Hardware Checker for Probe Test Fixtures

ELECOMP Capstone Design Project 2020-2021

Sponsoring Company:

Vicor Corporation
1 Albion Rd
Lincoln, RI 02865
http://www.vicorpower.com

Company Overview:

Vicor Corporation designs, develops, manufactures and markets modular power components and complete power systems based upon a portfolio of patented technologies. Headquartered in Andover, Massachusetts, Vicor sells its products to the power systems market, including enterprise and high-performance computing, industrial equipment and automation, telecommunications and network infrastructure, vehicles and transportation, aerospace and defense.

Technical Directors:

Al Binder
Product Engineering Manager
ABinder@vicr.com

Project Motivation:

We would like to develop custom Test Hardware Checkers to better support internal and off-site testing. The Hardware Checkers would also help to reduce test development time.

The product engineering department has created a new test method for verifying probe card hardware called, “the hardware checker.” The hardware checker verifies all passive or active components on the probe card before it is used to test production material.

The hardware checker uses the test system and test program paired with a **Resistor Ring** (seen in **Figure 1**). The resistor ring provides electrical paths from pins on the Probe test fixture to ground. These electrical paths are comprised of specifically designed resistor values. The resistor ring attaches to the circular grouped pins on the top of the probe card (seen in **Figure 2**).

By terminating the probe card outputs with the resistor ring we can verify the probe card is built correctly.

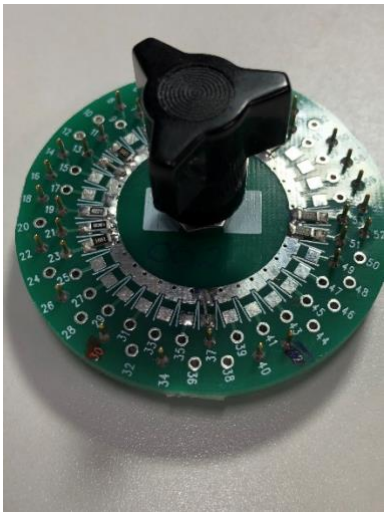


Figure 1

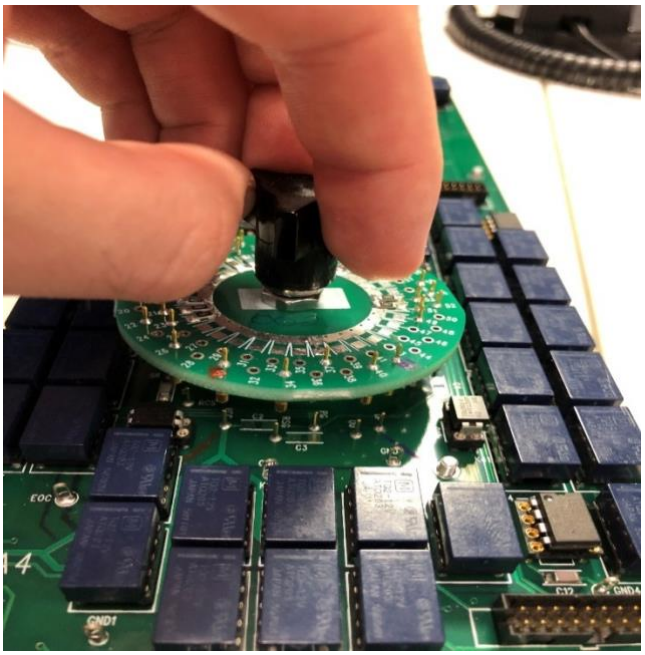


Figure 2

More Motivation:

The hardware checker can immediately tell the test development engineer that either a passive component value is incorrect, a component doesn't have a good connection/isn't soldered, a relay is in backwards or non-functioning, and/or an IC is non-functional. Having a hardware checker available gives us the ability to quickly debug the hardware and find out what is wrong with it. Plus, it can also be used to make sure the test system itself isn't having any major problems. One of the biggest perks of a hardware checker is you are able to test your hardware before starting to test silicon. Utilizing the Hardware Checker confirms that the Test hardware is correct and so all potential discrepancies that come across in the future are related to the silicon or test program instead of the testing hardware.

The hardware checker is also instrumental in contract manufacturer (CM) testing support. It simplifies the debugging process when a CM operator has set up production material and cannot generate a passing device. This saves money on company travel, where normally it would require an engineer to fly out to the CM to solve the issues. It also saves time that would be lost from having to mail hardware that could potentially run into shipping delays or having to go through customs, and money that it would cost. So before they need to mail the test fixture back, they can identify if there truly is something wrong PCB and not just a problem with the tester or silicon.

Anticipated Best Outcome:

The Anticipated Best Outcome is for the Team to create multiple hardware checkers that will be able to rigorously test each of their intended probe cards in order to confirm hardware functionality and completeness.

Project Details:

1st Month:

Students will install Schematic Capture on their laptops.
Teradyne will supply students with offline test licenses.

We will start by reviewing an existing HW Checker.
This example will include the schematics and hardware for a Probe Card and Resistor Ring. In addition, we will review the software needed to run the hardware checker.

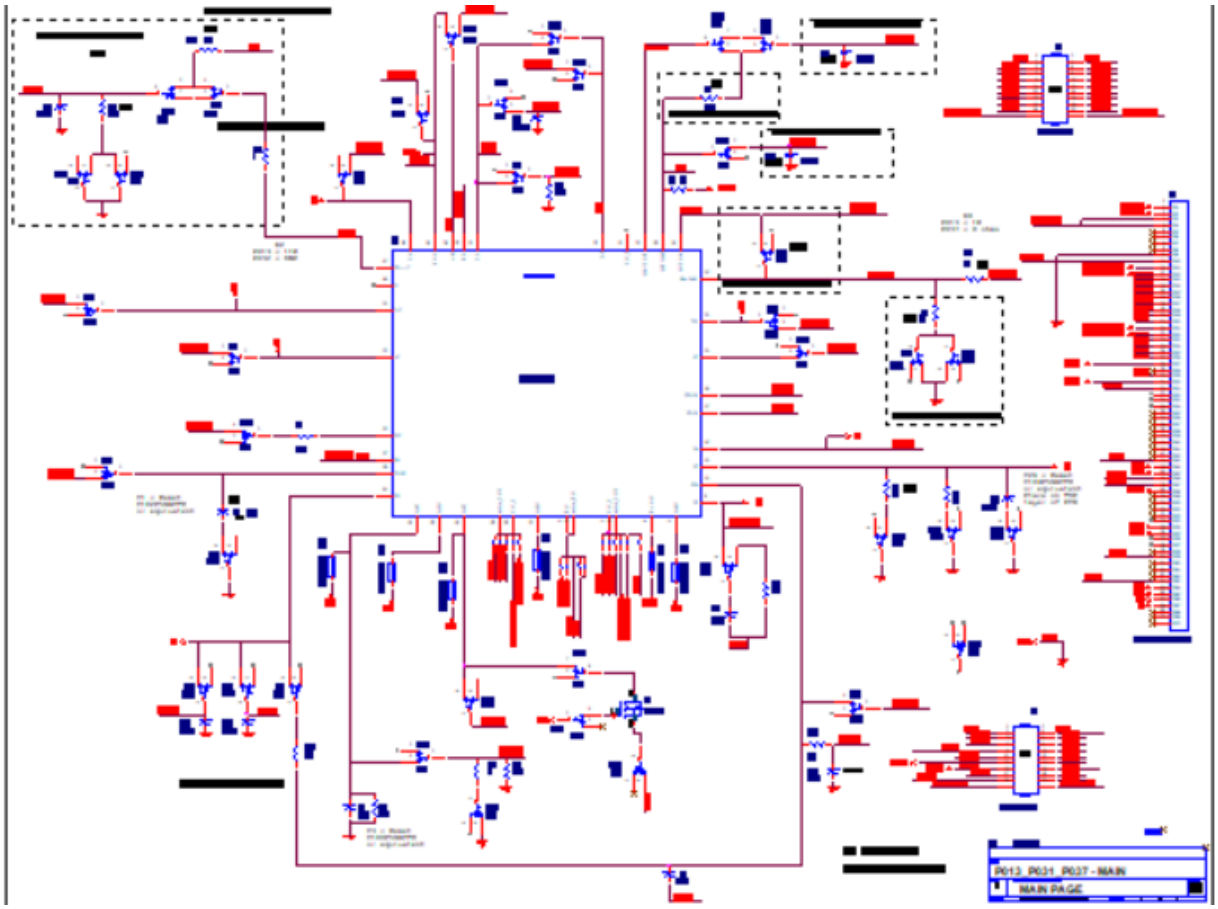
The technical director and support team will assist the students to step through the code and highlight every path that is being checked on the Probe Card.

For any missing coverage we will discuss what test methods should be used.

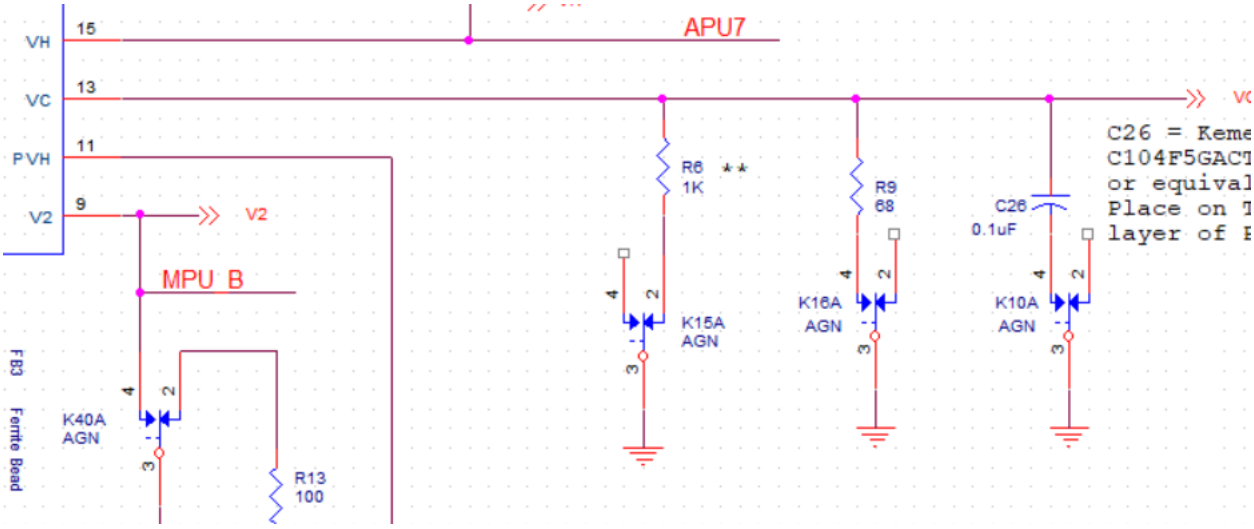
We will also discuss the best datalog naming convention and general strategies for organizing the HW Checker output.

After completing the review we will debug the HW Checker in the test room.

Schematic main page

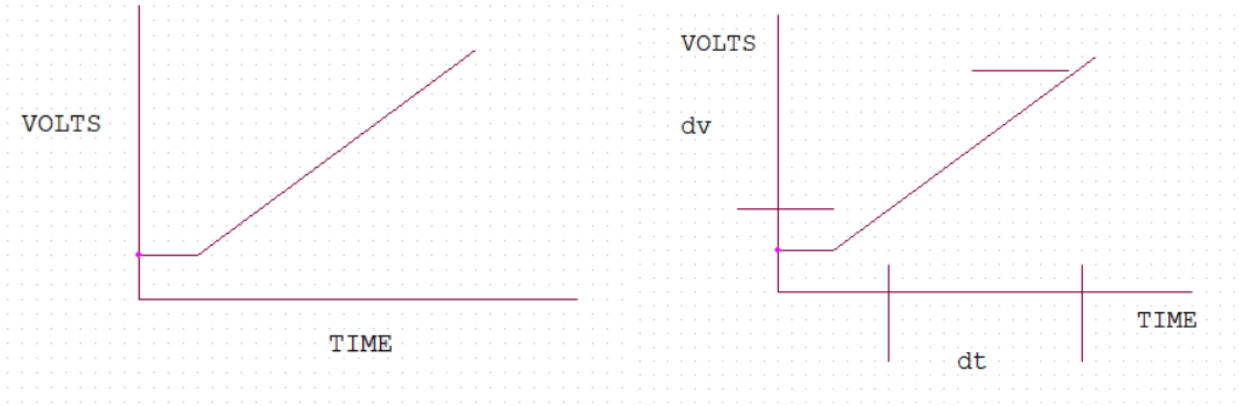


Looking closer on Main Page



If we were to Verify Capacitor C26 we would use the equation:
I = C * dv/dt
I*dt = C*dv
I*dt / dv = C
C = I* dt / dv
C = mA * mS / V = uF

So if we force a constant current at a certain point in time and digitize the voltage result,
It should look like this.



We would write code to get unknowns and solve.

```
sample_1 = whereat( "VC_DIGITIZE", 1, 400, 1 volt);
sample_2 = whereat( "VC_DIGITIZE", 1, 400, 6 volt);

dt= (sample_2 - sample_1)*1e-2 ;

//          mA *   mS   / V = uF

VC_cap = ( 5 *     dt)   /5;
```

Updated code will be tried on test system with corresponding test hardware.

After reviewing an existing HW Checker as a group:

Students will **each** be given another existing HW Checker.
They will look through code and identify what is missing.
Solutions for missing test coverage can be addressed as a team.
The final solutions will be debugged on test system with the corresponding test hardware.

After each student has reviewed their own existing HW Checker:

Students will **each** work on a **NEW** HW Checker.
The eagle test program with header files will be provided.
Students will be responsible to generate the rest of the code.

Students will create a HW Checker Resistor Ring schematic.
Students will populate a blank HW Checker Resistor Ring (PCB will be provided).
Each student will write tests and review schematics for coverage.
Solutions for missing test coverage can be addressed as a team.
The final solutions will be debugged on test system with the corresponding test hardware.

Students will be responsible to generate their own schedules based on their own anticipated progress. More emphasis should be placed on quality of work rather than quantity.

Students will be required to:

- Create a Resistor Ring schematic for each specific hardware checker.
- Utilize offline simulated tester to modify existing Teradyne ETS 300 test program in C++ to write and compile each test
- Copy existing Teradyne ETS 300 test program in C++ and modify it using a simulated tester to write and compile each test. ADVANCED
- Populate hardware checker PCB with appropriate passive components.
- Verify schematic and all active and passive components.
- Run hardware checker solution on existing production PCB fixtures.
- Evaluate results and improve on result descriptions to help debug process.

Students will learn:

- Test techniques for testing capacitors, resistors, low input offset voltage buffers, video-buffer, op-amps, gain amplifiers.
- Test techniques for testing ECL Ultrafast Comparators, parallel resistors, parallel capacitors.
- Analyze datasheet of active components to figure test requirements.
- How to handle separate force and sense lines.
- How to add tests and compile using the Teradyne ETS 300 test system.
- How to utilize the Teradyne ETS 300 test system debug tools.

Composition of Team:

The composition of the team will include:

- Two (2) Electrical Engineers (ELE)

Hardware Required:

- Students need their own laptops.

Skills Required:

Engineering Skills Required:

- Basic electrical engineering analysis skills (KVL, KCL, Ohm's Law).
- Basic debug skills.
- Basic ability to solder
- Ability to quickly learn and work with new software

Anticipated Best Outcome's Economic Impact on Company's Business:

We anticipate that each hardware check will save 10% of each development time. In addition, we anticipate that each hardware check will reduce in-house and off-site debug time by 50%.

Broader Implications of the Best Outcome on the Company's Industry:

The Hardware checker method can be used to better facilitate low cost off-site testing.