ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2015

Pipelined CORDIC Architecture for FFT Processor Implementation on FPGA

Prof. (Dr.) P. Malathi[1], Dr. Manish Sharma[2], Chetan Korde[3]

Professor, Dept. of E & TC, D. Y. Patil College of Engineering, Akurdi, Pune, India[1]

PG Co-ordinator, Dept. of E & TC, D. Y. Patil College of Engineering, Akurdi, Pune, India[2]

PG Student [VLSI], Dept. of E & TC, D. Y. Patil College of Engineering, Akurdi, Pune, India[3]

ABSTRACT: This proposed work is the study of an efficient CORDIC algorithm for FFT processor implementation on FPGA. Due to use of Radix-4 speed get increases than Radix-2 in FFT computation. For twiddle factor calculation Co-ordinate Rotation Digital Computer (CORDIC) algorithm is used, which help to reduce the computation time and make processor faster. The CORDIC provides the opportunity to calculate all the required functions in a rather simple and elegant fashion. In the next phase of this paper, actual Implementation of FFT processor on FPGA will be done using VHDL.

KEYWORDS:FFT, Radix-4, Radix-2, CORDIC, VHDL, FPGA.

I.INTRODUCTION

The digital signal processing has been dominated by microprocessors with enhancements such as special addressing modes and single cycle multiply-accumulate instructions. While these processors offer extreme flexibility with low cost, they are often not fast enough for most of Digital Signal Processing (DSP) tasks. The arrival of reconfigurable logic computers offers hardware solutions for higher speed at cost which are less than traditional software approach.[4] Unfortunately, algorithms used for these microprocessor based systems cannot be mapped into hardware. In such hardware efficient algorithms, there is a class of solutions for trigonometric functions that uses shifts and additions in its operation. The trigonometric functions are found out using vector rotations, while other functions like square root are realized using an incremental expression. So, implementation of all these function in to hardware is difficult.

Discrete Fourier Transform (DFT) is one of the core operations in digital signal processing and communication systems. Many fundamentalalgorithms can be realized by DFT, such as convolution, spectrum estimation, and correlation. Furthermore, DFT is widely used in standard embedded system applications such as wireless communication protocols requiring Orthogonal Frequency Division Multiplexing, Radar image processing using Synthetic Aperture Radar and Software Defined Radio etc. However, DFT is difficult to implement directly due to its computational complexity. [8]

In practice, Fast Fourier transform (FFT) is used for reducing the complexity of computations. For FFT processors, butterfly operation is the most computationally demanding stage. Traditionally, a butterfly unit is composed of complex adders and multipliers and the multiplier is usually the speedup bottleneck in the pipeline of the FFT processor. The CORDIC algorithm is an alternative method to realize the butterfly operation without using any dedicated multiplier hardware. The trigonometric algorithm is also called as CORDIC. The CORDIC algorithm is versatile and hardware efficient since it requires only add and shift operations, making it suitable for the butterfly operations in FFT.[8] This CORDIC algorithm is further used in Radix-4 FFT for faster computation. While converting samples taken in real time into equivalent frequency domain samples, FFT computation is used in which twiddle factor computation is done using CORDIC algorithm. Instead of storing actual twiddle factors in a ROM, the CORDIC-based FFT processor needs to store only the twiddle factor angles in a ROM for the butterfly operation.

Conventionally, a CORDIC-based FFT processor needs a dedicated memory bank to store the necessary twiddle factor angles for the rotation. This study proposes a modified CORDIC algorithm for FFT processors which eliminates the

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2015

need for storing the twiddle factor angles. The algorithm generates the angles successively by an accumulator. With this approach, memory requirements of an FFT processor can be reduced by more than 20%. Memory reduction improves with the increased radix size. [1] Furthermore, the angle generation circuit consumes less power consumption than angle memory accesses. Hence, system throughput does not change by using modified CORDIC angle calculation.

II. RELATED WORK

The digit-by-digit methods for the computationof the abovementioned elementary functions were described by Henry Briggs in 1624 in “ArithmeticaLogarithmica” [11, 12]. These methods are iterative pseudo division and pseudo multiplication processes, which resemble repeated-addition multiplication and repeated-subtraction division. In 1959, Volder has proposed a special purpose digital computing unit known as COordinate Rotation DIgital Computer (CORDIC), while building a real time navigational computer for use in an aircraft [7, 13]. This algorithm was initially developed for trigonometric functions which were expressed in terms of basic plane rotations.Also CORDIC based Radix 2 FFT was implementing by using Folding transformations. To design FFTarchitectures with reduced number of functional units. In the folding transformation, many butterflies in the same column can be mapped to one butterfly unit. The FFT block is designed to be capable of computing 8 point FFT and employsRadix 2 architecture which is simple, elegant and best suited for communication applications. [15]

III.RADIX-4 FFT ALGORITHM

In this section the procedure of Radix-4 FFT algorithm is given.Figure (1) shows an example of Radix-4 decimation in time method used for N=16 points FFT algorithm. As shown in FFT flow-graph inputs are in normal order while the outputs are in digit-reversed order. At input side the samples are taken from time domain which are processed with Radix-4 FFT and get equivalent components in frequency domain. The numbers over flow lines indicates the twiddle factor to be multiplied with the samples.

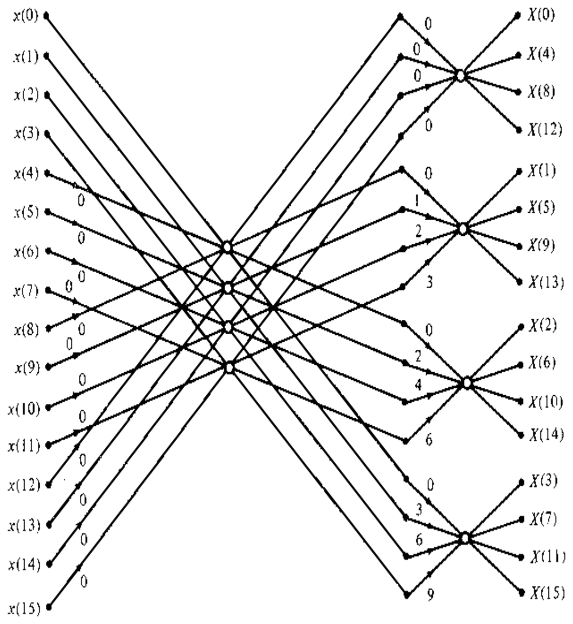

Fig. 116-point radix-4 FFT DIT algorithmwith input in normal order and output in digit-reversed order

Figure 2 (a) and (b) shows the basic butterfly structure of Radix-4 which have four inputs and four outputs, inputs are as x(n), x(n + n/4), x(n + n/2) and x(n +3n/4) outputs are in digit reversed order X(k). A Radix-r FFT uses N/r Radix-r butterflies for each stage and has [log $_r$ (N)] stages. Therefore, in case of 16 point Radix-4 FFT requires k=2 stages. [9]

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2015

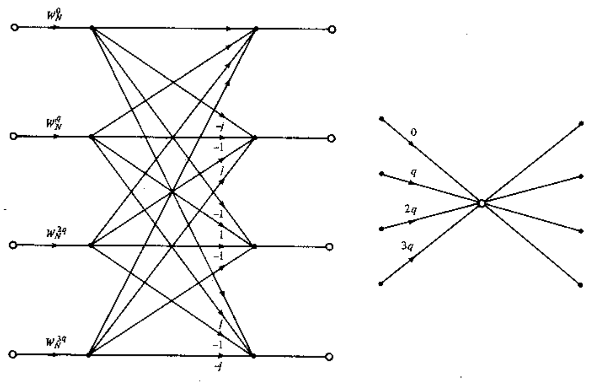

Fig. 2 The basic butterfly for radix-4 FFT algorithm[9]

The signal flow graph of Radix-4 DIT butterfly operation is illustrated in figure 3.Radix-4 algorithms have a computational advantage over Radix-2 algorithms because one Radix-4 butterflydoes the work of four Radix-2 butterflies, but Radix-4 butterfly requires only three complex multipliers compared to four complex multipliers of four Radix-2 butterflies. The arithmetic kernel of Radix-4 DIT FFT is the buttery operation defined as

$$X0 = P_0 + W_1 P_1 + W_2 P_2 + W_3 P_3 \ldots\ldots\ldots(1)$$
$$X1 = P_0 - jW_1 P_1 - W_2 P_2 + jW_3 P_3 \ldots\ldots\ldots(2)$$
$$X2 = P_0 - W_1 P_1 + W_2 P_2 - W_3 P_3 \ldots\ldots\ldots(3)$$
$$X3 = P_0 + jW_1 P_1 - W_2 P_2 - jW_3 P_3 \ldots\ldots(4)$$

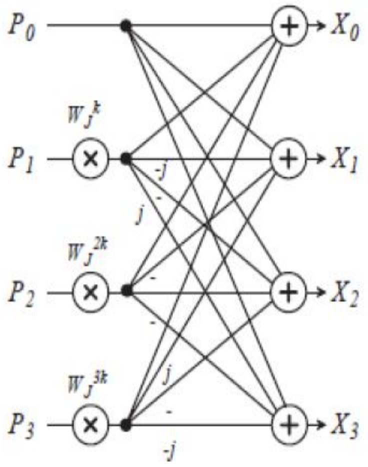

Fig. 3 Butterfly structure to show operation of radix 4 FFT [9]

IV. CORDIC

CORDIC algorithm was introduced in 1959 by Jack E.Volder for implementing a real-time navigation computer for aeronautical applications. The calculus courses provide with tools to compute the values of trigonometric functions, for example, via series expansions, polynomial, and rational function approximations. These implementations tend to require multiplication and division operations that make them expensive in hardware.

In contrast, CORDIC algorithms need only adders, shifters and comparators for computing a wide range of elementary functions. The method is efficient when fixed point implementations of signal processing algorithms on hardware are considered. For example, CORDIC is very popular in hardware accelerators and also in SIMD realizations.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2015

Furthermore, for all functional calculators employ CORDIC. CORDIC is a good choice for hardware solutions such as FPGA in which cost (gate count) minimization is more important than throughput maximization. Implementations of CORDIC by using software, enables mostof the code and data be shared between routines for trigonometric and hyperbolic functions, which helps to conserve memory. CORDIC algorithm is often used to implement rotations needed in modulators and demodulators.

V. DESIGN OF FFT PROCESSOR USING CORDIC ALGORITHM

The overall structure of processor is CORDIC based FFT processor model is shown in Figure 4. The entire model is made of the address generation unit, the control unit, the dual port RAM unit, the 4-point butterfly unit and the CORDIC twiddle factor generation unit. This model is characterized by setting the parameter, sampling points and the accuracy to meet the actual needs. To perform these operations concurrently, a dual port RAM has been employed. The control unit involves the timing control of the data storage, reading and writing to make the corresponding data and rotating factor coefficient flow into butterfly and CORDIC computing unit in sequence inFFT operation. Data and address of the 'twiddle factor' can be easily generated by the counter.

The address generation logic is very simple and does not limit the throughput of the system. When a start signal is asserted, at the same time, both to 4 Point FFT and Rotation factor generator block, the FFT block sends a Signal to CORDIC block for computing necessary twiddle factors consisting of sine-cosine terms.

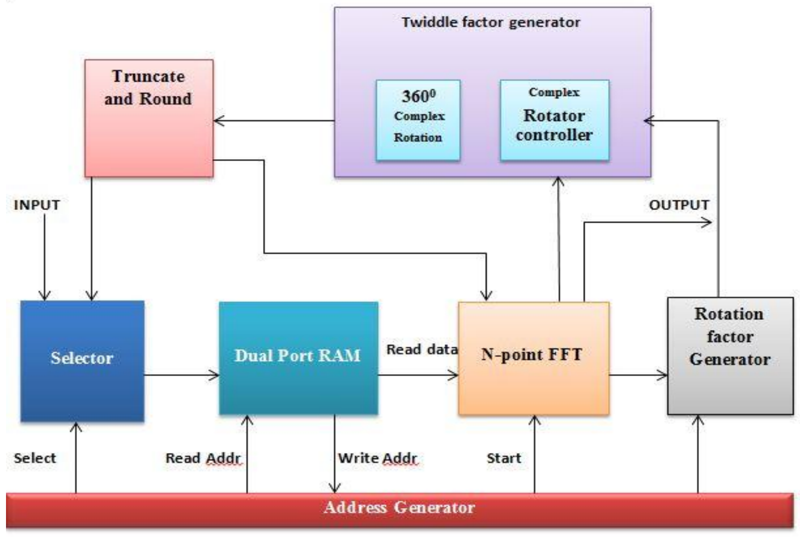

Fig. 4 Block diagram of Radix-4 FFT Processor using CORDIC.

This block is controlled by Rotation factor generator block. In truncate and round block, remapping of memory path and twiddle factors are held and fed back to FFT block. Now when address generator block sends read address signal to DRAM, it sends stored input data samples along with memory path in FFT block. Finally this twiddle factors are applied to the output of the butterflies, and a bit reverse scramble is done.

VI. RESULTS AND DISCUSSIONS

This chapter gives explanation of the design andimplementation of CORDIC core on FPGA. The simulation of CORDIC algorithm has been done on Altera Quartus IISoftware using VHDL Language and implementation done using ALTERA Cyclone II FPGA development board. After completing the design entry, simulation is done using several test benches. Figure 5 shows Pin structure of CORDIC processor.

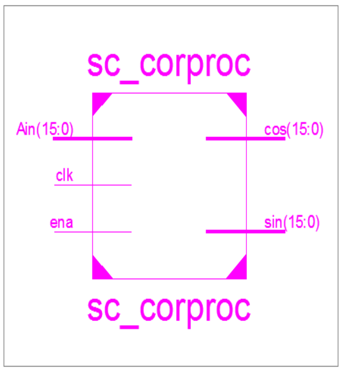

Fig. 5 PIN Diagram of CORDIC

Following diagram shows the RTL view of Pipelined CORDIC architecture, where Ain, clk and ena are input pin, and cos and sin are output pin. Table 1, explain every pins description. So, by using pipelined CORDIC in FFT processor, one can increase the computational speed of FFTprocessor.

Table 1 List of IO Ports for Sine/Cosine CORDIC Core

PORT	WIDTH	DIRECTION	DESCRIPTION
CLK	1	INPUT	SYSTEM CLOCK
ENA	1	INPUT	CLOCK ENABLE SIGNAL
AIN	16	INPUT	ANGLE INPUT
COS	16	OUTPUT	COSINE OUTPUT
SIN	16	OUTPUT	SINE OUTPUT

Figure 6 shows 15 stage pipelined RTL of CORDIC. Here, this simulation code is wrote by using VHDL and by using single CORDIC stage, it will further extend up to 15 stage called it as pipelined CORDIC architecture.

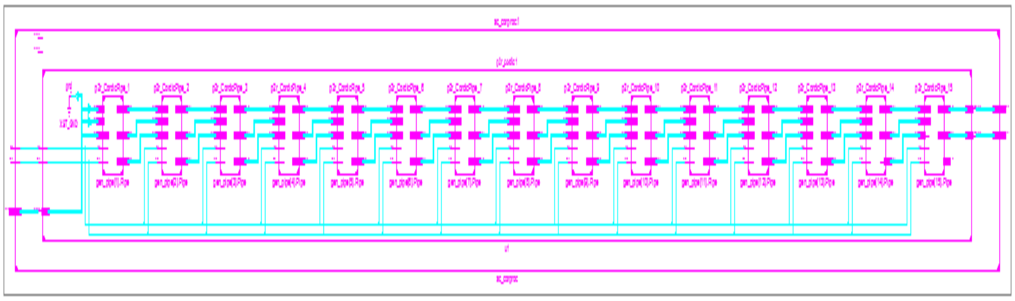

Fig. 6 RTL view of Pipelined CORDIC

VII. SIMULATIONS

Following waveform shows sine and cosine value in hexadecimal for angle0, 30, 45, 60 and 90degrees. After 15th clock pulse, sin and cos waveform shows the desired results. Timeline shows the span of 15 clock pulses. Now angles are provided at every clock pulse to show the pipeline strategy.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2015

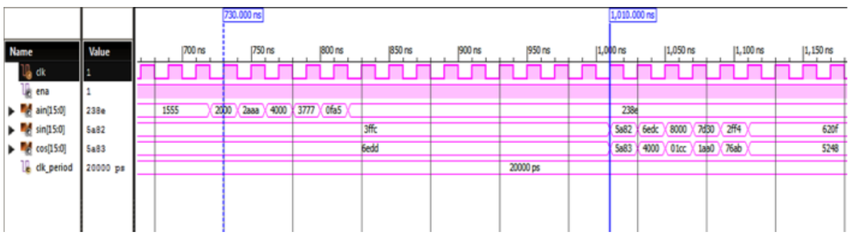

Fig. 7 Simulation result of some common Degrees.

So the output for the 45 degrees angle will take 15 clock pulses and then the output changes at every next pulse for the next four input angles. Table 2 shows hexadecimal value of an angle for some common degrees term.

Table 2 Sin/Cos outputs for some common angles

	0	30	45	60	90
SIN	0X01CC	0X3FFC	0X5A82	0X6EDC	0X8000
COS	0X8000	0X6EDD	0X5A83	0X4000	0X01CC

VIII. CONCLUSION

For generation of twiddle factor CORDIC algorithm is used. Various parts of FFT architecture such as Butterfly unit, Twiddle factor generator model, Dual port RAM are discussed. In the next phase of this paper actual Implementation of FFT processor on FPGA will be done using VHDL in which Pipelined CORDIC algorithm will initialise to optimised FFT processor.Proposed research work emphasize on the use of techniques to reduce the computational complexity of processor design and the algorithm used which result into improvement of the design significantly.

REFERENCES

[1] Yasodai A.,Ramprasad A. V., "A new memory reduced Radix-4 CORDIC processor for FFT operation", IOSR Journal of VLSI and Signal Processing, Vol. 2, Issue 5, May-Jun 2013.
[2] V. Charishma, K. SreekanthYadav, Neelimakoppala, "Design and simulation of 64 Point FFT using Radix 4 algorithm for FPGA implementation", International Journal of Engineering Trends and Technology, vol. 4, 2013.
[3] M. Vidya, M. Vijayakumar, G. Sriramulu. "Design and VLSI implementation of a radix-4 64-point FFT processor",International Journal of Research in Computer and Communication technology, Vol. 1, Issue-7, December 2012.
[4] A. S. Padekar, S. S. Belsare, "Design of a CORDIC based radix-4 FFT processor", International Journal of Computer Science and Information Technologies, Vol. 5, 2014.
[5] Hsi-Chin,HsinTze-Yun Sung, Lu-Ting Ko, "Reconfigurable VLSI architecture for FFT processor", WSEAS TRANSACTIONS on CIRCUITS and SYSTEMS, Vol. 8, Issue 6, June 2009.
[6] Javier Valls, "The use of CORDIC in software defined radios: A tutorial", Sep 2006.
[7] J. E. Volder, "The CORDIC trigonometric computing technique",Electronic Computers IRE Transactions, vol. 8, no. 3, pp. 330-334, 1959.
[8] ErdalOruklu, Xin Xiao, JafarSaniie, "Reduced memory and low power architectures for CORDIC-based FFT processors",Springer Science and Business Media, vol. 66, pp. 129-134, 16 April 2011.
[9] Ajay S. Padekar, Prof. S. S. Belsare, "Radix-4 FFT Architecture", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, pp. 337-340, May 2014.
[10] Wang Mingxing, Shi Jiangi, TianYinghui, Yang Zhe, "A Novel design of 1024-point pipelined FFT processor based on CORDIC algorithm", IEEE (ISDEA) 2012 second International Conference on Digital Object Identifier,pp. 80-83, 2012
[11]D. S. Cochran, "Algorithms and accuracy in the HP-35", Hewlett-Packard Journal, vol. 23, no. 10, 1972.
[12] J.-M Muller, "Elementary Functions: Algorithms and Implementation", Birkhauser, Boston, Mass, USA, 2004.
[13] J. E. Volder, "The birth of CORDIC," Journal of VLSI Signal Processing, vol. 25, no. 2, pp. 101–105, 2000.
[14]J. W. Cooley, J. W. Tukey, "An Algorithm for Machine computationof Complex Fourier Series, Mathematics of Computation" , vol. 19, pp. 297-301, April 1965.
[15] ShymnaNizar N. S, Abhila R. Krishna, "An Efficient Folded Pipelined Architecture For Fast Fourier Transform Using CORDIC Algorithm", IEEE International Conference on Advanced Communication Control and Computing Technologies, 2014.