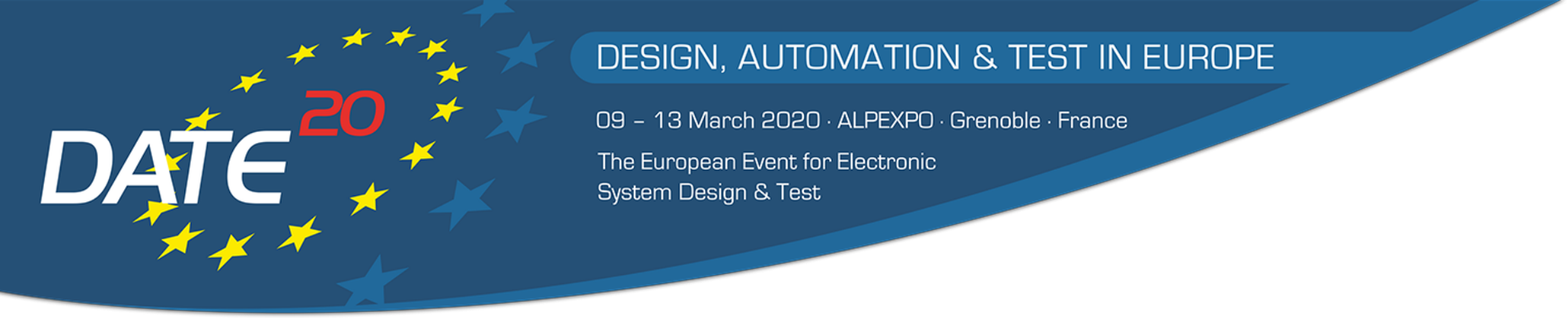


Fast Formal Verification of Multiplier Circuits using Computer Algebra

Daniela Kaufmann

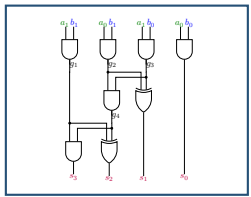
Supervisor: Prof. Armin Biere

JOHANNES KEPLER
UNIVERSITY LINZ

PhD Forum - Design, Automation & Test in Europe 2020

Formal Verification

System



Specification

For all $a_i, b_i \in \mathbb{B}$:

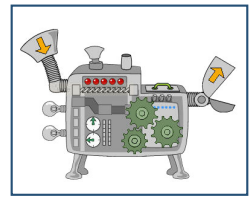
$(2a_1 + a_0) * (2b_1 + b_0) =$

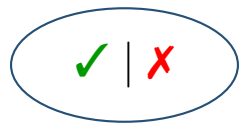
$8s_3 + 4s_2 + 2s_1 + s_0$?

Mathematical Model

$$
\begin{aligned}
B = \{ \\
& x - a_0 * b_0, \\
& y - a_1 * b_1, \\
& s_0 - x * y, \\
& \dots \\
\}
\end{aligned}
$$

Reduction Engine

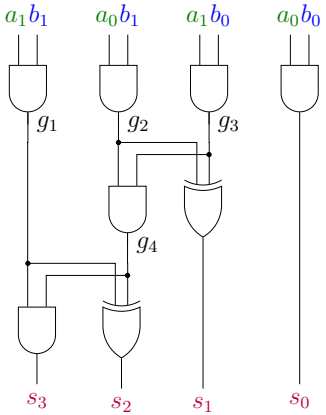




Computer Algebra - Model

Circuit polynomials G.

$$
\begin{aligned}
s_3 = g_1 \wedge g_4 \quad & -s_3 + g_1 g_4, \\
s_2 = g_1 \oplus g_4 \quad & -s_2 - 2g_1 g_4 + g_4 + g_1, \\
g_4 = g_2 \wedge g_3 \quad & -g_4 + g_2 g_3, \\
s_1 = g_2 \oplus g_3 \quad & -s_1 - 2g_2 g_3 + g_2 + g_3, \\
g_1 = a_1 \wedge b_1 \quad & -g_1 + a_1 b_1, \\
g_2 = a_0 \wedge b_1 \quad & -g_2 + a_0 b_1, \\
g_3 = a_1 \wedge b_0 \quad & -g_3 + a_1 b_0, \\
s_0 = a_0 \wedge b_0 \quad & -s_0 + a_0 b_0, \\
a_1 \in \mathbb{B} \quad & -a_1^2 + a_1, \\
a_0 \in \mathbb{B} \quad & -a_0^2 + a_0, \\
b_1 \in \mathbb{B} \quad & -b_1^2 + b_1, \\
b_0 \in \mathbb{B} \quad & -b_0^2 + b_0
\end{aligned}
$$



Computer Algebra - Reduction

Verification Algorithm

Divide specification polynomial $\sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right)$ by circuit polynomials until no further reduction is possible, then C is a multiplier iff remainder is zero.

$$
\begin{aligned}
G = \{ & \\
& -s_3 + g_1 g_4, \\
& -s_2 - 2g_1 g_4 + g_4 + g_1, \\
& -g_4 + g_2 g_3, \\
& -s_1 - 2g_2 g_3 + g_3 + g_2, \\
& -g_1 + a_1 b_1, \\
& -g_2 + a_0 b_1, \\
& -g_3 + a_1 b_0, \\
& -s_0 + a_0 b_0, \\
& -a_1^2 + a_1, \\
& -a_0^2 + a_0, \\
& -b_1^2 + b_1, \\
& -b_0^2 + b_0 \}
\end{aligned}
\qquad\qquad
8s_3 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0
$$

Computer Algebra - Reduction

Verification Algorithm

Divide specification polynomial $\sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right)$ by circuit polynomials until no further reduction is possible, then C is a multiplier iff remainder is zero.

$$
\begin{aligned}
G = \{ & \\
& -s_3 + g_1 g_4, \\
& -s_2 - 2g_1 g_4 + g_4 + g_1, \\
& -g_4 + g_2 g_3, \\
& -s_1 - 2g_2 g_3 + g_3 + g_2, \\
& -g_1 + a_1 b_1, \\
& -g_2 + a_0 b_1, \\
& -g_3 + a_1 b_0, \\
& -s_0 + a_0 b_0, \\
& -a_1^2 + a_1, \\
& -a_0^2 + a_0, \\
& -b_1^2 + b_1, \\
& -b_0^2 + b_0 \}
\end{aligned}
$$

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0$$

$$8g_1 g_4 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0$$

Computer Algebra - Reduction

Verification Algorithm

Divide specification polynomial $\sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right)$ by circuit polynomials until no further reduction is possible, then C is a multiplier iff remainder is zero.

$$
\begin{aligned}
G = \{ & \\
& -s_3 + g_1 g_4, \\
& -s_2 - 2g_1 g_4 + g_4 + g_1, \\
& -g_4 + g_2 g_3, \\
& -s_1 - 2g_2 g_3 + g_3 + g_2, \\
& -g_1 + a_1 b_1, \\
& -g_2 + a_0 b_1, \\
& -g_3 + a_1 b_0, \\
& -s_0 + a_0 b_0, \\
& -a_1^2 + a_1, \\
& -a_0^2 + a_0, \\
& -b_1^2 + b_1, \\
& -b_0^2 + b_0 \}
\end{aligned}
$$

$$
\begin{aligned}
& 8s_3 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
& 8g_1 g_4 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
& 4g_4 + 4g_1 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
& \vdots \\
& 0
\end{aligned}
$$

Contributions - Precise Formalization

[FMCAD'17]

Although this approach has been used before, until now nobody has showed that this method is **sound and complete**.

Theorem (Soundness and Completeness)

Verification successful. $\Leftrightarrow$ *Circuit is a correct multiplier.*

Excerpt:

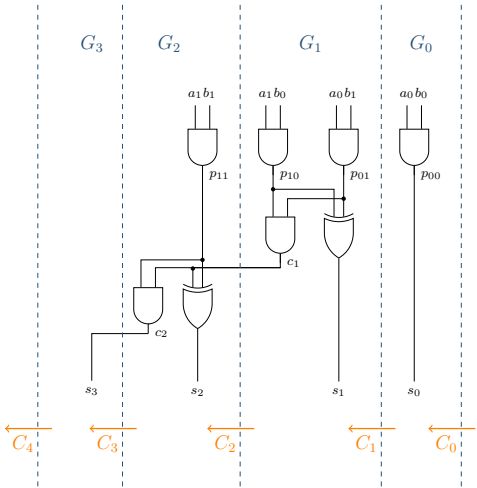
Let $I(C) = \{p \in \mathbb{Q}[X] \mid p(X) = 0 \text{ forall } a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1} \in \mathbb{B}\}$.
Let $J(C) = \langle -s_3 + g_1 g_4, -s_2 - 2g_1 g_4 + g_4 + g_1, \ldots \rangle$.

A circuit C is a multiplier if $\sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right) \in I(C)$.

Soundness and completeness: For all acyclic circuits C, we have $J(C) = I(C)$.

Contributions - Incremental Column-Based Algorithm

[FMCAD'17]



Column-Wise Checking Algorithm

input: Circuit C with sliced Gröbner bases G_i
output: Determine whether C is a multiplier

$C_{2n} \leftarrow 0$

for $i \leftarrow 2n-1$ **to** 0

$C_i \leftarrow$ Remainder $(2C_{i+1} + s_i - P_i,\ \ G_i)$

return $C_0 = 0$

Contributions - Simplified Rewriting Techniques

[DATE'18, FMCAD'19]

Rewriting

- Previously: Identify syntactic patterns in the circuit.
- Now: Identify single-dependency variables.
- Eliminate internal variables.
- Uses elimination theory of Gröbner bases.

Example: Full Adder

$$\left.\begin{array}{l} -c_2 + g_2 + g_1 - g_2 g_1, \\ -s_2 + g_0 + c_1 - 2g_0 c_1, \\ -g_2 + g_0 c_1, \\ -g_1 + p_{20} p_{11}, \\ -g_0 + p_{20} + p_{11} - 2p_{20} p_{11} \end{array}\right\} -2c_2 - s_2 + p_{20} + p_{11} + c_1$$

Contributions - Modular Reasoning

[FMCAD'19]

- Previous work: polynomial ring $\mathbb{Q}[X]$.
- Now: polynomial ring $\mathbb{Z}_{2^l}[X]$, with $l \in \mathbb{N}$.

- Eliminates certain vanishing monomials.
- Allows verification of truncated multipliers.

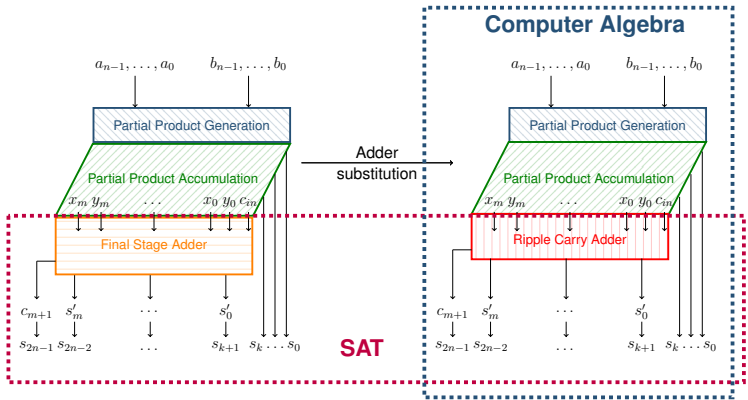
Example: 4-bit multiplier with $l = 2^8$

polynomial ring	sec	max. size
$\mathbb{Q}[X]$	0.07	2 264
$\mathbb{Z}_{2^8}[X]$	0.00	36

Contributions - SAT & Computer Algebra

[FMCAD'19, DATE'20]

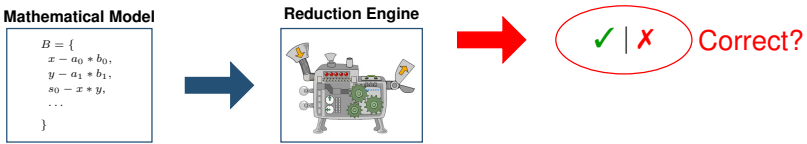
- Substitute final stage adders by simpler adder circuits.
- Correctness of replacement is verified using SAT.
- Rewritten circuit is verified using computer algebra.



Contributions - Proof Certificates

[SC2'18]

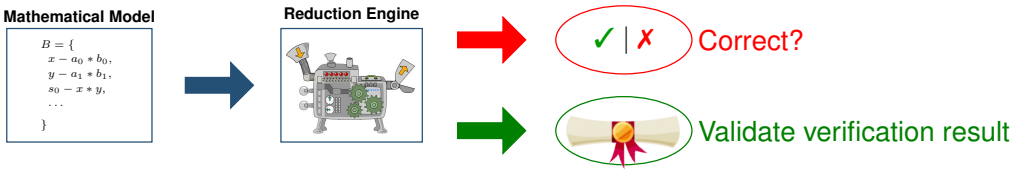
- Can we trust the decision process?
- Can we trust our own implementation of the complex optimizations?



Contributions - Proof Certificates

[SC2'18]

- Can we trust the decision process?
- Can we trust our own implementation of the complex optimizations?



Practical algebraic calculus: Sequence $P = (p_1, \ldots p_n)$, where each p_k is obtained by:

$+:\ p_i,\ p_j,\ p_i + p_j;$ p_i, p_j appearing earlier in the proof

$*:\ p_i,\ q,\ qp_i;$ p_i appearing earlier in the proof, q arbitrary.

Experimental Results - Verification

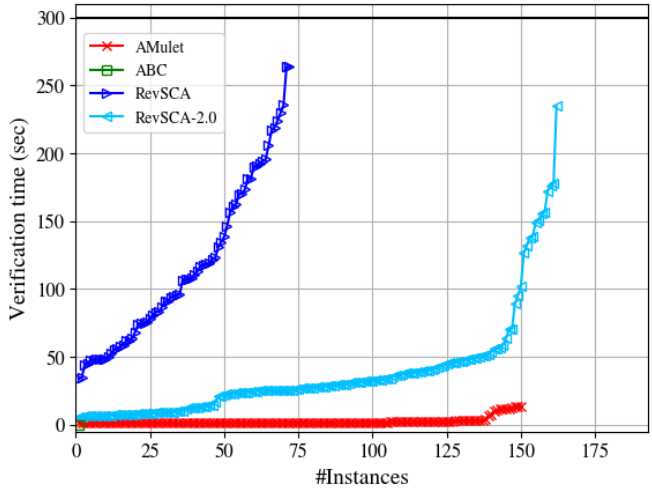
[https://github.com/d-kfmnn/amulet]

Verification Tool AMulet.

- 192 multiplier architectures.
- Input bit-width $n = 64$.
- Time out: 300 sec.

Comparing to

- ABC: M. Ciesielski et al., TCAD, 2019.
- RevSCA: A. Mahzoon et al., DAC, 2019.
- RevSCA-2.0: A. Mahzoon et al., 2020.



Experimental Results - Proof Certificates

[http://fmv.jku.at/pac/]

Verification Tool AMulet & Proof Checker PACtrim.

multiplier	bit-width	Verify				Verify + Certificates				Check Certificates			total
		sub	sat	c.alg	tot	sub	sat	c.alg	tot	sat	c.alg	tot	
btor	512	0	0	16	16	0	0	23	23	0	7	7	30
kjvnkv	512	0	0	13	13	0	0	15	15	0	9	9	25
sp-ar-rc	512	0	0	13	13	0	0	16	16	0	10	10	26
sp-dt-lf	512	1	0	25	26	1	0	25	26	0	11	11	37
sp-wt-bk	512	1	0	26	27	0	0	26	26	0	11	11	38
btor	1024	2	0	177	179	2	0	219	219	0	51	51	272
kjvnkv	1024	2	0	91	93	2	0	172	172	0	72	72	245
btor	2048	17	0	1 493	1 510	17	0	2 552	2 552	0	430	430	2 982
kjvnkv	2048	18	0	1 129	1 147	18	0	2 077	2 077	0	1 228	1 228	3 307

- Time in min.
- Scales up to bit-width 2048.
- None of related work produces certificates.

References

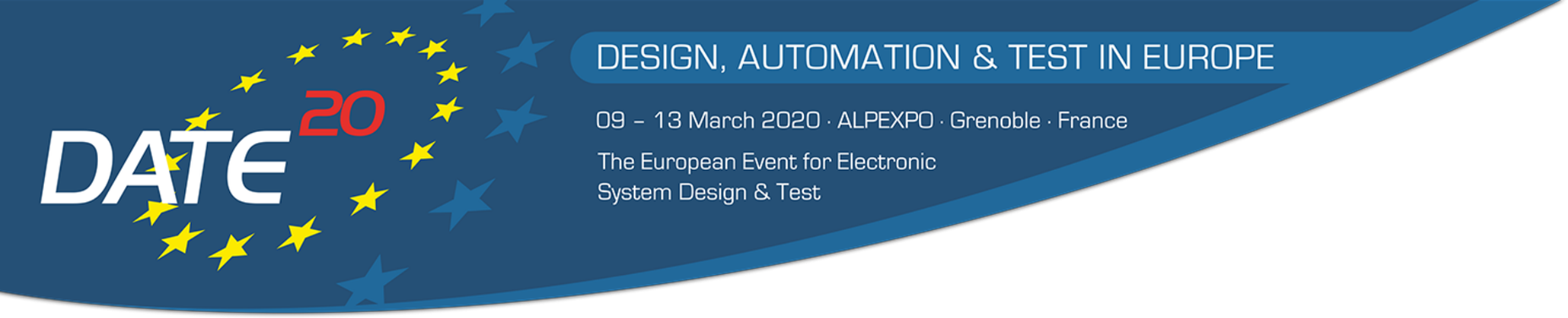
My Work:

- D. Ritirc[a], A. Biere, M. Kauers.
  Column-Wise Verification of Multipliers Using Computer Algebra.
  FMCAD, 2017: 23–30.
- D. Ritirc, A. Biere, M. Kauers.
  Improving and Extending the Algebraic Approach for Verifying Gate-Level Multipliers.
  DATE, 2018: 1556–1561.
- D. Ritirc, A. Biere, M. Kauers.
  A Practical Polynomial Calculus for Arithmetic Circuit Verification.
  SC^2, 2018: 61–76.
- D. Kaufmann, A. Biere, M. Kauers.
  Verifying Large Multipliers by Combining SAT and Computer Algebra.
  FMCAD, 2019: 28–36.
- D. Kaufmann, A. Biere, M. Kauers.
  From DRUP to PAC and Back.
  DATE, 2020: to appear.

[a] Some of my work is published under my maiden name ‘Ritirc‘.

Related Work:

- ABC:
  M. Ciesielski, T. Su, A. Yasin, and C. Yu.
  Understanding Algebraic Rewriting for Arithmetic Circuit Verification: a Bit-Flow Model.
  IEEE TCAD, pages 1–1, 2019.
- RevSCA:
  A. Mahzoon, D. Große, and R. Drechsler.
  RevSCA: Using Reverse Engineering to Bring Light into Backward Rewriting for Big and Dirty Multipliers.
  In DAC, pages 185:1–185:6. ACM, 2019.
- RevSCA-2.0:
  A. Mahzoon, D. Große, and R. Drechsler.
  http://sca-verification.org/



Fast Formal Verification of Multiplier Circuits using Computer Algebra

Daniela Kaufmann

Supervisor: Prof. Armin Biere

JOHANNES KEPLER
UNIVERSITY LINZ

PhD Forum - Design, Automation & Test in Europe 2020