Symbolic

Combinational 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**

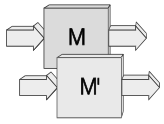
  C. Meinel, T. Theobald
  "Algorithms and Data Structure in VLSI Design"
  Springer-Verlag, Berlin, August 1998
  ISBN 3-540-64486-5

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

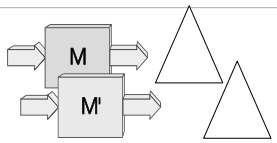
Combinational EC

- **Industrial EC checkers almost exclusively use an combinational EC paradigm**
  - **sequential EC is too complex, can only be applied to design with a few hundred state bits**
  - **combinational methods scale linearly with the design size for a given fixed size and "functional complexity" of the individual cones**
- **Still, pure BDDs are plain SAT solver cannot handle all cones**
  - **BDDs can be built for about 80% of the cones of high-speed designs**
  - **less for complex ASICs**
  - **plain SAT blows up on a "Miter" structure**
- **Contemporary method highly exploit structural similarity of designs to be compared**

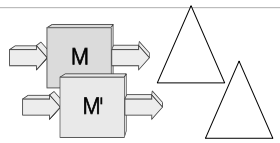
Combinational Verification



Combinational Verification

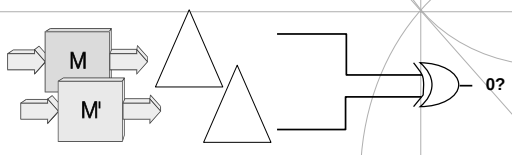


Combinational Verification



Verification (base method)
Compare output's BDDs

Combinational Verification

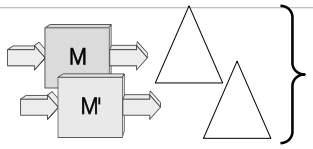


Verification (base method)
Compare output's BDDs

$M_i == M'_i$, $\forall$ i ?

$(M_i \oplus M'_i) == 0$, $\forall$i ?

Combinational Verification

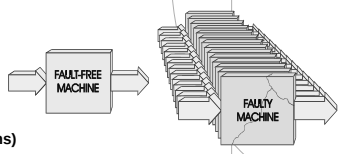


$M_i == M'_i$, $\forall$ i ?

$(M_i \oplus M'_i) == 0$, $\forall$i ?

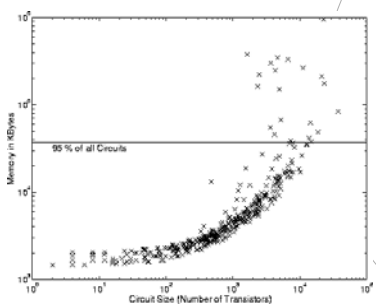
Verification (base method)
Compare output's BDDs

Testing
Fault
- **Stuck-at 0/1**
- **Detectable faults (patterns)**
- **Undetectable faults**



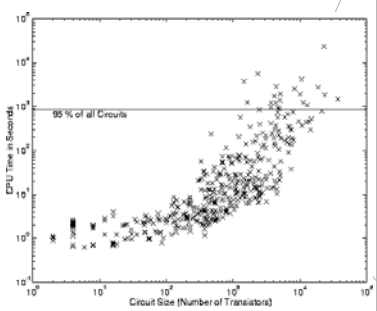
Application of Pure BDDs

Statistics on a PowerPC processor design:

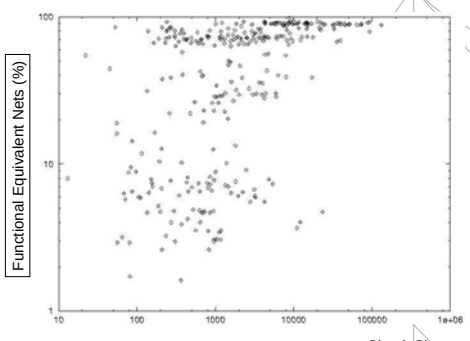


Application of Pure BDDs

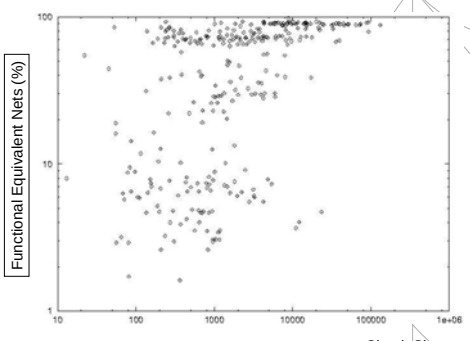
Time for identical set of circuits:



Structural Similarity



Structural Similarity



History of Equivalence Checking

- **SAS (IBM 1978 - 1994)**
  - **standard equivalence checking tool running on mainframes**
  - **based on the DBA algorithm ("BDDs in time")**
  - **verified manual cell-based designs against RTL spec**
  - **handling of entire processor designs**
    - **application of "proper cutpoints"**
    - **application of synthesis routines to make circuits structurally similar**
    - **special hacks for hard problems**
- **BEC (IBM 1991 - 1996)**
  - **workstation based re-implementation of SAS**
  - **mainly used in BooleDozer synthesis environment**
- **Verity (IBM 1992 - today)**
  - **originally developed for switch-level designs**
  - **today IBMs standard EC tool for any combination of switch-, gate-, and RTL designs**
- **Chrysalis (1994 - now Avanti)**
  - **based on ATPG technology and cutpoint exploitation**
  - **very weak if many cutpoints present**
  - **did not adopt BDDs for a long time**
- **Formality (1997 - Synopsys)**
  - **multi-engine technology including strong structural matching techniques**
- **Verplex (1998)**
  - **strong multi-engine based tool**
  - **to our knowledge heavy SAT-based**
  - **very fast front-end**