Design for Testability for PINPOINT and Development of a Power-Grid Modelling Framework

A Project Report

submitted by

SIDDHANT SARAF

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING,
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.
MAY 2015

THESIS CERTIFICATE

This is to certify that the thesis titled **Design for Testability for PINPOINT and Development of a Power-Grid Modelling Framework**, submitted by **Siddhant Saraf**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. V. Kamakoti
Research Guide
Professor
Dept. of Computer Science
IIT-Madras, 600 036

Place: Chennai

Date: May 2015

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor Professor Dr. V. Kamakoti. His support, patience and invaluable guidance has helped me immensely throughout this project. I am inspired by his dedication to research and teaching.

I am also deeply indebted to Dr. Seetal Potluri for guiding me on this project, and with patience, always clarifying my doubts.

I am grateful to the institute for providing me the perfect academic environment to pursue my work and homely atmosphere which shaped my thoughts; and to my friends, for the memorable time I have spent here for the last five years.

Last but not the least, my family has helped me along in unquantifiable but monumental ways.

ABSTRACT

KEYWORDS: Digital Systems Testing, Logically equivalent faults, Fault localization, Power assisted diagnosis, Design for Testability (DFT), VLSI Power Girds.

The current work aims to verify PINPOINT, an algorithm to distinguish logically equivalent faults, using measurements of average currents drawn from supply pins. Along the way, several tools were written to model and design realistic power grids, incorporating actual circuit elements, geometry, and design layouts.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

IITM	Indian Institute of Technology, Madras
RTFM	Read the Fine Manual

NOTATION

R	Resistance
L	Inductance
C	Capacitance
l	length (of interconnect)
W	width (of interconnect)
t	thickness (of interconnect)
s	spacing between interconnects
h	height (of interconnect from GND metal layer)
k	dielectric constant

CHAPTER 1

Introduction

Conventional testing methodologies, ATPG tools using logic-based testing of VLSI circuits fail to diagnose faults beyond the limit of logically equivalent fault(LEFs). PINPOINT, an power-based technique, attempts to cleave LEF sets further by using measurement of cycle average currents drawn by chips. However, for it to be successfully applicable, incorporating and understanding the effects of huge power grids network on a chip involved in the supply of those currents is necessary and unavoidable.

In the current work, we model power grids, study their effects on timing and voltage drops across the chip, and verify the feasibility of PINPOINT.

The rest of the work is organized in the following chapters:

- Chapter 2 provides the background for fault testing of VLSI circuits and provides motivation for the problem which this work aims to solve.
- Chapter 3 goes into the details of the geometry of an actual power grid on a circuit, tis electrical properties, and its layout. We also discuss voltage drops and typical values which characterize the grid.
- Chapter 4 presents the results of our work. It includes simulations results in terms of graphs and observations. We build incrementally from small circuits to complex ones and incorporating different aspects of the work to synthesize the final results. Each simulation is followed up by a short relevant discussion on its significance and lessons learnt.
- Finally, Chapter 5 concludes with a short summary of our work, while commenting on the limitations of it. We also provide several suggestions for building upon our work in any future endeavour.

CHAPTER 2

Background and Motivation

2.1 Fault testing of VLSI circuits

In testing of VLSI circuits, our aim is to classify each chip as usable or not. Yield of a manufacturing process is defined as the fraction of chips which are usable. After declaring a chip as faulty, it pays us to determine exactly where the problem exists–it gives us feedback about where our manufacturing process can be corrected to improve the overall yield. "Diagnostic resolution" aims to identify, through *localizing*, these faults with hopes that this level of resolution will help in correcting the mask and thus in an overall improvement in the process.

Under the single stuck fault (SSF) model for fault testing of digital circuits, there are chiefly two approaches: logic-based and power-based. The most common and classical method employing the logic-based approach is the *Fail/Pass* Fault-Dictionary (Abramovici, 1982). In Pomeranz and Reddy (2008), this method was extend to a *Same/Different* Fault Dictionary. Using these logic based methods, a *diagnostic tree* can be built whose leaves are *equivalence classes* (ie set of faults which are logically-equivalent). The existence of equivalence classes is an inherent consequence of both the method used, and the topology of the circuit. Among these sets on the leaves, those with cardinality equal to one, are *logically-localizable* faults. Those with greater than one, have *logically-non-localizable* faults. Consequently, this approach does not result in 100% diagnostic resolution.

Power-based fault-testing has the potential to overcome these limitations of logic-based methods. Common power-assisted algorithms include *IDDq* and *IDDt* methods, which work by measuring the supply currents (I_{DD}). (Sabade and Walker, 2004) provides a survey of these methods.While *IDDq* measurement based methods can be used both for testing and fault-localization, these methods have become less used due to the increased power leakage in the nanometre CMOS technology. Due to detailed simulation requirements in *IDDq*-based localization methods, they don't scale easily for larger circuits.

During testing, the power consumption, and hence the current consumption of a chip increases 2-3 times. These increased demands on the power network causes large voltage drops; larger than that envisioned by the designer. This in turn causes major fluctuations from the anticipated timing of devices. If the designer did not have foresight to design for testability, a large number of timing violations ma occur and thus result in large number of false positives.

Hence, simulating the behaviour of a power grid under testing is of huge importance. This work focuses chiefly on this part of power grid simulation. In fact, we use power grid modelling and simulation to actually improve diagnostic resolution during chip testing, using the PINPOINT technique.

2.2 PINPOINT

Automatic test pattern generation (ATPG) is a technique which aims to minimize the number of test patterns applied and to bin faulty chips into different fault classes. Though a very useful technique, a major limitation is that ATPG determines only the equivalence class to which a fault belongs, not the fault itself.

PINPOINT, is an algorithm that helps divide each such class into smaller sets. It uses measurement, at power supply pins, of power/current consumption of a chip on applying specially crafted test vectors. It was shown that logically equivalent faults exhibit different power/current signatures, thus allowing us to cleave the equivalence set into smaller sets. PINPOINT overcomes the limitations of number of available output pins by introducing power pins in to the picture. It exploits the fact that for a test vector, two logically equivalent faults produce different levels of activity on different interconnects within the network.

Given this state-of-affairs, PINPOINT, introduced in Potluri *et al.* (2013) proposes a novel approach to power-based fault localization.

2.2.1 Motivating PINPOINT

In a chain of gates, extending from input pin to output pin, a transition in a single gate (aka a *toggle* from 0 to 1 or vice-versa) causes current to be drawn. Moreover, intuitively, we can claim, that a toggle in the middle of the chain is most likely because the chain of gates upto that point would have toggled to finally effect a toggle on the middle gate. Similarly, a gate toggle towards the end of the chain will have been possible only because of successive toggling in the long chain preceding it. We also know, that if any stuck-at fault exists a any point in the chain, it will not allow the toggle to propagate further. Hence, *depending on the location of the fault, the number of toggles in a circuit vary.* Thus, current drawn from supply (pin) should vary too depending on the location of the fault. In an actual circuit, even the capacitances of each individual wires, and the local parameters, will affect power–either reduce, increase or keep it same. This is the crux of the motivation behind search for a better diagnostic resolution technique.

After we have the above observation, the challenge lies in identifying a pair of test vectors which magnify the difference in current(thus power) drawn and thus allowing us to cleave the equivalence class on that basis. To wit, two logically equivalent faults can be coaxed into betraying their different locations through their disparate *power 'signatures'*.

PINPOINT suggests measuring the *current averaged over one complete clock cycle* during the application of test vectors. Since we ought to know the so-called *power signatures* beforehand to differentiate between the two faults, we shall perform simulated testing on our circuit to calculate the currents. In fact, this simulation also involves the search for that pair of test vectors which can maximize the differences between the signatures. The paper ((Potluri *et al.*, 2013)) also suggests an algorithm of how to go about this in a systematic manner.

2.2.2 Test setup

The test setup is the *scan-based* testing: a test vector is serially clocked into the scan-chain, followed by the application of a test-capture cycle (test-per-scan). (A full and detailed explanation of this method can be found in (Abramovici *et al.*, 1994).) Further, the test equipment is capable of measuring the average current drawn during the entire test-capture cycle. We use this information as an indicator of the power drawn by the circuit for that applied test vector.

Though one can do a *only* power-information-based fault diagnostics, the paper shows that a more efficient computationally efficient approach is to use a *hybrid* attack– fist use logic-based methods to build the deepest diagnostic tree you can make, and *then* use power signature information for further cleaving of those leaves which have resisted the attacks of logic-based methods, ie the logically equivalent fault classes (LEFs). Hence, PINPOINT supplements, not supplants, logic-based fault diagnostics.

2.2.3 Inner workings and an Example

Classical logic-based methods ask a (Pass/Fail)? question after application of each test vector, performing successive splits on the universal fault set. These answers to (Pass/Fail)? decision allow us to construct a diagnostic (binary) *tree* as shown in Fig. 2.1. As said earlier, a leaf containing a single fault is logically diagnosable. Else, a leaf with a set of faults is not diagnosable by by Pass/Fail Signatures. The faults within these sets can be isolated by observing them at different outputs (Abramovici *et al.*, 1994). Logic-based diagnosis is over after these two steps, and we left with logically-unique faults and logically-equivalent sets of faults. Now, onto the power-based resolution.

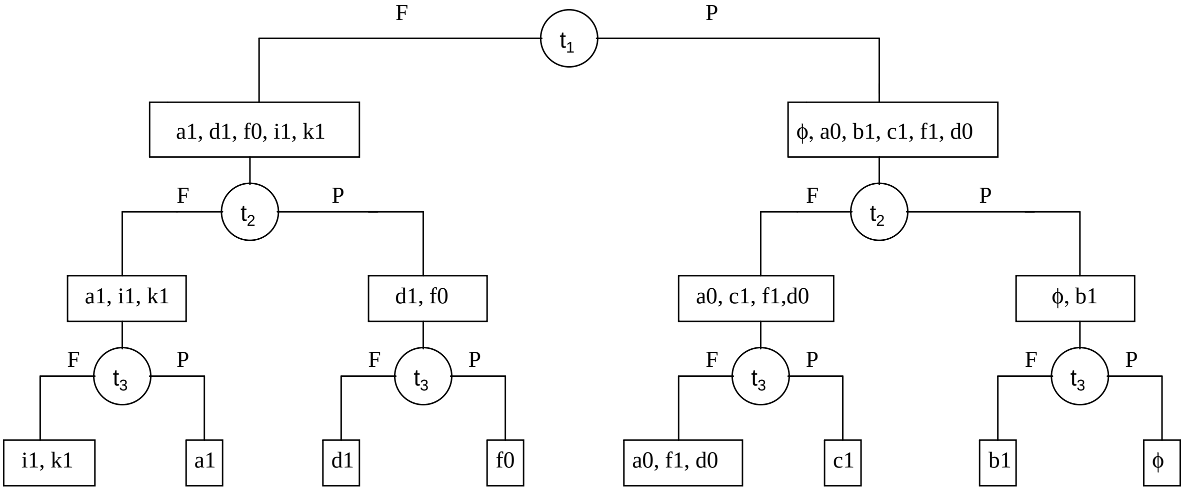


Figure 2.1: Single-Output Logic Diagnostic tree.

A new fault tree, not *necessarily* binary, is constructed, but this time we use the power signature values to subdivide the set of equivalent faults into smaller subsets based on their different power consumptions. In the presence of a fault, two obvious things can happen:

- Some wires which toggle in fault-free circuit, will not toggle in presence of fault.

- Some wires which do not toggle in fault-free circuit, will toggle in presence of fault

This information is available in the current drawn by the chip through the supply pins. Cycle-accurate measurement of this is used to deduce the location of the fault.

As an example, consider the *b01* circuit from ITC'9 benchmark. A small chain of gates, somewhere in the circuit, is shown in Fig 2.2.

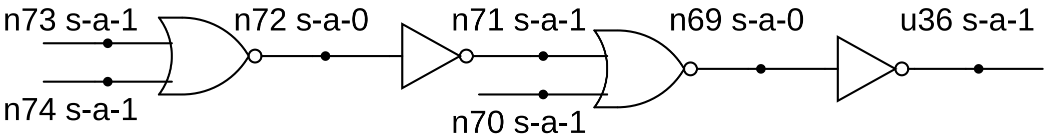


Figure 2.2: Locations of Equivalent faults of a gate-chain in b01 circuit.

All seven faults, *n73 s-a-1, n74 s-a-1, n72 s-a-0, n71 s-a-1, n70 s-a-1, n69 s-a-0, u36 s-a-1*, are logically equivalent faults. Application of suitably crafted test vectors can show, between logically equivalent faults, say *u36* and *n73*, power difference as high as 24% and hence aid in diagnosis.

2.2.4 Limitations

One major limitation stated in the paper on PINPOINT is the question whether the technique works under the presence of large power grids, which are commonplace today. Effects of inductances, resistors, capacitances (both implicit and explicit) might not allow PINPOINT to work as claimed. In the current work, we show that in fact PINPOINT does work and scales for large circuits too. Our current work also provides a simulation based verification of PINPOINT for more realistic chips.

CHAPTER 3

Modelling a Power Grid

3.1 Layout of a typical Power Grid

A typical power grid in a VLSI circuit is designed to meet the voltage and power requirements of *each and every* standard cell (or each logic block, if we talk at a higher abstraction level). In the following two sections, we describe the how this is achieved.

3.1.1 Preliminaries

In a die, the standard cells (the digital building blocks) are placed/etched on a silicon layer as shown in Fig 3.1. The digital signals between them are routed using "interconnects" (basically metal wires). These interconnects, being metallic, are "drawn" on metal layers. Since this network of signal carrying wires is huge and complex, a typical microprocessor can have upto 7-8 metal layers on top of the silicon layer to achieve the interconnections. For these metal layers, the nomenclature used is such: the first metal layer just above(adjacent to) the silicon layer is called "M1", the next one above(adjacent to) M1 is called "M2" and so on upto typically M5–M7.

The power grid, a huge mesh of thick metal wires, carrying V$_{dd}$ and GND from supply pins to cells, is also "drawn" in these metal layers. In fact, the power grid can typically occupy overall 40% of the metal layers. One clear reason for this is that one needs supply power to each and every cell. Moreover, to maintain supplied voltage within 15% of nominal voltage (a typical bounds in design), the wires are much thicker than the signal routing wires; and this is doubly needed since *both* V$_{dd}$ and GND networks must reach all cells, without exception. Hence, design of a reasonable power grid is a essential and can provide great gains in all three constraints of processor design, i.e. power, timing and area.

3.1.2 Grid Geometry

A typical design of a power grid on a die is as follows:

In the silicon layer, the standard cells are laid out in regular rows all with same uniform height. On M1, alternate paralle lines of V$_{dd}$ and GND are provided. Two adjacent rows of cells are so placed such that their V$_{dd}$ (or GND) pins are in contact, as shown in Fig. 3.1.

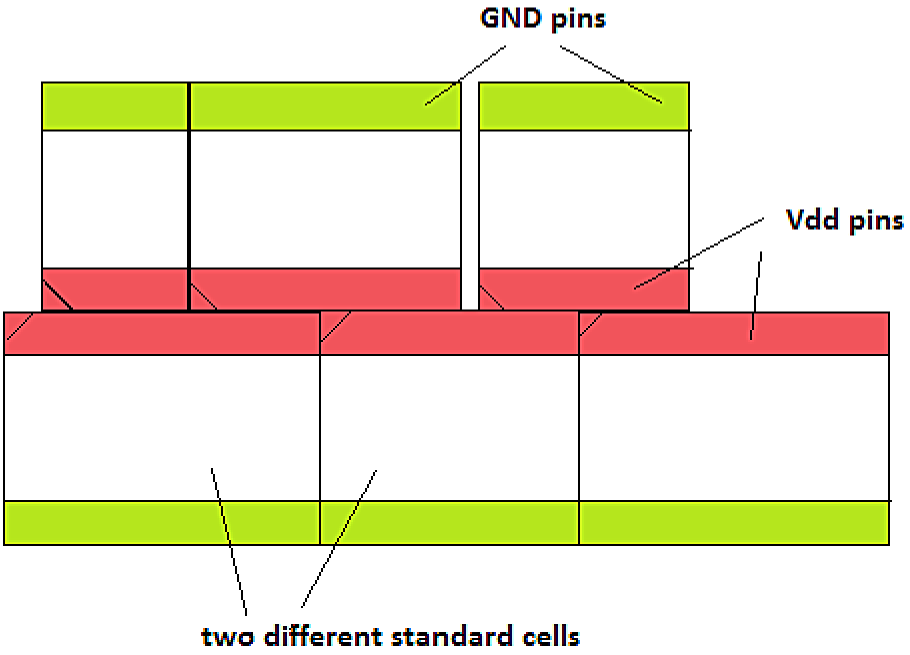


Figure 3.1: Standard cell placement on silicon layer.

Thus, one V_{dd} (or GND) line in M1 supplies *two* rows of cells as seen in Fig 3.2.

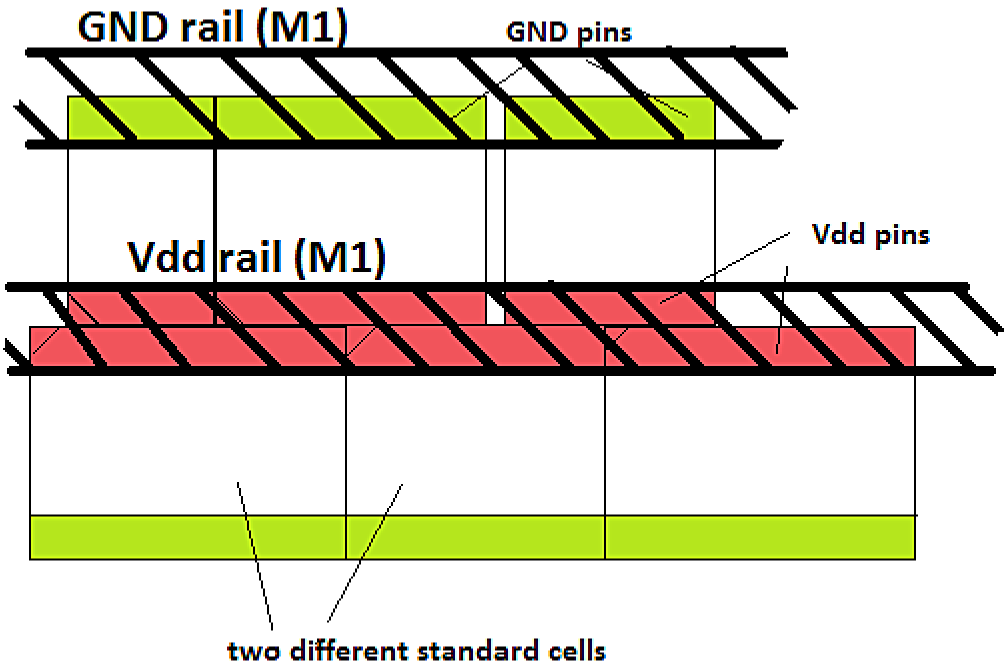


Figure 3.2: Vdd and GND power rails (in M1), each supplying two rows of standard cells.

This M1 network is connected to a higher layer where another set of parallel lines (but this time perpendicular to M1 lines) are charged to V_{dd} (or GND). This might be connected to another higher layers and so on. This layered approach, and having alternate perpendicular lines in each layer, is due to requirements of VLSI design. The topmost power lines are connected via pads to power pins on the package.

3.1.3 Decoupling capacitance (decaps)

Decoupling capacitance, shortened to "*decap*", is any capacitance between the power grid and ground. Decaps play a positive role in the power network by acting as charge storers and suppliers when the power is either not enough, or slow to come from power pins. Parasitic capacitance between any two metal wires, capacitance of devices, and capacitance present on silicon layer between different regions, etc. are *implicit* decaps. However, these are not sufficient to meet the requirements of voltage drops and thus a designer needs to place *explicit* decaps on the power network. These decaps are placed strategically and much research has been carried out to decide on their location, size and number. As a trade-off effect, these decaps bring with them area and leakage-power requirements. Hence, decaps present a challenging optimization problem to a designer.

In our power grid model, we place, indiscriminately, a very small decap at each power delivery point. This, we hope, captures both implicit and explicit decaps.

3.1.4 Voltage drops in a power gird

During the normal functioning of the digital circuit, power is drawn from the power grid every time any switching activity happens. The aim of a good power grid design is to maintain the supply voltage at the delivery point, ie the pins of standard cells, with the requirement that it does not vary more than 15% of the nominal voltage value. There are two major voltage drops which must be taken into consideration during power grid design: *IR-drop*, and *Ldi/dt-drop*. We discuss each drop briefly below.

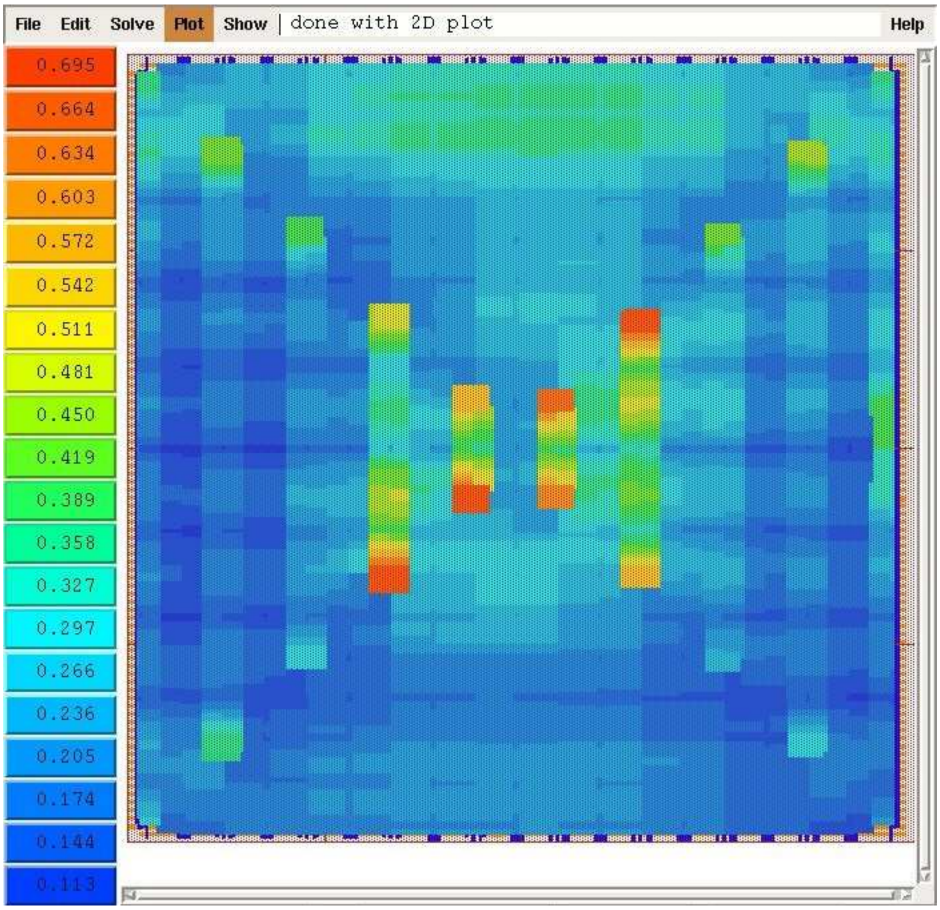
IR-drop

IR drop refers to the drop in supply voltage when it is routed through interconnects, primarily due their resistance. It is important to take this in account during both static and dynamic power analyses.

Ldi/dt-drop

Inductance of package leads and inherent inductance of each interconnect segment causes voltage drops due to time-varying currents. Mutual inductance between any neighbouring metal wires also contribute to this. This is called Ldi/dt-drop.

Hence, overall, the voltage at supply points seen by devices is the nominal voltage minus–the IR-drop and the Ldi/dt drop. Fig 3.3 is a *heatmap* showing the voltage drop in a digital circuit.



done with 2D plot

Figure 3.3: A *heatmap* of voltage across a power grid. (Reference: Li *et al.* (2011))

3.1.5 Static and dynamic power drops

Power consumption by a digital network etched on silicon can be divided into two parts: static and dynamic.

- **Static power:** Is chiefly due to leakage currents drawn by each MOSFET. This current is small, because a MOSFET mostly needs power only when switching, i.e. change of state from 0(1) to 1(0), not to maintain a state.
- **Dynamic power**: Power consumed by MOSFETs for each toggle between 0 and 1. This is a major contributor to power consumption.

Modelling of power grid during static power involes modelling only resistances. In such a quiescent mode, the power gris is modelled as purely resistive. Hence, this simulation is both simple and fast.

Modelling of power grid for dynamic power involves modelling a grid-wire as a RLC element. This simulation is both complex and time-consuming.

3.2 Simulation modelling

In the current work, we do not model the above explained power grid as it is. Instead, we simplify it to a square grid of criss-crossing metal wires; both horizontal and perpendicular wires in the same metal layer. This can be easily translated to the split/layered design in actual dies using vias to connect the horizontal and vertical set of parallel lines.

For static power analysis, the grid is purely resistive since capacitors and inductors do not play any role in the DC analysis of a circuit.

However, for dynamic power analysis, the capacitive and inductive components of interconnects cannot be ignored. We use the Berkeley Predictive Technology Model

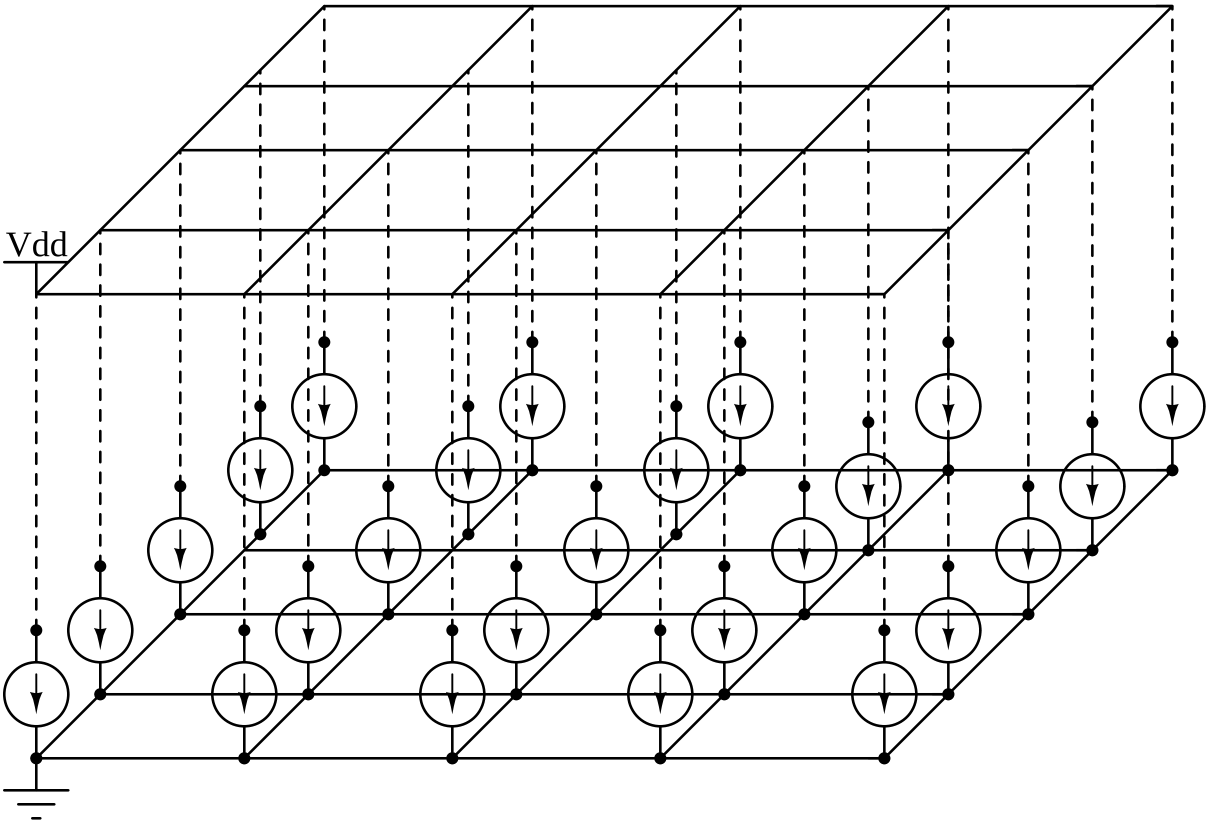


Figure 3.4: A 3-dimensional sketch of our simplified power grid.

(BPTM) to calculate R, L and C of wire segments. See Section 3.2.1.

3.2.1 Values from BPTM

The Berkeley Predictive Technology Model, available through the BPTM website (Nanoscale Integration and Modeling (NIMO) Group, Arizona State University, 2007–), allows one to input the geometrical parameters of an interconnect, and as output you get the electrical properties of that wire segment. We chose the "*Structure 1*" model made of Copper (*Cu*).

- Geometrical parameters
  $l = 7\mu m$
  $W = 1.2\mu m$
  $t = 0.22\mu m$
  $s = 0.8\mu m$
  $h = 1\mu m$
- Electrical parameters
  $R = 0.583\Omega$
  $L = 0.003nH$
  $C_{ground} = 0.5fF$ (we neglect coupling capacitances between wires)
  $k = 3.9$

Moreover, we model the wire using the π-model where the capacitance is split into two, which results in the Fig 3.5

Moreover, the decaps in the circuit must also be modelled. Since in our design, while placing cells, every available gap was filled up using decaps of various sizes, we chose to model them in our circuit at capacitors of capacitance 2fF placed at each node.

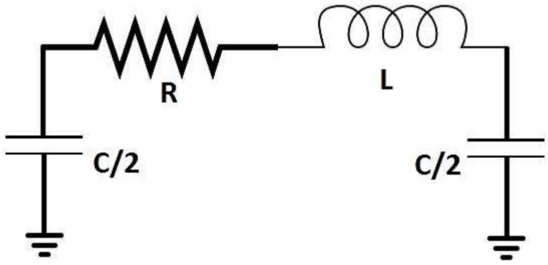


Figure 3.5: π-model of a wire segment.

CHAPTER 4

Simulations and results

The main objective of the current work is to research the following points wrt PINPOINT: feasibility, correctness, robustness, scalability, etc.

In the following sections, we go through the list of simulations performed to understand the phenomenon of power grid voltage drop and its effect on timing performances of the circuit. After that, we finally do the simulations for PINPOINT, and a discussion of our main results follows.

4.1 Our simulation flow

To manage the computational effort, simulation of large power grids is done in two steps:

First, at the silicon layer, digital (logic blocks) simulation is performed and currents (wrt time) drawn by each standard cell is measured. This simulation assumes that nominal (i.e. ideal) voltage, say 1.0 V, reaches all cells, ie, supply voltage drops are non-existent.

Next, at the metal layer, analog (RLC power grid) simulation is performed with each standard cell replaced by a time-varying current sink, where each sink has a value as measured in the first step.

Since, in typical chips, a voltge drop of 10-15% is acceptable, assumptions made in first step do not introduce significant errors.

Further, in a static power analysis, at the end of simulation, we create a so-called *heatmap* to visually represent the distribution of voltage across the grid.

As part of this work the following was done:

- A generalized code, in Python, was written to generate SPICE-compliant netlist of large power grids. One can specify the number of nodes horizontally and vertically. It also allow you to switch between different models of interconnect segments: purely resistive, or BPTM model; varying of parametric values is also possible.
- All simulation were done on two SPICE softwares: ngspice (see: Ngspice Authors) and eldo. Ngspice is an open-source SPICE simulator while eldo is a proprietary software from Mentor Graphics. In our experience, eldo performed much faster than ngspice.
- Various GNUplot scripts were written to plot the current profiles of different simulation results.
- Python code was written to generate "heataps" of voltage drops in a power grid. A heatmap gives a visual, at-a-glance view of voltage across the complete grid.

4.2 The smallest and simplest power gird

See Fig 4.1. It models a single gate during static (non-toggling) conditions.

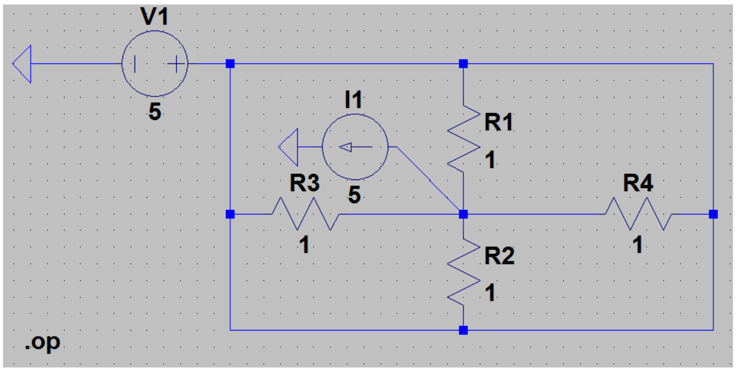


Figure 4.1: The simplest and smallest power grid.

4.3 100x100 grid with periphery sources

Power grid description: A 100 resistor by 100 resistor grid. Each resistor segment is 1 ohms. Current sources of, 1uA at all the internal nodes. Voltage sources of 1V at all the periphery nodes. We get the voltage map as shown in Fig 4.2. As we would expect, the figure is radially symmetrical, with the lowest voltage observed for the innermost gird nodes. The heatmaps are generated using the *matplotlib* Python module.

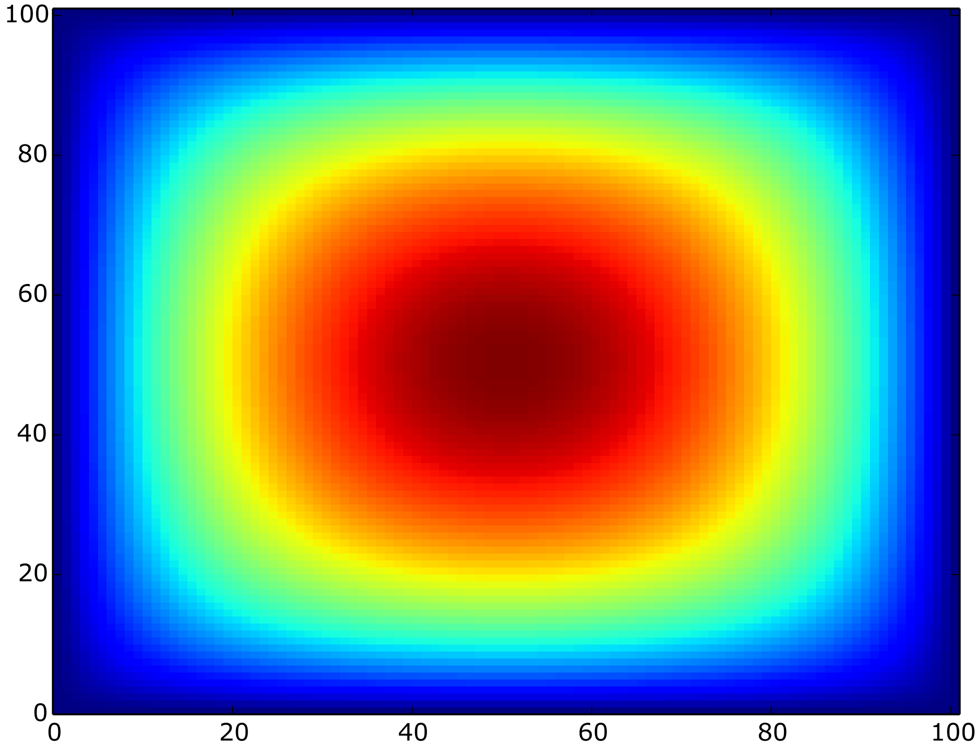


Figure 4.2: Voltage *heatmap* for a 100x100 resistance grid. Voltage spectrum starts from black, indicating 1 volts, and goes towards red as voltage value decreases.

4.4 Effect of voltage drop on timing

4.4.1 Motivation

Timing characteristics of any silicon device depends on the supply voltage. In fact, delay by a gate is inversely proportional to its supply voltage. Hence, a designer needs to make sure that the timing calculation for a circuit, which include clock routing design and critical path analyses, take into account the effect of supply voltage swings. Not ensuring within range maintenance of supply voltage to gates can lead to precarious failures at very late stages of designing and potentially cost a lot.

We use the following relationship between delay (aka *switching speed*) and supply voltage of a gate (note: we assume the best delay to be $100^{-12}s$ @ $1.0V$. Hence, $t_{delay,old}$ for all gates is $100^{-12}s$ and V_{old} is $1.0V$):

$t_{delay,new} = \frac{t_{delay,old} * V_{old}}{V_{new}}$

Thus, we assumes (for simplicity) an inverse relationship between voltage and switching speed.

Apart from timing issues, power fluctuations can reduce noise margins of the circuit, and inject noise.

4.4.2 Timing simulations

Circuit description: Consider the same 100by100 grid of 1 ohm resistors previously simulated. In the digital layer, we have 100 chains of inverters, each chain itself consisting of 100 inverters. After we have calculated the voltage drop for each supply point, we calculate the new delay for each gate. Input is a clock signal to the leftmost inverter of each chain. We compare the switching activity at the leftmost and the rightmost inverters, as shown in Fig 4.3

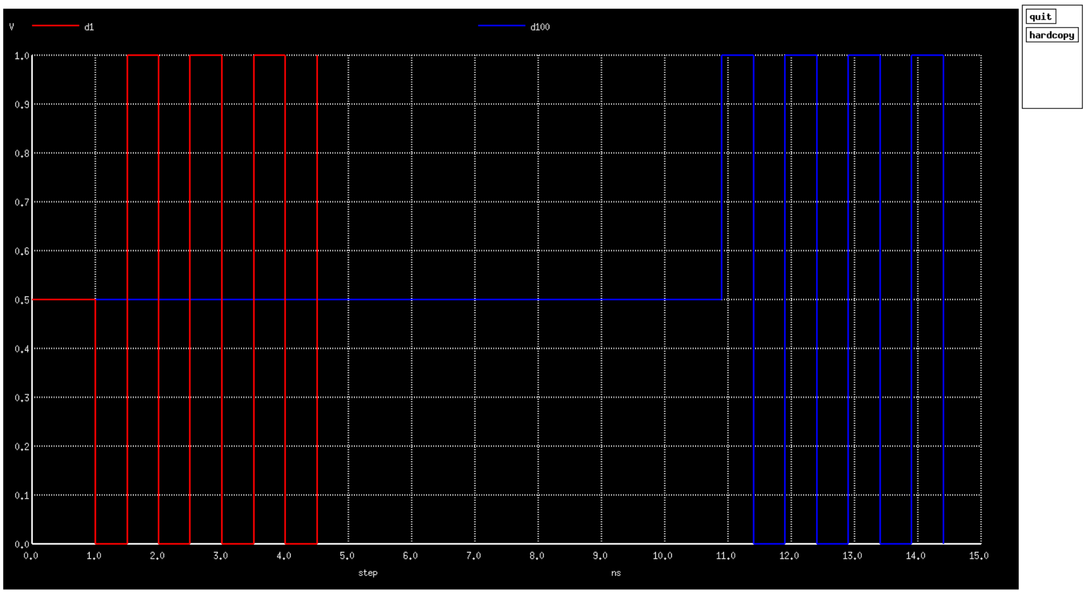


Figure 4.3: Comparison of the switching activity at two ends of an inverter chain. Note the delay in the arrival of the clock signal. Red is the first('d01'), whereas blue is the last('d100') inverter.

4.5 Designing a power grid for a given placement design

A *.def* file is a file generated by the Cadence First Encounter software. It is a description of the placement of standard cells withing a given die size. Given a .def file, with gates, their placement information (in terms of (x,y) co-ordinates), and their interconnection information, the aim is to design a a power grid. We follow the following simple algorithm to design a simple power gird:

- We first parse the given def file to extract (x,y) coordinate of each placed cell. Output of this is a '.points' file.
- We then collect statistics about the placement, such as min/max x/y co-ordinates, sort all coordinates, find out where do we have scope to use same power rail to supply power.

- Design a simple non-uniform power grid for these cells.
- Create netlist of this design. Also output a diagram of this. The netlist incorporates different values of parameters like R, L and C for an interconnect based on its length. Hence, one needs to define a new model for each different segment length.

Following the above steps, for example, we get a power grid as shown in Fig. 4.4. It is for a circuit called 'b01' from the ITC'99 benchmark.

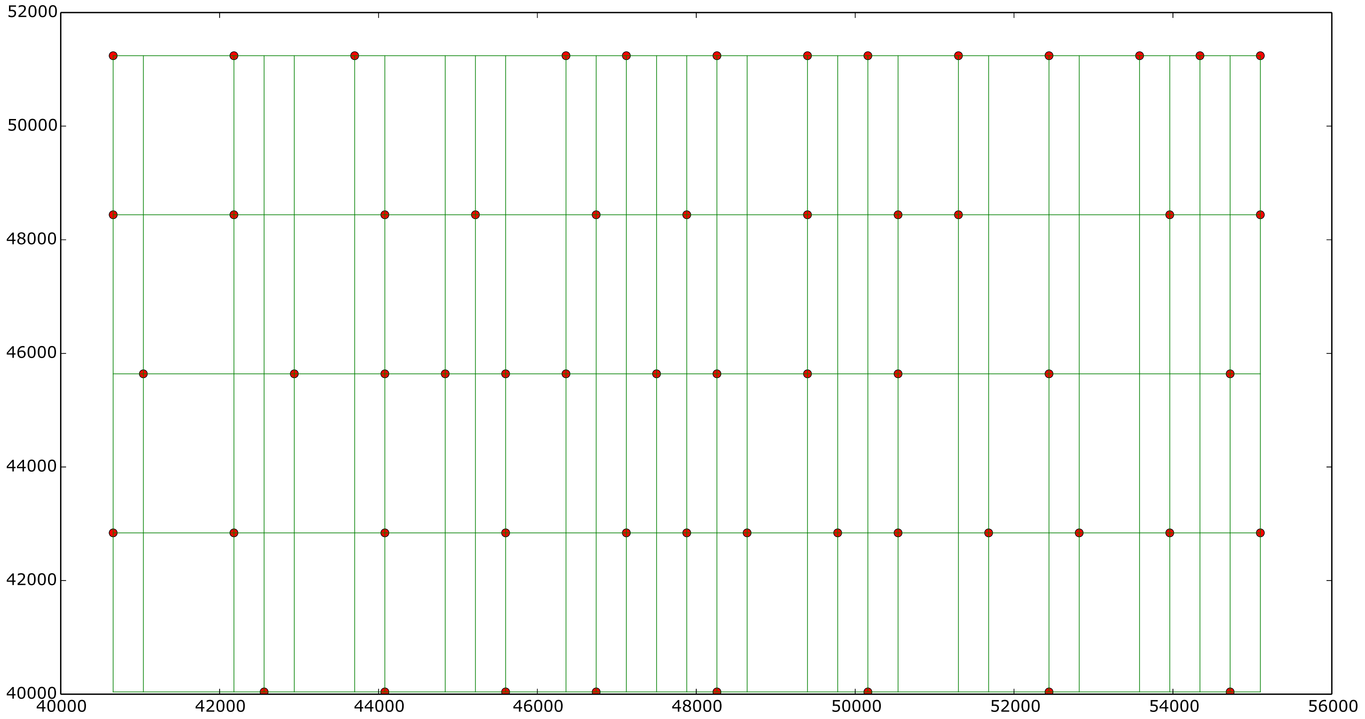


Figure 4.4: A power grid designed from a .def file. The red dots represent the (x,y) co-ordinate of a gate. The horizontal and vertical lines are the power lines. A gate will tap for power at the intersection node of a horizontal and vertical wires.

4.5.1 Limitations and future possible improvements

Since the design of power grid was not the main research objective of the current work, no optimizations were performed to reduce the amount of wiring length. Extension of the current deign can incorporate algorithms for design of minimum *Rectilinear Steiner tree* (Hanan, 1966). However, one must note that, decreasing the amount of wiring can increase the voltage drop for some points. Thus an optimal trade-off needs to be found based on the current requirements of each gate during functional mode. We speculate that this problem will be very hard to solve.

In the current version, only resistor value uses length information of a segment. Capacitors and inductances can and should be included.

We only generate a single metal layer solution for the problem at hand. However, seldom that is the case. At the very least, a power grid is split over two metal layes–one for the horizontal wires, and the other for vertical wires, with *vias* connecting them. Improvement of current work can allow the designer to decide the number of layers one want to divide the grid into and also guide the designer with these decisions.

And finally, this whole framework can be linked to a digital simulator to create a true, full-blown, from-scratch power grid designer and simulator.

4.6 Dynamic simulations

4.6.1 Modelling gate currents

When a gate toggles (aka *switches*) from 0 to 1 or vice-versa, it draws current from the power gird. To avoid a computationally inefficient process, as described earlier, we first perform a digital simulation, obtain he toggle information for each gate as a function of time, and then translate that into current *sinks* attached to the power grid.

Hence a (repeated) toggle looks like Fig 4.5 in terms of current demands placed on the circuit. Parameters of the current "impulse" (Well, strictly speaking, this is not an impulse. But it is so-called because of its very short period of non-zero value.) are as follows:

- $T_{delay} = 30ps$ (initial delay before rise)
- $T_{width} = 30ps$ (can also be called T_{ON})
- $T_{rise} = 1ps$
- $T_{fall} = 1ps$
- $T_{period} = 2ns$
- $I_{ON} = 30uA$

Why I_{ON} = 30uA?

Some "back-of-the-envelope" calculations follow: A typical general purpose commercial chip, is rated as consuming 100W power. Say it has a 100 million gates. During (functional mode of) operation, a 5% activity factor (i.e. fraction of gates switching) is typical i.e. 5 million gates are switching simultaneously. Hence, we have 20uW consumption by each gate per cycle. At an operating voltage of 1V, this translates to a current of 20uA per toggle per gate.

However, power consumption during testing mode is 2-3 times the power in functional mode. Hence, all simulations are performed taking 30uA per toggle per gate.

Note: A gate above can mean either a gate or a standard cell (comprising of multiple gates).

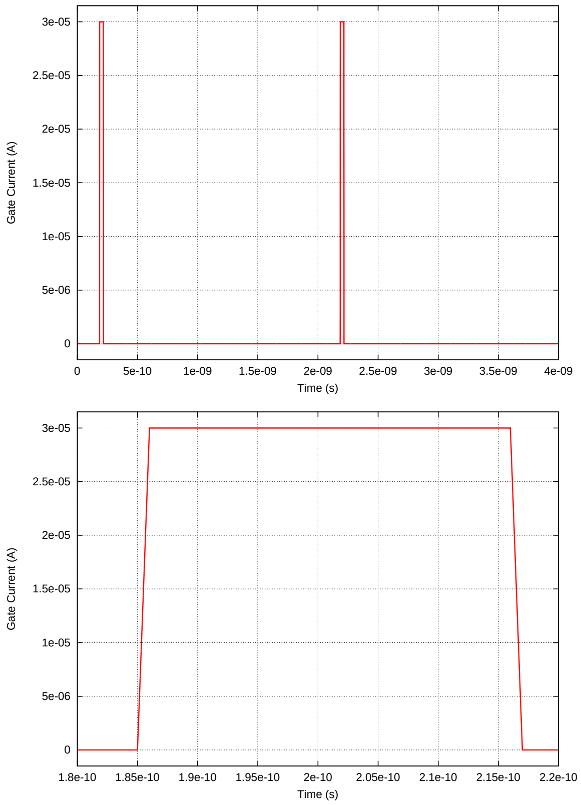


Figure 4.5: (top) A repeated toggle in terms of current. Time period is $2ns$. (bottom) A close-up of the current "impulse".

4.6.2 Adding decaps and inductances

When we add decaps and wire inductances to our purely resistive power grid, we get interesting results.

Fig 4.6 shows current through the supply pin when a single gate toggles at the center of the 100x100 grid. Compare that with Fig 4.7 where a single gate toggles at the opposite (to the voltage source) corner of the grid.

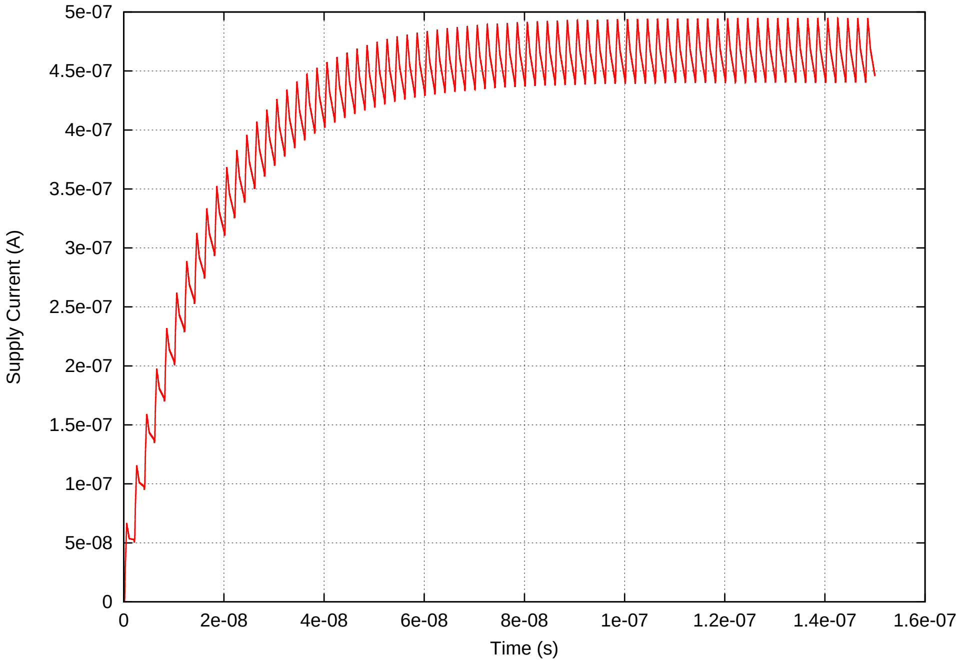


Figure 4.6: Supply pin current for a gate toggling at the center of the grid. Compare with Fig 4.7

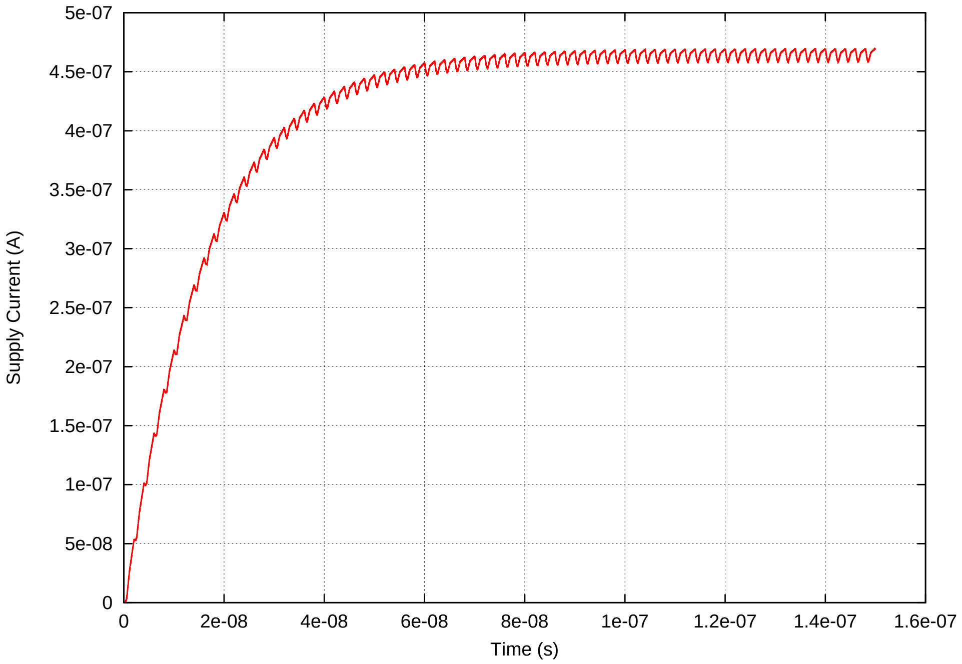


Figure 4.7: Supply pin current for a gate toggling at the opposite (to the voltage source) corner of the grid. Compare with Fig 4.6

4.6.3 Verifying PINPOINT

Recall Fig 2.2, reproduced here in Fig 4.8. Say, we wish to differentiate between (logically equivalent) faults *u36 s-a-1, n69 s-a-0, n70 s-a-1, n71 s-a-1.*

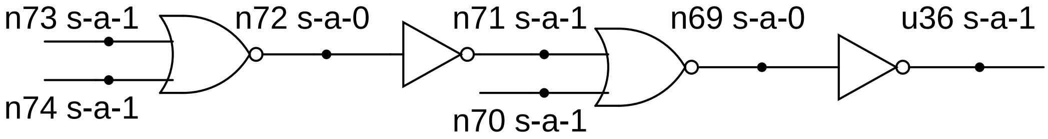


Figure 4.8: Locations of Equivalent faults of a gate-chain in b01 circuit.

Simulation results for these faults is shown in Fig 4.9. If the fault is *u36 s-a-1*, the toggle propagates upto net n69, but is stopped at the last NOT gate. Hence, a total of three toggles happen–the three gates before it. Hence, the most amount of current is drawn (red curve). Next, if the fault was *n69 s-a-0*, the toggle stops propogating only after two toggles (green curve). Lastly, if either of the *n71 s-a-1* or *n70 s-a-1* is the fault, then only one toggle occurs, resulting in a current waveform (blue curve) with average current value equal to the average current of a single toggle.

Hence, Fig 4.9 is the most significant result of this work, because it shows that PINPOINT does work (!).

Further analysis

One will have noticed, that the average value of each curve, after they have setteled, is multiple of a fixed number, in this case $4.6494 \times 10^{-07}A$. The average current in the three-toggle (red) curve is three times this value, in the two-toggle (green) curve is two times this value, and lastly, the average current in the value is self.

This can be simply seen by the *law of charge conservation*. After the circuit has settled down, the capacitors are charged (or at least have fallen into a periodic predictable charge-discharge cycle), and the inductor currents have stabilized, the charge required by the gates is fully met by the power pin. The average current consumed by the circuit is equal to the sum total of average currents provided by all the supply pins external to the chip. In this paper, we assume that the chip has one external supply pin.

This predictable value of the average current is a strong point in favour of PINPOINT.

Feasibility of Current Measurements

Since the proposed method relies on the measurement of currents through the circuit, we need to address the question of whether it is practical to implement such a method.

In this subsection, we would like to quantify if the difference in circuit power consumption for two different faults is sufficient for diagnosing them, through practical current measurements. As we have seen in the earlier sections, an increase in measurement noise degrades the diagnostic resolution. At lower noise levels, two faults can be diagnosed with little difference in power consumption in simulation. But in practice, unless this difference is measurable, the diagnosis process is not feasible.

Dynamic currents can be measured with the DC meter on the tester. In Jiang and Vinnakota (2000), the authors show that dynamic currents can be measured within an inaccuracy of $0.125\mu A$ and also show that power values in μW range can be easily measured.

From our experiments, we observed that the difference in power consumption for different faults is in μA range for circuits with gate count larger than 1000. At higher levels of measurement-noise, it becomes increasingly difficult to distinguish two faults. On the other hand, measurement problem mainly arises in distinguishing faults with very close power values, and the problem is mitigated for larger circuits since circuit power consumption, as well as difference in power consumption for different faults is significant.

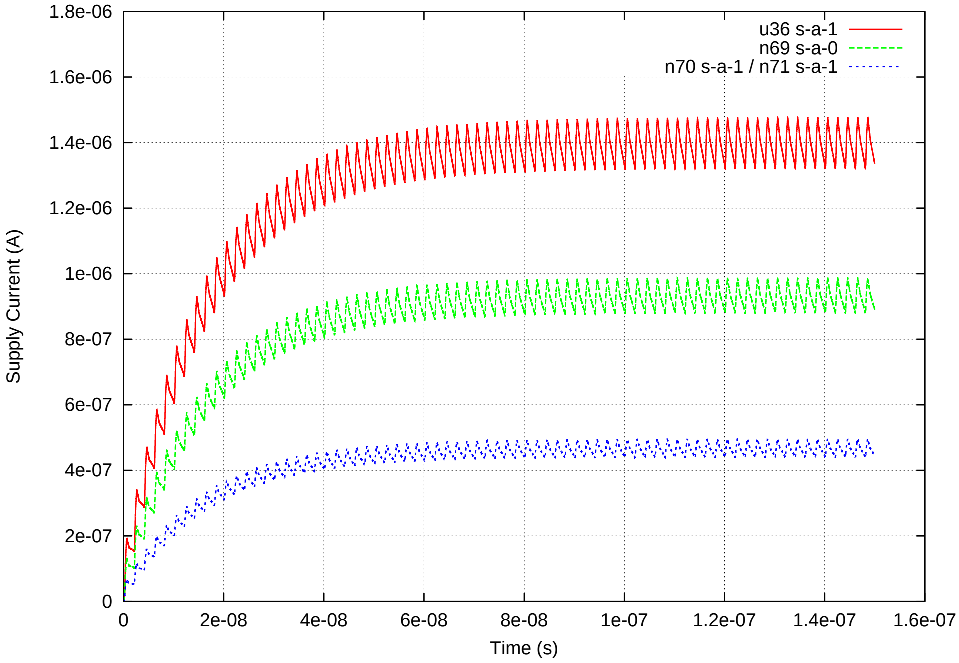


Figure 4.9: Dofferent current profile for different profile. It takes several cycles of repeated application of test patterns to achieve steady state.

Limitations due to Process Variations

Process Variations can be of two types, systematic die-die variations or within-die variations. In the presence of die-die variations, since the threshold voltage of all transistors on a die move towards one side, the power values computed during simulations vary from that of real-chips within a constant factor. This can be leveraged by normalizing the values, by using Energy Consumption Ratio (ECR) proposed in Kim *et al.* (2000).

CHAPTER 5

Conclusion and future work

In the current work, we have explored the PINPOINT algorithm with regards to its practical feasibility and scalability. Moreover, especially since PINPOINT relies on power/current measurement at the supply pin, the question of what happens when you incorporate the power gird into the model. We believe that we have answered these questions in favour of PINPOINT.

A general framework for modelling power grids was developed in Python. Further, simple delay timing analysis was done to ascertain effects of voltage drop on timing characteristics of gates. In future, this can be extended to become a full-blown tool to assist better power grid design, a domain which still is guided by only experience and intuition. If not experience and/or intuition, then multiple iterations through trial-and-error take their place. Whatever the case, the domain is ripe for entry and development of intelligent power grid modelling and design tools.

Work on extending the framework to link with a digital simulator, so that one can perform "full-chip" (hybrid) simulation in one place, is under way.

Design of a power grid algorithm, which incorporates optimization algorithms for making good trade-offs between reducing metal wiring and voltage drops. Reducing availability of wires paths to route power actually increases voltage drops. Hence, an optimum trade-off need to be analysed.

In conclusion, one lesson has clearly emerged from the effort put into compiling this work: Digital circuits, becoming larger *and* complex day by day, cannot meet the demands of low-power, low-area, faster computation without paying special attention to the power grids which are employed in enabling the simplest digital task: switching. Never forget the supply plumbing.

REFERENCES

1. **Abramovici, M.** (1982). A hierarchical, path-oriented approach to fault diagnosis in modular combinational circuits. *Computers, IEEE Transactions on*, **C-31**(7), 672–677. ISSN 0018-9340.

2. **Abramovici, M.**, **M. Breuer**, and **A. Friedman**, *Digital Systems Testing and Testable Design*. Wiley Publishers, 1994.

3. **Hanan, M.** (1966). On Steiner's Problem with Rectilinear Distance. *SIAM Journal on Applied Mathematics*, **14**(2), 255–265. URL `http://dx.doi.org/10.1137/0114025`.

4. **Jiang, W.** and **B. Vinnakota** (2000). Ic test using the energy consumption ratio. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, **19**(1), 129–141. ISSN 0278-0070.

5. **Kim, S.**, **S. Chakravarty**, and **B. Vinnakota**, An analysis of the delay defect detection capability of the ecr test method. *In Test Conference, 2000. Proceedings. International*. 2000. ISSN 1089-3539.

6. **Li, Z.** *et al.* (2011). IBM Power Grid Benchmarks. URL `http://dropzone.tamu.edu/~pli/PGBench/`.

7. **Nanoscale Integration and Modeling (NIMO) Group, Arizona State University** (2007–). Predictive Technology Model (PTM). URL `http://ptm.asu.edu/`.

8. **Ngspice Authors** (). Mixed-mode mixed-level circuit simulator based on berkeley's spice3f5.

9. **Pomeranz, I.** and **S. Reddy**, A Same/Different Fault Dictionary: An Extended Pass/Fail Fault Dictionary with Improved Diagnostic Resolution. *In Design, Automation and Test in Europe, 2008. DATE '08*. 2008.

10. **Potluri, S.**, **S. Trinadh**, **R. Baskaran**, **N. Chandrachoodan**, and **V. Kamakoti**, Pinpoint: An algorithm for enhancing diagnostic resolution using capture cycle power information. *In Test Symposium (ETS), 2013 18th IEEE European*. 2013.

11. **Sabade, S. S.** and **D. M. Walker** (2004). Iddx-based test methods: A survey. *ACM Trans. Des. Autom. Electron. Syst.*, **9**(2), 159–198. ISSN 1084-4309. URL `http://doi.acm.org/10.1145/989995.989997`.

12. **van Rossum, G.** *et al.* (1991–). The Python programming language. URL `http://www.python.org/`.

LIST OF PAPERS BASED ON THESIS

1. **Seetal Potluri, Siddhant Saraf, Satya Trinadh Adireddy, Kamakoti Veezhinathan** *PinPoint II : A DFT Assisted Fault Localization technique using Average Current Information on the Supply Pins* Under preparation.