DATA COLLECTION AND ANALYSIS OF READ-IN INTEGRATED CIRCUITS DESIGNED TO DRIVE ARRAYS OF INFRARED LIGHT EMITTING DIODES USING A SCALABLE AND MODULAR TESTING PLATFORM FOR INFRARED SCENE PROJECTORS

by

Miguel A. Hernandez

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

Spring 2016

DATA COLLECTION AND ANALYSIS OF READ-IN INTEGRATED CIRCUITS DESIGNED TO DRIVE ARRAYS OF INFRARED LIGHT EMITTING DIODES USING A SCALABLE AND MODULAR TESTING PLATFORM FOR INFRARED SCENE PROJECTORS

by

Miguel A. Hernandez

Approved: _____
Fouad Kiamilev, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____
Kenneth E. Barner, Ph.D.
Chair of the Department of Electrical and Computer Engineering

Approved: _____
Babatunde A. Ogunnaike, Ph.D.
Dean of the College of Engineering

Approved: _____
Ann L. Ardis, Ph.D.
Senior Vice Provost for Graduate and Professional Education

ACKNOWLEDGMENTS

I would like to thank a wonderful group of people that with their dedication and effort make this research possible. First I would like to thanks Dr. Fouad Kiamilev for giving a chance to form part of the CVORG team. It is a pleasure to work with him because he is the type of leader that helps you grow professionally and personally.

I would also like to thank many of the CVORG members who in some way or another assisted me in the process of my research. I would like to start with the people who helped me during the wafer testing phase of my work, Peyman Barackhshan, Garret Ejzak, Jon Dickason and Josh Marks. I would also like to thank Nick Waite for helping me with the software development part of my work. I would also like to thank all of those who made my work place an enjoyable place to be: Hamzah Ahmed, Jacob Benedict, Kassem Nahba, Rodney McGee, Tyler Browning, in addition to the people previously mention. I have learned many things from them because everyone provides a unique skillset that propels the research forward.

Last but not least I would like to thank my family who has been there since the very beginning. My parents have taught me many values that made me the person I am today; I am very grateful for having such loving parents.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The read-in integrated circuit (RIIC) is an integrated circuit that drives an array of infrared emitters inside of an infrared scene projector (IRSP) system. We have designed different RIICs for four future IRSP systems that are being built by our research group. This paper describes a single scalable testing platform (STP) capable of testing all of our RIICs. This approach reduces the design time and risk associated with RIIC testing. On the hardware side, our platform consists of several custom printed circuit boards. On the software side, our platform consists of a single code base. After gathering data from a RIIC using the testing platform, the data is analyzed to determine the efficiency of such RIIC, and collect statistical data to aid the development of future RIICs.

Chapter 1

INTRODUCTION

1.1 Infrared Scene Projector Background

The infrared (IR) light spectrum is not within the range of visible light that humans can perceive, as a result IR sensors have been developed to detect the wavelengths of IR light. These type of sensors are used for many different applications in a variety of industries. In the field of climatology, they are used to collect information about global warming and earth temperatures. In the military IR sensors are used as homing devices to guide a missile to a target. And televisions use the sensors to receive commands from a remote control. There are several other uses for IR sensors that involve receiving light from the environment around it. However, there aren't efficient ways to test these sensors for accurate calibration. An infrared scene projector (IRSP) is a way to provide a solution for this problem.

Up to this point the dominant technology for IRSP systems has been emitter resistor arrays that use the Micro-Electro-Mechanical System (MEMS) technology. This method has many limitations such as the inability to emulate a non-black-body spectrum, poor fill factor and the apparent temperature is limited by an actual temperature of less than 700 degrees Kelvin [1]. As a result of these inefficiencies the development of a more reliable IRSP using alternate technologies has gained significant

importance. Our research group built the first IRSP using a Super Lattice LED Array System (SLEDS) that projects scenes in the mid-wave infrared (MWIR) spectrum (3-5um) [4,5]. This system is fully functional and it has been subject to hundreds of hours of operation at multiple locations in the country [5]. The SLEDs scene projector has a 512x512 48um pixel resolution, it is single color, and it is also capable of reading in DVI video signals and project them in infrared [2,5].

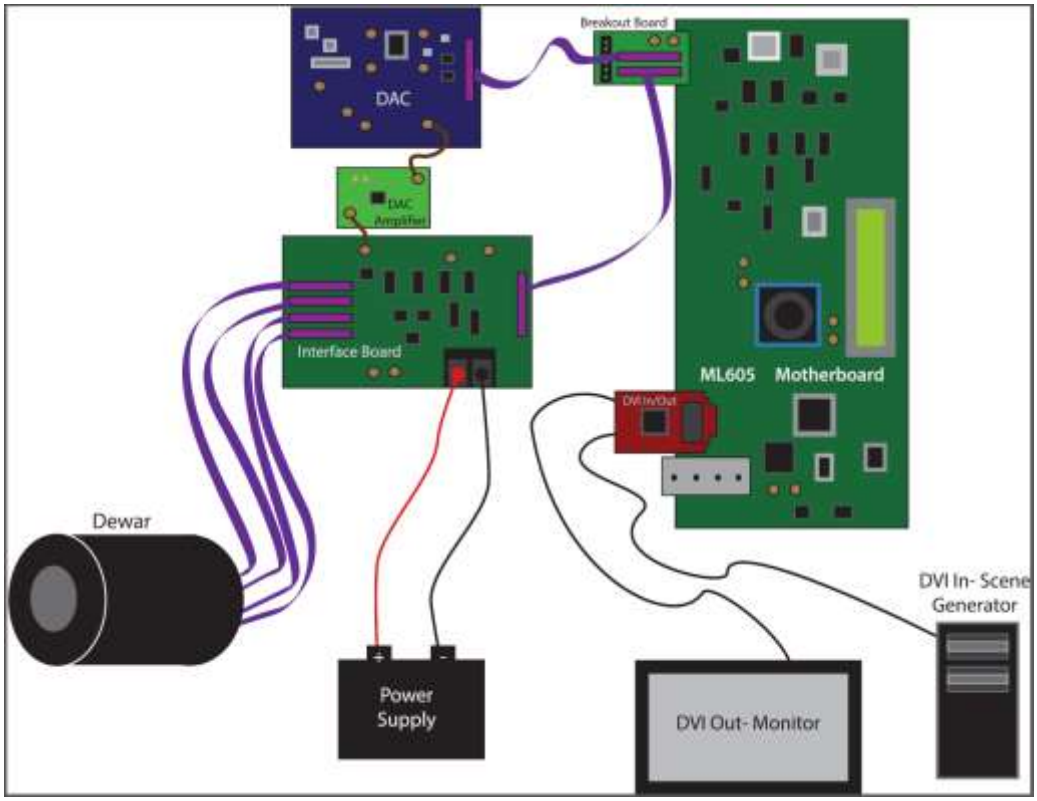


Figure 1.1: Hardware architecture of the 100Hz SLEDS IRSP

Since the development of the SLEDS projector was a success the team has been working on improving this technology. The next system that is nearing completion is the Two Color SLEDS Array (TCSA). This new IRSP has two different

emitters that output distinct wavelengths and it will operate at higher speeds reaching a 1kHz frame rate, ten times that of the SLEDS projector [6].

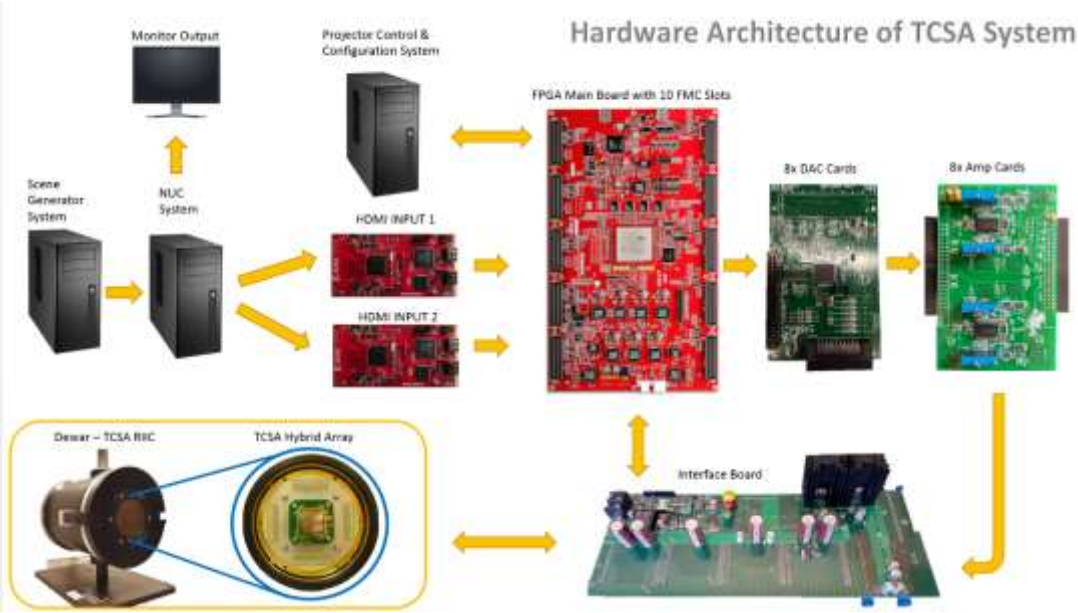


Figure 1.2: Hardware Architecture for the TCSA IRSP

1.2 Motivation

The SLEDS project was a great step forward in the understanding of MWIR projectors using LED technology. We were able to gather valuable information on the limitations of the projector and as a result in the upcoming IRSP the architecture of the RIIC has been improved. The continuing research involves a significant amount of testing of each component in order to assure maximum quality and performance. The RIIC is a key component that undergoes different rounds of testing as it is the heart of the system. Apart from SLEDS, there are three RIICs line up to be integrated to their own IRSP, these are the TSCA RIIC, Night-glow SLEDS (NSLEDS) RIIC, and the

High Definition Infrared LED (HDILED) RIIC. All these RIICs share a common architecture but there are several differences in the way they operate. This was the reason that pushed for a modular testing platform that could be used to test all the different versions of RIICs as it would reduce cost and provide a faster delivery of results. The work described in this paper is very important to determine the best performing RIICs which are chosen to undergo the process of hybridization. The process of hybridization is where the optical emitters are mated with the RIIC using flip chip bonding or post-foundry growth processes [7] as shown on figure 1.3.

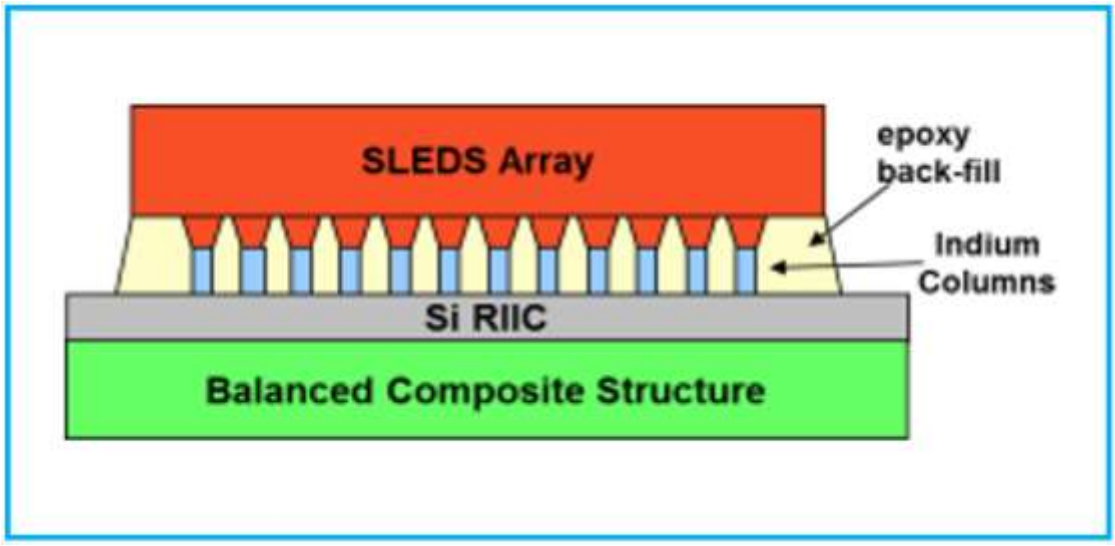


Figure 1.3: SLEDS hybridization using flip chip bonding

Chapter 2

RIIC OVERVIEW

2.1 Read-in Integrated Circuits Definition

Read-out Integrated Circuits (ROIC) currently dominate the market for 2D array driving chips. A ROIC is based on complementary metal-Oxide semiconductor (CMOS) logic and it is use to read the output data of different type of sensors arrays. The type of sensor arrays that these circuits are compatible with range from mid wave infrared sensors to x ray sensors. Currently the only major manufacturer of ROICs is FLIR, a company that specializes in developing thermal cameras. Their cameras used their ROICs to read the data from the infrared sensor pixels [9]. Our research team focused on the other end of these type of array driving chips. Our Read-in Integrated Circuits do not read in data from sensor, instead they drive emission devices, such as LEDs, that will provide the data for the sensors in a ROIC system.

The RIIC architecture is also based on CMOS logic. The RIIC is composed of many pixel driving circuits arranged into a square array that simultaneously drive each emitter with a variable current that can be controlled externally. After the hybridization process, each of these circuits controls one or more IR LEDs depending on the RIIC model. Therefore, a RIIC can be think of as an array of pixels that can't emit light just yet. Figure 2.1 shows the schematic design for the pixel that is part of the SLEDS projector. Two pairs of pass transistors, X to XB and Y to YB, control the gate voltage of the driver transistor which supplies current to the corresponding IR

LED [2]. More details on how the circuit design for the first SLEDS RIIC works can be found on Kassem Nabha's Master's thesis.

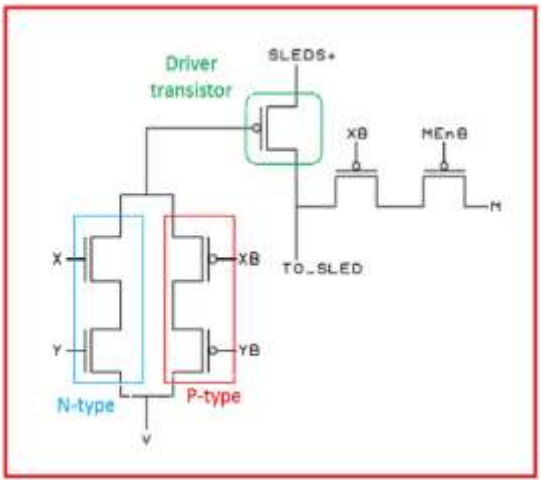


Figure 2.1: Single pixel schematic

2.2 Two Color SLEDs Array

The TCSA array is the second generation of RIICs developed in our research group. In this version of RIIC driving the LEDs is done by using two different driving transistors connected in parallel. A strong transistor is used to provide a current in the milliamps range, and a weak transistor that supplies currents in the micro-amp range. These two different intensity transistors allow the projection of finely tuned images and a more dynamic apparent temperature. Using this new method a hot body, such as a rocket taking off, will be drawn using the strong transistor, and the background will be drawn mostly by the weak transistor. In addition, the introduction of a second color means that every pixel in the RIIC drives two LEDs. The first color, which we call red

for simplicity purposes, is driven by P type Metal-Oxide-Semiconductor Field-Effect (PMOS) transistors, while the second color, blue, is driven by NMOS transistors.

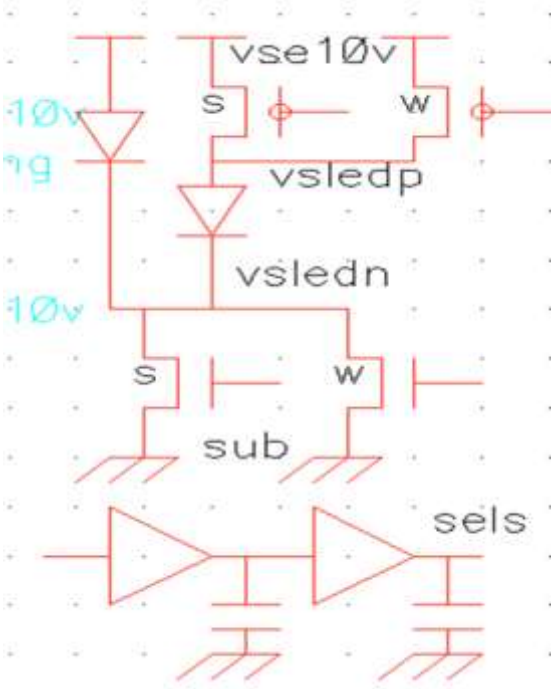


Figure 2.2: Pixel circuit for TCSA

2.3 Night-glow SLEDS Array

The NSLEDS array is a very special RIIC for two reasons, the first reason is that this is the first RIIC that uses 24-micron technology as opposed to 48 micron used by TCSA and SLEDS, and the second is that it completely moved away from having PMOS as the driver transistors for LEDs.

Using this new 24 um technology for chip design means that the pixel size in the NSLEDS array became smaller enabling the design to fit double the pixels in the

same area producing a resolution of 1024x1024. However, one limitation of the NSLEDS RIIC is that there are only enough address lines for a 512x512 array. The solution to this was to create a super pixel architecture where two different pixels, each containing two single color LEDs, share a common address line which uses a multiplexer signal to control each pixel individually. Figure 2.3 shows the super pixel structure, vsledp and vsledn are a pixel, vsledp2 and vsledn2 are the second pixel. The load line is the multiplexing line.

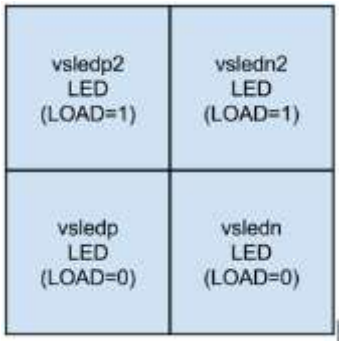


Figure 2.3: Super pixel in the NSLEDS RIIC

Having NMOS be the drivers of the LEDs in the NSLEDS RIICs saves space by eliminating the big n-well that PMOS transistors need. Figure 2.4 is the driving circuit of a single LED with both the strong and weak transistors being N type. In the super pixel there is four of these circuits because every subpixel has two LEDs.

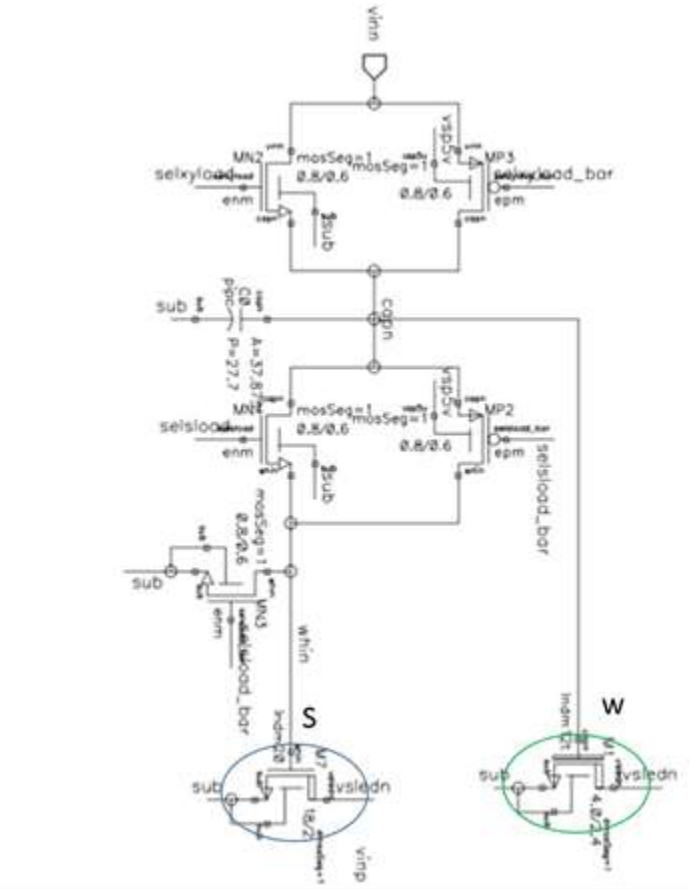


Figure 2.4: Circuit for a single LED in a pixel of NSLEDS

2.4 High Definition Infrared LED Array

The HDILED array is the second RIIC that uses the 24-micron technology for its pixel design. Thus the HDILED RIIC can be think of as a derivate design of the NSLEDS because the same architecture was taken but it was upgraded in three major areas.

- The number of pixels in this RIIC was doubled vertically and horizontally thus making the resolution 2048x2048. This required the addition of extra

X and Y addresses. This work required updating the logic of the RIIC in the schematic and layout design to enable it to handle the additional bits.

- Because the HDILED RIIC is four times the size of TCSA and NSLEDS RIICs the height and width had of the chip had to be reduced by 1000 um to allow a new stitching pattern. In addition, the corner pads were reduced by 10% in order to make it fit in the wafer reticle.
- As a result of the size changes a new stitch pattern was used in HDILED. TCSA and NSLEDS used the same stitching pattern it their design, this patters is a simple 2x2 core piece stitch as displayed on figure 2.5. HDILED uses an 8x8 core piece stitch which is significantly more complicated, this pattern takes four 2x2 core piece stitched blocks and fuses them using components called “cut pieces”. Figure 2.6 illustrates this stitching pattern.

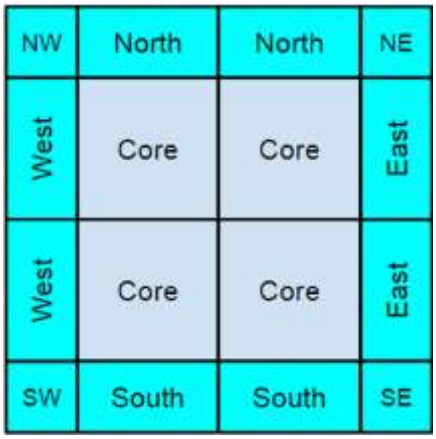


Figure 2.5: Stitching pattern for NSLEDS/TCSA

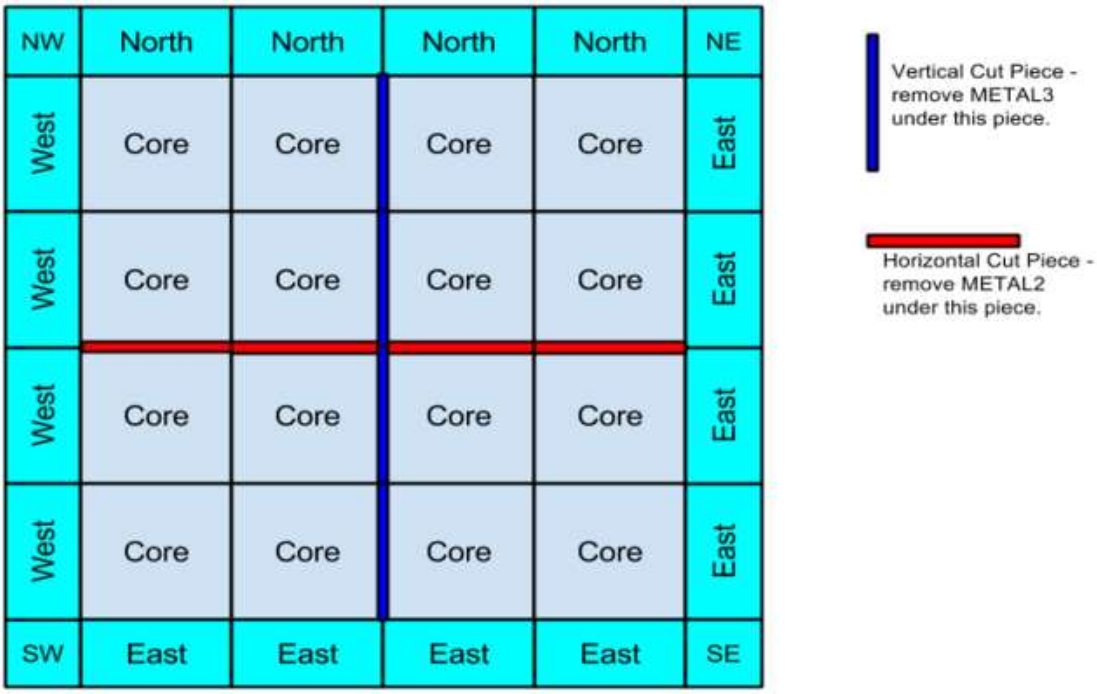


Figure 2.6: Stitching pattern for HDILED

2.5 Programming the RIIC

Starting with TCSA all RIICs use Serial Peripheral Interface (SPI) as a way to receive instructions from a master device. The SPI register of a RIIC chip holds forty-one bit registers in a chain of flip-flop blocks. Parallel to this chain there is another identical chain of registers which grabs data from the first and connects to the internal logic of the chip. The reason why there are two identical chains in parallel is because the first chain is connected to the outside and receives the data serially until the sender stops, then a load signal makes the data in the first set of registers be loaded instantaneously to the second set. The figure 2.7 shows the schematic for the SPI registers of the RIIC, full view and zoomed view. And appendix A has a table that explains all the configuration bits that get set through the SPI register.

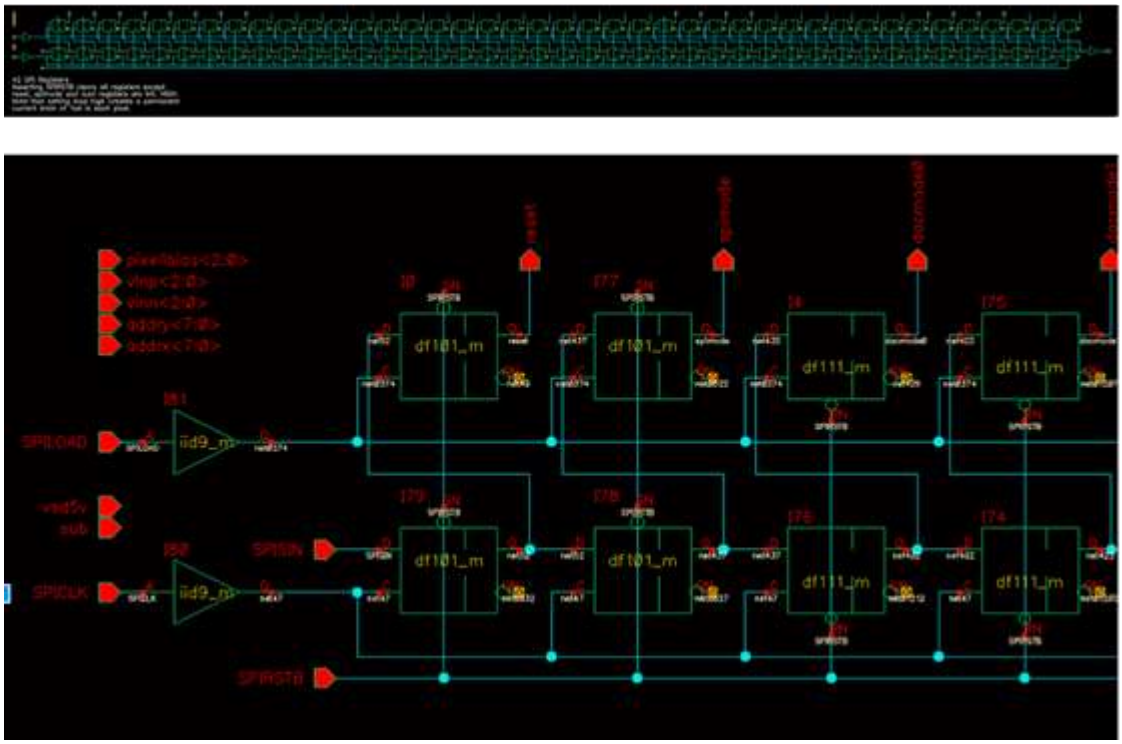


Figure 2.7: Schematic of the SPI architecture in a RIIC

Before writing to the SPI registers the “spiload” signal, which loads the second set of registers, has to be pulled low, the “spirstb” signal, which is an active low reset for the SPI registers, also needs to be low in order to reset any trash in the registers. During writing, the “spirstb” signal is pulled high to avoid resetting, then the “wspi” signal, which enables SPI writing to the RIIC, is pulled high, also the one of the four “wen” signals, which select which quadrant of the RIIC is enabled for writing, has to be pulled high. Finally, the “spiclk” signal and the “spisin” signals, clock and data respectively, shift in the data to the SPI registers. After writing, the “spiload” signal is pulsed once to allow the second set of registers to copy the contents of the first. At this

time if more data needs to be written all the control signals keep their values and the writing process happens again. Otherwise the “wspi” signal is pulled low to avoid any modification of the contents of the registers. Figure 2.8 is an example of how the SPI control signals would be configured to write two bytes of data to the RIIC’s SPI register. The last signal that forms part of the SPI architecture is “spiout,” which is an output signal that is connected to the output of the last register of the chain. The “spiout” pin is important because the SPI registers work as a first in first out (FIFO), meaning that after forty-one bits have been shifted in if there are more bits coming in, then they will start to come out on this pin.

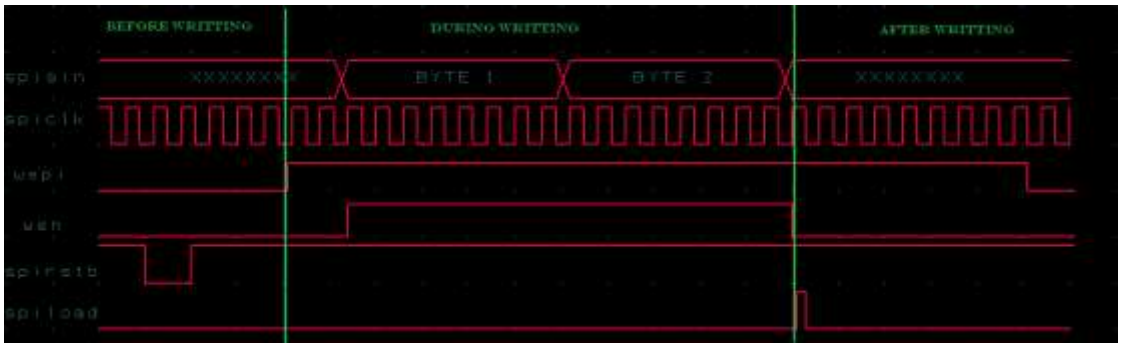


Figure 2.8: Control signals for SPI being configured for writing to the RIIC

Chapter 3

SCALABLE TESTING SYSTEM DESCRIPTION

3.1 Early Stages

In late 2014 our team received the first TCSA version 1 RIIC, this was a 16x16 array enclosed in a pin grid array (PGA) package. At that time the testing system for RIICs was just beginning to take shape. The first TCSA RIIC was tested using a simple test bench that consisted of an Arduino microcontroller, a custom printed circuit board (PCB) called the modular RIIC testing platform (MRTP), and an Oscilloscope. The Arduino's SPI interface was used to set the configuration bits in the RIIC's SPI register. Also at that time the code used in the Arduino was limited because every time a different instruction needed to be written to the RIIC the code needed to be changed manually and re-uploaded to the microcontroller. The MRTP board was used to provide a way of separating the all the signals from the RIIC into pins used power, and set the chip. More details on this board will be discussed on section 3.2. Finally, the Oscilloscope was used to monitor the signals of the RIIC during operation.

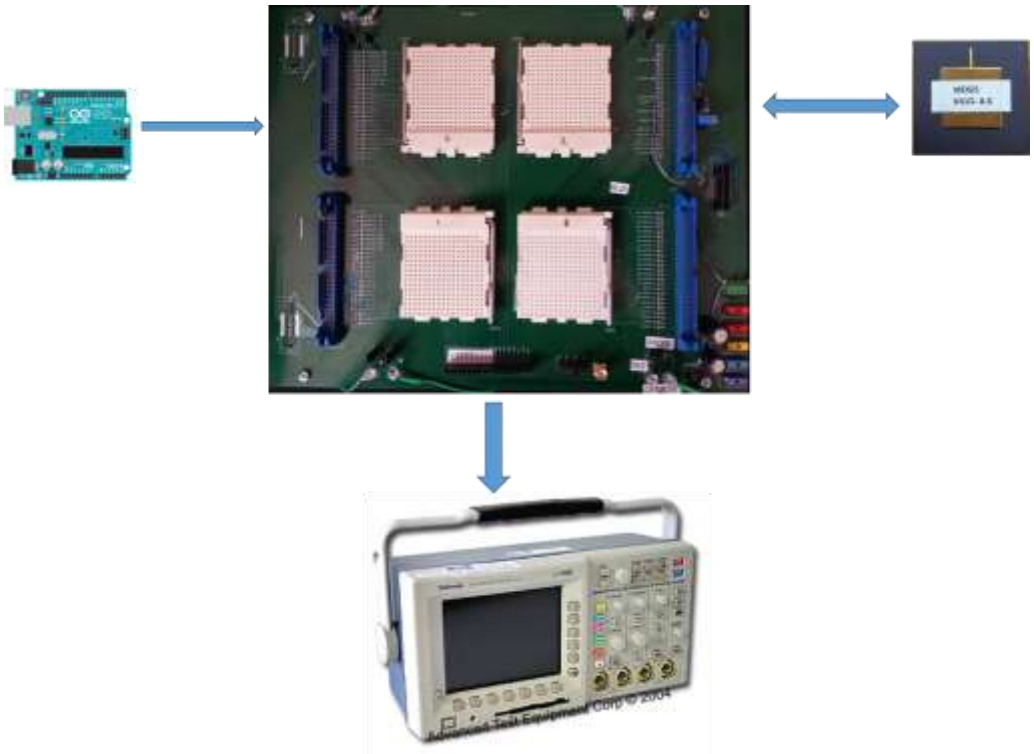


Figure 3.1: Initial stages of the test system used for testing RIICs

The initial testing setup made it difficult to test a RIIC effectively making it very hard to find a wiring mistake in the internal circuits of the RIIC. The first TCSA 16x16 RIIC that was tested had a wiring error where select strong signal in the SPI registers was connected to routing of the an internal three bit DAC in the RIIC designed to provide power the NMOS LED of the pixels. And the DAC was routed to control the select strong signal, which when HIGH it sets the pixel to be driven by the strong transistor.

After the wiring mistake was identify The TCSA RIIC designed was fixed and resubmitted for fabrication. The second version of the TCSA 16x16 PGA packaged RIIC arrived a couple months later without the wiring mistake. By then the testing

system had been improved, and it continued to become better as it was more scalable and modular, the next section is going to describe the current scalable and modular RIIC testing system that is used to test all the different version of RIICs so far.

3.2 Test System Description

The scalable testing platform use for RIIC testing has evolve into a robust, fast, and easy to use system. This section will explain all the different components that make up the testing platform.

3.2.1 Python Software

The brain of the system lies in the software that controls its functionality, all of the processing and computing power behind the testing platform happens on a computer running python code. Having the main code running on a computer allows for easy modification and modularity. Another advantage is that it can be saved on repositories and downloaded on any other machine, in addition it can be controlled remotely by the user.

The Python software was written in a way such that there are many small pieces divided across python classes and/or modules that can easily be imported and reused over different applications. The first thing that happens whenever a script is executed by a user is a scanning of all the COM connections to the host machine. By doing this the host machine establishes a serial communication channel to every device needed for the specific tests, such as the Arduino microcontroller or the Keithley meters that are part of the set up. This method ensures that the testing platform can be rearrange by moving the connections around and it will still work.

Once a serial connection has been established the user can select the type of test to be performed on the RIIC. The information is then packed into a packet that has a header field and a body. The header field contains information such as, the length of the packet being sent, the version of the RIIC that is being tested, the type of test that is being asked for, and various flags that turn on or off input signals. The body of the packet are six bytes of data that contain the configuration bits for programming the RIIC. The software has also eliminated the need for an Oscilloscope because it keeps logs of the test done and the data collected, which then can be graphed and analyzed.

3.2.2 Microcontroller Code

Figure 3.2: Arduino microcontroller

The Arduino microcontroller code is not responsible for much, its main job is to be a translator between the serial input from the computer and the SPI interface of the RIIC. When the Arduino is powered on, the first thing it does is to broadcast its ID to its serial port. When the python code finds that ID on one of the computer's ports it attempts to connect to the Arduino, if the connection is successful the Arduino replies

with an “ok” message. After that it just waits until it receives a packet from the serial line, then it looks at the header to choose the action to take and performs it. The Arduino code can also send data back to the computer but that feature is only used for debugging.

3.2.3 Bare RIIC Testing Board

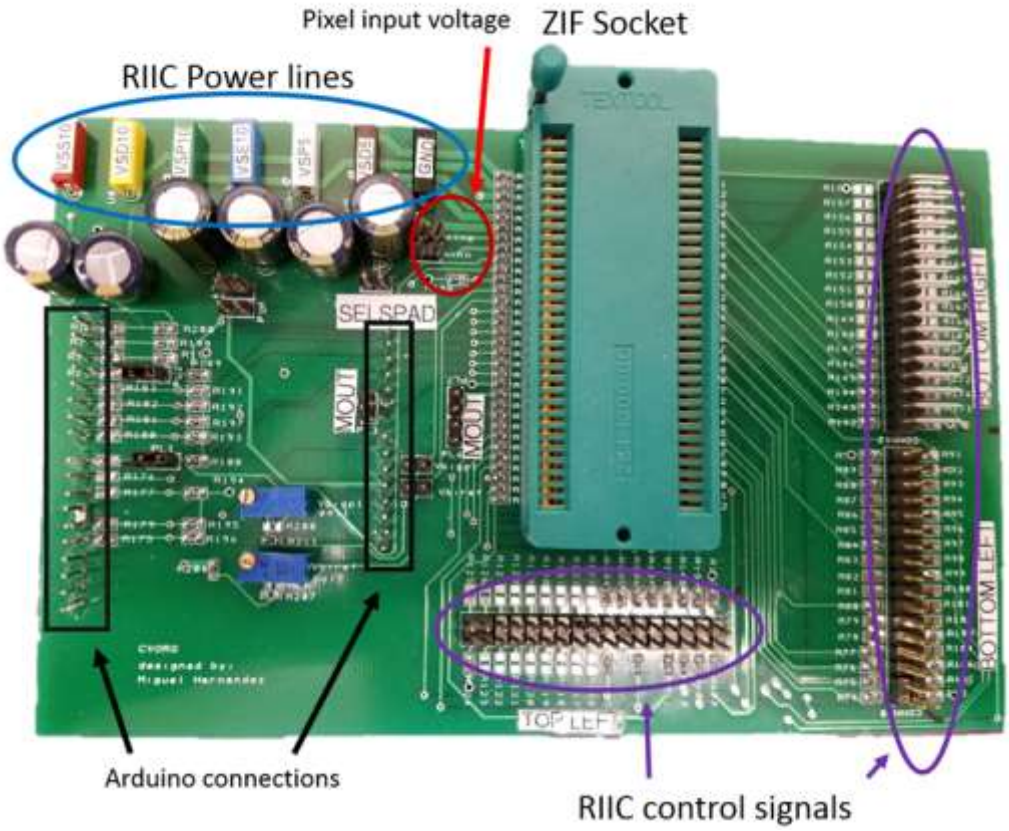


Figure 3.3: Bare RIIC Testing Board, a custom PCB made for the testing system

The Bare RIIC Testing (BRT) PCB is a custom board that attaches to the Arduino as a shield and routes the signals to the correct pins on the test RIIC

delivering power to it. The critical feature of the BRT board is a zero insertion force (ZIF) dip socket that sits in the middle and allows us to modify where the control signals are routed [3]. There are four different ways in which the signals can be routed because the full size RIIC is tested in quadrants. The contacts for each quadrant are place on the corners, thus the signals change orientation for every corner. Figure 3.3 shows the four small PCBs used for each of the corner pad configuration of the RIICs.

Figure 3.4: Four wire configurations for routing signals to the RIIC

3.2.4 Modular RIIC Testing Platform

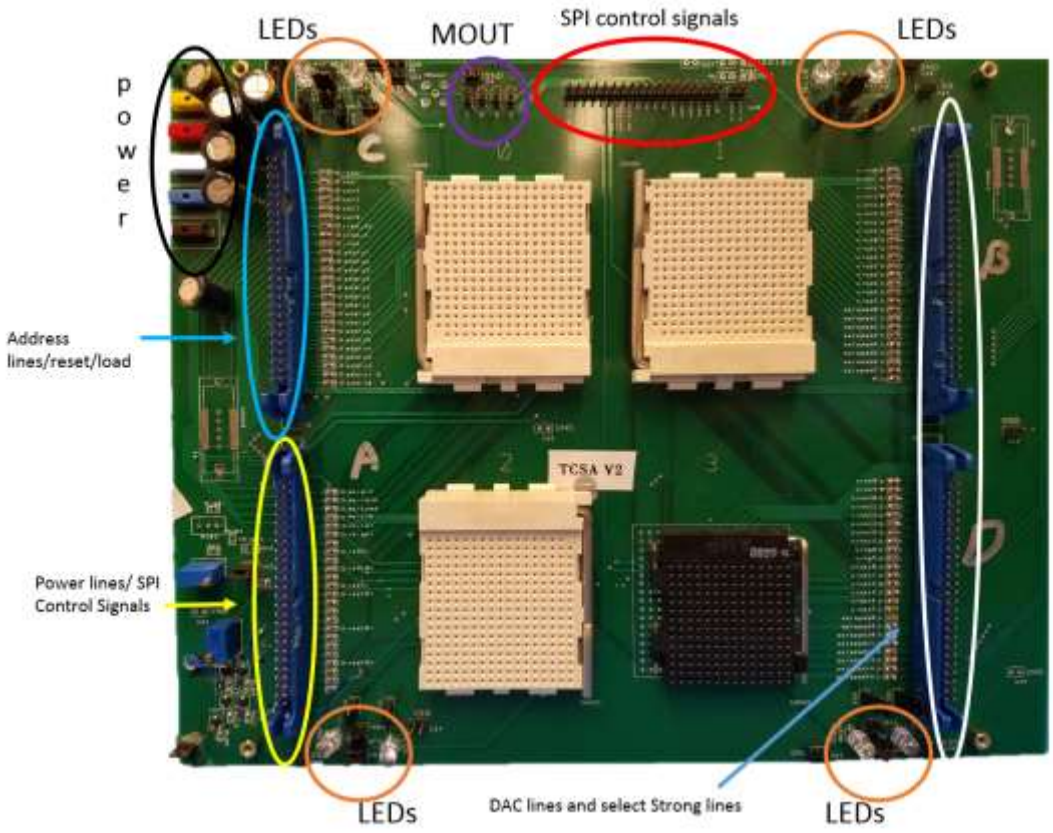


Figure 3.5: MRTP board used in the STP

The MRTP board is used to test the RIIC in its PGA package because it has a PGA socket that breaks out all of the necessary RIIC signals. This allows the control signals from the computer to be routed through the system and to the RIIC. The MRTP also has the ability to run up to four RIICs simultaneously allowing us to simulate a bigger system and test that RIICs work in harmony with each other. There are four sets of macro LEDs on every corner of the MRTP, these LEDs are used to simulate the behavior of IR LEDs during testing. This Board also has the same ribbon

cable pin configuration that the Dewar will use on the finish system which allows us to incorporate it to the system shown in Figure 1.2 and simulate a completed IRSP. Another feature of the MRTP is that it can communicate directly to the Arduino bypassing the BRT. The purpose of this is to have the ability to do quick simple tests of the RIIC [3].

3.2.5 Keithley Meters

Figure 3.6: 24xx Keithley model used in the test setup

Two programmable Keithley source meters are used in the STP. One is programmed by the python code to do voltage sweeps of the pixel circuits in the RIIC, and the other Keithley is programmed to measure the output SLED current on a monitor out (MOUT) pin. The MOUT pin can produce voltage or current readings depending on how the configuration bits of the SPI register of the RIIC are set. The two Keitleys are connected together by a cable that sends a sync signal. Every time the Keithley acting as the source steps through a voltage value it sends a trigger signal on that connection so that the other Keithley knows that it is time to take a measurement.

The measured data is sent to the computer through a serial connection, that data is saved and graphed at the end of each run.

3.2.6 RIIC in PGA Chip

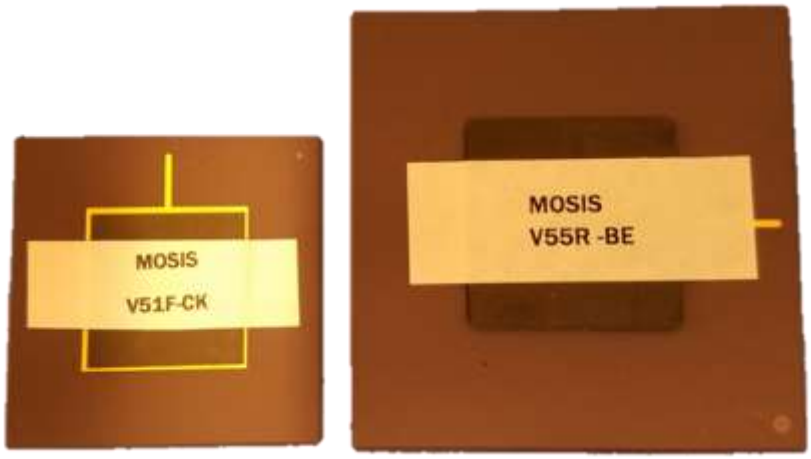


Figure 3.7: PGA package for TCSA and NSLEDS (left), and HDILED (right)

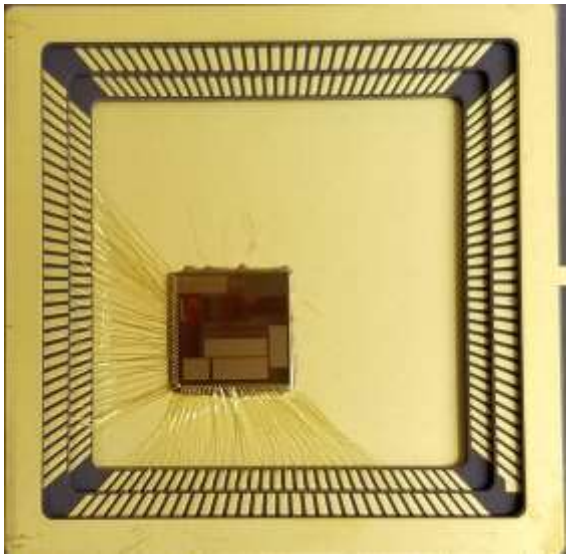


Figure 3.8: Wire bonding of HDILED 16x16 RIIC

The main reason to have a RIIC in a PGA package is because it allows easy handling and testing without worrying about damaging the die inside, Figure 3.8 shows the actual 16x16 RIIC grown on Silicon and how it is wire bonded to the pads that go to the pins. And figure 3.7 shows the top view of the PGA package. All versions of the 16x16 PGA RIIC have pixel (0,0) brought out to external pins, these pins are used to drive the macro LEDs shown on Figure 3.4.

3.2.7 HDILED Housing Structure

Figure 3.9: HDILED Housing Structure attachment

The HDILED PGA packaged RIIC is bigger than TCSA or NSLEDS, as a result it does not fit in the PGA sockets of the MRTP board. A custom pair of PCBs where created to provide a housing structure for the HDILED RIIC. The bottom piece of the structure plugs to the PGA socket of the MRTP, then it brings the signals to two sets of pin female connectors where the top piece connects to. The Top board takes those signals and arranges them in the right place on a bigger PGA socket where the HDILED RIIC sits.

3.2.8 Full Size RIIC on Wafer

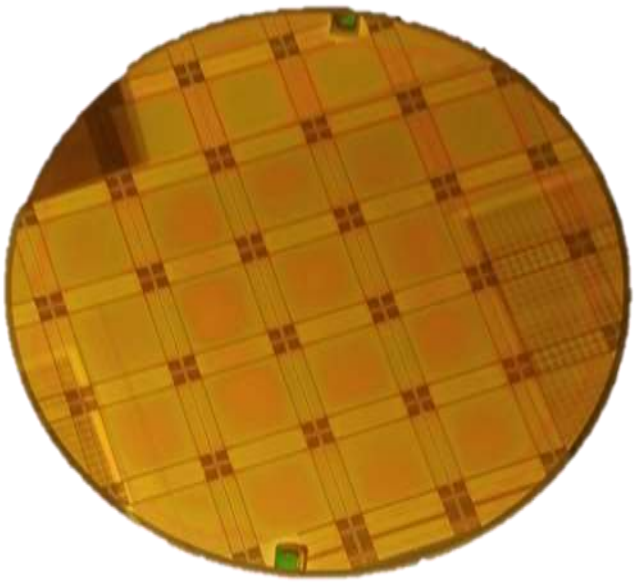


Figure 3.10: 1024x1024 NSLEDS RIICs on an 8” Silicon Wafer

The Full size RIIC comes in an 8” silicon wafer that usually holding about nineteen complete RIIC for TCSA and NSLEDS. Both TCSA and NSLEDS have been tested and the best performing RIICs were sent to Teledyne to undergo Hybridization. The HDILED wafer is schedule to arrive on mid-summer 2016, however that wafer will only fit four RIICs as HDILED is twice as big in size and resolution than NSLEDS.

3.2.9 Custom Probe Card

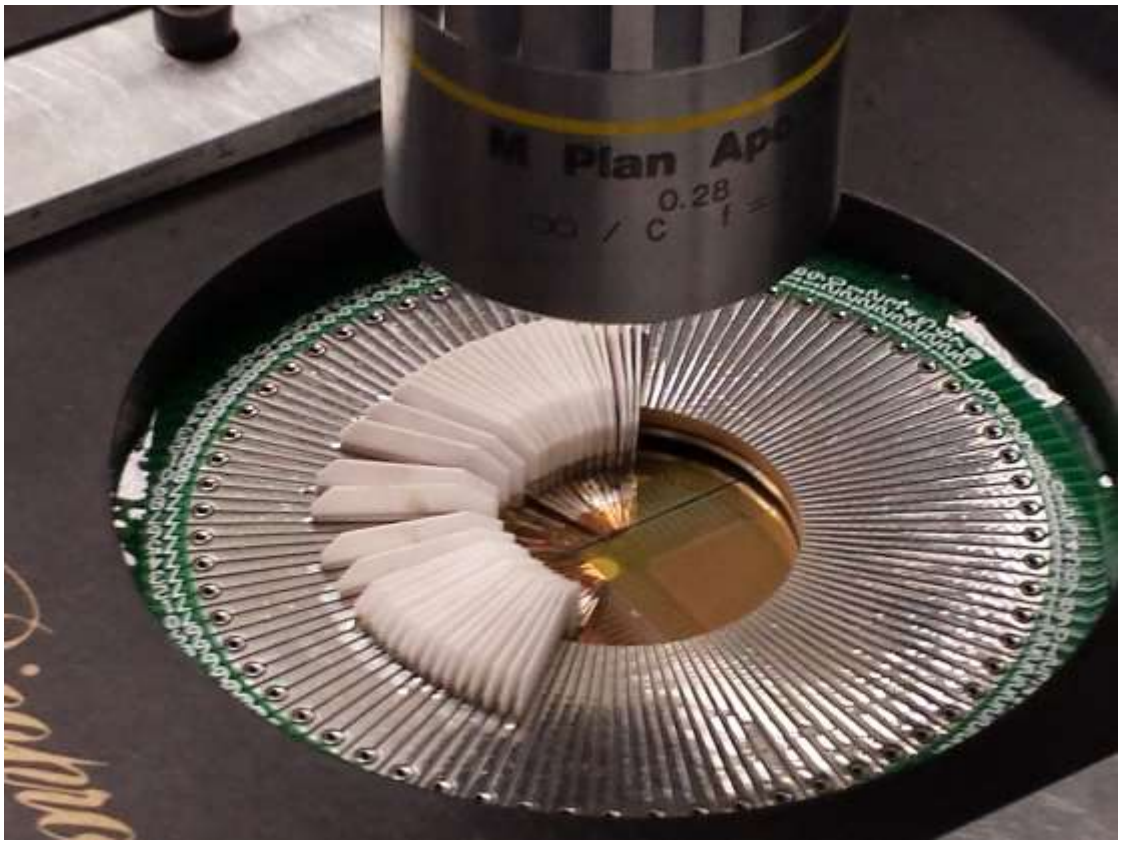


Figure 3.11: Custom probe card contacting the pads on a RIIC being tested

The probe card use for testing the RIIC receives the signals via ribbon cable from the BRT board. This probe card is set to probe only one corner of the RIIC at the time accessing only one quadrant of such RIIC. When probing pads so small that can't be seen with the naked eye the accuracy of coming in contact with the pads becomes a delicate process. To address this problem all the contact pads which control a single quadrant were placed symmetrically along the corners of the RIIC which is probed one quadrant at a time. This method of probing reduces the risk of placing the probing arms on the wrong pad because the corners make it easier to align. Both TCSA and

NSLEDS use the same probe card, however in HDILED the probe card will be a little bit different to account for the extra address line. Figure 3.12 shows how the four corners of the RIIC chips have identical pads, thus every time a new quadrant is to be probed the wafer is rotated 90°.

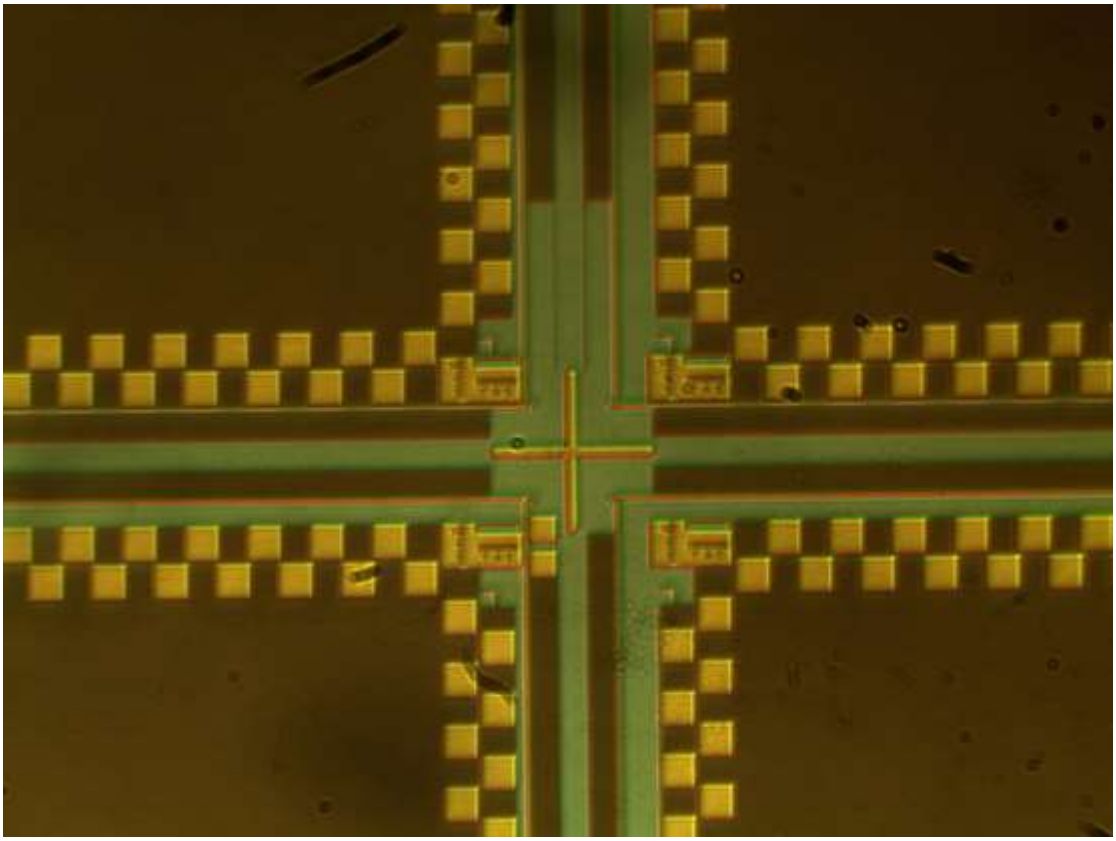


Figure 3.12: Contact pads on the corners of the full size RIIC chips on a wafer

Chapter 4

TESTING AND DATA GATHERING

4.1 Phase One

Testing the RIIC happens in two phases, during the first phase all of the tests that the RIIC is going to undergo are develop. Phase one can be think of as a preparation phase. Phase two happens in a clean room on the full size RIIC. Phase two can be think of as a selection process, no tests are developed here but the RIIC undergoes all the tests developed in phase one.

During the first phase of testing all of the test are developed using a 16x16 RIIC as the subject. Figure 4.1 shows the configuration of the testing set up used in phase one to test the 16x16 RIIC, the components of the set up are: (1) A computer running the Python software, (2) Arduino, (3) BRT, (4) interface connector between the BRT and MRTP, (4)MRTP, (5)16x16 PGA RIIC, (7) Keithley acting as voltage source, (8) Keithley acting as current meter. The type of testing done on this RIIC helps identify any possible malfunctions in the design so that they can be corrected before producing a full size RIIC.

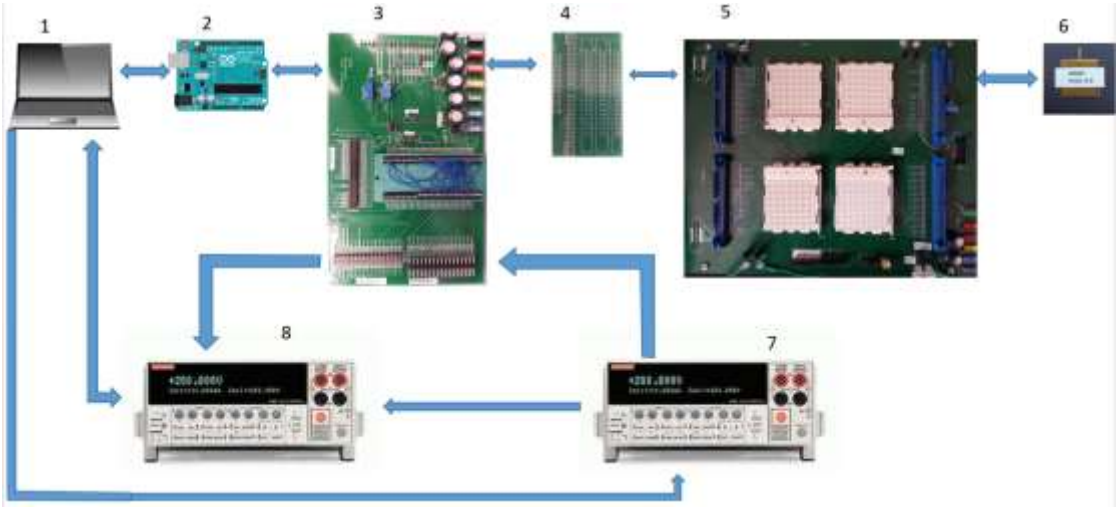


Figure 4.1: Test set up used in phase one

Since Different versions of RIICs have differences in their design it means that some of the tests written for the RIICs only work on a specific model. However most of the test are general meaning that all RIICs can be tested with the same tests. The following sections explain the different tests developed in phase one.

4.1.1 Power Rail Test

When semiconductors are grown on Silicon it is very common that different layers may have defects causing shorts in the design. The first test to do on any RIIC is to check for such shorts on the power rails. This is done by probing each power rail against each other and ground to make sure the resistance between them is infinite or a very large value in the Mega Ohms range.

4.1.2 Communication Test

As mention before the RIIC is programmed through SPI, therefore SPI has to work properly before proceeding to test. Since the SPI architecture in the chip is a FIFO that outputs its input forty two cycles later, the best way to test is to write specific hexadecimal bytes continuously and watch that the exact bytes show up on the spiout pin after forty two shift cycles have passed. An example of this method is shifting in the value 0x01 followed by the value 0x00 five times, and then the value 0xFF another six times, this is a total of 16 Bytes of data or 96 bits. Thus the output on the spiout pin by that point will be: 0x7f, 0x00, 0x00, x0x00, 0x00, 0x00, 0x01. Figure 4.2 shows the communication test for some arbitrary hexadecimal values, the blue line is the data coming out on spiout after 42 clock cycles, the green signal is the data being sifted into the RIIC SPI registers, the blue signal represents the same data coming out of the spiout pin.

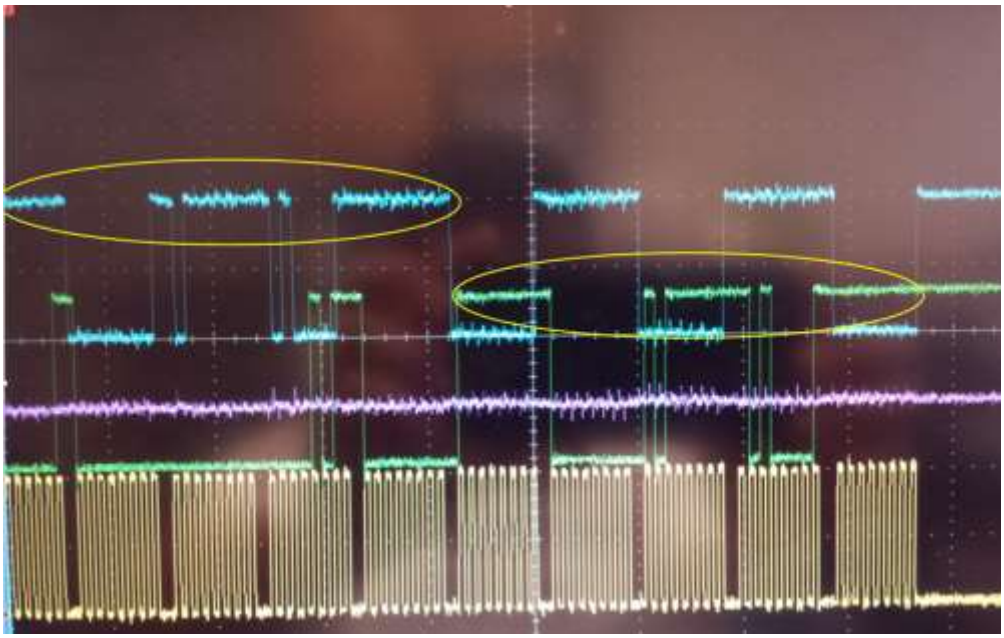


Figure 4.2: SPI communication test example

4.1.3 LED Test

As mention before RIICs in PGA form have their corner pixel (0,0) drive lines brought out to pins that are routed to a pair of macro LEDs placed on the MRTP board. Lighting up those LEDs with the RIIC is a great visual representation to ensure the driving circuitry works correctly. There are two ways in which the lighting LED test can be done, the first one is to configure the RIIC to “SPI mode” and use the internal DACs to drive the LEDs, the second is to configure the RIIC to “DAC mode 1” and use an external DAC to drive the pixel.

Using the first method the RIIC’s SPI registers are configured with spimode = 1, the address bits are set to 0x00 for x and 0x00 for y, then both internal DACs are configured with three bit values between “000” and “111”, where “111” is 0V and 000 is 5V. Using this method any time the voltage provided to the pixel drive circuitry is greater than the threshold voltage needed to light up a LED, the macro LEDs turn on with a brightness dependent of the voltage provided. This is only use as a fast check test as the internal DACs in the RIICs are not very accurate and their dynamic range is very limited.

Using “DAC mode 1” to light up LEDs is a better option. The way the SPI registers of the RIIC are configure is by having dacmode1= 1 , everything else is set to zeros, in this mode every pixel in the RIIC is addressed individually, and only the pixel being addressed receives commands from the outside while all others remain in an idle state. The Keithleys that are part of the testing set up act a bit different as both are used as voltages sources to provide the driving voltages for the pixel.

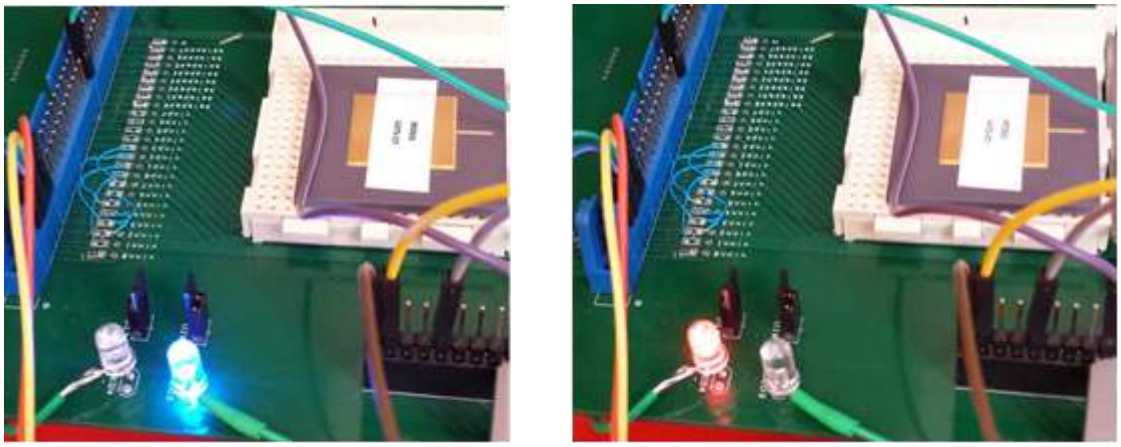


Figure 4.3: LED blinking tests on a TCSA RIIC

In the figure above the two macro LEDs are taking turns to blink, they are operating in dacmode1 and since the RIIC driving them is TCSA the blue is driven by a NMOS and the red is the PMOS. The LEDs may also be on at the same time if both Keithleys provide enough voltage at the same time.

4.1.4 Time Multiplexer and Two Step Reset test

NSLEDs and HDILED RIICs are special because they have super pixels which contain two pixels that share a common address line. The code written for this test sets a timer on the multiplexer signal, this time is set to be one second during phase one, meaning that every second the signal toggles form HIGH to LOW, or LOW to HIGH. When LOW half of the super pixel is selected and when HIGH the other half is selected. For instance if the multiplexer signal is LOW and the corresponding sub-pixel is told to turn on at that time, and then the signal goes HIGH and the other sub-pixel is selected, then the control over the first sub-pixel is loss, however the data that was sent to the first sub-pixel will remain thus it will stay on. This makes it possible to

have both sets of sub-pixel turn on at the same time which will be required many times when projecting an image on the IRSP. Figure 4.4 shows how using the multiplexer signal each sub-pixel can be addressed individually, and also on the right most image the four LEDs are lit meaning that one of those pairs of LEDs is frozen in an ON state from a previous fluctuation of multiplexing.

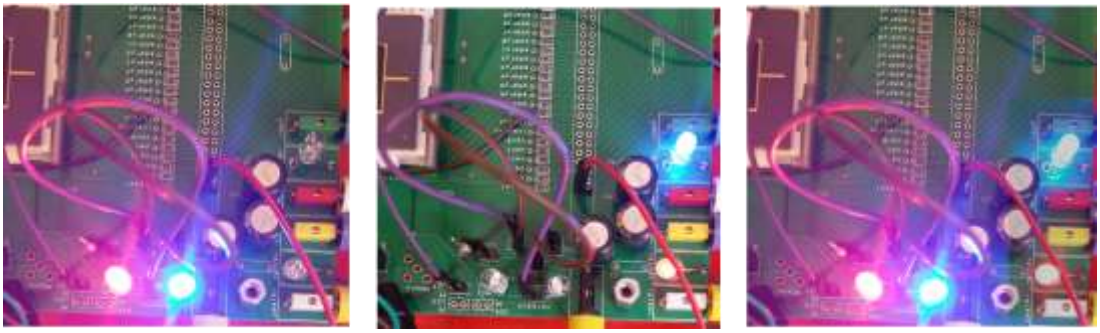


Figure 4.4: NSLEDs super pixel driving 2 pairs of LEDs

In order to reset the pixel memory and turn it off in an efficient way a two-step reset process is use. Basically right before the multiplexer signal toggles the reset signal is pull high for a brief period, this clears the pixel memory before selecting the other pixel, then the same process is repeated on the next pixel. Figure 4.5 displays how the two-step reset works.

Figure 4.5: Two-step reset of super pixel

4.1.5 Monitor Out Test

Driving the LEDs is a great way to visualize the behavior of the pixel logic, however, the greatest limitation of this is that only pixel (0,0) can use this method for all RIICs in a PGA package. The MOUT tests provide a way to tests all pixels in a RIIC regardless of its size, or architecture. Before writing code for running the different MOUT tests it is important to understand the DC response of the driving transistors in the pixel. Different simulations were ran using Cadence simulation tools. The simulations give ideal curves which are the reference points to compare the data that eventually is obtained from the transistors in a RIIC. Figures 4.6 and 4.7 display the four different types of driving transistors in TCSA and figures 4.8 and 4.9 are the two driving transistors found in NSLEDS and HDILED. All of the curves are for the drain current in micro Amperes (vertical scale) as a function of voltage at the gate of the transistor (horizontal scale).

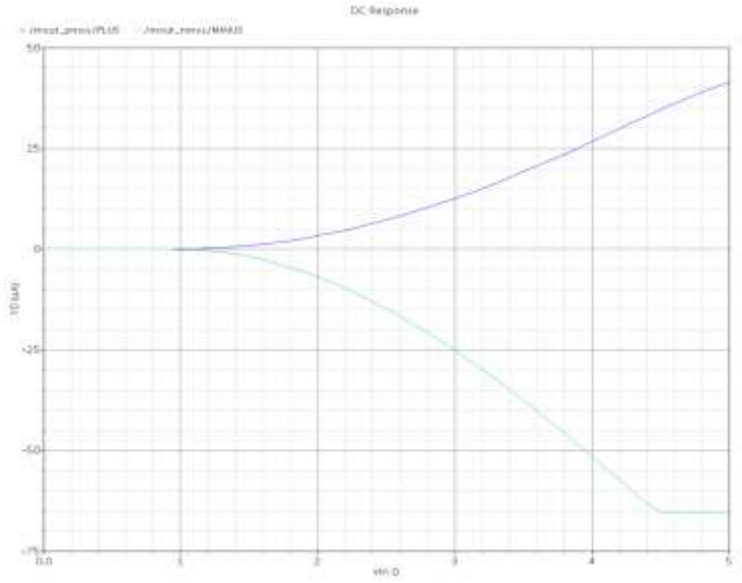


Figure 4.6: Simulated DC response for TCSA weak transistors

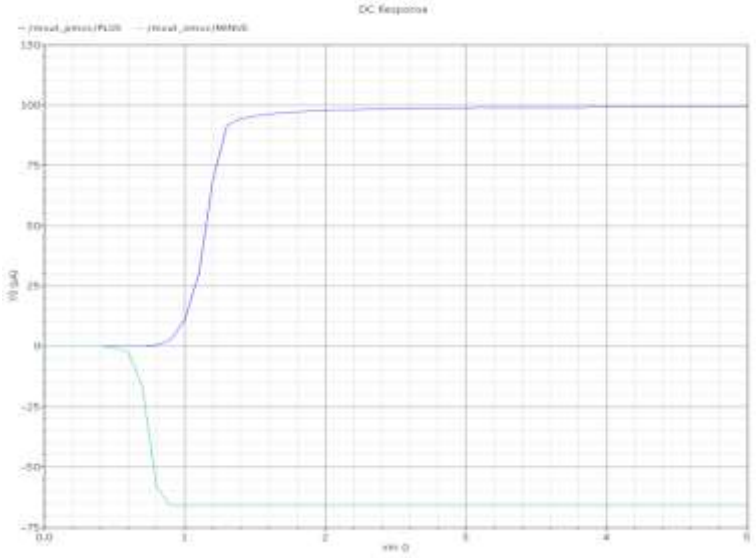


Figure 4.7: Simulated DC response for TCSA strong transistors

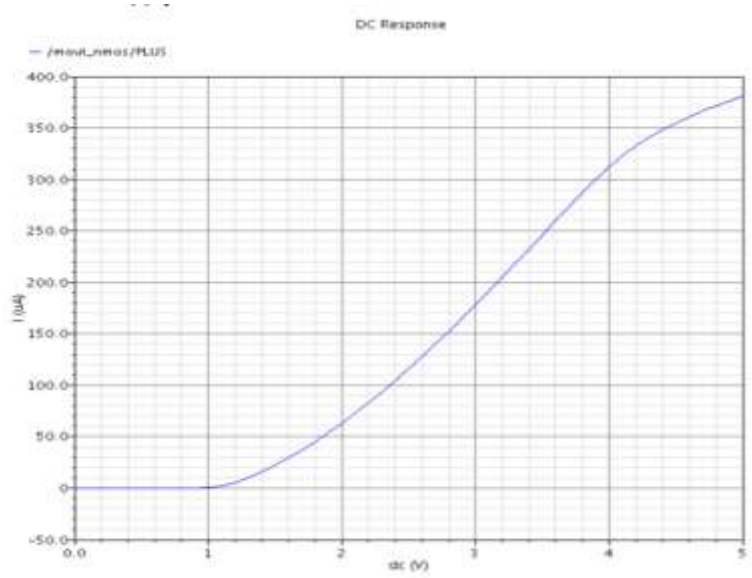


Figure 4.8: Simulated DC response of NSLEDS and HDILED weak transistor

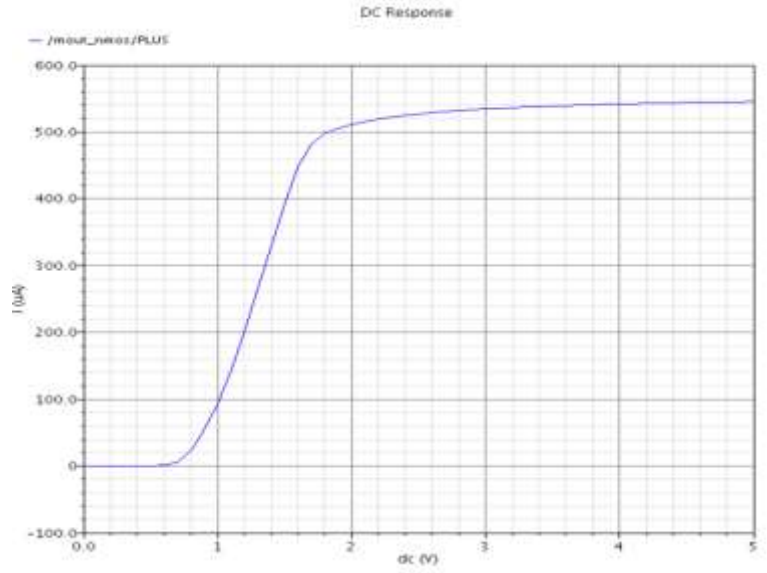


Figure 4.9: Simulated DC response of NLEDS and HDILED strong transistor

The TCSA RIIC requires the software to set four different configurations to test each specific drive transistor in a pixel, while on the other hand, NSLEDS and HDILED only required two configurations per pixel. For all the transistor sweeping tests the RIIC is configured to be in spimode, in addition, the MOUT pin has to be enabled appropriately by setting the SPI register “men”, which enables the MOUT pin, to logic HIGH, then there are two register bits that control which type of transistor to be monitored by the MOUT pin. To monitor a PMOS transistor register “mselp” has to be HIGH and register “mseln” has to LOW, or the other way around if testing a NMOS. These two registers cannot be HIGH at the same time because then the data on the MOUT line will be garbage. If they are both LOW nothing will be monitor on the MOUT pin even if it is enabled. On NSLEDS and HDILED the “mselp” register is not used since there are no PMOS drive transistors. The last register needed to choose a single transistor is the “sels” register. When “sels” is set to logic HIGH the strong transistor of a pair of either NMOS or PMOS is chosen, if LOW the weak transistor is chosen. All the mentioned registers will stay constant for the extent of the test, the only variable will be the pixel for which the specified transistor is being tested. In spimode only one pixel can be addressed at a time and the SPI control registers x<7:0> and y<7:0> select the pixel to be tested. During a sweeping test the python scripts set all the constants first, then a range of pixels is specified such as: from coordinate (x=0, y=0) to (x=15, y=15). Pixel (x=0, y =0) is the first to be tested, the next one is (x =1, y=0) because it is designed to increase in the x direction first. Once it gets to the end of the row moves to pixel (x=0, y=1), and the cycle continues until it gets to pixel (x=15, y=15).

When a sweeping test is completed it generates the data from figure 4.10 if the RIIC was a TCSA, or figure 4.11 if the device tested was a NSLEDS or figure 4.12 if it is a HDILED RIIC. The actual data collected from the 16x16 RIICs match the simulated curves for the most part. For example, by looking at the data collected from the weak NMOS transistor on the TCSA at 4 Volts the current is around 50uA where the simulation predicted 40uA at 5 Volts. These differences occur because the simulations do not take into consideration external factors such as noise, termination resistors, or capacitance of all the different components in the testing system. Once these result are successfully gather from the 16x16 RIIC all there is to do is change the range of pixel sweeping, which is a trivial thing, and everything is set to tests a full size RIIC on a wafer.

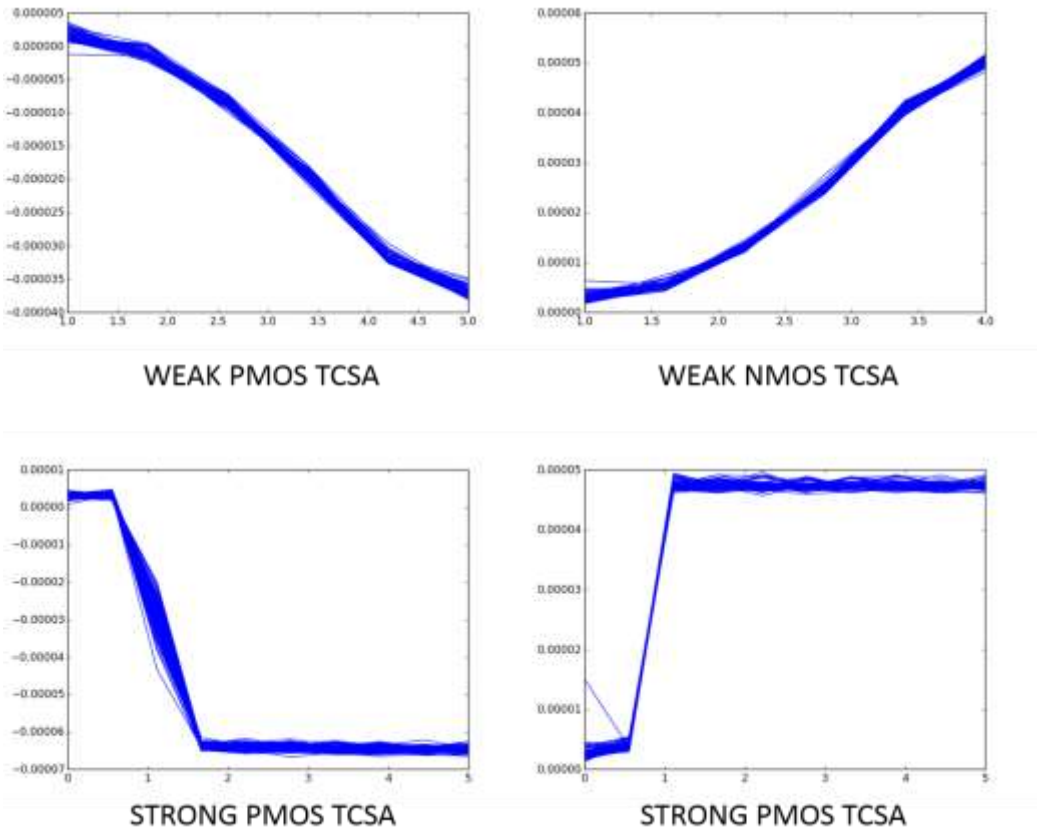


Figure 4.10: Data collected from sweeping a 16x16 TCSA RIIC

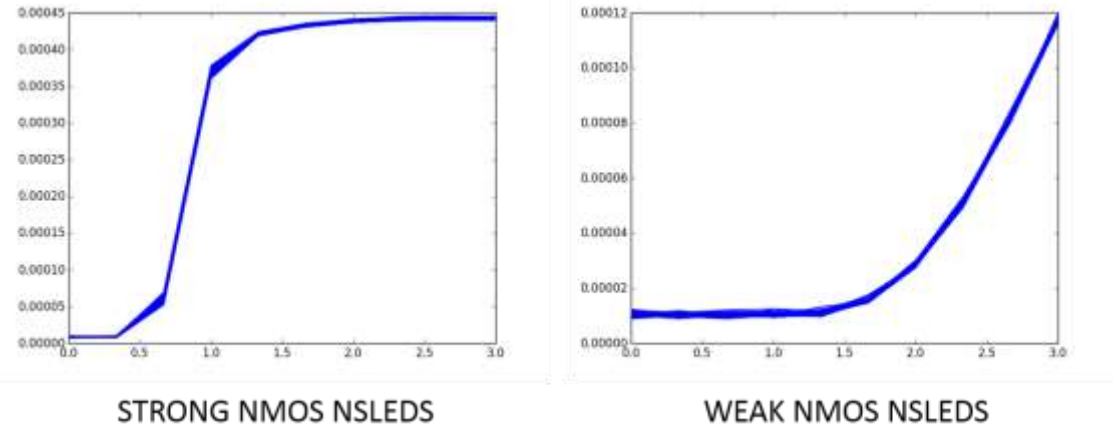


Figure 4.11: Data collected from sweeping a 16x16 NSLEDS RIIC

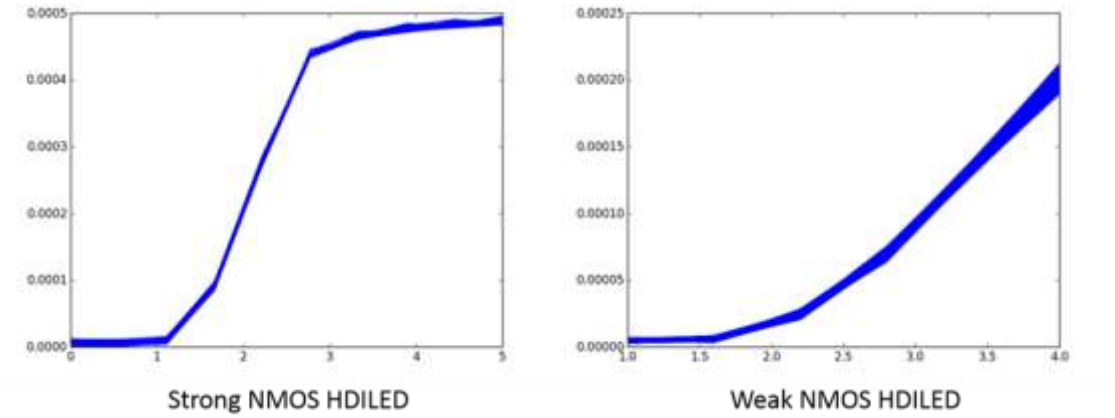


Figure 4.12: Data collected from sweeping a 16x16 HDILED RIIC

4.2 Phase Two

During the second phase the RIIC is not tested for functionally as that was done in phase one. The second phase of testing focuses on the sweeping tests of the pixels in the RIIC. All the testing is done in a clean room because any sort of contaminant could alter results or even damage the RIICs. Having a modular testing system facilitates the transition from normal laboratory test-bench to a clean room environment because all that is needed is to replace the MRTP with the probe card designed for testing our RIICs. Figure 4.13 displays the components that are used in the second phase of testing: (1) computer running the Python software, (2) Arduino, (3) BRT, (7) Keithley acting as voltage source, (8) Keithley acting as current meter, (9) custom RIIC probe card, (10) wafer with full size RIICs.

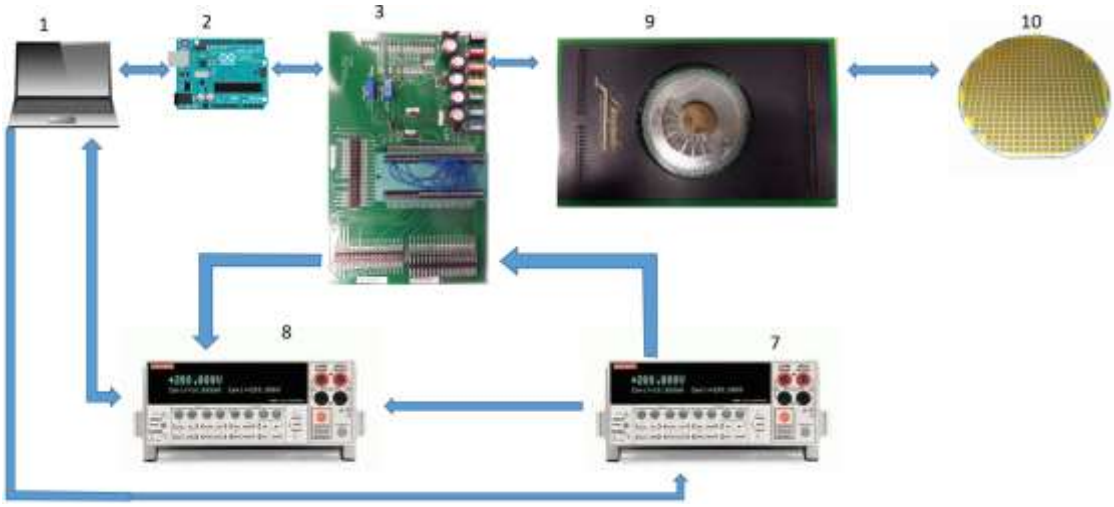


Figure 4.13: Setup used for phase Two

As mention previously the probe card and the RIIC are designed so that all the contacts for the control signals are located on the corners, therefore only one quadrant can be tested at a time. The four quadrants are South West, South East, North East and North West. First to maximize efficiency all of the South West quadrants are probed and tested. For instance figure 1.14 shows the wafer map naming system which is used to give every RIIC on the wafer a name. Using this map if the first RIIC tested is B1 on the South West corner (bottom left), the next test is done on C1 on the same quadrant, and then all consecutive RIICs are tested until D5's South West corner is tested. At that time the wafer is rotated 90° either clock wise or counter clock wise putting either the South East or the North West on the bottom right corner. The same process is then repeated until all quadrants have been tested.

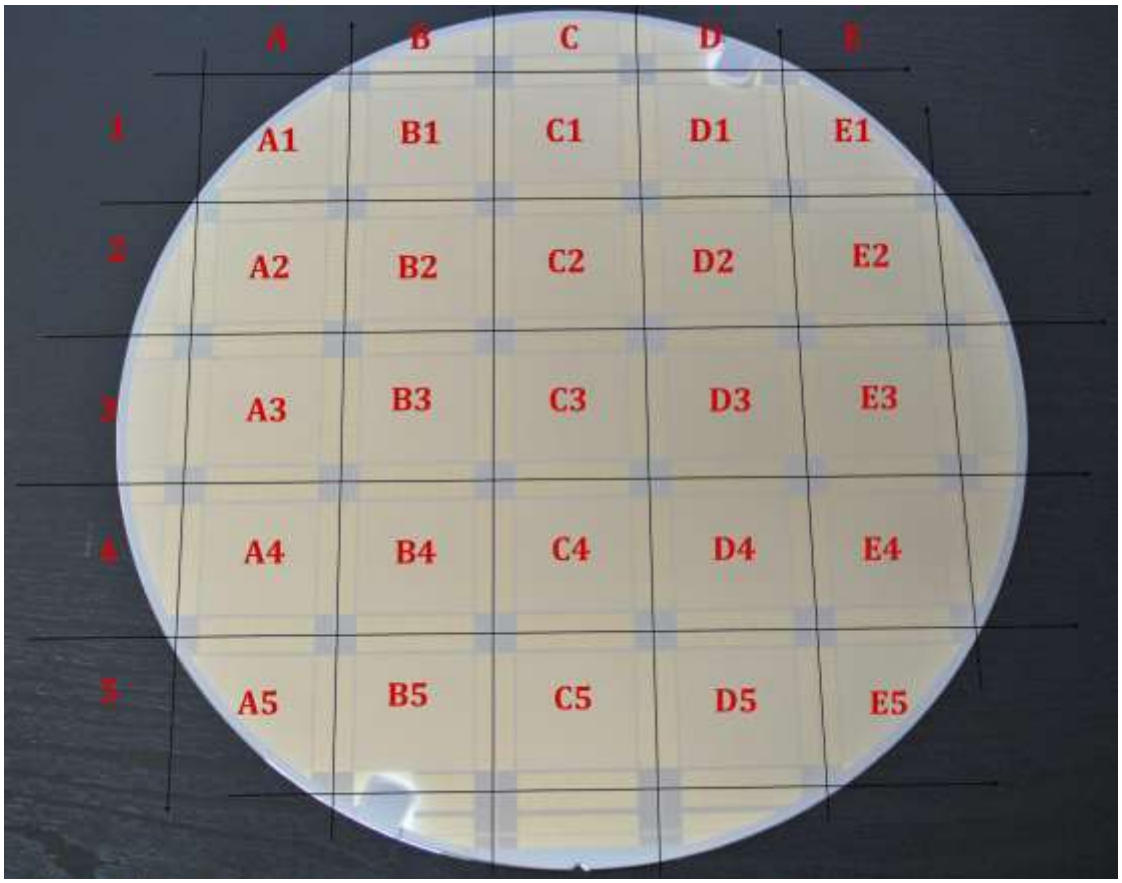


Figure 4.14: Wafer map used to identify RIICs by name

When testing in a bare RIIC on a wafer is really important to check all the power rails very carefully before applying power. The same power rail test is done in phase one is also done on phase two because a bigger RIIC has a slightly higher chance that growth defects happen shorting power rails together, in addition human error during placing the probe pins on the contact pads can kill a chip if power is applied. If the quadrant being probed passed the power rail test, it is then subject to sweeping tests of all its pixels. The same sweeping tests that were developed on phase one are used against all the different type of drive transistors in the pixels, the

difference is that on the second phase the number of curves is much bigger. Usually not all of the pixels in a quadrant are tested, a special semi-random distribution map is used. This map is generated beforehand and it is used all the quadrants, the way it works is that for every group of ten pixels only one pixel is randomly selected and tested. This approach makes the process much faster. Figures 4.15 and 4.16 show the graphical representation of the raw data collected from transistors of the same quadrant for TCSA and NSLEDS respectively. The graphical representation of all the transistor curves is a good metric to start the selection process for which RIICs perform better, however it is not very specific and if there are bad curves as the one showed in figure 4.17 then it is not possible to know which pixels failed by just a visual test. All the data that is collected from the RIICs is stored and later analyzed, the next chapter is going to explain how that process works.

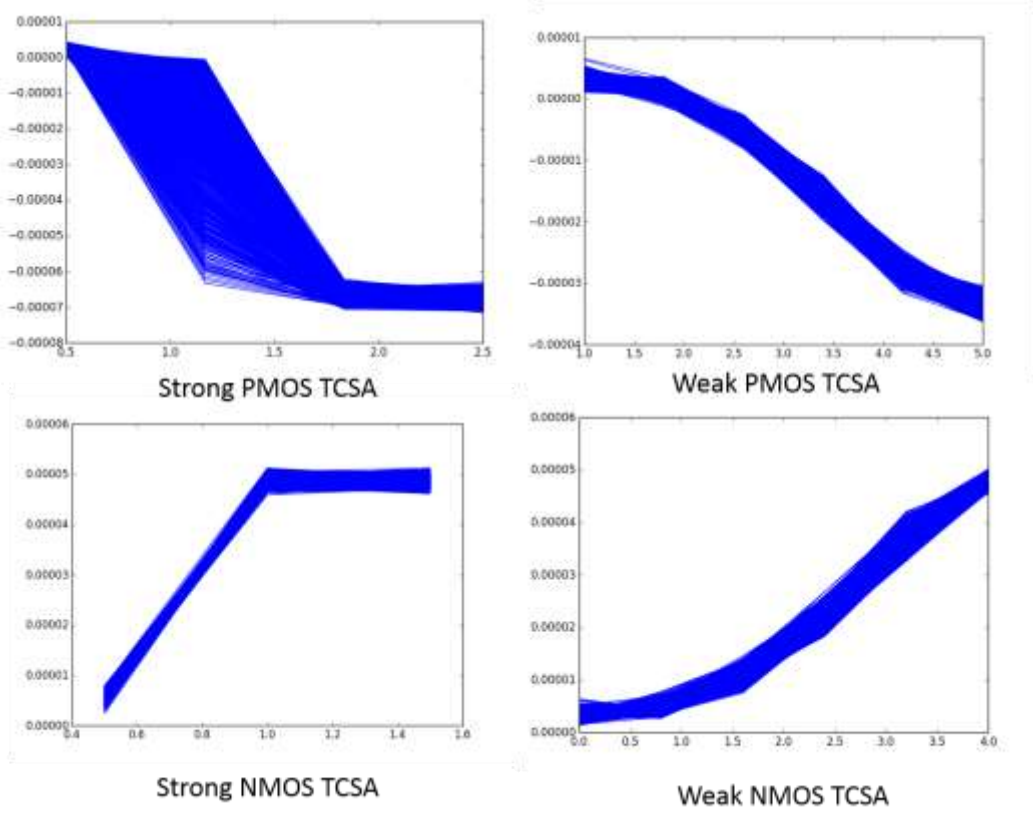


Figure 4.15: TCSA curves of raw data collected from one quadrant

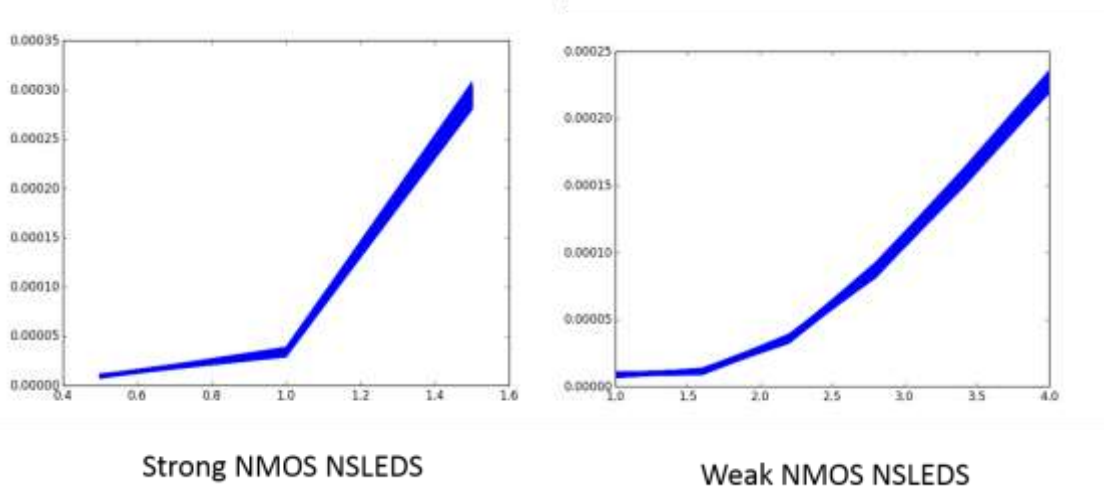


Figure 4.16: NSLEDS curves of raw data collected from a quadrant

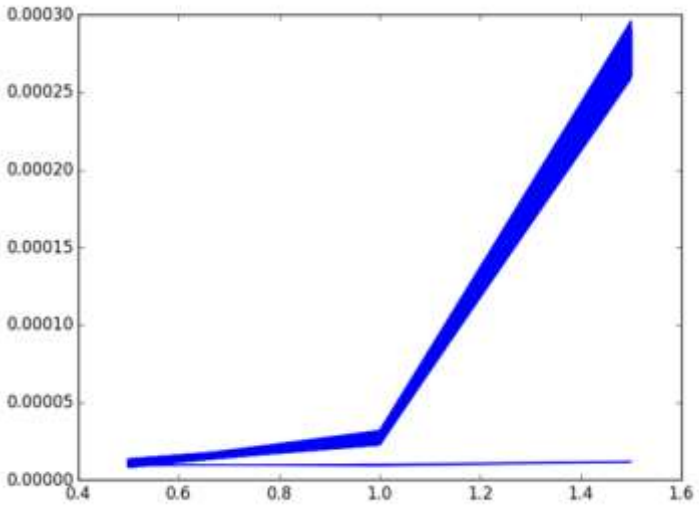


Figure 4.17: NSLEDS sample curves with bad pixels

.

Chapter 5

DATA ANALYSIS

5.1 Data Structures

During data collection on the second phase of testing there are some RIICs that fail during the initial power test due to yield problems. Others are disqualified to be good RIICs because by looking at the graphical representation of the raw data it can be determine that the transistors do not have the desired behavior. At the end, the yield of working RIICs per wafer is around 4 to 8 out of 19 complete RIICs. From those RIICs millions of data points are collected and saved for analysis. All the data is logged as npy files, a special type of file from the NumPy library which has many powerful tools for handling multidimensional array objects [10]. In addition the files are name in the following manner: RIIC type, wafer number, RIIC name, quadrant tested, transistor tested, test type, a time stamp and comments if needed.

The first thing to do with all the raw data is to organize it into manageable structure. To do this first the data is separated by wafer number. For example when the NSLEDS wafers were tested there were 3 of them, thus they are separated into three directories called w1, w2, and w3. Then a python script is run that creates a data structure similar to the one showed in figure 5.1. This structure is based on python dictionaries, the highest level dictionary is at the wafer level, it has keys that point to all of the functional RIICs in the wafer, the NSLEDS wafer 2 had four functional RIICs, B1, D5, E2 and C5. These RIIC names are also dictionaries that where each contains four keys that point to the four quadrants, “North-West”, “North-East”, “South-West”, and “South-East”. And for the last level each of the quadrants contain two keys which point to the data collected from the strong transistors and

weak transistor in said quadrant. This tree data structure makes it easier to access data faster because keys are used rather iterations.

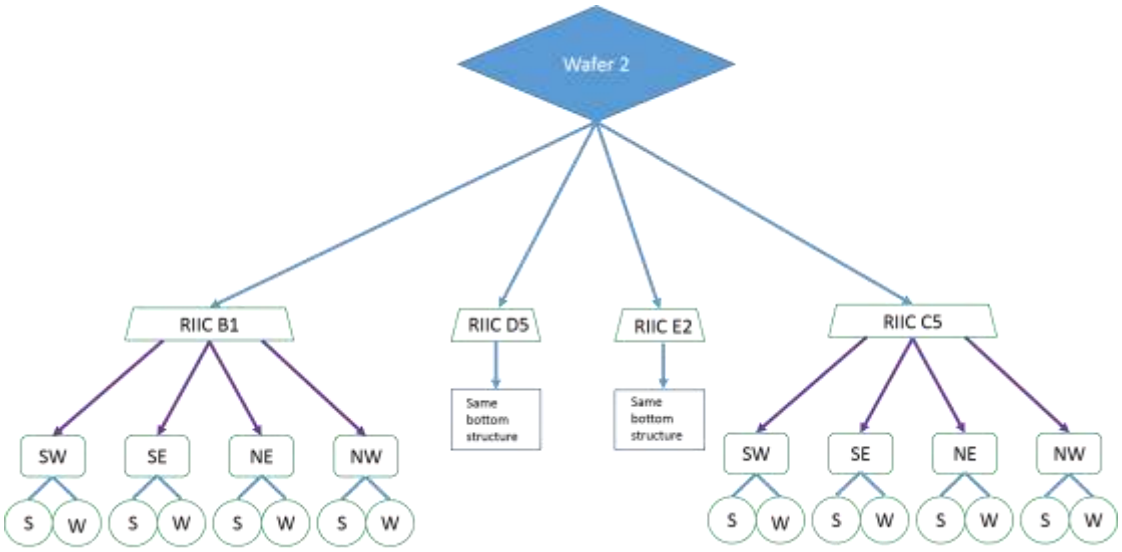


Figure 5.1: Data structure used for sorting raw RIIC data

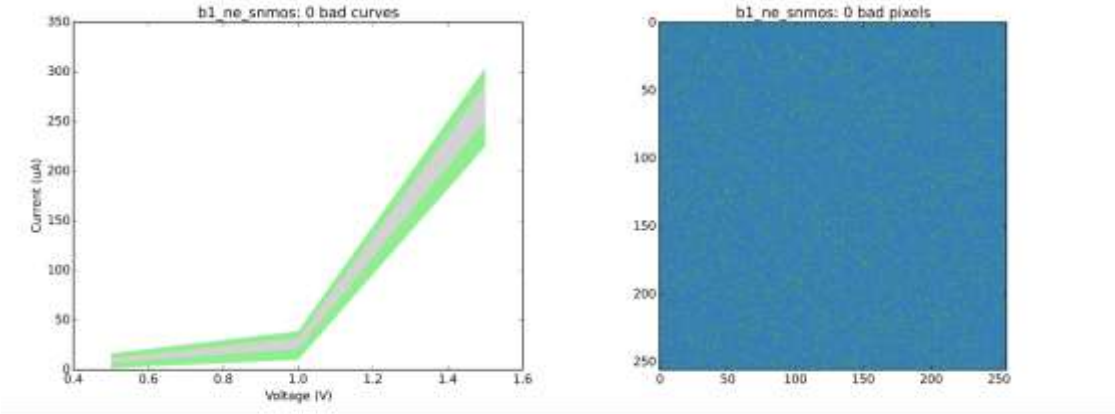
5.2 Analysis of Data

Once the data has been organized into a manageable data structure it is time to look at all the data points and turn them into something meaningful. The goal of data analysis is to determine whether a pixel curve is “good” or “bad.” The designation of “good” or “bad” is a qualitative measurement of whether the pixel will be useful in a projector application. The first step to determine whether a pixel is “good” or “bad” is to load a particular RIIC’s curves into an array and sort them to find the median curve. The reason why the median curve is used rather than the mean is because the median is outlier independent. Outlier independence means that extreme values do not affect the median or skew the view of the data. A constant distance from the median is use to determine range for which a curve is considered “good.” Which brings the question,

how is this range for good calculated? First a set of standard deviations was calculated from the first TCSA RIIC, unfortunately due to process variations it was impossible to correlate the standard deviation of one RIIC to another. However, that original standard deviation proved to be a useful quantum (q) for determining a good range. By visual inspection the range was set to be ±3q for data points of weak transistors at 0V. As the voltage values increase there is more variance on curves, thus the multiplicative factor increased uniformly to a value of ±17q. Similarly the range for good curves for strong transistors was set to ±6q at 0V and ±50q at 5V.

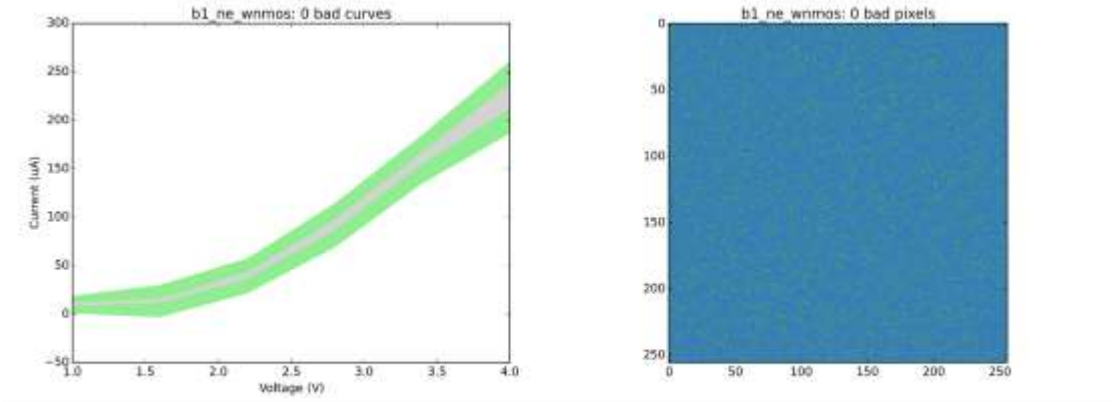
Once the range for determining "good" pixels has been establish curves can be analyzed. For every curve in a quadrant the distance between it and the median is calculated, and then it is check to see if the data points lie inside the range. Moreover two dictionaries are created, one holds the indices of all "good" transistor curves and the other holds the indices for all the bad transistor curves. Once all the curves have been analyzed they are plotted along with a map that shows the location of bad pixels if any. Figures 5.2 to 5.9 show the visual representation produced by the data analysis code. The images represent the analyzed data of all quadrants of RIIC B1of NSLEDS wafer 2 for both weak and strong drive transistors. For every set of curves, a map of pixels is created which plots all the pixels that were tested on their corresponding location. A green dot represents a pixel for which a drive transistor was tested and considered good. A yellow dot on the other hand represents a pixel were the transistor curves are bad, thus making it a semi functional or dead pixel. B1 has three bad pixels but they are not completely dead. The North West map for the strong transistor shows two bad pixels, however the same quadrant map for the weak transistor has good

curves for those pixels. Similarly, the South West quadrant has a bad weak transistor but the strong transistor works just fine in that pixel.



Strong NMOS curves , North East quadrant

Figure 5.2: Strong NMOS curves, North East quadrant



Weak NMOS curves , North East quadrant

Figure 5.3: Weak NMOS curves, North East quadrant

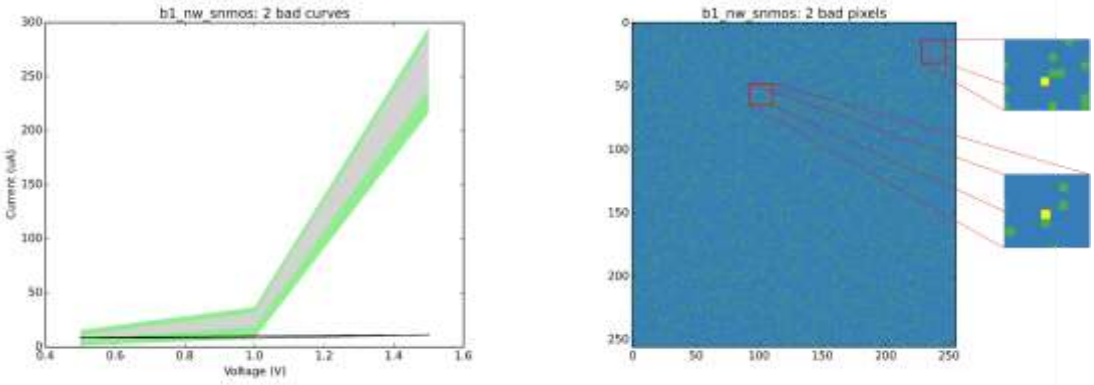


Figure 5.4: Strong NMOS curves, North West quadrant

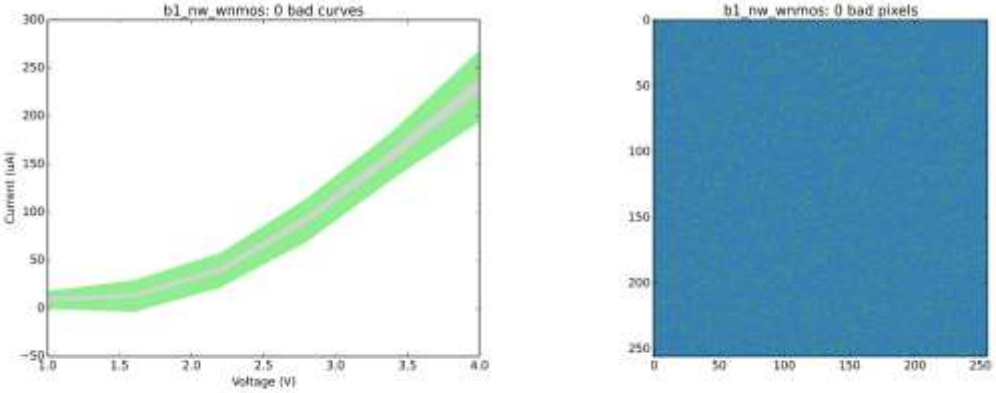


Figure 5.5: Weak NMOS curves, North West quadrant

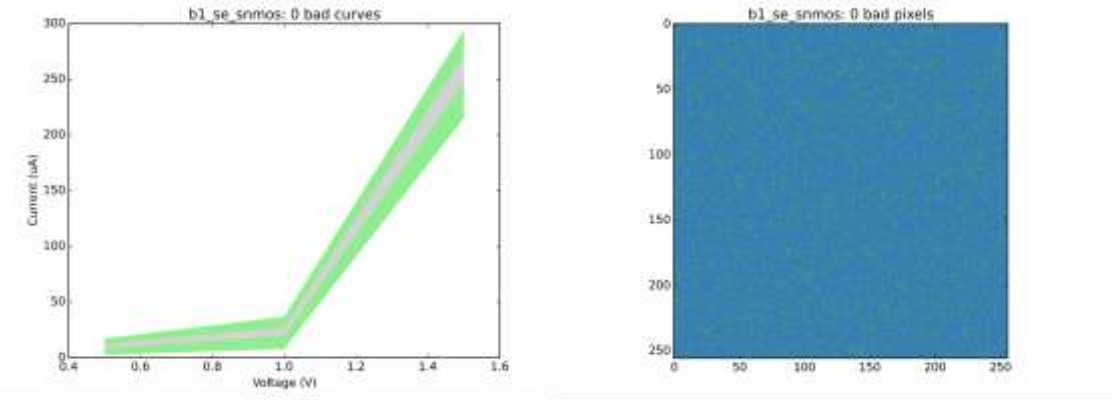


Figure 5.6: Strong NMOS curves, South East quadrant

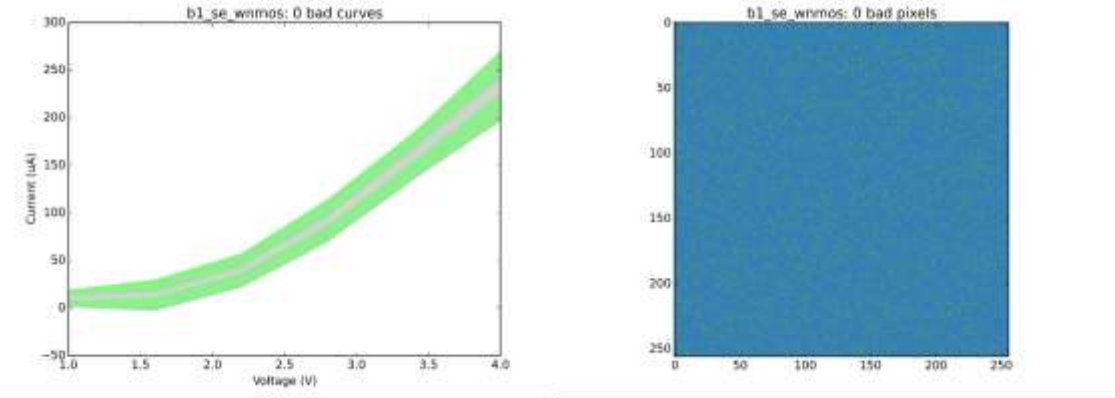


Figure 5.7: Weak NMOS curves, South East quadrant

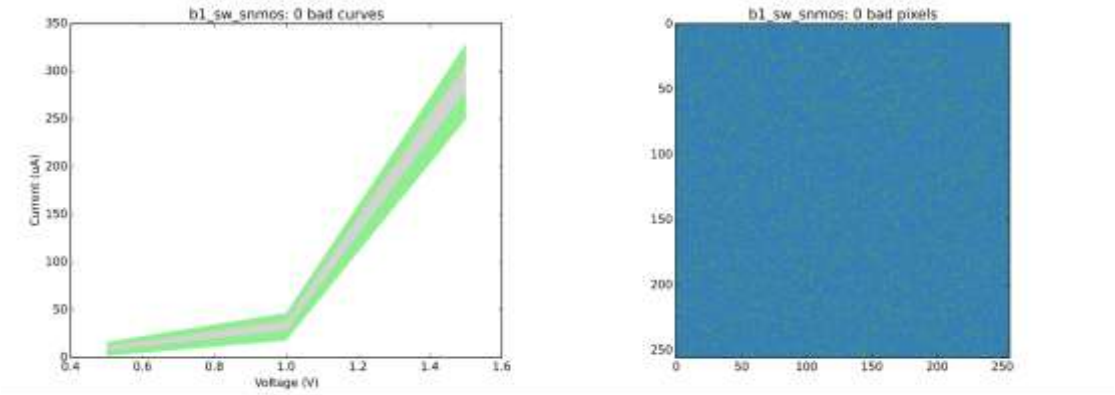


Figure 5.8: Strong NMOS curves, South West quadrant

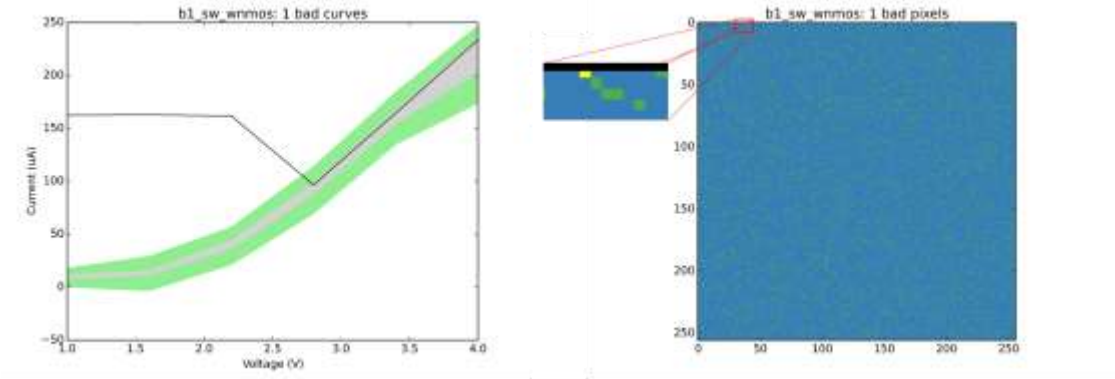


Figure 5.9: Weak NMOS curves, South West quadrant

5.3 Results

Once all the good RIICs in a wafer have been identified and analyzed the next step is to compare the results to figure out which ones are the best performing RIICs. For the example of wafer 2 of NSLEDS there were four fully working RIICs and after analyzing the raw data it was determine that three of those were optimal RIICs which can be turned into hybrids. What determines an optimal RIIC is the uniformity of the

transistor curves, the less variance in them means the more uniform pixel will light up. Another thing is the current values, higher current means that the drive transistors are more efficient. The last thing to consider is the number of bad pixels, this number has to be as small as possible, and it is very hard to find a RIIC for which the number of bad pixels is zero. Moreover, the location of the bad pixels is also very important as they need to be randomly distributed on the RIIC. For instance, if a RIIC has ten bad pixels all next to each other when projecting images on the projector there will be a noticeable spot where nothing is happening. On the other hand, if those pixels are isolated it would not be a problem as working pixels around it will cover it up.

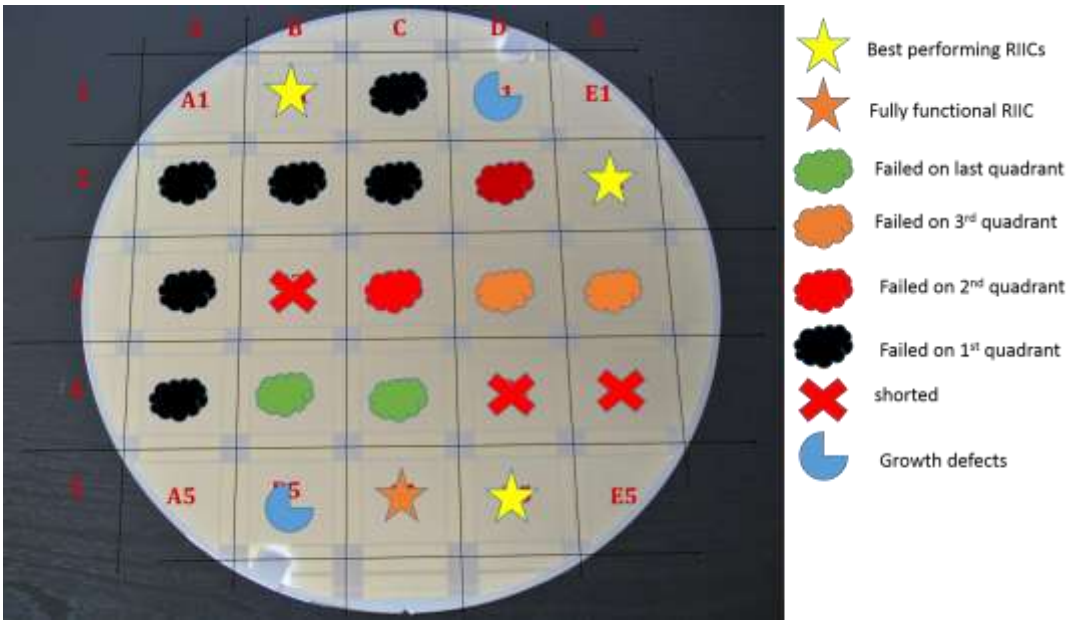


Figure 5.10: RIIC map showing conclusions of testing

For the NSLEDS example being discussed the map in figure 5.10 was created showing the conclusion after testing an analysis has been done. The three yellow stars

are those RIICs which meet the criteria to be considered for hybridization. The other fully functional RIIC, marked with the orange star, had a lower current output from the drive transistors. Following the convention that RIICs are tested one quadrant at a time, the clouds show all RIICs that failed at different stages of testing. The most common reason for the failure of those RIICs was that one or both drive transistors performed very poorly on the quadrants at which they failed. Lastly those marked with the red "x" and the blue semicircle, never had their pixels tested because the power rails were sorted and the RIICs were not fully grown due to the wafer holders used during processing. The same analysis process was used for all the TCSA wafers and the other two NSLEDS wafers.

Chapter 6

CONCLUSION

The TCSA RIIC was the first to be tested using a modular testing platform and throughout the different stages of testing TCSA the testing platform was improved to the stage where it is today. Now it is very easy to test and collect data from any version RIIC. As of writing of this paper the TCSA and NSLEDS full size RIICs have been fully tested and analyzed and the best performing RIICs have been sent off to be wire bonded and mated with the IR LEDs. HDILED RIICs are the next set of wafers to be tested in the near future and everything is already set for the process. Testing the RIICs is a very crucial element in the development of an IRSP because it is very important to have a state of the art hybrid. The hybrid is the one element that will interact directly with the hardware that will use our IRSP to test and calibrate their IR sensors.

In 2014 our team created the first IR LED projector which has been subject to many hours of intense testing at different laboratories in the United States. We have learned many valuable lessons from our first projector which are being applied to ensure the next generation of IRSP is much better than the first. Our TCSA projector is nearing completion and NSLEDS and HDILED will shortly follow.

Testing is only a portion of the whole project, there are many parts working in parallel with each other. Developing a system to drive the RIIC requires special hardware to be developed, such as DAC boards, amplifier board and Interface boards that bring all together. In addition, the firmware development is also very important because it dictates the autonomy of the system. Non uniformity correction algorithms are also being developed to ensure that all pixels behave uniformly on the RIIC. These

are just some of the biggest areas of work in the system, there are many more tasks involved to develop this technology. In the end everything comes together in a harmonious way to create the IRSP.

REFERENCES

1. Jacob Benedict, “Design and characterization of a mixed-signal pcb for digital-to-analog conversion in a modular and scalable infrared scene projector,” Master’s Thesis, University of Delaware, Department of Electrical and Computer Engineering, June 2015.

2. K. Nabha, “100 Hz 512x512 SLEDS system design,” Master’s Thesis, University of Delaware, Department of Electrical and Computer Engineering, June 2014.

3. “Scalable testing platform for CMOS Read-In Integrated Circuits”, M. Hernandez, J. Dickason, P. Barakhshan, N. Waite, R. McGee, F. Kiamilev, GOMAC Tech Conference, Orlando FL, March 2016.

4. “Thermal Performance Characterization of a 512x512 MWIR SLEDS Projector”, P. Barakhshan, G. Ejzak, K. Nabha, J. Lawler, F. Kiamilev, GOMAC Tech Conference, Orlando FL, March 2016.

5. “512x512, 100Hz Mid-wave Infrared LED Scene Projector System”, G Ejzak, J. Marks, J. Dickason, N. Waite, J. Benedict, M. Hernandez, S. Provence, D. Norton, J. Prineas, K. Goossen, F. Kiamilev, T. Boggess, 2016, Unpublished manuscript.

6. “512x512, Two-color Infrared LED Scene Projector”, J. Benedict, R. McGee, J. Marks, K. Nabha, N. Waite, G. Ejzak, J. Dickason, H. Ahmed, M. Hernandez, P. Barakhshan, T. Browning, J. Volz, T.Lassiter, K. Black, F. Kiamilev, T. Boggess, GOMAC Tech Conference, Orlando FL, March 2016.

7. “Read-In Intregrated Circuits for Large-format Multi-chip Emitter Arrays,” J. Marks, F. Kiamilev, N. Waite, R. Mcgee, GOMAC Tech Conference, St. Louis, MO, March 2015.

8. “512x512 Individually Addressable MWIR LED Arrays Based on Type-II InAs/GaSb Superlattices,” Dennis T. Norton, Jonathon T. Olesberg, Rodney T. McGee, Nick Waite, Jonathan Dickason, K.W. Goossen, John Lawler, Gerry Sullivan, Amal Ikhlassi, Fouad Kiamilev, Edwin J. Koerperick, Thomas F. Boggess, IEEE J. Quantum Electron. 49, 753 (2013).

9. FLIR OEM Modules & Components Website. http://www.flir.com/cores/display/?id=51946, April 2016.

10. NumPy Website. http://www.numpy.org, April 2016.

Appendix A

SPI REFERENCE TABLE FOR RIICS

		Function of the registers by RIIC model		
register number	name	TCSA	NSLEDS	HDILED
0	reset	resets to HIGH, when spimode = 1 drives array reset signal	same as TCSA	same as TCSA
1	spimode	operation mode,RIIC is criven from SPI register	same as TCSA	same as TCSA
2	dacmode0	operation mode, RIIC uses 32 DACs (parallel write to hex pixels)	same as TCSA	same as TCSA
3	dacmode1	operation mode, RIIC uses 2 DACs (serial write to single pixel)	same as TCSA	same as TCSA
4	dacmode2	operation mode, RIIC uses 4 DACs (parallel write to double pixels)	same as TCSA	same as TCSA
5	dacmode4	operation mode , RIIC uses 8 DACs (parallel write to quad pixels)	same as TCSA	same as TCSA
6	dacmode8	operation mode , RIIC uses 16 DACs (parallel write to octo pixels)	same as TCSA	same as TCSA
7	mselp	when set MOUT monitors pixel PMOS drain voltage	not used, NLEDS uses NMOS drive trasistors	same as NLEDS
8 to 15	addrx<0,7>	when spimode = 1, controls pixel X addressing	same as TCSA	same as TCSA

16 to 23	addry<0,7>	when spimode = 1, controls pixel Y addressing	same as TCSA	same as TCSA
24	load	not used	multiplexer signal , used to chose a subpixel from a superpixel	same as NLEDS
25	men	enables monitor out pin MOUT, this way pixels can be monitored for various things	same as TCSA	same as TCSA
26	prechg	precharge signal derived from wen signal	same as TCSA	same as TCSA
27	sels	select strong, when HIGH uses the strong trasistor to drive the pixel, LOW uses the weak	same as TCSA	same as TCSA
28 to 30	vinn<0,2>	3 bit internal DAC (0 to 5 volts), drives NMOS drive trasistor , 000 = 5V, 111 = 0V	removed from design	same as NLEDS
31 to 33	vinp<0,2>	3 bit internal DAC (0 to 5 volts), drives PMOS drive trasistor , 000 = 5V, 111 = 0V	same as TCSA, but now it drives a NMOS drive trasistor	same as NLEDS
34	mseln	MOUT monitors pixel NMOS drain voltage (SLEDn current)	same as TCSA	same as TCSA
35	mselvinp	MOUT monitors PMOS on chip DAC voltage	same as TCSA	same as TCSA
36	mselvinn	MOUT monitors NMOS on chip DAC voltage	same as TCSA	same as TCSA
37	mselibias	MOUT fucntions as an input sink for 1uA of current	same as TCSA	same as TCSA

38 to 40	pixelibias<0,2>	default "010", drives the in-pixel source followe current DAC, "000"= 0uA, "001" = 0.5uA, "010" = 1 uA, "100" = 2 uA	same as TCSA	

Table A.1: Explanation of the configuration bits of the SPI registers