# RECTIFICATION OF INTEGER ARITHMETIC CIRCUITS

**Arpitha Srinath**

A Master's Thesis Defense

THE UNIVERSITY OF UTAH

Electrical and Computer Engineering
University of Utah

# Outline

- Problem Statement: Rectification of integer arithmetic circuits
  - Focus: Integer multiplier circuits
- Contributions
- Notation and Background
- Verification of arithmetic circuits
- Identifying potentially rectifiable nets
- Rectification check
- Computing rectification function

# Problem Statement

- Given a word-level specification polynomial $f \in \mathbb{Z}[x_1, \ldots, x_n]$.
- Given a buggy $k$-bit integer multiplier $C_1$.

Once the presence of bug(s) is detected:

- Identify potentially rectifiable nets.
- Identify single-fix rectification target net $x_i$.
- Compute single-fix rectification function $x_i = U(X_{PI})$.

# Problem Description

*Given specification* : $Z = A \cdot B$

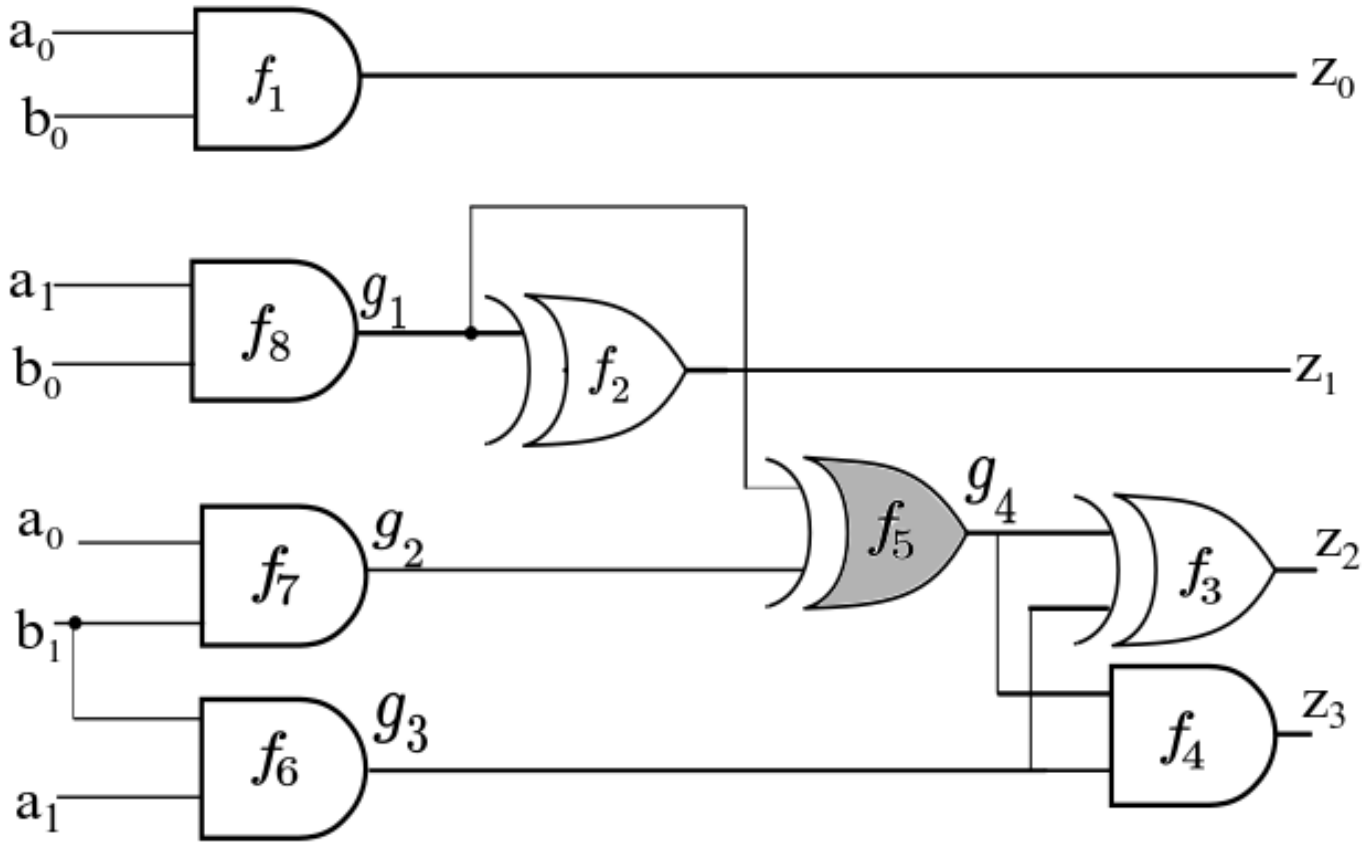



Figure: Buggy ($b$) 2-bit integer multiplier

# Contributions

- Identify the primary outputs affected by the bug(s).
- Identify a set of potentially rectifiable target nets.
- Ascertain whether or not a net admits single-fix rectification.
- Compute a rectification function.
- Synthesize rectification function and rectify the circuit.

# Notation and Background

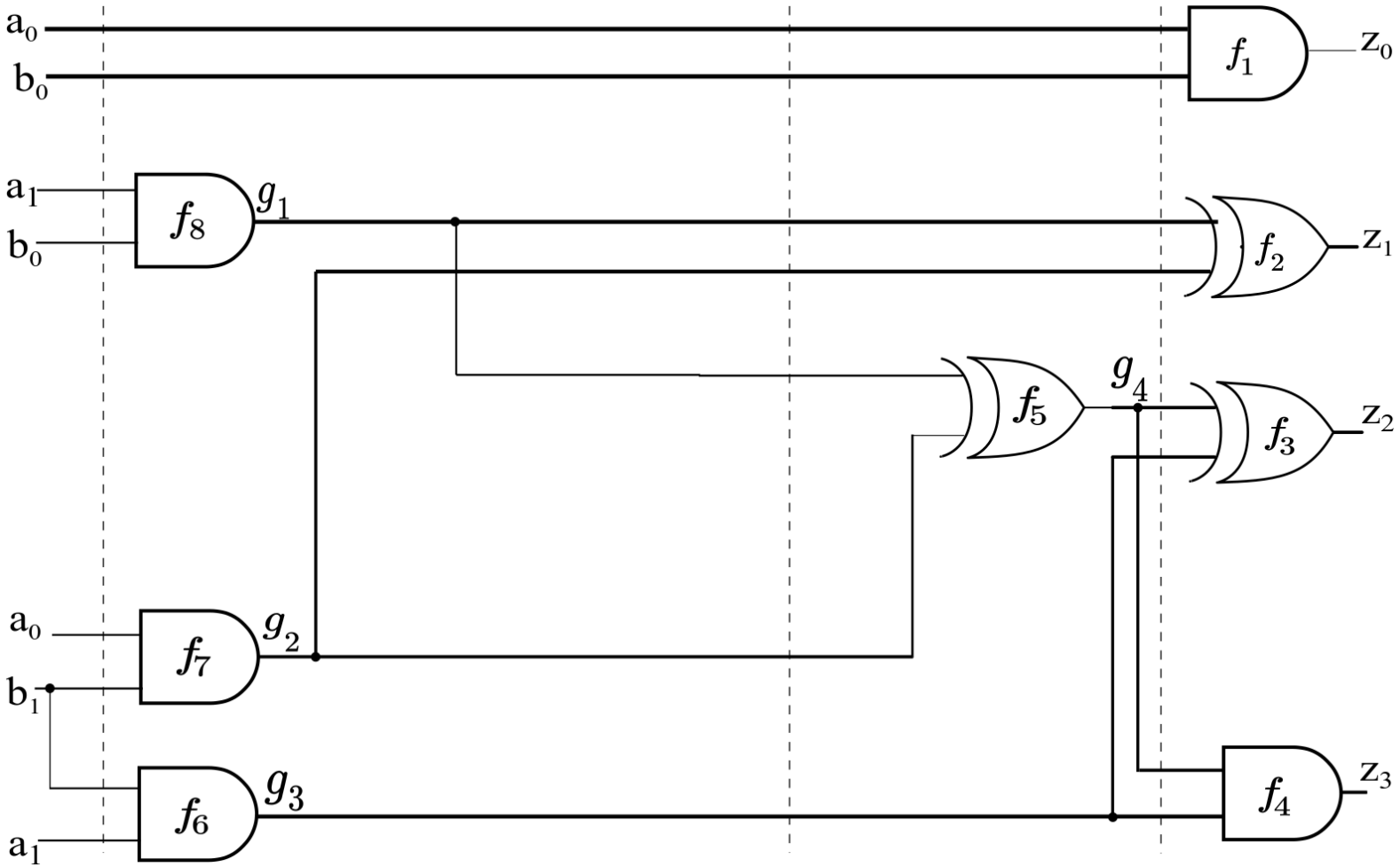
Model: Symbolic Computer Algebra

- Polynomial Ring $R = \mathbb{Q}[x_1, \ldots, x_n]$
- Polynomial $f \in R$, $f = c_1 X_1 + c_2 X_2 + \cdots + c_t X_t$
- $F = \{f_1, \ldots, f_s\} \in R$,
  Ideal $J = \langle F \rangle = \langle f_1, \ldots, f_s \rangle = \{\sum_{i=1}^{s} h_i \cdot f_i : \ h_i \in R\}$.
- Gröbner Basis (GB)
  - Canonical representation of the ideal
  - $f$ is reduced with respect an ideal: $f \xrightarrow{GB(J)}_{+} r$
  - Ideal membership test: $f \in J \iff f \xrightarrow{GB(J)} 0$.

# Notation and Background

- $F = \{f_1, \ldots, f_s\}$
  - $J = \langle f_1, \ldots, f_s \rangle$
  - Ideal generated by the circuit
- $F_0 = \{x_i^2 - x_i : i = 1, \ldots, n\}$
  - $J_0 = \langle F_0 \rangle$
  - Restrict solutions to Boolean values $\{0, 1\}$
- Reverse Topological Term Order (RTTO) [Lv. et al,TCAD 2013]
  - $\exists RTTO$ s.t. $\{F, F_0\} = GB(F, F_0)$

# Reverse Topological Term Order




**lex**: $\{a_0 < a_1 < b_0 < b_1\} < \{g_3 < g_2 < g_1\} < \{g_4\} < \{z_3 < z_3 < z_1 < z_1\}$

# Ring Model

From [D. Ritirc et al, FMCAD17]

- $R = \mathbb{Q}[x_1, \ldots, x_n]$
  - $\mathbb{Z} \subset \mathbb{Q}$
- $\{x_1, \ldots, x_n\}$ - nets from circuit
- Gates are modeled as polynomials as follows:
  - $u = \sim v \implies 0 = u - 1 + v$
  - $u = v \wedge w \implies 0 = u - vw$
  - $u = v \vee w \implies 0 = u - v - w + vw$
  - $u = v \oplus w \implies 0 = u - v - w + 2vw$
- System of polynomials - $F = \{f_1, \ldots, f_s\}$
- Set of vanishing polynomials - $F_0 = \{x_l^2 - x_l : l = 0, \ldots, n\}$
- Verification test: $f \xrightarrow{GB(F \cup F_0)} + r$

$$Ckt = Spec \iff r = 0$$

# Verification of arithmetic circuits

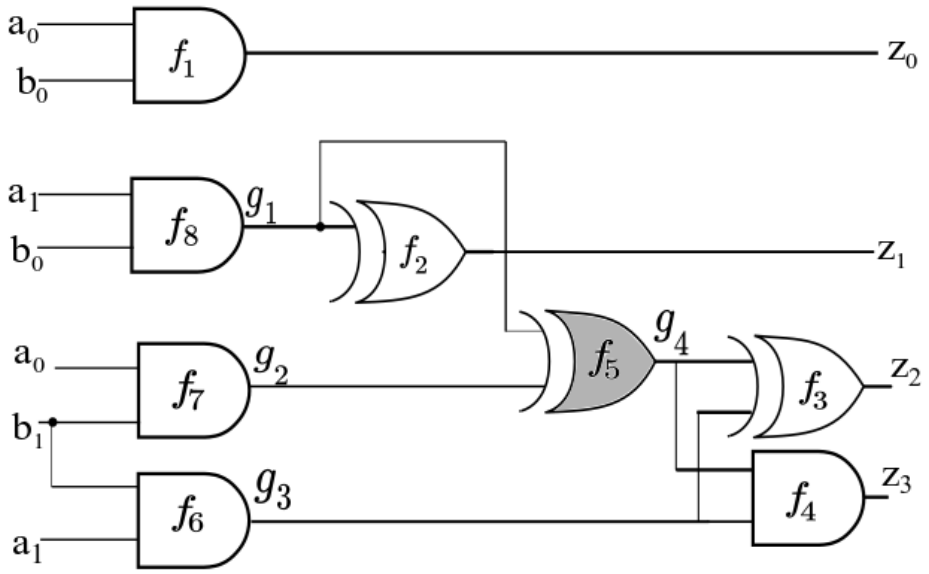



Figure: Buggy 2-bit integer multiplier

$$f_1 = z_0 - a_0 \cdot b_0$$
$$f_2 = z_1 - g_1 - g_2 + 2 \cdot g_1 \cdot g_2$$
$$f_3 = z_2 - g_3 - g_4 + 2 \cdot g_3 \cdot g_4$$
$$f_4 = z_3 - g_3 \cdot g_4$$
$$f_5 = g_4 - g_1 - g_2 + 2 \cdot g_1 \cdot g_2$$
$$f_6 = g_3 - a_1 \cdot b_1$$
$$f_7 = g_2 - a_0 \cdot b_1$$
$$f_8 = g_1 - b_0 \cdot a_1$$

# Verification of arithmetic circuits

- **RTTO:**
  Lexicographic order:
  $\{z_0 > z_1 > z_2 > z_3\} > \{g_4\} > \{g_3 > g_2 > g_1\} > \{a_0 > a_1 > b_0 > b_1\}$
- **Polynomial Ring:**
  $R = \mathbb{Q}[z_0, z_1, z_2, z_3, g_4, g_3, g_2, g_1, a_0, a_1, b_0, b_1]$
- **Specification polynomial:**
  $f : -(8 \cdot z_3 + 4 \cdot z_2 + 2 \cdot z_1 + z_0) + (2 \cdot a_1 + a_0) \cdot (2 \cdot b_1 + b_0)$

# Verification of arithmetic circuits

- $F = \{f_1, \ldots, f_8\}$
- $F_0 = \{x_1^2 - x_1, \ldots, x_n^2 - x_n\}$
- Result of verification:
  $f \xrightarrow{F \cup F_0} 12 \cdot a_0 \cdot a_1 \cdot b_0 \cdot b_1 - 4 \cdot a_0 \cdot b_1 - 4 \cdot a_1 \cdot b_0 \neq 0$

$$r \neq 0 \implies \textit{bug}!$$

# Identify Potentially Rectifiable Targets

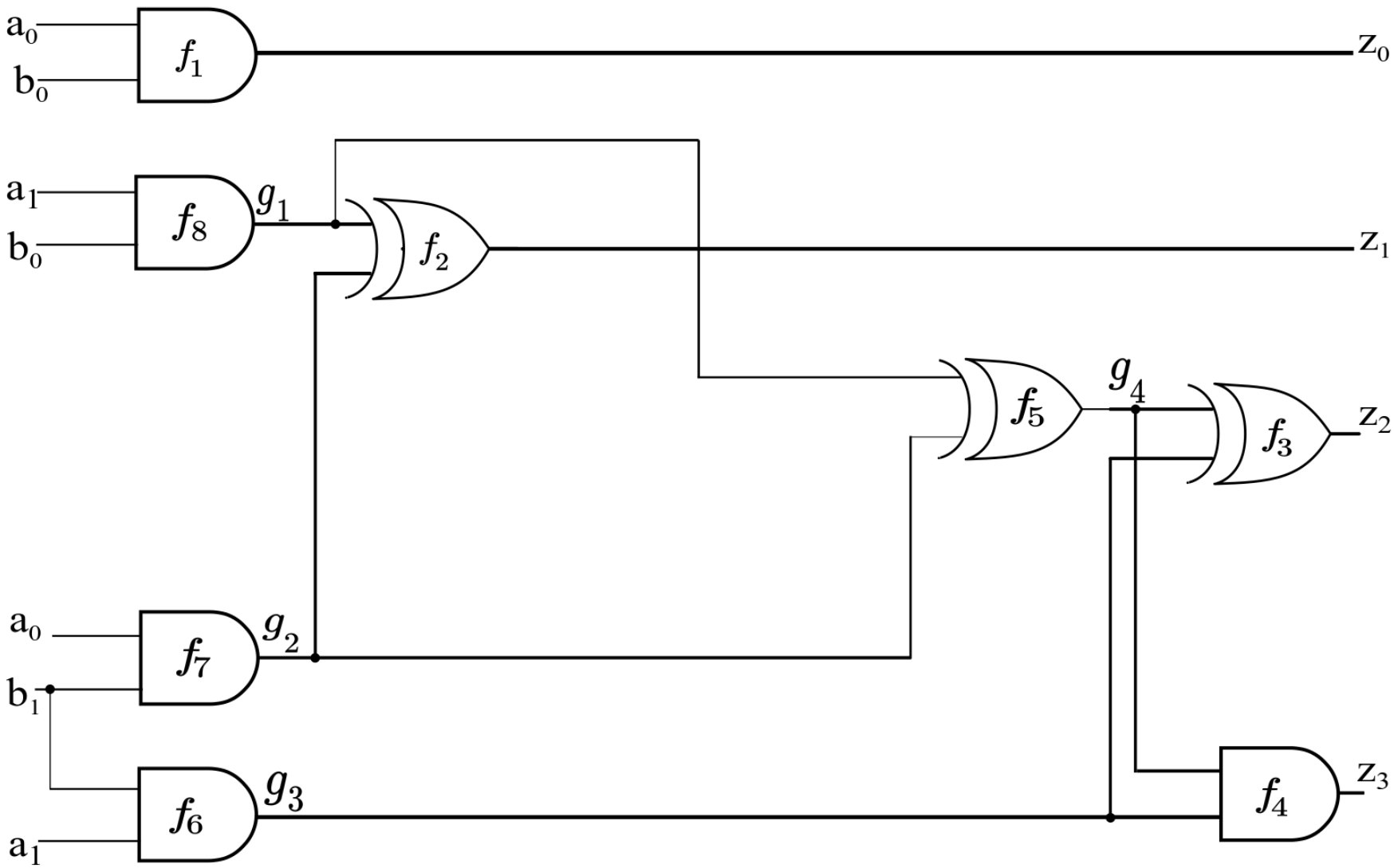



Figure: Buggy 2-bit integer multiplier

# Identify Potentially Rectifiable Targets

$$
\begin{aligned}
r &= 12 \cdot a_0 \cdot a_1 \cdot b_0 \cdot b_1 - 4 \cdot a_0 \cdot b_1 - 4 \cdot a_1 \cdot b_0 \\
&= 2^3(a_0 \cdot a_1 \cdot b_0 \cdot b_1 - 4 \cdot a_0 \cdot b_1) + 2^2(a_0 \cdot a_1 \cdot b_0 \cdot b_1 - a_0 \cdot b_1 - 4 \cdot a_1 \cdot b_0) \\
Z &= 2^3 z_3 + 2^2 z_2 + 2^1 z_1 + 2^0 z_0
\end{aligned}
$$

- Affected outputs: $z_2$ and $z_3$.

# Identify Potentially Rectifiable Targets

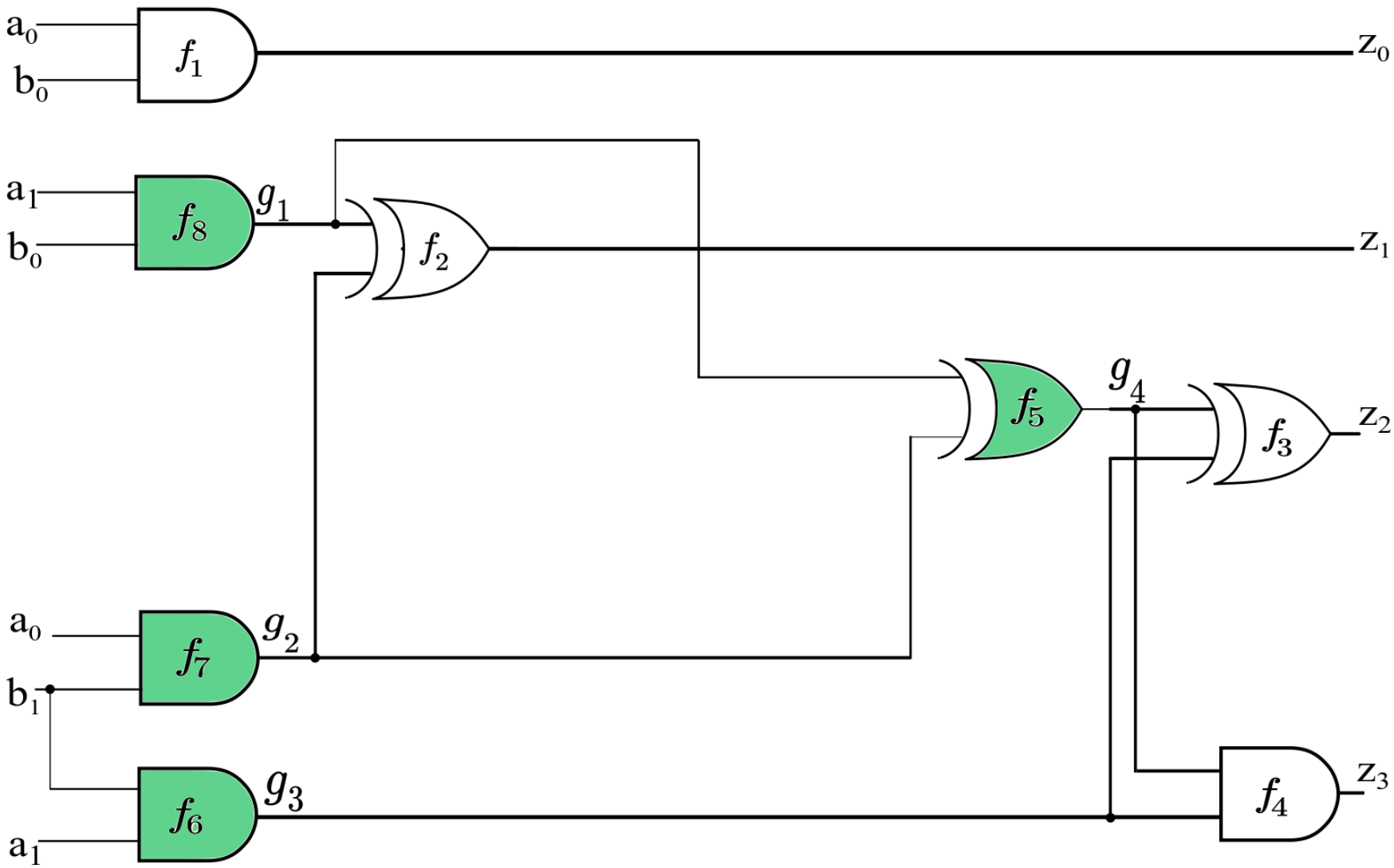



Figure: Intersection of fanin cones of $z_2$ and $z_3$.

# Single-fix rectification check

**Theorem 5.1**

- Single-fix rectification at net $x_i$:
  $J = \langle f_1, \ldots, f_s \rangle$
  $J_0 = \langle x_l^2 - x_l \rangle$
  1. Construct two ideals:
     - $J_L = \langle f_1, \ldots, f_i : x_i - 1, \ldots, f_s \rangle$
     - $J_H = \langle f_1, \ldots, f_i : x_i - 0, \ldots, f_s \rangle$
  2. Compute:
     - $f \xrightarrow{J_L + J_0} {}_+ r_L$
     - $f \xrightarrow{J_H + J_0} {}_+ r_H$
  3. Single-fix rectification possible *if and only if* $GB((r_L \cdot r_H) + J_0) = J_0$.

[Proved in thesis]

# Rectification check

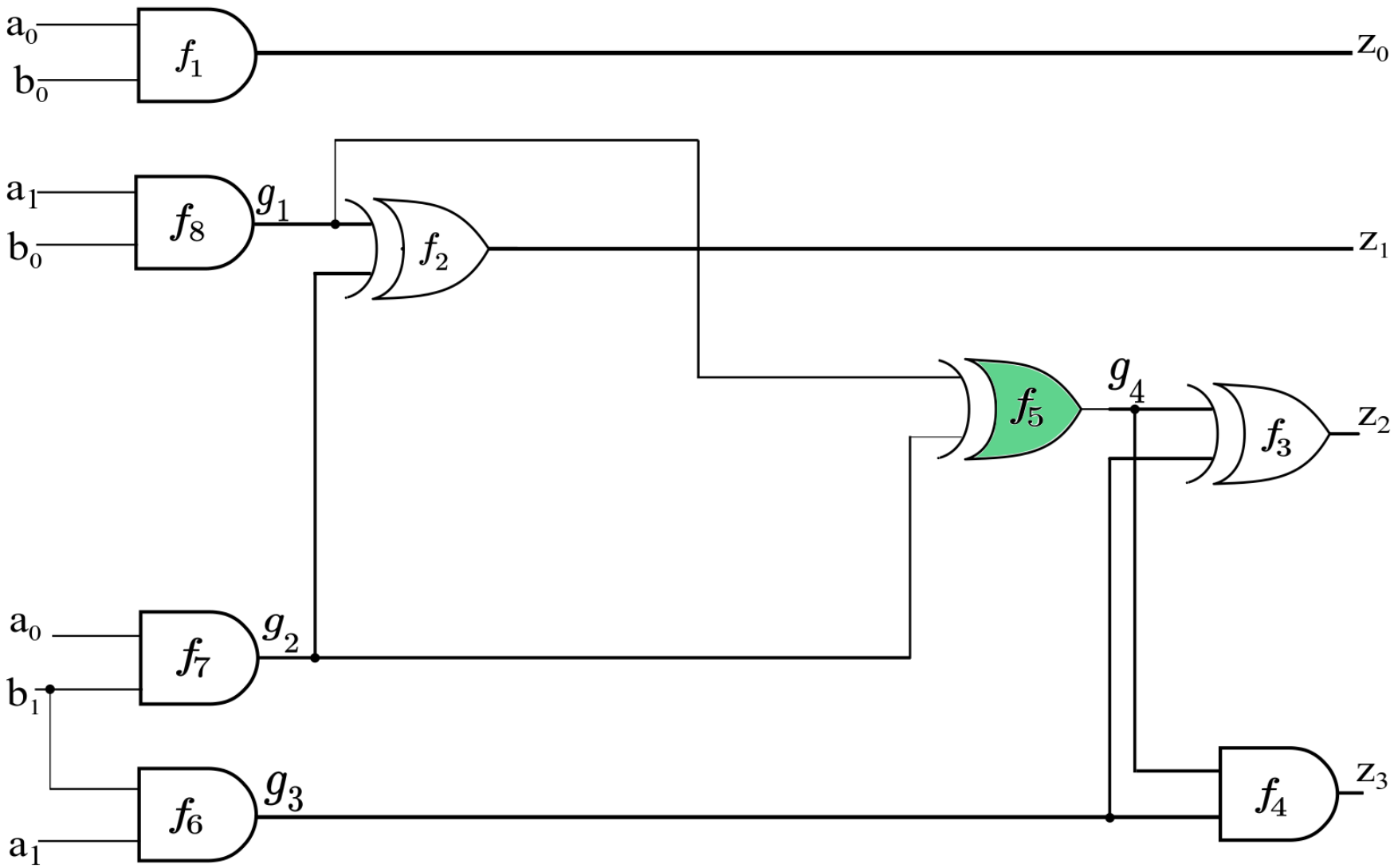



Figure: Rectification check at $g_4$

# Rectification check

Single-fix rectification check at net $g_4$:

- $J_L = \langle f_1, f_2, f_3, f_4, f_5 : g_4 - 1, f_6, f_7, f_8 \rangle$
- $J_H = \langle f_1, f_2, f_3, f_4, f_5 : g_4 - 0, f_6, f_7, f_8 \rangle$
- $f \xrightarrow{J_L + J_0}{}_+ r_L = 4 \cdot a_0 \cdot a_1 \cdot b_0 \cdot b_1 - 4$
- $f \xrightarrow{J_H + J_0}{}_+ r_H = 4 \cdot a_0 \cdot a_1 \cdot b_0 \cdot b_1$
- $GB((r_L \cdot r_H) + J_0) = \{a_0^2 - a_0, a_1^2 - a_1, b_0^2 - b_0, b_1^2 - b_1\}$

# Computing rectification function

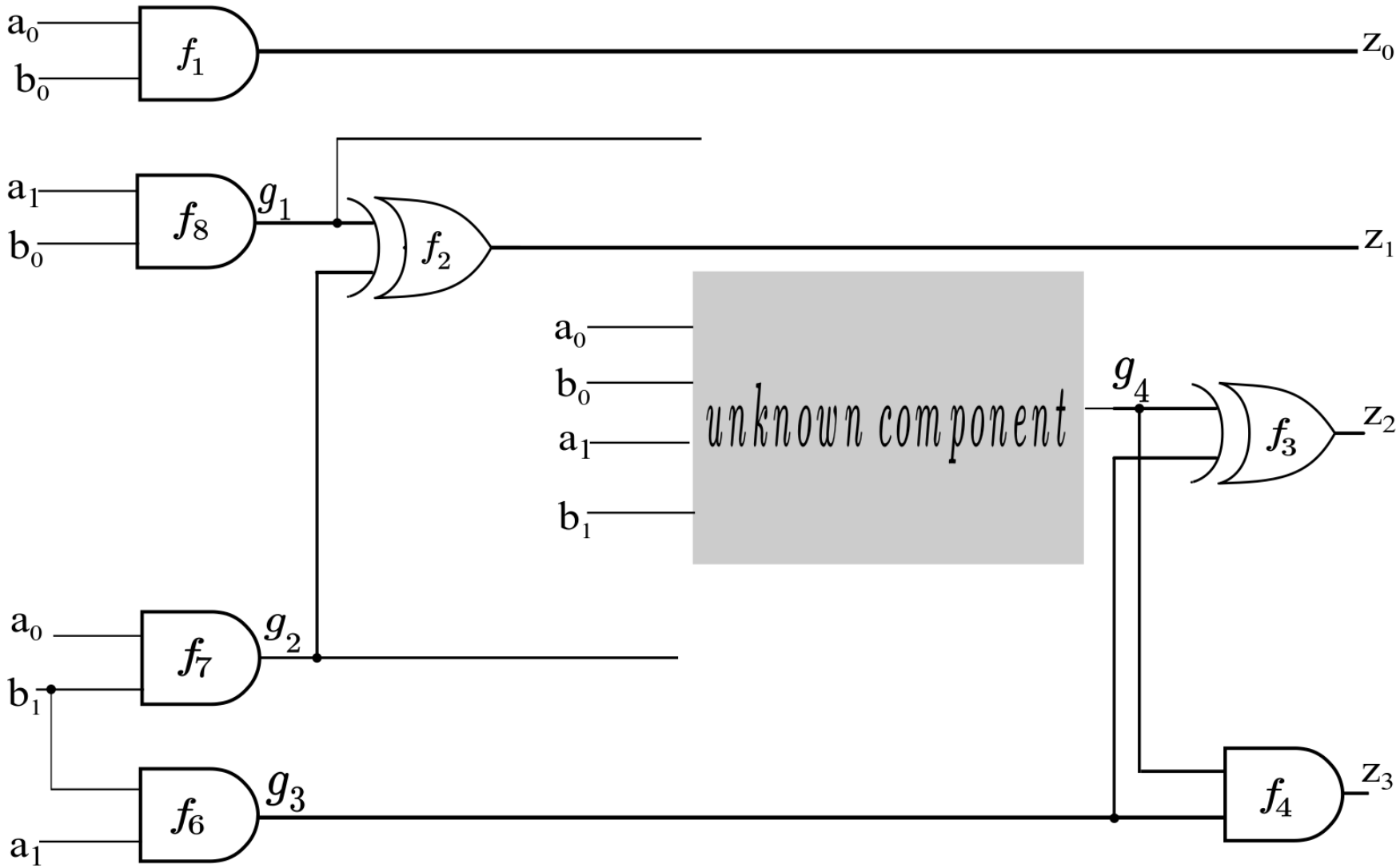



Figure: The Unknown Component Problem

# Computing rectification function

From [V.Rao et al, FMCAD 2018]

- $f \in \langle f_1, \ldots, f_i, \ldots, f_s \rangle$

# Computing rectification function

From [V.Rao et al, FMCAD 2018]

- $f \in \langle f_1, \ldots, f_i, \ldots, f_s \rangle$
- $f \in \langle f_1, \ldots, f_i : x_i - U, \ldots, f_s \rangle$

# Computing rectification function

From [V.Rao et al, FMCAD 2018]

- $f \in \langle f_1, \ldots, f_i, \ldots, f_s \rangle$
- $f \in \langle f_1, \ldots, f_i : x_i - U, \ldots, f_s \rangle$
- $f = h_1 f_1 + \cdots + h_{i-1} f_{i-1} + h_i x_i - h_i U + \cdots + h_s f_s$

# Computing rectification function

From [V.Rao et al, FMCAD 2018]

- $f \in \langle f_1, \ldots, f_i, \ldots, f_s \rangle$
- $f \in \langle f_1, \ldots, f_i : x_i - U, \ldots, f_s \rangle$
- $f = h_1 f_1 + \cdots + h_{i-1} f_{i-1} + h_i x_i - h_i U + \cdots + h_s f_s$
- Reduce $f$ in a specific order

# Computing rectification function

From [V.Rao et al, FMCAD 2018]

- $f \in \langle f_1, \ldots, f_i, \ldots, f_s \rangle$
- $f \in \langle f_1, \ldots, f_i : x_i - U, \ldots, f_s \rangle$
- $f = h_1 f_1 + \cdots + h_{i-1} f_{i-1} + h_i x_i - h_i U + \cdots + h_s f_s$
- Reduce $f$ in a specific order
- Record quotients and remainders

# Computing rectification function

From [V.Rao et al, FMCAD 2018]

- $f \in \langle f_1, \ldots, f_i, \ldots, f_s \rangle$
- $f \in \langle f_1, \ldots, f_i : x_i - U, \ldots, f_s \rangle$
- $f = h_1 f_1 + \cdots + h_{i-1} f_{i-1} + h_i x_i - h_i U + \cdots + h_s f_s$
- Reduce $f$ in a specific order
- Record quotients and remainders
- $h'_i$ computed as quotient of division

# Computing rectification function

From [V.Rao et al, FMCAD 2018]

- $f \in \langle f_1, \ldots, f_i, \ldots, f_s \rangle$
- $f \in \langle f_1, \ldots, f_i : x_i - U, \ldots, f_s \rangle$
- $f = h_1 f_1 + \cdots + h_{i-1} f_{i-1} + h_i x_i - h_i U + \cdots + h_s f_s$
- Reduce $f$ in a specific order
- Record quotients and remainders
- $h'_i$ computed as quotient of division
- $U = h'_i$ : Rectification polynomial

# Computing rectification function

- The rectification polynomial:
  $h'_i = U(X_{PI}) = a_0 \cdot a_1 \cdot b_0 \cdot b_1$
- This polynomial is equivalent to the function $a_0 \wedge a_1 \wedge b_0 \wedge b_1$

# Computing rectification function: Challenge

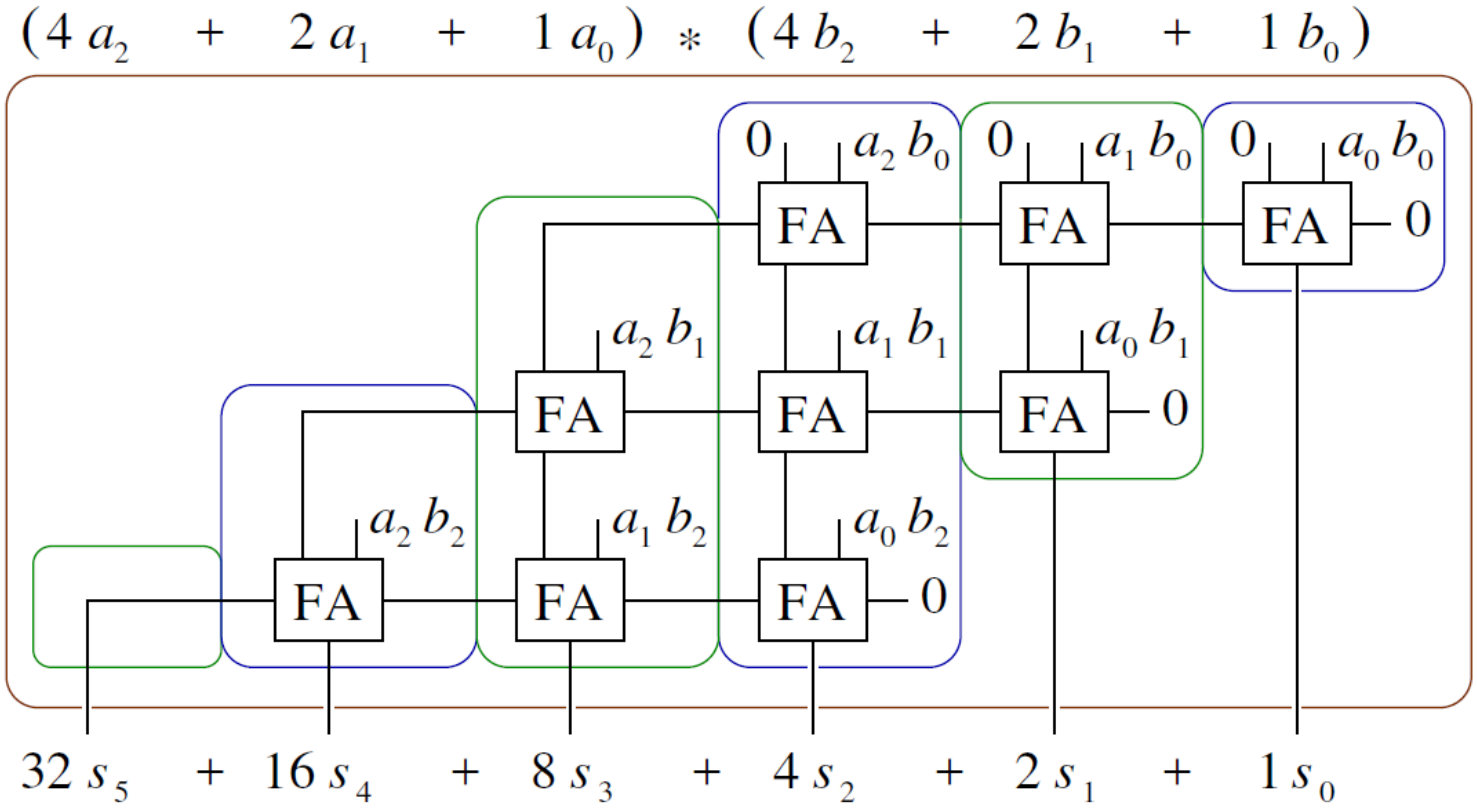



Figure: 3-bit multiplier from [D.Ritric et al, FMCAD2017].

- A bug is introduced a net $n_{38}$.
- Rectification attempted at net $n_{38}$.

# Computing rectification function : Challenge

$$n_{38} = n_{30} \cdot n_{31} - \frac{1}{2} \cdot n_{30} - \frac{1}{2} \cdot n_{31} + \frac{1}{2} \cdot n_{17} \cdot n_{22} - \frac{1}{4} \cdot n_{17} - \frac{1}{4} \cdot n_{22} -$$
$$2 \cdot n_{41} + 2 \cdot n_{33} \cdot n_{34} - n_{33} - n_{34} + \frac{1}{4} \cdot n_{14} \cdot n_{15} - \frac{1}{8} \cdot n_{14} -$$
$$\frac{1}{8} \cdot n_{15} + \frac{1}{8} \cdot a_0 \cdot b_1 + \frac{1}{4} \cdot a_0 \cdot b_2 + \frac{1}{8} \cdot a_1 \cdot b_0 + \frac{1}{4} \cdot a_1 \cdot b_1 + \frac{1}{2} \cdot a_1 \cdot b_2 +$$
$$\frac{1}{4} \cdot a_2 \cdot b_0 + \frac{1}{2} \cdot a_2 \cdot b_1 + a_2 \cdot b_2$$

- Contains fractional coefficients

# Computing rectification function: Challenge

$$n_{38} = n_{30} \cdot n_{31} - \frac{1}{2} \cdot n_{30} - \frac{1}{2} \cdot n_{31} + \frac{1}{2} \cdot n_{17} \cdot n_{22} - \frac{1}{4} \cdot n_{17} - \frac{1}{4} \cdot n_{22} -$$
$$2 \cdot n_{41} + 2 \cdot n_{33} \cdot n_{34} - n_{33} - n_{34} + \frac{1}{4} \cdot n_{14} \cdot n_{15} - \frac{1}{8} \cdot n_{14} -$$
$$\frac{1}{8} \cdot n_{15} + \frac{1}{8} \cdot a_0 \cdot b_1 + \frac{1}{4} \cdot a_0 \cdot b_2 + \frac{1}{8} \cdot a_1 \cdot b_0 + \frac{1}{4} \cdot a_1 \cdot b_1 + \frac{1}{2} \cdot a_1 \cdot b_2 +$$
$$\frac{1}{4} \cdot a_2 \cdot b_0 + \frac{1}{2} \cdot a_2 \cdot b_1 + a_2 \cdot b_2$$

- Contains variable $X_j \subset \{x_1, \ldots, x_n\}$.
- $X_j < x_i$ in RTTO.
- Maps from $\{0,1\}^{|X_j|} \rightarrow \{0,1\}$.

# Computing rectification function

- $U \xrightarrow{J+J_0} U(X_{PI})$
- $U(X_{PI})$ maps from $\{0,1\}^{|X_{PI}|} \rightarrow \{0,1\}$
- Monomials are multi-linear expression.
- Polynomial has integral coefficients.

$$\begin{aligned} n_{38} = & - a_0 \cdot a_1 \cdot a_2 \cdot b_0 \cdot b_1 \cdot b_2 + a_0 \cdot a_1 \cdot a_2 \cdot b_0 \cdot b_2 \\ & - a_0 \cdot a_1 \cdot a_2 \cdot b_1 \cdot b_2 + a_0 \cdot a_1 \cdot b_1 \cdot b_2 \\ & + a_0 \cdot a_2 \cdot b_0 \cdot b_1 \cdot b_2 - a_1 \cdot a_2 \cdot b_0 \cdot b_1 \cdot b_2 \\ & + a_1 \cdot a_2 \cdot b_1 \cdot b_2 \end{aligned}$$

# Computing rectification function: Synthesis

- Compute $U(X_{PI}) \pmod 2$.
- Interpret addition (+) as XOR operation and multiplication ($\cdot$) as AND operation.
- Rectification function in Reed-Muller's form.
- Synthesize AND-XOR expression as a logic circuit.

$$\begin{aligned} n_{38} = & a_0 \wedge a_1 \wedge a_2 \wedge b_0 \wedge b_1 \wedge b_2 \oplus a_0 \wedge a_1 \wedge a_2 \wedge b_0 \wedge b_2 \oplus a_0 \wedge a_1 \wedge a_2 \wedge b_1 \wedge b \\ & \oplus a_0 \wedge a_1 \wedge b_1 \wedge b_2 \oplus a_0 \wedge a_2 \wedge b_0 \wedge b_1 \wedge b_2 \oplus a_1 \wedge a_2 \wedge b_0 \wedge b_1 \wedge b_2 \\ & \oplus a_1 \wedge a_2 \wedge b_1 \wedge b_2 \end{aligned}$$

# Experimental Results

| $k$ | $t_1$ | $t_2$ | $t_3$ | $t_4$ |
|---|---|---|---|---|
| 2 | 0.002 | 0.003 | 0.007 | 0.005 |
| 4 | 0.005 | 5.820 | 0.024 | 5.824 |
| 8 | 0.027 | 9.103 | 0.206 | 9.111 |
| 12 | 0.120 | 23.137 | 0.641 | 23.158 |
| 16 | 0.400 | 42.915 | 1.782 | 42.981 |
| 18 | 0.647 | 48.964 | 2.479 | 49.060 |
| 28 | 6.329 | 288.448 | 26.707 | 289.860 |
| 32 | 12.119 | 368.319 | 44.965 | 370.579 |
| 56 | 292.203 | 980.283 | 1221.654 | 1040.504 |
| 64 | 577.162 | 16.049 | 28.597 | 20.147 |

Table: Single-fix rectification of integer multiplier against polynomial specification. Time in seconds; k = Datapath size, $t_1$ = verification time, $t_2$ = time to find potentially rectifiable nets , $t_3$ = time for rectification check, $t_4$ = time to compute rectification function. Time Out = 10800 seconds.

# Experimental Results

| *k* | | *area* | *delay* |
|---|---|---|---|
| 16 | correct | 1408.00 | 88.00 |
| | b1 | 3 | 2 |
| | b2 | 169.00 | 19.00 |
| | b3 | 200.00 | 18.00 |
| 32 | correct | 5888.00 | 328.00 |
| | b1 | 3 | 2 |
| | b2 | 62.00 | 10.00 |
| | b3 | 152.00 | 14.00 |
| 56 | correct | 18368.00 | 184.00 |
| | b1 | 3 | 2 |
| | b2 | 312.00 | 19.00 |
| | b3 | TO(*computation*) | TO(*computation*) |

Table: Synthesis Results. *area* in sq. units. *delay* = No. of gates.

# Limitations

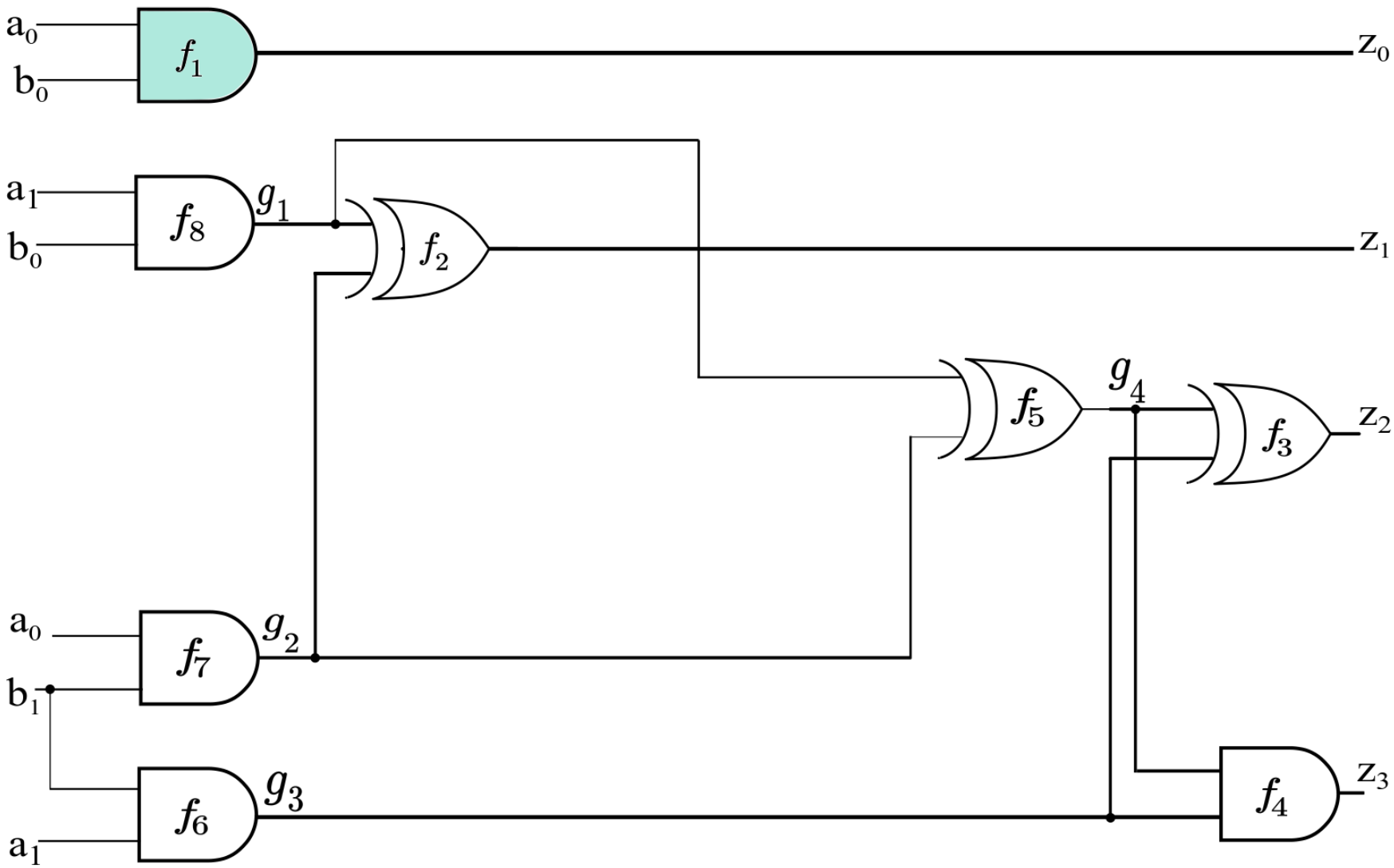
- Focus on single-fix rectification only.
- Practical only for array multipliers.
  - Booth multiplier: TO at 8-bits.
- Efficiency depends on location of bug.
- Rectification polynomial computed in primary inputs only.
- Computation of a single rectification function
  - Many functions may exist

# Conclusion

Once bug(s) are detected in the circuit:

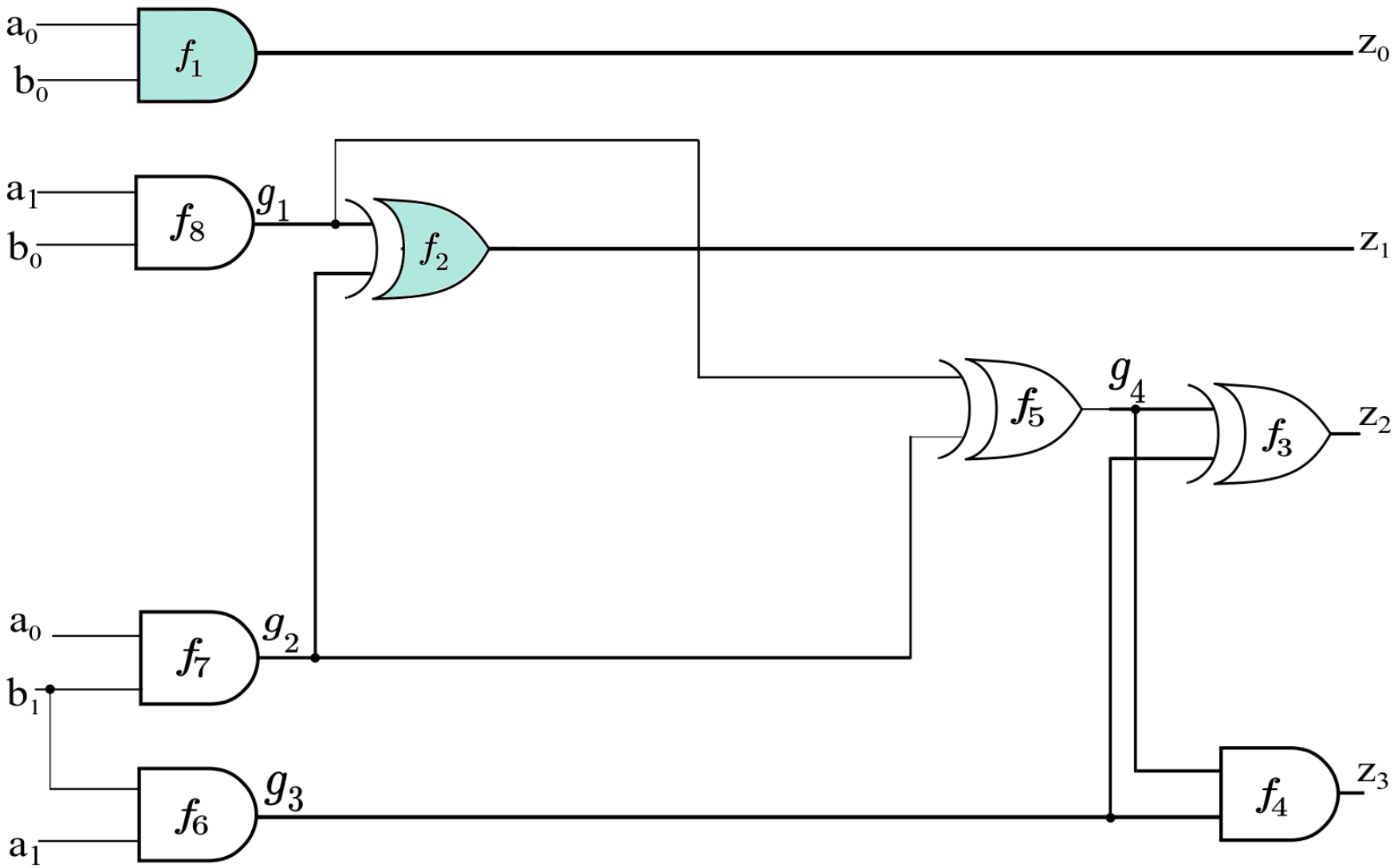
- Identify primary outputs affected by the bug(s)
- Identify set of potentially rectifiable target nets
- Ascertain if a net admits single-fix rectification.
- Compute a rectification polynomial in primary inputs
- Synthesize the rectification function and rectify the circuit

# Computing rectification function



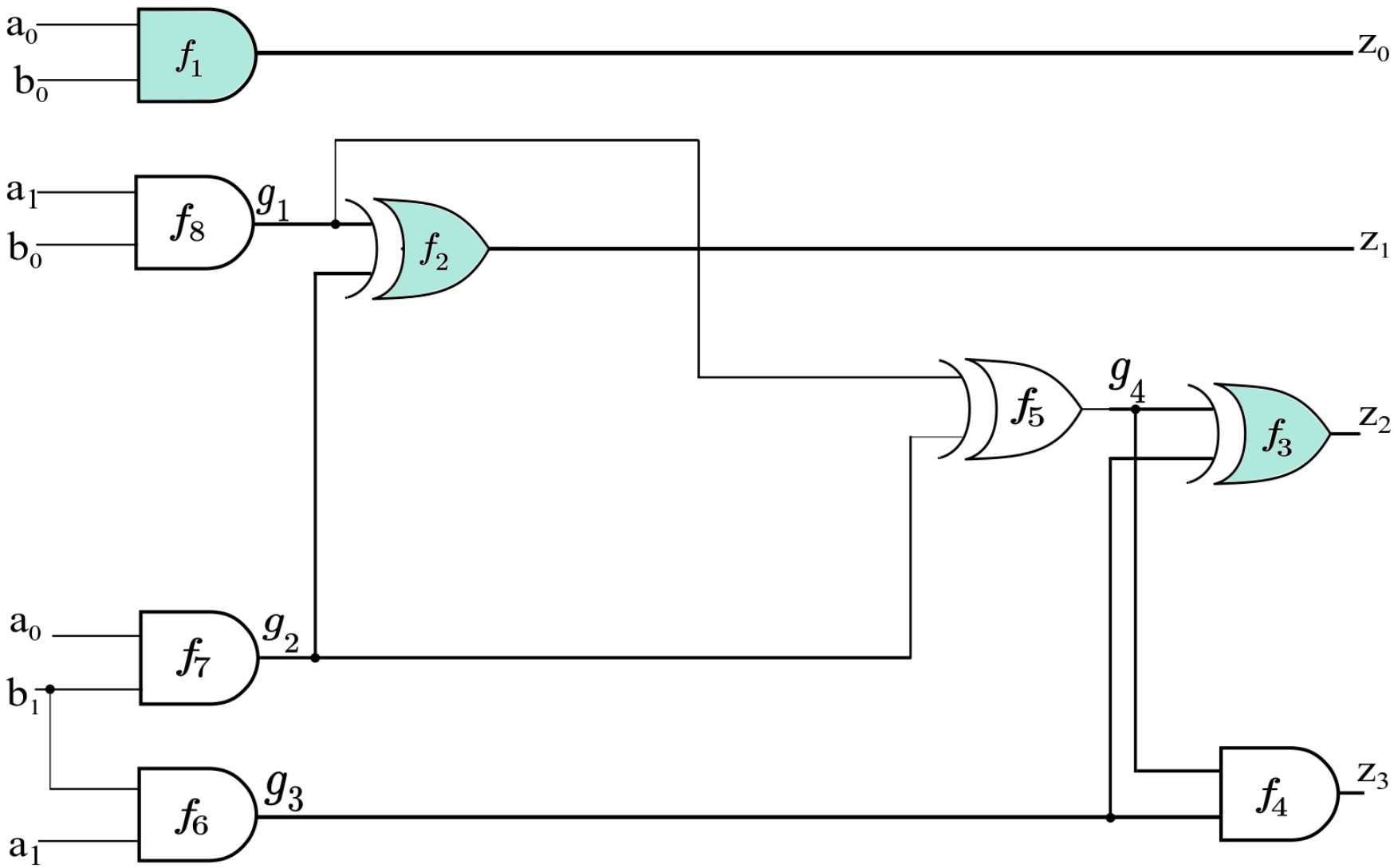

$$f \xrightarrow{f_1}$$

# Computing rectification function



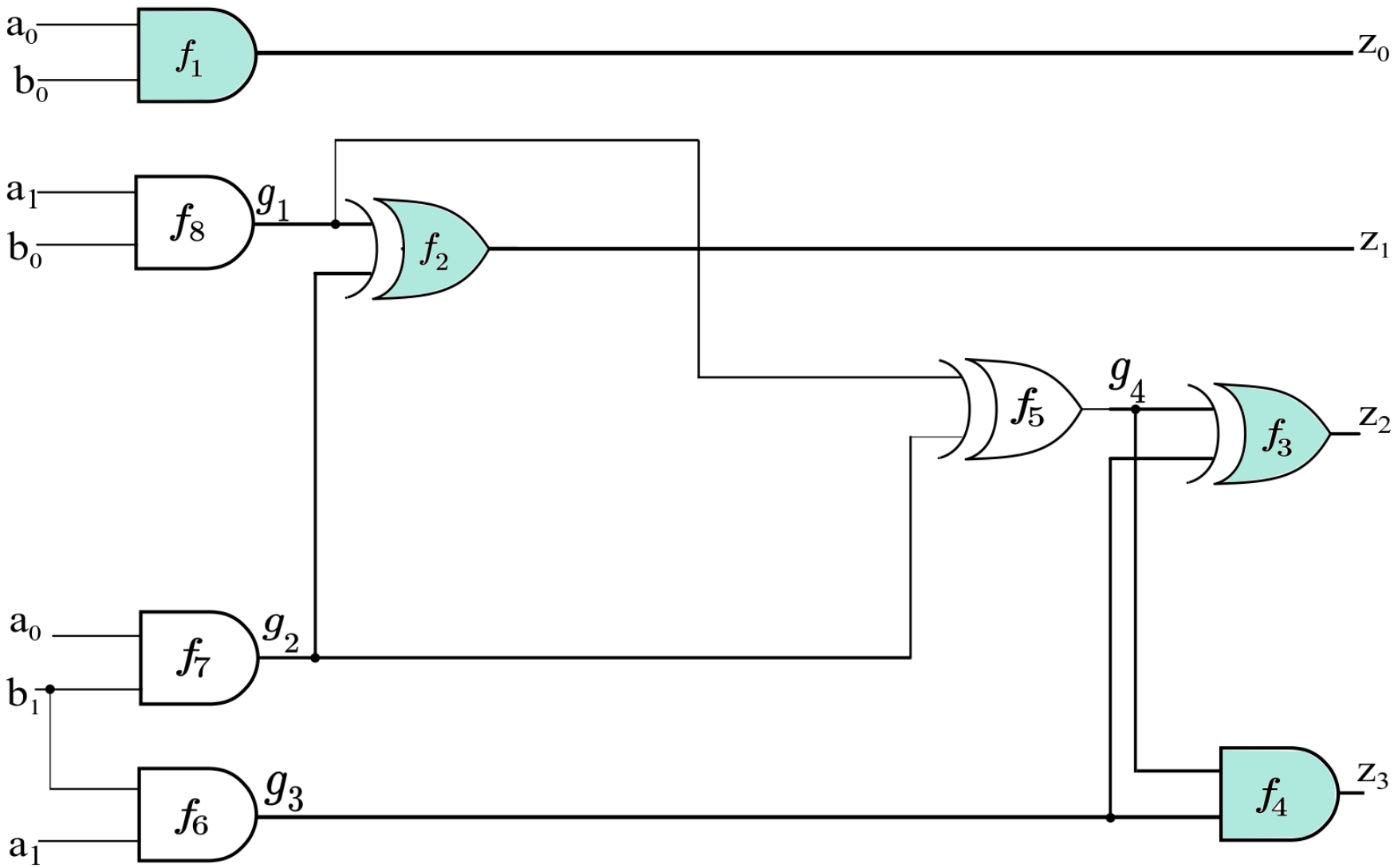

$$f \xrightarrow{f_1} \xrightarrow{f_2}$$

# Computing rectification function



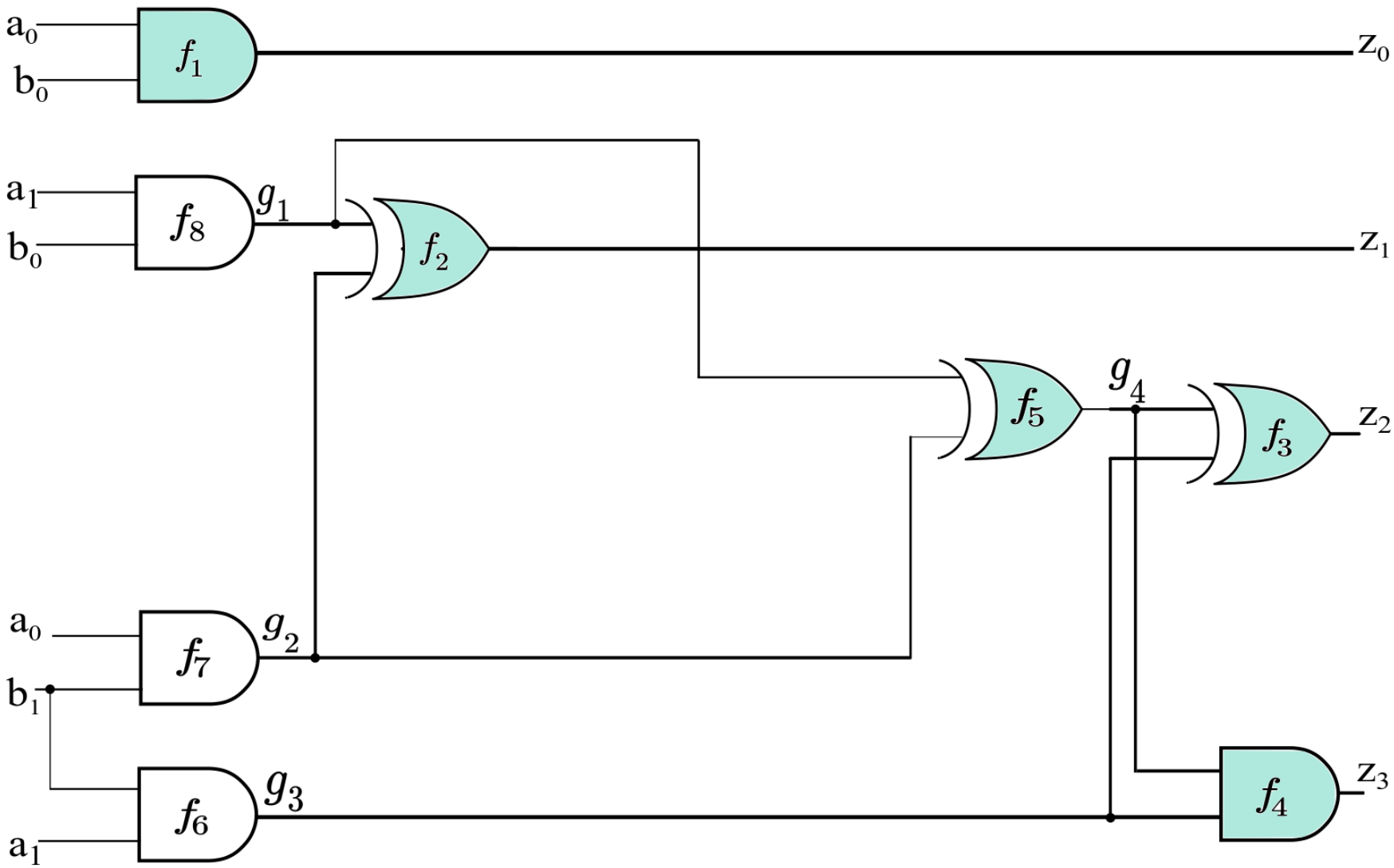

$$f \xrightarrow{f_1} \xrightarrow{f_2} \xrightarrow{f_3}$$

# Computing rectification function




$$f \xrightarrow{f_1} \xrightarrow{f_2} \xrightarrow{f_3} \xrightarrow{f_4}$$

# Computing rectification function




$$f \xrightarrow{f_1} \xrightarrow{f_2} \xrightarrow{f_3} \xrightarrow{f_4} \xrightarrow{g_4}$$

# Computing rectification function

- $f \xrightarrow{f_1} \xrightarrow{f_2} \xrightarrow{f_3} \xrightarrow{f_4} \xrightarrow{g_4} r$:
  $r = -z_0 - 2 \cdot z_1 - 4 \cdot g3 + a_0 \cdot b_0 + 2 \cdot a_0 \cdot b_1 + 2 \cdot a_1 \cdot b_0 + 4 \cdot a_1 \cdot b_1$
- Quotient $h_i = -4$
- $r \in \langle h_i, f_6, f_7, f_8 \rangle$
- $r = h'_i h_i + h'_6 f_6 + h'_7 f_7 + h'_8 f_8$

# Previous Work

- Automated Diagnosis
  - Boolean Reasoning [Madre. et al, ICCAD89][Liaw. et al, ICCAD90]
- Engineering Change Order (ECO)
  - Analogous to Rectification [Marek-Sadowska. et al, DAC95][Huang. et al, ICCAD10][Tang. et al, DAC11]
- Partial Synthesis
  - Quantified Boolean Formula(QBF) [Fujita. et al, IEEE12] ][Scholl. et al ICCD13]
  - Iterative incremental SAT [Fujita. et al, IEEE15]

# Previous Work

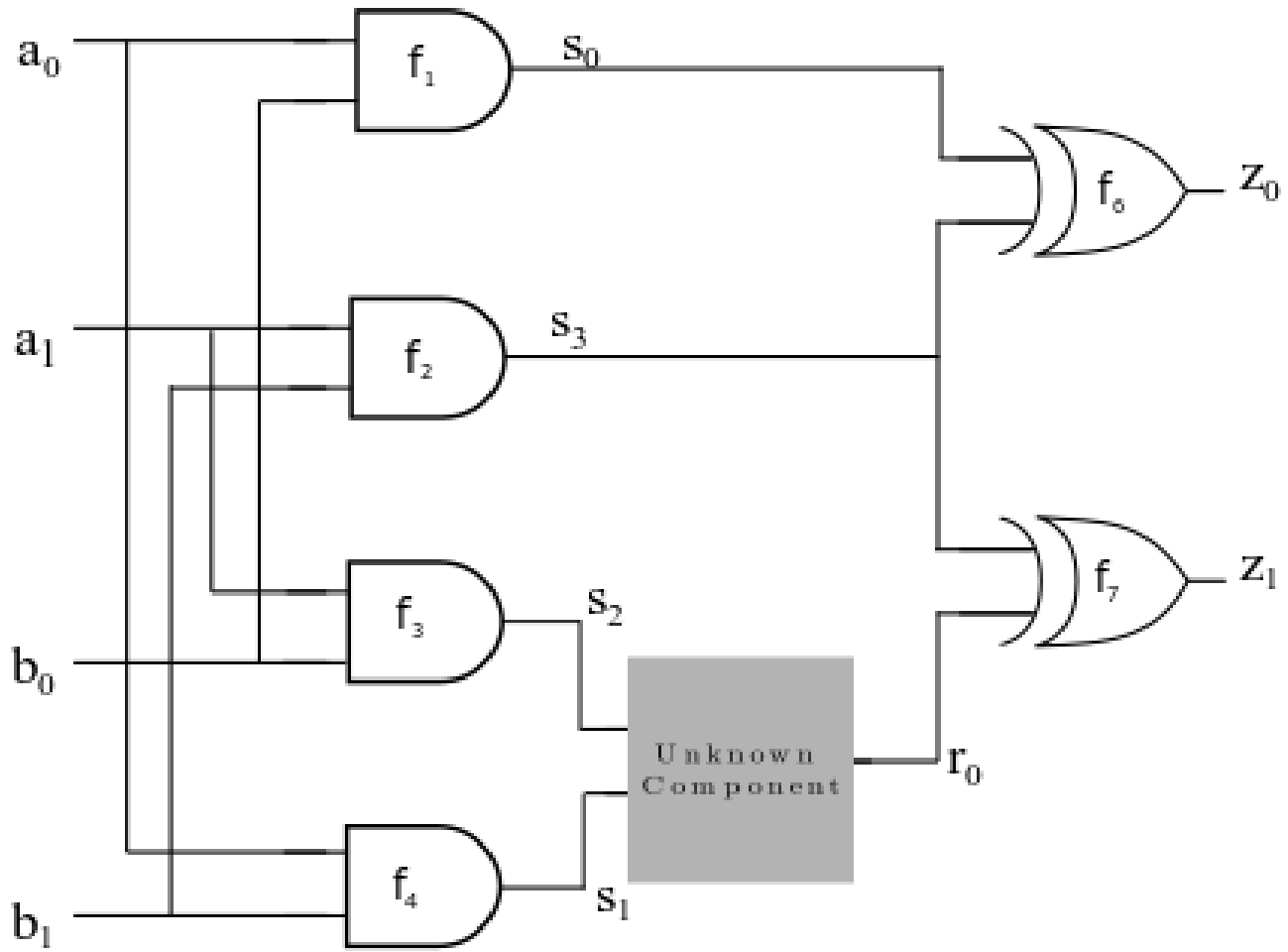



Figure: The Unknown Component Problem

# Previous Work

- Unknown component modeled as QBF
- Let logic function $U$ be the unknown component

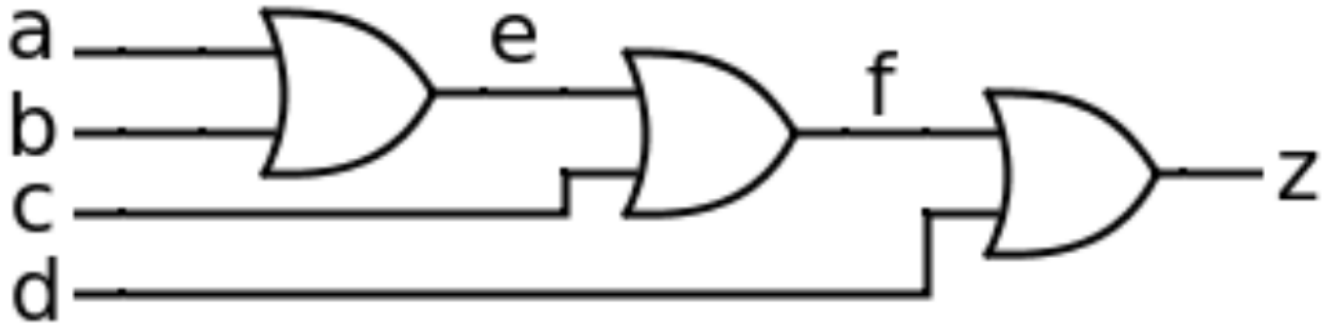
$$\exists U \forall P.I.\ Ckt = Spec \tag{1}$$

- Solved using CNF-SAT iteratively [Fujita. et al, IEEE15]
- SAT/QBF models - infeasible for arithmetic circuits

# Previous Work

- Rectification of finite field circuits [U.Gupta et al,VLSI-SOC 2018].
  - Craig Interpolants - alternative to Quantified Elimination.
  - Defined only for finite fields.
  - Craig Interpolants over infinite sets - an unresolved problem.
- Rectification of finite field circuits [V.Rao et al,FMCAD2018]
  - Using Gröbner Basis Reduction
  - Application over integers - remainder explosion.

# Example: Polynomial Reduction




$$f_1 : z + fd - f - d$$
$$f_2 : f + ec - e - c$$
$$f_3 : e + ba - b - a$$

$$z \xrightarrow{f_1} -fd + f + d$$
$$\xrightarrow{f_2} ecd - ec - ed + e - cd + c + d$$
$$\xrightarrow{f_3} -abcd + abc + abd - ab + acd - ac - ad + a + bcd$$
$$- bc - bd + b - cd + c + d$$

[1] D. Kaufmann, A. Biere, and M. Kauers, “Incremental column-wise verification of arithmetic circuits using computer algebra,” *Formal Methods in System Design*, Feb 2019. [Online]. Available: https://doi.org/10.1007/s10703-018-00329-2

[2] V. Rao, U. Gupta, I. Ilioaea, A. Srinath, P. Kalla, and F. Enescu, “Post-verification debugging and rectification of finite field arithmetic circuits using computer algebra techniques,” in *2018 Formal Methods in Computer Aided Design (FMCAD)*, Oct 2018, pp. 1–9.