5-2022

Incorporating in situ measurements of energy into a designed method of detection of radiation induced degradation in embedded systems

Delwyn Sam
University of Tennessee at Chattanooga, fvs291@mocs.utc.edu

Incorporating In Situ Measurements of Energy into a Designed Method of Detection of Radiation Induced Degradation in Embedded Systems

by

Delwyn Sam

Daniel Loveless, PhD
Guerry Associate Professor of Electrical Engineering
(Committee Chair)

Donald Reising, PhD
Associate Professor of Electrical Engineering
(Committee Member)

Incorporating In Situ Measurements of Energy into a Designed Method of Detection of Radiation Induced Degradation in Embedded Systems

by

Delwyn Sam

A Thesis Submitted to the Faculty of the University of Tennessee at Chattanooga in Partial Fulfillment of the Requirements of the Departmental Honors Distinction in Electrical Engineering

The University of Tennessee at Chattanooga
Chattanooga, Tennessee

May 2022

ABSTRACT

An embedded system, or a system-on-chip (SOC), is a miniaturized computer that combines hardware and software to accomplish a certain purpose. Embedded systems are used in various applications from household appliances to the automobile industry. Additionally, SOCs are used in space environments where the surroundings consist of radiation and extreme temperatures. In these harsh conditions, embedded systems must operate properly through the effects of radiation such as total ionizing dose (TID). Because embedded systems are filled with various components and modules, and since the internal signals of the system are often inaccessible, methods to locate which specific parts of the system are affected due to radiation are archaic and inefficient. The most common method of analyzing TID in SOC consists of a pass-fail strategy where the effects of TID are only known after the system is unable to communicate with the user. This thesis describes a novel method to identify the specific modules being affected by TID radiation through in situ power and energy measurements of the system and the performance of the components. The experimental methodology is demonstrated through measurement of radiation effects in the MSP430FR6989 microcontroller from Texas Instruments' and the EnergyTrace program that is installed in the system.

ACKNOWLEDGMENTS

I would like to first and foremost thank God for this wonderful opportunity and support. My Christian faith granted me much mental aid, motivation, and peace in this thesis writing process and I would not be here without it.

Next, I would like to thank my thesis advisor, Dr. Loveless, for accepting an unproven student like myself into his lab and introducing me into the unique world of radiation effects in microelectronics. His creativity, patience, and leadership were vital in this thesis and my general undergraduate career in research and studies. I would also like to thank Dr. Reising for being part of my thesis committee and for the insightful perspective he brings.

Finally, I would like to express my sincere gratitude towards my talented team: Jaekeon Kim, who was with me since the start of this project and aided me through the end; Stephen Lawrence for his extreme patience and contributing significantly to this project even when he was not obliged to; Adam Peterson, who was readily available to help me whenever I had a question; and Jake Carpenter, for helping in a manner that typically goes unnoticed yet leaves a meaningful impact.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

With the progressive interest and pursuit of space exploration, the vitality of functioning microelectronics is proportional with the exploration's growth. Current mainstream microelectronics on Earth cannot properly operate in the harsh space environment. When on Earth the electronics are protected by the Earth's atmosphere and geomagnetic fields from space radiation such as radiation belts, solar flares and cosmic rays [1]. However, when these electronics are transported into space, the probability of a particle or particles interacting with the components increases, engendering discrepancies within the part or system. Some of these discrepancies can be recoverable, yet some destructive to the specific device. Hence, when using electronics in space, the radiation affecting them must be studied, and the structures must be modified, or hardened, when exported outside of the atmosphere to behave as needed.

Within the field of microelectronics, the subset study of radiation effects in embedded systems consists of analyzing the performance of microcontrollers (MCU), microprocessors, and other Systems-on-chips (SOC) as they are exposed to radiation and various harsh space environments. As stated in [9], "An embedded system combines mechanical, electrical, and chemical components along with a computer, hidden inside, to perform a single dedicated purpose." The usage of embedded systems is prevalent in various technologies like automobiles, household appliances, planes, satellites, computers, hospital machines, and radios. The reliability of these embedded systems is crucial to a functioning society. Proceeding deeper within the field of embedded systems, microcontrollers consist of a miniature computer (mi-

croprocessor) in conjunction with physical components such as switches, input/output pins, and sensors, to execute a certain task. Microcontrollers, compared to a microprocessor and field programmable gate array (FPGA), are less versatile and less programmable as it is more restricted by physical components than the other embedded systems.

Within radiation effects, the three primary categories of effects on microelectronics are Single Event Effects (SEE), Total Ionizing Dose (TID), and Displacement Damage [10]. The category of focus in this paper for embedded systems is the characterization of TID in microcontrollers. TID effects are the result of charge accumulation in insulating portions of electronic devices like transistors [4], which may result in shifts in threshold parameters, leakage current, and decreased functionality.

Analysis of TID in individual transistors or isolated circuits, such as an amplifier, are simpler when compared to the analysis of an embedded system. For an amplifier, because the location of each transistor in the amplifier is known, it easier to detect which specific transistors are being affected due to radiation. The issue when analyzing radiation effects in an embedded system like a microcontroller, for instance, is that microcontrollers consist of several clocks, memories, communication modules, DC/DC converters, high number of inputs/outputs (I/Os), and because of these various components packaged into a single integrated chip, internal signals are often inaccessible. Therefore, due to this inaccessibility, it is difficult, or nearly impossible, to triangulate which components are being affected when radiation enters the system [5].

The objective of this thesis is to describe a method to identify which specific modules of the microcontroller are being affected due to total ionizing dose radiation by analyzing energy and performance profiles of the SOC and its components.

CHAPTER 2

Background

2.1 Total Ionizing Dose Physics

Total ionizing dose radiation impacts the insulated regions of a transistor by creating electron-hole pairs when interacting with the semiconductor material. Eventually through accumulated dosage, the holes are deeply trapped near the silicon/silicon dioxide interface and start attracting electrons from the Silicon to the holes, forming a small inversion channel for current to flow from the drain to source of the transistor [13]. This unintentional channel allows leakage current in the transistors, which can lead to power loss in the system [11]. Figure 2.1 shows the effect of ionizing particles in a transistor [10].

The principle of total ionizing dose effects described in figure 2.1 is for a singular transistor; however, combine numerous transistors together to form an amplifier, for example, it becomes more difficult to analyze TID effects in the system. Furthermore, combine several more amplifiers, transistors, and other components together to form a more complex system, it becomes significantly more arduous to characterize the TID effects in this more complex system. Nonetheless, as the accumulated dose increases, the leakage current within the system also increases [13], which consequently means that if the system can measure the current used to perform its function, it can comparatively measure any extra current that the system is using as well. This is precisely the method of interrogation that the design of this thesis explores: it will use a program that will capture energy and power profiles of the microcontroller when it is executing a particular task to characterize the effects of TID in the MCU. However, more of the concept behind the experiment's method will be explained

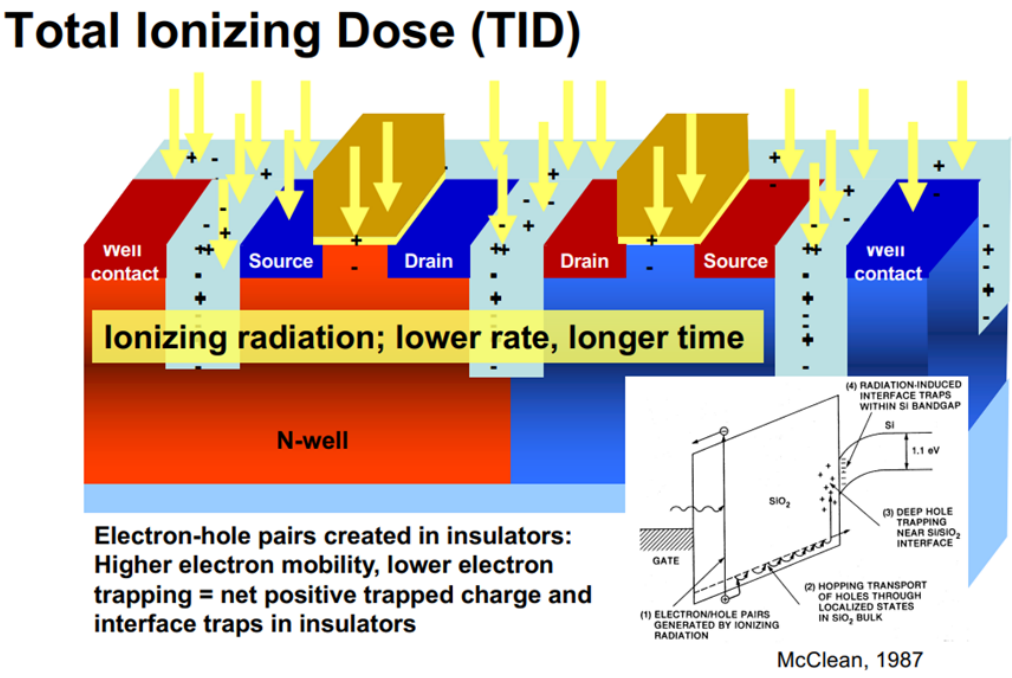


Figure 2.1 TID Physics Mechanism on a Transistor [10]

in the Methodology section.

2.2 Previous Methods

There are other methods of interrogation that are used to analyze dose accumulation in embedded systems. One of the current methods of analysis in the research field include component sampling, where radiation testers take three to four homogeneous components and test them together to view their TID response [13][14]. An example of component sampling is like taking multiple samples of a memory component like the Ferroelectric Random Access Memory (FRAM) and testing its performance after irradiation. However, component sampling seems to collect TID response data on only one component type which restricts the practical application of the testing method when used in embedded systems with multiple components. Another liability of this method is that many tests may not have access

to a room source, which "provides a real-time monitoring of the current and the functionality of the component simultaneously" [13]. This makes the testing process inefficient as testers must pause the test to take measurements of current consumption and component functionality.

Other previous methods of interrogating SOCs involve a pass-fail strategy where radiation is recognized only after the system has died and is no longer able to communicate with external sources. Therefore, TID testing "is usually considered a destructive test, in the sense that the tested device is no longer suitable for system use..." [6]. However, the problem of this testing method is also evident: the pass-fail strategy does not inform the tester of any current and power consumption data or the performance characteristics of the chip. Rather, it only gives a binary statistic of parametric failure: 1) able to communicate to system or 2) unable to communicate to system.

To combat these restrictions found in contemporary methods of TID testing in complex systems, a proper testing strategy must include one that measures any sub-circuit leakage current , possess the ability to test TID responses of multiple components simultaneously, and records parametric failures [13]. However, most importantly, the strategy must be able to do all these parameters concurrently. For example, the ability to measure leakage current must not limit the ability to measure multiple component functionality of the system. The ideal strategy encompasses all these facets to work harmoniously when testing complex systems for total ionizing dose, and this paper demonstrates this ideal strategy by using Texas Instrument's program called EnergyTrace [22].

CHAPTER 3

Methodology

3.1 Overview

To design a novel method for identification of radiation-induced degradation mechanisms within a microcontroller, the chosen device under test (DUT) is the Texas Instruments' (TI) MSP430FR6989 (MSP430) microcontroller. This MCU has numerous components and hardware peripherals combined into one body to operate collectively and, after the DUT is irradiated, the tester will execute a program to activate specific components to analyze the components' and the overall MCU's performance in response to radiation. While the methods described herein are technology agnostic and can be applied to other similar devices, the MSP430 has TI's EnergyTrace software preinstalled, which can be used to develop in situ energy and power profiles for the microcontroller. Then, degradation of these profiles following controlled radiation exposure will provide insight into radiation degradation mechanisms that are otherwise intractable.

3.2 Hardware Utilized and Test Setup

1. MSP430FR6989
2. Raspberry Pi
3. RS-232
4. External digital-to-analog converter
5. Computer (PC) to run programs and govern experiment

6. Ethernet Hub
7. Tektronix Digital Oscilloscope
8. Protoboard as a station for wired connections
9. USB to micro-USB wires
10. BNC cables and ports
11. Aracore X-ray source
12. USB flash drive

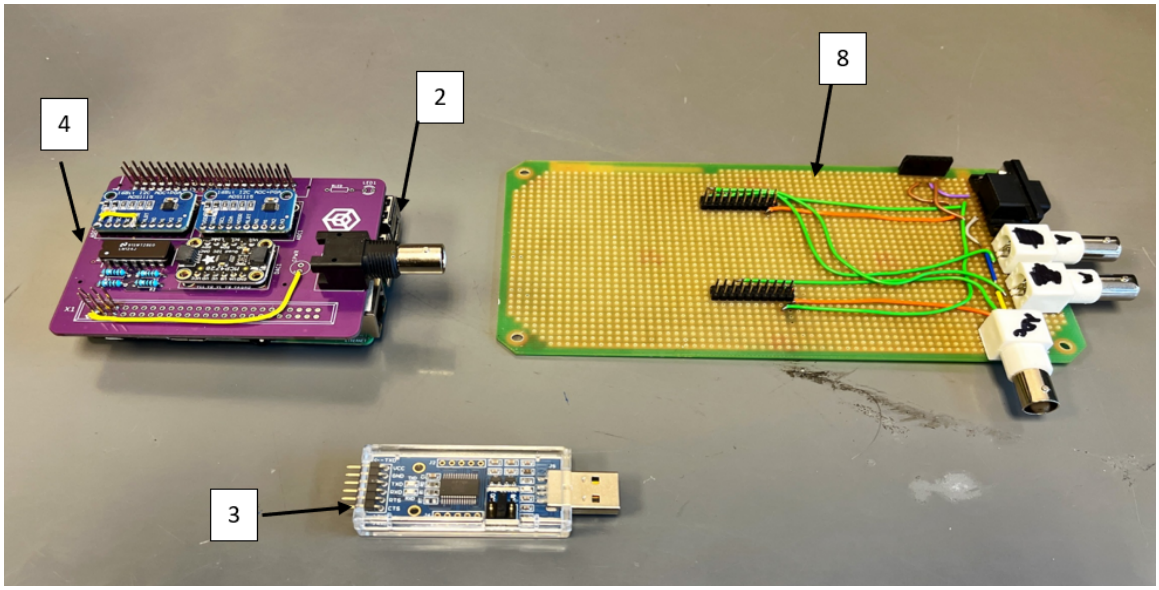


Figure 3.1 Raspberry Pi (4), DAC (2), RS-232 (3), and Protoboard (8)

Initially a printed circuit board (PCB) was designed and printed using Autodesk's Eagle resource instead of the hardware no. 8 (protoboard); however, the board did not comply to the design specifications of the MSP430, so it had to be replaced by a soldered protoboard. Also, to allow the X-rays to enter the chip, the packaging of the MCU had to be decapsulated and exposed as X-rays are not energized enough to penetrate the chip covering. The MSP430s were shipped to a company in California to decapsulate the chips. The schematics of the test setup are shown in Figure 3.2. Figure 3.3 shows the assembled test setup of the MSP430 with its wired connections in the Aracore X-ray source.

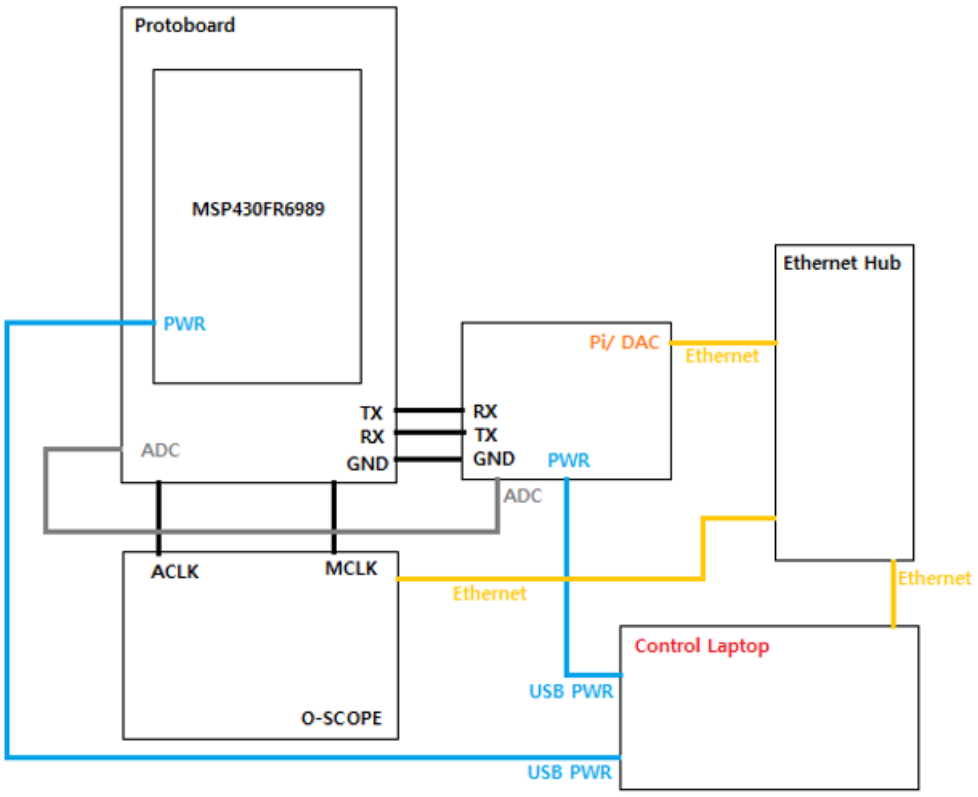


Figure 3.2 Test setup schematics

3.3 Test Procedure

Five tests were done on five separate MSP430s. Three of the five DUTs were programmed to operate in LPM4, another one programmed to operate in LPM0, and the last one programmed to operate in LPM2. Once all the wired connections, MSP430, oscilloscope, Raspberry Pi, and PC are assembled as shown in Figures 3.2 and 3.3, the X-ray source beam was closed in a metal chamber to prevent radiation exposure to the environment. Because the time length of each trial must be brief and efficient, the Python code from the Jupyter Notebook should start running now to save ADC data when ready. Then the chip was irradiated in dose increments of 5.44 krad (SiO2) starting from 0 krad (SiO2) till the MSP430

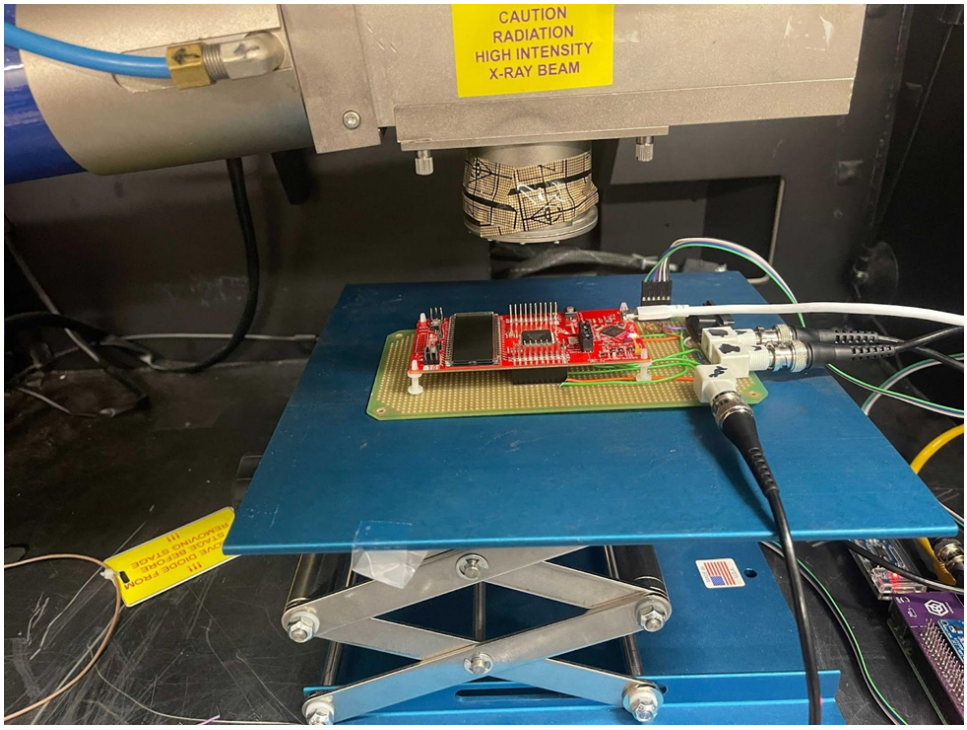


Figure 3.3 Assembled test setup

stops communicating to the PC (i.e., stops functioning). Table 5.1 in Appendix B shows the dose points for each trial. A single trial occurs once a dose point is reached (i.e., after the X-ray source irradiates the chip by 5.44 krad (SiO2)).

Once the MSP430 is irradiated to the specific dose point, the CCS code is executed in the MCU to activate the clock system and the ADC modules. First, the clock signals (ACLK and then MCLK) will be recorded and saved in the oscilloscope; then the ADC will activate and convert the analog input voltage values to digital values and save them in a CSV file in the PC; after the clocks and ADC are saved, the MSP430 enters the specified LPM for a short period of time till the program ends. Afterward, the EnergyTrace data is manually saved into the PC. This completes one trial, which takes about 1 minute and 40 seconds. Each trial was originally set to take only 30 seconds; however, the first trial took 1 minute and 40 seconds, so to keep the variables of each trial as consistent as possible, every trial was now set to take 1 minute and 40 seconds. A soft reset is done in the CCS IDE

for the CCS code, that way the MSP430 is not continually running the program but is in standby mode when it is irradiated again. Once the next incremental radiation dose point is reached, another trial is executed. This process repeats till there is a discontinuation of the communication between PC and the chip, which evident when the CCS code is no longer able to run on the MSP430.

3.4 Radiation Source

An ARACOR 10-keV x-ray irradiator was used to dose the MCU. The MCU package was de-lidded to expose the underlying electronics to the radiation. X-rays contain enough energy to dose a chip quickly and it was readily available to use for this thesis experiment. Figure 3.4 shows the different energy levels for different electromagnetic waves and particles, such as X-rays [11]. The MSP430 is irradiated out of a 2cm diameter beam from the X-ray source.

Initially, because the amount of dosage the MSP430 can withstand before failure was unknown, the first MSP430 part that was tested was irradiated at smaller dose increments of 2.72krad (SiO2) per trial. Once the threshold dosage was known from the first part, the following parts had higher dose increments of 5.44krad (SiO2).

3.5 Modules Under Test

The MSP430 consists of various modules and peripherals that allow it to execute a range of functions. Figure 3.5 shows all the modules within the MSP430FR6989 device [15].

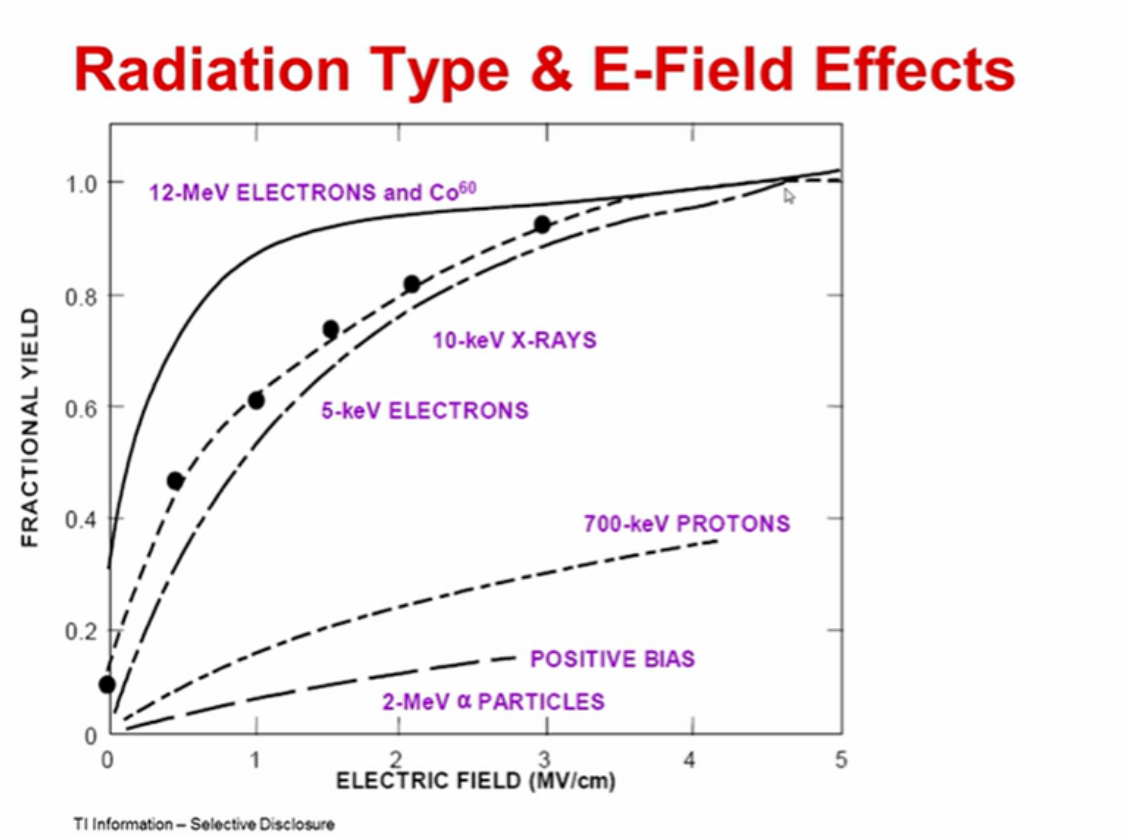


Figure 3.4 Radiation types and their effect on electric fields

3.5.1 Clock System

One of the most important parts within most embedded systems is the Clock System which contains all the clock sources and modules needed to govern the operating speed of the CPU, the memory bus, and other peripherals [14]. The clock sources also dictate the speed at which the "processor executes instructions, the baud rate of serial-communication signals, the amount of time needed to perform an analog-to-digital conversion, and so much more" [14]. Therefore, testing the Clock System module is crucial in analyzing the functionality of the MCU. The MSP430's user's guide has a list of all the clock sources and the clock signals associated with the clock sources in Figure 3.6.

The clock sources of interest in this experiment are the DCOCLK, which is a digitally controlled oscillator (DCO) with selectable frequencies, and the VLOCLK, which is a very

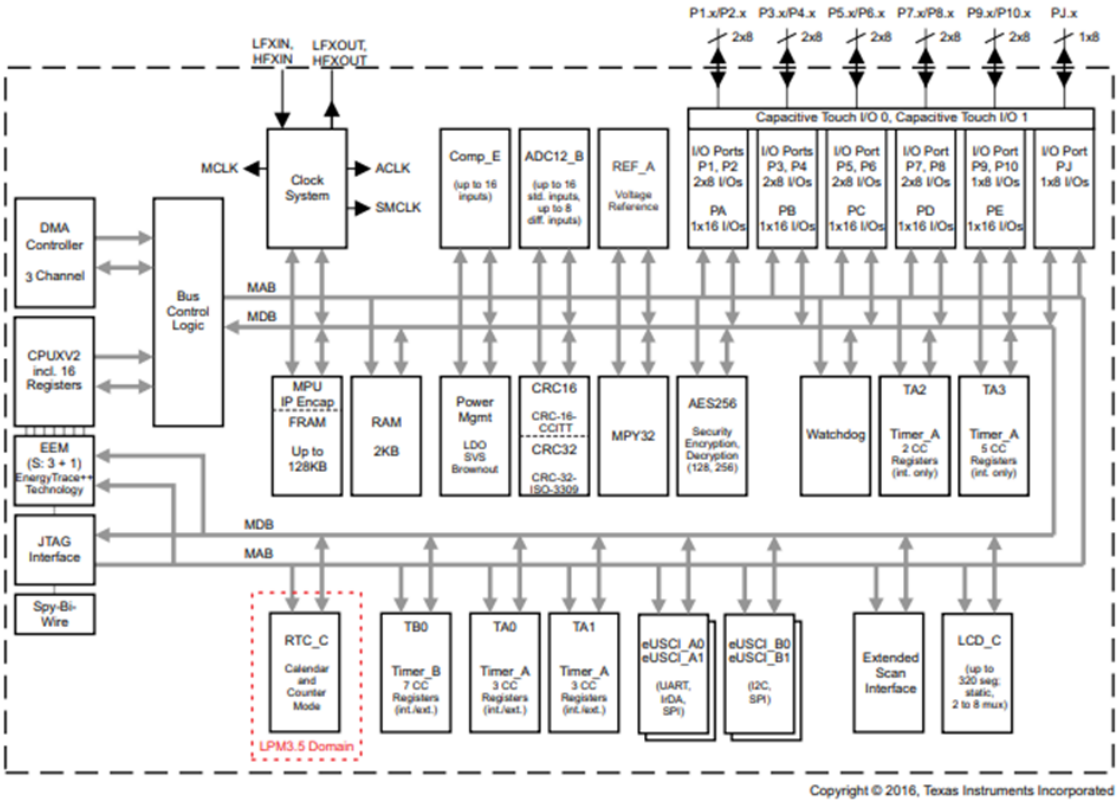


Figure 3.5 Functional Block Diagram of the MSP430FR698x [15]

low-frequency oscillator (VLO). The reason for choosing the DCO is because it provides a wide range of frequency options from 2 MHz to 24 MHz [15]. In this test the DCO was arbitrarily set to 16MHz. The VLO clock source was chosen to run a clock signal at a low frequency. In this test the VLO was set to 10KHz. The clock sources are then connected to two clock signals to output the waveforms of the clocks into general-purpose input/output (GPIO) pins. The DCO clock is connected to the Master Clock (MCLK) module and the VLO clock is connected to the Auxiliary Clock (ACLK). The signals are then programmed using the DUT's integrated development environment (IDE) to be outputted into GPIO pins, which are connected to an oscilloscope to capture the waveform.

The oscilloscope is configured to show at least hundred periods of the clock signal and it records the waveform and converts the data into a CSV file to store in an external drive. The data from the flash drive is then transferred to a personal computer (PC) where a

The clock system module includes the following clock sources:

- LFXTCLK: Low-frequency oscillator that can be used either with low-frequency 32768-Hz watch crystals, standard crystals, resonators, or external clock sources in the 50 kHz or below range. When in bypass mode, LFXTCLK can be driven with an external square wave signal.
- VLOCLK: Internal very-low-power low-frequency oscillator with 10-kHz typical frequency
- DCOCLK: Internal digitally controlled oscillator (DCO) with selectable frequencies
- MODCLK: Internal low-power oscillator with 5-MHz typical frequency. LFMODCLK is MODCLK divided by 128.
- HFXTCLK: High-frequency oscillator that can be used with standard crystals or resonators in the 4-MHz to 24-MHz range. When in bypass mode, HFXTCLK can be driven with an external square wave signal.

Four system clock signals are available from the clock module:

- ACLK: Auxiliary clock. The ACLK is software selectable as LFXTCLK, VLOCLK, or LFMODCLK. ACLK can be divided by 1, 2, 4, 8, 16, or 32. ACLK is software selectable by individual peripheral modules.
- MCLK: Master clock. MCLK is software selectable as LFXTCLK, VLOCLK, LFMODCLK, DCOCLK, MODCLK, or HFXTCLK. MCLK can be divided by 1, 2, 4, 8, 16, or 32. MCLK is used by the CPU and system.
- SMCLK: Subsystem master clock. SMCLK is software selectable as LFXTCLK, VLOCLK, LFMODCLK, DCOCLK, MODCLK, or HFXTCLK. SMCLK is software selectable by individual peripheral modules.
- MODCLK: Module clock. MODCLK may also be used by various peripheral modules and is sourced by MODOSC.
- VLOCLK: VLO clock. VLOCLK may also be used directly by various peripheral modules and is sourced by VLO.

NOTE: Not all devices contain both LFXT and HFXT clock sources. See the device-specific data sheet for availability.

Figure 3.6 Information of the clock sources and clock modules of the MSP40

python program is then used to calculate the frequency of the signals. After the frequencies are derived from the signal data, they will be compared to the control, non-radiated clock signal data. If there are any variations between the frequency of the irradiated clock signal and the frequency of the control clock signal, it could be attributed to TID effects. In conclusion, clock system analysis will be based on the variations of the signal frequencies.

3.5.2 Analog-to-Digital Converter Module

A significant portion of the tasks assigned to microcontrollers, and embedded systems in general, involves analyzing input data to perform certain output functions [9]. The analog-to-digital converter (ADC) module is vital in accomplishing this task as it "converts any analog signal into quantifiable data, which makes it easier to process and store" [17]. If the

system's ADC fails to work properly due to radiation, it can be detrimental to any data collection, which can lead to faults when the system is attempting to execute its purpose. Therefore, this test will also analyze the functionality of the ADC in response to TID.

For this test, a Raspberry Pi microprocessor is used in conjunction with an external digital-to-analog converter (DAC) to supply a controlled analog DC voltage ramp of 0V to 3.3V into the input of the MSP430's ADC module, which converts the analog voltage to a digital value. The digital value is then recorded in the MSP430's memory and an RS-232 (a serial data transmission device) is used to transfer the converted value from the MSP430 to the PC. The input voltage values are also saved into the Raspberry Pi's memory. To analyze the effects of radiation within the ADC module, a python program is used to compare the input voltage values to the output voltage values to find the error percentage between the values using Equation 3.1.

$$ErrorPercentage = \frac{|Vout - Vin|}{Vin} * 100 \tag{3.1}$$

Using the error percentages of each trial, the mean, the variance, and the standard deviation of the trial is calculated using a python program. The greater the mean, the variance, and standard deviation, the more likely the ADC module is affected by the TID radiation.

3.5.3 Peripheral Modules

Though the primary modules of interest are the Clock System and ADC modules, there are other peripheral modules that are indirectly tested. The Enhanced Universal Serial Communication Interface (eUSCI) module uses the Universal Asynchronous Receiver-Transmitter (UART) subset module to transmit data from the MSP430 to another device like the Raspberry Pi or a PC. This experiment uses a serial data transmission device, the RS-232, in conjunction with the MSP430's UART module to transfer ADC data from the

MSP430 to the host computer. Because these modules are only in use to test another module, no extensive research has been done to study how TID can potentially affect the eUSCI and UART modules. The only current method of testing them for TID responses is through the pass-fail strategy. The converted ADC values that are stored in the MSP430 can be viewed through the IDE; therefore, if the ADC values are being converted and the tester is able to see the stored converted values through the IDE, but those converted values are not being transferred to the PC through the UART module, then the tester can conclude that the TID radiation has possibly discontinued the functionality of the eUSCI and UART modules.

3.6 Low Power Modes

In addition to the different modules of the MSP430 that are being tested, the Low Power Mode (LPM) is another condition that will be studied for radiation response. Because the MSP430 is "designed for ultralow power applications" [15] it operates in various power states to save energy. The different power modes are Active Power mode, LPM0, LPM1, LPM2, LPM3, LPM4, LPM3.5, LPM4.5. When the MCU is in Active Power mode it uses the most amount of power and the higher the number is in the LPMx, the less amount of power is used by the MCU. In each subsequently higher LPM mode, more modules, like the CPU and Clock System, get deactivated and the MCU uses less energy to perform its function. Table 3.1 shows the different operation modes and the modules that are affected by each LPM. This test also seeks to understand whether different LPMs respond differently to TID radiation. A minor hypothesis that this experiment seeks to test is that the higher the LPM, the more susceptible the MCU will be to radiation effects. For example, if a program is run on the MSP430 at LPM0 and LPM4, the hypothesis states that the MSP430 will be more affected by radiation when it is in LPM4 than it is in LPM0.

Three LPMs will be tested in this experiment: LPM0, LPM2, and LPM4. These LPMs were chosen as it ranged from the lowest LPM in LPM0 and the highest LPM (other

Table 1 Operation Modes

SCG1[1]	SCG0	OSCOFF[1]	CPUOFF[1]	Mode	CPU and Clocks Status[2]
0	0	0	0	Active	CPU, MCLK are active. ACLK is active. SMCLK optionally active (SMCLKOFF = 0). DCO is enabled if sources ACLK, MCLK, or SMCLK (SMCLKOFF = 0). DCO bias is enabled if DCO is enabled or DCO sources MCLK or SMCLK (SMCLKOFF = 0).
0	0	0	1	LPM0	CPU, MCLK are disabled. ACLK is active. SMCLK optionally active (SMCLKOFF = 0). DCO is enabled if sources ACLK or SMCLK (SMCLKOFF = 0). DCO bias is enabled if DCO is enabled or DCO sources MCLK or SMCLK (SMCLKOFF = 0).
0	1	0	1	LPM1	CPU, MCLK are disabled. ACLK is active. SMCLK optionally active (SMCLKOFF = 0). DCO is enabled if sources ACLK or SMCLK (SMCLKOFF = 0). DCO bias is enabled if DCO is enabled or DCO sources MCLK or SMCLK (SMCLKOFF = 0).
1	0	0	1	LPM2	CPU, MCLK are disabled. ACLK is active. SMCLK is disabled.
1	1	0	1	LPM3	CPU, MCLK are disabled. ACLK is active. SMCLK is disabled.
1	1	1	1	LPM4	CPU and all clocks are disabled.
1	1	1	1	LPM3.5	When PMMREGOFF = 1, regulator is disabled. No memory retention. In this mode, RTC operation is possible when configured properly. See the *RTC* module for further details.
1	1	1	1	LPM4.5	When PMMREGOFF = 1, regulator is disabled. No memory retention. In this mode, all clock sources are disabled; that is, no RTC operation is possible.

Table 3.1 Primary modules that are on/off for each power mode

than the LPMx.5) in LPM4. A more polarized selection of LPMs like this will highlight significant differences with the responses of the system towards radiation. The MSP430 program that will be executed for each of the three LPMs will be the same program, except for the slight variations in the code informing the MSP430 to enter the different LPMs. The differences will occur within the system when the system turns off certain modules whenever it enters a specific LPM. For instance, based on Table 1, when the MSP430 enters LPM3 the MCLK and SMCLK are disabled, however, in LPM1, though the MCLK is still disabled, the SMCLK is optionally active. The turning off and turning on of the modules happen automatically by the system except when the specific modules are requested in the program. This allows the MCU to reduce overall energy consumption of the system by turning off modules that are not in use at the moment.

3.7 EnergyTrace

The most vital piece to identifying the areas of effect in the MCU due to radiation relies on the EnergyTrace program. The EnergyTrace technology "is an energy-based code analysis tool that measures and displays the energy profile of an application and helps optimize it for ultra-low-power consumption" [18]. The technology measures the MCU's current consumption and in the advanced versions of the EnergyTrace program, it also traces the CPU states and peripheral states. The CPU states show which power mode (Active Mode or LPM) the MCU is in during the program. The peripheral states show the ON-OFF status of each peripheral module during the program.

The method TI uses to measure current consumption is explained in [18]:

> "In debuggers that support EnergyTrace software, a software-controlled DC-DC converter generates the target power supply. The time density of the DC/DC converter charge pulses equals the energy consumption of the target microcontroller. A built-in calibration circuit in the debug tool

defines the energy equivalent for a single charge pulse. Because the width of each charge pulse remains constant, the debugger can just count each charge pulse and then sum them over time to calculate an average current which leads to very accurate measurements. Using this approach, even the shortest device activity that consumes energy contributes to the overall recorded energy."

The scope of this paper does not cover how the EnergyTrace mechanics works; it only focuses on the application of the program. The objective of the test is to use EnergyTrace to track leakage current within the system and implement the program in a manner that will allow the user to locate which specific components within the MSP430 is being affected by radiation. Figures 3.7, 3.8, and 3.9 show the dashboard system of the EnergyTrace program along with examples of the real-time energy expenditures and module usage.

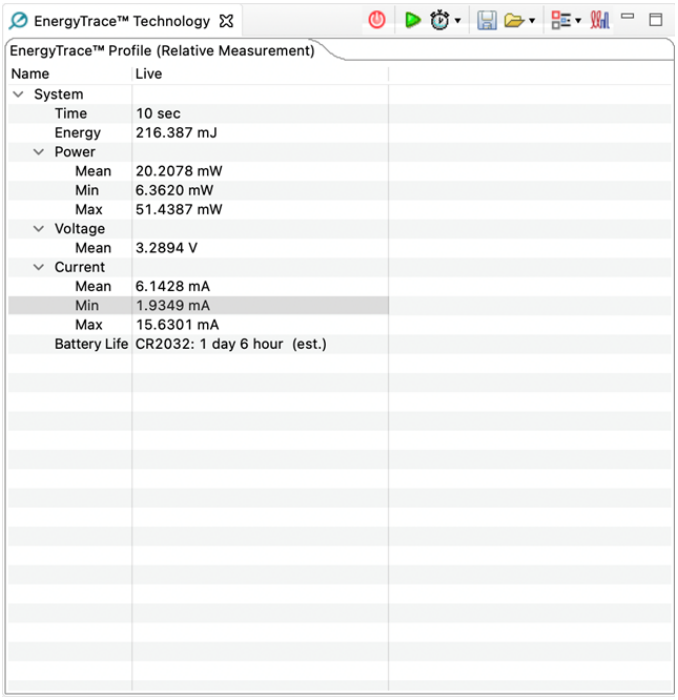


Figure 3.7 General energy and power profiles from EnergyTrace

Figures 3.7, 3.8, and 3.9 are power and energy measurements, CPU and peripheral states, and relative power and energy graphs of an example program that was run on the

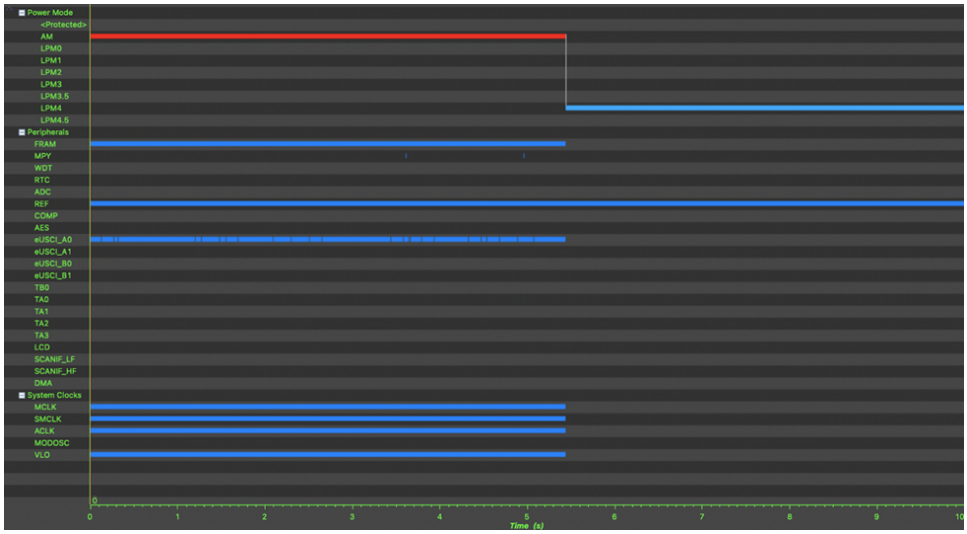
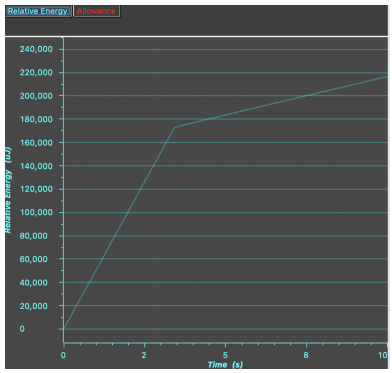
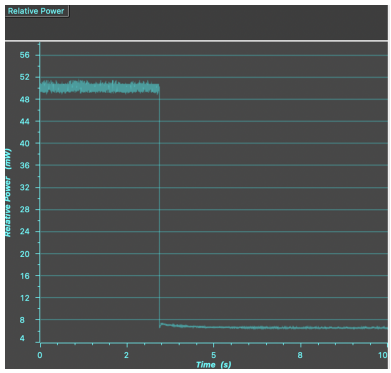


Figure 3.8 EnergyTrace displaying the real time states of the CPU and peripheral modules



(a) Relative energy usage graph of a program



(b) Relative energy usage graph of a program

Figure 3.9 EnergyTrace results

MSP430 to demonstrate the EnergyTrace technology. A similar EnergyTrace analysis will be used on the actual program that is run for the thesis experiment. The areas of interest from the general EnergyTrace data shown in Figure 3.7 are the mean value of the current consumption, the mean value of the power usage, and the total energy usage of the system. The EnergyTrace technology itself is outside of the MSP430 chip and is in a debugger which is found on the development board the MSP430 is also on. Since the debugger is protected with packaging and since it is on a different location from the MSP430 chip, the X-ray radiation will not affect the performance of the EnergyTrace technology.

3.8 Programming

To execute every step of the experiment process from using the MCU, to capturing clock signal and ADC data, and to data analysis, software must be used to accomplish these tasks.

3.8.1 MSP430 and Code Composer Studio

The MSP430 can be programmed using the Code Composer Studio (CCS) IDE. The reason CCS was used instead of other IDEs like the IAR Workbench is because CCS contains a simpler process to access EnergyTrace technology compared to the other IDEs. Within CCS, computer language C is used to program the MSP430 to activate the Clock System module, the ADC module, the UART module, and connect them to GPIO pins to communicate to the physical world and other cyber systems.

Through CCS, the clocks were set to certain frequencies (DCO = 16MHz and VLO = 10KHz) and outputted into specific GPIO pins so they can be recorded by the oscilloscope. The ADC module was also programmed through CCS to use certain clocks to govern the speed of the conversions and a GPIO pin was programmed to be the input to the ADC conversion channel. Vast amounts of C language and microcontroller processes had to be learned to program the MSP430. The final code for the MCU is run through CCS for the experiment [Appendix A].

3.8.2 Clock Signal, ADC Recording, and Data Analysis Programming

After the program for the MSP430 was written, the oscilloscope was programmed to record the clock signals. Because TID can anneal quickly, the experiment required a brief trial time so the TID effects can be captured accurately. Therefore, programming the

oscilloscope to save the data automatically was crucial in receiving accurate TID response results of the clock system module.

Using Jupyter notebook, which is a web-based interactive computing platform [19], and the computer language Python, the oscilloscope was programmed to capture at least 100 periods in the clock signal waveform and save them as a CSV file in a USB flash drive. A trigger was set in the oscilloscope to automatically save the data when the trigger was activated. So, when the MSP430's code activated the clock system module and outputted the signal to the oscilloscope, which then captured the clock signal automatically.

For the ADC module, Python was also used to program the Raspberry Pi and the external DAC component. The Raspberry Pi was programmed to communicate to the DAC to input a voltage ramp from 0V to 3.3V to the ADC module in the MSP430. A maximum voltage of 3.3V was chosen as the MSP430 typically functions with a voltage of 3.3V; occasionally a voltage of 5V is also utilized by the MSP430. The input voltage ramp values were saved in the Raspberry Pi. To record the converted output ADC values from the MSP430, Jupyter notebook, in conjunction with the RS-232, was used to send start tokens to begin the recording of converted values and saving them into a CSV file. Once all the converted values were stored, an end token was sent to stop the recording of converted values. Python was also used to analyze the clock and ADC data. For the clock, a Python program was written to calculate the frequencies of each signal. For the ADC module, a Python program was written to calculate the mean, variance, and standard deviation of the error between input voltage and output voltage.

CHAPTER 4

Results and Discussion

The data analysis is divided into three main categories: EnergyTrace results, clock frequency analysis, and ADC performance analysis. 3 DUTs were biased in LPM4, however one of the DUTs was off centered from the radiation beam and, therefore, did not receive any TID radiation. Due to this accident, one of the DUT's results will be disregarded; only 2 DUTs from LPM4 will be represented in the data analysis.

4.1 EnergyTrace Results

The EnergyTrace program measures the current, energy, and power usage statistics of the MCU. The analysis is initially separated into the different LPMs that each chip is in but is later combined to comparatively analyze which conditions led to the highest effects due to radiation. Though EnergyTrace displays current, energy, and power usage of the chip, only one characteristic (i.e., current usage) is required to understand the overall energy and power characteristics of the system. The reason for this is because energy is related to power, and power is related to current by (4.1), (4.2), and (4.3).

$$P = VI \tag{4.1}$$

$$Energy = Power * Time \tag{4.2}$$

Based on (4.2), (where P = power, V = voltage, and I = current) power is directly proportional to the current if V is constant or increases. The MSP430 uses a constant voltage of 3.3V to supply the chip to perform its functions; therefore, the voltage always stays around 3.3V regardless of any external factors such as radiation. Due to the constant voltage, the power is directly proportional to the current.

Furthermore, based on (4.2), energy is also directly proportional to the power if time is constant, and because the CCS program runs at a constant time of 30 seconds, the time parameter in (4.2) is constant. Since energy is proportional to power and power proportional to current, energy is also proportional to current based on equation (4.3).

$$Energy = V * I * T \tag{4.3}$$

Considering the correlations between the three recorded parameters of current, power, and energy, only current consumption data will be analyzed in this paper. To prove this concept experimentally, however, the power and energy data will also be included in the first LPM0 section of the data analysis but will be excluded in the other LPM tests. For the current analysis, the average current (in mA) used by the MSP430 is plotted against the apparent radiation dosage (in krad (SiO2)). The reason behind using the terminology "apparent radiation dosage" is because though the X-ray source emits a controlled radiation dosage of 5.44 krad (SiO2)/minute, TID anneals quickly and the exact dosage within the chip is not known. However, since each trial time is as brief as possible, this experiment assumes the dosage is not annealed away and is congruent with the accumulated dose deposited by the X-ray source. Also, because this experiment's scope does not measure the overall TID dose within the system, this assumption must be made to simplify data analysis. Nonetheless, annealing properties do seem to be evident within the data and some models could be derived in future testing.

4.1.1 LPM0

The average current used by the MSP430 to run the program when in the LPM0 state is plotted against the apparent TID dose point within the chip in Figure 4.1. The data set also includes a 24-hour anneal characterization of the system in conjunction with the irradiation data.

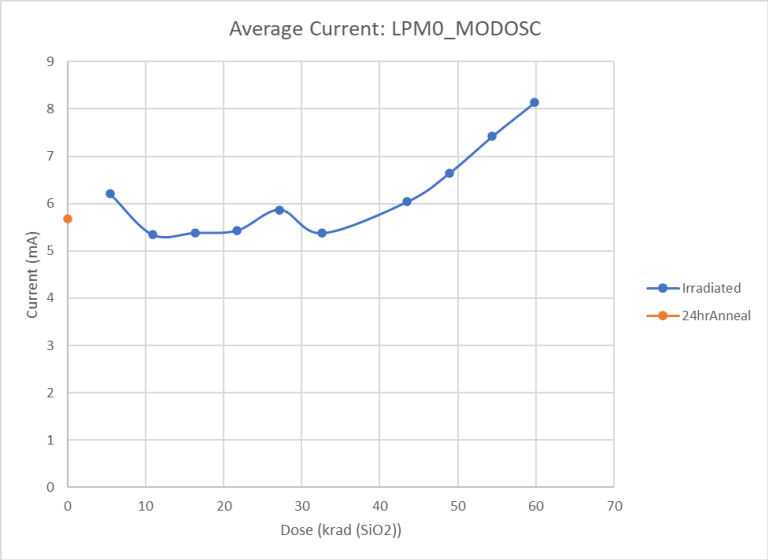


Figure 4.1 Average current usage vs. dose

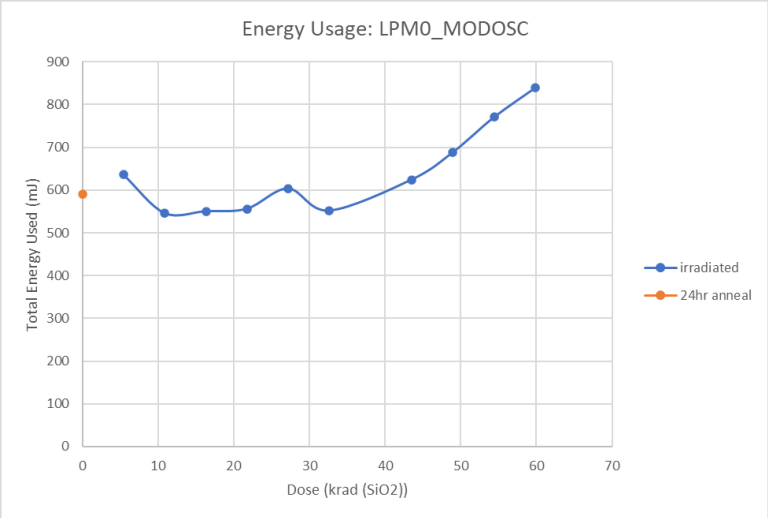


Figure 4.2 Total energy usage vs. dose

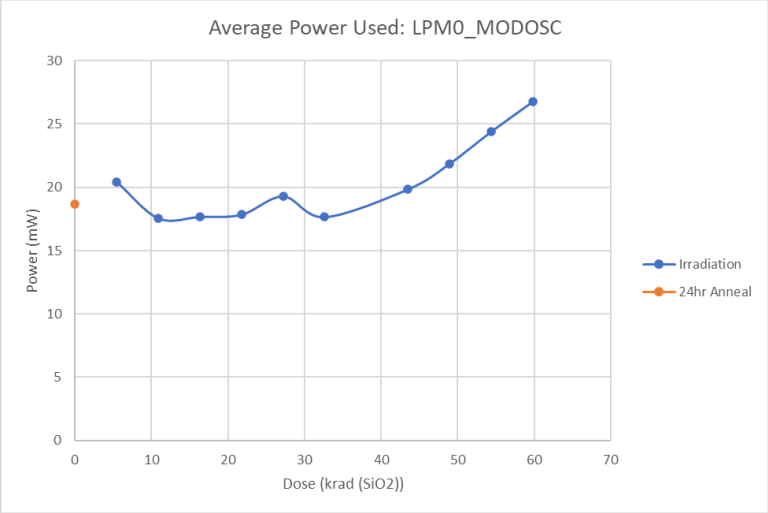


Figure 4.3 Average power usage vs. dose

The graphs found in Figure 4.1, 4.2, and 4.3 all carry almost an exact pattern and structure. The commonality between the graphs of current, energy, and power are proven by (4.3). Due to its commonality, the energy and power profiles will be excluded from the other LPM analysis.

More importantly, based on Figure 4.1, 4.2, and 4.3, the total current, energy, and power of the system increases as the MSP430 accumulates more dose. This pattern shows a correlation between the three recorded parameters and TID for LPM0. Because TID is known to affect the leakage current within the system, the fact that the current usage of the system has increased with the accumulated dose implies that the MSP430 is being affected by TID effects. When the accumulated dose point of 59.84krad (SiO2) was reached, the MSP430 was still functional and communicating to the PC; however, the test was stopped at this point as there was no desire to continue till the DUT was destroyed since based on the other tests the DUT stops working around 60krad (SiO2) - 70krad (SiO2).

4.1.2 LPM2

With LPM2, the pattern of the average current consumption by the MCU is not as clear as the pattern derived from the average current consumption of LPM0. The pattern from LPM0 was clearly illustrated as a positive correlation between dosage and average current consumption within the system, where the current consumption increases as the accumulated dose also increases. However, with LPM2 it is not as noticeable. Figure 4.4 shows the average current consumption by the MSP430 when it is the LPM2 state while the ADC clock is the MODOSC.

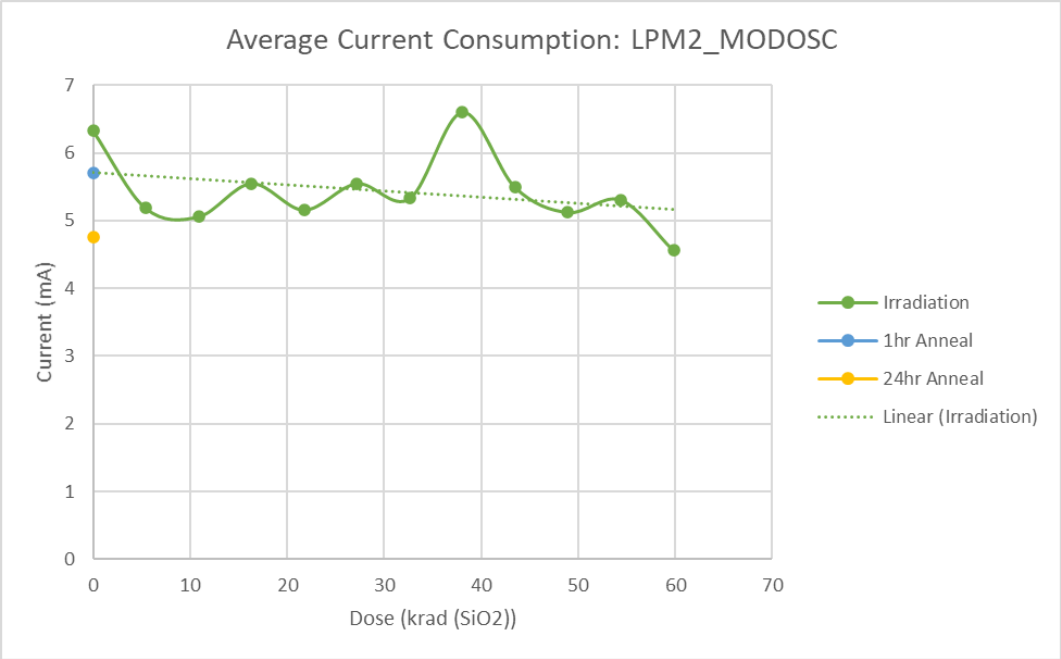


Figure 4.4 Current consumption of MSP430 during irradiation, and after 1-hr and 24-hr anneal periods

The current curve seems to oscillate in an increasing manner till it accumulates dosage of about 40krad (SiO2). After the 40krad (SiO2) mark, the average current consumption per dose point reduces significantly in a negative trend. However, at trial 7 of this test (which is at a dose point of 38.08krad (SiO2), the current consumption spikes abnormally where it is higher in current usage than the adjacent trials (trial 6 and trial 8). However, no changes were made in the program for trial 7 to warrant a current consumption increase. Trial

7's uptick in current usage might be attributed to radiation effects or merely an anomaly. Nonetheless, even with Trial 7's distinctly increased current consumption, the general trend (as illustrated by the dotted, linear trendline in Figure 4.4) of the current consumption is decreasing as the accumulated dose increases. This is contrary to the hypothesis that as TID increases, more current within the system is leaked, and therefore, more current is consumed by the system. The settings at which the MSP430 runs during LPM2 might attribute to the decreasing current consumption trend.

4.1.3 LPM4-0

The LPM4-0 test is the first test run during the entire experiment in Vanderbilt University. Because the highest accumulated dosage the MSP430 could endure was not known, the dose increments of this test were only 2.72 krad (SiO2) per trial compared to the 5.44 krad (SiO2) per trial increments of the other tests. Also, for this test the ADC clock source was not modified from the default MODOSC source. The other tests varied the ADC clock source from the MODOSC to the DCO (or vice versa) for a few trials of each test to record any ADC performance differences.

This test included the highest number of trails during irradiation: 22 trials were done until the MSP430 stopped communicating with the PC at 59.84 krad (SiO2). The 59.84 krad (SiO2) dose point gave a standard dose threshold for the MSP430, so, based on this test, the test with the MSP430 at LPM0 did not go past this dose limit. Figure 4.5 shows the plotted current consumption against the accumulated dose levels within the MSP430. The trend of the graph in Figure 4.5 shows that as the accumulated dose within the DUT increased, the average current consumption of the DUT also increased, implying the leakage current effects due to TID.

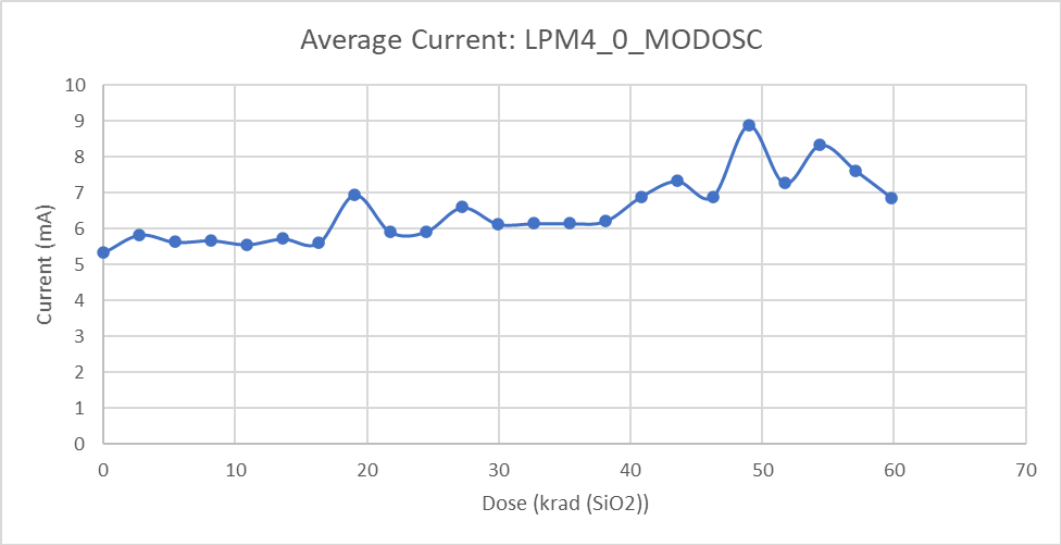


Figure 4.5 Current consumption of MSP430 in LPM4 during irradiation. The ADC clock source is the MODOSC for this test

4.1.4 LPM4-2

The LPM4-2 test differs from the LPM4-0 test only slightly: the overall program and the LPM are the same, except for the clock source of the ADC module. The variation of the ADC clock was done to see if there are any changes in the performance of the ADC based on the different clock sources. Additionally, the current consumption of LPM4-0 with the MODOSC clock source and LPM4-2 with the DCO clock source will be compared to propose any energy efficient modifications to be made when running a program in the MSP430. Figure 4.6 shows the average current consumption by the MSP430 when it is in LPM4 whilst using the DCO as the ADC's clock source.

Figure 4.6 also shows the current consumption of the MSP430 after a 2-hour and 24-hour anneal period. The trendline shown in Figure 4.6 indicates that the current consumption of the DUT decreased as the accumulated radiation increased within the chip. Like the results in LPM2, this trend is contrary to the earlier stated hypothesis and the trend of the other MSP430 in LPM4. The main difference between the two tests that ran at LPM4 is the ADC clock source; however, the reason for the opposite trends cannot be attributed to this minor

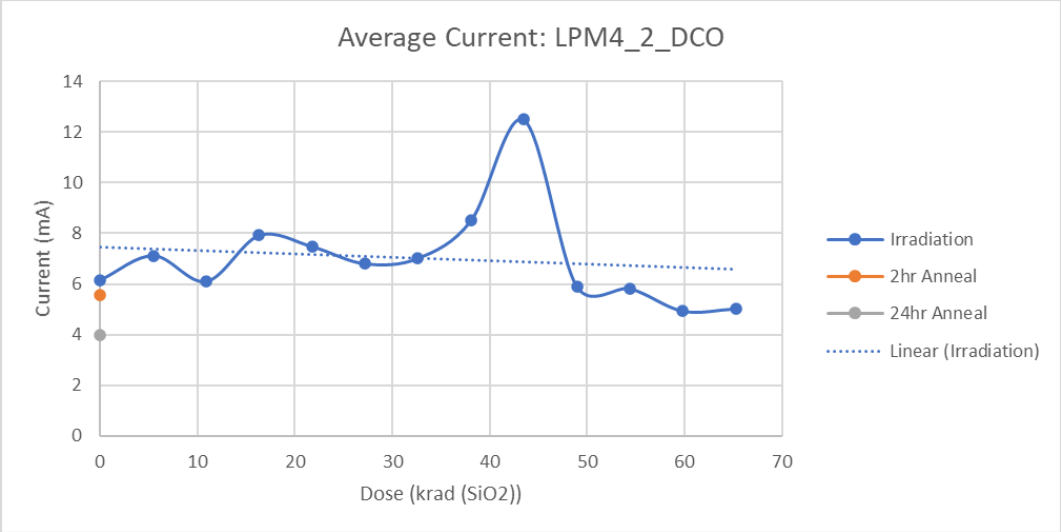


Figure 4.6 Current consumption of MSP430 in LPM4 during irradiation. The ADC clock source is the DCO for this test

modification alone. Other evidence must be present to suggest that point. The test for this DUT did not end in the destruction of the MSP430 completely, rather CCS was able to upload the code to the MSP430 (implying an active communication line), but the code was not able to execute on the MSP430 itself. Reasons attributed to this is not found or understood.

A current consumption comparison of the LPM4-0 and LPM4-2 tests is shown in Figure 4.7. Based on Figure 4.7, the MSP430 that uses the DCO as the ADC clock source usually expends more current than the MSP430 that uses the default MODOSC clock source. Because the DCO is not the default clock source option for the ADC module, the MSP430 might be using more energy to use the DCO when its innate build is for the MODOSC. This seems to be an insignificant reason, though, for the MSP430 in LPM4-2 to use more current than the LPM4-0's MSP. If a user wishes to have reduced energy when using the MSP430 for common use, then the ADC's clock source should be set to its default option of the MODOSC; however, if the user wishes to have the MSP430 not increase its current usage when it is irradiated, then the ADC clock should be set to the DCO option as that reduces the current consumption as more dosage is deposited into the system.

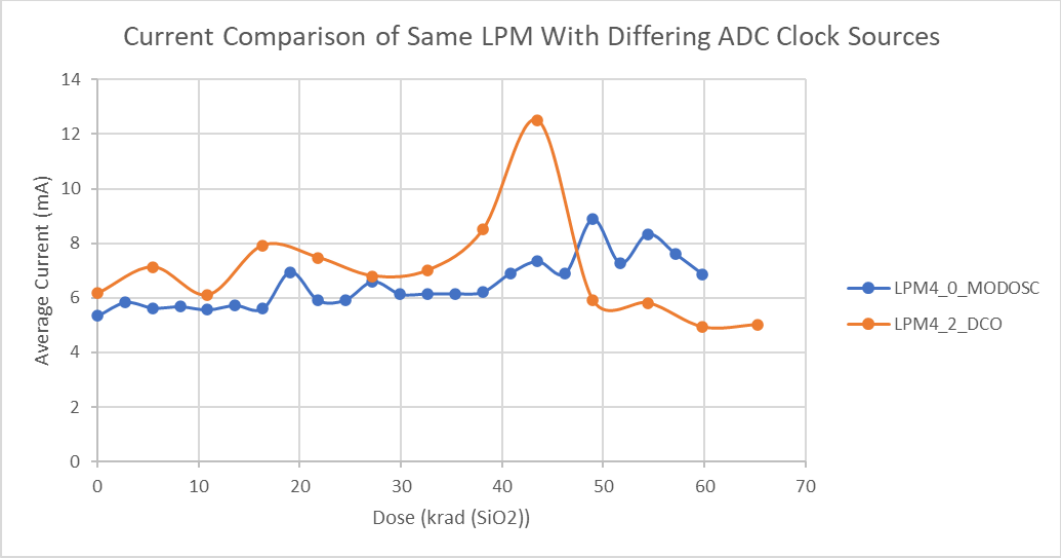


Figure 4.7 Current consumption of MSP430 in LPM4 but with different ADC clock sources

4.1.5 All LPM Current Analysis

The program that activated the clock system, ADC, and UART modules, was executed at varying LPMs and the current of the DUT at each LPM is recorded. A current consumption comparison of the DUT at each LPMs is shown in Figure 4.8. This is done so that the difference between the TID responses of each LPM can be depicted. Any increased susceptibility of a particular LPM to TID can be more easily detected through an overall comparison of data.

Based on Figure 4.8, the LPM with the most consistent correlation between accumulated dosage to current consumption is LPM0 as it has no major oscillations between high and low current values and is trending upward till the test was stopped. LPM4-0 also seems to have a consistent trend upward, but it has more oscillations than LPM0 that it does not allow a more certain conclusion that radiation is affecting like it does for LPM0. Nonetheless, radiation is affecting the DUT to consume more current, and therefore lead to more power loss within the system in LPM4-0. However, the LPM that consumes the most current is the MSP430 that was in LPM4-2 with the DCO as the ADC clock source. Because this is the only DUT that uses the DCO for the ADC module, that might be the reason why it requires

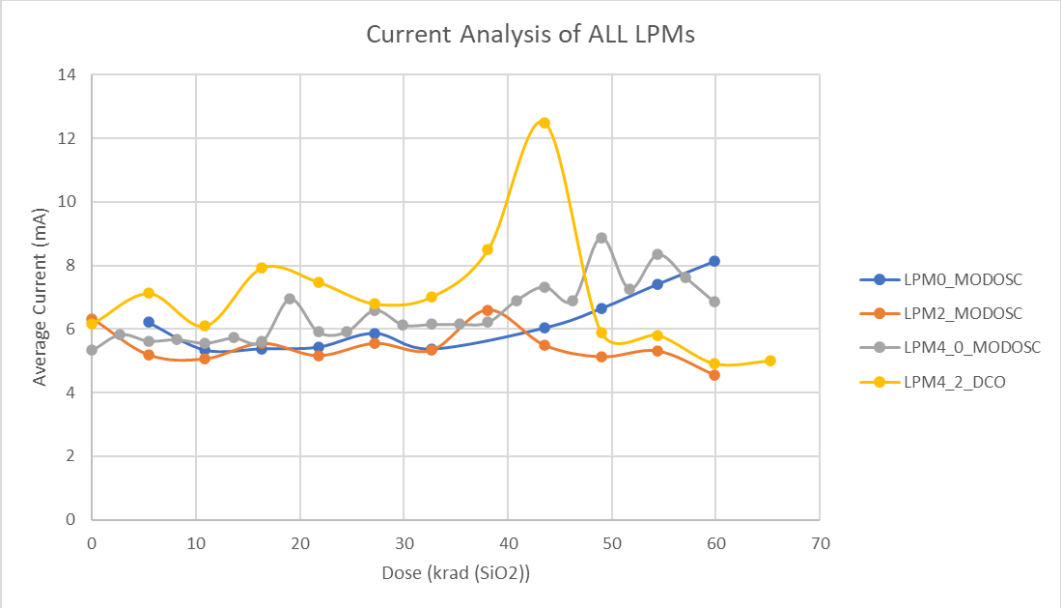


Figure 4.8 Current consumption of all DUTs in every LPM

the most current. However, both LPM4 models seem to use more current than the other two LPMs, which is counterintuitive to the design of the MCU. The higher the LPM equates the lower the energy usage of the MCU. But in this case, the MSP430s that were in the highest LPM has the highest current, energy, and power usage. Because of this counterintuitive result, increased current consumption must be attributed to TID. Other reasons like unique device-to-device characteristics could be considered, but the difference of the DUTs in LPM4 compared to the DUTs in LPM0 and LPM2 is significant. Also, since both the DUTs in the LPM4 states seem to be exhibiting a similar behavior with increased current consumption, the probability of the increased current consumption being due to device uniqueness can be disregarded.

The theory of higher LPMs making the DUT more susceptible to TID effects holds to be true thus far as per the comparative current consumption analysis in Figure 4.8. Once again, this is based on the concept of TID radiation allowing current to leak through the transistors in the system. So, the higher the current consumption of the system is from its non-radiated, control state, the more accumulated dose effects due to TID the system is

encountering. Still, the hypothesis has not been completely proven: the clock signals and ADC data will aid in justifying this observation.

Though that query is partially answered, the main objective of using EnergyTrace to identify which specific modules of the system are being affected has not been met. The assumption with EnergyTrace was that it will give energy and power profiles of each module in use, not just the energy and power profiles of the entire system. The assumption implied, for instance, that when the clocks and ADC are in use, EnergyTrace would measure the current consumption of the clocks and ADC individually, not collectively as it does now. EnergyTrace only gives the ON-OFF states of the modules, not the current, energy, and power profiles of those specific modules. This issue may have been resolved if we did not forget to save the EnergyTrace's energy and power data as CSV files. The data in a CSV format grants the ability to view the instantaneous energy and power of the system as the program is executed in the MSP430. However, even this data still only gives the overall energy and power usage of the system and not the individual components. Due to this, the valid conclusion is that EnergyTrace cannot be used to locate which parts of the MCU is being affected due to radiation. But it can provide a comprehensive indicator that the system is using more or less current than it should be, which can correlate to TID effects.

4.2 Clock Signal Analysis

The EnergyTrace program provided current, energy, and power statistics of the entire system. This section focuses on the clock system module and data analysis associated with it. The two clock modules that were analyzed were the DCO, which is connected to the MCLK clock signal, and the VLO, which is connected to the ACLK clock signal. For the remaining of the paper, MCLK analysis will represent the DCO analysis and the ACLK analysis will represent the VLO.

Figure 4.9 shows a small frame of the ACLK signal. It is a square wave that runs at

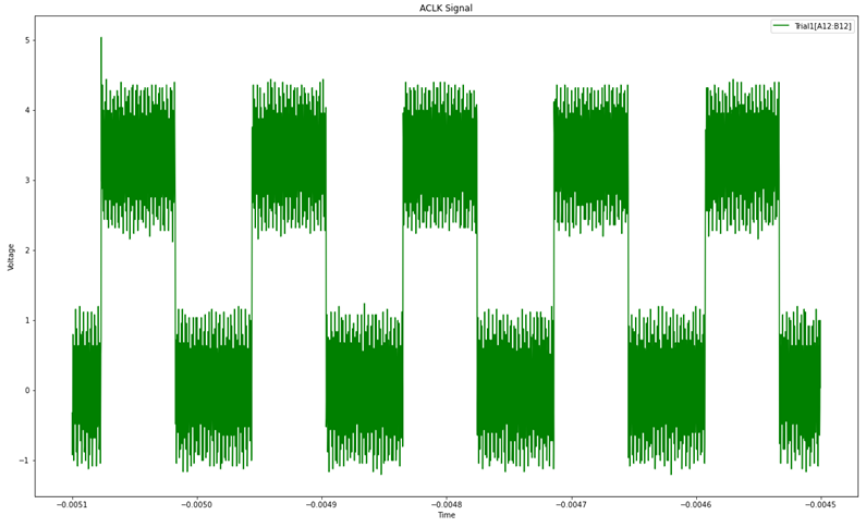


Figure 4.9 ACLK Signal

10 kHz. To calculate the frequency of the signal, a python program was written to activate a trigger at a rising edge and then count the number of triggers. The number of triggers divided by the time sample provided the frequency of the ACLK signal. The time sample was recorded by the oscilloscope when it saved the clock signal data into a CSV file. The same python program was used to calculate frequency of the MCLK signal. A portion of the MCLK signal is shown in Figure 4.10.

The theory is that the radiation is going to affect the clock frequencies: the frequencies of each clock source will be lower or higher than the programmed value. For example, if a clock is programmed to run at 8 MHz, it might run at 9 MHz, instead, due to radiation effects. Figure 4.11 shows the clock signal analysis for the ACLK and Figure 4.12 shows the clock signal analysis for the MCLK.

ACLK is programmed to run at a frequency of 10 kHz. Based on Figure 4.12, the non-radiated control frequency is at a dose of 0 krad (SiO2) and the frequency of each of the DUTs are around 8 kHz. Because all the MSP430s start at that frequency, 8 kHz not 10 kHz can be treated as the actual frequency the ACLK is supposed to run at. Indicated by

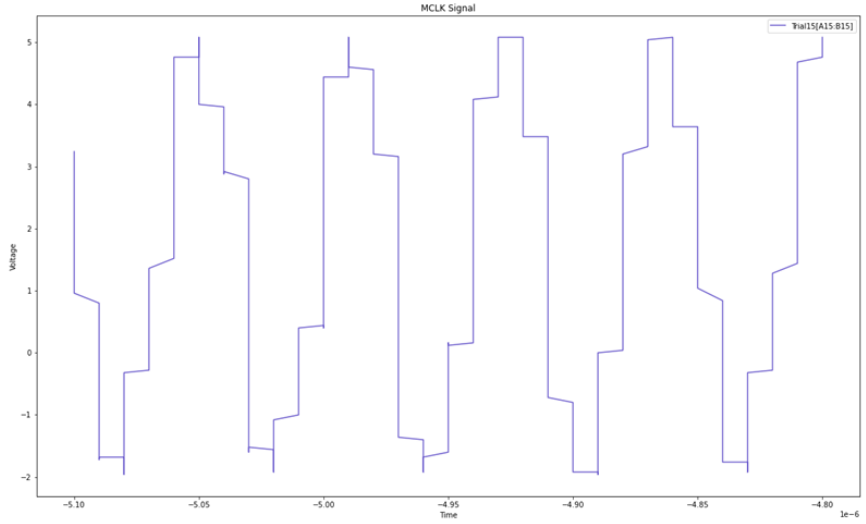


Figure 4.10 MCLK Signal

the graph, as accumulated dosage increases the frequency of the clocks changes regardless of the LPM during irradiation. All the clocks gradually increase in their frequency until the MSP430 reaches its threshold dosage before it fails around 55 krad (SiO2). The DUT which most deviates from the original frequency is the one in LPM4-0. The DUT holds a consistent frequency value till the threshold dosage is reached and it plummets close to 0 kHz before the MSP430 stops functioning. LPM2's DUT, however, has the highest increase in frequency when its collective dose levels increase.

The MCLK was programmed to be 16 MHz. The control non-radiated frequency of the DUT at LPM0, LPM2, and LPM4-2 is 16.2 MHz and the DUT at LPM4-0 is 16.4 MHz. The frequencies of LPM0 and LPM2 are hidden under the frequencies of LPM4-2, hence why it is not visible. Based on Figure 4.10, the MCLK has not been affected significantly due to radiation like the ACLK has. The only MSP430 that is affected is the one in LPM4-0. The reason for the changes in frequency in the LPM4-0 DUT might be attributed to device-by-device variation in the control frequency of the MSP430. Overall, the clock that is affected more due to radiation is the ACLK.

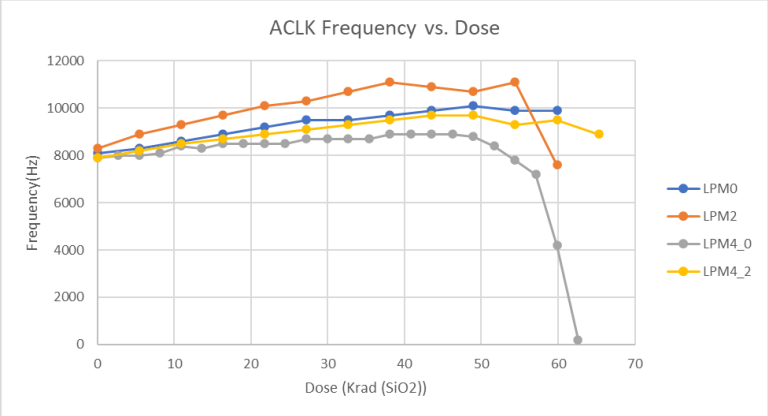


Figure 4.11 ACLK frequency comparison of LPMs

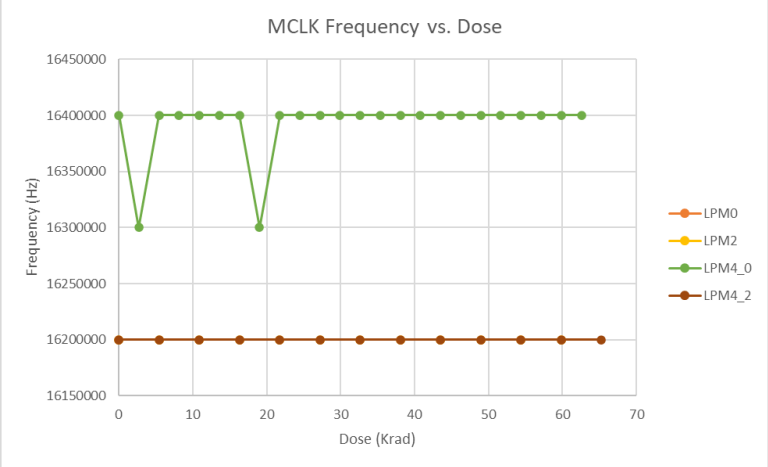


Figure 4.12 MCLK frequency comparison of LPMs

4.3 ADC Performance Analysis

To analyze the ADC, the initial method was to plot the input voltage values against the converted output voltage values. The input is the voltage values generated by the DAC through the Raspberry Pi and the output is the converted digital value of the corresponding input value. The reason for plotting in this manner is so that the input voltages are matched with the corresponding output voltages, and the plot should be a linear line with a slope of 1. If there are any deviations from the control non-radiated linear line to the input and output voltage values of the other trials, the other plotted lines will have a different slope which should clearly indicate any variations that could be attributed to TID effects.

However, when this method is used, the different trials are not easily distinguishable from each other, consequently making it difficult to analyze any differences within the ADC performance. Figure 4.13 shows an example of following this ADC analysis method. Although, this complication in distinguishing the different trials could simply be attributed to the fact that the ADC is not affected by the radiation, and therefore does not show any signs of deviation from the control.

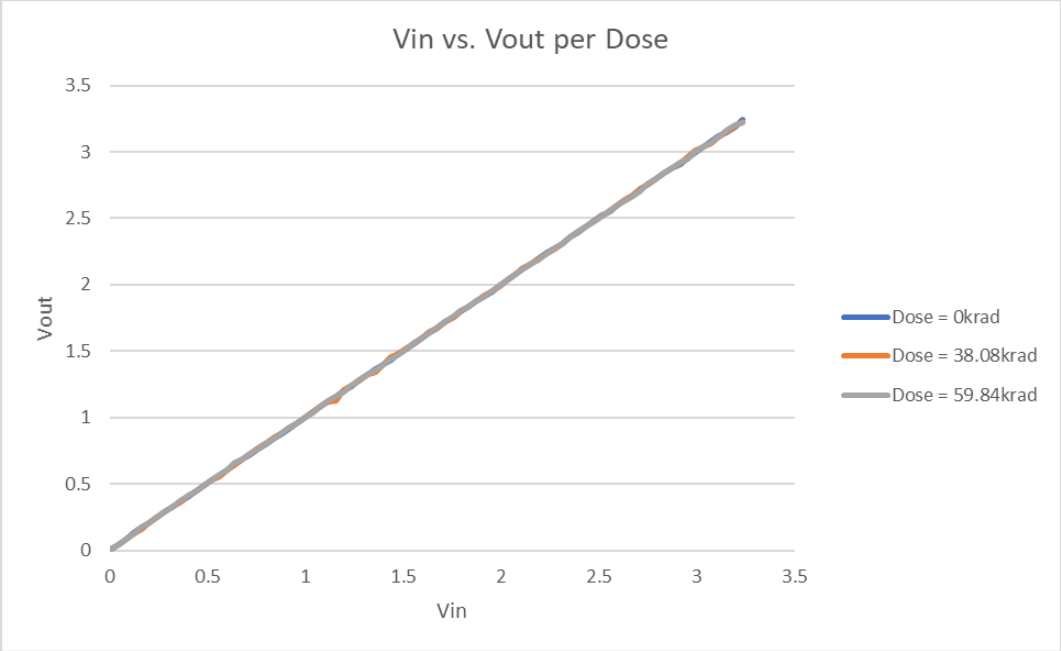
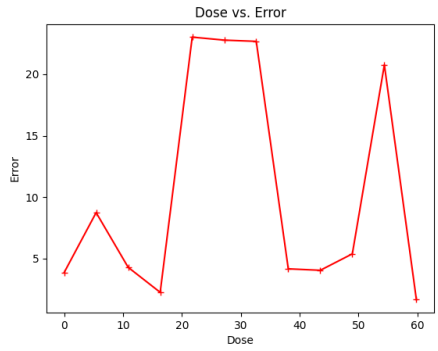


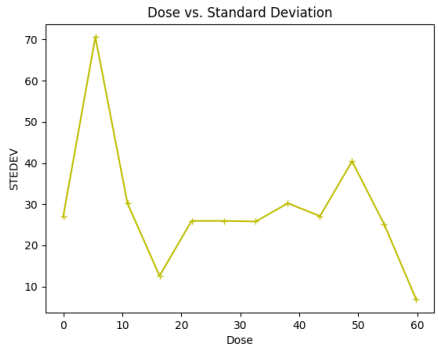
Figure 4.13 Initial ADC analysis method

Due to the complexity in analyzing the ADC this way, another method is executed to better understand the ADC module's response to radiation. The new method involves calculating the error percentage between the input voltage and output voltage values, which is then used to calculate the variance and standard deviation of each trial. However, since standard deviation is the square root of variance, the variance plot was excluded from the analysis. Figures 4.14 to 4.16 show the error percentages and standard deviation values of each LPM against the dose levels.

Based on Figures 4.14 to 4.16, the ADC does not show any trends of significant

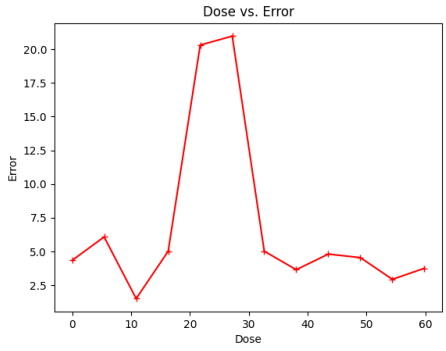


(a) LPM0 MODOSC Difference Error

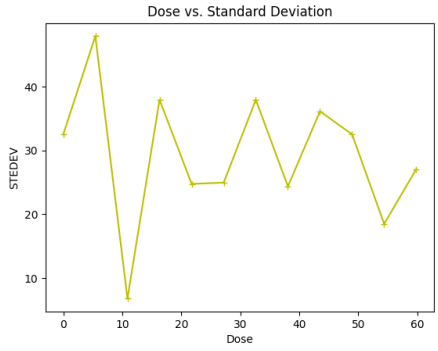


(b) LPM0 MODOSC Standard Deviation

Figure 4.14 LPM0 MODOSC ADC Analysis

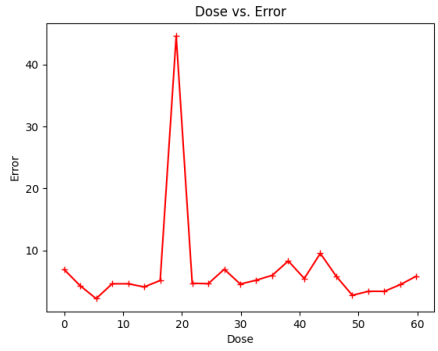


(a) LPM2 MODOSC Difference Error

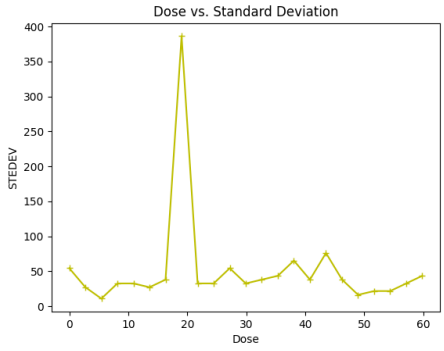


(b) LPM2 MODOSC Standard Deviation

Figure 4.15 LPM2 MODOSC ADC Analysis



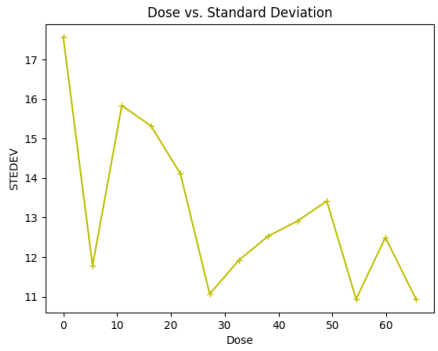
(a) LPM4 MODOSC Difference Error



(b) LPM4 MODOSC Standard Deviation

Figure 4.16 LPM4 MODOSC ADC Analysis

(a) LPM4-2 MODOSC Difference Error



(b) LPM4-2 MODOSC Standard Deviation

Figure 4.17 LPM4-2 MODOSC ADC Analysis

performance change due to the radiation. It consistently outputs the precise voltage value that is inputted into the DUT. There are high fluctuations in all three of the figures with no specific pattern, implying the immunity of the ADC to TID effects. No correlation between LPM and ADC performance due to radiation is found either. In Figure 4.17, however, the ADC of the DUT in LPM4-2 does have high fluctuations within the performance but with a general trend towards reduced error and standard deviation as the dose levels increase. The ADC module of this DUT uses the DCO clock source to govern the ADC conversions rather than the MODOSC source as the other tests did. The clock source of the ADC is the most significant difference between this test and the others and is the most likely reason for a general trend to exist in this test and not the others. The explanation, though, for the improvement of the ADC module as the dosage increases is unknown.

The most overlapping characteristic of all the ADC module analysis is the abnormally high error and standard deviation values. For an approximately normal data distribution, most of the data is within 1 standard deviation of the mean [21]. Anything above 3 standard deviations is an outlier. To better understand the data, a python program was written to convert the mean and standard deviation values based on z-scores. Based on this conversion, there were many outliers within the dataset. A pattern was found in the trials with the highest variations and is shown in Figure 4.18.

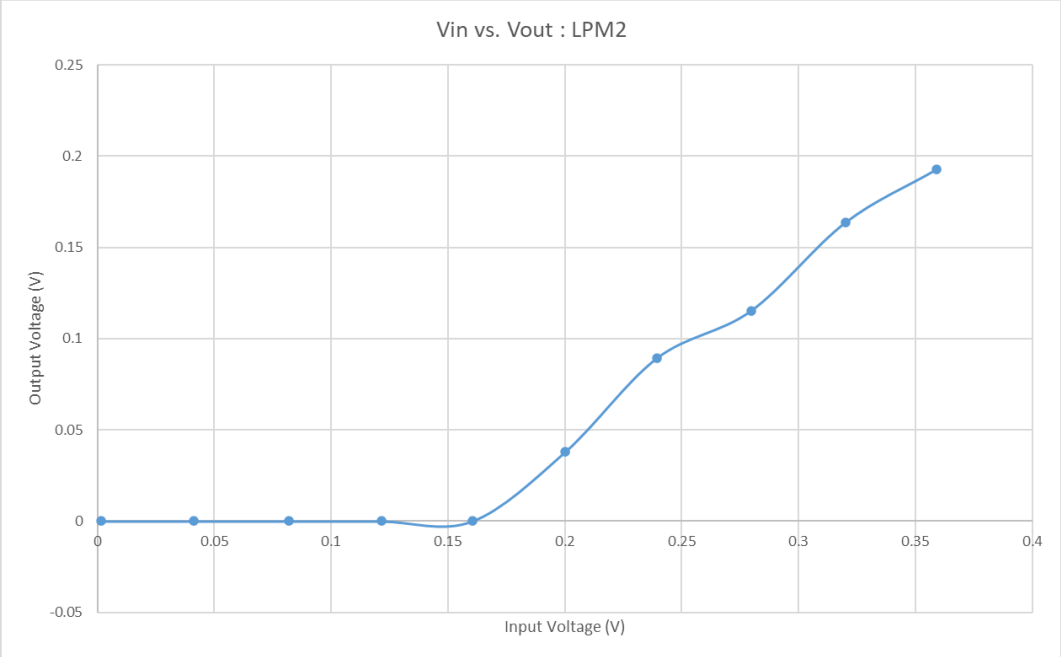


Figure 4.18 ADC input/output deficiencies

In every test, except the LPM4-0 test, the clock source of the ADC module was switched occasionally from the MODOSC to the DCO to test ADC performance based on different ADC clock sources. When the ADC clock source was switched back to the original source of the test, there was a period in which the ADC module read values of 0 for the input voltage instead of the correct voltage values supplied from the DAC. This anomalous behavior arose mostly when the ADC clock source was switched. To switch the clock source, the CCS code must be ended, modified, and debugged again to rerun entirely to implement the changes. During this break period, the ADC acts in this deficient manner. To understand whether the behavior is related to radiation or not, the entire experiment was rerun without radiation. The same number of trials, the same LPM, the same code, and the same switches of the ADC clock source were implemented. The only difference with the rerun setup compared to the original setup at Vanderbilt University is that the rerun setup did not use the Raspberry Pi and DAC to input specified voltages into the ADC module of the MSP430. Rather, it used a frequency generator to input an AC sine wave that ranged

between 0V and 3.3V.

Based on the data from the rerun experiment, the trials after the ADC clock source was switched did not show any anomalous behavior of outputting 0V when the ADC module is first activated in the code. In the rerun, the ADC module always outputs a non-zero value though the DUT went through a break period when the programmed had to be re-debugged. Due to the deficient behavior only being present under radiation, the faults occurring in the ADC module must be attributed to the TID effects. The radiation seems to be delaying the functionality of the ADC module when the MSP430 is restarted with a new program. The delay could be attributed to the increased power loss within the system due to the TID effects, and because of the power loss, the ADC module does not have sufficient power temporarily to properly convert analog input voltages to digital values. However, this explanation is only theoretical and not empirical; further studies need to be done to fully understand this defective mechanism within the ADC module.

Overall, the ADC module does not seem to be affected by the TID effects except for the occasional delay in ADC module's startup performance.

CHAPTER 5

Conclusion

The objective of the experiment is to identify specific modules of the SOC affected by total ionizing dose radiation using EnergyTrace technology and individual module performance characterization. During irradiation, in situ power and energy profiles of the DUT were captured using the EnergyTrace program, and clock signals' and ADC module's performance were also recorded. The strategy used in this experiment did not achieve the main objective desired. EnergyTrace results provided the power and energy data of the overall chip and not the individual components themselves. Furthermore, the individual component power and energy profiles could not be derived or extracted from the overall data values either. If TID affected the system, EnergyTrace alone did not provide data to explicitly locate which modules were affected; it only provided the effects of the total system through the tracking of overall leakage current.

However, when EnergyTrace is used in conjunction with the performance characterization of the modules like the clock system and ADC, a less refined approach to identifying radiation effects in individual components is still present. When TID radiation affects the system, it will cause the system to leak current which can be tracked by EnergyTrace; when leakage current is evident in the system through EnergyTrace, individual modules within the system can be tested to analyze any variations in its performance to locate the parts in the system being affected. The data analyzed in this experiment can function as an example of this strategy. Based on Figure 4.8, LPM4-0 shows the most significant and consistent increase in current consumption as dosage increases, implying that LPM4 is the most susceptible to

TID effects. Next, to identify which specific modules are being affected by the radiation, the clock signal data of LPM4-0 can be studied. Based on Figure 4.11, the frequency of the ACLK when in LPM4-0 is affected the most and similarly, based on Figure 4.12, the frequency of the MCLK of the LPM4-0 DUT is also being affected the most. Transitioning to the ADC data, the LPM4-0's ADC module performance is not drastically affected by the increase in total dose, as shown in Figure 4.14 to 4.16. When used collectively, this process concludes that the component that is being the most affected by the TID radiation must be the ACLK module when used in LPM4.

Using this integrated strategy, the individual component responses due to TID radiation can be studied, but this cannot be accomplished with EnergyTrace data alone. The subordinate test concept of the experiment involving the increased susceptibility of radiation effects based on LPM is also answered. Most significantly expressed through Figure 4.8, when the MSP430 is in LPM4 it is most susceptible to radiation effects.

5.1 Future Works

One of the extensions of this experiment that can be attempted in the future is to develop mitigation strategies within the chip to avoid TID effects. For example, based on the data crystal oscillator clock sources are more susceptible to TID effects than a digital oscillator. Therefore, if the system is using a crystal oscillator clock source like the VLO and the performance characteristics of the clock changes due to radiation, a mitigation technique to switch the clock source from the crystal oscillator to the digital oscillator can be implemented to maintain the system's performance.

Another work that can be investigated is the TID annealing effects using this data set. EnergyTrace, clock, and ADC module data were recorded after at a 1-to-24-hour period after radiation to capture any annealing effects within the system. Glimpses of the annealing effects can be seen in the graphs of the EnergyTrace current consumption figures in the

EnergyTrace Results section. A general trend of lowered current consumption is evident as the annealing time increases. However, this objective was out of the scope of this thesis paper and therefore, was not extensively studied.

Furthermore, though the main objective is not accomplished through EnergyTrace alone, an implementation of EnergyTrace in other embedded systems can prove helpful when identifying TID effects in the system. A future study can investigate how to record current, energy, and power measurements within all systems (not just TI's MSP430) to track leakage current and additional characteristics.

References

[1] J.-C. Boudenot "Radiation Space Environment." Radiation Effects on Embedded Systems, pp. 1-10, doi:10.1007/978-1-4020-5646-8_2.

[2] Anghel, L., et al. "Multi-Level Fault Effects Evaluation." Radiation Effects on Embedded Systems, pp. 69–88., doi:10.1007/978-1-4020-5646-8_4.

[3] F. Restrepo-Calle, S. Cuenca-Asensi, M. A. Aguirre, F. R. Palomo, H. Guzmán-Miranda and A. Martínez-Álvarez, "On the definition of real conditions for a fault injection experiment on embedded systems," 2011 12th European Conference on Radiation and Its Effects on Components and Systems, Seville, Spain, 2011, pp. 497-500, doi: 10.1109/RADECS.2011.6131425.

[4] Schrimpf, R. D. "Radiation Effects in Microelectronics." Radiation Effects on Embedded Systems, pp. 11–29., doi:10.1007/978-1-4020-5646-8_2.

[5] Di Mascio, S., Menicucci, A., Furano, G., Szewczyk, T., Campajola, L., Di Capua, F., . . . Ottavi, M. (2018). Towards defining a simplified procedure for cots system-on-chip tid testing. Nuclear Engineering and Technology, 50(8), 1298-1305. doi:10.1016/j.net.2018.07.010

[6] D. M. Fleetwood and H. A. Eisen, "Total-dose radiation hardness assurance," in IEEE Transactions on Nuclear Science, vol. 50, no. 3, pp. 552-564, June 2003, doi: 10.1109/TNS.2003.813130.

[7] G. Furano et al., "A novel method for SEE validation of complex SoCs using Low-Energy Proton beams," 2016 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Storrs, CT, USA, 2016, pp. 131-134, doi: 10.1109/DFT.2016.7684084.

[8] C. Oliveira et al., "Validation of an on-chip watchdog for embedded systems exposed to radiation and conducted EMI," 2014 IX Southern Conference on Programmable Logic (SPL), Buenos Aires, Argentina, 2014, pp. 1-6, doi: 10.1109/SPL.2014.7002212.

[9] Valvano And Yerraballi, J. V. R. Y. (2015). Introduction to Embedded Systems. Embedded Systems - Shape the World. http://users.ece.utexas.edu/%7Evalvano /Volume1/E-Book/C1_EmbeddedSystemsShapeTheWorld.html

[10] Loveless, Daniel. "Introduction to Radiation Effects in Microelectronics: What are radiation effects, where are radiation effects, terminology". ENEE 4999, Introduction to Radiation Effects, University of Tennessee at Chattanooga, Chattanooga, TN, January 28, 2021.

[11] Total Ionizing Dose (TID) Basics. (2021, October 1). TI Training. https://training.ti.com/total-ionizing-dose-tid-basics?context=1138916-1127560

[12] P. Luo, R. Ling, X. Zhou, P. Jiang and Y. Wu, "A Total Ionizing Dose Detecting Circuit Based on Off-state Leakage Current of NLDMOS in Power IC," 2019 31st International Symposium on Power Semiconductor Devices and ICs (ISPSD), 2019, pp. 119-122, doi: 10.1109/ISPSD.2019.8757653.

[13] H. Quinn, "Challenges in Testing Complex Systems," in IEEE Transactions on Nuclear Science, vol. 61, no. 2, pp. 766-786, April 2014,

doi: 10.1109/TNS.2014.2302432.

[14] Choosing the Right Oscillator for Your Microcontroller - Technical Articles. (2016, June 2). All About Circuits. https://www.allaboutcircuits.com/technical-articles/choosing-the-right-oscillator-for-your-microcontroller/:%7E:text=The%20CPU%2C%20the%20memory%20bus, conversion%2C%20and%20so%20much%20more.

[15] Texas Instruments. (2020). MSP430FR6989 Microcontroller: MSP430FR58xx, MSP430FR59xx, and MSP430FR6xx Family User's Guide (Rev. P). Dallas, Texas: Texas Instruments Incorporated.

[16] Texas Instruments. (2018). MSP430FR6989 Microcontroller: MSP430FR698x(1), MSP430FR598x(1) Mixed-Signal Microcontrollers datasheet (Rev. D). Dallas, Texas: Texas Instruments Incorporated.

[17] Why Do We Need Analog To Digital Converters? Tunable Phase Shifter. (2017, November 6). ADSANTEC. https://www.adsantec.com/why-do-we-need-analog-digital-converters/:%7E:text=An%20analog%20to%20digital%20converter,and%20reliable%20by%20minimizing%20errors.

[18] Texas Instruments. (2020, September 21). EnergyTrace: EnergyTrace Technology. https://www.ti.com/tool/ENERGYTRACE

[19] Project Jupyter. (2022). Home. https://jupyter.org/

[20] Digital Programmable Oscillator Is Smaller, Sturdier and More Versatile than Crystal Oscillators — Analog Devices. (2022). Analog Devices: Ahead of What's Possible. https://www.analog.com/en/technical-

articles/digital-programmable-oscillator-is-smaller-sturdier-and-more-versatile.html

[21] Muljadi, P. (2020, March 22). Standard deviation. Academia. https://www.academia.edu/42288574/Standard_deviation:%7E:text=%5B15%5D%20If%20a%20data%20distribution,within%20three%20standard%20deviations%20(%CE%BC

[22] EnergyTrace. (2020, June 29). Texas Instruments. https://software-dl.ti.com/ccs/esd/documents/xdsdebugprobes/emu$_e$*nergytrace.html* : %7*E* : *text* = *EnergyTrace*%*E*2%84%*A*2%20technology%20is%20a,ultra%2Dlow%2Dpower%20consumption.

Appendix A

```
#include <msp430.h>
#define samples 943
void clocks(void);
void adc();
void uart_tx_number(long number);
void uart_tx(char data);
void uart_tx_string(char *string);
int i = 0;
int result[samples];
int main(void)
{
        WDTCTL = WDTPW | WDTHOLD;     // stop watchdog timer
       P1OUT &= ~BIT0;
       // Clear P1.0 output latch for a defined power-on state
       P1DIR |= BIT0;
       // Set P1.0 to output direction
//==================Clocks GPIO Initialization====================
       P4DIR |= BIT1;
       P4SEL0 |= BIT1;                                 // Output ACLK
       P4SEL1 |= BIT1;
```

```
    P10DIR |= BIT2;
    P10SEL1 |= BIT2;                          // Output SMCLK
    P10SEL0 |= BIT2;

    P4DIR |= BIT0;
    P4SEL1 |= BIT0;                           // Output MCLK
    P4SEL0 |= BIT0;
//==================USCI_A0 UART operation====================
    P2SEL0 |= BIT0 | BIT1;     // USCI_A0 UART operation
    P2SEL1 &= ~(BIT0 | BIT1);
//==================ADC input GPIO pin configuration=========
    P1SEL1 |= BIT3;                 // Configure P1.3 for ADC
    P1SEL0 |= BIT3;
    // Disable the GPIO power-on default
    // high-impedance mode to activate
    // previously configured port settings
    PM5CTL0 &= ~LOCKLPM5;
//==================Calling Modules=========================
    adc();
    __bis_SR_register(LPM4_bits | GIE);
    // ----------------------------Change LPM here

//      clocks();         // Call Clock Function
    uart_tx('$');
    uart_tx('\n');
//      long period = 36956522;
```

```
//        while(period > 0){
//            period -= 1;
//        }
//==============Entering Low Power Mode============
        __bis_SR_register(LPM4_bits);  // Enter Low Power Mode 4
        // ------------------------------Change LPM here


    return 0;
}
void clocks(void){
    // Clock System Setup
    // Configure one FRAM waitstate as required by
    // the device datasheet for MCLK
    // operation beyond 8MHz _before_ configuring
    // the clock system.
        FRCTL0 = FRCTLPW | NWAITS_1;
        CSCTL0_H = CSKEY >> 8;
        // Unlock clock registers
        CSCTL1 = DCOFSEL_4 | DCORSEL;       // Set DCO to 16MHz
        CSCTL2 = SELA__VLOCLK | SELS__DCOCLK | SELM__DCOCLK;
        // Set SMCLK = MCLK = DCO,
        CSCTL3 = DIVA__1 | DIVS__1 | DIVM__1;
        // Set all divider
        CSCTL0_H = 0;
        // Lock CS registers
}
```

```
void adc(){
        PM5CTL0 &= ~LOCKLPM5;
        // By default, REFMSTR=1 => REFCTL is used to configure
                                    //   the internal reference
        // Configure one FRAM waitstate as required by
        // the device datasheet for MCLK
        // operation beyond 8MHz _before_ configuring
        // the clock system.
        FRCTL0 = FRCTLPW | NWAITS_1;
        CSCTL0_H = CSKEY >> 8;              // Unlock clock registers
        CSCTL1 = DCOFSEL_4 | DCORSEL;       // Set DCO to 16MHz
        CSCTL2 = SELA__VLOCLK | SELS__DCOCLK | SELM_DCOCLK;
        // Set SMCLK = MCLK = DCO,
        CSCTL3 = DIVA__1 | DIVS__1 | DIVM__1;
        // Set all divider
        CSCTL0_H = 0;
        // Lock CS registers
        while(REFCTL0 & REFGENBUSY);
        // If ref generator busy, WAIT
        REFCTL0 |= REFVSEL_0 | REFON;
        // Select internal ref = 1.2V
        ////////////////////// FOR ADC12 REF VOLTAGE
        // Internal Reference ON
        // Configure ADC12
        ADC12CTL0 = ADC12SHT0_2 | ADC12ON;
```

```
////////////////// ADC CONFIGURATION
ADC12CTL1 = ADC12SHP | ADC12SSEL_0;
//Choose ADC Clock source: ADC12SSEL_0 = MODOSC;
//ADC12SSEL_2 = MCLK (= DCO); sampling timer
ADC12CTL2 |= ADC12RES_2;
// 12-bit conversion results
ADC12IER0 |= ADC12IE3 | ADC12IE0;
// Enable ADC conv complete interrupt
ADC12MCTL0 |= ADC12INCH_3 | ADC12VRSEL_1;
// A3 ADC input select; Vref=1.2V
while(!(REFCTL0 & REFGENRDY));
// Wait for reference generator to settle
ADC12CTL0 |= ADC12ENC | ADC12SC;
// Sampling and conversion start
P1OUT |= BIT0; // turn on P1.0
// Configure USCI_A0 for UART mode
 UCA0CTLW0 = UCSWRST;         // Put eUSCI in reset
 UCA0CTLW0 |= UCSSEL__SMCLK;  // CLK = SMCLK
 // Baud Rate calculation
 // 16000000/(16*9600) = 52.083
 // Fractional portion = 0.083
 // User's Guide Table 21-4: UCBRSx = 0x04
 // UCBRFx = int ( (52.083-52)*16) = 1
 UCA0BR0 = 104;               // 16000000/16/9600
 UCA0BR1 = 0x00;
 UCA0MCTLW |= UCOS16 | UCBRF_2 | 0xD600;
```

```
        UCA0CTLW0 &= ~UCSWRST;          // Initialize eUSCI
        //UCA0IE |= UCRXIE;        // Enable USCI_A0 RX interrupt
        //__bis_SR_register(LPM3_bits + GIE);
        // LPM0, ADC10_ISR will force exit
        //__no_operation();     // For debugger
}
#if defined(__TI_COMPILER_VERSION__) ||
 defined(__IAR_SYSTEMS_ICC__)
#pragma vector = ADC12_VECTOR
__interrupt void ADC12_ISR(void)
#elif defined(__GNUC__)
void __attribute__ ((interrupt(ADC12_VECTOR))) ADC12_ISR (void)
#else
#error Compiler not supported!
#endif
{
 switch (__even_in_range(ADC12IV, ADC12IV_ADC12RDYIFG))
 {
  case ADC12IV_NONE:     break;     // Vector 0: No interrupt
  case ADC12IV_ADC12OVIFG: break;
  // Vector 2: ADC12MEMx Overflow
  case ADC12IV_ADC12TOVIFG: break;
  // Vector 4: Conversion time overflow
  case ADC12IV_ADC12HIIFG: break;     // Vector 6: ADC12BHI
  case ADC12IV_ADC12LOIFG: break;     // Vector 8: ADC12BLO
  case ADC12IV_ADC12INIFG: break;     // Vector 10: ADC12BIN
```

```
case ADC12IV_ADC12IFG0:
// Vector 12: ADC12MEM0 Interrupt
    P1OUT &= ~BIT0; // turn off P1.0
    if(i < samples){
        result[i] = ADC12MEM0;
        i++;
        ADC12CTL0 |= ADC12ENC | ADC12SC;
        // Sampling and conversion start
        P1OUT |= BIT0; // turn on P1.0
    }
    else{
        ADC12CTL0 &= ~ADC12ENC;
        ADC12IER0 &= ~ADC12IE0;
        P1OUT &= ~BIT0; // turn off P1.0
        // tx results
        i = 0;
        UCA0IE |= UCTXIE;
        uart_tx('[');
        uart_tx('\n');
        for(i = 0; i < samples; i++){
            uart_tx_number(result[i]);
            uart_tx('\n');
        }
        uart_tx(']');
        uart_tx('\n');
        __bic_SR_register_on_exit(LPM4_bits | GIE);
```

```
            // ------------------------------Change LPM here
             /*
            i = 0;
            ADC12CTL0 |= ADC12ENC;
            ADC12IER0 |= ADC12IE0;
            */
        }
        break;                  // Clear CPUOFF bit from 0(SR)
  default: break;
 }
}
void uart_tx(char data){          // writes a byte to serial comms
    while (!(UCA0IFG & UCTXIFG));   // USCI_A0 TX buffer ready?
    UCA0TXBUF = data;
}
void uart_tx_string(char *string){
// writes a string to serial comms
    int length = strlen(string);
    while(length>0)
    {
        uart_tx(*string);
        length--;
        *string++;
    }
}
void uart_tx_number(long number){
```

```
    int digits = 0;
    char character[10] = {'0','1','2','3','4',
                          '5','6','7','8','9'};
    if(number == 0){
        uart_tx('0');    // if number = 0, go ahead and send 0
    }
    long reverse = 0;
    while(number > 0){
        reverse = reverse * 10 + (number % 10);
        number /= 10;
        digits++;
    }
    while(digits > 0){
        uart_tx(character[reverse % 10]);
        reverse /= 10;
        digits--;
    }
}
```

Appendix B

LPM4_0	LPM4_2	LPM2	LPM0
Dose Point (krad (SiO2))	Dose Point (krad (SiO2))	Dose Point (krad (SiO2))	Dose Point (krad (SiO2))
0	0	0	0
2.72	5.44	5.44	5.44
5.44	10.88	10.88	10.88
8.16	16.32	16.32	16.32
10.88	21.76	21.76	21.76
13.6	27.2	27.2	27.2
16.32	32.64	32.64	32.64
19.04	38.08	38.08	38.08
21.76	43.52	43.52	43.52
24.48	48.96	48.96	48.96
27.2	54.4	54.4	54.4
29.92	59.84	59.84	59.84
32.64	65.28	65.28	
35.36	70.72		
38.08			
40.8			
43.52			
46.24			
48.96			
51.68			
54.4			
57.12			
59.84			
62.56			

Table 5.1 Dose points of each DUT