International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 9 (2015) pp. 23199-23205

http://www.ripublication.com

Efficient Singular Value Decomposition Using Cordic For Signal Processing Applications

[1]K.Mangala Gowri
[1]School of computing, SASTRA University, Thanjavur-613401, India
[1]e-mail: gowri9991@gmail.com
[2]R.Manikandan
[2]School of computing, SASTRA University, Thanjavur-613401, India

Abstract

CORDIC algorithm being the calculator algorithm is even used today in many applications. In this paper the concept of CORDIC is exploited to find SVD (Singular Value Decomposition) which has tremendous applications in digital signal processing. CORDIC can be used to its fullest by decreasing the complexity of scaling and barrel-shifter used in the CORDIC engine. In this paper kogge-stone adder is used to increase the performance. The parallel diagonalization method calculates two rotating angles simultaneously which increases the speed. Rotation matrix is found at a time by finding the two angles θ_l and θ_r . Two CORDIC engines are used for this calculation.Hence a parallel processing method is used which increases the speed and reduces power. The overall efficiency is increased by this method.

Keywords: CORDIC- CO-ordinate Rotation Digital Computer, SVD, kogge-stone adder, parallel diagonalization.

Introduction

CORDIC

The algorithm was described by Jack. E. Volder in 1959. This algorithm was first used for basic trigonometric functions and then extended to other functions like logarithmic, hyperbolic, division and finding square root. CORDIC finds applications where the requirement of hardware is to be reduced as it only uses shift and add operation to perform complex calculations. The latency of this algorithm can be reduced by angle recoding methods [8]. The basic CORDIC hardware is shown in fig 1.The iteration equation and angle update equation are given below[7].

The two iteration equations are

$$x_{i+1} = x_i - d_i y_i 2^{-i} \quad (1)$$

$$y_{i+1} = y_i + d_i x_i 2^{-i} \quad (2)$$

The angle update equation is

$$z_{i+1} = z_i - d_i \phi_i \text{ , } d_i = +1 \text{ or } -1 \quad (3)$$

$$\phi_i = \tan^{-1}(2^{-i}). \quad (4)$$

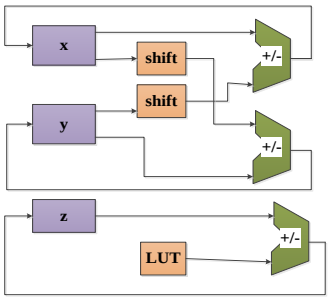

Figure 1: Basic CORDIC hardware

The method of micro-rotations performed in CORDIC:To obtain 60°:

$\Phi = 45^\circ, 45^\circ < 60^\circ$

$45^\circ + 26.565^\circ = 71.565^\circ, 71.565^\circ > 60^\circ$

$71.565^\circ - 14.036^\circ = 57.529^\circ, 57.529^\circ < 60^\circ$

$57.529^\circ + 3.576^\circ = 61.099^\circ, 61.099^\circ > 60^\circ$

$61.099^\circ - 1.7876^\circ = 59.31^\circ, 59.31^\circ < 60^\circ$

$59.31^\circ + 0.893^\circ = 60.2^\circ$.

This algorithm has two modesto work with.

1. Vectoring mode
2. Rotation mode.

Rotation of the input vector is by an angle theta in rotation mode. The final values of the x and y gives the trigonometric functions. The initial theta value is stored in the angle accumulator. Then the angle is rotated according to the value of $\tan^{-1}(2^{-i})$ [4]. The value of d_i is chosen so as it reduces the accumulation of angle in angle accumulator. The final value of z becomes zero in this algorithm. This mode is used to find the angular values i.e. rotation matrix can be found using this mode.

In vectoring mode the rotation angle is calculated. The y-component is made zero by rotating the vector towards x-axis. The result obtainedis the magnitude of vector scaled by K and the initial angle of the vector [4]. Here the y-component becomes 0, x gives the magnitude and z gives the angle.

SVD

Singular value decomposition is the extension of eigen value decomposition. Lots of applications are there for this singular values. The SVD of a matrix is generated by

orthogonal transformation which results in a diagonal matrix from which the singular values are obtained.

$$A=UEV^T \quad (5)$$

here

U , V - orthogonal matrices

E - diagonal matrix and the diagonal elements are called singular values [1].

Some of the areas which uses SVD is scientific computing, digital image processing, DSP (Digital Signal Processing). Particularly it is used in image and data compression, noise reduction, image de-blurring, automatic control. Kabsch algorithm uses SVD to compute optimal rotation which is used to compare structure of molecules. Its also used to find the pseudo inverse of a matrix, rank of a matrix and solve the least square problem.

The organization of the paper is as follows. Section 2 briefs the existing methods. Section 3 presents the description of proposed architecture. Section 4 discusses the results and the paper is concluded in section 5.

Existing Methods

Jacobi SVD calculation is used and a multiplier is used after each iteration [6].To eliminate a ROM a large barrel shifter is added. This leads to area and time overhead[3]. Optimal angles are used to reduce the time. Double rotation is performed to reduce the scaling factor. Hence the accuracy reduces [2]. Hestenes algorithm which is based on one sided Jacobi is used [10].

CORDIC algorithm can be implemented with different architecture forms iterative, parallel and pipelined. Every architecture has its own advantages and disadvantages. The parallel architecture can be changed into pipeline by adding registers in between as shown in fig 2.In pipelined architecture the shift registers have fixed shift and so the values are hardwired instead of keeping separate registers. Here the whole processor is changed into array of adders and subtractors. The latency is high for a pipelined structure [5]. It takes 16 clock cycles to get the first output for a 16-bit CORDIC. In this paper all architectures are tested and the best is chosen for implementing SVD. Parallel processing is good for finding SVD as the latency in pipelining leads to the gain compensation of the initial clock cycles.

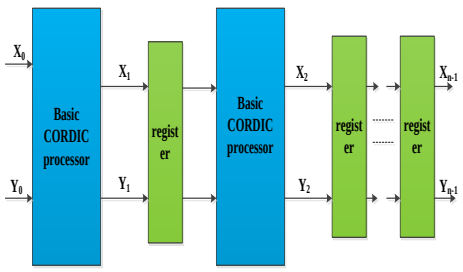

Figure 2: Pipelined architecture

Proposed Architecture

In the proposed architecture, the sin and cos of the angle and SVD of a matrix can be found. It consists of APB interface, SVD block and CORDIC engine as shown in fig 3. This architecture is compatible with APB (Advanced Peripheral Bus) .It is a part of AMBA (Advanced Microcontroller Bus Architecture). AMBA defines on-chip communication standards for high performance IP's. It interfaces to any peripheral which do not require high performance of pipelined architecture and which have low bandwidth. So APB is a un-pipelined protocol. The transitions are done in rising edge of the clock. The attractive features of APB are reduced interface complexity and low powerconsumption.

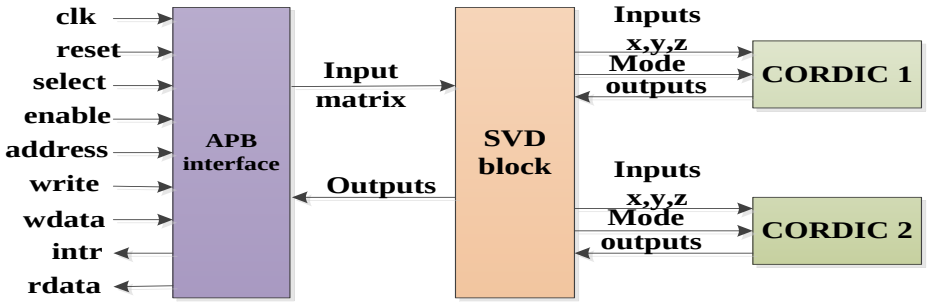

Figure 3: CORDIC SVD architecture

The signals for APB interface are clk which defines the system clock, intr which states the CORDIC process is completed. Select signal selects the IP if it is 1. Address specifies the address where the transition takes place. Enable is a signal to access the APB. It indicates a transition is going if it is 1. Wdata is an input signal where the data is written into the address. Rdata is an output signal. Write signal when 0/1, reads/writes the data respectively.

SVD Block

In the first stage theta sum and theta differenceare found using the formula,

$$\theta_s = \tan^{-1}\frac{n-k}{l+m} \quad (6)$$

$$\theta_d = \tan^{-1}\frac{n+k}{m-l} \quad (7)$$

k,l,m,n denotes the input values of the matrix. In the second stage rotate the vectors (a,b) and (c,d) by θ_r. in the third stage rotate the vectors (a,c) and (b,d) by θ_l. These two values are found in parallel diagonalization process. The third stage gives the eigen values. Finally calculate the sin and cos of θ_r, θ_l as shown in fig 4. The whole process is carried out using two CORDIC blocks.

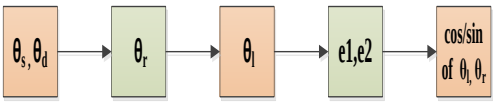

Figure 4: Stages of SVD block

Parallel Diagonalization Method

The input matrix values are given into arctan module. The theta sum and difference are calculated using the formula. Two cordic modules are needed to calculate the two angles simultaneously. The θ_r, θ_l values are found using the theta sum and difference.

$$\theta_r=\frac{\theta_s+\theta_d}{2} \quad (8)$$

$$\theta_l=\frac{\theta_s-\theta_d}{2} \quad (9)$$

This method increases the performance of the architecture. This method is depicted in fig 5.

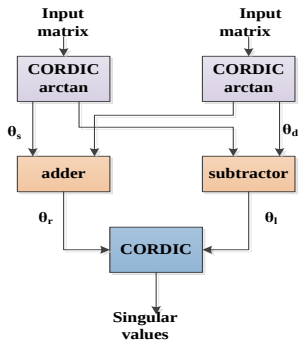

Figure 5: Parallel diagonalization

Kogge-Stone Adder

This adder is a version of carry look ahead adder. This architecture is widely considered as fastest and is used in industries for high performance. Fast computation is done at the cost of increased area. It has regular and low fan-out. The carry signals are generated in O(log n) time. The functionality of the adder is studied with three stages namely pre-processing, carry look ahead network and post-processing stage. Grey cell and blackcell are the two components. Black cell calculates both propagate and generate signals. Grey cell calculates only generate signal which is used in the final stage to calculate sum. This adder can be used even more efficiently by avoiding the redundant black cells because grey cells are only used in last stage [9].

The propagate and generate signals are generated in the first stage.

$$p_i = a_i \oplus b_i \quad (10)$$

$$g_i = a_i \square b_i \quad (11)$$

Group propagate and group generate signals are generated in second stage.

$$P_{i:j} = P_{i:l+1} \square P_{l:j} \tag{12}$$

$$G_{i:j} = G_{i:l+1} \mid (P_{i:l+1} \square G_{l:j}) \tag{13}$$

The last step is used to find the sum which is common to all carry look ahead adders.

$$S_i = p_i \oplus c_{i-1} \tag{14}$$

This architecture can be mapped with flexibly on FPGA.

Results

For the input matrix the singular valuesobtained using matlab andthe proposed system istabulated in table 1. Thearchitecture is simulated usingmodelsim and synthesis is carried out using synopsys. The area, power and timing report has been tabulated in table 2. The layout of the architecture is presented in fig 6.

Table 1: Singular values

Input matrix	Singular values	
	Matlab value	Proposed system value
$\begin{bmatrix} 2 & 6 \\ 5 & 8 \end{bmatrix}$	11.29 1.24	11.2893 1.2399

Table 2: Synthesis report

Area	Power	Time (minimum period)	Frequency (maximum frequency)
3648.2 um^2	Dynamic power = 586.6uW Cell leakage power = 17.3uW	20.7ns	48.29MHz

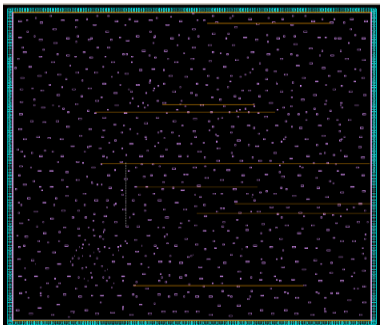

Figure 6: Layout of the proposed architecture

Conclusion

A CORDIC module to find SVD is developed here and results are presented. The kogge-stone adder and the parallel diagonalization method are used to increase the performance. It is found that area and power are optimized. The time period is also reduced. This IP core can be further developed to find different functions using CORDIC such as division, square root etc.

References

[1] Carla.D.Martin and Mason.A.Porter, "the extraordinary SVD", the mathematical association of America, pp 838-851, dec 2012.

[2] K.Dickson, Z.liu, "QRD and SVD processor design based on approximate angle rotation", ieee, SIPS 2004

[3] Leenavachani, K.Sridharan, "Efficient CORDIC Algorithms and Architectures for Low Area and High Throughput Implementation", ieee transactions on circuits and systems: express briefs, vol. 56, no. 1, january 2009

[4] Pramod K. Meher, Javier Valls, "50 Years of CORDIC: Algorithms, Architectures and Applications", ieee transactions on circuits and systems-2003.

[5] Ramesh bhakthavatsalu, sinith M.S," A comparison of pipelined parallel and iterative CORDIC design on FPGA", 5th ICIIS, pp 239-243, aug-2010.

[6] Ramanarayanmohanty, bibekkabi, "Design and performance analysis of jacobi SVD algorithm on reconfigurable systems", elsevier-2013.

[7] Ray andraka, "A survey of CORDIC algorithms for FPGA based computers", ACM,1998.

[8] Sukhpreet Kaur, Kulbirsingh,"implementation of high speed fixed point CORDIC techniques",international journal of computer applications, volume-56, no.3, pp 35-41, oct 2012.

[9] Sunil, Ankit, Manjunadha, Premananda, "design and implementation of faster parallel prefix kogge stone adder", volume 3,no.1, pp 116-121, jan2014.

[10] Tapanpradhan, Bibekkabi, Aurobindaroutray, Gonnabaktulaanirudh, "Fixed point hestenes SVD algorithm for computing eigen faces", IJOC, volume 7, pp312-321, 2013.