Spring 2010

Improvement of a Propagation Delay Model for CMOS Digital Logic Circuits

Rodger Lawrence Stamness
San Jose State University

IMPROVEMENT OF A PROPAGATION DELAY MODEL FOR CMOS DIGITAL LOGIC CIRCUITS

A Thesis

Presented to

The Faculty of the Department of Electrical Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Rodger Lawrence Stamness

May 2010

The Designated Thesis Committee Approves the Thesis Titled

IMPROVEMENT OF A PROPAGATION DELAY MODEL FOR CMOS DIGITAL LOGIC CIRCUITS

by

Rodger Lawrence Stamness

APPROVED FOR THE DEPARTMENT OF ELECTRICAL ENGINEERING

SAN JOSÉ STATE UNIVERSITY

May 2010

Dr. David W. Parent	Department of Electrical Engineering
Dr. Lili He	Department of Electrical Engineering
Dr. Sotoudeh Hamedi-Hagh	Department of Electrical Engineering

ABSTRACT

IMPROVEMENT OF A PROPAGATION DELAY MODEL FOR CMOS DIGITAL LOGIC CIRCUITS

by Rodger Lawrence Stamness

Propagation delay models, for CMOS Digital Circuits, provide an initial design solution for Integrated Circuits. Resources, both monetary and manpower, constrain the design process, leading to the need for a more accurate entry point further along in the design cycle. By verifying an existing propagation delay method, and its resulting delay model, calibration for any given process technology can be achieved. Literature reviews and detailed analysis of each step in the model development allow for greater understanding of each contributing parameter, and ultimately, adjustments to the model calibration result in a more accurate analytical model. An existing model was verified and improved upon using TSMC 0.18um and IBM 0.13um SPICE decks, and the resulting improvements can be used to further assist individuals needing a method and model for deriving an initial circuit design solution for integrated circuits.

ACKNOWLEDGEMENTS

I would like to thank Dr. David D. Parent for all his patience, support and motivation throughout the journey to complete this work. I would like to thank my parents for setting such a high bar, my brothers for keeping me grounded, and my beautiful wife for teaching me the most important lessons in life. Thank you all very much.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE
INTRODUCTION

Propagation delay models for CMOS digital logic can enable circuit designers to rapidly produce accurate initial circuit designs without the exhaustive efforts required of analyzing every transistor of each logic gate individually. Propagation delay models (PDMs) offer a cost-effective balance between two vastly different methods of circuit design. At one extreme is the analytical derivation of every element within a given design, accounting for second and third order effects. These results are extraordinarily accurate and even more extraordinarily resource intensive process. The other extreme is the implementation of digital architecture with only logical function and no timing based circuit design, resulting in the fastest possible design time. The first method is prohibitively expensive, and extraordinarily accurate, and the second method is relatively inexpensive, and inaccurate. Between full analysis without simulation and no analysis with exhaustive simulation exists the intermediate domain of PDMs.

Circuit design describes the stage between a circuit's logical definition and physical implementation. A logical definition is "synthesized" converted into an array of CMOS logic gates that represent the circuit's logical function. Gate placement and connectivity provide the designer with a close approximation of the timing problems the circuit will need to overcome. Metal-oxide-semiconductor field-effect-transistor (MOSFET) sizing controls the speed of a given logic block increasing transistor width produces increased speed and reducing size produces reduced speed.

Every logic block is dependent upon the speed of its input and the load of its output. Circuit design complexity comes from the interdependence of the individual logic blocks within a design. If one block is grown to speed up its timing, the block driving it sees an increase in load and subsequently slows down. Upsizing the previous stage can propagate the issue all the way to the first input of the entire circuit. Circuits can have thousands of initial timing issues that would lead to gross over-corrections if not addressed properly. This is where the use of a propagation delay models can provide significant help.

A propagation delay model provides the circuit designer with a close approximation of a circuit’s final device size. A propagation delay model can help the designer avoid numerous iterations of device sizing and testing required by an improperly chosen initial device-sizing scheme. The accuracy and complexity of a PDM varies based on the individual requirements of the designer. Simple designs can use less intricate PDMs while designs with greater complexity require PDMs with greater complexity.

This thesis is based on improvements to methods presented by Baum [1] in an earlier San Jose State University College of Engineering thesis. The published work from Baum [1] is based on the analytical propagation delay models presented in the engineering textbook by Kang and Leblebici [2]. This thesis is the second sequential work to verify and improve upon the analytical propagation delay equations from Kang and Leblebici [2]. Research resources for propagation-delay modeling exist in great abundance in the literature [1-15, 17-26, 28-31]. Choosing a reliable source for citation can be a daunting task because few bodies of work provide exhaustive evidence to substantiate their results. The need for independent verification is the catalyst for this thesis. Verification of existing work will confirm the methods and results presented in addition to serving as a valuable recourse to anyone looking for further research in the same field of study.

The process for building a propagation delay model is based on developing an understanding of common behaviors and effects for a given technology and translating those effects into a reproducible system for rapid analysis. The focus thesis [1] adapts a well-known analytical delay model [Equation 1], and simulation results to calibrate the original model with fitting coefficients. The resulting model accounts for second order effects omitted from the original analytical model. The calibrated model offers an alternative to rigorous and extensive circuit analysis, by trading accuracy for rapid design acquisition.

This thesis provides practical knowledge to an audience ranging from senior-level electrical engineering students, to an experienced (1-5 year) circuit design engineer. This paper also provides research-support to existing PDMs by verifying accuracy of published results [1-4]. Lastly, this work presents three areas of accuracy-improvements to existing PDMs.

The method and analytical models presented in this thesis are targeted for individuals, or small groups, designing a full custom, high-speed, CMOS, digital integrated circuit, with architectural specifications for small a relatively small fanout (typically less than a fanout of four). PDMs are typically designed for a single IC manufacturing technology the content herein can be used to calibrate PDMs for any IC manufacturing process. The method presented in this thesis can be tailored to improve timing accuracy, at a relatively small cost in effort, by applying more stringent modeling-constraints and boundary-conditions.

CHAPTER TWO
BASIC THEORY AND DEFINITIONS

Understanding the concepts throughout this thesis depends upon familiarity with terminology herein. The following terms and definitions are provided to supplement readers less familiar with fundamental elements of digital circuit design. The definitions below pertain to the scope of this thesis.

Body effect and body biasing: The degradation of a transistors performance due to the transistors threshold voltage increasing. The body of a transistor can be, intentionally or unintentionally, moved from the typical supply voltage. Under this effect, the electrical characteristics of the transistor no longer conform to the ideal device behavior.

Capacitance: Units: Farads (F). The amount of stored electrical charge between two electrically aware pieces of material. Capacitors are used as output loads to CMOS circuits to simulate the effects that would be encountered for driving different circuits at the output.

Channel Length Modulation: The shortening of the length of a transistors inverted channel region with increase in drain bias for large drain biases. The channel decrease causes greater current flow.

CMOS: (Complimentary Metal Oxide Semiconductor) Within this text describes the use of complimentary transistors for use in digital circuit design. Every transistor that is activated with a logical “1” (high-voltage or V_{DD}) has a corresponding transistor that is activated with a logical “0” (low-voltage or ground).

Current: Unit: Amperes (A). The amount of electron flow through a conductive media.

Delay/Propagation Delay/Skew: The measurement of a CMOS digital gate delay from the time the input terminal transitions across one-half the supply voltage (V_{DD}), until the output of the digital device responds, transitioning across V_{DD}.

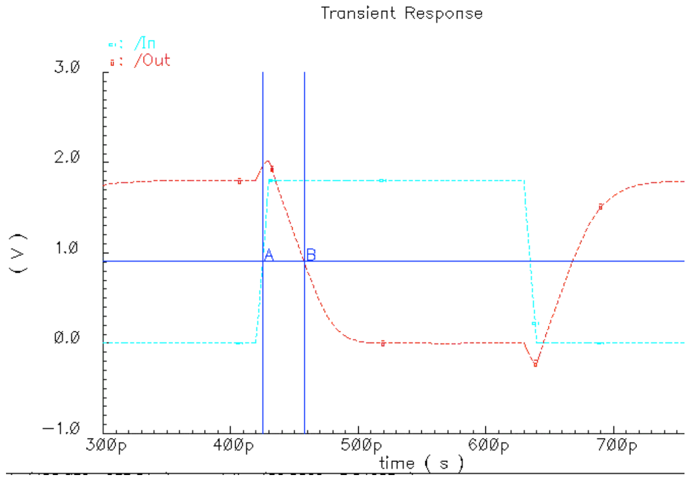


Figure 1. Propagation delay measurement of standard inverter.

Die: The term used to designate a single integrated circuit (IC) boundary on a manufacturing wafer. A wafer may contain 10’s to 100’s of individual dies, with every die being a replication (for large volume production) or completely unique (in the case of research and small volume manufacture).

Digital Design: In this text, refers to the (1's and 0's) of a circuit's logical behavior. Logical devices common to digital design include, but are not limited to; Inverter, NAND, NOR, XOR, AND, OR, and MUTEX.

Fanout: The ability of a given logic gate's output to drive a number of inputs of other logic gates of the same type. The number of logic gates that can be driven is called the fanout.

Input-Load: Describes the total capacitive magnitude, in Farads, that a given logic gate requires to be driven.

Inverter: The most basic architecture of all digital CMOS circuits. This device with reverse the polarity of it's input (if in=1 then out=0, if in=0 then out=1).

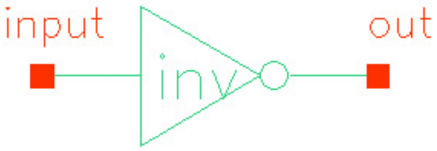


Figure 2. Basic CMOS digital symbol for an inverter.

MOS and/or MOSFET: (Metal Oxide Semiconductor Field Effect Transistor) Specific type of transistor characterized by the use of a thin oxide to isolate the control/gate-terminal.

Output-Load: Describes the total capacitive magnitude, in Farads, applied to the output of a logic gate in a circuit.

Process/Technology: Refers to a specific method for manufacturing MOSFET transistors. Each process contains numerous physically unique attributes from physical dimension to atomic structure.

Resistance: Unit: Ohm (Ω). Material resistance to electrical-flow of current.

Saturation Velocity: The saturation velocity represents the fastest rate that charge carries can transition through a transistor channel (path between the source and drain terminals). The velocity of the charge the carrying components through a transistor, increase with the increase of voltage across the source and drain terminals. This increase rolls off asymptotically to the saturation velocity.

Skew: See "Delay."

Slew/Slope: The time required by a signal/pin to transition from 10%-90% V_{DD} or from 90%-10%V_{DD}.

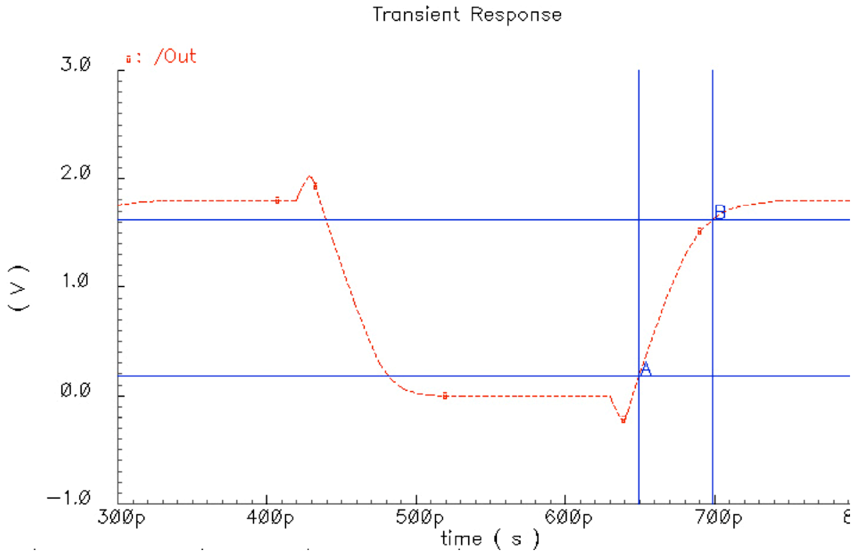


Figure 3. Slope/Skew measurement of rising waveform.

SPICE: (Simulation Program with Integrated Circuit Emphasis). SPICE is an electrical engineering industry standard tool for analog circuit simulation. SPICE provides accurate simulation data based on transistor process manufacturing data.

Stack Devices: In this text, refers to the connecting of transistors sources and drains to form a series path from the supply voltage to the output.

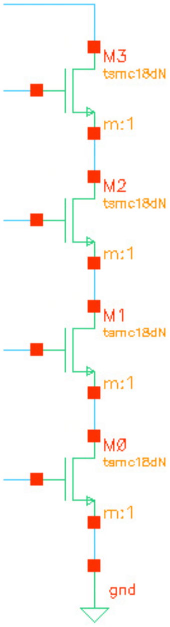


Figure 4. 4-Stacked NMOS devices in series.

Sub-Threshold Current: Amount of electrical current that flows through a transistor when it is logically off.

Transistor: Electrical/Voltage controlled switches with a control terminal and two other terminals that are either connected or disconnected depending on the control terminal voltage.

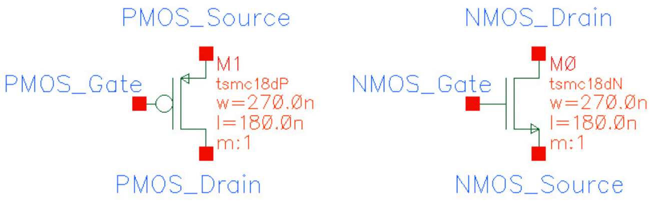


Figure 5. 3-Terminal standard PMOS and NMOS transistor schematics.

VLSI: (Very Large-Scale Integration) Electrical systems/circuits containing hundreds of thousands of transistor.

Voltage: Unit: Volt (V). Measure of electric potential between two points in a circuit.

CHAPTER THREE
LITERATURE REVIEW

To develop an accurate propagation delay model with minimal calibration effort, a well-defined circuit architectural specification is required. Examination of existing PDM calibration methodologies provides a platform for the development of improvements in accuracy [2-5]. The balance between accuracy and solution acquisition time is constrained by the architectural design specification. Every full-custom integrated circuit design presents unique accuracy and effort requirements and the best solutions are commonly comprised of a hybrid model of theoretical equations fitted with simulation-based fitting coefficients.

The method for developing a propagation delay model, presented in this thesis, is the result of understanding existing circuit modeling techniques and applying the key strengths of those models while mitigating the impact of any inherent flaws. Basic propagation delay models account for a small number of factors (output load, circuit voltage and manufacturing technology) that control an actual circuit delay. Empirical and theoretical work on the topics of input slope, fanout, interconnect, and logical effort provide modeling strategies to account for most modeling effects overlooked in the basic models. Updating a basic PDM with detailed modeling effects and fitting the model to a given process provides an increase in modeling accuracy with minimal increase to the modeling complexity.

3.1 Full Custom IC Design

Circuit design work, in the context of a full-custom digital CMOS IC design-flow, is represented in the flow diagram shown in Figure 6. The full custom design flow begins with an architectural specification that provides initial constraints on items including but not limited to manufacturing process technology, system clock-cycle time, circuit-topology, and interconnection or “fanout.” The initial calculations for the individual circuit sizes (transistor widths W_N and W_P) begin after the system level architectural specification in place. The integrated circuits are then simulated using a SPICE-based tool. The resulting circuit timing is analyzed to determine if the architecture’s specified timing has been achieved. Initial simulations often reveal timing paths that fail to meet the architectural specification that will require repeating the design process from the circuit-sizing step forward. The process of sizing, simulating, and evaluating repeats until the architectural timing specification is met.

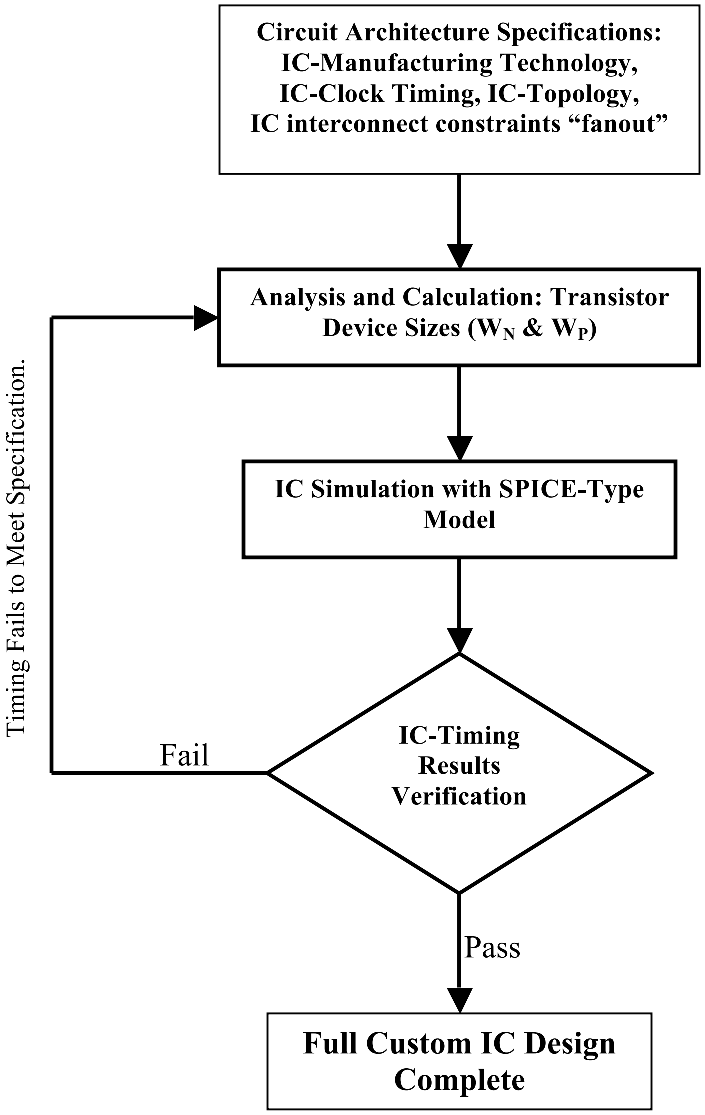

Circuit Architecture Specifications: IC-Manufacturing Technology, IC-Clock Timing, IC-Topology, IC interconnect constraints “fanout”
Analysis and Calculation: Transistor Device Sizes (W_N & W_P)
IC Simulation with SPICE-Type Model
IC-Timing Results Verification
Fail
Timing Fails to Meet Specification.
Pass
Full Custom IC Design Complete

Figure 6. Full custom IC design flow.

3.2 CMOS Digital Integrated Circuits Delay Model

Propagation delay models for CMOS digital logic often omit second-order effects due to their limited impact on modeling accuracy. Input-slope, device sizing, and output-load comprise 90%-95% of the total delay accuracy for most digital circuits [2]. The impact of second order effects are described within the scope of the long channel CMOS propagation delay model. Those effects include, but are not limited to; channel length modulation, carrier saturation velocity, body-effect, and substrate biasing.

The aforementioned exclusions greatly simplify the derivation and resulting propagation delay model. These effects can be accounted for to gain accuracy when precision is needed and when the exact application architecture is known. Channel length modulation is only accounted for in “short-channel” regimes, where the effective channel length of a MOS device is approximately equal to the source and drain junction depths.

Following all the simplifications above, the resulting propagation delay models for rising and falling transitions of a standard CMOS inverter are:

$$\tau_{PHL} = \frac{C_{load}}{k_n(V_{DD} - V_{T,n})}\left[\frac{2V_{T,n}}{V_{DD} - V_{T,n}} + \ln\left(\frac{4(V_{DD} - V_{T,n})}{V_{DD}} - 1\right)\right]$$ Equation 1

Where $k_n = \mu_n \cdot C_{OX}\left(\frac{W_n}{L_n}\right)$ Equation 2

$$\tau_{PLH} = \frac{C_{load}}{k_p(V_{DD} - |V_{T,p}|)}\left[\frac{2|V_{T,p}|}{V_{DD} - |V_{T,p}|} + \ln\left(\frac{4(V_{DD} - |V_{T,p}|)}{V_{DD}} - 1\right)\right]$$ Equation 3

Where $k_p = \mu_p \cdot C_{OX}\left(\frac{W_p}{L_p}\right)$ Equation 4

C_{load} : Capacitive load applied to the output of the inverter.

V_T: Threshold Voltage for a transistor.

V_{DD} : Drain Voltage applied to PMOS Drain Terminal.

COX: Gate-Oxide Capacitance

μ_n,μ_p: Mobility of electrons and holes through transistor channel.

k_n,k_p: Transconductance of the NMOS and PMOS transistors

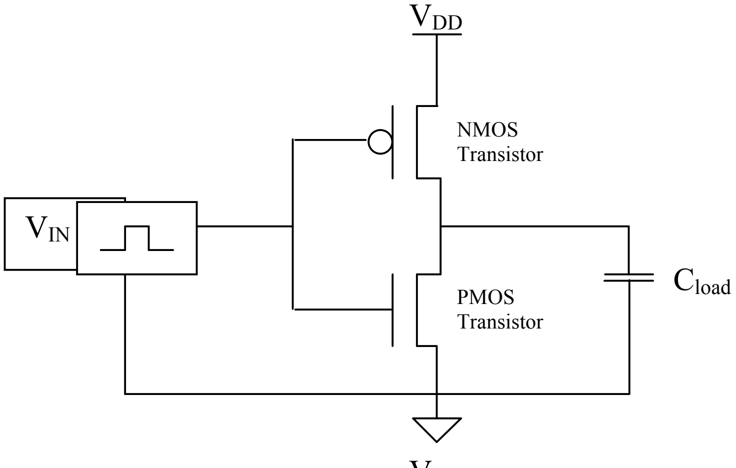


Figure 7. Inverter testbench for propagation delay model.

The propagation models above [Equations 1-4] are explicitly defined with out inclusion of channel length modulation, saturation velocity, and body biasing effects. To improve the accuracy of simplified propagation delay models, iterative analysis and back-fitting has been shown to provide a rapid and reliable solution [2]. Iterative analysis is supported by the Logical Effort method as well [3]. The magnitude of improvement fluctuates across different manufacturing technologies and reveals no simple trends that could allow for more accurate initial solutions.

3.3 Model for Propagation Delay Evaluation

CMOS inverter propagation delay requires consideration for input slope effects and modeling of the source-drain series resistances [4]. The resulting methodology consists of semi-empirical fitting coefficients matched to a propagation delay model for CMOS inverters. Many sources address the propagation delay for inverters [2,3,5-22] and few specifically focus on the effects related to the input slope and source-drain resistance.

Propagation delay is the measure of time from an input signal passing through $\frac{Vdd}{2}$, until the output transition in the opposing direction through $\frac{Vdd}{2}$. The propagation delay can be further deconstructed into two elements. The first element is the delay resulting from a step input, or instantaneous input and the second element is the contribution from the input slope. The second element can be found empirically by measuring the step input propagation delay (in a SPICE simulator) and then the realistic delay of a sloped input and subtracting the step delay from the sloped input delay. The difference between the two delays is the input slope contribution

The propagation delay due to the step response can be verified through the following derivation:

$$\frac{I_{DS}}{C_{load}} \cdot T^{step} = 0.5V_{dd} - \frac{1}{L_{eq}} \int_{0.5V_{dd}}^{V_{dd}} L_{SAT} \, dV_{OUT}$$

Equation 5

Where

$$\int_{0.5V_{dd}}^{V_{dd}} L_{SAT} \, dV_{OUT} = l^2 E_c (y \sinh y - \cosh y) \Big|_{Y_2}^{Y_1} + D$$

Equation 6

$$Y_1 = \frac{L_{SAT}\big|_{V_{ds} = V_{dd}}}{l}$$

Equation 7

$$Y_2 = \frac{L_{SAT}\big|_{V_{ds} = 0.5 \cdot V_{dd}}}{l}$$

Equation 8

3.4 CMOS VLSI Design

Full custom design for very large scale integrated circuits (VLSI) presents many unique design issues that require specialized design solutions. The four elements of full custom design that are inextricably linked together are area of the physical-design, cost of circuit manufacture, speed of circuit and power of circuit. The cost and area are often referenced interchangeably since the cost per die is directly proportional to the amount of dies one wafer can yield [7]. Put another way if the die size for a single design increases by 10% then there are approximately 10% less dies per wafer. The cost to manufacture a silicon wafer is typically fixed [7] and therefore the cost per die is directly linked to the area of the die. If a die grows in size, less will fit on a single wafer and the individual die cost then rises accordingly.

Floor planning is a way to help define the physical size limitations to a given design. Process technology dictates that there is a maximum die size that can be reliably manufactured and sets a limit to the amount or size of the circuits that one die may contain. This limitation is why entire motherboards within personal computers are not entirely on a single chip [7]. Though every technology comes closer and implements more per die than the previous generation, the ultimate goal of producing an entire system on a single chip is yet to be realized.

Maximum speed is a process technology limiting constant. There are many ways to define speed and the most practical definition is based on describing the digital speed. The digital speed limitation can be found by simply making an inverter chain, in a loop, of odd number of inverters. This circuit will oscillate at the “maximum” possible frequency for a given digital circuit. This speed value is not practical since most digital design is implemented with combinational logic. Therefore, the target speed for a system is usually derived from a more typical circuit topology and tested for maximum speed.

Power is a major component of VLSI full custom design. The power for a circuit is related to both the speed and area, but it does not have the direct correlation that area and cost share. Power can increase with area if the area is comprised of an active circuit, but it can also stay the same if the extra area is not being used in typical circuit operation. For example, the built in self-test circuits that will not actively work in the final product, but were installed to debug and test the initial product. Power can increase with speed, exponentially, but only if that speed is uniformly applied to the entire circuit [7].

The last major concept for VLSI design is “typical fanout.” When determining a capacitive load for a given circuit, the rule of thumb is to apply a load that represents four times the equivalent load of the driving device. If an inverter of total width 1um, 0.33um NMOS and 0.66um PMOS, and $\left(100\frac{fF}{\mu m}\right)$, then the inverter has a load of $(1\mu m)\cdot\left(100\frac{fF}{\mu m}\right)=1fF$. A fanout of four would yield a load of $400fF$. This is the fanout of four rule of thumb used as a typical load for a given CMOS device when testing in a test-bench.

3.5 Interconnect Propagation Delay

The objective of modeling interconnect propagation delay is to present a closed form solution to model the propagation delay associated with device performance and interconnected loads. Memory cell architectures have unique conditions for interconnect (array-like placement, interconnect with high resistance poly-silicon wires, and high-volume uniform structure) and require individual compensations to ultimately accumulate their effects into a propagation delay model [9].

Analysis begins with individual transistors and interconnections of a static random-access memory (SRAM) cell. The word line, running the length of an SRAM block is treated as discrete element, only accounting for where it intersects a given SRAM cell. Making every portion of the cell discrete, a singular solution for interconnect load and parasitic effects can be modularized [9]. Modularization provides design leverage since one cell with a particular behavior can be replicated many times. The cumulative impact of every cell detail is then much more important to scrutinize and control, similar to the impacts seen in VLSI design [8,9].

Most elements of an SRAM cell are so short that they can be modeled with simplified resistor-capacitor topologies (similar to a low pass filter). There are however, some interconnect elements that are made from poly-silicon, a high resistance material with transmission like behavior at smaller aspect ratios.

Transistors are modeled with voltage sources, resistors, and capacitors. Combinations of all the above elements results in a network of elements that resemble a fundamental circuits-course homework assignment [16].

The simplified discrete circuit-model enables the network analysis and the ultimate production of a closed-form transfer function. This closed-form solution is re-examined with feedback from actual layout extraction data, and adjusted to account for errors due to omitting nth order effects. High order effects are often omitted since their contributions are so small and accounting for their values is so time consuming [9]. The gap in model precision is bridged through adjustments derived from physical circuit layouts. The layouts measurements are much faster and equally as accurate for calculating high order parasitic effects. Accuracies from the modeling of interconnect propagation delay are within 5% of actual circuit delays [9].

3.6 Delay Model of a RC Chain

Propagation delay models for RC chains present another method of accounting for propagation error through the use of the current behavior in an RC chain. Three simplified RC models comprise the existing structures for modeling current networks propagation delay for interconnect, transmission-gate, and downstream load. Propagation delays can be emulated through equivalent RC transmission line models. A step-input current generator closely matches results of a transfer function model [8]. Final circuit optimizations, using the aforementioned method, result in circuit driving paths with less signal-buffer stages and therefore less total power and silicon area consumed.

Three transmission delay models represent circuit topologies for interconnect or line impedance, pass-gate or transmission-gate impedance, and CMOS logic buffers. The standard transmission line model is comprised of and input step-response current generator driving a resistor-capacitor network as shown in the Figure 7.

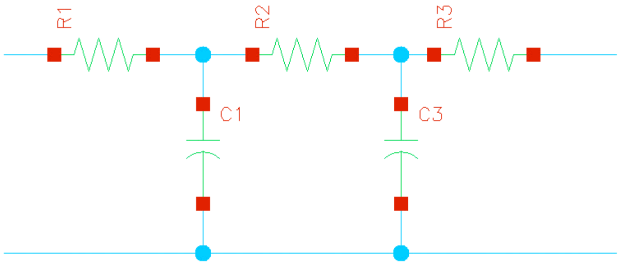


Figure 8. An RC-transmission line model.

The propagation delay for a transmission line is modeled with an input voltage source, rather than a current source. The behavior of an RC ladder network was sufficiently close to the first order circuit model when using Elmore's time constants [8] (with the assumption that the signal-transition was complete at full V_{DD} or ground and therefore effectively has a infinite period). However, the CMOS buffers that drive the RC ladders resemble current sources more than they do voltage sources. This behavior is the catalyst for choosing current input sources for the models rather than the traditional voltage inputs found in most transmission signal analysis techniques.

The result of using an input current source to drive RC ladder networks leads to a significantly simplified propagation delay model compared to traditional circuit propagation delay models. This method of optimizing paths has produced smaller propagation delays and ultimately required less signal repeaters than traditional methods. The use of less logic to achieve the same signal-timing objective means an overall savings of power and silicon area in the final product.

3.7 Propagation Delay Model Based on Charge Delay

The relationship between available charge and the resulting propagation delay can be expressed in the charge delay model. There is a method to evaluate propagation delay for complex CMOS gates from an inverter delay model. The inverter delay is based on and n^{th}-power law MOSFET model. Transistor collapsing techniques, developed for complex gates, take into account the effects of short-channel, internal coupling capacitance, and the body effect [5].

MOS device stacks can be simplified into slope delay curves. These curves represent a typical inverter with a varying output load. Making a complex stack equate to a simple inverter model, can radically simplify the evaluation of complex circuits at the gate level.

Capacitive values for the parasitic and load capacitors are lumped together to represent a single static load. The currents are derived from propagation-delay, slope and lumped capacitances. The charge delay concept may be expanded through deriving a delay-in vs. delay-out table. This table is the grand simplification of the complex circuits into a much simpler delay chart containing curves for each previously complex device that is now reduced down to an equivalent inverter.

3.8 Logical Effort

Logical Effort (LE) is a method for analyzing digital-circuit timing delays and using the resulting information to identify the relative trade-offs between circuit-design complexity and circuit-speed. The fastest circuits tend to have the greatest logical complexity and power consumption [3]. The LE method presents two mechanisms for understanding a circuit's abilities and limitations. These mechanisms are "electrical effort" and "logical effort" [3].

The basic premise of LE can be demonstrated through qualitative analysis of a simple circuit. For an inverter of any given manufacturing-technology there are design tradeoffs between speed, size, power, and capacitive-load. Output-delay for a device can be simplified with the following LE equation:

$$d_{abs} = d \cdot \tau \qquad \text{Equation 9}$$

Where “τ” is the basic delay unit for an inverter driving a fanout of one, without accounting for any parasitic capacitances. The “d” represents the collection of all other effects lumped into a singular quantity. The “d_{abs}“ is the realized delay for the inverter with all the parasitics and other effects combined.

The lumped-effects “d” is reduced to two major components:

$$d_{abs} = f + p \qquad \text{Equation 10}$$

The fixed portion of the delay is “parasitic delay” (p), and the variable portion is called the “effort delay” (f). The effort delay is the product of a circuit’s “output load” (h) and “logic complexity” (g).

$$f = g \cdot h \qquad \text{Equation 11}$$

The complexity of a circuit will change that circuit’s ability to drive a load. Less current is available to drive an output load for circuits with greater path complexity. An Inverter and a NAND gate of equal transistor sizes and driving equal capacitive loads will produce different magnitudes of current due to their relative logical complexity. This difference is accounted for in the term for logical effort (g). The same circuit driving different fixed capacitive loads will result in varying current delivery. This behavior is represented with the term for electrical effort (h).

Electrical effort represents the ratio of a circuit's output load capacitance relative to the input capacitance.

$$h = \frac{C_{out}}{C_{in}}$$ Equation 12

Combining the individual components for a particular circuit culminates in the following summary expression:

$$d = (g \cdot h) \cdot p$$ Equation 13

3.9 Document Review Summary

The citations for the literature review were selected by highest volume of citations in the thesis by Baum [1-7]. The concepts presented in the cited literature cover the key aspects needed to understand propagation models and their development. From the initial inverter-chain test-bench [3], to the extraction of an initial propagation delay τ [2], to the effective resistance calculations [4], all the essential elements are assembled from the cited papers. The approach taken by Baum is only one of many possible combinations, as demonstrated in the improvements presented in the results-conclusion of this thesis [1].

The collection of citations was selected for their contributions to each of the major steps in the calibration process presented by Baum [1]. Each citation provides research necessary to understanding the fundamental principals governing their respective stage. The inclusion of conflicting citations is intended to provide examples of where the methods from Baum [1], may be improved upon. The reference literature provides support to show that the method developed in Baum was well planned and thoughtfully executed [2-9].

CHAPTER FOUR
METHOD FOR CALIBRATION OF A PROPAGATION DELAY MODEL

Calibration of a propagation delay model requires six major steps. Each step provides data is that used to adjust an initial analytical delay-model and the resulting solutions. The purpose of the calibration steps is to improve the accuracy of an analytical delay-model solution from 90%-accuracy to greater than 95%-accuracy.

The fist step in the calibration method of a PDM is to determine a propagation delay target. The delay target will be used for all subsequent method-stages as the ideal propagation delay for a given logic block. The second step involves calibrating a single inverter to meet the target propagation delay. The calibration in this step refers to adjusting the W_N and W_P values until the target delay is met. The third step is comprised of extracting the timing constants from the inverter testbench to satisfy the Kang-Leblebici PDM. The fourth step consists of extracting fitting coefficients from the initial PDM found in step three. Step five and six consist of iterations through the modeling steps three and four with focus on the effects of the input-slope and output-loads to the test circuit. The changes in slope and load can result in discrepancies between the model and the actual circuit performance and therefore a range of behavior over a typical range of conditions will produce an average value for the PDM fitting coefficients.

The manufacturing process file provides manufacturing parameters for the transistors, specific to a given manufacturing technology. A process file is often called a "SPICE-deck" [7,9]. The physical device parameters and subsequent calculations are wholly dependent upon the technology file being evaluated. Values from one SPICE-deck do not be scale to another process for most cases. The TSMC 0.18 μm (TSMC0.18) process file is used in the following example for greater clarity

4.1 Propagation Delay

The first step in calibration of a PDM involves simulation test-benches. The test-benches are used to extract circuit behaviors. Those behaviors are used to adjust delay results in analytical circuit delay models.

The circuit topology used to measure single-stage propagation delay is an inverter chain, as shown in Figure 8. The use of seven stages is not required but has shown to be the sufficient number of stages to stabilize the stage delay. When the stage delay between the last stage and the second to last stages is within 0.25% total-delay, the chain is sufficiently long enough to extract an accurate reference stage delay value.

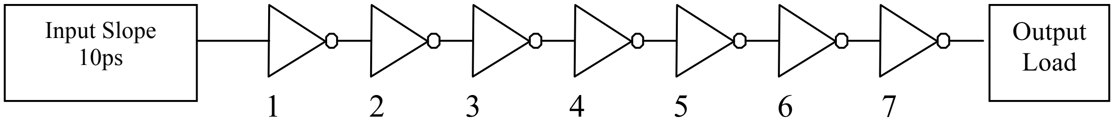


Figure 9. 7-stage inverter chain.

The MOS device sizes of the inverters in the seven-stage chain were implemented with two different schemes. Both schemes used a device ratio for PMOS to NMOS of two. Initial device sizes are minimum and two times minimum for the NMOS and PMOS transistors, respectively. The minimum device sizes for a manufacturing technology are listed as “TNOM” for both NMOS and PMOS data sheets. The units for TOX are in $4 \cdot 10^{-9} meters$. The listing in a datasheet “TOX=4e-9” as shown in Appendices A and B. The second set of MOS device sizes is twenty-five times greater than minimum initial sizes.

The first inverter of the chain (I0) was sized with initial NMOS and PMOS transistor width and duplicated seven times to avoid repetitive circuit device sizing of every inverter in the inverter-chain as shown in Figure 9. The test-bench uses the term “vdd!” to identify a global maximum voltage within the context of the Cadence simulation environment. The last inverter stage is connected to a capacitor to simulate a realistic circuit environment for the inverter chain. The size of the output capacitor is calculated to match the input capacitance of each of the inverter chains’ stages. The gate capacitance per micron, referenced from the process file, is multiplied with the total inverter MOS device size to generate the equivalent output inverter load.

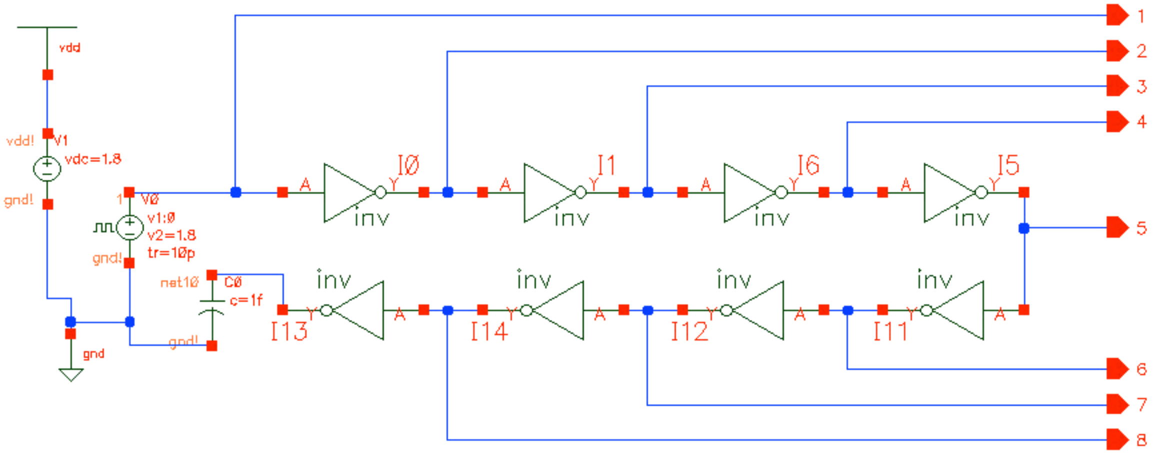


Figure 10. Inverter chain schematic test-bench.

After the inverter chain is drawn, connected, and sized, the next item to complete is the DC voltage source. This will allow for referencing the term “vdd” in other sources so that any central change to the supply voltage will automatically be reflected across all sources. The TSMC0.18 process file uses one and eight-tenths volts for the operating voltage “V_{dd}.” The last voltage source to complete is the “vpulse.” This source provides the input waveform to the inverter chain. The values for v1 and v2 represent the minimum and maximum values for the input wave. The period is the amount of time between the output voltage beginning the transition from v1 to v2 and the time the output voltage returns completely from v2 to v1. The slope is set to be infinite by applying 0 ρs as a rise time. A 0 ρs slope is the same as an infinite slope. An input signal with no slope delay is able to transition instantly from one voltage level to another. The set the appropriate period for the input pulse wave may require a few initial guesses. The simulation needs to be long enough to see all inverter stages toggle while avoiding excessive length that would result in redundant data. Users with more circuit simulation experience can make rough estimates based on scaling cycle-periods from the closest known technology. A test value of 260 ρs was used for the initial period and the pulse width was chosen to be half the period for an even waveform.

The termination-capacitor, at the end of the inverter chain, should be the same capacitive load as the input gate capacitance for all seven of the upstream inverters. Matching the capacitive value will result in the greatest accuracy. The capacitor value is calculated using values from the manufacturing datasheet (CGDO, CGSO, CGBO) as shown in Appendices A and B. The datasheet-values are the multiplied with the MOS devices' dimensions. CGDO represents the capacitance per unit-length of the gate to drain overlap. CGSO represents the capacitance per unit-length of the gate to source overlap. CGBO is the primary component of gate capacitance and represents the capacitance per unit length for the gate to body overlap.

The final steps in the set up of the single inverter test-bench simulation are the selection of simulation type and duration. A "transient analysis" is used for the inverter chain test-bench. The transient test-bench allows a simulated circuit to run without interference, for a duration specified by the user. The Cadence "Analog Simulation Environment" (ASE) derives initial conditions for the transient analysis. The initial conditions allow for measurement of node voltages prior to the arrival of the first input signal. The ASE identifies repetitive behavior and applies the appropriate starting conditions to the simulation. To ensure the ASE will behave in a predictive manner, the transient analysis must be set to a length of slightly more than one full test-bench period. If not using the "Cadence Design Suite," verify the results by running the transient analysis at least seven full cycles to ensure the results are equal to the single period run outlined above.

The propagation delay is calculated with the ASE built-in wave calculator. If using another analog simulator that does not have a wave calculator, point-analysis will suffice. For this simple case using point analysis is quicker than a wave calculator. To obtain the propagation delay for a specific device, the cursor cross hair is positioned over the input signal waveform where it transitions past $\frac{V_{dd}}{2}$ (rising or falling) and the simulation time is recorded. Next, the cross hairs are placed on the inversely corresponding output transition, at $\frac{V_{dd}}{2}$, and the simulation time is recorded. The propagation delay for the device results from subtracting the first recorded time from the second.

The simulation is repeated for a second inverter chain with transistor device sizes twenty-five times greater relative to the previous inverter chain. The resulting propagation delay for the final two stages should be stable (within 0.25%). The minimum delay of the four measurements is selected as the target delay for the calculations that follow.

4.2 Calibration of the Single Inverter

After the minimum value for propagation delay, "τ," has been select through the preceding steps ($\tau = 32.3ps$), the next goal is to build a single-inverter test-bench with a static capacitive load and user generated input slope.

The single-stage inverter test-bench requires an output load capacitance, device sizing for both PMOS and NMOS devices, and the input slope from the previous step. The device sizes will be calculated first, using analytical methods from reference texts [2,3]. The output capacitive load is calculated as a relative quantity with respect to the input capacitance of the initial inverter size.

Transistor device sizes will not be the minimum or twenty-five times the minimum, as used in the previous inverter chain. The PMOS and NMOS sizes have to be calculated using the "Kang and Leblebici propagation delay model" [2]. The Kang and Leblebici inverter propagation delay model is noted below in Equations 15 and 16 [2].

$$W_N = \frac{A \cdot C_{load} \cdot L_N}{\tau_{PHL}} \qquad A = \frac{1}{K_{NP} \cdot (V_{DD} - V_{TN})}\left[\frac{2 \cdot V_{TN}}{V_{DD} - V_{TN}} + \ln\left(\frac{4 \cdot (V_{DD} - V_{TN})}{V_{DD}} - 1\right)\right] \qquad \text{Equation 14}$$

$$W_P = \frac{B \cdot C_{load} \cdot L_P}{\tau_{PLH}} \qquad B = \frac{1}{K_{PP} \cdot (V_{DD} - |V_{TP}|)}\left[\frac{2 \cdot |V_{TP}|}{V_{DD} - |V_{TP}|} + \ln\left(\frac{4 \cdot (V_{DD} - |V_{TP}|)}{V_{DD}} - 1\right)\right] \qquad \text{Equation 15}$$

W_N = NMOS transistor width

W_P = PMOS transistor width

C_{load} = output load capacitor as shown in Figure 10.

L_N, L_P = Transistor channel lengths for both PMOS and NMOS transistors (TSMC0.18)

All other parameters are calculated from or taken directly from the TSMC0.18 datasheet (V_{DD}, K_{NP}, K_{PP},V_{TN}, V_{TP}) as shown in Appendices A and B.

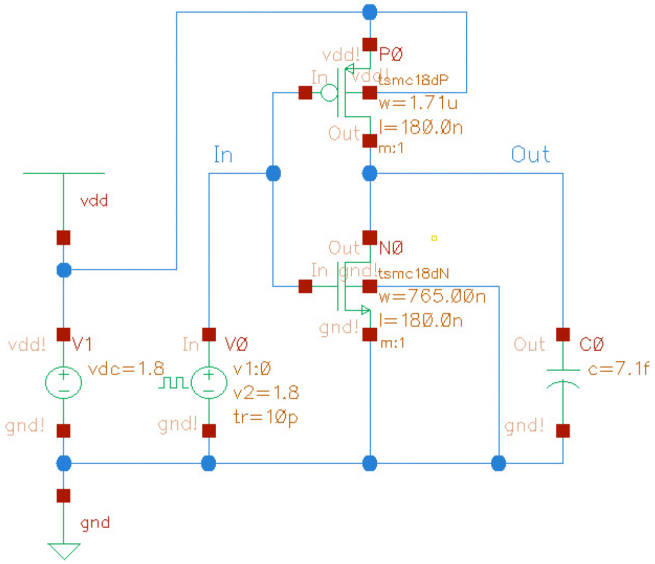


Figure 11. Single inverter test-bench for W_N & W_P.

The propagation delay value, $\tau = 32.3ps$, is used for both NMOS and PMOS device sizing. Symmetric propagation delay is a common practice to simplify design-sizing process due to elimination of delay variations that ultimately add complexity to a sizing methodology. This delay simplicity comes at a cost to power and total-delay and is detailed in the Results section of this thesis.

Initial device sizes for W_N and W_P come from the minimum device sizes used in the inverter chain test-bench. The capacitive load "C_{load}" at the output of the inverter needs to be calculated. The magnitude of the C_{load} will be equivalent to four times the capacitive load of the test-bench inverter. The value of four, or fanout of four, is an industry standard fanout [1]. More discussion on the accuracy of this assumption is detailed in the Results section. The output load of the inverter is calculated with the physical device parameters listed in the manufacturing process files as shown in Appendices A and B.

The initial values for the test-bench:

1) W_N = 0.484 μm.
2) W_P = 0.968 μm.
3) C_{load} = 7.14fF.
4) Input slope of 80ps (measured from the inverter chain test above).

The purpose of the single inverter test-bench is to calibrate the analytical solutions for NMOS and PMOS sizing, with results from SPICE-based simulations. From this starting point, iterative cycles of simulation, measurement, and transistor resizing, will be executed until the resulting propagation delay matches the timing target. The initial sizes will often not meet the timing target due to the nature of miscorrelation between analytical derivations and SPICE based simulations. The analytical equations, Equations 15 and16, are based on assumptions that omit important second order effects of saturation velocity and channel length modulation [2].

The error results from each simulation are used to update the transistor sizes. If the propagation delay was measured to be "62.6ps," for the output falling transition, the propagation delay is $\tau_{error} = \frac{64.6ps}{32.3ps} = 2$. The NMOS device is updated using the error percentage to increase the transistor size by the same amount $W_{N-new} = W_{N-current} \cdot \tau_{error} = 0.484\mu m \cdot 2 = 0.968\mu m$. The results, as shown in Table II, detail the process of using error to adjust device sizes and re-testing. These steps repeat until the transistor sizes result in a delay less than 1% from the target propagation delay. After seven simulations, the propagation delay error is less than 1%. The device sizes can have determined for matched propagation delay.

4.3 Derivation of Timing Constants

The simulation-based values for A, B and R can now be calculated. "R" is the PMOS to NMOS device ratio, "A" represents the effective device resistance of the NMOS transistor, and "B" represents the effective device resistance of the

PMOS transistor. Rearranging the earlier equations, as shown in Equations 15 and 16, for propagation delay:

$$W_N = \frac{A \cdot C_{load} \cdot L_N}{\tau_{PHL}} \qquad \text{Equation 16}$$

$$W_P = \frac{B \cdot C_{load} \cdot L_P}{\tau_{PLH}} \qquad \text{Equation 17}$$

Solving for A and B:

$$A = \frac{W_N \cdot \tau_{PHL}}{C_{load} \cdot L_N} \qquad \text{Equation 18}$$

$$B = \frac{W_P \cdot \tau_{PLH}}{C_{load} \cdot L_P} \qquad \text{Equation 19}$$

A and B values are calculated from the simulation based propagation delay as apposed to the process parameter-based calculation earlier. By using the simulation data, the results will implicitly incorporate all the secondary effects that were omitted from the original calculations. The values for A and B now include the saturation velocity, channel length modulation, and body bias effects.

The completed steps to this point:

1) The target propagation delay and slope were extracted from an inverter chain test-bench.

2) The slope and delay values were used to calculate the initial device sizes of an NMOS and PMOS transistor for the inverter test-bench.
3) The output capacitive load was calculated from the initial device sizes and the target fanout of four times the input.
4) Seven iterations of device sizes for the NMOS and PMOS transistors were run and resulted in the simulation based device sizes for the NMOS and PMOS transistors.
5) The values for A and B (effective device resistance) were calculated from the measured propagation delay of the single inverter simulations.

4.4 Fitting Coefficients for Stacked Devices

The simulation-based timing and subsequent calculations for A and B, enable the inverter device sizes to be calculated such that the resulting propagation delay will be $\tau = 32.3ps$. The next half of the method section is intended to extract fitting coefficients for stacked transistors. The fitting coefficients are used to enable the scaling of NMOS and PMOS transistors in a stacked configuration. The stacked device sizes will be generated using a scalar value of the original inverter device sizes.

An NMOS stack of two transistors will drive a load slower than an equally sized single stack NMOS due to the added resistance, capacitance and secondary effects of the stacked transistor. If the stacked transistors are scaled up in size until the propagation delay was matched to the original single stacked transistor delay, the ratio between the stacked NMOS device sizes and the single

NMOS size would be the fitting coefficient. This fitting coefficient can be determined through simulations of varying stack heights until the resulting delays meet the single stack height delay. This approach negates the need for sizing every combinational logic block individually thus allowing the process to be reduced to a simple scaling of devices based on a single analysis of an inverter and three subsequent extractions of scaling coefficients for stacked transistors.

An inverter is used as a template from which circuits of greater complexity can be modeled. A NAND2 (2-input NAND gate), can be sized in a similar manner as the inverter if a scalar value could be found to effectively match the inverter and NAND2 switching behavior. To model a circuit with inverter-like behavior, fitted models are made that reflect the effects of stacked transistors. The goal is to find scalar values that represent the effects of a stacked transistor. Circuit sizing can be performed by finding an inverter to drive a given load, replacing the inverter with the correct logic gate intended to drive that load, and sizing that logic gates' transistors with the scalar values extracted from the following simulations.

The following steps are taken to find the effects of the stacked devices on timing and ultimately extract the scalar values required for each stack to meet inverter like timing:

1) Build a single test-bench to measure the timing of stacked transistors or one, two, three, and four-high stacks.
2) Set the test-bench stimuli as seen in Figure 11.

   a. The source-diffusions of the transistors closest to the supply are connected to supply (gnd and V_{DD} for NMOS and PMOS, respectively).
   b. The gate-terminals for the transistors closest to the supply sources are set to 90% of the effective supply (90%-V_{DD} and 10%-V_{DD} for NMOS and PMOS, respectively).
   c. The gate-terminals of all other transistors are connected to the relative “on” supply (V_{DD} and gnd for NMOS and PMOS, respectively).
   d. The drain-diffusion connections of the devices furthest from the supply are connected to the transient input (to be swept up and down for the NMOS and PMOS stacks, respectively).

3) The series-currents through the stacked transistors are measured and then plotted for each set of stacked device.
4) The current waveform is integrated across the input voltage range to extract effective stack resistance using Ohm’s Law, in Equations 21 and 22.
5) The effective resistive differences between each stack are used to calculate the stack-based fitting coefficients.

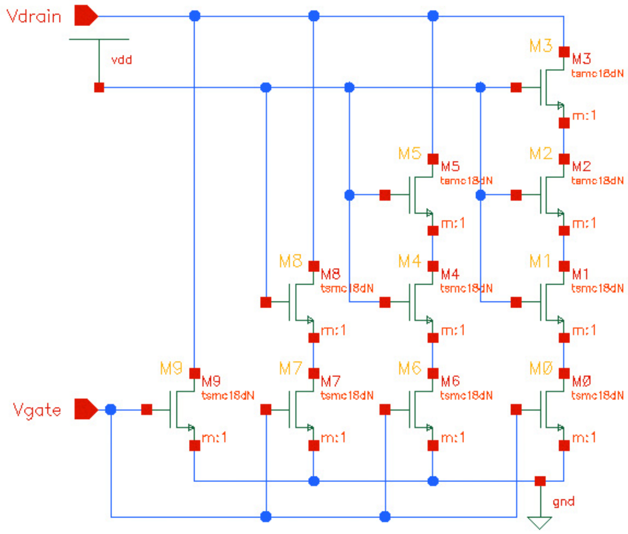


Figure 12. Schematic test-bench: NMOS stacked devices.

The stacked transistor test-bench, shown in Figure 11, is used to simulate and plot one the electrical-current waveform I(N_{SN}) for each stacks. The test-bench controls the voltage across the MOS stacks while measuring the I(N_{SN}). The voltage and I(N_{SN}) are used to calculate the effective resistances, RES(N_{SN}), based on Ohm's Law $(V = IR)$ as shown in Equations 21 and 22. The drain voltage (V_D) was swept (for NMOS from $ground(0) \rightarrow V_{DD}$ and for PMOS from $V_{DD} \rightarrow ground(0)$) resulting in a varying current.

$$RES(N_{SN}) = \int_{\frac{V_{DD}}{2}}^{V_{DD}} \frac{1}{I(N_{SN})} dV_D$$ Equation 20

$$RES(N_{SP}) = \int_{0}^{\frac{V_{DD}}{2}} \frac{1}{I(N_{SP})} dV_D$$ Equation 21

The fitting coefficients can be determined for each of the two, three, and four high stacks of NMOS and PMOS transistors. The stack-fitting coefficients are denoted with "γ." The scaling coefficient for a two-high PMOS-stack (γ_{P2}) represents the relative PMOS device sizes for the two-high stacked transistors relative to the PMOS size in an inverter. The coefficients are calculate using the Equation 23:

$$\gamma = \frac{RES(N_{SP})}{RES(N_{SP} = 1) \cdot N_{SP}}$$ Equation 22

The two-high PMOS, mentioned above, is found to have a γ_{P2} by:

$$\gamma_{P2} = \frac{RES(N_{SP} = 2)}{RES(N_{SP} = 1) * 2}$$ Equation 23

There is one more step to determine the device ratio "R," for the standard circuits of a given architecture. The calculation allows the device sizing to be determined through sizing a single NMOS or PMOS portion of a gate and then applying R to determine the other half of the device sizes. A table is generated to show the relative A and B values for each of the stacked device heights. If a device is complicated (has more than one output path, or multiple device stack heights for either NMOS or PMOS), the worst-case stack is used.

An example sizing for a NAND gate is calculated below using the inverter device sizes and the scaling value for the NMOS stack. A and R are calculated for a two-input NAND (NAND2) using Equations 25, 26, and 27:

$$A_{NAND2} = \frac{W_{N-NAND} \cdot \tau_{PHL}}{\gamma_{N=2} \cdot C_{LOAD} \cdot L_N \cdot (N_{SN} = 2)} \qquad \text{Equation 24}$$

After calculating A_{NAND2}, R_{NAND2} can be calculated:

$$R_{NAND2} = \frac{B}{A \cdot \gamma_{N=2} \cdot (N_{SN} = 2)} \qquad \text{Equation 25}$$

A_{NAND2} and R_{NAND2} can then be used to calculate the value for B_{NAND2} :

$$\frac{R_{NAND2}}{R_{INV}} = \frac{\frac{B}{A \cdot \gamma_{N=2} \cdot (N_{SN} = 2)}}{\frac{B}{A}} \Rightarrow R_{NAND2} = \frac{R_{INV}}{\gamma_{N=2} \cdot (N_{SN} = 2)} \qquad \text{Equation 26}$$

The major steps for the Method are now complete. The process described above will enable users to acquire device sizes for most process technologies with less effort than traditional custom design methods. However, two major simplifications were made to get through the derivation of scaling coefficients faster. These two delay components need to be considered for applications where initial timing accuracy is required to be greater than 90%. These two delay components are:

1) Static Input Slope
2) Static Output Load

4.5 Input Slope Variations

Previous work [1] attempts to analytically “circle back” to close the error margins from the two items mentioned above. To account for the variation in propagation delay due to input slope, the entire calibration process is repeated with one significant change. The “slow” input slope is derived from the use of a complex logic gate, AOI333, driving itself in a chain, similar to the seven-inverter chain before, with worst-case conditions applied. With the input slope determined, the single-inverter test-bench is repeated with only the slope input change. Rather than scaling both the NMOS and PMOS devices in the inverter, to meet the delay target, the NMOS device is held constant and the PMOS device is swept to create a balanced delay. The impact of scaling method has a significant effect on the propagation delay and on the final device ratio.

The input slope variations result in two new, and three total, sets of stacked device scaling coefficients. One set for slow slopes, one set for typical slopes, and the last set of scaling coefficients tailored for fast slopes. The application of the slope-dependent MOS scaling coefficients is based upon the unique timing conditions for each stage of a circuit design. Careful selection is needed to determine when to use the appropriate scaling coefficient, so the final circuit timing will remain within the constraints of the architectural specification.

4.6 Output Load Variation

Output load variations can have a significant impact on propagation delay model's accuracy [1-3,5,7]. The propagation delay model can mitigated the load-dependent impacts by using minimum and maximum (architecturally defined) output loads during calibration. By spanning the range of all potential output loads during calibration, the resulting PDM incorporates all the load-related behaviors thus resulting in a more predictable model [3,6]. The use of three output loads (minimum, average, and maximum) produces even greater accuracy than the required two loads.

The use of a third data point compensates for non-linear behaviors that exist at extreme circuit loading ranges. The three data points provide two discrete linear models that represent the relationship between device size and output load. Further inclusion of output load values, between the minimum and maximum loads, provide greater accuracy with a cost in added effort. Every delay model will require an evaluation, between effort and accuracy, to determine the requirements needed to meet the architectural specification.

4.7 Verification of the Final Model

The last stage of development for a PDM is performance-verification. To ensure the model is capable of producing sufficiently accurate results, a representative “test-circuit” is designed, simulated, and measured. The circuit chosen for verification is crucial to the ultimate success or failure of the PDM. The test-circuit topology must be representative of the typical complexity within a system-design for the test-results to provide a representative solution applicable to the rest of the design.

A 64-bit Kogge-Stone adder represents the typical circuit topology for a small microprocessor [1]. Individual logic-elements are sized using their output loads and input slopes as data-inputs to a PDM. This method allows for the individual MOSFET sizes to be calculated in parallel, rather than working from the output stage backwards. The architectural specification for a circuit defines the circuit’s interconnections and overall timing requirements. These interconnect and timing specifications can translate into slope and load magnitudes. Automation can rapidly improve the rate at which these calculations are performed. Given the regular nature of the design flow, manual calculation should only be performed as an initial PDM-calibration procedure.

The simulation timing results from the Kogge-Stone adder did not match well with the timing calculated from the PDM. The error for some logic stages reaches 60%, and the average error was around 18%. These results were confirmed manually for a small sample group of circuits from the design. Further details of the error source and potential solutions are presented in the Results section.

CHAPTER FIVE
RESULTS

The Results are composed of three sections. The first section is the verification of the method presented in the work by Baum [1]. The second section is the verification of the results presented in the work by Baum [1]. The third section is the results of the improved propagation delay model as applied to discrete and a logic-block level design.

5.1 Verification of Previous Method

Method verification is comprised of re-performing the method presented by Baum and then verifying the timing results against the previous published work [1]. The first step in repeating the PDM calibration is to build an inverter chain with the configuration of a ring as shown in Figure 13. The intermediate nodes are sampled with voltage-probes so each may be measured and plotted separately.

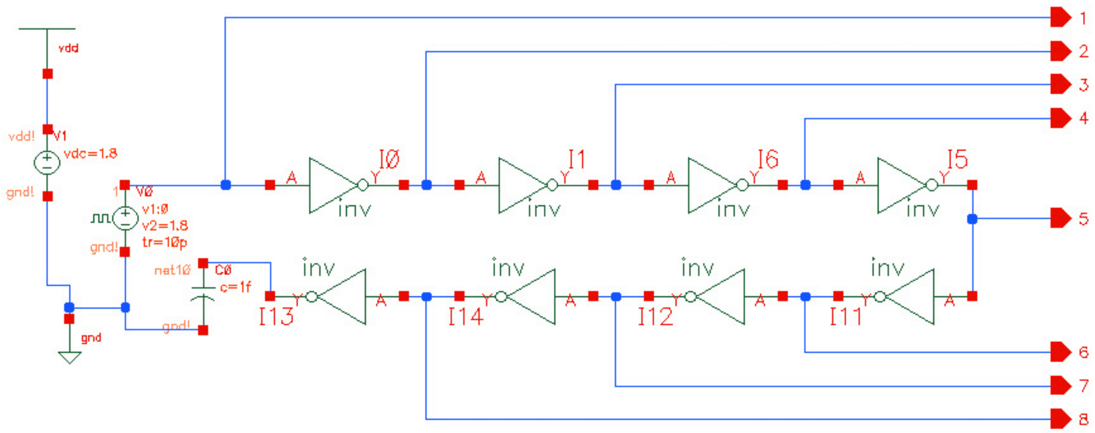
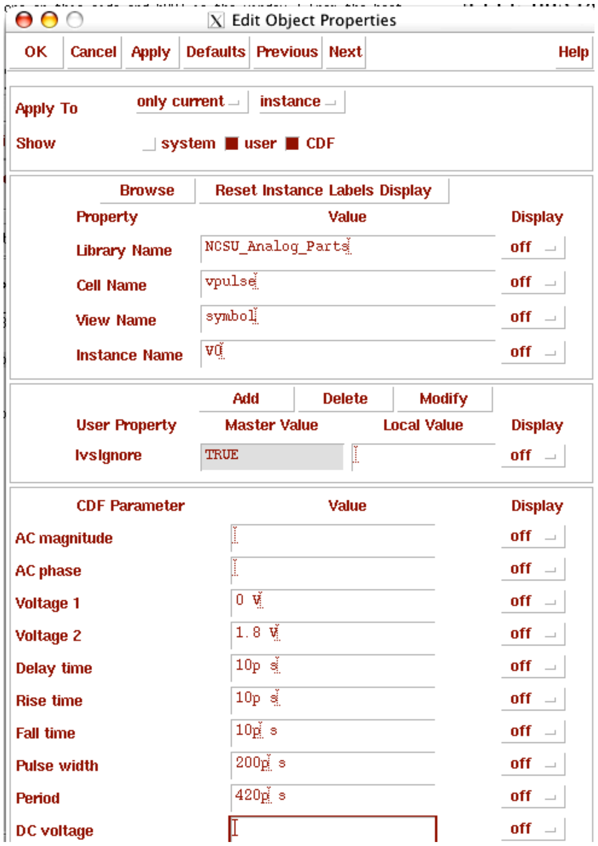


Figure 13. Schematic test-bench of an inverter chain.

The voltage-pulse generator, the right source at the far left of the schematic shown in Figure 13, is set with a slope of 10 ρs for both rising and falling input slopes. The period is set to 400 ρs, with 50% duty-cycle (voltage is at V_{DD} and Ground for equal measures of time). To achieve these conditions, the object properties for the voltage-pulse generator are filled out as shown in Figure 14.



Apply To: only current, instance

Show: □ system ■ user ■ CDF

Property	Value	Display
Library Name	NCSU_Analog_Parts	off
Cell Name	vpulse	off
View Name	symbol	off
Instance Name	V0	off

User Property	Master Value	Local Value	Display
lvsIgnore	TRUE		off

CDF Parameter	Value	Display
AC magnitude		off
AC phase		off
Voltage 1	0 V	off
Voltage 2	1.8 V	off
Delay time	10p s	off
Rise time	10p s	off
Fall time	10p s	off
Pulse width	200p s	off
Period	420p s	off
DC voltage		off

Figure 14. Pulse voltage source setup conditions.

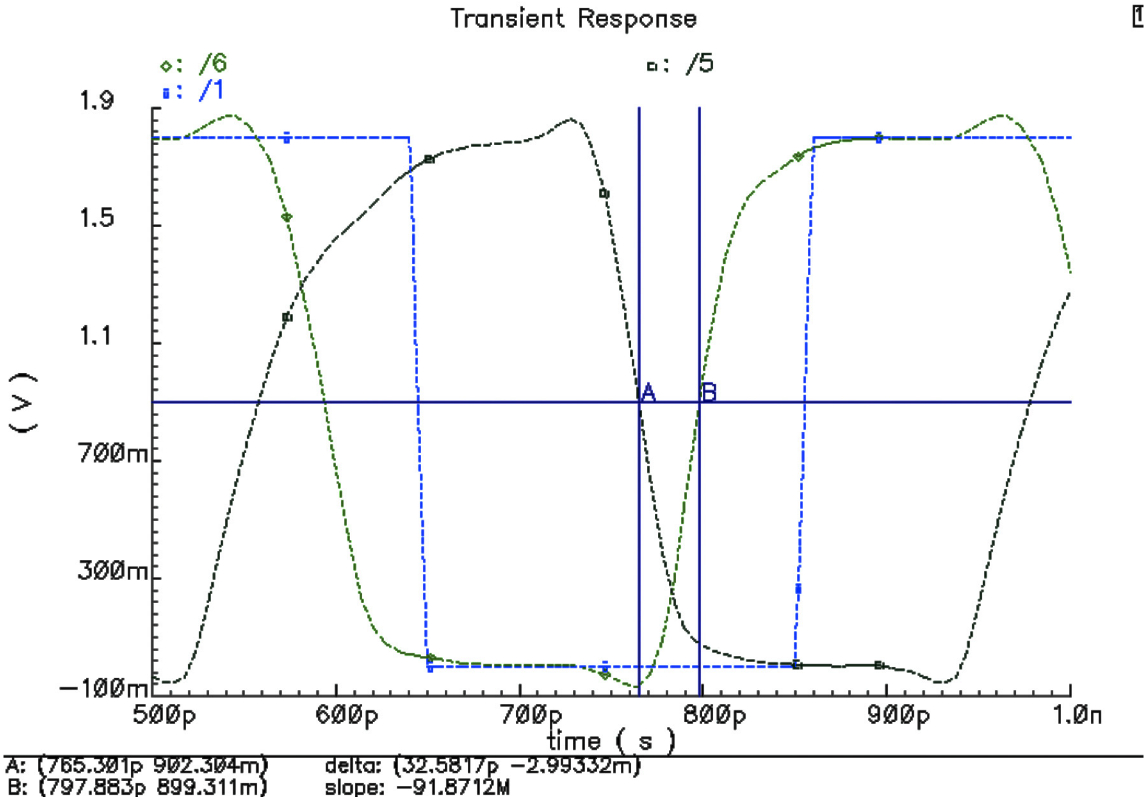


Figure 15. Waveform measurement verifying propagation delay.

The values labeled “delta,” indicate that the measured propagation delay between point-A and point-B is 32.6 ρs, as shown in Figure 15. This measure represents the τ_{PLH} for the sixth inverter of the inverter-chain. The delay is measured as time between the input-transition at 50% of V_{DD}, and the reciprocal output-transition reaching 50% of V_{DD}.

The next calculation is for initial device sizes of the single inverter test-bench. The propagation delay and output load are used as constraints to produce NMOS and PMOS device widths. The delay from the above measurement, 32.6 ρs, and the output load of 7.1fF are used to calculate the initial device sizes for the inverter test-bench as shown in Equations 28 and 29. The output load is set by measuring the input capacitance of the load-inverter and multiplying by a factor of four.

$$W_N = \frac{A \cdot C_{load} \cdot L_N}{\tau_{PHL}} \qquad \text{Equation 27}$$

$$A = \frac{1}{K_{NP} \cdot (V_{DD} - V_{TN})} \left[\frac{2 \cdot V_{TN}}{V_{DD} - V_{TN}} + \ln\left(\frac{4 \cdot (V_{DD} - V_{TN})}{V_{DD}} - 1 \right) \right] \qquad \text{Equation 28}$$

The constants for "A" are listed in Appendix B. The value for the NMOS device width (W$_{N)}$ is then calculated to be $0.27 \mu m$ using Equation 28 and Equation 29. The same process is repeated to calculate the PMOS device width W$_P$ using Equation 30 and Equation 31.

$$W_P = \frac{B \cdot C_{load} \cdot L_P}{\tau_{PLH}} \qquad \text{Equation 29}$$

$$B = \frac{1}{K_{PP} \cdot (V_{DD} - |V_{TP}|)} \left[\frac{2 \cdot |V_{TP}|}{V_{DD} - |V_{TP}|} + \ln\left(\frac{4 \cdot (V_{DD} - |V_{TP}|)}{V_{DD}} - 1 \right) \right] \qquad \text{Equation 30}$$

Following the calculations for W$_N$ and W$_P$ ($0.27 \mu m \,\&\, 0.54 \mu m$-respectively), the single-inverter test-bench can be run. The goal for the single-inverter test-bench is to adjust the NMOS and PMOS device-widths until the target delay of 32.6 ρs is reached. Ideally, the calculations for device sizes, as shown in Equations 28 through 31, would result in a model that is very close to the actual sizes needed. In reality there are simplifications made in the original derivation [2], that place the simulation results and analytical calculations significantly apart.

The test-bench for the single-stage inverter is set up as shown in Figure 16. The device sizes shown are for the final solution but the connectivity and the input stimulus provide an accurate representation of what the inverter test-bench looks like.

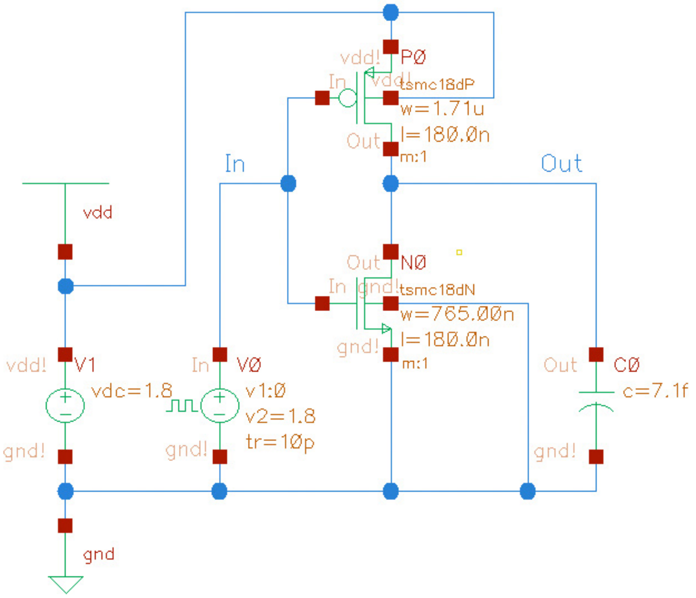


Figure 16. Single inverter test-bench for calculating W_N and W_P.

The NMOS and PMOS device sizes are the result of seven sizing iterations, as shown in Table I. The devices begin with minimum-NMOS ($0.484\mu m$) and with a ratio of R equal to two, the PMOS is ($0.968\mu m$). The delay results for each simulation are compared to the target delay, and a resulting error is calculated. The rising-propagation delay error is used to adjust the PMOS, and the falling delay error used to adjust the NMOS. This process is repeated until the resulting error is less than 1% for both delay arcs. Table I shows the seven steps required to meet the target delay. The final device sizes are ($0.768\mu m$) for the NMOS and ($1.71\mu m$) for the PMOS.

Table I. Results of single inverter test-bench iterations.

Simulation	W_N(cm) Current W_P(cm) Current	t_{PHL} Measured(ps) t_{PHL} Measured(ps)	%Error from target 32.3ps	W_N(cm) Next W_P(cm) Next
1	4.84E-05 9.68E-05	37.8 44.2	17.03 36.84	5.66E-05 1.32E-04
2	5.66E-05 1.32E-04	37.0 36.5	14.55 13.00	6.49E-05 1.50e-04
3	6.49E-05 1.50e-04	35.0 34.4	8.36 6.50	7.03E-5 1.59E-4
4	7.03E-5 1.59E-4	33.9 33.4	4.95 3.41	7.38E-5 1.65E-4
5	7.38E-5 1.65E-4	33.1 33.0	2.48 2.17	7.56E-5 1.68E-4
6	7.56E-5 1.68E-4	32.8 32.7	1.55 1.24	7.68E-5 1.71e-4
7	7.68E-5 1.71e-4	32.6 32.6	0.93 0.93	

The relative effects for stacking transistors are calculated from measurements of a test-bench timing and post-processing of the test-bench data. Four stacks of MOS transistors (NMOS in this example) are setup in the following configuration.

Each stack is configured with the following inputs:

1) Gate input voltage, only for device closest to the power supply (in this case “gnd”), is set to 90% of V_{DD} (1.62V)
2) Gate input voltage for all devices above the bottom stack of one device, are set to full supply voltage V_{DD} (1.8V)
3) Source-connection for all devices at the bottom of the stack are connected to DC-ground (0V)
4) Drain-connection for all device at the top of their individual stacks, are connected to the V_{PULSE}, Input-Voltage Sweep-Device

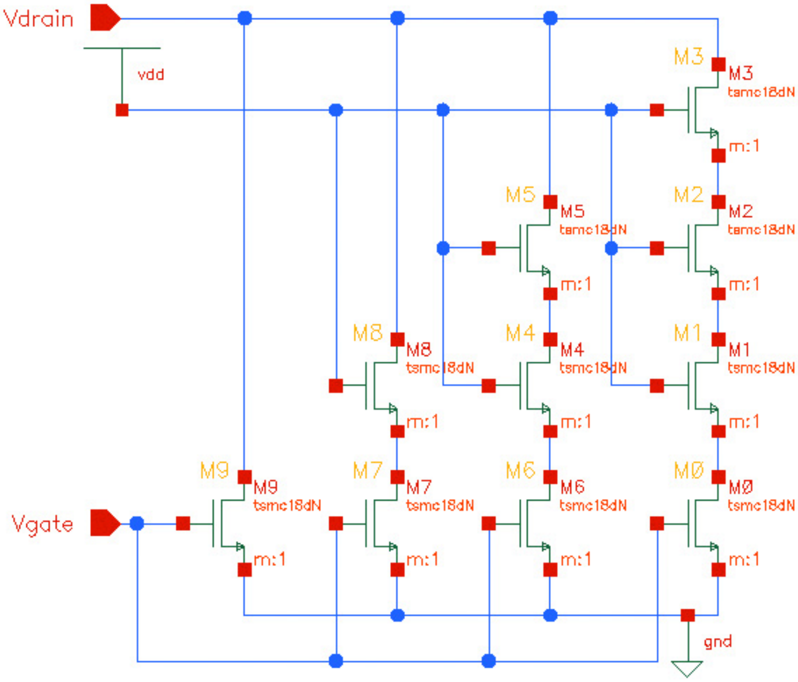


Figure 17. Stacked NMOS device test-bench.

The input (V_{DRAIN}) is swept from $\frac{V_{DD}}{2} \Rightarrow V_{DD}$, while the drain-current is measured (M9, M8, M5, M3 in the diagram). Using Ohm's law we can calculate the "effective-resistance" for each stacked device.

	$V = I \cdot R \Leftrightarrow R = \frac{V}{I} \Leftrightarrow R = \int_{\frac{V_{DD}}{2}}^{V_{DD}} \frac{1}{I_{DRAIN}(N_{SN})} \cdot \delta V_D$	Equation 31

The effective-resistances for the four-stacked NMOS devices are:

1)	One-high	= 5.988E3 $^{\Omega}$
2)	Two-high	= 7.377E3 $^{\Omega}$
3)	Three-high	= 8.743E3 $^{\Omega}$
4)	Four-high	= 9.868E3 $^{\Omega}$

The effective resistances are used to calculate the device scaling factor "γ_N":

1) $\gamma_{(NMOS)N=1} = \frac{5.988k\Omega}{5.988k\Omega} = 1$

2) $\gamma_{(NMOS)N=2} = \frac{7.377k\Omega}{5.988k\Omega} = 1.23$

3) $\gamma_{(NMOS)N=3} = \frac{8.743k\Omega}{5.988k\Omega} = 1.46$

4) $\gamma_{(NMOS)N=4} = \frac{9.868k\Omega}{5.988k\Omega} = 1.65$

The same process for simulation and calculation is repeated for the PMOS devices. The only changes are the relative Voltages used in the test-benches. Rather than 90%-V_{DD} for the gate-voltage (as used for NMOS), the PMOS gate-voltage is 10%-V_{DD}. The rest of the test bench is simply swept in the apposing direction, relative to the PMOS and the following values were found for "γ_P":

1) $\gamma_{(PMOS)N=1} = \frac{5.701k\Omega}{5.701k\Omega} = 1$

2) $\gamma_{(PMOS)N=2} = \frac{7.24k\Omega}{5.701k\Omega} = 1.27$

3) $\gamma_{(PMOS)N=3} = \frac{8.609k\Omega}{5.701k\Omega} = 1.51$

4) $\gamma_{(PMOS)N=4} = \frac{8.837k\Omega}{5.701k\Omega} = 1.77$

The sizing for a circuit can now be implemented based on the known behavior of the standard inverter and the scale factors for equivalent stacked devices. To demonstrate the final application for a sizing of a device, a common logic block will be made !(AB+C).

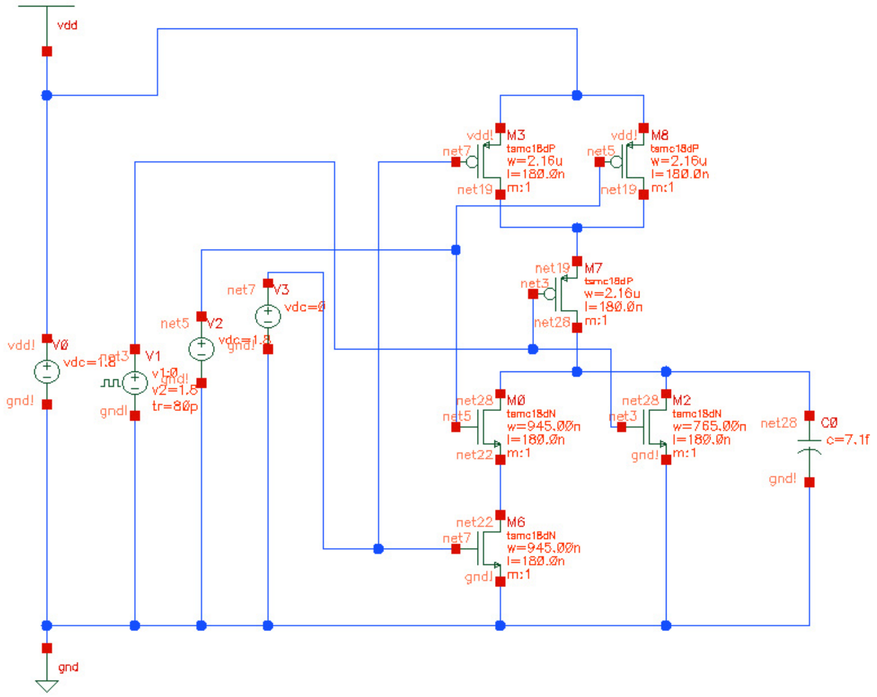


Figure 18. Complex logic gate sizing test-bench !(AB+C).

The device sizing was determined with the following steps.

1) The output load is 7.1fF.
2) The inverter driving the 7.1fF load, did so in 32.6 ρs, with device sizes:
   a. NMOS: 0.765 μm
   b. PMOS: 1.71 μm
3) For the NMOS that is single height, use the same size as template (0.765 μm)
4) For the two-stacked NMOS devices, use the scalar (1.23x) for size (0.945 μm)
5) All the PMOS paths are effectively two-high stacks. Using the scalar for PMOS $(1.71\mu m)\cdot(1.26) = 2.16\mu m$

This concludes the verification of the method presented by Baum. The values found for the initial inverter size, the scalar coefficients ($\gamma_{(PMOS)}$ & $\gamma_{(NMOS)}$) for stacked devices and final sizing iterations will be discussed in further detail in the following Results Verification section. The steps to complete the method verification, of the original work by Baum, were reproducible and followed a logically conclusive path toward the ultimate circuit-sizing goal.

5.2 Verification of Previous Results

The previous-results verification consists of matching the intermediate values in the method presented by Baum, as well at the final device sizes. The intermediate results are comprised of the initial inverter size, the inverter sizes tuning iterations and the stacked device scaling factors. The final results verification is based on the device sizes and circuit timing for the components of the Kogge-Stone adder.

The initial inverter device sizes calculated by Baum, for the NMOS and PMOS transistors, were $0.484\mu m$ and $0.968\mu m$, respectively. This matches the values calculated when reproducing the steps presented by Baum. The initial device sizes were used in an iterative loop to match the target delay, as shown in Table I, and each intermediate value matched as well as the final inverter sizes. The last portion of the intermediate verification steps is calculation of the gamma/stacked-device scaling factors. The gamma values calculated matched the ones presented by Baum in the original thesis [1]

The final modeling of over seven-families of logic, at three-slopes and three-loads was not replicated in its entirety. Each logic family presented by Baum was "spot-checked" at singular condition corners to verify the results were correct. This testing represented approximately 33% reproduction of the total process analysis. The reproduced circuits tested under the same conditions specified by Baum, matched and can be seen in Table II.

5.3 Improved-Method Results

The improved results from using the new methods, detailed in Chapter Four, are displayed in two key examples. The first example shows the device level accuracy improvements of individual logic gates, tested over varying input slopes and output loads. The second example shows the design results for the critical path through a Kogge-Stone adder.

The sizing error for discrete logic devices is shown in Table II. Previous work by Baum had an average error of 13.5%. The improved sizing methodology yields a maximum error of 9.5%. The source of this improvement is further detailed in the Discussion section. The accuracy of the improved method is most significant for the discrete devices with an input slope of 222ps, where the average error drops to 3.9%.

Table II. Discrete logic sizing results.

Device Type	Fanout-Used	Input Slope (ps)	Min Delay (ps)	Max Delay (ps)	Min Dev. width %Error	Max Dev. width %Error	Improved Min Device width %Error	Improved Max Device width %Error
Inverter	FO-1	34	21.6	24.1	-7.4	24.9	-42.0%	-35.3%
	FO-1	222	30	49.8	22.9	44.2	-19.5%	33.7%
	FO-1	410	30	68	10.8	24.6	-19.5%	82.6%
Inverter	FO-4	34	28.6	32.4	-25.4	-11.4	-42.8%	-35.2%
	FO-4	222	42.8	63.6	-20.8	25.3	-14.4%	27.2%
	FO-4	410	47	85	-23.8	23.3	-6.0%	70.0%
NAND2	FO-1	34	33.6	34	-11.6	-8.4	-32.8%	-32.0%
	FO-1	222	51.3	63.5	6.2	20.3	2.6%	27.0%
	FO-1	410	56.6	82.6	3.7	1.8	13.2%	65.2%
NAND2	FO-4	34	46.5	48.5	-28.8	-2.1	-31.6%	-28.7%
	FO-4	222	70.1	66.6	-32.5	24.9	3.1%	-2.1%
	FO-4	410	80.6	105	-32.4	9.1	18.5%	54.4%
NAND3	FO-1	34	41.5	46.5	-14.5	-0.8	-39.0%	-31.6%
	FO-1	222	66.6	74.5	-6.9	22.3	-2.1%	9.6%
	FO-1	410	76.6	95.7	-8.9	-4.8	12.6%	40.7%
NAND3	FO-4	34	55.1	63.1	-26.7	-19.3	-35.8%	-26.5%
	FO-4	222	86.4	91.4	-36.8	13.5	0.7%	6.5%
	FO-4	410	101	118	-34.7	-12.1	17.7%	37.5%
NAND4	FO-1	34	48.8	57.4	-17.8	-3.1	-38.0%	-27.0%
	FO-1	222	80.1	84.2	-20.3	-13.9	1.8%	7.1%
	FO-1	410	93.9	107.5	-15.6	-5.3	19.4%	36.7%
NAND4	FO-4	34	63	76.6	-28.6	-23.3	-35.9%	-22.0%
	FO-4	222	101	101	-40.1	4.2	2.8%	2.8%
	FO-4	410	118.9	129	-38.9	-16	21.0%	31.3%
NOR2	FO-1	34	51.9	52.1	8.5	25.1	-28.0%	-27.7%
	FO-1	222	72.2	78.1	20.8	34.1	0.2%	8.4%
	FO-1	410	82.3	95.8	20.6	10.5	14.2%	32.9%
NOR2	FO-4	34	68.3	70.6	-11.5	18.2	-26.2%	-23.7%
	FO-4	222	93.9	97.3	-12.8	33.1	1.5%	5.2%
	FO-4	410	107	118	-15.5	40.1	15.7%	27.5%
NOR3	FO-1	34	83.5	94.5	13.5	18.2	-25.2%	-15.4%
	FO-1	222	104	123	14.3	13	-6.9%	10.1%
	FO-1	410	111	154	20.7	15	-0.6%	37.9%
NOR3	FO-4	34	109	119	-9.6	-3.9	-20.9%	-13.7%
	FO-4	222	131	147	-11	2.3	-5.0%	6.7%
	FO-4	410	141	180	-14.3	29.4	2.3%	30.6%
NOR4	FO-1	34	123	129	17.3	17.6	-17.5%	-13.5%
	FO-1	222	144	156	8.1	-9.3	-3.5%	4.6%
	FO-1	410	152	191	22.2	16.2	1.9%	28.0%
NOR4	FO-4	34	156	159	-12.4	-6.1	-13.7%	-12.1%
	FO-4	222	179	185	-9.2	4.9	-1.0%	2.3%
	FO-4	410	188	218	-18.1	-2	4.0%	20.6%
Mean							-8.4%	9.5%

The final portion of the results consists of the design of a Kogge-Stone adder. The most-critical path through the Kogge-Stone adder was selected for the sizing example that follows. The simulation conditions, used by Baum in the previous work, were duplicated to provide the most accurate comparison of results to future research and verification.

The critical path through the Kogge-Stone adder consists of six stages. The six stages, shown in the Figure 19, consist of xor2 (shown as a red circle), four A+BC complex blocks (shown in green rectangles), and one sum gate (shown in a yellow trapezoid). The critical path has been highlighted in Figure 19, while the remaining paths for the Kogge-Stone adder were omitted for visual-clarity.

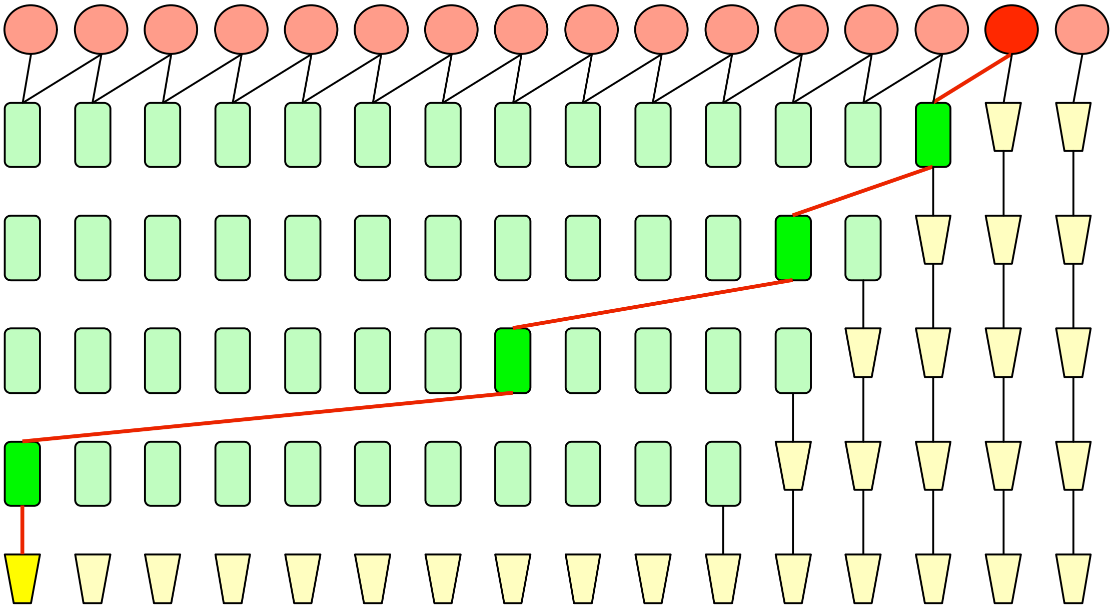


Figure 19. Kogge-Stone adder critical path.

Within the Kogge-Stone adder-stages the individual logic functions are comprised of different discrete logic elements. The elements and their design sizes are listed below in Table III.

Table III. Kogge-Stone critical path design results.

Cell	Sub-Cell	CG+CINT (fF)	Propagation Delay (ps)	A (ohm)	R	N	M	NSN	NSP	W_N (um)	W_P (um)
XOR2	INV (sum_out)	14.3	50	17546	2.2	1	1	1	1	0.97	2.14
	XNOR (sum)	6.22	100	10009	2.43	4	4	2	2	1.12	1.93
	INV (sum_in)	4.2	50	17546	2.2	1	1	1	1	0.27	0.27
A+BC	INV (black)	19.95	50	17546	2.2	1	1	1	1	1.35	2.98
	AOI (black)	11.95	100	11089	2.51	3	2.5	2	2	1.42	3.56
A+BC	INV (black)	29.4	50	17546	2.2	1	1	1	1	1.99	4.38
	AOI (black)	17.6	100	11089	2.51	3	2.5	2	2	2.09	5.24
A+BC	INV (black)	27.8	50	17546	2.2	1	1	1	1	1.88	4.13
	AOI (black)	16.6	100	11089	2.51	3	2.5	2	2	1.97	4.94
A+BC	INV (black)	23	50	17546	2.2	1	1	1	1	1.55	3.42
	AOI (black)	13.7	100	11089	2.51	3	2.5	2	2	1.62	4.08
XOR2	INV (sum_out)	11.4	50	17546	2.2	1	1	1	1	0.77	1.7
	XNOR (sum)	6.8	100	10009	2.43	4	4	2	2	1.22	2.98
	INV (sum_in)	11.5	50	17546	2.2	1	1	1	1	0.79	1.738

The conditions used for the Kogge-Stone adder design were taken from the previous work presented by Baum [1]. These conditions include the interconnect capacitance for the (A+BC) logic of 15.7fF [1]. The output load was also taken from the earlier work from Baum and was set at 14.3fF.

The Kogge-Stone adder critical path was intended to take 1000ps to propagate. The improved design resulted in a maximum delay difference of -4.6% (956ps) and a minimum delay difference of 1.5% (1015ps). The internal stage delays had a maximum variation of -18% (188ps-xnor) and a minimum stage variation of -3% (48.5ps-inverter).

CHAPTER SIX
DISCUSSION

This discussion will focus on the assumptions made in previous PDM papers [1-34], and the impact those assumptions have on propagation delay results. The concept of developing a process-independent PDM calibration methodology is uncommon among PDM publications. The work by (Baum, Jeremy. Calibration Method of an Analytical Propagation Delay Model. San Jose: SJSU, 2007.), presents a unique method of calibrating analytical propagation delay models, bounded only by device manufacturing technology. The intent is to provide a method for calibrating a standard propagation delay model for any given manufacturing technology. The broad range of application, constrained only by manufacturing technology, comes at a cost to accuracy as demonstrated in the Results section. The following section discusses the assumptions made in the development of the original propagation delay model [1], and the benefits or penalties those assumptions have on the model.

6.1 Inverter Chain Fanout Selection

The initial step for calibrating a PDM for a given technology began with an inverter chain. The target stage delay was selected from the fasted stable stage from that inverter chain. The inverter chain is setup to only drive sequential stages of the same transistor size and capacitive load. A device with an output load equal to it’s input capacitance, will result in exceptionally fast propagation delays that will not represent typical circuit timing behavior [2]. The circuit architecture for a target design would be a valuable contribution to the initial step of finding a target delay. If a design has an average fanout of five then the target stage delay, based on a fanout of one, will be unreasonably fast. Device sizing must grow disproportionately large to meet the unrealistic delay expectations that were measured in the initial inverter chain test-bench.

The accuracy of a PDM is dependent upon a practical target stage delay. Error caused by incorrect assumptions can be seen in the following case. Initial simulations show the NMOS and PMOS device-size errors as 17% and 36.8% respectively as shown in Appendix C. These errors are caused from using a fanout of one to generate the target propagation delay, rather than using the fanout of two or three, typical to the Kogge-Stone adder architecture. The result of using an intermediate load with fanout of two, saves significantly on the total number of simulations required by relaxing the target delay. The total number of required simulations, to determine the correct inverter device-sizes, can be reduced by 14%, or from eight simulations to seven simulations, as seen in Appendix C.

6.2 Single Inverter Test-bench

In the single inverter test-bench, the output load is held constant while the input MOSFET devices were re-sized to meet the target propagation delay. The stated intention of the single inverter test-bench was to calibrate inverter device sizes to drive a fanout of four, while meeting the target delay [1]. The initial device sizes of the NMOS and PMOS were increased by 63% and 56% respectively to meet the target propagation delay. The increases in the inverter's device sizes were applied without updating the inverter's output load, resulting in an output load much closer to an equivalent fanout of three. Nowhere is this mentioned in the published work from Baum [1], and this likely contributes to some of the 60% maximum error found between the calculated delays relative to the simulated delay.

6.3 Symmetric Propagation Delay

Symmetric propagation delay is the timing method used by Baum for the modeling and calibration for all MOSFET analysis [1]. Logic polarity becomes irrelevant when designing with symmetric timing delay because rising and falling transitions are uniform. The vast majority of VLSI designs are focused on one methodology for delay minimum average delay (MAD) [7]. MAD dominates VLSI design methodologies because most CMOS digital-logic architectures today are comprised of twelve to twenty stages [7]. The polarity is irrelevant in standard CMOS designs having more than eight stages thus the method of minimum average delay will produce the faster solution than symmetric propagation delay.

The extra device size required for a logic circuit to have symmetric propagation delay, ranges from 4% to 7%, depending on the semiconductor manufacturing process. The extra MOS size can be viewed as potential timing improvements (by adjusting the ratio without changing total device size) with no added capacitive load. To clarify the benefit of changing the device ratio, the following test was performed:

1) A symmetric delay inverter was built with:
    a. 32.3 ρs (1-picosecond = $1 \cdot 10^{-12}$ seconds) rising and falling delays.
    b. PMOS device size of 1.71 μm.
    c. NMOS device size of 0.765 μm.
2) The ratio between the PMOS and NMOS transistors was varied around a fixed total device size of 2.475 μm.
3) The PMOS transistor size was reduced to increase the NMOS transistor size resulting in 1.43 μm PMOS and 1.045 μm NMOS.
4) The final timing change went from 32.3 ρs for rising and falling delays, to 32.8 ρs to 28.8 ρs for the PMOS and NMOS transistors, respectively. The average delay decreased from 32.3 ρs to 30.8 ρs.

An improvement of 4.6% (average delay) was achieved using the minimum average delay transistor sizing technique. The most important aspect of the improved timing is the neutral effect to capacitance and power. A device driving the new inverter sees no capacitive change (NMOS and PMOS gate capacitance per unit length are identical) because the total transistor size remains constant. Slope degradation is the one drawback that results from the new device ratio. The inverter's rising output slope (controlled by PMOS device size) may reduce by 20%. Slope degradation for the rising transition is typically negated in subsequent circuit stages. The polarity is likely to invert in subsequent stages where the improvement benefit from the slope improvements gained in the NMOS transistor during the minimum average delay

6.4 Input Slope and Output Load Modeling

Calibration for alternate input-slope and output-load combinations was performed at the end of the calibration for a single slope and load combination. Modeling the input slope effects on propagation delay, in conjunction with modeling the output load effects, reduces the total number of simulation required. The simplification of modeling comes at a cost to the final PDM accuracy.

Experimental results agree with the analytical model when variation of a singular element is performed, either slope rate or load magnitude. The error is doubled when both are scaled simultaneously. This means that x-percent error from slope variation and y-percent error from load variation result in $2 \cdot x \cdot y$, or twice the error of the individual variations. This behavior is sufficient reason for a continuing evaluation of the methods presented by Baum [1]. Had the model constraints been applied better by using the Kogge-Stone adder topology to drive all the calibration boundary conditions, the results would yield significantly less error. The counter argument is that constraining any model enough can make it 100% accurate for 0.001% of applications [7].

Boundary conditions are an extension of the previous topic, using circuit architecture to guide circuit-testing conditions. The selection of boundary conditions can be more important that the equations they govern. The balance lay between two extreme model results:

1) Overly constrained, highly accurate and not widely applicable or usable.
2) Under constrained, very inaccurate but widely applicable.

The correct balance between these two extremes becomes evident with experience. The ambitious nature of the recently educated is tempered with the conservative realism of a seasoned veteran. There is no perfect solution to determining boundaries between the two. Propagation delay in digital CMOS logic, is a field with tremendous amounts of research available. Such availability makes design niches much more relevant. Broad generalizations within this field can be countered with countless citations showing contrary results [8,9]. It is for this reason that the scope of Baum's work needs to be reduced, and the amount of analysis be increased to achieve results with much smaller margins of error.

Well-defined boundary conditions [9] serve as a strong example to the effectiveness of stringent constraints. The topic of large SRAM array's won't apply to many readers directly, but the resulting error of <5% will grab any engineers attention. Juxtapose that to the model by Baum [1], which provides a propagation delay model for practically any CMOS digital design, but the error is typically 15% and sometimes as high as 60%. The significant magnitude of error mistakenly gives a sense that the method is wrong, rather than the method is being too loosely constrained. Constraining the application of the model by Baum would reduce the error and in turn become more likely to be cited as applicable peer research.

The proof of concept presented by Baum was performed on devices of significantly greater complexity than the elemental logic used in the calibration of the propagation delay model [1]. A Kogge-Stone adder, though it uses some basic gates, is comprised on many multi-stage, complex AOI-logic. Initial calibration was performed on devices with singular current paths (NAND, NOR, Inverter). The additional capacitance, present on intermediate circuit nodes, was not calibrated for in the original propagation delay model. Had the initial calibration procedure been performed under boundary conditions derived from the Kogge-Stone adder, the resulting accuracy would have been much more accurate, as presented in the Results section that follows.

6.5 Improved Methods

One method improvement consists of making one more round of data calibration, using the known result-errors, and scaling the results. This is sometimes called “back-fitting” data [3,5,6], and is conceptually similar to the intermediate model calibration steps. The goal of back fitting is to use the error results to adjust the model to ultimately improve result-accuracy. Other, more analytical methods, involve re-evaluation of initial assumptions to address the root-sources that cause the errors. Both of these methods are detailed in the Discussion section, and implemented below, in the final section of the results (Improved Methods).

The first stage of the propagation delay model calibration method involves building an inverter chain and extracting the fastest possible single propagation delay. The assumptions are:

1) Inverter Chain Method
    a. Apply an input slope, 10 ρs, to the first inverter of the inverter chain.
    b. Check the propagation delay of each inverter down the chain, until the delay stabilizes (less than 1% change from previous stage).
    c. Use the stable delay as the target delay for all the models and analysis for the rest of the calibration process.
2) Inverter Chain Sizing:
    a. Small inverter chain:

        i. Minimum process allowed device size for the NMOS.
        ii. Double the NMOS size to match the generic ratio of PMOS to NMOS of two.
    b. Large inverter chain:
        i. Scale the NMOS of the small chain inverter, up, by a factor of twenty-five.
        ii. Scale the PMOS of the small chain inverter, up, by a factor of twenty-five.

3) Inverter Chain Constraints
    a. Set up the chain so each inverter is only driving an effective fanout of one.
    b. Do not add parasitic capacitance wires connecting the inverter chain.

The last constraint, setting each stage to drive fanout of one, will have a tremendous impact on the target delay number, and the resulting number of simulations required to calibrate the process. If the propagation delay was derived from an inverter chain driving a fanout of two, the amount of simulation effort can be reduced by up to 14.3%. Table III shows the results from the method used by Baum [1]. The seven simulations show the steps of scaling an inverter to drive a fanout of four, with the propagation delay derived from the inverter-chain. Table IV shows the results of the modified method for the inverter-chain simulation. The modification involves simulation of an inverter chain, with a fanout of two, rather than the original fanout of one. All other conditions and assumptions, aside from the output load, are applied.

Table IV. Results of single inverter test-bench iterations.

Simulation	W_N(cm) Current W_P(cm) Current	t_{PHL} Measured(ps) t_{PHL} Measured(ps)	%Error from target 32.3ps	W_N(cm) Next W_P(cm) Next
1	4.84E-05 9.68E-05	37.8 44.2	17.03 36.84	5.66E-05 1.32E-04
2	5.66E-05 1.32E-04	37.0 36.5	14.55 13.00	6.49E-05 1.50e-04
3	6.49E-05 1.50e-04	35.0 34.4	8.36 6.50	7.03E-5 1.59E-4
4	7.03E-5 1.59E-4	33.9 33.4	4.95 3.41	7.38E-5 1.65E-4
5	7.38E-5 1.65E-4	33.1 33.0	2.48 2.17	7.56E-5 1.68E-4
6	7.56E-5 1.68E-4	32.8 32.7	1.55 1.24	7.68E-5 1.71e-4
7	7.68E-5 1.71e-4	32.6 32.6	0.93 0.93	

Table V. Improved results of single inverter test-bench iterations.

Simulation	W_N(cm) Current W_P(cm) Current	t_{PHL} Measured(ps) t_{PHL} Measured(ps)	%Error from target 39.3ps	W_N(cm) Next W_P(cm) Next
1	3.68E-04 7.36E-05	42.47 45.10	10.05 16.84	4.05E-05 8.60E-05
2	4.05E-05 8.60E-05	41.33 41.73	7.08 8.11	4.33E-05 9.29E-05
3	4.33E-05 9.29E-05	40.28 40.07	4.37 3.80	4.52E-05 9.64E-05
4	4.52E-05 9.64E-05	39.34 39.07	1.92 1.21	4.61E-05 9.75E-05
5	4.61E-05 9.75E-05	39.02 38.97	1.04 0.97	4.66E-05 9.84E-05
6	4.50E-05 9.90E-05			
7				

The 14.3% improvement over the original method presented by Baum [1], is achieved through performing one less simulation. The stage reduction is the direct result of the relaxed target delay (39.3 ρs). The elimination of one

calibration step is attributed to the relaxation of the target propagation delay and the closer relative magnitude between a fanout of two and a fanout of four.

The saving of 15% effort for characterization would not be beneficial if it came with any further penalty to the model propagation error. The entire method was completed, through to the derivation of ($\gamma_{(NMOS)N=1}, \gamma_{(NMOS)N=2}, \gamma_{(NMOS)N=3}, \gamma_{(NMOS)N=4}$) and ($\gamma_{(PMOS)N=1}, \gamma_{(PMOS)N=2}, \gamma_{(PMOS)N=3}, \gamma_{(PMOS)N=4}$). The models for the Inverter, NAND2 and the NOR3, were all calculated and simulated, resulting in an error within plus or minus 2% of the error presented by Baum, seen in Tables III and Table IV.

The aforementioned analytical modification demonstrates that significant reduction in effort is possible with relatively small adjustments to the initial assumptions by Baum [1]. The effort and accuracy can be exchanged throughout the model calibration process. Taking extra time to isolate and understand each element of the method allows for fine-tuning of the individual steps. This fine-tuning leads to increasingly accurate results.

The following example demonstrates the accuracy improvement potential from isolating one model-element, simulating different variations to understand the element's behavior, and the result improvement found from changing the initial method.

Effective resistance is an area for potential improvement and modification within the context of PDM calibration. The existing method [1] of comparing input voltage and output current for various stacks of MOS transistors, has many potential areas for improvement. Digital CMOS architecture shows that the higher the stacked devices, the greater the internal load from the complimentary devices' diffusions. A NAND2 (NMOS stack of two-devices) has two parallel PMOS devices attached to the output path, and NAND3 has three parallel devices. Therefore the load will not only increase the resistance for taller stacks but it will increase the capacitance too. The stacked-device test-bench only accounts for resistive increases and negates those from the added cap of the complimentary MOS devices.

There are many ways to incorporate realistic device impacts, into the stacked test-bench. To degrade the current path for the stacked devices, one or more of the stacked MOS transistors can be turned partially on, to impede the current path. This is similar to the singular MOS device gate voltage being applied in the original calibration technique from Baum [1]. The main problem with limiting the transistors current is the extra impedance, which behaves like a resistor rather than as a capacitor. Such constraints can be accurate over very small operating voltages, but fail to emulate the capacitive behavior needed over the full operational voltage range.

The PDM calibration and results, presented by Baum [1], were repeated and verified using the TSMC0.18 manufacturing process file as seen in Appendix A. The final PDM [1] was an improvement over the delay models presented by Kang [2], but delay errors were still as high as 60%. To improve the accuracy and provide a more intuitive methodology, the individual calibration stages were each characterized to clarify their contribution to the final PDM accuracy. The three most influential improvements were presented and provide an average accuracy improvement of 14.6% over the methods presented by Baum [1]. Any future work can be added as a fourth level of improvement to the foundation developed herein.

6.6 Analysis of Previous Results

The error results from the previous work [1] have some trends that are important to understand. Trends within the data show where the model is more accurate and where it is less accurate. The scope of application for the work by Baum is so vast, that it cannot accurately account for all the infinite conditions and design solutions possible. The TSMC0.18 μm process was put through the method-steps demonstrated in the previous pages. The results are as follows:

1) All devices with max-sized devices (on the order of twenty-five times the min device size) have:
    a. Average Error: 5.66% (This indicates the simulation was 5.66% faster than the analytical propagation delay model and fitting coefficients predicted, and the devices need to be decreased in size to remedy).
    b. Standard Deviation: of 16.58% (This indicates that even though the average error was small, the amount of variation was spread significantly wide).
2) All devices with the min-sized devices have:
    a. Average Error: -36.9% (This indicates the simulation was 36.9% slower than the analytical propagation delay model and fitting coefficients predicted, and the devices need to be increased in size to remedy).

b. Standard Deviation: of 8.15% (This indicates that even though the average error was large, the amount of variation was spread over a relatively narrow range).

Table VI. Standard cell delay calibration and error of TSMC0.18.

Device Type	Fanout-Used	Input Slope (ps)	Min Delay Arc (ps)	Max Delay Arc (ps)	Min Device Width %Error	Max Device Width %Error	Min Delay Arc: Gate Voltage of 86% VDD (Rather than 90%)
Inverter	FO-1	40	21	43	-29.4	23.3	-7.4
	FO-1	320	40	74	-25	37.9	22.9
	FO-1	600	51.3	90	-35	18.1	10.8
Inverter	FO-4	40	25.3	70	-40.8	13	-25.4
	FO-4	320	55	120	-46.5	26.2	-20.8
	FO-4	600	68.8	140	-48.	9.3	-23.8
NAND2	FO-1	40	33.7	69	-30.3	20.5	-11.6
	FO-1	320	56	100	-31.6	13.6	6.2
	FO-1	600	83.7	120	-38.1	5.9	3.7
NAND2	FO-4	40	39.8	125	-38.5	15.6	-28.8
	FO-4	320	70	150	-44.3	21.2	-32.5
	FO-4	600	101	160	-49.2	-5	-32.4
NAND3	FO-1	40	52	95	-33.5	14.2	-14.5
	FO-1	320	81.9	120	-36.3	15.6	-6.9
	FO-1	600	118	150	-40.2	-8.1	-8.9
NAND3	FO-4	40	58	170	-39.5	11.6	-26.7
	FO-4	320	93	180	-44.1	8.8	-36.8
	FO-4	600	135	160	-47.6	-13.1	-34.7
NAND4	FO-1	40	73.7	130	-35.5	8.7	-17.8
	FO-1	320	112	160	-38.1	1.8	-20.3
	FO-1	600	155	170	-42.1	-16.4	-15.6
NAND4	FO-4	40	79.8	220	-40.2	7.3	-28.6
	FO-4	320	123	180	-47.4	-2.9	-40.1
	FO-4	600	172	172	-30.3	-20	-38.9
NOR2	FO-1	40	47	73	-23.1	25	8.5
	FO-1	320	80	114	-20.4	36.1	20.8
	FO-1	600	94	140	-29.4	25.2	20.6
NOR2	FO-4	40	54	120	-35	17.6	-11.5
	FO-4	320	90	170	-40.8	23	-12.8
	FO-4	600	106	175	-47	14.3	-15.5
NOR3	FO-1	40	83.5	110	-23	18	13.5
	FO-1	320	113	155	-24.8	-7.6	14.3
	FO-1	600	133	180	-32.8	-12.6	20.7
NOR3	FO-4	40	98	170	-31	13.9	-9.6
	FO-4	320	133	200	-41.8	-9.8	-11
	FO-4	600	155	230	-47.3	-15	-14.3
NOR4	FO-1	40	137	163	-24.7	-11.9	17.3
	FO-1	320	177	200	-34.2	-16.7	8.1
	FO-1	600	195	195	-38.7	-19.2	22.2
NOR4	FO-4	40	156	230	-28.9	-11.8	-12.4
	FO-4	320	202	230	-44.3	-17.7	-9.2
	FO-4	600	221	221	-49.5	-20	-18.1
Column Average				-36.88	5.66	-13.5	
Standard Deviation			8.15	16.58	18.7		

Table VI shows the results for the analytical solution and final modeling of seven of the most common CMOS digital integrated circuits. The highlighted areas are key results that represent trends found in the original propagation delay model [1]. The purpose for identifying these trends is to determine if any modifications can be applied to the original method, with the goal of improving the overall accuracy, as well as leave definitive areas where future work could be effectively focused.

The "Min-Delay Arcs" shown in Table VI, average much higher than any other modeling corner. This increased average indicates that across all families the method for calculating propagation delay is not resulting in devices large enough to meet the target delay of 32.6 ρs. The only calculations that meet or exceed the timing requirements, are the inverter-logic, with minimum input slope, and two different output loads. The two results highlighted in yellow represent this distinguished and unique accuracy.

Minimum device-width error demonstrates the combined effect of scaling input slope and output load to the farthest corners of the PDM model. Every result that is at the maximum-scaled slope shows a magnitude of error greater than any of the entries within the same sub-category, are highlighted in pink as shown in Table II. Another interpretation would be that small devices with large input slopes are exceptionally inaccurate. This behavior demonstrates a clear correlation between error and input-conditions as shown in the far-right column of Table VI.

“Max Device-Width Error” shows no obvious patterns like that found in the “Min Device-Width Errors.” The span of the max device error is vast and non-uniform, ranging from -20% to +37%. The standard deviation is another way to view the erratic variation found between the propagation delay model and the simulation results. The CMOS logic-cells with the smallest error, are commonly made of intermediate complexity (NAND2 and NOR2), and the largest errors are correlate to cells with extremes in logic complexity, either most simple or most complex (Inverter, NAND4, NOR4).

The error between the calibrated-propagation model from Baum and the simulation-results shown in Table II are significant. The error does show unique behavior among each specific testing condition and knowing which conditions will aggravate which types of error can provide a starting point for rectification.

6.7 Results Conclusion

The final results show the improvement to the work presented by Baum [1] in both discrete and block level circuit designs. Accuracy improves over larger designs as alternating errors average out to smaller total effect. The methods detailed in the Method for Calibration section provide an improved procedure and greater accuracy in results than the previous work.

BIBLIOGRAPHY

[1] J. Baum, Calibration Method of an Analytical Propagation Delay Model, Thesis TK 2007 .B886, (San José State University, San José, CA, 2007), pp. 1-78.

[2] M. Kang and Y. Leblebici. CMOS Digital Integrated Circuits, (McGraw-Hill, New York, NY, 2003), pp. 83-270.I. Sutherland, B. Sproull and D. Harris,

[3] Logical Effort: Designing Fast CMOS Circuits, (Morgan Kaufmann Publishers, San Francisco, CA, 1999), pp 34-132.

[4] H. Chow and W. Feng, "*Model for Propagation Delay Evaluation of CMOS Inverter Including Input Slope Effects for Timing Verification*," Electronics Letters, **28**, 12, pp.1159-1160, (1992).

[5] V. Gerousis, N. Phan and D. Weaver, "*New Delay Model for* $0.5\mu m$ *CMOS ASIC*," in Proc. 6th Annual IEEE Int. ASIC Conf. and Exhibit, (Rochester, NY, 1993), pp. 511-514.

[6] J. Rossello and J. Segura, "*Simple and Accurate Propagation Delay Model for Submicron CMOS Gates Based on Charge Analysis*," Electronics Letters, **38**, 15, pp. 772-774, (2002).

[7] N. Weste and D. Harris, CMOS VLSI Design: A Circuit and Systems Perspective, (Pearson Education Inc., Boston, MA, 2005), pp. 152-332.

[8] G. Palumbo and M. Poli, "*Propagation Delay Model of a Current Driven RC Chain for an Optimized Design*," IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications, **50**, 4, (2003).

[9] A. Venkatapathi, N. Rayapati and B. Kaminska, "*Interconnect Propagation Delay Modeling and Validation for the 16-MB CMOS SRAM Chip*," IEEE Transactions on Components, Packaging, and Manufacturing Technology-Part B, **19**, 3, (1996).

[10] J. Rossello and J. Segura, "*An Analytical Charge-Based Compact Delay Model for Submicrometer CMOS Inverters*," IEEE Trans. Circuit and Systems, vol. **51**, pp. 1301-1311, (2004).

[11] L. Bisdounis, S. Nikolaidis, O. Koufopavlou and C. Goutis, "*Analytical Transient Response and Propagation Delay Evaluation of the CMOS Inverter for Short Channel Devices*," IEEE Journal of Solid State Circuits, **33**, pp. 302-306, (1998).

[12] S. Nikolaidis, A. Chatzigeorgiou and E. Kyriakis-Bitzaros, "*Delay and Power Estimation for CMOS Inverter Driving RC Interconnect Loads*," in Proc. 1998 IEEE Int. Symposium Circuits and Systems, (Monterey, CA, 1998), pp. 368-371.

[13] D. Burdia, G. Grigore and C. Ionascu, "*Delay and Short-Circuit Power Expressions Characterizing a CMOS Inverter Driving Resistive Interconnect*," in Int. Symposium Circuits Signals and Systems, (New York, NY, 2003), pp. 597-600, 2003.

[14] K. Chen, C. Hu. P, Fang and A. Gupta, "*Experimental Confirmation of an Accurate CMOS Gate Delay Model for Gate Oxide and Voltage Scaling*," IEEE Electron Device Letters, **18**, pp. 275-277, (1997).

[15] P. Maurine, N. Azemard and D. Auvergne, "*General Representation of CMOS Structures Transition Time for Timing Library Representation*," Electronics Letters, **38**, pp. 175-177, (2002).

[16] J. Cardell, "CSC 270 Homework 1: Basic Circuits," *maven.smith.edu*, 3 March 2009 http://maven.smith.edu/~jcardell/courses/CSC270/hw/HW1soln.pdf

[17] L. Bisdounis, O. Koufopavlou and S. Nikolaidis, "*Modeling Output Waveform and Propagation Delay of a CMOS Inverter in the Submicron Range*," IEEE Circuits, Devices and Systems, **145**, pp. 402-408, (1998).

[18] L. Bisdounis, S. Nikolaidis, O. Koufopavlou and C. Goutis, "*Switching Response Modeling of the CMOS Inverter for Sub-Micron Devices*," in Proc. Design, Automation and Test in Europe, (Paris, France, 1998), pp. 729-735.

[19] K. Tang and E. Freidman, "*Transient Analysis of a CMOS Inverter Driving Resistive Interconnect*," in Proc. 2000 IEEE Int. Symposium Circuits and Systems, (Geneva, Switzerland, 2000), pp. 269-272.

[20] Y. Wang and C. Wu, "*Analysis and Modeling of Initial Delay Time and It's Impact on Propagation Delay of VMOS Logic Gates*," IEEE Proc. Circuits, Devices and Systems, **136**, pp. 245-254, (1989).

[21] N. Hedenstierna and K. Jeppson, "*CMOS Circuit Speed and Buffer Optimization,"* IEEE Trans. Computer-Aided Design, **6**, pp. 270-281, (1987).

[22] D. Etiemble, V. Adeline, N. Duyet and J. C. Ballegeer, "*Micro-Computer Oriented Algorithms for Delay Evaluation of MOS Gates*," in 21st Conference Design Automation, (1984), pp. 358-364.

[23] J. Andre, J. Teixeira, I. Teixeira, J. Buxo and M. Bafleur, "*Propagation Delay Modeling of MOS Digital Networks*," in Proc. Electrotechnical Conference, (Lisbon, Portugal, 1989), pp. 311-314.

[24] M. Shams and M. Elmasry, "Delay Optimization of CMOS Logic Circuits Using Closed-Form Expressions," in Int. Conf. Computer Design, (Austin, TX, 1999), pp. 563-568.

[25] J. Rossello, C. de Benito and J. Segura, "*A Compact Gate-Level Energy and Delay Model of Dynamic CMOS Gates*," *IEEE Trans. Circuits and Systems*, **52**, pp. 685-689, (2005).

[26] C. Wu, J. Hwang, C. Chang and C. Chang, "*An Efficient Timing Model for CMOS Combinational Logic Gates*," IE*EE Trans. Computer-Aided Design*, **4**, pp. 636-650, (1985).

[27] B. Lasbouygues, J. Schindler, S. Engels, P. Maurine, N. Azemard and D. Auvergne, "Continuos Representation of the Performance of a CMOS Library." in Proc. 29th European Solid-State Circuits Conf., (England, 2003), pp. 595-598.

[28] S. Vemuru and A. Thorbjornsen, "Delay-Modeling of NAND Gates," in Proc. 33rd Midwest Symposium Circuits and Systems, (Calgary, Canada, 1990), pp. 922-925.

[29] A. Hirata, H. Onodera and K. Tamaru, "*Estimation of Propagation Delay Considering Short-Circuit Current for Static CMOS Gates*," *IEEE Trans. Circuits and Systems*, **45**, pp. 1194-1198, (1998).

[30] S. Nikolaidis and A. Chatzigeorgiou, "*Modeling the Transistor Chain Operation in CMOS Gates for Short Channel Devices*," *IEEE Trans. Circuits and Systems*, **46**, pp. 1191-1202, (1999).

[31] P. Maurine, M. Rezzoug, and D. Auvergne, "Output Transition Time Modeling of CMOS Structures," in 2001 IEEE Int. Symposium Circuits and Systems, (Sydney, New South Whales, 2001), pp. 363-366.

[32] J. Rosello and J. Segura, "Power-Delay Modeling of Dynamic CMOS Gates for Circuit Optimization," in IEEE/ACM Int. Conf. Computer Aided Design, (San Jose, CA, 2001), pp. 494-499.

[33] R. Baker, CMOS Circuit Design, Layout and Simulation, 2nd Edition (Piscataway NJ: IEEE Press, 2005), pp 45-47.

[34] K. Jeppson, "*Modeling the Influence of the Transistor Gain Ratio and the Input-to-Output Coupling Capacitance on the CMOS Inverter Delay*," IEEE Journal of Solid State Circuits, **29**, pp. 645-654, (1994).

APPENDIX A. TSMC 0.18 μm PROCESS FILE.

```
//  File: tsmc18d.scs
//  Abstract: TSMC 0.18u CMOS018/DEEP (6M, HV FET, sblock) Spectre Models
// simulator options simulator lang=spectre insensitive=yes
//  4-Terminal NMOS Model
//  DATE: Dec  9/02
//  LOT: T29B                WAF: 6003
//  Temperature_parameters=Default model tsmc18dn bsim3v3 type=n
+ version=3.1              tnom=27                  tox=4e-9
+ xj=1e-7                  nch=2.3549e17            vth0=0.3627858
+ k1=0.5873035             k2=4.793052e-3           k3=1e-3
+ k3b=2.2736112            w0=1e-7                  nlx=1.675684e-7
+ dvt0w=0                  dvt1w=0                  dvt2w=0
+ dvt0=1.7838401           dvt1=0.5354277           dvt2=-1.243646e-3
+ u0=263.3294995           ua=-1.359749e-9          ub=2.250116e-18
+ uc=5.204485e-11          vsat=1.083427e5          a0=2
+ ags=0.4289385            b0=-6.378671e-9          b1=-1e-7
+ keta=-0.0127717          a1=5.347644e-4           a2=0.8370202
+ rdsw=150                 prwg=0.5                 prwb=-0.2
+ wr=1                     wint=1.798714e-9         lint=7.631769e-9
+ xl=-2e-8                 xw=-1e-8                 dwg=-3.268901e-9
+ dwb=7.685893e-9          voff=-0.0882278          nfactor=2.5
+ cit=0                    cdsc=2.4e-4              cdscd=0
+ cdscb=0                  eta0=2.455162e-3         etab=1
+ dsub=0.0173531           pclm=0.7303352           pdiblc1=0.2246297
+ pdiblc2=2.220529e-3      pdiblcb=-0.1             drout=0.7685422
+ pscbe1=8.697563e9        pscbe2=5e-10             pvag=0
+ delta=0.01               rsh=6.7                  mobmod=1
+ prt=0                    ute=-1.5                 kt1=-0.11
+ kt1l=0                   kt2=0.022                ua1=4.31e-9
+ ub1=-7.61e-18            uc1=-5.6e-11             at=3.3e4
+ wl=0                     wln=1                    ww=0
+ wwn=1                    wwl=0                    ll=0
+ lln=1                    lw=0                     lwn=1
+ lwl=0                    capmod=2                 xpart=0.5
+ cgdo=7.16e-10            cgso=7.16e-10            cgbo=1e-12
+ cj=9.725711e-4           pb=0.7300537             mj=0.365507
+ cjsw=2.604808e-10        pbsw=0.4                 mjsw=0.1
+ cjswg=3.3e-10            pbswg=0.4                mjswg=0.1
+ cf=0                     pvth0=4.289276e-4        prdsw=-4.2003751
+ pk2=-4.920718e-4         wketa=6.938214e-4        lketa=-0.0118628
+ pu0=24.2772783           pua=9.138642e-11         pub=0
+ pvsat=1.680804e3         peta0=2.44792e-6         pketa=4.537962e-5
```

```
 //  DATE: Dec  9/02
//  LOT: T29B               WAF: 6003
//  Temperature_parameters=Default
//
model tsmc18dp bsim3v3 type=p
+ version=3.1           tnom=27                 tox=4e-9
+ xj=1e-7               nch=4.1589e17           vth0=-0.4064886
+ k1=0.5499001          k2=0.0389453            k3=0
+ k3b=11.4951756        w0=1e-6                 nlx=9.143209e-8
+ dvt0w=0               dvt1w=0                 dvt2w=0
+ dvt0=0.5449299        dvt1=0.3160821          dvt2=0.1
+ u0=117.9612996        ua=1.64867e-9           ub=1.165056e-21
+ uc=-1e-10             vsat=2e5                a0=1.7833459
+ ags=0.407511          b0=1.314603e-6          b1=5e-6
+ keta=0.0137171        a1=0.4610527            a2=0.6597363
+ rdsw=364.9443889      prwg=0.5                prwb=-0.1129203
+ wr=1                  wint=0                  lint=2.007556e-8
+ xl=-2e-8              xw=-1e-8                dwg=-2.835566e-8
+ dwb=8.003075e-9       voff=-0.1064646         nfactor=2
+ cit=0                 cdsc=2.4e-4             cdscd=0
+ cdscb=0               eta0=0.0141703          etab=-0.0398356
+ dsub=0.4441401        pclm=2.2364512          pdiblc1=9.167645e-4
+ pdiblc2=0.0209189     pdiblcb=-9.568266e-4    drout=9.976778e-4
+ pscbe1=1.731161e9     pscbe2=5e-10            pvag=14.337819
+ delta=0.01            rsh=7.5                 mobmod=1
+ prt=0                 ute=-1.5                kt1=-0.11
+ kt1l=0                kt2=0.022               ua1=4.31e-9
+ ub1=-7.61e-18         uc1=-5.6e-11            at=3.3e4
+ wl=0                  wln=1                   ww=0
+ wwn=1                 wwl=0                   ll=0
+ lln=1                 lw=0                    lwn=1
+ lwl=0                 capmod=2                xpart=0.5
+ cgdo=6.79e-10         cgso=6.79e-10           cgbo=1e-12
+ cj=1.176396e-3        pb=0.8607121            mj=0.4163285
+ cjsw=2.135953e-10     pbsw=0.6430918          mjsw=0.2654457
+ cjswg=4.22e-10        pbswg=0.6430918         mjswg=0.2654457
+ cf=0                  pvth0=4.364418e-3       prdsw=4.4192048
+ pk2=3.104478e-3       wketa=0.0270296         lketa=2.038008e-3
+ pu0=-2.3639825        pua=-8.41675e-11        pub=1e-21
+ pvsat=-50             peta0=1e-4              pketa=-1.444802e-3
```

APPENDIX B. IBM 0.13 μm PROCESS FILE.

```
T73J SPICE BSIM3 VERSION 3.1 PARAMETERS
SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level 8
* DATE: Aug 10/07
* LOT: T73J                WAF: 2001
* Temperature_parameters=Default MODEL CMOSN NMOS (LEVEL   = 49
+VERSION = 3.1          TNOM= 27                TOX= 3.2E-9
+XJ= 1E-7               NCH= 2.3549E17          VTH0= 0.0564776
+K1= 0.2897355          K2= -0.015383           K3= 1E-3
+K3B= 4.0710922         W0= 1E-7                NLX= 1E-6
+DVT0W= 0               DVT1W= 0                DVT2W= 0
+DVT0= 1.0145151        DVT1= 0.1685897         DVT2= 0.2406542
+U0= 445.1306953        UA= -4.57424E-10        UB= 3.44869E-18
+UC= 3.952766E-10       VSAT= 1.998507E5        A0= 0.8864242
+AGS= 0.8658495         B0= 6.191191E-6         B1= 5E-6
+KETA= 0.0262826        A1= 1.39548E-3          A2= 0.3
+RDSW= 150              PRWG= 0.3535806         PRWB= 0.1081166
+WR= 1                  WINT= 1.225721E-8       LINT= 1.036724E-8
+DWG= 4.018893E-9       DWB= 1.292839E-8        VOFF= -0.0406926
+NFACTOR = 2.5          CIT= 0                  CDSC= 2.4E-4
+CDSCD= 0               CDSCB= 0                ETA0= 2.769384E-6
+ETAB= 0.4385468        DSUB= 4.088069E-6       PCLM= 0.963888
+PDIBLC1= 0.9949239     PDIBLC2 = 0.01          PDIBLCB = 0.1
+DROUT= 0.9981743       PSCBE1= 7.959045E10     PSCBE2= 5E-10
+PVAG= 0.500353         DELTA= 0.01             RSH= 6.9
+MOBMOD= 1              PRT= 0                  UTE= -1.5
+KT1= -0.11             KT1L= 0                 KT2= 0.022
+UA1= 4.31E-9           UB1= -7.61E-18          UC1= -5.6E-11
+AT= 3.3E4              WL= 0                   WLN= 1
+WW= 0                  WWN= 1                  WWL= 0
+LL= 0                  LLN= 1                  LW= 0
+LWN= 1                 LWL= 0                  CAPMOD= 2
+XPART= 0.5             CGDO= 4E-10             CGSO= 4E-10
+CGBO= 1E-12            CJ= 8.385747E-4         PB= 0.8813098
+MJ= 0.5484215          CJSW= 2.460231E-10      PBSW= 0.8
+MJSW= 0.3063897        CJSWG= 3.3E-10          PBSWG= 0.8
+MJSWG= 0.3063897       CF= 0                   PVTH0= 2.009264E-4
+PRDSW= 0               PK2= 1.30501E-3         WKETA= -2.516447E-3
+LKETA= 5.135467E-3     PU0= 4.4729531          PUA= 1.66833E-11
+PUB= 0                 PVSAT= 653.2294237      PETA0= 1E-4
+PKETA= -0.0282915)
```

```
.MODEL CMOSP PMOS (LEVEL   = 49
+VERSION = 3.1        TNOM= 27               TOX= 3.2E-9
+XJ= 1E-7             NCH= 4.1589E17         VTH0= -0.2285194
+K1= 0.236504         K2= 0.0273863          K3= 0.0989953
+K3B= 6.4994037       W0= 1E-6               NLX= 2.709344E-7
+DVT0W= 0             DVT1W= 0               DVT2W= 0
+DVT0= 7.848909E-3    DVT1= 0.0871763        DVT2= 0.1
+U0= 110.9145614      UA= 1.460494E-9        UB= 1E-21
+UC= -2.14484E-11     VSAT= 2E5              A0= 0.6677214
+AGS= 0.1149671       B0= 8.195389E-6        B1= 3.845906E-6
+KETA= 0.0335186      A1= 1.14322E-3         A2= 0.4010086
+RDSW= 105.0859242    PRWG= -0.4995324       PRWB= 0.5
+WR= 1                WINT= 0                LINT= 8.79977E-9
+DWG= 1.248761E-9     DWB= -2.285216E-8      VOFF= -0.1022829
+NFACTOR = 1.5332272  CIT = 0                CDSC= 2.4E-4
+CDSCD= 0             CDSCB= 0               ETA0= 1.602419E-3
+ETAB= -7.975494E-3   DSUB= 1.660379E-3      PCLM= 0.1189766
+PDIBLC1 = 0.0169335  PDIBLC2= -1.81127E-11  PDIBLCB = -1E-3
+DROUT= 0             PSCBE1= 6.701825E9     PSCBE2  = 2.047831E-9
+PVAG= 3.671013E-4    DELTA= 0.01            RSH= 6.6
+MOBMOD= 1            PRT= 0                 UTE= -1.5
+KT1= -0.11           KT1L= 0                KT2= 0.022
+UA1= 4.31E-9         UB1= -7.61E-18         UC1= -5.6E-11
+AT= 3.3E4            WL= 0                  WLN= 1
+WW= 0                WWN= 1                 WWL= 0
+LL= 0                LLN= 1                 LW= 0
+LWN = 1              LWL = 0                CAPMOD= 2
+XPART= 0.5           CGDO= 3E-10            CGSO= 3E-10
+CGBO= 1E-12          CJ= 1.174293E-3        PB= 0.8219834
+MJ= 0.4095402        CJSW= 1.316489E-10     PBSW= 0.8813044
+MJSW= 0.1            CJSWG= 4.22E-10        PBSWG= 0.8813044
+MJSWG= 0.1           CF= 0                  PVTH0= 5.431055E-4
+PRDSW= 52.1485073    PK2= 1.86276E-3        WKETA= 0.0353662
+LKETA= 9.219417E-3   PU0= -1.2656982        PUA= -5.86504E-11
+PUB= 8.61298E-24     PVSAT= 50              PETA0= 1E-4
+PKETA= -2.855693E-3)
```

APPENDIX C. INVERTER SIZING TABLES.

The Inverter Sizing Table (Original Verified Method)

Simulation	W_N(cm) Current W_P(cm) Current	t_{PHL} measured(ps) t_{PHL} measured(ps)	%Error from target 32.3ps	W_N(cm) Next W_P(cm) Next
1	4.84E-04 9.68E-05	37.8 44.2	17.03 36.84	5.66E-05 1.32E-04
2	5.66E-05 1.32E-04	37.0 36.5	14.55 13.00	6.49E-05 1.50e-04
3	6.49E-05 1.50e-04	35.0 34.4	8.36 6.50	7.03E-5 1.59E-4
4	7.03E-5 1.59E-4	33.9 33.4	4.95 3.41	7.38E-5 1.65E-4
5	7.38E-5 1.65E-4	33.1 33.0	2.48 2.17	7.56E-5 1.68E-4
6	7.56E-5 1.68E-4	32.8 32.7	1.55 1.24	7.68E-5 1.71e-4
7	7.68E-5 1.71e-4	32.6 32.6	0.93 0.93	

The Improvement (New Shortened Method)

Simulation	W_N(cm) Current W_P(cm) Current	t_{PHL} measured(ps) t_{PHL} measured(ps)	%Error from target 38.6ps	W_N(cm) Next W_P(cm) Next
1	3.68E-04 7.36E-05	42.47 45.10	10.05 16.84	4.05E-05 8.60E-05
2	4.05E-05 8.60E-05	41.33 41.73	7.08 8.11	4.33E-05 9.29E-05
3	4.33E-05 9.29E-05	40.28 40.07	4.37 3.80	4.52E-05 9.64E-05
4	4.52E-05 9.64E-05	39.34 39.07	1.92 1.21	4.61E-05 9.75E-05
5	4.61E-05 9.75E-05	39.02 38.97	1.11 0.97	4.66E-05 9.84E-05
6	4.50E-05 9.90E-05	N/A	N/A	N/A
7	One less step			

APPENDIX D. RESULTS VERIFICATION.

Device Type	FO Used	Input Slope (ps)	Min Delay (ps)	Max Delay (ps)	Min Dev. width %Error	Max Dev. width %Error	Improved Min Device width %Error	Improved Max Device width %Error
Inverter	FO-1	34	18	39.9	-9.3	27	-7.4	24.9
	FO-1	222	NA	NA	21.5	60.2	22.9	44.2
	FO-1	410	55	69.2	9	34.1	10.8	24.6
Inverter	FO-4	34	20.3	55	-26.2	9.9	-25.4	-11.4
	FO-4	222	55	130	-22.6	43.9	-20.8	25.3
	FO-4	410	55	150	-25.4	24.3	-23.8	23.3
NAND2	FO-1	34	26.3	64.4	-12.5	16.3	-11.6	-8.4
	FO-1	222	65	83	5.7	48.7	6.2	20.3
	FO-1	410	70	103	2.6	27.8	3.7	1.8
NAND2	FO-4	34	31.8	100	-28.9	9.1	-28.8	-2.1
	FO-4	222	65	140	-32.6	33.7	-32.5	24.9
	FO-4	410	67	150	-33.5	15.4	-32.4	9.1
NAND3	FO-1	34	40.4	95.5	-15.7	8.7	-14.5	-0.8
	FO-1	222	70	115	-8.7	27.8	-6.9	22.3
	FO-1	410	77	135	-10	14.5	-8.9	-4.8
NAND3	FO-4	34	45.9	140	-27.5	5.4	-26.7	-19.3
	FO-4	222	66.8	150	-36.8	18.4	-36.8	13.5
	FO-4	410	89.6	160	-36.4	10.4	-34.7	-12.1
NAND4	FO-1	34	57.1	132	-19.5	2.1	-17.8	-3.1
	FO-1	222	79.5	151	-20.7	9.9	-20.3	-13.9
	FO-1	410	101	169	-17.4	12.9	-15.6	-5.3
NAND4	FO-4	34	62.5	200	-29.8	-0.7	-28.6	-23.3
	FO-4	222	88.8	150	-41.5	7.7	-40.1	4.2
	FO-4	410	113	170	-40.3	10.4	-38.9	-16
NOR2	FO-1	34	45	60.6	7.3	28.8	8.5	25.1
	FO-1	222	77	77	20.3	58.9	20.8	34.1
	FO-1	410	94.5	94.5	19.5	19.5	20.6	10.5
NOR2	FO-4	34	43	134	-12.7	19.1	-11.5	18.2
	FO-4	222	80	171	-13.9	41.3	-12.8	33.1
	FO-4	410	100	208	-16.4	41.2	-15.5	40.1
NOR3	FO-1	34	80	92	13.4	25.1	13.5	18.2
	FO-1	222	95	109	14.2	35.5	14.3	13
	FO-1	410	123	123	20.2	41.4	20.7	15
NOR3	FO-4	34	77	165	-11.5	18.9	-9.6	-3.9
	FO-4	222	100	203	-11.8	24.7	-11	2.3
	FO-4	410	120	237	-15.5	30.4	-14.3	29.4
NOR4	FO-1	34	115	134	16.1	22.1	17.3	17.6
	FO-1	222	115	150	7.5	14.6	8.1	-9.3
	FO-1	410	NA	NA	21.9	29.6	22.2	16.2
NOR4	FO-4	34	123	207	-12.9	18.3	-12.4	-6.1
	FO-4	222	132	244	-11	9.3	-9.2	4.9
	FO-4	410	150	273	-18.2	24.4	-18.1	-2
Mean			77.24	138.35	-10.48	20.35	-9.5	7.6
Std.	Dev.		31.88	53.5	18.7	14.4	18.7	17