Hardware Design System (HDS)

HDS Provides First Time Success for Signal Processing Algorithms into Hardware

Overview

CoWare® Signal Processing Designer HDS is a key component in the SPD product family. It accelerates the hardware design, verification, and analysis of complex, algorithm-intensive digital-signal processing (DSP) systems. The SPD HDS flow provides graphical RTL design capabilities with parameterized design capture that enable reuse. It also includes a large library of complex system building blocks that contributes to substantial reductions in product development time. Finally it provides DSP design content captured and validated with SPD at the data flow and implementation level to the verification engineers for integration and test.

With its unique mixed-level simulation, verification, and analysis capabilities, the SPD HDS flow dramatically improves designer productivity by creating an integrated design environment for signal-processing intensive designs.

A common model is used to specify all attributes of the target systems including multimedia, wireless, wired, or optical communications — SPD HDS for the architectural attributes and SPD for the behavioral attributes. The HDS fixed-point datatype can be used both to explore system design scenarios quickly, and to implement RTL for the IC and FPGA with optimized bit widths so the hardware consumes minimal silicon and power.

Shared testbenches and customizable analysis tools such as waveform viewers are supported through the entire design flow to enhance productivity. Design reuse is assured by parameterized design capture at both the system and RTL level.

Mixed-Level Design Environment

SPD HDS provides the bridge between the data flow design level handled by SPD and the architectural, RTL and ASIC or FPGA implementation levels. This means that all attributes of the system specification can be designed using one model. HDS has building blocks for the architectural level, graphical RTL capture of datapath portions of the design, and textual RTL for control-oriented aspects.

Mixed Model Import

SPD and SPD HDS offer simulation capabilities for models written in C/C++, SystemC, Verilog®, VHDL, VHDL/Verilog-AMS, Xilinx COREGen, or MATLAB to compose your systems. This gives engineers great flexibility and facilitates reuse of IP.

Mixed-Level Verification

Simulation support for both fixed and floating-point dataflow means that system options can be analyzed quickly in SPD with floating-point dataflow. After a system option is chosen in SPD, the mixed-level verification capability allows SPD HDS to use fixed-point dataflow to create a C-based, cycle accurate clocked RTL design.

Mixed-Level Analysis

SPD HDS lets designers use the most appropriate waveform viewers/analysis tools and source text debuggers for each level of design. For example, analysis tools may be optimized for logic-oriented debugging at the RTL level.

Design Integration Links

- SPD HDS is the key link for mapping the algorithms created in SPD into a hardware implementation
- SPD link to Cadence's Incisive for Verilog/VHDL verification and cross-debugging
- SPD link to Mentor Graphics' ModelSim for Verilog/VHDL verification
- SPD link with Synopsys' DC Ultra and Synplicity's Synplify Pro and Certify

Key Features

Fixed-Point Datatype for Efficient Design Analysis

The same datatype can be used at the system-level and the architectural level. The same floating-point datatype that can be used for design, simulation and verification can also

H I G H L I G H T S

- **Bridges the gap between design levels—behavioral/algorithmic, hardware architectural, and ASIC/FPGA—to create the fastest available path from "drawing-board" system-level constructs to silicon**
- **Generates HDL from hardware architecture libraries**
  - **Also generate HDL from generic arithmetic, combinatorial logic, and Mealy state (FSM) blocks in HDS Library and from the AHDS datapath blocks**
  - **HDL can also be imported into SPD for integrated verification and analysis of hand-crafted and legacy code**
- **Enables IP reuse by providing parameterized design capture**
- **Graphical design capability**
  - **Lets designers connect complex building blocks from a large and comprehensive signal-processing library**
- **Simplified debugging and monitoring**
  - **Makes it easy to create a system testbench and virtual instruments to probe the testbench**
- **Speeds design cycle and design trade-off analysis by using the same system testbench and virtual instruments as a common behavioral reference at all levels of the design**
  - **Consistent "golden" testbench throughout the design process (floating-point, fixed-point, clocked RTL, and post synthesis)**
  - **Dramatically reduces development and verification time by supporting both bottom-up and top-down flows**
- **Key tool in SPD design flow**
  - **Integrations with other CoWare and third party tools include many new links for analog/mixed-signal tools, new FPGA integration, and bit-error rate simulation as well as links to CoWare Platform Architect, Cadence Incisive, and Mentor Graphics ModelSim, and an enhanced datapath link for better synthesis**
- **Supports SoC, multi-ASIC, and multi-FPGA implementations**

be used at the architectural level by using this flexible fixed-point datatype. The fixed-point datatype can be used at the system-level as well as the architectural level for efficient hardware implementation.

Parameterized Design Capture for Code Optimization
Both the SPDLTI block library and its Micro Architecture Design Library are fully parameterized and generate efficient Verilog and VHDL for logic synthesis. The RTL code generated is optimized for Synplify Pro or DC Ultra.

HDS Library

Signal processing blocks allow the modeling of any fixed-point system by connecting one block to another. Fixed-point signal flow diagrams have the same precision as the target hardware. Multiple clocks typical of multimedia and communications systems are supported, as are most ways of handling overflow.

Completed HDS blocks have a direct path to synthesizable Verilog and VHDL for logic synthesis and this generated code is optimized for efficient datapath synthesis into gates. Users also have control over structure and style including how the hierarchy is generated and the choice of structural or procedural coding styles.

Micro Architecture Design Library for Faster Design
Blocks such as dual-port RAMS, FIFOs, and ALUs can be used from HDS library blocks or RAMs and ROMs can be imported.

Template-based Code Generation
Each HDS building block has an associated template that can generate conditional or parameterized HDL descriptions of the block. The customization available through template-based code means users can create custom libraries and links to datapath tools, memory compilers, and other HDL tools.

Signal Calculator
As a powerful debugging aid, a signal analyzer window is available to display logic states side by side with images of complex waveforms and graphs.

Interactive Simulation Blocks
The HDS library includes blocks that can be used to construct "virtual control panels" that mimic real-world interfaces. This feature allows designers to change simulation parameters easily as they debug their system.

Direct Co-Execution Link
Verilog or VHDL code can be linked to a symbolic block that can be used in a behavioral diagram. Simulation of code associated with the HDL blocks runs on the simulator while the rest of the design is simulated in C-code. This link enhances third-party integration efforts. Cross-selection of signals between the block diagram and the HDL environment eases debugging significantly.

Key Library Blocks

Bit Manipulation
Bit Fill, Bit Merge, Bit Modify, Bit Select Fixed, Bit Select Variable, Bit Split, Word Merge, Word Split

Clock
External, Gated, Generated

Communication
Correlators (Simple Correlate and Dump, Complex Correlate and Dump), Filters (FIR, Complex FIR), Despreaders (Simple Despreader and Complex Despreader) and Sequence Generators (PN Sequence Generator, Walsh Sequence Generator, OVSF Code Generator)

Control
Compare, 6-Function Compare, Vector Compare, Counter, Counter Asynch Reset, Mealy State Machine, Mux, Vector Mux, Switch In, Switch Out, Threshold

Datapath
Absolute Value, Add Carry, Add/Sub Carry, Barrel Shifter, Bulk Delay, Compare, Decoder, Decrement, Encoder, 6-Function Compare, Increment, Increment/Decrement, Multiply, Multiply Pipeline, Multiply & Add, Scalar Generic Arithmetic, Generic Arithmetic, Shift Left, Shift Right, Shift Right Arithmetic, Square, Subtract Borrow, Vector Adder

Logic
AND, Vector AND, AND-OR-Invert, Vector AND-OR-Invert, Buffer 3-State, Combinational Logic, D Flip-Flop, D Flip-Flop Asynch Clear/Preset, Inverter, Vector Inverter, Inverter 3-State, Latch, OR, Vector OR, OR-AND-Invert, Vector OR-AND-Invert, Tree AND, Tree OR, Tree XOR, XOR, Vector XOR

Math
Absolute Value, Vector Absolute Value, Add, Add Carry, Add/Sub Carry, Vector Adder, Coefficient, Constant, Decrement, Divide, Format, Scalar Generic Arithmetic, Increment, Increment/Decrement, Multiply, Multiply Pipeline, Multiply & Add, Negate, Vector Negate, Pass Through, Vector Pass Through, Priority Encoder, Vector Priority Encoder, Shift Fixed, Shift Variable, Square, Subtract, Subtract Borrow (Common I/O), Synchronous ROM, Unit Delay

Simulation I/O
Dbl to Fxp, Fxp to Dbl, Sink, Reset, Source, Clock To Hold

Vector
Generic Arithmetic, Component Sum, Constant, Dbl to Fxp, Extract Component, Format, Fxp to Dbl, Hold, Insert Component, Join Vector, Mult Pointwise, Reverse, Scalar to Vector, Scale, Sink, Source, Split Vector, Sub Component wise, Sum Component wise, Divide Component wise, Switch In, Switch Out, Tapped Delay Line, Unit Delay, Vector File Constant, Vector to Scalar

Memory
Bulk Delay, Bulk Delay Asynch Reset, Hold, RAM, ROM, Synchronous RAM (Separate I/O), Synchronous RAM

Advanced HDS (AHDS) Library

Starting with SPD 4.9, the Advanced HDS Library combines polymorphism and hardware generation. Blocks in this library are used to simulate double, long and fixed-point data types and generate RTL hardware that can be synthesized by downstream synthesis tools. They provide a smooth transition of the design from system specification to hardware implementation especially for datapath logic.

You can specify the behavior of the block by writing a set of expressions and storing them in an ASCII file called the datapath expression file. The expression file syntax is similar to "C" and supports C/C++ operators/constructs. The RTL generation is supported for a subset of C/C++ constructs, while C simulator (SPB-C) supports more general C/C++ operators.

Key Library Blocks

Advanced HDS Library

Datapath 16, Datapath 8, Datapath 4, Unit Delay

Supported Versions

CoWare Signal Processing Designer HDS is currently supported in all 4.x product versions. With release 2006.1 HDS will be offered on the next generation SPD infrastructure available for modeling, capture, simulation and analysis today in release 2005.2.

Customer Focus

CoWare 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 CoWare tools and methodologies to real-world designs. Training courses are available at CoWare offices or at the customer site and can be tailored to meet the specific needs of the design team.

Sales Offices

CoWare has sales offices in the U.S., Europe, Asia Pacific and Japan. For a complete sales office listing with contact information, visit www.CoWare.com. For technical or sales information call 1-888-CoWare8 or email info@CoWare.com.

The ESL Design Leader

CoWare, Inc.
Corporate Headquarters
1732 N. First Street
San Jose, CA 95112

Main: 408-436-4720
Fax: 408-436-4740
www.CoWare.com