PROCEEDINGS

Eighth Biennial University/Government/Industry Microelectronics Symposium

June 12-14, 1989
Westborough, Massachusetts

Sponsored by

The Institute of Electrical and Electronics Engineers, Inc.
Worcester Section
Electron Devices Society
Components, Hybrids, and Manufacturing Technology Society

The International Society for Hybrid Microelectronics

Massachusetts Microelectronics Center

VLSI Implementation of a CORDIC SVD Processor

Joseph R. Cavallaro, Michael P. Keleher, Russell H. Price, and Gregory S. Thomas

Dept. of Electrical and Computer Engineering

Rice University

Houston, Texas 77251

Abstract

The design and custom CMOS VLSI implementation of a CORDIC SVD processor is presented. Special-purpose parallel processor arrays have many important applications in real-time signal processing. The processor architecture is reviewed and the current CORDIC Z Control and X,Y Data Path chips are described. The hierarchical design methodology will lead next to a full CORDIC processor followed by a complete CORDIC SVD processor and array.

1. Introduction

The Singular Value Decomposition (SVD) of a matrix is an important computationally complex algorithm which can benefit from the recent advances in parallel architectures and VLSI. The array structure of Brent, Luk, and Van Loan [2] uses an expandable square systolic array of simple 2×2 processors to compute the SVD of a large matrix. A VLSI processor array for the SVD would have applications in real-time signal processing and image processing.

Algorithms for use on uniprocessor systems require many division and square root calculations to compute the necessary sine and cosine rotation parameters. The Coordinate Rotation (CORDIC) algorithms [11] provide efficient hardware calculation of vector rotations and inverse tangents. We have previously presented the architecture of a CORDIC 2×2 SVD processor [5]. In this paper, we describe the VLSI implementation of this architecture.

As part of the first author's thesis research at Cornell, the design of a prototype custom CMOS VLSI CORDIC X,Y Data Path Chip was begun. This chip consists of a 20-bit register, barrel shifter, and adder. There is also a data router that provides for both normal CORDIC and scale factor correction operation. The chip contains approximately 2100 transistors in addition to those contained in the pad frame. It will be submitted for fabrication by MOSIS.

At Rice, other design units are being used as projects for the VLSI Design course. These projects emphasize a conservative design approach and final product functionality as opposed to area and time minimization. A 10-bit CORDIC Z Control Chip has been completed and fabricated through the MOSIS service in a $3\text{-}\mu$ CMOS p-well process. Angle data, in this chip, is represented with 10 bits. This chip contains approximately 1200 transistors in addition to the pads. This chip is fully functional up to a clock speed of 8 MHz. A CORDIC processor can be implemented using the CORDIC Z Control Chip interconnected with two CORDIC X,Y Data Path Chips. Current plans include implementing the complete CORDIC SVD processor on a single chip and minimizing critical timing paths in the processor design.

2. Singular Value Decomposition

The singular value decomposition of an $p \times p$ matrix M is

$$M = U \Sigma V^T,$$

where U and V are orthogonal matrices and Σ is a diagonal matrix of singular values. In the Brent, Luk, Van Loan array, the matrix is divided into 2×2 submatrices. Each processor element contains a 2×2 submatrix. The array architecture is scalable. There are two types of data flowing in this array. Rotation angles generated by the diagonal processors flow systolically along the rows and columns of the array. Matrix data elements are exchanged diagonally, after the diagonal neighbor has received and applied the necessary rotation angles. This leads to "waves" of activity moving diagonally away from the main array diagonal.

A 2×2 SVD can be described as

$$R(\theta_l)^T \begin{bmatrix} a & b \\ c & d \end{bmatrix} R(\theta_r) = \begin{bmatrix} \psi_1 & 0 \\ 0 & \psi_2 \end{bmatrix},$$

where θ_l and θ_r are the left and right rotation angles, respectively. The rotation matrix is

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix},$$

and the input matrix is

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

The angles θ_l and θ_r are found by the direct two-angle method by computing the inverse tangents of the data elements of M . Given M as defined above, θ_{SUM} and θ_{DIFF} are

$$\theta_{SUM} = (\theta_l + \theta_r) = \tan^{-1} \left[\frac{c+b}{d-a} \right],$$

$$\theta_{DIFF} = (\theta_r - \theta_l) = \tan^{-1} \left[\frac{c-b}{d+a} \right].$$

The sum and difference angles yield the two angles, θ_l and θ_r , which can be applied to the two-sided rotation module to diagonalize M .

3. CORDIC Arithmetic for an SVD Processor

In a typical computer algorithm for the SVD, the sines and cosines of the rotation angles are computed through formulas that require division and square root operations. The explicit angles are not required and only the sines and cosines are computed. The rotations are then applied to the 2×2 submatrix using standard matrix multiplication techniques. However, time-consuming operations such as multiplication, division, and square root are needed.

The SVD is more closely mapped onto hardware through the use of the CORDIC algorithms [11]. The CORDIC algorithms can provide the calculation of vector rotation and inverse tangent. The CORDIC iteration equations which are to be implemented in hardware are:

$$x_{i+1} = x_i + \delta_i y_i 2^{-i},$$

$$y_{i+1} = y_i - \delta_i x_i 2^{-i},$$

$$z_{i+1} = z_i + \delta_i \theta_i.$$

The variable, z_i , contains the total rotation angle applied, θ_i is the current rotation angle increment, and $\delta_i = \pm 1$. Through the appropriate selection of each δ_i , either the initial z_0 value can be reduced to zero (vector rotation) or the initial y_0 value can be reduced to zero (inverse tangent). These equations can be implemented with simple structures: registers, shifters, adders, and a small ROM.

If CORDIC processors are used, then the rotation parameters can be calculated from the inverse tangents of the elements of M . The rotation angles are found explicitly, without penalty, since the inverse tangent function is a primitive CORDIC operation. The traditional matrix-vector multiplication can also be replaced since vector rotations are primitive CORDIC operations. The diagonalization of M can be performed by treating M as a pair of vectors and using the rotation angles to transform M . The computation of these vector rotations and inverse tangents can be performed efficiently by the CORDIC algorithms. Thus, a general algorithm for a 2×2 CORDIC SVD processor would be:

Algorithm CORDIC SVD ( ) :
begin
     Use CORDIC angle-solver module to
     find rotation angles;
     Use CORDIC rotation module to
     transform the 2×2 matrix;
end.

A novel architecture which applies the CORDIC arithmetic algorithms to create a special-purpose 2×2 SVD processor has been presented by us [5] and is shown in Figure 1. There is continuing interest in the implementation of CORDIC [6] and other rotation based algorithms [3].

Figure 1. CORDIC SVD Parallel Diagonalization Method.

4. CORDIC SVD Processor VLSI Implementation

In the prototype fixed-point arithmetic system, the CORDIC Parallel Diagonalization Method would be used since the least time and area are needed and the structure is regular. The basic floor-plan of a VLSI implementation, shown in Figure 2, contains three major sections: two CORDIC processors, and an interconnection network.

A 16-bit fixed-point implementation of the CORDIC SVD architecture is described. In a fixed-point implementation, the number of bits in the internal data representation must be chosen to prevent loss of significance due to rounding and overflow. Therefore, a 16-bit fixed-point implementation of the CORDIC SVD architecture requires 20-bit internal arithmetic. A prototype custom CMOS VLSI chip is currently being designed and will be fabricated through the MOSIS fabrication service.

The intra-module interconnection network will allow the same chip to function as both an angle solver and a rotation module, and will permit flexibility in designing, constructing, and reconfiguring a large array. Finite state control for the interconnection network and for the SVD algorithm will be provided by a PLA. The array will be connected in a mesh configuration. Each module will possess the necessary control for systolic I/O.

The systolic control network schedules the communication among the processor elements of the array. Each processor communicates with its eight nearest neighbors. This communication requirement increases both the complexity of the VLSI layout and of the packaging technology. However, the pin count of the chip can be reduced by restricting the external data paths to nibbles (4 bits). This multiplexing approach increases the time required for communication. Hence, 32 bi-directional pads may be used for data transfer. Various control signals are also necessary. Each processor has a particular address which effects the data communication algorithm, especially at the edges of the array. The horizontal and vertical coordinates can be input serially during the initial array configuration phase. Signals to indicate *start*, *stop*, *setup*, and *done* are required in addition to the standard two phase clock signals (ϕ_1, ϕ_2), power (V_{dd}), and ground (GND).

Figure 2. CORDIC SVD Diagonalization Module.

The internal structure of a fixed-point CORDIC module contains three data paths. The x and y data paths implement the CORDIC rotation equations. The z data path is used to accumulate angle information. Each module is composed of three registers and adders, two barrel shifters, and a local PLA control unit. The ROM which stores the angles can be shared between the two CORDIC modules on the chip. In order to begin the implementation of the processor array, the floorplan was first separated into smaller design units.

In order to approach the design of a moderately complex special-purpose processor in an academic setting, the CORDIC SVD processor was partitioned into *TinyChip* projects for initial design study. The full processor design is divided into the design of a CORDIC processor and a systolic interconnection network. The CORDIC processor is further divided into a CORDIC Z Control Chip and a CORDIC X,Y Data Path Chip as shown in Figure 3. Experience gained from the *TinyChip* projects is valuable to the design of larger chips. Each subsection is appropriate for a VLSI Design Course project, while the larger chips are appropriate for funded research projects.

At Rice, an initial CORDIC Control chip was recently designed. This chip includes the internal CORDIC control PLA and ROM. The 10-bit CORDIC Z Control Chip has

been completed and was fabricated through the MOSIS service. A block diagram of the chip is shown in Figure 4. The major structures are: control PLA, adder, ROM, ROM selector, and registers. CMOS transmission gates are used to control data flow within the chip. This chip contains approximately 1200 transistors in addition to the pads and a photograph is shown in Figure 5. The chip has been tested and is fully functional up to a clock speed of 8 MHz.

The design of the Data Path chip was started at Cornell using the 3- μ double-metal, single polysilicon, p-well process. The x and y data paths implement the CORDIC iteration equations. A plot of the current 20-bit Data Path chip design which could implement either the x or y data path is shown in Figure 6. The major cells are: register, input to barrel shifter for arithmetic shift, barrel shifter array, control for adder two-phase clocking, switch for normal CORDIC and CORDIC scale factor correction data routing, and adder array. In this project, basic ripple-carry addition has been assumed for simplicity of implementation.

When completed, the 20-bit Data Path chip will contain approximately 2100 transistors in addition to those contained in the twenty-eight I/O pads. The area is $1146\mu \times 2582\mu$. This design will fit into the MOSIS *TinyChip* padframe which is $2300\mu \times 3400\mu$. Currently, a CORDIC processor would be composed of two Data Path chips and a CORDIC Z Control Chip as indicated in Figure 3.

Structured VLSI design is used following the Mead and Conway design methodology [9,12]. In order to simplify design requirements, static complementary logic is used. In addition, a formal two-phase clocking strategy is

Figure 4. Block Diagram of CORDIC Z Control Chip.

Figure 5. Photograph of 10-bit CORDIC Z Control Chip.

imposed [7, 8]. The design was developed with the *Magic* layout editor and other VLSI design tools from the University of California, Berkeley [10]. Layout is done in a bit-slice manner, and PLA generation tools are used for the control structures. Simulation is done at the switch level with *esim*, at the timing level with *crystal*, and at the circuit level with *spice*. Chips are designed for the scalable CMOS technology from the MOSIS fabrication service.

Functional testing of the fabricated chips is performed with *digtest* [1]. This program utilizes the test vectors created for the *esim* simulator. *Digtest* runs on an IBM-PC equipped with a parallel interface board. High-speed testing is performed with a 15 MHz two-phase clock source and an oscilloscope.

7. Current Work and Summary

The goal of the CORDIC X,Y Data Path and CORDIC Z Control chips has been to give insight into the design of a full CORDIC SVD processor chip. Current work includes the expansion of the 10-bit CORDIC Z Control chip to a 20-bit design to complement the CORDIC X,Y Data Path design. The two designs will then be combined and replicated to yield two full CORDIC processors, as shown in Figure 2. The systolic control structure and internal data routing will then be added to complete the SVD processor element. Eventual design goals include a floating-point implementation based upon a hybrid floating-point architecture [4]. Initial timing results from the CORDIC Z Control Chip indicate that a 16-bit fixed-point array of 10×10 processors should compute the SVD of a 20×20 matrix in 1 millisecond. It is expected that greater speed can be achieved through the use of carry look-ahead adder designs and the

Figure 6. Plot of current 20-bit CORDIC X,Y Data Path Chip.

minimization of critical timing paths. Also, each processor chip will contain approximately 25,000 transistors, based upon the transistor count of the two chips currently designed.

An array of 100 chips to compute the SVD of a 20×20 matrix, could be built using either conventional packaging or surface mount technology. Larger CORDIC SVD processor arrays would greatly benefit from the emerging Wafer Scale Integration (WSI) technologies. Advances in fault tolerance techniques at both the architectural and manufacturing levels will improve the yield of these systems.

The design of special-purpose processor chips is always challenging. As general-purpose microprocessors continue to advance, special-purpose processors remain useful for custom applications. The CORDIC SVD processor array would be important for specialized, portable, real-time applications where the benefits of VLSI are great.

References

1. Aneshansley, D. J., Pottle, C., and Cavallaro, J. R., "Laboratory Workstations in Electrical Engineering," *Proc. IBM Academic Information Systems Advanced Education Projects Conference*, pp. 211-229, Alexandria, VA (June 1985).
2. Brent, R. P., Luk, F. T., and Van Loan, C. F., "Computation of the Singular Value Decomposition Using Mesh-Connected Processors," *Journal of VLSI and Computer Systems*, 1,(3), pp. 242-270 (1985).
3. Burleson, W. P., Scharf, L. L., Gabriel, A. R., and Endsley, N. H., "A Systolic VLSI Chip for Implementing Orthogonal Transforms," *IEEE Journal of Solid-State Circuits*, 24,(2), pp. 466-469 (April 1989).

4. Cavallaro, J. R. and Luk, F. T., "Floating-Point CORDIC for Matrix Computations," *IEEE Int. Conf. on Computer Design*, pp. 40-42, Rye Brook, NY (October 1988).
5. Cavallaro, J. R. and Luk, F. T., "CORDIC Arithmetic for an SVD Processor," *Journal of Parallel and Distributed Computing*, pp. 271-290 (June 1988).
6. Haviland, G. L. and Tuszynski, A. A., "A CORDIC Arithmetic Processor Chip," *IEEE Trans. Computers*, C-29,(2), pp. 68-79 (Feb. 1980).
7. Karplus, K., "A Formal Model for MOS Clocking Disciplines," Technical Report 84-632, Dept. of Computer Science, Cornell Univ., Ithaca, NY (August 1984).
8. Karplus, K., "Exclusion Constraints, A New Application of Graph Algorithms To VLSI Design," *Advanced Research in VLSI, Proc. 4th MIT Conference*, pp. 123-139, Cambridge, MA (April 1986).
9. Mead, C. A. and Conway, L. A., *Introduction to VLSI Systems*, Addison-Wesley, Reading, MA (1980).
10. Ousterhout, J. K., Hamachi, G. T., Mayo, R. N., Scott, W. S., and Taylor, G. S., "Magic, a VLSI Layout System," *ACM/IEEE 21st Design Automation Conference*, pp. 152-159 (June 1984).
11. Volder, J., "The CORDIC Trigonometric Computing Technique," *IRE Trans. Electronic Computers*, EC-8,(3), pp. 330-334 (Sept. 1959).
12. Weste, N. H. E. and Eshraghian, K., *Principles of CMOS VLSI Design*, Addison-Wesley Publishing Company, Reading, MA (1985).