RESEARCHERID
THOMSON REUTERS

[Verma* *et al.*, 6(4): April, 2017]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

EFFICIENCY IMPROVEMENT OF SD PROCESSOR BASED ON CRDC ALGORITHM

Raj Kumar Verma*, Mrs. Akanksha Awasthi
* Research Scholar, C.V.R.U, Bilaspur (c. g.) India
Asst. Prof. C.V.R.U., Bilaspur (c. g.) India

DOI: 10.5281/zenodo.556252

ABSTRACT

One such complex algorithm is Singular-value Decomposition (SD) which is an important algorithm with applications in varied domains of signal processing such as direction estimation, spectrum analysis and systems identification. It is a generalized extension to the eigen-decomposition for non-square matrices and is hence of great importance, particularly for subspace based algorithms in signal processing. But SD is known to be a very complicated algorithm with computational complexity ~O (N3) (for a N x N square matrix). For real-time computation of such a complex algorithm the use of a parallel and direct mapped hardware solution is indeed desired.

The Singular Value Decomposition (SD) is an important matrix factorization with applications in signal processing, image processing and robotics. It is generally acknowledged that the SD is the only generally reliable method for determining the rank of a matrix numerically. The SD is a very useful tool, for example, in analyzing data matrices from sensor arrays for adaptive beamforming, and low rank approximations to matrices in image enhancement. The wide variety of applications for the SD coupled with its computational complexity justifies dedicating hardware to this computation. Designed 2X2 CRDC based SD processor can be used as basic building block for an array of processors of N X N matrix.

Hardware and software resources

1. OS: Windows 9x or upper
2. RAM: Minimum 512 MB
3. Programming Language: XILINX 8.6 or upper
4. Tools: MODELSIM 5.2 or upper
5. Processor: AMD Sempron 1.6 GHz/ Intel P4 2.8 GHz.

INTRODUCTION

Keeping in mind the useful applications of SD with associated complexity involved in its computation motivates us towards an efficient hardware implementation. So, in this project design and implementation of a 2X2 matrix SD processor is presented. The problems which come on the way and the solutions for those are explained in next section.

In Digital Signal Processing (DSP) there are many complex algorithms for which an efficient hardware implementation is required in real time applications. One such complex algorithm is Singular-value Decomposition (SD)[1] which is an important algorithm with applications in varied domains of signal processing such as direction estimation, spectrum analysis and systems identification. It is a generalized extension to the eigen-decomposition for non-square matrices and is hence of great importance, particularly for subspace based algorithms in signal processing. But SD is known to be a very complicated algorithm with computational complexity ~O(N^3) (for a NxN square matrix). For real-time computation of such a complex algorithm the use of a parallel and direct mapped hardware solution is indeed desired.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IMPORTANCE

The Singular Value Decomposition (SD) is an important matrix factorization with applications in signal processing, image processing and robotics. It is generally acknowledged that the SD is the only generally reliable method for determining the rank of a matrix numerically. The SD is a very useful tool, for example, in analyzing data matrices from sensor arrays for adaptive beamforming [2], and low rank approximations to matrices in image enhancement [3]. The wide variety of applications for the SD coupled with its computational complexity justifies dedicating hardware to this computation. Designed 2X2 CRDC based SD processor can be used as basic building block for an array of processors of N X N matrix.

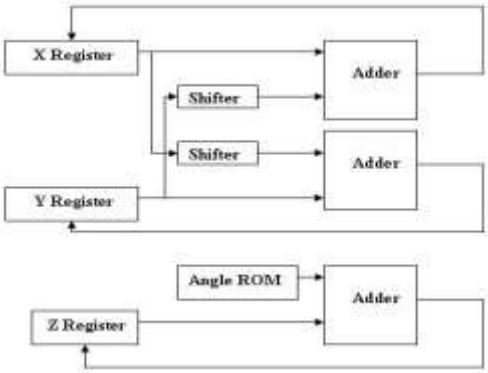
LINE OF APPROACH

The complexity of SD computation and its real-time applications were the inspiration behind efficient hardware implementation. In SD processor time consuming operations like multiplication, square root and division are eliminated by the use of CRDC algorithm. CRDC algorithm only requires simple shift and additions which efficiently maps SD processor**.**

Here for implementation of a 2X2 matrix CRDC based SD processor, parallel diagonalization architecture has been selected in which two CRDIC engines are utilized. Two CRDC engines operate parallely to compute SD of a given matrix in four steps. If only one CRDC engine had been used then it would have taken almost double time. Thus parallel pipelined CRDC architecture greatly shortens the computation time. Further Kogge-Stone adders are implemented in CRDC engines to speed up the addition and subtraction process. Kogge-Stone adders are known as the fastest adder and are from the family of parallel prefix adders. The net result is a 2X2 CRDC based SD processor with pipelined architecture and high speed Kogge-Stone adder in its CRDC engine to make the processor achieve maximum parallelism. The processor so designed is also able to calculate sine, cosine and inverse tangent values as these are basic CRDC functions. So the processor can be useful in those applications also where real time trigonometric values are needed to be calculated.

Basic CRDC architecture

An iterative CRDC architecture can be obtained simply by duplicating each of the three difference equations in hardware as shown in Figure.



The decision function, d_i is driven by the sign of the y or z register depending on whether it is operated in rotation ($\mathbf{z_i}$) or vectoring mode ($\mathbf{y_i}$). In operation, the initial values are loaded via multiplexers into the x, y and z registers. Then on each of the next n clock cycles, the values from the registers are passed through the shifters and adders-sub tractors and the results placed back in the registers. The shifters are modified on each iteration to cause the desired shift for the iteration. Likewise, the ROM (Look-up table) address is incremented on each iteration so that the appropriate elementary angle value is presented to the z adder-sub tractor.

Singular-value Decomposition (SD)

In linear algebra, the singular value decomposition (SD) is an important factorization of a rectangular real or complex matrix, with several applications in signal processing and statistics. The Singular-value Decomposition (SD) of a matrix is an important computationally complex algorithm which can benefit from the recent advances in parallel architectures and VLSI. A VLSI processor array for the SD would have applications in real-time signal processing and image processing.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

A singular value decomposition of a $P \times P$ matrix M is

$M = U\Sigma V^T$,

A 2×2 SD 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 ψ_1 and ψ_2 are the Eigen values. 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

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

If CRDC 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 CRDC operation. The traditional matrix-vector multiplication *can* also be replaced since vector rotations are primitive CRDC 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 CRDC algorithms.

Parallel Diagonalization SD

The Parallel Diagonalization Method is based on determining θ_{SUM} and θ_{DIFF} directly. This results in reduction in time and area necessary for a 2×2 SD processor as shown below in figure 3. From the block diagram elements of the matrix are provided to both CRDC modules which calculate arctan values (θ_{SUM} and θ_{DIFF}) then these values are added and subtracted to obtain angles (θ_l and θ_r). Finally, these angles are applied to the CRDC block to calculate sin/cos of the θ_l , θ_r using CRDC processor. Whole process has also been shown in algorithmic form below.

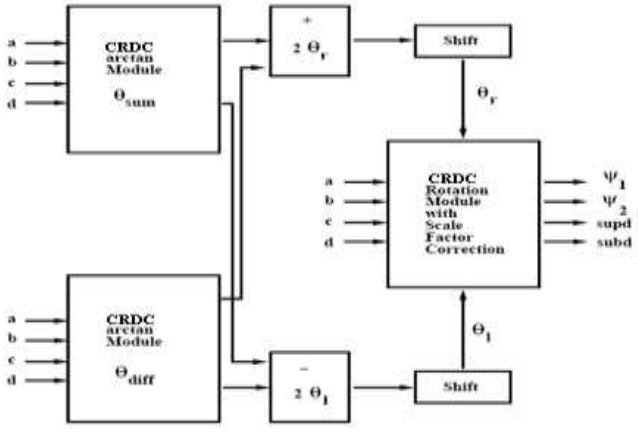


Figure: CRDC SD Parallel Diagonalization Method

The algorithm applying the parallel diagonalization method becomes:

CRDC SD Parallel ():

Begin

Parallel do $\{b+c, c-b, d-a, d+a\}$,

Parallel do begin

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Find $\theta_{SUM} = (\theta_r + \theta_l)$;

Find $\theta_{DIFF} = (\theta_r - \theta_l)$;

end

Parallel do separate θ_r, θ_l;

Apply θ_r, θ_l using CRDC two-sided Rotation module;

Parallel find sine/cosine of θ_r, θ_l using CRDC processors

end

System Block Diagram : -

The top-level system block diagram of the SD processor based on CRDC algorithm is as shown in figure

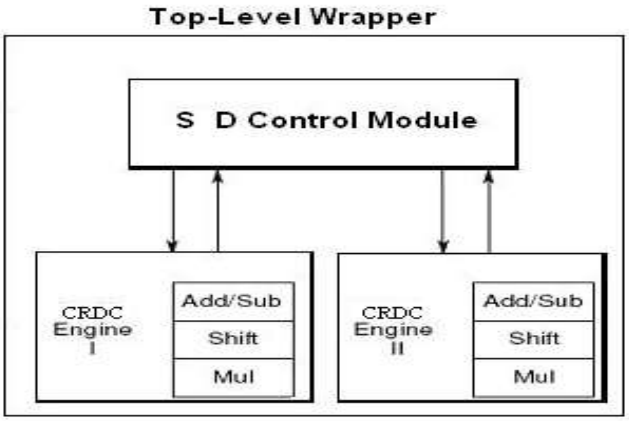


Figure 4.1: System Block Diagram

In system block diagram, there are four sub-blocks as listed below:

[1] CRDC Control
[2] CRDC Engine
[3] 32 bit Kegged –Stone adder
[4] Top level wrapper (System FSM)

First of all, Top level wrapper block has two states SYSFSM_IDLE and SYSFSM_BUSY which tells whether SD Control module is busy or not and if it can take any new task. When any new task arrives CRDC Control generates appropriate control signals for CRDC1 and CRDC2 engines. Now CRDC engines perform the task given to them by CRDC Control and after computation return the output data back to the CRDC Control block. Top level wrapper is also used to integrate all the units in one unit.

Simulation Results And Analysis

In this unit simulation results and synthesis report of each block is shown. Analysis of results has also been done. Design is written in Verilog HDL. Tools used for simulation and syntheses of the design are:
Compiler: Model Sim XE 6.2g
Synthesis: XILINX ISE 9.2i

Simulation results of Adder Block

Kogge – Stone adder is designed for 32-bit addition. In this block two 32-bit input data a _in and b _in are given along with the control signal add _n sub which determines whether to add (addendum=1) them or subtract (addendum=0). Sum is given as output. A simulation result is given in figure .

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Figure :- Simulation of Kogge -Stone adder block

Simulation result of CRDC Block

CRDC engine is the main data processing block. This block basically calculates sin/cos and inverse tan values when configured properly.

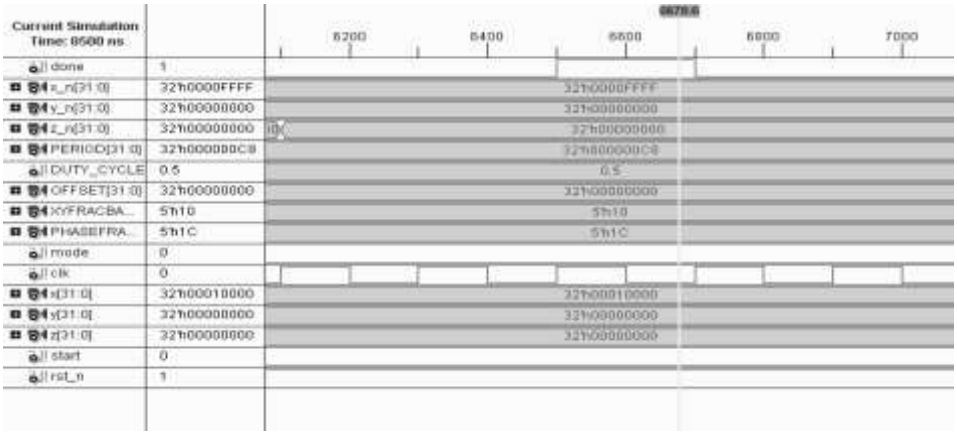


Figure: Simulation of CRDC engine for sin/cos of 0 degree

CONCLUSION AND FUTURE SCOPE

Conclusion

In linear algebra, the singular value decomposition (SD) is an important factorization of a rectangular real or complex matrix, with several applications in signal processing and statistics.

A 2X2 CRDC based SD processor was designed and successfully simulated. CRDC algorithm was used to simplify the computational complexities of SD calculation. Apart from calculating the SD of a 2x2 matrix the processor is able to calculate basic CRDC functions also that are sine/cosine, inverse tan

Achievements

1.Timing Analysis

To increase the frequency for the worst-case path, the following schemes were implemented.

2.Kogge-Stone Adder

Although the synthesis tool (Design Vision) supports the carry-look-ahead adder by default, a parallel- prefix tree adder “Kegged-Stone” was implemented.

Delay for a 32 bit Carry Look Ahead Adder: delay ≈ $2\log_2$ (n)

Delay for a 32 bit Kogge Stone Adder: delay ≈ log2 (n)

Where n is no. of bits.

This helps in reducing the delay through the critical path since the CRDC engine for the SD uses Adders at various stages.

FUTURE SCOPE

After the CRDC Control block,we can design APB (Advance Peripheral Bus) interface for enabling the processor to communicate with outside world. APB interface has 9 signals which are used to enter the input matrix data and the Control data. After calculations, output data are retrieved from their register locations. Thus, CRDC based SD

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

processor can be used as a coprocessor and this APB interface works as a bridge between main processor and coprocessor (SD processor).

REFERENCES

[1] Meggitt, J. E. (1962), "Pseudo Division and Pseudo Multiplication Processes", IBM Journal, pages 210-226
[2] Speiser J. M. and Whitehouse, H. J. (1983), "A Review of Signal Processing with Systolic Arrays", Proc. SPIE Real-Time Signal Processing, 431(VI) pp 2-6
[3] Ercegovac, M. D. and Lang, T. (1990), "Redundant and on-line CORDIC: application to matrix triangularization and SVD", IEEE Trans. Computers, vol. 39, pp. 725-74
[4] Götze, J. and Hekstra, G. (1995), "An algorithm and architecture based on orthonormal μ-rotations for computing the symmetric EVD", Integration VLSI Journal, vol. 20, pp. 21-39
[5] Cavallaro, J.R. and Luk, F.T. (1986), "Architectures for a CORDIC SVD processor", SPIE Conference on Real Time Signal Processing, vol. 698 pp 45-53
[6] Andraka, R. (1998), "A Survey of CORDIC Algorithms for FPGAs", in Proceedings of the 1998 ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays (FPGA '98), pp. 191-200.
[7] Walther, J. S. (1971), "A Unified Algorithm for Elementary Functions", AFIPS Spring Joint Computer Conf., pages 379-385
[8] Sibul, L. H. and Fogelsanger, A. L. (1984), "Application of Coordinate Rotation Algorithm to Singular Value Decomposition", IEEE Int. Symp. Circuits and Systems, pages 821-824.
[9] Speiser J. M. and Whitehouse, H. J. (1983), "A Review of Signal Processing with Systolic Arrays", Proc. SPIE Real-Time Signal Processing, 431(VI) pp 2-6
[10] Ercegovac, M. D. and Lang, T. (1990), "Redundant and on-line CORDIC: application to matrix triangularization and SVD", IEEE Trans. Computers, vol. 39, pp. 725-740
[11] Hemkumar, N. D. and Cavallaro, J. R. (1994), "Redundant and on-line CORDIC for unitary transformations", IEEE Trans. Computers, vol. 43, pp. 941-954
[12] Götze, J. and Hekstra, G. (1995), "An algorithm and architecture based on orthonormal μ-rotations for computing the symmetric EVD", Integration VLSI Journal, vol. 20, pp. 21-39
[13] Hekstra, G. J. and Deprettere, E. F. (1997), "Fast rotations: Low cost arithmetic methods for orthonormal rotation", Proc. 12th Symp. Comput. Arith, pp. 116-125[15] Walther, J. S. (1971), "A Unified Algorithm for Elementary Functions", AFIPS Spring Joint Computer Conf., pages 379-385[16] Kogge, P.M. and Stone, H.S. (1973), "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations", IEEE Trans. on Computers, Vol. 22.