



Three engineers cut development time by using configuration files to store instrument settings.

Save settings, save seconds

MARTIN ROWE, SENIOR TECHNICAL EDITOR

When characterizing a new digital or mixed-signal IC, you often must repeatedly make measurements such as rise time and fall time of digital signals over various temperatures and power-supply voltages. You can simplify these measurements—and shorten test-development time—by following the lead of three engineers who found ways to reuse instrument settings in various tests.

With reusable instrument settings, you can develop tests for new devices by selecting instrument setups from

menus rather than by writing a program from scratch. Thus, you can develop tests for characterizing new devices in just minutes rather than in hours.

Each of the three engineers I spoke with uses a set of configuration files and custom instrument drivers to control his test equipment (**Figure 1**).

DUT-based configuration files

Sorin Grama, principal at system integrator Cal-Bay Systems (San Rafael, CA) developed a technique that uses one configuration file per device under test (DUT). (Cal-Bay developed the test system with engineers from National Semiconductor.) Grama organizes the configuration file, which contains all the information needed to test a device, into groups called “signals,” “hardware sets,” and “test sets.” **Figure 2** provides the details for a simple configuration file.

The signals section contains information such as a digital signal’s logic levels. Two hardware sets, called “vsets” and “hsets,” represent the vertical and horizontal settings of an oscilloscope. Finally, a test set contains a combination of hardware sets that completely define the state of all system hardware.

With Grama’s application, other test engineers can reuse the hardware sets to develop tests for new devices. An engineer simply opens a template file that contains lists of hardware sets and selects the appropriate ones by navigating through menus and dialog boxes.



Sorin Grama and Marvin Kelly operate a station for testing mixed-signal devices. Courtesy of Cal-Bay systems and National Semiconductor.

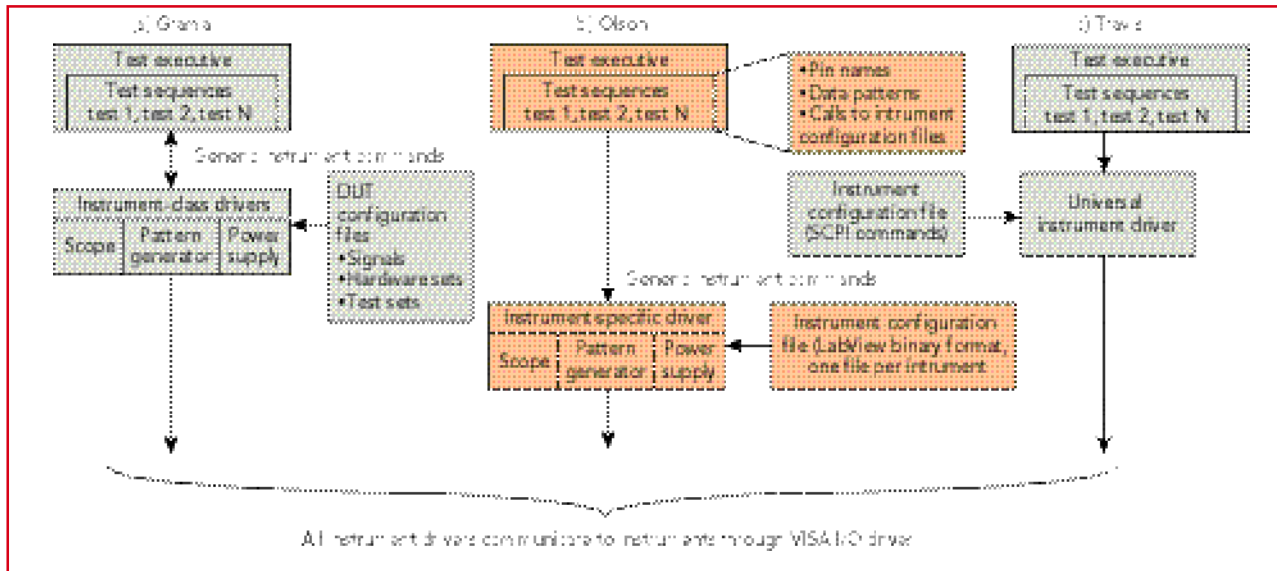


FIGURE 1 (a) Test systems developed by Cal-Bay Systems use configuration files that contain all the information needed for a system to test a device. (b) At Philips Semiconductor, configuration files store instrument setups for use with an instrument-specific driver. (c) Engineers at AMCC use a universal instrument driver and configuration files that contains instrument-specific commands.

Test engineers can also develop new hardware sets by setting the instruments manually and polling them through their IEEE 488 ports to get their settings. Grama’s application automatically creates the configuration file from the instrument settings it retrieves.

In his hardware sets, Grama’s uses generic names to describe each instrument setting, freeing test engineers from having to learn the instruments’ command strings. The generic names also let test engineers substitute instruments within a class for each other, so there’s no need to change a configuration file when changing an instrument.

When running a test with Grama’s application, an engineer selects a DUT, and the application opens the appropriate configuration file. The application then parses the configuration file and passes the information to a test executive, which executes the tests by programming the hardware settings into the instruments using the appropriate instrument-class driver (Figure 2a). The driver, which contains command strings, formats the strings for use with the virtual instrument software architecture (VISA) I/O driver.

Instrument and test-based files
 Test engineer Dave Olson from Philips Semiconductor (Albuquerque, NM) developed an application called “AcBench”

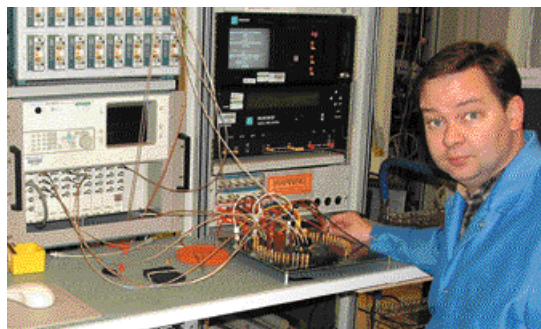
that lets test engineers characterize digital ICs used in PC motherboards. AcBench stores test and instrument data in separate files that provide data to instrument-specific drivers (Figure 1b).

Where Grama’s application uses one file per DUT, Olson’s AcBench uses one file per instrument per test setup, regardless of the DUT. AcBench organizes instrument data based on the instrument and the test it must perform. The application creates file names that indicate the type of test and the equipment used in that test. For example, for a rise-time test, Olson uses files such as Trise.scf (oscilloscope), Trise.pls (pulse generator), and Trise.psp (power supply). Olson or another test engineer can use these configuration files to perform rise-time tests on similar DUT pins under different thermal and power conditions.

Rather than use plain text files like the other two applications in this article, AcBench stores instrument setups in native LabView binary files. Olson explains, “I chose the binary format because LabView already knows how to extract data from native configuration files. I didn’t

need to write a parsing routine to get the instrument-setup data.”

When a test engineer at Philips develops a test, he or she creates a list of tests for each DUT pin. A test may contain a data pattern of 1’s and 0’s needed to place the part in a given operating mode. By defining each test pattern with a unique name, a test engineer can assign any pattern to more than one test. Therefore,



Dave Olson developed software that lets test engineers reuse settings and drivers from numerous instruments. Courtesy of Philips Semiconductor.

AcBench lets test engineers reuse data patterns as well as instrument setups.

When AcBench begins a test, the system prompts each instrument driver to load the appropriate configuration file. The instrument driver compares the configuration file’s settings to the previous file’s settings and transmits only those set-

tings that differ. Thus, the system minimizes traffic on the IEEE 488 bus.

Universal driver

At Applied Micro Circuits Corp. (AMCC; San Diego, CA), test engineer Joe Travis developed test stands that he and other engineers use to characterize digital and mixed-signal ICs. Through Travis's test executive, a test engineer can run tests by selecting from a library of DUTs and tests.

Rather than separate the test setups and instrument commands into separate configuration files and drivers the way Olson did, Travis developed a universal instrument driver (Figure 1c) that works with any instrument. The configuration file delivers instrument-specific information to the universal driver.

The LabView code in Figure 3 shows how Travis used the universal driver to initialize the test equipment. The driver, represented by a LabView virtual instrument (VI) box marked "GPIB," knows which configuration file to use based on a text box containing an instrument name.

Where Grama chose to put test strings in instrument-class drivers and Olson put them in instrument-specific drivers, Travis's configuration files contain complete SCPI test strings that they send to the driver. This approach requires fewer files than the other two, but the configu-

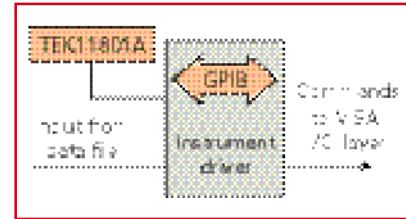


Figure 3 AMCC's universal instrument driver uses a text box to identify configuration files for the driver. *Courtesy of AMCC.*

ration files are far more complex. Travis builds his text-based configuration files manually, by writing SCPI command strings from an instrument's programming manual.

Travis's universal driver and configuration files let him substitute instruments, depending on which are available. For example, his Tektronix 11801A configuration file will work with a Tek 11801B, 11801C, or CSA803 oscilloscope. The configuration file also contains sections for bus addresses and for measurement setups. A section that sets up the instrument to measure deterministic jitter contains 29 lines of commands. Other setups, such as one that tests a reference clock, contain many more lines.

All query the bus

All three engineers take advantage of instrument queries to build configuration files. With any of the three methods, once

```
signals]
signal1 = "name=CLK, comment=input clock, type=Input, group=TTL"
signal2 = "name=INO, comment=TTL Input 0, type=Input, group=TTL"
signal4 = "name=OUT0, comment=LVDS Output 0, type=Output,
group=LVDS"

[vsets]
vset1 = "name=LVDS Vertical Set, comment=200 mV, vertRes=0.200000,
offset=0.000000"
vset2 = "name=TTL Vertical Set, comment=500 mV, vertRes=0.500000,
offset=0.900000"

[hsets]
hset1 = "name=test1 hset, comment=10 nsec, timebase=1.000000E-8,
pos=4.500000E-8"
hset2 = "name=RiseTime hset, comment=200 psec, timebase=2.000000E-10,
pos=6.470000E-8"

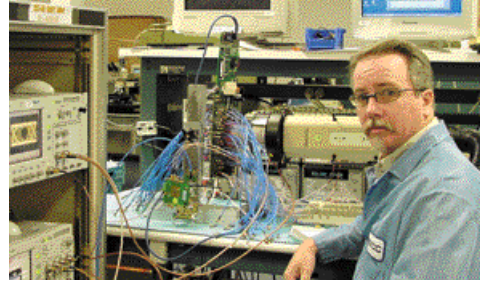
[tests]
test1 = "name=TCLK Freq; type=Frequency; comment=none; signals=signal1;
vsets=vset2; hsets=hset1;
test2 = "name=OUT0 RiseTime; type=RiseTime; comment=none; signals=signal4;
vsets=vset1; hsets=hset2;
```

FIGURE 2 A configuration file contains generic instrument settings organized into hardware sets such as vertical sets and horizontal sets for an oscilloscope. *Courtesy of Cal-Bay Systems.*

an engineer manually configures the test equipment, he or she can click a button, and the application will automatically build the configuration file. For example, an engineer who wanted to set up a rise-time test would adjust the pulse generator to correctly stimulate the part, adjust the scope to show the rising edge, enter the reference levels for the measurement, enter a name for the setup, and then click on a button. The application will query the instrument and stores its settings in a configuration file.

Grama and Olson met during NI Week 2002 (www.niweek.com), when each was presenting a paper about his work. They discussed their applications and offered each other advice. When I learned that Travis was working on a similar application, I put him in contact with the other two. Again, they were able to share ideas and learn from one another.

If you'd like to learn more about these approaches to reusable instrument set-



AMCC's Joe Travis operates a test system to test Serdes devices. This rack tests OC-48 components. *Courtesy of AMCC.*

tings, contact me, and I'll forward your questions to the appropriate engineer. E-mail: m.rowe@tmworld.com. T&MW

Martin Rowe has a BSEE from Worcester Polytechnic Institute and an MBA from Bentley College. Before joining T&MW in 1992, he worked for 12 years as a design engineer for manufacturers of semiconductor process equipment and as an applications engineer for manufacturers of measurement and control equipment. E-mail: m.rowe@tmworld.com.
