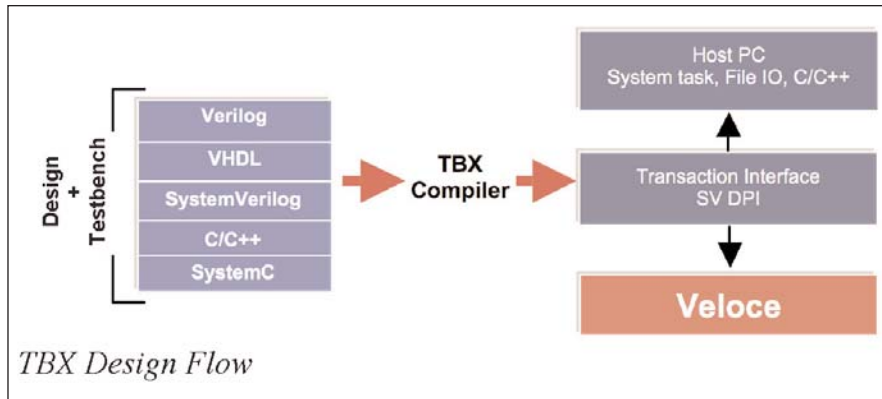


# TestBench Xpress

## Transaction-Based Acceleration

### Hardware-Assisted Verification

### D A T A S H E E T



As SoCs continue to grow in size and in complexity, traditional verification techniques are rendering themselves inadequate to perform system-level verification. The problem is further amplified by the need to verify enormous amounts of embedded software, which are prevalent in today's SoC designs.

TestBench Xpress™ (TBX), from Mentor Graphics along with Mentor's Veloce™ family of hardware-assisted verification products, offer a comprehensive transaction-based verification methodology to address these complex SoC designs.

The TBX transaction-based model utilizes high levels of abstraction modeling (e.g. models written in C/C++ or SystemC) to verify the RTL design inside the Veloce emulator. Instead of communicating at a pin-level or cycle-level, these abstract models communicate using high-level transactions. A transaction is a single or multi-clock cycle data that is encapsulated in a function call and passed from the software side to the hardware side. The "transactors" on the emulator side unroll these transactions into signal level activity for the Design Under Test (DUT). TBX eliminates the latency and bandwidth issues found in other co-verification environments and offers verification performance comparable to In-Circuit Emulation (ICE) while maintaining high fidelity at the interface level.

#### SystemVerilog DPI: SCE-MI 2.0

TBX supports SystemVerilog Direct Programming Interface (SV DPI) to provide the building blocks to create the transaction-based communication link between High-level Verification Language (HVL) and HDL. Based on these functions, tasks, and macro instances, TBX automatically generates a connection between the C or SystemC environment (on a workstation) with HDL design on the Veloce emulator. This connection transparently handles inter-language calls, coordination of threads between SystemC and Verilog, argument translation, and SCE-MI callbacks. This level of automation makes it simple to create a transaction-based verification environment that combines the abstraction and performance of untimed SystemC models with the performance of hardware acceleration.

#### Key Product Benefits & Features:

- Proven third generation transaction-based verification platform
- Up to 10,000X performance boost over SW simulation — up to 1.5 MHz
- 100% Standard compliant (SV DPI/SCE-MI 2.0)
  - Facilitates verification IP reuse
  - Easy integration with 3rd party solutions
- Transaction compiler de-couples software simulator from emulator to accomplish optimal mix of bandwidth and communication latency
- Various clock optimizations allow ICE-like performance and throughput in transaction-based target-less applications
- Interoperable with IEEE's SCE-MI 2.0; backward compatible with SCE-MI 1.1
- "Acceleration-friendly" SV subset (XRTL) provides modeling flexibility with no performance penalty
- Cost-effective scalability: soft models avoid additional board and HW development and eliminate the need to develop target systems with each new product
- Full integration with Veloce runtime and advanced debug provides simulation-like debug environment with full visibility

#### Product Specifications:

**OS Support:** Linux RH 4.0 and SuSE 10

**Veloce Platform Support:** Solo, Trio, Quattro

**Compile Times:** Up to 10-15 MG/hr

**Runtime Performance:** Up to 1.5 MHz

**Language Support:** HDL, SystemVerilog, Verilog RTL, VHDL SystemVerilog, C, C++, System C

**HW/SW Interface:** SV DPI

The basic transaction modeling constructs in TBX are the SV DPI exported and imported tasks/functions. A sample code fragment below shows an exported task. Exported task is a simple Verilog task called from HVL (C-side). The call is initiated by HVL side. It is used for sending transactions that consume simulation time (e.g. an Ethernet frame).

### Extended RTL (XRTL)

TBX extends the synthesizable RTL language subset by providing the ability to synthesize the most commonly used behavioral constructs inside the emulator. XRTL™ empowers designers

with a flexible set of hardware modeling guidelines that offer behavioral-like coding methodology without compromising performance.

### TBX supports the following in XRTL:

- RTL subset
- Initial blocks
- Clock generation & Reset logic using initial blocks
- SV DPI Functions & Tasks
- Events & Waits
- Loops & Conditions
- Implicit state machines
- Multiple drivers
- \$finish & \$display
- \$readmemh & \$writememh

### Hybrid (Transaction-based + In-circuit) Use Model

TBX seamlessly offers the ability to extend the transaction-based methodology to traditional in-circuit emulation environments in a single platform. Static target environments greatly benefit from the ability to access the stimulus/initialization data from the HVL environment.

### Virtual SW Development Platform

TBX offers Virtual SW development capabilities. Users can do embedded SW development on their virtual SoC inside the emulator at emulation speeds without having to wait for the chips to come back from the foundry.

Exported Task Sample Code

```

C Side
//Get the handle and set the scope
svScope my_handle =
svGetScopeFromName("top.exported_task");
svBitVecValData[] = {0x11111111,0x22222222};
for (svBitVecVal i=0; i<8; i++)
{
    ClockCount = i + 2;
    // call exported task
    svSetScope(my_handle);
    TransactionFromSWtoHW(&Data[i], &ClockCount, &CycleCount);
}

HDL Side
Export "DPI-C" task TransactionFromSWtoHW;
task TransactionFromSWtoHW(
input bit[31:0] DataIn,
input bit[7:0] ClockCountIn,
output bit[31:0] CycleCountOut;
begin
    Data = DataIn;
    ClockCount = ClockCountIn;
    CycleCountOut = CycleCount;
end
endtask

```

### SystemVerilog (SV) Functions & Tasks

SV function tasks are the basic building blocks for HW/SW communication in TBX.

- Functions implemented in C and given *import* declarations in SystemVerilog can be called from SystemVerilog. Such functions are referred to as *imported functions* (e.g. *interrupt*)
- Functions implemented in SystemVerilog and specified in *export* declarations can be called from C. Such functions are referred to as *exported functions* (e.g. configuration)
- Tasks implemented in SystemVerilog and specified in *export* declarations can be called from C. Such functions are referred to as *exported tasks* (e.g. PCI Read)
- The SystemVerilog DPI supports only SystemVerilog data types, which are the sole data types that can cross the boundary between SystemVerilog and a foreign language in either direction

Visit our website at [www.mentor.com/emulation](http://www.mentor.com/emulation) for more information.

© 2008 Mentor Graphics Corporation. All Rights Reserved.  
 Veloce, TestBench Xpress, and XRTL are trademarks, and Mentor Graphics is a registered trademark of Mentor Graphics Corporation.  
 All other trademarks mentioned in this document are trademarks of their respective owners.

**Corporate Headquarters**  
 Mentor Graphics Corporation  
 8005 S.W. Boeckman Road  
 Wilsonville, Oregon 97070 USA  
 Phone: 503-685-7000  
**North American Support Center**  
 Phone: 800-547-4303  
 Fax: 800-684-1795

**Silicon Valley**  
 Mentor Graphics Corporation  
 1001 Ridder Park Drive  
 San Jose, California 95131 USA  
 Phone: 408-436-1500  
 Fax: 408-436-1501

**Europe**  
 Mentor Graphics  
 Deutschland GmbH  
 Arnulfstrasse 201  
 80634 Munich  
 Germany  
 Phone: +49.89.57096.0  
 Fax: +49.89.57096.400

**Pacific Rim**  
 Mentor Graphics Taiwan  
 Room 1603, 16F,  
 International Trade Building  
 No. 333, Section 1, Keelung Road  
 Taipei, Taiwan, ROC  
 Phone: 886-2-27576020  
 Fax: 886-2-27576027

**Japan**  
 Mentor Graphics Japan Co., Ltd.  
 Gotenyama Gardens  
 7-35, Kita-Shinagawa 4-chome  
 Shinagawa-Ku, Tokyo 140  
 Japan  
 Phone: 81-3-5488-3030  
 Fax: 81-3-5488-3031



Printed on Recycled Paper