Online Nbti Wear-out Estimation

ONLINE NBTI WEAR-OUT ESTIMATION

A Thesis presented

by

MEHERNOSH H. DABHOIWALA

Submitted to the Graduate School of the
University of Massachusetts in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

September 2013

ELECTRICAL AND COMPUTER ENGINEERING

ONLINE NBTI WEAR-OUT ESTIMATION

A Thesis presented

by

MEHERNOSH H. DABHOIWALA

Approved as to style and content by:

_____
Wayne Burleson, Chair

_____
Russell Tessier, Member

_____
Sandip Kundu, Member

_____
C.V. Hollot, Department Head
Electrical and Computer Engineering

ACKNOWLEDGEMENTS

To begin with, I would like to sincerely thank my advisor, Prof. Wayne Burleson for all his support, faith in my abilities and encouragement throughout my tenure as a graduate student. Without his guidance, this thesis wouldn‘t have been possible. I am also very thankful to Justin Lu, who has been my constant tutor throughout this project. I couldn‘t have asked for a better teammate to work with. His focus and dedication to this project despite his own research is something I will always appreciate. I extend my gratitude towards Prof. Russell Tessier and Prof. Sandip Kundu, and would like to thank them for being on my thesis committee.

Next, I would like to thank all my wonderful current and former lab mates— Hari, Deepak, Kekai, Cory, Justin, Sandesh, Zach and Novak, for making me feel comfortable in the lab. I thank Zach for proof reading this document. I would also like to thank all my other friends that I have made over the past 2 years in Amherst, for making my stay so enjoyable. The town and its people have made my stay a truly worthwhile experience.

No acknowledgement is complete without expressing your gratitude and thankfulness towards one‘s family. They have always been, and will always be there through my best and worst of times. I deeply thank them for their support and faith in me. I feel truly blessed to have them in my life.

ABSTRACT

ONLINE NBTI WEAR-OUT ESTIMATION

SEPTEMBER 2013

MEHERNOSH H. DABHOIWALA

B.E., SARDAR PATEL UNIVERSITY, INDIA

M.S.E.C.E., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed By: Professor Wayne Burleson

CMOS feature size scaling has been a source of dramatic performance gains, but it has come at a cost of on-chip wear-out. Negative Bias Temperature Instability (NBTI) is one of the main on-chip wear-out problems which questions the reliability of a chip. To check the accuracy of Reaction-Diffusion (RD) model, this work first proposes to compare the NBTI wear-out data from the RD wear-out model and the reliability simulator - Ultrasim RelXpert, by monitoring the activity of the register file on a Leon3 processor. The simulator wear-out data obtained is considered to be the baseline data and is used to tune the RD model using a novel technique *time slicing*. It turns out that the tuned RD model NBTI degradation is on an average 80% accurate with respect to RelXpert simulator and its calculation is approximately 8 times faster than the simulator. We come up with a *waveform compression technique*, for the activity waveforms from the Leon3 register file, which consumes 131KB compared to 256MB required without compression, and also provides 91% accuracy in NBTI degradation, compared to the same obtained without compression. We also propose a NBTI ΔV_{th} *estimation/prediction technique* to reduce the time consumption of the tuned RD model threshold voltage calculation by an order of 10^2 with one day degradation being 93% within the same of the tuned RD model. This work further proposes to a novel NBTI Degradation Predictor (NDP), to predict the future NBTI degradation, in a DE2 FPGA for WCET benchmarks. Also we measure the ΔV_{th} variation across the 4 corners of the DE2 FPGA running a single Leon3, which varies from 0.08% to 0.11% of the base V_{th}.

TABLE OF CONTENTS

P a g e

LIST OF TABLES

P a g e

T A B L E

LIST OF FIGURES

P a g e

F I G U R E

CHAPTER 1

MOTIVATION

Continuous transistor scaling leads to an increase in current density and temperature, which results in high on-chip wear-out. This wear-out results in need for wear-out sensing or wear-out estimation. Sensing can be characterized using delay, canary and dummy devices (discussed in Section 3.1). Delay sensors [1-5] provides a continuous aging report of the module they monitor. They only work well for combinational logic and fail to provide wear-out information for storage units like SRAM cells. Canary based [6] and dummy device based [7] wear-out sensors provide just a binary report, and not one during the course of degradation for carrying out some management to slow down wear-out and prolong the lifetime of the device. Thus wear-out estimation becomes necessary for wear-out management.

Negative Bias Temperature Instability (NBTI) is the main reliability concern for CMOS circuits [28]. The Reaction-Diffusion (RD) model [1] (explained in Section 4.1) is a widely used model NBTI prediction. To the best of our knowledge, no work has been done to explain how the RD model is implemented. This work proposes to use the RD model to predict NBTI degradation on the register file of a Leon3 processor. The same analysis is performed using the Ultrasim - RelXpert simulator [22], which is regarded as the baseline. Comparing these results would give an idea of how accurate the RD model is. Results from this more time consuming simulator are used to tune the RD model and calibrate its results. Using design time simulation tools, such as RelXpert, at run-time is slow and impracticable. The RD model, based on run-time waveforms, has the potential to be fast and feasible.

Run-time wear-out prediction requires the run-time activity information to be stored at run-time which increases the cost and complexity. This work presents a novel *waveform compression* technique which minimizes the memory cost from 256MB to 131KB for a Leon 3 processor register file.

Wear-out occurs over a long time period. The RD model cannot be used to calculate the threshold voltage degradation due to NBTI over a long time period , as it would take a long time to simulate the equations (presented in Section 4), say in hours, which is infeasible at run-time and would degrade the system performance. In this work a novel *ΔV_{th} estimation/prediction technique* is proposed, which would not require the RD model to run for the length of the degradation, but only for a fraction of the time, to provide an accurate degradation result. This work shows that for the Dhrystone benchmark running on a Leon3 processor, ΔV_{th} Estimation/Prediction Technique would reduce the run-time NBTI prediction by an order of 10^2 with 93% accuracy, compared to the tuned RD model, for a period of one day.

To the best of our knowledge no online NBTI predictor has been designed which can predict the future NBTI ΔV_{th} degradation on a real system. This prediction can be used to do task management which can reduce the future degradation and increase the system's lifetime. Here we design a novel NBTI Degradation Predictor (NDP) running on a Leon3 in a DE2 FPGA. This predictor is designed to predict the future ΔV_{th} degradation of the Leon3 register file cells. This design also shows how we can measure the actual on-chip degradation history of the Leon3 register file, which is necessary for implementing the RD model. This prediction can be used to estimate how the processor would behave in coming years, and necessary management steps can be taken to prevent it from crashing. Lastly, we present a novel technique to measure the process variation across the 4 corners of the DE2 FPGA, running a Leon3 processor, using ring oscillators.

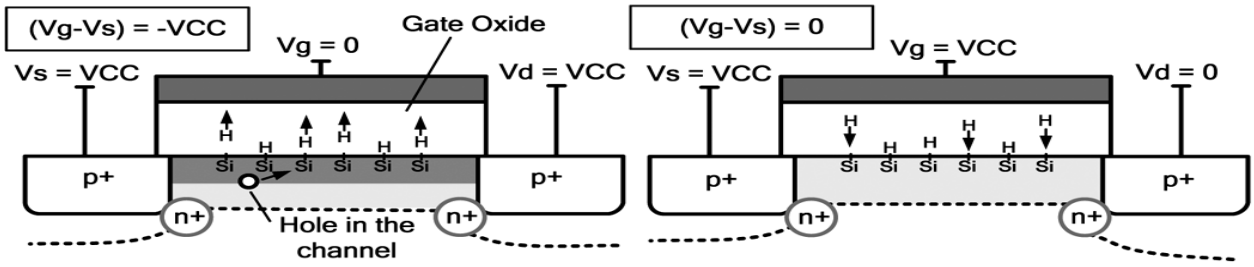
CHAPTER 2

INTRODUCTION

Microprocessors have been designed with worst case operating conditions in mind, and manufacturers have employed guardbands to make sure that the processors will meet a predefined lifetime qualification. However shrinking feature size has made process variation extremely difficult to mitigate simply by provisioning for the worst case. This makes a necessity for designers to provide on-chip wear-out sensors [1-7] or perform out wear-out estimation [9,10]. These sensors provide fresh online wear-out data over a period of time. The run-time degradation can be estimated using degradation models, like the RD model, which are faster than simulation tools, and can work with activity data. Negative Bias Temperature Instability (NBTI) is the main reliability concern which limit a circuit's lifetime [28]. Storage devices, like the register file, have a biased value at the input of its PMOS transistors for a quite long time, which results in more NBTI degradation.

In CMOS fabrication, during the hydrogen passivation process that follows oxidation, dangling Si bonds are transformed into Si-H bonds. These bonds are weak enough to break during device operation, causing H atoms to diffuse into gate oxide, and the broken bonds that remain become traps (called interface traps), effectively degrading the drive current of PMOS transistors. NBTI is caused by this trap generation in the Si–SiO_2 interface of PMOS transistors. Structural mismatch at the Si-SiO_2 interface causes dangling bonds, which act as interfacial traps. NBTI is characterized by a positive shift in the absolute value of the PMOS threshold voltage $|V_{tp}|$, which occurs when the device is stressed (V_{gs}= -VCC). When the stress conditions are removed (i.e. V_{gs}=0), the device enters a recovery phase, where H atoms diffuse back towards the Si–SiO2 interface and anneal the broken Si–H bonds, thereby reducing $|V_{tp}|$[Fig. 2-1(a) and (b)]. It has been observed that NBTI can increase V_{th} by as much as 50mV for

devices operating at 1.2V or below [11] and the circuit performance degradation may reach upwards of 20% in 10 years [12]. When the input of the PMOS is '0', i.e. V_{gs}= -VCC, it is on and $|V_{th}|$ increases, which is known as the stress phase. When its input is '1', i.e. V_{gs}=0, it is off and $|V_{th}|$ decreases, which is known as the recovery phase.



(a)*NBTI Stress Phase* (b)*NBTI Recovery Phase*

Figure 2-1. NBTI stress (a) and recovery (b) phases [8]

The Reaction-Diffusion (RD) Model [8], is a predictive NBTI model used to predict the effect of NBTI in the form of an increase in the Threshold Voltage (V_{th}) . It provides two equations (mentioned in Section 4.1) to calculate V_{th} change during multiple NBTI stress and recovery periods. V_{th} is believed to exhibit a power-law dependency on time and is an exponential function of the stress voltage level as well as temperature. The RD model is discussed in section 4.1.

A question arises. How accurate is this model? If it is not accurate, how do we improve the model to capture the actual degradation? This work first proposes to compare the performance degradation due to NBTI wear-out, obtained from the RD model and RelXpert [22], the inbuilt reliability simulator in Ultrasim. This is carried out by monitoring the activity of the register file by running Dhrystone benchmark on a Leon3 processor. The reliability simulator, RelXpert, uses the same RD model for carrying out NBTI calculations. The RD mechanism is frequency independent [16], but the RD model is frequency dependent, which questions its accuracy and needs to be tuned. Our proposed tuning

method is mentioned in Section 5. The RD model is tuned, using a *time slicing technique*, to match the RelXpert simulator. The tuned RD model can be used to perform wear-out management on the Leon 3 processor. Figure 2-2 shows the flow of the proposed work. This method provides 80% accuracy with respect to the RelXpert simulator and is also 8 times faster than the simulator.

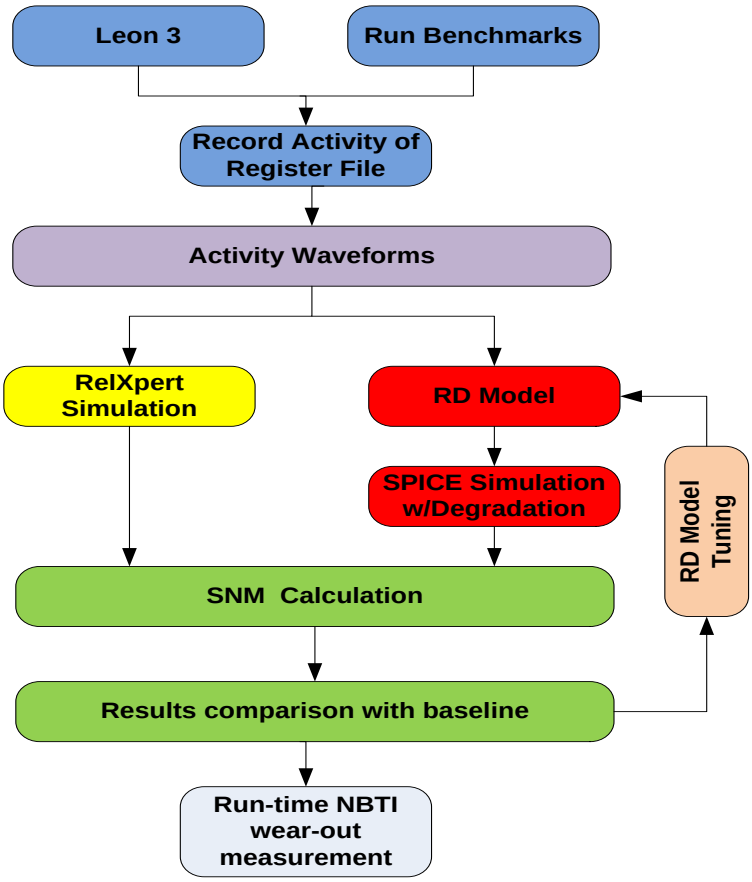


Figure 2-2. Flow of the proposed work

The next question is - "What is the performance and cost of the above proposed technique?" While monitoring the activity of the register file, the waveforms need to be stored into memory which would increase the cost. Also running the RD model to predict a long term V_{th} degradation could affect the run-time performance.

Monitored register file activity waveforms need to be stored in memory to carry out NBTI wear-out estimation. Storing these activity waveforms from a 256 32-bit register file in Leon3 in memory is very

costly. Thus we develop a *waveform compression technique* which stores only the statistics, mean and standard-deviation (SD), of the activity waveforms which occupy 131KB memory instead of 256MB (for Dhrystone benchmark running once, i.e. 58ms), and at the same time provide 91% accuracy with respect to the tuned RD model. To predict the long term degradation, say 1 day, running the RD model is infeasible due to the huge amount of time it takes. For Dhrystone benckmark running on Leon3, it takes nearing 4 hours to predict ΔV_{th} of a single bit, for 1 day prediction, using the tuned RD model. So we come up with a *ΔV_{th} estimation/prediction technique* that can predict the threshold voltage degradation due to NBTI, running the RD model just for a small duration of time. We show that our technique is faster than the RD model, running for a time period of 1 day, by an order of 10^2 with 93% accuracy (93% within the tuned RD model results).

The above proposed techniques need to be implemented online on a real system. Predicting what is going to happen ahead would allow the system to carry out online task management which can reduce the future degradation and increase it's lifetime. For example, in a 4-core multi-core system, from its current degradation characteristics, core-2 predicts a high future degradation which would hamper its lifetime. So the system can allocate the tasks of core-2 to another core, say core-4, whose future degradation prediction is less.

This work also proposes to implement a NBTI degradation predictor on Leon 3 in a DE2 FPGA that carries out waveform compression, and ΔV_{th} estimation and prediction, to predict the future NBTI ΔV_{th} degradation, for the WCET benchmark suite [39]. We compare the NBTI degradation of the Leon3 register file bits, MSB and LSB, by running WCET benchmarks. We will see that the WCET benchmarks result in an average of 19.39mV NBTI degradation for LSB and 27.75mV for MSB, over a period of 10 years, for the Leon3 register file. We obtain the ΔV_{th} history by placing 41-stage NAND gate ring

oscillators besides the Leon3 register file in a DE2 FPGA, and measuring the worst case NBTI degradation.

We know that there process, temperature and voltage variations exist across any chip. Process variations are due to variation in manufacturing process. This variation problem is a big concern for technology beyond 90nm [33], and it gets worse with scaling. Here we present a method to measure the initial ΔV_{th} across the four corners of the DE2 FPGA running a single Leon3 core. We achieve a 0.08% to 0.11% variation in ΔV_{th}, from the base V_{th}, across the four corners of the FPGA.

2.1 Organization of the document:

The rest of the document is organized as follows. Section 3 describes the background of NBTI wear-out sensing and estimation. The RD model and the Time Slicing technique are explained in Section 4 and Section 5 respectively. Section 6 explains the novel waveform compression and ΔV_{th} estimation/prediction techniques, and discusses the proposed work for defense, i.e. designing and implementing the NDP on a Leon3 processor in a DE2 FPGA. Section 7 discusses the design and implementation of the NDP and measuring the process variation across the FPGA; while Section 8 discusses its results. Section 9 and 10 are future work and conclusion respectively.

CHAPTER 3

BACKGROUND WORK

Feature size scaling has resulted in considerable gains in area and performance, but it has come at a cost of reliability. Reliability budgeting can no longer be considered an afterthought and should be considered as important as power and area by the designer. Technology shrinking has caused a considerable increase in power density and temperature. Thus wear-out sensing or wear-out estimation should be considered.

3.1 Wear-out Sensors

Various wear-out sensors have been designed to measure the on-chip degradation. They can be classified into delay, canary based and dummy device sensors. The following sub-section discusses these wear-out sensors.

3.1.1 Delay sensors

Delay sensors measure the on-chip wear-out using the performance degradation metric as 'delay'. Maximum wear-out sensors come into this category. This sub-section describes two of them.

An adaptive error prediction flip-flop architecture with built-in aging sensor is proposed in [3], performing on-line monitoring of long-term performance degradation of CMOS synchronous digital circuits in 65nm CMOS technology. The sensor is out of the signal path. Performance error prediction is implemented by the detection of late transitions at flip-flop data input, caused by NBTI. It also shows that the impact of aging degradation and/or PVT (process, power supply voltage and temperature) variations on the sensor enhance error prediction. Such sensors are inserted at selected Flip-flops (FF) on the chip. Figure 3-1 represents the proposed design in [3]. The delay element introduced an observation (or guard-banding) interval, t_g, at the end of the clock cycle. With the sensor's architecture,

its sensitivity (measured by t_g) increases with its PVT variations. This way, the sensor FF will adapt and increase the guard-band, as circuit variability increases with aging.

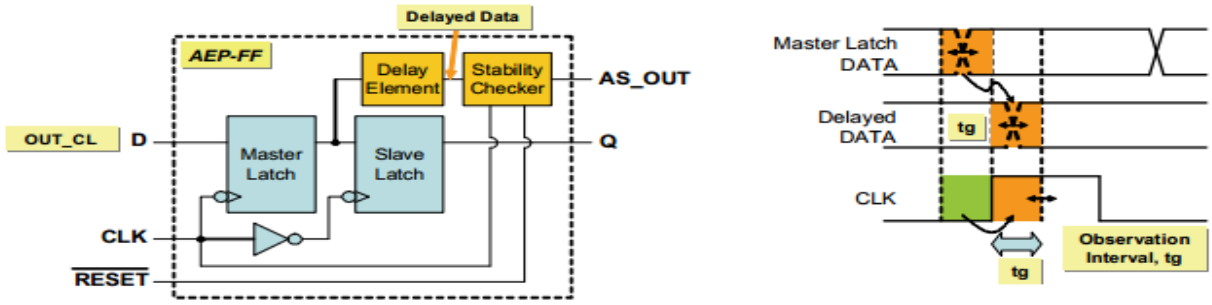


Figure 3-1. NBTI degradation measuring sensor placed at selected FFs [3]

As shown in Figure 3-1, the Delay Element (DE) delays data signals captured at the Master Latch output, during the CLK low state. The Stability Checker (SC) analyzes data transitions during the CLK high state. The DE propagation delay is the effective observation (or guardband) interval, t_g, used by the sensor. Late transitions at the FF data input (propagated to the Master Latch output) will be identified by the SC.

A mechanism of detecting degradation in delay due to NBTI (in 90nm and 65nm technologies), by placing sensors in selected Flip Flops across the chip is proposed in [5]. This design provides an initial short guard banding interval to the circuit design. Figure 3-2 shows the idea behind the design. The output of a combinational logic is fed to the input of the flip flop. If there is a transition in the combinational logic's output during the guardbanding interval, t_g, it results in guardband violation, and an error is detected. There is a stability checker inbuilt into the FFs which checks whether guardband violation has occurred or not. As shown in Figure 3-2, if the stability checker senses a transition during the guardbanding interval, t_g, error is detected.

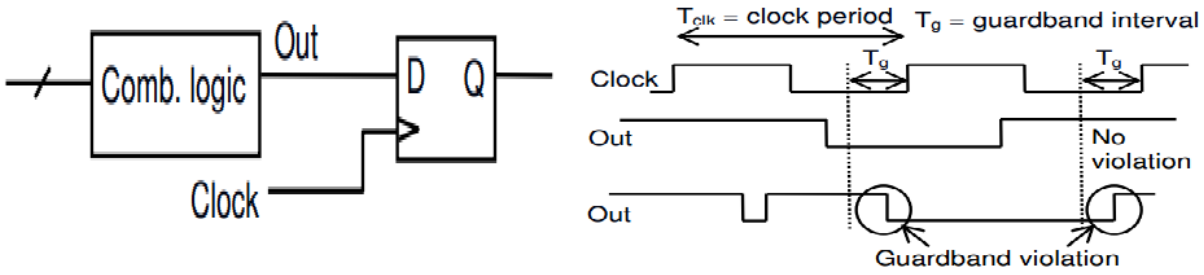


Figure 3-2. Detection in Change in 'Out' will be regarded as a Guardband violation [5]

3.1.2 Canary based sensors

Canary circuits degrade faster to provide an early warning alarm for wear-out.

A Built-In Proactive Tuning (BIPT) System consisting of the existing main circuit augmented with a Test Pattern Generator (TPG), Body Bias Circuitry, Canary Circuit and Control circuit is proposed in [6]. At power-on or periodically, the BIPT system can launch test vectors from the Test Pattern Generator (TPG) and then tune the circuit body voltage according to the observations from the canary circuit. The canary circuit plays the role of predicting aging-induced performance degradations. A warning signal is generated by the canary flip-flops when the timing constraint is tight on one or more of the few critical paths where these circuits are inserted. The top-level warning signal is the OR of all the individual canary flip-flop warning signals. The control circuit is used to generate control signals to tune the body bias of the main circuit. Compared to DVS, BIPT can achieve the same aging resilience with about 30% less power dissipation. Figure 3-3 shows the BIPT design.

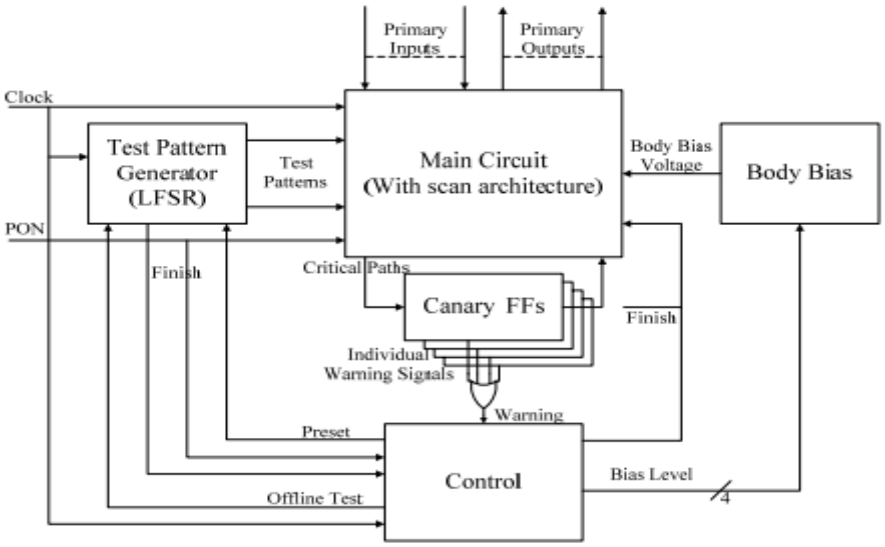


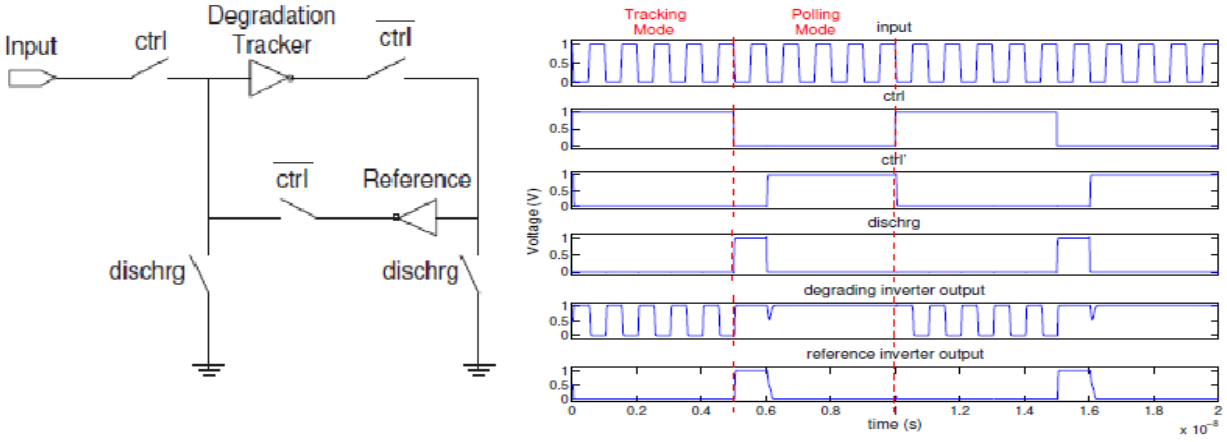
Figure 3-3. Built-in proactive tuning system [6]

3.1.3 Dummy devices based sensors

Some sensors [7] are designed having dummy devices. When the main device gets worn-out, the dummy device comes into picture, like the degradation tracking inverter gets worn out switches to the reference inverter in Figure 3-5 . This switch in device can indicate NBTI degradation.

This design (as shown in Figure 3-4(a) [7]) is similar to the design of a SRAM cell. Here two inverters are cross-coupled with one having a stronger PMOS than the other. The inverter with a stronger PMOS (by ΔI%) is called the tracking inverter and the other is called reference inverter. During normal operation, a critical path signal is fed into the design with the degradation inverter disconnected from the reference inverter through the pass-transistor with input CTRL(bar). This mode of operation is called tracking mode. This mode degrades the PMOS of the tracking inverter, due to NBTI, and makes it weak. When the PMOS of the tracking inverter becomes weak than the one of the reference inverter, due to NBTI, the circuit switches and the reference inverter starts working. During polling mode the input is disconnected from the tracking inverter and the two inverters are cross-coupled. If the PMOS of the tracking inverter is still stronger than the one of the reference inverter, the tracking inverter will pull the

reference inverter down. In the opposite case the reference inverter will pull the tracking inverter down. Figure 3-4(b) shows the Timing Diagram of the signals in the two modes in 65nm technology.



(a) Gate Level diagram of NBTI sensor[7] (b) Timing diagram of Tracking and Polling modes [7]

Figure 3-4. Sensor [7] and its operation

3.2 Wear-out Estimation:

Delay sensors provide a continuous aging report of the module they monitor and they only work well for combinational logic. They fail to provide wear-out scenario for storage units like SRAM cells. Canary based and dummy device based wear-out sensors provide just a binary report, i.e. a device reaches the degradation limit or not. The warning does not occur during the course of degradation to assist carrying out management to stop wear-out. Implementing the run-time wear-out models with run-time waveforms would give us very accurate information about the current scenario of the on-chip wear-out. Also, implementing run-time wear-out models is much faster than implementing design-time tools. In Section 5.3 we show that RD model, which is a widely used run-time wear-out model, is 8 times faster than design-time RelXpert simulator, for Dhrystone benchmark running on a Leon 3 processor.

Wear-out estimation can be used to predict wear-out in order to start doing management to overcome it. To the best of our knowledge, very little work has been done on wear-out estimation, and the work done does not into account the online on-chip degradation.

The temporal NBTI degradation in static noise margin (SNM) of an SRAM array and f_{MAX} of random logic circuits are highly correlated to the standby leakage current (I_{DDQ}) measurement, and this relationship can be used to predict long term circuit reliability [9]. This reference proposes an efficient NBTI characterization technique based on the I_{DDQ} measurement. Since increase in threshold voltage (V_{th}), due to NBTI, decreases I_{DDQ}, this information can be used to carry out on-chip wear-out prediction. A test chip is fabricated in 130nm 1.2V CMOS technology and a simple 1000 stage inverter chain was selected as target circuitry. NBTI stress was controlled by both voltage and temperature. During stress period, the input to the PMOS in the inverter chain is V_{in}=V_{stress}=1.7V, 1.5V, 1.3V (Figure. 3-5). During I_{DDQ} measurement V_{in} is set to 0, so that the leakage can be measured. V_{in} is flipped back to V_{stress} after the 0.2s measurement period.

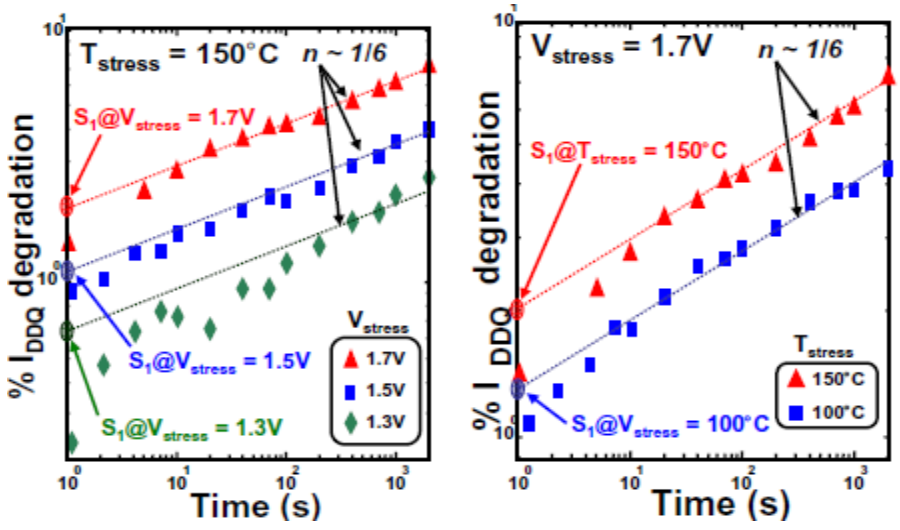


Figure 3-5. % NBTI I_{DDQ} degradation with Vdd and T [9]

SNM degradation of SRAM assuming the input storage nodes of an SRAM being stressed at 50% signal probability is measured [9]. It shows that there is a 53mV V_{th} degradation after a period of 3 years.

A gate level simulation methodology which can accurately model NBTI degradation of digital circuits is developed in [10]. The research shows that their proposed model can be almost as accurate as the PTM NBTI models [27] developed at Arizona State University. It presents a two-state model for circuits having PMOS transistors parallel and connected to the supply (example, NOT and NAND), and a three-state model for circuits having PMOS transistors in a stack (example, NOR). Here the model is implemented for various ISCAS'85 and MCNC'91 circuits, having an input of f=1GHz at T=100°C and VDD=1.2V. The proposed model is validated for an inverter in 45nm, 65nm and 90nm technology nodes, for a period of 10 years, at different temperature and duty cycles.

In both [9] and [10], no use of run-time waveforms (input/activity) has been made which is necessary to obtain the real picture of NBTI degradation on a core. Also, these works have not been done online. Carrying out degradation prediction offline would require the user to bring the device to a service station every time the degradation needs to be predicted increasing the burden on the user. The current degradation characteristics need to be extracted, from the device being tested. This extraction would increase the time consumption and prediction cost. Reference [29] presents ReverseAge, an online NBTI combating technique. Suppose there are few combinational logic circuits separated from one another by flip-flops. If any of the combinational logics fail to meet the setup time of its successive flip-flop, a warning alarm rings. This delay is due to NBTI degradation. Figure 3-6 shows how this problem is solved using time borrowing.

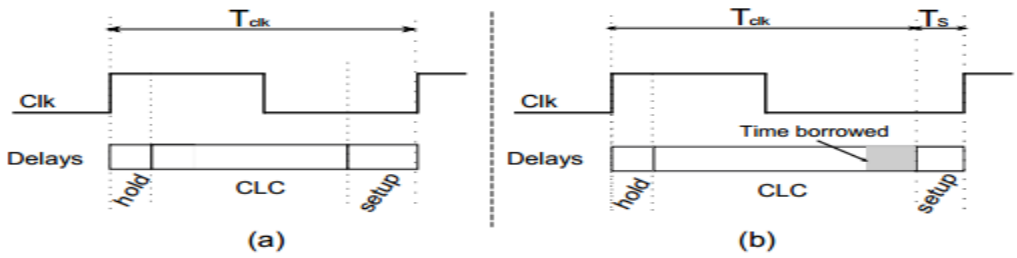


Figure 3.6 (a)Max. allowable delay of CL and FFs (b) Time borrowing by CL using setup margin of FF in the next stage [29]

Time borrowing is performed by relaxing the setup time of the succeeding FF, i.e. shifting the edge later as shown in Figure 3-7(b). The time (Ts) represented by the shaded block is the time borrowed from the succeeding stage and is achieved by shifting (i.e. delaying) the clock rising edge into the FF of the next stage. This technique carries out NBTI sensing but does not predict the future degradation. Predicting the NBTI future would allow the system to carry out NBTI management, like task migration in a multi-core architecture. This will not overcome degradation, but stop it from occurring at the first place.

In this work, we propose to make use of run-time register-file activity of Leon3 in a DE2 FPGA [24] by running various SPEC2000 benchmarks, to predict online the future NBTI degradation.

CHAPTER 4

REACTION-DIFFUSION (RD) MODEL

The Reaction-Diffusion (RD) model is one of the widely used wear-out models used to predict NBTI degradation over a period of time. It is designed based on the Reaction-Diffusion process which takes place during stress of a PMOS transistor. As discussed in Section 2, a PMOS is stressed when its gate-source input, V_{gs}= -VCC, and recovered when V_{gs}=0V.

4.1 The Reaction-Diffusion (RD) Model

The RD model [8] helps us predict the NBTI degradation over a period of time. Equations 1 & 2 show us the stressed V_{th} and recovered V_{th} respectively. The two critical steps that occur in NBTI degradation over time are Reaction and Diffusion. Reaction is where some Si–H or Si–O bonds at the substrate/gate oxide interface are broken under the electrical stress . The holes trigger this reaction. Consequently, interface charges are induced, which cause the increase of V_{th}. In diffusion reaction-generated species diffuse away from the interface toward the gate, driven by the gradient of the density.

$$\Delta Vth(t) = \left(K_v\left(t - t_o\right)^{1/2} + \sqrt[2n]{\Delta Vth\left(t_0\right)}\right)^{2n} \quad \text{.....equation (1)}$$

$$\Delta Vth(t) = \Delta Vth(t_1)\left(1 - \frac{2\xi_1 t_e + \sqrt{\xi_2 C(t - t_1)}}{(1+\delta)t_{ox} + \sqrt{Ct}}\right) \quad \text{.....equation (2)}$$

$$K_v = \left(\frac{q \times t_{ox}}{\varepsilon_{OX}}\right)^3 K_1{}^2 C_{OX}\left(V_{gs} - V_{th}\right)\sqrt{C}\exp\left(\frac{2\mathrm{E}_{OX}}{\mathrm{E}_{01}}\right)$$

$$C = \exp\left(\frac{-\mathrm{E}_a / kT}{T_0}\right)$$

where *t* is the time a which the stress and recovery periods end, t_0 is the time at which the stress period begins and t_1 is the time at which the recovery period begins. $n = \frac{1}{6}$, t_{ox} is the oxide thickness and $t_e = t_{ox}$. δ, ξ_1 and ξ_2 are constants.

4.2 Effects of PVT variations on NBTI using RD model

NBTI is a time-dependent degradation, but this section presents how the process, voltage and temeperature parameters effect it. The V_{th} stress and recovery equations (1 & 2) from RD model, have the initial P, V and T parameters which can be varied and their effect can be studied.

Here we measure the worst and best corners of P, V and T for NBTI wear-out. We provide a stress time of 80us and a recovery time of 20us to equations 1 & 2 (in Section 4.1). The degradation is measured after 1, 2, 3 and 4 days. Equations 1 & 2 are implemented in Matlab [26].

Results were measured with:

- **T at 25°C and 100°C**
- **Vt varying ±20%**
- **V at 1.1V and 0.55V**

Tables 1 & 2 and Figures 4-1 & 4-2 show the the variation in V_{th} degradation with variation in V, T and V_t.

Table 1. presents the V_{th} degradation values achieved for time period of one day to four days under different intial V_t and temperature values at a supply of 0.55V. From Figure 4-1. we see that maximum V_{th} degradation is achieved at low initial V_t and high temperature.

Table 1. V_{th} degradation with Varation in Vt and T with V=0.55V

Time (days)	High Vt, 25°C (uV)	Nom Vt, 25°C (uV)	Low Vt, 25°C (uV)	High Vt, 100°C (uV)	Nom Vt, 100°C (uV)	Low Vt, 100°C (uV)
1	5.51	7.83	10.9	19.89	28.26	39.35
2	6.2	8.81	12.26	22.3	31.7	44.2
3	6.64	9.43	13.13	23.9	34	47.3
4	6.97	9.9	13.8	25.1	35.7	49.66

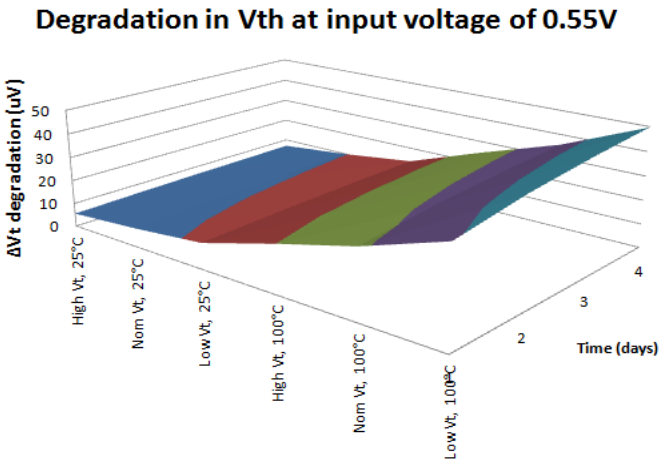


Figure 4-1. Degradation in V_{th} at voltage of 0.55V

Table 2. presents the V_{th} degradation values achieved for time period of one day to four days under different intial V_t and temperature values at a supply of 1.1V. From Figure 4-2. we can say that maximum V_{th} degradation is achieved at low initial V_t and high temperature.

Table 2. V_{th} degradation with Varation in Vt and T with V=1.1V

Time (days)	High Vt, 25°C (uV)	Nom Vt, 25°C (uV)	Low Vt, 25°C (uV)	High Vt, 100°C (uV)	Nom Vt, 100°C (uV)	Low Vt, 100°C (uV)
1	91.3	121	160.92	329.35	437.72	580.6
2	102.65	136.4	180.97	369.79	491.4	651.92
3	109.92	146.09	193.78	395.69	525.89	697.59
4	115.38	153.34	203.4	415.16	551.76	731.9

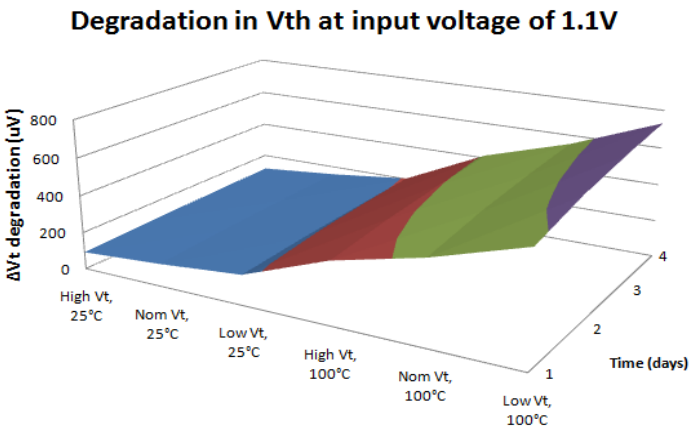


Figure 4-2. Degradation in V_{th} at input voltage of 1.1V

From Tables 1 & 2 and Figures 4-1 & 4-2 it is clear that NBTI degradation is worst at the corner: **P low, V high and T high** and best at **P high, V low and T low.**

CHAPTER 5

ONLINE NBTI WEAR-OUT ESTIMATION TECHNIQUE

This work proposes to perform NBTI wear-out estimation of the register file on a Leon 3 processor using run-time data from benchmarks. This is done by monitoring its 256 32-bit register file. The NBTI wear-out estimation is obtained from the RelXpert simulator [22], our baseline, and the RD model [8]. These values are then compared. The RD model is then tuned, using *time slicing* technique, to match the simulator.

5.1 Time Slicing Technique

The threshold voltage degradation over a period of 10 years is 50mV [14]. Our experiments show that implementing the RD model without tuning does not lead to this degradation. The reaction-diffusion process is frequency independent [16]. The same number of interface traps is generated (as discussed in Section 2) irrespective of the frequency of the PMOS's input. For example, if a periodic wave with time period of 1s and another with 2s, having same duty cycle, are applied as PMOS input for n seconds, they both will generate the same interface traps and thus cause the same V_{th} degradation (as shown in Figure 5-1(a)). The wave with period of 1s will initially generate less interface traps per cycle, but at the end of n seconds the number of interface traps generated will be same as the wave with period of 2s.

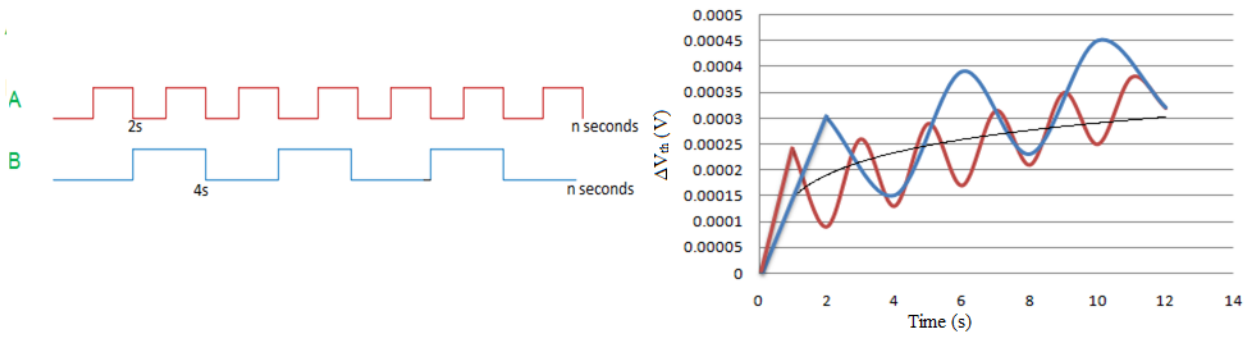


Figure 5-1(a) RD mechanism is frequency independent

The RD model is frequency dependent. Figure 5-1 (b) shows the plot when the V_{th} degradation was measured using the RD model (equations 1 & 2), inputs to the PMOS being the same as in Figure 5-1(a). It is clearly visible in Figure 5-1(b) that more V_{th} degradation is observed for the wave with less activity.

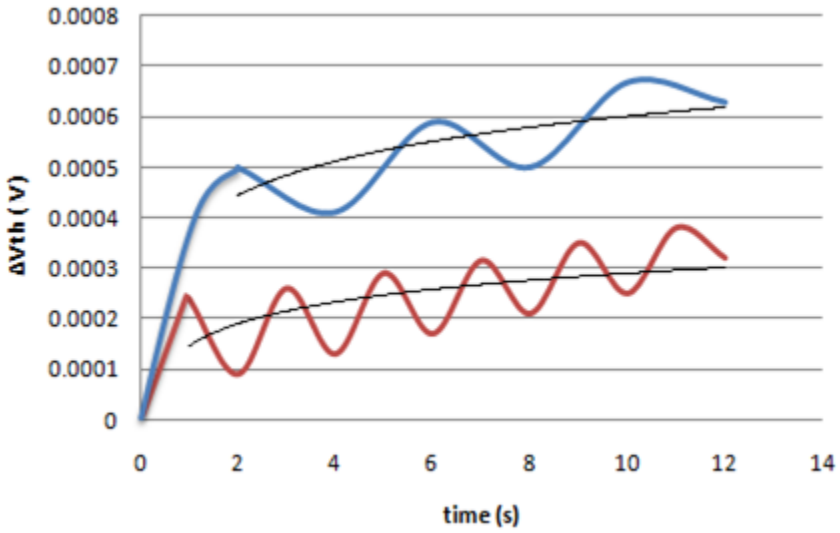


Figure 5-1 (b). RD model is frequency dependent

This shows that the RD model is somewhat imperfect and needs to be improved. Here a novel technique is presented which tunes the RD model that improves it and provides more accurate results. The RD model has an important property of being non-additive [15]. For example, if stress exists for time t and the stress equation is implemented in two ways, one for time 0 to t and other for 0 to t1 and t1 to t (in

two steps), where t1< t, the latter will give a higher threshold voltage degradation. For the latter case, at t1, it takes the history and then carries out stress from t1 to t. This is because more interface traps are generated as it takes the history at t1. As shown in Figure 5-2, V2 > V1, where V2 and V1 are the ΔV_{th} values resulted from the above two cases. We term this technique as *time slicing*, resulting in degradation near to [14]. This concludes that increasing the granularity of applying stress on a PMOS increases its NBTI degradation. This technique is used to tune to RD model to match the RelXpert Simulator.

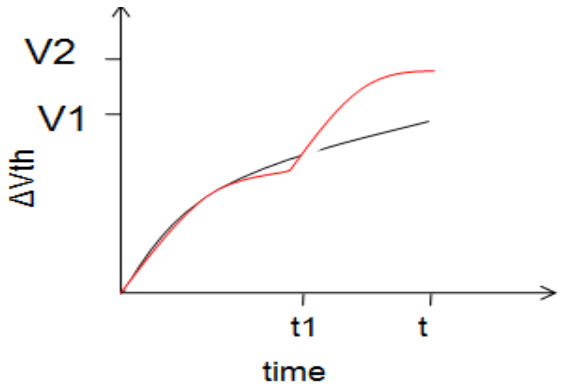


Figure 5-2. Time Slicing

5.2 Online NBTI Wear-out Estimation Technique

This section describes the flow of our methodology of Online NBTI Wear-out Estimation.

- Run benchmarks on Leon 3 processor
- Obtain Register File activity
- Carry out waveform compression
- Regenerate a random wave using the stored statistics
- Calculate V_{th} degradation for all the PMOS transistors in the register file using the RD model
- The activity waveforms go as input to the Ultrasim (for RelXpert) and SPICE (RD model) netlists
- The degraded V_{th} values also go into SPICE netlist

- Simulate both netlists and compare performance degradation (SNM)
- Tune the RD model, by time slicing, simulate, and match the Simulator

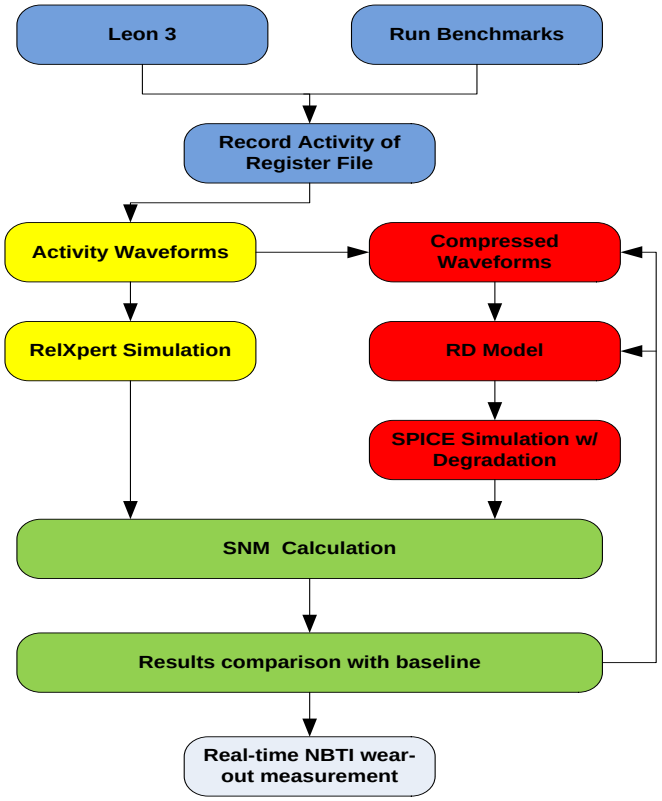


Figure 5-3. Online NBTI wear-out estimation technique

5.3 Results

This section presents results carried out on the Leon 3 DE2 design running Dhrystone benchmark on Modelsim [25]. The Leon3 has 256 32-bit register file, and we design them in SPICE Virtuoso using 6-T SRAM cells. The degradation measuring metric for our experiment is the Signal-to-Noise Ratio (SNM).

SNM is the minimum DC noise voltage necessary to change the state of a SRAM cell. It can be computed as the side of the length of the maximum square enclosed in the butterfly curve of a SRAM cell. Figure 5-4 shows the butterfly curves for a SRAM cell [8]. The length of the side of light grey square represents the fresh SNM. The length of the side of the dark grey square represents the degraded SNM. When the

PMOS degrades, or becomes weak due to NBTI, the butterfly curve of the SRAM cell shifts to the left. It shows that there is 14% SNM degradation due to NBTI.

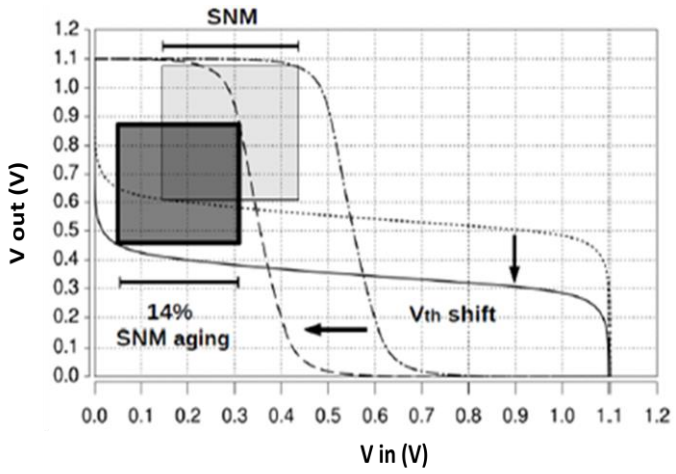
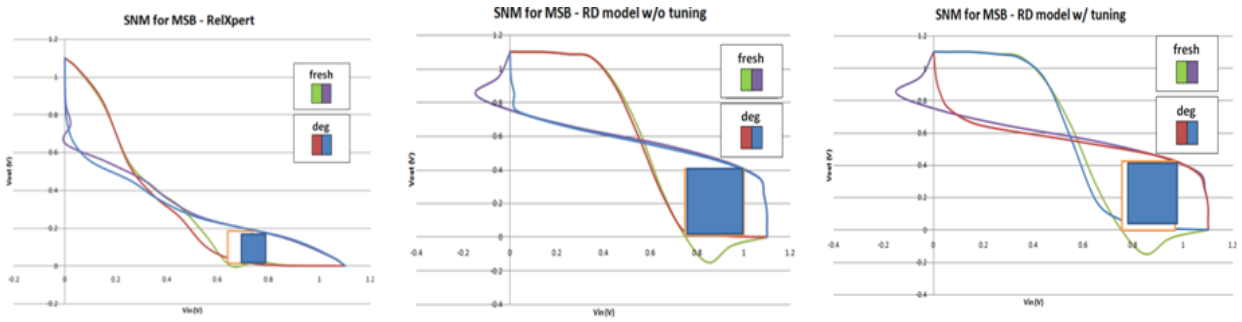


Figure 5-4. SNM degradation measurement of SRAM cell [21]

5.3.1 Online NBTI Wear-out Estimation technique running Dhrystone benchmark on Leon 3

We run the Dhrystone Benchmark (for time it runs once, i.e. 58ms) on Leon 3 DE2 design on Modelsim [25] and monitor the activity waveforms of its Register File. We calculate ΔV_{th} values, using equations 1 & 2 (discussed in Section 4.1), for both original and tuned RD models. Figure 5-5 shows the SNM degradation calculation from RelXpert simulator, the original RD model and the tuned RD model for bit 31 of Leon 3 register file. The RelXpert Simulator shows 9% SNM degradation, the original RD model with 2% and the tuned RD model shows 7% SNM degradation. This is carried for all bits of the Register File.



(a)From RelXpert simulator (b) From original RD model (c) From tuned RD model

Figure 5-5. SNM degradation calculation for Bit 31

Figure 5-6 shows the SNM degradation comparison between the RelXpert simulator, original RD model and the tuned RD model, for 1 year degradation, for few bits of the register file. It shows that the tuned RD model achieves about 80% accuracy with respect to the RelXpert simulator. The simulator takes an average of 48 hours for the simulation of each bit compared to 6 hours (5 hours for ΔV_{th} calculation and 1 hour for SPICE simulation) average with RD model. Thus carrying out NBTI wear-out estimation with RD model is 8 times faster than doing the same on the simulator.

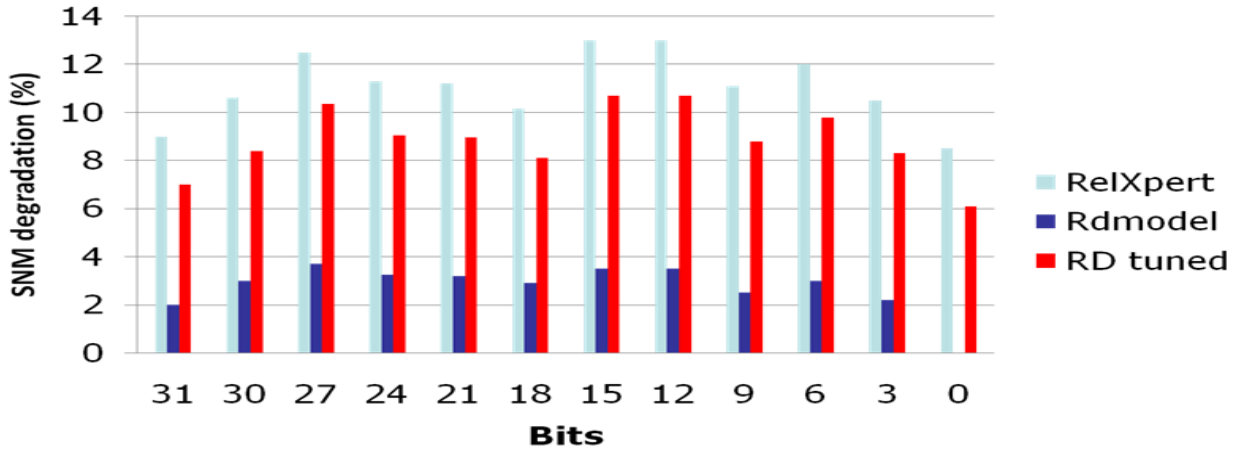


Figure 5-6. SNM degradation for RelXpert, RD model and tuned RD model

CHAPTER 6

WAVEFORM COMPRESSION AND ΔV_{th} ESTIMATION/PREDICTION TECHNIQUE

Section 5 presented a novel online NBTI wear-out estimation technique and showed how time slicing improves the accuracy of the RD model. The question is - "How this technique is going to impact the performance and cost of the system?" Monitoring the register file activity on a Leon 3 processor requires it to be stored in memory. Running the Dhrystone benchmark on the Leon 3 would require storing nearly 256MB memory for this activity, which greatly increases the cost of the system. NBTI is a long term mechanism, and the online prediction needs to be predicted after a long time period, say in years. Implementing equations 1 & 2 (discussed in Section 4.1) for such a long period would degrade the performance of the system. The dhrystone benchmark, running on the Leon 3, would take around 4 hours to predict ΔV_{th} for a day for a single bit, which is impracticable at run-time.

This section first presents *waveform compression* and *ΔV_{th} estimation/prediction* techniques, to overcome the huge cost and performance impact, respectively, caused by the online NBTI wear-out estimation technique presented in Section 5. Online NBTI wear-out prediction needs to be implemented on a real system. This section then proposes to implement a NBTI degradation predictor (NDP) on the Leon 3 in a DE2 FPGA [24], which predicts NBTI ΔV_{th} degradation for different Spec2000 benchmarks. This section further proposes to compare the actual on-chip degradation on the Leon3 in a DE2 FPGA, with the RD model. We do so by designing a n-stage (n is odd) Ring Oscillator on the Leon3 in a DE2 FPGA and measuring the frequency degradation due to NBTI. This is then compared to the RD model.

6.1. Waveform Compression Technique

As mentioned above, we need to overcome the huge memory cost resulting from the technique in Section 5. Here we propose a waveform compression technique which will significantly reduce this memory cost.

In this technique we propose to store just the statistics, i.e. mean and standard deviation, of the activity waveforms in the memory rather than storing the entire waveforms. Storing the entire waveforms (nearing 125K cycles) would take approximately 250KB of memory, for each register file bit. For the waveform compression technique, we need to store only the mean and standard deviation of 0s and 1s of the activity waveform for each bit. Here the mean and standard deviation values are stored in IEEE floating point single precision format (32 bits for each value). This only takes 128bits/register-bit (4x32).

For calculation of NBTI V_{th} degradation, these statistics are retrieved from the memory and a random wave is generated, on the fly, using random normal distribution. This wave should have good V_{th} degradation accuracy with respect to the V_{th} degradation obtained with original waveforms. Figure 6-1 shows the flow of the proposed waveform compression technique.

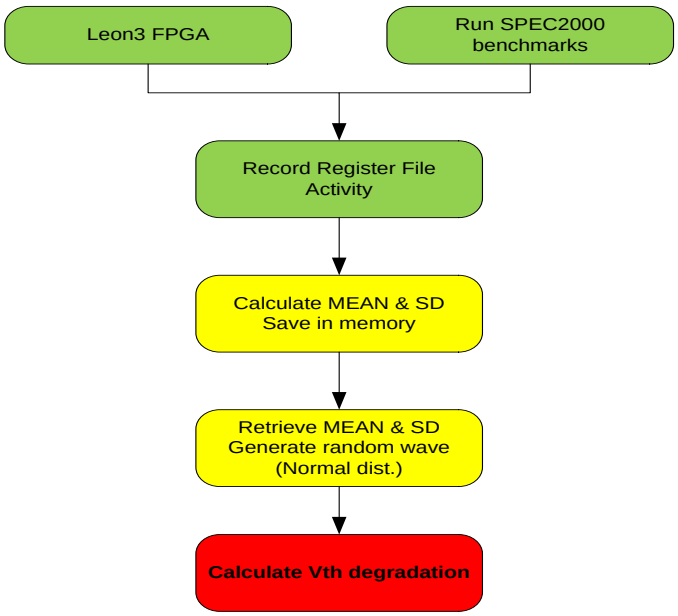


Figure 6-1. Waveform Compression Technique

6.1.1 Results

We carry out the Waveform Compression Technique experiment using the Register File activity waveforms derived by running Dhrystone Benchmark on the Leon3 using Modelsim [25].

We calculate the Mean and standard-deviation (SD) of the activity waveforms for each bit of the Register File. These statistics are calculated from the activity waveforms obtained by running the Dhrystone benchmark once (58ms) and can be stored in the memory. For V_{th} degradation calculation these statistics are retrieved from the memory and a random waveform is generated, on the fly, using random normal distribution. This randomly generated waveform is used to calculate V_{th} degradation. In our experiment this waveform provides 91% accuracy in V_{th} degradation calculation with respect to the tuned RD model. Figure 6-2 shows the V_{th} degradation for a few bits of the 256 32-bit register file of the Leon3, for uncompressed, compressed (retrieved with mean and standard-deviation) and compressed (retrieved with just mean) waveforms. The compressed waveforms generated using both mean and standard deviation provide a more accurate result compared to the one with just the mean.

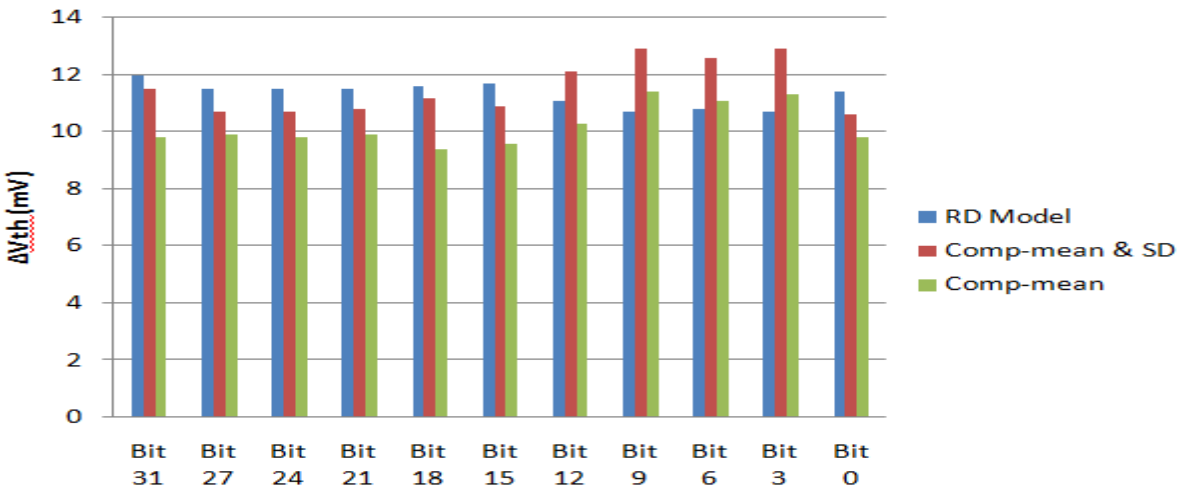


Figure 6-2. ΔV_{th} degradation for RD model, compressed (retrieved using mean and SD) and compressed (retrieved using just mean) giving an average accuracy of 91%

Storing the entire waveforms would take approximately 250KB of memory (i.e. the waveform flips 250K times or has 125K cycles), for each register bit. For the waveform compression technique, we need to store the mean and standard deviation of 0s and 1s of the activity waveform for each bit, which only takes 128bits/register-bit (4x32) or 16bytes/register-bit. For the whole register file it will consume

128x32x256 = 131KB of memory compared to 250Kx32x256 = 256MB of memory for the activity waveform without compression. Table 3 shows the comparison of memory needed to store activity with and without compression.

Table 3. Memory consumption for activity waveforms with and without COMPRESSION

Type	Without COMPRESSION	With COMPRESSION
One Register Bit	31KBytes	16 Bytes
One Register	1MBytes	512 Bytes
Whole Register File	256MBytes	131KB

6.2 ΔV_{th} estimation/prediction technique

Running the RD model for a long time, say 1 day, with fine granularity, during run-time would consume a lot of time, 4 hours for Dhrystone to predict ΔV_{th}, and degrade the performance of the system. Thus we need to come up with a technique which would predict the future NBTI V_{th} degradation by running the RD model only for a small period of time.

References [7,8,17,18,19,20] show that RD model works best in the range of 0s to 10^5 seconds. Thus, by running the RD model for a small period (say the time to run the application once), we can obtain degraded V_{th} values at selected points. Using ΔV_{th} Estimation/Prediction Technique we can build a curve and derive its function. The RD model exhibits a logarithmic trend due to continuous stress-recovery cycles. Figure 6-3. shows its logarithmic nature when a square wave with a period of 0.5s and a duty cycle of 50% is the PMOS's input. Thus a logarithmic function is preferable. We will prove that a second order logarithmic curve is the best fit to the ΔV_{th} points obtained from running the RD model.

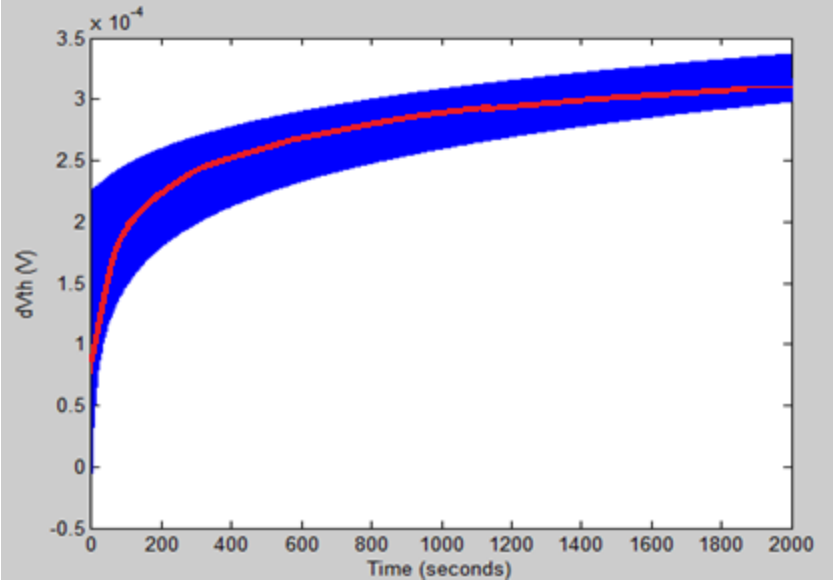


Figure 6-3. Logarithmic nature of RD model for a square wave with 2s period and 50% duty cycle

Just plugging in the number of years we can get the respective V_{th} degradation. Figure 6-4 shows Curving Fitting and obtaining the function to calculate future V_{th} degradation.

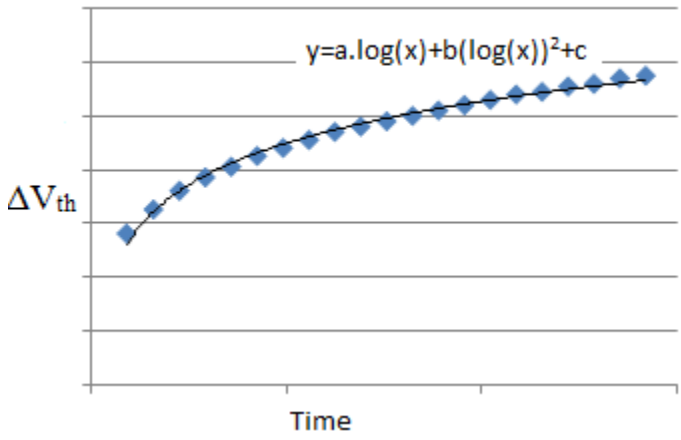


Figure 6-4. ΔV_{th} Estimation/Degradation Technique for ΔV_{th} prediction after plotting few data points. Here $y=\Delta V_{th}$ and x=time=t

6.2.1 Results

The flow of our experiment is as below;

- We run the RD model, for bit 3, for a duration of time the Dhrystone benchmark runs once (i.e. 58ms) and note down 21 V_{th} degradation values at regular time intervals and the respective time
- We plot these data points using a 'scatter plot'
- We use the Curve Fitting feature in to plot a fitting curve and derive an equation for the same
- The equation for bit 3 is $\Delta V_{th}=a.\log(x)+b.(\log(x))^2+c$ (Figure 6-5)
- We plug in any value of t and get the respective ΔV_{th}

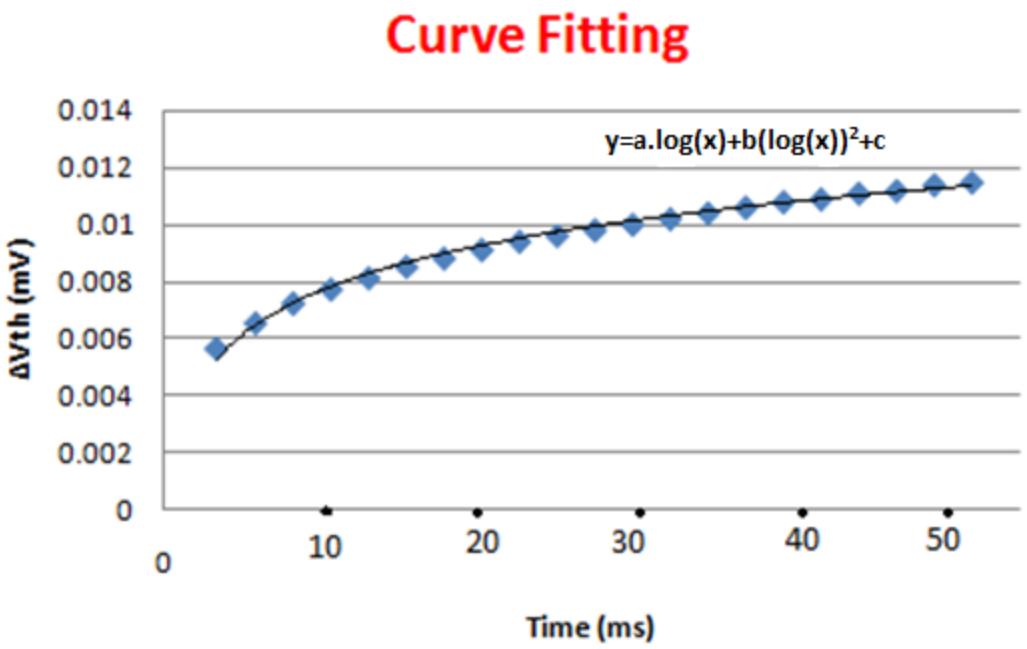


Figure 6-5. Curve Fitting for bit 3 for Dhrystone

Table 4. shows the % accuracy in ΔV_{th} after 1 day degradation, for few register file bits, after ΔV_{th} Estimation/Prediction Technique, by running Dhrystone benchmark on the Leon3.

Table 4. 93% accuracy is achieved in ΔV_{th} , for 1 day NBTI, after ΔV_{th} Estimation/Prediction Technique

BITS	ΔV_{th} (mV) (original RD Model)	ΔV_{th} (mV) (after Est./Pred.)	% Accuracy
0	44	47	93.5
3	44	47	93.5
6	46	50	92
9	42	47	89
12	49	51	96
15	49	52	94
18	41	47	87
21	47	49	96
24	48	50	96
27	53	55	96
31	40	44	91

From Figure 6-5 plugging in x=t=1year we get ΔV_{th}=47mV for bit 3. Running the RD model simulation without this technique provides ΔV_{th}=44mv. This results in 93.5% accuracy. Thus we get an average **accuracy of 93%** (from Table 4) for 21 data points**,** with respect to running RD model equations for a year.

Running the RD model for a period of 1 day would take approximately 4 hours to calculate V_{th} degradation. With this technique we can achieve the ΔV_{th} value in approximately **10 seconds**.

Thus the *ΔV_{th} estimation/prediction technique* is faster by an order of 10^2 with 93% accuracy with 21 data points.

6.2.1.1 Why we use Logarithmic curve fitting?

Once the ΔV_{th} data points are obtained by running the RD model for the duration Dhrystone runs once, we need to implement curve fitting to obtain the function to predict the future NBTI degradation. The question is, which function will best fit the these ΔV_{th} data points? Also how can we prove that the function we use is appropriate on physical basis, i.e. does the RD model/process trend follow the same of the chosen function, and also how accurate is it? First we find the trend of the RD model/process using NBTI physics and conclude which function follows this trend, so that it can be used to carry out curve fitting. Then we perform data analysis to measure accuracy.

We know that H_2 ions are released when the Si-H bonds at the Si-SiO_2 interface break under operation [8]. During stress these H_2 ions diffuse into the oxide in the reaction phase and do so in the poly-Si in the diffusion phase. The diffusion of H_2 ions in oxide is faster compared to that in the poly-Si. Due to the widely different diffusivity of H_2 in the oxide and poly-Si, the recovery becomes a two-step process, with fast recovery driven by H_2 in the oxide, followed by slow recovery of H_2 by backdiffusion from the poly-Si. The number of annealed traps can be due to two parts: 1) recombination of H_2 in the oxide and 2) backdiffusion of H_2 in the poly-Si. Due to this not all the H_2 ions are able to bond again with Si, to form Si-H bonds. Thus the number of interface traps generated during the next stress phase will be smaller compared to the first one. Also $\Delta V_{th} = \frac{qN_{IT}}{C_{OX}}$; where N_{IT} is the rate of interface trap generation, C_{OX} is the oxide capacitance, and q is the charge of holes. Due to this ΔV_{th} in each and every stress phase will be smaller than that in the previous one, as shown in Figure 6-6.

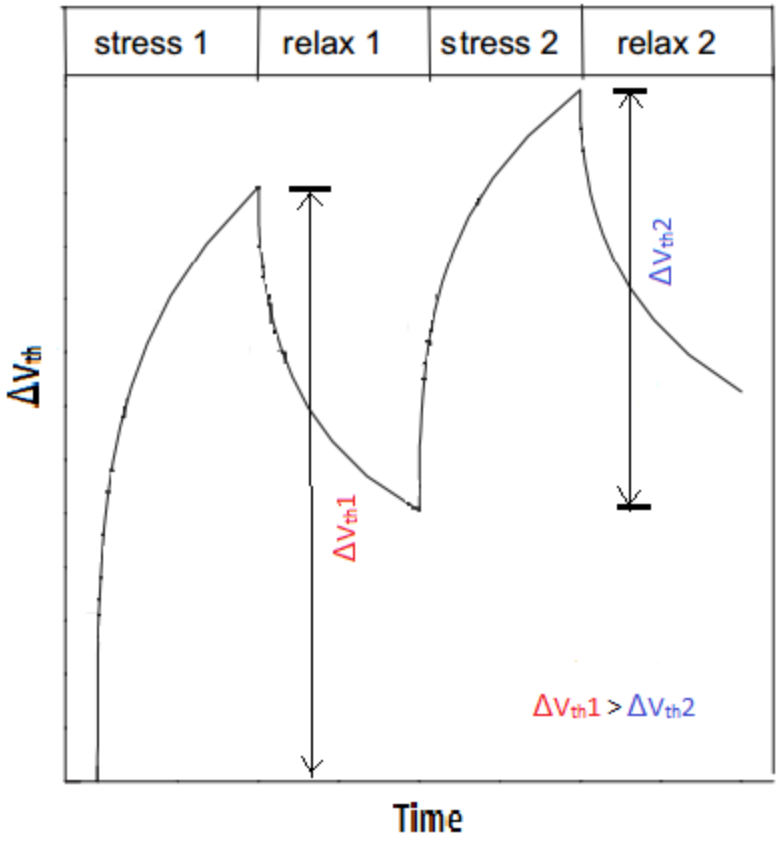


Figure 6-6. ΔV_{th} for second stress phase is smaller compared to that of first

The number of interface traps generated during stress is given by equation (3), and number of interface traps annealed during recovery is given by equation (4).

$$N_{IT}(t) = \left(\frac{\sqrt{k_H k_F N_o P}}{k_R} \right)^{\frac{2}{3}} \left((1+\delta) t_{OX} + \sqrt{D_{H_2} t} \right)^{\frac{1}{3}} \quad \text{.....equation (3)}$$

$$N_{IT}{}^{A}(t) = N_{IT}(t) \left(\frac{2\xi_1 t_e + \sqrt{\xi_2 C (t - t_1)}}{(1+\delta) t_{OX} + \sqrt{Ct}} \right) \quad \text{.....equation (4)}$$

t_1 is the time for which recovery takes place. All other parameters are RD model parameters.

Inserting the values of RD model parameters into equation (4) shows that the term $\left(\frac{2\xi_1 t_e + \sqrt{\xi_2 C(t-t_1)}}{(1+\delta)t_{OX} + \sqrt{Ct}}\right)$ < 1. Thus we can say that number of interface traps to be generated during the next stress phase will be lesser compared to the first one, due to not all the H_2 ions form bond with Si during recovery.

If we see this stress recovery process for a long-run, each stress phase will generate lesser interface traps compared to the previous one, and thus result in lesser ΔV_{th} for that cycle compared to all the previous ones. Thus the ΔV_{th} degradation for an RD model/process increases at a high rate first and then start becoming stable, or increases at a very small rate. We can say that this trend is very similar to the trend of the logarithmic function. Figure 6-7 shows the ΔV_{th} degradation trend for a 1Hz square wave generated from the RD model for a period of 1 year, which similar to the logarithmic function.

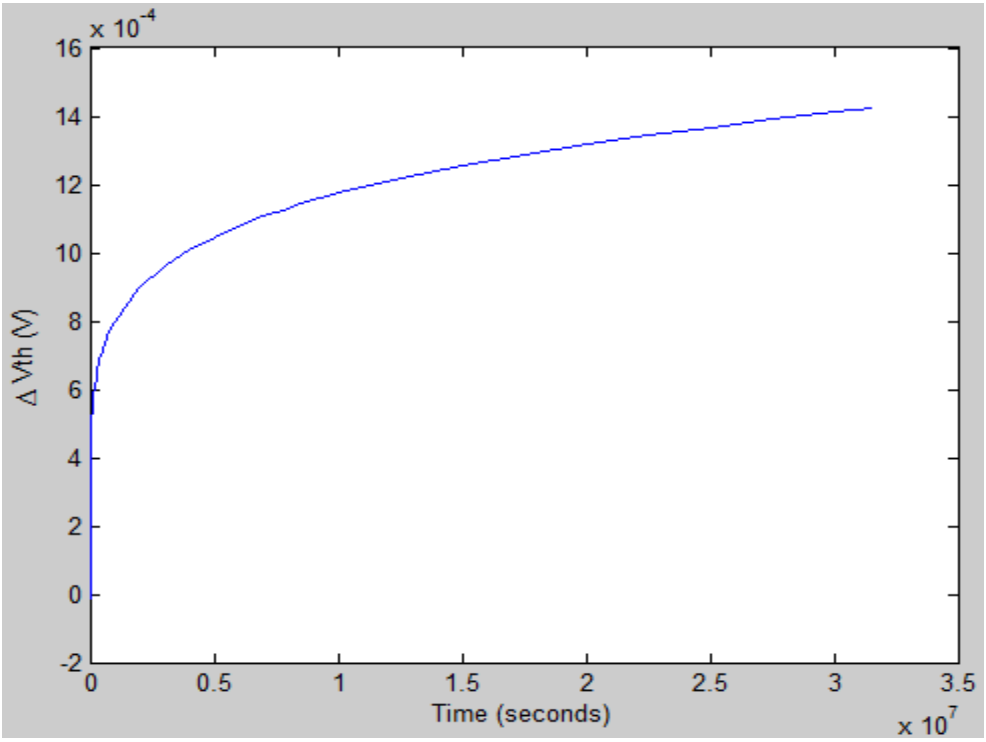


Figure 6-7. RD model trend for 1Hz square wave for 1 year degradation (similar to logarithmic)

Now we carry out data analysis for the curve plotted in Figure 6-7, and show that it is very close to the logarithmic function. There can be numerous functions which can fit a set of data points, but here we need to find one which does so with minimum residual, or error. Residual of an observed value is the difference between the observed value and the estimated function value. Figure 6-8 shows the residuals when any curve is fit for a set of data points.

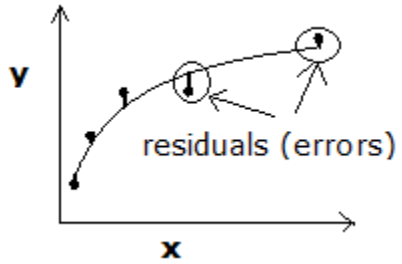


Figure 6-8. Residuals in curve fitting

When we carry out curve fitting, for a set of data points, using different functions, the best fit function will be the one with minimum residual (or error). The value R^2 quantifies goodness of fit. It is a fraction between 0.0 and 1.0, and has no units. Higher values indicate that the function fits the data points better. Thus if the value of R^2 is as near to 1, the chosen curve fits the data points better. When R^2=1.0, all points lie exactly on the curve with no scatter. R^2 is computed from the sum of the squares of the residuals, and this is denoted by SS_{res}, which is in the units of the Y-axis squared. To turn R^2 into a fraction, the result is normalized to the sum of the square of the distances of the points from a horizontal line through the mean of all Y values. This value is called SS_{tot}. So R^2 is calculated by the equation $R^2 = 1.0 - \frac{SS_{res}}{SS_{tot}}$. Thus the curve is best fit when SS_{res} is quite smaller than $SS_{tot.}$

We extract data points from the curve in Figure 6-7 and try to perform curve fitting. Figure 6-9 shows how the logarithmic function follows the trend of these RD model generated data points for 1 Hz

square wave for a period of 1 year. Here we achieve a value of R^2=0.9947, or accuracy of 99.47%. Thus we can say that the logarithmic function is 99.47% close to the RD model trend.

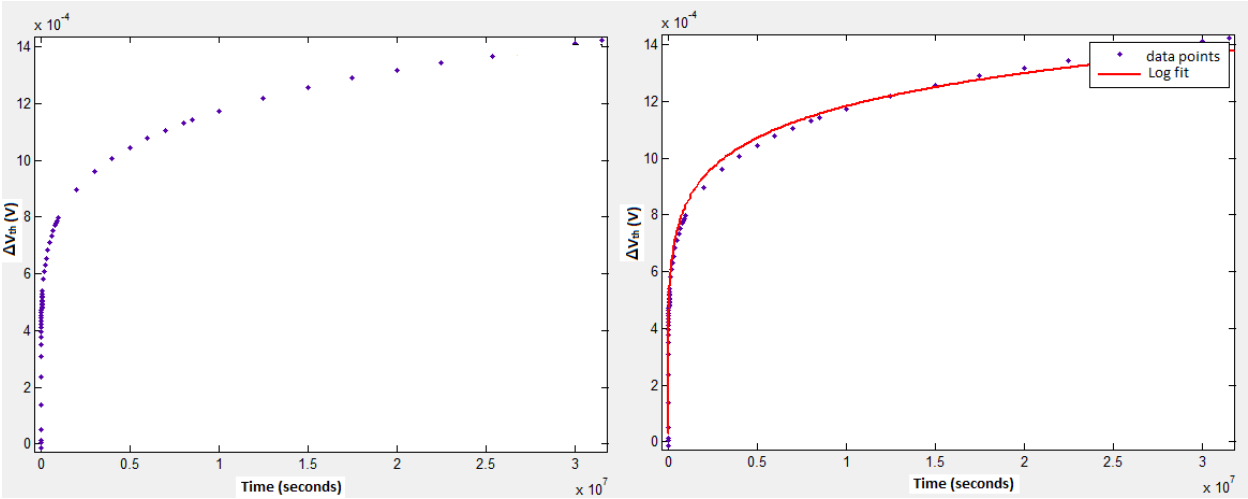


Figure 6-9. Logarithmic function fit has 99.47% accuracy

Table 5 shows the accuracy when curve fitting is carried out with different functions, for the data points obtained from RD model curve in Figure 6-7.

Table 5. R^2 values for different functions used to fit ΔV_{th} data points

Functions	R^2 value
Exponential	0.604
Linear Polynomial	0.6818
Quadratic Polynomial	0.8058
Cubic Polynomial	0.8480

Posynomial	0.8556
1st order Logarithmic	0.8732
2nd order Logarithmic	0.9947

6.3 NBTI degradation Predictor on the Leon 3 FPGA

In this section we showed results for the waveform compression and ΔV_{th} Estimation/Prediction techniques implemented on the Leon 3 running on Modelsim [25]. But these techniques should be implemented in a real system. We propose to implement a NBTI degradation predictor (NDP) on the Leon3 in a DE2 FPGA [24]. This module would predict the final ΔV_{th} degradation value, for the Leon 3 register file, after a specified time period. We also have a 41-stage ring oscillator placed at the four corners of the FPGA to measure the process variation across it.

The flow of this predictor is as follows and as in Figure 6-10.

- Run Spec2000 benchmarks on the Leon 3 FPGA
- Perform waveform compression technique on these waveforms and store the statistics
- Retrieve these statistics
- Carry out NBTI degradation estimation using curve fitting
- Predict the final ΔV_{th}, after a specified time period, using the curve fitting function

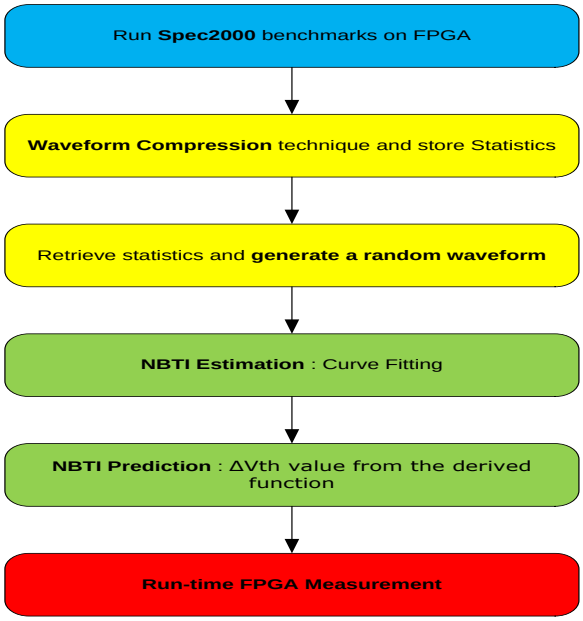


Figure 6-10. Proposed design of NBTI Degradation Predictor on the Leon 3 FPGA

6.3.1 Architecture

As mentioned above, we implement the NDP module in hardware to predict the run-time degradation of the register file on the the Leon 3 in a DE2 FPGA, running Spec2000 benchmarks. Figure 6-11 shows the proposed architecture of this module. It consists of 2 main sub-modules, viz. Activity Monitor module and Predictor module. It also consists of the on-chip memory block to store activity statistics and the register file to be monitored. We also plan to measure the actual frequency degradation, of N-stage ring oscillator (n=odd), on the Leon3 in a DE2 FPGA. This will then be compared to the results of the RD model.

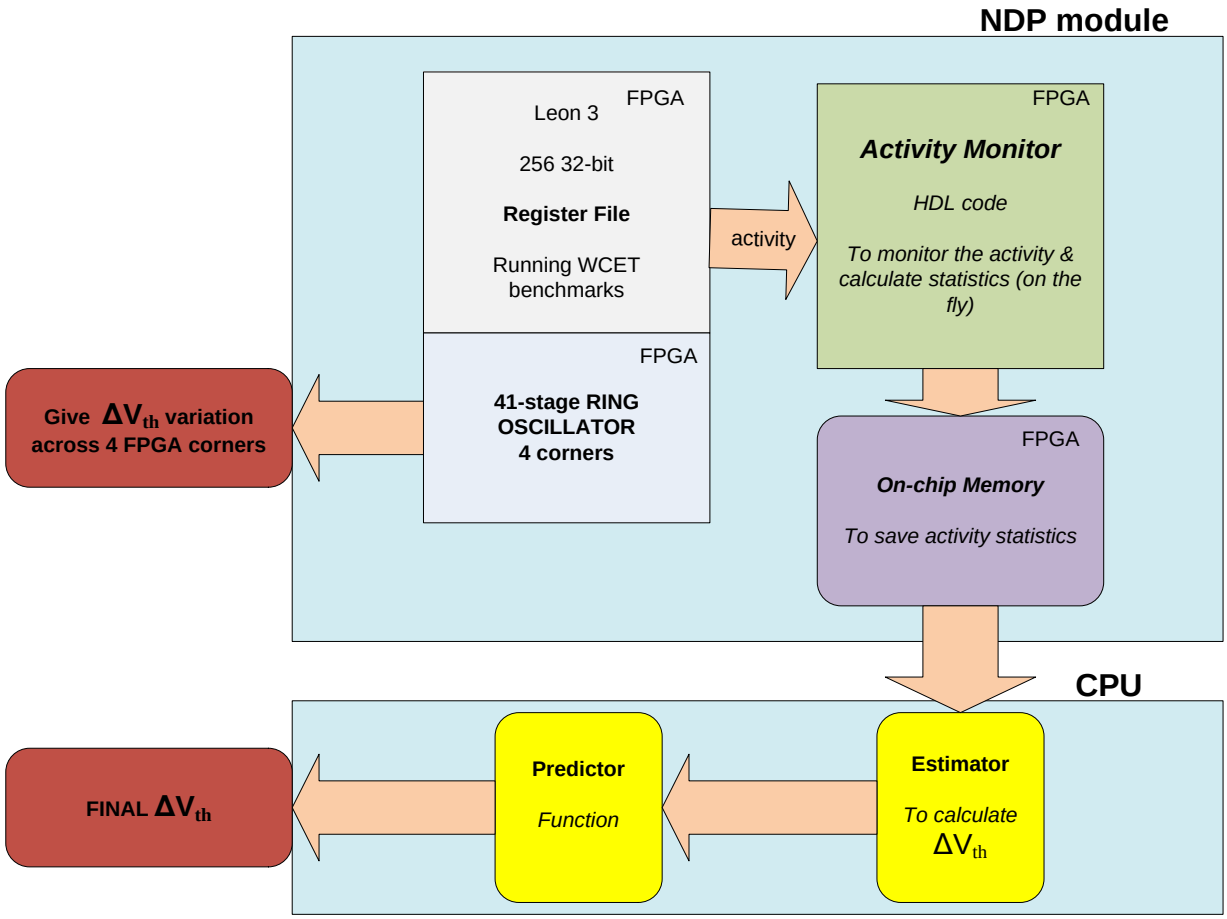


Figure 6-11. Proposed NDP module Architecture

6.3.2 Function of each blocks

This section describes the function of each block of the NDP module architecture proposed in 6.3.1.

- **Register-File**: This is the in-built 256 32-bit register file on the Leon 3 in a DE2 FPGA. This register file needs to be monitored through HDL coding to carry out its NBTI degradation prediction.
- **N-stage Ring Oscillator:** This is designed in hardware. The ring-oscillator waveforms will be monitored through HDL coding to finally measure the frequency degradation due to NBTI.
- **Activity Monitor:** This monitor is the HDL code (Verilog) to monitor the input of all the registers, while running Spec2000 benchmarks, to predict its NBTI wear-out. It monitors and captures the

waveforms of ring oscillator too. It also calculates the statistics, i.e. mean and standard variation (as discussed in section 6.1), of the waveforms and save them in the on-chip memory.

- **On-chip Memory:** It is used to store the statistics calculated by the monitor.
- **Estimator:** It runs the application (or benchmark) for a specific time (discussed in section 6.3.3), to collect specific number the ΔV_{th} data points at regular intervals (how many - discussed in section 6.3.3).
- **Predictor:** This block carries out ΔV_{th} Estimation/Prediction Technique (discussed in Section 6.2) to obtain the final ΔV_{th} functions for different register file bits. These functions are used to predict the future NBTI degradation.
- **Calculate Ring Oscillator frequency:** When the frequency of the ring oscillator is to be measured, the statistics, of the same, are retrieved. This measured frequency from the FPGA is then compared to the same achieved from the RD model.

The question is which particular register file bits will be monitored to predict degradation on the above proposed NDP module. In combinational logic circuits the critical path in a circuit will undergo maximum degradation, and it remains fixed. Here, we monitor the register file inputs to measure NBTI degradation of the PMOSs in each of the SRAM cells. Each SRAM cell degrades depending on the value and the time a value is stored in it. Each of them will undergo different amount of degradation for different applications. So their degradation is application dependent. Thus, it is in the hands of the designer to decide which particular register file bits will be monitored to predict NBTI degradation.

CHAPTER 7

DESIGN AND IMPLEMENTATION OF THE NBTI DEGRADATION PRIDICTOR (NDP)

This section explains the design and implementation of the NBTI Degradation Predictor (NDP) proposed in Section 6.3. The hardware design of the Leon3 Register File (RF) activity monitor and the ring oscillators are done in VHDL. We implement these hardware designs on the Leon3 processor in a DE2 FPGA.

The LEON3 processor is a synthesizable VHDL model of a 32-bit processor compliant with the SPARC V8 architecture [30]. The model is highly configurable and particularly suitable for SoC designs. The Leon3 is distributed as integrated parts of the GRLIB IP Library [31]. GRLIB IP is an open source library downloadable from www.gaisler.com.

7.1 Design of NDP

Here we discuss the design of the Leon3 RF online activity monitor and the ring oscillator units used to measure the NBTI history and calculate the process variation across the FPGA.

7.1.1 Online Activity Monitor

As discussed in Section 6.3.1, we need to monitor the activity of the register file of the Leon3 on a DE2 FPGA. This is done through an online activity monitor module designed in VHDL. Figure 7-1 shows how we insert the online activity monitor module into the VHDL Leon3 core. The top level entity file is 'leon3mp' and the Leon3 core file is named as '*leon3s*'. The register file '*regfile_3p*' is instantiated in the core. We insert the activity monitor module called '*actmonitor*', and instantiate it in the '*regfile_3p*'. Every time the register file cells are written, the input to the cross-coupled inverters in a SRAM cell changes, which plays the major role in their NBTI degradation. Thus by monitoring the activity of the register file SRAM cells, we can estimate/predict its NBTI degradation. Once the activity is known, we

can calculate its statistics, i.e. mean and variance, and store them into the memory, as discussed in earlier sections.

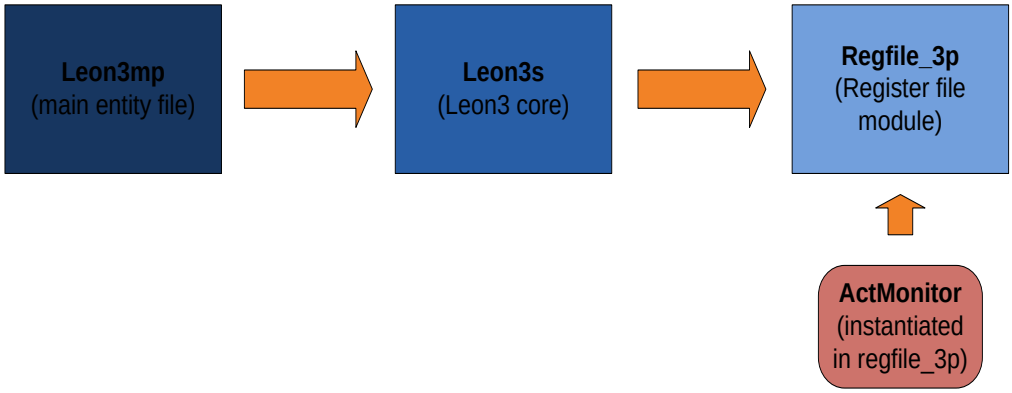


Figure 7-1. Inserting our Activity Monitor in the Leon3 VHDL core

We monitor the activity of the register file using counters. Figure 7-2 shows the design flow of the online activity monitor. To carry out NBTI estimation/prediction, we need to know the time the signal stays at value '0' and the time it stays at value '1'. On an FPGA, using VHDL, we can do this by counting the cycles of the clock, when the signal to be monitored is at value '0' or '1'. In this experiment, we do so by counting the positive edge of the clock when the register file cell activity signal is '0' or '1', using '*counter1*'. Say we are counting the amount of time the register file signal stays at '0'. The counter increments at every positive edge of the clock when the RF signal is '0'. When the RF signal is '1' the counter stops and the statistics, i.e. mean and variance, are calculated. There is a second counter, '*counter2*', which increments at every positive edge of the RF signal, to calculate the number of it's periods.

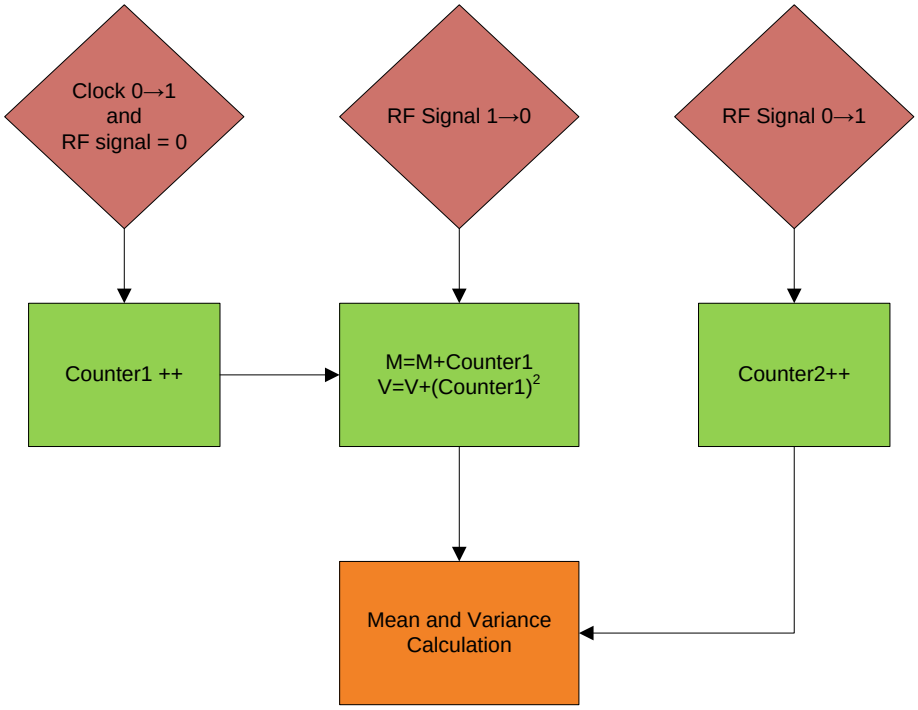


Figure 7-2. Leon3 Register File Activity Monitor design flow

The statistics represent the Mean and Standard Deviation of the time the RF signal stays at 0 and 1. They are calculated by the following formulas.

Mean: $$M = E(X) = \frac{\Sigma X}{n}$$

Standard Deviation: $$SD = \sqrt{E(X^2) - (E(X))^2}$$

where X is the number of transitions of the clock when RF signal is 0 or 1, and n is the number of transitions of the RF signal.

Once the statistics are calculated, they are stored into the on-chip memory. When estimation/prediction is carried, out these statistics are retrieved from the same memory to do the same.

7.1.2 ΔV_{th} Estimation/Prediction

As discussed above, when estimation/prediction is to be carried out, the RF signal statistics stored in the memory need to be retrieved. These statistics is based on the number of clock transitions when the RF signal is '0' or '1', but not the time for which it is at those values. Thus we need to multiply these number of transitions by the period of the FPGA clock, i.e. 0.02us. (50MHz clock of DE2 FPGA).

As discussed in Section 6.3.2, the estimator block collects ΔV_{th} data points from equations 1 and 2 (RD model equations) when applied to the random waveforms generated through statistics (waveform compression technique). These random waveforms are generated using the mean and standard variation calculated by the activity monitor module using random normal distribution. The predictor block then does the work of carrying out curve fitting, with the collected ΔV_{th} points, to plot an estimate trend of the future degradation and get its function. This function can be used to predict the future ΔV_{th} degradation for a given time. As we saw in Section 6.2, this method gives on an average 93% accuracy compared to the tuned RD model, and is 10^2 times faster for Drystone benchmark.

Using the statistics retrieved from the memory, random waveforms are generated using random normal distribution. These waveforms act as inputs in estimating the NBTI ΔV_{th} degradation using equations 1 and 2. Here we generate random waveforms for a time of 1 second and collect 20 ΔV_{th} data points at regular intervals for the same. These data points are used to carry out curve fitting as shown in Section 6.2. Hence, through curve fitting, we will get a function (2^{nd} order logarithmic in our case), which can be used to predict the future ΔV_{th} degradation.

7.1.3 History for NBTI ΔV_{th} estimation/prediction

We know that RD model is designed in such a way that it depends on the history, that is ΔV_{th} at the time we start estimation/prediction. Suppose we want to predict NBTI degradation from, say, 1 year to x years in future. For this we need the ΔV_{th} at the time 1 year, to input in equations 1 and 2, which we

term here as history. This history should be the actual current (1 year) ΔV_{th} of the particular device, register file in our case. Here we present a method to measure the actual current ΔV_{th} of the register file on the Leon3 in a DE2 FPGA.

To get this current ΔV_{th}, or history, of the Leon3 register file (RF) on a DE2 FPGA, we place a NAND gate 41-stage ring oscillator just besides the RF. We place this unit just besides the RF with the idea that it will experience approximately the same on-chip variations, like temperature, voltage, process etc., as the RF. NBTI also depends on run-time temperature. The ring oscillator, place besides the register file, will experience approximately the same run-time temperature as the register file. One of the input to the NAND gates in a ring oscillator is a control signal which selects between the module working as a ring oscillator or as buffer (holding the same value).

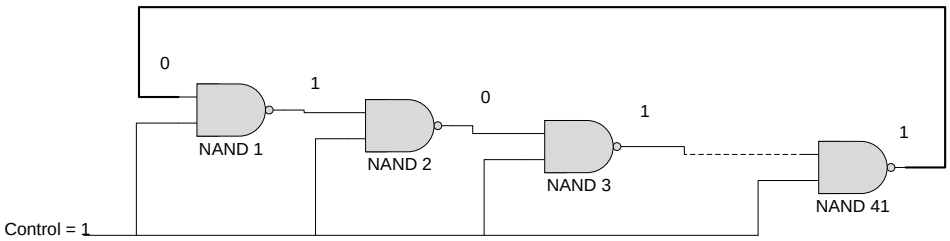


Figure 7-3. NAND gate closed loop circuit working as RO

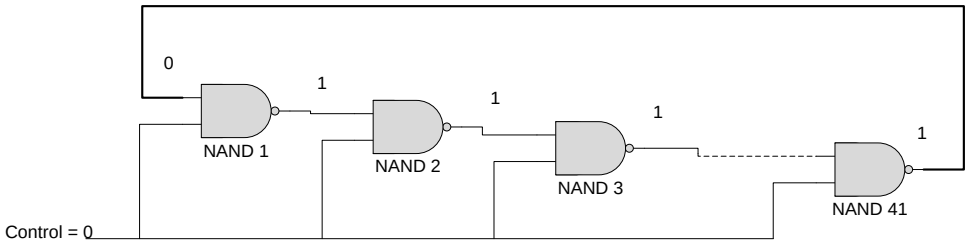


Figure 7-4. NAND gate closed loop circuit which holds a value

We all know that when the input to the PMOSs is 0 continuously, it degrades the most, and when it is 1, it recovers. The degradation in Figure 7-4 will be more compared to that in Figure 7-3. In Figure 7-4, one input will be continuous 0 and other will be continuous 1, which can work as the worst case scenario for a register file cell, where when the input on one side is 0, the one at the other side is 1. This happens when nothing is written onto the register file cell. Thus this 41-stage NAND gate ring oscillator, placed next to the Leon3 RF, works as a module to measure frequency degradation which can be converted into ΔV_{th} degradation due to NBTI through equation (5) [32]. During normal operation the select signal is 0 and while frequency measurement it is changed to 1, where it operates as a ring oscillator to make the measurement. Figure 7-5 shows the Chip Planner in Altera Quartus. The shaded part is the register file, and the greenish-blue 3 labs in that shaded part is the 41-stage NAND gate ring oscillator.

$$\Delta V_{th} = \frac{\Delta f\,(V_{gs} - V_{th})}{f\alpha.} \quad \text{.....} \quad \text{equation (5)}$$

where Δf is the change in frequency from start to finish of running the benchmark, and f is the original frequency when we start running the benchmark. α is the velocity saturating index with a value of 1.3.

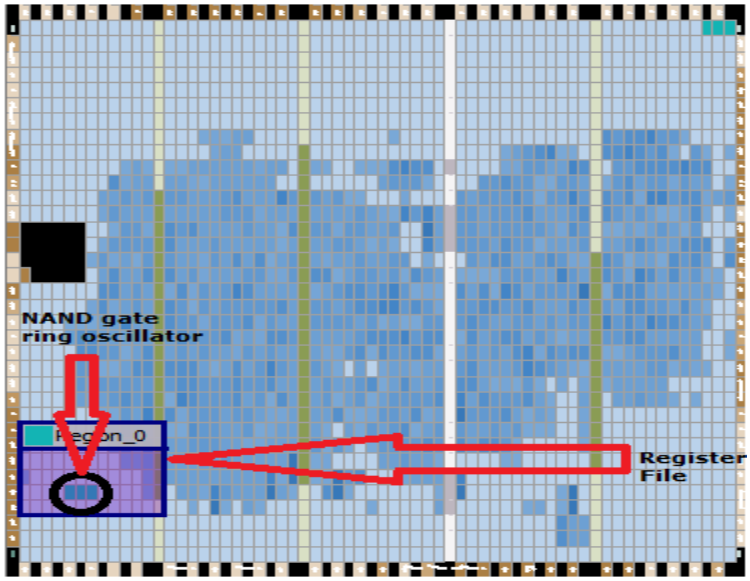


Figure 7-5. Chip Planner in Altera II Quartus showing the RF and the RO placed next to it

We implemented this module on the DE2 FPGA and ran the WCET benchmark suite [39]. We measure the frequency before running the benchmark and again after running it for an hour. The frequency degradation, Δf, which we calculate from the frequency measurements is due to NBTI + temperature. Here we need to cancel out the temperature effect on the frequency degradation. Thus we run the ring oscillator (with control signal = 1), with WCET benchmarks, for first 1 hour, so that the temperature reaches a stable point. Once the temperature becomes stable we again run benchmarks with worst case (control signal = 0), and make frequency measurements at $t_{60} = t + 60$ and $t_{120} = t + 120$. Here t is the time from when the chip was manufactured till the time we start running the benchmark, and this is unknown, which we need to find out to determine the respective ΔV_{th}. From equation (5), ΔV_{th} is mainly dependent on Δf, where $\Delta f = f_{current} - f_{initial}$, where $f_{initial}$ is unknown. Here we propose a technique to measure and calculate the actual ΔV_{th} value which can be used as history.

As discussed above we measure the values of frequency of the NAND gate ring oscillator at t_{60} and t_{120}. We calculate the respective ΔV_{th} values from equation (5) as ΔV_{th60} and ΔV_{th120}. We can find the rate of change in ΔV_{th} between ΔV_{th60} and ΔV_{th120}, and we denote it as '$r_{60\text{-}120}$'. We know that NBTI ΔV_{th} degradation is frequency independent from 1Hz to 2GHz [16]. We generate a RD model ΔV_{th} degradation curve for 1 year for a 1Hz square wave, as shown in Figure 7-6.

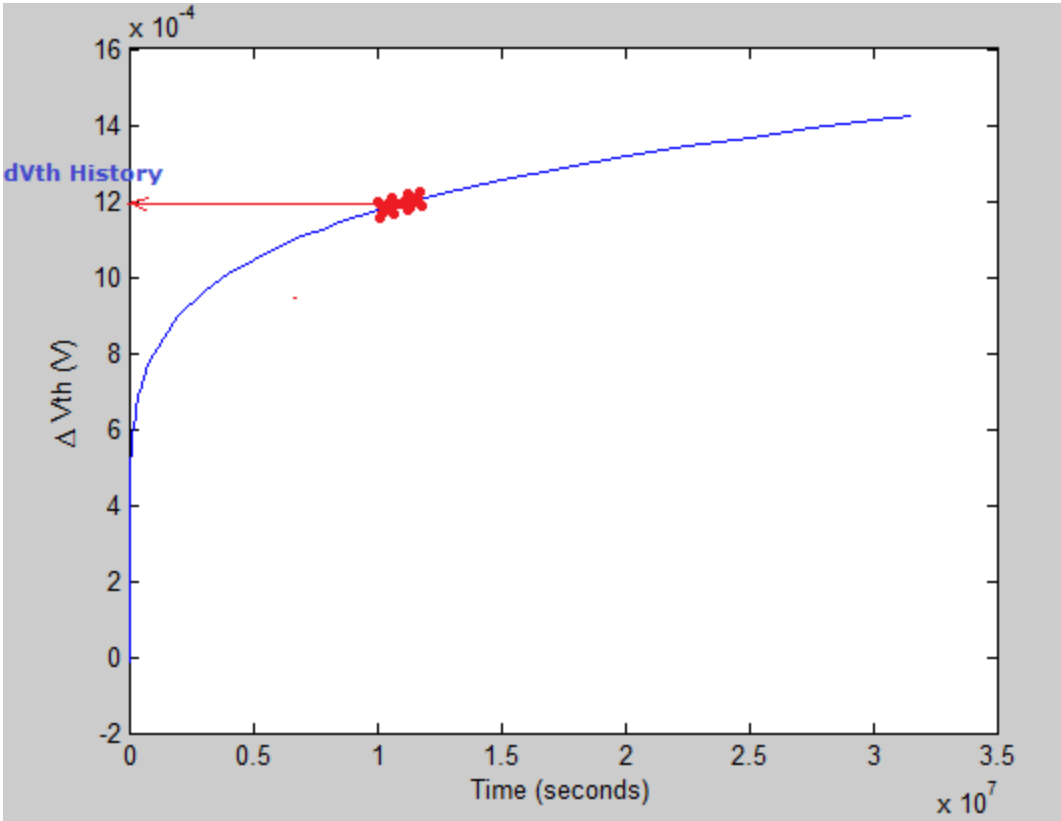


Figure 7-6. Matching $r_{60\text{-}120}$ with the 1Hz RD model degradation curve

Here we try to find a rate on the curve in Figure 7-6, which matches $r_{60\text{-}120}$. When we get the same, the respective ΔV_{th} values on Y-axis will be the history at t_{60} and t_{120}. This ΔV_{th} history can be plugged into the RD model to predict the future degradation.

7.1.4 Variation in ΔV_{th} across the FPGA

Technology scaling has resulted into worsening the process, voltage, and temperature (PVT) variations across any microchip [33]. The demand for low power causes supply voltage scaling and hence making voltage variations a significant part of the overall challenge. Also, the quest for growth in operating frequency has manifested in significantly high junction temperature and within die temperature variation. Due to manufacturing process variations, the initial threshold voltage also varies across the

chip which results in process variation. Process variations result in variations in frequency and leakage across the chip. This can result in the same circuit at different places across the chip performing differently. This variation problem is a big concern for technology beyond 90nm [33]. The within die process variations can be characterized into systematic (process shift) and stochastic (process spread) [34]. Systematic variations can be caused by inaccuracies in process model, lithographic off-axis focusing errors, etc. Stochastic variations are caused by sources like vibrations during lithography, wafer unevenness and non-uniformity in resist thickness. The frequency variations due to on-chip process variation are measured across the Cyclone II EP2C35 device in [34]. It presents an array of ring oscillators connected with each other, with each ring oscillator places at different places across the EP2C35 device. Figure 7-7 shows the frequency variation of the same. These frequency variations across the EP2C35 device are due to the process variation across it.

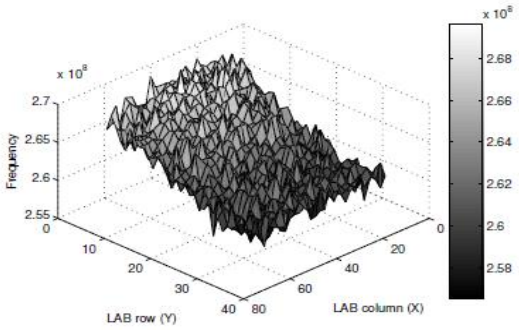


Figure 7-7. The observed frequency of each RO in a single EP2C35 device [34]

It would be really helpful for the designer to know these process variations across the chip while designing any unit. Here we present an initial ΔV_{th} measuring method to know the process variation across the four corners of a single Leon3 on a DE2 FPGA. This can help us to measure the Initial threshold voltage change, $\Delta V_{th\text{-}initial}$, at the four corners of the Leon3.

We place 41-stage NOT gate ring oscillators at the 4 corners of the Leon3 in a DE2 FPGA. The threshold voltage across these corners will be different due to variations in manufacturing process. These 4 ring oscillators will report different frequencies when we run the Leon3 on a DE2 FPGA. We run these ring oscillators for a period of 4 hours, measuring frequency at different time intervals. Change in frequency from t_0 (time when the ring oscillators start running) to t_n (time we measure the frequencies) can be calculated, which we term as Δf. The respective change in threshold voltage, ΔV_{th}, can be calculated from equation 3.

$$\Delta V_{th} = \frac{\Delta f (V_{gs} - V_{th})}{f\alpha.} \quad \text{.....} \quad \text{equation 3}$$

When we run the ring oscillator, we do so from t_0 to t_n, where $t_0 = t + 0$ and $t_n = t + n$. Here t is the time from when the chip was manufactured till the time we start running the ring oscillator, and this is unknown, which we need to find out to determine the respective ΔV_{th}. From equation 3, ΔV_{th} is mainly dependent on Δf, where $\Delta f = f_{current} - f_{initial}$, where $f_{initial}$ is unknown. We follow the same technique presented in Section 7.1.3, to find out the change in threshold voltage at t_0, which we can term as the $\Delta V_{th\text{-}inital}$. We can find the rate of change in ΔV_{th} between ΔV_{th0} and $\Delta V_{th\text{-}n}$, and we denote it as '$r_{0\text{-}n}$'. We know that NBTI ΔV_{th} degradation is frequency independent from 1Hz to 2GHz [16]. We generate a RD model ΔV_{th} degradation curve for 1 year for a 1Hz square wave, as shown in Figure 7-8.

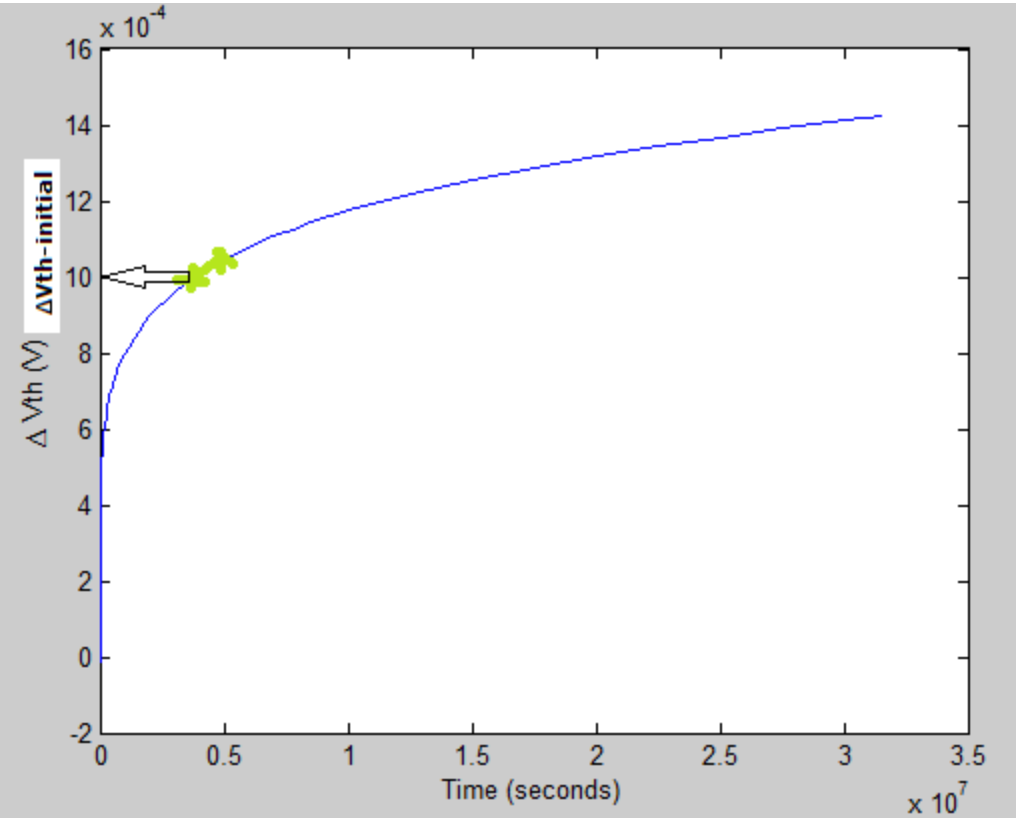


Figure 7-8. Finding $\Delta V_{th\text{-}inital}$ across the 4 corners of the FPGA

We try to find a rate on the curve in Figure 7-8, which matches $r_{0\text{-}n}$. When we get the same, the respective ΔV_{th}, at t_0, on Y-axis will be $\Delta V_{th\text{-}inital}$.

7.2 Implementation of NDP on the Leon3 in a DE2 FPGA

The Leon3 is compiled and synthesized using Altera Quartus II [35]. Altera Quartus II provides everything needed to design with FPGAs, SoCs, etc.. It is a complete development package that comes with a user friendly GUI and best-in-class technology to help ideas bring into reality. Compiling and synthesizing will form a .qsf file of the core, for example, in our case leon3mp.qsf. Next we need to form the image of the synthesized core, which can be loaded into the DE2 FPGA board. This needs a Cygwin [36] environment.

The command 'make quartus' will form the image with a file named, leon3mp.sof (in our case), which can be loaded onto the FPGA board.

Again Altera Quartus II is used to load the Leon3 image onto the FPGA. In Quartus, select Tools->Programmer, and select the .sof image file from the design directory. We connect the FPGA board with the computer, with a JTAG in our case, and load the Leon3 onto the board. We use the Aeroflex Gaisler GRMON2 [37] debugger to load and run the benchmarks on the DE2 FPGA.

7.2.1 Altera DE2 Development and Education board

After designing we implement the NBTI Prediction Module on the Leon3 Processor in an Altera DE2 FPGA board [39]. Figure 7-9 shows the layout of the Altera DE2 board. The highlighted pins are the ones which we use in our experiment.

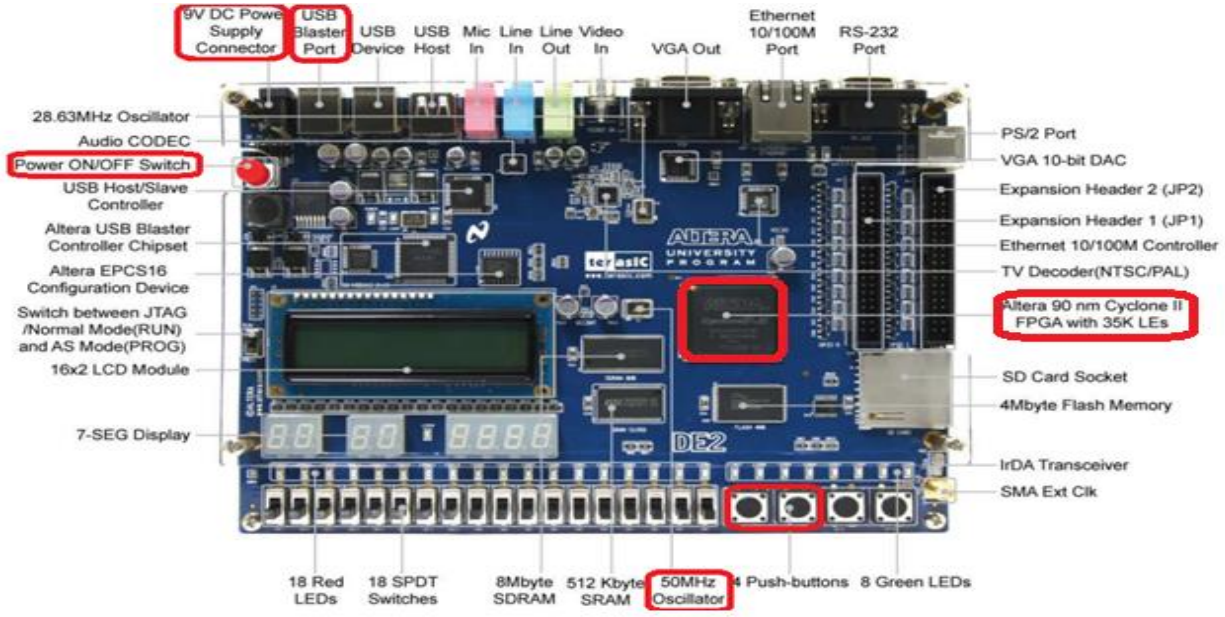


Figure 7-9. Layout of Altera DE2 Development and Education Board [39]

Power ON/OFF Switch: Turn ON/OFF the board

9V DC Power Supply Connector: We connect this to the power supply through an adapter

USB Blaster Port: We connect this to the computer through JTAG for downloading and debugging

50MHz Oscillator: We use this oscillator as clock in our design

Altera 90nm Cyclone II FPGA: FPGA chip

7.2.2 Benchmarks used

To carry out wear-out estimation/prediction, we need to use a benchmark suite. In our experiments we use the WCET benchmark suite [39], to carry out wear-out estimation/prediction of the Leon3 register file in a DE2 FPGA. It is primarily a numerical benchmark suite. Following are the benckmarks from the WCET benchmark suite, which me make use of in our experiments.

ADPCM	Adaptive Pulse Code Modulation algorithm
COMPRESS	Data Compression Program
BS	Binary Search
JFDCTINT	Discrete-cosine transformation on a 8x8 pixel block
NS	Search in a multi-dimensional array
NSICHNEU	Simulate an extended petri net
STATEMATE	Automatically generated code
UD	Calculation of matrices
NDES	Complex embedded code
MINVER	Inversion of float point matrix

The source C codes for all these WCET benchmarks are downloaded from [39], and then compiled using the Bare-C Cross-Compiler (BCC) System for Leon3 gcc-3.4.4 [40]. Compiling the C codes of the above benchmarks using this compiler will generate a binary file, which can be loaded onto the Leon3 core in a DE2 FPGA.

7.2.3 Debugger to enter the DE2 FPGA environment

To work on the FPGA environment we use the Aeroflex Gaisler GRMON2 debugger [37]. GRMON is a general debug monitor for the LEON processor, and for SOC designs based on the GRLIB IP library. Only LEON 3 and later are supported. We connect the DE2 FPGA board to the system through a JTAG cable. Through the debugger, we can enter the FPGA environment using the command './grmon.exe -jtag' for Windows. After entering the FPGA environment the system information can be obtained through the command 'info sys', as shown in Figure 7-10.

```
grmon2> info sys
info sys
  cpu0      Aeroflex Gaisler  LEON3 SPARC V8 Processor
            AHB Master 0
  ahbuart0  Aeroflex Gaisler  AHB Debug UART
            AHB Master 1
            APB: 80000700 - 80000800
            Baudrate 115200, AHB frequency 50.00 MHz
  ahbjtag0  Aeroflex Gaisler  JTAG Debug Link
            AHB Master 2
  svga0     Aeroflex Gaisler  SVGA frame buffer
            AHB Master 3
            APB: 80000600 - 80000700
            clk0: 25.00 MHz  clk1:   inf MHz  clk2:   inf MHz  clk3:   inf MHz
  mctrl0    European Space Agency  LEON2 Memory Controller
            AHB: 00000000 - 20000000
            APB: 80000000 - 80000100
            8-bit prom @ 0x00000000
  apbmst0   Aeroflex Gaisler  AHB/APB Bridge
            AHB: 80000000 - 80100000
  dsu0      Aeroflex Gaisler  LEON3 Debug Support Unit
            AHB: 90000000 - A0000000
            AHB trace: 128 lines, 32-bit bus
            CPU0:  win 8, hwbp 2, itrace 128, V8 mul/div, lddel 1
                   stack pointer 0x407ffff0
                   icache 2 * 4 kB, 32 B/line rnd
                   dcache 1 * 4 kB, 16 B/line
  sdctrl0   Aeroflex Gaisler  PC133 SDRAM Controller
            AHB: 40000000 - 40800000
            AHB: FFF00100 - FFF00200
            32-bit sdram: 1 * 8 Mbyte @ 0x40000000,
            col 8, cas 2, ref 7.8 us
  uart0     Aeroflex Gaisler  Generic UART
            APB: 80000100 - 80000200
            IRQ: 2
            Baudrate 38343
  irqmp0    Aeroflex Gaisler  Multi-processor Interrupt Ctrl.
            APB: 80000200 - 80000300
  gptimer0  Aeroflex Gaisler  Modular Timer Unit
            APB: 80000300 - 80000400
            IRQ: 8
            16-bit scalar, 2 * 32-bit timers, divisor 50
  adev11    Aeroflex Gaisler  LCD Controller
            APB: 80000400 - 80000500
  ps2ifc0   Aeroflex Gaisler  PS2 interface
            APB: 80000500 - 80000600
            IRQ: 5
  gpio0     Aeroflex Gaisler  General Purpose I/O port
            APB: 80000900 - 80000A00
  gpio1     Aeroflex Gaisler  General Purpose I/O port
            APB: 80000A00 - 80000B00
```

Figure 7-10. Debug window using Aeroflex Gaisler GRMON2 debugger

7.2.4 Loading and running the Leon3 core and benchmarks onto the FPGA board.

The image of the Leon3 core generated can be loaded using Altera Quartus II. It is done by loading the .sof file generated from the Cygwin environment, using Tools->Programmer in Altera Quartus II. Once the core is loaded onto the FPGA board, we can enter its environment using the GRMON2 debugger. Once entered into the FPGA environment, we load the benchmarks on to the board using the command 'load benchmarkname.exe'. To verify whether the program is loaded properly we can specify 'verify benchmarkname.exe'. The 'run' command will start running the program on the Leon3 in a DE2 FPGA.

In our experiments we need to display the register file data onto the screen. The Leon3 has a 8 windowed register file. The data of each of the windows can be viewed using the command 'reg w#', where # is the register window number 0 to 7, as shown in Figure 7-11.

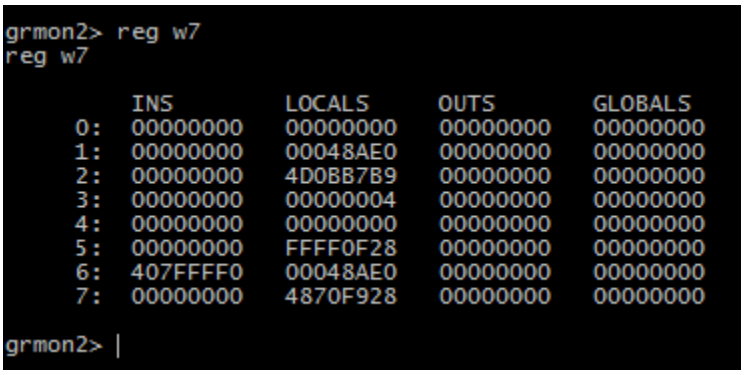


Figure 7-11. Register window 7 using the debugger

7.2.5 Displaying calculated statistics onto the debugger screen

Section 7.1.1 described the design of the online activity monitor to calculate the register file signal statistics. But we need to display this data, i.e. mean and variance, on the debugger screen. We should do so without affecting the ongoing process in the register file. Thus, we plan to keep the mean and variance data in a shadow register, which will be displayed in the register window on the debugger screen for particular register file read addresses. The idea is as shown in Figure 7-12. For example, if the register file read address, 'ra', is 00000001H, we will bypass the shadow register window and display in the register window instead of the data from the register file. For this we design a multiplexer which selects between the register file data and the shadow register data to be read. If ra=00000001H, select line will be 1 and the multiplexer will choose the shadow register data to be displayed in the register window, otherwise it chooses the register file data to do the same.

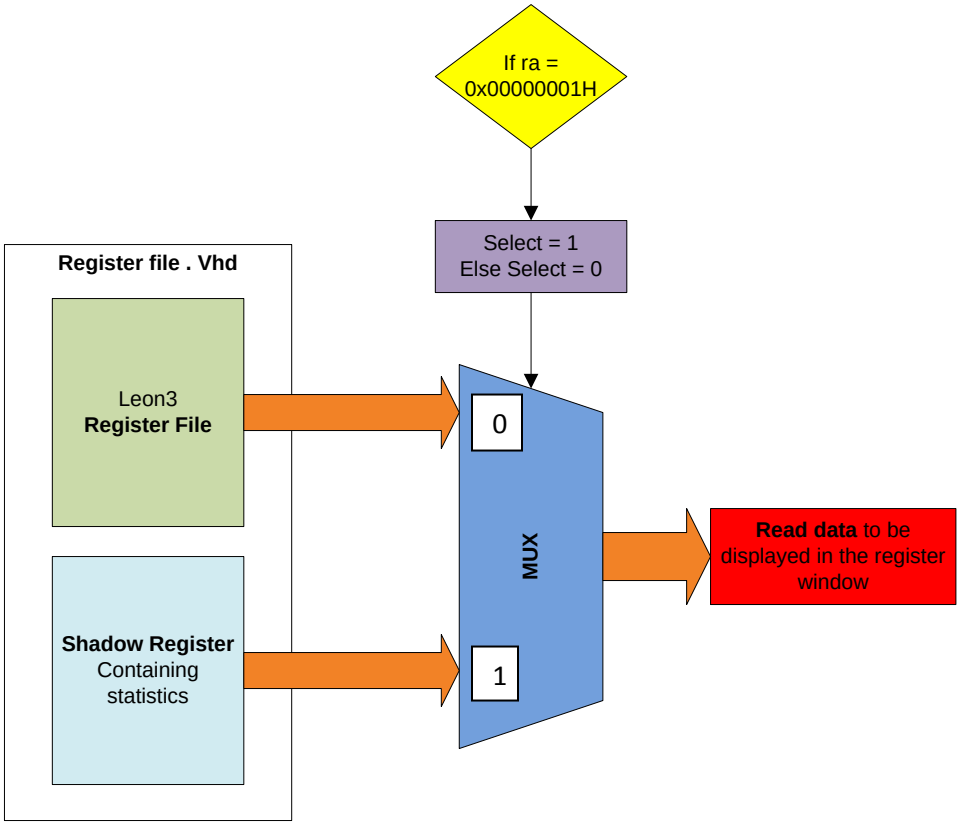


Figure 7-12. Shadow Register displaying statistics in the register window.

CHAPTER 8

RESULTS OF NDP AND MEASURING PROCESS VARIATION TECHNIQUE

We discussed about the design and implementation of the NBTI Degradation Predictor (NDP) in Section 7. Here we present the results when this NDP is implemented on a Leon3 Processor in a DE2 FPGA. Section 8.1 presents results from the 41-stage NAND gate ring oscillator placed besides the Leon3 register file to measure the history. Section 8.2 will show the average NBTI degradation for LSB and MSB bits of the Leon3 register file running various WCET Benchmarks. Section 8.3 will present the varying initial ΔV_{th} across the four corners of the FPGA due to process variations.

8.1 Measuring History for NBTI ΔV_{th} estimation/prediction

Section 7.1.3 presented the technique of measuring the actual current ΔV_{th} which can be used as history to carry out NBTI estimation/prediction using the RD model. Here we present the results for the same, when running the WCET benchmark suite for 1 hour, and capturing the frequency of the 41-stage NAND gate ring oscillator at t_{60} and t_{120}, where t_{60} is the time when we start running the benchmarks and t_{120} is the time we finish. Using this we calculate the respective ΔV_{th} at t_{60} and t_{120}, using equation (5), and measure the rate of change in ΔV_{th} degradation, denoted as $r_{60\text{-}120}$.

$$\Delta V_{th} = \frac{\Delta f (V_{gs} - V_{th})}{f\alpha.} \quad \text{.....} \quad \text{equation (5)}$$

Next a 1Hz RD model degradation curve is generated (Figure 8-1), and two points are found out on the same whose rate of change in ΔV_{th} degradation is same as $r_{60\text{-}120}$. The respective ΔV_{th} value on the Y-axis

will give us the actual ΔV_{th} degradation at t_{120}, from the time of chip manufacture, which can be used as history.

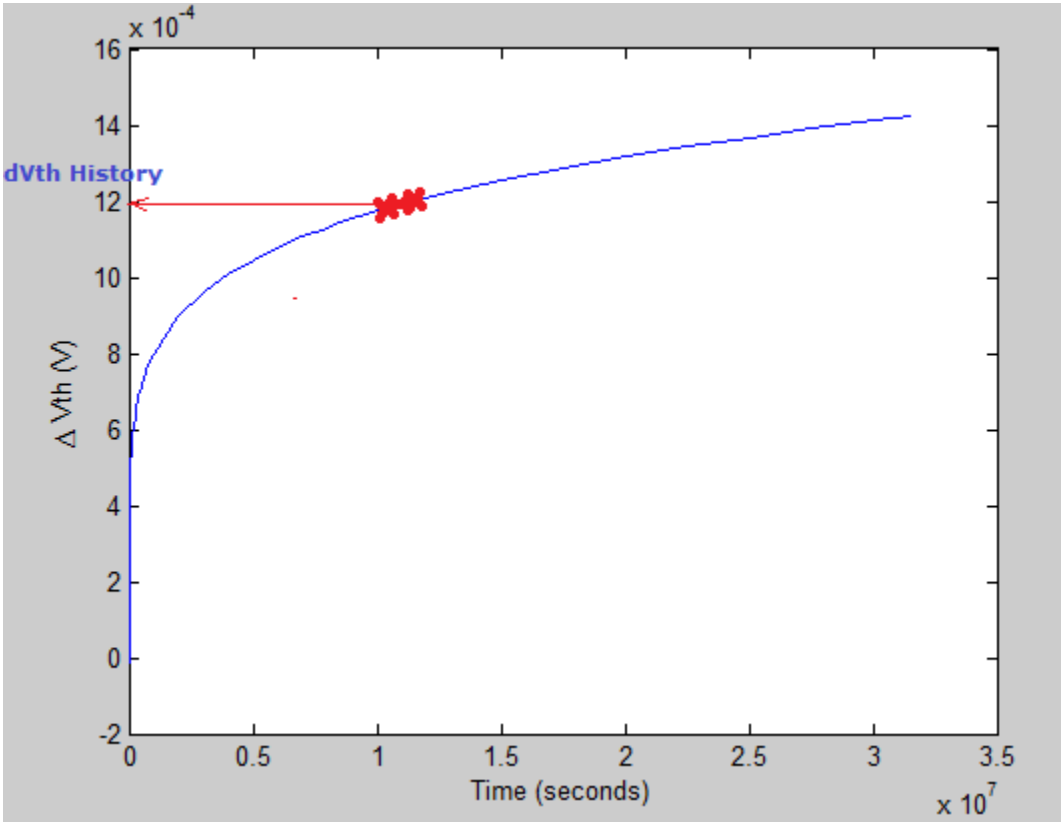


Figure 8-1. Matching $r_{60\text{-}120}$ with the 1Hz RD model degradation curve

Table 6 shows the ΔV_{th} history for different WCET benchmarks , running on the Leon3 in a DE2 FPGA for a period of 1 hour.

Table 6. ΔV_{th} history at t_{60} and t_{120}

WCET BENCHMARKS	ΔV_{th} history at t_{60} (mV)	ΔV_{th} history at t_{120} (mV)
adpcm	0.383	0.411
compress	0.394	0.414
bs	0.349	0.373

jfdctint	0.375	0.393
ns	0.386	0.401
nsichneu	0.440	0.449
statemate	0.441	0.423
ud	0.438	0.445
ndes	0.349	0.373
minver	0.380	0.406

This history can be used in the RD model equations to estimate/predict future from t_{60} or t_{120} to future. Similarly if we want to carry out NBTI degradation estimation/prediction from x years to future, we need to measure the actual ΔV_{th} degradation on the FPGA board at x years, which can be used as history.

8.2 NBTI degradation estimation/prediction for WCET benchmark suite

Here we carry out experiments for NBTI degradation estimation/prediction from 2 hours to future values, i.e. 1 year, 5 years and 10 years. For this we need the history, i.e. ΔV_{th} at 2 hours, to input into the RD model, which we get from Table 6 for various WCET benchmarks.

We first run the WCET benchmark suite on the Leon3 in a DE2 FPGA, for 1 minute each, and collect the statistics, i.e. mean and standard deviation, as discussed in Section 7.1.1. These statistics are used to generate a random waveform using random normal distribution for 1 second. We then implement the RD model using these randomly generated waveforms and collect 20 ΔV_{th} data points at regular intervals and perform curve fitting. The function, 2nd order logarithmic in our case, derived from curve fitting can be used to predict the future ΔV_{th} degradation by just inputting the time. Section 6.2 showed that running the Dhrystone benchmark and generating random normally distributed waveforms for 58ms, gave us an accuracy of 93% with the RD model.

Figure 8-2 shows ΔV_{th} degradation for LSB (bit 0) of the Leon3 register file running WCET benchmarks, from 2 hours to x years in future, where x = 1 year, 2 years and 10 years.

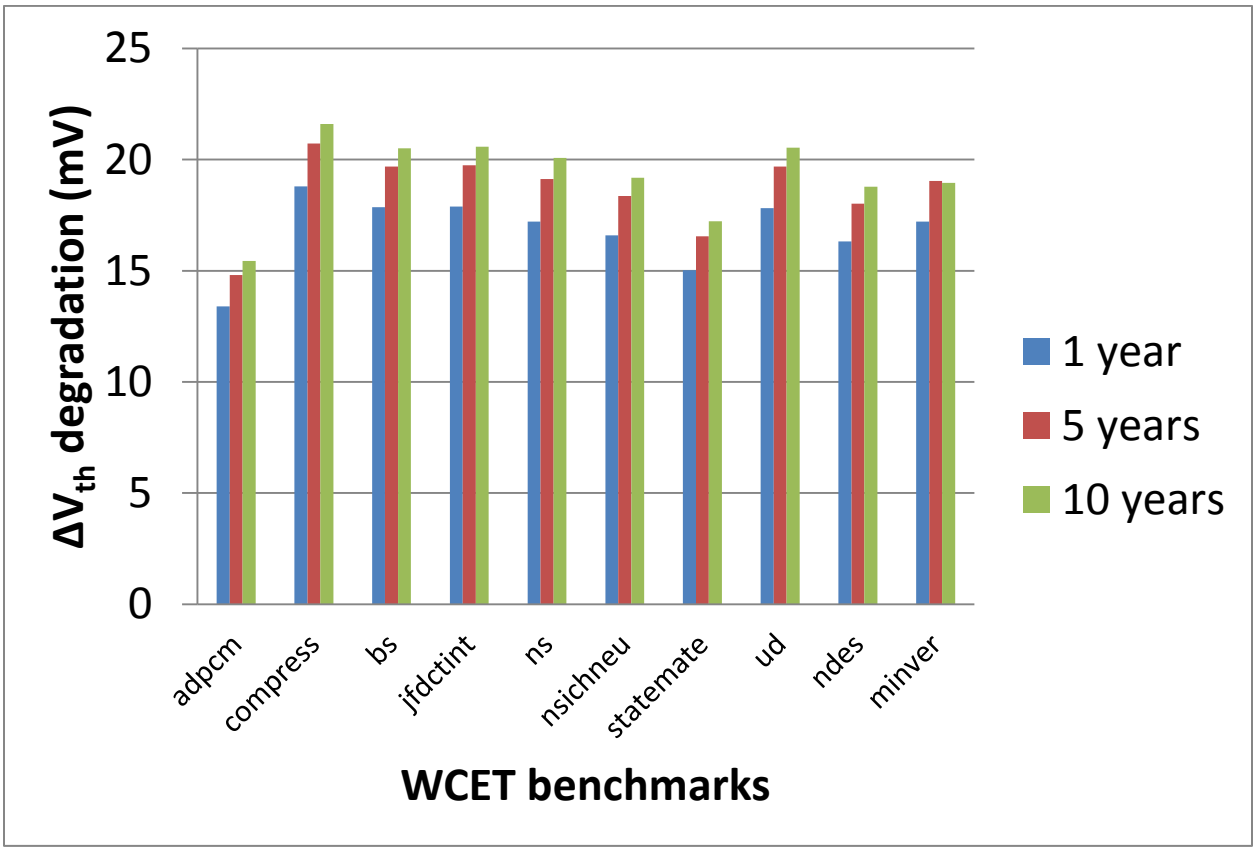


Figure 8-2. NBTI degradation for bit 0 of the Leon3 register file

From Figure 8-2. we can say that for the Leon3 register file bit 0, *adpcm* has the least NBTI degradation and c*ompress* has the highest NBTI degradation; i.e. activity of bit 0 for *adpcm* is the most and that of bit 0 for *compress* is the least. Similarly, Figure 8-3 shows ΔV_{th} degradation for MSB (bit 31) of the Leon3 register file running WCET benchmarks, from 2 hours to x years in future, where x = 1 year, 2 years and 10 years. For MSB *jfdctint* has the most NBTI degradation and again *adpcm* has the least.

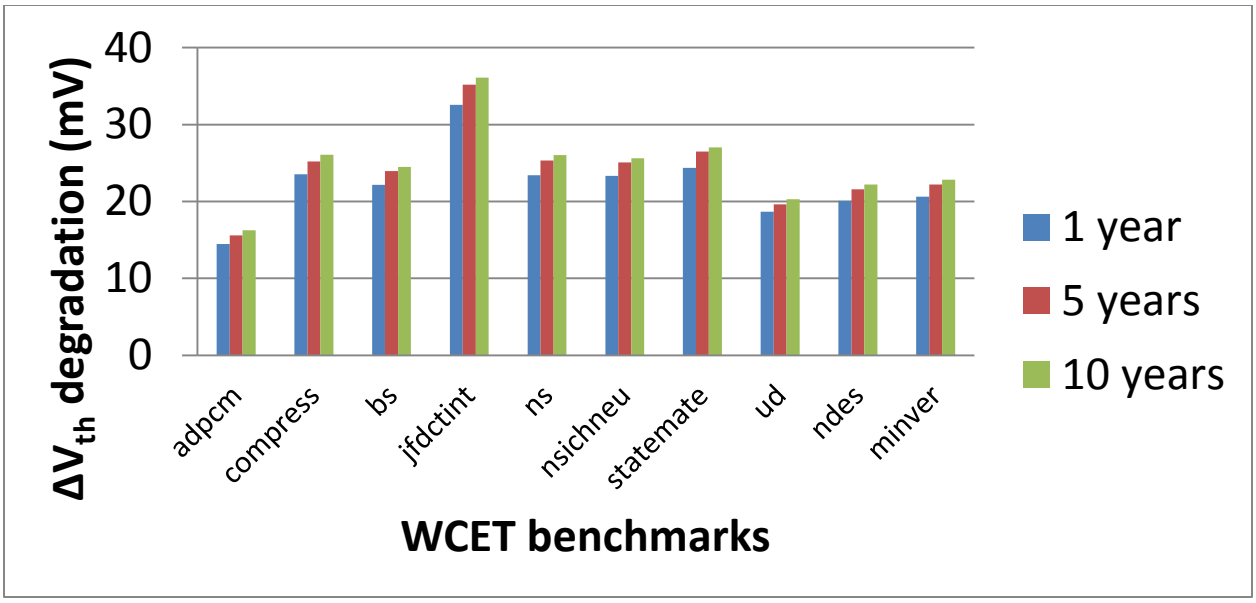


Figure 8-3. NBTI degradation for bit 31 of the Leon3 register file

Figure 8-4 shows the comparison between bit 0 and bit 31 NBTI degradation for a period of 10 years, running WCET benchmarks. It is clearly visible that the activity of MSB (bit 31) is less than that of LSB (bit 0), as the NBTI degradation of MSB is more than that of LSB.

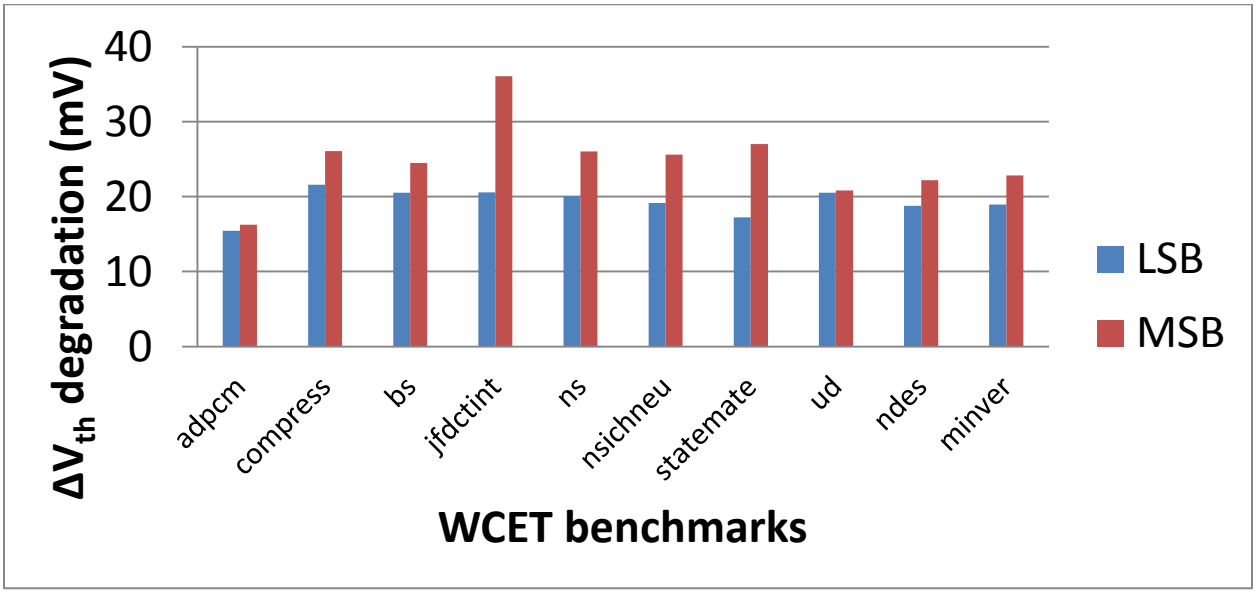


Figure 8-4. NBTI degradation LSB and MSB of theLeon3 register file for 10 years

From Figures 8-2 and 8-3, we can say that the WCET benchmarks result in an average of 19.39mV NBTI degradation for LSB and 27.75mV for MSB, over a period of 10 years. This is due to MSB having less activity than LSB.

8.3 Variation in ΔV_{th} across the FPGA

In Section 7.1.4 we talked about process variations across a chip and presented a technique to measure the ΔV_{th} across the four corners of the DE2 FPGA consisting of a single Leon3 core. It would be beneficial for the designer to know these process variations across the chip while designing any unit. Figure 8-5 shows the frequency degradation, obtained for 4 hours, of the 41-stage ring oscillators placed at the four corners of the FPGA. We convert these frequency degradation values into ΔV_{th} using equation 3. Then we calculate the rate of change of ΔV_{th} of the ring oscillators in each of the corners, and try to match it with the rate of the curve shown in Figure 8.6 (This curve is generated from the RD model with a square wave of 1Hz frequency) . The point t_0 at which this rate matches will be our $\Delta V_{th\text{-}initial}$.

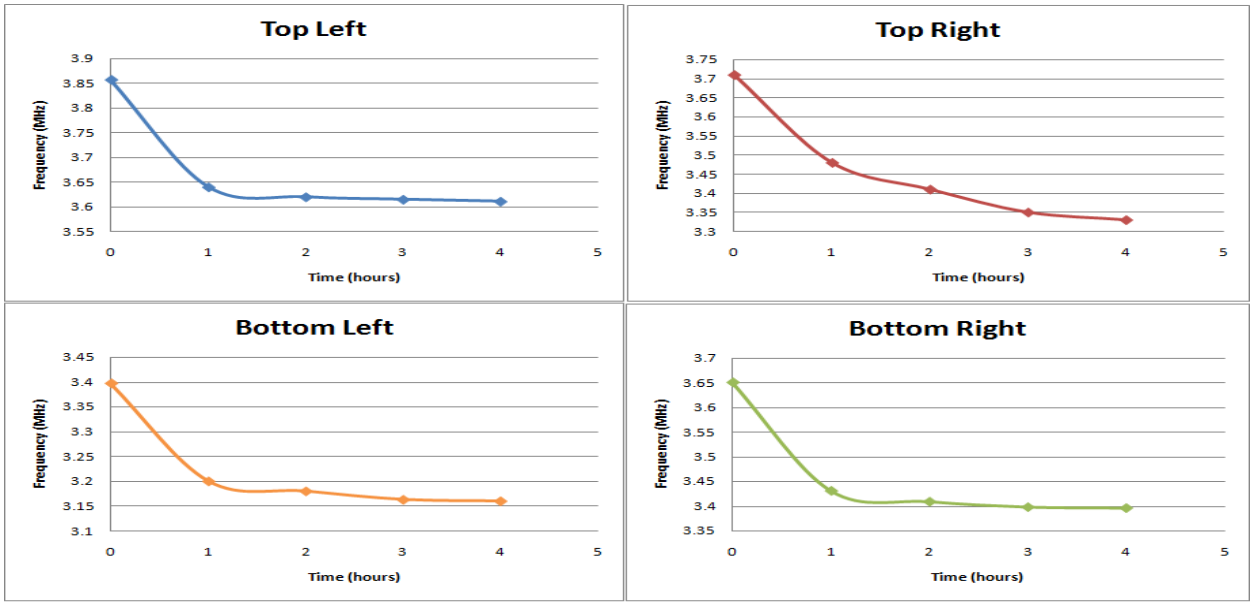


Figure 8-5. Frequency and ΔV_{th} degradation of the ROs placed in the 4 corners of DE2 FPGA

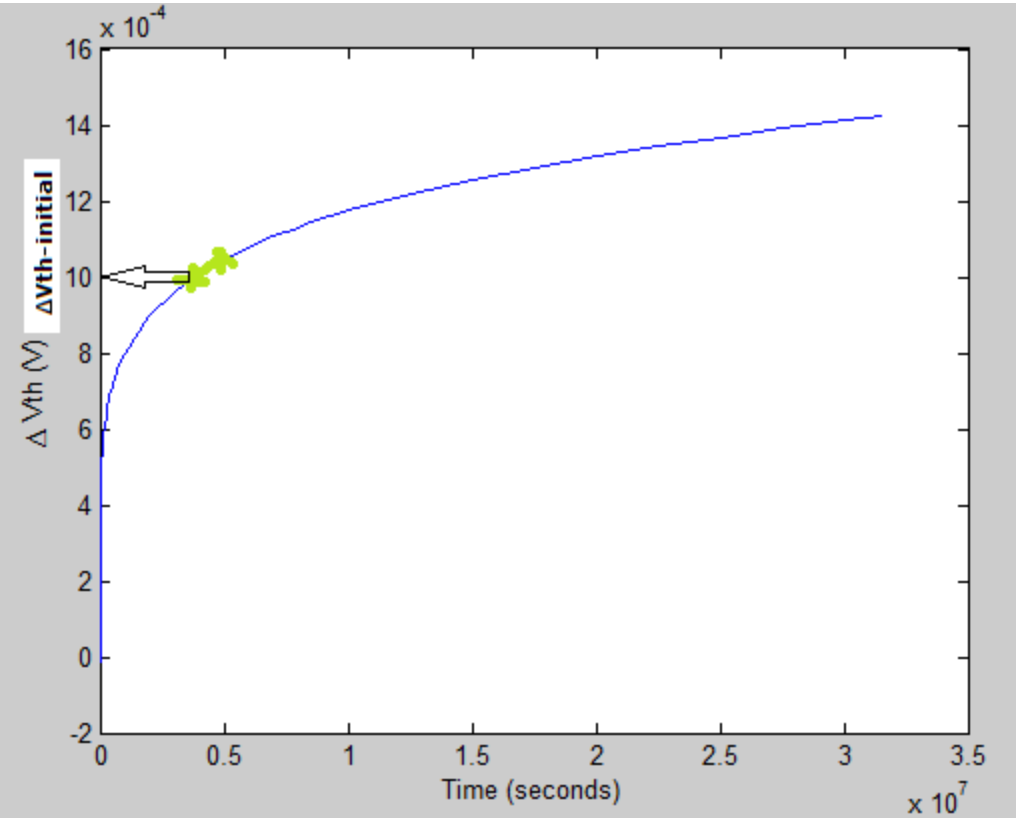


Figure 8-6. Technique to match the rate of the RO degradation with 1HZ RD model degradation curve

Table 7 shows the $\Delta V_{th\text{-}initial}$ values measured by the above technique for the four FPGA corners.

Table 7. $\Delta V_{th\text{-}initial}$ for 4 FPGA corners

FPGA Corner	$\Delta V_{th\text{-}initial}$ (mV)
Top Left	0.219
Top Right	0.24
Bottom Left	0.196
Bottom Right	0.165

From Table 7 we achieve a 0.08% to 0.11% variation in ΔV_{th}, from the base V_{th}, across the four corners of the FPGA.

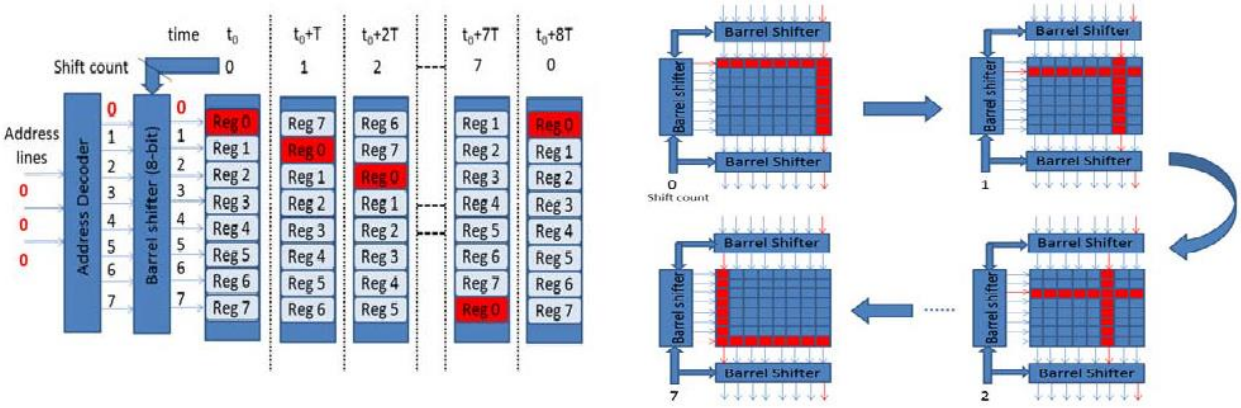
CHAPTER 9

FUTURE WORK

Here we presented a novel technique to predict future NBTI degradation which is faster by a factor of 10^2 than the RD, consumes almost 2000 times less memory, and also provides greater than 90% accuracy compared to the RD model. We also designed an online NBTI degradation predictor on the Leon3 in a DE2 FPGA and implemented these techniques to obtain future NBTI degradation for WCET benchmarks.

The question is, why do we need to carry out online NBTI prediction and how can this data be helpful? The answer is, we need this information to carry out some online/offline management which can increase the lifetime of the chip. There is a need for an online/offline model to adjust the parameters of the CMOS circuit to help it recover.

For a register file designed with RAM cells, bit flipping is one of the techniques which can be implemented to get at 50-50% degradation time of the PMOSs. But this technique results in large overhead. [42] proposes the technique of interleaving to reduce NBTI. Here register rotation is carried out to get a 50-50% degradation times. Zero bias probability (ZBP) is the amount of time a register file cell stores a 0. The degradation is the least when ZBP is 0.5, i.e. half amount of time the cell stores 0 and in the other half stores 1. A barrel shifter dynamically rotates the select line by *shift count.* This technique is shown in Figure 9-1(a). If Reg0 is mapped to row1, after time interval T it gets mapped to row2 and so on. We also need to rotate the columns which is done by Bit Level Rotation. The entire setup is as shown in Figure 9-1(b). This way the overall average ZBP over the entire register file will be 0.5, leading to minimum NBTI degradation.



(a) Shifting of Registers in SRAM stack (b) Shifting of Rows & Columns in SRAM stack

Figure 9-1. A technique to bring the average ZBP to 0.5 [42]

Another technique (for multi-cores) like task management can be implemented to reduce the effect of NBTI in a degraded core. For example, in a 4-core system, at some point of time core-1 is the most degraded and core-3 is the least. The task scheduled for core-1 can be transferred to core-3, so that core-1 can start recovering.

Various techniques, similar to above two, can be implemented which can lower the NBTI degradation of the register file and increase its lifetime.

CHAPTER 10

CONCLUSION

Technology scaling increases on-chip degradation making Online Wear-out Estimation a necessity. We proposed a novel Online NBTI Wear-out Estimation technique which tunes the RD model, provides 80% accuracy with respect to the RelXpert simulator and is 8 times faster than the RelXpert Simulator. The novel Waveform Compression Technique is used to save only the statistics of the Leon3 Register File activity waveforms, reducing 256MB memory cost to 131KB, with 91% accuracy with respect to storing the entire waveforms. The ΔV_{th} estimation/degradation technique is implemented to minimize the performance degradation of the system due to the novel online wear-out estimation technique, by predicting V_{th} degradation faster by an order of 10^2 with NBTI degradation within 93% of the tuned RD model. The above experiments were done for Dhrystone benchmarks. We implement the NBTI degradation predictor module on the Leon 3 in a DE2 FPGA to predict NBTI degradation of its register file, for WCET benchmarks. WCET benchmarks result in an average of 19.39mV NBTI degradation for LSB and 27.75mV for MSB, over a period of 10 years. We also measure the ΔV_{th} variation across the 4 corners of the DE2 FPGA running a single Leon3, which varies from 0.08% to 0.11% of the base V_{th}.

Carrying out some type of NBTI wear-out management using our designed NBTI degradation predictor remains as the future work.

REFERENCES

[1] Blome, J.; Feng, S.; Gupta, S.; Mahlke, S.; , "Self-calibrating Online Wearout Detection," *40th Annual IEEE/ACM International Symposium on Microarchitecture*, Dec. 2007

[2] Kim, T.; Persaud, R.; Kim, C.; , "Silicon Odometer: An On-Chip Reliability Monitor for Measuring Frequency Degradation of Digital Circuits," *2007 IEEE Symposium on VLSI Circuits,* Apr. 2008

[3] Martins, C.; Semiao, J.; Vazquez, J.; Champac, V.; Santos, M.; Teixeira, I.; Teixeira, J.; , "Adaptive Error-Prediction Flip-flop for performance failure prediction with aging sensors," *VLSI Test Symposium (VTS),* May 2011

[4] Vazquez, J.; Champac, V.; Teixeira, I.; Santos, M.; Teixeira, J.; , "Programmable aging sensor for automotive safety-critical applications," *Design, Automation & Test in Europe Conference & Exhibition,* March 2010

[5] Agarwal, M.; Paul, B.; Zhang, M.; Mitra, S.; , "Circuit Failure Prediction and Its Application to Transistor Aging," *VLSI Test Symposium,* May 2007

[6] Shah, N.; Samanta, R.; Zhang, M.; Hu, J.; Walker, D.; , "Built-In Proactive Tuning System for Circuit Aging Resilience," *IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems. DFTVS,* Oct. 2008

[7] Cabe, A.; Qi, Z.; Wooters, S.; Blalock, T.; Stan, M.; , "Small embeddable NBTI sensors (SENS) for tracking on-chip performance decay," *Quality of Electronic Design, ISQED,* March 2009

[8] Wang, W.; Reddy, V.; Krishnan, A.T.; Vattikonda, R.; Krishnan, S.; Cao, Y.; , "Compact Modeling and Simulation of Circuit Reliability for 65-nm CMOS Technology," *IEEE Transactions on Device and Materials Reliability,* Dec. 2007

[9] Kang, K.; Kim, K.; Islam, A.; Alam, M.; Roy, K.; , "Characterization and Estimation of Circuit Reliability Degradation under NBTI using On-Line IDDQ Measurement," *Design Automation Conference,* June 2007

[10] Saluja, K.; Vijayakumar, S.; Sootkaneung, W.; Yang, X.; "NBTI Degradation: A Problem or a Scare?," *International Conference on VLSI Design,* Jan. 2008

[11] Peters, L.; , "NBTI: A Growing Threat to Device Reliability", *Semiconductor International, 2004.*

[12] Bernstein, K.; Frank, D.; Gattiker, A.; Haensch, W.; Ji, B.; Nassif, S.; Nowak, E.; Pearson, D.; Rohrer, N.; , "High-performance CMOS Variability in the 65 -nm Regime and Beyond",*IBM J . Res. & Dev.,* July 2006.

[13] Siddiqua, T.; Gurumurthi, S., "Enhancing NBTI Recovery in SRAM Arrays Through Recovery Boosting," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* April 2012

[14] Agarwal, M.; Balakrishnan, V.; Bhuyan, A.; Kim, K; Paul, B.C.; Wenping Wang; Yang, B.; Yu Cao; Mitra, S, "Optimized Circuit Failure Prediction for Aging: Practicality and Promise," *IEEE International Test Conference*, Oct. 2008

[15] Siddiqua, T.; Gurumurthi, S.; Stan, M.R., "Modeling and analyzing NBTI in the presence of Process Variation," *International Symposium on Quality Electronic Design,* March 2011

[16] Kumar, S.; “Reliability-aware and Variation-aware CAD techniques,” 2009. University of Minnesota. *purl.umn.edu/57462*

[17] Zheng, R; Velamala, J.; Reddy, V.; Balakrishnan, V.; Mintarno, E; Mitra, S; Krishnan, S.; Cao, Y.; , "Circuit aging prediction for low-power operation," *Custom Integrated Circuits Conference,* Sept. 2009

[18] Wang, W.; Reddy, V.; Yang, B.; Balakrishnan, V.; Krishnan, S.; Cao, Y., "Statistical prediction of circuit aging under process variations," *Custom Integrated Circuits Conference,* Sept. 2008

[19] Tiwari, A.; Torrellas, J., "Facelift: Hiding and slowing down aging in multicores," *IEEE/ACM International Symposium on Microarchitecture,* Nov. 2008

[20] Wang, W.; Yang, S.; Bhardwaj, S.; Vrudhula, S.; Liu, T.; Cao, Y., "The Impact of NBTI Effect on Combinational Circuit: Modeling, Simulation, and Analysis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* Feb. 2010

[21] Calimera, A.; Macii, E.; Poncino, M., "Analysis of NBTI-induced SNM degradation in power-gated SRAM cells," *IEEE International Symposium on Circuits and Systems (ISCAS)*, May-June 2010

[22] Ultrasim - RelXpert Reliability Simulator www.celestry.com/pdf/relxpert.pdf

[23] BSIMPro+ models www.celestry.com/pdf/celestry.pro.pdf

[24] DE2 FPGA www.altera.com/education/univ/materials/boards/de2/unv-de2-board.html

[25] Altera-Modelsim

http://www.altera.com/products/software/quartus-ii/modelsim/qts-modelsim-index.html

[26] Mathworks - Matlab www.mathworks.com/products/matlab/

[27] PTM Reliability Model www.ptm.asu.edu

[28] Seyab; Hamdioui, S., "NBTI modeling in the framework of temperature variation," *Design, Automation & Test in Europe Conference & Exhibition (DATE),* March 2010

[29] Khan, S.; Hamdioui, S., "ReverseAge: An online NBTI combating technique using time borrowing," *IEEE 6th International Design and Test Workshop (IDT),* Dec. 2011

[30] SPARC V8 Architecture http://www.sparc.org/standards/V8.pdf

[31] GRLIB IP Library http://gaisler.com/products/grlib/grlib.pdf

[32] Slimani, M.; Matherat, P., "Multiple threshold voltage for glitch power reduction," *Faible Tension Faible Consommation (FTFC), May-June* 2011

[33] Borkar, S.; Karnik, T.; Narendra, S.; Tschanz, J.; Keshavarzi, A.; De, V., "Parameter variations and impact on circuits and microarchitecture," *Design Automation Conference,* June 2003

[34] Sedcole, P.; Cheung, P. , "Within-die delay variability in 90nm FPGAs and beyond," *IEEE International Conference on Field Programmable Technology,* Dec. 2006

[35] Altera Quartus II
http://www.altera.com/products/software/quartus-ii/about/qts-performance-productivity.html

[36] Cygwin http://www.cygwin.com/

[37] Aeroflex Gaisler GRMON2 debugger http://www.gaisler.com/doc/grmon2.pdf

[38] Altera DE2 Development and Education Board
ftp://ftp.altera.com/up/pub/Altera_Material/12.1/Boards/DE2/DE2_User_Manual.pdf

[39] WCET Benchmark Suite http://www.mrtc.mdh.se/projects/wcet/benchmarks.html

[40] Bare-C Cross-Compiler (BCC) System for Leon3 gcc 3.4.4

http://www.gaisler.com/index.php/products/operating-systems/bcc

[41] Bild, D.; Bok, G.; Dick, R., "Minimization of NBTI performance degradation using internal node control," *Design, Automation & Test in Europe Conference & Exhibition,* April 2009

[42] Kothawade, S.; Chakraborty, K.; Roy, S.; , "Analysis and mitigation of NBTI aging in register file: An end-to-end approach," *12th International Symposium on Quality Electronic Design (ISQED)*, Mar. 2011