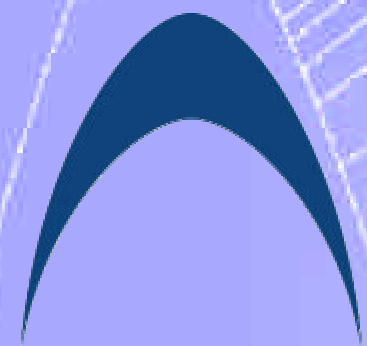




TRANSEDA

VERIFICATION FROM CONCEPT TO REALITY

兼松エレクトロニクス株式会社

現実的なVerificationに向けて、

➤ TransEDA's goals

- ICデザインにおける検証時間の短縮
- verificationメソッドの構築をサポート
- モジュールレベルの検証を提供
- 簡易な導入

ユーザの声 Motorola

“We discovered that code coverage and analysis were the missing links in our golden RTL methodology ...

TransEDA's products were the only ones that met our criteria and provided all of the Verilog code coverage metrics requested by our team.”

Maureen Garnett

Senior Staff Engineer/Scientist, M-CORE team

Motorola SPS

Design Reuseにおける コードカバレッジの重要性

“It is essential to catch bugs as early as possible in the design process, since the time to find and correct a bug increases by an order of magnitude at each level of design integration. ...

Coverage tools such as VeriSure and VHDLCover provide a means of measuring statement and path coverage for RTL designs and testbenches. A coverage tool must be run on the final design and it should indicate 100 percent statement and path coverage ...”

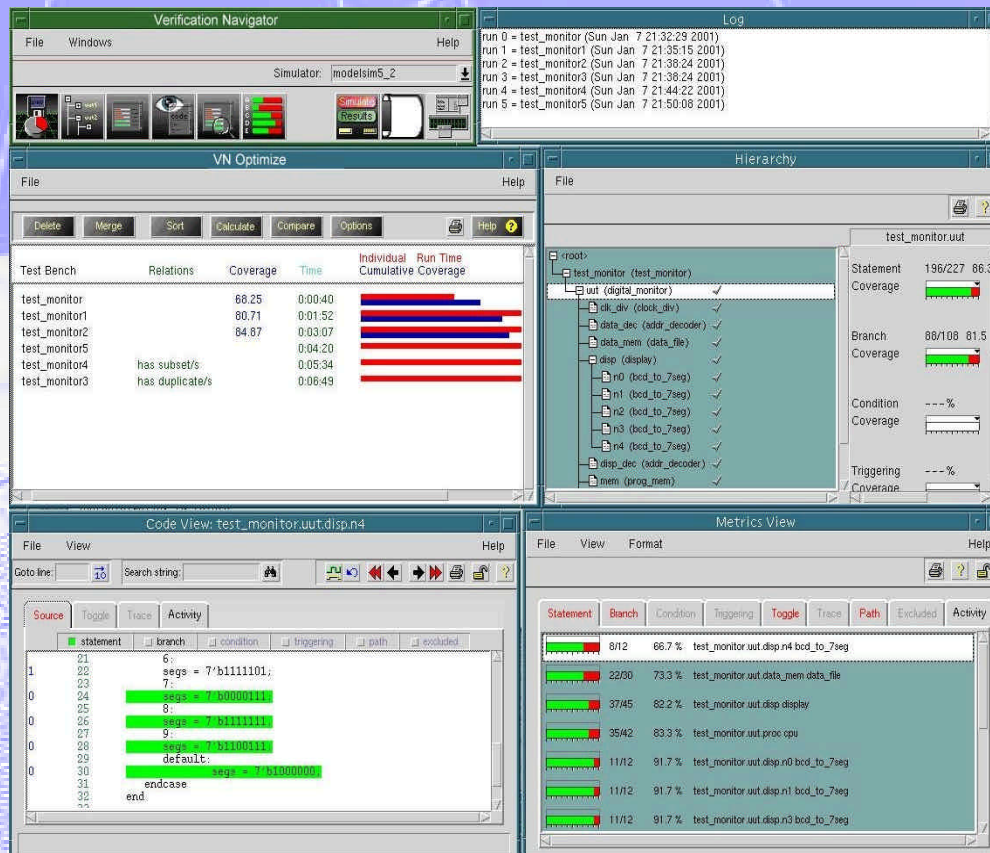
*Reuse Methodology Manual
For System-on-a-Chip Designs
Second Edition
Michael Keating, Synopsys
Pierre Bricaud, Mentor Graphics*

Verification Navigator™

- **Design Verificationに必要なコンポーネントを包括**
 - Code Coverage, Test Suite Optimization, Circuit Activity Analysis, FSM Coverage
- **Dual-Language**
 - シングルカーネルによるVerilogとVHDLのサポート
- **対応Simulator**
 - Verilog-XL, Affirma-NC, VCS, VSS, ModelSim
 - UNIX and Windows-NT

VN-Cover

Verilog & VHDL Code Coverage



- 効果的なテストベンチの作成を示唆
- Verilog/VHDLに対応しIPを有効に活用するdesign re-useをサポート
- カバレッジという定量的な手法によりデザインの質を提供
- 最適なテストベンチの選択

Customer view ...

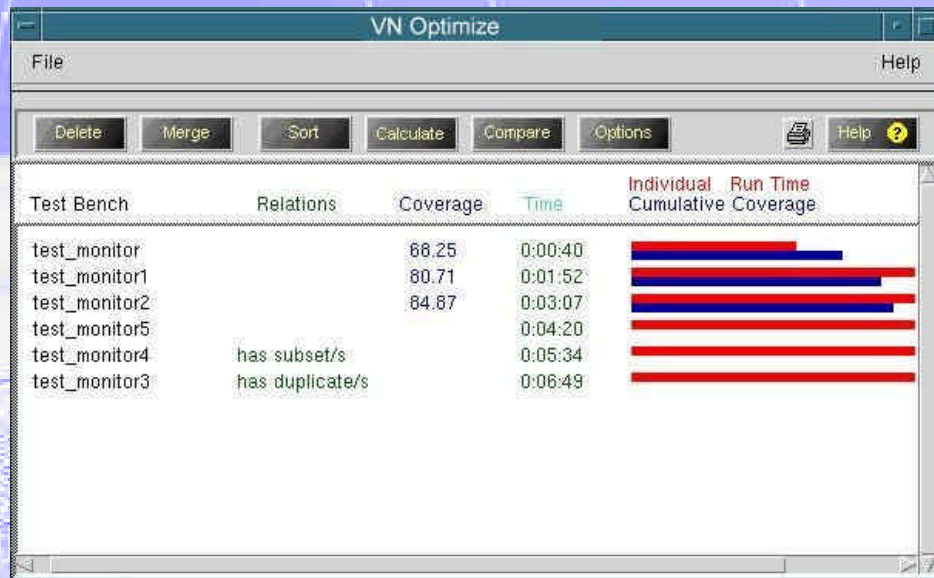
“Verification Navigator is a key element of the design verification environment ...

Its dual-language capability allows us to reduce verification time of complex SOC by quickly identifying areas of low code-coverage whether they are in newly-designed blocks or imported IP cores”

Al Carlson, Director of Hardware Engineering
Charles Industries, Chicago Illinois

VN-Optimize

Test Suite Optimization



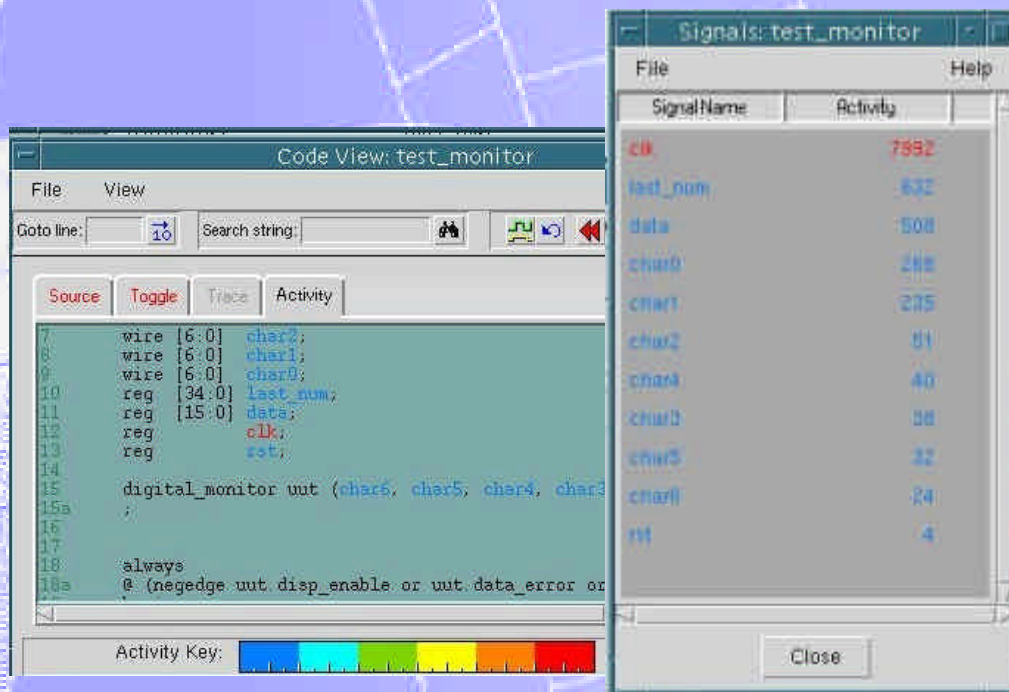
The screenshot shows the VN Optimize application window. It has a menu bar with 'File' and 'Help', and a toolbar with buttons for 'Delete', 'Merge', 'Sort', 'Calculate', 'Compare', 'Options', a printer icon, and a help icon. Below the toolbar is a table with the following columns: 'Test Bench', 'Relations', 'Coverage', 'Time', 'Individual Cumulative Coverage', and 'Run Time'. The table contains data for several test benches, with the last two rows indicating subset and duplicate relationships.

Test Bench	Relations	Coverage	Time	Individual Cumulative Coverage	Run Time
test_monitor		68.25	0:00:40		
test_monitor1		80.71	0:01:52		
test_monitor2		84.87	0:03:07		
test_monitor5			0:04:20		
test_monitor4	has subset/s		0:05:34		
test_monitor3	has duplicate/s		0:06:49		

- 一番有効なテストベンチをオーダリング
- ECO用のレグレッションテストを高速に行なうために必要なテストを、選択
- 過乗検証を行なっている余分なテストを削除

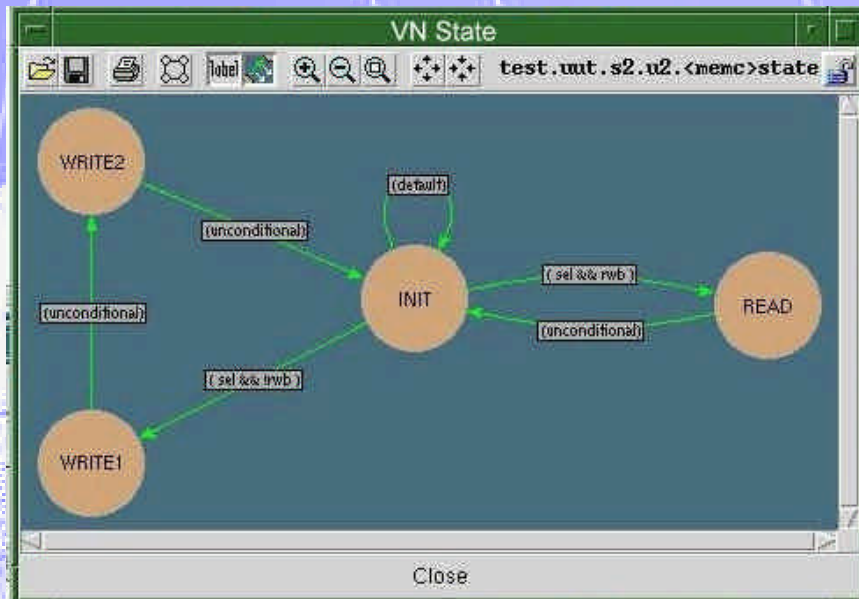
VN-Activity Circuit Activity Analysis

- RTLデザインにおいて、早い段階にて効果的な電力解析を定量的に計測
- 信号やブロックレベルのアクティビティ合計値を表示
- ソースコードとの相対的な表示が可能



VN-State

State Machine Coverage



- Transition coverageにてFSMの検証品質を測る
- FSMを考慮したグラフィカルなドキュメントとして使用
- 簡易な使用方法
 - コーディングスタイルの変更不要
 - ステートマシンの自動抽出
 - バブルチャートを自動描画

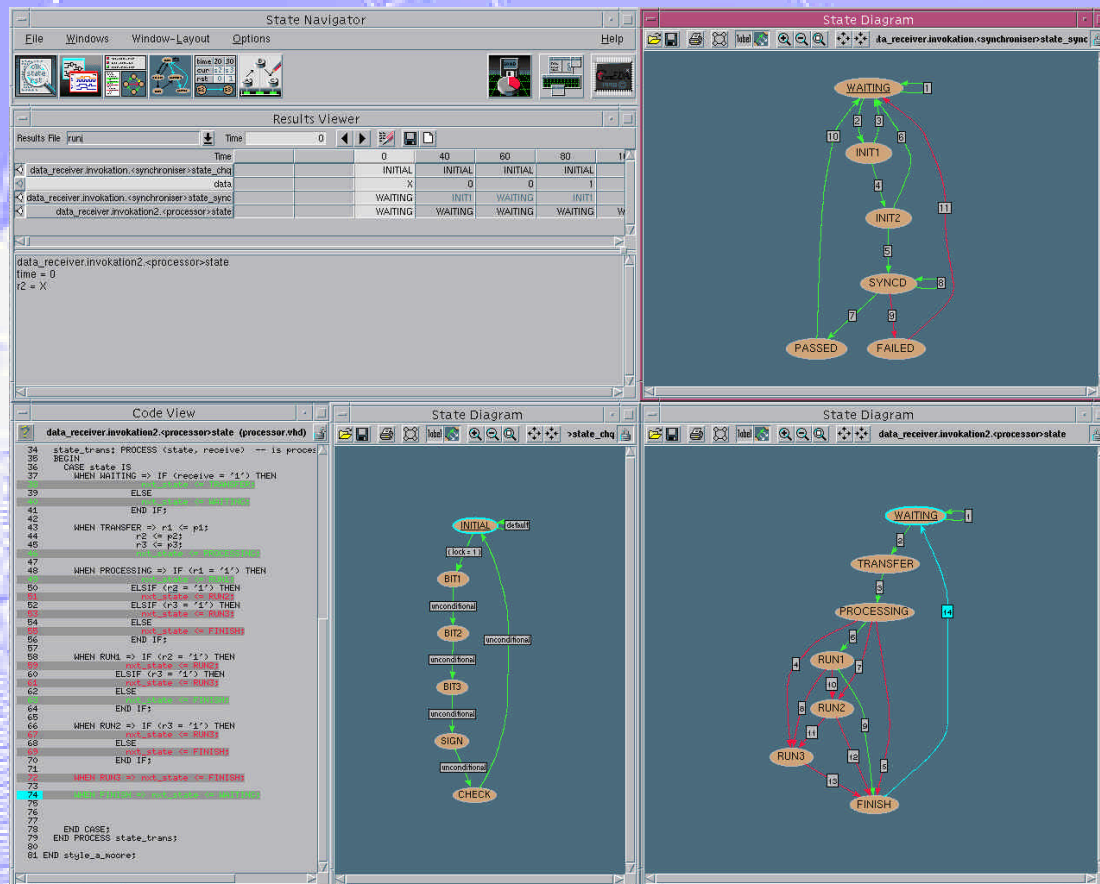
Verification Navigator ロードマップ

- Verilog / VHDL ルールチェック
 - RMM, OMI326, IEEE 1076.6, IEEE 1364.1
 - Custom rules for strategic partners
- より高速なシミュレーション
- ハードウェア設計に対する、C/C++ コードカバレッジ

State Navigatorのアドバンテージ

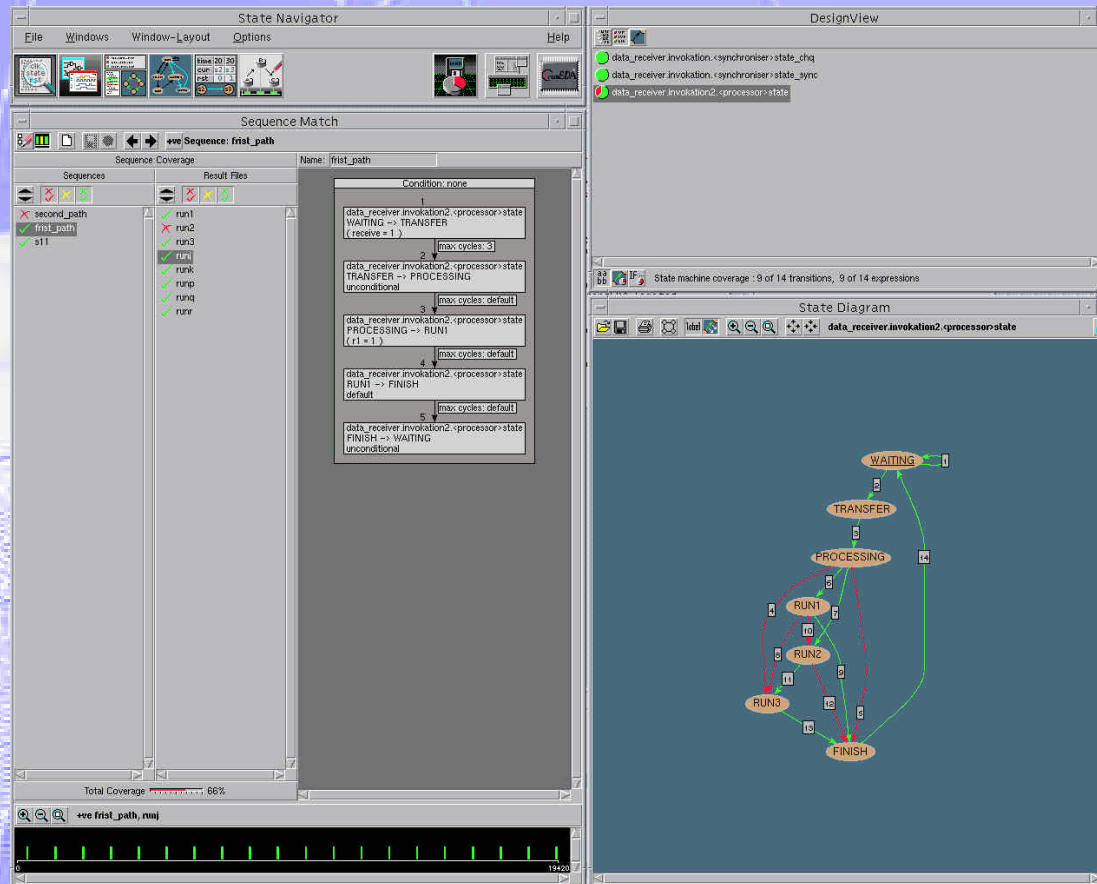
- **Finite State Machine デザインの表示、デバッグとVerification 環境**
 - インタラクティブな FSM Debug, Behavioral Verification, Static Verification, FSM カバレッジ
- **Language Neutral**
 - Verilog とVHDL両言語に対応
- **数多くのSimulatorをサポート**
 - Verilog-XL, Affirma-NC, VCS, VSS, ModelSim
 - UNIX and Windows-NT

FSMsが複数かつ複雑に関連している状態でのデバッグ



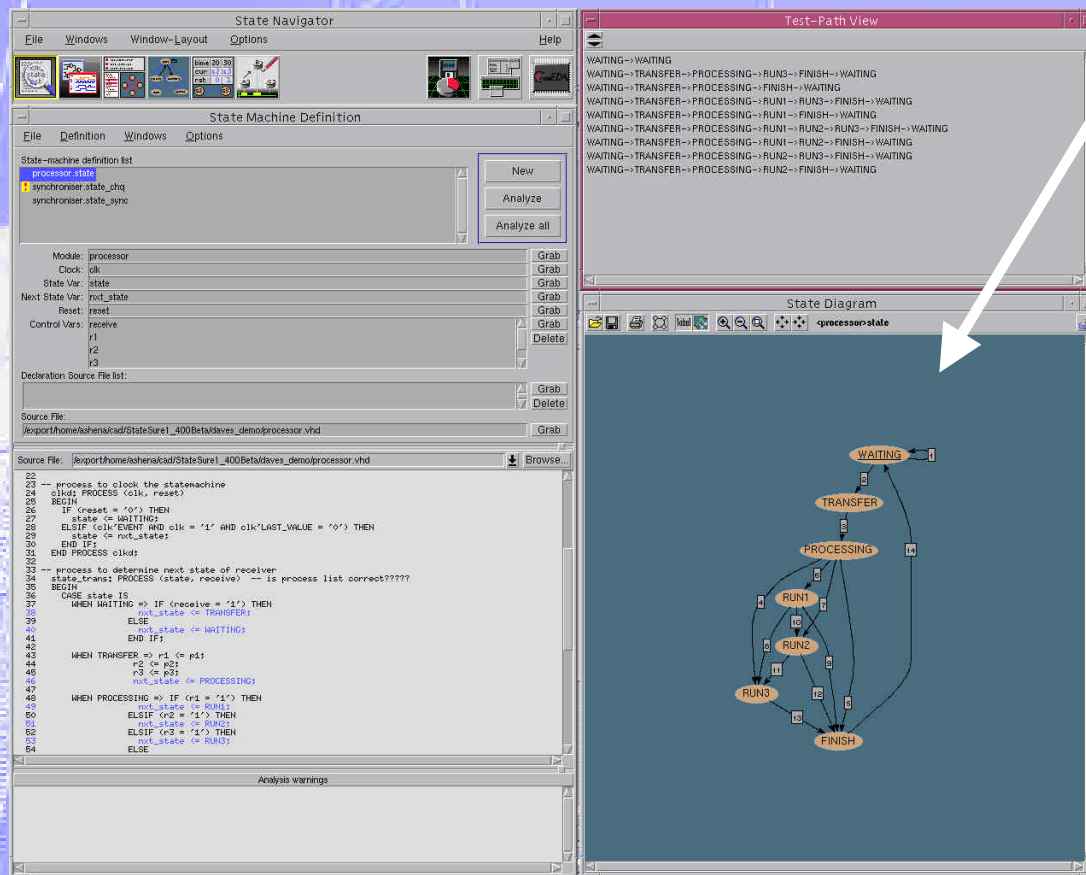
- 多量な FSMs の表示とデバッグが可能
- シミュレーションの結果から状態遷移をアニメーション表示
- 解析のために、ソースコードの同時表示と相対比較

Behavioral Verification



- Dramatically increase verification productivity by raising the level of abstraction
- Define FSM behavior in terms of sequences
 - Powerful graphical behavioral description language
- Match expected behavior with actual behavior

スタティックな Verification

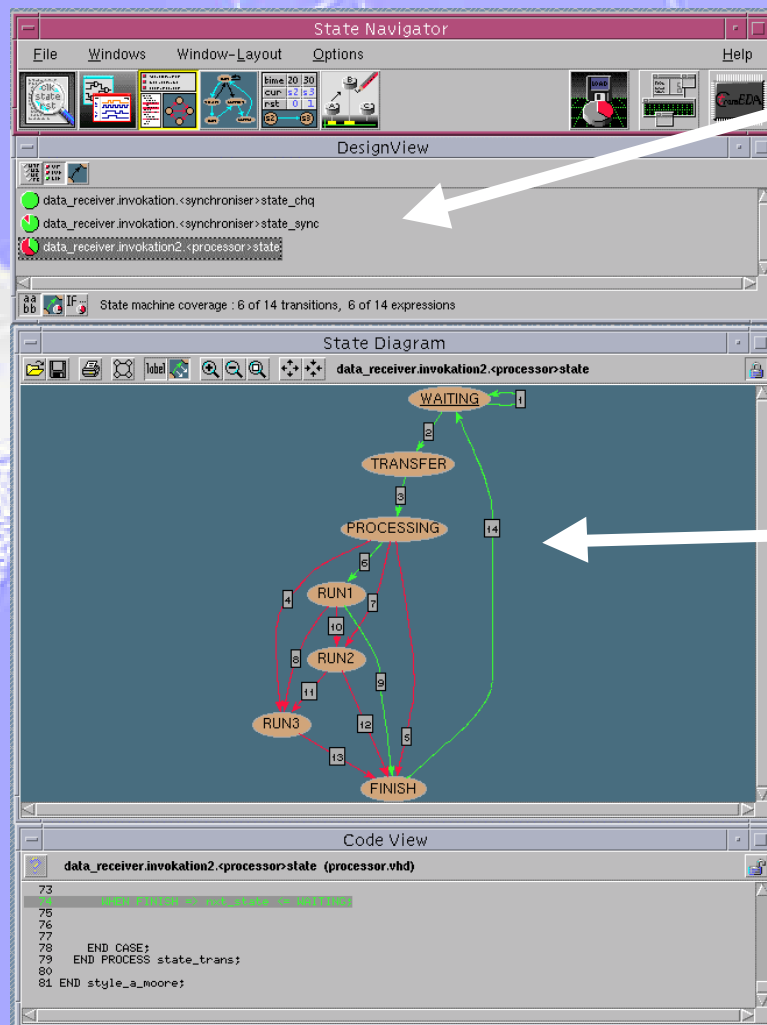


➤ シミュレーション前の
チェック

- デッドステート
- 未到達なステート
- 何度も経過する
トランジション

➤ FSM の複雑さと
verification の難しさを
パスの計測を事前に行
なうことで確認

FSM カバレッジ



トランジションカバレッジと
エクスプレッションカバレッジ

- FSM verification テストの
品質結果を最初に確認

➤ 結果をまず目視チェック

- 追加の verification が
必要な部分を見分ける

トロニクス (株)

State Navigator ロードマップ

- Non-FSM のシーケンシャルロジックのサポート
- Formal FSM チェック
 - デッドロック
 - Constant logic
 - Value reachability
 - Encoding
 - リダンダントロジック
- False シーケンシャル チェック