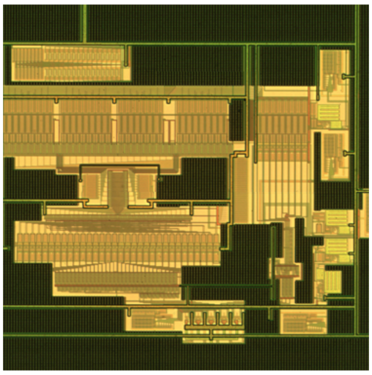
UNIVERSIDADE DO ALGARVE

INSTITUTO SUPERIOR DE ENGENHARIA



AUTOMATIC ANALYSIS OF SUBTHRESHOLD OPERATION IN CMOS DIGITAL CIRCUITS

ANÁLISE AUTOMÁTICA DA OPERAÇÃO A TENSÕES SUB-LIMIARES EM CIRCUITOS DIGITAIS CMOS

Hugo Alexandre Nunes Cavalaria

Dissertation for obtaining the Master degree in

Electrical and Electronic Engineering

Specialization Area in Information Technologies and Telecommunications

Tutor: Professor Doutor Jorge Filipe Leal Costa Semião

September, 2017

Title: Automatic Analysis of Subthreshold Operation in CMOS Digital Circuits.

Authorship: Hugo Alexandre Nunes Cavalaria.
I hereby declare to be the author of this original and unique work. Authors and references in use are properly cited in the text and are all listed in the reference section.

Título: Automatic Analysis of Subthreshold Operation in CMOS Digital Circuits.

Autoria: Hugo Alexandre Nunes Cavalaria.
Eu declaro ser o autor deste trabalho original e único. Os autores e as referências utilizadas são devidamente citados no texto e estão todas listadas na seção de bibliografia.

Hugo Alexandre Nunes Cavalaria

Hugo Alexandre Nunes Cavalaria

To my family

ACKNOWLEDGMENTS

First of all, I want to express my most sincere gratitude to Dr Jorge Semião, my tutor, for his great dedication, both personal as professional.

I want to express also sincere gratitude for all his support and for his superior guidance, continuously pushing me forward, giving me the strength and confidence necessary to keep developing this research work, to pursue and obtain conclusions and effectively achieve my goals.

To complete successfully a task of this complexity would be impossible without his guidance, Dr. Jorge Semião has wide range of knowledge as well as great expertise on this research field, expertise that he enthusiastically shares with sympathy. I want to thank him for his time, his attention and help and specially for introducing me to a new and interesting research field.

I have to express my sincere gratitude to all my family, specially to my father, my mother and my brother for all their help and support during the development of this research work. I also want to thank my family for believing in me, for their strength and their continuous encouragement to pursue and achieve this goal, as they always did during all of my academic and professional career.

Finally I would like to express gratitude to my friends for all the good times we shared, giving me the happiness and confidence necessary to complete this thesis.

Hugo Alexandre Nunes Cavalaria, Faro, September, 29, 2017

ABSTRACT

The Internet of Things (IoT) paradigm is enabling easy access and interaction with a wide variety of devices, some of them self-powered, equipped with microcontrollers, sensors and sensor networks. Low power and ultra-low-power strategies, as never before, have a huge importance in today's CMOS integrated circuits, as all portable devices quest for the never-ending battery life, but also with smaller and smaller dimensions every day.

The solution is to use clever power management strategies and reduce drastically power consumption in IoT chips. Dynamic Voltage and Frequency Scaling techniques can be rewardingly, and using operation at subthreshold power-supply voltages can effectively achieve significant power savings. However, reducing power-supply voltages impose reduction of performance and, consequently, delay increase, in turn it makes the circuit more vulnerable to operational-induced delay-faults and transient-faults. What is the best compromise between power, delay and performance?

This thesis proposes an automatic methodology and tool to perform power-delay analysis in CMOS gates and circuits, to identify automatically the best compromise between power and delay. By instantiating SPICE simulator, the proposed tool can automatically perform analysis such as: power-delay product, energy-delay product, power dissipation, or even dynamic and static power dissipations.

The optimum operation point in respect to the power-supply voltage is defined, for each circuit or sub-circuit and considering subthreshold operation or not, to the minimum power-supply voltage where the delays do not increase too much and that implements a compromise between delay and power consumption.

The algorithm is presented, along with CMOS circuit examples, all the analysis' results are shown for typical benchmark circuits. Results indicate that subthreshold voltages can be a good compromise in reducing power and increasing delays.

KEYWORDS: Subthreshold voltage operation; Power-Delay Product; Dynamic Voltage and Frequency Scaling; Power and Frequency Optimization.

RESUMO

O aparecimento e a expansão de novas tendências da indústria electrónica fortemente direccionadas ao paradigma da *Internet of Things* (IoT) têm vindo a dar uma relevância cada vez maior à necessidade da evolução da electrónica, no sentido da interligação e intercomunicação entre equipamentos, no sentido da miniaturização em geral e, consequentemente, no sentido de uma melhor eficiência energética.

Temos assim, na prática, vindo recentemente a assistir em diversas áreas ao surgimento progressivo de um número exponencial de pequenos dispositivos electrónicos, altamente compactos, com elevado grau de integração de funções e habitualmente interligados entre si em redes de dados. Habitualmente têm como missão genérica a recolha, processamento e transmissão de dados acerca do ambiente que os rodeia. Esta grande variedade de diferentes dispositivos habitualmente relacionados ao campo de IoT tem como principais funções a recolha e transdução de dados obtidos do ambiente circundante por sensores. Tem por isso geralmente uma muito limitada interação com o ambiente circundante, e nesse sentido, justifica-se que as suas principais características sejam as pequenas dimensões e fácil portabilidade. Justifica-se também que não é estritamente essencial que tenham elevada performance a nível de processamento. Sendo alimentados por baterias, ou nalguns casos alimentados por energia do ambiente, estes dispositivos precisam obrigatoriamente de consumir muito pouca energia, sendo os seus requisitos de energia de alimentação muito restritos. Dados os restritos requisitos de consumo energético, são tipos de circuitos muito adequados à aplicação das mais recentes e avançadas estratégias de gestão de potência destinadas a reduzir drasticamente a potência nos modernos circuitos integrados CMOs.

Torna-se assim claro, que os mais importantes requisitos futuros de dispositivos na área de IoT, assim como de diversas famílias de dispositivos electrónicos em geral, serão tendencialmente a necessidade de redução de consumo energético, ainda que esta redução seja feita à custa de algum nível de redução em performance. Esta tendência baseia-se no crescimento de importância da temática da eficiência energética em circuitos, num momento em que a concentração de consumo energético e

consequentemente de dissipação térmica, em áreas muito reduzidas de circuitos integrados CMOs atinge níveis muito elevados e preocupantes.

Uma possível solução para enfrentar este complexo desafio, com crescentes requisitos e restrições para actuais e futuros circuitos CMOs, tendo em atenção princípios globais de eficiência energética, consiste em conjugar as habituais técnicas de gestão de potência dinâmica em circuitos, com as mais recentes e avançadas técnicas de alimentação em '*ultra-low-power voltage*', tentando alcançar assim ganhos de potência muito consideráveis e significativos. Assim, associando as conhecidas técnicas de gestão de potência como por exemplo a *Dynamic Voltage and Frequency Scaling* (DVFS) com as mais recentes técnicas de *ultra-low-power voltage* como a recente técnica de operação em tensões de alimentação *subthreshold* pode potencialmente se revelar como a melhor solução para enfrentar este complexo problema e assim melhorar significativamente a eficiência energética em futuros circuitos CMOS.

Contudo, quando aplicamos técnicas de potência de *very-low-power* ou *ultra-low-power*, como as técnicas de operação a tensões *subthreshold*, existem algumas desvantagens e alguns efeitos adversos que devem ser cuidadosamente considerados e, se possível, contidos e minimizados. A mais importante destas consequências directas é a perda de performance do circuito que deriva naturalmente do aumento nos atrasos de propagação internos do circuito. As restantes desvantagens da utilização de técnicas de alimentação a níveis muito baixos derivam todas elas do facto do circuito se tornar em geral muito mais sensível a perturbações internas ou externas. Esta é claramente uma consequência natural para uma operação a este nível de reduzida energia.

Como seria de esperar, pelo exposto, a operação a níveis de tensão *ultra-low-voltage* têm a consequência de torná-lo mais sensível a distúrbios e interferências, aumentado assim o risco de falhas operacionais, dado que o nível dos seus sinais internos de operação ao longo do circuito é muito reduzido. Alguns efeitos adversos afectos ao uso de técnicas de *ultra-low-power* em circuitos CMOs incluem, portanto, o aumento da vulnerabilidade do circuito a *Single Event Upsets* (SEUs), incluem também o aumento de vulnerabilidade a falhas induzidas de *delay* de operação, assim como um aumento de sensibilidade do circuito a falhas geradas por transientes.

Tendo consciência do incremento de riscos operacionais envolvido em circuitos *subthreshold*, são necessários cuidados no sentido de conter e minimizar tanto quanto possíveis efeitos indesejados, por exemplo controlando cuidadosamente as condições operacionais do circuito e melhorando a sua blindagem a interferências.

Considerando que o uso das técnicas de *ultra-low-power* pode ser provavelmente a melhor solução para cumprir rigorosos requisitos de eficiência energética para um circuito CMOs, é necessário considerar também que estas técnicas podem gerar uma considerável perda de performance, traduzida por um maior atraso interno. Assim, torna-se necessário estudar claramente, em *subthreshold voltages,* a evolução da perda de performance face aos grandes ganhos de energia quando caminhamos no sentido da redução da tensão de alimentação de um circuito CMO's.

Tendo como base um estudo custo/benefício da evolução de dois factores cruciais na operação de um circuito, como o factor energia e o factor performance, torna-se possível tentar alcançar uma solução de compromisso entre a potência dissipada (energia consumida) e o atraso de propagação, traduzido como a performance do circuito.

O trabalho aqui apresentado propõe uma metodologia automatizada, capaz de enfrentar os desafios do estudo mencionado. Propõe ainda uma ferramenta de *software* desenhada para analisar em detalhe portas lógicas CMOs de uma livraria de portas existente, assim como circuitos completos composto por diversas portas lógicas. O *software* proposto analisa um circuito ou sub-circuito lógico, identificando automaticamente o melhor nível de alimentação de baixa tensão (ponto de operação óptimo) que permite obter o melhor compromisso entre potência e atraso, em termos gerais o melhor compromisso entre energia e performance. Como suporte e assistência à metodologia proposta esta ferramenta foi criada para acelerar os testes de simulação Hspice sobre portas lógicas e circuitos, executando cálculos rápidos sobre resultados de simulação e acelerando a obtenção de resultados de eficiência energética e de performance para análise.

Através da instanciação directa do simulador Hspice, a ferramenta facilita a análise de importantes parâmetros de definição de portas lógicas e circuitos, como por

exemplo: o atraso de propagação, o *power-delay-product* (PDP), o *energy-delay-product* (EDP), e a dissipação de potência total e parcial (estática e dinâmica).

O desenvolvimento inicial da ferramenta permitiu realizar múltiplos testes e simulações e através da análise destes resultados desenvolver a metodologia *low-power* apresentada no trabalho, a posterior aplicação da metodologia pela ferramenta a um circuito CMO's permite eficientemente identificar o seu ponto de operação óptimo para operação em baixo nível.

Um ponto de operação óptimo de uma porta lógica é definido pelo método como o mais baixo nível de tensão de alimentação que não compromete a operação válida da porta, reduzindo por isso fortemente a potência dissipada. No entanto este ponto deve ainda minimizar (tanto quanto possível) os atrasos de propagação na porta.

Assim, este ponto deriva de um compromisso ponderado para uma alimentação com consumo de energia muito baixo, que contudo não gere ainda atrasos na porta que provoquem significativas perdas em performance. Acima de tudo, o trabalho desenvolvido pretende apresentar uma abordagem clara e directa ao *design* e implementação de lógica digital em modo de *subthreshold*, aplicado ao contexto dos modernos circuitos de electrónica digital.

Pretende-se estabelecer um conjunto de técnicas e métodos simples e claros, suportados num estudo incidente em regras teóricas e em simulações prácticas, que possam servir como normativos propostos para o design de circuitos adaptados ao funcionamento em modos de muito baixa energia.

O objectivo final será enfrentar e a longo prazo tentar resolver o problema cada vez maior e mais importante da melhoria de eficiência energética em circuitos electrónicos genéricos.

PALAVRAS-CHAVE: Operação com tensão de alimentação sub-limiar; Produto Potência-Atraso; Escalonamento Dinâmico da Tensão e da Frequência; Optimização da Potência e da Frequência.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AEP-FF	Adaptive Error-Prediction Flip-Flop
ASIC	Application Specific Integrated Circuit
CMOS	Complementary Metal-Oxide Semiconductor
CP	Critical Path.
FF	Flip-flop
FPGA	Field Programmable Gate Array
H	Hidrogen (chemical symbol)
HCA	Hot Carrier Aging
HCI	Hot Carrier Injection
HSPICE	Synopsys Spice Simulation Program
IC	Integrated Circuit
IoT	Internet of Things
LTD	Long Term Degradation
MERIT	cMos EneRgy sImulaTor
NBTI	Negative Bias Temperature Instability
NMOS	N-type Metal-Oxide Semiconductor.
NMOSFET	N-type Metal Oxide Semiconductor Field-Effect Transistors
PBTI	Positive Bias Temperature Instability
PDF	Path Delay Faults
PI	Primary Input
PMOS	P-type Metal-Oxide Semiconductor
PMOSFET	P-type Metal Oxide Semiconductor Field-Effect Transistors
PO	Primary Output
PTM	Predictive Technology Model
PVT	Process, power-supply Voltage and Temperature
PVTA	Process, power-supply Voltage, Temperature and Aging
Si	Silícon (chemical symbol)
SiO2	Silicon Dioxide (chemical substance)

SO	Secondary Output
SPICE	Simulation Program with Integrated Circuit Emphasis
T	Temperature
TDDB	Time Dependent Dielectric Breakdown
TT	Truth Table
V	Power-supply Voltage
VDD	Positive Voltage Supply
VHDL	Very High (Speed IC) Hardware Description Language
VSS	Negative Voltage Supply
VT	Power-supply Voltage and Temperature
Vth	Threshold Voltage

1. INTRODUCTION

With the recent advent of the Internet of Things (IoT) paradigm nowadays, we witness an exponential growing number of small highly integrated and highly interconnected devices, usually designed to collect, process and transmit environmental data and information, to several types of data networks. This wide variety of different electronic devices related to IoT are usually mainly focused on small size and portability and less constrained by demanding performance requirements, as their main work objective is sensor data gathering, or eventually a very limited surround ambient interaction. Therefore, being self-powered or battery operated, these devices need to have very low energy requirements, and by these clear constraints, they are very suitable and even required to use efficient power management strategies to re-duce its power consumption.

Moreover, this increasing use of battery operated devices, increase the demand of aggressive power-reduction techniques. The never-ending battery life is a new trend, and, as Moore's law pushes technology to the size limit, in the same way companies push this new trend forward by creating batteries with higher density and capacity, but also by decreasing power consumption in today's SoC (Silicon-on-Chips).

1.1. LOW-POWER DESIGN CHALLENGES IN IOT

LP design techniques at lower levels of abstraction have to carefully manage, not only power supply voltages, but also clock frequencies, in synchronous circuits. Usually, sophisticated low-level power management techniques require an optimized design of two basic layout infrastructures, namely, the power grid, and the clock distribution network (CDN). Power savings may dictate that on-chip power gating, i.e., VDD shutdown (and power-up) functionality, is installed. Multi-value static VDD do-mains may be defined. However, more aggressive techniques may apply DVS (Dynamic Voltage

Scaling), or even DVFS (Dynamic Voltage and Frequency Scaling) [26][27][28]. This means that, according to system operation, local modules may be powered by variable VDD, which is also referred as Adaptive Power Management [29]. Power-saving schemes for DVFS have been presented in the past [24][25][26][27]. In fact, there are already solutions used widely in industry, like *SpeedStep* from Intel [24] or *PowerNow*!, from AMD [25], and some promising solutions that are not yet implemented in commercial products, like [26][27].

In energy savings techniques, the bottom line is to use power only when needed. In this way, aggressive techniques like power-gating are gaining ways in several application levels of abstraction. Not only at transistor level, where the classical power-gating is used, but also at system-level, where programmable RTC (Real-Time-Clocks) are being widely used to power-off the circuit to the minimum operation, in order to re-duce power consumption when it is not needed. Then, periodically and sparsely in time the RTC will wake-up the system and allow the minimum operation time to perform the required operation, so it can put the system back to sleep again.

The future requirements of IoT devices, as well as general electronic devices, tend strongly towards the reduction of power consumption even at the expense of some performance reduction as the energy efficiency paradigm becomes increasingly important. IoT (Internet-of-Things) applications require the use of nearly-zero power consumption sensors, to be available gathering data from everywhere using nearly-zero energy, and sparsely can transmit these data to the web. These never-ending battery achievements can only be done with more than one technique, and working at different VDD levels.

But DVFS techniques can even be pushed forward, by working at subthreshold voltages in the power-supply (and of course, also by relaxing even more the clock frequency). Subthreshold operation is not new, and literature has several previous works dealing with these technique [4][5][6][7][9][12][13][15][16]. However, the problem of working at subthreshold voltage is that it puts the circuits in a very vulnerable situation, due to the limited energy available in the circuit, considerably reducing performance (due to the increased circuit delays), and increasing sensibility to several different external and internal factors. Errors are prone to happen due to: operation induced delay-faults, transient faults, electromagnetic interference (EMI), temperature variations, process variations, power-supply variations, single-event upsets (SEU), radiation, etc… Fortunately, aging effects like BTI (Bias Temperature Instability) are reduced, due to the lowered voltages applied to transistors' gates.

Being aware of the increase on operational risks for subthreshold circuits, care must be taken to carefully contain and minimize these increased negative effects, for example closely controlling the operational conditions of the circuit itself as well as improving shielding around the circuit. Nevertheless, it is important to state that, for IoT, power is probably the most critical factor and there are previous works that can help on avoiding errors, like [21][22][23][30]. Moreover, typical IoT sensor applications are not critical, i.e., they do not deal with human lives, and some errors may be acceptable and even corrected at higher abstraction levels (for example, by statistically identifying outliers in a data set collected from IoT sensors). Therefore, it is important to identify a compromise where power is drastically reduced, but most errors are still avoided or prevented.

1.2. MOTIVATION AND OBJECTIVE

The motivation of this work is to build a step forward on this never-ending battery life devices, with nearly-zero power consumption from the battery. The DVFS methodology presented in [23] was already updated in [30] to include a control mechanism to allow dynamically tuning the best power-supply voltage and operating frequency, in order to maximize power savings or performance, respectively. In addition, it is now possible to use multiple operation modes, allowing a sleep/low-power mode, and a high-performance/power mode. In the sleep/low-power mode, VDD is optimized and reduced, reducing power to the minimum (e.g., when the IoT sensor gathers data from the environment); in the high-performance/power mode, performance is improved, and the work is executed in the less time possible (e.g., when the IoT sensor sends data to the web). The future perspective of this work is to reuse this multi-mode DVFS technique from [23] and [30], with predictive error detection and error tolerance features, in ultra-low-power VDD ranges, but subthreshold voltage operation must be analyzed first.

In this work, the main objective is to develop a new methodology to analyze automatically circuits at subthreshold operation and define the best VDD voltage to maximize power savings, but also working with an acceptable performance (clock frequency) that does not jeopardize circuit operation. A new software is needed, to support and implement the new methodology automatically. Therefore, the MERIT (cMos EneRgy sImulaTor) software tool development is also a main objective, to automatically determine for a circuit, and for each individual gate, the minimum VDD for a correct operation, and the optimal VDD, obtained from a compromise between power and delay operation. As power decreases when VDD is reduced (and propagation delays increase), the optimal VDD should be obtained by determining the minimum VDD for which the propagation delays are not drastically increased. The MERIT tool, based on low-level SPICE simulations, should deliver the best compromise VDD result between power and delay (energy and performance) for a circuit, guaranteeing that, with this optimal VDD, power can be drastically reduced but performance is not drastically affected.

1.3. THESIS OUTLINE

This thesis is organized as follows.

In chapter 2, a general review concerning the improvement of energy efficiency applied to modern digital systems is presented. The low-power techniques and methods for dynamic and adaptive voltage and frequency scaling are generally explained. Moreover, regarding some techniques and methods for ultra-low power circuit operation, an overview on subthreshold operation in digital systems is also presented, mainly by addressing the most important design techniques and its principles.

In chapter 3, the new methodology proposed to achieve the best compromise between power and performance in a digital circuit is here presented. The trade-off to minimize power without jeopardizing performance is analyzed for a PTM (Predictive Technology Model) 65nm CMOS cell library [37], using spice simulations to calculate and compare multiple parameters such as power, delay, energy. The new methodology is defined by analysis of individual logic cells; however, the method extends to multiple logic cells of a defined library (complete circuit). In this chapter, the subthreshold operation reliability problems are also summarized, and possible solutions are presented.

Chapter 4 presents the MERIT Software Tool, which is a tool developed to analyze automatically a complete digital circuit based on systematic digital logic simulations. This automatic analysis allows to characterize the circuit for power dissipation, energy and propagation delays, in respect to different supply voltages. Therefore, this software tool implements the methodology developed in this thesis and explained in chapter 3 to find the best compromise between power and performance

Simulation results are described in Chapter 5, using MERIT simulation software and its underlying methodology to different components and benchmark circuits. Results are presented in detail to illustrate the application of the method presented by this work as well as several observations and results that served as a basis for the definition of the method.

Finally, Chapter 6 summarizes all the main conclusions of the work developed in this thesis and points out directions for further work.

2. ENERGY EFFICIENCY WITH VOLTAGE SCALING

The need to improve energy efficiency during design and conception of digital modern systems becomes increasingly important as technology further reduces in size becoming more compressed, having overall smaller dimensions.

Globally, the widely used strategy to further reduce power consumption while maintaining circuit's functional integrity is defined as "voltage scaling". This technique allows voltage supply variations that can be dynamically controlled according to the circuit needs at each moment, therefore using this technique the energy supplied to the circuit is minimized, as the circuit processing power required for some operations can be significantly higher at some moments than the circuit processing power required during most of its idle time.

Makes sense to efficiently manage power supplied to a circuit as some types of circuits (increasingly more circuits) have strict power requirements, determined by smaller sized devices or battery operated devices running continuously or even by the important notion that general electronic digital systems must quickly evolve towards energy efficiency.

In Voltage Scaling techniques, power reduction is achieved at the expense of performance reduction. This means that decreasing energy in a circuit slows down its operation, decreasing processing speed. However, several types of new small circuits value much more energy than processing speed, and this concerns mainly (but not only) to circuits designed to operate for large periods of time, periodically executing reduced performance tasks. Some application examples for this kind of circuits include, for example, modern small wearable devices, as well as any kind of "sensor node" or "actuator node" designed for IoT (Internet-of-Things). Generally, any kind of small device operating on batteries for large periods of time with low energy requirements can greatly benefit from this method.

As this "voltage scaling" overall method has a clear cost regarding lower performance, there are also digital circuit applications that require high performance, processing intensive tasks, for most of the time. Therefore, this kind of circuits have much lower benefits from the use of this technique.

More recently, new types of circuits implement energy harvesting methods, collecting small amounts of energy from surroundings to supply its operation. This new and modern approach to energetic supply strongly require, however, ultra-low energy requirements for circuit supply. In this case, it is clear that this kind of circuits can highly benefit from the subthreshold mode of operation. Thus, the concept of “voltage scaling” can be applied to the subthreshold mode of operation. When both methods are associated, the use of voltage scaling at subthreshold level allows reducing supply voltage of a circuit much lower than the common normal voltage threshold for transistor operation.

Circuits operating at subthreshold level are considered “ultra-low power circuits“. This definition results from the square magnitude decrease on consumed power on ratio to circuits supply voltage.

The next section 2.1 presents the basis of Dynamic Voltage and Frequency Scaling, while section 2.2 presents the basis of subthreshold voltage design methods already available in literature.

2.1. DYNAMIC VOLTAGE AND FREQUENCY SCALING

Usually the workload of a processor is not constant, as far as performance is concerned. Instead, for most real-life applications, the workload of a processor varies significantly over time. It means that there are moments when the performance required from the system is much higher, compared to the remaining working time when the system is probably idle, or performing non-intensive routine tasks. Since normal operation of a processor system does not have a constant level of performance, instead has a dynamic level which changes its workload at each moment, also the energy supplied for system operation should be dynamic. Therefore, the energy supplied should be higher when the system needs more performance and lower when the system does not need to favor performance, but should favor energy savings. Dynamically changing voltage supplied to the system allows important savings on energy consumption as the supply voltage restriction minimizes energy spent on idle times. Then, energy obtained from the power supply defined as the integration of power over time can be highly reduced.

The following Fig. 1 presents the main principle of DVFS, Dynamic Voltage and Frequency Scaling. As we can see, both tasks, ‘task 1’ and ‘task 2’ are completed quickly at 100% voltage (fixed voltage), but could still comply to timing constraints (being completed within allocated time slot) by running more slowly, at lower voltages and lower frequencies.

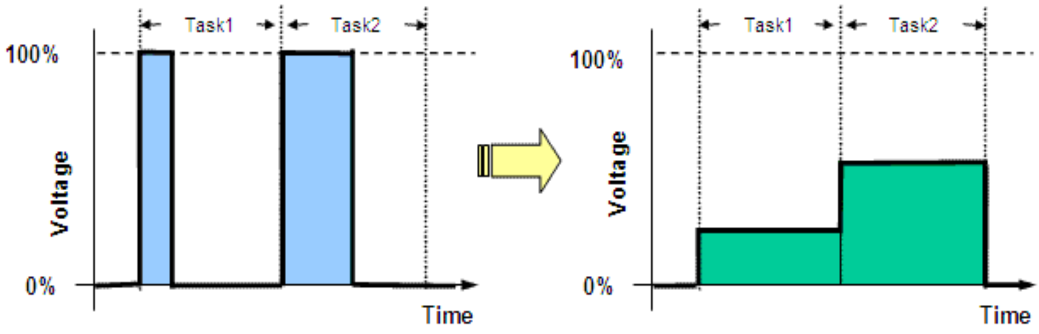


Fig. 1 - DVFS Main principle, increasing energy efficiency while complying with time constraints [39].

By using dynamic control of voltage and frequency, it is possible that time constraints are met and energy is saved, this way within timing boundaries, by reducing processor idle times we increase energy efficiency.

2.1.1. DYNAMIC FREQUENCY SCALING

The technique defined as “dynamic frequency scaling” can be applied to reduce power dissipated by a circuit, considering this technique during moments of lower workload for a system, the frequency of operation can be reduced, sacrificing operation speed not needed for non-intensive tasks, reducing frequency of operation strongly reduces power dissipated, even while maintaining supply voltage constant.

To reduce energy for modern microprocessor based digital systems, we must consider the different sources of power dissipation for common digital CMOS circuits, as stated by the following equation (1):

$$P_{total}= C * V_{DD}^2 * f + (I_{sub}+I_{diode}+I_{gate}) * V_{DD} \qquad (1)$$

As we can see, there are two main components for total power dissipated by a circuit, the static power dissipation component and also the most important component, the dynamic power dissipation component.

The dynamic power dissipation (or switching power dissipation) is the dominant component of the total power dissipation value, and is proportional to the total capacitance of the circuit, proportional to frequency and also to the square of the supply voltage. Considering that this is the dominant factor, by reducing operation frequency, the overall dissipated power for the circuit can be significantly reduced.

The static power dissipation component of a digital system is related to various leakage currents occurring on the circuit and may generally be considered much lower than the switching power dissipation value.

From equation (1), we can see that the reduction of the power-supply voltage and the frequency can reduce the dynamic power, which is the most important part of the power in a digital CMOS. But, reducing the power-supply voltage may impose the need for level-shifters, when only part of the circuit is subject to VDD reduction. This increases circuit complexity, and in these cases, the reduction of the frequency can alleviate and avoid these problems. However, simultaneous voltage and frequency scaling can achieve further energy improvements.

2.1.2. DYNAMIC VOLTAGE AND FREQUENCY SCALING PRINCIPLES

Modern CMOS circuits that are the building base of common microprocessors and general integrated circuits operate over a certain supply voltage range with reasonable reliability. Therefore, usually the supply voltage value is defined as a securely over dimensioned value that can guarantee reliable operation for the system.

Considering a specific manufacturing technology, there is a maximum voltage limit beyond which the circuit can be damaged; similarly, there is a minimum voltage limit beyond which the circuit operation becomes unreliable and functionality failures occur. Somewhere between these limits there is the supply voltage range for reliable operation, and the supply voltage value can be defined statically to be constant, or the supply voltage value can have dynamic variations during circuit's lifetime.

To guarantee a system's reliable operation, all circuit's "delay paths" time delays must be covered by the delay of operation frequency pulses. Within the reliable operation supply voltage range, all the delay paths of the system increase proportionally to the decrease in supply voltage value. Therefore, the time delay of the critical path defines and restricts the choice of the frequency's minimum value. The trend to progressively reduce the supply voltage of a system to harness energy savings during its lifetime must consider consequences, and the most important one is the increase of the critical path and possible system failure. Therefore, there has to be concern also regarding adjustments in the system's operation frequency.

The technique defined as DVFS (Dynamic Voltage and Frequency Scaling) intends to maintain performance and guarantee system's reliability at all times, while obtaining energy savings. To achieve it, the technique works by dynamically reducing frequency along with supply voltage, thus compensating critical path increase. According to this method, the system's main features such as supply voltage, frequency and critical path must be carefully monitored and permanently adjusted, so that energy consumed can be minimized, while maintaining system's performance as required.

Defining the supply voltage to match the workload required at each moment is the main goal of the DVFS method. As supply voltage level squarely affects overall switching power dissipation, it is the most important factor to be reduced, on an effort to reduce total power dissipation. The implementation of a fast DVFS method from [40] using a new and better integrated voltage regulator is presented in Fig. 2, illustrating the continuous scaling process of this method.

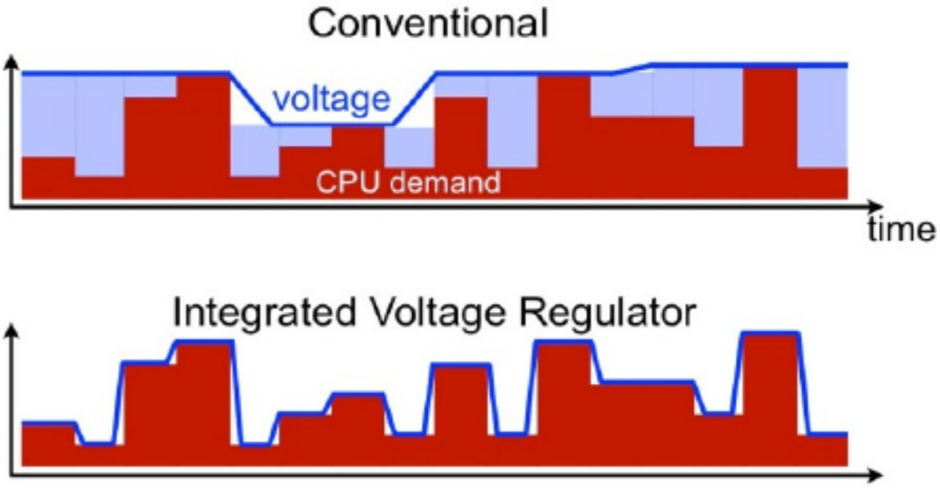


Fig. 2. Detail on the scaling process of an improved DVFS method [40].

2.1.3. DYNAMIC VOLTAGE AND FREQUENCY SCALING PROCESS

To better understand ground of application of the DVFS method, as well as its main principle of operation, we should consider analyzing the hardware building blocks necessary to implement DVFS control on a system. This way, not only the theoretical process can be defined, but also the practical implementation can be reviewed.

To match the changing demand of processing power and performance, not just power but also frequency must be dynamically adjusted. This process requires high level of control over a circuit, and for this reason, the hardware blocks described in this section are required and must be integrated. Moreover, understanding their application and function clarifies the complete DVFS principle and process.

The block diagram model for DVS hardware is presented in Fig. 3, obtained from reference [41]. This model describes the block diagram for a typical DVFS scheme.

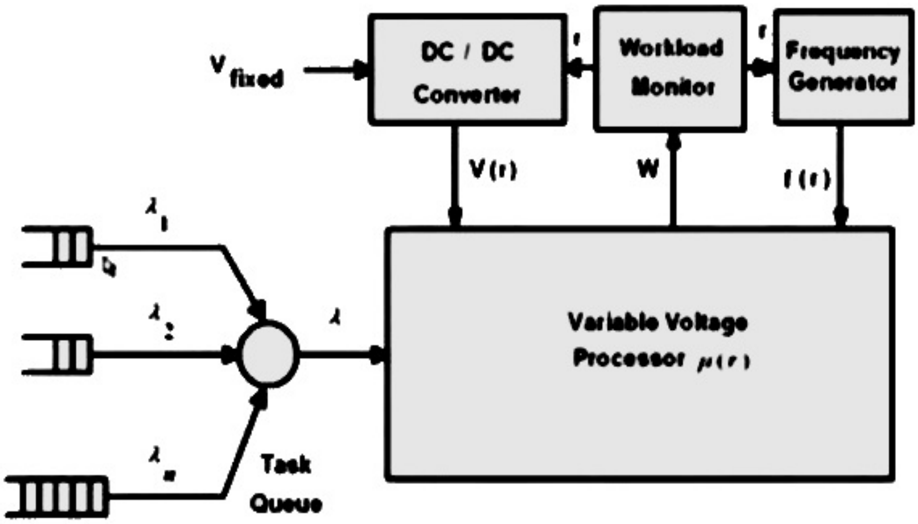


Fig. 3. Block diagram of the model for DVFS hardware [41].

2.1.3.1.WORKLOAD MONITOR

To apply the DVFS method, the main factor to be considered is the necessary workload and required processing power at each moment. Therefore, current workload of a micro processed system must be monitored from a special kind of hardware block, i.e., a block that can quickly and easily measure work being done. Usually the workload monitor can be provided by the processor itself, thru its registers and internal functions. To improve application of the DVFS method, the workload monitor block must be able not only to be extremely fast at workload measurement, but also should be able to predict, to some extent, probable workload increase or decrease on a very immediate future. This prediction must be as accurate as possible, in order to quickly fine tune system features for new operation conditions.

Prediction of future workload on micro processed systems is extremely hard to do, as those are non-deterministic applications. However, the prediction process is key of success for effective implementation of the DVFS method. An accurate and fast prediction on the future trend of the workload being processed by a system can allow immediate and correct redefinition on system power and frequency necessities, improving overall DVFS efficiency, and consequently, improving overall system's energetic efficiency.

2.1.3.2.VARIABLE FREQUENCY PROCESSOR/GENERATOR

To quickly and dynamically generate new operating frequencies for the system being controlled, DVFS control system relies heavily on a fast frequency processor/generator.

Dynamically variable frequency is generated by the use of a high performance phase locked loop (PLL) system. The PLL system consists of a phase frequency detector (PFD), a programmable on-chip filter (PF), as well as a voltage controlled oscillator (VCO). The PLL core generates a high speed clock which drives a frequency divider the divider creates the output variable frequency to be used as the DVFS output frequency.

2.1.3.3.VARIABLE VOLTAGE PROCESSOR

To implement the DVFS technique to achieve improved energy efficiency on such a system controlled by a microprocessor, a special kind of processor is required, as this processor must accept a wide range of supply voltage levels as well as a wide range of frequency values. Several such processors can be manufactured using present technology families and are commercially available, and possible examples include Strong ARM processor, as well as Transmeta's Crusoe processor [47].

For example, "Transmeta Crusoe" [47] is a new generation processor that provides several variable operating modes. Due to a specially designed dynamic core, they can vary voltage and frequency dynamically under dynamic load conditions. Fixed pre-defined frequency range and dynamic voltage provides for 300 MHz-1.20 V, 400 MHz-1.23 V, 500 MHz-1.35 V,600 MHz-1.53 V,700 MHz-1.75 V, 800 MHz-2.00 V, 900 MHz-2.35, 1000 MHz- 2.80 V. These ranges must vary mostly depending upon the workload necessary.

2.1.3.4.VARIABLE VOLTAGE GENERATOR

The efficiency of a DVFS system requires also that most of its core components can be extremely efficient at its function, or they may be contributing to energy waste they were meant to reduce. This applies not only to the variable voltage generator, but also to the other three hardware blocks presented.

Presently is possible to build a very efficient variable voltage generator based on a direct current DC-DC converter. Such hardware converter, when built using good quality components such as integrated gate bipolar junction transistors (IGBJT), can achieve a very good efficiency of 90% or better. One of the best and more efficient ways to define a good Variable Voltage Generator for DVFS consist on a DC-DC Converter based on pulse-width modulation.

The variable voltage generator has the specific task to manage the supply voltage delivered to the processor, according with the amount of workload being done at the moment. For that reason, the variable voltage generator must receive feedback on instant performance needs from the workload monitor hardware block.

2.1.4. ADAPTIVE VOLTAGE SCALING

Conventional Dynamic Voltage Scaling, (DVS) approach is an open loop technique, where a finite number of voltage-frequency pairs are initially defined to be used and can be dynamically selected during different workload conditions, at runtime, during a system's operation lifetime. However, these pre-defined frequency-voltage pairs are designed to keep sufficient margin to meet demanding application throughput requirements, for each of the pre-defined ranges or steps. For each of the value pairs there is a security margin, to guarantee reliable operation across the range of best and worst case PVT conditions.

This conservative approach of defining operation ranges, considering wide margins to secure reliable operation, means that usually there is an excessive margin of power wasted, resulting in efficiency loss. DVFS is commonly based on pre-defined voltage/ frequency pairs covering different ranges corresponding to different discrete step values. As these range intervals have a discrete nature, they are usually over dimensioned to cover worst-case scenarios of performance requirements for each voltage/frequency range.

It's clear, that the benefits on power efficiency obtained by applying common open loop DVFS methods based on pre-defined power-frequency steps can be improved as efficiency obtained is lesser than the real efficiency that can be achieved. Adaptive Voltage Scaling can be a solution to optimize this energy efficiency.

As shown in Fig. 4 obtained from [42], for a 130 nm technology it is possible to achieve gains of 36% energy saved when using DVS, while 64% energy saved if using AVS, both values compared with a fixed voltage supply.

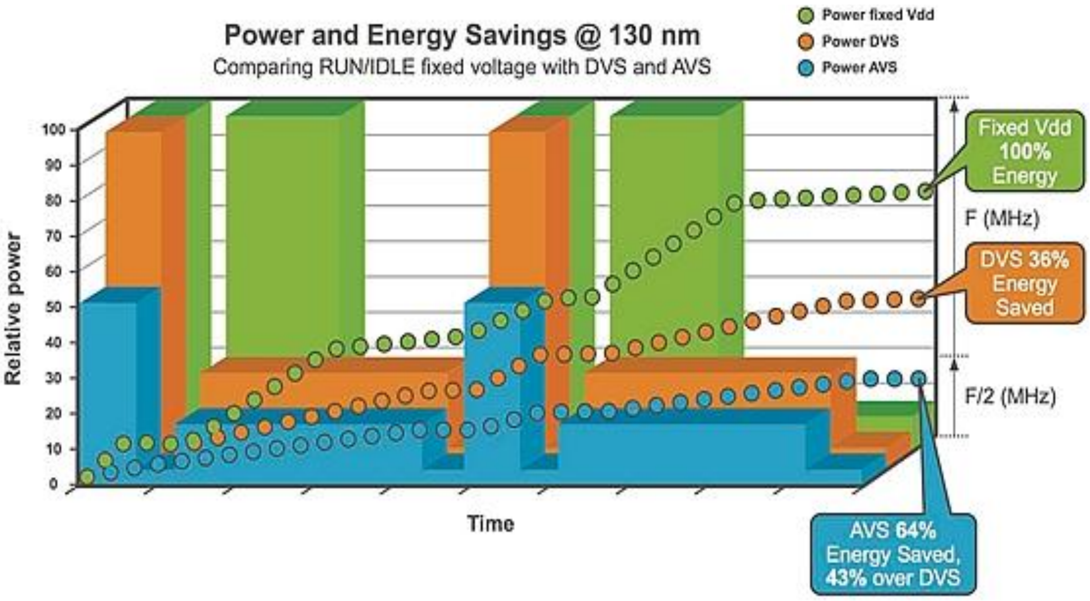


Fig. 4. Power and Energy Savings at 130nm, comparison for DVS, AVS, Fixed Voltage [42].

2.1.4.1.FUNCTIONAL METHOD

Adaptive Voltage Scaling (AVS) emerges as an important alternative to common DVFS, aiming to overcome limitations related to fixed interval ranges and trying to improve efficiency through the use of continuous variations of supply voltage. Vendors like AMD, already apply AVFS techniques to some families of CPUs, like for example its 'Excavator' CPUs cores, [43][44] incorporating 10 AVFS modules with energy savings as presented in Fig. 5.

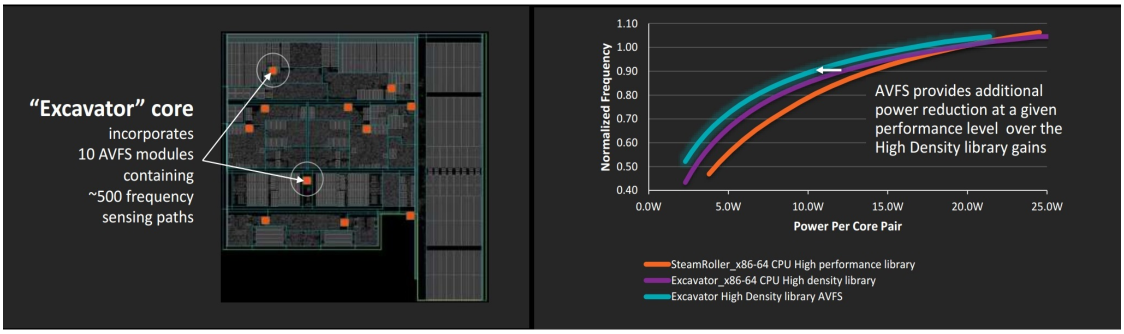


Fig. 5. AMD Excavator CPU Cores incorporating AVFS modules, detail on energy savings [43][44].

To create continuous variations on the supply voltage, the improved AVS method must rely on a highly efficient close loop feedback system, designed to measure workload directly at runtime and immediately act, defining output of a high precision variable power supply.

An AVS system concept must be carefully analyzed focusing its main functional blocks in order to better understand the AVS operational method. The Voltage Supply of the variable power source is directly controlled by a "delay sensing performance monitor". This sensor, usually embedded into the processor, not only monitors voltage, but also senses performance using high quality speed detectors. Whenever "delay sensing performance monitor" internal detectors sense slowness and performance loss under high workload requirements, they act by increasing supply voltage on power supply; whenever detectors sense high performance under low workload requirements, they act by slowly decreasing supply voltage, until the edge of detection. This way the supply voltage is regularly (continuously) and permanently adjusted to its optimum value.

2.1.4.2.IMPLEMENTATION

It is possible to assure that the performance of the system is exactly enough as required. It is also possible to assure that the system performs reliably. And mainly, it is possible to guarantee that the supply voltage defined is exactly the minimum voltage required for operation at each moment.

Fig. 6 obtained from reference [45], details the block diagram of a functional method for AVS implementation, highlighting the adaptive mechanism of control for the voltage desired to run the Processor/ASIC (VAVS).

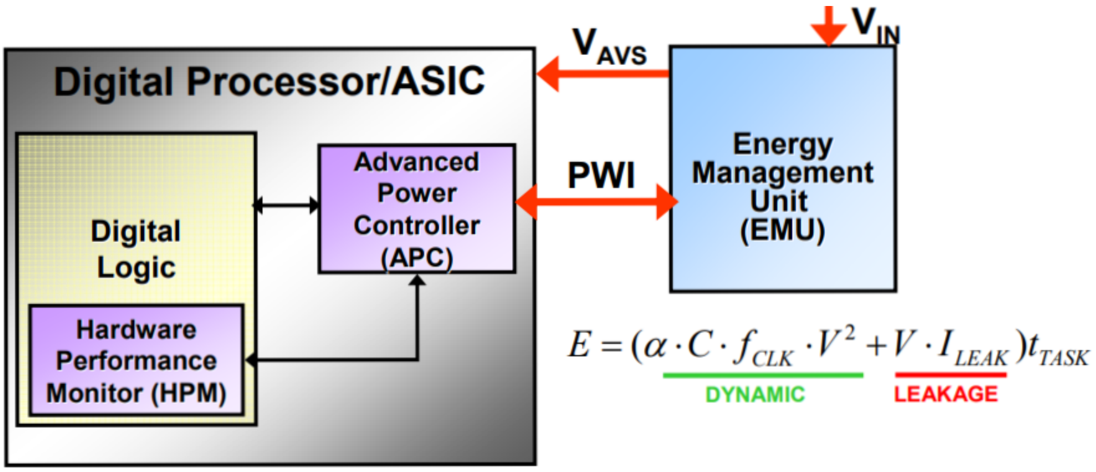

Digital Processor/ASIC
Digital Logic
Hardware Performance Monitor (HPM)
Advanced Power Controller (APC)
V_{AVS}
PWI
V_{IN}
Energy Management Unit (EMU)
$E=(\alpha\cdot C\cdot f_{CLK}\cdot V^2+V\cdot I_{LEAK})t_{TASK}$
DYNAMIC
LEAKAGE

Fig. 6. Block Diagram for Adaptive Voltage Scaling Process [45].

The main principles for AVS implementation are the following.

A hardware block defined as Dynamic Voltage Controller emulates critical path characteristics of the system, by means of a delay synthesizer. By analogy, the AVS controller estimates at runtime the current critical path delays and acts accordingly.

Another important hardware block is defined as DFC, or Dynamic Frequency Controller, which continuously adjusts clock frequency according to system activity. DFC monitors activity by processing data from multiple sources of information, but mainly from data received from the microcontroller, gathered periodically from RAM, BUS and core CPU. This information is used to calculate total large-scale integrated activity factor (LSI), and obtained results are then used to act upon the frequency adjuster to accurately define a new operating frequency for the processor.

According to AVS defined principles of implementation, it is clear that the dynamic voltage and frequency management system can track the required performance, considering process deviations at all moments, and very accurately predict the required minimum values for voltage and frequency.

According to reference [45], for custom ASIC/SoC Design, with process 65nm and frequency greater than 750MHz, by applying the defined AVS methods it's possible to expect energy savings varying from roughly 27% to 40%, compared with a fixed voltage power.

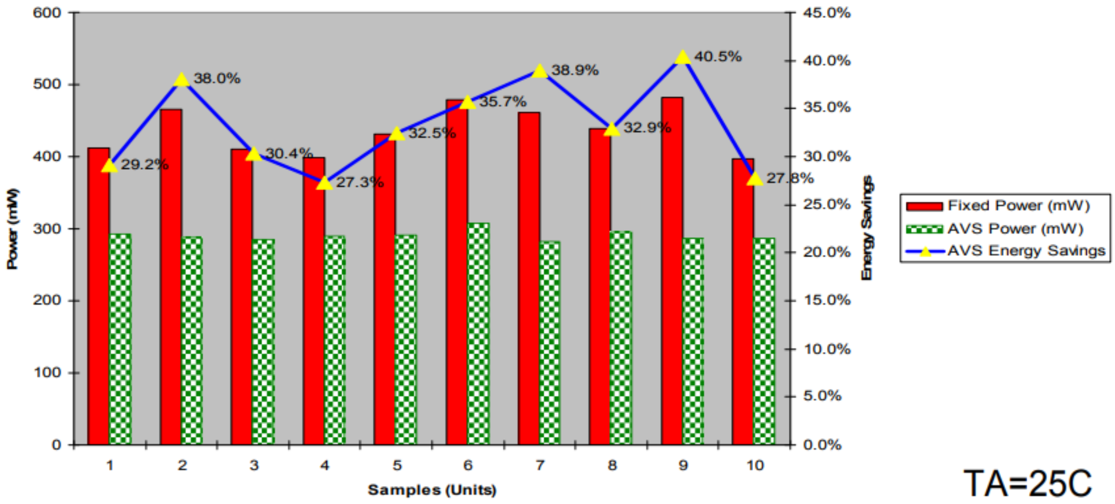


Fig. 7. Custom ASIC/SoC Design (65nm) energy saving by AVS methods [45].

2.2. SUBTHRESHOLD OPERATION DESIGN TECHNIQUES

One of the most important research areas regarding optimization of energy for digital circuits operation, in the last few years, is the research on "sub threshold design techniques". This topic became increasingly important for circuits whose application does not require permanent and intensive performance, or for applications where processing speed is not a critical factor. Its applications can be wide, from digital circuits [15][16], to analog circuits [11][12], mixed-signal applications [10], or even at memory applications [8][9]. To understand the trend concerning the evolution of subthreshold design techniques, the work presented in [6] must be considered, as it presents a complete review of several studies on the field and explores all aspects of subthreshold design methodology.

Considering the increasing need of better and more efficient designs to optimize the required energy for a circuit, some works are focused on the modeling and characterization of new devices designed specifically for operation at subthreshold levels [13][14]. Other works focus on trying to establish ground rules and methods on how to design logic devices, that can fully work on optimum energy points at sub thresh-old modes [1][2][3][4]. In [1][2] the concept of energy minimization is defined, and analytical methods are presented to allow calculating the Optimum Supply Voltage (optimum VDD) and Vth (Threshold Voltage), for a specific operating frequency and minimizing power. Transistors' sizing is extremely important to optimize energy reduction in subthreshold circuits, as denoted in [3]. Yet, it is also important the characterization of design techniques that aim to minimize operational errors by introducing new fault tolerant methods, as well as new and more robust cell design techniques to significantly improve liability of digital circuits. This is the purpose of [5], where authors propose to apply techniques of adaptive body biasing as well as body dimensions adjustments to achieve a reliable minimum energy operation.

To determine the optimal operating conditions to reduce energy in a complete circuit is not an easy task. Previous works define the optimal VDD for each gate using ring-oscillators simulations [1][2], but for a complete circuit, the optimal VDD for a certain gate will not certainly be the optimal VDD for all gates in the circuit. The complete design of a new standard cell library fitted to work at subthreshold voltage levels is certainly rewarding for the optimal energy operation, as well as for restricting operational errors, as presented in several works in literature [15][16]. However, a compromise between several parameters and different gates should be considered when de-fining the optimal VDD value for an ultra-low-power operation. Moreover, reusing an existing standard cell library to work at subthreshold voltages can also be rewarding, as long as methods and strategies can be define to find a specific set of optimal operating parameters (such as VDD and clock frequency) for energy optimization across all circuit, and that reliability and fault-tolerant methodologies are used to restrict error occurrence.

Although this work is based on determining the optimal power-performance operation point for an existing library cell, as mentioned before, the standard cell design targeting the work at subthreshold voltage levels is certainly rewarding, both for optimal energy operation, as well as (and most important) for restricting operational errors [15][16]. The following section 2.2.1 presents the basis of a standard cell design, presenting the main aspects and objectives of design using typical cells' examples.

2.2.1. STANDARD CELL DESIGN FOR SUBTHRESHOLD OPERATION

To clearly understand the main problems of cell design for subthreshold operation, let us consider in this section the design of several basic cells such as inverters and combinatorial XOR gates, these gates are here presented as examples and important basis for developing a complete library of logic gates suitable for subthreshold.

2.2.1.1.INVERTER SIZING

If we consider a proper inverter operation for a full output swing, we must understand the causes and effects of variations of the following parameters: process variations, low supply voltage and transistor sizing design. Generally, when designing the simple CMOS inverter for subthreshold operation, the first step is to define the size of the upper PMOS device. Defining its dimensions allow us to give him the strength enough to pull up the inverter output to an "high" voltage level, as required, reducing and cancelling the effect of the fast and leaky lower NMOS device, that more easily pulls the output down.

When analyzing a logic gate operation it is important to consider the I_{on}/I_{off} ratio, as this is the ratio that validates if a logic gate can function properly. I_{on} is defined as the drive current along the devices and I_{off} is defined as the idle current on the entire logic gate. To fully understand this issue, we must analyze both truth table states for the CMOS inverter, considering the control of drive and idle currents on both devices to improve the gate reliability and ensure proper function.

- **First State (Input: 0; Output: 1)**

When we apply a "low" logic state to the input, input="0", the PMOS device must pull-up the output of the inverter to output="1", to achieve the normal desired result. First, let us assume that we want to decrease V_{dd} on the inverter. Decreasing the supply voltage of the inverter, we greatly increase the idle currents across the logic gate, not only on the PMOS, but also in NMOS. Considering equal dimensions for the NMOS and the PMOS transistor, we can verify that the NMOS transistor acts faster, therefore is leakier than the slow PMOS transistor for this logic gate configuration, and this is called the fast NMOS/slow PMOS effect (FS).

For low supply voltages the increase on idle currents can trigger first the NMOS transistor pulling down the output to 0 when a logic “high” would be expected, for this case the “I_{on}/I_{off} ratio” decreases (as I_{off} increases) and a gate failure is generated.

To avoid this failure we must balance the “fast” nature of the NMOS transistor, allowing a better current drive on the PMOS transistor. This can be accomplished by increasing the size of the PMOS, allowing more current to flow. However, this increase must be carefully calculated. Therefore, to match equal current flow conditions on both transistors, not only for dynamic but also for leakage currents, the transistors must have different dimensions.

For this case, the PMOS must have a larger width to be able to drive the output to V_{dd}, this modification is increasingly important for lower supply voltage values and higher idle (static) currents.

For our first case, we analyzed that there is a minimum width for the PMOS transistor, because of the fast NMOS/slow PMOS effect on the inverter, to ensure proper function of this logic gate.

- **Second State (Input: 1 – Output: 0)**

If a logic “high” is applied to the input, input=”1” the transistor NMOS pulls the output node to low, output=”0” as expected.

As explained before, the PMOS dimensions (width) must be increased to match the current flow allowed by the NMOS transistor. However, for the second case we need the NMOS to bring output low and PMOS must stay OFF to prevent the output going to a “high” logic state. In order to prevent the PMOS pulling up the output, leakage (idle) currents flowing thru the PMOS must be minimized, so the transistor cannot have a “large” size, large sized transistors allow larger idle currents, then there is a maximum limit to the size of W for the PMOS transistor. If we continuously increase the PMOS width size, we will get to a point where the magnitude of the idle current flowing on PMOS will match the magnitude of the drive current on the NMOS, pulling the output high and therefore generating a gate failure. This failure results from the slow NMOS/ fast PMOS effect.

Consequently, there is a maximum boundary on W size defined for the PMOS transistor to allow the NMOS transistor to pull down the output as expected.

- **Analysis of Inverter Sizing**

From the analysis of the first scenario, it is possible to verify that the PMOS transistor must be increased in size to balance the contribution for the inverter from the negative side transistor NMOS. It is clear that there is a minimum Wp size for PMOS for a proper logic gate function.

From the analysis of the second scenario it is possible to verify that there exists a maximum Wp size for the PMOS transistor, a size that when increased can lead to undesirable large idle currents disturbing the proper function of the logic gate.

Therefore, an interval exists for the sizing of positive transistors that allows for strict control of both the dynamic and static currents through the gate, balancing the contribution of positive and negative transistors on different states. The analysis of a CMOS inverter proves to be a key element to properly design a logic gate prepared to operate on low or very low supply voltage conditions, as such on subthreshold operation. It also provides a basis to define rules and methods to design the configuration of larger and more complex logic gates, while ensuring that these gates work as expected.

2.2.1.2.TRANSMISSION GATES FOR CELL DESIGN

When designing logic cells for very low power it’s important to carefully analyze traditional logic design of these cells mainly regarding internal transistor structure and arrangement. The analysis of traditional logic cell structure that needs to be done must focus on key effects that affect functionality of the logic cell at subthreshold level, potentially rendering the logic cell unusable at very low voltages. These effects are mostly due to stacked transistors and parallel leakage.

Problems arise when the order of magnitude of transistor drive currents (when transistors are at “on” state) becomes comparable to the order of magnitude of leakage currents (when transistors are at “off” state). Erroneous situations are more likely to happen when many parallel transistors should be off simultaneously, for multiple possible states. Then parallel leakage occurs and the sum of this leakage currents may compare to magnitude order of the sum of active drive currents.

On the other hand, erroneous states can also occur when multiple transistors are stacked together. This situation can also greatly increase the sum of leakage currents, increasing probability of failure on the output state logic level. Then, becomes clear that measures must be taken into account to avoid multiple parallel transistors, or multiple serial transistors, when designing logic cells to operate at very low voltages. An adapted logic cell must be carefully balanced as to its structure.

Transmission gates are logic gates with internal circuits that are created by a method intended to minimize parallel devices and balance contribution of internal transistors. These properties make these kind of logic cells ideal for low and very low voltage. Therefore, transmission gates must be considered as a primary methodology for the design of logic cells created for low voltage operation, and this technique should be applied whenever conventional logic of a gate consists of a strongly unbalanced positive/negative transistor arrangement.

- **Transmission Gate Example – XOR Gate**

The XOR Gate is a case of an important logic cell that requires special care, as traditional design of this gate exposes strong vulnerabilities regarding low voltage operation, mainly because of parallel leakage.

Consider the schematic in Fig. 8 from [7] for a common two port XOR logic gate, as generally expressed on libraries. Consider also internal transistor arrangement schematic, as an evidence of the parallel leakage effect occurring on this gate, and therefore creating logic output errors.

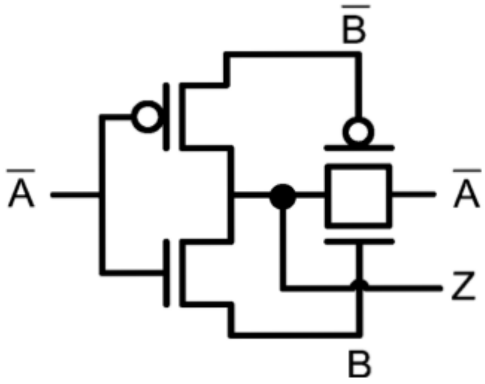


Fig. 8. Tiny XOR Logic Gate [7].

Functionality of the gate is severely compromised as several "off" parallel transistors increase idle current (I_{off}), decreasing functionality ratio I_{on}/I_{off}. The following Fig. 9, as presented by work in reference [7], details the tiny XOR 'unbalanced' leakage, leading to possible logic output errors.

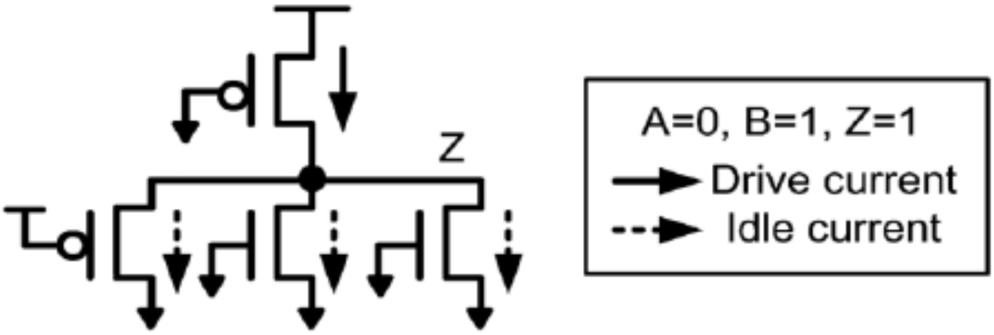


Fig. 9. Tiny XOR Gate 'unbalance' parallel leakage currents [7].

This low functionality ratio strongly compromise gate operation. In practice, the idle current of multiple "off" transistors "pull-down" the output level to "low", when it should be output "high", creating output degradation, or even an output level error. Note the output characteristics for a tiny XOR gate, operating at low voltage, which is presented in Fig. 10, represented in [46] where detail is shown for the logic error occurred due to leakage idle currents.

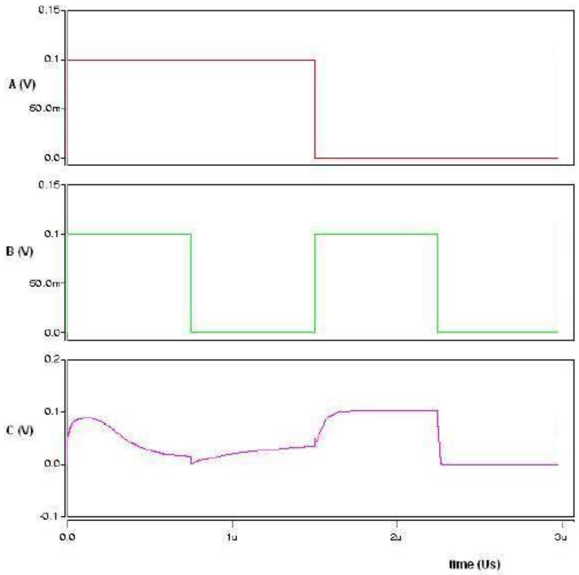


Fig. 10. Output characteristics for Tiny XOR Gate operating at low voltage [46].

Designing an XOR gate for low level, therefore, implies the internal balance required to improve and increase function ratio I_{on}/I_{off}, and it can be achieved by using an XOR logic gate implemented by transmission gates. The following Fig. 11 from reference [7] presents an XOR gate approach by using transmission gates more suited for operation at low voltages.

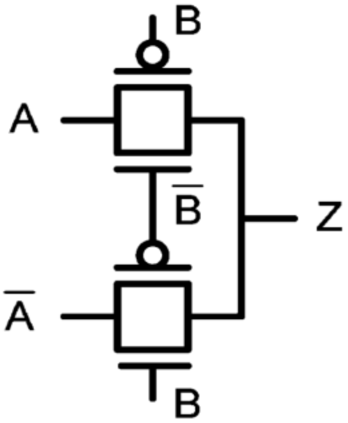


Fig. 11. Transmission XOR Logic Gate [7].

Consider also internal transistor arrangement schematic for this transmission gate. By inspection we find evidence that the parallel leakage effect occurring on this gate is severely minimized by this specific transistor arrangement on this gate. Fig. 12, as presented in work from [7], details the transmission XOR internal transistor arrangement, as well as the several idle and drive currents for each of the internal transistors.

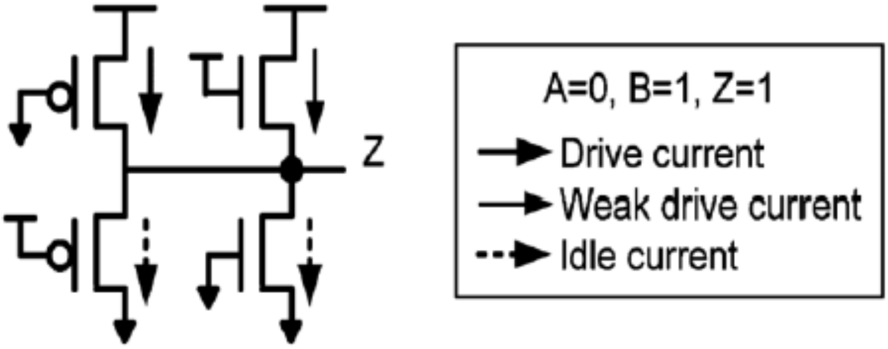


Fig. 12. Transmission XOR Gate ‘balanced’ leakage currents [7].

By analysis of presented schematics, it is clear that the logic gate maintains expected functionality, while improving its internal structure balanced transistor contributions to the output logic level. The overall balance on the gate is achieved, as both CMOS transistors (positive and negative) are coupled together, not only on the pull-up path, but also on the pull-down path. The coupling of complementary transistors, not only balance current flowing on each “analog switch”, but also balances current flowing on each logic path.

The following Fig. 13, as defined by [46], clearly presents output characteristics for a transmission XOR gate, operating at low voltage, detailing valid logic values for operation at very low voltages.

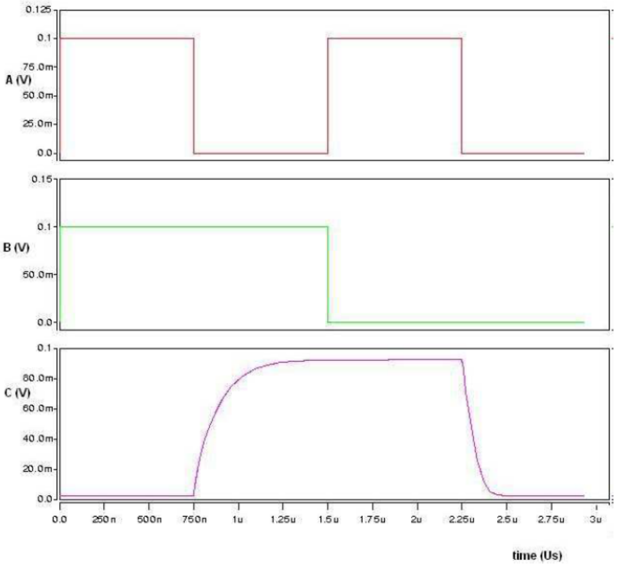


Fig. 13. Output characteristics for Transmission XOR Gate operating at low voltage [46].

As a consequence of the study, Fig. 14, as defined in [7], presents an implementation example of a transmission flip-flop, a sequential type gate suitable for operation at low voltage level. Note that by using a transmission gate to improve design, it greatly benefits functionality ratio, increasing this factor as most negative effects affecting subthreshold operation are greatly reduced.

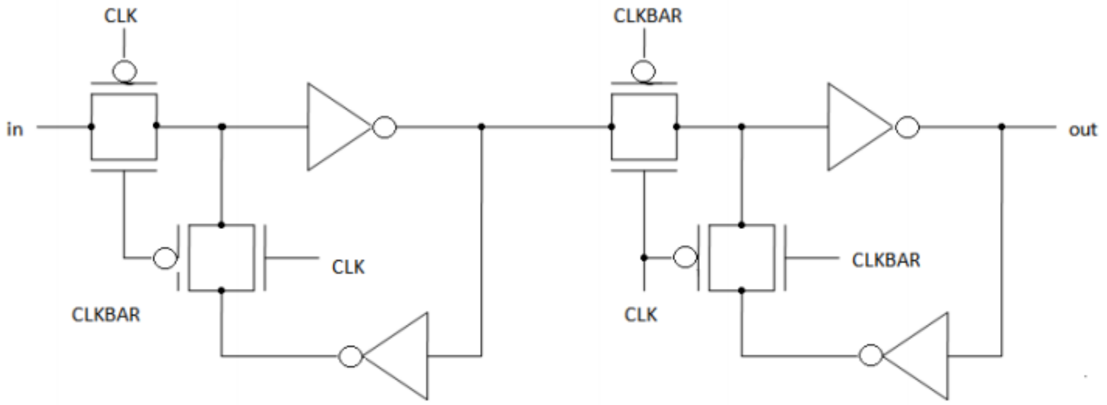


Fig. 14. Transmission Flip-flop, sequential gate suited for operation at low voltage [7].

3. THE COMPROMISE BETWEEN POWER AND PERFORMANCE

In digital circuits, a well-known figure of merit related with the energy efficiency of a logic gate or logic family is the power-delay product (PDP). This figure of merit is also known as the switching energy, because it is the product of the average power consumption over a switching event times the input-output propagation delay of the event, or duration of the switching event. As it is power time's delay, it has the dimension of energy, and measures the energy consumed per switching event. This figure of merit is used usually to measure the efficiency of performing an operation in a given technology, therefore is mostly a technology parameter and not suited to detect the optimum VDD for an ultra-low-power operation with a compromise between power and performance.

Instead of PDP, several authors prefer to consider the energy-delay product (EDP), which can establish a better balance between performance and power consumption [32] [33]. In these works [32][33]), minimizing the EDP (Power*Delay^2) of a circuit results in a particular design point in the energy-delay space where 1% of energy can be traded off for 1% of delay. But, although the EDP metric is useful for comparison of different implementations of a design, the design optimization points targeting EDP may not correspond to an optimum under desired operating conditions [31]. Instead, other metrics have been used in the past, like the works in [31][34][35][36]. If more weight is needed on the delay, a similar metric to EDP can be used, the Energy-Delay^n, i.e, Power*Delay^n with *n* greater than 2. Another approach possible is the sensibility metrics used in [31].

However, in this work the desired optimum VDD is not just the lowest energy point, but the best compromise between Power and Performance, reducing considerably power but not jeopardizing performance, neither the correct operation with an increased vulnerability to errors.

3.1. METHODOLOGY FOR BEST POWER PERFORMANCE TRADE-OFF

Let us consider a typical NAND3 gate, and let us analyze the average power and delay obtained from the events in one of its inputs (the A input), for the different and possible V_{DD} values. Simulations were performed using a Predictive Technology Model 65nm technology [37], with V_{DD} =1.1V and T=27ºC for nominal conditions (NC). Fig. 15 presents 2 line plots in the same graph, for delay and power over the possible V_{DD} values, obtained from NAND3 gate, input A to output events. Just by inspection, we can say that the optimum V_{DD} would be comprised within 0,2V and 0,5V, because for lower V_{DD} values the delay increase enormously, and for higher values is the power who massively increases.

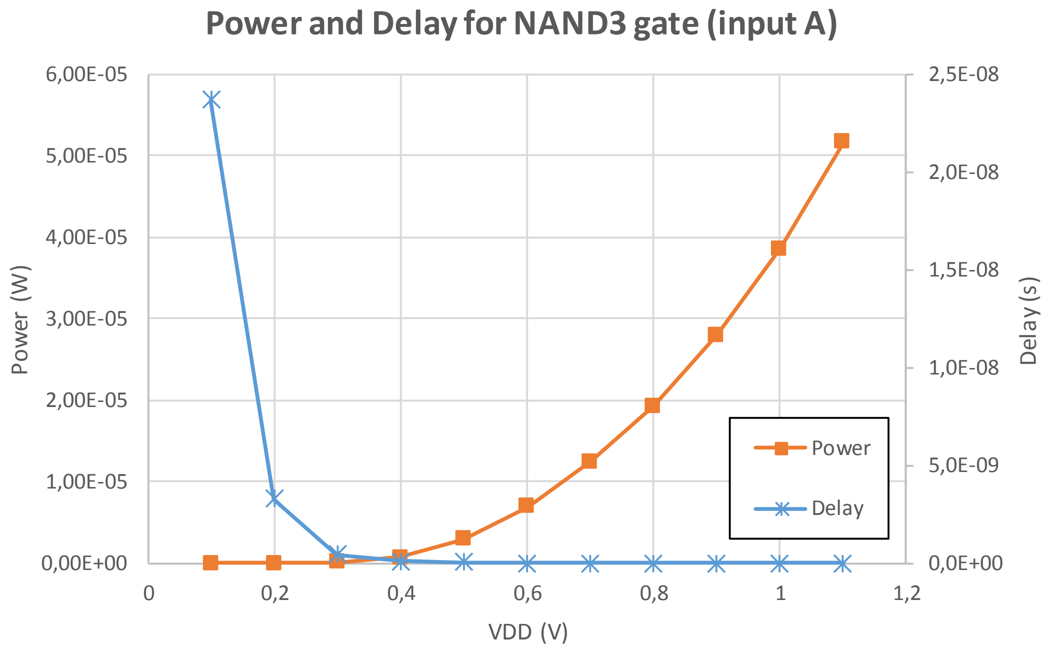


Fig. 15. Power and Delay plots over VDD, for a NAND3, input A to output transitions.

However, these plots are based on different scales and units, so they cannot be compared directly and, therefore, this will lead us to a very rough optimal V_{DD} value. Let us consider instead the normalized slopes of these curves, to analyze how this curves progress.

Fig. 16 shows the 2 normalized slopes over V_{DD} range. Note that: (i) the slopes are obtained from the derivative function of each plots from Fig. 15, i.e., the derivative function of the power and the delay for NAND3, input A, gate; (ii) for easier understanding, both slopes are normalized in respect to their maximum values, so that they can be represented in the same axis and can be compared.

The Delay growth shows us that for V_{DD} values below 0.3V, the delay will start to increase enormously. As for power growth, we can see that for V_{DD} values higher than 0.3V, power will grow immensely. Therefore, as they have opposite growth tendencies, it is easy to understand that the optimum V_{DD} value is 0.3V, for the provided simulation data. Plotting the distance between these two slopes (also plotted in Fig. 16), i.e., the absolute value for the difference of the normalized slopes, we can see that the minimum distance between these two slopes will determine the optimum V_{DD} value, where power is considerably reduced, and the delay is not considerably enhanced.

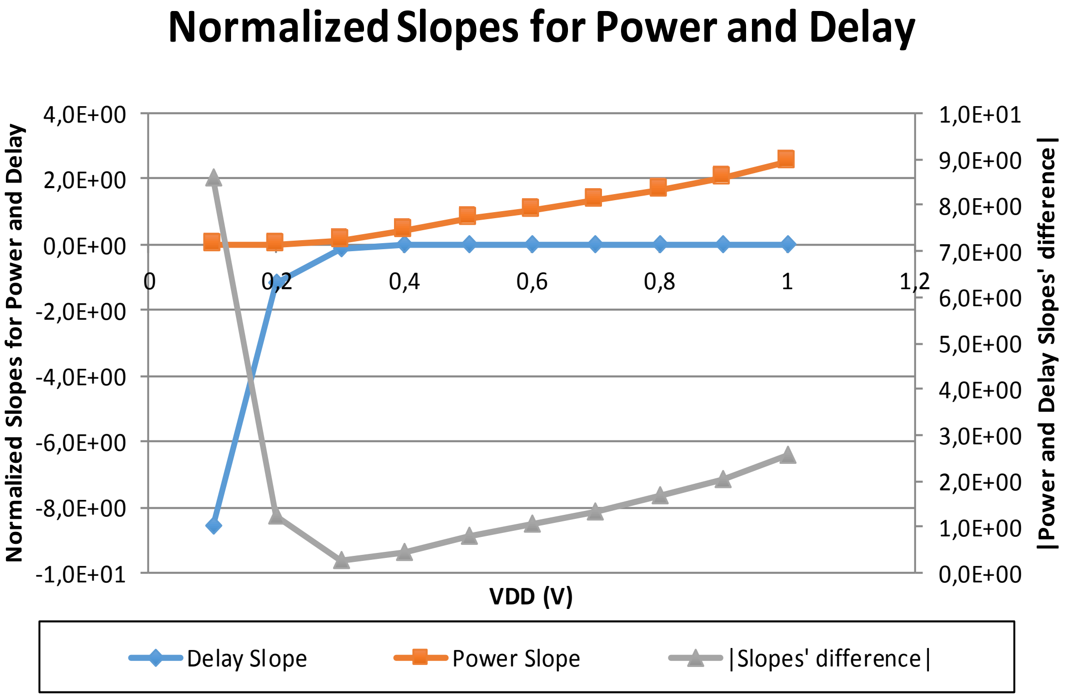


Fig. 16. Normalized slopes for Power and Delay over VDD, for NAND3 gate, input A.

This is the optimum V_{DD} where the best compromise between power and performance is achieved. Nevertheless, this V_{DD} value represents a coarse analysis result, and more data should be given in the proximity of this coarse optimum V_{DD}, to define a final optimum V_{DD} value.

Applying the same principle, but with a different representation, Fig. 17 shows a better visual representation of this optimum V_{DD} for this NAND3 gate, input A to output transitions. If the slopes for delay and power are represented in their normalized absolute values, and if a logarithmic scale is used to allow an easier representation of all values, the optimum V_{DD} value can be obtained in the same way, i.e., using the minimum distance between the slopes' lines.

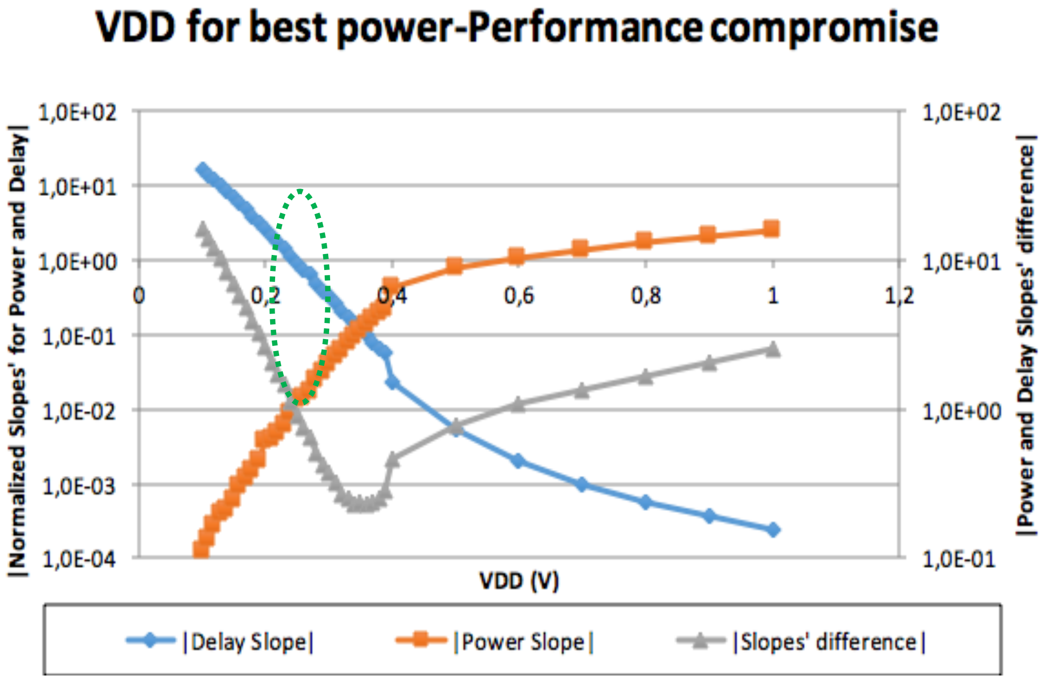


Fig. 17. VDD for best Power-Performance compromise.

Using the secondary axis on the right side of the graph, the absolute value of the difference between the normalized power and delay slopes is presented in Fig. 17 (using also a logarithmic scale representation).

As it can be seen, the minimum value for the distance is where the two lines cross (the distance is zero), and again the minimum value for the difference between the slopes is obtained for V_{DD} =0.3V, for the provided simulation data.

In resume, the optimum V_{DD} for a gate path is the V_{DD} values that satisfies equation (2). V_{DDopt} is obtained from the derivative function of power and delay functions in respect to V_{DD}, and calculating the V_{DD} for which we obtain the minimum distance between power and delay functions.

$$\frac{\partial\left(\left|\frac{\frac{\partial Delay_{gate\,path}}{\partial VDD}}{max\left[Delay_{gate\,path}\right]}\right| - \left|\frac{\frac{\partial Power_{gate\,path}}{\partial VDD}}{max\left[Power_{gate\,path}\right]}\right|\right)}{\partial VDD} = 0 \qquad (2)$$

3.2. EXTENDING THE OPTIMUM VDD CONCEPT FOR GATE AND CIRCUIT

To extend the previous methodology and calculate the optimum V_{DD} for power and performance compromise for a gate, all the delays and the corresponding power dissipations should be considered. As power reduction is the main goal of this methodology, let us define the gate optimum V_{DD} as the weighted average of all the optimum V_{DD} values in that gate (obtained from all the paths), weighted by the corresponding power dissipation. Equation (3) summarizes this gate optimum V_{DD} calculus.

$$VDD_{opt_{gate}} = \frac{\sum\left(VDD_{opt_{gate\,path}} \times Power_{gate\,path}\right)}{\sum Power_{gate\,path}} \qquad (3)$$

Consequently, for a complete circuit the optimum V_{DD} can also be obtained using a similar approach, as denoted in equation (4). The circuit optimum V_{DD} is the weighted average of all the optimum V_{DD} values in all the gates in the circuit (obtained from all the paths of each gate), weighted by the corresponding power dissipation.

$$VDD_{opt_{circuit}} = \frac{\sum_{gates}\left(\sum\left(VDD_{opt_{gate\ path}} \times Power_{gate\ path}\right)\right)}{\sum_{gates}\left(\sum Power_{gate\ path}\right)} \tag{4}$$

It is important to note that, the optimum V_{DD} of gate may be lower than the minimum V_{DD} allowed for a different gate in the same circuit. Although this is improbable to happen in a well-designed standard cell library, this means that one of the gates should be re-designed and improved, or an alternative circuit synthesis could be done. If this is not possible, than the optimum V_{DD} chosen should not be lower than the minimum V_{DD} allowed in one gate. In fact, accounting with unpredictability and using safety margins, working at subthreshold voltages impose that the working V_{DD} must have a safety margin from the minimum allowed V_{DD} (considered the minimum V_{DD} that delivers a correct logic behavior in the circuit).

3.3. RELIABILITY AT SUBTHRESHOLD OPERATION

By working at subthreshold levels, the main problem is reliability of the system. Therefore, a brief note about this is mandatory and is presented in this section.

As previously mentioned, integrated circuits operate over a certain supply voltage range with reasonable reliability, and the allowed supply voltage values are defined as a securely over dimensioned value that can guarantee reliable operation for the system. The reduction of the supply voltage into subthreshold levels imposes extreme risks that should be very well studied and tested.

The first problem is the increase of delays, which imposes a reduction of the operating frequency. This is the purpose of this work, to define the best compromise between the power-supply voltage reduction and the frequency reduction.

However, the reduction of the power budget available in the circuit makes it very vulnerable to all parameters' variations, environmental or operation induced. For example, process variations impose distinct path delay distributions in different circuit samples, and the differences obtained in the path delays are much amplified when power-supply voltage is reduced to subthreshold levels. The same happens with temperature variations or even temperature hotspots. Moreover, EMI can also be a source for additional problems, as the low energy available on chip may expose the circuit to EMI-induced errors, or soft-errors in general.

Therefore, the use of subthreshold power-supply voltages should only be used carefully. For applications where operation is critical and no errors should happen, this is not a solution. Fortunately, for several IoT applications like battery operated smart sensors, working remotely for long periods of time (where power is critical) to gather non-critical information, errors may happen occasionally. In this cases, this is a good solution to extend the battery life of the smart device.

Nevertheless, errors may still be avoided, or reliability may be increased, even with subthreshold operation and several parametric variations (environmental or operation-induced, or other). As explained in section 2.2, the standard-cell library used and specifically cell design should be done focusing subthreshold operation. Proper dimensions of the transistors can alleviate error occurrence. Moreover, existing reliability and fault-tolerance techniques can also be used and adapted for these subthreshold operation circuits.

Regarding the reliability and fault-tolerance techniques, if subthreshold operation is the focus, slack margins (or safety margins) used should all be increased, to account for unpredicted variability. Moreover, performance sensors can also be used, and this is a future work perspective. Previous works like [23] and [30] will be adapted for subthreshold operation and can effectively reduce errors, as this sensors work predictively, i.e., they trigger corrective actions before errors actually occur, and they sense performance changes, regardless of their origin (environmental or operational-induced).

Additionally, it is important to note that aging effects are reduced when power-supply voltage is reduced. At least aging is a lower source for problems.

4. THE MERIT SOFTWARE TOOL

This chapter presents the MERIT (cMos EneRgy sImulaTor), which is a software tool designed to automatically analyze a digital circuit or logic gate, and characterize it for power dissipation, energy and propagation delays, in respect to the power-supply voltage (VDD). MERIT operation is based on HSPICE simulations, which are invoked automatically to allow automated multiple circuit simulations and results' analysis.

The main purpose of this software tool is to implement and support the methodology proposed in chapter 3, to determine the optimum VDD for which the best compromise in power and performance is obtained for a given circuit or gate. Whether the analysis is done for a gate, or for a complete circuit, the operation is almost the same. In a first stage, the tool can be applied to single logic gates for multiple parameter simulation testing, and on a second stage the tool can be applied to full circuits and sub-circuits, automatically and sequentially simulating all the logic gates of the circuit for multiple parameters. The overall results for the complete circuit are thus obtained, gathered from multiple partial simulations. These multiple and sequential simulations over different logic gates and parameter variations allows to obtain results that are automatic analyzed within the tool and the results delivers conclusions regarding energy and delay for different initial conditions and parameters. As a final result, the optimum VDD is obtained, defined here as the best compromise between power and performance, for an ultra-low-power circuit.

4.1. IMPLEMENTATION AND FOCUS

The MERIT is an automatic software tool designed to assist on HSPICE automated circuit simulation. It was developed using the Delphi RAD Studio 2010 platform, and created to quickly and efficiently run and obtain multiple progressive simulation results from CMOs digital circuits.

To support the methodology proposed in chapter 3, the MERIT tool operation is based on two different stages: in a first stage the tool can be applied to single logic gates for multiple parameter simulation testing, while on a second stage the tool can be applied to full circuits and sub-circuits. The basis for operation over a complete circuit would be almost the same as the operation over a single gate. Thus, all the logic gates of the circuit are automatic and sequentially simulated for multiple parameters, and the overall results for the complete circuit are thus obtained, gathering the results from multiple partial simulations.

The software tool has the main objective of delivering quickly and efficiently HSpice simulation results that can be post processed to clearly present to the user detailed results on energy efficiency and performance/delays of the circuits. To achieve this result, the tool instantiates HSPICE simulator [49] to quickly perform multiple sequential simulations over different logic gates with small parameter variations. The results obtained are then used for further calculations within the tool, to perform the automatic analysis of post processed results and deliver conclusions regarding energy and path-delays for different initial conditions and parameters.

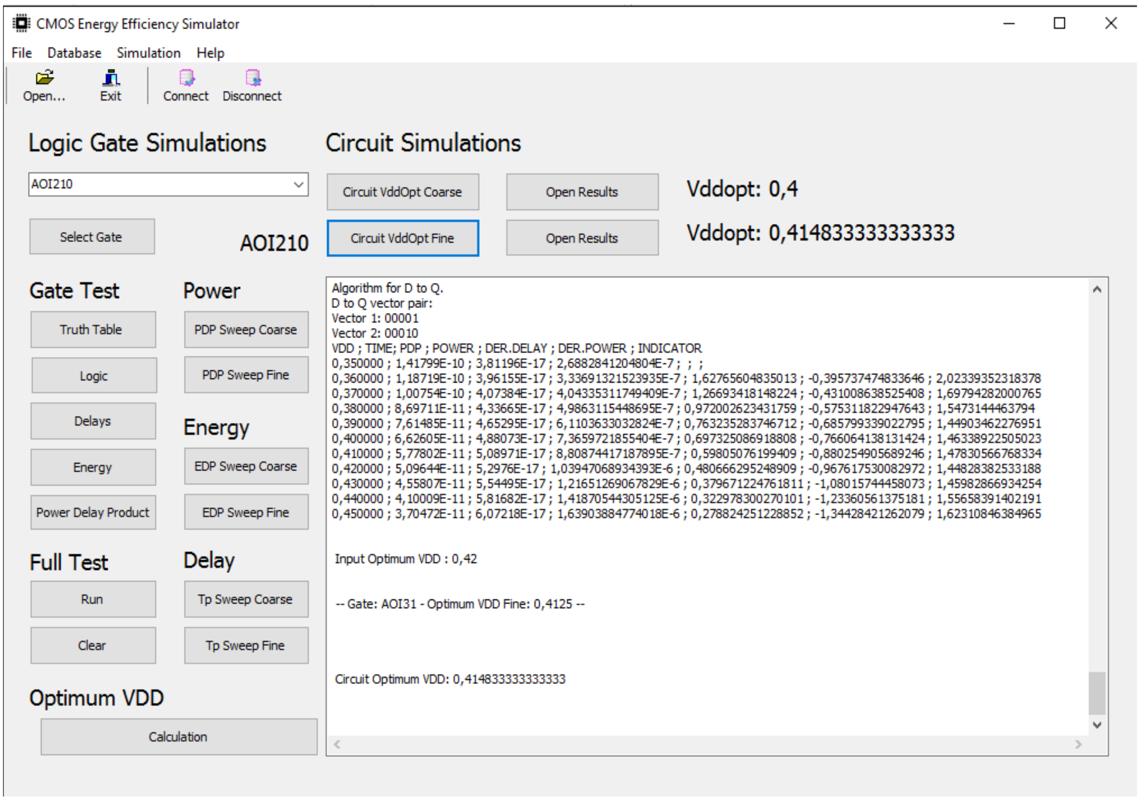
All the processed results are organized and displayed to the user in the main screen of the tool, but they are also recorded to different data text files for external usage and log information.

4.2. USER INTERFACE

The main interface of the MERIT software tool is presented in Fig. 18.

As it can be seen, several power and performance analysis are available to the user, and all the processed results are organized and displayed in the main screen. All individual simulations performed by the tool are organized and stored on different text files, allowing external usage and analysis.

To allow the connection of the tool to the logic cell library database in use, the tool provides a top menu and several buttons on the left panel, to allow running simulation tasks for individual logic gates, and several buttons on the top panel, to allow running simulations on a complete circuit. A third panel provides an area to display the simulation results.



AOI210

AOI210

Vddopt: 0,4

Vddopt: 0,414833333333333

```
Algorithm for D to Q.
D to Q vector pair:
Vector 1: 00001
Vector 2: 00010
VDD ; TIME; PDP ; POWER ; DER.DELAY ; DER.POWER ; INDICATOR
0,350000 ; 1,41799E-10 ; 3,81196E-17 ; 2,6882841204804E-7 ;  ;  ;
0,360000 ; 1,18719E-10 ; 3,96155E-17 ; 3,33691321523935E-7 ; 1,62765604835013 ; -0,395737474833646 ; 2,02339352318378
0,370000 ; 1,00754E-10 ; 4,07384E-17 ; 4,04335311749409E-7 ; 1,26693418148224 ; -0,431008638525408 ; 1,69794282000765
0,380000 ; 8,69711E-11 ; 4,33665E-17 ; 4,9863115448695E-7 ; 0,972002623431759 ; -0,575311822947643 ; 1,5473144463794
0,390000 ; 7,61485E-11 ; 4,65295E-17 ; 6,1103633032824E-7 ; 0,763235283746712 ; -0,685799339022795 ; 1,44903462276951
0,400000 ; 6,62605E-11 ; 4,88073E-17 ; 7,3659721855404E-7 ; 0,697325086918808 ; -0,766064138131424 ; 1,46338922505023
0,410000 ; 5,77802E-11 ; 5,08971E-17 ; 8,80874417187895E-7 ; 0,59805076199409 ; -0,880254905689246 ; 1,47830566768334
0,420000 ; 5,09644E-11 ; 5,2976E-17 ; 1,03947068934393E-6 ; 0,480666295248909 ; -0,967617530082972 ; 1,44828382533188
0,430000 ; 4,55807E-11 ; 5,54495E-17 ; 1,21651269067829E-6 ; 0,379671224761811 ; -1,08015744458073 ; 1,45982866934254
0,440000 ; 4,10009E-11 ; 5,81682E-17 ; 1,41870544305125E-6 ; 0,322978300270101 ; -1,23360561375181 ; 1,55658391402191
0,450000 ; 3,70472E-11 ; 6,07218E-17 ; 1,63903884774018E-6 ; 0,278824251228852 ; -1,34428421262079 ; 1,62310846384965


 Input Optimum VDD : 0,42


 -- Gate: AOI31 - Optimum VDD Fine: 0,4125 --



 Circuit Optimum VDD: 0,414833333333333
```

Fig. 18 Main interface of the MERIT software tool

Note that the main interface provides the results for analysis on a single gate and on a complete circuit. Considering the analysis of a single logic gate, the main parameters simulated and post processed for analysis by the tool are presented in the main panel. The parameters are: multiple propagation delays, partial and total power-delay-product (PDP) of the gate, the value of the Energy-Delay-Product (EDP), as well as multiple values for total and partial dissipated power (static and dynamic power), and values for total and partial Energy consumption in the gate.

Considering the results of a complete logic circuit, full analysis on the circuit is done by running multiple partial tests on each of the different logic gates that compose the circuit, and the obtained results regarding energy efficiency, the partial results and the circuit results, are presented in the main panel. The optimum power-supply voltage value to achieve the best compromise in power and performance for the circuit is presented at the end of all simulations and calculus.

4.3. SINGLE LOGIC GATE SIMULATION

Every single logic gate test option instantiates HSPICE simulations one or multiple times, automatically changing several parameters for each of the simulations running. Results returned by spice simulator are obtained and processed by the MERIT tool. Simulation tests performed by the tool over a single logic gate allow for complete characterization of the gate regarding logic operation, delay performance, energetic profile, as well as important defining parameters such as PDP and EDP.

Concerning individual logic gates, currently the software tool provides an important set of logic simulation tests such as:

- **Truth table** – running the truth table option allows for the display and record of the full truth table of a logic gate
- **Logic Test** – the logic test option runs simulation of the current logic gate for the current Vdd comparing Vout values obtained from different sets of inputs with the expected result from the gate’s truth table.
- **Delay Test** – the delay test is designed to obtain by simulation the propagation times of the gate (tphl and tplh) both for each of the different gate inputs, the propagation times are also calculated for SS and FF conditions.
- **Energy Test** – the energy test is designed to obtain by simulation energy results for a logic gate, results are comprised of dynamic, static and total energy for each of the different input transitions, the values are also calculated for SS and FF conditions.
- **PDP Test** – the Power Delay Product runs sequential simulations to obtain the PDP parameter of each of the input transitions of a logic gate (PDP rise and PDP fall), this calculation is repeated for SS and FF conditions.

- **Full Test** – the full test option runs all of the above logic tests sequentially, obtaining results and conclusions, it is then possible to check for operation and logic validity, besides checking max delay and energy efficiency.

4.3.1. TRUTH TABLE TEST OPTION

Running the "truth table option" for a single gate allows for the display and record of the full truth table of a logic gate. In the following, an example of result is presented when running this option on an AND-OR-INVERTER (AOI210) gate:

```
Truth Table for :AOI210

Line:0 - A: 0 - B: 0 - C: 0 - Q: 1
Line:1 - A: 0 - B: 0 - C: 1 - Q: 0
Line:2 - A: 0 - B: 1 - C: 0 - Q: 1
Line:3 - A: 0 - B: 1 - C: 1 - Q: 0
Line:4 - A: 1 - B: 0 - C: 0 - Q: 1
Line:5 - A: 1 - B: 0 - C: 1 - Q: 0
Line:6 - A: 1 - B: 1 - C: 0 - Q: 0
Line:7 - A: 1 - B: 1 - C: 1 - Q: 0

RESULTS SAVED TO: /results/truth_table_results.dat
```

4.3.2. LOGIC TEST OPTION

The "logic test option" for a single gate performs Spice simulations for the current logic gate, comparing Vout values obtained from different sets of input transitions, with the expected result from the gate's truth table.

This test effectively allows checking the logic function of the gate by simulating the operational result for a given set of parameters like: *Vdd* and *freq*. The following excerpt is an output example of the results obtained when running this option for an AND-OR-INEVRTER (AOI210) logic gate:

```
Test:1 -  Input -  A: 0  - Input B: 0  - Input C: 0; - Output A: 1
----
 Simulation Results - Logic
Simulation Result - Q:1
Database Result - Q:1
Logic Gate State Test Correct
 MTO:     1.1000          27.0000          1.0000
----
Test:2 -  Input -  A: 0  - Input B: 0  - Input C: 1; - Output A: 0
----
 Simulation Results - Logic
Simulation Result - Q:0
Database Result - Q:0
Logic Gate State Test Correct
 MTO:   6.373e-05         27.0000          1.0000
----
RESULTS SAVED TO: /results/logic_results.dat
```

4.3.3. DELAY TEST OPTION

Running the “delay test” option allows to obtain, by simulation, the propagation delays of the gate ($t_{\mathrm{p\,HL}}$ and $t_{\mathrm{p\,LH}}$), both for each of the different gate inputs.

As we can see from the following example, the simulator considers vector pairs to calculate each propagation delay. A vector pair used in simulation represents the change of an input that affects a change in the output. So, the propagation delays are calculated twice for each of the vector, once for $t_{\mathrm{p\,HL}}$ and another for $t_{\mathrm{p\,LH}}$.

As mentioned before the simulations are repeated for extreme adverse conditions and design corners, Slow-Slow (SS) and Fast-Fast (FF), and also for nominal conditions (NC). The following snippet is an abbreviated version of a result output for the “Delay Test” option on an AND-OR-INVERTER (AOI20) gate:

```
Algorithm for A to Q.
```

```
A to Q vector pair:
Vector 1: 0101
Vector 2: 1100
----
 -- Simulation Results - Delays: tpHL + tpLH --
 -- NOM Delays --
Propagation Time Low to High (tpLH): 1,02493E-11
Propagation Time High to Low (tpHL): 1,11009E-11
 -- SS Delays --
 tpLH ss : 22.421399p
 tpHL ss: 17.012734p
 -- FF Delays --
 tpLH ff : 6.674136p
 tpHL ff : 9.605718p
----
Algorithm for B to Q.
B to Q vector pair:
Vector 1: 1001
Vector 2: 1100
----
 -- Simulation Results - Delays: tpHL + tpLH --
 -- NOM Delays --
Propagation Time Low to High (tpLH): 1,23963E-11
Propagation Time High to Low (tpHL): 1,24703E-11
 -- SS Delays --
 tpLH ss : 27.236519p
 tpHL ss: 19.332825p
 -- FF Delays --
 tpLH ff : 8.146662p
 tpHL ff : 10.327933p
```

4.3.4. ENERGY TEST OPTION

Running the “energy test” option allows to obtain, by simulation, the energy results for a logic gate. These results are comprised for dynamic, static and total energy, for each of the different input-to-output transitions.

As we can see from the following example, the simulator also considers vector pairs to calculate the energy results, where a vector pair represents the change of an input that produces a change in the output. Therefore, the energy results are calculated three times for each of the vector pairs: for static energy, dynamic energy, and total energy. This energy results are associated with each particular input-to-output transition. Moreover, simulations for the 3 design corners are also presented: SS, FF and NC.

In the following is an abbreviated result output for the "Energy Test" option on an AND-OR-INVERTER (AOI20) gate:

```
Algorithm for A to Q.
A to Q vector pair:
Vector 1: 0101
Vector 2: 1100
----
Simulation Results - Energy
 -- NOM Energy --
Static Energy: 4,38278E-14
Dynamic Energy: 2,33419E-15
Total Energy: 4,83246E-14
 -- SS Energy --
 sta_nrg=  50.392519f
 dyn_nrg=  -1.500248f
 tot_nrg= -59.437343f
 -- FF Energy --
 sta_nrg=  79.628739f
 dyn_nrg=  -3.489847f
 tot_nrg= -92.524924f
----
Algorithm for B to Q.
B to Q vector pair:
Vector 1: 1001
Vector 2: 1100
----
Simulation Results - Energy
 -- NOM Energy --
Static Energy: 3,58206E-14
Dynamic Energy: 2,72017E-15
Total Energy: 4,1128E-14
 -- SS Energy --
 sta_nrg=  43.539352f
 dyn_nrg=  -1.766101f
 tot_nrg= -47.452156f
 -- FF Energy --
 sta_nrg=  59.173846f
 dyn_nrg=  -3.983849f
 tot_nrg= -52.620121f
```

4.3.5. PDP Test Option

Running “PDP test” option allows to run sequential simulations to obtain the PDP parameter of each of the input transitions in a logic gate (PDP rise and PDP fall). Results comprised for ‘PDP Rise’ are associated with an output transition from low to high, while ‘PDP Fall’ are associated with an output transition from high to low, and these transitions are considered for each of the different input-to-output paths.

Also, the simulator runs an algorithm to generate the vector pairs to improve calculations for the PDP results, where a vector pair represents the change of an input that produces a change at the output. The PDP results are calculated twice for each of the vector pairs, once for the LH transition and another for the HL transition, both associated with each particular input-to-output transition. And again, the simulations are also repeated for SS, FF, and NC conditions.

In the following is a result output for the “PDP Test” option executed on an AND-OR-INVERTER (AOI20) gate:

```
Algorithm for A to Q.
A to Q vector pair:
Vector 1: 0101
Vector 2: 1100
----
Simulation Results - Power Delay Product (PDP)
 -- NOM PDP --
PDP - Output rise event: 1,81234E-15
PDP - Output fall event: 5,2185E-16
 -- SS PDP --
 pdp_energy_rise=  -1.156502f
 pdp_energy_fall=-343.745963a
 -- FF PDP --
 pdp_energy_rise=  -2.674636f
 pdp_energy_fall=-815.211727a
----
Algorithm for B to Q.
B to Q vector pair:
Vector 1: 1001
Vector 2: 1100
----
Simulation Results - Power Delay Product (PDP)
 -- NOM PDP --
PDP - Output rise event: 1,96441E-15
PDP - Output fall event: 7,55768E-16
 -- SS PDP --
 pdp_energy_rise=  -1.258453f
 pdp_energy_fall=-507.648069a
 -- FF PDP --
 pdp_energy_rise=  -2.859548f
 pdp_energy_fall=  -1.124300f
```

4.3.6. FULL TEST OPTION

The "full test" option runs all of the previously described logic gate tests, sequentially, and obtains the results and the conclusions. It completely verifies operation and logic validity of the gate, by direct analysis of the truth table and logic test results obtained from the simulator. This option allows also the possibility to verify maximum allowed gate transition delays, analyzing the several delay tests available.

Most importantly, it is possible to obtain and verify detailed results for power-delay-products on a single gate, as well as detailed energy efficiency reports and result analysis for all tests, which allows to complete characterize the single gate.

As for all other tests, also the full logic test results are saved on disk in a detailed data file, for log information and external usage

4.4. COMPLETE CIRCUIT SIMULATION

The “Complete Circuit Test” option executes multiple sequential and partial tests on all the logic gates that compose a circuit or sub-circuit, to obtain general results and conclusions for the overall operation of the circuit. The main objective of the complete circuit test is to automatically and efficiently apply the proposed energy efficiency methodology and clearly identify the optimum operating point for the power-supply voltage that minimizes energy requirements, while not jeopardizing performance.

The activation of the method over the considered circuit runs multiple simulations over all components in the circuit. For each of the logic gates present in the circuit, the method tries to calculate its best V_{DDopt}, also called as the optimum operation point. To achieve the value of V_{DDopt} for a single logic gate, the method runs multiple sweep simulations with small step variations on V_{DD} for each of the input-to-output possible transitions of the gate. For each small V_{DD} step, simulations values are obtained and calculated for Delay, PDP, EDP, Energy, as well as multiple derivative values and the final V_{DDopt} value. Therefore, the best value for V_{DD} is chosen for each input-to-output transition of the gate.

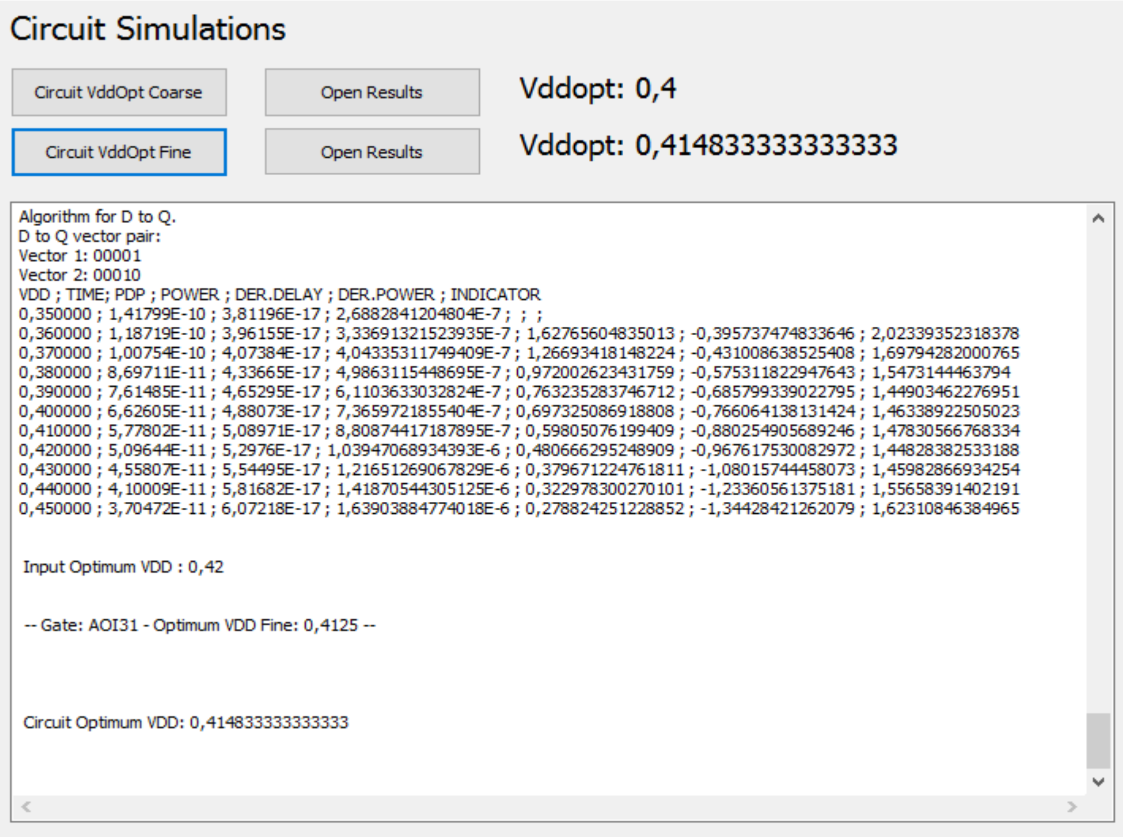
Considering all the V_{DDopt} values for the different input-to-output transitions of the gate and also considering the minimum V_{DD} values for a proper operation, the method calculates a weighted mean for the final V_{DDopt} value of the gate.

Once chosen, the V_{DDopt} values for each logic gate, several simulation tests are performed to ensure correct operation of the gate at proposed V_{DDopt} value. For testing of validity of operation at proposed V_{DDopt} value, tests for logic compliance with the truth table are performed. Also for validity of operation, a test of signal quality is also applied to each of the gates running at the proposed V_{DDopt} value.

Moreover, when calculated a V_{DDopt} value, if it has a low precision, the method of obtaining V_{DDopt} for the gate is repeated, and MERIT tool reduces the simulation sweep steps around the calculated value. This way a more precise V_{DDopt} value can be achieved for the gate.

The algorithm that serves as a basis for the proposed methodology gathers all the V_{DDopt} values for all the logic gates that constitute the circuit and, again, applies a carefully designed weighted mean on all the V_{DDopt} values to obtain the final V_{DDopt} value for the circuit.

The following Fig. 19 presents a partial result detail on the interface of the tool designed to trigger the method and apply it for a complete circuit simulation, in order to obtain the circuit V_{DDopt}.



Vddopt: 0,4

Vddopt: 0,414833333333333

```
Algorithm for D to Q.
D to Q vector pair:
Vector 1: 00001
Vector 2: 00010
VDD ; TIME; PDP ; POWER ; DER.DELAY ; DER.POWER ; INDICATOR
0,350000 ; 1,41799E-10 ; 3,81196E-17 ; 2,6882841204804E-7 ;  ;  ;
0,360000 ; 1,18719E-10 ; 3,96155E-17 ; 3,33691321523935E-7 ; 1,62765604835013 ; -0,395737474833646 ; 2,02339352318378
0,370000 ; 1,00754E-10 ; 4,07384E-17 ; 4,04335311749409E-7 ; 1,26693418148224 ; -0,431008638525408 ; 1,69794282000765
0,380000 ; 8,69711E-11 ; 4,33665E-17 ; 4,9863115448695E-7 ; 0,972002623431759 ; -0,575311822947643 ; 1,5473144463794
0,390000 ; 7,61485E-11 ; 4,65295E-17 ; 6,1103633032824E-7 ; 0,763235283746712 ; -0,685799339022795 ; 1,44903462276951
0,400000 ; 6,62605E-11 ; 4,88073E-17 ; 7,3659721855404E-7 ; 0,697325086918808 ; -0,766064138131424 ; 1,46338922505023
0,410000 ; 5,77802E-11 ; 5,08971E-17 ; 8,80874417187895E-7 ; 0,59805076199409 ; -0,880254905689246 ; 1,47830566768334
0,420000 ; 5,09644E-11 ; 5,2976E-17 ; 1,03947068934393E-6 ; 0,480666295248909 ; -0,967617530082972 ; 1,44828382533188
0,430000 ; 4,55807E-11 ; 5,54495E-17 ; 1,21651269067829E-6 ; 0,379671224761811 ; -1,08015744458073 ; 1,45982866934254
0,440000 ; 4,10009E-11 ; 5,81682E-17 ; 1,41870544305125E-6 ; 0,322978300270101 ; -1,23360561375181 ; 1,55658391402191
0,450000 ; 3,70472E-11 ; 6,07218E-17 ; 1,63903884774018E-6 ; 0,278824251228852 ; -1,34428421262079 ; 1,62310846384965


 Input Optimum VDD : 0,42


 -- Gate: AOI31 - Optimum VDD Fine: 0,4125 --



 Circuit Optimum VDD: 0,414833333333333
```

Fig. 19 Detail on the tool interface for complete circuit simulation

4.4.1. OBTAINING VDDOPT FOR A CIRCUIT

In the following, a detailed example is presented for method application to a small circuit, which include the following logic gates: NOR20; CLKIN0; OAI210; NAND20; AOI310. For simplicity, only the NAND20 logic gate complete results are presented in this example, while information of the other gates is abbreviated:

Gate: NOR20

```
…
-- Gate: NOR2 - Optimum VDD Fine: 0,405 --

## Gate: CLKIN0 ##
…
-- Gate: CLKIN - Optimum VDD Fine: 0,44 --

## Gate: OAI210 ##
…
-- Gate: OAI21 - Optimum VDD Fine: 0,416666666666667 --

## Gate: NAND20 ##
Algorithm for A to Q.
A to Q vector pair:
Vector 1: 011
Vector 2: 110
VDD ; TIME; PDP ; POWER ; DER.DELAY ; DER.POWER ; INDICATOR
0,350000 ; 1,33922E-10 ; 2,68297E-17 ; 2,00338256597124E-7 ;  ;  ;
0,360000 ; 1,10461E-10 ; 2,78754E-17 ; 2,52355129864839E-7 ; 1,7518406236466 ; -0,398851017285546 ; 2,15069164093215
0,370000 ; 9,52654E-11 ; 2,97364E-17 ; 3,1214270868542E-7 ; 1,13466047400726 ; -0,458434641215329 ; 1,59309511522259
0,380000 ; 8,09428E-11 ; 3,11782E-17 ; 3,85188058727892E-7 ; 1,06947327548872 ; -0,560091569181284 ; 1,62956484467
0,390000 ; 7,13706E-11 ; 3,33503E-17 ; 4,67283447245785E-7 ; 0,714759337524828 ; -0,629484764612646 ; 1,34424410213747
0,400000 ; 6,32308E-11 ; 3,568E-17 ; 5,64281963853059E-7 ; 0,607801556129687 ; -0,743757834595032 ; 1,35155939072472
0,410000 ; 5,40354E-11 ; 3,688E-17 ; 6,82515536111512E-7 ; 0,686623556995863 ; -0,906582376258604 ; 1,59320593325447
0,420000 ; 4,79553E-11 ; 3,89153E-17 ; 8,1149111777009E-7 ; 0,454003076417616 ; -0,988949137422426 ; 1,44295221384004
0,430000 ; 4,25289E-11 ; 4,0769E-17 ; 9,58618727500594E-7 ; 0,405191081375726 ; -1,12813387513285 ; 1,53332495650858
0,440000 ; 3,80743E-11 ; 4,27532E-17 ; 1,12288866768398E-6 ; 0,332626454204686 ; -1,2595765303764 ; 1,59220298458108
0,450000 ; 3,4429E-11 ; 4,49012E-17 ; 1,30416799790874E-6 ; 0,272195755738415 ; -1,3899998352624 ; 1,66219559100081

 Input Optimum VDD : 0,39

Algorithm for B to Q.
B to Q vector pair:
Vector 1: 101
Vector 2: 110
VDD ; TIME; PDP ; POWER ; DER.DELAY ; DER.POWER ; INDICATOR
0,350000 ; 1,57021E-10 ; 3,49016E-17 ; 2,22273453869228E-7 ;  ;  ;
0,360000 ; 1,31726E-10 ; 3,5794E-17 ; 2,71730713754308E-7 ; 1,61093102196521 ; -0,346303743904569 ; 1,95723476586978
0,370000 ; 1,12052E-10 ; 3,76014E-17 ; 3,35570984899868E-7 ; 1,25295342661173 ; -0,447014754981591 ; 1,69996818159332
0,380000 ; 9,91519E-11 ; 4,16459E-17 ; 4,20021199795465E-7 ; 0,821552531190096 ; -0,59132725225469 ; 1,41287978344479
0,390000 ; 8,39404E-11 ; 4,3251E-17 ; 5,1525844527784E-7 ; 0,968755771521007 ; -0,666858915078189 ; 1,6356146865992
0,400000 ; 7,32499E-11 ; 4,56225E-17 ; 6,22833614789918E-7 ; 0,68083250011145 ; -0,753250059541598 ; 1,43408255965305
0,410000 ; 6,32957E-11 ; 4,55326E-17 ; 7,19363242684732E-7 ; 0,633940683093344 ; -0,675908281521543 ; 1,30984896461489
0,420000 ; 5,62959E-11 ; 4,80921E-17 ; 8,54273579425855E-7 ; 0,445787506129753 ; -0,944653116922289 ; 1,39044062305204
0,430000 ; 5,10447E-11 ; 5,34564E-17 ; 1,04724682484176E-6 ; 0,334426605358519 ; -1,35121431143219 ; 1,6856409167907
0,440000 ; 4,56718E-11 ; 5,60779E-17 ; 1,2278451911245E-6 ; 0,342177161016679 ; -1,26456440433793 ; 1,60674156535461
0,450000 ; 4,11158E-11 ; 5,87194E-17 ; 1,42814684379241E-6 ; 0,290152272625954 ; -1,40252841322684 ; 1,6926806858528

 Input Optimum VDD : 0,41
 -- Gate: NAND2 - Optimum VDD Fine: 0,4 --

## Gate: AOI310 ##
...
 -- Gate: AOI31 - Optimum VDD Fine: 0,4125 --

 Circuit Optimum VDD: 0,414833333333333
```

5. SIMULATION RESULTS

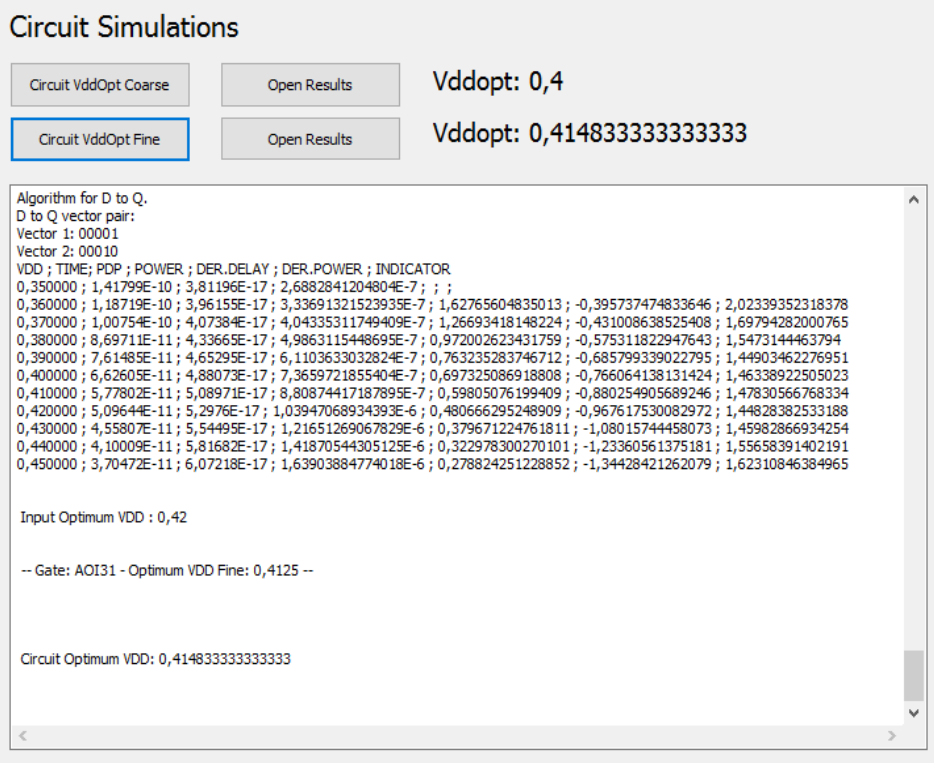
In this chapter, the simulation results are presented, both for SPICE results and for MERIT results.

5.1. SPICE SIMULATION RESULTS

Spice simulation results are presented for a Predictive Technology Model 65nm CMOS technology [37], with V_{DD} =1.1V and T=27ºC for nominal conditions (NC).

Moreover, test circuits used are: 7 ITC’99 benchmark circuits (B01, B02, B03, B06, B08, B09 and B10) [38], and PM4-2, which is a Pipeline Multiplier with 4 bits and 2 balanced pipeline stages, 9 inputs (8 data and 1 clock), 8 data outputs.

The detailed results obtained from the SPICE simulations are presented in the MERIT software tool, as the following Fig. 20 shows (for a given benchmark test circuit).



```
Algorithm for D to Q.
D to Q vector pair:
Vector 1: 00001
Vector 2: 00010
VDD ; TIME; PDP ; POWER ; DER.DELAY ; DER.POWER ; INDICATOR
0,350000 ; 1,41799E-10 ; 3,81196E-17 ; 2,6882841204804E-7 ;  ;  ;
0,360000 ; 1,18719E-10 ; 3,96155E-17 ; 3,33691321523935E-7 ; 1,62765604835013 ; -0,395737474833646 ; 2,02339352318378
0,370000 ; 1,00754E-10 ; 4,07384E-17 ; 4,04335311749409E-7 ; 1,26693418148224 ; -0,431008638525408 ; 1,69794282000765
0,380000 ; 8,69711E-11 ; 4,33665E-17 ; 4,9863115448695E-7 ; 0,972002623431759 ; -0,575311822947643 ; 1,5473144463794
0,390000 ; 7,61485E-11 ; 4,65295E-17 ; 6,1103633032824E-7 ; 0,763235283746712 ; -0,685799339022795 ; 1,44903462276951
0,400000 ; 6,62605E-11 ; 4,88073E-17 ; 7,3659721855404E-7 ; 0,697325086918808 ; -0,766064138131424 ; 1,46338922505023
0,410000 ; 5,77802E-11 ; 5,08971E-17 ; 8,80874417187895E-7 ; 0,59805076199409 ; -0,880254905689246 ; 1,47830566768334
0,420000 ; 5,09644E-11 ; 5,2976E-17 ; 1,03947068934393E-6 ; 0,480666295248909 ; -0,967617530082972 ; 1,44828382533188
0,430000 ; 4,55807E-11 ; 5,54495E-17 ; 1,21651269067829E-6 ; 0,379671224761811 ; -1,08015744458073 ; 1,4598286693425 4
0,440000 ; 4,10009E-11 ; 5,81682E-17 ; 1,41870544305125E-6 ; 0,322978300270101 ; -1,23360561375181 ; 1,55658391402191
0,450000 ; 3,70472E-11 ; 6,07218E-17 ; 1,63903884774018E-6 ; 0,278824251228852 ; -1,34428421262079 ; 1,62310846384965


 Input Optimum VDD : 0,42


 -- Gate: AOI31 - Optimum VDD Fine: 0,4125 --



 Circuit Optimum VDD: 0,414833333333333
```

Fig. 20. MERIT detailed window for test circuit V_{DDopt}.

Regarding simulations of a single logic gate the first result shown here is for an Or-And-Inverter gate with 3 inputs (OAI21) shown on Fig. 21 and Fig. 22. Let us consider for analysis an event on input A and the A-OUT path.

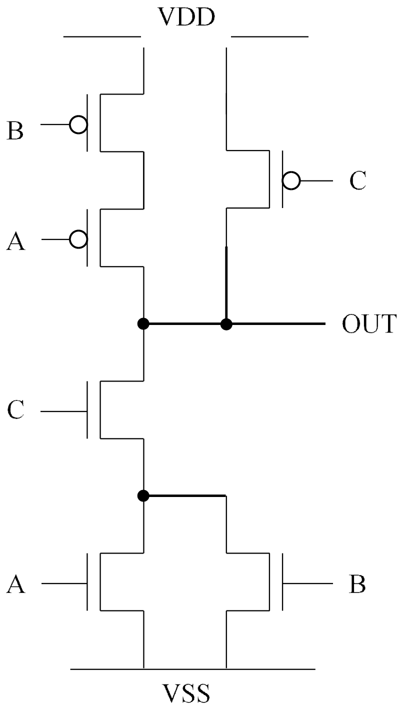


Fig. 21. Or-And-Inverter (OAI21) gate for example, transistor detail.

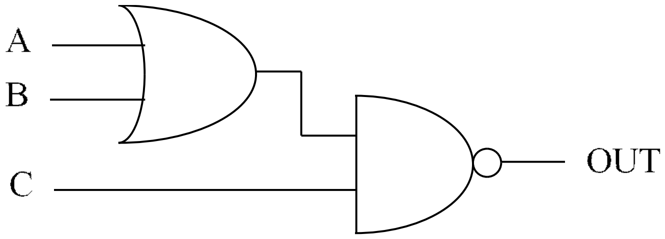


Fig. 22. Or-And-Inverter (OAI21) gate for example, logic detail.

Fig. 23 shows an example of PDP analysis and graphs, generated by the tool for the test gate OAI21, input A. Note that the coarse analysis has V_{DD} changing from 0.1V to 1.1V, while in the fine analysis V_{DD} changes from 0.1V to 0.2V. Note also that the parameter product of power x delay on both cases show an increasing trend.

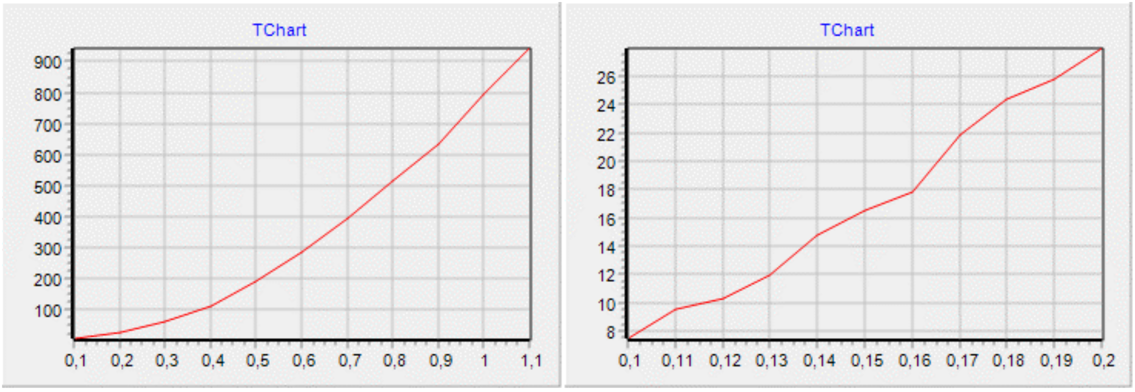


Fig. 23. PDP for OAI21 gate, input A, coarse and fine analysis.

Fig. 24 represents the EDP (energy delay product) analysis for the mentioned test gate, both for the coarse and fine analysis (on the left and on the right, respectively), also in this case both charts were generated by the tool from simulation values obtained and post processed to calculate this parameter.

Note that also for this case the coarse analysis has V_{DD} changing from 0.1V to 1.1V, (left chart) while in the fine analysis V_{DD} changes from 0.1V to 0.2V (right chart)

Note also that the parameter product of energy × delay on both cases show a decreasing trend, therefore this parameter provides for proper characterization on energy profile of a given component as well as a good comparison method between components.

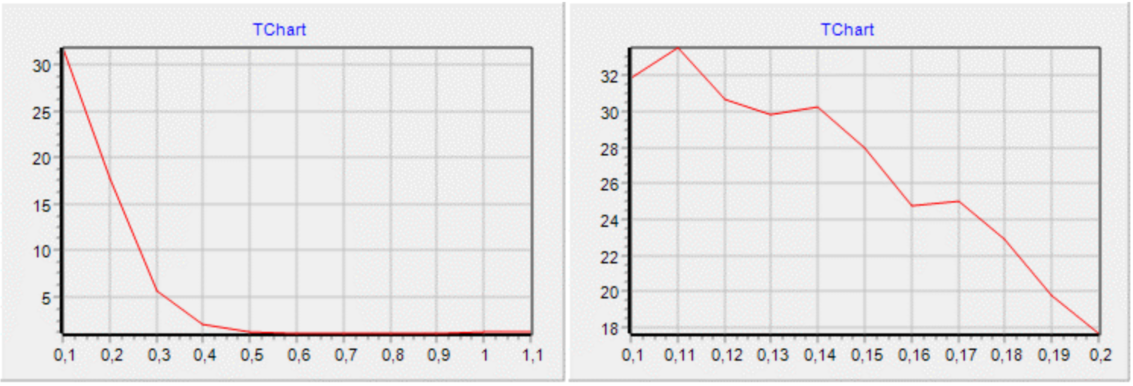


Fig. 24. EDP for OAI21 gate, input A, coarse and fine analysis.

5.2. MERIT SIMULATION RESULTS

The following Table 1 presents a resume and all simulation values and calculations done for finding the V_{DDopt} in a coarse analysis for the same OAI21 gate. Note that the | Slopes' distance| column represents the final results where V_{DD} is chosen. In this case, the minimum value is obtained for 0.3V, which is the result of this analysis.

Table 5. V_{DDopt} calculation for OAI21 gate, input A (coarse analysis).

VDD	Delay	PDP	Power	EDP	|Delay's slope|	|Power's slope|	|Slopes' Distance|
0,1	3,96E-08	7,44E-18	1,88E-10	1,39505E-27	8,455811958	0,000587228	8,45522473
0,2	6,12E-09	2,8E-17	4,57E-09	1,27977E-25	-1,32726227	0,009232557	1,336494823
0,3	8,6E-10	6,32E-17	7,35E-08	4,64692E-24	-0,17489496	0,080949556	0,255844521
0,4	1,67E-10	1,13E-16	6,78E-07	7,65803E-23	-0,026999	0,342657688	0,369656691
0,5	5,96E-11	1,93E-16	3,24E-06	6,242E-22	-0,00660079	0,713824955	0,720425745
0,6	3,34E-11	2,86E-16	8,57E-06	2,45271E-21	-0,00238582	1,062688317	1,065074141
0,7	2,4E-11	3,95E-16	1,65E-05	6,52547E-21	-0,00124589	1,423577562	1,424823452
0,8	1,9E-11	5,16E-16	2,71E-05	1,40046E-20	-0,00073995	1,6522298	1,652969753
0,9	1,61E-11	6,35E-16	3,95E-05	2,50671E-20	-0,000505	2,232071165	2,232576168
1	1,41E-11	7,91E-16	5,61E-05	4,43994E-20	-0,00037217	2,48215605	2,482528222
1,1	1,26E-11	9,42E-16	7,47E-05	7,03322E-20	-	-	-

Considering now the application to complete circuits, Table 2 resumes the results on the 8 test circuits used. The V_{DDopt} obtained depends on the gates used and on the library itself. As previously mentioned, a 65nm CMOS technology was used and all the V_{DDopt} results are, approximately 0.4V. Moreover, V_{DDopt} are presented for both coarse and fine analysis. First, the coarse V_{DDopt} is obtained, and this value is used to trigger several simulations in the proximity of this V_{DDopt} coarse value, to find the fine V_{DDopt} value.

Table 6. MERIT tool results for optimum VDD in benchmark circuits.

Circuit Name	# Gates	VDDopt Coarse (V)	VDDopt Fine (V)
b01	31	0.400224375660014	0.415341675992641
b02	16	0.400354092253938	0.412577419283703
b03	59	0.401047906759778	0.414314467549099
b06	42	0.400289020834876	0.416428075588673
b08	87	0.400332728390883	0.415603265123358
b09	96	0.400449522861927	0.416402218448702
b10	91	0.400250624354835	0.420715182574678
pipeline_multiplier_4_2	87	0.392307692307692	0.390725153846154

As transistor-level simulators such as SPICE take a long time to simulate ultra-low voltage circuits, a last information is given about the system used in the simulations and on the simulation times. Table 3 presents the duration of each simulation analysis for all benchmark circuits. The system used to perform all simulations is characterized as follows: hardware with an Intel Core i7-4790 CPU @ 3.60GHz, with 8.00GB RAM; software with Windows 10 Pro 64 Bit OS, MERIT (cMos EneRgy sImulaTor) tool ver. 1.3 developed with Embarcadero RAD Studio XE8 Version 22.0.19, instantiating Synopsys HSPICE for Windows ver. A-2008.03 for the low-level circuits' simulations.

Table 7. MERIT tool simulation times for each benchmark circuit.

Circuit Name	# Gates	Coarse analysis time duration (s)	Fine analysis time duration (s)
b01	31	94,544849	102,53379
b02	16	62,286182	66,734428
b03	59	261,350067	279,981292
b06	42	145,563385	156,396804
b08	87	310,610015	339,194549
b09	96	347,079803	373,437103
b10	91	311,065063	373,437103
pipeline_multiplier_4_2	87	405,170532	424,28247

6. CONCLUSIONS AND FUTURE WORK

In this work, a new power and performance trade-off analysis was presented, to find the optimum power-supply voltage where the best compromise between power and performance is achieved. The presented methodology is based on the growth of power and delay over the possible V_{DD} values for each cell in a circuit.

This growth, in this case, is obtained from the derivative function of power and delay functions in respect to V_{DD}. As these two functions have contradictory growth over V_{DD}, the minimum distance between these two functions gives us the optimum balance between power and performance, allowing to obtain the optimum V_{DD} value where a considerable reduction in power is achieved, while the delay of the gate is not extremely enhanced.

Moreover, a new software tool to perform power and delay analysis over a gate or circuit, in respect to V_{DD}, was also presented. The MERIT software tool, based on HSPICE simulations, obtains automatically the optimum V_{DD} value for the best power-delay trade-off, using the previous algorithm for power-delay analysis.

The work was mainly focused on the study of individual logic cells operation at low levels of supply voltages, trying to characterize the behavior of each cells and trying to establish its performance and energy profile for different (and progressively lower) power-supply voltage levels (V_{DD}). To achieve this objective, a complete logic cell library of 65nm technology was used when running simulations.

6.1. CONCLUSIONS

Running multiple progressive simulations on several components of the cell library for different progressive V_{DD} supply voltage values was possible to establish multiple performance curves based on cell internal propagation times for progressively lower sweeps on V_{DD} supply values. Similarly, running multiple progressive simulations on components of the cell library for progressively lower V_{DD} supply voltages was possible to obtain multiple energy values for analysis regarding the progression of energy used by the cell in operation for different V_{DD} values,

To completely understand a logic cell energetic profile, as well as its complete energetic requirements for each supply level, multiple parameters were obtained by simulation or calculated by post processing simulated values. Parameters such as dynamic energy, static energy, total energy, power delay product, and energy delay product, were used to draw energy curves specific for each of the logic cells on a wide span of possible supply voltages.

By manual and automated inspection, and with the analysis of multiple curves detailing energetic and performance profiles obtained for each of the logic cells, became progressively clearer and apparent the need to achieve a compromise point between power and performance. Through this compromise point would be possible to reduce energy consumed by the cell, while maintaining performance to acceptable levels.

By defining a methodology to measure or calculate the best compromise point between power and performance we can pursue the main objective of the research work, lowering the supply voltage to very low levels, trying to reach subthreshold operation and reducing power consumption, while still trying to maintain acceptable propagation delays, and so an acceptable performance.

The definition of this methodology is the essence of the work presented here, as it allows to describe a method of quickly calculate an optimum voltage level for the power supply (V_{DDopt}) of a digital system, maintaining logic validity, maintaining performance, minimizing power consumption and so improving energy efficiency,

This methodology, resulting from the research study presented in chapter 2, theoretically defined in chapter 3, analytically defined in chapter 3 by calculation and inspection of charts and, finally, applied to a real cell library by the automated software tool presented on chapter 4, can be applied to any existing cell library. This is an important result, as it can be used to improve energy efficiency of already existing digital circuits and cell libraries for different technologies.

One of the main requirements of the research work was the definition of a clear and simple method to improve energy efficiency, by using low supply levels in complete digital circuits and systems that could be quickly and easily applied to existing standard logic cell libraries and removing the need to define a new suitable cell library. This objective was fully accomplished, not only by defining an analytical solution but also by defining a software tool that automates the application of the methodology to any complete circuits and systems.

When reducing power consumption in digital systems by reducing supply voltage levels, it is inevitable to suffer severe losses in performance of the system. For this reason, the method takes into account the huge reduction of performance occurring below a certain supply voltage level. By simulation, it was demonstrated that below a certain level, a small reduction on power can lead to a large loss of performance. This objective of achieving a good energy efficiency without a large sacrifice on propagation delays and performance was also achieved, as the method identifies the power supply voltage where the best compromise between power and performance is obtained.

Generally, the results show that this tool and method can effectively obtain the optimum V_{DD} for each particular technology and cell library given.

Moreover, the main objective of the work, being the study of subthreshold operation of digital circuits and systems, was also fully completed. This study was performed not only theoretically over the current state-of-the-art in this field, but also in practice, through results and charts inspection, and through analytical calculation of values form practical simulations on real components of a complete existing digital cell library.

Finally, it must be mentioned as a global conclusion of this research work that the opportunity to study this new and advanced area of electronics, and also the possibility to provide research contribute to advance in such an important area (as the field of energy efficiency on digital systems) was very fulfilling. It was a very important source of motivation to progress through the work and to achieve the best principles, methods and solutions to address and solve this important problem, thus improving research in this field.

6.2. FUTURE WORK

Every research work has unfinished tasks, or its result opened new future perspectives. In the present work, the same happened, and we will summarize all future work opportunities.

Regarding unfinished tasks, all the cells of the target library should be analyzed and results for V_{DDopt} should be compared. This will reveal if similar or different V_{DDopt} are obtained among all the cells. For a good result, all the V_{DDopt} from the different cells should be similar, for a well-balanced library. For a future work, this type of analysis should be performed on a commercial cell library. Moreover, it also can help in the design on a new cell library focusing subthreshold operation.

In the same context, a future work perspective includes the analysis of different cell libraries and technologies, to compare for the same circuit, the V_{DDopt} evolution. This could reveal which technology and library would be the best solution for a circuit.

Another future application perspective for this work is to apply it in close association to recent and improved dynamic voltage and frequency scaling techniques, such as this new multi-mode DVFS technique described in [23] and [30], with predictive error detection and error tolerance features. The use of a reliability and fault-tolerant method in subthreshold operation should, certainly, improve its operation, as errors are avoided.

In fact, the development of a new and ultra-low-power IoT smart device is a long-term future work perspective, not only using the present methodology to specify the best power-supply voltage for subthreshold operation, but also including a new version of the adaptive DVFS presented in [23][30], and also performance sensors for memories (SRAM and DRAM). This would allow to define a new smart sensor device for IoT applications, with never-ending battery life.

Furthermore, real circuit tests are also very important, to validate the power and delay trade-off method with real data obtained by multiple measurements in real circuits. For this matter, a test-chip was recently produced by authors on [23] and [30] , and was designed to work at sub-threshold voltage levels. It is expected that new results could be obtained in the near future, also improving further research in this area.

REFERENCES

1. Benton H. Calhoun and Anantha Chandrakasan. Characterizing and modeling minimum energy operation for subthreshold circuits. In International Symposium on Low Power Electronics and Design, pages 90–95, 2004.
2. Benton H. Calhoun, Alice Wang, and Anantha Chandrakasan. Modeling and sizing for minimum energy operation in subthreshold circuits. IEEE Journal of Solid-State Circuits, 40(9):1778–1786, September 2005.
3. Benton H. Calhoun, Alice Wang, and Anantha Chandrakasan. Device Sizing for Minimum Operation in Subthreshold Circuits. Custom Integrated Circuits Conference, 2004. Proceed-ings of the IEEE 2004
4. Benton H. Calhoun, Alice Wang, N. Verma and Anantha Chandrakasan. SubThreshold Design: The Challenges of Minimizing Circuit Energy, Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED), October, 2006.
5. Sean Keller, Siddarth Bhargav, Chris Moore, and Alain J. Martin. Reliable Minimum Energy CMOS Circuit Design. In Vari'11: 2nd European Workshop on CMOS Variability, 2011.
6. Mohsen Radfar, Kriyang Shah, and Jugdutt Singh, "Recent Subthreshold Design Techniques," Active and Passive Electronic Components, vol. 2012, Article ID 926753, 11 pages, 2012. doi:10.1155/2012/926753
7. Kanitkar, Hrishikesh, "Subthreshold circuits: Design, implementation and application" (2008). Thesis. Rochester Institute of Technology
8. Yoo H.J. Dual vt self-timed CMOS logic for low subthreshold current multigigabit synchronous DRAM. IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, 45(9):1263–1271, September 1998.

9. S. Hanson, Mingoo Seok, D. Sylvester, and D. Blaauw. Nanometer device scaling in subthreshold logic and SRAM. IEEE Transactions on Electron Devices, 55:175–185, 2008.

10. Saurav Chakraborty, Abhijit Mallik, and Chandan Kumar Sarkar. Subthreshold performance of dual-material gate CMOS devices and circuits for ultra-low power analog/mixed-signal applications. IEEE Transactions on Electron Devices, 55(3): 827–832, March 2008.

11. Aaron V. Do, Chirn Chye Boon, Manh Anh, Kiat Seng Yeo, and Alper Cabuk. A sub-threshold low-noise amplifier optimized for ultra-low-power applications in the ISM band. IEEE Transactions on Microwave Theory and Techniques, 56(2): 286–292, February 2008.

12. G. Giustolisi, G. Palumbo, M. Criscione, and F. Cutri. A low-voltage low-power voltage reference based on subthreshold MOSFETs. IEEE Journal of Solid-State Circuits, 38(1):151–154, January 2003.

13. Jae-Joon Kim and Kaushik Roy. Double gate MOSFET subthreshold circuit for ultralow power applications. IEEE Transactions on Electron Devices, 51(9):1468–1474, September 2004.

14. Toshinori Numata and Shinichi Takagi. Device design for subthreshold slope and threshold voltage control in sub-100-nm fully depleted SOI MOSFETs. IEEE Transactions on Elec-tron Devices, 51(12):2161–2167, December 2004.

15. Li Ming-Zhong et al. "Sub-threshold standard cell library design for ultra-low power bi-omedical applications" Engineering in Medicine and Biology Society (EMBC) 2013 35th Annual International Conference of the IEEE pp. 1454 2013.

16. Alok Sahu , Geoffrey Eappen. Sub-Threshold Logic and Standard Cell Library. International Journal of Innovative Research in Science, Engineering and Technology, Vol. 3, Issue 1, January 2014

17. S.V. Kumar, C.H. Kim, S. Sapatnekar, "Adaptive Techniques for Overcoming Performance Degradation due to Aging in Digital Circuits", Proc. IEEE ASP-DAC, pp. 284-289, Jan. 2009.

18. V. Reddy, et al., “The Impact of NBTI on the Performance of Combinational and Sequential Circuits”, Proc. ACM/IEEE Design Automation Conf. (DAC), pp. 364-369, 2007.

19. J. Semião et al., “Delay-Fault Tolerance to Power Supply Voltage Disturbances Analysis in Nanometer Technologies”, Proc. IEEE Int. On-Line Testing Symp. (IOLTS), pp. 223-228, 2009.

20. K. Kang, S. Gangwal, S. Phil Park, and K. Roy, “NBTI Induced Performance Degradation in Logic and Memory Circuits: How Effectively Can We Approach a Reliability Solution?”, Proc. Asia / South Pacific Design Autom. Conf. (ASP-DAC), pp. 726-731, 2008.

21. C. V. Martins, J. Semião, J. C. Vazquez, V. Champaq, M. Santos, I. C. Teixeira, J. P. Teixeira, "Adaptive Error-Prediction Flip-flop for Performance Failure Prediction with Ag-ing Sensors", 29th IEEE VLSI Test Symposium 2011 (VTS'11), Dana Point, California, USA, May 1st - 5th, 2011.

22. C. Martins, J. Pachito, J. Semião, I. C. Teixeira, J. P. Teixeira, "Adaptive Error-Prediction Aging Sensor for On-Line Monitoring of Performance Errors", Proceedings of the 26th Conference on Design of Circuits and Integrated Systems - DCIS'2011, Albufeira, Portugal, November 16-18, 2011.

23. J. Semiao, J. Pachito, C. Martins, B. Jacinto, J. Vazquez, V. Champac, M. Santos, I. Teixeira, J. Teixeira, "Aging-aware Power or Frequency Tuning with Predictive Fault Detection", IEEE Design & Test of Computers, Volume 29 , Issue 5, September/October 2012, DOI: 10.1109/MDT.2012.2206009.

24. S. Gochman, R. Ronen, I. Anati, A. Berkovits, T. Kurts, A. Naveh, A. Saeed, Z. Sperber and R. C. Valentine, "The Intel® Pentium® M Processor: Microarchitecture and Performance," Intel® Technology Journal, vol. 7, no. 2, Mai. 2003.

25. “AMD PowerNow!™ Technology Dynamically Manages Power and Performance,” Nov. 2000. [Online]. Available at: http://support.amd.com/us/Embedded_TechDocs/24404a.pdf.

26. D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner and T. Mudge, "Razor: A Low-Power Pipeline Based on

Circuit-Level Timing Speculation," in Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on, Dez. 2003.

27. Shidhartha Das, Carlos Tokunaga, Sanjay Pant, Wei-Hsiang Ma, Sudherssen Kalaiselvan, Kevin Lai, David M. Bull, David T. Blaauw, “RazorII: In Situ Error Detection and Correction for PVT and SER Tolerance”, IEEE Journal of Solid-State Circuits, Vol. 44, No. 1, January 2009.

28. Bo Zhai, David Blaauw, Dennis Sylvester, Krisztian Flautner. “Theoretical and Practical Limits of Dynamic Voltage Scaling”, Proc. ACM/IEEE Design Automation Conference (DAC), June 2004.

29. S. Khursheed, U. Ingelsson, P. Rosinger, B.M. Al-Hashimi and P. Harrod, “Bridging Fault Test Method with Adaptive Power Management Awareness” IEEE Transactions on CAD of Integrated Circuits and Systems, vol. 27, nº. 6, June 2008, pp. 1117-1127.

30. J. Semião, R. Cabral, M. B. Santos, I. C. Teixeira, J. P. Teixeira, "Dynamic Voltage and Frequency Scaling for Long-Term and Fail-Safe Operation", The Finale Workshop on Manufacturable and Dependable Multicore Architectures at Nanoscale (MEDIAN Finale'15), Tallinn, Estonia, November 10-11, 2015.

31. Dejan Markovic, Vladimir Stojanovic, Borivoje Nikolic, Mark A. Horowitz, Robert W. Brodersen, “Methods for True Energy-Performance Optimization”, IEEE Journal of Solid-State Circuits, Vol. 39, no. 8, August 2004.

32. J. Burr and A. M. Peterson, “Ultra low power CMOS technology,” in Proc. NASA VLSI Design Symp., Oct. 1991, pp. 4.2.1–4.2.13.

33. R. Gonzalez, B. Gordon, and M. A. Horowitz, “Supply and threshold voltage scaling for low power CMOS,” IEEE J. Solid-State Circuits, vol. 32, pp. 1210–1216, Aug. 1997.

34. P. I. Penzes and A. J. Martin, “Energy-delay efficiency of VLSI computations,” in Proc. Great Lakes Symp. VLSI, Apr. 2002, pp. 104–111.

35. H. P. Hofstee, “Power-constrained microprocessor design,” in Proc. Int. Conf. Computer Design, Sept. 2002, pp. 14–16.

36. A. J. Martin, “Toward an energy complexity of computation,” Inform. Process. Lett., vol. 77, pp. 181–187, Feb. 2001.

37. Predictive Technology Model (PTM), http://www.eas.asu.edu/~ptm/.

38. S. Davidson, "ITC'99 Benchmark Circuits – Preliminary Results", Proc. IEEE International Test Conf (ITC), pp. 1125, 1999; available at http://www.cerc.utexas.edu/itc99-benchmarks/bench.html

39. Prex Embedded Real Time OS - https://www.openhub.net/p/prex

40. Geronimo, Guilherme & Werner, Jorge & Weingartner, Rafael & Westphall, Carlos & Merkle Westphall, Carla. (2014). Provisioning, Resource Allocation, and DVFS in Green Clouds. International Journal on Advances in Networks and Services. 7. 108-117.

41. Pal A. (2015) Supply Voltage Scaling for Low Power. In: Low-Power VLSI Circuits and Systems. Springer, New Delhi

42. Hartman, Mark - National Semiconductor's Advanced Power Group, "Processor Energy Savings Through Adaptive Voltage Scaling" By in March, 2008 - Portable Design.

43.Sam Naffziger, AMD Corporate Fellow / IEEE Fellow, AMD's Commitment to Accelerating Energy Efficiency - http://www.amd.com/Documents/energy-efficiency-whitepaper.pdf

44.K. Wilcox et al., "4.8 A 28nm x86 APU optimized for power and area efficiency," 2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers, San Francisco, CA, 2015, pp. 1-3.

45.Vainberg, Alex; National Semiconductor – Adaptive Voltage Scaling (AVS) - PowerWise, October 13, 2010

46.K., Hrishikesh, "Subthreshold circuits: Design, implementation and application" (2008). Rochester Institute of Technology. From http://scholarworks.rit.edu/theses/5609

47.Dehnert J.C. (2003) The Transmeta Crusoe: VLIW Embedded in CISC. In: Krall A. (eds) Software and Compilers for Embedded Systems. SCOPES 2003. Lecture Notes in Computer Science, vol 2826. Springer, Berlin, Heidelberg

48.Banerjee, A.; Calhoun, B.H. An Ultra-Low Energy Subthreshold SRAM Bitcell for Energy Constrained Biomedical Applications. J. Low Power Electron. Appl. 2014, 4, 119-137.

49.Synopsys HSPICE web page (browsed on September 28th, 2017): https://www.synopsys.com/verification/ams-verification/circuit-simulation/hspice.html

APPENDIX

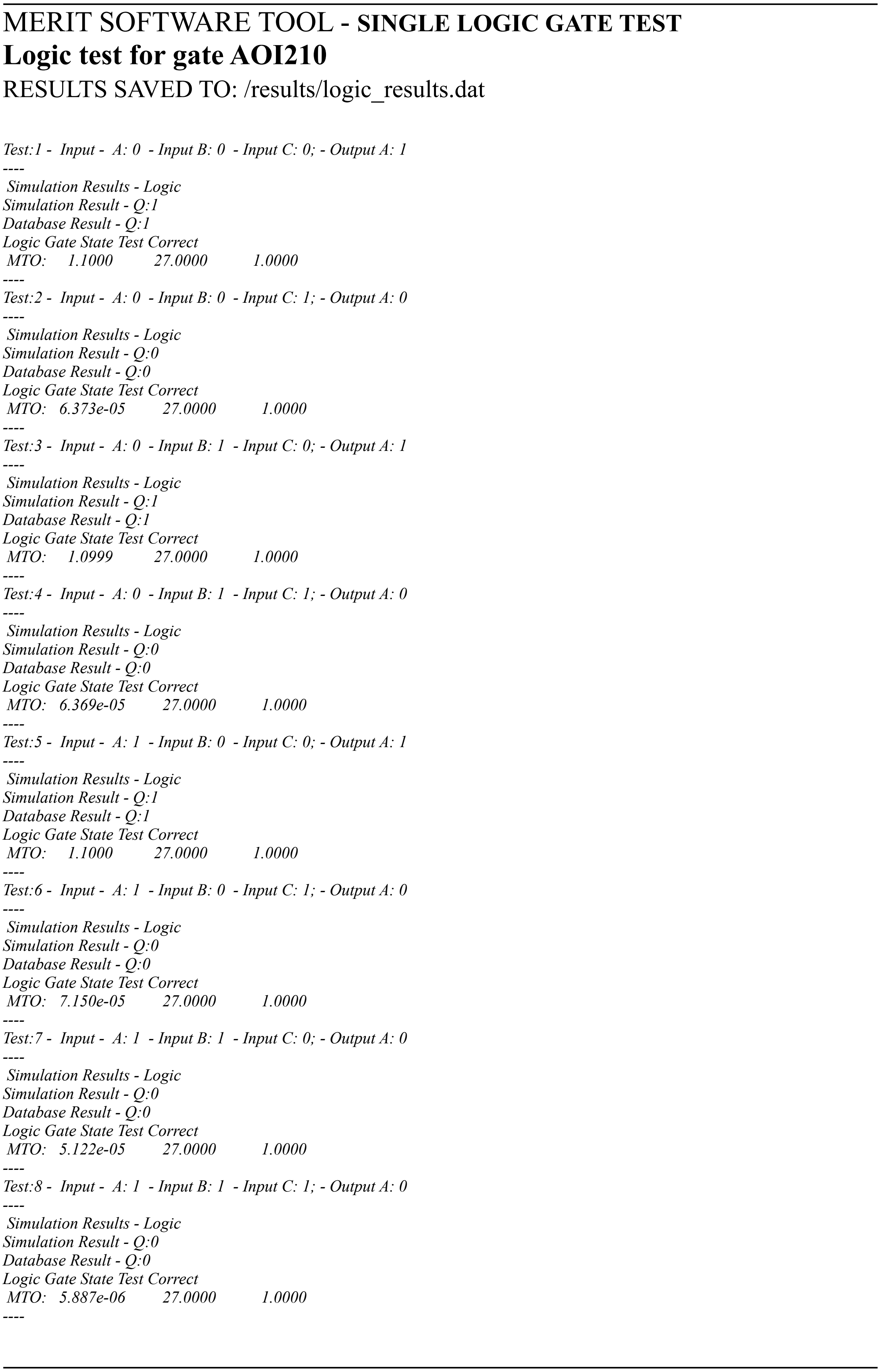
MERIT SOFTWARE TOOL - **SINGLE LOGIC GATE TEST**

Truth table test for gate AOI210

RESULTS SAVED TO: /results/truth_table_results.dat

Truth Table for :AOI210

```
Line:0  -  A: 0  -  B: 0  -  C: 0  -  Q: 1
Line:1  -  A: 0  -  B: 0  -  C: 1  -  Q: 0
Line:2  -  A: 0  -  B: 1  -  C: 0  -  Q: 1
Line:3  -  A: 0  -  B: 1  -  C: 1  -  Q: 0
Line:4  -  A: 1  -  B: 0  -  C: 0  -  Q: 1
Line:5  -  A: 1  -  B: 0  -  C: 1  -  Q: 0
Line:6  -  A: 1  -  B: 1  -  C: 0  -  Q: 0
Line:7  -  A: 1  -  B: 1  -  C: 1  -  Q: 0
```



MERIT SOFTWARE TOOL - **SINGLE LOGIC GATE TEST**

Logic test for gate AOI210

RESULTS SAVED TO: /results/logic_results.dat

```
Test:1 -  Input -  A: 0  - Input B: 0  - Input C: 0; - Output A: 1
----
 Simulation Results - Logic
Simulation Result - Q:1
Database Result - Q:1
Logic Gate State Test Correct
 MTO:     1.1000          27.0000          1.0000
----
Test:2 -  Input -  A: 0  - Input B: 0  - Input C: 1; - Output A: 0
----
 Simulation Results - Logic
Simulation Result - Q:0
Database Result - Q:0
Logic Gate State Test Correct
 MTO:   6.373e-05        27.0000          1.0000
----
Test:3 -  Input -  A: 0  - Input B: 1  - Input C: 0; - Output A: 1
----
 Simulation Results - Logic
Simulation Result - Q:1
Database Result - Q:1
Logic Gate State Test Correct
 MTO:     1.0999          27.0000          1.0000
----
Test:4 -  Input -  A: 0  - Input B: 1  - Input C: 1; - Output A: 0
----
 Simulation Results - Logic
Simulation Result - Q:0
Database Result - Q:0
Logic Gate State Test Correct
 MTO:   6.369e-05        27.0000          1.0000
----
Test:5 -  Input -  A: 1  - Input B: 0  - Input C: 0; - Output A: 1
----
 Simulation Results - Logic
Simulation Result - Q:1
Database Result - Q:1
Logic Gate State Test Correct
 MTO:     1.1000          27.0000          1.0000
----
Test:6 -  Input -  A: 1  - Input B: 0  - Input C: 1; - Output A: 0
----
 Simulation Results - Logic
Simulation Result - Q:0
Database Result - Q:0
Logic Gate State Test Correct
 MTO:   7.150e-05        27.0000          1.0000
----
Test:7 -  Input -  A: 1  - Input B: 1  - Input C: 0; - Output A: 0
----
 Simulation Results - Logic
Simulation Result - Q:0
Database Result - Q:0
Logic Gate State Test Correct
 MTO:   5.122e-05        27.0000          1.0000
----
Test:8 -  Input -  A: 1  - Input B: 1  - Input C: 1; - Output A: 0
----
 Simulation Results - Logic
Simulation Result - Q:0
Database Result - Q:0
Logic Gate State Test Correct
 MTO:   5.887e-06        27.0000          1.0000
----
```

MERIT SOFTWARE TOOL - **SINGLE LOGIC GATE TEST**

Delay test for gate AOI210

RESULTS SAVED TO: /results/delay_results.dat

```
Algorithm for A to Q.
A to Q vector pair:
Vector 1: 0101
Vector 2: 1100
----
 -- Simulation Results - Delays: tpHL + tpLH --
 -- NOM Delays --
Propagation Time Low to High (tpLH): 1,02493E-11
Propagation Time High to Low (tpHL): 1,11009E-11
 -- SS Delays --
 tpLH ss : 22.421399p
 tpHL ss: 17.012734p
 -- FF Delays --
 tpLH ff : 6.674136p
 tpHL ff : 9.605718p
----
Algorithm for B to Q.
B to Q vector pair:
Vector 1: 1001
Vector 2: 1100
----
 -- Simulation Results - Delays: tpHL + tpLH --
 -- NOM Delays --
Propagation Time Low to High (tpLH): 1,23963E-11
Propagation Time High to Low (tpHL): 1,24703E-11
 -- SS Delays --
 tpLH ss : 27.236519p
 tpHL ss: 19.332825p
 -- FF Delays --
 tpLH ff : 8.146662p
 tpHL ff : 10.327933p
----
Algorithm for C to Q.
C to Q vector pair:
Vector 1: 0001
Vector 2: 0010
----
 -- Simulation Results - Delays: tpHL + tpLH --
 -- NOM Delays --
Propagation Time Low to High (tpLH): 5,21832E-12
Propagation Time High to Low (tpHL): 7,32078E-12
 -- SS Delays --
 tpLH ss : 9.459073p
 tpHL ss: 10.541445p
 -- FF Delays --
 tpLH ff : 3.463736p
 tpHL ff : 6.348617p
----
```

MERIT SOFTWARE TOOL - **SINGLE LOGIC GATE TEST**

Energy test for gate AOI210

RESULTS SAVED TO: /results/energy_results.dat

```
Algorithm for A to Q.
A to Q vector pair:
Vector 1: 0101
Vector 2: 1100
----
Simulation Results - Energy
 -- NOM Energy --
Static Energy: 4,38278E-14
Dynamic Energy: 2,33419E-15
Total Energy: 4,83246E-14
 -- SS Energy --
 sta_nrg=  50.392519f
 dyn_nrg=  -1.500248f
 tot_nrg= -59.437343f
 -- FF Energy --
 sta_nrg=  79.628739f
 dyn_nrg=  -3.489847f
 tot_nrg= -92.524924f
----
Algorithm for B to Q.
B to Q vector pair:
Vector 1: 1001
Vector 2: 1100
----
Simulation Results - Energy
 -- NOM Energy --
Static Energy: 3,58206E-14
Dynamic Energy: 2,72017E-15
Total Energy: 4,1128E-14
 -- SS Energy --
 sta_nrg=  43.539352f
 dyn_nrg=  -1.766101f
 tot_nrg= -47.452156f
 -- FF Energy --
 sta_nrg=  59.173846f
 dyn_nrg=  -3.983849f
 tot_nrg= -52.620121f
----
Algorithm for C to Q.
C to Q vector pair:
Vector 1: 0001
Vector 2: 0010
----
Simulation Results - Energy
 -- NOM Energy --
Static Energy: 3,24411E-14
Dynamic Energy: 1,01685E-15
Total Energy: 3,66763E-14
 -- SS Energy --
 sta_nrg=  34.188389f
 dyn_nrg=-611.674266a
 tot_nrg= -39.757660f
 -- FF Energy --
 sta_nrg=  60.717128f
 dyn_nrg=  -1.606807f
 tot_nrg= -68.872529f
```

MERIT SOFTWARE - **SINGLE LOGIC GATE TEST**

Power Delay Product test for gate AOI210

RESULTS SAVED TO: /results/pdp_results.dat

```
Algorithm for A to Q.
A to Q vector pair:
Vector 1: 0101
Vector 2: 1100
----
Simulation Results - Power Delay Product (PDP)
 -- NOM PDP --
PDP - Output rise event: 1,81234E-15
PDP - Output fall event: 5,2185E-16
 -- SS PDP --
 pdp_energy_rise=  -1.156502f
 pdp_energy_fall=-343.745963a
 -- FF PDP --
 pdp_energy_rise=  -2.674636f
 pdp_energy_fall=-815.211727a
----
Algorithm for B to Q.
B to Q vector pair:
Vector 1: 1001
Vector 2: 1100
----
Simulation Results - Power Delay Product (PDP)
 -- NOM PDP --
PDP - Output rise event: 1,96441E-15
PDP - Output fall event: 7,55768E-16
 -- SS PDP --
 pdp_energy_rise=  -1.258453f
 pdp_energy_fall=-507.648069a
 -- FF PDP --
 pdp_energy_rise=  -2.859548f
 pdp_energy_fall=  -1.124300f
----
Algorithm for C to Q.
C to Q vector pair:
Vector 1: 0001
Vector 2: 0010
----
Simulation Results - Power Delay Product (PDP)
 -- NOM PDP --
PDP - Output rise event: 1,17457E-15
PDP - Output fall event: 1,57717E-16
 -- SS PDP --
 pdp_energy_rise=-703.827306a
 pdp_energy_fall=  92.153040a
 -- FF PDP --
 pdp_energy_rise=  -1.799379f
 pdp_energy_fall= 192.571951a
----
```

MERIT SOFTWARE - **SINGLE LOGIC GATE TEST**
Power Tests – PDP Sweep Coarse for gate AOI210
RESULTS SAVED TO: /results/pdp_results_higher.dat

```
Algorithm for A to Q.
A to Q vector pair:
Vector 1: 0101
Vector 2: 1100
VDD ; PDP
0,100000;6,48614E-18
0,200000;2,46765E-17
0,300000;5,43294E-17
0,400000;9,85458E-17
0,500000;1,61035E-16
0,600000;2,41159E-16
0,700000;3,25103E-16
0,800000;4,27342E-16
0,900000;5,32593E-16
1,000000;6,47131E-16
1,100000;7,55045E-16

Algorithm for B to Q.
B to Q vector pair:
Vector 1: 1001
Vector 2: 1100
VDD ; PDP
0,100000;6,30762E-18
0,200000;2,71858E-17
0,300000;6,14725E-17
0,400000;1,12076E-16
0,500000;1,84031E-16
0,600000;2,79202E-16
0,700000;3,87751E-16
0,800000;5,07971E-16
0,900000;6,35306E-16
1,000000;7,79409E-16
1,100000;9,30163E-16

Algorithm for C to Q.
C to Q vector pair:
Vector 1: 0001
Vector 2: 0010
VDD ; PDP
0,100000;2,7548E-18
0,200000;1,16807E-17
0,300000;2,57082E-17
0,400000;4,48877E-17
0,500000;6,81205E-17
0,600000;8,85073E-17
0,700000;1,05669E-16
0,800000;1,35366E-16
0,900000;1,68483E-16
1,000000;2,08049E-16
1,100000;2,49344E-16
```

MERIT SOFTWARE - **SINGLE LOGIC GATE TEST**

Power Tests – PDP Sweep Fine for gate AOI210

RESULTS SAVED TO: /results/pdp_results_lower.dat

```
Algorithm for A to Q.
A to Q vector pair:
Vector 1: 0101
Vector 2: 1100
VDD ; PDP
0,100000;6,48614E-18
0,110000;7,49802E-18
0,120000;9,25547E-18
0,130000;1,0563E-17
0,140000;1,12216E-17
0,150000;1,21783E-17
0,160000;1,37372E-17
0,170000;1,64078E-17
0,180000;1,93989E-17
0,190000;2,22258E-17
0,200000;2,46765E-17

Algorithm for B to Q.
B to Q vector pair:
Vector 1: 1001
Vector 2: 1100
VDD ; PDP
0,100000;6,30762E-18
0,110000;7,37239E-18
0,120000;9,00178E-18
0,130000;1,17193E-17
0,140000;1,46964E-17
0,150000;1,6104E-17
0,160000;1,66845E-17
0,170000;2,01854E-17
0,180000;2,22103E-17
0,190000;2,52744E-17
0,200000;2,71858E-17

Algorithm for C to Q.
C to Q vector pair:
Vector 1: 0001
Vector 2: 0010
VDD ; PDP
0,100000;2,7548E-18
0,110000;3,31768E-18
0,120000;4,53593E-18
0,130000;4,78426E-18
0,140000;4,67784E-18
0,150000;5,93119E-18
0,160000;7,4662E-18
0,170000;8,36069E-18
0,180000;1,02739E-17
0,190000;1,05749E-17
0,200000;1,16807E-17
```

MERIT SOFTWARE - **SINGLE LOGIC GATE TEST**

Energy Tests – EDP Sweep Coarse for gate AOI210

RESULTS SAVED TO: /results/edp_results_higher.dat

```
Algorithm for A to Q.
A to Q vector pair:
Vector 1: 0101
Vector 2: 1100
VDD ; EDP
0,100000;2,20329E-25
0,200000;1,34304E-25
0,300000;4,31475E-26
0,400000;1,52736E-26
0,500000;8,80423E-27
0,600000;7,2975E-27
0,700000;6,88639E-27
0,800000;7,05107E-27
0,900000;7,28022E-27
1,000000;7,60907E-27
1,100000;7,86046E-27

Algorithm for B to Q.
B to Q vector pair:
Vector 1: 1001
Vector 2: 1100
VDD ; EDP
0,100000;2,49733E-25
0,200000;1,67644E-25
0,300000;5,41605E-26
0,400000;1,92781E-26
0,500000;1,1312E-26
0,600000;9,68634E-27
0,700000;9,50651E-27
0,800000;9,82946E-27
0,900000;1,02464E-26
1,000000;1,09208E-26
1,100000;1,15694E-26

Algorithm for C to Q.
C to Q vector pair:
Vector 1: 0001
Vector 2: 0010
VDD ; EDP
0,100000;5,40179E-26
0,200000;2,99944E-26
0,300000;8,97839E-27
0,400000;3,05171E-27
0,500000;1,63526E-27
0,600000;1,15929E-27
0,700000;9,30066E-28
0,800000;9,15619E-28
0,900000;9,55084E-28
1,000000;1,05631E-27
1,100000;1,15614E-27
```

MERIT SOFTWARE - **SINGLE LOGIC GATE TEST**

Energy Tests – EDP Sweep Fine for gate AOI210

RESULTS SAVED TO: /results/edp_results_lower.dat

```
Algorithm for A to Q.
A to Q vector pair:
Vector 1: 0101
Vector 2: 1100
VDD ; EDP
0,100000;2,20329E-25
0,110000;2,15422E-25
0,120000;2,25331E-25
0,130000;2,162E-25
0,140000;1,91393E-25
0,150000;1,73701E-25
0,160000;1,62604E-25
0,170000;1,6078E-25
0,180000;1,55703E-25
0,190000;1,46671E-25
0,200000;1,34304E-25

Algorithm for B to Q.
B to Q vector pair:
Vector 1: 1001
Vector 2: 1100
VDD ; EDP
0,100000;2,49733E-25
0,110000;2,48002E-25
0,120000;2,54371E-25
0,130000;2,77704E-25
0,140000;2,87517E-25
0,150000;2,61358E-25
0,160000;2,24191E-25
0,170000;2,23952E-25
0,180000;2,03394E-25
0,190000;1,88425E-25
0,200000;1,67644E-25

Algorithm for C to Q.
C to Q vector pair:
Vector 1: 0001
Vector 2: 0010
VDD ; EDP
0,100000;5,40179E-26
0,110000;5,30669E-26
0,120000;5,93521E-26
0,130000;5,06715E-26
0,140000;4,0365E-26
0,150000;4,20615E-26
0,160000;4,28952E-26
0,170000;3,9198E-26
0,180000;3,92902E-26
0,190000;3,29477E-26
0,200000;2,99944E-26
```

MERIT SOFTWARE - **SINGLE LOGIC GATE TEST**

Delay Tests – Tp Sweep Coarse for gate AOI210

RESULTS SAVED TO: /results/time_results_higher.dat

```
Algorithm for A to Q.
A to Q vector pair:
Vector 1: 0101
Vector 2: 1100
VDD ; TIME
0,100000;3,31013E-8
0,200000;5,32613E-9
0,300000;7,59847E-10
0,400000;1,46397E-10
0,500000;5,11323E-11
0,600000;2,85652E-11
0,700000;2,03221E-11
0,800000;1,60607E-11
0,900000;1,3563E-11
1,000000;1,18587E-11
1,100000;1,06751E-11

Algorithm for B to Q.
B to Q vector pair:
Vector 1: 1001
Vector 2: 1100
VDD ; TIME
0,100000;3,70385E-8
0,200000;5,95166E-9
0,300000;8,42622E-10
0,400000;1,63506E-10
0,500000;5,79727E-11
0,600000;3,2914E-11
0,700000;2,35472E-11
0,800000;1,87587E-11
0,900000;1,58326E-11
1,000000;1,39085E-11
1,100000;1,24487E-11

Algorithm for C to Q.
C to Q vector pair:
Vector 1: 0001
Vector 2: 0010
VDD ; TIME
0,100000;1,96938E-8
0,200000;2,62924E-9
0,300000;3,46957E-10
0,400000;6,69918E-11
0,500000;2,45861E-11
0,600000;1,4435E-11
0,700000;1,05305E-11
0,800000;8,64874E-12
0,900000;7,52104E-12
1,000000;6,7945E-12
1,100000;6,30494E-12
```

MERIT SOFTWARE - SINGLE LOGIC GATE TEST

Delay Tests – Tp Sweep Fine for gate AOI210

RESULTS SAVED TO: /results/time_results_lower.dat

```
Algorithm for A to Q.
A to Q vector pair:
Vector 1: 0101
Vector 2: 1100
VDD ; TIME
0,100000;3,31013E-8
0,110000;2,81712E-8
0,120000;2,40312E-8
0,130000;2,02434E-8
0,140000;1,6807E-8
0,150000;1,40089E-8
0,160000;1,16328E-8
0,170000;9,63295E-9
0,180000;7,87718E-9
0,190000;6,46642E-9
0,200000;5,32613E-9
```

```
Algorithm for B to Q.
B to Q vector pair:
Vector 1: 1001
Vector 2: 1100
VDD ; TIME
0,100000;3,70385E-8
0,110000;3,17633E-8
0,120000;2,69302E-8
0,130000;2,26703E-8
0,140000;1,89401E-8
0,150000;1,57494E-8
0,160000;1,30109E-8
0,170000;1,07606E-8
0,180000;8,88403E-9
0,190000;7,23264E-9
0,200000;5,95166E-9
```

```
Algorithm for C to Q.
C to Q vector pair:
Vector 1: 0001
Vector 2: 0010
VDD ; TIME
0,100000;1,96938E-8
0,110000;1,63208E-8
0,120000;1,34582E-8
0,130000;1,10289E-8
0,140000;9,06416E-9
0,150000;7,38585E-9
0,160000;5,99758E-9
0,170000;4,89559E-9
0,180000;3,96506E-9
0,190000;3,22154E-9
0,200000;2,62924E-9
```

MERIT SOFTWARE - SINGLE LOGIC GATE TEST

Full test for gate AOI210

RESULTS SAVED TO: /results/complete_results.dat

```
Test:1 -  Input -  A: 0  - Input B: 0  - Input C: 0; - Output A: 1
----
 Simulation Results - Logic
Simulation Result - Q:1
Database Result - Q:1
Logic Gate State Test Correct
 MTO:     1.1000         27.0000          1.0000
----
Test:2 -  Input -  A: 0  - Input B: 0  - Input C: 1; - Output A: 0
----
 Simulation Results - Logic
Simulation Result - Q:0
Database Result - Q:0
Logic Gate State Test Correct
 MTO:   6.373e-05        27.0000          1.0000
---
Test:3 -  Input -  A: 0  - Input B: 1  - Input C: 0; - Output A: 1
----
 Simulation Results - Logic
Simulation Result - Q:1
Database Result - Q:1
Logic Gate State Test Correct
 MTO:     1.0999         27.0000          1.0000
---
Test:4 -  Input -  A: 0  - Input B: 1  - Input C: 1; - Output A: 0
----
 Simulation Results - Logic
Simulation Result - Q:0
Database Result - Q:0
Logic Gate State Test Correct
 MTO:   6.369e-05        27.0000          1.0000
----
Test:5 -  Input -  A: 1  - Input B: 0  - Input C: 0; - Output A: 1
----
 Simulation Results - Logic
Simulation Result - Q:1
Database Result - Q:1
Logic Gate State Test Correct
 MTO:     1.1000         27.0000          1.0000
---
Test:6 -  Input -  A: 1  - Input B: 0  - Input C: 1; - Output A: 0
----
 Simulation Results - Logic
Simulation Result - Q:0
Database Result - Q:0
Logic Gate State Test Correct
 MTO:   7.150e-05        27.0000          1.0000
---
Test:7 -  Input -  A: 1  - Input B: 1  - Input C: 0; - Output A: 0
----
 Simulation Results - Logic
Simulation Result - Q:0
Database Result - Q:0
Logic Gate State Test Correct
 MTO:   5.122e-05        27.0000          1.0000
---
Test:8 -  Input -  A: 1  - Input B: 1  - Input C: 1; - Output A: 0
----
 Simulation Results - Logic
Simulation Result - Q:0
Database Result - Q:0
Logic Gate State Test Correct
 MTO:   5.887e-06        27.0000          1.0000
----

Algorithm for A to Q.
A to Q vector pair:
Vector 1: 0101
Vector 2: 1100
----
```

```
 -- Simulation Results - Delays: tpHL + tpLH --
 -- NOM Delays --
Propagation Time Low to High (tpLH): 1,02493E-11
Propagation Time High to Low (tpHL): 1,11009E-11
 -- SS Delays --
 tpLH ss : 22.421399p
 tpHL ss: 17.012734p
 -- FF Delays --
 tpLH ff : 6.674136p
 tpHL ff : 9.605718p
----
----
Simulation Results - Energy
 -- NOM Energy --
Static Energy: 4,38278E-14
Dynamic Energy: 2,33419E-15
Total Energy: 4,83246E-14
 -- SS Energy --
 sta_nrg=  50.392519f
 dyn_nrg=  -1.500248f
 tot_nrg= -59.437343f
 -- FF Energy --
 sta_nrg=  79.628739f
 dyn_nrg=  -3.489847f
 tot_nrg= -92.524924f
----
----
Simulation Results - Power Delay Product (PDP)
 -- NOM PDP --
PDP - Output rise event: 1,81234E-15
PDP - Output fall event: 5,2185E-16
 -- SS PDP --
 pdp_energy_rise=  -1.156502f
 pdp_energy_fall=-343.745963a
 -- FF PDP --
 pdp_energy_rise=  -2.674636f
 pdp_energy_fall=-815.211727a
----

Algorithm for B to Q.
B to Q vector pair:
Vector 1: 1001
Vector 2: 1100
----
 -- Simulation Results - Delays: tpHL + tpLH --
 -- NOM Delays --
Propagation Time Low to High (tpLH): 1,23963E-11
Propagation Time High to Low (tpHL): 1,24703E-11
 -- SS Delays --
 tpLH ss : 27.236519p
 tpHL ss: 19.332825p
 -- FF Delays --
 tpLH ff : 8.146662p
 tpHL ff : 10.327933p
----
----
Simulation Results - Energy
 -- NOM Energy --
Static Energy: 3,58206E-14
Dynamic Energy: 2,72017E-15
Total Energy: 4,1128E-14
 -- SS Energy --
 sta_nrg=  43.539352f
 dyn_nrg=  -1.766101f
 tot_nrg= -47.452156f
 -- FF Energy --
 sta_nrg=  59.173846f
 dyn_nrg=  -3.983849f
 tot_nrg= -52.620121f
----
----
Simulation Results - Power Delay Product (PDP)
 -- NOM PDP --
PDP - Output rise event: 1,96441E-15
PDP - Output fall event: 7,55768E-16
 -- SS PDP --
 pdp_energy_rise=  -1.258453f
 pdp_energy_fall=-507.648069a
 -- FF PDP --
 pdp_energy_rise=  -2.859548f
 pdp_energy_fall=  -1.124300f
----

Algorithm for C to Q.
C to Q vector pair:
Vector 1: 0001
Vector 2: 0010
----
 -- Simulation Results - Delays: tpHL + tpLH --
 -- NOM Delays --
Propagation Time Low to High (tpLH): 5,21832E-12
Propagation Time High to Low (tpHL): 7,32078E-12
 -- SS Delays --
 tpLH ss : 9.459073p
 tpHL ss: 10.541445p
 -- FF Delays --
 tpLH ff : 3.463736p
 tpHL ff : 6.348617p
----
----
Simulation Results - Energy
 -- NOM Energy --
Static Energy: 3,24411E-14
Dynamic Energy: 1,01685E-15
Total Energy: 3,66763E-14
 -- SS Energy --
 sta_nrg=  34.188389f
 dyn_nrg=-611.674266a
 tot_nrg= -39.757660f
 -- FF Energy --
 sta_nrg=  60.717128f
 dyn_nrg=  -1.606807f
 tot_nrg= -68.872529f
----
----
Simulation Results - Power Delay Product (PDP)
 -- NOM PDP --
PDP - Output rise event: 1,17457E-15
PDP - Output fall event: 1,57717E-16
 -- SS PDP --
 pdp_energy_rise=-703.827306a
 pdp_energy_fall=  92.153040a
 -- FF PDP --
 pdp_energy_rise=  -1.799379f
 pdp_energy_fall= 192.571951a
----
```

PSPICE Simulation Templates

Project File

```
* FF CIRCUIT SIMULATION WITH AGING - TIMINGS *

*********************************************************
* AGING CONDICTIONS
* 65nm_bulk_library.sp
* Vdd = 1.1 [V]
*********************************************************
.include "c:\synopsys\hspice_a-2008.03\projs\proj\65nm_bulk_library.sp"
.include "c:\synopsys\hspice_a-2008.03\projs\proj\new_blocks.sp"
.param Vss=0 vdd=0.1 tempo=0
.param periodo=1u
.param startsim=0
.param stepsim=0.1
.param stopsim=1.1
.param trtf=10p
.temp 27
*********************************************************

** TIME MEASURE *****************************************
* .measure tran sensor trig v(out1) val='vdd/2' td=0 cross=2 targ v(din) val='vdd/2' td=0 cross=2

** signal quality vdd test ********************************************************************
** vdd: 0 < vdd/2; 1 > vdd/2 **
* .measure tran crit1vdd FIND V(out1) AT='periodo*2.8'
* .measure tran crit2vdd FIND V(out1) AT='periodo*3.8'
** sinal quality tpd test *********************************************************************
** tpd: LH<T/2; HL<T/2 **
.measure tran tpdhl trig v(din) val='vdd/2' td=0 cross=2 targ v(out1) val='vdd/2' td=0 cross=2
* .measure tran tpdlh trig v(out1) val='vdd/2' td=0 fall=3 targ v(din) val='vdd/2' td=0 rise=3
***********************************************************************************************

*********************************************************
*.measure tran energy max v(energ)  from=0s to=200ns
*********************************************************

vddfonte Vdd! 0 dc Vdd
vssfonte Vss! 0 dc Vss
Vdin din Vss! dc vss pulse(vss vdd 'periodo*0.8' trtf trtf  'periodo-trtf' '2*periodo')
vclock clk Vss! dc vss pulse(vdd vss 'periodo/2' trtf trtf 'periodo/2-trtf'  periodo)

vreset reset vss! DC Vss PWL('periodo*2.2' vss 'periodo*2.2+trtf' Vdd)

** SIMULATION TIME  *************************************
   .tran 10p '10*periodo'

   *.tran 10p '10*periodo' sweep vdd startsim stopsim stepsim

   *.tran 10p '10*periodo' sweep vdd 0.005 0.2 0.015
   *.tran 10p '2*periodo' sweep tempo 3.99n 4.05n 0.002n
   *sweep periodo 140p 250p 10p
   *.plot tran v(clk) v(din) v(out1) v(energ)

** CIRCUIT *********************************************

xcomp din vss! out1 Vss! Vdd! NAND20

********************************************************

** PDP Subcircuit
.IC V(energ)=0

F1 energ 0 vddfonte 'vdd*5e-15*1e15'
C1 energ 0  5e-15
R1 energ 0  1G

********************************************************

.end
```

PSPICE Simulation Templates
Behaviour File

```
* CIRCUIT SIMULATION - BEHAVIOR *


********************************************************
* AGING CONDICTIONS
* 65nm_bulk_library.sp
* Vdd = 1.1 [V]
********************************************************
.include "c:\synopsys\hspice_a-2008.03\projs\proj\65nm_bulk_library.sp"
.include "c:\synopsys\hspice_a-2008.03\projs\proj\new_blocks.sp"
.param Vss=0 vdd=1.1 tempo=0
.param periodo=1u
.param trtf=10p
.temp 27
********************************************************
** TIME MEASURE ****************************************


**.measure tran sensor trig v(outa) val='vdd/2' td=0 cross=2 targ v(clk) val='vdd/2' td=0 cross=2
.measure tran logic_out FIND V(outa) AT=1us


********************************************************
vddfonte Vdd! 0 dc Vdd
vssfonte Vss! 0 dc Vss
Vdin din Vss! dc vss pulse(vss vdd 'periodo*0.8' trtf trtf  'periodo-trtf' '2*periodo')
vclock clk Vss! dc vss pulse(vdd vss 'periodo/2' trtf trtf 'periodo/2-trtf'  periodo)


vreset reset vss! DC Vss PWL('periodo*2.2' vss 'periodo*2.2+trtf' Vdd)


** SIMULATION TIME  ************************************


  .tran 10p '2*periodo'


** CIRCUIT ********************************************


xcomp vdd! vdd! vdd! outa Vss! Vdd! IMUX30


.end
```

PSPICE Simulation Templates

Transitions File

```
* CIRCUIT SIMULATION NOMINAL *


*********************************************************

* AGING CONDICTIONS

* 65nm_bulk_library.sp

* Vdd = 1.1 [V]

*********************************************************


*********************************************************

** INCLUDE **********************************************


.include "c:\synopsys\hspice_a-2008.03\projs\proj\65nm_bulk_library.sp"

.include "c:\synopsys\hspice_a-2008.03\projs\proj\new_blocks.sp"


*********************************************************

** PARAMS ***********************************************


.param Vss=0 vdd=1.1 tempo=0

.param periodo=1u

.param trtf=10p

.temp 27


*********************************************************

** OPTIONS **********************************************

*measures digits


.options MEASDGT=6


*********************************************************

** MEASURES - RESULTS ***********************************


** ENERGY CALC - FIRST METHOD **


** TIME ** ---------------------------------------------


*RISE DELAY

.measure rise_delay trig v(din) val='vdd/2' fall=1 targ v(outa) val='vdd/2' rise=1
```

```
*FALL DELAY
.measure fall_delay trig v(din) val='vdd/2' rise=1 targ v(outa) val='vdd/2' fall=1


*TOTAL TIME RISE
.measure tran T1 WHEN v(din)='0.9*vdd' fall=1
.measure tran T2 WHEN v(outa)='0.9*vdd' rise=1
.measure time_rise PARAM='T2-T1'


*TOTAL TIME FALL
.measure tran T3 WHEN v(din)='0.1*vdd' rise=1
.measure tran T4 WHEN v(outa)='0.1*vdd' fall=1
.measure time_fall PARAM='T4-T3'


** CURRENT ** --------------------------------------------


*AVERAGE CURRENT RISE
.measure tran i_rise avg I(vddfonte) from=T1 to=T2


*AVERAGE CURRENT FALL
.measure tran i_fall avg I(vddfonte) from=T3 to=T4


** ENERGY ** ---------------------------------------------


*POWER DELAY PRODUCT (rise event) - SWITCHING ENERGY (switching event) - FIGURE OF MERIT
*ENERGY RISE
.measure PDP_ENERGY_RISE PARAM='i_rise*vdd*(T2-T1)'


*POWER DELAY PRODUCT (fall event) - SWITCHING ENERGY (switching event) - FIGURE OF MERIT
*ENERGY FALL
.measure PDP_ENERGY_FALL PARAM='i_fall*vdd*(T4-T3)'


*ENERGY (max V(energy))
.measure tran tenergy max v(energy) at=(2*periodo)


*TOTAL ENERGY
.measure tran TOTAL_ENERGY PARAM='tenergy*1E-14'


** POWER ** ---------------------------------------------


*POWER RISE
.measure POWER_RISE PARAM='PDP_ENERGY_RISE/(T2-T1)'
```

```
*POWER FALL
.measure POWER_FALL PARAM='PDP_ENERGY_FALL/(T4-T3)'

*TOTAL POWER
.measure TOTAL_POWER PARAM='TOTAL_ENERGY/(2*periodo)'

*******************************************************
** ENERGY CALC - SEC METHOD **

** CHARGE ** -------------------------------------------

*CHARGE RISE
.measure tran q_rise integral I(vddfonte) from=T1 to=T2

*CHARGE FALL
.measure tran q_fall integral I(vddfonte) from=T3 to=T4

*TOTAL CHARGE
.measure tran q_total integral I(vddfonte) at=(2*periodo)

** ENERGY ** --------------------------------------------

*POWER DELAY PRODUCT (rise event) - SWITCHING ENERGY (switching event) - FIGURE OF MERIT
*2ND ENERGY RISE
.measure PDP_E_RISE PARAM='vdd*q_rise'

*POWER DELAY PRODUCT (fall event) - SWITCHING ENERGY (switching event) - FIGURE OF MERIT
*2ND ENERGY FALL
.measure PDP_E_FALL PARAM='vdd*q_fall'

*2ND TOTAL ENERGY
.measure tran TOT_NRG PARAM='vdd*q_total'

** POWER ** --------------------------------------------

*2ND POWER RISE
.measure P_RISE PARAM='PDP_E_RISE/(T2-T1)'

*2ND POWER FALL
.measure P_FALL PARAM='PDP_E_FALL/(T4-T3)'
```

```
*2ND TOTAL POWER

.measure T_POWER PARAM='TOT_NRG/(2*periodo)'

********************************************************

** STATIC/DYNAMIC ENERGY *******************************

*TOTAL DYNAMIC ENERGY

.measure DYN_NRG PARAM='PDP_ENERGY_FALL+PDP_ENERGY_RISE'

*TOTAL STATIC ENERGY

.measure STA_NRG PARAM='TOTAL_ENERGY-DYN_NRG'

********************************************************

** DELAY ***********************************************

*RISE DELAY

.measure tpLH trig v(din) val='vdd/2' fall=1 targ v(outa) val='vdd/2' rise=1

*FALL DELAY

.measure tpHL trig v(din) val='vdd/2' rise=1 targ v(outa) val='vdd/2' fall=1

********************************************************

** SOURCES  ********************************************

vddfonte Vdd! 0 dc Vdd

vssfonte Vss! 0 dc Vss

vdin din Vss! dc vss pulse(vss vdd 'periodo*0.8' trtf trtf  'periodo-trtf' '2*periodo')

* vclock clk Vss! dc vss pulse(vdd vss 'periodo/2' trtf trtf 'periodo/2-trtf'  periodo)

*******************************************************

** SIMULATION TIME  ***********************************

.tran 10p '2*periodo'

*******************************************************

** CIRCUIT ********************************************

xcomp din vss! outa Vss! Vdd! IMUX20

*******************************************************
```

```
** ENERGY **********************************************

** PDP Subcircuit
.IC V(energy)=0

F1 energy 0 vddfonte 'vdd*5e-15*1E14'
C1 energy 0  5e-15
R1 energy 0  1G

********************************************************
********************************************************
** Characterization **

** Case SS
.alter CIRCUIT SIMULATION SS
.param vdd='1.1*0.8'
.temp 100

** Case TYP
.alter CIRCUIT SIMULATION TYP
.param vdd='1.1'
.temp 25

** Case FF
.alter CIRCUIT SIMULATION FF
.param vdd='1.1*1.2'
.temp 0

.end
```

PSPICE Simulation Templates
Sweep File

```
* CIRCUIT SIMULATION NOMINAL *


*********************************************************
* AGING CONDICTIONS
* 65nm_bulk_library.sp
* Vdd = 1.1 [V]
*********************************************************


*********************************************************
** INCLUDE **********************************************


.include "c:\synopsys\hspice_a-2008.03\projs\proj\65nm_bulk_library.sp"
.include "c:\synopsys\hspice_a-2008.03\projs\proj\new_blocks.sp"


*********************************************************
** PARAMS ***********************************************
* default periodo=1u


.param Vss=0 vdd=1.1 tempo=0
.param periodo=1u
.param trtf=10p
.temp 27


*********************************************************
** OPTIONS **********************************************
*measures digits


.options MEASDGT=6


*********************************************************
** MEASURES - RESULTS ***********************************


** ENERGY CALC - FIRST METHOD **


** TIME ** --------------------------------------------


*RISE DELAY
```

```
.measure rise_delay trig v(din) val='vdd/2' fall=1 targ v(outa) val='vdd/2' rise=1

*FALL DELAY
.measure fall_delay trig v(din) val='vdd/2' rise=1 targ v(outa) val='vdd/2' fall=1

*TOTAL TIME RISE
.measure tran T1 WHEN v(din)='0.5*vdd' fall=1
.measure tran T2 WHEN v(outa)='0.5*vdd' rise=1
.measure time_rise PARAM='T2-T1'

*TOTAL TIME FALL
.measure tran T3 WHEN v(din)='0.5*vdd' rise=1
.measure tran T4 WHEN v(outa)='0.5*vdd' fall=1
.measure time_fall PARAM='T4-T3'

** CURRENT ** --------------------------------------------

*AVERAGE CURRENT RISE
.measure tran i_rise avg I(vddfonte) from=T1 to=T2

*AVERAGE CURRENT FALL
.measure tran i_fall avg I(vddfonte) from=T3 to=T4

** ENERGY ** ---------------------------------------------

*POWER DELAY PRODUCT (rise event) - SWITCHING ENERGY (switching event) - FIGURE OF MERIT
*ENERGY RISE
.measure PDP_ENERGY_RISE PARAM='i_rise*vdd*(T2-T1)'

*POWER DELAY PRODUCT (fall event) - SWITCHING ENERGY (switching event) - FIGURE OF MERIT
*ENERGY FALL
.measure PDP_ENERGY_FALL PARAM='i_fall*vdd*(T4-T3)'

*ENERGY (max V(energy))
.measure tran tenergy max v(energy) at=(2*periodo)

*TOTAL ENERGY
.measure tran TOTAL_ENERGY PARAM='tenergy*1E-14'

** POWER ** ----------------------------------------------
```

```
*POWER RISE
.measure POWER_RISE PARAM='PDP_ENERGY_RISE/(T2-T1)'

*POWER FALL
.measure POWER_FALL PARAM='PDP_ENERGY_FALL/(T4-T3)'

*TOTAL POWER
.measure TOTAL_POWER PARAM='TOTAL_ENERGY/(2*periodo)'

********************************************************
********************************************************
** ENERGY CALC - SECOND METHOD **

** CHARGE ** ------------------------------------------

*CHARGE RISE
.measure tran q_rise integral I(vddfonte) from=T1 to=T2

*CHARGE FALL
.measure tran q_fall integral I(vddfonte) from=T3 to=T4

*TOTAL CHARGE
.measure tran q_total integral I(vddfonte) at=(2*periodo)

** ENERGY ** -------------------------------------------

*POWER DELAY PRODUCT (rise event) - SWITCHING ENERGY (switching event) - FIGURE OF MERIT
*2ND ENERGY RISE
.measure PDP_E_RISE PARAM='vdd*q_rise'

*POWER DELAY PRODUCT (fall event) - SWITCHING ENERGY (switching event) - FIGURE OF MERIT
*2ND ENERGY FALL
.measure PDP_E_FALL PARAM='vdd*q_fall'

*MEAN PDP - (PDP RISE + PDP FALL)/2
.measure MEAN_PDP PARAM='(PDP_E_RISE+PDP_E_FALL)/2'

**------------------------------------------------------

*ENERGY DELAY PRODUCT (rise event)
*2ND ENERGY X DELAY - RISE
```

```
.measure EDP_E_RISE PARAM='PDP_E_RISE*(T2-T1)'

*ENERGY DELAY PRODUCT (fall event)
*2ND ENERGY X DELAY - FALL
.measure EDP_E_FALL PARAM='PDP_E_FALL*(T4-T3)'

*MEAN EDP - (EDP RISE + EDP FALL)/2
.measure MEAN_EDP PARAM='(EDP_E_RISE+EDP_E_FALL)/2'

**-------------------------------------------------------
* first interval - Tp
.measure INT1 PARAM='T2-T1'

* second interval - Tp
.measure INT2 PARAM='T4-T3'

* mean Tp (Propag Time)
.measure MEAN_TP PARAM='(INT1+INT2)/2'

**-------------------------------------------------------

*2ND TOTAL ENERGY
.measure tran TOT_NRG PARAM='vdd*q_total'

** POWER ** ---------------------------------------------

*2ND POWER RISE
.measure P_RISE PARAM='PDP_E_RISE/(T2-T1)'

*2ND POWER FALL
.measure P_FALL PARAM='PDP_E_FALL/(T4-T3)'

*2ND TOTAL POWER
.measure T_POWER PARAM='TOT_NRG/(2*periodo)'

*********************************************************
** SIGNAL QUALITY  **************************************

*VDD QUALITY
*.measure vdd_qua PARAM=' '
```

```
*TIME QUALITY
*.measure tpd_qua PARAM=' '

*********************************************************
** STATIC/DYNAMIC ENERGY ********************************

*TOTAL DYNAMIC ENERGY
.measure DYN_NRG PARAM='PDP_ENERGY_FALL+PDP_ENERGY_RISE'

*TOTAL STATIC ENERGY
.measure STA_NRG PARAM='TOTAL_ENERGY-DYN_NRG'

*********************************************************
** DELAY ************************************************

*RISE DELAY
.measure tpLH trig v(din) val='vdd/2' fall=1 targ v(outa) val='vdd/2' rise=1

*FALL DELAY
.measure tpHL trig v(din) val='vdd/2' rise=1 targ v(outa) val='vdd/2' fall=1



*********************************************************
** SOURCES  *********************************************

vddfonte Vdd! 0 dc Vdd
vssfonte Vss! 0 dc Vss
vdin din Vss! dc vss pulse(vss vdd 'periodo*0.8' trtf trtf  'periodo-trtf' '2*periodo')
* vclock clk Vss! dc vss pulse(vdd vss 'periodo/2' trtf trtf 'periodo/2-trtf'  periodo)

********************************************************
** SIMULATION TIME  ************************************

.tran 10p '10*periodo' sweep vdd 0.1 1.1 0.1

********************************************************
** CIRCUIT *********************************************

xcomp vss! vss! vss! din outa Vss! Vdd! AOI310

********************************************************
```

```
** ENERGY **********************************************

** PDP Subcircuit
.IC V(energy)=0

F1 energy 0 vddfonte 'vdd*5e-15*1E14'
C1 energy 0  5e-15
R1 energy 0  1G

********************************************************
********************************************************
** Characterization **

** Case SS
*.alter CIRCUIT SIMULATION SS
*.param vdd='1.1*0.8'
*.temp 100

** Case TYP
*.alter CIRCUIT SIMULATION TYP
*.param vdd='1.1'
*.temp 25

** Case FF
*.alter CIRCUIT SIMULATION FF
*.param vdd='1.1*1.2'
*.temp 0

********************************************************
********************************************************


.end
```

MERIT - cMos EneRgy sImulaTor

INI FILE (EXAMPLE)

```
[User]
Admin=H
Date=28/09/2017
Last=H
[Paths]
Sim_Path=c:\synopsys\Hspice_A-2008.03\BIN\hspice.exe
Run_Path=C:\Users\Admin\Documents\Software Projects\Simulation220417\Win32\Release\
Proj_Path=C:\Users\ Admin\Documents\Software Projects\Simulation220417\Win32\Release\proj\
Proj_Debug=C:\synopsys\Hspice_A-2008.03\projs\proj\
[Simulation]
Proj_File=proj
Beha_File=behaviour
Trans_File=transitions
Sweep_File=sweep
[Software]
Name=MERIT - cMos EneRgy sImulaTor
Version=1.3
Date=20-09-2017
[Database]
Path=C:\Users\Admin\Documents\Software Projects\Simulation220417\Win32\Release\db\
Path_Debug=C:\sqlite\
File=mosdb.s3db
Version=1.0
Date=04-2017
```