APPLICATION NOTE AN-129

Data Driven Testing

I) Introduction

Most Reedholm customers have been able to eliminate most, if not all of the programming, required by other brands of parametric testers. As others have defined, Data-driven testing (DDT), also known as table-driven testing or parameterized testing, is a software testing methodology that is used in the testing of computer software to describe testing done using a table of conditions directly as test inputs and verifiable outputs as well as the process where test environment settings and control are not hard-coded.

Whether one is using the original EMPAC program (Electrical Measurements for Process Analysis and Control) or the BUILD version in the Intranet/Windows 2.0 release, an engineer can quickly build a test plan for a process/product, including test optimization, ensuring that the tests are executed as rapidly as possible without compromising results.

II) Benefits of Data Driven Testing

With data-driven software, Reedholm systems have been placed into production within one week of delivery. Iit takes about a day to create a complete 50 parameter test plan. By design, data-driven test engines are flexible, and durable. A data-driven engine performs tests based upon passed-in parameters, not on hard-coded parameters. Every pin number, voltage level, current limit, and connection scheme used per test is stored in a database file, and not in any source code. This allows the customer to upgrade the software and computer platforms without having to port test code. A data-driven test engine also allows users to:

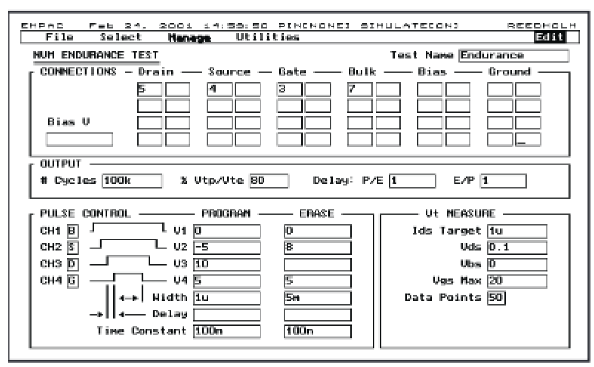
- Create huge test plans
- Obtain hard-copy documentation of the test
- Create and maintain test plans easily

Because such test plans are data records linked in memory, the number of tests in a single plan is nearly unlimited. It is possible to have test plans containing more than 100,000 tests.

III) Editing Test Plans

Test plans are edited either by filling cells (most common) or during a question-answer session:

WHAT IS THE TEST NAME? VT0@PMS Vd=0.1
Is this a (P) or (N) channel device? N
Will current be measured in (D)rain or (S)ource leg? S
Will transconductance(G) or threshold(V) be measured? V
DEVICE PIN ASSIGNMENT
Which is the drain pin (Hipin)? 1
Which is the source pin (Lopin)? 4
Which is the gate pin (Ctrlpin)? 2
Which is the substrate pin (Enter "0" if not used) (Biaspin)? 0
INSTRUMENTATION SETUP
What drain-source(Vds) voltage will be used? 0.1
What should the drain current limit be? 1m
What is the upper bound of the gate voltage search(PS2)? 6
What is the lower bound of the search(Calc1)? 0.5
What should the gate current limit be (PS2)? 10u



Compliance with system instrumentation is checked as each test is edited. Attempts to set impossible hook-ups, e.g., more pins than are in the matrix, voltages or currents greater than those available from the installed instrumentation, are detected and prevented. When editing off-line, the instrumentation is simulated, enforcing test plan compliance and helping avoid non-working tests.

IV) Comprehensive Coverage

Almost any dc test can be performed on a Reedholm test system and most are standard library choices. Since many test types named are generic and can be used for other “unnamed” test types, there is no need to individually list breakdown-drain-source, gate-substrate, collector-emitter, etc. Any test that forces current while measuring voltage is appropriate for many types of breakdown measurements.

Conversely, the generic test which forces voltage while measuring current can be used for many types of leakage current measurements. The ability to connect extra pins provides even more flexibility—pins can be grounded, have voltage or current bias applied, or be tied to the high or low terminal, as dictated by the test structure.

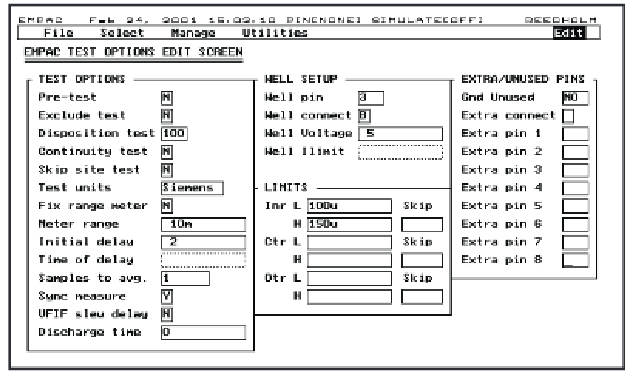
The following is a list of the test types supported. Tests for WLR are described in a stand-alone data sheet.

Resistance 2 & 4 Terminal	General Purpose Tests	Field Effect Transistor
Force V, Measure I, Calculate R/unit	Force I, Measure V (±100V)	I_{DS} @ V_{GS}
Force I, Measure V, Calculate R/unit	Force I, Measure V (+200V)	V_{GS} @ I_{DS}
van der Pauw	Force V, Measure I (±100V)	V_{GS} @ % I_{DS}
Bipolar	Force V, Measure I (+250V)	G_M @ I_{DS}
Beta @ I_C	Force V, Measure I (+1500V)	G_M @ % I_{DS}
Beta @ I_B	Ramp V Until I Test, Return V	G_M @ 2 V_{GS}
Beta @ I_E	Force V, Measure Picoamps	G_M @ 2 I_{DS}
Peak Beta	Snapback BV (+1500V)	G_M @ PMS
Small Signal Beta	Continuous BV (+1500V)	Saturated V_T
Early Effect	Force V_{DD}, Measure Frequency	Stress @ V_{GS}
NVM	Stress @ I_S, Measure V	V_{TO} @ I_{DS}
Endurance	Capacitance	V_{TO} @ % I_{DS}
V_T Measurement	User Written Tests	V_{TO} @ 2 V_{GS}
Program Pulse	Delta Length	V_{TO} @ 2 I_{DS}

V) Test Options

Each test has a group of options available that can affect the test setup and action taken after the test has finished. One of the most-used options is the ability to execute an equation after the test has finished. Equations can extract prior test results and/or test values. Such equations can be used to reverse the polarity of a test result, change the result units of a test, calculate the series resistance of the last five tests, etc. Other test options include the:

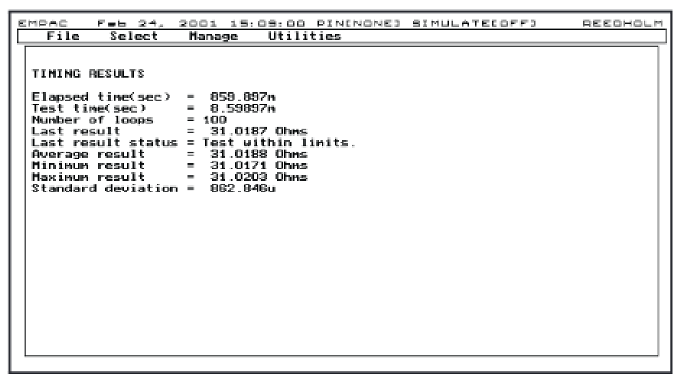
- Availability of three sets of result limits for binning
- Skipping of tests, sites, or wafers upon test failure
- Exclusion of test results from lot reports
- Making any test a continuity test
- Selection of delays, averaging, meter range, etc.
- Connection of extra pins and voltage supply
- Discharge of cable/DUT at end of test



VI) Test Optimization

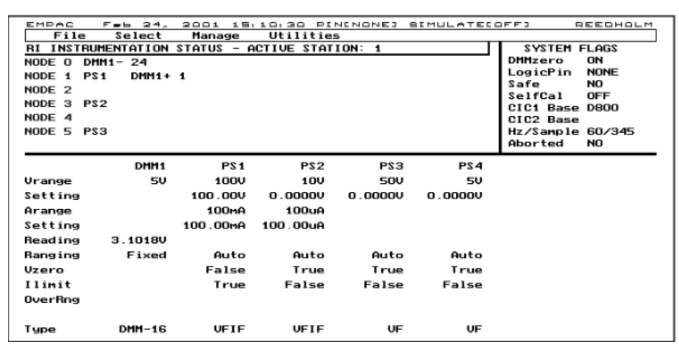
Accuracy and repeatability are improved upon by increasing the delay between force and measure and by increasing the number of readings and to automatically set the sample rate to average out 50Hz or 60Hz noise.

Of course, some of these options can result in a decrease in tester throughput. To handle the speed versus repeatability dilemma, the execution time for each test can be measured. By experimenting with delay times and averaging and using the built in timing option, one can optimize a test both for speed and repeatability. Again, because the test engine is data driven, optimization requires no compiling and takes only minutes.



In addition, the test engine has built-in error checking that reports error conditions, including if a::

- Meter is over ranged
- Power supply is in current limit
- Search target is out of bounds



VII) Extending the Test Library

The majority of Reedholm customers have used data driven software for decades. Out of hundreds of installed systems, 95% of the plans consist of standard tests - meaning no programming is required.

The typical customer has only one or two unique tests that do not exist in the library. These unique tests are also usually proprietary. When this occurs, the library is extended with user-written code called User Functions. This custom test code can be written to conform with the

Standard tests so that all the parameters are passed in, including pin numbers, voltages, delays, and sampling options. By conforming to the standard format, this new test can be accessed like all the other test types andre-used in dozens of plans without ever having to re-visit the source code. Complete user functions include their own investigation code, which is enabled when the built in error checking debug flag is set.

Reedholm provides several user examples, as well as a manual. Reedholm can also be contracted to write custom functions and equations.