Word-Level Multi-Fix Rectifiability of Finite Field Arithmetic Circuits

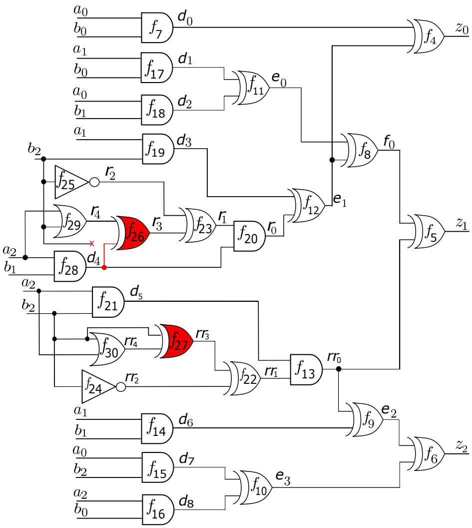
Vikas Rao[1], Irina Ilioaea[2], Haden Ondricek[1], Priyank Kalla[1], and Florian Enescu[3]

[1]Electrical & Computer Engineering, University of Utah
[2]Department of Mathematics, Louisiana State University Shreveport
[3]Mathematics & Statistics, Georgia State University

Outline

- Problem Description
- Preliminaries
- Problem Statement and Objective
- Single-fix Application
- Multi-Fix setup
    - Mathematical Challenges
- Rectifiability Check
- Experimental Results
- Summary and Future work

Problem Description:Rectification



A faulty implementation of a 3-bit modulo multiplier ($Z = A \cdot B \mod P(x)$)

Problem Description: Rectification

- Agnostic to the fault model, check for rectification at particular targets

Problem Description: Rectification

- Agnostic to the fault model, check for rectification at particular targets
  - Single-fix Rectification (SFR)
    - Correct circuit by changing function at a single net

Problem Description: Rectification

- Agnostic to the fault model, check for rectification at particular targets
  - Single-fix Rectification (SFR)
    - Correct circuit by changing function at a single net
- In a general setting, SFR might not be desired or may not exist

Problem Description: Rectification

- Agnostic to the fault model, check for rectification at particular targets
  - Single-fix Rectification (SFR)
    - Correct circuit by changing function at a single net
- In a general setting, SFR might not be desired or may not exist
  - Multi-fix Rectification (MFR)
    - Correct circuit by changing functions at multiple nets
    - Contribution: Multi-fix rectifiability setup and check

Preliminaries: Finite fields

- Fields - set of elements over which operations $(+, \cdot, /)$ can be performed
  - Ex. $\mathbb{R}, \mathbb{Q}, \mathbb{C}$

Preliminaries: Finite fields

- Fields - set of elements over which operations $(+, \cdot, /)$ can be performed
  - Ex. $\mathbb{R}, \mathbb{Q}, \mathbb{C}$
- Finite fields (Galois fields) - Finite set of elements
  - Ex. $\mathbb{F}_q$, where $q = p^n,\ p \ = \ prime,\ n \in \mathbb{Z}_{\geq 1}$
    - With $n = 1,\ and\ p = 2$, $\mathbb{F}_2 = \mathbb{B} = \{0, 1\}$
  - On circuits, $p = 2$, $n =$ data-operand width

iSQED® Symposium 2021

- Fields - set of elements over which operations $(+, \cdot, /)$ can be performed
  - Ex. $\mathbb{R}, \mathbb{Q}, \mathbb{C}$
- Finite fields (Galois fields) - Finite set of elements
  - Ex. $\mathbb{F}_q$, where $q = p^n,\ p\ =\ prime,\ n \in \mathbb{Z}_{\geq 1}$
    - With $n = 1,\ and\ p = 2$, $\mathbb{F}_2 = \mathbb{B} = \{0, 1\}$
  - On circuits, $p = 2$, $n =$ data-operand width
- Hardware cryptography extensively based on $\mathbb{F}_{2^n}$ (we use $\mathbb{F}_{2^n}$)

- Boolean logic gates in $\mathbb{F}_2$ $(\mathbb{F}_2 \subset \mathbb{F}_{2^n})$. Over $\mathbb{F}_2$, $-1 = +1 \pmod 2$

$$
\begin{aligned}
z &= \ \sim a & &\implies z + a + 1 & &\pmod 2\\
z &= a \wedge b & &\implies z + a \cdot b & &\pmod 2\\
z &= a \vee b & &\implies z + a \cdot b + a + b & &\pmod 2\\
z &= a \oplus b & &\implies z + a + b & &\pmod 2
\end{aligned}
$$

Preliminaries: Circuit polynomials over $\mathbb{F}_{2^n}$

- Boolean logic gates in $\mathbb{F}_2$ $(\mathbb{F}_2 \subset \mathbb{F}_{2^n})$. Over $\mathbb{F}_2$, $-1 = +1 \pmod 2$

$$
\begin{aligned}
z &= \sim a & \implies z + a + 1 & \pmod 2 \\
z &= a \wedge b & \implies z + a \cdot b & \pmod 2 \\
z &= a \vee b & \implies z + a \cdot b + a + b & \pmod 2 \\
z &= a \oplus b & \implies z + a + b & \pmod 2
\end{aligned}
$$

- Word-level polynomials [γ = primitive element of $\mathbb{F}_{2^n}$]

$$Output : Z + z_0 + \gamma \cdot z_1 + \cdots + \gamma^{n-1} \cdot z_{n-1}$$

$$Input : A + a_0 + \gamma \cdot a_1 + \cdots + \gamma^{n-1} \cdot a_{n-1}$$

Problem Statement and Objective

- A multivariate specification polynomial $f \in \mathbb{F}_{2^n}$
  - n is the operand width

Problem Statement and Objective

- A multivariate specification polynomial $f \in \mathbb{F}_{2^n}$
  - n is the operand width
- A faulty circuit implementation C for specification f
  - Model gates as polynomials over $\mathbb{F}_{2^n}$

Problem Statement and Objective

- A multivariate specification polynomial $f \in \mathbb{F}_{2^n}$
  - n is the operand width
- A faulty circuit implementation C for specification f
  - Model gates as polynomials over $\mathbb{F}_{2^n}$
- A primitive polynomial $P_n(x)$ used to construct $\mathbb{F}_{2^n}$
  - $\mathbb{F}_{2^n} = \mathbb{F}_2[x] \pmod{P_n(x)}$

Problem Statement and Objective

- A multivariate specification polynomial $f \in \mathbb{F}_{2^n}$
  - n is the operand width
- A faulty circuit implementation C for specification f
  - Model gates as polynomials over $\mathbb{F}_{2^n}$
- A primitive polynomial $P_n(x)$ used to construct $\mathbb{F}_{2^n}$
  - $\mathbb{F}_{2^n} = \mathbb{F}_2[x] \pmod{P_n(x)}$
- A set of m targets from C
  - Model it over $\mathbb{F}_{2^m}$ - challenges?

ISQED® Symposium 2021

Problem Statement and Objective

- A multivariate specification polynomial $f \in \mathbb{F}_{2^n}$
  - n is the operand width
- A faulty circuit implementation C for specification f
  - Model gates as polynomials over $\mathbb{F}_{2^n}$
- A primitive polynomial $P_n(x)$ used to construct $\mathbb{F}_{2^n}$
  - $\mathbb{F}_{2^n} = \mathbb{F}_2[x] \pmod{P_n(x)}$
- A set of m targets from C
  - Model it over $\mathbb{F}_{2^m}$ - challenges?
- **Check if C is rectifiable at these m targets**

- Let $R = \mathbb{F}_{2^n}[x_1, \ldots, x_d, Z]$
    - $\{f_1, \ldots, f_s\} \in R$

Preliminaries: Variables, Polynomials and order

- Let $R = \mathbb{F}_{2^n}[x_1, \ldots, x_d, Z]$
  - $\{f_1, \ldots, f_s\} \in R$
- In our context
  - $x_1, \ldots, x_d$: Variables (nets of the circuit)
  - Z: bit-vector representation for variables
  - $f_1, \ldots, f_s$: Polynomials from the circuit (logic gate relations)

- Let $R = \mathbb{F}_{2^n}[x_1, \ldots, x_d, Z]$
  - $\{f_1, \ldots, f_s\} \in R$
- In our context
  - $x_1, \ldots, x_d$: Variables (nets of the circuit)
  - Z: bit-vector representation for variables
  - $f_1, \ldots, f_s$: Polynomials from the circuit (logic gate relations)
- Multivariate polynomials: Impose monomial order "$>$" on R
  - We utilize lexicographic term order

Preliminaries: Variables, Polynomials and order

- Let $R = \mathbb{F}_{2^n}[x_1, \ldots, x_d, Z]$
  - $\{f_1, \ldots, f_s\} \in R$
- In our context
  - $x_1, \ldots, x_d$: Variables (nets of the circuit)
  - Z: bit-vector representation for variables
  - $f_1, \ldots, f_s$: Polynomials from the circuit (logic gate relations)
- Multivariate polynomials: Impose monomial order "$>$" on R
  - We utilize lexicographic term order
- Vanishing Polynomials: $F_0 = \langle x_1^2 + x_1, \ldots, x_d^2 + x_d, Z^{2^n} + Z \rangle$
  - Restrict solutions to x_i in $\mathbb{F}_2$
  - Restrict solutions to Z in $\mathbb{F}_{2^n}$

Algebraic Geometry: Ideals and Varieties

- $J = \langle F \rangle = \langle f_1, \ldots, f_s \rangle \subseteq \mathbb{F}_{2^n}[x_1, \ldots, x_d, Z]$
  - $\{h_1 f_1 + \cdots + h_s f_s : \; h_i \in R\}$

- $J = \langle F \rangle = \langle f_1, \ldots, f_s \rangle \subseteq \mathbb{F}_{2^n}[x_1, \ldots, x_d, Z]$
  - $\{h_1 f_1 + \cdots + h_s f_s : \; h_i \in R\}$
- Let $\boldsymbol{a} = (a_1, \ldots, a_d) \in \mathbb{F}_{2^n}^d$ $s.t.$ $f_1(\boldsymbol{a}) = \cdots = f_s(\boldsymbol{a}) = 0$

$$V(J) = \text{Set of all } \{\boldsymbol{a}\} \text{ s.t. } \begin{cases} f_1(\boldsymbol{a}) = 0, \\ f_2(\boldsymbol{a}) = 0, \\ \vdots \\ f_s(\boldsymbol{a}) = 0 \end{cases}$$

- $J = \langle F \rangle = \langle f_1, \ldots, f_s \rangle \subseteq \mathbb{F}_{2^n}[x_1, \ldots, x_d, Z]$
  - $\{h_1 f_1 + \cdots + h_s f_s : \; h_i \in R\}$
- Let $\boldsymbol{a} = (a_1, \ldots, a_d) \in \mathbb{F}_{2^n}^d$ $s.t.$ $f_1(\boldsymbol{a}) = \cdots = f_s(\boldsymbol{a}) = 0$

$$V(J) = \text{Set of all } \{\boldsymbol{a}\} \text{ s.t. } \begin{cases} f_1(\boldsymbol{a}) = 0, \\ f_2(\boldsymbol{a}) = 0, \\ \vdots \\ f_s(\boldsymbol{a}) = 0 \end{cases}$$

- $V(J)$ correspond to function mappings (Truth tables)

Gröbner Basis and Ideal membership

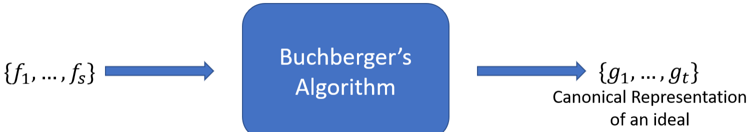
- An ideal can have many generators.
  - $J = \langle f_1, \ldots, f_s \rangle = \langle p_1, \ldots, p_m \rangle = \cdots = \langle g_1, \ldots, g_t \rangle$
  - Gröbner Basis (GB) is one such set with special properties

Gröbner Basis and Ideal membership

- An ideal can have many generators.
  - $J = \langle f_1, \ldots, f_s \rangle = \langle p_1, \ldots, p_m \rangle = \cdots = \langle g_1, \ldots, g_t \rangle$
  - Gröbner Basis (GB) is one such set with special properties
- Let $J = \langle f_1, \ldots, f_s \rangle = \langle g_1, \ldots, g_t \rangle$ and $G = \{g_1, \ldots, g_t\}$.
  - G is a Gröbner basis of $J \iff \forall f \in J, f \xrightarrow{g_1, \ldots, g_t}_+ 0$
  - Ideal membership: Let f be a polynomial in R:
    - if $f \xrightarrow{g_1, \ldots, g_t}_+ 0$, then f is a member of J.

Gröbner Basis and Ideal membership

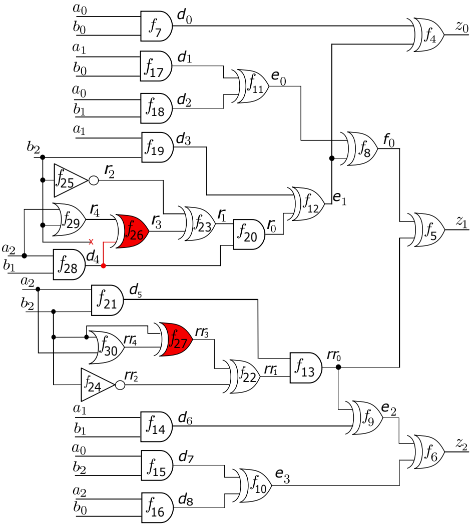
- An ideal can have many generators.
  - $J = \langle f_1, \ldots, f_s \rangle = \langle p_1, \ldots, p_m \rangle = \cdots = \langle g_1, \ldots, g_t \rangle$
  - Gröbner Basis (GB) is one such set with special properties
- Let $J = \langle f_1, \ldots, f_s \rangle = \langle g_1, \ldots, g_t \rangle$ and $G = \{g_1, \ldots, g_t\}$.
  - G is a Gröbner basis of $J \iff \forall f \in J, f \xrightarrow{g_1, \ldots, g_t}_{+} 0$
  - Ideal membership: Let f be a polynomial in R:
    - if $f \xrightarrow{g_1, \ldots, g_t}_{+} 0$, then f is a member of J.



Preliminaries: Term order

- Reverse Topological Term Order (RTTO)
  - Exploit circuit structure to avoid expensive GB computation
  - Standard practice to order variables topologically from POs to PIs

Application: Single-Fix Rectification



A faulty implementation of a 3-bit modulo multiplier ($Z = A \cdot B \ \ mod \ P_3(x)$)

Preliminaries: Term order

- Reverse Topological Term Order (RTTO)
  - Exploit circuit structure to avoid expensive GB computation
  - Standard practice to order variables topologically from POs to PIs

- Reverse Topological Term Order (RTTO)
  - Exploit circuit structure to avoid expensive GB computation
  - Standard practice to order variables topologically from POs to PIs
- Impose RTTO $>$ on circuit C:

$$
\begin{array}{rl}
f_1 : Z + z_0 + \gamma \cdot z_1 + \gamma^2 \cdot z_2; & f_{22} : rr_1 + rr_3 + rr_2; \\
f_2 : A + a_0 + \gamma \cdot a_1 + \gamma^2 \cdot a_2; & f_{23} : r_1 + r_2 + r_3; \\
f_3 : B + b_0 + \gamma \cdot b_1 + \gamma^2 \cdot b_2; & \color{red}{f_{26} : r_3 + r_4 + d_4;} \\
f_4 : z_0 + d_0 + e_1; & \color{red}{f_{27} : rr_3 + rr_4 + b_2;} \\
f_5 : z_1 + f_0 + rr_0; & \dots \\
\dots & f_{30} : rr_4 + a_2 + b_2 + a_2 b_2;
\end{array}
$$

Preliminaries: Term order

- Reverse Topological Term Order (RTTO)
  - Exploit circuit structure to avoid expensive GB computation
  - Standard practice to order variables topologically from POs to PIs

- Impose RTTO $>$ on circuit C:

$$
\begin{aligned}
f_1 &: Z + z_0 + \gamma \cdot z_1 + \gamma^2 \cdot z_2; & f_{22} &: rr_1 + rr_3 + rr_2;\\
f_2 &: A + a_0 + \gamma \cdot a_1 + \gamma^2 \cdot a_2; & f_{23} &: r_1 + r_2 + r_3;\\
f_3 &: B + b_0 + \gamma \cdot b_1 + \gamma^2 \cdot b_2; & f_{26} &: r_3 + r_4 + d_4;\\
f_4 &: z_0 + d_0 + e_1; & f_{27} &: rr_3 + rr_4 + b_2;\\
f_5 &: z_1 + f_0 + rr_0; & & \ldots\\
& \ldots & f_{30} &: rr_4 + a_2 + b_2 + a_2 b_2;
\end{aligned}
$$

- $F = \{f_1, \ldots, f_{30}\}$, $F_0 = \{a_0^2 - a_0, \ldots, z_2^2 - z_2, A^8 - A, \ldots, Z^8 - Z\}$.
  - Ideal $J + J_0 = \langle F \cup F_0 \rangle$ models C.

Application: Single-Fix Rectification

- Denote polynomial $f : Z + A \cdot B$ as the design specification.

- Denote polynomial $f : Z + A \cdot B$ as the design specification.
- Ideal $J + J_0 = \langle F \cup F_0 \rangle$ representing circuit C.

- Denote polynomial $f : Z + A \cdot B$ as the design specification.
- Ideal $J + J_0 = \langle F \cup F_0 \rangle$ representing circuit C.
- Circuit designed over $\mathbb{F}_{2^n} = \mathbb{F}_{2^3} (n = 3)$ using $P_n(x) = P_3(x) = x^3 + x + 1$ with $P_3(\gamma) = 0$

- Denote polynomial $f : Z + A \cdot B$ as the design specification.
- Ideal $J + J_0 = \langle F \cup F_0 \rangle$ representing circuit C.
- Circuit designed over $\mathbb{F}_{2^n} = \mathbb{F}_{2^3} (n = 3)$ using $P_n(x) = P_3(x) = x^3 + x + 1$ with $P_3(\gamma) = 0$
- **Is this circuit rectifiable at net r_3?**

SFR Application: Rectification Check

1. Rectification check at net r_3:
   - $J_1 = \langle F_1 \rangle$, where $F_1 = \{f_1, \ldots, f_{26} = r_3 + 0, \ldots, f_{30}\}$
   - $J_2 = \langle F_2 \rangle$, where $F_2 = \{f_1, \ldots, f_{26} = r_3 + 1, \ldots, f_{30}\}$

SFR Application: Rectification Check

1. Rectification check at net r_3:
   - $J_1 = \langle F_1 \rangle$, where $F_1 = \{f_1, \ldots, f_{26} = r_3 + 0, \ldots, f_{30}\}$
   - $J_2 = \langle F_2 \rangle$, where $F_2 = \{f_1, \ldots, f_{26} = r_3 + 1, \ldots, f_{30}\}$
2. Compute rem_1 and rem_2:
   - $rem_1 = f \xrightarrow{J_1, J_0}_+ (\gamma + 1) \cdot a_2 b_1 b_2 + (\gamma^2 + \gamma) \cdot a_2 b_2$
   - $rem_2 = f \xrightarrow{J_2, J_0}_+ (\gamma + 1) \cdot a_2 b_1 b_2 + (\gamma + 1) \cdot a_2 b_1 + (\gamma^2 + \gamma) \cdot a_2 b_2$

1. Rectification check at net r_3:
   - $J_1 = \langle F_1 \rangle$, where $F_1 = \{f_1, \ldots, f_{26} = r_3 + 0, \ldots, f_{30}\}$
   - $J_2 = \langle F_2 \rangle$, where $F_2 = \{f_1, \ldots, f_{26} = r_3 + 1, \ldots, f_{30}\}$
2. Compute rem_1 and rem_2:
   - $rem_1 = f \xrightarrow{J_1, J_0}_+ (\gamma + 1) \cdot a_2 b_1 b_2 + (\gamma^2 + \gamma) \cdot a_2 b_2$
   - $rem_2 = f \xrightarrow{J_2, J_0}_+ (\gamma + 1) \cdot a_2 b_1 b_2 + (\gamma + 1) \cdot a_2 b_1 + (\gamma^2 + \gamma) \cdot a_2 b_2$
3. SFR possible **iff** $V(rem_1) \cup V(rem_2) = \mathbb{F}_{2^3}^{|X_{PI}|} = V(J_0)$

SFR Application: Rectification Check

1. Rectification check at net r_3:
   - $J_1 = \langle F_1 \rangle$, where $F_1 = \{f_1, \dots, f_{26} = r_3 + 0, \dots, f_{30}\}$
   - $J_2 = \langle F_2 \rangle$, where $F_2 = \{f_1, \dots, f_{26} = r_3 + 1, \dots, f_{30}\}$
2. Compute rem_1 and rem_2:
   - $rem_1 = f \xrightarrow{J_1, J_0}_+ (\gamma + 1) \cdot a_2 b_1 b_2 + (\gamma^2 + \gamma) \cdot a_2 b_2$
   - $rem_2 = f \xrightarrow{J_2, J_0}_+ (\gamma + 1) \cdot a_2 b_1 b_2 + (\gamma + 1) \cdot a_2 b_1 + (\gamma^2 + \gamma) \cdot a_2 b_2$
3. SFR possible **iff** $V(rem_1) \cup V(rem_2) = \mathbb{F}_{2^3}^{|X_{PI}|} = V(J_0)$
   - Compute $G = GB(rem_1 \cdot rem_2, J_0)$ and check if $G = J_0$
   - In this example, target r_3 doesn't admit SFR

iSQED® Symposium 2021

Unified framework motivation

- For Single-fix, $m = 1$
  - Rectification patch modeled over $\mathbb{F}_{2^m} = \mathbb{F}_{2^1} = \mathbb{F}_2$
  - Circuit modeled over $\mathbb{F}_{2^n}$

Unified framework motivation

- For Single-fix, $m = 1$
  - Rectification patch modeled over $\mathbb{F}_{2^m} = \mathbb{F}_{2^1} = \mathbb{F}_2$
  - Circuit modeled over $\mathbb{F}_{2^n}$
    - $\forall n \in \mathbb{Z}_{>1}, 1 \mid n, \mathbb{F}_2 \subset \mathbb{F}_{2^n}$,

Unified framework motivation

- For Single-fix, $m = 1$
  - Rectification patch modeled over $\mathbb{F}_{2^m} = \mathbb{F}_{2^1} = \mathbb{F}_2$
  - Circuit modeled over $\mathbb{F}_{2^n}$
    - $\forall n \in \mathbb{Z}_{>1}, 1 \mid n, \mathbb{F}_2 \subset \mathbb{F}_{2^n},$
- For Multi-fix, since $m > 1$, $\mathbb{F}_{2^m}$ might not be contained in $\mathbb{F}_{2^n}$

Unified framework motivation

- For Single-fix, $m = 1$
  - Rectification patch modeled over $\mathbb{F}_{2^m} = \mathbb{F}_{2^1} = \mathbb{F}_2$
  - Circuit modeled over $\mathbb{F}_{2^n}$
    - $\forall n \in \mathbb{Z}_{>1}, 1 \mid n, \mathbb{F}_2 \subset \mathbb{F}_{2^n},$
- For Multi-fix, since $m > 1$, $\mathbb{F}_{2^m}$ might not be contained in $\mathbb{F}_{2^n}$
  - Ex. For $m = 2, n = 3, 2 \nmid 3, \mathbb{F}_{2^2} \not\subset \mathbb{F}_{2^3}$

iSQED® Symposium 2021

Unified framework motivation

- For Single-fix, $m = 1$
  - Rectification patch modeled over $\mathbb{F}_{2^m} = \mathbb{F}_{2^1} = \mathbb{F}_2$
  - Circuit modeled over $\mathbb{F}_{2^n}$
    - $\forall n \in \mathbb{Z}_{>1}, 1 \mid n, \mathbb{F}_2 \subset \mathbb{F}_{2^n},$
- For Multi-fix, since $m > 1$, $\mathbb{F}_{2^m}$ might not be contained in $\mathbb{F}_{2^n}$
  - Ex. For $m = 2, n = 3, 2 \nmid 3, \mathbb{F}_{2^2} \not\subset \mathbb{F}_{2^3}$
- Composite field $\mathbb{F}_{2^k}$
  - $\mathbb{F}_{2^m} \subset \mathbb{F}_{2^k}$ and $\mathbb{F}_{2^n} \subset \mathbb{F}_{2^k}$

Unified framework motivation

- For Single-fix, $m = 1$
  - Rectification patch modeled over $\mathbb{F}_{2^m} = \mathbb{F}_{2^1} = \mathbb{F}_2$
  - Circuit modeled over $\mathbb{F}_{2^n}$
    - $\forall n \in \mathbb{Z}_{>1}, 1 \mid n, \mathbb{F}_2 \subset \mathbb{F}_{2^n}$,
- For Multi-fix, since $m > 1$, $\mathbb{F}_{2^m}$ might not be contained in $\mathbb{F}_{2^n}$
  - Ex. For $m = 2, n = 3, 2 \nmid 3, \mathbb{F}_{2^2} \not\subset \mathbb{F}_{2^3}$
- Composite field $\mathbb{F}_{2^k}$
  - $\mathbb{F}_{2^m} \subset \mathbb{F}_{2^k}$ and $\mathbb{F}_{2^n} \subset \mathbb{F}_{2^k}$
  - What are the mathematical challenges?

Unified framework motivation

- For Single-fix, $m = 1$
  - Rectification patch modeled over $\mathbb{F}_{2^m} = \mathbb{F}_{2^1} = \mathbb{F}_2$
  - Circuit modeled over $\mathbb{F}_{2^n}$
    - $\forall n \in \mathbb{Z}_{>1}$, $1 \mid n$, $\mathbb{F}_2 \subset \mathbb{F}_{2^n}$,
- For Multi-fix, since $m > 1$, $\mathbb{F}_{2^m}$ might not be contained in $\mathbb{F}_{2^n}$
  - Ex. For $m = 2, n = 3$, $2 \nmid 3$, $\mathbb{F}_{2^2} \not\subset \mathbb{F}_{2^3}$
- Composite field $\mathbb{F}_{2^k}$
  - $\mathbb{F}_{2^m} \subset \mathbb{F}_{2^k}$ and $\mathbb{F}_{2^n} \subset \mathbb{F}_{2^k}$
  - What are the mathematical challenges?
  - What $P_K(x)$ should be used for constructing $\mathbb{F}_{2^k}$

Multi-fix Rectification: Prior work

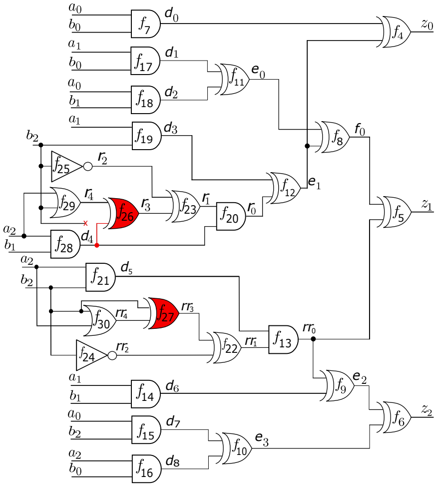
- Craig interpolation and/or iterative SAT solving [*Huang. et al*, DAC'11][*Huang. et al*, DATE'12]
  - Iteratively and incrementally patch the circuit
  - Compute multiple partial single-fix functions at the given *m* targets
- Resource aware ECO patch generation [*Jiang. et al*, DAC'18][*Mishchenko. et al*, DAC'18] [*Fujita. et al*, ISCAS'19]
- Symbolic sampling technique [*Jiang. et al*, DAC'19]
  - Enumerate rectification points functionally and match the circuitry of patches implicitly
  - Scalability achieved by modeling computations in symbolic sampling domain

Multi-fix Rectification: Prior work

- Craig interpolation and/or iterative SAT solving [*Huang. et al*, DAC'11][*Huang. et al*, DATE'12]
  - Iteratively and incrementally patch the circuit
  - Compute multiple partial single-fix functions at the given *m* targets
- Resource aware ECO patch generation [*Jiang. et al*, DAC'18][*Mishchenko. et al*, DAC'18] [*Fujita. et al*, ISCAS'19]
- Symbolic sampling technique [*Jiang. et al*, DAC'19]
  - Enumerate rectification points functionally and match the circuitry of patches implicitly
  - Scalability achieved by modeling computations in symbolic sampling domain

- Approaches infeasible on arithmetic circuits

Application: Multi-fix Rectification



A faulty implementation of a 3-bit (n=3) Mastrovito multiplier

MFR Notation: Field setup

- Circuit with data-path size n modeled over $\mathbb{F}_{2^n}$

MFR Notation: Field setup

- Circuit with data-path size n modeled over $\mathbb{F}_{2^n}$
  - $\mathbb{F}_{2^n} = \mathbb{F}_2[x] \pmod{P_n(x)}$
    - $P_n(x) \in \mathbb{F}_2[x]$ is a given degree-n primitive polynomial; $P_n(\gamma) = 0$

MFR Notation: Field setup

- Circuit with data-path size n modeled over $\mathbb{F}_{2^n}$
  - $\mathbb{F}_{2^n} = \mathbb{F}_2[x] \pmod{P_n(x)}$
    - $P_n(x) \in \mathbb{F}_2[x]$ is a given degree-n primitive polynomial; $P_n(\gamma) = 0$
  - Word-level polynomials for Z, A:
    - $f_Z : Z + \sum_{i=0}^{n-1} \gamma^i z_i, \; f_A : A + \sum_{i=0}^{n-1} \gamma^i a_i$

MFR Notation: Field setup

- Circuit with data-path size n modeled over $\mathbb{F}_{2^n}$
  - $\mathbb{F}_{2^n} = \mathbb{F}_2[x] \pmod{P_n(x)}$
    - $P_n(x) \in \mathbb{F}_2[x]$ is a given degree-n primitive polynomial; $P_n(\gamma) = 0$
  - Word-level polynomials for Z, A:
    - $f_Z : Z + \sum_{i=0}^{n-1} \gamma^i z_i,\ f_A : A + \sum_{i=0}^{n-1} \gamma^i a_i$
- Patch size m modeled over $\mathbb{F}_{2^m}$

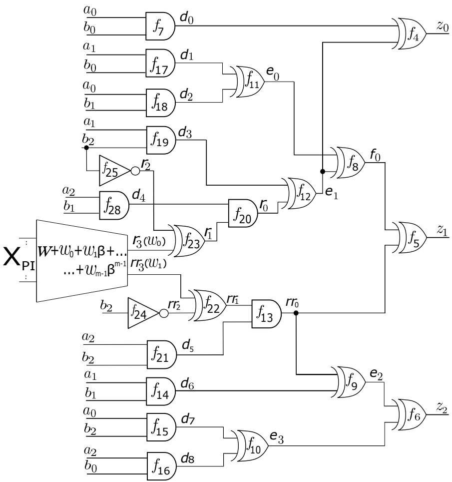
MFR Notation: Field setup

- Circuit with data-path size n modeled over $\mathbb{F}_{2^n}$
  - $\mathbb{F}_{2^n} = \mathbb{F}_2[x] \pmod{P_n(x)}$
    - $P_n(x) \in \mathbb{F}_2[x]$ is a given degree-n primitive polynomial; $P_n(\gamma) = 0$
  - Word-level polynomials for Z, A:
    - $f_Z : Z + \sum_{i=0}^{n-1} \gamma^i z_i, \; f_A : A + \sum_{i=0}^{n-1} \gamma^i a_i$

- Patch size m modeled over $\mathbb{F}_{2^m}$
  - $\mathbb{F}_{2^m} = \mathbb{F}_2[x] \pmod{P_m(x)}$
    - We select a degree-m primitive polynomial $P_m(x) \in \mathbb{F}_2[x]$; $P_m(\beta) = 0$

MFR Notation: Field setup

- Circuit with data-path size n modeled over $\mathbb{F}_{2^n}$
  - $\mathbb{F}_{2^n} = \mathbb{F}_2[x] \pmod{P_n(x)}$
    - $P_n(x) \in \mathbb{F}_2[x]$ is a given degree-n primitive polynomial; $P_n(\gamma) = 0$
  - Word-level polynomials for Z, A:
    - $f_Z : Z + \sum_{i=0}^{n-1} \gamma^i z_i,\ \ f_A : A + \sum_{i=0}^{n-1} \gamma^i a_i$
- Patch size m modeled over $\mathbb{F}_{2^m}$
  - $\mathbb{F}_{2^m} = \mathbb{F}_2[x] \pmod{P_m(x)}$
    - We select a degree-m primitive polynomial $P_m(x) \in \mathbb{F}_2[x]$; $P_m(\beta) = 0$
  - Word-level polynomial for W:
    - $f_w : W + \sum_{i=0}^{m-1} \beta^i w_i$
    - $\{w_0, \ldots, w_{m-1}\} \subset \{x_1, \ldots, x_d\}$



Patch function modeled as a 2-bit-vector word (m=2), $f_W : W + r_3 + \beta \cdot rr_3$

MFR Challenges: $\mathbb{F}_{2^k}$ and $P_k(x)$

- Smallest k is $LCM(n, m)$
  - $\mathbb{F}_{2^k} \supset \mathbb{F}_{2^n}$ and $\mathbb{F}_{2^k} \supset \mathbb{F}_{2^m}$
  - $\mathbb{F}_{2^k} = \mathbb{F}_2[x] \pmod{P_k(x)}$
    - $P_k(x)$ is a degree-k primitive polynomial; $P_k(\alpha) = 0$

MFR Challenges: $\mathbb{F}_{2^k}$ and $P_k(x)$

- Smallest k is $LCM(n, m)$
  - $\mathbb{F}_{2^k} \supset \mathbb{F}_{2^n}$ and $\mathbb{F}_{2^k} \supset \mathbb{F}_{2^m}$
  - $\mathbb{F}_{2^k} = \mathbb{F}_2[x] \pmod{P_k(x)}$
    - $P_k(x)$ is a degree-k primitive polynomial; $P_k(\alpha) = 0$
- Mathematical challenge: Given $P_n(x)$ and $P_m(x)$, compute $P_k(x)$ such that $P_n(\gamma) = P_m(\beta) = P_k(\alpha) = 0$
  - $\gamma = \alpha^{(2^k-1)/(2^n-1)} = \alpha^{\lambda}$
  - $\beta = \alpha^{(2^k-1)/(2^m-1)} = \alpha^{\mu}$

MFR Challenges: $\mathbb{F}_{2^k}$ and $P_k(x)$

- Smallest k is $LCM(n, m)$
  - $\mathbb{F}_{2^k} \supset \mathbb{F}_{2^n}$ and $\mathbb{F}_{2^k} \supset \mathbb{F}_{2^m}$
  - $\mathbb{F}_{2^k} = \mathbb{F}_2[x] \pmod{P_k(x)}$
    - $P_k(x)$ is a degree-k primitive polynomial; $P_k(\alpha) = 0$
- Mathematical challenge: Given $P_n(x)$ and $P_m(x)$, compute $P_k(x)$ such that $P_n(\gamma) = P_m(\beta) = P_k(\alpha) = 0$
  - $\gamma = \alpha^{(2^k-1)/(2^n-1)} = \alpha^\lambda$
  - $\beta = \alpha^{(2^k-1)/(2^m-1)} = \alpha^\mu$
- Solved using factorization of univariate polynomials over finite fields

Contribution: Computing $P_k(x)$

- Obtain UPFs of $P_n(x^\lambda)$ and $P_m(x^\mu)$ in $\mathbb{F}_2[x]$
- Then, $\exists P_k(x) \in \mathbb{F}_2[x]$ as a common factor of $P_n(x^\lambda)$ and $P_m(x^\mu)$, such that:
  - $P_k(x)$ is a degree-k primitive polynomial in $\mathbb{F}_2[x]$ with $P_k(\alpha) = 0$

Application: Computing $P_k(x)$

- $P_3(x) = x^3 + x + 1, \;\; P_2(x) = x^2 + x + 1, \;\; \gamma = \alpha^9, \;\; \beta = \alpha^{21}$
- Composite field: $k = LCM(2, 3) = 6$

Application: Computing $P_k(x)$

- $P_3(x) = x^3 + x + 1, \;\; P_2(x) = x^2 + x + 1, \;\; \gamma = \alpha^9, \;\; \beta = \alpha^{21}$
- Composite field: $k = LCM(2,3) = 6$
  - $UPF(P_3(x^9)) = (x^9)^3 + (x^9) + 1 = (x^6 + x^5 + x^2 + x + 1)(x^6 + x^5 + 1)(x^6 + x^4 + x^3 + x + 1)(x^6 + x^4 + x^2 + x + 1)(x^3 + x + 1);$

Application: Computing $P_k(x)$

- $P_3(x) = x^3 + x + 1, \;\; P_2(x) = x^2 + x + 1, \;\; \gamma = \alpha^9, \;\; \beta = \alpha^{21}$
- Composite field: $k = LCM(2,3) = 6$
  - $UPF(P_3(x^9)) = (x^9)^3 + (x^9) + 1 = (x^6 + x^5 + x^2 + x + 1)(x^6 + x^5 + 1)(x^6 + x^4 + x^3 + x + 1)(x^6 + x^4 + x^2 + x + 1)(x^3 + x + 1);$
  - $UPF(P_2(x^{21})) = (x^{21})^2 + (x^{21}) + 1 = (x^6 + x^5 + x^2 + x + 1)(x^6 + x^5 + 1)(x^6 + x^4 + x^3 + x + 1)(x^6 + x^5 + x^3 + x^2 + 1)(x^6 + x^5 + x^4 + x + 1)(x^6 + x + 1)(x^6 + x^3 + 1);$

Application: Computing $P_k(x)$

- $P_3(x) = x^3 + x + 1, \;\; P_2(x) = x^2 + x + 1, \;\; \gamma = \alpha^9, \;\; \beta = \alpha^{21}$
- Composite field: $k = LCM(2,3) = 6$
  - $UPF(P_3(x^9)) = (x^9)^3 + (x^9) + 1 = (x^6 + x^5 + x^2 + x + 1)(x^6 + x^5 + 1)(x^6 + x^4 + x^3 + x + 1)(x^6 + x^4 + x^2 + x + 1)(x^3 + x + 1);$
  - $UPF(P_2(x^{21})) = (x^{21})^2 + (x^{21}) + 1 = (x^6 + x^5 + x^2 + x + 1)(x^6 + x^5 + 1)(x^6 + x^4 + x^3 + x + 1)(x^6 + x^5 + x^3 + x^2 + 1)(x^6 + x^5 + x^4 + x + 1)(x^6 + x + 1)(x^6 + x^3 + 1);$
  - We choose $P_6(x) = x^6 + x^5 + 1$ as the required $P_k(x)$.

MFR Notation: Incorrect $P_k(x)$

- If we incorrectly choose $P_k(x) = x^6 + x^3 + 1$

MFR Notation: Incorrect $P_k(x)$

- If we incorrectly choose $P_k(x) = x^6 + x^3 + 1$
- For its root α, we have

MFR Notation: Incorrect $P_k(x)$

- If we incorrectly choose $P_k(x) = x^6 + x^3 + 1$
- For its root α, we have

$$\begin{aligned} \alpha^6 + \alpha^3 + 1 &= 0 \\ (\alpha^3)(\alpha^6 + \alpha^3 + 1) &= 0 \text{ (multiply by } \alpha^3\text{)} \\ \alpha^9 + \alpha^6 + \alpha^3 &= 0 \\ \gamma + 1 &= 0 \end{aligned}$$

MFR Notation: Incorrect $P_k(x)$

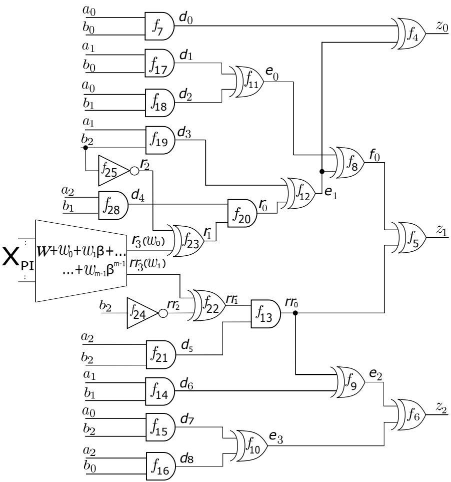
- If we incorrectly choose $P_k(x) = x^6 + x^3 + 1$
- For its root α, we have

$$\begin{aligned}
\alpha^6 + \alpha^3 + 1 &= 0 \\
(\alpha^3)(\alpha^6 + \alpha^3 + 1) &= 0 \text{ (multiply by } \alpha^3\text{)} \\
\alpha^9 + \alpha^6 + \alpha^3 &= 0 \\
\gamma + 1 &= 0
\end{aligned}$$

- However, $\gamma \neq 1$, as γ is a primitive element of $\mathbb{F}_{2^n}$
- Selecting arbitrary $P_k(x)$ leads to erroneous results

iSQED® Symposium 2021

Application: Word-level representation



Patch function modeled as a 2-bit-vector word (m=2), $f_W : W + r_3 + \beta \cdot rr_3$

iSQED Symposium 2021

MFR Notation: Word-level reasoning

- Obtain each w_i as a polynomial function in W, β
  - $\forall i \in 1, \ldots, m, \quad w_i = \mathcal{F}_i(W, \beta)$

MFR Notation: Word-level reasoning

- Obtain each w_i as a polynomial function in W, β
  - $\forall i \in 1, \ldots, m, \quad w_i = \mathcal{F}_i(W, \beta)$

$$
\begin{aligned}
W &= w_0 + \cdots + \beta^{m-1} \cdot w_{m-1} \\
W^2 &= w_0^2 + \cdots + \beta^{2(m-1)} \cdot w_{m-1}^2 \\
&\cdots \\
W^{2^{m-1}} &= w_0 + \cdots + \beta^{2^{m-1}(m-1)} \cdot w_{m-1}
\end{aligned}
$$

MFR Notation: Word-level reasoning

- Obtain each w_i as a polynomial function in W, β
  - $\forall i \in 1, \dots, m, \quad w_i = \mathcal{F}_i(W, \beta)$

$$
\begin{aligned}
W &= w_0 + \cdots + \beta^{m-1} \cdot w_{m-1} \\
W^2 &= w_0^2 + \cdots + \beta^{2(m-1)} \cdot w_{m-1}^2 \\
&\dots \\
W^{2^{m-1}} &= w_0 + \cdots + \beta^{2^{m-1}(m-1)} \cdot w_{m-1}
\end{aligned}
$$

- Solved using Gaussian elimination

MFR Notation: Word-level Formulation steps

1. Setup a new ring $R' = \mathbb{F}_{2^k}[x_1, \dots, x_d, Z, A, W]$

MFR Notation: Word-level Formulation steps

1. Setup a new ring $R' = \mathbb{F}_{2^k}[x_1, \ldots, x_d, Z, A, W]$
2. $\mathbb{F}_{2^k}$ is constructed using $P_k(x)$

MFR Notation: Word-level Formulation steps

1. Setup a new ring $R' = \mathbb{F}_{2^k}[x_1, \ldots, x_d, Z, A, W]$
2. $\mathbb{F}_{2^k}$ is constructed using $P_k(x)$
3. Construct a polynomial set F' as follows:

MFR Notation: Word-level Formulation steps

1. Setup a new ring $R' = \mathbb{F}_{2^k}[x_1, \ldots, x_d, Z, A, W]$
2. $\mathbb{F}_{2^k}$ is constructed using $P_k(x)$
3. Construct a polynomial set F' as follows:
   - Start with $F' = F$

iSQED® Symposium 2021

MFR Notation: Word-level Formulation steps

1. Setup a new ring $R' = \mathbb{F}_{2^k}[x_1, \ldots, x_d, Z, A, W]$
2. $\mathbb{F}_{2^k}$ is constructed using $P_k(x)$
3. Construct a polynomial set F' as follows:
   - Start with $F' = F$
   - Remove polynomials with w_i's as leading terms

MFR Notation: Word-level Formulation steps

1. Setup a new ring $R' = \mathbb{F}_{2^k}[x_1, \dots, x_d, Z, A, W]$
2. $\mathbb{F}_{2^k}$ is constructed using $P_k(x)$
3. Construct a polynomial set F' as follows:
   - Start with $F' = F$
   - Remove polynomials with w_i's as leading terms
   - Substitute $\forall i \in 1, \dots, m, \quad w_i = \mathcal{F}_i(W, \beta)$

MFR Notation: Word-level Formulation steps

1. Setup a new ring $R' = \mathbb{F}_{2^k}[x_1, \ldots, x_d, Z, A, W]$
2. $\mathbb{F}_{2^k}$ is constructed using $P_k(x)$
3. Construct a polynomial set F' as follows:
   - Start with $F' = F$
   - Remove polynomials with w_i's as leading terms
   - Substitute $\forall i \in 1, \ldots, m, \quad w_i = \mathcal{F}_i(W, \beta)$
   - Add $f_w : W + \sum_{i=0}^{m-1} \beta^i w_i$

MFR Notation: Word-level Formulation steps

1. Setup a new ring $R' = \mathbb{F}_{2^k}[x_1, \ldots, x_d, Z, A, W]$
2. $\mathbb{F}_{2^k}$ is constructed using $P_k(x)$
3. Construct a polynomial set F' as follows:
   - Start with $F' = F$
   - Remove polynomials with w_i's as leading terms
   - Substitute $\forall i \in 1, \ldots, m, \quad w_i = \mathcal{F}_i(W, \beta)$
   - Add $f_w : W + \sum_{i=0}^{m-1} \beta^i w_i$
   - Substitute $\beta = \alpha^{\mu}, \gamma = \alpha^{\lambda}$

- Impose RTTO $>$ on circuit C:

$$
\begin{array}{rl}
f_1 : Z + z_0 + \gamma \cdot z_1 + \gamma^2 \cdot z_2; & f_{22} : rr_1 + rr_3 + rr_2; \\
f_2 : A + a_0 + \gamma \cdot a_1 + \gamma^2 \cdot a_2; & f_{23} : r_1 + r_2 + r_3; \\
f_3 : B + b_0 + \gamma \cdot b_1 + \gamma^2 \cdot b_2; & f_{26} : r_3 + r_4 + d_4; \\
f_4 : z_0 + d_0 + e_1; & f_{27} : rr_3 + rr_4 + b_2; \\
f_5 : z_1 + f_0 + rr_0; & \dots \\
\dots & f_{30} : rr_4 + a_2 + b_2 + a_2 b_2;
\end{array}
$$

- Impose RTTO $>$ on circuit C:

$$
\begin{aligned}
f_1 &: Z + z_0 + \gamma \cdot z_1 + \gamma^2 \cdot z_2; & f_{22} &: rr_1 + rr_3 + rr_2;\\
f_2 &: A + a_0 + \gamma \cdot a_1 + \gamma^2 \cdot a_2; & f_{23} &: r_1 + r_2 + r_3;\\
f_3 &: B + b_0 + \gamma \cdot b_1 + \gamma^2 \cdot b_2; & f_{26} &: r_3 + r_4 + d_4;\\
f_4 &: z_0 + d_0 + e_1; & f_{27} &: rr_3 + rr_4 + b_2;\\
f_5 &: z_1 + f_0 + rr_0; & &\dots\\
&\dots & f_{30} &: rr_4 + a_2 + b_2 + a_2 b_2;
\end{aligned}
$$

- $F = \{f_1, \dots, f_{30}\}$, $F_0 = \{a_0^2 - a_0, \dots, z_2^2 - z_2, A^8 - A, \dots, Z^8 - Z\}$.
  - Ideal $J + J_0 = \langle F \cup F_0 \rangle$ models C.

MFR Application: Word-level Formulation

- 2-bit rectification patch over the 3-bit circuit can be performed over the field $\mathbb{F}_{2^6}$
  - $\mathbb{F}_{2^6} = \mathbb{F}_2[x] \pmod{P_6(x)}$
  - $P_6(x) = x^6 + x^5 + 1$

- Update polynomial set F to F' as:

$$
\begin{aligned}
& rr_3 = W^2 + W, \quad r_3 = \beta W^2 + \beta^2 W \\
& f'_{22} : rr_1 + (W^2 + W) + rr_2 \\
& f'_{23} : r_1 + r_2 + (\beta W^2 + \beta^2 W) \\
& f_W : W + r_3 + \beta \cdot rr_3 \\
& \beta = \alpha^{21} \text{ and } \gamma = \alpha^9 \\
& F' = \{f_1, \ldots, f_{21},\ f'_{22}, f'_{23}, f_W, \ldots, f_{30}\} - \{f_{26}, f_{27}\}
\end{aligned}
$$

MFR Contribution: Rectification Check

- Multi-fix rectification at target W
  - Construct the following ideals:

    where F'_l is obtained from F' by replacing $f_W \in F'$ with $f'_W : W + \delta[l], 1$

  - Performing the reductions for all $1 \leq l \leq 2^m$:
    - $f \xrightarrow{F'_l, F_0}_{+} rem_l$
  - Let $V_{\mathbb{F}_q}(rem_l)$ denote the varieties of the respective rem_l's
  - Multi-fix rectification exists at target W:

$$\textbf{if and only if} \bigcup_{l=1}^{2^m} V_{\mathbb{F}_q}(rem_l) = \mathbb{F}_q^{|X_{PI}|} = V(J_0)$$

MFR Application: Rectification Check

- Constructing the J_i ideals:
  - $J_1 = \langle F_1' \rangle$, where $F_1'[f_w] = W + \delta(1) = W$,
  - $J_2 = \langle F_2' \rangle$, where $F_2'[f_w] = W + \delta(2) = W + 1$,
  - $J_3 = \langle F_3' \rangle$, where $F_3'[f_w] = W + \delta(3) = W + \beta$,
  - $J_4 = \langle F_4' \rangle$, where $F_4'[f_w] = W + \delta(4) = W + \beta^2$
- Reducing the specification $f : Z + A \cdot B$ modulo these ideals, we get:
  - $rem_1 = f \xrightarrow{F_1' \cup F_0'}_+ \alpha^{27}(a_2 b_1 b_2) + \alpha^{36}(a_2 b_2)$
  - $rem_2 = f \xrightarrow{F_2' \cup F_0'}_+ \alpha^{27}(a_2 b_1 b_2 + a_2 b_1) + \alpha^{36}(a_2 b_2)$
  - $rem_3 = f \xrightarrow{F_3' \cup F_0'}_+ \alpha^{27}(a_2 b_1 b_2)$
  - $rem_4 = f \xrightarrow{F_4' \cup F_0'}_+ \alpha^{27}(a_2 b_1 b_2 + a_2 b_1)$
- Compute $GB(r_1 \cdot r_2 \cdot r_3 \cdot r_4, F_0) = F_0$
- Target W with nets r_3 and rr_3 admits MFR

Future work: Rectification function

- A polynomial which can be computed to rectify the circuit
  - $W = a_2 b_1 b_2 + \beta \cdot a_2 b_2$
  - $r_3 = (a_2 \wedge b_1 \wedge b_2), \quad rr_3 = (a_2 \wedge b_2)$

Focus: Finite Field Arithmetic Circuits

- Applications:
  - RSA, ECC, Error correcting codes, RFID, etc.
    - Crypto-system bugs can leak secret keys [*Biham. et al*, Crypto'08]
    - RFID tag cloning could cause counterfeiting [*Batina. et al*, Security'09]
  - Large datapath sizes in ECC crypto systems
    - In $\mathbb{F}_{2^n}$, $n = 163, 233, 283, 409, 571$ (NIST standard)

Focus: Finite Field Arithmetic Circuits

- Applications:
  - RSA, ECC, Error correcting codes, RFID, etc.
    - Crypto-system bugs can leak secret keys [*Biham. et al*, Crypto'08]
    - RFID tag cloning could cause counterfeiting [*Batina. et al*, Security'09]
  - Large datapath sizes in ECC crypto systems
    - In $\mathbb{F}_{2^n}$, $n = 163, 233, 283, 409, 571$ (NIST standard)
- Rectification Motivation:
  - Synthesize sub-functions as opposed to complete redesign
  - Automated debugging

Table: Word-level multi-fix rectifiability check against word level specification. Time is in seconds; rows marked '*' indicates $m \nmid n$; Benchmark = Mastrovito architecture, n = Datapath Size, #Gates = No. of gates, K = 10^3, m = patch size, k = encompassing composite field size, PF = time for polynomial factorization and computing minpoly for the composite field, RC = time for rectification check

n	#Gates	m	k	PF	RC
12	0.45K	2	12	NA	0.4
16	0.8K	2	16	NA	3.2
*16	0.8K	3	48	–	–
*20	0.0	3	60	–	–
32	2.8K	2	32	NA	184
48	6.4K	3	48	NA	–
64	11.2K	2	64	NA	–

- Algebraic approach for *m*-target MFR checking
  - Efficiency derived by interpreting targets as a bit-vector

Contributions and future work

- Algebraic approach for m-target MFR checking
  - Efficiency derived by interpreting targets as a bit-vector
- New mathematical insights for unified framework
  - Field incompatibility
  - Primitive polynomial computation

Contributions and future work

- Algebraic approach for m-target MFR checking
  - Efficiency derived by interpreting targets as a bit-vector
- New mathematical insights for unified framework
  - Field incompatibility
  - Primitive polynomial computation
- Computation of rectification function at the word-level
  - Define and formulate existence of don't cares at the word-level

Contributions and future work

- Algebraic approach for *m*-target MFR checking
  - Efficiency derived by interpreting targets as a bit-vector
- New mathematical insights for unified framework
  - Field incompatibility
  - Primitive polynomial computation

- Computation of rectification function at the word-level
  - Define and formulate existence of don't cares at the word-level
- Extend the approach to integer arithmetic circuits

MFR Function Example

- Compute a rectification function of the form $W = U(X_{PI})$
  - Here U is the *unknown component* computed as an m-bit-vector word
  - It represents the function $W = \sum_{i=0}^{m-1} \beta^i u_i$
    - Where u_i's represent the individual Boolean functions for the respective w_i's.

MFR Function Example

- Compute a rectification function of the form $W = U(X_{PI})$
  - Here U is the *unknown component* computed as an m-bit-vector word
  - It represents the function $W = \sum_{i=0}^{m-1} \beta^i u_i$
    - Where u_i's represent the individual Boolean functions for the respective w_i's.
- The *unknown component* problem is then formulated as an ideal membership test and solved using extended Gröbner Basis:

$$W + \beta^0 e_0 + \beta d_5 = W + U = W + \beta^0(a_1 b_2 + a_2 b_1) + \beta a_2 b_2;$$
$$e_0 = a_1 b_2 + a_2 b_1; \quad d_5 = a_2 b_2;$$

THANK YOU

Email: vikas.k.rao@utah.edu

Sum, Product, and Quotient of Ideals

$$\text{Given } J_1 = \langle f_1, \ldots, f_s \rangle \in R \text{ and } J_2 = \langle h_1, \ldots, h_r \rangle \in R$$

- Sum of ideals:
  - $J_1 + J_2 = \langle f_1, \ldots, f_s, h_1 \ldots, h_r \rangle$
- Product of ideals:
  - $J_1 \cdot J_2 = \langle f_i \cdot h_j : 1 \leq i \leq s, 1 \leq j \leq r \rangle$
- Ideal quotient of J_1 by J_2:
  - $J_1 : J_2 = \{ f \in R \mid f \cdot h \in J_1, \forall h \in J_2 \}$
- Ideals and varieties are dual concepts
  - $V(J_1 + J_2) = V(J_1) \cap V(J_2)$
  - $V(J_1 \cdot J_2) = V(J_1) \cup V(J_2)$
  - $V(J_1 : J_2) = V(J_1) - V(J_2)$

MFR Application: Word-level Formulation

- Update ring properties
  - $R = \mathbb{F}_q[x_1, \ldots, x_d, Z, A, W]$
  - Modify RTTO $>$ to place the target W before the lowest indexed target e_0
    - $\{Z\} > \{A > B\} > \{z_0 > z_1 > z_2\} > \{f_0 > e_2 > e_3\} > \{\mathbf{W} > e_0 > e_1 > d_5 > d_6 > d_7 > d_8\} > \{d_0 > d_1 > d_2 > d_3 > d_4\} > \{a_0 > a_1 > a_2 > b_0 > b_1 > b_2\}$.

- Update polynomial set F to F':
  - Delete polynomials for w_i's
  - Delete polynomials in the transitive fan-in of w_i's only
  - Transitive fan-outs of w_i's need to be replaced with their equivalent word-level representations in terms of W
  - Add $f_w : W + \sum_{i=0}^{m-1} \beta^i w_i$