8-2012

Ultra-low Voltage Digital Circuits and Extreme Temperature Electronics Design

Aaron J. Arthurs
University of Arkansas, Fayetteville

ULTRA-LOW VOLTAGE DIGITAL CIRCUITS AND EXTREME TEMPERATURE ELECTRONICS DESIGN

ULTRA-LOW VOLTAGE DIGITAL CIRCUITS AND EXTREME TEMPERATURE ELECTRONICS DESIGN

A dissertation submitted in partial fullfillment
of the requirements for the degree of
Doctor of Philosophy in Computer Engineering

By

Aaron J Arthurs
University of Arkansas
Bachelor of Science in Computer Engineering, 2004
University of Arkansas
Master of Science in Computer Engineering, 2007

August 2012
University of Arkansas

ABSTRACT

Certain applications require digital electronics to operate under extreme conditions e.g., large swings in ambient temperature, very low supply voltage, high radiation. Such applications include sensor networks, wearable electronics, unmanned aerial vehicles, spacecraft, and energy-harvesting systems. This dissertation splits into two projects that study digital electronics supplied by ultra-low voltages and build an electronic system for extreme temperatures. The first project introduces techniques that improve circuit reliability at deep subthreshold voltages as well as determine the minimum required supply voltage. These techniques address digital electronic design at several levels: the physical process, gate design, and system architecture. This dissertation analyzes a silicon-on-insulator process, Schmitt-trigger gate design, and asynchronous logic at supply voltages lower than 100 millivolts. The second project describes construction of a sensor digital controller for the lunar environment. Parts of the digital controller are an asynchronous 8031 microprocessor that is compatible with synchronous logic, memory with error detection and correction, and a robust network interface. The digitial sensor ASIC is fabricated on a silicon-germanium process and built with cells optimized for extreme temperatures.

This dissertation is approved for recommendation
to the Graduate Council.

Dissertation Director:

_____

Dr. Jia Di

Dissertation Committee:

_____

Dr. H. Alan Mantooth

_____

Dr. Scott C. Smith

_____

Dr. James P. Parkerson

DISSERTATION DUPLICATION RELEASE

I hereby authorize the University of Arkansas Libraries to duplicate this dissertation when needed for research and/or scholarship.

Agreed _____

Aaron J Arthurs

Refused _____

Aaron J Arthurs

ACKNOWLEDGMENTS

Special thanks are due to the staff of MIT Lincoln Laboratory for all of their help with their FD-SOI process. Their help and FD-SOI process added great value to this research.

Also, special thanks goes out to the faculty and staff at the University of Arkansas for their commitment to the University and their students.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

There are applications of digital electronics that demand reduced energy consumption, minimal noise production, and high tolerance for environmental variations. Applications that manifest these requirements include sensor networks, wearable electronics, unmanned aerial vehicles, and more generally mixed-signal and energy-harvesting systems. Mixed-signal and energy-harvesting electronics utilize digital circuits as a source of intelligence or control, data processing, and high-level communication. For mixed-signal systems, special attention is given to isolating each signal domain. Energy-harvesting systems require prolonged operation period and limit energy availability. Techniques suitable for energy-harvesting systems include subthreshold supply operation and clockless system design. Designing digital systems for subthreshold operation must account for variations in process, supply voltage, and temperature (i.e., PVT variation). At low voltages, digital signal integrity is another problem to be addressed. This research work focuses on several facets of digital design among two projects: ultra-low voltage and extreme temperature operation. The first project analyzes several techniques to improve digital signal integrity at ultra-low voltages, targeting lower than 100 millivolts. The second project is a digital sensor network module built on IBM's SiGe 5AM process that is tailored for the lunar environment.

Scaling down supply voltages and device sizes increases the effects of intra-die process variation. Reducing the supply voltage decreases the margin between supply and the device's threshold voltage. With increased process variation effects, that margin decreases further due to variance in threshold voltage. Also as device size approaches fundamental dimensions (e.g., atom dimensions and light wavelengths), process variation contributes to a larger percentage of each

device's dimensions [14], further increasing threshold fluctuations.

Intra-die process variation adversely affects functionality especially for the synchronous paradigm adopted by most digital systems. In a synchronous circuit, each active clock edge must trigger all synchronous memory elements simultaneously in order to maintain correct data flow, but clock skew is inevitable due to wire delay. Minimizing clock skew due to wire delay is often solved by clock tree synthesis (CTS), which creates clock buffer trees that add propagation delay to non-critical clock paths. However, CTS assumes that identical buffers have identical propagation delay, so the buffers themselves can create significant clock skew if process variation is great enough. There also exists variation aware CTS that simultaneously factors in buffer modification and wire sizing [32]. Commercial place-and-route tools support on-chip variation (OCV) aware CTS that references slow and fast gate timing to calculate worst-case skew per clock sink; this method is known as OCV de-rating. An alternative method to OCV aware CTS is clock mesh synthesis where multiple, tunable buffers drive a shared clock net prearranged into a regular grid [24]; however, clock meshes add more wire area and, consequently, higher load capacitance and power dissipation [33]. Several more approaches of reducing clock skew effects are reducing the number of clock sinks, partitioning the system into multiple clock domains, and clockless system design.

Another cause of uncertainty in timing is temperature fluctuation. Temperature fluctuations are caused by the ambient temperature and resistive components in a circuit. While ambient temperature affects the circuit as a whole, heat from resistive components create localized timing variations. The characteristics of heat generated by a digital circuit depends on the switching ac-

tivity of each component. Given the supply voltage, fluctuations in temperature can have a direct, absent, or reversed effect on MOSFET delay [15, 16].

Integrating digital electronics alongside analog components brings concern of digital noise. Digital signals switch between two power rails at high frequencies, resulting in significant substrate noise. Some submicron processes support triple-well isolation for analog circuits. Triple-well isolation provides excellent substrate noise isolation for multi-gigahertz designs and allows simultaneous use of multiple supply voltages (body bias control) [8]. There are also techniques applicable to digital circuits that mitigate the effects of digital signal noise. Such techniques include scaling down the supply voltage, designing clockless circuits, and isolating each digital device.

A MOSFET technology that isolates digital devices is silicon-on-insulator (SOI). In addition to device isolation, SOI technology provides a number of other advantages over bulk-Si technology that make SOI more suitable for large environmental fluctuations. SOI technology began its development circa 1978 and evolved into two primary SOI flavors: fully-depleted (FD-SOI) and partially-depleted (PD-SOI). PD-SOI technology is tailored for high-voltage applications and I/O circuits. FD-SOI is preferred for digital circuits because it offers better subthreshold performance [25].

Designing clockless circuits requires an asynchronous paradigm, either bounded-delay or delay-insensitive [27]. The bounded-delay asynchronous model is similar to the synchronous model in that data processing paths must operate faster than the flow control paths. The difference is that control paths, instead of being driven by external sources such as clocks, are driven by the same data flowing through the data processing paths. In contrast, the delay-insensitive (DI)

asynchronous model utilizes output completion detection to allow for unbounded wire and gate delays. The DI model is most practical for PVT-tolerant, energy-efficient circuits because functionality is independent to timing. Several DI schemes have been proposed. In this research work, NULL Convention Logic (NCL), a symbolically complete quasi-delay-insensitive paradigm, is studied in designing digital electronics for ultra-low voltage and extreme environment.

Signal drive strength degrades as the supply voltage drops into the subthreshold region. Digital circuits operate on leakage current in the subthreshold region, and the difference between active and inactive source-to-drain current diminishes as the supply voltage decreases. The reliability of a digital circuits depends on how well each cell drives the inputs of subsequent cells. Failure occurs when a digital cell evaluates any of its inputs incorrectly.

Subthreshold supply operation, FD-SOI technology, and NCL logic have benefits in the areas of energy efficiency, PVT tolerance, and mixed-signal systems. It is of interest to investigate subthreshold operation, FD-SOI, and NCL as tools in building digital circuits for future systems. The first project applies the said techniques to two circuits, a pipelined ripple-carry adder and an IEEE floating-point coprocessor. Two processes are chosen for bulk-silicon and FD-SOI respectively. Each circuit is built with transistor-based schematics, and parasitics derived from each process are excluded from analysis.

The second project involves constructing a digital ASIC suited for extreme temperature operation at an above-threshold supply voltage. A complete workflow is presented along with methods used to improve reliability of operation. Several revisions of the digital ASIC were released, and each release addresses improvements made to the workflow.

2 ULTRA-LOW VOLTAGE CIRCUIT DESIGN

2.1 BACKGROUND

The entire region of operation at ultra-low voltages is below threshold. Only leakage current drives transistors, so drive strength significantly degrades. The quality of drive strength differentiates circuits designed for subthreshold operation from circuits that rely on the saturation region of each transistor. One major difference is the large increase in propagation delay since subthreshold operation limits the source-to-drain current and is only suitable for low-speed applications. Another important difference is that the structure of each gate is given more consideration in regard to inactive leakage current. Lastly, circuits operating in subthreshold regions are more susceptible to process/voltage/temperature variations, and their timing are therefore more difficult to predict than systems operating at above threshold.

Pushing subthreshold circuit design to extreme, ultra-low voltage circuit design requires unconventional methods of construction. Described in later sections, this research defines ultra-low voltage as being less than 100 millivolts since many conventional designs are capable of operating as low as 100 millivolts. Ultra-low voltages are considered to be deep in the subthreshold region because most processes share an approximate range of 300 to 700 millivolts for the threshold voltage.

Until now, ultra-low voltage digital design is an active area of research. There are no universal solutions on design methodology. Designing a circuit for subthreshold operation depends on the application, much like above-threshold design. Previous research in process variation studied different logic implementations and FD-SOI; Figure 1 shows a layout of an array of digital circuits

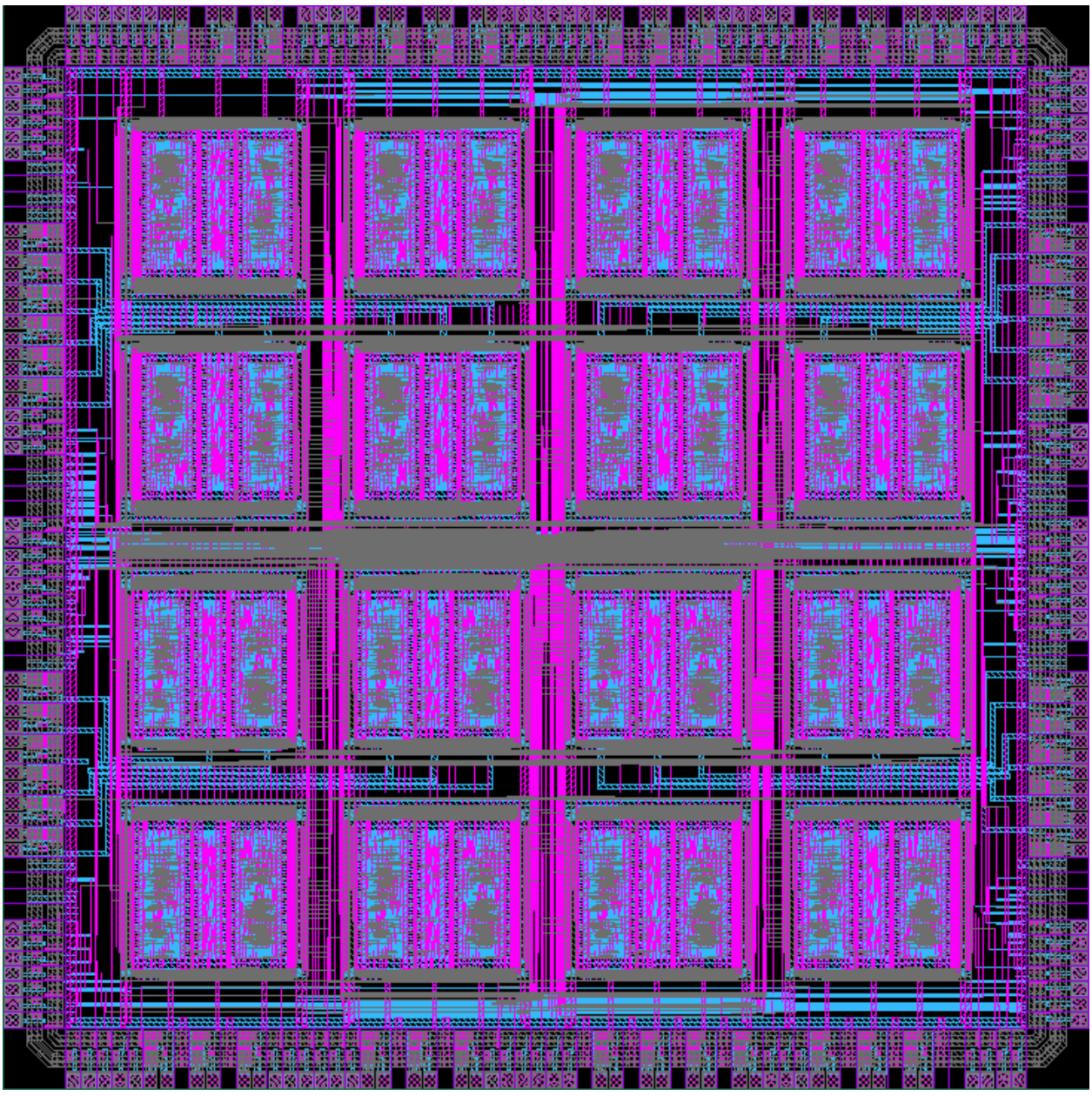

Figure 1: ALU array layout on the MIT Lincoln Laboratory 150 nm FD-SOI process.

to capture the effects of intra-die process variation. Several candidate techniques are analyzed as promising tools for both subthreshold and ultra-low voltage design.

2.1.1 SUBTHRESHOLD OPERATION

The subthreshold region represents weak inversion such that load capacitances are charged by leakage current. Subthreshold current is exponentially dependent on the threshold voltage [11], which in turn is strongly dependent on various device model parameters. On deep submicron processes, device model parameters vary considerably, and subthreshold designs are therefore expected to be prone to process variations. Circuits operating at above-threshold are equally affected by geometric (e.g., L_{eff}) variations and random dopant fluctuations (RDF), but under subthreshold voltages, RDF is the dominant component of variation [35]. RDF is independent of the supply voltage and is proportional to the inverse square root of channel area [23].

Inside a logic gate, sizing of MOSFETs affects the variability in the output logic swing [17]. For example, if process variation strengthens PMOS relative to NMOS, a pull-down network may not be able to drive the logic output close enough to ground. Subsequent evaluation may fail to process the output as being logically low. The ratio of active to inactive subthreshold current is critical to functionality.

A common expression used to evaluate subthreshold current is given by [3] and shown in Equation 1 and Equation 2

$$I_{sub} = I_0 e^{\frac{V_{gs}-V_T+\eta V_{ds}}{nV_{th}}}(1 - e^{-\frac{V_{ds}}{V_{th}}}) \tag{1}$$

$$I_0 = \mu_0 C_{ox} \frac{W}{L}(n-1)V_{th}^2 \tag{2}$$

where n is the subthreshold swing factor, V_{th} is the thermal voltage, and η is the drain-induced barrier lowering (DIBL) coefficient. V_T is a distribution of the threshold voltage whose standard deviation scales with $\frac{1}{\sqrt{WL}}$, consistent with the assumption that RDF dominates process variation at subthreshold. Subthreshold current scales linearly with the ratio of $\frac{W}{L}$ and exponentially with the threshold voltage.

Inactive subthreshold current is given by $I_{sub}(V_{gs} = 0, V_{ds} = V_{dd}) = I_0 e^{\frac{-V_T+\eta V_{dd}}{nV_{th}}}(1 - e^{-\frac{V_{dd}}{V_{th}}})$ while active subthreshold current is given by $I_{sub}(V_{gs} = V_{ds} = V_{dd}) = I_0 e^{\frac{-V_T+V_{dd}(1+\eta)}{nV_{th}}}(1 - e^{-\frac{V_{dd}}{V_{th}}})$. The ratio of active to inactive subthreshold current is therefore $\frac{I_{on}}{I_{off\ ideal}} = e^{\frac{V_{dd}}{nV_{th}}}$, assuming that the transistor gate input fully swings between the power and ground rails. For a non-ideal transistor gate swing between a low input voltage (V_{IL}) and a high input voltage (V_{IH}), the active-to-inactive subthreshold current ratio is given in Equation 3.

$$\frac{I_{on}}{I_{off}} = e^{\frac{V_{IH}-V_{IL}}{nV_{th}}} \tag{3}$$

Process variation affects subthreshold current exponentially and is an issue with the functionality of low-voltage circuits. One solution is to increase the channel area to reduce the effectiveness of RDF. As evident in Equation 1, the standard deviation of the distribution, V_T, decreases, and the range in subthreshold current narrows exponentially for larger transistors. Considering Equation 3, the gate input voltage swing strongly affects the difference between active and inactive subthreshold current, regardless of transistor size. The required magnitude of the active-to-inactive current ratio depends on the fan-out load driven.

Inactive leakage paths is another issue with subthreshold operation that manifests in cell

designs, particularly in static CMOS designs. Standard static CMOS design employs a minimum number of transistors required to implement the pull-up and pull-down networks. In order to simplify timing analysis, the resistance of the pull-up and pull-down networks are balanced, equalizing the cell output's rise and fall transition times. Balancing the resistance of the two networks primarily involves adjusting each transistor's channel width. However, subthreshold current depends more on the gate input voltage than the transistor channel width. Because the gate input voltage never reaches power or ground, an inactive transistor conducts subthreshold current that is exponentially proportional to the difference between the gate input voltage and the power rails (i.e., $(V_{dd} - V_{IH})$ for PMOS and $(V_{IL} - V_{ground})$ for NMOS). Inactive leakage current from the cell output accumulates as the number of parallel transistor paths increases. In contrast, a series of inactive transistors allows less leakage current than a single inactive transistor. Therefore, the topology of the pull-up and pull-down networks strongly influences the cell's output voltage swing, more so than the network resistance.

A solution to managing inactive leakage paths is transistor stacking. Transistor stacking can either involve exploiting existing transistor stacks (e.g., the NMOS transistor stack in a NAND gate) via optimal input vectors or inserting extra transistors in series (i.e., forced transistor stacking). Transistor stacking causes a slight reverse bias between the gate and source of each transistor in series [4]. With transistor stacking, inactive leakage current decreases substantially because of its exponential dependence on gate bias. At the expense of area and speed, forced transistor stacking is a method in designing the topology of a cell that reduces inactive leakage current and potentially improves the active-to-inactive current ratio.

Another problem is predicting timing throughout the system in presence of process variation. Process variation strongly affects subthreshold currents in low-voltage circuits, so propagation delays and output transition times throughout a circuit deviate from nominal timing. In effect, the critical paths in a system may change throughout fabrication. Process variation also introduces clock skew, which is detrimental to a synchronous circuit's functionality. Clock tree buffers operating at subthreshold are susceptible to variable timing.

2.1.2 DEEP SUBTHRESHOLD OPERATION AND STATIC NOISE MARGIN

This research work distinguishes deep subthreshold operation as the region where circuit functionality begins to breakdown. The minimum supply voltage required by a digital circuit largely depends on the cell library. Two key characteristics of a cell library that determine the minimum supply voltage are each cell's input voltage swing requirement and output driving capability. The weakest pull-up and pull-down DC paths from the cell output define its output drive capability. In addition, the input voltage swing strongly affects the pull-up and pull-down drivers. Determining the minimum supply voltage for a cell library involves analyzing how each cell drives subsequent cells.

Because finding the minimum supply voltage cannot be isolated to each cell, two approaches are running voltage sweeps on a given digital circuit and modeling an infinite chain of alternating cells [18]. The advantage of running voltage sweeps is finding the smallest possible supply voltage needed by the given digital circuit; however, voltage sweeps involve long simulation times that depend on the size of the digital circuit as well as the range in supply voltages. In contrast, modeling an infinite chain of alternating cells requires short simulation times and returns

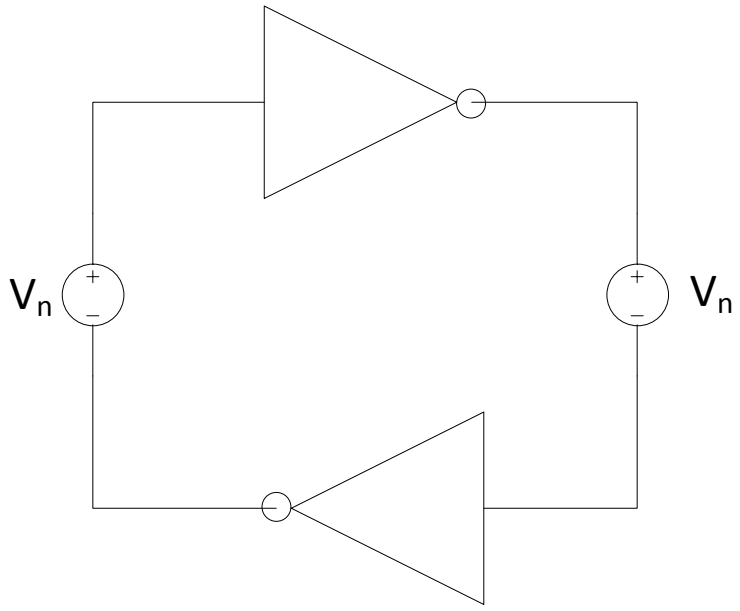

Figure 2: Simplified model of an infinite chain of alternating negative unate cells.

a definitive answer for the minimum supply voltage supported by the cell library. The disadvantage is that the minimum supply voltage supported by the cell library is based on worst-case conditions and may therefore be higher than the supply voltage required by a given circuit. The two approaches can be combined by first determining the minimum supply voltage supported by the cell library followed by running voltage sweeps over small ranges in supply.

The purpose of modeling an infinite chain of alternating cells is to calculate the smallest static noise margin (SNM) of a cell's output voltage relative to another cell's input voltage. For a given supply voltage, inter-cell SNM (or gate-pair SNM) represents how much fluctuation in a cell's output voltage can be tolerated by the next cell. Inter-cell SNM is directly proportional to the supply voltage, and the supply voltage at which inter-cell SNM is zero volt is the minimum supply voltage required by the infinite cell chain [26].

In order to calculate inter-cell SNM, the infinite cell chain is simplified into two inter-

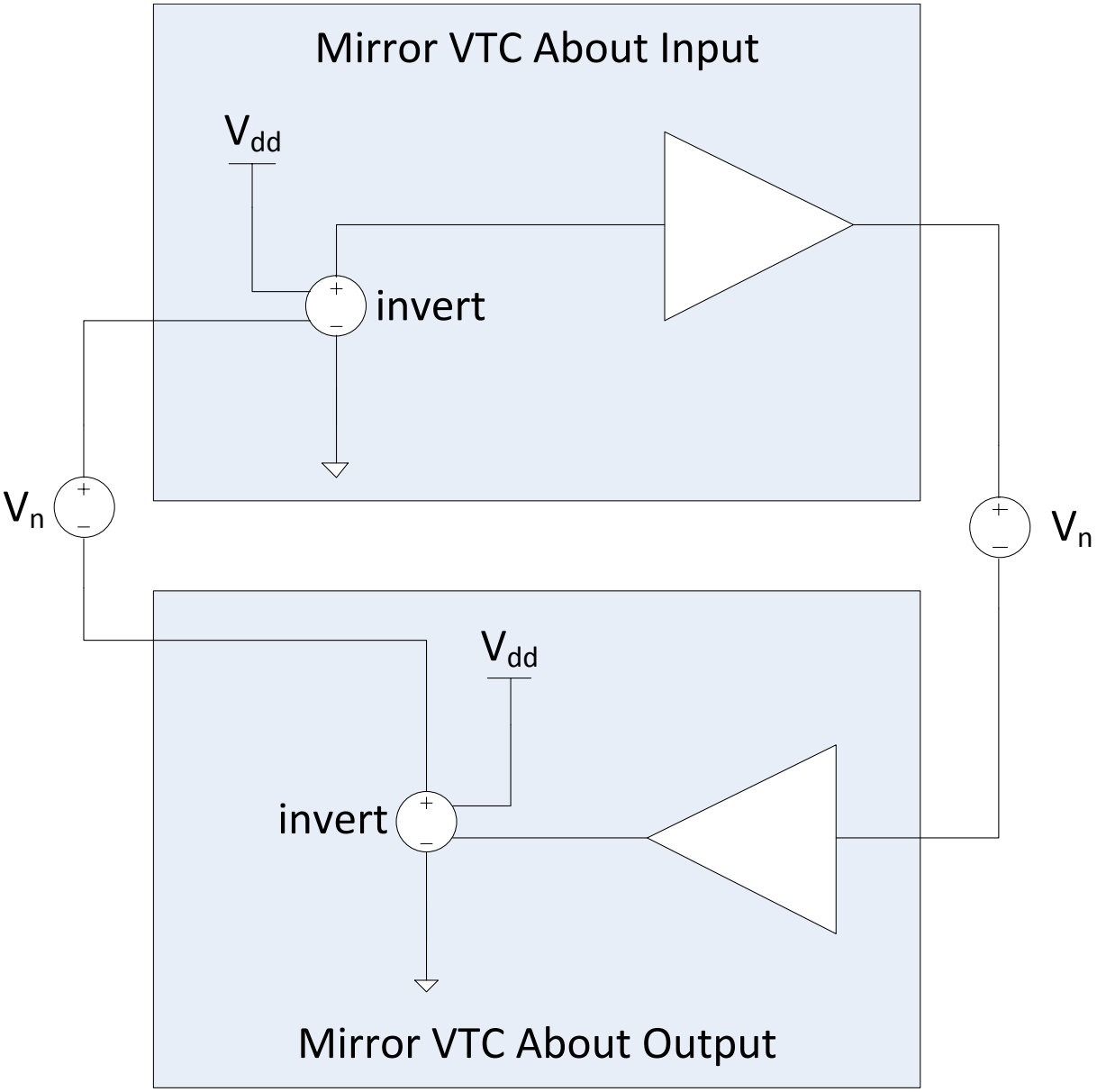

Figure 3: Simplified model of an infinite chain of alternating positive unate cells.

locked cells separated by voltage noise sources. Figure 2 illustrates a simplified model for two negative unate (inverting) cells, and Figure 3 shows a similar model for a pair of positive unate (non-inverting) cells. The difference between the two SNM models is that the input and output characteristics of positive unate cells are translated through inversion; the input and output characteristics of negative unate cells are referenced directly. In Figure 3, the model inverts the input of one non-inverting gate and inverts the output of the other positive unate gate using ideal voltage-controlled voltage sources (VCVS). The inverting VCVSs translates input and output voltage curves without altering the said curves.

The input-to-output voltage transfer curves (VTC) are extracted from each cell, and a butterfly plot is drawn by combining the VTC of one cell with the inverse VTC of the second cell; Figure 4 shows a butterfly plot of two negative unate curves. Inter-cell SNM is graphically calculated from the butterfly plot by drawing the two largest squares between the VTCs, choosing the smaller of the two squares, and measuring one of the sides of that square. Numerically calculating SNM requires translating the VTCs to a coordinate system that is rotated 45 degrees about the origin, which is orthogonal to the diagonals of the squares representing the inter-cell SNMs. The new coordinate system introduces u and v coordinates such that u is an independent variable and v represents the difference between the two VTCs, effectively defining the diagonals of each SNM square. The lengths of the SNM square diagonals are derived from the difference between the VTCs with respect to the 45-degree coordinate system. Each SNM is calculated from the square diagonal lengths.

Figure 5 is a butterfly plot of two non-inverting gates. In order to calculate SNM, the

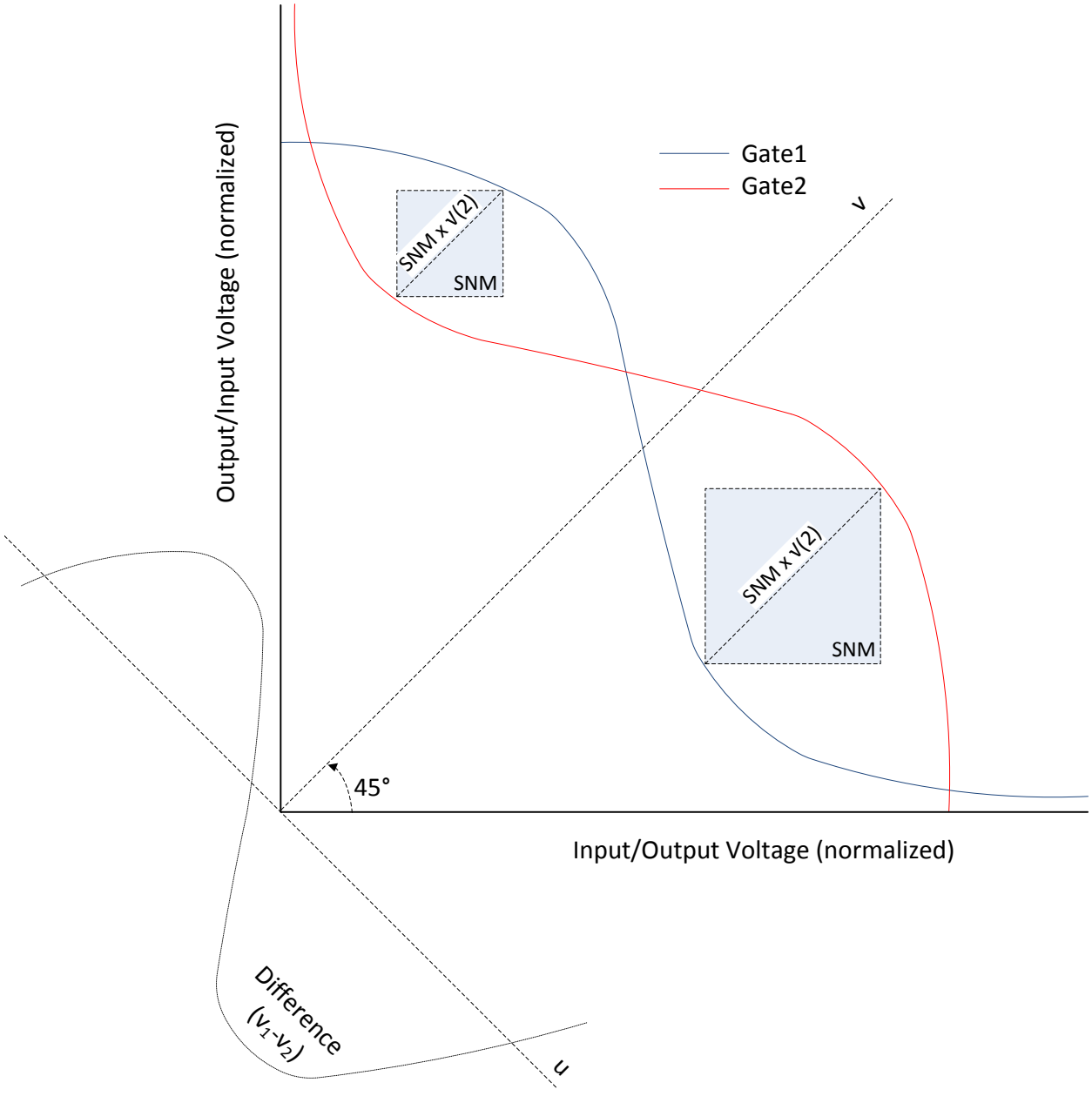

Figure 4: Input/output voltage transfer curves of two negative unate gates.

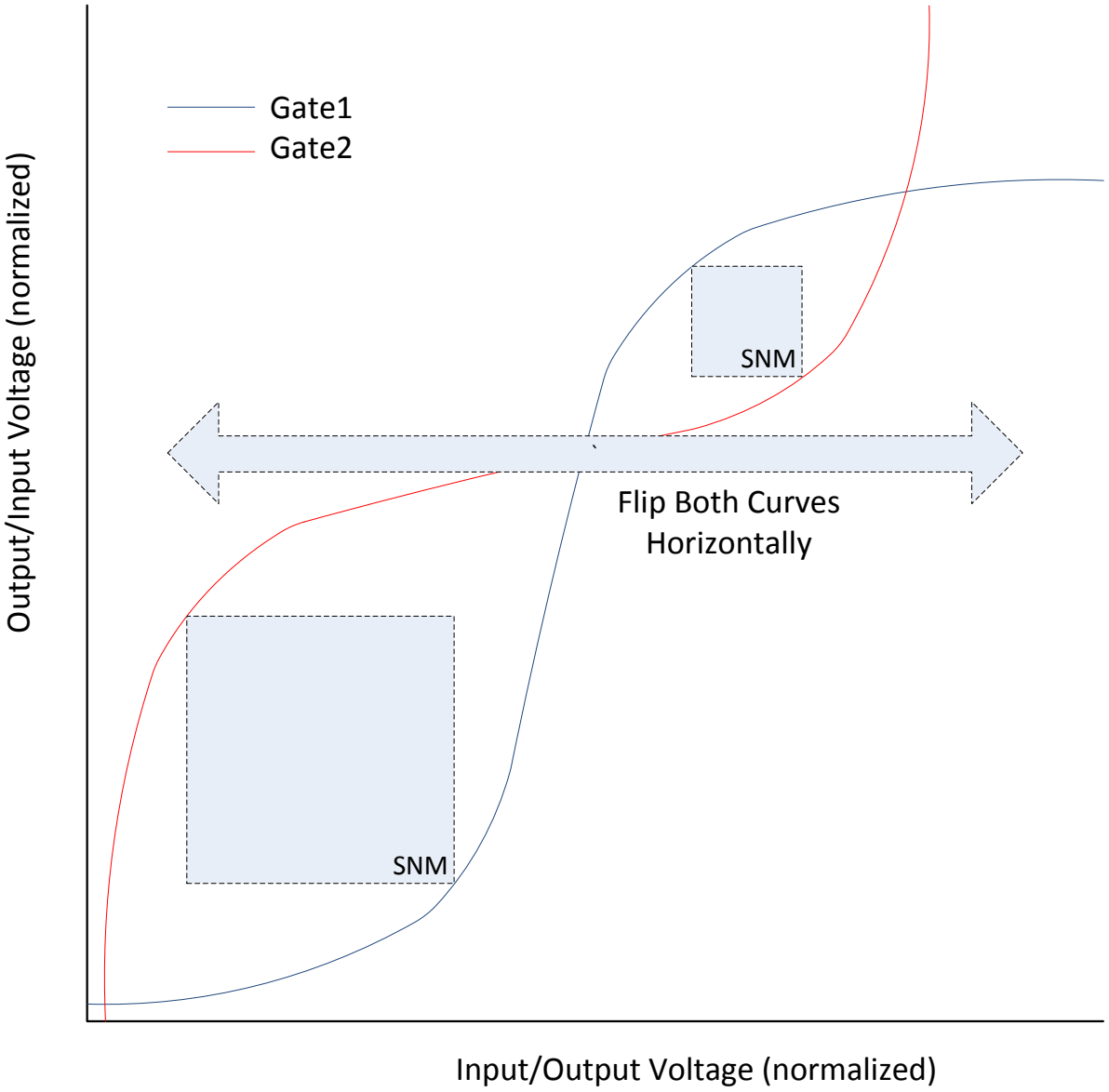

Figure 5: Input/output voltage transfer curves of two positive unate gates.

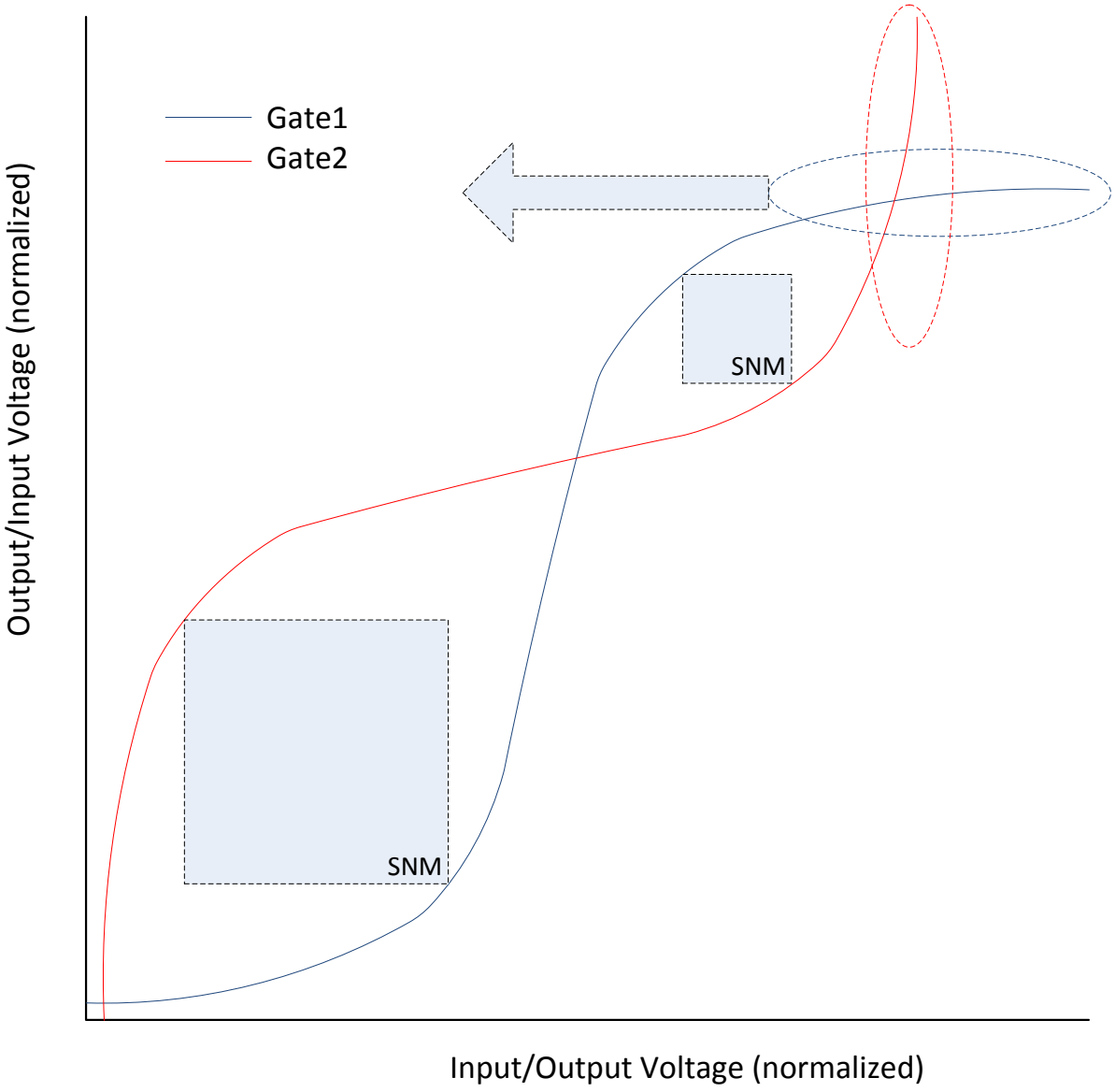

Figure 6: Translating positive unate voltage curves for SNM analysis.

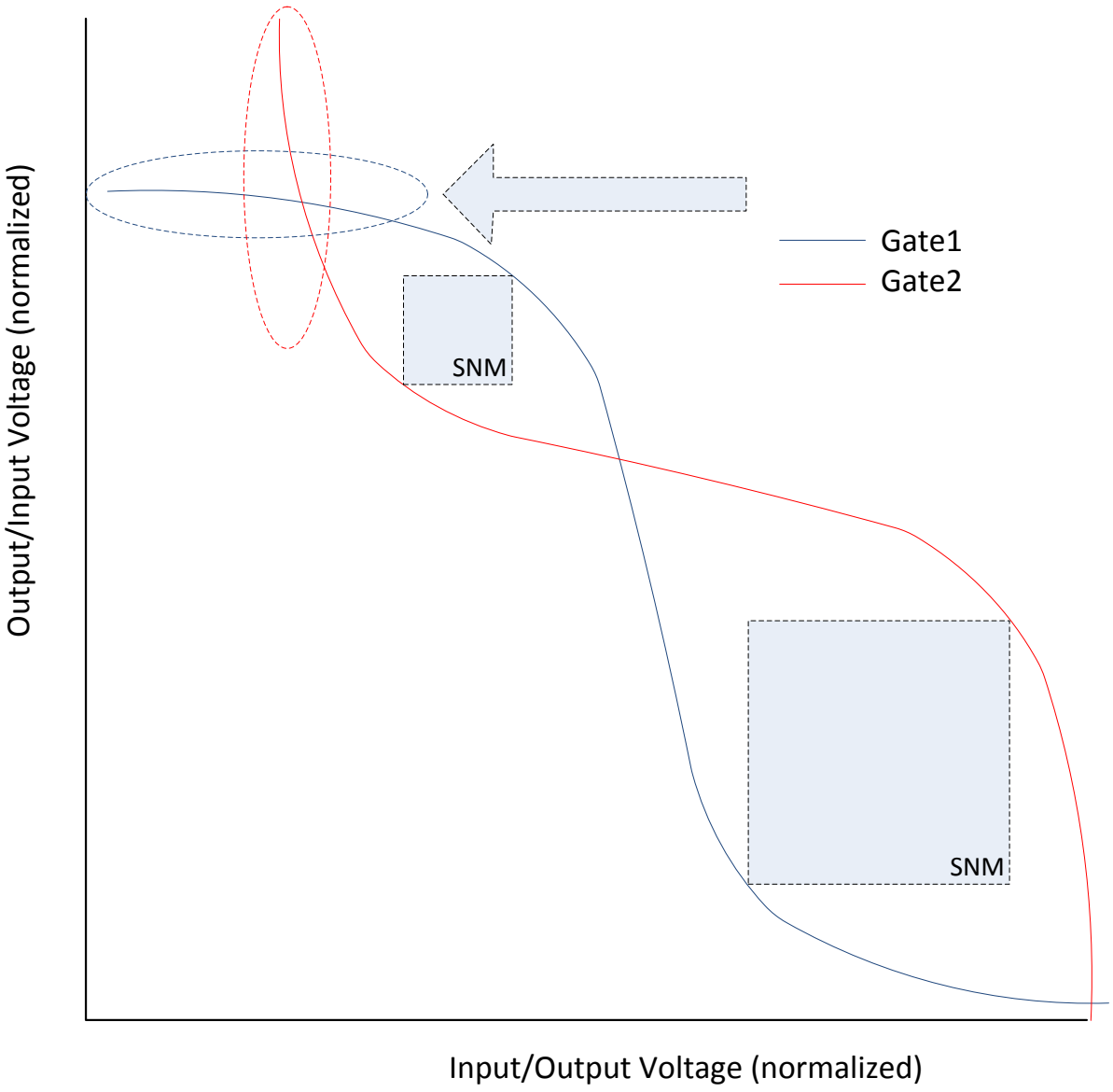

Figure 7: Positive unate curves after translation.

positive unate VTCs must be transformed into inverting VTCs before translation into the 45-degree coordinate system. Both curves are flipped horizontally, implying that the output from one gate (Gate 2 in Figure 5) and the input voltage into the other gate (Gate 1 in Figure 5) are to be inverted. Moreover, inversion implies taking the difference between the specified supply voltage, V_{dd}, and the VTC. Figure 6 and Figure 7 illustrate the data points on each VTC being inverted, highlighted in dashed circles. In effect, the size of each SNM squares remain unchanged and are measurable in the 45-degree coordinate system.

2.1.3 NULL CONVENTION LOGIC (NCL)

NCL is a solution that addresses systematic timing issues at subthreshold. It replaces synchronous pipelines with NCL registers, threshold gates, and multi-rail encoded signals. The primary motivation of switching to NCL is designing clockless systems where the asynchronous pipeline is controlled by data instead of external clock signals. Removing external clocks relaxes timing requirements in critical path analysis, clock transition time, and clock skew. The lack of external clocks also relieve the circuit of clock buffer trees, which are largely susceptible to process variations at subthreshold. In general, large-scale systems benefit more from NCL than small digital circuits because the performance offered by NCL is based on average timing instead of worst-case (critical path) timing.

NCL defines 27 fundamental threshold gates. A threshold gate asserts its output when its inputs satisfy a predefined Boolean expression. After asserting its output, the threshold gate waits until all inputs have cleared to logical zero before unasserting its output (i.e., hysteresis). Figure 8 is a general symbol of a threshold gate that counts the number of asserted inputs (n) against a

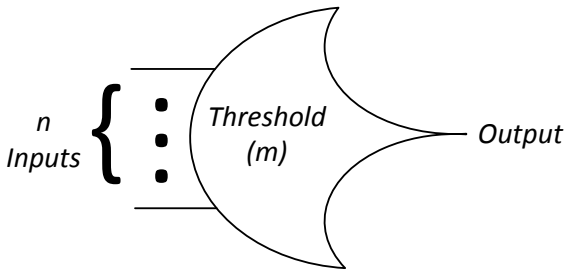

Figure 8: THmn gate symbol.

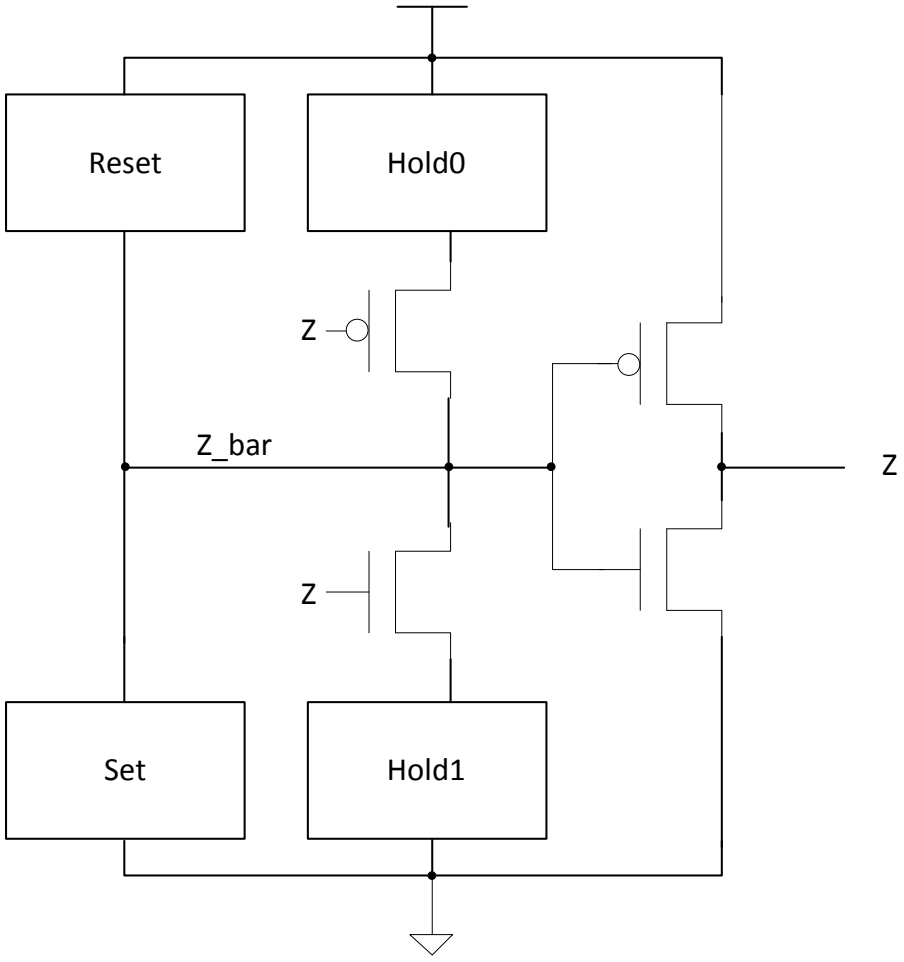

Figure 9: General threshold gate static CMOS schematic.

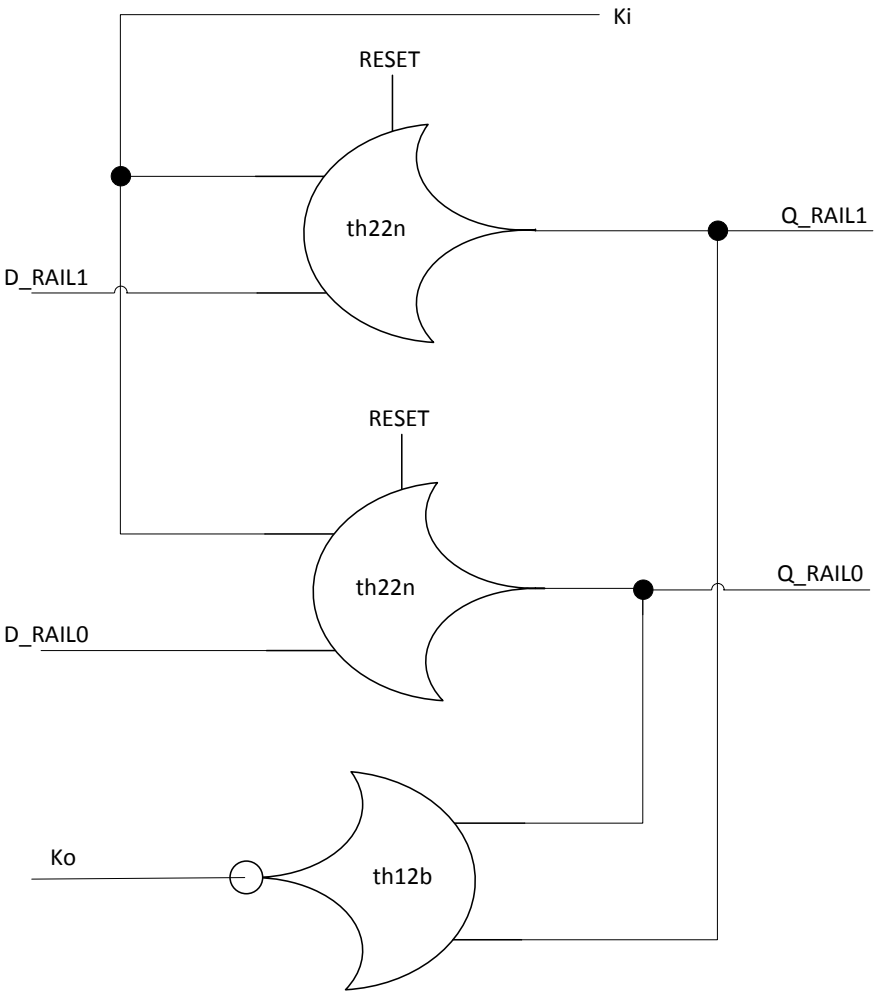

Figure 10: Dual-rail NCL register.

threshold value (m) required to assert the output. Figure 9 shows a general static CMOS schematic of a threshold gate, which contains blocks that set and hold the state of the gate output.

NCL registers manage data flow using each data signal's multi-rail encoding and output completion logic; Figure 10 shows a threshold gate schematic of a dual-rail NCL register. Multi-rail encoding specifies several signal states indicating presence or absence of data, and the NCL register detects whether or not data is available on a multi-rail encoded signal. The NCL register also monitors for output completion before changing its state.

Data flows through each stage of the NCL pipeline in a sequence of alternating DATA and

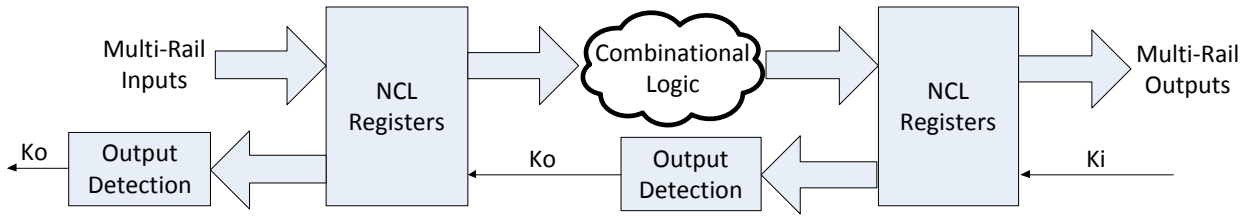

Figure 11: General NCL pipeline.

NULL cycles; Figure 11 illustrates a general NCL pipeline with its output completion logic. At the beginning of operation, the pipeline requests and waits for DATA by asserting each Ko signal. When DATA is fed into the pipeline through its first NCL register, the pipeline propagates DATA through each stage until the final NCL register outputs the processed data. Once the pipeline presents the processed data, the pipeline receives an acknowledgment through its Ki input, which propagates backwards through each stage's output completion logic. After the first NCL register receives acknowledgment of output completion, it requests for NULL by unasserting its Ko output. NULL is then fed into the pipeline and clears DATA in each stage, eventually causing the final NCL register to output NULL. The pipeline's Ki input is unasserted to indicate NULL output detection, which propagates back through the pipeline. The first NCL register receives NULL output completion and requests for DATA.

2.1.4 SCHMITT-TRIGGER GATE DESIGN

Proposed by [19, 20] for ultra-low voltage operation, an alternative to static CMOS gate design is based on the CMOS Schmitt-trigger inverter. The Schmitt-trigger gate structure increases SNM by suppressing inactive subthreshold current using positive output feedback. This structure introduces virtual power and ground rails that are controlled by the positive output feedback such that the virtual power rails follow the gate output, reducing the potential between the gate output

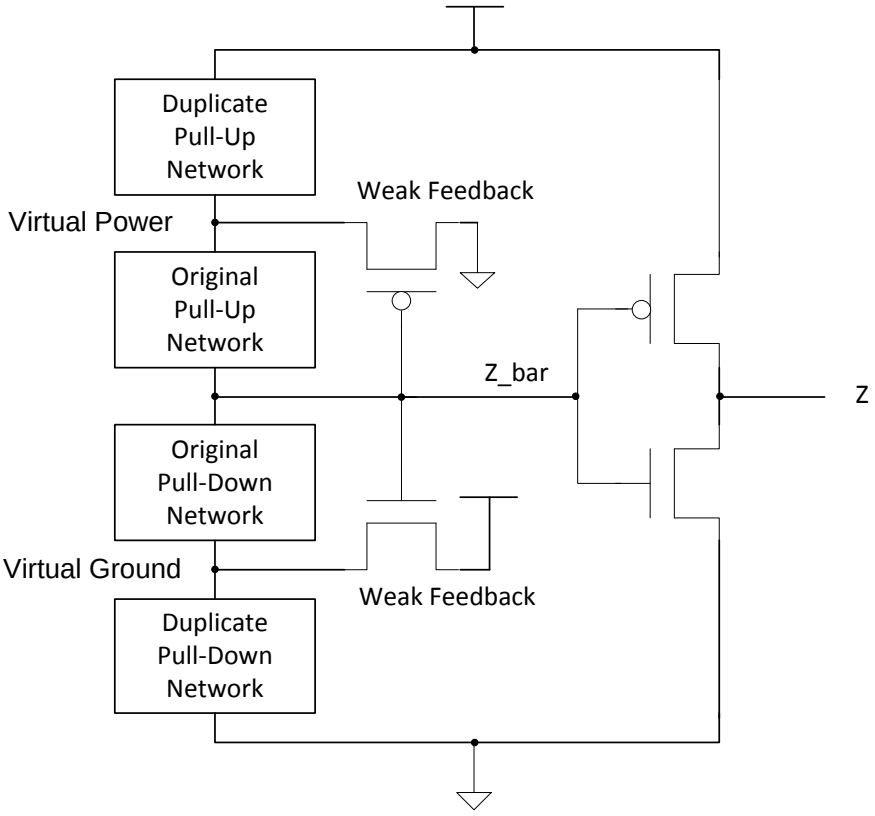

Figure 12: Schmitt-trigger architecture for a positive unate gate.

and the virtual power rails.

Figure 12 and Figure 13 show two similar architectures of a positive unate Schmitt-trigger gate. Each network is duplicated and feedback transistors are added to drive the virtual power rails to the opposite power rail. The difference between Figure 12 and Figure 13 is the type of transistor used for output feedback. Figure 12 is presented by [19, 20] and relies on weak feedback transistors to reduce leakage current from the gate output. Figure 13 swaps the feedback transistors to give a stronger output feedback because NMOS is used to drive virtual power to ground while a PMOS transistor drives virtual ground to supply. However, having a stronger output feedback imposes a limit on the maximum supply voltage because the output feedback becomes too strong at high supply voltages, locking the gate output to a certain state.

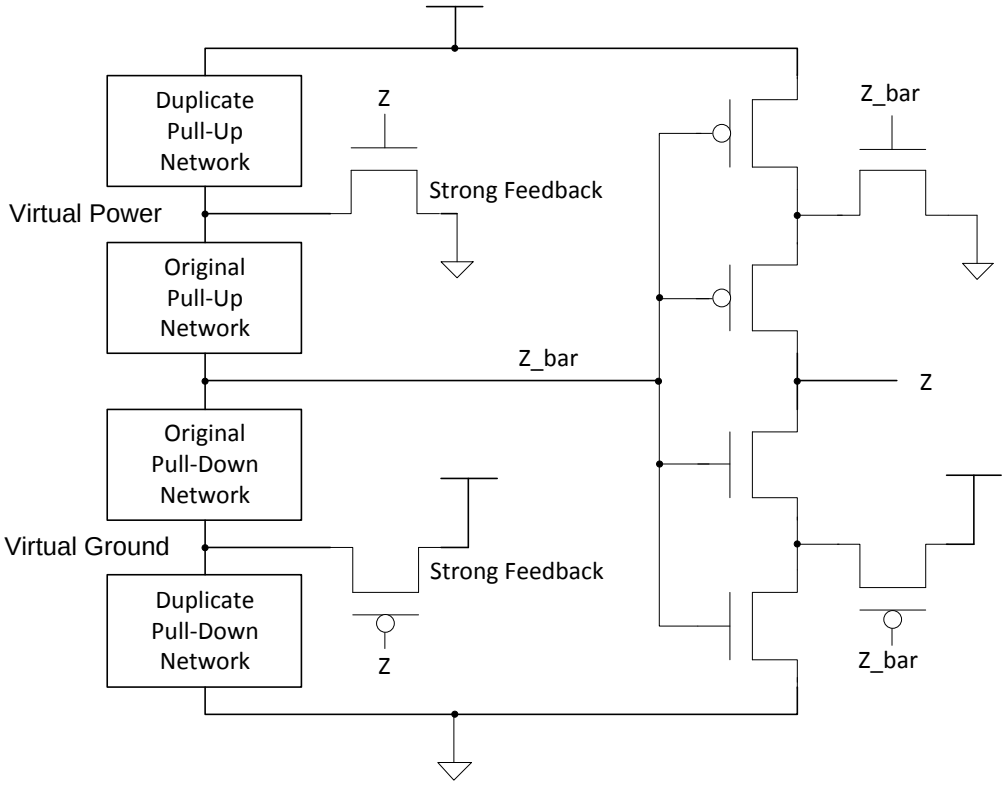

Figure 13: Alternative Schmitt-trigger architecture for a positive unate gate.

Both Schmitt-trigger architectures operate transparently, meaning that the functionality of the gate and its inputs and output remain unchanged. Unlike static CMOS gates, all Schmitt-trigger gates have an internal state kept by the positive output feedback loop. Virtual power and ground follow the gate output and are either driven by the respective duplicate network or by the corresponding feedback transistor. The gate inputs control both the original and duplicate networks, and in order for a Schmitt-trigger gate to change state, the active duplicate network must overpower the corresponding feedback transistor and activate the respective virtual power rail. Activating a virtual power rail means driving that virtual rail close to the actual power rail. Conversely, the inactive duplicate network must allow the corresponding feedback transistor to drive the respective virtual power rail towards the opposite supply voltage. For example, a Schmitt-

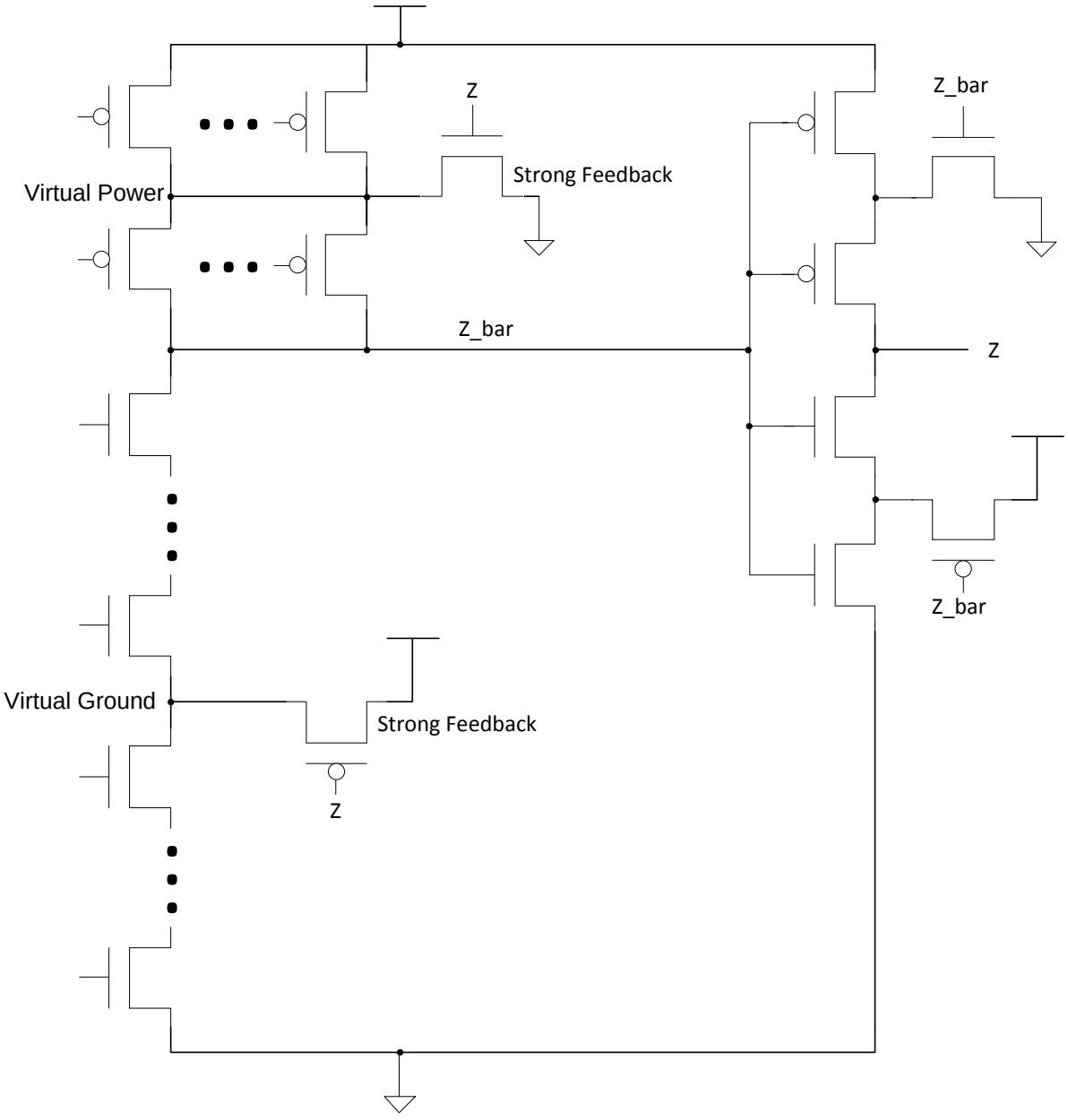

Figure 14: Schmitt-trigger AND gate.

trigger AND gate, illustrated in Figure 14, is given an all logic one input pattern and its output is logic one. Virtual ground is approximately the same voltage as the ground rail due to the active duplicate pull-down network, and virtual power is at a mid-supply voltage level because the pull-up networks are inactive and the pull-down feedback transistor is active. When one of the AND gate inputs transition to logic zero, the pull-down networks deactivate while the pull-up networks open one or more DC paths to power. The duplicate pull-up network drives the virtual power node close to the supply voltage level, and the original pull-up network drives the immediate output node (Z_bar in Figure 14) close to the same voltage as the virtual power node. Over time, the immediate output rises and deactivates the pull-down feedback transistor and simultaneously activates the pull-up feedback transistor, causing virtual ground to rise in voltage. The state of the AND gate becomes that the immediate output and virtual power are approximately at the supply voltage level while virtual ground is at a mid-supply voltage. Whether the AND gate uses weak or strong feedback transistors affects each duplicate network's ability to overpower their respective feedback transistor, and it modifies the output feedback loop such that weak feedback transistors tie directly to the output node and strong feedback transistors require an inverter at the gate output.

2.2 DESIGN METHODOLOGY AND TEST CIRCUITS

This research focuses on several techniques and processes for ultra-low voltage operation: bulk-silicon and FD-SOI, forced transistor stacking, static CMOS and Schmitt-trigger gates, synchronous logic and NCL. Steps are taken to employ these techniques in building four test circuits: synchronous and NCL four-bit, two-stage ripple-carry adders, and synchronous and NCL IEEE single-precision floating-point coprocessor.

IBM's 130 nm process and MIT Lincoln Laboratory's 150 nm process are chosen for the bulk-silicon and FD-SOI processes respectively. Eight gate libraries are constructed to implement synchronous and NCL gates on each process using static CMOS and Schmitt-trigger structures. The static CMOS libraries are first built before implementing the Schmitt-trigger gate libraries. The static CMOS gates use forced transistor stacking to balance pull-up and pull-down drive strengths. The Schmitt-trigger gates are then constructed from the static CMOS gates using both Figure 12 and Figure 13 as the base architectures.

Prior to adopting the 150 nm FD-SOI transistor models, the NMOS and PMOS transistors are characterized by extracting current and voltage curves followed by fitting curves generated by the Berkeley Short-Channel Insulated-Gate Field-Effect Transistor (IGFET) Model (BSIM) for SOI processes (BSIMSOI). Modeling SOI devices differs from bulk-silicon processes due to the buried oxide layer. The buried oxide layer introduces floating-body MOSFETs, meaning that the transistor body has no external contact to control its bias. Instead, the bias of the body must be predicted using a floating-body current model. BSIMSOI estimates floating-body currents for both partially-depleted and fully-depleted SOI devices. Extracting SOI model parameters for above-threshold operation shares a similar methodology as bulk-silicon modeling. Several differences in methodology are sweeping the back-gate (wafer) voltage and sweeping the body bias using body-contacted SOI devices such as H-gate transistors. Additional steps are required to accurately model transistors in the subthreshold region. Measurements are taken to sweep the drain and gate voltages over a small range, giving $I_d(V_d)$-V_g curves based on a subthreshold supply. After extracting model parameters from above-threshold measurements, the model parameters are fine-tuned to match the

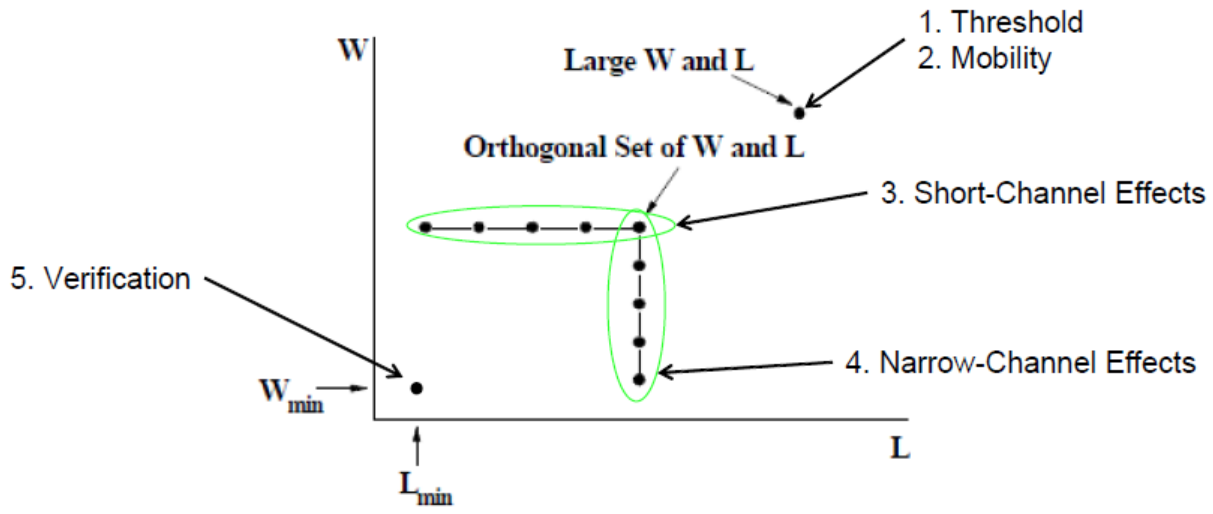

Figure 15: Steps taken to characterize MIT Lincoln Laboratory's FD-SOI transistors.

subthreshold I-V curves.

Figure 15 outlines the steps taken to characterize the FD-SOI transistors of different sizes. The 150 nm FD-SOI transistors are fabricated on a die, shown in Figure 16, as an array of transistors of varying channel lengths and widths, illustrated in Figure 17. Channel lengths range from 150 nm to 8 μm while channel widths vary from 500 nm to 8 μm. The long-channel threshold voltage and carrier mobility are extracted from $I_d(V_g)$ log and linear curves, shown in Figure 18 and Figure 19. NMOS has a long-channel threshold voltage of 430 mV and mobility of 306 $m^2/(Vs)$ while PMOS has a long-channel threshold voltage of -320 mV and mobility of 150 $m^2/(Vs)$, which indicates that NMOS is approximately two times stronger than PMOS. Figure 20 shows $I_d(V_g)$-V_b curves related to modeling short-channel, narrow-width, and back-wafer effects. Figure 21 shows curves measured from the subthreshold region (below 500 mV) while Figure 22 shows curves measured from supply voltages below 100 mV. The subthreshold curves provide for accurate and linear drain current characteristics at low supply voltages.

For model verification and to briefly characterize process variation, the minimum sized

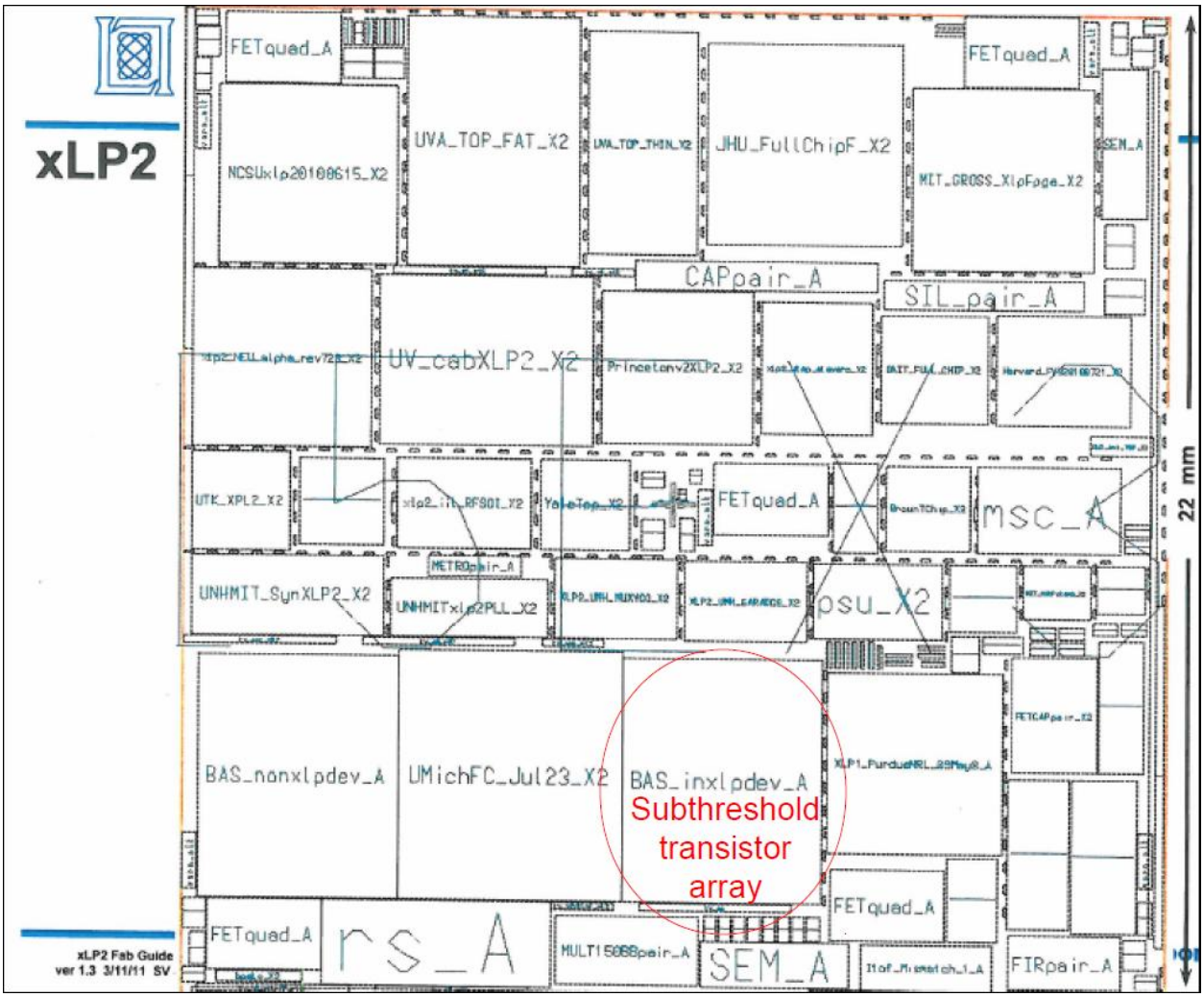

Figure 16: MIT Lincoln Laboratory FD-SOI die.

Figure 17: MIT Lincoln Laboratory FD-SOI transistor array.

transistor (500 nm by 150 nm) is measured across several copies of the FD-SOI die. Figure 23 and Figure 24 show log and linear-scale $I_d(V_g)$ curves derived from a single minimum sized transistor, and Figure 25 and Figure 26 show measurements taken from multiple die and compared to curves in Figure 23 and Figure 24. The discrepancy between the BSIMSOI model and measurements of the minimum sized transistor closely matches variations present in the drain current measured across each FD-SOI die.

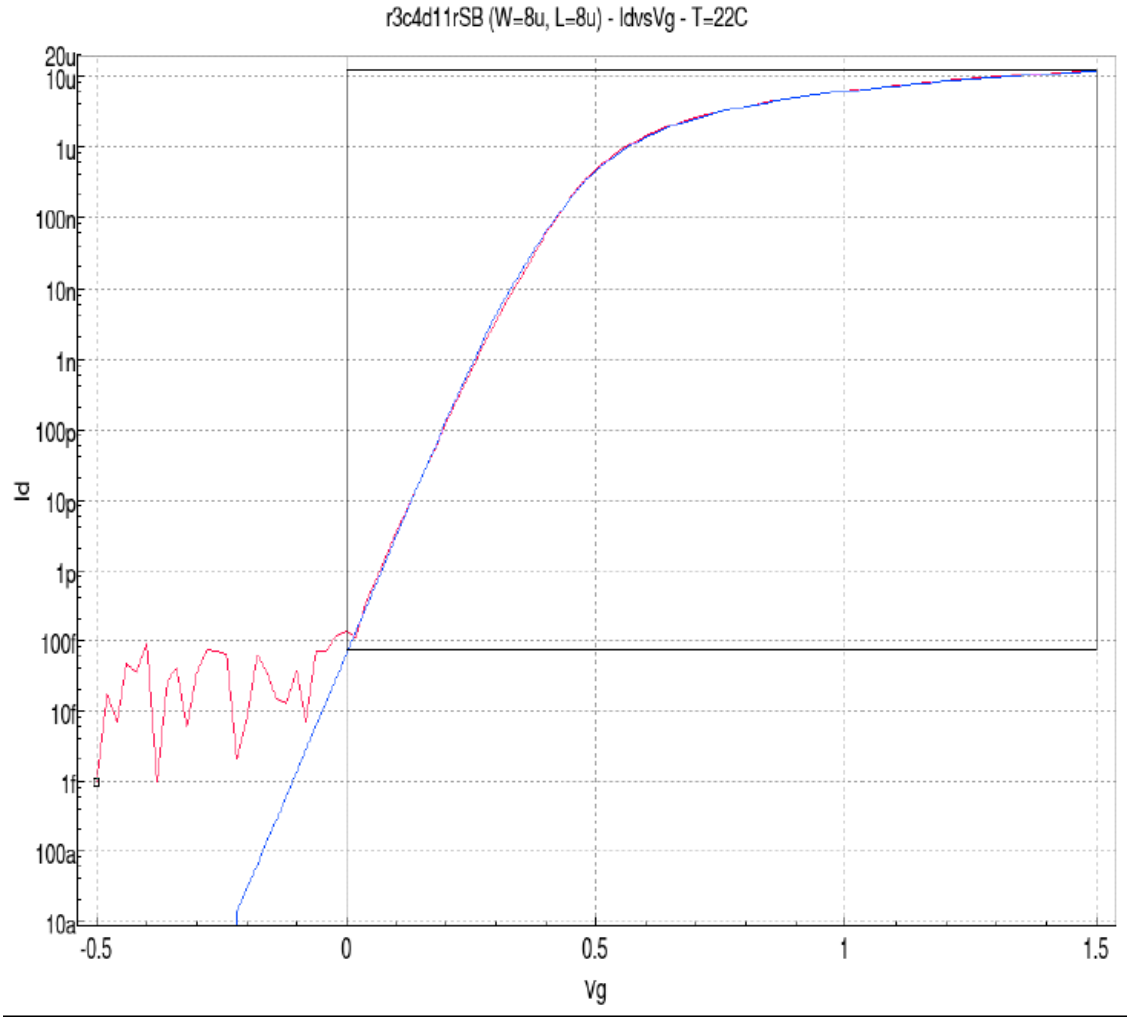

Figure 18: $I_d(V_g)$ **log-scale curve of the 8** μ**m by 8** μ**m FD-SOI NMOS transistors; measured curves (red), BSIMSOI curves (blue).**

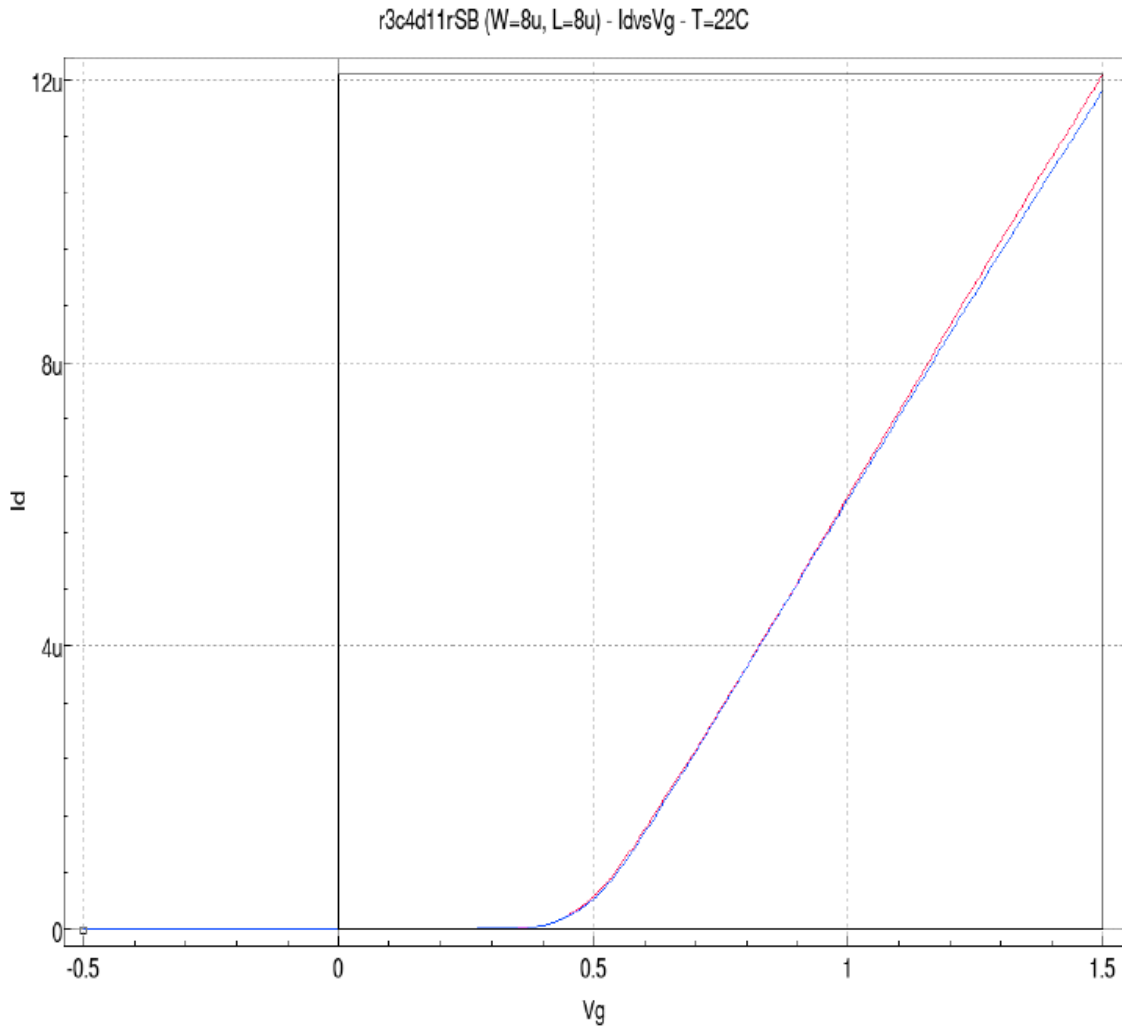

Figure 19: $I_d(V_g)$ **linear-scale curve of the 8** μ**m by 8** μ**m FD-SOI NMOS transistors; measured curves (red), BSIMSOI curves (blue).**

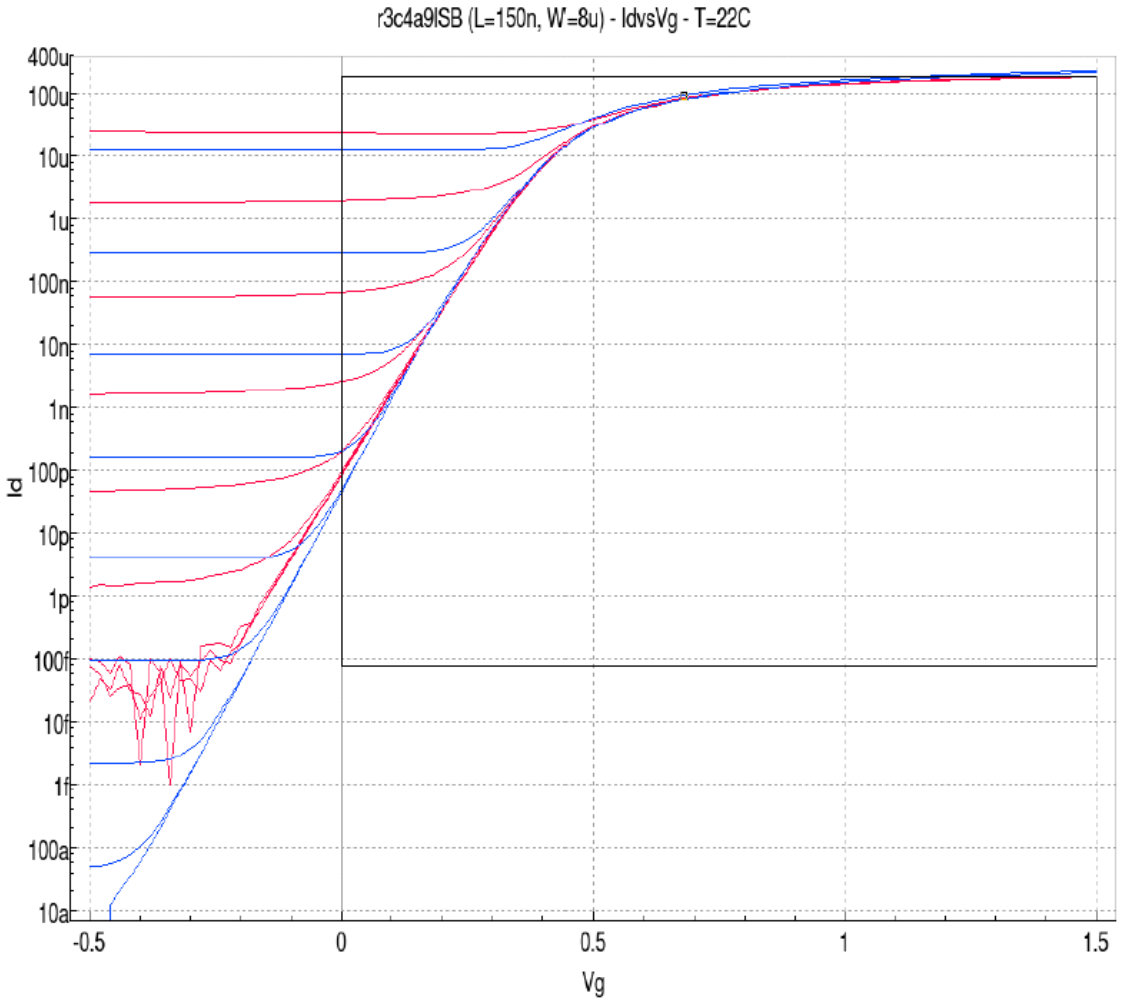

Figure 20: $I_d(V_g)$**-**V_b **curves of the 8** μ**m by 150 nm FD-SOI NMOS transistors; measured curves (red), BSIMSOI curves (blue).**

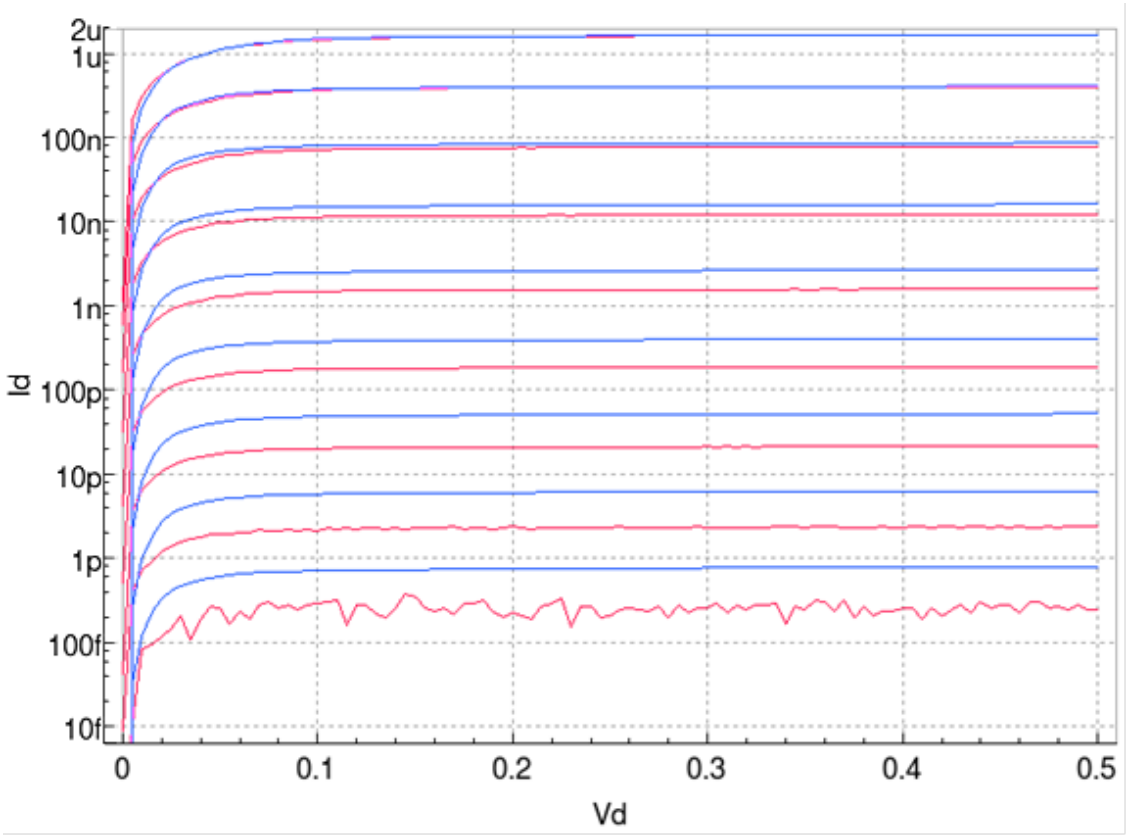

Figure 21: $I_d(V_d)$**-**V_g **curves (log-scale) of the 150 nm FD-SOI NMOS transistors at below 500 mV; measured curves (red), BSIMSOI curves (blue).**

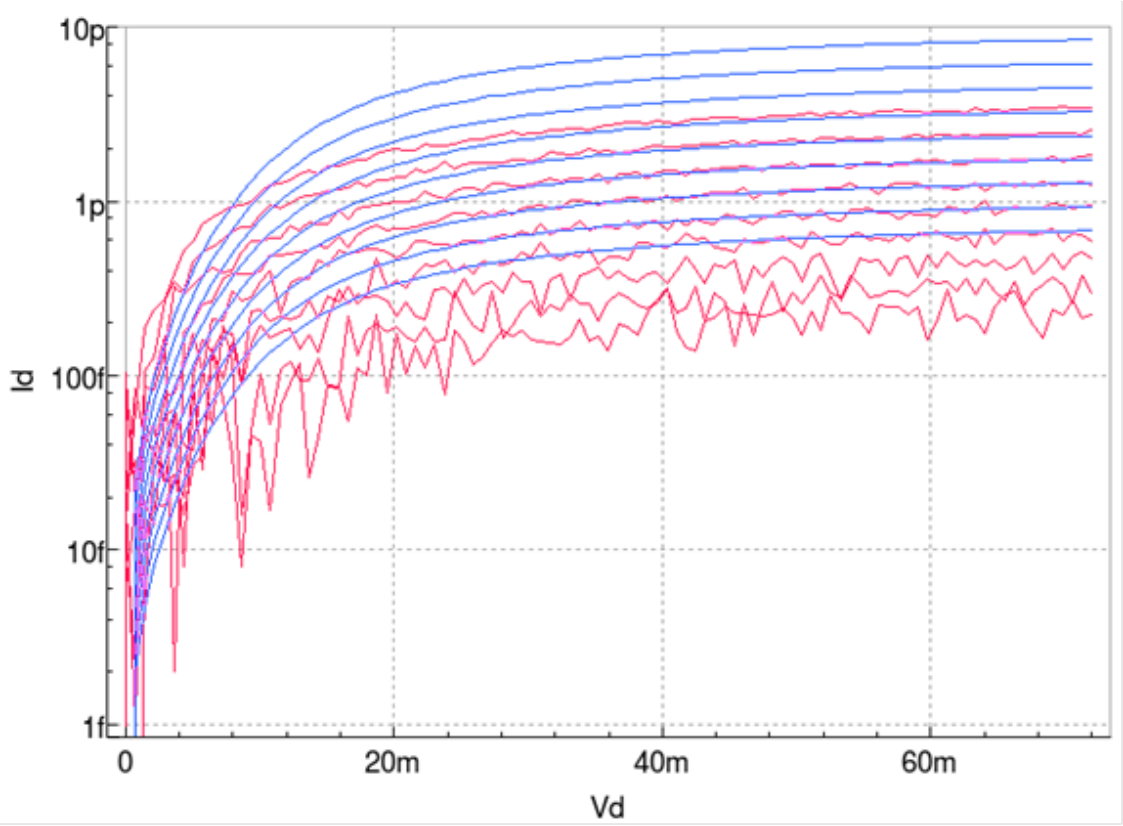

Figure 22: $I_d(V_d)$**-**V_g **curves (log-scale) of the 150 nm FD-SOI NMOS transistors at below 100 mV; measured curves (red), BSIMSOI curves (blue).**

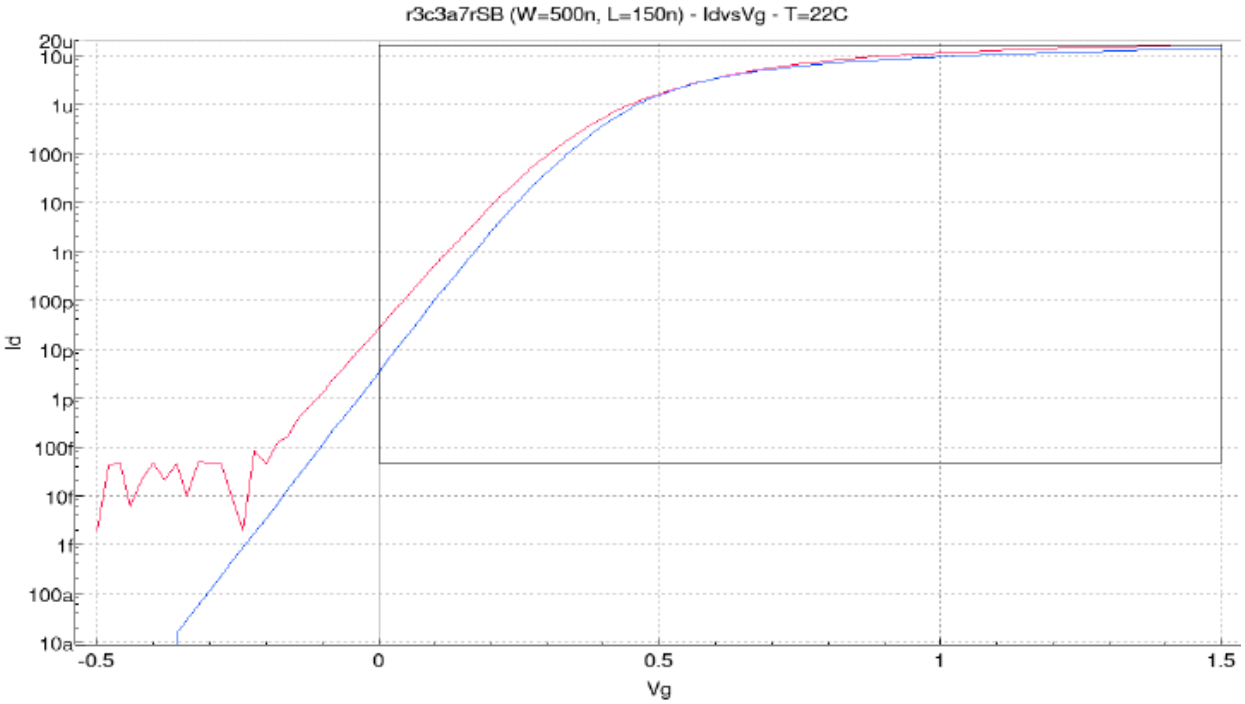

Figure 23: $I_d(V_g)$ **log-scale curves of the 500 nm by 150 nm FD-SOI NMOS transistors; measured curves (red), BSIMSOI curves (blue).**

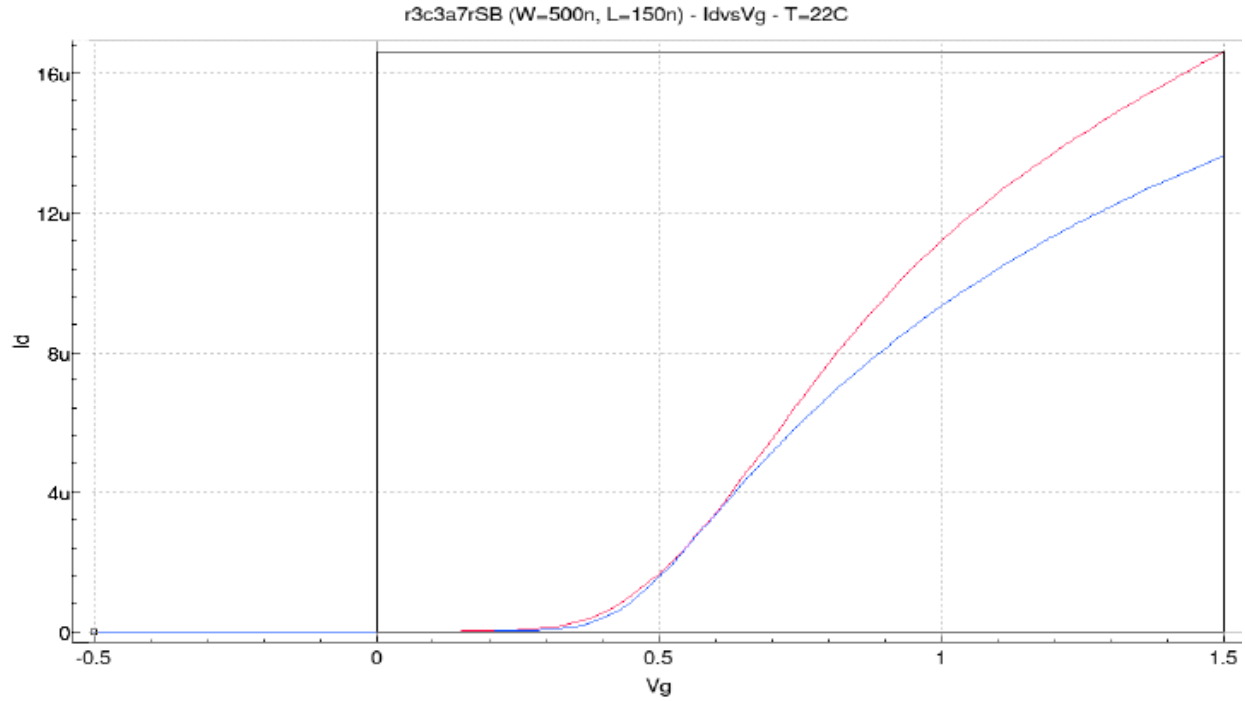

Figure 24: $I_d(V_g)$ **linear-scale curves of the 500 nm by 150 nm FD-SOI NMOS transistors; measured curves (red), BSIMSOI curves (blue).**

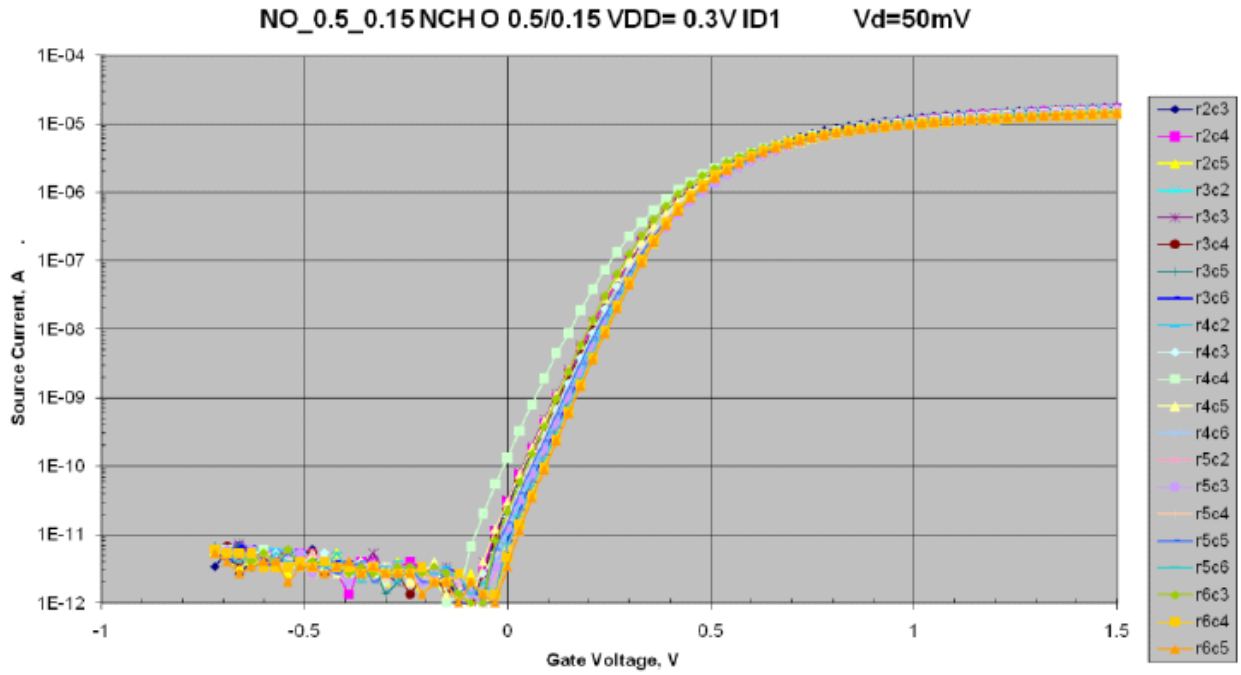

Figure 25: $I_d(V_g)$ **log-scale curves of the 500 nm by 150 nm FD-SOI NMOS transistors across multiple die.**

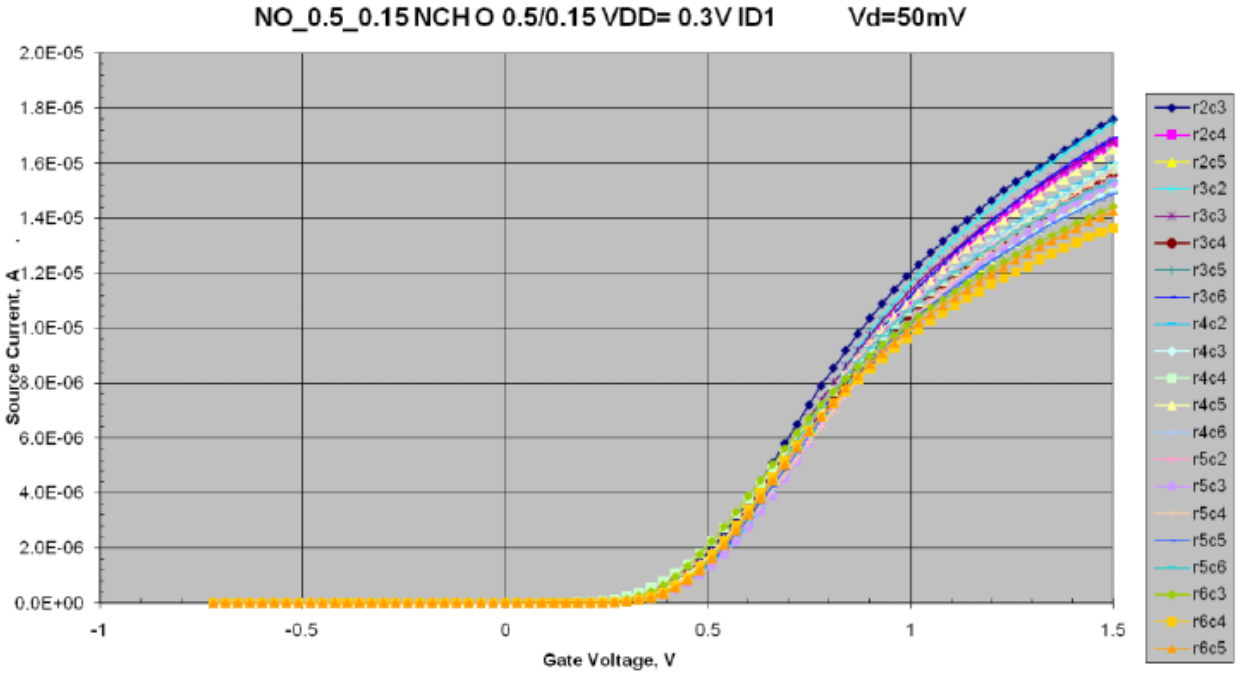

Figure 26: $I_d(V_g)$ **linear-scale curves of the 500 nm by 150 nm FD-SOI NMOS transistors across multiple die.**

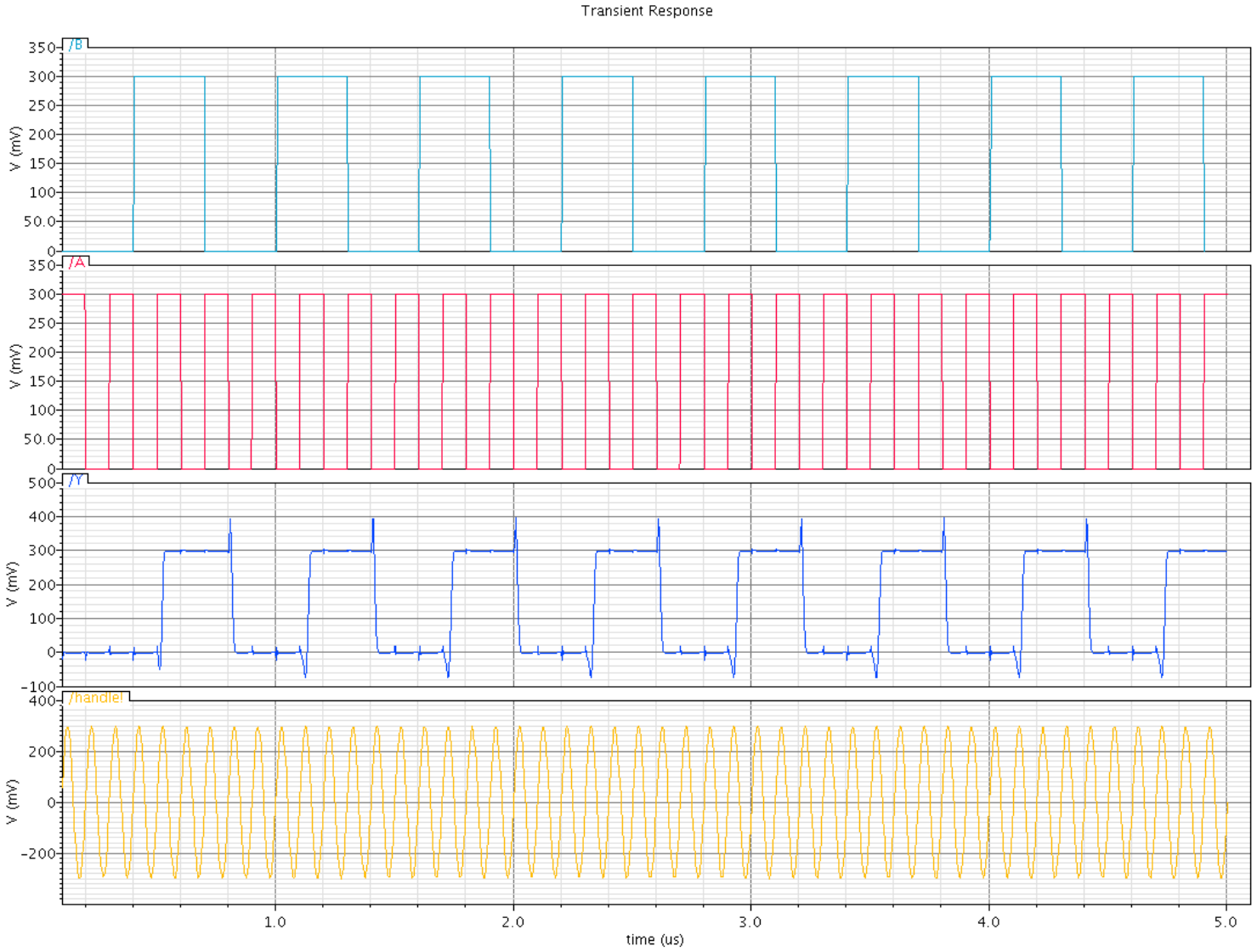

Figure 27: Transient analysis of a static TH22 gate on the FD-SOI process at 300 mV.

The FD-SOI models is further analyzed through a test threshold gate. Figure 27 shows transient analysis of a TH22 threshold gate running at a 300 mV supply voltage. Both the inputs and back wafer voltages oscillate to observe effects on the gate output. While the threshold functions correctly, the back wafer voltage (named handle! in Figure 27) creates noise in the gate output when it transitions. From the BSIMSOI equations, the back wafer voltage causes the threshold voltage to shift, depending on narrow and short channel effects. The threshold voltage of each 150 nm transistor is inversely related to the back wafer voltage.

The reason for choosing IBM's 130 nm process is that its channel length is comparable to MIT Lincoln Laboratory's 150 nm FD-SOI process. Another reason is that it is a mature bulk-silicon process that supports statistical parameters for process variation, unlike the 150 nm FD-SOI process. In contrast, the 150 nm FD-SOI models uses constant model parameter values based on BSIMSOI. Statistical parameters essentially parameterize certain BSIM model values that relate to physical dimensions and material characteristics. Model parameters affecting threshold voltage include width, length, channel thickness, gate oxide thickness, channel doping, and terminal doping.

Constructing the static CMOS libraries involves balancing pull-up and pull-down resistance. Forced transistor stacking is used in lieu of increased channel length in order to increase resistance through certain DC paths. Balancing the pull-up and pull-down resistances for the synchronous library assumes the drive strength of an optimized inverter that is minimally sized. Transient analysis on this inverter reveals relative drive strengths of the PMOS and NMOS transistors; Figure 28–33 show the steps in finding the optimum PMOS and NMOS transistor sizes for the

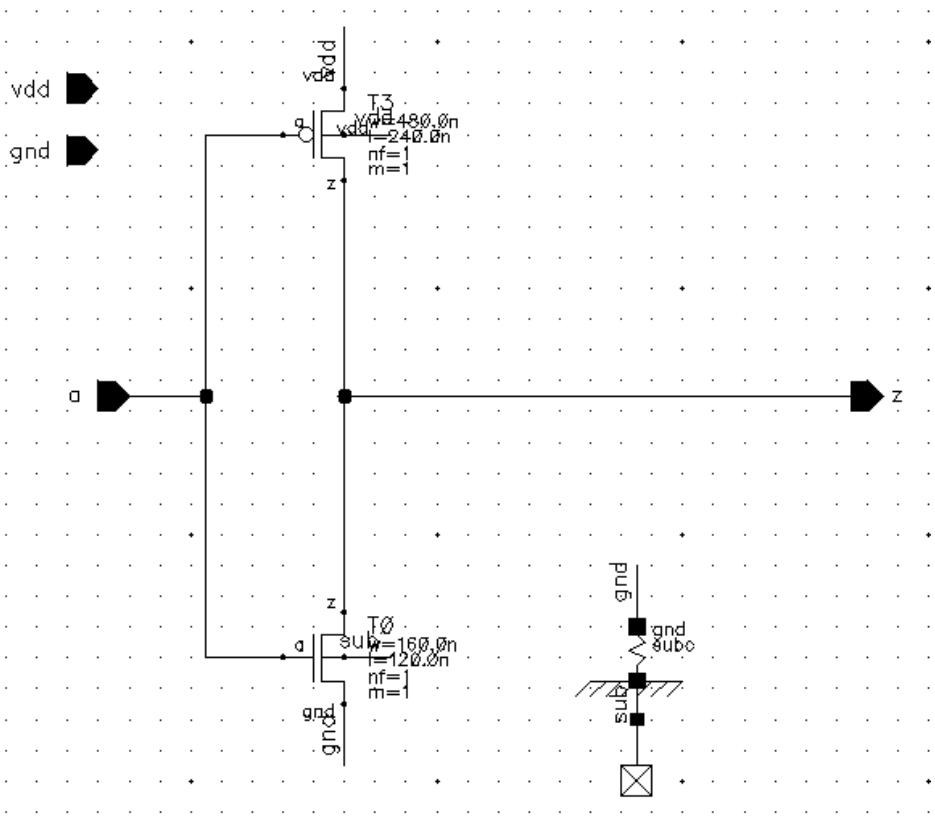

Figure 28: Optimized inverter on bulk-silicon.

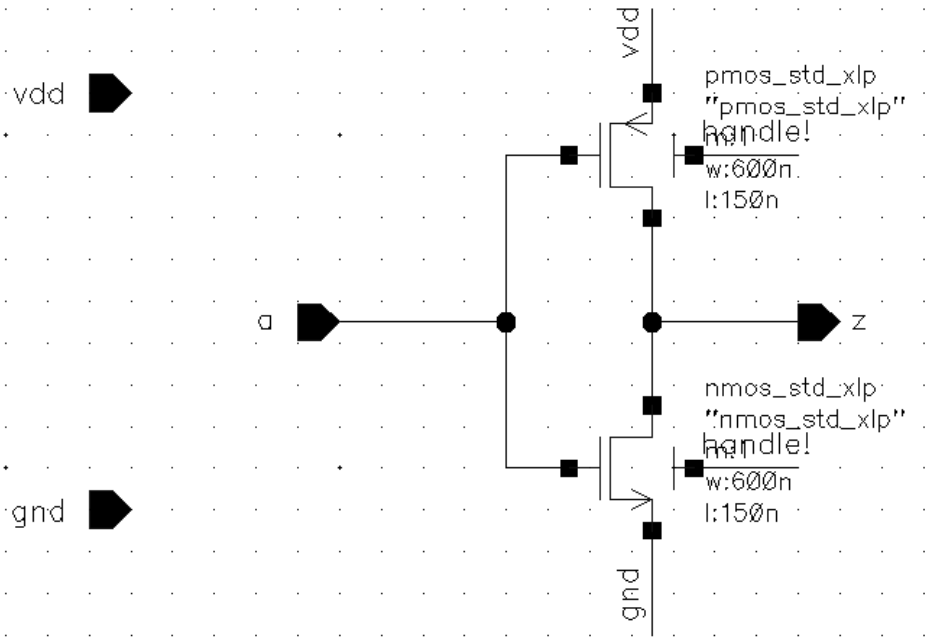

Figure 29: Optimized inverter on FD-SOI.

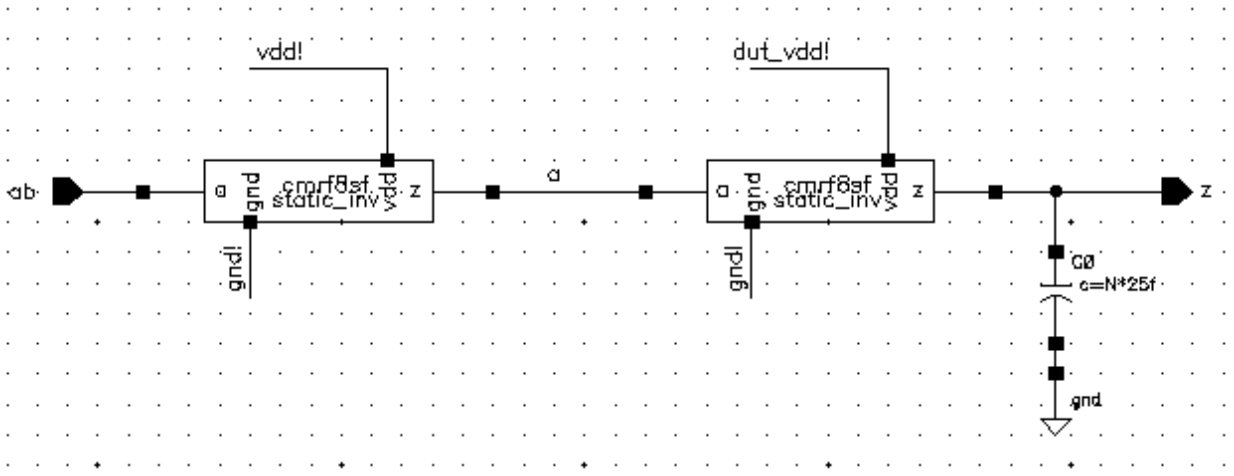

Figure 30: Testbenches of optimized inverters on bulk-silicon with 25 fF load capacitor.

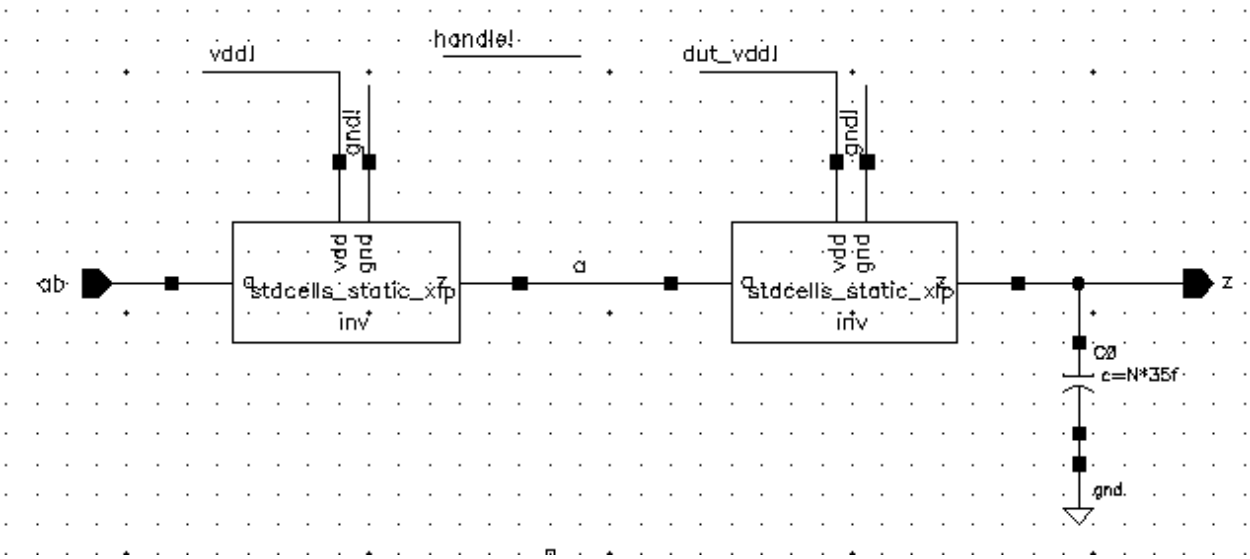

Figure 31: Testbenches of optimized inverters on FD-SOI with 35 fF load capacitor.

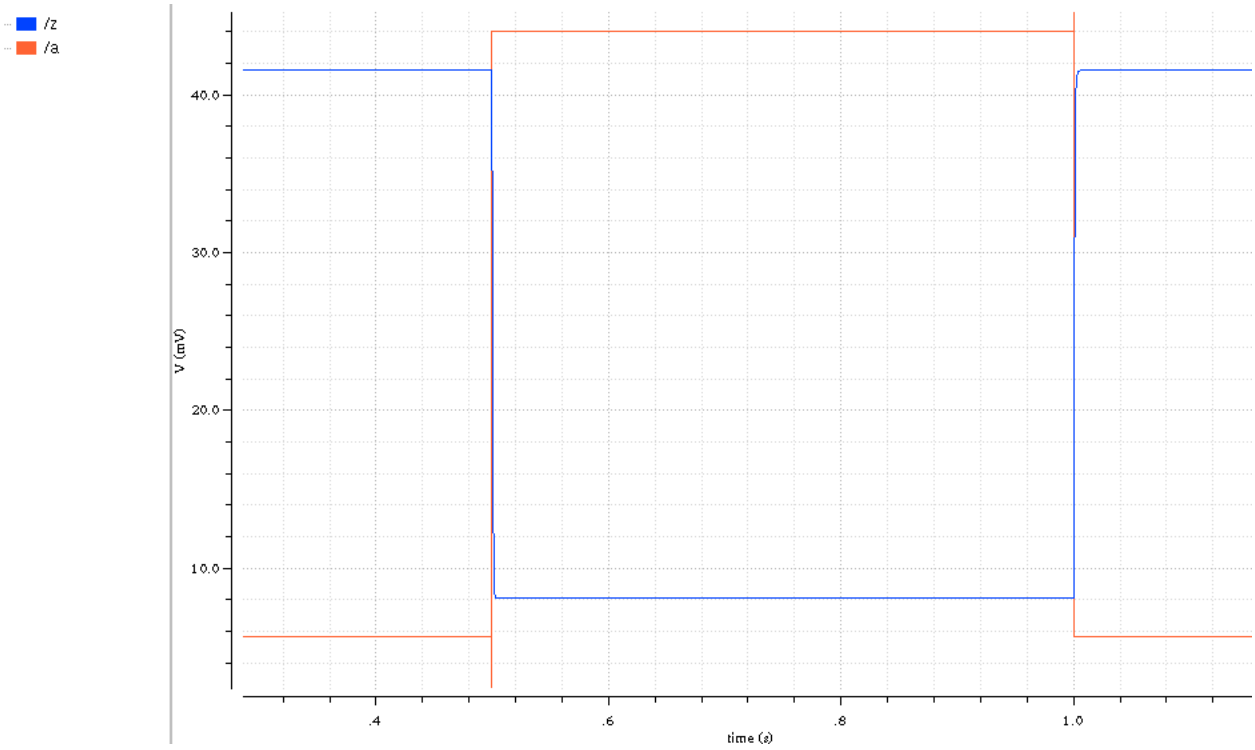

Figure 32: Transient analysis of optimized inverters at 50 mV on bulk-silicon with rise time of 1.7 ms and fall time of 1.6 ms.

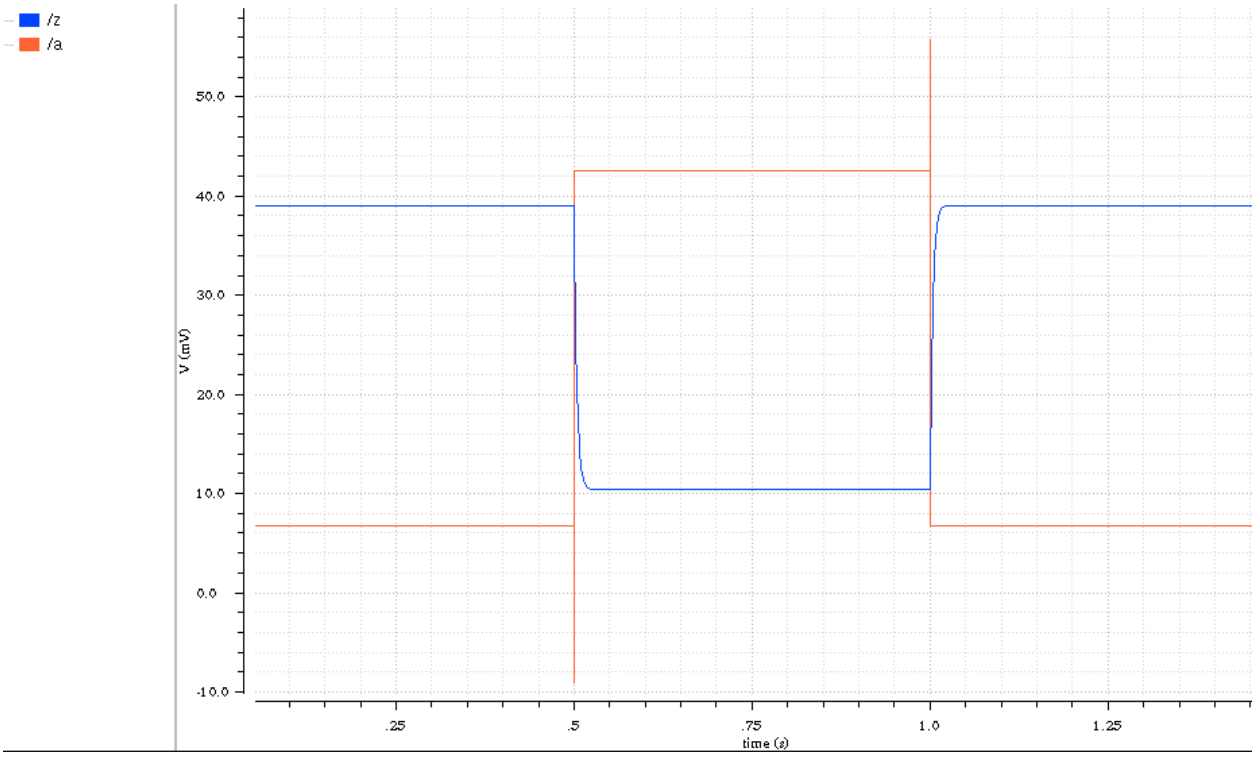

Figure 33: Transient analysis of optimized inverters at 50 mV on FD-SOI (right) with rise time of 5.1 ms and fall time of 4.9 ms.

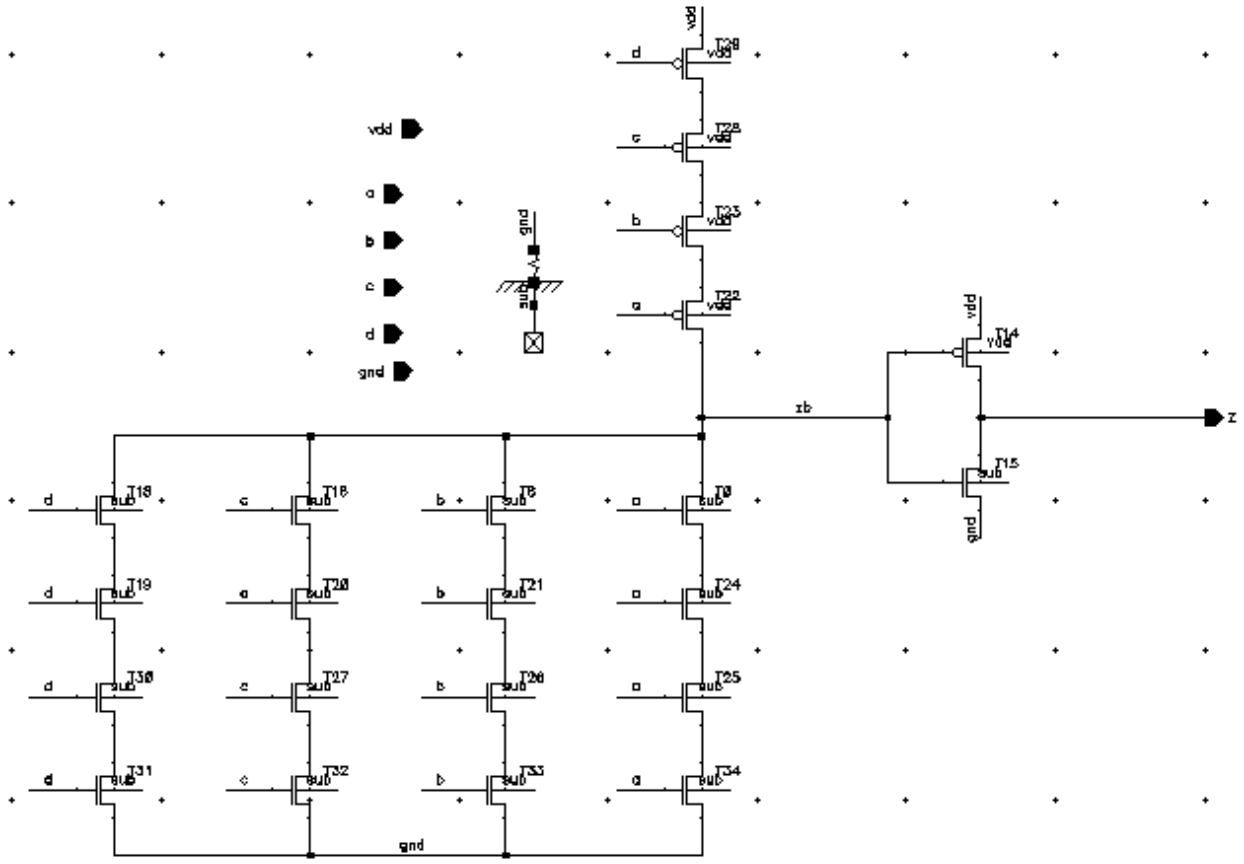

Figure 34: Four-input static OR gate schematic on the bulk-silicon process.

bulk-silicon and FD-SOI processes. Each testbench drives the gate with realistic input waveforms and places an output load representing fan-out on the gate. Transient analyses show approximately identical rise and fall transition times. For both processes, the NMOS transistor is minimum sized while for the bulk-silicon process, the PMOS transistor's channel length is doubled and channel width is multiplied by four; for the FD-SOI process, the PMOS transistor is sized the same as the NMOS transistor. The reason for doubling the length of the bulk-silicon PMOS transistor is to exploit channel length modulation present in its BSIM model; increasing the channel width further increases the PMOS drive strength against a bulk-silicon NMOS transistor.

With the optimum PMOS and NMOS transistor dimensions, each gate is built using the said

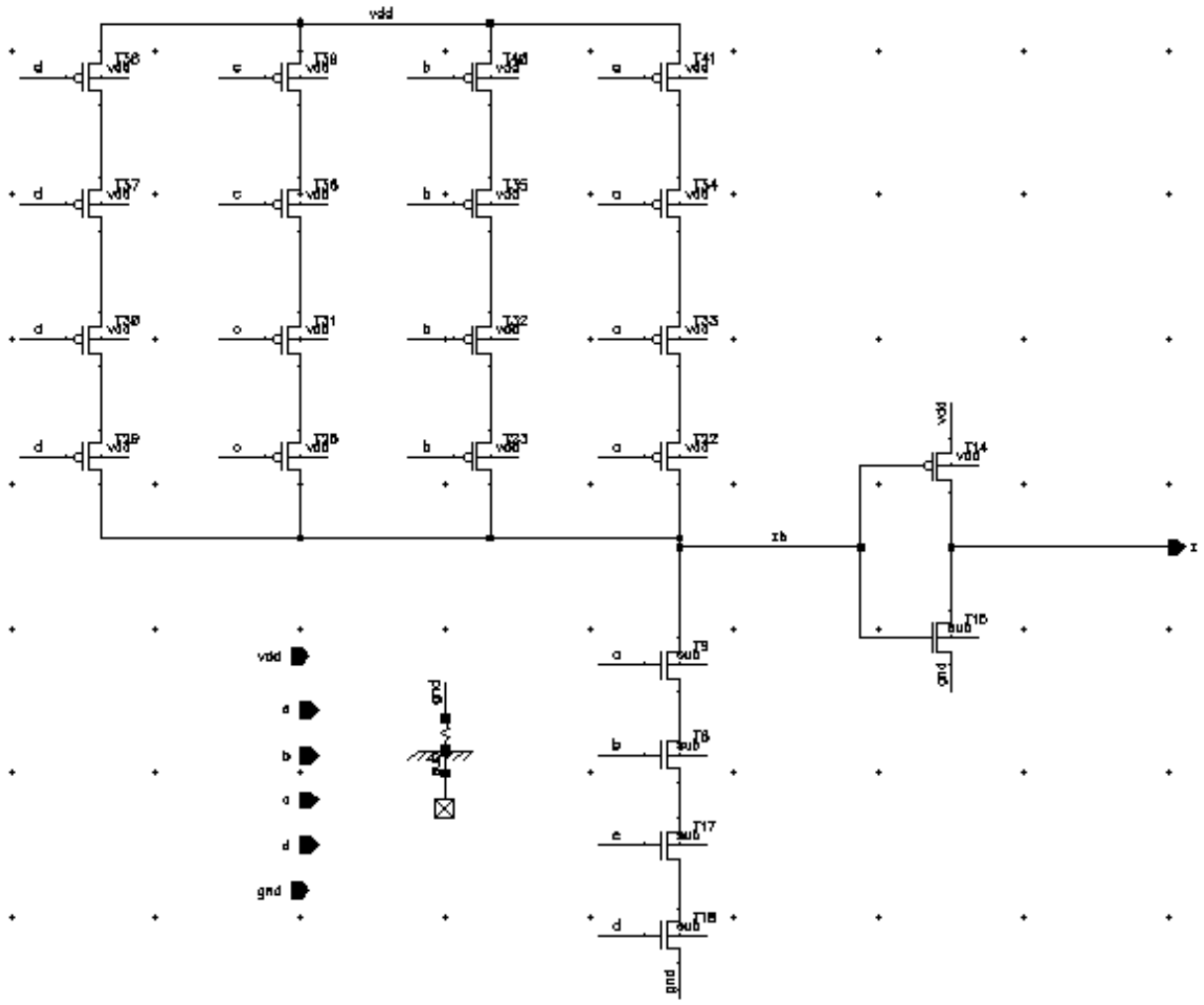

Figure 35: Four-input static AND gate schematic on the bulk-silicon process.

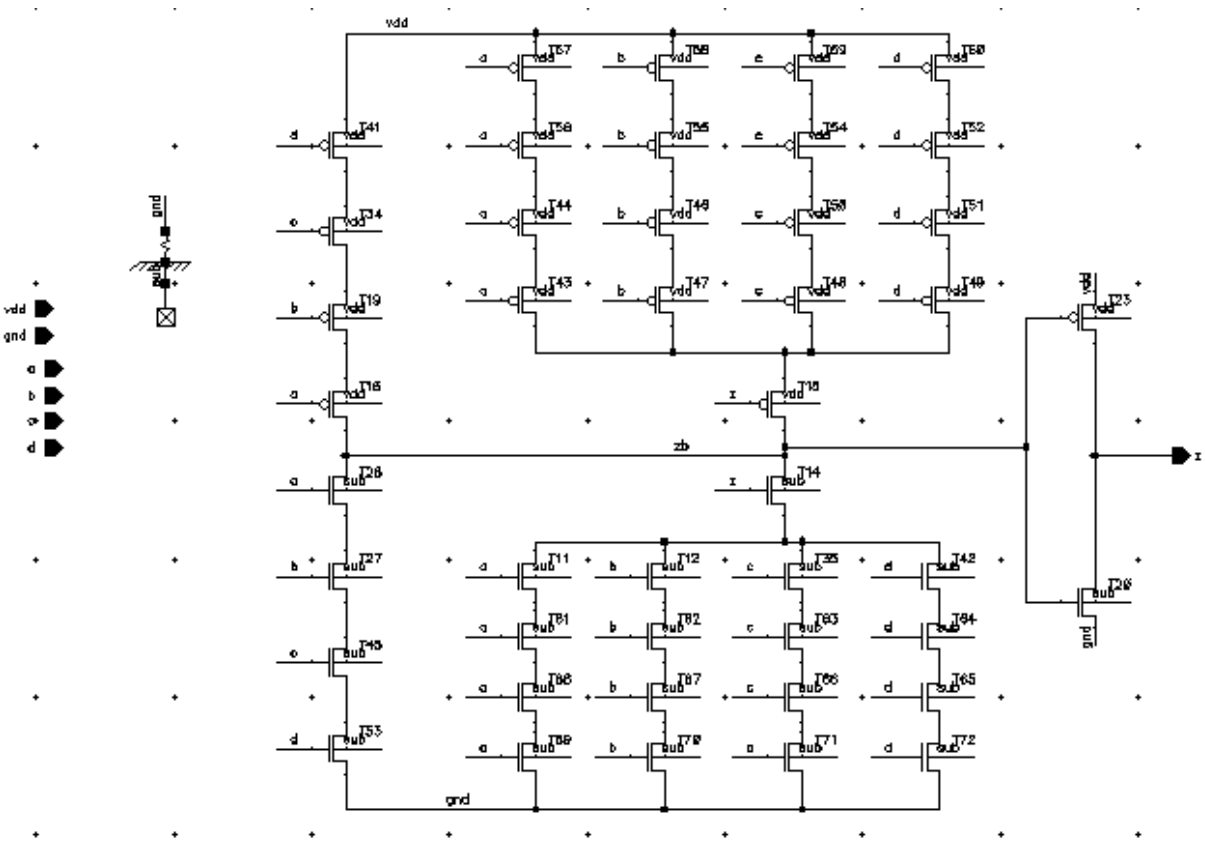

Figure 36: Four-input static TH44 gate schematics on the bulk-silicon process.

PMOS and NMOS channel dimensions. Figure 34 and Figure 35 shows two synchronous static gates for the bulk-silicon process. As illustrated, the AND gate uses forced transistor stacking in the pull-up network while the OR does the same for its pull-down network in order to reduce conductance through the parallel DC paths. Each transistor involved in a forced transistor stack are minimum sized (e.g., bulk-silicon PMOS transistors are 480 nm by 240 nm; NMOS are 160 nm by 120 nm). Transistors in series are widened; for example, each NMOS transistor in the pull-down network in the AND gate are quadrupled in width. For the FD-SOI process, the same design methodology is applied.

Figure 36 and Figure 37 show schematics of the static TH44 and TH34 threshold gates on the bulk-silicon process. Unlike the synchronous gates, threshold gates entail a more complex

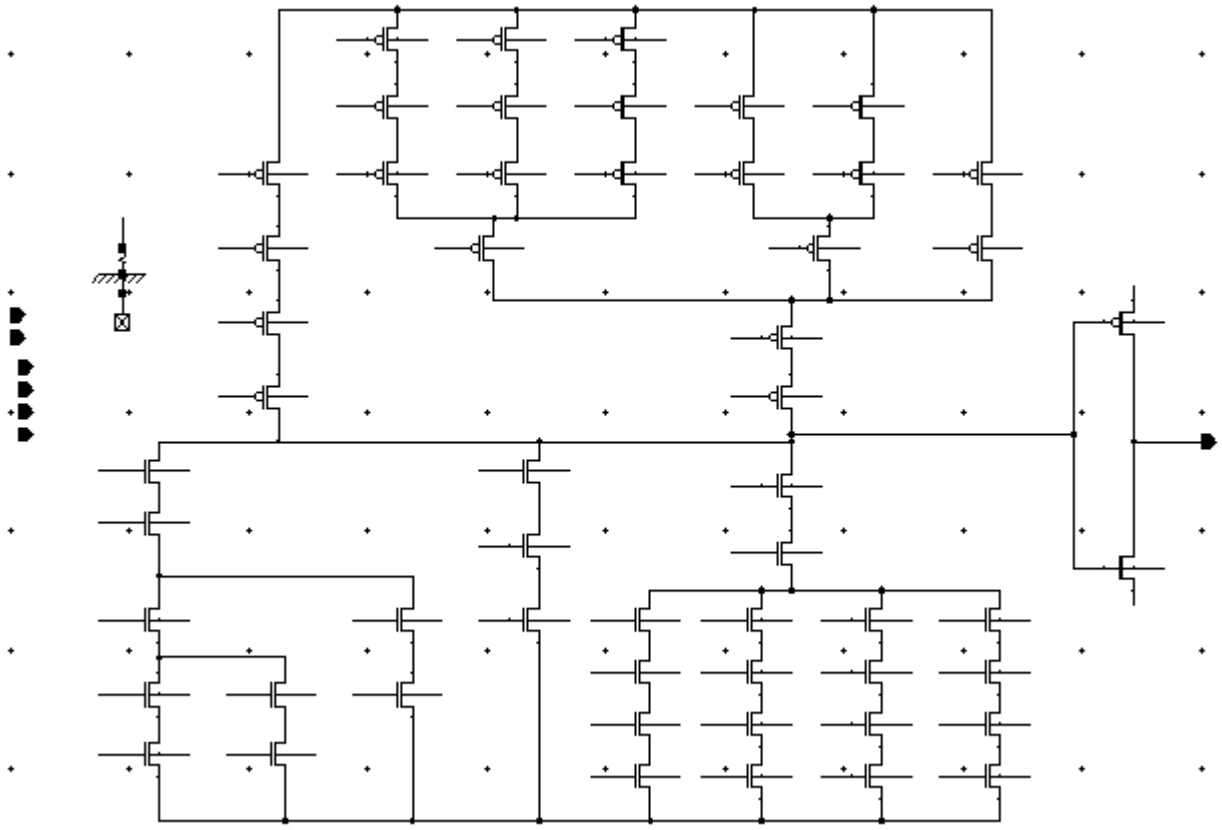

Figure 37: Four-input static TH34 gate schematic on the bulk-silicon process.

process in balancing the pull-up and pull-down networks. To recall the general threshold gate structure from Figure 9, the *set* and *hold* networks operate for different purposes. The *set* networks change the state of the threshold gate while the *hold* networks provide at least one DC path to power or ground. In effect, the *set* networks are inactive when the threshold gate is not changing state and the input vector points to an intermediate state (i.e., input combinations that do not cause the gate output to transition) while the *hold* networks activate in a similar manner as a combinational gate, where the current state provides an additional input to each *hold* network. Because the *set* networks are by definition transistor paths that cause the gate output to transition, a method in balancing the rise and fall transition times is balancing the resistances of the pull-up and pull-down *set* networks. However at ultra-low voltages, the topology of each network strongly affects active and inactive leakage current more than transistor channel dimensions. Instead, this research optimizes the *set* and *hold* networks combined in order to balance each *set* network's drive strength against the opposite *set* and *hold* networks. The disadvantage to this approach is increased area overhead as opposed to leaving the *hold* networks unoptimized.

The Schmitt-trigger gate structures from Figure 12 and Figure 13 entail positive output feedback optimization. Functionality and SNM depends on the relative drive strengths of the feedback transistors versus the respective pull-up and pull-down networks. Constructing a gate using weak feedback transistors ensure that the pull-up and pull-down networks can overpower the positive output feedback but at the cost of reduced SNM. Weak feedback transistors are increased in size to allow operability at or below target supply voltage. Using strong feedback transistors (i.e., NMOS to drive ground, PMOS to drive power) and given target supply voltage range, the feedback

transistors are optimized in size through experimentation in order to ensure that the output does not lock up. The target supply voltage range specified by the application is inversely related to the strength of the feedback transistors because as the supply voltage increases, the leakage current through the feedback transistors increase as well.

Figure 38 and Figure 39 shows schematics of the Schmitt-trigger OR4 and AND4 gates, built using weak feedback transistors. Transistor sizing in both the original and duplicate networks are copied from the static CMOS counterparts. The feedback transistors are minimum sized, and the inverters at the outputs of the AND and OR gates employ the Schmitt-trigger architecture. Concerning the output inverters, empirical simulations show an indifference in SNM between Schmitt-trigger and static CMOS inverters, so static CMOS inverters may be utilized in the synchronous gates to reduce area. However, simulations show that Schmitt-trigger inverters improve SNM when employing strong feedback transistors.

Two Schmitt-trigger threshold gates are shown in Figure 40 and Figure 40. Observed from the static CMOS threshold gates, hysteresis in each threshold gate creates a side effect where the state of the gate degrades over time if the supply voltage is too low. More specifically, the threshold gate loses its current state over time at low supply voltages when the gate input vector points to an intermediate (hold) state. Regarding the Schmitt-trigger threshold gates, gate simulations reveal that using strong feedback transistors prevent each threshold gate from losing current state; utilizing weak feedback transistors improves SNM but does not mitigate state loss.

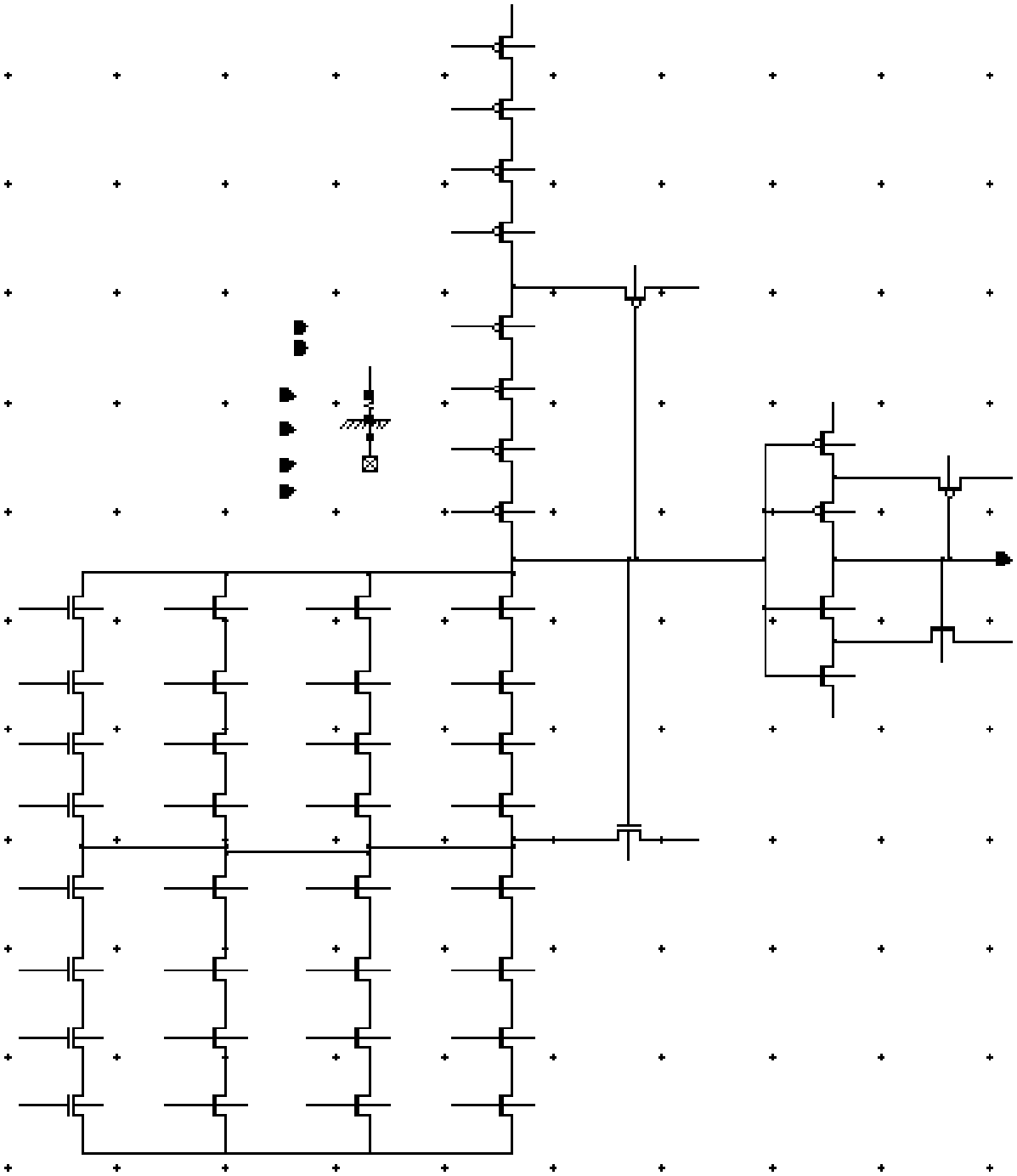

Figure 38: Four-input Schmitt-trigger OR gate schematic on the bulk-silicon process.

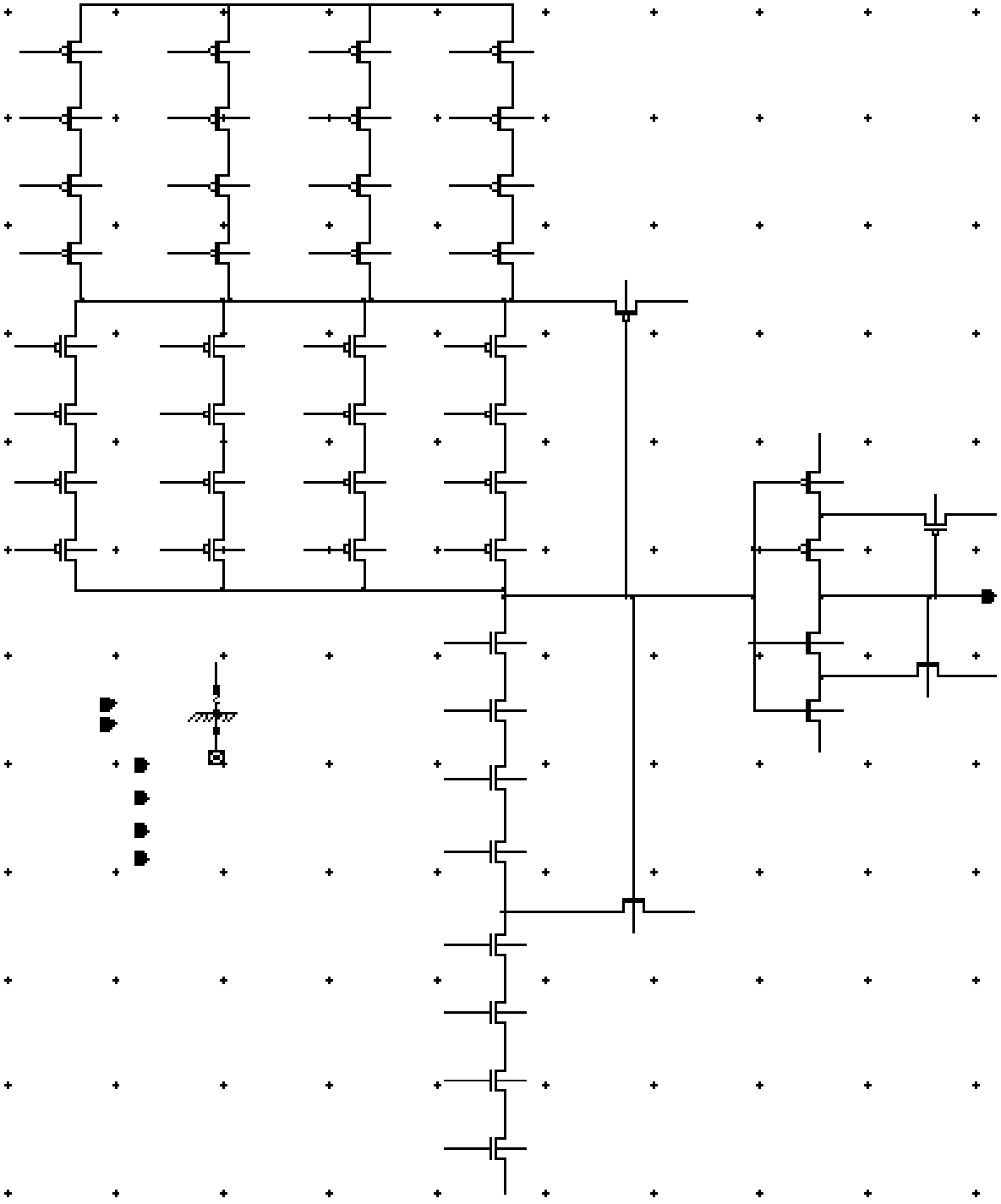

Figure 39: Four-input Schmitt-trigger AND gate schematic on the bulk-silicon process.

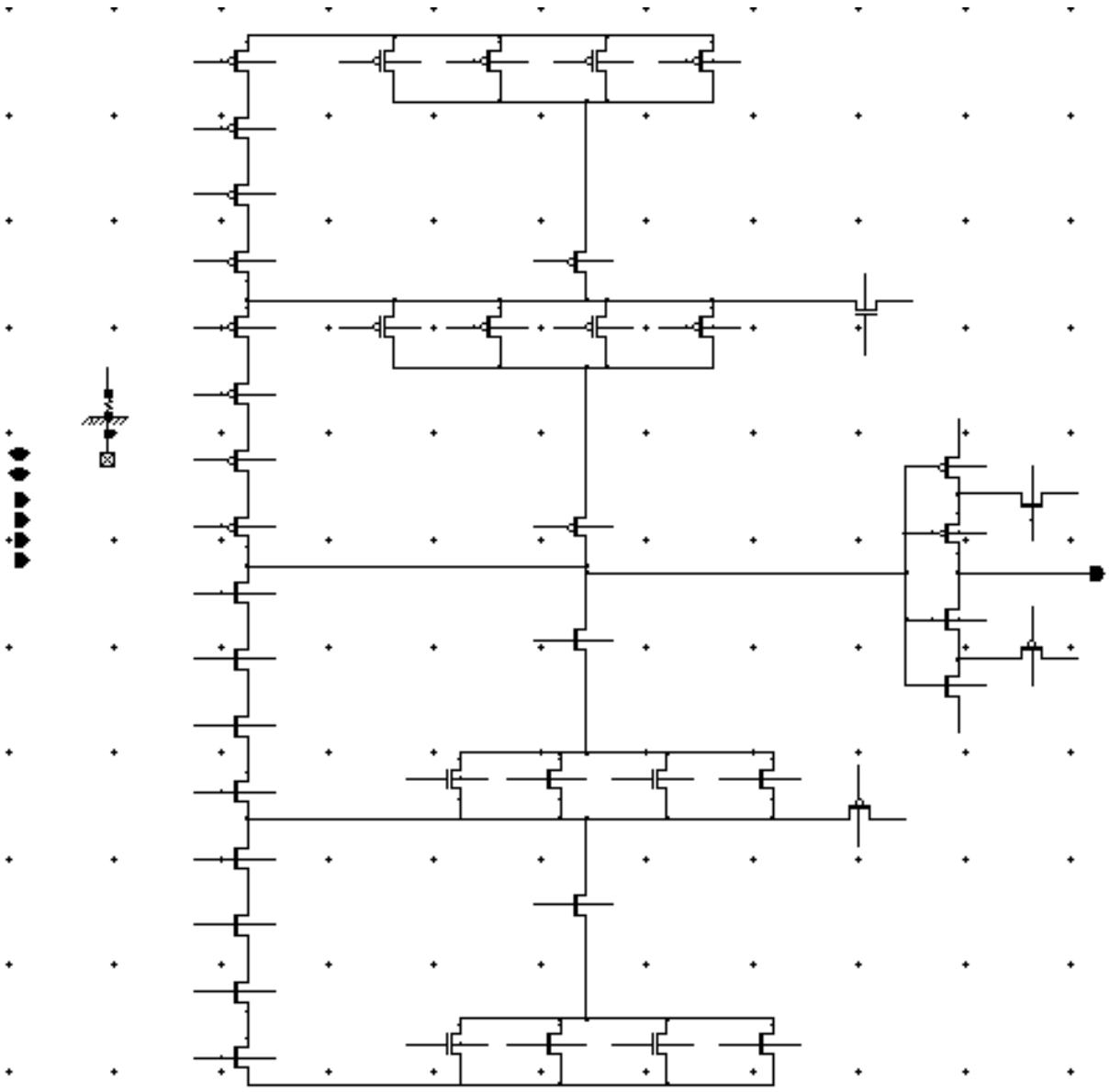

Figure 40: Four-input Schmitt-trigger TH44 gate schematic on the bulk-silicon process.

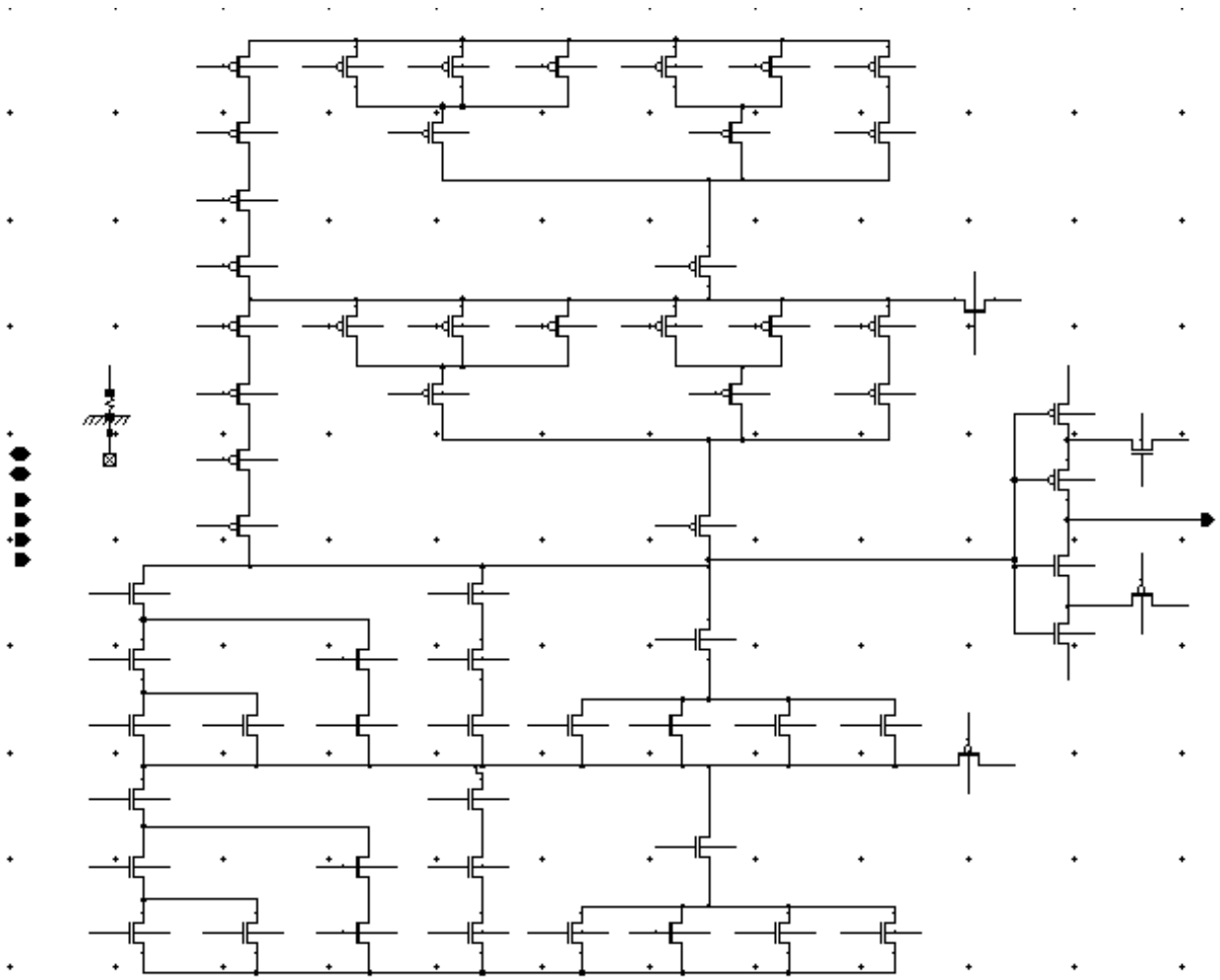

Figure 41: Four-input Schmitt-trigger TH34 gate schematic on the bulk-silicon process.

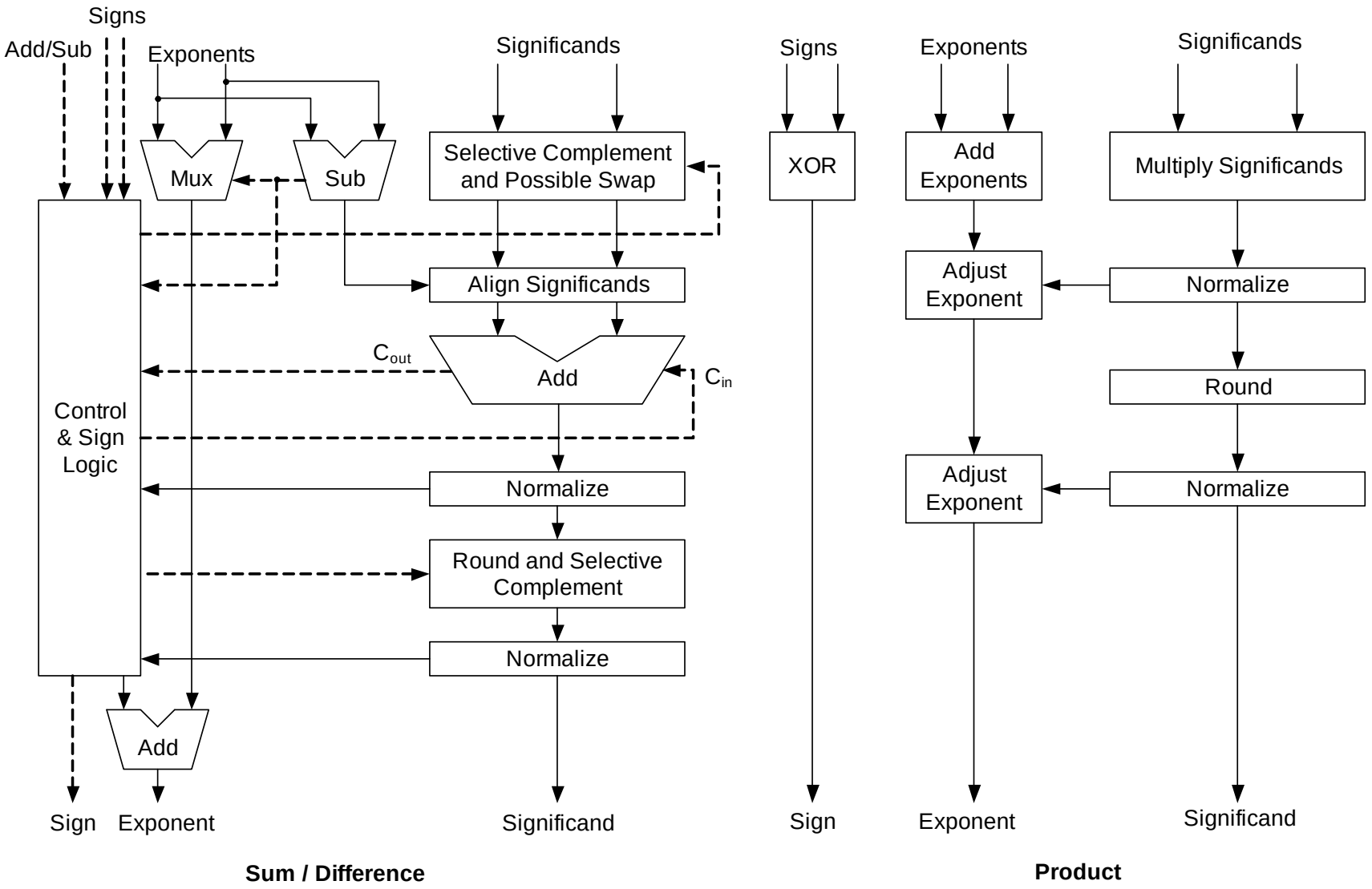

Figure 42: IEEE single-precision (32-bit) floating-point coprocessor architecture.

Figure 42 shows the architecture of the IEEE 32-bit floating-point coprocessor. It supports addition, subtraction, and multiplication on two IEEE single-precision input values. Both the synchronous and NCL coprocessors contain a four-stage pipeline. The second test circuit is the four-bit ripple-carry adder, which contains a two-stage pipeline for both the synchronous and NCL implementations. All test circuits are designed and built at the gate level, allowing for consistent implementation at the circuit level as each gate library is analyzed.

2.3 SIMULATION AND ANALYSIS SETUP

Minimum supply voltage, active energy and leakage power consumption, performance, and yield are measured in SPICE-level simulations. Finding the minimum supply voltage is performed for the synchronous gate libraries using gate-pair SNM analysis. Due to the hysteresis effect in each threshold gate, inter-cell SNM analysis does not yield a minimum supply voltage that is representative of the NCL circuits. Instead of SNM analysis, the minimum supply voltage for NCL is determined by sweeping the supply until the circuit produces correct results. Active energy and leakage power consumption are measured only for IBM's bulk-silicon process. MIT Lincoln Laboratory's FD-SOI models are not fully compatible with fast-SPICE simulators such as Cadence UltraSim, which are simulators optimized for digital circuits. In terms of performance, the maximum clock frequency is found for the synchronous circuits while average throughput is measured for the NCL circuits. Yield is simulated with Monte Carlo analysis for IBM's bulk-silicon process; the FD-SOI models lack statistical parameters required to model process variation.

The FD-SOI transistors come with an additional signal representing the back wafer. For comparable analysis with the bulk-silicon process, the back wafer is tied to ground, eliminating

its effect on the FD-SOI threshold voltage. The 150 nm FD-SOI transistor models are tested using both Cadence Spectre and UltraSim through the test circuits. In UltraSim, convergence errors occur from the FD-SOI models when setting UltraSim to Mixed-Signal mode, which references simplified transistor models containing analog look-up tables and performs digital circuit partitioning. UltraSim supports a less accurate mode that references simplified models suitable for functional simulations but does not support accurate DC current draw measurements required to estimate active energy and leakage power. Cadence Spectre and UltraSim's most accurate simulation mode, SPICE, are fully compatible with the 150 nm FD-SOI models. In short, only small FD-SOI circuits may be measured for power by referencing the BSIMSOI equations directly.

Figure 43 and Figure 44 show a schematics that perform SNM analysis on a pair of inverting and non-inverting gates respectively. It implements the simplified model shown in Figure 3, which in turn represents the gate pair driving one another. The testbench in Figure 44 applies to a pair of positive unate, stateless gates (i.e., synchronous gates such as AND and OR). Using Cadence's built-in calculator and Spectre simulator, SNM is measured by running DC analysis to sweep the voltage source, V_u, from $-V_{dd}$ to $+V_{dd}$ where V_{dd} is the target supply voltage. The nodes v_1 and v_2 represent output voltages from the respective gates after translation to the 45-degree coordinate system. Two SNM values are found by taking the difference, $v_1 - v_2$, and finding the maximum value for negative u and the negated minimum value for positive u. The minimum of the two SNM values is the SNM value of the gate pair for a given supply voltage, V_{dd}. Extending on SNM analysis, V_{dd} is swept to graph the curve of SNM versus the supply voltage, which shows the point at which SNM transitions from negative to positive values as V_{dd} increases. Where SNM crosses

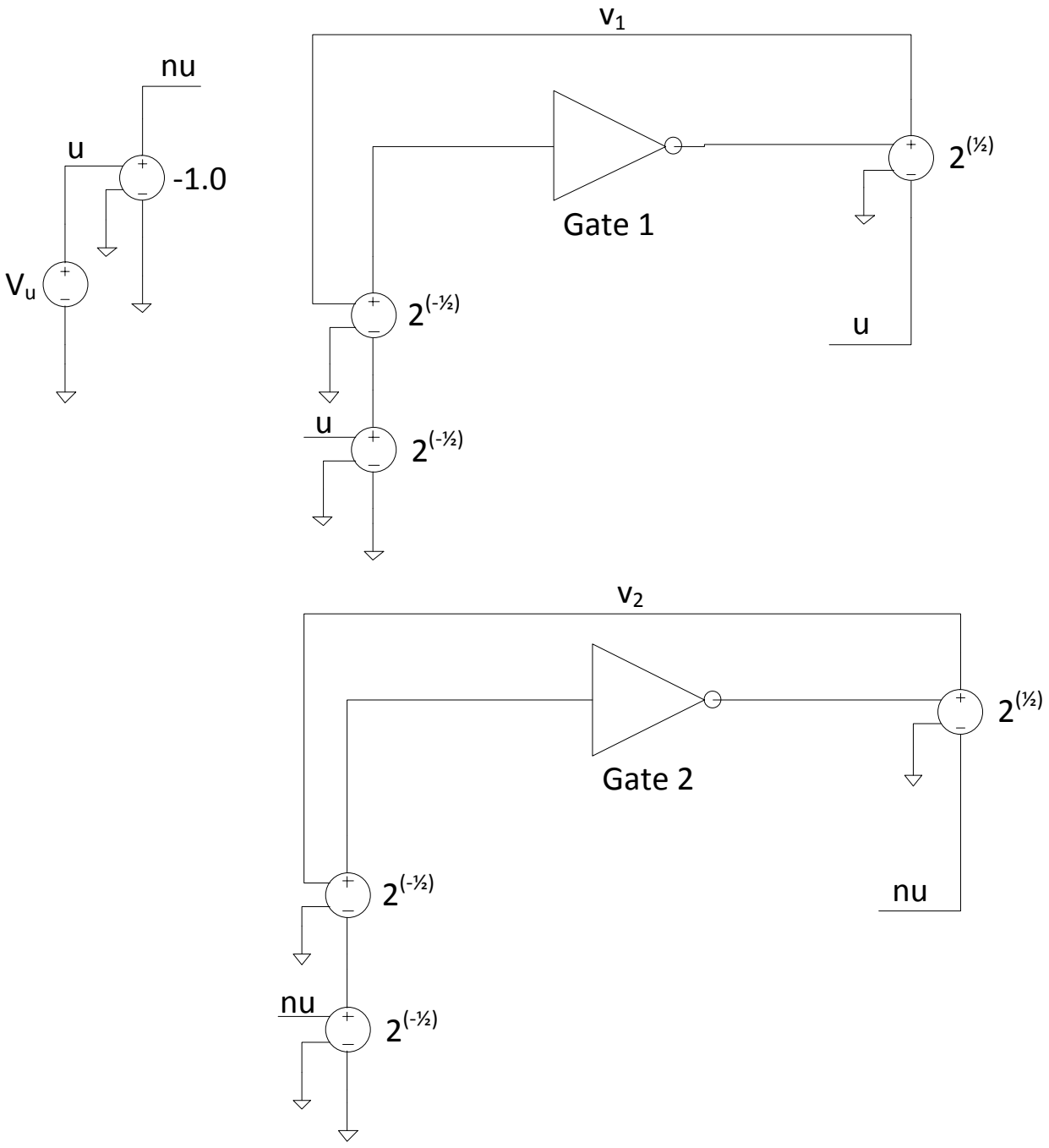

Figure 43: Testbench schematic used for static noise margin analysis of two negative unate gates.

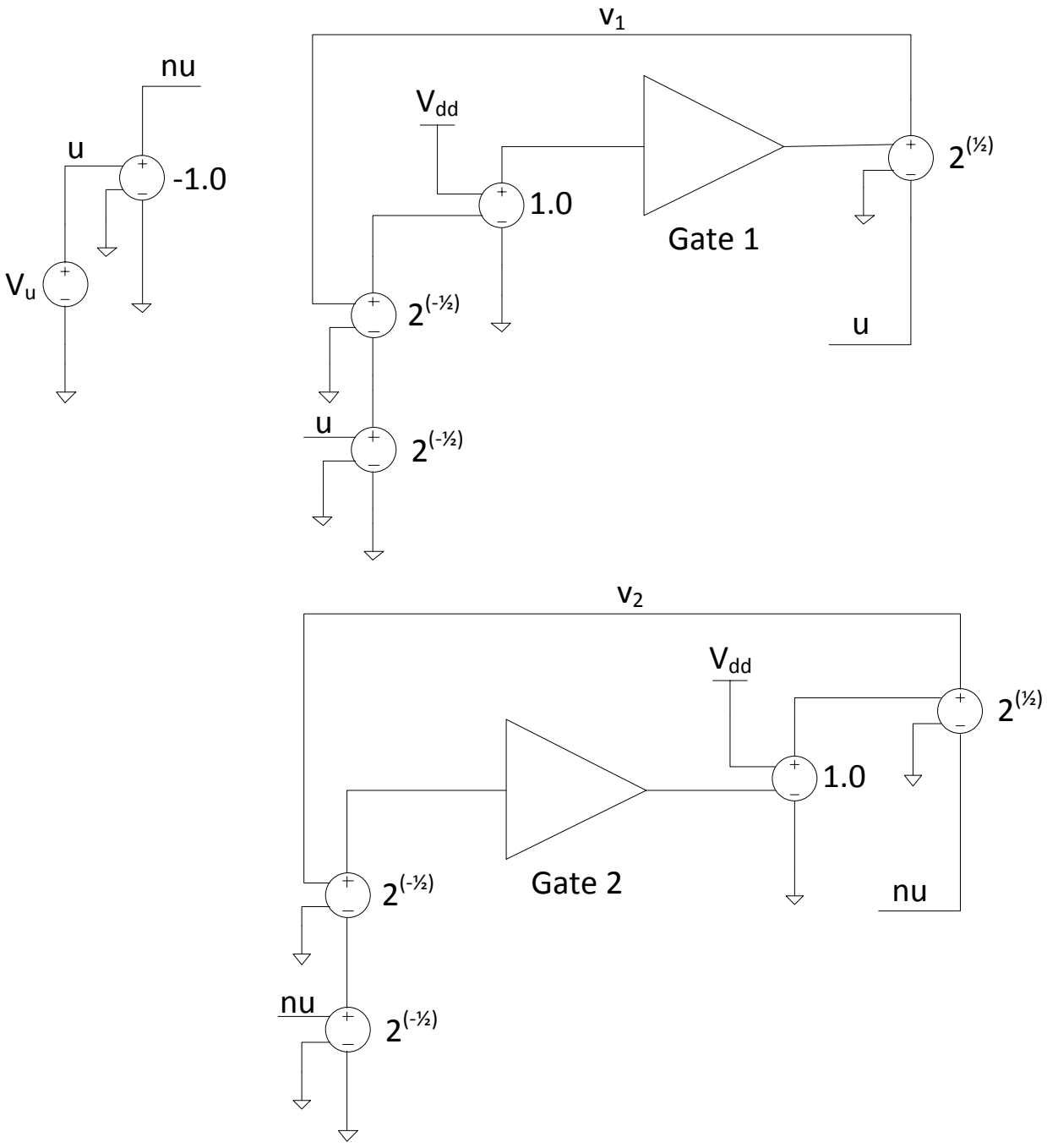

Figure 44: Testbench schematic used for static noise margin analysis of two positive unate gates.

Table 1: Input and expected output vectors applied to the floating-point coprocessor.

Vector	Operation	X (input)	Y (input)	Z (output)
1	Addition	0x447CC000	0x448AE000	0x4504A000
2	Addition	0xC070B7FE	0xC12E057D	0xC16A337D
3	Addition	0x490D7523	0xD011CC0C	0xD011C9D6
4	Subtraction	0x447CC000	0x448AE000	0xC2C80000
5	Subtraction	0xC070B7FE	0xC12E057D	0x40E3AEFA
6	Subtraction	0x490D7523	0xD011CC0C	0x5011CE42
7	Multiplication	0x447CC000	0x448AE000	0x49891CA8
8	Multiplication	0xC070B7FE	0xC12E057D	0x4223A238
9	Multiplication	0x490D7523	0xD011CC0C	0xD9A12032

zero defines the minimum supply voltage required for the gate pair.

The floating-point coprocessor undergoes transient analysis to check for functionality and measure active energy, leakage power, and performance. This test circuit is given input vectors shown in Table 1 and expected to produce the respective outputs. Active energy and leakage power are calculated from the bulk-silicon models by measuring the supply current draw during operation. Equation 4 and Equation 5 calculate active energy and leakage power respectively. Active energy is measured during circuit operation over the time period $t_2 - t_1$ while leakage power is measured after circuit operation over the time period $t_3 - t_2$. The minimum clock period required for the synchronous counterparts at a given supply voltage is approximated through experimentation. NCL throughput is recorded from and to the beginning of a data cycle, covering one data cycle and one NULL cycle. NCL throughput is then averaged among the given vectors.

$$ActiveEnergy = V_{dd} \int_{t_1}^{t_2} i(t)dt \tag{4}$$

$$LeakagePower = \frac{V_{dd} \int_{t_2}^{t_3} i(t)dt}{t_3 - t_2} \tag{5}$$

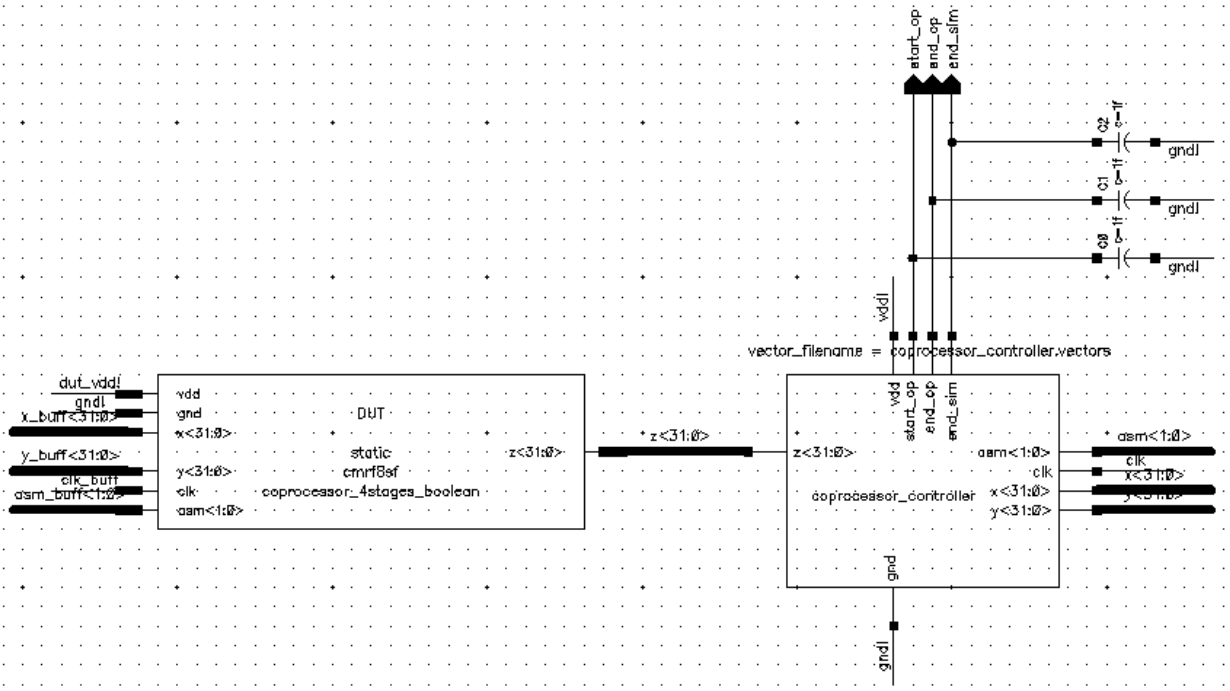

Figure 45: Synchronous coprocessor testbench.

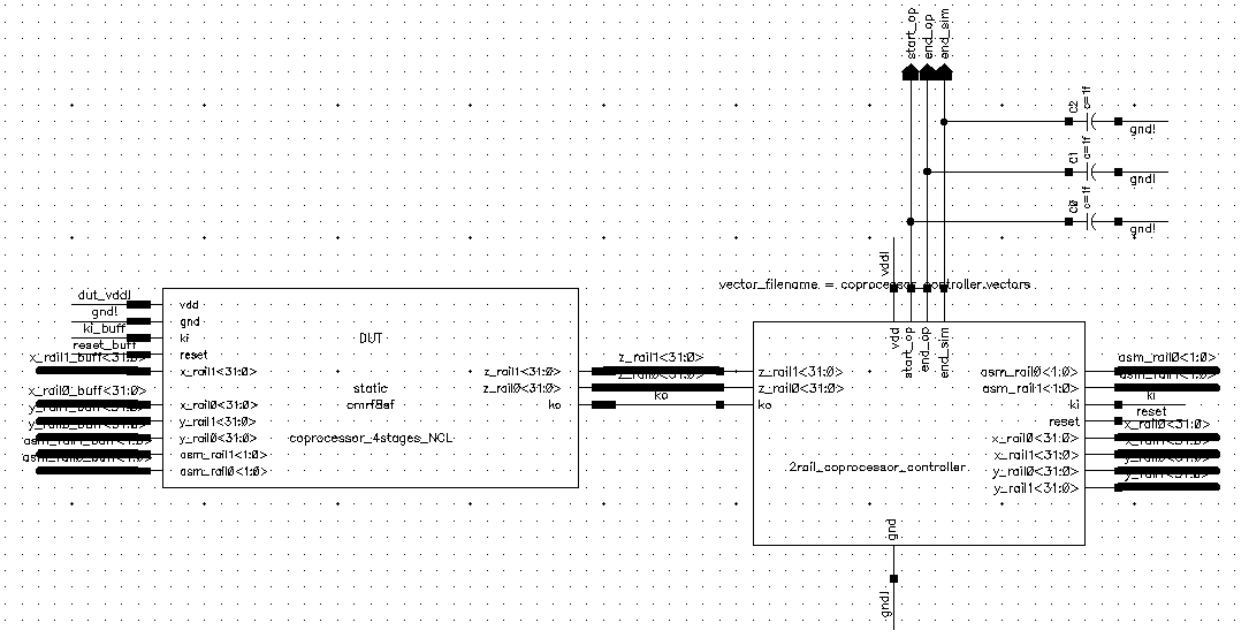

Figure 46: NCL coprocessor testbench.

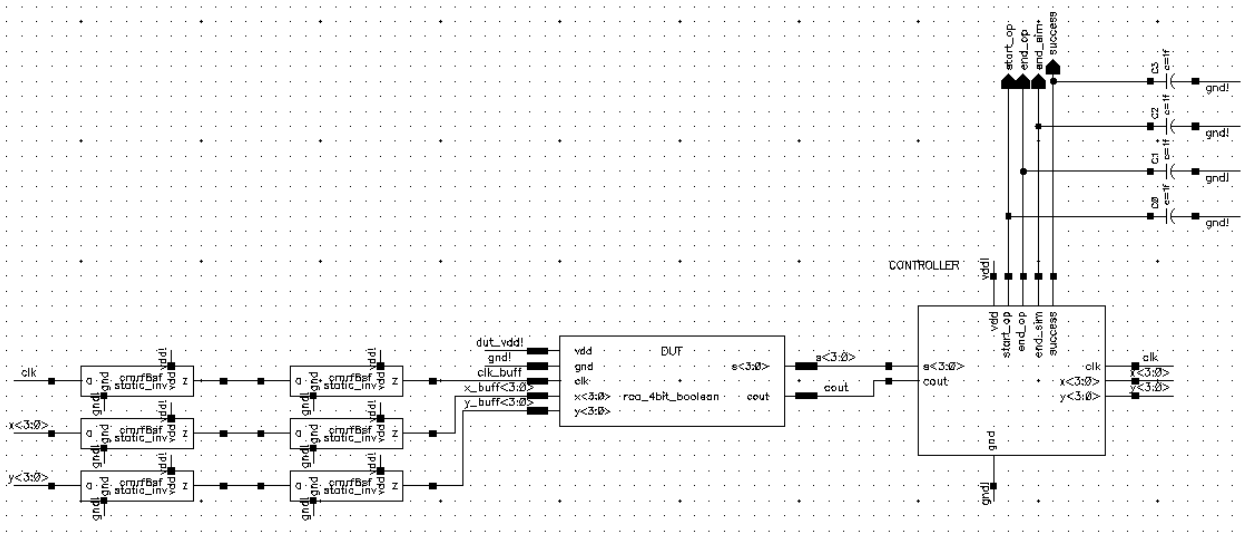

Figure 47: Synchronous adder testbench.

Figure 45 is a testbench used in Monte Carlo analysis of a synchronous coprocessor on the bulk-silicon process. The NCL testbench shown in Figure 46 is similar to the synchronous testbench except for the dual-rail signals in between the NCL Verilog-A controller and coprocessor. Another difference is that the NCL Verilog-A controller is a finite state machine that steps through each part of NCL's four-phase handshaking protocol.

Yield analysis is performed on the synchronous static ripple-carry adder on the bulk-silicon process. Transient analysis provides the basis for Monte Carlo analysis in order to sample active energy, leakage power, and effective threshold voltage of a randomly selected NMOS transistor. The ripple-carry adder is given random input vectors, and its functionality is checked and recorded for each sample. Yield analysis is executed using the minimum supply voltage approximated from SNM analysis and experimentation. Figure 47 is a testbench used in Monte Carlo analysis of a synchronous adder on the bulk-silicon process.

2.4 RESULTS

Figure 48–55 show SNM plots of each gate library over the supply voltage range from 0 V to 150 mV. Each plot points to the minimum required supply voltage for each gate library where SNM crosses zero. The bulk-silicon Schmitt-trigger NCL gate library shows the lowest supply requirement followed by its FD-SOI counterpart. The two static NCL gate libraries require a higher supply voltage than their Schmitt-trigger counterparts where the bulk-silicon static NCL library need the highest supply among all NCL libraries. The NCL libraries have consistently lower supply requirements than the synchronous libraries. Among the synchronous libraries, the FD-SOI Schmitt-trigger synchronous library has the lowest supply demand while the FD-SOI static synchronous library needs the highest supply voltage. Between the static and Schmitt-trigger libraries, the Schmitt-trigger libraries have consistently lower supply requirements than their static counterparts.

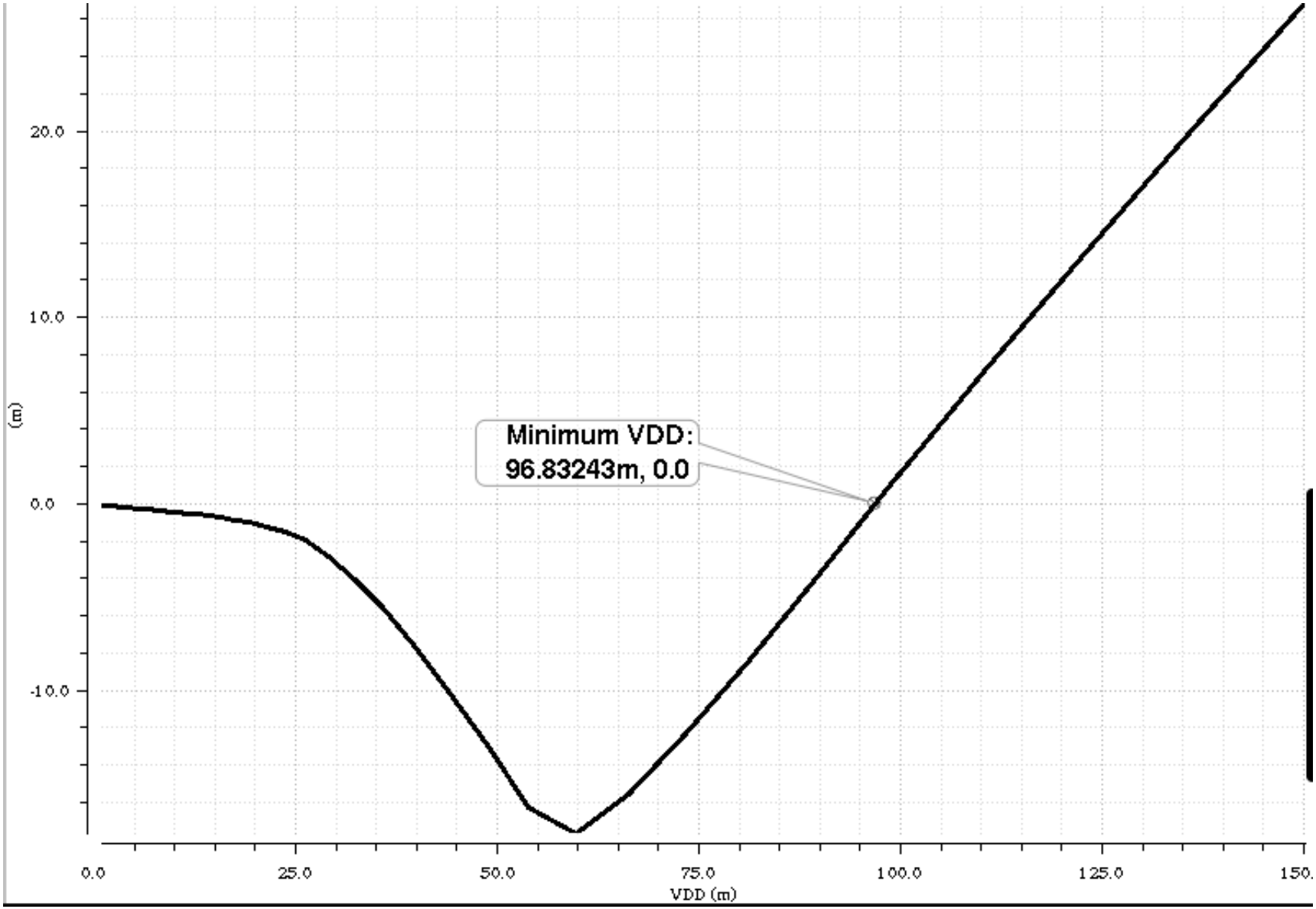

Figure 48: SNM versus supply voltage on bulk-silicon static synchronous gate libraries.

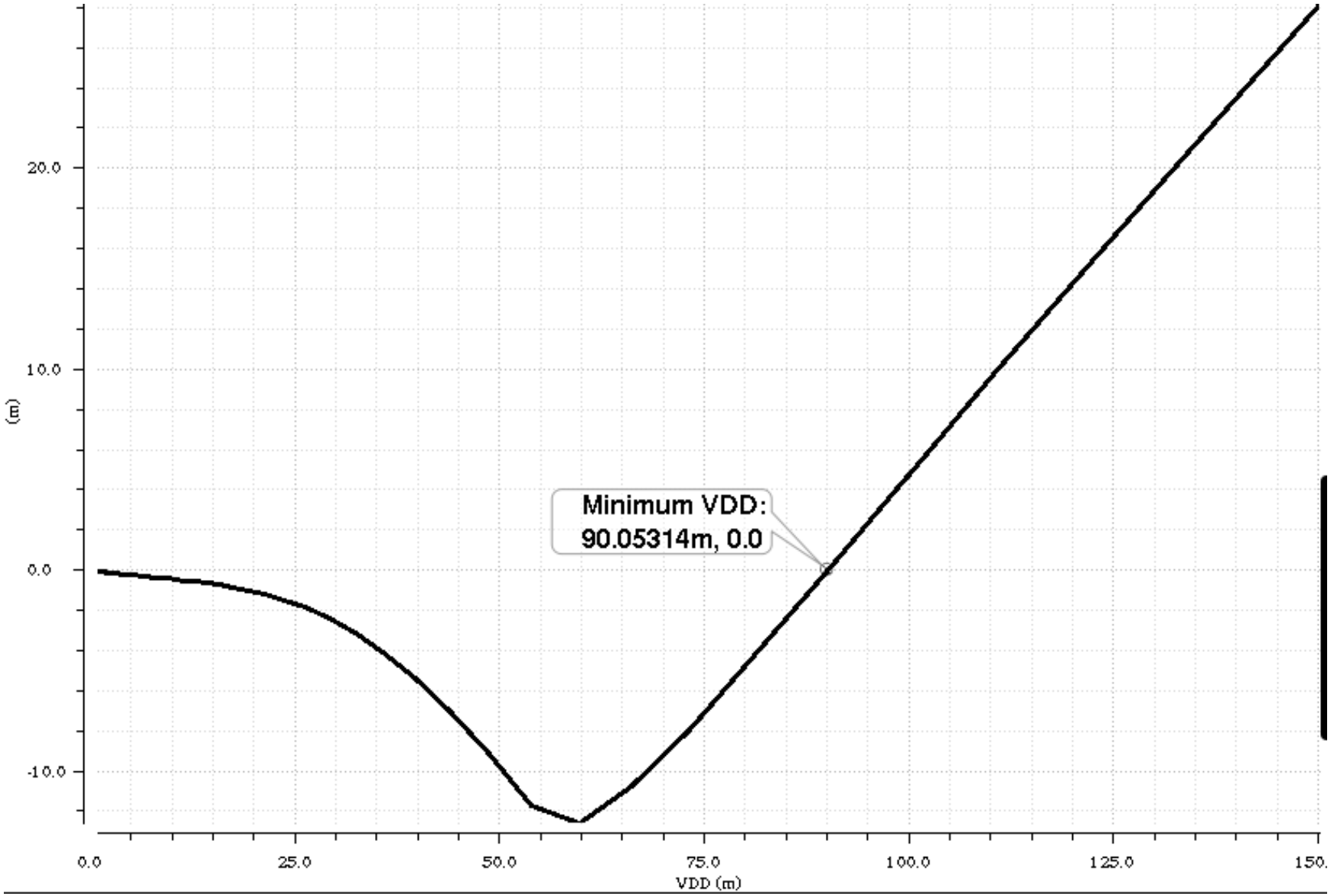

Figure 49: SNM versus supply voltage on bulk-silicon static NCL gate libraries.

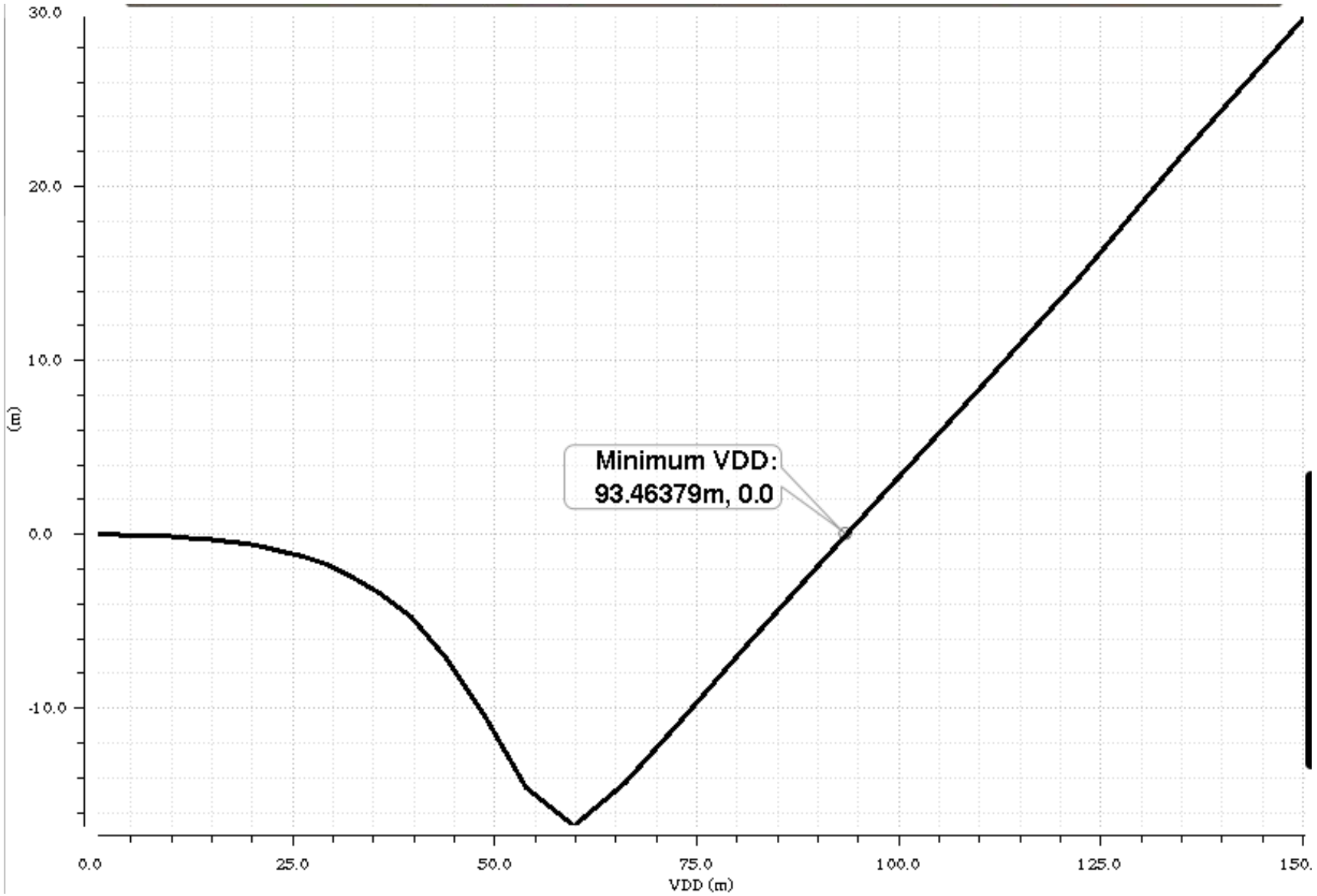

Figure 50: SNM versus supply voltage on bulk-silicon Schmitt-trigger synchronous gate libraries.

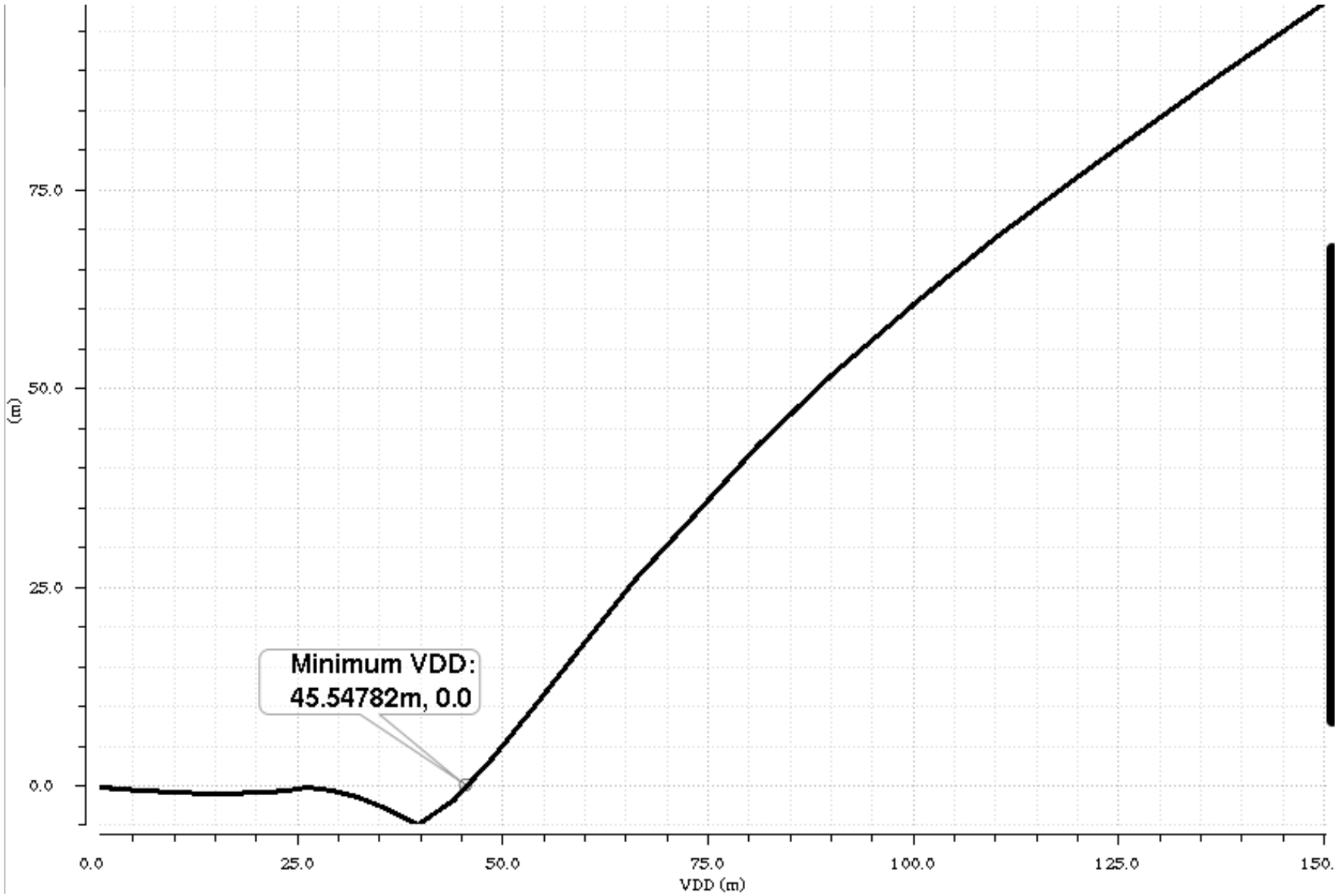

Figure 51: SNM versus supply voltage on bulk-silicon Schmitt-trigger NCL gate libraries.

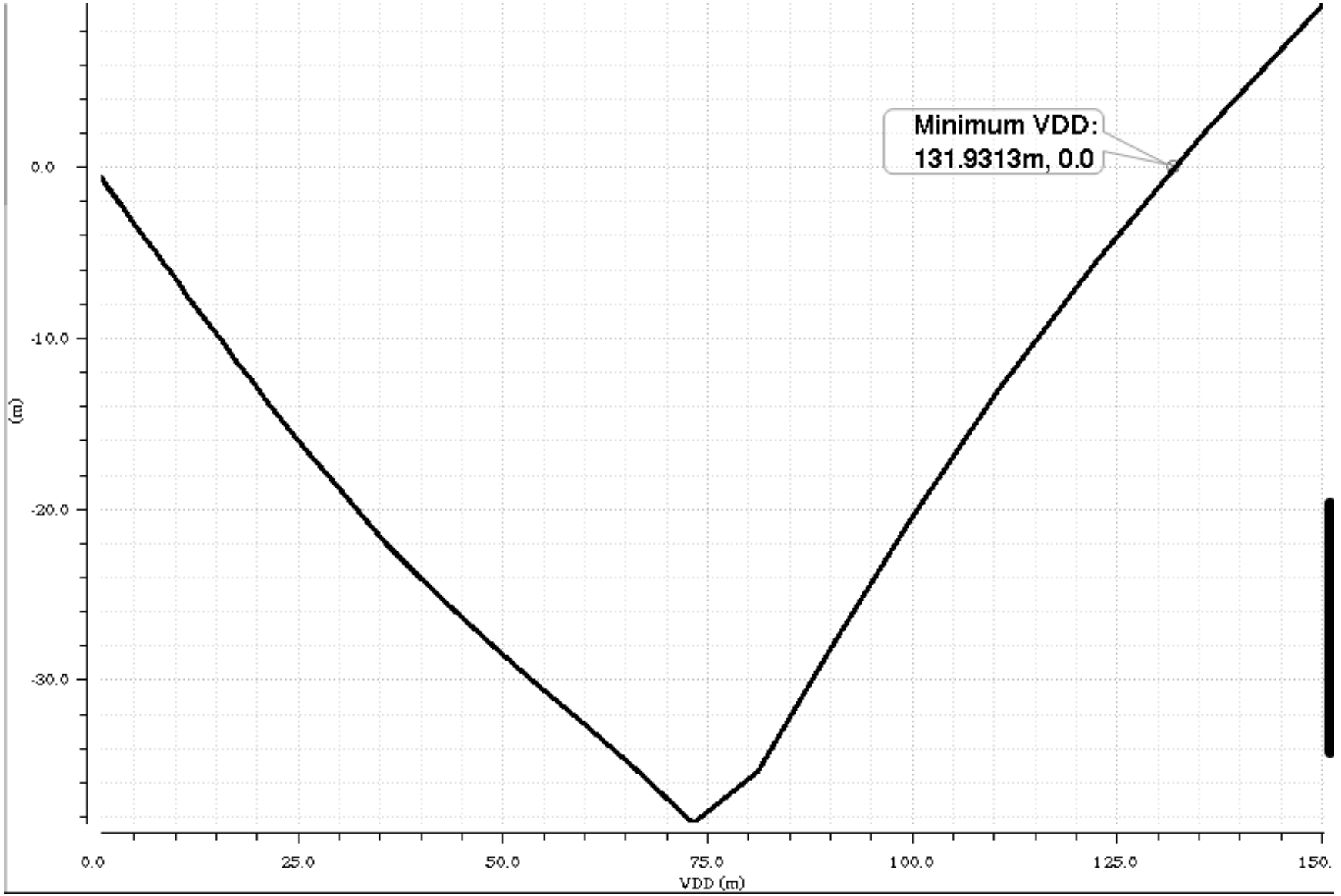

Figure 52: SNM versus supply voltage on FD-SOI static synchronous gate libraries.

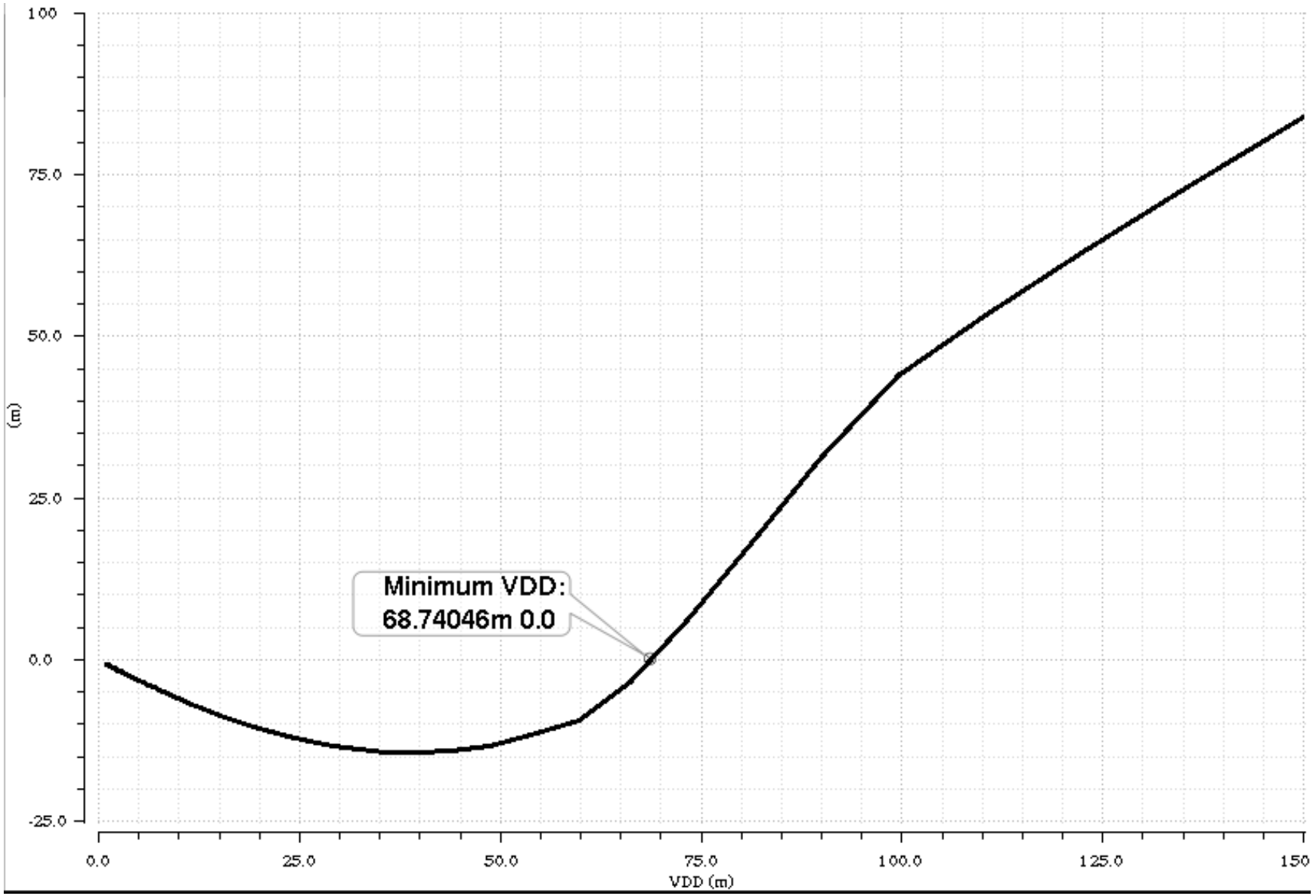

Figure 53: SNM versus supply voltage on FD-SOI static NCL gate libraries.

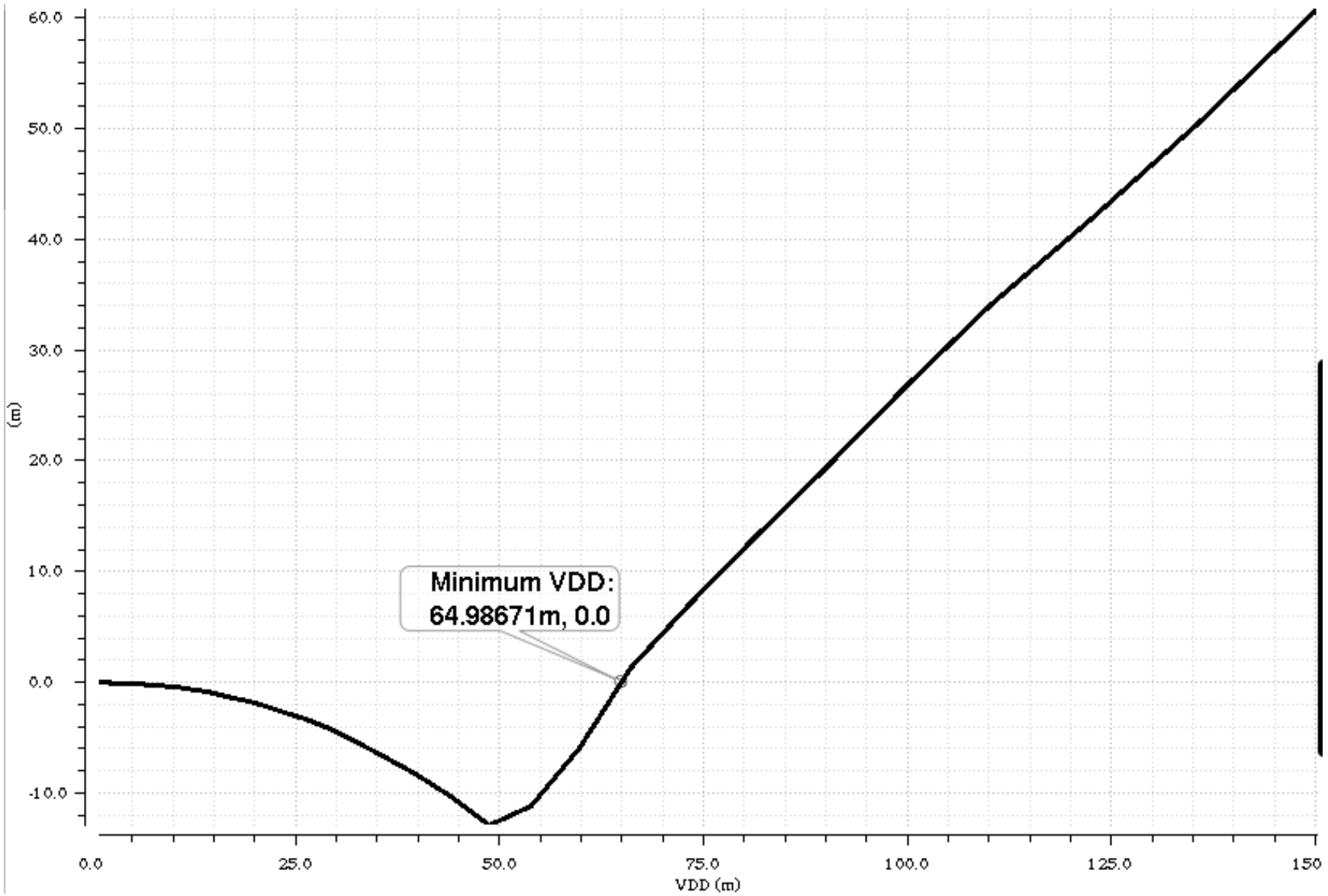

Figure 54: SNM versus supply voltage on FD-SOI Schmitt-trigger synchronous gate libraries.

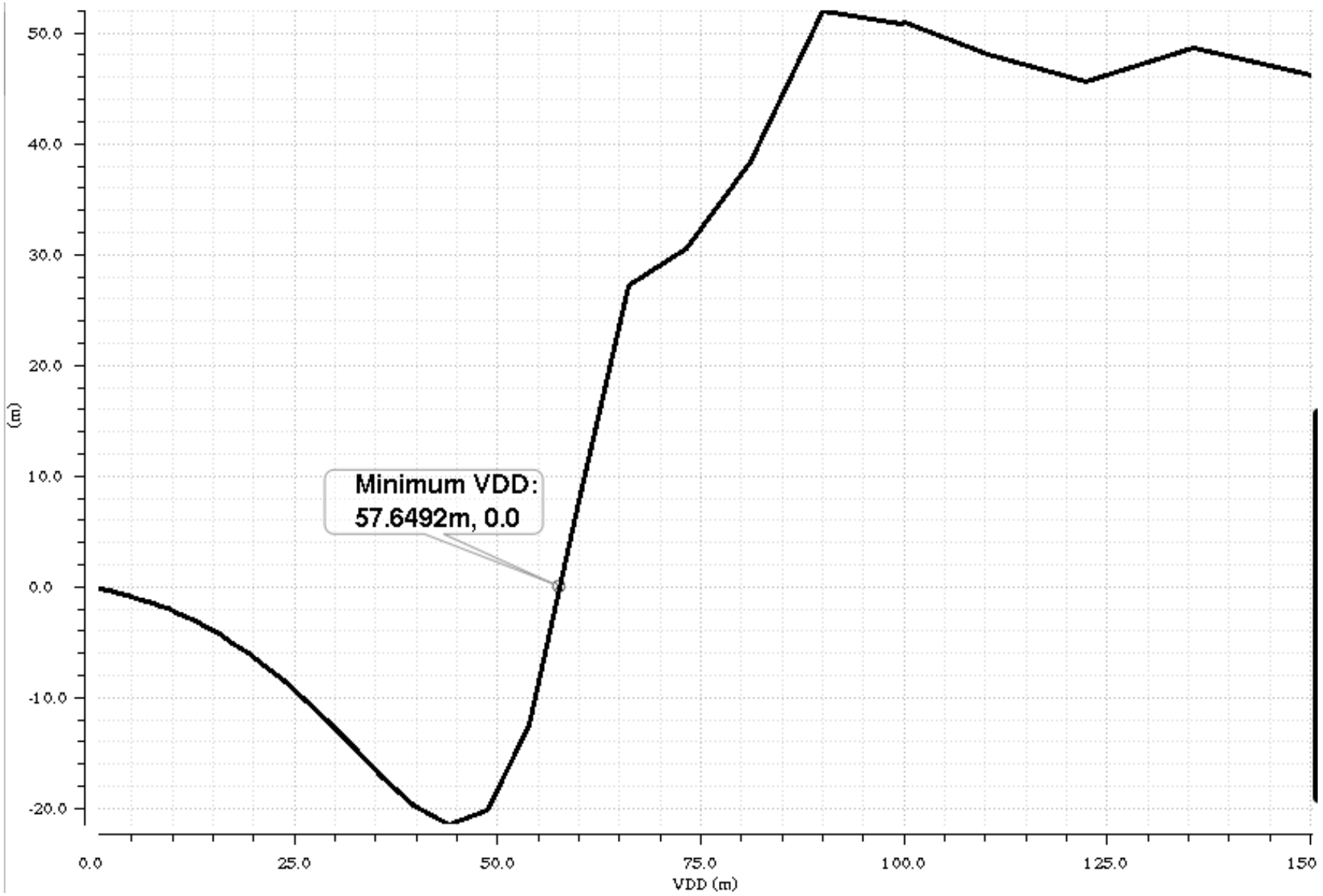

Figure 55: SNM versus supply voltage on FD-SOI Schmitt-trigger NCL gate libraries.

Table 2: Bulk-silicon static synchronous floating-point coprocessor measurements.

V_{dd} (mV)	Active Energy (pJ)	Leakage Power (nW)	Average Performance (μs)
500	48	697	1
300	59	304	10
200	246	176	100
150	341	114	200
125	417	99	300
115	Failure	Failure	Failure

Table 3: Bulk-silicon static NCL floating-point coprocessor measurements.

V_{dd} (mV)	Active Energy (pJ)	Leakage Power (nW)	Average Performance (μs)
500	119	1121	0.6
300	195	516	11.8
200	255	296	85.6
150	469	207	234.7
125	602	168	377.5
115	Failure	Failure	Failure

Table 2 and Table 3 compare the synchronous and NCL bulk-silicon static coprocessor designs. Both designs function properly at supply voltages as low as 125 mV. At all supply voltage points, the synchronous coprocessor consumes less active energy and leakage power. The synchronous coprocessor performs slightly faster than the NCL coprocessor, namely at 125 mV where it takes the NCL design over 25% more time to process each input vector.

Table 4 and Table 5 compare the two bulk-silicon Schmitt-trigger coprocessor designs.

Table 4: Bulk-silicon Schmitt-trigger synchronous floating-point coprocessor measurements.

V_{dd} (mV)	Active Energy (pJ)	Leakage Power (nW)	Average Performance (μs)
100	Failure	Failure	Failure
90	17070	257	5000
80	13137	227	10000
75	Failure	Failure	Failure

Table 5: Bulk-silicon Schmitt-trigger NCL floating-point coprocessor measurements.

V_{dd} (mV)	Active Energy (pJ)	Leakage Power (nW)	Average Performance (μs)
125	Failure	Failure	Failure
115	12680	293	4500
100	15160	252	6303
90	14060	223	6630
80	12980	196	7013
75	12440	182	7249
70	Failure	Failure	Failure

Table 6: FD-SOI static synchronous floating-point coprocessor measurements.

V_{dd} (mV)	Active Energy (pJ)	Leakage Power (nW)	Average Performance (μs)
500	162	398	1
150	557	33	1000
115	Failure	Failure	Failure

Both designs have limited dynamic ranges in supply voltage in that the Schmitt-trigger gate structure introduces a maximum supply voltage before failure. The synchronous design functions correctly from 80 mV to 90 mV while the NCL counterpart operates properly from 75 mV to 115 mV. Both Schmitt-trigger coprocessors operate at lower supply voltages than their static counterparts, and the NCL design supports a wider supply range that is slightly lower than that of the synchronous design. From 80 mV to 90 mV, active energy and leakage power are comparable such that the NCL Schmitt-trigger coprocessor consume slightly less power. Results in performance are mixed between the Schmitt-trigger NCL and synchronous designs.

Table 7: FD-SOI static NCL floating-point coprocessor measurements.

V_{dd} (mV)	Active Energy (pJ)	Leakage Power (nW)	Average Performance (μs)
500	183	358	0.7
150	114	22	347.8
115	Failure	Failure	Failure

Table 8: FD-SOI Schmitt-trigger synchronous floating-point coprocessor measurements.

V_{dd} (mV)	Active Energy (pJ)	Leakage Power (nW)	Average Performance (μs)
150	Failure	Failure	Failure
125	4655	70	5000
100	6515	50	10000
90	5239	41	10000
80	Failure	Failure	Failure

Table 9: FD-SOI Schmitt-trigger NCL floating-point coprocessor measurements.

V_{dd} (mV)	Active Energy (pJ)	Leakage Power (nW)	Average Performance (μs)
200	Failure	Failure	Failure
150	17950	151	12571
125	12130	150	8674
100	Failure	Failure	Failure

Table 6 and Table 7 show measurements taken from the two FD-SOI static coprocessor designs. As with the bulk-silicon counterparts, both designs have wide dynamic ranges in supply voltage, as low as 115 mV before failure. At 500 mV, active energy, leakage power, and average performance are similar. At 150 mV, the NCL design shows improvement over the synchronous design in power consumption and performance.

Table 8 and Table 9 show measurements taken from the two FD-SOI Schmitt-trigger coprocessor designs. Similar to the bulk-silicon counterparts, the FD-SOI Schmitt-trigger designs have limited dynamic ranges in supply voltage. The NCL design operates at a lower supply range than the synchronous version, overlapping at around 125 mV.

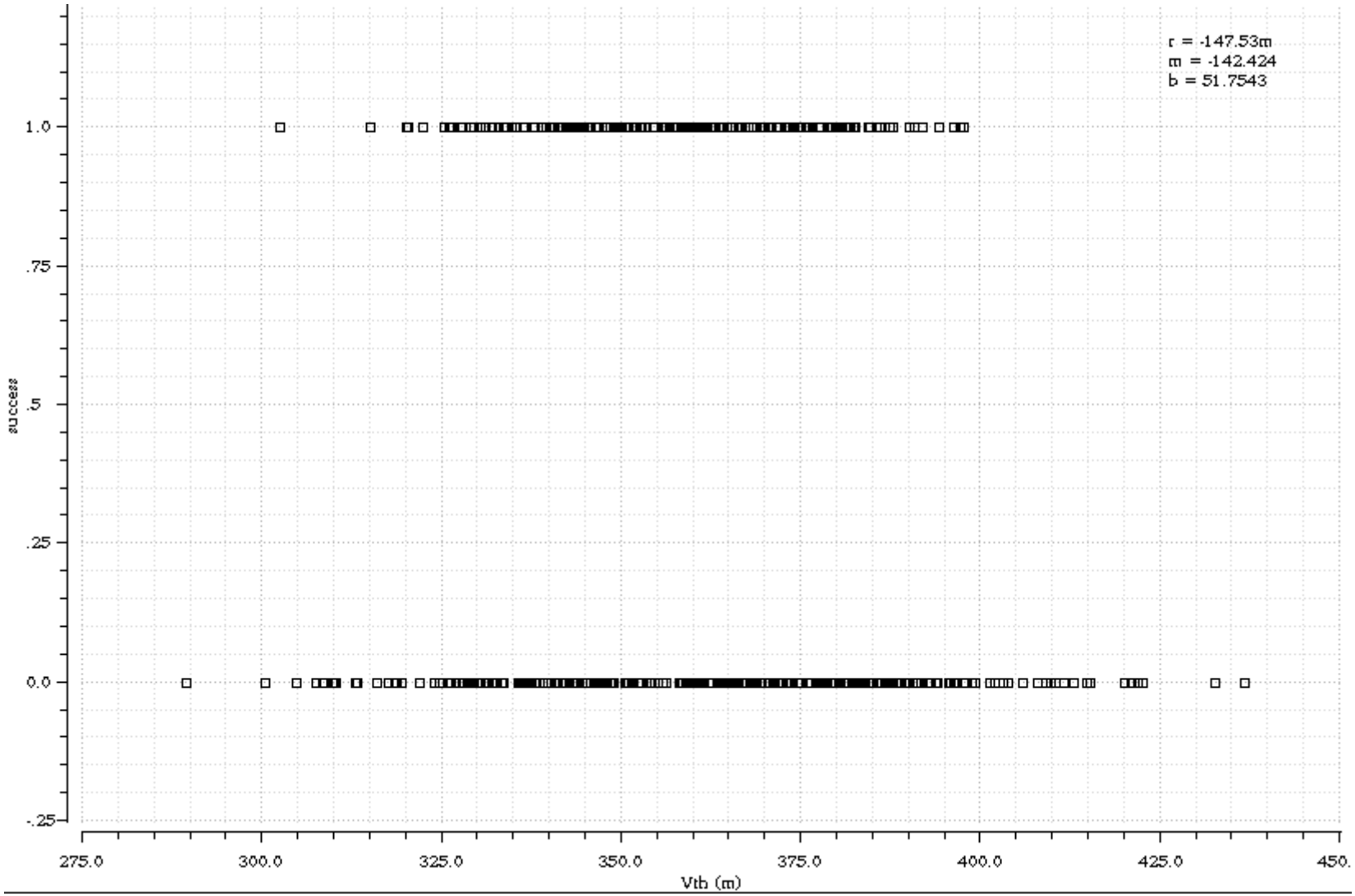

Figure 56: Success versus threshold scatter plot on bulk-silicon static synchronous ripple-carry adder at 90 mV over 500 samples.

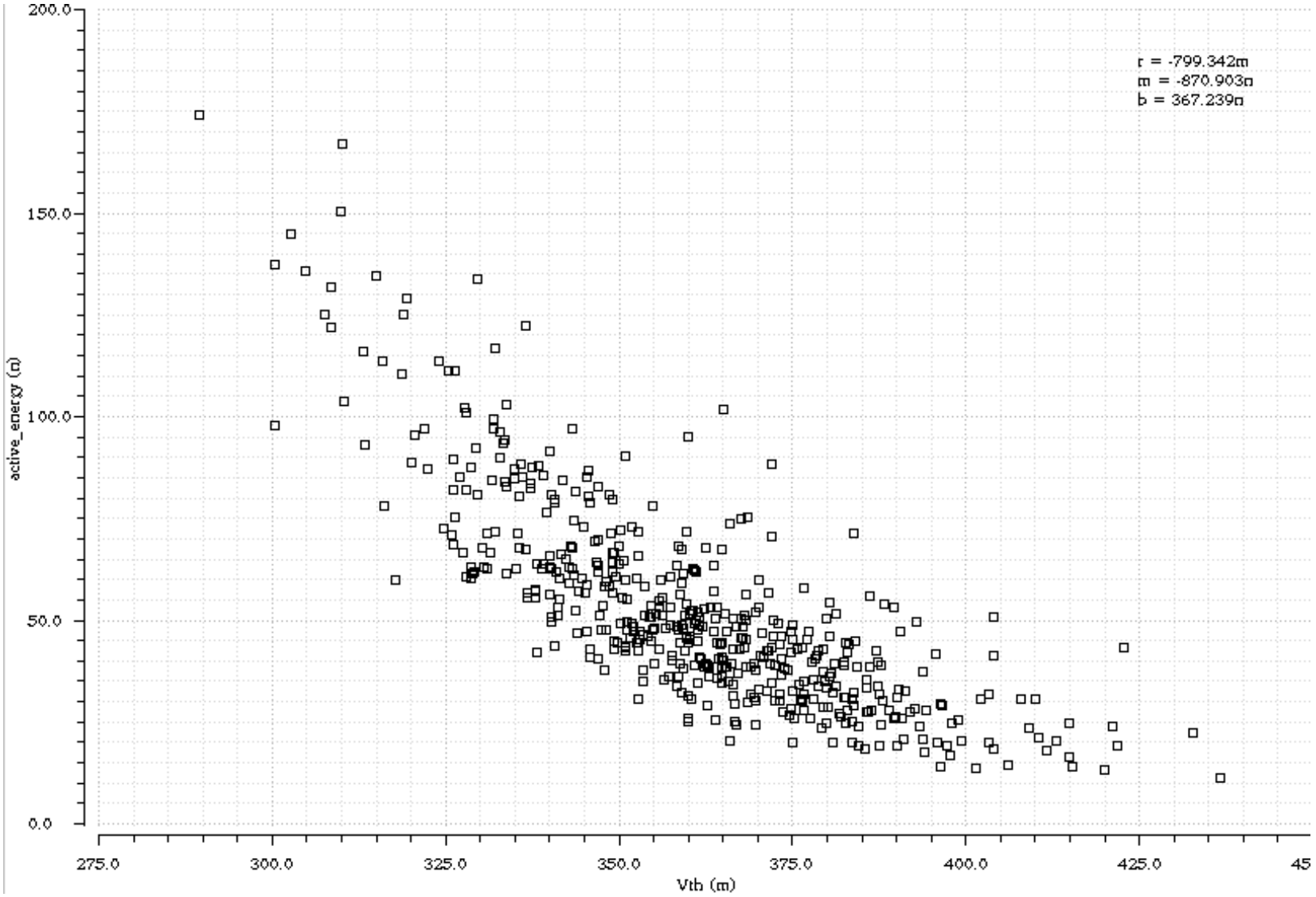

Figure 57: Active energy versus threshold scatter plot on bulk-silicon static synchronous ripple-carry adder at 90 mV over 500 samples.

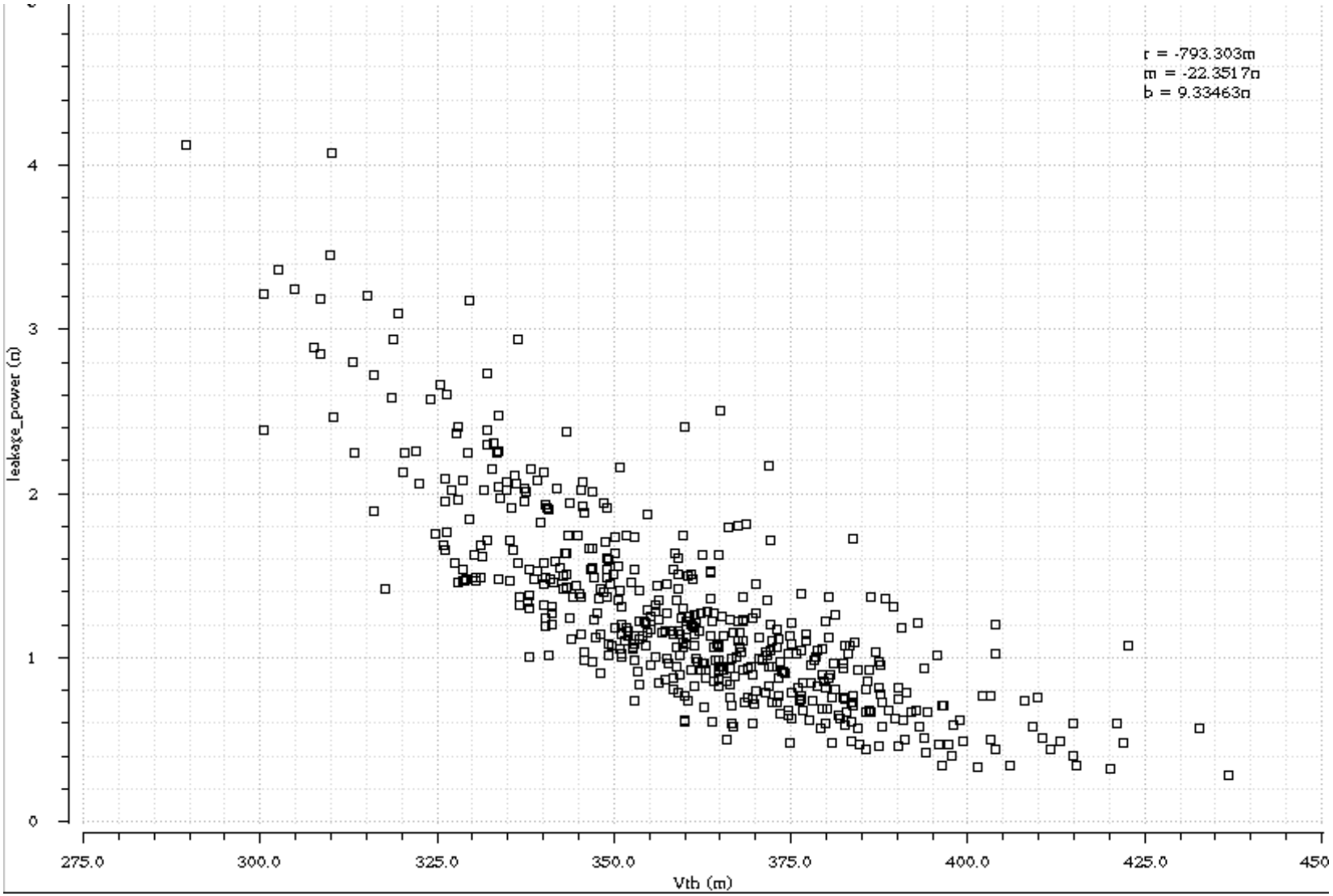

Figure 58: Leakage power versus threshold scatter plot on bulk-silicon static synchronous ripple-carry adder at 90 mV over 500 samples.

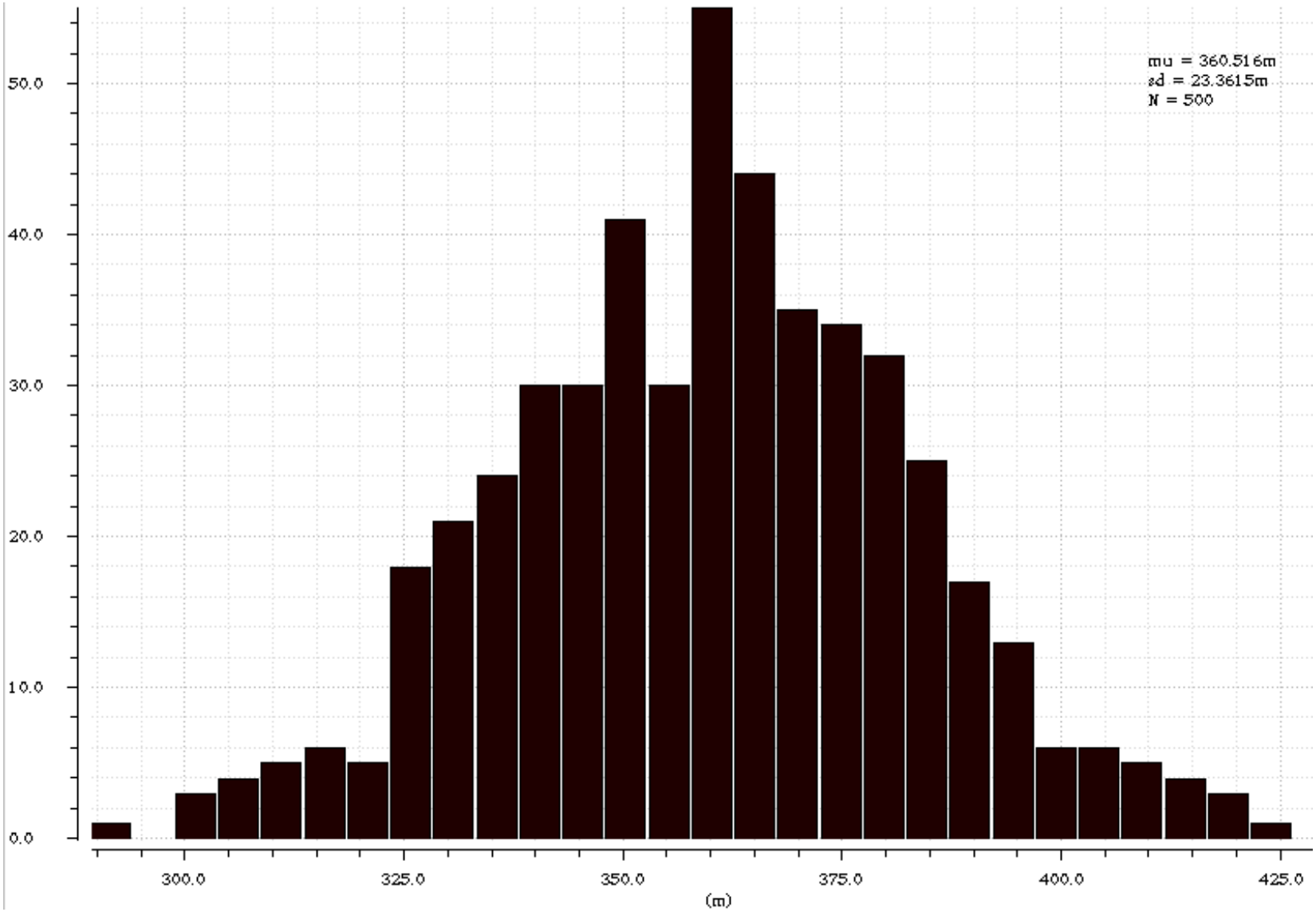

Figure 59: Threshold histogram on bulk-silicon static synchronous ripple-carry adder at 90 mV over 500 samples.

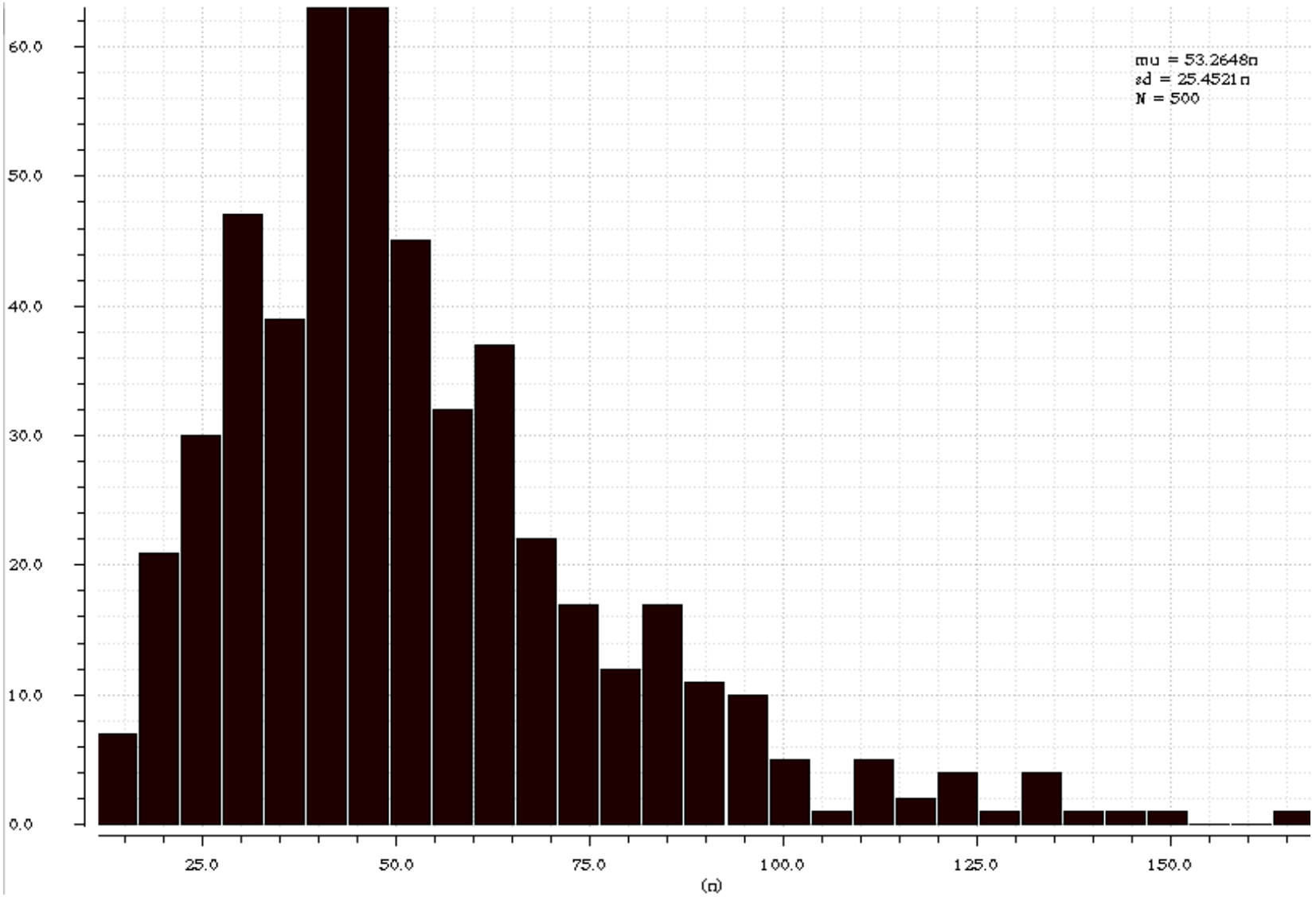

Figure 60: Active energy histogram on bulk-silicon static synchronous ripple-carry adder at 90 mV over 500 samples.

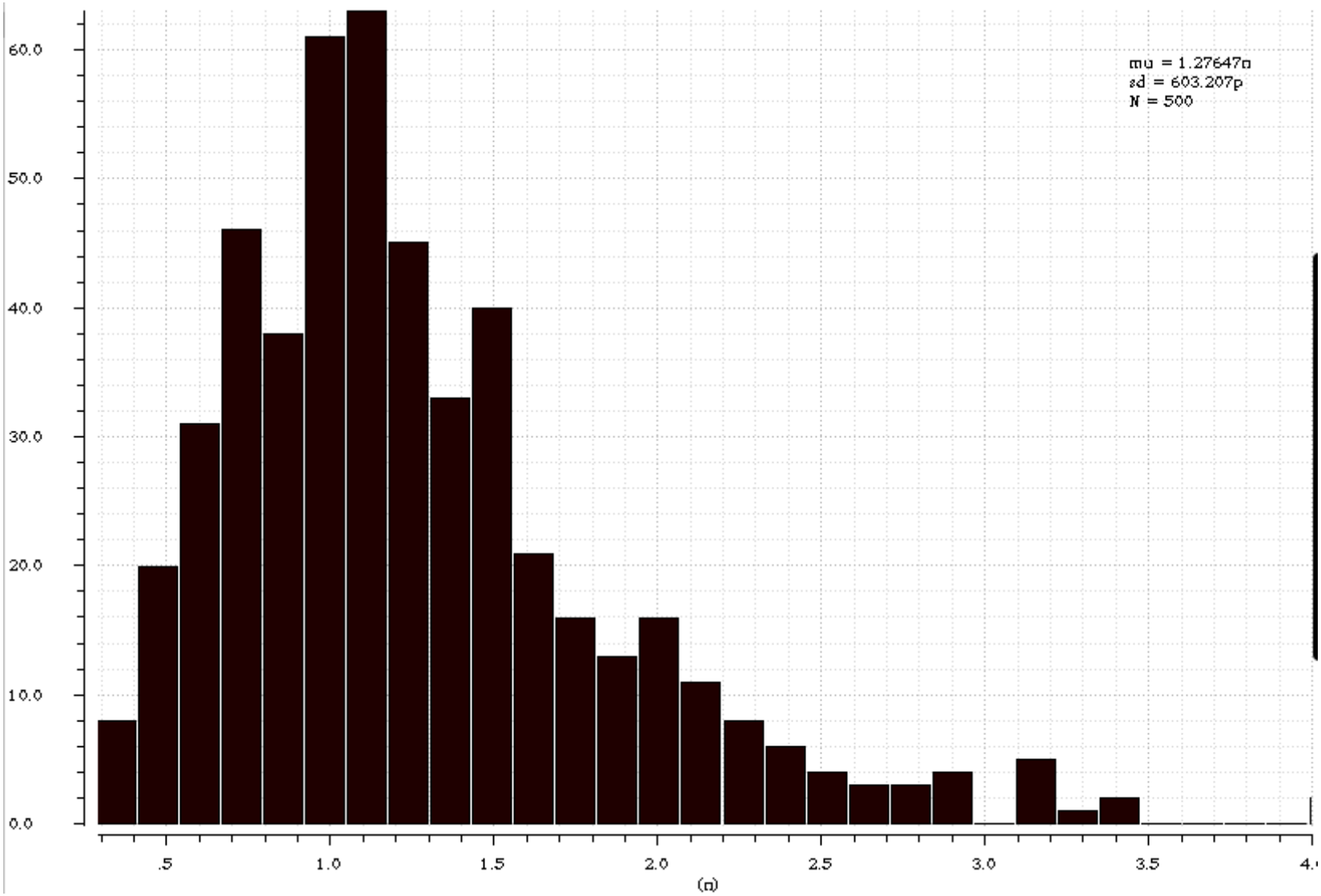

Figure 61: Leakage power histogram on bulk-silicon static synchronous ripple-carry adder at 90 mV over 500 samples.

Table 10: Bulk-silicon static synchronous ripple-carry adder measurements at 90 mV over 500 samples.

	Minimum	Maximum	Mean
V_T (mV)	289	437	361
Energy (pJ)	11700	174300	53260
Power (nW)	0.3	4.1	1.3
Success (%)			41

Figure 56–61 and Table 10 are results from Monte Carlo analysis on the static CMOS ripple-carry adder. The scatter plots show a strong negative correlation between power consumption and threshold voltage. Success rate degrades significantly to 40 percent with a 20 percent variation in threshold voltage, and success rate increases near the mean threshold voltage of 361 mV as indicated in by the top line of samples in Figure 56. Shown in the histograms, most samples converge around the mean threshold, active energy of 40 to 45 nJ, and leakage power of 1.0 to 1.1 nW.

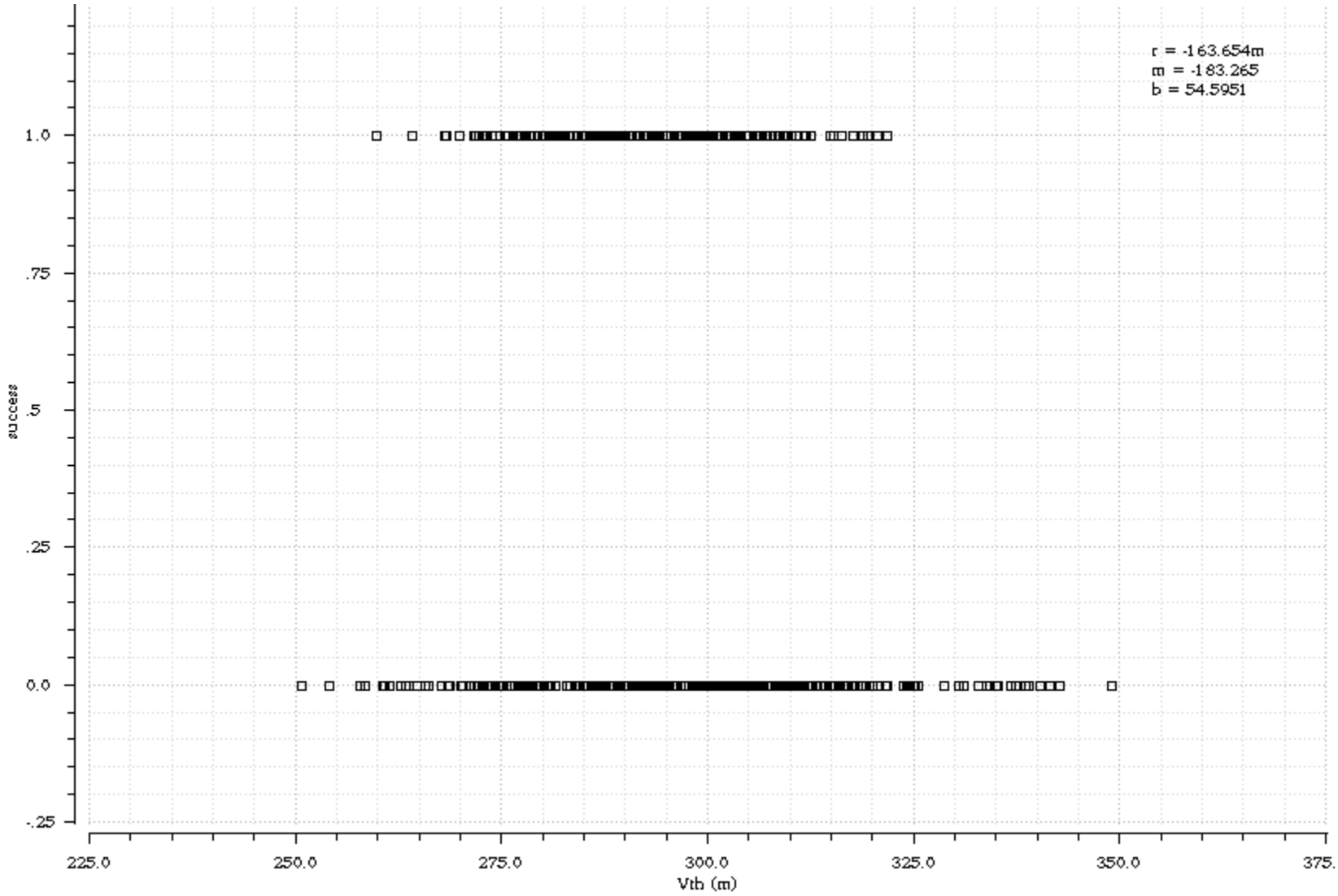

Figure 62: Success versus threshold scatter plot on bulk-silicon Schmitt-trigger synchronous ripple-carry adder at 70 mV over 500 samples.

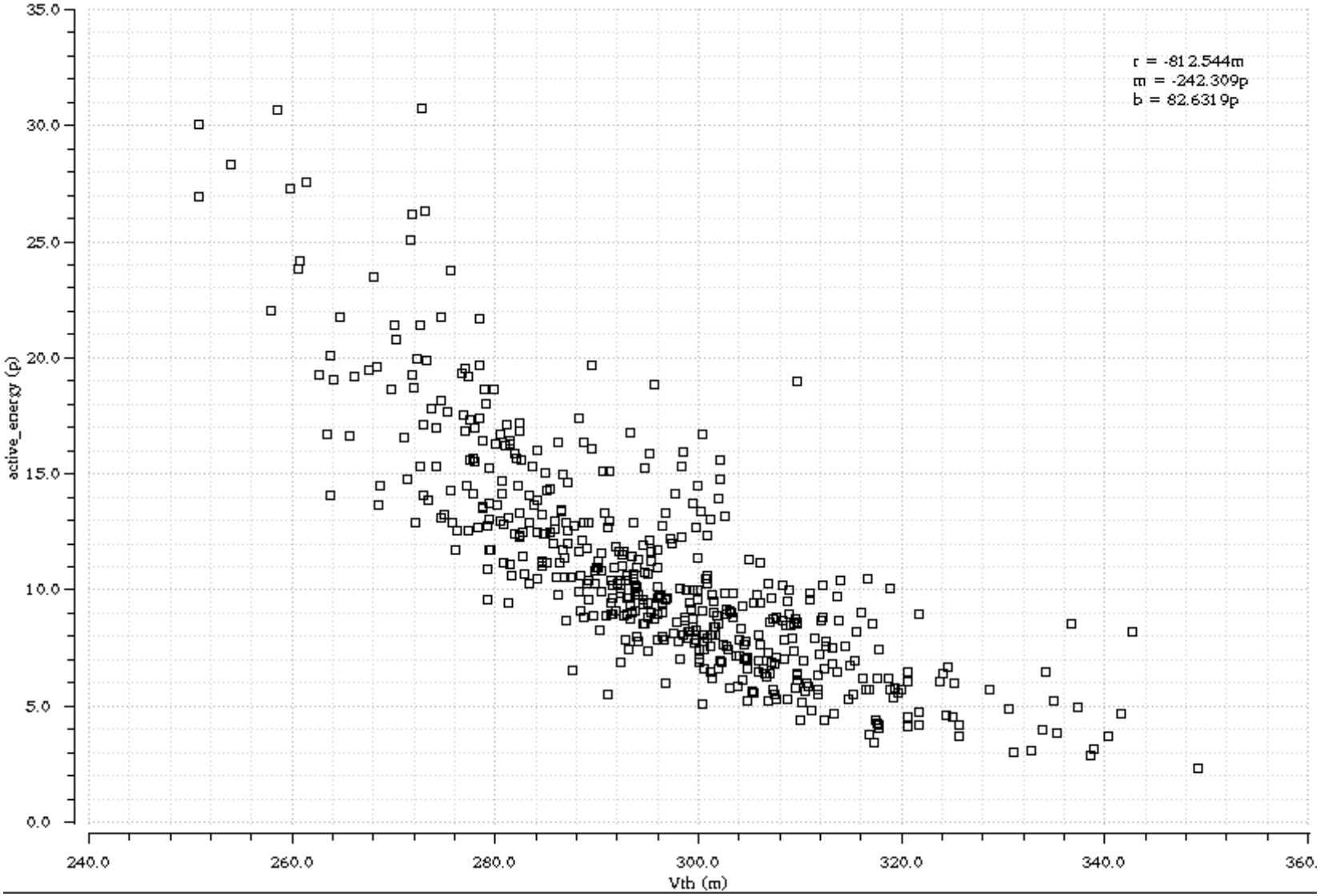

Figure 63: Active energy versus threshold scatter plot on bulk-silicon Schmitt-trigger synchronous ripple-carry adder at 70 mV over 500 samples.

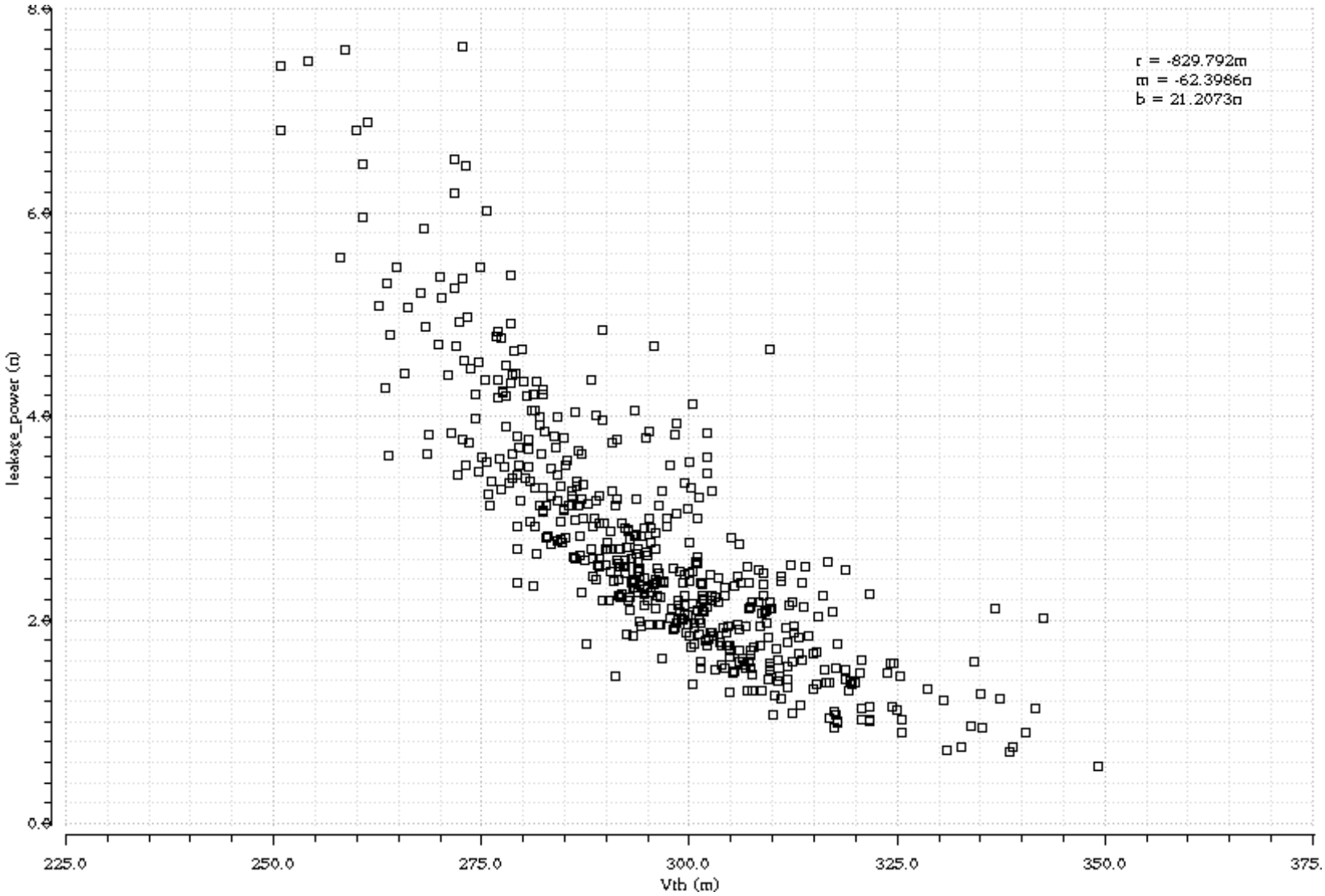

Figure 64: Leakage power versus threshold scatter plot on bulk-silicon Schmitt-trigger synchronous ripple-carry adder at 70 mV over 500 samples; from top to bottom.

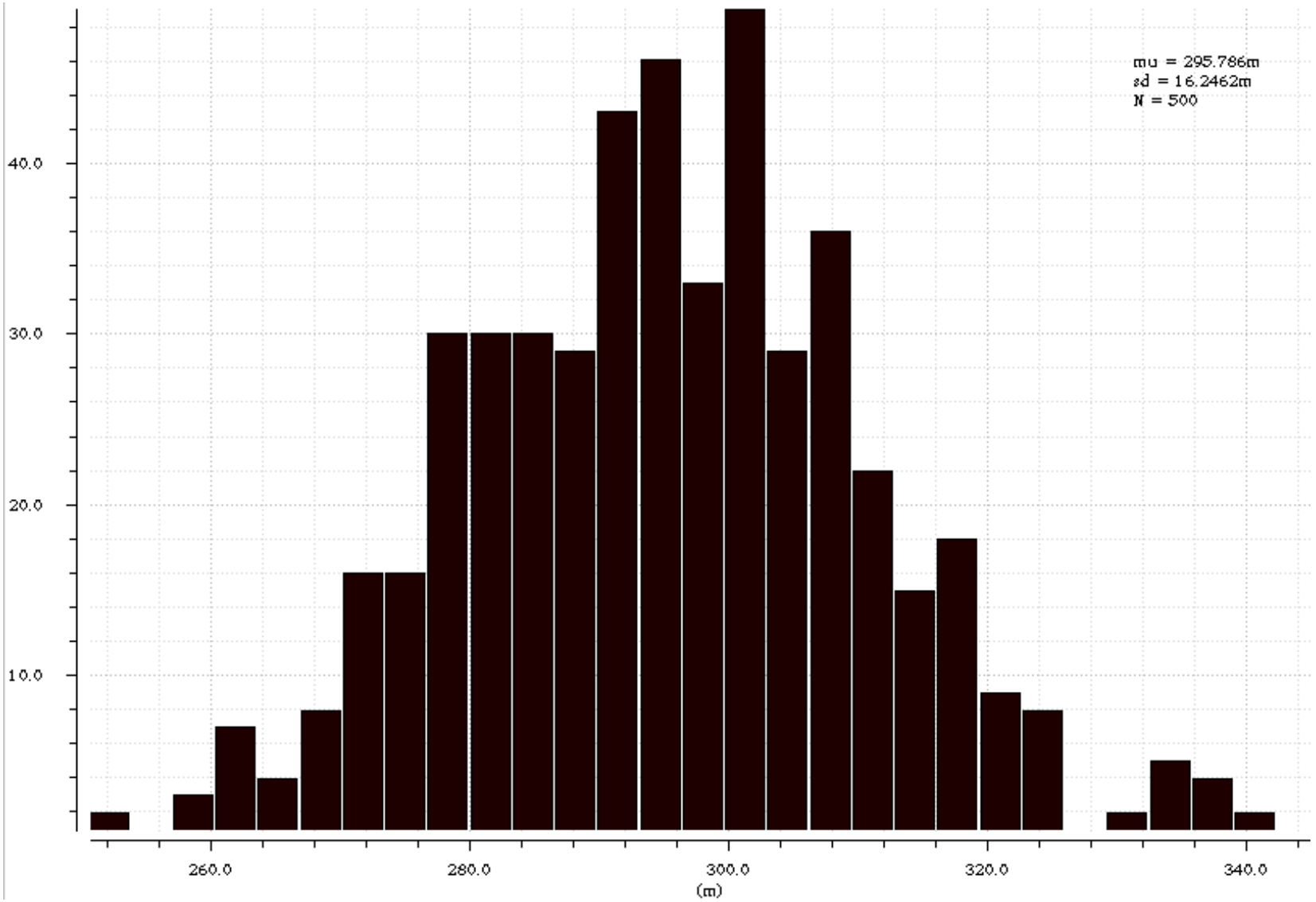

Figure 65: Threshold histogram on bulk-silicon Schmitt-trigger synchronous ripple-carry adder at 70 mV over 500 samples.

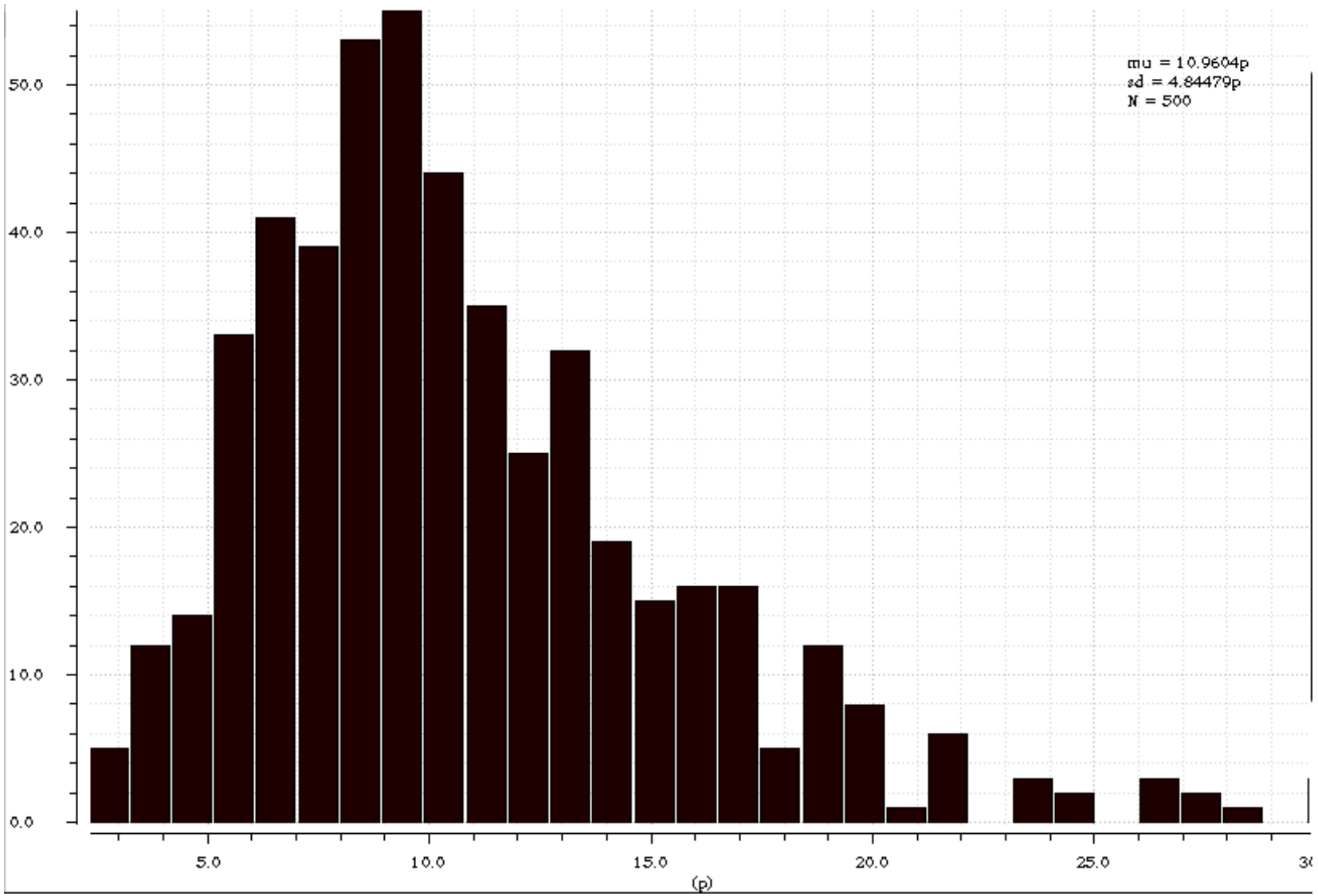

Figure 66: Active energy histogram on bulk-silicon Schmitt-trigger synchronous ripple-carry adder at 70 mV over 500 samples.

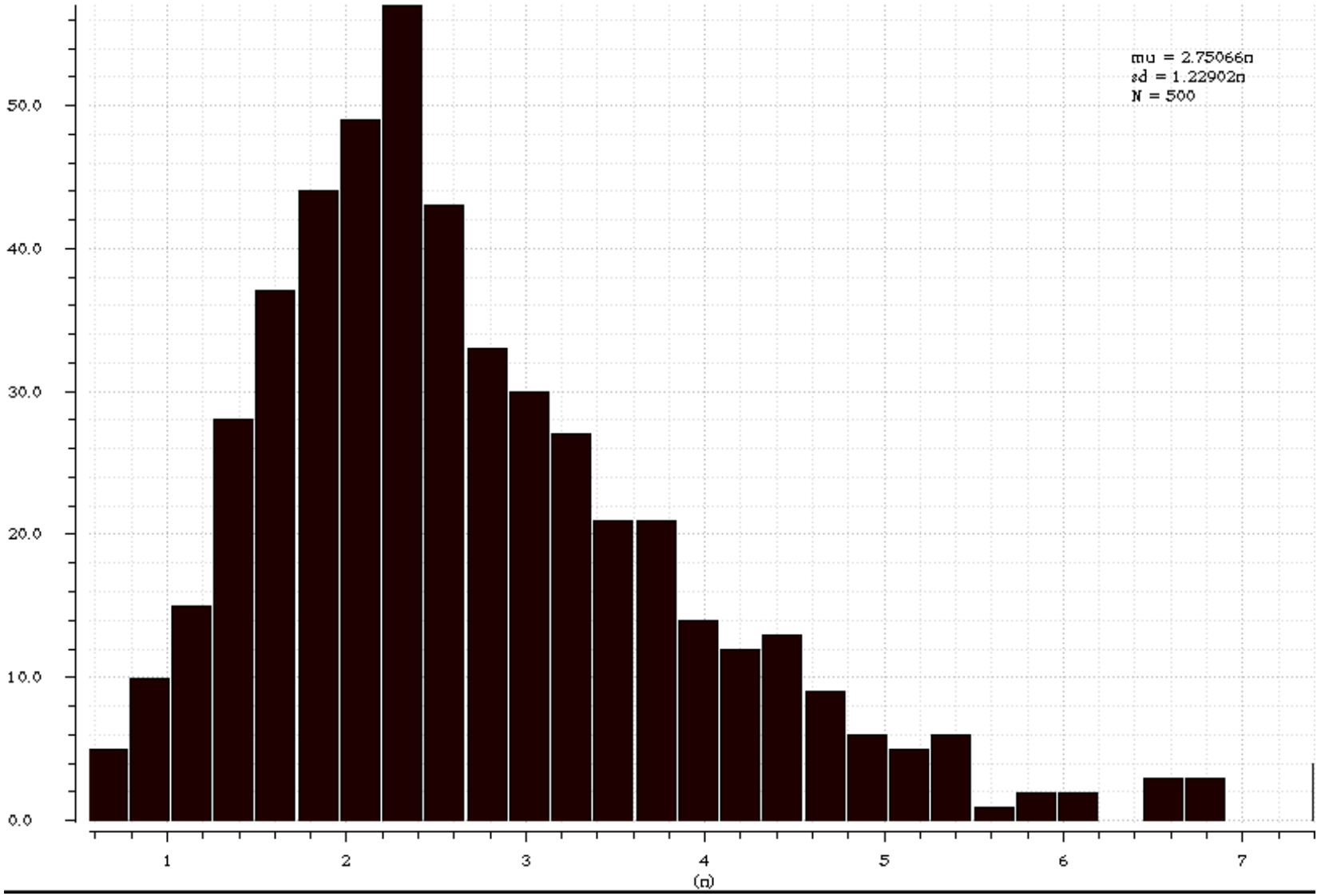

Figure 67: Leakage power histogram on bulk-silicon Schmitt-trigger synchronous ripple-carry adder at 70 mV over 500 samples.

Table 11: Bulk-silicon Schmitt-trigger synchronous ripple-carry adder measurements at 70 mV over 500 samples.

	Minimum	Maximum	Mean
V_T (mV)	251	349	296
Energy (pJ)	2.4	30.8	11.0
Power (nW)	0.6	7.6	2.8
Success (%)			39

Figure 62–67 and Table 11 are results from Monte Carlo analysis on the Schmitt-trigger ripple-carry adder. Similar to static CMOS, the scatter plot shows a strong negative correlation between power consumption and threshold voltage. Success rate degrades significantly to 40 percent with a 15 to 17 percent variation in threshold voltage, and success rate increases near the mean threshold voltage of 296 mV as indicated in by the top line of samples in Figure 62. Shown in the histograms, most samples converge around the mean threshold, active energy of 8 to 10 pJ, and leakage power of 2.0 to 2.2 nW.

2.5 ANALYSIS AND CONCLUSION

There are discrepancies and similarities between the SNM plots and measurements taken from each coprocessor design. All gate libraries show lower minimum supply voltage requirements than the supply voltage at which the respective coprocessor breaks down. However, the SNM plots follow a similar pattern as the coprocessor measurements. The Schmitt-trigger library SNM plots indicate lower supply voltage requirements, and the respective coprocessors operate correctly at lower supply voltages than their static counterparts. Comparing the static designs, the bulk-silicon static coprocessor operates at lower voltages than the FD-SOI counterpart, and the respective gate library SNM plots favor the bulk-silicon static gate library at lower voltages.

In contrast, the bulk-silicon Schmitt-trigger coprocessors operate at lower supply voltages than their FD-SOI counterparts, but the SNM plots indicate lower supply requirements for the FD-SOI Schmitt-trigger gate libraries.

Further research on SNM analysis is needed to improve prediction of minimum supply voltage requirements. The testbench from Figure 44 assumes that the gate pair represents the worst combination of gates. In the case of positive unate gates, the worst combination is if both gates drive a weak low or high output. However, choosing the worst combination of gates becomes non-intuitive for NCL gates and when transistor sizes are optimized. An additional step to take is extracting weak pull-up and pull-down VTCs from each gate. Another consideration is the fan-out load on each gate, which is not accounted for in the testbench. SNM analysis may be extended by determining the minimum required supply voltage for a given maximum fan-out, applying that fan-out as a load capacitance to each gate.

Results indicate that the Schmitt-trigger gate structure improves SNM and allows for operability at lower supply voltages. Another indication is that a large digital circuit implemented in NCL (e.g., the floating-point coprocessor) not only operates at lower voltages than a synchronous counterpart, it also supports a wider dynamic supply range when Schmitt-trigger gates are employed. The performance penalty of NCL's four-phase handshaking protocol is less critical since the data processing time inside logic gates becomes significantly longer. The clock period of synchronous design must account for the worst-case pipeline stage delay, but the NCL handshaking protocol absorbs delay variations across all stages and exhibits average-case performance.

Measurements taken from the FD-SOI process are mixed and therefore no conclusion is

made in comparing FD-SOI against the bulk-silicon process. Finally, yield analysis on the static synchronous ripple-carry adder suggest a significant dependence on the threshold voltage at ultra-low voltages.

Both Schmitt-trigger gate design and NCL prove to be beneficial at ultra-low voltages. For a given ultra-low voltage application, there is an optimum design methodology depending on supply voltage specifications (e.g., energy-harvesting power supplies), energy budget, and performance requirements. Other considerations are process node availability and real estate.

3 REMOTE ELECTRONIC UNIT DIGITAL CONTROLLER IC DESIGN FOR EXTREME ENVIRONMENTS

3.1 BACKGROUND

For a spacecraft, a number of sensors are needed to monitor its status in space. Shown in Figure 68, the sensor network consists of two systems: the sensor network controller and the remote sensor node. The sensor network controller communicates with each sensor node for configuration and sample data collection. Each sensor node is implemented by the remote electronic unit (REU). The REU is a two-chip module that processes samples from one or more sensors that each runs at a configurable sample rate. Sensors connect to the REU sensor interface (RSI), an ASIC that transforms analog signals into digital sample values. The RSI interfaces with the REU digital controller (RDC), which is the second ASIC that performs sample data averaging and over-sampling, yielding processed sample data that is sent to the sensor network controller. While the sensor network controller resides in a controlled environment, the REU must tolerate radiation and

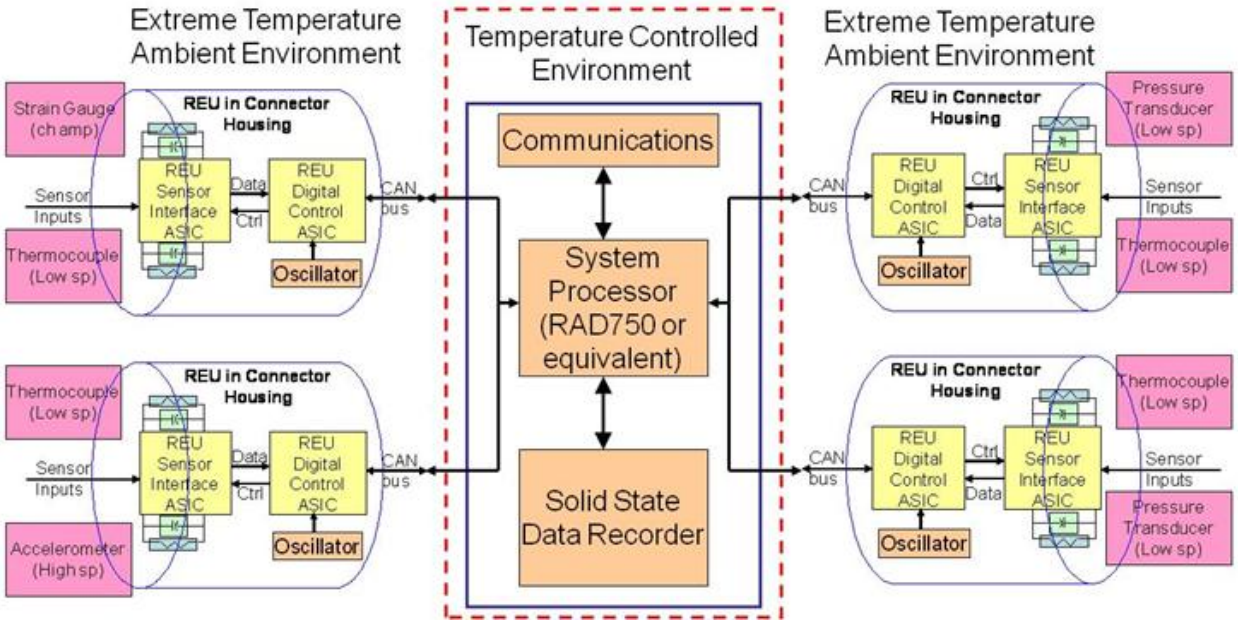

Figure 68: Remote sensor network architecture [31].

extreme temperatures from the space environment.

Figure 69 shows the RDC architecture. The REU digital controller is an 8-bit embedded system composed of a microcontroller, SRAM, ROM, interface controllers, sample data processors, and a multiplexed system bus. The microcontroller is an Intel 8031 processor that supports the 8051 instruction set. Firmware is supplied to the microcontroller by the ROM, and volatile software is supplied by the one-kilobyte SRAM modules. There are two interface controllers, the RSI interface and CANBus controller, which sandwich the sample data pipeline. The RSI interface contains a register file, accessible by the microcontroller, which defines settings for each sensor channel and controls channel initialization. The CANBus controller communicates with the sensor network controller over a transport protocol called controller area network (CAN), which layers on top of the standard ISO11898 two-wire connection. Similar to the RSI interface, the CANBus controller contains a register file that the microcontroller accesses to send and receive packets of

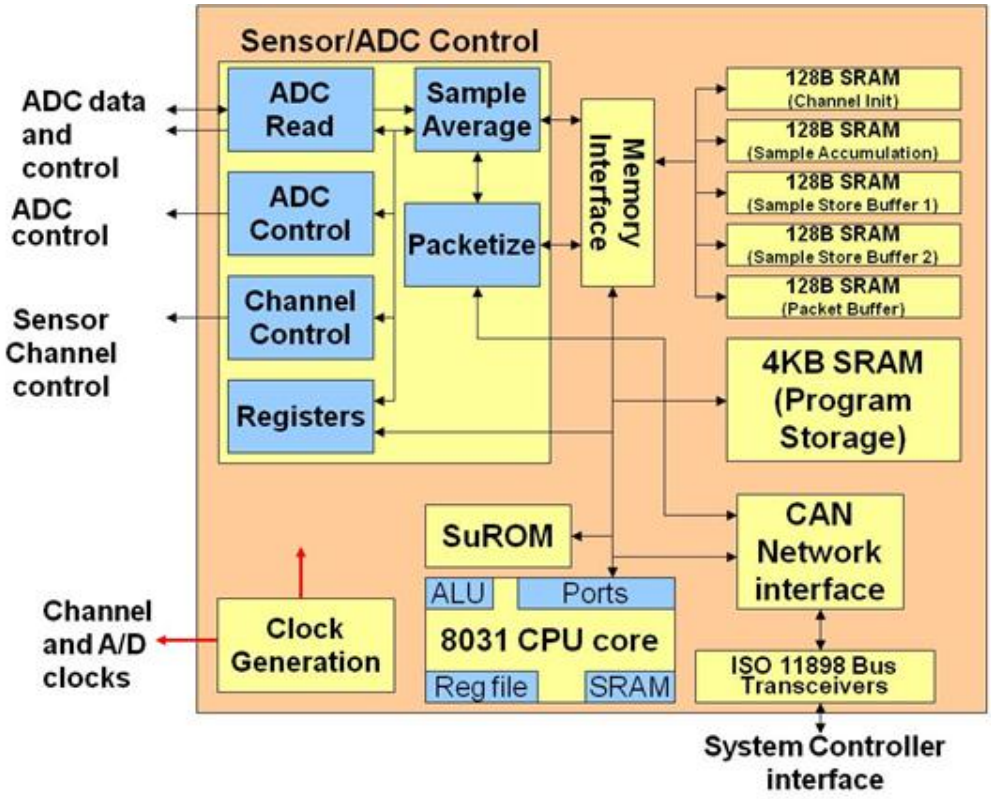

Figure 69: Block diagram of the REU digital controller's architecture [31].

data. Sample data is processed in two steps: sample averaging and CAN packetization. The first sample data processor accumulates samples into 128-byte SRAMs based on the oversample ratio and sample rate. The second sample processor monitors for averaged sample data and constructs CAN packets to be transmitted by the CANBus controller. Lastly, the multiplexed system bus connects the microcontroller and sample data processors to the interface controllers and memory modules.

Each component of the RDC consists of a variety of primitive building blocks. The microcontroller is an asynchronous system based on NULL convention logic (NCL) that contains an internal 128-byte SRAM and threshold gates. In addition, the NCL 8031 processor employs a synchronous wrapper controlled by an external clock that latches the inputs and outputs. This

synchronous wrapper is needed to interface with other components of the RDC, which make use of boolean gates and registers. Each primitive module, including SRAMs and ROM, is built to tolerate the space environment.

3.2 DESIGN METHODOLOGY OF RDC ASIC CORE

A commercially available process chosen for both the RSI and RDC is IBM's half-micron silicon-germanium (SiGe) process, SiGe 5AM [6]. This process is optimized for low-voltage applications, but supply voltage is capped at 3.3 volts for performance [13]. High radiation is a key concern for space operation and SiGe devices have a natural hardness to ionization [22]. The target range in temperature for lunar applications is from -180 °C to +120 °C. It has been demonstrated from a fully differential amplifier that the SiGe 5AM process performs well across the said temperature range [5].

Building the RDC involves several stages in order: cell library development, hardware description language (HDL) and firmware design, logic synthesis, automatic place-and-route, layout sign-off, and post-fabrication bonding and testing. Between stages are various methods of verification. A key characteristic of building a digital system is that details about the system accumulates in later stages, requiring incremental verification. For example, a circuit before logic synthesis goes through behavioral simulations. After synthesis when a cell netlist is given, the same circuit is analyzed for timing. The following sections describe each stage of design in detail.

The RDC references three primitive cell libraries: boolean gates, threshold gates, and 8-bit wide memory array devices. The boolean gate library composes the majority of the digital controller. The NCL 8031 processor contains an internal 128-btye SRAM and threshold gates.

Other memory devices utilized throughout the RDC are 64-byte FIFO SRAMs used by the CAN-Bus controller, 128-byte SRAMs used by the sample data processors, one-kilobyte SRAMs with error detection and correction that store supplementary firmware, and a one-kilobyte ROM that stores boot-up firmware. All three cell libraries are delivered as HDL models for logic simulations, schematics and layouts, and Synopsys Liberty [28] code for synthesis and timing analysis across a range of temperatures.

The boolean gate library includes standard synchronous logic (e.g., NAND, NOR, XOR, etc.) and scan flip-flops. The scan flip-flops allow selection between two different data sources and are part of a technique known as design-for-test (DFT). These flip-flops effectively form two circuits, the data pipeline of the RDC and scan chains used in manufacturer testing. All boolean gates are static CMOS circuits, which guarantee that each signal has at least one DC path to either power or ground.

Several design and layout techniques are applied to the boolean and threshold gate libraries. All gate layouts share two characteristics that make each gate tolerate to extreme environments; Figure 70 presents one of the boolean cell layouts. The first characteristic is the guard ring, a series of substrate contacts surrounding the layout. The guard ring mitigates latch-up in MOSFETs [34] and provides cross-talk immunity [12]. Another layout technique used is doubling the length and width of each NMOS transistor. For the half-micron process, the minimum lengths are 0.5 μm for PMOS and 1.0 μm for NMOS. Having longer channel lengths improves the lifetime of MOSFETs under extreme temperatures at the expense of increased real estate [9]. The flip-flops are dual interlocked cells (DICE). A DICE flip-flop has four storage nodes with four interconnected

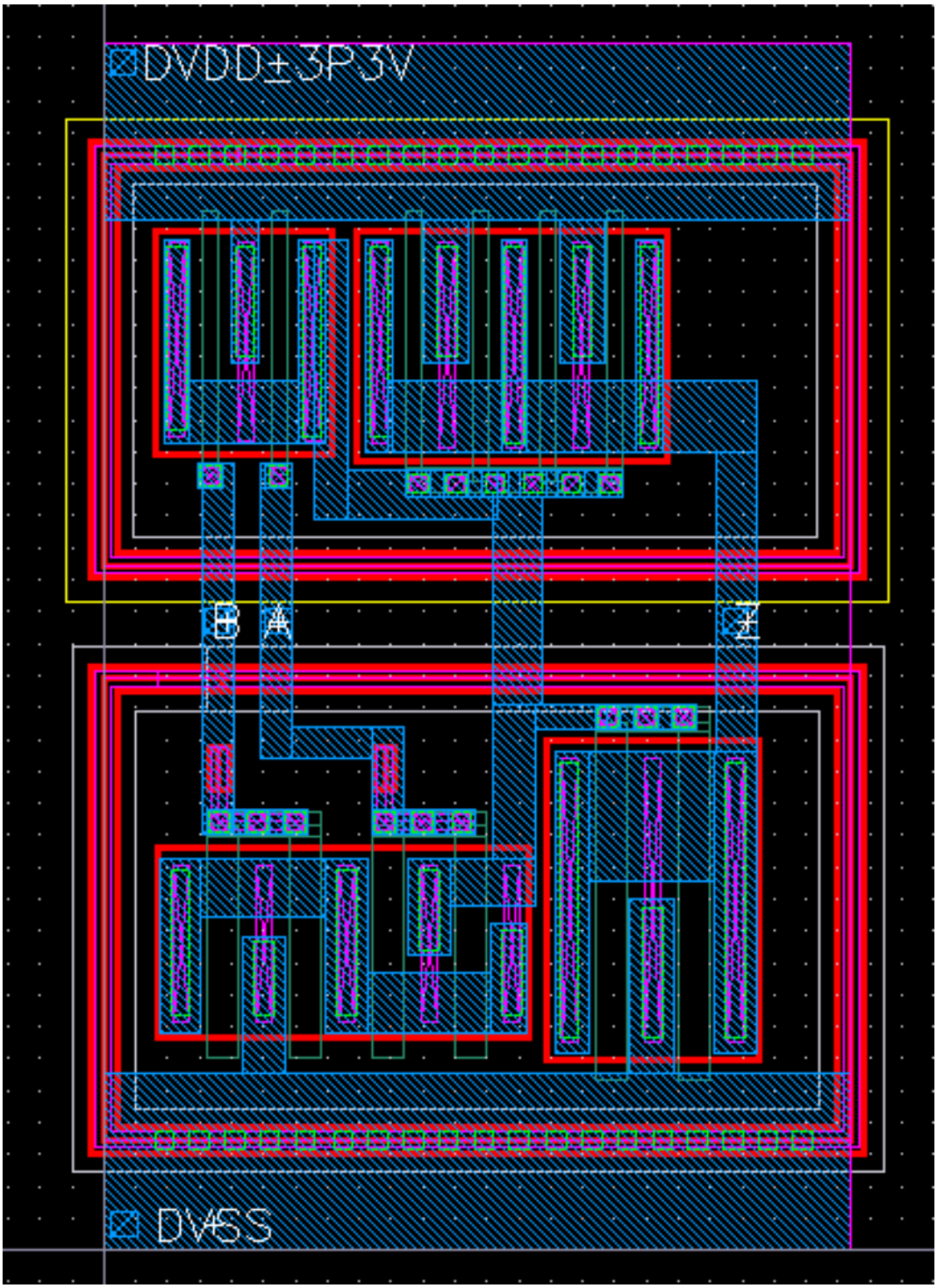

Figure 70: SiGe 5AM layout of a two-input AND gate with 16 times drive strength than a minimum inverter.

inverters, and the purpose of DICE is to add immunity to single event upsets caused by charge sharing [1].

In regard to automatic place-and-route, boolean and threshold cell layouts follow strict dimension and placement rules; memory array cells do not follow such rules since large primitive cells are manually placed. Cell heights are identical while cell widths are multiples of the minimum cell width. These cell width and height rules prevent odd gaps from appearing in the final layout. Input and output pin placement is grid-based, and the pin placement grid divides the cell into squares. The pin placement grid is necessary for automatic routing.

All cell libraries come in three forms. The first form created is schematic and layout. Cadence is the primary tool environment used to manage schematics and layouts [2]. The schematics and layouts lead to additional cellviews (Cadence's term for a basic unit of design) used for accurate timing analysis and automatic place-and-route. The second form is Synopsys Liberty code, a versatile language that describes each cell's functionality, area, power, and timing given certain environmental conditions. Liberty code applies to synthesis and timing analysis. The third form is the HDL model, which is written in Verilog and is used in both functional and timing logic simulations.

There are three versions of Liberty code: best-case, worst-case, and nominal. Because timing is the primary concern for the RDC, namely the synchronous logic and memory arrays, the corners cover a range from worst-case to best-case timing. Worst-case timing defines a high temperature (+120 °C), a low supply voltage (3.0 V), and slow process skew; best-case timing defines a low temperature (-180 °C), a high supply voltage (3.6 V), and fast process skew; nominal

timing defines room temperature (25 °C), the target supply voltage (3.3 V), and no process skew. The worst-case and best-case timing corners reveal setup and hold time violations respectively, and the nominal timing corner models expected circuit timing.

The process of developing the Liberty code is called cell characterization. One of the first steps in cell characterization is extracting parasitics from each cell layout. Parasitic extraction is an automated task performed in Cadence that annotates each transistor-based schematic with resistors and capacitors. Another task in characterization is Liberty template construction. A Liberty template, for the most part, contains code that will appear in the final Liberty code, specifically environmental conditions, cell functionality, input and output pins, and area. Liberty templates also contain characterization commands that describe relationships between each output and inputs, input vectors to apply, input transition times to simulate, and output load capacitances to apply. With the Liberty templates in place, the next step in characterization is simulating each cell's parasitic-extracted transistor schematic. Running simulations on each cell is automated by characterization tools, namely Synopsys NCX [30] for the RDC project. The results from timing characterization are propagation and transition tables per output pin with respect to each input pin rising or falling in voltage; each table is indexed by the same set of input transition times and output load capacitances.

The RDC is organized in hierarchy. The major components instantiated at the top level of the RDC are the NCL 8031 microcontroller, a one-kilobyte ROM, two one-kilobyte SRAMs with error detection and correction, five 128-byte SRAMs, a system clock divider, a reset signal generator, a manufacturer testing control block, the RSI interface, the CANBus controller with

a redundant ISO11898 interface, the sample accumulator, the sample packetizer, and the system bus interface. The NCL 8031 microcontroller is a project developed from [10]. The CANBus controller is based on an OpenCores project [21] and contains synchronous logic and two 64-byte FIFO SRAMs. The SRAMs and ROM are manually built in schematic and layout, and they are modeled in behavioral HDL code. All other modules are developed in RTL HDL code.

There are two programs developed for booting up the RDC and subsequent operations in accepting commands from and sending sample data to the sensor network controller. The boot-up program is permanently stored in the one-kilobyte ROM while the supplementary program is later fetched from the network controller and stored in the two one-kilobyte SRAMs. At the beginning of operation, the RDC fetches and executes the firmware from ROM. The boot-up firmware commands the RDC to send a message to the network controller indicating that it is ready to accept commands. In response, the network controller sends back program data in the form of memory-write commands followed by a command to execute the sent program. The boot-up firmware transfers execution to the supplementary program stored in SRAM. With the supplementary program in execution, the RDC accepts further commands from the network controller, most notably channel initialization and sample collection commands. Both the boot-up and supplementary firmware include programmable hardware interrupts, which are tied to the CANBus controller for received messages and the RSI interface for channel initialization.

The firmware is written in an annotated variation of the C language that supports constructs specific to an 8051-based processor. Among the 8051-specific constructs supported include defining interrupt routines, referencing the 8051 registers and ports, and addressing external data

memory such as the SRAMs and register files controlling the RSI interface and CANBus controller. A cross-compiler known as small-device C compiler (SDCC) [7] maps the annotated C code into 8051 assembly code followed by translating the said assembly code into bytes of machine code. The machine code is stored in a standard format called Intel HEX that associates each address with a byte of program data. The Intel HEX format is later referenced to program the ROM's layout and in logic simulations to populate the ROM and SRAM memory models.

A synthesis script is broken into six parts. First, synthesis references the cell Liberty files. The nominal cell Liberty code is selected to be the target cell library, indicating that room temperature conditions is assumed when mapping logic and reporting timing. The worst-case and best-case cell Liberty code are selected to be the maximum and minimum cell libraries respectively. At the time Design Compiler optimizes the netlist for timing, the maximum Liberty code is referenced for setup time violations while the minimum Liberty code is referenced for hold time violations. Following cell library referencing, synthesis reads the RTL HDL code and checks it against the cell libraries, making sure that the HDL code is synthesizable. In the third part, the synthesis script defines constraints such as target clock frequencies. Because the RDC utilizes scan chains, the next part sets up DFT [29] by specifying signals related to manufacturer testing, namely the reset signal, scan enable input, scan clock, scan data inputs, and scan data outputs. The fifth part of the synthesis script compiles the RTL code by simultaneously mapping RTL to gates, optimizing the resulting gate netlist, and creating the scan chains. In the sixth part, synthesis concludes by exporting the gate netlist to Verilog as well as saving all data into Design Compiler's native format.

Defining constraints is an important step in synthesis. The most dominant constraint is

the target clock frequency, and Design Compiler accepts commands that associate each clock net with either a constant or derived frequency. Design Compiler differentiates external clocks from internally generated clocks. In the case of the RDC, there is a primary clock input that feeds a internal clock divider. The primary clock input is assigned 85 MHz, defined in specifications, while outputs from the clock divider are each defined as a constant division of the primary clock net. Two other constraints are maximum transition time and maximum load capacitance. The maximum load capacitance constricts the fan-out of each gate and is determined at the time the cell libraries are developed. Setting the maximum transition time causes Design Compiler to buffer signals that fail to transition fast enough.

After extracting blockage abstracts, the resulting layout exchange format (LEF) code feeds into the place-and-route tool along with the cell Liberty code, the circuit's cell netlist produced from synthesis, and constraints carried over from synthesis. The place-and-route tool has sufficient information to create a floorplan. A floorplan specifies the circuit's core dimensions, border space for power and ground rings, and rows in which to place the cells. The cell rows are double-backed such that the power and ground rails between rows join together. Figure 71 shows a floorplan of the RDC and its major components.

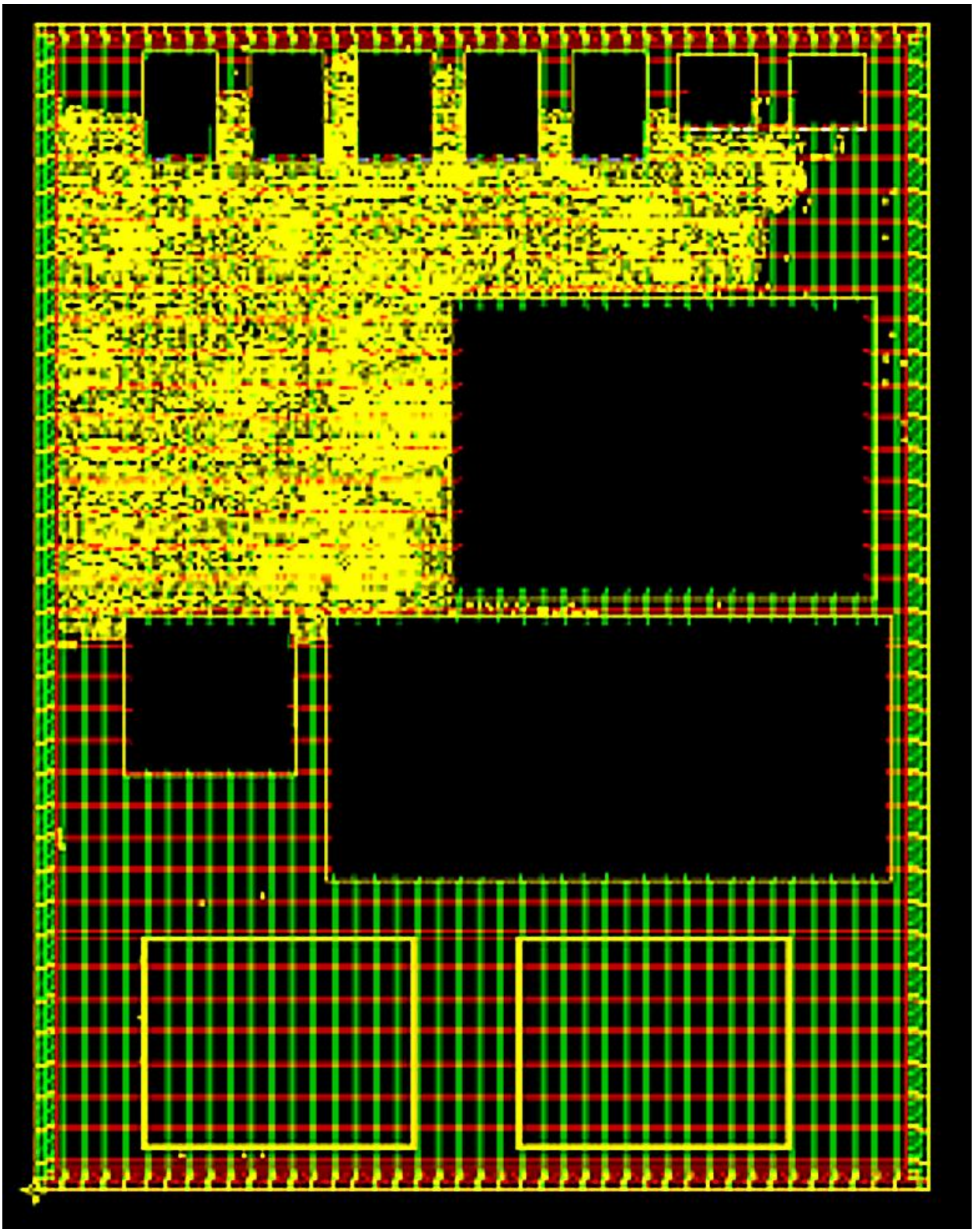

Figure 71: A floorplan of the REU digital controller after cell and power rail placement.

3.3 VERIFICATION

While building the digital controller, each step is verified. Cell libraries run through a variety of checks, including DRC and LVS, before being released to later stages of development. Cell characterization verifies each cell through parasitic-extracted transistor-level simulations. The RTL HDL code is checked against specifications while ensuring that it sufficiently describes hardware. All gate netlists are logically matched against the RTL HDL code, and the gate netlist optimized during place-and-route goes through timing analysis to ensure its operability among all operating conditions. Once the digital controller's layout and schematic are built, several steps are taken to sign off the layout before submitting for fabrication and packaging. After fabrication and packaging, each copy of the RDC ASIC is tested using equipment that feeds sample data and communicates over the CANBus interface.

The purpose of the testbench suite is to verify the RDC as a fully intact system by monitoring outputs from the RDC and feeding it input vectors when needed. The testbench suite is divided into several components based on the devices that the RDC will connect to after fabrication. Those components are the testbench suite controller, a CANBus host, a sample data generator, an RSI control monitor, a clock generator, and a power source model. Both clock generator and power source model are simple components that provide the external clock with reset and constant logic values respectively. The testbench suite controller is the primary component that controls simulation and all other components. In addition, it manages a log system used by all testbench components to report states of operations. The CANBus host communicates with the RDC over CAN and behaves similar to the sensor network controller. The sample data generator reads 12-bit

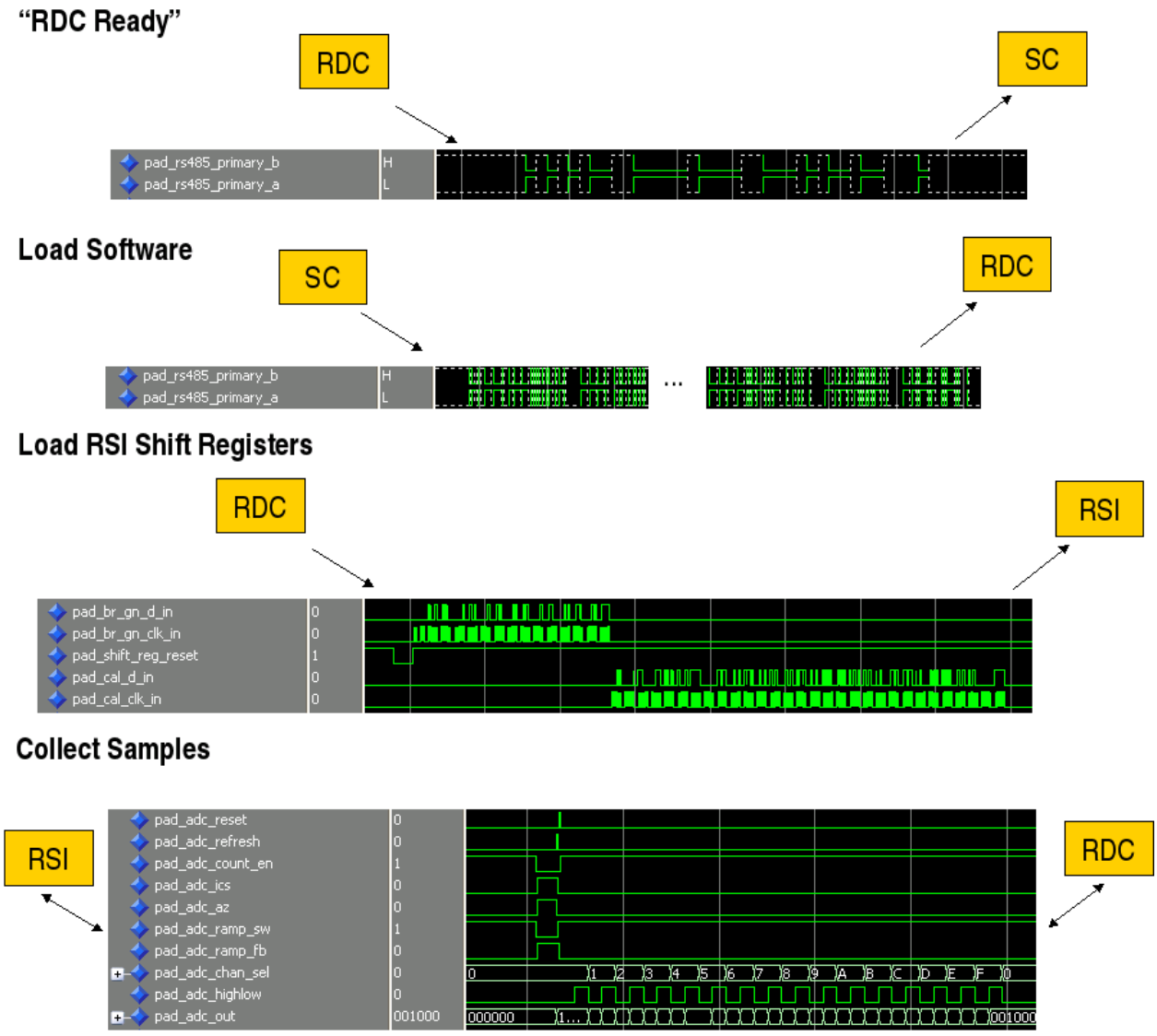

Figure 72: Logic waveforms generated from Mentor Graphics ModelSim showing the REU digital controller's flow of operation.

sample data from a file and drives the six-bit ADC data bus going into the RDC. The RSI control monitor performs a variety of tasks related to the RSI control signals driven by the RDC. Those tasks are verifying certain sequences of RSI control such as channel initialization and raising flags when the RSI control signals meet certain conditions. Some of the components communicate with one another by raising certain events. For example, the RSI control monitor raises a flag whenever the RDC requests for sample data through its RSI interface, and the sample data generator is triggered by that flag and feeds in the next sample data. Another example is the testbench suite controller waiting for notification from the RSI control monitor that sample collection has stopped before terminating simulation.

Logic simulations follow through a sequence of operation that the RDC normally performs. At the beginning of simulation, the ROM model populates its internal memory array with the machine-encoded boot-up firmware, and the power source model sets constant logic values to power, ground, and RDC inputs related to manufacturer testing. The clock generator resets the RDC before supplying a clock and notifying other components that the RDC is running. At this point, the 8031 microcontroller in the RDC begins executing the boot-up firmware. The boot-up firmware first sends a message out to the CANBus indicating that the RDC is ready to accept commands. The CANBus host component responds by sending CAN packets of supplementary firmware to the RDC, which in turn stores the firmware into the one-kilobyte SRAMs. Next, the CANBus host testbench sends a command to execute the supplementary firmware, and the 8031 microcontroller transfer execution to the code stored in the one-kilobyte SRAMs. The supplementary firmware itself sends a message to the CANBus host that it is accepting commands. The

CANBus host sends channel initialization data to the RDC, which is stored in a 128-byte SRAM, before transmitting a command to initialize the RSI channels. During channel initialization, the RSI control monitor checks the channel data sent from the RDC's RSI interface, reporting an error if the channel data does not match. After the RDC initializes the RSI channels, the firmware responds with another message that it is ready to accept commands. The CANBus host sends several commands to configure and start sample collection. The RDC will start requesting for sample data through its RSI interface. The RSI control monitor raises a flag to feed in the next six bits of sample data, and the sample data generator fetches the next piece of data from file. The RDC processes the sample data and eventually starts sending sample data packets to the CANBus, depending on the sample rates of each channel. The CANBus host receives the sample data packets, calculates expected data given the sample rate configuration and the sample data file referenced by the sample data generator, and reports errors for mismatching data. Finally, the testbench suite controller raises an flag to terminate simulation, and the CANBus host sends a command to the RDC to stop sample collection. The testbench suite controller waits for the RDC to stop requesting for sample data before stopping simulation. If the RDC continues requesting for sample data long enough, the testbench suite controller halts simulation with an error. Figure 72 shows relevant logic waveforms that occur during each sequence of operation.

After static timing analysis (STA) and buffer insertion, timing logic simulation tests the functionality of the circuit against different environmental conditions. Timing logic simulation shares the same setup as ideal logic simulation except that cell and wire delays are included. ModelSim supports the standard delay format (SDF), which closely resembles Verilog's syntax for

specifying timing arcs. SDF mirrors the digital controller's cell netlist hierarchy with cell and interconnect propagation delays; each cell instance and wire inherits unique rise and fall delays. An SDF file is generated during STA for each of the three timing corners. Under worst-case timing, STA and timing logic simulation reveal that 64 MHz is the maximum clock frequency supported by the RDC.

3.4 RDC ASIC TESTING

The RDC ASIC is tested separately from the RSI ASIC under room temperature. A test board is designed to facilitate testing the RDC and its major components. Figure 73 shows a general block diagram of the test board architecture. RDC's interface divides into four parts: clock and reset, RSI control and data, CANBus ID and interface, and DFT interfaces for the RDC and 8031 microcontroller. Each part communicates with various test equipment. While Figure 73 represents an ideal setup that is suitable for cryogenic test runs, some parts are replaced to expedite room temperature testing. A pattern generator replaces the on-board clock oscillator, power-on reset generator, and the RSI FPGA. The pattern generator is given input vectors to initialize and run the RDC ASIC while a logic analyzer monitors relevant output signals. The CANBus ID is driven by DIP switches, and the CANBus two-wire interface connects to a CAN-USB PC adapter. Software is developed on a PC to drive the CAN-USB adapter and monitor for CAN packets sent by the RDC ASIC.

The CAN-USB adapter software performs the tasks of sending and receiving CAN packets. The software is tailored for the part of the RDC being tested. Because the majority of RDC features are available over CAN such as memory and register file access, most tasks in testing are incor-

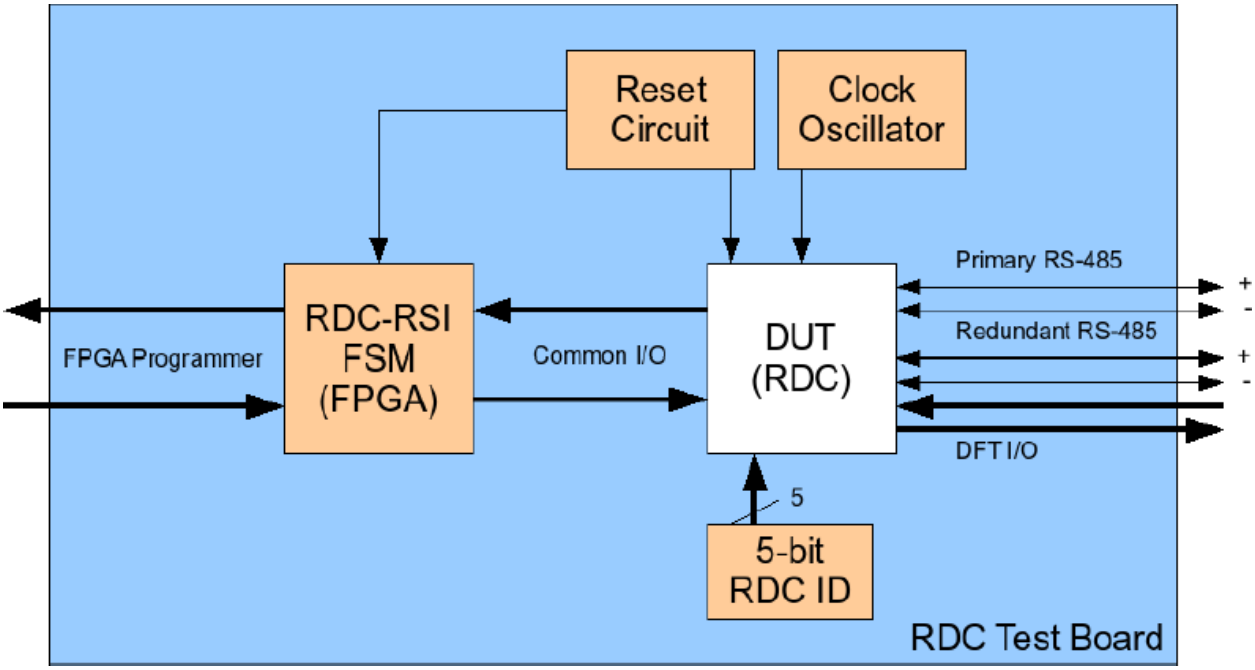

Figure 73: General block diagram of RDC ASIC testing.

porated in the software. The tasks performed by the CAN-USB software is similar to that of the sensor network controller. Those tasks begin with delivering supplementary firmware, followed by channel configuration and initialization, and commanding sample collection. Simplified versions of the CAN-USB software are also developed to perform basic CANBus communications.

Due to issues between the 8031 microcontroller and the external one-kilobyte ROM, the microcontroller cannot boot-up on its own. Therefore, the microcontroller's boundary scan interface is required to boot-up the RDC. The boot-up firmware is reformed to execute from the one-kilobyte SRAMs and to disable all hardware interrupts since the ROM stores interrupt vectors. After compilation with the small device C compiler (SDCC) [7], the modified firmware is translated into a series of instructions that write the firmware into external SRAM. Each byte of those instructions are fed into the microcontroller's boundary scan input chain for execution. The pattern generator initiates each session by injecting the modified firmware into SRAM and transferring execution to

it.

With the help of the microcontroller's boundary scan interface, two basic programs are separately written into SRAM to test the RDC's CANBus and RSI interface components. The first firmware checks the RDC's ability to communicate over CAN. It listens for CAN packets from the CAN-USB adapter and echoes received CAN packets back to the CANBus, confirming that the RDC ASIC receives and sends CAN packets. The second firmware tests the RSI interface in the RDC by running channel initialization followed by sample collection. On success, the RDC's sample data pipeline is verified.

3.5 REVISIONS RELEASED

This section briefly describes each revision of the RDC ASIC. Design methodology and verification evolved through each revision. Issues and their effect on design and verification are explained.

3.5.1 CRYO5 RDC1

The first RDC release started the initial development of the RTL HDL code and testbench suite. A preliminary workflow was created for synthesis and place-and-route. Verification consisted of running logic simulations on the RTL HDL code and cell netlist after synthesis but before place-and-route. No verification nor timing analysis took place after place-and-route. The resulting ASIC failed testing after discovering that the place-and-route tool shorted signals of similar naming. Figure 74 shows the blocks in the layout, and Figure 75 shows a micrograph the RDC.

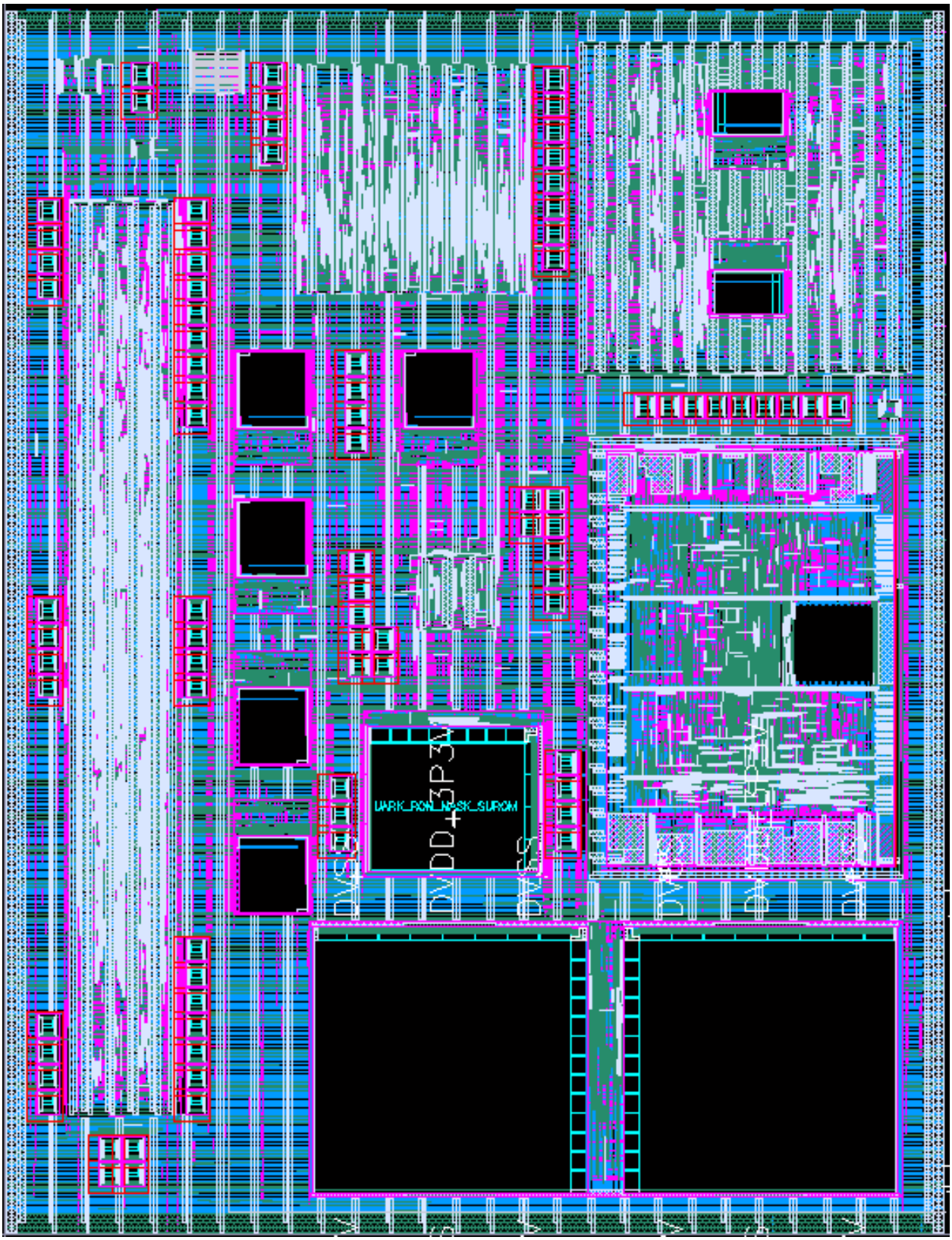

Figure 74: Layout of the CRYO5 RDC1 ASIC.

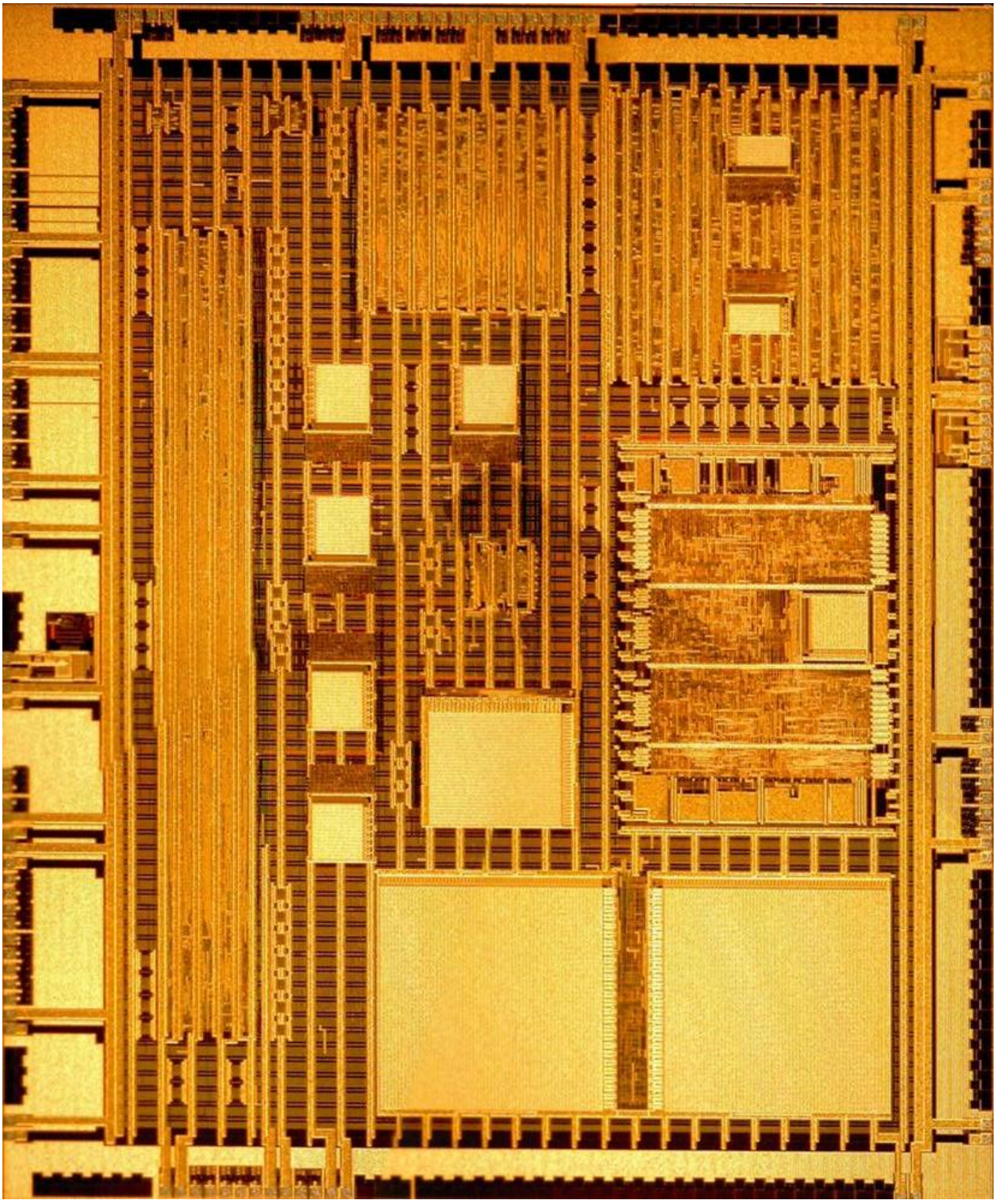

Figure 75: Micrograph of the CRYO5 RDC1 ASIC.

3.5.2 CRYO5A RDC2

Figure 76 and Figure 76 show the RDC2 layout and micrograph respectively. The lack of verification after place-and-route in the previous release was addressed. Only functional verification was setup to simulate the optimized cell netlist after place-and-route. In addition, preliminary static timing analysis was setup to find critical paths. However, STA was based solely on wire-load models and therefore not representative of the ASIC layout. Another issue was the lack of coherency between the memory array HDL models and the memory arrays they represent. During ASIC testing, the microcontroller and its boundary scan interface functioned and allowed for external memory access. The SRAMs and ROM behaved against expectations by constantly reading out the same byte value across all addresses. Further testing took thermal images of the ASIC on standby power, one of which is shown in Figure 78, and the SRAMs and ROM showed the greatest degree of heat. Investigation of the memory array schematics and layouts revealed discrepancies in the SRAM and ROM design.

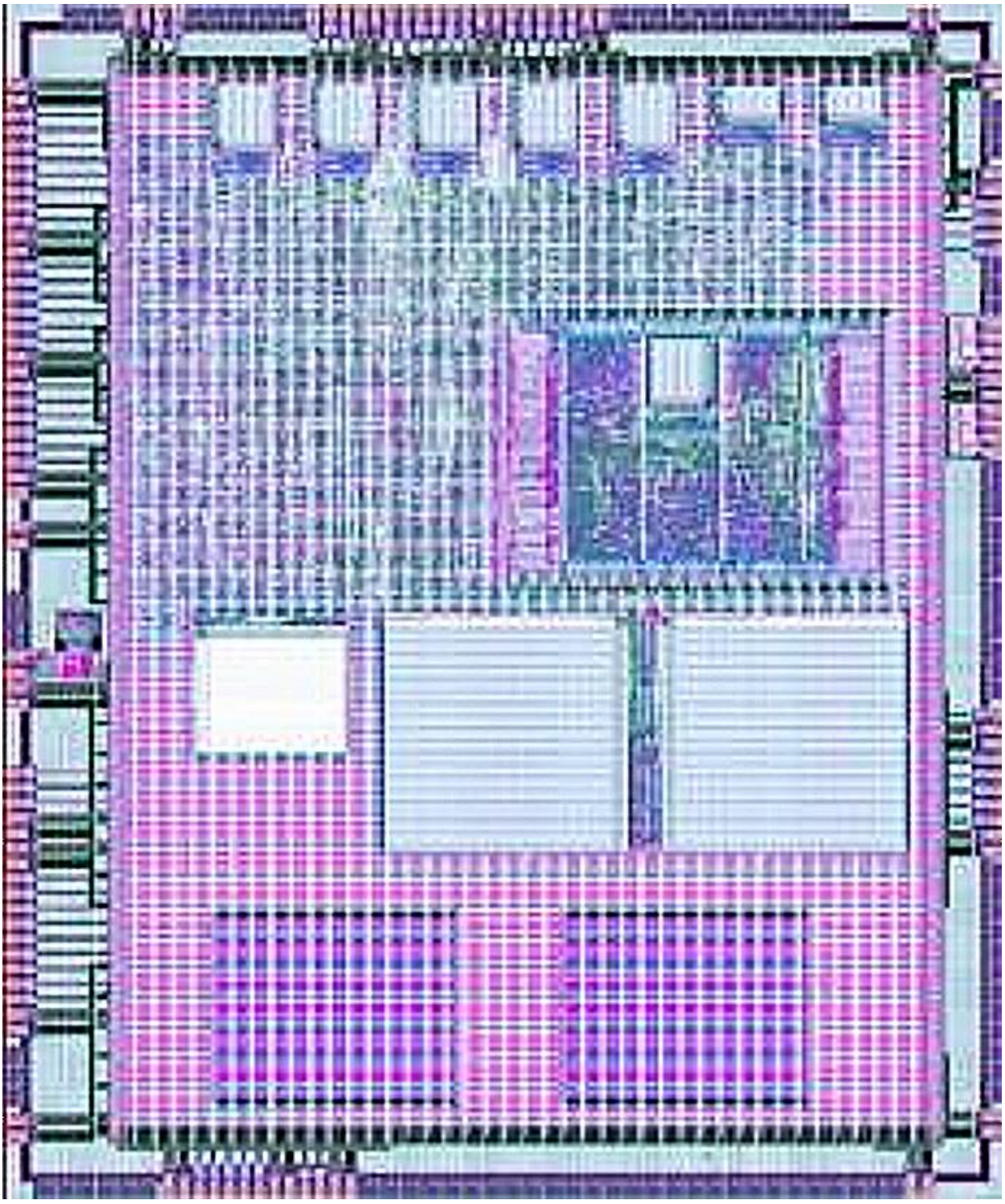

Figure 76: Layout of the CRYO5A RDC2 ASIC.

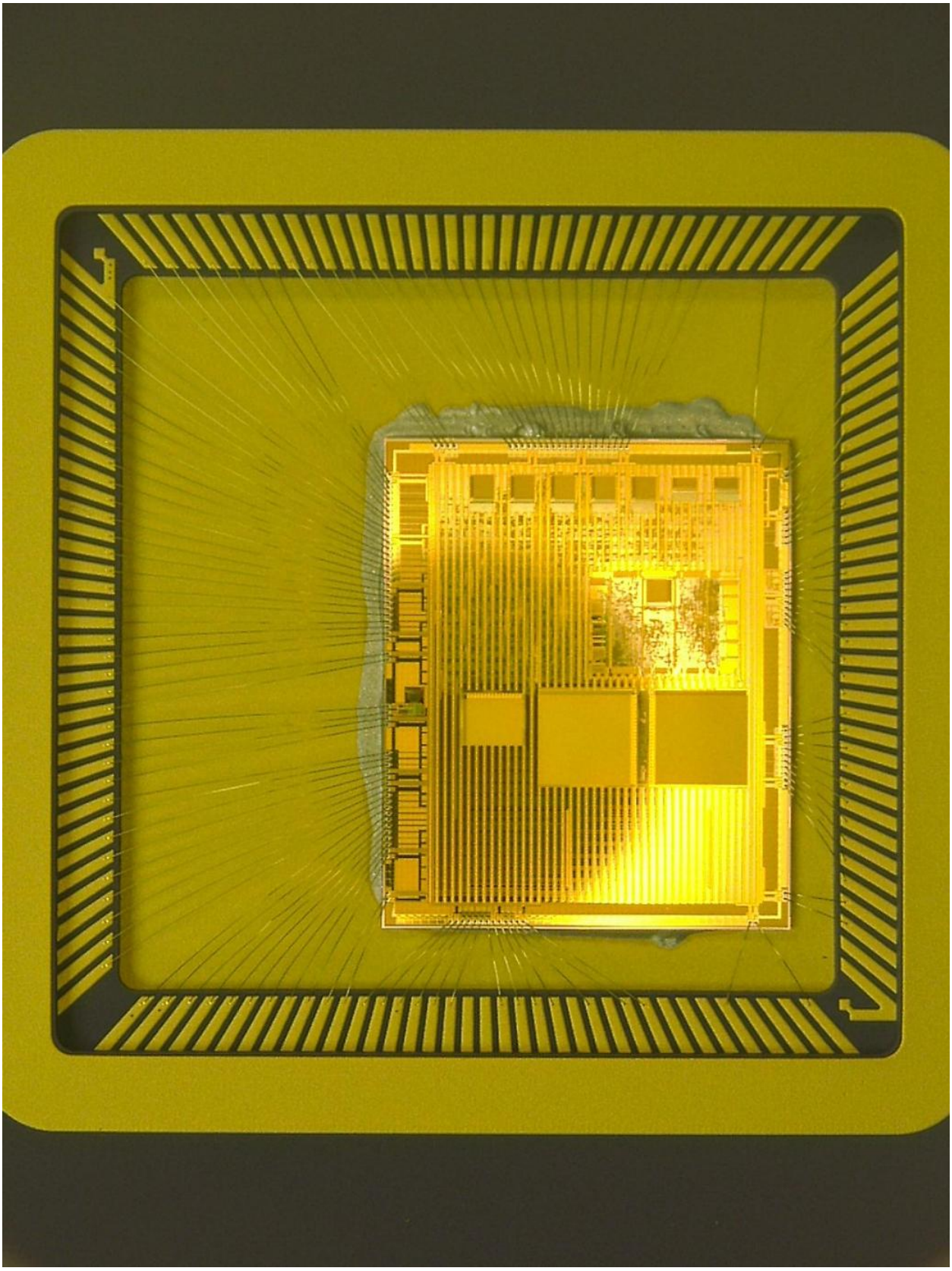

Figure 77: Micrograph of the RDC2 ASIC.

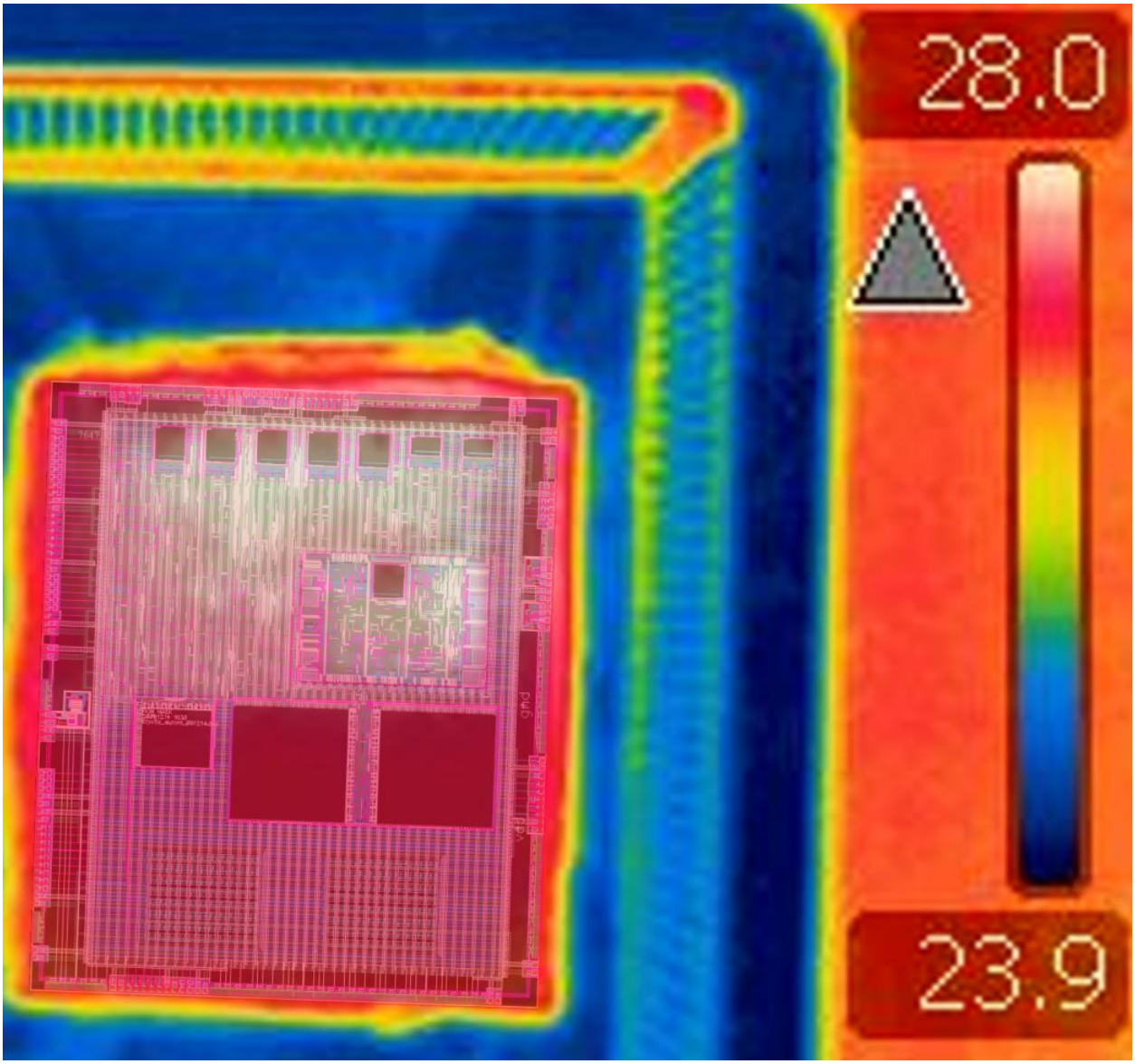

Figure 78: Thermal image of the CRYO5A RDC2 ASIC on standby power of 3.3 volts.

3.5.3 RDC3

Figure 79 shows the RDC3 layout. In light of issues with the memory arrays from the second ASIC release, further verification is setup to check for coherency between the HDL models and the devices they represent. Additional verification involved running Spectre simulations on each memory array design using input and output vectors generated from timing logic simulations, via piece-wise linear (PWL) voltage sources. Also, the memory array devices that supported error detection were simplified by moving the error detection logic outside the memory array device and into the surrounding logic. The 8031 microcontroller was another complex device that was simplified by removing its tri-state interface, which did not fit well with the RDC's multiplexed system bus. STA was enhanced with wire parasitics defined by SPEF and generated from place-and-route. Consequently, STA provided representative cell and wire delay values to timing logic simulations. With the more extensive timing analysis, extra steps were added to the design workflow to ensure operability across the wide range of temperatures. Those extra steps were buffer insertion during STA and optimization during place-and-route. After release, ASIC testing revealed an issue in the design where address and data buses are reversed at each memory device. Two causes of this issue are that the Verilog language connects buses without specifying whether the bit index order ascends or descends and that the memory array Liberty code specified ascending bit indexes while logic simulations assume descending bit indexes. At the time of this writing, testing continues for the third ASIC release.

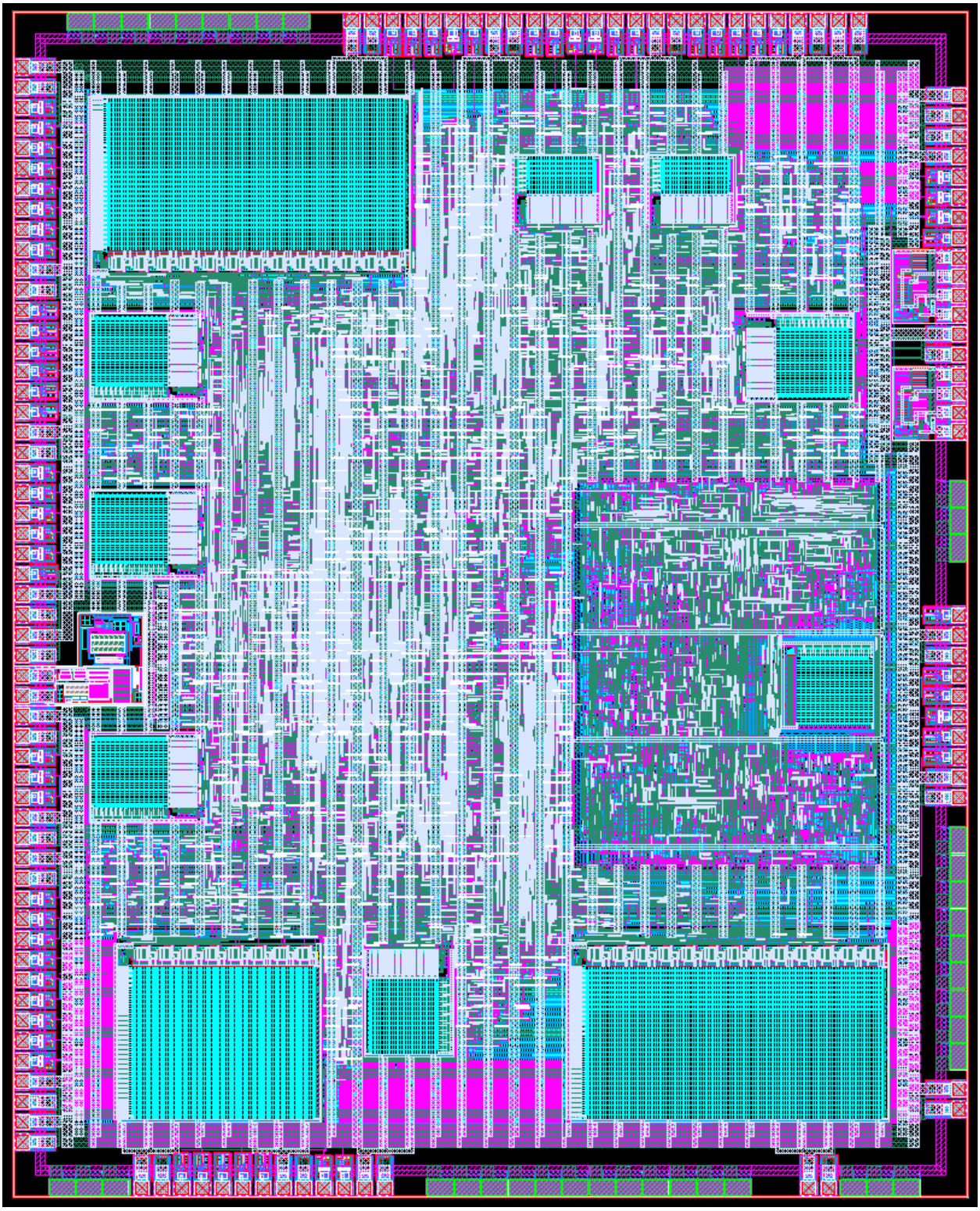

Figure 79: Layout of the RDC3 ASIC.

4 CONCLUSION

The work done for ultra-low voltage circuit design involved FD-SOI transistor characterization, analyzing transistor and gate behavior at ultra-low supply voltages, and the study and application of NCL and Schmitt-trigger gate design. The FD-SOI transistor arrays were fabricated and tested at MIT Lincoln Laboratory using a probe station with software to measure I-V curves. Each FD-SOI transistor in the array connects through four pads for the gate, source, drain, and an optional body connection, and the probe station's chuck provides back wafer voltage to the FD-SOI die. The measured I-V curves were exported to Silvaco UTMOST, a curve-fitting program that references BSIM and BSIMSOI transistor models. Steps were taken to tune each BSIMSOI model parameter while matching each I-V curve on both the log and linear scales, ultimately generating FD-SOI Spectre transistor models for PMOS and NMOS. The transistor models were tested in a threshold gate to verify behavior. Further work required for the FD-SOI models is incorporating statistical parameters in the BSIMSOI models. Multiple FD-SOI dies were measured but not translated to Spectre.

Regarding Schmitt-trigger gate design, the NCL threshold gates exhibited behavior that is different than the synchronous gates, specifically the degrading hysteresis effect. It should be noted that there is a similarity between the topologies of a threshold gate and a Schmitt-trigger gate. Both gate designs share an output feedback that is driven by an inverter. Additional research may reveal a gate design method that exploits the similarity between threshold and Schmitt-trigger gates. Results founded in this research show a possible use for Schmitt-trigger gates as well as limitations with two alternative Schmitt-trigger architectures.

The RDC project involved constructing an embedded system using an asynchronous microprocessor. Tasks performed in this research are synchronous gate library development and characterization, HDL and SDCC programming, synthesis, preliminary place-and-routing, logic simulations with timing data. A contribution made is proving operability of NCL on a silicon-germanium process, which is tailored for extreme environments.

The research work in ultra-low voltage design and the RDC share a similarity in extreme operations that require unconventional design methodologies. Ultra-low voltage operation requires consideration in designing the gate libraries and digital circuit (e.g., Schmitt-trigger gates and NCL), and extreme temperature operation needs attention in cell schematic and layout design (e.g., double the NMOS channel length and add guard rings). When designing large-scale digital circuits for extreme operations, the greatest issue is predicting circuit behavior at the transistor level due to the lack of SPICE simulator performance and that device models are typically characterized under nominal conditions. Moreover, some transistor models exclude statistical data required to simulate process variation, which is a problem that must be taken into account under extreme conditions. Predicting large-scale digital circuit behavior under extreme conditions requires theoretical analysis, specialized device and cell modeling, and logical verification.

5 REFERENCES

[1] O.A. Amusan, L.W. Massengill, M.P. Baze, A.L. Sternberg, A.F. Witulski, B.L. Bhuva, and J.D. Black. Single event upsets in deep-submicrometer technologies due to charge sharing. *Device and Materials Reliability, IEEE Transactions on*, 8(3):582 – 589, September 2008.

[2] Cadence. *Integrated Circuit (Virtuoso) 5.1.41 Product Documentations*, December 2011.

[3] A. Chandrakasan, W. Bowhill, and F. Fox. Techniques for leakage power reduction. In *Design of High-Performance Microprocessor Circuits*, pages 46 – 62, 2001.

[4] Z. Chen, M. Johnson, W. Liqiong, and W. Roy. Estimation of standby leakage power in cmos circuit considering accurate modeling of transistor stacks. In *Low Power Electronics and Design, International Symposium on*, pages 239 – 244, August 1998.

[5] K.J Cornett, G. Fu, I. Escorcia, and H.A. Mantooth. SiGe BiCMOS fully differential amplifier for extreme temperature range applications. In *Aerospace conference, IEEE*, pages 1 – 10, March 2009.

[6] IBM Microelectronics Division. *SiGeHP (BiCMOS) Design Manual*, March 2004.

[7] Sandeep Dutta, Jean Loius, Daniel Drotos, Kevin Vigor, Johan Knol, Scott Dattalo, Karl Bongers, Bernhard Held, Frieder Ferlemann, Jesus Calvino-Fraga, Borut Ražem, Vangelis Rokas, Erik Petrich, Dave Helton, Paul Stoffregen, Michael Hope, Maarten Brock, Raphael Neider, Philipp Klaus, and Leland Morrison. Small device C compiler. http://opencores.org/project,can, February 2003.

[8] M. Hartwell, K. Ryan, S. Netherton, P. Milliken, and D. Kerwin. Total ionizing dose testing of a RadHard-by-Design FET driver in a 0.35 um triple-well process. In *Radiation Effects Data Workshop, 2006 IEEE*, pages 94 –100, July 2006.

[9] J.R. Hoff, R. Arora, J.D. Cressler, G.W. Deptuch, P. Gui, N.E. Lourenco, G. Wu, and R.J. Yarema. Lifetime studies of 130nm nMOS transistors intended for long-duration, cryogenic high-energy physics experiments. In *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), IEEE*, pages 685 – 693, October 2011.

[10] B. Hollosi, M. Barlow, G. Fu, C. Lee, J. Di, S.C. Smith, H.A. Mantooth, and M. Schupbach. Delay-insensitive asynchronous ALU for cryogenic temperature environments. In *Circuits and Systems, MWSCAS 51st Midwest Symposium on*, pages 322 – 325, August 2008.

[11] R. Jaramillo-Ramirez, J. Jaffari, and M. Anis. Variability-aware design of subthreshold devices. In *Circuits and Systems, IEEE International Symposium on*, pages 1196 – 1199, May 2008.

[12] K. Joardar. Signal isolation in BiCMOS mixed mode integrated circuits. In *Bipolar/BiCMOS Circuits and Technology Meeting, Proceedings of*, pages 178 – 181, October 1995.

[13] A.S Kashyap, A. Mantooth, T.A Vo, and M. Mojarradi. Compact modeling of LDMOS transistors for extreme environment analog circuit design. *Electron Devices, IEEE Transactions on*, 57(6):1431 – 1439, June 2010.

[14] C. Kenyon, A. Kornfeld, K. Kuhn, M. Liu, A. Maheshwari, W. Shih, S. Sivakumar, G. Taylor, P. VanDerVoorn, and K. Zawadzki. Managing process variation in Intel's 45nm CMOS technology. *Intel Technology Journal*, 12, June 2008.

[15] R. Kumar and V. Kursun. Temperature-adaptive energy reduction for ultra-low power-supply-voltage subthreshold logic circuits. In *Electronics, Circuits and Systems, 2007. ICECS 2007. 14th IEEE International Conference on*, pages 1280 – 1283, December 2007.

[16] Ranjith Kumar and Volkan Kursun. A design methodology for temperature variation insensitive low power circuits. In *GLSVLSI '06: Proceedings of the 16th ACM Great Lakes symposium on VLSI*, pages 410 – 415, New York, NY, USA, 2006. ACM.

[17] J. Kwong and A.P. Chandrakasan. Variation-driven device sizing for minimum energy subthreshold circuits. In *Low Power Electronics and Design, International Symposium on*, pages 8 – 13, October 2006.

[18] J. Lohstroh, E. Seevinck, and J. deGroot. Worst-case static noise margin criteria for logic circuits and their mathematical equivalence. *Solid-State Circuits, IEEE Journal of*, 18(6):803 – 807, December 1983.

[19] N. Lotze and Y. Manoli. A 62mV 0.13m CMOS standard-cell-based design technique using Schmitt-trigger logic. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pages 340 – 342, February 2011.

[20] N. Lotze and Y. Manoli. A 62 mv 0.13 μm cmos standard-cell-based design technique using schmitt-trigger logic. *Solid-State Circuits, IEEE Journal of*, 47(1):47 – 60, January 2012.

[21] Igor Mohor. CAN protocol controller. http://opencores.org/project,can, February 2003.

[22] L. Najafizadeh, C. Zhu, R. Krithivasan, J.D. Cressler, Y. Cui, G. Niu, S. Chen, C. Ulaganathan, B.J. Blalock, and A.J. Joseph. SiGe BiCMOS precision voltage references for extreme temperature range electronics. In *Bipolar/BiCMOS Circuits and Technology Meeting*, pages 1 – 4, 2006.

[23] M.J.M. Pelgrom, A.C.J. Duinmaijer, and A.P.G Welbers. Matching properties of MOS transistors. *Solid-State Circuits, IEEE Journal of*, 24(5):1433 – 1439, October 1989.

[24] P.J. Restle, T.G. McNamara, D.A. Webber, P.J. Camporese, K.F. Eng, K.A. Jenkins, D.H. Allen, M.J. Rohn, M.P. Quaranta, D.W. Boerstler, C.J. Alpert, C.A. Carter, R.N. Bailey, J.G. Petrovic, B.L. Krauter, and B.D. McCredie. A clock distribution network for microprocessors. In *VLSI Circuits, 2000. Digest of Technical Papers. 2000 Symposium on*, pages 184 – 187, 2000.

[25] Takayasu Sakura, Akira Matsuzawa, and Takakuni Douseki. *Fully-Depleted SOI CMOS Circuits and Technology for Ultralow-Power Applications*. Springer, first edition, April 2006.

[26] E. Seevinck, F.J. List, and J. Lohstroh. Static-noise margin analysis of MOS SRAM cells. *Solid-State Circuits, IEEE Journal of*, 22(5):748 – 754, October 1987.

[27] Scott Smith and Jia Di. *Designing Asynchronous Circuits using NULL Convention Logic (NCL)*. Morgan & Claypool, 2009.

[28] Synopsys. *Liberty User Guide*, December 2003.

[29] Synopsys. *DFT Compiler User Guide: Scan*, December 2008.

[30] Synopsys. *Liberty NCX User Guide*, December 2008.

[31] BAE Systems. *Remote Electronics Unit Digital Control (RDC) ASIC Functional Requirements Specification*, June 2010.

[32] Jeng-Liang Tsai and C. Chung-Ping Chen. Process-variation robust and low-power zero-skew buffered clock-tree synthesis using projected scan-line sampling. In *Design Automation Conference, 2005. Proceedings of the ASP-DAC 2005. Asia and South Pacific*, volume 2, pages 1168 – 1171, January 2005.

[33] G. Venkataraman, Zhuo Feng, Jiang Hu, and Peng Li. Combinatorial algorithms for fast clock mesh optimization. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 18(1):131 – 141, January 2010.

[34] S.H. Voldman and A. Watson. The influence of deep trench and substrate resistance on the latchup robustness in a BiCMOS silicon germanium technology. In *Reliability Physics Symposium Proceedings, IEEE International*, pages 135 – 142, April 2004.

[35] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester. Analysis and mitigation of variability in subthreshold design. In *Low Power Electronics and Design, Proceedings of the International Symposium on*, pages 20 – 25, August 2005.