Xpriori Technical Paper

Finite Fields and Properties of the Xpriori Icon Generator, Associative Processing Unit, and Associative Memory Controller used in Digital Pattern Processing (DPP™)

by Harry Direen

Release 1.1

NOTE: In October 2003, Xpriori, LLC acquired NeoCore Holdings, LLC including all technology and patents. Any references to Neo, NeoCore or NeoCore Holdings, LLC technology or patents as such are now the property of Xpriori, LLC.

Xpriori, LLC
2864 S. Circle Dr.
Ste. 401
Colorado Springs, CO 80906
(719) 425-9840
www.xpriori.com

Version 1.1

INTRODUCTION

Key to the NeoCore Digital Pattern Processing^{TM} (DPP) is the linear algebraic coding scheme used in the Icon Generator (IG). Fundamentally the Icon generator consists of a linear feedback shift register implemented with tables to maximize performance. The coding scheme was designed to possess a number of special features:

- Fixed Length output. Typically the IG produces a fixed length code of 64 bits.
- Chaotic distribution. No matter what the input data consists of, or the pathology it possesses, the IG will produce numbers that are evenly spread over the range of values the Icon length can represent. In particular, blocks of sequential input values are guaranteed to be splattered across the Icon field.
- Perfect transformation. The IG produces a perfect transform. If one instance of every possible 64-bit value are transformed the resulting Icons will be, likewise, one instance of every possible 64-bit value – only in a different order. No two inputs will produce the same Icon value if the inputs are less than or equal to 64-bits. Longer length input data will result in an imperfect transformation, which is still useful in most all applications.
- Fractal transformation. If a shorter Icon length is desired (32-bit, for example), a subset of the 64-bit code can be used as a shorter Icon. The transform will still be perfect within it's shorter field, so perfect codes of any size can be produced without having to re-program the IG.
- Extensible transformation. The IG can also be used to produce codes of more than 64 bits. Longer codes can be produced piecemeal that possess the same qualities as the 64-bit code. Codes of any desired length can be produced without re-programming the IG.
- Transformation of variable length input data. The IG process input data one byte at a time. The Icon can be read at any time, so codes based on any input data length(s) can be produced.
- Support for Icon Algebra. Connected to the IG is the Associative Processing Unit (APU). Icons can be processed combinatorially, producing the same result had original input data been manipulated before transformation. For example, if "Albert" is transformed, and then "Einstein"; the tow resulting Icons can be combined to produce the same Icon that would have been produced from "AlbertEinstein". The APU can accomplish a variety of virtual functions including: concatenation (as in the example above); removal of data from the beginning or end of input data; combination (XOR) of input data; or changing of components of input data. The primary benefit of "Icon Algebra" is that the APU accomplishes these functions quickly on short fixed-length Icons in lieu of re-processing input data. The Icons "stand in" for arbitrarily large blocks of input data.

The mathematical description of the Icon Generator is based on finite fields. The vectors of $0s$ and $1s$ are made into polynomials over the simple two element field $\{0,1\}$. For example, the vector

$$v = (0,1,1,1)$$

becomes the polynomial

$$f(x) = 0x + x + x^2 + x^3.$$

These polynomials are divided by a fixed irreducible polynomial, each input leaving a remainder which is in effect the Icon for that polynomial. The space of inputs is thus neatly partitioned into disjoint classes, called residue classes. The set of all these classes is another finite field, an extension of the field $\{0, 1\}$. There is an addition $+$ and a multiplication $\cdot$ in this *Residue Class Field* which behaves in many respects just as addition and multiplication in the field of rational numbers. The operations allow algebraic manipulations of Icons which greatly facilitates recognition and analysis.

In the chapters which follow, this paper develops the rudiments of the theory of finite fields and its application to the Icon Generator. The following Table of Contents outlines the paper.

Chapter One

Algebraic Context and Background

[1.1] Modular Arithmetic; the Integers $\mathrm{mod}(d)$. The set of integers is denoted

$$Z = \{\cdots, -2, -1, 0, 1, 2, \cdots\}.$$

Think of d as a fixed positive integer, $d > 1$. If we divide an integer n by d then we can write

$$n = qd + r, \quad q, r \in Z \text{ and } 0 \le r \le d - 1. \qquad [1.1]$$

In this notation d is the divisor, q is the quotient, and r is the remainder. This formula, and the procedure for finding q and r for given d, is called the *division algorithm* for the integers. Each remainder is in the set

$$Z_d = \{0, 1, 2, \cdots, d - 1\}.$$

Define a function φ from Z to Z_d by

$$\varphi(n) = r.$$

If $\varphi(m) = \varphi(n)$, then m and n are said to be *congruent modulo* d, written

$$n \equiv m \,\mathrm{mod}(d).$$

The function φ partitions the integers into disjoint sets called *equivalence classes*, namely for each integer $r \in Z_d$ the equivalence class determined by r is

$$\langle r \rangle = \{n \in Z : \varphi(n) = r\}.$$

We have

$$Z = \cup_{r=0}^{d-1} \langle r \rangle.$$

For example if $d = 5$ and $n = 37$ then

$$37 = 7 \cdot 5 + 2; \; d = 7, \; r = 2;$$

$$\varphi(37) = 2; \quad 37 \equiv 2 \,\mathrm{mod}(5).$$

Since $17 = 3 \cdot 5 + 2$,

$$\varphi(17) = 2 \quad \text{and } 17 \equiv 37 \,\mathrm{mod}(5).$$

Taking a negative n,

$$-4 = (-1) \cdot 5 + 1 \text{ and } -4 \equiv 1 \,\mathrm{mod}(5).$$

Addition $\oplus$ and multiplication $\odot$ are defined in the set Z_d by dividing the usual sum and product in Z by d and using the remainder for the definitions of $\oplus$ and $\odot$. For example, in Z_5 we have

$$2 + 4 = 6 = 1 \cdot 5 + 1, \text{ and so } 2 \oplus 4 = 1;$$

$$2 \cdot 4 = 8 = 1 \cdot 5 + 3, \text{ and so } 2 \odot 4 = 3.$$

In terms of the function φ the definitions for Z_d are

$$r \oplus s = \varphi(r + s), \quad r \odot s = \varphi(r \cdot s).$$

The element $0 \in Z_d$ is an identity for $\oplus$ and 1 is an identity for $\odot$ because

$$0 \oplus r = \varphi(0 + r) = r,$$
$$1 \odot r = \varphi(1 \cdot r) = r.$$

Using this notation, the addition and multiplication tables for Z_5 are

$\oplus$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

and

$\odot$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Each operation is easily seen to be commutative because

$$s \oplus r = \varphi(s + r) = \varphi(r + s) = r \oplus s$$

and similarly with $\odot$.

The mapping φ satisfies

$$\varphi(n + m) = \varphi(n) \oplus \varphi(m), \qquad [1.3]$$
$$\varphi(n \cdot m) = \varphi(n) \odot \varphi(m).$$

That is, φ is a *homomorphism* for both $+$ and $\cdot$. To prove the second, write

$$n = d_n q + r_n, \; m = d_m q + r_m$$

so

$$n \cdot m = (d_n d_m q + d_n r_m + d_m r_n) q + r_n r_m$$

and

$$\varphi(n \cdot m) = \varphi(r_n \cdot r_m) = r_n \odot r_m = \varphi(n) \odot \varphi(m).$$

Each operation inherits associativity from Z; the calculation for $\odot$ is

$$r \odot (s \odot t) = r \odot \varphi(s \cdot t) = \varphi(r) \odot \varphi(s \cdot t) = \varphi(r \cdot (s \cdot t)) = \varphi((r \cdot s) \cdot t)$$
$$= \varphi(r \cdot s) \odot \varphi(t) = \varphi(r \cdot s) \odot t = (r \odot s) \odot t$$

The distributive law for the two operations in Z_d is

$$r \odot (s \oplus t) = (r \odot s) \oplus (r \odot t).$$

To prove it, calculate:

$$r \odot (s \oplus t) = r \odot \varphi(s + t) = \varphi(r) \odot \varphi(s + t) = \varphi(r \cdot (s + t))$$
$$= \varphi(r \cdot s + r \cdot t) = \varphi(r \cdot s) \oplus \varphi(r \cdot t) = (r \odot s) \oplus (r \odot t).$$

[1.2] Groups and Rings. A set G with a binary operation $*$ defined on pairs of elements of G is a *group* if

1. $g * h \in G$ *for all* $g, h \in G$
2. *idenity* $e \in G$ *satisfying* $e * g = g * e = g$ *for all* $g \in G$
3. *associative* : $g * (h * k) = (g * h) * k$ *for all* $g, h, k \in G$
4. *inverses* *for all* $g \in G$, *there is unique* $h \in G$ *satisfying* $g * h = h * g = e$.

[1.4]

The integers Z form a group under the usual addition $+$ in which the identity is 0 and the inverse of n is $-n$. The usual multiplication in the nonzero integers

$$Z^* = Z \backslash \{0\}$$

satisfies conditions $1, 2, 3$ with $e = 1$, but does not satisfy condition 4. Thus Z^* is not group under normal multiplication.

The rational numbers Q form a group under addition and the nonzero rationals

$$Q^* = Q \backslash \{0\}$$

form a group under multiplication.

By the calculations in section [1.1], $(Z_d, \oplus)$ is a group for each positive integer d. Letting

$$Z_d^* = Z_d \backslash \{0\},$$

the pair $(Z_d^*, \odot)$ is not, in general, a group. For example, the multiplication table for $(Z_4^*, \odot)$ is

$\odot$	1	2	3
1	1	2	3
2	2	0	2
3	3	2	1

and we see that 2 has no inverse, so $(Z_4^* \odot)$ is not a group. Indeed, $0 \notin Z_4^*$. However, $(Z_5^*, \odot)$ is a group, having multiplication table

$\odot$	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

.

To verify from the table that $\odot$ satisfies definition [1.2], note that properties 1, 2, and 4 are all apparent from the table. The associativity of $\odot$ was proved in general in section [1.1].

In [1.6] we will prove that $(Z_p^*, \odot)$ is a group for any prime number p.

A *ring* is a set R with 2 binary operatons $\boxplus$ and $\boxdot$ such that there is an identity 0 for addition $\boxplus$ and

1. $(R, \boxplus)$ *is a group*
2. $0 \boxdot x = 0$ *for all* $x \in R$
3. *the distributive law* $m \boxdot (n \boxplus k) = (m \boxdot n) \boxplus (m \boxdot k)$ *holds* for all $m, n, k \in R$

[1.5]

The sets with operations

$$(Z,+,\bullet),\ (Q+,\bullet),\ (Z_d,\oplus,\odot)$$

are all rings. In each 1 is an identity for multiplication.

We have used the symbols $\boxplus$ and $\boxdot$ for the abstract definition of a ring. But we will drop them from now on and use $+$ and $\bullet$ in the general case. When clarity demands it we will continue to use $\oplus$ and $\odot$ for Z_d.

[1.3] The Order of an Element in a Group or Ring. If $(G,+)$ is a group and n is a positive integer, then

$$nx = x + x + \cdots x = sum\ of\ n\ x's.$$

A recursive definition is

$$1x \doteq x$$

$$nx \doteq (n-1)x + x.$$

Note that the juxtaposition of n and x in the notation nx is not a group operation. If there is an n for which

$$nx = 0 \qquad [1.6]$$

then x is said to have *finite order* and the order of x is the smallest positive integer for which $[1.6]$ holds. For example, in Z_5 the order of 1 is 5. In fact, it is easy to check that the order of every nonzero element in Z_5 is 5.

If the binary operation is written multiplicatively, then instead of nx we have x^n. For example, the multiplicative order of 1 in $(Z_5^*,\bullet)$ is 1, because

$$1^1 = 1.$$

The multiplicative order of 2 is found by

$$2^1 = 2,\ 2^2 = 4,\ 2^3 = 3,\ 2^4 = 1.$$

Thus the multiplicative order of 2 is 4.

[1.4] The Characteristic of a Finite Ring If $(R,+,\bullet)$ is a finite ring, $x \in R$, and $N = \{1,2,3\cdots\}$, then the set

$$\{nx : n \in N\}$$

is finite and so there are positive integers m and n for which $nx = mx$, say $m < n$. Then we have

$$(n-m)x = 0.$$

Thus each $x \in R$ has finite additive order, say $o(x)$. It follows that there is a smallest integer s for which $sx = 0$ for all $x \in R$. The smallest such integer is defined as the characteristic of the ring.

In the case of the ring $(Z_p,\oplus,\odot)$ for prime p,

$$px = 0$$

for all $x \in Z_p^*$ because px is divisible by p; that is, $px \equiv 0 \bmod(p)$. Thus the additive characteristic of Z_p is less than or equal to p. If $1 \le k < p$ and $x \in Z_p^*$ satisfies

$$kx = 0,$$

then $k \odot x = 0$. Thus $\varphi(k \cdot x) = 0$, and so $k \cdot x$ is a multiple of p and so p must divide either k or x; but it does not. Thus

$$\textit{characteristic of } (Z_p, \oplus) = p.$$

In the case of Z_4,

$$o(0) = 1,\ o(1) = 4,\ o(2) = 2,\ o(3) = 4.$$

Thus $\text{char}(Z_4) = 4$, but not all elements have the same order.

[1.5] Fields. A *field* $(F, +, \cdot)$ is a ring with an identity 1 for $\cdot$ and for which $(F^*, \cdot)$ is a group. Each of

$$(Q +, \cdot),\ (R +, \cdot),\ (C +, \cdot),\ (Z_5, \oplus, \odot)$$

is a field, where *Q* is the rational numbers and *R* is the real numbers. The complex numbers *C* are defined as the set

$$C = \{x + iy : x, y \in R\}$$

where

$$i^2 = -1.$$

The fields *Q*, *R*, *C* are, of course, infinite fields. The field $(Z_5, \oplus, \odot)$ is a *finite field;* that is, a field with a finite number of elements, 5 elements in this case.

A subset K of F which is closed under the operations of $(F, +, \cdot)$ and is itself a field with these operations is called a *subfield*. The field F is also called an extension field of the field K. For example, *Q* is a subfield of *R* and *R* is a subfield of *C*; *C* is an extension of *R*; *R* is an extension of *Q*.

[1.6] Theorem. If p is a prime number, then $(Z_p, \oplus, \odot)$ is a field.

Proof: Since $(Z_p, \oplus, \odot)$ is a ring with identity 1 for $\odot$, all that is required is to show that $(Z_p^*, \odot)$ is a group. Properties 2 and 3 of [1.2] are valid because $(Z_p, \oplus, \odot)$ is a ring. For fixed $n \in Z_p^*$, the elements in the set

$$\{k \odot n : k \in Z_p\}$$

are all distinct, for if

$$k \odot n = m \odot n,$$

then

$$(k - m) \odot n = 0.$$

Thus the prime p divides the product

$$(k - m) \cdot n.$$

Since $1 \le n < p$, p does not divide n and so p must divide $(k - m)$. Since

$$-p < k - m < p,$$

we conclude that $k - m = 0$ and $k = m$. Thus as k ranges over all of Z_p there are p distinct elements $k \odot n \in Z_p$. Exactly one element $k \odot n$ is 1; this gives the inverse $k = n^{-1}$ of n. Exactly one element $k \odot n$ is 0 and this must be $0 = 0 \odot n$. Thus $k \odot n \in Z_p^*$ if $k, n \in Z_p^*$. Thus conditions 1 and 4 of [1.2] hold, completing the proof that $(Z_p^*, \odot)$ is a group with

identity 1, and therefore that $(Z_p, \oplus, \odot)$ is a field.

[1.7] Theorem. The characteristic of a finite field F is a prime p, and the set

$$F_p = \{k1 : k \in Z_p\}$$

is a subfield of F which isomorphic to the field $(Z_p, \oplus, \odot)$.

Proof. If the characteristic is a composite

$$n = km$$

then for all $x \in F^*$,

$$0 = (km)x = k(mx) = k(y \cdot x)$$

where

$$y = m1.$$

If $y \neq 0$ then the elements $y \cdot x$ range over all of F^* as x ranges over F^*, because F is a field. Thus $kz = 0$ for all $z \in F$ and so $k = n$ and $m = 1$. Thus n is not composite, it is prime, say $n = p$. If $y = 0$, then 1 has order m or less, and for any x

$$mx = m(1 \cdot x) = (m \cdot 1)x = 0$$

and so characteristic of F is m or smaller. So, $m = n$, $k = 1$ and again n is not composite.

Since $p1 = 0$ and $k1 \neq 0$ if $1 \leq k < p$, there are exactly p elements in F_p. The mapping

$$\psi(k) = k1$$

is an isomorphism of $(Z_p, \oplus, \odot)$ onto F_p. In particular, F_p is a field.

[1.8] The Field Z_2. This is the simplest field, having tables

+	0	1
0	0	1
1	1	0

,

·	0	1
0	0	0
1	0	1

.

[1.9] Products of Groups and Rings. If $(G, +)$ and $(H, +)$ are commutative groups, their product group is defined as the set

$$G \times H = \{(g, h) : g \in G,\ h \in H\}$$

with operation

$$(g, h) + (g', h') = (g + g', h + h').$$

It is easily verified that $(G \times H, +)$ is a group.

If $(G, +, \cdot)$ and $(H, +, \cdot)$ are rings, then $\cdot$ in $G \times H$ is defined as

$$(g, h) \cdot (g', h') = (g \cdot g', h \cdot h').$$

It is easily verified that $(G \times H, +, \cdot)$ is a ring.

As an example, the addition table for $Z_2 \times Z_2$ is

$+$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,0)$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,1)$	$(0,1)$	$(0,0)$	$(1,1)$	$(1,0)$
$(1,0)$	$(1,0)$	$(1,1)$	$(0,0)$	$(0,1)$
$(1,1)$	$(1,1)$	$(1,0)$	$(0,1)$	$(0,0)$

and the multiplication table for $[Z_2 \times Z_2]^*$ is

$\cdot$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,1)$	$(0,1)$	$(0,0)$	$(0,1)$
$(1,0)$	$(0,0)$	$(1,0)$	$(1,0)$
$(1,1)$	$(0,1)$	$(1,0)$	$(1,1)$

Note that

$$Z_2^* = \{1\},\ Z_2^* \times Z_2^* = \{(1,1)\}$$

so

$$[Z_2 \times Z_2]^* \neq Z_2^* \times Z_2^*.$$

The product $Z_2 \times Z_2$ is a ring but not a field. The elements $(0,1)$ and $(1,0)$ have no multiplicative inverses.

[1.10] Linear Algebra for Fields. We begin with the field R of real numbers. The set R^n is all vectors

$$x = (x_1, x_2, \cdots x_n) = (x_j)_{j=1}^n, \quad x_j \in R.$$

The sum of vectors x and y is defined coordinate-wise as

$$x + y = (x_j + y_j)_{j=1}^n.$$

The set $(R^n, +)$ is a commutative group. The *scalar product* of $a \in R$, $x \in R^n$ is

$$ax = (ax_j)_{j=1}^n. \qquad [1.9]$$

With these operations R^n is a *vector space over* R.

The field R can be replaced with any field F. The axioms for a vector space V over F are as follows. First there is a binary operation $+$ on V making $(V, +)$ a commutative group. There is scalar product

$$(a, v) \rightarrow av$$

mapping $F \times V \rightarrow V$ satisfying

1. $a(u+v) = au + av,$
2. $(a+b)v = av + bv.$

The set of vectors F^n with coordinate addition and the scalar product $[1.9]$ is a vector space over F.

A finite set of vectors $\{v_j : 1 \leq j \leq n\}$ is *linearly dependent* if one of the vectors is a linear combination of the others, say

$$v_1 = \sum_{j=2}^{n} a_j vj$$

for $a_j \in F$. It is *linearly independent* if it is not linearly dependent. Equivalently, it is linearly independent if the equality

$$\sum_{j=1}^{n} a_j vj = 0$$

for $a_j \in F$ implies that $a_j = 0,$ all j. An infinite set is linearly independent if every finite subset is linearly independent.

For a set $S \subset V$ the *span* of S is

$$span(S) = \text{ Finite Linear Combinations of Elements of } S$$
$$= \left\{ \sum_{j=1}^{n} a_j v_m : a_j \in F,\ v_j \in S,\ n \in N \right\}.$$

A basis for V is a linearly independent set whose span is all of V :

$$span(B) = V.$$

It can be proved that any two basis' for V have the same number of elements. The vector space V is finite dimensional of dimension n if V has a basis of size n. If it is not finite dimensional then it is infinite dimensional.

We illustrate these ideas for vector spaces over the field Z_2. First consider

$$Z_2^2 = \{(x_1, x_2) : x_j \in Z_2\}$$
$$= \{(0,0),(0,1),(1,0),(1,1)\}.$$

The standard basis for Z_2^2 is

$$B = \{(0,1),(1,0)\}$$

but each of

$$B' = \{(0,1),(1,1)\},\ \ B'' = \{(1,0),(1,1)\}$$

is also a basis.

The vector space Z_2^n is all vectors of length n in which each entry is 0 or 1. The dimension of Z_2^n is n and the standard basis is

$$B = \{\beta_k : 1 \le k \le n\},$$
$$(\beta_k)_j = \left\{ \begin{array}{ll} 0 & if \quad j \neq k \\ \\ 1 & if \quad j = k \end{array} \right\}.$$

This vector space has 2^n elements. If $n = 64,$ it has

$$2^{64} = 18,446,744,073,709,551,616 = 1.844674407370\,96 \times 10^{19}$$

elements.

The vector space Z_2^{∞} is all infinite seqences of 0 and $1,$ namely

$$Z_2^{\infty} = \{(x_j) : x_j \in \{0,1\}\}.$$

The subset

$$[Z_2]^{\aleph_o} = \{(x_j) : x_j \in \{0,1\} \text{ and } \exists\ n \in N \text{ for which } x_j = 0 \text{ if } j > n\}$$

is also a vector space. Letting β_k be defined as above but for all $k \geq 1$, the set

$$B = \{\beta_k : 1 \leq k\}$$

is a basis for $[Z_2]^{\aleph_o}$. The basis B is countably infinite and the vector space $[Z_2]^{\aleph_o}$ is countably infinite.

Note that B is not a basis for Z_2^∞. The vector space Z_2^∞ as an infinite set has the same cardinality as the real numbers, because every real number has a dyadic expansion. The space Z_2^∞ has a basis but it is not constructable, and therefore of little use.

Chapter Two

Polynomial Rings

[2.1] The Ring of Polynomials over a Field F. For this section the field F is arbitrary, finite or infinite. Polynomials are functions on F to F of the form

$$P(x) = \sum_{j=0}^{n} a_j x^j$$

in which $a_j \in F$. The polynomial has degree n if $a_n \neq 0$. It is *monic* if $a_n = 1$. Addition of polynomials is "by addition of coefficients" within the field F, namely

$$(P+Q)(x) = P(x) + Q(x) = \sum_{j=0}^{n} a_j x^j + \sum_{j=0}^{n} b_j x^j = \sum_{j=0}^{n} (a_j + b_j) x^j.$$

The zero polynomial is the field element 0; it is clearly an identity for polynomial addition. Letting the set of polynomials be denoted $F[x]$ it is easy to see that $(F[x], +)$ is a group. Polynomial multiplication is

$$P \cdot Q(x) = \sum_{j=0}^{n} a_j x^j \cdot \sum_{k=0}^{m} b_j x^j = a_n b_m + \cdots + c_p x^p + \cdots + a_0 b_0,.$$

where

$$c_p = \sum_{j+k=p} a_j b_k.$$

The field element $1 = 1x^0$ is an identity for $\cdot$ and it is straightforward to verify that $(F[x], +, \cdot)$ is a ring. Note that the elements of the field F are polylnomials of degree 0.

The group $(F[x], +)$ is also a vector space over F in which the scalar product is

$$(a, P) \rightarrow aP,$$

taking $F \times F[x] \rightarrow F[x]$. The product aP can be viewed as polynomial multiplication $a \cdot P$ or as multiplication of each coefficient of P by a in the field multiplication. The properties for a vector space, properties 1 and 2 of section [1.10], are easily verified.

The polynomial P *factors* into polynomials Q and R if

$$P = Q \cdot R$$

and

$$1 \leq \deg(Q) < \deg(P), \quad 1 \leq \deg(R) < \deg(P).$$

For example, the factorization

$$x^2 - 1 = (x-1)(x+1)$$

holds for any field. If P has no factors then it is *irreducible*. The field element r is a root of the polynomial equation

$$P(x) = 0$$

if $P(r) = 0$.

[2.2] Examples for the rational field Q, the real field R, and the complex field C. The polynomial

$$P(x) = x^2 - 2$$

has coefficients in Q, and so it is in each of

$$Q[x], R[x], C[x]$$

because

$$Q \subset R \subset C.$$

It factors in $R[x]$ as

$$x^2 - 2 = (x - \sqrt{2})(x + \sqrt{2}),$$

but it does not factor in $Q[x]$ because $\sqrt{2} \notin Q$. It follows that P is reducible in $R[x]$ but it is irreducible in $Q[x]$.

The polynomial

$$Q(x) = x^2 + 1$$

factors in $C[x]$ as

$$x^2 + 1 = (x - i)(x + i)$$

but it does not factor in $R[x]$ or $Q[x]$, so it is irreducible in $R[x]$ and $Q[x]$.

The Fundamental Theorem of Algebra states that every polynomial equation

$$P(x) = 0$$

for $P \in C[x]$ has a root in C.

[2.3] Examples in Z_2. For $Z_2 = \{0, 1\}$, the field is $(Z_2, \oplus, \odot)$ with addition and multiplication carried out $\mathrm{mod}(2)$ as described in chapter 1. The ring $Z_2[x]$ is all polynomials having every coefficient either 0 or 1, with polynomial multiplication and addition carried out $\mathrm{mod}(2)$. There are two polynomials of degree 1 in $Z_2[x]$, namely x and $x + 1$. The polynomials of degree 2 are

$$P(x) = x^2 = xx,$$
$$Q(x) = x^2 + x = x(x + 1)$$
$$R(x) = x^2 + 1 = (x + 1)(x + 1)$$
$$S(x) = x^2 + x + 1.$$

The polynomials P, Q, R are clearly reducible. There is only one root of P, namely $r = 0$. The roots of Q are 0 and 1, the only root of R is $r = 1$. The polynomial S is irreducible and it has no roots because

$$S(0) = 1,\ S(1) = 1.$$

[2.4] The Division Algorithm in $F[x]$. Let $D \in F[x]$, $\deg(D) = n$, $d_n = 1$, so

$$D(x) = x^n + \sum_{j=0}^{n-1} d_j x^j.$$

Every polynomial P in $F[x]$ has a unique expression as

$$P = Q \cdot D + R, \quad 0 \le \deg(R) < n. \tag{2.1}$$

Proof. Let $\deg(P) = m$. If $m < n$, then the representation is

$$P = 0 \cdot D + P.$$

We will continue by induction on m for $m \ge n$. First, if $m = n$ then write

$$\begin{aligned} P(x) &= \sum_{j=0}^{n} p_j x^j = p_n x^n + \sum_{j=0}^{n-1} p_j x^j \\ &= p_n \left[x^n + \sum_{j=0}^{n-1} d_j x^j \right] - p_n \sum_{j=0}^{n-1} d_j x^j + \sum_{j=0}^{m-1} p_j x^j = p_n D(x) + R(x), \end{aligned} \tag{2.2}$$

Since $\deg(R) \le n-1$, this completes the proof for the case $m = n$.

For the inductive step assume that $[2.1]$ holds for polynomials of degree k for $n \le k < m$. Assume that $\deg(P) = m$. The equalities

$$\begin{aligned} P(x) &= \sum_{j=0}^{m} p_j x^j = p_m x^m + \sum_{j=0}^{m-1} p_j x^j \\ &= p_m x^{m-n} \left[x^n + \sum_{j=0}^{n-1} d_j x^j \right] - p_m x^{m-n} \sum_{j=0}^{n-1} d_j x^j + \sum_{j=0}^{m-1} p_j x^j \\ &= p_m x^{m-n} D(x) + R_{m-1}(x) \end{aligned} \tag{2.3}$$

determine R_{m-1}. Since $\deg(R_{m-1}) \le m-1$, by induction there are polynomials Q^* and R with $\deg(R) < n$ satisfying

$$R_{m-1}(x) = Q^*(x) \cdot D(x) + R(x).$$

Thus

$$P(x) = [p_m x^{m-n} + Q^*(x)]D(x) + R(x),$$

thus completing the induction with

$$Q(x) = p_m x^{m-n} + D^*(x).$$

To prove uniqueness, suppose that

$$\boxed{P = Q_1 \cdot D + R_1,\ 0 \le \deg(R_1) < n} \quad P = Q_2 \cdot D + R_2,\ 0 \le \deg(R_2) < n$$

are two representations. Subtracting the second from the first gives

$$0 = (Q_1 - Q_2) \cdot D + (R_1 - R_2).$$

since the $\deg(D) = n$ and the $\deg(R_1 - R_2) < n$, it must the case that $Q_1 - Q_2 = 0$. From $Q_1 = Q_2$ it follows that $R_1 = R_2$.

Notice that the steps $[2.2]$ and $[2.3]$ are the basis for the ”long division” algorithm.

[2.5] Illustrations in Z_2. Let

$$D(x) = x^2 + x + 1, \ \ P(x) = x^4.$$

The division algorithm gives

$$x^4 = [x^2 + x][x^2 + x + 1] + x,.$$

so the Q and R of the division algorithm are

$$Q(x) = x^2 + x, \quad R(x) = x.$$

Following the steps of the algorithm in the format common to most people, and keeping in mind that arithmetic is $\mathrm{mod}(2)$:

$$\begin{array}{r} x^2 + x \\ x^2 + x + 1 \overline{)\, x^4 + 0x^3 + 0x^2 + 0x + 0} \\ \underline{x^4 + x^3 + x^2} \\ x^3 + x^2 \\ \underline{x^3 + x^2 + x} \\ x \end{array}$$

One more example of the division algorithm in the ”usual” format:

$$\begin{array}{r} x^3 + 0x^2 + 0x + 1 \\ x^4 + x^3 + 0x^2 + 0x + 1 \overline{)\, x^7 + x^6 + 0x^5 + x^4 + 0x^3 + x^2 + x + 0} \\ \underline{x^7 + x^6 + 0x^5 + 0x^4 + x^3} \\ x^4 + x^3 + x^2 + x + 0 \\ \underline{x^4 + x^3 + 0x^2 + 0x + 1} \\ x^2 + x + 1 \end{array}$$

Thus the result of the division algorithm is

$$P(x) = Q(x)D(x) + R(x),$$

$$x^7 + x^6 + 0x^5 + x^4 + 0x^3 + x^2 + x + 0 = [x^3 + 0x^2 + 0x + 1][x^4 + x^3 + 0x^2 + 0x + 1] + [x^2 + x + 1]$$

[2.6] The Euclidean Algorithm for $F[x]$. This algorithm constructs a common divisor of two polynomials P and D, Assume that $\deg(D) \leq \deg(P)$ and use the division algorithm repeatedly to write

$$\begin{aligned} P &= QD + R_0, \quad \deg(R_0) < \deg(D); \\ D &= Q_1R_0 + R_1, \quad \deg(R_1) < \deg(R_0), \\ R_0 &= Q_2R_1 + R_2, \quad \deg(R_2) < \deg(R_1), \\ &\vdots \\ R_j &= Q_{j+2}R_{j+1} + R_{j+2}, \quad \deg(R_{j+2}) < \deg(R_{j+1}). \end{aligned} \qquad [2.4]$$

Since the degrees of the polynomials R_j are decreasing, there is j for which $\deg(R_{j+2}) = 0$; that is,

$$R_{j+2}(x) = a \in F.$$

There are two cases. If $a = 0$, then we have

$$R_j = Q_{j+2}R_{j+1}.$$

Backtracking, we can write

$$R_{j+1} = R_{j-1} - Q_{j+1}R_j.$$

Since R_{j+1} is a factor of R_j the last formula shows that it is also a factor of R_{j-1}. Repeating these steps we find that

$$R_{j+1} = S \cdot P + T \cdot D \qquad [2.5]$$

for polynomials S and T.

In the case that $a \neq 0$, the sequence of division algrorithm steps is

$$\begin{aligned} P &= Q_0 D + R_0, \quad \deg(R_0) < \deg(Q); \\ D &= Q_1 R_0 + R_1, \quad \deg(R_1) < \deg(R_0), \\ R_0 &= Q_2 R_1 + R_2, \quad \deg(R_1) < \deg(R_0), \\ &\vdots \\ R_j &= Q_{j+2} R_{j+1} + a.. \end{aligned}$$

By dividing all P and Q by a, we can assume that $a = 1$. Back substituting as before, there are polynomials S, T for which

$$1 = S \cdot P + T \cdot D. \qquad [2.6]$$

[2.7] Theorem. The polynomial R_{j+1} obtained by the Euclidean algorithm is a factor of both P and D. Any other polynomial which is a factor of both P and D is also a factor of R_{j+1}. Any other polynomial with these two properties is a constant multiple of R_{j+1}.

Proof. Since R_{j+1} is a factor of R_j, we see ”up the line” that R_{j+1} is a factor of P and D. If U is another polynomial which is a factor of both P and D, then by $[2.5]$ U is a factor of R_{j+1}. If R_{j+1} is also a factor of U, then that factor must be a non-zero constant.

[2.8] Definitions. By this theorem, the polynomial R_{j+1} is unique up to a nonzero constant multiple. It is called the *greatest common divisor*, or $\gcd$ of P and D. In the case of $[2.6]$

$$\gcd(P, Q) = 1$$

and P and D are said to be *relatively prime*. Irreducible polynomials are also called *prime polynomials*.

[2.9] Prime Factorization. Every element of $F[x]$ is a product of prime polynomials. The proof of the existence of a prime factorization is easy. If P is prime, then there is just one factor namely P. If Q is not prime, then it is a product of two other polynomials; etc. So each Q may be written as

$$Q = P_1^{\alpha_1} P_2^{\alpha_2} \cdots P_k^{\alpha_k} = \prod_{j=1}^{k} P_j^{\alpha_j} \qquad [2.7]$$

where the P_j are all irreducible and the exponents α_j are positive integers. The representation is unique. To see this, suppose that

$$\prod_{j=1}^{k} P_j^{\alpha_j} = \prod_{j=1}^{k} P_j^{\beta_j}$$

for a set of irreducible polynomials, in which we allow the exponents to be 0 or positive. By cancellation we can assume that

$$\prod_{j=1}^{J} P_j^{\sigma_j} = \prod_{j=J+1}^{H} P_j^{\sigma_j}$$

in which the P_j on the right and on the left are distinct and $\sigma_j \geq 0$. Isolating P_1,

$$P_1 W = \prod_{j=J+1}^{H} P_j^{\sigma_j}$$

for a polynomial W. Since P_1 and $P_j^{\sigma_j}$ are relatively prime for $j > J$, by the Euclidean algorithm there are polynomials S_j and T_j for which

$$1 = S_j P_1 + T_j P_j^{\sigma_j}.$$

Thus

$$T_j P_j^{\sigma_j} = (1 - S_j P_1),$$

$$\prod_{j=J+1}^{H} T_j P_j^{\sigma_j} = \prod_{j=J+1}^{H} (1 - S_j P_1),$$

$$P_1 W \prod_{j=J+1}^{H} T_j = 1 + P_1 U_1 + P_1^2 U_2 + \cdots + P_1^K U_K,$$

for polynomials W and U_j. Hence there is a polynomial Y satisfying

$$P_1 Y = 1$$

Hence P_1 and Y are both constants, and all $\sigma_j = 0$.

[2.10] Example of the Euclidean Algorithm in $R[x]$

Let

$$P(x) = x^4,\ D(x) = x^2 + x + 1.$$

Then the steps of the Euclidean algorithm are

$$x^4 = [x^2 - x][x^2 + x + 1] + x,$$

$$[x^2 + x + 1] = [x + 1]x + 1.$$

Thus it results that x^4 and $[x^2 + x + 1]$ are relatively prime; no surprise as $[x^2 + x + 1]$ is irreducible in $R[x]$. The equality $[2.6]$ is found by back substitution:

$$
\begin{aligned}
1 &= [x^2 + x + 1] - [x + 1]x \\
&= [x^2 + x + 1] + [x + 1][x^4 - [x^2 - x][x^2 + x + 1]] \\
&= [1 + (x + 1)(x^2 - x)][x^2 + x + 1] + [x + 1]x^4 \\
&= [x^3 - x + 1][x^2 + x + 1] + [x + 1]x^4.
\end{aligned}
$$

[2.11] Examples of the Euclidean algorithm in $Z_2[x]$.

Example 1: As for $R[x]$, let

$$P(x) = x^4,\ D(x) = x^2 + x + 1.$$

The computations in $R[x]$ are valid in $Z_2[x]$. Since $-1 = 1,$ the result may be written

$$1 = [x^3 + x + 1][x^2 + x + 1] + [x + 1]x^4.$$

Example 2: Let

$$P(x) = x^7 + x^6 + x^4 + x^2 + x,\ D(x) = x^4 + x^3 + 1,$$

as in [2.5]. The first step of the Euclidean algorithm was done in [2.5]. This and the other steps are

$$
\begin{aligned}
x^7 + x^6 + x^4 + x^2 + x &= [x^3 + 1][x^4 + x^3 + 1] + [x^2 + x + 1] \\
x^4 + x^3 + 1 &= [x^2 + 1][x^2 + x + 1] + x \\
x^2 + x + 1 &= [x + 1]x + 1.
\end{aligned}
$$

Thus P and Q are relatively prime. The formula $[2.6]$ is obtained by back substitution :

$$1 = [x^3 + x^2 + x][x^7 + x^6 + x^4 + x^2 + x] + [x^6 + x^5 + x^4 + x^3 + x^2 + 1][x^4 + x^3 + 1]$$

Example 3: Let

$$P(x) = x^3 + x^2 + x,\ D(x) = x^3 + 1.$$

The Euclidean algorithm is

$$
\begin{aligned}
x^3 + x^2 + x &= 1 \cdot [x^3 + 1] + [x^2 + x + 1] \\
x^3 + 1 &= [x + 1][x^2 + x + 1] + 0.
\end{aligned}
$$

Hence $\gcd(P, Q) = x^2 + x + 1$ and formula $[2.6]$ is

$$x^2 + x + 1 = [1][x^3 + x^2 + x] + [1][x^3 + 1].$$

[2.12] Residue Class Fields by Irreducible Polynomials. Returning to the general case, let F be any field. For each positive integer k define a subset of $F[x]$ by

$$F_k[x] = \{S \in F[x] : \deg(S) \leq k\}.$$

Each $F_k[x]$ is a vector space over $F,$ and

$$
\begin{aligned}
F_0[x] &= span\{1\} = F,\\
F_1[x] &= span\{1, x\},\\
F_2[x] &= span\{1, x, x^2\},\\
&\vdots\\
F_k[x] &= span\{1, x, x\cdots x^k\}.
\end{aligned}
$$

Clearly $\dim(F_k[x], F) = k+1$ and

$$F_k[x] \subset F_{k+1}[x],$$

the containment being proper. The vector spaces $F_k[x]$ are not rings because they are not closed under multiplication, except for $k = 0$.

Let D be an irreducible polynomial in $F[x]$. For any polynomial $P \in F[x]$ write

$$P(x) = Q(x) \cdot D(x) + R(x)$$

for unique Q and R for which $\deg(R) < \deg(D)$. This is the division algorithm. Let $n = \deg(D)$. In analogy with section [1.1], define a mapping

$$\psi : F[x] \rightarrow F_{n-1}[x]$$

by

$$\psi(P) = R.$$

The sets

$$\psi^{-1}(R) = \{P \in F[x] : \psi(P) = R\}$$

are disjoint for different $R \in F_{n-1}[x]$ and

$$F[x] = \cup_{R \in F_{n-1}[x]} \psi^{-1}(R).$$

If $P, P' \in \psi^{-1}(R)$ then we say that P and P' are congruent $\mod[D]$ and write

$$P \equiv P' \bmod(D).$$

The sets $\psi^{-1}(R)$ are called the *equivalence classes* of $F[x]$ modulo D. As for the case of numbers, we write them as

$$\langle R \rangle = \psi^{-1}(R).$$

The set of all equivalence classes is written

$$\frac{F[x]}{[D(x)]} = \{\langle R \rangle : R \in F_{n-1}[x]\}.$$

[2.13] Addition and Multiplication in $\frac{F[x]}{[D(x)]}$. The operations are defined using the mapping ψ, as follows.

$$
\begin{aligned}
\langle R \rangle \oplus \langle S \rangle &= \psi(R + S);\\
\langle R \rangle \odot \langle S \rangle &= \psi(R \cdot S).
\end{aligned}
$$

Much as in the "number case", it is proved that $\left(\frac{F[x]}{[D(x)]}, \oplus, \odot\right)$ is a ring with identity $\langle 1 \rangle$. The mapping ψ is a *homomorphism* of the ring $F[x]$ onto the ring $\frac{F[x]}{[D(x)]}$.

As defined, the ring $\frac{F[x]}{[D(x)]}$ is a set of equivalence classes. Each element of $F_{n-1}[x]$ defines a unique equivalence class, and conversely every equivalence class has a unique representative in $F_{n-1}[x]$. The notation is cleaner if $\oplus$ and $\odot$ are performed on elements of

$F_{n-1}[x]$ in place of equivalence classes, and we will usually do so.

[2.14] Theorem. If D is irreducible in $F[x]$, then $\left(\frac{F[x]}{[D(x)]}, \oplus, \odot\right)$ is a field.

Proof. Since $\left(\frac{F[x]}{[D(x)]}, \oplus, \odot\right)$ is a ring with identity, the only thing that needs to be shown is that there exists an inverse for every element of $\frac{F[x]}{[D(x)]}$. Let $R \in F_{n-1}[x]$, $R \neq 0$. Since D is irreducible, by the Euclidean Algorithm there are polynomials S and T for which

$$1 = S \cdot R + T \cdot D,$$

and so

$$S \cdot R = 1 - T \cdot D$$

Apply ψ to this equality:

$$\psi(S \cdot R) = \psi(1) - \psi(T \cdot D) = 1 - 0 = 1.$$

Thus

$$S \odot R = \psi(S \cdot R) = 1$$

and $R^{-1} = S$.

[2.15] An Example in $R[x] : \frac{R[x]}{[x^2+1]}$. The polynomial

$$D(x) = x^2 + 1$$

is irreducible in $R[x]$. Since $n = 2$,

$$\begin{aligned} F_{n-1}[x] &= F_1[x] = \text{Linear Polynomials and constants} \\ &= \{a + bx : a, b \in R\}. \end{aligned}$$

The equivalence class of the polynomial $R(x) = x$ is

$$\langle x \rangle = \{Q(x)[x^2 + 1] + x : Q \in F[x]\}.$$

Taking $Q = 1, x, x^2 \cdots x^k \cdots$, all the polynomials

$$x^2 + 1 + x,\ x^3 + x + x,\ \cdots, x^{k+2} + x^k + x, \cdots$$

are in $\langle x \rangle$.

The division algorithm for x^2 is

$$x^2 = 1 \cdot [x^2 + 1] + (-1).$$

Hence we have

$$\begin{aligned} (a + bx) \odot (c + dx) &= \psi((a + bx) \cdot (c + dx)) = \psi(ac + (bc + ad)x + bdx^2) \\ &= ac \oplus (bc \oplus ad)x \oplus bd(-1) = (ac \ominus bd) \oplus (bc \oplus ad)x, \end{aligned}$$

so multiplication in $\frac{R[x]}{[x^2+1]}$ using representatives in $F_1[x]$ is

$$(a + bx) \odot (c + dx) = (ac \ominus bd) \oplus (bc \oplus ad)x.$$

Multiplication in the complex numbers $(C, +, \cdot)$ is

$$(a + bi) \cdot (c + di) = (ac - bd) + (bc + ad)i.$$

Thus the mapping σ defined by

$$\sigma(a + ib) = a + bx,$$

is an isomorphism from the complex numbers onto $\frac{R[x]}{[x^2+1]}$. In terms of equivalence classes

$$\sigma(a + ib) = \langle a + bx \rangle.$$

We have seen that the complex numbers can be constructed by residue class fields. The field C, or $\frac{R[x]}{[x^2+1]}$, is a field extension of R in which the equation $x^2 + 1 = 0$ has solutions i and $-i$.

Before giving more examples, we make explicit the connection between roots and reducibility.

[2.16] Roots of Polynomials. Return to the general case of a field F and $P \in F[x]$. If $r \in F$ and $P(r) = 0$, then r is a *root of* P.

Theorem. The element $r \in F$ is a root of $P \in F[x]$ if and only if $(x - r)$ is a factor of P.

Proof. By the division algorithm, there is a polynomial $Q \in F[x]$ and $a \in F$ for which

$$P(x) = Q(x)(x - r) + a.$$

Since $P(r) = a$, we have $P(r) = 0$ if and only if $(x - r)$ is a factor of P.

[2.17] The Example $\frac{Z_2[x]}{[x^2+x+1]}$. The polynomial $D(x) = x^2 + x + 1$ is irreducible in $Z_2[x]$ because the only 2 elements available to be roots of D are $0, 1$ and $D(0) = 1$, $D(1) = 1$. The polynomials of degree 1 or 0 are

$$F_1[x] = \{0, 1, x, x + 1\}.$$

To find the multiplication table for $\frac{Z_2[x]}{[x^2+x+1]}$, we need the division algorithm for x^2, namely

$$x^2 = 1 \cdot [x^2 + x + 1] + [x + 1].$$

Thus $\psi(x^2) = x + 1$, so $x^2 = x \oplus 1$ in $\frac{Z_2[x]}{[x^2+x+1]}$. This gives the multiplication table for $\frac{Z_2[x]}{[x^2+x+1]}$. The addition table is easy, and we have:

$\odot$	0	1	x	$1 \oplus x$
0	0	0	0	0
1	0	1	x	$1 \oplus x$
x	0	x	$1 \oplus x$	1
$1 \oplus x$	0	$1 \oplus x$	1	x

,

$\oplus$	0	1	x	$1 \oplus x$
0	0	1	x	$1 \oplus x$
1	1	0	$1 \oplus x$	x
x	x	$1 \oplus x$	0	1
$1 \oplus x$	$1 \oplus x$	x	1	0

Summarizing, $\frac{Z_2[x]}{[x^2+x+1]}$ is a field of 4 elements having the above addition and multiplication tables. It is an extension field of Z_2 in which Z_2 has been augmented with the two elements $\{x, x + 1\}$. The multiplication and addition tables for Z_2 are the upper left 3×3 matrices..

We conclude this example by pointing out two isomorphisms. The tables for $\left[\frac{Z_2[x]}{[x^2+x+1]}\right]^* = K^*$ and $(Z_3,+)$ are

K^*	1	x	$1\oplus x$
1	1	x	$1\oplus x$
x	x	$1\oplus x$	1
$1\oplus x$	$1\oplus x$	1	x

$(Z_3,+)$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

.

The group $(K^*,\odot)$ is isomorphic to $(Z_3,+)$ under the mapping

$\sigma(1)=0$	$\sigma(x)=1$	$\sigma(1\oplus x)=2$

The addition tables for $K=\frac{Z_2[x]}{[x^2+x+1]}$ and $(Z_2\times Z_2,+)$ are

$\oplus$	0	1	x	$1\oplus x$
0	0	1	x	$1\oplus x$
1	1	0	$1\oplus x$	x
x	x	$1\oplus x$	0	1
$1\oplus x$	$1\oplus x$	x	1	0

$+$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,0)$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,1)$	$(0,1)$	$(0,0)$	$(1,1)$	$(1,0)$
$(1,0)$	$(1,0)$	$(1,1)$	$(0,0)$	$(0,1)$
$(1,1)$	$(1,1)$	$(1,0)$	$(0,1)$	$(0,0)$

and we see that these groups are isomorphic under the mapping τ defined by the table

$k\in K$	0	1	x	$1\oplus x$
$\tau(k)\in Z_2\times Z_2$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$

.

[2.18] Counting Polynomials in $Z_2[x]$ Having Linear Factors. This short section and the next address the problem of counting and constructing irreducible polynomials of a given degree. First, in a rather clumsy notation, the vector space $Z_2[x]$ has subspaces $(Z_2)_k[x]$ for each positive integer k. For example, the polynomials of degree 2 over Z_2 are

$$(Z_2)_2[x]=\{0,1,x,x+1,x^2,x^2+1,x^2+x,x^2+x+1\}.$$

For a polynomial of degree n, say

$$P(x)=x^n+\sum_{j=1}^{n-1}a_jx^j+a_0$$

there are n coefficients, each of which can be 0 or 1. Hence there are 2^n polynomials of degree n. The polymomial x is a factor if and only if $a_0=0$; this is half the polynomials of degree n, or 2^{n-1} polynomials. The number of polynomials have $x+1$ as factor is also 2^{n-1}, because the other factor is a polynomial of degree $n-1$. Finally the number of polynomials having both x and $(x+1)$ as a factor is simply the number of polynomials of degree $(n-2)$, namely 2^{n-2}. The number of polynomials having at least one linear factor is thus

$$\#\left[\text{degree } n, \text{ at least one linear factor}\right]=2^{n-1}+2^{n-1}-2^{n-2}=3\cdot 2^{n-2}.$$

and

$$\#\left[\text{degree } n \text{ with no linear factors}\right] = 2^n - 3 \cdot 2^{n-2} = 2^{n-2}.$$

[2.19] Low Order Irreducible Polynomials in $Z_2[x]$. The linear and quadratic irreducibles are

$$x,\ x+1,\ x^2+x+1.$$

The cubics with two or more prime factors can be written down by simply generating all combinations of the irreducible first and second degree polynomials whose product is cubic. They are

$$\begin{bmatrix} x^3 & (1+x)^3 = x^3+x^2+x+1 \\ x^2(1+x) = x^3+x^2 & x(x^2+x+1) = x^3+x^2+x \\ x(1+x)^2 = x^3+x & (x+1)(x^2+x+1) = x^3+1 \end{bmatrix}.$$

There are eight cubic polynomials, so there are two irreducible cubics, namely

$$x^3+x+1, \quad x^3+x^2+1.$$

The $4th$ degree polynomials with 2 or more prime factors are the first 13 in the following list and the irreducible ones are the last 3.

x^4	$(x^2+x+1)^2 = x^4+x^2+1$
$x^3(x+1) = x^4+x^3$	$x(x^3+x+1) = x^4+x^2+x$
$x^2(x+1)^2 = x^4+x^2$	$(x+1)(x^3+x+1) = x^4+x^3+x^2+1$
$x(x+1)^3 = x^4+x^3+x^2+x$	$x(x^3+x^2+1) = x^4+x^3+x$
$(x+1)^4 = x^4+1$	$(x+1)(x^3+x^2+1) = x^4+x^2+x+1$
$x^2(x^2+x+1) = x^4+x^3+x^2$	x^4+x+1
$(x+1)(x^2+x+1) = x^4+x^3+x+1$	x^4+x^3+1
$x(x+1)(x^4+x^3+x^2+x+1) = x^4+x$	$x^4+x^3+x^2+x+1$

Thus there are seven irreducible polynomials of degrees $1,\ 2,\ 3$, and 4.

The reducible polynomials of degree 5 may be counted as follows

All Linear Prime Factors	6
Three Linear and One Quadratic	4
Two Linear and One Cubic	6
One Linear, Two Quadratic	2
One Linear, One 4th Degree	6

so there are 24 reducible and 8 irreducible polynomials of degree 5.

It would be interesting to find a good recursion for the number of irreducible polynomials of degree n using this method method of generating all products of primes.

[2.20] Summary. In this chapter we have presented the basic theory of residue class fields in polynomial rings over fields and given examples which are relevant for the NeoCore Icon Generator. For the most part we are interested in finite fields. Residue class fields for finite fields are treated briefly in most books on modern algebra. Detailed treatments of finite fields are treated in the books by Niederrieter and Lidl [xx] and by McEliece [yy].

In chapter 3 we describe the Icon Generator in terms of residue class fields.

Chapter Three

The Neocore Icon Generator

[3.1] Definitions and Examples. The Icon generator transforms vectors with elements from $Z_2 = \{0,1\}$ into numbers in the range $[0, 2^n - 1]$ for a fixed positive integer n. The output number of the Icon generator will be called the "Icon". The domain vectors can have any length; they are "bit streams". To mathematically model the transform, it is convenient to first consider the domain to be the vector space

$$V = \{(v_j)_{j=0}^{\infty} : v_j \in \{0,1\} \text{ and } \exists\, h \in N \text{ for which } v_j = 0 \text{ if } j > h\},$$

defined and discussed in section [1.10]. For each $v \neq 0$ there is a unique integer h for which

$$v_h = 1 \text{ and } v_k = 0 \text{ if } k > h.$$

This vector space V is isomorphic to the vector space $Z_2[x]$ under the one-to-one mapping

$$v \leftrightarrow P(x) = x^h + \sum_{j=0}^{h-1} v_j x^j.$$

We take $Z_2[x]$ to be the domain of the Icon generator.

Elements of V, and of $Z_2[x]$, correspond to dyadic expansions of non-negative integers by the mapping

$$\delta(v) = \sum_{j=0}^{\infty} v_j 2^j. \qquad [3.1]$$

Of course, for any v the sum is actually finite, with largest non-zero term 2^h.

As an example, the element

$$v = (0,1,1,0,1,0,1,1,\overline{0},\cdots)$$

satisfies $\gamma(v) = 7$ and v corresponds to the polynomial

$$f(x) = 0 + x + x^2 + 0 \cdot x^3 + x^4 + 0 \cdot x^5 + x^6 + x^7$$

in $Z_2[x]$. The corresponding number is

$$\delta(v) = 2 + 2^2 + 2^4 + 2^6 + 2^7 = 2 + 4 + 16 + 64 + 128 = 214.$$

Note that for the vector v the index h is the largest index for which $v_h \neq 0$. That is, 2^h is the largest power of 2 in the dyadic expansion $\delta(v)$. This is standard mathematical notation, but it is reverse of common bitstring representations. As a bitstring of length 8, v would be written

$$v' = (1,1,0,1,0,1,1,0)$$

and the coordinate on the right, the $8th$, is the "least significant bit"; the $1st$ is the most significant. The bitstring notation corresponds to writing the polynomial as

$$f(x) = x^7 + x^6 + 0x^5 + x^4 + 0x^3 + x^2 + x + 0.$$

[3.2] Projection For An Irreducible Polynomial. In the definition of the Icon generator, the fixed integer n of the interval $[0, 2^n - 1]$ is the degree of a fixed irreducible polynomial

$$D(x) = x^n + \sum_{j=0}^{n-1} d_j x^j. \qquad [3.2]$$

The Icon generator is defined in terms of the projection map

$$\psi : Z_2[x] \to \frac{Z_2[x]}{[D(x)]}$$

discussed in section [2.12]. If the division algorithm for g and D is

$$g(x) = Q(x)D(x) + R(x), \quad \deg(R) < n$$

then the definition of ψ is

$$\psi(g) = R.$$

Alternatively,

$$\psi(g) = \langle R \rangle = \text{equivalence class determined by } R.$$

The first notation is cleaner and we will take the range of ψ to be the vector space $F_{n-1}[x]$ of all polynomials of degree less than or equal to $n - 1$. Addition of polynomials in $F_{n-1}[x]$ is by addition of coordinates in F; that is, it is addition in $F_{n-1}[x]$ as a vector space. We will use the symbol $+$ for this addition.The multiplication which makes $F_{n-1}[x]$ a field is defined in section [2.13]. To avoid any possible confusioin with polynomial multiplication in the ring $F[x]$, we will use the symbol $\odot$ for field multiplication - as in chapter 2.

The Icon generator is a modification of the composition map

$$\delta(\psi(g)).$$

Continuing the example in [3.1], suppose that $n = 4$ and the irreducible polynomial is

$$D(x) = x^4 + x^3 + 1,$$

discussed in sections [2.5] and [2.19]. The division algorithm for f and D, found in [2.5], is

$$x^7 + x^6 + 0x^5 + x^4 + 0x^3 + x^2 + x + 0 = [x^3 + 0x^2 + 0x + 1][x^4 + x^3 + 0x^2 + 0x + 1] + \left[\, x^2 + x + 1 \,\right].$$

Hence

$$\psi(f) = R(x) = x^2 + x + 1.$$

We have

$$\delta(\psi(f)) = \delta(R) = 4 + 2 + 1 = 7.$$

For another example, take the same D and

$$g(x) = x^7 + x^5.$$

The division algorithm gives

$$x^7 + x^5 = [x^3 + x^2][x^4 + x^3 + 1] + [x^3 + x^2]$$

so

$$\psi(g) = x^3 + x^2.$$

In terms of input-output for strings of $\{0,1\}$, strings of length 8 are mapped to strings of length 4. The two examples are

$$(1,1,0,1,0,1,1,0) \rightarrow (0,1,1,1)$$
$$(1,0,1,0,0,0,0,0) \rightarrow (1,1,0,0)$$

and as numbers

$$(1,1,0,1,0,1,1,0) \rightarrow 1+2+4 = 7$$
$$(1,0,1,0,0,0,0,0) \rightarrow 4+8 = 12.$$

[3.3] The Icon Generator with Preconditioning. Again, the field is Z_2 and D is a fixed irreducible polynomial in $Z_2[x]$, $\deg(D) = n$. Note, however, that most of the theory is valid for an arbitrary field and the corresponding ring of polynomials $F[x]$ and we will use this notation.

The Icon generator is essentially the projection ψ, or the composition $\delta \circ \psi$. To prevent the Icon generator from being an identity map for shorter input polynomials, the input polynomials are ”preconditioned” by another transformation. Preconditioning adds a polynomial to elements of $F[x]$. This polynomial is of the form

$$M(x) = P_c(x)x^m$$

for a fixed positive integer m and a fixed conditioning polynomial P_c. The preconditioning map on $F[x]$ to $F[x]$ is

$$f \rightarrow P_c x^m + f.$$

Mathematically, this is called an *affine map*. It is often called a linear map from the vector space $F[x]$ into itself, though it is not actually linear. It is analogous to a map of the form $t \rightarrow a + t$ from line R onto R.

The *preconditioned Icon generator* is the mapping

$$\Psi(f) = \psi(P_c x^m + f).$$

We have

$$\Psi(f) = \psi(P_c x^m + f) = \psi(P_c x^m) + \psi(f)$$
$$= \psi(P_c) \odot \psi(x^m) + \psi(f)$$

so

$$\Psi(f) = \psi(P_c) \odot \psi(x^m) + \psi(f). \qquad [3.3]$$

The map ψ is a ring homomorphism and Ψ is affine. Formula $[3.3]$ is general in that m is any non-negative integer and P_c is arbitrary. In fact, it is usually the case that $\deg(P_c) = n-1$, and so $P_c \in F_{n-1}[x]$. Since

$$\psi(R) = R \text{ if } R \in F_{n-1}[x]$$

the formula becomes

$$\Psi(f) = P_c \odot \psi(x^m) + \psi(f), \text{ if } P_c \in F_{n-1}[x]. \qquad [3.3a]$$

For $f \in F_{n-1}[x]$ we have

$$\Psi(f) = P_c \odot \psi(x^m) + f, \quad \text{if } P_c\ ,\ f \in F_{n-1}[x]. \qquad [3.3b]$$

Finally, if $m < n$ then $x^m \in F_{n-1}[x]$ and so in this case

$$\Psi(f) = P_c \odot x^m + f, \text{ if } m < n \text{ and } P_c\ ,\ f \in F_{n-1}[x]. \qquad [3.3c]$$

For fixed m and P_c, the element $P_c \odot \psi(x^m)$ in formula $[3.3b]$ is a constant, showing that Ψ is a one-to-one map of $F_{n-1}[x]$ onto itself. This result is important for reconstruction, and we state it as a theorem.

[3.4] Theorem. The Icon Generator Ψ gives perfect reconstruction when restricted to the space $F_{n-1}[x]$. That is, Ψ is one-to-one from $F_{n-1}[x]$ onto $F_{n-1}[x]$.

The proof precedes the statement.

[3.5] Example. For this section, $F = Z_2$. Let $n = 2$ and use the irreducible polynomial

$$D(x) = x^2 + x + 1.$$

The 4 elements of $F_1[x]$ are

$$F_1[x] = \{0, 1, x, x+1\}.$$

Take

$$P_c(x) = x + 1 \text{ and } m = 2.$$

First we need $\psi(x^2)$; to find it, write

$$x^2 = 1 \cdot [x^2 + x + 1] + x + 1.$$

Thus

$$\psi(x^2) = x + 1.$$

Next

$$P_c \odot \psi(x^2) = (x+1) \odot (x+1) = x$$

Thus the transform Ψ is

$$\Psi(f) = x + \psi(f), \quad \text{all } f;$$
$$\Psi(f) = x + f, \qquad f \in F_1[x].$$

We calculate $\Psi(R)$ for each $R \in F_1[x]$, using a table:

R	$\psi(R)$	$x + R$	$\Psi(R)$
0	0	x	x
1	1	$x+1$	$x+1$
x	x	0	0
$x+1$	$x+1$	1	1

We see explicitly that Ψ permutes the elements of $F_1[x]$. Next we calculate Ψ on all polynomials of degree 2. See example $[2.17]$. We have

R	$DivAlg$	$\psi(R)$	$\Psi(R) = x + \psi(R)$
x^2	$= [x^2 + x + 1] + [x + 1]$	$x + 1$	1
$x^2 + 1$	$= [x^2 + x + 1] + [x]$	x	0
$x^2 + x$	$= [x^2 + x + 1] + [1]$	1	$x + 1$
$x^2 + x + 1$	$= [x^2 + x + 1] + [0]$	0	x

To make a table for the cubics we need only $\psi(x^3)$. Then calculations of ψ can use the table of ψ for quadratic polynomials and the fact that ψ is a homomorphism. For x^3, we have

$$x^3 = x[x^2 + x + 1] + x^2 + x;$$
$$\psi(x^3) = x.$$

The result :

$f = x^3 + g$	$\psi(x^3) + \psi(g)$	$\psi(f)$	$\Psi(f) = x + \psi(f)$
$x^3 + 0$	$x + 0$	x	0
$x^3 + 1$	$x + 1$	$x + 1$	1
$x^3 + x$	$x + x$	0	x
$x^3 + x + 1$	$x + x + 1$	1	$x + 1$
$x^3 + x^2$	$x + x + 1$	1	$x + 1$
$x^3 + x^2 + 1$	$x + x$	0	x
$x^3 + x^2 + x$	$x + 1$	$x + 1$	1
$x^3 + x^2 + x + 1$	$x + 0$	x	0

.

We see that there are 2 elements in each $\psi^{-1}(R)$, $R \in F_1[x]$. This is true in general; that is, each equivalence class contains the same number of elements of $F_k[x]$.

[3.6] Icon Algebra and Concatenation. The fact that the Icon generator is based on the homomorphic projection map

$$\psi : Z_2[x] \rightarrow \frac{Z_2[x]}{[D(x)]}$$

implies that Icons preserve some of the characteristics of the input data and may therefore stand in place of the input data during various manipulations. A good example of this is concatenation.

In practice, typical inputs to the Icon generator are character strings of various lengths. For example, consider the character strings ”Albert” and ”Einstein”. Albert contains 48 bits in using an ASCII representation and Einstein contains 64 bits. Let f_a be the polynomial corresponding to Albert and f_e be the polynomial corresponding to Einstein. The most

significant bit need not be a 1, so

$$\deg(f_a) \leq 47, \quad \deg(f_e) \leq 63.$$

The transforms (Icons) of the two independent polynomials representing "Albert" and "Einstein" are given by $[3.3]$ for $m = 48$ and $m = 64$ respectively, and are :

$$\begin{aligned} \Psi_{Albert} &= \Psi(f_a) = \psi(P_c) \odot \psi(x^{48}) + \psi(f_a) \\ \Psi_{Einstein} &= \Psi(f_e) = \psi(P_c) \odot \psi(x^{64}) + \psi(f_e). \end{aligned} \qquad [3.4]$$

If we wish to concatenate "Albert" and "Einstein" to give "AlbertEinstein", (a bit string of length 112) the new input polynomial will be

$$f = f_a x^{64} + f_e,$$

in which f_a is shifted left. Using $m = 112$, the transform of "AlbertEinstein" is

$$\Psi_{AlbertEinstein} = \Psi(f_a x^{64} + f_e) = \psi(P_c) \odot \psi(x^{112}) + \psi(f_a) \odot \psi(x^{64}) + \psi(f_e). \qquad [3.5]$$

The same result may be obtained by Icon algebra from the Icons Ψ_{Albert} and $\Psi_{Einstein}$:

$$\Psi_{AlbertEinstein} = \Psi_{Albert} \odot \psi(x^{64}) + [\Psi_{Einstein} - \psi(P_c) \odot \psi(x^{64})]. \qquad [3.6]$$

Shifting an input bit stream to the "left" by m bits is equivalent to multiplying the corresponding polynomial by x^m. In the residue class field the same shift is accomplished by multiplying by the Icon $\psi(x^m)$, and adjusting for P_c as in $[3.6]$.

Since the Icon space is a finite field, the inverse $[\psi(x^m)]^{-1}$ of $\psi(x^m)$ exists, so the effects of shifting to the left by m bits can be undone by multiplying the Icon by $[\psi(x^m)]^{-1}$. Using this information, if we have the Icon for "AlbertEinstein" and the Icon for "Einstein" we can obtain the Icon for "Albert" via the Icon algebra.

$$\begin{aligned} \Psi_{Albert} = [\Psi_{AlbertEinstein} - \Psi_{Einstein} + \psi(P_c) \odot \psi(x^{64})] \odot [\psi(x^{64})]^{-1} \\ [\Psi_{AlbertEinstein} - \Psi_{Einstein}] \odot [\psi(x^{64})]^{-1} + \psi(P_c) \end{aligned}$$

Icons may be built in an iterative fashion by using Icon algebra. Using "Albert" as an example and starting with the preconditioning polynomial we have:

$$\begin{aligned} \Psi_{P_c} &= \psi(P_c) \\ \Psi_A &= \Psi_{P_c} \odot \psi(x^8) + \psi(A) \\ \Psi_{Al} &= \Psi_A \odot \psi(x^8) + \psi(l) \\ \Psi_{Alb} &= \Psi_{Al} \odot \psi(x^8) + \psi(b) \\ \Psi_{Albe} &= \Psi_{Alb} \odot \psi(x^8) + \psi(e) \\ \Psi_{Alber} &= \Psi_{Albe} \odot \psi(x^8) + \psi(r) \\ \Psi_{Albert} &= \Psi_{Alber} \odot \psi(x^8) + \psi(t) \end{aligned}$$

Once again, since the Icon space is a finite field, there exists both additive and multiplicative inverse for every $\psi(\cdot)$, so whatever is done to build an Icon, can be undone. Other useful manipulations may be performed in the Icon space based on the fact that ψ is

a homomorphism. The sliding window operation presented in the next section is yet another example. All these manipulations are referred to as Icon algebra.

[3.7] Sliding Window Operation. One very useful search technique uses a sliding window. Suppose we are searching though a long stream of data looking for the word: ”Albert”. The search process would involve iconizing the first six characters of the data stream and then checking to see if the icon matched the icon for ”Albert”: Ψ_{Albert}. After this check, we would move over one character in the data stream, create the icon for the next six characters and again check to see if the icon matched the icon for ”Albert”. This process would continue through the whole data stream, recording every place the word ”Albert” was found.

The brute force method for this search would be to recompute the whole icon every time we move over one character. The more efficient method is to use icon algebra to remove the oldest character and then add the next character on to the current transform. Suppose the data stream that we are searching is:

”There are many books about Albert Einstein in the library.”

The search process would start by iconizing the first six characters creating the icon: $\Psi_{There\_}$. The icon $\Psi_{There\_}$ would be checked against the icon Ψ_{Albert}, this is done very efficiently, and a record made if there is a match. We then move one character forward in the data stream and compute the icon for: Ψ_{here_a}. This may be accomplished using icon algebra via the manipulations:

$$\Psi_{here\_a} = (\Psi_{There\_} - \psi(T) \odot \psi(x^{48})) \odot \psi(x^8) + \psi(a)$$

This process may be continued through the entire data stream. In practice instead of searching for a single item ”Albert”, hundreds, thousands, or even millions of items may be searched for simultaneously.

[3.8] Extensibility of the Transform. Continue to assume that D is an irreducible polynomial of degree n and $\Psi = \Psi_m$ is the corresponding Icon generator with power m. Generally m will be the length of input strings, but the mathematical model is valid for any m. If $m < n$, then Ψ is a one-to-one mapping of $F_m[x]$ into $F_{n-1}[x]$. If $m = n - 1$, then Ψ is also onto. Using the ideas of concatenation in the previous section, we can extend the domain of uniqueness without increasing the degree of the irreducible polynomial D.

Suppose that $n = 64$, so Ψ is perfect (that is, $1:1$) on $F_{63}[x]$ onto $F_{63}[x]$. The mapping is extended to strings f of length 128 by writing such a string as concatenation of two 64 bit strings, written for the moment as

$$f = f_u \vee f_\ell.$$

As a polynomial of maximal degree 127, f may be written as

$$f(x) = f_u(x)x^{64} + f_\ell(x).$$

where f_ℓ and f_u are polynomials of maximal degree 63; that is, $f_\ell, f_u \in F_{63}[x]$. If we apply Ψ_{128} to f we obtain

$$\Psi_{128}(f) = \psi(P_c) \odot \psi(x^{128}) + \psi(f_u) \odot \psi(x^{64}) + \psi(f_\ell), \qquad [3.7]$$

as in section [3.6]. Of course Ψ is not $1:1$ on $F_{127}[x]$, but $[3.7]$ together with

$$\Psi_{64}(f_u) = \psi(P_c) \odot \psi(x^{64}) + \psi(f_u) \qquad [3.8]$$

create the transform pair

$$[\Psi_{64}(f_u), \Psi_{128}(f)]$$

uniquely determining f. We state this as a theorem.

[3.8] Theorem. Let $f \in F_{127}[x]$. With notation as above, the two polynomials

$$\begin{array}{|l|}\hline \Psi_{64}(f_u) = \psi(P_c) \odot \psi(x^{64}) + f_u \\ \hline \Psi_{128}(f) = \psi(P_c) \odot \psi(x^{128}) + f_u \odot \psi(x^{64}) + f_\ell \\ \hline \end{array} \qquad [3.9]$$

in $F_{63}[x]$ uniquely determine $\dot{f} \in F_{128}[x]$.

Proof: First, the ψ disappeared in $[3.7]$ because ψ is the identity on $F_{63}[x]$. Clearly the first polynomial determines f_u. The second then determines f_ℓ.

The method extends to longer strings. Strings of length 192 require 3 transforms, of 256 require 4 transforms etc. Of course we are now dealing with matrix transforms, and the simple algebra is lost.

Chapter Four

Probability and Combinatorics of the Icon Generator

[4.1] Introduction. The Icon generator is often used to code a long vector of $0s$ and $1s$ - a bitstring - to a vector of much shorter length. The method associates with the vector a polynomial f having as coefficients the given vector. After some initial preconditioning, f is placed in one of the classes of a *residue class field*. It is this class that represents f for classification, identification, and algebraic manipulation by other algorithms. It is important that no two polynomials are placed in the same class, or at least that the probability of two polynomials falling in the same class is extremely small.

In this section we count the number of elements in certain sets of polynomials and compute probabilities of various events. The most important being the probability of collisions; that is, that two or more polynomials fall in the same class. This probability is given in section [4.5].

We begin be reviewing the setting in which we work.

[4.2] The Setting. We develop most of the theory in this chapter for an arbitrary finite field F. The most important case for the ICON generator is the two element field $Z_2 = \{0, 1\}$, with multiplication $\odot$ and addition $\oplus$ carried out $\mathrm{mod}(2)$. We anticipate applications for fields containing $(Z_2, \oplus, \odot)$, and so keep the development general.

We will use the symbol $+$ for addition in F and the symbol $\bullet$, or simple juxtapositiion, for multiplication. These are not, of course, the usual multiplication and addition in the field of rational numbers. There is a unique prime number p and a unique positive integer e for which the number elements of F is

$$|F| = q = p^e.$$

See Theorem [1.7] of Chapter 1 for detail.

In the case of $(Z_2, \oplus, \odot)$ and all the fields which contain $(Z_2, \oplus, \odot)$, the prime is $p = 2$.

The polynomial ring $F[x]$ is all polynomials

$$f(x) = a_0 + a_1 x + \cdots + a_k x^k = \sum_{j=0}^{k} a_j x^j$$

in which the coefficients $\{a_j\}_{j=0}^{k}$ are in F. The polynomial f has degree k if $a_k \neq 0$. The ring $F[x]$ is a vector space over the field F and each subset

$$F_k[x] = \{f : \deg(f) \leq k\}$$

is vector subspace of $F[x]$. The theory of these spaces is developed in Chapter 2.

[4.3] Counting Polynomials. The number of polynomials in $F_k[x]$ is the the same as the number of vectors of length $k+1$, with coordinates being elements of F. There are q choices for each coordinate, and so there are q^{k+1} polynomials; that is

$$|F_k[x]| = q^{k+1} . \qquad [4.1]$$

The set of polynomials

$$B = \{1, x, x^2, \cdots, x^k\}$$

form a basis for $F_k[x]$ over F and so

$$\dim\{F_k[x] : F\} = k+1. \qquad [4.2]$$

In the case $F = Z_2$, $|Z_2[x]| = 2^{k+1}$.

[4.4] Counting in Residue Class Fields. Residue class fields are generated by monic irreducible polylnomials in $F[x]$, as described in sections [2.12] - [2.17]. Let D be such a polynomial, $\deg(D) = n$. By the division algorithm, each $P \in F[x]$ has a unique representation

$$P = QD + R, \quad \text{where } R \in F_{n-1}[x] \text{ and } Q \in F[x]. \qquad [4.3]$$

The *residue class* $\langle R \rangle$ of a fixed $R \in F_{n-1}[x]$ is all $P \in F[x]$ which can be written in the form $[4.3]$ for some Q. In other words, the class $\langle R \rangle$ is all polynomials which have the remainder R when divided by D.. The set of all residue classes is denoted $\frac{F[x]}{D}$. Addition $+$ and multiplication $\cdot$ are defined in $\frac{F[x]}{D}$ as in $[2.12]$ and $[2.13]$. With these operations $\frac{F[x]}{D}$ is a finite field; see [2.14]. There is clearly a one-to-one correspondence between $\frac{F[x]}{D}$ and $F_{n-1}[x]$, namely $\langle R \rangle \leftrightarrow R$. As in chapter 2, we carry out the operations in $F_{n-1}[x]$.

Clearly each class $\langle R \rangle$ is infinite. We count the number of elements in $F_k[x]$ which are in a given class $\langle R \rangle$. First, if $k \le n-1$, then

$$F_k[x] \subset F_{n-1}[x]$$

and so the only element of $F_k[x]$ which is in $\langle R \rangle$ is R itself; in set language,

$$\langle R \rangle \cap F_k[x] = \{R\} \text{ if } k \le n-1.$$

If $k > n-1$, then each element P in $\langle R \rangle \cap F_k[x]$ is obtained for a unique Q in formula $[4.3]$. Thus

$$\begin{aligned} \langle R \rangle \cap F_k[x] &= \{P : \deg(Q) \le k-n\} \\ &= \{P : Q \in F_{k-n}[x]\} \end{aligned}$$

if $k \ge n$. Thus

$$\left| \langle R \rangle \cap F_k[x] \right| = q^{k-n+1}. \qquad [4.4]$$

Formula $[4.4]$ also shows that each residue class has the same number of elements in $F_k[x]$, namely q^{k-n+1}.

[4.5] Example in $Z_2[x]$. We continue with Example [2.17]. The irreducible polynomial is

$$D(x) = x^2 + x + 1$$

and so $n = 2$ and

$$F_1[x] = \{0, 1, x, x+1\}.$$

There are four residue classes. Since $q = p = 2$,

$$|F_2[x]| = 2^3 = 8.$$

The sets $\langle R\rangle \cap F_2[x]$ each have $q^{k-n+1} = 2^{2-2+1} = 2$ elements. They are obtained by letting Q range through $F_0[x] = F$. The intersections are listed in *Table* 1.

$\langle 0\rangle \cap F_2[x]$	=	$\{0, x^2+x+1\}$,
$\langle 1\rangle \cap F_2[x]$	=	$\{1, x^2+x\}$,
$\langle x\rangle \cap F_2[x]$	=	$\{x, x^2+1\}$,
$\langle x+1\rangle \cap F_2[x]$	=	$\{x+1, x^2\}$.

Residue Classes of $x^2 + x + 1$ in $F_2[x]$

Table 1

For $k = 3$, each set $\langle R\rangle \cap F_3[x]$ has four elements, corresponding to Q ranging through $F_1[x]$. In the table above, two cubics are added to each set. The results are in table 2.

$\langle 0\rangle \cap F_3[x]$	=	$\{0, x^2+x+1, x^3+x^2+x, x^3+1\}$,
$\langle 1\rangle \cap F_3[x]$	=	$\{1, x^2+x, x^3+x^2+x+1, x^3\}$,
$\langle x\rangle \cap F_3[x]$	=	$\{x, x^2+1, x^3+x^2\}$,
$\langle x+1\rangle \cap F_3[x$	=	$\{x+1, x^2, x^3+x+1, x^3+x\}$

Residue Classes of $x^2 + x + 1$ in $F_3[x]$

Table 2

[4.6] Probability of Distinct Residue Classes. Let D be an irreducible polynomial, $\deg(D) = n$. Let S be a subset of $F[x]$. Let k be the largest possible degree of a polynomial in S. Suppose that the number $s = |S|$ satisfies

$$s \le \min\{q^{k-n+1}, q^n\}. \qquad [4.5]$$

Let $m = q^n$. The probablity that s elements of $F[x]$ lie in distinct residue classes of $F_{n-1}[x]$ is

$$P(s \text{ elements in distinct classes}) = P(m, s) = \frac{m!}{m^s (m-s)!}. \qquad [4.6]$$

First Proof. The possibilities of distributing S in the residue classes range from the

extreme of putting all of S in one class to the opposite extreme in which each class contains either no elements of S or 1 element of S. The condition that $s \le q^{k-n+1}$ means that s is no larger than the size of $\langle R\rangle \cap F_k[x]$ and so any one residue class could hold all of the elements of S. Thus the first extreme is possible. The condition that $s \le q^n$ means that it is possible for each element of S is in a distinct residue class. It follows by an easy induction on s that the total number of ways that S can be partitioned into the residue classes is m^s.

Next we count the number of partitions of S for which each residue class contains either 0 or 1 element of S. The classes which contain 1 element of S can be chosen in binomial coefficient $\binom{m}{s}$ ways. For any one of these choices the s elements can be placed in the classes in $s!$ ways. Hence the number ways that S is partitioned into s distinct classes is

$$s!\binom{m}{s} = \frac{m!}{(m-s)!}.$$

Dividing by the total number of ways gives $[4.6]$.

Second Proof. Think of the polynomials in S as ordered. We count the number of ways of putting the polynomials in distinct residue classes. The first polynomial can be put in any class and there are m ways to choose this class. The second polynomial can be put any class except the class holding the first polynomial, so there are $(m-1)$ ways to place the second polynomial; and so, $m(m-1)$ ways to place the first 2 polynomials. Continuing, there are $m(m-1)(m-2)$ ways to place the first 3 in distinct classes. By this argument there are

$$m(m-1)\cdots(m-s+1)$$

ways to place the s polynomials in distinct classes. This number is precisely $\frac{m!}{(m-s)!}$. Dividing by m^s gives $[4.6]$.

[4.7] Illustration of the Proof. Suppose that $q = 2, n = 2,$ and $k = 3$. Then $m = 2^2 = 4$. The size of each $\langle R\rangle \cap F_3[x]$ is $q^{k-n+1} = 2^{3-2+1} = 4$. We build partitions by incrementing the number of elements in a set S with s elements. If $s = 1,$ then the one element of S may be in any of the 4 residue classes. Letting $S = \{1\},$ the possible partitions of S are giving in table 3. The rows $1-4$ indicate the residue classes and the columns indicate the partition.

$C\backslash P$	P_1	P_2	P_3	P_4
1	1	∅	∅	∅
2	∅	1	∅	∅
3	∅	∅	1	∅
4	∅	∅	∅	1

Table 3 – All Partitions of $\{1\}$ into 4 Classes

Each of the 4 partitions of a set of 1 object yields 4 partitions of the set $S = \{1,2\},$ by simply placing the new element $\{2\}$ in one of the 4 partitions of $\{1\}$. This gives a total of 16 different partitions of $\{1,2\}$ into residue classes. In table 4 this process is illustrated by following each partition of $\{1\}$ with its 4 childern.

$\frac{C}{P}$	P_1	P_{11}	P_{11}	P_{13}	P_{14}	P_2	P_{21}	P_{22}	P_{23}	P_{24}	P_3	P_{31}	P_{32}	P_{33}	P_{34}	P_4	P_{41}	P_{42}	P_{43}	P_{44}
1	1	1,2	1	1	1	∅	2	∅	∅	∅	∅	2	∅	∅	∅	∅	2	∅	∅	∅
2	∅	∅	2	∅	∅	1	1	1,2	1	1	∅	∅	2	∅	∅	∅	∅	2	∅	∅
3	∅	∅	∅	2	∅	∅	∅	∅	2	∅	1	1	1	1,2	∅	∅	∅	∅	2	∅
4	∅	∅	∅	∅	2	∅	∅	∅	∅	2	∅	∅	∅	∅	2	1	∅	∅	∅	1.2

Table 4 – *All Partitions of* $\{1,2\}$ *into* 4 *Classes*

There are 16 partitions of the set $\{1,2\}$ and of these

$$\frac{m!}{(m-s)!} = \frac{4!}{(2)!} = 12$$

of the partitioins have at most 1 element in each class. The probability of distinct classes is thus

$$P(4,2) = \frac{12}{16} = \frac{3}{4}.$$

If we add 3 to our set S, so $S = \{1,2,3\}$, then we obtain $4^3 = 64$ ways to partition the set. Of these,

$$\frac{m!}{(m-s)!} = \frac{4!}{(1)!} = 24$$

of the partitions place all elements in distinct classes. Thus the probability of distinct classes is

$$P(4,3) = \frac{24}{64} = \frac{3}{8}.$$

[4.8] Computation of Probabilities In the previous section we illustrated formula $[4.6]$ for small numbers. The formula for $P(m,s)$ may be written as

$$P(s,m) = \prod_{j=0}^{s-1} \frac{m-j}{m}.$$

It can also be computed recursively as

$$P(0) = 1;$$

$$\text{For } 1 \le j \le s-1, \qquad [4.7]$$

$$P(j) = \frac{m-j}{m} P(j-1).$$

In *table* 5 we calculate $P(m,s)$ for some moderate values of m and s. The numbers are rounded to 3 decimal places.

$m\backslash s$	3	5	10	15	20	30	40	50	60	70	80	90
32	.908	.720	.208	.020	0							
64	.954	.852	.477	.168	.036	0						
128	.977	.924	.698	.426	.209	.025						
256	.988	.961	.837	.658	.467	.171						
512	.994	.981	.915	.813	.687	.420	.209	.084	.027	.007	.001	0
1024	.997	.990	.957	.902	.830	.651	.462	.296	.172	.089	.004	.002
2048	.999	.995	.978	.950	.911	.808	.682	.547	.418	.303	.209	.137
4096	.999	.998	.989	.975	.955	.899	.826	.741	.648	.553	.460	.373

Probabilities $P(m,s)$ of s Objects in s Distinct Classes of m Classes $(s < m)$.

Table 5

In application n is as large as 2^{32} or 2^{64}. We make a short table of $P(m,s)$ for some of these values. Again we have rounded to 3 decimals, except in the case of $s = 1000,$ in which case we round to 5 decimals.

$m\backslash s$	10	30	50	100	150	200	250	300	500	700	1000
2^{16}	.999	.993	.981	.927	.843	.738	.622	.504	.148	.024	.00047
2^{32}	1	1	1	1	1	1	1	1	1	1	.99988

Probabilities $P(m,s)$ of s Objects in s Distinct Classes of m Classes $(s < m)$.

Table 6

[4.9] Probability of Collision with a Specified Set (NeoStore). In this section we take a different approach to deriving the probability of collisions within a specified set, denoted the NoeStore. This development takes into acccount the dimension of the vector space from which the input bitstrings are drawn. We make the same basic assumptions as in section [4.5], but with $F = Z_2$. . The vector space $F_k[x]$ is the space of possible data bitstrings of length $k+1$. The irreducible monic polynomial D with $\deg(D) = n$ generates the residue class field $F_{n-1}[x]$ and the corresponding residue classes

$$\{\langle R\rangle : R \in F_{n-1}[x]\}.$$

For this development, a NeoStore will be a random collection of items. The items are chosen from $F_k[x],$ iconized (transformed into their residue class in the residue class field $F_{n-1}[x]$), and then added to the NeoStore. A NeoStore will but built up to contain s items. Ideally there will be no collisions of these s items within the store. A collision occurs if two or more items drawn from the vector space $F_k[x]$ reside in the same residue class. First we consider the probability of randomly choosing an item from $F_k[x]$ that collides with an item in withi;n the NeoStore. In the next section we will determine the probability of collisions

withon a NeoStore containing s items.

Suppose that $S \subset F_k[x]$, $s = |S|$, and that S, k and n satisfy condition [4.5]. Suppose also that all the residue classes of elements of S, that is the elements of

$$\Psi(S) = \{\Psi(f) : f \in S\}, \qquad [4.8]$$

are distinct. We calculate the probability that a random element of $F_k[x] \setminus S$ is in one of the residue classes $\langle\Psi(f)\rangle$. That is, we compute the probability

$$P\Big\{g \in \langle\Psi(f)\rangle \;\Big|\; f \in S,\ g \notin S\Big\}.$$

Each class in [4.8] contains 2^{k-n+1} items, by formula [4.4]. Each $\langle\Psi(f)\rangle$ contains 1 element of S, namely f. Hence the number of possibilities for g is

$$(2^{k-n+1} - 1) \cdot s.$$

The total number of elements in $F_k[x] \setminus S$ is

$$2^k - s$$

and so

$$P\Big\{g \in \langle\Psi(f)\rangle \;\Big|\; f \in S,\ g \notin S\Big\} = \frac{(2^{k-n+1} - 1) \cdot s.}{2^{k+1} - s}. \qquad [4.9]$$

If we eliminate the condition "$g \notin S$" then we obtain the probability that a random element of $F_k[x]$ is in one of s distinct residue classes; namely,

$$P\Big\{g \in \Psi(f) \;\Big|\; f \in S,\ g \in F_k[x]\Big\} = \frac{(2^{k-n+1} - 1) \cdot s}{2^{k+1}}. \qquad [4.10]$$

In cases where s is much smaller than 2^{k+1}, (which is true for all practial applications) [4.10] is a good approximation of [4.9]. We can also eliminate the term "-1" and obtain

$$P\Big\{g \in \Psi(f) \;\Big|\; f \in S,\ g \notin S\Big\} \sim \frac{s}{2^n}. \qquad [4.11]$$

To reiterate in words, [4.9] gives the probability that a random g in $F_k[x] \setminus S$ is in one of s distinct residue classes. Formula [4.10] gives the probability that a random g in $F_k[x]$ is in one of s distinct residue classes, with no reference to a particular set S. Equation [4.11] is precisely what our intuition would tell us concerning the probability of a random g colliding with a NeoStore containing s items.

[4.10] Probability of Collisions within a NeoStore. If $k < n$, then

$$\Psi : F_k[x] \longrightarrow F_{n-1}[x]$$

is one-to-one so there is no possibility of collisions within the NeoStore, that is, thel probability of collisions is zero. The ensuing development assumes $k \geq n$. The number of residue classes in $F_{n-1}[x]$ is 2^n, so if $s > 2^n$ then the probability of collisions is one. Therefore we assume that $s \leq 2^n$. We will use a recursive approach to determine the probability of collision within a NeoStore.

Assumptions are again as in section [4.9]. In section [4.5], formula [4.6], we calculated the probability that there are no collisions. We now approach the same problem in a different way and obtain a recursion. We use formula [4.9] to calculate the probability that 2 or more items of S are in the same residue class. We will let the number of elements of S vary, so we will let $S = S_s$. The collision probability is

$$P\{\Psi(f) = \Psi(g), \text{ some } f,g \in S\} = 1 - P\left\{\Psi(f) \neq \Psi(g) : f,g \in S,\ f \neq g\right\}.$$

The probability on the right was calculated in section [4.5], but for this derivation we do not use the formula obtained there. Again let

$$P(2^n,s) = P\left\{\Psi(f) \neq \Psi(g) : f,g \in S_s,\ f \neq g\right\}$$

If

$$\Psi(f) \neq \Psi(g) \text{ for } f,g \in S_s,\ f \neq g$$

then certainly

$$\Psi(f) \neq \Psi(g) \text{ for } f,g \in S_{s-1},\ f \neq g$$

and so we have

$$P(2^n,s) = \left[1 - P\left\{g \in \Psi(f) \;\middle|\; f \in S_{s-1},\ g \notin S_{s-1}\right\}\right]P(2^n,s-1). \qquad [4.12]$$

In words, the term in bracketts is the probability that an item added to S_{s-1} to obtain S_s is not in $\Psi(f)$ for any $f \in S_{s-1}$. The formula holds because the events corresponding to the two factors are independent. Using formula [4.9], the recursion [4.12] becomes

$$P(2^n,s) = \left[1 - \frac{(2^{k-n+1}-1)\cdot(s-1)}{2^{k+1}-s+1}\right]P(2^n,s-1). \qquad [4.13]$$

Formula [4.6] for $P(2^n,s)$ is

$$P(2^n,s) = \frac{[2^n]!}{2^{ns}\cdot(2^n-s)!},$$

under the restriction

$$s \leq \min\{2^{k-n+1}, 2^n\}.$$

This restriction is needed for both proofs of formula [4.6], because the proofs require that s elements will fit into any equivalence class. The recursion [4.13] has no such restriction, and indeed k appears in the recursion. It is easily shown that [4.13] is equivalent to [4.7] under the conditions: $2^{k+1} >> (s-1)$, $j = s$, and $m = 2^n$.

[4.11] Exponential Approximation to P. The recursion [4.12] may be written in the form

$$P(2^n,s) - P(2^n,s-1) = -\frac{(2^{k-n+1}-1)\cdot(s-1)}{2^{k+1}-(s-1)}P(2^n,s-1). \qquad [4.14]$$

The function of the postive integer s given by

$$\mu(s) = \frac{(2^{k-n+1}-1)\cdot s}{2^{k+1}-s}$$

for fixed k and n extends to real numbers s as a function of s which has derivatives of all orders on the interval $(-\infty, 2^{k+1})$. The recursion [4.13] thus extends to a functional equation in the real variable s; namely

$$P(2^n,s) - P(2^n,s-1) = -\mu(s-1)P(2^n,s-1).$$

For fixed s the function of the right is the difference in values of P at the end points of the interval $[s-1,s]$. By the mean value theorem there is a $t = t_s \in [s-1,s]$ for which the left

side is $P'(2^n, t)$, giving the equation

$$P'(2^n, t) = -\mu(s-1)P(2^n, s-1).$$

The function μ varies slowly as a function s. We will give a more precise analysis below. This analysis justifies approximating $\mu(s-1)$ and $P(2^n, s-1)$ by $\mu(t)$ and $P(2^n, t)$. Thus an approximate model of the recursion [4.13] is the differential equation

$$P'(t) = -\mu(t)P(t), \qquad [4.14]$$

in which we have dropped the dependence on n. This equation has the exact solution

$$P(t) = A\exp\left(-\int_0^t \mu(x)dx\right) \qquad [4.15]$$

for a constant A. Since $P(0)$ is the probability that 0 items do not collide, we have $P(0) = 1$ and so $A = 1$.

For $s \ll 2^k$, which is true for all practial applications, the function μ may in turn be approximated by the function

$$\nu(s) = \frac{(2^{k-n+1} - 1) \cdot s}{2^{k+1}}.$$

The corresponding solution of the differential equation is

$$\begin{aligned} P(t) &= \exp\left(-\int_0^t \nu(x)dx\right) \\ &= \exp\left[-\frac{(2^{k-n+1} - 1)}{2^{k+2}}t^2\right]. \end{aligned}$$

This gives

$$P(\textit{ one or more collisions}) = 1 - P(2^n, s) \simeq 1 - \exp\left(-\frac{(2^{k-n+1} - 1) \cdot s^2}{2^{k+2}}\right). \qquad [4.16]$$

[4.12] Probability Graphs A few examples will help clarify things.

Example 1: The first example demonstrates the validity of the approximation [4.16] even under the extreme case when n is small. Let $n = 8$ and $k = 15$. Figure 4.1 is a graph of the probability of collisions occurring within a NeoStore containing various numbers of items in the store. The stair step curve was generated from the exact solution using the recursion [4.13]. The ”x” were generated using the recursion [4.7]. The smooth curve was generated from the expediential approximation [4.16]. It is clear from the graph that [4.16] is a good approximation even under the condition when n is relatively small. The approximation become better as n become larger.

Figure 4.2 is the same conditions as Figure 4.1, only $k = 8$. With $k = 8$, the probability of collisions within the NeoStore is lower because we have a smaller input space. Recall that if $k = n - 1$ then the probabiliy of collisions will be zero. With $k = n$ the probability of collisions is greater than zero but not as bad as when $k >> n$. The recursion [4.7] does not take into account the size of the input space, that is why there is such a large difference between the probability generated by this equation and [4.13]/[4.16]. In most all practical applications where k is greater than $n - 1$, k will be greater than $n - 1$ by at least 8, i.e. $k \geq n + 7$.

.

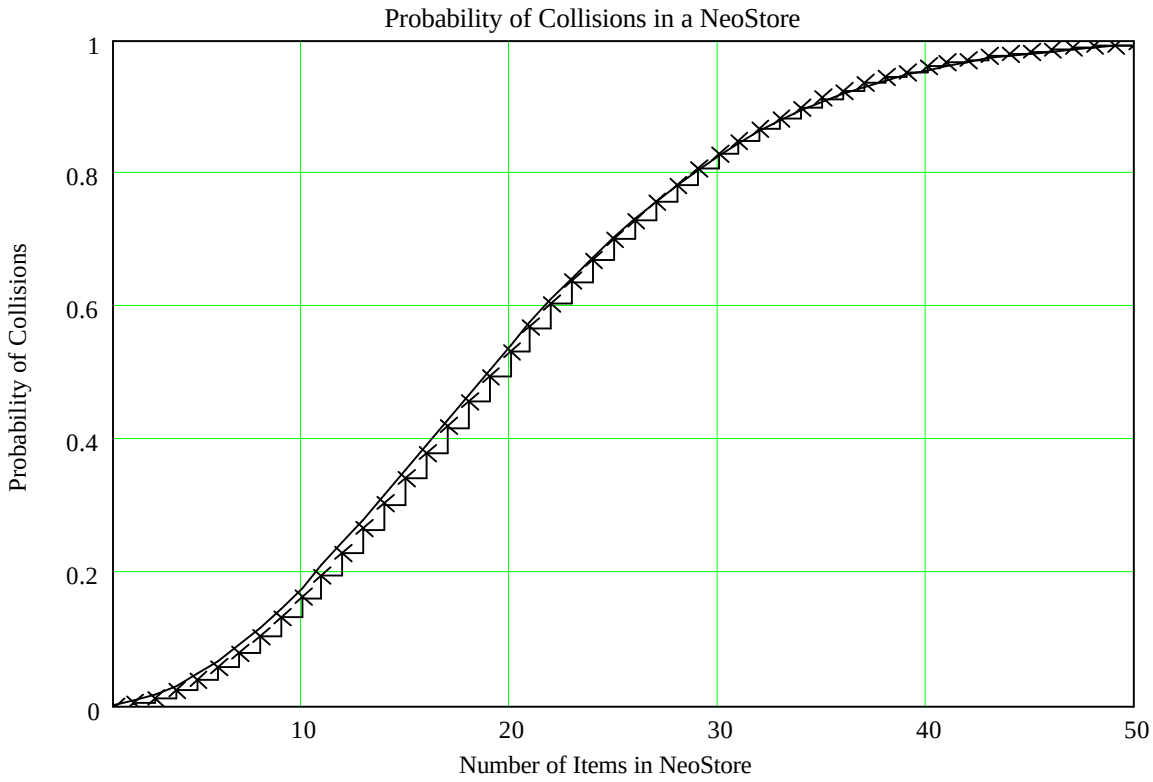
Probability of Collisions in a NeoStore
Probability of Collisions
1
0.8
0.6
0.4
0.2
0
10
20
30
40
50
Number of Items in NeoStore

Figure 4.1

.

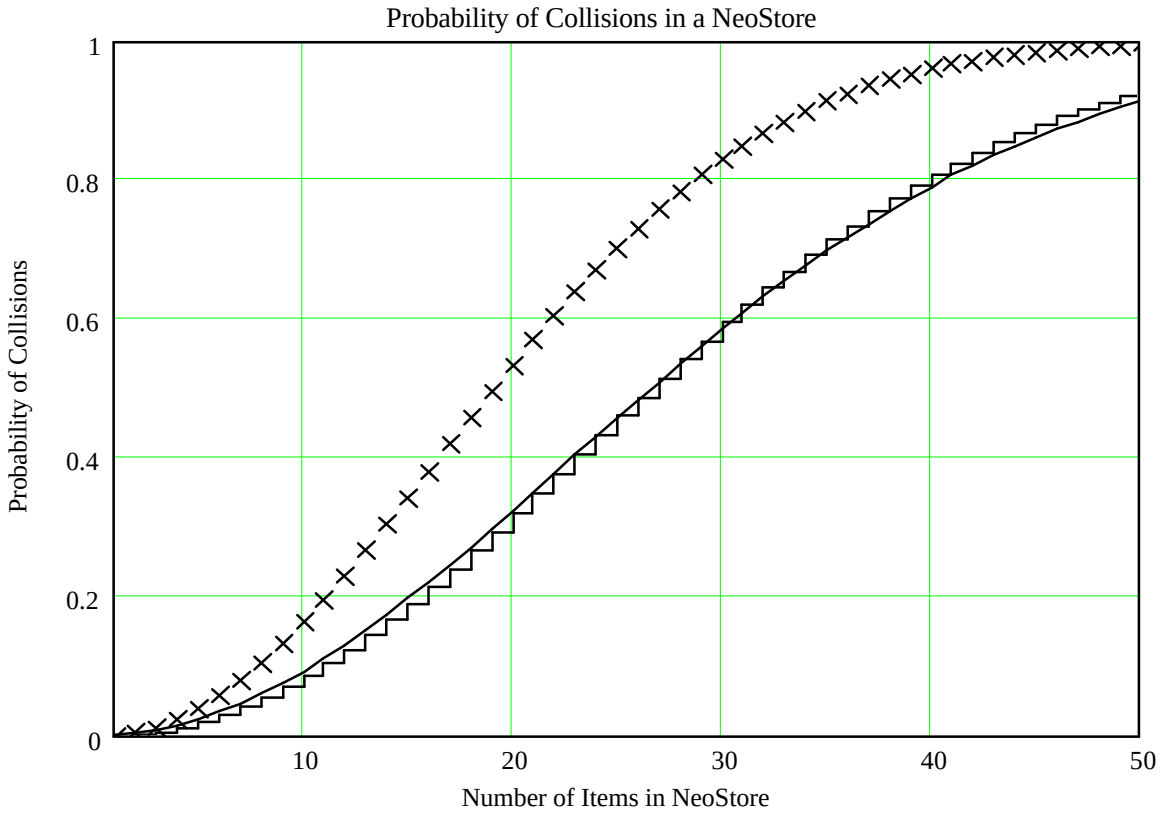

Figure 4.2

Example 2: A typical application uses $n = 64$. Figures 4.3 and 4.4 show the probability of collisions within a NeoStore for this case. We see from Figure 4.3 that the probability of collisions within a NeoStore reaches approximately 50% when the number of items in the store equals $1.18\sqrt{2^n}$ or five billion unique items. This is seen by solving [4.16] under the condition that $k >> n$. The graph shows that even with $10,000,000$ items in the store, there is only 1 chance $370,000$ of having collisions within the store.

. .

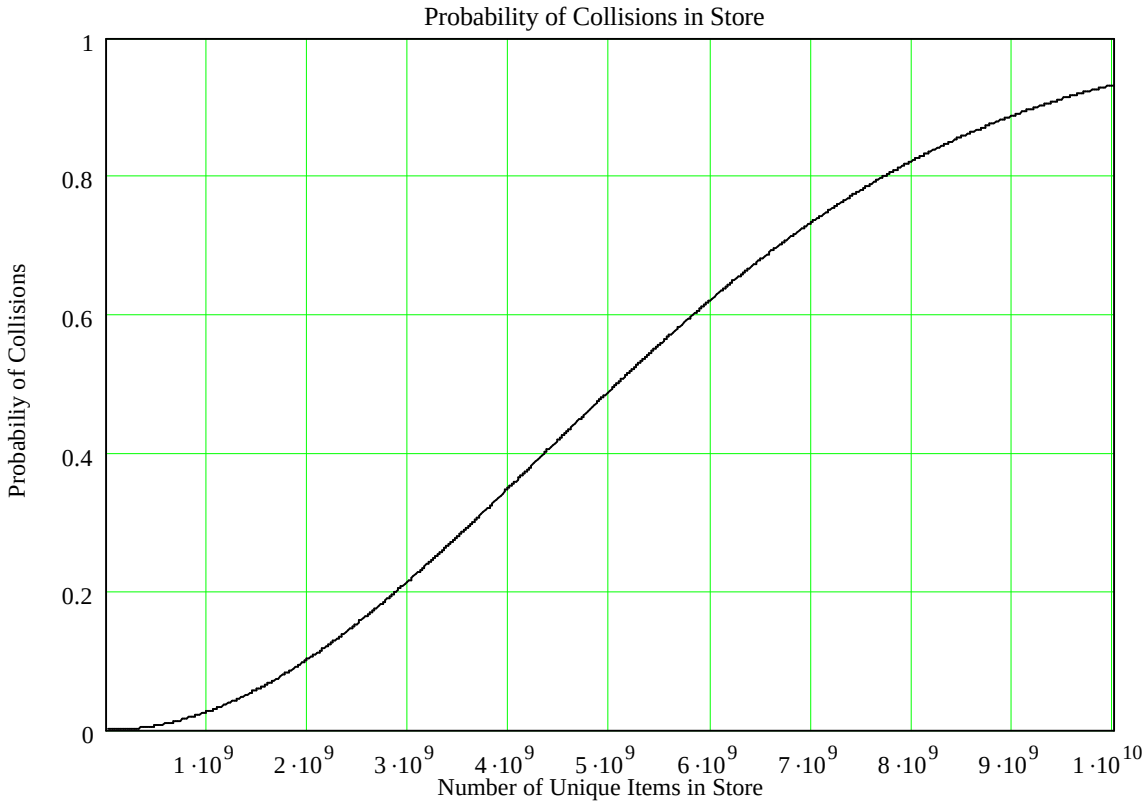
Probability of Collisions in Store
Probabiliy of Collisions
1
0.8
0.6
0.4
0.2
0
1 ·10^9
2 ·10^9
3 ·10^9
4 ·10^9
5 ·10^9
6 ·10^9
7 ·10^9
8 ·10^9
9 ·10^9
1 ·10^10
Number of Unique Items in Store

Figure 4.3

. .

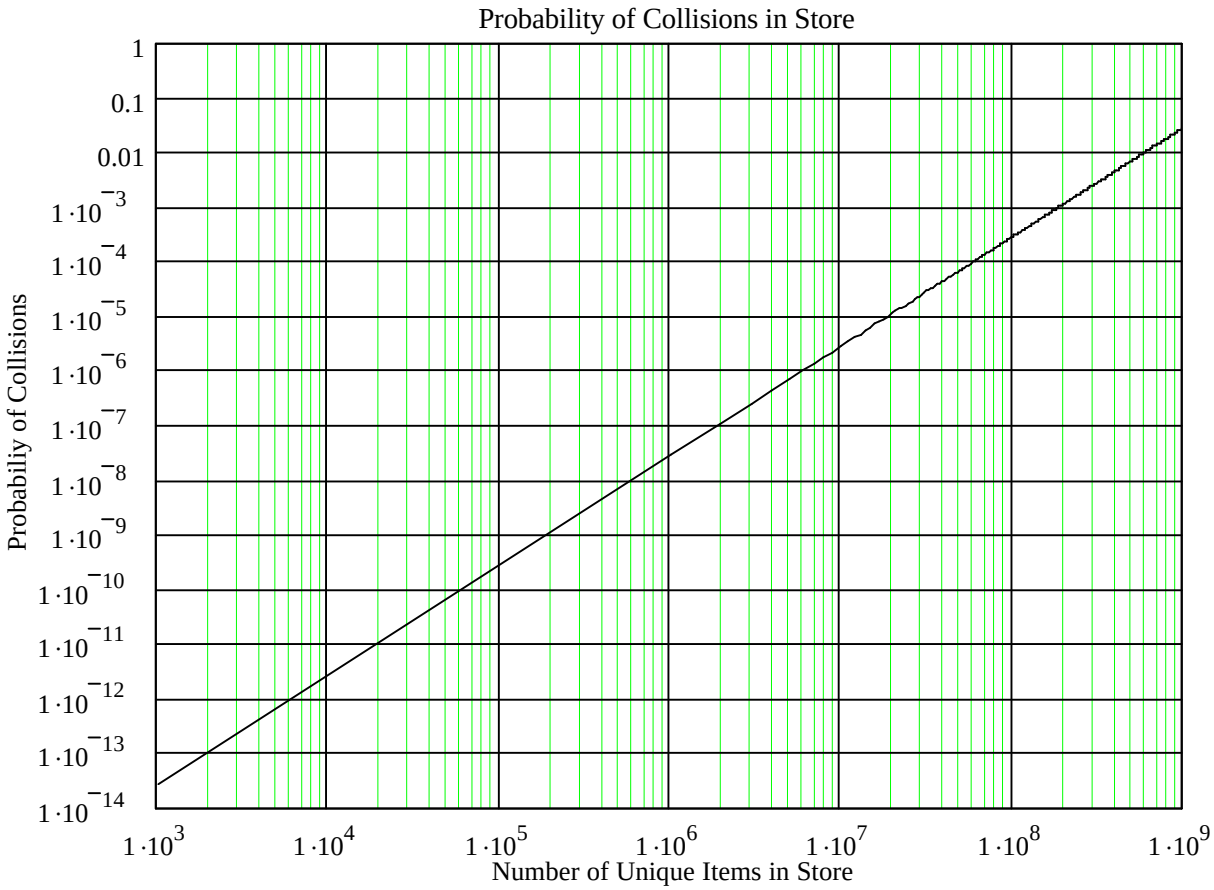
Probability of Collisions in Store
Probability of Collisions
1
0.1
0.01
1·10^-3
1·10^-4
1·10^-5
1·10^-6
1·10^-7
1·10^-8
1·10^-9
1·10^-10
1·10^-11
1·10^-12
1·10^-13
1·10^-14
1·10^3
1·10^4
1·10^5
1·10^6
1·10^7
1·10^8
1·10^9
Number of Unique Items in Store

Figure 4.4

Chapter Five

Virtual Content Addressable Memory Efficiency

[5.1] Introduction: Virtual Content Addressable Memory (vCAM) may be created with the use of Icons and an Associative Memory Controller (AMC). Other NeoCore technical papers describe the operational details of vCAMs and the AMC. This chapter takes up some of the performance issues of vCAMs and NeoStores. A NeoStore is essentially a vCAM.

[5.2] NeoStore and Associative Memory Controller: A NeoStore is a block of memory which contains quanta as a primary elements. A quanta contains various sub-elements, the details of which are not germane to this development. A NeoStore is typically established with a fixed number of quanta, which is the size of the NeoStore. We will let N denote the size of the NeoStore. For computer implementation reasons, N is typically a power of two. When an item is added to the NeoStore, the information or association of the item is placed in a quanta, and that quanta is considered allocated to that item. We will let K denote the number of items in the NeoStore, or equivalently, the number of allocated quanta. Clearly K must be less than or equal to N. If $K = N$ then we say that the NeoStore is full.

Associated with each item placed in the NeoStore is an Icon. The lower n bits of the Icon are used as a primary address into the NeoStore. If $N = 2^n$ is the size of the NeoStore, n is the number of bits from the Icon used as the primary address. One of the key properties of the Iconization transform is that the transformed randomly distributes the items being transformed across the transform space. This means that if we transform (Iconize) K different items, the K associated icons viewed as numbers in the range $[0, 2^{64}]$ (assuming a 64 bit Icon), will be evenly distributed across that range. This characteristic of the transform is clearly developed in the literature, see references [2] and [3].

As items are added to the NeoStore, we run into the distinct possibility that multiple items will have the same primary address. It is the job of the Associative Memory Controller to resolve these collisions on the address space. These collisions are resolvable and not related to the Icon collisions addressed in chapter 4. The AMC builds all items that transform into the same primary address into a duplicate chain structure. The duplicate chain structure uses quanta from the NeoStore in such a fashion that the NeoStore can be completely filled without destroying any unique primary addresses. Because of this unique feature of the AMC, the NeoStore can be viewed as a flexible structure. That is, when the NeoStore is full it contains the same number of bins as unique primary address, and each bin will hold exactly the number items that resolve to the same primary address. For example, if the size of the NeoStore were 1024 and if we placed 1024 items in the store, and if for some strange coincidence the lower 10 bits of all the icons were exactly the same, then the NeoStore would be viewed as containing 1 bin with 1024 items in it. If on the other hand the lower 10 bits for each icon were unique, then the NeoStore would be viewed as containing 1024 bins with 1 item in each bin. All other combinations in between these

extremes can also be handled.

The efficiency of finding an item/association in the NeoStore is measured in terms of the number of memory cycles it takes to locate that item. A quanta is typically of such a size that it can be read in one cache line read on a modern computer. Because of the random distribution of the Icons used to address a quanta in the NeoStore, it is assumed that a quanta being read will not be in cache memory, so a memory cycle will be required for each quanta accessed. This memory access is typically the gating item when finding an association in the NeoStore. If an item being searched for in the NeoStore is in a bin by itself, or if it is the first item in a duplicate chain, then only one memory cycle is required to find and access this item. If the item is the second on a duplicate chain, then two memory cycles will be required to find and access the item. If the item is the k^{th} item on a duplicate chain, then it will take k memory cycles to find and access the item.

The remainder of this chapter looks at the maximum expected duplicate chain lengths based on the size of the NeoStore and how full the NeoStore is, along with the average number of memory cycles required to find and access an item in the NeoCore vCAM.

[5.3] Maximum Expected Duplicate Chain Length: As noted above, a NeoStore contains a maximum of N bins in which to place an item/association. The probability of an item being placed in a given bin in the NeoStore is then:

$$p = \frac{1}{N} \qquad [5.1]$$

We will assume that K items are placed in the NeoStore where $K \leq N$. The iconization process evenly distributes the K items into the N potential bins in the NeoStore. For any given bin in the NeoStore, the probability of having n items in that bin is a Binomial distribution:

$$P_B(n) = \binom{K}{n} p^n (1-p)^{K-n} \qquad [5.2]$$

where

$$\binom{K}{n} = \frac{K!}{(K-n)!n!}$$

Typical store sizes are large, and we are interested in what happens when the NeoStore is full or close to full. This implies that p is small and that N and K are large. Under these conditions, the Binomial distribution is well approximated by the Posson distribution:

$$P_P(n) = \frac{\alpha^n}{n!} e^{-\alpha} \qquad [5.3]$$

where $\alpha = pK$. References [4] and [5] provide proofs that [5.3] approximates [5.2] under the given condition. The Posson distribution is more connivent for calculation purposes. I will use $P(n)$ in the ensuing development to represent the probability distribution of having n items in a given duplicate chain.

When the NeoStore is full, $K = N$ so $\alpha = pK = \frac{1}{N}N = 1$. Therefore the probability distribution characteristics of the number of items in a given bin (the length of a duplicate chain) is independent of the size of the NeoStore. Figure 1 shows the probability distribution under these conditions. This figure shows that approximately 37% of the bins in the NeoStore will contain 1 item. 18% of the bins will contain 2 items, 6% will contain 3 items, and only 1.5% will contain 4 items.

.

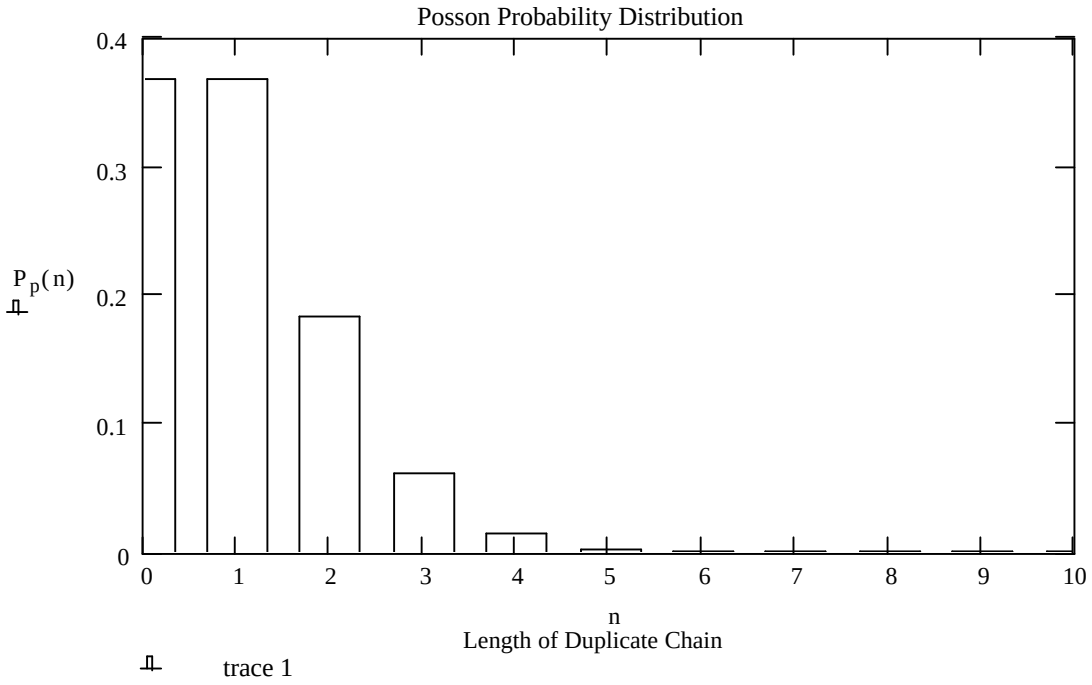

Figure 1

The expected number of bins (duplicate chains) in the NeoStore with n items is simply the size of the NeoStore times the probability of having n items in a duplicate chain, or

$$N_{dc}(n) = NP(n) \qquad [5.4]$$

Looking at the problem a little differently, we would like to know the expected number of quanta residing at different levels, i.e. the number of quanta at level 1, level 2, level 3, and so on. The expected number of quanta residing at level k is:

$$L(k) = N \sum_{n=k}^{K} P(n) \qquad [5.5]$$

It is interesting to plot [5.5] for different NeoStore sizes. The maximum expected duplicate chain length occurs when these plots cross the $L(k) = 1$ axis. Figure 2 show a NeoStore with $N = K = 1024 = 2^{10}$. The maximum expected duplicate chain length is only 5 to 6 quanta. Figure 3 show a NeoStore with $N = K = 2^{30}$. This is over 1 billion items. The maximum expected duplicate chain length is only 12 to 13 quanta. Chaining the size of the NeoStore by 6 orders of magnitude only doubled the expected duplicate chain length! The worst case number of memory cycles to find and access an item/association in a NeoCore vCAM will be the equal to the length of the longest duplicate chain. Therefore, for a full NeoCore vCAM with 1 billion items, the expected worst case number of memory cycles to access an association will be 12 memory cycles.

.

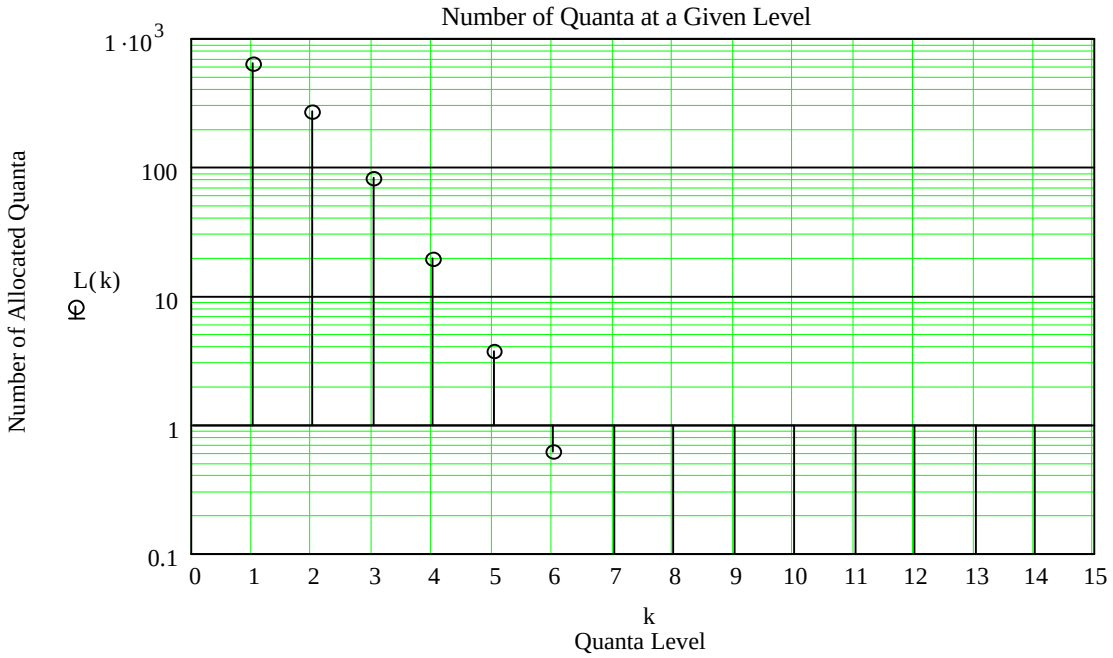

Figure 2, NeoStore with 1024 Items

.

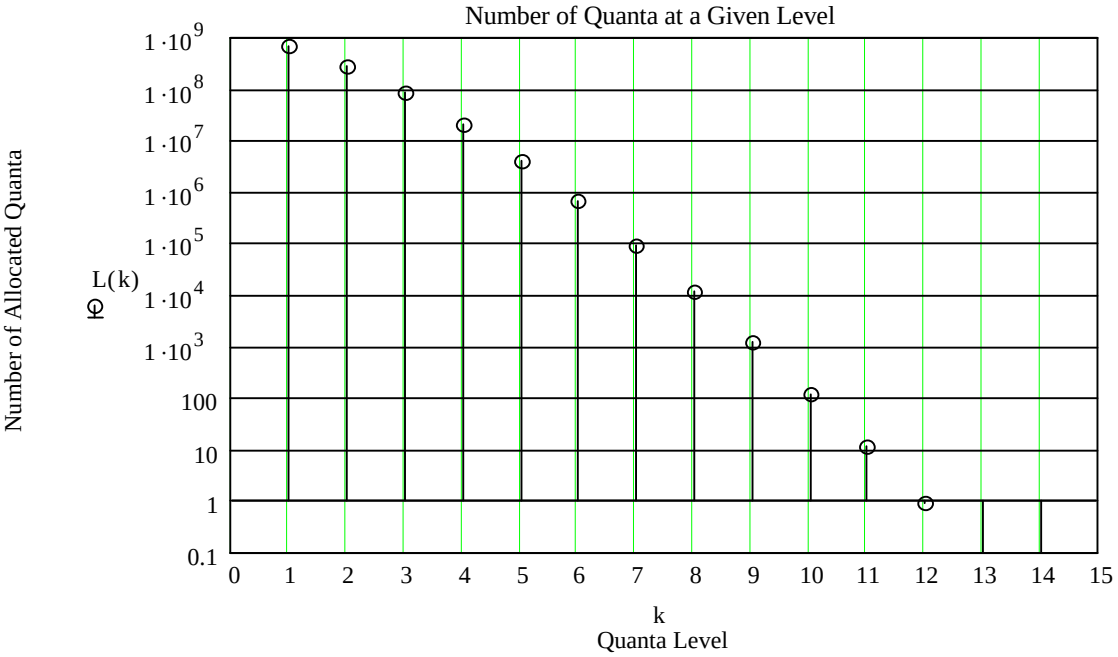

Figure 3, NeoStore with 2^{30} Items

[5.4] Expected Number of Memory Cycles to Access a Quanta: Dividing the expected number of level k quanta by the number of allocated quanta will give the probability of choosing at random a level k quanta:

$$P_L(k) = \frac{1}{K}L(k) = \frac{N}{K}\sum_{n=k}^{K} P(n) \qquad [5.6]$$

Once again, for a full store $K = N$. And as noted above, for a full store $P(n)$ is independent of the NeoStore size, so the probability of choosing a level k quanta is independent of the store size for all practical size NeoStores. The expected (average) value of k is therefore:

$$E[k] = \sum_{k=1}^{K} kP_L(k) = 1.5 \qquad [5.7]$$

What this tells us is that for a full NeoStore of any size, the average look-up and access time for a given item/association in the NeoCore vCAM will be 1.5 memory cycles on the given computer system. If the NeoStore is not full, the average number of memory cycles will be less than 1.5.

References

[1] Fraleigh, John B. ”Abstract Algebra”, Addison-Wesley, 1999

[2] Lidl, Rudolf and Niederreiter, ”Introduction to Finite Fields and their Applications”, Cambridge University Press, 1994

[3] McEliece, Robert, ”Finite Fields for Computer Scientists and Engineers”, Kluwer Academic Publishers, 1987

[4] Papoulis, Athanasios, ”Probability, Random Variables, and Stochastic Processes”, McGraw-Hill Publishing Co., 1984

[5] Leon-Garcia, Alberto, ”Probability and Random Processes for Electrical Engineering”, Addison-Wesley Publishing Co., 1994