World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:9, No:12, 2015

Modified Scaling-Free CORDIC Based Pipelined Parallel MDC FFT and IFFT Architecture for Radix 2^2 Algorithm

C. Paramasivam, K. B. Jayanthi

***Abstract*—**An innovative approach to develop modified scaling free CORDIC based two parallel pipelined Multipath Delay Commutator (MDC) FFT and IFFT architectures for radix 2^2 FFT algorithm is presented. Multipliers and adders are the most important data paths in FFT and IFFT architectures. Multipliers occupy high area and consume more power. In order to optimize the area and power overhead, modified scaling-free CORDIC based complex multiplier is utilized in the proposed design. In general twiddle factor values are stored in RAM block. In the proposed work, modified scaling-free CORDIC based twiddle factor generator unit is used to generate the twiddle factor and efficient switching units are used. In addition to this, four point FFT operations are performed without complex multiplication which helps to reduce area and power in the last two stages of the pipelined architectures. The design proposed in this paper is based on multipath delay commutator method. The proposed design can be extended to any radix 2^n based FFT/IFFT algorithm to improve the throughput. The work is synthesized using Synopsys design Compiler using TSMC 90-nm library. The proposed method proves to be better compared to the reference design in terms of area, throughput and power consumption. The comparative analysis of the proposed design with Xilinx FPGA platform is also discussed in the paper.

***Keywords*—**Coordinate Rotational Digital Computer(CORDIC), Complex multiplier, Fast Fourier transform (FFT), Inverse fast Fourier transform (IFFT), Multipath delay Commutator (MDC), modified scaling free CORDIC, complex multiplier, pipelining, parallel processing, radix-2^2.

I. Introduction

FAST FOURIER TRANSFORM (FFT) and Inverse Fast Fourier Transform (IEFT) are used in telecommunication and discrete signal processing. FFT/IFFT architectures are consuming high power and occupy more area. At present FFT/IFFT designs have optimized architectures with increased speed and low power consumption. FFT algorithm helps efficient DFT computation with increased speed. Researchers always try to reduce the chip area without affecting the performance of the design. FFT architectures can be classified into two categories: pipelined and in-place architectures. Pipelined architectures contain either feedforward or feedback data paths [1], [4]. The feedback architectures are divided into two types: Single-path Delay Feedback (SDF) and Multipath Delay Feedback (MDF). The feedforward architectures are divided into two types: Multipath Delay Commutator (MDC) and Single-path Delay Commutator (SDC). Many researches are carried out to design pipelined architectures for computing the FFT for high throughput applications [2], [3]. Pipelined FFT architectures require $\log_r(N)$ processing elements and the calculation is $\log_r(N)$ times faster than the memory architecture. So Many parallel architectures for FFT are available in [1]-[28]. In [1], [4]-[10] Single Path Delay Feedback (SDF) architectures are presented. In [11]-[19] Multipath delay Feedback (MDF) architectures are presented. In [3], [4], [20]-[26] Multipath Delay Commutator (MDC) architectures are presented. In [27], [28] single path delay commutator (SDC) architectures are presented. In [29]-[31] CORDIC based in-place FFT architectures are presented. Based on the above literature review the Multipath Delay Feedback (MDC) architecture is suitable for real time application like Orthogonal Frequency Division Multiplexing (OFDM) modulation.

In [1], [4], the radix 2^2 Single path Delay Feedback (R2^{2}SDF) FFT architectures are presented. In [6], coordinate Rotational Digital Computer(CORDIC) based radix 4 single path delay feedback (R4SDF) FFT architecture is presented. In [10], CORDIC based Single Path Delay Feedback (SDF) architecture is presented. In [29], CORDIC based radix2 and radix4 in-place FFT architectures are presented. In [30] modified CORDIC based radix2 and radix4 in-place FFT architectures are presented. In [31], modified scaling free CORDIC based radix 2^2 in-place FFT architecture is presented. In [7], comparative study of parallel pipelined Feed Forward and Feedback architectures of the radix4 FFT algorithm is presented. The Feedforward architectures are suitable for pipelining and parallel processing and produces optimum throughput [7]. Therefore, feedforward FFT architectures are suitable for real time applications like OFDM. This paper presents Modified Scaling-Free CORDIC based radix 2^2 feedforward Multipath Delay Commutator (R2^{2}MDC) Fast Fourier Transform (FFT) architecture.

The rest of the paper is organized as follows. Section II explains radix 2^2 DIF FFT algorithm. In Section III, proposed two parallel FFT and IFFT architectures are described. In Section IV, proposed architectures are compared with previous architectures. In Section V, conclusions are given based on the analysis.

II. Radix 2^2 DIF FFT Algorithm

The DFT of size N is defined in (1):

C. Paramasivam and K. B. Jayanthi are with the K. S. Rangasamy College of Technology, Tiruchengode, Tamilnadu, India (e-mail: sivamvlsi@gmail.com, jayanthikb@gmail.com).

$$X(k)=\sum_{n}^{N-1} x(n)W_N^{nk} \qquad \text{k=0,1…..N-1} \tag{1}$$

$W_N^{nk}=e^{(-j2\pi/N)nk}$ is called "twiddle factor", x(n) is the discrete time signal and X(k) is the signal in frequency domain. Direct calculation of DFT is uneconomical, because it does not exploit the symmetry property and periodicity property of the twiddle factor [32], whereas FFT algorithms are computationally efficient algorithms since this algorithm uses symmetry and periodicity property of twiddle factor [32]. FFT calculates N number of discrete data samples in O (N $\log_2$N) computational time but Direct DFT method takes O (N^2) computational time.

The complexity of FFT architectures is decided by the radix of the FFT algorithm. On chip communication requires fan-in and fan-out of each processing node in the signal flow graph to be small. The radix 2 butterfly has less number of fan-in and fan-out than radix 4 and requires more number of complex multiplication than radix4. In [1] hardware oriented radix 2^2 FFT algorithm has been derived to restructure the signal flow graph of FFT for efficient hardware implementation. The radix 2^2 butterfly is simple as similar to radix 2 butterfly and requires less number of complex multiplications as similar to radix 4 butterfly. In radix 2^2 algorithm the main idea is to implement two stages of DIF simultaneously. This provides simplification of twiddle factors in these two stages. Time and frequency indices are decomposed as:

$$\text{n}=\left\langle \frac{N}{2}n_1+\frac{N}{4}n_2+n_3\right\rangle_N \tag{2}$$

$$\text{k}=\langle k_1+2k_2+4k_3\rangle_N \tag{3}$$

where N represents the length of the FFT and $<>_N$ represents the decomposition of FFT is evaluated based on modulo *N*. k_1, k_2, k_3, n_1, n_2, *and* n_3 in (2) and (3) are integer numbers used to decompose *k* and *n* respectively. Substituting (2) and (3) in (1) results:

$$\text{X}[k_1+2k_2+4k_3]=\sum_{n_3=0}^{\frac{N}{4}-1}\sum_{n_2=0}^{1}\sum_{n_1=0}^{1} X\left(\frac{N}{2}n_1+\frac{N}{4}n_2+n_3\right)W_n^{\left(\frac{N}{2}n_1+\frac{N}{4}n_4+n_3\right)k}=\sum_{n_3=0}^{\frac{N}{4}-1}\sum_{n_2=0}^{1}[\text{x}(\frac{N}{4}n_2+n_3)+(-1)^{k_1}x\frac{N}{2}+\frac{N}{4}n_2+n_3)]W_8^{(\frac{N}{4}n_2+n_3)k}$$

$$\text{BF}(\frac{N}{4}n_2+n_3)=\sum_{n_3=0}^{\frac{N}{4}-1}[(\text{BF}(n_3+(-j)^{k_1+2k_2}BF(\frac{N}{4}+n_3))W_N^{n_3(k_1+2k_2)}]W_N^{4n_3k_3}\text{H}\,(k_1,k_2,n_3)=\text{H}\,(k_1,k_2,n_3)W_{\frac{N}{4}}^{n_3k_3} \tag{4}$$

From (4), it is understood that N-point FFT of $X[k]$ is converted to $\frac{N}{4}$ point *FFT of H* (k_1, k_2, n_3). By changing k_1 and k_2 four different values of *H* are obtained. Fig. 1 shows the *radix-*2^2 flow graph *for 16 point FFT.*

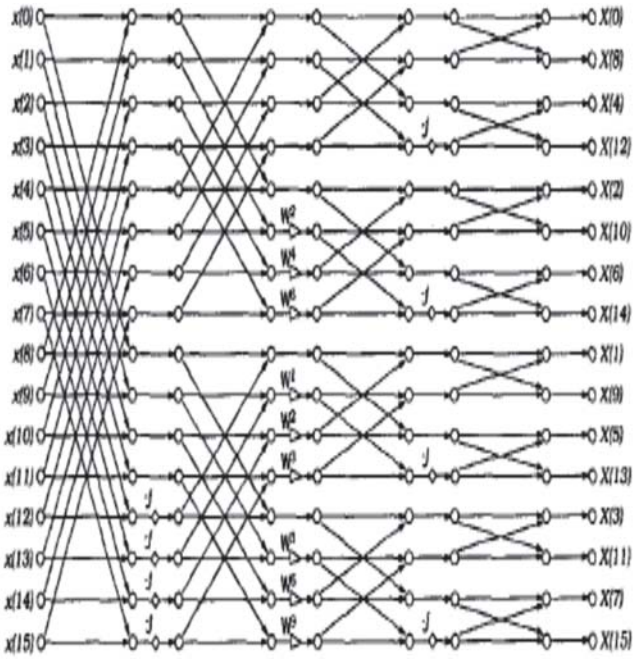

Fig. 1 Flow graph of a radix-2^2 16-point DIF FFT

III. Proposed CORDIC Based Two Parallel MDC Architecture for FFT and IFFT

A. Two Parallel Multipath Delay Commutator Architecture for radix 2^2 DIF 16- point FFT

The Proposed Modified Scaling-Free CORDIC based pipelined two parallel Multipath Delay Commutator (MDC) architecture for radix 2^2 FFT is shown in Fig. 2. The architecture includes two types of butterfly unit, two types of switching unit, delay registers, modified scaling-free CORDIC based twiddle factor generator unit and modified scaling free CORDIC based complex multiplier. Circle with a cross mark in the design shows complex multiplier. Switching unit with delay elements are used to give the input index values as same as in the flow diagram shown in Fig. 1. In [6], [7], CORDIC based twiddle factor generator unit is used. In proposed work modified scaling-free CORDIC based twiddle generator unit is used to reduce area and memory accessing power. Twiddle factors are generated by CORDIC unit and that can be used whenever it is needed in the flow diagram. To reduce hardware complexity modified scaling-free CORDIC based Complex multiplier and SW2 switching circuit and some enhancement in the butterfly unit is done in the new architecture for FFT.

B. Butterfly Units

In the proposed FFT architecture butterfly unit I and butterfly unit II are used. Butterfly unit I is shown in Fig. 3 and it is used in stage1, stage3 and stage 4, this unit performs complex addition and complex subtraction based on two's complement number representation.

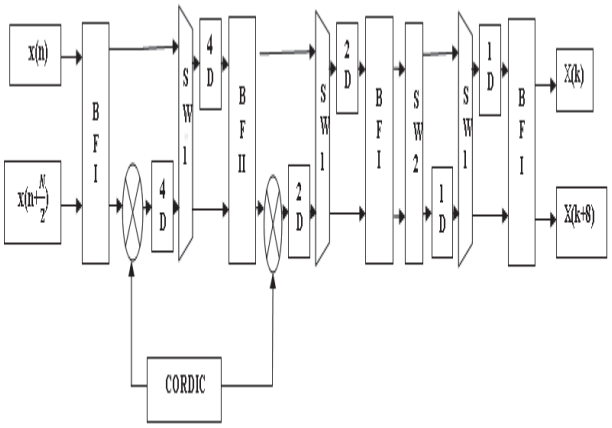

Fig. 2 Proposed Modified Scaling-Free CORDIC based two-parallel pipelined MDC architecture for a radix-2^2 16-point FFT

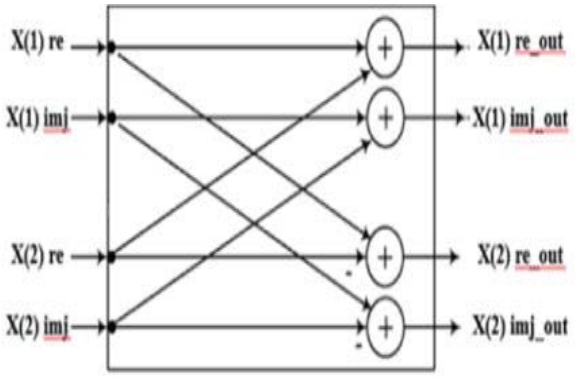

Fig. 3 Structure of the Butterfly Unit I

In this unit, inputs are separated as real and imaginary components. For addition operation real and imaginary components are added. For subtraction, subtrahend value is converted to two's complement number and then added which is nothing but the subtraction operation in the two's complement number system.

Butterfly unit II is shown in Fig. 4 and it is used in stage1 and stage2, this unit performs complex addition and complex subtraction based on two's complements number representation. In the proposed FFT architecture, first stage and second stage needs twiddle factor access. Therefore, this butterfly unit additionally has simple counter logic. This counter logic is used to generate twiddle factor address generation in the butterfly unit itself. By using this twiddle factor address the value is received from the modified scaling-free CORDIC based twiddle generator unit. Both butterfly unit output and twiddle factor values are forwarded to modified scaling-free CORDIC based complex multiplier unit.

C. Switching Units

In the proposed FFT architecture switching unit SW1 and switching unit SW2 are used. The Switching unit with appropriate delay helps to give correct index values from one stage to the next stage as in the flow diagram shown in Fig.1. The structure of the switching unit SW1 is shown in Fig. 5. This unit contains two MUX and a single control. The control signal is used to allow the value as it is in the input when the control signal is zero. When the control signal is one, the output will be interchanged inputs. The shift registers are act as delay registers; they give required delay in all the stages. For every clock cycle there will be a shift in the delay registers. After first stage four delays are used with the switching circuit, because eight point FFT should be processed in the second stage. Similarly, after second stage two delay shift register and after third stage one delay shift register is used.

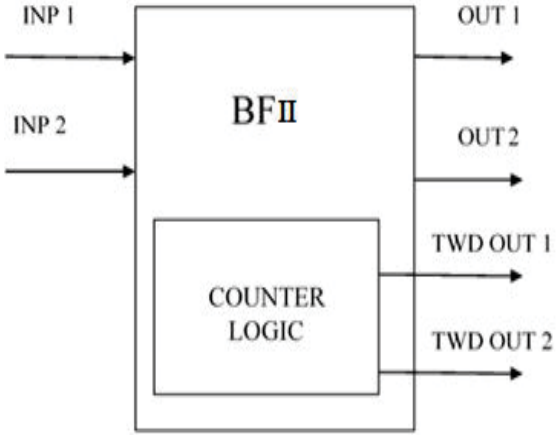

Fig. 4 Structure of the Butterfly Unit II

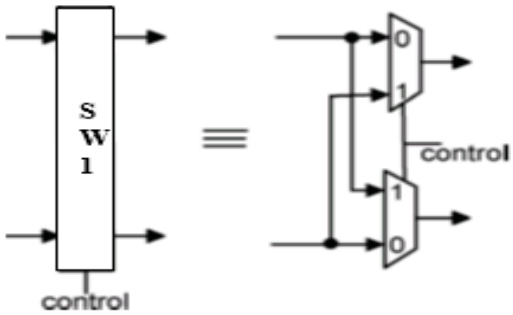

Fig. 5 Switching unit SW1

The structure of the Switching unit SW2 is shown in Fig. 6. This unit contains one MUX and a single control. In this unit first one set of inputs will be simply transferred to output. In the second set of inputs, first input is simply transferred and second input is operated with the switching policy. As in the flow graph shown in Fig. 1 third stage performs four point FFT operations. Normally four point FFT needs butterfly units and a complex multiplier unit. In four point FFT twiddle factor values are multiplied with -j, therefore instead of using complex multiplier, the switching unit SW2 and butterfly unit I perform multiplication with -j. Therefore, four complex multiplications are completely eliminated.

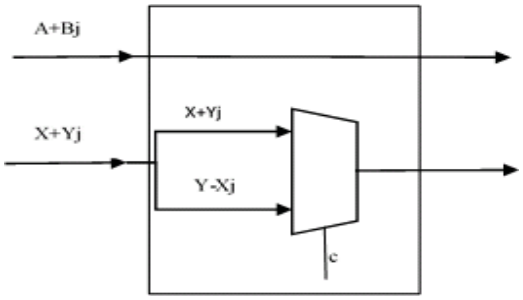

Fig. 6 Switching unit SW2

D. Modified Scaling-Free CORDIC Algorithm

Normally twiddle factors are stored in the RAM unit and that will be used based on the requirement. In the proposed method modified scaling free CORDIC algorithm is used to generate the twiddle factors. Fig. 7 shows the flow chart of the CORDIC algorithm operated in circular coordinate. In this CORDIC x input is initialized to 0.6072 and y is set to zero and z is nothing but the required angle. Based on z and d values iteration continues and stops when z is equal to zero. Where d determines the direction of the iteration. After n iteration sinθ and cosθ values are obtained. This algorithm requires scaling operation to get accurate value.

The proposed modified scaling free CORDIC algorithm has less number of iteration and it completely eliminates the scaling operation. In this algorithm most-significant-1 detector method is used to reduce the number of iteration and Taylor series method is used to eliminate the scaling operation. Flow chart of this algorithm is shown in Fig.8.Pseudo code of the modified scaling free CORDIC algorithm is given below.

Input: angle to be rotated (θ_i)
Begin
M = Most_Significant_1 location of θ_i
if (M = = 15) then
α = 0.25 radians
Shift, $s_i = 2$ and $\theta_{i+1} = \theta_i - \alpha$
Else
Shift, $s_i = 16 - M$
$\theta_{i+1} = \theta_i$ with θ_i[M] = '0'
End

E. Modified Scaling-Free CORDIC Based Complex Multiplier

Multipliers are always requiring high area and consume more power. Fig. 9 shows modified scaling free CORDIC based complex multiplier architecture. Multiplication of two complex numbers can be expressed mathematically as shown below.

$$(A+Bj) * (C+Dj) = (AC-BD) + (AD+BC)\,j$$

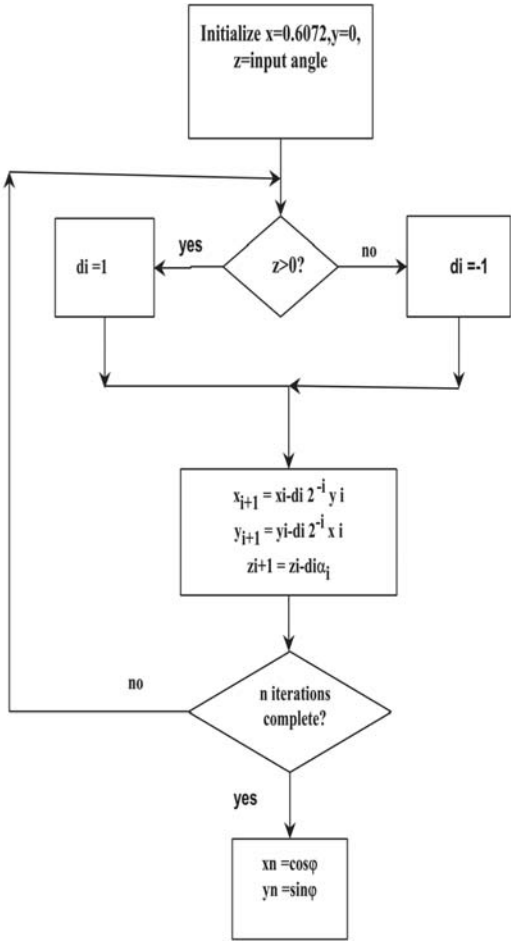

Fig. 7 Flow Chart of CORDIC algorithm

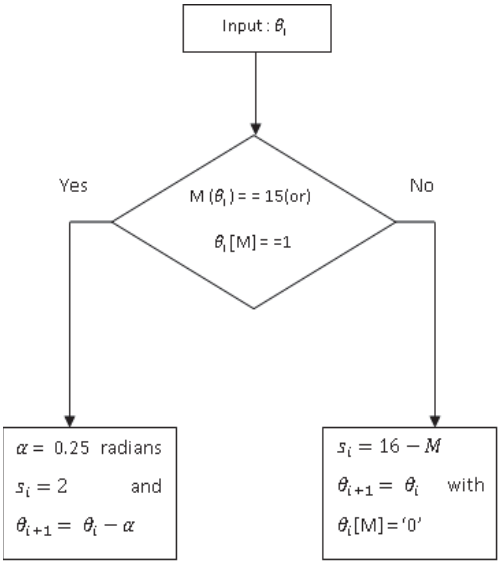

Fig. 8 Flow Chart of modified scaling free CORDIC algorithm

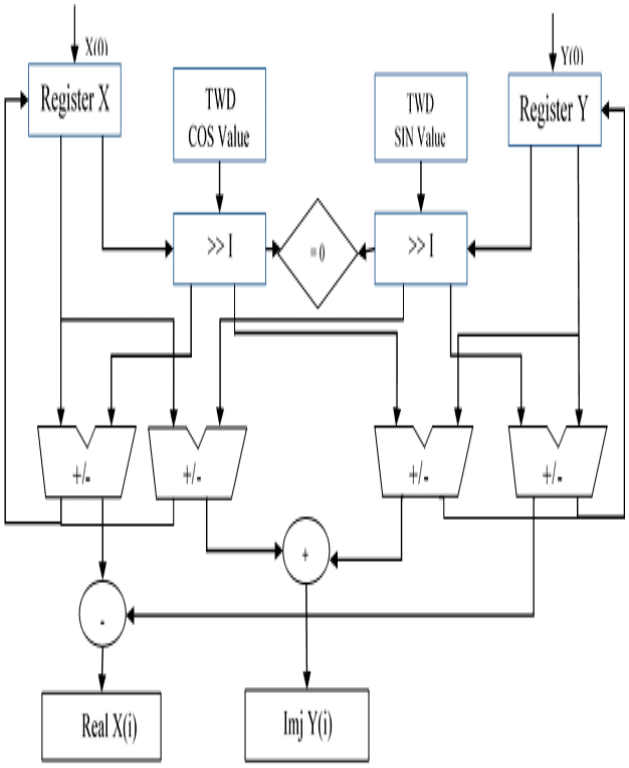

Fig. 9 Modified Scaling free CORDIC based Complex Multiplier

This complex multiplication requires four multiplications and two additions. Instead of using more registers for multiplication, this scheme uses two 32 bit registers. Inputs for this multiplier are complex multiplicand and a twiddle factor value which act as the multiplier value. For the complex multiplication operation real and imaginary values of both multiplicand and multiplier are separated. For every clock cycle twiddle factor value is right shifted to one position and checked the zeroth position of the value. If the value is zero no operation is performed and if it is one, then the multiplicand is shifted up to current shifting index and added with the previous results using accumulator. Every clock signal twiddle factor value is checked, if the value is not zero operation continues. If the value is zero, then the operation is stopped. In the proposed multiplier complex multiplication is performed by shift and addition operation. Therefore, power and area of proposed multiplier is reduced to large amount as compared to existing complex constant multiplier.

F. Two Parallel Multipath Delay Commutator Architecture for Radix 2^2 DIF 16- Point IFFT

The proposed Modified Scaling-Free CORDIC based pipelined two parallel Multipath Delay Commutator (MDC) architecture for radix 2^2 IFFT is shown in Fig. 10. The inverse of the FFT architecture shown in Fig. 2 describes IFFT architecture. As similar to FFT architecture, IFFT architecture also has two types of butterfly unit, two types of switching unit, delay registers; modified scaling-free CORDIC based twiddle factor generator unit and modified scaling free CORDIC based complex multiplier. The operation of IFFT architecture is just opposite to FFT architecture.

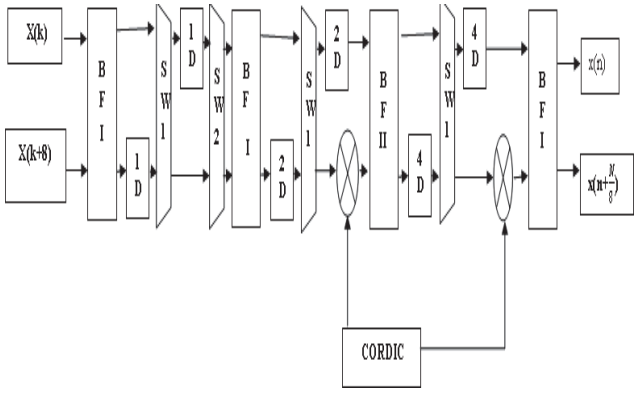

Fig. 10 Proposed Modified Scaling-Free CORDIC based two-parallel pipelined MDC architecture for a radix-2^2 16-point IFFT

IV. Implementation and Performance Evaluation

The proposed architectures are feed-forward which can process 2 samples in parallel, thereby achieving a higher performance than previous designs which are serial in nature. The performance of the proposed two parallel MDC FFT architectures are verified using synthesis tools of both ASIC and FPGA technologies. For FPGA technology the synthesize of the proposed FFT/IFFT architecture is done using Xilinx ISE13.4 software with the target device Virtex 5 ML506 (XC5VSX50T). For ASIC technology the synthesize of the proposed FFT/IFFT architectures is done by Synopsys Design Compiler using TSMC 90-nm library. The power, area and delay calculations for ASIC and FPGA technologies are presented in Tables I and II respectively.

TABLE I
Performance Comparison of Proposed and Existing FFT Architectures for ASIC Technology (Target Library TSMC 90nm)

FFT ARCHITECTURE	AREA mm²	POWER mW	Max. Clock rate (MHz)
Radix 2^2 SDF FFT [4]	0.34	2.58	125
Radix 2^2 SDF IFFT [4]	0.35	2.62	125
Proposed Radix 2^2 MDC FFT	0.28	2.46	143
Proposed Radix 2^2 MDC IFFT	0.30	2.51	143

TABLE III
Performance Comparison of Proposed and Existing FFT Architectures for FPGA Technology [Target Device: Xilinx Virtex 5 ML506 (XC5VSX50T)]

FFT Architecture	Slices	Slice Luts	Max. Clock rate (MHz)	Power (mW)
Radix 2^2 SDF FFT [4]	1095	1912	162	3.35
Radix 2^2 SDF IFFT [4]	1081	1898	162	4..02
Proposed Radix 2^2 MDC FFT	1056	1732	190	3.28
Proposed Radix 2^2 MDC IFFT	1051	1740	189	3.92

Fig. 11 shows the FPGA slice comparison chart of proposed and exiting FFT and IFFT architecture for FPGA technology. The proposed two parallel MDC FFT architecture occupies 3.56% less slice than the existing SDF FFT architecture and proposed two parallel MDC IFFT architecture occupies 2.3% less slice than the existing SDF IFFT architecture as shown in Fig. 11.

Fig. 11 Slice Comparison chart of FFT/IFFT for FPGA Technology

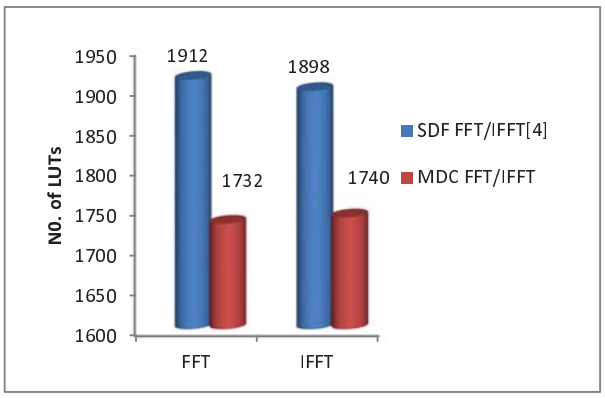

Fig. 12 Slice LUTs Comparison chart of FFT/IFFT for FPGA Technology

Fig. 12 shows the FPGA slice LUT comparison chart of proposed and exiting FFT and IFFT architecture for FPGA technology. The proposed two parallel MDC FFT architecture occupies 9.4% less LUT than the existing SDF FFT architecture and proposed two parallel MDC IFFT architecture occupies 8.3% less LUT than the existing SDF IFFT architecture as shown in Fig. 12.

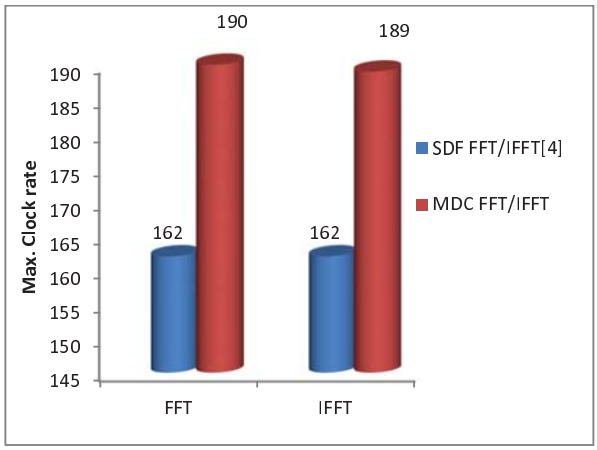

Fig. 13 Maximun clock rate Comparison chart of FFT/IFFT for FPGA Technology

Fig. 13 shows the maximum clock rate comparison chart of proposed and exiting FFT and IFFT architecture for FPGA technology. The proposed two parallel MDC FFT architecture has 14.7% higher clock rate than the existing SDF FFT architecture and proposed two parallel MDC IFFT architecture has 14.2% higher clock rate than the existing SDF IFFT architecture as shown in Fig. 13.

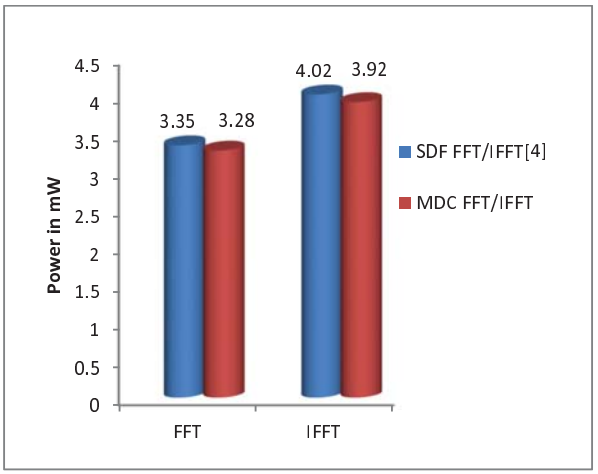

Fig. 14 Power Consumption Comparison chart of FFT/IFFT for FPGA Technology

Fig. 14 shows the power consumption comparison chart of proposed and exiting FFT and IFFT architecture for FPGA technology. The proposed two parallel MDC FFT architecture consume 2.08% less power than the existing SDF FFT architecture and proposed two parallel MDC IFFT architecture consume 2.4% less power than the existing SDF IFFT architecture as shown in Fig. 14.

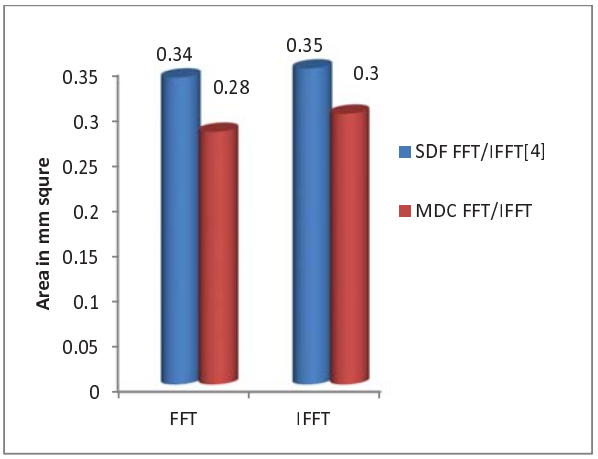

Fig. 15 Core area Comparison chart of FFT/IFFT for ASIC Technology

Fig. 15 shows the core area comparison chart of proposed and exiting FFT and IFFT architecture for ASIC technology. The proposed two parallel MDC FFT architecture occupies 17.6% less core area than the existing SDF FFT architecture and proposed two parallel MDC IFFT architecture occupies

14.2% less core area than the existing SDF IFFT architecture as shown in Fig. 15.

Fig. 16 Power Consumption Comparison chart of FFT/IFFT for ASIC Technology

Fig. 16 shows the power consumption comparison chart of proposed and exiting FFT and IFFT architecture for ASIC technology. The proposed two parallel MDC FFT architecture consume 4.06% less power than the SDF FFT architecture and proposed two parallel MDC IFFT architecture consume 4.1% less power than the existing SDF IFFT architecture as shown in Fig. 16.

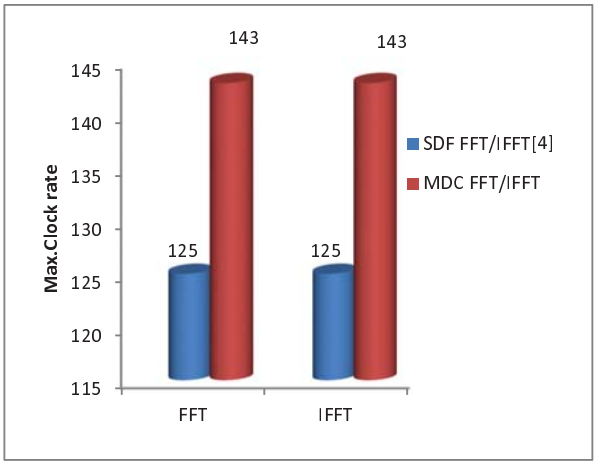

Fig. 17 Maximum Clock Rate Comparison chart of FFT/IFFT for ASIC Technology

Fig. 17 shows the maximum clock rate comparison chart of proposed and exiting FFT and IFFT architecture for ASIC technology. The proposed two parallel MDC FFT architecture has 13% higher clock rate than the existing SDF FFT architecture and proposed two parallel MDC IFFT architecture has 13% higher clock rate than the existing SDF IFFT architecture as shown in Fig. 17.

V. Conclusion

An innovative methodology to develop the FFT and IFFT architectures for radix2^2 FFT/IFFT algorithm is proposed. The projected approach can be used to develop parallel architectures of any arbitrary parallelism. The efficient modified scaling free CORDIC based complex multiplier is used to reduce power consumption during the complex multiplication. The proposed architecture performs four point FFT operations without complex multiplication. The proposed design has higher clock rate due to the ability of processing two input samples in parallel. Therefore, the proposed design consumes less area, less power and gives high throughput than the reference design. The chip area and power consumption can be reduced up to 18%,4% respectively in the projected 2-parallel architectures. The maximum clock rate can be increased up to 13% in the projected 2-parallel architecture. The design is suitable for applications in implantable or portable devices due to their less area and low power consumption. Future work will be focused towards increase the parallelism of the design.

References

[1] S. He and M. Torkelson, "A new approach to pipeline FFT processor," in *Proc. of IPPS*, 1996, pp. 766–770.

[2] J. Lee, H. Lee, S. I. Cho, and S. S. Choi, "A high-speed two parallel radix-2^4 FFT/IFFT processor for MB-OFDM UWB systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 4719–4722.

[3] M. Ayinala, M. Brown, and K. K. Parhi, "Pipelined parallel FFT architectures via folding transformation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 6, pp. 1068–1081, Jun. 2012.

[4] S. He and M. Torkelson, "Design and implementation of a 1024-point pipeline FFT processor," in *Proc. IEEE Custom Integr. Circuits Conf.*, 1998, pp. 131–134.

[5] L. Yang, K. Zhang, H. Liu, J. Huang, and S. Huang, "An efficient locally pipelined FFT processor," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 7, pp. 585–589, Jul. 2006.

[6] A. M. Despain, "Fourier transform computers using CORDIC iterations," *IEEE Trans. Comput.*, vol. C-23, pp. 993–1001, Oct. 1974.

[7] M. A. Sánchez, M. Garrido, M. L. López, and J. Grajal, "Implementing FFT-based digital channelized receivers on FPGA platforms," *IEEETrans. Aerosp. Electron. Syst.*, vol. 44, no. 4, pp. 1567–1585, Oct. 2008.

[8] A. Cortés, I. Vélez, and J. F. Sevillano, "Radix r ^{k} FFTs: Matricial representation and SDC/SDF pipeline implementation," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2824–2839, Jul. 2009.

[9] H. L. Groginsky and G. A.Works, "A pipeline fast Fourier transform," *IEEE Trans. Comput.*, vol. C-19, no. 11, pp. 1015–1019, Oct. 1970.

[10] E. H. Wold and A. M. Despain, "Pipeline and parallel-pipeline FFT processors for VLSI implementation," *IEEE Trans. Comput.*, vol. C-33, no. 5, pp. 414–426, May 1984.

[11] S.-N. Tang, J.-W. Tsai, and T.-Y. Chang, "A 2.4-GS/s FFT processor for OFDM-based WPAN applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 6, pp. 451–455, Jun. 2010.

[12] H. Liu and H. Lee, "A high performance four-parallel 128/64-point radix-2^4 FFT/IFFT processor for MIMO-OFDM systems," in *Proc. EEE Asia Pacific Conf. Circuits Syst.*, 2008, pp. 834–837.

[13] L. Liu, J. Ren, X. Wang, and F. Ye, "Design of low-power, 1 GS/s throughput FFT processor for MIMO-OFDM UWB communication system," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2007, pp. 2594–2597.

[14] J. Lee, H. Lee, S. I. Cho, and S.-S. Choi, "A high-speed, low-Complexity radix-2^4 FFT processor for MB-OFDM UWB systems," in *Proc. IEEE* Int.Symp. Circuits Syst., 2006, pp. 210–213.

[15] N. Li and N. P. van der Meijs, "A radix 2^2 based parallel pipeline FFT processor for MB-OFDM UWB system," in *Proc. IEEE Int. SOC Conf.*,2009, pp. 383–386.

[16] S.-I. Cho, K.-M. Kang, and S.-S. Choi, "Implemention of 128-pointfast Fourier transform processor for UWB systems," in Proc. Int.Wirel. Commun. Mobile Comput. Conf., 2008, pp. 210–213.

[17] W. Xudong and L. Yu, "Special-purpose computer for 64-point FFT based on FPGA," in Proc. Int. Conf. Wirel. Commun. Signal Process., 2009, pp. 1–3.

[18] Y. Chen, Y. Tsao, Y. Wei, C. Lin, and C. Lee, "An indexed-scaling pipelined FFT processor for OFDM-basedWPAN applications," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 55, no. 2, pp. 146–150, Feb. 2008.
[19] Pushparaj, B and Paramasivam, C., "High performance and low power modified radix-2^5 FFT architecture for high rate WPAN application," in Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT), 2013 International Conference on, vol., no., pp.1-4, 7-9 Jan. 2013.
[20] C. Cheng and K.K. Parhi, "High-throughputVLSI architecture for FFT computation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 10, pp. 863–867, Oct. 2007.
[21] J. A. Johnston, "Parallel pipeline fast Fourier transformer," *IEE Proc. F Commun. Radar Signal Process.*, vol. 130, no. 6, pp. 564–572, Oct. 1983.
[22] B. Gold and T. Bially, "Parallelism in fast Fourier transform hardware," *IEEE Trans. Audio Electroacoust.*, vol. 21, no. 1, pp. 5–16, Feb. 1973.
[23] E. E. Swartzlander, W. K. W. Young, and S. J. Joseph, "A radix 4 delay commutator for fast Fourier transform processor implementation," *IEEE J. Solid-State Circuits*, vol. 19, no. 5, pp. 702–709, Oct.1984.
[24] J. H. McClellan and R. J. Purdy, "Applications of digital signal processing to radar," in Applications of Digital Signal Processing. Englewood Cliffs, NJ: Prentice-Hall, 1978, ch. 5.
[25] E. E. Swartzlander, W. K. W. Young, and S. J. Joseph, "A radix-4 delay commutator for fast Fourier transform processor implementation," *IEEE J. Solid-State Circuits*, vol. SC-19, no. 5, pp. 702–709, Oct. 1984.
[26] E. E. Swartzlander, V. K. Jain, and H. Hikawa, "A radix-8 wafer scale FFT processor," *J. VLSI Signal Process.*, vol. 4, no. 2/3, pp. 165–176, May 1992.
[27] G. Bi and E. V.Jones, "A pipelined FFT processor for word sequential data," *IEEE Trans. Acoust., Speech, Signal Processing,* 37(12):1982-1985, Dec. 1989.
[28] E. Bidet, D. Castelain, C. Joanblanq, and P. Stenn, "A fast singlechip implementation of 8192 complex point FFT," *IEEE J. Solid-State Circuits,* 30(3):300-305, Mar. 1995.
[29] Erdal Oruklu., Xin Xiao., Jafar Saniie, "Reduced memory and Low Power architecture for CORDIC-based FFT Processor," Journal of Signal Processing, volume 66 issue2 (2012) 129-134.
[30] Malashri, A and Paramasivam, C, "Low power and memory efficient FFT architecture using modified CORDIC algorithm," in Information Communication and Embedded Systems (ICICES), 2013 International Conference on, vol., no., pp.1041-1046, 21-22 Feb. 2013.
[31] Paramasivam, C and Dr.K.B. Jayanthi, "Modified Scaling-Free CORDIC based in Place FFT and IFFT architecture for Radix 2^2 Algorithm," in Innovations in Computer Science and Information Technology (ICICSIT), 2015 International Conference on, pp 114-118,28-29 August,2015.
[32] A. V. Oppenheim, R.W. Schafer, and J.R. Buck, *Discrete-Time Signal* Processing, 2nd ed. Englewood Cliffs, NJ, USA: Prentice Hall, 1998.