May1625

Latent Damage and Reliability in Semiconductor Devices

FINAL REPORT

SEAN SANTELLA, HAYLE OLSON, DAVID ACKERMAN, JAEHYUK HAN

Advisor & Client: Dr. Randall Geiger, ECpE

Tuesday, April 26, 2016

Table of Contents

1. REVISED PROJECT DESIGN **3**
1.1 ABSTRACT 3
1.2 HYPOTHESIS 3
1.3 BLOCK DIAGRAMS 3
2. IMPLEMENTATION DETAILS **5**
2.1 BURN-IN PCB 5
2.1.1 Functionality *5*
2.1.2 Schematic *5*
2.1.3 Truth Table *6*
2.1.4 Test Mode (Functionality Check) *7*
2.1.5 High Stress Mode (Burn-in) *7*
2.2 ESD STRESS PCB 8
2.2.1 ESD Stress PCB *8*
2.2.2 ESD Stress Procedure *9*
3. TESTING PROCESS AND TESTING RESULTS **10**
3.1 SUMMARY 10
3.2 TEST EQUIPMENT 10
3.2.1 ESD Stress PCB & DUT burn-in PCB *10*
3.2.2 Burn-in Oven *10*
3.3 TESTING PROCEDURE 11
3.3.1 ESD Stress *11*
3.3.2 Accelerated Lifetime *12*
3.3.3 Pass & Fail Conditions *12*
3.3.4 Safety Concerns *12*
4. APPENDIX I: OPERATION **13**
4.1 SET-UP 13
4.1.1 Power Supplies *13*
4.1.2 Insert DUTs *14*
4.1.3 Functionality Check the DUTs *14*
4.1.4 Activate the Burn-in Oven *14*
4.2 TESTING AND DATA COLLECTION 14
4.2.1 Begin Burn-in *14*
4.2.2 Gather Data *14*
4.3 ANALYSIS OF RESULTS 15
4.3.1 Data Analysis *15*
4.3.2 Draw a Conclusion *15*
5. APPENDIX II: PREVIOUS VERSIONS **16**
5.1 INTRODUCTION 16

5.2 Original Burn-in PCB's 16
5.2.1 Initial Idea *16*
5.2.2 Identifying Issues *18*
5.2.3 Troubleshooting Issues *19*
5.2.4 Our Solution *20*
5.3 The Second Burn-in Design 20
6. APPENDIX III: OTHER CONSIDERATIONS 22
6.1 Probing the Original Burn-in PCB 22
6.2 Advantages & Disadvantages 23
6.3 Budget 23

1. Revised Project Design

This final report includes a detailed description of our Senior Design Project that was completed during the Fall 2015 and Spring 2016 semesters at Iowa State University. We will be discussing the implementations that have been created, how we plan run the testing, the exact operation of all implementations, and briefly outlining the previous implementations that led us to our final product.

1.1 Abstract

Latent damage due to an Electrostatic Discharge (ESD) event is a topic of debate within the Semiconductor industry. Our goal, as a team, is to research if latent damage does or does not exist within semiconductor devices after experiencing an ESD event. Our next task is to conclude whether these devices are reliable after proving latent damage exists.

1.2 Hypothesis

We expect that if an ESD event occurs on a semiconductor device, then latent damage exists. Furthermore, this latent damage can cause the reliability of these devices to decrease; resulting in the Mean Time to Failure (MTTF) to be substantially shorter than the manufacturing specification (20 years).

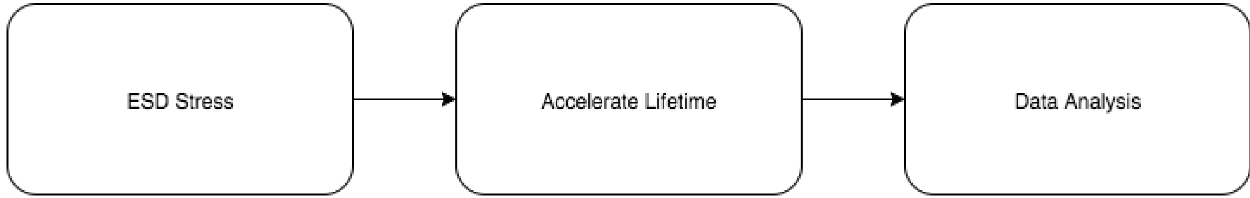
1.3 Block Diagrams

On the next page, *Figure 1* shows the different system level block diagrams that our project encompasses. There are three distinct sections to our project: ESD Stress, Accelerate Lifetime, and Data Analysis.

Within the ESD Stress block there are two sub-blocks: Stress DUTs and Check Functionality. Approximately 200 devices will be stressed by simulating an ESD event. Then, the functionality of each device will be checked and recorded; we are shooting for a 50% failure rate.

Following ESD Stress, the Accelerate Lifetime block contains three sub-blocks: Test 50% of Stressed DUTs, Burn-in DUTs, and Check Functionality. The devices that passed the first functionality check will be used as our sample (approximately 50%). Next, the burn-in oven will be used to accelerate their lifetime. While in the burn-in oven, functionality will be checked at a desired interval. We will continue to check the functionality until 100% of the devices have failed.

Lastly, after all of the devices have failed and the burn-in step has been completed, we will move onto our last block, Data Analysis. We will be performing statistical calculations and hopefully verifying our original hypothesis.



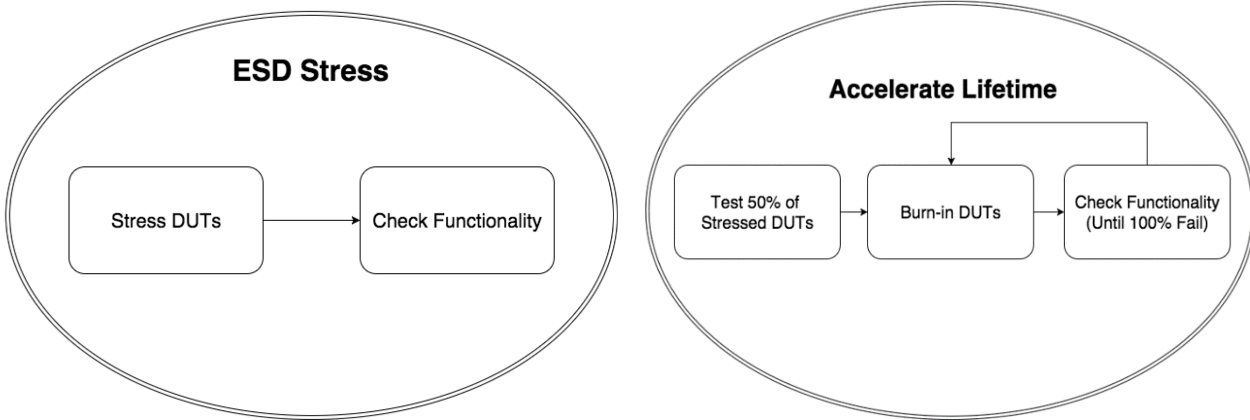


Figure 1: System Level Block Diagrams

2. Implementation Details

2.1 Burn-in PCB

2.1.1 Functionality

The burn-in Printed Circuit Boards have two functionalities/operation modes built into them:

1. DUT functionality testing
2. DUT high-stress burn-in

The functionality check allows the user to periodically test the DUT to see if the device is still working and functioning as expected. The high-stress burn-in mode forces the DUT into a state where both transistors are stressed simultaneously.

Each board was designed to stress and check 10 DUTs individually. The set-up for each individual DUT is referred to as a "testing cluster" within this appendix.

2.1.2 Schematic

Below you will find a schematic for a single testing cluster (*Figure 2*). This specific schematic allows the user to test functionality of the inverters and visual see which transistor failed when a DUT has a failure.

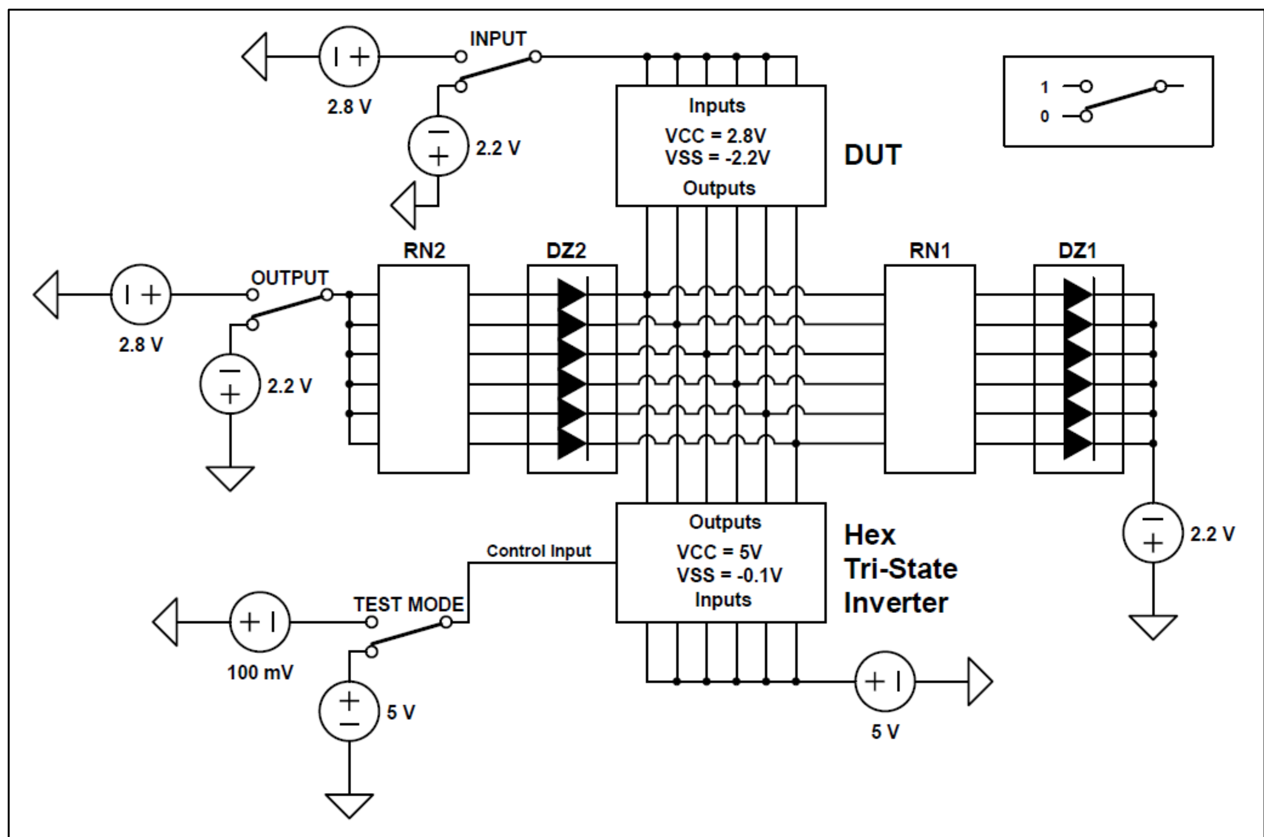


Figure 2: Schematic of a single testing cluster on the Burn-in PCB

RN# and DZ# are resistor and LED arrays, respectively. They act as a visual indicator that can be quickly assessed by simple eyesight; no multimeter measurements are necessary to check the logic levels of devices. The LEDs are in series with current limiting resistor arrays for protection.

The tri-state inverters included in our schematic allow for us to electrically switch between two operation modes (Test mode and High Stress mode).

The user should note the 6 individual output busses in the center of the above schematic. To completely understand the concept behind the design of the circuit, it's important that the user comprehends the central bus structure, including the left-most (left of RN2) and right-most (right of DZ1) nodes. Think of this line as a river flowing from the left side to the right, and controlled by the DUT and the tri-state inverter.

All 10 of the testing clusters on a board are connected to the same centralized switches and sources. This will result in all of the testing clusters behaving identically when the DUTs are functioning. The various voltage levels presented in the schematic above are explained in a later section.

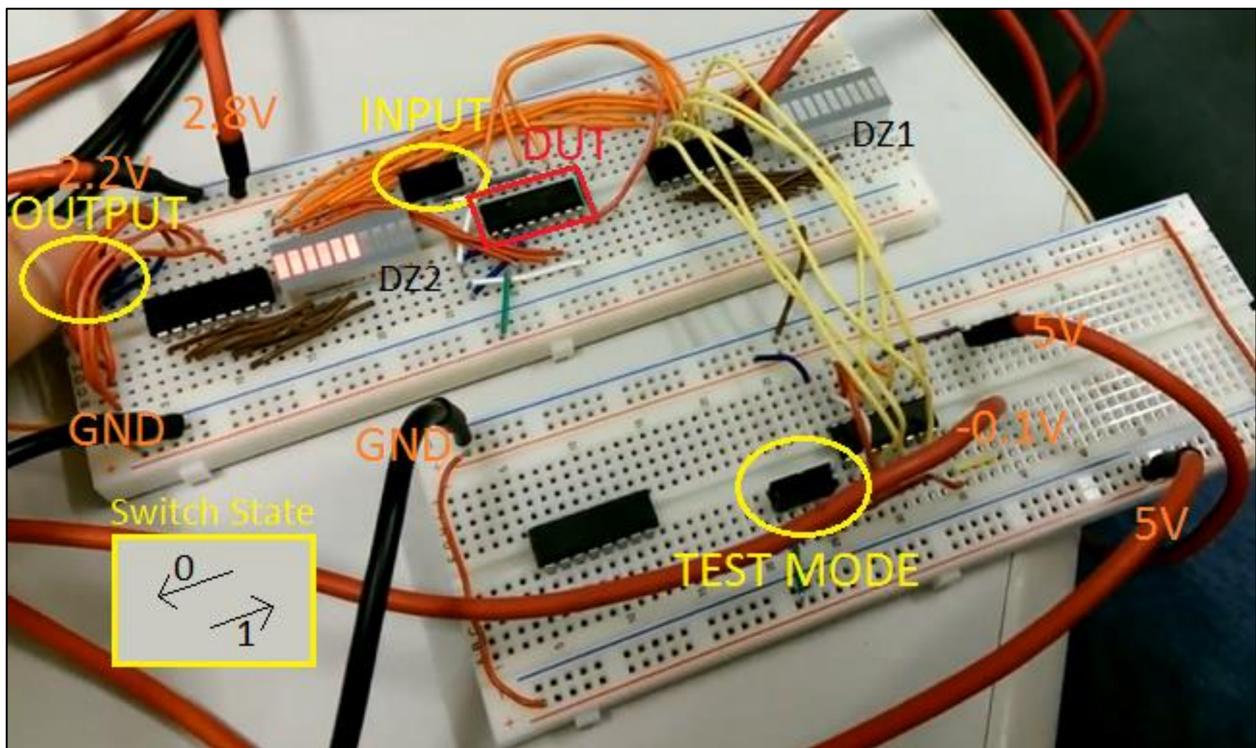


Figure 3: Breadboard Implementation of the Burn-in PCB

2.1.3 Truth Table

Switch		Diodes		Transistors	
"Output"	"Input"	DZ2	DZ1	PMOS	NMOS
0	0	0	1	1	0
0	1	0	0	0	1
1	0	0	1	1	0
1	1	1	0	0	1

Figure 4: Truth Table for Logical Operation of a Functionality Check

In the above table, “1” represents the high voltage for the switch states, lit for the diode arrays, and conducting for the transistors. “0” represents the compliment of each state (low, not lit, and not conducting, respectively). In order to check the functionality in the table above, the test mode must be in operation.

Note that the table has been arranged to represent the physical placement of the components on the breadboard.

2.1.4 Test Mode (Functionality Check)

The first mode of operation is the test mode, which allows us to disconnect the output of the device from the circuitry.

As mentioned previously the use of LED arrays allows us to quickly check the functionality of our DUTs. If any one of the LEDs in an array are lit the DUT is considered no longer functional.

Functionality testing is achieved when the “test mode” switch is set to high voltage, and high-stress burn-in mode is accomplished with the low voltage state.

2.1.5 High Stress Mode (Burn-in)

The high-stress mode works by forcing the output gates of each DUT to the trip point, at which both of the transistors are conducting simultaneously. By doing this, both transistors are evenly stressed during the burn-in, allowing for data to be fairly collected for each failed transistor.

The voltage levels attached to each device, as shown in *Figure 2*, allow for the circuit to accomplish this task. The DUTs were measured via oscilloscope to determine their trip point. In order to make the implementation easier, a high and low voltage level were experimentally found. This allowed us to set the DUTs trip point at ground.

Such experimentation revealed that $V_{CC,\ DUT}$ = 2.8V_{DC} and $V_{SS,\ DUT}$ = -2.2V_{DC} met our requirements, for keeping the voltage difference at 5V_{DC}. Having 5 volts across each DUT is necessary to continue operating the device within the specifications outlined by datasheet.

Because the V_{CC} and V_{SS} connections determine what voltage levels qualify as “high” and “low” from a binary viewpoint, the rest of the circuit interfacing with the DUT needed to account for this. Thus, the “Input” and “Output” switches are attached to the voltage levels mentioned. The connections can be seen up in *Figure 3*.

For the tri-states, 5V_{DC} to -0.1V_{DC} is used to accomplish the trip-point forcing requirement. By attaching the outputs of both the DUT and the tri-state inverters to the output busses of the circuit, that node is under contention when the operation mode is set to high-stress mode. As a direct result, the DUT tries to pull the output high, while the tri-state tries to pull it low. By setting $V_{SS,\ Tri}$ to about -0.1V, the output bus is forced to roughly ground, the trip point of the DUT.

The choice to directly set the input gates of the hex tri-state inverters to 5V_{DC} is rather straight-forward. Our circuitry only has two desired modes of operation which was achieved by connecting the Control Inputs of the tri-state inverters (pin 1 and pin 15) together. Connecting these two pins allows us to switch between a high-impedance output and an output at the voltage of $V_{SS,\ Tri}$.

2.2 ESD Stress PCB

The ESD Stress System is the mechanism and procedure that our group will use to hopefully inhibit latent damage in the COTS semiconductor devices. This system contains the ESD Stress PCB and the ESD Stress Procedure.

2.2.1 ESD Stress PCB

The ESD Stress PCB will be used to stress CMOS hex inverters (CD4049UBE) manufactured by Texas Instruments using a bulk-CMOS process, which contain six individual inverters in each package.

Stressing the devices is based off of the Human Body Model (HBM). Our approach is to charge a 100pF capacitor with a couple of kilovolts (kV). This capacitor is charged by a high-voltage source. After the capacitor is completely charged, it can then be discharged by the flip of a switch into the DUT, causing the ESD event to occur.

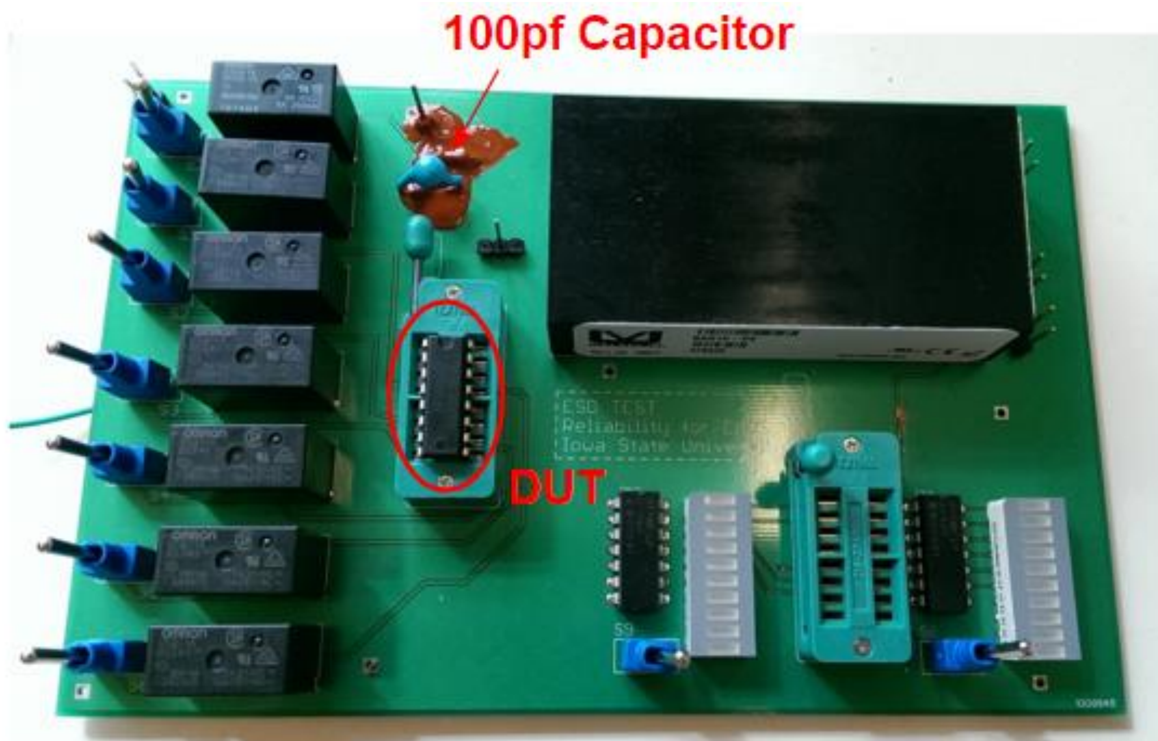


Figure 5: ESD Stress PCB

Our high-voltage source is intended for agricultural purposes, and as such, the cost is minimal. It's possible to obtain a programmable lab-grade high-voltage source, but due to the very low demand in the market, doing so would be expensive.

The output of our high-voltage source is a voltage of 4.92kV, but only for 1/4000 of every second. Therefore during the charging phase of the stressing procedure a set of diodes must be used to prevent any reverse current from flowing out of the capacitor when the output of the source is low. As always, there needs to be a resistor in series with these diodes to ensure that the voltage drop across each diode doesn't exceed the maximum specification of the devices. These diodes also must be able to

withstand a particularly high reverse voltage without experiencing breakdown. Our design has specified diodes that can withstand a reverse bias of 4kV. Four of these diodes are being used to safely prevent capacitance charge loss.

Furthermore, the voltage of our source is higher than needed, and as such the voltage must be stepped down through the use of a simple voltage divider. Doing this allows us to further control what size voltage is used to stress the devices.

LED arrays on the PCB also allow a stressed part to be checked for immediate failure. The voltage being applied to the DUT could very well cause catastrophic damage. The goal of our experiment is to check for latent damage, not catastrophic damage. As such, non-working devices will be immediately discarded instead of studied in accelerated lifetime testing.

A schematic of the ESD stressing circuitry is shown in *Figure 6*.

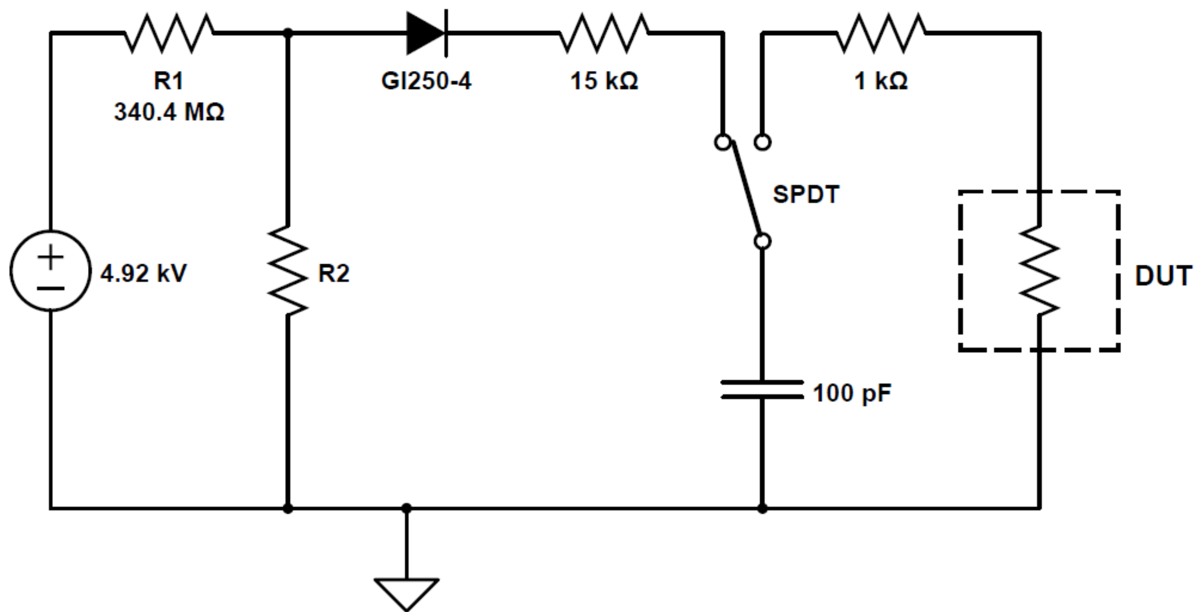


Figure 6: Schematic of ESD stressing circuitry

2.2.2 ESD Stress Procedure

The procedure by which to stress the devices can be partially understood through the previous section. However, for the usability of this document and the completeness of this section, it is detailed here.

The switch shown on the previous page is set to the high voltage side, to charge the capacitor to the desired voltage, set by the voltage divider's intentionally selected resistance values and the voltage drops across the diodes. This is the charging phase of the procedure.

During this charging phase, the device to be stressed is inserted into the board's stressing mechanism. Once the capacitor is given sufficient time to charge (which can also be determined through intelligent selection of resistor values) the switch is then flipped to discharge the capacitor through the device.

Once the device is “fully” discharged (given enough time for the RC circuit to mostly undergo most of its discharging), the device is then taken into the testing phase. Once again, the resistor values can be intelligently chosen to make the “sufficient time” short enough to cause a quick ESD event to occur.

The device is now stressed and must be tested before it can be used in our sample. Given the nature of the damage we are looking for, any device that has a failure cannot be used for our study. This testing phase is performed on the same PCB. A set of LED arrays are used to test each gate of the DUT.

This stressing procedure is further to be used to determine what level of voltage we should use to stress the devices. Different experimentally determined stress conditions have been suggested, one of which was to stress until roughly half of the stressed parts fail. Given this, the remaining parts can be tested for latent damage by accelerating their lifetime until failure.

3. Testing Process and Testing Results

3.1 Summary

Our team has a goal to design and run an experiment to show whether or not latent damage exists in COTS semiconductor devices after an ESD event has occurred. Specifically, our device of interest is a CMOS hex inverter (CD4049UBE) manufactured by Texas Instruments using a bulk-CMOS process.

Latent damage is the type of damage that cannot be measured through the devices electrical characteristics, but a physical defect is present and as a direct result, the device’s lifetime is reduced. This kind of phenomenon would mean that our COTS devices can have unforeseen reliability issues, which in turn could mean that present repair procedures (i.e. swapping out boards on a failed system until the functionality returns) for any system undergoing stress could be invalid.

Our primary interest in this project is to electrically stress a large sample of a COTS device by ESD and measure the failure time versus expected lifetime of the devices.

3.2 Test Equipment

3.2.1 ESD Stress PCB & DUT burn-in PCB

The ESD Stress PCB is designed to charge a capacitor via a high-voltage source, and then by flipping a switch, discharge that capacitor to the Device under Test (DUT). This will allow us to simulate an ESD event on a device.

The Burn-in PCB is used to monitor 10 devices at the same time while accelerating their lifetime in a burn-in oven. It has a set of LED arrays which provide output levels of the devices as well as a control switch to vary the input level of the devices.

3.2.2 Burn-in Oven

Under normal operating conditions, the expected lifetime for a COTS device is just around 20 years. To allow completion of this experiment in sufficient time, the lifetime of the COTS devices will need to be accelerated.

It is well known, a device under operation while at a particularly high temperature will experience an accelerated lifetime. To allow the experiment to be completed on the order of months, we will use a burn-in oven to accelerate our parts' remaining lifespan after the ESD stress.

From there, we can test the logical functionality of the devices and record how much longer they last *after* the ESD stress.

However, it should be noted that the existing burn-in boards have outstanding issues. As of right now, when powered, the boards treat an empty socket in the exact same way as a populated socket. Progress has been made toward resolving the PCB issues and will hopefully be resolved soon.

3.3 Testing Procedure

Much of the testing procedure has already been detailed. However, in the interest of completeness and usability of this document, our current plan for testing is detailed here.

3.3.1 ESD Stress

The first part of our experiment will involve taking a sample of 200 COTS devices and subjecting them to a high-voltage ESD event. This procedure will use the PCB described in **Section 2.2.1**. A capacitor will be charged with a high-voltage source, then discharged into the input gate of a hex inverter. This ESD stress procedure is repeated for each hex inverter until the Experimental Group has been created. To be clear, we want 50% of the devices to catastrophically fail and 50% of the devices to function after being stressed. The 100 stressed devices still working will be considered the experimental group (*Figure 7*).

During this type of ESD event, all outputs will be tied low (ground) to put the pMOS transistors in a high-current mode inducing a higher stress condition because of the raise in temperature.

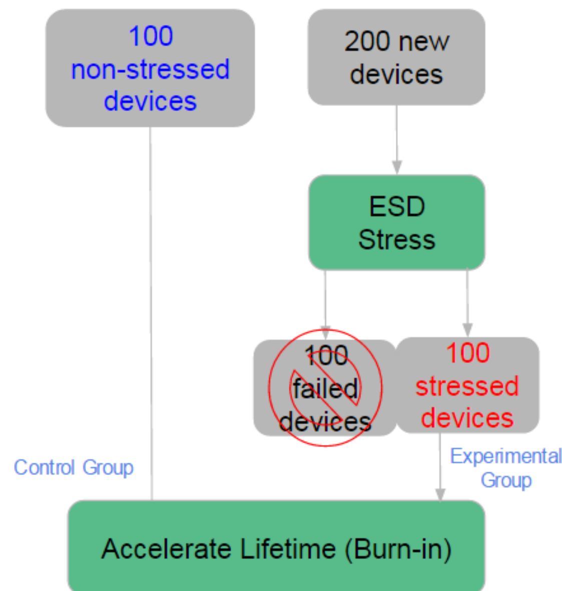


Figure 7: Control Group and Experimental Group Devices

After gathering the Control Group (brand-new devices) and the Experimental Group, send the devices to burn-in to accelerate the lifetime.

3.3.2 Accelerated Lifetime

Once the devices have been stressed, they will be inserted into a burn-in oven to accelerate their lifetime. As mentioned in **Section 2.2.2**, this will accelerate the overall lifetime of the devices to much shorter, manageable durations. Keep in mind that the devices will need to be powered-on and kept in the high-current mode during the accelerated lifetime testing. This process is generally called *burn-in*.

3.3.3 Pass & Fail Conditions

Our metric for determining failure of the device during burn-in will be through the logic levels. As each device contains multiple inverters, the status of the LED indicators on the DUT burn-in PCB should be opposite that of the shared input of the testing board. A failure will be considered when any of the inverters on a device have incorrect logic.

The truth table of a single inverter is shown in *Figure 8*. If at any time a single inverter has an incorrect logic level, then the DUT has failed. At which point, the device lifetime will be recorded for later analysis.

Figure 8

Input	Output
0	1
1	0

Given this, it will also be worth noting *how* the device failed. Is the device always giving a high output? Is the device always giving a low output? Or is it giving reverse operation? Which one of these three possible outcomes occurs for failed devices might also be worthwhile to study for our project. If a pattern emerges, it may suggest that the pull-up network (PUN) or pull-down network (PDN) of COTS devices are more vulnerable to ESD events and latent damage. If so, this may have further implications for future studies and preventative measures.

3.3.4 Safety Concerns

Safety during this experiment is definitely of great importance. The dangers to us during experimentation are the high temperatures of the burn-in oven and the high-voltages used to perform the ESD stress.

The current plan is to use a standalone, commercial device to produce the high-voltages. However, it can't be expected that this device will be able to output high amounts of *energy*. Still, the higher voltage needs to be considered and isolated from other conductive sources to prevent arcing.

The burn-in oven also could pose burn hazards, but not any more dangerous than a standard conventional oven. After placing boards in the burn-in oven for a desired amount of time, we will need to use heat-resistant mitts to handle the boards and devices.

4. Appendix I: Operation

4.1 Set-up

4.1.1 Power Supplies

Begin by attaching the appropriate voltage sources to the boards (note *Figure 9*).

Below is part of a computer-generated 3D preview of the burn-in PCB from NI Ultiboard. As you can see, there are *Power pins* on the far left indicated with the voltages (Figure 10) that should be attached to each header. The diodes (DZ#), resistor networks (RN#), devices under test (DUT), and hex tri-state inverters (Tri-State) are all labeled correspondingly.

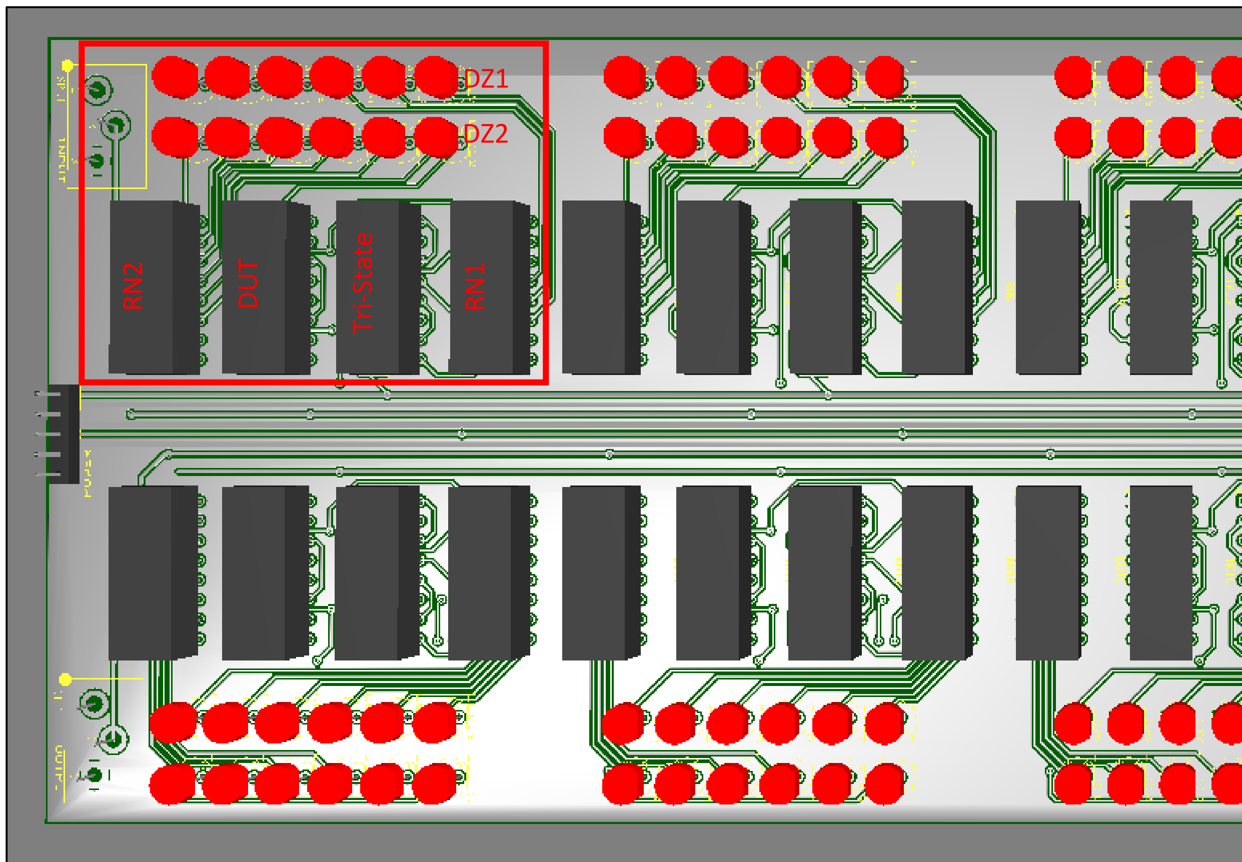


Figure 9: 3D Preview of the Burn-in PCB – one testing cluster is outlined

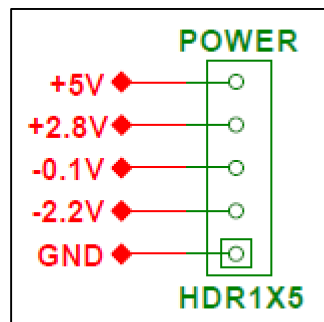


Figure 10: Power Pin connections

4.1.2 Insert DUTs

Place the stressed DUTs into the sockets on the burn-in PCB, which should look very similar to the one shown here.

4.1.3 Functionality Check the DUTs

Ensure that the DUTs are working before beginning. Set the operation mode switch to its high state (Test Mode). Toggle the states of the “Output” and “Input” switches and check the DUT according to the truth table (*Figure 4*) in **Section 2.1.3**.

If all diodes within the array meet the logic levels, then the DUT is working. If all DUTs are working properly, move onto the next step.

Note: If at any time both diode arrays for a testing cluster are active, then the incorrect mode of operation has been selected. Toggle the operation mode switch to correct this.

4.1.4 Activate the Burn-in Oven

Turn on the burn-in oven. Set the temperate to 112 °C (as per design calculations). Verify that the oven is heating.

4.2 Testing and Data Collection

4.2.1 Begin Burn-in

Set the switch states to 0, 0. Then set the operation mode to high stress mode (switch state should be low.) Both diode arrays should be lit in the clusters.

Place burn-in boards into burn-in oven. Check the LED arrays to verify that the power supplies are properly connected.

4.2.2 Gather Data

Periodically check the functionality of the devices:

1) Remove devices from burn-in oven. Be cautious as PCBs and devices will be hot.
2) **Change the operation mode to the test mode to check functionality**.
3) Perform the functionality check, as described above in **Section 4.1.3**.
4) Record any failed devices and time of failure.
    a. Optional: record how the device failed. The Truth Table in *Figure 4* can help pinpoint the failing transistor of the DUT.
5) **Set the switch states to “output” 1, “input” 0. Then change the operation mode back to high-stress mode.**
6) Return the boards into the oven.
7) Repeat data collection steps 1 through 6 for duration of experiment.

Note: the expected time to perform the above test at the calculated temperature should be on the order of 1 week. This has not been yet verified by a control group. A control group should be run first to ensure a proper temperature is being used.

4.3 Analysis of Results

4.3.1 Data Analysis

Perform statistical analysis on the data to obtain MTTF. It has been suggested that a log-normal "bathtub" distribution is used to do this, but details are sparse due to lack of results. Our project was to design the system for gathering data, not to determine the proper method for analysis.

4.3.2 Draw a Conclusion

From the control group's MTTF and the experimental group's or groups', draw a conclusion from the data. If the experimental group(s) exhibited much earlier failures, then this supports the theory of Latent Damage. Otherwise, it would show evidence against the theory.

5. Appendix II: Previous Versions

5.1 Introduction

As should be aforementioned, there was a previous group that worked on this project for our advisor/client. That group of graduate students had produced a set of burn-in boards and a stress board for us to work with. Their work was our starting point for this project.

Unfortunately, the boards were riddled with problems, which was what hindered us from obtaining final results. At this time, we *still* have not been able to fully work out all of the issues.

However, we accomplished many design challenges. The largest being we reverse-engineered all of the previous work and created a new design. To fully understand this process, the next few sections will discuss the previous work that we were provided and how we were able to work through the existing issues.

5.2 Original Burn-in PCB's

5.2.1 Initial Idea

Below you will find a few images of the physical PCBs, along with the documentation we were given for them:

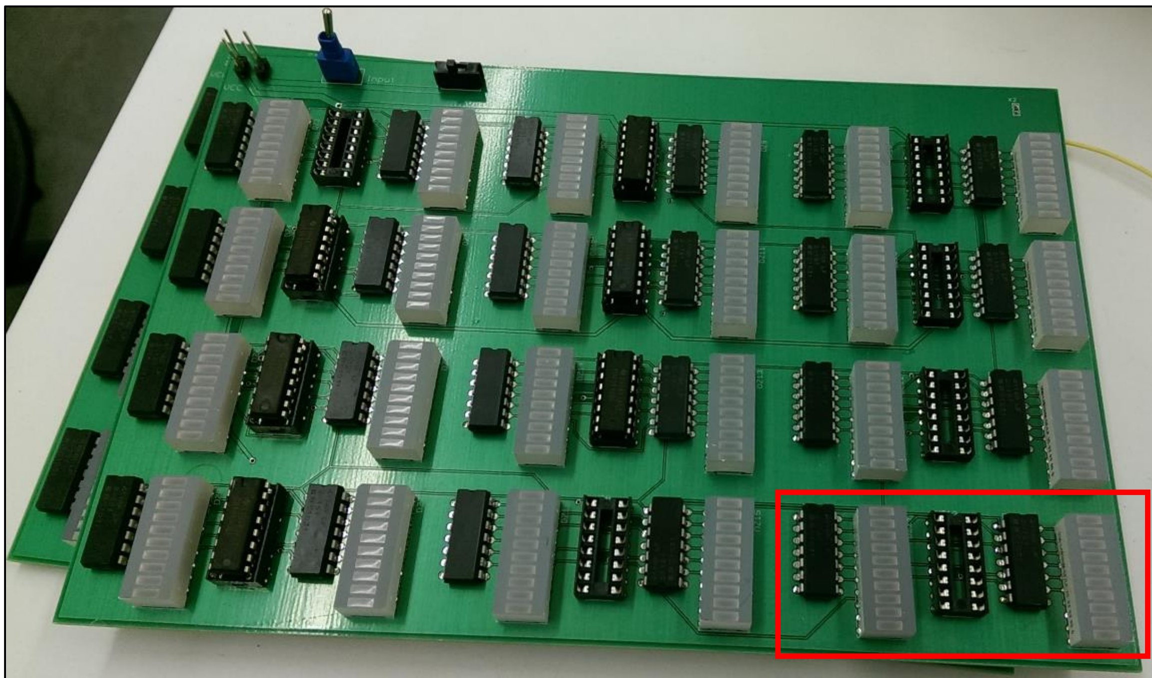


Figure 11: Original Burn-in PCB (populated) with 12 testing clusters

Figure 11 shows a populated version of the burn-in boards. Each testing cluster consists of two diode arrays, 2 resistor networks, and a socket for the DUT. An example of one testing cluster is outlined in red in *Figure 11*. As shown, there are 12 testing clusters per PCB.

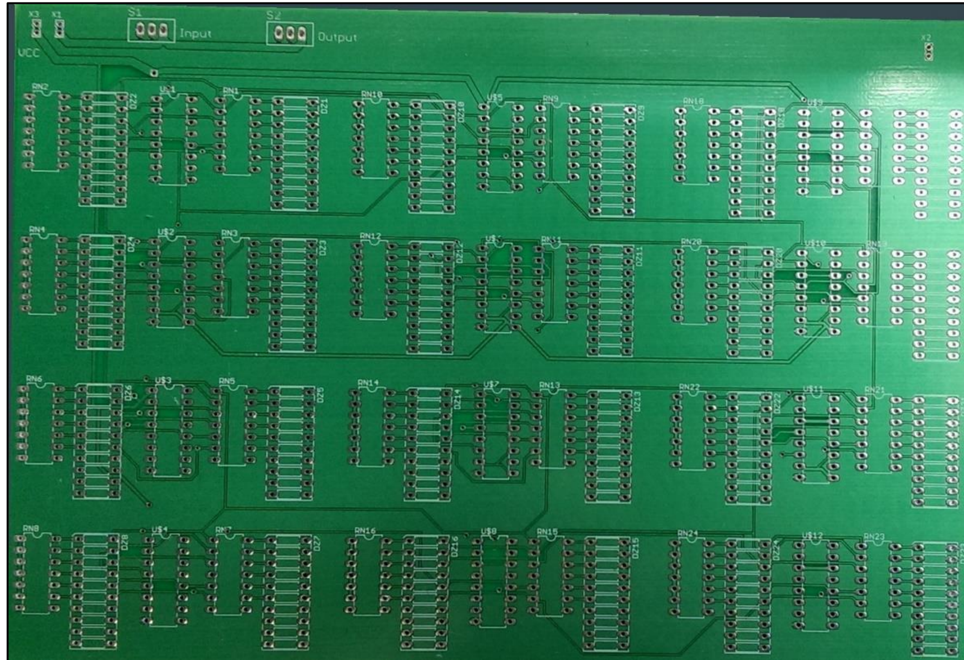


Figure 12: Original Burn-in PCB (not populated with components)

Figure 12 shows an unpopulated version of the original design of the Burn-in PCB. This turned out to be extremely valuable when determining issues with the design.

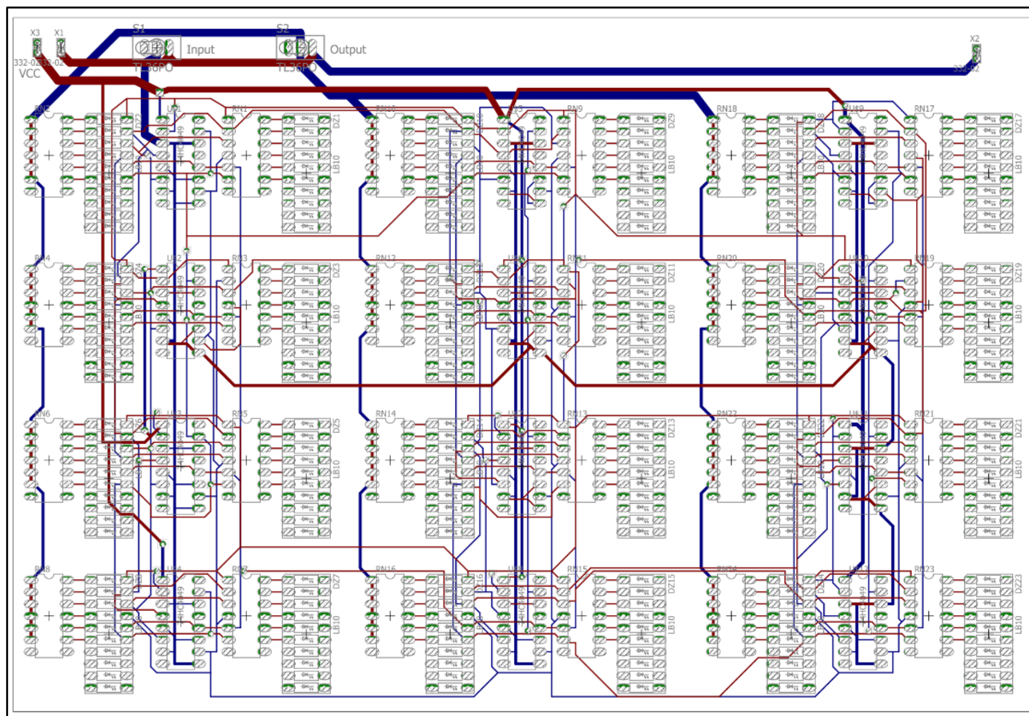


Figure 13: Original Burn-in PCB Layout

Figure 13 is the PCB layout of the original Burn-in PCB design that was given to our team to determine the schematic and operation of the PCB. A hand-drawn circuit schematic would have been easier to work with however, the reverse engineering experience was well worth the troubles.

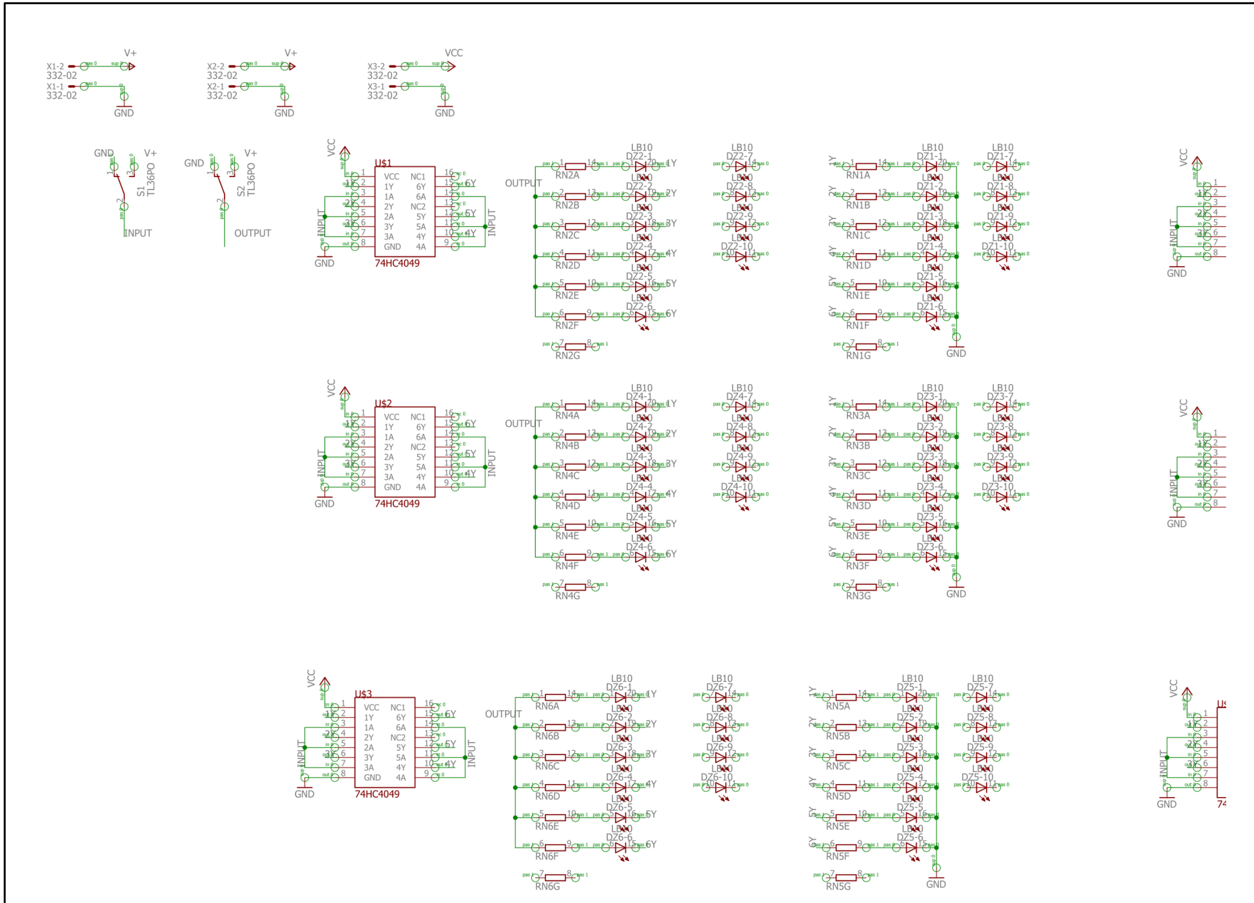


Figure 14: Original Burn-in PCB Netlist Schematic

Figure 14 is part of the original Burn-in PCB netlist schematic that was given to our team. To be honest, this added more confusion to the reverse engineering process because of all the mistakes found in the design.

5.2.2 Identifying Issues

One of the first things we did with the boards was to try to figure out what their functionality was, how they worked and what they did for us. The schematic and the layout shown above were all we had, along with the physical boards.

If you look at the documents themselves, they don't really lend much insight to the operation and purpose of the boards. So instead, we rolled up our sleeves, got out our old lab kits, and sat down with the fully populated board shown above, at a lab bench.

With a little trial and error, we figured out what pins needed to be connected to a voltage supply to power the board, and then we started playing with the switches, "input" and "output."

Right from the start, we immediately were at a total loss for understanding. As we flipped switches, LEDs lit on and off. But the most concerning part of it all… thing is, we had some of the testing clusters with DUTs in them, and others empty. What was scary, was the fact that the completely empty, DUT-less testing clusters were lighting up the exact same way as the rest of the clusters with DUTs.

At this point, with nowhere else to turn to, one of us (guess who?) was given the responsibility of figuring out what was happening. This is where the unpopulated board became insanely useful. The populated boards had the actual parts in it, but the unpopulated one has all of its traces exposed. This allowed me to follow those traces to see how the board was laid out.

The details of the story aren't particularly relevant, but if you're interested, those details are in a sub-section of Appendix III.

What is relevant is the fact that this trace following, knowledge of what components were on the board, and a little bit of knowledge of circuits and digital logic, was what produced the basis of our current design. And it also revealed the two major problems with the original boards.

5.2.3 Troubleshooting Issues

There were two problems that came out of the burn-in board probing. These problems are outlined below in their respective sections:

Problem 1: Bussed Resistor Networks

As it turned out, the resistor networks that were included in the original boards were bussed networks instead of isolated. What this means is illustrated in circuit schematics shown in *Figure 15*:

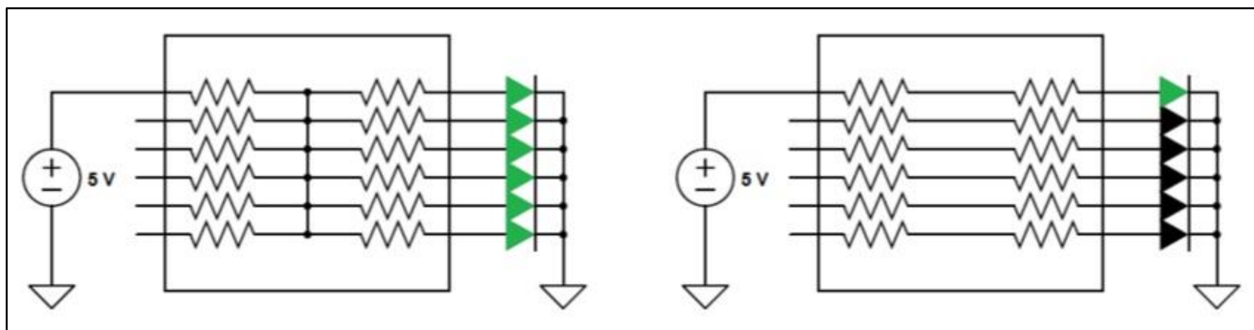


Figure 15: Bussed Resistor Network (left) vs. Isolated Resistor Network (right)

On the left is a *bussed* resistor network (what was in the boards) and on the right is an *isolated* resistor network (what should have been in the boards.) In both cases, one and only pin on the network is fed voltage. In the *bussed* case, this causes the central node in the network to become hot, which in turn will turn on all of the resistor network's diodes, as shown in green above. Conversely, the *isolated* network gives us only one lit diode when the single pin is given voltage.

In terms of our circuit schematic, this means that (for an individual DUT) if at least one inverter gate inside the hex-inverter DUT is working, the testing cluster will show that all six gates are still working properly.

Since our metric for a failed DUT is having *any* of its 6 gates failing a logical test, this means that we can't properly tell when a DUT has failed.

But even worse was the second problem.

Problem 2: Output Connections

This one's a little harder to explain due to the complexity of the problem. This was also the major problem with the board that made me question my sanity while I was probing the board (see Appendix III.)

In order to make this easier, let me just define a few things: On the original boards, we had 12 DUTs, and each DUT has 6 gates. So when I say DUT 1, gate 1 it should be obvious which gate I'm talking about. Thus, we have DUT 1, gates 1, 2, 3, 4, 5, 6, DUT 2, gates 1, 2, and so on and so forth, all the way up to DUT 12, gate 6. So there are a total of 6×12 = 72 individual gates. To make it easier, I'll use the notation D1g1 and so on.

The problem, is that D1g1's output is attached to D2g1's output, D3g1's output, and so on all the way up to D12g1's output. Similarly, D1g2 is connected to all of the second gates for the other 11 DUTs on the board! The pattern is the same for the other four sets of gates on the board too!

This again presents another design-breaking problem with the board's intended functionality. If D1g1 is broken and D2g1 is fine, we can't tell that D1g1 is broken!

The combined effect of both

Obviously, either one of these problems on their own is problematic itself. They prevent us from getting data from failing DUTs' individual gates, which qualify the entire DUT as failed.

But together, if you think about it, what they cause is that the **entire board will behave as if everything is perfectly fine so long as at least one DUT has one working gate**.

If I haven't convinced you yet, then let me make the statement of what that means, in terms of the experiment. With these boards, we would only be able to tell if every single gate on the board was broken. Otherwise, we'd know at least one gate was working. That doesn't help us with what we want to do.

5.2.4 Our Solution

The best way we could think to solve the problem was to simply just make new boards.

We did the math. In order to replace the resistor networks, we would need to de-solder over 3000 pins, get the new arrays, and solder in the same number of pins. And that's just the first problem solved. We'd still need to go in and manually cut a huge number of traces. The complexity of that second problem kept us from figuring out how many traces needed to be cut, but that didn't matter. The pin replacement alone was enough to drive us away, and frankly, if we missed even one incorrect trace on those boards, it'd invalidate our results!

It just wasn't worth the time.

5.3 The Second Burn-in Design

Through troubleshooting and reverse engineering we were able to create a basic schematic of what should have been happening on the burn-in PCBs. While it was only a highly educated guess, the schematic in *Figure 16* should look rather familiar. Given the grievous nature of the flaws, it was only a guess, but the design did what we needed it to do: test the DUT and see which transistor is conducting.

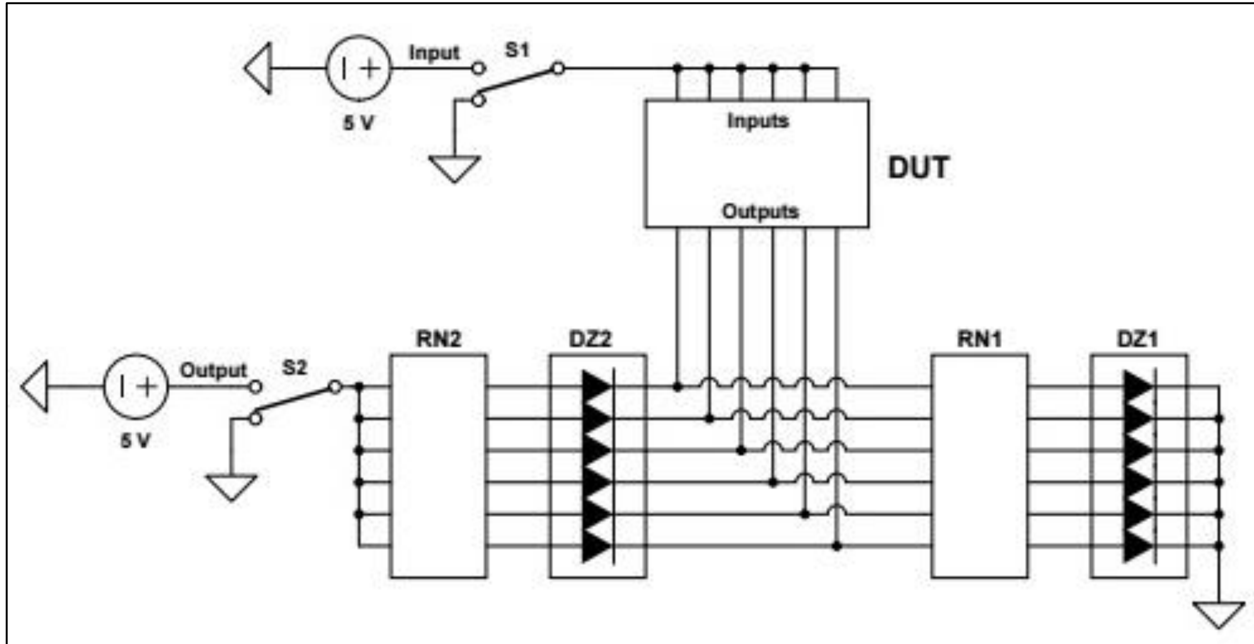


Figure 16: Second Burn-in Design

The schematic included only the DUT and the two sets of networks, set between 0 to 5 volts. The truth table is exactly the same as the one shown in the body of the main report and in Appendix I.

The only thing that this implementation lacks from our current one is the high-stress mode accomplished through the tri-states. We would have stuck with this design, but our client/advisor asked us to add in the high-stress mode of operation. This resulted in our third burn-in PCB design.

6. Appendix III: Other Considerations

6.1 Probing the Original Burn-in PCB

Part of this story was originally in appendix II, but after a while I realized that this would better belong here. In case if you don't want to go back, the start of it has been repeated here. For those who have seen the appendix 2 truncation, the repeated parts are the first 5 paragraphs.

One of the first things we did with the boards was to try to figure out what their functionality was, how they work and what they do for us. The schematic and the layout shown above were all we had, along with the physical boards.

If you look at the documents themselves, they don't really lend much insight to the operation and purpose of the boards. So instead, we rolled up our sleeves, got out our old lab kits, and sat down with the fully populated board shown above, at a lab bench.

With a little trial and error, we figured out what pins needed to be connected to a voltage supply to power the board, and then we started playing with the switches, "input" and "output."

Right from the start, we immediately were at a total loss for understanding. As we flipped switches, LEDs lit on and off. But the most concerning part of it all… thing is, we had some of the testing clusters with DUTs in them, and others empty. What was scary, was the fact that the completely empty, DUT-less testing clusters were lighting up the exact same way as the rest of the clusters with DUTs.

At this point, with nowhere else to turn to, one of us (guess who?) was given the responsibility of figuring out what was happening. This is where the unpopulated board became insanely useful. The populated boards had the actual parts in it, but the unpopulated one has all of its traces exposed. This allowed me to follow those traces to see how the board was laid out.

After a couple of hours of sitting down with the board, I began to get a feel of what was going on. It took a little bit of research to figure out what the other components were. (Research as in, a couple Google searches.) It turned out they were resistor networks, which as explained in the burn-in PCB details section, are only there to protect the diodes from burning out.

As I began following traces for a single testing cluster (which we had identified as a group) I started drawing out a schematic for what was connected to what. This was a bit of a task to do, as the board needed to use both sides to route everything. This was easier by literally taking a resistor out of my lab kit and sticking it through the hole I was following a trace through.

Long story short, I slowly started shelling out a useful circuit schematic on the board. In fact, it was that very same schematic that formed the foundation of our current design, minus the design flaws that broke the functionality of the board.

The huge problem that was in the traces was complicated and convoluted enough that I began to disbelieve my eyes. It took four trace tracings before I came to accept the fact that my eyes weren't lying to me. I then took the unpopulated board to lab to check my sanity. And I was right.

6.2 Advantages & Disadvantages

There are, of course, advantages and disadvantages of our experimental approach.

If latent damage is found, it can be found valid only for COTS devices, specifically bulk-CMOS hex inverters made by Texas Instruments. This can be seen as an advantage or disadvantage.

Another disadvantage of our approach is that we also are not looking at the electrical properties of the devices. While latent damage shouldn't have an impact on the electrical properties of the DUT, there could still be non-latent damage we weren't measuring.

However, this approach does give us advantages. Since we aren't considering the electrical properties of the devices, we can simply run the experiment and by using visual indicators (LEDs) for evaluating the data.

6.3 Budget

While considering all factors, our team managed to stay within the project budget of $1200. *Figure 17* shows the bill of materials which includes all of the purchases from both semesters of work.

Bill of Materials - May1625						Updated: April 12th 2016
Item	**Qty.**	**Reference**	**Cost**	**Part Description**	**Supplier**	**Supplier #**
1	10	Murata Cermic Disc Capacitors 100pf	-	DHRB34A101M2BB	Mouser	81-DHRB34A101M2BB
2	2	1st Design ESD Stress PCB	-	Provided	-	-
3	10	1st Design Burn-in PCB	-	Provided	-	-
4	1	Electric Fencer High-Voltage Source	$53.49	5kV 0.2J	Gallagher	M20
5	1	Digital Electric Fence Tester	$36.37	Tests up to 20kV 3 Digit Read Out	Zareba	DEFT-Z
6	50	10M Resistors (.25W)	$5.74	RNV14FAL10M0	Digi-Key	RNV14FAL10M0CT-ND
7	1	MG Chemicals Insulating Varnish	$9.44	Red GLPT 55ml Bottle	Amazon	4228-55ML
8	10	Texas Instruments Tri-St. Inverters (Hex)	$5.81	CD74HC366E	Digi-Key	296-33070-5-ND
9	20	Vishay Diode 4kV 250mA	$7.08	GI250-4-E3/54	Digi-Key	GI250-4-E3/54GICT-ND
		TOTAL COST	$117.93			

Figure 17: Bill of Materials for Fall 2015 and Spring 2016 Semesters