VERIFYING LARGE MULTIPLIERS BY COMBINING SAT AND COMPUTER ALGEBRA

Daniela Kaufmann, Armin Biere and Manuel Kauers
Johannes Kepler University
Linz, Austria

FMCAD 2019

October 23, 2019
San Jose, CA, USA

Der Wissenschaftsfonds.

Rigorous Systems Engineering

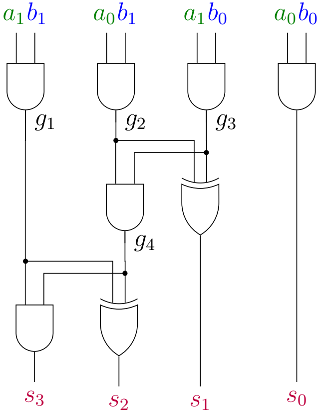
JOHANNES KEPLER UNIVERSITÄT LINZ

Circuits

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

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

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

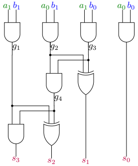


Verification Techniques

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

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

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

Ideal Membership Test

$= 0$ ✓

$\neq 0$ ✗

Contributions

1. Modular Reasoning
2. Combine SAT and Computer Algebra
3. Preprocessing Techniques
4. Tool: AMULET

Multiplier Specification

Unsigned integers:

$$\mathcal{U}_n = \sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right) \qquad \in \mathbb{Z}[X]$$

Multiplier Specification

Unsigned integers:

$$\mathcal{U}_n = \sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right) \qquad \in \mathbb{Z}[X]$$

Signed integers:

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

Multiplier Specification

Unsigned integers:

$$\mathcal{U}_n = \sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right) \qquad \in \mathbb{Z}[X]$$

Signed integers:

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

Truncated multiplication of integers:

$$\mathcal{T}_n = \sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right) \qquad \in \mathbb{Z}_{2^n}[X]$$

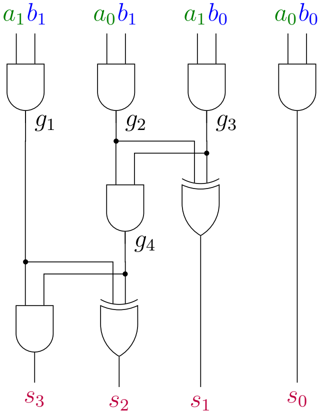
Circuit Polynomials

Gate polynomials $G(C)$.

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

Boolean value constraints $B_0(C)$.

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



Ideals

Ideal. Let R be a ring. A nonempty subset $I \subseteq R[X]$ is called an ideal if

$$\forall\, p, q \in I : p + q \in I \qquad \text{and} \qquad \forall\, p \in R[X]\; \forall\, q \in I : pq \in I$$

Ideal membership test. Given a polynomial $q \in R[X]$ and a (finite) set of polynomials $P \subseteq R[X]$, decide whether $q \in \langle P \rangle$, where $\langle P \rangle$ is the smallest ideal containing all elements of P, also known as the ideal *generated by* P.

$$\begin{aligned}
\mathcal{U}_n &\in \langle G(C) \cup B_0(C) \rangle \subseteq \mathbb{Z}[X] \\
\mathcal{S}_n &\in \langle G(C) \cup B_0(C) \rangle \subseteq \mathbb{Z}[X] \\
\mathcal{T}_n &\in \langle G(C) \cup B_0(C) \rangle \subseteq \mathbb{Z}_{2^n}[X]
\end{aligned}$$

Ideals

Ideal. Let R be a ring. A nonempty subset $I \subseteq R[X]$ is called an ideal if

$$\forall\, p, q \in I : p + q \in I \qquad \text{and} \qquad \forall\, p \in R[X]\ \forall\, q \in I : pq \in I$$

Ideal membership test. Given a polynomial $q \in R[X]$ and a (finite) set of polynomials $P \subseteq R[X]$, decide whether $q \in \langle P \rangle$, where $\langle P \rangle$ is the smallest ideal containing all elements of P, also known as the ideal *generated by* P.

UMLT. Let $P \subseteq R[X]$. If for a certain term order, all leading terms of P only consist of a single variable with exponent 1 and are unique and further $\mathrm{lc}(p) \in R^\times$ for all $p \in P$, then we say P has *unique monic leading terms*.

Soundness and completeness

- $P \vdash_R q \iff q \in \langle P \rangle + \langle B_0(P) \rangle$
- $P \models_R q \iff \forall \varphi : \forall p \in P : \varphi(p) = 0 \Rightarrow \varphi(q) = 0$

Theorem (Soundness)

Let $P \subseteq R[X]$ be a finite set of polynomials with UMLT and $q \in R[X]$, then

$$P \vdash_R q \quad \Rightarrow \quad P \models_R q.$$

Theorem (Completeness)

Let $P \subseteq R[X]$ be a finite set of polynomials with UMLT. Then for every $q \in R[X]$ we have

$$P \models_R q \quad \Rightarrow \quad P \vdash_R q.$$

Modular reasoning

Previous work: $\mathbb{Q}[X]$

- unsigned integers
- contains $\mathbb{Z}[X]$
- $\mathbb{Q}$ is a field
- Gröbner basis theory

Now: $\mathbb{Z}_l[X]$ for $l \in \mathbb{N}$

- truncated multiplication
- elimination of monomials

Modular reasoning

Unsigned integers:

$$\mathcal{U}_n = \sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right) \qquad \in \mathbb{Z}_{2^{2n}}[X]$$

Signed integers:

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

Truncated multiplication of integers:

$$\mathcal{T}_n = \sum_{i=0}^{n-1} 2^i s_i - \sum_{i=0}^{n-1} \sum_{j=0}^{n-1-i} 2^{i+j} a_i b_j \qquad \in \mathbb{Z}_{2^n}[X]$$

Modular reasoning

Previous work: $\mathbb{Q}[X]$

- unsigned integers
- contains $\mathbb{Z}[X]$
- $\mathbb{Q}$ is a field
- Gröbner basis theory

Now: $\mathbb{Z}_l[X]$ for $l \in \mathbb{N}$

- truncated multiplication
- elimination of monomials

$\mathbb{Z}[X]$

- $\mathbb{Z}$ is a principal ideal domain
- D-Gröbner basis theory

D-Gröbner basis

- Gröbner bases theory, where coefficient domains D are PIDs.
- Offers decision procedure (D-reduction) for ideal membership test in $D[X]$.

Let $q \in D[X]$ and $P \subseteq D[X]$: $$q \in \langle P \rangle \quad \Leftrightarrow \quad q \xrightarrow{P} 0$$

- Every ideal of $D[X]$ has a D-Gröbner basis.
- There is an (expensive) algorithm which, given an arbitrary basis of an ideal, computes a D-Gröbner basis.

D-Gröbner basis applied to circuit verification

Theorem

Let R be a PID and let $G(C) \cup B_0(C) \subseteq R[X]$ have UMLT.
Then $\quad G(C) \cup B_0(C) \quad$ *is a D-Gröbner basis of* $\quad \langle G(C) \cup B_0(C) \rangle \subseteq R[X]$.

D-Gröbner basis applied to circuit verification

Theorem

Let R *be a PID and let* $G(C) \cup B_0(C) \subseteq R[X]$ *have UMLT.*
Then $G(C) \cup B_0(C)$ *is a D-Gröbner basis of* $\langle G(C) \cup B_0(C) \rangle \subseteq R[X]$.

- We want $R = \mathbb{Z}_l$ for $l \in \mathbb{N}$.
- $\mathbb{Z}_l$ is not a PID.
- $\mathbb{Z}$ is a PID.

D-Gröbner basis applied to circuit verification

Theorem

Let R be a PID and let $G(C) \cup B_0(C) \subseteq R[X]$ have UMLT.
Then $G(C) \cup B_0(C)$ is a D-Gröbner basis of $\langle G(C) \cup B_0(C) \rangle \subseteq R[X]$.

- We want $R = \mathbb{Z}_l$ for $l \in \mathbb{N}$.
- $\mathbb{Z}_l$ is not a PID.
- $\mathbb{Z}$ is a PID.

Lemma

Let $l \in \mathbb{N}$ and let $G(C) \cup B_0(C) \subseteq \mathbb{Z}[X]$ have UMLT.
Then $G(C) \cup B_0(C) \cup \{l\}$ is a D-Gröbner basis of $\langle G(C) \cup B_0(C) \rangle + \langle l \rangle \subseteq \mathbb{Z}[X]$.

Correspondence lemma

Lemma

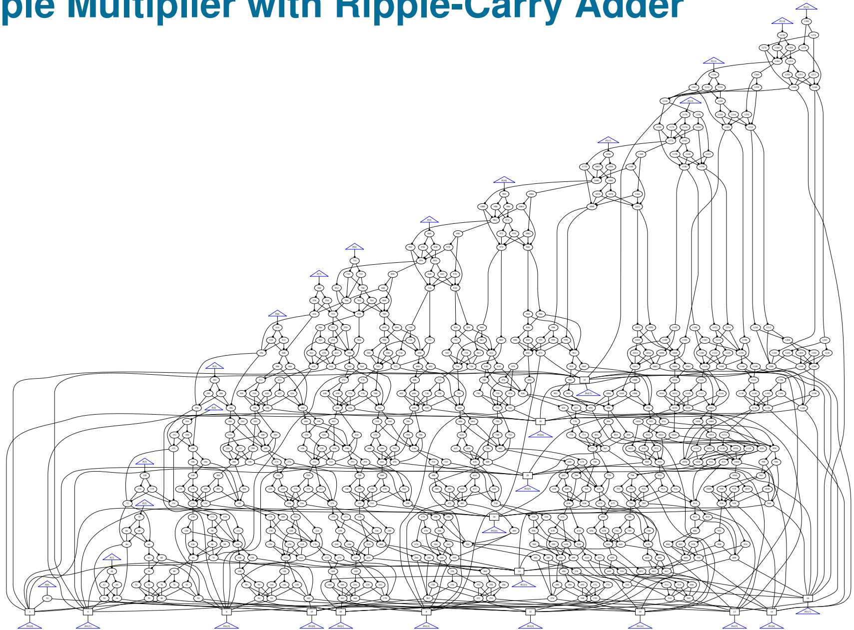
Let $l \in \mathbb{N}$ *and let* $I \subseteq \mathbb{Z}[X]$ *be an ideal. There is a bijective correspondence from*

$$q \in I + \langle l \rangle \subseteq \mathbb{Z}[X] \qquad \text{to} \qquad [q] \in \{[p] \mid p \in I\} \subseteq \mathbb{Z}[X]/\langle l \rangle,$$

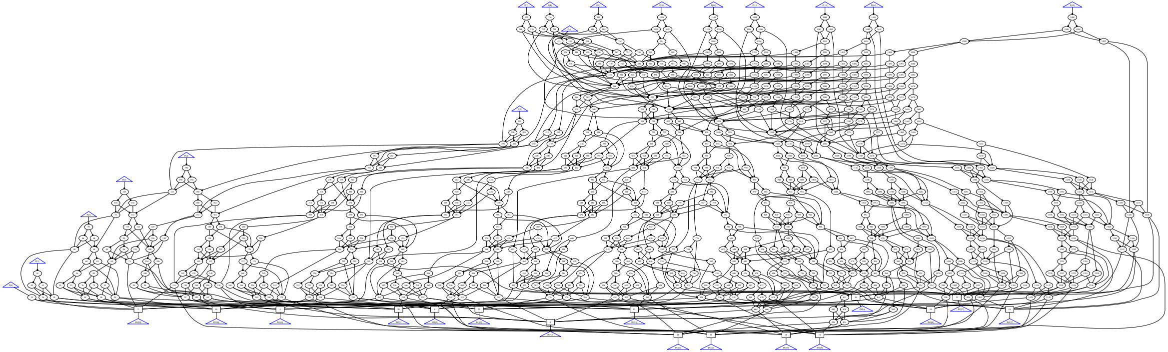
where $[q]$ *is the equivalence class of* q.

Furthermore $\mathbb{Z}[X]/\langle l \rangle \cong \mathbb{Z}_l[X]$.

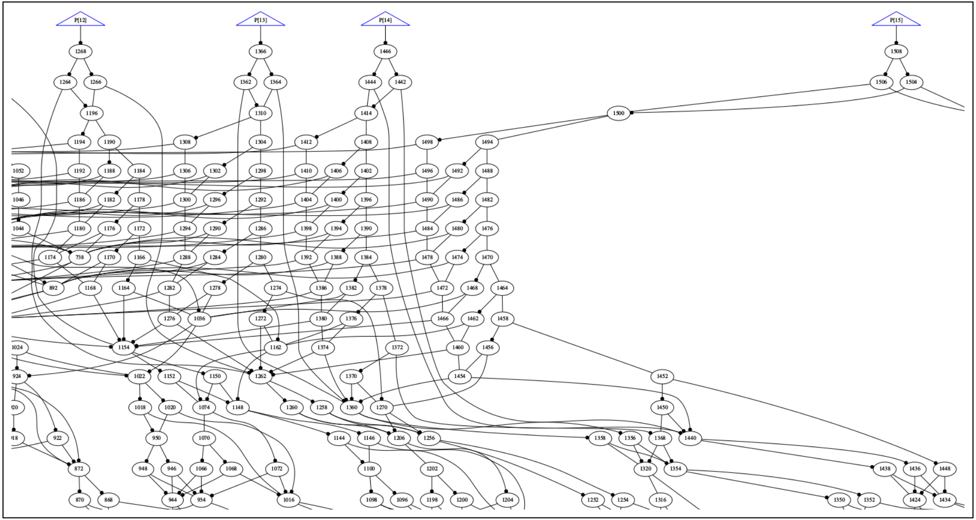
Simple Multiplier with Ripple-Carry Adder



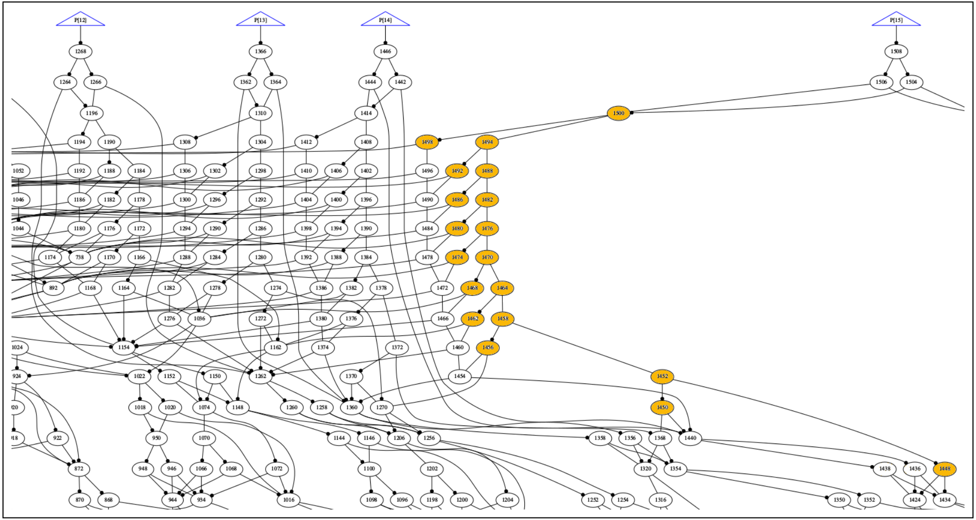
Complex Multiplier with Generate-and-Propagate Adder



Complex Multiplier with Generate-and-Propagate Adder

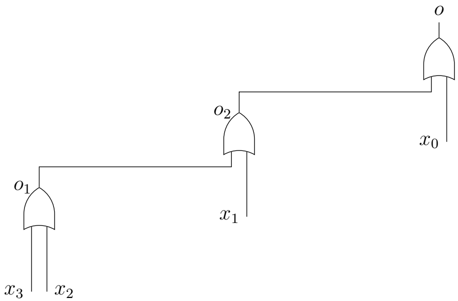


Complex Multiplier with Generate-and-Propagate Adder



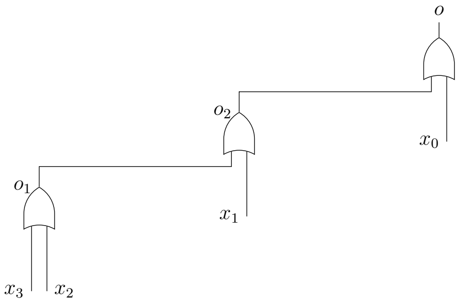
OR Gates

$$
\begin{aligned}
o = o_2 \vee x_0 \quad & -o + o_2 + x_0 - o_2 x_0, \\
o_2 = o_1 \vee x_1 \quad & -o_2 + o_1 + x_1 - o_1 x_1, \\
o_1 = x_3 \vee x_2 \quad & -o_1 + x_3 + x_2 - x_3 x_2
\end{aligned}
$$



OR Gates

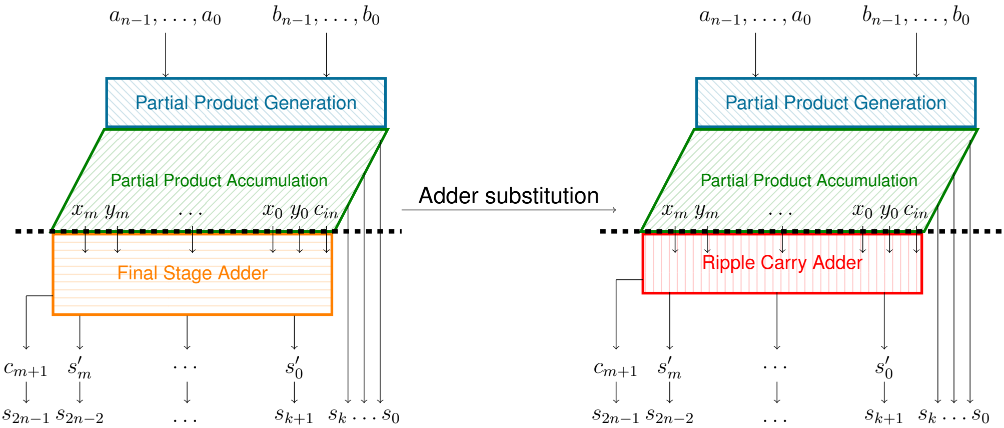
$$
\begin{aligned}
o &= o_2 \vee x_0 & -o + o_2 + x_0 - o_2 x_0, \\
o_2 &= o_1 \vee x_1 & -o_2 + o_1 + x_1 - o_1 x_1, \\
o_1 &= x_3 \vee x_2 & -o_1 + x_3 + x_2 - x_3 x_2
\end{aligned}
$$



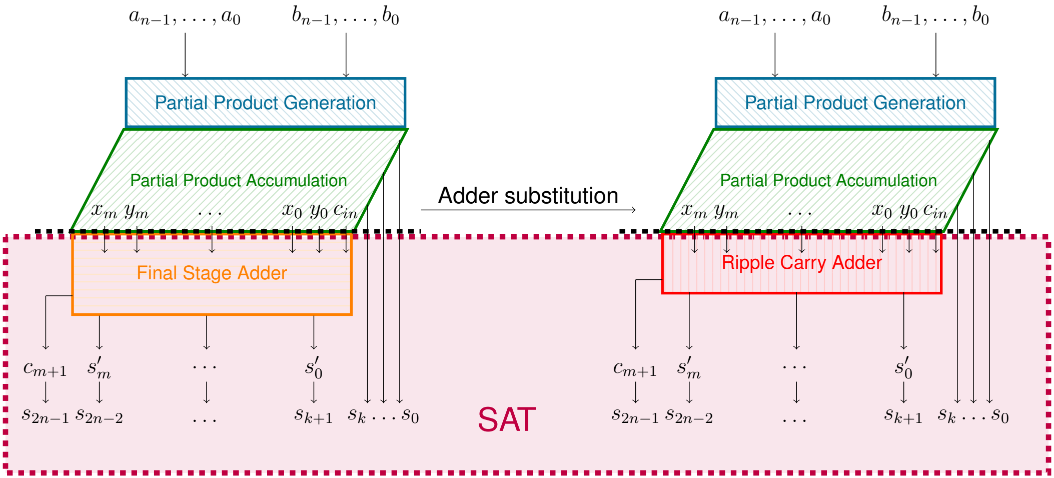
$o = x_0 + x_1 - x_0x_1 + x_2 - x_0x_2 - x_1x_2 + x_0x_1x_2 + x_3 - x_0x_3 - x_1x_3 + x_0x_1x_3 - x_2x_3 + x_0x_2x_3 + x_1x_2x_3 - x_0x_1x_2x_3$

$15 = 2^4 - 1$ monomials

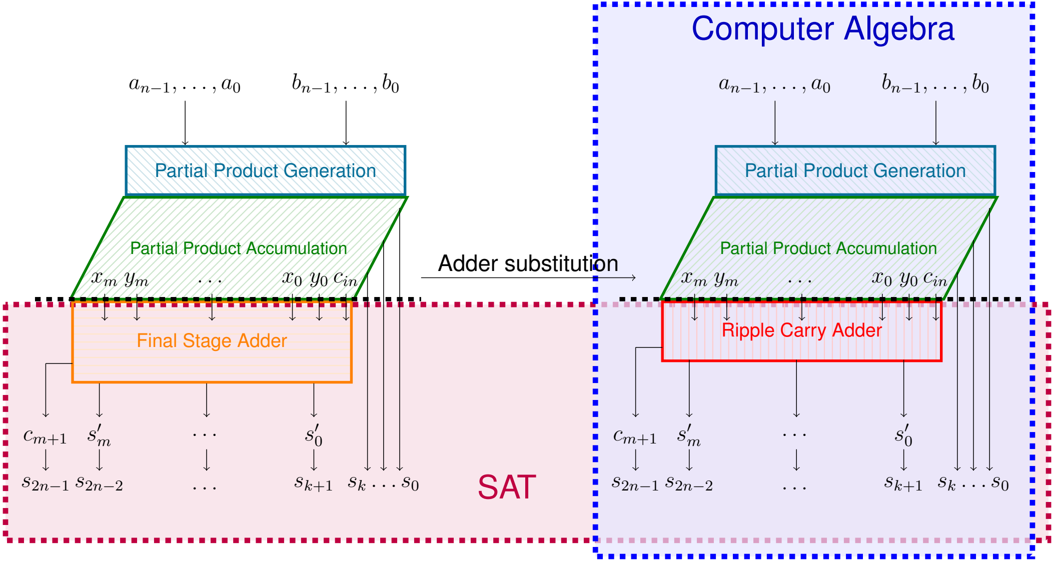
Adder Substitution



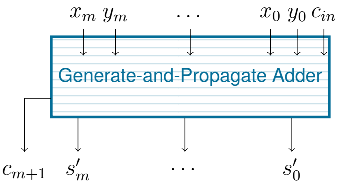
Adder Substitution



Adder Substitution

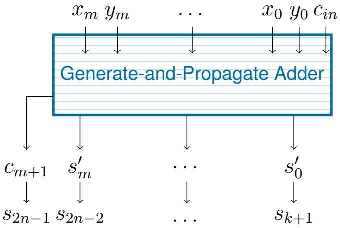


Adder Substitution



- $s'_i = p_i \oplus c_i$
- $p_i = x_i \oplus y_i$
- $c_i = (x_{i-1} \wedge y_{i-1}) \vee (c_{i-1} \wedge p_{i-1})$

Adder Substitution



- $s'_i = p_i \oplus c_i$
- $p_i = x_i \oplus y_i$
- $c_i = (x_{i-1} \wedge y_{i-1}) \vee (c_{i-1} \wedge p_{i-1})$

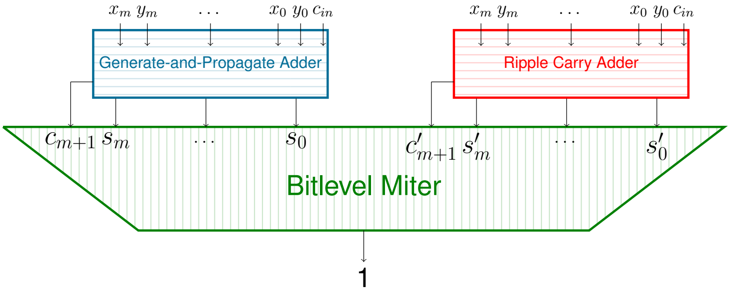
Algorithm: Identifying GP adders in AMULET

Input : Circuit C in AIG format

Output: Determine whether C might contain a GP adder

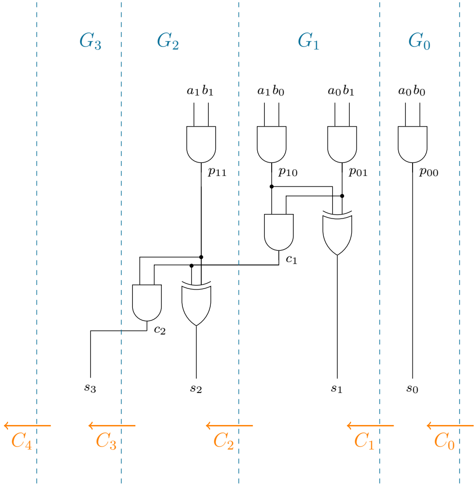
$j \leftarrow 2n-2$, $\tau \leftarrow 1$;
while τ **and** $j \geq 0$ **do**
    $\tau, c_j, p_j \leftarrow$ Check-if-XOR-and-Identify-p$_j$-and-c$_j$ (s_j);
    $x_j, y_j \leftarrow$Declare-Adder-Inputs (p_j, τ);
    $j \leftarrow j-1$;
end
$c_{in} \leftarrow c_j$;
for $i \leftarrow j$ **to** $2n-1$ **do**
    $m \leftarrow$ Follow-and-Mark-Paths(s_i);
end
return $m = 0$

SAT



- AIG is translated to CNF.
- CNF is given to SAT solver.
- To show correctness SAT solver needs to return UNSAT.

Computer Algebra



Algorithm: Verification flow in AMULET

Input : Substituted circuit C in AIG format

Output: Determine whether C is a multiplier

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

   $S_i \leftarrow$ Define-Cone-of-Influence(i);

   Order (S_i);

   Search-for-Booth-Encoding (S_i);

   Local-Elimination (S_i);

end

Global-Elimination ();

$C_0 \leftarrow$ Incremental-Reduction (); *FMCAD'17*

return $C_0 = 0$

Variable Elimination

Local Elimination

- iterate over each slice
- eliminate leading variable if it only occurs in one other polynomial inside the slice
- repeat until all leading variables are either contained in other slices or occur in multiple polynomials
- subsumes “Adder-Rewriting”, “XOR-Rewriting”, “Common-Rewriting”.

Global Elimination

- eliminate marked variables found in “Search-for-Booth-Encoding()”
- need to consider all slices

Incremental Reduction

Computer Algebra System

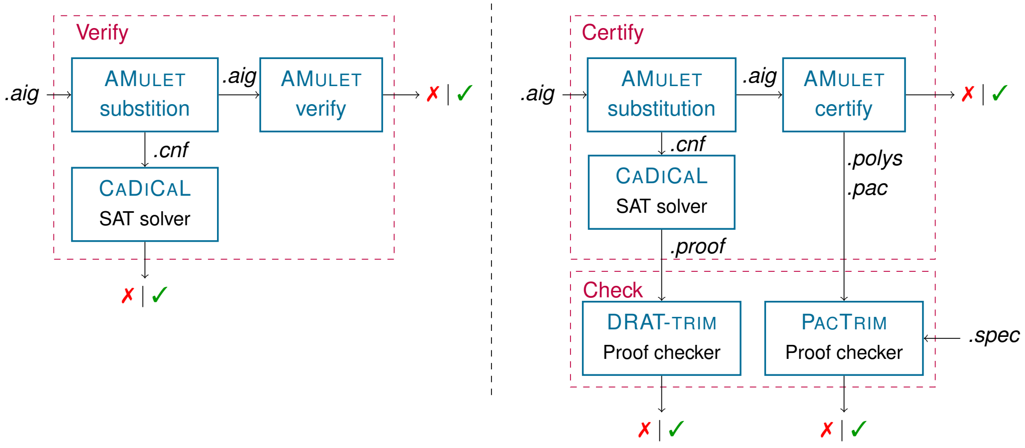
- Too general for our purpose.

Reduction Engine: AMULET

- Designed to make use of UMLT property.
- D-reduction amounts to substitution.
- On-the-fly reduction of Boolean value constraints.
- On-the-fly generation of proof certificates in PAC format[1].

[1] D. Ritirc, A. Biere, M. Kauers. A Practical Polynomial Calculus for Arithmetic Circuit Verification. In SC2, 2018.

Tool Flow


Verify
.aig
AMULET substition
.aig
AMULET verify
✗ | ✓
.cnf
CADICAL SAT solver
✗ | ✓
Certify
.aig
AMULET substitution
.aig
AMULET certify
✗ | ✓
.cnf
CADICAL SAT solver
.polys
.pac
.proof
Check
DRAT-TRIM Proof checker
PACTRIM Proof checker
.spec
✗ | ✓
✗ | ✓

Verification Time

architecture	n	SPEC	nosub	nomod	noelim	Verify sub	Verify cnf	Verify aig	Verify tot	MGD DAC19	CSYY TCAD19	KBK FMSD19
sp-ar-rc	64	u	1	1	2	0	0	1	1	NA_2	0	11
sp-dt-lf	64	u	TO	1	3	0	0	2	2	31	NA_3	TO
sp-wt-cl	64	u	TO	TO	3	0	9	1	11	96	NA_3	TO
sp-bd-ks	64	u	TO	TO	2	0	1	1	3	162	NA_3	TO
sp-ar-ck	64	u	TO	1	2	0	0	1	1	143	NA_3	TO
bp-ar-rc	64	u	1	TO	118	0	0	1	1	53	NA_3	TO
bp-ct-bk	64	u	TO	TO	100	0	0	1	2	119	NA_3	TO
bp-os-cu	64	u	2	TO	TO	0	0	2	2	95	NA_3	TO
bp-wt-cs	64	u	1	TO	114	0	0	1	1	75	NA_3	TO
sp-ar-rc	64	s	1	1	2	0	0	1	1	NA_1	0	NA_1
bp-wt-cl	64	s	TO	3	109	0	10	1	11	NA_1	NA_3	NA_1
btor	64	t	0	NA_3	1	0	0	0	1	NA_1	NA_1	NA_1

time in sec NA_1: tool not applicable to type SPEC NA_2: tool not yet available NA_3: incompleteness TO: 3600 sec

Verification and Certification Time

architecture	n	SPEC	Verify				Certify				Check			total	proof size	
			sub	cnf	aig	tot	sub	cnf	aig	tot	cnf	aig	tot		cnf	aig
sp-ar-rc	64	u	0	0	1	1	0	0	2	2	0	3	3	5	0	188 290
sp-dt-lf	64	u	0	0	2	2	0	0	2	3	0	3	3	6	34 423	186 170
sp-wt-cl	64	u	0	9	1	11	0	9	2	12	7	3	10	21	264 471	191 623
sp-bd-ks	64	u	0	1	1	3	0	2	2	4	1	3	4	8	78 567	190 915
sp-ar-ck	64	u	0	0	1	1	0	0	2	2	0	3	3	5	1 432	187 251
bp-ar-rc	64	u	0	0	1	1	0	0	2	2	0	3	3	5	0	161 815
bp-ct-bk	64	u	0	0	1	2	0	0	2	2	0	3	3	5	27 552	138 179
bp-os-cu	64	u	0	0	2	2	0	0	3	3	0	4	4	7	0	166 967
bp-wt-cs	64	u	0	0	1	1	0	0	2	2	0	3	3	6	0	161 747
sp-ar-rc	64	s	0	0	1	1	0	0	2	2	0	3	3	6	0	188 426
bp-wt-cl	64	s	0	10	1	11	0	10	2	12	7	3	10	22	261 650	151 355
btor	64	t	0	0	0	1	0	0	1	1	0	1	1	2	0	70 374

time in sec

Verification and Certification Time of Large Multipliers

architecture	n	Verify				Certify				Check			total	input	proof size	
		sub	cnf	aig	tot	sub	cnf	aig	tot	cnf	aig	tot		aig	cnf	aig
btor	512	0	0	16	16	0	0	23	23	0	7	7	30	2 m	0	7 m
kjvnkv	512	0	0	13	13	0	0	15	15	0	9	9	25	3 m	0	12 m
sp-ar-rc	512	0	0	13	13	0	0	16	16	0	10	10	26	3 m	0	12 m
sp-dt-lf	512	1	1	25	26	1	1	25	26	0	11	11	37	3 m	1 m	12 m
sp-wt-bk	512	1	0	26	27	0	0	26	26	0	11	11	38	3 m	626 k	12 m
btor	1024	2	0	177	179	2	0	219	219	0	51	51	272	8 m	0	31 m
kjvnkv	1024	2	0	91	93	2	0	172	172	0	72	72	245	12 m	0	49 m
btor	2048	17	0	1 493	1 510	17	0	2 552	2 552	0	430	430	2 982	33 m	0	125 m
kjvnkv	2048	18	0	1 129	1 147	18	0	2 077	2 077	0	1 228	1 228	3 307	50 m	0	196 m

time in min

Conclusion

- SAT and Computer Algebra
- Adder substitution

- Polynomial reasoning over more general rings
- Modular reasoning

- AMULET
- Variable elimination

- Combination of these ideas scales up to 2048 bits

VERIFYING LARGE MULTIPLIERS BY COMBINING SAT AND COMPUTER ALGEBRA

Daniela Kaufmann, Armin Biere and Manuel Kauers
Johannes Kepler University
Linz, Austria

FMCAD 2019

October 23, 2019
San Jose, CA, USA

Der Wissenschaftsfonds.

Rigorous Systems Engineering

JOHANNES KEPLER UNIVERSITÄT LINZ

Variable Elimination

Elimination ideal.

Let $I \subseteq D[X] = D[Y, z]$ be an ideal.

The ideal $I \cap D[Y] \subseteq D[Y]$ is an **elimination ideal** of I.

Variable Elimination

Elimination ideal.
Let $I \subseteq D[X] = D[Y, z]$ be an ideal.
The ideal $I \cap D[Y] \subseteq D[Y]$ is an **elimination ideal** of I.

- In general we need to compute a D-Gröbner basis w.r.t. a different term order.
- Element to be eliminated has to be largest element in order.
- Special case: UMLT property.

Variable Elimination

Elimination ideal.
Let $I \subseteq D[X] = D[Y, z]$ be an ideal.
The ideal $I \cap D[Y] \subseteq D[Y]$ is an **elimination ideal** of I.

- In general we need to compute a D-Gröbner basis w.r.t. a different term order.
- Element to be eliminated has to be largest element in order.
- Special case: UMLT property.

Example: $I \subseteq D[o, x, y, a, b]$. Eliminate y.

Ideal	Reduced ideal	Elimination ideal
$I = \{$	$I = \{$	$I \cap D[o, y, a, b] = \{$
$-o + xy - x - y + 1$	$-o + xab - x - ab + 1$	$-o + xab - -x - ab + 1$
$-x + ab - a - b + 1$	$-x + ab - a - b + 1$	$-x + ab - a - b + 1\}$
$-y + ab\}$	$-y + ab\}$	

Variable Elimination

Definition

Let $P \subseteq D[X]$ be a D-Gröbner basis of $\langle P \rangle$ with UMLT. We say P is reduced for z if the variable $z \in X$ is contained in exactly one polynomial $p \in P$ and $\mathrm{lt}(p) = z$.

Lemma

Let P be a D-Gröbner basis with UMLT and let $p \in P$. Let $H = P$ be such that all polynomials $h \neq p \in H$ are D-reduced w.r.t. p. Then H is reduced for $\mathrm{lt}(p) = z$.

Theorem

Let $I \subseteq D[X]$ be an ideal. Let P be a D-Gröbner basis of I with UMLT which is reduced for z. Let $p \in P$ be the polynomial with $\mathrm{lt}(p) = z$. Then $P \setminus \{p\}$ is a D-Gröbner basis with UMLT for the ideal $J = I \cap D[X \setminus \{z\}]$.