Galois Spinor Fields: A Home for All Computer Calculations?

Irvin Riley Jackson Jr. II

E-Systems, Inc. Garland Division, Garland, TX 75042

Abstract -This paper introduces the idea of a new and novel mathematical object, the Spinor Extension Galois Field. This object is conjectured to be of value in the decomposition of algorithms for a MIMD processor.

Introduction

MIMD computing has changed the way we look at algorithm design and implementation. The different computing elements are not generally executing the same flow in the algorithm. While structured programming seeks to enforce a planar structure on the algorithmic decomposition, the MIMD scene is quite a jumbled mess.

From instruction set definition through high-level algorithm design, there is no general mathematical model that is universally applied. From considerations in Digital Signal Processing theory we know that many computations can be factored into Fast Transforms (FT) when they are cast into a Galois Field (GF). Closure of the primitive polynomial field allows a measure of both elegance and representational fluidity that was not heretofore possible. Can all computation be expressed in a Galois Field? Is the Universal Computer in Ed Fredkin's Digital Mechanics operating on a GF? How do we handle parallel processing?

This paper addresses these concerns from the point of view of an extension to the standard Galois Field through the imbedding of the GF in a spinor space. The new mathematical objects, spinorial Galois Fields, are conjectured to yield a faithful representation. The paper addresses the following highlights:

1. Galois Field example - FFT
2. Motivation *via* Ungerboeck Coding Theory
3. Spinorial extensions to Galois Fields
4. Massively Parallel Computing

Galois Field example - FFT

A standard treatment of the FFT and other fast DSP style algorithms requires a knowledge of GF's. As an example, the Cooley-Tukey algorithm casts the one-dimensional Discrete Fourier Transform into a two-dimensional setting. The resultant factorization yields the faster algorithm.

The factoring of the input vector into primes allows the use of Fast Winograd Prime Transforms. These transforms change the DFT into a convolution using the Rader prime algorithm to convert to a convolution, where we can use a prime factor Winograd convolution algorithm. The original problem, to compute the Fourier transform of the input vector, is cast into a representation that is faster to compute. Can this be extended to a more general class of problems? The discovery of a good representation for the initial problem is the most critical step.

Motivation *via* Ungerboeck Coding Theory

While studying the original Ungerboeck Set Partitioning algorithm, I noticed the decoding trellis looked to be double valued, depending on which signal set the input came from. I am used to using an imbedded spinor representation when things get multivalued. I was surprised to find that spinors and GF's are considered to be orthogonal ideas. Can we marry the two mathematical forms? Does the use of N-dimensional spinors lead to an N-dimensional signal set partitioning algorithm?

Spinorial Extensions to Galois Fields

Just as we form an extenuation field for the Galois Extension Fields, can we imbed the GF into a spinor field? In Quantum Mechanics the natural use of spinors occurs as an orthogonal basis element in the description of the hydrogen atom's spectra. The radial component of the wave function separates as do the angular components. The spin is introduced orthogonally to this. Using the spinor representation with a GF allows the GF to become multivalued. Each orthogonal spinor component describes an independent part of the algorithm! Can this be generally used for algorithm design and implementation?

Massively Parallel Computing

The whole idea here is to be able to track and decompose the algorithm into a set of hardware. One method I designed in early 1985 formed an hierarchy of increasingly complex planar graphs (non-planar graphs, mapped onto K_{33} and K_5, were not handled automatically, but as exceptions by hand). These graphs were considered as the basis graphs for a decomposition of general algorithms over parallel hardware. The idea was extended to Hyper-graphs to handle the different data types (integer, float, double, *etc.*). An heuristic expert system mapped the algorithm on the simplest set of basis graphs. The biggest problem was the lack of a good representation that would allow us to remove the heuristic step. By factoring the algorithm over an N component spinor, where N is the number of hardware processors, the heuristic step can be avoided.

It has been argued that Spinorially extended GF's are useful in the algorithmic decomposition into hardware. It is this author's belief that the extension is global in applicability. Must the Digital Universe require the equivalent representation for electron spins?

Bibliography

1. Blahut, Richard. 1985. *Fast Algorithms for Digital Signal Processing,* Addison-Wesley Publishing Company, Reading, Massachusetts.

2. Jacobson, Nathan. 1962. *Lie Algebras.* Dover Publications, Inc., New York.

3. Messiah, Albert. 1962. *Quantum Mechanics.* North-Holland Publishing Company, Amsterdam.

4. Penrose, Roger and Rindler, Wolfgang. 1984. *Spinors and Space-Time, Volume I, Two-spinor calculus and relativistic fields.* Cambridge University Press, Cambridge, England.