Journal of Mathematics and Statistics 5 (4): 241-250, 2009
ISSN 1549-3644

Hardware Implementation of a Genetic Algorithm Based Canonical Singed Digit Multiplierless Fast Fourier Transform Processor for Multiband Orthogonal Frequency Division Multiplexing Ultra Wideband Applications

Mahmud Benhamid and Masuri Bin Othman
VLSI Design Center, Institute of Microengineering and Nanoelectroincs,
National University of Malaysia, 43650 Bangi, Selangor Darul Ehsan, Malaysia

Abstract: Problem statement: Ultra Wide Band (UWB) technology has attracted many researchers' attention due to its advantages and its great potential for future applications. The physical layer standard of Multi-band Orthogonal Frequency Division Multiplexing (MB-OFDM) UWB system is defined by ECMA International. In this standard, the data sampling rate from the analog-to-digital converter to the physical layer is up to 528 M sample sec^{-1}. Therefore, it is a challenge to realize the physical layer especially the components with high computational complexity in Very Large Scale Integration (VLSI) implementation. Fast Fourier Transform (FFT) block which plays an important role in MB-OFDM system is one of these components. Furthermore, the execution time of this module is only 312.5 ns. Therefore, if employing the traditional approach, high power consumption and hardware cost of the processor will be needed to meet the strict specifications of the UWB system. The objective of this study was to design an Application Specific Integrated Circuit (ASIC) FFT processor for this system. The specification was defined from the system analysis and literature research. **Approach:** Based on the algorithm and architecture analysis, a novel Genetic Algorithm (GA) based Canonical Signed Digit (CSD) Multiplier less 128-point FFT processor and its inverse (IFFT) for MB-OFDM UWB systems had been proposed. The proposed pipelined architecture was based on the modified Radix-2^2 algorithm that had same number of multipliers as that of the conventional Radix-2^2. However, the multiplication complexity and the ROM memory needed for storing twiddle factors coefficients could be eliminated by replacing the conventional complex multipliers with a newly proposed GA optimized CSD constant multipliers. The design had been coded in Verilog HDL and targeted Xilinx Virtex-II FPGA series. It was fully implemented and tested on real hardware using Virtex-II FG456 prototype board and logic analyzer. **Results:** From the synthesis reports, the proposed GA optimized CSD constant complex multiplier achieved 79 and 50% equivalent gates and latency efficiency when compared to the conventional complex multiplier. **Conclusion:** As a conclusion, we successfully implemented 128-points FFT/IFFT processor with the proposed architecture that can meet the requirement of MB-OFDM UWB system with higher throughput and less area compared to conventional architecture.

Key words: VLSI, UWB, OFDM, FFT, GA, CSD

INTRODUCTION

The concept of Ultra-Wideband (UWB) was formulated in the early 1960s through research in time-domain electromagnetic and receiver design, both performed primarily by Gerald F. Ross. Through his work, the first UWB communications patent was awarded for the short-pulse receiver which he developed while working for Sperry Rand Corporation[1]. Throughout that time, UWB was referred in broad terms as carrier less or impulse technology. The term UWB was coined in the late 1980s to describe the development, transmission and reception of ultra-short pulses of Radio Frequency (RF) energy. Even though the knowledge has been in existence for over thirty years, UWB technology is an emerging research topic in the wireless communications field for a variety of reasons.

For communication applications, high data rates are possible due to the large number of pulses that can

Corresponding Author: Mahmud Benhamid, VLSI Design Center, Institute of Microengineering and Nanoelectroincs, National University of Malaysia, 43650 Bangi, Selangor Darul Ehsan, Malaysia
Tel: +603-89216009 Fax: +603-89250439

be created in short time duration. Due to its low power spectral density, UWB can be used in military applications that require low probability of detection. Other common uses of UWB are in radar and imaging technologies, where the ability to resolve multipath delay is in the nanosecond range, allowing for finer resolution, whether it is from a target or for an image.

After recognizing the potential advantages of UWB, the Federal Communications Commission (FCC) developed a report to allow UWB as a communications and imaging technology. A UWB definition was created as a signal with a fractional bandwidth greater than 0.2 or which occupies more than 500 MHz of spectrum. The fractional bandwidth is defined as $2(f_H - f_L)/(f_H + f_L)$, where f_H and f_L are the upper and lower frequencies, respectively, measured at -10 dB below the peak emission point. To allow government and industry to conduct UWB testing, frequency spectrum from 3.1-10.6 GHz was allocated for communications use below specified power levels, as shown in Fig. 1 below. For indoor systems, the average output power spectral density is limited to -41.3 dBm MHz^{-1}, which complies with the long standing Part 15 general emission limits to successfully control radio interference[2].

Although the FCC has regulated spectrum and power levels for UWB, there is currently no standard for industry to follow. Discussions have developed on the use of two standards, specifically, MB-OFDM and Direct Sequence Spread Spectrum (DS-SS), which is based on impulse radio technology. The MB OFDM alliance and Motorola (DS-SS) are presently attempting to persuade the IEEE to adopt their respective approach. Each of these schemes has their advantages in a communications system but OFDM are currently getting more attention.

UWB communication systems, which enable one to deliver data from a rate of 110 Mb sec^{-1} at a distance of 10 m to a rate of 480 Mb sec^{-1} at a distance of 2 m in realistic multipath environment while consuming very little power and silicon area, are currently the focus of research and development of Wireless Personal Area Networks (WPANs). Orthogonal Frequency Division Multiplexing (OFDM) is considered as the leading choice by the 802.15.3a standardization group for use in establishing a physical-layer standard for UWB communications[3]. OFDM-based UWB not only has reliably high-data-rate transmission in time-dispersive or frequency-selective channels without having complex time-domain channel equalizers but also can provide high spectral efficiency. However, because the data sampling rate from the analog-to-digital converter to the physical layer is up to 528 M sample sec^{-1} or more, it is a challenge to realize the physical layer of the UWB system especially the components with high computational complexity in VLSI implementation.

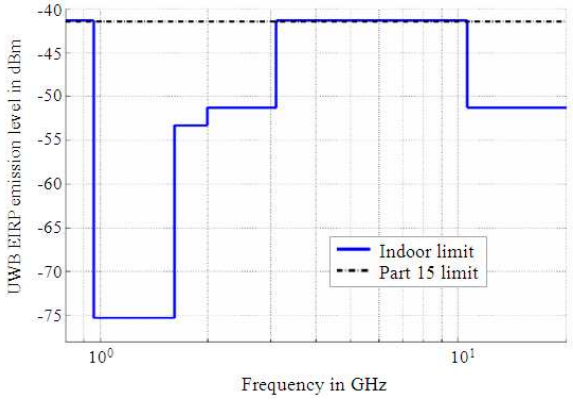

Fig. 1: FCC spectral mask for UWB indoor communication systems

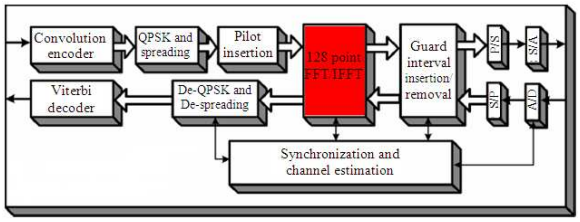

Fig. 2: Block diagram of the physical layer of OFDM-based UWB system

The FFT/IFFT processor is one of the modules having high computational complexity in the physical layer of the UWB system and the execution time of the 128 point FFT/IFFT in UWB system is only 312.5 ns. Therefore, if employing the traditional approach, a great deal of power consumption and high hardware cost of the FFT/IFFT processor will be needed to meet the strict specifications of the UWB system.

A block diagram of the proposed physical layer of OFDM-based UWB system is shown in Fig. 2. It contains a convolutional encoder, a Viterbi decoder, a pilot insertion, a QPSK-modulator/demodulator, a spreading/de-spreading, a guard interval insertion/ removal, a 128-point FFT/IFFT, a Serial-to-Parallel (S/P) converter/parallel-to serial (P/S) converter, an Analog-to-Digital Converter (ADC), a Digital-to-Analog Converter (DAC) and a synchronization and channel estimation block. In the UWB system, the data rate is from 53.3-480 Mb sec^{-1} with code rates 1/3, 11/32, 1/2, 5/8 and 3/4. The bandwidth of the transmitted signal is 528 MHz and the OFDM symbol duration is 312.5 ns, including 60.61 ns for cyclic prefix duration and 9.47 ns for guard interval duration[3]. Thus, an FFT/IFFT has to compute one OFDM symbol within 312.5 ns and the throughput rate is up to 409.6 M sample sec^{-1}.

Table 1: Comparison of the pipeline FFT architectures

Type	Complex multipliers	Complex adders	Memory	Control
R2MDC	$2\log_4 N-1$	$4\log_4 N$	3N/2-2	Simple
R2SDF	$2\log_4 N-1$	$4\log_4 N$	N-1	Simple
R4SDF	$\log_4 N-1$	$8\log_4 N$	N-1	Medium
R4MDC	$3(\log_4 N-1)$	$8\log_4 N$	5N/2-4	Simple
R4SDC	$\log_4 N-1$	$3\log_4 N$	2N-2	Complex
$R2^2SDF$	$\log_4 N-1$	$4\log_4 N$	N-1	Simple

The FFT and its inverse IFFT are the key components of OFDM systems. Recently, the demand for long length, high-speed and low-power FFT has increased in the OFDM wireless applications. There are three kinds of main design architectures for implementing a FFT processor. One is the single-memory architecture. It has one processing element and one main memory. Hence, it occupies a small area. The second is the dual-memory architecture, which has two memories. This architecture has a higher throughput than the single-memory architecture because it can store butterfly outputs and read butterfly inputs at the same time. These two memory architectures require a relatively small area. However, they have lower throughput and require higher clock frequency than the third architecture. The third is the pipeline architecture. This architecture is used for high throughput applications. It requires $\log_r$ (N) processing elements; therefore its calculations are $\log_r$ (N) times faster than the processor based on the single-memory architecture. However, this scheme has the disadvantage of consuming a large power/area[4,5].

Hence, the efficient FFT algorithm should be chosen to minimize hardware complexity to employ the pipeline FFT architecture in wideband OFDM wireless applications. Among various FFT algorithms, the Cooley-Turkey algorithm is very popular because it can reduce the computational complexity from $O(N^2)$ to $O(N \log_2 N)$. The regularity of the algorithm makes it suitable for VLSI implementation. To further reduce the computational complexity, radix-4, split-radix, radix-2^2, radix-2/4/8 and higher radix versions have been proposed. In general, all of these algorithms decompose a length-N (2^n) FFT into an odd half and an even half recursively and effectively reduce the number of complex multiplications by utilizing symmetric properties of the FFT kernel.

Several architectures for pipeline FFT processors have been proposed over the last 3 decades, along with the increasing interest and the rapid progress of the technology. The pipeline FFT architectures listed in Table 1 have the distinctive merits and common requirements of the different approaches[6]. The Single-path Delay Feedback (SDF) approaches are always more efficient than the corresponding Multi-path Delay Commutator (MDC) approaches in terms of memory requirements. Among them the $R2^2SDF$ has reached the minimum requirement for both multiplier and storage and only second to the Radix-4 Single-path Delay Commutator (R4SDC) for adder. The R4SDC has also reached the minimum requirement for both multiplier and adder. However for the long-point FFT processor used in wideband OFDM systems, the $R2^2SDF$ is more profitable than R4SDC because of its relatively less memory requirement.

In 1998, He and Torkeson[6] suggested radix-2^2 and radix-2^3 FFT algorithms. These algorithms are characterized by the trait that reduces the number of the non-trivial multiplications in the radix-2 algorithm architectures. It has the same number of non-trivial multiplications at the same positions in the Signal Flow Graph (SFG) as of the radix-4 algorithm, but has the same butterfly structure as that of the radix-2 algorithm. Even if the complexity of multiplications was reduced in the radix-2^2 algorithm, that is still an essential point for the pipeline FFT implementation, because it consists of four real multiplications and two real additions.

In this study, we propose the modified radix-2^2 algorithm and its pipeline FFT architecture as the scheme to reduce the area and power consumption of the multiplication. The proposed pipeline approach has the characteristic that can replace the whole complex multipliers with the novel CSD constant multipliers. Radix 2^2 is a hardware oriented algorithm that has the same number of non-trivial multiplications at the same positions in the SFG as of radix-4 algorithms, but has the same butterfly structure as that of radix-2 algorithms[7]. The notation of radix-2^2 is used to clearly reflect the structural relation with radix-2 algorithm and the identical computational requirement with radix-4 algorithm. The DFT of size N is defined by:

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

where, W_N denotes, $e^{-j2\pi/N}$, the primitive N^{th} root of unity, with its exponent evaluated modulo N. The 'n' is the time index and the 'k' is the frequency index. We can derive the modified radix 2^2 algorithm by integrating the twiddle factor decomposition through divide and conquer approach. To make the derivation of the modified Radix-2^2 algorithm, we will consider the first 4 stages in cascade decomposition through a divide and conquer approach instead of the first 2 as applied in the original Radix-2^2[6]. Then we apply a 5-dimentional linear index map as follows:

$$\begin{aligned} n &= \left\langle \frac{N}{2}n_1 + \frac{N}{4}n_2 + \frac{N}{8}n_3 + \frac{N}{16}n_4 + n_5 \right\rangle_N \\ k &= \langle k_1 + 2k_2 + 4k_3 + 8k_4 + 16k_5 \rangle_N \end{aligned} \tag{2}$$

The DFT equation has the form of:

$$X\left(k_1+2k_2+4k_3+8k_4+16k_5\right)=\sum_{n_5=0}^{\frac{N}{16}-1}\sum_{n_4=0}^{1}\sum_{n_3=0}^{1}\sum_{n_2=0}^{1}\sum_{n_1=0}^{1}\times\left(\begin{matrix}\frac{N}{2}n_1+\frac{N}{4}n_2\\+\frac{N}{8}n_3+\frac{N}{16}n_4+n_5\end{matrix}\right)w_N^{nk} \tag{3}$$

Decomposing the composite twiddle factors, it can be expressed in Eq. 4:

$$W_N^{\left(\frac{N}{2}n_1+\frac{N}{4}n_2+\frac{N}{8}n_3+\frac{N}{16}n_4+n_5\right)\left(k_1+2k_2+4k_3+8k_4+16k_5\right)}=\left\{(-1)^{n_1k_1}(-j)^{n_1k_1}(-1)^{n_2k_2}W_{16}^{(2n_3+n_4)(k_1+2k_2)}\right\}.\left\{(-1)^{n_3k_3}(-j)^{n_4k_3}(-1)^{n_4k_4}W_{16}^{n_5(k_1+2k_2+4k_3+8k_4)}\right\}W_{\frac{N}{16}}^{n_5k_5} \tag{4}$$

Substituting equation (4) in equation (3) and expand the summation with index n_1, n_2, n_3 and n_4, we have a set of 16 DFTs of length N/16:

$$X\left(k_1+2k_2+4k_3+8k_4+16k_5\right)=\sum_{n_5=0}^{\frac{N}{16}-1}\left[G_{\frac{N}{16}}^{k_1k_2k_3k_4}(n_5)W_N^{n_5(k_1+2k_2+4k_3+8k_4)}\right]W_{\frac{N}{16}}^{n_5k_5} \tag{5}$$

Where:

$$G_{\frac{N}{16}}^{k_1k_2k_3k_4}(n_5)=T_{\frac{N}{8}}^{k_1k_2k_3}(n_5)+(-1)^{k_4}(-j)^{k_3}T_{\frac{N}{8}}^{k_1k_2k_3}\left(n_5+\frac{N}{16}n_4\right) \tag{6}$$

and

$$T_{\frac{N}{8}}^{k_1k_2k_3}\left(n_5+\frac{N}{16}n_4\right)=W_{16}^{n_4(k_1+2k_2)}H_{\frac{N}{4}}^{k_1k_2}\left(n_5+\frac{N}{16}n_4\right)+(-1)^{k_3}W_{16}^{(2n_3+n_4)(k_1+2k_2)}H_{\frac{N}{4}}^{k_1k_2}\left(n_5+\frac{N}{16}n_4+\frac{N}{8}n_3\right) \tag{7}$$

and

$$H_{\frac{N}{4}}^{k_1k_2}\left(n_5+\frac{N}{16}n_4+\frac{N}{8}n_3\right)=B_{\frac{N}{2}}^{k_1}\left(n_5+\frac{N}{16}n_4+\frac{N}{8}n_3\right)+(-1)^{k_2}(-j)^{k_1}B_{\frac{N}{2}}^{k_1}\left(n_5+\frac{N}{16}n_4+\frac{N}{8}n_3+\frac{N}{4}n_2\right) \tag{8}$$

and

$$B_{\frac{N}{2}}^{k_1}\left(n_5+\frac{N}{16}n_4+\frac{N}{8}n_3+\frac{N}{4}n_2\right)=x\left(n_5+\frac{N}{16}n_4+\frac{N}{8}n_3+\frac{N}{4}n_2\right)+(-1)^{k_2}x\left(n_5+\frac{N}{16}n_4+\frac{N}{8}n_3+\frac{N}{4}n_2+\frac{N}{2}\right) \tag{9}$$

The flow graph of the modified Radix 2^2 algorithm for N = 16 example is shown in Fig. 3. It has the same multiplicative complexity as radix-4 algorithms, but still retains the radix-2 butterfly structures. The multiplicative operations are in such an arrangement that only every other stage has non-trivial multiplications. This is a great structural advantage over other algorithms when pipeline/cascade FFT architecture is under consideration.

Figure 4 shows the conventional Implementation of the FFT Processor which is the single memory architecture. It has one processing element that performs butterfly operation and one memory element. Butterfly outputs are stored in the same memory location used by butterfly inputs[8]. This architecture requires small area. However, it have low throughput and requires high clock frequency. For high throughput applications Fig. 5 shows the pipeline architecture[6], which is characterized by non-stopping processing on a clock frequency of the input data sampling. With the pipeline architecture[9], a high-speed FFT processor can be implemented. However, it requires more hardware resources (especially more complex multipliers), which is not suitable for the portable application of OFDM systems.

So, in the design of FFT processors for OFDM systems, we should not only enhance the speed by introducing more parallelization and pipelines, but also reduce the hardware resource consumption as possible as we can. Based on the rule, a multiplier less architecture to replace the conventional complex multiplier using the CSD representation will be used.

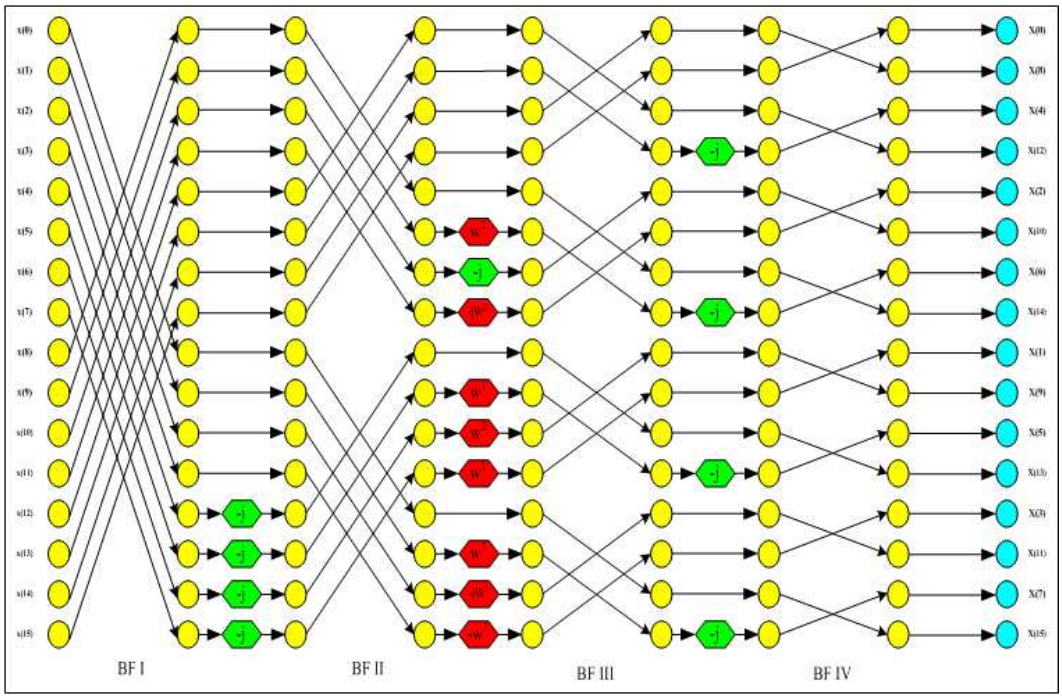

Fig. 3: Modified Radix 2^2 DIF FFT flow graph for N = 16

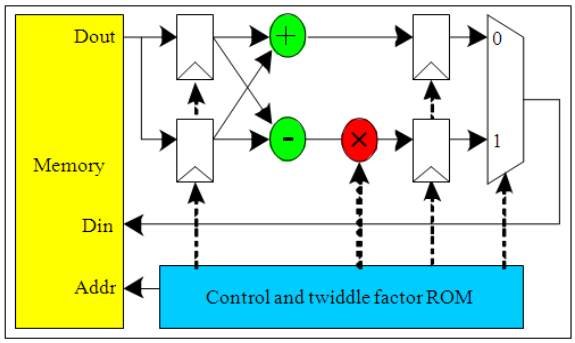

Fig. 4: Single memory architecture

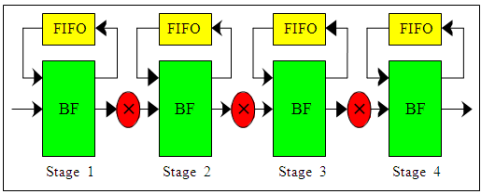

Fig. 5: Pipeline architecture

The complex multiplier in each stage is the key component and the most power consuming modules in FFTs. The direct implementation of a complex multiplier requires one subtractor, one adder and four real multipliers as shown in Fig. 6. However, the coefficient for the stages can be previously calculated.

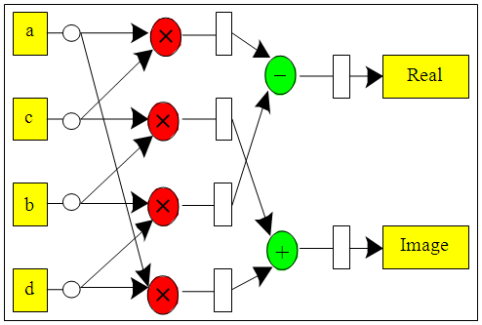

Fig. 6: Complex multiplication

In our application twiddle factor coefficients and variable inputs are generated in MATLAB tool.

A close look at Fig. 3 reveals that, the nontrivial coefficient is available is stage 2 only the rest of stages contain only trivial coefficients. For the trivial coefficient like (0000, 8000) which are the quantized representation for (0,-1) in 16 bit two's complement format, the complex multiplication with the input is not necessary. Only an additional unit, which swaps the real and imaginary parts of input data and inverts the imaginary parts of input data, is needed. So the input can be passed without the multiplication, this reduces the extra hardware. The nontrivial coefficients are w^1, w^2 and w^3 or (7642, CF04),

(5A82, A57E) and (30FC, 89BE) which are the quantized representation for (0.9239, -0.3827), (0.7071, -0.7071) and (0.3827, -0.9239) in 16-bit two's complement format. These coefficients are composed of only three constants (0.9239, 0.7071 and 0.3827). This means implementing a constant multiplier of these three constant will be enough to eliminate the need for the whole complex multipliers and the ROM to store the twiddle factor coefficients. The multiplication with a constant can be carried out by adding the partial product terms corresponding to the nonzero bit positions in the constant multiplier. To reduce the area and power consumption, the constant coefficient can be encoded such that it contains the fewest number on nonzero bits, which can be accomplished using CSD representation. A number in CSD format has fewer nonzero digits than its binary equivalent and, hence, will require fewer additions during multiplication. The CSD number system is based on signed digit number systems[10], which allow individual digits to have a sign as well as a value:

$$\text{digit} \in \left\{-\frac{r}{2},...,-1,0,1,...,\frac{r}{2}\right\} \quad (10)$$

Generally, the digits of these number systems are chosen as shown in Eq. 10 and can have any base. As a replacement of the binary system for high-speed multiplication, the ternary number system where r = 2 is used. This allows the digits to have values of 0, 1, or -1. Typically the -1 digit is written as $\bar{1}$. In this number system, the sign and value of the overall number are determined by the weighted sum of the signed digits as shown in Eq. 11:

$$\text{value} = d_0 d_1 d_2,...,d_{N-1} = \sum_{i=0}^{N-1} d_i 2^{-i} \quad (11)$$

In multiplication, the shift and add operation of the binary number system is extended to include subtraction for the case when a digit has a value of -1. Subtraction and addition are comparable in terms of speed of execution, so allowing -1 digits will not hinder the multiplication time, yet the extra freedom offers a great potential to increase the number of zero digits used to represent a given value. The signed digit number system is a redundant number system because a given value may be represented by more than one sequence of digits. For example, $0.01 = 2^{-2} = 0.25$ and $0.1\bar{1} = 2^{-1}-2^{-2} = 0.25$ are two different representations with the same value. However, for any given value with two or more redundant representations, there will be only one representation where a signed digit number of length N follows the constraint of Eq. 12:

$$d_n d_{n+1} = 0 \qquad \text{for} \quad 0 \le n \le N-2 \quad (12)$$

Such a number is said to be the canonical form of the signed digit number or simply the CSD form. Following from Eq. 12 is the property that the number has no adjacent nonzero digits. Another property of the CSD form is that it has the fewest number of nonzero digits among the redundant forms. An N-bit number in CSD format is able to uniquely express every value of an N-bit two's complement binary number, but it will never have more than (N + 1)/2 nonzero bits. This makes it a very desirable form for high throughput FFT.

Genetic algorithm: The Genetic Algorithm (GA) is a relatively new optimization technique that was originally developed by[11]. It was further modified by Goldberg and others. Their primary goal was to abstract and rigorously explain the adaptive processes of natural systems and to design artificial system software that retains the important mechanisms of natural systems[12]. The characteristics of GA include multi-objective[13], exponential convergence rate, coded variables and natural selection that provide advantages in solving discrete space problems. Unlike most of the well-known optimization techniques such as simulated annealing[14], branch and bound optimization technique[15], GA searches a population of points rather than a single point at each iteration. This feature prevents the search from being captured by the error surface minima and provides different search directions to seek a global solution. Another characteristic of GA is that it utilizes stochastic rather than deterministic operators which allow the GA to perform on a discontinuous space without disruption.

GA is an artificial genetic system which is based on the processes of natural selection and natural genetic[12] and has been effectively implemented in an optimization scheme. The genetic based optimization scheme is modeled by three major operators: Reproduction, Crossover and Mutation. Unknown variables are stored in a place, named Population and will be manipulated by the operators consecutively as shown in the basic GA Cycle diagram, Fig. 7.

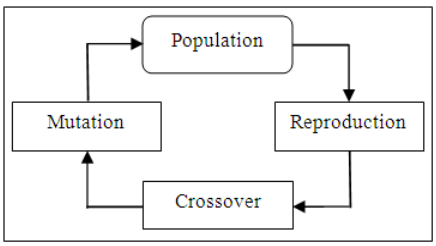

Fig. 7: Genetic algorithm cycle

The population is a collection of chromosomes. Each chromosome is a bit Stream of cascaded and encoded variables. For a given parent selection scheme within the Reproduction operator, more suitable chromosomes are picked from the population for further genetic enhancements. A chromosome is more suitable if it results in a smaller error cost-function. Selected chromosomes are sent to the crossover operator to create new chromosomes by exchanging their partial bit Stream. These new chromosomes are called off-springs. The off-springs are then passed on to the Mutation operator for further manipulation and then returned to the population. One completed cycle is called a generation.

While genetic algorithms are quite suitable for handling discrete search space, they cannot be directly applied to the case of CSD number space. The direct application of genetic operators of crossover and mutation to CSD numbers may cause the resulting offspring coefficients cease to conform to CSD format and they have to be either discarded or restored. In this study the offspring is simply discarded. Figure 8 shows the GA-based design flow to implement the twiddle factor coefficient in CSD representation. First the population initialization generates N (population size) chromosomes randomly. Each gene has M-bit word length and maximum of L nonzero digits. M and L can be set to any desired value. Secondly, the Roulette Wheel Selection[16] has been used in this GA.

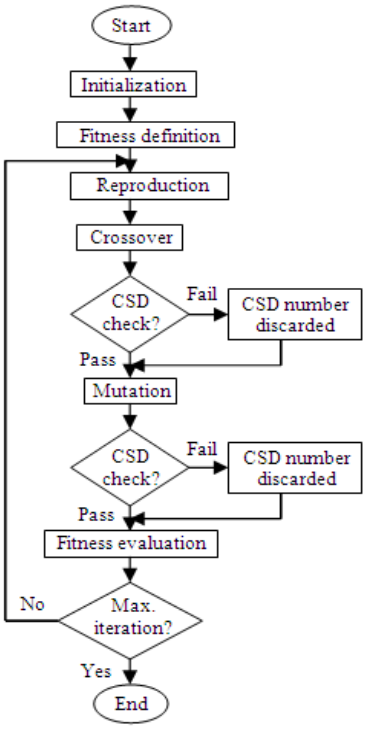

Fig. 8: GA-based design flow

The Roulette Wheel selection chooses chromosomes in a statistical fashion, based solely upon their relative fitness values. Fitter chromosomes have a higher chance of surviving in the subsequent generation. Then the one-point crossover is used in this GA and the crossover point is randomly selected. After crossover operation, the coefficient where the crossover point lies in will be checked upon CSD format. If the coefficient is found violated, it will be discarded. Finally Mutation operator is the simple single bit flip. After mutation, each coefficient in the offspring is checked upon CSD format. Any violated coefficient discarded.

RESULTS

Simulink implementation of the modified Radix-2^2 Decimation In Frequency (DIF) Fast Fourier Transform (FFT) module for 16-points is shown in Fig. 9 and the entire structure is shown in Fig. 10. The module consists of four stages and a bit reversal block that is used to reverse the sequence in order to get the output in normal order.

Figure 11-13 shows the comparison in terms of addition elements between the implementation of the constant multiplication with a twiddle factor's coefficients in 2's complements and the genetic algorithm based CSD representation form for different twiddle factors. They show the optimization of three non trivial twiddle factors available in 16-pionts FFT; they are 0.9239, 0.7071 and 0.3827 respectively.

Figure 14 shows the ModelSim simulation of the conventional radix 2^2 16 points FFT where the twiddle factors are stored in ROM and the butterfly uses the normal complex multiplier as shown in the RTL schematic in Fig. 15. Each stage require 4 clock cycles to finish processing the data and store the results in RAM to be processed by the next stage, this means that the over all time is (4 cycles x 4 stages) equals 16 clock cycles.

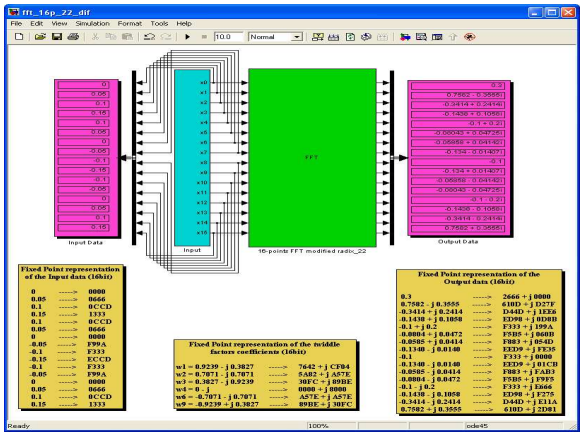

Fig. 9: Modified radix 2^2 DIF 16-points FFT

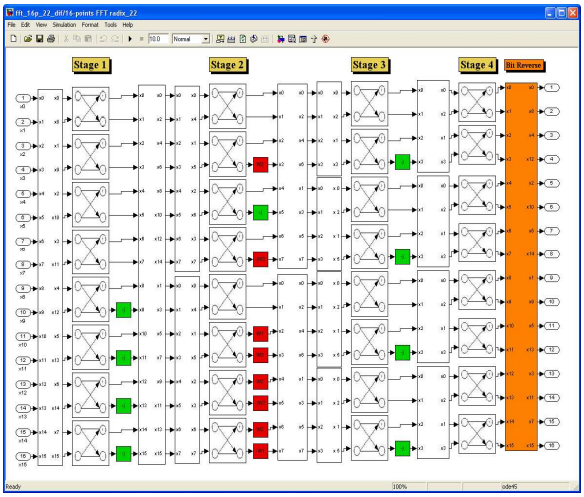

Fig. 10: Entire structure of the design

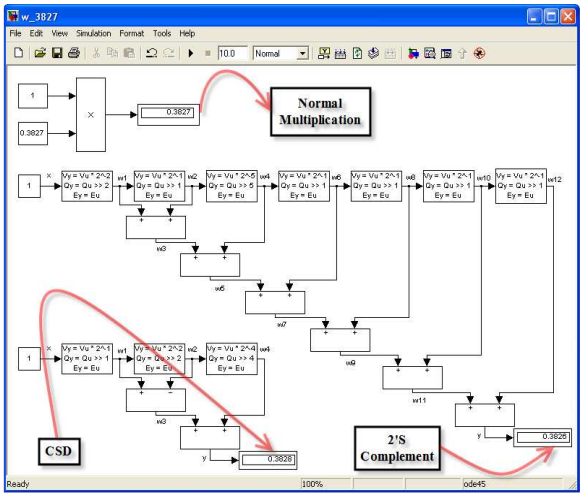

Fig. 13: CSD implementation of the 0.3827

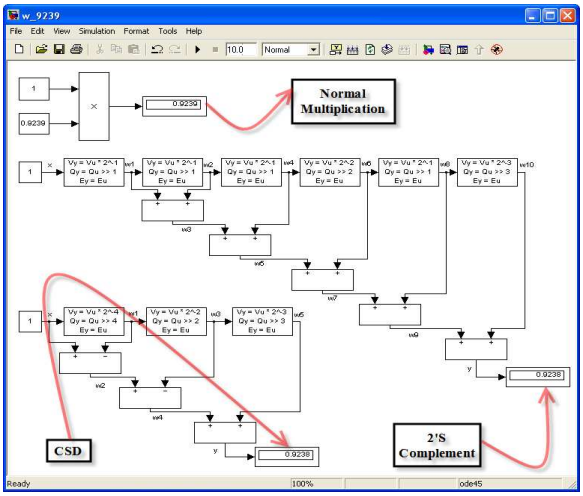

Fig. 11: CSD implementation of the 0.9239

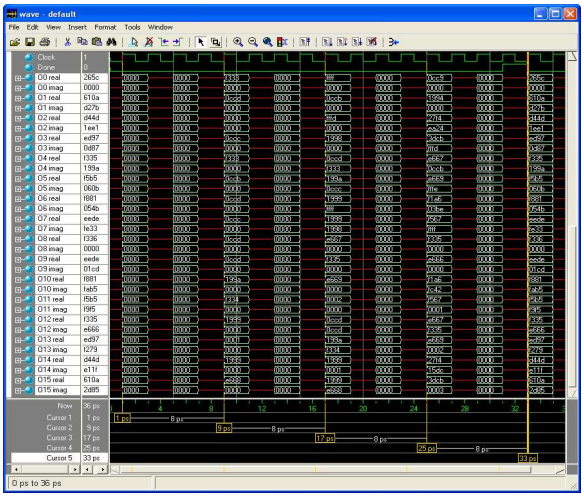

Fig. 14: ModelSim simulation of the conventional radix 2^2 16-points FFT

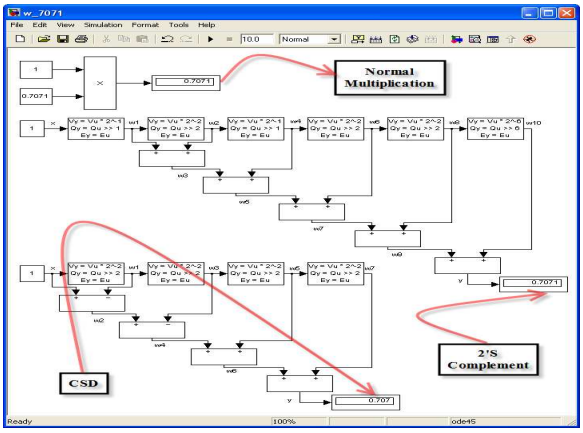

Fig. 12: CSD implementation of the 0.7071

DISCUSSION

Based on the resulting Fig. 11-13 it is clear that our CSD implementation for the twiddle factor coefficients outperforms the 2's complement implementation. Its addition elements reduced by 20%, 40% and 65% for the twiddle factor's coefficients (0.9239), (0.7071) and (0.3827), respectively.

Our novel approach is implemented using the modified Radix 2^2 FFT algorithm without any complex multipliers by representation the multiplication with the twiddle factor coefficients in terms of its partial product terms corresponding to the nonzero bit positions. To further reduce the area and power consumption, we have encoded the coefficient in CSD representation to

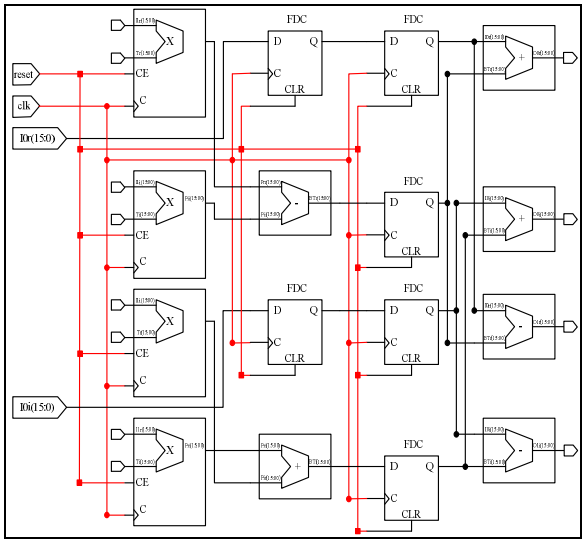

Fig. 15: RTL schematic for the conventional butterfly

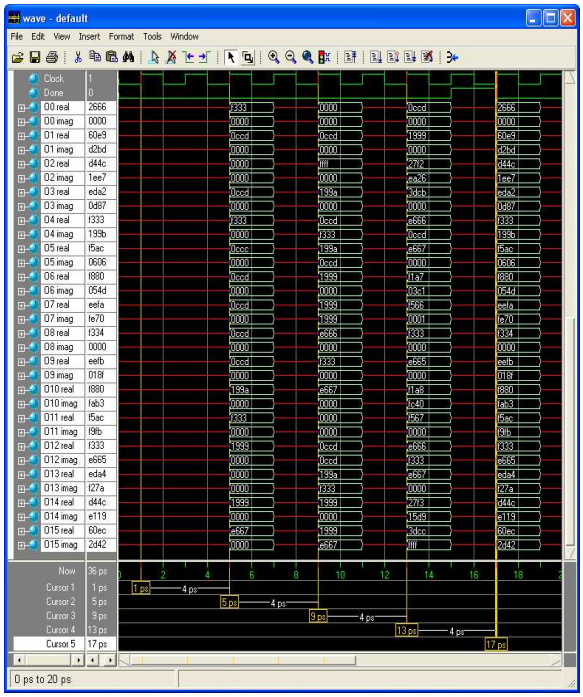

Fig. 16: ModelSim simulation of the CSD Multiplier less modified Radix 2^2 16-points FFT

reduce the number on nonzero bits. Figure 16 and 17 shows the ModelSim simulation of the CSD multiplier less modified Radix 2^2 16-points FFT and the RTL schematic for the CSD multiplier less butterfly respectively. Each stage require only 2 clock cycles to finish processing the data and store the results in RAM to be processed by the next stage, this means that the over all time is (2 cycles x 4 stages) equals 8 clock cycles.

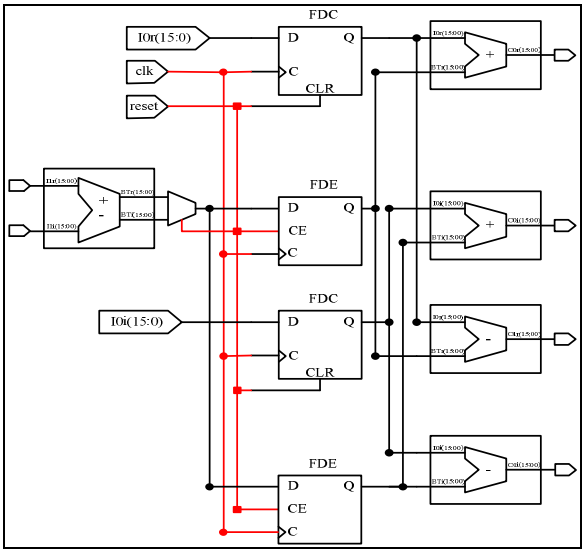

Fig. 17: RTL schematic for the CSD multiplierless butterfly

CONCLUSION

In this study, a hardware-oriented modified radix-2^2 algorithm is used which has the radix-4 multiplicative complexity but retains radix-2 butterfly structure in the SFG. Based on this algorithm, a novel genetic algorithm based CSD multiplier less architecture is put forward to replaces the traditional complex multiplier that uses four real multiplications and two additions by a multiplication free operation at the cost of addition and shift. The hardware requirement of the proposed architecture as compared with the traditional approach is shown. The architectures have been coded in Verilog HDL. The architectures were synthesized using the Xilinx ISE Navigator. The target board was the Xilinx Virtex II FPGA. The main idea of this structure is to make the operation without multiplications (mainly contains addition operations), because the area that addition units occupy is very small. Performance evaluation proved that the FFT processor with this architecture is suitable for wireless PAN UWB (802.15.3a) applications.

REFERENCES

1. Ross, G.F., 1973. Transmission and reception system for generating and receiving base-band duration pulse signals without distortion for short base-band pulse communication system. US Patent 3728632.
http://www.freepatentsonline.com/3728632.html

2. FCC, 2002. New public safety applications and broadband internet access among uses envisioned by FCC authorization of ultra-wideband technology. First Report and Order (FCC 02-48), Action by the Commission. http://www.fcc.gov/Bureaus/Engineering_Technology/News_Releases/2002/nret0203.html
3. Batra, A. *et al*., 2004. Multi-Band OFDM Physical Layer Proposal for IEEE 802.15 Task Group 3a. IEEE P802.15-03/268r3. http://grouper.ieee.org/groups/802/15/pub/2003/Jul03/03268r2P802-15_TG3a-Multi-band-CFP-Document.pdf
4. Bass, B.M., 1999. A low power, high performance, 1024-point FFT processor. IEEE. J. Solid-State Circ., 34: 380-387. http://direct.bl.uk/bld/PlaceOrder.do?UIN=057807702&ETOC=RN&from=searchengine
5. Lo, H.F., M.D. Shieh and C.M. Wu, 2001. Design of an efficient FFT processor for DAB systems. Proceeding of the IEEE International Symposium on Circuits and Systems, May 6-9, IEEE Xplore Press, Washington DC., USA., pp: 654-657. DOI: 10.1109/ISCAS.2001.922322
6. He, S. and M. Torkelson, 1998. Designing pipeline FFT processor for OFDM (de)modulation. Proceeding of the IEEE URSI International Symposium on Signals, System and Electronics, Sept. 29-Oct. 2, IEEE Xplore Press, Pisa, Italy, pp: 257-262. DOI: 10.1109/ISSSE.1998.738077
7. He, S. and M. Torkelson, 1996. A new approach to pipeline FFT processor. Proceeding of the 10th International Symposium on Parallel Processing, Apr. 15-19, IEEE Xplore Press, Honolulu, HI., USA., pp: 766-770. DOI: 10.1109/IPPS.1996.508145
8. Son, B.S., B.G. Jo, M.H. Sunwoo and Y.S. Kim, 2002. A high-speed FFT processor for OFDM systems. Proceedings of the IEEE International Symposium on Circuits and Systems, (ISCAS'02), IEEE Xplore Press, USA., pp: 281-284. DOI: 10.1109/ISCAS.2002.1010215
9. Wold, E.H. and A.M. Despain, 1984. Pipeline and parallel pipeline FFT processor for VLSI implementation. IEEE Trans. Comput., 33: 414-426. DOI: 10.1109/TC.1984.1676458
10. Hwang, K., 1979. Computer Arithmetic Principle, Architecture and Design. John Wiley and Sons, New York, USA., ISBN: 0471034967, pp: 423.
11. Holland, J., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press, ISBN: 0472084607, pp: 183.
12. Goldberg, D.E., 1988. Genetic Algorithm in Search, Optimization and Machine Learning. Addison-Wesley, ISBN: 0201157675, pp: 112.
13. Fonseca, C.M. and P.J. Fleming, 1993. Genetic algorithm for multi-objective optimization: Formulation, discussion and generalization. in genetic algorithms. Proceedings of the 5th International Conference on Genetic Algorithm, (GA'93), Morgan Kaufmann, ACM Press, San Mateo, CA., pp: 416-423. http://portal.acm.org/citation.cfm?id=645513.657757
14. Kirkpatrick, S., C.D. Gelatt Jr. and M.P. Vecchi, 1983. Optimization by simulated annealing. Science, 220: 671-679. DOI: 10.1126/science.220.4598.671
15. Ashrafzadeh, F., B. Nowrouzian and A.T.G. Fuller, 1998. A novel modified branch-and-bound technique for discrete optimization over canonical signed-digit number space. Proceedings of IEEE International Symposium on Circuits and Systems, May 31-June 3, IEEE Xplore Press, USA., pp: 391-394. DOI: 10.1109/ISCAS.1998.694507
16. Tang, K.S., K.F. Man, S. Kwong and Q. He, 1997. Genetic algorithms and their applications. IEEE Signal Process. Mag., 13: 22-37. http://www.informatics.indiana.edu/fil/CAS/PPT/Davis/