Efficient FPGA implementation of FFT/IFFT Processor

Ahmed Saeed, M. Elbably, G. Abdelfadeel, and M. I. Eladawy

Abstract—The Fast Fourier Transform (FFT) and its inverse (IFFT) are very important algorithms in signal processing, software-defined radio, and the most promising modulation technique; Orthogonal Frequency Division Multiplexing (OFDM). This paper explains the implementation of radix-2^2 single-path delay feedback pipelined FFT/IFFT processor. This attractive architecture has the same multiplicative complexity as radix-4 algorithm, but retains the simple butterfly structure of radix-2 algorithm. The implementation was made on a Field Programmable Gate Array (FPGA) because it can achieve higher computing speed than digital signal processors, and also can achieve cost effectively ASIC-like performance with lower development time, and risks. The processor has been developed using hardware description language VHDL on an Xilinx xc5vsx35t and simulated up to 465MHz and exhibited execution time of 0.135µS for transformation length 256-point. This results show that the processor achieves higher throughput and lower area and latency.

***Keywords*—**FFT, FPGAs, Radix-2^2 single-path delay feedback (R2^2SDF), Pipelining, VHDL.

I. INTRODUCTION

THE Fast Fourier Transform (FFT) has become almost ubiquitous and most important in high speed signal processing. Using this transform, signals can be moved to the frequency domain where filtering and correlation can be performed with fewer operations [1].

It has been widely applied in the analysis and implementation of digital communication systems and television terrestrial broadcasting systems, such as the xDSL (de)modulator, phase correlation system, mobile receiver, as well as fault characterization and classification. [2], [3], [4].

When considering the alternate implementations, the FFT/IFFT algorithm should be chosen to consider the execution speed, hardware complexity, and flexibility and precision. Nevertheless, for real time systems the execution speed is the main concern [5], [6], [7]. Several architectures have been proposed over the last 3 decades like: single-memory architecture, dual-memory architecture, cached-memory architecture, array architecture, and pipelined architecture [6]. Pipelined architectures characterized by real-time, non-stopping processing and present smaller latency with low power consumption [8] which makes them suitable for most application.

The pipelined architectures can be classified into two types: single-path architectures and multi-path architectures. Several single-path architectures have been proposed: Radix-2 single-path delay feedback, Radix-4 single-path delay feedback, Radix-2^2 single-path delay feedback, Radix-2^4 single-path delay feedback, Split-Radix single-path delay feedback, and Radix-4 single-path delay commutator. The multi-path architectures: Radix-2 multi-path delay commutator, Radix-4 multi-path delay commutator, Split-Radix multi-path delay commutator, and Mixed-Radix multi-path delay commutator.

The observation made on the listed architectures reveals that the delay feedback architecture is more efficient than the corresponding delay commutator in terms of memory utilization and Radix-2^2 has simpler butterfly and higher multiplier utilization [3], [5], [9]. This makes Radix-2^2 single-path delay feedback an attractive architecture for implementation.

Classical implementation of the FFT/IFFT algorithm, with digital signal processors (DSPs), requires a sequential algorithm. This slows down the execution time. On the other hand, the modern programmable circuits, like an FPGA, utilizes a tens of thousands of lists and triggers during operation, resulting of parallel processing system, putting the FPGA computing speed at a significant advantage over DSPs [10], [11], [12], [13].

This paper presents the implementation of Radix-2^2 single-path delay feedback pipelined FFT/IFFT processor on an FPGA. We will focus on the implementation of FFT as the IFFT is computed by conjugating the twiddle factors of the corresponding forward FFT. The processor is compared to other implementations based on the maximum operating frequency, execution speed and power consumption. The results expose that our design achieves higher operating frequency and appropriate execution speed.

The paper is structured as follows. The Radix-2^2 FFT algorithm is illustrated in Section II. In Section III, the implementation of Radix-2^2 Algorithm by FPGA will be debated. The synthesis results and consumed resources are revealed in Section IV. At last, the concise statements remark this paper.

Manuscript received June 30, 2009.

Ahmed Saeed is with Electrical Engineering Department, Future University in Egypt, Helwan, Cairo, Egypt (phone: +2-010-2052-382; fax: +202-2618-6111; e-mail: asaeed@fue.edu.eg).

M. Elbably, is with the Electronics, Communication, and Computer Department, Helwan University, Helwan, Egypt (e-mail: elbably_55@yahoo.com).

G. Abdelfadeel, is with the Electronics, Communication, and Computer Department, Helwan University, Helwan, Egypt (e-mail: gam_hel@yahoo.com).

M. I. Eladawy, is with the Electronics, Communication, and Computer Department, Helwan University, Helwan, Egypt (e-mail: mohamed@eladawy.com).

II. RADIX-2^2 FFT ALGORITHM

The Discrete Fourier Transform (DFT) *X(k)*, *k=0, 1, ...,N-1* of a sequence *x(n), n=0, 1,..., N-1* is defined as:

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

Where *N* is the transform size, $W_N = e^{-j2\pi/N}$, and $j=\sqrt{-1}$

According to the decomposition method of [9] that done by substituting with

$$n=\left\langle \frac{N}{2}n_1+\frac{N}{4}n_2+n_3\right\rangle_N$$
$$k=\left\langle k_1+2k_2+4k_3\right\rangle_N$$

This yield

$$X(k_1+2k_2+4k_3)=\sum_{n_3=0}^{\frac{N}{4}-1}\left[H(k_1,k_2,n_3).W_N^{n_3(k_1+2k_2)}\right]W_{\frac{N}{4}}^{n_3k_3} \tag{2}$$

Where

$$\begin{aligned}H(k_1,k_2,n_3)=&\left[x(n_3)+(-1)^{k_1}.x\left(n_3+\frac{N}{2}\right)\right]\\&+(-j)^{(k_1+2k_2)}\left[x\left(n_3+\frac{N}{4}\right)+(-1)^{k_1}.x\left(n_3+\frac{3N}{4}\right)\right]\end{aligned} \tag{3}$$

After this simplification, we have a set of four DFTs of length *N/4*.

Each term in equation (3) represents a Radix-2 butterfly (BFI), and the whole equation also represents Radix-2 butterfly (BFII) with trivial multiplication by *-j*.

Fig. 1 shows an example of *N=16-points* Radix-2^2 decimation in frequency (DIF)—the method used for the pipelined, streaming I/O architecture —FFT algorithm. It is to be noted that the inputs are in normal order whereas the outputs are in permuted (digit-reversed) order. The pentagons between BFI and BFII represent the trivial multiplication by –*j*. After these two butterflies, full twiddle factor multipliers (TFM) are required to compute the multiplication by the twiddle factor $W^{n_3(K_1+2k_2)}$.

Fig. 3 shows the block diagram for Radix-2^2 *N*-point FFT processor. The *N*-point FFT processor has *log₄ (N)*-stages with *i* is the stage number.

A typical stage consists of BFI, BFII, delay-feedback, ROM, and TFM. A *log₂ (N)* counter is used to control the processor.

The structure of the last stage is different according to the size of FFT; if *N* is power of 2, the last stage is composed of BFI only. But if *N* is power of 4, the last stage composed of BFI and BFII.

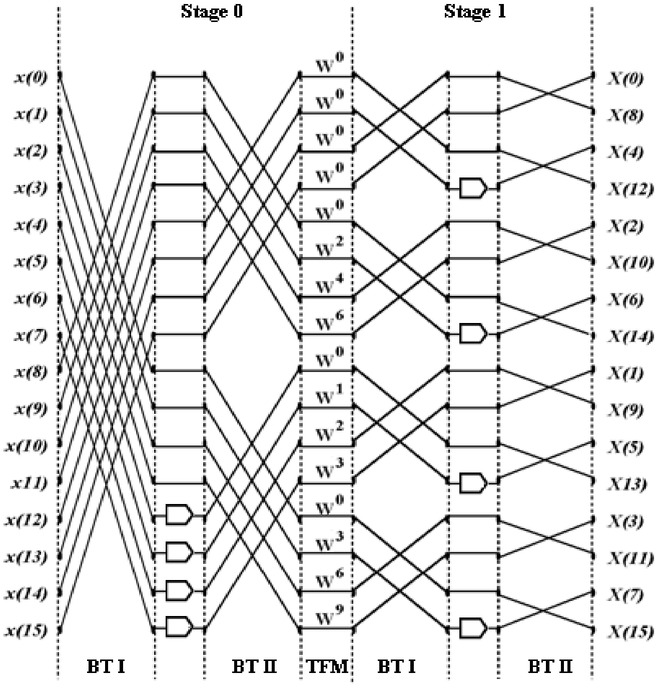


Fig. 1 Flow graph of Radix2^2 DIF for N=16 FFT algorithm

III. IMPLEMENTATION OF RADIX-2^2 BY FPGA

Concerning the FPGA implementation, the selection of target FPGA should consider the required resources for the pipelined architecture to be implemented (for *N=256*): complex multipliers, complex adders/subtractors for BFI and BFII, registers and memory for delay feedback and pipelining, ROM for storing the twiddle factor, and the control unit.

A. BFI Structure

The detailed structure of BFI is shown in Fig. 2. The A input comes from the previous component, TFM. The B output fed to the next component, normally BFII. In first *$N/2^{i+1}$* cycles, multiplexors direct the input data to the feedback registers until they are filled (position “0”). On next *$N/2^{i+1}$* cycles, the multiplexors select the output of the adders/subtractors (position “1”), the butterfly computes a 2-point DFT with incoming data and the data stored in the feedback registers.

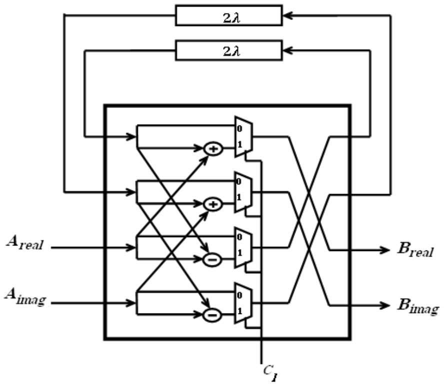


Fig. 2 BFI Structure

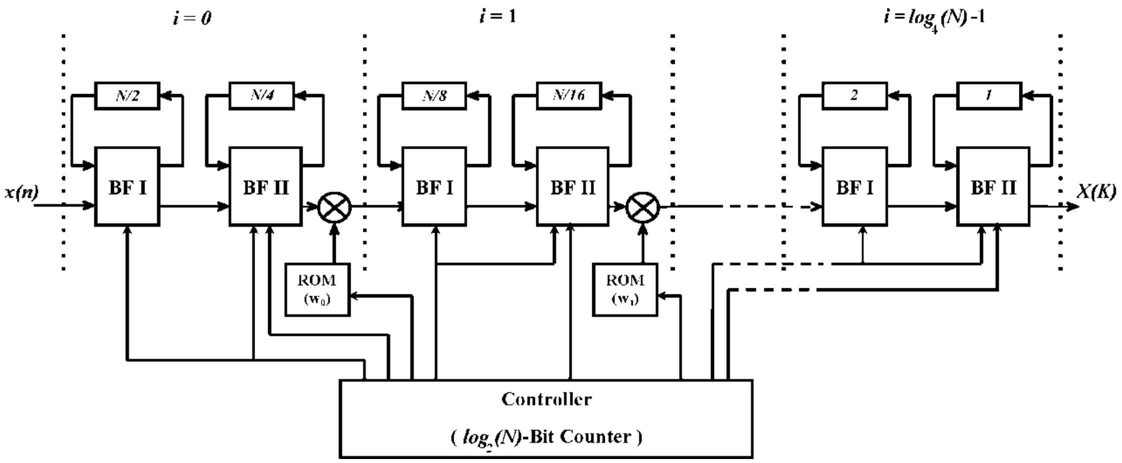


Fig. 3 Block diagram for Radix-2^2 N-points FFT processor

B. BFII Structure

The detailed structure of BFII is shown in Fig. 4. The B input comes from the previous component, BFI. The Z output fed to the next component, normally TFM.

In first $N/2^{i+2}$ cycles, multiplexors direct the input data to the feedback registers until they are filled (position "0").

In next $N/2^{i+2}$ cycles, the multiplexors select the output of the adders/subtractors (position "1"), the butterfly computes a 2-point DFT with incoming data and the data stored in the feedback registers.

The multiplication by $-j$ involves real-imaginary swapping and sign inversion. The real-imaginary swapping is handled by the multiplexors MUXim, and the sign inversion is handled by switching the adding-subtracting operations by mean of MUXsg.

When there is a need for multiplication by $-j$, all multiplexors switches to position "1", the real-imaginary data are swapped and the adding-subtracting operations are switched. Fig. 5 shows the sign inversion structure.

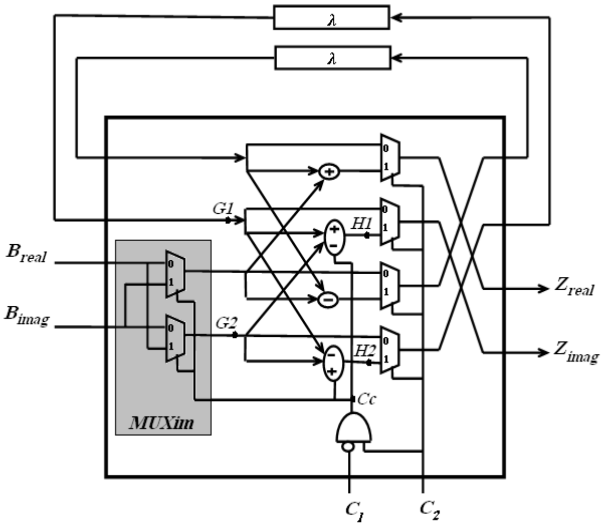


Fig. 4 BFII Structure

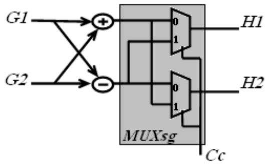


Fig. 5 Sign-inversion structure

The adders and subtractors in BFI and BFII are fully-pipelined (Fig. 6) and followed by divide-by-2 scaling and rounding. The divide-by-2 scaling is used in order to not lose any precision where the word-length imply successive growth as the data goes through add/subtract and multiply operations (i.e. constrain the dynamic range of the variables to a certain word-length). If scaling is insufficient, a butterfly output may grow beyond the dynamic range and cause an overflow. Rounding has been also applied to reduce the scaling errors [6], [17].

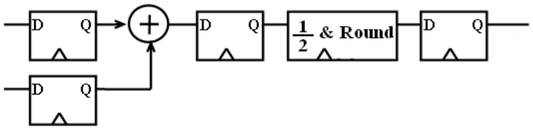


Fig. 5 Pipelined adder with divide-by-2 and round

C. TFM Structure

A six clock cycle fully-pipelined complex-multiplier has been implemented to multiply the twiddle factor by the output of BFII.

According to Equation (4), the algorithm of multiplying the twiddle factor ($c + js$) by BFII output ($Z_r + jZ_i$) uses four multipliers, one adder, and one subtractor. The structure of TFM shown in Fig. 6.

$$\left(Z_r + jZ_i\right)\cdot\left(c+js\right)=\left(Z_r\cdot c - Z_i\cdot s\right)+j\left(Z_i\cdot c+Z_r\cdot s\right) \qquad (4)$$

Twiddle factor generator is a key component in IFFT/FFT computation. There exist many popular generation techniques for twiddle factor: COordinate Rotation DIgital Computer

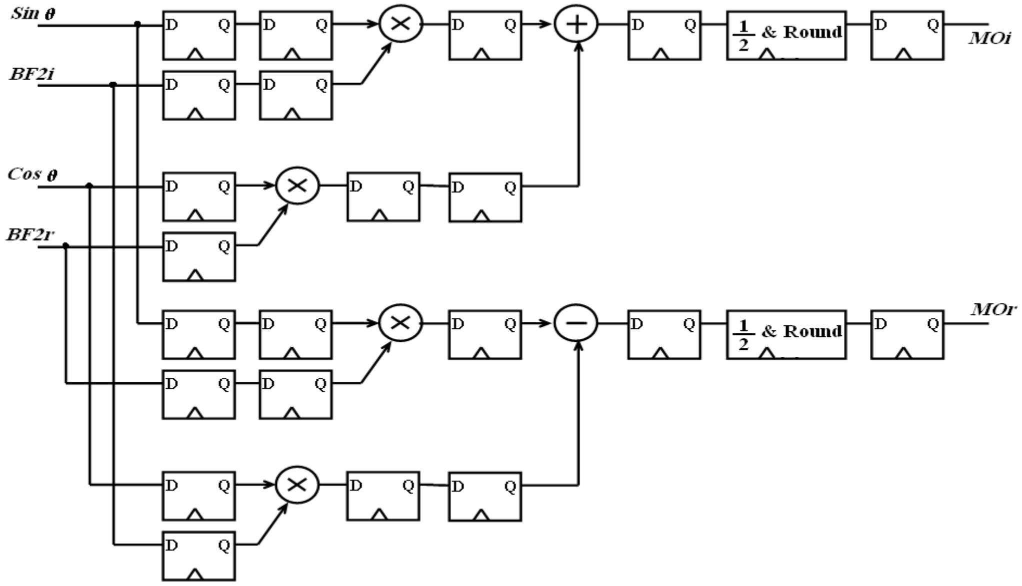


Fig. 6 Structure of Fully-pipelined TFM

(CORDIC) algorithm, pipelined-CORDIC algorithm, polynomial-based approach, ROM-based scheme, and the recursive function generators. For small lengths such as 64 up to 512, ROM-based is a better choice [14].

The twiddle factors are generated using MATLAB according to equation (5), converted to fixed point, and then stored in ROM.

The twiddle factor at the i^{th}-stage, with $i = 0, 1, \ldots, (\log_4 N)-2$ is given by $W_i = \{u_x\};\ x = 0,1,\ldots, N/2^{2i}$ with $u_x = e^{-j2\pi v/N}$

$$v = \begin{cases} 0, & 0 \le x < a \\ 2^{2i+1} \cdot (x-a), & a \le x < 2a \\ 2^{2i} \cdot (x-2a), & 2a \le x < 3a \\ 3 \cdot 2^{2i} \cdot (x-3a), & 3a \le x < 4a \end{cases} \tag{5}$$

With $a = \frac{N}{2^{2+2i}}$ [6].

D. Delay-feedback Structure

In order to reuse the existing hardware, the delay feedback is used. The delay feedback mechanism provides a solution where the first input to the butterfly is delayed until the second input is presented, after which the calculation can proceed. This done by accepting part of the data stream into the butterfly elements, but instead of computing on the block, it is redirected to a feedback delay line by mean of multiplexers. By the time, the data appears again at the input of the butterfly. First-in First-out shift register is used to implement the delay-feedback.

The feedback delay at the i^{th} stage is given by:

$$\lambda = \frac{N}{2^{2(1+i)}} \tag{6}$$

E. Control Unit

Radix-2^2 control unit is very simple. A *log₂(N)* counter is used to switch the butterflies between modes. It also used as address to ROMs in order to pick the twiddle factors.

The DSP48E are a DSP slices introduced by Xilinx and available in Virtex-4 and Virtex-5 FPGAs. It has been used to implement the Adders, subtractors, control unit, and complex multipliers. DSP48E slices provide reduced overall power consumption, increased maximum frequency, and reduced set-up plus clock-to-out time. It can implement 25x18-bit multipliers, with add, subtract, accumulate, and bit-wise logic features [15]. These slices have optional connections between the multiplier and adder/subtractor units inside it; this saves the routing resources, and increases performance, because all connections are in the DSP slice. In order to save the registers in the FPGA, the pipelining of multipliers and Adder/subtractor units was made also by mean of DSP48E.

We chose the two's complement format to represent the signals in digital domain due to its ability to handle negative numbers inherently without an extra sign bit. Positive and negative numbers can be distinguished by the most significant bit of the given number.

Fig. 7 shows the effect of the data word length for FFT length *N=256*. The figure shows the Signal-to-Noise Ratio (SNR) is linear increment of 6 dB with data word length. Using this figure, the word length can be selected to guarantee the required SNR [16]. The data word length is chosen to be 16-bits in order to matches the characteristics of DSP48E slices. After choosing the data word length, the effect of twiddle factor quantization can be studied. Fig. 8 shows the effect of twiddle factor word length on the SNR. Note from figure that the SNR will remain constant at 46 dB even with the increase in the word length for greater than 12-bits.

The word length of twiddle factor is chosen to be 11-bits to guarantee a SNR of 46 dB.

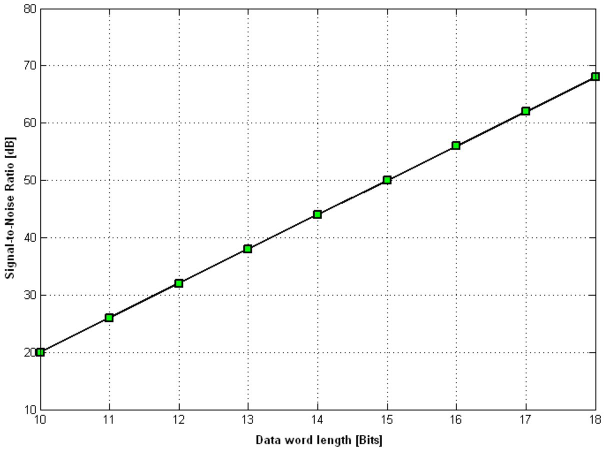


Fig. 7 SNR of FFT (N=256, no twiddle factor quantization)

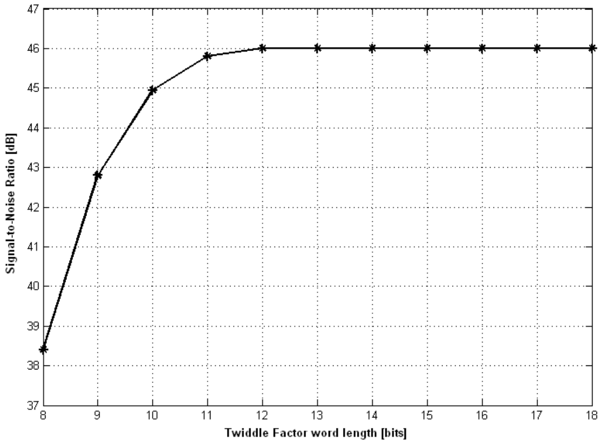


Fig. 8 Effect of twiddle factor word length on SNR

IV. RESULTS

The FFT processor was described with hardware description language VHDL as fixed-point arithmetic and synthesized with XST tool in Xilinx ISE version 10.1 on Xilinx *xc5vsx35t* FPGA chip, the high-performance signal processing applications chip with advanced serial connectivity, and simulated using ModelSim Xilinx Edition (MXE III). The top-level design shown in Fig. 9, the Xn is input data (16-bit real and 16-bit imaginary), Xk output data (16-bit real and 16-bit imaginary), synchronous reset; rst, clock, chip enable, start, busy, and finish.

The synthesis tool has allocated the following resources (Fig. 10): 838 slice registers (3%), 1058 lookup table (4%) 256 of them used as memory element, and 44 specific feature elements (DSP48E) (22%).

The results of the synthesis tool and the timing analysis using the ModelSim simulator indicate a maximum operating frequency of 465 MHz; this provides an execution time of a 256 complex data points transform in 0.135µS.

A comparison done between Xilinx FFT core, FFT processor in [6], and our processor. Our Radix-2^2 processor achieves highest operating frequency of all the processors implemented with FPGA of Table I.

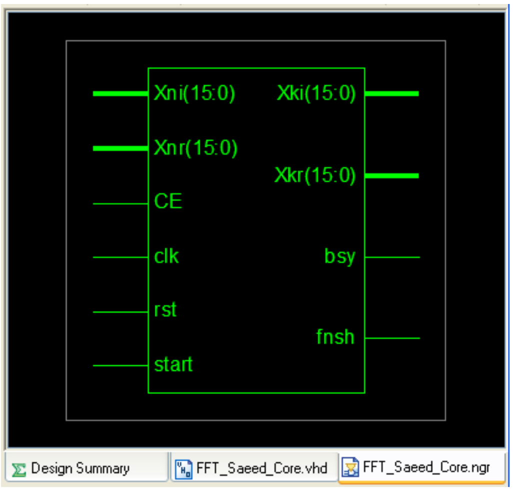


Fig. 9 Top level

Table I
Comparison with other FFT Processors (N=256, Data-Bits=16)

	Xilinx Core[a] [17]	[6]	Our processor
Execution time [µS]	0.59	0.123	0.135
Frequency [MHz]	432	350	465
FPGA Family	Virtex 5	Altera-Stratix II	Virtex 5
Device	5VSX35T	EP2S15F672C3	5VSX35T

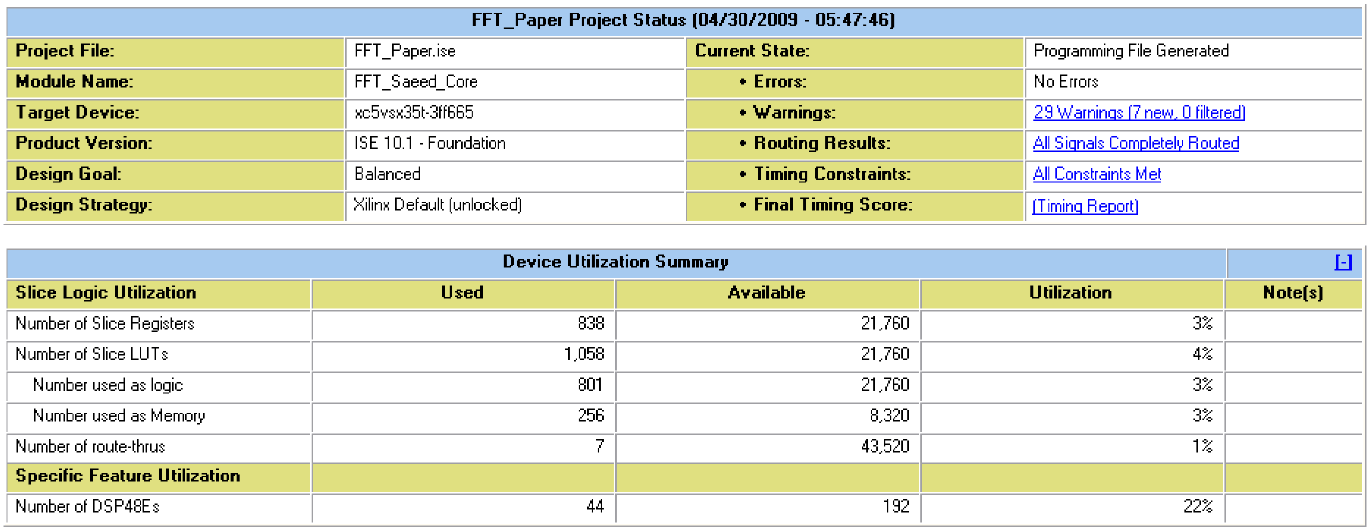
The power consumed by the implemented processor was estimated by XPower analyzer, Xilinx tool, after the place and route process. The processor consumes a total power of 0.479 watt; 0.426 watt quiescent power and 0.053 dynamic power. Table II shows the details of power report.

Comparison of our processor with other FFT chips like special purpose chips, board-level products, soft/synthesizable processors, and programmable DSP chips in terms of power consumption, reveals that our processor achieve appropriate power.

Table II
Estimated Power Summary (Ambient temp = 25C)

Name	Power [W]	Used	Total Available	Utilization [%]
Logic	0.006	1027	21760	4.9
Signals	0.019	2059	----	----
DSP	0.028	44	192	22.9
Total Quiescent Power	0.426			
Total Dynamic Power	0.053			
Total Power	0.479			

[a] The data concerning Xilinx core is extracted from [6].



FFT_Paper Project Status (04/30/2009 - 05:47:46)			
Project File:	FFT_Paper.ise	**Current State:**	Programming File Generated
Module Name:	FFT_Saeed_Core	• **Errors:**	No Errors
Target Device:	xc5vsx35t-3ff665	• **Warnings:**	29 Warnings (7 new, 0 filtered)
Product Version:	ISE 10.1 - Foundation	• **Routing Results:**	All Signals Completely Routed
Design Goal:	Balanced	• **Timing Constraints:**	All Constraints Met
Design Strategy:	Xilinx Default (unlocked)	• **Final Timing Score:**	(Timing Report)

Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	838	21,760	3%	
Number of Slice LUTs	1,058	21,760	4%	
Number used as logic	801	21,760	3%	
Number used as Memory	256	8,320	3%	
Number of route-thrus	7	43,520	1%	
Specific Feature Utilization				
Number of DSP48Es	44	192	22%	

Fig. 10 Design Summary

Fig. 11 shows the components and routes allocated on the FPGA. The areas surrounded by the squares are the clock areas. Note that the placer and router placed the components in their positions in order to obey the designer constraints found in the user constraints file (ucf) that describing the used pins and the required clock period (i.e. operating frequency). The device utilization can be observed from the figure to verify the results obtained in fig. 10.

As mention above, in order to obtain IFFT processor, the same design might be used but with conjugated twiddle factor. One important feature of the FFT algorithm is that the choosing the input order and output order carefully can lead to significant memory and latency savings [18]. For example, consider where the input to the FFT is in normal order and the output is in digit-reversed order, if the IFFT processor is configured to accept digit-reversed inputs, it will produce normal order, which provides a saving of *N* complex memory words, and a latency saving of *N* clock cycles.

V. Conclusion and Future Work

The design and implementation of Radix2^2 single-path delay feedback pipelined FFT/IFFT processor on an FPGA has been presented. The description was made by VHDL in Xilinx ISE on Virtex-5 family and the functionality was verified by ModelSim Xilinx Edition. The outputs from the VHDL described architecture are validated against the standard FFT in Matlab.

The multipliers, Adder/subtractor units, control unit, and their pipelining were implemented by efficient inferring the DSP48E Blocks in order to obtain a faster and low power design. The data and twiddle factor word length were chosen to achieve an acceptable signal-to-noise ratio and also to match the feature of DSP48E slices. The design can maintain the SNR since scaling and rounding are applied in all pipeline stages.

The power analysis shows that the processor consumes appropriate power with respect to the transformation length and operating frequency.

The synthesis and simulation of the processor indicates the execution of 256 data points R2^2SDF in 0.135µS with maximum operating frequency of 465MHz. The comparison with other FFT processors reveals the power of our processor.

The implemented design gives an easy way to increase the number of points of FFT as well as IFFT by imposing simple modification. Future work includes the development of complete OFDM system and upgrade it to a multiple input multiple outputs (MIMO) system by using high density FPGA device.

Acknowledgment

Ahmed Saeed thanks Eng. Heba Sadek, with whom he discussed a lot of FPGA implementation issues, whose comments substantially improved this paper. Her help, advice and supporting during the period this research took place is greatly acknowledged.

References

[1] P. Jackson, C. Chan, C. Rader, J. Scalera, and M. Vai. “A systolic FFT architecture for real time FPGA systems” In High Performance Embedded Computing Conference (HPEC04), Sept. 2004.

[2] C. Lin, Y. Yu, and L. Van, "A low-power 64-point FFT/IFFT design for IEEE 802.11a WLAN application" in Proc. International Symposium on circuit and systems, 2006, pp. 4523-4526.

[3] L. Yang, K. Zhang, H. Liu, J. Huang, and S. Huang, "An Efficient Locally Pipelined FFT Processor," IEEE transactions on circuits and systems—II: Express Briefs, VOL. 53, NO. 7, JULY 2006, pp. 585-589.

[4] E. E Ngu, K. Ramar and R. Montano, V. Cooray “Fault characterization and classification using wavelet and Fast Fourier Transform,” WSEAS transaction on signal processing, Volume 4, Issue 7, July 2008, pp. 398-408. Available: www.wseas.us/e-library/transactions/signal/2008/27-735.pdf

[5] A. Cortés, I. Vélez, I. Zalbide, A. Irizar, and J. F. Sevillano, “An FFT Core for DVB-T/DVB-H Receivers,” VLSI Design, vol. 2008, Article ID 610420, 9-pages , 2008.

[6] Jesús García1, Juan A. Michell, Gustavo Ruiz, and Angel M. Burón, "FPGA realization of a Split Radix FFT processor," Proc. of SPIE. Microtechnologies for the New Millennium, vol. 6590, 2007, pp. 65900P-1 to 65900P-11.

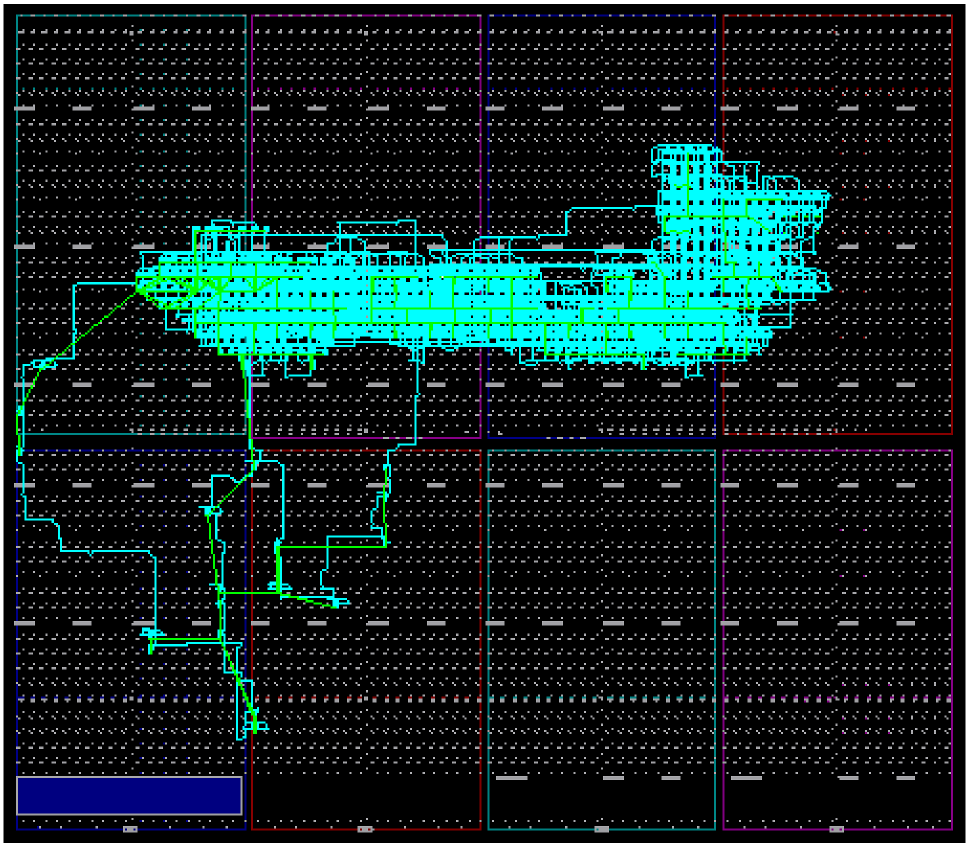


Fig. 11 The Routed Design

[7] Ioan Jivet, Alin Brindusescu, and Ivan Bogdanov “Image Contrast Enhancement using Morphological Decomposition by Reconstruction,” WSEAS transactions on circuits and systems, Volume 7, Issue 8, August 2008, pp. 822-831. Available: http://www.wseas.us/e-library/transactions/circuits/2008/27-1425.pdf

[8] T. Sansaloni, A. P´erez-Pascual, V. Torres, and J. Valls, “Efficient pipeline FFT processors for WLAN MIMO-OFDM systems,” Electronics Letters, vol. 41, no. 19, 2005, pp. 1043–1044.

[9] S. He and M. Torkelson, “A new approach to pipeline FFT processor,” in Proc. 10th Int. Parallel Processing Symp., 1996, pp. 766–770.

[10] Zhijian Sun, Xuemei Liu, and Zhongxing Ji, "The Design of Radix-4 FFT by FPGA," International Symposium on Intelligent Information Technology Application Workshops, 2008, pp.765-768.

[11] J. Mar, Y. Lin, T. Lung, and T. Wei, "Realization of OFDM Modulator and Demodulator for DSRC Vehicular Communication System Using FPGA Chip," International Symposium on Intelligent Signal Processing and Communications (ISPACS'06), 2006, pp. 477-480.

[12] M. Nilsson, “FFT, Realization and Implementation in FPGA”, Master thesis, Ericsson Microwave Systems AB / Griffith University, 2000 – 2001.

[13] N.Balaji, K.Subba Rao, and M.Srinivasa Rao. 2009. " FPGA Implementation of Ternary Pulse Compression Sequences with Superior Merit Factors," NAUN international Journal of Circuits, systems and signal processing. Volume 2, issue 3, pp. 47-54. Available: http://www.naun.org/journals/circuitssystemssignal/cssp-98.pdf

[14] Chi, J.C., and S. Chen, "An efficient FFT twiddle factor generator," in Proc. European Signal Process. Conf., 2004, pp. 1533–1536.

[15] User Guide. 2009, January. “Virtex-5 FPGA XtremeDSP Design Considerations,” Version 3.3, Xilinx Inc. Available: http://www.xilinx.com

[16] Ainhoa Cortés, Igone Vélez, Juan F. Sevillano, and Andoni Irizar, "An Approach to Simplify the Design of IFFT/FFT Cores for OFDM Systems," IEEE Transactions on Consumer Electronics, No. 1, February 2006, pp. 26-32.

[17] Datasheet. 2008, September. “Xilinx LogiCore Fast Fourier Transform,” Version 4.1, Xilinx Inc. Available: http://www.xilinx.com

[18] User Guide. 2008, November. “FFT MegaCore Function,” Version 8.1, Altera Corporation. Available: http://www.Altera.com

Ahmed Saeed was born in Cairo, Egypt, in September 1980. He obtained his BSc degree in Communications and Electronics Engineering from Helwan University, Egypt in 2004 with rank, second. He now is working in the MSc degree at Helwan University. From 2004 to 2007 he was Teaching Assistant in the department of Electronics Engineering and Communication Technology at MAM for Engineering & Technology. Currently, he is Teaching Assistant in Electrical Engineering Department, Future University in Egypt. He also working as part-time Trainer with many training companies in the field of digital system design with VHDL and Verilog. His research interests are in implementation of signal processing functions and wired/wireless communication systems on FPGA and ASIC. Eng. Ahmed is an IEEE member.

M. Elbably was born in Cairo, Egypt, in April 1955. He received the B.Sc. from Helwan University, M.Sc. from Manchester University, and Ph.D. degree from Brunel University in 1978, 1984, and 1988, respectively, all in Electrical Engineering. From 1978 to 1983 he was an Administrator, and from 1985 to 1988 he was an Assistant Lecturer at Department of Communication and Electronics Engineering, Helwan University, Egypt. From 1991 to1996 he was an Lecturer in Technical College- Riyadh- KSA and part time Lecturer at 6th-October university and MAM for Engineering & Technology.

Currently, he is a Assistant Professor in Department of Communication and Electronics Engineering, Helwan University, Egypt. His research interests are in testing, diagnostic, and fault tolerant techniques of digital systems.

G. Abdelfadeel was born in Cairo, Egypt, in November 1954. He received the B.Sc and M.Sc. in communication Engineering from Helwan University, Egypt, and Ph.D. degree in Computer Communication Networks from Imperial College of Science, Technology and Medicine, London University, London in 1977, 1983, and 1990, respectively.

From 1978 to 1983 he was Maintenance Engineer for Biomedical systems. During this period, he work also as Assistant Researcher. From 1983 to 1984, he was an Assistant Lecturer, from 1992 to1999 he was a Lecturer, all at Department of Communication and Electronics Engineering, Helwan University.

Currently, he is a Professor and the head of Department of Communication and Electronics Engineering, Helwan University. He also working as part time Lecturer at 6th-October university, Egypt and Supervisor of the Diploma graduation's projects at the National Telecommunication Institute, NTI, Egypt. His research interests are in wireless communications and multiple-access networks.

Mohamed I. Eladawy graduated from the Department of Electrical Engineering, Faculty of Engineering of Assiut University in May 1974; M.Sc. from Cairo University in May 1979; Ph.D. from Connecticut State University, School of Engineering, in May 1984. He worked as an Instructor at the Faculty of Engineering, Helwan University since 1974.

Currently he is a Professor at the Department of Communication and Electronics Engineering and the Vice Dean for Student Affairs in the same faculty. He was working for the general organization for technical and vocational training for 6 years from 1989 to 1995 in Saudi Arabia. His research interests are in signal processing and its medical applications.