

CORDIC based high Speed VLSI Architecture using Multipath R2MDC

KUTHADI. NAGARAJU ¹, Dr. DNAGA SUDHA ²

¹ PG scholar, Dept. Of Electronics & communication Engineering, JNTUH College of Engineering, Jagtial, Telangana.

² Associate Professor, Dept. of Electronics & communication Engineering, JNTUH College of Engineering, Jagtial, Telangana.

Abstract

In ultra-high sampling rates, FFT is widely used for acoustic emission signals. In this manuscript, the effectual architecture of hardware is presented based on the execution of FFT due to radix-2 frequency decimation algorithm (R2DIF) and channelled method that allows data to be effectively shared through storage by shift registers. An optimal rotation method/design uses the modified digital coordinate rotation computer algorithm (m-CORDIC) as well as Radix- 2r depending on coding scheme to replace complex multiplier as FFT. The m-CORDIC algorithm enhances computing confluence, while Radix-2r allows the logarithmic reduction of the adder steps.

Keywords: Modified digital coordinate rotation computer algorithm, radix-2 frequency decimation algorithm.

1. Introduction

The CORDIC is a commonly used algorithm in digital signal processing areas, such as imaging applications and communication systems. Image processing requires computation sizes as high as 222 [1], whereas communication systems apply several computation lengths simultaneously, such as the sizes of 128–2048 in 3GPP long term evolution (LTE) systems [2]. Consequently, long and variable-size CORDIC applications are becoming popular. CORDIC processors can be categorized into two architectures: pipeline- and memory-based architectures. Memory-based CORDICs pass the data multiple times through a single butterfly processing element (PE) or set of PEs, with several memory banks to hold intermediate results, processing the data recursively regardless of computation length. Memory-based CORDICs achieve better hardware efficiency, compared with pipeline ones. Nevertheless, the throughput of memory based CORDICs is usually restricted by the butterfly radix and concurrent data access contentions. Conflict-free address schemes for concurrent data access from different memory banks become an essential problem. A parity bit check method for one or more radix-2 PEs is first introduced in [3] and [4]. An in-place strategy from [5] reduces the total memory storage to minimum N . In [6], a mixed radix-4/2 in-place scheme makes the input and output bits symmetric, and then, the conflict-free scheme is extended to a mixed-radix algorithm. In [7], a multiple radix-2 PE scheme is demonstrated to increase the throughput of CORDIC processors. However, the methods in [6] and [7] are only suitable for power-of-two point CORDICs. In [8], a single- and multiple-PE method for arbitrary radix- b algorithm is discussed. However, it requires memory in every stage.

This paper presents a memory-based CORDIC processor design methodology with a generalized conflict-free address scheme for arbitrary-length CORDICs. We unify the conflict-free address schemes of three different CORDIC lengths, including the singlepower point (SPP) CORDICs, the

common nonsingle-power point (NSPP) CORDICs, and the NSPP CORDICs applied with the PFA, to the same address generation format. The memory bank index and the internal address are all generated by modulo and multiplication operations of the decomposition digits. Moreover, a decomposition algorithm, named highradix–small-butterfly (HRSB), utilizes high-radix PE to reduce the total number of computation cycles and small butterfly units (BUs) to eliminate the complexity of the computation engine. Thus, the CORDICs can complete N -point computations within N cycles to support the continuous-flow mode with low complexity. In the previous methods [13], [14], two successive identical symbols produce two slightly different results, because the factorization changes after each transform. To avoid this problem, we apply the same factorization to every data symbol. Furthermore, an efficient index generator, a simplified configurable MDC unit, and a unified Winograd Fourier transform algorithm (WFTA) butterfly core for point-2, 3, 4, 5 DFTs are designed for the prime-factor CORDICs. We designed two CORDIC examples in LTE systems, including a $2n$ -point CORDIC unit and a DFT unit with 35 different points (12–1296). The techniques proposed previously can be extended to arbitrary-length CORDICs.

2. Literature survey

In [9]–[11], a multipath delay commutator (MDC) architecture with high radix is used to replace the complex PE in conventional memory-based CORDICs. This method provides low-power dissipation, high data rates, good computational efficiency, and refined length flexibility. However, none of them provides a detailed conflict-free address scheme. In [12], a generalized conflict-free address scheme with single or multiple radix- 2^q MDC architectures for $2n$ -point CORDICs is illustrated. However, this scheme does not extend the principle to arbitrary-length CORDICs and more general decomposition algorithms. Generalized mixed-radix algorithms that support both traditional $2n$ -point and prime-sized CORDICs are proposed in [13] and [14]. Although [13] and [14] obtain the memory bank and address rules according to the decomposition algorithms, they do not state the data distribution procedure clearly, especially when prime factor algorithm (PFA) is applied within the CORDIC. Moreover, they cannot fully support the continuous-flow working mode.

With the upsurge of portable and standalone electronic gadgets, real-time signal processing increasingly extends its applicability. On the other hand, FFT is being extensively employed as a transform domain mathematical tool in most of the real-time signal processing algorithms [5,8]. Naturally, the relevance of designing any dedicated hardware for FFT algorithms is being highlighted nowadays. Being motivated by the widespread applicability of the FFT-based algorithms and by the pertinence of realizing the FFT algorithms in terms of hardware, we concentrate on designing dedicated general-purpose VLSI architecture for FFT. Though there are many FFT architectures available in the literature, the number of such architectures can easily be outnumbered by the vast applicability and massive demand for FFT architectures. Most of the real-time signals processing systems demand hardware modules with high operating speed, high throughput, and low latency [14]. Similarly, the area efficiency is of utmost importance for any signal processing systems operating on any resource-constrained portable standalone devices. Keeping these on our mind, we set our objective to design a low-latency area-efficient FFT architecture having high throughput and high maximum operating frequency. So far, most of the available.

3. Preliminaries

3.1 Overview of FFT

Cooley and Tukey established FFT in 1965. Fast Fourier transform is a maximum effectual procedure to calculate DFT in a finite series and necessary for fewer calculations. FFT structure is decomposed into minimum transform and then combine them to obtain a total transform. FFT contains two properties twiddle factor, and they are symmetry and periodicity property. Depending upon the principle of fundamental decomposing calculation time, DFT sequence represents length I into consecutively reduces DFT. Decimation in time and decimation in frequency denotes two types of FFT algorithm. In FFT algorithm decimation in frequency is one of the most popular forms. The sequence of output is split into smaller and smaller subsequence's, which is the frequency decimation name. At first, the sequence of input is split into two sequences contains the first $i/2$ samples of the sequence of input, respectively. In FFT processor, block diagram contains the controller, RAM, ROM, an address generator unit, and an input buffer. Here, the input data from the buffer unit disintegrated as two streams of parallel data as well as entering element process after a suitable delay moment utilizing the commutator. After suitable address generation to data processing, output comes out as R . (Figures 1, 2)

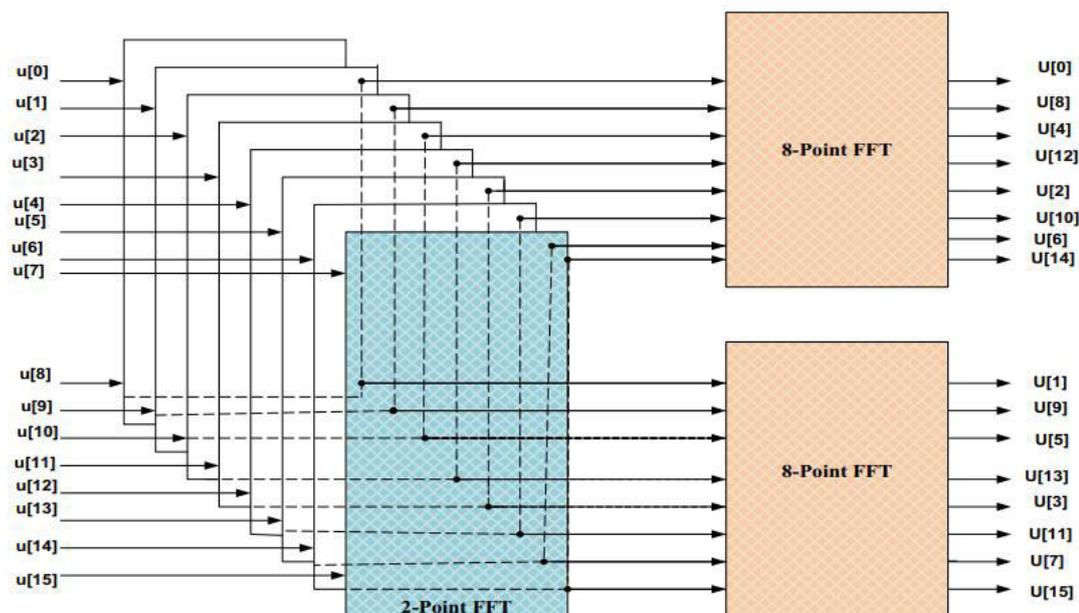


Fig. 1 Graphical representation of Cooley–Tukey algorithm

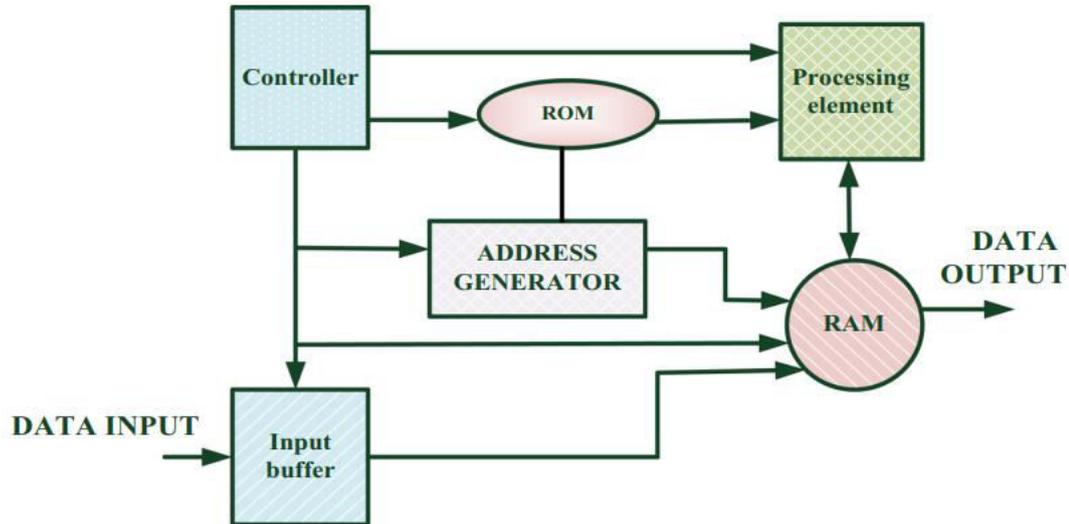


Fig. 2 Basic block diagram of FFT Processor

3.1.1 Radix-2 DIF Algorithm for Computing DFT

In DFT, input sequences of domain time are computed by using an efficient computing algorithm called FFT) time domain input sequences. In this, an input sequence $u(i)$ is mapped into its relative frequency domain representation $U(l)$. Therefore, Eq. (1) shown represents the I-point DFT of the input sequence $u(i)$. where W_N^{li} denotes the complex twiddle factor of FFT and the equation for twiddle factor (TF). Symmetry as well as complex twiddle factor duration is used to calculate the DFT of the input sequence. The computational complexity of DFT is reduced by using symmetry as well as complex twiddle factor duration is used to calculate the DFT of the input sequence. The computational complexity of DFT is reduced by using FFT algorithm. Thus for calculating DFT, FFT disintegrates the output sequence of I-point as $U(l)$ into $U(2l)$ and $U(2l+1)$.

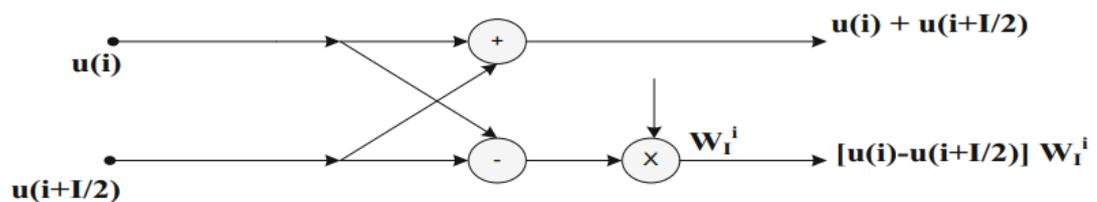


Fig. 3 Radix -2 butterfly operation in DIF

3.1.2 FFT Using Radix-2 Butterfly Operation in DIF

In DIF algorithm, Fig. 3 depicts the Radix-2 butterfly operation. Thus, FFT is calculated through reproducing this butterfly operation since it imperatives less number of adders and subtractors followed by complex twiddle factor multipliers for computation [27]. Figure 3 depicts a basic butterfly diagram to radix-2 DIF. Thus an N-point FFT used to calculate as series $(I/2) \log_2 I$ stages, in which each stage performs $(I/2)$ butterfly operations. In Fig. 4, FFT algorithm is depicted as signal flow diagram for 16-point R2DIF. It contains four methods with 32 butterfly units in the entire structure. Here, the output obtained is in bit-reversal order, since the input is in normal order. Therefore, it is reciprocal to that of DIT.

3.2 CORDIC

Algorithm Jack E. Volder introduced CORDIC algorithm in 1995 for implementing a real-time navigation computer for aeronautical applications. CORDIC needs expensive hardware for computing the values of trigonometric functions, through series expansions, polynomial, and by rational function approximations. Also, in contrast for computing a variety of elementary functions, CORDIC algorithms require only adders, shifters, and comparators. CORDIC is a good choice for gate count reduction in hardware solutions such as FPGA. CORDIC implementation saves memory while performing trigonometric and hyperbolic functions using software that enables most of the data to be shared between routines. Also, the rotations in modulators and demodulators are implemented by using CORDIC. Linear, trigonometric, logarithmic, and hyperbolic functions are the wide range of functions used to compute the iterative unrolled CORDIC algorithm based on simple operations such as shift, add, and subtract. The multiplication of twiddle factor is performed based on butterfly function without dedicated multipliers or integrated functional blocks. Figure 5 represents the CORDIC processor block diagram, and it has a controller unit, a register block as well as block for providing the data path. The data path unit involves a CORDIC block, a program counter, and an instruction decoder.

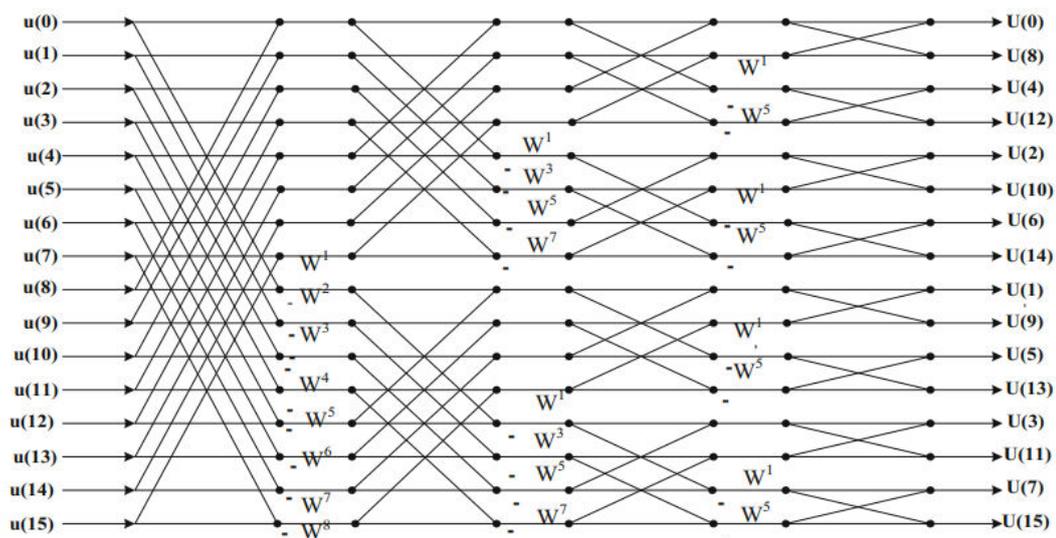


Fig. 4 Butterfly diagram for 16-point FFT algorithm

4. Proposed Methodology

The work proposes the design based on the processor of FFT in FPGA devices due to performance of CORDIC and RADIX-2r algorithm. The usage of improved CORDIC and Radix-2r Algorithm reduces the number of adders in the architecture. The hardware complexity is high in the advanced CORDIC architecture. The number of additions is thus reduced in design due to the novel angle set. Combination of minimum repetition count, minimum area, as well as minimum energy is achieved in this design. When compared to prior designs, the advanced CORDIC design occur minimum latency, possesses very less hardware overhead and thereby requires some adders. With the help of RADIX-2r , the critical path (speed) as well as the power consumption is measured in a good manner. The dominant factor in power consumption involves the shortest path that reduces the number of glitches. Therefore, running of online version with the ‘cost adder’ and ‘depth adder’ options, cost-oriented and depth-oriented values based on RADIX-2r are obtained.

In other case, RADIX-2r includes the reduction of logarithmic based on the adder steps. RADIX-2r solution represents the word length result and possible results never overflow to every individual solution. Each restriction is involved based on constant shares as similar odd multiple set without RADIX-2r solution. Critical path can be derived with the most precise delay metric (bit depth) based on the bit-level] description of RADIX-2r. Thus, based on the improved coordinate rotation digital computer (CORDIC) approach with RADIX-2r, a highly efficient rotation algorithm is presented in this paper.

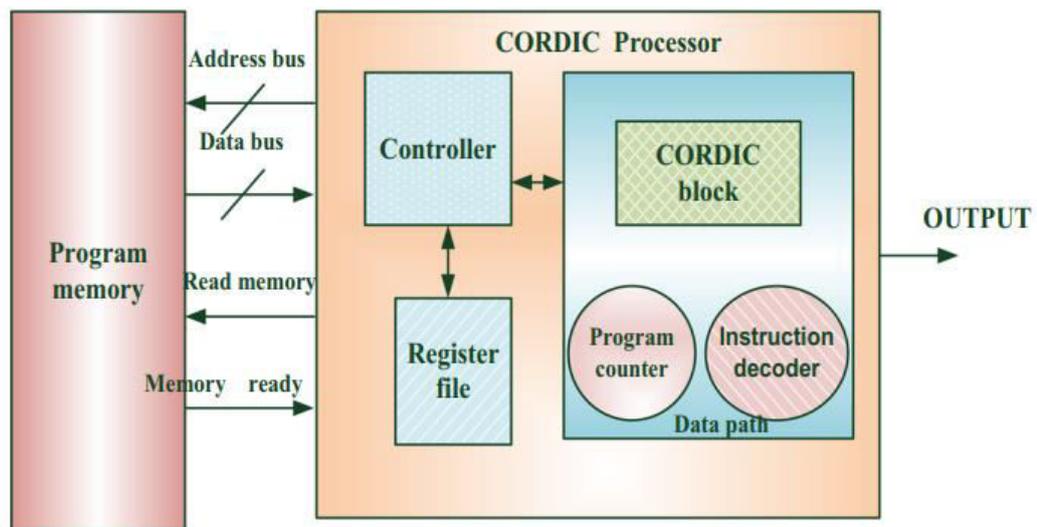


Fig. 5 Block diagram of CORDIC Processor

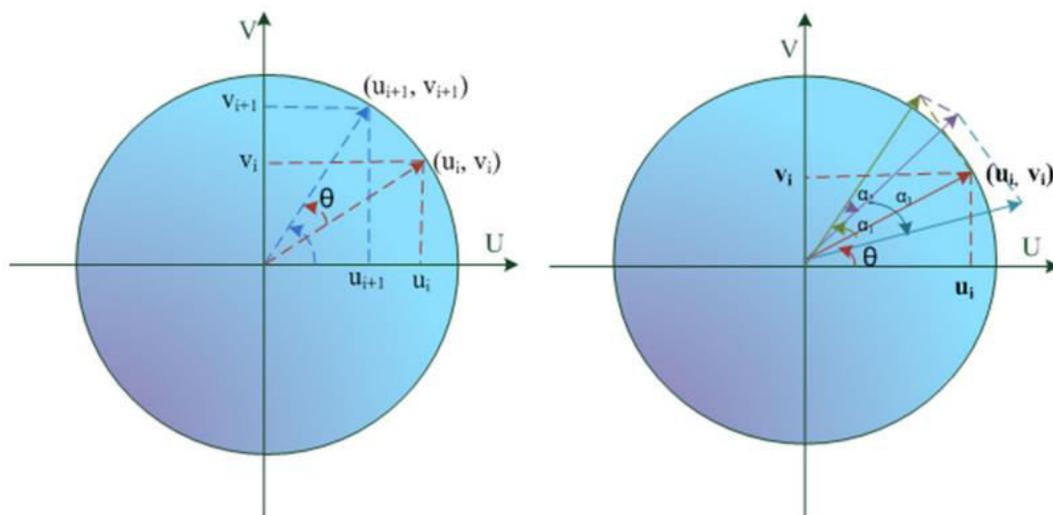


Fig. 6 a Rotation of a vector u_i, v_i . by the angle θ b Rotation through smaller angle α_i .

4.1 Hardware Implementations of Pipelined FFT Processor

The implementation of FFT processor [10] is carried out in the work with the help of an efficient pipelined design on hardware with less number of adders and there is no need of memory blocks for storage and thereby it reduces the hardware complexity as well as it saves chip area. Thus, the execution speed and consumption of hardware resources are reduced in the proposed design with the use of a new optimal rotation scheme called improved-CORDIC and Radix-2r for optimal hybrid

multiplication. Thus, the established pipelined design due to FFT is more flexible, and it is easy to implement on FPGAs.

4.1.1 Implementation of R2DIF Parallel Pipelined FFT Architecture

Hardware implementation of FFT algorithms is done using two types of architectures like memory-based architecture or by pipelined architecture. Memory-based architectures commonly use more memory blocks number utilized to save as input data as well as thereby it lowers efficiency. But the Pipeline architecture achieves high speed and thereby it gains better performance with the use of shift registers at each pipelined stage to store the input data. Thus, the proposed R2DIF pipelined FFT architecture replaces the complex multipliers with the help of an improved-CORDIC rotation scheme and Radix-2r for optimization. In DIF, radix-2 is FFT's basic principle to disintegrate an I-point DFT as two I/2-point DFTs is computed. Two I/4-point DFTs utilized to compute and disintegrate to every I/2-point DFT. The frequency domain representation for I point sequence of finite length $u(i)$ is produced due to the discrete Fourier transform and is denoted as $U(l)$ represent. The two equations, complex sequences multiply rotation factors in the radix2r computation, and therefore, the process of Data scale correction could not be utilized the method and multiply correction scale factor that can lose the meaning of the utilized CORDIC algorithm.

4.1.2 Hardware Design of MDC

Conventional Radix-2 Multi-Path Delay Commutator (MDC) Parallel-Pipelined FFT Architecture Multi-path delay Commutator (MDC) chooses work based on the architecture of the pipeline architecture for R2DIF hardware implementation to maximize the performances and eliminate latency during calculation. At this architecture the input data is processed in parallel, and it is separated as two parallel streams that are then forwarded to a butterfly unit based on the sequential order. Hence, shift register utilized to store data in the butterfly unit and also, they are used to establish suitable delay in that.

Thus, the MDC architecture cannot require any complex multipliers for multiplication as well as it does not require large number of memory block for storing intermediate data. Figure 7 shows the block diagram of Radix-2 DIF MDC architecture. The hardware architecture for a conventional R2MDC pipelined stage is depicted in Fig. 8. In butterfly unit, adder and subtractor are the input and it is divided into streams of two parallel-data namely, Positive input data as well as the Negative input data, respectively. Here, the output of the adder, $u(i) + u(i + I/2)$, straight fed to switch, while the subtractor output $u(i) - u(i + I/2)$ is directly fed unit of multiplier while performing the multiplication of twiddle factor. Thus, the multiplier is kept in shift registers prior they are forwarded to the switch. Output from both adder and the multiplier are received on the switch unit. Then, the rearrangement of data takes place in the switch unit before forwarding to the next stage. At last, the positive output data and negative output data are received from the switch unit.

Typical R2MDC parallel channelling architecture based on 16 point FFT consists of four stages, which denotes the multiplier complex, register shift to input data storage switch, to rearrange the needed data based on calculation, two adders and a controller to produce suitable control signals. It shows the 16 point FFT architecture, using parallel-pipelined R2MDC. In butterfly unit, adder and subtractor are the input and it is divided into streams of two parallel data namely Positive input data as well as the

Negative input data, respectively. In this, the first stage possesses $I/2$ registers before the butterfly unit, and it consists of $I/4$ registers after the multiplier. Thus, there are $\log_2 I - 1$ multipliers, $2 \log_2 I$ adders, and $1.5 I - 2$ shift registers for I -point DFT in the architecture of R2MDC parallel-pipelined stage. Architecture of the parallelMDC at radix-2 is suggested to the CoDIDOS to improve performance as well as to lower the latency during FFT computation. MDC stands for multi-path delay commutator; Co DIDOS stands for continuous dual input and output streams. Established the channeling of parallel R2MDC design establishes latency with the architecture of MDC and utilize the Co DIDOS. Architecture of MDC is continually fed with data streams as input and produces data streams as output. During dataflow CoDIDOS processing with FFT 16-point. Here, the data flow takes place in 4 stages. At first 16 samples input data, 0–15 splitted with two parallel streams, each containing 8 samples. Thus the samples fed to the butterfly unit, in which the FFT computation takes place in the input data points and then the data points used to store with shift register. Data samples stored with shift register are then fed to the Commutator unit in which the rearranging of data samples takes place before transferring to the next stage.

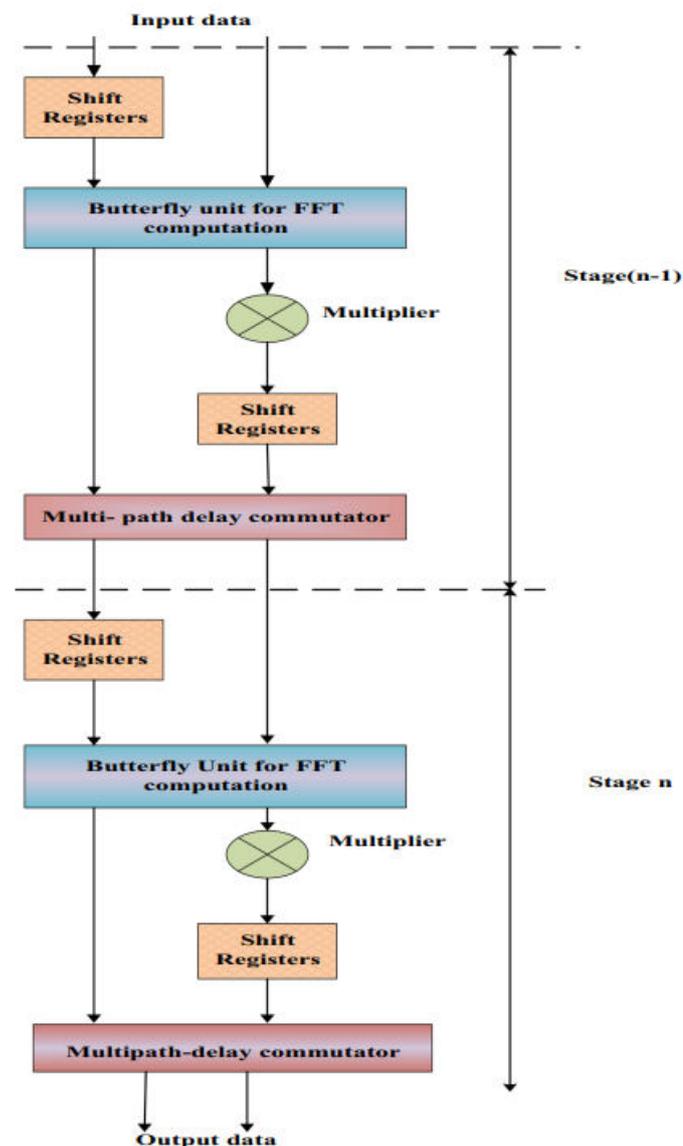


Fig. 7 MDC architecture for Radix-2 DIF FFT

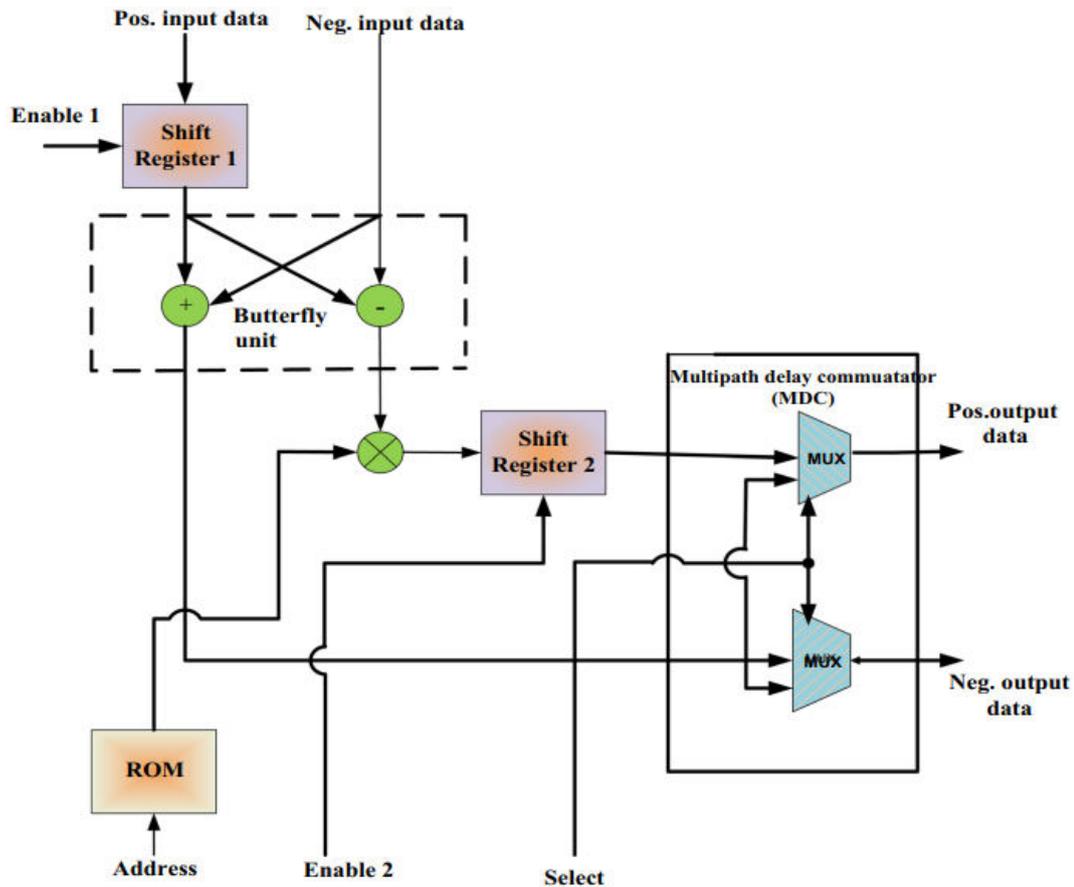


Fig. 8 Architecture of a R2MDC with pipelined stage

4.2 Hardware Architecture of Pipelined Unrolled CORDIC Algorithm

In realization, CORDIC algorithm done by utilized the channelled architecture, shows in Fig. 9. In pipelined architecture there consists of devices like shifters, registers, adders and subtractors, which are placed between different stages of the pipeline. Based on CORDIC conventional consists of less response time due to the nature of an undefined number of repetitions which requires converging. In CORDIC algorithm, number of repetitions requires converging. Therefore, an improved CORDIC and Radix-2r are proposed to constant multiplication due to the radix-2r arithmetic in this manuscript to replace the use of complex multipliers. Radix-2r possesses an important advantage while forming the critical path being completely predictable with additions (upper bound) with maximum, hence, average additions, cascaded adders (adder depth). Thus, in RADIX-2r representation of two's complement every stable is separated with segments of same bit-length. Some metrics in Radix-2r for performing constant multiplication are as follows:

- Upper bound (Upb): let A_i represent certain additions with $C_i \times X$ execution. Here C_i represents the N-bit constant. Here, $Upb = \max(A_i)$.
- Average (Avg): Number of average additions is represented
- Adder Depth (Ath): D_i represents cascaded adders' amount among any path i from input to any outputs at logic circuit based on stable multiplication.

Thus, Ath represents similar for $\max(D_i)$. Therefore, twiddle factor multiplication is carried out in the proposed design without requiring any complex multipliers and there is no need of storing the input

data, it can be used directly. Thus, in an unrolled pipelined CORDIC architecture, the input data is first shifted with the help of shifters and the controller is responsible for calculating number of bits required for shifting operation at each clock cycle, and it selects the operation types to be executed. Therefore, the operation types depend upon the direction of vector rotation. While the direction of rotation of vector is represented to the bit value of sign and the value of accumulator angle w_i . Then by using a shift register, the division operation executed with shift right as simple. Thus, after I repetitions, based on accumulator angle as w_{I-1} access as zero, in which initial angle w_0 is similar with desired angle of rotation vector. Therefore, I -bit output precision is achieved using I iterations amount for CORDIC.

The iterations amount in this technique was considered as 10. Here, the structure of repetitions is simply channeled by inserting records between various steps utilize to store the compromised data for final calculations. Here inputs are u_i , v_i , w_i , were allowed based on substantial minimizing at buffer memory block as well as full elimination of complex multipliers occurs at FFT. Controller determines shift amount and compute to perform the process in every clock cycle.

After completing N iterations using the approach, the initial angle w_0 w_{I-1} equals the desired angle value and the achieved angle w_{I-1} is similar to or less threshold specified, that also converge with the algorithm. However, during rotation of vector after I iterations, L represents constant scaling factor is developed. Therefore, the repetition number is maximal value of L approaches to 0.326. This scaling factor value is multiplied by the final outcome. Thus, Radix-2 r used to carry out this multiplication in the proposed method. Radix-2 r reduces the hardware requirement needed for the multiplication thereby increasing its execution speed as well as throughput. Hence Radix-2 r and an improved CORDIC algorithm are used for rotating the angle to get the desired output value.

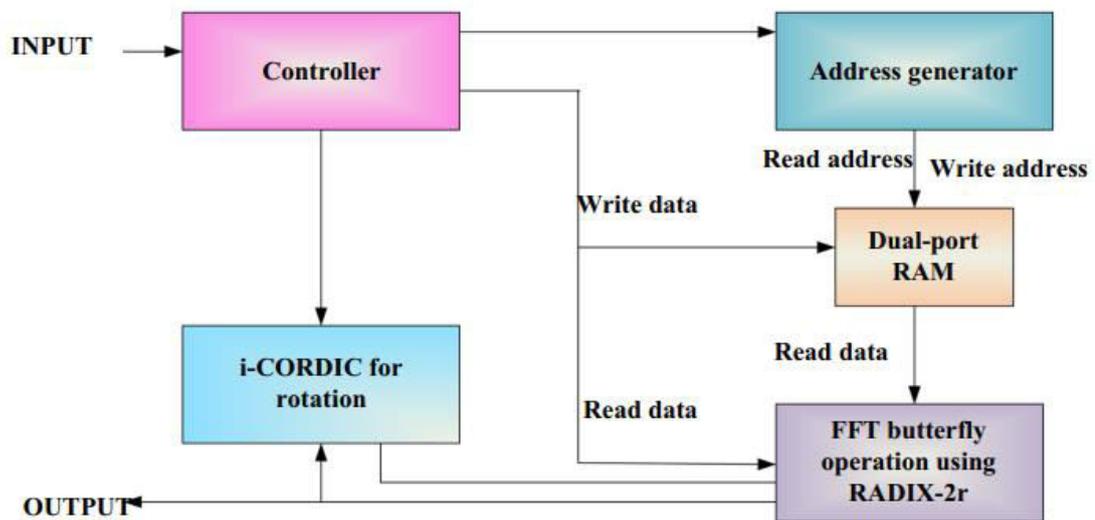


Fig. 9. Hardware architecture of Reconfigurable FFT processor.

5. simulation results

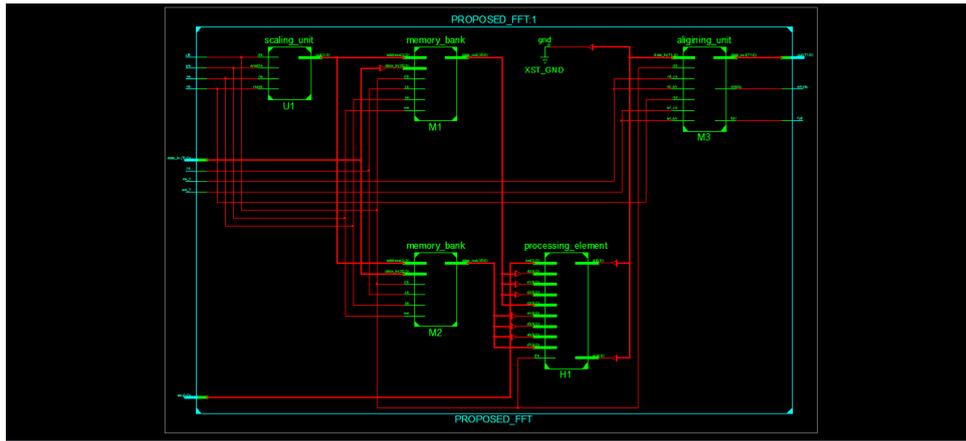


Figure 10. Existing RTL schematic

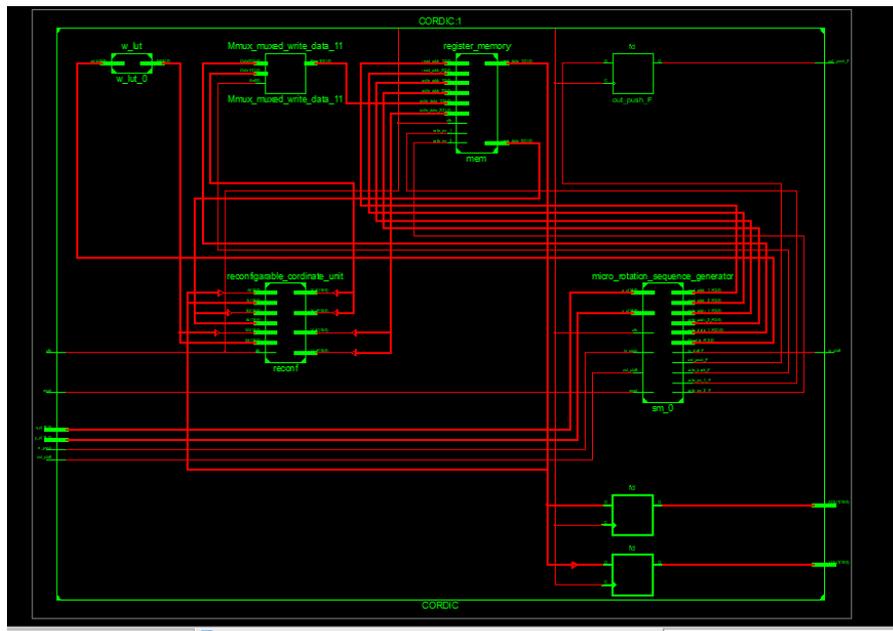


Figure 11. Proposed RTL Schematic

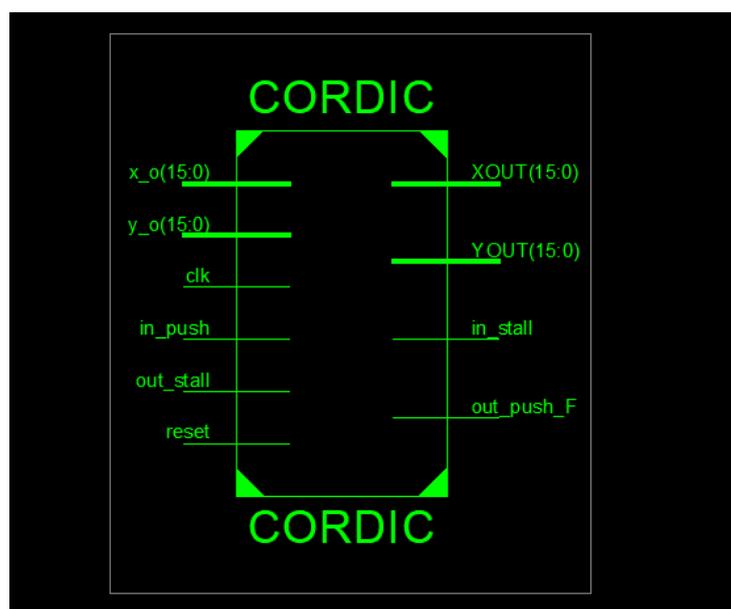


Figure 12. Proposed CORDIC Schematic

Table 1. Performance comparison

Method	Existing	Proposed
Total time	1.26ns	0.670ns
Logic delay	0.279ns	0.236ns
Route delay	0.981ns	0.434ns
Slice registers	1330	206
Slice LUTs	1309	340
Fully used LUT-FFs	536	149
IOBS	156	69
Buffers	17	1
Power consumption	0.169w	0.122w

6. Conclusion

In this manuscript, maximum performance, and minimum latency R2MDC channelled FFT processor was established and performed by the Xilinx Virtex-7 FPGA. Implementation in established design assessed based on performance, speed, latency, hardware complexity, resource utilization compared along conventional design depend on memory as well as another architecture to calculating FFT of variable length. Optimal multipliers based on CORDIC and Radix-2r work quickly with less resource of hardware. Established design uses shift registers to store complex as twiddle factors in terms of storing them in large blocks of memory. Improvements in established design outcome shows maximum performance and minimum latency in FFT processor while it is more exact, quicker, simpler, more effectual, and least cost as existing designs, less hardware costs offering the best performance in resources, cost as well as power. An experimental outcome shows established FFT processor, due to architecture of R2MDC, with optimal multiplier depend on CORDIC, Radix-2r and storage based on shift registers with twiddle factors, achieve maximum performance based on throughput, latency, speed, efficiency resources utilization. Based on the architecture regularity, established design is appropriate for calculating maximum point FFT operates at maximum speeds. Hence the established architecture cannot utilize the functional blocks as dedicated, it is more sufficient with further execution in ASIC. Experimental result shows that the proposed algorithm is better in terms of frequency, area, and power consumption as compared to other algorithms.

References

- [1]. C.-L. Yu, K. Irick, C. Chakrabarti, and V. Narayanan, "Multidimensional DFT IP generator for FPGA platforms," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 4, pp. 755–764, Apr. 2011.
- [2]. C.-H. Yang, T.-H. Yu, and D. Markovic, "Power and area minimization of reconfigurable CORDIC processors: A 3GPP-LTE example," *IEEE J. Solid-State Circuits*, vol. 47, no. 3, pp. 757–768, Mar. 2012.
- [3]. D. Cohen, "Simplified control of CORDIC hardware," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, no. 6, pp. 577–579, Dec. 1976.

- [4]. M. C. Pease, "Organization of large scale Fourier processors," J. ACM, vol. 16, no. 3, pp. 474–482, Jul. 1969.
- [5]. L. G. Johnson, "Conflict free memory addressing for dedicated CORDIC hardware," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 39, no. 5, pp. 312–316, May 1992.
- [6]. B. G. Jo and M. H. Sunwoo, "New continuous-flow mixedradix (CFMR) CORDIC processor using novel in-place strategy," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 5, pp. 911–919, May 2005.
- [7]. J. Baek and K. Choi, "New address generation scheme for memorybased CORDIC processor using multiple radix-2 butterflies," in Proc. Int. SoC Design Conf., vol. 1. Nov. 2008, pp. I-273–I-276.
- [8]. D. Reisis and N. Vlassopoulos, "Conflict-free parallel memory accessing techniques for CORDIC architectures," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 11, pp. 3438–3447, Dec. 2008.
- [9]. K. H. Chen and Y. S. Li, "A multi-radix CORDIC processor using pipeline in memory-based architecture (PIMA) FOR DVB-T/H systems," in Proc. Int. Conf. Mixed Design Integr. Circuits Syst., Jun. 2008, pp. 549–553.
- [10]. Kiranmaye, G., and Srinivasulu Tadisetty. "A novel ortho normalized multi-stage discrete fast Stockwell transform based memory-aware high-speed VLSI implementation for image compression." *Multimedia Tools and Applications* 78 (2019): 17673-17699.
- [11]. Zhang, Dong, et al. "Fast Fourier Transform (FFT) Using Flash Arrays for Noise Signal Processing." *IEEE Electron Device Letters* 43.8 (2022): 1207-1210.
- [12]. Garrido, Mario, et al. "Hardware architectures for the fast Fourier transform." *Handbook of signal processing systems* (2019): 613-647.
- [13]. Padma, Challa, Palapati Jagadamba, and Patil Ramana Reddy. "Efficient Cached 64 Point FFT Processor Using Floating Point Arithmetic for OFDM Application." *Instrumentation, Measures, Métrologies* 21.1 (2022).
- [14]. Sharma, Rahul, Rahul Shrestha, and Satinder K. Sharma. "Hardware-Efficient and Short Sensing-Time Multicoset-Sampling Based Wideband Spectrum Sensor for Cognitive Radio Network." *IEEE Transactions on Circuits and Systems I: Regular Papers* (2022).
- [15]. Singh, Karam, and Shaik Rafi Ahamed. "Scalable VLSI architecture for Hadamard transforms of HEVC/H. 265 video coding standard." *2020 24th International Symposium on VLSI Design and Test (VDATE)*. IEEE, 2020.
- [16]. Sivanandam, Kaliannan, and P. Kumar. "Design and performance analysis of reconfigurable modified Vedic multiplier with 3-1-1-2 compressor." *Microprocessors and Microsystems* 65 (2019): 97-106.
- [17]. Wang, Jian, Songting Li, and Xianbin Li. "Scheduling of data access for the Radix-2k fft processor using single-port memory." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28.7 (2020): 1676-1689.
- [18]. Hua, Siliang, et al. "Optimization and implementation of the number theoretic transform butterfly unit for large integer multiplication." *Journal of Information Security and Applications* 59 (2021): 102857.

- [19]. Dhilipkumar, P., and G. Mohanbabu. "Energy Conservation of Adiabatic ECRL-Based Kogge-Stone Adder Circuits for FFT Applications." *Intelligent Automation & Soft Computing* 32.3 (2022).
- [20]. Eleftheriadis, Charalampos, and Georgios Karakonstantis. "Energy-efficient fast fourier transform for real-valued applications." *IEEE Transactions on Circuits and Systems II: Express Briefs* 69.5 (2022): 2458-2462.