# Formal Verification of Sequential Galois Field Circuits using Word-Level FSM Traversal via Algebraic Geometry


Xiaojun Sun*, Priyank Kalla*, Florian Enescu†

*Electrical & Computer Engineering, University of Utah

†Mathematics & Statistics, Georgia State University


## I. Preliminaries

### A. Computer Algebra Preliminaries

Let $\mathbb{F}_{2^k}[x_1,\dots,x_d]$ be the polynomial ring with indeterminates $x_1,\dots,x_d$.

**Definition I.1.** *Let $f_1,\dots,f_s$ be polynomials in $\mathbb{F}_q[x_1,\dots,x_n]$, then we set*

$$\langle f_1,\dots,f_s\rangle = \left\{\sum_{i=1}^{s} h_i f_i : h_1,\dots,h_s \in \mathbb{F}_q[x_1,\dots,x_n]\right\}.$$

*We call $\langle f_1,\dots,f_s\rangle$ an ideal, and $f_1,\dots,f_s$ is the generator of this ideal.*

For $k$-bit vector $A = (a_0,a_1,\dots,a_{k-1})$ in $\mathbb{F}_{2^k}$, if polynomial $f \in \mathbb{F}_{2^k}[x]$ satisfies $f(A) = 0$, we say polynomial $f$ vanishes on $A$. The set of all possible $A$ in $\mathbb{F}_{2^k}$ is called affine varieties. The variety that vanishes all generators of an ideal $J$ will also vanish all polynomials in that ideal, it is called variety of ideal $J$, denoted by $V(J)$.

Consider Fermat's little theorem in Galois field $\mathbb{F}_{2^k}$, all vectors in this field vanish polynomial $x^{2^k}+x$. Let ideal $J_0 \subseteq \mathbb{F}_{2^k}[x_1,\dots,x_d]$ include all these polynomials, then $J_0 = \langle x_1^{2^k}+x_1,\dots,x_d^{2^k}+x_d\rangle$ is called the ideal of all vanishing polynomials.

An ideal may have different sets of generators. There is a special set of generators known as Gröbner basis. Gröbner basis has an important property: the leading term of an arbitrary polynomial from ideal $J$ can be divided by leading term of at least one polynomial from $J$'s Gröbner basis.

There are many applications of Gröbner basis, one of them is to eliminate variables we do not need. For example, an ideal $J = \langle f_1,\dots,f_s\rangle$ in polynomial ring $\mathbb{F}_{2^k}[x_1,x_2,\dots,x_d]$ contains variables $x_1,x_2,\dots,x_l$ $(l<d)$ to be eliminated. Our goal is to compute elimination ideal $J_l = J \cap \mathbb{F}_{2^k}[x_{l+1},\dots,x_d]$, which cannot be achieved by simply removing generators containing variables $x_1,x_2,\dots,x_l$. However by computing Gröbner basis it is straightforward to eliminate arbitrary variables from an ideal, which is described by following theorem:

**Theorem I.1.** *(Elimination Theorem[**?**]) Let $J \subset \mathbb{F}_{2^k}[x_1,\dots,x_d]$ be an ideal and let $G$ be a Gröbner basis of $J$ with respect to a lexicographic ordering where $x_1 > x_2 > \cdots > x_d$. Then for every $0 \le l \le d$, the set $G_l = G \cap \mathbb{F}_{2^k}[x_{l+1},\dots,x_d]$ is a Gröbner basis of $l$-th elimination ideal $J_l$.*

### B. Abstraction using Gröbner Basis

Above theorem makes it possible to abstract system input/output function out of an arithmetic circuit(cite Tim's)[**?**]. Given circuit with word-level inputs $A = \{a_0,\dots,a_{k-1}\}$ and $B = \{b_0,\dots,b_{k-1}\}$, as well as word-level output $R = \{r_0,\dots,r_{k-1}\}$. They are defined by polynomials from $\mathbb{F}_{2^k}[x]$: for example using standard basis representation, the word-definition polynomials are $A + a_0 + a_1\alpha + \cdots + a_{k-1}\alpha^{k-1}, B + b_0 + b_1\alpha + \cdots + b_{k-1}\alpha^{k-1}$ and $R + r_0 + r_1\alpha + \cdots + r_{k-1}\alpha^{k-1}$ when minimal polynomial $P(\alpha) = 0$. All gates inputs/outputs inside this circuit are denoted by $\{x_1,\dots,x_d\}$. Define ideal $J_{ckt}$ as it includes all gates description polynomials and word-definition polynomials, and $J_0$ as the ideal of all vanishing polynomials. According to elimination theorem, if we arrange variable order to put primary input/output word-level variables to lowest priority (and $R > \{A,B\}$), let $G$ be Gröbner basis of merged ideal $J_{ckt}+J_0$ under this term order , then $G_l = G \cap \mathbb{F}_{2^k}[R,A,B]$ will only contain polynomials in $R,A,B$, and its first polynomial generator will be of the form $R + \mathcal{F}(A,B)$.

**Definition I.2.** *Abstraction term order $>$ is a lexicographic term order limiting variable order as $\{x_1,\dots,x_d\} > R > \{A,B\}$ on polynomial ring $\mathbb{F}_{2^k}[x_1,\dots,x_d,R,A,B]$.*

**Theorem I.2.** *Reduced Gröbner basis RGB with abstraction term order of ideal must include one and only one polynomial of the form $R + \mathcal{F}(A,B)$, such that $R = \mathcal{F}(A,B)$ is the unique, minimal and canonical representation of input-output function implemented by the circuit.*

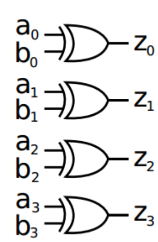


Fig. 1: 4-bit adder over $\mathbb{F}_{2^4}$

**Example I.1.** *Fig.**??** shows a 4-bit adder over $F_{2^4}$. Word-level variables $A,B$ are input operands and $R$ is output sum. Input/output function of this circuit is $R = A + B$, where $A = a_0 + a_1\alpha + a_2\alpha^2 + a_3\alpha^3$ with standard basis representation (similarly for $B,R$; minimal polynomial $P(\alpha) = \alpha^4 + \alpha + 1 = 0$). Circuit variable ideal $J_{ckt}$ consists of following generators: $f_1 : r_0 + a_0 + b_0; f_2 : r_1 + a_1 + b_1; f_3 : r_2 + a_2 + b_2; f_4 : r_3 + a_3 + b_3; f_5 : a_0 + a_1\alpha + a_2\alpha^2 + a_3\alpha^3 + A; f_6 : b_0 + b_1\alpha + b_2\alpha^2 + b_3\alpha^3 + B; f_7 : r_0 + r_1\alpha + r_2\alpha^2 + r_3\alpha^3 + R$. $J_0$ is generated by all vanishing polynomials. Impose following abstraction term order: $\{a_0,\dots,a_3,b_0,\dots,b_3,r_0,\dots,r_3\} > R > \{A,B\}$ and compute Gröbner basis $G$ of $J + J_0$. The generators of result GB includes: $g_1 : b_0 + b_1\alpha + b_2\alpha^2 + b_3\alpha^3 + B; g_2 : a_0 + a_1\alpha +$*

*$a_2\alpha^2 + a_3\alpha^3 + A; g_3 : r_3 + a_3 + b_3; g_4 : r_2 + a_2 + b_2; g_5 : r_1 + a_1 + b_1; g_6 : r_0 + a_0 + b_1\alpha + b_2\alpha^2 + b_3\alpha^3 + B; g_7 :$ **$R$+$A$+$B$** and polynomial generators from $J_0$. Polynomial $g_7 : R + A + B$ indicates function $R = A + B$ as the canonical polynomial function implemented by this circuit.*

## II. Verification of Sequential Arithmetic Circuits

Fig.**??** shows the basic structure of a sequential arithmetic circuit without primary inputs. There are 3 register files representing input operands $A, B$ and result output $R$. After $k$ clock cycles, data stored in register file $R$ will be the result of an arithmetic operation.

The combinational logic block takes present state variable $A, B$ and $R$ as inputs, and next state variable $A', B'$ and $R'$ as outputs. After $k$ clock cycles, the output $R_{final}$ equals to desired function of initial input operands $A_{init}, B_{init}$. Our approach aims to find a polynomial indicating this function without unrolling. An algorithm is used to describe how our approach works to verify this kind of sequential arithmetic circuits.

**ALGORITHM 1:** Sequential arithmetic circuit inductive verification

**Input**: Input-output circuit characteristic polynomial ideal $J_{ckt}$, initial state polynomial $\mathcal{F}(R), \mathcal{G}(A_{init}), \mathcal{H}(B_{init})$

1 $from^0(R,A,B) = \mathcal{F}(R), \mathcal{G}(A_{init}), \mathcal{H}(B_{init})$;
2 $i = 0$;
3 **repeat**
4 $\quad i \leftarrow i+1$;
5 $\quad to^i(R',A',B') \leftarrow$GB w/ abstraction term order$\langle J_{ckt}, J_0, from^{i-1}(R,A,B)\rangle$;
6 $\quad from^i \leftarrow to^i(\{R,A,B\} \setminus \{R',A',B'\})$;
7 **until** $i == k$;
8 **return** $from^k(R_{final})$

In this algorithm, $from^i$ and $to^i$ are polynomial ideals whose varieties are evaluations of word-level variables $R, A, B$ and $R', A', B'$ in $i$-th iteration. However in most cases, an arithmetic circuit should give the result $R_{final}$ in certain function of initial loaded operands $A_{init}$ and $B_{init}$ (they are fixed values during one calculation task); e.g. for a multiplier, $R_{final} = A_{init} \cdot B_{init}$. So in our approach we record all intermediate relations $R = \mathcal{F}(A_{init}, B_{init})$ for each clock cycle to evaluate output $R$. Run algorithm 1, the return value should be desired function $R_{final} = \mathcal{F}(A_{init}, B_{init})$.

**Example II.1.** *Fig.**??** shows the detailed structure of a 5-bit RH-multiplier (AESMPO). The transition function for operands $A, B$ is doing cyclic shift, while transition function for $R$ has to be computed through Gröbner basis abstraction approach. Following ideal $J_{ckt}$ from line 5 in algorithm 1 is the ideal for all gates in combinational logic block and definition of word-level variables.*

$$\begin{aligned} J_{ckt} = & d_0 + a_4b_4, c_1 + a_0 + a_4, c_2 + b_0 + b_4, d_1 + c_1c_2, c_3 + a_1a_4, \\ & c_4 + b_1b_4, d_2 + c_3c_4, e_0 + d_0 + d_1, e_3 + d_1 + d_2, e_4 + d_2, \\ & R_0 + r_4 + e_0, R_1 + r_0, R_2 + r_1, R_3 + r_2 + e_3, R_4 + r_3 + e_4, \\ & A + a_0\alpha^5 + a_1\alpha^{10} + a_2\alpha^{20} + a_3\alpha^9 + a_4\alpha^{18}, \\ & B + b_0\alpha^5 + b_1\alpha^{10} + b_2\alpha^{20} + b_3\alpha^9 + b_4\alpha^{18}, \\ & R' + r_0'\alpha^5 + r_1'\alpha^{10} + r_2'\alpha^{20} + r_3'\alpha^9 + r_4'\alpha^{18}, \\ & R + R_0\alpha^5 + R_1\alpha^{10} + R_2\alpha^{20} + R_3\alpha^9 + R_4\alpha^{18}; \end{aligned}$$

*In our implementation here, since we only focus on the output variable $R$, evaluations of intermediate input operands $A, B$ are unnecessary. Polynomials about $A$ and $B$ can be removed from $J_{ckt}$, and $R$ is directly evaluated by initial operands $A_{init}$ and $B_{init}$, which are associated with present state bit-level inputs $a_0, a_1, \ldots, a_4$ and $b_0, b_1, \ldots, b_4$ by polynomials in $from^i$.*

*According to line 5 of algorithm 1, we merge $J_{ckt}$, $J_0$ and $from^i$, then compute its Gröbner basis with abstraction term order (copy details here). There is a polynomial in form of $R' + \mathcal{F}(A_{init}, B_{init})$, which should be included by $to^{i+1}$. $to^{i+1}$ also exclude next state variable $A'$ and $B'$, instead we redefine $A_{init}$ and $B_{init}$ using next state bit-level variables $\{a_i', b_j'\}$. Next state Bit-level variables $a_i' = a_{i-1 \pmod k}, b_j' = b_{j-1 \pmod k}$ according to definition of cyclic shift.*

*Line 6 in algorithm 1 is implemented by replacing $R'$ with $R$, $\{a_i', b_j'\}$ with $\{a_i, b_j\}$.*

*All intermediate results for each clock cycle are listed below:*

- *Clock 1:* $from^0 = \{R, A_{init} + a_0\alpha^5 + a_1\alpha^{10} + a_2\alpha^{20} + a_3\alpha^9 + a_4\alpha^{18}, B_{init} + b_0\alpha^5 + b_1\alpha^{10} + b_2\alpha^{20} + b_3\alpha^9 + b_4\alpha^{18}\}$, $to^1 = \{R' + (\alpha^4+\alpha^3+1)A_{init}^{16}B_{init}^{16} + (\alpha^4+\alpha^2)A_{init}^{16}B_{init}^{4} + (\alpha^3+1)A_{init}^{16}B_{init}^{2} + (\alpha^4+\alpha^3+1)A_{init}^{16}B_{init} + (\alpha^4+\alpha^3+\alpha^2+1)A_{init}^{8}B_{init}^{8} + (\alpha^4+\alpha^3+\alpha+1)A_{init}^{8}B_{init}^{4} + (\alpha^3+\alpha+1)A_{init}^{8}B_{init}^{2} + (\alpha^4+\alpha^2)A_{init}^{8}B_{init} + (\alpha^4+\alpha^2)A_{init}^{4}B_{init}^{16} + (\alpha^4+\alpha^3+\alpha+1)A_{init}^{4}B_{init}^{8} + (\alpha^2)A_{init}^{4}B_{init}^{4} + (\alpha^3+\alpha^2+\alpha+1)A_{init}^{4}B_{init}^{2} + (\alpha^4+\alpha^3+\alpha+1)A_{init}^{4}B_{init} + (\alpha^3+1)A_{init}^{2}B_{init}^{16} + (\alpha^3+\alpha+1)A_{init}^{2}B_{init}^{8} + (\alpha^3+\alpha^2+\alpha+1)A_{init}^{2}B_{init}^{4} + (\alpha^3+\alpha^2+\alpha)A_{init}^{2}B_{init}^{2} + (\alpha^4+\alpha)A_{init}^{2}B_{init} + (\alpha^4+\alpha^3+1)A_{init}B_{init}^{16} + (\alpha^4+\alpha^2)A_{init}B_{init}^{8} + (\alpha^4+\alpha^3+\alpha+1)A_{init}B_{init}^{4} + (\alpha^4+\alpha)A_{init}B_{init}^{2} + (\alpha^3+\alpha+1)A_{init}B_{init}, A_{init} + a_4'\alpha^5 + a_0'\alpha^{10} + a_1'\alpha^{20} + a_2'\alpha^9 + a_3'\alpha^{18}, B_{init} + b_4'\alpha^5 + b_0'\alpha^{10} + b_1'\alpha^{20} + b_2'\alpha^9 + b_3'\alpha^{18}\}$
- *Clock 2:* $from^1 = \{R + (\alpha^4+\alpha^3+1)A_{init}^{16}B_{init}^{16} + (\alpha^4+\alpha^2)A_{init}^{16}B_{init}^{4} + (\alpha^3+1)A_{init}^{16}B_{init}^{2} + (\alpha^4+\alpha^3+1)A_{init}^{16}B_{init} + (\alpha^4+\alpha^3+\alpha^2+1)A_{init}^{8}B_{init}^{8} + (\alpha^4+\alpha^3+\alpha+1)A_{init}^{8}B_{init}^{4} + (\alpha^3+\alpha+1)A_{init}^{8}B_{init}^{2} + (\alpha^4+\alpha^2)A_{init}^{8}B_{init} + (\alpha^4+\alpha^2)A_{init}^{4}B_{init}^{16} + (\alpha^4+\alpha^3+\alpha+1)A_{init}^{4}B_{init}^{8} + (\alpha^2)A_{init}^{4}B_{init}^{4} + (\alpha^3+\alpha^2+\alpha+1)A_{init}^{4}B_{init}^{2} + (\alpha^4+\alpha^3+\alpha+1)A_{init}^{4}B_{init} + (\alpha^3+1)A_{init}^{2}B_{init}^{16} + (\alpha^3+\alpha+1)A_{init}^{2}B_{init}^{8} + (\alpha^3+\alpha^2+\alpha+1)A_{init}^{2}B_{init}^{4} + (\alpha^3+\alpha^2+\alpha)A_{init}^{2}B_{init}^{2} + (\alpha^4+\alpha)A_{init}^{2}B_{init} + (\alpha^4+\alpha^3+1)A_{init}B_{init}^{16} + (\alpha^4+\alpha^2)A_{init}B_{init}^{8} + (\alpha^4+\alpha^3+\alpha+1)A_{init}B_{init}^{4} + (\alpha^4+\alpha)A_{init}B_{init}^{2} + (\alpha^3+\alpha+1)A_{init}B_{init}, A_{init} + a_4\alpha^5 + a_0\alpha^{10} + a_1\alpha^{20} + a_2\alpha^9 + a_3\alpha^{18}, B_{init} + b_4\alpha^5 + b_0\alpha^{10} + b_1\alpha^{20} + b_2\alpha^9 + b_3\alpha^{18}\}$, $to^2 = \{R' + (\alpha^3+\alpha+1)A_{init}^{16}B_{init}^{16} + (\alpha^4+\alpha^3+1)A_{init}^{16}B_{init}^{8} + (\alpha^2)A_{init}^{16}B_{init}^{4} + (\alpha^3+1)A_{init}^{16}B_{init}^{2} + (\alpha^4+\alpha^3+1)A_{init}^{8}B_{init}^{16} + (\alpha^4+\alpha^2)A_{init}^{8}B_{init}^{8} + (\alpha^4)A_{init}^{8}B_{init}^{4} + (\alpha^4 +$

$\alpha^3 + 1)A_{init}^{8}B_{init}^{2} + (\alpha^3 + 1)A_{init}^{8}B_{init} + (\alpha^2)A_{init}^{4}B_{init}^{16} + (\alpha^4)A_{init}^{4}B_{init}^{8} + (\alpha^4)A_{init}^{4}B_{init}^{4} + (\alpha^4 + \alpha^3 + \alpha + 1)A_{init}^{4}B_{init}^{2} + (\alpha)A_{init}^{4}B_{init} + (\alpha^3 + 1)A_{init}^{2}B_{init}^{16} + (\alpha^4 + \alpha^3 + 1)A_{init}^{2}B_{init}^{8} + (\alpha^4 + \alpha^3 + \alpha + 1)A_{init}^{2}B_{init}^{4} + (\alpha^2)A_{init}^{2}B_{init}^{2} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}^{2}B_{init} + (\alpha^3 + 1)A_{init}B_{init}^{8} + (\alpha)A_{init}B_{init}^{4} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}B_{init}^{2} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}B_{init}, A_{init} + a'_3\alpha^5 + a'_4\alpha^{10} + a'_0\alpha^{20} + a'_1\alpha^9 + a'_2\alpha^{18}, B_{init} + b'_3\alpha^5 + b'_4\alpha^{10} + b'_0\alpha^{20} + b'_1\alpha^9 + b'_2\alpha^{18}\}$

- *Clock 3:* $from^2 = \{R + (\alpha^3 + \alpha + 1)A_{init}^{16}B_{init}^{16} + (\alpha^4 + \alpha^3 + 1)A_{init}^{16}B_{init}^{8} + (\alpha^2)A_{init}^{16}B_{init}^{4} + (\alpha^3 + 1)A_{init}^{16}B_{init}^{2} + (\alpha^4 + \alpha^3 + 1)A_{init}^{8}B_{init}^{16} + (\alpha^4 + \alpha^2)A_{init}^{8}B_{init}^{8} + (\alpha^4)A_{init}^{8}B_{init}^{4} + (\alpha^4 + \alpha^3 + 1)A_{init}^{8}B_{init}^{2} + (\alpha^3 + 1)A_{init}^{8}B_{init} + (\alpha^2)A_{init}^{4}B_{init}^{16} + (\alpha^4)A_{init}^{4}B_{init}^{8} + (\alpha^4)A_{init}^{4}B_{init}^{4} + (\alpha^4 + \alpha^3 + \alpha + 1)A_{init}^{4}B_{init}^{2} + (\alpha)A_{init}^{4}B_{init} + (\alpha^3 + 1)A_{init}^{2}B_{init}^{16} + (\alpha^4 + \alpha^3 + 1)A_{init}^{2}B_{init}^{8} + (\alpha^4 + \alpha^3 + \alpha + 1)A_{init}^{2}B_{init}^{4} + (\alpha^2)A_{init}^{2}B_{init}^{2} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}^{2}B_{init} + (\alpha^3 + 1)A_{init}B_{init}^{8} + (\alpha)A_{init}B_{init}^{4} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}B_{init}^{2} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}B_{init}, A_{init} + a_3\alpha^5 + a_4\alpha^{10} + a_0\alpha^{20} + a_1\alpha^9 + a_2\alpha^{18}, B_{init} + b_3\alpha^5 + b_4\alpha^{10} + b_0\alpha^{20} + b_1\alpha^9 + b_2\alpha^{18}\}$, $to^3 = \{R' + (\alpha^4 + \alpha^3 + 1)A_{init}^{16}B_{init}^{16} + (\alpha)A_{init}^{16}B_{init}^{8} + (\alpha^4 + \alpha^3 + \alpha^2 + 1)A_{init}^{16}B_{init}^{4} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}^{16}B_{init}^{2} + (\alpha^4 + \alpha^3 + \alpha^2 + 1)A_{init}^{16}B_{init} + (\alpha)A_{init}^{8}B_{init}^{16} + (\alpha + 1)A_{init}^{8}B_{init}^{8} + (\alpha^4)A_{init}^{8}B_{init}^{4} + (\alpha^3 + \alpha^2 + 1)A_{init}^{8}B_{init}^{2} + (\alpha^4 + \alpha^3 + \alpha + 1)A_{init}^{8}B_{init} + (\alpha^4 + \alpha^3 + \alpha^2 + 1)A_{init}^{4}B_{init}^{16} + (\alpha^4)A_{init}^{4}B_{init}^{8} + (\alpha^4 + \alpha^3 + \alpha + 1)A_{init}^{4}B_{init}^{4} + (\alpha^3 + \alpha + 1)A_{init}^{4}B_{init}^{2} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}^{4}B_{init} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}^{2}B_{init}^{16} + (\alpha^3 + \alpha^2 + 1)A_{init}^{2}B_{init}^{8} + (\alpha^3 + \alpha + 1)A_{init}^{2}B_{init}^{4} + (\alpha^3 + \alpha + 1)A_{init}^{2}B_{init}^{2} + (\alpha^4 + \alpha^3 + \alpha^2 + 1)A_{init}B_{init}^{16} + (\alpha^4 + \alpha^3 + \alpha + 1)A_{init}B_{init}^{8} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}B_{init}^{4} + (\alpha^4 + \alpha)A_{init}B_{init}, A_{init} + a'_2\alpha^5 + a'_3\alpha^{10} + a'_4\alpha^{20} + a'_0\alpha^9 + a'_1\alpha^{18}, B_{init} + b'_2\alpha^5 + b'_3\alpha^{10} + b'_4\alpha^{20} + b'_0\alpha^9 + b'_1\alpha^{18}\}$
- *Clock 4:* $from^3 = \{R + (\alpha^4 + \alpha^3 + 1)A_{init}^{16}B_{init}^{16} + (\alpha)A_{init}^{16}B_{init}^{8} + (\alpha^4 + \alpha^3 + \alpha^2 + 1)A_{init}^{16}B_{init}^{4} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}^{16}B_{init}^{2} + (\alpha^4 + \alpha^3 + \alpha^2 + 1)A_{init}^{16}B_{init} + (\alpha)A_{init}^{8}B_{init}^{16} + (\alpha + 1)A_{init}^{8}B_{init}^{8} + (\alpha^4)A_{init}^{8}B_{init}^{4} + (\alpha^3 + \alpha^2 + 1)A_{init}^{8}B_{init}^{2} + (\alpha^4 + \alpha^3 + \alpha + 1)A_{init}^{8}B_{init} + (\alpha^4 + \alpha^3 + \alpha^2 + 1)A_{init}^{4}B_{init}^{16} + (\alpha^4)A_{init}^{4}B_{init}^{8} + (\alpha^4 + \alpha^3 + \alpha + 1)A_{init}^{4}B_{init}^{4} + (\alpha^3 + \alpha + 1)A_{init}^{4}B_{init}^{2} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}^{4}B_{init} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}^{2}B_{init}^{16} + (\alpha^3 + \alpha^2 + 1)A_{init}^{2}B_{init}^{8} + (\alpha^3 + \alpha + 1)A_{init}^{2}B_{init}^{4} + (\alpha^3 + \alpha + 1)A_{init}^{2}B_{init}^{2} + (\alpha^4 + \alpha^3 + \alpha^2 + 1)A_{init}B_{init}^{16} + (\alpha^4 + \alpha^3 + \alpha + 1)A_{init}B_{init}^{8} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}B_{init}^{4} + (\alpha^4 + \alpha)A_{init}B_{init}, A_{init} + a_2\alpha^5 + a_3\alpha^{10} + a_4\alpha^{20} + a_0\alpha^9 + a_1\alpha^{18}, B_{init} + b_2\alpha^5 + b_3\alpha^{10} + b_4\alpha^{20} + b_0\alpha^9 + b_1\alpha^{18}\}$, $to^4 = \{R' + (\alpha^3 + \alpha + 1)A_{init}^{16}B_{init}^{16} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}^{16}B_{init}^{8} + (\alpha^4 + \alpha)A_{init}^{16}B_{init}^{4} + (\alpha^3 + 1)A_{init}^{16}B_{init}^{2} + (\alpha^3 + \alpha + 1)A_{init}^{16}B_{init} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}^{8}B_{init}^{16} + (\alpha^3 + 1)A_{init}^{8}B_{init}^{8} + (\alpha^4 + \alpha^2 + \alpha)A_{init}^{8}B_{init}^{4} + (\alpha^2 + \alpha)A_{init}^{8}B_{init}^{2} + (\alpha^3 + \alpha^2 + 1)A_{init}^{8}B_{init} + (\alpha^4 + \alpha)A_{init}^{4}B_{init}^{16} + (\alpha^4 + \alpha^2 + \alpha)A_{init}^{4}B_{init}^{8} + (\alpha^4 + \alpha^2 + \alpha)A_{init}^{4}B_{init}^{4} + (\alpha^2 + \alpha)A_{init}^{4}B_{init} + (\alpha^3 + 1)A_{init}^{2}B_{init}^{16} + (\alpha^2 + \alpha)A_{init}^{2}B_{init}^{8} + (\alpha^4 + \alpha^2)A_{init}^{2}B_{init}^{2} + (\alpha^3 + \alpha^2 + 1)A_{init}^{2}B_{init} + (\alpha^3 + \alpha + 1)A_{init}B_{init}^{16} + (\alpha^3 + \alpha^2 + 1)A_{init}B_{init}^{8} + (\alpha^2 + \alpha)A_{init}B_{init}^{4} + (\alpha^3 + \alpha^2 + 1)A_{init}B_{init}^{2} + (\alpha)A_{init}B_{init}, A_{init} + a'_1\alpha^5 + a'_2\alpha^{10} + a'_3\alpha^{20} + a'_4\alpha^9 + a'_0\alpha^{18}, B_{init} + b'_1\alpha^5 + b'_2\alpha^{10} + b'_3\alpha^{20} + b'_4\alpha^9 + b'_0\alpha^{18}\}$
- *Clock 5:* $from^4 = \{R + (\alpha^3 + \alpha + 1)A_{init}^{16}B_{init}^{16} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}^{16}B_{init}^{8} + (\alpha^4 + \alpha)A_{init}^{16}B_{init}^{4} + (\alpha^3 + 1)A_{init}^{16}B_{init}^{2} + (\alpha^3 + \alpha + 1)A_{init}^{16}B_{init} + (\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)A_{init}^{8}B_{init}^{16} + (\alpha^3 + 1)A_{init}^{8}B_{init}^{8} + (\alpha^4 + \alpha^2 + \alpha)A_{init}^{8}B_{init}^{4} + (\alpha^2 + \alpha)A_{init}^{8}B_{init}^{2} + (\alpha^3 + \alpha^2 + 1)A_{init}^{8}B_{init} + (\alpha^4 + \alpha)A_{init}^{4}B_{init}^{16} + (\alpha^4 + \alpha^2 + \alpha)A_{init}^{4}B_{init}^{8} + (\alpha^4 + \alpha^2 + \alpha)A_{init}^{4}B_{init}^{4} + (\alpha^2 + \alpha)A_{init}^{4}B_{init} + (\alpha^3 + 1)A_{init}^{2}B_{init}^{16} + (\alpha^2 + \alpha)A_{init}^{2}B_{init}^{8} + (\alpha^4 + \alpha^2)A_{init}^{2}B_{init}^{2} + (\alpha^3 + \alpha^2 + 1)A_{init}^{2}B_{init} + (\alpha^3 + \alpha + 1)A_{init}B_{init}^{16} + (\alpha^3 + \alpha^2 + 1)A_{init}B_{init}^{8} + (\alpha^2 + \alpha)A_{init}B_{init}^{4} + (\alpha^3 + \alpha^2 + 1)A_{init}B_{init}^{2} + (\alpha)A_{init}B_{init}, A_{init} + a_1\alpha^5 + a_2\alpha^{10} + a_3\alpha^{20} + a_4\alpha^9 + a_0\alpha^{18}, B_{init} + b_1\alpha^5 + b_2\alpha^{10} + b_3\alpha^{20} + b_4\alpha^9 + b_0\alpha^{18}\}$, $to^5 = \{R' + A_{init}B_{init}, A_{init} + a'_0\alpha^5 + a'_1\alpha^{10} + a'_2\alpha^{20} + a'_3\alpha^9 + a'_4\alpha^{18}, B_{init} + b'_0\alpha^5 + b'_1\alpha^{10} + b'_2\alpha^{20} + b'_3\alpha^9 + b'_4\alpha^{18}\}$

*The final result is* $from^5(R_{final}) = R_{final} + A_{init} \cdot B_{init}$