

Test Vector Extraction Methodology

For Power Integrity Analysis

Master of Science Thesis in Integrated Electronic System Design

MARTIN OLSSON

The Author grants to Chalmers University of Technology the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology store the Work electronically and make it accessible on the Internet.

Test Vector Extraction Methodology
For Power Integrity Analysis

MARTIN OLSSON

© MARTIN OLSSON, June 2010.

Examiner: Prof. Per Larsson-Edefors

Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Cover:

A 32-bit microprocessor power grid with the supply voltage deviation at the found maximum, along with time-domain representation of voltage fluctuation at the worst node.

Department of Computer Science and Engineering
Göteborg, Sweden, June 2010

TEST VECTOR EXTRACTION METHODOLOGY FOR POWER INTEGRITY ANALYSIS

MARTIN OLSSON

Department of Computer Science and Engineering

CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2010

Test Vector Extraction Methodology for Power Integrity Analysis

MARTIN OLSSON

Department of Computer Science and Engineering

Chalmers University of Technology

ABSTRACT

In order to decrease performance pessimism due to supply voltage uncertainties in integrated circuits, detailed power integrity analysis is necessary. Knowing the worst-case voltage drop that the circuit will encounter is a step towards this goal. The voltage drop is input-dependent, which means the outcome depends on how the chip is used.

In this thesis, methods to extract the worst-case clock cycle out of a microprocessor run-trace are developed. The methods considered are based on time-based power simulations, considering full-chip total power in several time-resolutions, frequency based approaches using FFT and wavelets, and the spatial locality of switching activity. SPICE voltage drop simulations are performed while considering R and L components of the power grid, as well as decoupling capacitance and the gate switching extracted from the run-trace.

Results show that the voltage drops found when focusing on spatial locality exceed the previous worst-case for the chip design by a factor of 2. This method considers the worst-case power grid node, finding the time-instance where maximum power dissipation of its adjacent nodes coincides with the maximum power dissipation of the chip's CPU core.

Attempts at alleviating these sparse and localized large voltage drops are performed through the use of skew-spreading. This method is shown to decrease the largest voltage drop found by over 20%.

KEYWORDS:

Power Integrity, IR drop, VLSI, Test Vector, VCD, $L \, dI/dt$, Skew Spreading

Contents

List of Figures	vii
List of Tables	viii
Dedication	ix
Acknowledgments	xi
1 Background	1
1.1 Power Integrity Analysis	2
1.1.1 Static IR Drop, Dynamic IR Drop and Power Integrity	2
1.2 The Power Grid and Chip Design Under Study	3
1.2.1 The Power Grid Model	4
1.2.2 Modeling Switching Information	5
1.3 The Choice of Test Vectors	6
1.3.1 Test Vector Extraction	6
1.3.2 The PrimeTime PX Power Trace Waveform file	9
1.4 Results of Previous Research	10
2 Methods for Test Vector Extraction	11
2.1 Power Analysis in PrimeTime PX	12
2.1.1 Total Power Analysis	12
2.1.2 Extracting Local Power Trace Information	13
2.2 Spatial Locality	14
2.2.1 Combining localities to find worst-case vectors	16
2.2.2 Finding Worst Node on Chip	18
2.3 Frequency Domain	28
2.3.1 Finding the chip resonant frequencies	29
2.3.2 FFT based approaches	30
2.3.3 Wavelet based approaches	30
2.4 Cell-level Power Information	32
2.4.1 Types of Cells Distribution for Test Vectors	33

2.4.2	Alleviating Simultaneous Switching Noise Through Skew Spreading	36
3	Results	38
3.1	Full-chip Maximum Power	38
3.2	Frequency and dP/dt based approaches	41
3.3	Localized Voltage drops: Finding the Where and When	43
3.3.1	Choosing the Worst Node	45
3.4	Voltage Drop Characteristics and Worst-case	49
3.4.1	The effect of on-chip inductance on voltage drop	51
3.4.2	Chip-wide distribution of voltage drop	53
3.5	On-chip Inductance and Bond-wire Inductance	55
3.5.1	Frequency Components and Inductance Contributions	59
3.6	Results of Skew-Spreading on Voltage Drop	60
4	Discussion	63
4.1	Weaknesses in Model and Methodology	63
4.2	Timing implications	64
4.3	Consequences for design flow	66
4.4	Conclusions	67
	Bibliography	68

List of Figures

1.1	Power Grid of the Chip Design Studied	4
1.2	Workflow of Current Source Generation	7
1.3	The powergrid model in two layers	8
2.1	Illustrating the relationship between <i>units</i> and <i>nodes</i>	14
2.2	Illustrating nodes and adjacent nodes	15
2.3	Linear fit of current to voltage drop	16
2.4	Overlapping top-list entries 100ns List index	18
2.5	Total power dissipation per node	20
2.6	Maximum power per node, timescale 1 ns	22
2.7	Power for adjacent nodes at node power maximum, timescale 1 ns . .	23
2.8	Maximum power per node, timescale 0.1 ns	25
2.9	Power for adjacent nodes at node power maximum, timescale 0.1 ns .	26
2.10	Power for CPU nodes at node power maximum, timescale 0.1 ns . . .	27
2.11	Impedance of on-chip power grid	29
2.12	Scalogram of wavelet transform, 1 ns resolution full-chip power trace .	31
2.13	Voltage Drop with different components highlighted	34
2.14	Voltage Drop with different components highlighted, worst-case drop	35
3.1	Full-chip power over time	40
3.2	Current signatures for nodes 62 and 64. Time window at node 62 power peak.	47
3.3	Current signatures for nodes 62 and 64. Time window at node 64 power peak.	48
3.4	Voltage Drop for worst node	51
3.5	Voltage drop for worst node, without on-chip inductance	53
3.6	Voltage drop magnitude distribution	54
3.7	Voltage drop by node location	55
3.8	Bond-wire vs on-chip inductance for two randomly selected nodes . .	57
3.9	Bond-wire vs on-chip inductance for two nodes	58
3.10	Frequency break-down of Bond-wire vs On-chip inductance	60
3.11	Voltage drop for worst-case node, with and without skew-spreading .	61
3.12	Effect of skew-spreading on current profiles	62
4.1	Datapath with launch and capture flip-flops	64

List of Tables

2.1	File size for waveform files by resolution	13
2.2	Correlation coefficient of voltage drop to current properties	16
2.3	Switching Cell Types in a Voltage Drop	34
2.4	Switching Cell Types in a worst-case Voltage Drop	35
2.5	Switching Cell Types in a Voltage Drop	36
3.1	Total Power Approach, time resolution Vs maximum voltage drop . .	39
3.2	Voltage drop of randomly selected test vectors	41
3.3	Voltage drop of dP/dt-based vectors	41
3.4	Voltage drop of FFT-based vectors	42
3.5	Voltage drop of wavelet-based vectors	43
3.6	Localized power results	44
3.7	Localized power results, overlapping	44
3.8	CPU and adjacent nodes overlapping time windows, Voltage Drops .	45
3.9	Test vectors found by considering each node's power maximum	48
3.10	Current profile max of two test vectors [A]	50
3.11	Bond-L vs. On-Chip L, number of dominating nodes per metal layer .	58
3.12	Bond-L vs. On-Chip L, number of dominating nodes per frequency component	60

Dedication

To Annasofia.

Acknowledgments

I would like to thank my examiner at Chalmers - Per Larsson-Edefors - and my supervisors at Atmel Norway - Johnny Pihl and Daniel Andersson.

Chapter 1

Background

Power integrity analysis deals with verifying the power supply grid of an integrated circuit. In order to gain a greater understanding of what happens in the on-chip power grid, research work has been devoted to this field as prior work leading up to this master's thesis.

With increased understanding it is hoped to be possible to move away from the pessimism introduced by the standard practice of corner-based design, where the performance of an integrated circuit is limited by the variability of the Process, Voltage and Temperature (PVT) variables.

In the prior work, a detailed model of the power supply grid of a 32-bit microprocessor was developed. Several research papers ([1] [2] [3] [4] [5]) have been published using the developed model, analyzing power integrity issues from different perspectives.

While this model is input-dependent, and the outcome of analyses made could be different for different input patterns, only one assumed worst-case stimulus has been used. Thus, there is a gap in the methodology of analyzing power integrity issues with this model.

Specifically, one clock cycle has been used under the assumption that it will cause the largest problems for power integrity. Realizing that this might not necessarily be the case, the goal of this master's thesis is to investigate the space of possible input stimuli, and develop a methodology to extract the worst-case clock cycle, defined here as the clock cycle resulting in the largest observed supply voltage deviation from the nominal value.

Starting with a primer on power integrity analysis, this chapter will then describe the power grid model considered, and the methodology of power integrity analysis in this

context. Chapter 2 will present methodologies of worst-case test vector extraction developed in the work of this thesis, and the results will be presented in chapter 3.

1.1 Power Integrity Analysis

The on-chip power supply grid of an integrated circuit must be designed carefully to be able to support the switching gates with a stable voltage. Any deviation from the nominal voltage leads to increased gate delay and might cause the chip to malfunction. Since the metal carrying the current has a finite conductivity, a large current will lead to a voltage drop according to Ohm's law, $V = IR$. Power grid wires can, of course, be made wider to reduce resistance, but it comes at the cost of increased routing congestion. The other standard way to deal with IR drop is to add decoupling capacitances. These act as temporary repositories of charge that can feed their nearby gates, and are implemented on-chip using transistors. [6] is a good resource on this subject.

Ensuring that a chip design's power grid will satisfy its current demand is known as power integrity analysis. It is an important step in any chip design sign-off stage, but recent years miniaturization of integrated circuits comes with new issues that must be considered.

A supply voltage drop is dominated by two terms, such that the voltage drop can be described as $V = IR + LdI/dt$. The first term, or *IR-Drop*, has been the main focus of traditional power integrity analysis.

While the parasitic inductance, L , in the on-chip power grid and the bond wires connecting the die to the package has always been present, dI/dt has traditionally been negligible. However, as feature-sizes in integrated circuits become smaller with each new process node, transition times become shorter and the LdI/dt term becomes more important due to the faster current rate-of-change.

Technology trends have also gone towards decreasing the supply voltage in order to reduce power dissipation. This makes power integrity analysis still more critical, since the noise margin of the power supply decreases when the difference between the supply voltage and the threshold voltage of the transistors, V_t , decreases [7].

1.1.1 Static IR Drop, Dynamic IR Drop and Power Integrity

The most common practical way of working in a real chip sign-off, is to perform IR drop analysis through use of EDA (Electronic Design Automation) tools. The

analysis is usually a static IR drop analysis, which means only average power dissipation is considered along with the dimensions and technology data of the power grid. This gives an average voltage drop across the chip [8]. Using a corner-based design methodology, the power grid is verified by making sure that the average voltage drop as found by the static IR drop analysis falls below some specified limit.

Dynamic IR drop considers instantaneous current surges and can thus find voltage drops due to brief current demands in some areas of the chip. In order to perform a dynamic IR drop analysis, gate switching patterns must be applied which are usually not available until late in the design flow. Statistical switching probabilities can be used but for greater accuracy, use-case data is preferred.

The inclusion of inductance into power grid models turns the IR drop analysis into a power integrity analysis. Apart from the IR bit of voltage drop, we now have the $L \, dI/dt$ drop caused by fast current transitions. This behavior can only be captured using dynamic analysis methods. To avoid confusion, the term *voltage drop* will be used in this thesis to mean the combined efforts of IR and LdI/dt drop, and *power integrity analysis* will be used as a general term for static and dynamic methods.

1.2 The Power Grid and Chip Design Under Study

The chip design studied in this master's thesis is an AVR32 32-bit microcontroller from Atmel. It is designed in a 130-nm process using 6 metal layers, running at clock frequencies up to 200 MHz. The chip dimensions are approximately 5x5 mm and the nominal supply voltage is 1.2 V.

In figure 1.1, the chip's powergrid consisting of horizontal and vertical stripes as well as ring and block rings is shown. The power and ground nets are routed in a *paired grids* configuration, such that a vdd conductor is placed close to a gnd conductor, with a larger pitch to the next pair of vdd/gnd conductors.

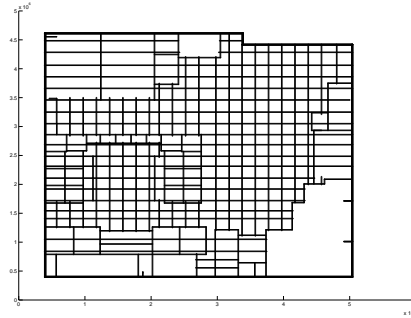


Figure 1.1: Power Grid of the Chip Design Studied

1.2.1 The Power Grid Model

As part of previously published research in the Atmel / Chalmers University of Technology collaboration, an extensive power grid model has been developed, along with a work-flow for extracting the model from a chip design. This section will describe the model and the work-flow in order to increase understanding of the work made as part of this master's thesis, which will be presented in chapter 2 through 4.

In this work-flow, sign-off data and extracted parasitics are used to create a SPICE netlist representing the power grid. In the context of this thesis, SPICE netlists created through this work-flow are simulated using HSPICE to get voltage levels at each node of the power grid.

First, the power grid geometry is extracted from the design's DEF (Design Exchange Format) file, which describes the physical layout of the chip. In the power grid model, only the vdd net is modeled. The gnd is considered an ideal ground sinking all currents without introducing return currents through the grid to supply pins. This point will be discussed later in this thesis.

The extracted geometry is then fed to a commercial field solver, Synopsys Raphael, that creates a SPICE netlist of inductances and resistances representing the parasitics in the power grid given specific process data. Vias connecting metal layers are modeled as resistors.

At this point, the power grid is a passive network of inductance and resistance only. To make some proper simulations, some notion of the active components of the chip must also be included.

1.2.2 Modeling Switching Information

In order to model the gates of a design while keeping simulation times reasonable, some simplifications must be made. In this model, all gates are modeled with the same current waveform, which is the current consumption curve of a standard-sized NAND2 gate in the process used. For each gate in the design, this base current waveform is shaped and scaled according to their individual load capacitances and rise times.

The model is simplified further by lumping gates together in a fixed number of nodes. Each intersection of the power grid is defined as a node. In each node, nearby gates are lumped together.

The procedure thus far is as follows:

1. For each gate, create current waveform according to C_{load} and t_{rise} .
2. Find closest node.
3. Superimpose the current waveforms of all gates belonging to the same node.

To connect this to the power grid model, the nodes' current waveforms are modeled as ideal time-varying current sources attached between the power grid nodes and the ideal ground. In parallel with the current sources, a capacitance is placed between the node and ground. This capacitance represents both the capacitance implicitly present at non-switching gates, as well as explicitly added decoupling capacitance. Two layers of the model is shown in figure 1.3 with resistance, inductance, decaps and current sources.

The above treatment of gates assumes that all gates of the chip are switching, which is not the case. In fact, only a small portion of the chip's gates are switching at a given point in time. Thus, we need a way to capture actual switching of the gates.

Such a switching pattern can either be vector based or vector less. Vector less approaches usually use some probabilistic switching. A more realistic and accurate scenario is achieved using a vector based approach.

From running a simulation of the design, a VCD (Value Change Dump) file can be extracted. A VCD file describes all value changes in a simulation run and is thus event-based rather than cycle-based. In the flow of this model, a part of a VCD file is used together with chip-extracted RC parasitics in a combined EDA / in-house script setting to create the intermediate `.cs` format (which is a specific format for this flow, and not a standard format).

The developments so far are summarized in figure 1.2. Here, intermediate input/output files are stated to increase understanding. From the figure, the “Processing Tools” and “CS Tool” are in-house scripts specific for this flow. Special attention will in the following chapters be given to the dotted box “Test Vector Extraction” and the intermediate `.out` file.

The resulting `.cs` file contains, for a certain switching pattern as defined by a part of a VCD file, a list of all switching gates. The gates are represented with their geometrical coordinates on the chip as well as their load capacitance and rise time, as described above. Also, all switching of the gates is presented along with their respective times.

This step connects the VCD switching information with the power grid model and a realistic switching pattern can be applied to the nodes of the grid to emulate transistors drawing current. The `.cs` file is then used with the Raphael extracted grid data to put together a SPICE netlist using an in-house scripting flow. This process is explained in great detail in [5].

The next section explains the “Test vector Extraction” step that selects a part of the VCD to use as switching information in the power grid model.

1.3 The Choice of Test Vectors

With a vector based approach, the current waveforms that load the power grid is very much a function of how the chip design is used. Early in the design flow, this information might not be available. In the case of this design, a simulation test case was available. It does not represent a real in-field use case, but it is a realistic application with software running on the CPU of the microcontroller. The duration of this test case is $460\mu\text{s}$. It involves the start-up phase of the microcontroller, as well as the execution of a software application computing the Fibonacci series.

1.3.1 Test Vector Extraction

Out of the test case, a short time window is chosen to create the current source switching information. In this context, the time window chosen should represent the input pattern creating the greatest challenge for the power grid, in order to see what the largest voltage drop that the chip experiences will be. Choosing this time window corresponds to the dotted “Test Vector Extraction” box in figure 1.2, which takes its input from the “PrimeTime PX” box above it. PrimeTime PX is a power analysis

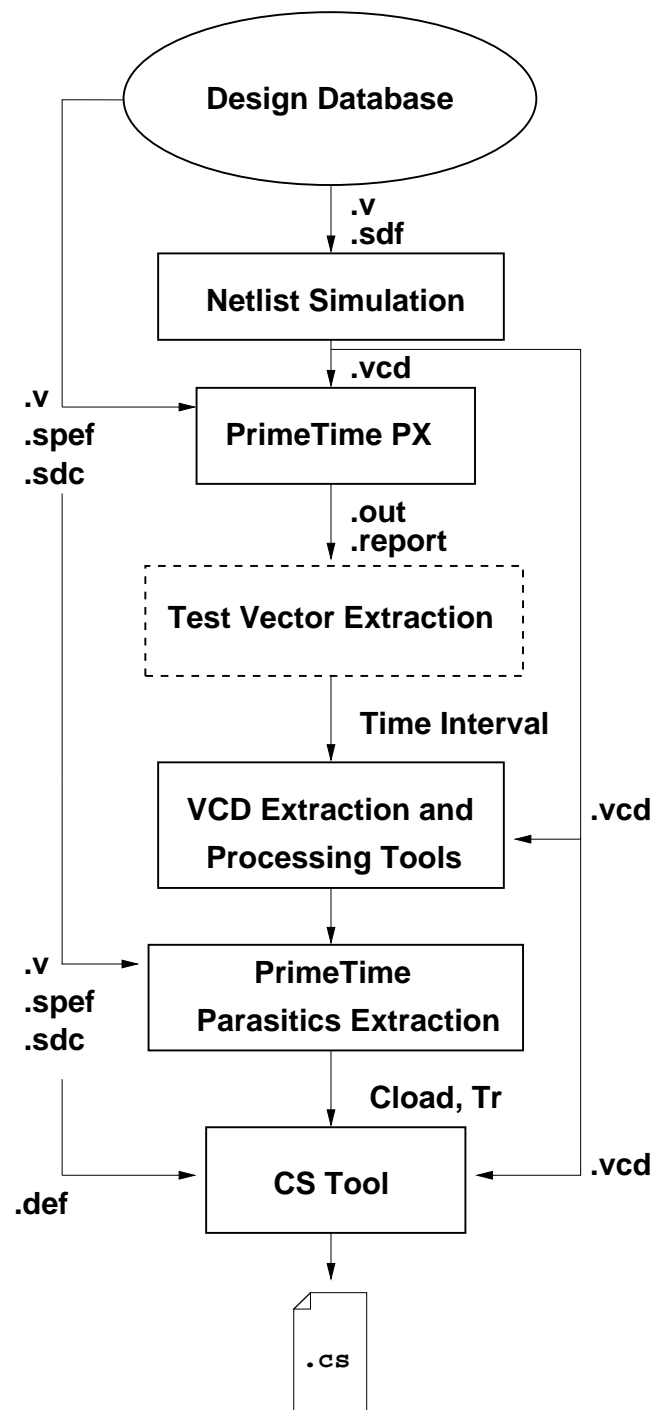


Figure 1.2: Workflow of Current Source Generation

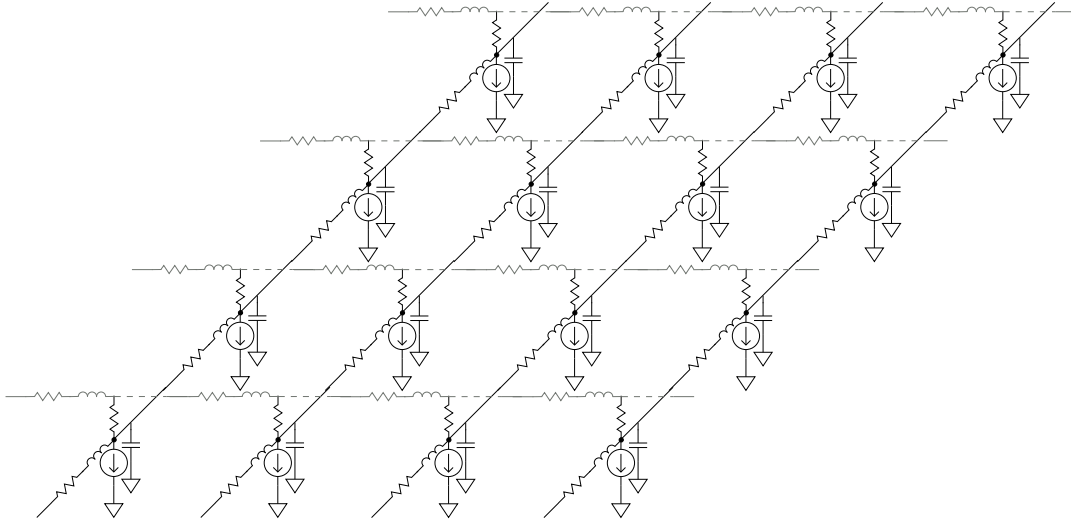


Figure 1.3: The powergrid model in two layers

tool from Synopsys [9].

In previous work using this grid model ([1]), the test vector extraction has been based on PrimeTime PX's top power consuming clock cycle from its detailed report. This approach is a common method for selecting a vector and has been used in other studies on IR drop, as mentioned in [10] which argues the method is non-optimal in terms of worst-case voltage drop. This master's thesis deals with replacing the dotted box and investigating the power grid using test vectors extracted with a new methodology, which as we shall see is based on the intermediate .out file from figure 1.2.

PrimeTime PX supports both average and time-based power analysis. In the work of this thesis, the time-based power analysis has been used. By providing a gate-level VCD file containing all switching activity of a certain run trace, detailed power dissipation data of every part of the chip is reported. Power consumption is calculated over intervals of time as specified by the user. For example, a 5 ns time interval power analysis sums all energy consumed over the 5 ns period and divides by that time to get the power dissipation.

Other than reporting power dissipation of different parts of the chip, PrimeTime PX can be made to produce power dissipation waveforms. These waveforms specify power dissipation for every instance of time, defined as above. The waveforms contain power information either for all hierarchical levels except leaf cells, or for all hierarchical levels including leaf cells. That is, they contain not only the total chip power dissipation per time unit, but power dissipation per time unit *for each cell*. With the leaf cell

option, the number of power waveforms becomes extremely large. On the other hand, the leaf cell option enables fine-grained monitoring of power consumption and creates a direct mapping between power consumption and cell instances defined in the DEF file of the design.

1.3.2 The PrimeTime PX Power Trace Waveform file

The power waveforms from PrimeTime PX can be given as either a binary FSDB file, or a text-based .OUT file. Both formats can be used to graphically view the results in waveform viewers, but the text-based .OUT format is easier to use for extracting information to be post-processed using other applications.

The power waveform .out file format is exemplified below. First, all cell instances in the design are listed with instance name and an instance number given by PrimeTime.

```
.index Pc(INSTANCENAME) 4532 Pc
.index PC(HIERARCHICAL/INSTANCENAME) 3211 Pc
```

Including leaf cells, the number of cell instances with index numbers were for this design over 400 000. Next, for every time instance, the time is followed by all cell instance numbers switching during that time interval, along with the power it consumes for that time.

```
1
342 1.401e-06
721 3.130e-05
2
523 4.232e-04
211 2.144e-03
```

For each time instance, there can be a huge number of cells listed with power information. The interval between time instances is the time interval defined by the user at the start of the analysis and is repeated in the header lines of the .out file. (.time_resolution 0.1).

The next chapter will describe the methodology developed in this master's thesis which starts with parsing the PrimeTime PX .out file.

1.4 Results of Previous Research

The work-flow of the power grid model used in this thesis has previously been used in the Atmel/Chalmers research collaboration, where several publications have been made based on variations of the model.

In [1], supply voltage drops were studied and the importance of on-chip inductance was investigated. Here, the test vector which has the highest chip-wide power dissipation over one clock cycle is used as stimulus. The article concludes that on-chip self inductance can have a significant impact on voltage drop, while the effects of mutual inductance remain inconclusive.

Derating of timing margins is the focus of [4]. The same setup as above is used. The article claims derating could be decreased, since voltage drops found are too small to necessitate the standard 10% supply voltage derating. Note that also in this article, the worst-case test vector was assumed to be the most power consuming clock cycle.

A different power grid extracted in the same work-flow was investigated in [2] and [3]. Here, different variables of power grid design were analyzed with the conclusion that supply rail width is the most important design variable. In this approach, all current sources were switching simultaneously instead of extracting switching patterns from VCD.

Chapter 2

Methods for Test Vector Extraction

The goal of this master's thesis is to find the test vector out of a large VCD trace that results in the largest possible voltage drop in the modeled power grid. To select one 200 MHz clock cycle out of a $460\text{ }\mu\text{s}$ long trace, a choice among approximately 92 000 possible cycles must be made. Performing SPICE-level simulations on each of the possible choices is infeasible, so some method is needed to locate the most interesting cycle among the thousands of alternatives.

One initial consideration when cutting out a part of a VCD trace, is the length of the time windows chosen. Too large time windows lead to slow processing times for generating SPICE netlists, and also long SPICE simulation runtimes. If the time window is too short, we might miss some interesting behavior in the switching activity. In the results presented in this thesis, the time window was chosen to be 200 ns.

Starting with a detailed power simulation of the entire trace, one can begin to make qualified guesses at which cycles would generate the largest voltage drop. One such guess could be made by assuming that the cycle which consumes the most power of all the cycles in the trace is the one causing the largest voltage drop. This is readily available from the information in the power simulation, and has commonly been used as a metric for finding the worst vector ([1], [10]).

However, this qualified guess would not necessarily give the largest voltage drop, as was shown in [10]. In this section, several different methods of making qualified guesses of cycles that give large voltage drop will be presented. The different methods all rely on power information as given by PrimeTime PX, but differ in the way the power information has been gathered and how the information is analyzed.

2.1 Power Analysis in PrimeTime PX

A pre-processing flow has been developed by the author to analyze the PrimeTime PX power waveform. Because of the large amounts of data in the waveform file, as was described in the previous chapter, parsing the file can be quite time consuming if care is not taken. In fact, the .out files for this particular design and simulation trace can be over 8 GB in a gzipped (compressed) format.

To cope with the large amounts of data and create a good balance of performance and flexibility, a combination of C, Python and UNIX shell scripts has been used. The method and the resulting data produced will be described in this section.

2.1.1 Total Power Analysis

To analyze power dissipation of the entire chip over time, the power trace waveforms can be used. Waveform viewers can read the file directly, but to work with the data in a more flexible way, some preprocessing is necessary.

In the total chip power context, four items of information was initially of interest:

- 1: Time-Power data,
- 2: Time-dP/dt data,
- 3: Top 100 Power values,
- 4: Top 100 dP/dt values.

This means that for each instance of time, the total chip power dissipation must be calculated. Also, the running differential between two adjacent time instances was calculated to find possible dP/dt peaks. In the same processing step, top-lists of power and dP/dt was maintained by the program. The program was implemented in C, using UNIX tools cat and zcat to pipe data from the huge .out files in either a gzipped or uncompressed format. A shell script was used to manage analyses of different waveform files in different time resolutions and to order the output appropriately in the file system.

The time resolutions of the power analysis and power waveforms from PrimeTime PX, as mentioned in section 1.3.2, is an important consideration. We will see later on that it can affect the choice of test vectors. The time resolution can be adjusted by the user, but waveform files grow quickly with increased resolution as summarized in table 2.1. The file sizes are given in gzipped format, the leaf option corresponds to including the leaf cells of the design in the power waveform. With leaf, resolutions above 0.1 ns were neglected due to the large file sizes.

Resolution	File size (leaf)	File size (no leaf)
5 ns	3.5 GB	445 MB
1 ns	5.9 GB	825 MB
0.1 ns	7.4 GB	1.3 GB
0.01 ns	-	2.5 GB
0.001 ns	-	4.2 GB

Table 2.1: File size for waveform files by resolution

2.1.2 Extracting Local Power Trace Information

As we shall discuss later, it is sometimes useful to get a power trace of a subset of the full chip. The PrimeTime PX trace contains (Time:Instance Number:Power) tuples and can be used for this purpose. A specification of localities more coarse-grained than cell instances is usually wanted, and thus an intermediate conversion step is needed. To this end, a Python script has been developed, that takes a list of nodes (as defined in section 1.2.2) and produces a (longer) list of instance numbers.

Each .out file contains a list of (Instance Number \rightarrow Instance Name) mappings (refer to the example in section 1.3.2). This list is the same for all power analyses of the same chip design. For each design, a DEF file containing (Instance Name \rightarrow X-Y Coordinates) mappings is usually available. Since these mappings never change within the same design, performance is increased by only performing the translation (Instance Number \rightarrow X-Y Coordinates) once. Python has a programming construct called Pickles that is useful for this. A binary file is saved as output from the translation, which is implemented as a hash table that can be quickly indexed to find the mapping.

To realize the (Node \rightarrow Instance Numbers) mapping we need a rule that tells which instances should belong to which node. In the flow of section 1.2.2, each gate is lumped to the node closest to its coordinates. The same approach is used here: For each node stated with x-y-coordinates, list all instances which has that node as its closest node (by using the Instance Number \rightarrow X-Y Coordinates mapping from above).

This procedure gives us the more coarse-grained locality we are after. Taking the pickle approach one step further, the results of the above will be pickled since it does not change within a design. We now have, for each node, a list of all the unit numbers (as defined by PrimeTime PX) belonging to that node.

The relation between *units* and *nodes* is illustrated in figure 2.1. Cell is used inter-

changeably with unit in this thesis.

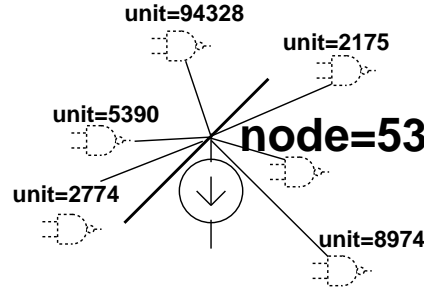


Figure 2.1: Illustrating the relationship between *units* and *nodes*

The found mapping of (Instance Number \rightarrow Node) is now combined with the C program used for extracting power data from the waveform files. A list of nodes is given to the Python program that produces a list of instances. The list of instances is then given as input to the C program, that will then create the four items of interest stated in section 2.1.1, *but only for those instances belonging to that subset of nodes*.

This gives us a general approach for extracting power data for any part, large or small, of the chip. This ability will be useful in finding a worst-case vector, as will be explained in the next section.

2.2 Spatial Locality

When a test vector has been extracted and a SPICE simulation has been performed using switching data of that test vector, the end result is voltage drop waveforms for all nodes in the power grid. Also, current profiles for all current sources loading the power grid are available.

It would seem reasonable that a large current drawn in a power grid node would always result in a large voltage drop in that node. To some extent this is true, but results obtained in this thesis show that there is no absolute one-to-one mapping between current and voltage drop. This section will investigate this relationship, while introducing spatial locality of currents and how it can be used for test vector extraction.

To investigate the property of spatial locality, experiments involving 47 test vectors were performed. Each test vector's maximum drop was noted along with the node where the drop occurred. For each vector, current profiles for the worst node and for the nodes directly adjacent to the worst node was loaded. The concept of adjacent

nodes is illustrated in figure 2.2. In the figure, the middle node is the node with the observed maximum voltage drop. Current sources with filled lines indicate the currents sources considered for the two cases. Node current means the current drawn by the current source in the node with the maximum voltage drop. Adjacent nodes current mean the superimposed currents of the two nodes directly adjacent to the node with the maximum voltage drop.

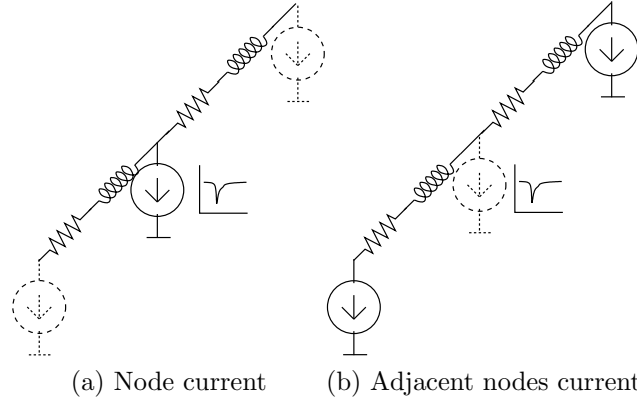


Figure 2.2: Illustrating nodes and adjacent nodes

The goal of the experiment was to see how great the correlation between a node's current consumption and its voltage drop was. Furthermore, the experiment investigates the correlation between adjacent nodes' current and the worst drop.

Other current properties that was explored was maximum dI/dt and maximum total power spectral density over frequencies from 0 Hz - 200 MHz.

For each of these properties, the correlation coefficient was calculated. The correlation coefficient is defined as

$$R(i, j) = \frac{C(i, j)}{\sqrt{C(i, i)C(j, j)}} \quad (2.1)$$

where $C(i, j)$ is the covariance of vectors i and j . The correlation coefficient is a standardized version of covariance that describe the way i and j influence each other [11]. A correlation coefficient of 0 means the two variables are uncorrelated, while close to 1 indicates a strong correlation.

Table 2.2 summarizes the correlation coefficients of different current properties to

voltage drop that was found in the experiment. The strongest correlation can be found in maximum currents. Also, note that adjacent nodes current have stronger correlation for all properties.

	I_{max}	dI/dt	PSD
Node	0.9782	0.8200	0.5486
Adj. nodes	0.9818	0.8969	0.6623

Table 2.2: Correlation coefficient of voltage drop to current properties

In figure 2.3 are scatter plots of node current vs voltage drop (deviation from $V_{nom} = 1.2$ V) with a linear fit. Note that some outliers have a higher voltage drop than another with a larger current.

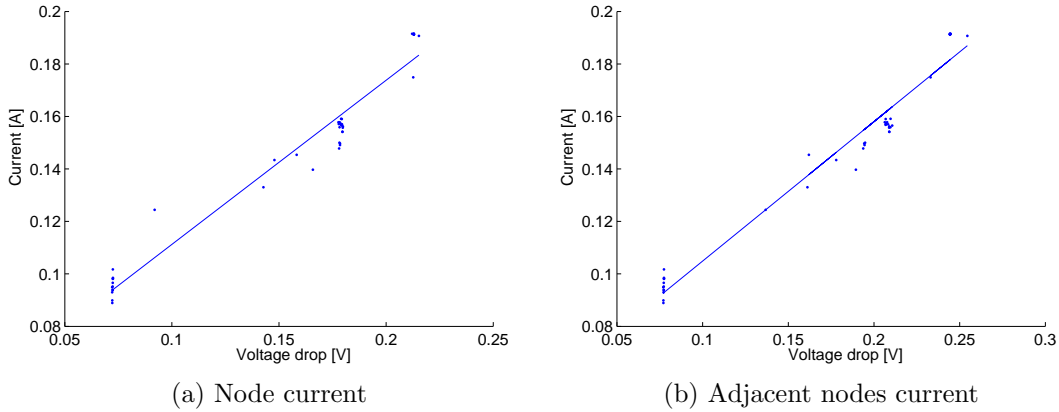


Figure 2.3: Linear fit of current to voltage drop

2.2.1 Combining localities to find worst-case vectors

The above discussion on the importance of adjacent nodes currents for a voltage drop leads us to develop a new method of extracting a worst-case test vector from the PrimeTime PX power trace.

As described in section 2.1.2, we have a way to extract all power information for a certain set of nodes. We can also list all the top power consuming time windows for that set of nodes.

Using the extracted power information and list of top power consuming time windows for a node which we somehow know to be critical, we can find the time window where that node dissipates the largest amount of power over a certain time interval.

Secondly, we can do the same for that node's adjacent nodes to get the time window where those nodes' superposed power dissipation reach a maximum.

Thirdly, a larger area of nodes surrounding the critical node can be decided upon. Here, the area holding the CPU of the chip has been decided as a larger localization of interest. The power dissipation of nodes making up the CPU is thus extracted and the maximum time window is found.

Now, since from table 2.2 both the critical node and its adjacent nodes' current correlate to the voltage drop, a combined approach is of interest. Using a programmatic approach, the power dissipation top lists of the critical node and the adjacent nodes are compared for overlaps in time. That is, if the maximum power dissipating cycle of the critical node occurs at the same time as its adjacent nodes, that cycle should be very interesting in terms of worst-case voltage drop. If the maximum two does not overlap, perhaps some other of the 100 time windows in the two lists overlap. To increase yield, a time window of 100 ns is defined in which any overlaps are considered to be simultaneous.

The overlap concept is illustrated in figure 2.4. The figures in the left indicate two top-list entries which are not overlapping. To the right, there is an overlap. The list index represents the order of significance in the top-100 list. Thus, two overlapping time windows of index 100 would be given more attention than numbers of 10 and 20. This way, the sum of indices give a total ordering of the time windows.

The three levels of localization presented above was combined in the overlap methodology to successfully identify test vectors with large voltage drop. The results are presented in the next chapter.

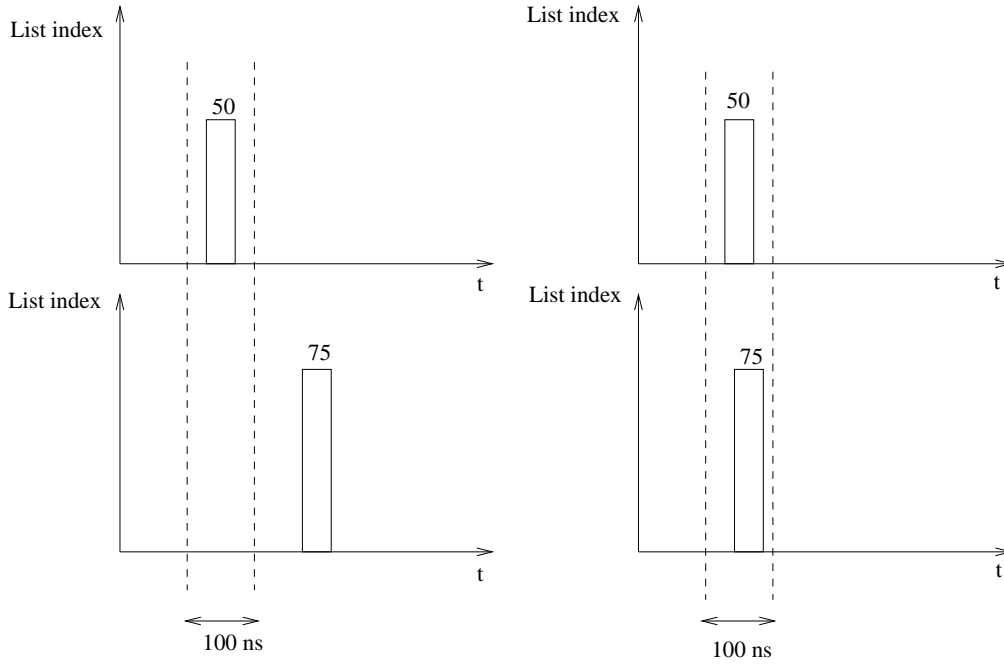


Figure 2.4: Overlapping top-list entries 100ns List index

2.2.2 Finding Worst Node on Chip

The above experiments were performed on vectors more or less randomly selected, and certain nodes were frequently more prone to have large IR drop.

Without any prior knowledge of our design and without results from earlier voltage drop simulations, can we find out which node will cause the largest IR drop from looking at the PrimeTime PX power trace?

To investigate this, an experiment was conducted where each node's maximum power dissipation per time unit is found. Using a PrimeTime PX power trace, the steps outlined in Algorithm 1 yields every node's maximum by adding, for every unit that is dissipating power in a time unit, each node's power dissipation to the corresponding node's sum. A maximum is updated and the sum is reset.

Algorithm 1 Finding the maximum power dissipating node

```

for  $t = 1 \dots \text{endtime}$  do
  for  $\text{Unit} \in P(t)$  do
     $\text{Node} \leftarrow \text{getClosestNode}(\text{Unit})$ 
     $P_{\text{sum}}(\text{Node}) \leftarrow P_{\text{sum}}(\text{Node}) + P(t, \text{Unit})$ 
     $P_{\text{tot}}(\text{Node}) \leftarrow P_{\text{tot}}(\text{Node}) + P(t, \text{Unit})$ 
  for  $\text{Node} \in \text{all nodes}$  do
    if  $P_{\text{sum}}(\text{Node}) > P_{\text{max}}(\text{Node})$  then
       $P_{\text{max}}(\text{Node}) \leftarrow P_{\text{sum}}(\text{Node})$ 
     $P_{\text{sum}}(\text{Node}) \leftarrow 0$ 

```

This procedure yields two node metrics for each node: 1) The node's maximum power dissipation over a time unit and 2) The node's total power dissipation over the entire VCD duration.

Figure 2.5 shows a 3D-plot of the total power dissipation. It tells us of one specific node with high total power dissipation. It coincides with the clock tree root and the high total is explained by noting that this area dissipates power for each clock cycle. The second most noticeable area coincides with the CPU. However, we are particularly interested in instantaneous maximums rather than how often a certain node dissipates power. Thus, figure 2.6 is of particular interest. Figure 2.6 display each node's maximum in a time instance. In (a), the maximum is plotted by node index. In (b), the nodes are displayed with the geometrical location on the chip, where darker areas mean more power. The annotated numbers correspond to the x-axis of figures 2.6(a).

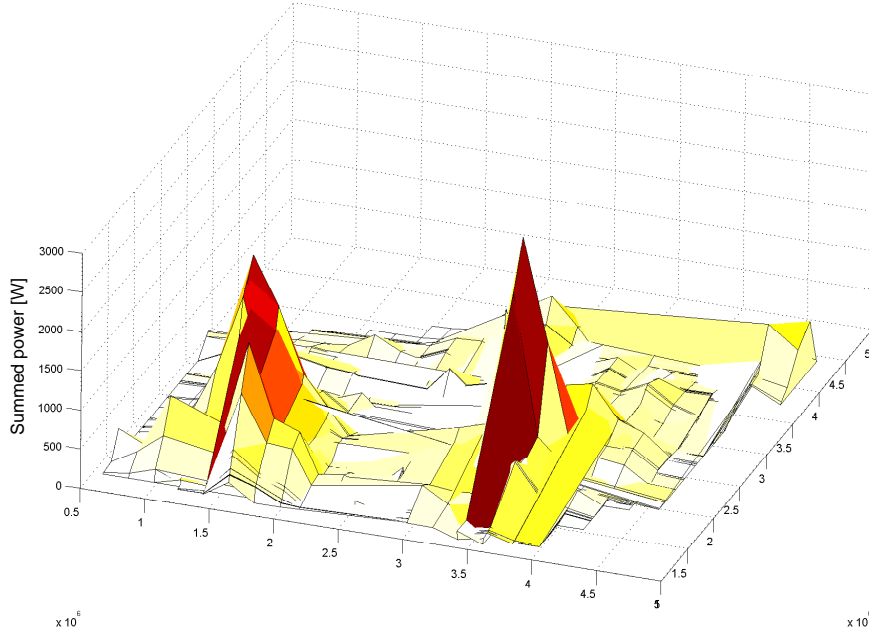


Figure 2.5: Total power dissipation per node

Since it has already been established that a node's voltage drop is influenced to a large part by its adjacent nodes currents, the above procedure can be modified to include this dependence. It could be argued that figure 2.6(b) gives the information about power dissipation in adjacent nodes, a wider dark area in the plot could correspond to more power dissipation in a certain area. However, this approach does not consider time. The plot only presents the maximum power for each node over the whole trace. The adjacent node power dissipation is of interest when occurring simultaneously with a high-peak node.

To include this consideration we modify Algorithm 1 to yield Algorithm 2 where adjacent nodes' power dissipation is noted at each node's maximum.

Algorithm 2 Finding the maximum power dissipating node, including adjacent nodes

```

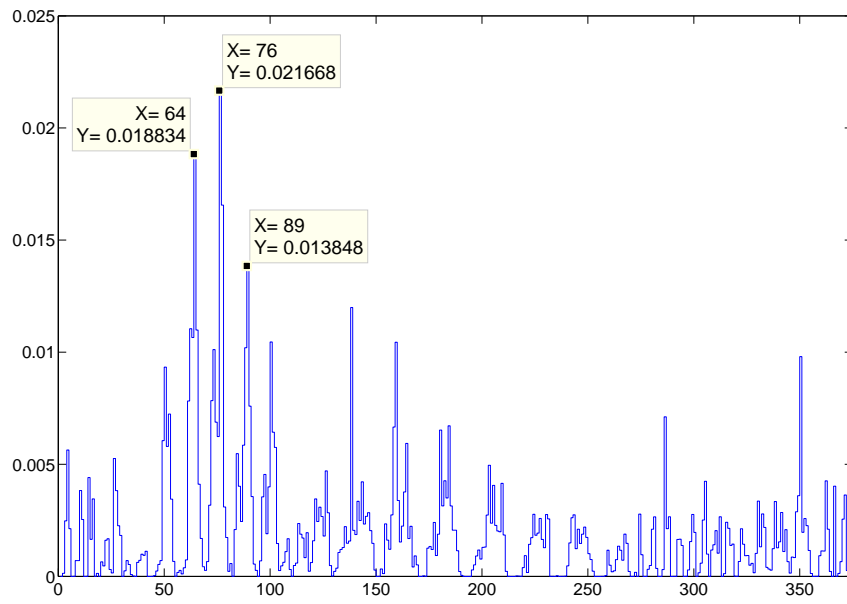
for t = 1 ... endtime do
  for Unit  $\in P(t)$  do
    Node  $\leftarrow$  getClosestNode(Unit)
     $P_{sum}(\text{Node}) \leftarrow P_{sum}(\text{Node}) + P(t, \text{Unit})$ 
     $P_{tot}(\text{Node}) \leftarrow P_{tot}(\text{Node}) + P(t, \text{Unit})$ 
  for Node  $\in$  all nodes do
    if  $P_{sum}(\text{Node}) > P_{max}(\text{Node})$  then
       $P_{max}(\text{Node}) \leftarrow P_{sum}(\text{Node})$ 
       $P_{adjacent}(\text{Node}) \leftarrow P_{sum}(\text{Node}-1) + P_{sum}(\text{Node}+1)$ 
       $P_{sum}(\text{Node}) \leftarrow 0$ 

```

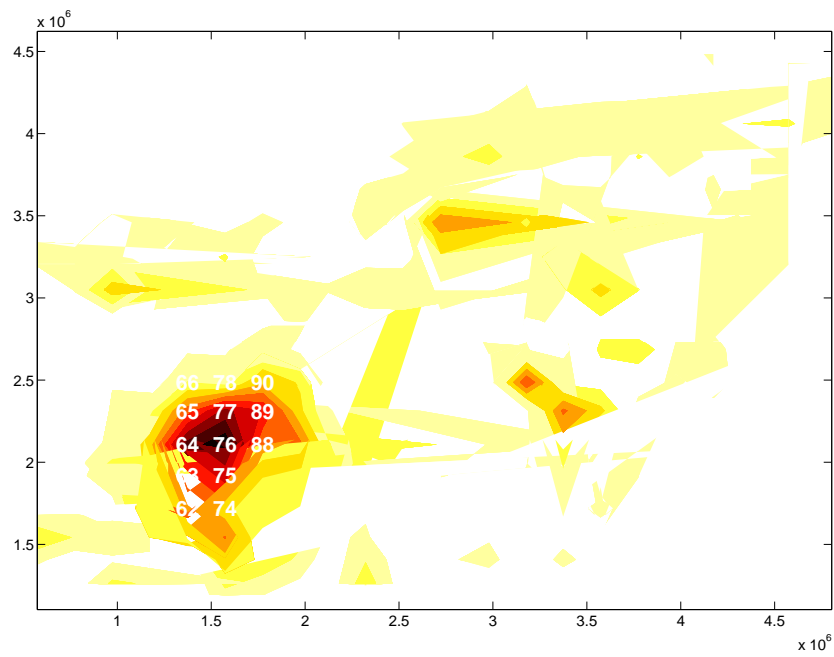
which yields a list of each node's maximum current, and its adjacent nodes' current at that time instance. To find the node that should experience the worst-case voltage drop according to this criterion is thus found by finding the node with maximum combined node/adjacent node power.

Figure 2.7 shows, for each node, its adjacent nodes' power at the instance of time where the node has its maximum.

Some interesting observations can be made regarding these figures. First, the areas of maximum power roughly make up the area of the CPU. Secondly, while node number 76 is the most power consuming node, its adjacent nodes in that time instance is not as prominent. Considering nodes/adjacent nodes, the focus has shifted to node number 63/64 and node 89.

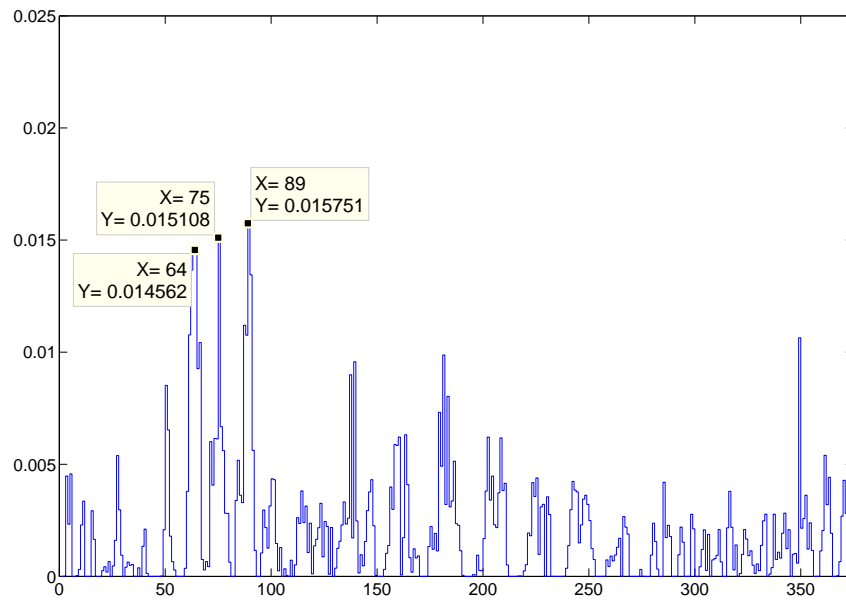


(a) By node index

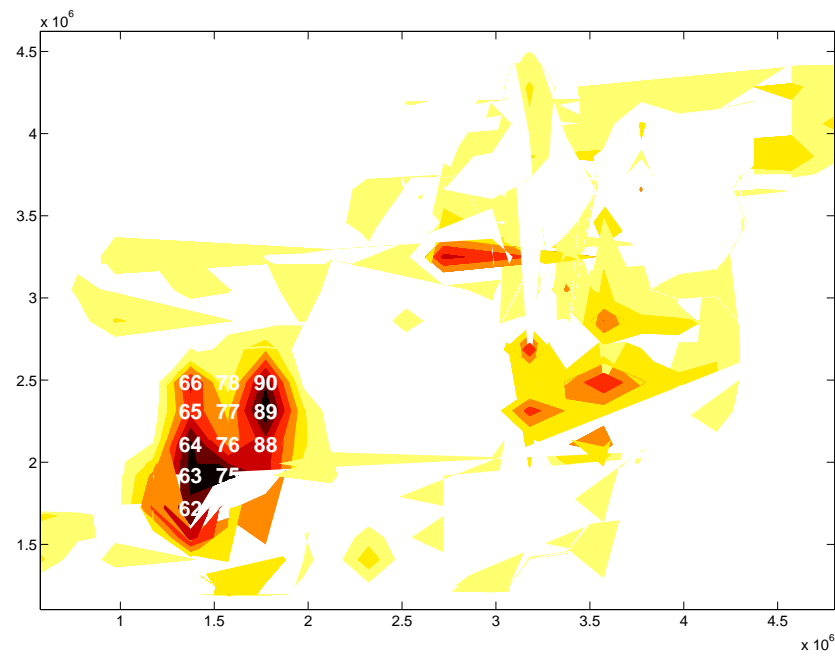


(b) By chip location

Figure 2.6: Maximum power per node, timescale 1 ns



(a) By node index



(b) By chip location

Figure 2.7: Power for adjacent nodes at node power maximum, timescale 1 ns

Thus, three nodes in particular attract our interest. Node 63, node 76 and node 89. But which characteristic of power dissipation is the most important? Will a lonely node with large maximum power dissipation cause the larger drop or will a node with relatively large power dissipation surrounded by other power dissipating nodes be worse?

For all found node maximums, the time instance will be presented along with the maximum by using the method outlined above. Thus, we can easily make test vectors from the interesting cases and evaluate the voltage drops by running SPICE simulations (results follow in the next chapter).

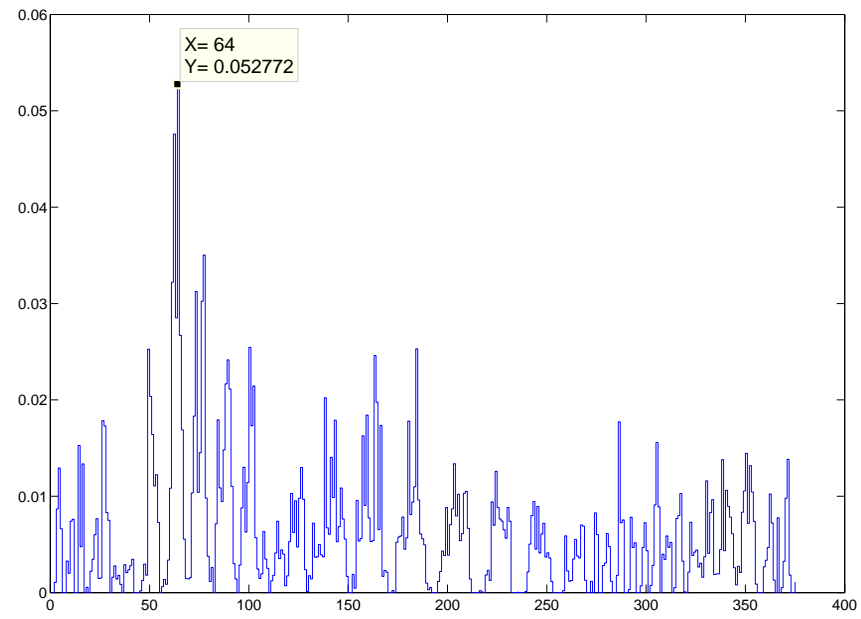
Changing the Time Resolution

As it was noted in section 3.1, it is important to select a proper time resolution when performing peak power analysis. The above experiments were repeated with the time resolution 0.1 ns, which yielded different results.

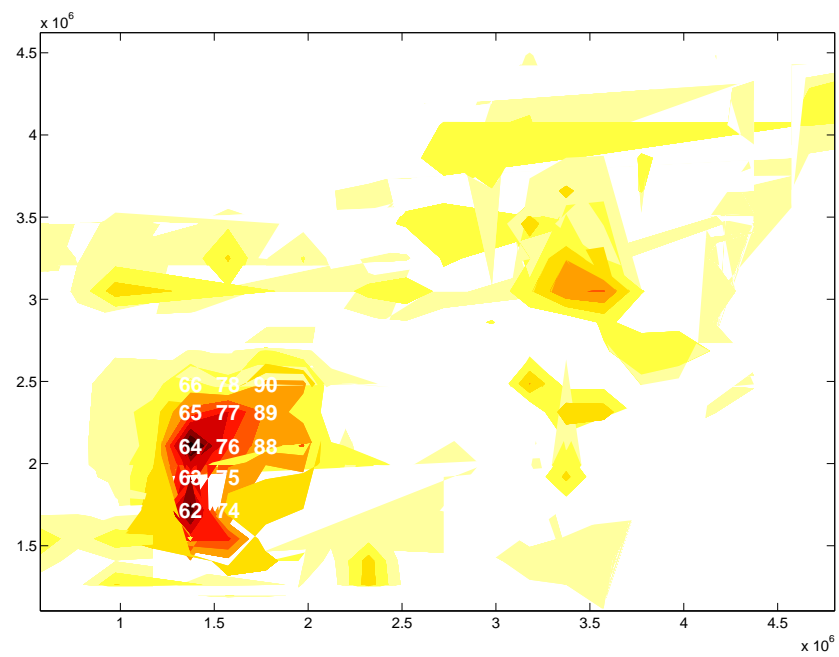
Figure 2.8 shows each node's maximum power dissipation over 0.1 ns, by node index (a) and by chip location (b). Compared to figure 2.6, focus has shifted from node 76 to node 62. Node 64 is the other node of particular interest, from figure 2.8(a) we note that it is the maximum power dissipating node.

In figure 2.9, the summed power of adjacent nodes for each node is plotted. Here, node 62 is the maximum. That means that node 62 is the node which, at its maximum 0.1 ns power interval, has the greatest power consuming adjacent nodes. Given the earlier discussion about the importance of adjacent nodes power dissipations, we could expect node 62 to experience a large voltage drop at the time when this scenario occurs.

The next figure (2.10) extends the locality concept. Here, for each node, the total power dissipation of the CPU area at the time of node power maximum, is presented. In this setting, node 64 is dominating. This means that node 64 is the node that has its maximum power dissipation at the time instance when the CPU dissipates the most power for any node's power maximum. This time instance should be particularly interesting, because node 64 was the top power consuming node (figure 2.8), and peaked when the surrounding CPU nodes were particularly active.

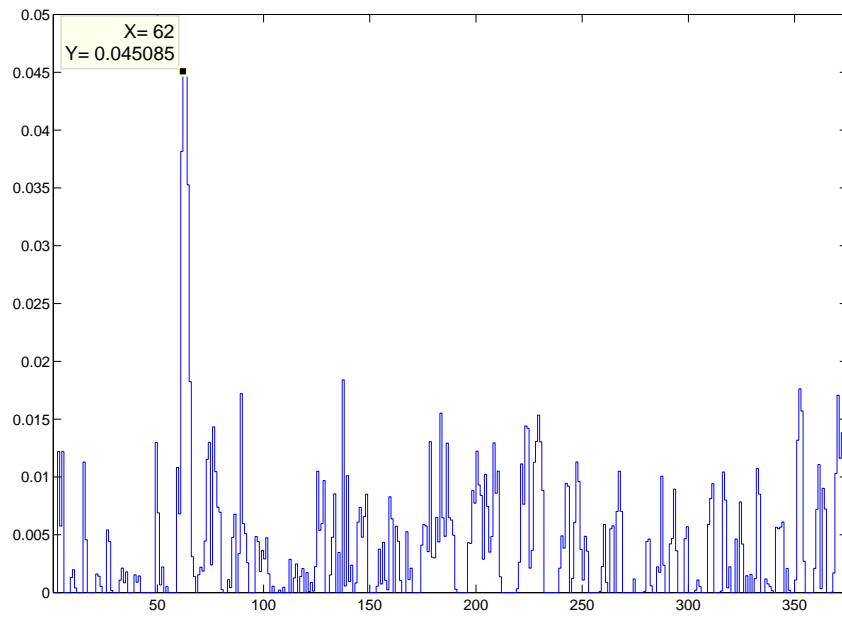


(a) By node index

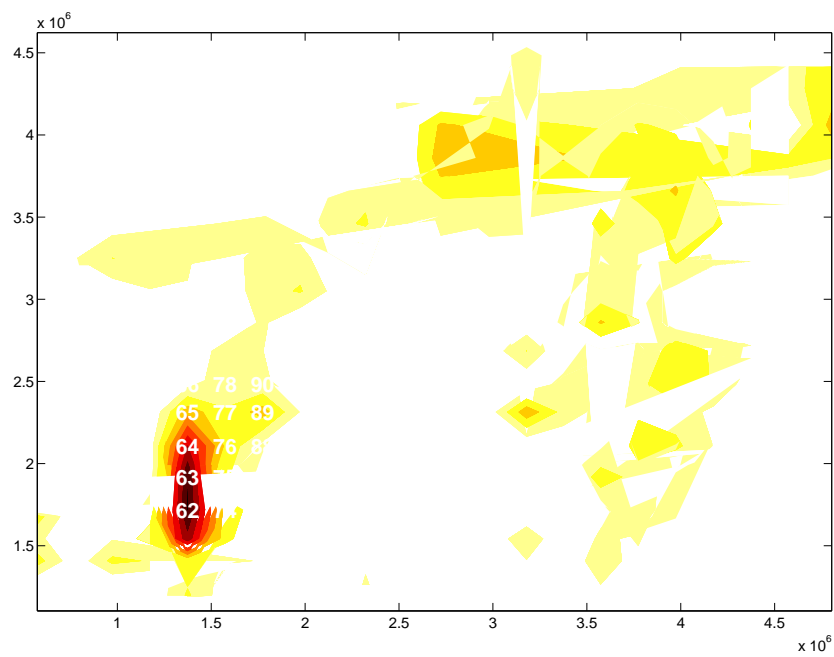


(b) By chip location

Figure 2.8: Maximum power per node, timescale 0.1 ns

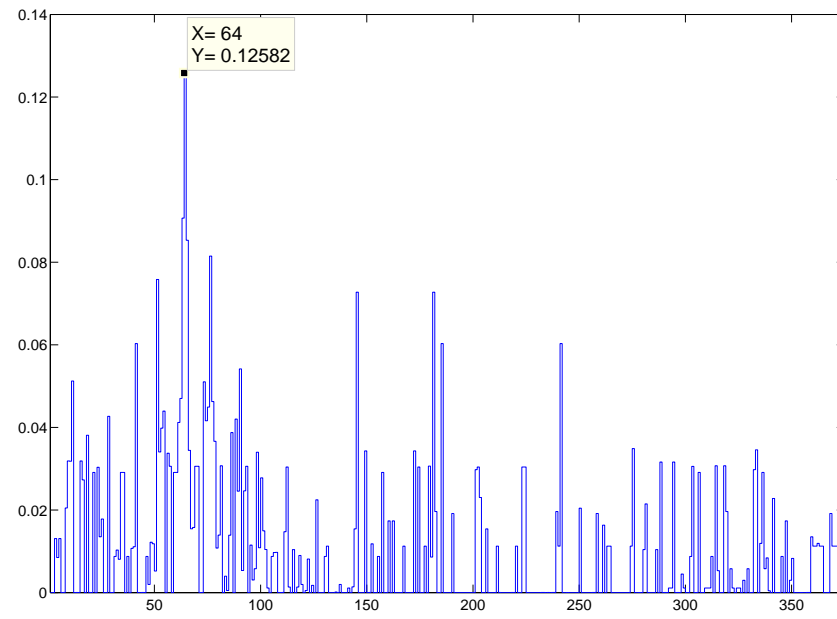


(a) By node index

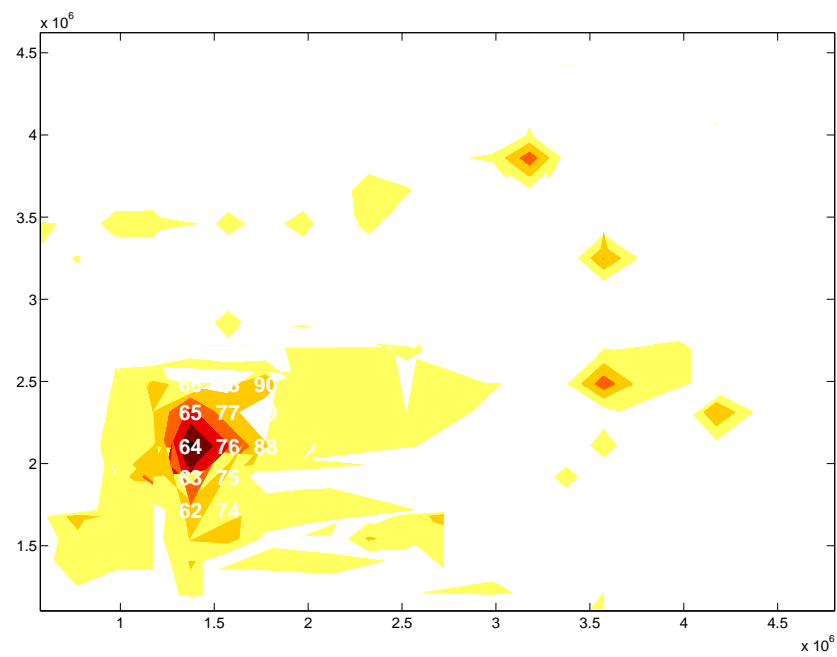


(b) By chip location

Figure 2.9: Power for adjacent nodes at node power maximum, timescale 0.1 ns



(a) By node index



(b) By chip location

Figure 2.10: Power for CPU nodes at node power maximum, timescale 0.1 ns

2.3 Frequency Domain

Since a voltage drop is made up of both IR-drop (caused by parasitic resistance in the power grid) and $L \, dI/dt$ drop (caused by parasitic inductance in the power grid), where $L \, dI/dt$ can constitute a significant contribution [1], we will investigate how the frequency dependence of the power grid relates to the voltage drop.

It is customary to characterize a circuit and power delivery system by its frequency response using an impedance plot. An impedance plot gives the system's impedance for all frequencies of interest. For a characterization of the complete power delivery system, the power supply and voltage regulators along with the PCB, die-packaging, bond-wires and on-chip power grid should be considered.

Impedance plots can be used to discover resonant frequencies in the power grid. Resonance in an electric circuit can cause energy to oscillate back and forth between magnetic energy in an inductance to electric energy in a capacitance. Another way to view the problem is to consider the impedance of a resistor and an inductor in series connection, $Z = R + j\omega L$. This impedance increases with frequencies so that fast edge-rates would become problematic. Using bypass capacitors in parallel creates an impedance of $Z = \frac{1}{j\omega C} + R + j\omega L$, which has its minimum at its self-resonant frequency $f_{resonant} = \frac{\omega_{resonant}}{2\pi} = \frac{1}{2\pi\sqrt{LC}}$ [12].

The bypass capacitor approach can thus be used to lower impedance over any range of frequencies necessary. However, several bypass capacitors in parallel can cause another phenomenon called antiresonance [6], which at specific frequencies can increase impedance dramatically (the term antiresonance describes the phenomenon more accurately, but this thesis will use the term resonance to refer to the above, unwanted, behavior). It is therefore important to carefully design the power delivery system and control the type and amount of added decoupling capacitance to keep system impedance low over the frequency range of chip operation. Note that the frequency range here includes all frequency components of switching times of on-chip signals, and not only the clock frequency.

From the above discussion, we can apply the concept of (anti) resonant frequencies to the search for test vectors causing voltage drop. If we were to find certain switching patterns with large components near resonant peaks, this could lead to high impedance and thus high voltage drop in the circuit. Some earlier research has focused on this subject. In [10], the authors claim to find worst-case vectors based on resonant frequencies rather than maximum power dissipation. Their work is based on FFT techniques. In [13], wavelets are used to create worst-case currents containing frequencies coinciding with the impedance peaks of the power delivery systems.

2.3.1 Finding the chip resonant frequencies

In order to use frequency-based approaches to finding worst-case vectors, some notion about the frequency behavior of the chip is needed. In figure 2.11 is an impedance plot of the chip, including bond-wire and on-chip parasitics. We can see an antiresonance peak at 2.9 GHz. For this simulation, an AC source of 1.2 V was placed in one of the current source nodes of the grid. The current drawn by the chip was then measured and the impedance was calculated as $Z = V/I$.

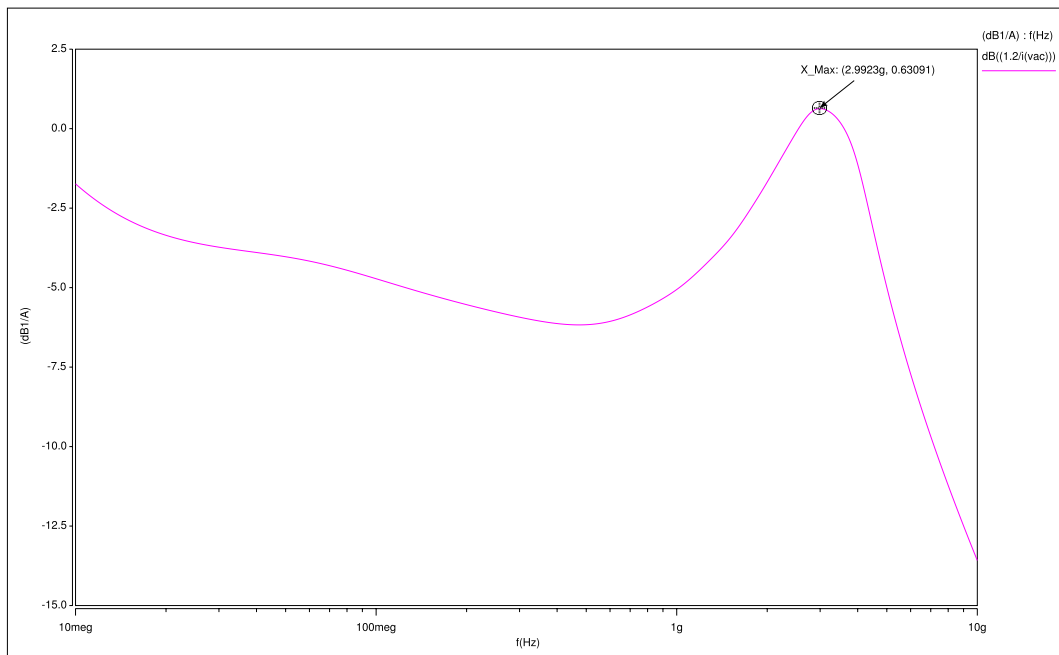


Figure 2.11: Impedance of on-chip power grid

Another simulation was performed by connecting an AC source to the external vdd supply, which will find the frequency behavior of the bond-wire RLC networks. As expected, lower resonant frequencies were identified, ranging from about 37 MHz to 180 MHz for the different supply pins.

The frequencies of interest found through these methods were used in the subsequent experiments on frequency-content of the power dissipation defined by the PrimeTime PX power waveforms.

2.3.2 FFT based approaches

The FFT (Fast Fourier Transform) based approach was influenced by the work done in [10], where a worst-case test vector from a large VCD trace is found by looking for the time window with the largest frequency components close to the chip resonance frequency.

In this thesis, the frequency-based worst-case was found in the following way. Starting with a full-chip PrimeTime PX power trace, the power data was broken down into time windows. A time window was defined to be 100 ns. For each time window, the power spectral density was calculated using FFT. The time window was then moved 10 ns forward in time to produce a new time window. For each time window, the summed power spectral density was calculated over a range of frequencies, $[f_{center} - \frac{f_{\Delta}}{2}, \dots, f_{center} + \frac{f_{\Delta}}{2}]$, where f_{center} is the interesting resonance frequency and f_{Δ} defines a tolerance range of nearby frequencies.

The time at which each interesting frequency has its maximum was noted and later used to extract a test vector from the VCD trace.

2.3.3 Wavelet based approaches

Wavelet transforms are mathematical transformations that can be used to combine the frequency-domain information of the Fourier transform with time-domain information of the signal. They have been used in many applications of signal processing, such as climate research, medical analysis, financial analysis and image de-noising and compression [14].

In the context of power integrity analysis in VLSI, a few research papers have been written that utilize the wavelet transforms ([13] [15]). In [13], the time-frequency properties of the wavelet transforms are used to construct a worst-case current stimulus depending on the system's frequency behavior. Processor architectural power simulations are targeted in [15] to predict how large the power ripple will be for a given benchmark.

In the work of this thesis, the wavelet approach and the idea of combining the time and frequency domain was combined with the ideas of [10] to extract parts of a large VCD trace with interesting frequency components.

Using Matlab and the Signal Processing Toolbox, the Continuous Wavelet Transform (CWT) of a PrimeTime PX power trace was calculated. The CWT divides the signals frequencies in a number of scales. The scales represent frequencies according to some mapping. For each instance of time, the magnitude of each scale is calculated.

This means we can get information about *when* in time certain frequency components exist. This is interesting when we are considering a long power trace and look for certain frequencies. We can immediately find the time instance where a certain frequency of interest is large.

The data produced by the CWT can be plotted graphically in a scalogram, which contain all scales and all time instances for the analyzed signal. Figure 2.12 shows such a scalogram. Below the time-domain representation of the power trace is the (zoomed-in) scalogram, with scales on the y-axis representing higher frequencies in lower scale numbers. Bright parts of scalogram represents a higher percentage of the signals energy belonging to that particular scale.

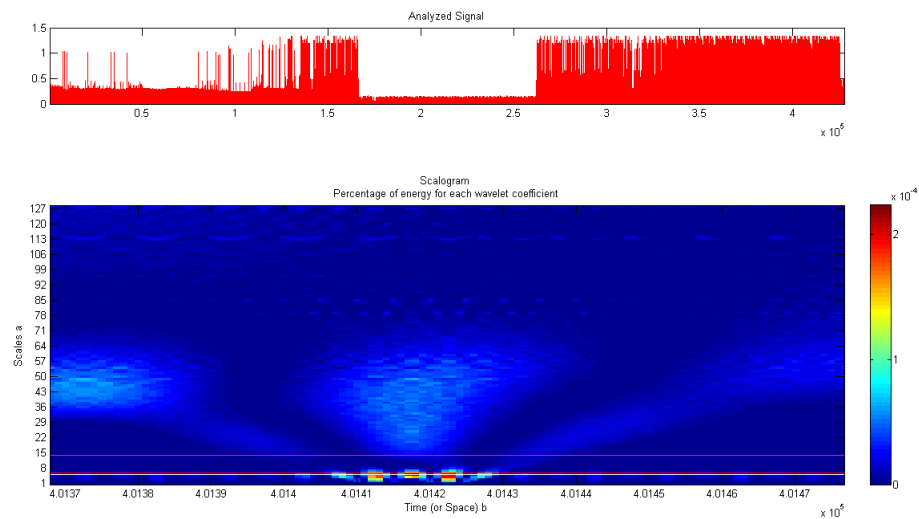


Figure 2.12: Scalogram of wavelet transform, 1 ns resolution full-chip power trace

In the context of this thesis, the scalogram can be used in three ways. First, to graphically identify regions of large activity, not caring too much about what frequencies are involved. Secondly, to focus on particular frequencies of interest, find their respective scales and graphically identify regions of high activity in those scales. The horizontal lines in figure 2.12 indicate scales of interest and the scalogram can thus be used to see where the lines coincide with particularly bright spots. Thirdly, the scalogram can be used to programmatically identify the time instances of maximum energy for certain scales. This gives the precise moment of the maximum of an interesting frequency, which can then be used to extract a test vector from the VCD.

Several test vectors have been extracted using the wavelet scalogram as a guide. Their resulting voltage drop will be presented in the next chapter. Some of them were guided by certain anti-resonant frequencies, while some were identified only by looking at regions of large overall activity. In order to capture anti-resonant frequencies identified in the on-chip grid, which are of the order of magnitude of a few gigahertz, the wavelet approach presented here could not be used. A fine time-resolution of 0.1 ns or less needs to be used to capture the high frequency content, and a high time-resolution means very large amounts of data. This caused the wavelet transform to run very slowly, and the experiments were abandoned.

It should be noted that the possibilities of using wavelet transforms are far from exhausted in this work. For example, the discrete wavelet transform (DWT) was used in [15] but it was not attempted here. Also, the choice of base wavelet used could affect the outcome. Further experiments with finer time resolutions should also be tried out. The concept of time-frequency information contained in the wavelet transform is suiting for this type of application, and if frequency information would prove to be important for the test vector extraction, the wavelet approach should be revisited.

2.4 Cell-level Power Information

Simultaneous switching noise occurs largely due to the synchronous nature of integrated circuits - the clock is designed to arrive more or less simultaneously to all flip-flops within a clock domain. When many flip-flops change state at the same time, large currents are drawn to the flip-flops, in turn causing a quick voltage drop.

This knowledge warranted a further investigation of how different parts of a voltage drop waveform is made up of different types of cells. Due to naming conventions used in the design of this chip, the cells specified in a PrimeTime PX power trace could be differentiated as flip-flop, clock buffer and combinatorial gates.

For a certain test case, an intermediate step in constructing a current source netlist is to create a file listing all cells that are switching in that test case, along with its load capacitance, transition time and the switching times for all transitions. The cells, however, are only specified as X-Y coordinates in this listing, the cell name is not explicitly stated.

Given a voltage drop waveform for a certain node, we are then interested in breaking down the cell listing into the types of cells it contains. Furthermore, we want to list those cells that switch in a certain time interval, say, the time span of the largest spike. Also, since we look at a waveform of a certain node, we want only to see the

cells belonging to that node.

The pieces of information needed for such a scheme is 1) cell home-node, 2) cell name, 3) transition times.

Each cell's home-node can be found by using the X-Y coordinates of the switching file cell listing and a mapping function designed for other purposes as part of this thesis work (see section 2.1.2). The cell name can be found in a similar manner, the chip design DEF files contain mappings from cell instance name to X-Y coordinates. The transition times are available in the switching current (.cs) file.

The algorithm for extracting the cell type breakdown is presented as Algorithm 3, and yields information about how many cells of each type switch in a given time interval, T .

Algorithm 3 Extract number of cells of each type

```

for line in cs-file do
  if Cell  $\in$  Node then
    if  $Cell_{switching} \in T$  then
       $cellname \leftarrow getCellName$ 
       $celltype \leftarrow getCellType(cellname)$ 
       $counter(celltype) \leftarrow counter(celltype) + 1$ 

```

2.4.1 Types of Cells Distribution for Test Vectors

Before conducting the above experiments, it was hypothesized how the distribution of cells would look.

Refer to figure 2.13, where the annotations describe three distinct parts of the waveform. The first part is drop leading up to before the deep spike. The second part is the spike. The third and last part is the time after the spike up until the voltage has recovered.

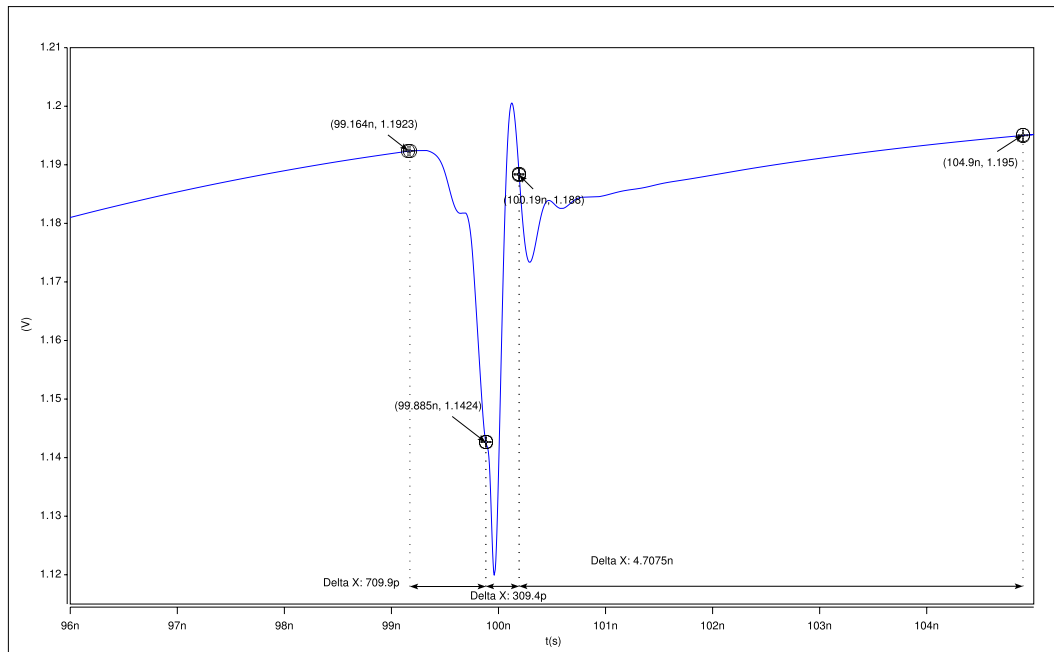


Figure 2.13: Voltage Drop with different components highlighted

This section's initial reasoning about simultaneous switching noise would lead us to believe that the short spike would consist of flip-flops switching. We can also argue that before the clock reaches its destination flip-flops, a number of clock buffers leading up to the flip-flops should consume power. This interval is the first interval in the figure. After the switching of the flip-flops, combinatorial logic in between registers switch to calculate a new circuit state. This would correspond to the final interval.

The results of this analysis confirm the theoretical reasoning, except in this case no combinatorial logic switched in the post-spike part. In table 2.3 the three voltage drop parts and their constituent cells are stated.

	Cells	Flip-Flops	Clock-buffers
Pre-spike	9	0	9
Spike	28	18	0
Post-spike	0	0	0

Table 2.3: Switching Cell Types in a Voltage Drop

The experiment was repeated for one of the worst-case drops, as shown in figure 2.14 and table 2.4.

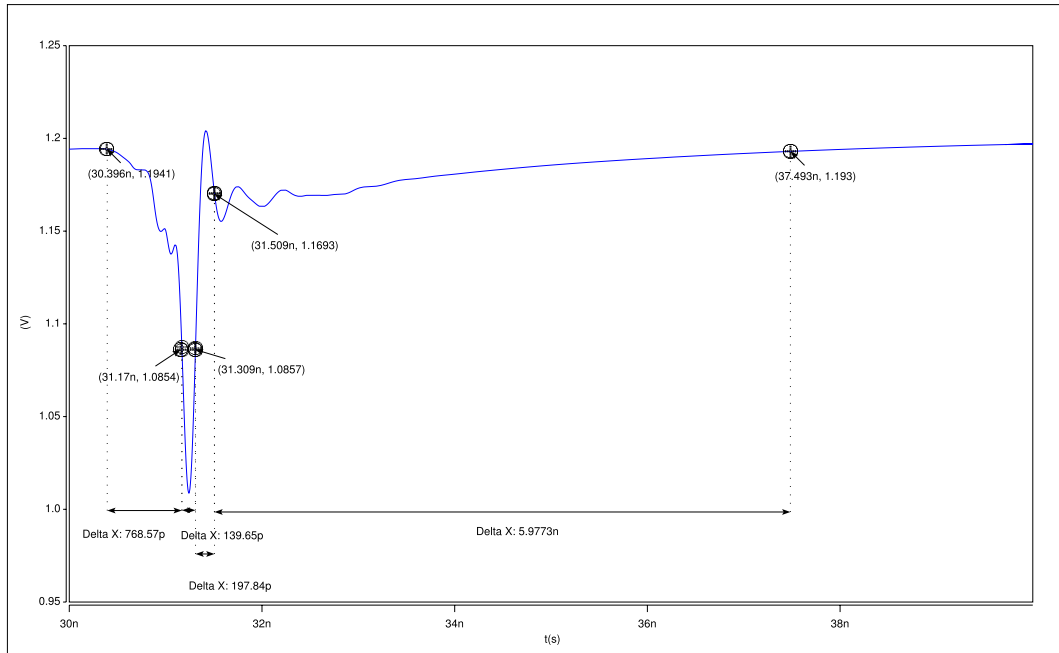


Figure 2.14: Voltage Drop with different components highlighted, worst-case drop

	Cells	Flip-Flops	Clock-buffers
Pre-spike	46	28	3
Spike	21	20	0
Post-spike	5	5	0
End	11	1	3

Table 2.4: Switching Cell Types in a worst-case Voltage Drop

Confirming the theory and gaining more understanding for the voltage drop waveforms we are dealing with is a goal in itself. However, an even more interesting result spawned from this methodology.

The experiment was performed for the reference test vector and for a test vector with an especially large voltage drop, table 2.5 contains the results.

	Cells	Flip-Flops
Reference vector	48	1
Large drop vector	148	78

Table 2.5: Switching Cell Types in a Voltage Drop

This indicates that a large number of simultaneously switching flip-flops localized to a single node on the chip gives rise to large voltage drops. A way to search for worst-case test vectors could be to extract the portion of a PrimeTime PX trace only made up of flip-flops. Pin-pointing the maximum flip-flop intensive clock cycle then lead to vectors causing large voltage drops. This method is not included in this thesis, but is rather left as an area of future work.

2.4.2 Alleviating Simultaneous Switching Noise Through Skew Spreading

Now, knowing the location of areas that can lead to large voltage drops, and knowing that a large part of the switching that causes the voltage drop is flip-flops, can this information be used to alleviate the simultaneous switching noise and thus the voltage drop?

In fact, a fairly common technique for dealing with simultaneous switching noise is to intentionally skew the arrival of clocks to nearby flip-flops, so called *useful-skew* or *skew-spreading*. Many research papers have been written on the subject, ([16], [17] and [18] are good examples) and some EDA tools have features to help the backend engineer create a clock skew scheme that lowers IR drop by introducing non-zero clock skew.

The skew-spreading approach was applied to this model in order to evaluate how a successful skew-spreading scheme could affect the voltage drop. Since the cell switching information file contains switching times for each switching cell, the above approach of singling out switching registers can be modified to emulate skew-spreading.

Algorithm 3 is then modified as Algorithm 4.

Algorithm 4 Extract number of cells of each type

```

for line in cs-file do
  if Cell  $\in$  Node then
    if  $Cell_{switching} \in T$  then
       $cellname \leftarrow getCellName$ 
       $celltype \leftarrow getCellType(cellname)$ 
      if  $celltype == register$  then
        if  $cellcount$  is even then
           $Cell_{switching} \leftarrow Cell_{switching} + skewvalue$ 
           $cellcount \leftarrow cellcount + 1$ 

```

This algorithm skews every other register in the time-span/area of interest by a specified amount through adding or subtracting time from the transition time.

Note that this method does not ensure timing-correct functionality of the design. Neither does it ensure any kind of optimal skewing strategy. It is merely a first naive method applied to a model of the design to show possible benefits.

However, even a first-guess value (kept within reasonable values of skew spreading) yielded surprisingly good results, which are presented in the results section.

Chapter 3

Results

In the work of this master's thesis, over one hundred SPICE simulations have been performed, with different test vectors used to create the switching patterns loading the power grid model. The test vectors were found using the different methods presented in chapter 2. This chapter presents the results of these simulations, focusing on the maximum voltage drops of each test vector extraction methodology.

3.1 Full-chip Maximum Power

In the work preceding this master's thesis, the time window identified through finding the full-chip maximum power calculated over a 5 ns time interval was used as a worst-case test vector.

While this vector was previously considered to exhibit a large drop, its drop is dwarfed already by vectors found by other naive approaches. For example, the total power analysis can be made more detailed by calculating the maximum power over a power trace of a finer time resolution.

While the reference vector was found by considering a 5 ns resolution power trace, the work in this thesis has also considered resolutions of 1 ns, 0.1 ns, 0.01 ns and 0.001 ns. As it was mentioned in section 2.1.1, increased resolution leads to larger file sizes. However, when considering full-chip power, the leaf option was deemed unnecessary so finer time resolutions could be used.

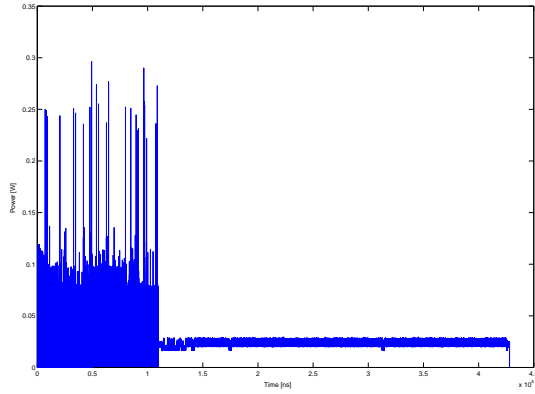
In table 3.1, the maximum voltage drop of vectors found using the total power analysis approach is summarized along with the times at which the time window for the test vectors are centered. The maximum voltage drop has gone from 8% of the nominal voltage supply to 13%. This represents a 62% increase in maximum voltage drop estimation only by analyzing power at a marginally finer time resolution.

Power Trace Resolution (ns)	V_{min}	At Time (ns)	Worst node
5	1.1034	49228	203
1	1.0430	310481	64
0.1	1.0430	356241	64
0.01	1.0428	382481	64
0.001	1.0422	190641	64

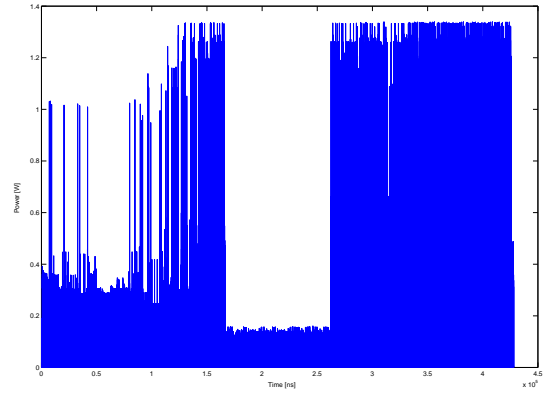
Table 3.1: Total Power Approach, time resolution Vs maximum voltage drop

The result is easy to reach, yet meaningful for vector-based power integrity analysis: selecting a proper resolution for power calculations is crucial when using a total-power based approach.

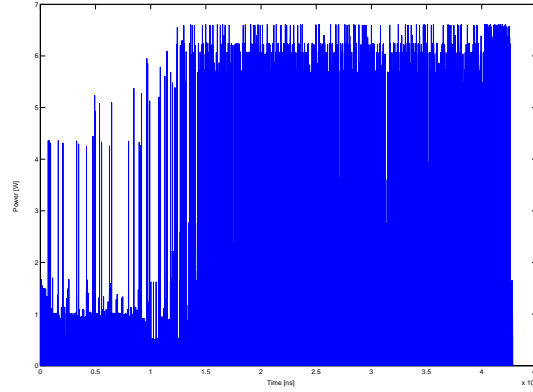
The importance of time-resolution can also be seen in the following figures (3.1), which plot the full-chip power over the entire time of the VCD trace, in three different time resolutions. Annotated in the figures are voltage drops for 42 test vectors, which were taken from the part of the figures where they appear. Their location in the y-axis represent their relative magnitude and shows us that large drops tend to be located near the end of the trace. Note that in the 5 ns resolution, the activity seen in the later part of the trace is negligible, while all the large drops actually occur here.



(a) 5 ns resolution



(b) 1 ns resolution



(c) 0.1 ns resolution

Figure 3.1: Full-chip power over time

From table 3.1, the time of the 5 ns drop is in the beginning of the trace, which is consistent with figure 3.1 - test vectors found using this time resolution will always be in the first part of the trace, even though the largest drop occur in the last part.

In order to compare the usefulness of the full-chip maximum power approach, the voltage drop simulation results of randomly chosen test vectors are presented in table 3.2. This experiment also considers the nature of the power trace as seen by different time resolutions, where the second half of the trace is more active in finer resolutions. The time of the power trace was split in half and five random time instances were chosen from the first half, and five from the second half. We can see the tendency of larger drops in the later part of the trace. Also, all voltage drops are of about the same magnitude as or larger than the reference test vector, which proves that it was

not a good way to select a test vector if peak voltage drop was the objective. Another thing to note is that some random vectors chosen from the second half of the trace had as large voltage drop as those found by finer time resolutions of the full-chip maximum power approach (see table 3.1) - some better method must be used to find vectors that are worse than those selected randomly.

Test Vector	V_{min}	At Time (ns)	Worst node
First half, 1	1.1097	96091	203
First half, 2	1.1090	23452	203
First half, 3	1.1090	61409	203
First half, 4	1.1037	49199	203
First half, 5	1.1028	90147	203
Second half, 1	1.0435	300490	64
Second half, 2	1.0430	362770	64
Second half, 3	1.0522	342730	64
Second half, 4	1.1018	409920	203
Second half, 5	1.0443	388500	64

Table 3.2: Voltage drop of randomly selected test vectors

3.2 Frequency and dP/dt based approaches

The dP/dt approach was motivated by the L dI/dt part of voltage drops. By finding the fastest power transitions, this could translate to fast current transitions which would lead to large voltage drops. These vectors were found at the same time as full-chip maximum power vectors, as explained in 2.1.1.

The largest dP/dt for different time resolutions along with voltage drops are presented in table 3.3, and do not differ significantly from the full-chip maximum power approach.

Time Resolution	V_{min}	At Time (ns)	Worst node
5 ns	1.1051	79933	203
1 ns	1.0429	310482	64
0.1 ns	1.0459	407772	64
0.01 ns	1.0458	191711	64
0.001 ns	1.1019	117931	203

Table 3.3: Voltage drop of dP/dt-based vectors

The FFT based approach, described in section 2.3.2, could be used in a number of different ways. The frequencies of interest can be changed to focus on different resonant peaks. The time window size and range of tolerance for frequencies close to the peak resonant frequencies can be varied.

A few combinations of these parameters were used to find test vectors which were then simulated to find the worst voltage drop. The results can be seen in table 3.4. Again, no voltage drops larger than those found through random selection were identified.

Number	V_{min}	At Time (ns)	Worst node
1	1.1053	109500	203
2	1.1063	131100	203
3	1.0542	138191	64
4	1.0440	322991	64
5	1.0449	370521	64

Table 3.4: Voltage drop of FFT-based vectors

With the wavelet transform method of identifying time windows of large activity in certain frequency ranges, some large-drop vectors were identified, but not enough to consider the method fully successful. In fact, the results are similar to random vector selection, as can be seen in table 3.5. Note the very large drop of number 13. While this is larger than any earlier found voltage drops, it was found rather arbitrarily with the wavelet method, and does not give too much credit to the methodology itself since there are no other similar drops among these experiments.

Number	V_{min}	At Time (ns)	Worst node
1	1.1101	8166	203
2	1.0756	7126	77
3	1.1111	5363	203
4	1.1070	7980	203
5	1.1062	47676	203
6	1.0500	370517	64
7	1.0503	308812	64
8	1.0670	53625	77
9	1.0566	96526	77
10	1.0432	401415	64
11	1.1061	55475	203
12	1.0410	312582	64
13	1.0085	425131	62
14	1.0509	414483	64
15	1.0409	340843	64
16	1.1048	38051	203
17	1.0422	356241	64

Table 3.5: Voltage drop of wavelet-based vectors

3.3 Localized Voltage drops: Finding the Where and When

By far the most successful method to find vectors creating large voltage drops, was the method of localized currents, as described in section 2.2. A method was presented that given a node known to cause large voltage drops, the worst timespan in a VCD trace was found. Also, a method for answering the “where” question was developed. Here, the worst node is assumed to be node 62.

Different combinations of localization around the node of interest (node 62) was tried. The three localization levels were: 1) The node of interest, 2) The node’s immediately adjacent nodes (refer to figure 2.2) and 3) The area of the CPU, where the node was located. More fine-grained levels of localization could, if time permitted, be evaluated to fine-tune the technique.

Using the three levels of localization separately, without the overlapping method, yielded the results in table 3.6.

Locality	V_{min}	Worst node
CPU	1.0441	64
Node	1.0251	62
Adj. Nodes	1.0546	64

Table 3.6: Localized power results

While the $V_{min} = 1.0251$ is the largest drop seen so far, the combined efforts using the overlapping method gives even larger drops. The three localizations can be combined two-and-two in three ways: CPU-node, CPU-adj and adj-node. The voltage drop results are shown in 3.7.

Locality	V_{min}	Worst node
CPU-Node	1.0438	64
CPU-Adj.	1.0086	62
Adj.-Node	1.1015	203

Table 3.7: Localized power results, overlapping

The combination of the CPU and the adjacent nodes dissipating large amounts of power gives a very large voltage drop in the node enclosed by the adjacent nodes. This result is significant, and more experiments were conducted to verify generality.

In table 3.8, seven more test vectors were selected in the same way as the CPU-adj above. The total ordering of significant overlapping time windows (see 2.2.1) was used to select the ones after the top-most significant. The fact that all seven of the vectors in the experiment gave such a large voltage drop confirms with much confidence that the method is indeed useful, at least for the model and chip design used. In fact, the voltage drops observed using this method are twice as larger as the reference test vector, a significant result indeed.

No.	V_{min}	Worst node
1	1.0086	62
2	1.0088	62
3	1.0087	62
4	1.0087	62
5	1.0088	62
6	1.0086	62
7	1.0084	62

Table 3.8: CPU and adjacent nodes overlapping time windows, Voltage Drops

3.3.1 Choosing the Worst Node

In the results presented above, a “worst node” was assumed. This node was found through performing voltage drop simulation on a lot of different test vectors and noting which node most frequently had the largest voltage drop.

Using the methods developed in 2.2.2, other nodes are pointed out as being the worst. This section will document some experiments done while focusing on these nodes.

In figure 2.6(a), node number 76 was shown to have the largest power dissipation over a 1 ns time interval. This should make it interesting from a worst-case drop perspective. Switching information was generated for the time window enclosing the 1 ns time interval where the node power peaked and a simulation was performed. The results showed that the minimum voltage observed was $V_{min} = 1.0313$ V. This value can be compared to $V_{min} = 1.0251$ V from table 3.6. Interestingly, this drop occurred in node 63, and not 76 where the maximum power occurred.

Studying the current of the two nodes (63 and 76) reveals that node 63 had in fact a much higher peak current than 76. It also had the maximum current over a 1 ns time interval. This node should thus have been found to be the maximum power consuming node.

A possible explanation of this phenomenon is that the 1 ns time interval of PrimeTime PX coincided with the peak in node 63 in such a way that the whole peak was not enclosed in the time interval. This means a finer time resolution must be used to detect power peaks. The concept of choosing a proper time resolution for peak-power analysis was discussed in section 3.1, and these results further motivate that consideration.

Still, a minimum observed voltage drop of $V_{min} = 1.0313$ V is better than many other

methods of finding a worst-case test vector presented here. It is a simple method that could be useful in identifying the worst node and at the same time the worst time window.

Another interesting node from figures 2.6 and 2.7 is node 63. It is a central point in figure 2.7(b), which indicates a large node/adjacent node power combination. Simulation results yielded $V_{min} = 1.05$ V in node 64 (which is one its adjacent nodes). This results can be partly explained by figure 2.6(b) and 2.7(b). In 2.6(b), 64 has a larger max than 63, and from 2.7(b), 63 has large adjacent nodes. We would expect this node/time to show a large drop in 64. The voltage drop is rather small, however.

Moving to power traces in 0.1 ns time resolution and the figures 2.8, 2.9 and 2.10, nodes 62 and 64 are the most prominent. The current profile was generated for the time when node 64 peaks, and was compared to the current profile for the peak of node 62. Surprisingly, the time window identified by node 62 had the larger current than the time window for node 64. The result is similar to the above paragraph about node 63 and 64. The issue could possibly be that an even finer time resolution is required to get a one-to-one mapping between node power dissipation and node current maximum.

The currents discussed above are plotted in figures 3.2 for the time of the peak of node 62, and 3.3 for the time of the peak of node 64. The above plots of the figures show current consumption in node 62 while the plots below show the same for node 64. From figure 2.8, we would expect figure 3.3 to show a larger maximum current than figure 3.2.

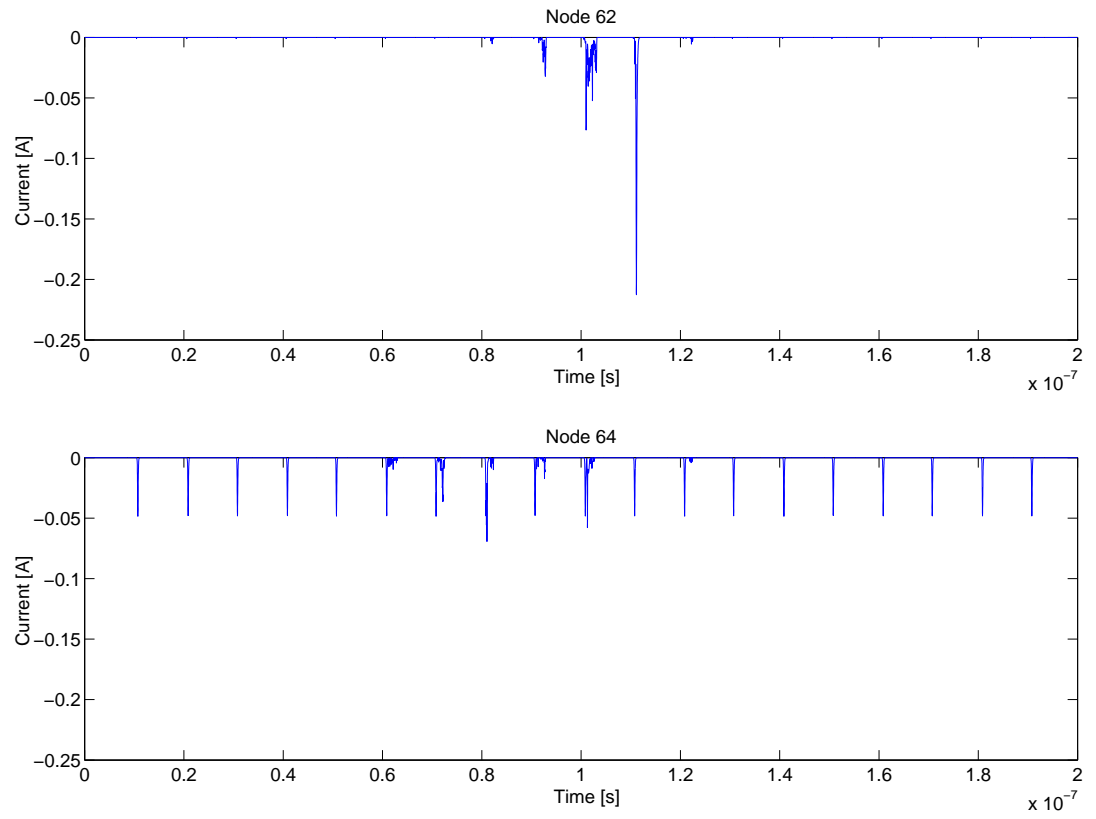


Figure 3.2: Current signatures for nodes 62 and 64. Time window at node 62 power peak.

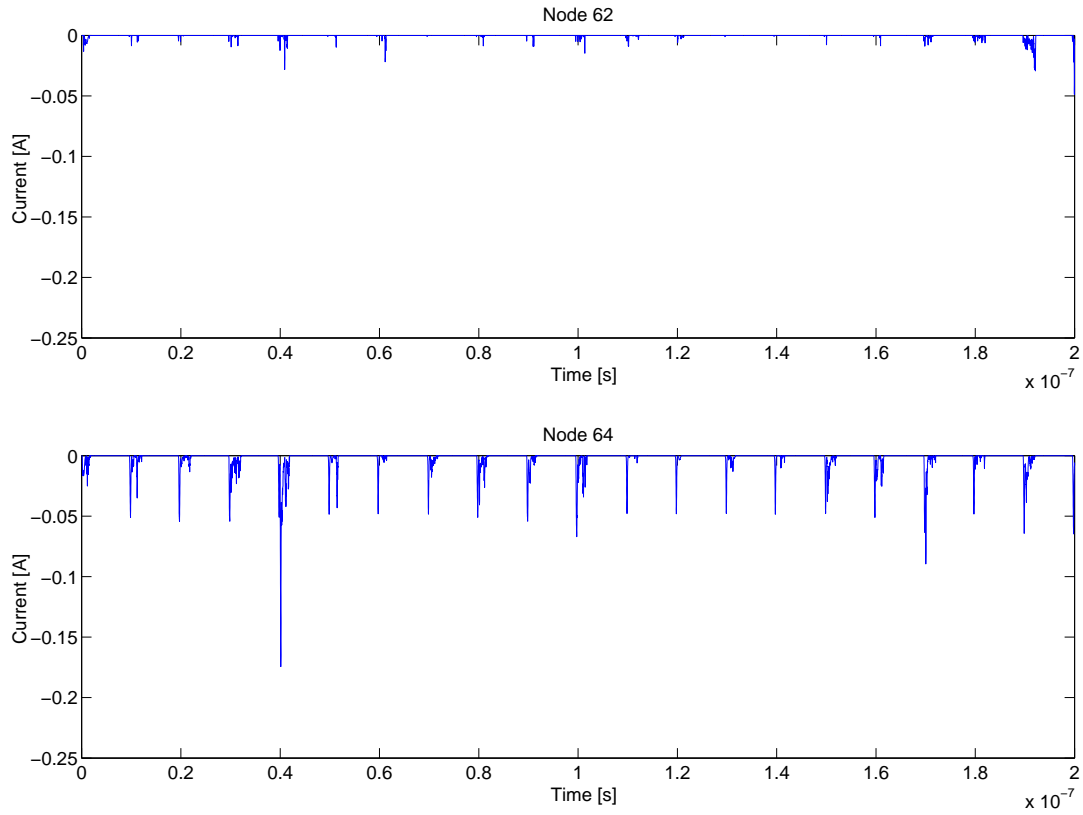


Figure 3.3: Current signatures for nodes 62 and 64. Time window at node 64 power peak.

The resulting voltage drop for the test vectors found by taking the peak power time of each of interesting nodes from the above experiments are presented in table 3.9. Also, the node where each maximum voltage drop occurred is presented. For two cases (76 and 89), the largest voltage drop did not occur in the predicted node.

Vector	V_{min}	V_{min} node
62	1.0251	62
64	1.0544	64
76	1.0313	62
89	1.0280	62

Table 3.9: Test vectors found by considering each node's power maximum

So far, the results from these power analyses give no particular preference to node 62. It is, however, the top node in 0.1 ns time resolution considering adjacent nodes, as shown in figure 2.9, but the choice of this particular node remains quite arbitrary. Yet it is this node that is the worst node in all test vectors exhibiting a large drop, and focusing on its power dissipation peak yields the largest voltage drops. This makes the methodology developed an iterative process rather than an ability to immediately foresee where and when the largest drop will occur.

3.4 Voltage Drop Characteristics and Worst-case

Test vectors have been found that have larger maximum voltage drop than the reference vectors. But how do other characteristics of the voltage drop curves compare? In figure 3.4, annotated voltage curves are plotted for the reference vector (3.4(a)) and a vector with a higher maximum voltage drop (3.4(b)).

One interesting characteristic is the duration of the voltage drop. A short pulse might have other implications for timing than a long drop. Looking at the “Delta X” annotation for both plots, we can see that the duration of the large drop is longer for the reference case (a) than for the worst vector case (b). Furthermore, another characteristic that can be read from these plots is the time duration where the worst-case drop in (b) has fallen below the level of the worst-case for the reference vector(a). From the figure, this is approximately two thirds of the voltage drop duration.

Near the right edge of the figures, an annotation on 1.1968 V (close to nominal 1.2 V) tells us something of the time to recovery of the drops. Measuring from the lowest V in the figure, for (a), this time is 12 ns while for (b) it is 8 ns.

In other words, while the drop in figure 3.4(b) has the largest maximum, figure 3.4(a) has longer time durations. These differing characteristics might have different implications for timing, which will be discussed later on.

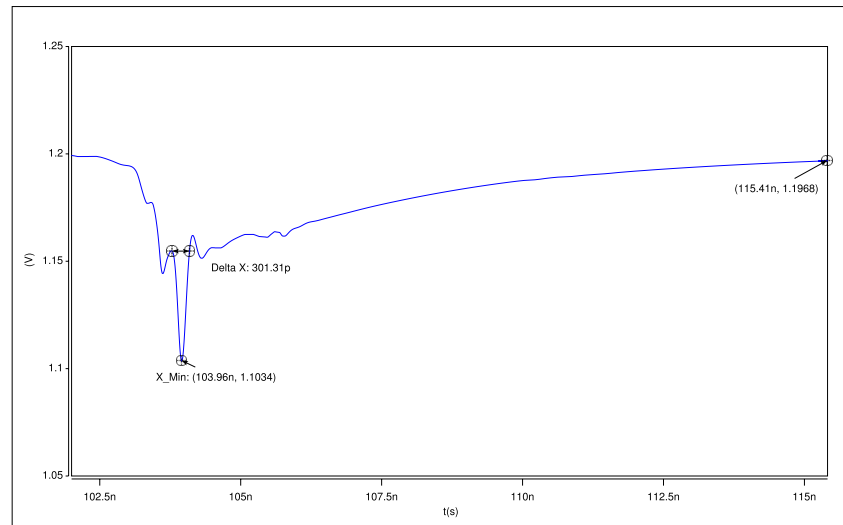
An interesting question is what causes these differences in characteristics. Going back to the current profiles which give rise to the voltage drops hints to a possible answer. Table 3.10 specifies the maximum current of the test vectors in different levels of locality: 1) full-chip current, 2) superposition of adjacent nodes of worst-case voltage drop node and the node itself and 3) worst-case voltage drop node itself only.

Locality	Ref. vector	Worst vector
Full-chip	1.9397	1.9211
Adj. nodes	0.0774	0.2447
Node	0.0724	0.2132

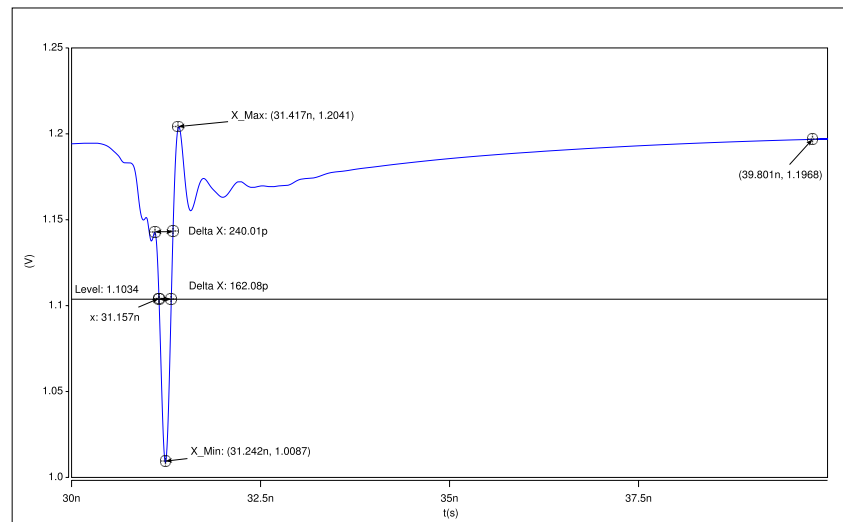
Table 3.10: Current profile max of two test vectors [A]

The full-chip current is larger for the reference vector. This is to be expected, since this vector was found by taking the top power consuming clock cycle for a full-chip power trace. For the other levels of locality, the worst vector has the larger max current. Another thing to note is the difference between the two last localities. For the worst vector, this means that its adjacent nodes also have a large current surge. This is not the case for the reference vector.

The above numbers, although perhaps not representative for a general truth, indicate that large chip-wide current consumption give slow recovery times while large currents in a few local nodes give quick voltage drops of larger magnitude.



(a) Reference test vector



(b) Worst test vector

Figure 3.4: Voltage Drop for worst node

In these plots we can also see the increased inductive behavior of the worst case vector. This case (figure 3.4(b)) shows a large inductive overshoot followed by a oscillatory behavior which can not be seen in the reference case.

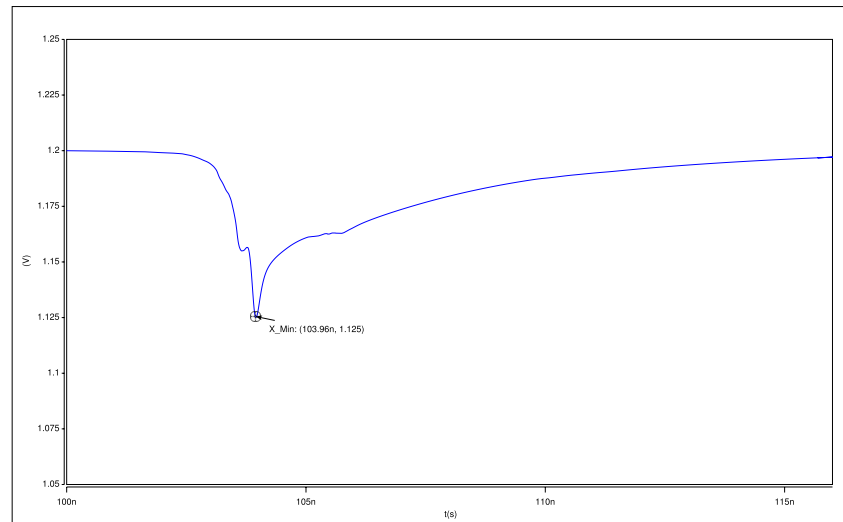
3.4.1 The effect of on-chip inductance on voltage drop

In figure 3.5, we have the same two test cases but this time with on-chip inductance removed. The overshoot and oscillatory behavior has disappeared, and the maximum

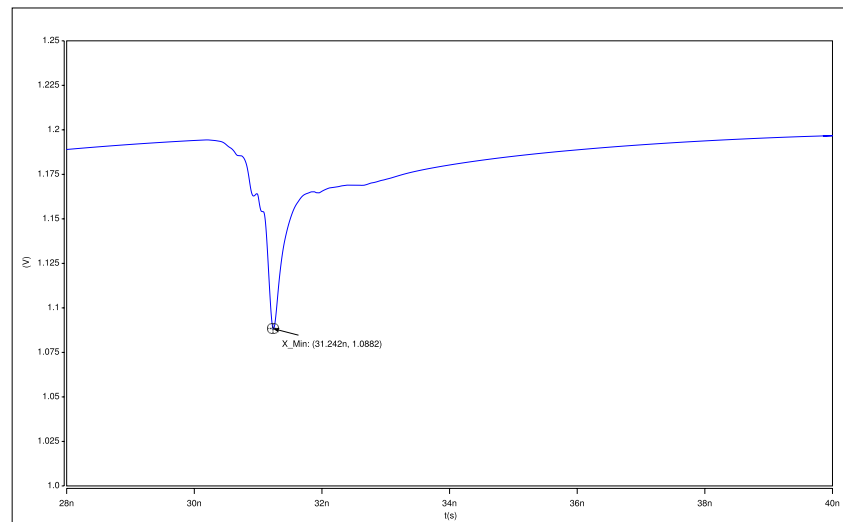
voltage drop is much lower.

For the reference case (3.5(b)), the maximum drop has also decreased. However, since there was not much inductive behavior *with* inductance, the shape of the drop has not changed much.

Quantifying the impact of the on-chip inductance for the two cases reveals the following. For the reference vector, the addition of on-chip inductance caused the voltage drop to increase from 75 mV to 96 mV, or about 29%. For the worst-case vector, on-chip inductance caused the voltage drop to increase from 112 mV to 191 mV, or 71%. This result shows a huge increase in estimation of the importance of on-chip inductance. In [1], it was concluded (with the same model used in this thesis) that inclusion of on-chip inductance can increase the voltage drop by over 50%. By simply applying a different test vector to the very same model, it has now been shown that on-chip inductance can increase the voltage drop by 70%.



(a) Reference test vector



(b) Worst test vector

Figure 3.5: Voltage drop for worst node, without on-chip inductance

3.4.2 Chip-wide distribution of voltage drop

So far, we have only considered the node experiencing the largest drop on the chip, for each test case. Another interesting issue is how *many* nodes on the chip experience large voltage drops, and where the large drops are located.

Comparing again the two test cases from above, reference and worst-case, the histograms in figure 3.6 tells us how many nodes on the chip experience voltage drops

of various magnitudes. The shape of the histograms look similar, with the worst-case vector leaning more towards higher voltage drops, but the important thing to note are the few high-drop nodes in 3.6(b). Two nodes experience drops larger than 10%. These nodes are 62 and 64.

The geometric location on chip of the nodes presented in the histograms can be seen in figure 3.7. In (b), we see the two nodes above 10% (62 and 64) having a larger drop than any other node on the chip. Their locations coincide with the CPU, compare with figure 1.1. In (a), there is a much more evenly distributed drop. The top is near the clock tree root. The appearance of these 3D chip plots can be related to the way the test vectors were found. The reference test vector was found using the clock cycle of highest chip-wide power dissipation. The fairly large chip-wide voltage drop of figure 3.7 can intuitively be linked to this behavior. The worst-case was found by singling out one node and finding the time window where the node and its neighbors had high power dissipation. This corresponds to the more localized voltage drop scenario of 3.7(a).

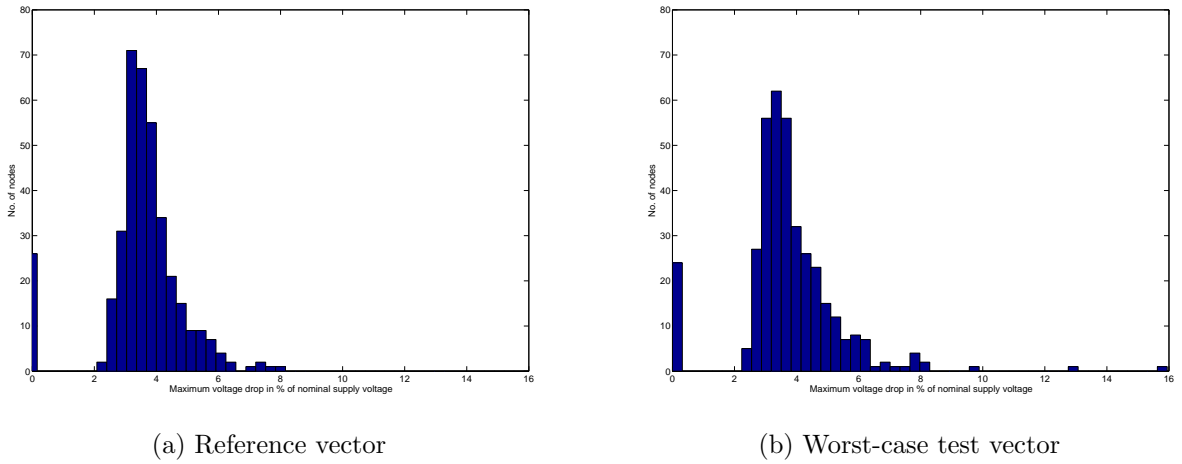


Figure 3.6: Voltage drop magnitude distribution

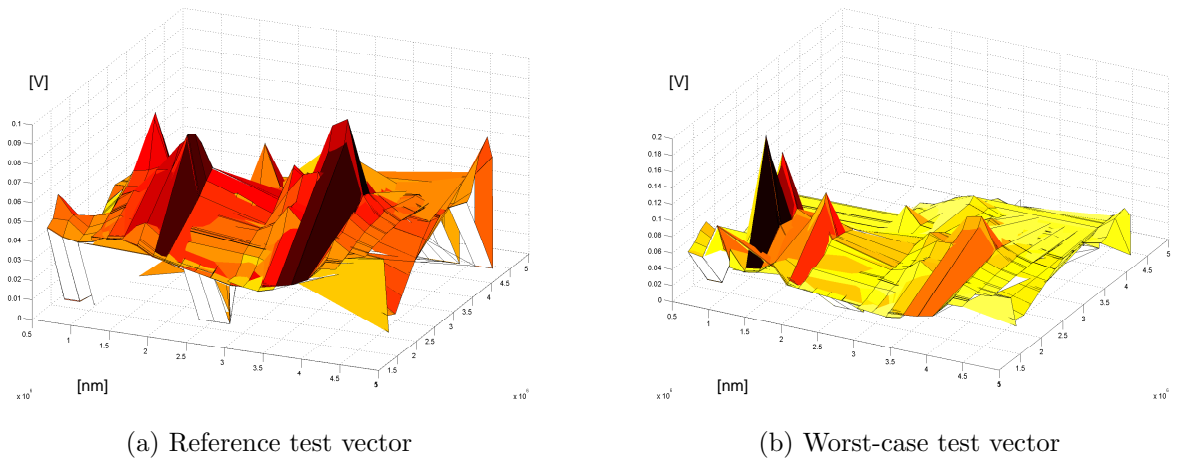


Figure 3.7: Voltage drop by node location

3.5 On-chip Inductance and Bond-wire Inductance

For chips with bond-wire packaging, the inductance of the bond-wires is typically much larger than the inductance associated with the on-chip power grid, on-chip inductance. Bond-wire inductance of a single pin is normally a few nH, while the on-chip inductance of a power grid segment connecting two nodes (in this power grid model) is on the order of 0.001 nH - 0.1 nH.

It has thus commonly been assumed that the bond wire inductance has a larger impact on the integrity of the power grid, and the supply voltage drop. The goal of earlier papers [1] has been to show that the on-chip inductance is non-negligible.

In this thesis, although it was not the primary goal, experiments have been conducted on the impact of on-chip inductance vs. the impact of bond-wire inductance which paint a different picture. According to these experiments, not only are the relative influences of the two sources of inductance comparable across the chip, but *the on-chip inductance is by far the larger contributor for the worst-case voltage drop*. It will be discussed later how much faith can be put in these results given model consideration, but first the simulations results will be presented.

In this experiment, four different cases are considered:

- 1) Bond L, On-chip L. This is the case on which most of the rest of the thesis builds.
- 2) No Bond-L, No On-chip L. Neither bond-wire nor on-chip inductance is modeled. Acts as a reference when the L-components contributions are measured.

- 3) Bond L, No On-chip L. Only the bond-wire inductance is modeled.
- 4) No Bond L, On-Chip L. Only the on-chip inductance is modeled.

In figure 3.8, simulation results for two randomly chosen nodes are shown for the four cases. In figure 3.8(a), bond-wire inductance has a larger impact on voltage drop than does the on-chip inductance. For figure 3.8(b) however, it is the on-chip inductance that contributes the most to the voltage drop. The on-chip inductance adds 28.6% to the voltage drop without inductance. Bond-wire inductance adds 0.2% to the voltage drop. In (a), the randomly chosen node is in the M3 layer while (b) is in M1 which directly supplies the transistors. This layer dependence seems to be general for the design, as is summarized in table 3.11. Here, the number of nodes that have Bond L and On-Chip L, respectively, as their main voltage drop contributors is stated for each metal layer of the model.

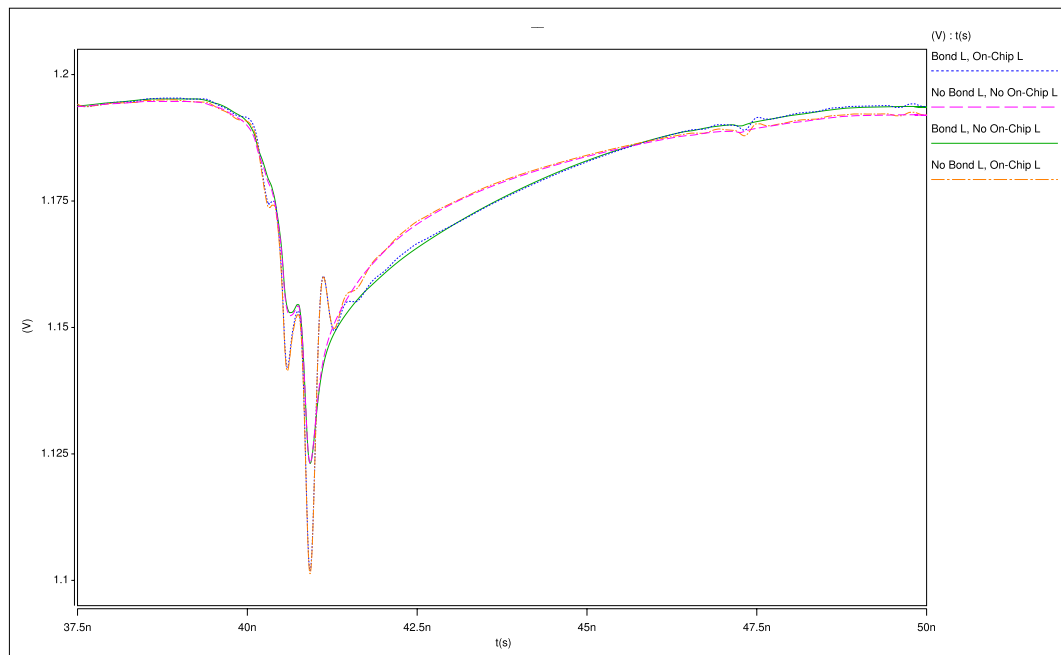
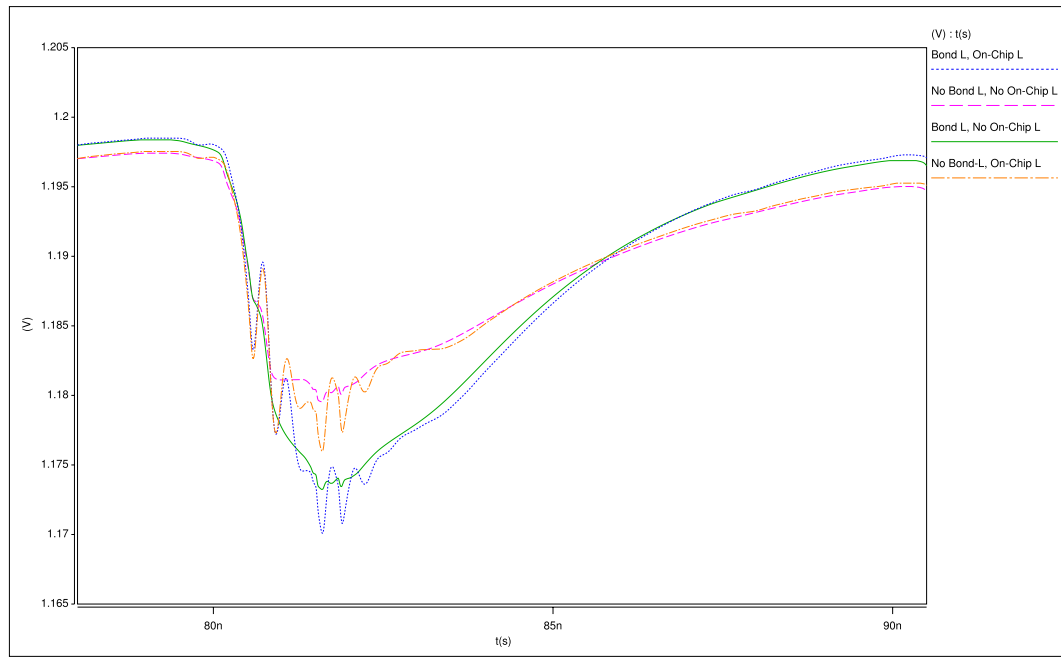
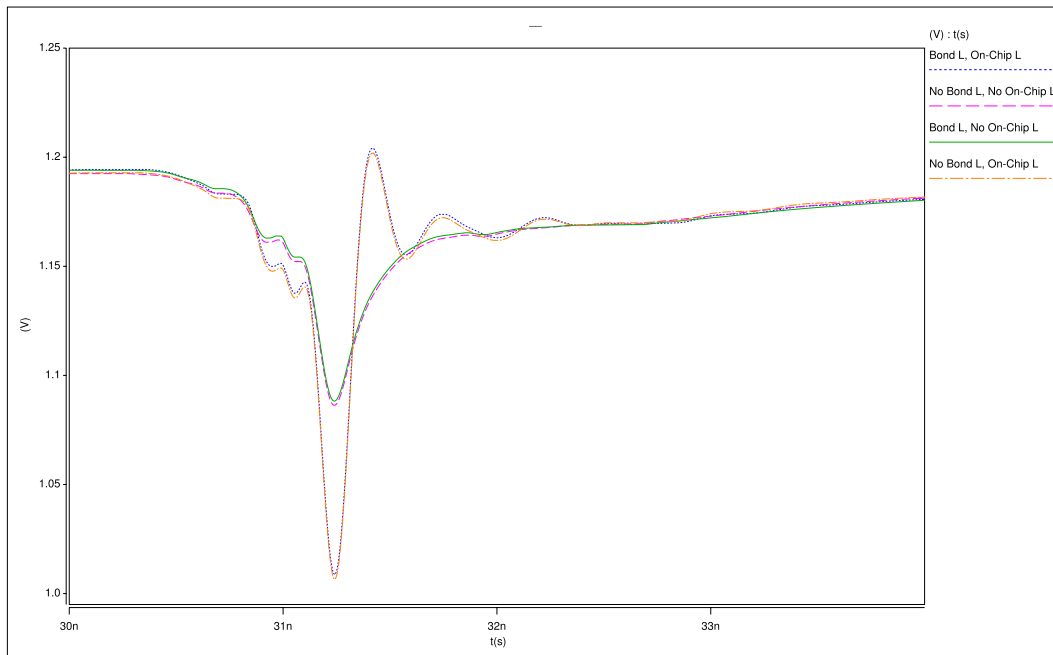


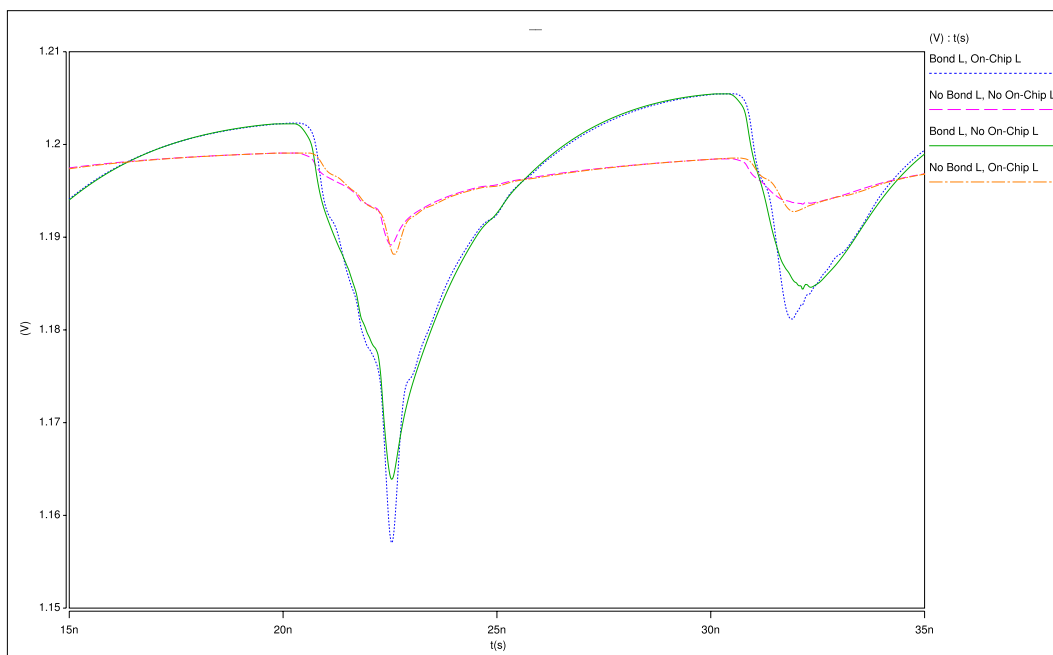
Figure 3.8: Bond-wire vs on-chip inductance for two randomly selected nodes

[H]		
Metal Layer	No. of nodes w/ Bond-L dominating	No. of nodes w/ On-Chip L dominating
Full-chip	689	882
M1	103	271
M2	350	395
M3	236	136

Table 3.11: Bond-L vs. On-Chip L, number of dominating nodes per metal layer



(a) Worst-case drop node



(b) Supply pin node

Figure 3.9: Bond-wire vs on-chip inductance for two nodes

In figure 3.9(a), the same comparison is repeated for the node with the largest voltage drop. Here, the contribution of bond-wire inductance is dwarfed by the on-chip inductance. Figure 3.9(b) shows a node sitting at a supply pin. It shows the expected result that bond-wire inductance is more significant here. The worst-node case is particularly interesting. It shows us that a large part of the worst voltage drop identified on the chip is an $L \, dI/dt$ drop caused by the on-chip inductance. In fact, the bond-wire inductance seems to slightly decrease the drop. Putting the result into numbers tells us that case 4 has a 70% larger voltage drop than case 2, while case 3 has a 1% smaller voltage drop than case two. Comparing this result with that of figure 3.8(b) shows us that on-chip inductance has a much larger relative impact in the worst case than for a randomly chosen node. The values of extracted inductances close to these nodes are similar so the reason for the difference must be found elsewhere.

3.5.1 Frequency Components and Inductance Contributions

Bond-wire inductance generally result in chip-wide voltage fluctuations while on-chip inductance usually cause spatially localized variations. The chip-wide disturbances translate to a low frequency voltage drop curve while localized behavior usually consist of higher frequencies.

In order to test this hypothesis in the context of the model used in this thesis, an experiment was conducted on the influence of bond-wire vs on-chip inductance on different frequency components of a voltage drop waveform.

Using FFT based techniques, the voltage drop waveform for each node on the chip was filtered to extract the low-frequency components and the high-frequency components. The division into frequency components is illustrated in figure 3.10, where the worst-case drop node is plotted for bond-wire L only (a) and on-chip L only (b).

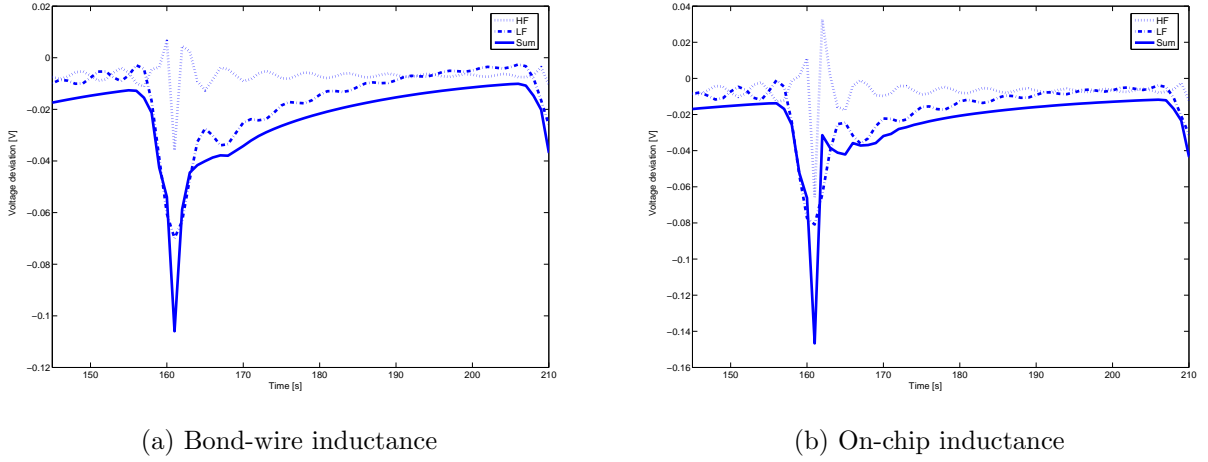


Figure 3.10: Frequency break-down of Bond-wire vs On-chip inductance

For each node in both of the frequency parts, the maximum voltage drop was calculated. This extraction and calculation was performed for two cases: 1) No Bond-wire L, With On-chip L and 2) With Bond-wire L but no On-chip L.

From this data, the number of nodes where the on-chip or the bond-wire inductance caused the largest drop was calculated, both for high and low frequencies.

From the results found in table 3.12, we can see that low frequency part of the voltage drop is mostly caused by bond-wire inductance, while the high frequency part is mostly caused by on-chip inductance. These results confirm the hypothesis stipulated above.

	Bond-wire L	On-Chip L
HF	271	1221
LF	1302	190

Table 3.12: Bond-L vs. On-Chip L, number of dominating nodes per frequency component

3.6 Results of Skew-Spreading on Voltage Drop

In section 2.4, the types of switching cells constituting a voltage drop were extracted, and a method for emulating a skew-spreading of the clock signal to some registers

was presented.

In figure 3.11, the results of voltage drop simulations for two cases of the same test vector and node are presented. Here we can see the effect of the simple skew-spreading that was introduced (dotted line). The voltage drop has decreased from 191 mV to 148 mV, or 22%.

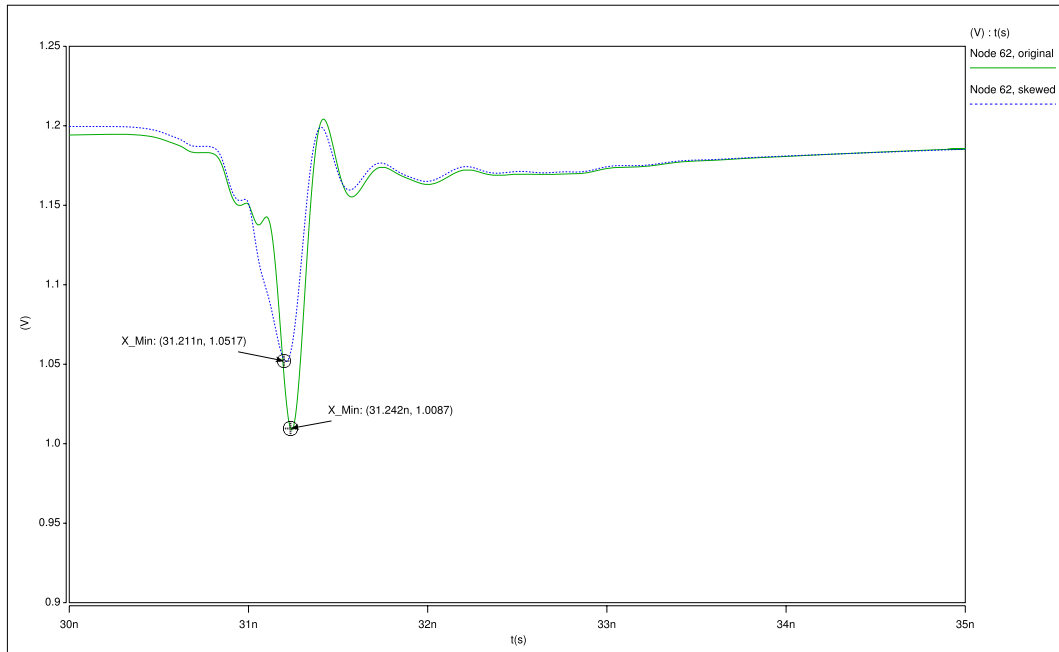


Figure 3.11: Voltage drop for worst-case node, with and without skew-spreading

Note in the figure that the peak drop has decreased while the shape of the drop has widened. This is a natural consequence of skew-spreading since all we do is moving switching activity, rather than removing it altogether. The same tendency can be seen in the current profiles for the voltage drops presented. They are shown in figure 3.12, where the widened but lowered current peak in (b) corresponds well to the voltage drop figure.

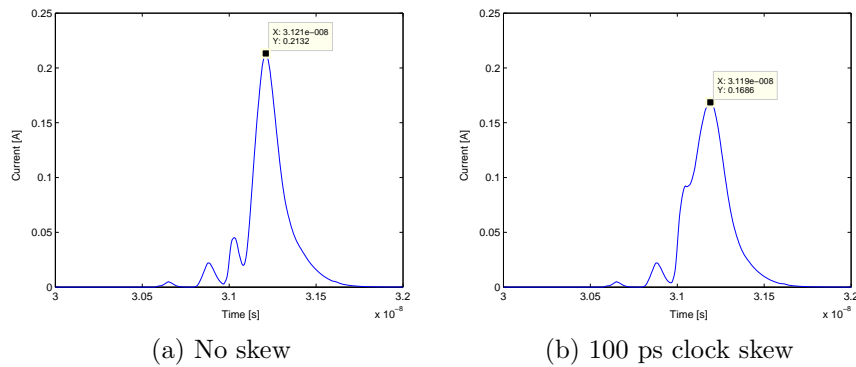


Figure 3.12: Effect of skew-spreading on current profiles

These results were found through a naive first-guess method of skew-spreading. The arrival of every other switching register in the worst-case node was moved 100 ps earlier in time. More elaborate methods for clock-skewing could possibly be used to decrease the voltage drop further. Some research have been focusing on the problem, notably [16], [17] and [18].

The main point to be made here is rather about the possibility of reduced voltage drop through skew-spreading, and the realization that only a few nodes suffer the very large drops (see 3.4.2). Consequently, the clock-skewing methodology could be performed on each node showing large voltage drops, and thus the entire chip dynamic voltage drop could be reduced significantly.

Chapter 4

Discussion

Test vectors leading to large voltage drops have been found, and a methodology of finding them has been developed. This section discusses the problems involved in drawing definite conclusions on the results and suggests future work that could further the process of understanding power integrity.

4.1 Weaknesses in Model and Methodology

While the developed methodology for identifying worst-case input test vectors from a large simulation trace proved successful in this context, it is too early to say that it will be as successful for other models or for other designs. Since no similar model for another design existed at the time that the experiments were conducted, it cannot be said with certainty that the methods developed here would show the same result for another design with a different power grid topology.

Within the same design, more test cases should also be analyzed. Perhaps the developed methodology proves to be the best for identifying worst-cases for those test-cases too, but other test-cases could show even larger drops than those found here. It would be good to analyze run traces of real in-field use-case data to establish if the used simulation trace is representative in terms of voltage drop tendencies.

It has been argued that the model used in this thesis has weaknesses that could introduce too much uncertainty in the result, especially considering quantitative results. For instance, in this thesis the ground net has been neglected. Depending on the power/ground grid topology, this could lead to overestimating or underestimating the impact of on-chip inductance. Currents flowing in opposite directions of wires with large mutual inductance would cause smaller loop inductance, according to [19]. The paired grid-approach to power planning places conductors with opposite current (vdd/gnd) close to each other, which should increase this effect.

Modeling of decoupling capacitance is another consideration that perhaps should be given more attention. Now, the total capacitance of the chip is spread out evenly to each switching node, so that each node that is active in a given test vector has the same capacitance. In reality, this distribution can be different depending on the density of gates in different areas of the chip. A more accurate capacitance modeling could lead to simulations showing less voltage drop.

4.2 Timing implications

Much of the problems related to voltage drops are due to the timing problems they can lead to. Since gate delay is proportional to supply voltage, too much noise in the power supply can lead to uncertainties in the final performance of the circuit.

In order to be able to say something about the timing problems that arise, a brief explanation of the problems of setup violations and hold violations will be given (see [12] for a thorough explanation).

Setup and hold violations both relate to two consecutive flip-flops, one which is the launching flip-flop and one which is the capturing flip-flop, separated by some datapath of combinatorial logic (figure 4.1). Flip-flops in a given process have specification on hold-time and setup-time. The setup-time states how long before the arrival of the clock that the input data to the flip-flop must be stable. Hold-time states how long after the clock signal arrives the input data must remain stable.

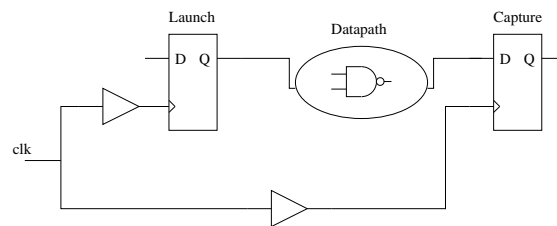


Figure 4.1: Datapath with launch and capture flip-flops

A hold-time failure can occur if the data-path between the flip-flops is short. A signal can get launched from the first flip-flop, rushing through the data-path and propagate through the two flip-flops in only one clock edge.

A setup-time failure can occur if the combinatorial logic in the datapath is too long and slow, which prevents the signal propagating through the logic to meet it specified

setup-time, when it should have settled at the input of the capturing flip-flop.

In this thesis, the min (or fast) corner of PVT has been used. That means the simulation results apply for the case of high voltage and low temperature. For this reason, the hold-time failures are the most interesting, because setup-time failures are unlikely to occur with the fast propagation of the min corner. Setup-timing problems would be more likely to occur if we had large voltage drops in the later part of a clock cycle, since this is when the combinatorial logic in the datapath between sequential elements switch.

Also, the goal of finding the worst-case test vector has been to find the maximum voltage drop, not worrying so much about *when* in a specific clock cycle it occurs. Since the maximum drops tended to be in the flip-flop switching point of the cycle, they are more likely to cause hold-time failures. Focusing on the hold-time failures, how can the worst-case voltage drop translate to timing implications?

The most hazardous situation for hold-time occurs when the clock path between two subsequent flip-flops is short, so that there is a small margin for any skewing introduced in the clock signals propagating towards the flip-flops. In this situation, if the clock buffers of the capturing flip-flops clock path should experience a voltage drop significantly larger than that of the launching flip-flop, a hold-time failure could occur (see figure 4.1).

Refer to figure 2.13, where a typical voltage drop waveform is shown with annotation of different parts of the voltage drop. Table 2.3 shows the distribution of types of cells. The “deepest” spike of the drop consists mainly of flip-flops, as was discussed. However, before the deepest spike, a certain part of the large drop is often made up of clock buffer cells. These clock buffers could, since they are in the same node and therefore located close to each other, be sensitive to voltage variation from a hold-time violation point of view.

Further work needs to be done to be able to say something conclusive about the timing impacts of the voltage drops found in this thesis. For example, in order to investigate the difference in voltage drop between clock buffers of launch/capture flip-flops clock path, the model needs to be made more fine grained. Today, the model would probably include the voltage drop of both these clock buffers in the same node and therefore voltage drop waveform.

By knowing the location of large voltage drops, and by knowing the exact time when they will occur, more detailed analyses can be made based on this knowledge. From STA (Static Timing Analysis) tools, the most critical hold-time paths can be ex-

tracted. If any of these happen in the area of large voltage drop, this path can be simulated with a gate-level netlist using current data from the small part of the VCD extracted through the use of the methodology developed here.

While the worst-case vectors found in this work give a pessimistic view of the power grid studied, the chip design is known to work satisfactory even with these power supply drops. The results do not directly help to reduce the pessimism of corner-based design, since they tell us the drops are larger than expected, but they do increase understanding of dynamic behavior in a power supply grid.

4.3 Consequences for design flow

This thesis has established a methodology for finding a worst-case test vector for maximized voltage drop. But how does this finding relate to the standard practice of design and sign-off in a real project?

In the introductory chapter, the practice of performing static IR drop analyses was explained. In this context, transient and localized behaviors are not captured. Circuit performance is usually derated in order to account for a conservative value of supply voltage noise. With detailed dynamic voltage drop studies, this pessimism could possibly be reduced. Some EDA tools also include dynamic IR drop analyses, which are usually vector based. A sign-off team working in a dynamic IR drop analysis setting could use the developed method to find a worst-case for their design and use as input for the dynamic analysis.

An interesting experiment would be to compare the dynamic voltage drops found by EDA tools to those found using SPICE as was done in this thesis. Even if the magnitudes of the drops are not the same, the relative voltage drops between worst-case vectors and random vectors should be investigated. If the same tendency of difference between worst-case and random vectors is seen in commercial EDA tools as in the model in this thesis, then that adds to the usefulness of the developed methodology.

A problem with this methodology of finding a worst-case test vector for a design, is that a floorplanned design late in the design flow must be used. At this point it could be troublesome to introduce changes to the design to deal with the large voltage drops found.

4.4 Conclusions

Voltage drops in the model of a power supply grid have been simulated using different input test vectors. The voltage drops found were shown to have a large dependency on the choice of test vectors. It was also shown that the test vector extracted by finding the most power consuming clock cycle resulted in a voltage drop that was much smaller than those found using other methods.

Time-resolution in calculating power consumption is very significant. Using a slightly finer resolution in full-chip power calculations revealed a 60% voltage drop increase.

Frequency-content of the full-chip power consumption turned out to have little or no effect on voltage drop.

The best method for identifying test vectors leading to large voltage drops turned out to be combining different areas of spatial locality around a geometric point on the chip which was known to experience particularly high voltage drops. The single best method was to choose the points in time when nodes adjacent to the known worst node dissipate much power while at the same time the area making up the CPU dissipates much power. Using this methodology, voltage drops of 16% of the nominal supply voltage were found, a voltage drop that is twice that of the reference case.

The process of identifying the worst node remains iterative. It is difficult to state with confidence a general method of identifying this node. Performing several SPICE simulations and observing which node experiences the largest drop will probably be necessary if the methodology should be applied to a different design.

The fact that voltage drops were found that should have been too large for the chip to function, shows that there is still a lot we do not understand about the power supply of an integrated circuit and the timing implications of noise in the grid. The topic remains active in research, and should continue to do so since process scaling and lowered supply voltages make power integrity more problematic.

The input-dependent nature of dynamic power grid analyses makes it natural to consider the worst time instance of a simulation run. This thesis has developed methods for extracting worst-case test vectors which can add a piece to the puzzle of power integrity analysis.

Bibliography

- [1] Andersson, D.A., Nilsson, B., Pihl, J., Svensson, L., and Larsson-Edefors, P., “Supply voltage drop study considering on-chip self inductance of a 32-bit processor’s power grid,” *Signal Propagation on Interconnects, 2009. SPI ’09. IEEE Workshop on*, 12-15 2009
- [2] Andersson, D.A., Svensson, L.J., and Larsson-Edefors, P., “Toward a systematic sensitivity analysis of on-chip power grids using factor analysis,” *Signal Propagation on Interconnects, 2007. SPI 2007. IEEE Workshop on*, 13-16 2007
- [3] Andersson, Daniel A., Svensson, Lars J., and Larsson-Edefors, Per, “Noise-Aware On-Chip Power Grid Considerations Using a Statistical Approach,” *ISQED ’08: Proceedings of the 9th international symposium on Quality Electronic Design*, IEEE Computer Society, Washington, DC, USA, ISBN 978-0-7695-3117-5, 2008
- [4] Svensson, L., Pihl, J., Andersson, D.A., Nilsson, B., and Larsson-Edefors, P., “Towards supply-grid-based derating of timing margins,” *Signal Propagation on Interconnects, 2009. SPI ’09. IEEE Workshop on*, 12-15 2009
- [5] Nilsson, B., *Integrated Circuit Supply Noise Study Based on an Extensive Power Grid Model*, Master’s thesis, Chalmers University of Technology, January 2009
- [6] Popovich, Mikhail, Mezhiba, Andrey V., and Friedman, Eby G., *Power Distribution Networks with On-Chip Decoupling Capacitors*, chapter 6, Springer Publishing Company, Incorporated, ISBN 0387716009, 9780387716008, pp. 125–174, 2007
- [7] Pant, Sanjay and Blaauw, D., “Static timing analysis considering power supply variations,” *Computer-Aided Design, 2005. ICCAD-2005. IEEE/ACM International Conference on*, 6-10 2005
- [8] Nithin, S.K., Shanmugam, G., and Chandrasekar, S., “Dynamic voltage (IR) drop analysis and design closure: Issues and challenges,” *Quality Electronic Design (ISQED), 2010 11th International Symposium on*, ISSN 1948-3287, 22-24 2010

- [9] Synopsys, *PrimeTime PX User Guide Version C-2009.06*, 2009
- [10] Cheng, Wheling, Sarkar, Aveek, Lin, Shen, and Zheng, Ji, “Worst Case Switching pattern for Core Noise Analysis,” *DesignCon 2009*, 2009
- [11] Dekking, F.M., Kraaikamp, C., Lopuhaa, H.P., and Meester, L.E., *A Modern Introduction to Probability and Statistics*, chapter 10, Springer Publishing Company, Incorporated, pp. 135–150, 2005
- [12] Weste, N and Harris, D, *CMOS VLSI Design: A Circuits and Systems Perspective*, chapter 12, 3rd edition, Pearson Edu., pp. 767–780, 2005
- [13] Ferzli, I. A., Chiprout, E., and Najm, F. N., “Verification and Codesign of the Package and Die Power Delivery System Using Wavelets,” *Computer Aided Design of Integrated Circuits and Systems IEEE Transactions on*, volume 29, no. 1, pp. 92 –102, ISSN 0278-0070, jan. 2010
- [14] Addison, Paul S., *The Illustrated Wavelet Transform Handbook*, Taylor & Francis, 2002
- [15] Joseph, Russ, Hu, Zhigang, and Martonosi, Margaret, “Wavelet Analysis for Microprocessor Design: Experiences with Wavelet-Based dI/dt Characterization,” *HPCA '04: Proceedings of the 10th International Symposium on High Performance Computer Architecture*, IEEE Computer Society, Washington, DC, USA, ISBN 0-7695-2053-7, 2004
- [16] Vuillod, P., Benini, L., Bogliolo, A., and De Micheli, G., “Clock-skew optimization for peak current reduction,” *Low Power Electronics and Design, 1996., International Symposium on*, 12-14 1996
- [17] Vittal, A., Ha, H., Brewer, F., and Marek-Sadowska, M., “Clock skew optimization for ground bounce control,” *Computer-Aided Design, 1996. ICCAD-96. Digest of Technical Papers., 1996 IEEE/ACM International Conference on*, 10-14 1996
- [18] Nieh, Yow-Tyng, Huang, Shih-Hsu, and Hsu, Sheng-Yu, “Minimizing peak current via opposite-phase clock tree,” *Design Automation Conference, 2005. Proceedings. 42nd*, 13-17 2005
- [19] Mezhiba, A.V. and Friedman, E.G., “Inductive properties of high-performance power distribution grids,” *Very Large Scale Integration VLSI Systems IEEE Transactions on*, volume 10, no. 6, pp. 762 – 776, ISSN 1063-8210, dec 2002