SYNOPSYS®
Accelerating Innovation

Synopsys® SPW

The Fastest Path from Innovation to Implementation of Digital Signal Processing Systems

Highlights

- *Faster innovation for digital signal processing based products*
- *More than 3,000 models included, all in source code*
- *Standard-specific end-to-end system configurations*
- *Reduction of design effort by modeling with CDF*
- *Drastic design cost reduction through systematic reuse*
- *Best ROI in the industry for DSP system simulation*
- *Proven hardware implementation path*
- *RTL cosimulation interface*

Overview

Synopsys® SPW (Signal Processing WorkSystem) is the fastest path from innovation into implementation for digital signal processing systems, applying a model-based design approach. At its core is the C data flow (CDF) modeling paradigm which enables the most efficient description of digital signal processing systems which may be implemented in dedicated digital hardware or embedded software. CDF is an intuitive way of describing highly-parallel systems which can produce a fully deterministic execution schedule, critical for the design of today's heterogeneous, multi-core platforms. Implementing complex digital signal processing systems for consumer, infrastructure, medical, automotive and aerospace and defense companies is the key challenge for innovation, since most of the design time is spent on investigating the effects of individual implementation decisions on the performance of the entire system. Pure language-based approaches (MATLAB® or C/C++) fail at these challenges as they do not constrain the modeling approach enough to improve implementation and simulation productivity. SPW comes with a rich set of model libraries and containing more than 3,000 blocks, all included with the base configuration. It also includes standard specific end-to-end simulation setups configured according to the standard specification. All models come in source code, so they can be used as a starting point for modifications.

Core Technologies

The core technologies in SPW lie in the transformations of the CDF model for simulation and implementation in hardware and software. CDF models can capture the entire range of digital processing systems. The modeling paradigm includes static data flow (SDF) as well as dynamic data flow (DDF). In addition, a hierarchical finite state machine (FSM) based on C expressions is available. All models are graphically integrated into hierarchical subsystems, which make the entire design very transparent to individual developers as well as larger design teams, including the key system architects responsible for the entire system specification.

The simulation technology takes full advantage of the CDF modeling paradigm. It allows for optimum scheduling of SDF models by suitable clustering and code optimization within the clusters. Many systems may also contain DDF subsystems which are automatically scheduled along with the CDF and FSMs into one optimized simulation executable. All of this happens without user intervention and results in simulation performance superior to other model-based approaches as proven by many competitive benchmarks performed by our customers.

CDF models take full advantage of polymorphism. A single model can have a floating-point, integer, fixed-point, complex, vector, or matrix (and their combinations) representation which are available to the user through either the graphical user frontend or through the scripting interface for the simulation engine. The implementation of the data types is highly optimized so that simulation performance for mixed floating-point/fixed-point system exploration is maintained.

SPW provides unlimited scalability, without the need for user intervention. Every user can take advantage of multi-core servers to accelerate an individual simulation by a factor almost linear to the number of processors, which is automatically enabled by the partitioning technology inside the SPW scheduler. For larger exploration or verification runs, every user can start as many simulations automatically from his or her SPW frontend as possible within the limitations provided by the server farm. Commercial load sharing programs can become a layer for simulation job optimization between the user and the server farm.

The CDF simulation technology allows for efficient integration of other modeling and simulation technologies as well. MATLAB legacy models are interpreted using their own native engine through a highly optimized, thin interface, and because of the data flow paradigm, simulation will only slow down if the MATLAB function is a significant part of the system. HDL simulators are integrated through direct co-execution where the CDF simulation is a part of the HDL design, thereby eliminating the communication overhead required by other co-simulation techniques such as interprocess communication (IPC). This enables the most efficient reuse of testbenches and reference models developed with SPW within a hardware verification flow, significantly improving overall verification efficiency. Finally, as design teams embrace transaction based verification using SystemC TLM as their standard interface, SPW produces a self-contained SystemC TL model that adds the SystemC calls on top of the highly optimized CDF simulation executable.

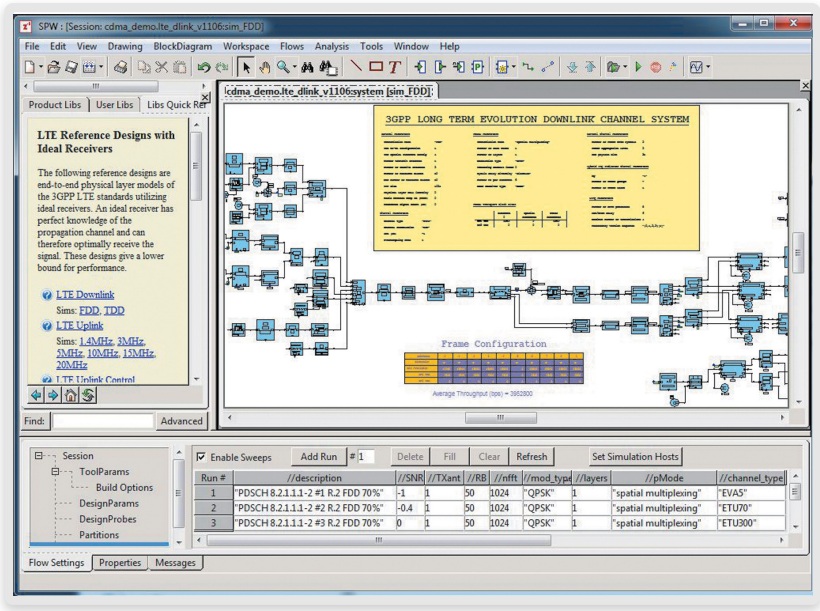

Figure 1: Library browser, LTE reference system and simulation sweep table

The CDF implementation technology for hardware components complements the RTL code generated from the rich HDS (Hardware Design System) library model set. Specific attention is paid to optimize the implementation for each and every arithmetic fixed-point operation considering the specification given by the polymorphic type selection. The rapid hardware implementation flow extension to HDS adds a direct CDF to RTL translation technology. This allows the user to switch between floating-point, bit-true cycle-accurate fixed point and RTL models with one block. This is very attractive to users who prefer coding their models rather than capturing them graphically from the HDS library. Compared to high level synthesis, this flow allows for very straightforward verification and tight control on quality of results.

RTL cosimulation interfaces are already included in the base configuration, enabling a seamless reuse of RTL in a system simulation, as well as an efficient reuse of system-testbenches during the RTL verification phase.

Key Features

Models

- SPW comes with the richest set of models, all included in the standard product package for no additional cost, and provided in source code. For a detailed listing of all the models included with SPW, ask for the SPW model list
- Signal processing base library: noise generation, filter, analysis, modulators, etc.
- Communications library: modulators, demodulators, equalizers, channel models etc.
- Wideband CDMA (3GPP) library: 3GPP standard compliant, includes models and complete end-to-end system setups according to the standard specifications
- TD-SCDMA library: complements the Wideband CDMA library by adding support of the Low Chip Rate (LCR) Time Division Duplex (TDD) of the 3GPP standard

- GSM/GPRS/EDGE library:standard compliant models,
- WiMAX library: 802.16 compliant, includes models and complete end-to-end system simulation setups
- CDMA2000 (3GPP2) library
- WLAN/WPAN library

Filter Design System

- Interactive design, simulation and analysis of FIR and IIR digital filters
- Automatic computation of the filter coefficients, based on a set of design specifications
- Provides extensive filter design methods that include classical and non-classical IIR and FIR design methods (including Bessel, Butterworth, Chebyshev and more) as well as user-defined filters
- Plot of frequency response and impulse response
- User may also convert the filter from floating-point to fixed-point and view the impact on the filter response
- Generated filters can be used in simulation, as well as in the Analysis User Interface

Simulation

- Stream-driven data flow, compiled simulation
- Optimizing static scheduler for optimized simulation performance
- Multi-rate simulation and dynamic scheduling support without user intervention
- Support for server farms running Grid Engine and LSF, extendable to other load sharing facilities
- Multi-threaded simulation for multi-core hosts
- Scripting with dynamically changeable parameters
- Parameter import from spreadsheets
- Switching off parts of a design during simulation
- Statistically equivalent, parallelized bit-error-rate simulations

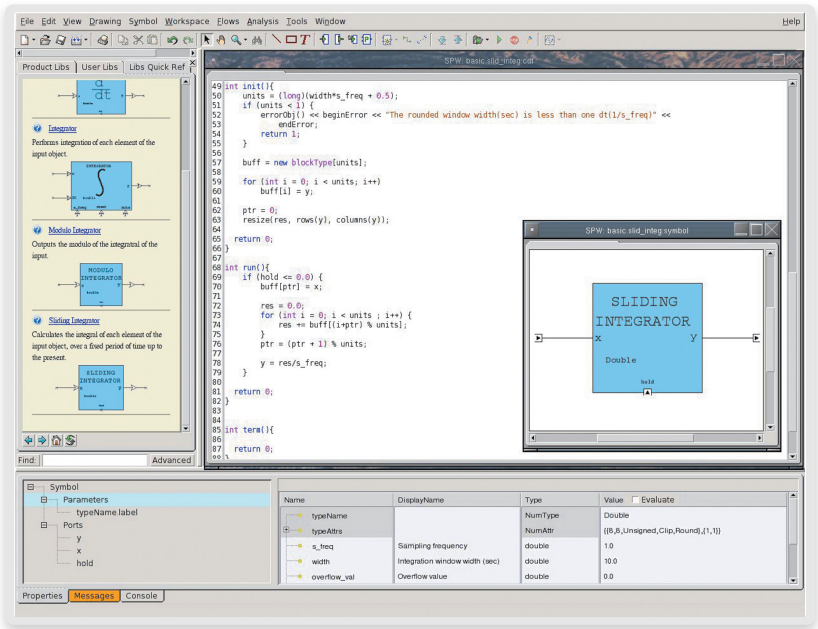

Figure 2: Symbol, parameter and source code view of basic polymorphic model

- Hierarchical simulation profiler for system optimization
- Co-simulation interface with VCS and other RTL simulators
- User-friendly error message connected to cause of errors
- Support for 64-bit simulation to cater to very large designs

C Data Flow (CDF) Modeling

- Signal processing functions operating across a wide range of different polymorphic data types
- Packages the power of C++ into a C-like environment with a simple structured modeling approach
- Parameters can be defined using complex functions provided by a C-compliant parameter expression language, supporting the single model simulation approach
- Built-in source code editor or user-defined external editor

Graphical Design Environment

- Microsoft® Windows style graphical user interface with supported tabbed-design panes for compact design representation and easy navigation
- High connectivity between different design panes eases the tasks of assembling, parameterising and simulating a system
- Selecting an object in one pane will cause the other panes to select the same object view
- Small number of clicks to perform commonly-used design tasks
- Full flexibility for symbol editing, colors and bitmaps to generate high quality documentation of systems
- Tooltips for instances in designs, models in the library browser and simulation parameters

Scalable Standard Data Model

- Standard XML views for hierarchical models with clean separation of logical and graphical information
- Database scalable for enhancements with more attributes, data type and parameters for future design flow requirements
- Multiple, static manipulations and analysis of the database can be performed through Tcl and C++ programming interfaces

Block Diagram Editor

- Easy graphic system capture
- Unlimited undo/redo operations
- Hierarchical block diagram environment allows easy viewing of an entire system, including system parameters and source code
- Block terminals indicate both signal direction and connectivity status
- Acrobat-style pan mode with additional pan-overview window facilitating viewing of complex designs
- Hierarchy refactor for collapsing a group of models into one block

Library Browser

- Provides overview of libraries, models and design hierarchies with short descriptions and symbol views, logical views and structural views
- Powerful search function across entire product and user data base
- Direct link from library browser into the model documentation

Analysis

- Interactive analysis supported by predefined analysis widgets (multitrace oscilloscopes, signal generators, and spectrum analyzers, eye, scatter, x-y, parameters)
- Post-simulation analysis with rich set of plot functions and post-processing and mathematical transformations
- Very powerful C-based scripting of post-processing tasks, enabling subsequent one-click analysis
- Bit error rate (BER) and block error rate (BLER) curves are automatically merged from a large number of simulations conducted on multiple machines
- Analysis UI handles manipulation of a variety of data types, including floating-point, fixed-point and complex
- Signals and data attributes can be collated and stored in a single file for analysis
- Virtual control panels to mimic real-world interfaces. Changing parameters during simulation/debug

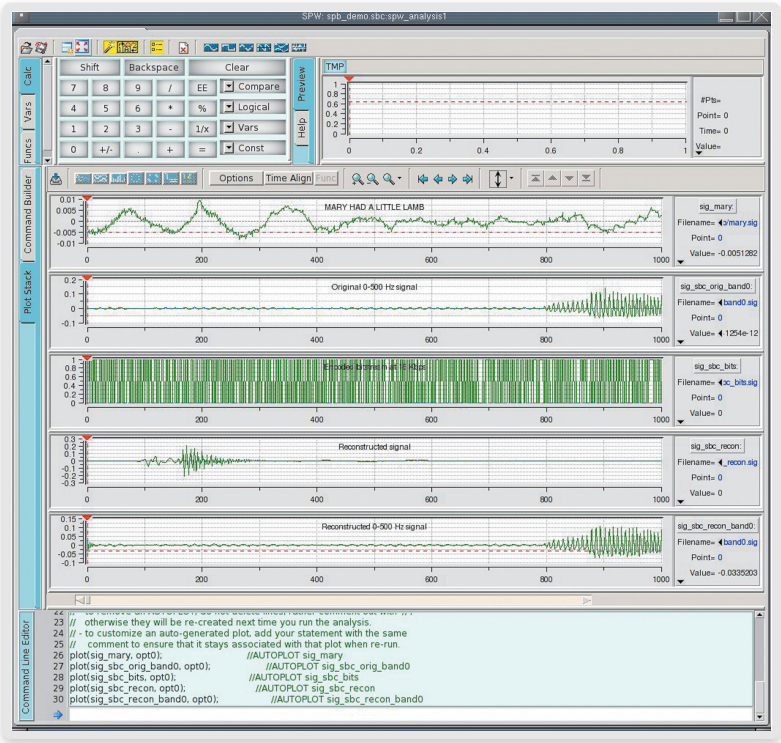

Figure 3: Post processing analysis with signal calculator, plot window and scripting language

- Simulation output can be automatically redirected into MATLAB analysis

Model Wizard

- Straightforward infrastructure to create templates for new blocks, which can be either user-written polymorphic models in C, MATLAB functions, legacy C-functions, or any other language with a C interface (e.g. Tcl/Tk)

Workgroup/Data Management/ Infrastructure

- Clean hierarchy of model management with workgroups, libraries and models
- Control access and modifications as well as revision history and track changes option with CVS, SVN and Clearcase
- Accommodates other revision control infrastructures
- Simulation control functions via scripts in addition to the GUI
- Easy archival of design data for distribution to other users

Available Options

- Hardware Design System

Available Add-on Libraries

- LTE/LTE-A Library
- LTE Library for Xilinx IP

Supported Platforms

- Windows and Linux

Customer Focus

Synopsys provides a complete range of training, support, design methodology consulting, and integration services. Technical support requests are handled directly by experienced design engineers who are fully familiar with the application of Synopsys tools and methodologies to real-world designs. Training courses are available at Synopsys offices or at the customer site and can be tailored to meet the specific needs of the design team.

For more information about SPW, visit us on the web at www.synopsys.com/dsp, contact your local sales representative or call 650.584.5000.

SYNOPSYS®
Accelerating Innovation

Synopsys, Inc. • 700 East Middlefield Road • Mountain View, CA 94043 • www.synopsys.com

05/12.CE.CS3146.