

# Vector space of bit vectors

Ralf Poeppel

`mailto:ralf@poeppel-familie.de`

2026-02-04

## Abstract

This article is a supplementary documentation to the Go package `gf2vs` [10]. The package implements data types and functions for the vector space of bit vectors.

## 1 Introduction

Bit vectors are very common in computer science. They are used in for integers, combinatorial algorithms, coding theory and for logical and arithmetic operations [7, 5]. All aspects of the vector space of bit vectors are examined relatively rarely.

Bits are based on the the finite set of integers of order 2. This set  $\{0, 1\}$  with the operations addition and multiplication modulo 2 satisfy the axioms of a field. This finite field is named Galois Field [8, 15]<sup>1</sup>  $GF(2) = \mathbb{F}_2$ . On this field there is the vector space  $\mathbb{F}_2^n$ .

The aim of this article is to document the properties of the vector space of bit vectors  $\mathbb{F}_2^n$ , as implemented in the Go package `gf2vs`. This is a brief reference of the properties collected from several sources.

## 2 Field $GF(2)$

### 2.1 Supporting Set

The supporting set of  $GF(2) = \mathbb{F}_2$  is

$$\mathbb{Z}_2 = \mathbb{Z}/\mathbb{Z}2 = \{0, 1\} \subset \mathbb{Z} \subset \mathbb{R} \quad (1)$$

the subset  $\mathbb{Z}_2$  of  $\mathbb{Z}$ , which is a subset of  $\mathbb{R}$ . This set is equal to  $\mathbb{Z}/\mathbb{Z}2$  the cyclic set of order 2. This set hold the values of a bit in computer science. In logic we have the boolean values False  $F = 0$  and True  $T = 1$  [6].

### 2.2 Operations

The operations of  $\mathbb{F}_2 = \mathbb{Z}/\mathbb{Z}2$  are defined modulo 2 see [2] ch. 2.2.6 and [5].

The operations addition and multiplication of the field  $\mathbb{F}_2$  satisfies the group axioms [2, 8, 16], both operations are commutative.

---

<sup>1</sup>We cite Wikipedia for reused wordings.

Similarly the operations addition and multiplication of the field  $\mathbb{R}$  satisfies the group axioms [2], both operations are commutative.

Please note the different definition of the addition in  $\mathbb{F}_2$  and  $\mathbb{R}$ .

For reference we give here only the operations for  $\mathbb{F}_2$ .

### 2.2.1 Negation

$$- : \mathbb{F}_2 \rightarrow \mathbb{F}_2, \quad -x = x, \quad -0 = 0, \quad -1 = 1. \quad (2)$$

The negation of 1 in  $\mathbb{F}_2$  is computed as  $(-1) \bmod 2 = 1$

### 2.2.2 Complement

$$\neg : \mathbb{F}_2 \rightarrow \mathbb{F}_2, \quad \neg x = 1 - x, \quad \neg 0 = 1, \quad \neg 1 = 0. \quad (3)$$

### 2.2.3 Absolut-value

We define the mapping absolute-value  $|x|$  of an element  $x$  of  $\mathbb{F}_2$  to  $\mathbb{R}$ :

$$|x| : \mathbb{F}_2 \rightarrow \mathbb{R}, \quad |0| = 0, \quad |1| = 1. \quad (4)$$

We use this mapping, when we need the default classical definition of the addition as in  $\mathbb{R}$ .

### 2.2.4 Addition

The addition is named exclusive disjunction in logic and XOR [6, 14] in computer science. The definition of addition is given in equation 5 obeying  $(1 + 1) \bmod 2 = 0$ .

$$+ : \mathbb{F}_2 \times \mathbb{F}_2 \rightarrow \mathbb{F}_2, \quad 0 + 0 = 0, \quad 0 + 1 = 1, \quad 1 + 0 = 1, \quad 1 + 1 = 0, \quad (5)$$

The group axioms [2] ch. 2.2.8 for the Group  $G = \mathbb{F}_2$  and the operation addition are satisfied:

#### Associativity

$$\forall a, b, c \in G : (a + b) + c = a + (b + c).$$

#### Identity element $e = 0$

$$\exists e \in G, \forall a \in G : e + a = a \text{ and } a + e = a, e = 0, e \text{ is unique.}$$

#### Inverse element $(-a) = a$

$$\forall a \in G \exists (-a) \in G : a + (-a) = e \text{ and } (-a) + a = e, e \text{ identity element, } (-a) = a \text{ is unique for each } a.$$

#### Commutativity

$$a + b = b + a.$$

So  $\mathbb{F}_2$  with the operation addition is an abelian group.

### 2.2.5 Multiplication

The multiplication is named conjunction in logic and AND [6, 18] in computer science. The multiplication is identically defined as in  $\mathbb{Z}$ .

$$\cdot : \mathbb{F}_2 \times \mathbb{F}_2 \rightarrow \mathbb{F}_2, \quad 0 \cdot 0 = 0, \quad 0 \cdot 1 = 0, \quad 1 \cdot 0 = 0, \quad 1 \cdot 1 = 1. \quad (6)$$

We may use the notation  $ab$  instead of  $a \cdot b$ , omitting the multiplication sign if there is no ambiguity.

The group axioms for the Group  $G = \mathbb{F}_2$  and the operation multiplication are satisfied:

#### Associativity

$$\forall a, b, c \in G : (a \cdot b) \cdot c = a \cdot (b \cdot c).$$

#### Identity element $e = 1$

$$\exists e \in G, \forall a \in G : e \cdot a = a \text{ and } a \cdot e = a, e = 1, e \text{ is unique.}$$

#### Inverse element $a^{-1} = a$

$$\forall a \in G, a \neq 0, \exists a^{-1} \in G : a \cdot a^{-1} = e \text{ and } a^{-1} \cdot a = e, e \text{ identity element, } a^{-1} = 1 \text{ is the only inverse element.}$$

#### Commutativity

$$a \cdot b = b \cdot a.$$

So  $\mathbb{F}_2$  with the operation multiplication is an abelian group.

### 2.2.6 Disjunction

In boolean logic we have the operation disjunction, named OR in computer science [6, 19].

$$\vee : \mathbb{F}_2 \times \mathbb{F}_2 \rightarrow \mathbb{F}_2, \quad 0 \vee 0 = 0, \quad 0 \vee 1 = 1, \quad 1 \vee 0 = 1, \quad 1 \vee 1 = 1 \quad (7)$$

The operation  $\vee$  does not satisfy the group axioms; there is no inverse element.

## 2.3 Field axioms

The set  $K := \mathbb{F}_2$  with the operations addition and multiplication satisfies the field axioms [2, 5]. We use  $K$  as symbol for any field satisfying the field axioms.

**K1**  $K$  with the addition  $+$  is an abelian group.

**K2**  $K^* := K \setminus \{0\}$  with the multiplication  $\cdot$  for every element of  $K^*$  is an abelian group.

**K3** distributive property [1, 12] is satisfied  $\forall a, b, c \in K$

$$\begin{aligned} a \cdot (b + c) &= a \cdot b + a \cdot c, \\ (a + b) \cdot c &= a \cdot c + b \cdot c. \end{aligned} \quad (8)$$

### 3 Vector space $\mathbb{F}_2^n$

#### 3.1 Vectors

Bit vectors are the elements of the vector space. We define a bit vector  $x$  of size  $n$  as a tuple  $(x_i)$  of values  $x_i \in \mathbb{F}_2$ :

$$x := (x_i) := (x_1, x_2, \dots, x_n), \quad \forall x_i \in \mathbb{F}_2 \quad (9)$$

We define the set of all bit vectors of size  $n$  see [2] 2.4.1:

$$\mathbb{F}_2^n := \{x = (x_1, \dots, x_n) : x_i \in \mathbb{F}_2\} \quad (10)$$

In addition we define two distinguished constant elements of  $\mathbb{F}_2^n$ :

**Zero**  $\mathbf{0}$  zero vector, all components are 0.

**Ones**  $\mathbf{1}$  ones vector, all components are 1.

#### 3.2 Operations on vectors

We define bitwise operations on the bit vectors  $x, y, z$  see [2] 2.4.1 and [7] (1), (2), (3).

$$\left. \begin{array}{lll} - : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n, & -x = x & \Leftrightarrow & -x_i = x_i \\ \sim : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n, & \sim x = y & \Leftrightarrow & \neg x_i = y_i \\ | : \mathbb{F}_2^n \rightarrow \mathbb{R}^n, & |x| = y & \Leftrightarrow & x_i = y_i, \\ \oplus : \mathbb{F}_2^n \times \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n, & x \oplus y = z & \Leftrightarrow & x_i + y_i = z_i, \\ \& : \mathbb{F}_2^n \times \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n, & x \& y = z & \Leftrightarrow & x_i \cdot y_i = z_i, \\ | : \mathbb{F}_2^n \times \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n, & x|y = z & \Leftrightarrow & x_i \vee y_i = z_i, \\ \cdot : \mathbb{F}_2 \times \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n, & \lambda \cdot x = y & \Leftrightarrow & \lambda \cdot x_i = y_i, \end{array} \right\} i = 1, \dots, n. \quad (11)$$

We define the operation  $|$  for formal mapping of a bit vector from  $\mathbb{F}_2^n$  to  $\mathbb{R}^n$ .

For the constants we have:  $\sim \mathbf{0} = \mathbf{1}$  and  $\sim \mathbf{1} = \mathbf{0}$ .

We adopt the main identities from [7] (4),  $\dots$ , (14) for bit vectors of size  $n$  here:

$$x \oplus y = y \oplus x, \quad x \& y = y \& x, \quad x|y = y|x; \quad (12)$$

$$(x \oplus y) \oplus z = x \oplus (y \oplus z), \quad (x \& y) \& z = x \& (y \& z), \quad (x|y)|z = x|(y|z); \quad (13)$$

$$(x \oplus y) \& z = (x \& z) \oplus (y \& z); \quad (14)$$

$$(x \& y)|z = (x|z) \& (y|z), \quad (x|y) \& z = (x|z) \& (y|z); \quad (15)$$

$$x \oplus y = (x \& y) \oplus (x|y); \quad (16)$$

$$(x \& y)|x = x, \quad (x|y) \& x = x; \quad (17)$$

$$x \oplus \mathbf{0} = x, \quad x \& \mathbf{0} = \mathbf{0}, \quad x|\mathbf{0} = x; \quad (18)$$

$$x \oplus x = \mathbf{0}, \quad x \& x = x, \quad x|x = x; \quad (19)$$

$$x \oplus \mathbf{1} = \sim x, \quad x \& \mathbf{1} = x, \quad x|\mathbf{1} = \mathbf{1}; \quad (20)$$

$$x \oplus (\sim x) = \mathbf{1}, \quad x \& (\sim x) = \mathbf{0}, \quad x|(\sim x) = \mathbf{1}; \quad (21)$$

$$-(x \oplus y) = (\sim x) \oplus y = x \oplus (\sim y), \quad \sim (x \& y) = (\sim x)|(\sim y), \quad \sim (x|y) = (\sim x) \& (\sim y); \quad (22)$$

### 3.3 Axioms of vector space

The set  $\mathbb{F}_2^n$  with the binary operation of vector addition  $\oplus$  and the binary function of scalar multiplication  $\cdot$ , as given in (11), defines a vector space see [2, 21].

The axioms of a vector space are satisfied for  $\mathbb{F}_2^n$ :

**Associativity of vector addition**

$$u \oplus (v \oplus w) = (u \oplus v) \oplus w, \quad \forall u, v, w \in \mathbb{F}_2^n.$$

**Commutativity of vector addition**

$$u \oplus v = v \oplus u, \quad \forall u, v \in \mathbb{F}_2^n.$$

**Identity element of vector addition**

$$\exists 0 \in \mathbb{F}_2^n : v \oplus 0 = v, \quad \forall v \in \mathbb{F}_2^n.$$

**Inverse elements of vector addition**

$$\forall v \in \mathbb{F}_2^n \exists -v \in \mathbb{F}_2^n : v \oplus (-v) = 0, \text{ and } -v = v, \text{ i.e. each vector is its own additive inverse.}$$

**Compatibility of scalar multiplication with field multiplication**

$$\lambda(\eta v) = (\lambda\eta)v, \quad \lambda, \eta \in \mathbb{F}_2, \quad v \in \mathbb{F}_2^n.$$

**Identity element of scalar multiplication**

$$1v = v, \quad 1 \in \mathbb{F}_2, \quad v \in \mathbb{F}_2^n, \text{ where } 1 \text{ is the multiplicative identity of } \mathbb{F}_2.$$

**Distributivity of scalar multiplication with respect to vector addition**

$$\lambda(u \oplus v) = \lambda u \oplus \lambda v, \quad \lambda \in \mathbb{F}_2, \quad u, v \in \mathbb{F}_2^n.$$

**Distributivity of scalar multiplication with respect to field addition**

$$(\lambda + \eta)v = \lambda v \oplus \eta v, \quad \lambda, \eta \in \mathbb{F}_2, \quad v \in \mathbb{F}_2^n.$$

In the vector space  $\mathbb{F}_2^n$  we are not limited to the operations vector addition and scalar multiplication. We can use the Boolean operations as well.

**Negation, Complement, Not**

$$\sim v = \mathbb{1} - v = \mathbb{1} \oplus v, \text{ swap all bits.}$$

**Disjunction, Or**

$$u|v = (u_i)|(v_i) = (u_i \vee v_i), \text{ element-wise Or.}$$

**Exclusive or, Xor**

$$u \oplus v = u \oplus v = (u_i) \oplus (v_i) = (u_i + v_i), \text{ element-wise Xor, equal to vector addition.}$$

**Conjunction, And**

$$u \wedge v = (u_i) \cdot (v_i) = (u_i \cdot v_i), \text{ element-wise And.}$$

As we apply the operations element-wise  $i = 1, \dots, n$ , we satisfy the laws of associativity and commutativity inherited from the field.

### 3.4 Vector space base

We give here the definition of the base, the norm and the scalar product implemented in the Go package. The symbol  $K$  is used in definitions applicable by each of the fields  $\mathbb{F}_2^n, \mathbb{Q}, \mathbb{R}, \mathbb{C}$ .

### 3.4.1 Unit vector

We define the unit vectors  $e_i, i = 1, \dots, n$ , of the vector space as the vectors where the  $i$ th element is  $x_i = 1$  and all other elements are 0.

$$e_i = (x_k), \quad e_i \in K^n, \quad x_k \in K, \quad x_k = \begin{cases} 1, & k = i, \quad \text{identity element of multiplication,} \\ 0, & k \neq i, \quad \text{identity element of addition.} \end{cases} \quad (23)$$

Please note the  $e_i$  are linearly independent.

We observe the identity elements are identical for the fields and the unit vectors are identical for all vector spaces over a field  $K$ .

### 3.4.2 Generating system

We define the subset  $\mathbb{E} := \{e_i\}, \mathbb{E} \subset \mathbb{F}_2^n$ , of unit vectors  $e_i$ . The subset  $\mathbb{E}$  forms a generating system. Each vector  $v$  of  $\mathbb{F}_2^n$  is a linear combination of scalars  $a_i$  and the  $e_i$ :

$$v = \sum_{i=1}^n a_i e_i, \quad a_i \in \mathbb{F}_2, \quad e_i \in \mathbb{E}, \quad \forall v \in \mathbb{F}_2^n. \quad (24)$$

Here the addition is modulo 2.

Equation (24) is used equally for each vector space on any field  $K$  using the operation addition as defined for the field  $K$  and the 1 the identity element of the operation multiplication.

Thus the subset  $\mathbb{E}$  spans  $\mathbb{F}_2^n$ . In this vector space it is one spanning set, and the decomposition of a vector  $v$  into a linear combination of unit vectors  $e_i$  is unique.

### 3.4.3 Base

The subset  $\mathbb{E}$  is one base of the vector space  $\mathbb{F}_2^n$ . As it is a base of every vector space over a field  $K$ .

### 3.4.4 Index

We call  $i = 1, \dots, n$  the index of the unit vector  $e_i$  in the base.

### 3.4.5 Norm via Hamming weight

The addition in the field  $\mathbb{F}_2^n$  is the addition modulo 2. Hence each summation gives one of the values 0 or 1.

$$\sum_i^n x_i = \begin{cases} 0, & n \bmod 2 = 0, \quad n \text{ even,} \\ 1, & n \bmod 2 = 1, \quad n \text{ odd.} \end{cases} \quad (25)$$

So each sum will have either the value 0 or the value 1. From this it follows we cannot use any usual definition of a norm for computing the length of a vector in  $\mathbb{F}_2^n$ .

In coding theory [4, 17] the *Hamming weight*  $w_H$  (number of 1-entries) of  $x \in \{0, 1\}^n$  is defined as:

$$w_H : \mathbb{F}_2^n \rightarrow \mathbb{Z}, \quad w_H(x) = |\{i \in \{1, \dots, n\} : x_i = 1\}|, \quad (26)$$

and the associated *Hamming distance*  $d_H$  is:

$$d_H : \mathbb{F}_2^n \times \mathbb{F}_2^n \rightarrow \mathbb{Z}, \quad d_H(x, y) = w_H(x - y). \quad (27)$$

If we first apply operation  $| \cdot |$  we map a vector from  $\mathbb{F}_2^n \rightarrow \mathbb{R}^n$ , by embedding  $\mathbb{F}_2^n = \{0, 1\}^n \subset \mathbb{R}^n$ . On vectors in  $\mathbb{R}^n$  norms are defined and we get:

$$\|x\|_1 = \sum_{i=1}^n |x_i| = w_H(x). \quad (28)$$

So the  $l_1$ -norm on  $\mathbb{R}^n$  see [3, 11] equals the Hamming weight on bit vectors. This norm is sometimes called absolute-value norm [20]. The value of the norm is an element of the set  $\{0, 1, \dots, n\} \subset \mathbb{Z} \subset \mathbb{R}$ .

In the programming languages C the function is named `popcount()` and in the language Go it is the function `OnesCount(uint x) uint` in package `math/bits`.

Obviously this norm satisfies the axioms of a norm:

#### Subadditivity / Triangle inequality

$$w_H(x + y) \leq w_H(x) + w_H(y) \quad \forall x, y \in \mathbb{F}_2^n,$$

#### Absolute homogeneity

$$w_H(s \cdot x) = s \cdot w_H(x) \quad \forall s \in \mathbb{F}_2, x \in \mathbb{F}_2^n,$$

#### Positive definiteness

$$w_H(x) = 0 \Rightarrow x = 0.$$

Please note the norm of the unit vectors  $\|e_i\| = 1, \forall i \in \{1, \dots, n\}$ .

### 3.4.6 Scalar product

We define the scalar product, dot product [2, 13] or inner product of two vectors as given in equation (29):

$$\langle \cdot, \cdot \rangle: \mathbb{F}_2^n \times \mathbb{F}_2^n \rightarrow \mathbb{R}, \quad \langle x, y \rangle = \sum_i^n |x_i \cdot y_i| = \|x \& y\| = w_H, \quad (29)$$

The obtained value equals the standard scalar product of  $x, y \in \{0, 1\}^n \subset \mathbb{R}^n$

$$\langle \cdot, \cdot \rangle: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}, \quad \langle x, y \rangle = \sum_i^n x_i \cdot y_i, \quad (30)$$

We can easily verify by computation the properties of the scalar product:

#### Distributivity

$$\begin{aligned} \langle x \oplus x', y \rangle &= \langle x, y \rangle + \langle x', y \rangle, \\ \langle x, y \oplus y' \rangle &= \langle x, y \rangle + \langle x, y' \rangle, \\ \langle \lambda x, y \rangle &= \lambda \langle x, y \rangle, \\ \langle x, \lambda y \rangle &= \lambda \langle x, y \rangle, \end{aligned}$$

#### Commutativity

$$\langle x, y \rangle = \langle y, x \rangle,$$

#### Positive definiteness

$$\langle x, x \rangle \geq 0, \text{ with } \langle x, x \rangle = 0 \text{ only if } x = 0,$$

#### Orthogonality

$$\langle x, y \rangle = 0, \text{ we say the two vectors are orthogonal.}$$

In coding theory [9] the orthogonal bit vector is called the dual code.

### 3.4.7 Orthonormal Basis

With the properties of the norm and the scalar product it follows  $\mathbb{E}$  is an orthonormal base, and this is the only orthonormal base of  $\mathbb{F}_2^n$ .

## References

- [1] Sheldon Jay Axler. *Linear Algebra Done Right*. 4th ed. Cham: Springer Nature; 2024. ISBN: 9783031410260. URL: <https://linear.axler.net/LADR4e.pdf> (visited on 02/03/2026).
- [2] Gerd Fischer and Boris Springborn. *Lineare Algebra: Eine Einführung für Studienanfänger*. de. 19th ed. Wiesbaden, Germany: Springer Spektrum, 2020.
- [3] Otto Forster. *Analysis / Otto Forster ; 1: Differential- und Integralrechnung einer Veränderlichen*. Wiesbaden: Springer Spektrum; 2016. ISBN: 9783658115449. URL: <https://www.tib.eu/de/suchen/id/TIBKAT%3A842657355>.
- [4] R. W. Hamming. “Error detecting and error correcting codes”. In: *Bell System Technical Journal* 29.2 (1950), pp. 147–160. URL: <https://dn710109.ca.archive.org/0/items/bstj29-2-147/bstj29-2-147.pdf> (visited on 02/03/2026).
- [5] Raymond Hill. *A first course in coding theory*. Oxford: Clarendon Press; 2004. ISBN: 0198538030. URL: <https://sites.math.rutgers.edu/~zeilberg/akherim/RHcoding.pdf> (visited on 01/29/2026).
- [6] Donald E. Knuth. *The Art of Computer Programming, Vol 4, Fasc 0. Introduction to Combinatorial Algorithms and Boolean Functions*. Vol. 4 Fascicle 0. Addison-Wesley, 2008. URL: <https://www-cs-faculty.stanford.edu/~knuth/fasc0a.ps.gz> (visited on 01/17/2018).
- [7] Donald E. Knuth. *The Art of Computer Programming, Vol 4, Fasc 1. Bitwise Tricks and Techniques*. Vol. 4 Fascicle 1. Addison-Wesley, 2008. URL: <https://www-cs-faculty.stanford.edu/~knuth/fasc1a.ps.gz> (visited on 10/08/2025).
- [8] Rudolf Lidl and Harald Niederreiter. *Finite fields*. Vol. 20. Encyclopedia of mathematics and its applications, volume 20. Cambridge: Cambridge University Press; 2000. ISBN: 9780511525926. DOI: 10.1017/CB09780511525926. URL: [https://api.pageplace.de/preview/DT0400.9780511832468\\_A23680063/preview-9780511832468\\_A23680063.pdf](https://api.pageplace.de/preview/DT0400.9780511832468_A23680063/preview-9780511832468_A23680063.pdf) (visited on 01/29/2026).
- [9] Jacobus H. Lint. *Introduction to Coding Theory and Algebraic Geometry*. Vol. 12. Oberwolfach seminars, 12. Basel: Birkhäuser Basel; 1988. ISBN: 9783034892865. DOI: 10.1007/978-3-0348-9286-5. URL: <https://theswissbay.ch/pdf/Gentoomen%20Library/Information%20Theory/Coding%20Theory/Introduction%20To%20Coding%20Theory%20And%20Algebraic%20Geometry%20-%20Jacobus%20Van%20Lint.pdf> (visited on 02/03/2026).
- [10] Ralf Poeppel. *Go package documentation gf2vs*. <https://pkg.go.dev/github.com/rpoe/gf2vs>. [Online; accessed 10-January-2026]. Jan. 6, 2026.
- [11] Eric W. Weisstein.  *$L^1$  – Norm From MathWorld—A Wolfram Resource*. <https://mathworld.wolfram.com/L1-Norm.html>. [Online; accessed 09-October-2025]. July 27, 2025.
- [12] Wikipedia contributors. *Distributive property — Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Distributive\\_property&oldid=1329322251](https://en.wikipedia.org/w/index.php?title=Distributive_property&oldid=1329322251). [Online; accessed 28-January-2026]. 2025.
- [13] Wikipedia contributors. *Dot product — Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Dot\\_product&oldid=1328631745](https://en.wikipedia.org/w/index.php?title=Dot_product&oldid=1328631745). [Online; accessed 2-February-2026]. 2025.
- [14] Wikipedia contributors. *Exclusive or — Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Exclusive\\_or&oldid=1316886803](https://en.wikipedia.org/w/index.php?title=Exclusive_or&oldid=1316886803). [Online; accessed 12-January-2026]. 2025.
- [15] Wikipedia contributors. *Finite field — Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Finite\\_field&oldid=1330855394](https://en.wikipedia.org/w/index.php?title=Finite_field&oldid=1330855394). [Online; accessed 12-January-2026]. 2026.



- [16] Wikipedia contributors. *Group (mathematics)* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Group\\_\(mathematics\)&oldid=1330839314](https://en.wikipedia.org/w/index.php?title=Group_(mathematics)&oldid=1330839314). [Online; accessed 12-January-2026]. 2026.
- [17] Wikipedia contributors. *Hamming weight* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Hamming\\_weight&oldid=1306107874](https://en.wikipedia.org/w/index.php?title=Hamming_weight&oldid=1306107874). [Online; accessed 13-January-2026]. 2025.
- [18] Wikipedia contributors. *Logical conjunction* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Logical\\_conjunction&oldid=1324909528](https://en.wikipedia.org/w/index.php?title=Logical_conjunction&oldid=1324909528). [Online; accessed 12-January-2026]. 2025.
- [19] Wikipedia contributors. *Logical disjunction* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Logical\\_disjunction&oldid=1317551960](https://en.wikipedia.org/w/index.php?title=Logical_disjunction&oldid=1317551960). [Online; accessed 12-January-2026]. 2025.
- [20] Wikipedia contributors. *Norm (mathematics)* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Norm\\_\(mathematics\)&oldid=1326013131](https://en.wikipedia.org/w/index.php?title=Norm_(mathematics)&oldid=1326013131). [Online; accessed 14-January-2026]. 2025.
- [21] Wikipedia contributors. *Vector space* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Vector\\_space&oldid=1326882436](https://en.wikipedia.org/w/index.php?title=Vector_space&oldid=1326882436). [Online; accessed 12-January-2026]. 2025.