INTEGRATION, the VLSI journal 42 (2009) 346–355

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

An embedded, FPGA-based computer graphics coprocessor with native geometric algebra support

Silvia Franchini [a], Antonio Gentile [a], Filippo Sorbello [a], Giorgio Vassallo [a], Salvatore Vitabile [b,*]

[a] Dipartimento di Ingegneria Informatica, Università di Palermo, V.le delle Scienze (Edificio 6), 90128 Palermo, Italy
[b] Dipartimento di Biotecnologie Mediche e Medicina Legale, Università di Palermo, Via del Vespro, 90127 Palermo, Italy

ARTICLE INFO

Article history:
Received 19 March 2008
Received in revised form
16 September 2008
Accepted 17 September 2008

Keywords:
Clifford algebra
Computational geometry
Embedded coprocessors
Application-specific processor
FPGA-based prototyping

ABSTRACT

The representation of geometric objects and their transformation are the two key aspects in computer graphics applications. Traditionally, computer-intensive matrix calculations are involved in modeling and rendering three-dimensional (3D) scenery. Geometric algebra (aka Clifford algebra) is attracting attention as a natural way to model geometric facts and as a powerful analytical tool for symbolic calculations. In this paper, the architecture of Clifford coprocessor (CliffoSor) is introduced. CliffoSor is an embedded parallel coprocessing core that offers direct hardware support to Clifford algebra operators. A prototype implementation on a programmable gate array (FPGA) board is detailed. Initial test results show the potential to achieve a 20× speedup for 3D vector rotations, a 12× speedup for Clifford sums and differences, and more than a 4× speedup for Clifford products, compared to the analogous operations in GAIGEN, a standard geometric algebra library generator for general-purpose processors. An execution analysis of a raytracing application is also presented.

1. Introduction

Many research fields, such as machine vision, robotics, and computer graphics, rely heavily on geometric models of external reality. Software packages and graphics hardware accelerators have been designed to deal with the representation of points, lines, and planes and their transformations in three-dimensional (3D) space, such as rotations, translations, reflections, and projections.

Traditionally, computer graphics packages are implemented using homogeneous coordinates and a matrix package [1]. This approach creates a separation between geometric reasoning and matrix-based processing. Programming geometric models requires reasoning in affine and projective geometries; whereas the implementation of graphic objects and their transformations using a matrix formulation require a geometric interpretation of vector calculations that is often problematic and prone to error.

Geometric algebra, which has been studied for more than a century, is attracting attention in recent years as a powerful new computing paradigm for a number of research fields. It offers a natural way to model geometric objects independent of their coordinates with its powerful and simple symbolic formalism [2]. Also known as Clifford algebra (from its main contributor, W. K. Clifford [3]), this type of algebra offers an integrated approach to geometric modeling and algorithms. It permits the specification of geometric objects at a coordinate-free, unified level, where points, lines, and planes become basic elements of computation and their transformations are executed directly [2,4,5].

The exploitation of geometric algebra's symbolic computing power requires efficient support for its powerful operators and data types. Previous implementations had to resort to changing complex translations into matrix equations and standard matrix libraries. Many software libraries (CLU [6], GluCat [7]), packages for symbolic and numerical environments (Maple [19,20], Mathematica, MatLab), and stand-alone programs (CLUit [6], CLICAL [8]) have been developed to solve geometric algebra expressions directly on general-purpose processors. In these implementations, a hierarchy of abstraction layers is used to program geometric algebra expressions, lowering their computational advantage through the introduction of significant overhead. These approaches address general *n*-dimensional Clifford algebra, whereas in the case of computer graphics and machine vision applications, an optimized four-dimensional (4D) implementation should be sought as a good compromise between complexity and benefits. The GAIGEN library generator [9] is a software library that addresses this need.

This paper examines the direct support of geometric algebra data types and operators in hardware. Related research [10] presents a coprocessor for native geometric product execution, for algebras of dimension up to 8. A geometric product is executed by computing the components of its basis blades in a pipeline

* Corresponding author. Fax: +390916529124.
E-mail address: vitabile@unipa.it (S. Vitabile).

0167-9260/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.vlsi.2008.09.010

fashion. No other geometric algebra operation is supported directly by the coprocessor, with the exception of a prototype implementation on a field-programmable gate array (FPGA) board.

This paper presents a novel coprocessing architecture, the Clifford coprocessor (*CliffoSor*). Aimed at embedded machine vision and computer graphics applications, either as part of robotic platforms or dedicated graphic boards, this architecture directly supports 4D homogeneous operands (namely scalars, vectors, bivectors, trivectors, and pseudoscalars) and their operations (geometric products, outer products, left and right contractions, sums and differences). Clifford numbers, their expressions in terms of homogeneous numbers, and related operations are described in Section 2. The CliffoSor architecture is currently implemented as a coprocessor core hosted on a PCI-based FPGA board. The core is described using a mixed hardware description language design that incorporates both Handel-C and VHDL hardware description languages. An early version of the CliffoSor architecture was presented in [11,12]. Test results are presented for 4D geometric products, 4D sums/differences, and 3D rotations on CliffoSor, compared to the same operations implemented for general-purpose processors by the 4D library generated by the GAIGEN library generator [9]. In addition, a raytracer application and 3D model rotation have been implemented and executed on CliffoSor to prove its effectiveness on its application domain. These results suggest that there is great potential for speedup in comparison with the execution on a traditional processor. To realize this potential, however, the core must be placed as close as possible to the memory subsystem, with dedicated DMA access. The potential cycle speedup is demonstrated for products ($4\times$), sums ($12\times$), and rotations ($20\times$).

The paper is organized as follows. Section 2 briefly introduces geometric algebra and its implications for computer graphics. In Section 3, design considerations are discussed along with their implications for efficient hardware implementation of operand types and operations. Section 4 details the architecture of CliffoSor and its implementation of FPGA. Experimental results are described in Section 5, and Section 6 summarizes the conclusions from this work.

2. Geometric algebra and computer graphics

Geometric algebra (Clifford algebra) is a powerful mathematical tool for symbolic calculations. It is applied to different fields of research such as computer graphics, CAD/CAM, robotics, physics, and any application in which description and manipulation of geometric entities are very important. As a unifying mathematical language, geometric algebra comprises a number of mathematical descriptions widely used in computer graphics, such as quaternions to represent rotations and Lie algebras for rigid body motion descriptions [13]. A brief introduction to the theory is given in the following sections. Interested readers may find further details in [2,4,5,14,17,18,22].

2.1. Theoretical background

Clifford algebra expands classical linear algebra concepts, such as scalars and vectors, by introducing two-, three-, or higher-dimensional subspaces, called blades. In Clifford algebra, vectors can be combined using the outer product to obtain higher-dimensional entities, such as bivectors (representing planes) and trivectors (representing 3D subspaces). For example, if $\mathbf{a}$ and $\mathbf{b}$ are vectors, their outer product $\mathbf{a}\wedge\mathbf{b}$ is a blade of *grade* 2 (or a 2-*blade*), which represents the two-dimensional oriented subspace that contains $\mathbf{a}$ and $\mathbf{b}$. Such blades of grade 2 are called *bivectors*. The outer product of three vectors results in a blade of grade 3 (also called a *trivector*), and so on. According to this formalism, vectors are 1-blades, and scalars are 0-blades. Geometric algebra also defines the operators with which these subspaces can be manipulated. It is possible, in fact, to add and subtract subspaces of different dimensions by means of composition, and even to multiply them, resulting in powerful expressions that can express many geometric relations and concepts.

Considering an orthonormal basis $\{\boldsymbol{e}_1,\boldsymbol{e}_2,\ldots,\boldsymbol{e}_n\}$ of R^n, a possible basis of the n-dimensional Clifford algebra, Cl_n, consists of all k-dimensional subspaces with $k\leqslant n$, as listed in Table 1. The total number of these subspaces is 2^n, while the number of k-dimensional subspaces is given by the binomial coefficient $n!/(k!(n-k)!)$. The basis blade of the highest dimension is called the *pseudoscalar*. A generic number in Cl_n is called a *multivector*, which is a linear combination with real coefficients of the previously mentioned basis elements.

The most important operator of Clifford algebra is the *geometric product* (also called the *Clifford product*). The geometric product between two multivectors is performed by "multiplying" each component of an operand individually with each component of the other operand, and simplifying the resulting terms by means of the set of axioms listed in Table 2.

A particular type of multivector is the *rotor*, formed by a scalar element and a bivector element. Rotors are used to perform rotations of subspaces of any dimension. If $\boldsymbol{R}$ denotes a rotor and $\boldsymbol{v}$ denotes a vector, the rotated vector $\boldsymbol{v}'$ can be calculated as $\boldsymbol{v}'=\boldsymbol{R}\boldsymbol{v}\boldsymbol{R}^{\dagger}$, where $\boldsymbol{R}^{\dagger}$ is the reverse of $\boldsymbol{R}$. Interestingly, a generic rotor can be derived from the geometric product of two unitary vectors, where the rotation is in the plane determined by the two vectors, and the rotation angle is twice the angle between them.

2.2. 4D Clifford homogeneous numbers

In the case of computer graphics and machine vision applications, 4D geometric algebra is a reasonable compromise between complexity and benefits. A generic Clifford algebra multivector is a linear combination with real coefficients of the basis elements of Clifford space; in four dimensions, a generic multivector can be written as

$$a_0+a_1\boldsymbol{e}_1+a_2\boldsymbol{e}_2+a_3\boldsymbol{e}_3+a_4\boldsymbol{e}_4+a_{12}\boldsymbol{e}_1\boldsymbol{e}_2+a_{13}\boldsymbol{e}_1\boldsymbol{e}_3+a_{14}\boldsymbol{e}_1\boldsymbol{e}_4+a_{23}\boldsymbol{e}_2\boldsymbol{e}_3+a_{24}\boldsymbol{e}_2\boldsymbol{e}_4+a_{34}\boldsymbol{e}_3\boldsymbol{e}_4+a_{123}\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3+a_{124}\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_4+a_{134}\boldsymbol{e}_1\boldsymbol{e}_3\boldsymbol{e}_4+a_{234}\boldsymbol{e}_2\boldsymbol{e}_3\boldsymbol{e}_4+a_{1234}\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3\boldsymbol{e}_4$$

It is possible to collect generic multivector basis elements of the same grade to build 4D homogeneous Clifford numbers: *scalar*, *vector*, *bivector*, *trivector*, and *pseudoscalar* as shown in Table 3.

Table 1
Basis elements (blades) in n-dimensional Clifford algebra.

Dimension	Element	Blade
0	Scalar	1
1	Vector	$\boldsymbol{e}_i$
2	Bivector	$\boldsymbol{e}_i\boldsymbol{e}_j$
3	Trivector	$\boldsymbol{e}_i\boldsymbol{e}_j\boldsymbol{e}_k$
…	…	…
n	Pseudoscalar	$\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3\ldots\boldsymbol{e}_n$

Table 2
Clifford product axioms.

$\boldsymbol{e}_i\boldsymbol{e}_i=\pm1$	$i=1,2,\ldots,n$
$\boldsymbol{e}_j\boldsymbol{e}_i=-\boldsymbol{e}_i\boldsymbol{e}_j$	$i\neq j=1,2,\ldots,n$
$\lambda\boldsymbol{e}_i=e_i\lambda$	$i=1,2,\ldots,n$