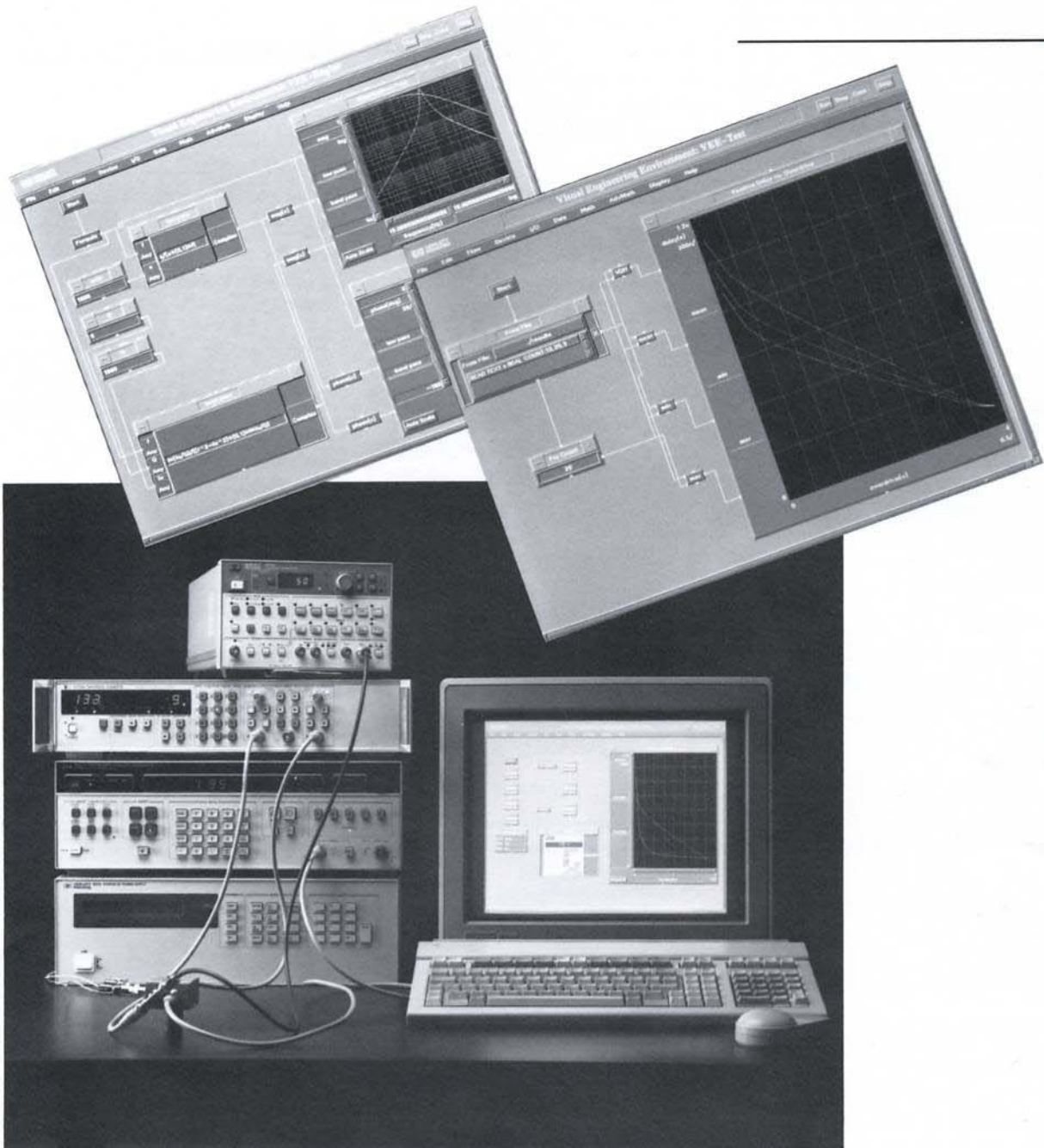

Design Characterization Using HP VEE-Test

Application Note 1206-02



Introduction

Engineers often need to characterize a design. After a working model is built, various characteristics of the prototype must be measured to verify that it meets the design goal. The designer understands the characteristics to be measured and the instruments used to make the measurements.

Computers can be valuable tools in performing design characterization. Instruments are available which can be directly controlled by the computer, eliminating the tedium of setting up instruments for large numbers of data points. The computer's computational power makes analysis of the collected data easy. Unfortunately, many programming environments demand more attention to the mechanics of writing a program than to the measurements and analysis themselves. A good environment allows the designer to focus on the problem, not the program.

HP's visual engineering environment for test (HP VEE-Test) is a software tool which allows you to create complete solutions by linking visual objects or icons, rather than by using traditional programming languages. Since the process of creating solutions is similar to creating a block diagram, HP VEE-Test offers users an intuitive approach to using a computer, thus saving considerable time and effort.

HP VEE-Test provides a wealth of objects which aid in the collection, analysis, and presentation of data, in addition to objects and features for data

storage, flow, modularity, debugging, and documenting. This power, combined with its ease-of-use, makes HP VEE-Test an effective tool for solving design characterization problems.

This application note outlines a design characterization problem and demonstrates how it can be solved using HP VEE-Test. It covers instrument control, analysis, and displaying and storing results.

The Problem

Consider the system shown in **Figure 1**, in which digital signals are applied to a device and the response is detected. The characteristics of all components in the system contribute to the overall time which is required between the application of new signals. The remainder of this application note focuses on the receiver and its characteristics.

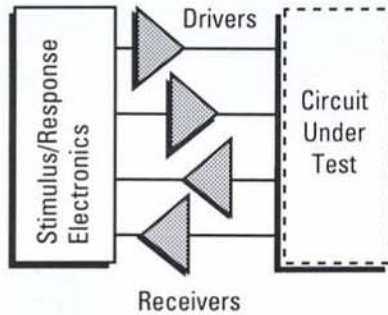


Figure 1. Block diagram of a system that uses a receiver.

In this example, the receiver is actually a comparator whose inputs are a signal from the circuit under test and a programmable threshold. To properly set the threshold, important characteristics of a comparator must be understood. Setting the threshold far from the logic levels makes the

receiver respond rapidly. Setting the threshold close to the logic levels avoids false triggering. A trade-off must be made between speed and reliability of the sampled signal. Making the proper trade-off requires knowledge of the receiver's characteristics.

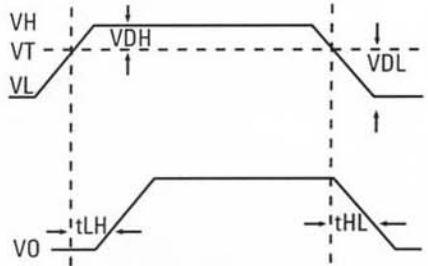


Figure 2. Timing diagram for receiver signals.

Figure 2 illustrates the signals involved. The top waveform is the signal from the circuit under test. The bottom waveform is the output of the receiver. VT is the threshold. VDH is the overdrive voltage for positive edge signals; VDL is the overdrive for negative edge signals; and tLH and tHL are the corresponding delays.

The Solution

Instruments that might be used to measure how delay varies with overdrive voltage are illustrated in **Figure 3**. The HP 6621 is a power supply used to power the receiver. The HP 3314 and HP 3325 are function generators. The HP 3325 supplies a DC voltage which simulates the programmable threshold. The HP 3314 generates a square wave simulating the signal from the circuit under test. Finally, the HP 5334 is a universal counter used to measure the time interval from the input edge to the output edge. All the instruments have an HP-IB interface.

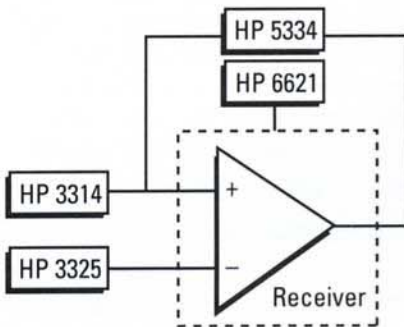


Figure 3. Schematic of a test configuration.

Configuring the Instruments

Before the actual solution is written, HP VEE-Test must have information about which instruments are connected, what their addresses are, and what parameters are needed to control them. HP VEE-Test then provides objects which correspond to those instruments. The objects can be specified as either instrument panels or direct I/O objects. Instrument panels are objects in the form of soft front panels, and drivers for these panels are available for over 170 HP instruments. Direct I/O objects allow you to interactively define your own I/O transactions and send instrument commands. Both methods will be shown later.

A dialog box is used to set all the parameters needed by HP VEE-Test to configure and control the instruments. Included in this dialog box is the capability to name the instrument panels, instead of having to refer to them by their physical addresses. Additional information can also be given to HP VEE-Test about conformance to various standards, data separators and terminators, appropriate timeout values, and more. Once an instrument is configured, HP VEE-Test remembers the information so the user is freed from remembering minute details.

Initializing the Instruments

Once the instruments are configured, the creation of the actual solution can begin. The first portion of the solution involves setting the instruments to good starting states. In this example, we must set the power supply's voltage and current. The universal counter must be made to measure timer interval and its trigger level and input conditioning must be set. The HP 3314 must be programmed to a square wave, a reasonable frequency, and voltage levels of 0 to 4 into 50 ohms. The function of the HP 3325 must be set to DC only with a DC offset of 1 volt. With these settings, the output of the receiver will toggle and the counter will measure a delay.

Instrument drivers are the perfect tool for this initial setup. You don't need to have any knowledge of the bus programming codes; they are all contained within the instrument driver. The graphic interface (instrument panel) for setting values is friendly and straightforward.

To initialize the power supply, the configured instrument panel for the power supply is placed in the work area. The values for the voltage and current are set by clicking on the data field and typing in an appropriate value. We'll set the voltage to 5 volts and the current to .11 amps. The instrument is now "programmed." The other instruments are initialized in a similar manner.

We want the DC offset of the HP 3325 programmed before the HP 5334 performs a time interval measurement. This sequencing is enforced by connecting the bottom pin of the HP 3325 panel to the top pin of the HP 5334 panel.

The **Logging Alphanumeric** display provides a convenient way of viewing all the measurements from the counter. With each new number, the display is scrolled up and the new number placed at the bottom.

Measuring Both Edges

While Figure 5 is a complete and useful model, we can make it better. The delay of a comparator can be different for positive and negative edges, so the test should handle both cases. Presenting the measurements graphically would also be beneficial. Both of these

enhancements are easy to add. The model in **Figure 6** shows the few changes that are needed.

To do the negative edges, the HP 3325 and HP 5334 panels are cloned. The new HP 3325 is left unchanged. The new HP 5334, however, is set up to trigger on negative edges in both channels. Remember, the cloned panels control the same physical instrument at different times in the model. We need the new panels to control the state of the instrument at a specific time.

The overdrive voltage for negative edges, V_{DL} , is $V_T - V_L$. Since V_L is zero, $V_{DL} = V_T$. The formula object is $V_{DL}/2$. The division by two is still needed for the mismatch in impedances. A sequencing line is connected from the first HP 5334 panel

to the second HP 3325 panel so that the positive edge measurement is made for a specific overdrive and then the negative edge measurement is made with the same amount of overdrive.

When each panel for an instrument panel is executed, HP VEE-Test sends commands which set up the instrument in the state represented by the panel. Because the current state of the instrument is known by HP VEE-Test, a minimum number of commands are sent. Only the DC offset is specified in the panels for the HP 3325. The slope is changed when each panel of the HP 5334 is executed.

Plotting the Data

Plotting data is very easy with HP VEE-Test. The graph in Figure 6 is an X vs. Y Plot. The overdrive voltage is an independent variable, so it is connected to the X input, top pin. The positive edge delay and negative edge delay are dependent variables and are connected to separate Y inputs. As each iteration of the loop is executed, a point for each trace is generated and automatically plotted. By clicking on the Autoscale button on the display, both axes are scaled so the traces fit and fill the display. The labels on the graph have been edited to reflect the data being captured.

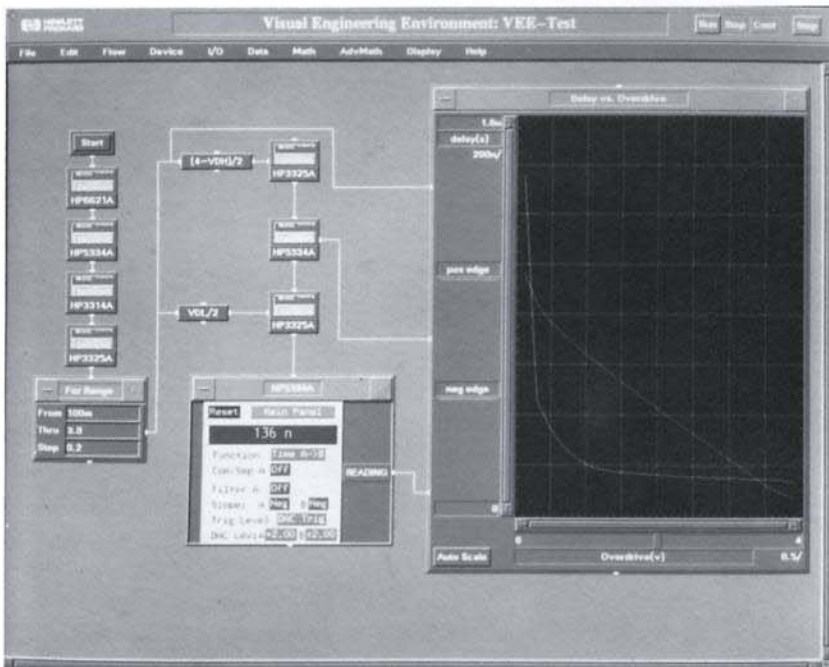


Figure 6. Model for both edges and graphical results.

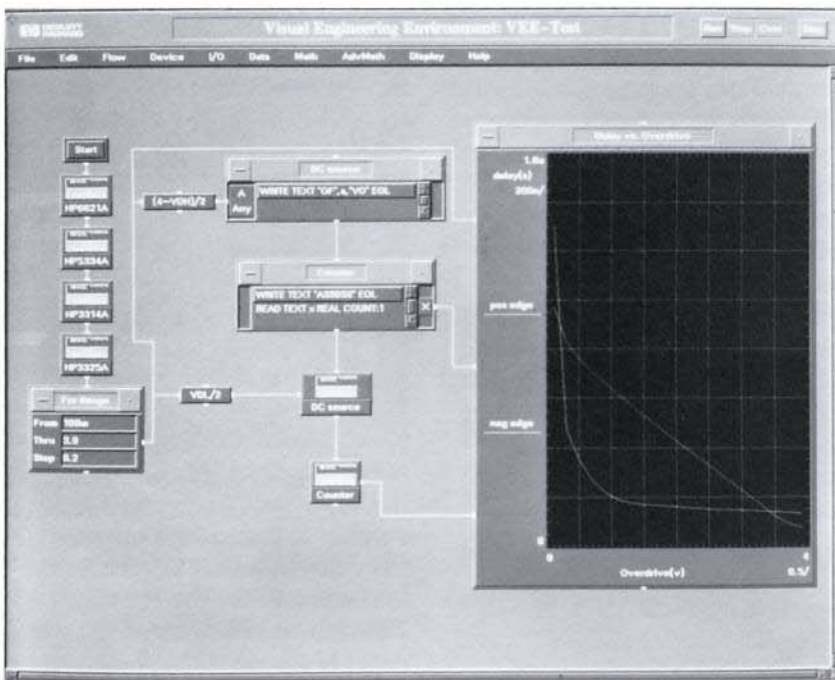


Figure 7. Model using Direct I/O.

Instrument Control Using Direct I/O

While instrument drivers provide much functionality and ease of use, sometimes they are impractical. Because instrument drivers guarantee the reliability of the instrument's state, performance is not maximal. Some instruments in the system may not have an instrument driver. If either maximum performance is essential or the instrument driver is non-existent, another technique may be used to program the instrument. A Direct I/O object lets you explicitly control what is sent to and received from the instrument. To use Direct I/O, however, you must know the effects of all messages sent to the instrument.

In **Figure 7**, the instrument panels in the loop have been replaced with Direct I/O objects. The transactions used to set the DC offset of the HP 3325 and the slopes of the HP 5334 contain the actual codes sent over the inter-

face. The second object for Counter contains a transaction which sends "AS1BS1" to program negative slopes. The measured results, as shown in the display, are the same as when using instrument drivers.

Table 1

| | | |
|------|-----------|-----------|
| 0.10 | 1.591E-06 | 1.000E-01 |
| 0.30 | 4.900E-07 | 3.000E-01 |
| 0.50 | 3.600E-07 | 5.000E-01 |
| 0.70 | 2.780E-07 | 7.000E-01 |
| 0.90 | 2.260E-07 | 9.000E-01 |
| 1.10 | 1.960E-07 | 1.100E+00 |
| 1.30 | 1.850E-07 | 1.300E+00 |
| 1.50 | 1.790E-07 | 1.500E+00 |
| 1.70 | 1.740E-07 | 1.700E+00 |
| 1.90 | 1.690E-07 | 1.900E+00 |
| 2.10 | 1.660E-07 | 2.100E+00 |
| 2.30 | 1.620E-07 | 2.300E+00 |
| 2.50 | 1.600E-07 | 2.500E+00 |
| 2.70 | 1.570E-07 | 2.700E+00 |
| 2.90 | 1.550E-07 | 2.900E+00 |
| 3.10 | 1.520E-07 | 3.100E+00 |
| 3.30 | 1.500E-07 | 3.300E+00 |
| 3.50 | 1.480E-07 | 3.500E+00 |
| 3.70 | 1.450E-07 | 3.700E+00 |
| 3.90 | 1.390E-07 | 3.900E+00 |

Using Direct I/O meant we had to know that the mnemonic "AS1" programs the slope of channel A for negative edges. When we used instrument drivers, this arcane knowledge was not needed. Using Direct I/O, however, did gain us some performance advantages.

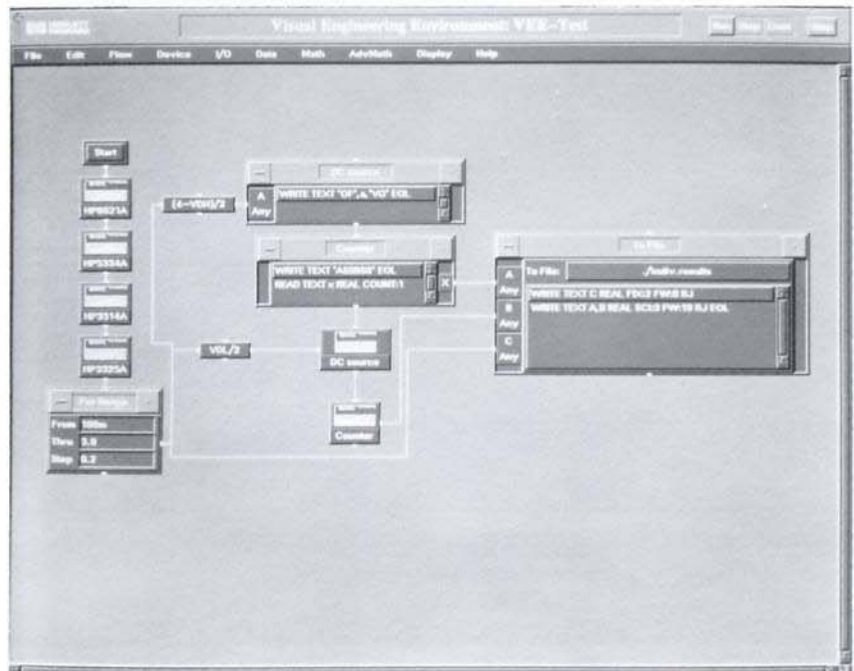


Figure 8. Model storing results to a file.

Storing the Data

Graphic presentation is important and useful. Sometimes, however, the data must be stored in a file either for archival purposes or later analysis. An object, labeled **To File** in **Figure 8**, shows how HP VEE-Test can write data to a file. After both measurements for a given overdrive are made, overdrive, positive edge delay, and negative edge delay are written to the file called "results" as a single line. The name of the file can be changed after each run or the results can be appended to the same file. Many formats are available, so the data can be presented in any form needed. The file recorded when **Figure 8** was run is shown in **Table 1**. Once the file is written, it can be accessed by other tools to further analyze or present the data, including by another HP VEE-Test model.

Testing Multiple Parts

So far, all the models written have been oriented toward an engineer testing a single part. For better characterization, a significant number of units should be tested and the results analyzed using some statistical functions. **Figure 9** is a model which performs measurements and stores the results for any number of parts. It also demonstrates the use of UserObjects (similar to subroutines) and operator interaction.

Like subroutines in traditional programming languages, UserObjects in HP VEE-Test are a means to encapsulate parts of a solution. Underlying complexity can be hidden from the higher level user. The object labeled **Initialize Instruments** is a UserObject containing the solution shown in **Figure 4**. The UserObject labeled **Perform Measurements** encapsulates the objects which perform and store the measurements.

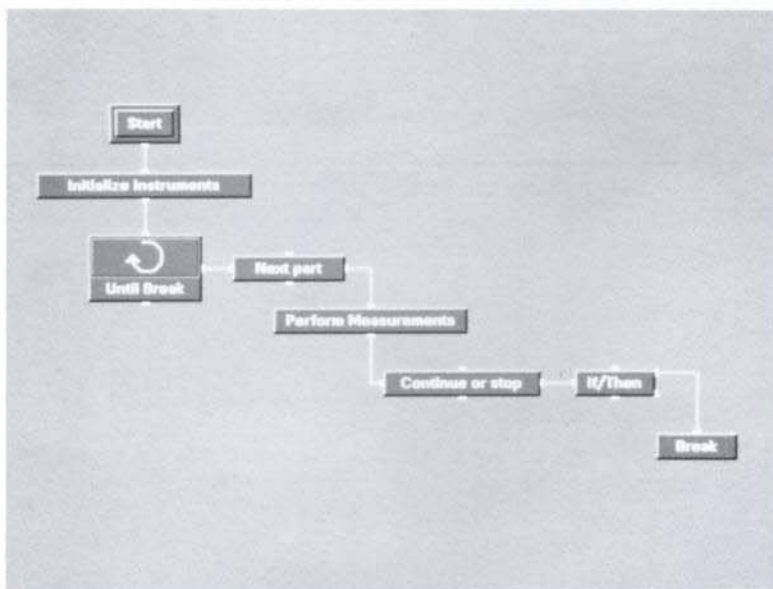


Figure 9. Model for testing multiple parts.

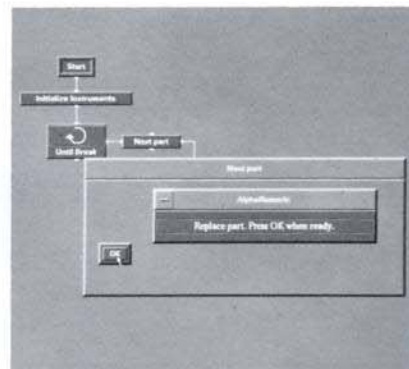


Figure 10. Model illustrating operator interaction.

UserObjects can be configured to display a panel upon execution and the operator can interact with the panel. When the UserObject completes its execution, the Panel disappears. Selected objects are chosen to appear in the panel, typically controls and displays. The underlying connections, iterators, flow control, etc. are usually not included.

The **Until Break** object continues to re-execute the objects attached to its output until a **Break** object is executed. The UserObject labeled **Next Part** is configured to show a Panel when it executes. **Figure 10** shows what the operator sees when **Next Part** executes. The operator installs a new part and presses **OK** when ready. Next, the **Perform Measurements** UserObject, in **Figure 9**, loops through the overdrive values and records the results in a file, as was done in **Figure 8**. The **Continue or Stop User** object contains another panel which asks the operator if more parts remain to be tested. Depending on the answer, either the loop is executed again or the **Break** object is executed, thus terminating the iterator.

Conclusion

The power of HP VEE-Test lies in its balance between ease-of-use and functionality. While this example may have taken days to write in a traditional programming language, it took only a few hours using HP VEE-Test. HP VEE-Test saves time by letting you focus on results rather than on the programming routines to get at the results.

Now, *This* is Engineering!

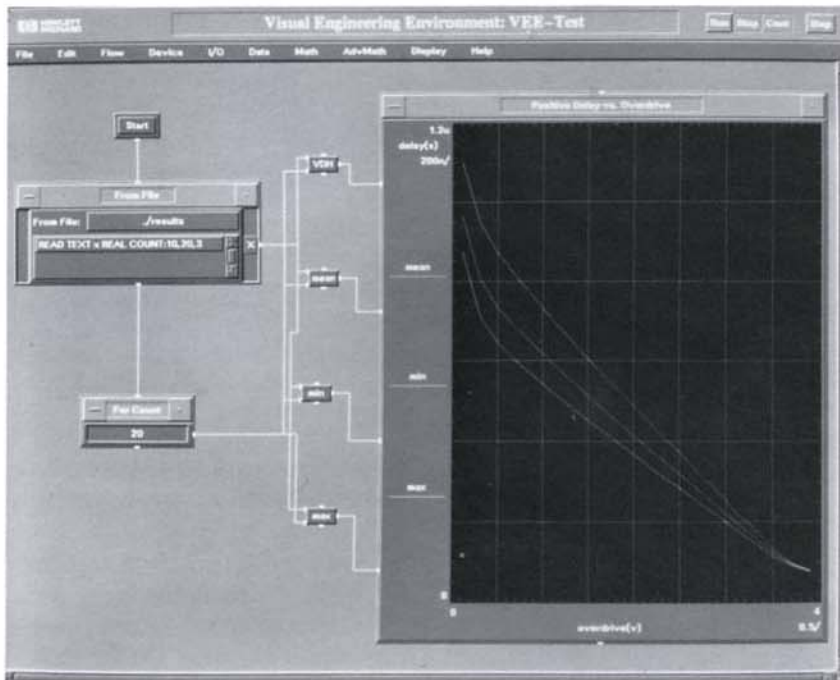


Figure 11. Model for analysis.

Post-Acquisition Analysis

Collecting data is an important aspect of design characterization. Just as important, however, is being able to easily analyze that same data. HP VEE-Test is as useful for analyzing the collected data as it is for collecting the data. **Figure 11** shows how data collected from 10 tested parts can be processed. The **From File** object in Figure 11 reads the file used to store the data collected in Figure 9. The READ transaction reads the entire file into a three-dimensional array.

The **For Count** object cycles through the 20 values of overdrive. The VDH formula object retrieves an actual overdrive voltage given an index. The object labeled **Mean** computes the mean across all parts for an overdrive index. Likewise, the **Min** and **Max** formula objects compute the minimum and maximum values at every overdrive.

The display is similar to the one used in Figure 6. The three traces are the result of processing the positive delay data.

You can now decide on the specifications for the system based on all the acquired data. The graph can also be used to report the data to those concerned about the decision.

For more information, call your local HP sales office listed in your telephone directory or an HP regional office.

Data Subject to Change
Printed in U.S.A. 5/91
Copyright © 1991
Hewlett-Packard Company
5091-1140E