An Algebraic Approach to Partial Synthesis of Arithmetic Circuits

Bhavani Sampathkumar, Ritaja Das, Bailey Martin, Florian Enescu, Priyank Kalla

Presenter: Priyank Kalla
Professor
Electrical & Computer Engineering
http://www.ece.utah.edu/~kalla

Introduction

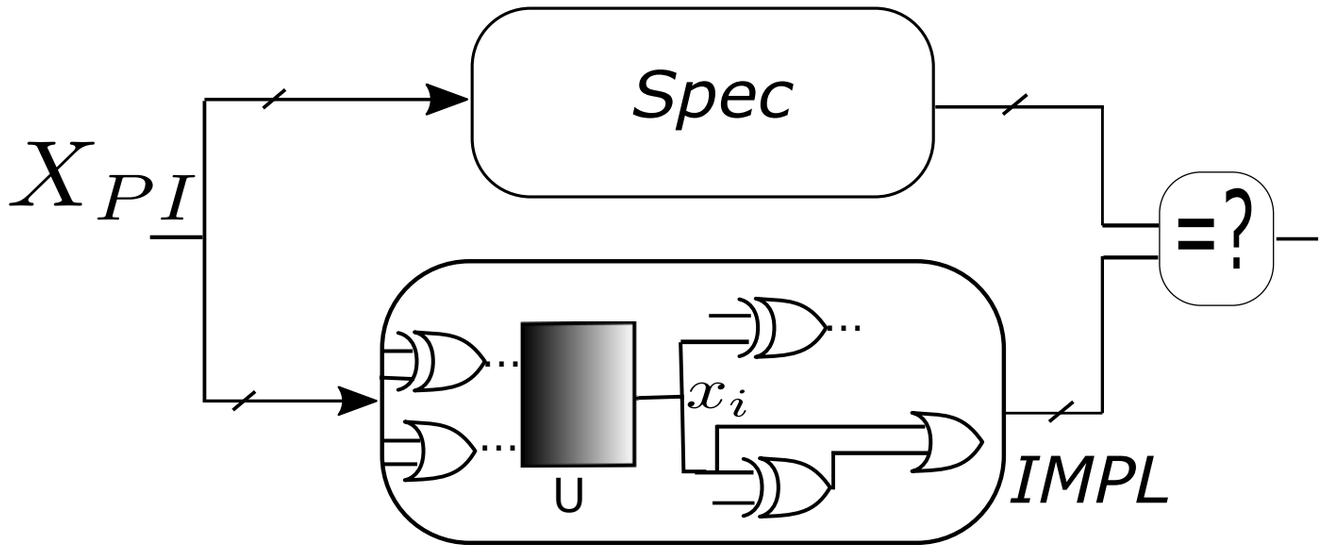
- Introduction to the Problem of Partial Logic Synthesis
  - Our focus: ***Integer Arithmetic Circuits***
  - Our approach uses a computer algebra model
  - Polynomial modeling of arithmetic circuits
  - Ideals, varieties and Gröbner bases (Buchberger's algorithm)
  - Verification and synthesis techniques
  - Experimental results
  - Conclusions and Future Work

The Foundation of our Work

- Arithmetic Circuits: Functions over k-bit vectors:
- k-bit vectors $\mapsto$ functions $f : \mathbb{B}^k \to \mathbb{B}^k$ [not efficient for arithmetic circuits]
- Model the circuit as polynomials over the quotient ring $R = \dfrac{\mathbb{Q}[x_1, \ldots, x_n]}{\langle x_1^2 - x_1, \ldots, x_n^2 - x_n \rangle}$
  - Algebraic Geometry results are valid over fields
    - Integers $\mathbb{Z} \subset$ rationals $\mathbb{Q}$
    - $\langle x_i^2 - x_i \rangle$: Impose Boolean idempotency at a polynomial level
- Represent circuits using ***polynomial ideals***, whose (zeros) ***varieties*** are the functions implemented by the circuits
- Algorithms make use of ideal membership using ***Gröbner bases***
- Go from polynomial ideals to Boolean functions, and employ conventional logic synthesis tools for optimizing circuits

Partial Logic Synthesis

- Given a Spec polynomial $f(x_1, \ldots, x_n) \in R = \frac{\mathbb{Q}[x_1, \ldots, _n]}{\langle x_1^2 - x_1, \ldots, x_n^2 - x_n \rangle}$, along with a partially completed circuit C.
- Does there exist a function U at some internal net of the circuit, s.t. C matches the *Spec* f?



- Mathematically, $\exists U \ \forall X_{PI}$ spec $f \equiv$ circuit C?
- There can be many such functions U, relationship to *don't cares and optimization*
- We focus on *Single-Fix Rectification* in this paper

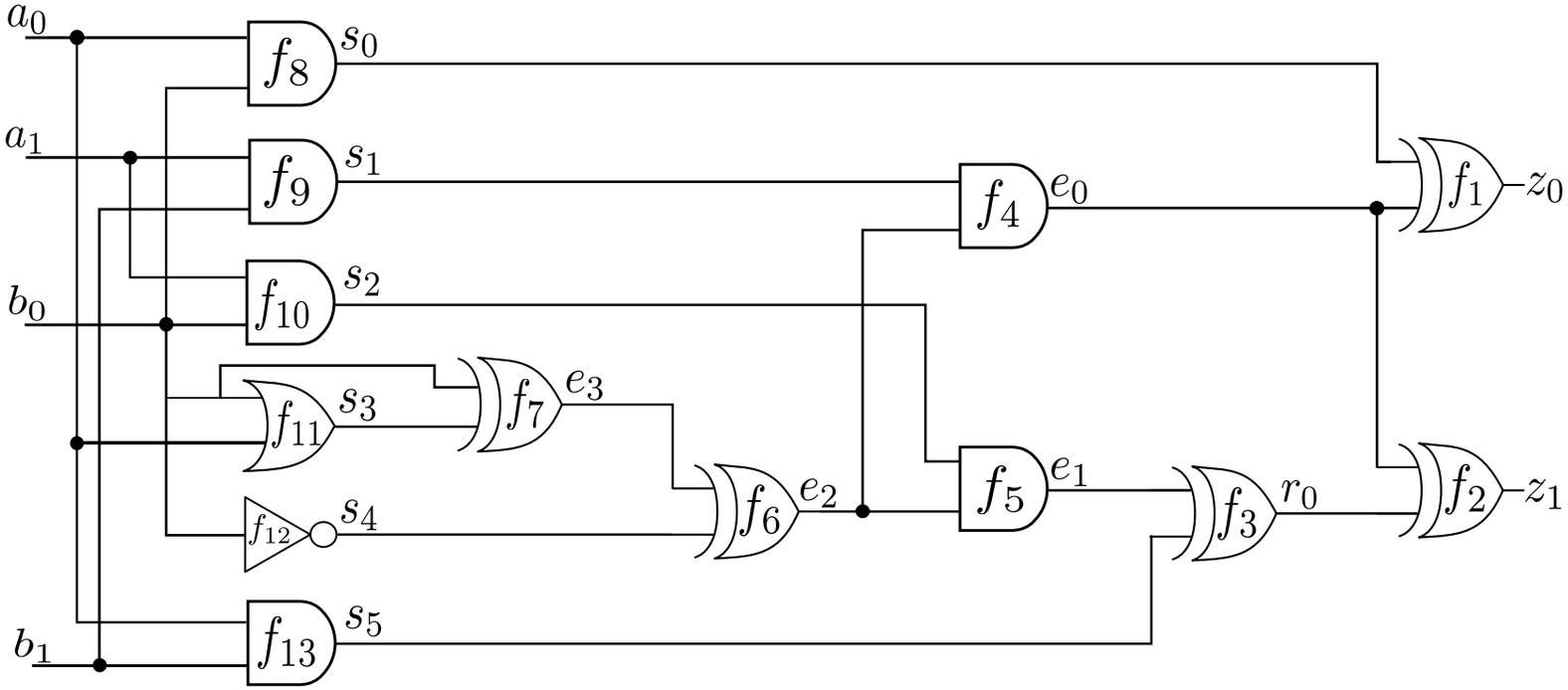
Polynomial Modeling of Arithmetic Circuits

- Given a ***Spec*** polynomial

- $$f_{spec} : z_0 + 2z_1 - 2a_0a_1b_0b_1 + 4a_0a_1b_1 - a_0b_0 - 2a_0b_1 + 4a_1b_0b_1 - 2a_1b_0 - 3a_1b_1$$

- Circuit ***Impl:*** Inputs $\{a_0, a_1\}, \{b_0, b_1\}$, Output $\{z_0, z_1\}$

- Integer value of the bit-vector $\{z_0, z_1\} : z_0 + 2z_1$



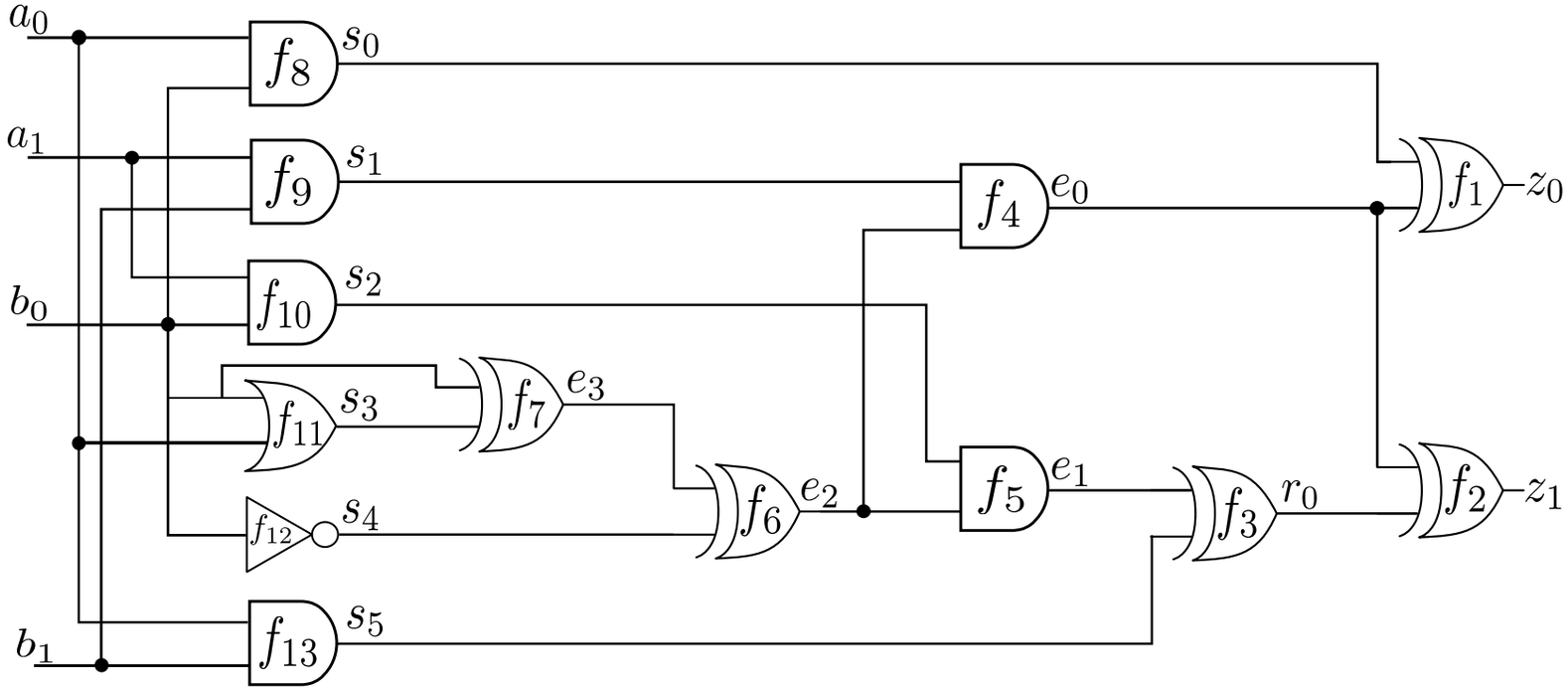
Polynomial Modeling of Arithmetic Circuits

- Given a ***Spec*** polynomial

- $f_{spec} : z_0 + 2z_1 - 2a_0a_1b_0b_1 + 4a_0a_1b_1 - a_0b_0 - 2a_0b_1 + 4a_1b_0b_1 - 2a_1b_0 - 3a_1b_1$

- Circuit ***Impl:*** Inputs $\{a_0, a_1\}, \{b_0, b_1\}$, Output $\{z_0, z_1\}$

- Integer value of the bit-vector $\{z_0, z_1\} : z_0 + 2z_1$

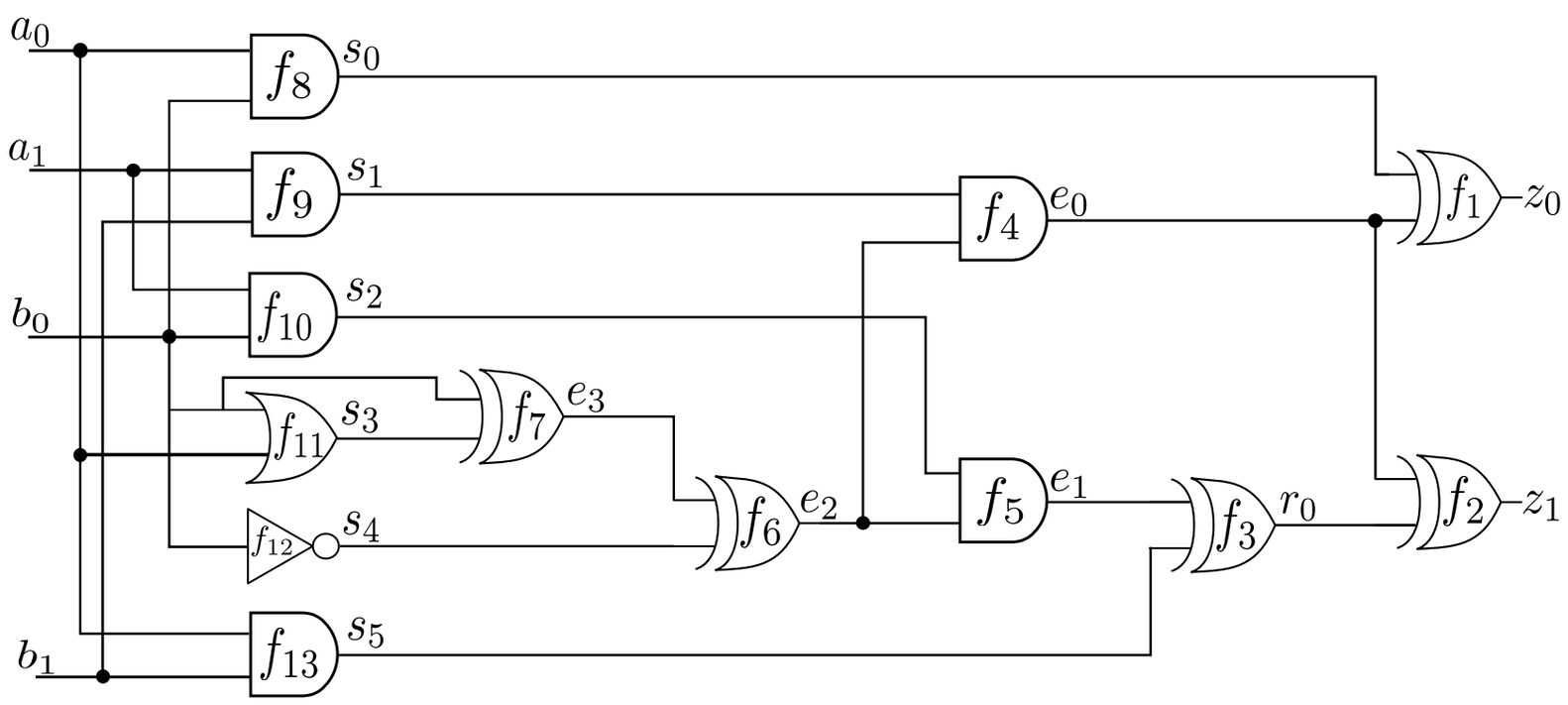


The circuit is actually buggy!! $C \not\equiv f_{spec}$

Application: Verification & Rectification

- $f_{spec} : z_0 + 2z_1 - 2a_0a_1b_0b_1 + 4a_0a_1b_1 - a_0b_0 - 2a_0b_1 + 4a_1b_0b_1 - 2a_1b_0 - 3a_1b_1$

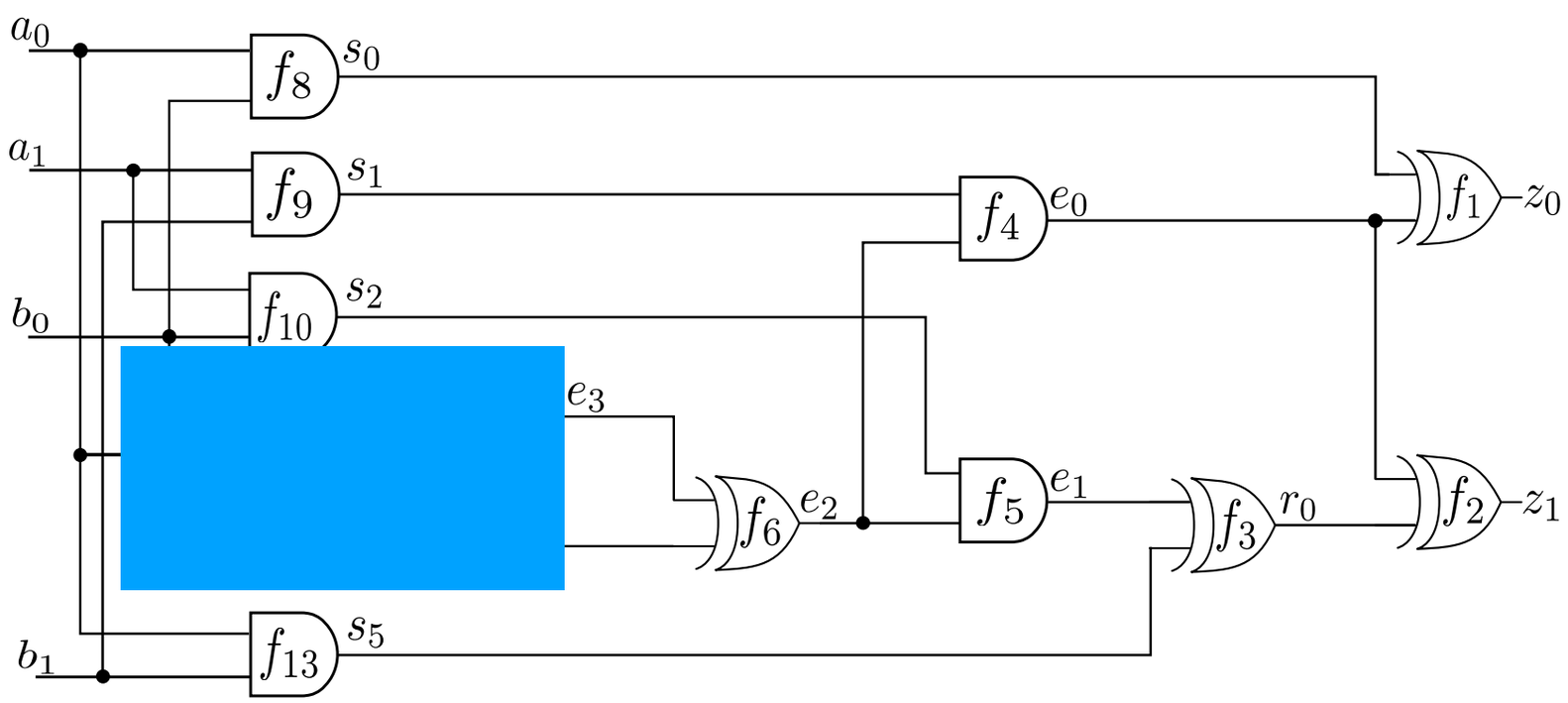
- First Verify, then Rectify!!



Application: Verification & Rectification

- $f_{spec} : z_0 + 2z_1 - 2a_0a_1b_0b_1 + 4a_0a_1b_1 - a_0b_0 - 2a_0b_1 + 4a_1b_0b_1 - 2a_1b_0 - 3a_1b_1$

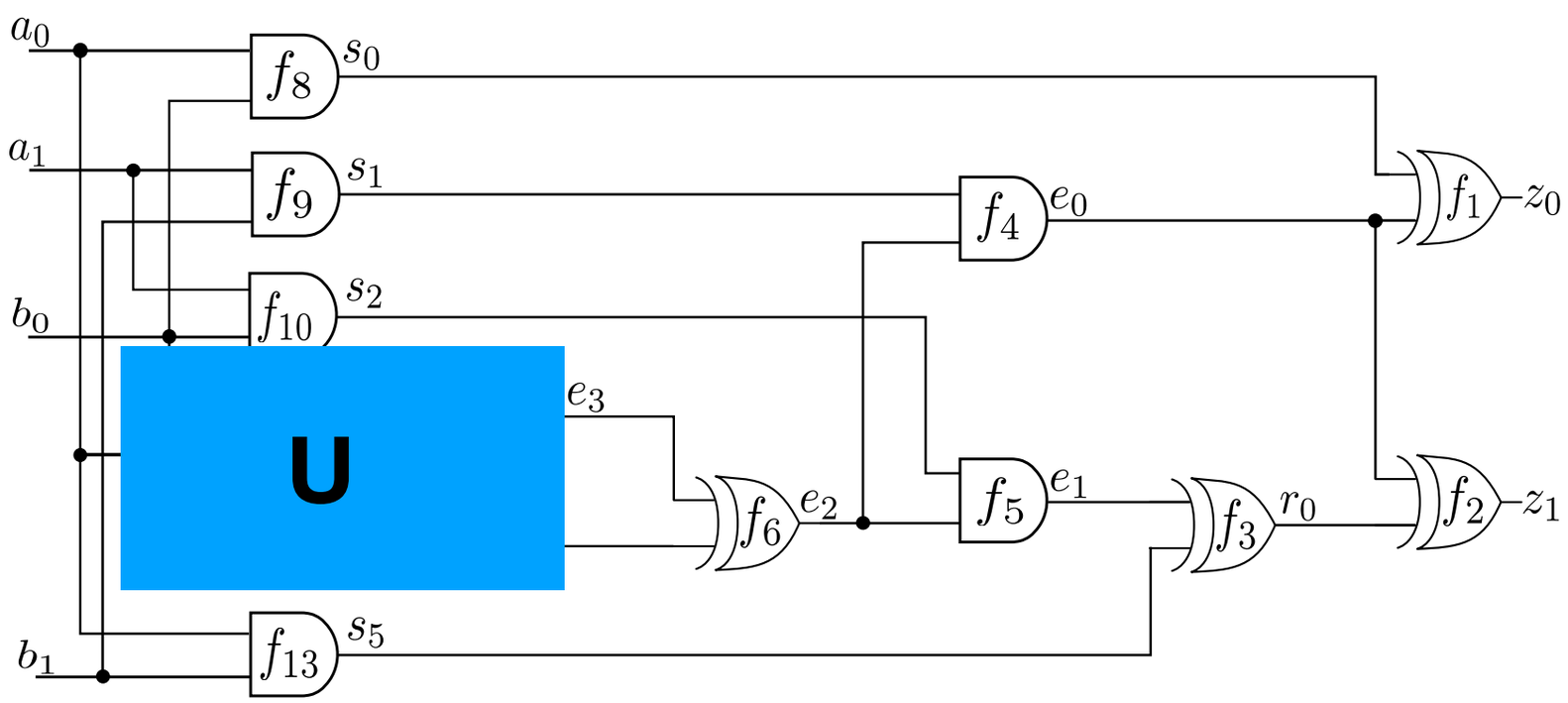
- First Verify, then Rectify!!



Application: Verification & Rectification

- $f_{spec} : z_0 + 2z_1 - 2a_0a_1b_0b_1 + 4a_0a_1b_1 - a_0b_0 - 2a_0b_1 + 4a_1b_0b_1 - 2a_1b_0 - 3a_1b_1$

- First Verify, then Rectify!!



Model Boolean Logic Gates as Polynomials

- NOT Gate:
  $u = \neg v$: $u = 1 - v$: or as polynomial $u - (1 - v)$

  - u, v, are Boolean: $u^2 - u, v^2 - v$

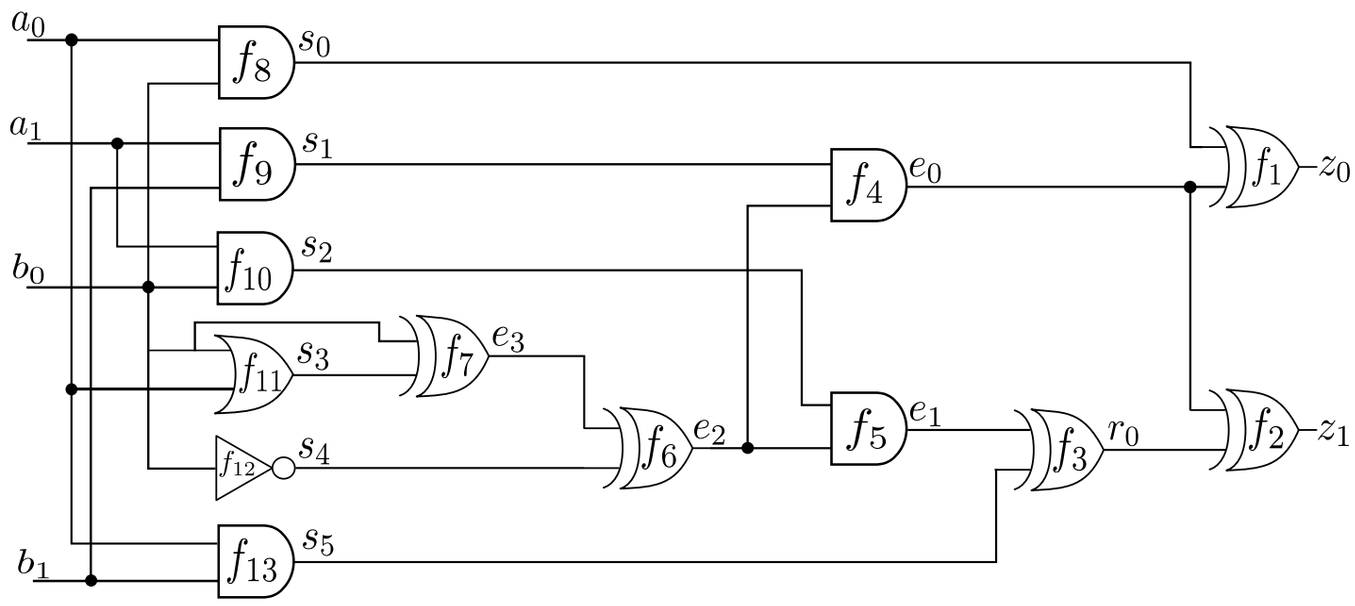
- AND gate $u = v \wedge w : u - v \cdot w,$

  - $u^2 - u, v^2 - v, w^2 - w$

- OR Gates: $u = v \wedge w \mapsto u - (v + w - vw)$

- XOR Gates: $u = v \oplus w \mapsto u - (w + w - 2vw)$

Polynomial Modeling of a Circuit



$$f_1 : z_0 - (s_0 + e_0 - 2s_0e_0); \qquad f_2 : z_1 - (e_0 + r_0 - 2 \cdot e_0 \cdot r_0);$$
$$f_3 : r_0 - (e_1 + s_5 - 2e_1s_5); \qquad f_4 : e_0 - (s_1 \cdot e_2);$$
$$f_5 : e_1 - (s_2 \cdot e_2); \qquad f_6 : e_2 - (e_3 + s_4 - 2e_3s_4);$$
$$f_7 : e_3 - (b_0 + s_3 - 2b_0s_3); \qquad f_8 : s_0 - (a_0 \cdot b_0);$$
$$f_9 : s_1 - (b_1 \cdot a_1); \qquad f_{10} : s_2 - (a_1 \cdot b_0);$$
$$f_{11} : s_3 - (a_0 + b_0 - a_0b_0); \qquad f_{12} : s_4 - (1 - b_0);$$
$$f_{13} : s_5 - (a_0b_1);$$

$$a_0^2 - a_0, a_1^2 - a_1, \ldots, e_0^2 - e_0, \ldots, z_0^2 - z_0, z_1^2 - z_1$$

Some Commutative Algebra

- We will model the circuit with a set of polynomials $F = \{f_1, \ldots, f_s\}$
- In verification, we need solutions to the system of equations:

$$f_1 = 0$$
$$f_2 = 0$$
$$\vdots$$
$$f_s = 0$$

  - Variety: Set of all solutions to a given system of polynomial equations: $V(f_1, \ldots, f_s)$
  - Variety depends on the ideal generated by the polynomials
  - Reason about the Variety by analyzing the Ideals

- Varieties = sets of points = (Boolean) functions in our setup

Some Commutative Algebra

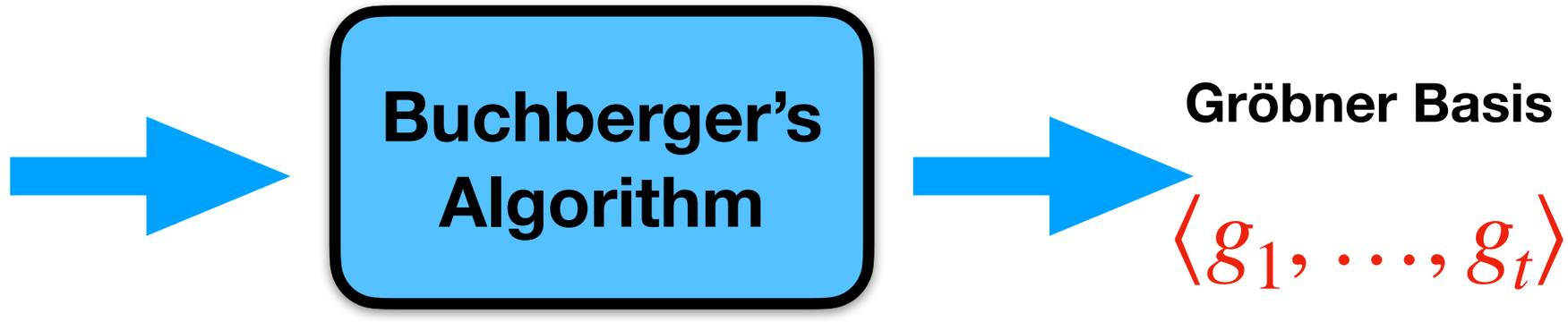
- Given a ring $R = \mathbb{Q}[X = x_1, \ldots, x_n]$, and a set of polynomials $\{f_1, \ldots, f_s\}$ the ideal $J = \langle f_1, \ldots, f_s \rangle \subseteq R$ is:
  - $J = \langle f_1, \ldots f_s \rangle = \{f_1 h_1 + f_2 h_2 + \ldots + f_s h_s : h_i \in R\}$
    - Then $J = \langle f_1, \ldots, f_s \rangle \subseteq R$ is an ***ideal*** generated by the polynomials, and $\{f_1, \ldots, f_s\}$ are the generators of J
- Also, if $f = f_1 h_1 + \ldots + f_s h_s$, then $f \in J$, i.e. f is a member of the ideal J
  - This is called ***ideal membership***

Ideal Membership Test Requires a Gröbner Basis

- An ideal has many generating sets of polynomials
- Ideal $J = \langle f_1, \ldots, f_s \rangle = \langle p_1, \ldots, p_r \rangle = \cdots = \langle g_1, \ldots, g_t \rangle$
- ***Gröbner Basis***: a canonical representation of the ideal, with special properties

Polynomials from the circuit

$\langle f_1, \ldots, f_s \rangle$



- Gröbner basis helps to solve Ideal Membership:
  - If $f \in \langle f_1, \ldots, f_s \rangle$, then Gröbner basis can give the relation:
  - $f = f_1 h_1 + f_2 h_2 + \ldots + f_s h_s$

Verification Problem Formulation

- Take the ***Impl*** circuit C, represent each gate with a polynomial in the ring $R = \mathbb{Q}[x_1, \ldots, x_n]$

- This gives a set of polynomials $\{f_1, \ldots, f_s\}$, which generates an ideal $J = \langle f_1, \ldots, f_s \rangle \subseteq R$

- For each variable (wire in the circuit) add the polynomials $\{x_i^2 - x_i\}\, \forall i$: generate ideal $J_0 = \langle x_i^2 - x_i \rangle$

- The function of the circuit is modeled by variety of ideal $J + J_0$, i.e. $V(J + J_0)$

- The Spec polynomial $f_{spec} \in R$

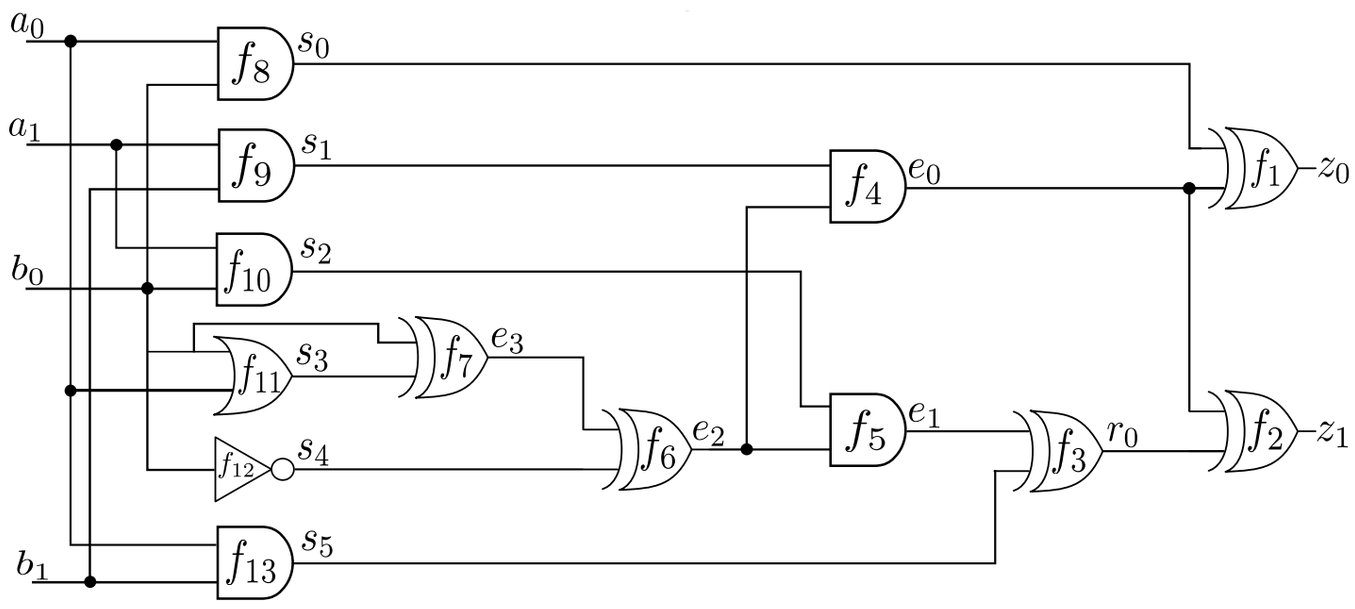
- The circuit C implements $f_{spec} \iff f_{spec} \in J + J_0$

Verification Problem Formulation

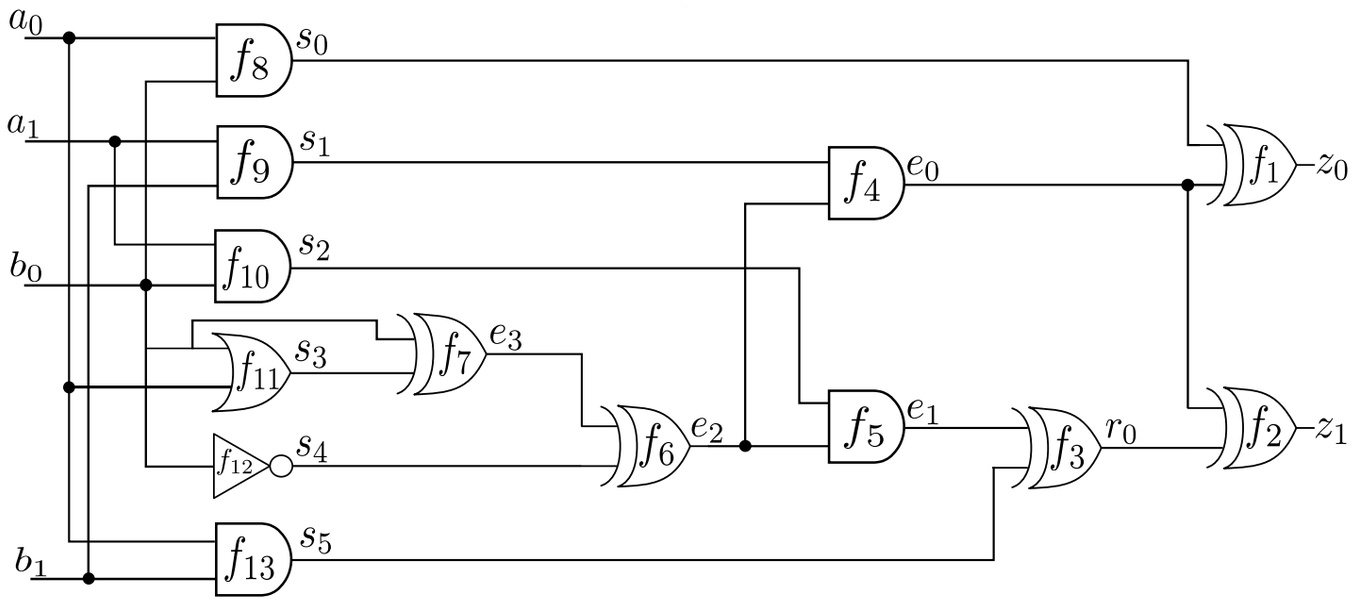
- The circuit C implements $f_{spec} \iff f_{spec} \in J + J_0$
- Formulate verification as ideal membership $f_{spec} \in J + J_0$?
  - Compute Gröbner basis $G = GB(J + J_0) = \{g_1, \ldots, g_t\}$
  - Compute a remainder of division by the GB $f_{spec} \xrightarrow{G=\{g_1,\ldots,g_t\}}_{+} r$ and see if $r = 0$?
    - Circuit C implements $f_{spec} \iff r = 0$
- However, computing a GB(J) is computationally infeasible for large circuits
- In our work, we use a trick: We derive a term order that makes the set of polynomials $\{f_1, \ldots, f_s, x_i^2 - x_i\}$ itself a $GB(J + J_0)$
  - So, no need to “compute” a GB, we already have it!!

Gröbner Bases depend on Term Orders

- Perform a reverse topological ordering of the variables of the circuit:
- $\{z_0 > z_1\} > \{r_0\} > \{e_0 > e_1\} > \{e_2\} > \{e_3\} > \{s_0 > s_1 > s_2 > s_3 > s_4 > s_5\} > \{a_0 > a_1 > b_0 > b_1\}$
- Using this variable order, impose a LEX order on the monomials of the circuit: RTTO $>$
- Under $>$ the polynomials are themselves a Gröbner basis (WHY?)
  - [Lv, Kalla, Enescu, TCAD'2013]



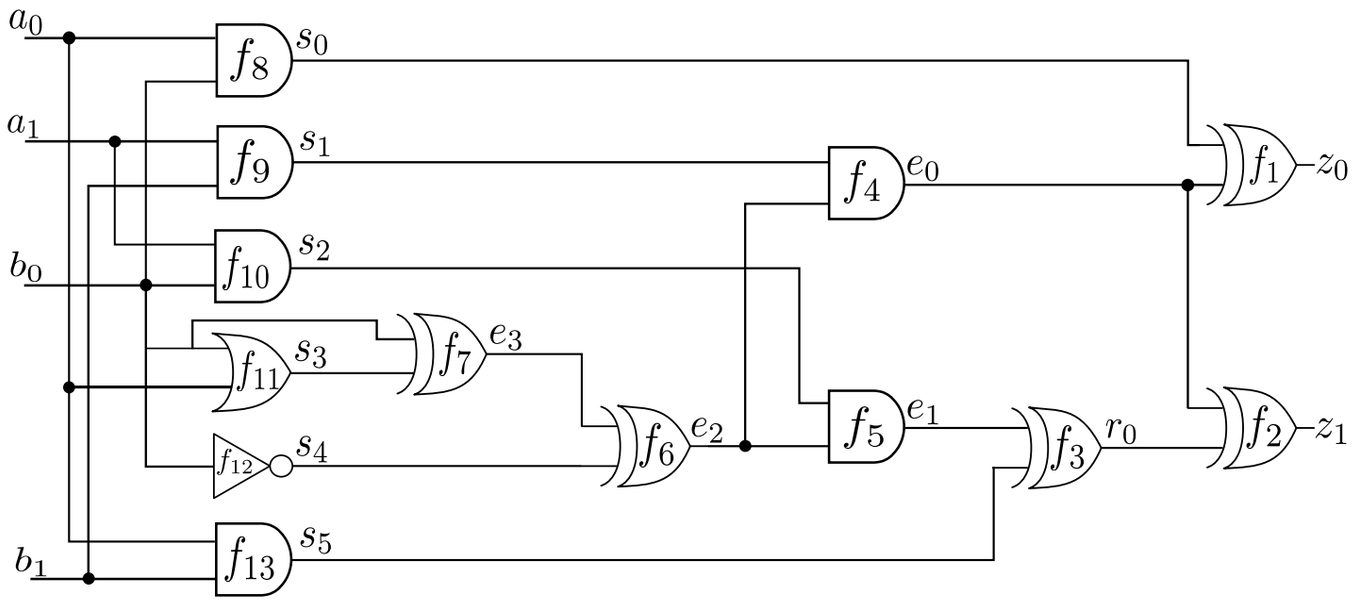
Polynomial Modeling of a Circuit



$$
\begin{aligned}
&f_1 : z_0 - (s_0 + e_0 - 2s_0e_0); && f_2 : z_1 - (e_0 + r_0 - 2 \cdot e_0 \cdot r_0);\\
&f_3 : r_0 - (e_1 + s_5 - 2e_1s_5); && f_4 : e_0 - (s_1 \cdot e_2);\\
&f_5 : e_1 - (s_2 \cdot e_2); && f_6 : e_2 - (e_3 + s_4 - 2e_3s_4);\\
&f_7 : e_3 - (b_0 + s_3 - 2b_0s_3); && f_8 : s_0 - (a_0 \cdot b_0);\\
&f_9 : s_1 - (b_1 \cdot a_1); && f_{10} : s_2 - (a_1 \cdot b_0);\\
&f_{11} : s_3 - (a_0 + b_0 - a_0b_0); && f_{12} : s_4 - (1 - b_0);\\
&f_{13} : s_5 - (a_0b_1);
\end{aligned}
$$

$$a_0^2 - a_0, a_1^2 - a_1, \ldots, e_0^2 - e_0, \ldots, z_0^2 - z_0, z_1^2 - z_1$$

Polynomial Modeling of a Circuit

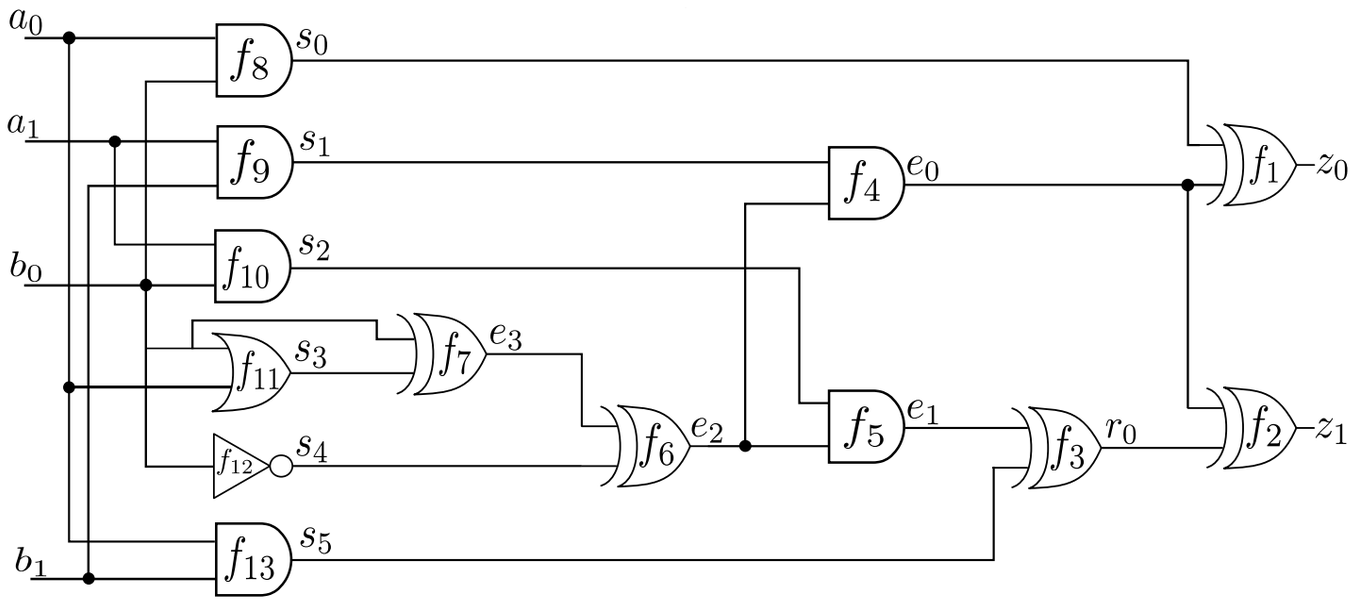


$$
\begin{aligned}
&f_1 : z_0 - (s_0 + e_0 - 2s_0e_0); && f_2 : z_1 - (e_0 + r_0 - 2 \cdot e_0 \cdot r_0);\\
&f_3 : r_0 - (e_1 + s_5 - 2e_1s_5); && f_4 : e_0 - (s_1 \cdot e_2);\\
&f_5 : e_1 - (s_2 \cdot e_2); && f_6 : e_2 - (e_3 + s_4 - 2e_3s_4);\\
&f_7 : e_3 - (b_0 + s_3 - 2b_0s_3); && f_8 : s_0 - (a_0 \cdot b_0);\\
&f_9 : s_1 - (b_1 \cdot a_1); && f_{10} : s_2 - (a_1 \cdot b_0);\\
&f_{11} : s_3 - (a_0 + b_0 - a_0b_0); && f_{12} : s_4 - (1 - b_0);\\
&f_{13} : s_5 - (a_0b_1);
\end{aligned}
$$

= Ideal J

$$a_0^2 - a_0, a_1^2 - a_1, \ldots, e_0^2 - e_0, \ldots, z_0^2 - z_0, z_1^2 - z_1$$

Polynomial Modeling of a Circuit



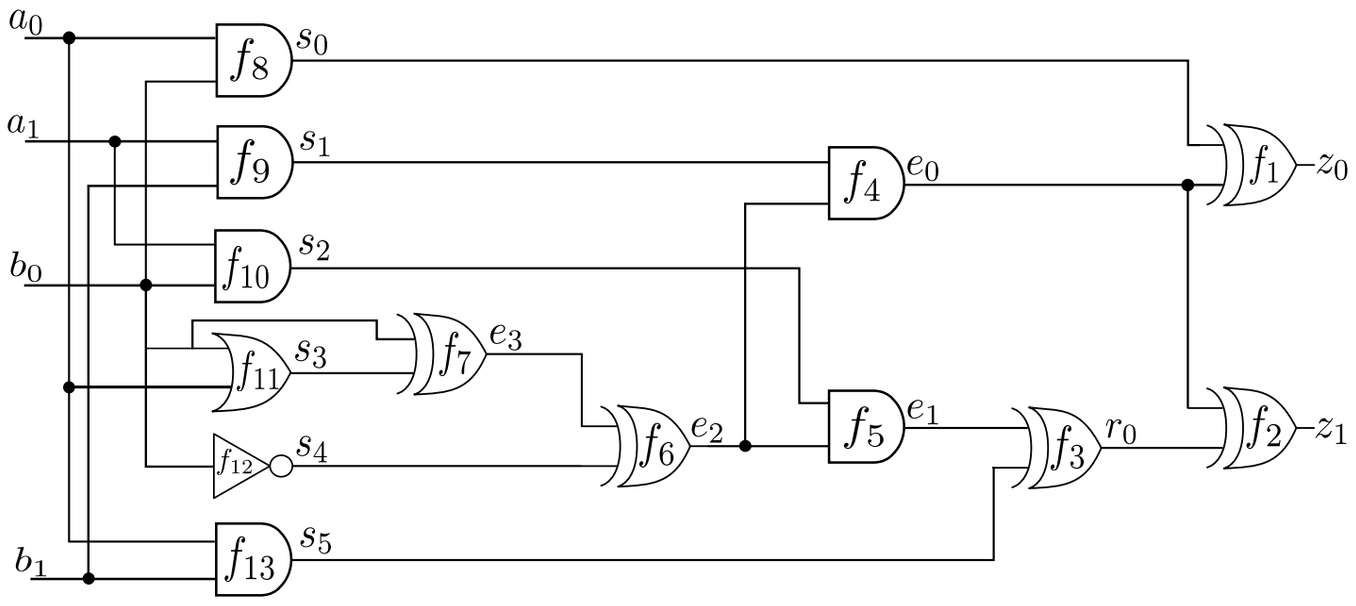
$$
\begin{aligned}
&f_1 : z_0 - (s_0 + e_0 - 2s_0e_0); && f_2 : z_1 - (e_0 + r_0 - 2 \cdot e_0 \cdot r_0);\\
&f_3 : r_0 - (e_1 + s_5 - 2e_1s_5); && f_4 : e_0 - (s_1 \cdot e_2);\\
&f_5 : e_1 - (s_2 \cdot e_2); && f_6 : e_2 - (e_3 + s_4 - 2e_3s_4);\\
&f_7 : e_3 - (b_0 + s_3 - 2b_0s_3); && f_8 : s_0 - (a_0 \cdot b_0);\\
&f_9 : s_1 - (b_1 \cdot a_1); && f_{10} : s_2 - (a_1 \cdot b_0);\\
&f_{11} : s_3 - (a_0 + b_0 - a_0b_0); && f_{12} : s_4 - (1 - b_0);\\
&f_{13} : s_5 - (a_0b_1);
\end{aligned}
$$

= Ideal J

$$a_0^2 - a_0, a_1^2 - a_1, \ldots, e_0^2 - e_0, \ldots, z_0^2 - z_0, z_1^2 - z_1$$

= Ideal J0

Polynomial Modeling of a Circuit



Ideal J + J0 is already a Gröbner Basis because of the term order

$$
\begin{aligned}
&f_1 : z_0 - (s_0 + e_0 - 2s_0e_0); && f_2 : z_1 - (e_0 + r_0 - 2 \cdot e_0 \cdot r_0);\\
&f_3 : r_0 - (e_1 + s_5 - 2e_1s_5); && f_4 : e_0 - (s_1 \cdot e_2);\\
&f_5 : e_1 - (s_2 \cdot e_2); && f_6 : e_2 - (e_3 + s_4 - 2e_3s_4);\\
&f_7 : e_3 - (b_0 + s_3 - 2b_0s_3); && f_8 : s_0 - (a_0 \cdot b_0);\\
&f_9 : s_1 - (b_1 \cdot a_1); && f_{10} : s_2 - (a_1 \cdot b_0);\\
&f_{11} : s_3 - (a_0 + b_0 - a_0b_0); && f_{12} : s_4 - (1 - b_0);\\
&f_{13} : s_5 - (a_0b_1);
\end{aligned}
$$

= Ideal J

$$a_0^2 - a_0, a_1^2 - a_1, \ldots, e_0^2 - e_0, \ldots, z_0^2 - z_0, z_1^2 - z_1$$

= Ideal J0

Verification for our Example

- In our example

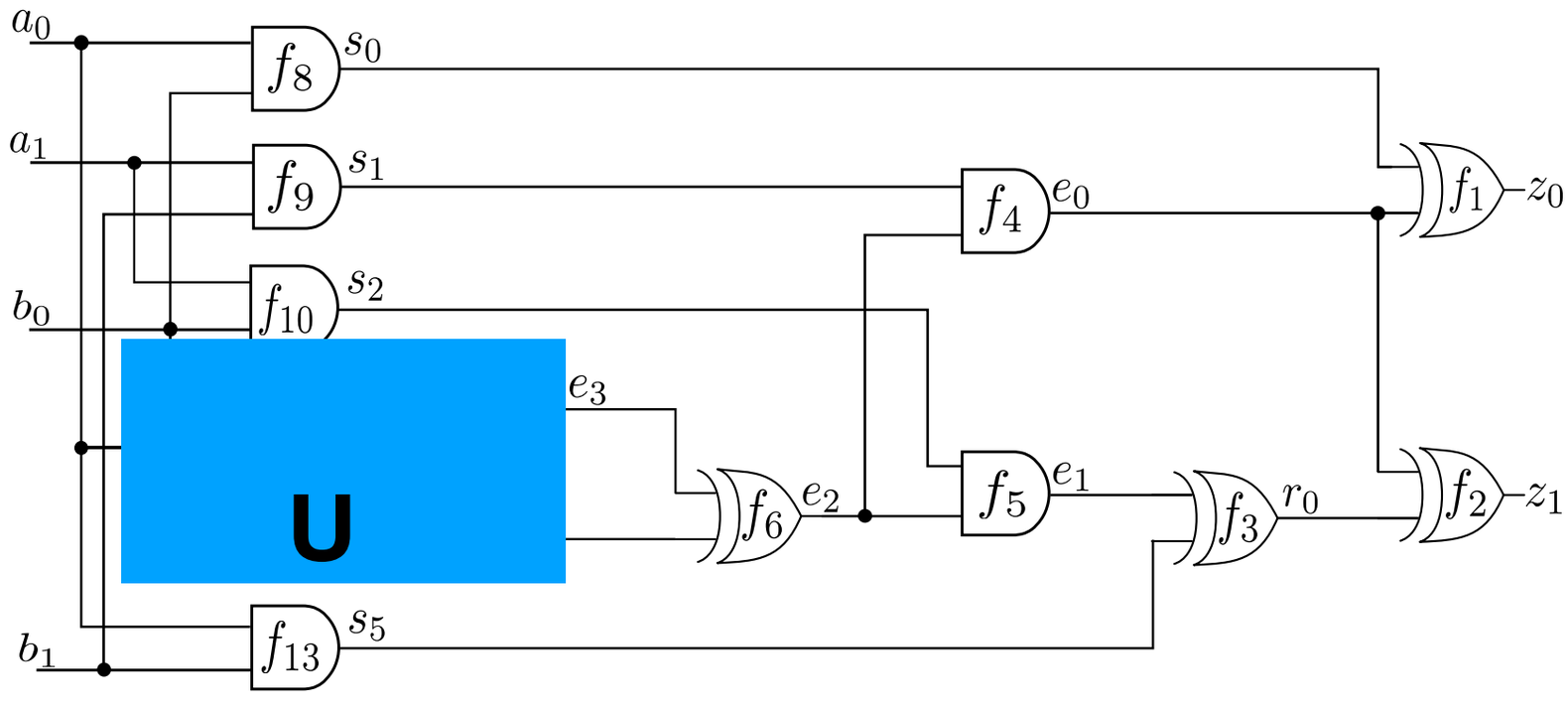
$$f_{spec} : z_0 + 2z_1 - 2a_0a_1b_0b_1 + 4a_0a_1b_1 - a_0b_0 - 2a_0b_1 + 4a_1b_0b_1 - 2a_1b_0 - 3a_1b_1$$

- $f_{spec} \xrightarrow{J+J_0}_{+} r = a_0a_1b_0b_1 + a_0a_1b_1 + a_1b_0b_1 - 2a_1b_0$

- Since $r \neq 0$, the circuit is buggy, and needs to be rectified!

Now Rectify the Circuit

- I want to check if the circuit can be rectified at net e_3



Rectification Check

- We're given $J + J_0 = \langle f_1, \ldots, f_i, \ldots, f_s, x_i^2 - x_i \rangle$
- Note leading term of $LT(f_i) = x_i$: rectification target
- Can I find a new polynomial $f_i : x_i - U,$ such that U patches the circuit?
- Create two ideals:
  - $J_L = \langle F_L \rangle = \{f_1, \ldots, f_{i-1}, \mathbf{f_i = x_i - 1}, f_{i+1}, \ldots, f_s\}$
  - $J_H = \langle F_H \rangle = \{f_1, \ldots, f_{i-1}, \mathbf{f_i = x_i - 0}, f_{i+1}, \ldots, f_s\}$
- Compute $f_{spec} \xrightarrow{J_L + J_0}_+ r_L$ and $f_{spec} \xrightarrow{J_H + J_0}_+ r_H$
- Circuit C can be rectified at the net $x_i \iff r_L \cdot r_H \xrightarrow{J_0}_+ 0$
- In our example, C can be rectified at e_3, e_2, but not at s_0

Compute a Rectification Function

- If rectification check passes at net x_i, it means **there exists** a function U s.t. $f_i : x_i = U$ rectifies the circuit
  - Polynomially, $f_i : x - U$, find U
- Verification check should pass $f_{spec} \in \langle f_1, \ldots, \mathbf{f_i : x_i - U}, \ldots, f_s \rangle + J_0$

$$f_{spec} = h_1 f_1 + h_2 f_2 + \cdots + \boldsymbol{h_i f_i} + \cdots + h_s f_s + \sum_{x_l \in X_{PI}} H_l \cdot (x_l^2 - x_l),$$

$$f_{spec} = h_1 f_1 + h_2 f_2 + \cdots + \boldsymbol{h_i(x_i - U)} + \cdots + h_s f_s + \sum_{x_l \in X_{PI}} H_l \cdot (x_l^2 - x_l)$$

$$f_{spec} - h_1 f_1 - h_2 f_2 - \cdots - h_{i-1} f_{i-1} - \boldsymbol{h_i x_i} = -\boldsymbol{h_i U} + h_{i+1} f_{i+1} \cdots + h_s f_s + \sum_{x_l \in X_{PI}} H_l \cdot (x_l^2 - x_l)$$

Computing a Rectification Function U

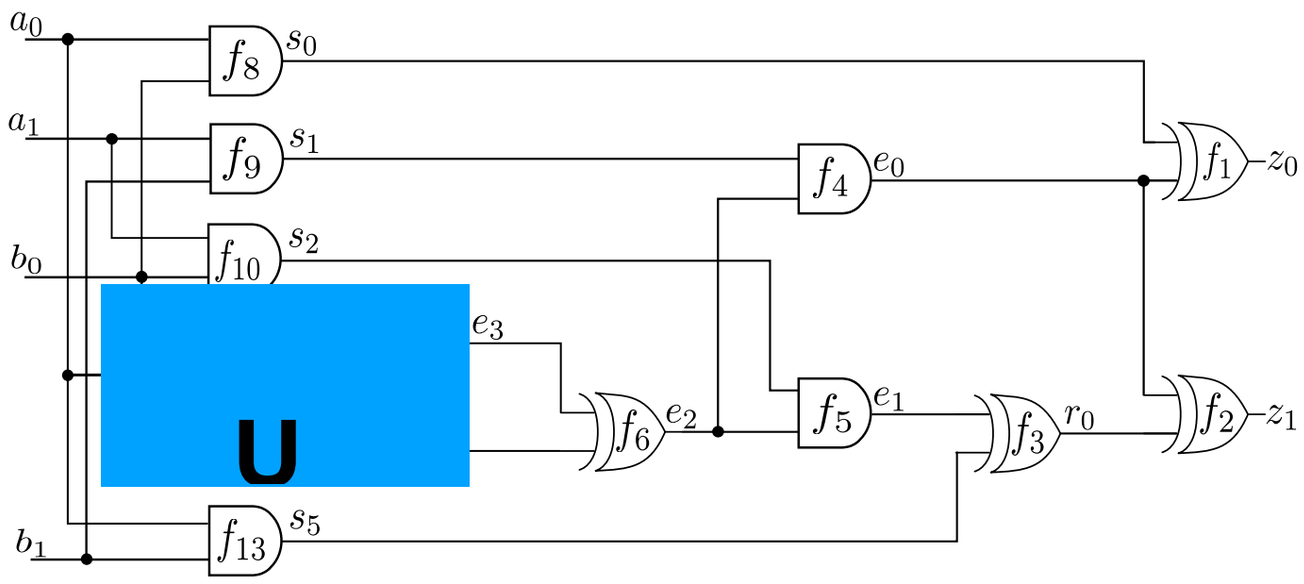
$$f_{spec} = h_1 f_1 + h_2 f_2 + \cdots + \boldsymbol{h_i(x_i - U)} + \cdots + h_s f_s$$
$$+ \sum_{x_l \in X_{PI}} H_l \cdot (x_l^2 - x_l)$$
$$f_{spec} - h_1 f_1 - h_2 f_2 - \cdots - h_{i-1} f_{i-1} \boldsymbol{- h_i x_i}$$
$$= \boldsymbol{-h_i U} + h_{i+1} f_{i+1} \cdots + h_s f_s + \sum_{x_l \in X_{PI}} H_l \cdot (x_l^2 - x_l)$$

$$f_{spec} - h_1 f_1 - \cdots - h_i x_i \quad \in \langle h_i, f_{i+1}, \ldots, f_s,\ x_l^2 - x_l \rangle,$$
$$r \quad \in \langle h_i, f_{i+1}, \ldots, f_s,\ x_l^2 - x_l \rangle,$$

- $r = h_i' h_i + h_{i+1}' f_{i+1} + \ldots + h_s' f_s + \sum_{x_l \in X_{PI}} H_l (x_l^2 - x_l)$

- We can use $h_i' = -U$ as the rectification function (polynomial!!)

In our Example Circuit…



- U is computed as a polynomial over $\mathbb{Q}[X]/X^2 - X$
- It may have rational coefficients, and it may evaluate to rational non-Boolean values
- For our example, the computed polynomials
  $h_i = -6a_0a_1b_0b_1 + 4a_0a_1b_1 + 2a_1b_0b_1 + 2a_1b_0 - 3a_1b_1$, and
- $U = h_i' = 56/5a_0b_0b_1 - 56/5a_0b_0 - 56/5a_0b_1 + 56/5a_0 + b_0$
- What does this mean? This is related to the “care-set” and the “Don’t care” set!

The Care-Set and the Don't Care Set of the Rectification Function

- Verification relation: $r = h_i' h_i + h_{i+1}' f_{i+1} + \ldots + h_s' f_s + \sum_{x_l \in X_{PI}} H_l(x_l^2 - x_l)$

- For a point a, $r(a) = h_i(a) \cdot h_i'(a) + \ldots + h_s'(a) f_s(a) + J_0(a)$, if $h_i(a) = 0$, $h_i'(a)$ can be anything

- The points a where $h_i(a) = 0$ are the "don't care" points, and the remaining points $h_i(a) \neq 0$ are the "care" points

a_0, a_1, b_0, b_1	h_i	h_i'	a_0, a_1, b_0, b_1	h_i	h_i'
0,0,0,0	0	0	1,0,0,0	0	$\frac{56}{5}$
0,0,0,1	0	0	1,0,0,1	0	0
0,0,1,0	0	1	1,0,1,0	0	1
0,0,1,1	0	1	1,0,1,1	0	1
0,1,0,0	0	0	1,1,0,0	0	$\frac{56}{5}$
0,1,0,1	-3	0	1,1,0,1	1	0
0,1,1,0	2	1	1,1,1,0	1	1
0,1,1,1	1	1	1,1,1,1	-1	1

Overall Approach

$$r = h_i' h_i + h_{i+1}' f_{i+1} + \ldots + h_s' f_s + \sum_{x_l \in X_{PI}} H_l(x_l^2 - x_l)$$

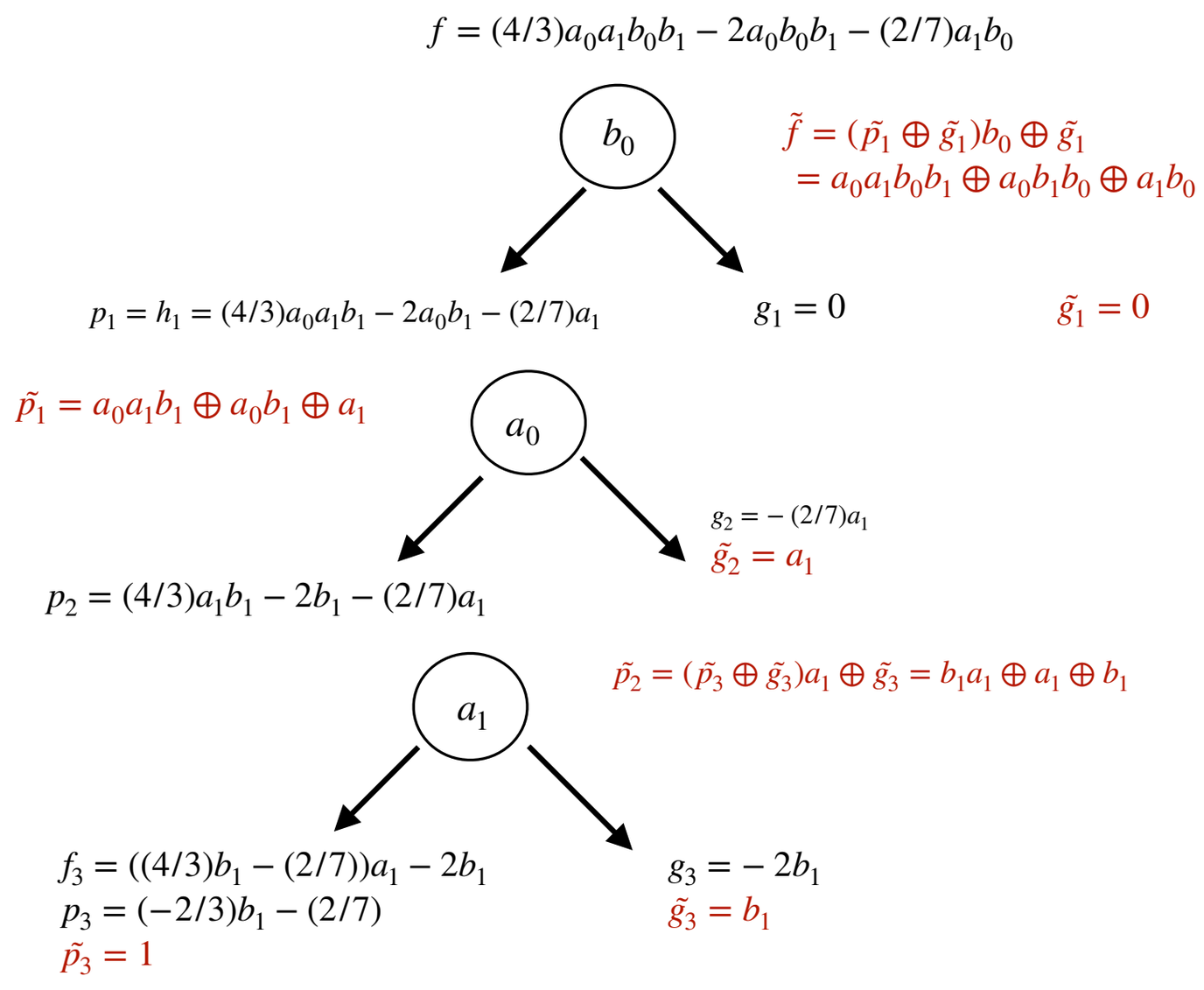
- Using the Division by Gröbner bases, compute two polynomials $h_i, U = h_i'$
- The zeros of h_i, i.e. the variety $V(h_i,\ x^2 - x) = \{a\}$ is the don't care set
- The remaining points are the care-set, where h_i' evaluates to 0 or 1
- Use a logic simplification tool to simplify the care-set w.r.t. the don't care set
- Practical challenge: we cannot "compute" the varieties for large functions

Symbolic Manipulation of Polynomials from $\mathbb{Q}[X]$ to $\mathbb{F}_2[X]$

- Given a polynomial $U \in \frac{\mathbb{Q}[X]}{X^2 - X}$
- Compute a polynomial $\widetilde{U} \in \mathbb{F}_2[X]$ such that U, $\widetilde{U}$ have the same zeros
  - $V_{\mathbb{Q}}(U, X^2 - X) = V_{\mathbb{F}_2}(\widetilde{U}, X^2 - X)$
- Since $\widetilde{U} \in \mathbb{F}_2[X]$ it only evaluates in $\{0,1\}$: Boolean
- $\mathbb{F}_2 \equiv \mathbb{B}$, where "+ = XOR" and " $\cdot$ = AND"
- So, $\widetilde{U}$ can be translated to Boolean functions
- $\widetilde{U}$ = Boolean function care set, $\widetilde{h_i}$ = Boolean function of the don't care set
- Use a Logic Synthesis tool to simplify $\widetilde{U}$ w.r.t. $\widetilde{h_i}$ to generate an optimized rectification patch function
- Refer to our paper [Intl. Symp. Multivalued Logic (ISMVL) 2023]

A Depiction of Boolean Translation

- $f = (4/3)a_0a_1b_0b_1 - 2a_0b_0b_1 - (2/7)a_1b_0$
- $\tilde{f} = a_0a_1b_0b_1 + a_0b_1b_0 + a_1b_0$



Experimental Results: Rectify Buggy Integer Multiplier Circuits

Benchmarks	*n*	*target location*	**Algebraic**					*SAT/CI*	**SPC Singular**		**SPC CI**	
			revsca	*amulet*	*RC*	*CPF*	*TT*	*TT*	*A*	*D*	*A*	*D*
SP-AR-RC	4	n41	0.02	0	0.04	0.06	0.1	39.70	3	2	3	2
	8	n38	0.02	0	0	0.05	0.05	0.15	1	1	1	1
	16	n156	0.04	0.02	0.04	0.06	0.12	39.06	7	4	2512	84
	32	n277	0.26	0.06	0	0.05	0.11	1332.11	14	7	775191	8280
	64	n279	TO	TO	NA	NA	NA	888.87	NA	NA	1	1
SP-WT-CL	4	n43	0	0	0.04	0.05	0.09	0.06	5	3	5	3
	8	n51	0.12	TO	0.19	0.18	0.49	210.9	7	4	1188753	44955
	16	n98	613.46	TO	0.06	6377.56	6991.08	TO	20	8	NA	NA
BP-AR-RC	4	n22	0.02	0	0.04	0.05	0.09	0.25	2	2	7	3
	8	n67	0.02	0	0	0.09	0.09	419.36	23	8	1858717	139738
	16	n72	0.08	0.02	0.04	TO	NA	TO	NA	NA	NA	NA
	32	n140	0.87	0.14	0.04	0.05	0.23	TO	13	9	NA	NA
	64	n337	15.63	0.44	0.02	0.23	0.69	TO	4	4	NA	NA
BP-WT-CL	4	n48	0.02	0	0	0.08	0.10	0.18	21	10	9	4
	8	n84	0.06	TO	0.05	0.06	0.17	357.49	4	4	1858717	139738
	16	n116	1359.18	TO	0.06	0.06	1359.3	TO	4	4	NA	NA

Experimental Results: Optimization of Integer Multipliers with Observability Don't Cares

- Compute ODCs at various nets, perform logic optimization
- SCA = synthesized circuit area, OCA = original circuit area

Benchmarks	n	*# targets*	*SCA*	*SCD*	*OCA*	*OCD*
SP-AR-RC	4	4	86	8	76	9
	8	3	323	18	386	23
	16	3	1324	44	1866	51
	32	4	5051	93	7704	107
SP-WT-CL	4	4	58	10	78	8
	8	4	421	17	467	14
	16	3	1618	23	2240	21
	16	4	1668	23	2240	21
BP-AR-RC	16	3	1269	37	1338	45
	32	4	4732	70	4912	89
BP-WT-CL	8	4	347	15	439	16
	16	4	1638	22	1789	22

Conclusion and Future Work

- Partial Logic Synthesis for arithmetic circuits
- Modeling of circuits using polynomial ideals in $\frac{\mathbb{Q}[X]}{X^2 - X}$
- Use Gröbner basis techniques to verify circuits and rectify them if they are buggy
- Compute rectification polynomials with rational coefficients and convert them to Boolean functions with the same zero-sets
- Perform Logic Optimization using care-set and don’t-care set
- We are now extending this work to multiple targets

Buchberger's Algorithm Computes a Gröbner Basis

Buchberger's Algorithm
INPUT : $F = \{f_1, \ldots, f_s\}$, and term order $>$
OUTPUT : $G = \{g_1, \ldots, g_t\}$
$G := F$;
REPEAT
  $G' := G$
  For each pair $\{f, g\}, f \neq g$ in G' DO
  $S(f, g) \xrightarrow{G'}_{+} r$
  IF $r \neq 0$ THEN $G := G \cup \{r\}$
UNTIL $G = G'$

$$S(f, g) = \frac{L}{lt(f)} \cdot f - \frac{L}{lt(g)} \cdot g$$

$L = \mathrm{LCM}(lm(f), lm(g))$, $lm(f)$: leading monomial of f