The Unknown Component Problem

Vikas Rao
Department of Electrical and Computer Eng.
University of Utah
Vikas.k.rao@utah.edu

November 15, 2017

1 Preliminaries

Given a specification polynomial $f \in \mathbb{F}_q[x_1..x_n] = \mathbb{R}$ where $q = 2^k$, and a circuit C with S gates. Write the gates as polynomials in $\mathbb{R}$ as $F = \{f_1, .., f_i, .., f_s\}$: $J = \langle f_1, .., f_i, .., f_s \rangle$. Let us consider f_i to be the unknown component and of the special form $y_i + P(u_i)$, where y_i is the leading term of the polynomial representing the unknown gate and P as the function implementing the tail in terms of variables u_i, with $y_i > u_i$ as our variable order.

Let's assume that the circuit C correctly implements f. Then $f \in I(V_{\mathbb{F}_q}(J)) = J + J_0 : (J_0 = \langle x_i^q - x_i \rangle)$, let's also assume $J \subset J_0$.

$$\begin{aligned} f \in J \implies & f = h_1 f_1 + h_2 f_2 ... + h_i f_i + ... + h_s f_s : \text{where } h_i \in \mathbb{R} \\ & h_i f_i = f + h_1 f_1 + h_2 f_2 ... + h_{i-1} f_{i-1} + h_{i+1} f_{i+1} + ... + h_s f_s \\ & h_i f_i \in \langle f, f_1, f_2 ... f_{i-1}, f_{i+1} ... f_s \rangle \end{aligned} \tag{1}$$

Let J' represent this ideal $\langle f, f_1, f_2 ... f_{i-1}, f_{i+1} ... f_s \rangle$. Given the setup, can we project the variety of J' on y_i and u_i coordinates and recover f_i?

$$\text{Is } h_i f_i \in J' \cap [y_i, u_i]$$

2 Debug Example

Consider the circuit given in fig. 1 with specification given as $f : z + a * c + a + b * c + b + c$ and variables from ring $\mathbb{R} = \mathbb{F}_2[z, z_1, z_2, d_0, e_2, e_1, e_0, a, b, c]$. Let us assume f_4 to be the unknown gate in the design.

Polynomials for the given circuit are given as:

$$\begin{aligned} f_1 = e_0 + a + b; &\quad f_5 = z1 + e_0 * d_0 + e_0 + d_0; \\ f_2 = e_1 + b * c + b + c; &\quad f_6 = z_2 + d_0 + e_2; \\ f_3 = e_2 + c + 1; &\quad f_7 = z + z_1 * z_2; \\ f_4 = d_0 + P(e_1, c); & \end{aligned} \tag{2}$$

We shall add the vanishing polynomials of primary inputs, outputs, and intermediate variables and call this *ideal* J_0. Here 'α' is the root of primitive polynomial used to build the field.

$$\begin{aligned} f_8 &: a^2 + a; & f_{12} &: e_1^2 + e_1; & f_{16} &: z_2^2 + z_2; \\ f_9 &: b^2 + b; & f_{13} &: e_2^2 + e_2; & f_{17} &: z^2 + z; \\ f_{10} &: c^2 + c; & f_{14} &: d_0^2 + d_0; & & \\ f_{11} &: e_0^2 + e_0; & f_{15} &: z_1^2 + z_1; & & \end{aligned}$$

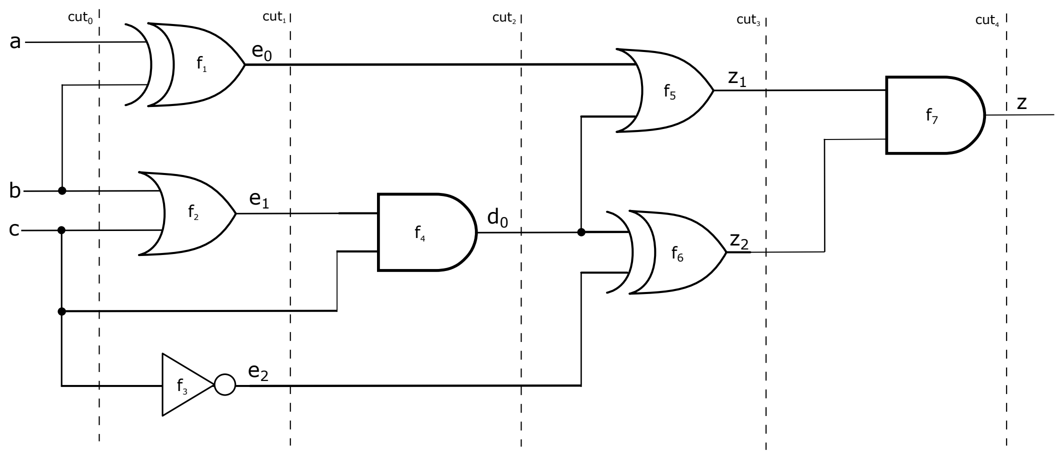

Figure 1: circuit with redundancy

From equation(1):

$$\begin{aligned} f \in \langle f_1, f_2 ... f_6, f_7 \rangle &: \text{ where } f_4 \text{ is unknown} \\ h_4 f_4 &\in \langle f, f_1, f_2, f_3, f_5, f_6, f_7 \rangle \\ h_4 f_4 = f + h_1 f_1 + h_2 f_2 + h_3 f_3 &+ h_5 f_5 + h_6 f_6 + h_7 f_7 \end{aligned} \tag{3}$$

Since we know that the unknown component lies between cuts cut_0 and cut_1, and given our RTTO>, we can compute $h_7, h_6, h_5,$ andh_4 with polynomial reduction as shown below. We will be using the notations: '[]' to represent quotient-h_i's, '()' to represent divisor-f_i's, and '{}' to represent the partial remainder of every reduction step-fp_i's.

Reduction order: $f_7 \rightarrow f_6 \rightarrow f_5 \rightarrow f_4$

Variable order:$\{z, z_2, z_1, d_0, e_2, e_1, e_0, a, b, c\}$

$$\begin{aligned} f &\xrightarrow{f_7} [1](z + z_2 * z_1) + \{z_2 * z_1 + a * c + a + b * c + b + c\} \rightarrow fp_1 \\ fp_1 &\xrightarrow{f_6} [z_1](z_2 + d_0 + e_2) + \{z_1 * d_0 + z_1 * e_2 + a * c + a + b * c + b + c\} \rightarrow fp_2 \\ fp_2 &\xrightarrow{f_5} [d_0 + e_2](z_1 + d_0 * e_0 + d_0 + e_0) + \{d_0 * e_0 * e_2 + d_0 * e_2 + d_0 + e_0 * e_2 + ac + a + bc + b + c\} \rightarrow fp_3 \\ fp_3 &\xrightarrow{f_4} [e_0 * e_2 + e_2 + 1](d_0 + P(e_1, c)) + \{fp_4\} \end{aligned}$$

Equation (3) can now be re-written as:

$$\begin{aligned}
h_4 f_4 + h_1 f_1 + h_2 f_2 + h_3 f_3 &= f + h_5 f_5 + h_6 f_6 + h_7 f_7;\\
h_4(d_0 + P(e_1, c)) + h_1 f_1 + h_2 f_2 + h_3 f_3 &= f + h_5 f_5 + h_6 f_6 + h_7 f_7;\\
h_4 * d_0 + h_4 * P(e_1, c) + h_1 f_1 + h_2 f_2 + h_3 f_3 &= f + h_5 f_5 + h_6 f_6 + h_7 f_7;\\
h_4 * P(e_1, c) + h_1 f_1 + h_2 f_2 + h_3 f_3 &= h_4 * d_0 + f + h_5 f_5 + h_6 f_6 + h_7 f_7;\\
h_4 * P(e_1, c) + h_1 f_1 + h_2 f_2 + h_3 f_3 &= e_0 * e_2 + a * c + a + b * c + b + c;\\
h_4 * P(e_1, c) + h_1 f_1 + h_2 f_2 + h_3 f_3 &= e_0 * e_2 + a * c + a + b * c + b + c;
\end{aligned}$$

Since, we know h_4, f_1, f_2, f_3, this can be formulated as a ideal membership testing:

$$e_0 * e_2 + a * c + a + b * c + b + c \in \langle h_4, f_1, f_2, f_3 \rangle \tag{4}$$

Term re-writing(yet to figure it out in singular), we can arrive at:
$e_0 * e_2 + a * c + a + b * c + b + c = [c](e_0 * e_2 + e_2 + 1) + [e_0 * c + e_0 + c](e_2 + c + 1) + (e_1 + b * c + b + c) + [c + 1](e_0 + a + b);$

Thus, $P(u_1) = P(e_1, c) = c$, which can implemented as a simple AND gate with c as both inputs.

Since any P_i which satisfies $P_i - P_j \in (f_1, f_2, f_3) : h_4$, where $i \neq j$, works for the circuit, the computed P is not considered unique and can take any values. We need to come up with better heuristics to identify the exact form and variables in which we want the P to be.