**MEMO NAME:** GFAU_MATHEMATICAL_BACKGROUND
**DATE:** February 1, 2018
**TO:** EFC LaBerge
**SUBJECT:** Mathematical Background on the Galois Field Arithmetic Unit

# 1 Background

A Galois Field is a field with a finite number of elements. The nomenclature $GF(q)$ is used to indicate a Galois Field with $q$ elements. In $GF(q)$, the parameter $q$ must be a power of a prime. For each prime power there exists exactly one finite field. The binary field $GF(2)$ is the most frequently used Galois field. (1)

## 1.1 Purpose and Scope

The Galois Field Arithmetic Unit operates in fields of $q^n$, where $2 \leq n \leq 15$. This document will demonstrate the functionality of the unit by deriving all its functionality through mathematical approaches. The mathematical algorithms will be followed by their corresponding implementations in digital design.

## 1.2 Terms and Keywords

### 1.2.1 Input Primitive Polynomials

Input primitive polynomials in the Galois Field are represented as:

$$c_n x^n + \ldots + c_2 x^2 + c_1 x^1 + c_0 x^0, \text{ where } c, x \in GF(2) = \{0, 1\}$$

For convenience and simplicity, all the examples provided will refer to the following polynomial: $x^3 + x^2 + x^0$ .

### 1.2.2 Elements

The elements of an input polynomial refer to the $2^n - 1$ elements in the field.

### 1.2.3 Polynomial Form

The polynomial forms of the elements refer to the $2^n - 1$ symbolic representations of the input primitive polynomials in the field.

### 1.2.4 Example

An example of the elements and their corresponding polynomials is provided below:

**Table 1:** The 8 Element Vectors of $x^3 + x^2 + x^0$ in $GF(2)[x]$

| **Element** | **Symbol** | **Polynomial Form** | **Symbol** |
|---|---|---|---|
| $0$ | (1) | $0 + 0 + 0$ | 000 |
| $\beta^0$ | 000 | $0 + 0 + \beta^0$ | 001 |
| $\beta^1$ | 001 | $0 + \beta^1 + 0$ | 010 |
| $\beta^2$ | 010 | $\beta^2 + 0 + 0$ | 100 |
| $\beta^3$ | 011 | $\beta^2 + 0 + \beta^0$ | 101 |
| $\beta^4$ | 100 | $\beta^2 + \beta^1 + \beta^0$ | 111 |
| $\beta^5$ | 101 | $0 + \beta^1 + \beta^0$ | 011 |
| $\beta^6$ | 110 | $\beta^2 + \beta^1 + 0$ | 110 |
| $\beta^7$ | (2) | $0 + 0 + \beta^0$ | 001 |

(1) Zero in its element form is reserved where its binary symbol does not represent its decimal value.

(2) Elements beyond the $(2^n - 1)$th element will be handled with special conditions since they cycle back to previous elements.

### 1.2.5 Notations

Several notations are used in this document to assist in linking the mathematical and digital design concepts. A list of selected notations are provided below:

**Table 2:** Mathematical and Logical Notations

| Notation | Definition |
|---:|---|
| $a + b$ | Arithmetic addition of $a$ and $b$ |
| $a - b$ | Arithmetic subtraction of $a$ and $b$ |
| $a \cdot b$ | Arithmetic multiplication of $a$ and $b$ |
| $a / b$ | Arithmetic Division of $a$ and $b$ |
| $a \wedge b$ | Logical conjunction of $a$ and $b$ |
| $a \vee b$ | Logical disjunction of $a$ and $b$ |
| $a \oplus b$ | Logical exclusive disjunction of $a$ and $b$ |
| $\overline{a}$ | Logical complement of $a$ |
| $a \ll b$ | Logical shift-left $a$ by $b$ bits |
| $a_{2's}$ | Two's complement of $a$ |
| $\lvert a \rvert$ | Number of bits in $a$ |
| $< x_n,\ x_{n-1},\ \ldots,\ x_0 >$ | Ordered array of size $n$ |
| $\{x_n,\ x_{n-1},\ \ldots,\ x_0\}$ | Unordered set of size $n$ |
| $< x_m,\ \overleftarrow{0_{m-1},\ \ldots,\ 0_n}, x_{n-1},\ \ldots,\ x_0 >$ | Zero-padding between the $(m-1)$th and $n$th bits |
| $A[i]$ | $i$th bit of array $A$ |
| $A[i:j]$ | Subarray of array $A$ from its $i$th to $j$th index inclusive; where $i > j$ |

# 2 Algorithms

Input polynomials will be represented as 16 bit zero- base arrays. For example, the polynomial $x^3 + x^2 + x^0$ will be represented as

$< 0000\ 0000\ 0000\ 1101 >$ (3rd, 2nd, and 0th bits)

## 2.1 Primitive Polynomial

A polynomial is said to be irreducible if and only if there exists no roots for it. A primitive polynomial is an irreducible polynomial that generates all elements of an extension field from a base field. (2)

## 2.2 Symbols

Once a polynomial is determined irreducible, its symbols may be generated. The number of terms grow exponentially, $2^n - 1$, where $n$ is the highest degree of the polynomial.

### 2.2.1 Default Symbols

Default symbols refer to terms in the field that exist for Galois Fields of all irreducible polynomials of $q^n$, where $2 \leq n \leq 15$. Since the number of elements cannot be smaller than $2$, only zero and the 0th and 1st elements are shared among all fields.

**Table 3:** Default Symbols Generated for All Irreducible Polynomials

| Element | Polynomial Form | Symbol |
|---|---|---|
| $0$ | $0_{15} + \ldots + 0_2 + 0_1 + 0_0$ | $< \overleftarrow{0_{15}\ \ldots\ 0_2} 0_1 0_0 >$ |
| $\alpha^0$ | $0_{15} + \ldots + 0_2 + 0_1 + \alpha_0^0$ | $< \overleftarrow{0_{15}\ \ldots\ 0_2} 0_1 1_0 >$ |
| $\alpha^1$ | $0_{15} + \ldots + 0_2 + \alpha_1^1 + 0$ | $< \overleftarrow{0_{15}\ \ldots\ 0_2} 1_1 0_0 >$ |

Table 3 have all the bits of their values set to $0$ except where indicated.

#### 2.2.1.1 Analytical Approach

Using the example polynomial $x^3 + x^2 + x^0$ , show that 0, the 0th element and the first element exist in $GF(2^3)$.

$$\text{Let } \beta \in GF(2^3) \text{ be a root of } x^3 + x^2 + x^0 \quad \implies \beta^3 + \beta^2 + \beta^0$$

$$\therefore \text{The coefficients are in } GF(2) \implies \beta^3 = \beta^2 + \beta^0$$

$$\because \text{a field has additive and multiplicative identities:}$$

$$\therefore \{0, 1 = \beta^0\} \in GF(2^3)$$

$$\therefore \beta^1 \in GF(2^3) \ (\because \text{closure of multiplication})$$

### 2.2.2 Automatic Symbols

Automatic symbols refer to terms up to $x^{n-1}$. Automatic symbols may be generated concurrently, and consist of the following attributes:

1. The symbols for $\{x^0, x^1, \ldots, x^{n-1}\}$ are generated by setting the corresponding bits to 1.
2. The symbol for $x^n$ is generated by setting the corresponding bits for the terms in the polynomial after the highest degree term.
3. The symbol for $x^{2^n-1}$ cycles back to $x^0$, and is set to $x^0$.

Automatic symbols consist of $n+1$ terms. Therefore, the maximum of 15 bits would have 16 terms generated by default.

#### 2.2.2.1 Analytical Approach

Assuming the preceding values exist from the proof above, use the example polyno-

**Table 4:** Automatic Symbols Generated for An Irreducible Polynomial

| Element | Polynomial Form | Symbol |
|---|---|---|
| $\alpha^2$ | $0_{15} + \ldots + 0_{n-1} + \ldots + \alpha_2^2 + 0_1 + 0_0$ | $< \overleftarrow{0_{15} \ \ldots \ 0_{n-1}} \ \ldots \ 1_2 0_1 0_0 >$ |
| $\ldots$ | $\ldots$ | $\ldots$ |
| $\alpha^{n-1}$ | $0_{15} + \ldots + \alpha_{n-1}^{n-1} + \ldots + 0_2 + 0_1 + 0_0$ | $< 0_{15} \ \ldots \ 1_{n-1} \ \ldots \ 0_2 0_1 0_0 >$ |
| $\alpha^n$ | $0_{15} + \ldots + \alpha_{n-1}^{n-1} + \ldots + \alpha_2^2 + \alpha_1^1 + \alpha_0^0$ | $< 0_{15} \ \ldots \ x_{n-1} \ \ldots \ x_2 x_1 x_0 >$ |
| $\alpha^{2^n-1}$ | $0_{15} + \ldots + 0_{n-1} + \ldots + \alpha_2^2 + 0_1 + 0_0$ | $< \overleftarrow{0_{15} \ \ldots \ 0_2} 0_1 0_0 >$ |

mial $x^3 + x^2 + x^0$ to show that the $n-1$th and the $n$th elements exist in $GF(2^3)$.

$$\text{Let } \beta \in GF(2^3) \text{ be a root of } x^3 + x^2 + x^0 \implies \beta^3 + \beta^2 + \beta^0$$

$$\therefore \text{The coefficients are in } GF(2) \implies \beta^3 = \beta^2 + \beta^0$$

$$\therefore \beta^2 \in GF(2^3) \ (\because \text{ closure of multiplication})$$

$$\therefore \beta^3 \in GF(2^3) \ (\because \beta^3 = \beta^2 + \beta^0)$$

### 2.2.3 Generated Symbols

The rest of the symbols for the elements $x^{n+1}$ to $x^{2^n-2}$ must be generated. In total, that would require $2^n - 2 - n - 1 + 1 = 2^n - 2 - n$ terms. Therefore, the maximum of 15 bits would require 32,751 terms to be generated.

#### 2.2.3.1 Analytical Approach

Assuming the preceding values exist from the proof above, use the example polynomial $x^3 + x^2 + x^0$ to show that the elements up to the $(2^{n-2})$th elements exist in $GF(2^3)$.

$$\text{Let } \beta \in GF(2^3) \text{ be a root of } x^3 + x^2 + x^0 \implies \beta^3 + \beta^2 + \beta^0$$

$\therefore$ The coefficients are in $GF(2) \implies \beta^3 = \beta^2 + \beta^0$

$$\begin{aligned}
\because \beta^4 &= \beta^1 \times \beta^3 \\
&= \beta^1(\beta^2 + \beta^0) \\
&= \beta^3 + \beta^1 \\
&= \beta^2 + \beta^1 + \beta^0
\end{aligned}$$

$\therefore \beta^4 \in GF(2^3)$

$$\begin{aligned}
\because \beta^5 &= \beta^1 \times \beta^4 \\
&= \beta^1(\beta^2 + \beta^1 + \beta^0) \\
&= \beta^3 + \beta^2 + \beta^1 \\
&= \beta^2 + \beta^0 + \beta^2 + \beta^1 \\
&= \beta^1 + \beta^0
\end{aligned}$$

$\therefore \beta^5 \in GF(2^3)$

$$\begin{aligned}
\because \beta^6 &= \beta^1 \times \beta^5 \\
&= \beta^1(\beta^1 + \beta^0) \\
&= \beta^2 + \beta^1
\end{aligned}$$

$\therefore \beta^6 \in GF(2^3)$

$$\begin{aligned}
\because \beta^7 &= \beta^1 \times \beta^6 \\
&= \beta^1(\beta^2 + \beta^1) \\
&= \beta^3 + \beta^2 \\
&= \beta^2 + \beta^0 + \beta^2 \\
&= \beta^0 = 1
\end{aligned}$$

$\therefore \beta^7 \in GF(2^3)$

#### 2.2.3.2 Digital Logic

Generating the rest of the symbols may be implemented using the following recurrence relation:

$$\begin{aligned}\alpha^{n+m} &= \alpha^{n+(m-1)} \times \alpha^{n} \\ &= \begin{cases} \alpha^{n+(m-1)} \ll 1 & \text{if } \alpha^{n+(m-1)}[n-1] = 0 \\ (\alpha^{n+(m-1)} \ll 1) \oplus \alpha^{n} & \text{if } \alpha^{n+(m-1)}[n-1] = 1 \end{cases}\end{aligned}$$

**Table 5:** The 8 Element Vectors of $x^3 + x^2 + x^0$ in $GF(2)[x]$

| Element | Symbol | Polynomial Form | Symbol |
|---|---|---|---|
| $0$ | (1) | $0+0+0$ | 000 |
| $\beta^0$ | 000 | $0+0+\beta^0$ | 001 |
| $\beta^1$ | 001 | $0+\beta^1+0$ | 010 |
| $\beta^2$ | 010 | $\beta^2+0+0$ | 100 |
| $\beta^3$ | 011 | $\beta^2+0+\beta^0$ | 101 |
| $\beta^4$ | 100 | $\beta^2+\beta^1+\beta^0$ | 111 |
| $\beta^5$ | 101 | $0+\beta^1+\beta^0$ | 011 |
| $\beta^6$ | 110 | $\beta^2+\beta^1+0$ | 110 |
| $\beta^7$ | (2) | $0+0+\beta^0$ | 001 |

## 2.3 Operations

The Galois Field Arithmetic Unit supports addition, subtraction, multiplication, division and logarithm of Galois terms.

### 2.3.1 Addition and Subtraction

Binary addition and binary subtraction are synonymous in the Galois Field. Addition and subtraction of Galois operands may be done by computing the bitwise exclusive disjunction of the operands.

$$\begin{aligned}\alpha^i \pm \alpha^j &= \{x_{i,n}, \dots x_{i,2}, x_{i,1}, x_{i,0}\} \pm \{x_{j,n}, \dots x_{j,2}, x_{j,1}, x_{j,0}\} \\ &= \{(x_{i,n} \oplus x_{j,n}), \dots, (x_{i,2} \oplus x_{j,2}), (x_{i,1} \oplus x_{j,1}), (x_{i,0} \oplus x_{j,0})\} \\ &= \alpha^k\end{aligned}$$

#### 2.3.1.1 Digital Design

Only polynomial forms of inputs are valid for these operations.

The implementation of Galois addition and subtraction may be computed with a single-level parallel array of XOR gates.

### 2.3.2 Multiplication

Binary multiplication of Galois operands is congruent to the sum of the indices of the operands. If the indices sum to greater than or equal to $2^n - 1$, then $2^n - 1$ is subtracted from the sum to prevent overflow.

$$\begin{aligned}\alpha^i \cdot \alpha^j &= \{x_{i,n-1}, \dots, x_{i,2}, x_{i,1}, x_{i,0}\} \cdot \{x_{j,n-1}, \dots, x_{j,2}, x_{j,1}, x_{j,0}\} \\ &= \alpha^{(i+j)\ mod\ (2^n-1)} \\ &= \begin{cases} \alpha^{(i+j)-(2^n-1)} & \text{if } (i+j) \geq 2^n - 1 \\ \alpha^{(i+j)} & \text{if } (i+j) < 2^n - 1 \end{cases}\end{aligned}$$

#### 2.3.2.1 Digital Design

Only element forms of inputs are valid for this operation.

Galois multiplication requires multiple binary additions and condition checks. To find the product of $\alpha^i$ and $\alpha^j$, a carry-lookahead adder may be used to sum the elements $i+j$. The most significant bits of the sum and the sum $+1$ will then be OR-ed to compute a single-bit control signal. The control signal will be multiplexed with a binary $1$ to be added to the sum.

$$\begin{aligned}\alpha^i \cdot \alpha^j &= \{x_{i,n-1}, \dots, x_{i,2}, x_{i,1}, x_{i,0}\} \cdot \{x_{j,n-1}, \dots, x_{j,2}, x_{j,1}, x_{j,0}\} \\ &\implies \text{Let } n = |i| = |j| \\ &= \begin{cases} \alpha^{(i+j)} & \text{if } \Big((i+j)[n] \ \vee \ (i+j+1)[n]\Big) = 0 \\ \alpha^{(i+j+1)} & \text{if } \Big((i+j)[n] \ \vee \ (i+j+1)[n]\Big) = 1 \end{cases}\end{aligned}$$

### 2.3.3 Division

Binary division of Galois operands is congruent to the difference of the indices of the operands. If the difference is negative, then the absolute value of the difference is subtracted from $2^n - 1$ to prevent underflow. If the difference is zero, then the quotient is $\alpha^0$.

$$\begin{aligned}\alpha^i/\alpha^j &= \{x_{i,n-1}, \dots x_{i,2}, x_{i,1}, x_{i,0}\}/\{x_{j,n-1}, \dots x_{j,2}, x_{j,1}, x_{j,0}\} \\ &= \alpha^{(i-j) \ mod \ (2^n-1)} \\ &= \begin{cases} \alpha^{(2^n-1)-(j-i)} & \text{if } (i-j) < 0 \\ \alpha^{(i-j)} & \text{if } (i-j) > 0 \\ \alpha^0 & \text{if } i = j, \ i \neq 0 \\ ERROR & \text{if } j = 0 \end{cases}\end{aligned}$$

#### 2.3.3.1 Digital Design

Only element forms of inputs are valid for this operation.

Galois division requires multiple binary additions and condition checks. To find the quotient of $\alpha^i$ and $\alpha^j$, the two's complement of $j$ has to first be summed with $i$. Converting to two's complement may be done with parallel inverters and a carry-lookahead adder. The overflow bit of the sum will then be used as a control signal to a multiplexer.

If activated, the multiplexer will add the two's complement of a binary $1$ to the sum.

$$\begin{aligned}
\alpha^i/\alpha^j &= \{x_{i,n-1},\ldots,x_{i,2},x_{i,1},x_{i,0}\}/\{x_{j,n-1},\ldots,x_{j,2},x_{j,1},x_{j,0}\} \\
&\implies \text{Let } n = |i| = |j| \\
&= \begin{cases} \alpha^{(i+j_{2's})} & \text{if } (i+j_{2's})[n] = 1 \\ \alpha^{(i+j_{2's}+1_{2's})} & \text{if } (i+j_{2's})[n] = 0 \end{cases}
\end{aligned}$$

### 2.3.4 Logarithm

Logarithm is considered a unary operation in the Galois Field, where only one operand is required. The base of the logarithm operation in the Galois Field is implicitly 2. The logarithm of a Galois operand is congruent to its index.

$$log_{GF(2)}(\alpha^i) = i$$

#### 2.3.4.1 Digital Design

Only element forms of inputs are valid for this operation.

The logarithm of a Galois operand is the index of the term itself.

## 2.4 Arithmetic Exceptions

### 2.4.1 Zero

Zero, or the null symbol, is the numerical zero used to fulfill its role as the additive identity in the Galois field. The following operations demonstrate the characteristics of zero in the Galois field.

#### 2.4.1.1 Digital Design

To handle zero, multiplexers are used to decode the intended operation to determine the forms if the inputs. Once the form is determined, the module raises a `zero flag`

**Table 6:** Operations with $\mathbf{0}$ in $GF(2)[x]$

| Operand 1 | Operation | Operand 2 | Output |
|---|---|---|---|
| $\mathbf{0}$ | $+/-$ | $\beta^x$ | $\beta^x$ |
| $\beta^x$ | $+/-$ | $\mathbf{0}$ | $\beta^x$ |
| $\mathbf{0}$ | $\times$ | $\beta^x$ | $\mathbf{0}$ |
| $\beta^x$ | $\times$ | $\mathbf{0}$ | $\mathbf{0}$ |
| $\mathbf{0}$ | $\div$ | $\beta^x$ | $\mathbf{0}$ |
| $\beta^x$ | $\div$ | $\mathbf{0}$ | ERROR |
| $\mathbf{0}$ | $\log_{GF(2)}$ | - | ERROR |

which is handled by individual operations to raise exceptions.

$$\texttt{zero flag} = \begin{cases} \wedge(\text{operand}), & \text{if operation} \in \{\text{addition, subtraction}\} \\ \vee(\text{operand}), & \text{if operation} \in \{\text{multiplication, division, logarithm}\} \end{cases}$$

### 2.4.2 Upper Bound

## References

(1) E. W. Weisstein. Finite field. *MathWorld – A Wolfram Web Resource*, Dec 2017.

(2) E. W. Weisstein. Primitive polynomial. *MathWorld – A Wolfram Web Resource*, Dec 2017.