



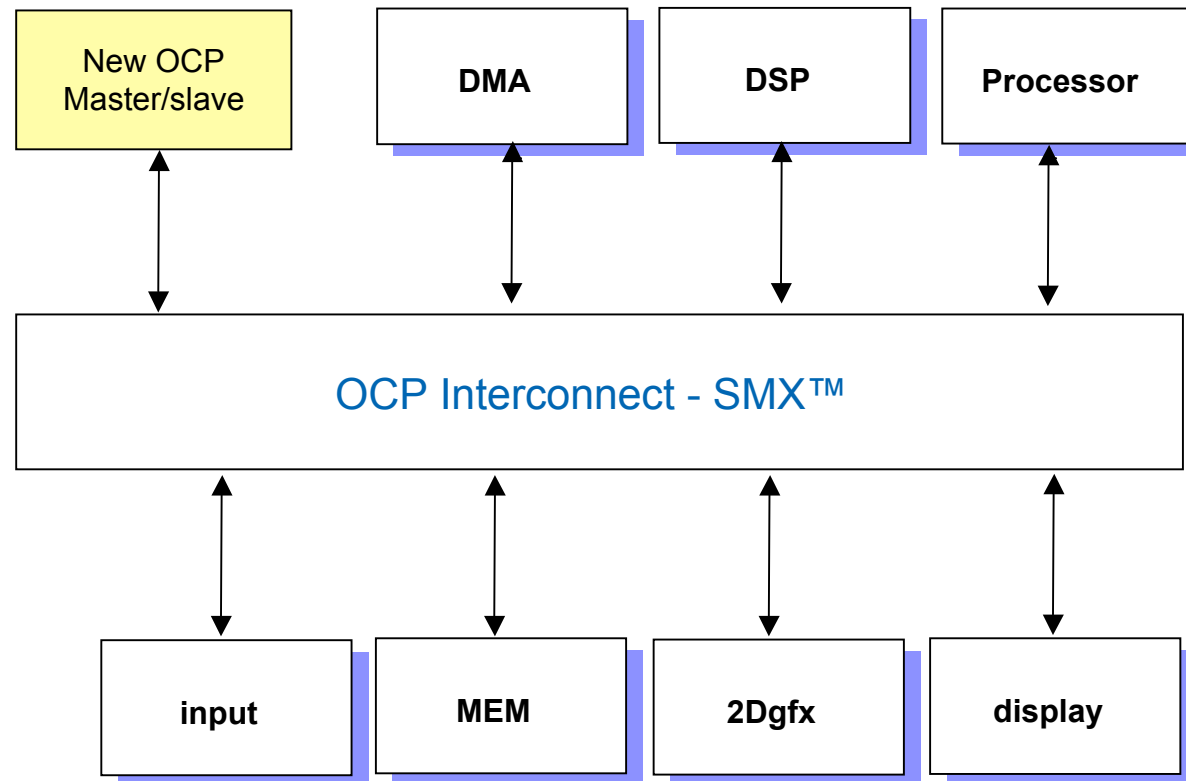
Validation and Analysis of OCP based Virtual Platforms

March 2008

Agenda

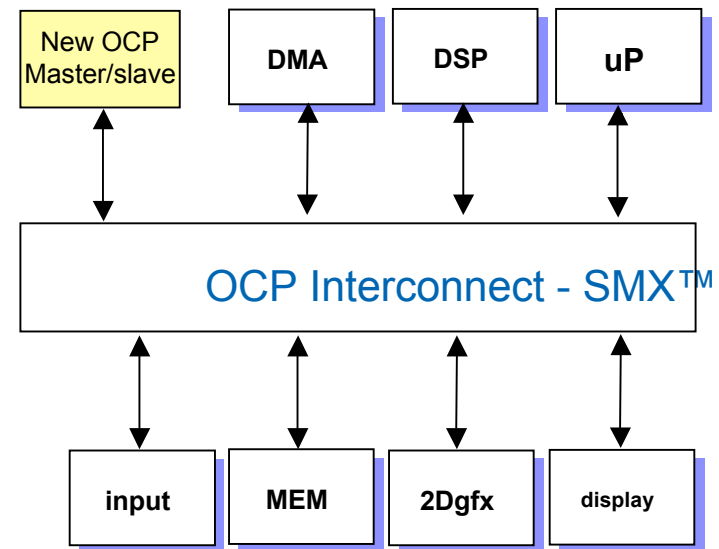
- OCP design platform
- OCP platform verification challenges
- How JEDA's OCP Products address the challenges encountered
 - ▶ OCPchecker - TL1 protocol checking
 - ▶ OCP TL2 Performance checker
 - ▶ OCPcov - TL1 Transaction coverage

A typical OCP design platform



OCP Verification Challenges

- Debug protocol violations at IP interfaces, pinpoint problem areas
- Analysis automation
- Performance checking and measurement



JEDA Products

NSC – ESL verification suite

Any-Architecture Applications

OCP Design Applications

Assertions

NSCa: Assertions across TLM levels

OCPchecker:

OCP Compliance Checks

Coverage

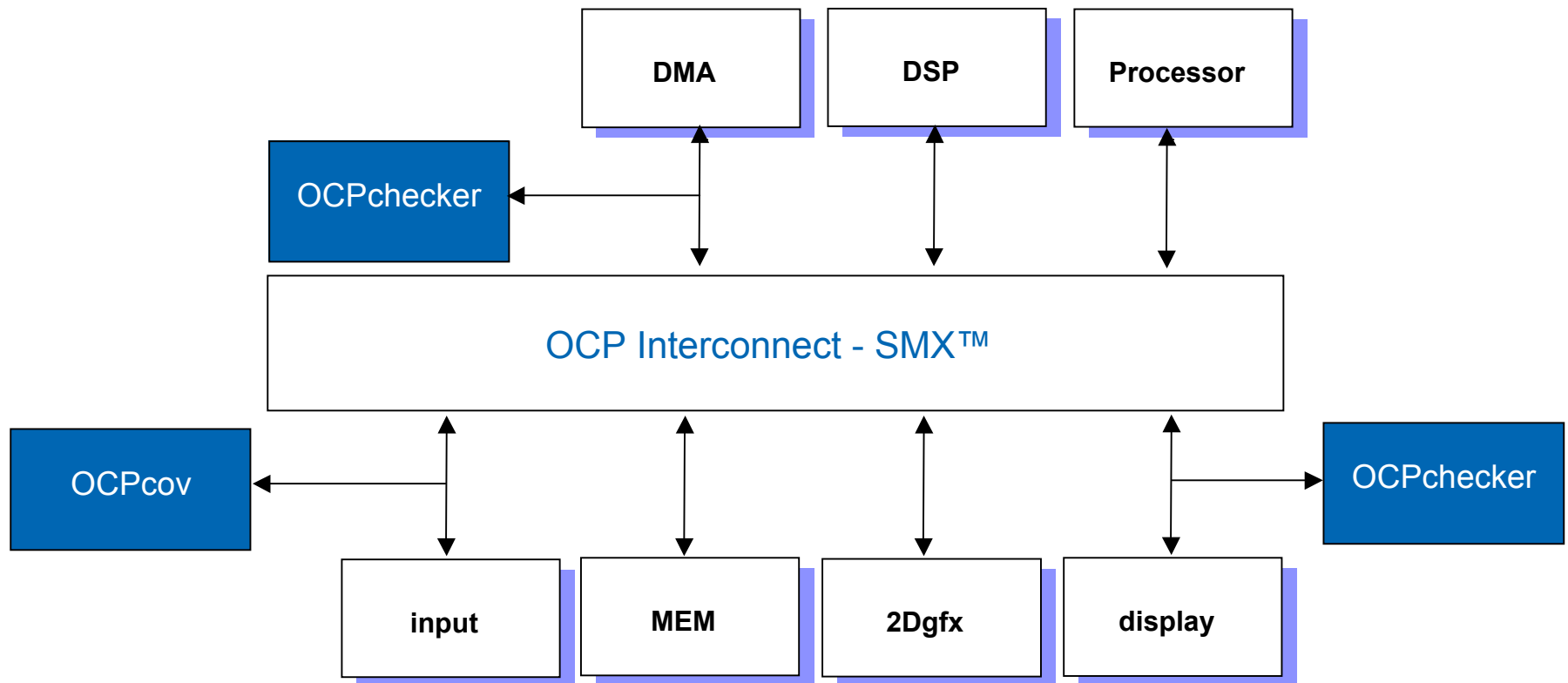
NSCv: TLM Verification

- NSCvDC:** Data/Functional Coverage
- NSCvCC:** Code/line coverage

OCPcov:

OCP Transaction and Data coverage

Applications under OCP Virtual Platform





JEDA OCP Verification Library Overview

- Complete Protocol Checker (OCPchecker)
 - ▶ Based on the OCP-IP Functional Verification Working Group (FVWG) compliance check specification
 - ▶ Support OCP2.0, OCP2.1 and OCP2.2
- OCP protocol coverage (OCPcov)
 - ▶ Use state, toggle, cross method
 - ▶ Implements signal, transaction coverage
- Higher-level (TL2) performance check



OCPchecker

OCP TL1 Protocol Compliance
checking

OCPchecker Update

- Three Main Features
 - ▶ Runtime 2.2/2.1/2.0 protocol violation checks
 - ▶ **OCP parameter checking (new)**
 - ▶ **Incomplete transaction checking (new)**
- Flexible runtime execution control
 - ▶ Assertion, dump on/off control
- Easy steps to use
 - ▶ Create OCP monitor object
 - ▶ Attach OCP monitor to OCP channel
 - ▶ Create OCP checker object
 - ▶ Set property on/off for the checker
 - ▶ Bind, OCP checker to OCP monitor
 - ▶ Set debug dump options

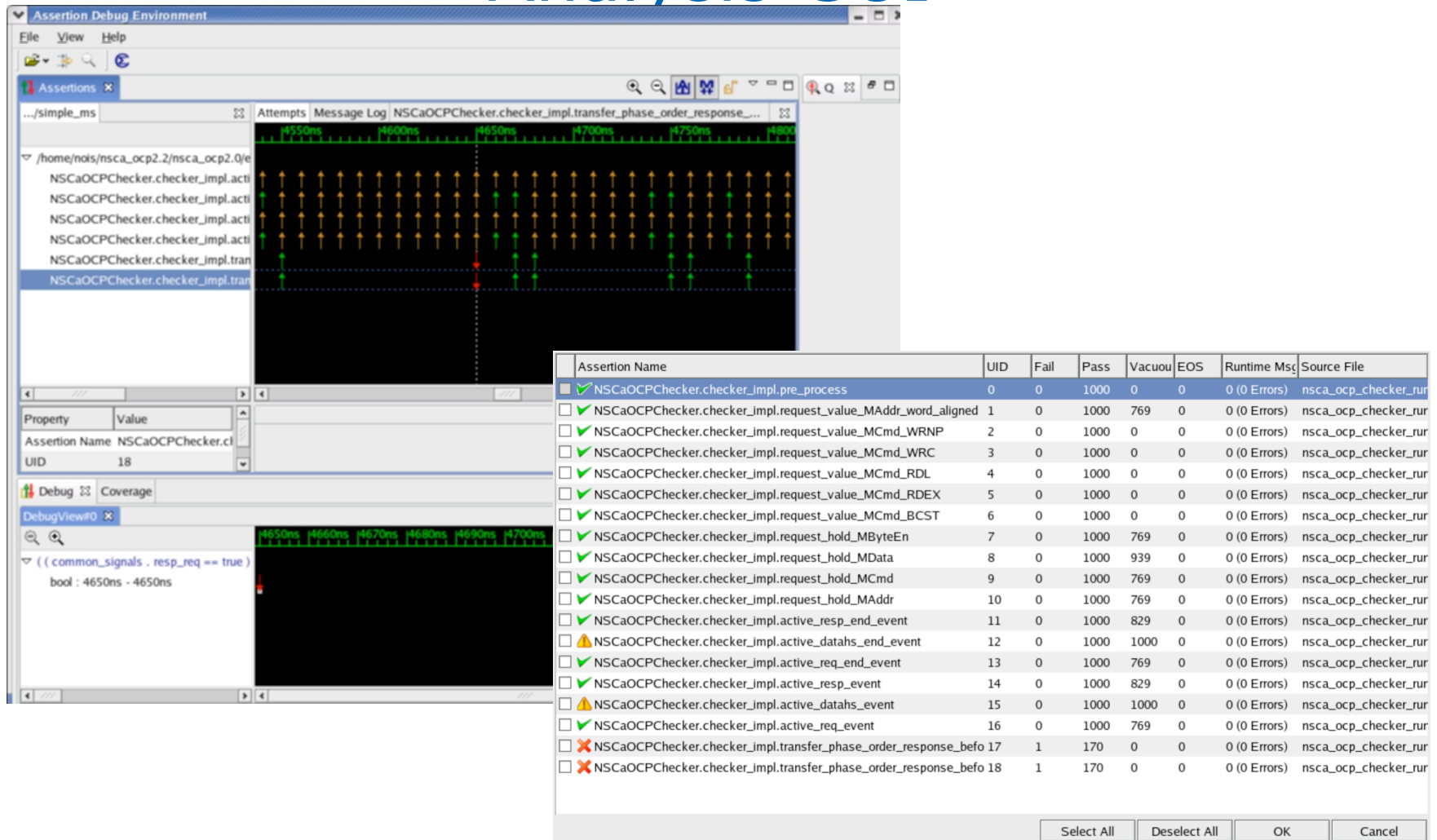
OCPchecker

- Runtime protocol violation checks(150+)
 - ▶ Pinpoints errors – reduce debug time
- Flexible runtime execution control
 - ▶ Assertion, dump on/off control
- Pre-verified and ready to use
- Easy steps to use
 - ▶ Create OCP monitor object
 - ▶ Attach OCP monitor to OCP channel
 - ▶ Create OCP checker object
 - ▶ Set property on/off for the checker
 - ▶ Bind, OCP checker to OCP monitor
 - ▶ Set debug dump options

TL2 Performance Check

- At transaction level
- Implemented with NSCa-TLA
- Outstanding request measurement
 - ▶ Max of outstanding request for a bus
 - ▶ Measure coverage info on the bus
- Access performance
 - ▶ Min/Max response time
 - ▶ Arrival rate, departure rate
 - ▶ Limit on out of order

Eclipse Based OCP Protocol Debug & Analysis GUI



Assertion Name	UID	Fail	Pass	Vacuou	EOS	Runtime Msc	Source File
<input checked="" type="checkbox"/> NSCaOCPChecker.checker_impl.pre_process	0	0	1000	0	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.request_value_MAddr_word_aligned	1	0	1000	769	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.request_value_MCmd_WRNP	2	0	1000	0	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.request_value_MCmd_WRC	3	0	1000	0	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.request_value_MCmd_RDL	4	0	1000	0	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.request_value_MCmd_RDEX	5	0	1000	0	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.request_value_MCmd_BCST	6	0	1000	0	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.request_hold_MByteEn	7	0	1000	769	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.request_hold_MData	8	0	1000	939	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.request_hold_MCmd	9	0	1000	769	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.request_hold_MAddr	10	0	1000	769	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.active_resp_end_event	11	0	1000	829	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.active_dataahs_end_event	12	0	1000	1000	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.active_req_end_event	13	0	1000	769	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.active_resp_event	14	0	1000	829	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.active_dataahs_event	15	0	1000	1000	0	0 (0 Errors)	nsc_otp_checker_rur
<input type="checkbox"/> NSCaOCPChecker.checker_impl.active_req_event	16	0	1000	769	0	0 (0 Errors)	nsc_otp_checker_rur
<input checked="" type="checkbox"/> NSCaOCPChecker.checker_impl.transfer_phase_order_response_befo 17	17	1	170	0	0	0 (0 Errors)	nsc_otp_checker_rur
<input checked="" type="checkbox"/> NSCaOCPChecker.checker_impl.transfer_phase_order_response_befo 18	18	1	170	0	0	0 (0 Errors)	nsc_otp_checker_rur



OCPcov

Transaction and Data
Coverage

OCPcov Use Models

- During Platform & Model development
 - ▶ Measure progress of simulation
 - ▶ Identify vectors that provide broad coverage maximal coverage
 - ▶ To obtain confidence about model being sufficiently exercised
- Testbench environment
 - ▶ Validate that new tests added are not breaking testbench environment

Coverage Highlight

- Toggle Coverage
 - ▶ We support for each coverage turn on/off enable function and for multiple bit signals, we support mask function for them.
- State Coverage
 - ▶ We support each coverage turn on/off enable function, each signals, we support state range configuration function according user's design spec.
- Cross coverage
 - ▶ We support each coverage turn on/off enable function, all cross combination between categories is forbidden in spec has been capture out from this application.
- All configuration information controlled from a single configuration file
 - ▶ Enable/disable
 - ▶ toggle mask
 - ▶ state range

OCPcov1.0 features(Released)

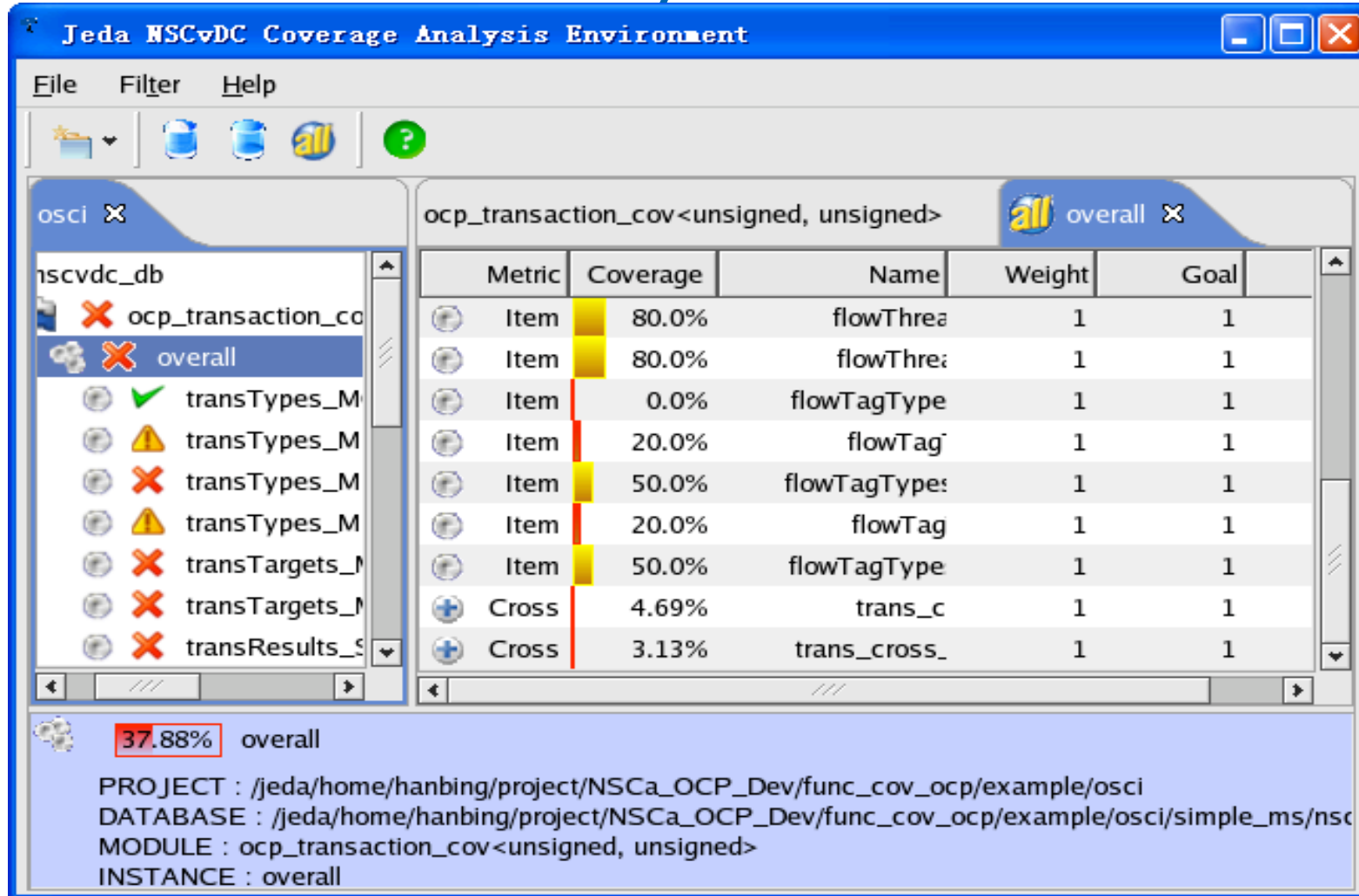
- Support all toggle coverages specified in OCP specification that sample on every cycle;
- Support all state coverages specified in OCP specification that sample on every phase when the sampled OCP signal is meaningful;
- Support all the cross coverages specified in OCP specification that sample on every transaction when the OCP signals referenced in the cross are all meaningful;
- Support all above features for both OCP2.1 and OCP2.2.



OCP Transaction Coverage

- Examples (Cross Coverage)
 - ▶ Transaction types vs address ranges
 - ▶ Transaction types vs threads

Eclipse OCP Protocol Coverage Analysis GUI



The screenshot displays the Eclipse OCP Protocol Coverage Analysis GUI. The main window is titled "Jeda NSCvDC Coverage Analysis Environment". The interface includes a menu bar (File, Filter, Help), a toolbar with icons for file operations and a help icon, and a main workspace divided into several panes.

The left pane shows a tree view of the database structure under "nscvdc_db", including "ocp_transaction_co" (marked with a red X), "overall" (marked with a red X), and several "transTypes_M" and "transTargets_M" entries (marked with green checkmarks or yellow triangles).

The right pane displays a table of metrics for the "ocp_transaction_cov<unsigned, unsigned>" module. The table has columns for Metric, Coverage, Name, Weight, and Goal. The overall coverage is 37.88%.

Metric	Coverage	Name	Weight	Goal
Item	80.0%	flowThrea	1	1
Item	80.0%	flowThre:	1	1
Item	0.0%	flowTagType	1	1
Item	20.0%	flowTag	1	1
Item	50.0%	flowTagType:	1	1
Item	20.0%	flowTag	1	1
Item	50.0%	flowTagType	1	1
Cross	4.69%	trans_c	1	1
Cross	3.13%	trans_cross_	1	1

At the bottom of the window, the overall coverage is displayed as **37.88%** overall. Below this, the project and database paths are shown:

```

PROJECT : /jeda/home/hanbing/project/NSCa_OCP_Dev/func_cov_ocp/example/osci
DATABASE : /jeda/home/hanbing/project/NSCa_OCP_Dev/func_cov_ocp/example/osci/simple_ms/nsc
MODULE : ocp_transaction_cov<unsigned, unsigned>
INSTANCE : overall
  
```

Summary

- Complete verification solution
 - ▶ Protocol Checking
 - ▶ Performance Checking
 - ▶ Coverage Measurement
- Save time to verify model correctness
- Save time to debug model failures
- Better system level simulation coverage



Thank You.

www.jedatechnologies.net