USING GRÖBNER BASES FOR ARITHMETIC CIRCUIT VERIFICATION

Daniela Kaufmann

TU Wien, Austria

PolSys Seminar
Sorbonne Université - LIP6
Paris, France

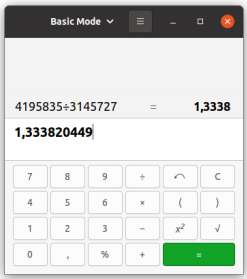
April 7, 2025

TU WIEN Informatics 20 YEARS

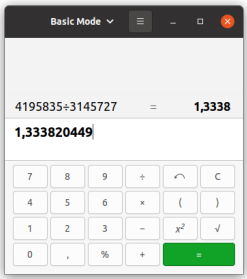
FWF Austrian Science Fund

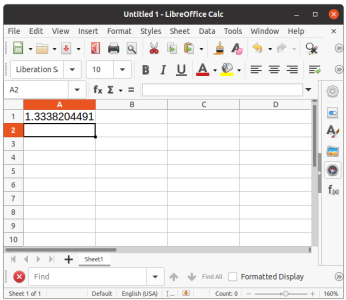
$$4\,195\,835 \div 3\,145\,727 = ?$$

$$4\,195\,835 \div 3\,145\,727 = ?$$

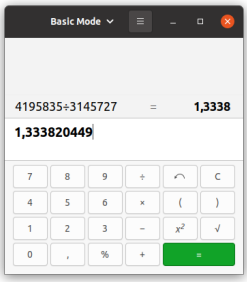


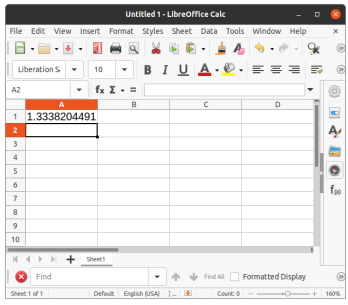
$$4\,195\,835 \div 3\,145\,727 = ?$$

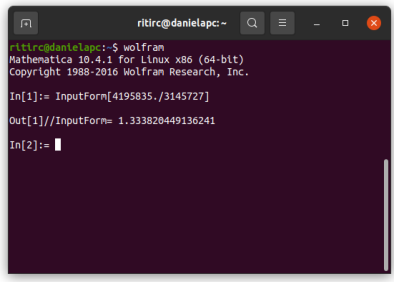


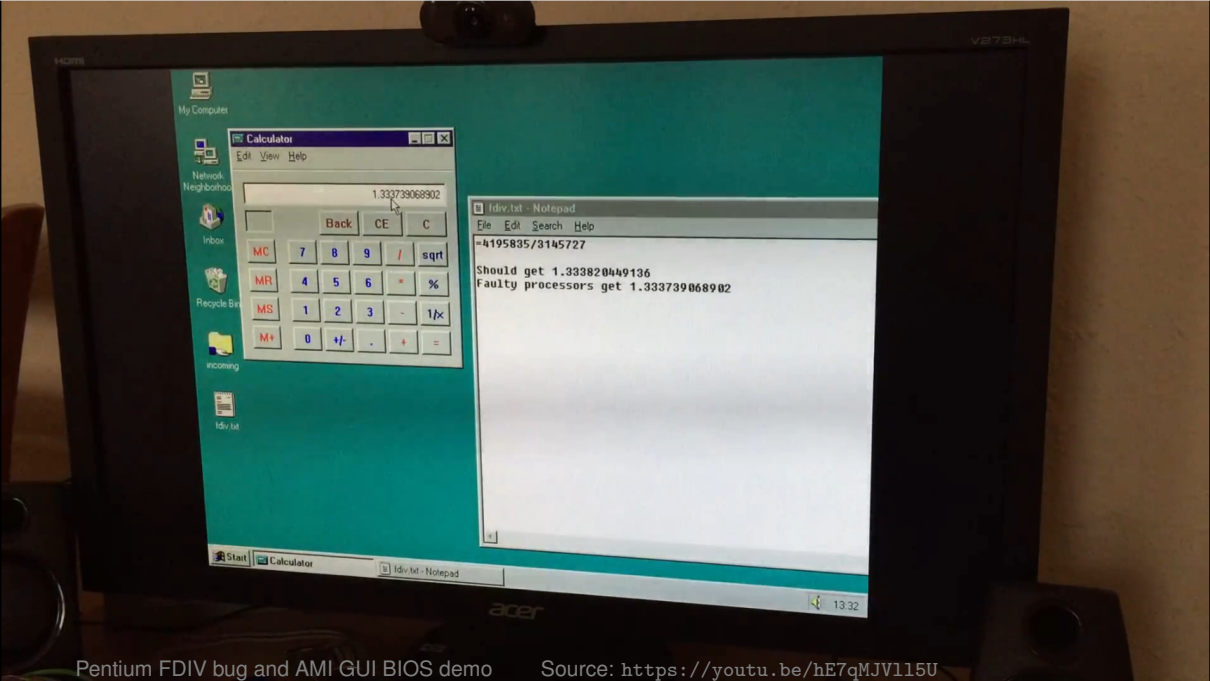


$$4\,195\,835 \div 3\,145\,727 = ?$$









Pentium FDIV bug and AMI GUI BIOS demo Source: https://youtu.be/hE7qMJV115U

Intel Pentium FDIV-Bug 1994

Source: http://neology.com.au/portfolios/a80502-90-sx923/

- Affected floating point unit (FPU) in early Intel processors.
- Processor might return incorrect result for division.
- Cost in 1994: 500 million dollars.

Even more than 30 years later verification of arithmetic circuits is considered to be hard. Correctness proofs are not fully automated yet.

Challenge: Integer multipliers

Integer Multiplication

$$\begin{array}{r} 1\ 1 \cdot 1\ 1 \\ \hline \end{array}$$

$3 \cdot 3 = 9$

Integer Multiplication

$$\begin{array}{cccc} 1 & 1 & \cdot & 1 & 1 \\ \hline & & & 1 & 1 \end{array}$$

$3 \cdot 3 = 9$

Integer Multiplication

$$
\begin{array}{cccc}
1 & 1 & \cdot & 1 & 1 \\
\hline
 & & & 1 & 1 \\
 & & 1 & 1 & \\
\hline
\end{array}
$$

$3 \cdot 3 = 9$

Integer Multiplication

$$
\begin{array}{rrcrr}
1 & 1 & \cdot & 1 & 1 \\
\hline
 & & & 1 & 1 \\
 & & 1 & 1 & \\
 & & & & 0 \\
\hline
 & & & & 1
\end{array}
$$

$3 \cdot 3 = 9$

Integer Multiplication

$$
\begin{array}{rrcrr}
1 & 1 & \cdot & 1 & 1 \\
\hline
 & & & 1 & 1 \\
 & & 1_{1} & 1_{0} & \\
\hline
 & & & 0 & 1
\end{array}
$$

$3 \cdot 3 = 9$

Integer Multiplication

$$
\begin{array}{rrcrr}
1 & 1 & \cdot & 1 & 1 \\
\hline
 & & & 1 & 1 \\
 & & 1 & 1 & \\
 & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 0} & \\
\hline
 & & 0 & 0 & 1
\end{array}
$$

$3 \cdot 3 = 9$

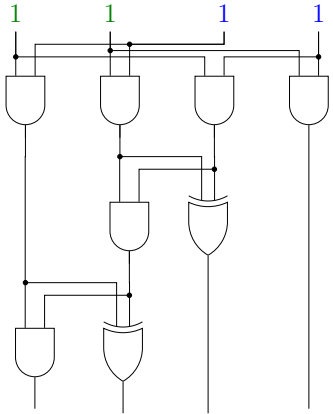
Integer Multiplication

$$
\begin{array}{cccc}
1 & 1 & \cdot & 1 & 1 \\
\hline
 & & & 1 & 1 \\
 & & 1 & 1 & \\
 & 1 & 1 & 0 & \\
\hline
 & 1 & 0 & 0 & 1
\end{array}
$$

$3 \cdot 3 = 9$

Integer Multiplication

$$
\begin{array}{cccc}
1 & 1 & \cdot & 1 & 1 \\
\hline
 & & & 1 & 1 \\
 & & 1 & 1 & \\
\hline
 & 1 & 0 & 0 & 1
\end{array}
$$



$3 \cdot 3 = 9$

Integer Multiplication

AND-Gate

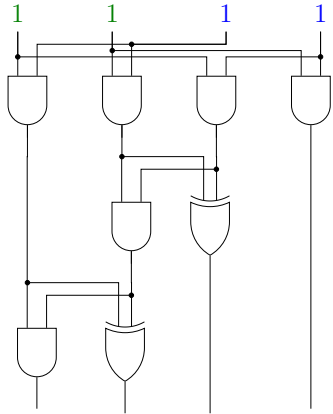
$f \wedge g = y$

f	g	y
0	0	0
0	1	0
1	0	0
1	1	1

XOR-Gate

$f \oplus g = y$

f	g	y
0	0	0
0	1	1
1	0	1
1	1	0



Integer Multiplication

AND-Gate

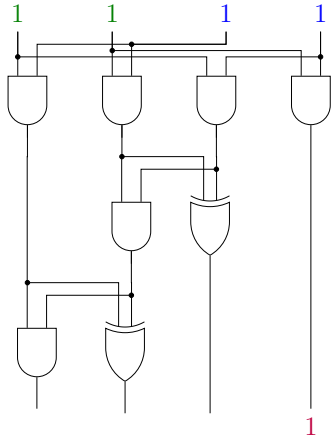
$f \wedge g = y$

f	g	y
0	0	0
0	1	0
1	0	0
1	1	1

XOR-Gate

$f \oplus g = y$

f	g	y
0	0	0
0	1	1
1	0	1
1	1	0



Integer Multiplication

AND-Gate

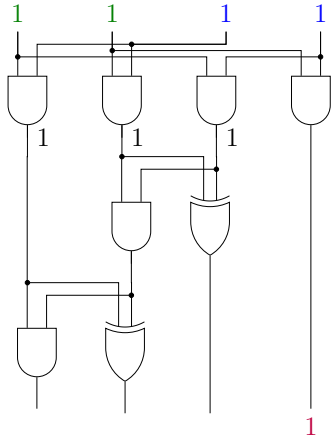
$f \wedge g = y$

f	g	y
0	0	0
0	1	0
1	0	0
1	1	1

XOR-Gate

$f \oplus g = y$

f	g	y
0	0	0
0	1	1
1	0	1
1	1	0



Integer Multiplication

AND-Gate

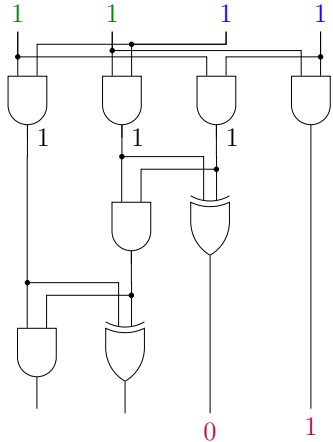
$f \wedge g = y$

f	g	y
0	0	0
0	1	0
1	0	0
1	1	1

XOR-Gate

$f \oplus g = y$

f	g	y
0	0	0
0	1	1
1	0	1
1	1	0



Integer Multiplication

AND-Gate

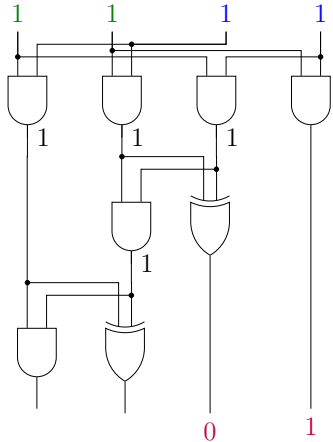
$f \wedge g = y$

f	g	y
0	0	0
0	1	0
1	0	0
1	1	1

XOR-Gate

$f \oplus g = y$

f	g	y
0	0	0
0	1	1
1	0	1
1	1	0



Integer Multiplication

AND-Gate

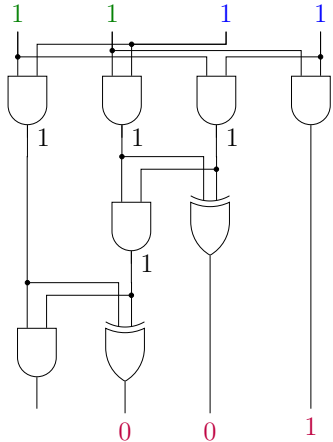
$f \wedge g = y$

f	g	y
0	0	0
0	1	0
1	0	0
1	1	1

XOR-Gate

$f \oplus g = y$

f	g	y
0	0	0
0	1	1
1	0	1
1	1	0



Integer Multiplication

AND-Gate

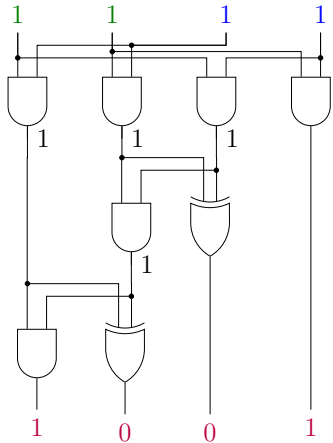
$f \wedge g = y$

f	g	y
0	0	0
0	1	0
1	0	0
1	1	1

XOR-Gate

$f \oplus g = y$

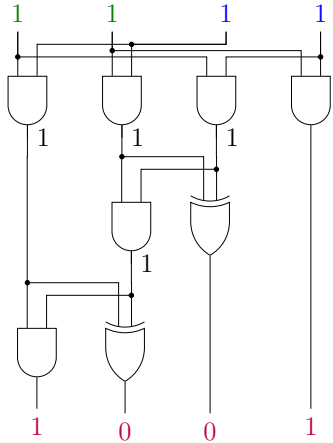
f	g	y
0	0	0
0	1	1
1	0	1
1	1	0



Is the circuit correct?

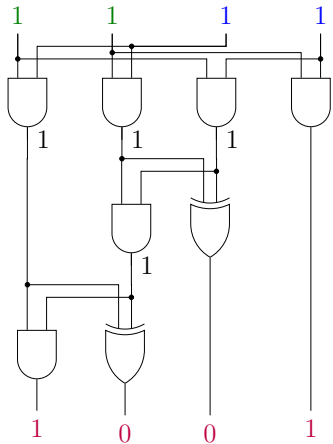
Multiplier circuit

$$
\begin{array}{cccc}
1 & 1 & \cdot & 1 & 1 \\
\hline
 & & & 1 & 1 \\
 & & 1 & 1 & \\
 & 1 & 1 & 0 & \\
\hline
1 & 0 & 0 & 1 &
\end{array}
$$



Multiplier circuit

Circuit seems correct!



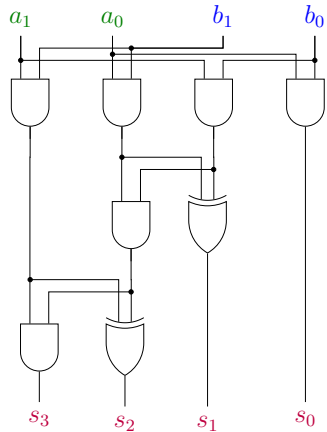
Is the circuit really correct?

Multiplier Circuits

Given: Gate-level multiplier for fixed bit-width.

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

$$(2a_1 + a_0) * (2b_1 + b_0) = 8s_3 + 4s_2 + 2s_1 + s_0?$$

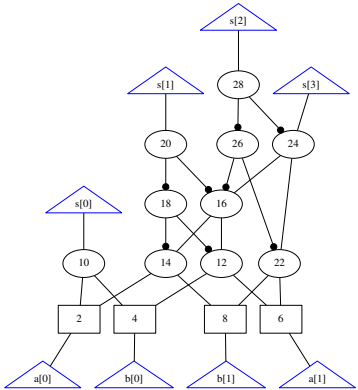


Multiplier Circuits

Given: Gate-level multiplier for fixed bit-width.

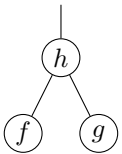
Question: For all possible $a_i, b_i \in \mathbb{B}$:

$$(2a_1 + a_0) * (2b_1 + b_0) = 8s_3 + 4s_2 + 2s_1 + s_0?$$

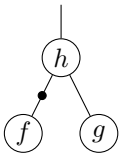


And-Inverter Graph

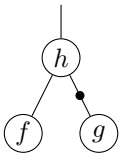
And-Inverter Graph



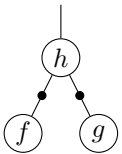
$h = f \wedge g$



$h = \neg f \wedge g$

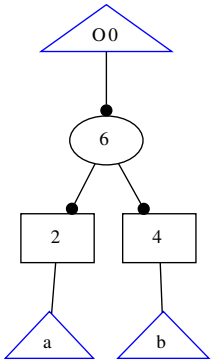


$h = f \wedge \neg g$

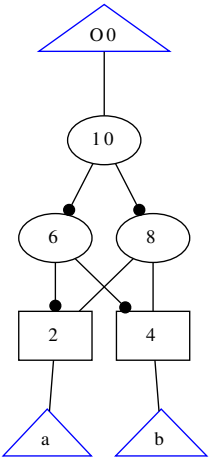


$h = \neg f \wedge \neg g$

And-Inverter Graph



$a \vee b \equiv \neg(\neg a \wedge \neg b)$



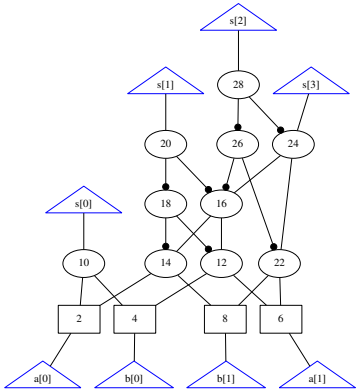
$a \otimes b \equiv \neg(\neg a \wedge \neg b) \wedge \neg(a \wedge b)$

Multiplier Circuits

Given: Gate-level multiplier for fixed bit-width.

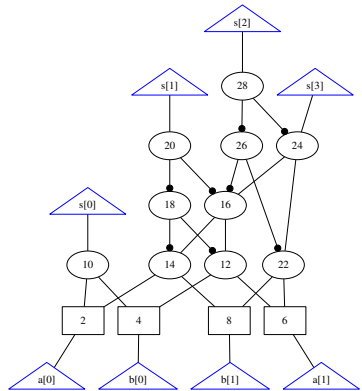
Question: For all possible $a_i, b_i \in \mathbb{B}$:

$$(2a_1 + a_0) * (2b_1 + b_0) = 8s_3 + 4s_2 + 2s_1 + s_0?$$



Simulation

$a_1 a_0$	·	$b_1 b_0$
00	·	00
00	·	01
00	·	10
00	·	11
01	·	00
01	·	01
01	·	10
01	·	11
10	·	00
10	·	01
10	·	10
10	·	11
11	·	00
11	·	01
11	·	10
11	·	11

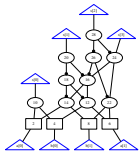


Simulation

Bit-width	Cases	
2	16	
3	64	
4	256	
5	1 024	
6	4 096	
7	16 384	
8	65 536	
⋮	⋮	
32	18 446 744 073 709 551 616	quintillion
⋮	⋮	
64	340 282 366 920 938 463 463 374 607 431 768 211 456	undecillion

Formal Verification

System



Specification

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

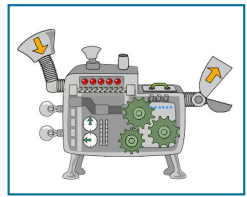
$(2a_1 + a_0) * (2b_1 + b_0) =$

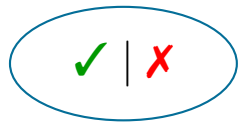
$8s_3 + 4s_2 + 2s_1 + s_0$?

Mathematical Model

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

Automated Decision Process





Formal Verification Techniques

Decision Diagrams

- First technique to detect Pentium bug

  [ChenBryant DAC'95]

- Requires knowledge of the layout

Formal Verification Techniques

Decision Diagrams

- First technique to detect Pentium bug
  [ChenBryant DAC'95]
- Requires knowledge of the layout

Theorem Proving

- Used in industry, e.g., ACL2
  [TemelSlobodovaHunt CAV'20]
- Automated on the RTL level
  [Temel TACAS'24]

Formal Verification Techniques

Decision Diagrams

- First technique to detect Pentium bug [ChenBryant DAC'95]
- Requires knowledge of the layout

Theorem Proving

- Used in industry, e.g., ACL2 [TemelSlobodovaHunt CAV'20]
- Automated on the RTL level [Temel TACAS'24]

Satisfiability Checking (SAT)

- SAT 2016: Exponential run-time of solvers [Biere SATComp'16]
- SAT 2024: Equivalence checking of structural similar circuits [BiereFallerFazekasFleuryFroleyksPollitt SATComp'24]

Formal Verification Techniques

Decision Diagrams

- First technique to detect Pentium bug [ChenBryant DAC'95]
- Requires knowledge of the layout

Satisfiability Checking (SAT)

- SAT 2016: Exponential run-time of solvers [Biere SATComp'16]
- SAT 2024: Equivalence checking of structural similar circuits [BiereFallerFazekasFleuryFroleyksPollitt SATComp'24]

Theorem Proving

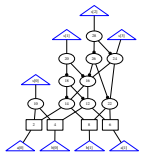
- Used in industry, e.g., ACL2 [TemelSlobodovaHunt CAV'20]
- Automated on the RTL level [Temel TACAS'24]

Algebraic Approach

- Seminal work: [LvKallaEnescu TCAD'13, CiesielskiYuBrownLiuRossi DAC'15] [SayedGroßeKühneSoekenDrechsler DATE'16]
- Polynomial encoding
- Works for non-trivial multiplier designs

Basic Idea of Algebraic Approach

Multiplier



Polynomials

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

Specification

$$\sum_{i=0}^{2n-1} 2^i s_i - (\sum_{i=0}^{n-1} 2^i a_i)(\sum_{i=0}^{n-1} 2^i b_i)$$

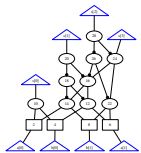
Implication

$= 0$ ✓

$\neq 0$ ✗

Basic Idea of Algebraic Approach

Multiplier



Polynomials

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

Specification

$$\sum_{i=0}^{2n-1} 2^i s_i - (\sum_{i=0}^{n-1} 2^i a_i)(\sum_{i=0}^{n-1} 2^i b_i)$$

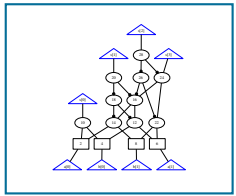
Ideal Membership

$= 0$ ✓

$\neq 0$ ✗

Basic Idea of Algebraic Approach

Multiplier



Polynomials

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

Specification

$$\sum_{i=0}^{2n-1} 2^i s_i - (\sum_{i=0}^{n-1} 2^i a_i)(\sum_{i=0}^{n-1} 2^i b_i)$$

Ideal Membership

$= 0$ ✓

$\neq 0$ ✗

Multiplier Specification

Unsigned Integers:

$$0 = \sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right) \quad \in \mathbb{Q}[X]$$

Multiplier Specification

Unsigned Integers:

$$\sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right) \quad \in \mathbb{Q}[X]$$

Multiplier Specification

Unsigned Integers:

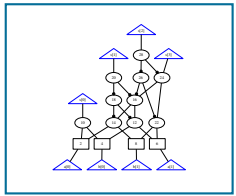
$$\sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right) \quad \in \mathbb{Q}[X]$$

Signed Integers:

$$-2^{2n-1}s_{2n-1} + \sum_{i=0}^{2n-2} 2^i s_i - \left(-2^{n-1}a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i\right)\left(-2^{n-1}b_{n-1} + \sum_{i=0}^{n-2} 2^i b_i\right) \quad \in \mathbb{Q}[X]$$

Basic Idea of Algebraic Approach

Multiplier



Polynomials

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

Specification

$$\sum_{i=0}^{2n-1} 2^i s_i - (\sum_{i=0}^{n-1} 2^i a_i)(\sum_{i=0}^{n-1} 2^i b_i)$$

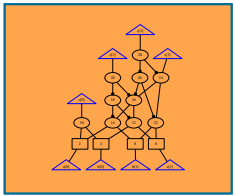
Ideal Membership

$= 0$ ✓

$\neq 0$ ✗

Basic Idea of Algebraic Approach

Multiplier



Polynomials

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

Specification

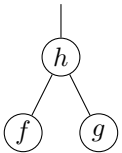
$$\sum_{i=0}^{2n-1} 2^i s_i - (\sum_{i=0}^{n-1} 2^i a_i)(\sum_{i=0}^{n-1} 2^i b_i)$$

Ideal Membership

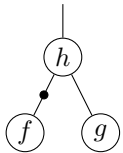
$= 0$ ✓

$\neq 0$ ✗

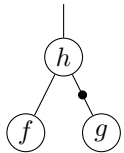
From AIGs to Polynomials



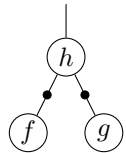
$h = f \wedge g$



$h = \neg f \wedge g$

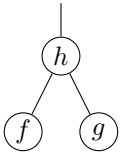


$h = f \wedge \neg g$



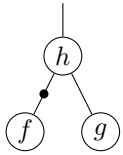
$h = \neg f \wedge \neg g$

From AIGs to Polynomials



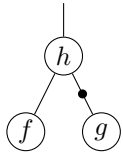
$h = f \wedge g$

f	g	h
0	0	0
0	1	0
1	0	0
1	1	1



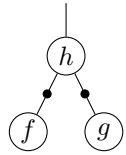
$h = \neg f \wedge g$

f	g	h
0	0	0
0	1	1
1	0	0
1	1	0



$h = f \wedge \neg g$

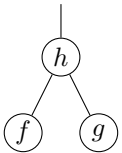
f	g	h
0	0	0
0	1	0
1	0	1
1	1	0



$h = \neg f \wedge \neg g$

f	g	h
0	0	1
0	1	0
1	0	0
1	1	0

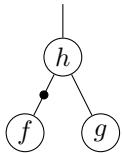
From AIGs to Polynomials



$h = f \wedge g$

f	g	h
0	0	0
0	1	0
1	0	0
1	1	1

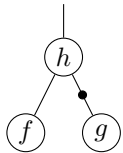
$h - fg$



$h = \neg f \wedge g$

f	g	h
0	0	0
0	1	1
1	0	0
1	1	0

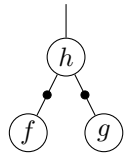
$h + fg - g$



$h = f \wedge \neg g$

f	g	h
0	0	0
0	1	0
1	0	1
1	1	0

$h + fg - f$



$h = \neg f \wedge \neg g$

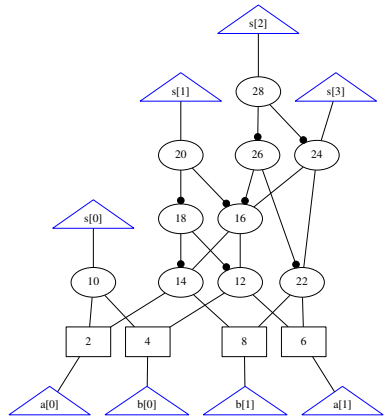
f	g	h
0	0	1
0	1	0
1	0	0
1	1	0

$h - fg + f + g - 1$

From AIGs to Polynomials

Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

$$s_3 - l_{24}$$
$$s_2 - l_{28}$$
$$s_1 - l_{20}$$
$$s_0 - l_{10}$$
$$l_{28} - l_{26}l_{24} + l_{26} + l_{24} - 1$$
$$l_{26} - l_{22}l_{16} + l_{22} + l_{16} - 1$$
$$l_{24} - l_{22}l_{16}$$
$$l_{22} - a_1b_1$$
$$l_{20} - l_{18}l_{16} + l_{18} + l_{16} - 1$$
$$l_{18} - l_{14}l_{12} + l_{14} + l_{12} - 1$$
$$l_{16} - l_{14}l_{12}$$
$$l_{14} - a_0b_1$$
$$l_{12} - a_1b_0$$
$$l_{10} - a_0b_0$$



From AIGs to Polynomials

Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

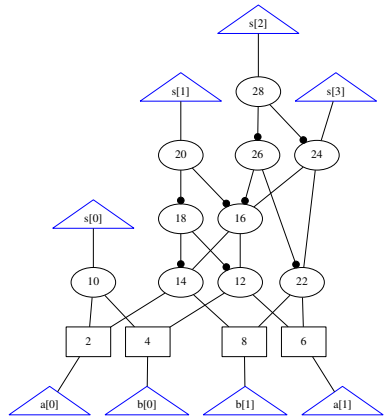
$$
\begin{array}{ll}
s_3 - l_{24} & l_{22} - a_1 b_1 \\
s_2 - l_{28} & l_{20} - l_{18} l_{16} + l_{18} + l_{16} - 1 \\
s_1 - l_{20} & l_{18} - l_{14} l_{12} + l_{14} + l_{12} - 1 \\
s_0 - l_{10} & l_{16} - l_{14} l_{12} \\
l_{28} - l_{26} l_{24} + l_{26} + l_{24} - 1 & l_{14} - a_0 b_1 \\
l_{26} - l_{22} l_{16} + l_{22} + l_{16} - 1 & l_{12} - a_1 b_0 \\
l_{24} - l_{22} l_{16} & l_{10} - a_0 b_0
\end{array}
$$

Boolean input constraints $B(C) \subseteq \mathbb{Q}[X]$.

$$
\begin{array}{ll}
a_1, a_0 \in \mathbb{B} & a_1^2 - a_1,\ a_0^2 - a_0, \\
b_1, b_0 \in \mathbb{B} & b_1^2 - b_1,\ b_0^2 - b_0
\end{array}
$$

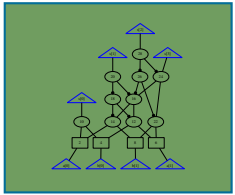
Specification $\mathcal{S}_n \in \mathbb{Q}[X]$.

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



Basic Idea of Algebraic Approach

Multiplier



Polynomials

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

Specification

$$\sum_{i=0}^{2n-1} 2^i s_i - (\sum_{i=0}^{n-1} 2^i a_i)(\sum_{i=0}^{n-1} 2^i b_i)$$

Ideal Membership

$= 0$ ✓

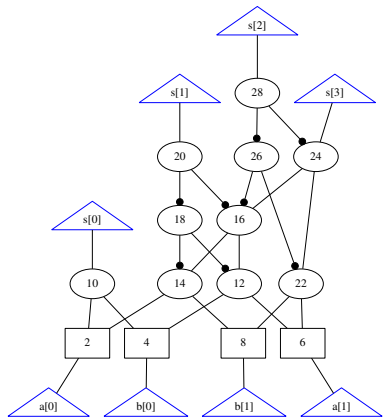
$\neq 0$ ✗

Ideals Associated to Circuits

[RitircBiereKauers FMCAD'17, KaufmannBiereKauers FMSD'20]

More circuit relations:

- $s_0 - a_0 b_0$ AND-gate
- $a_1^2 - a_1$ a_1 Boolean
- $l_{22}^2 - l_{22}$ l_{22} Boolean
- $l_{20} l_{16}$ XOR-AND constraint
- ...



Polynomial Circuit Constraints.

A polynomial $p \in \mathbb{Q}[X]$ is a polynomial circuit constraint (PCC) if for all

$$(a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}) \in \{0,1\}^{2n}$$

and resulting values $l_1, \ldots, l_k, s_0, \ldots, s_{2n-1}$ implied by the gates of the circuit C the substitution of these values into p gives zero.

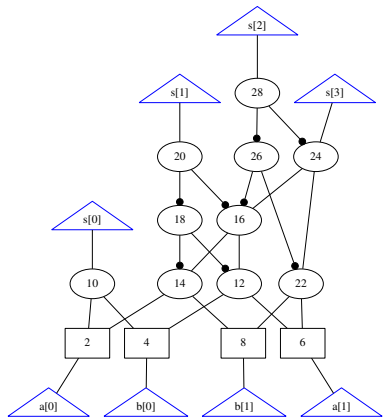
Ideals Associated to Circuits

[RitircBiereKauers FMCAD'17, KaufmannBiereKauers FMSD'20]

- The set of all PCCs for C is denoted by $I(C)$.
- $I(C)$ contains all relations of the circuit C.
- $I(C)$ is an ideal.

Multiplier. A circuit C is called a multiplier if

$$\sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right) \quad \in \quad I(C).$$



Ideal Membership Problem

[RitircBiereKauers FMCAD'17]

- We can deduce some elements of $I(C)$:
  - Gate polynomials $G(C)$
  - Boolean input constraints $B(C)$

Ideal Membership Problem

[RitircBiereKauers FMCAD'17]

- We can deduce some elements of $I(C)$:
  - Gate polynomials $G(C)$
  - Boolean input constraints $B(C)$
- Let $J(C) = \langle G(C) \cup B(C) \rangle$.
- Lexicographic term order: Reverse topological
  Output variable of a gate is greater than input variables [LvKallaEnescu TCAD'13].

Ideal Membership Problem

[RitircBiereKauers FMCAD'17]

- We can deduce some elements of $I(C)$:
  - Gate polynomials $G(C)$
  - Boolean input constraints $B(C)$
- Let $J(C) = \langle G(C) \cup B(C) \rangle$.
- Lexicographic term order: Reverse topological
  Output variable of a gate is greater than input variables [LvKallaEnescu TCAD'13].

Theorem

$G(C) \cup B(C)$ is a Gröbner basis for $J(C)$.

Proof idea: Application of Buchberger's Product criterion.

Soundness and Completeness

[RitircBiereKauers FMCAD'17]

Theorem

For all acyclic circuits C, we have $J(C) = I(C)$.

- $J(C) \subseteq I(C)$: soundness
- $I(C) \subseteq J(C)$: completeness

Verification Algorithm

Verification Algorithm

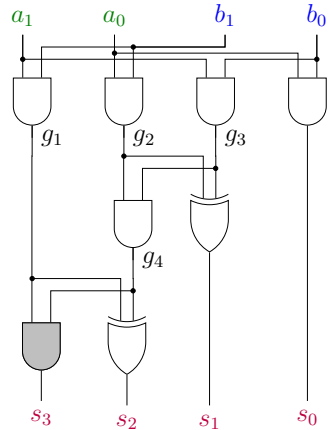
Reduce specification $\sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right)$ by elements of $G(C) \cup B(C)$

until no further reduction is possible, then C is a multiplier iff remainder is zero.

Verification

Gate polynomials $G(C)$.

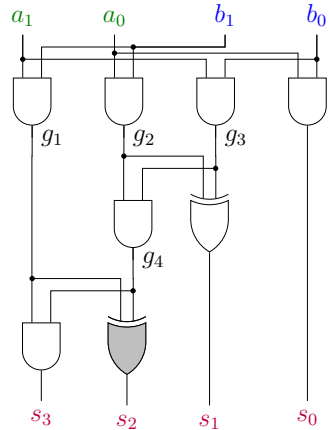
$$s_3 = g_1 \wedge g_4 \qquad -s_3 + g_4 g_1,$$



Verification

Gate polynomials $G(C)$.

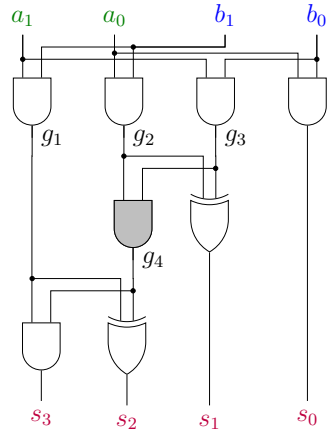
$$s_3 = g_1 \wedge g_4 \qquad -s_3 + g_4 g_1,$$
$$s_2 = g_1 \oplus g_4 \qquad -s_2 - 2g_4 g_1 + g_4 + g_1,$$



Verification

Gate polynomials $G(C)$.

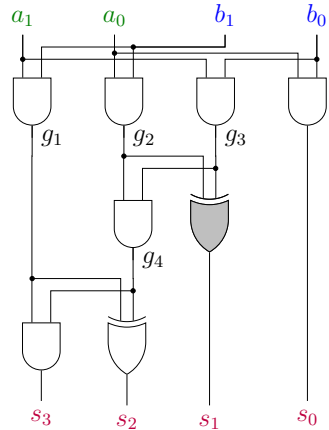
$$
\begin{aligned}
s_3 = g_1 \wedge g_4 \qquad & -s_3 + g_4 g_1, \\
s_2 = g_1 \oplus g_4 \qquad & -s_2 - 2g_4 g_1 + g_4 + g_1, \\
g_4 = g_2 \wedge g_3 \qquad & -g_4 + g_2 g_3,
\end{aligned}
$$



Verification

Gate polynomials $G(C)$.

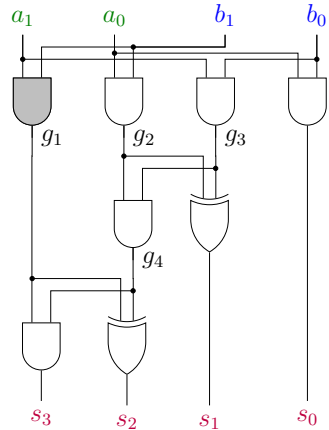
$$
\begin{aligned}
s_3 = g_1 \wedge g_4 \quad & -s_3 + g_4 g_1, \\
s_2 = g_1 \oplus g_4 \quad & -s_2 - 2g_4 g_1 + g_4 + g_1, \\
g_4 = g_2 \wedge g_3 \quad & -g_4 + g_2 g_3, \\
s_1 = g_2 \oplus g_3 \quad & -s_1 - 2g_2 g_3 + g_2 + g_3,
\end{aligned}
$$



Verification

Gate polynomials $G(C)$.

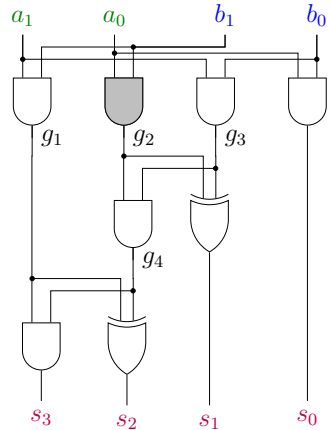
$$
\begin{aligned}
s_3 = g_1 \wedge g_4 \quad & -s_3 + g_4 g_1, \\
s_2 = g_1 \oplus g_4 \quad & -s_2 - 2g_4 g_1 + g_4 + g_1, \\
g_4 = g_2 \wedge g_3 \quad & -g_4 + g_2 g_3, \\
s_1 = g_2 \oplus g_3 \quad & -s_1 - 2g_2 g_3 + g_2 + g_3, \\
g_1 = a_1 \wedge b_1 \quad & -g_1 + a_1 b_1,
\end{aligned}
$$



Verification

Gate polynomials $G(C)$.

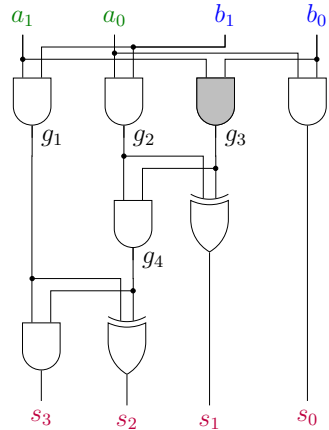
$$
\begin{aligned}
s_3 = g_1 \wedge g_4 \quad & -s_3 + g_4 g_1, \\
s_2 = g_1 \oplus g_4 \quad & -s_2 - 2g_4 g_1 + g_4 + g_1, \\
g_4 = g_2 \wedge g_3 \quad & -g_4 + g_2 g_3, \\
s_1 = g_2 \oplus g_3 \quad & -s_1 - 2g_2 g_3 + g_2 + g_3, \\
g_1 = a_1 \wedge b_1 \quad & -g_1 + a_1 b_1, \\
g_2 = a_0 \wedge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



Verification

Gate polynomials $G(C)$.

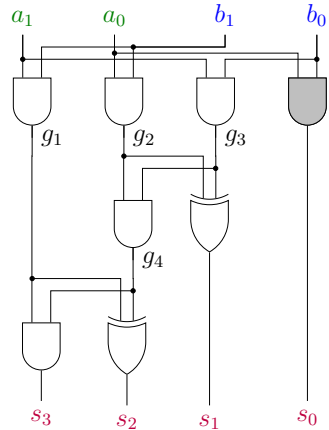
$$
\begin{aligned}
s_3 = g_1 \wedge g_4 \qquad & -s_3 + g_4 g_1, \\
s_2 = g_1 \oplus g_4 \qquad & -s_2 - 2g_4 g_1 + g_4 + g_1, \\
g_4 = g_2 \wedge g_3 \qquad & -g_4 + g_2 g_3, \\
s_1 = g_2 \oplus g_3 \qquad & -s_1 - 2g_2 g_3 + g_2 + g_3, \\
g_1 = a_1 \wedge b_1 \qquad & -g_1 + a_1 b_1, \\
g_2 = a_0 \wedge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



Verification

Gate polynomials $G(C)$.

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



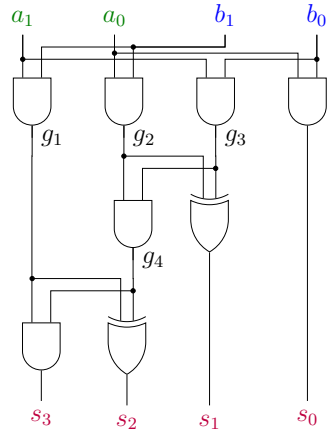
Verification

Gate polynomials $G(C)$.

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

Boolean input constraints $B(C)$.

$$
\begin{aligned}
a_1, a_0 \in \mathbb{B} \quad & a_1(1-a_1),\ a_0(1-a_0), \\
b_1, b_0 \in \mathbb{B} \quad & b_1(1-b_1),\ b_0(1-b_0)
\end{aligned}
$$



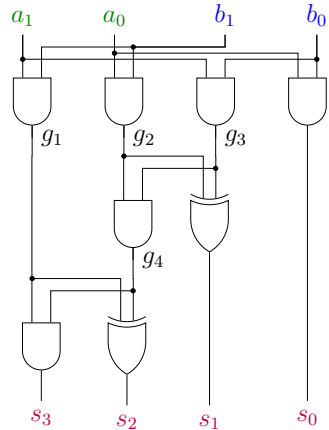
Verification

Gate polynomials $G(C)$.

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

Boolean input constraints $B(C)$.

$$
\begin{aligned}
a_1, a_0 \in \mathbb{B} \quad & -a_1^2 + a_1,\ -a_0^2 + a_0, \\
b_1, b_0 \in \mathbb{B} \quad & -b_1^2 + b_1,\ -b_0^2 + b_0
\end{aligned}
$$



Verification

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

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

Verification

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

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

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

Verification

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

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

Verification

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

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

Verification

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

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

Verification

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

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

Verification

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

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

Verification

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

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

Verification

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

$$
\begin{aligned}
8s_3 + 4s_2 + 2s_1 + s_0 - 4a_1b_1 - 2a_1b_0 - 2a_0b_1 - a_0b_0 \\
8g_1g_4 + 4s_2 + 2s_1 + s_0 - 4a_1b_1 - 2a_1b_0 - 2a_0b_1 - a_0b_0 \\
4g_4 + 4g_1 + 2s_1 + s_0 - 4a_1b_1 - 2a_1b_0 - 2a_0b_1 - a_0b_0 \\
4g_2g_3 + 4g_1 + 2s_1 + s_0 - 4a_1b_1 - 2a_1b_0 - 2a_0b_1 - a_0b_0 \\
4g_1 + 2g_3 + 2g_2 + s_0 - 4a_1b_1 - 2a_1b_0 - 2a_0b_1 - a_0b_0 \\
2g_3 + 2g_2 + s_0 - 2a_1b_0 - 2a_0b_1 - a_0b_0 \\
2g_3 + s_0 - 2a_1b_0 - a_0b_0 \\
s_0 - a_0b_0 \\
0
\end{aligned}
$$

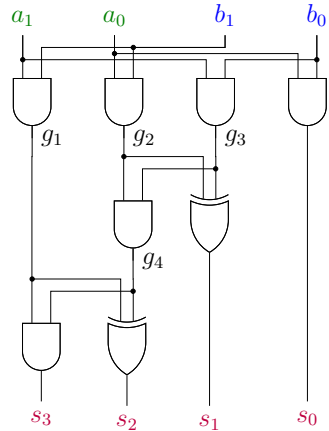
Faulty multiplier

Gate polynomials $G(C)$.

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

Boolean input constraints $B(C)$.

$$
\begin{aligned}
a_1, a_0 &\in \mathbb{B} && a_1(1-a_1),\ a_0(1-a_0), \\
b_1, b_0 &\in \mathbb{B} && b_1(1-b_1),\ b_0(1-b_0)
\end{aligned}
$$



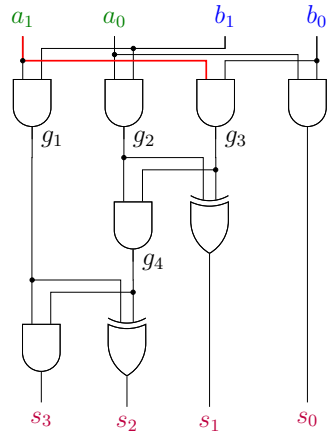
Faulty multiplier

Gate polynomials $G(C)$.

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

Boolean input constraints $B(C)$.

$$
\begin{aligned}
a_1, a_0 &\in \mathbb{B} && a_1(1-a_1),\ a_0(1-a_0), \\
b_1, b_0 &\in \mathbb{B} && b_1(1-b_1),\ b_0(1-b_0)
\end{aligned}
$$



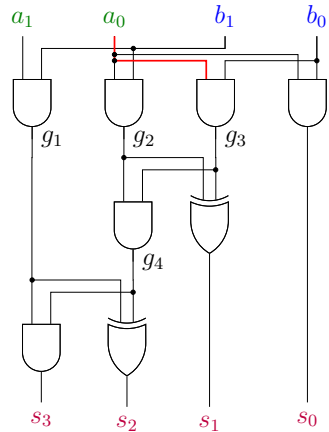
Faulty multiplier

Gate polynomials $G(C)$.

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

Boolean input constraints $B(C)$.

$$
\begin{aligned}
a_1, a_0 &\in \mathbb{B} && a_1(1-a_1),\ a_0(1-a_0), \\
b_1, b_0 &\in \mathbb{B} && b_1(1-b_1),\ b_0(1-b_0)
\end{aligned}
$$



Verification of a faulty multiplier

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

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

Verification of a faulty multiplier

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

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

Verification of a faulty multiplier

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

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

Verification of a faulty multiplier

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

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

Counter Example

Remainder: $-2a_1b_0 + 2a_0b_0 \neq 0$

Counter Example

Remainder: $-2a_1b_0 + 2a_0b_0 \neq 0$

$a_1 = 0, \quad a_0 = 1$
$b_1 = 0, \quad b_0 = 1$

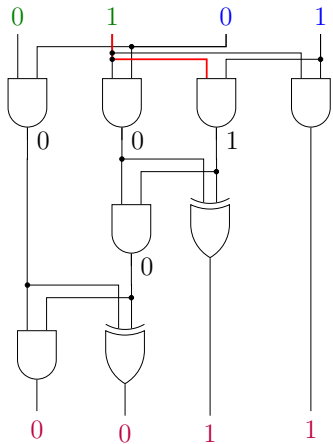
$01 \cdot 01 = 0001$

Counter Example

Remainder: $-2a_1b_0 + 2a_0b_0 \neq 0$

$a_1 = 0, \quad a_0 = 1$
$b_1 = 0, \quad b_0 = 1$

$01 \cdot 01 = 0001$



Verification Algorithm

Reduce specification $\sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right)$ by elements of $G(C) \cup B(C)$

until no further reduction is possible, then C is a multiplier iff remainder is zero.

Verification Algorithm

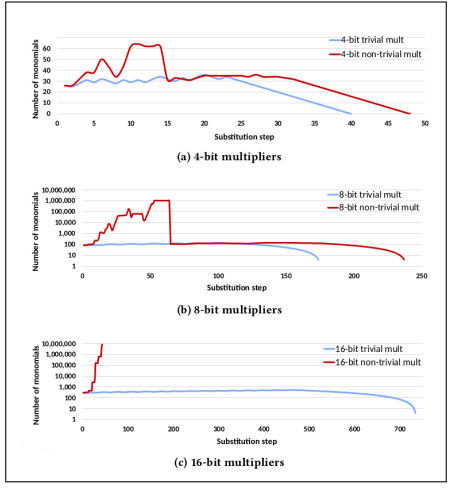
Reduce specification $\sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right)$ by elements of $G(C) \cup B(C)$ until no further reduction is possible, then C is a multiplier iff remainder is zero.

Computational Problems

- The number of monomials in the intermediate results blows-up.
- 8-bit multiplier cannot be verified within 20 minutes.

Verification Algorithm

[MahzoonGroßeDrechsler ICCAD'18]



Strategies

1. Encoding
   - Embedding different phases [KaufmannBeameBiereNordström DATE'22, KonradScholl FMCAD'24]
2. Preprocessing
   - Variable Elimination [MahzoonGroßeDrechsler DAC'19, RitircBiereKauers DATE'18]
3. Reduction
   - Incremental Algorithm [RitircBiereKauers FMCAD'17]
   - Dynamic Reduction Order [MahzoonGroßeScholIDrechsler DATE'20, KonradScholl FMCAD'24]
4. Tricky: OR Gates in final stage adder
   - Include SAT or BDDs [KaufmannBiereKauers FMCAD'19, DrechslerMahzoon ISEEIE'22]

Strategies

1. Encoding
   - Embedding different phases [KaufmannBeameBiereNordström DATE'22, KonradScholl FMCAD'24]
2. Preprocessing
   - Variable Elimination [MahzoonGroßeDrechsler DAC'19, RitircBiereKauers DATE'18]
3. Reduction
   - Incremental Algorithm [RitircBiereKauers FMCAD'17]
   - Dynamic Reduction Order [MahzoonGroßeSchollDrechsler DATE'20, KonradScholl FMCAD'24]
4. Tricky: OR Gates in final stage adder
   - Include SAT or BDDs [KaufmannBiereKauers FMCAD'19, DrechslerMahzoon ISEEIE'22]

All of these strategies rely on a **lexicographic term ordering**.

Change of Order[1]

	$\prec_{\text{lex}}$	$\prec_{\text{drl}}$
GB Computation	✅ Easy	⚠️ Hard
Spec Reduction	⚠️ Hard	✅ Easy

[1] D. Kaufmann and J. Berthomieu. Extracting Linear Relations from Gröbner Bases for Formal Verification of And-Inverter Graphs. Accepted at TACAS 2025. Preprint at https://arxiv.org/abs/2411.16348

Change of Order[1]

	$\prec_{\text{lex}}$	$\prec_{\text{drl}}$
GB Computation	✅ Easy	⚠️ Hard
Spec Reduction	⚠️ Hard	✅ Easy

If the specification polynomial is linear,

a Gröbner basis with respect to a
degree reverse lexicographic term ordering

contains linear polynomials that suffice
to derive correctness of the circuit.

[1] D. Kaufmann and J. Berthomieu. Extracting Linear Relations from Gröbner Bases for Formal Verification of And-Inverter Graphs. Accepted at TACAS 2025. Preprint at https://arxiv.org/abs/2411.16348

Theorem

Theorem

Let $p \in \mathbb{K}[X]$ with $\deg(p) = 1$, $I \subseteq \mathbb{K}[X]$ be an ideal. Let G be a Gröbner basis of I with respect to $\prec_{\mathrm{drl}}$ and let $G_1 = \{g \in G \mid \deg(g) \leq 1\}$. We have $p \in I$ if and only if $p \to_{G_1} 0$. In particular, $p = \alpha_1 g_1 + \cdots + \alpha_m g_m$ with $g_i \in G_1$, $\alpha_i \in \mathbb{K}$.

Linear Gröbner Basis Reduction Algorithm

Algorithm: Linear Gröbner basis reduction

Input : Circuit C in AIG format, Specification polynomial $\mathcal{S}$

Output: Determine whether C fulfills the specification

$G_{\text{init}} \leftarrow$ Gate-Polynomials$(C) \cup$ Boolean-Input-Polynomials(C);

$\mathcal{S}_{\text{lin}}, G_{\text{ext}} \leftarrow$ Linearize$(\mathcal{S})$;

$G_{\text{drl}} \leftarrow$ Compute-$\prec_{\text{drl}}$-Gröbner-Basis$(G_{\text{init}} \cup G_{\text{ext}})$

$G_1 \leftarrow \{g \mid g \in G_{\text{drl}} \wedge \deg(g) \leq 1\}$;

while $\text{lm}(\mathcal{S}_{\text{lin}}) \in \{\text{lm}(g) | g \in G_1\}$ **do**

  $p_{\text{lin}} \leftarrow g \in G_1$ such that $\text{lm}(g) = \text{lm}(\mathcal{S}_{\text{lin}})$;

  if $\nexists p_{\text{lin}}$ **then return** $\perp$;

  $\mathcal{S}_{\text{lin}} \leftarrow$ Linear-Reduce$(\mathcal{S}_{\text{lin}}, p_{\text{lin}})$;

return $\mathcal{S}_{\text{lin}} = 0$

Linear Gröbner Basis Reduction Algorithm

Algorithm: Linear Gröbner basis reduction

Input : Circuit C in AIG format, Specification polynomial $\mathcal{S}$

Output: Determine whether C fulfills the specification

$G_{\text{init}} \leftarrow$ Gate-Polynomials$(C) \cup$ Boolean-Input-Polynomials(C);

$\mathcal{S}_{\text{lin}}, G_{\text{ext}} \leftarrow$ Linearize$(\mathcal{S})$;

$G_{\text{drl}} \leftarrow$ Compute-$\prec_{\text{drl}}$-Gröbner-Basis$(G_{\text{init}} \cup G_{\text{ext}})$

$G_1 \leftarrow \{g \mid g \in G_{\text{drl}} \wedge \deg(g) \leq 1\}$;

while $\text{lm}(\mathcal{S}_{\text{lin}}) \in \{\text{lm}(g) | g \in G_1\}$ **do**

  $p_{\text{lin}} \leftarrow g \in G_1$ such that $\text{lm}(g) = \text{lm}(\mathcal{S}_{\text{lin}})$;

  if $\nexists p_{\text{lin}}$ **then return** $\perp$;

  $\mathcal{S}_{\text{lin}} \leftarrow$ Linear-Reduce$(\mathcal{S}_{\text{lin}}, p_{\text{lin}})$;

return $\mathcal{S}_{\text{lin}} = 0$

Linear Gröbner Basis Reduction Algorithm

Algorithm: Linear Gröbner basis reduction

Input : Circuit C in AIG format, Specification polynomial $\mathcal{S}$

Output: Determine whether C fulfills the specification

$G_{\text{init}} \leftarrow$ Gate-Polynomials$(C) \cup$ Boolean-Input-Polynomials(C);

$\mathcal{S}_{\text{lin}}, G_{\text{ext}} \leftarrow$ Linearize$(\mathcal{S})$;

$G_{\text{drl}} \leftarrow$ Compute-$\prec_{\text{drl}}$-Gröbner-Basis$(G_{\text{init}} \cup G_{\text{ext}})$

$G_1 \leftarrow \{g \mid g \in G_{\text{drl}} \wedge \deg(g) \leq 1\}$;

while $\text{lm}(\mathcal{S}_{\text{lin}}) \in \{\text{lm}(g) | g \in G_1\}$ **do**

  $p_{\text{lin}} \leftarrow g \in G_1$ such that $\text{lm}(g) = \text{lm}(\mathcal{S}_{\text{lin}})$;

  if $\nexists p_{\text{lin}}$ **then return** $\perp$;

  $\mathcal{S}_{\text{lin}} \leftarrow$ Linear-Reduce$(\mathcal{S}_{\text{lin}}, p_{\text{lin}})$;

return $\mathcal{S}_{\text{lin}} = 0$

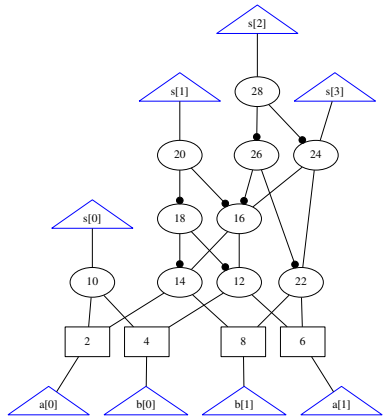
$\mathcal{S}_{\text{lin}}, G_{\text{ext}} \leftarrow$ **Linearize($\mathcal{S}$)**

Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

$$-s_3 + l_{24}$$
$$-s_2 + l_{28}$$
$$-s_1 + l_{20}$$
$$-s_0 + l_{10}$$
$$-l_{28} + l_{26}l_{24} - l_{26} - l_{24} + 1$$
$$-l_{26} + l_{22}l_{16} - l_{22} - l_{16} + 1$$
$$-l_{24} + l_{22}l_{16}$$
$$-l_{22} + a_1b_1$$
$$-l_{20} + l_{18}l_{16} - l_{18} - l_{16} + 1$$
$$-l_{18} + l_{14}l_{12} - l_{14} - l_{12} + 1$$
$$-l_{16} + l_{14}l_{12}$$
$$-l_{14} + a_0b_1$$
$$-l_{12} + a_1b_0$$
$$-l_{10} + a_0b_0$$

Specification $\mathcal{S} \in \mathbb{Q}[X]$.

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4b_1a_1 - 2b_1a_0 - 2b_0a_1 - b_0a_0$$



$\mathcal{S}_{\text{lin}}, G_{\text{ext}} \leftarrow$ **Linearize($\mathcal{S}$)**

Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

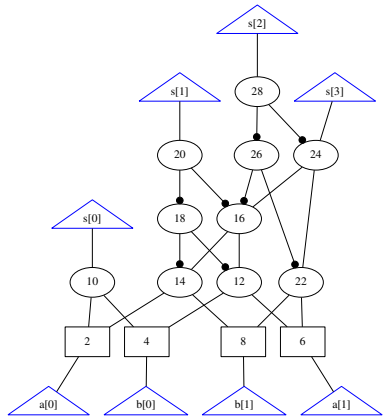
$$
\begin{array}{ll}
-s_3 + l_{24} & -l_{22} + a_1 b_1 \\
-s_2 + l_{28} & -l_{20} + l_{18} l_{16} - l_{18} - l_{16} + 1 \\
-s_1 + l_{20} & -l_{18} + l_{14} l_{12} - l_{14} - l_{12} + 1 \\
-s_0 + l_{10} & -l_{16} + l_{14} l_{12} \\
-l_{28} + l_{26} l_{24} - l_{26} - l_{24} + 1 & -l_{14} + a_0 b_1 \\
-l_{26} + l_{22} l_{16} - l_{22} - l_{16} + 1 & -l_{12} + a_1 b_0 \\
-l_{24} + l_{22} l_{16} & -l_{10} + a_0 b_0
\end{array}
$$

Extension polynomials $G_{\text{ext}} \subseteq \mathbb{Q}[X]$.

$$
\begin{array}{ll}
-t_{11} + a_1 b_1 & -t_{01} + a_0 b_1 \\
-t_{10} + a_1 b_0 & -t_{00} + a_0 b_0
\end{array}
$$

Linear Specification $\mathcal{S}_{\text{lin}} \in \mathbb{Q}[X]$.

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4t_{11} - 2t_{10} - 2t_{01} - t_{00}$$



$\mathcal{S}_{\text{lin}}, G_{\text{ext}} \leftarrow$ **Linearize($\mathcal{S}$)**

Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

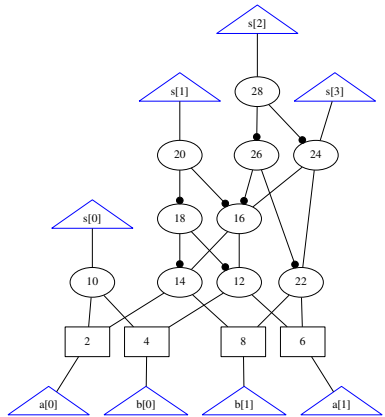
$$-s_3 + l_{24}$$
$$-s_2 + l_{28}$$
$$-s_1 + l_{20}$$
$$-s_0 + l_{10}$$
$$-l_{28} + l_{26}l_{24} - l_{26} - l_{24} + 1$$
$$-l_{26} + l_{22}l_{16} - l_{22} - l_{16} + 1$$
$$-l_{24} + l_{22}l_{16}$$
$$-l_{22} + a_1b_1$$
$$-l_{20} + l_{18}l_{16} - l_{18} - l_{16} + 1$$
$$-l_{18} + l_{14}l_{12} - l_{14} - l_{12} + 1$$
$$-l_{16} + l_{14}l_{12}$$
$$-l_{14} + a_0b_1$$
$$-l_{12} + a_1b_0$$
$$-l_{10} + a_0b_0$$

Extension polynomials $G_{\text{ext}} \subseteq \mathbb{Q}[X]$.

$$-t_{11} + a_1b_1 \qquad -t_{01} + a_0b_1$$
$$-t_{10} + a_1b_0 \qquad -t_{00} + a_0b_0$$

Linear Specification $\mathcal{S}_{\text{lin}} \in \mathbb{Q}[X]$.

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4t_{11} - 2t_{10} - 2t_{01} - t_{00}$$



$\mathcal{S}_{\text{lin}}, G_{\text{ext}} \leftarrow$ **Linearize($\mathcal{S}$)**

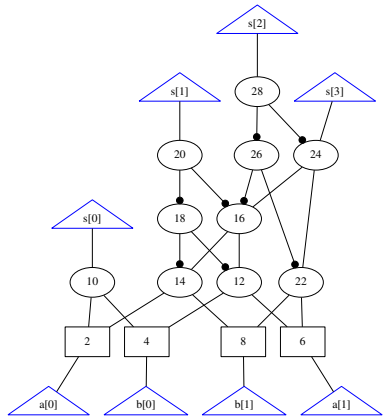
Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

$$
\begin{array}{ll}
-s_3 + l_{24} & -l_{22} + a_1 b_1 \\
-s_2 + l_{28} & -l_{20} + l_{18} l_{16} - l_{18} - l_{16} + 1 \\
-s_1 + l_{20} & -l_{18} + l_{14} l_{12} - l_{14} - l_{12} + 1 \\
-s_0 + l_{10} & -l_{16} + l_{14} l_{12} \\
-l_{28} + l_{26} l_{24} - l_{26} - l_{24} + 1 & -l_{14} + a_0 b_1 \\
-l_{26} + l_{22} l_{16} - l_{22} - l_{16} + 1 & -l_{12} + a_1 b_0 \\
-l_{24} + l_{22} l_{16} & -l_{10} + a_0 b_0
\end{array}
$$

Extension polynomials $G_{\text{ext}} \subseteq \mathbb{Q}[X]$.

Linear Specification $\mathcal{S}_{\text{lin}} \in \mathbb{Q}[X]$.

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$



Linear Gröbner Basis Reduction Algorithm

Algorithm: Linear Gröbner basis reduction

Input : Circuit C in AIG format, Specification polynomial $\mathcal{S}$

Output: Determine whether C fulfills the specification

$G_{\text{init}} \leftarrow$ Gate-Polynomials$(C) \cup$ Boolean-Input-Polynomials(C);

$\mathcal{S}_{\text{lin}}, G_{\text{ext}} \leftarrow$ Linearize$(\mathcal{S})$;

$G_{\text{drl}} \leftarrow$ Compute-$\prec_{\text{drl}}$-Gröbner-Basis$(G_{\text{init}} \cup G_{\text{ext}})$

$G_1 \leftarrow \{g \mid g \in G_{\text{drl}} \wedge \deg(g) \leq 1\}$;

while $\text{lm}(\mathcal{S}_{\text{lin}}) \in \{\text{lm}(g) | g \in G_1\}$ **do**

  $p_{\text{lin}} \leftarrow g \in G_1$ such that $\text{lm}(g) = \text{lm}(\mathcal{S}_{\text{lin}})$;

  if $\nexists p_{\text{lin}}$ **then return** $\perp$;

  $\mathcal{S}_{\text{lin}} \leftarrow$ Linear-Reduce$(\mathcal{S}_{\text{lin}}, p_{\text{lin}})$;

return $\mathcal{S}_{\text{lin}} = 0$

Linear Gröbner Basis Reduction Algorithm

Algorithm: Linear Gröbner basis reduction

Input : Circuit C in AIG format, Specification polynomial $\mathcal{S}$

Output: Determine whether C fulfills the specification

$G_{\text{init}} \leftarrow$ Gate-Polynomials$(C) \cup$ Boolean-Input-Polynomials(C);

$\mathcal{S}_{\text{lin}}, G_{\text{ext}} \leftarrow$ Linearize$(\mathcal{S})$;

$G_{\text{drl}} \leftarrow$ Compute-$\prec_{\text{drl}}$-Gröbner-Basis$(G_{\text{init}} \cup G_{\text{ext}})$;

$G_1 \leftarrow \{g \mid g \in G_{\text{drl}} \wedge \deg(g) \leq 1\}$;

while $\text{lm}(\mathcal{S}_{\text{lin}}) \in \{\text{lm}(g) | g \in G_1\}$ **do**

  $p_{\text{lin}} \leftarrow g \in G_1$ such that $\text{lm}(g) = \text{lm}(\mathcal{S}_{\text{lin}})$;

  if $\nexists p_{\text{lin}}$ **then return** $\perp$;

  $\mathcal{S}_{\text{lin}} \leftarrow$ Linear-Reduce$(\mathcal{S}_{\text{lin}}, p_{\text{lin}})$;

return $\mathcal{S}_{\text{lin}} = 0$

Linear Gröbner Basis Reduction Algorithm

Algorithm: Linear Gröbner basis reduction

Input : Circuit C in AIG format, Specification polynomial $\mathcal{S}$

Output: Determine whether C fulfills the specification

$G_{\text{init}} \leftarrow$ Gate-Polynomials$(C) \cup$ Boolean-Input-Polynomials(C);

$\mathcal{S}_{\text{lin}}, G_{\text{ext}} \leftarrow$ Linearize$(\mathcal{S})$;

$G_{\text{drl}} \leftarrow$ Compute-$\prec_{\text{drl}}$-Gröbner-Basis$(G_{\text{init}} \cup G_{\text{ext}})$; `// Expensive!!!`

$G_1 \leftarrow \{g \mid g \in G_{\text{drl}} \wedge \deg(g) \leq 1\}$;

while $\text{lm}(\mathcal{S}_{\text{lin}}) \in \{\text{lm}(g) | g \in G_1\}$ **do**

  $p_{\text{lin}} \leftarrow g \in G_1$ such that $\text{lm}(g) = \text{lm}(\mathcal{S}_{\text{lin}})$;

  if $\nexists p_{\text{lin}}$ **then return** $\perp$;

  $\mathcal{S}_{\text{lin}} \leftarrow$ Linear-Reduce$(\mathcal{S}_{\text{lin}}, p_{\text{lin}})$;

return $\mathcal{S}_{\text{lin}} = 0$

Linear Gröbner Basis Reduction Algorithm

Algorithm: Linear Gröbner basis reduction

Input : Circuit C in AIG format, Specification polynomial $\mathcal{S}$

Output: Determine whether C fulfills the specification

$G_{\text{init}} \leftarrow$ Gate-Polynomials$(C) \cup$ Boolean-Input-Polynomials(C);

$\mathcal{S}_{\text{lin}}, G_{\text{ext}} \leftarrow$ Linearize$(\mathcal{S})$;

Preprocessing(G_{ext});

while $\text{lm}(\mathcal{S}_{\text{lin}}) \in \{\text{lm}(g) | g \in G\}$ **do**

  $p \leftarrow g \in G$ such that $\text{lm}(g) = \text{lm}(\mathcal{S}_{\text{lin}})$;

  $p_{\text{lin}} \leftarrow$ Linearize-Single-Polynomial(p, G) ; `// On-the-fly`

  if $p_{\text{lin}} = 0$ **then return** $\bot$;

  $\mathcal{S}_{\text{lin}} \leftarrow$ Linear-Reduce$(\mathcal{S}_{\text{lin}}, p_{\text{lin}})$;

return $\mathcal{S}_{\text{lin}} = 0$

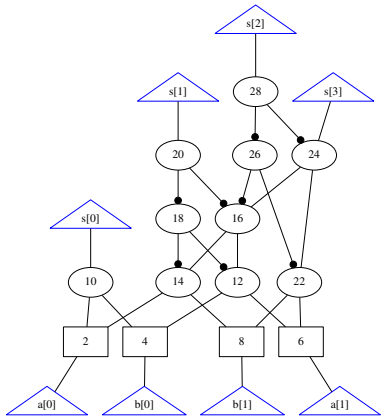
On-the-fly Linearization

Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

$$
\begin{array}{ll}
-s_3 + l_{24} & -l_{22} + a_1 b_1 \\
-s_2 + l_{28} & -l_{20} + l_{18} l_{16} - l_{18} - l_{16} + 1 \\
-s_1 + l_{20} & -l_{18} + l_{14} l_{12} - l_{14} - l_{12} + 1 \\
-s_0 + l_{10} & -l_{16} + l_{14} l_{12} \\
-l_{28} + l_{26} l_{24} - l_{26} - l_{24} + 1 & -l_{14} + a_0 b_1 \\
-l_{26} + l_{22} l_{16} - l_{22} - l_{16} + 1 & -l_{12} + a_1 b_0 \\
-l_{24} + l_{22} l_{16} & -l_{10} + a_0 b_0
\end{array}
$$

Specification $\mathcal{S}_n \in \mathbb{Q}[X]$.

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$



On-the-fly Linearization

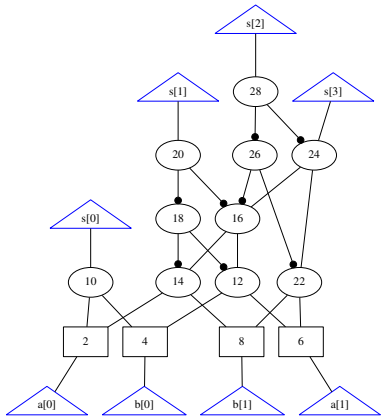
Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

$$
\begin{array}{ll}
-s_3 + l_{24} & -l_{22} + a_1 b_1 \\
-s_2 + l_{28} & -l_{20} + l_{18} l_{16} - l_{18} - l_{16} + 1 \\
-s_1 + l_{20} & -l_{18} + l_{14} l_{12} - l_{14} - l_{12} + 1 \\
-s_0 + l_{10} & -l_{16} + l_{14} l_{12} \\
-l_{28} + l_{26} l_{24} - l_{26} - l_{24} + 1 & -l_{14} + a_0 b_1 \\
-l_{26} + l_{22} l_{16} - l_{22} - l_{16} + 1 & -l_{12} + a_1 b_0 \\
-l_{24} + l_{22} l_{16} & -l_{10} + a_0 b_0
\end{array}
$$

Specification $\mathcal{S}_n \in \mathbb{Q}[X]$.

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$

$$4s_2 + 2s_1 + s_0 + 8l_{24} - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$



On-the-fly Linearization

Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

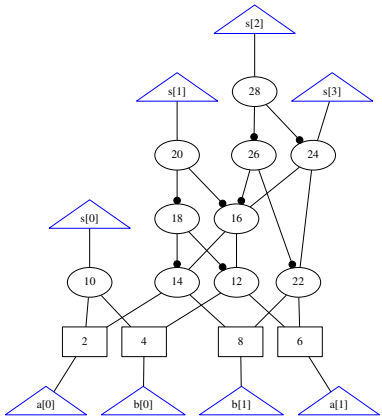
$$
\begin{array}{ll}
-s_3 + l_{24} & -l_{22} + a_1 b_1 \\
-s_2 + l_{28} & -l_{20} + l_{18} l_{16} - l_{18} - l_{16} + 1 \\
-s_1 + l_{20} & -l_{18} + l_{14} l_{12} - l_{14} - l_{12} + 1 \\
-s_0 + l_{10} & -l_{16} + l_{14} l_{12} \\
-l_{28} + l_{26} l_{24} - l_{26} - l_{24} + 1 & -l_{14} + a_0 b_1 \\
-l_{26} + l_{22} l_{16} - l_{22} - l_{16} + 1 & -l_{12} + a_1 b_0 \\
-l_{24} + l_{22} l_{16} & -l_{10} + a_0 b_0
\end{array}
$$

Specification $\mathcal{S}_n \in \mathbb{Q}[X]$.

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$

$$4s_2 + 2s_1 + s_0 + 8l_{24} - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$

$$4l_{28} + 8l_{24} - 4l_{22} + 2l_{20} - 2l_{14} - 2l_{12}$$



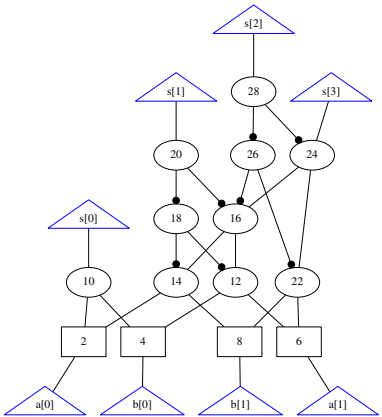
On-the-fly Linearization

Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

$$-s_3 + l_{24}$$
$$-s_2 + l_{28}$$
$$-s_1 + l_{20}$$
$$-s_0 + l_{10}$$
$$\boxed{-l_{28} + l_{26}l_{24} - l_{26} - l_{24} + 1}$$
$$-l_{26} + l_{22}l_{16} - l_{22} - l_{16} + 1$$
$$-l_{24} + l_{22}l_{16}$$
$$-l_{22} + a_1b_1$$
$$-l_{20} + l_{18}l_{16} - l_{18} - l_{16} + 1$$
$$-l_{18} + l_{14}l_{12} - l_{14} - l_{12} + 1$$
$$-l_{16} + l_{14}l_{12}$$
$$-l_{14} + a_0b_1$$
$$-l_{12} + a_1b_0$$
$$-l_{10} + a_0b_0$$

Specification $\mathcal{S}_n \in \mathbb{Q}[X]$.

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$
$$4s_2 + 2s_1 + s_0 + 8l_{24} - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$
$$4l_{28} + 8l_{24} - 4l_{22} + 2l_{20} - 2l_{14} - 2l_{12}$$



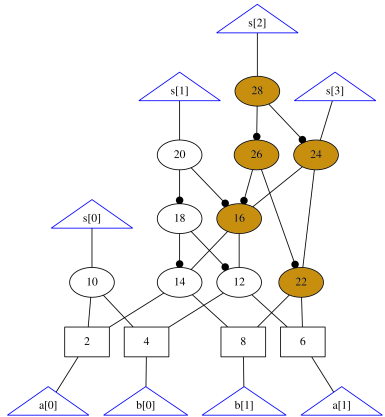
On-the-fly Linearization

Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

$$-s_3 + l_{24}$$
$$-s_2 + l_{28}$$
$$-s_1 + l_{20}$$
$$-s_0 + l_{10}$$
$$\boxed{-l_{28} + l_{26}l_{24} - l_{26} - l_{24} + 1}$$
$$-l_{26} + l_{22}l_{16} - l_{22} - l_{16} + 1$$
$$-l_{24} + l_{22}l_{16}$$
$$-l_{22} + a_1b_1$$
$$-l_{20} + l_{18}l_{16} - l_{18} - l_{16} + 1$$
$$-l_{18} + l_{14}l_{12} - l_{14} - l_{12} + 1$$
$$-l_{16} + l_{14}l_{12}$$
$$-l_{14} + a_0b_1$$
$$-l_{12} + a_1b_0$$
$$-l_{10} + a_0b_0$$

Specification $\mathcal{S}_n \in \mathbb{Q}[X]$.

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$
$$4s_2 + 2s_1 + s_0 + 8l_{24} - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$
$$4l_{28} + 8l_{24} - 4l_{22} + 2l_{20} - 2l_{14} - 2l_{12}$$



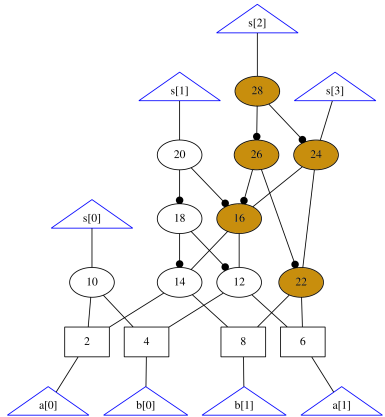
On-the-fly Linearization

Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

$$-s_3 + l_{24}$$
$$-s_2 + l_{28}$$
$$-s_1 + l_{20}$$
$$-s_0 + l_{10}$$
$$-l_{28} - l_{26} - l_{24} + 1$$
$$-l_{26} + l_{24} - l_{22} - l_{16} + 1$$
$$-l_{24} + l_{22} l_{16}$$
$$-l_{22} + a_1 b_1$$
$$-l_{20} + l_{18} l_{16} - l_{18} - l_{16} + 1$$
$$-l_{18} + l_{14} l_{12} - l_{14} - l_{12} + 1$$
$$-l_{16} + l_{14} l_{12}$$
$$-l_{14} + a_0 b_1$$
$$-l_{12} + a_1 b_0$$
$$-l_{10} + a_0 b_0$$

Specification $\mathcal{S}_n \in \mathbb{Q}[X]$.

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$
$$4s_2 + 2s_1 + s_0 + 8l_{24} - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$
$$4l_{28} + 8l_{24} - 4l_{22} + 2l_{20} - 2l_{14} - 2l_{12}$$



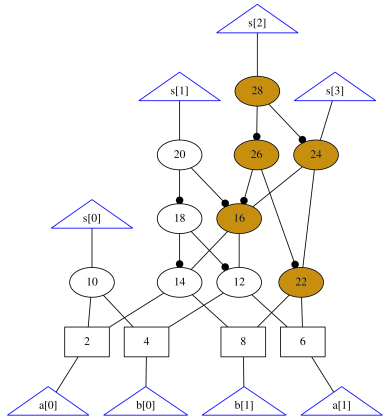
On-the-fly Linearization

Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

$$-s_3 + l_{24}$$
$$-s_2 + l_{28}$$
$$-s_1 + l_{20}$$
$$-s_0 + l_{10}$$
$$-l_{28} - l_{26} - l_{24} + 1$$
$$-l_{26} + l_{24} - l_{22} - l_{16} + 1$$
$$-l_{24} + l_{22} l_{16}$$
$$-l_{22} + a_1 b_1$$
$$-l_{20} + l_{18} l_{16} - l_{18} - l_{16} + 1$$
$$-l_{18} + l_{14} l_{12} - l_{14} - l_{12} + 1$$
$$-l_{16} + l_{14} l_{12}$$
$$-l_{14} + a_0 b_1$$
$$-l_{12} + a_1 b_0$$
$$-l_{10} + a_0 b_0$$

Specification $\mathcal{S}_n \in \mathbb{Q}[X]$.

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$
$$4s_2 + 2s_1 + s_0 + 8l_{24} - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$
$$4l_{28} + 8l_{24} - 4l_{22} + 2l_{20} - 2l_{14} - 2l_{12}$$
$$- 4l_{26} + 4l_{24} - 4l_{22} + 2l_{20} - 2l_{14} - 2l_{12} + 4$$



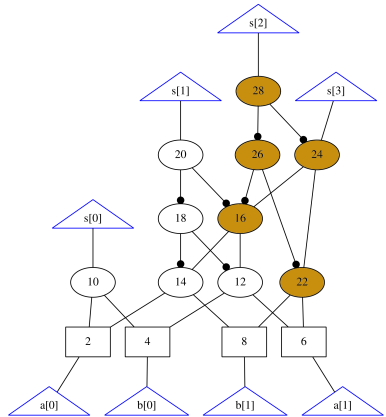
On-the-fly Linearization

Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

$$
\begin{aligned}
&-s_3 + l_{24} && -l_{22} + a_1 b_1 \\
&-s_2 + l_{28} && -l_{20} + l_{18} l_{16} - l_{18} - l_{16} + 1 \\
&-s_1 + l_{20} && -l_{18} + l_{14} l_{12} - l_{14} - l_{12} + 1 \\
&-s_0 + l_{10} && -l_{16} + l_{14} l_{12} \\
&-l_{28} - l_{26} - l_{24} + 1 && -l_{14} + a_0 b_1 \\
&-l_{26} + l_{24} - l_{22} - l_{16} + 1 && -l_{12} + a_1 b_0 \\
&-l_{24} + l_{22} l_{16} && -l_{10} + a_0 b_0
\end{aligned}
$$

Specification $\mathcal{S}_n \in \mathbb{Q}[X]$.

$$
\begin{aligned}
8s_3 + 4s_2 + 2s_1 + s_0 - 4l_{22} - 2l_{14} - 2l_{12} - l_{10} \\
4s_2 + 2s_1 + s_0 + 8l_{24} - 4l_{22} - 2l_{14} - 2l_{12} - l_{10} \\
4l_{28} + 8l_{24} - 4l_{22} + 2l_{20} - 2l_{14} - 2l_{12} \\
- 4l_{26} + 4l_{24} - 4l_{22} + 2l_{20} - 2l_{14} - 2l_{12} + 4 \\
2l_{20} + 4l_{16} - 2l_{14} - 2l_{12}
\end{aligned}
$$



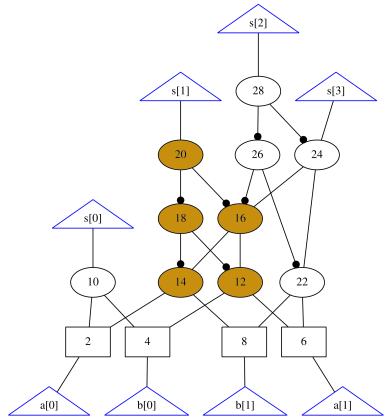
On-the-fly Linearization

Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

$$-s_3 + l_{24}$$
$$-s_2 + l_{28}$$
$$-s_1 + l_{20}$$
$$-s_0 + l_{10}$$
$$-l_{28} - l_{26} - l_{24} + 1$$
$$-l_{26} + l_{24} - l_{22} - l_{16} + 1$$
$$-l_{24} + l_{22} l_{16}$$
$$-l_{22} + a_1 b_1$$
$$-l_{20} + l_{18} l_{16} - l_{18} - l_{16} + 1$$
$$-l_{18} + l_{14} l_{12} - l_{14} - l_{12} + 1$$
$$-l_{16} + l_{14} l_{12}$$
$$-l_{14} + a_0 b_1$$
$$-l_{12} + a_1 b_0$$
$$-l_{10} + a_0 b_0$$

Specification $\mathcal{S}_n \in \mathbb{Q}[X]$.

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$
$$4s_2 + 2s_1 + s_0 + 8l_{24} - 4l_{22} - 2l_{14} - 2l_{12} - l_{10}$$
$$4l_{28} + 8l_{24} - 4l_{22} + 2l_{20} - 2l_{14} - 2l_{12}$$
$$-4l_{26} + 4l_{24} - 4l_{22} + 2l_{20} - 2l_{14} - 2l_{12} + 4$$
$$2l_{20} + 4l_{16} - 2l_{14} - 2l_{12}$$



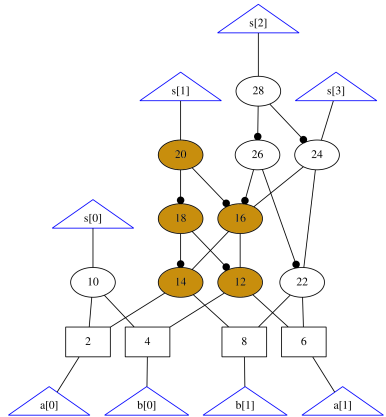
On-the-fly Linearization

Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

$$
\begin{aligned}
&-s_3 + l_{24} && -l_{22} + a_1 b_1 \\
&-s_2 + l_{28} && -l_{20} - l_{18} - l_{16} + 1 \\
&-s_1 + l_{20} && -l_{18} + l_{16} - l_{14} - l_{12} + 1 \\
&-s_0 + l_{10} && -l_{16} + l_{14} l_{12} \\
&-l_{28} - l_{26} - l_{24} + 1 && -l_{14} + a_0 b_1 \\
&-l_{26} + l_{24} - l_{22} - l_{16} + 1 && -l_{12} + a_1 b_0 \\
&-l_{24} + l_{22} l_{16} && -l_{10} + a_0 b_0
\end{aligned}
$$

Specification $\mathcal{S}_n \in \mathbb{Q}[X]$.

$$
\begin{aligned}
8s_3 + 4s_2 + 2s_1 + s_0 - 4l_{22} - 2l_{14} - 2l_{12} - l_{10} \\
4s_2 + 2s_1 + s_0 + 8l_{24} - 4l_{22} - 2l_{14} - 2l_{12} - l_{10} \\
4l_{28} + 8l_{24} - 4l_{22} + 2l_{20} - 2l_{14} - 2l_{12} \\
- 4l_{26} + 4l_{24} - 4l_{22} + 2l_{20} - 2l_{14} - 2l_{12} + 4 \\
2l_{20} + 4l_{16} - 2l_{14} - 2l_{12} \\
- 2l_{18} + 2l_{16} - 2l_{14} - 2l_{12} + 2
\end{aligned}
$$



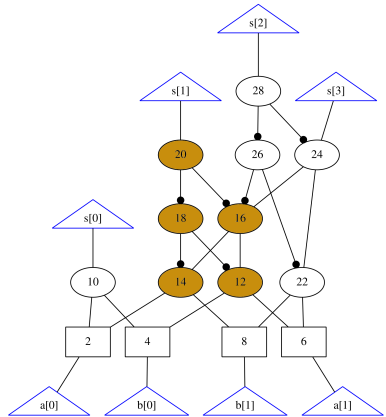
On-the-fly Linearization

Gate polynomials $G(C) \subseteq \mathbb{Q}[X]$.

$$
\begin{aligned}
&-s_3 + l_{24} && -l_{22} + a_1 b_1 \\
&-s_2 + l_{28} && -l_{20} - l_{18} - l_{16} + 1 \\
&-s_1 + l_{20} && -l_{18} + l_{16} - l_{14} - l_{12} + 1 \\
&-s_0 + l_{10} && -l_{16} + l_{14} l_{12} \\
&-l_{28} - l_{26} - l_{24} + 1 && -l_{14} + a_0 b_1 \\
&-l_{26} + l_{24} - l_{22} - l_{16} + 1 && -l_{12} + a_1 b_0 \\
&-l_{24} + l_{22} l_{16} && -l_{10} + a_0 b_0
\end{aligned}
$$

Specification $\mathcal{S}_n \in \mathbb{Q}[X]$.

$$
\begin{aligned}
8s_3 + 4s_2 + 2s_1 + s_0 - 4l_{22} - 2l_{14} - 2l_{12} - l_{10} \\
4s_2 + 2s_1 + s_0 + 8l_{24} - 4l_{22} - 2l_{14} - 2l_{12} - l_{10} \\
4l_{28} + 8l_{24} - 4l_{22} + 2l_{20} - 2l_{14} - 2l_{12} \\
- 4l_{26} + 4l_{24} - 4l_{22} + 2l_{20} - 2l_{14} - 2l_{12} + 4 \\
2l_{20} + 4l_{16} - 2l_{14} - 2l_{12} \\
- 2l_{18} + 2l_{16} - 2l_{14} - 2l_{12} + 2 \\
0
\end{aligned}
$$



MULTILING

- Builds on AMULET 2.2, written in C++
- Variables are sorted based on minimum distance to primary inputs
- Gröbner basis engine: MSOLVE[2]
- Non-linear rewriting as fall-back, when distance is below 6.

[2] J. Berthomieu, C. Eder, and M. Safey El Din. msolve: A Library for Solving Polynomial Systems. ISSAC, 2021

Evaluation - Optimized Multipliers

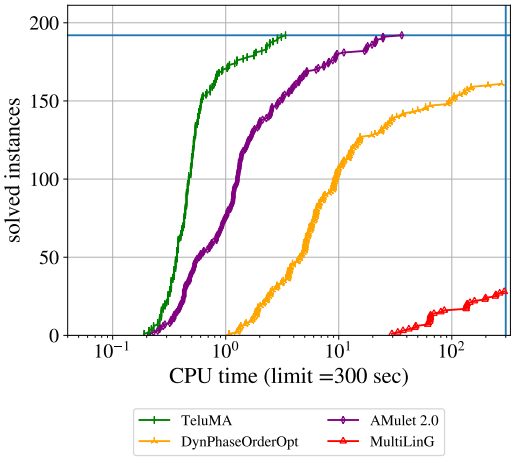
ABC-benchmarks			Related work			MULTILING		
n	Optimization	Nodes	TELUMA	AMULET 2.2	DPOO	Time	PP-Nodes	#msolve
64	resyn	32064	0.3	TO	1.0	5.6	7996	10
64	resyn3	32064	0.3	0.2	1.0	5.6	8000	0
64	dc2	32064	0.2	0.3	1.0	5.8	8000	0
64	complex[3]	32063	TO	TO	1.0	6.3	7996	9
128	resyn	129664	1.3	TO	5.7	200.6	32380	10
128	resyn3	129664	1.2	TO	7.7	209.3	32384	0
128	dc2	129664	1.1	TO	6.6	214.6	32384	0
128	complex	129663	TO	TO	5.8	214.1	32380	9

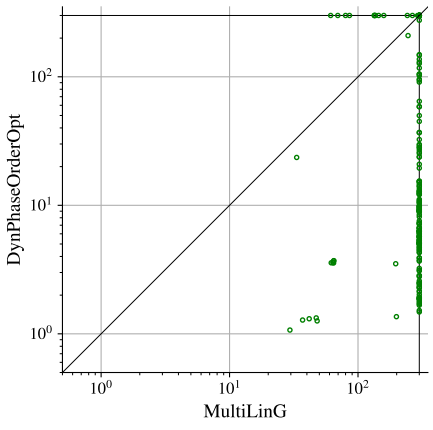
time in sec, TO = 1200 sec, DPOO = DYNPHASEORDEROPT

[3] `-c "logic; mfs2 -W 20; ps; mfs; st; ps; dc2 -l; ps; resub -l -K 16 -N 3 -w 100; ps; logic; mfs2 -W 20; ps; mfs; st; ps; iresyn -l; ps; resyn; ps; resyn2; ps; resyn3; ps; dc2 -l; ps;"`

Evaluation - 64-bit Multipliers

Verification of 192 unsigned 64-bit multipliers





Evaluation - 64-bit Multipliers

Benchmarks		PP	Time (s)		MSOLVE Calls					Nlin(%)
Name	Nodes	Nodes	Total	MSOLVE (%)	#Calls	d=3	d=4	d=5	d=6	
sparrc	48000	11968	92.7	79.5 (86.0)	3968	0	0	0	0	0.0
spwtcl	68747	25177	176.2	122.6 (69.8)	4003	13	0	0	0	0.0
spbdrc	49116	12247	110.0	87.3 (79.8)	3968	1	0	0	0	0.0
sposcl	71291	26835	151.8	105.5 (69.9)	3966	0	0	0	0	0.0
spctrc	41248	8402	196.2	162.9 (83.3)	7763	1884	1883	0	0	0.0
spcnrc	47236	11136	136.6	105.7 (77.6)	3037	0	0	0	0	0.0
bparrc	38311	8931	98.7	38.8 (39.6)	2112	2	1	1	1	38.2
bpwtcl	57556	22139	159.1	46.6 (29.5)	2137	11	1	1	1	25.5
bpbdrc	37365	8696	110.0	49.0 (44.9)	2110	2	1	1	1	39.2
bposcl	58759	22880	170.9	49.5 (29.1)	2109	1	1	1	1	24.9
bpdtrc	36044	8351	108.5	47.2 (43.6)	2087	1	1	1	1	40.6
bpcnrc	35557	7916	89.0	39.1 (44.0)	1609	1	1	1	1	41.2

Table: Results on solved aoki benchmarks.

Conclusion

If the specification polynomial is linear,

a Gröbner basis with respect to a
degree reverse lexicographic term ordering

contains linear polynomials that suffice
to derive correctness of the circuit.

- Full Gröbner basis computation is hard
- Our approach linearizes polynomials on the fly
- Robust on optimized benchmarks and complements existing $\prec_{\text{lex}}$ techniques.

References I

[Biere SATComp'16] A. Biere. Collection of Combinational Arithmetic Miters Submitted to the SAT Competition 2016. In SAT Competition 2016, pages 65–66, Dep. of Computer Science Report Series B, University of Helsinki, 2016.

[BiereFallerFazekasFleuryFroleyksPollitt SATComp'24] A. Biere, T. Faller, K. Fazekas, M. Fleury, N. Froleyks and F. Pollitt CaDiCaL, Gimsatul, IsaSAT and Kissat Entering the SAT Competition 2024. In SAT Competition 2024, pages 8–10, Dep. of Computer Science Report Series B, University of Helsinki, 2024.

[Buchberger'65] B. Buchberger. Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal. PhD Thesis, University of Innsbruck, 1965.

[CiesielskiYuBrownLiuRossi DAC'15] M. Ciesielski, C. Yu, W. Brown, D. Liu, and A. Rossi. Verification of Gate-level Arithmetic Circuits by Function Extraction. In Proc. of DAC'15, pages 52:1–52:6, ACM, 2015.

[ChenBryant DAC'95] Y. Chen and R. Bryant. Verification of Arithmetic Circuits with Binary Moment Diagrams. In Proc. of DAC'95, pages 535–541, ACM, 1995.

References II

[DrechslerMahzoon ISEEIE'22] R. Drechsler and A. Mahzoon. Design Modification for Polynomial Formal Verification. In Proc. of ISEEIE'22, pages 187–194, IEEE, 2022.

[KaufmannBeameBiereNordström DATE'22] D. Kaufmann, P.Beame, A. Biere and J. Nordström. Adding Dual Variables to Algebraic Reasoning for Gate-Level Multiplier Verification. In Proc. of DATE'22, pages 1431–1436, IEEE, 2022.

[KaufmannBiereKauers FMCAD'19] D. Kaufmann, A. Biere and M. Kauers. Verifying Large Multipliers by Combining SAT and Computer Algebra. In Proc. of FMCAD'19, pages 28–36, IEEE, 2019.

[KaufmannBiereKauers FMSD'20] D. Kaufmann, A. Biere and M. Kauers. Incremental Column-Wise Verification of Arithmetic Circuits Using Computer Algebra. In FMSD, vol 56, pages 22–54, 2020.

[KaufmannFleuryBiere FMCAD'20] D. Kaufmann, M. Fleury and A. Biere. Pacheck and Pastèque Checking Practical Algebraic Calculus Proofs. In Proc. of FMCAD'20, pages 264–269, TU Vienna Academic Press, 2020.

References III

[KaufmannFleuryBiereKauers FMSD'21] D. Kaufmann, M. Fleury, A. Biere and M. Kauers. Practical Algebraic Calculus and Nullstellensatz with the Checkers Pacheck and Pastèque and Nuss-Checker. In FMSD, online first, 2021.

[KonradScholl FMCAD'24] A. Konrad and C. Scholl. Practical Algebraic Calculus and Nullstellensatz with the Checkers Pacheck and Pastèque and Nuss-Checker. In FMSD, online first, 2021.

[LvKallaEnescu TCAD'13] J. Lv, P. Kalla, F. Enescu. Efficient Gröbner Basis Reductions for Formal Verification of Galois Field Arithmetic Circuits. In IEEE TCAD, vol. 32, pages 1409–1420, 2013.

[MahzoonGroßeDrechsler DAC'19] A. Mahzoon, D. Große and R. Drechsler. RevSCA: Using Reverse Engineering to Bring Light into Backwards Rewriting for Big and Dirty Multipliers. In Proc. of DAC'19, pages 185:1–185:6, ACM, 2019.

[MahzoonGroßeDrechsler ICCAD'18] A. Mahzoon, D. Große and R. Drechsler. PolyCleaner: Clean your Polynomials before Backward Rewriting to verify Million-gate Multipliers. In Proc. of ICCAD'18, pages 129:1–129:8, ACM, 2018.

References IV

[MahzoonGroßeSchollDrechsler DATE'20] A. Mahzoon, D. Große, Christoph Scholl and R. Drechsler. Towards Formal Verification of Optimized and Industrial Multipliers. In Proc. of DATE'20, pages 544–549, DATE, 2020.

[RitircBiereKauers DATE'18] D. Ritirc, A. Biere and M. Kauers. Improving and Extending the Algebraic Approach for Verifying Gate-Level Multipliers. In Proc. of DATE'18, pages 1556–1561, IEEE, 2018.

[RitircBiereKauers FMCAD'17] D. Ritirc, A. Biere and M. Kauers. Column-Wise Verification of Multipliers Using Computer Algebra. In Proc. of FMCAD'17, pages 23–30, IEEE, 2017.

[SayedGroßeKühneSoekenDrechsler DATE'16] A. Sayed-Ahmed, D. Große, U. Kühne, M. Soeken, and R. Drechsler. Formal verification of integer multipliers by combining Gröbner basis with logic reduction. In Proc. of DATE'16, pages 1048–1053, IEEE, 2016.

[SchollKonradMahzoonGroßeDrechsler DATE'21] C. Scholl, A. Konrad, A. Mahzoon, D. Große and R. Drechsler. Verifying Dividers Using Symbolic Computer Algebra and Don't Care Optimization. In Proc. of DATE'21, pages 1110–1115, IEEE, 2021.

References V

[Temel TACAS'24] M. Temel eSCMul: Verified Implementation of S-C-Rewriting for Multiplier Verification. In Proc. of TACAS'24, pages 485–507, Springer, 2024.

[TemelSlobodovaHunt CAV'20] M. Temel, A. Slobodova and W. Hunt. Automated and Scalable Verification of Integer Multipliers. In Proc. of CAV'20, pages 485–507, Springer, 2020.