0.5V$

<table>
<thead>
<tr>
<th></th>
<th>Before Timing Closure Flow</th>
<th>After Step-1 Analysis</th>
<th>After Step-2 Analysis</th>
<th>Overall Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Setup Constraint</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Start-End Pairs</td>
<td>7458</td>
<td>104</td>
<td>51</td>
<td>99.3%</td>
</tr>
<tr>
<td>Run Time</td>
<td>5 days</td>
<td>2 hours</td>
<td>40 mins + 50 mins³</td>
<td>98.75%</td>
</tr>
<tr>
<td><strong>Hold Constraint</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Start-End Pairs</td>
<td>7458</td>
<td>622</td>
<td>246</td>
<td>96.7%</td>
</tr>
<tr>
<td>Run Time</td>
<td>5 days</td>
<td>10 hours</td>
<td>2 hours + 4 hours³</td>
<td>95%</td>
</tr>
</tbody>
</table>

The results show the significance of the reducing the number of timing paths that need to be analyzed in the design and its impact of the overall run time. This technique plays a key role in enabling the use of NLOPALV approach for timing closure of entire ICs in practically acceptable run time.

4.2 Timing Closure for ICs

With the discussion of timing closure flow, we have completed all the steps in the SSTA design methodology for local variations at low voltage operation. Let us now review the overall flow and see how it is used for timing closure of an entire IC.

We begin with the SPICE netlist for the entire design and extract all the start-end pairs in the design. Then we analyze all these start-end pairs for reducing the number of timing paths. After completing both the steps of timing closure flow, we are left with a small number of start-end pairs that are critical for setup/hold time constraint. We then extract all the logic paths associated with these critical start-end pairs and
4.3 Summary

perform the detailed NLOPALV timing path analysis on each of these paths. The detailed NLOPALV timing path analysis determines the timing paths that actually fail the $f$-sigma setup/hold requirement for the IC. If there are any such failing paths in the design, modifications need to be made and the failing paths need to be re-analyzed to ensure that the design meets the $f$-sigma setup/hold requirement. The overall flow is summarized in Fig. 4-6.

4.3 Summary

In this chapter we outlined the concluding steps in the SSTA design methodology. A two-step timing closure flow is described in section 4.1. This two-step approach makes use of the basic concepts of NLOPALV. It significantly reduces the number of timing paths that need to be analyzed in detail using the NLOPALV setup/hold analysis.

As the results show, the timing closure flow eliminates more than 95% of the timing paths at $V_{DD} = 0.5V$ and reduces the overall analysis run time for the entire design to practically acceptable level. In section 4.2, we outlined the overall flow for timing closure of ICs.

With the Timing Closure Flow, we have now covered all the stages in the SSTA design methodology. We started with the standard cell library characterization in chapter 2. Then we described how the NLOPALV approach is used for performing timing path analysis in chapter 3. Finally we introduced the two-step timing closure flow in this chapter, which completes the design flow and enables us to analyzed entire ICs using this SSTA design methodology.

In the next chapter, we will look at the design of a shared, reconfigurable transform engine for H.264/AVC and VC-1 video coding standards. We will describe how the SSTA design methodology for low voltage is used in the chip design flow, to perform timing closure for this chip.
Timing Closure Flow for ICs

SPICE Netlist for the entire design

Extract all the Start-End pairs

Perform Critical Path Reduction to determine the Critical Start-End pairs for Setup/Hold analysis

Extract all the Logic paths corresponding to each critical Start-End pair

Perform detailed NLOPALV Setup/Hold analysis on all the critical timing paths

Determine the paths that fail the $\sigma$ Setup/Hold requirement

Modify the failing paths

Re-run NLOPALV timing path analysis on the modified timing paths

Failing paths for $\sigma$ Setup/Hold requirement?

Timing Closure Complete!

Figure 4-6: SSTA Design Methodology for Low Voltage: Timing Closure Flow for an IC
Chapter 5

Case Study: Reconfigurable Transform for Video Coding

Multimedia applications, such as video playback, are becoming increasingly pervasive on battery-operated handheld devices such as camera phones, digital still cameras, personal media players, etc. While exploring power reduction techniques at various design stages (algorithms, architectures and circuits) of the video CODEC, an effective method of reducing power involves aggressive voltage scaling. Transistor mismatches are becoming more and more prominent in deep-sub-micron processes requiring larger design margins for high yield. Moreover, the effect of mismatches are exacerbated at lower voltages making low-voltage designs very challenging to implement due to functionality problems in SRAMs and due to timing violations in critical paths. This is one of the many applications where the SSTA design methodology will be extremely useful for performing timing closure.

So far, we have used the timing paths taken from the DSP designed using commercial 28nm CMOS technology, operating at $V_{DD} = 0.5V$ to verify different stages of the SSTA design methodology for low voltage. Let us now consider a case study: a shared, reconfigurable transform engine for multi-standard video codecs. We will describe how such a shared, reconfigurable transform module can be designed and
demonstrate how the SSTA design methodology can be used in the chip design flow to achieve timing closure for the transform chip at $V_{DD} = 0.5V$.

5.1 Transform Coding

Transform and Quantization are some of the most basic operations in any video codec. High coding efficiency often comes at a cost of increased complexity in the Transform and Quantization modules. H.264/AVC [25] and VC-1 [12] are the most recent mainstream video coding standards that employ different kinds of Transform and Quantization schemes to achieve coding efficiency.

The H.264/AVC and VC-1 video coding standards have many innovations when compared to previous video coding standards, such as integer transforms. The transforms employ only integer arithmetic without multiplications, with coefficients and scaling factors that allow for 16-bit arithmetic computation on first-level transforms. These changes lead to a significant complexity reduction.

5.1.1 H.264/AVC Transform: Key Features

The key features of the H.264/AVC transform are variable size transform, hierarchical transform, 16-bit arithmetic and exact inverse transform.

- H.264/AVC primarily uses 4x4 transform, as opposed to previous standards that mainly use 8x8 transform. Baseline profile and Main profile only support 4x4 transform, however High profile also supports 8x8 transform. A variable size transform allows the standard to more effectively exploit the spatial correlation in the video frames. We implement both 4x4 and 8x8 size transform in order to support all the H.264/AVC profiles.

- While in most cases, using the small 4x4 transform block size is perceptually beneficial, there are some signals that contain sufficient correlation to call
for some method of using a representation with longer basis functions. The H.264/AVC standard enables this by using a hierarchical transform to extend the effective block size use for low-frequency chroma information to an 8x8 array and by allowing the encoder to select a special coding type for intra coding, enabling extension of the length of the luma transform for low-frequency information to a 16x16 block size.

- All prior standards have effectively required encoders and decoders to use more complex processing for transform computation. While previous transforms have generally required 32-bit processing, the H.264/AVC design requires only 16-bit arithmetic.

- In previous standards, residual decoding contains the possibility of drift (mismatch between the decoded data in the encoder and decoder). The drift arises from the fact that the inverse transform is not fully specified in integer arithmetic; rather it must satisfy statistical tests of accuracy compared with a floating point implementation of the inverse transform. In H.264/AVC, integer transform is used to avoid any drift.

### 5.1.2 VC-1 Transform: Key Features

The key features of VC-1 transform are integer transform, variable sized transform and 16-bit arithmetic.

- Like H.264/AVC, VC-1 also uses an integer transform, which is an approximation of the discrete cosine transform (DCT). The transform is designed for 16-bit arithmetic.

- VC-1 uses variable size transform. Intra MBs always use 8x8 transform. Motion compensated blocks in Inter MBs are transformed using one of the four transform types: 8x8, 8x4, 4x8, 4x4. In contrast, H.264/AVC uses a fixed transform size of 4x4 or 8x8.
Motion in video sequences can cause motion compensation to be more effective in some areas of a macro block while residual in the other areas is still large. In such cases, a variable size transform allows a more effective de-correlation of the residual signal.

The most commonly used transform in video and image coding applications is the Discrete Cosine Transform (DCT). DCT has excellent energy compaction property, which leads to good compression efficiency of the transform. However, the irrational numbers in the transform matrix make its exact implementation impossible, leading to a drift between forward and inverse transform coefficients.

H.264/AVC as well as VC-1 video coding standards use a variation of the DCT, known as Integer transform. In these transforms, the transform matrices are defined to have only integers. This makes exact inverse possible.

Let us now look at the definitions of these transforms for the two standards, H.264/AVC and VC-1.

### 5.1.3 H.264/AVC Integer Transform

The separable 2-D 8x8 forward transform for H.264/AVC can be written as:

\[
T^F_{8x8} = H_{8x8} \cdot X_{8x8} \cdot H^T_{8x8}
\]  

(5.1)

and the separable 2-D 8x8 inverse transform can be written as:

\[
T^I_{8x8} = (H^I_{8x8}) \cdot Y_{8x8} \cdot (H^I_{8x8})^T
\]  

(5.2)
5.1 Transform Coding

Where, the 1-D 8x8 forward integer transform for H.264/AVC is defined as:

\[
H_{8x8} = \begin{bmatrix}
8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\
8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\
10 & -3 & -12 & -6 & 6 & 12 & 3 & -10 \\
8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\
6 & -12 & 3 & 10 & -10 & -3 & 12 & -6 \\
4 & -8 & 8 & -4 & -4 & 8 & -8 & 4 \\
3 & -6 & 10 & -12 & 12 & -10 & 6 & -3
\end{bmatrix}
\]  

(5.3)

And the 1-D 8x8 inverse integer transform is defined as:

\[
H_{8x8}^I = \begin{bmatrix}
8 & 12 & 8 & 10 & 8 & 6 & 4 & 3 \\
8 & 10 & 4 & -3 & -8 & -12 & -8 & -6 \\
8 & 6 & -4 & -12 & -8 & 3 & 8 & 10 \\
8 & 3 & -8 & -6 & 8 & 10 & -4 & -12 \\
8 & -3 & -8 & 6 & 8 & -10 & -4 & 12 \\
8 & -6 & -4 & 12 & -8 & -3 & 8 & -10 \\
8 & -10 & 4 & 3 & -8 & 12 & -8 & 6 \\
8 & -12 & 8 & -10 & 8 & -6 & 4 & -3
\end{bmatrix}
\]  

(5.4)

Similarly, the separable 2-D 4x4 forward transform for H.264/AVC can be written as:

\[
T_{4x4}^{F} = H_{4x4} \cdot X_{4x4} \cdot H_{4x4}^{T}
\]  

(5.5)

and the separable 2-D 4x4 inverse transform can be written as:

\[
T_{4x4}^{I} = (H_{4x4}^{I}) \cdot Y_{4x4} \cdot (H_{4x4}^{I})^{T}
\]  

(5.6)
Where, the 1-D 4x4 forward transform for H.264/AVC is defined as:

\[
H_{4x4} = \begin{bmatrix}
1 & 1 & 1 & 1 \\
2 & 1 & -1 & -2 \\
1 & -1 & -1 & 1 \\
1 & -2 & 2 & -1 \\
\end{bmatrix}
\] (5.7)

And the 1-D 4x4 inverse integer transform is defined as:

\[
H_{4x4} = \begin{bmatrix}
1 & 1 & 1 & \frac{1}{2} \\
1 & \frac{1}{2} & -1 & -1 \\
1 & -\frac{1}{2} & -1 & 1 \\
1 & -1 & 1 & -\frac{1}{2} \\
\end{bmatrix}
\] (5.8)

### 5.1.4 VC-1 Integer Transform

Let us now look at the definitions of the different size transforms for VC-1.

The 8x8 inverse integer transform for VC-1 is given as:

\[
R_{8x8} = \frac{V_{8x8} \cdot X_{8x8} \cdot V_{8x8}^T}{1024}
\] (5.9)

The denominator is chosen to be the power of 2 closest to the squared norm of the basis functions (288, 289 and 292) of the 1D transformation.

In order to preserve one extra bit of precision, the 1-D transform operation is performed as:

\[
D_1 = \frac{X_{8x8} \cdot V_{8x8}^T}{16} \quad \text{and} \quad R_{8x8} = \frac{V_{8x8} \cdot D_1}{64}
\] (5.10)
The 1-D transform matrix is defined as:

\[
V_{8\times8} = \begin{bmatrix}
12 & 16 & 16 & 15 & 12 & 9 & 6 & 4 \\
12 & 15 & 6 & -4 & -12 & -16 & -16 & -9 \\
12 & 9 & -6 & -16 & -12 & 4 & 16 & 15 \\
12 & 4 & -16 & -9 & 12 & 15 & -6 & -16 \\
12 & -4 & -16 & 9 & 12 & -15 & -6 & 16 \\
12 & -9 & -6 & 16 & -12 & -4 & 16 & -15 \\
12 & -15 & 6 & 4 & -12 & 16 & -16 & 9 \\
12 & -16 & 16 & -15 & 12 & -9 & 6 & -4
\end{bmatrix}
\] (5.11)

and the 1-D 4x4 inverse transform matrix is defined as:

\[
V_{4\times4} = \begin{bmatrix}
17 & 22 & 17 & 10 \\
17 & 10 & -17 & -22 \\
17 & -10 & -17 & 22 \\
17 & -22 & 17 & -10
\end{bmatrix}
\] (5.12)

5.2 Reconfigurable Transform Design

In this section we will describe some key ideas in designing a shared, reconfigurable transform for H.264/AVC and VC-1 video coding standards.

5.2.1 Structural Similarity of the Transforms

The structure of matrices for both H.264/AVC and VC-1 video coding standards is identical, as defined in eq. 5.4 and eq. 5.11. The matrix structure for H.264/AVC and VC-1 can be represented as:
Where, for H.264/AVC:

\[ a = 8 \quad b = 12 \quad c = 10 \quad d = 6 \quad e = 3 \quad f = 8 \quad g = 4 \]

and for VC-1:

\[ a = 12 \quad b = 16 \quad c = 15 \quad d = 9 \quad e = 4 \quad f = 16 \quad g = 6 \]

The identical structure is very significant in designing the shared, reconfigurable transform, since it allows us to share a large number of computations between the two transforms.

### 5.2.2 Symmetry of the Transform Matrix

The transform matrices for both H.264/AVC and VC-1 video coding standards are highly symmetric. The symmetry is apparent from the structure shown in eq. 5.13.

This symmetry can be exploited to reduce the number of computations needed to perform the transform operation. A simple permutation of the transform matrix allows us to represent the transform matrix as:

\[ M = P \cdot \tilde{M} \]
Where,

\[
\begin{bmatrix}
a & f & a & g & 0 & 0 & 0 & 0 \\
a & g & -a & -f & 0 & 0 & 0 & 0 \\
a & -g & -a & f & 0 & 0 & 0 & 0 \\
a & -f & a & -g & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -e & d & -c & b \\
0 & 0 & 0 & 0 & -d & b & -e & -c \\
0 & 0 & 0 & 0 & -c & e & b & d \\
0 & 0 & 0 & 0 & -b & -c & -d & -e \\
\end{bmatrix}
\]

(5.15)

The 8x8 transform matrix is thus reduced to two 4x4 matrices. These 4x4 matrices can be further decomposed into smaller matrices to simplify computations.

An added advantage of splitting the big matrix into multiple smaller matrices is that the splitting can be done such that both H.264/AVC and VC-1 have same smaller matrices. The transform computations can be directly shared by this matrix splitting.

5.3 Low Voltage Transform

The shared, reconfigurable transform provides us computational savings over a direct implementation. We would also like the transform to consume as low power as possible.

5.3.1 Motivation for Low Voltage operation

In order to achieve a low-power transform, it is necessary to operate the design at a very low voltage. The delay of the system increases exponentially as \(V_{DD}\) is lowered and the energy decreases quadratically. Fig. 5-1 shows how the delay and energy vary as a function of \(V_{DD}\).
Low-voltage operation is a trade-off between throughput and power. This also necessitates careful design considerations to make sure that the throughput requirements are met. Furthermore, it makes accurate timing analysis extremely important to ensure than timing closure is achieved at the desired throughput level when the system is operated at $V_{DD} = 0.5V$.

### 5.3.2 Transform IC Implementation

The shared, reconfigurable transform for H.264/AVC and VC-1 video coding standards is designed for operation at $0.5V \, V_{DD}$ and implemented using commercial 45nm CMOS technology. Fig. 5-2 shows the layout of the chip.

### 5.4 Timing Analysis using SSTA Methodology

The transform design is synthesized using the 28nm standard cell library, which was characterized for SSTA using the NLOPALV cell characterization approach. The SSTA design methodology is used to perform timing closure for the transform IC at
5.4 Timing Analysis using SSTA Methodology

First step is to synthesize the design using the pre-characterized 28nm standard cell library. Once a gate level netlist is generated, we extract all the start-end pairs form the and the corresponding timing paths from the netlist. Then we use the 2-step timing closure flow, as described in section 4.1, to reduce the number of timing paths that need to be analyzed in detail using the NLOPALV setup/hold analysis. The results of this analysis at $V_{DD} = 0.5V$ are presented in Table 5.1.

As the results show, more than 99% of the timing paths in the design are eliminated during the timing closure flow. The detailed NLOPALV setup/hold analysis is performed on the timing paths that appear to be critical for the setup/hold constraint after the completion of the two-step timing closure flow. For one of the critical paths,

---

4Step-1 and Step-2 are the steps in the Timing Closure Flow, as described in sec. 4.1.1 and sec. 4.1.2 respectively

5The additional 1 hour of run time for setup constraint and 30 mins for the hold constraint is the time required to analyze the remaining critical timing paths using the detailed NLOPALV analysis after Step-2 is completed.
Table 5.1: Timing Closure Flow on the Transform IC at $V_{DD} = 0.5V$

<table>
<thead>
<tr>
<th>Setup Constraint</th>
<th>Number of Start-End Pairs</th>
<th>Before Timing Closure Flow</th>
<th>After Step-1 Analysis$^4$</th>
<th>After Step-2 Analysis$^4$</th>
<th>Overall Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>20186</td>
<td>397</td>
<td>65</td>
<td>99.67%</td>
</tr>
<tr>
<td>Run Time</td>
<td></td>
<td>15 days</td>
<td>7 hours</td>
<td>1 hour 30 min + 1 hour$^5$</td>
<td>99.30%</td>
</tr>
<tr>
<td>Hold Constraint</td>
<td></td>
<td>20186</td>
<td>438</td>
<td>16</td>
<td>99.9%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15 days</td>
<td>9 hours</td>
<td>1 hour 40 min + 30 min$^5$</td>
<td>99.4%</td>
</tr>
</tbody>
</table>

we computed the entire delay PDF at 0.5V $V_{DD}$. The PDF is shown in Fig. 5-3.

For rest of the critical paths, we performed the detailed NLOPALV analysis and determined the 3-sigma stochastic delay. Results of the detailed timing path analysis on some of these critical paths at $V_{DD} = 0.5V$ are shown in Table 5.2.

### 5.5 Summary

In this chapter, we described the design of a shared, reconfigurable transform for H.264/AVC and VC-1 video coding standards. We begin, in sec. 5.1, with the key ideas in the H.264/AVC and VC-1 transforms and take a look at the definitions of transform matrices for these two standards. The shared reconfigurable implementation for the transform module is achieved by mainly exploiting two ideas:
5.5 Summary

![Timing Path Delay PDF](image)

Figure 5-3: Timing Path Delay PDF for one of the critical paths from the Transform IC at $V_{DD} = 0.5V$

<table>
<thead>
<tr>
<th>TP</th>
<th>Nominal Delay (ns)</th>
<th>3-Sigma Stochastic Delay (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15.35</td>
<td>14.30</td>
</tr>
<tr>
<td>2</td>
<td>15.45</td>
<td>16.15</td>
</tr>
<tr>
<td>3</td>
<td>17.01</td>
<td>15.55</td>
</tr>
<tr>
<td>4</td>
<td>18.85</td>
<td>19.04</td>
</tr>
<tr>
<td>5</td>
<td>21.20</td>
<td>19.45</td>
</tr>
</tbody>
</table>

Table 5.2: Detailed Timing Path Analysis on Transform IC at $V_{DD} = 0.5V$

1. Structural similarities between the two transform matrices, as described in sec. 5.2.1

2. Symmetry of the transform matrices, that allows the larger matrix to be split into several smaller matrices to minimize computations and achieve reconfigurability, as described in sec. 5.2.2.
The SSTA design methodology is used in the chip design flow to achieve timing closure for the transform IC at 0.5V $V_{DD}$, as described in section 5.4. We synthesize the design using 28nm standard cell library and use the timing closure flow to eliminate more than 99% of the paths. Then we analyze the remaining critical paths using detailed NLOPALV approach to achieve timing closure for the IC.
Chapter 6

Conclusions

A computationally efficient SSTA design methodology for low voltage operation, based on the Non-Linear Operating Point Analysis for Local Variations (NLOPALV), has been developed. It has been implemented using commercial CAD tools and integrated into commercially used design flow.

6.1 Key Features

Major highlights of this NLOPALV based SSTA design methodology are:

- It is a computationally efficient approach for determining statistical circuit performance in the region where delay is a non-linear function of transistor random variables and the delay PDFs are non-Gaussian.

- The approach covers different aspects of SSTA, starting from Standard Cell Library Characterization to Timing Closure for ICs.

- The concept of operating point greatly simplifies computations without sacrificing accuracy even though the delays are non-linear and delay PDFs are non-Gaussian.

- Comparison with SPICE based Monte-Carlo analysis shows excellent accuracy
of the approach while significantly reducing the run time.

- No computationally expensive Monte-Carlo simulations required during timing closure.

- The approach is automated and integrated with the commercially used CAD flow.

6.2 Summary of Results

At each stage in the development of SSTA design methodology for low voltage, the accuracy of the approach was verified and different trends were analyzed.

- The cell characterization flow was verified by characterizing a standard cell library consisting of 130 cells implemented in commercial 28nm CMOS technology at 0.5V $V_{DD}$. As the results in section 2.4 show, the accuracy of NLOPALV approach is within 5% compared to 10,000 point SPICE based Monte-Carlo analysis.

- We looked at the trade-off of accuracy and spacing between the characterization points. The analysis, in section 2.5, shows that the accuracy is affected by less than 2% if we change the spacing from 0.25 to 0.5, at $V_{DD} = 0.5V$.

- In section 2.6, we looked at the trends shown by cell delay PDFs depending on factors such as $V_{DD}$, cell size and cell drive strength and analyzed their impact on the accuracy of characterization. We found a similar effect in all three cases, that the cell delay PDF tends to become more Gaussian and the sigma reduces as the we increase the $V_{DD}$, cell size or the drive strength. This leads to better accuracy in characterization as the non-linearities are reduced.

- For the timing path analysis, we used timing paths taken from a Digital Signal Processor, designed using commercial 28nm CMOS technology and operating at $V_{DD} = 0.5V$. The NLOPALV approach is used to compute the 3-sigma TP
delay which is compared with the 3-sigma delay obtained from 10,000 point Monte-Carlo analysis. For some of the paths, we computed entire delay PDFs using the NLOPALV approach and compared them with the PDFs predicted by Monte-Carlo analysis. Results shown in section 3.3.4 highlight that:

1. Predictions of NLOPALV approach match very well with those from Monte-Carlo analysis, at $V_{DD} = 0.5V$.

2. At 0.5V $V_{DD}$, the 3-sigma stochastic delay for the timing path can be comparable to, or in some case more than the nominal delay in a technology like 28nm CMOS.

3. Gaussian approximations cannot be justified at very low voltage operation in advance technologies like 28nm CMOS and can lead to errors in the order of 80-100% in predicting the 3-sigma delay at $V_{DD} = 0.5V$.

- The trade-off between accuracy and number of iterations for timing path analysis was analyzed in section 3.3.5. In general it takes about 6-7 iterations for the timing path operating point to settle at 0.5V $V_{DD}$. The analysis shows that it is possible to reduce the number of iterations to four without significant degradation in accuracy. The non-monotonic nature of variation in operating point and % error during iterations, puts a limit on how early the iterations can be terminated without suffering large degradation in accuracy.

- In the timing closure flow for ICs, we demonstrated the usefulness of the approach on a module from the same 28nm DSP. We analyzed this module using the 2-step timing closure flow, described in section 4.1. Results of this analysis show that more than 95% of the timing paths are eliminated and simulation run time is also reduced by about 95% for timing closure at $V_{DD} = 0.5V$.

- Finally, in chapter 5, we considered a case study. We described the design of a shared, reconfigurable transform for H.264/AVC and VC-1 video coding standards. The design was synthesized using the pre-characterized 28nm standard cell library and the SSTA design methodology was used to achieve timing clo-
sure for this IC at $V_{DD} = 0.5V$. The two-step timing closure flow succeeded in eliminating more than 99% of the timing paths. The remaining paths were analyzed using NLOPALV timing path analysis to achieve timing closure.

### 6.3 Limitations

There are three categories of process variations that are important in design of modern CMOS logic: Global random variations, Systematic or Predictable variations and Local variations. In this work, we focussed on the effects of local variations in logic timing at low voltage operation. Our approach does not take into account the effects of global variations. The NLOPALV analysis needs to be carried out in conjunction with standard ways of dealing with global variations, such as corner based analysis.

We also do not explicitly account for systematic variations, though if a post-layout parasitic extracted netlist is used for timing path analysis, the correlations due to spatial proximity effects are implicitly accounted for.

In this work, we do not consider device switching and false paths. This could lead to some unwanted pessimism in the 3-sigma delay estimate for the IC during timing closure.

### 6.4 Future Work

In the design flow, it is generally required to modify the timing paths multiple times in order to satisfy all the setup/hold constraints. In order to achieve this, critical path optimization becomes very significant. Since the NLOPALV approach determines the individual contribution of each stage towards the overall stochastic variation in the timing path, it is possible to modify the stages that contribute most stochastic variation, in order to minimize the overall variation in the timing path.
The knowledge of individual stage operating points could also be used for proper
device sizing to minimize power consumption.

In path based analysis, since one path is considered at a time, the effect of convergent
paths is not very significant and is generally not considered. However, NLOPALV
framework has the potential to be extended to block-based SSTA, where convergent
paths will be significant. We have looked into approaches such as application of
operating point in computing MAX at the nodes of convergence and introducing a
correction for the path convergence in the PDF computed using normal NLOPALV
analysis. Our future work will focus on exploring these areas in more depth.

The SSTA analysis is specific to the PVT conditions used for analysis. Any change
in these conditions requires a re-run of the entire SSTA analysis. An area of great
interest would be to analyze the effects of change in PVT conditions on the operating
point analysis and try to develop a heuristic approach that could be used to determine
a close enough estimate of the statistical circuit performance based on the nominal
delays obtained from STA analysis at low voltage.

The concepts behind the Non-Linear Operating Operating Point Analysis are very
general mathematical concepts and could be applied to a wide range of problems where
statistical performance determination is necessary and the individual components
could be modeled as PDFs.

One very important application in circuits is statistical power analysis. As the tech-
nology scaling continues, low voltage operation of ICs not only results into statistical
delay variations but it also leads to statistical variations in power. For ultra-low power
applications, it is critical to be accurately able to predict the power consumption of
the IC. The SSTA design methodology could be extended to take into account vari-
ations in power consumption and perform statistical power analysis on the ICs.
Bibliography


