

IONIZING RADIATION EFFECTS SPECTROSCOPY

By

Bharat Patel

T. Daniel Loveless
UC Foundation Associate Professor
of Electrical Engineering
(Chair)

Abdul R. Ofoli
UC Foundation Associate Professor
of Electrical Engineering
(Committee Member)

Donald R. Reising
Assistant Professor of Electrical Engineering
(Committee Member)

IONIZING RADIATION EFFECTS SPECTROSCOPY

By

Bharat Patel

A Thesis Submitted to the Faculty of the University of
Tennessee at Chattanooga in Partial
Fulfillment of the Requirements of the Degree
of Master of Science: Engineering

The University of Tennessee at Chattanooga
Chattanooga, Tennessee

December 2019

ABSTRACT

The reliability of microelectronics operating in harsh environments is a concern for space systems. Various stresses, such as exposure to ionizing radiation and extreme temperatures can result in performance degradation, transient anomalies, and hard failures. These issues have become evident in recent years with the increasing interest in the use of commercial-off-the-shelf (COTS) electronics in space systems. While COTS parts offer maximum performance, their use results in unavoidable increases in risk. This work presents a noninvasive technique for the measurement of cumulative and transient radiation effects in arbitrary circuits, termed ionizing radiation effects spectroscopy (IRES). IRES identifies radiation effects based on statistical time-frequency analyses of native waveform behavior, thus having a minimal impact on operating performance. The resulting measures are used to identify parametric shifts as well as transient anomalies. IRES, which exploits the subtle characteristics inherent in the waveforms, shows promise for in-situ health monitoring and radiation mitigation.

DEDICATION

I dedicate this work to my mom for supporting me to ensure that I have a bright career ahead. I also want to dedicate this work to my teachers, family members and friends who had supported and encouraged me to follow my passion amidst tough times in my life.

ACKNOWLEDGEMENT

I would like to thank the Defense Threat Reduction Agency, the Tennessee Higher Education Commission through the Center of Excellence in Applied Computational Science and Engineering Program at The University of Tennessee at Chattanooga (UTC), and the National Science Foundation (NSF) for supporting my work. I would like to thank Dr. Daniel Loveless for mentoring and supporting me through my undergraduate and graduate school experience. I cannot thank him enough for helping me develop the technical skills that would be invaluable when I go to industry. I would also like to thank Dr. Donald Reising for advising me on engineering problems during my graduate school experience at UTC. I would like to thank my committee: Dr. Daniel Loveless, Dr. Donald Reising and Dr. Abdul Ofoli for reviewing the thesis and giving me suggestions for improvement such that this work can be presented most accurately. I would also like to thank my colleagues at UTC for their support in the research.

TABLE OF CONTENTS

ABSTRACT.....	iii
DEDICATION.....	iv
ACKNOWLEDGEMENTS.....	v
LIST OF TABLES.....	viii
LIST OF FIGURES.....	ix
LIST OF ABBREVIATION.....	xiii
LIST OF SYMBOLS.....	xv
CHAPTER	
I. INTRODUCTION.....	1
II. BACKGROUND.....	4
Space Environment.....	4
Total Ionizing Dose.....	6
Single Event Effects.....	7
RF-DNA Fingerprinting.....	8
Chapter Conclusion.....	10
III. IONIZING RADIATION EFFECTS SPECTROSCOPY FOR MEASUREMENT OF CUMULATIVE AND TRANSIENT RADIATION EFFECTS.....	12
Introduction.....	12
IRES developed for TID.....	12
Time-Frequency Spectrographic Analysis.....	14
IRES developed for SETs.....	17
Chapter Conclusion.....	18

IV.	TOTAL-IONIZING DOSE ANALYSIS WITH IRES.....	20
	Introduction.....	20
	Experiment details for VCOs and PLLs designed in 130 nm CMOS process.....	23
	IRES Image Formation	28
	Classification of TID levels using ML.....	34
	Chapter Conclusion.....	43
V.	SINGLE EVENT TRANSIENT ANALYSIS WITH IRES	45
	Introduction.....	45
	Experimental Analysis with IRES	46
	Simulation Analysis with IRES	54
	Chapter Conclusion.....	58
VI.	DISCUSSION AND CONCLUSION	59
	REFERENCES	61
	APPENDIX	
	A. MATLAB SCRIPTS.....	65
	VITA.....	84

LIST OF TABLES

1	Classification of TID levels for VCO A	37
2	Classification of the TID levels for both VCOs and PLLs	42

LIST OF FIGURES

1	Near Earth space environment illustration, after [6]	5
2	Motion of charged trapped particles in Earth’s magnetic field [8, 9]	5
3	Schematic of n-channel MOSFET showing radiation induced charges in the gate oxide [10]: (A) Normal operation of the transistor: Gate voltage is applied and MOSFET device is turned on(B) Post-irradiation: Trapped charges in gate oxide turns on the device with $V_G = 0$ V	7
4	Single Event Transient current spike representation through a double exponential curve	8
5	Average RF-DNA fingerprints of four Cisco devices: Each column represents the RF-DNA fingerprint of one device [15]	10
6	Waveform segmentation for the development of a feature sequence (F) used in an IRES image. An arbitrary time sequence is divided into N blocks, each consisting of M time steps. Various statistical measures are then computed within each of the N blocks and concatenated to form a feature sequence [15]	13
7	Spectrogram of the PSD of the PLL’s output waveforms at ~150 MHz frequency versus time with window size of 15 samples and window shift of 1 sample [4]	15
8	Spectrogram of PSD of the cycle-to-mean phase jitter versus time using ST-DFT with a Hamming window function with window size of 15 samples and window shift of 1 sample [4].....	16
9	Example waveforms used to generate spectrograms in Figure 7 (Top waveform) and Figure 8 (Bottom waveform). Top waveform represents PLL’s output waveform after a SE strike. Bottom waveform represents extracted cycle-to-mean jitter from the output waveform	17
10	Illustration of the window sampling method used for the calculation of statistical measures for identification of SET [4]	18
11	General block diagram of a phase-locked loop circuit consisting of phase-frequency detector (PFD), charge pump (CP), low-pass filter (LPF), voltage controlled oscillator (VCO) , and frequency divider (β) [26]	21

12	Measured phase noise at 1-MHz offset as function of TID of a 32-nm VCO at a constant frequency of 20.4 GHz [31]	22
13	Measured frequency tuning curves showing output frequency versus input control voltage V_{inVCO} of VCO A designed in a 130-nm CMOS technology at TID levels of 0, 40, and 300 krad(SiO ₂) [3]	24
14	Measured frequency tuning curves showing output frequency versus input control voltage V_{inVCO} of VCO B designed in a 130-nm CMOS technology at TID levels of 0, 40, and 300 krad(SiO ₂) [3]	25
15	Measured frequency tuning curves showing output frequency versus input control voltage V_{inVCO} of PLL A designed in a 130-nm CMOS technology at TID levels of 0, 40, and 300 krad(SiO ₂) [3]	25
16	Measured frequency tuning curves showing output frequency versus input control voltage V_{inVCO} of PLL B designed in a 130-nm CMOS technology at TID levels of 0, 40, and 300 krad(SiO ₂) [3]	26
17	Example waveforms used for generation of IRES: Pre-irradiation 0 krad(SiO ₂) (Top plot) and Post irradiation at 300 krad(SiO ₂) (Bottom plot) [3].....	27
18	Probability density functions of the VCO's instantaneous frequency: Pre-irradiation at 0 krad(SiO ₂) (Blue/Dashed plot) and Post irradiation at 300 krad(SiO ₂) (Red/Solid plot) [3]	28
19	IRES image formation process for TID.....	29
20	IRES images representing the mean and standard deviation measures of instantaneous frequency of VCO A at frequency values of 62.6, 119.8, and 193.8 MHz for V_{inVCO} values of 0.5, 0.6, and 0.7 V, respectively [3]	30
21	IRES images containing the mean, standard deviation, and variance measures of instantaneous frequency of VCO A at fixed bias of 0.6 V with respect to 0, 40, and 300 krad(SiO ₂) [3]	31
22	IRES images of VCO A at a fixed frequency of ~302 MHz at 0 krad(SiO ₂) (left) and 300 krad(SiO ₂) (right). VCO A was retuned to maintain the constant frequency [3]	32
23	IRES images of PLL A at a fixed frequency of ~178 MHz at 0 krad(SiO ₂) (left) and 300 krad(SiO ₂) (right). PLL A was re-tuned to maintain the constant frequency [3]	33
24	Scatter plot of standard deviation versus mean of the instantaneous frequency feature for VCO A at V_{inVCO} values between 0.4 and 1.2 V at 0 krad(SiO ₂) [3]	35

25	Scatter plot of the standard deviation versus mean of the instantaneous frequency feature for VCO A at V_{inVCO} values between 0.5 and 0.8 V at 0 and 300 krad(SiO_2). The data clusters are labeled with a unique V_{inVCO} and TID pair [3].....	38
26	Corresponding IRES images (40 samples) of standard deviation and mean of the instantaneous frequency for VCO A at various V_{inVCO} and TID pairs for the scatter plots shown in Figure 25 [3]	39
27	Scatter plot of the standard deviation versus mean of the instantaneous frequency feature for PLL A at V_{inVCO} values between 0.5 and 0.8 V at 0 and 300 krad(SiO_2). The data clusters are labeled with a unique V_{inVCO} and TID pair [3].....	40
28	Corresponding IRES images (40 samples) of standard deviation and mean of the instantaneous frequency for PLL A at various V_{inVCO} and TID pairs for the scatter plots shown in Figure 27 [3].....	41
29	Scatter plot of the standard deviation versus mean of the instantaneous frequency feature for VCO B at V_{inVCO} values between 0.5 and 0.8 V at 0 and 300 krad(SiO_2). The data clusters are labeled with a unique V_{inVCO} and TID pair [3].....	42
30	Example of an output frequency and phase transient following a laser perturbation on the CP sub-circuit of the PLL fabricated in a 130 nm CMOS technology. The PLL was operating at 200 MHz. Following the laser strike with incident energy of 5.4 nJ, the output frequency was reduced to approximately 50 MHz, thus increasing the output phase displacement to approximately 15 radians. The recovery time of the PLL was over 200 ns and it is indicated by red arrow in the middle plot of output frequency (f_{out}) versus time [26].....	47
31	An image of the output phase displacement versus x-y location fused to the layout image of the CP sub-circuit. The experiment was performed using laser TPA at a step size of 0.2 μm [26]	49
32	IRES image generated from experimental data obtained for SET originated from a strike on a NMOS transistor at the output node of the CP of the PLL . SET was observed when the PLL was running approximately at 150 MHz. A total of 15 samples were used for one window and statistical measures were calculated. The shift value of the window was 1. Skewness and kurtosis of the cycle-to-mean jitter show the onset of the SET indicated by red color column on the top left corner of the image. Radiation and circuit response mechanisms are characterized by the statistical measures of the cycle-to-mean jitter.	51

33	IRES image generated from experimental data obtained for SET originated from a strike on a PMOS transistor at the output node of the CP of the PLL . SET was observed when the PLL was running approximately at 150 MHz. Total of 15 samples were used for one window and statistical measures were calculated. The shift value of the window was 1	53
34	Time-domain waveforms used to generate IRES images in Figures 32 (top plot in this figure) and 33 (bottom plot in this figure).....	53
35	Simulated SE strikes at the output of the CP after various levels of charge deposited from a current source representing the NMOS transistor strike. Recovery times between 200 and 300 ns was observed. The drop in voltage was between 0.15 and 0.4 V [4].....	55
36	IRES image of SET simulated in PLL at 260 MHz for 249.2 fC charge deposited at the output of the CP	56
37	IRES image of SET simulated in PLL at 260 MHz for 373.8 fC charge deposited at the output of the CP	57
38	IRES image of SET simulated in PLL at 260 MHz for 498.3 fC charge deposited at the output of the CP	57

LIST OF ABBREVIATIONS

CME, Coronal Mass Ejections

CMOS, Complementary Metal-Oxide-Semiconductor

COTS, Commercial-Off-the-Shelf

CP, Charge Pump

CPLL, Charge Pump Phase-Locked Loop

DD, Displacement Damage

DFT, discrete Fourier transform

DUT, Device Under Test

ESA, European Space Agency

GPS, Global Positioning Systems

GSRs, Galactic Cosmic Rays

IoT, Internet-of-Things

IRES, Ionizing Radiation Effects Spectroscopy

LD, Linear Discriminant

LEO, Low Earth Orbit

LET, Linear Energy Transfer

LPF, Low-Pass Filter

ML, Machine Learning

MOS, Metal-Oxide-Semiconductor

MOSFET, Metal Oxide Silicon Field Effect Transistor

PFD, Phase-Frequency Detector

PLL, Phase Locked Loop

PSD, Power Spectral Density

RF-DNA, Radio Frequency-Distinct Native Attribute

RHBD, Radiation Hardened by Design

SAA, South Atlantic Anomaly

SEE, Single Event Effects

SEGR, Single Event Gate Rupture

SEL, Single Event Latchups

SET, Single Event Transients

SEU, Single Event Upsets

ST-DFT, Short-time discrete Fourier transform

STFT, Short-Time Fast-Fourier Transform

TID, Total-Ionizing Dose

TPA, Two-Photon Absorption

VCO, Voltage-Controlled Oscillator

LIST OF SYMBOLS

V_G , gate voltage

β , frequency divider

V_{DD} , supply voltage

V_{inVCO} , VCO input voltage

v_{out} , output voltage

v_n , P -point sampled sequence of signal $v_{out}(t)$

I , irradiance

f_{out} , output frequency

ϕ_{out} , output phase

CHAPTER I

INTRODUCTION

Complementary metal-oxide-semiconductor (CMOS) devices have been shrinking in size in accordance with Moore's Law since 1965 [1, 2]. This reliable scaling of integrated circuit technologies has fundamentally changed society, enabling widespread use of smartphones, Global Positioning Systems (GPS), and Internet-of-Things (IoT) connected devices, for example. All of these services regularly use satellites as the information relaying routers. However, the reliability of the commercially available parts, and in particular the radiation-related reliability, for operation in space is not improving as fast as dimensional scaling and is often seen to worsen with scaling. Consequently, space systems often use a costly combination of radiation-hardened by design (RHBD) and commercial off the shelf (COTS) components to improve the radiation-related reliability. The inevitable regular use of COTS for obtaining the maximum performance capabilities offered by a technology raises the concern for feasibility of these devices which are meant for long term space missions.

Satellites in space are vulnerable to radiation generated from Galactic Cosmic Rays (GSRs) and solar windstorms. The harsh space environment exerts various type of stresses dynamically (i.e., temperature and/or radiation) to such devices. These devices are protected by shielding materials which lets a minimal number of energized particles to penetrate and reduces the risk of damaging the devices. However, not all parts of the satellite are protected by shielding and it is not economically ideal to shield the entire satellite's electronics. A CMOS device can sustain certain

amount of cumulative radiation (i.e. total ionizing dose or TID) before it alters from nominal operation or stops functioning permanently. Also, various transient phenomena associated with the interaction of single ionizing particles with the semiconductor material (i.e. single event effects or SEE) can negatively impact the operation of the systems. These types of radiation phenomena can be assessed using various dosimetry techniques, but these techniques are somewhat invasive to the circuit operations and require additional hardware.

This work proposes a non-invasive technique, termed Ionizing Radiation Effects Spectroscopy (IRES), that allows for the in-situ identification of TID and Single Event Transient (SET) radiation effects without the need for internal node interrogation or modification of the circuits and systems. IRES is based on a technique called radio frequency distinct native attribute fingerprinting, shortly called RF-DNA fingerprinting, developed for wireless radio identification. RF-DNA relies on the statistical features of a device's output signal for identification of authorized device from rogue devices within a network. Similarly, for system changes induced by TID or SEE, the IRES technique measures statistical features of data samples collected from an arbitrary circuit signal. Shifting statistical features of an arbitrary signal are used to indicate the presence of radiation-induced parametric shifts or transient phenomena. The calculated statistical measures are then used to train a Machine Learning (ML) algorithm for analysis of the system state. In this work, the TID levels and operational voltage states of Voltage-Controlled Oscillator (VCO) and Phase Locked Loop (PLL) circuits are used to demonstrate the use of the IRES technique. IRES provides radiation damage illustrations that could be very useful when considering lifespan of a device in space orbit which could not be manually fixed or replaced. These illustrations give specific information about the damage on the circuit such as instability of the circuit or charge

deposited through the radiation effects. Work presented in this thesis is demonstrated in the paper [3] and [4].

CHAPTER II

BACKGROUND

Space Environment

There are various space radiation phenomena, such as TID, Displacement Damage (DD), SEE, and Charging/discharging effects, for example, known to adversely affect the nominal circuit/satellite operations. These events are caused by the interaction of ionizing radiation with semiconductor material within the electronics systems. The ionizing radiation present in the space environment is due to Galactic Cosmic Rays (GCRs), solar events and charged particles, such as protons and electrons trapped within the Earth's magnetosphere.

Heavy ions, protons and electrons from solar particle events and GCRs are trapped within the magnetic field lines of the Earth creating the Van Allen belts [5], illustrated in Figure 1 [6]. The electronics in satellites are vulnerable when they are passing through these belts. The interactions of satellite devices or electronics with the charged particles can disturb the satellite operations by setting or resetting bits, creating transient signals that compete with legitimate signals within the systems, and by degrading device parameters. Sometimes these interactions can permanently damage the satellite and jeopardize the mission. For example, the HIPPARCOS satellite mission of the European Space Agency (ESA) was terminated due to radiation effects on satellite components resulting difficulties in satellite communication between the ground and the onboard computer [7].

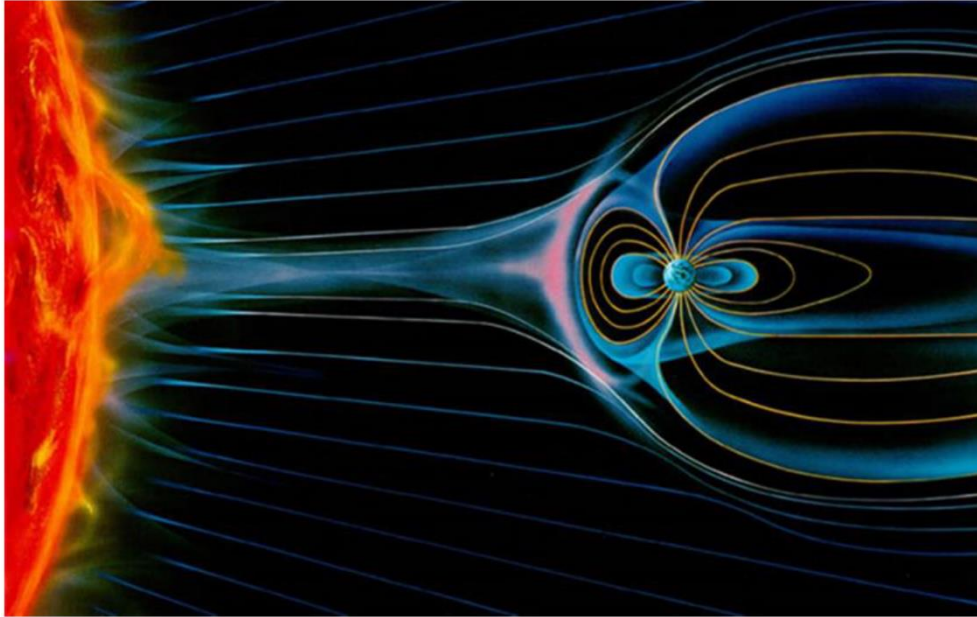


Figure 1

Near Earth space environment illustration, after [6]

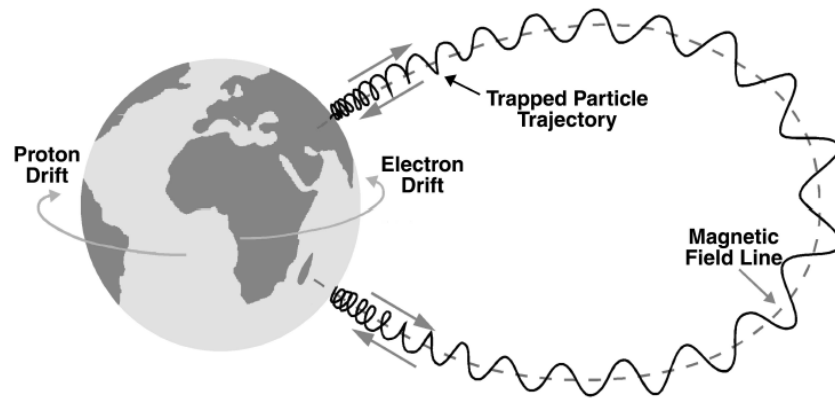


Figure 2

Motion of charged trapped particles in Earth's magnetic field [8, 9]

Once a charged particle gets trapped into Earth's magnetic field, it moves into a spiral motion illustrated in Figure 2 [8, 9]. As the particle approaches the Earth's polar region, its spiral

motion tightens due to the high magnetic field strength and subsequently reverses its direction. The trapped protons in Low Earth Orbit (LEO) are the cause of intense and penetrating radiation in the South Atlantic Anomaly (SAA) region [8]. In addition to the protons and electrons, there are heavy ions with very high-energy levels travelling from outer space to the solar system called GCR.

Trapped protons can cause TID effects, DD effects and SEE, whereas trapped electrons can cause TID effects, DD effects, and charging/discharging effects on the spacecraft. GCRs contain high-energy charged particles with energy as high as 10^{11} GeV that are originated outside of the solar system and they can cause SEE. Solar particle event such as Coronal Mass Ejections (CME) can cause TID effects, DD effects, and SEE [9]. To demonstrate the feasibility of the IRES technique, only TID and SEE radiation effects on the circuits are analyzed in this work.

Total Ionizing Dose

In MOS transistors, radiation induced charge is trapped in oxides. In MOS transistors, for example, trapped charge within the gate oxides causes shifts in the threshold voltages of MOS transistors [10]. Sometimes, the shifts are large enough to turn on the MOS transistors without applying any gate voltages. The mechanism of charge trapping in gate oxides of a Metal Oxide Silicon Field Effect Transistor (MOSFET) device is illustrated in Figure 3 [10]. The radiation accumulates in the gate oxides and eventually even without applying the gate voltage, the potential from the trapped radiation charges turns on the transistor. Thus, TID is a cumulative or long-term radiation effect. The charging of the oxides due to ionizing radiation can affect the devices' overall response and also result in degradation in device performance over time.

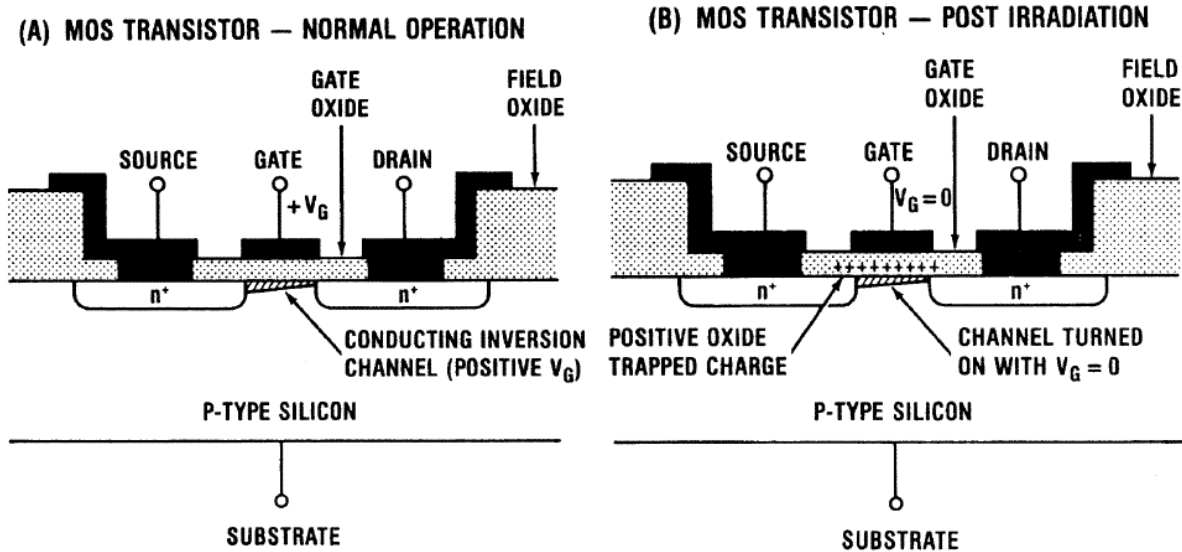


Figure 3

Schematic of n-channel MOSFET showing radiation induced charges in the gate oxide [10]:
 (A) Normal operation of the transistor: Gate voltage is applied and MOSFET device is turned on
 (B) Post-irradiation: Trapped charges in gate oxide turns on the device with $V_G = 0$ V

Single Event Effects

Single Event Effects (SEE) are caused by ionized particles traversing semiconductor devices. After a strike near a p-n junction, electron-hole pairs are generated, and these charges are swept across the p-n junction generating a photocurrent spike. Charge deposition can vary based on the Linear Energy Transfer (LET) of the ionizing particle. The energy transferred to the semiconductor device is defined by LET. The charge deposited by an ionized particle during the event can result in a temporary disruption in the nominal circuit operation, an upset (i.e. a change in the state of a memory bit from 0 to 1 or 1 to 0), or permanent damage to the device. Upsets resulting from SEE can propagate through the entire circuit, causing errors visible at the circuit output nodes. The current generated due to the transient strike are typically represented by double

exponential curves shown in Figure 4 as it is relatively easy to use the double exponential current source in circuit simulations [11].

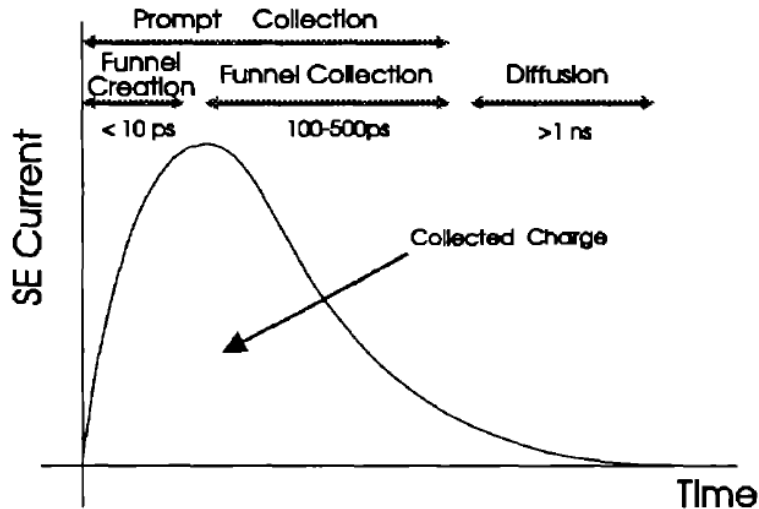


Figure 4

Single Event Transient current spike representation through a double exponential curve [11]

As the dimensions of semiconductor devices is decreasing with the technology trends, SEE is becoming more effective in disruption of the circuit operation [12]. A SEE can be categorized as a Single Event Transient (SET), Single Event Upset (SEU), Single Event Latchup (SEL), Single Event Gate Rupture (SEGR), and Single Event Burn out, to name a few. This work focuses on effects generated by SETs, as the first response of a given semiconductor device is generally transient in nature.

RF-DNA Fingerprinting

The IRES technique is based on radio frequency-distinct native attribute (RF-DNA) fingerprinting developed for identification of wireless devices using statistics-based features

extracted from their transmissions. RF-DNA fingerprinting has successfully demonstrated unique identification of wireless transmitters of the same manufacturer and model (i.e., serial number discrimination), which represents the most difficult case [13-19]. RF-DNA fingerprinting technique is based on the idea of human biometrics in which an individual's identity is established from attributes present within a given physical trait. For example, fingerprints and the retinal patterns are unique to each individual.

Figure 5 shows the RF-DNA fingerprints of four distinct devices [15]. Each RF-DNA fingerprint consists of three different statistical measures : kurtosis, skewness, and variance. It can be seen that each device's statistical measures are different from one another. Other statistical measures such as mean and standard deviation can be used to represent the RF-DNA fingerprint. There are three basic approaches for the extraction of RF-DNA fingerprints: 1) time-domain fingerprints extracted from instantaneous amplitude, phase, and frequency [17, 19-21]; 2) frequency-domain fingerprints extracted from the waveform power spectral density [13]; 3) joint time-frequency-based fingerprints [14, 18].

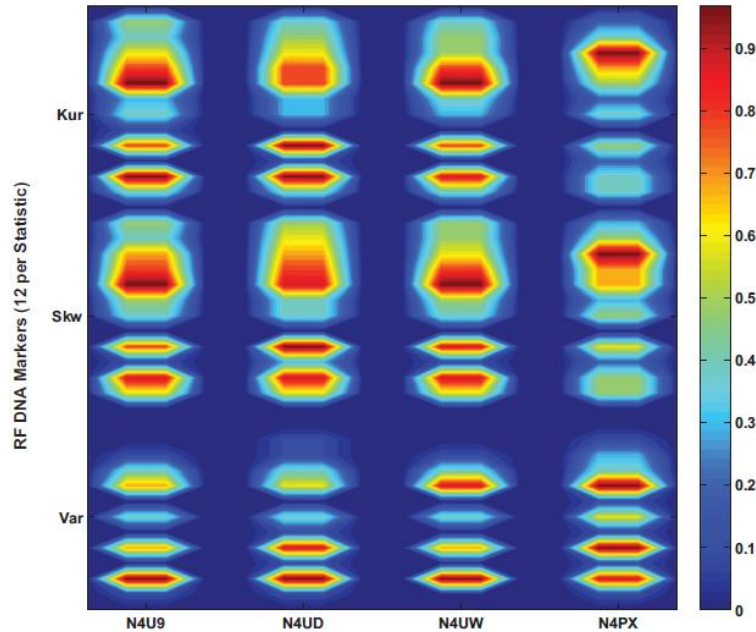


Figure 5

Average RF-DNA fingerprints of four Cisco devices: Each column represents the RF-DNA fingerprint of one device [15]

Chapter Conclusion

The most basic devices in microelectronic circuits, such as MOSFETs, are vulnerable to the radiation in the space environment. The stress placed on these devices by the effects of space radiation can alter circuits' nominal operations and jeopardize the mission. As satellites pass through the Van Allen belts, for example, the heavy presence of ionizing particles must be taken into consideration while estimating the reliability of the designed circuit. TID can degrade the device over long periods of time before completely failing, whereas SEE can result in spurious transient anomalies. This ionizing radiation can be measured using different dosimetry techniques, but these techniques are invasive to the operations of the circuit. The next chapter presents ionizing

radiation effect spectroscopy (IRES), a technique that shows a promise in eliminating the need of probing circuit nodes to identify radiation effects degradation and anomalies.

CHAPTER III

IONIZING RADIATION EFFECTS SPECTROSCOPY FOR MEASUREMENT OF CUMULATIVE AND TRANSIENT RADIATION EFFECTS

Introduction

Device performance parameters have been shown to worsen with technology scaling due to inherent variation present in devices [22]. While variability is undesirable due to the additional constraints imposed on operating performance, the unavoidable variation allows for the identification of distinct and inherent features at the transistor, circuit, or system levels. With an accumulation of TID, these fundamental features shift with respect to the amount of charge trapping within the oxides, resulting in modification of the statistical parameters observable in the output waveform. These feature shifts result in an ability to characterize performance degradation through the measurement of waveform statistics without the need for invasive device interrogation.

IRES developed for TID

Similar to RF-DNA fingerprinting, IRES exploits sufficiently distinct and native attributes present within the waveform and it serves as the discriminatory information utilized by the technique. Here, the IRES technique is described for identification of TID levels and operational circuit voltage levels. In the proceeding sections, the IRES technique is developed for identification of SETs through statistical measurements of signal metrics (e.g., cycle-to-mean jitter and instantaneous frequency). Figure 6 shows the process of the development of a feature sequence

(F) from the output signal of the circuit. An arbitrary time sequence is divided into N blocks, each consisting of M time steps. Various statistical measures are then computed within each of the N blocks and concatenated to form a feature sequence [15]. In the case of TID, the N signal blocks generally do not overlap, but as discussed in the proceeding sections, overlapping signal blocks may be employed to enhance the identification of spurious, short-lived anomalies.

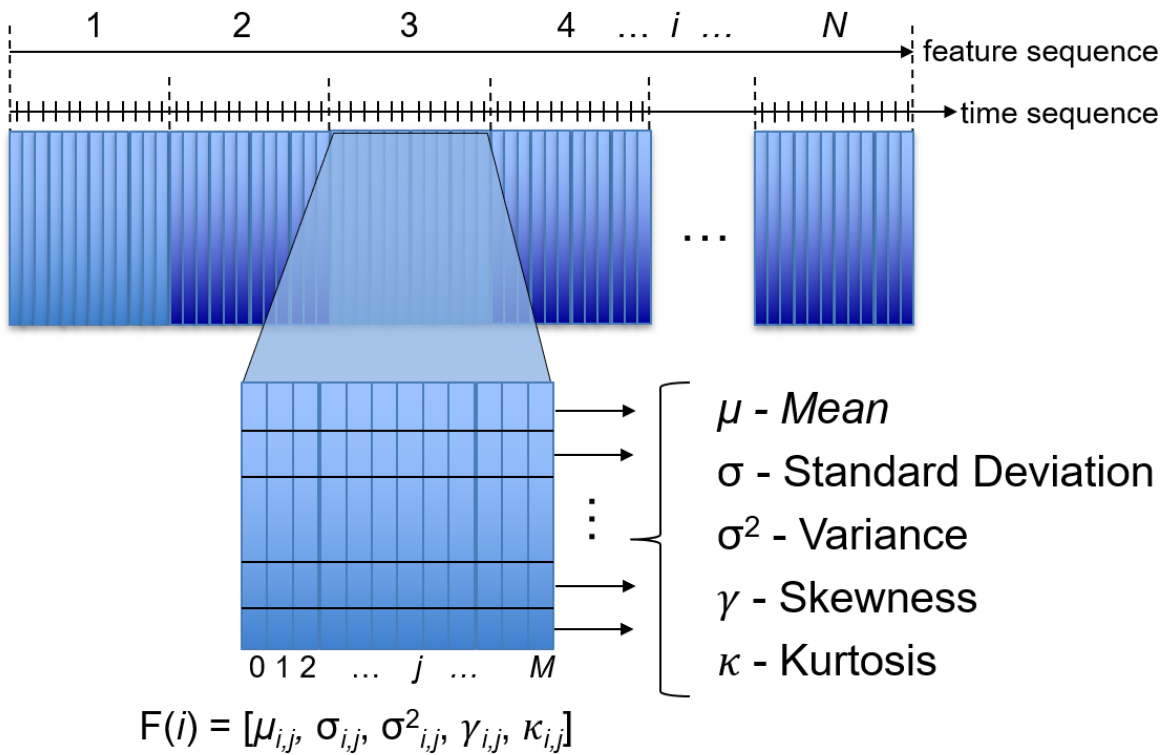


Figure 6

Waveform segmentation for the development of a feature sequence (F) used in an IRES image. An arbitrary time sequence is divided into N blocks, each consisting of M time steps. Various statistical measures are then computed within each of the N blocks and concatenated to form a feature sequence [15]

Time-Frequency Spectrographic Analysis

Prior to describing IRES for identification of SET, time-frequency analysis is discussed for general use in spectroscopy applications. Spectroscopy has been employed in reliability applications for the extraction of the time-dependence associated with specific physical degradation phenomena. Characterization of the bias temperature instability has been performed using a similar spectroscopy method [23]. In general, time-dependent spectroscopy involves the measurement of timing information associated with a defect or anomalous behavior, and the development of a statistical profile of such data. This work employs time-frequency spectral analysis and develops a methodology for quantifying the statistical parameters associated with SET behavior. This section delivers an overview of time-frequency spectrographic analysis, time-frequency Fourier analysis, and overviews the various tradeoffs associated with time and frequency domains. Then, IRES is presented as a method to augment time-frequency analysis to isolate erroneous transient behavior embedded within dynamic waveforms.

In this work, -a phase-locked loop (PLL) circuit (see Chapter V) was used to demonstrate the feasibility of this technique. The Short-Time Fast-Fourier Transform (ST-DFT) [24] for viewing the frequency and phase content of a signal versus time is added with IRES to allow for isolating the spurious events within a signal. The Discrete Fourier Transform (DFT) [24] using (3.1) can decompose the discrete sampled signal ($v_{out}(t)$) into its component frequencies.

$$V(f_n) = \sum_{n=0}^{P-1} v_n e^{-\frac{j2\pi n}{P}} \quad (3.1)$$

where $v_n=v[n]$ is the P -point sampled sequence of signal $v_{out}(t)$ for $n = 0, 1, \dots, N-1$. The Power Spectral Density (PSD) of the signal can be determined using (3.2)

$$S_v(f_n) = \frac{1}{P} |V(f_n)|^2 \quad (3.2)$$

where, S_v is the PSD. Power for each frequency content (in units of dB/Hz) from total signal power is deconstructed by PSD [25]. Work in [4] shows that 2D joint time-frequency analysis allows for the extraction of short time-frame spectral content where DFT analysis is conducted in sample-limited window $w[n,m]$ according to (3.3).

$$V(m, f_n) = \sum_{n=0}^{P-1} v_n w[n - m] e^{-\frac{j2\pi n}{P}} = V(f_n) * w[n] \quad (3.3)$$

Example spectrograms of the PLL's output waveforms and the cycle-to-mean phase jitter versus time with a window size of 15 samples and window shift of 1 sample using the ST-DFT with a Hamming window function [24] are shown in Figures 7 and 8 [4], respectively. The spectrograms in Figures 7 and 8 show the PSD of the frequency content contained within a sample window.

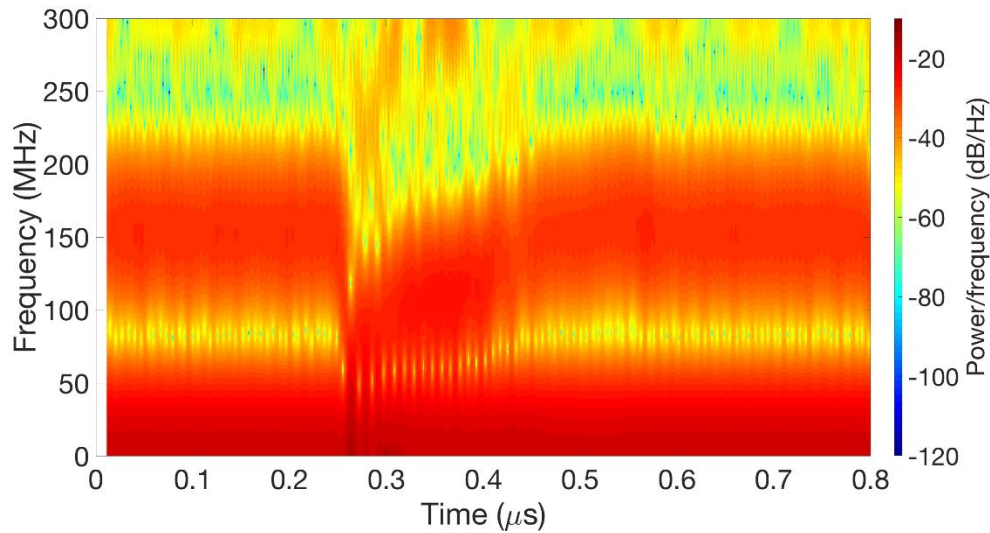


Figure 7

Spectrogram of the PSD of the PLL's output waveforms at ~150 MHz frequency versus time with window size of 15 samples and window shift of 1 sample [4]

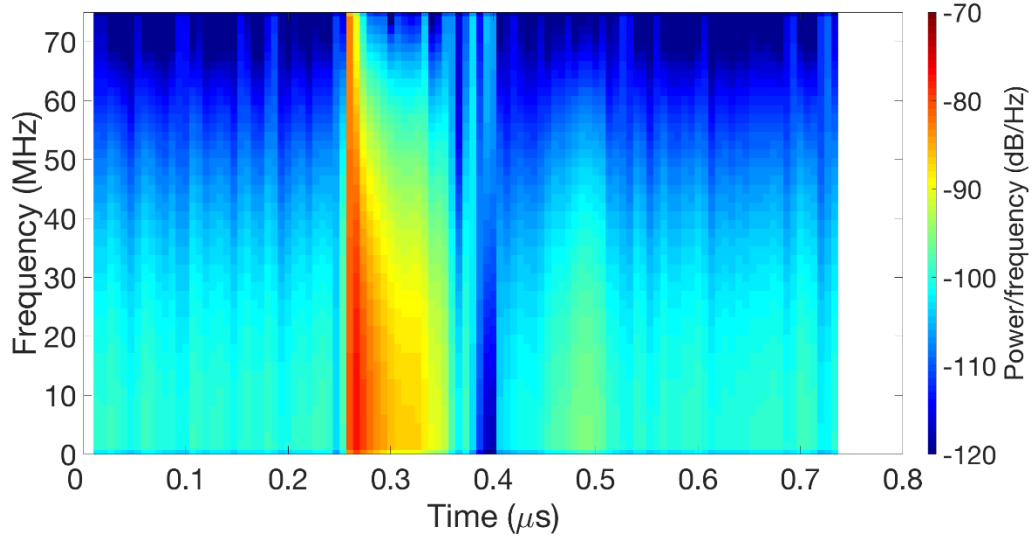


Figure 8

Spectrogram of PSD of the cycle-to-mean phase jitter versus time using ST-DFT with a Hamming window function with window size of 15 samples and window shift of 1 sample [4]

The spectrogram shown in Figure 8 was computed from cycle-to-mean phase jitter waveforms using the equation in 3.4

$$\phi_k(v_n) = \frac{2\pi |T_k - E(T_k)|}{E(T_k)} \quad (3.4)$$

where the signal v_n is sampled at each rising clock edge k , T_k corresponds to the instantaneous period at the k^{th} edge, and $E(T_k)$ is the expected period or population mean [26]. Work in [4] includes more spectrograms to show the tradeoff in time and frequency using varying window sizes and overlapping windows. It illustrates that a low number of data samples (i.e., 15 data samples from total of approximately 110 data samples) generated from the PLL's output waveform and cycle-to-mean phase jitter within each window results in significant noise while improving the ability to detect an anomaly. Figure 9 shows corresponding waveforms that were used to generate the spectrograms in Figures 7 and 8. The top waveform in Figure 9 shows the PLL's

output waveform, while the bottom waveform shows cycle-to-mean phase jitter extracted from the PLL's output waveform. IRES augments time-frequency analysis presented in this section to leverage the computational savings of windowed spectral analysis while employing statistical analysis for isolating erroneous transient behavior embedded within dynamic waveforms.

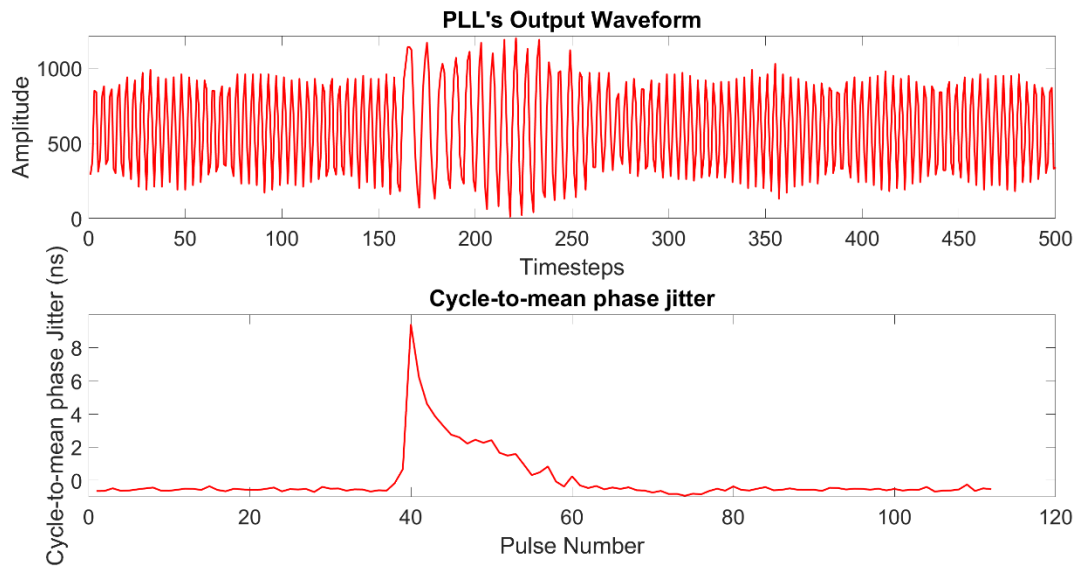


Figure 9

Example waveforms used to generate spectrograms in Figure 7 (Top waveform) and Figure 8 (Bottom waveform). Top waveform represents PLL's output waveform after a SE strike. Bottom waveform represents extracted cycle-to-mean jitter from the output waveform

IRES developed for SETs

The IRES technique, from the previous section, is adopted here to identify SET anomalies. Statistical functions are computed within each window to assess the SET behavior and quantify the severity of the SETs. Figure 10 illustrates the window sampling method used for calculation of statistical measures computed for each window for identification of SET. As mentioned earlier in the Chapter II, Section "RF-DNA Fingerprinting", the statistical measures such as mean,

standard deviation, variance, kurtosis, and skewness are calculated for each window. More detail about both IRES methods will be presented in subsequent Chapters IV and V.

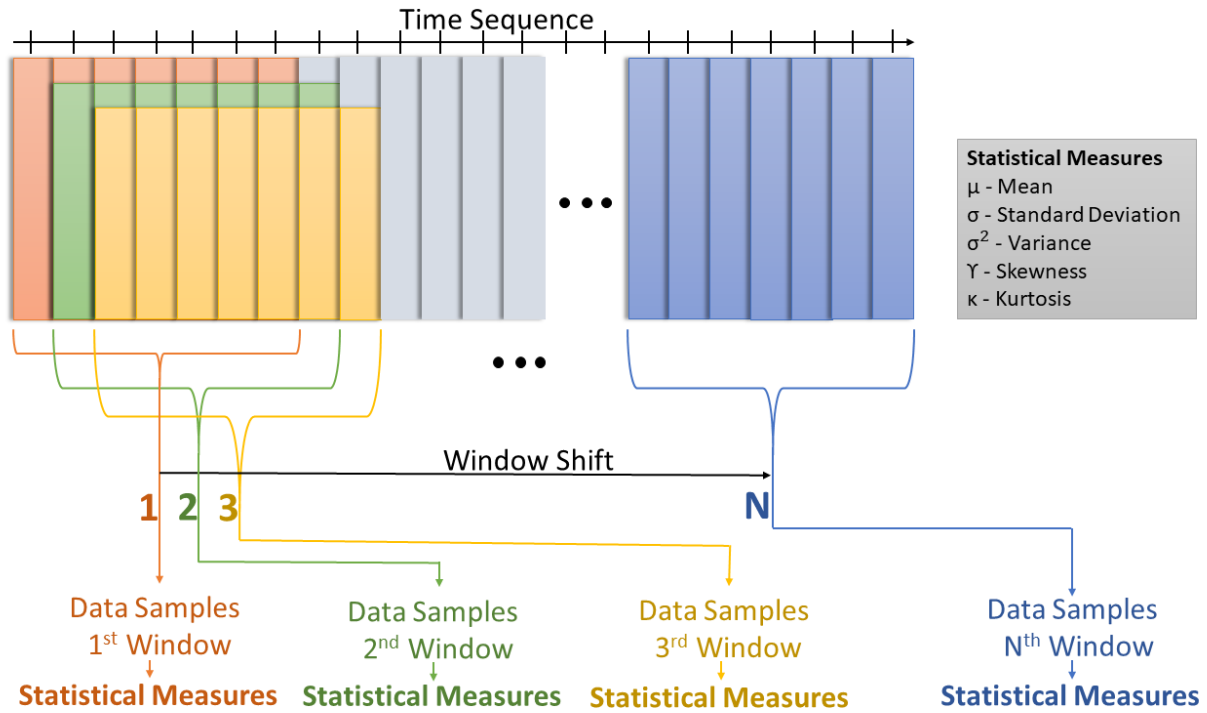


Figure 10

Illustration of the window sampling method used for the calculation of statistical measures for identification of SET [4]

Chapter Conclusion

A non-invasive technique, named IRES, was developed from the concept of RF-DNA fingerprinting. It uses the statistical analysis of any arbitrary signal of a circuit to analyze the circuit behavior in the presence of ionizing radiation. The IRES technique augmented with statistical analysis and time-domain spectroscopy analysis demonstrates practicality of it in the radiation effects domain. The IRES technique allows for a non-invasive method for determining the level

of ionizing dose as well as the operational voltage level of the circuit. In other words, IRES allows for in-situ determination of the circuit's operational health. Also, the IRES technique allows for identification of SETs through statistical analysis. The IRES technique gives insight into the mechanism of a single event strike through various statistical features. Detailed analysis of TID and SETs using IRES technique presented in this chapter is demonstrated in subsequent Chapters IV and V.

CHAPTER IV

TOTAL-IONIZING DOSE ANALYSIS WITH IRES

Introduction

This section presents the response of the RF circuits, designed and fabricated in 130-nm CMOS technology, to Total-Ionizing Dose (TID). This chapter primarily focuses on the analysis of two fundamental circuits: VCO and PLL circuits. In such circuits, degradation in operating frequency, supply current, and phase noise due to increases in leakage current, shifts in threshold voltages, and degraded transconductance results from exposure to TID [27-29]. The VCO circuit is an essential part of the PLL circuit. The PLL circuit is a closed loop system that helps maintain the frequency under lock. A block diagram of a PLL circuit is shown in Figure 11 [26]. The PLL circuit used in this work is a Charge Pump Phase-Locked Loop (CPPLL). The CPPLL sub-circuit consists of a Phase-Frequency Detector (PFD), Charge Pump (CP), Low-Pass Filter (LPF), VCO, and a frequency divider (β) [30]. The PLL used in this work was designed such that the output of open-loop VCO and closed-loop PLL can be measured separately on different output ports.

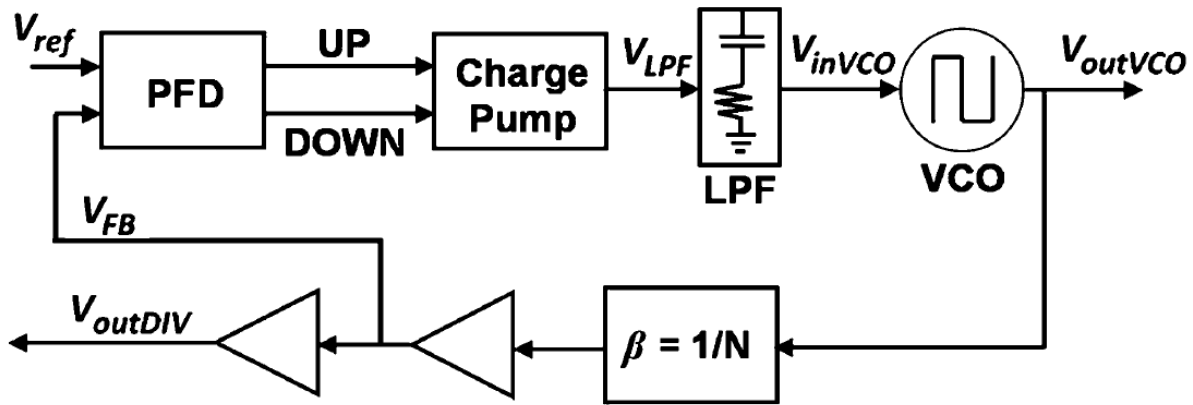


Figure 11

General block diagram of a phase-locked loop circuit consisting of phase-frequency detector (PFD), charge pump (CP), low-pass filter (LPF), voltage controlled oscillator (VCO) , and frequency divider (β) [26]

Factors such as changes in noise and loop stability may cause an eventual failure in the PLL [31]. A real-time measurement of the cause of the change in noise or loop stability can be difficult or impossible until a failure is observed. Minor increasing frequency shifts were measured with increase in TID dose levels, while significant changes in the noise performance were noted [3]. These significant changes in the noise performance will be described with illustration of the IRES images in the next section. Of particular note is the ability to capture the shifts in noise performance without interrogating the internal nodes of the circuit. The need for tracking the parametric shifts due to the risk of sudden failure has been emphasized in a recent work on 32-nm VCO [31]. Figure 12 shows a VCO's phase noise (measured in dBc/Hz) versus TID (measured in krad(SiO₂)) plotted at three different temperatures [31]. It shows that the failure points of VCO operation are dependent upon temperature and TID radiation level. It is also observed that the phase noise is unique at each failure.

The IRES technique captures various statistical measures from measurements of the waveforms of circuits to assess the stability. Instantaneous frequency is calculated from in-situ measurements of time-domain output waveforms of the VCO and PLL. Mean, standard deviation, and variance are the statistical measures used to create the IRES images of the circuit behavior. These statistical measures allow for the evaluation of the effect of TID while capturing multiple key performance metrics at the same time. ML classification is used to identify the operational state of the circuit (e.g., TID level and circuit bias voltage) using the spectral information from the IRES images. The IRES method is demonstrated using VCOs and PLLs designed and fabricated in a 130-nm CMOS process stressed in TID radiation effects.

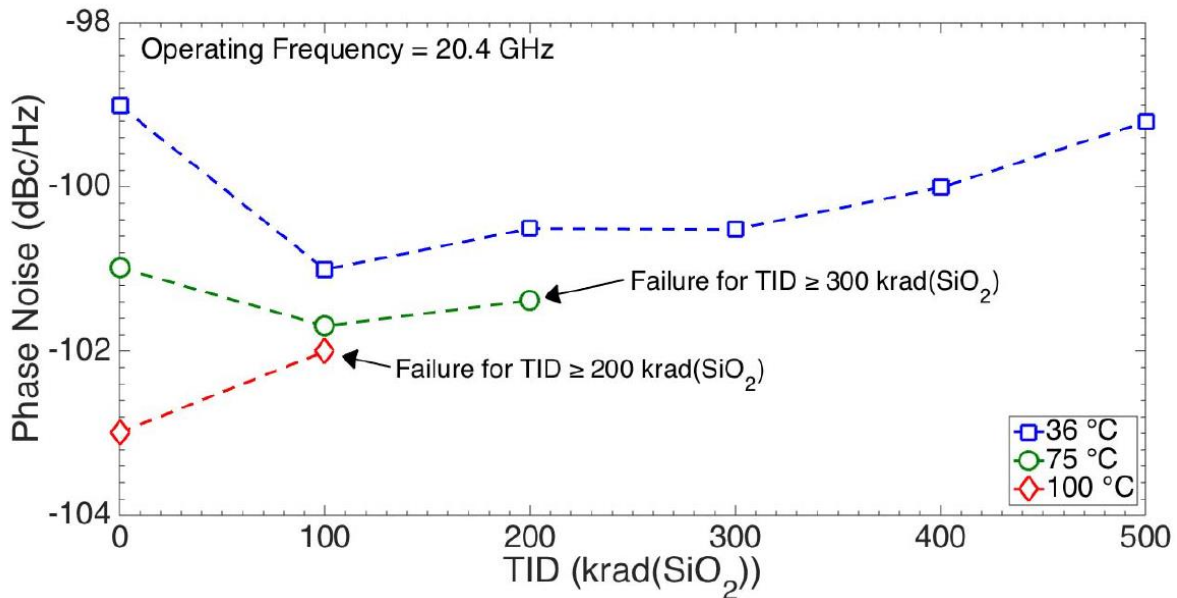


Figure 12

Measured phase noise at 1-MHz offset as function of TID of a 32-nm VCO at a constant frequency of 20.4 GHz [31]

This chapter shows work for development of IRES images generated from circuits exposed to TID radiation. The IRES technique uses subtle and significant differences in the behavior of the circuit waveforms to identify TID exposure. A predictive ML-based model was generated using the data obtained from the generation of the IRES images. The training sets for the ML models were obtained from measured waveforms from the devices with known bias conditions and TID radiation levels. Two dimensional statistical measures (e.g., mean and standard deviation) obtained using IRES were used to create a ML model and prediction accuracy of between 97.5% and 100% was achieved with 400 training samples model checked against 100 testing samples. More details about the prediction accuracy of the ML models will be presented later in this chapter.

Experiment details for VCOs and PLLs designed in 130 nm CMOS process

In this chapter, the utility of IRES for the identification of device degradation using two VCOs and two PLLs designed and fabricated in a 130-nm CMOS process are presented. The VCOs (VCO A and VCO B) are designed to operate in open- or closed-loop configuration. Both VCOs had similar but unique operating performance within a frequency range of 10-350 MHz, a gain of approximately 500 MHz/V, and a linear operating range of 10-225 MHz [32].

Both VCOs and PLLs were irradiated with an Aracor X-ray source at a dose rate of approximately 31.5 krad(SiO₂)/min at Vanderbilt University, Nashville, TN, USA. The VCOs were biased in the OFF state with the VCO control voltage (V_{inVCO}) at 0 V and with a voltage supply (V_{DD}) of 1.2 V. Both devices were irradiated up to 300 krad(SiO₂). V_{inVCO} was swept in steps of 50-100 mV at each radiation dose level (e.g., 10, 20, 40, 100, 200, and 300 krad(SiO₂)). Output waveforms were captured at each step of the V_{inVCO} . The waveforms were captured with a Tektronix DPO7104 Digital Phosphor Oscilloscope at a sample rate of 5 GS/s for each biased

condition and dose level. The experiments were performed at room temperature of approximately 25°C and measurements were made in-situ.

Figure 13 and 14 show the output frequency versus input control voltage V_{inVCO} of VCO A and VCO B, respectively, at TID levels of 0, 40, and 300 krad(SiO_2) [3]. Negligible changes were observed in the gain of the VCOs after the irradiation. An increase in the frequency of approximately 30 MHz was observed for TID level of 300 krad(SiO_2) throughout the linear operating region. This increase in the frequency indicates an increase in the drive current due to the changes in threshold voltage. Figure 15 and 16 show the output frequency versus input control voltage V_{inVCO} of PLL A and PLL B, respectively, at TID levels of 0, 40, and 300 krad(SiO_2) [3]. The VCO linear region was tracked to approximately 225 MHz for the input frequency using the PLLs. An increase of 30 MHz in the frequency was observed for a TID level of 300 krad(SiO_2) over the entire linear operating region. Phase-locking characteristics of the PLL were not degraded due to the TID irradiation.

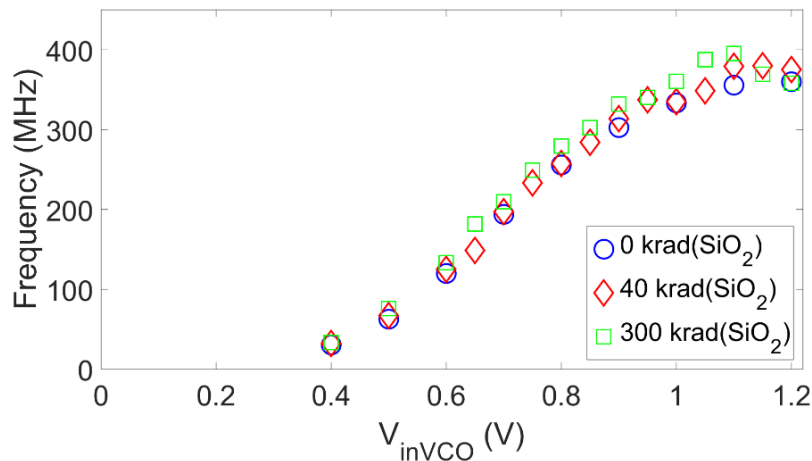


Figure 13

Measured frequency tuning curves showing output frequency versus input control voltage V_{inVCO} of VCO A designed in a 130-nm CMOS technology at TID levels of 0, 40, and 300 krad(SiO_2) [3]

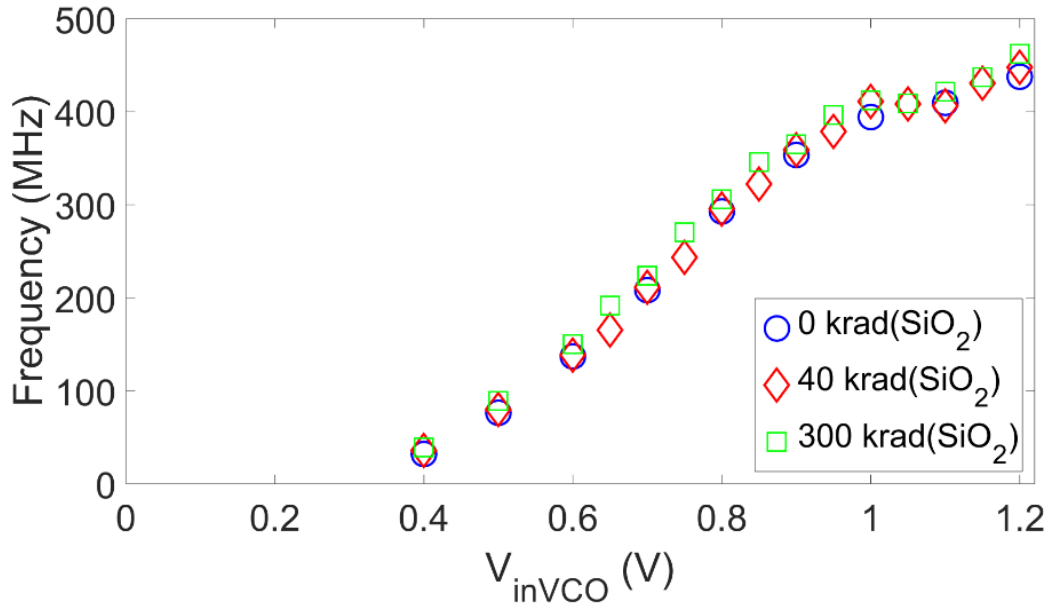


Figure 14

Measured frequency tuning curves showing output frequency versus input control voltage V_{inVCO} of VCO B designed in a 130-nm CMOS technology at TID levels of 0, 40, and 300 krad(SiO₂) [3]

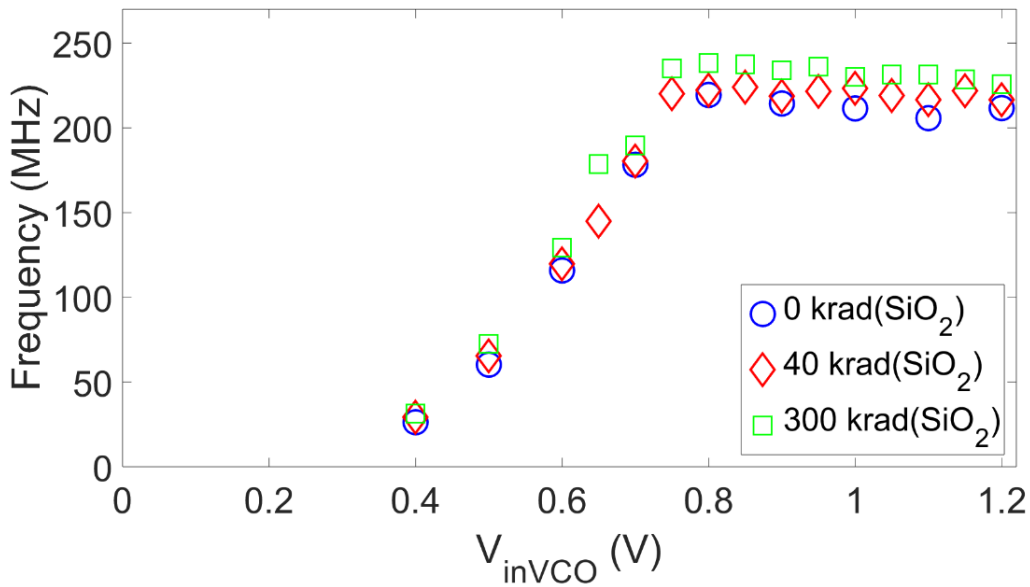


Figure 15

Measured frequency tuning curves showing output frequency versus input control voltage V_{inVCO} of PLL A designed in a 130-nm CMOS technology at TID levels of 0, 40, and 300 krad(SiO₂) [3]

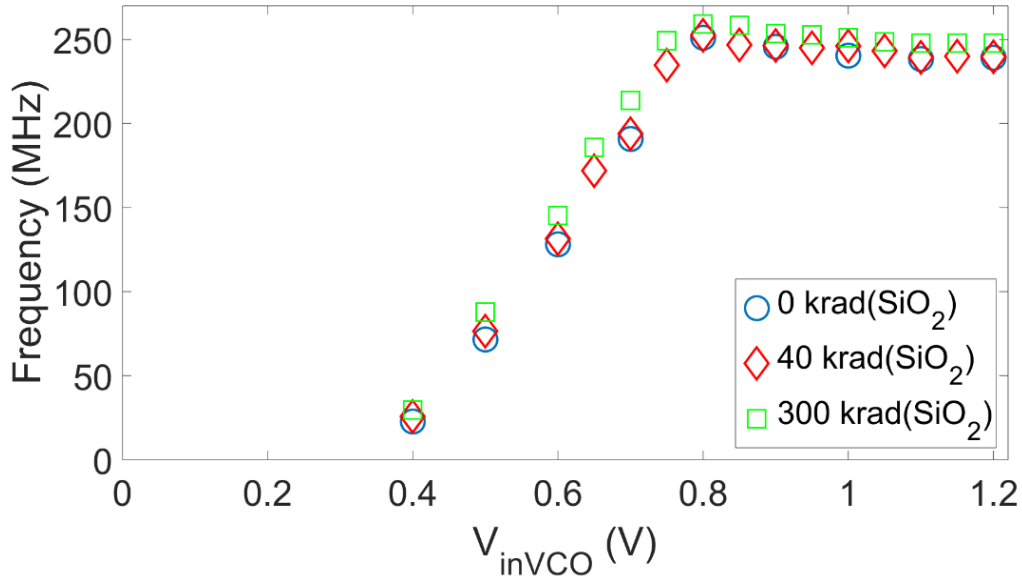


Figure 16

Measured frequency tuning curves showing output frequency versus input control voltage V_{inVCO} of PLL B designed in a 130-nm CMOS technology at TID levels of 0, 40, and 300 krad(SiO₂) [3]

Example output waveforms of the VCO are shown in the Figure 17 [3]. The top waveform and the bottom waveform represent the pre-irradiation (at, 0 krad(SiO₂)) and post-irradiation (at 300 krad(SiO₂)) measurements of VCO A, respectively, with y-axis representing amplitude and x-axis representing time. The nominal frequency of approximately 110 MHz was generated using bias voltage (V_{inVCO}) of 0.6 V.

Figure 18 illustrates the probability density functions of the instantaneous frequency for pre-irradiation at 0 krad(SiO₂) (Blue/Dashed plot) and Post irradiation at 300 krad(SiO₂) (Red/Solid plot) [3]. The plots in Figure 18 show that there is a slight increase in the mean frequency as the TID does level increases and it is indicated by the right shift in the distribution. The variance, post-irradiation, decreases which is indicated by the sharper peak of the distribution and it indicates a decrease in instability (reduced noise) of the circuit. It is quite clear that methods

of analysis using statistical measures can be exploited to further the analysis of behavior of the circuits that are exposed to ionizing radiation. The Gaussian distribution of the frequency was generated from the data obtained at 0.6 V V_{inVCO} voltages for 0 and 300 krad(SiO_2). It should not be assumed that the distribution for other V_{inVCO} voltages will exhibit a similar Gaussian distribution. However, the techniques presented here are applicable to any arbitrary distribution.

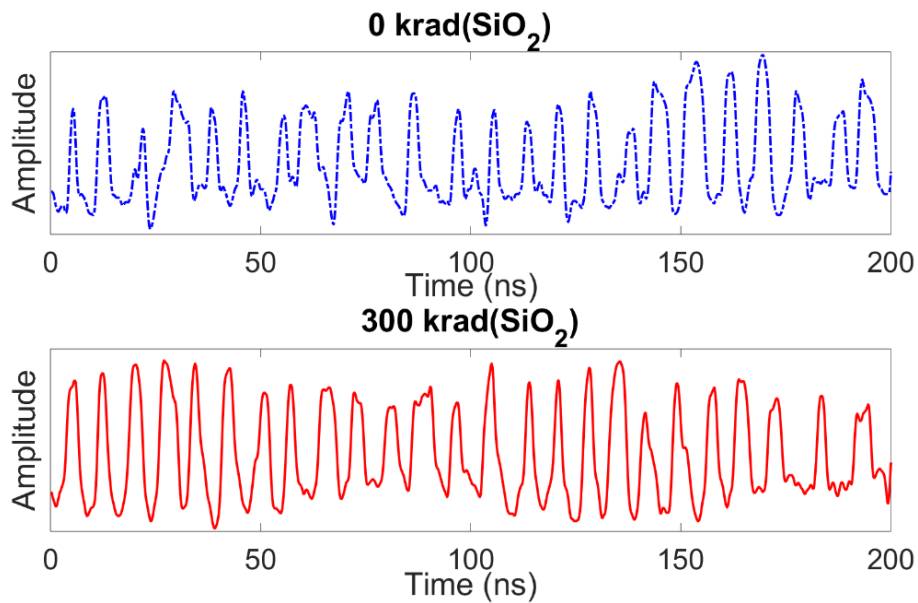


Figure 17

Example waveforms used for generation of IRES: Pre-irradiation 0 krad(SiO_2) (Top plot) and Post irradiation at 300 krad(SiO_2) (Bottom plot) [3]

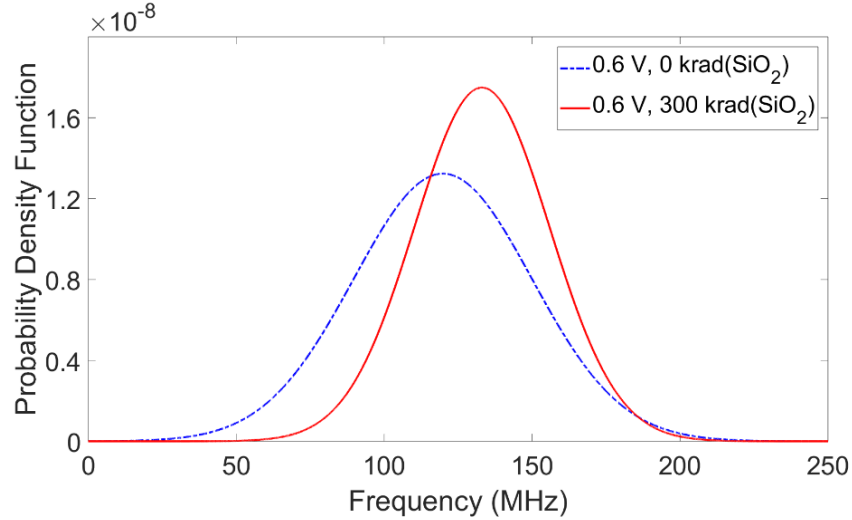


Figure 18

Probability density functions of the VCO's instantaneous frequency: Pre-irradiation at 0 krad(SiO₂) (Blue/Dashed plot) and Post irradiation at 300 krad(SiO₂) (Red/Solid plot) [3]

IRES Image Formation

As mentioned earlier, VCO and PLL output waveforms were obtained for 50 and 100 mV increments of V_{inVCO} within the range of 0.1-1.2 V V_{DD} . A total of 500 waveforms were captured with a Tektronix DPO7104 Digital Phosphor Oscilloscope for each biased condition and radiation dose level. The sampling rate of the oscilloscope was 5 GS/s. A Minimum of 1000 time steps were captured for each waveform. The instantaneous frequency was calculated for each cycle in the waveform and concatenated to create a frequency vector F . The frequency vector was then segmented into 50 blocks and up to five statistical measures (mean, standard deviation, variance, kurtosis, and skewness) were computed for each block. This process of generating a characteristic feature and calculating statistical measures is illustrated in Figure 6. Finally, the block features (i.e., mean, standard deviation, variance, skewness, and kurtosis) were normalized with respect to all statistical measures and all waveforms within an IRES image such that the maximum value of

any measure within an image was 1. These normalized values were used to plot the IRES image using a MATLAB function. The resulting discretized time sequence of statistical measures forms the IRES image and provides a visual indicator of microscopic (micro) transient variation of a feature on the y-axis of the IRES image as well as any macroscopic (macro) shifts in performance on the x-axis of the IRES image.

The entire IRES image formation process is illustrated in the Figure 19. An IRES image of mean instantaneous frequency of VCO output waveform is shown in Figure 19. A waveform captured for 0.6 V V_{inVCO} at 0 krad(SiO_2) was used to generate the IRES image as shown in Figure 19 (A). Figure 19 (B) shows the captured time-domain waveform and Figure 19 (C) shows the zoomed in version of the time-domain waveform. Figure 19 (D) shows the instantaneous frequency vector extracted from the time-domain waveform. This vector was then segmented into 50 blocks and mean of each block was calculated to generate the IRES image shown in Figure 19 (E).

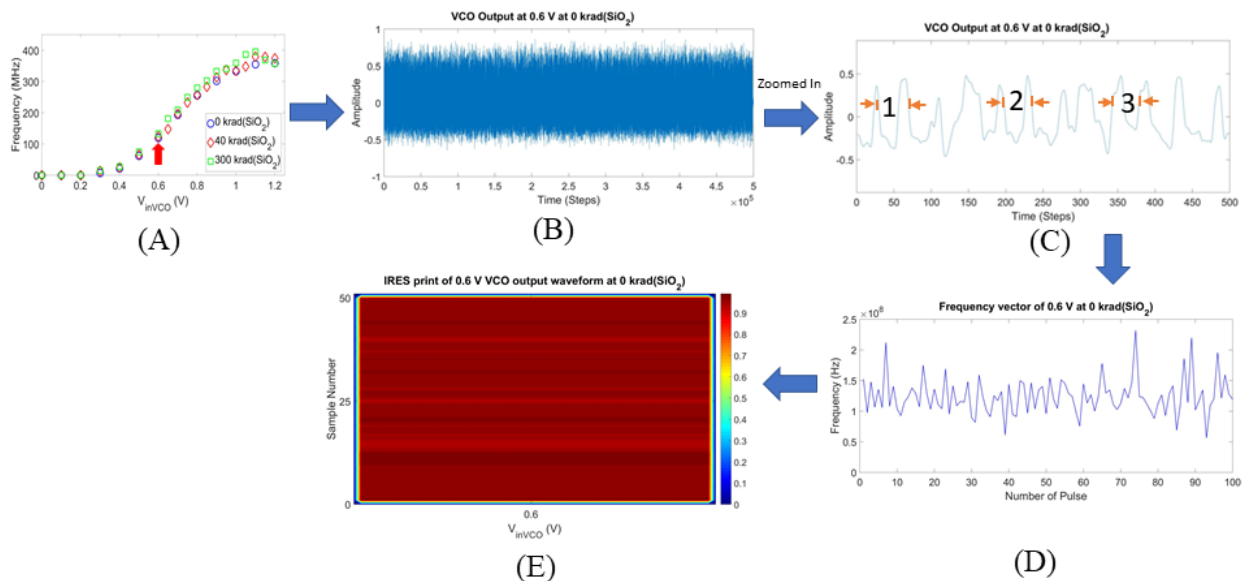


Figure 19

IRES image formation process for TID

IRES images of the frequency for VCO A at frequency values of 62.6, 119.8, and 193.8 MHz corresponding to V_{inVCO} values of 0.5, 0.6, and 0.7 V, respectively, are shown in Figure 20 [3]. The mean and standard deviation statistical measures of the instantaneous frequency are shown in the IRES image. Macro changes due to changes in the value of V_{inVCO} are represented by the x-axis. The width of the vertical bars is arbitrary and used to enhance the ability to visualize the variability. As the value of V_{inVCO} increases, the mean of the frequency increases. It is indicated by a change from a dark tone to a light tone. Micro variations in the measurements are indicated along the y-axis for each block.

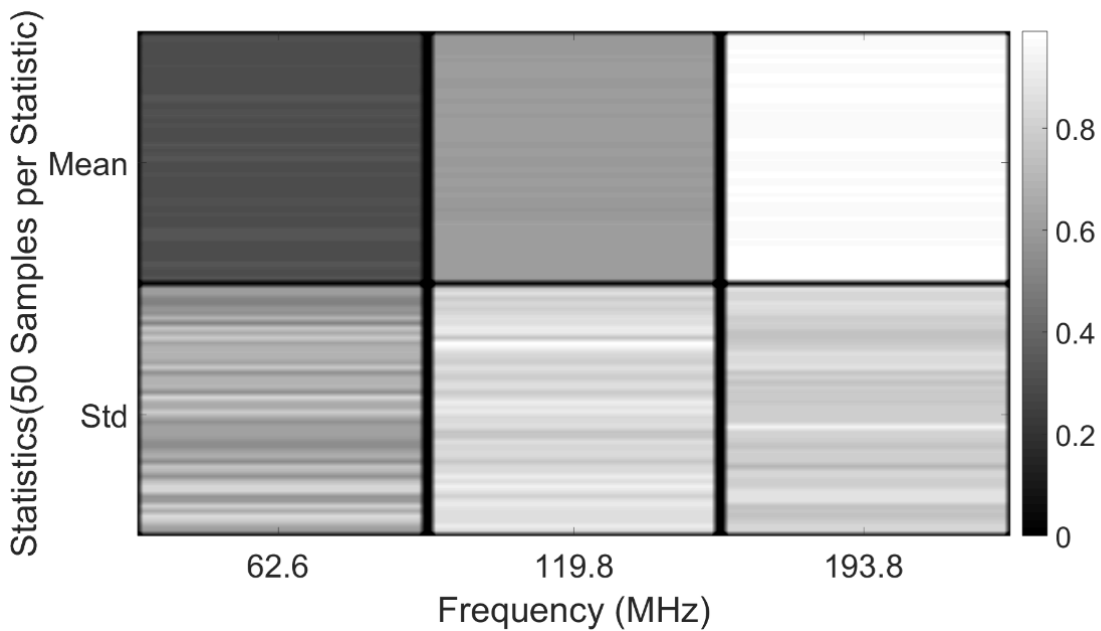


Figure 20

IRES images representing the mean and standard deviation measures of instantaneous frequency of VCO A at frequency values of 62.6, 119.8, and 193.8 MHz for V_{inVCO} values of 0.5, 0.6, and 0.7 V, respectively [3]

There are six IRES images within Figure 20 [3]. Each IRES image consists of 50 blocks. A solid color IRES image would indicate that all 50 measurements were identical, and no noise was measured. However, these IRES images have slight changes in their color, which indicates the presence of noise within a signal. However, each IRES image has similar average color indicating that the standard deviation measure is similar.

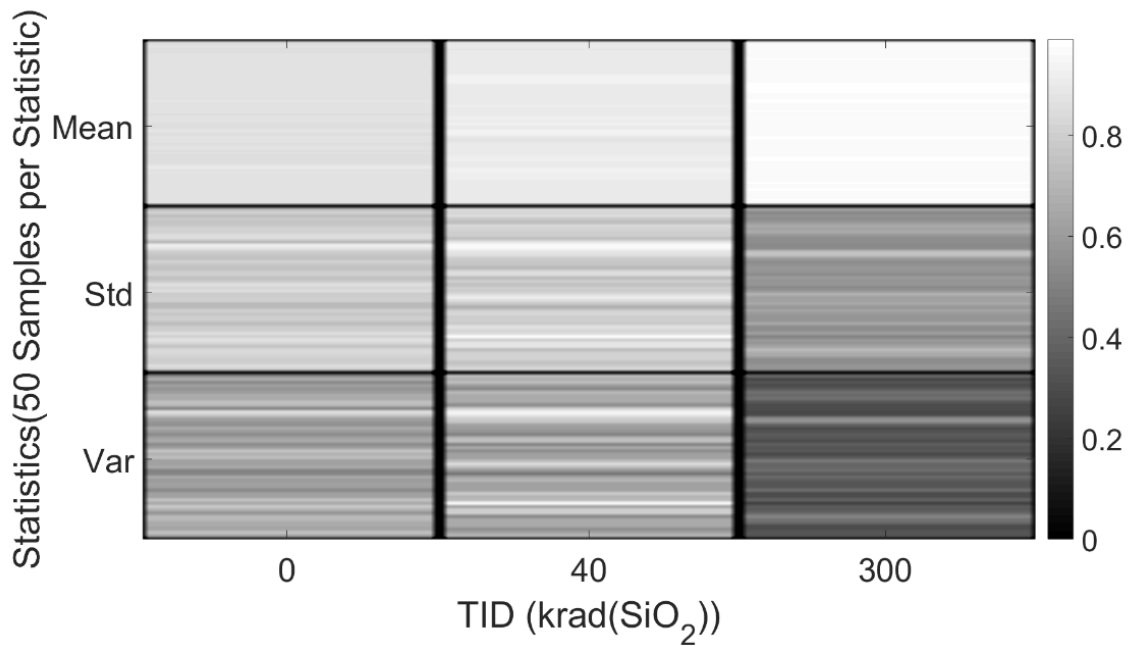


Figure 21

IRES images containing the mean, standard deviation, and variance measures of instantaneous frequency of VCO A at fixed bias of 0.6 V with respect to 0, 40, and 300 krad(SiO₂) [3]

IRES images of VCO A instantaneous frequency with mean, standard deviation, and variance measures at a fixed bias 0.6 V at 0, 40, and 300 krad(SiO₂) TID levels are shown in Figure 21 [3]. Comparing the three blocks of mean statistical measures indicate that the frequency is increasing as the TID level increases. It is indicated by the subtle change in tone from dark to light

of the mean measure block. Standard deviation and variance measure blocks change tone from light to dark indicating the increase in the stability of the circuit output. The change in stability is not significant when going from 0 to 40 krad(SiO₂), but at 300 krad(SiO₂) the change in stability is significant and it indicates a non-linear change in stability.

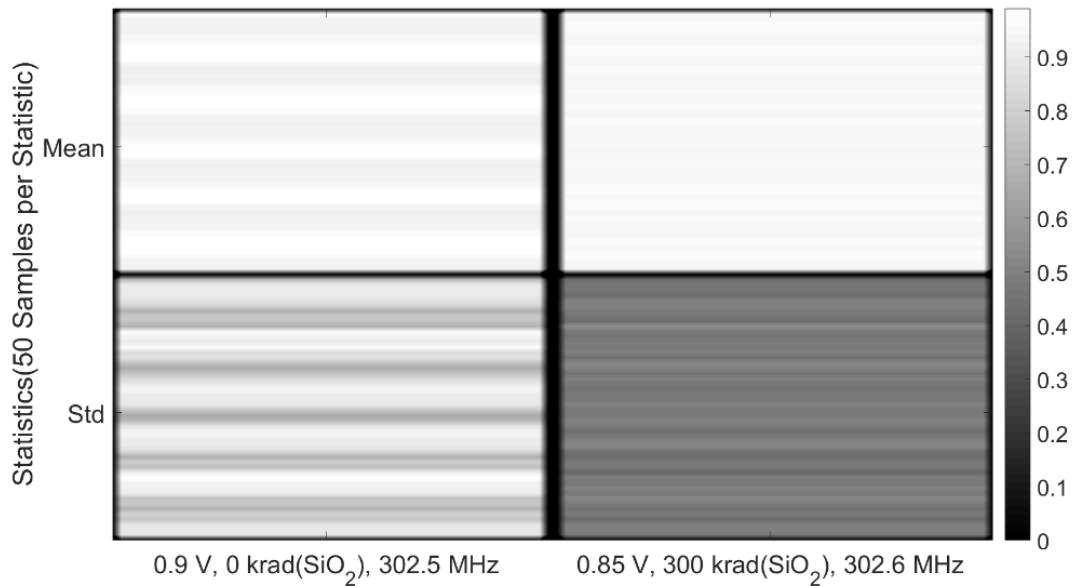


Figure 22

IRES images of VCO A at a fixed frequency of ~302 MHz at 0 krad(SiO₂) (left) and 300 krad(SiO₂) (right). VCO A was retuned to maintain the constant frequency [3]

Figure 22 shows the IRES images of VCO A at a fixed value of approximately at 302 MHz [3]. Increase in TID results in increased in frequency as indicated in Figure 22. Therefore, to compensate for the increased frequency, the PLL decreases the bias voltage to maintain a fixed frequency output. Lower bias currents and different noise characteristics are the results of a decrease in bias voltage. As indicated by the same average color tone in the mean frequency blocks,

the operating frequency is fixed in Figure 22. However, the increased horizontal striping on the y-axis of the IRES image indicates the significant changes in transient variation. Such variation could be an indicator of changes in loop stability parameters, which can impact lock time, noise, natural frequency, and frequency pull-in characteristics. Standard deviation of the instantaneous frequency decreases with an increase in TID level to 300 krad(SiO₂). Figure 23 shows the IRES images of PLL A at a fixed value of approximately at 178 MHz [3]. Similar to the results in Figure 20, the stability of the circuit output is increased indicated by the change from a light tone to a dark tone of the standard deviation.

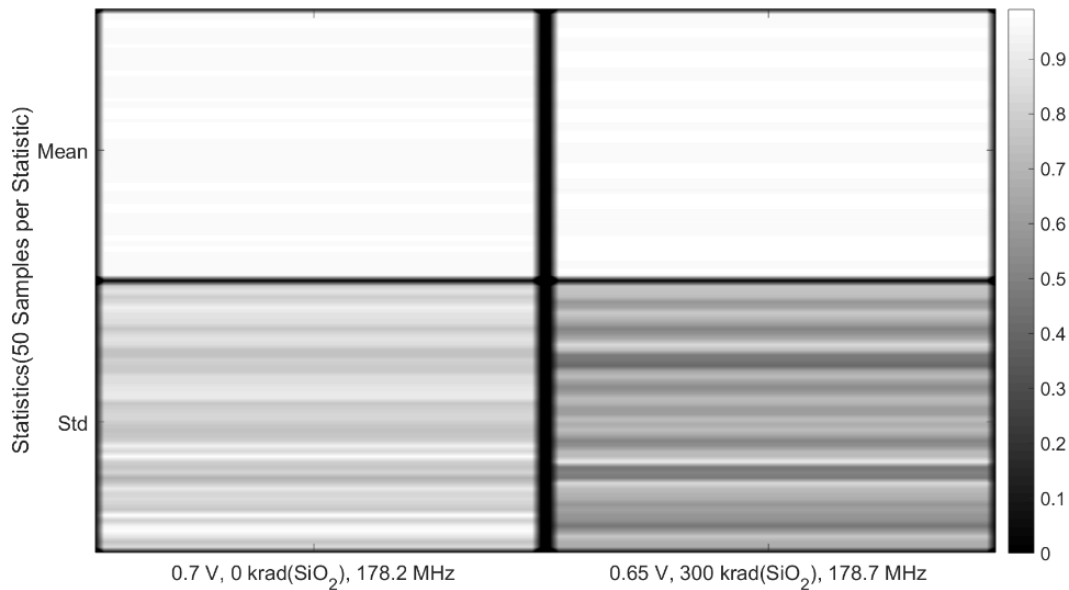


Figure 23

IRES images of PLL A at a fixed frequency of ~178 MHz at 0 krad(SiO₂) (left) and 300 krad(SiO₂) (right). PLL A was re-tuned to maintain the constant frequency [3]

Phase noise exhibits a nonmonotonic response with respect to TID due to the competing effects of the bias current, operating frequency, signal power, and transistor bias points, all of which vary with TID as shown in Figure 12 and discussed in [31]. As indicated by smaller standard deviations of instantaneous frequency shown in Figures 22 and 23, the observed improvement in stability with increased TID indicates that the increase in bias current, and thus, signal power dominates the observed improvement in noise. Additional increases in TID may result in eventual degradation of noise characteristics and loop stability, according to [29].

Classification of TID levels using ML

Figures 22 and 23 represent the IRES images generated from the mean and standard deviation statistical measures of instantaneous frequency. They show promise in identification of TID levels by using the difference between statistical measures generated from pre-rad and post-rad conditions. Operational states of the circuit, such as TID levels and tuning voltage, were identified using mean and standard deviation statistical measures from the IRES image data. MATLAB was used to classify the TID levels and VCO tuning voltage levels with a 2-D Linear Discriminant (LD) model [33].

A total of 500 waveforms were measured for each bias condition and dose level for VCO A, VCO B, PLL A, and PLL B on the same chip. Each waveform consisted of 1000 time-steps. As the circuits were on the same chip, the measurement setup for all four circuits was identical and included the same measurement noise characteristics as described in the IRES image Formation section. For each block, mean and standard deviation statistical measures of the instantaneous frequency vectors were computed. The dimensionality of the LD model can be increased by adding more statistical measures.

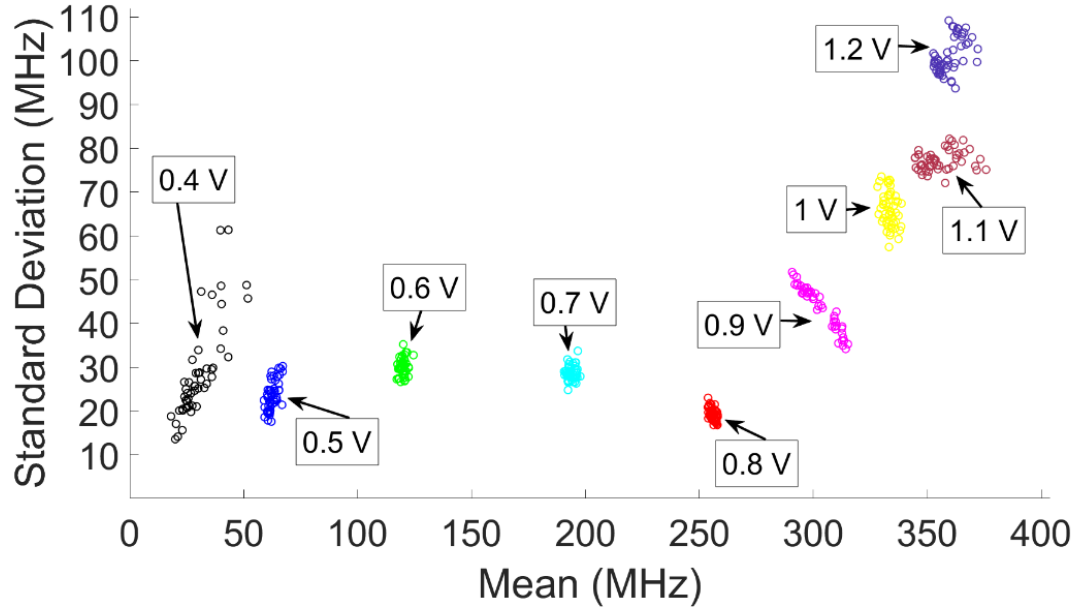


Figure 24

Scatter plot of standard deviation versus mean of the instantaneous frequency feature for VCO A at V_{inVCO} values between 0.4 and 1.2 V at 0 krad(SiO_2) [3]

The scatter plot of the standard deviation versus the mean of the instantaneous frequency feature for V_{inVCO} values between 0.4 to 1.2 V at 0 krad(SiO_2) is shown in Figure 24 [3]. The data samples are clustered based on the input bias voltage as shown in Figure 24. Due to changes in stability, the clusters of data samples have different spreads in the x- and y- dimensions. As seen in Figure 24, below the value of 0.4 V and above the value of 0.8 V the standard deviation increases indicating the increased noise and non-optimal operation of the circuit outside of the linear operating range.

A two-fold cross-validation method was used to identify a LD model that results in the best prediction accuracy using MATLAB's classification learner application [34]. In two-fold cross-validation, the training data is divided into two partitions. The first partition is used by the machine learning algorithm to develop the LD model, while the second partition is held out to validate how

well the developed model predicts the correct class using a set of 'unseen' data. The process is repeated, but with the second partition used for LD model development and the first for validation of model performance. The results of each fold are averaged to create the final LD model. A 'blind' test data set is used to find the prediction accuracy of the LD model. Initially, the LD model was developed using 400 training samples drawn from the 500 total available samples. The remaining 100 were set aside to serve as a 'blind' test set of the final LD model and provide the prediction accuracy presented in Tables I and II. The number of samples comprising the training set was varied from 30 up to 400 measured samples. Varying of the number of training samples was done to find the minimum number of training set size from which a LD model can be created and used to predict the classification values. The 'blind' test set samples were projected onto the line generated from the trained LD model to calculate the prediction accuracy of the classification. MATLAB's "*predictFcn*" function was used to obtain predicted classification values. These predicted classification values were checked against the actual values of the training set samples and were used to determine the percent classification accuracy of the LD models. The prediction accuracy of the LD model for the prediction of the TID level and bias voltage for VCO A with respect to the size of the training set is shown in Table I. It also includes the size of the testing data set.

Table 1

Classification of TID levels for VCO A [3]

Training Set size (Samples)	Testing Set size (Samples)	Classification Accuracy
400/300/200 100/50	100/200/300 400/450	100 %
40	460	96.47 %
30	470	94.95 %

As seen from Table I, the classification accuracy of 100% for prediction of TID level and tuning voltage for training set sizes of 50 samples and higher using the LD classification model was achieved [3]. Data were included for V_{inVCO} values between 0.5 and 0.8 V and TID levels between 0 and 300 krad(SiO_2). For the training set size of 40 samples, the classification accuracy was 96.47%. For a training set size of 30 samples, the classification accuracy was 94.95%. The drop in the prediction accuracy was because of the proximity of the radiation data near the pre-irradiation data for a V_{inVCO} value of 0.5 V. It can be seen in the Figure 25, where a scatter plot of the standard deviation versus the mean indicate the cluster behavior [3].

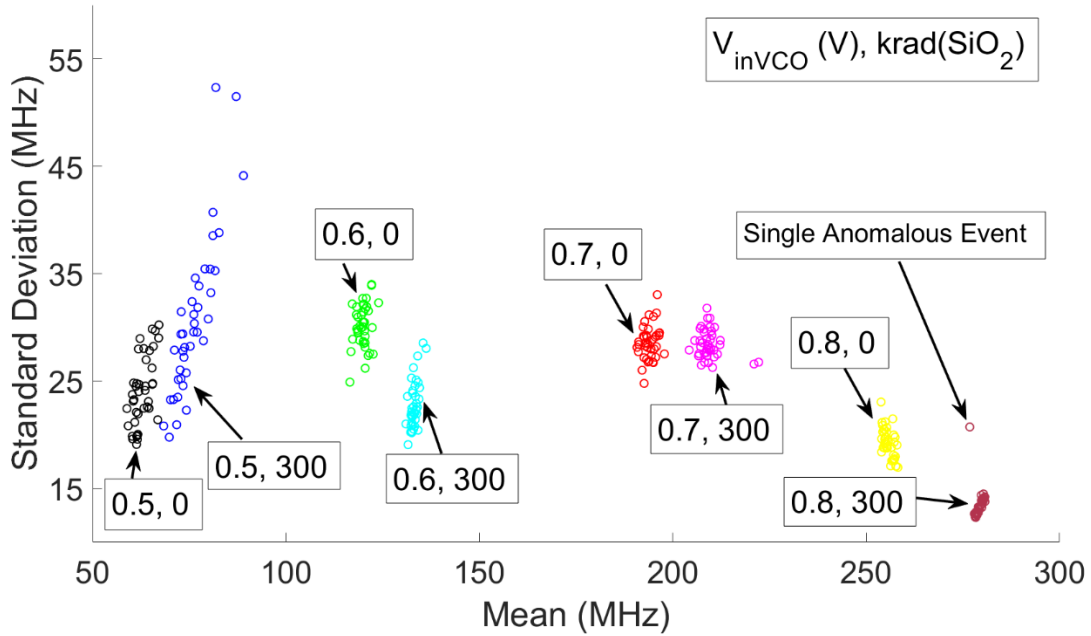


Figure 25

Scatter plot of the standard deviation versus mean of the instantaneous frequency feature for VCO A at V_{inVCO} values between 0.5 and 0.8 V at 0 and 300 krad(SiO₂). The data clusters are labeled with a unique V_{inVCO} and TID pair [3]

The IRES images provide information about the circuit timing and can identify the occurrence of anomalous events. For example, in Figure 25 the 0.8 V / 300 krad(SiO₂) cluster appears to consist of a single anomalous event represented by one circle indicator away from its cluster. This anomalous event can be seen from Figure 26 as lightly toned horizontal bar in the lower right standard deviation feature block.

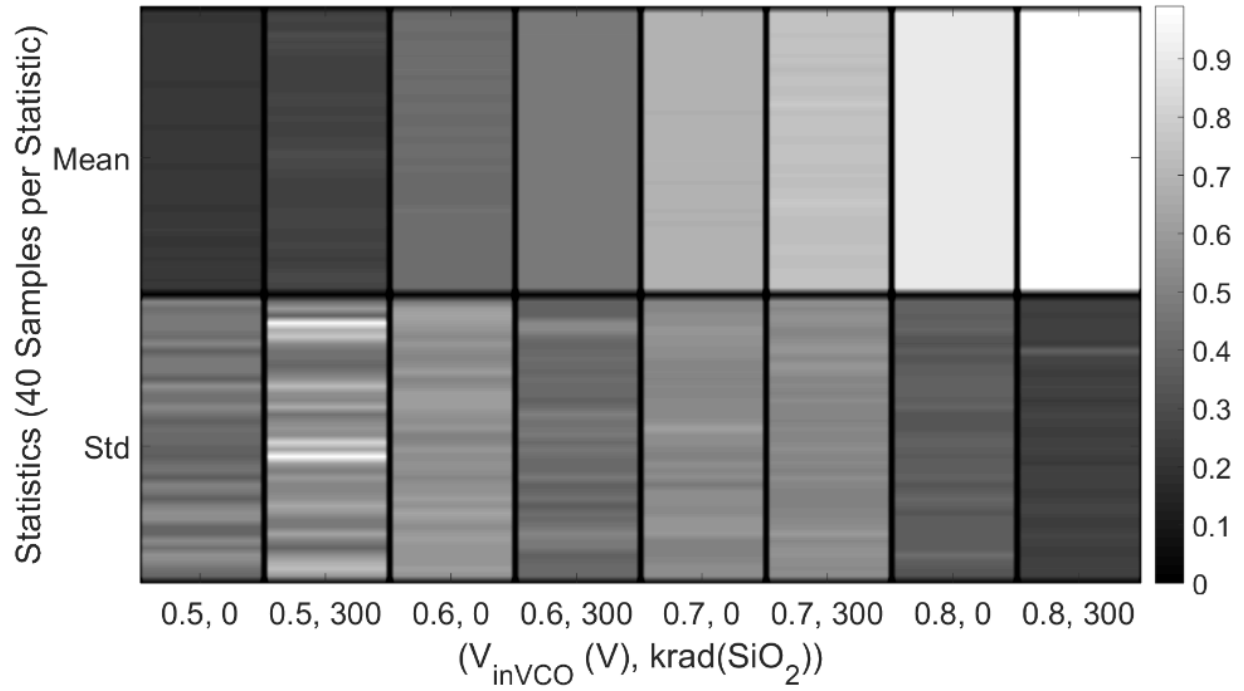


Figure 26

Corresponding IRES images (40 samples) of standard deviation and mean of the instantaneous frequency for VCO A at various V_{inVCO} and TID pairs for the scatter plots shown in Figure 25 [3]

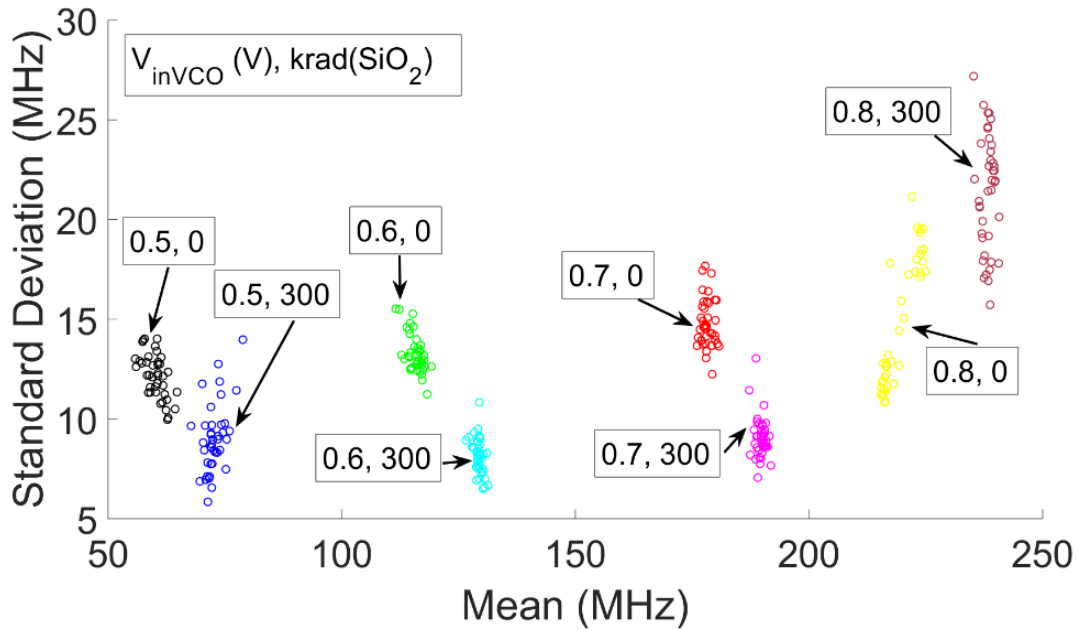


Figure 27

Scatter plot of the standard deviation versus mean of the instantaneous frequency feature for PLL A at V_{inVCO} values between 0.5 and 0.8 V at 0 and 300 $\text{krad}(\text{SiO}_2)$. The data clusters are labeled with a unique V_{inVCO} and TID pair [3]

Figures 27 and 28 show the scatter plot and corresponding IRES images of the standard deviation versus mean statistical measures of the instantaneous frequency of PLL A, respectively [3]. Each statistical measure block represents one IRES image within Figure 28. The data samples of V_{inVCO} values between 0.5 and 0.8 V at 0 and 300 $\text{krad}(\text{SiO}_2)$ are shown in both figures. The PLL loses the lock on the reference frequency at V_{inVCO} values greater than 0.8 V. This instability occurs at a V_{inVCO} boundary value of 0.8 V and it can be seen as a nominal increase in the standard deviation values of the samples as well as increased spread in the measurements. The last two columns of the IRES images, in Figure 28, represents the loss of lock of the PLL A.

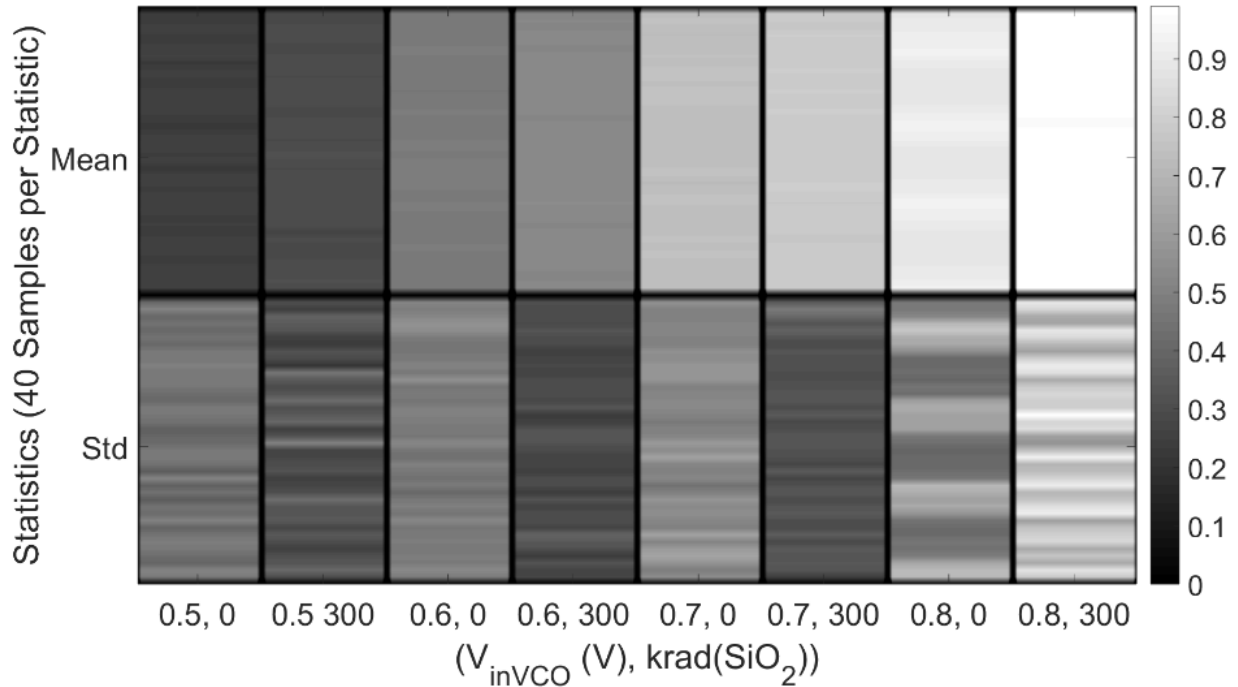


Figure 28

Corresponding IRES images (40 samples) of standard deviation and mean of the instantaneous frequency for PLL A at various V_{inVCO} and TID pairs for the scatter plots shown in Figure 27 [3]

VCO B and PLL B were analyzed similar to the VCO A and VCO B analysis. Figure 29 shows the scatter plots of the mean and standard deviation statistical measures for the instantaneous frequency of VCO B [3]. The general behavior of VCO B with respect to the changes in the bias voltage and the ionizing dose is similar to that of VCO A and PLL B. But, as seen from the Figure 29, the VCO B exhibits increased noise and instability. VCO B being fabricated on the same chip exhibited unique noise characteristics due to manufacturing process variability. Increased variability in tone within the IRES images and wider spreads in the measurements result in misclassification.

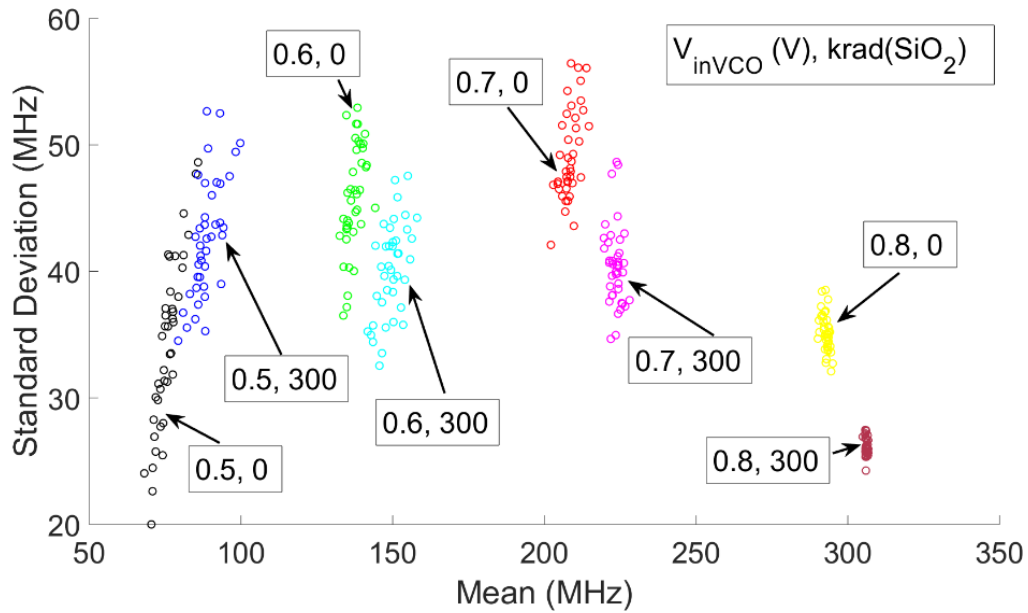


Figure 29

Scatter plot of the standard deviation versus mean of the instantaneous frequency feature for VCO B at V_{inVCO} values between 0.5 and 0.8 V at 0 and 300 krad(SiO_2). The data clusters are labeled with a unique V_{inVCO} and TID pair [3]

Table 2

Classification of the TID levels for both VCOs and PLLs [3]

VCO/ PLL	Training Set size (Samples)	Testing Set size (Samples)	Classification Accuracy
VCO A/ PLL A	400	100	100 %
PLL B			
VCO B	400	100	97.5 %

The prediction accuracy of the classification for each device with a fixed training set size of 400 measured samples and a testing set size of 100 measured samples is shown in Table II [3]. The decrease in prediction accuracy of TID levels and values of V_{inVCO} to 97.5% for VCO B is

due to the proximity of the data samples because of the noisy characteristics of the circuit. The prediction accuracy of TID levels and values of V_{inVCO} for VCO A, PLL A, and PLL B was 100%.

Chapter Conclusion

This chapter presents a non-invasive technique, termed Ionizing Radiation Effects Spectroscopy, for measurement of TID radiation effects in electronics. In this chapter, IRES is used to analyze the effects of TID and bias voltage on the operation of RF circuits such as a VCO and PLL. TID radiation is a cumulative effect and the effects on the circuit operation can be seen after a long-term radiation exposure. The circuits can fail to operate after seeing certain level of TID radiation [31] which can put the satellite system at risk if preventive measures are not taken before the circuit passes a certain level of TID radiation.

This technique allows for monitoring of similar situations in which a circuit is at risk of failure. The ability to identify the dose levels as well as bias voltage with higher level of certainty is of particular note. A two-dimensional ML model using statistical features calculated from the non-invasive output waveforms measurements of the RF circuits helps identify the circuit's operational health.

ML classification is used to perform an *in-situ* prediction of the TID level and circuit bias voltage with a single measurement. The TID response of VCO and PLL circuits designed and fabricated in a 130-nm CMOS technology is presented. In response to the TID radiation up to 300 krad(SiO₂), the RF circuits (VCO and PLL) show a shift of 30 MHz in frequency. It is important to note that the behavior observed in the circuits used for this experiment is limited to the specific technology and circuit design. Other circuit designs and device fabrication parameters may exhibit different response to the similar TID radiation exposure described in this work. The IRES

technique for the TID radiation enhances typical measurements by capturing global parametric shifts as well as transient variations. Transient variation analysis using IRES will be explored in more depth in next chapter.

IRES provides visualization of statistical measures such as mean and standard deviation of the instantaneous frequency. The IRES features were then used to classify the operational state of the circuit with an LD ML model. For the training set sizes of 50 and greater, a prediction accuracy of between 97.5% and 100% was achieved. The prediction accuracy was decreased to 96.47% and 94.95% with lower values of training set samples of 40 and 30, respectively. IRES for TID radiation shows promise in radiation dosimetry applications as well as in-situ monitoring of device and circuit operational health when internal nodes are difficult or impossible to monitor.

CHAPTER V

SINGLE EVENT TRANSIENT ANALYSIS WITH IRES

Introduction

The work presented in the previous chapter demonstrated that IRES can be used to determine the steady-state operational conditions (i.e., the bias voltage and TID exposure level) using machine learning [3]. This chapter presents utilization of the IRES technique for characterization of transient phenomena in devices and integrated circuits. The presented method uses IRES for characterization of SEE with a non-invasive methodology that allows for identification of SET vulnerable circuit nodes. These circuit nodes are identified based on statistical measures of SET features identified in IRES images.

Data presented in Figures 25 and 26 indicate that the IRES methodology can be useful in identifying transient anomalies. This can be seen by the outlier representing the single anomalous event represented by one circle away from its cluster (0.8 V, 300 krad(SiO₂)) and a lightly toned horizontal bar in the lower right standard deviation feature block in Figures 25 and 26, respectively. Standard measurement approaches for identification of transient events include measurement of magnitude of voltage and the length of the event [35, 36]. Sometimes, it is very difficult to interrogate the internal node of the circuit without disturbing the operations of the circuit. The IRES technique does not require the interrogation of the internal nodes of the circuit as the technique solely requires the output waveform of the circuit. The ability to quantify the strike location of the SE is of particular note. Subsequent sections of this chapter will provide IRES

images of different strike locations with unique spectral information of each strike location. As mentioned in the previous chapter, the IRES technique for TID radiation used instantaneous frequency of the signal as the signal metrics. Similarly, this chapter will use cycle-to-mean jitter and instantaneous frequency as the signal metrics.

TID radiation effects were easily identifiable with minimum of two-dimensional statistical measures (i.e. mean and standard deviation). This chapter involves multi-dimensional statistical measurements with more than one signal metrics (i.e. mean, standard deviation, variance, skewness, and kurtosis of the cycle-to-mean jitter and mean of the instantaneous frequency). This allows for understand the underlying radiation effect mechanism of the SET on a circuit, as IRES uses a mechanism of processing sampled data using statistical features which allows for identification of spurious events even in the noisy waveforms.

This chapter presents analysis of SETs through IRES technique for both experimental data and simulation data obtained for a PLL circuits. The experimental data was obtained two-photon absorption laser experiments on PLL circuits designed and fabricated in a 130 nm CMOS technology. The simulation data was obtained from a PLL behavioral model designed with Verilog-A.

Experimental Analysis with IRES

To obtain experimental data, Two-Photon Absorption (TPA) laser excitation experiments [26, 37-40] on a PLL circuit designed and fabricated in a 130 nm CMOS technology [26, 32] were used for characterization of SETs using IRES.

Single transient perturbations were injected into the PLL circuit using laser-induced carrier generation based on TPA using high peak power femtosecond pulses at sub-bandgap optical

wavelengths [37-40]. The Device Under Test (DUT) was mounted on a motorized xyz translation platform with 0.1 μm resolution. Optical pulses are focused through the wafer onto the front surface of the DUT with a 100 \times microscopic objective, resulting in a near-Gaussian beam profile with a typical diameter of approximately 1.6 μm at focus [39]. Because the carrier deposition varies as the square of irradiance (I^2) [37-39], this corresponds to a Gaussian carrier density distribution with an approximate diameter of 1.1 μm (full-width-at-half-maximum). These experiments were performed at room temperature.

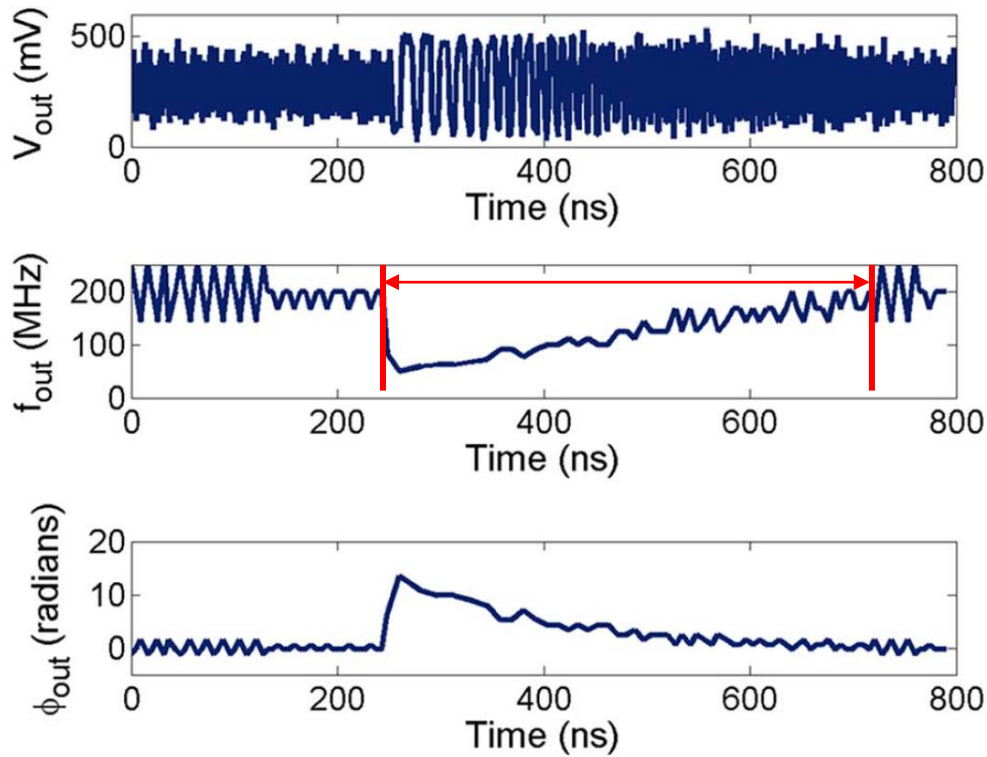


Figure 30

Example of an output frequency and phase transient following a laser perturbation on the CP sub-circuit of the PLL fabricated in a 130 nm CMOS technology. The PLL was operating at 200 MHz. Following the laser strike with incident energy of 5.4 nJ, the output frequency was reduced to approximately 50 MHz, thus increasing the output phase displacement to approximately 15 radians. The recovery time of the PLL was over 200 ns and it is indicated by red arrow in the middle plot of output frequency (f_{out}) versus time [26]

A Tektronix-TDS5104 oscilloscope was used to characterize output transients following laser strikes within the PLL. The output waveforms were sampled at 625 MS/s. The oscilloscope was set to trigger on the rising edge of the pulsed laser sync pulse, and the FastFrame feature was utilized to capture multiple transients (10 transient perturbation) per injection location [26]. Figure 30 shows an example transient where the output voltage magnitude V_{out} , frequency f_{out} , and phase ϕ_{out} versus time plots of a PLL output following a laser perturbation in the CP sub-circuit is provided. The PLL was operating at 200 MHz. Following the laser strike with incident energy of 5.4 nJ, the output frequency was reduced to approximately 50 MHz, thus increasing the output phase displacement to approximately 15 radians. The drop in the frequency can be seen in the middle plot of the Figure 30. The recovery time of the PLL was over 200 ns [26]. Initial lock time of the PLL is similar to the recovery time. Detailed IRES methodology and generation of the IRES spectrograms for SETs is presented in [4].

Figure 31 shows a 2D spatial map of the region of the CP sub-circuit in the PLL. The highlighted regions are sensitive regions in the CP sub-circuit. Each pixel in the x-y plane represents the average phase displacement of 10 transient perturbations [26]. The image provides important information about the most sensitive locations on the circuit at the output of the CP circuit.

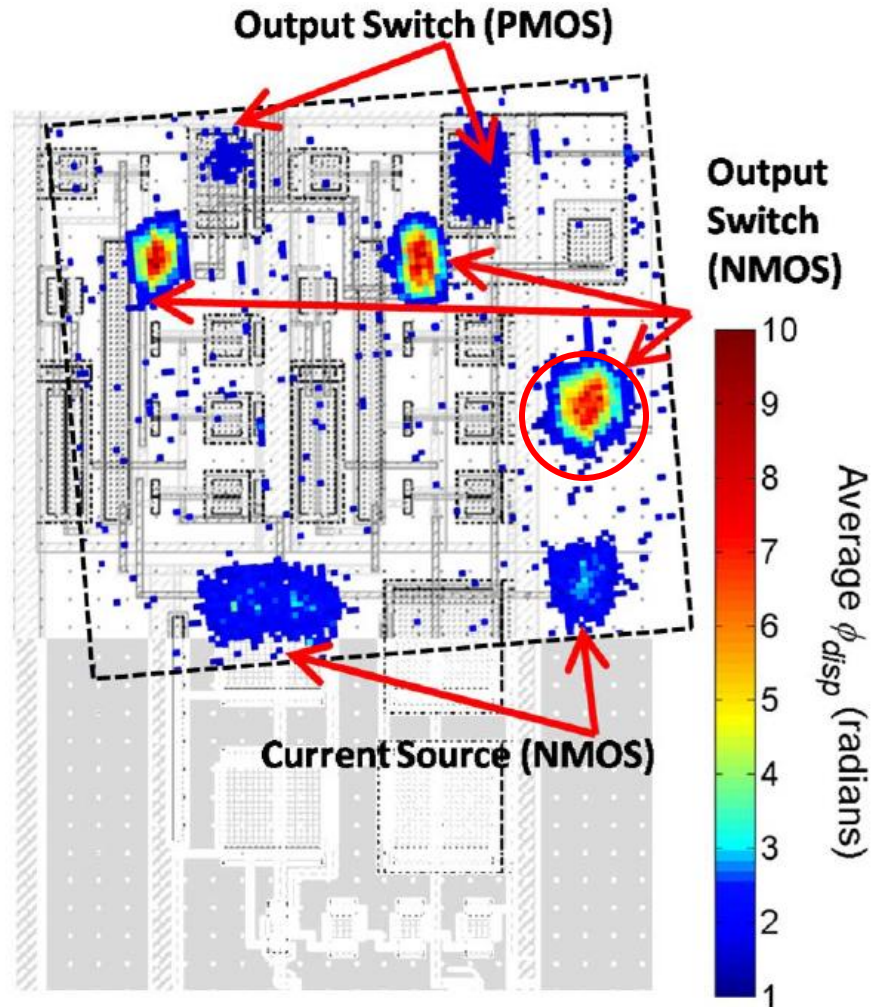


Figure 31

An image of the output phase displacement versus x-y location fused to the layout image of the CP sub-circuit. The experiment was performed using laser TPA at a step size of $0.2 \mu\text{m}$ [26]

Figure 10 illustrates the window sampling process where statistical measures are computed for each window. The statistical measures include the mean, standard deviation, variance, skewness, and kurtosis of the cycle-to-mean jitter feature. The mean of the instantaneous frequency is also calculated for each window. Once the statistical measures are computed within each window and normalized such that the maximum value of any measure is 1, they are concatenated

to form a column vector. These column vectors are arranged as a time-sequence, forming an IRES image of the SET characteristic behavior.

Experimental data was contained in the time-domain waveforms where each waveform consisted of minimum of 500 timesteps. Cycle-to-mean phase jitter was extracted for each clock cycle in the waveform and used to create a vector. A window of the first 15 cycle data (i.e., cycle-to-mean jitter computed from cycles 1 to 15) was used to calculate the statistical measures for the first window. The sampling window was then shifted to the right by one cycle and the process of calculating the statistical measures was repeated such that the second window consisted of cycles 2 to 16, and the third window consisted of cycles 3 to 17, and so until the window traversed the entire length of clock cycles.

An IRES image for a SET is shown in Figure 32. The SET was originated at the output node of the CP sub-circuit. The SET in the Figure 32 was arranged in a manner such that the start of the SET is on the left side of the IRES image. The labels on the y-axis represent the statistical measurements used to characterize the SET strike. The height of each statistical measure was arbitrary and was increased to increase the visibility of the SET characteristics in x-axis. Figure 32 shows a SET generated while the PLL was running approximately at 150 MHz. The incident energy of the laser was approximately 5.4 nJ. This IRES image was generated from the SET data originated from a SE strike on a NMOS transistor at the output node of the CP of the PLL and this node can be seen in the Figure 31 indicated by a red circle around the strike region. The efficiency of IRES technique to isolate the strike location only using the spectral information generated from the output waveforms is of particular note.

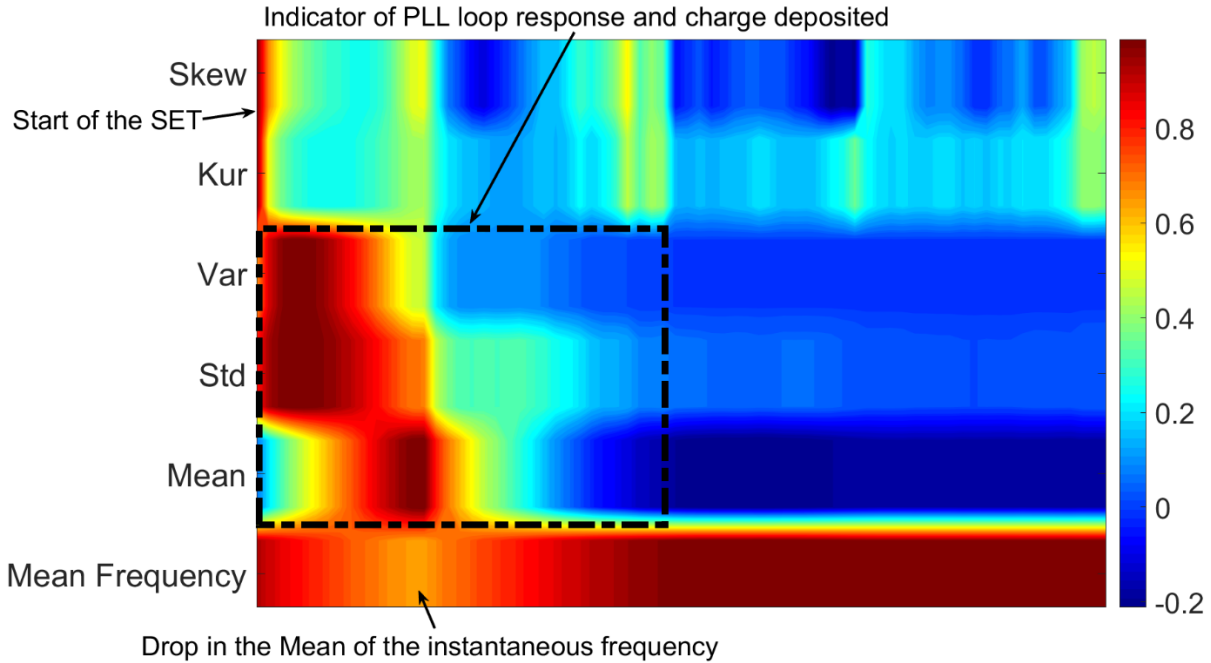


Figure 32

IRES image generated from experimental data obtained for SET originated from a strike on a NMOS transistor at the output node of the CP of the PLL . SET was observed when the PLL was running approximately at 150 MHz. A total of 15 samples were used for one window and statistical measures were calculated. The shift value of the window was 1. Skewness and kurtosis of the cycle-to-mean jitter show the onset of the SET indicated by red color column on the top left corner of the image. Radiation and circuit response mechanisms are characterized by the statistical measures of the cycle-to-mean jitter

As seen from the Figure 32, skewness and kurtosis of the cycle-to-mean jitter represent abrupt changes in the transient behavior. The start of the SET can be seen as a sudden change in the sample value with red color. Other statistical measures such as variance, standard deviation, and mean of the cycle-to-mean jitter are measures of the charge deposition and collection processes as well as the PLL loop response. The time between the peak skewness and the peak mean represents the initial charge collection, whereas the magnitude of the variance represents the severity of the event (i.e., total energy transferred). As seen from the IRES image, mean, standard deviation and variance return to value of 0 following the maximum mean value. It indicates that

the circuit has begun to recover following the SE strike. The mean of the instantaneous frequency (i.e., bottom statistical measure on the IRES image) indicates the overall loop characteristic response as it shifts, indicated by the color change from orange to red. It also shows the gradual recovery of the loop after the SE strike. In addition to showing the magnitude and the length of the time of the SE strike which standard measurement techniques can provide, SET quantifies other features such as charge deposition and collection, severity of the event.

Figure 33 shows the IRES image generated from experimental data obtained for SET originated from a strike on a PMOS transistor at the output node of the CP of the PLL. SET was observed when the PLL was running approximately at 150 MHz. Here, skewness of the cycle-to-mean jitter shows the onset of the SE illustrated by blue color column at the top left of the Figure 33. Comparing the spectral characteristics of Figure 32 with Figure 33 will provide visual information for identifying location of the SE strike. Both images are fundamentally different, but the cause of generating the SET is similar which is caused by the laser experiment with similar incident energy.

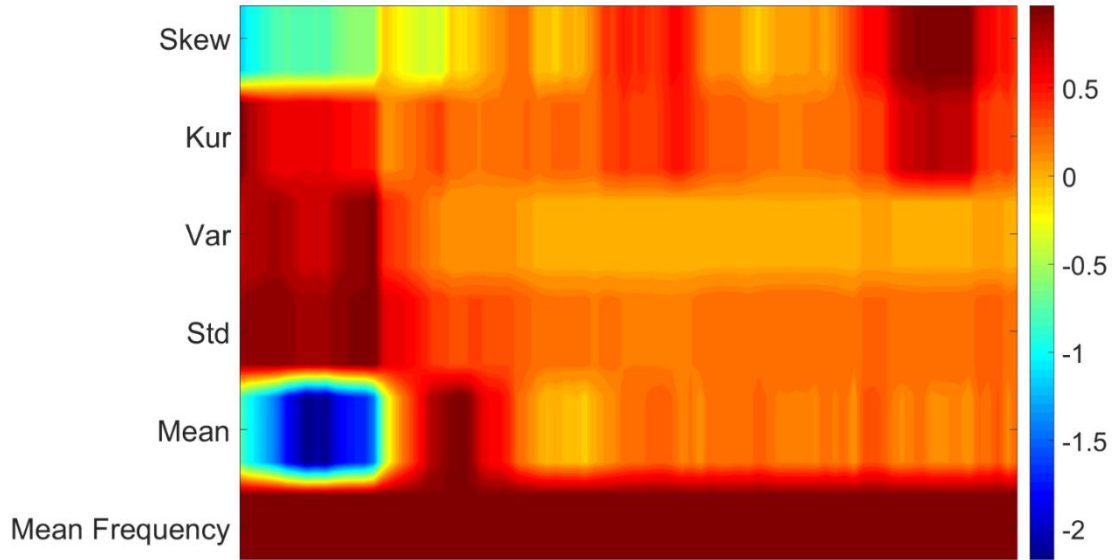


Figure 33

IRES image generated from experimental data obtained for SET originated from a strike on a PMOS transistor at the output node of the CP of the PLL . SET was observed when the PLL was running approximately at 150 MHz. Total of 15 samples were used for one window and statistical measures were calculated. The shift value of the window was 1

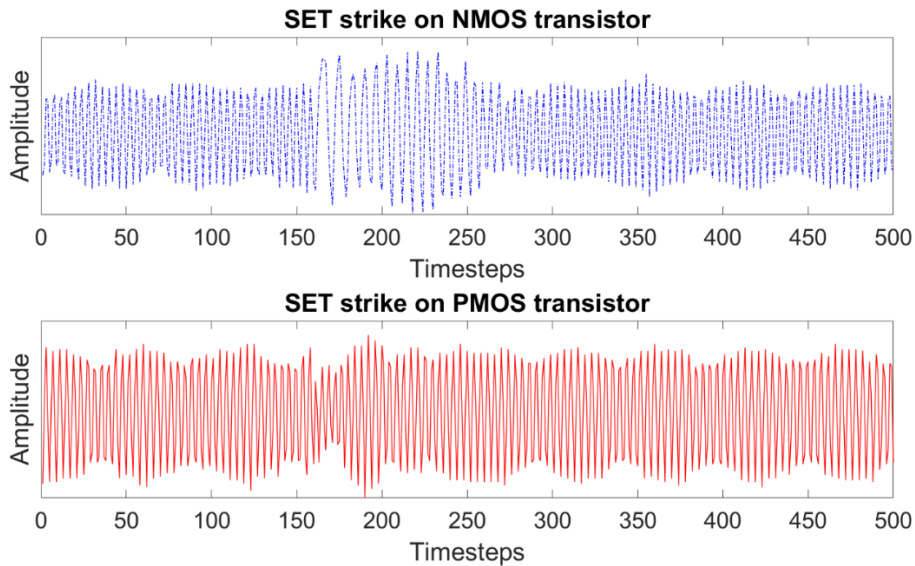


Figure 34

Time-domain waveforms used to generate IRES images in Figures 32 (top plot in this figure) and 33 (bottom plot in this figure)

Figure 34 shows time-domain waveforms of the SET strike at the output of NMOS transistor (top plot) and PMOS transistor (bottom plot). Comparing these plots with the IRES spectrograms in Figures 32 and 33 provides knowledge about the ability of the IRES to identify the features obtained using standard techniques (e.g., identify the change in frequency and amplitude) as well as highlight very important features (e.g., charge deposition and severity of the event) that the standard process misses.

Simulation Analysis with IRES

Simulation data for the SETs was obtained using a behavioral model of the PLL designed with Verilog-A. The Verilog-A PLL behavior model was designed to match the circuit dynamics of the experimental device. The PLL was designed to have a center frequency of 200 MHz to match with the experimental PLL designed in 130-nm CMOS technology. The Cadence Spectre simulator was used to perform the SET simulations. The behavioral PLL model shows a lock time of 200 ns.

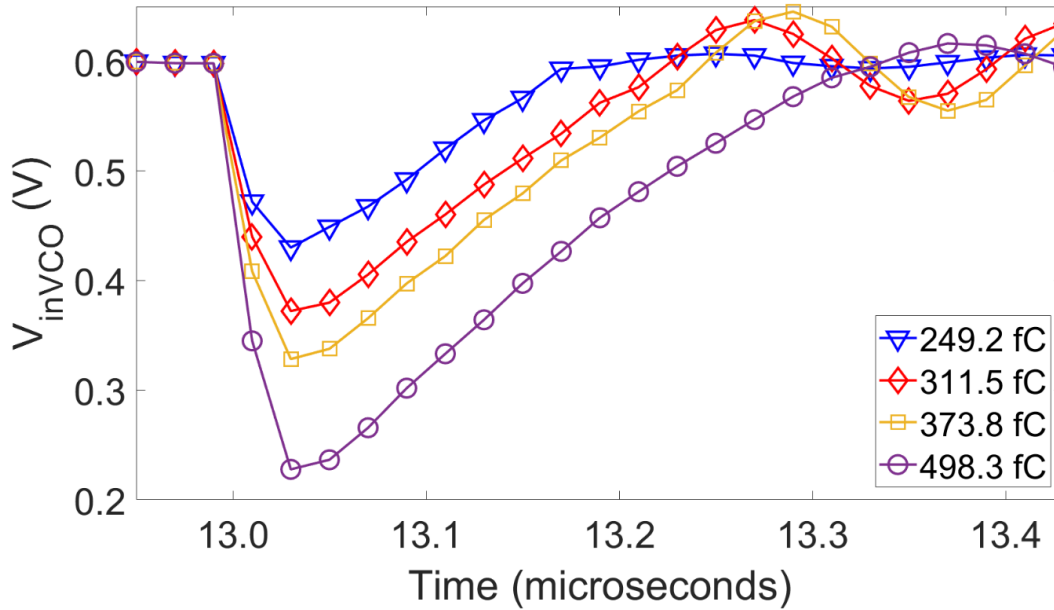


Figure 35

Simulated SE strikes at the output of the CP after various levels of charge deposited from a current source representing the NMOS transistor strike. Recovery times between 200 and 300 ns was observed. The drop in voltage was between 0.15 and 0.4 V [4]

Figure 35 shows the simulated SE strikes at the output of the CP using VCO control node versus time [4]. It shows that the voltage drop and loop recovery time increases as the deposited charge value increases. The NMOS simulated SE strikes are captured at 260 MHz. This node is internal to the PLL circuit and it may not be accessible experimentally. As the VCO is linear, the PLL's output frequency is proportional to V_{inVCO} . SET response of the PLL circuit is dominated by the loop parameters, is largely independent of ion strike time constant, and is proportional to charge deposited as shown in [26]. To simulate the ion strike, a double exponential model from a current source was used.

For the simulation analysis, the output waveforms were sampled at 10 GS/s to generate IRES images using the method described in the previous section of this chapter. IRES images for

SETs for various charge deposition levels were generated. Figures 36, 37, and 38 shows IRES images for SETs of 249.2 fC, 373.8 fC, and 498.3 fC of charge deposited at the output of the CP, respectively. Comparing these three IRES images, mean, standard deviation, and variance of the cycle-to-mean jitter are elongated in the x-axis direction for increased deposited charge. When the mean frequency of the instantaneous frequency is compared for all three IRES images, it shows that the loop recovery time also increases as the deposited charge increases. The color of the mean instantaneous frequency immediately following the SE strike is light orange in Figure 36 and green in Figures 37 and 38. The changes in color indicate that the reduction in frequency was higher for the SE strike with 373.8fC and 498.3 fC of deposited charge when compared to 249.2 fC of deposited charge.

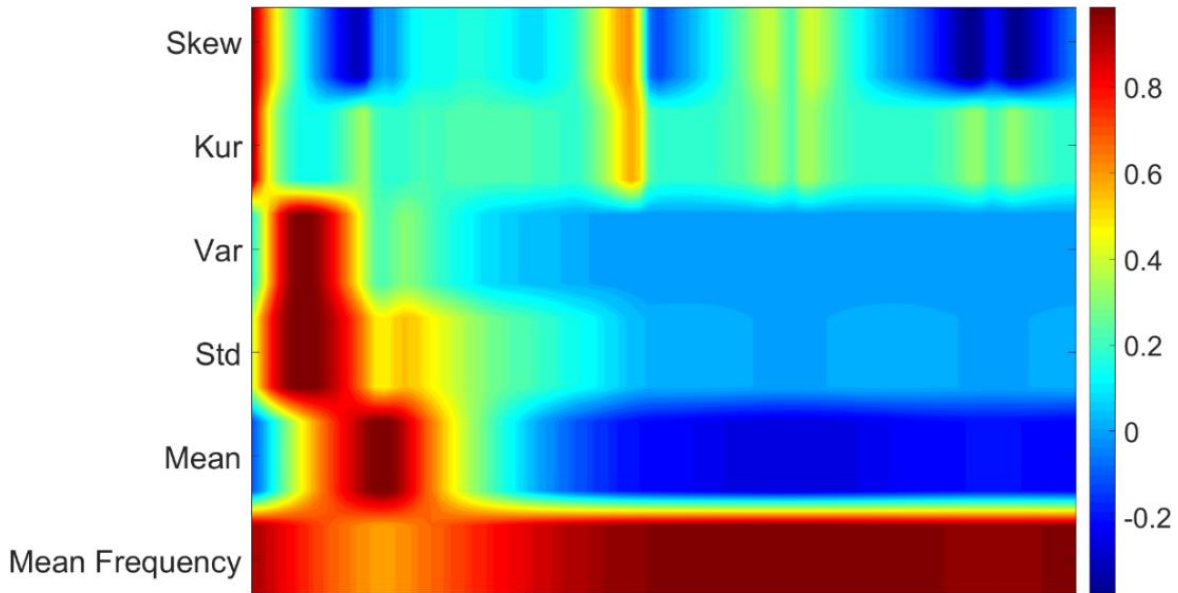


Figure 36

IRES image of SET simulated in PLL at 260 MHz for 249.2 fC charge deposited at the output of the CP

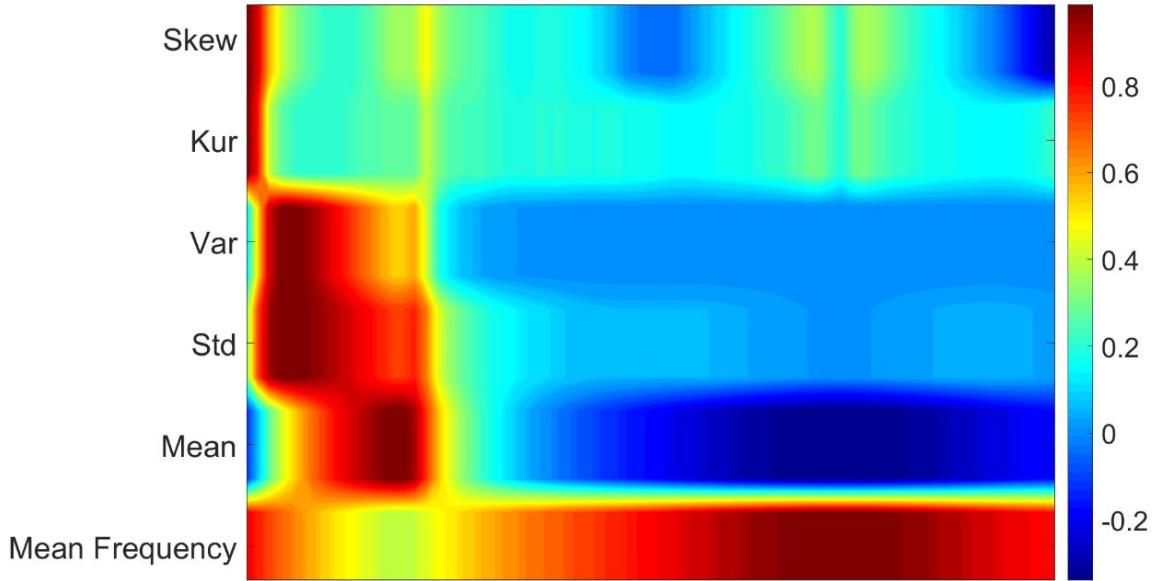


Figure 37

IRES image of SET simulated in PLL at 260 MHz for 373.8 fC charge deposited at the output of the CP

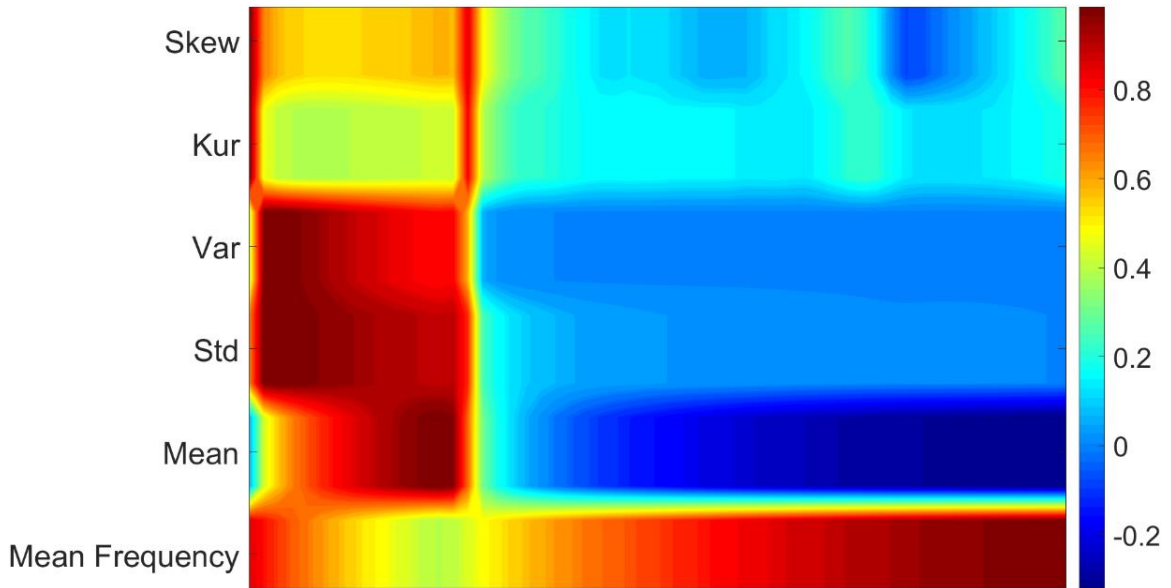


Figure 38

IRES image of SET simulated in PLL at 260 MHz for 498.3 fC charge deposited at the output of the CP

Simulation-generated IRES spectrograms show similar characteristics to the experimentally generated IRES images. Minor changes in the behavior of the circuit (e.g., charge collection) can be observed with IRES. IRES shows the characteristics of the SET that are indicators of the charge deposited. The ability of the IRES technique to identify the SETs in a noisy measurement is of particular interest. The statistical analysis performed on the cycle-to-mean phase jitter allows for suppression of noisy features. A detailed insight into radiation effect mechanism from a SET can be observed through IRES.

Chapter Conclusion

This chapter presents the utilization of the IRES technique for the non-invasive characterization of SETs in complex arbitrary circuits. The IRES technique uses multi-dimensional statistical feature analysis for identification of transient events in noisy measurements. In this work, IRES was used to image the cycle-to-mean jitter of the output waveform. Both experimental and simulation data show that IRES is able to characterize SE strikes within a PLL circuit. For experiment results, a PLL designed and fabricated in 130 nm CMOS technology was used and SETs were generated from TPA experiments. For simulation analysis, a behavioral model of PLL using Verilog-A was designed to match the performance of the PLL fabricated for experimental analysis. A window approach was used to characterize the SETs as opposed to the block approach used for characterization of TID. In addition to the standard measurement approach results, IRES allows for capturing of charge deposited during the SE strike.

CHAPTER VI

DISCUSSION AND CONCLUSION

Devices used in the microelectronic circuits are vulnerable within the space radiation environment; for example, integrated transistors tend to change their operating characteristics such as threshold voltage based on the amount and duration of radiation exposure. This work analyzes two different space radiation effects that are prominent within the space environment: 1) Total-Ionizing Dose and 2) Single Event Transients. TID is a cumulative radiation effect, while SET is a transient effect. Based on the severity of the effect, the operating device can be temporarily or permanently damaged.

An approach to identify TID radiation effects in the presence of noise in RF circuits with very high accuracy is achieved through the proposed IRES technique. The IRES technique is also utilized for identification of spurious transient events in PLL circuit. IRES is based on RF-DNA fingerprinting, a technique for identification of different devices based on statistical features extracted from their transmitted waveforms.

The IRES technique for TID uses only two statistical features (mean and standard deviation of the instantaneous frequency) to identify the operational health of the circuit. LD models were used to classify the TID radiation level as well input bias voltage of the VCO and PLL. Classification accuracy of 100 % was observed for VCO A, PLL A, PLL B, while classification accuracy of 97.5% was achieved for VCO B for training data set size of 400 samples and testing data set size of 100 samples.

The IRES technique for SET uses multi-dimensional statistical analysis for identification of transient events in the presence of noise. Statistical measures such as mean, standard deviation, variance, kurtosis, and skewness were calculated using the cycle-to-mean phase jitter extracted from the PLL's output waveform. The IRES technique for SETs allows for easy identification of strike location through the spectral information generated from the cycle-to-mean jitter of the output waveform of the PLL circuit. IRES images generated from laser strikes on different locations (i.e., NMOS transistor and PMOS transistor at the output node of the CP circuit) were visually identifiable through the spectral characteristics. A Verilog-A behavioral PLL model designed to match the circuit dynamics of the experimental device was used to generate simulation results. Both experimental and simulation results showed similarity in the spectral characteristics of the IRES images.

This work presents that the IRES technique can be applied for both cumulative as well as transient radiation effects. In addition to standard measurement technique approach, IRES gives ability to capture global parametric shifts in the arbitrary complex circuits, charge dynamics of the radiation effects on the devices, and identification of spurious transients in the presence of noise. The IRES technique can be utilized in the applications that are not specifically related with radiation effects. The versatility of the statistical analysis approach used in the IRES technique makes it very effective in finding behavioral changes of a system in the data driven world.

REFERENCES

- [1] G. E. Moore. (1965, April 19,1965) Cramming more components onto integrated circuits. *Electronics*.
- [2] R. R. Schaller, "Moore's law: past, present and future," *IEEE Spectrum*, vol. 34, no. 6, pp. 52-59, 1997, doi: 10.1109/6.591665.
- [3] B. Patel *et al.*, "Ionizing Radiation Effects Spectroscopy for Analysis of Total-Ionizing Dose Degradation in RF Circuits," *IEEE Transactions on Nuclear Science*, vol. 66, no. 1, pp. 61-68, 2019, doi: 10.1109/TNS.2018.2884642.
- [4] T. D. Loveless *et al.*, "Ionizing Radiation Effects Spectroscopy for Analysis of Single Event Transients," *IEEE Transactions on Nuclear Science*, pp. 1-1, 2019, doi: 10.1109/TNS.2019.2950385.
- [5] J. A. Van Allen, C. E. McIlwain, and G. H. Ludwig, "Radiation observations with satellite 1958 ϵ ," *Journal of Geophysical Research (1896-1977)*, vol. 64, no. 3, pp. 271-286, 1959, doi: 10.1029/JZ064i003p00271.
- [6] K. Endo, "Earth's Magnetic Field," *Nikkei Science Inc. of Japan*, 2006.
- [7] (1993) The Hipparcos Mission - Four Years After Launch. *ESA Bulletin*. 14.
- [8] E. G. Stassinopoulos and J. P. Raymond, "The space radiation environment for electronics," *Proceedings of the IEEE*, vol. 76, no. 11, pp. 1423-1442, 1988, doi: 10.1109/5.90113.
- [9] M. Xapsos, "Modeling the Space Radiation Environment," presented at the 2006 IEEE Nuclear and Space Radiation Effects Conference Short Course Notebook, Ponte Vedra Beach, Florida, 2006.
- [10] T. R. Oldham and F. B. McLean, "Total ionizing dose effects in MOS oxides and devices," *IEEE Transactions on Nuclear Science*, vol. 50, no. 3, pp. 483-499, 2003, doi: 10.1109/TNS.2003.812927.
- [11] L. Massengill, "SEU Modeling and Prediction Techniques," presented at the IEEE Nuclear and Space Radiation Effects Conference Short Course, Snowbird, Utah, 1993.

- [12] P. E. Dodd and L. W. Massengill, "Basic mechanisms and modeling of single-event upset in digital microelectronics," *IEEE Transactions on Nuclear Science*, vol. 50, no. 3, pp. 583-602, 2003, doi: 10.1109/TNS.2003.813129.
- [13] W. C. S. II, M. A. Temple, M. J. Mendenhall, and R. F. Mills, "Radio frequency fingerprinting commercial communication devices to enhance electronic security," *Int. J. Electron. Secur. Digit. Forensic*, vol. 1, no. 3, pp. 301-322, 2008, doi: 10.1504/ijesdf.2008.020946.
- [14] R. W. Klein, M. A. Temple, and M. J. Mendenhall, "Application of wavelet-based RF fingerprinting to enhance wireless network security," *Journal of Communications and Networks*, vol. 11, no. 6, pp. 544-555, 2009, doi: 10.1109/JCN.2009.6388408.
- [15] M. D. Williams, S. A. Munns, M. A. Temple, and M. J. Mendenhall, "RF-DNA Fingerprinting for Airport WiMax Communications Security," in *2010 Fourth International Conference on Network and System Security*, 1-3 Sept. 2010 2010, pp. 32-39, doi: 10.1109/NSS.2010.21. [Online]. Available: <https://ieeexplore.ieee.org/document/5636117/>
- [16] P. K. Harmer, D. R. Reising, and M. A. Temple, "Classifier selection for physical layer security augmentation in Cognitive Radio networks," in *2013 IEEE International Conference on Communications (ICC)*, 9-13 June 2013 2013, pp. 2846-2851, doi: 10.1109/ICC.2013.6654972. [Online]. Available: <https://ieeexplore.ieee.org/document/6654972/>
- [17] H. J. Patel, M. A. Temple, and R. O. Baldwin, "Improving ZigBee Device Network Authentication Using Ensemble Decision Tree Classifiers With Radio Frequency Distinct Native Attribute Fingerprinting," *IEEE Transactions on Reliability*, vol. 64, no. 1, pp. 221-233, 2015, doi: 10.1109/TR.2014.2372432.
- [18] D. R. Reising, M. A. Temple, and J. A. Jackson, "Authorized and Rogue Device Discrimination Using Dimensionally Reduced RF-DNA Fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 6, pp. 1180-1192, 2015, doi: 10.1109/TIFS.2015.2400426.
- [19] T. J. Bihl, K. W. Bauer, and M. A. Temple, "Feature Selection for RF Fingerprinting With Multiple Discriminant Analysis and Using ZigBee Device Emissions," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1862-1874, 2016, doi: 10.1109/TIFS.2016.2561902.
- [20] D. R. Reising, M. A. Temple, and M. J. Mendenhall, "Improving Intra-Cellular Security Using Air Monitoring with RF Fingerprints," in *2010 IEEE Wireless Communication and Networking Conference*, 18-21 April 2010 2010, pp. 1-6, doi: 10.1109/WCNC.2010.5506229. [Online]. Available: <https://ieeexplore.ieee.org/document/5506229/>

- [21] D. R. Reising, M. A. Temple, and M. J. Mendenhall, "Improved wireless security for GMSK-based devices using RF fingerprinting," *Int. J. Electron. Secur. Digit. Forensic*, vol. 3, no. 1, pp. 41-59, 2010, doi: 10.1504/ijesdf.2010.032330.
- [22] N. Agrawal, H. Liu, R. Arghavani, V. Narayanan, and S. Datta, "Impact of Variation in Nanoscale Silicon and Non-Silicon FinFETs and Tunnel FETs on Device and SRAM Performance," *IEEE Transactions on Electron Devices*, vol. 62, no. 6, pp. 1691-1697, 2015, doi: 10.1109/TED.2015.2406333.
- [23] T. Grasser, H. Reisinger, P. Wagner, F. Schanovsky, W. Goes, and B. Kaczer, "The time dependent defect spectroscopy (TDDS) for the characterization of the bias temperature instability," in *2010 IEEE International Reliability Physics Symposium*, 2-6 May 2010 2010, pp. 16-25, doi: 10.1109/IRPS.2010.5488859. [Online]. Available: <https://ieeexplore.ieee.org/document/5488859/>
- [24] J. Allen, "Short term spectral analysis, synthesis, and modification by discrete Fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 3, pp. 235-238, 1977, doi: 10.1109/TASSP.1977.1162950.
- [25] M. H. Roy, "The Power Spectral Density," in *Principles of Random Signal Analysis and Low Noise Design: The Power Spectral Density and its Applications*: IEEE, 2002, pp. 59-91.
- [26] T. D. Loveless, L. W. Massengill, W. T. Holman, B. L. Bhuvu, D. McMorrow, and J. H. Warner, "A Generalized Linear Model for Single Event Transient Propagation in Phase-Locked Loops," *IEEE Transactions on Nuclear Science*, vol. 57, no. 5, pp. 2933-2947, 2010, doi: 10.1109/TNS.2010.2066287.
- [27] S. Jagannathan *et al.*, "Sensitivity of High-Frequency RF Circuits to Total Ionizing Dose Degradation," *IEEE Transactions on Nuclear Science*, vol. 60, no. 6, pp. 4498-4504, 2013, doi: 10.1109/TNS.2013.2283457.
- [28] T. D. Haeffner *et al.*, "Irradiation and Temperature Effects for a 32 nm RF Silicon-on-Insulator CMOS Process," *IEEE Transactions on Nuclear Science*, vol. 61, no. 6, pp. 3037-3042, 2014, doi: 10.1109/TNS.2014.2360455.
- [29] Z. Chen *et al.*, "Study of Total-Ionizing-Dose Effects on a Single-Event-Hardened Phase-Locked Loop," *IEEE Transactions on Nuclear Science*, vol. 65, no. 4, pp. 997-1004, 2018, doi: 10.1109/TNS.2018.2812806.
- [30] F. Gardner, "Charge-Pump Phase-Lock Loops," *IEEE Transactions on Communications*, vol. 28, no. 11, pp. 1849-1858, 1980, doi: 10.1109/TCOM.1980.1094619.
- [31] T. D. Loveless *et al.*, "Combined Effects of Total Ionizing Dose and Temperature on a K-Band Quadrature LC-Tank VCO in a 32 nm CMOS SOI Technology," *IEEE Transactions on Nuclear Science*, vol. 64, no. 1, pp. 204-211, 2017, doi: 10.1109/TNS.2016.2637699.

- [32] T. D. Loveless *et al.*, "A Single-Event-Hardened Phase-Locked Loop Fabricated in 130 nm CMOS," *IEEE Transactions on Nuclear Science*, vol. 54, no. 6, pp. 2012-2020, 2007, doi: 10.1109/TNS.2007.908166.
- [33] R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, vol. 7, pp. 179-188, 1936. [Online]. Available: <http://hdl.handle.net/2440/15227>.
- [34] R. T. Trevor Hastie, Jerome Friedman, *The Elements of Statistical Learning: Data mining, Inference, and Prediction*. New York, New York, USA: Springer, 2001.
- [35] Y. Boulghassoul *et al.*, "Circuit modeling of the LM124 operational amplifier for analog single-event transient analysis," *IEEE Transactions on Nuclear Science*, vol. 49, no. 6, pp. 3090-3096, 2002, doi: 10.1109/TNS.2002.805400.
- [36] S. Buchner and D. McMorrow, "Single-Event Transients in Bipolar Linear Integrated Circuits," *IEEE Transactions on Nuclear Science*, vol. 53, no. 6, pp. 3079-3102, 2006, doi: 10.1109/TNS.2006.882497.
- [37] E. W. V. Stryland *et al.*, "Two Photon Absorption, Nonlinear Refraction, And Optical Limiting In Semiconductors," *Optical Engineering*, vol. 24, no. 4, pp. 613-623, 11, 1985. [Online]. Available: <https://doi.org/10.1117/12.7973538>.
- [38] T. Boggess, K. Bohnert, K. Mansour, S. Moss, I. Boyd, and A. Smirl, "Simultaneous measurement of the two-photon coefficient and free-carrier cross section above the bandgap of crystalline silicon," *IEEE Journal of Quantum Electronics*, vol. 22, no. 2, pp. 360-368, 1986, doi: 10.1109/JQE.1986.1072964.
- [39] D. McMorrow, W. T. Lotshaw, J. S. Melinger, S. Buchner, and R. L. Pease, "Subbandgap laser-induced single event effects: carrier generation via two-photon absorption," *IEEE Transactions on Nuclear Science*, vol. 49, no. 6, pp. 3002-3008, 2002, doi: 10.1109/TNS.2002.805337.
- [40] D. McMorrow *et al.*, "Three-dimensional mapping of single-event effects using two photon absorption," *IEEE Transactions on Nuclear Science*, vol. 50, no. 6, pp. 2199-2207, 2003, doi: 10.1109/TNS.2003.820742.

APPENDIX A

MATLAB SCRIPTS

MATLAB SCRIPTS

The MATLAB scripts below were used to generate the IRES images for TID and SET. Scripts A and B are the knob setting and IRES generation for TID, respectively. Scripts C and D are the knob setting and IRES generation code for SET, respectively. Idea of this code was imported from the code for RF-DNA fingerprinting received from Dr. Donald Reising. Knob setting files are used to setup the simulation based on input waveform.

1. Knob settings for generation of the IRES from TID

```
clear all;
close all;
%% Load data after this line
%% Captured Waveforms
%% Enter samplerate of the data collection
samplerate = 5E9;
%% Required Inputs from user
FeatureUsed = 'f'; % type f, c2c, c2m, rt freq_c2c
RecordLength = 1000;
NumFrames = 400;
CalVar = 0;
CalSkew = 0;
CalStd = 1;
CalMean = 1;
CalKur = 0;
NumStats = CalVar+CalSkew+CalStd+CalMean+CalKur;
NumFeature = 1;
Normalize = 1;      % Enter 0 for normalizing the data for the signal itself
                   % Or Enter Normalize = 1 for normalizing the data
                   % across all signals at the input

distribution = 1;
EntireSigStat = 0; % enter number of divisions needed in "NumSubdivision"
if EntireSigStat == 0
    NumSubdivisions = 40; % Number of subdivision per signal
elseif EntireSigStat == 1
    NumSubdivisions = 1; % Do not Change this number
end
%% Enter the Signal name for IRES prints
% For frequency curve
% CDPLL in
NameSignals =
{in_05_0,in_05_300,in_06_0,in_06_300,in_07_0,in_07_300,in_08_0,in_08_300};
%% Enter the name of the function
Statistics_Generation_V3(FeatureUsed, CalVar, CalSkew, CalMean, CalKur,...
    CalStd, EntireSigStat,NumSubdivisions,NameSignals, samplerate, Normalize,
    NumStats, NumFrames, RecordLength, distribution)
```

```
clearvars distribution CalKur RecordLength NumFrames CalMean CalSkew CalStd CalVar
NameSignals NumFeature NumSubdivisions samplerate EntireSigStat FeatureUsed Normalize
NumStats;
```

2. Code for generation of the IRES for TID

```
%% Statistics Calculation and IRES Generation
%% Function
function Statistics_Generation_V3(FeatureUsed, CalVar, CalSkew, CalMean,
CalKur,...
    CalStd, EntireSigStat,NumSubdivisions,NameSignals,samplerate, Normalize,
NumStats, NumFrames, RecordLength, distribution)
% Calculate the period of the signal and save data into cells
[~, NumSignals] = size(NameSignals); %Get the number of the signals to be
processed
if strcmp(FeatureUsed, 'rt')
    for p=1:NumSignals
        data = cell2mat(NameSignals(:,p));
        FirstDataPoint = 0;
        LastDataPoint = FirstDataPoint+RecordLength;
        Inst_rise_added =
risetime(data(FirstDataPoint+1:LastDataPoint),samplerate,'StateLevels',[0.0
0.2]);
        FirstDataPoint = LastDataPoint;
        LastDataPoint = RecordLength+LastDataPoint;
        for j = 2: NumFrames
            data_new = data(FirstDataPoint+1:LastDataPoint);
            Inst_rise = risetime(data_new,samplerate,'StateLevels',[0.0
0.2]);
            Inst_rise_added = [Inst_rise_added; Inst_rise];
            FirstDataPoint = LastDataPoint;
            LastDataPoint = RecordLength+LastDataPoint;
        end
        Inst_rise_Total{p} = Inst_rise_added;
        clear data;
    end
else
    for p=1:NumSignals
        data = cell2mat(NameSignals(:,p));
        FirstDataPoint = 0;
        LastDataPoint = FirstDataPoint+RecordLength;
        Inst_Period_added =
pulseperiod(data(FirstDataPoint+1:LastDataPoint),samplerate,'StateLevels',[0.
0 0.2]);
        number_of_pulses_in_each_frame(:,1) = length(Inst_Period_added);
        FirstDataPoint = LastDataPoint;
        LastDataPoint = RecordLength+LastDataPoint;
        for j = 2: NumFrames
            data_new = data(FirstDataPoint+1:LastDataPoint);
            Inst_Period = pulseperiod(data_new,samplerate,'StateLevels',[0.0
0.2]);
            number_of_pulses_in_each_frame(:,j) = length(Inst_Period);
            Inst_Period_added = [Inst_Period_added; Inst_Period];
            FirstDataPoint = LastDataPoint;
            LastDataPoint = RecordLength+LastDataPoint;
```

```

end
freq = 1 ./ Inst_Period_added;
freq = mean(freq);
freq_mean(p,:) = freq;
Inst_Period_Total{p} = Inst_Period_added;
PulseNumbers_each_frame{p} = number_of_pulses_in_each_frame;
clear data FirstDataPoint LastDataPoint Inst_Period data_new freq j;
end
end
% Calculate features such as cycle to cycle jitter, cycle to mean jitter
% frequency from the instantaneous period of the pulse
for q=1:NumSignals

if strcmp(FeatureUsed, 'rt')
    data = cell2mat(Inst_rise_Total(:,q));
    rt{q} = data;
    transformed_signal{q} = rt{q};
    clearvars n rt data ;
end
% Calculate cycle to cycle jitter
if strcmp(FeatureUsed, 'c2c')
    data = cell2mat(Inst_Period_Total(:,q));
    for n = 1:length(data)-1
        cc(n,:) = data(n+1,1)-data(n,1);
    end
    c2c{q} = cc;
    transformed_signal{q} = c2c{q};

    clearvars n cc data;
end
% Calculate cycle to mean jitter
if strcmp(FeatureUsed, 'c2m')
    data = cell2mat(Inst_Period_Total(:,q));
    data_mean = mean(data);
    for n = 1:length(data)-1
        cm(n,:) = data(n,1)-data_mean;
    end
    c2m{q} = cm;
    transformed_signal{q} = c2m{q};
    clearvars n data cm c2m data_mean;
end
% Calculate instantaneous frequency
if strcmp(FeatureUsed, 'f')
    data = cell2mat(Inst_Period_Total(:,q));
    Freq = 1 ./ data;
    Inst_Freq{q} = Freq;
    transformed_signal{q} = Inst_Freq{q};
    clear data Freq Inst_Freq;
end
end
clearvars p q;
%% Divide the signal into sub regions and calculate the statistics
if EntireSigStat == 0
    for p = 1:NumSignals
        data = cell2mat(transformed_signal(:,p));
        cntr = 1;

```

```

    initChunk = 0;
    ChunkSize = floor(length(data)/NumSubdivisions);
    endChunk = ChunkSize;
    for k = 1:NumSubdivisions
        RegionVar(k,p) = var(data(initChunk+1:endChunk));
        RegionSkew(k,p) = skewness(data(initChunk+1:endChunk));
        RegionKur(k,p) = kurtosis(data(initChunk+1:endChunk));
        RegionStd(k,p) = std(data(initChunk+1:endChunk));
        RegionMean(k,p) = mean(data(initChunk+1:endChunk));
        initChunk = endChunk;
        endChunk = ChunkSize + endChunk;
        cntr = cntr+1;
    end
end
end
clearvars p data cntr initChunk ChunkSize endChunk k;
%% Calculate statistic over entire signal
if EntireSigStat == 1
    for p = 1:NumSignals
        data = cell2mat(transformed_signal(:,p));
        RegionVar(:,p) = var(data);
        RegionSkew(:,p) = skewness(data);
        RegionKur(:,p) = kurtosis(data);
        RegionMean(:,p) = mean(data);
        RegionStd(:,p) = std(data);
    end
end
clearvars p data transformed_signal;
%% Normalize the data with respect to highest value in the data
if Normalize == 0
    RegionVar = RegionVar ./ (max(RegionVar));
    RegionSkew = RegionSkew ./ (max(RegionSkew));
    RegionKur = RegionKur ./ (max(RegionKur));
    RegionStd = RegionStd ./ (max(RegionStd));
    RegionMean = RegionMean ./ (max(RegionMean));
elseif Normalize == 1
    RegionVar = RegionVar ./ max(max(RegionVar));
    RegionSkew = abs(RegionSkew ./ max(max(abs(RegionSkew))));
    RegionKur = RegionKur ./ max(max(RegionKur));
    RegionStd = RegionStd ./ max(max(RegionStd));
    RegionMean = RegionMean ./ max(max(RegionMean));
end
clearvars Normalize;
%% Assign statistics to the IPrint variable
% Skewness, Kurtosis, Mean, Standard deviation, Variance
%(CalVar == 0) && (CalStd == 0) && (CalMean == 0) && (CalKur == 0) &&
%(CalSkew == 0);
for s = 1:NumSignals
    if (CalVar == 0) && (CalStd == 0) && (CalMean == 0) && (CalKur == 0) &&
(CalSkew == 1)
        IPrint(:, :, s) = RegionSkew(:, s);
        StatLabel = {'Skew'};
    elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 0) && (CalKur == 1)
&& (CalSkew == 0)
        IPrint(:, :, s) = RegionKur(:, s);
        StatLabel = {'Kur'};

```

```

elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 0) && (CalKur == 1)
&& (CalSkew == 1)
    IPrnt = [RegionKur(:,s);zeros(1,1);RegionSkew(:,s)];
    StatLabel = {'Kur','Skew'};
elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 1) && (CalKur == 0)
&& (CalSkew == 0)
    IPrnt(:, :, s) = RegionMean(:,s);
    StatLabel = {'Mean'};
elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 1) && (CalKur == 0)
&& (CalSkew == 1)
    IPrnt(:, :, s) = [RegionMean(:,s);zeros(1,1);RegionSkew(:,s)];
    StatLabel = {'Mean','Skew'};
elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 1) && (CalKur == 1)
&& (CalSkew == 0)
    IPrnt(:, :, s) = [RegionMean(:,s);zeros(1,1);RegionKur(:,s)];
    StatLabel = {'Mean','Kur'};
elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 1) && (CalKur == 1)
&& (CalSkew == 1)
    IPrnt(:, :, s) =
[RegionMean(:,s);zeros(1,1);RegionKur(:,s);zeros(1,1);RegionSkew(:,s)];
    StatLabel = {'Mean','Kur','Skew'};
elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 0) && (CalKur == 0)
&& (CalSkew == 0)
    IPrnt(:, :, s) = RegionStd(:,s);
    StatLabel = {'Std'};
elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 0) && (CalKur == 0)
&& (CalSkew == 1)
    IPrnt(:, :, s) = [RegionStd(:,s);zeros(1,1);RegionSkew(:,s)];
    StatLabel = {'Std','Skew'};
elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 0) && (CalKur == 1)
&& (CalSkew == 0)
    IPrnt = [RegionStd(:,s);zeros(1,1);RegionKur(:,s)];
    StatLabel = {'Std','Kur'};
elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 0) && (CalKur == 1)
&& (CalSkew == 1)
    IPrnt =
[RegionStd(:,s);zeros(1,1);RegionKur(:,s);zeros(1,1);RegionSkew(:,s)];
    StatLabel = {'Std','Kur','Skew'};
elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 1) && (CalKur == 0)
&& (CalSkew == 0)
    IPrnt(:, :, s) = [RegionStd(:,s);zeros(1,1);RegionMean(:,s)];
    StatLabel = {'Std','Mean'};
elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 1) && (CalKur == 0)
&& (CalSkew == 1)
    IPrnt(:, :, s) = [RegionStd(:,s);zeros(1,1);
RegionMean(:,s);zeros(1,1);RegionSkew(:,s)];
    StatLabel = {'Std','Mean','Skew'};
elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 1) && (CalKur == 1)
&& (CalSkew == 0)
    IPrnt(:, :, s) = [RegionStd(:,s);zeros(1,1);
RegionMean(:,s);zeros(1,1);RegionKur(:,s)];
    StatLabel = {'Std','Mean','Kur'};
elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 1) && (CalKur == 1)
&& (CalSkew == 1)
    IPrnt(:, :, s) = [RegionStd(:,s);zeros(1,1);
RegionMean(:,s);zeros(1,1);RegionKur(:,s);zeros(1,1);RegionSkew(:,s)];

```

```

    StatLabel = {'Std', 'Mean', 'Kur', 'Skew'};
elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 0) && (CalKur == 0)
&& (CalSkew == 0)
    IPrnt(:, :, s) = RegionVar(:, s);
    StatLabel = {'Var'};
elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 0) && (CalKur == 0)
&& (CalSkew == 1)
    IPrnt(:, :, s) = [RegionVar(:, s); zeros(1, 1); RegionSkew(:, s)];
    StatLabel = {'Var', 'Skew'};
elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 0) && (CalKur == 1)
&& (CalSkew == 0)
    IPrnt(:, :, s) = [RegionVar(:, s); zeros(1, 1); RegionKur(:, s)];
    StatLabel = {'Var', 'Kur'};
elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 0) && (CalKur == 1)
&& (CalSkew == 1)
    IPrnt(:, :, s) =
[RegionVar(:, s); zeros(1, 1); RegionKur(:, s); zeros(1, 1); RegionSkew(:, s)];
    StatLabel = {'Var', 'Kur', 'Skew'};
elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 1) && (CalKur == 0)
&& (CalSkew == 0)
    IPrnt(:, :, s) = [RegionVar(:, s); zeros(1, 1); RegionMean(:, s)];
    StatLabel = {'Var', 'Mean'};
elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 1) && (CalKur == 0)
&& (CalSkew == 1)
    IPrnt(:, :, s) =
[RegionVar(:, s); zeros(1, 1); RegionMean(:, s); zeros(1, 1); RegionSkew(:, s)];
    StatLabel = {'Var', 'Mean', 'Skew'};
elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 1) && (CalKur == 1)
&& (CalSkew == 0)
    IPrnt(:, :, s) = [RegionVar(:, s);
RegionMean(:, s); zeros(1, 1); RegionKur(:, s)];
    StatLabel = {'Var', 'Mean', 'Kur'};
elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 1) && (CalKur == 1)
&& (CalSkew == 1)
    IPrnt(:, :, s) =
[RegionVar(:, s); zeros(1, 1); RegionMean(:, s); zeros(1, 1); RegionKur(:, s); zeros(1,
1); RegionSkew(:, s)];
    StatLabel = {'Var', 'Mean', 'Kur', 'Skew'};
elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 0) && (CalKur == 0)
&& (CalSkew == 0)
    IPrnt(:, :, s) = [RegionVar(:, s); zeros(1, 1); RegionStd(:, s)];
    StatLabel = {'Var', 'Std'};
elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 0) && (CalKur == 0)
&& (CalSkew == 1)
    IPrnt(:, :, s) =
[RegionVar(:, s); zeros(1, 1); RegionStd(:, s); zeros(1, 1); RegionSkew(:, s)];
    StatLabel = {'Var', 'Std', 'Skew'};
elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 0) && (CalKur == 1)
&& (CalSkew == 0)
    IPrnt(:, :, s) =
[RegionVar(:, s); zeros(1, 1); RegionStd(:, s); zeros(1, 1); RegionKur(:, s)];
    StatLabel = {'Var', 'Std', 'Kur'};
elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 0) && (CalKur == 1)
&& (CalSkew == 1)

```

```

        IPrint(:, :, s) =
[RegionVar(:, s); zeros(1, 1); RegionStd(:, s); zeros(1, 1); RegionKur(:, s); zeros(1, 1)
]; RegionSkew(:, s)];
        StatLabel = {'Var', 'Std', 'Kur', 'Skew'};
        elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 1) && (CalKur == 0)
&& (CalSkew == 0)
            IPrint(:, :, s) = [RegionVar(:, s); zeros(1, 1); RegionStd(:, s); zeros(1, 1);
RegionMean(:, s)];
            StatLabel = {'Var', 'Std', 'Mean'};
            elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 1) && (CalKur == 0)
&& (CalSkew == 1)
                IPrint(:, :, s) = [RegionVar(:, s); zeros(1, 1); RegionStd(:, s); zeros(1, 1);
RegionMean(:, s); zeros(1, 1); RegionSkew(:, s)];
                StatLabel = {'Var', 'Std', 'Mean', 'Skew'};
                elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 1) && (CalKur == 1)
&& (CalSkew == 0)
                    IPrint(:, :, s) = [RegionVar(:, s); zeros(1, 1); RegionStd(:, s); zeros(1, 1);
RegionMean(:, s); zeros(1, 1); RegionKur(:, s)];
                    StatLabel = {'Var', 'Std', 'Mean', 'Kur'};
                    elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 1) && (CalKur == 1)
&& (CalSkew == 1)
                        IPrint(:, :, s) = [RegionVar(:, s); zeros(1, 1); RegionStd(:, s); zeros(1, 1);
RegionMean(:, s); zeros(1, 1); RegionKur(:, s); zeros(1, 1); RegionSkew(:, s)];
                        StatLabel = {'Var', 'Std', 'Mean', 'Kur', 'Skew'};
                    end
                end
            end
clearvars s;
%% IRES PRINT DEVELOPMENT
IPrint1 = repmat(IPrint, 1, NumSubdivisions); % repeat the vertical column
"NumSubdivisions" time
[rows, ~, ~] = size(IPrint1); % Get number of rows
for s1 = 1:NumSignals
    zerocolumn = zeros(rows, 1); % Creates 1 vertical column of zeros
    IPrint2(:, :, s1) = [zerocolumn IPrint1(:, :, s1) zerocolumn]; %
    [~, columns, ~] = size(IPrint2);
    zerorow = zeros(1, columns);
    IPrint3(:, :, s1) = [zerorow; IPrint2(:, :, s1); zerorow];
end
clear rows columns zerorow zerocolumn s1 IPrint1 IPrint2;
%% Concatenate all IRES prints horizontally and save into one variable
[~, ~, pages] = size(IPrint3);
if pages == 1
    IRES = IPrint3(:, :, 1);
    SigLabel = {'S1'};
elseif pages == 2
    IRES = [IPrint3(:, :, 1) IPrint3(:, :, 2)];
    SigLabel = {'S1', 'S2'};
elseif pages == 3
    IRES = [IPrint3(:, :, 1) IPrint3(:, :, 2) IPrint3(:, :, 3)];
    SigLabel = {'S1', 'S2', 'S3'};
elseif pages == 4
    IRES = [IPrint3(:, :, 1) IPrint3(:, :, 2) IPrint3(:, :, 3) IPrint3(:, :, 4)];
    SigLabel = {'S1', 'S2', 'S3', 'S4'};
elseif pages == 5
    IRES = [IPrint3(:, :, 1) IPrint3(:, :, 2) IPrint3(:, :, 3) IPrint3(:, :, 4)
IPrint3(:, :, 5)];

```

```

    SigLabel = {'S1','S2','S3','S4','S5'};
elseif pages == 6
    IRES = [IPrint3(:, :, 1) IPrint3(:, :, 2) IPrint3(:, :, 3) IPrint3(:, :, 4)
IPrint3(:, :, 5) IPrint3(:, :, 6)];
    SigLabel = {'S1','S2','S3','S4','S5','S6'};
elseif pages == 7
    IRES = [IPrint3(:, :, 1) IPrint3(:, :, 2) IPrint3(:, :, 3) IPrint3(:, :, 4)
IPrint3(:, :, 5) IPrint3(:, :, 6) IPrint3(:, :, 7)];
    %SigLabel = {'S1','S2','S3','S4','S5','S6','S7'};
    SigLabel = {'0','10','20','40','100','200','300'};
elseif pages == 8
    IRES = [IPrint3(:, :, 1) IPrint3(:, :, 2) IPrint3(:, :, 3) IPrint3(:, :, 4)
IPrint3(:, :, 5) IPrint3(:, :, 6) IPrint3(:, :, 7) IPrint3(:, :, 8)];
    SigLabel = {'S1','S2','S3','S4','S5','S6','S7','S8'};
elseif pages == 9
    IRES = [IPrint3(:, :, 1) IPrint3(:, :, 2) IPrint3(:, :, 3) IPrint3(:, :, 4)
IPrint3(:, :, 5) IPrint3(:, :, 6) IPrint3(:, :, 7) IPrint3(:, :, 8) IPrint3(:, :, 9)];
    SigLabel = {'S1','S2','S3','S4','S5','S6','S7','S8','S9'};
elseif pages == 10
    IRES = [IPrint3(:, :, 1) IPrint3(:, :, 2) IPrint3(:, :, 3) IPrint3(:, :, 4)
IPrint3(:, :, 5) IPrint3(:, :, 6) IPrint3(:, :, 7) IPrint3(:, :, 8) IPrint3(:, :, 9)
IPrint3(:, :, 10)];
    SigLabel = {'S1','S2','S3','S4','S5','S6','S7','S8','S9','S10'};
elseif pages == 11
    IRES = [IPrint3(:, :, 1) IPrint3(:, :, 2) IPrint3(:, :, 3) IPrint3(:, :, 4)
IPrint3(:, :, 5) IPrint3(:, :, 6) IPrint3(:, :, 7) IPrint3(:, :, 8) IPrint3(:, :, 9)
IPrint3(:, :, 10) IPrint3(:, :, 11)];
    SigLabel = {'S1','S2','S3','S4','S5','S6','S7','S8','S9','S10','S11'};
elseif pages == 12
    Attention = ' This code only prints 11 signals at a time. Please consider
changing the code!!!';
    show(Attention);
elseif pages > 12
    error('Too many signals!!! :( Cannot process!');
end
clearvars pages;
figure1 = figure('Name','IRES Print');
colormap(gray);
axes1 = axes('Parent',figure1);
hold(axes1,'on');
contourf(IRES,100,'LineStyle','none');
% ylabel(['Statistics','(' num2str(NumSubdivisions) ' Samples per
Statistic)']);
% xlabel('Signals');
% title(['IRES Prints of ' num2str(NumSignals) ' signals']);
box(axes1,'on');
axis(axes1,'tight');
% Create tick locations for labels on x-axis based on number of the input
% signals
[rows, columns] = size(IRES);
spacing1 = columns/NumSignals;
first_tick1 = spacing1/2;
last_tick1 = first_tick1;
for t = 1:NumSignals-1
    next_tick1 = last_tick1+(spacing1*t);
    first_tick1 = [first_tick1 next_tick1];

```



```

end
% Create tick locations for labels on y-axis based on number of the plotted
statistics
spacing2 = floor(rows/NumStats);
first_tick2 = spacing2/2;
last_tick2 = first_tick2;
for t = 1:NumStats-1
    next_tick2 = last_tick2+(spacing2*t);
    first_tick2 = [first_tick2 next_tick2];
end
clearvars spacing1 spacing2 next_tick1 next_tick2 last_tick1 last_tick2;
% Set font size and add Labels
set(axes1, 'BoxStyle', 'full', 'FontSize', 26, 'Layer', 'top', 'XTick', ...
    first_tick1, 'XTickLabel', ...
    SigLabel, 'YTick', first_tick2, ...
    'YTickLabel', StatLabel);
colorbar('peer', axes1); %Adds the colorbar on the side of the plot

```

3. Knob settings for generation of the IRESS from SET

```

clc;
clear all;
close all;
%% Load data here
load scan_4_data.mat;
load single_event_transient_data.mat;
% pll_out = scan_4_data{1,3043};
% pll_out = pll_out(1:500);
pll_out = pll_out_40(129500:134000);
% pll_out = awgn(pll_out, 20);
% if simulation data then remove state level from the V5 function
%Check sample rate
% add +1 -1 to state level calculations for experimental data
% remove +0.05 -0.05 for simulation data
NameSignals = {pll_out};
[~, NumSignals] = size(NameSignals);
samplerate = {10E9};
% samplerate = {625E6};
RecordLength = {length(NameSignals{1,1})};
NumFrames = {1};
for sig = 1:NumSignals
    [~, mid_level(sig)] = midcross(NameSignals{sig});
    Low_state(sig) = mid_level(sig)-0.1;
    High_state(sig) = mid_level(sig)+0.1;
    StateLevels(sig) = {[Low_state(sig) High_state(sig)]};
end
tf =
isequal(length(NameSignals), length(samplerate), length(RecordLength), length(NumFrames), length(StateLevels));
if tf == 0
    error('Check all input parameters');
end
FeatureUsed = 'c2m';
CalVar = 1;

```

```

CalSkew =1;
CalStd = 1;
CalMean = 1;
CalKur = 1;
NumStats = CalVar+CalSkew+CalStd+CalMean+CalKur;
WindowLength = 15;
Shift = 1;
Normalize = 0; % 0 for Normalized to itself, 1 for Normalized to all
clear sig tf High_state Low_state mid_level;
%% V5 will only take one signal at a time
IRES_Generation_for_SET_V5(FeatureUsed, CalVar, CalSkew, CalMean, CalKur,...
    CalStd, WindowLength, NameSignals, samplerate, NumStats, NumFrames,...
    RecordLength, StateLevels, Shift, NumSignals, Normalize)

```

4. Code for generation of the IRES from SET

```

%% Only one signal accepted in this script
function IRES_Generation_for_SET_V5(FeatureUsed, CalVar, CalSkew, CalMean,
CalKur,...
    CalStd, WindowLength , NameSignals, samplerate, NumStats, NumFrames,...
    RecordLength, StateLevels, Shift, NumSignals, Normalize, data_color)
data = cell2mat(NameSignals);
samplerate_data = cell2mat(samplerate);
RecordLength_data = cell2mat(RecordLength);
NumFrames_data = cell2mat(NumFrames);
StateLevels_data = cell2mat(StateLevels);
FirstDataPoint = 0;
LastDataPoint = FirstDataPoint+RecordLength_data;
Inst_Period_added{: ,1} =
pulseperiod(data(FirstDataPoint+1:LastDataPoint), samplerate_data, 'StateLevels
', StateLevels_data);
transformed_signal_freq{: ,1} = 1 ./Inst_Period_added{: ,1};
number_of_pulses_in_each_frame{: ,1} = length(Inst_Period_added);
FirstDataPoint = LastDataPoint;
LastDataPoint = RecordLength_data+LastDataPoint;
for j = 2: NumFrames_data
    data_new = data(FirstDataPoint+1:LastDataPoint);
    Inst_Period =
pulseperiod(data_new, samplerate_data, 'StateLevels', StateLevels_data);
    number_of_pulses_in_each_frame{: ,j} = length(Inst_Period);
    Inst_Period_added{: ,j} = Inst_Period;
    transformed_signal_freq{: ,j} = 1 ./Inst_Period_added{: ,j};
    FirstDataPoint = LastDataPoint;
    LastDataPoint = RecordLength_data+LastDataPoint;
end
All_frames= Inst_Period_added;
clear data j;
for y=1:NumFrames{1,1}
    if strcmp(FeatureUsed, 'rt')
        data = cell2mat(Inst_rise_Total(:,q));
        rt{q}= data;
        transformed_signal{q} = rt{q};
        clearvars n rt data ;
    end
end

```

```

% Calculate cycle to cycle jitter
if strcmp(FeatureUsed, 'c2c')
    data = cell2mat(Inst_Period_Total(:,q));
    for n = 1:length(data)-1
        cc(n,:) = data(n+1,1)-data(n,1);
    end
    c2c{q}= cc;
    transformed_signal{q} = c2c{q};

    clearvars n cc data;
end
% Calculate cycle to mean jitter
if strcmp(FeatureUsed, 'c2m')
    data = cell2mat(All_frames(1,y));
    data_mean = mean(data);
    for n = 1:length(data)
        cm(n,:) = data(n,1)-data_mean;
    end
c2m{1}= cm;
    transformed_signal{1,y} = c2m{1};
    clearvars n data cm c2m data_mean;
end
% Calculate instantaneous frequency
if strcmp(FeatureUsed, 'f')
    data = cell2mat(Inst_Period_Total(:,q));
    Freq = 1 ./ data;
    Inst_Freq{q} = Freq;
    transformed_signal{q} = Inst_Freq{q};
    clear data Freq Inst_Freq;
end
end
clear p j q y data;

for y=1:NumFrames{1,1}
    data = cell2mat(transformed_signal(1,y));
    data_length = length(data);
    initChunk = 0;
    endChunk = WindowLength;
    end_k = ceil((data_length-WindowLength)/Shift);
    for k = 1:end_k
        RegionVar(:,k,y) = var(data(initChunk+1:endChunk));
        RegionSkew(:,k,y) = skewness(data(initChunk+1:endChunk));
        RegionKur(:,k,y) = kurtosis(data(initChunk+1:endChunk));
        RegionStd(:,k,y) = std(data(initChunk+1:endChunk));
        RegionMean(:,k,y) = mean(data(initChunk+1:endChunk));
        initChunk = Shift + initChunk;
        endChunk = Shift + endChunk;
    end
    clear data_length;
end

clear k p y end_k data_length transformed_signal samplerate initChunk
endChunk;
clear NameSignals number_of_pulses_in_each_frame;

for y=1:NumFrames{1,1}

```

```

data = cell2mat(transformed_signal_freq(1,y));
data_length = length(data);
initChunk = 0;
endChunk = WindowLength;
end_k = ceil((data_length-WindowLength)/Shift);
for k = 1:end_k
    RegionMean_Freq(:,k,y) = mean(data(initChunk+1:endChunk));
    initChunk = Shift + initChunk;
    endChunk = Shift + endChunk;
end
clear data_length;
end
[M_Kur,I_Kur] = max(RegionKur);
[M_Skew,I_Skew] = max(RegionSkew);
RegionKur(:, :, y) = circshift(RegionKur(:, :, y), -I_Kur(:, :, y)+1);
RegionSkew(:, :, y) = circshift(RegionSkew(:, :, y), -I_Kur(:, :, y)+1);
RegionVar(:, :, y) = circshift(RegionVar(:, :, y), -I_Kur(:, :, y)+1);
RegionStd(:, :, y) = circshift(RegionStd(:, :, y), -I_Kur(:, :, y)+1);
RegionMean(:, :, y) = circshift(RegionMean(:, :, y), -I_Kur(:, :, y)+1);
RegionMean_Freq(:, :, y) = circshift(RegionMean_Freq(:, :, y), -I_Kur(:, :, y)+1);
clear k end_k p y data;
RegionKur = RegionKur(1:end-I_Kur+1);
RegionSkew = RegionSkew(1:end-I_Kur+1);
RegionVar = RegionVar(1:end-I_Kur+1);
RegionStd = RegionStd(1:end-I_Kur+1);
RegionMean = RegionMean(1:end-I_Kur+1);
RegionMean_Freq = RegionMean_Freq(1:end-I_Kur+1);
if Normalize == 1
    RegionVar = repmat(RegionVar ./max(max(RegionVar)), 3, 1);
    RegionSkew = repmat((RegionSkew ./max(max(RegionSkew))), 3, 1);
    RegionKur = repmat(RegionKur ./max(max(RegionKur)), 3, 1);
    RegionStd = repmat(RegionStd ./max(max(RegionStd)), 3, 1);
    RegionMean = repmat(RegionMean ./max(max(RegionMean)), 3, 1);
    RegionMean_Freq = repmat(RegionMean_Freq
./max(max(RegionMean_Freq)), 3, 1);
elseif Normalize == 0
    RegionVar = repmat(RegionVar ./ (max(RegionVar)), 3, 1);
    RegionSkew = repmat((RegionSkew ./ (max(RegionSkew))), 3, 1);
    RegionKur = repmat(RegionKur ./ (max(RegionKur)), 3, 1);
    RegionStd = repmat(RegionStd ./ (max(RegionStd)), 3, 1);
    RegionMean = repmat(RegionMean ./ (max(RegionMean)), 3, 1);
    RegionMean_Freq = repmat(RegionMean_Freq ./ (max(RegionMean_Freq)), 3, 1);
end
if strcmp(FeatureUsed, 'f')
    for s = 1:NumFrames{1,1}
        if (CalVar == 0) && (CalStd == 0) && (CalMean == 0) && (CalKur == 0)
&& (CalSkew == 1)
            IPrint(:, :, s) = RegionSkew(:, :, s);
            StatLabel = {'Skew'};
        elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 0)
            IPrint(:, :, s) = RegionKur(:, :, s);
            StatLabel = {'Kur'};
        elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 1)
            IPrint = [RegionKur(:, :, s);RegionSkew(:, :, s)];

```

```

        StatLabel = {'Kur', 'Skew'};
    elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 0)
        IPrnt(:, :, s) = RegionMean(:, :, s);
        StatLabel = {'Mean'};
    elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 1)
        IPrnt(:, :, s) = [RegionMean(:, :, s); RegionSkew(:, :, s)];
        StatLabel = {'Mean', 'Skew'};
    elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 0)
        IPrnt(:, :, s) = [RegionMean(:, :, s); RegionKur(:, :, s)];
        StatLabel = {'Mean', 'Kur'};
    elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 1)
        IPrnt(:, :, s) =
[RegionMean(:, :, s); RegionKur(:, :, s); RegionSkew(:, :, s)];
        StatLabel = {'Mean', 'Kur', 'Skew'};
    elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
0) && (CalSkew == 0)
        IPrnt(:, :, s) = RegionStd(:, :, s);
        StatLabel = {'Std'};
    elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
0) && (CalSkew == 1)
        IPrnt(:, :, s) = [RegionStd(:, :, s); RegionSkew(:, :, s)];
        StatLabel = {'Std', 'Skew'};
    elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 0)
        IPrnt(:, :, s) = [RegionStd(:, :, s); RegionKur(:, :, s)];
        StatLabel = {'Std', 'Kur'};
    elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 1)
        IPrnt(:, :, s) =
[RegionStd(:, :, s); RegionKur(:, :, s); RegionSkew(:, :, s)];
        StatLabel = {'Std', 'Kur', 'Skew'};
    elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 0)
        IPrnt(:, :, s) = [(RegionStd(:, :, s)); (RegionMean(:, :, s))];
        StatLabel = {'Std', 'Mean'};
    elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 1)
        IPrnt(:, :, s) =
[RegionStd(:, :, s); RegionMean(:, :, s); RegionSkew(:, :, s)];
        StatLabel = {'Std', 'Mean', 'Skew'};
    elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 0)
        IPrnt(:, :, s) =
[RegionStd(:, :, s); RegionMean(:, :, s); RegionKur(:, :, s)];
        StatLabel = {'Std', 'Mean', 'Kur'};
    elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 1)
        IPrnt(:, :, s) =
[RegionStd(:, :, s); RegionMean(:, :, s); RegionKur(:, :, s); RegionSkew(:, :, s)];
        StatLabel = {'Std', 'Mean', 'Kur', 'Skew'};
    elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 0) && (CalKur ==
0) && (CalSkew == 0)

```

```

        IPrint(:, :, s) = RegionVar(:, :, s);
        StatLabel = {'Var'};
        elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 0) && (CalKur ==
0) && (CalSkew == 1)
            IPrint(:, :, s) = [RegionVar(:, :, s); RegionSkew(:, :, s)];
            StatLabel = {'Var', 'Skew'};
            elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 0)
                IPrint(:, :, s) = [RegionVar(:, :, s); RegionKur(:, :, s)];
                StatLabel = {'Var', 'Kur'};
                elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 1)
                    IPrint(:, :, s) =
[RegionVar(:, :, s); RegionKur(:, :, s); RegionSkew(:, :, s)];
                    StatLabel = {'Var', 'Kur', 'Skew'};
                    elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 0)
                        IPrint(:, :, s) = [RegionVar(:, :, s); RegionMean(:, :, s)];
                        StatLabel = {'Var', 'Mean'};
                        elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 1)
                            IPrint(:, :, s) =
[RegionVar(:, :, s); RegionMean(:, :, s); RegionSkew(:, :, s)];
                            StatLabel = {'Var', 'Mean', 'Skew'};
                            elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 0)
                                IPrint(:, :, s) = [RegionVar(:, :, s);
RegionMean(:, :, s); RegionKur(:, :, s)];
                                StatLabel = {'Var', 'Mean', 'Kur'};
                                elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 1)
                                    IPrint(:, :, s) =
[RegionVar(:, :, s); RegionMean(:, :, s); RegionKur(:, :, s); RegionSkew(:, :, s)];
                                    StatLabel = {'Var', 'Mean', 'Kur', 'Skew'};
                                    elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
0) && (CalSkew == 0)
                                        IPrint(:, :, s) = [RegionVar(:, :, s); RegionStd(:, :, s)];
                                        StatLabel = {'Var', 'Std'};
                                        elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
0) && (CalSkew == 1)
                                            IPrint(:, :, s) =
[RegionVar(:, :, s); RegionStd(:, :, s); RegionSkew(:, :, s)];
                                            StatLabel = {'Var', 'Std', 'Skew'};
                                            elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 0)
                                                IPrint(:, :, s) =
[RegionVar(:, :, s); RegionStd(:, :, s); RegionKur(:, :, s)];
                                                StatLabel = {'Var', 'Std', 'Kur'};
                                                elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 1)
                                                    IPrint(:, :, s) =
[RegionVar(:, :, s); RegionStd(:, :, s); RegionKur(:, :, s); RegionSkew(:, :, s)];
                                                    StatLabel = {'Var', 'Std', 'Kur', 'Skew'};
                                                    elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 0)

```

```

        IPrint(:, :, s) =
[RegionVar(:, :, s);RegionStd(:, :, s);RegionMean(:, :, s)];
        StatLabel = {'Var', 'Std', 'Mean'};
        elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 1)
        IPrint(:, :, s) =
[RegionVar(:, :, s);RegionStd(:, :, s);RegionMean(:, :, s);RegionSkew(:, :, s)];
        StatLabel = {'Var', 'Std', 'Mean', 'Skew'};
        elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 0)
        IPrint(:, :, s) =
[RegionVar(:, :, s);RegionStd(:, :, s);RegionMean(:, :, s);RegionKur(:, :, s)];
        StatLabel = {'Var', 'Std', 'Mean', 'Kur'};
        elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 1)
        IPrint(:, :, s) =
[RegionVar(:, :, s);RegionStd(:, :, s);RegionMean(:, :, s);RegionKur(:, :, s);RegionS
kew(:, :, s)];
        StatLabel = {'Var', 'Std', 'Mean', 'Kur', 'Skew'};
    end
end
else
    for s = 1:NumFrames{1,1}
        if (CalVar == 0) && (CalStd == 0) && (CalMean == 0) && (CalKur == 0)
&& (CalSkew == 1)
            IPrint(:, :, s) = RegionSkew(:, :, s);
            StatLabel = {'Skew'};
            elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 0)
            IPrint(:, :, s) = RegionKur(:, :, s);
            StatLabel = {'Kur'};
            elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 1)
            IPrint = [RegionKur(:, :, s);RegionSkew(:, :, s)];
            StatLabel = {'Kur', 'Skew'};
            elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 0)
            IPrint(:, :, s) = RegionMean(:, :, s);
            StatLabel = {'Mean'};
            elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 1)
            IPrint(:, :, s) = [RegionMean(:, :, s);RegionSkew(:, :, s)];
            StatLabel = {'Mean', 'Skew'};
            elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 0)
            IPrint(:, :, s) = [RegionMean(:, :, s);RegionKur(:, :, s)];
            StatLabel = {'Mean', 'Kur'};
            elseif (CalVar == 0) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 1)
            IPrint(:, :, s) =
[RegionMean(:, :, s);RegionKur(:, :, s);RegionSkew(:, :, s)];
            StatLabel = {'Mean', 'Kur', 'Skew'};
            elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
0) && (CalSkew == 0)
            IPrint(:, :, s) = RegionStd(:, :, s);
            StatLabel = {'Std'};

```

```

elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
0) && (CalSkew == 1)
    IPrnt(:, :, s) = [RegionStd(:, :, s); RegionSkew(:, :, s)];
    StatLabel = {'Std', 'Skew'};
elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 0)
    IPrnt(:, :, s) = [RegionStd(:, :, s); RegionKur(:, :, s)];
    StatLabel = {'Std', 'Kur'};
elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 1)
    IPrnt(:, :, s) =
[RegionStd(:, :, s); RegionKur(:, :, s); RegionSkew(:, :, s)];
    StatLabel = {'Std', 'Kur', 'Skew'};
elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 0)
    IPrnt(:, :, s) = [(RegionStd(:, :, s)); (RegionMean(:, :, s))];
    StatLabel = {'Std', 'Mean'};
elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 1)
    IPrnt(:, :, s) =
[RegionStd(:, :, s); RegionMean(:, :, s); RegionSkew(:, :, s)];
    StatLabel = {'Std', 'Mean', 'Skew'};
elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 0)
    IPrnt(:, :, s) =
[RegionStd(:, :, s); RegionMean(:, :, s); RegionKur(:, :, s)];
    StatLabel = {'Std', 'Mean', 'Kur'};
elseif (CalVar == 0) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 1)
    IPrnt(:, :, s) =
[RegionStd(:, :, s); RegionMean(:, :, s); RegionKur(:, :, s); RegionSkew(:, :, s)];
    StatLabel = {'Std', 'Mean', 'Kur', 'Skew'};
elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 0) && (CalKur ==
0) && (CalSkew == 0)
    IPrnt(:, :, s) = RegionVar(:, :, s);
    StatLabel = {'Var'};
elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 0) && (CalKur ==
0) && (CalSkew == 1)
    IPrnt(:, :, s) = [RegionVar(:, :, s); RegionSkew(:, :, s)];
    StatLabel = {'Var', 'Skew'};
elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 0)
    IPrnt(:, :, s) = [RegionVar(:, :, s); RegionKur(:, :, s)];
    StatLabel = {'Var', 'Kur'};
elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 1)
    IPrnt(:, :, s) =
[RegionVar(:, :, s); RegionKur(:, :, s); RegionSkew(:, :, s)];
    StatLabel = {'Var', 'Kur', 'Skew'};
elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 0)
    IPrnt(:, :, s) = [RegionVar(:, :, s); RegionMean(:, :, s)];
    StatLabel = {'Var', 'Mean'};
elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 1)

```



```

        IPrint(:, :, s) =
[RegionVar(:, :, s);RegionMean(:, :, s);RegionSkew(:, :, s)];
        StatLabel = {'Var', 'Mean', 'Skew'};
        elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 0)
        IPrint(:, :, s) = [RegionVar(:, :, s);
RegionMean(:, :, s);RegionKur(:, :, s)];
        StatLabel = {'Var', 'Mean', 'Kur'};
        elseif (CalVar == 1) && (CalStd == 0) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 1)
        IPrint(:, :, s) =
[RegionVar(:, :, s);RegionMean(:, :, s);RegionKur(:, :, s);RegionSkew(:, :, s)];
        StatLabel = {'Var', 'Mean', 'Kur', 'Skew'};
        elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
0) && (CalSkew == 0)
        IPrint(:, :, s) = [RegionVar(:, :, s);RegionStd(:, :, s)];
        StatLabel = {'Var', 'Std'};
        elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
0) && (CalSkew == 1)
        IPrint(:, :, s) =
[RegionVar(:, :, s);RegionStd(:, :, s);RegionSkew(:, :, s)];
        StatLabel = {'Var', 'Std', 'Skew'};
        elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 0)
        IPrint(:, :, s) =
[RegionVar(:, :, s);RegionStd(:, :, s);RegionKur(:, :, s)];
        StatLabel = {'Var', 'Std', 'Kur'};
        elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 0) && (CalKur ==
1) && (CalSkew == 1)
        IPrint(:, :, s) =
[RegionVar(:, :, s);RegionStd(:, :, s);RegionKur(:, :, s);RegionSkew(:, :, s)];
        StatLabel = {'Var', 'Std', 'Kur', 'Skew'};
        elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 0)
        IPrint(:, :, s) =
[RegionVar(:, :, s);RegionStd(:, :, s);RegionMean(:, :, s)];
        StatLabel = {'Var', 'Std', 'Mean'};
        elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
0) && (CalSkew == 1)
        IPrint(:, :, s) =
[RegionVar(:, :, s);RegionStd(:, :, s);RegionMean(:, :, s);RegionSkew(:, :, s)];
        StatLabel = {'Var', 'Std', 'Mean', 'Skew'};
        elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 0)
        IPrint(:, :, s) =
[RegionVar(:, :, s);RegionStd(:, :, s);RegionMean(:, :, s);RegionKur(:, :, s)];
        StatLabel = {'Var', 'Std', 'Mean', 'Kur'};
        elseif (CalVar == 1) && (CalStd == 1) && (CalMean == 1) && (CalKur ==
1) && (CalSkew == 1)
        IPrint(:, :, s) =
[RegionMean_Freq(:, :, s);RegionMean(:, :, s);RegionStd(:, :, s);RegionVar(:, :, s);R
egionKur(:, :, s);RegionSkew(:, :, s)];
        StatLabel = {'Mean Freq', 'Mean', 'Std', 'Var', 'Kur', 'Skew'};
    end
end
end

```

```

clear s CalKur CalMean CalSkew CalStd CalVar;
for p = 1:NumFrames{1,1}
    f{p} = figure('visible','on','units','normalized');
    colormap(jet);
    axes1{p} = axes('Parent',f{p});
    contourf(IPrnt(:, :, p), 100, 'LineStyle', 'none');
    box(axes1{p}, 'on');
    axis(axes1{p}, 'tight');
    set(axes1{p}, 'XTick', [], 'XTickLabel', [], 'YTick', [2 5 8 11 14
17], 'YTickLabel', {'Mean
Frequency', 'Mean', 'Std', 'Var', 'Kur', 'Skew'}, 'FontSize', 24);

end

end

```

VITA

Bharat Patel was born in Gujarat, India to the parents of Alpa and Paresh Patel. He is the only child of his parents. He graduated from School of Achiever in 2010 and pursued Electronics and Communication Engineering in LDRP Institute of Technology and Research at Gujarat Technological University. He moved to United States of America in 2013 and attended Cleveland State Community College and University of Tennessee at Chattanooga (UTC). In 2017, he graduated with Bachelor of Science in Electrical Engineering from the University of Tennessee at Chattanooga. He worked at Edwin Bohr Electronics in Chattanooga before starting the master's degree in Electrical Engineering at UTC. He started working as graduate research assistant at UTC under Dr. Daniel Loveless. Bharat graduated with a Master of Science degree in Electrical Engineering in December 2019. He hopes to pursue a career in the field of space avionics design in the industry.