



Verdi & VMM: Components for SVTB Success

Synopsys Interoperability Forum
October 25, 2007

SystemVerilog Evolution

SystemVerilog 3.1a → IEEE 1800-2005

Testbench

Assertions

OO Classes

APIs

Semaphores

Queues and
Lists

SystemVerilog 3.0

Interfaces

Data types
and enums

Structures
and unions

Advanced
operators

Casting,
Control flow

Verilog 2k

MDAs

Generate

Automatic tasks

Verilog 95

Gate-level modeling and
timing

Hardware concurrency

The Path of SystemVerilog Adoption

SystemVerilog Assertion

- Ad-hoc methodology
- Low entry barrier
- Low learning curve

SystemVerilog Testbench

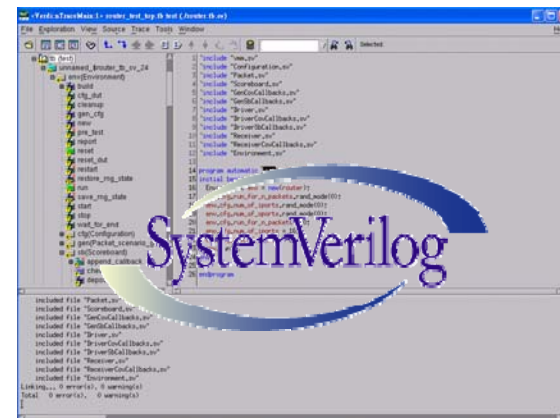
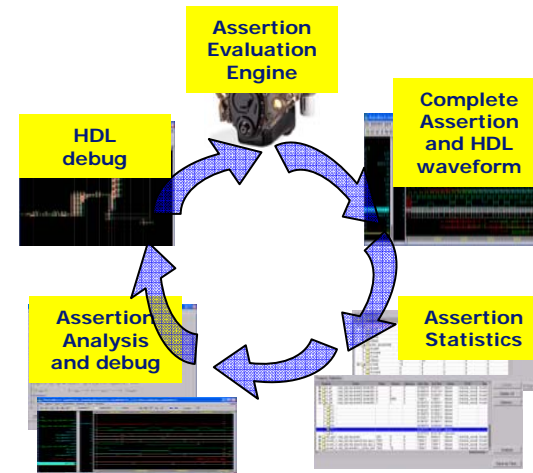
- New verification methodology
- Solve current testbench problems

SystemVerilog Design

- Legacy design

Where is Novas for SystemVerilog?

- SystemVerilog Assertions
 - Assertion acquisition
 - Assertion visualization
 - Assertion debug
 - Assertion statistic
- SystemVerilog Testbench
 - SVTB source code support
 - VMM, AVM library support
 - Pre-simulation analysis and debug
 - Post-simulation analysis and debug
- SystemVerilog Design
 - Same as Verilog design

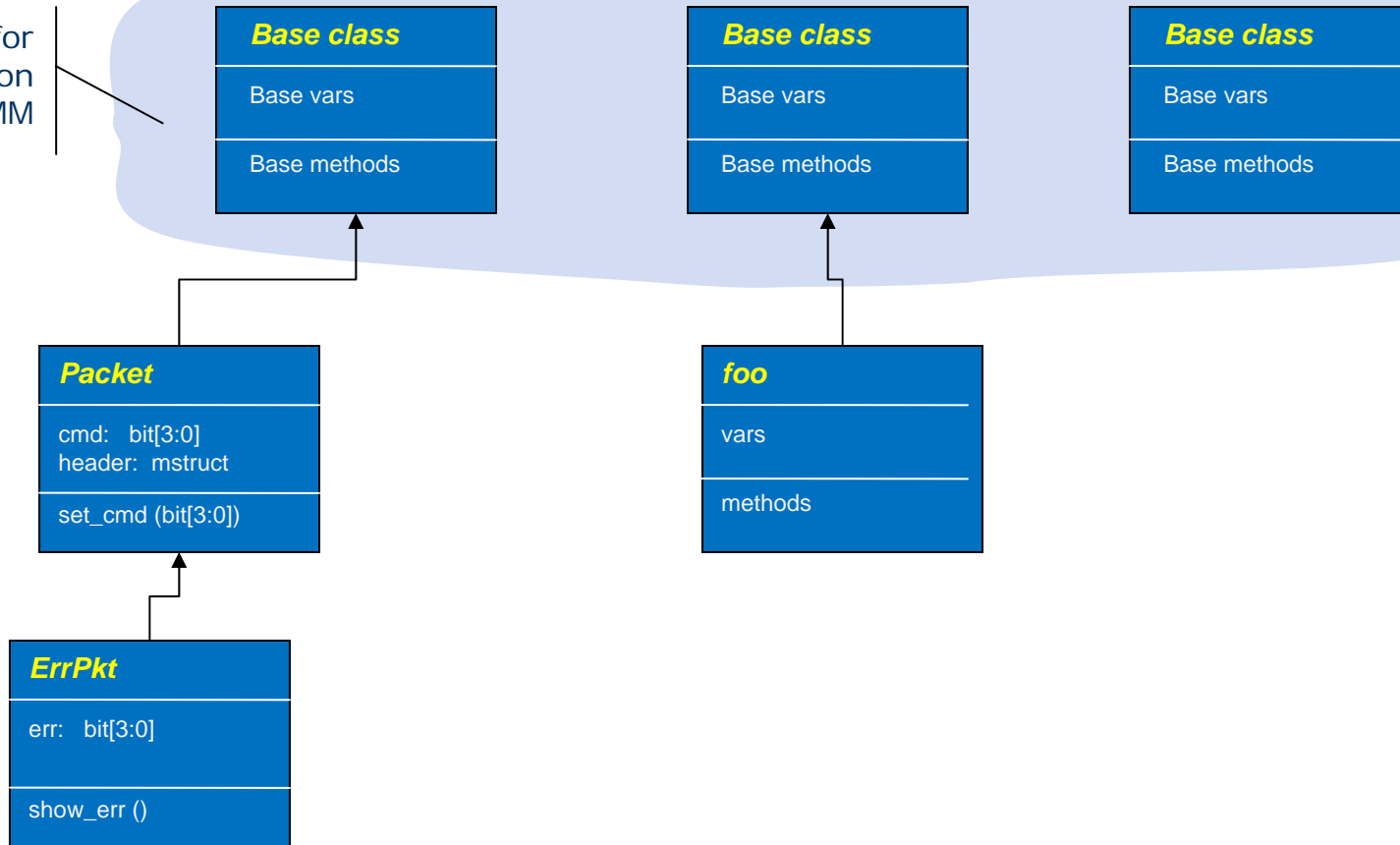


The Difficulties of SystemVerilog Testbench Debug

- SystemVerilog Testbench is not a hardware design language
 - Software-like coding style
 - Declaration base instead of instance base
 - Function based debugging instead of signal based
 - Complex inheritance relations
 - Zero time call stack
 - Testbench specific coding style
 - Coverage
 - Constraint
- When using a library
 - How can I use library efficiently?
 - What does VMM and AVM provide?
 - Any tools that help to understand and use the libraries?

OO = Reuse

Base classes for verification such as VMM



Verdi's New SVTB Comprehension Window (Early Access Version in 2008 Q1)

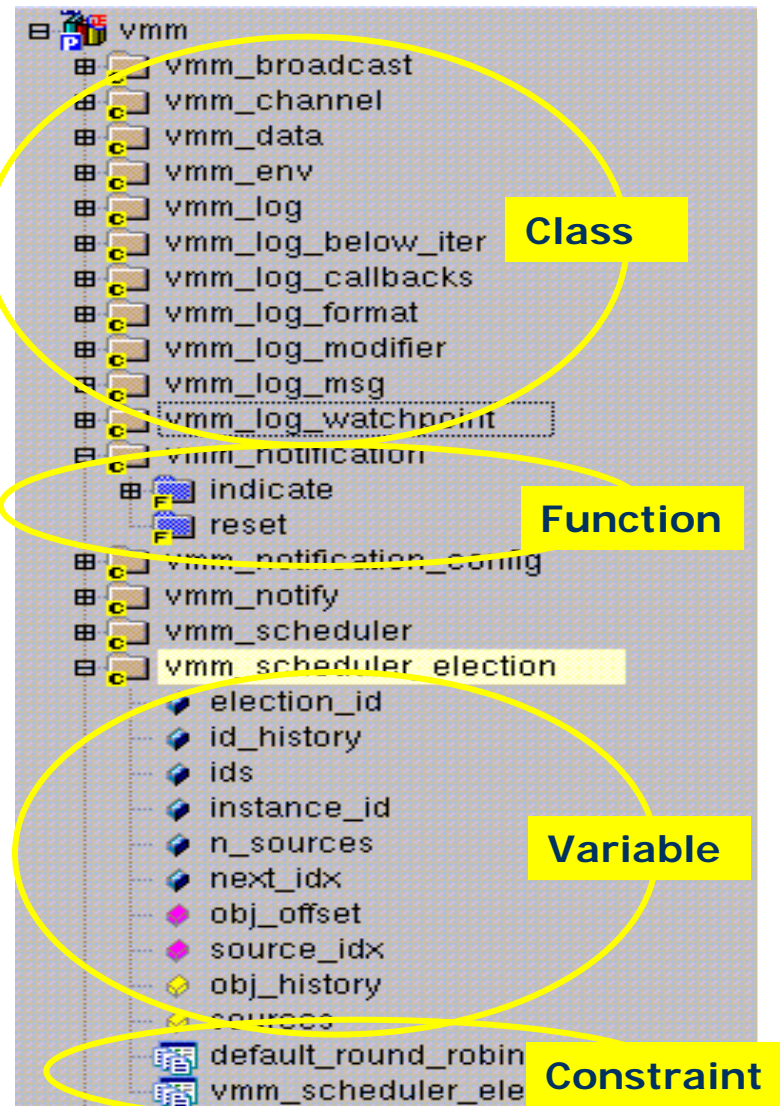
The screenshot displays the Verdi Testbench Browser interface, which is divided into several panes:

- Declaration Based Hierarchy Browser:** Located on the left, it shows a tree view of the testbench hierarchy. The `vmm_xactor_callbacks` directory is highlighted.
- SVTB Analysis Pane:** Located in the center, it shows an inheritance view of the `vmm_xactor_callbacks` class, listing its sub-classes: `DriverCallbacks`, `DriverCovCallbacks`, `DriverSbCallbacks`, `Packet_scenario_gen_callbacks`, `GenCovCallbacks`, `GenSbCallbacks`, `ReceiverCallbacks`, and `ReceiverCovCallbacks`.
- SVTB Source Code Pane:** Located on the right, it displays the source code for the `vmm_xactor_callbacks` class. The code includes a virtual class declaration and a class definition with various attributes and typedefs.
- Message/Log Pane:** Located at the bottom, it is currently empty.

```
1 //  
2 // SYNOPSIS CONFIDENTIAL - SYNOPSIS CONFIDENTIAL - SYNOPSIS COI  
3 //  
4 // This is an unpublished, proprietary work of Synopsys, Inc.,  
5 // fully protected under copyright and trade secret laws. You r  
6 // view, use, disclose, copy, or distribute this file or any ir  
7 // contained herein except pursuant to a valid written license  
8 // Synopsys.  
9 //  
10 // SYNOPSIS CONFIDENTIAL - SYNOPSIS CONFIDENTIAL - SYNOPSIS COI  
11 //  
12 //  
13 //  
14 virtual class vmm_xactor_callbacks;  
15 endclass  
16 //  
17 class vmm_xactor;  
18     vmm_log log;  
19 //  
20     int stream_id;  
21 //  
22     vmm_notify notifi  
23     typedef enum int  
24 //  
25 //  
26         XACTOR_STOPPED,  
27         XACTOR_RESET} notifications_e;  
28 //  
29     typedef enum int {SOFT_RST,
```

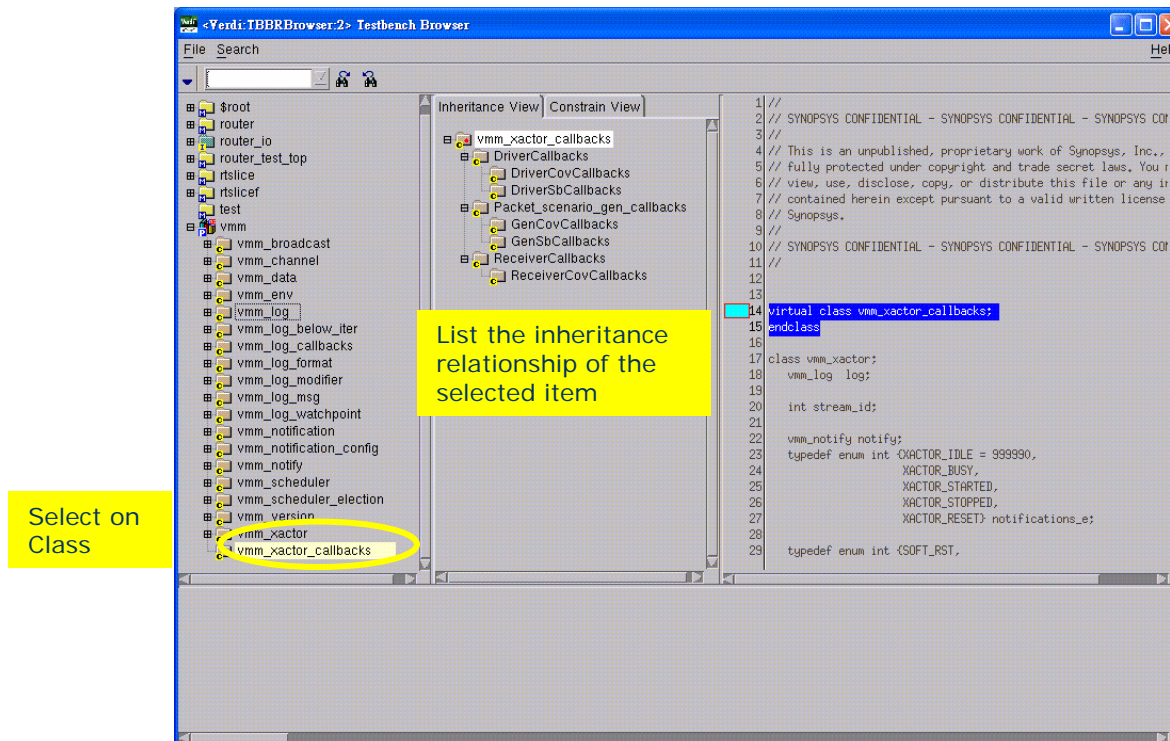
Details of SVTB Comprehension Window – Class View

- Declaration Oriented
 - List all the class declarations in design
 - List all the members under class definition
 - Function, Constraint and Covergroup
 - With variable, function, constraint and cover points
 - Derived data member will be listed in different manner



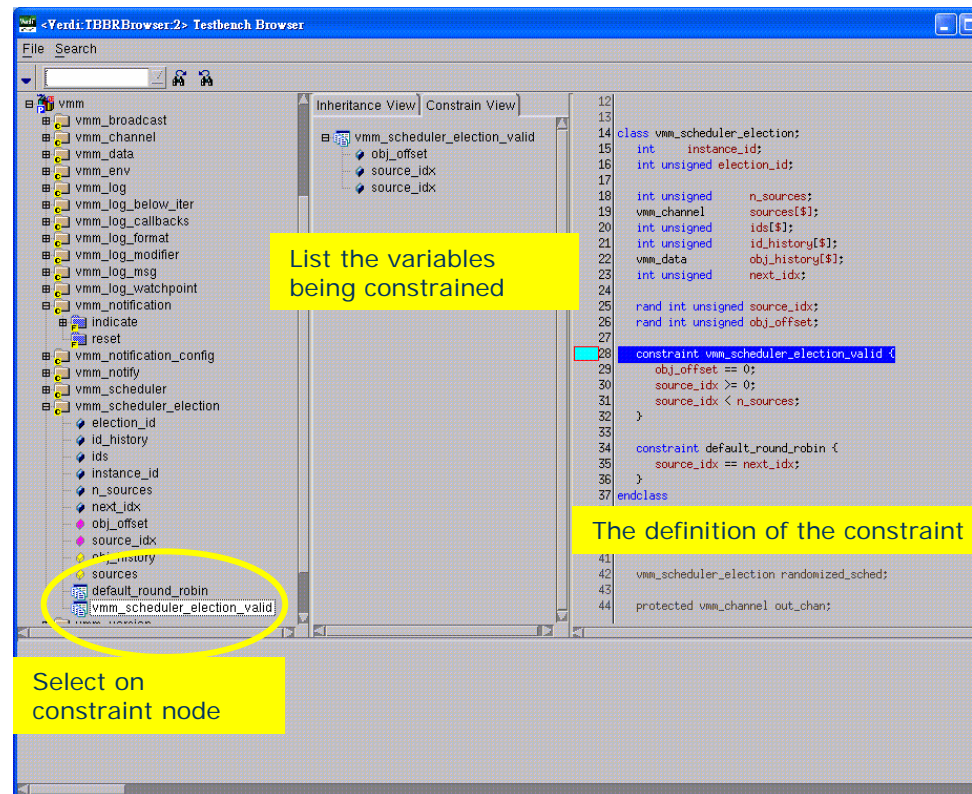
Details of SVTB Comprehension Window – Class View

- Display inheritance relationships
 - When you select classes, functions, constraints and covergroup
 - List inheritance history in tree view
 - Allow user to trace parent/child of selected item



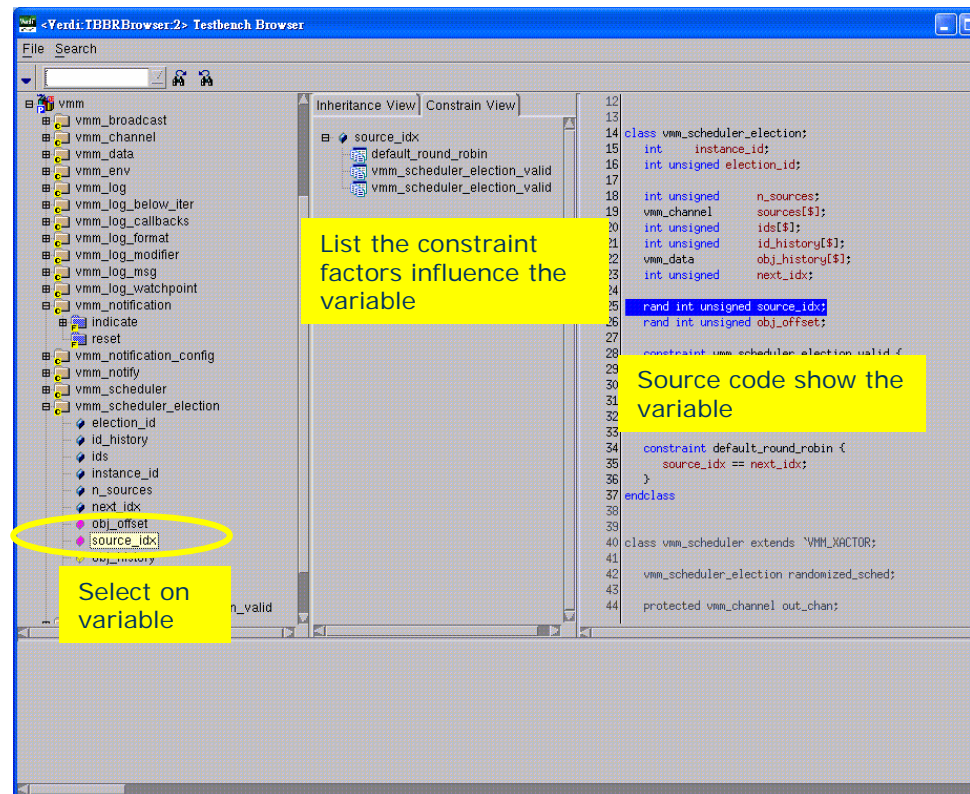
Details of SVTB Comprehension Window – Constraint View

- Constraint list for selected variable
 - When you select variable
 - List the constraint factors that influence the variable
- Constrained variables
 - When you select constraint node
 - List constraint definition under selected node



Details of SVTB Comprehension Window – Constraint View

- Constraint list for selected variable
 - When you select variable
 - List the constraint factors that influence the variable
- Constrained variables
 - When you select constraint node
 - List constraint definition under selected node



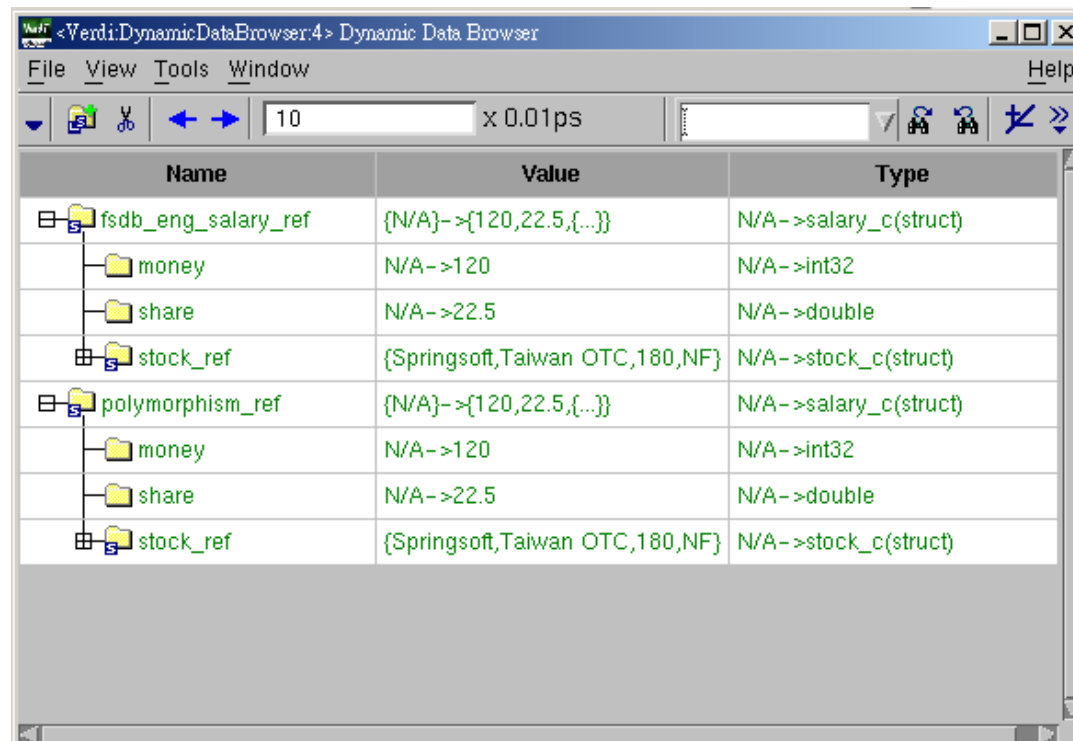
Details of SVTB Comprehension Window – Source Code Pane

- Provide source viewing and tracing capabilities
 - Get definition
 - Show calling
 - Expand macro dynamically

```
Help
Source Code
1 class Environment extends vmm_env;
2 virtual router_io_TB router;
3 Configuration cfg;
4 Packet_scenario_gen gen;
5 Scoreboard sb;
6 GenCovCallbacks gen_cov_cb;
7 GenSbCallbacks gen_sb_cb;
8 Driver drv[];
9 DriverCovCallbacks drv_cov_cb;
10 DriverSbCallbacks drv_sb_cb;
11 Rece:
12 Rece: SVTB Source Code
13 vmm_t Pane
14 vmm_s
15
16 function new(virtual router_io_TB router);
17     super.new("Environment");
18     this.router = router;
19     cfg = new();
20 endfunction
21
22 virtual function void gen_cfg();
23     super.gen_cfg();
24     if (!cfg.randomize())
25         `vmm_fatal(this,log,"Configuration Randomization Failed!\n");
26 endfunction
27
28 virtual function void build();
29     super.build();
30     gen = new("gen", 0);
31     gen.stop_after_n_insts = cfg.run_for_n_packets;
```

Dumping for SVTB

- SVTB dumping (Class dumping) (**2007.07 Beta for VCS**)
 - Use ordinary task (\$fsdbDumpvars) to dump variables in class
 - Use Verdi's Dynamic Data Browser to see the results

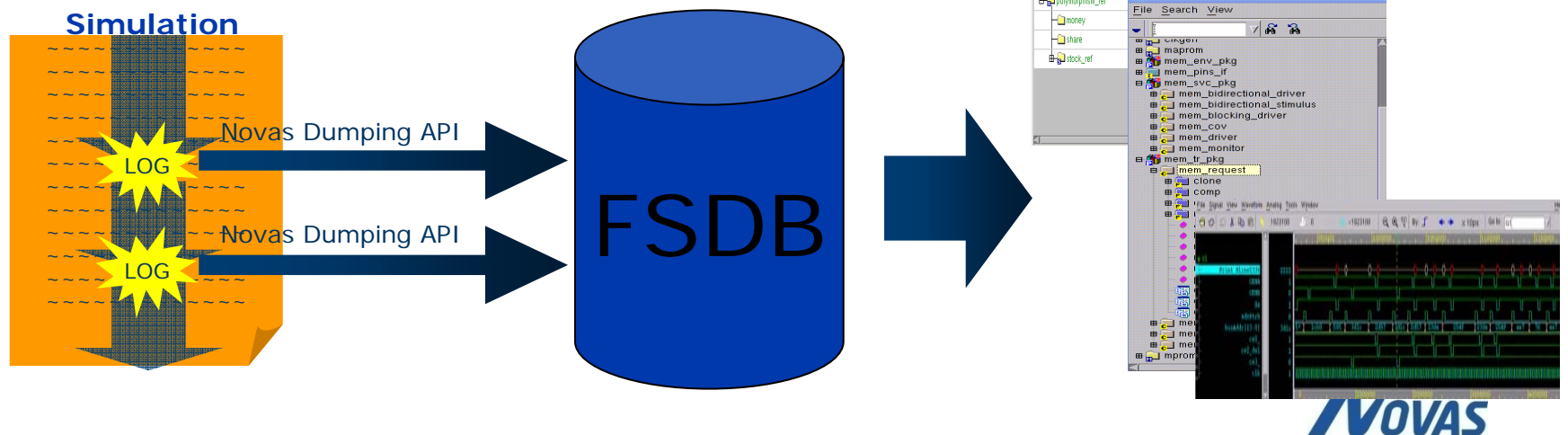


The screenshot shows the Verdi Dynamic Data Browser window. The title bar reads "<Verdi> <Verdi>DynamicDataBrowser:4> Dynamic Data Browser". The menu bar includes File, View, Tools, Window, and Help. The toolbar contains icons for file operations and navigation, along with a time scale set to 10 x 0.01ps. The main area displays a table with three columns: Name, Value, and Type. The table contains several rows representing different variables and their sub-structures.

Name	Value	Type
fsdb_eng_salary_ref	{N/A}->{120,22.5,{...}}	N/A->salary_c(struct)
money	N/A->120	N/A->int32
share	N/A->22.5	N/A->double
stock_ref	{Springsoft,Taiwan OTC,180,NF}	N/A->stock_c(struct)
polymorphism_ref	{N/A}->{120,22.5,{...}}	N/A->salary_c(struct)
money	N/A->120	N/A->int32
share	N/A->22.5	N/A->double
stock_ref	{Springsoft,Taiwan OTC,180,NF}	N/A->stock_c(struct)

Dumping for SVTB

- Dumping on logging (**TBD**)
 - Logging is widely used in SVTB
 - VMM provides vmm_log class
 - User have customized logging mechanism
- When fatal/error log happens
 - Bug happens
 - The initial point for debugging

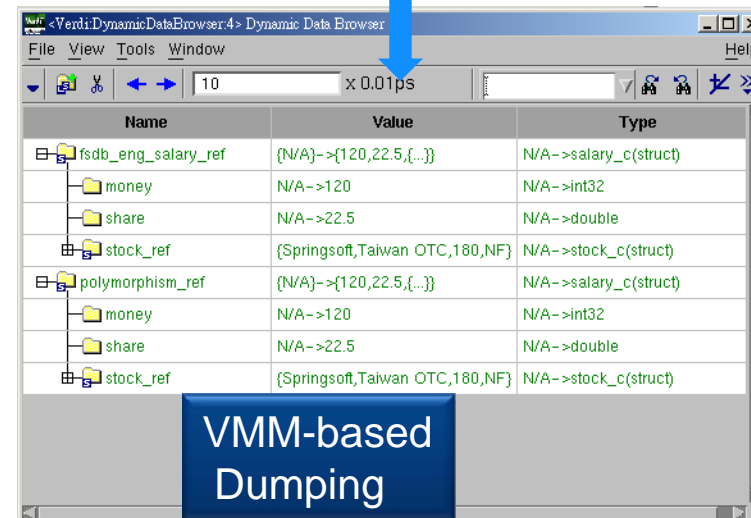
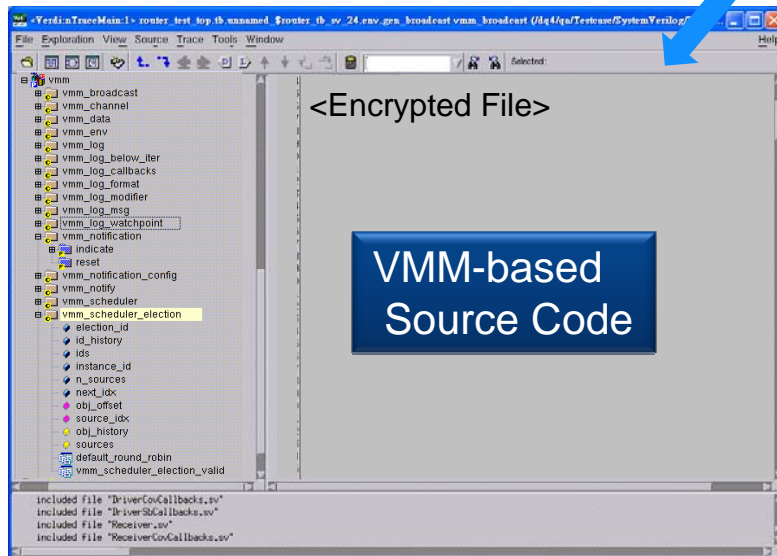
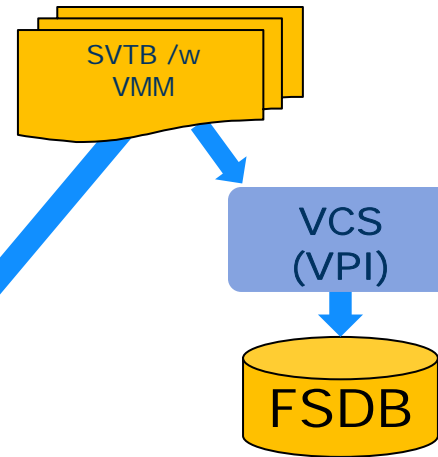


VMM Library Support

- FACT: VMM is SVTB
- VMM library is encrypted
 - No source code to compile
 - Complex scenarios and tests based on VMM
 - Complex transaction-level tracing
 - Many methods and functions to choose from, in the VMM library
- Through a partnership with Synopsys, Verdi now supports VMM (**Today**)
 - Designs based on VMM (even encrypted) can be imported
 - Hierarchy tree that reflects the VMM components
 - Designs with VMM can be traced
 - VMM blocks will be treated as black box (Macro)
 - Internal details not shown
- Help user to use VMM more efficiently (**Future**)
 - List all available functions from VMM library
 - Help user to understand what function is used
 - Provide more info on a function – its members and what it does for example

Debugging Testbench with VMM

- Allow user to import design with encrypted VMM
 - Pre-compiled VMM library included in Verdi package
 - Build the complete hierarchy tree for design comprehension
 - When VMM component selected, show empty source code
 - Easier to get definition from VMM library
- Dumping
 - VPI used to dump SVTB data including VMM



Summary

- Support import for popular libraries
 - VMM and AVM support available today
- Brand new Testbench Browser to understand testbench
 - Software like declaration based hierarchy tree
 - Inheritance relationships for class and function
 - Easily to understand variable vs constraints (constraint vs. variables)
- Static SVTB analysis
 - Simulator independent static SVTB comprehension and analysis
 - Popular SVTB libraries support
- Simulation vs. SVTB debugging
 - Various dumping methods allow user to get desired simulation results
 - Coverage visualization and post simulation analysis
- Comprehension Support
 - Help user to understand/use SVTB libraries efficiently
 - Encrypted VMM library will still see the header of function/class