SINGLE EVENT TRANSIENT MODELING AND MITIGATION TECHNIQUES

FOR MIXED-SIGNAL DELAY LOCKED LOOP (DLL) AND CLOCK CIRCUITS

By

Pierre Maillard

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

May, 2014

Nashville, Tennessee

Approved by:

Dr. Lloyd W. Massengill

Dr. W. Timothy Holman

Dr. Ronald D. Schrimpf

Dr. T. Daniel Loveless

Dr. Stephen Buchner

ACKNOWLEDGMENTS

ABSTRACT

The purpose of this PhD work has been to investigate, model, test, develop and provide hardening techniques and guidelines for the mitigation of single event transients (SETs) in analog mixed-signal (AMS) delay locked loops (DLLs) for radiation-hardened applications.

Delay-locked-loops (DLLs) are circuit substructures that are present in complex ASIC and system-on-a-chip designs. These circuits are widely used in on-chip clock distribution systems to reduce clock skew, to reduce jitter noise, and to recover clock signals at regional points within a global clock distribution system. DLLs are critical to the performance of many clock distribution systems, and in turn, the overall performance of the associated integrated system; as such, complex systems often employ multiple DLLs for clock deskew and distribution tasks. In radiation environments such as on-orbit, these critical circuits represent at-risk points of malfunction for large sections of integrated circuits due to vulnerabilities to radiation-generated transients (i.e. single event transients) that fan out across the system.

The analysis of single event effects in analog DLLs has shown that each DLL sub-circuit primitive is vulnerable to single event transients. However, we have identified the voltage controlled delay line (VCDL) sub-circuit as the most sensitive to radiation-induced single event effects generating missing clock pulses that increase with the operating frequency of the circuit. This vulnerability increases with multiple instantiation of DLLs as clock distribution nodes throughout an integrated system on a chip. To our knowledge, no complete work in the rad-hard community regarding the hardening of mixed-signal DLLs against single event effects (missing pulses) has been developed. Most of the work present in the literature applies the "brute force" and well-established digital technique of triple modular redundancy (TMR) to the digital subcomponents.

We have developed two novel design techniques for the mitigation of DLL missing pulses that are fully implementable in modern CMOS technologies. These techniques offer to the community the choice of hardening using a restoring current technique in the VCDL sub-circuit to inhibit the creation of missing pulse errors, or using a combinational logic error monitoring technique to correct missing pulses after they occur in real time. We have implemented both of these techniques with minimal area and power penalties when compared to TMR. In addition, these hardening techniques have been extrapolated to other clock circuits, such as digital PLLs.

The first hardening technique uses a hardened complementary differential pair VCDL to increase the critical charge (Q_{crit}) necessary for single event transient generation and thus mitigate missing pulses at the source. Our implementation of this technique at 180 nm, 90 nm and 40 nm required less than a 2% area penalty over a non-hardened design. To experimentally validate this technique, hardened VCDLs were designed and fabricated in 180-nm IBM and 40-nm UMC technologies, then tested at the Naval Research Lab in Washington D.C. The second hardening technique, based on combinational logic pulse monitoring, uses an error correction circuit to mitigate the missing pulses as they occur. This ECC technique is implemented via a "peeled" VCDL (i.e. each transistor is split in area but doubled in multiplicity). We have shown the effectiveness of this technique by implementing it in a Xilinx Virtex 5 FPGA. Furthermore, this new ECC technique is independent of technology scaling – a highly valuable attribute for sub-50 nm design applications.

In addition to the formulation, simulation, prototyping, fabrication, and testing of these new hardening solutions, we developed a unique single event analytical model to guide future hardened DLL designs at advanced technology nodes. The model was furthermore generalized to PLL and DLLs. These analytical models were then used to provide a set of equations to the designer for important insight into hardening choices and tradeoffs based on design

specifications, in conjunction with a broad set of guidelines for the design of hardened DLLs regarding circuit topology choices and parameter sensitivity on radiation exposure.

We are confident that these results, tools, and guidelines will significantly expand the state-of-the-art in the design of hardened DLL clocking circuits for rad-hard applications.

TABLE OF CONTENTS

LIST OF FIGURES

x

LIST OF TABLES

CHAPTER I

INTRODUCTION

In electrical engineering, a delay-locked loop (DLL) is an integrated circuit (IC) similar in topology to a phase-locked loop (PLL), with the main difference being that the voltage controlled oscillator is replaced by a voltage controlled delay line [3-8, 18, 21, 25]. On practical or commercial chips, the RC delay of clock distribution networks is very long, because of the wire resistance and gate load. Variations in this delay cause clock signals to reach different elements of the IC at different times, this undesirable effect is called clock skew. Repeaters are often used in commercial ICs to buffer the clock and further equalize the delay. These circuits reduce skew but do not eliminate it. To control both skew and clock jitter, delay-locked loops are commonly used in high-speed data acquisition systems in space-deployed electronics. In addition, DLL are typically used to enhance the output clock timing characteristics of integrated circuits by changing the phase of a clock signal, such as DDR-SRAM or DRAM devices or for clock-recovery circuits [3-8, 20, 21, 24, 36, 46].

The main component of a DLL is the voltage-controlled delay line (VCDL), constituting of several delay cells connected in series. The input of the delay chain and by extension of the DLL is connected to the input reference clock that will be delayed and the resulting delay clock signal is the output of the DLL. In other words, a DLL uses a variable phase, equivalent to a delay block, whereas a PLL uses a variable frequency block. A DLL compares the phase between the circuit's input reference clock and the output of the circuit to generate a digital error signal that is then integrated and used to control all of the delay cells in the voltage controlled delay line. Because the control signal directly affects the output phase of the DLL, the integration allows

generated error to be reduced to zero while keeping the control signal, and output delays at the required values to allow the DLL to acquire phase lock.

A PLL compares the phase of an incoming reference clock signal to its voltage controlled oscillator (VCO) to create an error signal that is then integrated to generate a control voltage for the VCO. Since the control voltage affects the output frequency of the oscillator, a second integration by the oscillator itself is required because the phase is the integral of frequency. Therefore, the DLL is a loop of 1st order and the PLL is a loop of 2nd order or higher. In DLLs, the number of delay cells elements in the controlled delay chain must be even otherwise the duty cycle of the clock at the intermediate nodes of the chain might become irregular. Compared to phase-locked loops, delay-locked loops are a fairly recent invention, first presented by Dr. M. Combes in [84], then commercialized and popularized by Xilinx in their Virtex FPGAs.

The DLLs or PLLs operating in hostile environments (such as in space) are continuously exposed to radiation particles that can modify the circuit behavior. Single-event transients (SETs) have been identified as the primary failure mechanism behind several spacecraft malfunctions in recent years. Several techniques have been developed to characterize SETs in integrated circuits. Experimental approaches based on heavy-ion beams, laser strikes, and ion micro-beams have been used to characterize the effects of SETs in integrated circuits. The single-event vulnerability of DLL and PLL circuits are of particular concern for space-deployed systems, as SETs occurring within a clock distribution system can result in global errors across the entire IC. Since DLLs typically are preferred over their PLL counterparts when frequency synthesis is not required, the DLL is an excellent candidate for use in spacecraft clock distribution networks. While recent publications have shown that ion strikes in PLL circuits can result in SETs on the order of several microseconds in length [12-14, 15], no research to date has specifically characterized SETs in, or developed hardening techniques for, analog DLLs. Therefore, it is critical that radiation-

hardened-by-design (RHBD) DLLs, resilient to errors due to radiation effects, be developed for space or high irradiative environment applications.

A.   Objective of Research

This dissertation proposes the use of circuit simulations and radiation experiments to develop radiation-hardened-by-design (RHBD) techniques to mitigate single event transients in mixed-signal DLLs, drawn upon previous single event effect (SEE) characterization of unhardened analog DLLs. RHBD DLLs, resilient to single event transients but with reduced power and area penalty when compared to existing hardening techniques, will be developed to validate experimentally the proposed hardening techniques. The proposed hardening techniques will be portable to a wide variety of clock circuit topologies, such as digital PLLs. Tradeoff studies of the performance and radiation response of the developed hardening techniques (implemented in RHBD DLLs), obtained using ISDE bias-dependent modeling [17] and two-photon absorption (TPA) laser testing [11, 22], has also been performed for different technology nodes. Furthermore, a single event analytical model for DLLs, to guide future hardened DLL designs at advanced technology nodes has been provided. These analytical models, in addition to the new hardening techniques developed for DLL, will be the base of a set of broad guidelines for the design of hardened DLLs regarding circuit topology choices and parameter sensitivity to radiation exposure levels.

B.   Organization of the dissertation

The research effort proposed in this dissertation is organized as follows:

- Chapter I introduces the motivation for this work.

- Chapter II provides a background on delay locked loops.

- Chapter III provides a detailed discussion of the analog DLL, considered in this work.

- Chapter IV offers a background on radiation effects, specifically single events

- Chapter V offers a background of the SEE work on DLLs and PLLs, especially single event effects characterization and mitigation

- Chapter VI RHBD techniques to mitigate single event transient in DLL's sub-circuits, and focus mainly on missing pulses in the voltage controlled delay line.

  1. Characterize the SET response of the mixed-signal analog DLL (MSEE 2010).

  2. Develop or provide RHBD techniques to mitigate missing pulses in DLLs, including techniques to minimize or eliminate inverted lock error or duty-cycle errors induced by highly energized ionized particles.

- Chapter VII provide a new and unique SET analytical model for DLLs, and then combined also DLLs/PLLs, for the first time, under a unique set of equations and finally guidelines for rad-hard DLL and clock circuits design.

CHAPTER II

THE DELAY LOCKED LOOP

A.   Introduction

The DLL is widely used for clock generation, clock de-skewing, and data recovery. When compared to PLLs, DLLs are a relatively recent innovation, first described by Dr. M. Combes in the early 1990s, then commercialized in the Virtex FPGA products family by Xilinx [84]. PLLs and DLLs have been typically employed for the purpose of clock signal synchronization. Due to differences in their configurations, the DLLs are preferred over PLLs for their absolute stability and faster locking time. Furthermore, because noise does not accumulate over several clock cycles in the VCDL, DLLs have better jitter performance than PLLs [18, 19].

B.   Conventional delay-locked loop vs. digital phase locked loops

This work focuses, on AMS analog DLLs, conventionally used in applications where small, accurate, and precise delay is required. [18]. In the literature, PLLs and delay-locked loops DLLs have been extensively adopted for clock recovery or eliminate clock signal skews and jitter in high-speed communication integrated circuits [3-8, 18-21, 24, 25, 33-37, 52, 54]. This sub-section compares both DLL and PLL topologies.

In Figure II-1 (a), the DLL consists of four main sub-circuits: a phase detector (PD), a charge pump (CP), a low-pass filter (LPF), and a voltage-controlled delay line (VCDL). From a topological perspective, DLLs and PLLs are similar in many respects. As shown in Figure II-1 (b), the digital PLL consists of a phase-frequency detector (PFD), a CP, a LPF, and a voltage controlled oscillator (VCO). The main difference is that a PLL uses the VCO to generate an

oscillating signal that is synchronized with the reference signal, while the DLL adjusts the phase of a reference signal using the VCDL. Therefore, PLLs are often used as global clock generation circuit in clock tree networks.

One of the main differences between PLLs and DLLs is that DLLs are typically single-pole systems with a first-order loop filter (LPF pole), while PLLs are, at minimum, two-pole (VCO and LPF poles) systems. Consequently, DLLs provide improved stability over PLLs. Furthermore, whereas PLLs offer a greater phase capture range than the DLL ($\pm 2\pi$ versus $\pm\pi$ radians), DLLs have no phase error accumulation and no self-generated jitter, both of which are present in PLLs.

Correct timing is critical for the proper operation of high-performance digital- and mixed-signal circuits. As the size and operating frequency of VLSI systems increase, designing appropriate clock-distribution systems poses numerous challenges. Many factors, such as the operating frequency, duty-cycle, phase, jitter, and clock skew define the attribute of clock pulses. Therefore, a design that would minimize all the negative effects mentioned earlier is a good solution. Traditionally, PLLs are used for high-frequency clock synthesis. When frequency multiplication is not required, DLLs offer better performance than do PLLs, since their design is less complex they are immune to on-chip noise and their stability is better and because a first-order DLL is more stable than a higher-order loop filter PLL. Furthermore, because of the accumulation of jitter, PLLs are more susceptible to substrate and power-supply substrate noise than are DLLs [6].
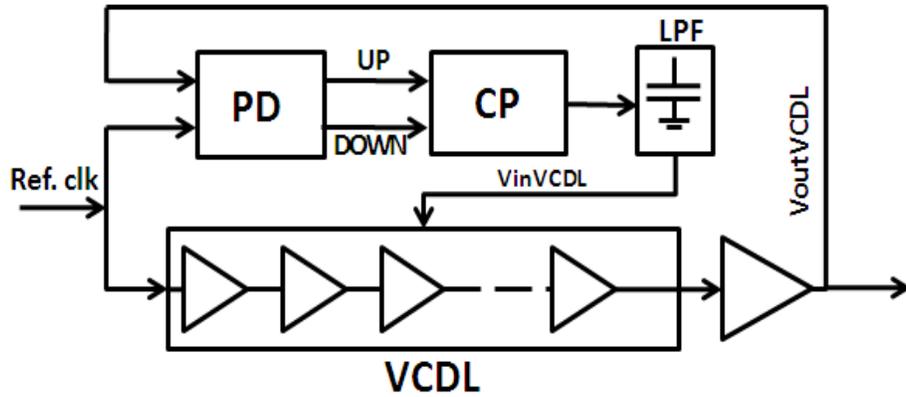
(a)
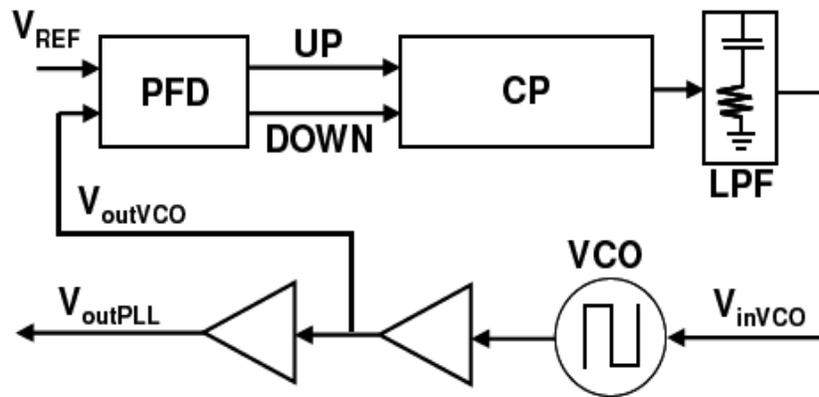
(b)

Fig. II-1 a). Block diagram of a DLL consisting of a phase detector (PD), charge-pump (CP), low-pass filter (LPF), and voltage-controlled delay line (VCDL). b). Block Diagram of a PLL consisting of a phase-frequency detector (PFD), charge-pump (CP), low-pass filter (LPF), and voltage-controlled oscillator (VCO) [1].

C. Delay Locked Loop Topologies

In the literature, a wide variety of delay-locked loop designs are proposed to mitigate skew and jitter in microprocessors, Ethernet transceivers or communication integrated circuits. Designs, such as wide-range DLLs, all-analog multiphase DLLs, clock multiplier DLLs, all digital with clock divider DLLs, etc. were developed [3-8, 18-21, 32-39, 50, 51, 53]. However, according to the principle of phase shift generation, DLL architectures can be classified into three

7

classes: analog mixed-signal also known as analog [5, 10, 20-21,36, 56], digital [26, 27,36, 37,41], and dual loop [7, 9, 57].

1.   The Analog Delay Locked Loop

The conventional DLL design is the analog DLL [6, 18, 21], illustrated in Figure II-2. However, phase detectors constituting of digital components are generally used in both digital and analog DLLs and it is therefore more accurate to refer to the conventional analog DLL as an analog mixed-signal DLL, but, for simplicity, they are referred to as analog DLL. The analog DLL is composed of four sub-circuits: a voltage- controlled delay line, a phase detector, a charge pump, and a first order loop filter. The VCDL is composed of several variable delay elements connected in series. The reference clock (Ref.clk) drives the input of the VCDL. In order to determine the phase alignment error, the PD compares the rising edges of *Ref.clk* and *DLL$_{out}$*. The CP and LPF are combined to act as an integrator, and generate the VCDL's control voltage *VinVCDL*. Under proper operating conditions, the DLL forces a phase difference between the circuit's input clock *Ref.clk* and the output of the DLL, *DLL$_{out}$*, to align them. In proper lock state, the total delay of VCDL should be equal to one period of the input reference clock, *Ref.clk*. Analog. DLLs are appropriate for fine-grain delay variation.

Analog DLLs can be designed to exhibit less jitter than digital DLLs because they use a continuously variable delay line and thus do not suffer from quantization error observed in digital DLLs, due to the discrete delay steps. Analog DLLs are more process dependent and therefore less portable than digital DLLs, but can be designed to consume less silicon area and use less power than digital DLLs [6, 18, 85]. Thus, analog DLLs are effective in applications requiring small, accurate, and precise delays.
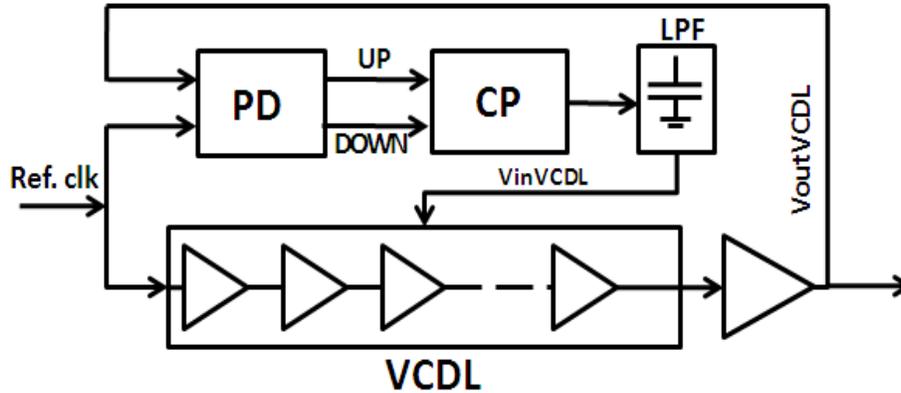
Fig. II-2 Schematic of the analog mixed-signal DLL consisting of: Voltage Controlled Delay Line (VCDL), Phase Detector (PD), Charge Pump (CP), and first order Loop Filter (LPF) [1]

Conventional DLLs may suffer from harmonic locking or false locking over a wide operating frequency range as illustrated in Figure II-3. If the initial state of the DLL is such that the error at the phase detector is greater than 2π radians, the DLL will be forced to add delay. The VCDL gain will be increased beyond the usual operating range of the DLL and final delay in the VCDL will be larger than one clock period. This is referred to as a harmonic lock [62, 85]. If the DLL initial phase error is below -2π radians, the DLL will try to remove delay because the PD will report a positive phase error, and since the gain of the VCDL tend to zero when approaching minimum delay, it is probable that the VCDL will saturate and the DLL will never lock. This is referred to as a false locking [62, 85]. To avoid false or harmonic lock, conventional mixed-signal DLLs are typically designed to operate in a limited range of $\pm\pi$.
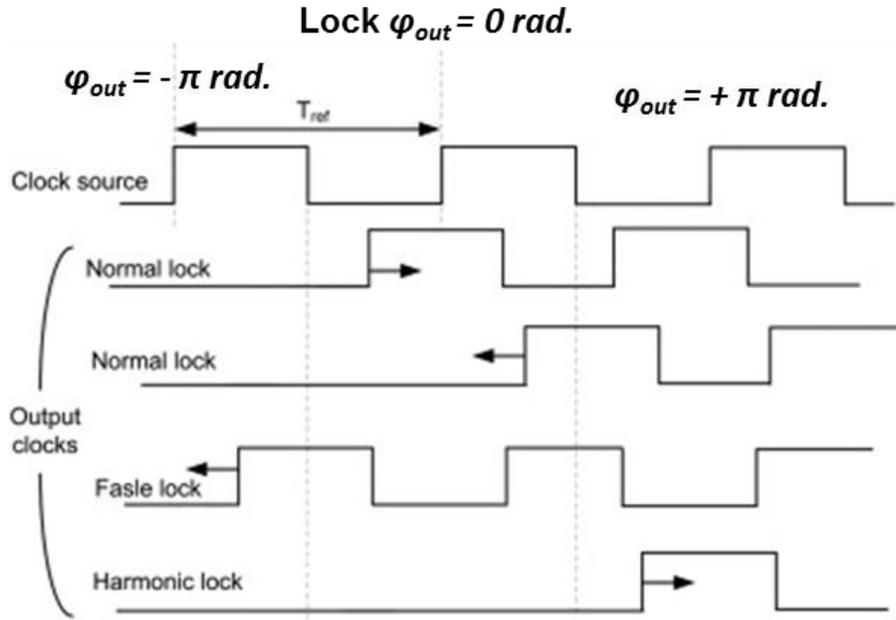
Fig. II-3 Illustration of harmonic and false lock in DLLs.

To solve the false locking problem, the DLL must operate within limited operating range. To increase the width of the operating range, a family of wide-range DLL architectures has been developed [3, 25, 35, 36]. Also, DLLs with dual-loop architectures [32, 85] have been proposed to solve the problem of a limited operating range, using more than one VCDL. In [3], an all-analog DLL uses the replica delay line to solve the narrow operating frequency-range problem of a conventional DLL. To detect and prevent harmonic lock, DLLs using auxiliary VCDL to inspect the main VCDL has been developed [62]. DLLs creating multiple tap point from the VCDL and then using a phase selection scheme to extend the operating range of the DLL have also been developed [34]. These designs improvements usually come at the price of increased complexity, and if phase inversion is applied, the duty cycle of the input clock is no longer sustained and the AMS DLLs become more affected by process variations.

2.   The All-digital Delay Locked Loop

The digital DLL, shown in Figure II-4, consists of a digitally controlled delay line (DCDL), phase detector (PD), phase selector (PS), and finite state machine (FSM). The DCDL is implemented as a chain of adjustable length delay cells. The amount of the delay is determined by the length of the DCDL chain. The PS is typically a multiplexer and at its output a pulse of defined shift delay is selected [6, 18, 85]. The amount of delay is expressed by the FSM's output signal. A quantized and fixed time delays is provided by the DCDL delay cells and is used for coarse-grain delay variation over a broad range of operation, meaning that the clock signal of the digital DLL is quantized into several coarse-grain discrete delay steps.

A digital DLL does not experience the same range restrictions as does a conventional analog DLL, and false locking can be easily prevented. Digital DLLs have the advantage of easier processes portability and absolute stability of a zero-order transfer function. The design advantages come at the price of increased jitter due to the quantization of the discrete delay steps, higher silicon layout area and increased power consumption, when compared to the conventional analog DLL.
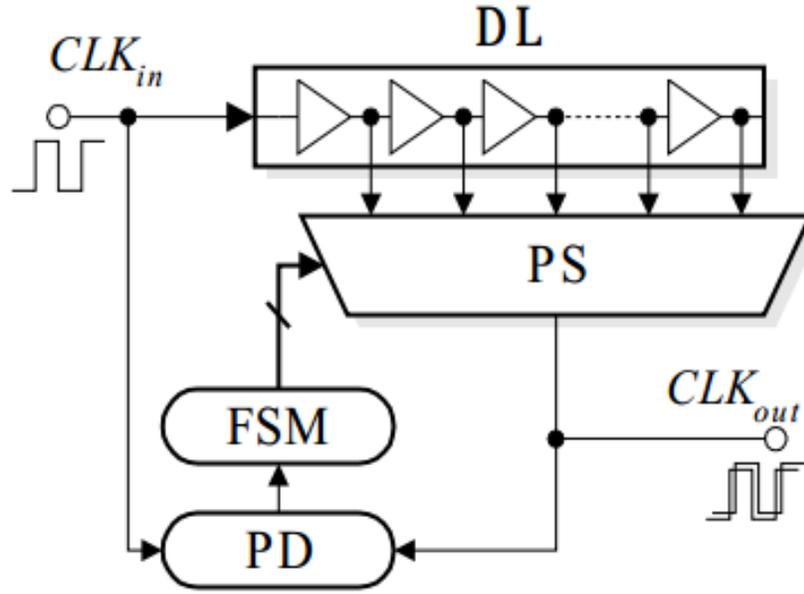
Fig. II-4 Schematic of the all-digital DLL consisting of Digitally Controlled Delay Line (DL), Phase Selector (PS), Phase Detector (PD) and Finite State Machine (FSM) [6].

Like the analog DLL, a digital DLL is also inclined to false lock. If the DLL resets with the minimum delay for the DCDL and the phase detector wants to remove delay, the DLL will never lock. A harmonic lock would happen because the phase detector wants to add delay , if the reset condition of the DCDL was set for maximum delay. In the literature, there are several digital DLL designs developed to avoid false lock. One solution that does not require complex initialization circuitry is for the center of the delay line to be reset and ensures that if there is a lock condition, the DLL will most likely locate it [85]. In [67], an all-digital multiphase clock generator is developed and proposed to mitigate the false locking condition in digital DLLs. In [34], a phase-selector circuit and a circuit using a start-control function and a phase selector is proposed to eliminate false locking problems. The drawback of the developed digital DLLs is an increase in design complexity translating in an increase in layout area, power consumption and jitter.

3. The Dual-Loop Delay Locked Loop

Only rarely reported on in the literature, the dual-loop DLL, illustrated in Figure II-5, is composed of a series of digital and analog DLLs [5, 6, 32]. Typically, the dual-loop DLL offers a wide operating delay range, but the jitter performance of the circuit is not practical for many design applications because the clock propagates through two large controlled delay lines and loops, making this dual-loop DLL not competitive in terms of jitter when compared to the conventional digital PLL. In addition, dual-loop DLLs are not suitable for space environment application because their architecture complexity and power consumption are high.
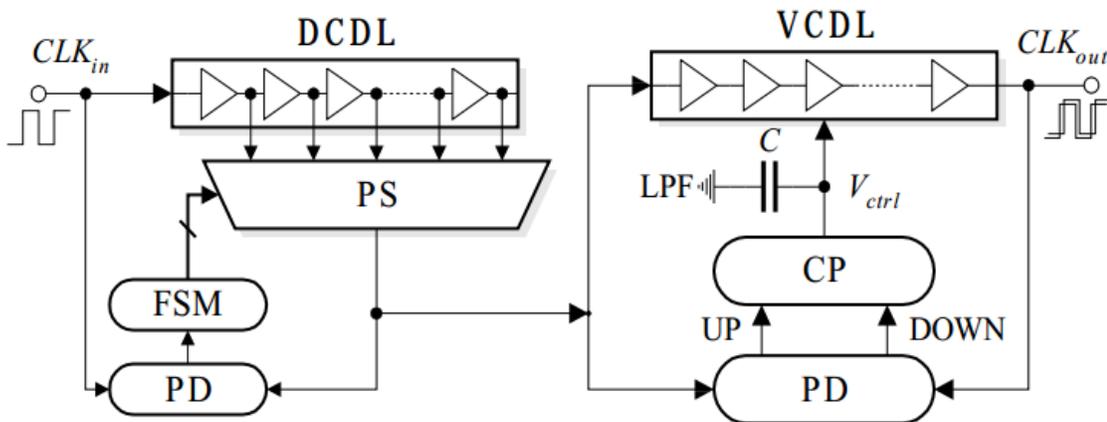
Fig. II-5 Schematic of the dual-loop DLL consisting of a series of the all-digital DLL and the analog DLL [6]

D. Delay locked loop performances comparison

Table II.1 summarizes typical DLL parameters such as operating frequency, locking time, jitter etc. The detailed parameters for different technology node and DLL architectures can be obtained from [3-8, 21, 23-25, 32-39, 46, 52-56, 62-64, 67, 71]. The comparison shows that the analog DLL can achieve a better tradeoff between operating frequency, jitter, area and power

than can the digital DLL or the dual-loop DLL [6, 85]. However, the conventional analog DLL does not offer a wide operating range when compared to the two other classes of DLLs.

In recent papers, the analog DLL was designed to operate at multi-gigahertz frequencies using a frequency multiplying delay-locked loop (DLL) [51, 53, 56]. However, for high output frequencies, the edge combiner used in these circuits is hard to design, increasing the design complexity and consuming a large amount of power.

Table II-1. Comparison of the performances between: analog, digital and dual-loop DLLs.

DLL topology	Analog	Digital	Dual loop
Operating range	~ 60MHz - 400 MHz	~ 20MHz -760MHz	~ 60MHz - 1GHz
Lock time	~ 22 clk cycles	~ 120 clk cycles	~ 11clk cycles
Jitter	25ps - 84ps	5ps - 120ps	60ps -150ps
Power dissipation	Up to 100mW	Up to 340mW	Up to 30mW
Example of active area	$0.08mm^2$ in 350nm process [62]	$1.9mm^2$ in 250nm process [33]	$0.13mm^2$ in 250nm process [32]

E.   Summary

This chapter has presented a brief introduction to the basic delay locked loop topologies and described the main differences between DLLs and PLLs. The different classes of delay locked loops: analog, digital and dual mode are presented in detail and a summary of the DLLs performances, based on the existing literature, has been provided.

CHAPTER III

ANALOG MIXED-SIGNAL DELAY LOCKED LOOP DESIGN

A.   Introduction

Since this work focuses, on the mixed-signal analog DLL [1, 8, 18], this section will present in detail each analog DLL sub-circuit. As presented earlier, the analog DLL is composed of four modules (c.f. Figure II-2):

- Voltage Controlled Delay Line (VCDL),

- Phase Detector (PD),

- Charge Pump (CP),

- First order Low-Pass Filter (LPF)

B.   Analog Delay Locked Loop Modules

1.   The Phase Detector (PD)

A phase detector is an analog multiplier or a frequency mixer circuit that generates a voltage signal, which represents the difference in phase between two input signals and is an indispensable element of the DLL. Analog mixed-signal DLLs usually use phase detectors made from digital components such as flip-flop, XOR gates and phase frequency detectors (PFD). Other designs, such as the bang-bang PD or the dynamic PD, illustrated in Figure III-1, are also described in the literature [3, 6, 18, 21, 85]. The output digital signal generated by the phase detector must be translated to an analog voltage in order to control the VCDL output delay [6, 18, 21, 85].
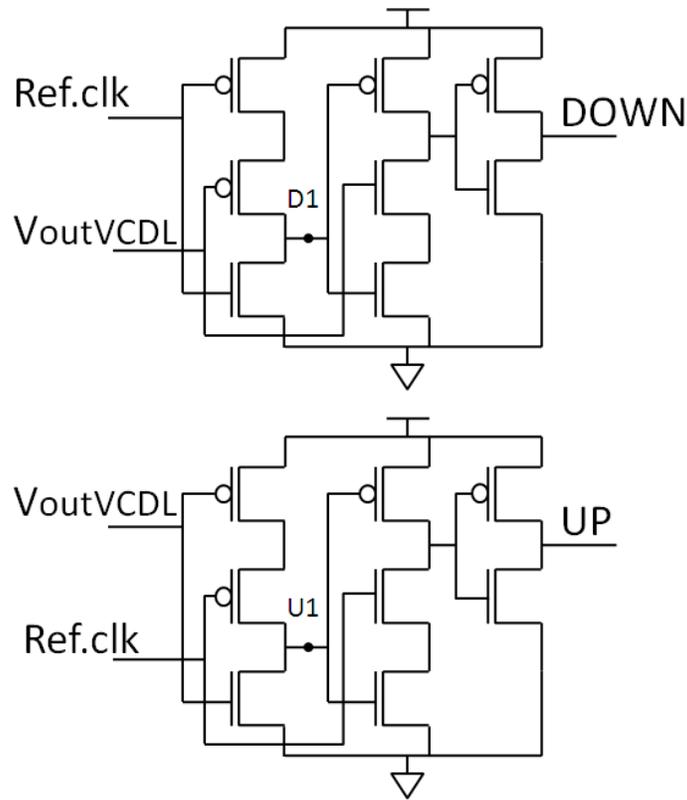
Fig. III-1 Example of phase detector designs, the dynamic PD composed of two stages generating the outputs UP (top Figure) and DOWN (bottom Figure) [3]

When the output of the $V_{outVCDL}$ leads the rising edge of the input reference signal (*Ref.clk*), the output DOWN of the PD is active and generates a pulse proportional to the phase difference between the input signals, *Ref.clk* and $V_{outVCDL.,}$ Meanwhile the output, UP, remain ideally null. Conversely, the UP output is active when $V_{outVCDL}$ lags *Ref.clk* in phase, and the output, DOWN, remains null. When both signals are in phase the outputs of the PD are, ideally, null values.

When the DLL is close to phase lock, i.e. the phase difference between *Ref.clk* and the output of the DLL ($V_{outVCDL}$ or DLL_{out}) is nearly zero radian, the digital PD exhibits a region of very low or zero gain, called the dead zone. Ideally, when the phase of the DLL output differs only very slightly from the input signal (*Ref.clk*), the PD should be able to generate a very short

pulse to drive the VCDL to an ideal phase difference of zero. Because the PD cannot generate such extremely narrow pulses, the input control voltage (V_{inVCDL}) will fluctuate randomly between the bounds determined by the shortest digital pulses the phase detector is able to generate, which in turn defines the dead zone region for the DLL, as illustrated in Fig III-4.

As in PLL circuits, DLLs typically utilize a current-based charge pump topology to convert the detected phase error into a proportional current, while also reducing the phase jitter associated with power supply fluctuations [18]. The CP remains idle when both signals are synchronized. When the PD generates an UP/DOWN signal, the CP sources/sinks current to/from the low pass filter (LPF). The LPF converts the CP current into a voltage (V_{inVCDL}) that is used to adjust the delay of the VCDL. The value of capacitor C in the low pass filter can be calculated using the following equation [18]:

$$Tr = \frac{2.2 * C * T_{clk}}{K_v * 2 * I_{pump}} = \# \; clock \; cycles * T_{clk} \tag{1}$$

where T_r is the response time of the DLL in second, T_{clk} is the input signal period in second, Kv the gain of the VCDL in second per volts and I_{pump} is the CP current in Amperes.

2.   The Charge Pump (CP)

The output of the phase detector is combined into a single output for driving the loop filter. There are two methods for doing this. The first method is called a tri-state output (also known as voltage-based charge pump), but this technique is sensitive to power supply fluctuations [18], and the effect is to modulate the VCDL input voltage. However, tri-state charge pumps, also known as voltage charge pumps [14, 15, 18], are vulnerable to power supply

fluctuations. The second method is the so-called charge pump. In this method, MOS current sources are used, and the modulation of the VCDL input voltage is reduced since the current sources can be made insensitive to power supply fluctuations.

3.   The Voltage Controlled Delay Line (VCDL)

Typically VCDL is based on a single-ended current starved driver topology, composed of several variable-delay elements connected in series, as shown in Figure III-2, to provide a controlled delay of the input clock signal. The single-ended current starved delay cell topology is not practical for most designs because of high sensitivity to noise and low power supply rejection. Alternate VCDL topologies, such as current-mode logic, voltage-controlled resistors, and complementary differential pair topology, are also proposed in the literature to improve noise performance and power supply variation rejection [18, 20, 21]. It is important to note that an analog DLL will either fail to lock or go into false lock if the initial delay of the VCDL is shorter than 0.5 Tclk or longer than 1.5 Tclk, where Tclk is the period of the clock signal. Therefore, the delay of the VCDL must satisfy Equation (2) [18, 21, 85]:

$$\frac{1}{2} * \mathrm{T}_{clk} \ \leq \ \mathrm{t_d} \ \leq \ \frac{3}{2} * \mathrm{T}_{clk} \tag{2}$$
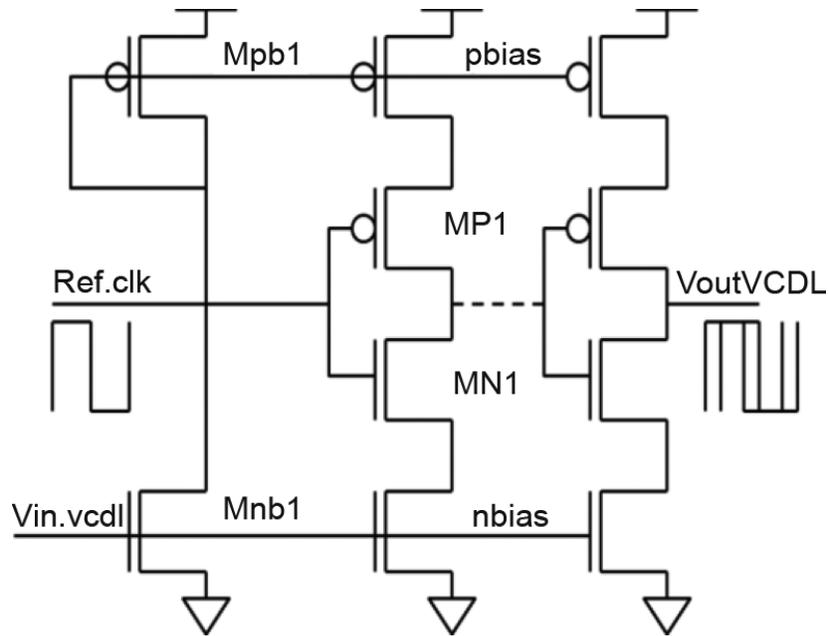
Fig. III-2 Conventional single ended current starved VCDL [2]

Figure III-3 shows the VCDL output delay versus input control voltage (V_{inVCDL}) with a delay range of 0.5 Tclk to 1.5 Tclk for typical analog DLLs operating at 500 MHz, 700 MHz, 850 MHz and 1 GHz.
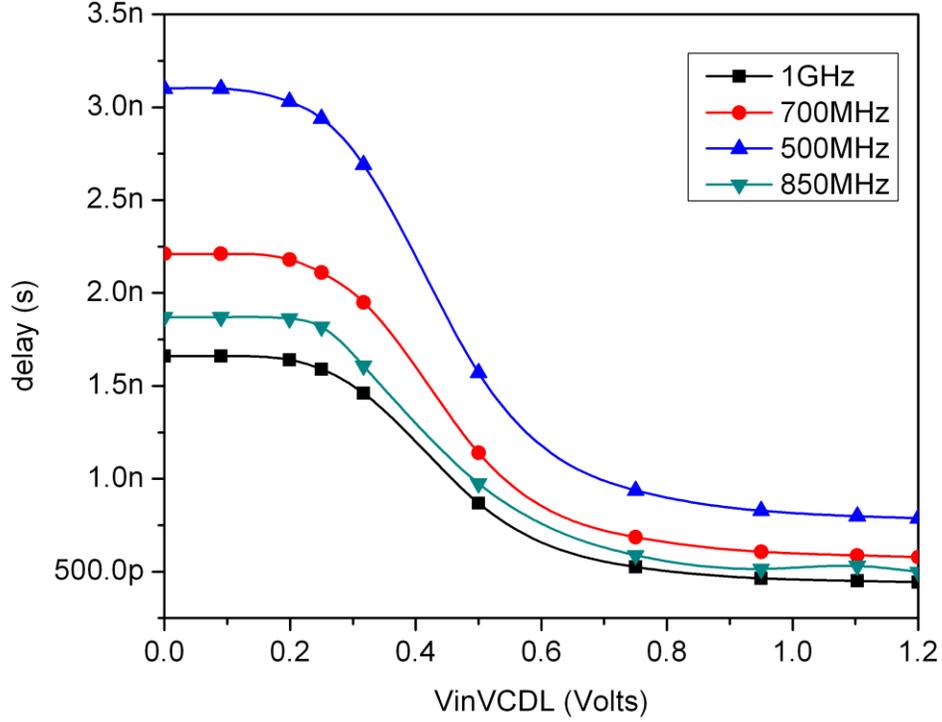
Fig. III-3 VCDL output delay versus control voltage (VinVCDL) for analog DLLs operating at

500 MHz, 700MHz, 850 MHz and 1 GHz [1].

C.   DLL Electrical Characteristics

The basic function of the DLL is to synchronize the delayed output signal ($V_{outVCDL}$) to the reference signal (*Ref.clk*).

The phase output (Φ_{out}), in radians, is related to the reference signal input (T_{clk}) by the following equation:

$$\phi_{out} = \phi_{in} + \phi_d = \phi_{out} + t_o * \frac{2\pi}{T_{clk}} \tag{3}$$

20

Where Φ_{in} is the input phase and t_d is the output delay. The delay can be written in terms of the gain of the VCDL, K_v, and the VCDL input voltage $V_{in.vcdl}$ by the equation:

$$t_d = K_v * V_{in.vcdl} \tag{4}$$

The output of the DLL loop filter, which is also the VCDL input voltage Vin.vcdl can be written as a function of the charge pump current, Ipump, and the low pass filter capacitor C (capacitor inducing the pole in the DLL):

$$V_{in.vcdl} = -\phi_{out} * \frac{I_{pump}}{\pi} * \frac{1}{sC} \tag{5}$$

Therefore the overall transfer function of the DLL can be written using the previous equations:

$$\frac{\phi_{out}}{\phi_{in}} = \frac{1}{1 + \frac{I_{pump}}{\pi} * \frac{1}{sC} * \frac{t_o}{V_{in.vcdl}} * \omega_{clk}} \tag{6}$$

On startup, the DLL passes through two operating phases: the acquisition phase, followed by the lock phase, where the phase difference between $V_{outVCDL}$ and *Ref.clk* is a constant value of 0 or π radians, depending on the design.
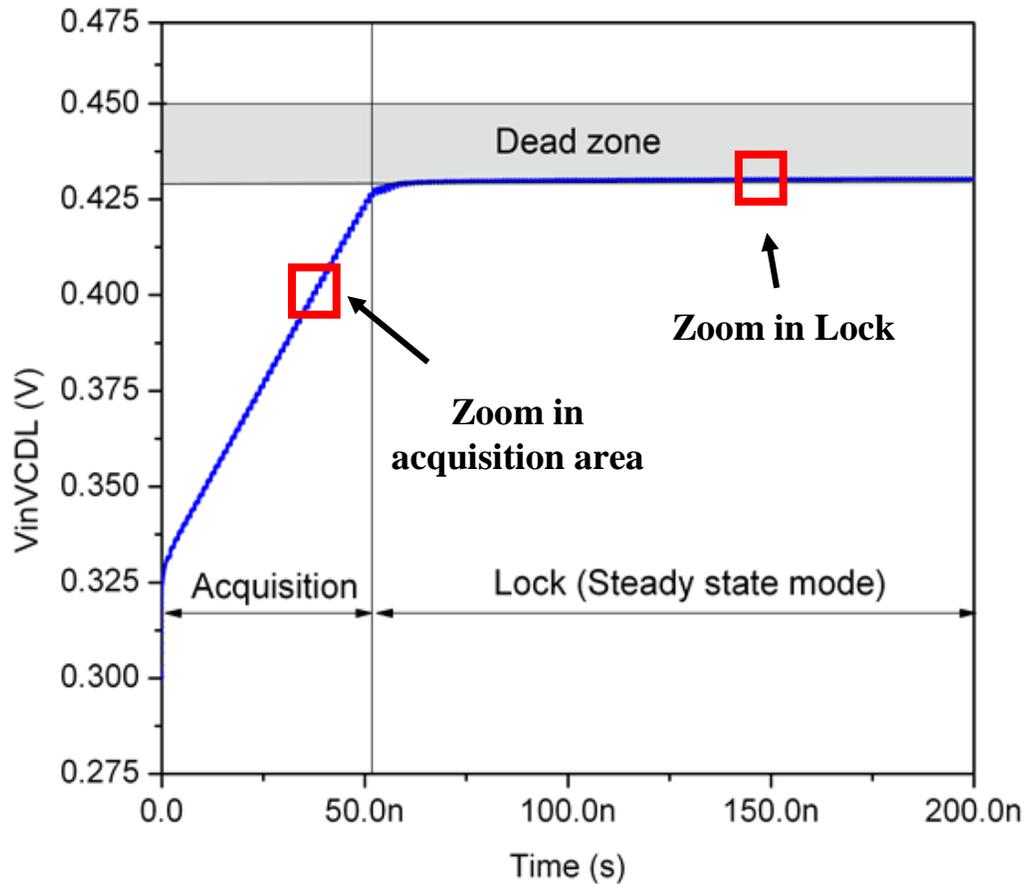
Fig. III-4 DLL acquisition curve for operation at 1 GHz, the dead zone has been measured

(~ 94 ps) and represented by the grey region in terms of V_{inVCDL}.

Figure III-4 shows the acquisition curve of a typical DLL operating at 1 GHz. The acquisition mode represents the time period where clock signal $V_{outVCDL}$ is lagging in phase compared to the input reference signal, *Ref.clk* (Figure III-5). During this time the value of V_{inVCDL} changes with every clock cycle as the PD compares the phases of the two signals and the CP sources current to the LPF. Once signals $V_{outVCDL}$ and *Ref.clk* are in phase (or $\pm \pi$ out of phase) as in Figure III-6, the DLL goes into lock state, where the control voltage V_{inVCDL} ideally remains constant. The value of V_{inVCDL} in lock mode is determined by the dead zone region.
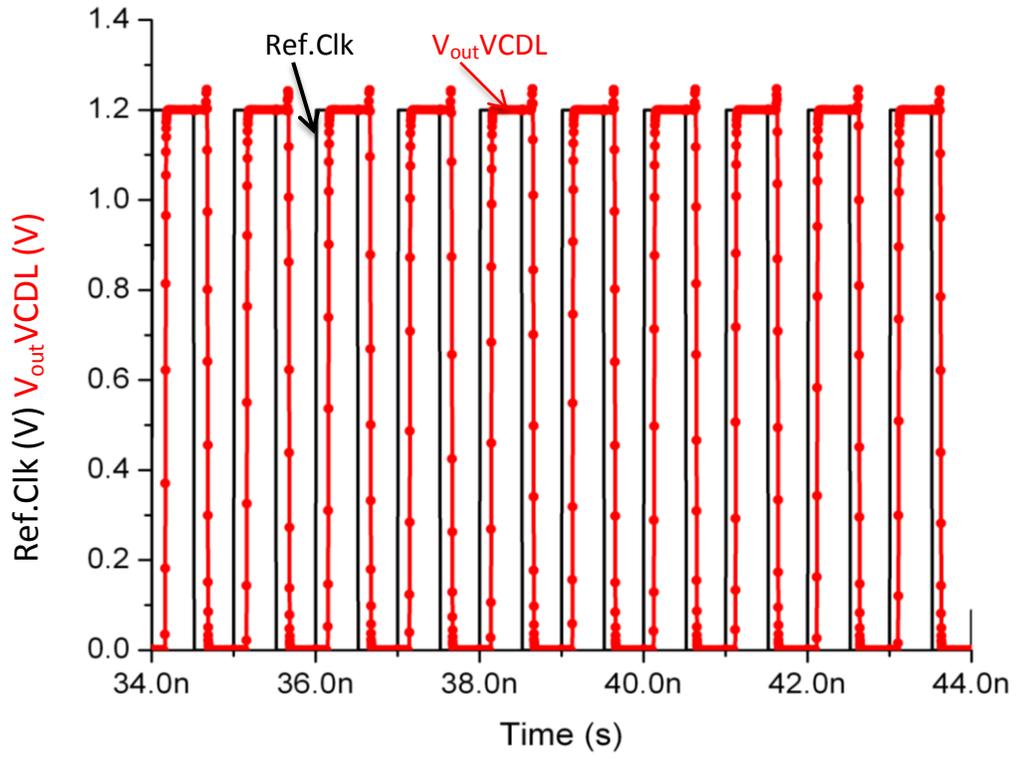
Fig. III-5 Zoom in the acquisition area for an analog DLL operating at 1 GHz.
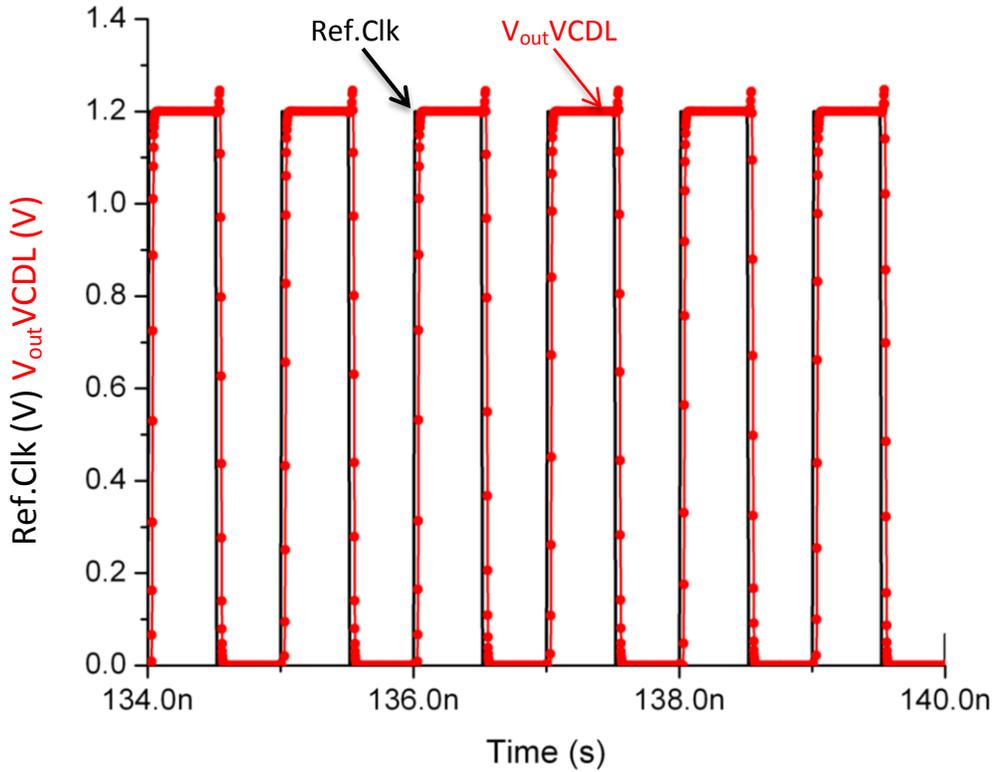
Fig. III-6 Zoom in the lock area for an analog DLL operating at 1 GHz.

D.   Summary

This chapter has provided background information about the mixed-signal analog DLL and its different sub-circuits, i.e. the voltage controlled delay line, the charge pump and the phase detector. The two operating phases of the DLL have been described along with the critical electrical performances and design requirement of the analog DLL.

CHAPTER IV

SINGLE EVENT EFFECTS

A.   The space environment

A variety of energetic particles with energies varying from keV to GeV lie within the natural space environment. Among these particles are electrons and protons are trapped by planetary magnetic fields and produced by either coronal mass ejections (CME) on the sun or cosmic rays produced in supernova explosions [26]. Inside a large spacecraft fuselage, the collision of the primary cosmic beam composed of mostly protons and a small fraction of heavy nuclei, with the spacecraft material, is partially converted into secondary neutrons. There particles fall into categories [26]:

Trapped particles: The earth magnetic field traps a large spectrum of energetic particles also known as the Van Allen Belts. There are two radiation belts, an inner belt and an outer belt. The inner belt is comprise primarily of energetic protons, with energies up to hundreds of mega electron volts (MeV), and electrons up to several MeV, whereas the outer belt is composed primarily composed of accelerated electrons [26].

Galactic cosmic rays: This consists of low fluxes of energetic charged particles that originate outside of our solar system. These cosmic rays are comprised of 87% protons, 12% alpha particles and 1% heavy ions with energies extending up to 1 GeV. The relative abundance of heavy ions in the space environment is shown in Figure IV-1 [26].
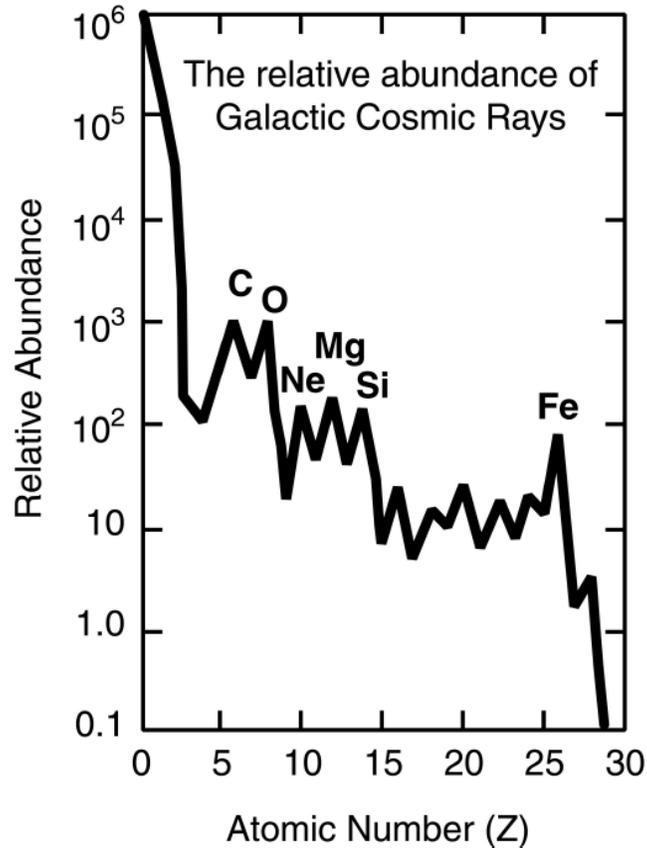
Fig. IV-1 Relative abundance of galactic cosmic rays [26]

Solar particle: are continuously emitted by the sun and consist primarily of protons and heavy ions. During solar events, such as solar flares, the particles have higher energies (MeV to GeV) [26]. In spacecraft, layers of shielding material can be used to block the low energy particles.

B.   Single event effects

When a charged particle passes through an electronic devices it transfers energy to the active silicon area, resulting in an undesirable effect called a single event effect. The charge created by ionization in, or near, a sensitive node of a logic element results in a change of state of the cell, known as a soft error or single event upset (SEU). SEUs do not cause permanent damage

to the logic device or IC's functionality unlike destructive single event effects, such as single event burnout (SEB). SEEs can be divided in two classes of effects, non-destructive and destructive effects, as shown in Tables IV.1 and IV.2, respectively [26, 28].

Table IV-1 Non-destructive single event effects [28]

Single Event Type	Error signature	Targets
Single Event Upset (SEU)	corruption of the information stored in a memory element	Memories cell, latches in logic devices
Multiple Bit Upset (MBU)	several memory elements corrupted by a single strike	Memories, latches in logic devices
Single Event Functional Interrupt (SEFI)	corruption of a data path leading to loss of normal operation	Complex devices with built-in state machine/control sections
Single Hard Error (SHE)	permanent change of state in a memory element	Memories, latches in logic devices
Single Event Transient (SET)	transient response corruption of certain amplitude and duration	Analog and Mixed-signal circuits, Photonics
Single Event Snapback (SESB)	high-current conditions	N-channel MOSFET, SOI devices
Single Event Disturb (SED)	temporary corruption of the information stored in a bit	combinational logic, latches in logic devices

Table IV-2 Destructive single event effects [28]

Single Event Gate Rupture (SEGR)	rupture of gate dielectric due to high electrical field conditions	Power MOSFETs, Non volatile NMOS structures, VLSIs, linear devices
Single Event Burnout (SEB)	damaging burnout due to high-current conditions	BJT, N-channel Power MOSFET
Single Event Latchup (SEL)	high-current conditions	CMOS, BiCMOS devices

C. SEE basic mechanism

There are two different mechanisms involved in SEEs:

1. Charge Deposition:

Incident particle particles deposit charge in a semiconductor device by two primary methods: direct ionization and by indirect ionization involving the emission of secondary particles following a nuclear interaction. Both charge deposition mechanisms can induce single event effects [10].

a) Direct Ionization:

When an ionizing particle, like a heavy ion or a proton, passes through a semiconductor device it loses energy and frees electron-hole pairs along its path, until it loses all its energy in the semiconductor. The linear energy transfer (LET) is the metric used to define the energy loss per unit path length of the particle passing through the semiconductor and has units of MeV.cm^2/mg. The LET of a particle can be easily associated to its charge deposition per unit path length [10]. As a reference, in silicon, a charge deposition of 1pC/um corresponds to an LET of 97 MeV-cm^2/mg. This conversion factor of about 100 MeV-cm^2 /mg is very handy and frequently used to convert LET in charge deposition. The Bragg curve in Figure IV-2 describes the LET of the particle versus depth as it travels through the target material. The Bragg peak represents the peak in charge deposition, usually occurring as the particle reaches energy near 1 MeV/nucleon. As a simple rule, the atomic number of an ion is roughly proportional to its maximum LET [10].

Direct ionization caused by heavy ions is the main charge deposition mechanism for SEUs. However even if protons or lighter particles do not typically produce enough charge to

cause SEUs by direct ionization, recent research has demonstrated that as devices become even more sensitive with technology scaling, upsets due to direct ionization by protons do happen [10, 86, 87].
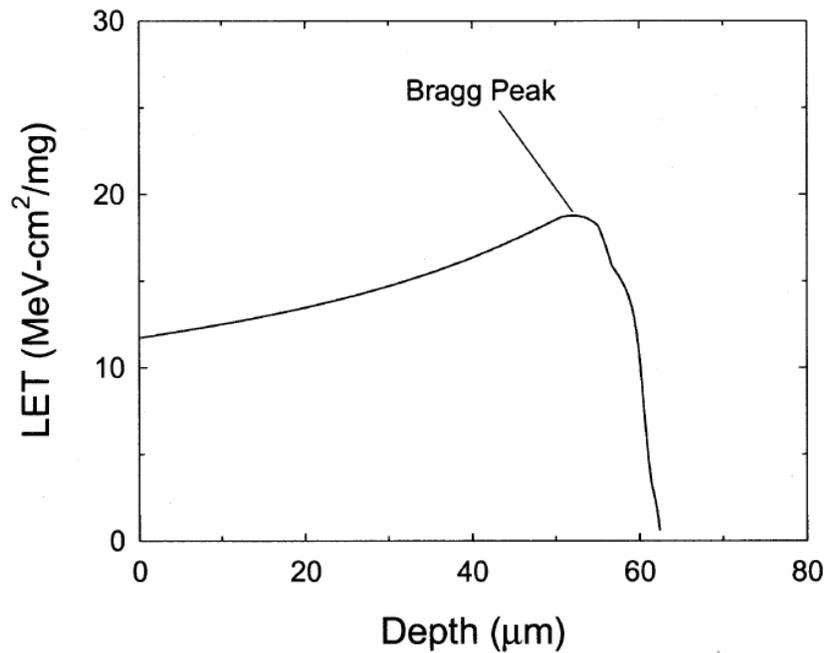
Fig. IV-2 Linear energy transfer curve versus depth for 210 MeV chlorine ions in silicon [10].

b)   **Indirect Ionization:**

Light particles such as protons and neutrons can induce single event upsets due to indirect ionization mechanisms. When high-energy protons or neutrons interact with the semiconductor lattice they may experience elastic or inelastic collision with a target nucleus. Several nuclear reactions may occur may occur from the interaction, examples for protons and neutrons can be found in [29] and possible reactions include the emission of alpha particles or gamma rays [10]. Because alpha particles are much heavier than the original light particles (proton or neutron), they

deposit higher charge densities as they travel resulting in a SEU. Finally, the charge deposited from direct ionization particles is the same as the charge deposited by secondary particles from a indirect ionizing [10], for the same particle with the same energy.

2. Charge collection mechanism:

When a particle passes through a semiconductor device, the reverse-biased P/N junctions are typically the most sensitive regions, due to the high field present. A prominent feature of the event is the distortion of the potential into a funnel shape as illustrated in Figure VI-3 [30]. The drift collection mechanism is significantly enhanced by the funnel, allowing the high field depletion region to extend deeper into the substrate. The substrate doping of the device will directly affect the size of the funnel.
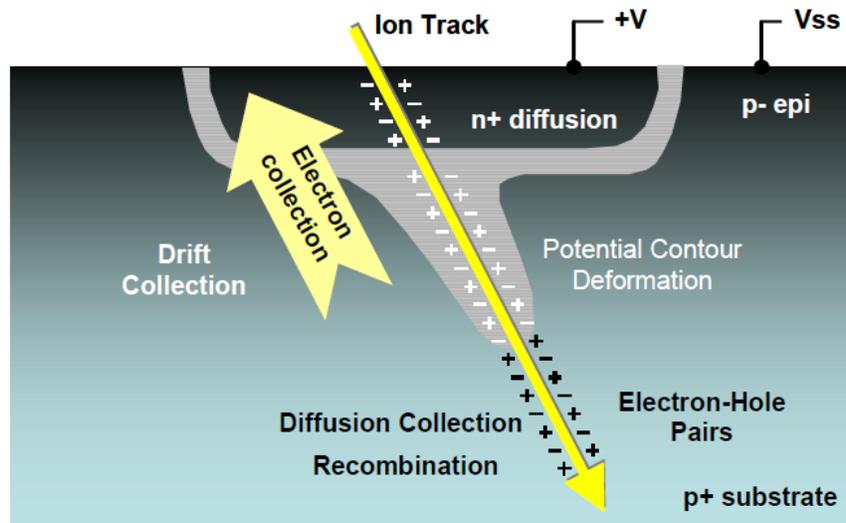
Fig. IV-3 Illustration of reverse-biased N$^+$/P junction struck by an ion [30].

This electric field can efficiently collect the charge deposited by the particle through drift process, leading to a transient current at the sensitive junction [10]. If the charge is within the

built-in electric field of a reverse-biased p-n junction, the electric field causes the holes to be swept into the p-region and the electrons into the n-region. Drift transport is a quick process because the carriers are limited only by their saturation velocity. This process is on the order of picoseconds in duration. The second mechanism inducing charge collection is diffusion transport. Charge deposited within a diffusion length of the junction can be collected and contribute to the voltage transient at the node. Charge collection may occur in multiple nodes depending on the size of the diffusion length and the spacing of transistors. The diffusion process is a slow process in the order of microseconds. The two mechanisms are illustrated in Figure VI-4. The shape of the current pulse in Figure VI-4 is the direct result of the charge collection mechanisms discussed above. The current spike is due to the prompt collection of charge via drift whereas the "tail" part of the current pulse is due to the diffusion induced charge collection. In more advanced technology the current spike will be followed by a plateau effect resulting from the circuit load [31]. Finally, the total charge collected by the node corresponds to the integral of the current over the total duration of the single event.
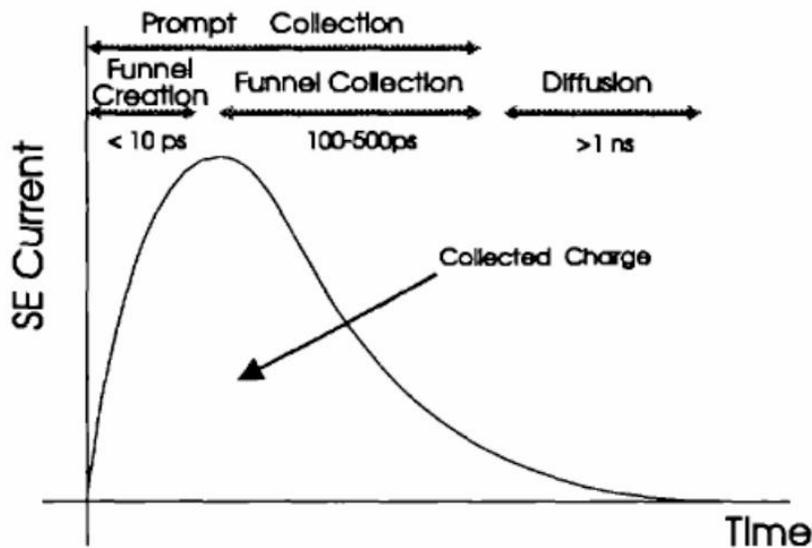
Fig. IV-4 Typical shape of the SE current at a junction. The total collected charge corresponds to the area under the curve [9].

D.   Summary

This chapter has presented a brief introduction to the space environment, the different types of single-event effects (destructive and non-destructive) and the mechanisms involved in the interaction of a particle with a semiconductor. Mechanisms for charge deposition and charge collection crucial to the creation of SETs and SEUs in ICs, such as DLLs or PLLs, have also been described.

CHAPTER V

SINGLE EVENT EFFECTS CHARACTERIZATION AND MITIGATION IN DLLS

A.   Prior SEE Work Related to PLLs and DLLs

As mentioned in section II, Phase Lock Loops (PLLs) and delay-locked loops (DLLs) have been extensively adopted for clock recovery and to eliminate clock signal skews and jitter in high-speed communication integrated circuits [6, 14-16, 18, 20-21, 33-42, 46-48] and, therefore, play a critical role in the proper behavior of the high-speed clock systems. While extensive works regarding SEE characterization and mitigation in PLLs have been published [12-16, 40, 43-44], limited work has been published regarding the single-event effect vulnerability of DLLs in any technology nodes [1, 8]. In fact, there are no published experimental data available to use for validating proposed SEE hardening techniques in digital DLLs.

However, studies focused on hardening digital PLLs [12-16, 43, 44] could be very useful as a guide in the development of SEE-hardened analog DLLs as both circuits are very similar in terms of topology. Some of the relevant work on radiation-hardened PLLs will be presented next.

The charge pump has been identified as the most SEE-sensitive sub-circuit leading to the generation of erroneous pulses [12, 14, 15]. In an effort to develop radiation-hardened digital PLLs, Dr. Loveless et al. [15, 16] proposed several hardening techniques, validated through simulation and experimental data, that could be applied to analog DLLs, such as a redundant biasing circuit to harden the VCO, or voltage-charge pumps to reduce the number of erroneous pulses generated by frequency modulation in the PLL.

In [14, 15], Dr. Loveless et al. proposed a new RHBD technique for the voltage CP (VCP), identified as the most vulnerable module within the PLL, to eliminate its sensitivity to

single event transients. The voltage charge pump is an attractive technique because it is area efficient and would improve the locking time of the PLL. An increase in phase jitter is the main tradeoff of the implementation of the VCP over the current charge pump. Simulations, using double exponential current pulses for charge deposition up to 500 fC show that the VCP can significantly reduce the number of erroneous pulses generated by an ion strike within the PLL.
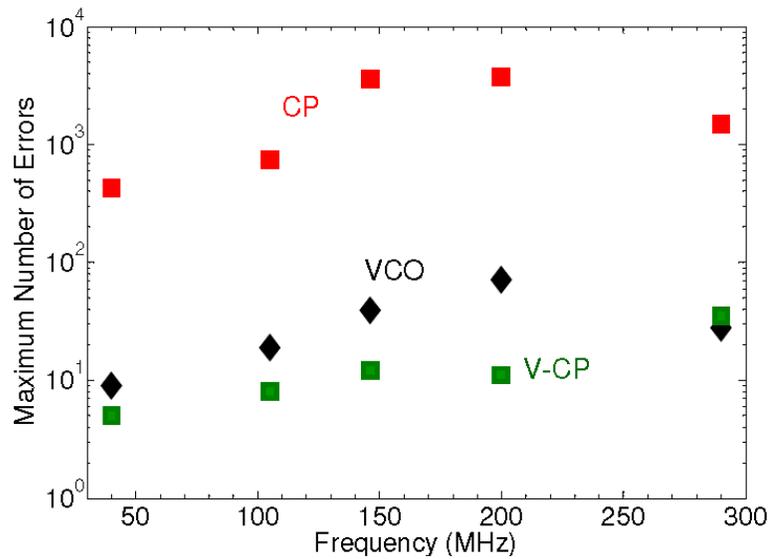
Fig. V-1 Maximum number of erroneous pulses resulting from strikes of energy 30 nJ, in various PLL components, validating experimentally the voltage charge pump as a hardening technique for digital PLLs, the number of erroneous pulses is reduced by 2 orders of magnitudes. CP represents the charge pump of the PLL, V-CP is the voltage charge pump and VCO is the voltage controlled oscillator [14].

Furthermore, experimental validation of the simulation results using a two photon absorption (TPA) laser technique on the PLL implementing the VCP show a dramatic reduction in the number of erroneous pulses generated at the output of the PLL, when compared with the experimental result of the PLL implementing the current charge pump, as shown in Fig V-1 [14].

Based on the experimental results, it would be very interesting to investigate the exportability of such a hardening techniques to reduce the SET sensitivity of the charge pump in analog DLLs.

In [16], the author reduces the single event effects sensitivity of the PLL by targeting the bias stage of the VCO, through redundancy. Since the input-bias stage of the VCO controls the output frequency of the PLL, an ion strike in the bias circuit will result in a control voltage perturbation and thus generates an erroneous frequency modulation of the output signal of the VCO. This control voltage perturbation can be eliminated by replicating it several times and averaging the output using identical resistors in the bias circuit as shown in Figure V-2, thereby minimizing any perturbation due to an ion strike [16]. Simulations results show that duplicating of the bias stage reduced the phase displacement error at the output of the PLL by 35%.
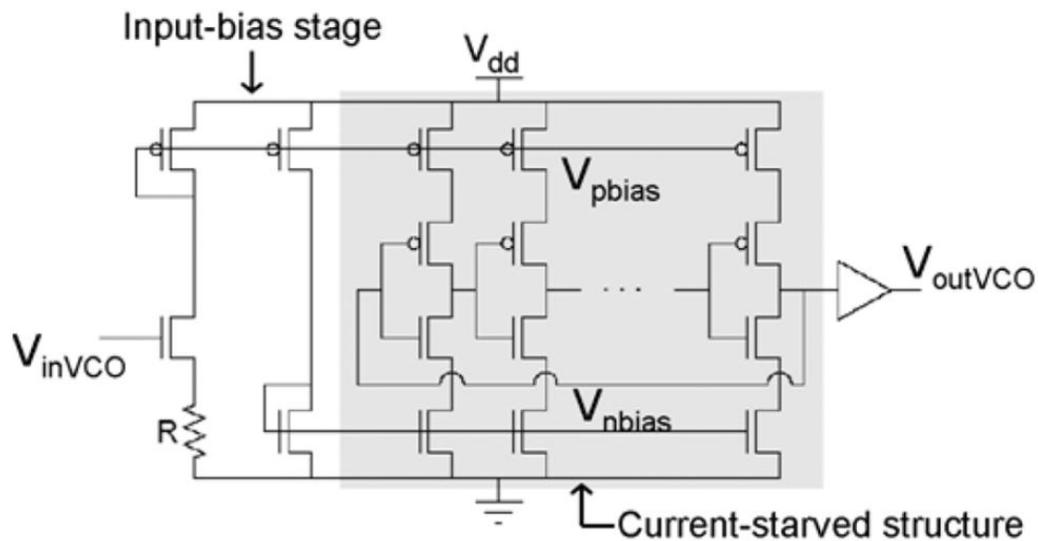
Fig. V-2 Schematic of the VCO implementing the analog redundancy hardening technique applied to the bias stage [16].

In [42], D. Matsuura et al. propose a radiation hardened PLL, where the triple modular redundancy hardening technique is applied to the VCO, as illustrated in Figure V-3. In addition,

since the voltage perturbation derived from the current induced by ion strike lead to frequency modulation of the output clock, the author inserts an additional transistor in series with each transistor of the delay cell, to decrease the induced current.
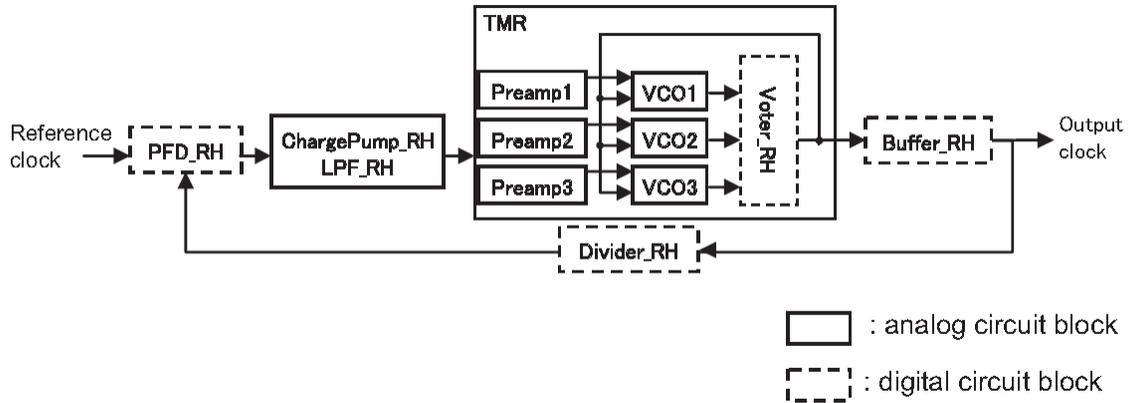
Fig. V-3 Schematic block diagram of the RHBD PLL proposed in [42].

Similar hardening solutions, based on the VCO or CP redundancy to harden the PLL, can be found in the literature [42, 43, 48]. In [44], a radiation hardened PLL design is proposed using a VCO design based on two current-starved ring oscillator structures, with cross-coupled signals, that help to ensure that the effect of an ion strike on one ring is compensated by the other ring is proposed. As in [42, 43] the hardened design also comes with significant area penalty.

As mentioned earlier, extensive studies on SEEs in PLLs are available in the literature, but there is a dearth of such studies for DLLs. In this PhD work, we are proposing to remedy this knowledge gap by investigating and developing hardening techniques for analog delay locked loops. The next section will be devoted to the background work on SEEs in DLLs.

In their work, R. Sengupta et al. proposed a radiation-hardened digital DLL, in [8]. The proposed 133 MHz DLL is illustrated in Figure V-4. In the proposed DLL, all mixed-signal components are replaced with digital circuit blocks. As an example the charge pump is replaced by a digital integrator. In [8], the all-digital DLL is then hardened using error-correction logic and triple modular redundancy (TMR) techniques, known for the associated power and area penalties. In addition, those solutions are non-innovative in terms hardening techniques. In the hardened all-digital DLL, an ion strike in the master-slave architecture will always generates a reliable output, even if any of the circuit blocks have been struck by an ionizing particle, causing either SETs or SEUs. These results were achieved by protecting with TMR the slave delay line and the error correction logic (ECL), which feeds into the slave delay lines. In addition, multiple bit upsets (MBUs) in the digital-loop filter were mitigated by interleaving the layout. The power and area penalty of the radiation-hardened DLL are 32% and 37%, respectively. The circuit design and layout were completed on the TSMC 130-nm process. While the authors claims that a similar architecture can be used for designing radiation-hardened digital PLLs, circuits very different in their topology (c.f. Figure II-4(b) and Figure V-4), this document relies on simulation results obtained with CADENCE ADE tools and does not present any experimental data to validate the RHBD ADDLL design. In addition, this study does not propose a new hardening technique, since the hardened ADDLL design uses the most common RHBD technique: TMR.
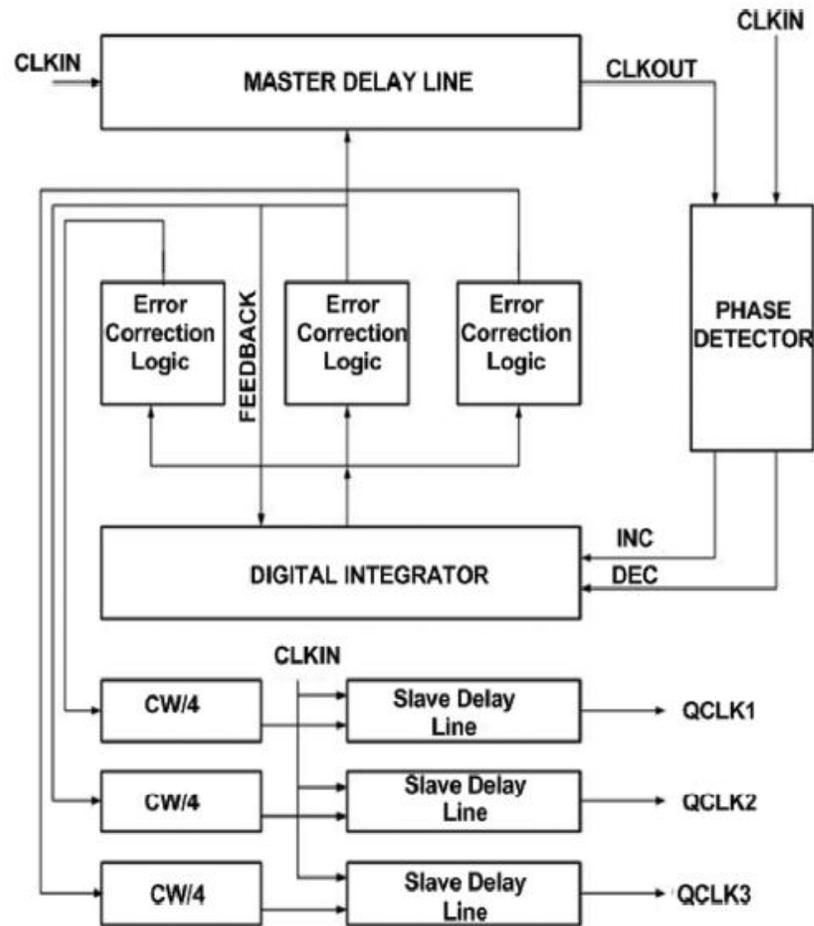
Fig. V-4 133MHz radiation hardened all-digital DLL, using TMR to harden the error correction

logic and slave delay line [8]

B.   SE Characterization of Analog Mixed-signal DLLs

A more detailed study on the impact of SETs in mixed-signal analog DLLs was realized in [1]. After simulating and quantifying the SET error signatures of the DLL sub-circuits, a strong correlation was found between the sub-circuit exposed to an ion strike and the vulnerability of the DLL to SETs. After each simulated ion strike, the perturbations induced in the lock operating conditions of the DLL were measured. Since digital PLLs and analog DLLs are very similar in

their topologies, phase errors and missing pulses due to ion strikes were quantified using the phase displacement error metric described by Dr. Loveless et al. to analyze the SET vulnerability and developed for radiation-hardened solutions for PLL circuits [14, 15, 16], using the equation:

$$\phi_{disp} = 2\pi * \frac{t_e}{T_{clk}} \tag{7}$$

Where Φ_{disp}, is the phase displacement error (or phase error), te is the time error, and T_{clk} is the ideal input clock period. The phase displacement error, Φ_{disp}, represents the time difference (t_e) between the rising (or falling) edge of the reference clock signal and the rising (or falling) edge of the DLL output clock signal, normalized by the period of the reference signal. Following an ion strike in the DLL three type of errors can be observed as illustrated in Figure V-5: 1) missing pulses (where one or more pulses are absent from the output signal), are represented by phase errors in multiplies of 2π, i.e. each missing pulse equals an additional 2π of phase error, 2) inverted lock (IL) errors, represented by phase displacement of π phase radians and 3) duty cycle errors represented by a variation in the output transient pulse width. Due to the large amount of data produced by the simulations, the analysis was limited to worst-case simulated strikes, i.e. at LET of 80 MeV-cm^2/mg in this section.
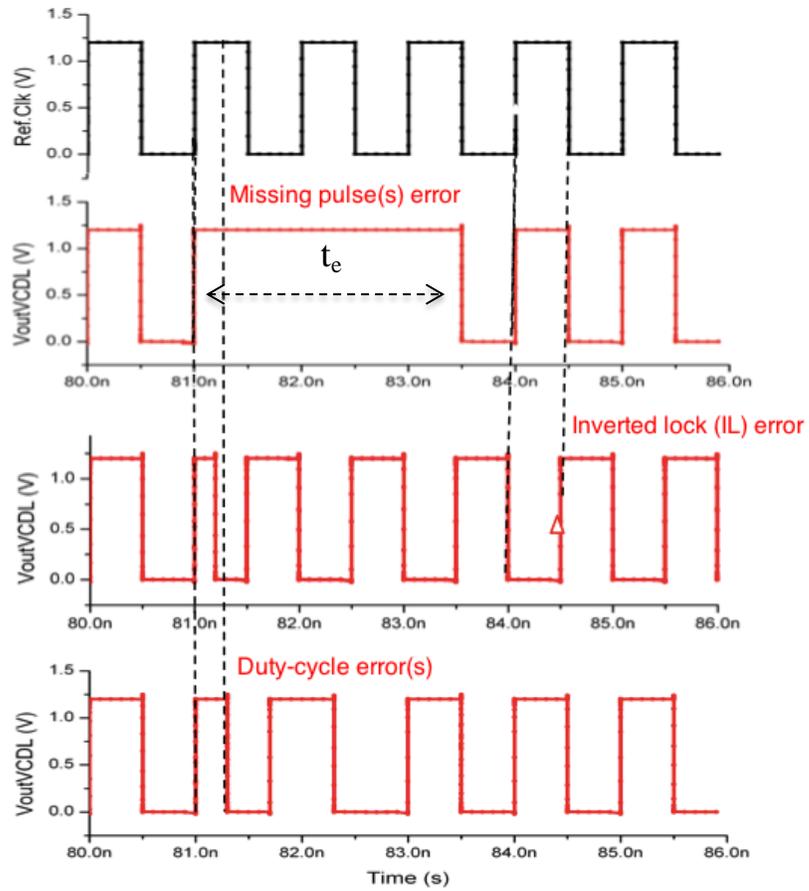
Fig. V-5 Error signatures observed in the DLL following an ion strike: missing pulses (in this illustration two missing pulses), inverted lock error and duty-cycle errors. All the errors are quantified using the phase displacement error metric [1].
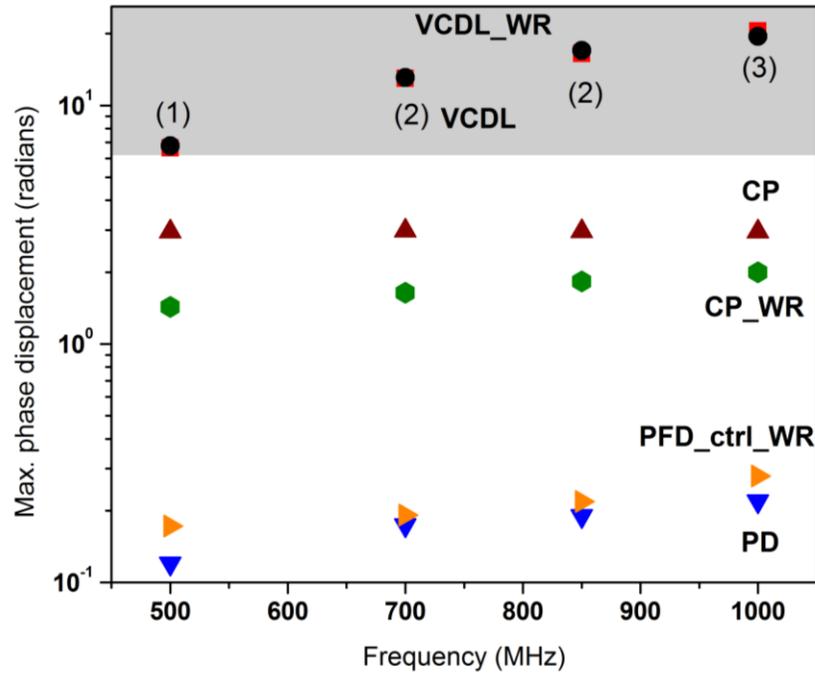
Fig. V-6 DLL SET response in terms of erroneous pulses/phase displacement vs. operating frequency (500, 700, 850 MHz and 1 GHz) for an LET= 80 MeV-cm^2/mg. The VCDL was identified as the most sensitive sub-circuit. The grey region shows where missing pulses are observed, where no missing pulses are observed in the white region. The numbers in parenthesis are the number of missing pulses [1]. VCDL is the voltage controlled delay line, CP is the charge pump, PD is the phase detector and PFD_ctrl is the phase frequency detector. The label "_WR" refers to the wide range DLL topology presented in [1].

Figure V-6 illustrates the phase displacement and number of erroneous pulses, respectively, for four operating frequencies (500 MHz, 700 MHz, 850 MHz and 1 GHz), and for the different DLLs modules, the PD, CP, the VCDL for the conventional DLL, and the start controlled phase frequency detector (PFD_ctrl_WR), the charge pump (CP_WR) and the voltage controlled delay line (VCDL_WR) of the wide range DLL described in [1]. The figure is divided in two parts: the white parts represents the area where no missing pulses are observed, as the phase displacement error is less that 2π radians; the grey region shows where erroneous pulses

were observed with a phase displacement greater than 2π radians. The numbers in parenthesis are the maximum number of missing pulses for each operating frequency.

1.   **Analysis of PD and PFD/Control Circuit Vulnerability**

The phase detector exhibits a maximum phase displacement approximately two orders of magnitude below that of the VCDL. Additionally, no erroneous pulses or significant phase displacement errors were observed due to strikes in the PD. Therefore, as was previously observed with the PLL's PFD sub-circuit [12, 14] and in the wide-range DLL's PFD with control circuit module, ion strikes on the DLL phase detector result in the least or no significant errors at the output [1]. This result is a consequence of the continuous signal comparison of the PD or the PFD, reducing the sub-circuit's output signal perturbation duration after an ion strike. The PD or PFD, comparison process is much faster than the DLL response time, which corrects the error before the CP can significantly alter the VCDL input voltage value. The dead zone of the PD is less than 40 ps, and slightly decreases with operating frequency, so the PD phase displacement will increase as the operating frequency increases. The decreasing dead zone effectively increases the PD lock range slightly. Figure V-7 illustrates the phenomenon with the PD dead zone, for a strike on the current CP of a 1 GHz DLL with an LET of 20 MeV-cm^2/mg. After the phase perturbation, the value of V_{inVCDL} is different from the original locking value, thus allowing one to also identify the dead zone boundaries [1].

2.   **Analysis of Voltage-Controlled Delay Line Vulnerability**

As shown in Figure V-5, the VCDL presents the highest sensitivity to irradiation, and is the dominant contributor to the SET vulnerability of both of the DLL topologies. Strikes in the VCDL result in significantly larger phase displacement values over any other DLL sub-circuit,

and the phase displacement increases with increasing clock frequency. In addition, as illustrated in Figure V-5, the VCDL is the only DLL sub-circuit that can create erroneous pulses at the output. The maximum phase displacement and the number of erroneous pulses both increase with increasing operating frequency, and strikes within any stage the VCDL, including the bias circuit, can result in erroneous output pulses and duty cycle errors [1].

3.   Analysis of the Charge Pump Vulnerability

Depending on the topology used, the sensitivity to single event transients of the CP, for worst-case SET response, varies. An ion strike in the charge pump results in the second highest phase displacement as well as the longest recovery time after irradiation. A maximum recovery time of 295 ns was measured for strikes in the CP at an LET of 80 MeV-cm^2/mg. Similar strikes in the VCDL and PD sub-circuits did not result in persistent errors [1]. Therefore, in terms of recovery time, the strikes in the CP result in the worst-case response. However, as the PD acquisition range (± π radians) can result in a DLL lock at 0 or π radians out of phase, the DLL is expected to return to a stable locked condition following the ion strike where the input and reference signal are π radians out of phase rather than the initial locked condition of 0 radians. Therefore, regardless of the operating frequency, the maximum phase displacement for strikes in the CP is π radians, as shown in Figure V-6. Consequently, no erroneous clock pulses were measured for any simulated LET values for SE hits on the CP. This is in contrast to previous studies on PLL circuits [14-15] where phase displacement values at least 2 orders of magnitude greater than π radians were observed for hits on CP. Strikes in the CP of a PLL, for example, can generate phase displacement values and erroneous pulses orders of magnitude larger than ion strikes on any other sub-circuit, as these strikes perturb the VCO's control voltage and modify the PLL's output frequency. Because the DLL does not contain an internal oscillator, any changes in

the VCDL's control voltage in a DLL result in only phase errors, and cannot modulate the output frequency.

For the worst-case simulated LET strike, the CP was the only sub-circuit exhibiting the inverted lock error. This error is a consequence of the CP/ LPF and PD operating modes. The ion strike forces the VCDL input voltage to change abruptly, thereby drastically changing the DLL output delay and inducing the DLL to leave the lock condition. As the PD acquisition range is $\pm \pi$, the system will be forced to lock at this "false" phase instead of recovering to 0 phase radians (as can be observed in Figure V-8, for a strike within the CP at an LET = 80 MeV-cm^2/mg), leading to a permanent phase error at π radians. This error requires hard resetting the DLL to restore the circuit's proper behavior.
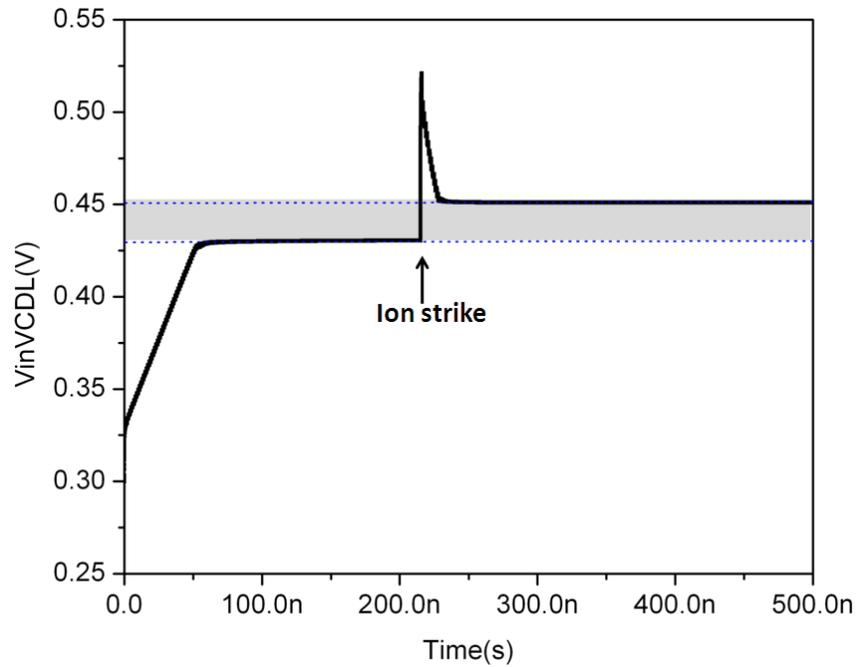
Fig. V-7 Illustration of the PD dead zone phenomenon for a 1 GHz DLL acquisition curve with strike on the CP for an LET of 20 MeV-cm^2/mg, generating a significant phase shifts. VCDL input voltage V$_{inVCDL}$ does not regain its original voltage value after the perturbation because of the dead zone (grey region) [1].
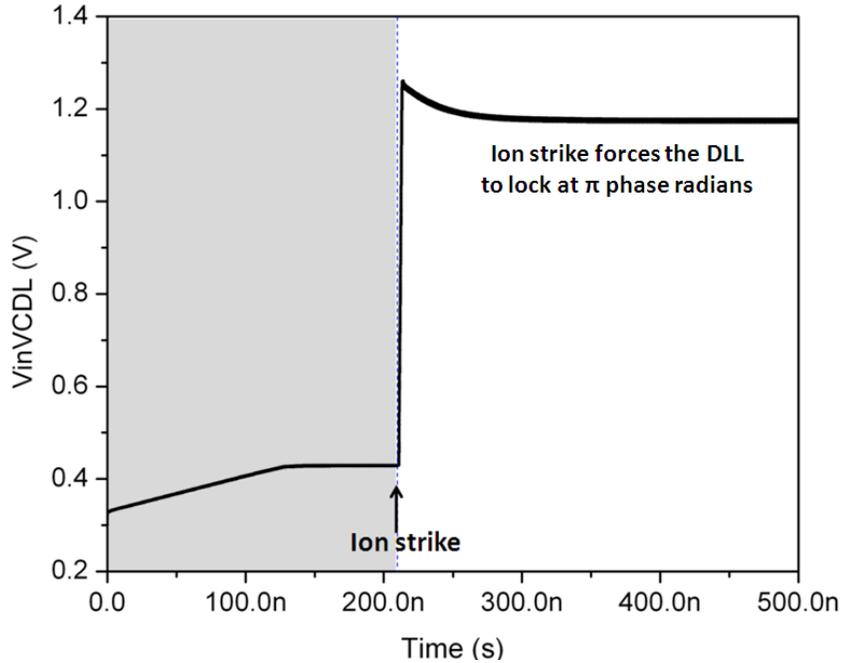
Fig. V-8 DLL acquisition curve, at 1 GHz for an LET of 80 MeV-cm^2/mg, illustrating the

persistent inverted lock error generated by an ion strike in the analog DLL charge pump [1].

The SET responses observed in the DLL for strikes on the VCDL and the CP have different signatures when compared to those of a PLL. In the PLL, the charge pump was the most sensitive sub-circuit, followed by the voltage-controlled oscillator [12, 14, 15]. In contrast, the VCDL is the most sensitive sub-circuit of the DLL, as this sub-circuit exhibited the highest phase displacement along with missing pulses. The CP had the second highest phase displacement of the system, but also generated the persistent inverted lock error.

To summarize, simulations showed that single-events in the phase detector or the phase frequency detector for wide-range DLL have little impact on the DLL output signal in lock mode. Strikes in the charge pump resulted in moderate phase displacement values and increased recovery times at the output of the DLL, but no missing output pulses were observed. However, an ion strike on the CP can generate a persistent false lock (or inverted lock) error in the analog

DLL, when an ion strike within the CP of a wide-range DLL generates no inverted lock error. Simulation results also showed and verified that the implementation of a voltage-based CP in the analog DLL was a very efficient RHBD technique to mitigate inverted-lock errors generated by the CP, and presented better SET response than the wide-range DLL. In terms of phase displacement and erroneous pulses, the VCDL exhibited the highest vulnerability in the DLL for both topologies, with ion strikes leading to worst-case SET error signatures.

C.   Summary

This chapter provided the background information on the SEE characterization and mitigation work in PLLs and DLLs. While a lot of work has been published on radiation hardened PLLs, the published work on the hardening DLLs is nearly non-existent. Similar to PLLs, DLLs are crucial for ensuring the proper behaviour of clock distribution networks or pipeline ADCs. In this PhD work, we are filling the knowledge gap regarding the characterization and mitigation of single events in DLLs.

CHAPTER VI

RADIATION HARDENED BY DESIGN (RHBD) TECHNIQUES TO MITIGATE MISSING PULSES IN ANALOG DLLS

A.   Introduction

Following an ion strike within the DLL, three distinct error signatures, missing pulses, persistent inverted lock error and duty cycle errors, can be observed. In order to develop a radiation hardened by design DLL, these errors must be mitigated or reduced. Since some of them are more severe than others, the hardening effort will be primarily directed toward the worst-case SEE response, i.e. the missing pulses and inverted lock errors.

B.   Hardening Technique 1: The Hardened Complementary Differential Pair VCDL

A RHBD solution based on a complementary differential pair delay cell topology, illustrated in Figure IV-3, is proposed in this section [2, 73]. When compared to all other DLL sub-circuits, strikes in the VCDL result in significantly larger phase displacement values. Furthermore, the VCDL is the only sub-circuit that can generate missing pulses at the DLL output due to single-event errors [1]. As concluded in section V, the hardening effort should be focused on this sub-circuit.

The single-ended current starved delay cell topology, illustrated in Figure VI-1, is typically used for DLL and PLL designs due to its low design complexity and power consumption [1, 2, 6, 16, 18, 21]. It provides a wide operating frequency range through the adjustment of the internal currents and, hence, the switching delay, of each delay cell. Figure VI-2 shows the worst-case SET response of the DLL output following strikes on the drains of PFET

transistors Pbias1, MP1, and Mpb1 in a current-starved delay cell [2]. The maximum phase error and the number of missing pulses both increase as operating frequency increases. Ion strikes within any portion of the VCDL, including the bias circuit and individual delay stages, can result in missing output pulses.
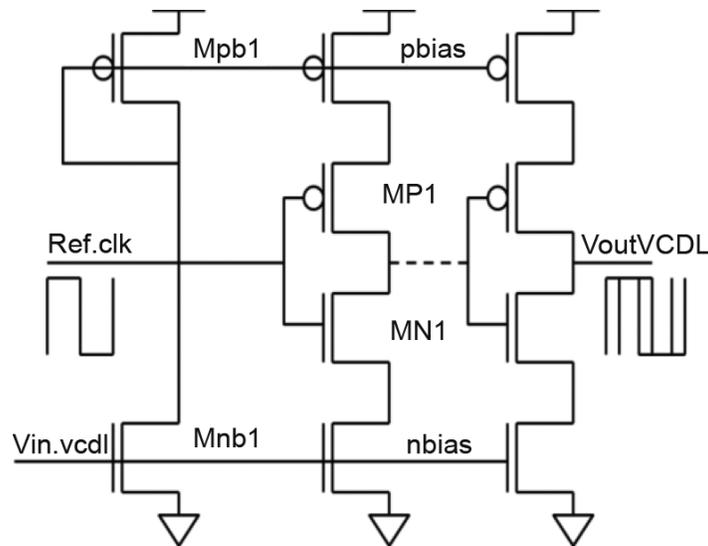
Fig. VI-1 Conventional single ended current starved VCDL [2]

RHBD techniques have been developed to reduce or mitigate erroneous pulses (additional or missing pulses) in current-starved VCDL designs. For example, hardening techniques using a double-redundant biasing circuit or a triple-modular redundancy (TMR) technique have been demonstrated [16, 42]. The redundant biasing circuit technique duplicates the VCO (or VCDL) input biasing circuit, using resistors in the bias circuit to attenuate the impact of a single-event transient, while the TMR based technique is applied to the delay cells, excluding the input bias circuit [16]. Although effective, these RHBD techniques have the disadvantage of increased area requirements, either by adding resistors to the bias circuits or

tripling the number of delay cells, and increased power consumption. Any alternative RBHD circuit must be competitive with previous techniques in terms of area and power requirements.

1.   The Hardened Complementary Differential Pair VCDL Topology

Differential delay elements are typically used in general analog mixed-signal DLLs and PLLs when power supply or substrate noise is a concern, as the differential topology provides common-mode noise rejection. Various differential delay cell topologies can be found in the literature, including cells with resistive loads, triode loads, symmetric loads, and cross-coupled loads [18, 20, 21, 77]. As shown in Figure VI-2, the topology used for this work was based upon the cross-coupled differential pair delay cell, due to the inherent switching hysteresis and improved noise immunity of this type of load. Nominal MOSFET width-to-length ratios are also shown in Fig VI-2, where the gate length equals 80 nm.
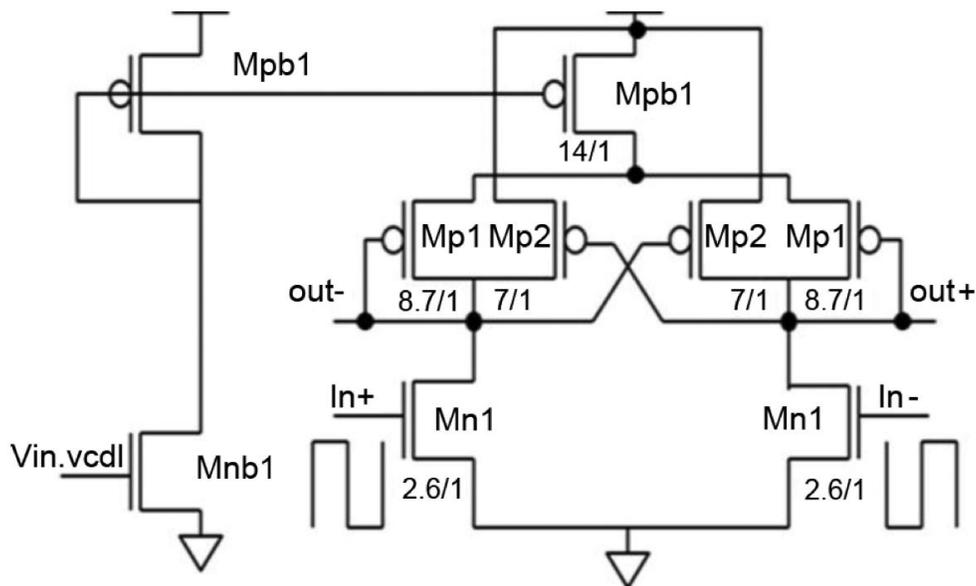
Fig. VI-2 Unhardened complementary differential pair delay cell with its bias stage with transistors width over length ratio in 90nm technology node [2]

The complementary differential pair delay cell has three additional transistors when compared to a current-starved delay cell (typically used for DLL and PLL designs due to its low design complexity and power consumption), thus requiring somewhat greater area and power. For the purposes of simulation, inverters with negligible delay were added to the outputs of the complementary differential pair VCDL to provide a rail-to-rail comparison between the reference input signal and the differential outputs, and for comparison with the results of the current-starved VCDL design.
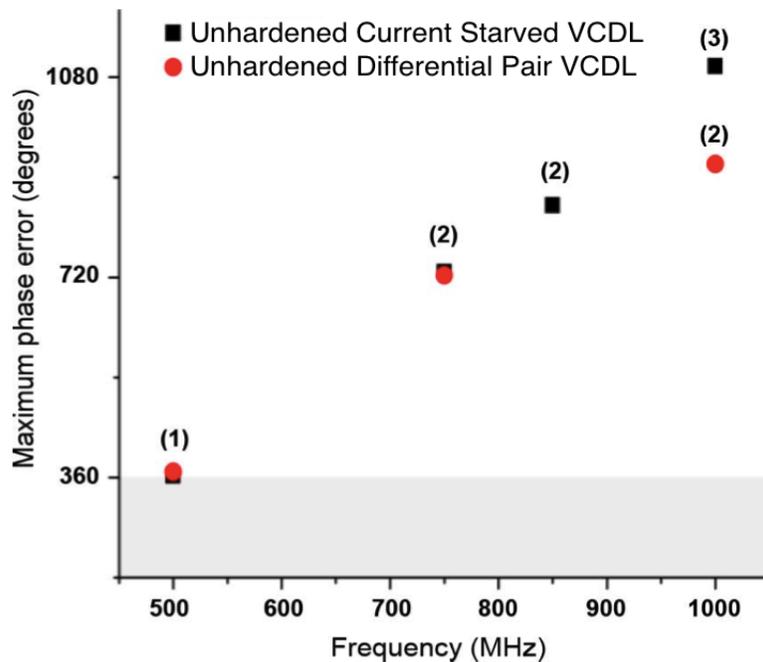
Fig. VI-3 90 nmVCDLs maximum phase error versus operating frequency for an LET of 80 MeV-cm^2/mg. Missing pulses are observed in the unhardened current starved and complementary diff. pair VCDLs. The numbers in the parentheses represent the missing pulses generated by the VCDL. [2].

Simulation results of the unhardened complementary differential pair VCDL showed that the output nodes of the differential delay cells were the most vulnerable to single-event transients, generating missing pulses as illustrated in Figure VI-3. In contrast, the remaining internal circuit node, i.e. the drain of transistor Mpb1 in the delay cell, exhibited no significant sensitivity to SETs.

The novelty in this hardening technique is that the complementary differential pair VCDL was subsequently hardened by modifying the feedback transistors within each delay cell composed of PFETs Mp1 and Mp2 in each half-cell of Figure VI-4. In other words, by increasing the width-to-length (W/L) ratios of both Mp2 feedback transistors and Mp1 with respect to the other transistors in the cell, single-event transients were significantly reduced while maintaining the required tuning range and operating frequency for the VCDL, with only a small penalty in increased area and power.

(a)

(b)
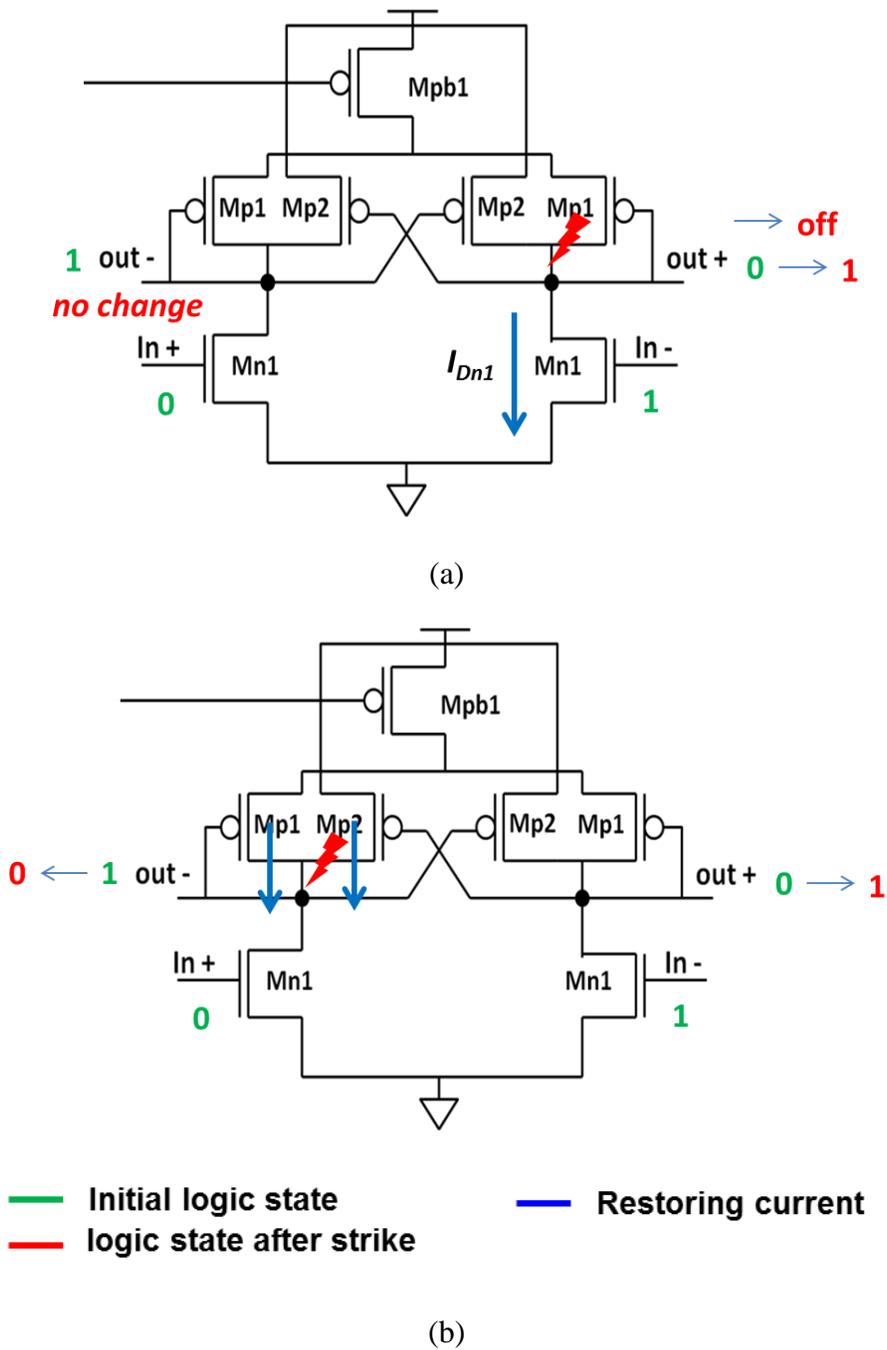
Fig. VI-4 RHBD VCDL SET mitigation mechanism for a) a N-hit at the drains of the input transistor, I_{Dn1} is the restoring current of the delay cell, b) a P-hit at the drains of the diode connected or feedback transistors, I_{Dp1} and I_{Dp2} are the restoring current of the delay cell.

The SEE mitigation mechanism varies depending on the ion strike location as illustrated in Figure VI-4. If an ion strikes the drain of Mn1 input transistor (c.f. Figure VI-4 (a)), the output

out+ of the DLLs will go from '0' to '1' logic. The expression of the restoring current of the VCDL, I_{rest}, is given by the following equation:

$$I_{rest} = I_{Dn1} = K_n * \frac{W_n}{2L_n} * (V_{GS} - V_{tn})^2 \tag{8}$$

Since the transistors Mp1 and Mp2 on the output side "out+" are off, the drain current of the NMOS transistor Mn1 is the major contributor of the restoring current of the VCDL. When an ion strikes the drain of Mp1 and Mp2 PMOS transistors (c.f. Figure VI-4 (b)), the VCDL output "out –" will switch from '1' logic to '0' logic. The restoring current of the VCDL is given by:

$$I_{rest} = I_{Dp1} + I_{Dp2} = 2 * (K_p * \frac{W_p}{2L_p} * (V_{GS} - V_{tp})^2) \tag{9}$$

The value of the drain current of Mp1 dependents on the delay cell's bias voltage Vbias. Therefore, I_{rest} value will also vary with V_{bias}. When $V_{bias} \sim 0V$, Mpb1 is close to saturation and thus $I_{rest} = 2*I_{Dp1}$. As a rule of thumb, V_{bias} should be approximately $\sim 0.3*VDD$ for I_{Dp1} to significantly contribute to the restoring current.

When compared to the restoring current of the single ended current starved for a strike on the drain of the PMOS, the restoring current of the complementary pair will be at least twice that of the current starved. This shows the advantage of the complementary diff. pair over the current starved VCDL in terms of resiliency to SETs. However, this advantage of higher restoring current when compared to the single ended current starved will not eliminate missing pulses in the unhardened complementary diff. pair VCDL, as shown in Figure VI-5 (a). By increasing Mp2 and Mp1's W/L, the restoring current at the output node is significantly increased, thereby reducing the SET width as shown in Figure VI-5 (b). As a transistor sizing rules, to harden the

VCDL, the minimum width of the NMOS Mn1 is $W_{n\text{-}min} = 1.5\ W_{nom}$, where W_{nom} is the nominal size of Mn1 in the unhardened VCDL, and the minimum width of the PMOS Mp1 and Mp2: $W_{p1} = W_{p2} = \sim 2\ W_{n\text{-}min}$. Increasing the W/L ratios of the Mp2 feedback transistors will also increase the capacitances of the output nodes, "out+" and "out-" in Figure VI-4, increasing the feedback delay of the cell and thereby reducing the probability that the upset will be latched at the differential output and propagated to the next stage.
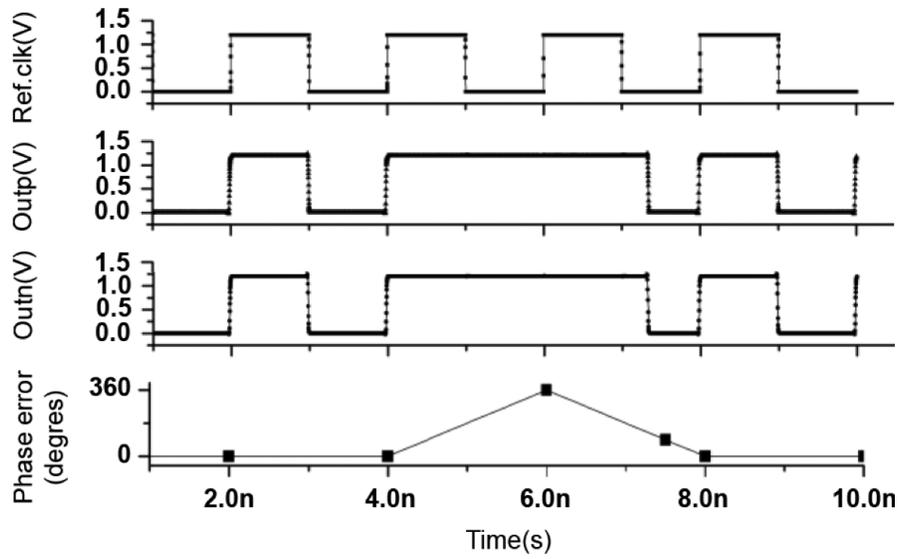
(a)
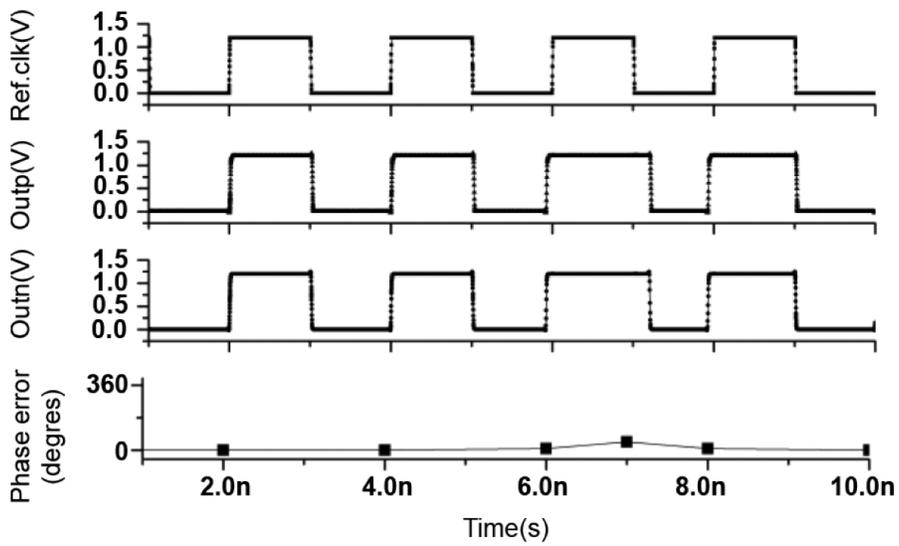
(b)

Fig. VI-5 SET response, in terms of missing pulses and phase error (in degree), generated by the 90 nm complementary differential pair VCDL for an LET of 80 MeV-cm^2/mg at 500MHz and V_{inVCDL} = 340 mV for a) the minimum-sized complementary diff. pair VCDL and b) for the hardened complementary diff. pair VCDL [2]. No missing pulses are observed at the complementary outputs (outp or outn) of the hardened VCDL.

Figure VI-6 compares the simulated response of the hardened and unhardened VCDL for different operating frequencies and VCDL input voltages over the operating range of the DLL. The two color maps represent the number of missing pulses as a function of the phase error, i.e. one missing pulse equals 2π of phase error. The SET sensitivity, in terms of missing pulses and phase displacement error, of the hardened FD VCDL was significantly lower than that of the unhardened VCDL, over all operating conditions and design frequencies, with missing pulses mitigated by approximately 84% for all simulations combined.
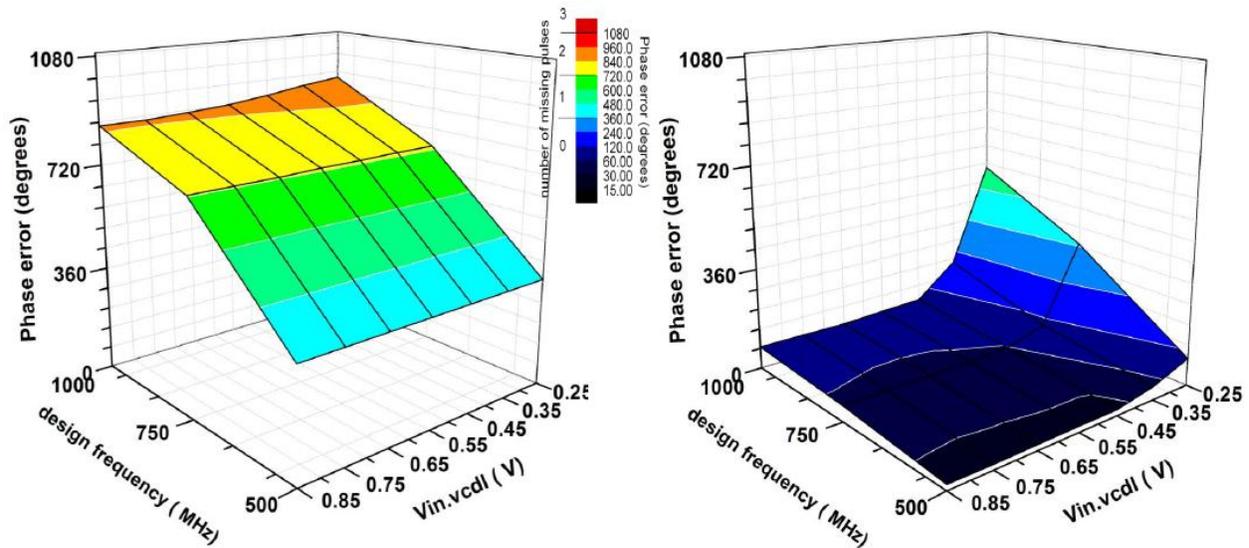
Fig. VI-6 SET response, in terms of missing pulses and phase error (degrees), generated by the complementary differential pair (DP) VCDL for an LET of 80 MeV-cm2/mg at 500 MHz, 750 MHz, 1 GHz, over all the VCDL operating range, for: (left) the unhardened DP VCDL, (right) the hardened DP VCDL [2]. Missing pulses were reduced by ~ 80% in the hardened VCDL. The color mapping represents the number of missing pulse(s) [2].

If we compare the RHBD VCDL power and area penalties for unhardened and conventional TMR RHBD designs, the RHBD VCDL design requires only 12.6% more area, as

opposed to a 300% area penalty for a TMR design. The obvious tradeoff is that a properly implemented triple-redundant design has the potential to completely eliminate missing pulses as opposed to the 84% reduction of the proposed FD design. While the RHBD VCDL circuit cannot replace spatially redundant designs, it does provide much of the hardening benefit of triple-modular redundancy with only modest increases in area and power.

a)   ***RHBD solution to mitigate inverted lock errors in the charge pump***

Most modern charge pumps are current-based since voltage-based (also called tri-state) charge pumps can be significantly affected by power supply fluctuations [18]. As in a current-based charge pump, the output stage of the sub-circuit is directly connected to the capacitive node of the loop filter. An ion strike within the sub-circuit will either deplete or deposit charge on the loop filter capacitor, thus directly affecting the VCDL control voltage by abruptly increasing or decreasing its value and therefore forcing the DLL to exit the lock mode. In the worst-case scenario, this voltage perturbation will generate inverted lock errors, i.e. the DLL lock at π phase radians, as shown in Figure VI-8.

One of the solutions to harden the CP to SET is the hardened voltage VCP, presented in section IV, and illustrated in Figure VI-7. This hardening solution was originally developed for PLLs to reduce the voltage perturbation generated by ion strikes [14, 15] but no study on the applicability of this technique to DLLs has been done, as mentioned in section IV. Since the SET responses observed in the DLL for strikes on the VCDL and the CP have different signatures when compared to those of digital PLLs [14, 15], the implementation of the VCP in the DLL is an interesting new study. The voltage charge pump has fewer vulnerable nodes than does the current charge pump. Also, the rate of charge sourcing and sinking is increased leading to a faster

59

acquisition/recovery time which mitigates the inverted lock error, as shown in Figure VI-8 and Figure VI-9 (for the IBM 90-nm and IBM 180-nm technology nodes, respectively). Finally, by adding a resistor to the loop filter (R1, in Figure VI-7), the vulnerable output node of the charge pump is isolated from the control voltage of the VCDL. Another solution to harden the CP in the DLL would be to use a complementary current limiter (CCL) in addition to the current charge pump [40]. This solution shows responses to single events similar to those in the VCP. However, even if this hardening technique does not add critical nodes to the circuit, the size of the overall hardened DLL is increased by introducing two operational amplifiers in addition to the sense resistor. Power consumption of the circuit is also increased.
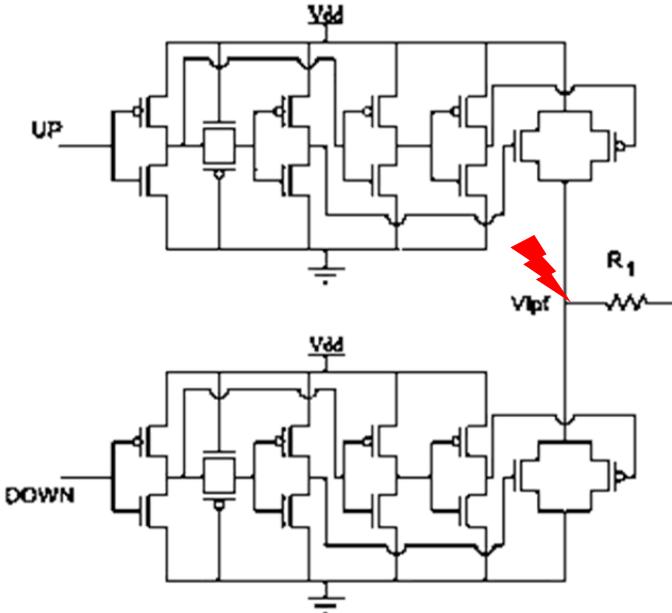
Fig. VI-7 Schematic (transistor level) of the voltage charge pump [15].
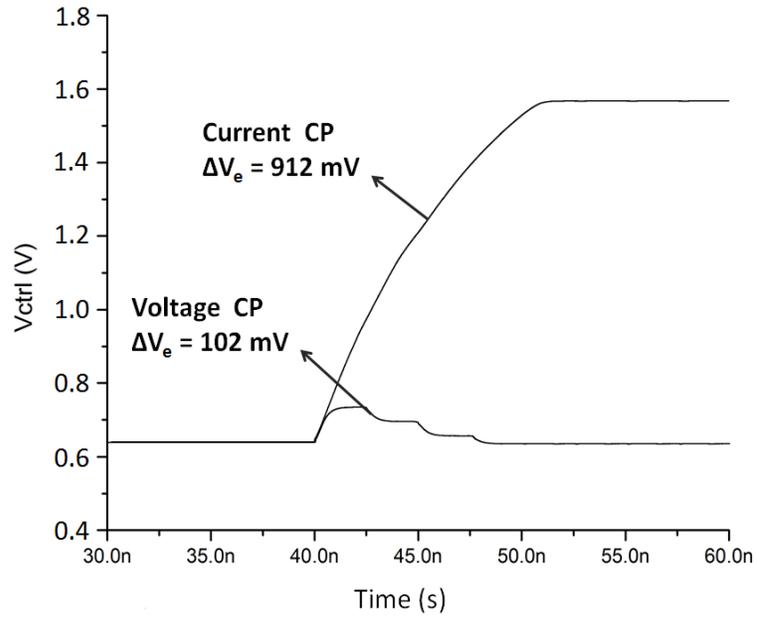
Fig. VI-8 SET response of the 90nm DLL for a strike within the charge pump at an LET = 100 MeV-cm^2/mg, for the current based charge pump, generating an inverted lock error, resulting from a control voltage (V_{ctrl}) perturbation of 912 mV, and the voltage based charge pump, reducing this voltage perturbation to 102 mV, therefore eliminating the inverted lock error.

61
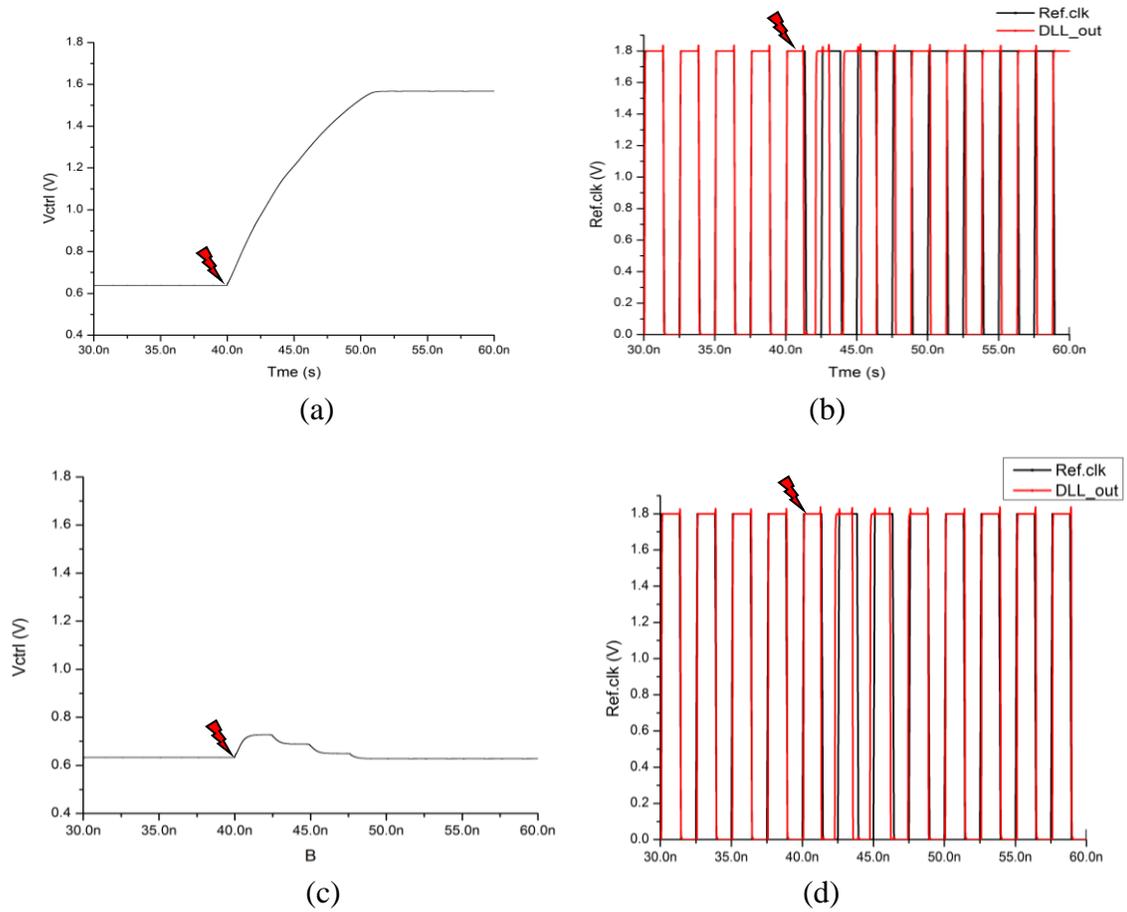
Fig. VI-9 SET response strike within the charge pump for an LET = 100 MeV-cm²/mg designed in IBM 180-nm PDK, a) for the current based charge pump, generating an inverted lock error, b) unhardened DLL output transient illustrating the inverted lock errors for operating frequency of 400MHz, c) acquisition curve of the hardened DLL implementing a voltage charge pump for an LET = 100 MeV-cm²/mg, d) corresponding transient response of the hardened DLL [73]. Ref.Clk is the DLL input reference clock and DLL_out is the output of the DLL and by extension the output of the VCDL

b)   ***Mitigate duty cycle errors in DLLs:***

The ratio of the average power to the peak pulse power is the duty cycle and represents the percentage of time the power is present. In the case of a square wave, the duty cycle is 50% since the pulses are present half the time. Duty cycle errors represent changes in logic high/low of

a clock signal pulse width, with phase displacement errors less than 2π. SET simulations show that duty cycle errors were generated in the phase detector and the hardened sub-circuit (i.e. RHBD VCDL and voltage CP) [1]. Simulations and experimental results indicate that errors usually do not exceed more than three clock cycles, independent of which DLL module is struck or which technology node is used, for LETs up to 100 MeV-cm^2/mg as shown in Figure VI-10, showing that duty-cycle errors are quickly flushed by the DLL. This observation can be explained by the perpetual comparison operation of the phase detector, which corrects the error before the CP can significantly affect the VCDL input control voltage value. This is because the response time of the PD is much faster than that of the DLL. This error is a lesser concern than the missing pulses or persistent inverted lock errors observed in analog DLLs.

If an application requires that the duty cycle errors be mitigated, a duty-cycle correction circuit can be implemented with the hardened analog mixed-signal DLL. Several duty-cycle correction circuits can be found in the literature [65, 66]. Some of these cited circuits are specifically designed for PLLs, like in T. Gawa's work [65], or even DLLs can be used to correct the duty cycle [68]. However, most of these solutions consume a large amount of chip area. Since simulation and experimental data show that duty cycle error errors are observed for a very short period at the output of the circuit, it is a designer's choice whether or not to implement a duty-cycle correction circuit that would, unfortunately, significantly increase the area and power consumption of the RHBD DLL. In another hand, modern systems do implement duty cycle correction circuits at the end of their distribution network, therefore it the most likely that the duty-cycle errors will be corrected by within the chip.
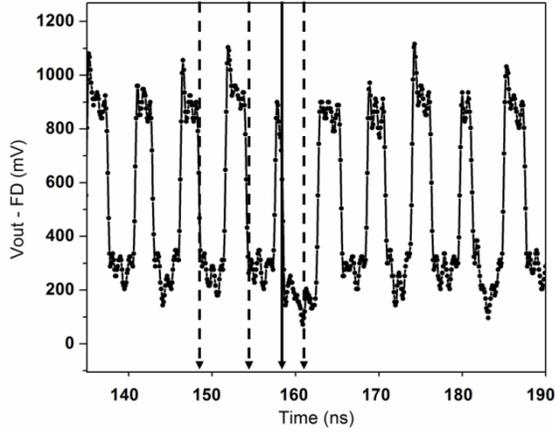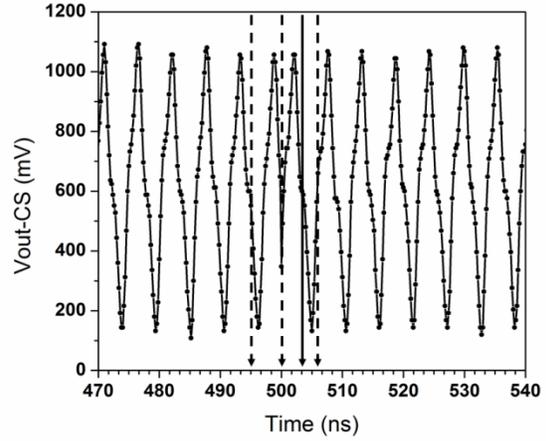
(a)   (b)
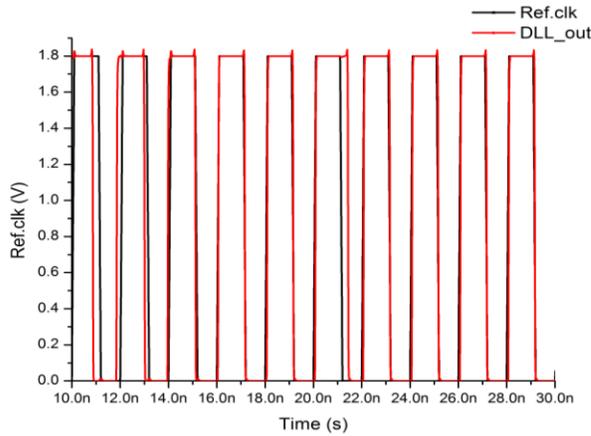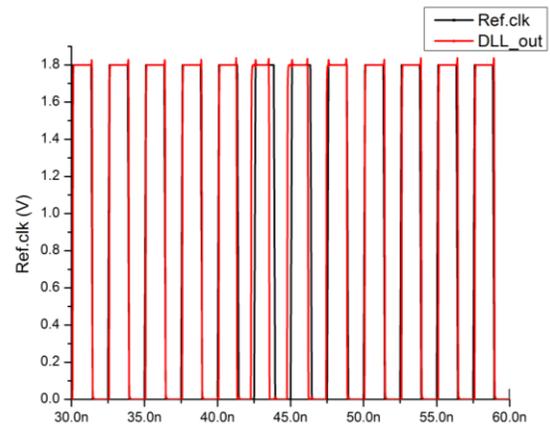
(c)   (d)

Fig. VI-10 DLL sub-circuit SET response in terms of phase displacement for different operating frequencies and technology node illustrating duty-cycle errors. a): for the unhardened complementary diff. pair VCDL designed in IBM 90-nm PDK for a laser energy = 3.2 nJ, b): 90-nm IBM unhardened single ended current starved VCDL for a laser energy = 3.2 nJ, c): for the 180 nm RHBD VCDL hit for an LET=100MeV-cm^2/mg, and d): the voltage charge implemented in 180-nm IBM PDK. Ref.Clk is the DLL input reference clock and DLL_out is the output of the DLL and by extension the output of the VCDL [73, 74].

64

2. Validation of the Hardened VCDL Solution: Implementation and Development of RHBD DLL Design & Simulation Results

A radiation hardened analog delay-locked loop (RHBD_DLL), illustrated in Figure VI-11, is developed and characterized using SET simulations [73]. The hardened mixed-signal DLL implements the hardening solutions for the missing pulses and the persistent inverted lock error. This means that a hardened voltage-controlled delay line (RHBD_VCDL) based on a complementary differential pair together with a hardened voltage based charge pump may be used to mitigate the worst cases of SET in the analog DLL. Hardening the phase detector module is not required, since the phase detector is not sensitive to SET because the module constantly compares the output of the DLL to the reference clock. As a consequence, the SET perturbation is quickly recovered by the next clock edge. However since this sub-circuit requires only a small area in the DLL, TMR may be used to harden the sub-circuit.
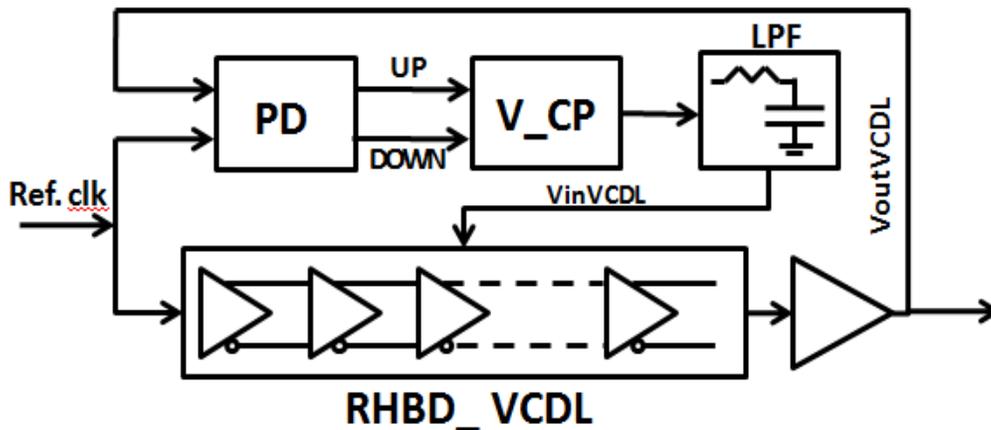
Fig. VI-11 RHBD DLL block diagram implementing the hardened complementary differential pair voltage-controlled delay line (RHBD_VCDL), a phase detector (PD), a voltage charge-pump (V_CP) and low-pass filter (LPF) [73].

Figure VI-12 compares the RHBD DLL to the unhardened DLL for worst-case SET response, for the 90 nm case at LET= 80 MeV-cm^2/mg and for the 180 nm case at LET= 100 MeV-cm^2/mg – both during lock mode. The simulation was carried out in CADENCE using both 3D TCAD ion-induced current profiles [59] and ISDE's bias-dependent single-event model [17] with strikes uniformly distributed over an entire clock period. When comparing Figure VI-12 (a, b, and c), we observe that all the transient responses generate maximum phase displacement errors under the grey region (representing missing pulses following ion strikes in the VCDL of the unhardened DLL). Moreover, all the missing pulses are mitigated and in some cases only duty-cycle errors remain. Also, the simulated SET response of the RHBD_DLL was very similar for different node technologies.

Finally, the radiation-hardened VCP's phase displacement error is also reduced and no persistent inverted lock-errors were observed for the worst-case conditions, i.e. at an LET of 100 MeV-cm^2/mg, as shown in Figure VI-12. Therefore, we can conclude that implementing the complementary differential pair RHBD voltage-controlled delay line and the voltage based charge pump, dramatically reduces the DLL sensitivity to single event transients, with small area and power penalties (when compared to TMR) and also improves the noise and power-supply rejection of the circuit. RHBD DLLs were fabricated for different operating frequencies in IBM 180-nm bulk technology node.
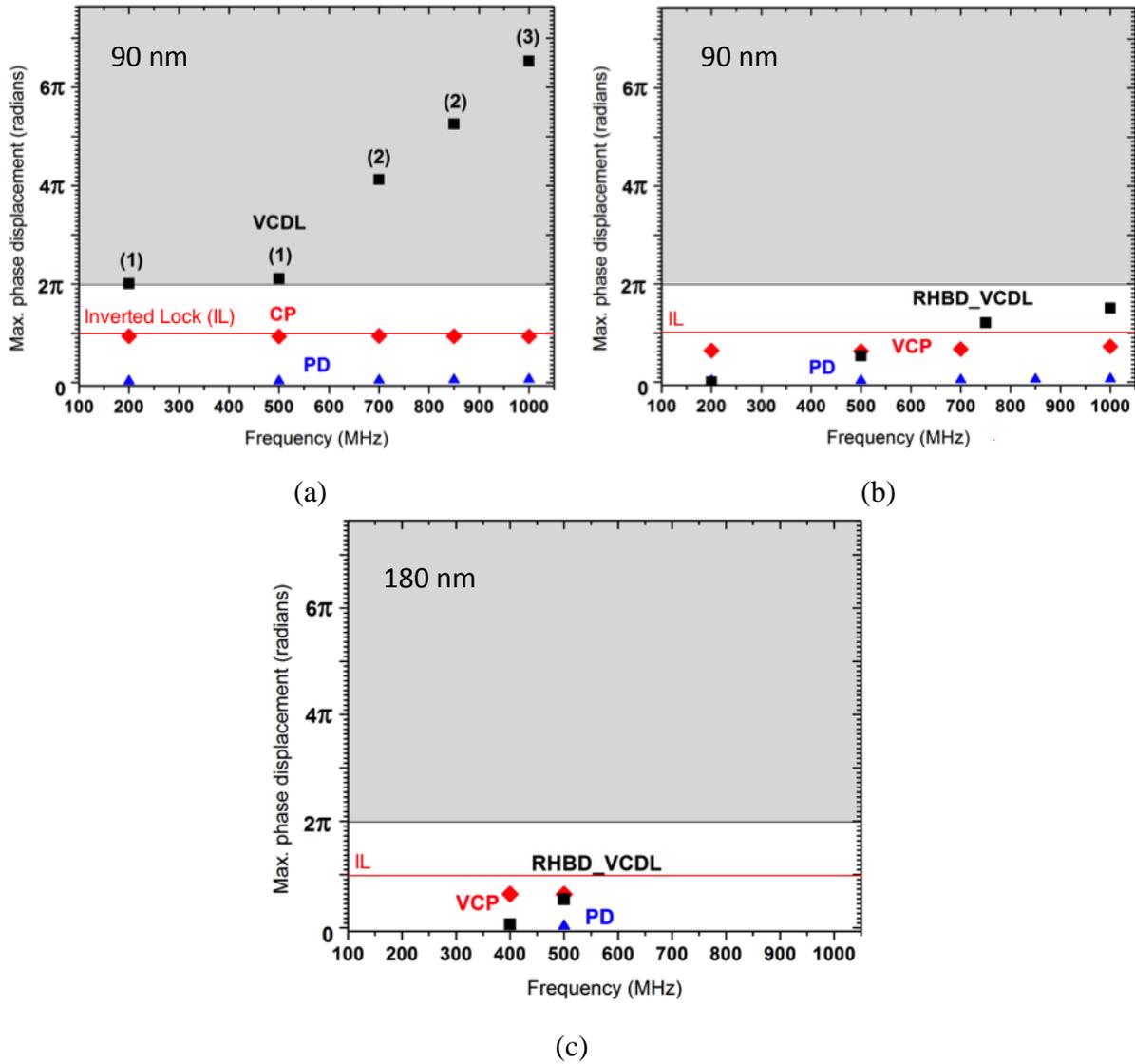
Fig. VI-12 RHBD DLL SET response in terms of phase displacement for operating frequencies between 200MHz and 1GHz, a) for the 90 nm unhardened DLLs at an LET =80 MeV-cm²/mg and b) for the 90nm RHBD DLL designs at an LET =80 MeV-cm²/mg, c) for the 180nm RHBD DLL designs at LET =100 MeV-cm²/mg. The grey region shows where missing pulses are observed, where no missing pulses are observed in the white region. The numbers in parenthesis are the maximum number of missing pulses observed after an ion strike. VCDL represent the voltage controlled delay line, CP the current charge pump, VCP the voltage charge pump, PD the phase detector and RHBD_VCDL is the hardened complementary differential pair VCDL [73].

a)   Simulation the proposed complementary differential pair based RHBD VCDL (missing pulses hardening technique)

To validate the proposed hardened VCDL design using a complementary differential pair hardened delay cell [2], and to verify the simulation results obtained using the ISDE bias-dependent current model [17], two RHBD_VCDLs, operating at 400MHz and 500MHz, were fabricated in IBM 180-nm technology. Both the VCDLs utilize the topology displayed in Figure VI-2. Figure VI-13 and Figure VI-14 show the final layout of these RHBD_VCDLs with the additional buffers. Figure VI-15 and Figure VI-16 show examples of simulated SEE response of the unhardened VCDL and the RHBD VCDL at 400 MHz. At the output of the DLLs (DLL 400 MHz outputs) and at the output of the chip (DLL output pos. (or neg.) 400 MHz chip buffer), no missing pulses were observed at the output of the 180 nm RHBD DLL, for an LET of 100 MeV-cm^2/mg.

To test the hardened VCDLs, custom printed circuit boards (PCB) were designed to configure these designs for radiation testing using the two-photon absorption (TPA) laser at the Naval Research Laboratory (NRL) [11, 22]. Single event transient experiments were conducted on these VCDLs, and the maximum phase displacement following SETs for operating frequencies will be measured at different laser energy levels (The energy levels can be approximated into LET values). The simulation results obtained using CADENCE ADE tools will be verified using the collected experimental data.
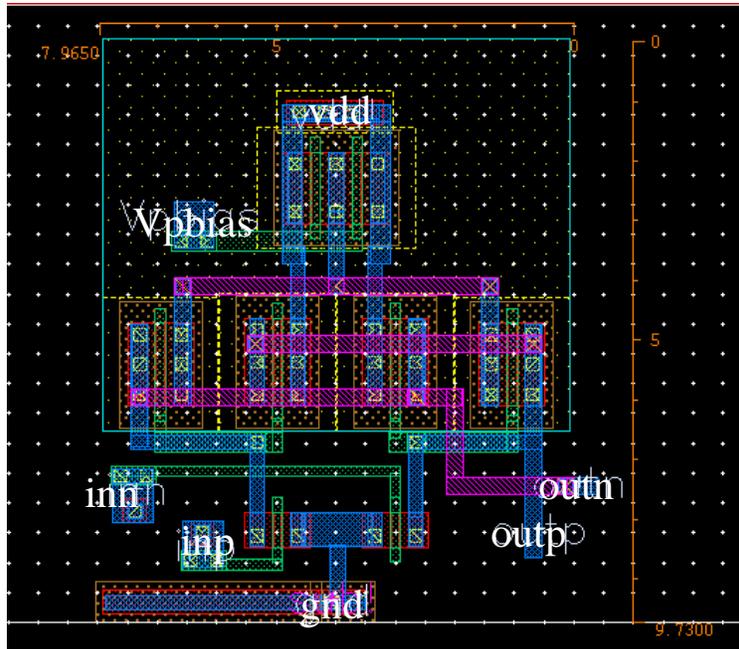
Fig. VI-13 Layout of the hardened VCDL delay cell designed in IBM 180-nm technology.
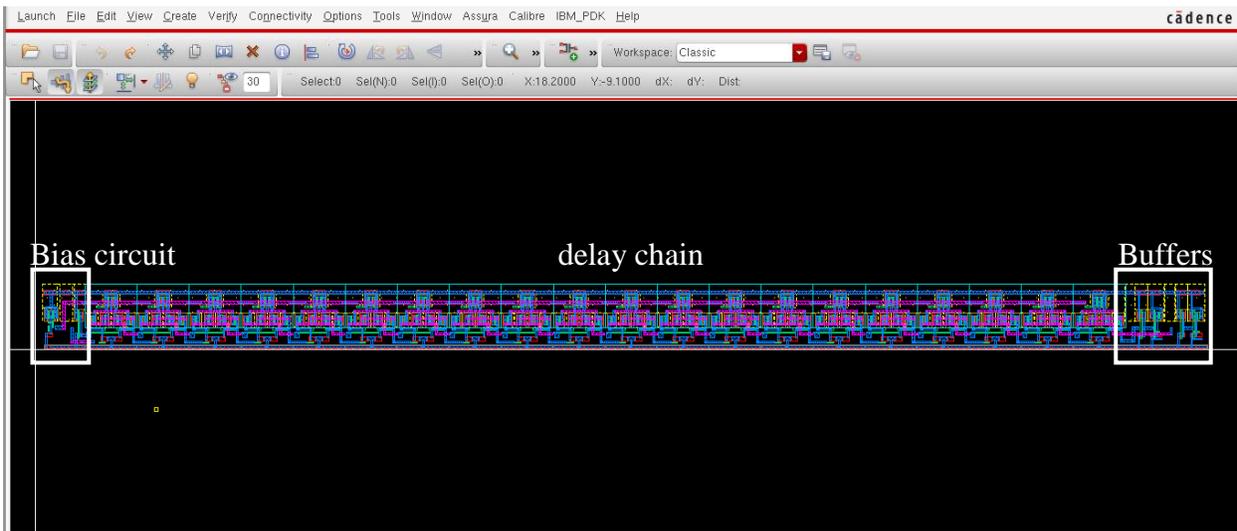
Fig. VI-14 Layout of the 500MHz RHBD voltage controlled delay line designed in IBM 180-nm technology. The VCDL is composed by a bias circuit, a delay chain with 20 delay elements and output buffers.
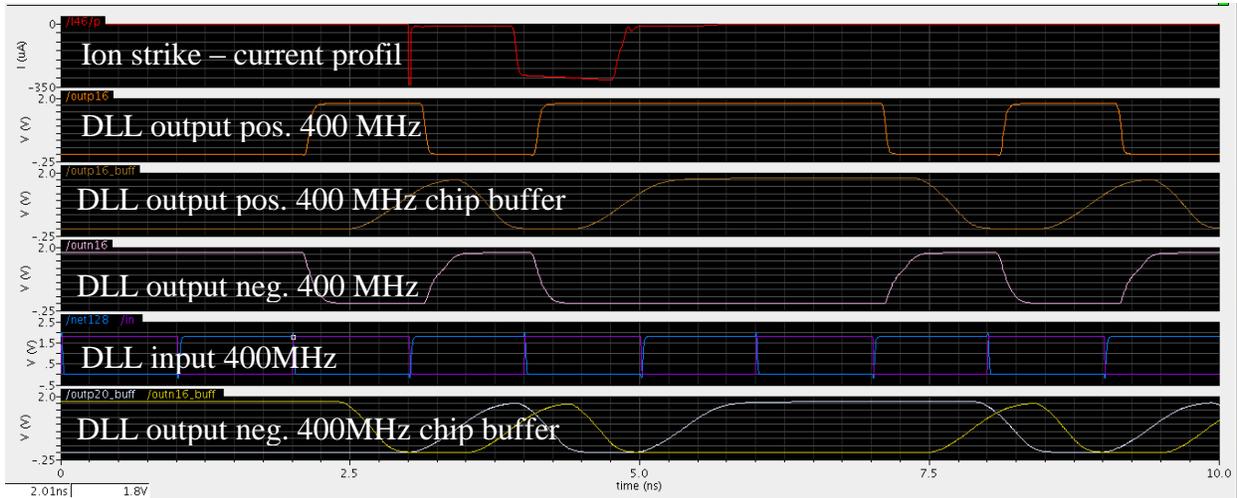
Fig. VI-15 Simulation of an ion strike generating a missing pulse in the 400MHz unhardened VCDL (DLL output pos. 400MHz chip buffer) at an LET=100 MeV-cm^2/mg.
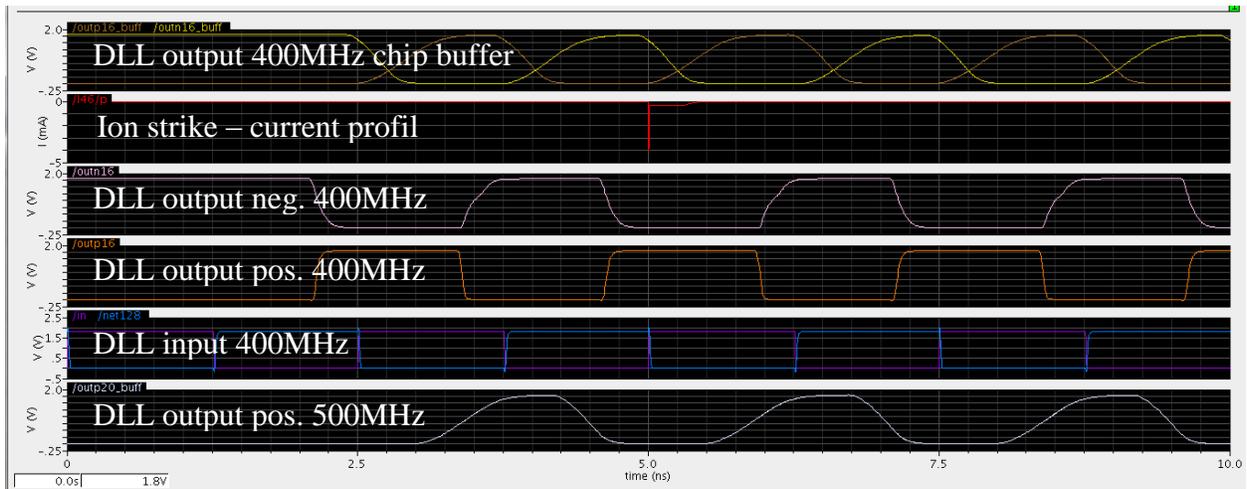
Fig. VI-16 Simulation of an ion strike within the 400MHz RHBD VCDL (DLL output pos. 400MHz chip buffer) at LET$_{max}$=100 MeV-cm^2/mg. No missing pulses observed at the outputs of the DLL or the chip.

b) *Simulation of the voltage based charge pump as a hardening solution to mitigate persistent*

inverted lock errors

In order to validate the inverted-lock error-mitigation technique, using the voltage-charge pump (c.f. section 1.a of this chapter), illustrated in Figure VI-17, and to prove that hardening techniques developed for PLLs [14, 15] could also be used to harden analog DLLs, even if the SET error signatures are different, voltage-charge pumps, fabricated in IBM 180-nm technology, were implemented in RHBD DLLs, operating at 400MHz and 500MHz.

SET simulations of the hardened voltage-charge pumps, presented in Figure VI-18, show that an ion strike within the charge pump forces the DLL out of lock for several nanoseconds, but no persistent inverted lock errors are observed for a maximum charge deposition of 1pC. In Fig VI-18, the SET response of the RHBD DLL following an ion strike within the voltage controlled delay line, at operating frequencies 500MHz and 400 MHz for the 180 nm hardened DLL design, is presented. In all cases, no missing pulses were recorded.

To test the voltage-charge pump, custom-printed circuit boards (PCB) were designed to implement these designs for radiation testing using the two-photon absorption laser at the Naval Research Laboratory (NRL) [11]. Single event transient experiments will be conducted on the voltage-charge pump. The VCDL control voltage perturbation measured in millivolts and the phase displacement following SEE for operating frequencies will be recorded at different laser energy levels (The energy level can be correlated to LET). Results of simulations obtained using CADENCE ADE tools will be verified with the collected experimental data.
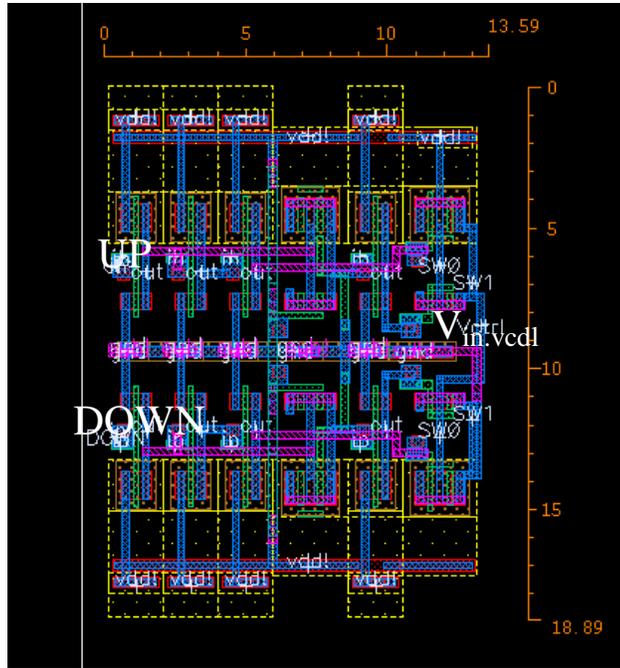
Fig. VI-17 Layout of the voltage charge pump designed in IBM 180-nm technology
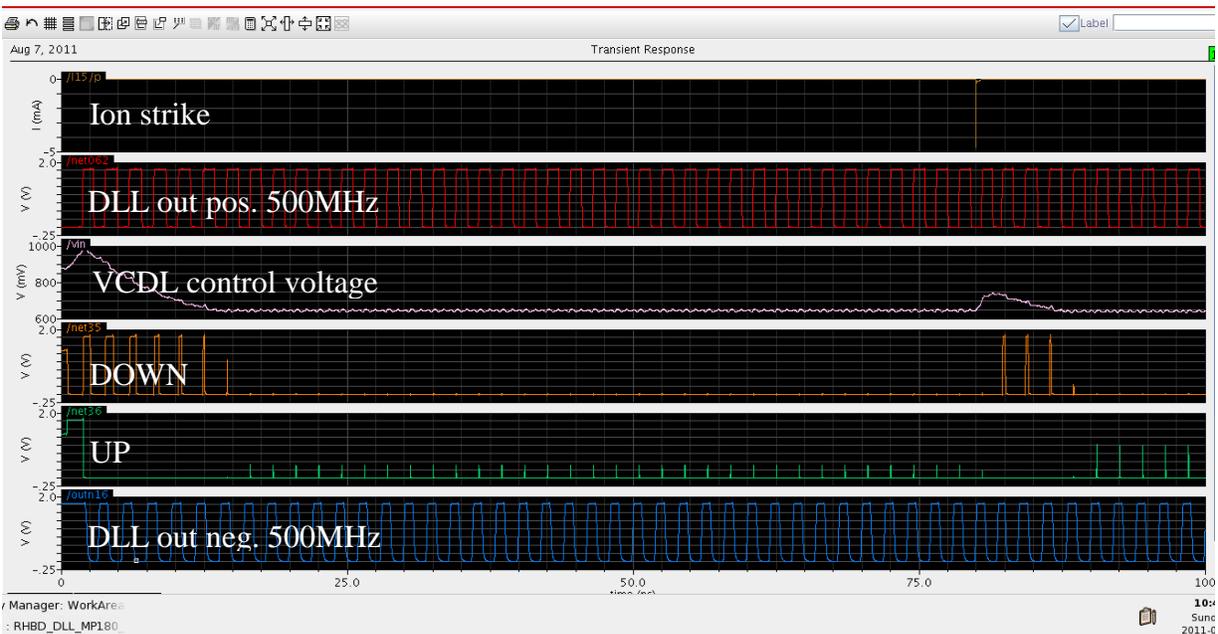
Fig. VI-18 Simulation of an ion strike (LET=100 MeV-cm²/mg) within the voltage charge pump of the 500MHz RHBD DLL (designed in IBM 180-nm technology), during lock mode. No missing pulses or inverted lock error are observed.

3. Implementation of the Diff. Pair VCDL Hardening Techniques in RHBD DLLs

Two RHBD VCDLs, implementing the RHBD VCDL (to mitigate the missing pulses) and the voltage-charge pump (to mitigate the persistent inverted lock error), operating at 400 MHz and 500 MHZ, were designed and fabricated in IBM 180-nm technology (test vehicle was funded by DRAPER). The choice of the DLL's operating frequencies was dictated by the operating range of commercial DLLs, i.e. ~ 100MHz to 400MHz. Fig VI-14 shows the block diagram of the RHBD DLLs with the required buffers at the output to match 5pF and 50Ω parasitics. Figure VI-19 shows the final layout of the two RHBD DLLs with all the sub-circuit and buffers.

To test the RHBD DLLs, a PCB test board was designed for radiation testing using the two-photon absorption (TPA) laser technique at NRL. Single event transient experiments were conducted on these DLL designs, the maximum phase displacement following SET and the VCDL control voltage perturbation following a strike in the charge pump (for different operating frequencies and at different laser energies) were captured, details regarding the laser experiment test bench and results will be provided in the next section. The simulations results obtained using CADENCE ADE tools, presented in Figure VI-20, VI-21 and VI-22, were verified with the processed experimental data.
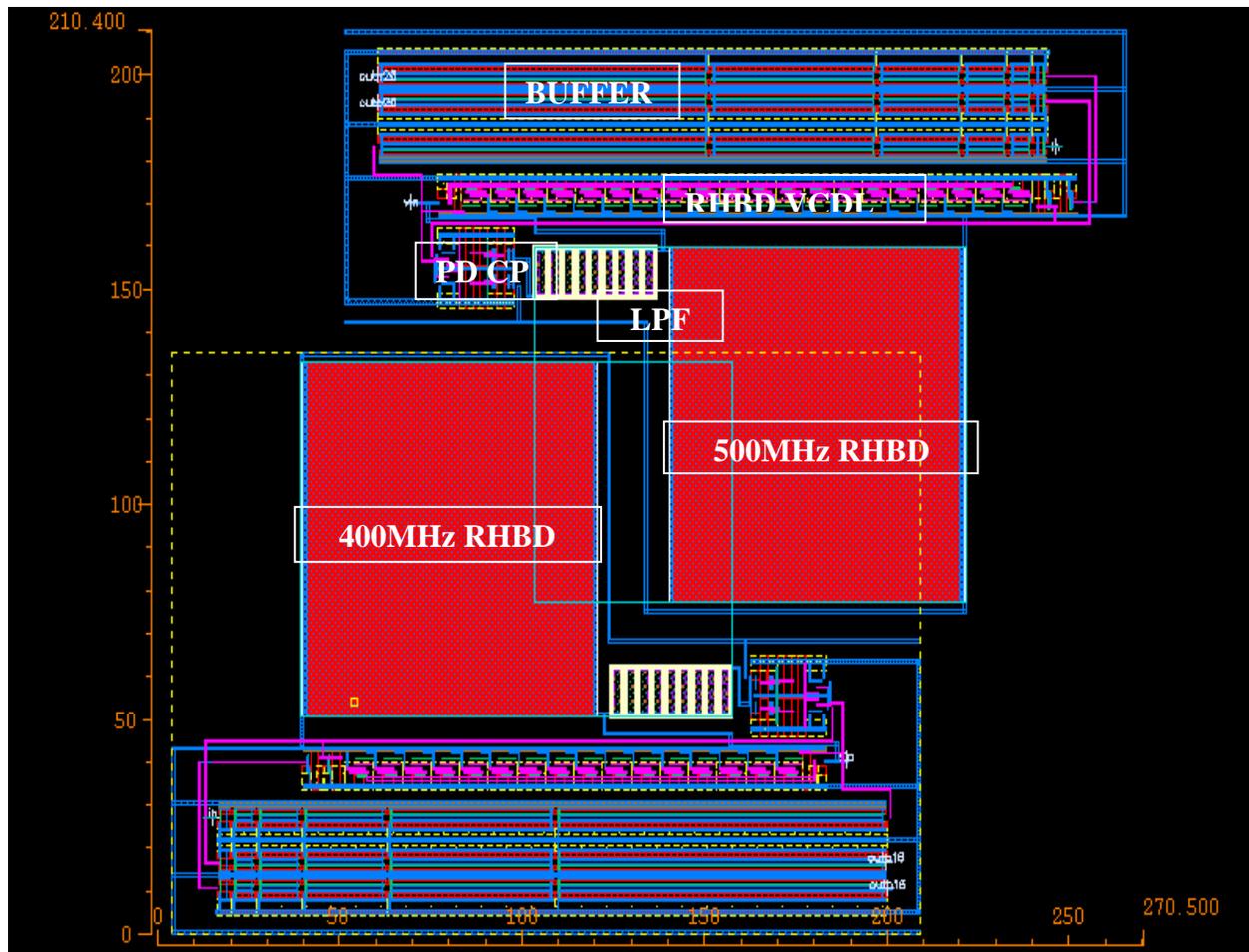
Fig. VI-19 Layout of the 400 and 500 MHz RHBD DLLs designed in IBM 180-nm technology (test vehicle funded by DRAPER).
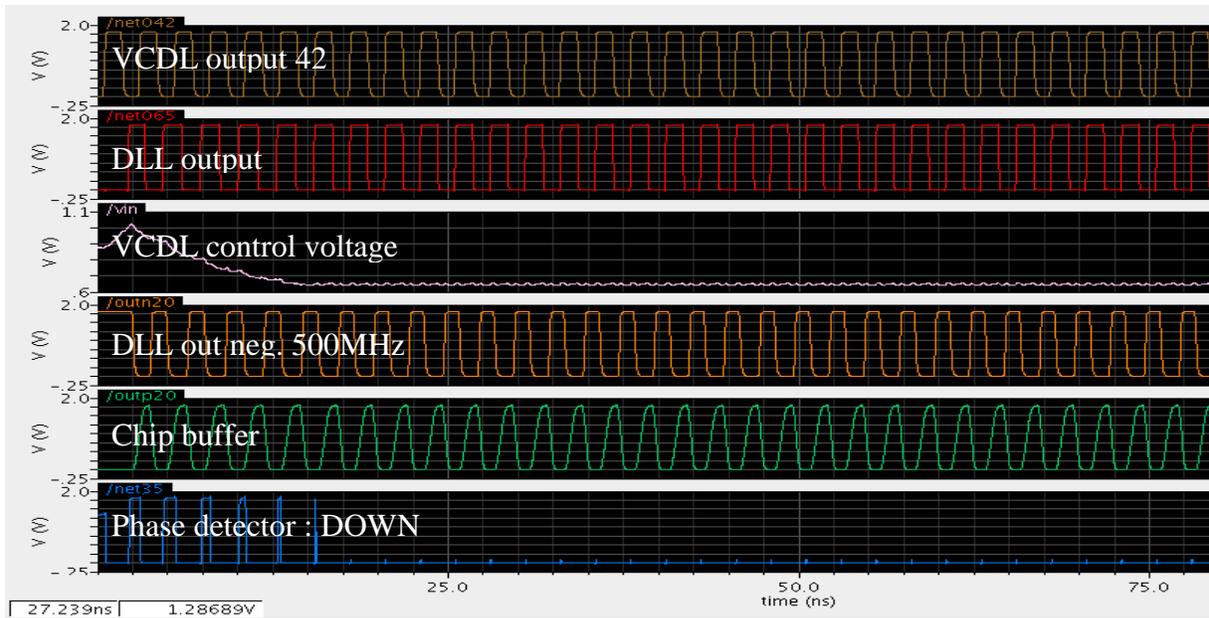
Fig. VI-20 Simulation of the electrical performances of the 500MHz RHBD DLL, designed in IBM 180-nm technology. In this example, the DLL locks within ~ 15ns.
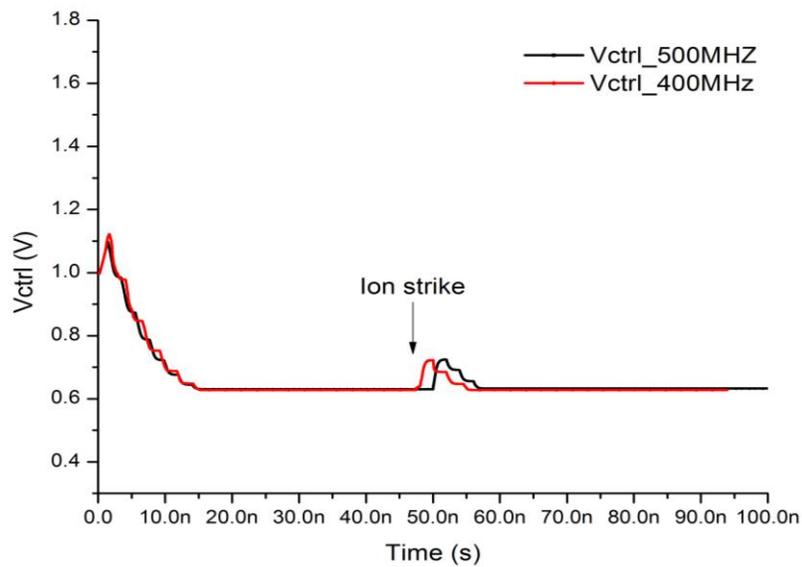
Fig. VI-21 Simulation of an ion strike (LETs of 100 MeV-cm^2/mg) within the voltage charge pump of the 400 and 500MHz RHBD DLLs, during lock mode. No inverted lock error observed.
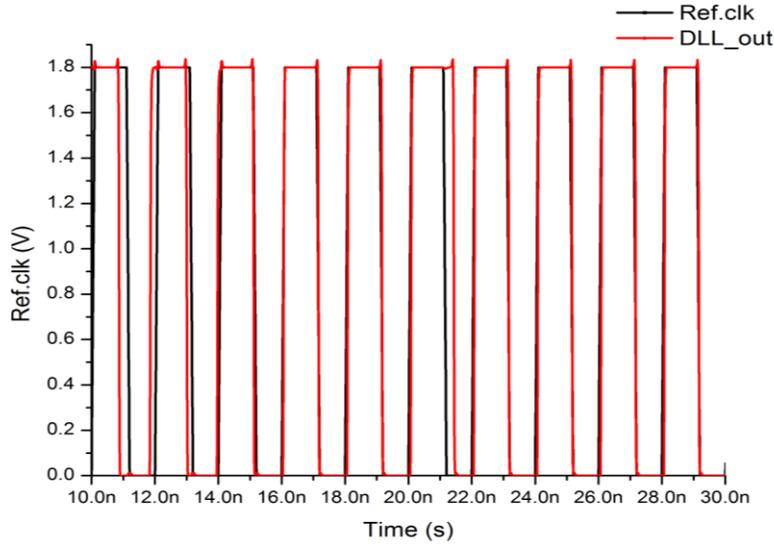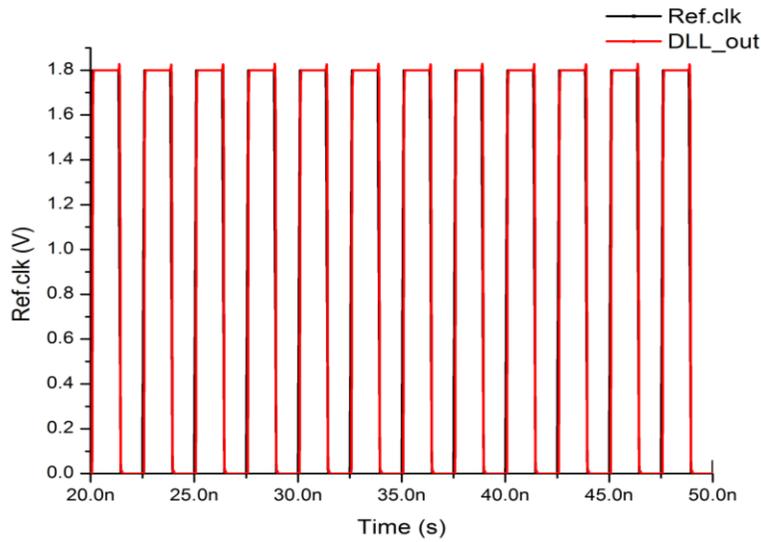
Fig. VI-22 SET response at transient level of the RHBD DLL following an ion strike within the voltage controlled delay line at operating frequencies 500MHz and 400 MHz, for the 180 nm hardened DLL design, for worst case LETs of 100 MeV-cm^2/mg, for a: 500MHz hardened DLL output, the strike occurs at $t_s = 21ns$, b: 400MHz hardened DLL output, the strike occurs at $t_s = 31ns$, no missing pulses were observed at the output of the DLL (DLL_out) [73].

4.   Experimental Validation of Hardened Complementary Diff. Pair VCDL Using Two Photon Absorption (TPA) Laser Testing

The SET response of the fabricated 180 nm RHBD VCDLs was characterized experimentally, using focused TPA laser testing at the Naval Research Laboratory (NRL) to induce single-event transients by charge deposition, as described in [11, 22].

The pulsed two-photon absorption laser (TPA) has been successfully applied to the evaluation of single event effects in a clock circuit like PLLs, VCDLs, analog-to-digital converters, etc. [14, 15, 74, 83]. This method of laser-induced carrier generation for SEE applications based on two-photon absorption (TPA) using femtosecond pulses at sub-bandgap optical wavelengths between 590 nm and 610 nm (typical NRL range), has also been used to successfully validate proposed hardening techniques [11, 14, 15]. This pulse length is a factor of 10 to 100 shorter than pulse length values used for single-photon absorption experiments. For experiment using two photons absorption, the laser wavelength is chosen so that the photon energy is less than the bandgap of the semiconductor material and under those conditions the generation of electron-hole pairs by single-photon absorption is absent, as described in [11, 22]. For sufficiently large laser intensities, electron-hole pairs are created in the semiconductor transistor by the simultaneous absorption of two photons. The required laser intensity is obtained by using highly focused (~1 micron in diameter) light pulses nominally 150 fs in duration [11, 22]. Since the carrier generation in the two photons absorption laser process is relative to the laser pulse irradiance in a quadratic manner, the carrier generation occurs mainly in the high-intensity focal region of the focused TPA laser pulse [11, 22]. The capability of the TPA laser experiment technique to investigate single event effect phenomena by irradiating simple and complex circuits through back of the wafer, eliminating interference from the metallization layers on the top surface that are prevalent in modern devices [22], and avoiding most of issues related to the test

of modern semiconductors using flip-chip, is the major advantage of this technique as explained by the authors in [22]. In addition, the pulsed TPA laser technique allows testing specific areas in complex circuits composed of different sub circuits, such as DLLs, PLLs, field programmable gate arrays (FPGAs), analog to digital converters (ADC), etc.

In this work, the TPA laser experiment technique is utilized to validate the SET response of a hardened differential pair VCDL for DLL circuits. The device under test (DUT) was mounted on a custom printed circuit board (PCB) with top and bottom ground shielding to minimize the effects of noise and wire capacitance on the DLL output signals. The top and bottom of the PCB are presented in Figure V-23. The device under test was mounted in a 40-pin DIP socket. In previous experiments [76], crystal oscillators were used to generate the DLL's reference clock, however the crystals are limited in operation frequency (less than 200MHz) and highly sensitive to power supply fluctuation or current spikes. Therefore in the latest experiment, the 400 MHz and 500 MHz input reference clocks were generated using an Agilent RF signal generator (50 Ω matching) and the I/O signals were monitored using a 2.5 GHz Tektronix oscilloscope. DLLs were both tested at 400 MHz and 500 MHz pre and post irradiation to avoid any erroneous data due to circuit malfunction. Phase lock was observed in both designs, even when the output signals of the DLLs were attenuated by the high noise of the PCB board and the parasitic capacitance of the bonding wires. The complete 180 nm RHBD DLL test bench is illustrated in Figure V-23 and Table VI-I summarizes the different design parameters of the 180 nm RHBD DLLs in terms of power, area, cycle to lock and jitter for each operating frequency

Table VI-1 Summary of the different parameters of the 180nm RHBD DLL such as power, silicon area and operating frequency.

PDK (tech.)	180-nm IBM	
F (MHz)	400	500
Power (mW)	25	20
Area (mm^2)	0.0312	0.0244
Lock (cycles)	~ 10	~10
Jitter (ps)	~ 32	~ 32
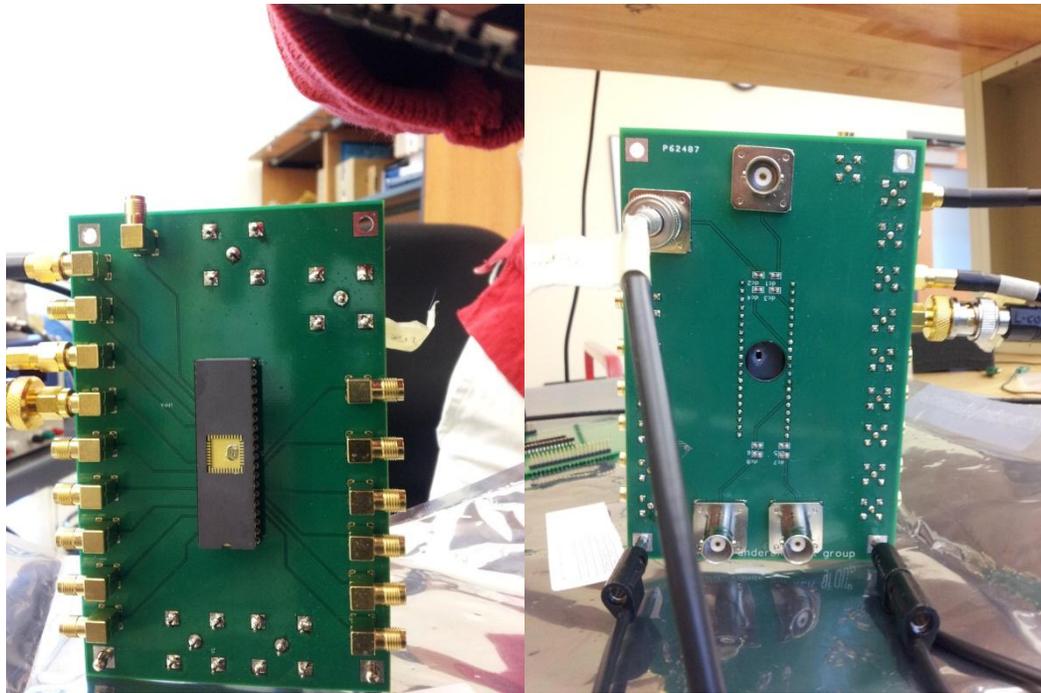
Fig. VI-23 Front and back view of the 180nm RHBD DLLs printed circuit board (PCB) with mounted device under test (DUT) used at NRL.

For testing at NRL, the test board with the DUT was mounted on a motorized x-y-z translation platform with 0.1 μm resolution. Optical pulses were focused through the wafer onto the front surface of the DUT with a 100X microscope objective, resulting in a near Gaussian beam profile with a typical diameter of approximately 1.4 μm. All experiments were performed at room temperature. The test setup is presented in Figure VI-24 and the DUT mounted for laser testing is presented in Figure VI-25. An infrared image of the 180 nm test chip was used to differentiate the four RHBD DLLs designs and to target the different sub-circuits. The DLLs acquired lock at approximately 400 mV, meaning that the reference clock of the DLL and the output of the circuit are in phase. Laser energies from 2 nJ to 5 nJ were applied to the delay cells, and the outputs of the complementary differential pair VCDLs were monitored and recorded on an oscilloscope. The maximum phase displacement errors were extracted for each operating frequency, in order to compare them to the simulations.
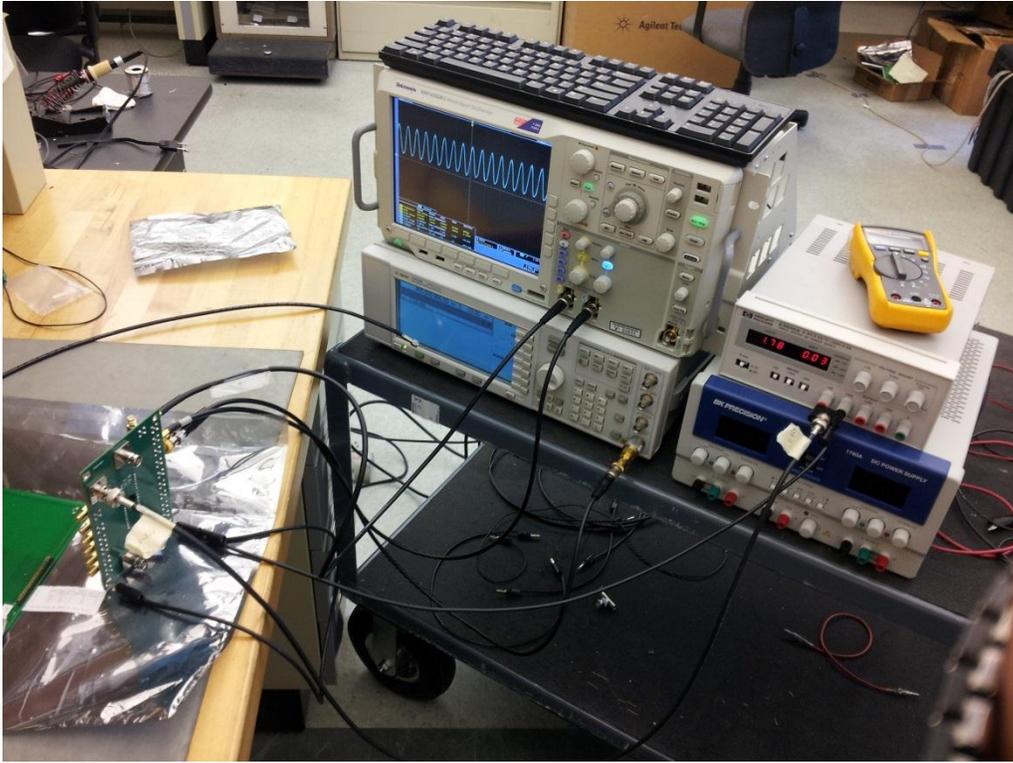
Fig. VI-24 180 nm RHBD DLLs test bench for pulsed laser experiments at NRL.
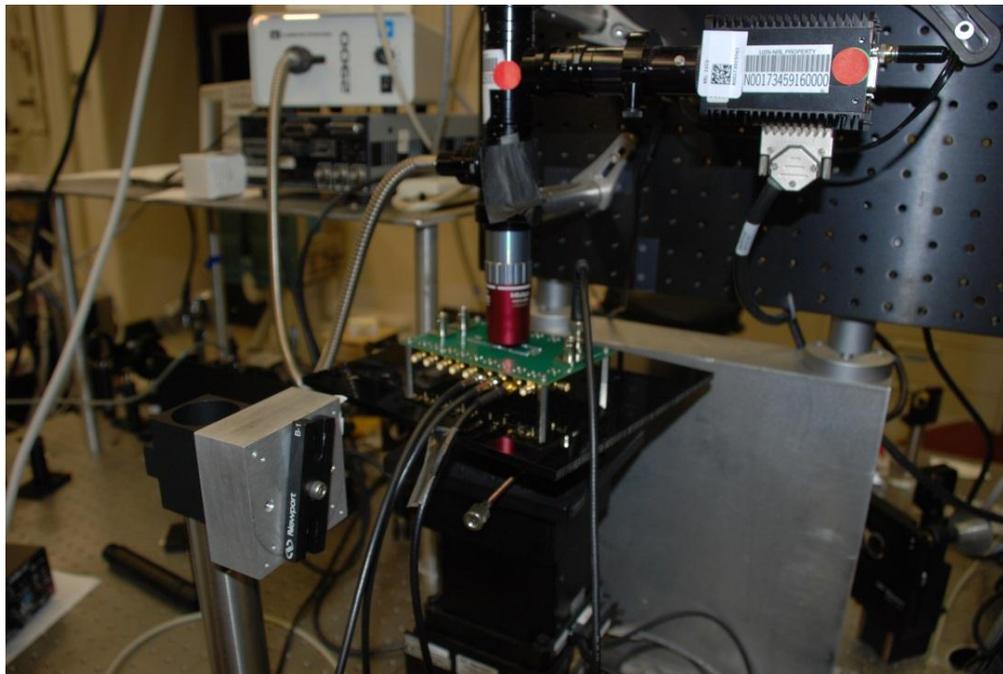
Fig. VI-25 180nm RHBD DLL device under test mounted on NRL bench for pulsed TPA laser experiment. The pulsed laser comes from the back of the test chip.

Figure VI-26 illustrates a single event transient at the output of the RHBD DLL following a strike in the hardened VCDL. The phase displacement error following an ion strike was measured by comparing the falling and rising edge of each VCDL output cycle to cycle, i.e. that each output clock period is compared to its neighboring periods, the single event detection script coded in python is available in APPENDIX-D. When an SET occurs, the rising or falling edge of the output voltage will be observed before its expected (normal) time, indicating a phase displacement error.
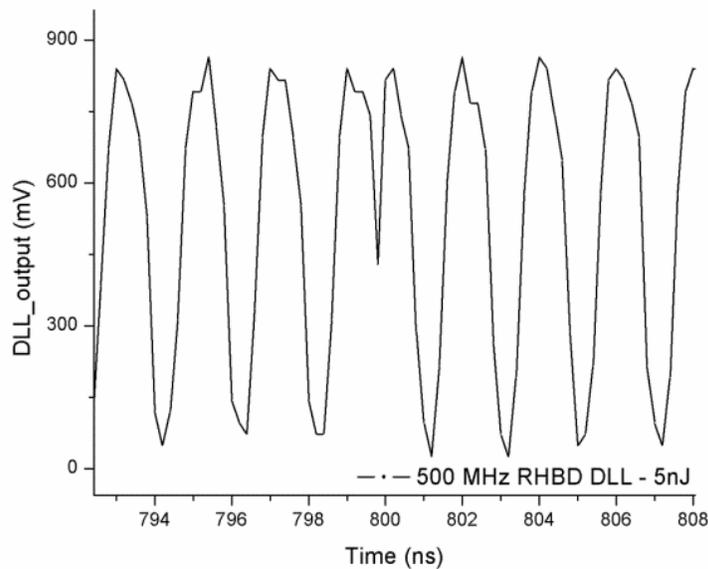
Fig. VI-26 Example of single event transient occurring in the IBM 180-nm RHBD DLL operating at 500 MHz, for laser energy of 5nJ.

The DLLs output transient phase displacement error response at 400 MHz and 500 MHz were then plotted as a function of the DLL's operating frequency in Figure VI-27. Experimental data show good correlation with the simulation data obtained by using the ISDE bias-dependent

current model (illustrated in Figure VI-12 (c)) and therefore proves that the hardened complementary differential pair presented in this work eliminates missing pulses in DLLs operating at high frequencies VI-27 (a) and in addition, data collected at NRL for the VCP show that no inverted lock errors were observed at the output of the DLL, as shown in Figure VI-27 (b). The data obtained using the TPA laser at NRL were combined with experimental and simulation data at different operating frequencies for different technology nodes to observe the trend of the proposed RHBD technique vs. technology scaling, as illustrated in Figure VI-28. In Figure VI-28, the SET response of the RHBD DLL for different process nodes and operating frequencies shows that below $F_{DLL} = 1$ GHz, simulations and experiments demonstrate that missing pulses are eliminated for technology nodes between 180 nm and 40 nm. Only duty cycle errors (increasing with operating frequency) remain in the DLL. However, simulations and looking at the experimental data trend show that missing pulses could still be observed at operating frequencies at 1 GHz and higher in 40 nm technology node designs. This increased SET sensitivity in low technology nodes has been observed in many digital circuits, typically due to reduced critical charge at reduced feature sizes. In the case of the hardened differential VCDL, the restoring currents cannot compensate for charge deposition at very high operating frequencies while maintaining normal VCDL performance requirements. Therefore we are proposing, in addition to the hardened complementary differential pair, a new hardening technique to mitigate missing pulses in DLLs that is immune to technology scaling at all operating frequencies.
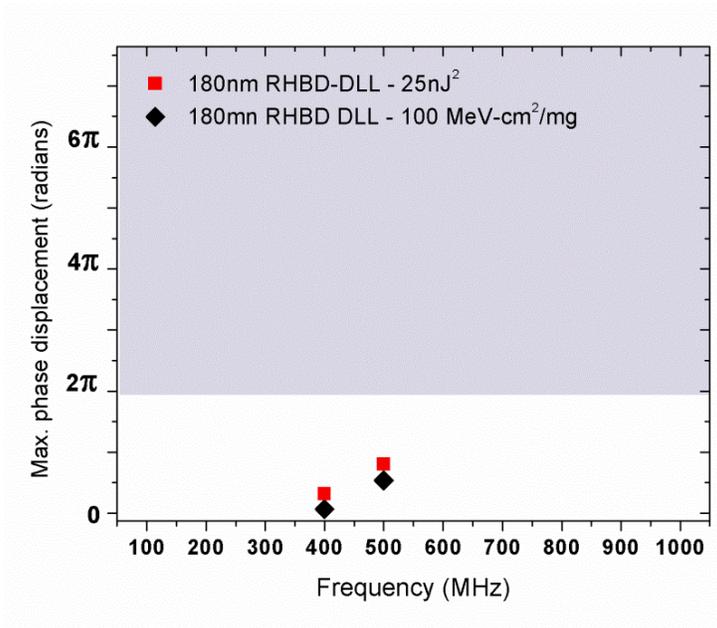
(a)
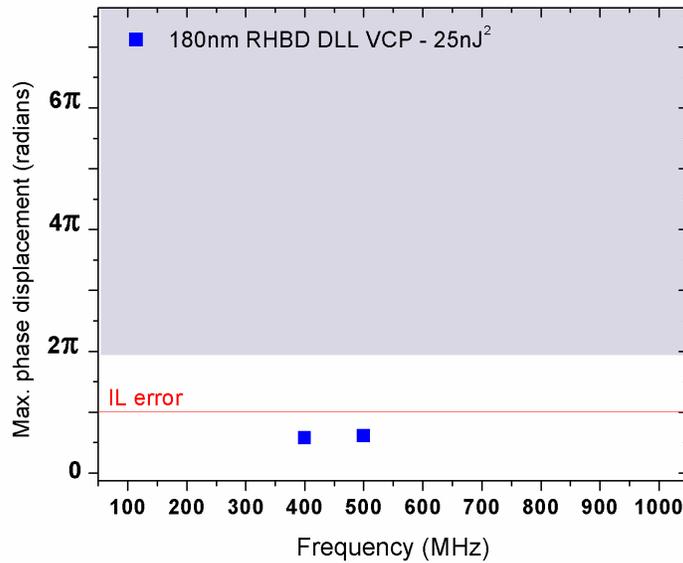
(b)

Fig. VI-27 Maximum phase displacement vs. operating frequency for a) 180 nm hardened complementary diff. pair VCDL simulations using the ISDE bias dependent model at an LET=100 MeV-cm²/mg and the TPA laser data at 25 nJ². No missing pulses were observed in the hardened circuit at 400 MHz and 500 MHz, b) for the VCP at 25 nJ² laser energy, no inverted lock (IL) errors were observed at 400 MHz or 500 MHz.
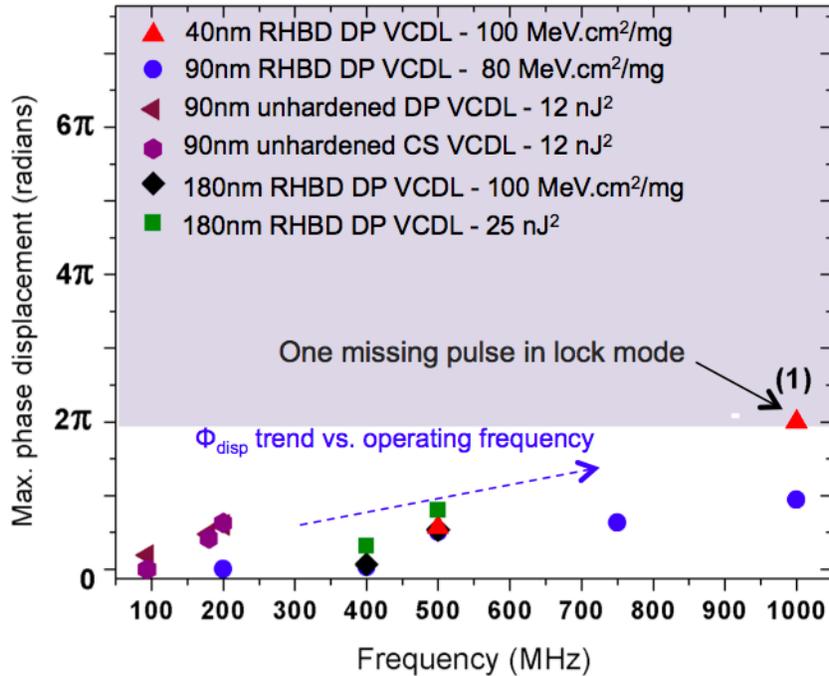
Fig. VI-28 Maximum phase displacement vs. operating frequency for VCDLs designed in 40 nm, 90 nm and 180 nm technology nodes. Data show an increase of the phase displacement with the operating frequency of the DLL. With technology scaling, missing pulses were observed in the 40 nm RHBD VCDL following an ion strike at operating frequency ≥ 1GHz.The number in parenthesis represents the number of missing pulse(s) [76]. The complementary differential pair VCDL (DP VCDL) will be preceded with the label RHBD to refers to the RHBD VCDL technique presented in this section. "CS VCDL" represents the single ended current starved VCDL [2].

C.   Hardening Technique 2: The Clock Error Correction Circuit

To evaluate the impact of technology scaling on the developed hardening solution, RHBD DLLs implementing the hardened VCDL based on complementary diff. pair topology (c.f. Figure VI-28 (a)) were designed in different technology nodes and simulated for various amounts of deposited charge (between 100 fC and 1 pC), using the ISDE bias-dependent current model [17].

Simulations in various technology nodes and experiments using two-photon absorption laser measurements at the NRL showed that missing pulses were mitigated in this hardened design. In Figure VI-28, the SET response of the RHBD DLL is shown as a function of different process nodes and different operating frequencies. At frequencies below 1 GHz, simulations and experiments showed that missing pulses are eliminated at technology nodes of 180 nm, 90 nm, and 40 nm. Only duty cycle errors (increasing with operating frequency) remain in the DLL. However, simulations show a missing pulse could still be observed at operating frequencies of 1 GHz and higher in the 40 nm designs. This increased SET sensitivity has been observed in many digital circuits, typically due to reduced critical charge at reduced feature sizes. In the case of the hardened differential VCDL, the restoring currents cannot compensate for charge deposition at very high operating frequencies while maintaining normal VCDL performance requirements. Consequently, a new error correction circuit was developed to mitigate these missing pulses at frequencies above 1 GHz for technology nodes of 40 nm and below. This circuit technique is described in the next section, and can easily be implemented across different technologies and operating frequencies.

(a)

(b)
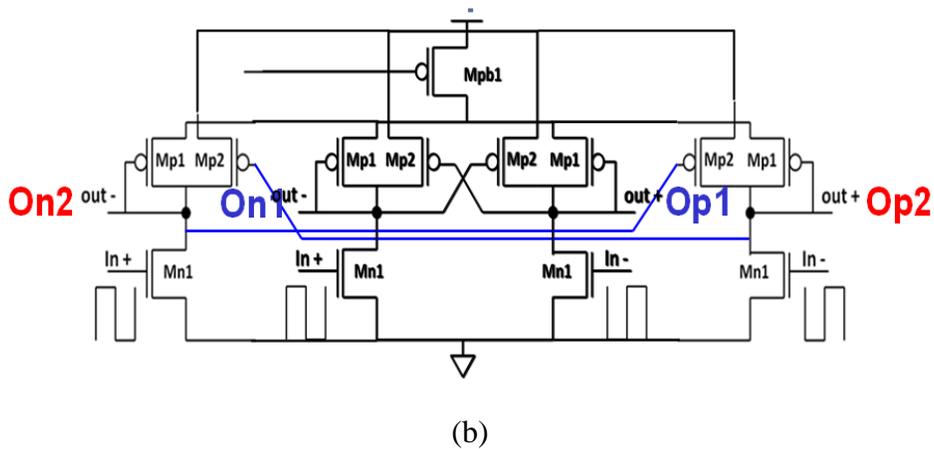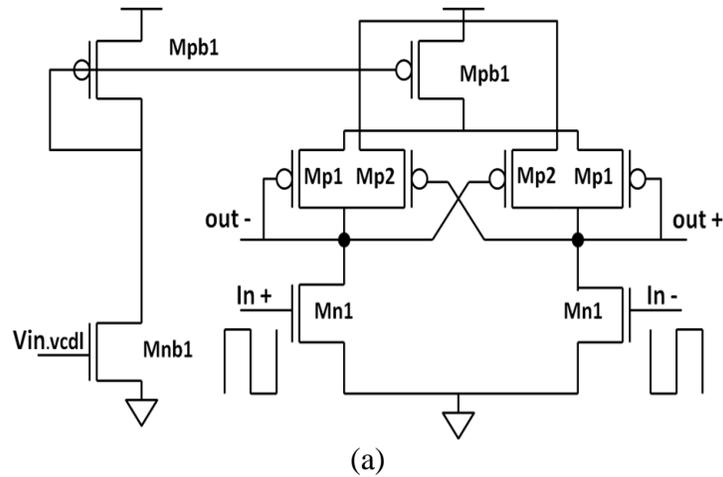
Fig. VI-29 a). Schematic of the complementary differential pair topology [2] and (b). The unhardened "peeled" differential delay cell [76].

1. The Error Correction Circuit

The proposed error correction circuit (ECC) uses combinational logic to mitigate the impact of technology scaling for DLLs operating above 1 GHz. This ECC requires two complementary VCDL outputs in the DLL (which can be achieved either with a peeled VCDL design or by using a dual redundant VCDL) combined with a missing-pulse detection and correction block as illustrated in Figure VI-30.
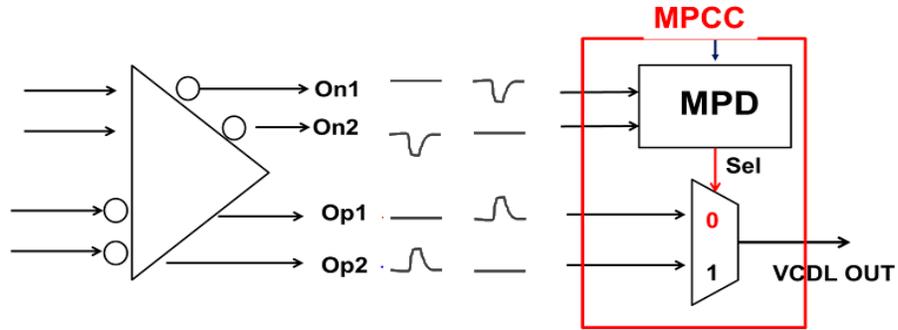
Fig. VI-30 Schematic block of the error correction circuit implemented with a peeled VCDL (on left). The technique relies on the complementary of the VCDL outputs to detect and eliminate missing pulses.

The "peeled" VCDL design is shown in Figure VI-29 (b), where the complementary differential pair (Figure VI-29 (a)) is split into two parallel differential paths. Each path has width-to-length transistor ratios half that of an unhardened VCDL. The peeled VCDL offers several performance advantages. In particular, it always generates an uncorrupted clock signal assuming that no charge sharing occurs between the peeled halves of the circuit. (Design consideration to avoid charge sharing is discussed on section V of this document). For example, if the output *On1* upsets, then the error will be mirrored to output *Op1*, but outputs *Op2* and *On2* will generate a clock signal without a single-event transient.

Since the width-to-length transistor ratios are halved as compared to an unhardened VCDL, the implementation of the peeled VCDL at the layout level requires the same chip area, which results in virtually no area penalty. Note, however, that a peeled VCDL layout is not required, as similar immunity could be obtained with dual VCDL circuits at the expense of somewhat greater area.

The proposed error correction circuit (Figure VI-31) takes advantage of these complementary differential VCDL paths with a guaranteed uncorrupted clock signal by using

XOR gates to compare outputs *On1* and *On2* to the DLL (or VCDL) reference clock and a 2:1 MUX to propagate the uncorrupted VCDL output according to the truth table of Table I. If there are no upsets in *On1* or *On2*, the output of both XOR gates will be logic "0", the MUX select (*Sel*) will be set to logic "0", and the ECC will output signal *Op1*. (Note that a phase shift between the reference clock and the *On1* and *On2* outputs will not affect *Sel*, as both XOR outputs switching to "1" and back to "0" will only swap the AND gate inputs.) If an upset occurs in *Op1*, and is propagated to *On1*, signals *On2* and *Op2* remain uncorrupted, and *Sel* switches to a '1' logic level. In that case the multiplexer will output the signal *Op2*. If upsets occur instead in the signal *Op2* (and therefore *On2*), *Sel* will switch to a '0' logic level, and the MUX will output signal *Op1*. The truth table of the ECC is presented in Table VI-2. In any case, the output of the VCDL remains uncorrupted. The next section will present simulation and experimental result to validate this ECC technique.
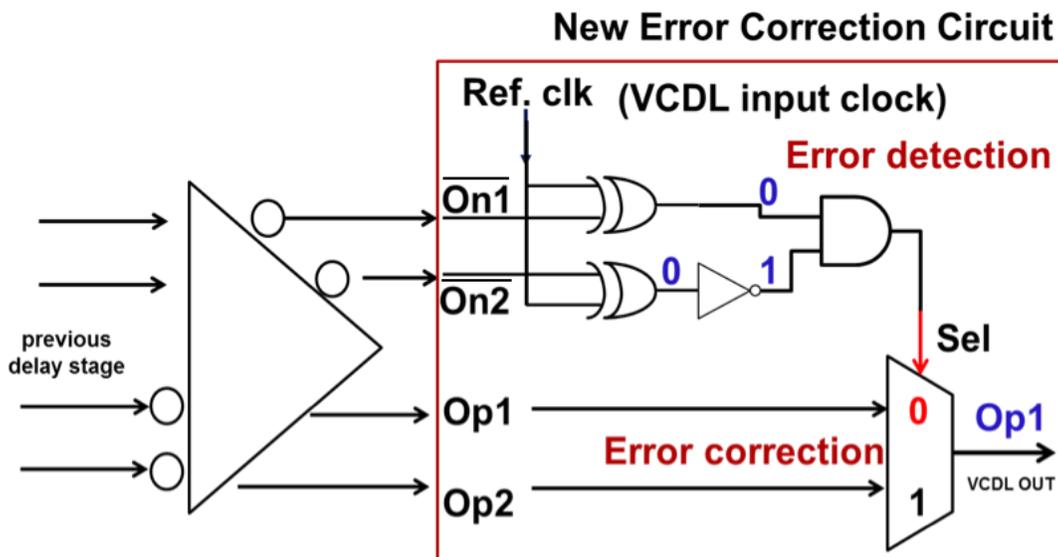
Fig. VI-31 Schematic of the proposed error correction circuit implemented with the peeled VCDL. The ECC consists of a missing-pulse detection block and an error correction block.

Table VI-2 Truth table of the ECC. The signal Sel is used to control the 2:1 MUX when missing pulses are detected. If no missing pulses are detected, the input signal Sel. is set by default to '0' logic. Even if the probability of Op1 and Op2 upsetting at the same time both is low. Row 5 of the table where both signals are at '1' logic (in Italic) represents the case where charge sharing occurs in the VCDL

Op1	Op2	Sel.	DLL_out
0	0	0	Op1
0	1	0	Op1
1	0	1	Op2
1	1	1	Op2

2. Experimental and Simulation Setup

a) Simulation setup

To simulate the error correction circuit, the ECC was implemented in the UMC 40-nm and IBM 90-nm CMOS process design kits (PDKs), and simulated at frequencies between 500 MHz and 1 GHz. Both circuits were simulated using ion-induced current profiles, illustrated in Figure VI-32, obtained from the ISDE bias-dependent current model generator [17], with simulated linear energy transfer (LET) values between 5 MeV-cm^2/mg and 40 MeV-cm^2/mg.

For both circuits, the VCDL operating frequency ranges were adjusted by changing the number of identical delay cells. Ion strikes were varied over the entire clock period at every peeled or dual VCDL sub-circuit node and for each ECC stage and internal node, representing more than 100,000 simulations. After all results were recorded, the worst-case SET responses were identified and plotted as a function of the phase displacement error. The simulation results were then compared to those of the unhardened DLL and RHBD differential pair based DLL.
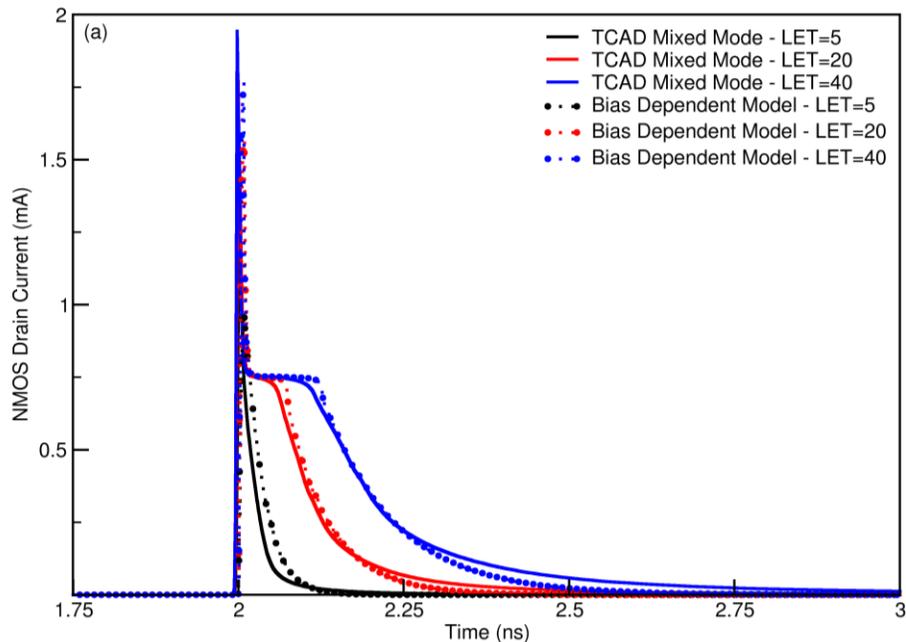
Fig. VI-32 Example of Ion induced current profiles generated by the ISDE bias dependent current model for LET values between 5 and 40 MeV.cm²/mg [17].

b)   Experimental (emulation) setup

To further validate the effectiveness of this new error correction circuit, the ECC block was experimentally validated using a Xilinx Virtex 5 field programmable gate array (FPGA) implemented on a GENESYS board, as shown in Figure VI-33 [78]. The ECC was implemented with an FPGA-based fault injection circuit using pseudo-SET pulse injection to emulate high-energy ion strikes generating missing pulses.

Fig. VI-33 Xilinx Virtex-5 FPGA used to implement the ECC and the fault injection block. The FPGA is mounted on a GENESYS Diligent development board.

The fault injection block of Figure VI-34 is composed of a missing pulse generator circuit, with errors injected into signals *Op1* or *Op2* using 2:1 MUXs activated by an "Error injection select", forcing *On1/Op1* or *On2/Op2* to oscillate for one or multiple clock cycles. The reference clock, sampling clock, *Op1,* and *Op2* are all synchronized using an on-chip delay locked loop called the Xilinx DLL/DCM primitive [79]. Therefore *Op1*, *Op2* and the reference input clock are all in phase, just as they would be at the output of an analog DLL. The register-transfer level (RTL) schematic of the full experimental setup, implementing the Xilinx DLL/DCM, is illustrated in Figure VI-35.

The output *DLL_out* and the inputs of the ECC were then captured and analyzed using the Xilinx ChipScope Pro Analyzer tool [80]. In order to capture accurately each ECC I/O signal, a sampling clock equal to twice the reference clock of the DLL was implemented, to avoid any aliasing in the results. In addition, onboard flag LEDs were used to count the number of missing pulses and also monitor the signal *Sel*, XOR outputs, and DLL output (V_{out}) of the error correction circuit. After all results were recorded using ChipScope, the experimental results were then compared to simulation results for the same number of missing pulses, in order to compare and validate the behavior of the ECC, the VHDL code is available in APPENDIX-C.
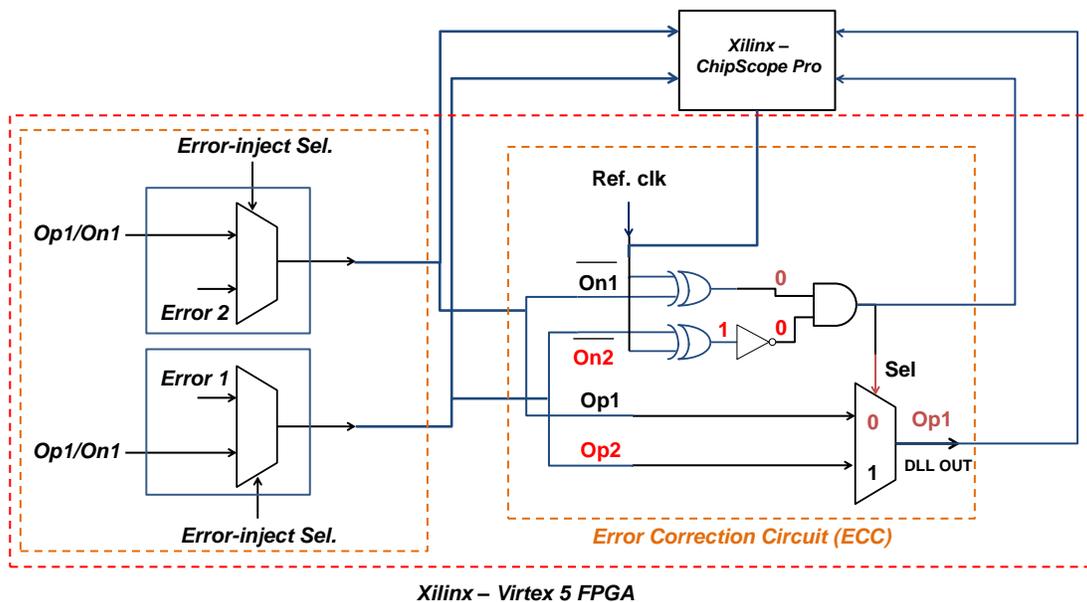
Fig. VI-34 Schematic of the ECC test bench implemented with the missing-pulse injection block and the DCM/DLL in Xilinx Virtex 5 FPGA. The outputs and inputs of the circuit are monitored using the Xilinx ChipScope Pro Analyzer tool [76].
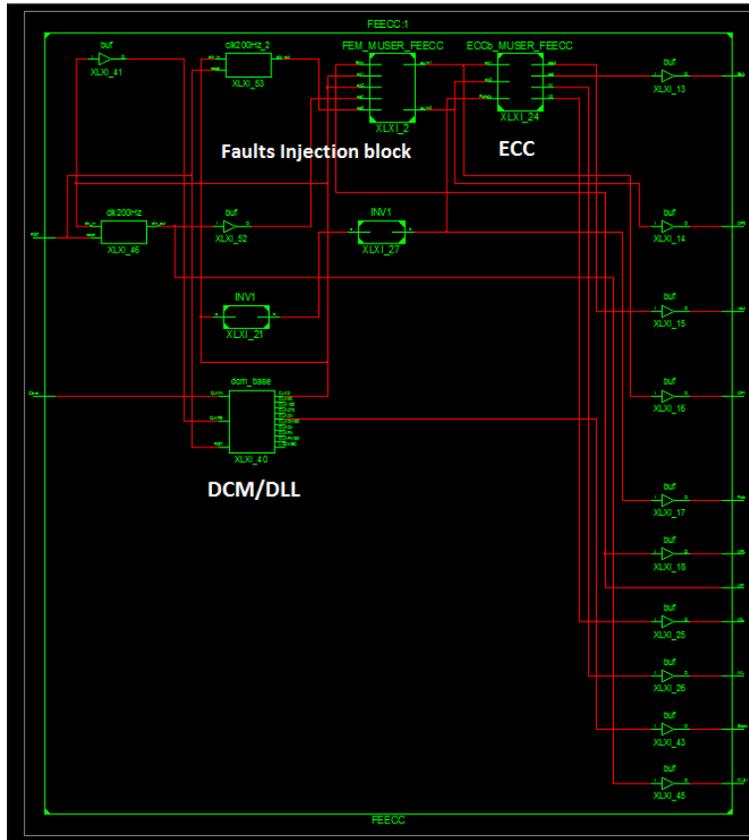
Fig. VI-35 RTL schematic of the ECC implemented with the missing-pulse injection block and the DCM/DLL. The outputs and inputs of the circuit are monitored using the Xilinx ChipScope Pro Analyzer tool.

3. Simulation Results of the ECC

The ECC was simulated at operating frequencies between 500 MHz and 1 GHz with LETs between 10 MeV-cm^2/mg and 100 MeV-cm^2/mg, respectively. This type of simulation approach has been proven effective via the experimental validation of the SET responses of mixed-signal circuits such as digital PLLs [14, 15, 75, 83] and the comparison of RHBD DLL designs implementing the hardened complementary diff. pair VCDL.

94
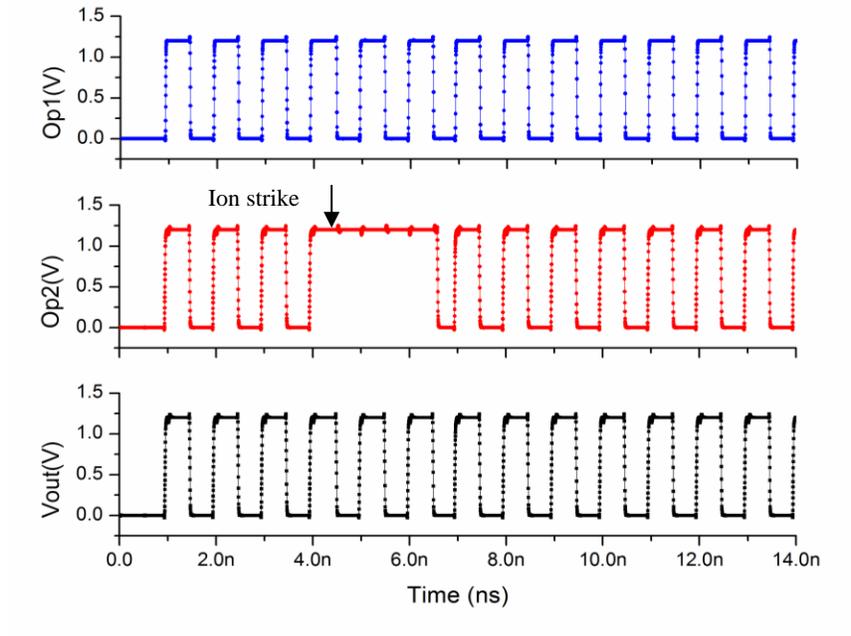
(a)
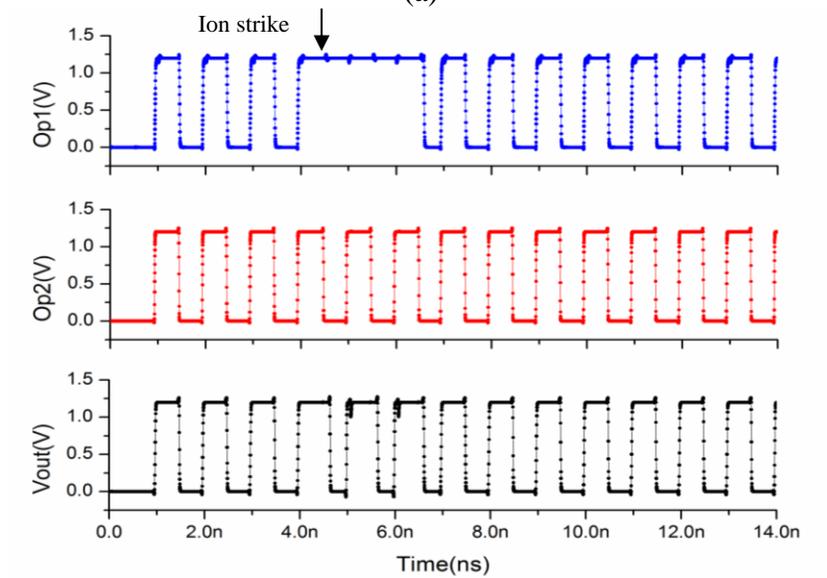
(b)

Fig. VI-36 Worst case SET response (in terms of missing pulses and phase error) generated by peeled VCDL (outputs Op1, Op2) and the ECC (Vout) at LET = 100 Mev-cm^2/mg for a) a hit in Op2/On2, b) a hit in Op1/On1, Only 15% duty cycle variations are observed following the switching operation of the MUX. In all cases, no missing pulses were observed at the output of the ECC or DLL.

Figure VI-36 illustrates the worst-case phase displacement error generated by the ECC when implemented with a peeled VCDL layout at LET = 100 MeV-cm²/mg. The transient responses of the outputs *Op1* and *Op2* and the outputs of the DLLs (*Vout*) are compared. Figure VI-36 (a) represents the case where *Op2* is perturbed and two missing pulses are generated. However, no missing pulses are observed at the DLL output *Vout*, and therefore no errors were propagated into the clock system. Figure VI-36 (b) represents the case where *Op1* is perturbed and two missing pulses are generated. As in the first case, the missing pulses are filtered at the output of the DLL, *Vout*. Note that a duty cycle variation is observed for a few clock periods, caused primarily by pulse skew following the switching operation of the MUX. By using transmission gate based XOR gates, this ECC design penalty was minimized to a 15% duty cycle variation.

Simulations at LET = 100 MeV-cm²/mg show that ion strikes have a very small impact on the error correction circuit itself. Any strike on the XOR can only temporarily 'flip' the logic state of the signal *Sel*, forcing the ECC to toggle between two identical MUX inputs, so the MUX output signal remains the same with no upset propagated into the clock network. A strike in the MUX has a 50% chance to be propagated to the output of the ECC depending on the internal MUX signal path. In the worst case, the ion strike will generated a duty cycle error for less than a clock cycle in the output signal *Vout* of the ECC (c.f. Figure VI-37), equivalent to a phase displacement of 1.4 radians at 1 GHz in a 40 nm process. Considering jitters, one of the disadvantages of XOR gates (phase detectors) is that the voltage output signal varies with the duty cycle of the input signals. Thus the XOR gates used in the error detection block may increase jitter at the output of the DLL that could be fed back into the circuit. As long as the phase detector does not capture jitter, the stability of the DLL is ensured. In the 40 nm design, the dead zone of the phase detector is approximately 80 ps, therefore jitter (less than 40 ps with the additional ECC block) were not captured by the PD and the stability of the DLL in lock state was

not compromised. Guidelines to improve the jitter response of the circuit would be to use AND-OR-Invert (AOI) XOR gates with less logic and fast operation, decrease the charge pump current (i.e. decrease the gain of the phase detector) [18] or use a toggle in the ECC block (this solution may increase the ECC block logic complexity).

The additional area and power required by the ECC is not significant, particularly when compared to the area and power requirements of the other DLL sub-circuits. Therefore, this new error correction technique provides an excellent tradeoff between SET response and area-power penalty. The area penalty is less than 2% for the entire DLL, as compared to triple modular redundancy of the VCDL with an area increase of approximately 30%.
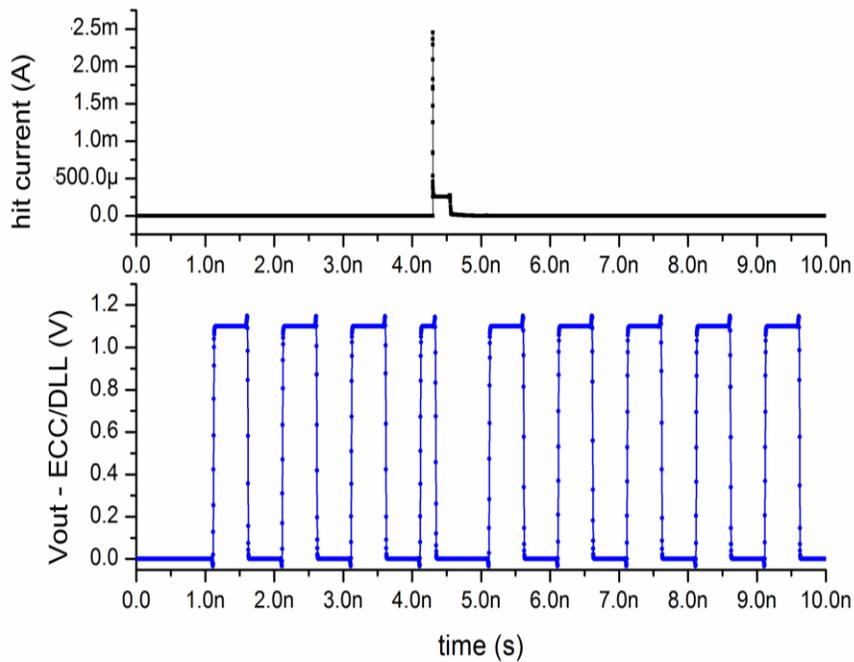
Fig. VI-37 Worst case SET following a strike in the 2:1 MUX of the ECC block, at LET =100 MeV-cm^2/mg. One clock cycle duty-cycle error was recorded at the output of the DLL (Vout – ECC/DLL) for strikes in the error detection block.

Figure VI-38 compares the SET response of a 1 GHz DLL with the ECC (implemented in 180 nm, 90 nm and 40 nm technology nodes) with the original hardened complementary diff. pair VCDL [2]. The missing pulses generated by an ion strike within the DLL implementing a hardened complementary diff. pair VCDL (RHBD_DLL) or the DLL implementing the ECC hardening technique (ECC_DLL) are significantly reduced or mitigated when compared to the SE response of the DLL using an unhardened VCDL (unhardened_DLL). Simulations at high LETs show that the ECC_DLL also mitigates missing pulses at the 40 nm technology node, with even worst-case simulations showing only duty cycle errors.

These results lead to the conclusion that both hardening techniques are effective for mitigating missing pulses at frequencies below 1 GHz in 90 nm and 180 nm technologies. However, the duty cycle error of the ECC hardened DLL remains approximately constant as

frequency increases, with no missing pulse observed at 1 GHz, demonstrating that this new hardening technique is both effective and scalable.
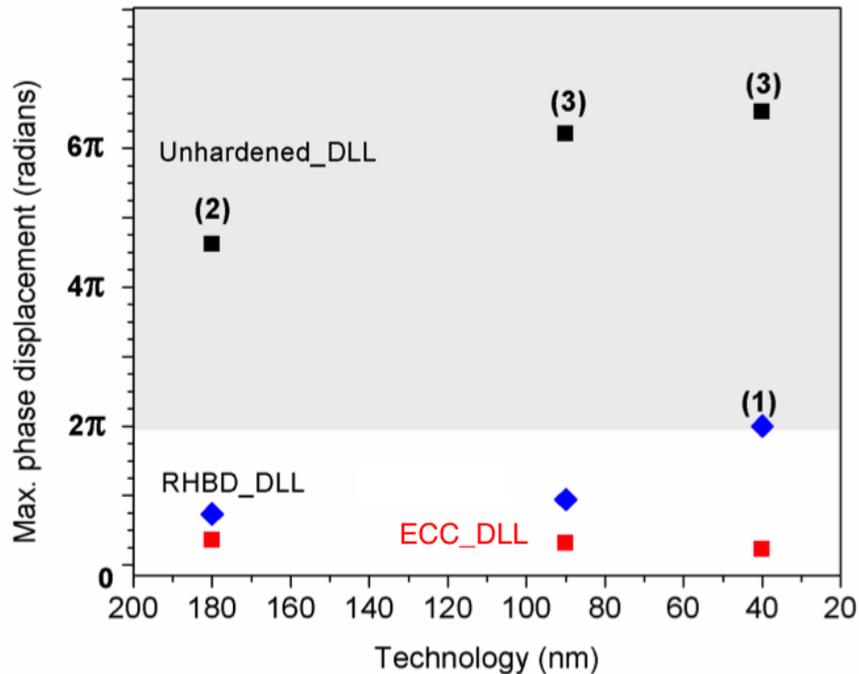
Fig. VI-38 1 GHz DLL maximum phase error vs. technology node for worst-case LET = 100 MeV.cm²/mg within the unhardened DLL implementing the current starved VCDL, the DLL implementing the hardened compl. diff. pair VCDL (RHBD_DLL) and the hardened DLL implementing the ECC technique (ECC_DLL). The ECC mitigates missing pulse(s) in a 40 nm technology at 1 GHz, and is similarly effective in other technology nodes. The number in the parenthesis represents the missing pulses generated by the VCDL [76]

4.   Experimental Results of the ECC

To further validate the effectiveness of this error correction circuit, high-speed FPGAs were used to characterize the operation of different VCDLs using pseudo-SET pulse injection to emulate ion strikes. Figures VI-39 to VI-41 show the response of the ECC following the injection of nine missing pulses in the signals *Op1* and *Op2*. Three different cases were characterized:

Missing pulses generated in *Op1* (and *On1*), missing pulses generated in *Op2* (and *On2*), and missing pulses generated in *Op1*, and then in *Op2* one clock cycle later.

As illustrated in Figure VI-39, the signal *Sel* remains at a '0' logic level when *Op1* is corrupted, and the MUX outputs the uncorrupted signal *Op2*, with no missing pulses observed in the DLL output *Vout*. The results in Figure VI-40 show that when injecting missing pulses in the signal *Op2*, the signal *Sel* switches to a logic '1' level and the missing pulses are filtered from *Vout* by outputting the uncorrupted signal *Op1*. Finally, in the unlikely case where missing pulses happen very close in time between *Op1* and *Op2* (c.f. Figure VI-41), the ECC filters the missing pulses and the output *Vout* remains unperturbed.

Note that no duty cycle variation occurs at the output of the ECC_DLL (*Vout*), which can be explained by the clock and routing optimization of the entire ECC design by the Xilinx-ISE compilation tool within the FPGA. These FPGA results experimentally confirm that the error correction circuit hardening technique mitigates missing pulses in DLLs, and validate the previous simulation results. In addition, the use of an on-chip DLL (DMC design by Xilinx) in these experiments indicates that this ECC hardening technique should be effective over a broad range of DLL designs and process nodes.
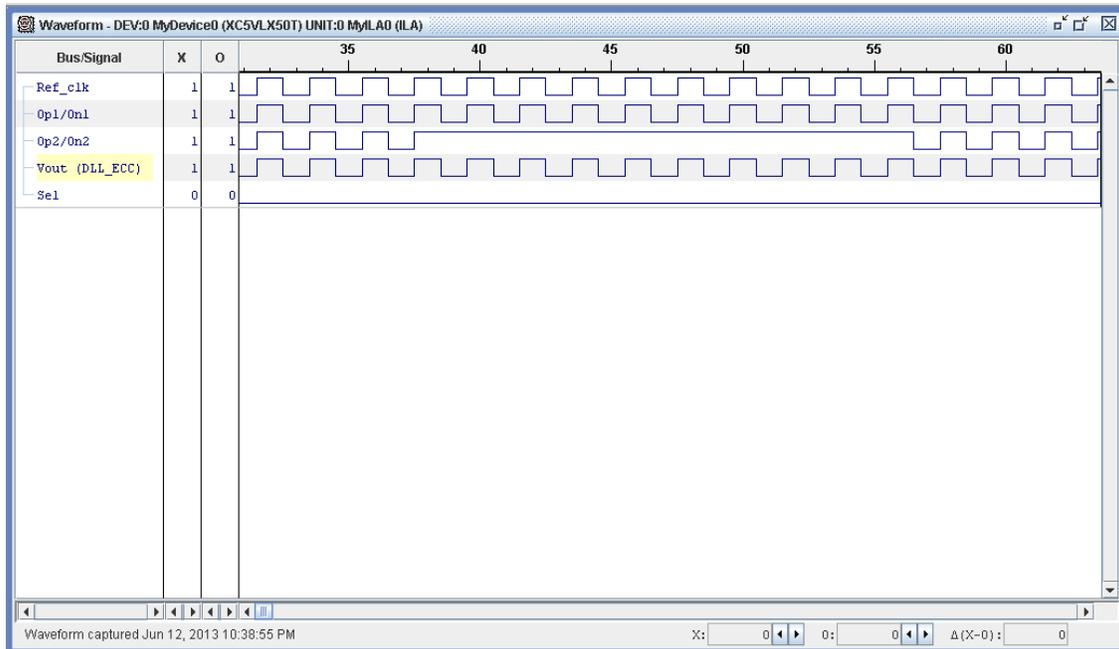
Fig. VI-39 Experimental results of the ECC implemented on the Virtex 5 FPGA. Missing pulses were injected in Op1 and Op2 using a fault injection block controlling the number of missing pulses injected, for a hit in Op2/On2. No missing pulses were observed at the output of the DLL V_{out}. The control signal Sel. of the 2:1 MUX was also monitored [76].
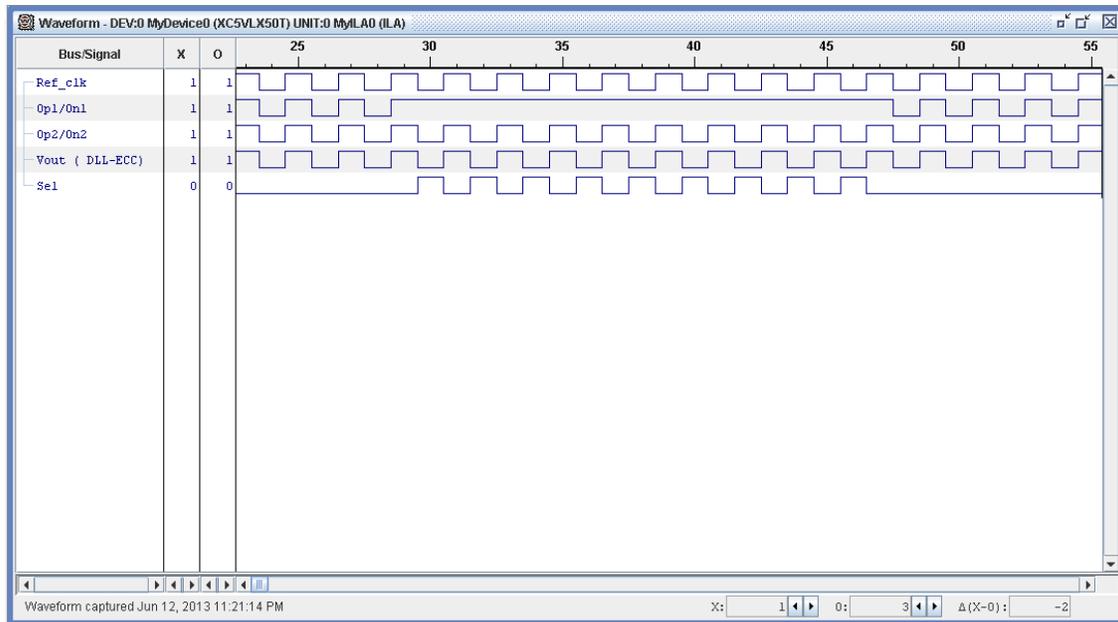
Fig. VI-40 Experimental results of the ECC implemented on the Virtex 5 FPGA. Missing pulses were injected in Op1 and Op2 using a fault injection block controlling the number of missing pulses injected, for a hit in Op1/On1. No missing pulses were observed at the output of the DLL V_{out}. The control signal Sel. of the 2:1 MUX was also monitored [76].
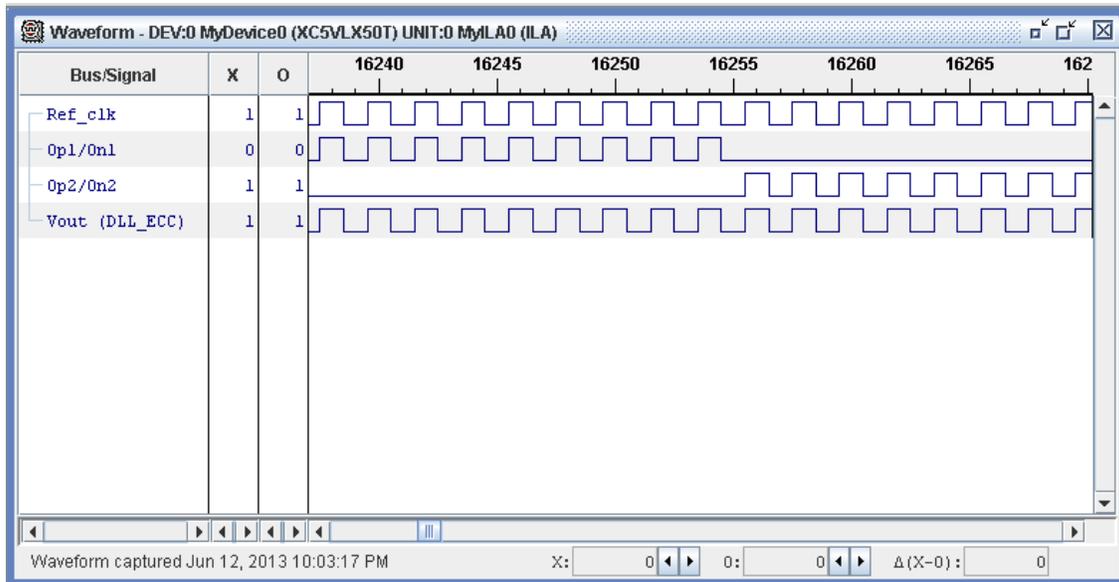
Fig. VI-41 Experimental results of the ECC implemented on the Virtex 5 FPGA. Missing pulses were injected in Op1 and Op2 using a fault injection block controlling the number of missing pulses injected, for a hit in Op1/On1 and Op2/On2. No missing pulses were observed at the output of the DLL V_{out}. The control signal Sel. of the 2:1 MUX was also monitored [76].

a)   *Charge sharing considerations*

As IC processes scale, the reduced spacing between transistor diffusions increases the probability of radiation-induced charge sharing, thereby reducing the effectiveness of redundancy-based hardening techniques [41, 58]. The proposed ECC depends upon the redundancy of the VCDL differential pair to be effective and that charge sharing would not occur in the VCDL. Consequently, layout techniques should be used to minimize charge sharing and avoid any simultaneous corruption of signals *Op1* and *Op2* in case of an ion strike. For the peeled VCDL layout, redundancy comes from creating two parallels signal paths with halved transistor W/L ratios. Charge sharing in the peeled VCDL can be reduced by grouping the PMOSs and NMOSs sharing the same complementary outputs into the same layout area, the sensitive active area associated with each complementary output is then separated by the sources of each

103

transistors as illustrated in Figure VI-42. Therefore the outputs Op1 and Op2 cannot be perturbed by an energetic particle at the same time.
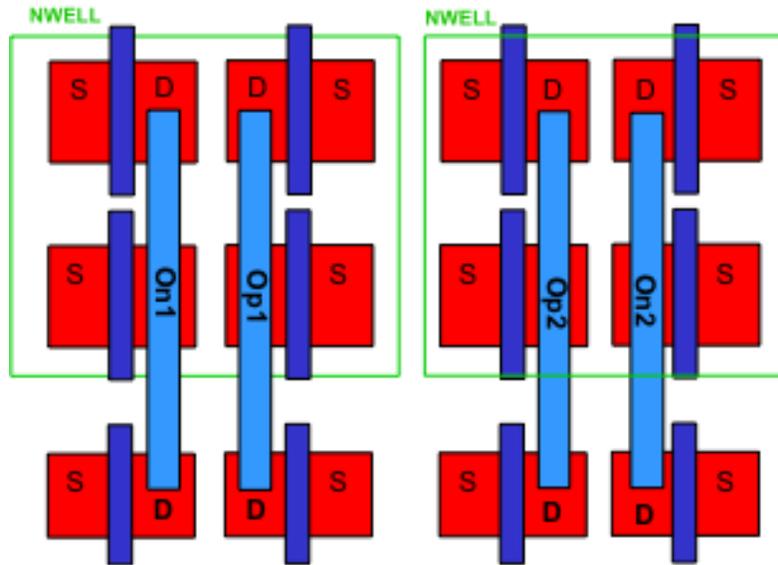
Fig. VI-42 Illustration of peeled delay cell layout to minimize the charge sharing between complementary outputs. PMOS with differential outputs are separated by the source of each PMOSs device. The NWELLs are also separated to avoid charge sharing in the delay cell.

D.   Summary/Discussion

This chapter has presented two hardening techniques to mitigate single event transients - missing pulses - in DLLs. The first technique relies on increasing the critical charge in the VCDL to eliminate missing pulses and the second technique uses combinational logic to detect and eliminate missing pulses at the output of the DLL. The two techniques were implemented with minimal power and area penalty and can be implemented over a broad range of analog DLL designs or circuits implementing analog mixed-signal delay cell elements like digital PLL. Designers can chose the appropriate technique to implement in their circuit based on parameters such as the operating frequency of the circuit or the technology node.

In implementing a practical product, some trade-offs and modifications will be required depending on the chosen hardening technique. Both hardening techniques previously discussed require less than 10% increase of the VCDL layout to be implemented, which would translate at the final product level into a small area and power increase. Differential pair topologies area already being widely used in ICs to reduce power supply noise and jitter in DLLs. Therefore the implementation of the hardened complementary differential pair, requiring less than 5% of area penalty, which on a product size level is a small price for hardening a circuit becomes an attractive design choice, and would allow designers to replace the single ended current starved delay cell in analog mixed-signal circuits with the hardened complementary diff. pair, wherever possible. However this technique is sensitive to technology scaling and does depend on the operating frequency of the system, which may lead to missing pulses in the gigahertz frequency range. As we move towards ever smaller technologies like 32nm, 28nm, 20nm, the scaling dependency may be more pronounced. Therefore I would recommend that this technique should be implemented in applications operating at less than 800MHz, which is still within the market standards for most DLLs. For products sensitive to technology scaling or for applications operating in the gigahertz range, the error correction circuit may be more attractive as it is independent of technology scaling and still results in a less than 5% of area penalty. This error correction technique may increase jitter in the final design. Even if the increase in jitter is below the jitter alignment requirement (10% of the operating clock period, typically) in products, this is a penalty that designers may not be willing to pay. In that case, the designer can implement a toggle in the ECC, reducing duty cycle error and minimizing the jitter response of the overall product. I suggest that this solution should be implemented in products operating in the gigahertz range. Further design optimizations may be required for systems operating in the multi-gigahertz range.

CHAPTER VII

SINGLE EVENT TRANSIENT LINEAR MODEL AND GUIDELINES

FOR ANALOG MIXED-SIGNAL DLLs

A.   Introduction

The final chapter of this dissertation is a unique single event transient analytical model and a set of broad hardening guidelines for the design of analog mixed-signal DLLs and similar clock circuits. The guidelines are based on the developed single event effect DLL analytical model, a set of broad design rules and RHBD techniques proposed to eliminate single event transients in this dissertation (c.f. Chapter VI). The guidelines will assist designers to predict the SET responses and the different design tradeoffs based on design specifications, in order to develop SET immune analog mixed-signal DLLs or similar circuits for space applications.

B.   A Single Event Transient Linear Analytical Model for DLLs

The goal of this work is to develop a general model for single event transient propagation through mixed-signal DLL topologies and similar circuits for space applications. The transient model is based on a conventional linear DLL model commonly used to define the value of the charge pump current, loop filter capacitor and behavioral models [18].

An analytical model predicting the generation and propagation of single event transients in digital PLLs for each sub-circuit was developed by Dr. Loveless in [83]. A single event transient analytical model for analog DLL designs, predicting the impact of single event transient in each DLL sub-circuits is presented in this work. This analytical model is unique because the set of design equations are derived for specific DLL parameters. The analytical model was

106

developed using IBM 90-nm technology node, but the equations can be used to predict the impact of single event transient in analog mixed-signal DLLs at any technology node, including deep sub-micron technologies. In addition, this model is also unique since it is used to derive for the first time a single event linear model that includes delay locked loops and phase locked loop (PLL) circuits, into a single set of equations, generalizing the single event transient model to a broad set of analog mixed-signal clock circuits, by combining the analytical model for DLLs with Dr. Loveless's work in [83].

Finally, throughout the DLL analytical model, a list of general design guidelines is proposed. These guidelines can be applied to mixed-signal DLL topologies and furthermore be applied to similar clock circuit such as digital PLLs. The set of design rules can be applied to minimize and/or eliminate the sensitivity of DLLs to single event transients resulting from an ion strike. Furthermore, the tradeoffs related to each rules used to improve the resiliency of the analog mixed-signal DLL circuits in highly irradiative environments is addressed.

The basic linear model for transient propagation in the DLL is shown in figure VII-1 and represents the cases where the voltage perturbation following an ion strike has the form of a voltage, current or charge. As in [83], all voltage transient perturbations are assumed to be the result of external noise and are generated by a single current pulse perturbation at the output of the voltage controlled delay line, charge pump or phase detector. A set of equations is derived to estimate the impact of single event transients generated in DLLs, following an ion strike in each DLL sub-circuits. To characterize the SET response of the DLL two metrics are being used: the control voltage perturbation (V_e) fed into the VCDL and the DLL's recovery time (T_{rec}) following each control voltage perturbation. Both metrics are derived for each DLL sub-circuit following the ion strike.

The linear model of the DLL is illustrated in Figure VII-1, where K_D is the gain of the phase detector and the charge pump, $K_{LF}(s)$ is the loop filter transfer function, and K_v is the gain of the voltage controlled delay line. K_D has units of Amperes/radians and K_v has units of Volts/seconds. The single pole of the DLL is introduced by the loop filter capacitor and therefore the circuit has a first order transfer function.
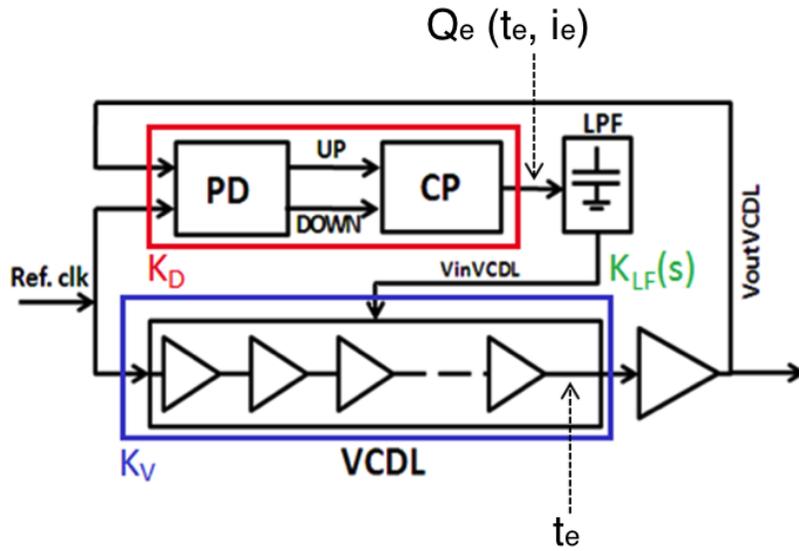
Fig. VII-1 Close loop model for the analog DLL.

The output phase of the DLL is related to the VCDL input by Equation (10), where Φ_{out} is the output phase of the circuit, Φ_{in} is the input phase, Φ_d is the initial phase error due to the input clock skew and T_{clk} is the period of the DLL's input clock. The output phase can further be expressed as a function of the time delay, t_d, between the reference clock and the output of the DLL, where typically $-\pi \leq t_d \leq +\pi$.

$$\phi_{out} = \phi_{in} + \phi_d = \phi_{out} + t_d * \frac{2\pi}{T_{clk}} \tag{10}$$

The transcription of the DLL, H(s), is represented in Equation (11), where K_v represents the gain of the voltage controlled delay line, T_{clk} is the operating frequency of the DLL, I_p is the current of the charge pump, C_{LF} represents the loop filter capacitor and ω_n is the DLL's bandwidth. The derivation of the transfer function equation and gains of each DLL sub-circuits are presented in APPENDIX-B.

$$H(s) = \frac{\phi_{out}}{\phi_{in}} = \frac{1}{1 + \dfrac{s}{\dfrac{K_v}{T_{clk}} * \dfrac{2I_p}{C_{LF}}}} = \frac{1}{1 + \dfrac{s}{\omega_n}} \tag{11}$$

The DLL is a single pole system i.e. the system is stable providing that ω_n satisfies the following condition: $\omega_n < 0.1* \omega_{loop}$, where ω_{loop} represents the loop bandwidth. A wide loop bandwidth, i.e. a too fast response of the DLL would result in a jitter increase.
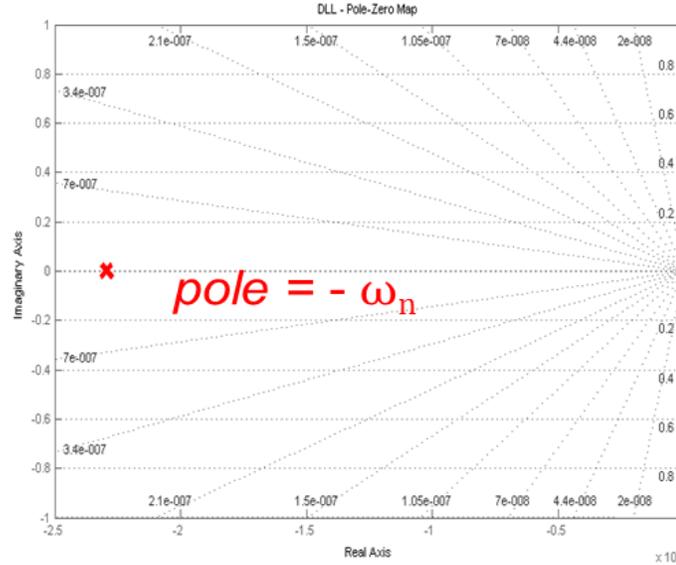
Fig. VII-2 Zero-pole map of the DLL, demonstrating the stability of the single pole clock circuit. The DLL has a negative Real pole meaning the absolute stability of the system.

Following an ion strike within the DLL, the VCDL's control voltage abruptly changes. This voltage perturbation (V_e) will be translated into a change in the VCDL output delay, which may force the DLL to go out of lock, if the voltage perturbation V_e is large enough. The amount of time it takes for the DLL to recover its lock state is defined as the DLL recovery time T_{rec}. T_{rec} can be expressed as a function of the voltage perturbation V_e and therefore will strongly depend on the DLL's sub-circuit being hit (since the expression of V_e itself varies with the sub-circuits). Equation (12) represents the general relation between the recovery time of the DLL following an ion strike, where V_e represents the DLL's control voltage perturbation (as a function of the charge deposited Q_e and the SET duration t_e), I_p represents the charge pump current and C_{LF} is the loop filter capacitor. t_e is the difference between the input reference clock period and the SET perturbation period.

110

$$T_{rec} = \frac{C_{LF}}{2I_p} * V_e\,(Q_e, t_e) \tag{12}$$

Equation (13) describes the control voltage perturbation following an ion strike in the PD (V_{ePD}), where t_e is the amount of output delay error generated by a strike within the charge pump, I_p is the charge pump current and C_{LF} is the loop filter capacitor. The plus or minus emulate the case if the UP or DOWN path of the phase detector is being hit. The charge deposited in the PD will affect the output response of the DLL, if t_e is larger that the dead zone of the DLL.

$$V_{ePD} = \pm\frac{t_e * 2I_p}{C_{LF}} \tag{13}$$

Equation (14) describes the control voltage perturbation following an ion strike in the charge pump (V_{eCP}), where again t_e is the amount of delay error generated by a strike within the charge pump, Q_e is the charge deposited into the module and C_{LF} is the loop filter capacitor. The plus or minus mimics the case where the UP or DOWN path of the PD is being activated and the CP sources/sinks current to/from the loop filter capacitor. In the case of the charge pump, the charge deposited in the CP will directly affect the DLL control voltage (V_{ctrl}) and therefore should be added in the expression of the CP voltage perturbation equation. It also means that the largest voltage perturbations (and by extension recovery time) in the DLL will be observed following an ion strike within the charge pump sub-circuit.

$$V_{eCP} = \pm\frac{Q_e + (t_e * I_p)}{C_{LF}} \tag{14}$$

Equation (15) describes the control voltage perturbation following an ion strike in the VCDL (V_{eVCDL}), where t_e is the amount of delay error generated by a strike within the voltage controlled delay line, I_p is the charge pump current and C_{LF} is the loop filter capacitor. As for the phase detector, the voltage perturbation V_{eVCDL} will affect the DLL only if t_e is larger that the dead zone of the circuit.

$$V_{eVCDL} = \frac{t_e * 2I_p}{C_{LF}} \tag{15}$$

Detailed demonstration of the equations in each of the DLL's sub-circuit are derived in APPENDIX-B.

It is noticeable from Equations (13), (14) and (15) that a charge deposited within the CP will generate the largest control voltage perturbation, when compared to the VCDL and PD, forcing the circuit to go out of lock, if V_e is large enough. Typically, following the single event transient, the DLL will regain lock by adjusting the VCDL control voltage to its original lock value. In some particular cases, like the charge pump, the circuit will recover its lock state but not in the initial locking phase due to the nature of the phase detector sub-circuit in analog DLLs. This off phase locked state is then considered as an error resulting from the single event strike and refers to a persistent inverted lock error, c.f. Chapter V of this dissertation. Therefore, evaluating the recovery times following single event transients within the phase detector, charge pump or voltage controlled delay line sub-circuits is also an important parameter, since the DLL needs to reacquire lock as fast as possible to ensure the proper operation of the overall system.

1. DLL SET Model Validation and Critical Constant V_{crit}

Figure VII-3 illustrates an example of a voltage transient perturbation on the charge pump resulting from a perturbation at the output of the CP sub-circuit. The simulation was performed on a DLL designed in 90 nm PDK. A DC current source (IDC) was used to model single event transients in CADENCE and the charge deposited was obtained by integrating the current over time using CADENCE calculator tool. The single event transient was induced by a 65.4 fC charge deposited at the output node of the CP and collected by the loop filter capacitor. As indicated in Figure VII-3, the simulated recovery time, ignoring any settling effects is approximately 34.74 ns. Using the Equations (12) and (14), for an ion strike in the CP, the estimated recovery time is 32.3 ns, which represents a 7% deviation from the simulated DLL recovery time. As indicated in Figure VII-2, the simulated recovery time ignoring any settling effects is approximately 140 ns. Equations (16), (17) show the calculation of V_e and T_{rec} following a perturbation in the charge pump using the parameters in Figure VII-3:

$$V_e = \frac{65.4f + (1\mu * 215p)}{1p} = 64.7 \text{ mV, where Ve (sim.)} = 63.4 \text{ mV} \rightarrow \Delta Ve \sim 2\% \qquad (16)$$

$$T_{rec} = \frac{1p}{2 * 1\mu} * 67.4m = 32.3 \text{ ns, where Trec (sim.)} = 34.74 \text{ ns} \rightarrow \Delta Trec \sim 7\% \qquad (17)$$

The calculated recovery time T_{rec} and voltage perturbation V_e show less than a 10% deviation when compared to the simulated DLL voltage perturbation and recovery time. This first example shows a good correlation between the SE model and the simulations.
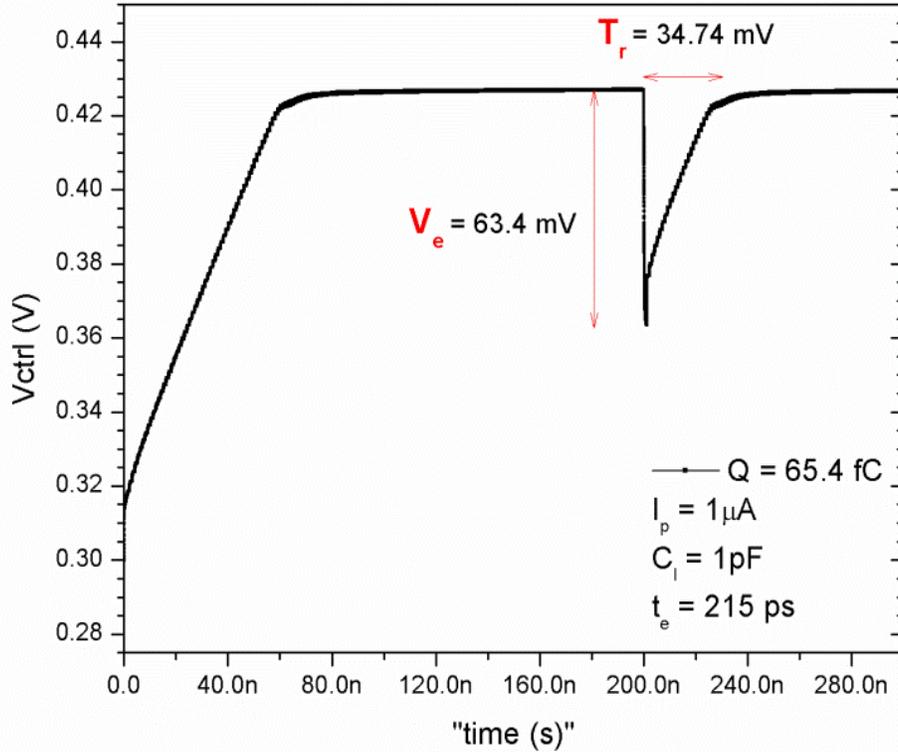
Fig. VII-3 DLL acquisition curve, at 1 GHz for a charge deposition of ~66 fC within the charge pump of the 90nm DLL, illustrating the voltage perturbation V_e and the recovery time T_{rec} of the circuit.

In order to validate the proposed SET model for DLLs, the single event transient response of the DLL was compared for different charge depositions against the DLL analytical model for 1,000 data points. Figure VII-4 illustrates the comparison between simulations and the linear model within a range of ~ 110 data points for the voltage perturbation model. The analytical model matches the simulation with a 95% confidence level. In the same manner, Figure VII-5 illustrates the comparison between simulations and the linear model within a range of ~ 110 data points for the recovery time model. Figures VII-4 and VII-5 reveal a noticeable divergence between the DLL linear model and the simulations that becomes noticeable at higher values of

deposited charge. When the amount of charge deposited in the circuit exceeds a critical value, then the DLL will not recover and the recovery time of the circuit will tend to infinity ($T_{rec} \rightarrow \infty$).

The divergence observed between the DLL linear model and the simulations can be explained by the illustration in Figure VII-6. As shown in the figure, increasing the amount of charge deposited in the DLL linearly increases the voltage perturbation generated by the circuit until, for a threshold charge deposition value, in this example 150 fC, the DLL will be forced to lock at π due to the limitation of the phase detector operating range, thus the recovery time of the analog DLL is considered infinite.
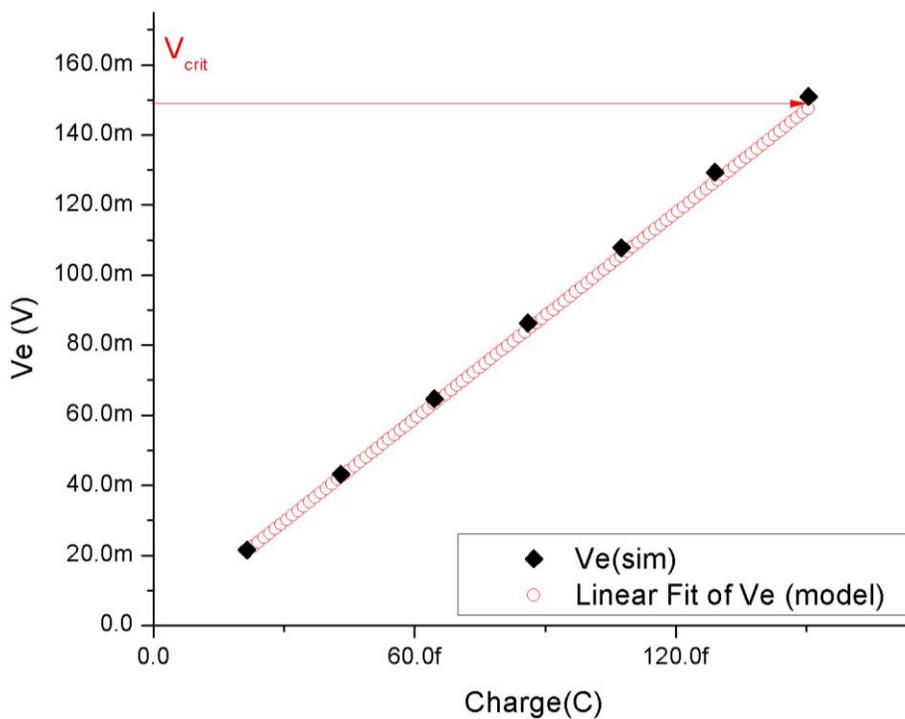
Fig. VII-4 Comparison between the simulated (V_e (sim)) and calculated (Linear Fit) voltage perturbation (V_e) for different charge deposition within the charge pump of the DLL, in 90 nm technology node . The DLL analytical model matches very well the simulations. V_{crit} represent the critical voltage perturbation where the DLL does not recover to its initial phase lock value.

115
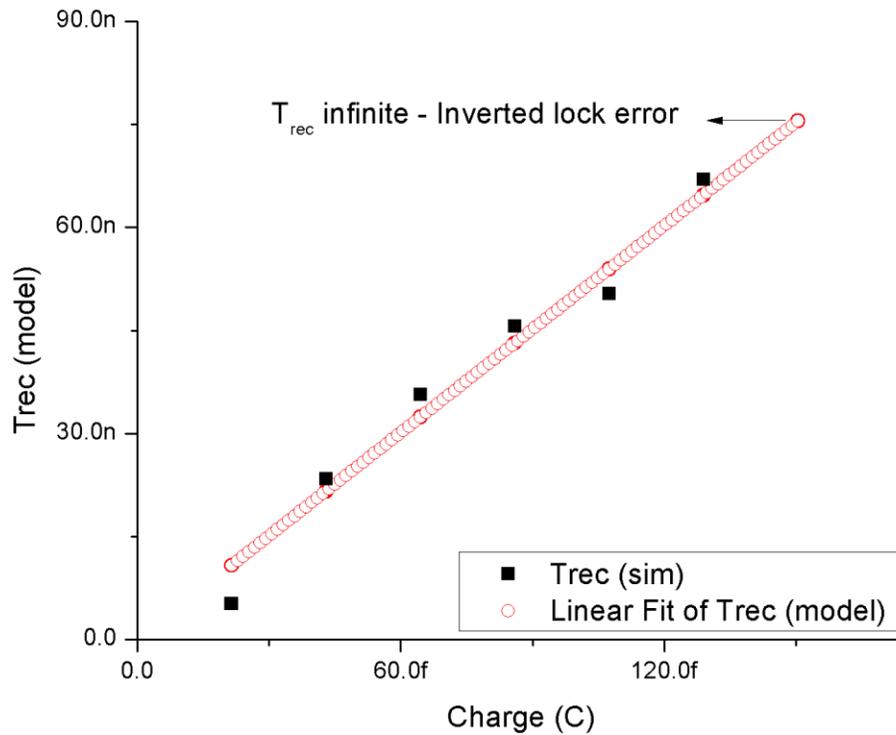
Fig. VII-5 Comparison between the simulated (T_{rec} (sim)) and calculated (Linear Fit) recovery time (T_{rec}) for different charge deposition within the charge pump of the DLL, in 90 nm technology node. The DLL analytical model matches very well the simulations until V_e reaches V_{crit} value where the recovery time is infinite because the DLL does not recover to its initial lock phase.
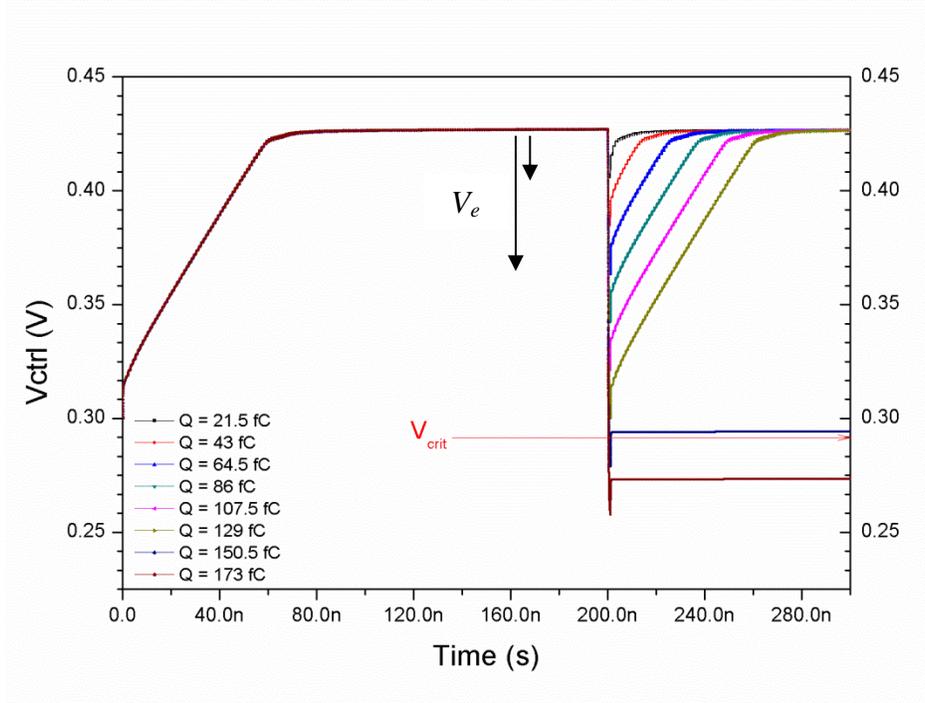
Fig. VII-6 DLL acquisition curve (DLL control voltage vs. time), at 1 GHz at different charge deposition (Q) values, illustrating the relation between the voltage perturbation V_e and the persistent inverted lock error generated by an ion strike in analog DLLs. V_{crit} represents the minimum value of the control voltage perturbation V_e, which generates inverted lock errors and the DLL to not recover to its original lock phase.

If Q_e is higher or equal to that critical charge deposition value and even if the DLL re-locks, the recovery time of the circuit is considered infinite since the DLL does not regain its original locking phase. Therefore, we introduce a new critical constant for DLLs called the critical voltage perturbation V_{crit}. V_{crit} represents the minimum value (or threshold value) for the control voltage perturbation V_e, before the DLL will be forced to lock at plus or minus π phase radians and therefore never recovers. In addition, it is interesting to note that V_{crit} usually represents the limits of the VCDL linear operating region. This observation will be useful for developing general guidelines for hardened analog DLLs and similar clock circuits.

Now that the critical constant V_{crit} is defined, we can calculate and plot the single event transient analytical response of the DLL as a function of deposited charge, voltage perturbation and dead zone (Fig. VII-7). Note that the DLL recovery time is bounded by the phase detector's dead zone. Within that range the voltage pertubation will not force the DLL to go out of lock and therefore the recovery time is very small (a few picoseconds or less). Therefore, we can predict the behavior of the DLL for any value of deposited charge. In addition, the designer can implement any desired value of the loop filter capacitor, charge pump current, etc. in the SEE DLL analytical model, at any technology node, to predict the single event transient response of his analog mixed-signal circuit and extrapolate it to his system. Finally, broad design rules in combination with the hardening techniques (developed in this work) for the design of high-speed radiation hardened analog mixed-signal DLLs will be provided later in this chapter. However, first let's consider the possibility of combining analog mixed-signal DLLs and PLLs single event models into a single and unique SET linear model.
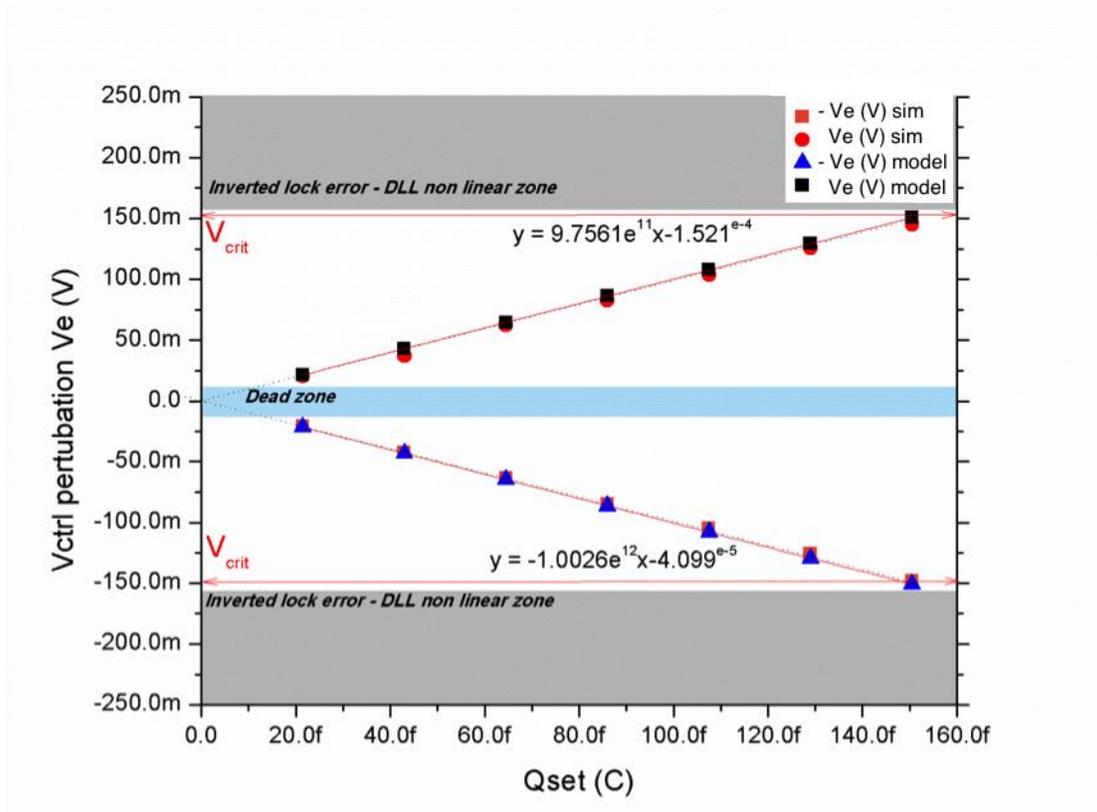
Fig. VII-7 DLL single event transient linear model as a function of the deposited charge in the CP. The SET linear response of the DLL is bounded by the critical voltage V_{crit} (linear operating range of the VCDL) and the dead zone of the circuit. Any voltage perturbation within the dead zone will not impact the DLL's lock state.

C.   A New Generalized Linear Model for DLLs and PLLs

In addition to the unique single event effect analytical model for DLLs, this work introduces a new general SET linear DLL/PLL model that describes the SET propagation in both DLLs and PLLs using the constant "ρ" to combine this dissertation work on DLLs with Dr. Loveless's single event analytical model for PLLs in [83].

The equations for the DLL voltage perturbation V_e and recovery time T_{rec} are very similar to the ones for PLLs, as shown in Equations (18) and (19). The expressions for the recovery times

119

of the DLL (presented in this dissertation) and the PLL following an ion strike have a common parameter called "m" [83].

The recovery time T_{rec} for the analog DLL sub-circuits is given by Equation (18), where V_e represents the DLL's control voltage perturbation, I_p represents the charge pump current and C_{LF} is the loop filter capacitor:

$$T_{rec} = \frac{C_{LF}}{2I_p} * V_e \qquad (18)$$

The recovery time T_{rec} following an ion strike within the PLL is given by the following equation [83]:

$$T_{rec} = \frac{1}{m} * V_e; \; with \; \frac{1}{m} = \frac{C_{LF}}{I_p} \qquad (19)$$

Where V_e is the PLL control voltage perturbation, "m" is a constant that represents the ratio between the PLL's first order loop filter capacitor C_{LF} and the charge pump's current I_p. The phase detection range of the PLL is two times larger than the detection range of the phase detector in DLLs, i.e. $-2\pi \leq \Phi_{PLL} \leq +2\pi$ for the PLL vs. $-\pi \leq \Phi_{DLL} \leq +\pi$, for the DLL. Therefore, for a first order linear model of the DLL a factor of two differentiates the single event analytical model of the PLL from the DLL. Using the ratio "m" to combine both DLL and PLL recovery time equations into a single analytical model shows a ratio of 1 and ½ for PLLs and DLLs, respectively. Thus, a new parameter "ρ", is introduced to generalize the single event linear model for both PLLs and DLLs, where ρ= 1 for PLLs and ρ= ½ for DLLs. The Equations (18) and (19) can then be combined into a single equation expressed in Equation (20).

$$T_{rec} = \frac{\rho}{m} * V_e; \quad \rho = \begin{cases} \frac{1}{2}; & for\ DLL \\ 1; & for\ PLL \end{cases} \tag{20}$$

Table VII-1 describes the generalized linear model for each DLL/PLL sub-circuit, where t_e is the delay error generated by the circuit following an ion strike (i.e. the difference between the SET period and the reference clock period), Q_e is the charge deposited at the sensitive node, I_p is the charge pump current and C_{LF} is the loop filter capacitor. The new parameter "ρ" is implemented in the table below to develop a generalized set of single event transient linear equations to analyze the single event response of the DLLs and PLLs. This generalized model implies that most of the single event guidelines, hardening techniques and rules, proposed in the dissertation for DLLs can also be applied to PLLs and similar clock circuit.

Table VII-1 Generalized single event DLL/PLL linear model for each sub-circuits. The model combines the DLL single event linear model with PLL single event transient model in [83] using a new parameter " ρ ". The parameter ρ = 1 for PLLs and ½ for DLLs

Sub-circuit	V_e	T_{rec}
PD/PFD	$\frac{1}{\rho} * \left(\pm \frac{t_e * 2I_p}{C_{LF}} \right)$	t_e
CP	$\pm \frac{Q_e + (t_e * I_p)}{C_{LF}}$	$\rho * \left(\pm \frac{Q_e}{I_p} + t_e \right)$
VCDL/VCO	$\frac{1}{\rho} * \left(\pm \frac{t_e * 2I_p}{C_{LF}} \right)$	$|t_e|$

D.   Guidelines for Single Event Transient Mitigation in DLLs and Similar Clock Circuits

The observations and results regarding the developed DLL analytical model and hardening solutions are the basis for circuit-design guidelines to reduce or mitigate SETs in the DLL sub-circuits. The first set of design guidelines uses the observations and derived equations from the DLL's single-event analytical model to define rules to help designers customize and optimize their DLL designs by predicting the SET response of the circuit before building it. The second portion of the guidelines, presented in table VII-7, helps the designer choose the appropriate hardening technique to keep their DLL operations optimum based on their circuit specifications.

A first-pass single-event estimation of a circuit's SEE vulnerability prior to being built would be very beneficial for the designer. Using the single-event analytical model for DLLs (extended to DLLs/PLLs), the designer could choose specific design parameters, such as the voltage-controlled delay-line linear operating range, charge-pump current, and loop-filter capacitor to control the type of single-event perturbation and recovery time expected from the design. The designer could also predict the single-event response of the DLL being developed as a function of specific design parameters provided by a client. This would allow the designer to propose a more optimized solution. In addition, in order to predict the single-event response of a particular DLL, we used observations made on the SE analytical model for DLLs, presented earlier in this chapter, to develop a set of general design guidelines to help reduce the SEE sensitivity of the circuit without applying dedicated hardening techniques (c.f. Table VII-7) that would require a modification of the "original" sub-circuit's topology. For each new SET design rule, a tradeoff between single event effect performances of the circuit and power/area penalty is discussed.

1.   **Rule 1- Increase the loop filter capacitor C_{LF}:**

The voltage perturbation (V_e) of the DLL, following an ion strike, is inversely proportional to the loop-filter capacitor, as shown in the following equation:

$$V_e = \frac{\Delta Q_e * 2I_p}{C_{LF}} \qquad (20)$$

Where C_{LF} is the loop filter capacitor, I_p is the charge pump current and ΔQ_e is the charge deposited in the DLL. As the loop filter capacitor increases, the voltage perturbation generated in the DLL for a fixed charge deposition value decreases. Therefore, increasing the loop-filter capacitor will increase the charge deposition required for the voltage perturbation to be equal to V_{crit}, as illustrated in Figure VII-8. As an example, in Figure VII-8 a 2X increase in the loop capacitor increases the critical charge for V_e to reach V_{crit} value by 2.3X. Another impact of increasing the loop-filter capacitor is an increase in the DLL's recovery time. Therefore, a tradeoff between the size of C_{LF} and the DLL recovery time must be considered by the designer, to not penalize the timing specification of the overall chip. In addition, the loop filter capacitor must be balanced against DLL design lock-time specifications and available silicon area, since the layout of the capacitor usually requires a significant silicon area increase.
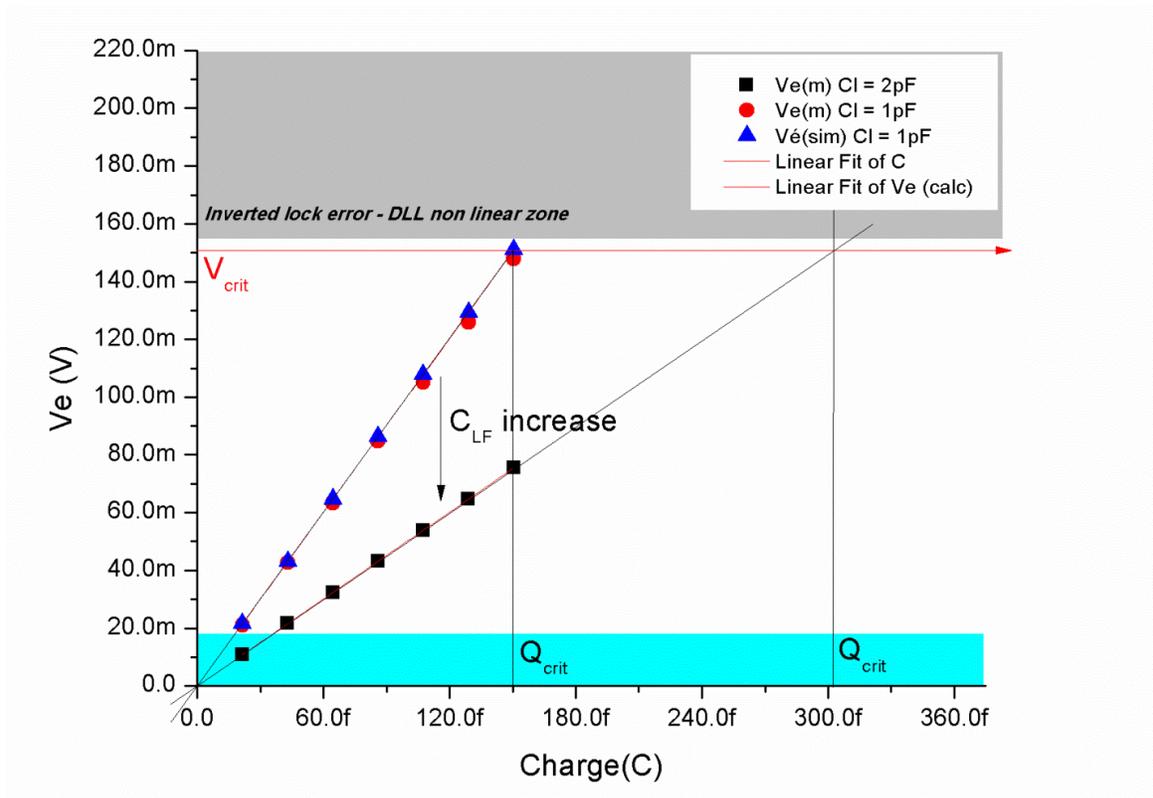
Fig. VII-8 Single event voltage perturbation (V_e) within the charge pump of the DLL as a function of the charge deposited. Increasing the loop filter capacitor (C_{LF}) increases the critical charge related to V_{crit}. Ve(m) are simulation results and Linear Fit are the calculated data points based on the DLL SET analytical model.

2.   Rule 2 - Increase the linear range of the VCDL:

The value of V_{crit} can be estimated by looking at the boundaries of the VCDL linear operation zone as illustrated in Figure VII-9. Therefore, the linear operating range of the sub-circuit should be as large as possible to increase the deposited critical charge required to reach V_e = V_{crit}. This is because the upper and lower limits of V_{crit} are bounded by the linear operating range of VCDL, as represented by the red delay vs. control curve in Figure VII-9. As an example, in Figure VII-9, increasing the linear range of the VCDL increases the value of V_{crit} by 30%. Therefore, a larger amount of charge must be deposited in the DLL to reach V_{crit}. However, while

increasing the VCDL's linear operating range, the designer must still comply with the VCDL basic operating range requirements, $-\pi \leq td \leq +\pi$, to avoid harmonic or false locking. Increasing the VCDL's linear range requires increase the size of the delay cells and therefore would require a larger layout size for the sub-circuit. Therefore, the silicon area necessary for laying out the VCDL by implementing this rule must be weighed against the circuit's power and area requirements.
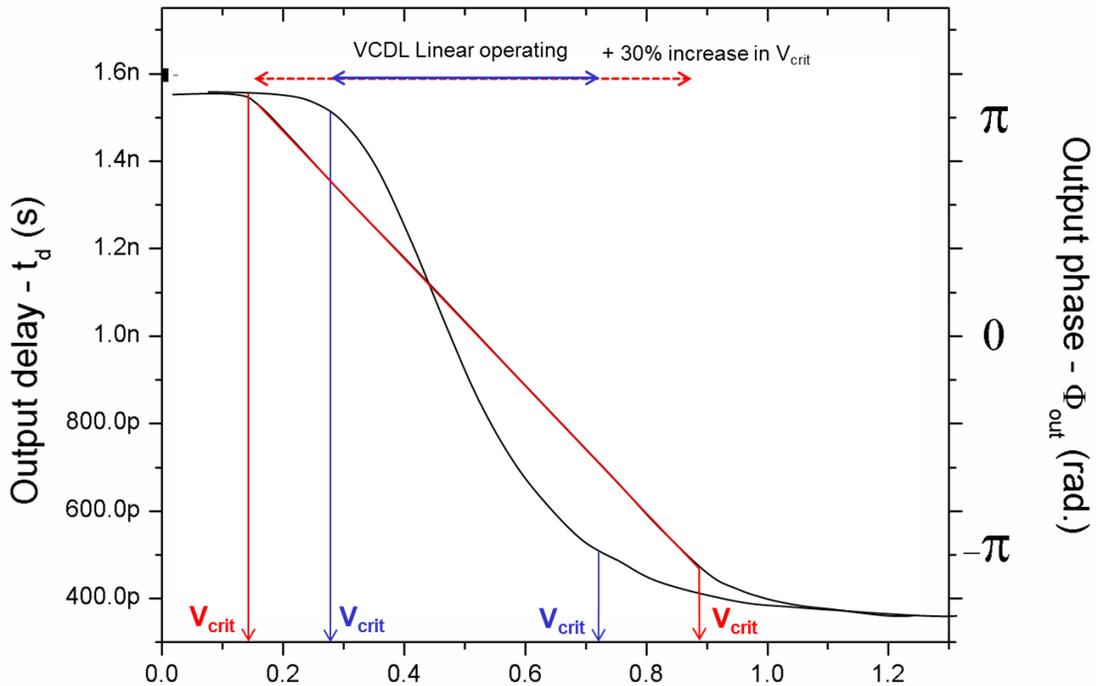
Fig. VII-9 DLL output delay (td) vs. control voltage illustrating the linear range of the VCDL and V_{crit} boundaries. Increasing the linear range of the VCDL increases V_{crit} values and by extension the critical charge related to $V_e = V_{crit}$.

3.   **Rule 3 – Increase V_{lock} of the DLL:**

In lock state, the DLL locking voltage V_{lock} for a given operating frequency, i.e. the voltage for which the phase difference between the DLL input and output signals is very close to

zero radians should be greater than half the supply voltage, as illustrated by the red delay vs. VCDL control voltage curve in Figure VII-10. By "shifting V_{lock} to the right" of the delay vs. control voltage curve, the restoring current of the VCDL, following an ion strike, increases. Therefore, the higher the control voltage (V_{lock}), the higher is the VCDL restoring current and therefore the more resilient to single event transients is the VCDL. There is a limit to this rule, as the DLL must still to operate within 0.5 and 1.5 of the reference clock period. Increasing the value of V_{lock} also means a narrow linear operating range and, therefore this is a counter rule to rule 2 and the designer must carefully weight each rules or apply the rule that would offer the best single event transient response for his design requirements.
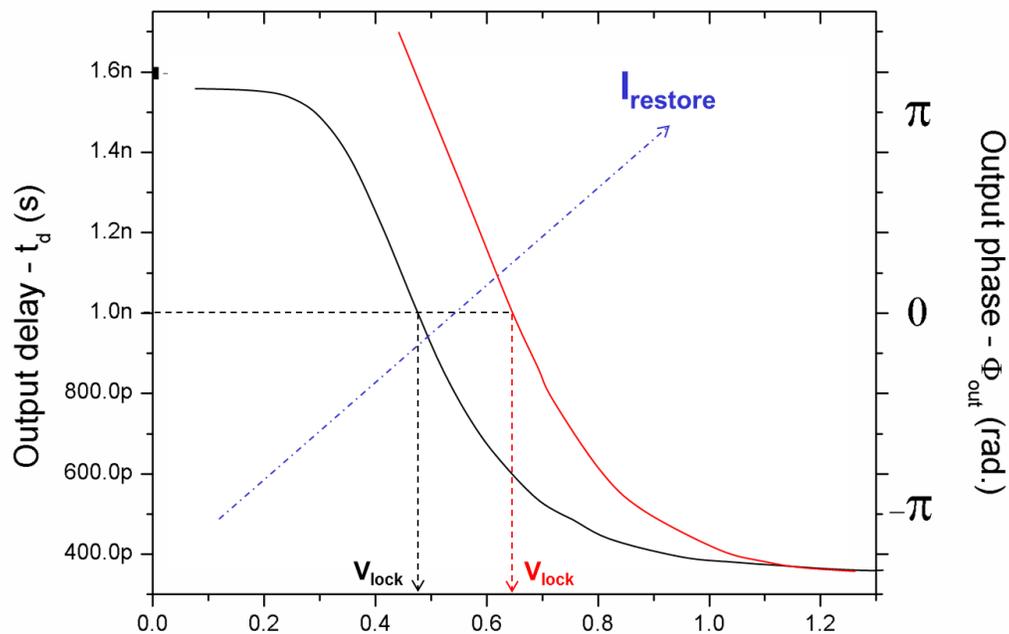
Fig. VII-10 VCDL operating range, output delay (t_d) vs. input control voltage. In black: basic DLL design locking voltage. In red: illustration of guideline rule number 3, the lock voltage V_{lock} should be large, in order to increase the value of the VCDL's restoring current and improve the resiliency of the DLL to single event transients.

4.   **Rule 4 – Decrease the charge pump current I_p:**

As shown in Equation (20), the charge-pump current (I_p) is proportional to the voltage perturbation V_e of the DLL following an ion strike. Since V_{crit} represents the minimum value for the control voltage perturbation V_e to force an analog DLL to lock at π phase radians (i.e. that the circuit will never recover and T_{rec} is infinite), the appropriate strategy is to minimize V_e. Minimizing the charge pump current would reduce the voltage perturbation generated by an ion strike in the DLL thus reducing the critical voltage required for V_e to be equal to V_{crit}. Decreasing the current I_p will increase the DLL's recovery time and therefore the circuit's lock time following the ion strike. Therefore, a tradeoff between the charge pump current and the design timing specifications must be made to guarantee the optimum response of the DLL in terms of design performances and single-event resiliency of the circuit.

The broad single event transient design rules proposed to reduce single event transient for typical analog DLL design (that could be expended to general digital PLL designs and similar clock circuits) depend on the design parameters, such as the charge pump current, loop filter capacitor, operating range of the delay chain, etc. and therefore can also be used to reduce single event in low technology nodes (i.e. 40 nm and below). However, as technology scales down the amount of critical charge deposition for V_e to reach V_{crit} decreases and as a consequence the size of the design parameters may increase to obtain the same single event results as in older technologies. Therefore a tradeoff between the performance of the analog mixed-signal clock circuit and acceptable single event transient response will have to be made when implementing the rules to the system.

In addition to the four design guidelines for optimizing the single event transient response and performances of the analog DLL designs, the proposed rules can be combined with the

hardening technique summarized in Table VII-2 to increase the resiliency of the DLL to single event transients, based on design requirements. Table VII-2 summarizes the different types of hardening techniques, developed in this dissertation, for mitigating the different types of single event transient shapes observed in the DLL following an ion strike and provides, for each SET error signature, a solution to eliminate or reduce the single event effect based on the function of the sub-circuit being hit. The development of this table has been another main objective of this document, as it will guide the designer when building hardened analog mixed-signal locked loop circuits, based on design specification such the operating frequency of the DLL or the technology node. As an example: for high frequency applications, i.e. strictly higher than 500 MHz, the error correction technique using combinational logic for the differential delay line in general would be a better choice to mitigate missing pulses than the complementary differential pair for a DLL designed in a 40 nm process.

Table VII-2 Summary of different hardening solution proposed in this document, as a function of the error signature, operating frequency, technology node and LET value. The green "V" means that the hardening solution can be applied to a circuit meeting the design specs and the red "X" means that the technique could not be implemented in the DLL because the design specifications do not allow it.

RHBD solution	SET Error	Technologies ≥ 40 nm				Technologies ≤ 40 nm			
		Frequency ≤ 500 MHz	Frequency ≥ 500 MHz to 1GHz	LET ≤ 50 *	LET ≥ 50 * up to 100 *	Frequency ≤ 500 MHz	Frequency ≥ 500 MHz to 1GHz	LET ≤ 50*	LET ≥ 50* up to 100 *
RHBD diff. pair VCDL	Missing pulse(s)	V	V	V	V	V	X	V	V
ECC	Missing pulse(s)	V	V	V	V	V	V	V	V
VCP	Inverted Lock error	V	V	V	V	V	V	V	V
Increase C_{LF}	Inverted Lock error	V	V	V	V	V	V	V	V

E.   Summary/Discussion

A unique single-event transient analytical model for analog DLL designs is developed and expanded into a new generalized DLL/PLL single event linear model. The analytical model provides a set of equations to the designer with important insights into the generation and propagation of single event transients in analog mixed-signal clock circuits, such as DLLs or PLLs and by extension to the overall chip based on the circuit design parameters.

In addition, a broad set of guidelines for analog DLLs that can be applied to digital PLLs and similar clock circuits was provided. The broad guidelines are based on a set of general design rules and a table of hardening techniques presented in this document to reduce the single event transients in the chip based on the design specification of their analog mixed signal clock circuits.

As technology scales into low voltage, very deep submicron processes, the hardening of the complementary diff. pair VCDL technique presented in the previous chapter will reach its limit due to the decrease of the critical charge related to technology scaling, therefore increasing the restoring current will require increasing the delay cell size and will impact the overall performance of the chip in term of silicon area, power and timing performance. Moreover, the error correction circuit is independent of technology scaling and will be useful as we reach even deeper submicron technologies, as the jitter response of analog mixed-signal DLLs is not expected to significantly degrade.

The single event transient design rules proposed to reduce SETs in analog DLL designs (that could be expended to mixed-signal PLLs and similar clock circuits) depend on design parameters, such as the charge pump current, loop filter capacitor, operating range of the VCDL, etc. and therefore can also be used to reduce single event transients in low technology nodes for analog mixed-signal DLLs, PLLs and similar clock circuits. The size of the design parameters may increase to obtain the same single event results as in older technologies and therefore a tradeoff between the performance of the circuit and acceptable single event transient response will have to be made when implementing the guidelines to the system.

CHAPTER VIII

CONCLUSION

With completion of this PhD work, we have significantly expanded the state-of-the-art in the design of hardened DLL clocking circuits and similar clock circuits for rad-hard applications by the following contributions to the rad-hard community and the Department of Defense (DoD).

Based on the analysis of single event effects of analog mixed-signal DLLs, we have identified the voltage controlled delay line (VCDL) sub-circuit as the most sensitive to single particle radiation – generating missing clock pulses that increase in number with the operating frequency of the circuit. To our knowledge, this is the first work where the hardening solution for SEs (missing pulses) in analog mixed-signal DLLs was investigated and developed.

We have developed two novel design techniques for the mitigation of analog DLL missing pulses that are fully implementable in modern CMOS technologies. Both developed techniques can be implemented with minimal area and power penalty when compared to the well-established digital technique of triple modular redundancy (TMR), associated with which are large silicon area and power penalties. These techniques offer to the community the choice of hardening solution using a restoring current technique in the VCDL sub-circuit to inhibit the creation of missing pulse errors, or using a combinational logic error monitoring technique to correct missing pulses after they occur in real time. In addition, these hardening techniques have been extrapolated to general analog DLLs circuits and other clock circuits, such as digital PLLs.

The first hardening technique uses a hardened complementary differential pair VCDL to increase the critical charge (Q_{crit}) necessary for single event transient generation and thus mitigate missing pulses at the source. Our implementation of this technique at 180 nm, 90 nm and 40 nm

131

required less than a 5% area penalty over a non-hardened design. To experimentally validate this technique, hardened VCDLs were designed and fabricated in 180 nm and 40 nm technologies. The second hardening technique is based on combinational logic pulse monitoring, and uses an error correction circuit to mitigate the missing pulses as they occur. This ECC technique is implemented via a "peeled" VCDL (i.e. each transistor is split in area but doubled in multiplicity) in a Xilinx Virtex 5 FPGA. We have shown the effectiveness of this technique in other DTRA-supported projects in analog signal hardening with virtually zero power and area penalty (the only penalty occurs due to routing constraints). Furthermore, this new ECC technique is independent of technology scaling – a highly valuable attribute for sub-40 nm design applications, some solution to reduce jitter in the error correction block are also proposed, for high speed applications.

In addition to the two novel hardening solutions, we developed, for the first time, a unique single event transient analytical model for general analog DLL design expanded into a new generalized analog mixed-signal DLL/PLL single event linear model in combination with an analytical model for PLLs previously developed at Vanderbilt. These analytical models provides a set of equations to the designer for important insight into the generation and propagation of single events transient in clock circuits, such as analog mixed-signal DLLs or PLLs to help the designer optimize the circuit performance based on design specifications.

Finally, a broad set of guidelines was provided to reduce single event transients in mixed-signal DLLs and similar clock circuits such a digital PLLs, using a set of: general design rules, that are independent of technology scaling since they depend on the circuit's loop filter capacitor or other design parameters and a table of hardening techniques to help designers optimize their chip based on design specifications for different types of irradiative environment applications.

APPENDIX-A: List of Recurrent Technical Anachronisms

ANACHRONISM	**DEFINITION**
AMS	Analog Mixed-Signal
CMOS	Complementary Metal-Oxide-Semiconductor
CP	Charge Pump
CS	Single Ended Current Starved
DLL	Delay Locked Loop
DP	Complementary Differential Pair
ECC	Error Correction Circuit
IC	Integrated Circuit
LET	Linear Energy Transfer
LPF	Loop Filter (or Low Pass Filter)
NMOS	N-Type Metal Oxide Semiconductor
PD	Phase Detector
PFD	Phase Frequency Detector
PLL	Phase Locked Loop
PMOS	P-Type Metal Oxide Semiconductor
RHBD	Radiation Hardened By Design
SEE	Single Event Effect
SET	Single Event Transient
SEU	Single Event Upset
TMR	Triple Modular Redundancy
TPA	Two Photon Absorption
VCDL	Voltage Controlled Delay Line
VCO	Voltage Controlled Oscillator
VCP	Voltage Charge Pump

APPENDIX-B: Linear Model Demonstration

The relation between the input and output phases of an analog DLL is:

$$\phi_{out} = \phi_{in} + \phi_d = \phi_{in} + 2\pi * \frac{t_d}{T_{clk}} \qquad \text{(1-B)}$$

With: $-\pi \leq t_d \leq +\pi$ and where t_d is the DLL's addition al delay at a given time t and where Φ_{out} is the output phase of the circuit, Φ_{in} is the input phase, Φ_d is the initial phase error due to the input clock skew (related to the input delay t_d) and T_{clk} is the period of the DLL's input clock.

The gain of the phase detector and charge pump, K_D is:

$$K_D = \frac{\Delta I_p}{\Delta \phi_{out}} = \frac{2 * I_p}{2 * \pi} = \frac{I_p}{\pi} \qquad \text{(2-B)}$$

Where I_p is the charge pump current and $\Delta\Phi_{out}$ is the DLL output phase variation.

The gain of the first order loop filter capacitor K_{LF} is:

$$K_{LF}(s) = \frac{1}{sC_{LF}} \qquad \text{(3-B)}$$

Where C_{LF} is the loop filter capacitor and s is the Laplace variable (integrator).

The gain of the voltage controlled delay line K_v is:

$$K_v = \frac{\Delta t_d}{\Delta V_{ctrl}} = \frac{t_d}{V_{ctrl}} \qquad \text{(4-B)}$$

Where V_{ctrl} is the control voltage of the VCDL and t_d is the delay introduce in the DLL.

The transfer function of an analog mixed-signal DLL H(s) is given by:

$$H(s) = \frac{\phi_{out}}{\phi_{in}} = \frac{1}{1 + \dfrac{1}{K_D * K_{LF}(s) * K_v * \omega_{clk}}} \tag{5-B}$$

Including Equations (2-B), (3-B) and (4-B) in (5-B), the transfer function become:

$$H(s) = \frac{\phi_{out}}{\phi_{in}} = \frac{1}{1 + \dfrac{1}{\dfrac{I_p}{\pi} * K_v * \dfrac{2\pi}{sC_{LF}} * \dfrac{1}{T_{clk}}}} \tag{6-B}$$

$$H(s) = \frac{1}{1 + \dfrac{s}{\dfrac{K_v}{T_{clk}} * \dfrac{2I_p}{sC_{LF}}}} \tag{7-B}$$

Where K_v is the gain of the VCDL, K_{LF} is the gain of the loop filter, K_D is the gain of the PD and CP, T_{clk} is the input clock period of the DLL, I_p is the charge pump current, s is the Laplace variable and C_{LF} is the loop filter capacitor.

General expression of the recovery time (or locking time of the DLL):

With the loop filter in steady state mode, C_{LF} is a constant. Therefore:

$$\Delta V_{ctrl} = \frac{\Delta t_d}{K_v} = 2\pi * \frac{\Delta t_d}{T_{clk}} * \frac{1}{C_{LF}} * K_D \qquad (8\text{-B})$$

If we assume that the voltage perturbation V_e is the variation of the control voltage V_{ctrl} and T_{rec} is the variation of the DLL output delay over time, then:

$$V_e = \frac{2 * I_p}{C_{LF}} * T_{rec} \qquad (9\text{-B})$$

From Equation 10-B the amount of time it requires for an analog DLL to lock, or recover from a single event transient perturbation, T_{rec}, is then:

$$T_{rec} = \frac{2 * I_p}{C_{LF}} * V_e \qquad (10\text{-B})$$

Where I_p is the charge pump current, and C_{LF} is the loop filter capacitor.

Voltage perturbation generated by the phase detector:

An ion strike in the phase detector will be translated by a shift in the DLL's output delay referred as t_e. Therefore the voltage perturbation V_e in the phase detector is:

$$V_e = 2\pi * t_e * \frac{K_D}{C_{LF}} \qquad (11\text{-B})$$

By replacing K_D by its expression in Equation (2-B),

$$V_e = 2\pi * t_e * \frac{I_p}{\pi} * \frac{1}{C_{LF}} = 2 * t_e * \frac{I_p}{C_{LF}} \qquad (12\text{-B})$$

Since the ion strike can affect the output UP or DOWN of the PD, the voltage perturbation of the PD is defined by two equations.

$$V_e \ (UP) = + \ 2 * t_e * \frac{I_p}{C_{LF}}, for \ the \ utput \ UP \ of \ the \ PD \qquad \text{(13-B)}$$

$$V_e \ (DOWN) = - \ 2 * t_e * \frac{I_p}{C_{LF}}, for \ the \ output \ DOWN \ of \ the \ PD \qquad \text{(14-B)}$$

Voltage perturbation generated by the charge pump:

An ion strike in the charge pump will directly affect the control voltage V_{ctrl} of the VCDL because the CP is directly connected to the loop filter capacitor that generates V_{ctrl}. Therefore, the charge deposited in the CP will add or remove the charge from the loop filter capacitor C_{LF}. In steady state mode:

$$Total \ charge \ \varDelta Q_e = C_{LF} * \varDelta V_{ctrl}, with \ \varDelta V_{ctrl} = Ve \ \rightarrow \ Ve = \frac{\varDelta Q_e}{C_{LF}} \qquad \text{(15-B)}$$

Since the charge deposited in the charge pump, Q_e, directly impact the control voltage of the DLL, ΔQ_e will represent the sum of the charge deposited in the CP and the charge induced in the activation of the CP for the duration of the error t_e.

$$\varDelta Q_e = Q_e + I_p * t_e \qquad \text{(16-B)}$$

By adding the expression of ΔQ_e in the expression of the CP voltage perturbation V_e in (15-B), the control voltage error V_e that is generated by the CP is given by the Equation (17-B),

where C_{LF} is the loop filter capacitor, I_p is the charge pump current and t_e the output delay generated by the ion strike.

$$V_e = \frac{\Delta Q_e}{C_{LF}} = \frac{(Q_e + I_p * t_e)}{C_{LF}} \qquad \text{(17-B)}$$

Voltage perturbation generated by the voltage controlled delay line:

The voltage perturbation V_e generated following an ion strike in the VCDL, will require that the delay error generated by the single event transient will be converted into a phase error and integrated by the charge pump and low pass filter. This means that the single event transient will propagate through the phase detector and the charge pump. Therefore in steady state mode, V_e of the VCDL is given by:

$$V_e = 2\pi * t_e * \frac{I_p}{\pi} * \frac{1}{C_{LF}} = \frac{2 * I_p}{C_{LF}} * t_e \qquad \text{(18-B)}$$

Where I_p is the current of the charge pump, t_e is the delay error generated by the ion strike and C_{LF} is the loop filter capacitor.

APPENDIX-C: VHDL program for Error Correction Circuit Test Bench

Schematic block of the ECC test bench

```vhdl
--   \___\/\___\
--
--Command: sch2hdl -intstyle ise -family virtex5 -flat -suppress -vhdl
C:/Users/pierrem/Desktop/Data-6613/ECC/FEECC.vhf -w
C:/Users/pierrem/Desktop/Data-6613/ECC/FEECC.sch
--Design Name: FEECC
--Device: virtex5
--Purpose:
--    This vhdl netlist is translated from an ECS schematic. It can be
--    synthesized and simulated, but it should not be modified.
--


library ieee;
use ieee.std_logic_1164.ALL;
use ieee.numeric_std.ALL;
library UNISIM;
use UNISIM.Vcomponents.ALL;


entity ECCb_MUSER_FEECC is
   port ( cout   : inout std_logic;
          op1    : inout std_logic;
          op2    : inout std_logic;
          Refclk : inout std_logic;
          sel    : inout std_logic;
          X1     : inout std_logic;
          X2     : inout std_logic);
```

```vhdl
end ECCb_MUSER_FEECC;


architecture BEHAVIORAL of ECCb_MUSER_FEECC is

    signal XLXN_37 : std_logic;

    component XORG

        port ( a    : in    std_logic;

               b    : in    std_logic;

               xout : inout std_logic);

    end component;


    component MUX2

        port ( a    : in    std_logic;

               b    : in    std_logic;

               sel  : in    std_logic;

               mout : inout std_logic);

    end component;


    component AND2

        port ( a    : in    std_logic;

               b    : in    std_logic;

               aout : inout std_logic);

    end component;


    component INV1

        port ( a : in    std_logic;

               b : inout std_logic);
```

```vhdl
    end component;


begin
    XLXI_4 : XORG

        port map (a=>Refclk,

                  b=>op2,

                  xout=>X2);


    XLXI_5 : XORG

        port map (a=>Refclk,

                  b=>op1,

                  xout=>X1);


    XLXI_7 : MUX2

        port map (a=>op1,

                  b=>op2,

                  sel=>sel,

                  mout=>cout);


    XLXI_8 : AND2

        port map (a=>X1,

                  b=>XLXN_37,

                  aout=>sel);


    XLXI_9 : INV1

        port map (a=>X2,
```

```vhdl
                    b=>XLXN_37);



end BEHAVIORAL;




library ieee;

use ieee.std_logic_1164.ALL;

use ieee.numeric_std.ALL;

library UNISIM;

use UNISIM.Vcomponents.ALL;


entity FEM_MUSER_FEECC is

    port ( Emp   : inout std_logic;

           op1   : inout std_logic;

           op2   : inout std_logic;

           outp1 : inout std_logic;

           outp2 : inout std_logic;

           sei1  : inout std_logic;

           sei2  : inout std_logic);

end FEM_MUSER_FEECC;


architecture BEHAVIORAL of FEM_MUSER_FEECC is

    component MUX2

       port ( a    : in    std_logic;

              b    : in    std_logic;
```

```vhdl
                 sel  : in    std_logic;

                 mout : inout std_logic);

    end component;


begin

    XLXI_1 : MUX2

        port map (a=>op1,

                    b=>Emp,

                    sel=>sei1,

                    mout=>outp1);


    XLXI_2 : MUX2

        port map (a=>op2,

                    b=>Emp,

                    sel=>sei2,

                    mout=>outp2);


end BEHAVIORAL;



library ieee;

use ieee.std_logic_1164.ALL;

use ieee.numeric_std.ALL;

library UNISIM;

use UNISIM.Vcomponents.ALL;
```

```vhdl
entity FEECC is

   port ( Glock : in    std_logic;

          RST   : in    std_logic;

          CL31  : out   std_logic;

          MPL   : out   std_logic;

          OP1   : out   std_logic;

          OP2   : out   std_logic;

          Rclk  : out   std_logic;

          SelL  : out   std_logic;

          Vout  : out   std_logic;

          X1L   : out   std_logic;

          X2L   : out   std_logic;

          MP    : inout std_logic;

          Slock : inout std_logic);

end FEECC;


architecture BEHAVIORAL of FEECC is

   attribute BOX_TYPE              : string ;

   attribute CLK_FEEDBACK          : string ;

   attribute CLKDV_DIVIDE          : string ;

   attribute CLKFX_DIVIDE          : string ;

   attribute CLKFX_MULTIPLY        : string ;

   attribute CLKIN_DIVIDE_BY_2     : string ;

   attribute CLKIN_PERIOD          : string ;

   attribute CLKOUT_PHASE_SHIFT    : string ;
```

```vhdl
attribute DCM_PERFORMANCE_MODE  : string ;

attribute DESKEW_ADJUST         : string ;

attribute DFS_FREQUENCY_MODE    : string ;

attribute DLL_FREQUENCY_MODE    : string ;

attribute DUTY_CYCLE_CORRECTION : string ;

attribute FACTORY_JF            : string ;

attribute PHASE_SHIFT           : string ;

attribute STARTUP_WAIT          : string ;

attribute DCM_AUTOCALIBRATION   : string ;

signal Sel1    : std_logic;

signal Sel2    : std_logic;

signal XLXN_19 : std_logic;

signal XLXN_20 : std_logic;

signal XLXN_24 : std_logic;

signal XLXN_25 : std_logic;

signal XLXN_29 : std_logic;

signal XLXN_30 : std_logic;

signal XLXN_35 : std_logic;

signal XLXN_37 : std_logic;

signal XLXN_52 : std_logic;

signal XLXN_61 : std_logic;

signal XLXN_64 : std_logic;

component FEM_MUSER_FEECC

   port ( Emp   : inout std_logic;

          sei2  : inout std_logic;

          sei1  : inout std_logic;
```

```vhdl
            op1   : inout std_logic;

            op2   : inout std_logic;

            outp1 : inout std_logic;

            outp2 : inout std_logic);

end component;



component BUF

   port ( I : in    std_logic;

          O : out    std_logic);

end component;

attribute BOX_TYPE of BUF : component is "BLACK_BOX";



component INV1

   port ( a : in    std_logic;

          b : inout std_logic);

end component;



component ECCb_MUSER_FEECC

   port ( op1    : inout std_logic;

          op2    : inout std_logic;

          Refclk : inout std_logic;

          cout   : inout std_logic;

          X1     : inout std_logic;

          X2     : inout std_logic;

          sel    : inout std_logic);

end component;
```

```vhdl
component DCM_BASE

  -- synopsys translate_off

  generic( CLK_FEEDBACK : string :=  "1X";

             CLKDV_DIVIDE : real :=  2.0;

             CLKFX_DIVIDE : integer :=  1;

             CLKFX_MULTIPLY : integer :=  4;

             CLKIN_DIVIDE_BY_2 : boolean :=  FALSE;

             CLKIN_PERIOD : real :=  10.0;

             CLKOUT_PHASE_SHIFT : string :=  "NONE";

             DCM_PERFORMANCE_MODE : string :=  "MAX_SPEED";

             DESKEW_ADJUST : string :=  "SYSTEM_SYNCHRONOUS";

             DFS_FREQUENCY_MODE : string :=  "LOW";

             DLL_FREQUENCY_MODE : string :=  "LOW";

             DUTY_CYCLE_CORRECTION : boolean :=  TRUE;

             FACTORY_JF : bit_vector :=  x"F0F0";

             PHASE_SHIFT : integer :=  0;

             STARTUP_WAIT : boolean :=  FALSE;

             DCM_AUTOCALIBRATION : boolean :=  TRUE);

  -- synopsys translate_on

  port ( CLKIN    : in    std_logic;

          CLKFB    : in    std_logic;

          RST      : in    std_logic;

          CLK0     : out   std_logic;

          CLK90    : out   std_logic;

          CLK180   : out   std_logic;
```

```vhdl
            CLK270   : out   std_logic;

            CLK2X    : out   std_logic;

            CLK2X180 : out   std_logic;

            CLKDV    : out   std_logic;

            CLKFX    : out   std_logic;

            CLKFX180 : out   std_logic;

            LOCKED   : out   std_logic);

    end component;

    attribute CLK_FEEDBACK of DCM_BASE : component is "1X";

    attribute CLKDV_DIVIDE of DCM_BASE : component is "2.0";

    attribute CLKFX_DIVIDE of DCM_BASE : component is "1";

    attribute CLKFX_MULTIPLY of DCM_BASE : component is "4";

    attribute CLKIN_DIVIDE_BY_2 of DCM_BASE : component is "FALSE";

    attribute CLKIN_PERIOD of DCM_BASE : component is "10.0";

    attribute CLKOUT_PHASE_SHIFT of DCM_BASE : component is "NONE";

    attribute DCM_PERFORMANCE_MODE of DCM_BASE : component is

"MAX_SPEED";

    attribute DESKEW_ADJUST of DCM_BASE : component is

"SYSTEM_SYNCHRONOUS";

    attribute DFS_FREQUENCY_MODE of DCM_BASE : component is "LOW";

    attribute DLL_FREQUENCY_MODE of DCM_BASE : component is "LOW";

    attribute DUTY_CYCLE_CORRECTION of DCM_BASE : component is "TRUE";

    attribute FACTORY_JF of DCM_BASE : component is "F0F0";

    attribute PHASE_SHIFT of DCM_BASE : component is "0";

    attribute STARTUP_WAIT of DCM_BASE : component is "FALSE";

    attribute DCM_AUTOCALIBRATION of DCM_BASE : component is "TRUE";
```

```vhdl
    attribute BOX_TYPE of DCM_BASE : component is "BLACK_BOX";


    component clk200Hz

        port ( clk_in  : in    std_logic;

                reset   : in    std_logic;

                clk_out : out   std_logic);

    end component;


    component clk200Hz_2

        port ( clk_in  : in    std_logic;

                reset   : in    std_logic;

                clk_out : inout std_logic);

    end component;


begin

    XLXI_2 : FEM_MUSER_FEECC

        port map (Emp=>MP,

                    op1=>XLXN_64,

                    op2=>XLXN_64,

                    outp1=>XLXN_20,

                    outp2=>XLXN_19,

                    sei1=>Sel1,

                    sei2=>Sel2);


    XLXI_13 : BUF

        port map (I=>XLXN_24,
```

```
                    O=>SelL);


    XLXI_14 : BUF

        port map (I=>XLXN_19,

                    O=>OP2);


    XLXI_15 : BUF

        port map (I=>XLXN_25,

                    O=>Vout);


    XLXI_16 : BUF

        port map (I=>XLXN_20,

                    O=>OP1);


    XLXI_17 : BUF

        port map (I=>XLXN_37,

                    O=>Rclk);


    XLXI_18 : BUF

        port map (I=>MP,

                    O=>MPL);


    XLXI_21 : INV1

        port map (a=>XLXN_64,

                    b=>XLXN_35);
```

```
XLXI_24 : ECCb_MUSER_FEECC

   port map (cout=>XLXN_25,

             op1=>XLXN_20,

             op2=>XLXN_19,

             Refclk=>XLXN_37,

             sel=>XLXN_24,

             X1=>XLXN_30,

             X2=>XLXN_29);


XLXI_25 : BUF

   port map (I=>XLXN_29,

             O=>X2L);


XLXI_26 : BUF

   port map (I=>XLXN_30,

             O=>X1L);


XLXI_27 : INV1

   port map (a=>XLXN_35,

             b=>XLXN_37);


XLXI_40 : DCM_BASE

   port map (CLKFB=>XLXN_52,

             CLKIN=>Glock,

             RST=>RST,

             CLKDV=>open,
```

```vhdl
                    CLKFX=>open,

                    CLKFX180=>open,

                    CLK0=>XLXN_64,

                    CLK2X=>XLXN_61,

                    CLK2X180=>open,

                    CLK90=>open,

                    CLK180=>open,

                    CLK270=>open,

                    LOCKED=>open);


XLXI_41 : BUF

   port map (I=>XLXN_64,

             O=>XLXN_52);


XLXI_43 : BUF

   port map (I=>XLXN_61,

             O=>Slock);


XLXI_45 : BUF

   port map (I=>Sel2,

             O=>CL31);


XLXI_46 : clk200Hz

   port map (clk_in=>XLXN_64,

             reset=>RST,

             clk_out=>Sel2);
```

```vhdl
   XLXI_48 : clk200Hz_2

      port map (clk_in=>XLXN_64,

                reset=>RST,

                clk_out=>Sel1);


end BEHAVIORAL;
```

-- Copyright (c) 1995-2012 Xilinx, Inc. All rights reserved.

```
--------------------------------------------------------------------------

---------

--    ____  ____

-- /    /\/   /

-- /___/  \  /     Vendor: Xilinx

-- \   \   \/      Version : 14.5

--  \   \          Application : sch2hdl

--  /   /          Filename : ECCb.vhf

-- /___/   /\      Timestamp : 06/12/2013 17:02:16

-- \   \  /  \

--  \___\/\___\

--

--Command: sch2hdl -intstyle ise -family virtex5 -flat -suppress -vhdl

C:/Users/pierrem/Desktop/Data-6613/ECC/ECCb.vhf -w

C:/Users/pierrem/Desktop/Data-6613/ECC/ECCb.sch

--Design Name: ECCb

--Device: virtex5

--Purpose:

--    This vhdl netlist is translated from an ECS schematic. It can be

--    synthesized and simulated, but it should not be modified.

--


library ieee;

use ieee.std_logic_1164.ALL;

use ieee.numeric_std.ALL;

library UNISIM;
```

```vhdl
use UNISIM.Vcomponents.ALL;


entity ECCb is

    port ( cout   : inout std_logic;

           op1    : inout std_logic;

           op2    : inout std_logic;

           Refclk : inout std_logic;

           sel    : inout std_logic;

           X1     : inout std_logic;

           X2     : inout std_logic);

end ECCb;


architecture BEHAVIORAL of ECCb is

    signal XLXN_37 : std_logic;

    component XORG

       port ( a    : in    std_logic;

              b    : in    std_logic;

              xout : inout std_logic);

    end component;


    component MUX2

       port ( a    : in    std_logic;

              b    : in    std_logic;

              sel  : in    std_logic;

              mout : inout std_logic);

    end component;
```

156

```vhdl
    component AND2

      port ( a    : in    std_logic;

             b    : in    std_logic;

             aout : inout std_logic);

    end component;


    component INV1

      port ( a : in    std_logic;

             b : inout std_logic);

    end component;


begin

  XLXI_4 : XORG

      port map (a=>Refclk,

                b=>op2,

                xout=>X2);


  XLXI_5 : XORG

      port map (a=>Refclk,

                b=>op1,

                xout=>X1);


  XLXI_7 : MUX2

      port map (a=>op1,

                b=>op2,
```

```
                        sel=>sel,

                        mout=>cout);



    XLXI_8 : AND2

       port map (a=>X1,

                    b=>XLXN_37,

                    aout=>sel);



    XLXI_9 : INV1

       port map (a=>X2,

                    b=>XLXN_37);



end BEHAVIORAL;
```

-- Company: Vanderbilt - Xilinx

-- Engineer: Pierre Maillard

--

-- Create Date: 21:31:50 06/11/2013

-- Design Name:

-- Module Name: AND2 - Behavioral

-- Project Name:

-- Target Devices:

-- Tool versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

----

library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating

-- any Xilinx primitives in this code.

--library UNISIM;

```
--use UNISIM.VComponents.all;


entity AND2 is

  Port ( a : in  STD_LOGIC;

        b : in  STD_LOGIC;

        aout : inout  STD_LOGIC);

  end AND2;
```

-- Company: Vanderbilt-Xilinx

-- Engineer: Pierre Maillard

--

-- Create Date: 15:35:36 06/12/2013

-- Design Name:

-- Module Name: COUNT_UPSET - Behavioral

-- Project Name:

-- Target Devices:

-- Tool versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

-- 111110100 500

----

library IEEE;

```vhdl
use IEEE.STD_LOGIC_1164.ALL;


entity clk200Hz is

    Port (

        clk_in : in  STD_LOGIC;

        reset  : in  STD_LOGIC;

        clk_out: out STD_LOGIC

    );

end clk200Hz;


architecture Behavioral of clk200Hz is

    signal temporal: STD_LOGIC;

    signal counter : integer range 0 to 100 := 0;

begin

    frequency_divider: process (reset, clk_in) begin

        if (reset = '1') then

            temporal <= '0';

            counter <= 0;

        elsif rising_edge(clk_in) then

            if (counter = 100) then

                temporal <= NOT(temporal);

                counter <= 0;

            else

                counter <= counter + 1;
```

```vhdl
        end if;

      end if;

    end process;


    clk_out <= temporal;

end Behavioral;
```

```vhdl
-- Company:  Vanderbilt-Xilinx

-- Engineer: Pierre Maillard
```

```
--
-- Create Date:    20:27:19 06/11/2013
-- Design Name:
-- Module Name:    MUX2 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;


-- Uncomment the following library declaration if instantiating
```

-- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity MUX2 is

  Port (a : in STD_LOGIC;

     b : in STD_LOGIC;

     sel : in STD_LOGIC;

     mout : inout STD_LOGIC);

end MUX2;

architecture Behavioral of MUX2 is

begin

      process(sel, a, b)

         begin

            if (sel='0') then

                  mout <= a;

            else

                  mout <= b;

            end if;

         end process;

end Behavioral;

```vhdl
----------------------------------------------------------------------------------
-- Company:  Vanderbilt-Xilinx

-- Engineer: Pierre Maillard

--

-- Create Date:    21:37:49 06/11/2013

-- Design Name:

-- Module Name:    XOR2 - Behavioral

-- Project Name:

-- Target Devices:

-- Tool versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

----------------------------------------------------------------------------------
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;


-- Uncomment the following library declaration if using
```

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating

-- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity XORG is

  Port (a : in STD_LOGIC;

     b : in STD_LOGIC;

     xout : inout STD_LOGIC);

end XORG;

architecture Behavioral of XORG is

begin

       xout <= a xor b;

end Behavioral;

APPENDIX-D: Python Code for Phase Displacement Error Measurement

#!//usr/bin/python

```
################################################################################

#to run: call python getFreq filename clkcolNumber vectorcolNumber startTime#

################################################################################

#Description: this script reads a peridoc clock signal, and DLL (vector)    #

#output and determines the frequency, number of pulses, and phase disp.    #

#Author: Daniel Loveless (daniel.loveless@vanderbilt.edu)                #

#Author: Pierre Maillard (pierre.maillard@vanderbilt.edu)                #

#Last Update: 5/18/2013                                          #
################################################################################
import sys, csv

datfile=sys.argv[1]        #filename

clockvec=sys.argv[2]        #column in file corresponding to the clock signal

vecnum=sys.argv[3]         #column in file corresponding to the output vector

starttime=float(sys.argv[4]) #start time for data comparison

print "\n" + datfile


thresh=800 #threshold voltage (0.5 VDD) for determining rising clock edges

tcol=0    #column in vector corresponding to the timestamps

#################
```

```
#import the file#

##################

IOflag=0

try:

    infile=open(datfile,'r')

    line=infile.readlines()

    infile.close()

except IOError, (errno, strerror):

    IOflag=1

  #print "I/O error(%s): %s" % (errno, strerror)

if IOflag==0:

 #####################

    #grab the waveforms#

    #####################

    vec=[]    #vector for output data points

    cvec=[]   #vector for clock data points

    tvec=[]   #vector for time data points

    temp=0    #temp for vec

    clktemp=0 #temp for cvec

    ttemp=0   #temp for tvec


#    print vecnum

#    print len(line)

 #loop through the length of line (number of vectors in file)
```

```python
for i in range(len(line)):

    a=[] #vector to hold data from line

    a=line[i].split()

    #place appropriate data points in file in temps

    temp=float(a[int(vecnum)])

    clktemp=float(a[int(clockvec)])

    ttemp=float(a[int(tcol)])

    #append temp value to appropriate vectors

    vec.append(temp)

    cvec.append(clktemp)

    tvec.append(ttemp)
```

###

```python
#find the rising edge of each clock pulse (used to determine the frequency)
```

###

```python
flag=0    #flag for det if the last value was above or below threshold

rise=0    #variable describing current value

count=0    #counter incrementing every time a rising edge occurs


tchk=[]    #vector for timestamps of each rising edge (50% point)

numpulses=0 #the number of pulses (rising edges)

for j in range(1,len(cvec)):      #loop through the clock vector
```

```python
            rise=cvec[j]                #assign the current value in vector

        if rise>=thresh:                #if the value is above the threshold

            if flag==0:                 #and if the last value was below the th

                count=count      #then incremnent the counter

                tchk.append(tvec[j]) #and record the timestamp

            flag=1                      #flag=1 when the threshold is exceeded

        else:                           #if the value is below the threshold

            flag=0                      #set the flag back to 0

    numpulses=count                     #the number of pulses

    print "Total number of clock pulses:\t" + str(numpulses)

    #print tchk


    ##############################################################################


    #               find the time difference between each timestamp              #
    ##############################################################################
    tdiff=[] #vector for the time differences between current and previous

    diff=0   #variable for time difference for current

    for k in range(2,count):    #loop through the timestamp vector


        diff=tchk[k]-tchk[k-1] #the time difference between pulses

        tdiff.append(diff)     #append the current value to a new vector

    ##############################################################################
```

#average the resulting time diffs to get the per and freq (of clock data) #

##

sum=0   #sum of all time differences

count=0   #counter (total number of differences)

per=0   #the period of the signal

freq=0   #the frequency of the signal (units Hz)

freqMHz=0 #the frequency in units (MHz)

for l in range(1,len(tdiff)): #loop through the vector of time differences

    sum=tdiff[l]+sum   #increment the sum of the values

print len(tdiff)

count=len(tdiff)   #the total number of values

per=sum/count   #the per is the avg of all of the time diffs

freq=1/(per*1e-9)   #frequency=1/period (Hz)

freqMHz=round(freq/1e6)   #convert frequency into MHz

print "Clock Frequency:\t" + str(freqMHz) + " MHz"

##

find the rising edge of each PLL output pulse

##

flag=0   #same as previous

rise=0   #same as previous

count=0   #counter for the total number of pulses

countst=0   #count for the number of pulses following the start time

```python
tchk=[]          #vector for holding the timestamps of the pulses

index=0          #count of the first pulse following the start time

for m in range(1,len(vec)):      #loop through the vector of output data

    rise=vec[m]               #set rise to current value in vector

    if rise>=thresh:          #if the value exceeds the threshold

        if flag==0:           #and the previous value was below th

            count=count+1     #increment the number of pulses

            tchk.append(float(tvec[m]))   #record the current timestamp

            if float(tvec[m])>=starttime: #greater than the startime

                if index==0:          #and the 1st one

                    index=count       #record which pulse number

                countst=countst+1     #increment

        flag=1                        #threshold is exceeded

    else:                             #value is below the threshold

        flag=0                        #set the flag to 0

print "Total number of vector pulses:\t" + str(count)

print "Total number of vector pulses after " + str(starttime) + "s:\t" + str(countst)

#############################################################################
#       find the time difference and the jitter of the signal        #
#############################################################################

tdiff=[]  #vector for holding the time differences

diff=0    #value of current difference

sum=0     #sum of all time time differences

sumd=0    #sum of residues (sum of (current tdiff-mean tdiff))
```

173

```python
jitter=0  #the average of the residues

sqrd=0    #the sum of the sqaure of the residues

std=0     #standard deviation of the sum of the squares

jitVal=0

jitVec=[]

for k in range(index+1,count):     #loop through the number of pulses

    diff=tchk[k]-tchk[k-1]         #find the time difference

    tdiff.append(diff)             #append the current tdiff to vectr

for k in range(1,len(tdiff)):      #loop through the time differences

    sum=sum+tdiff[k]               #increment the sum of the differences

mean=sum/len(tdiff)                #find the average time difference

for k in range(1,len(tdiff)):      #loop through the time differences

    jitVal=tdiff[k]-mean

    jitVec.append(jitVal)

    sumd=(tdiff[k]-mean)+sumd   #increment the sum of the residuals

    sqrd=((tdiff[k]-mean)**2)+sqrd #increment the sum of the squares

print tchk

#print jitVec

jitter=float(sumd/len(tdiff))      #average jitter

sqrd=float(sqrd/len(tdiff))        #average of the sum of the squares

std=sqrd**(0.5)                    #standard deviation

jitter=jitter              #average jitter in ps

std=std                    #standard deviation of jitter in ps

print "Jitter:\t" + str(jitter) + " ns +- " + str(std) + " ns\n"
```

```
for k in range(0,len(tdiff)-1):

    print str(tchk[k]) + "\t" + str(jitVec[k])

fout = open(datfile + "_r.tsv", "w")

writer = csv.writer(fout, delimiter="\t")

writer.writerows(zip(tchk,jitVec))

fout.close()
```

REFERENCES

1. P. Maillard, "Radiation-hardened-by-design (RHBD) delay locked loops (DLLs): single event transient analysis, simulation, and hardening", MSEE, Vanderbilt University, Mar. 2010.

2. P. Maillard, W. T. Holman, T. D. Loveless, B. L. Bhuva and L. W. Massengill, "An RHBD technique to mitigate missing pulses in DLLs", IEEE Trans. Nucl. Sci., vol. 57, no. 6, pp. 3634-3639, Dec. 2010.

3. Y. Moon, J. Choi, K. Lee, D. K. Jeong, and M. K. Kim, "An all-analog multiphase delay-locked loop using a replica delay line for wide-range operation and low-jitter performance," IEEE J. Solid-State Circuits, vol. 35, no. 3, pp. 377–384, Mar. 2000.

4. A. Hatakeyama, H. Mochizuki, T. Aikawa, M. Takita, Y. Ishii, H. Tsuboi, S. Fujioka, S. Yamaguchi, M. Koga, Y. Serizawa, K. Nishimura, K. Kawabata, Y. Okajima, M. Kawano, H. Kojima, K. Mizutani, T. Anezaki, M. Hasegawa, and Masao Taguchi, "A 256-Mb SDRAM Using a Register Controlled Digital DLL", IEEE J. Solid-State Circuits, vol. 32, no. 11, 1997.

5. S. Sidiropoulos and M. Horowitz, "A Semi-digital Dual Delay Locked Loop", IEEE J. Solid-State Circuits, vol. 32, no. 11, pp. 1683-1692, Nov. 1997.

6. G. Jovanović, M. Stojčev, "Voltage Controlled Delay Line for Digital Signal", Facta Universitatis, Series: Electronics and Energetic, vol. 16. no. 2, pp. 215-232, Aug. 2003.

7. K. Bult, H.Wallinga, "A Class of Analog CMOS Circuits Based on the Square-Law Characteristic of an MOS Transistor in Saturation", IEEE J. Solid-State Circuits, vol. 22, No.3, pp. 357-365, June 1987.

8. R. Sengupta, B. Vermeire, L. T. Clark, and B. Bakkaloglu, "A 133 MHz Radiation-Hardened Delay-Locked Loop", IEEE Trans. Nucl. Sci., vol. 57, no. 6, pp. 3626_3633, Dec. 2010.

9. L. W. Massengill, "SEU Modeling and Prediction Techniques", IEEE NSREC Short Course Text, 1993, pp. III-1–III-93, Sept. 1993.

10. P. E. Dodd and L. W. Massengill, "Basic Mechanisms and Modeling of Single-Event Upset in Digital Microelectronics", IEEE Trans. Nucl. Sci , Vol. 50, No. 3, pp. 583-602, June 2003.

11. D. McMorrow, W. T. Lotshaw, J. S. Melinger, S. Buchner, Y. Boulghassoul, L. W. Massengill, and R. Pease, "Three-dimensional mapping of single-event effects using two photon absorption", IEEE Trans. Nucl. Sci , vol. 50, no. 6, pp. 2199–2207, Dec. 2003.

12. Y. Boulghassoul, L. W. Massengill, A. L. Sternberg, B. L. Bhuva and W. T. Holman, "Towards SET Mitigation in RF Digital PLLs: From Error Characterization to Radiation Hardening Considerations," RADECS, IEEE Conf. proceeding on, Sept. 2005.

13. Y. Boulghassoul, L. W. Massengill, A. L. Sternberg, and B. L. Bhuva, "Effects of technology scaling on the SET sensitivity of RF CMOS voltage-controlled oscillators," IEEE Trans. Nucl. Sci., vol. 52, no. 6, pp. 2426–2432, Dec. 2005.

14. T. D. Loveless, "A radiation-hardened-by-design charge pump for phase-locked loop circuits", MSEE, Vanderbilt University, May 2007.

15. T. D. Loveless, L. W. Massengill, B. L. Bhuva, W. T. Holman, A. F. Witulski, and Y. Boulghassoul, "A hardened-by-design technique for RF digital phase-locked loops," IEEE Trans. Nucl. Sci., vol. 53, no. 6, pp. 3432–3438, Dec. 2006.

16. T. D. Loveless, L. W. Massengill, B. L. Bhuva, W. T. Holman, M. C. Casey, R. A. Reed, S. A. Nation, D. McMorrow, J. S. Melinger, "A Probabilistic Analysis Technique Applied to a Radiation-Hardened-by-Design Voltage-Controlled Oscillator for Mixed-Signal Phase-Locked Loops," in IEEE Trans. Nucl. Sci., vol. 55, no 6, pp. 3447-3455, Dec. 2008

17. J. S. Kauppila, A. L. Sternberg, M. L. Alles, A. M. Francis, J. Holmes, O. A. Amusan, and L. W. Massengill, "A Bias-Dependent Single-Event Compact Model Implemented Into BSIM4 and a 90 nm CMOS Process Design Kit", IEEE Trans. Nucl. Sci., vol. 56, no. 6, pp. 3152-3157, Dec. 2009.

18. R. Jacob Baker, H.W. Li, D.E. Boyce, "CMOS Circuit Design, Layout, and Simulation 2nd Edition ", Wiley Press, IEEE Press Series on Microelectronic Systems Series, 2008.

19. R. E. Best, "Phase-Locked Loops: Design, Simulation, and Applications", McGraw-Hill, 2003

20. A. Rofourgan, J. Rael, M. Rofourgan, and A. Abidi, "A 900 MHz CMOS LC-oscillator with quadrature outputs," in IEEE Int. Solid-State Circuit Conf. Dig. Tech. Papers, pp. 392–393, Feb. 1996.

21. J. G. Maneatis, "Low-Jitter Process-Independent DLL and PLL Based on Self-Biased Techniques," IEEE Journal of Solid-State Circuits, vol. 31, no. 11, pp. 1723-1732, Nov. 1996.

22. D. McMorrow, W. T. Lotshaw, J. S. Melinger, S. Buchner, and R. L. Pease, "Sub-bandgap laser-induced single event effects: carrier generation via two-photon absorption," IEEE Trans. Nucl. Sci., vol. 49, no. 6, pp. 3002-3008, Dec. 2002.

23. C. Gagliano and R. J. Baker, "A Compact Delay-Locked Loop for Multi-Phase Non-Overlapping Clock Generation", 2010.

24. C. Chang, "A wide-range all-digital delay-locked loop in 65nm CMOS technology" VLSI Design Automation and Test (VLSI-DAT), 2010 International Symposium, 26-29 April 2010, pp. 66- 69.

25. A. Tajalli, P. Torkzadeh and M. Atarodi, "A Wide Tuning Range, Fractional Multiplying Delay-Locked Loop Topology for Frequency Hopping Applications", Analog Integrated Circuits And Signal Processing, Vol. 46, No. 3, pp. 203-214, 2006.

26. M. Xapsos, "Modeling the space radiation environment," in IEEE NSREC Short Course, pp. II.1-II.62, 2006.

27. L. W. Massengill and P. W. Tuinenga, "Single-event transient pulse propagation in digital CMOS," IEEE Trans. Nucl. Sci., vol.55, no. 6, pp. 2861- 2871, Dec. 2008.

28. F. Sturesson, "Single Event Effects (SEE) Mechanism and Effects", in IEEE RADECS Short Course 2003, June 2009.

29. F. Wrobel, J. M. Palau, M. C. Calvet, O. Bersillon, and H. Duarte, "Incidence of multi-particle events on soft error rates caused by n-Si nuclear reactions," IEEE Trans. Nucl. Sci., vol. 47, pp. 2580–2585, Dec. 2000.

30. R. Baumann, "Single-Event Effects in Advanced CMOS Technology," in Nuclear and Space Radiation Effects Conf. Short Course, Seattle, WA, 2005.

31. S. DasGupta, S. DasGupta, A. F. Witulski, B. L. Bhuva, M. L. Alles, R. A. Reed, O. A. Amusan, J. R. Ahlbin, R. D. Schrimpf and L. W. Massengill, "Effect of Well and Substrate Potential Modulation on Single Event Pulse Shape in Deep Submicron CMOS," IEEE Trans. On. Nucl. Sci., vol. 54, no. 6, pp. 2407–2412, Dec. 2007.

32. Y. J. Jung, S. W. Lee, D. Shim, W. Kim, and S. I. Cho, "A dual-loop delay-locked loop using multiple voltage-controlled delay lines," IEEE J. Solid-State Circuits, vol. 36, no. 5, pp. 784–791, May 2001.

33. C. C. Chung and C. Y. Lee, "A new DLL-based approach for all-digital multiphase clock generation," IEEE J. Solid-State Circuits, vol. 39, no. 3, pp. 469–475, Mar. 2004.

34. H. H. Chang, J. W. Lin, C. Y. Yang, and S. I. Liu, "A wide-range delay locked loop with a fixed latency of one clock cycle," IEEE J. Solid-State Circuits, vol. 37, no. 8, pp. 1021–1027, Aug. 2002.

35. T. Kim, S. H.Wang, and B. Kim, "Fast locking delay-locked loop using initial delay measurement," Electron. Letter, vol. 38, pp. 950–951, Aug. 2002.

36. K. Minami, M. Mizuno, H. Yamaguchi, T. Nakano, Y. Matsushima, Y. Sumi, T. Sato, H. Yamashida, and M. Yamashina, "A 1-GHz portable digital delay-locked loop with infinite phase capture ranges," IEICE Trans. Electron., vol. E84-C, pp. 220–228, Feb. 2001.

37. F. Lin, J. Miller, A. Schoenfeld, M. Ma, and R. J. Baker, "A register-controlled symmetrical DLL for double-data-rate DRAM," IEEE J. Solid-State Circuits, vol. 34, no. 4, pp. 565–568, Apr. 1999.

38. R. B. Watson, Jr. and R. B. Iknaian, "Clock buffer chip with multiple target automatic skew compensation," IEEE J. Solid-State Circuits, vol. 30, pp. 1267–1276, Nov. 1995.

39. C. H. Kim, J. H. Lee, J. B. Lee, B. S. Kim, C. S. Park, J. G. Roh, H. S. Nam, D. G. Kim, D. Y. Lee, T. S. Jung, H. Yoon, and S. I. Cho, "A 64-Mbit, 640-Mbyte/s bidirectional data strobed, double-data-rate SDRAM with a 40-mW DLL for a 256-Mbyte memory system," IEEE J. Solid-State Circuits, vol. 33, no. 11, pp. 1703–1710, Nov. 1998.

40. Z. Zhenyu, Z. Minxua, C. Shuming, C. Jihua and L. Junfeng, "A radiation hardened-by-design technique for improving single-event transient tolerance of charge pumps in PLLs", J. of Semiconductors, vol. 30, no. 12, Dec. 2009.

41. O. A. Amusan, L. W. Massengill, M. P. Baze, B. L. Bhuva, A. F. Witulski, J. D. Black, A. Balasubramanian, M. C. Casey, D. A. Black, J. R. Ahlbin, R. A. Reed, M. W. McCurdy, "Mitigation Techniques for Single-Event-Induced Charge Sharing in a 90-nm Bulk CMOS Process", IEEE Trans. on Nucl. Science, vol. 9, no. 2, pp. 309–317, June 2009.

42. D. Matsuura, K. Hirose, D. Kobayashi, S. Ishii, M. Kusano, Y. Kuroda, H. Saito, "Radiation-Hardened Phase-Locked Loop Fabricated in 200 nm SOI-CMOS", RADECS, IEEE conf. proceeding on, pp. 150- 155, Sept. 2011.

43. R. Kumar, V. Karkala, R. Garg, T. Jindal, S. P. Khatri, "A radiation tolerant Phase Locked Loop design for digital electronics", ICCP, IEEE conf. proceeding on, pp. 505 - 510, 2009.

44. D. Pan, W. L. Harry, I. Bogdan, M. Wilamowsk, "A Radiation-hard Phase-Locked Loop", IEEE International Symposium on Industrial Electronics, pp. 901-906, 2003.

45. S. Vishwanthaiah, "Dynamic termination output driver for a 600-MHz microprocessor," in IEEE Int. Solid-State Circuit Conf. Dig. Tech. Papers, Feb. 2000, pp. 248–249.

46. T. Matano, T. Takai, Y, T. Takahashi, Y. Sakito, I. Fujii, Y. Takaishi, H. Fujisawa, S. Kubouchi, S. Narui, K. Arai, M. Morino, M. Nakamura, S. Miyatake, T. Sekiguchi and K. Koyama "A 1-Gb/s/pin 512-Mb DDR II SDRAM using a digital DLL and a slew-rate-controlled output buffer," IEEE Journal of Solid-State Circuits, Vol. 38, No. 5, pp. 762-768, , May 2003.

47. T. H. Lee, K. Donnelly, J. Ho, J. Zerbe, M. Johnson, T. Ishikawa, "A 2.5-V delay-locked loop for an 18-Mb 500-MB/s DRAM," in IEEE Int. Solid-State Circuit Conf. Dig. Tech. Papers, pp. 300–301, Feb. 1994.

48. F. M. Gardner, "Charge-pump phase-lock loops," IEEE Trans. Communication, vol. 28, no. 11, pp. 1849-1858, Nov. 1980.

49. R. C. H. van de Beek, C. S. Vaucher, D. M. W. Lenaerts, E. A. M. Klumperink, B. Nauta, "A 2.5–10-GHz Clock Multiplier Unit With 0.22-ps RMS Jitter in Standard 0.18 um CMOS", IEEE Journal of Solid-State Circuits, vol. 39, no. 11, pp. 1872-1872, Nov. 2004.

50. G. Chien and P. R. Gray, "A 900-MHz local oscillator using a DLL-based frequency multiplier technique for PCS applications," IEEE J. Solid-State Circuits, vol. 35, pp. 1996–1999, Dec. 2000.

51. R. Farjad-rad, W. Dally, N. Hoik-Tiaq, J. Poulton, T. Stone, R. Rathi, E. Lee, D. Huang and R. Nathan, "A 0.2–2 GHz 12 mW multiplying DLL for low jitter clock synthesis in highly-integrated data communication chips," in IEEE Int. Solid-State Circuit Conf. Dig. Tech. Papers, Feb. 2002, pp. 76–77.

52. C. Y. Yang and S. I. Liu, "A One-Wire Approach for Skew-Compensating Clock Distribution Based on Bidirectional Techniques," in IEEE J. Solid-State Circuits, vo1.36, no. 6, pp. 266-272, Feb. 2001.

53. D. Birm,"A Novel Delay-Locked Loop Based CMOS Clock Multiplier." IEEE Trans. Consumer Electronics. vo1.44. no. 4, pp. l~19-1322, Nov. 1998.

54. T. Hamamoto, "A skew and jitter suppressed DLL architecture for high frequency DDR SDRAMs", in IEEE Symposium VLSI Circuits Dig. Tech. Papers, pp. 76–7, Jun. 2000.

55. R. C. H. van de Beek, E. A. M. Klumperink, C. S. Vaucher, and B. Nauta, "Low-jitter clock multiplication: a comparison between PLL's and DLLs," IEEE Trans. Circuits Syst. II, vol. 49, no. 8, pp. 555–566, Aug. 2002.

56. B. Garlepp, K. S. Donnely, J. Kim, P. S. Chau, J. L. Zerbe, C. Huang, C. V. Tran, C. L. Portman, D. Stark, Y. Chan, T. H. Lee, M. A. Horowitz, "A portable digital DLL for high-speed CMOS interface circuits," IEEE J. of Solid-State Circuits, vol. 34, no. 5, pp. 632-644, May 1999.

57. C. M. Hsieh, P. C. Murley, and R. R. O'Brien, "A Field-Funneling Effect on the Collection of Alpha-Particle-Generated Carriers in Silicon Devices," IEEE Electron Device Letters, vol. 2, no. 4, pp. 103–105, Apr. 1981.

58. O. A. Amusan, A. F. Witulski, L. W. Massengill, B. L. Bhuva, P. R. Fleming, M. L. Alles, A. L. Sternberg, J. D. Black, and R. D. Schrimpf, "Charge collection and charge sharing in a 130 nm CMOS technology," IEEE Trans. Nucl. Sci., vol. 53, no. 6, pp. 3253–3258, Dec. 2006.

59. S. DasGupta, "Trends in Single Event Pulse Widths and Pulse Shapes in Deep Submicron CMOS," MS Thesis, Vanderbilt University, Dec. 2007.

60. S. E. Kerns, L. W. Massengill, D. V. Kerns, Jr., M. L. Alles, T. W. Houston, H. Lu, and L. R. Hite, "Model for CMOS/SOI single-event vulnerability," IEEE Trans. Nucl. Sci., vol. 36, no. 6, pp. 2305–2310, Dec. 1989.

61. T. Saeki, Y. Nakaoka, M. Fujita, A. Tanaka, K. Nagata, K. Sakakibara, T. Matano, Y. Hoshino, K. Miyano, S. Isa, S. Nakazawa, E. Kakehashi, J. Mark Drynan, M. Komuro, T. Fukase, H. Iwasaki, .Y. Takenaka, J. Sekine, M. Igeta, N. Nakanishi, T. Itani, K. Yoshida, H. Yoshino, S. Hashimoto, T. Yoshii, M. ichinose, T. imura, M. Uziie, S. Kikuchi, K. Koyama, Y. Fukuzo, and T. Okuda., , "A 2.5-ns clock access, 250-MHz, 256-Mb SDRAM with synchronous mirror delay." IEEE J. of Solid-State Circuits, vol 31, no. 11, pp. 1656-1668, Nov. 1996.

62. E. Song, S. Lee, J. Lee, J. Park and S. Chae, "A reset-free anti-harmonic delay-locked loop using a cycle period detector," IEEE J. of Solid-State Circuits, vol. 39, pp. 2055-2061, Nov. 2004.

63. S. J. Kim, S. H. Hong, J. K. Wee, J. H. Cho, P. S. Lee, J. H. Ahn and J. Y. Chung, "A low-jitter wide-range skew-calibrated dual-loop DLL using antifuse circuitry for high-speed DRAM," IEEE J. of Solid-State Circuits, vol. 37, no. 6, pp. 726-734, Jun. 2002.

64. W. Kim, K. Kim, Y. Jeon, and S. Kim, "A Ring & Counter Controlled Delay Line for a Wide Operation Range and Low-Jitter Performance," Journal of the Korean Physical Society, vol. 42, pp. 246-250, Feb 2003.

65. T. Gawa, "A 50% duty-cycle correction circuit for PLL output", ISCS, IEEE Proceeding on, vol. 4, pp. 21-21, 2002.

66. K. Agarwal, "A Duty-Cycle Correction Circuit for High-Frequency Clocks", in IEEE Symposium on VLSI circuit Dig. Tech. Papers, pp. 106-107, Jun. 2006.

67. Y. Okuda, M. Horiguchi, Y. Nakagome, "A 66-400 MHz, adaptive-lock-mode DLL circuit with duty-cycle error correction [for SDRAMs]", in IEEE Symposium on VLSI circuit Dig. Tech. Papers , pp. 37-38, Jun. 2001.

68. K. Nakamura, M. Fukaishi, Y. Hirota, Y. Nakazawa and M. Yotsuyanagi, "A CMOS 50% Duty Cycle Repeater Using Complementary Phase Blending", in IEEE Symposium on VLSI circuit Dig. Tech. Papers , pp. 48-49, Jun. 2000.

69. B. W. Garlepp, K. S. Donnelly, J. Kim, P. S. Chau, J. L. Zerbe, C. Huang, C. V. Tran, C. L. Portmann, D. Stark, Y. Chan, T. H. Lee, and Mark A. Horowitz, "A Portable Digital DLL for High-Speed CMOS Interface Circuits", IEEE J. Solid-State Circuits, vol. 34, no 5, pp. 632-644, May 1999.

70. K. H. Kim, G.H. Cho, J. B. Lee and S.-I. Cho, "Built-in Duty Cycle Corrector Using Coded Phase Blending Scheme for DDR/DDR2 Synchronous DRAM Application", Dig. Symp. in IEEE Symposium on VLSI circuit Dig. Tech. Papers , pp. 287-288, Jun. 2003.

71. W. Yun, H. Lee, D. Shin, S. Kang, J. Yang, H. Lee, D. Lee, S. Sim, Y. Ju Kim, W. Choi, K. Song, S. Shin, H. Choi, H. Moon, S. Kwack, J. Lee, Y. Kyoung Choi, N. Park, K. Kim, Y. Choi, J. Ahn, Y. Yang, "A 0.1-to-1.5GHz 4.2mW All-Digital DLL with Dual Duty-Cycle Correction Circuit and Update Gear Circuit for DRAM in 66nm CMOS Technology", in IEEE ISSCC Dig. Tech. Papers, pp. 282-613, Feb. 2008.

72. S. Patil, Kawasaki, S. B. Rudraswamy, "Duty cycle correction using negative feedback loop", Mixed Design of Integrated Circuits & Systems. MIXDES-16th International Conference, pp. 424- 426, 2009.

73. P. Maillard, W.T. Holman, T.D. Loveless, B.L. Bhuva and L.W. Massengill, "An RHBD Delay-Locked Loop (DLL) Utilizing a Differential Delay Line Topology", RADECS, IEEE Conf. proceeding on, Aug. 2011.

74. P. Maillard, L. W. Massengill, W. T. Holman, T. D. Loveless, Y. Chen, N. Roche, "SET characterization of two 90nm voltage controlled delay line topologies", RADECS, IEEE Conf. proceeding on, Aug. 2012

75. P. Maillard, W. T. Holman, T. D. Loveless, B. L. Bhuva and L. W. Massengill, "Design Choices for High Speed RHBD DLL", GOMAC' Tech12, Jan. 2012

76. P. Maillard, W. T. Holman, T. D. Loveless, L. W. Massengill, "A New Error Correction Circuit for Delay Locked Loops", in IEEE Trans. Nucl. Sci., vol. 60, no. 6, pp. 2387–2393, Dec. 2013.

77. K. Cheng, C. li, S. Chang, S. Jiang, Wei-Bin Yang, "A 2GHz fully differential DLL-based frequency multiplier for high speed serial link circuit", ISCAS, IEEE International Symposium on, pp. 1174 – 1177, May 2005. .

78. Genesys™ Board Reference Manual, rev. C., Diligent Inc., San Jose, CA, 2013.

79. Virtex-5 FPGA User Guide, UG190 v.5.4, Xilinx Co., San Jose, CA, 2012.

80. ChipScope Pro Software and Cores User Guide, UG029 v.14.3, Xilinx Co., San Jose, CA, 2012

81. B. D. Olson, D. R. Ball, K. M. Warren L. M. Massengill, N. F. Haddad, S. E. Doyle, D. McMorrow, "Simultaneous single event charge sharing and parasitic bipolar conduction in a highly-scaled SRAM design", in IEEE Trans. Nucl. Sci., vol. 52, no. 6, pp. 2132-2136, Dec. 2005.

82. T. Hamamoto, K. Furutani, T. Kubo, S. Kawasaki, H. Iga, T. Kono, Y. Konishi, and T. Yoshihara, "A 667-Mb/s operating digital DLL architecture for 512-Mb DDR SDRAM", in IEEE J. Solid-State Circuits, vol. 39, no. 1, pp. 194–206, Jan. 2004.

83. T. D. Loveless, L. W. Massengill, W. T. Holman, B. L. Bhuva, D. McMorrow, G. H. A. Warner, "A Generalized Linear Model for Single Event Transient Propagation in Phase-Locked Loops", in IEEE Trans. Nucl. Sci., vol. 57, no. 5, pp. 2933- 2947, Oct. 2010.

84. M. Combes, K. Dioury and A. Greiner, "A portable clock multiplier generator using digital CMOS standard cells", in IEEE J. Solid-State Circuits, vol. 31, no. 7, pp. 958–965, Jul. 1996.

85. E. R. Booth, "wide range, low jitter delay-locked loop using a graduated digital delay line and phase interpolator", MSEE, Boise State University, Nov. 2006.

86. J. Barak, J. Levinson, M. Victoria, and W. Hajdas, "Direct processes in the energy deposition of protons in silicon," IEEE Trans. Nucl. Sci., vol.43, pp. 2820–2826, Dec. 1996.

87. S. Duzellier, R. Ecoffet, D. Falguère, T. Nuns, L. Guibert, W. Hajdas and M. C. Calvet, "Low energy proton induced SEE in memories," IEEE Trans. Nucl. Sci., vol. 44, pp. 2306–2310, Dec. 1997.