AMSE JOURNALS-AMSE IIETA publication-2017-Series: Advances B; Vol. 60; N°2; pp 331-336
Submitted Apr. 04, 2017; Revised Sept. 25, 2017; Accepted Sept. 30, 2017

FPGA Implementation of Discrete Fourier Transform Using CORDIC Algorithm

*Debaprasad De, **K. Gaurav Kumar, **Archisman Ghosh, **Anurup Saha

*Department of ECE, Techno India, Salt Lake,
Kolkata, India (dpdatju@gmail.com)
**Department of ETCE, Jadavpur University, Kolkata, India
(kgauravkumar35@gmail.com, archismanghosh12@gmail.com, sahaanurup24@gmail.com)

Abstract

Discrete Fourier Transform (DFT) is a very useful algorithms, playing an important role in various Digital Signal Processing (DSP) applications from sonar, image processing, telecommunication, radar, etc. This paper presents architecture for computing DFT of discrete time sequences using the CORDIC algorithm. The twiddle factors, i.e. the phase rotation factors, required in DFT computations are calculated by CORDIC algorithm. Moreover, by utilizing some trigonometric identities in the DFT calculation CORDIC rotators are effectively used. The proposed architecture can be reconfigured to calculate DFT for any point discrete time sequence.

Key words

CORDIC, DFT, FPGA, ASM, Architecture.

1. Introduction

Discrete Fourier Transform (DFT) is an extensively used algorithm in the analysis and implementation of discrete time signal processing, image processing, data compression, communication systems, etc. The complex sequences in time domain are converted into frequency domain via DFT computation and frequency domain sequences are converted back to time domain by Inverse Discrete Fourier Transform (IDFT) [1].

The COordinate Rotation Digital Computer (CORDIC) algorithm [2][3][4] is employed for calculating a vector rotated through a given angle, i.e. twiddle factors for computation of DFT.

Original algorithm is given by Volder [2]. Nowadays it is used for wide ranges of computer hardware aspects. This algorithm has found to a broader way into diverse applications including the HP-35 calculator, 8087 math coprocessors [5], radar signal processors [6] and robotics.

CORDIC algorithm is suitable to be implemented in DSP algorithms because complex arithmetic operations can be simply calculated. Besides, since it avoids using multiplications, adopting the CORDIC algorithm can reduce the complexity. CORDIC is implemented through repeated shift add operations.

Discrete Fourier transform coefficients are mainly complex numbers with sine and cosine terms. Thus, complex multiplications and additions are to be done for both real and imaginary parts of the input sample. If an N -point DFT is implemented, the requirement of arithmetic operations is of the order of $O(N^2)$ that is N^2 multiplications and N (N-1) additions and 2N number of registers are used to store the DFT coefficients [7].

In this work, architecture for DFT is presented, which has been implemented on Xilinx Spartan 3E FPGA. The hardware descriptions of our architectures are coded in Verilog HDL. The input sequence samples are taken serially from ROM.

The structure of rest of the paper is as follows. We discuss about DFT and CORDIC algorithms in Section 2. The proposed architecture and implementation methodologies are presented in Section 3. FPGA implementation details and simulation results are provided in Section 4. Conclusions are drawn in Section 5.

2. Background

In this section, we discuss about DFT and CORDIC algorithms.

2.1 DFT

DFT and IDFT techniques [1] [7] are used to transform signals from time domain to frequency domain and vice versa. They origin from the corresponding continuous time Fourier and Inverse Fourier transforms. Given a sequence x(n) for n = 0, 1…, N-1, its DFT is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi kn}{N}}, for\ k = 0,1, \dots, N-1$$

$$= \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad (1)$$

Where, W_N is known as twiddle factor and X (k) is the frequency sample corresponding to input samples.

The Inverse of the DFT is given by

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\, e^{\frac{j2\pi kn}{N}}, for\ n = 0,1,\dots,N-1$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X(k) W_N^{-kn} \quad (2)$$

The twiddle factor, W_N is given by

$$W_N = e^{\frac{-j2\pi}{N}} = \cos\left(\frac{2\pi}{N}\right) - j\sin\left(\frac{2\pi}{N}\right) \quad (3)$$

To calculate DFT and IDFT, the values of twiddle factors (i.e., the cosine and sine values) are needed to be calculated. For this, CORDIC algorithm has been used.

2.2 CORDIC

CORDIC [2] [3] [4] was developed by Volder in 1949 based on the observation that if a unit-length vector is rotated by an angle φ then its new end-point will be at (cos φ, sin φ).

The algorithm can be derived from the rotation transform, i.e. if a point (x, y) is rotated by an angle φ, the new coordinates (x', y') is given by equations (4) and (5).

$$x' = x\cos\phi - y.\sin\phi \quad (4)$$

$$y' = y.\cos\phi + x.\sin\phi \quad (5)$$

On rearranging the terms, this can be given as:

$$x' = \cos\phi\,[\,x - y.\tan\phi\,] \quad (6)$$

$$y' = \cos\phi\,[\,y + x.\tan\phi\,] \quad (7)$$

The implementation of these equations still seems complex due to the presence of the trigonometric functions. However, if the rotation angles are restricted to values such that $\tan\phi = \pm 2^{-i}$, the multiplication by the tangent can be greatly simplified as it can be implemented using

simple shift operations. Thus, arbitrary angles can be obtained by performing a series of rotations iteratively. At each rotation, the direction of rotation is chosen by obtaining the difference between the actual angle and the angle obtained by rotation. By choosing a proper sequence of rotations α_0, α_1..., α_{n-1}, we can evaluate necessary trigonometric functions with ease and less complex operations.

3. Proposed Architectures

Algorithmic State Machine (ASM) for DFT is presented in this section. This ASM is then coded in Verilog, implemented on FPGA and finally performance is analyzed.

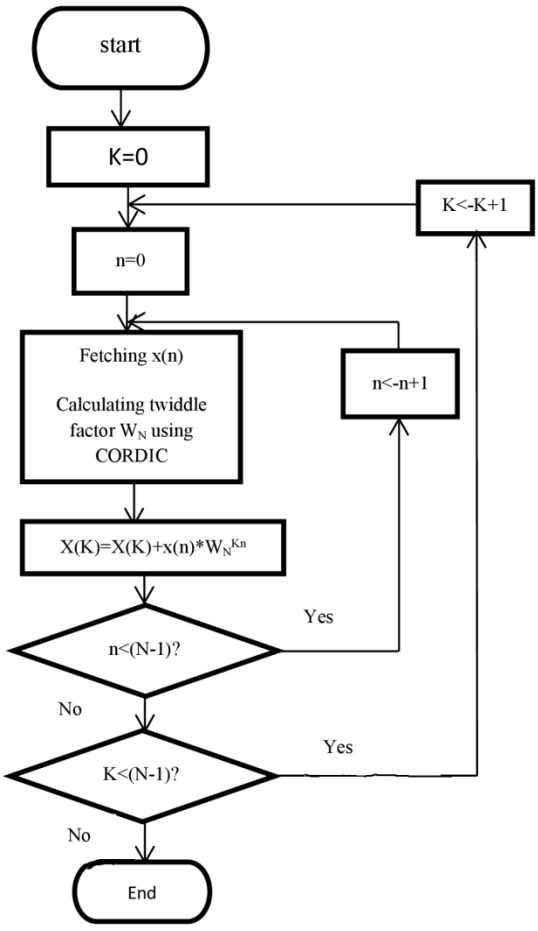

Fig.1. ASM for DFT

Similarly, ASM for IDFT can be implemented and realized in a similar manner [8].

4. FPGA Implementation

The ASM, shown in Fig. 1, is implemented on Spartan 3E FPGA. The realized hardware are clocked with 50 MHz clock. The timing waveform report and the device utilization summary are

provided in Table 1 and Table 2 respectively. The performance of the hardware is tested for 10, 12 and 20-point DFT.

Tab.1. Timing Waveform Analysis (Clock Cycle = 20 ns)

Number of input sequences	Transform Type	Number of clock cycles	Time taken (in ns)
10	DFT	24179	483580
12		28999	579980
20		96509	1930180

Table 2 describes the device utilization summary and the hardware resources needed for the proposed architecture.

Tab.2. Device Utilization Summary

No. of input sequences	Transform type	Slice Registers	Slice LUTs	Fully used LUT-FF pairs	Bonded IOBs	No. of BRAMs	MULT18XSIOs	GCLKs
10	DFT	655	616	1232	34	0	4	1
12		682	648	1285	34	0	4	1
20		798	776	1505	34	1	4	1

Conclusion

Architecture for DFT has been presented in this paper. It has been observed that as the number of N-point samples increase, the time and hardware requirements of the system increase. Faster algorithms like FFT can solve this problem. Pipelining and Systolic arrays can also be incorporated into the system to improve the throughput and speed of operation. Application Specific Integrated Circuit (ASIC) can be implemented which can be used in a number of DSP applications.

Acknowledgement

The authors would like to thank Prof. M.K. Naskar, Dr. S.K. Mitra and Asim Maiti for their helpful comments at the time of preparing the manuscript.

References

1. J.G. Proakis, D.G. Manolakis, Digital signal processing, principles, algorithms and applications, Prentice Hall India Publication, pp. 459-462.
2. J.E. Volder, The CORDIC trigonometric computing technique, 1959, IRE Transactions on Electronic Computers, no. 3, pp. 330-334.
3. R. Andraka, A survey of CORDIC algorithms for FPGA based computers, February1998, Proc. of the 1998 CM/SIGDA Sixth International Symposium on FPGAs, Monterey, CA, pp. 191-200.
4. B. Parhami, Computer Arithmetic, Oxford University Press, pp. 361-371.
5. J. Duprat, J.M. Muller, The CORDIC Algorithm: New results for fast VLSI Implementation, 1993, IEEE Transactions on Computers, Vol. 42, pp. 168-178.
6. R.J. Andraka, Building a high performance bit serial processor in an FPGA, Jan 1996, Proceedings of Design Super Con '96, pp. 5.1-5.21
7. A.V. Oppenheim, Digital signal processing, 1975, Prentice Hall, pp. 541-582.
8. P.P. Chu, FPGA prototyping by Verilog examples, John Wiley & Sons, pp. 139-141.