Symbolic Sequential Verification

Gianpiero Cabodi *Stefano Quer*

Politecnico di Torino
Torino, Italy

{gianpiero.cabodi,stefano.quer}@polito.it
http://staff.polito.it/{gianpiero.cabodi,stefano.quer}/

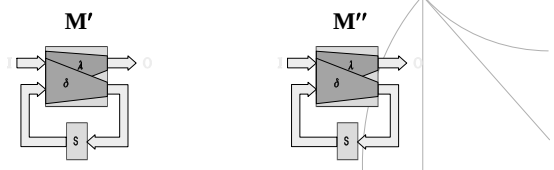
Reference

- **Paper**
- **Books**

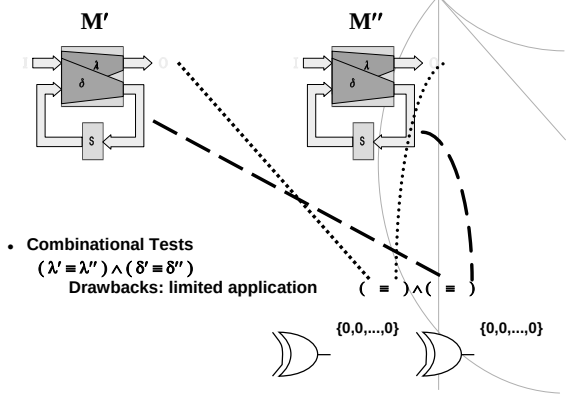
C. Meinel, T. Theobald
"Algorithms and Data Structure in VLSI Design"
Springer-Verlag, Berlin, August 1998
ISBN 3-540-64486-5

G. D. Hachtel, F. Somenzi
"Loginc Synthesis and Verification Algorithms"
Kluwer Academic Publishers

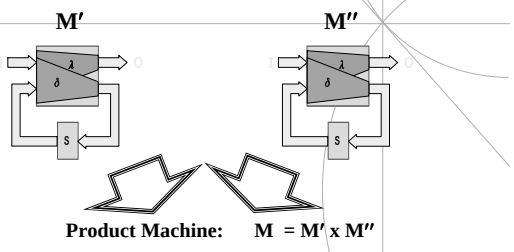
Sequential Verification



Sequential Verification

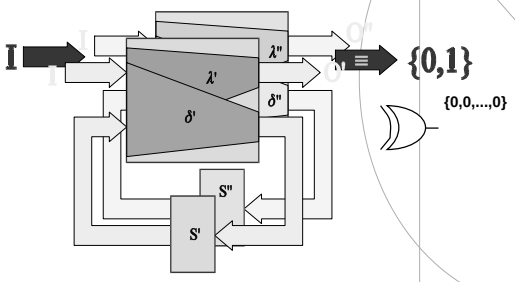


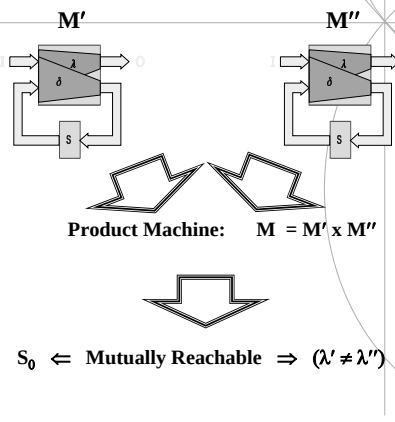
- **Combinational Tests**
  $(\lambda' \equiv \lambda'') \wedge (\delta' \equiv \delta'')$
  Drawbacks: limited application



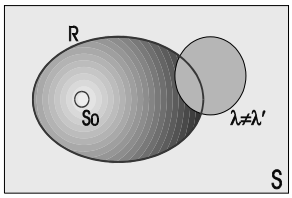
Product Machine (PM)

M = M′ x M″ = (I, {0, 1}, S′ x S″, (δ′, δ″), (λ′ ≡ λ″))

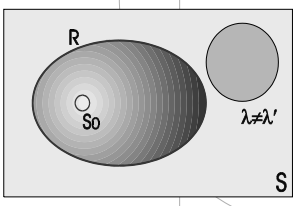




Exact Forward Traversal

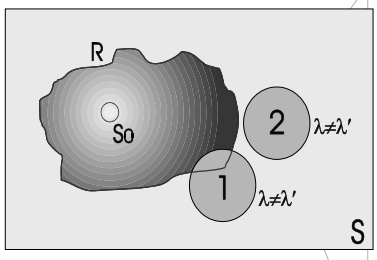


M ≠ M'



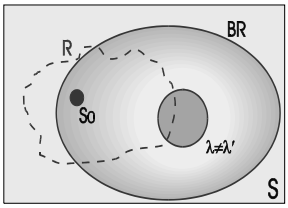
M ≡ M'

Problems

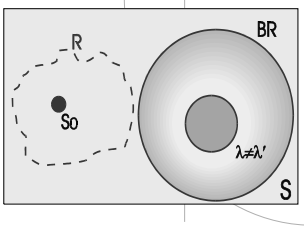


R is
- **too large**
- **too difficult to evaluate**

Exact Backward Traversal

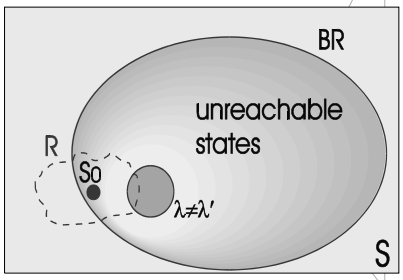


M ≠ M'

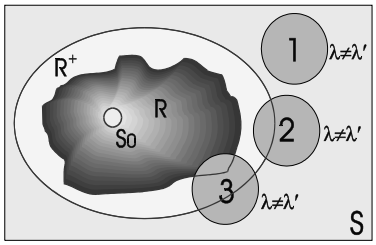


M ≡ M'

Problems



Approximate Forward Traversal



R⁺ = over-estimation of R

Verification

1. **Equivalent in R & R⁺**
2. **NOT Equivalent in R⁺ Equivalent in R**
3. **NOT Equivalent in R & R⁺**

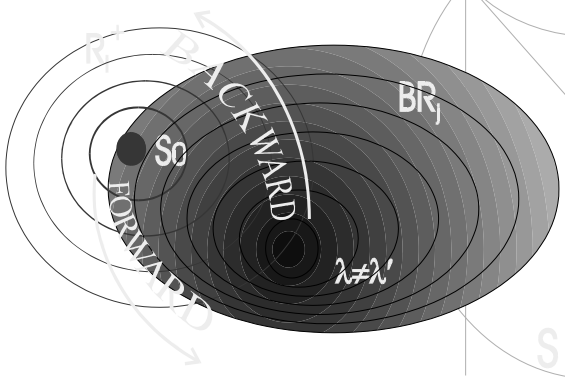
Problems

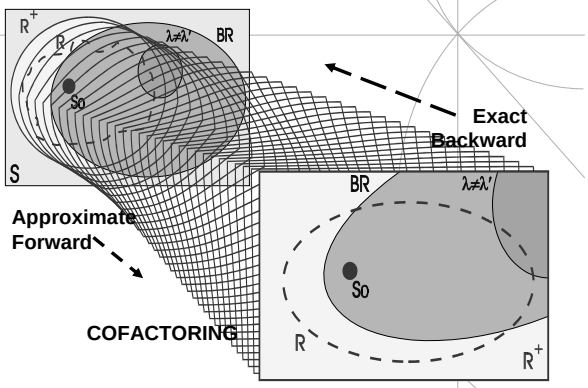
Sufficient condition for equivalence

$$(\lambda \neq \lambda') \cdot R^{+} = 0$$

NOT NECESSARY !!!

Mixed Approaches

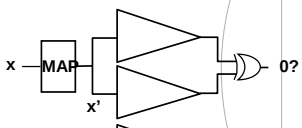




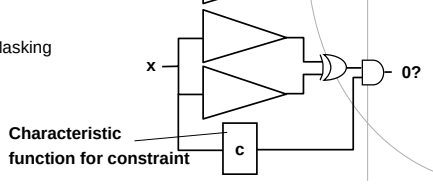
Handling Constraints

- **Input constraints**
  - **Non-occuring input values (don't cares)**
  - **Non-reachable states**
  - **Candidate for R**

1. Input Mapping

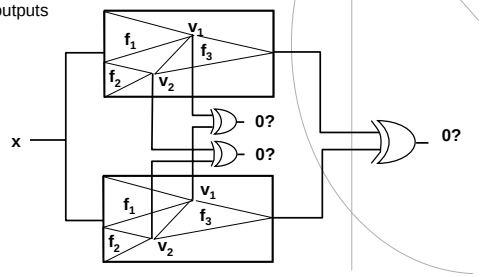


2. Output Masking



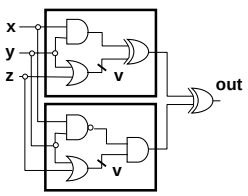
Cutpoint-based EC

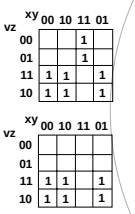
- Cutpoints are used to partition the miter
- Cutpoint guessing
  - Compute net signature with random simulator
  - Sort signatures + select cutpoints
  - Iteratively verify and refine cutpoints
  - Verify outputs



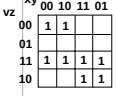
False Negatives

- **Outputs may miscompare for invalid cutpoint values**





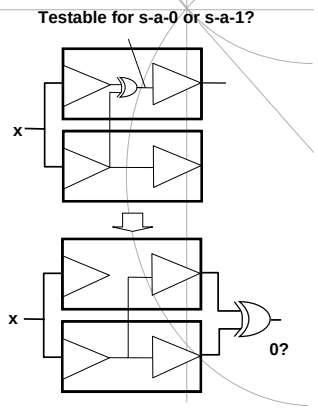
Constraint:
c = (v ≡ y+z)



- **What can we do about false negatives**
  - **Constrain input space to c = (v ≡ y+z)**
  - **If (v in SUPPORT(out)) then out = compose(out, v, fv)**

Permissible Cutpoints

- Based in ATPG
  - Test for s-a-0 at output
  - Checks for permissible functions
  - Test for s-a-1 out output
  - Checks for inverse permissibe functions
- Permissible functions
  - Successively merge circuits
  - From input to outputs

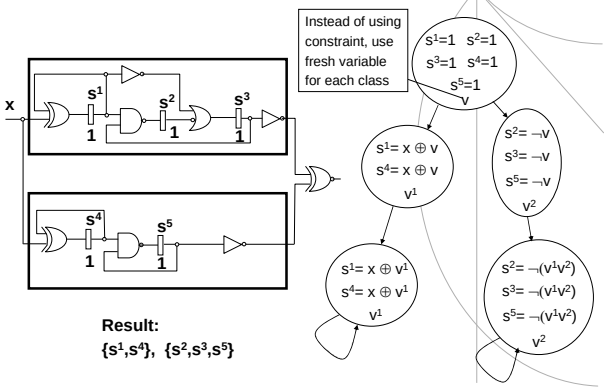


Register Correspondence

- Find registers in product machine that implement identical or complemented function
  - These are matching registers in the two machines under comparison
  - BUT: might be more, we may have redundant registers
- Definition: A register correspondence is an equivalence relation in the set of registers (This definition includes only identical functions, it can be extended to also include complemented functions)
- A register correspondence can be used as a candidate for R
  - $R(s) = \Pi (s_i \equiv s_j)$

$R \subseteq S \times S$

Example



Result:
$\{s^1,s^4\}$, $\{s^2,s^3,s^5\}$

Problems with Functional Reg. Correspondence

- In case of micomparing designs
  - Effect of miscomparing cone may ripple through entire algorithm and split all equivalence classes until they contain only single registers
  - Difficult to debug since no hint of error location
- Solution
  - Relaxation of equivalence criteria
    - e.g. structural register correspondence algorithm based on support set of registers
    - combined techniques with name mapping, functional/structural criteria

Verification Tools

- SMV
  - CMU, Clarke & co.
  - Based on the FSM model
    - From completely synchrounous to completely asynchrounous
    - From detailed to abstract
  - CTL
- COSPAN
  - AT&T Bell Labs, Kurshan & co., LNCS, 1996
  - Language containment, ω-automaton
- Check-Off
  - Abstract Hardware Ltd.
  - Core technology by Siemens
- Design Verifier
  - Chrysalis
  - Equivalence Checker
  - Model Checker under development
- Vformal
  - Compass, core technology by BULL
  - Equivalence Checker
  - Model Checker under development
- RuleBase
  - IBM, core technology by CMU
- FormalCheck
  - Lucent Technologies, core technology COSPAN
  - Properties are defined using templates
  - Increased simplicity, decreased flexibility
- In-house support
  - Intel and Motorola , core technology by CMU

VIS

- **HSIS**
  - **UC-Berkeley, Brayton & co., 1994**
  - **Temporal logic model checking**
  - **Language containment**
  - **Unacceptably Slow for large examples**
- **VIS (Verification Interacting with Synthesis)**
  - **UC-Berkeley, Brayton & co, 1996**
  - **Front-End**
    - **Verilog (VL2MV translator), BLIF, BLIF-MV**
  - **VIS-V**
    - **Simulation**
    - **Temporal logic model checking**
    - **Equivalence checking (combinational and sequential)**
  - **VIS-S**
    - **Synthesis optimizations through SIS**

Verity

- **IBM , Kuehlamnn & co., IBM Journal 1995**
  - **Targeting large CMOS design**
  - **Based on combinational verification (identification of corresponding registers)**
  - **Hierarchical design verification (identical partitioning of the two design compared)**
  - **Commercial name: BoolesEye**
- **Pros**
  - **Different Engines run Subsequently**
  - **Large problems (up to “macros” of 25000 CMOS transistors)**
- **Cons**
  - **Combinational verification**
  - **Structurally similar circuits (often the case)**