A Typical Test Designer Scenario

Test Designer is a powerful new program that defines an entirely new type of software; Analog and Mixed-Signal Design and Test Automation and (AMDTA). In this application note we will explore an example of how Test Designer is run.

The initial step in using Test Designer is to enter the schematic drawing for the circuit you want to test. Test Designer includes a special schematic entry tool specially designed for analog and mixed signal designs. The design(s) are simulated using the IsSpice4 simulator which is integrated with the product.

If you are using Test Designer in conjunction with an initial product design, then you will probably make the first pass through Test Designer without real hardware comparisons. If the product already exists, then you can verify the simulation results by comparing them with the actual hardware. The following steps outline the typical test set design flow using Test Designer for a laser amplifier (figure 1).

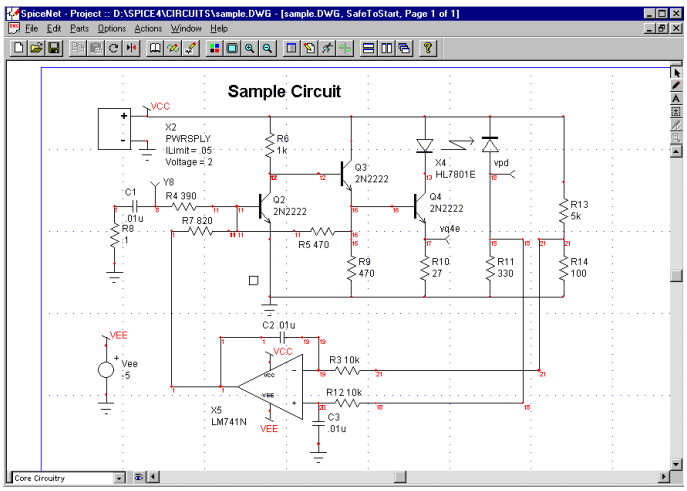

Figure 1, a laser diode and amplifier in the Safe-To-Turn-On configuration.

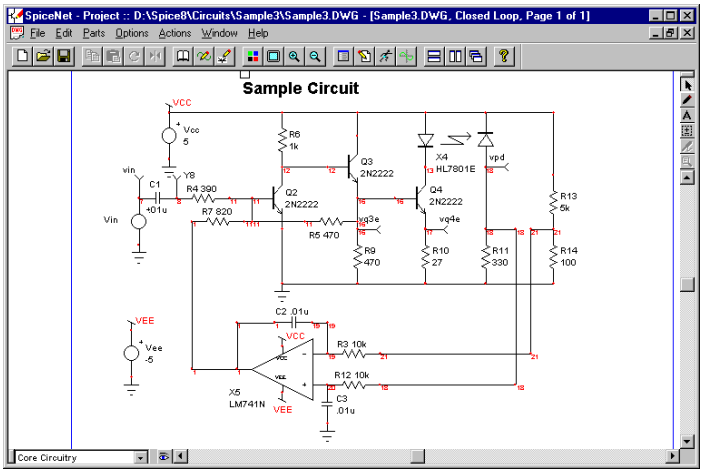

Figure 2, the closed loop configuration of the amplifier circuit.

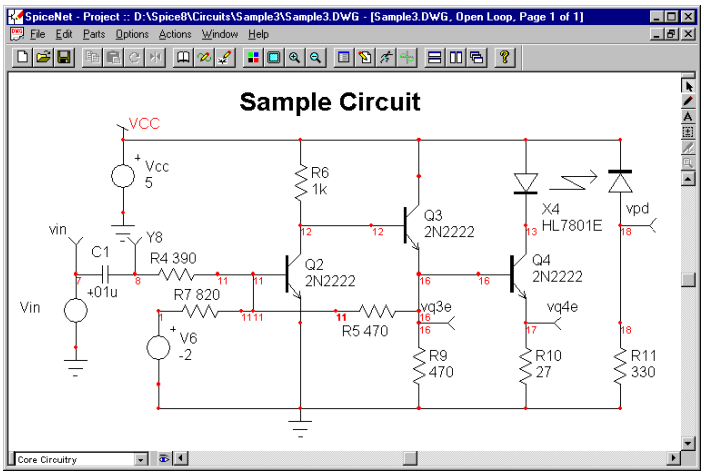

Figure 3, the open loop configuration of the amplifier circuit.

The schematic provides some unique features including the ability to combine multiple layers into various schematic configurations. Each layer can have any type or amount of circuitry. In this case, there is a closed loop configuration, an open loop configuration as shown in figures 2 and 3, a second open loop configuration where only the circuit stimulus has changed, and a special “safe to start” configuration (figure 1). There are also production and component test configurations, but these will not be utilized to design any tests. They are used for PCB layout and component behavior verification, respectively. These configurations will be paired with different IsSpice4 analyses (AC, DC, transient, etc.) as shown in figures 4 and 5.

Figure 4, the Simulation Control dialog and the pairings of circuit configuration and analyses that will be simulated. The Simulation Control dialog drives the test design process.

It is important to note that multiple independent schematics are NOT utilized. All of the design information is kept in <u>ONE</u> schematic database.

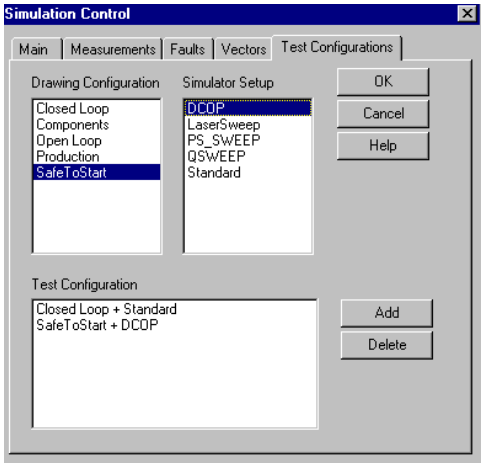

Figure 5, the Test Configuration dialog is used to combine different circuit configurations and simulation setups.

The Standard analysis combines operating point and transient analyses. Next, measurements will be assigned to each circuit configuration-simulation setup pairing to form our tests.

Test & Measurement Setup

Two test groupings are setup. The simplest test groups are handled first (for example, the DC operating point).

1.1) The first group (Operating Point) measures ALL of the IsSpice4 vectors (node voltages, device currents, etc.) that you may want to use to characterize your circuit's performance. Test points can be disabled for future calculations, but you will have to repeat your simulations if you add new vectors later. The measurements are assigned with a Wizard-like approach whereby a series of simple dialogs are used to choose the measurements and the vectors that will be measured (figures 6 and 7).

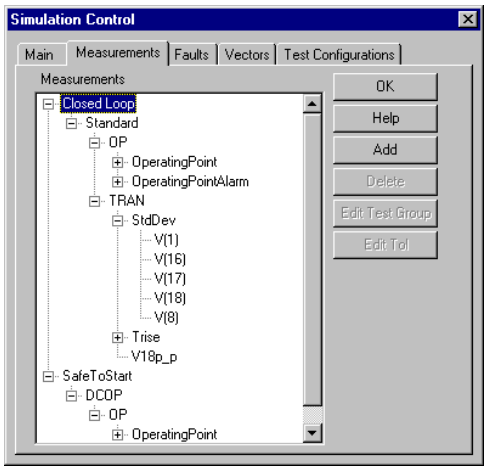

Figure 6, the Measurement dialog shows the various measurements (operating point value, RMS, pk-pk, rise time, etc.) assigned to each circuit/simulation combination in a treelike fashion.

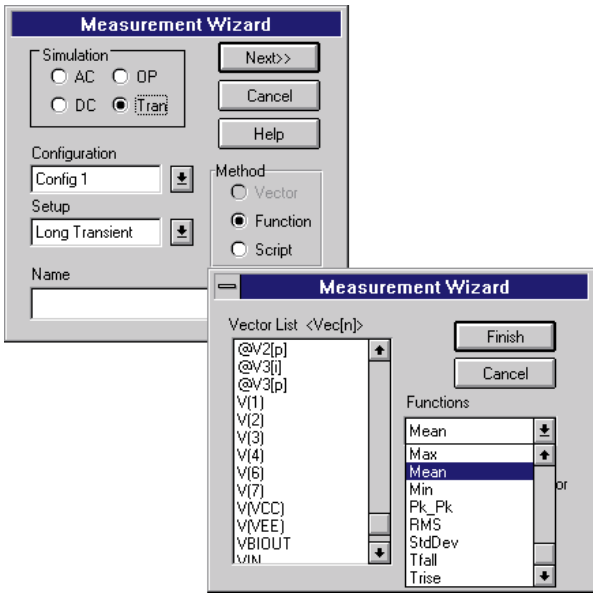

Figure 7, the Measurement Wizard dialogs make measurement selection easy. A powerful cursor scripting language is also available for you to use to make advanced measurements.

1.2) The second group (Operating PointAlarm) checks for component stress. Usually this means checking device power dissipation, but it may also include maximum voltage and current, if needed. This will be called the "Alarm" Group.

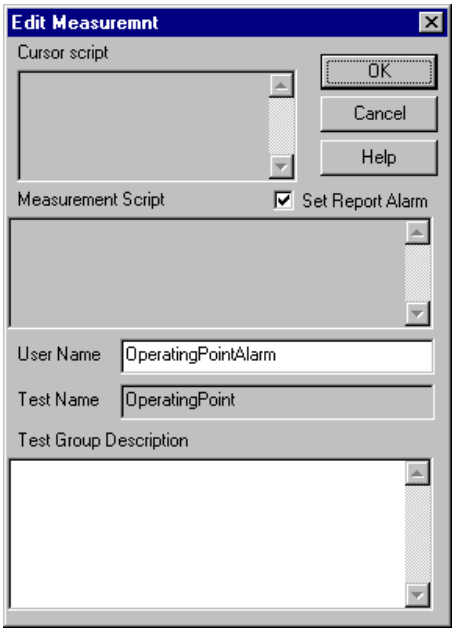

Figure 8, the Edit Measurement dialog allows you to edit your measurement and cursor script. You may also set an alarm to trigger if this measurement's pass/fail bands are exceeded.

When the Set Report Alarm flag is active (figure 8), subsequent fault testing will trigger the alarm if a particular failure mode produces an overstress condition. It is customary to set the stress limits to 50mw for low power parts, and 10 or so percent above normal operation for higher powered parts.

1.3) We may also select the data reduction options before running any simulations. The Script button (checked in figure 4), means the measurements we just set up will be recorded automatically when a simulation is run. If you want to see the simulator run, select the interactive button; if you are running a lot of simulations, you probably want to run IsSpice4 without seeing the Real-Time display, so we will use the Batch radio button.

Tolerance Setup

2) Next, we must set the measurement tolerances for the functional test groups. A number of methods are available for setting tolerances.

Key Test Designer Feature

Nominal and tolerance data can be extracted from the simulation data, making it a snap to set tolerances.

For this example, we will run a Monte Carlo simulation - somewhere around 30 cases gives a fairly good estimate of the standard deviation.

2.1) When the Monte Carlo simulation finishes, a nominal reference simulation is run. The results, shown in figure 9, are used to set the nominal measurement values. You may check the default tolerances and make any additions or changes before continuing or we can use the "Set Limits" dialog (figure 10) to setup the measurement tolerances. Tolerances shown in the report dialogs use 3 sigma values.

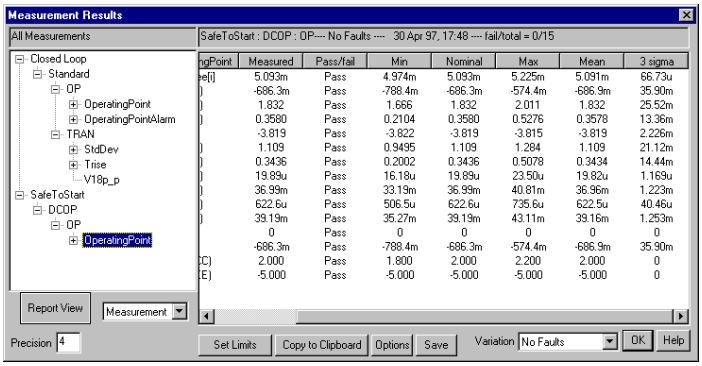

Figure 9, the results of the Monte Carlo and nominal simulation runs.

Key Test Designer Features

Reasonable defaults are setup for all failure modes and measurement tolerances, so the designer can be productive immediately.

At this point, we set the tolerances to that of the nominal reference run by selecting the Last Measurement option. Then we will expand the tolerances based on the Monte Carlo results (mean/3 sigma in figure 9) using the Sigma limit function.

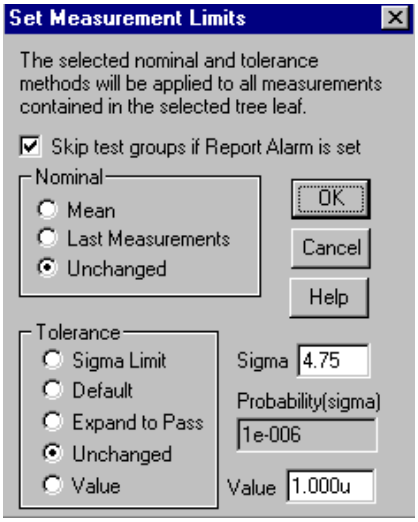

Figure 10, the Set Measurements dialog allows you to set tolerances for a group of measurements using several powerful methods.

The probability calculator tells you the odds that a measurement may fail. While individual part tolerances of 3 sigma make sense, a higher sigma limit for a measurement tolerance is justified because we would otherwise reject an unacceptably high number of units. It's up to you, however, to make sure that performance limits are well outside these Monte Carlo limits. You should also expand any failed alarm settings to pass.

You can also make additional simulations after changing temperature, input power supply levels or other parameters. After each simulation, you can expand the tolerances to pass these results by selecting "Expand to Pass".

2.2) The measurement limits should all be less than the product performance specification. Any that are larger must be resolved through design changes, specification revision or simulation model changes.

Run The Simulations

3) Next, we automatically simulate all of the failures in the test grouping by selecting the appropriate Test Configuration in the Simulation Control dialog and clicking the Test Designer radio button (figure 4). This operation simulates all assigned part failures, one after the other, without any user intervention required. To avoid long simulations, Test Designer makes available a time limit setting that will terminate any simulation taking longer than a specified time.

Key Test Designer Feature

For each test, you may enable or disable each part's fault mode so that simulations don't have to be repeated. This allows removal of fault modes that the designer doesn't want to consider for a test, and addition of special tests under the fault mode umbrella (e.g., power supply variations).

After the failure analysis runs are completed, we can review the results to make sure that all of the simulations ran. You can highlight the "NoFault" item in the Variation box at the bottom of the report window, then use the keyboard arrows to scroll through the results for each fault.

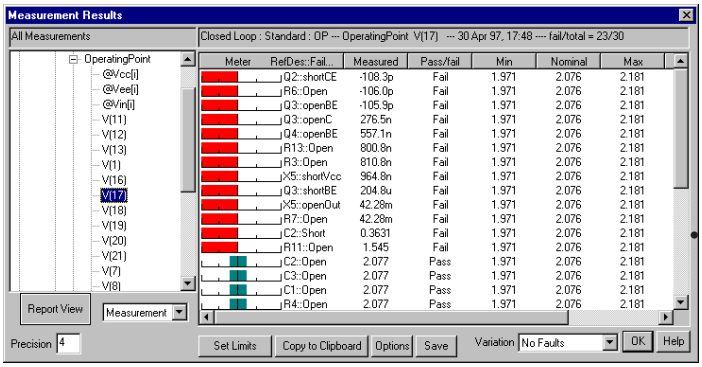

Figure 11a, the Test Designer Report showing all the fault mode values for node V(17).

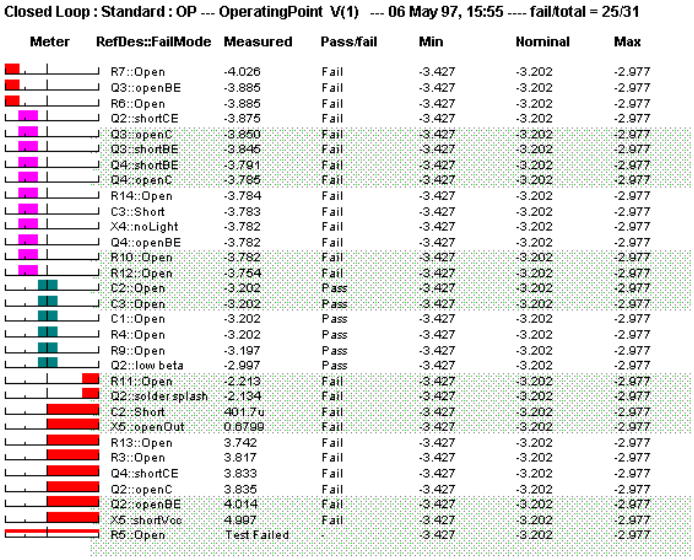

Figure 11b, a second view of the Test Designer Report showing a sorted list of the fault mode values.

A flat red line accompanied by "Test Failed" in the Measured column indicates a simulation error. If you performed several analyses, like OP and Tran during the same simulation, a failure in one and not the other will invalidate all measurements for the simulation. You may want to simulate these faults individually with different simulator options. The Convergence Wizard available in Test Designer can help you adjust IsSpice4's options.

Note: Faults may also be simulated one at a time.

Key Test Designer Feature

The Test Designer Report can show all the measurements (one fault at a time), or it can show all the faults in a list or in a histogram fashion.

Test results are reported for groups of tests. An easy-to-read graphical histogram meter shows the state of each resultant measurement in the group, along with numerical results and test limits. Each measurement can be folded out to show the results for all faults. When ordered by group, each measurement is decomposed into a histogram that corresponds to the graphical meter. Each failure mode is then displayed in the histogram bins, along with a count of the number of failures in the bin. When displayed by measurement, the failure modes are sorted by the relative value shown in the histogram meter.

At this point we must eliminate the failures that couldn't be detected by the tests, that is, the ones that didn't converge, and failures that overstress the circuit. The over-stressed failures will be reported in the Alarm Summary dialog.

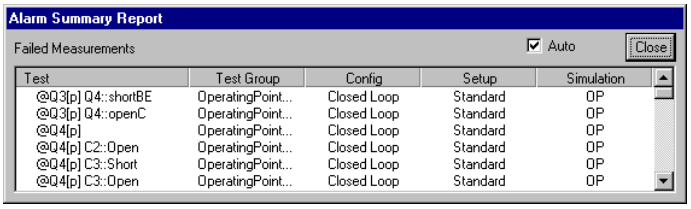

Figure 12, the alarm dialog automatically shows the measurements that did not pass.

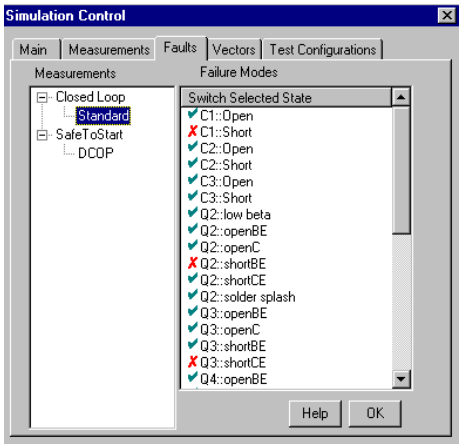

Figure 13, the faults tab can be used to turn selected faults on and off for each test.

The Faults tab in the Simulation Control dialog can be used to remove the unsimulatable faults from the existing tests (figure 13).

Now comes a more difficult problem; we must make a new test that will detect all of the faults we've just removed, without causing any overstress or convergence failures.

These tests can take many forms. Operating the circuit with reduced power supply voltages, using current limited power supplies, performing a ramped power transient or doing dead circuit multimeter tests are among your options. For our sample, we've used a reduced power supply in combination with a current limit feature. This assures us of not having enough power to damage anything, and solves our convergence problem, which was really an overstress problem in disguise.

Test Sequencing and Synthesis

4) Now we get to design our test

4.1) The actual test synthesis and sequencing is accomplished in the Test Designer Fault Tree dialog (figure 14).

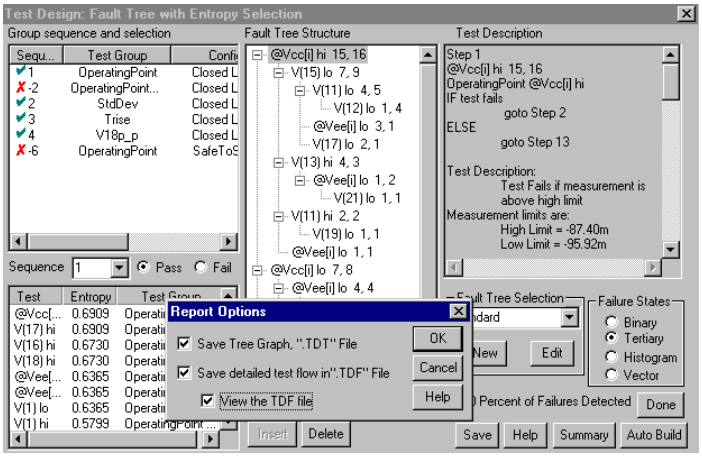

Figure 14, the Fault Tree definition dialog is the heart of the Test Designer system.

First, we enter a test name and a description, if desired.

4.2) Next, we set the test group to use for this test (Safe to start, in this case) and assign it a positive sequence number, 1. Other test groups may also be used in this fault tree.

4.3) Then we select the sequence number, using the Sequence drop down list.

4.4) Now we select the Failure States that are used for the tests. Usually, this will be binary or tertiary. Binary treats each test as having a pass/fail outcome. Tertiary divides a measurement into 2 tests; first, Fail hi or Pass and second, Fail low or Pass. Histogram and vector are experimental at this time. Histogram divides a test in the same bins as the Results dialog histogram meter. Vector assigns a measurement region about the observed state for each failure mode. Both approaches require greater simulation accuracy for failed outcomes, an assumption that requires very accurate models for out-of-tolerance operation. You can use these to see if something better is possible; the additional tests will not typically provide improvements in fault isolation.

4.5) Next, we let Test Designer automatically build the fault tree. The easiest way to do this is to press the Auto Build button. It will complete the tree by selecting the first test it finds in the Test entropy list. The list is sorted from highest entropy to lowest entropy.

Entropy is calculated by summing (-p*log(p) - f*log(f)) for each fault mode being tested, where p and f are the pass and fail outcomes. Probabilities are assigned using the failure weights in the Part Properties dialog (Failure Modes tab, Failure Mode Definitions subdialog), except for the no fault probability that you may assign. Test Designer uses failure weights where a weight of 1.0 is the default for each failure mode. You can change these to actual failure rates if desired. The no fault weight depends on the test. If you are performing a production acceptance test, you would expect most units to pass because of your zero defect policy, so the no fault weight would be set very high. A high weight would force the test sequence to be most efficient in detecting the no fault case. On the other hand, a unit that is returned from the field is likely to be bad, so we would assign a low weight. This causes the higher failure weighted items to come to a conclusion with fewer tests. The failure probabilities are normalized for each test since the probability of exiting each test must be 1.0.

Alternatively, you can use the insert button to insert tests one at a time in a manual fashion. The pass/fail radio buttons show whether the test connects to the selected test's pass or fail output. The test you pick from the entropy list, by clicking on the desired test, is used.

You can delete any tree item, its siblings and children by selecting the item and pressing the delete button.

Key Test Designer Feature

Tests are designed using a Fault Tree approach. Each tree node has 1 input and 2 outputs, pass or fail. The input contains a list of the undetected faults. The pass output contains a list of faults that passed the test, and the failed output contains the remainder of the input fault list. The fault lists are referred to as ambiguity groups. The objective of the test design is to get to the end leaf(s) with the least amount of work. For fault isolation, this means reducing the ambiguity group size to a reasonable number of parts that need to be replaced. For product acceptance tests, it's to get to the tree roots "pass" output. The root test input includes the no fault test and its probability weight. Each fault is also weighted. The best results occur for the higher weight; that is, Test Designer selects tests whose outcome has the highest probability (highest entropy). For fault isolation, no fault carries a small weight and for product acceptance test it has a high weight, causing the tree forming logic to produce the best test sequence.

Test sequencing is accomplished by changing the Failure State mode or the Sequence number. Either action will build a new entropy list and you can repeat the procedure in item 4.5. The percentage of failure modes detected is shown next to the Done button, as you go. If it gets to 100%, you will not be able to add further tests. For example, if you prefer a binary test, select Binary and then press Auto Build, then select Tertiary, then press Auto Build. If you want to consider Operating point and transient tests together, assign them the same sequence number.

You can get a summary report by pressing the Summary button. It will show how the failure modes are broken down by group size. The details for each test are viewed in the Test Description test field. The goal is to have only one result in this group (that is, no fault). When that is true, you have 100% testability of the failure modes that were input to this test.

Key Test Designer Feature

Test design is accomplished graphically, by either automatically or manually selecting fault tree nodes from an ordered list. The list is sorted by test entropy. The results for each fault tree design are saved with user names and descriptions. A fault tree can be designed all at once, or in steps that gradually add more complex tests. The user can look at the results for each fault tree node, examine the underlying tests and modify the design strategy.

Test Designer Reports

Test Designer generates several types of reports.

sample.TDT

```
File D:\SPICE4\CIRCUITS\sample.TDT
Fault Tree for: safe to start
30 Apr 97, 17:50
if( OperatingPoint V(11) hi == FAIL)
{
    if( OperatingPoint V(13) hi == FAIL)
    {
        Q3::shortCE
    }
    else
    if( OperatingPoint V(7) lo == FAIL)
    {
        C1::Short
    }
    else
    {
        Q2::shortBE

    }
}
else
if( OperatingPoint @Vee[i] lo == FAIL)
{
    R5::Open
}
else
if( OperatingPoint @Vee[i] hi == FAIL)
{
    X5::shortVee
}
else
{
    NoFault

}
```

Figure 15, one of the several report formats put out by Test Designer. Here a pseudo-code output of the fault tree is shown.

Key Test Designer Feature

Test results are output to a file in several formats that can be imported into the designer's test programming language. These files contain the logical structure of the test. The designer can then add coding for specific test equipment. The data format is suitable for reports and user documentation.

You can refine your tests by eliminating test vectors. This allows you may to trade off test complexity for test point availability. With a little luck, you will have the same testability and slightly less efficient fault isolation. The best result is subject to the electro-political constraints placed on your test design by the outside world.