Finite Field Arithmetic and Implementations

Xinmiao Zhang

Case Western Reserve University

Applications of Finite Field Arithmetic

- Error-correcting codes
  - Hamming codes
  - BCH codes
  - Reed-Solomon codes
  - Low-density parity-check codes

- Cryptosystems:
  - Elliptic curve cryptography — Public key cipher
  - Advanced Encryption Standard — Symmetric key cipher
  - Biometric encryption

Group

- A group is a set of objects ***G*** on which a binary operation "." is defined. The binary operation takes any two elements in ***G*** and generates as its result an element that is also in ***G*** . The operation must satisfy the following requirements if ***G*** is a group

   1. Associativity: (a . b) . c =a . (b . c) for all a, b, c $\in$***G***
   2. Identity: there exists e $\in$***G*** such that a . e =e . a=a for all a $\in$***G***
   3. Inverse: for each a $\in$***G*** there exists a unique element $a^{-1} \in$***G*** such that $a \cdot a^{-1} = a^{-1} \cdot a = e$

- A group is said to be commutative (or abelian) if it also satisfies

   4. Commutativity: for all a,b $\in$***G*** , a . b = b . a

Field

- Let ***F*** be a set of objects on which two operations + and . are defined, ***F*** is said to be a field iff
  1. ***F*** forms a commutative group under +, the additive identity element is labeled '0'.
  2. ***F***-{0} forms a commutative group under . The multiplicative identity element is labeled '1'.
  3. The operation + and . distributes: a . (b+c)=(a .b)+(a .c)

- A field with finite number of elements is called a finite field, also called Galois Field, denoted by GF(p). p can be a prime number or power of prime.

Examples of Finite Fields

- Finite field GF(2) consists of elements 0 and 1

  ‘+’ XOR operation

  additive identity: 0

  ‘.’ AND operation;

  multiplicative identity: 1

- Finite field GF(7) consists of elements 0,1, …6

  ‘+’ mod 7 integer addition

  additive identity: 0

  ‘.’ mod 7 integer multiplication

  multiplicative identity: 1

Irreducible Polynomial & Extension Field

- A polynomial $P(x) = p_m x^m + p_{m-1} x^{m-1} + \cdots + p_0$, whose coefficients p_i are elements of a field GF(q), is called a polynomial over GF(q).
- A polynomial P(x) is irreducible over GF(q) if P(x) is only divisible by c∈GF(q) or itself.
- P(x) can be used to construct *extension field* GF(q^m). Each element in GF(q^m) can be represented as a polynomial of degree m-1 over GF(q)

  ‘+’ polynomial addition

  ‘.’ polynomial multiplication modulo P(x)

Example of Extension Field

- $P(x)=x^3+x+1$ can be used to construct $GF(2^3)$. An element $A \in GF(2^3)$ can be expressed by three bits (a_2,a_1,a_0), which can be considered as the coefficients of a polynomial $A(x)=a_2x^2+a_1x+a_0$. Similarly, another element B can be expressed as: $B(x)=b_2x^2+b_1x+b_0$.

  '+' $A(x)+B(x)=(a_2 \oplus b_2)x^2 +(a_1 \oplus b_1)x +(a_0 \oplus b_0)$

  additive identity: 000

  '.' $A(x) . B(x) \bmod P(x)$

  multiplicative identity: 001

- $\{x^2, x, 1\}$, where x is a root of $P(x)$, is called a standard (polynomial) basis of $GF(2^3)$.

Representations of Finite Field Elements

- Standard basis
- Normal basis
- Dual basis
- Power representation

  For an element $\alpha \in GF(q^m)$, the minimum integer r, such that $\alpha^r=1$, is called the order of α. The maximum order of any element is q^m-1, and an element with order q^m-1 is called a primitive element. All the nonzero elements of $GF(q^m)$ can be represented by the powers of a primitive element α

$$\{1, \alpha, \alpha^2, \cdots \alpha^{q^m-2}\}$$

- An irreducible polynomial whose roots have order q^m-1 is called a primitive polynomial

Representation Conversion

- Binary extension field, $GF(2^m)$, is usually adopted for hardware implementations, since each element can be represented by a m-bit binary tuple.
- $P(x)=x^3+x+1$ is a primitive polynomial, and can be used to construct $GF(2^3)$.

Power	Standard basis
0	000
1	001
α	010
α^2	100
α^3	011
α^4	110
α^5	111
α^6	101

- α, x: root of P(x)

$\alpha^2 \sim 100 = x^2$

$\alpha^3 \sim x^2 \cdot x = x^3 \xrightarrow{\text{modulo } P(x)} x+1$

$\alpha^4 \sim 110 = x^2+x$

$\alpha^5 \sim (x^2+x) \cdot x = x^3+x^2 \xrightarrow{\text{modulo } P(x)} x^2+x+1$

Composite Field

- Two pairs $\{GF(q^n),\ G(y) = y^n + \sum_{i=0}^{n-1} g_i y^i\}$ and $\{GF((q^n)^m),\ P(x) = x^m + \sum_{i=0}^{m-1} p_i x^i\}$ are called a composite field if
  - $GF(q^n)$ is constructed from $GF(q)$ by $G(y)$.
  - $GF((q^n)^m)$ is constructed from $GF(q^n)$ by $P(x)$.
- A composite field $GF((q^n)^m)$ is *isomorphic* to the field $GF(q^k)$ with $k=nm$.
- Each element of $GF((q^n)^m)$ can be expressed as m elements of $GF(q^n)$

Implementation of Finite Field Arithmetic

Finite Field Addition

- Additions over GF(2^m) using basis representations can be performed by bit-wise XOR operation
- Additions over GF(2^m) using power representation need look-up tables of size $2^m \times m$

Example: GF(2^3) can be constructed using irreducible polynomial $P(x)=x^3+x+1$

0	000	α^3	011
1	001	α^4	110
α	010	α^5	111
α^2	100	α^6	101

$\alpha^2+\alpha^4 \rightarrow 100+110$

$=010 \rightarrow \alpha$

Finite Field Multipliers

Multiplication using Power Representation

- Using power representation, multiplication over GF(2^m) can be implemented by adding up the exponents of the operands modulo 2^m-1.

Example: multiplications over GF(2^3)

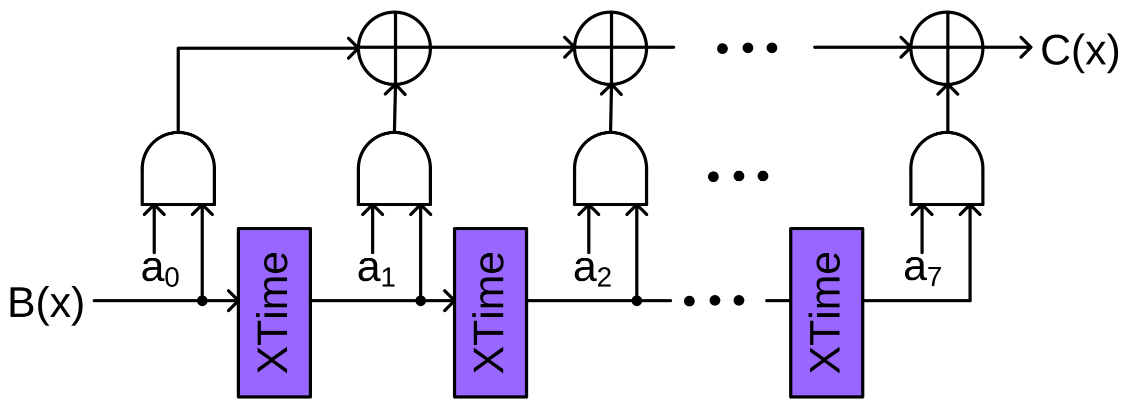
$$\alpha^5 \cdot \alpha^6 = \alpha^{(5+6) \bmod 7} = \alpha^4$$

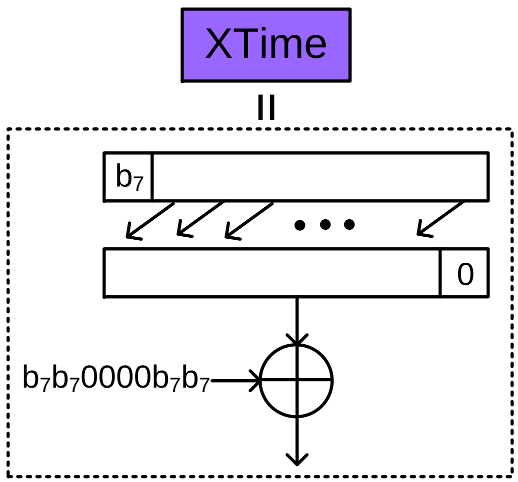
- Implementation: m-bit unsigned integer addition with the carry-out added to the least significant bit.

Multiplication using Standard Basis

- The product of A(x) and B(x) is C(x)=A(x)B(x) mod P(x)

$$C(x) = a_0 B(x) \bmod P(x) + a_1 (xB(x)) \bmod P(x) + \cdots + a_{m-1}(x^{m-1}B(x)) \bmod P(x)$$





- Xtime: multiply an element by x modulo P(x)

$P(x)=x^8+ x^7 + x^6 +x+1$

Multiplication over Composite Field

- In composite field $GF((2^n)^m)$, the elements are represented by polynomials with maximum degree m-1 over $GF(2^n)$

$$A(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots a_0 \qquad a_i \in GF(2^n)$$

- The complexity of composite field multiplication can be reduced by the Karatsuba-Ofman Algorithm (KOA), which reduces the number of sub-field multiplications at the cost of more additions.

Multiplication over Composite Field

- C(x)=A(x)B(x)

$$A(x) = x^{\frac{m}{2}}(x^{\frac{m}{2}-1}a_{m-1} + \cdots + a_{\frac{m}{2}}) + (x^{\frac{m}{2}-1}a_{\frac{m}{2}-1} + \cdots + a_0) = x^{\frac{m}{2}}A_h(x) + A_l(x)$$

$$B(x) = x^{\frac{m}{2}}(x^{\frac{m}{2}-1}b_{m-1} + \cdots + b_{\frac{m}{2}}) + (x^{\frac{m}{2}-1}b_{\frac{m}{2}-1} + \cdots + b_0) = x^{\frac{m}{2}}B_h(x) + B_l(x)$$

define

$$D_0(x) = A_l(x)\,B_l(x)$$

$$D_1(x) = [A_l(x) + A_h(x)][B_l(x) + B_h(x)]$$

$$D_2(x) = A_h(x)\,B_h(x)$$

$$C(x) = x^m D_2(x) + x^{\frac{m}{2}}[D_1(x) - D_0(x) - D_2(x)] + D_0(x)$$

- The number of element multiplications is reduced

$m^2 \rightarrow 3/4m^2$

Finite Field Inverters

Inversion Using Look-up Table & Power representation

- Inversions over $GF(2^m)$ can be implemented by look-up tables of size $2^m \times m$
- Assuming an element $A \in GF(2^m)$ can be expressed in power representation as $A=\alpha^b$, where α is a primitive element of $GF(2^m)$ and $0 \leqslant b < 2^m - 1$

$$A^{-1} = \alpha^{-b} = \alpha^{-b} \times \alpha^{2^m - 1} = \alpha^{2^m - 1 - b}$$

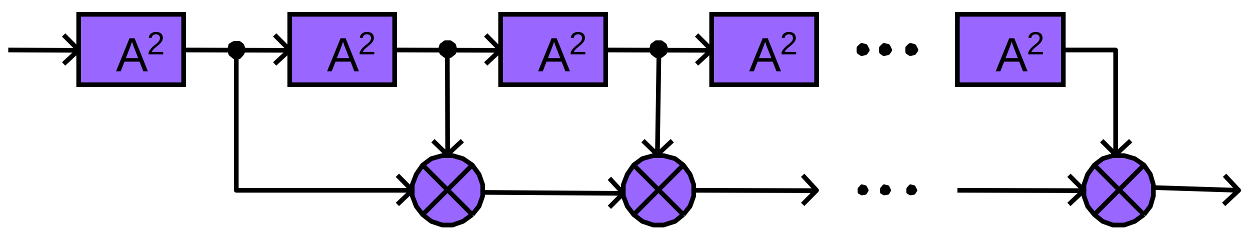
Example: for $A=\alpha^6 \in GF(2^4)$, $A^{-1}=\alpha^{16-1-6}=\alpha^9$

- Implementation: m-bit integer subtraction

Inversion using Multiply-square

- For an element A∈GF(2^m) (A≠0), since $A^{2^m-1}=1$

$$A^{-1}=A^{-1+(2^m-1)}=A^{2^m-2}=A^2\cdot A^4\cdots A^{2^{m-1}}$$



- Compared to a general multiplier, a squarer can be implemented by simpler architecture
- Has high complexity and long latency

Inversion over Composite Field GF($(2^m)^2$)

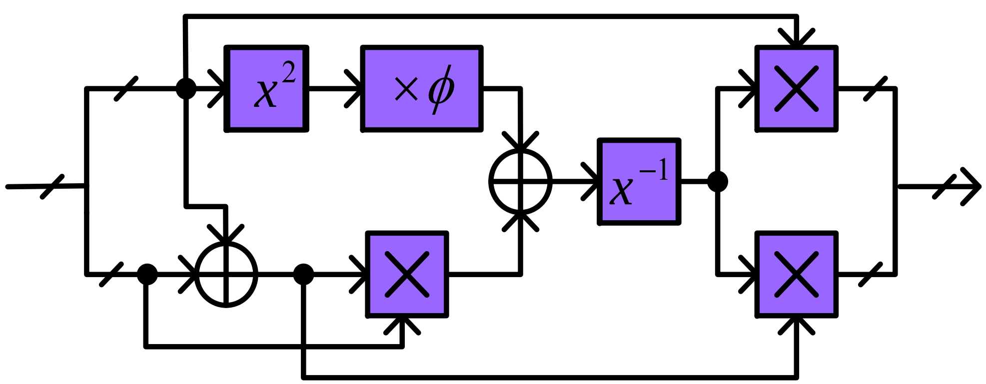
- The inversion over GF($(2^m)^2$) can be converted to computations over GF(2^m)

- An element A $\in$ GF($(2^m)^2$) can be represented as

$$A(x) = a_0 + a_1 x, \quad a_0, a_1 \in GF(2^m)$$

- Assume that the irreducible polynomial used to construct GF($(2^m)^2$) from GF(2^m) is $P(x) = x^2 + x + \phi$. Apply the Extended Euclidean algorithm

$$A^{-1}(x) = a_1(a_0(a_0 + a_1) + \phi a_1^2)^{-1} x + (a_0 + a_1)(a_0(a_0 + a_1) + \phi a_1^2)^{-1}$$



Example of Finite Field Inverter over $GF(2^8)$

Complexity of Constant Multiplier over GF(2^8)

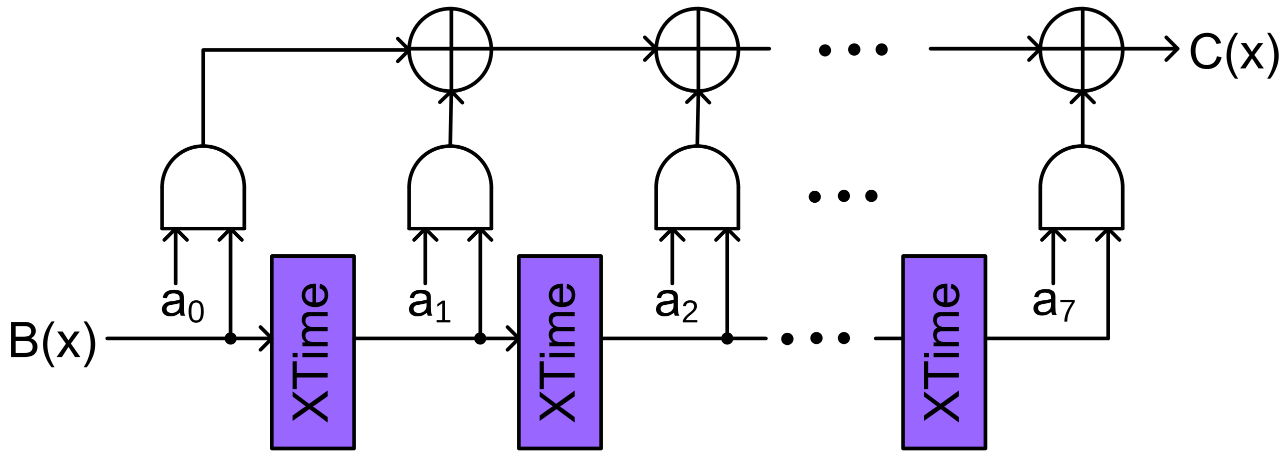
- Assuming irreducible polynomial $P(x)=x^8+x^4+x^3+x+1$ is used to construct GF(2^8), the coefficients of $C=A^2$ can be computed as:

c7=a7+a6	c3=a7+a6+a5+a4
c6=a5+a3	c2=a5+a1
c5=a6+a5	c1=a7+a6+a4
c4=a7+a4+a2	c0=a6+a4+a0

Complexity: 11 XOR gate

Critical path: 2 XOR gate

Complexity of Standard Basis Multiplier

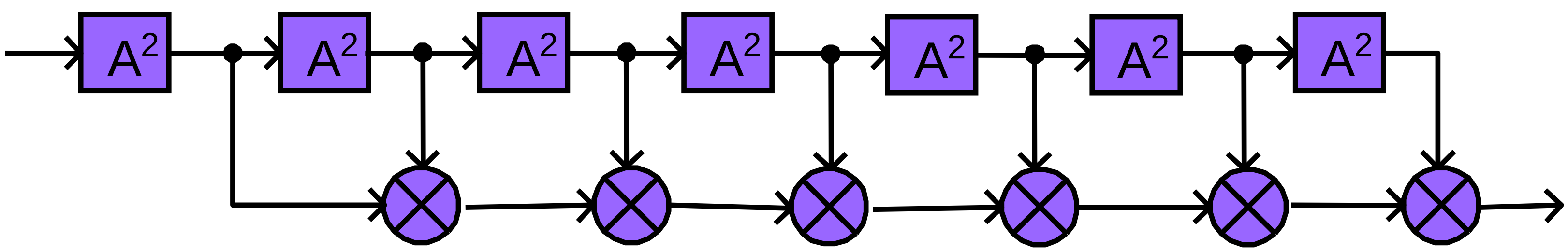


Complexity: 77 XOR, 64 AND
Critical path: 10 gates

Inverter Based on Multiply-square

Approach A

$$A^{-1} = A^{254} = A^2 A^4 A^8 A^{16} A^{32} A^{64} A^{128}$$



# of XOR: 539

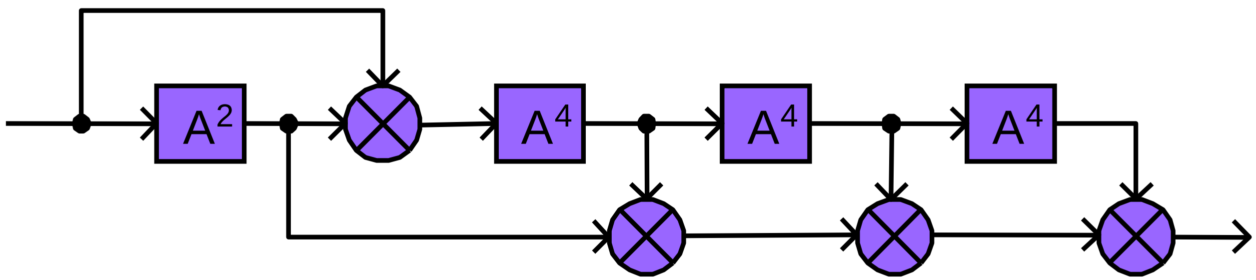
# of AND: 384

Critical path: 64 gates

Inverter Based on Multiply-square

Approach B

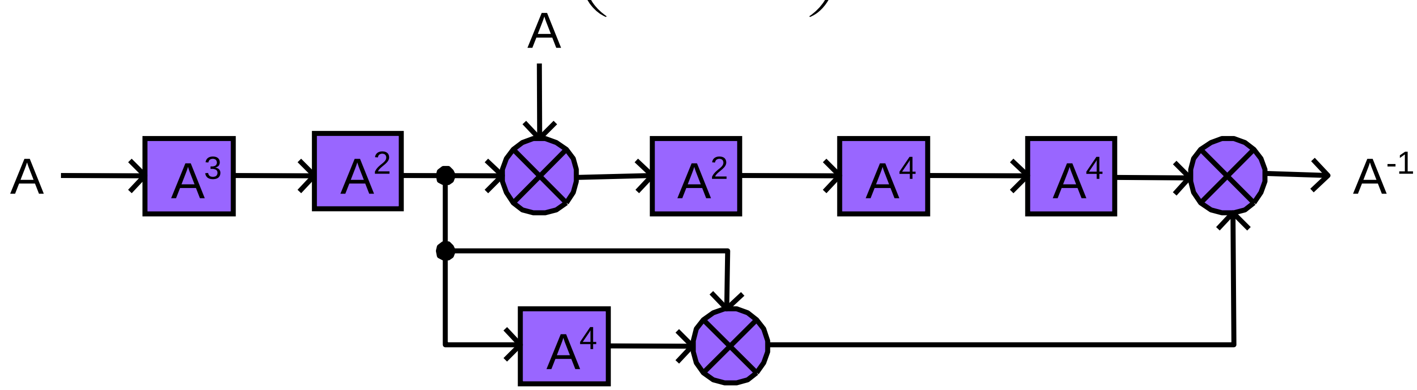
$$A^{-1} = A^2 (A^3)^4 (A^3)^{4^2} (A^3)^{4^3}$$



- # of XOR: 11*(1+2*3)+77*4=385
- # of AND: 64*4=256
- Critical path: 2+10+2*2+3*10=46 gates

Inverter Based on Multiply-square (C)

Approach C $$A^{-1} = \left(\left(\left(A^7 \right)^2 \right)^4 \right)^4 \left(\left(A^3 \right)^2 \right)^4 \left(A^3 \right)^2$$



- A^3:

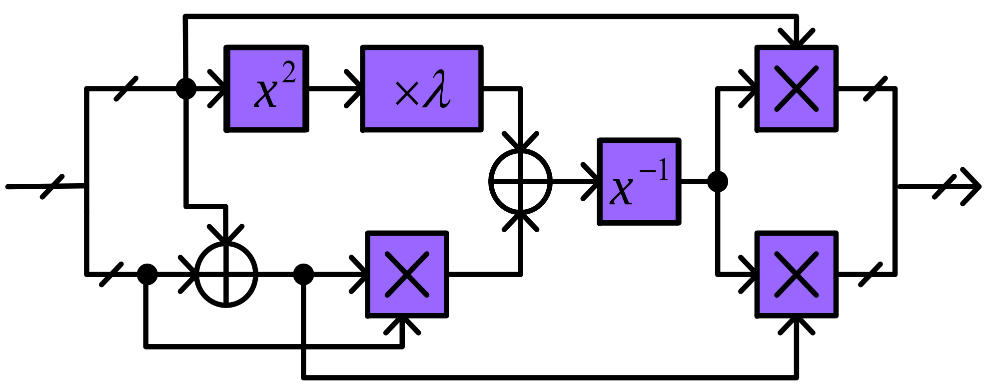
  # of XOR: 75, # of AND: 40, Critical path: 6 gates

- # of XOR: 75+2*11+3*2*11+3*77=394
- # of AND: 40+3*64=232
- Critical path: 6+2*2+2*10+2*2*2=38 gates

Inverter based on Composite Field

- The following irreducible polynomials can be used to construct the composite field of $GF(2^8)$:
  - $GF(2) \rightarrow GF(2^2)$: $P0(x)=x^2+x+1$
  - $GF(2^2) \rightarrow GF((2^2)^2)$: $P1(x)=x^2+x+\phi$
  - $GF((2^2)^2) \rightarrow GF(((2^2)^2)^2)$: $P2(x)=x^2+x+\lambda$

$\phi = \{10\}_2,\ \lambda = \{1100\}_2$



Inverter Based on Composite Field

- Multiplications over $GF(2^4)$ can be further decomposed into $GF(2^2)$, then into $GF(2)$.
- Multiplications over $GF(2)$ are simply AND operations.

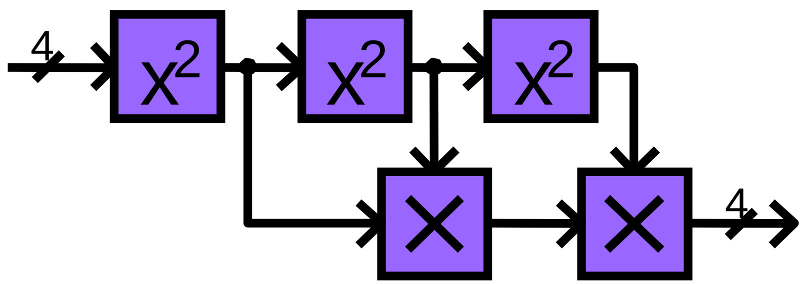
$$\phi = \{10\}_2, \ \lambda = \{1100\}_2$$

Block	# of gates	Critical path
$\times \lambda$	3 XOR	2 XOR
x^2	4 XOR	2 XOR
Multiplier in $GF(2^2)$	4 XOR + 3 AND	2 XOR + 1 AND
Multiplier in $GF(2^4)$	21 XOR +9 AND	4 XOR + 1 AND

Implementation of Inversion over GF(2^4)

Approach D: Inversion over GF(2^4) can be implemented by multiply-square

for $A \in GF(2^4)$, $A^{-1} = A^{14} = A^2 \cdot A^4 \cdot A^8$

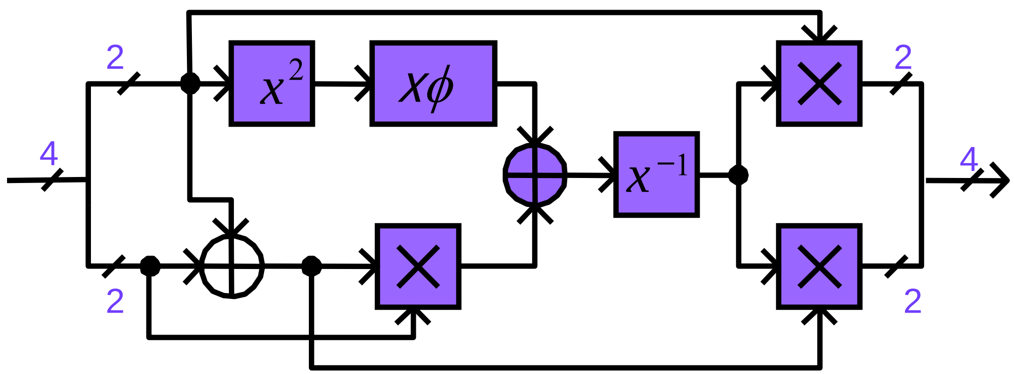


- # of XOR: 54
- # of AND: 18
- Critical path: 14 gates

Implementation of Inversion over GF(2^4)

Approach E: Inversion over GF(2^4) can be further decomposed into inversion over GF($(2^2)^2$) using the Extended Euclidean algorithm

$$A^{-1}(x) = (a_0 + a_1)(a_0(a_0 + a_1) + \phi a_1^2)^{-1} + a_1(a_0(a_0 + a_1) + \phi a_1^2)^{-1} x$$



- # of XOR: 17
- # of AND: 9
- Critical path: 9 gates

Implementation of Inversion over GF(2^4)

Approach F: The bits in $A^{-1}=\{a_3^{-1}, a_2^{-1}, a_1^{-1}, a_0^{-1}\}$ can be expressed in terms of the bits in $A=\{a_3, a_2, a_1, a_0\}$ by following the computations done by each of the blocks in the previous figure

$$a_3^{-1} = a_3 + a_3a_2a_1 + a_3a_0 + a_2$$

$$a_2^{-1} = a_3a_2a_1 + a_3a_2a_0 + a_3a_0 + a_2 + a_2a_1$$

$$a_1^{-1} = a_3 + a_3a_2a_1 + a_3a_1a_0 + a_2 + a_2a_0 + a_1$$

$$a_0^{-1} = a_3a_2a_1 + a_3a_2a_0 + a_3a_1 + a_3a_1a_0 + a_3a_0 + a_2 + a_2a_1 + a_2a_1a_0 + a_1 + a_0$$

- # of XOR: 14
- # of AND: 9
- Critical path: 5 gates

Complexities of Inverters over $GF(2^8)$

Approach	# of XOR	# of AND	Critical path
A	539	384	64
B	385	256	46
C	394	232	38
D	159	45	34
E	122	36	29
F	119	36	25