EFFICIENT WINDOW ARCHITECTURE DESIGN USING COMPLETELY SCALING FREE CORDIC PIPELINE

Archana C.B[1], Nagaraju C.[2]
Department of Electronics and Communication Engineering
BGS institute of Technology, B.G Nagar
Nagamangala Taluk, Mandya Dist. Karnataka, India.

Abstract: Window filtering is a well-known processing technique for limiting any signal to short time segment in various fields, like audio or video signal processing, communication systems etc. The Coordinate Rotation Digital Computer (CORDIC) is an arithmetic technique, which makes it possible to perform two dimensional rotations using simple hardware components. The algorithm can be used to evaluate elementary functions such as cosine, sine, arctangent, sinh, cosh, tanh, ln and exp. Here FPGA implementation of various window-functions using CORDIC algorithm by replacing linear CORDIC with multiple shift-add network and conventional circular CORDIC processor by a completely scaling-free CORDIC processor to improve the area-time efficiency.

Keywords: Window filtering, CSF, CORDIC.

I. INTRODUCTION

CORDIC (Coordinate Rotation DIgital Computer) is a very well-known algorithm that due to its versatility and very simple hardware implementation that only needs add and shift operations is widely used for VLSI digital signal processing systems. The conventional hardware implementation of window functions uses Look-Up Tables (for trigonometric computations) which gives rise to various area - time complexities, increase in word-lengths, and they do not allow user defined variations in the window-length. Realization of window functions using conventional CORDIC algorithm and Scaling-free CORDIC [5] and Enhanced scaling-free CORDIC algorithm [6] and in [4] a completely scaling-free CORDIC algorithm and it recursive architecture which has been suggested in previous implementations allows user-defined variations in window-length, but latency is the major problem of their design.

In this work, replacing conventional linear CORDIC processor with the multiple optimized shift-add network and conventional circular CORDIC with the completely scaling-free CORDIC processor reduce the pipelining depth and area of the existing system which minimizes the latency. Rest of this paper organized as follows: in section II deals with different window functions, conventional CORDIC algorithm and completely scaling-free CORDIC algorithm introduced. Section III details the proposed system. Section IV deals with the FPGA implementation and section V details the results and complexity issues. Section VI concludes the paper.

II. BACKGROUND

A window function is a mathematical function that is zero-valued outside of some chosen interval. When another function or waveform/data-sequence is multiplied by a window function, the product is also zero-valued outside the interval; all that is left is the part where they overlap. There are different windows [7] like Hanning, Hamming, Blackman, Kaizer etc. Selection of windows will be based on the requirement of the spectrum. Equation (1) shows the Hanning and Hamming windows.

$W_{Hann} = 0.5 + 0.5\cos(\frac{2\pi n}{N-1})$................... (1)

Where N is the length of the window

$W_{Hamm} = \alpha + \beta\cos(\frac{2\pi n}{N-1})$.................. (2)

Where $\alpha = 1 - \beta$

The values of α and β are determined to achieve maximum side-lobe cancellation. For Hamming window, the coefficients are calculated as α =25/46 and β =21/46.

A. Conventional CORDIC Algorithm

CORDIC Algorithm is an iterative algorithm for calculating the rotation of a two-dimensional vector in linear, circular and hyperbolic coordinate systems, using only add and shift operations. It consists of two operating modes, the rotation mode (RM), where a vector is $[x_A,y_A]$ rotated by an angle (θ) to obtain a new vector $[x_B,y_B]$ and the vectoring mode (VM) where the algorithm computes the length (r) and the angle (θ) towards the x-axis of a vector $[x_A,y_A]$. The coordinates of two vectors V_A $[x_A,y_A]$ and V_B $[x_B,y_B]$ separated by an angle 'θ' are related as:

$$\begin{bmatrix} x_B \\ y_B \end{bmatrix} = \mathrm{R} \cdot \begin{bmatrix} x_A \\ y_A \end{bmatrix}, \qquad \mathrm{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

……….. (1)

$$\theta = \sum_{i=0}^{b} \mu_i \cdot \alpha_i$$

Where $[x_B,y_B]$ is the final vector and θ is the target angle of rotation and is expressed as the summation of α_i = $\tan^{-1} 2^{-i}$ a decreasing sequence of elementary angles so that Where b is the word length and $\mu i \in \{1,-1\}$ is known as the direction of the vector rotation for ith iteration. The rotation matrix **R** in its original form (1) requires, determining the sine and cosine values and four multiplication operations. Factoring the cosine term, simplifies the rotation matrix by converting the multiplication operations to shift, as the tangent of elementary angles are defined in the negative powers of two (2).

$$\mathrm{R_i} = K_i \cdot \begin{bmatrix} 1 & -2^{-i} \\ 2^{-i} & 1 \end{bmatrix}, \quad K_i = \frac{1}{\sqrt{1+2^{-2i}}}$$

………..(2)

The rotation matrix **Ri** in (2) is applicable for anticlockwise vector rotations. To support both clockwise and anticlockwise CORDIC rotations, the rotation matrix is altered as:

$$\mathbf{R_i} = K_i \cdot \begin{bmatrix} 1 & -\mu_i \cdot 2^{-i} \\ \mu_i \cdot 2^{-i} & 1 \end{bmatrix}$$

In its original form, the CORDIC algorithm suffers from major disadvantages like, scale-factor compensation, latency and optimal identification of micro-rotations.

B. Conventional CORDIC Algorithm

In completely scaling-free CORDIC algorithm, the coordinate rotation-matrix is derived using the third order of approximation of Taylor series expansion of sine and cosine functions. To facilitate shift-add implementation of rotation matrix, the Taylor series coefficient 3! is approximated as 2^3.
For $\alpha_i = 2^{-s_i}$

$$\mathrm{R_i} = \begin{bmatrix} 1-2^{-(2s_i+1)} & -(2^{-s_i}-2^{-(3s_i+3)}) \\ 2^{-s_i}-2^{-(3s_i+3)} & 1-2^{-(2s_i+1)} \end{bmatrix}$$

The elementary angles are defined as

$$\alpha_i = 2^{-s_i}$$

The largest elementary angle α_{max} and the basic-shift (S_{basic}) depend on the order of Taylor series approximation used to realize the coordinate rotation matrix. For third order of approximation S_{basic} = 2 and α_{max} = 0.25 radians. Any rotation angle, is realized by using $n1$ iterations of α_{max} and $n2$ iterations corresponding to the other elementary angles, such that, the total number of iterations ($n1$ +$n2$) is always a constant.

$$\theta = n_1 \cdot \alpha_{max} + \sum^{n_2 \text{ iterations}} \alpha_{s_i}$$

Where $S_i > S_{basic}$ and n=n1+n2

A total of seven iterations are required to realize any rotation angle in the range [0, π/4], which is extended to the entire coordinate space using the octant wave symmetry of sine and cosine functions.

III. PROPOSED SYSTEM

The block diagram of proposed window-architecture using the pipelined architecture of the completely scaling free CORDIC algorithm to minimize area and latency of the existing window-architecture is shown in fig 1. The window-length can be selected by the user at the run-time. We have designed the window-architecture for 16-bit output-width.

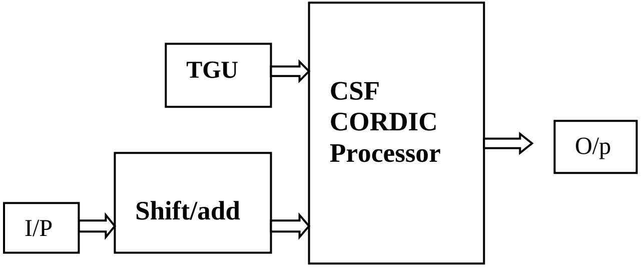

Fig.1: Block diagram of window architecture

The major blocks of the system are Shift/add network, circular CORDIC processor (CCP) and TGU. Shift/add network multiplies the input signal samples with the window constants CSF CORDIC is used for generating the cosine terms in the window function. The TGU divides the entire coordinate space into octants, so that the input angle to CCP always lies in the range [0, π/4]. The octants are distinguished as shown in Fig. 3, the TGU also generates signals for proper octant mapping of values generated by CSF CORDIC processor.

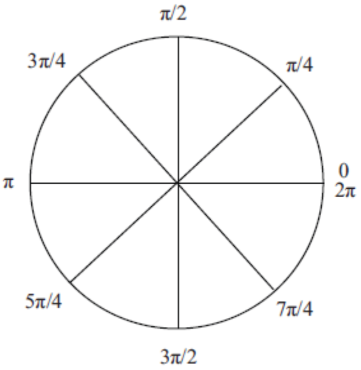

Fig. 2: Mapping of co-ordinate space into octant

A. Shift-add network

The Shift/add network multiplies the input samples with the window coefficients. The coefficients of the Hanning window are 0.5. In radix-2 representation system, multiplication with 0.5 is equivalent to single right shift. For Hamming window, the coefficient α is represented in 16-bit fixed-point format as 0010 0010 1000 0000 and the coefficient β is represented 0001 1101 1000 0000.

B. Circular CORDIC processor

The CCP is pipelined implementation of the completely scaling-free CORDIC algorithm of shown in Fig. 4. The complete pipeline is seven stages long. In the proposed work, processor identifies the octant in which the given angle is located, and then starts convergence of the same by taking maximum of seven iterations.

IV. IV FPGA IMPLEMENTATION OF THE PROPOSED SYSTEM

Table 1 summarizes the FPGA features used in this design.

TABLE I
FPGA FEATURES USED IN THE PROPOSED SYSTEM

Top level source type	HDL
Preferred language	Verilog
Family	Spatran-3
Device	XC3S400
Package	PQ208
Speed	-5
Simulator	Isim

V. RESULTS AND DISCUSSIONS

Area and Latency - The Xilinx CORDIC core is optimized for circular CORDIC computation with maximum pipelining for 16-bit word-length. The gate count is 20122. In , the complexity of 16-bit scaling free CORDIC is computed to be equivalent to 1000 1-bit full adders and 597 1-bit registers, which requires 16776 gates for implementation. The enhanced scaling-free (ESF) CORDIC 512 1-bit full adders and 420 1-bit registers, which is equal to 9504 gates. In the proposed system 266 slices, 501 4-i/p LUTs. The latency is related to number of iterations (or pipeline stages) in circular CORDIC and linear CORDIC processor. The 16-bit linear CORDIC processor uses 16 stages long pipeline. The conventional circular CORDIC processor again uses 16-stages pipeline for 16-bit word-length. For the same 16-bit word-length, the scaling-free CORDIC processor uses 12-stages long pipeline, while the ESF-CORDIC pipeline is 9 stages long. Therefore, the latency of the existing design in with conventional circular CORDIC is 32-stages, while with scaling-free is 28-stages and with ESF-CORDIC is 25-stages. The proposed CORDIC pipeline is 7-stages long. The delay of the optimized Shift/add which can be considered equivalent to network three linear CORDIC iterations. Hence, the total latency of the proposed design is 10-stages, which is far less as compared to existing design using the best of the available circular CORDIC hardware.

Delay: The delay is the time required to generate one set of window coefficients for a window-length of N, when the design is operating at the maximum clock frequency ($fclk$). The first sample is obtained after ($L \cdot f_{\mathrm{clk}}^{-1}$), where L is the latency of the design. The other ($N - 1$) samples due to pipelining are obtained after every clock cycle, implying a delay of [$(N - 1) \cdot f^{-1}{}_{\mathrm{clk}}$]. Therefore, the total delay D is given the maximum clock frequency for the existing design using ESF-CORDIC is 101.983 MHz and for other designs is 101.284 MHz. The pipelined architecture and the implementation of Carry Look-ahead adder which reduces the critical path delay. The proposed design uses 56.384MHz of clock frequency which gives better performance than the conventional CORDIC. Table II summarizes the proposed work.

TABLE III
SYNTHESIS REPORT SUMMARY OF THE CSF CORDIC

Area	266 slices, 501 LUTs
Latency	7 pipelining stages
Maximum clock frequency	56.384MHz
Delay	17.7ns
Type of Architecture	Pipelined
Window size	Variable

VI. CONCLUSION

In this paper we present an area-time efficient CORDIC based processor for computing various window-functions by saving the area, reduce the latency and optimizing the delay for maximum extent by replacing the linear CORDIC processor with multiple shift-add units and the circular conventional CORDIC processor with the completely scaling-free CORDIC processor**.**

REFERENCES

[1] K. C. Ray and A. S. Dhar, “Cordic-based unified vlsi architecture for implementing window functions for real time spectral analysis,” *IET Proc. Circuits Devices Sys.*, vol. 153, no. 6, pp.539–544, Dec. 2006.

[2] “High throughput vlsi architecture for blackman windowing in real time spectral analysis,” *Journal of Computers*, vol. 3, no. 5, pp. 54–59, May 2008.

[3] P. K. Meher *et al.*, “50 years of cordic: Algorithms, architectures and applications,” *IEEE Trans. Circuits Syst. I*, vol. 56, no. 9, pp. 1893–1907, Sep. 2009.

[4] S. Aggarwal, P. K. Meher, and K. Khare, “Area-time efficient Scaling free CORDIC using generalized micro-rotation selection,” *IEEE Trans. VLSI Syst.*, vol. 20, no. 8, pp. 1542– 1546, 2012.

[5] K. Maharatna *et al.*, “Modified virtually scaling-free adaptive cordic rotator algorithm and architecture,” *IEEE Trans. Circuits Syst. Video Technol.*, vol 11, no. 11, pp. 1463–1474, Nov. 2005.

[6] F. Jaime *et al.*, “Enhanced scaling-free cordic,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 7, pp. 1654–1662, Jul. 2010.

[7] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing Principles, Algorithms and Applications*. Prentice Hall Third Edition, 2006.

[8] “HDL – programming (VHDL & Verilog)” Author- Nazeih M Botros

[9] “Digital Design – An Embedded Systems Approach Using Verilog” Author- Peter J. Ashenden