SWORD: A SAT like Prover Using Word Level Information

Robert Wille, Görschwin Fey, Daniel Große, Stephan Eggersglüß, Rolf Drechsler

University of Bremen, Germany

{rwille,fey,grosse,segg,drechsle}@informatik.uni-bremen.de
Outline

- Motivation
- SWORD
  - Architecture
  - Using Word Level Information
- Experimental Results
- Conclusion & Future Work
Motivation #1

- Solving NP-hard problems
  - Circuit Verification, Property/Equivalence Checking
  - ATPG
  - Graphcoloring
  - ...

- Common: Exploiting **Boolean Satisfiability**
  For a given Boolean function $f$ find an assignment $A$, such that $f(A)=1$ or prove that no such assignment exist.
Motivation #2

SAT Solver

Advanced Techniques
- Efficient Implication Strategies (BCP)
- Conflict based Learning
- Non-chronological Backtracking
- ...

Loss of information
- Properties of modules in circuits
- Position of modules in circuits
- Neighbours of a node in a graph
- ...
Goal

- New Solver that
  - Uses state-of-the-art SAT techniques
  - Works on Word Level
  - Utilizes Word Level information for dedicated solve strategies

→ More compact representation
→ More efficient algorithms
Outline

- Motivation

- SWORD
  - Architecture
  - Using Word Level Information

- Experimental Results

- Conclusion & Future Work
• Modules defined over bitvectors
• Each circuit element has to be supported separately
Architecture

global

Solver

global data
assignments
implGraph
...
do
propagate()
if (confl)
analyzeAndBT()
else
decide()
while (Undef)

interface

Module

decide()
propagate()

local

Adder

decide()
propagate()

AndGate

decide()
propagate()

OrGate

decide()
propagate()
...

...
Flow

**global**

- free var left?
  - yes
  - SAT
  - choose module m
  - for all potentially affected modules n
    - done
    - resolve conflict
      - ok
      - UNSAT
      - failed
    - done
    - conflict
    - next
  - m->decide
  - n->propagate

**interface**

**local**
Outline

- Motivation

- SWORD
  - Architecture
  - Using Word Level Information

- Experimental Results

- Conclusion & Future Work
Global Decision-Heuristic

• Which module makes the best decision?
  – Multiplier often better than an MUX-gate

→ Classify modules into priority-classes

• Priority-class influences the probability that a module makes a decision
Local Decision Heuristic

- **ADDER:**

  \[
  \begin{array}{cccc}
  a_2 & a_1 & a_0 \\
  + & b_2 & b_1 & b_0 \\
  \hline
  c_3 & c_2 & c_1 \\
  s_3 & s_2 & s_1 & s_0 \\
  \end{array}
  \]

  → Deciding unassigned **least significant** bit first provides the most benefit

- realized as FSM
Local Implication

- ADDER:

\[
\begin{array}{c}
+ \\
\hline
\end{array}
\begin{array}{cccc}
a_2 & a_1 & a_0 \\
b_2 & b_1 & b_0 \\
\hline
c_3 & c_2 & c_1 \\
s_3 & s_2 & s_1 & s_0 \\
\end{array}
\]

→ If \( a_i \) and \( b_i \) are assigned, then \( c_i \) and \( s_{i+1} \) are implied
Conflict Analysis and Learning

- Quite similar to the classical approach
  - Separate implication graph
  - Additional clause module

- Improved identification of reasons for conflicts

- Conflict clauses are not learned if they contain variables associated to complex modules like multiplier
Outline

• Motivation

• SWORD
  – Architecture
  – Using Word Level Information

• Experimental Results

• Conclusion & Future Work
Experimental Setup

- Benchmarks
  - Equivalence Checks using multiplier
    - Word Level vs. Word Level (ec_mul_mul)
    - Word Level vs. Partial Products (ec_mul_pp)
    - Word Level vs. Gate Level (ec_mul_gt)
    - Failed Equivalence Check (ec_mul_mul_li)
  - Property Checks (pc_arith_mul)
- Solver:
  - MiniSat v1.14
  - K*BMDs
  - SMT (Yices)
  - SWORD
- AMD Athlon 3500+, 1 GB main memory
Experimental Results

The diagram shows the experimental results for various benchmarks under different conditions. The y-axis represents the performance metric, while the x-axis lists the benchmarks: ec_mul_mul_9, ec_mul_pp_9, ec_mul_gt_9, ec_ml_mul_li_9, pc_arith_mul, and pc_arith_add. The legend indicates the categories SAT, K*BMDS, SMT, and SWORD, each represented by a different color.
Outline

• Motivation

• SWORD
  – Architecture
  – Using Word Level Information

• Experimental Results

• Conclusion & Future Work
Conclusion

- Compact problem representation
- Word Level information is utilized during search process
- Powerful reasoning
Future Work

- Better heuristics & implications-strategies
- Apply further SAT-techniques (restarts, activities ...)
- New conflict analysis (better use of information)
- (Half-)Automatic creation and verification of modules
Questions

- Motivation

- SWORD
  - Architecture
  - Using Word Level Information

- Experimental Results

- Conclusion & Future Work