Almost Ten Years of Process Algebras and Model Checking for Multiprocessor Architectures

Hubert Garavel

joint work with many other persons at BULL and INRIA

INRIA Rhône-Alpes / VASY
655, avenue de l'Europe
F-38330 Montbonnot Saint-Martin
http://www.inrialpes.fr/vasy

Formal methods for hardware design

- Characteristics of hardware circuits:
  - increasingly complex
  - shortened design cycles
  - errors are expensive
    - fundry costs
    - no patches
- Hardware designers are:
  - open to formal methods and verification
  - but used to synchronous, deterministic systems

In this talk

- Almost 10 years of BULL-INRIA collaboration
- On the Bull side
  - Asynchronous issues in multiprocessor architectures
  - Bus arbitration protocols
  - Cache coherency protocols
- On the INRIA/VASY side
  - Process algebra and model checking
  - LOTOS specification language (ISO 8807)
  - CADP toolbox (*Construction and Analysis of Distributed Processes*)

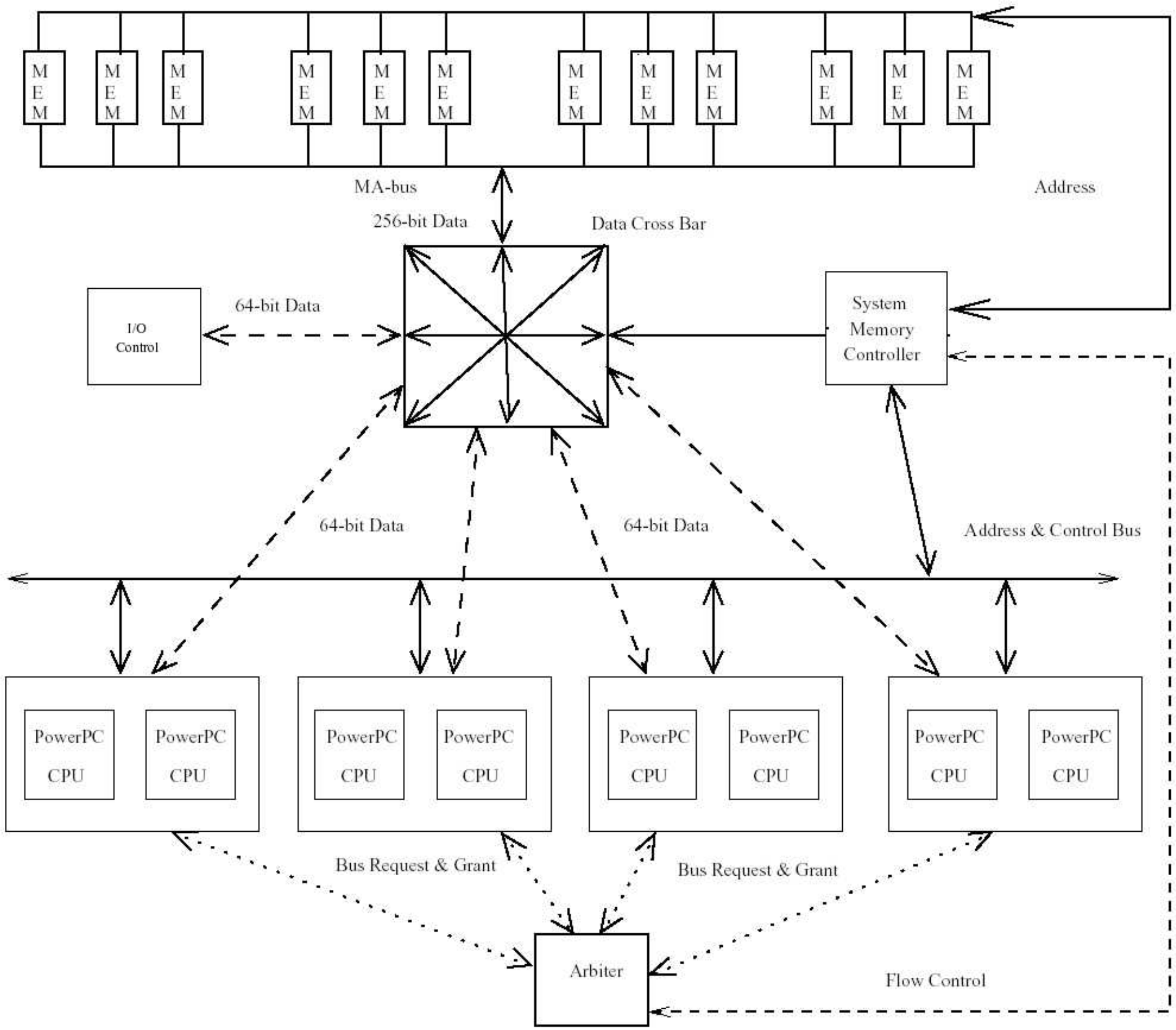
The CADP toolbox
http://www.inrialpes.fr/vasy/cadp

- Main features:
  - LOTOS -> C compilers
  - equivalence checking (bisimulations)
  - model checking (modal mu-calculus)
  - visual checking (graph drawing)
  - exhaustive, partial, on the fly, compositional verification
  - step by step simulation, random execution
  - C code generation, rapid prototyping
  - test case generation
- Wide dissemination:
  - license agreement signed with 310 organizations
  - installed on 840 machines in 2003
  - 72 case studies done with CADP
  - 16 research tools connected to CADP
  - 17 academic courses using CADP

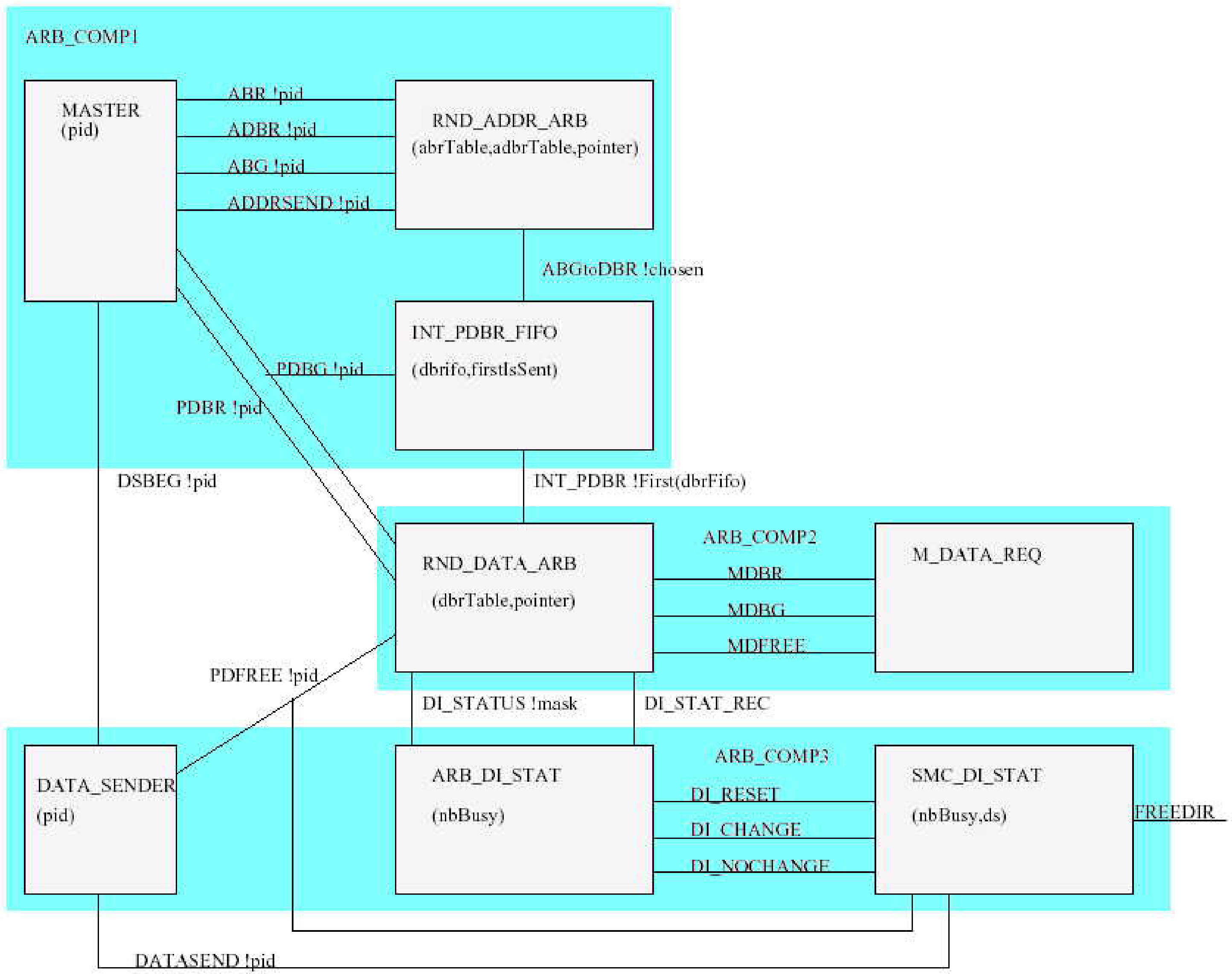
Phase 1 (1995–1996)

Target: Powerscale

- Multiprocessor architecture
- based on PowerPC microprocessors
- used in Bull's Escala servers and workstations
- With a hidden bug



Formal specification



- 720 lines of LOTOS
- 7 concurrent processes:
  - processors
  - memory controller
  - bus arbiter

Verification results

- Four correctness properties identified:
  - *Proper response to bus requests*
  - *Fairness of the arbitration*
  - *Order of grants for address-data requests*
  - *Correctness of the DBG flow control*

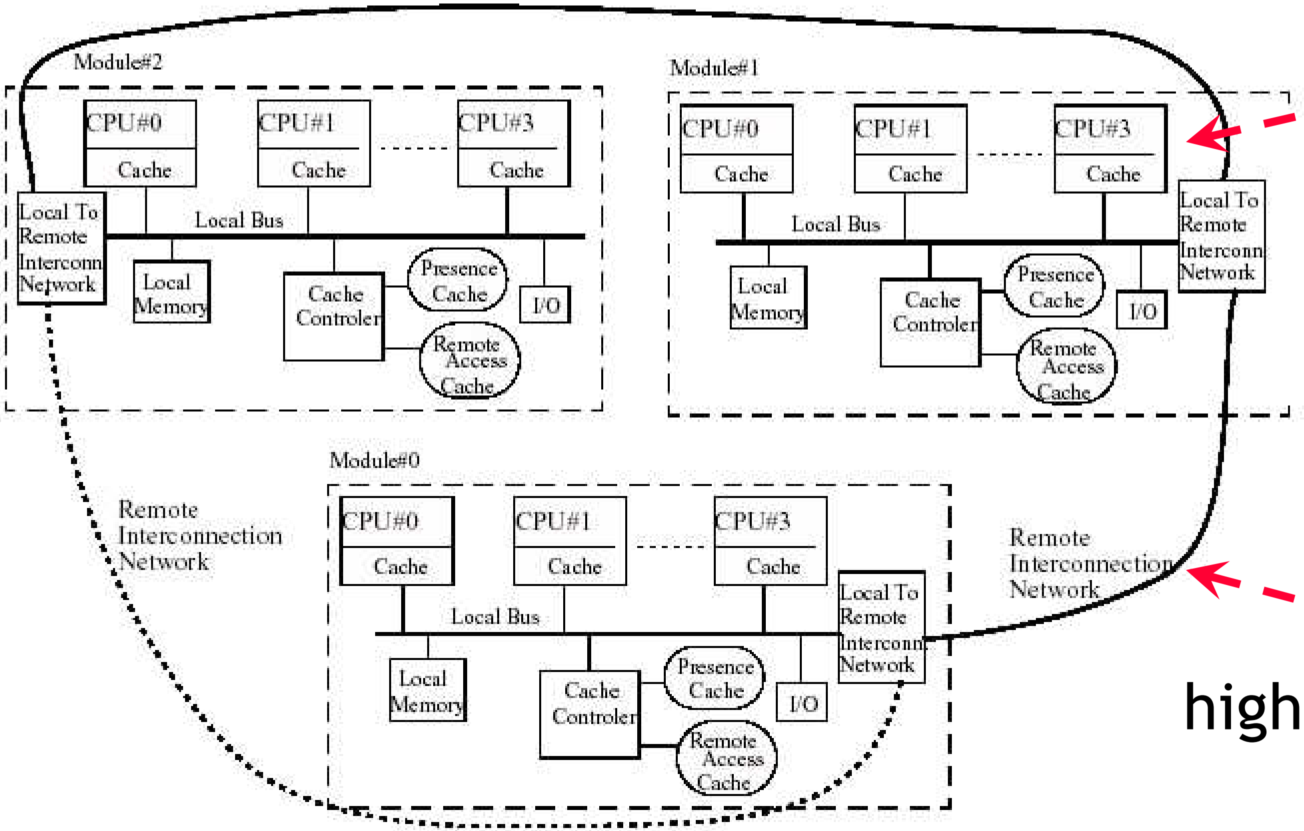
- State enumeration would fail (potentially 10^12 states)
- Compositional verification (bisimulations) was used
- Using CADP, the bug could be found in a few minutes

FORTE'96 paper [Chehaibar-Garavel-Mounier-Tawbi-Zulian-96]

Phase 2 (1996—Sep. 1998)

Target: Polykid

- A multiprocessor architecture under design at Bull Italy
- based on PowerPC processors
- CC-NUMA memory model



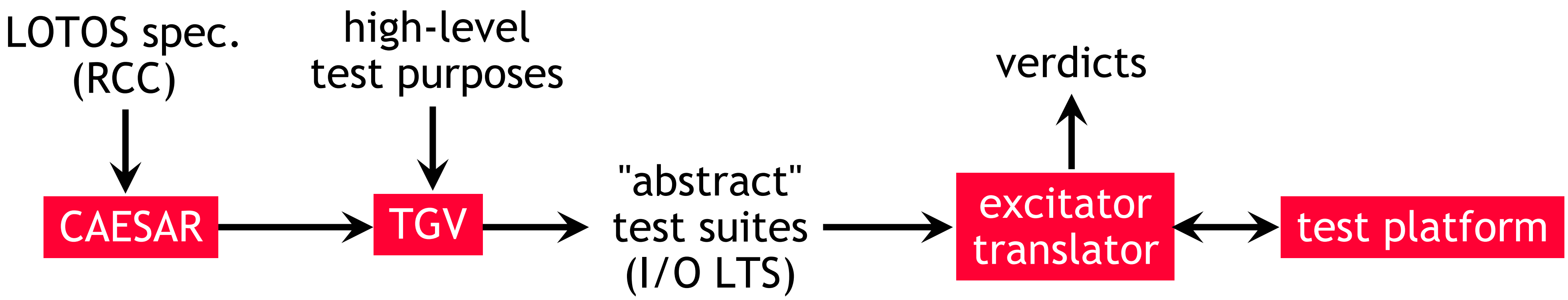
lower level: SMP snoopy based cache coherence

higher level: loose coupling directory based cache coherence

Specification and verification

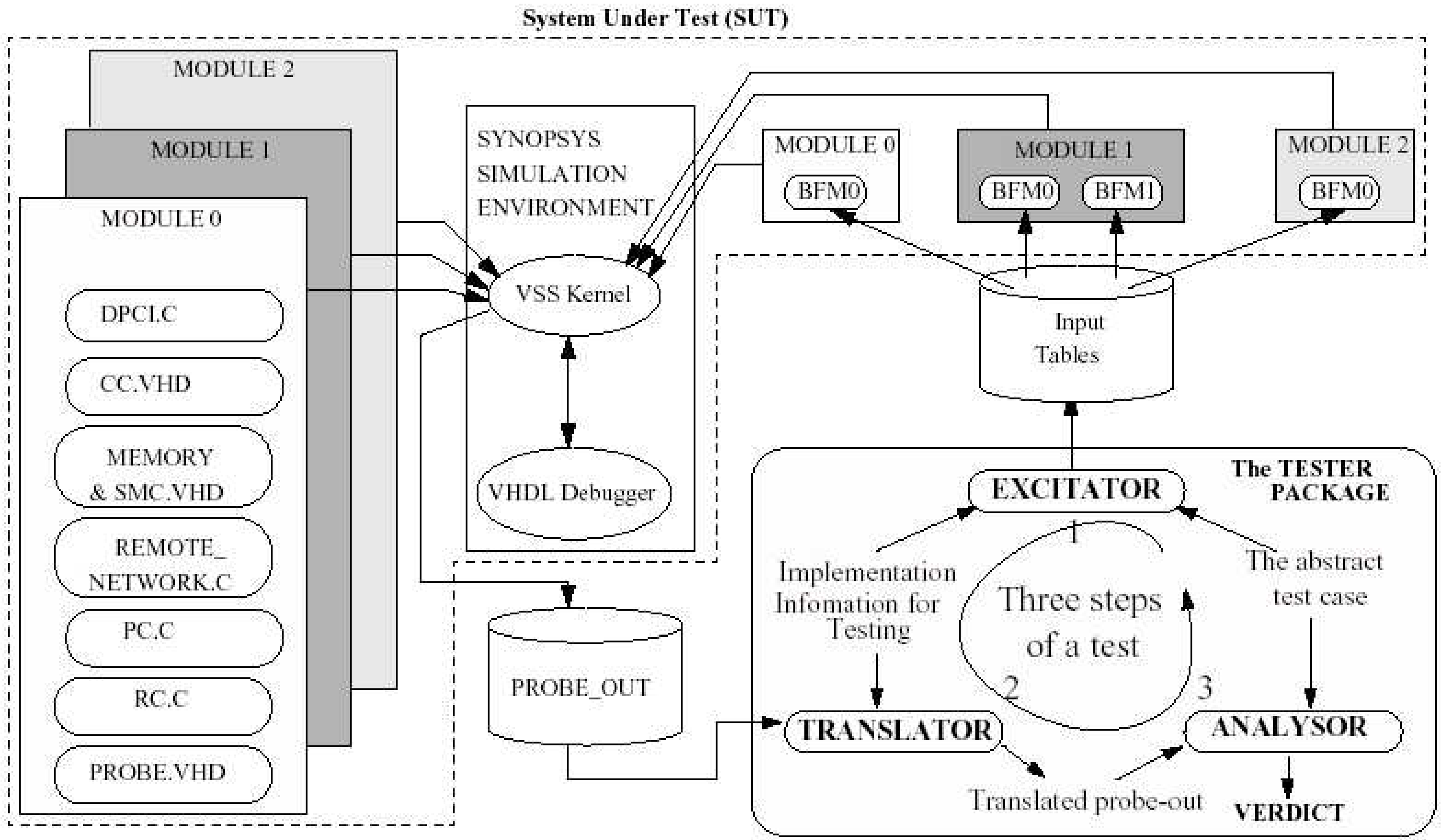
- Several specifications developed
  - Polykid architecture: 4,000 lines of LOTOS
  - Cache coherency rules: 2,000 lines of LOTOS
- Validation by simulation and model checking on abstracted subsets (2,000 lines of LOTOS, 10 concurrent processes)
- Several problems (deadlocks, memory consistency violation, undocumented behaviours) found:
  - phase 1: 55 questions
  - phase 2: 20 questions, 7 serious issues
  - phase 3: 13 serious issues

Test generation using TGV

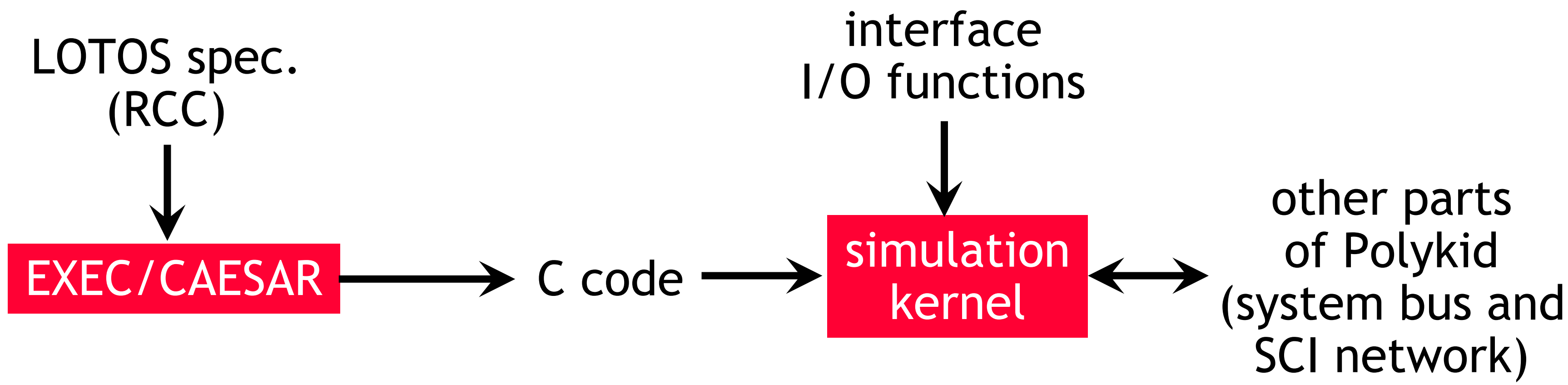


- 75 tests generated (more than 400 states each) in 1 man.month
- Tools developed for automated test execution
- Test execution requires less than 20 hours
- 5 new bugs discovered in VHDL design
- IWTCS'98 paper: [Kahlouche-Viho-Zendri-98]

The actual Polykid testbench



Hardware emulation using EXEC/CAESAR



- Replacement of a missing ASIC by a software emulation running on a PowerPC microprocessor
- Target: RCC (*Remote Cache Controller*)
- 3,400 lines of LOTOS, 7,000 lines of C
- Exec/Caesar: *The* correct scheme to interface process algebra specs with a real environment

The end of Polykid: A sad story...

- Polykid was too late on market
- Eventually, Bull cancelled the Polykid project
- Bull's Italian plant was closed

But:

- Formal methods had proven to be valuable
- BULL-INRIA collaboration would continue with a new architecture

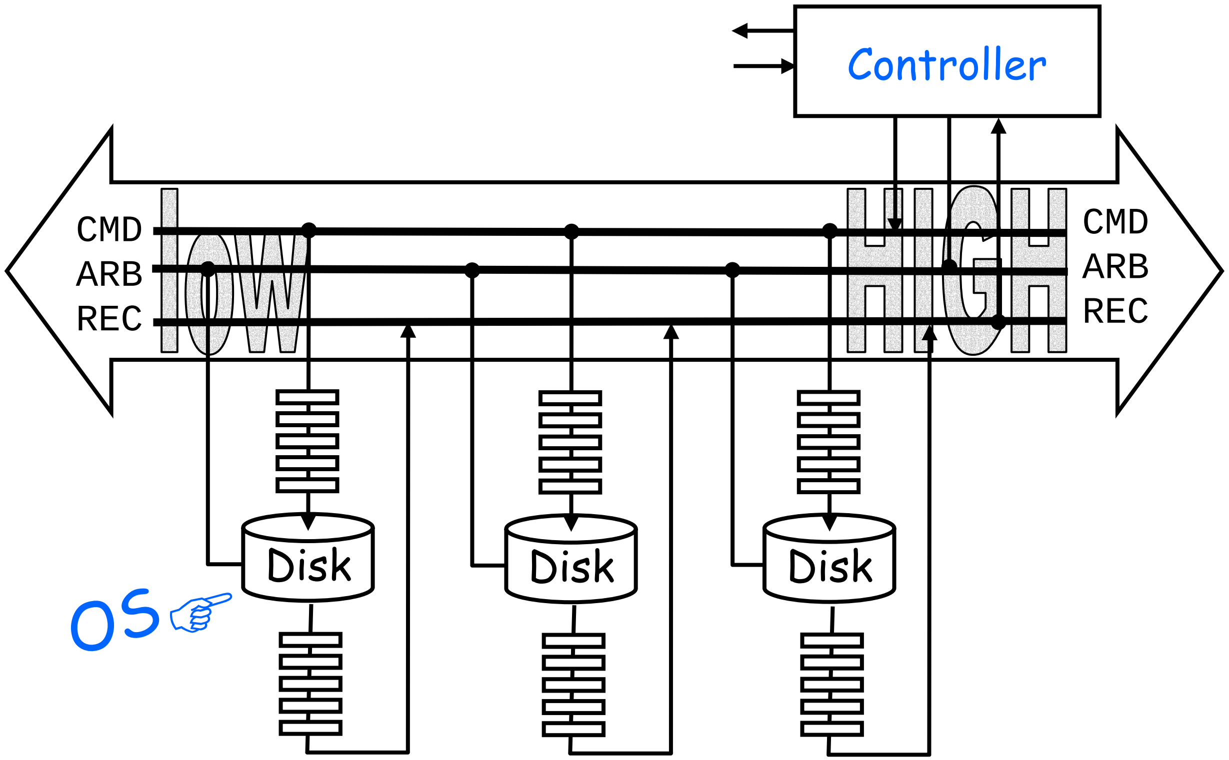
STTT paper in 2001: [Garavel-Viho-Zendri-01]

Intermezzo (1998—2002)

joint work with Holger Hermanns

Target: SCSI-2 bus arbitration protocol

- SCSI-2: a former IEEE computer bus standard
- Bus grant based on fixed priorities (SCSI numbers)



- Unexpected OS deadlocks reported by Bull

Specification and verification

- SCSI-2 bus arbiter: only 220 lines of LOTOS
  The n-party rendezvous of LOTOS with its value negociation features is unavoidable for concise modeling (a challenge for other formalisms!)

- Compositional state space generation and model checking using CADP
  The starvation problem was confirmed
  (This problem was fixed in SCSI-3 standard)

Performance evaluation

- Application of H. Hermanns' PhD thesis:
  Performance models can be obtained by limited changes in a LOTOS specification

- Compositional generation of Markov chains using CADP
  Steady state analysis suggests strategies to avoid starvation and increase throughput

- FME'02 paper [Garavel-Hermanns-02]

Phase 3 (Sep. 1998—now)

The FormalFame Team

Bull

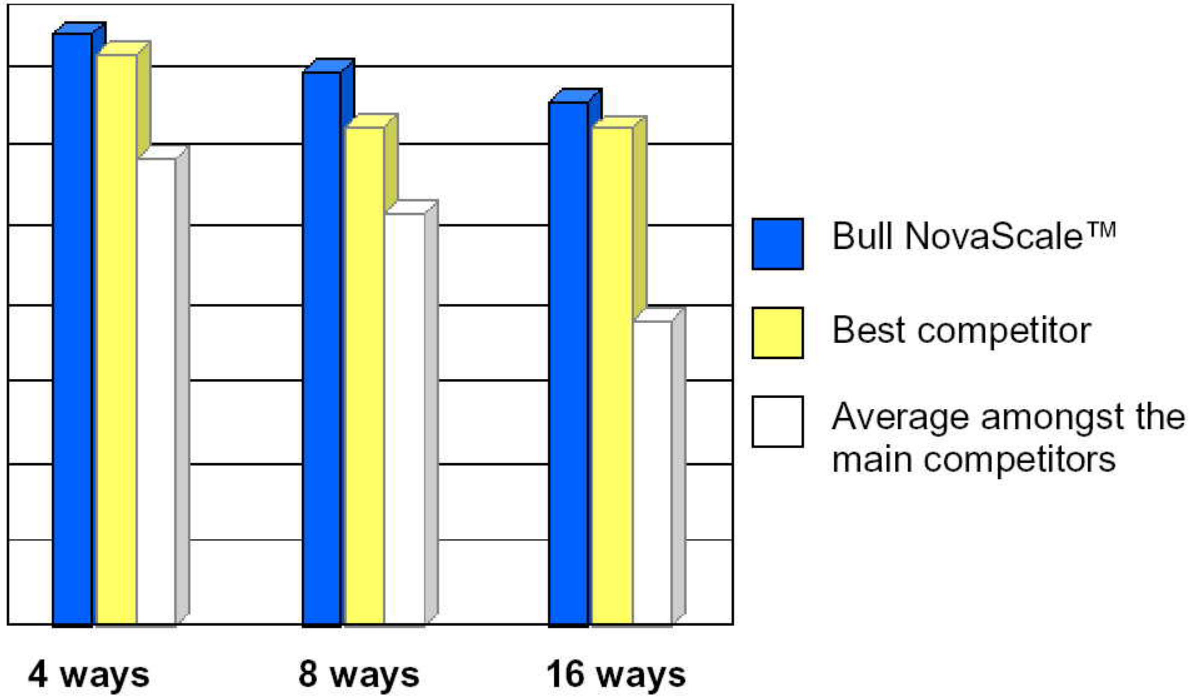
- Jacques Abily
- Anne Kaszynski
- Sylvie Lesmanne
- Solofo Ramangalahy
- Yehong Xing
- Massimo Zendri
- Nicolas Zuanon

INRIA RHÔNE-ALPES

- Damien Bergamini
- Hubert Garavel
- Marc Herbert
- Radu Mateescu
- Bruno Ondet
- Frédéric Perret

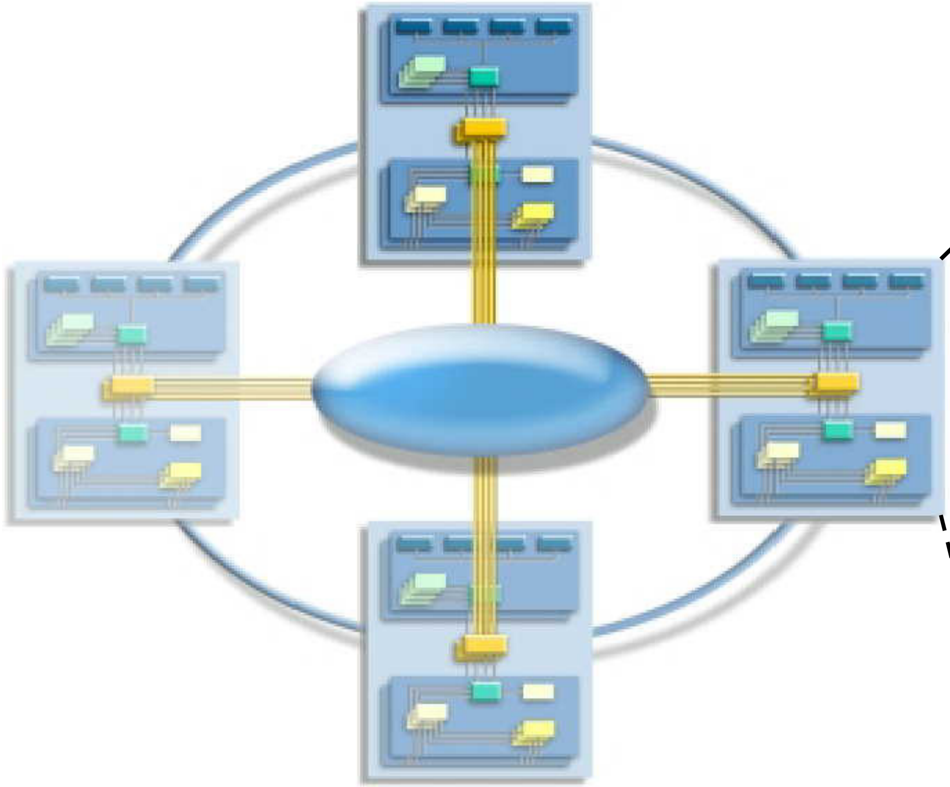
Target: Bull's NovaScale servers

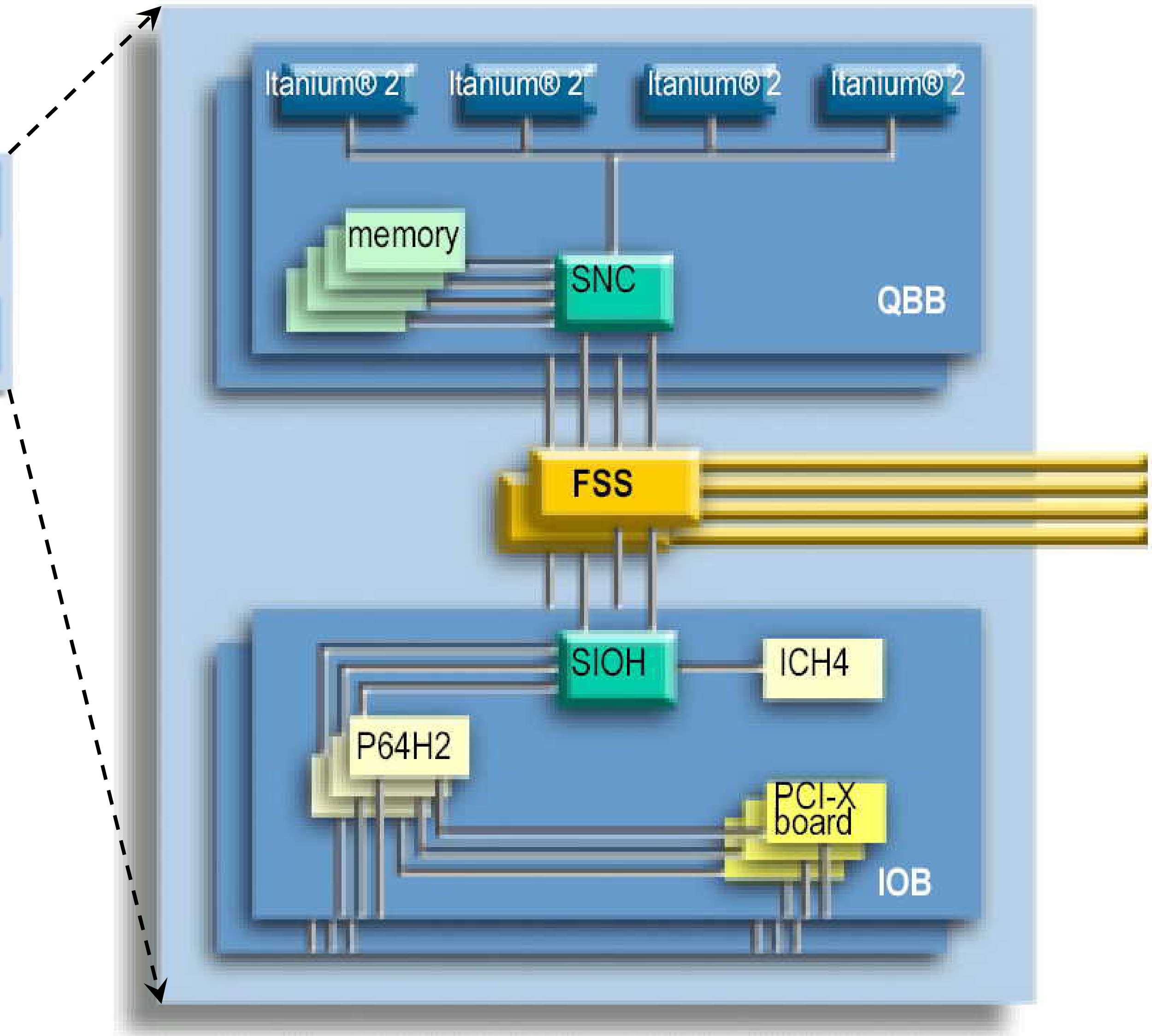
- 64-bit high-end servers
- based on Intel's Itanium-2
- CC-NUMA architecture: "FAME"
- Windows, Linux, GCOS 7 and 8



Servers performance/price ratio (May 2003)

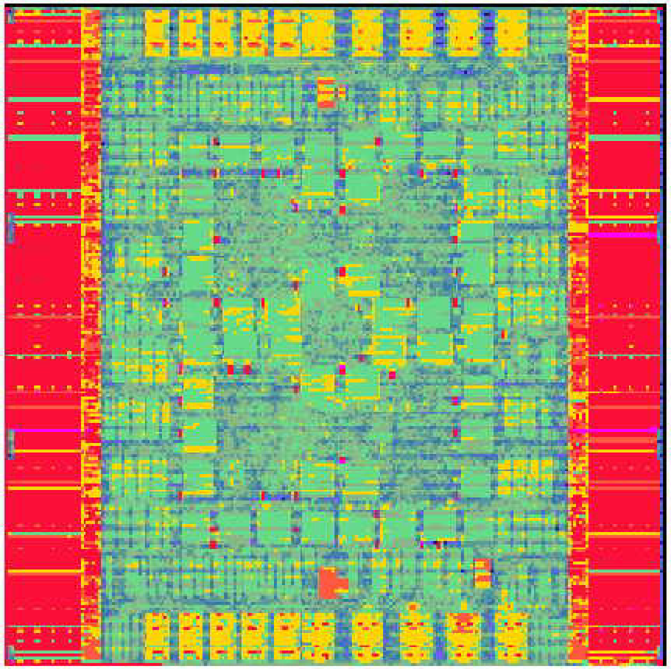
FAME (Flexible Architecture for Multiple Environments)





Focus on most critical, asynchronous parts

- Chipset components for an early prototype of FAME based on Itanium-1 ("Merced") processors:
  - CCS (*Core Chip Set*)
  - NCS (*Network Chip Set*)
- B-SPS / FSS (*Fame Scalability Switch*)
  - core of the FAME architecture
  - implements message routing and cache coherency protocol
  - contains several "units", which themselves contain "blocks"



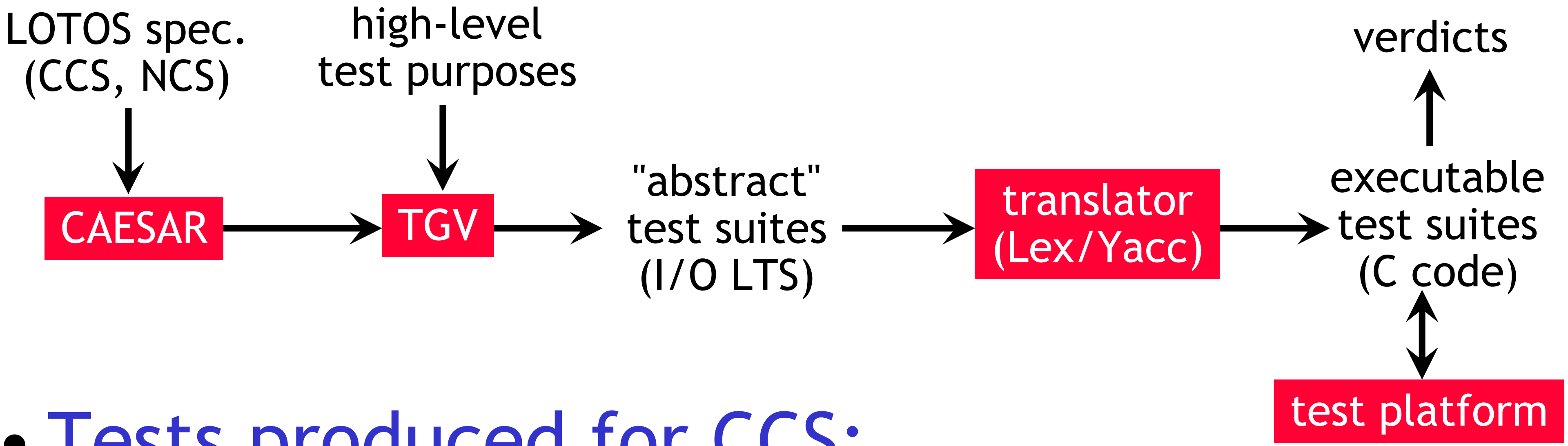
Formal specification activities

- CCS (*Core Chip Set*)
  1,200 lines of LOTOS, 10 concurrent processes
- NCS (*Network Chip Set*)
  1,200 lines of LOTOS, 16 concurrent processes
- B-SPS/FSS (*Fame Scalability Switch*)
  5,000 lines of LOTOS, 12 concurrent processes
  4,500 lines of LOTOS, 7 concurrent processes
- ILU (*Interleaving Unit*)
  8,900 lines of LOTOS, 3,400 lines of C
- PRR (*Pending Requests Response*)
  7,500 lines of LOTOS, 200 lines of C

Formal specification results

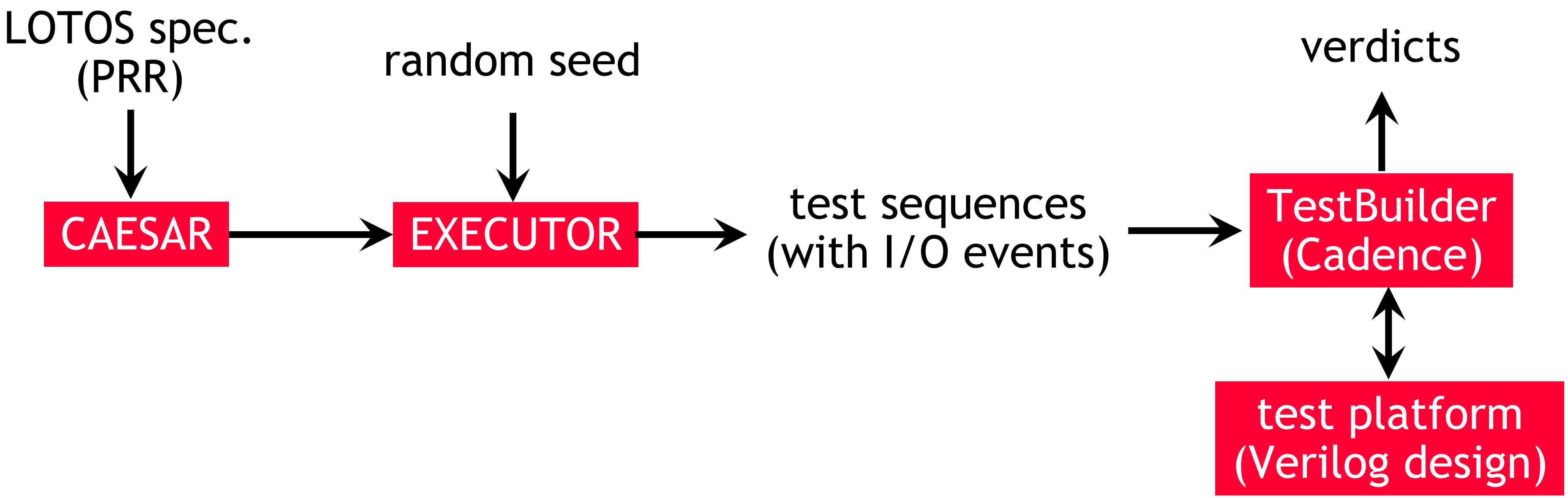
- All the LOTOS code was written by Bull
- Several design levels addressed:
  - system-level: CCS, NCS, B-SPS/FSS
  - unit-level: ILU
  - block-level: PRR
- Various issues detected, e.g., in the cache coherence protocol
  - In 2000: 10 issues raised
  - In 2001: 2 ambiguities pointed out

Directed test generation using TGV



- Tests produced for CCS:
  - 21 base tests (1 mn/test)
  - 50 collision tests (15 mn/test)
  - 1 generalized test (1 day)
- Tests produced for NCS:
  - 50 base tests (30 sec/test)

Random test generation using Executor



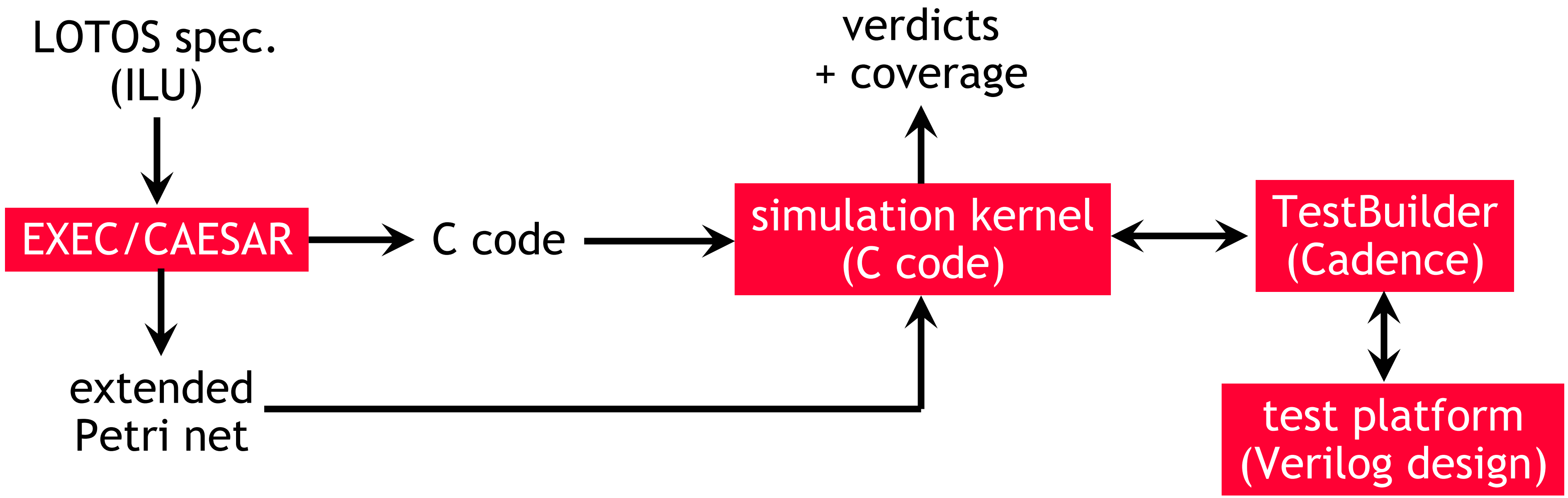
Assumptions:

- PRR is deterministic (same inputs => same outputs)
- randomness introduced in C implementation of LOTOS types

Results:

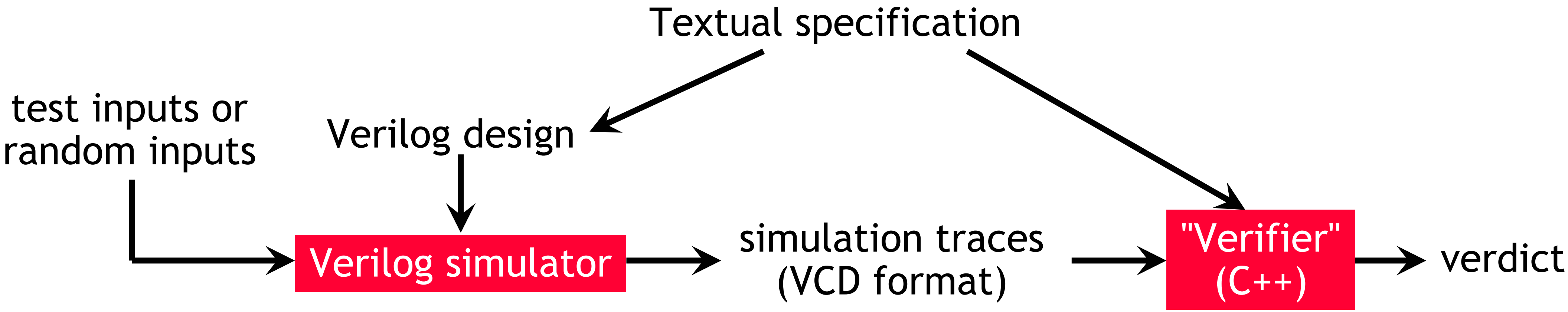
- Generation of large sequences (100,000 transactions)
- Detection of a non-conformity between the LOTOS and Verilog codes for PRR v1 (not detected otherwise)

Co-simulation using Exec/Caesar



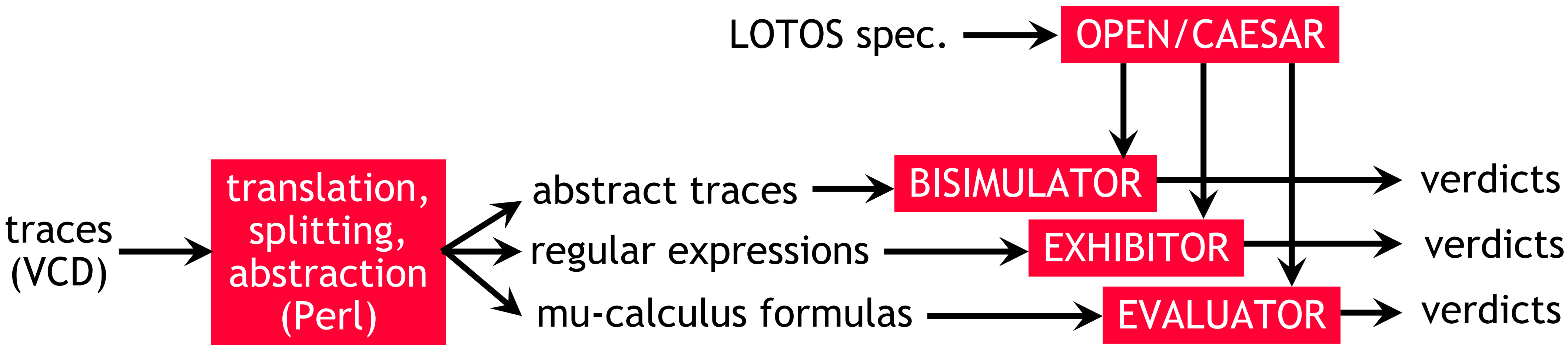
- Various coverage criteria (Petri net transitions, LOTOS visible labels and their offers)
- Combination of random and directed approaches
  - Random firing of tau transitions
  - History-based guidance to maximize coverage

Trace validation: Former approach



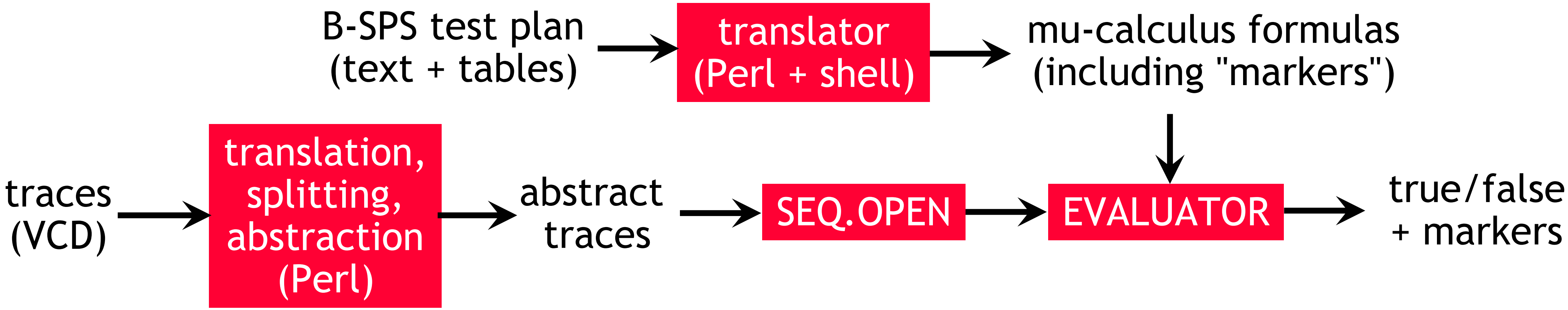
- Goal:
  - find bugs in VCD traces
  - measure coverage of test effort
- Traces are large (> 10,000 bus transactions)
- Traces are complex (nested transactions)
- Writing a dedicated "Verifier" is costly (and it may contain errors)

Trace validation: Formal approaches



- Principle: reuse the LOTOS spec. to check traces
  - *BISIMULATOR: trace inclusion*
  - *EXHIBITOR: regular expression matching*
  - **EVALUATOR: temporal formula satisfaction**
- What about coverage?

Trace validation with coverage



Principles:

- Temporal formulas generated from state/transitions tables
- "Markers" indicate if a formula is "activated" by a given trace
- Formula activated by no trace => more traces needed to cover the test plan
- This measures "functional coverage" (wrt B-SPS specification) — different from "structural coverage" (wrt Verilog design) — not done before by Bull
- SPIN'04 paper on SEQ.OPEN [Garavel-Mateescu-04]

Trace validation with coverage

Main results:

- In 2000: major bug found: ambiguity of informal spec. (also found by "Verifier")
- Collision traces: 130 Mbytes of traces analyzed (about 24,000 transactions): no issue detected
- Interface traces: 761 properties verified, 216 not covered (in fact, 24) => 2 missing tests added in 2001
- Directory traces: 518 properties verified, 196 not covered => 1 missing test added in 2001

- The approach is used at every B-SPS revision (official part of Bull's design methodology)
- Performance: 7.4 millions of model checking jobs done in 23 hours on a standard PC (Pentium III 700 MHz, 1 Gbyte RAM)

Conclusion

Summary

- A long standing research-industry collaboration
- Four different case-studies tackled
- Three different design levels addressed:
  - System (bus arbiters, cache coherency protocols)
  - Unit (ILU)
  - Block (PRR)
- Many functionalities supported:
  - Formal specification
  - Simulation, random execution
  - Hardware emulation, co-simulation
  - Test generation, execution, and coverage
  - Model checking verification
  - Performance evaluation

Conclusions

- The approach is integrated in Bull's industrial process
- Main lessons:
  - LOTOS is usable by architects and verification engineers
  - CADP tools are robust enough (with some maintenance)
  - "High quality" errors have been detected
  - Components developed with LOTOS are more reliable
  - Test effort is better focused on difficult parts
- Future work:
  - Comparative benchmarks with industrial PSL tools
  - Application to asynchronous circuits (VASY + CEA/LETI)

Bibliographic references

- [Chehaibar-Garavel-Mounier-Tawbi-Zulian-96] — "Powescale" case study

  Gh. Chehaibar, H. Garavel, L. Mounier, N. Tawbi, and F. Zulian. *Specification and Verification of the PowerScale Bus Arbitration Protocol: An Industrial Experiment with LOTOS*. Proceedings FORTE/PSTV'96, IFIP, October 1996.

- [Kahlouche-Viho-Zendri-98] — "Polykid" case study (testing aspects)

  H. Kahlouche, C. Viho, and M. Zendri. *An Industrial Experiment in Automatic Generation of Executable Test Suites for a Cache Coherency Protocol*. Proceedings IWTCS'98, IFIP, September 1998.

- [Garavel-Viho-Zendri-01] — "Polykid" case study (all aspects)

  H. Garavel, C. Viho, and M. Zendri. *System Design of a CC-NUMA Multiprocessor Architecture using Formal Specification, Model-Checking, Co-Simulation, and Test Generation*. Springer International Journal on Software Tools for Technology Transfer (STTT), 3(3), July 2001.

- [Garavel-Hermanns-02] — "SCSI-2" case study (performance evaluation aspects)

  H. Garavel and H. Hermanns. *On Combining Functional Verification and Performance Evaluation using CADP*. Proceedings FME'02, Springer Verlag, LNCS 2391, Jul. 2002.

- [Garavel-Mateescu-04] — "FormalFame" case study (trace validation aspects)

  H. Garavel and R. Mateescu. *SEQ.OPEN: A Tool for Efficient Trace-Based Verification*. Proceedings SPIN'04, Springer Verlag, LNCS 2989, April 2004.

- Available from http://www.inrialpes.fr/vasy/Publications
- See also http://www.inrialpes.fr/vasy/dyade