**Motivation & Solving Techniques**

**Given:** a (gate level) multiplier circuit $C$ for fixed-size bitwidth $n$

$$ (4a_2 + 2a_1 + 1a_0) \times (4b_2 + 2b_1 + 1b_0) $$

**Motivation**
- verify circuits to avoid issues like Pentium FDIV bug

**Solving Techniques**
- SAT using CNF encoding
- Binary Moment Diagrams (BMD)
- Algebraic reasoning

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

$$ 2^{n-1} \sum_{i=0}^{n-1} 2^i s_i = \left( \sum_{i=0}^{n-1} 2^i a_i \right) \left( \sum_{i=0}^{n-1} 2^i b_i \right) ? $$
Related Work

- **Binary moment diagrams**

- **Algebraic reasoning**

- **Proofs**
Basic Idea of Algebraic Approach

Multiplier

Polynomials

\[ B = \{ x - a_0 \cdot b_0, \\
    y - a_1 \cdot b_1, \\
    s_0 - x \cdot y, \\
    \ldots \} \]

Specification

Membership Test

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

\[ = 0 \checkmark \\
\neq 0 \times \]
**N-bit Multipliers**

- **inputs**: $a_0, \ldots, a_{n-1}$
- **inputs**: $b_0, \ldots, b_{n-1}$
- **outputs**: $s_0, \ldots, s_{2n-1}$
- **internal**: $g_1, \ldots, g_k$
Circuit Polynomials

Let \( X = a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}, g_1, \ldots, g_k, s_0, \ldots, s_{2n-1} \).

Definition 1 (Gate polynomial).
Polynomial \( p \in \mathbb{Q}[X] \) representing a circuit gate.

\[
\begin{align*}
    s_3 &= g_1 \land g_4 & -s_3 + g_1 g_4, \\
    s_2 &= g_1 \lor g_4 & -s_2 + g_1 + g_4 - 2g_1 g_4, \\
    g_4 &= g_2 \land g_3 & -g_4 + g_2 g_3, \\
    s_1 &= g_2 \lor g_3 & -s_1 + g_2 + g_3 - 2g_2 g_3, \\
    g_1 &= a_1 \land b_1 & -g_1 + a_1 b_1, \\
    g_2 &= a_0 \land b_1 & -g_2 + a_0 b_1, \\
    g_3 &= a_1 \land b_0 & -g_3 + a_1 b_0, \\
    s_0 &= a_0 \land b_0 & -s_0 + a_0 b_0
\end{align*}
\]

Definition 2 (Field polynomial).
Polynomial \( p \in \mathbb{Q}[X] \) which models the domain.

\[
\begin{align*}
    a_1, a_0 &\in \mathbb{B} & a_1(1 - a_1), a_0(1 - a_0), \\
    b_1, b_0 &\in \mathbb{B} & b_1(1 - b_1), b_0(1 - b_0)
\end{align*}
\]
Circuit Polynomials

Let $X = a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}, g_1, \ldots, g_k, s_0, \ldots, s_{2n-1}$.

Definition 1 (Gate polynomial).
Polynomial $p \in \mathbb{Q}[X]$ representing a circuit gate.

\[
\begin{align*}
    s_3 &= g_1 \land g_4 & -s_3 + g_1 g_4, \\
    s_2 &= g_1 \lor g_4 & -s_2 + g_1 + g_4 - 2g_1 g_4, \\
    g_4 &= g_2 \land g_3 & -g_4 + g_2 g_3, \\
    s_1 &= g_2 \lor g_3 & -s_1 + g_2 + g_3 - 2g_2 g_3, \\
    g_1 &= a_1 \land b_1 & -g_1 + a_1 b_1, \\
    g_2 &= a_0 \land b_1 & -g_2 + a_0 b_1, \\
    g_3 &= a_1 \land b_0 & -g_3 + a_1 b_0, \\
    s_0 &= a_0 \land b_0 & -s_0 + a_0 b_0
\end{align*}
\]

Definition 2 (Field polynomial).
Polynomial $p \in \mathbb{Q}[X]$ which models the domain.

\[
\begin{align*}
    a_1, a_0 &\in \mathbb{B} & a_1(1-a_1), a_0(1-a_0), \\
    b_1, b_0 &\in \mathbb{B} & b_1(1-b_1), b_0(1-b_0)
\end{align*}
\]
**Ideals associated to Circuits**

**Definition 3 (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 \( g_1, \ldots, g_k, s_0, \ldots, s_{2n-1} \) implied by the gates of the circuit \( C \) the substitution of these values into \( p \) gives zero.

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

**Definition 4 (Ideal).** A nonempty subset \( I \subseteq \mathbb{Q}[X] \) is called an ideal if

- \( \forall p, q \in I : p + q \in I \)
- \( \forall p \in \mathbb{Q}[X] \forall q \in I : pq \in I \)
Examples for PCCs:

- $s_0 - a_0b_0$
- $a_1 - a_1$
- $g_2^2 - g_2$
- $s_1g_4$
- $\ldots$

**Definition 5 (Multiplier).** A circuit $C$ is called a *multiplier* if

$$\sum_{i=0}^{2n-1} 2^i s_i - \left( \sum_{i=0}^{n-1} 2^i a_i \right) \left( \sum_{i=0}^{n-1} 2^i b_i \right) \in I(C).$$
**Definition 6 (Term order).**
An order $\leq$ is fixed on the set of terms compatible with multiplication.

- every ideal $I$ of $\mathbb{Q}[X]$ has a Gröbner basis $G$ with $I = \langle G \rangle$.

- ideal membership test
  - multivariate polynomial division with remainder
  - remainder $r$ contains no term that is a multiple of any of the leading terms of $G$

- construction algorithm by Buchberger which given an arbitrary basis of an ideal $I$ computes a Gröbner basis of it (doubly exponential)
GRÖBNER BASIS

We can deduce at least some elements of \( I(C) \):

- \( G = \) Gate Polynomials + (Input) Field Polynomials
- Let \( J(C) = \langle G \rangle \).
- Term order: output variable of a gate is greater than input variables

THEOREM

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

Proof idea: Application of Buchberger’s Product criterion.

THEOREM (SOUNDNESS AND COMPLETENESS)

For all acyclic circuits \( C \), we have \( J(C) = I(C) \).
**Non-Incremental Algorithm**

**Algorithm 1 (Non-Incremental Checking Algorithm).**
Divide 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 elements of $G$ until no further reduction is possible, then $C$ is a multiplier iff remainder is zero.

**Implications**
- Leading coefficient $-1$ of all gate polynomials, computation stays in $\mathbb{Z}$.
- Completeness proof allows to derive input assignment if $C$ is incorrect.
- Still can use rational coefficients $\mathbb{Q}$ (important for Singular).

Generally the size of intermediate results in Algorithm 1 increases drastically:
- 8-bit multiplier can not be verified within 20 minutes.
- Tailored heuristics become very important.
**Rows and Columns**

\[(4a_2 + 2a_1 + 1a_0) \times (4b_2 + 2b_1 + 1b_0)\]

\[32s_5 + 16s_4 + 8s_3 + 4s_2 + 2s_1 + 1s_0\]

\[(4a_2 + 2a_1 + 1a_0) \times (4b_2 + 2b_1 + 1b_0)\]

\[32s_5 + 16s_4 + 8s_3 + 4s_2 + 2s_1 + 1s_0\]
Row-Wise Order

\[(4a_2 + 2a_1 + 1a_0) \times (4b_2 + 2b_1 + 1b_0)\]

not robust under permutation
**Definition 7 (Partial Products).** Let \( P_k = \sum_{k = i+j} a_i b_j \).
Definition 8 (Input Cone).
For each output bit $s_i$ we determine its input cone

$$l_i = \{\text{gate } g \mid g \text{ is in input cone of output } s_i\}$$

Definition 9 (Slice).
Slices $S_i$ are defined as the difference of consecutive cones $l_i$:

$$S_0 = l_0 \quad S_{i+1} = l_{i+1} \setminus \bigcup_{j=0}^{i} S_j$$

Definition 10 (Sliced Gröbner Bases).
Let $G_i$ be the set of polynomial representations of the gates in slice $S_i$. 
**Definition 11 (Carry Recurrence Relation).**

- A sequence of $2n + 1$ polynomials $C_0, \ldots, C_{2n}$ over the variables of $C$ is called a *carry sequence of carry polynomials*.

- For $0 \leq i < 2n$, carry polynomial $C_i$ and output $s_i$ let

$$-C_i + 2C_{i+1} + s_i - P_i$$

denote the *carry recurrence relation* $R_i$ for column $i$.

- Then $R_i$ *holds* on $C$ if it vanishes in $I(C)$, i.e., $R_i \in I(C)$.

**Theorem**

*Let $C$ be a circuit where all carry recurrence relations hold. Then $C$ is a multiplier in the sense of Def. 6, iff $C_0 - 2^{2n}C_{2n} \in I(C)$.***
Algorithm 2 (Incremental 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

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

return $C_0 = 0$
MULTIPLIERS

Column-Wise Verification of Multipliers Using Computer Algebra
COLUMN-WISE VERIFICATION OF MULTIPLIERS USING COMPUTER ALGEBRA
ENGINEERING

AIG

\textbf{AigMulToPoly}
Polynomial representation

CAS-File

Mathematica/Singular
Application of Alg. 2

$C_0$
# Experiments

| multiplier | Bit width | Mathematica | | | Singular |
|------------|-----------|-------------|---|---|---|---|
|            |           | Alg. 1 | Alg. 2 | Alg. 1 | Alg. 2 |
|            | col | row | col | row |
| btor       | 16   | 12   | 12   | 4   | 2   | 2   | 1   |
| btor       | 24   | 102  | 101  | 14  | 12  | 13  | 4   |
| btor       | 32   | 531  | 491  | 35  | 53  | 58  | 16  |
| btor       | 40   | TO   | TO   | 78  | 210 | 219 | 55  |
| btor       | 48   | TO   | TO   | 156 | 602 | 621 | 145 |
| btor       | 56   | MO   | MO   | 263 | MO  | MO  | 226 |
| btor       | 64   | MO   | MO   | 409 | MO  | MO  | MO  |
| sp-ar-rc   | 8    | TO   | TO   | 2   | TO  | TO  | 1   |
| sp-ar-rc   | 16   | TO   | TO   | 7   | TO  | TO  | 1   |
| sp-ar-rc   | 32   | TO   | TO   | 67  | TO  | TO  | 39  |
| sp-ar-rc   | 64   | MO   | MO   | 841 | MO  | MO  | MO  |

**Table:** time in sec; TO = 1200 sec, MO = 14 GB
Conclusion & Future Work

Conclusion:
- simple and precise mathematical formulation
- new incremental column-based verification approach
- magnitude faster than previous non-incremental approach
- using computer algebra systems

Future Work:
- other word-level operators (shift, division, ...)
- more complex multipliers
- extend our methods to floating points
- negative numbers

Experimental data, source code, benchmarks, and scripts are available at http://fmv.jku.at/cwmulverca.
COLUMN-WISE VERIFICATION OF MULTIPLIERS USING COMPUTER ALGEBRA

Daniela Ritirc, Armin Biere, Manuel Kauers
Johannes Kepler University
Linz, Austria

FMCAD 2017
October 02 - 06, 2017
Vienna, Austria