International Journal of Research (IJR) Vol-1, Issue-8, September2014 **ISSN 2348-6848**

FPGA Implementation of an FFT Processor Using Cordic Algorithm

Niveathasaro v[1],Vijayasaro V[2]

[1]niveathasaro@gmail.com, [2]viji.saro17@gmail.com

[1] PG scholar, ECE Department, JJ college of engineering and Technology, Thiruchirapalli.

[2] Teaching Fellow, EEE Department, Anna University CEG Campus, Chennai.

ABSTRACT

In this paper, energy efficient high speed 128 point CORDIC based FFT processor is developed in the FPGA board. This paper develops the algorithm, architectures, and circuits necessary for high-performance and energy-efficient FFT processor. Nowadays Configurable devices are more popular. The current trend has drawbacks towards developinghardware signal processing architectures. So in this paper developing FFT architectures using ALTERA DE2 FPGA kit is a way of obtaining high performance. To increase the speed of execution further we design a CORDIC Algorithm based FFT processor using FPGA.CORDIC Algorithm is a powerful algorithm which is shift-add algorithm for computing wide range of applications including certain trigonometric, hyperbolic, linear and logarithmic functions. Reduction in complexity and increase the accuracy of rotations in the Fast Fourier transform (FFT) at algorithmic and arithmetic level carried out by CORDIC Algorithm. Here decimation in frequency algorithm is used for FFT Computation.

Keywords: FFT Processor, signal Processing, FPGA, CORDIC Algorithm.

I INTRODUCTION

Fourier Transform is virtually used in all areas of engineering and science. Fourier Transform is used to connect frequency domain and time domain. There are two kinds of FT. They areContinuous Fourier Transform and Discrete Fourier Transform. DFT is used to convert time domain signal into frequency domain signal. DFT is widely used for Digital Signal Processing Algorithms. FFT Algorithms are efficient algorithms involving a wide-range of mathematics, from simple complex number arithmetic to group theory and number theory. FFT algorithms are based on the fundamental principle of decomposing the computation of DFT.The discrete Fourier transform finds limitless applications in many areas of signal processing. The discrete Fourier transform (DFT) is an important signal processing block in various applications, such as communication systems, patient monitoring and speech, signal and image processing. The efficient implementation of DFT is fundamental in many cost and hardware constraint applications. In the era of fast computing it has become increasingly important to enhance the existing FFT algorithms to meet the ever increasing applications in the field of digital signal processing. There are basically two algorithms. They are **decimation in time** and **decimation in frequency**.Among the different kinds of FFT algorithms, the algorithms based on the approachproposed by James W. Cooley and JohnW.Tukey[7].These Cooley–Tukey (CT) algorithms present a very regular structure, which facilitates an efficient implementation. The computations of the FFT is divided into $logr(N)$ stages, where N is the number of points of the FFT and r is called the radix of the algorithm[10]. Within each stage, data shuffling and the so-called butterfly computation and twiddle factormultiplications are performed. Usually, the butterfly operations are all identical and the twiddle factor multiplications and data shuffling follow some kind of pattern. This regular structure makes them very attractive for VLSI circuit implementation.

Different hardware architectures have been used in the literature for the implementation of the CT algorithms[1]. The FFT hardware architectures can be classified into three groups:

• Monoprocessor: A single hardware element is used to perform all the butterflies, twiddle factor multiplications and data shuffling of each stage. The same hardware is reused for all the stages.

• Parallel: The computation of the butterflies, twiddle factor multiplications and data shuffling within one

stage is accelerated by using several processing elements. The same hardware elements are again reused for all the stages.

• Pipeline: A single hardware element is used to perform all the butterflies, twiddle factor multiplications and data shuffling of each stage. However, in contrast to former categories, a different hardware element is used to process each stage.

Field Programmable Gate Array is an integrated circuit to be configured by a user by using Hardware description language.The advancement of submicron CMOS technologies results in developing FPGA boards.

Advantages of FPGA

Long time availability

Can be updated and upgraded at your customer's site.

Extremely short time to Market

Fast and efficient systems

Performance gain for software applications

Real time applications

Massively parallel data processing.

II ARCHITECTURE OF FFT

The FFT processor designed is to compute the 128 pt. FFT in decimation in frequency algorithm is shown figure 5.1. It has three RAM blocks, namely RAMXI, RAMXII and RAMY each of size 128x8bit. RAMXI and RAMXII is operated here in parallel, when 128 external data is written into RAMXI, FFT computation is done, the operation on these two RAM is interchanged, FFT operation is done on RAMXI data and external data is written into RAMXII. The data is fetch in such a rate that within the data fetching time 128pt FFT computation is over along with data display. The operation of RAMs in parallel result in real time FFT computation although in this FFT computation input and output data are real but due to internal complex arithmetic involved intermediate imaginary data is generated. So there is an another RAMY block used to store intermediate imaginary at the final output power spectrum is given by vectoring X and Y data, which make Y data again zero so RAMY is needed only for intermediate storage of imaginary data. The FFT computation is done by steering data in proper sequence in butterfly unit and the results are written back in memory in proper location. Proper data sequencing is done by the address block unit which implement the signal flow graph, in sequential manner, while the butterfly unit.

CORDIC block and SUMDIFF block, implement the elementary 2 point butterfly operation with the scale output done by the SCALE block. ADDRESS block is generating the two ADDRESS location FI&SE at a time from which data is to be fetched from ram block corresponding to P&Q input of butterfly operation .It also generates desire angle of rotation **k** to perform in butterfly operation. ADDRESS block describes in detail sec 5.2. The COUNT block is 7 bit binary up counter generates address for fetching data for vectoring operation. The DISP block is also a 7 bit binary up counter generates 128 successive address location (INP) for the external input data storing in RAM block. It also generates address (DIS) for outputting data in external display device, after computation is completed there are three multiplexer having five address from ADDRESS, COUNT and DISP block has input and one of them as output depending on select input are placed on each before RAMXI, RAMXII and blocks to select approximate address as per the mode of operation. Multiplexer placed before RAMY has four address input, one less than above two since RAMY does not send data at final output. The data fetched from the RAM block by ADDRESS block is fed into two SUMDIFF block one for real data (X)and another for imaginary data (y).SUMDIFF block comprises of one 16 bit data register followed by controlled adder/subtractor.

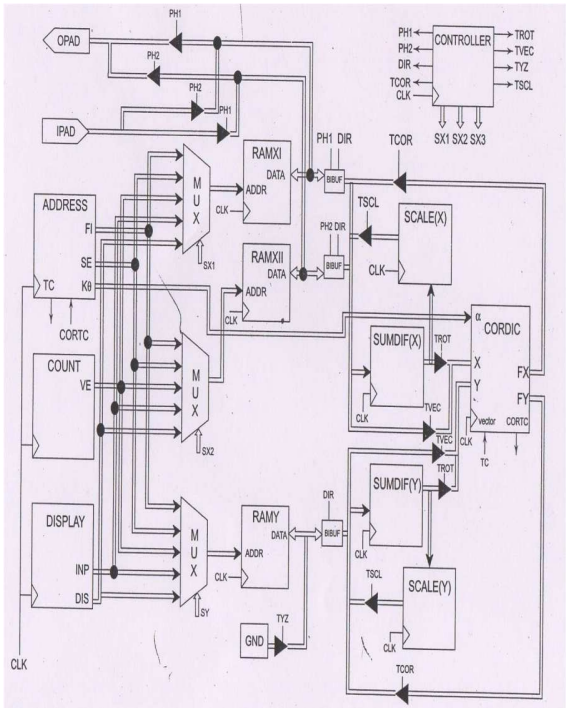

Figure 2.1: Architecture of FFT Processor Chip courtesy Neil.H.E.Weste[7]

This block first takes data P corresponding to address FI of ADDRESS block and store in register and then fed to one input of adder block and in next clock it takes data of Q corresponding to address SE and fed directly to other input of adder block. It first computes the difference P-Q and fed into CORDIC block and then it compute summation P+Q and fed into SCALE block. The CORDIC block is doing the rotational operation the real and imaginary data taken from the output of two SUMDIFF block and angle mentioned by ADDRESS block. CORDIC block is described in in detail CORDIC block output X and Y is again written back in same location (SE) of Q data of RAMX and RAMY block respectively which will be used again as new Q value for subsequent butterfly operation. So FFT operations proceed stage by stage by taking output of previous stage as input of next state. They are two SCALE block (X&Y) used for multiplication of P+Q output by the same factor of scaling inherent in CORDIC operation in order to maintain ratio at output same and output is fed to same location(FI) of P data thus updating P for subsequent butterfly operation. This scaling operation is done at the same time when CORDIC block is doing it operation thereby requiring no additional time for scaling.

III CORDIC ALGORITHM

The trigonometric algorithm is called CORDIC, Coordinate Rotation Digital Computer.The trigonometric functions are based on vector rotations, while other functions such as square root are implemented using an incremental expression of the desired function. The CORDIC Algorithm provides as an iterative method of performing vector rotations by arbitrary angles for shifts and adds.

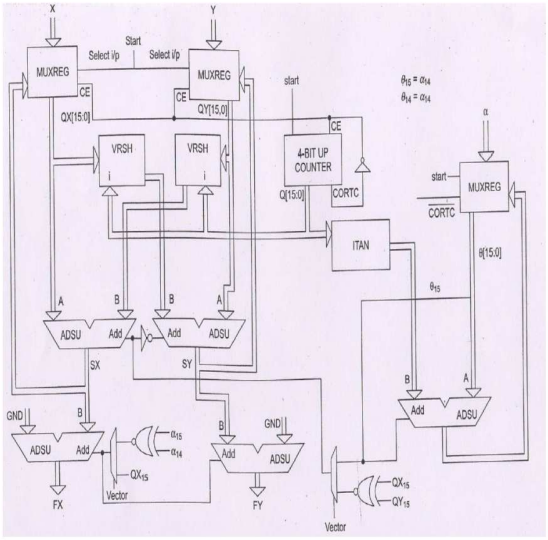

Figure 3.1: Architecture of CORDIC block courtesy Neil.H.E.Weste[7]

The CORDIC rotator is normally operated in one of two modes. The first, called rotation by Volder, rotates the input vector by a specified angle (given as an argument). The CORDIC equation by Volder expressed as

$$x' = x\cos\emptyset - y\sin\emptyset \quad (1)$$

$$y' = y\cos\emptyset - x\sin\emptyset \quad (2)$$

The second mode, called vectoring, rotates the input vector to the x axis while recording the angle required to make that rotation.

The CORDIC block performs the vectoring operation after rotation operation to compute power spectrum. After vectoring operation is complete data is fed into output device. Here the RAM block used to have bidirectional data bus. So the various data input and output to the RAM block are coming through tristate buffer, so that when one bus is active, other buses are in high impedance state. These tristatebuffer, when control input is 1, connect the input to output, otherwise remain tristated. There is a controller, which generated all the necessary control input, such as select input of multiplexer, enable input of tristate buffer, for the overall coordination of various block. For example, when PH1 control is high and PH2 control is low, external input is written in RAM XII and output data is read from RAM XI and vice versa. Similarly for CORDIC operation when TROT is high, data is entering in CORDIC from SUMDIFF block for rotational operation, and when TVEC is high, data is entering in CORDIC directly from the RAM. Similarly VECTOR input

coming from controller, when high CORDIC unit perform vectoring operation. Three different select input s SX1, SX2 &SY each of three bit of three mux block are generate from the controller to steer to proper address at the input of respective RAM block. There are 10690 clock cycle required for the 128 pt. FFT operation this is the multiplexer less architecture, and so less hardware intensive with reasonable speed. This processor is implemented in XILINK 4025 series FPGA consuming 766 CLBs.

IV RESULTS AND DISCUSSIONS

This chapter describes the design and implementation of FFT Processor on FPGA. It describes, in brief, about simulation had been done in XILINX Software using Verilog Language and implementation done using ALTERA FPGA BOARD. System development is done in incremental steps. At each successive step, test cases are developed and simulation is done to verify the correct behavior. At any step, if any violation from the expected behavior is found, the design entry is modified to rectify the violation and the process is repeated until all design expectations are met. Initially, after completing the design entry, simulation is done using several test benches.

4.1Simulation Results

4.1.1 Output of Ram Memory Block:

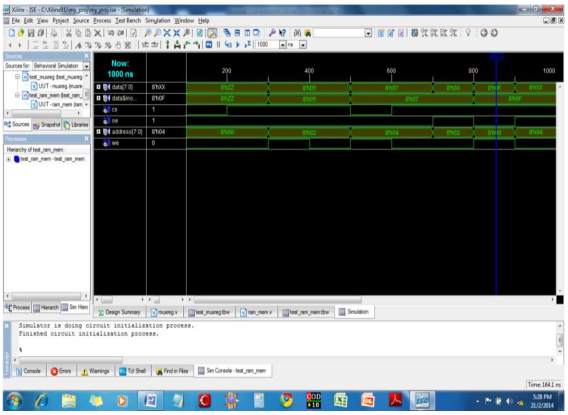

Figure 4.1: Simulation Result of RAM Block

4.1.2 Output of Multiplexer:

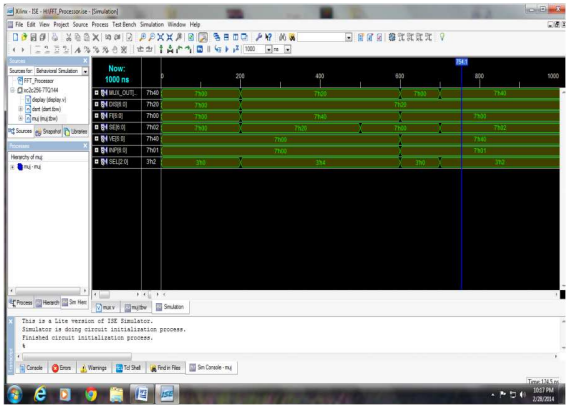

Figure 4.2: Simulation Result of Multiplexer

4.1.3 Output of Counter:

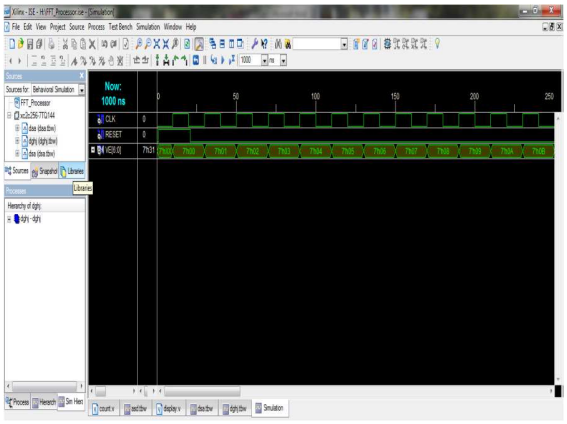

Figure 4.3: Simulation Result of Counter

4.1.4 Output of Display:

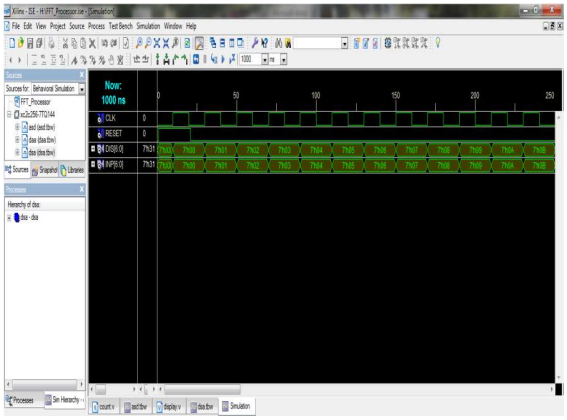

Figure 4.4: Simulation Result of Display

4.1.5 Output of Scale Block:

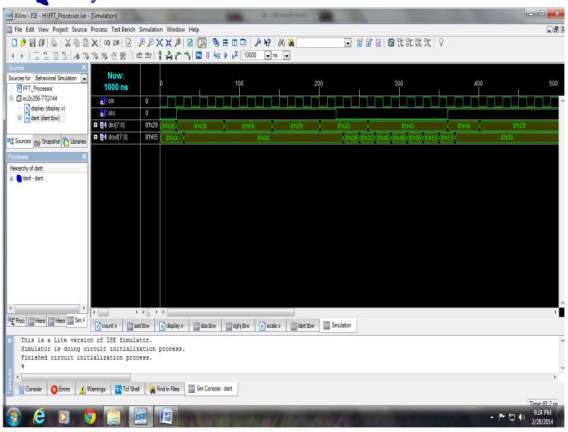

Figure 4.4: Simulation Result of Display

4.1.6 Output of Sumdifference Block:

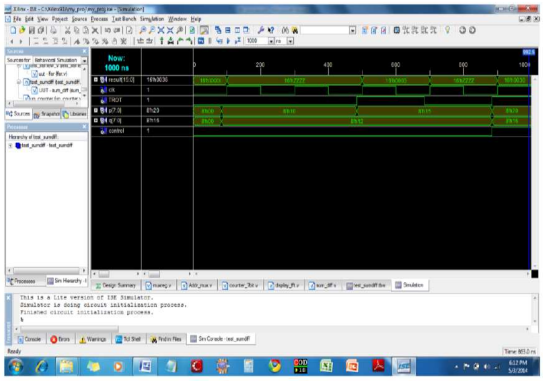

Figure 4.6: Simulation Result of Sumdifference Block

4.1.7: Output of address Block:

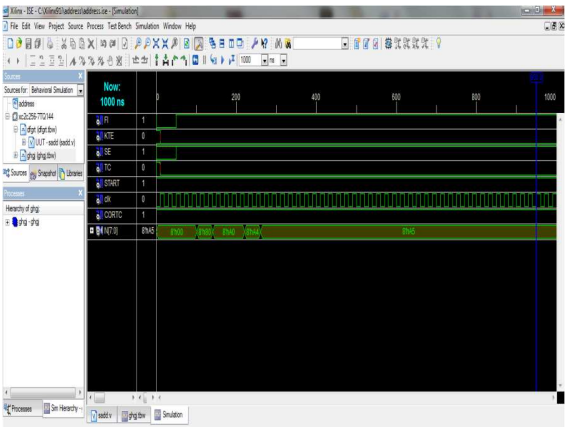

Figure 4.7: Simulation Result of address Block

4.1.8: Output of Cordic Block:

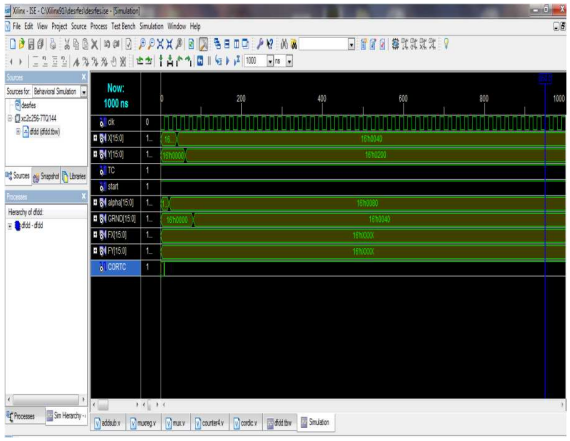

Figure 4.8: Simulation Result of Cordic Block

4.1.9:Output of Controller Block:

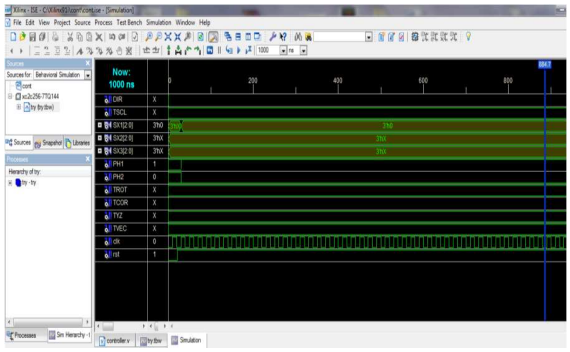

Figure 4.9: Simulation result of Controller block

V CONCLUSION

The low power and high speed FFT processor based on CORDIC algorithm Presented in this paper is well known in research and super-computing circles. However the majority of today's hardware Designs are done by engineers with little or no background in hardware efficient DSP algorithm. The new DSP designers must become familiar with these algorithms and the techniques for implementing them in FPGA's in order to remain competitive. The CORDIC algorithm is a powerful tool in DSP tool box. This paper shows that the tool is available for use in FPGA based computing machines, which are the likely basics for next generation DSP systems.

REFERENCES

[1] A.Cortes, I. Velez, M. Turrillas and J. F. Sevillano, "Fast Fourier Transform Processors: Implementing FFT and IFFT Cores for OFDM

Communication Systems" TECNUN (Universidad de Navarra) and CEITSpain.
[2] Alvin M. Despain. Very Fast Fourier Transform Algorithms for Hardware Implementation.IEEE Transactions on Computers, 28:333–341, May 1979.
[3] Cheng-Shing Wu and An-Yeu Wu. Modified Vector Rotational CORDIC (MVR-CORDIC) Algorithm and its Application to FFT. In Proceedings of the2000 IEEE International Symposium on Circuits and Systems, volume 4, pages 529–532, 2000.
[4] Digital Signal Processing: Principles, Algorithms and Applications(3rd Edition) by John G. Proakis and Dimitris K Manolakis -Prentice-Hall, Inc., - Oct 5, 1995.
[5] N. Weste and K. Eshraghian.*Principles of CMOS VLSI Design.*Addison-Wesley, 1985.
[6]Ray Andraka, A survey of CORDIC algorithms for FPGA Computers, North Kingstown,andraka@ids.net.
[7] W. Cooley and 1.Tukey, An algorithm for the machine calculation of Complex Fourier series, Math.Comput., vol. 19, pp. 297-301, April 1965.
[8] Xilinx, Inc. http://www.xilinx.com/.
[9] Xilinx, Inc. High-Performance 1024-Point Complex FFT/IFFT V2.0
[10],YousriOuerhani, Maher Jridi and A. Alfalou,implementation techniques of higher order FFT into low cost FPGA Equipe Vision, LaboratoireL@bISEN, CS 42807, 29228 Brest Cedex 2, Francee-mail: {yousri.ouerhani.maher.jridi and ayman.al-falou}@isen.fr.