

Review paper on Design of DFT using CORDIC algorithm

Mr. Ashwin D. Baragawane¹, Mrs. P. Gajbhiye²

¹M. tech 4th semester, E & EC Department, KITS, Raamtek-441106

ashwinbaragawne1610@gmail.com

²Associate Professor, E & EC Department, KITS, Ramtek-441106

pradnyag_ramteke@rediffmail.com

ABSTRACT

CORDIC is an algorithm developed by Jack E. Volder in the year 1959 which is an acronym for COordinate Rotation Digital Computer. It is a class of shift, adds algorithm for rotating vectors in a plane, generally used for the computation of trigonometric functions, multiplications, division and conversion between binary and mixed radix number system, such as Fourier Transform. The Jack E. CORDIC algorithm is resulting from the general equation for vector rotation. Cordic algorithm which is highly efficient, having low complexity and robust technique to calculate exponential function. It also contains the implementation of Discrete Fourier Transform using radix-2 decimation-in-time algorithm on the FPGA kit. Due to the high speed, low cost and greater flexibility offered by FPGAs over DSP processors, the FPGA based computing is becoming the heart of all digital signal processing systems of recent world. Furthermore the generation of test bench by Xilinx ISE 9.2i verifies the results.

Keywords: Discrete Fourier Transform (DFT), CORDIC Algorithm (CA), VHDL, Sine, Cosine, FPGA (Field Programmable Gate Array).

INTRODUCTION

The CORDIC algorithm was first introduced by Jack E. Volder [1] in the year 1959 for the computation of Trigonometric functions, Multiplication, Division, Data type conversion, Square Root and Logarithms. It is a highly efficient, low-complexity, and robust technique to compute the elementary functions. The basic Algorithm structure is described in [2] [3]. The CORDIC algorithm has establish its way in various applications such as pocket calculators, numerical co-processors, to high performance radar signal processing, supersonic bomber aircraft with a digital counterpart. Bekooij, Huisken, and Nowak research tells about the application of CORDIC in the calculation of the

(Fast Fourier Transform) FFT, and at the effects on the numerical precision. It calculates the value of trigonometric functions like sine, cosine, magnitude and phase (arctangent) to any desired accuracy. It can also estimate hyperbolic functions (such as sinh, cosh, and tanh).

TheCORDIC algorithm does not use Calculus based methods such as polynomial or rational function approximation. It is used as approximation function values on all popular graphic calculators, including HP-

48G as the hardware restriction of calculators require that the elementary functions should be computed using only additions, subtractions, digit shifts, comparisons and stored constants. Today cordic algorithm is used in Neural Network VLSI design [4], high performance vector rotation DSP applications, advanced circuit design, optimized low power

Design [5]. CORDIC algorithm revolves around the idea of "rotating" the phase of a complex number, by multiplying it by a sequence of constant values. However, the "multiplies" can all be powers of 2, so in binary arithmetic they can be done using just shifts and adds; no actual "multiplier" is needed thus it simpler and do not require complex hardware structure as in the case of multiplier. Previous methods used are Table look up method, Polynomial approximation method etc. for estimation of trigonometric functions. It is hardware efficient algorithm. No multiplier requirement as in the case of microcontroller.

II. LITERATURE SURVEY

Large numbers of architectures have been proposed in the literature for CORDIC algorithm, which differ from bitserial implementations to word parallel pipelined architectures. The selection depends on the requirements

for computing throughput and restrictions that hold for area usage, latency and power dissipation. Basically [1] [6], implementations of the CORDIC algorithm have been carried out on word serial architectures by means of conventional non-redundant arithmetic with radix-2 microrotations and fixed point internal design. 8-point DFT using CORDIC algorithm has been implemented earlier. Radix-2, Radix-4 DFT FFT has been implemented in earlier paper[7]. In this paper we are implementing the 16-point DFT using CORDIC algorithm. There are two types of fourier transform. These are Discrete Fourier Transform and Continuous Fourier Transform. The Fourier transform when applied to digital (Discrete) rather than an analog (continuous) signal is called as Discrete Fourier Transform.

III. BACKGROUND/ HISTORY

CORDIC algorithm has set up its way in many applications. The CORDIC was introduced in 1959 by Jack Volder as a highly efficient, low-complexity, and robust technique to compute the elementary functions. It is originally intended for navigation technology, the CORDIC algorithm has create its way in a wide range of applications. After discovery CORDIC worked as the substitute for the analog navigation computers aboard the B-58 supersonic bomber aircraft with a digital counterpart. The CORDIC airborne navigational computer built for this reason, outperformed conventional contemporary computers by a factor of 7, mainly due to the revolutionary progress of the CORDIC algorithm. Further Steve Walther [6] continues work on CORDIC, with the application of the CORDIC algorithm in the Hewlett-Packard calculators, such as the HP-9100 and the well-known HP-35 in year 1972, the HP-41C in year 1980. He told how the integrated CORDIC algorithm i.e. combining rotations in the circular, hyperbolic, and linear coordinate systems and how it was useful in the HP-2116 floating-point numerical co-processor. Today's fast rotation techniques are intimately connected to CORDIC, to achieve orthonormal rotation at a very low cost. Even though the fast rotations exist for certain angles only, they are adequately flexible, and have already been broadly applied in signal processing. Hekstra found a large range of known, and previously unknown, fast rotation methods. An in general estimate of the methods exposes the trade-offs that exist between the angle of rotation, the precision in scaling and the cost of rotation. Van der Kolk, Deprettere, and Lee [8] formalized the difficulty of (approximate) vectoring for fast rotations in year 2000. They treated the quick and efficient selection of the appropriate fast rotation, and give you an idea about the advantage to be gained when applied to

Enhanced Versatile Disc (EVD). The selection technique works similarly well for redundant arithmetic and floatingpoint computations. Antelo, Lang, and Bruguera[9] considers going to a higher radix than the radix-2 for the standard algorithm, so that less iterations are required. The selection of a higher radix implies that the scaling factor is no longer constant. The authors suggest an on-line calculation of the logarithm of the scale factor and successive compensation. Hsiao, Lau, and Delosme [10] considered multi-dimensional alternatives of CORDIC, such as the 4-D (dimension) familiar CORDIC transform, and their application to singular value deposition (SVD). Rather than building a multi-dimensional transform out of a series of 2-D (dimension) CORDIC operations, they proposed to work with multi-dimensional micro-rotations, instantly at the iteration level. Their method is evaluated and benchmarked against solutions by others. Kwak, Choi, and Swartzlander [11] aimed to overcome the critical path in the iteration through sign prediction and addition. They proposed to overlap the sign prediction with the addition, by computing the results for both outcomes of the sign, and to select the proper one at the very end of the iteration. Story in their approach is to join the adder logic for the computation of both results.

A. Comparision between CORDIC algorithm and other Algorithm Before the generation of CORDIC algorithm there are table look up method, polynomial approximation method etc. for evaluation of trigonometric functions were generated. But these are the hardware efficient algorithm, whereas the CORDIC algorithm is the hardware inefficient algorithm. CORDIC algorithm is generally faster than the other approaches when a hardware multiplier is unavailable. On the other hand, when the hardware multiplier is available, table look up method and power series are generally faster than CORDIC algorithm.

B. Advantages

- 1)** CORDIC algorithm developed by Jack E. Volder who proposed that, this algorithm is highly efficient, lowcomplexity to calculate the elementary functions than the other algorithm.
- 2)** Furthermore the Steve Walther continues the work on CORDIC, he told the application of CORDIC that is combining the rotation in circular, hyperbolic and linear coordinate systems and how it was applied in floating point numerical co-processor.
- 3)** Van der Kolk, Deprettere, and Lee has concerned with the difficulty of vectoring for fast rotation. He then remove the problem of vectoring for fast rotation.

4) Antelo, Lang, and Bruguera again redeveloped the CORDIC algorithm for less number of iteration. Because the earlier CORDIC algorithm was developed for large number of iteration.

5) Hardware requirement and cost of CORDIC processor is less as only shift registers, adders and look-up table (ROM) are required.

C. Disadvantages

1) Jack E. Volder’s developed the CORDIC algorithm for large number of iteration. Large number of iteration required for accurate results and thus the speed is slow and the time delay is high.

2) Power consumption in some type of architecture are high.

3) As the Hardware multiplier is available then the table look-up method and power series methods are generally quicker than the CORDIC algorithm. The CORDIC rotator is usually operated in one of two modes, i.e., the rotation mode and the vectoring mode[13].

D. Rotation mode

The first mode of operation, called rotation by Volder, rotates the input vector by a particular angle(given as argument). Here, the angle accumulator is initialized with the preferred rotation angle. The rotation decision based on the sign of the residual angle is made to reduce the magnitude of the residual angle in the angle accumulator. If the input angle is previously expressed in the binary arctangent base, the angle accumulator is not needed

E. Vectoring mode

In the vectoring mode, the CORDIC rotator rotates the input vector through whatever angle is necessary to line up the result vector with the x-axis. The result of the vectoring operation is a rotation angle and the scaled magnitude i.e. the x-component of the original vector. The vectoring function works by seeking to minimize the y component of the residual vector at each rotation. The sign of the remaining y component is used to determine which direction to rotate next. When initialized with zero, accumulator contains the traversed angle at the end of the iterations.

IV. DISCRETE FOURIER TRANSFORM (DFT)

A. CORDIC for DFT calculation

In this paper it has discussed that how sine and cosine can be calculated using CORDIC algorithm and now using this algorithm how Discrete Fourier Transform (DFT) can be calculated has been discussed later in this paper. A Fourier transform is a particular case of a wavelet transform with source vectors defined by trigonometric functions sine and cosine. There are quite a few ways to calculate the Discrete

Fourier Transform (DFT), such as solving simultaneous linear equations or the correlation method. The Fast Fourier Transform (FFT) is another method for calculating the DFT. While it produces the same result as the other approaches, it is incredibly more efficient, often reducing the computation time by hundreds. A Fast Fourier transform (FFT) is an efficient algorithm to compute the Discrete Fourier transform (DFT) and it’s inverse.

B. FFT method for DFT calculation

The Fast Fourier Transform (FFT) (Figure 1. Eight point decimation-in-time FFT algorithm) is another method for calculating the DFT. Fast Fourier transform (FFT) is an efficient algorithm for computing the discrete Fourier transform. The discovery of the FFT algorithm covered the way for general use of digital methods of spectrum estimation which influenced the research in almost every field of engineering and science. DFT is a tool to estimate the samples of the CFT at uniformly spaced frequencies. It requires less multiplication than a simple approach of calculating DFT [12]. The basic computation using FFT is called butterfly computation which s shown in figure 1.

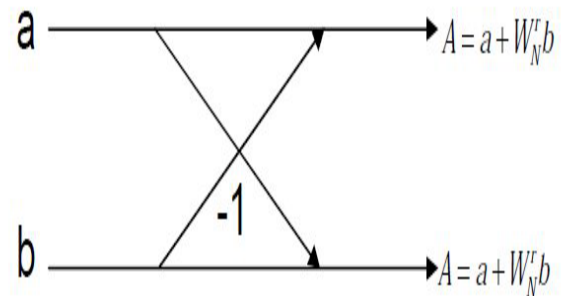


Figure 1: Basic butterfly computation in the decimation-in time.

Using the above butterfly computation technique an 8*8 or 16*16 DFT can be calculated.

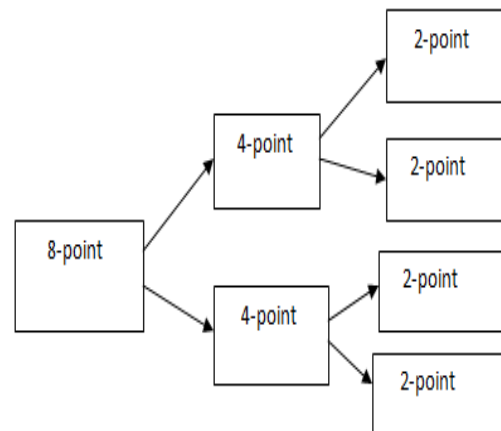


Figure 2: 8-point Fast Fourier Transform

Block diagram for FFT from figure 2 which tells about the different stages to calculate DFT of a sampled signal.

V. RESULTS AND CONCLUSION

The CORDIC algorithm is highly efficient, low complexity and tough technique to calculate the trigonometric function. Earlier CORDIC algorithm was developed for large number of iteration and as the iterations are increases, the hardware multiplier also increases. For this purpose the Antelo, Lang, and Bruguera again redeveloped the CORDIC algorithm for less number of iteration. This will gives the faster response and the requirement of hardware multiplier is less. The design is more efficient and consumes less resources and is less time intensive.

VI. REFERENCES

1. Volder J. E., .The CORDIC trigonometric computing technique., IRE Trans. Electronic Computing, Volume EC-8, pp 330 - 334, 1959.
2. Lindlbauer N. www. cnmat. berkeley. Edu / ~norbert/ CORDIC /node3.htmlhtt.
3. Avion J.C., <http://www.argo.es/~jcea/artic/CORDIC.htm>
4. Qian M., .Application of CORDIC Algorithm to Neural Networks VLSI Design., IMACS Multiconference on .Computational Engineering in Systems Applications (CESA)., Beijing, China, October 4-6, 2006.
5. Lin C. H. and Wu A. Y., .Mixed-Scaling-Rotation CORDIC (MSR-CORDIC) Algorithm and Architecture for High-PerformanceVector Rotational DSP Applications., Volume 52, pp 2385- 2398, November 2005
6. Walther J.S.A, .Unified algorithm for elementary functions., Spring Joint Computer Conference, pp 379 - 385, Alantic city, 1971.
7. Ajay S. Padekar, S. S. Belsare, "Study of a Cordic based Radix-4 FFT Processor", International Journal of Electrical, Electronics and Data Communication, ISSN: 2320-2084
8. Kolk K. J. V., Deprettere E.F.A. and Lee J. A., . A Floating Point Vectoring Algorithm Based on Fast Rotations., Journal of VLSI Signal Processing, Volume25, pp 125.139, Kluwer Academic Publishers, Netherlands, 2000.
9. Antelo E., LangT. and Bruguera J. D., .Very-High Radix CORDIC Rotation Based on Selection by Rounding., Journal of VLSI Signal Processing, Vol.25, 141.153, Kluwer Academic Publishers, Netherlands, 2000.
10. Delosme M. J.,Lau C. Y. and Hsiao S. F., .Redundant Constant-Factor Implementation of Multi-Dimensional CORDIC and Its Application to Complex SVD., Journal of VLSI Signal Processing, Volume 25, pp 155.166, Kluwer Academic Publishers, Netherlands, 2000.
11. Choi J. H., Kwak J. H. and Swartzlander, Journal of VLSI Signal Processing, Volume 25, Kluwer Academic Publishers, Netherlands, 2000.
12. Proakis J. G.,Manolakis D. G., .Digital signal processing principles, algorithms and applications., Prentice Hall, Delhi, 2008.
13. Srinivasa Murthy H N, Roopa M,"FPGA Implementation of Sine and Cosine Generators using CORDIC Algorithm", in International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-1, Issue-6, November 2012
14. P.K. Meher, J. Valls, T.B. Juan, K. Sridharan and K. Maharatna, "50 Years of CORDIC: Algorithms, Architectures, and Applications", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS