

idly plotted or re-plotted against different interpretations just by changing pointers to columns.

A key advantage is taking data on a carefully controlled grid, which allowed a single data set to be post-analyzed many different ways. Another innovation was building a software tool to provide very tight coupling between the generator and the recorded data prod-

uct, and the operator's worksheet. Together, these allowed the operator to sweep the jitter stimulus quickly along any of three dimensions and focus on the response of the system under test (response was jitter transfer ratio, or performance degradation to the symbol or codeword error rate). Additionally, managing multi-tone and noise waveforms automated a tedious manual process,

and provided almost instantaneous decision-making control over test flow. The code was written in LabVIEW, and calls Agilent instrument drivers to write to the generator hardware.

This work was done by Chatwin Lansdowne and Adam Schlesinger of Johnson Space Center. Further information is contained in a TSP (see page 1). MSC-24814-1

μShell Minimalist Shell for Xilinx Microprocessors

NASA's Jet Propulsion Laboratory, Pasadena, California

μShell is a lightweight shell environment for engineers and software developers working with embedded microprocessors in Xilinx FPGAs. (μShell has also been successfully ported to run on ARM Cortex-M1 microprocessors in Actel ProASIC3 FPGAs, but without project-integration support.) μShell decreases the time spent performing initial tests of field-programmable gate array (FPGA) designs, simplifies running customizable one-time-only experiments, and provides a familiar-feeling command-line interface. The program comes with a collection of useful functions and enables the designer to add an unlimited number of custom commands, which are callable from the command-line. The commands are parameterizable (using the C-based command-line parameter idiom), so the designer can use one function to exercise hardware with different values. Also,

since many hardware peripherals instantiated in FPGAs have reasonably simple register-mapped I/O interfaces, the engineer can edit and view hardware parameter settings at any time without stopping the processor.

μShell comes with a set of support scripts that interface seamlessly with Xilinx's EDK tool. Adding an instance of μShell to a project is as simple as marking a check box in a library configuration dialog box and specifying a software project directory. The support scripts then examine the hardware design, build design-specific functions, conditionally include processor-specific functions, and complete the compilation process. For code-size constrained designs, most of the stock functionality can be excluded from the compiled library.

When all of the configurable options are removed from the binary, μShell has an unoptimized memory footprint of

about 4.8 kB and a size-optimized footprint of about 2.3 kB. Since μShell allows unfettered access to all processor-accessible memory locations, it is possible to perform live patching on a running system. This can be useful, for instance, if a bug is discovered in a routine but the system cannot be rebooted: μShell allows a skilled operator to directly edit the binary executable in memory. With some forethought, μShell code can be located in a different memory location from custom code, permitting the custom functionality to be overwritten at any time without stopping the controlling shell.

This work was done by Thomas A. Werne of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47495.

Software Displays Data on Active Regions of the Sun

Lyndon B. Johnson Space Center, Houston, Texas

The Solar Active Region Display System is a computer program that generates, in near real time, a graphical display of parameters indicative of the spatial and temporal variations of activity on the Sun. These parameters include histories and distributions of solar flares, active region growth, coronal mass ejections, size, and magnetic configuration.

By presenting solar-activity data in graphical form, this program accelerates, facilitates, and partly automates what had previously been a time-consuming mental process of interpretation of solar-activity data presented in tabular and textual formats. Intended for origi-

nal use in predicting space weather in order to minimize the exposure of astronauts to ionizing radiation, the program might also be useful on Earth for predicting solar-wind-induced ionospheric effects, electric currents, and potentials that could affect radio-communication systems, navigation systems, pipelines, and long electric-power lines.

Raw data for the display are obtained automatically from the Space Environment Center (SEC) of the National Oceanic and Atmospheric Administration (NOAA). Other data must be obtained from the NOAA SEC by verbal communication and entered manually. The Solar Active Region Display System

automatically accounts for the latitude dependence of the rate of rotation of the Sun, by use of a mathematical model that is corrected with NOAA SEC active-region position data once every 24 hours. The display includes the date, time, and an image of the Sun in H α light overlaid with latitude and longitude coordinate lines, dots that mark locations of active regions identified by NOAA, identifying numbers assigned by NOAA to such regions, and solar-region visual summary (SRVS) indicators associated with some of the active regions.

Each SRVS indicator is a small pie chart containing five equal sectors, each