



A Modified Low Power FFT/IFFT Processor for OFDM Applications

S.Lavanya*, Ms.M.Jasmin
PG Scholar*, Assistant Professor
Electronics & Communication Engineering
Bharath University
alps_lavanya@yahoo.co.in*, rifrizz@gmail.com

Abstract: Orthogonal Frequency Division Multiplexing (OFDM) often require an inverse fast Fourier transform (IFFT) to produce multiple sub-carriers. The read-only memories (ROM's) used to store the twiddle factors, are eliminated in the proposed architecture which applies a reconfigurable complex multiplier and bit-parallel multipliers to achieve a ROM-less FFT/IFFT processor. It adopts a single-path delay feedback style as the proposed hardware architecture, thus consuming lower power than the existing methodology.

Keyword: single-path delay feedback, bit-parallel multipliers, low power.

I. INTRODUCTION

The Fast Fourier transform (FFT) is the suitable choice for communication purpose since the computational complexity is reduced from $O(N^2)$ to $O(N \log N)$. Therefore a high-speed performance and hardware reduction can be attained.

The main trends of FFT hardware development are towards high throughput and low power consumption. The pipelined structure is the most common choice for high-throughput FFT processor. Many methods have been proposed to implement the pipelined FFT hardware architectures[6]. They can be categorized into three kinds, namely, the multiple-path delay commutator (MDC), single-path delay feedback (SDF), and single-path delay commutator (SDC) architectures. In comparison with the three pipeline architectures, the SDF architecture is the most suitable for FFT implementation[7]. Its advantages are (i).the SDF architecture is very simple to implement the different length FFT, (ii).the required registers in SDF architecture is less than MDC and SDC architectures,(iii).The control unit of SDF architecture is easier than the others.

A. Discrete Fourier Transform

Discrete Fourier transform (DFT) is a very important technique in modern digital signal processing (DSP) and telecommunications, especially for applications in orthogonal frequency demodulation multiplexing (OFDM) systems, such as IEEE 802.11a/g, Worldwide Interoperability for Microwave Access (WiMAX)[3], Long Term Evolution(LTE), and Digital Video Broadcasting—Terrestrial(DVB-T). However, DFT is computational intensive and has a time complexity of $O(N^2)$. The fast Fourier Transform(FFT) was proposed by Cooley and Tukey to efficiently reduce the time complexity to $O(N \log 2N)$, where N denotes the FFT size[4][5].

For hardware implementation, various FFT processors have been proposed. These implementations can be mainly classified into memory-based and pipeline architecture styles. Memory-based architecture is widely adopted to design an FFT processor, also known as the single processing element (PE) approach. This design style is

usually composed of a main PE and several memory units, thus the hardware cost and the power consumption are both lower than the other architecture style. However, this kind of architecture style has long latency, low throughput, and cannot be parallelized. On the other hand, the pipeline architecture style can get rid off the disadvantages of the foregoing style, at the cost of an acceptable hardware overhead. The single-path delay feedback (SDF) pipeline FFT is good in its requiring less memory space (about $N-1$ delay elements) and its multiplication computation utilization being less than 50%, as well as its control unit being easy to design[1]. Such implementations are advantageous to low-power design, especially for applications in portable DSP devices. Based on these reasons, the SDF pipeline FFT is adopted in this work. However, the FFT computation often needs to multiply input signals with different twiddle factors for an outcome, which results in higher hardware cost because a large size of ROM is needed to store the wanted twiddle factors. Therefore, to throw off these ROM's for area-efficient consideration.

B. Complex Multipliers:

The complex multipliers used in the processor are realized with shift-and-add operations. Hence the processor uses only a two-input digital multiplier and does not need any ROM for internal storage of coefficients. However, low speed and higher hardware cost caused by the proposed complex multiplier are the pay-off. In order to further improve the power consumption and chip area of previous works, this paper proposes an efficient pipeline architecture with low power consumption for the FFT/IFFT processor[2].

II. ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING

OFDM is the abbreviation for Orthogonal Frequency Division Multiplexing, and describes a digital modulation scheme that distributes a single data stream over a large number of carriers for parallel transmission. These carriers are called the subcarriers of the signal. In the frequency domain, they are equally spaced around a central RF carrier,

so the frequency $f_{n,rf}$ of the n th subcarrier out of N can be expressed as

$$f_{n,rf} = f_c + n \cdot f_d, \quad \text{with} \quad (1)$$

$$n \in \left[-\frac{N-1}{2}, \frac{N-1}{2}\right] \text{ if } N \text{ is odd and} \quad (2)$$

$$n \in \left[-\frac{N}{2}, \frac{N}{2}-1\right] \text{ if } N \text{ is even.} \quad (3)$$

f_d is the frequency spacing between the subcarriers and f_c is the center frequency of the OFDM signal.

In the baseband, we obtain

$$f_n = n \cdot f_d, \quad \text{with} \quad (4)$$

$$n \in \left[-\frac{N-1}{2}, \frac{N-1}{2}\right] \text{ for odd and} \quad (5)$$

$$n \in \left[-\frac{N}{2}, \frac{N}{2}-1\right] \text{ for even} \quad (6)$$

where f_n is the baseband frequency of the n th subcarrier.

The subcarriers $\exp(2\pi j n f_d t)$ build an orthogonal set of functions, hence the name of the modulation. Each of these subcarriers is modulated separately. The modulation symbols result from encoding the binary data using traditional PSK or QAM mappings. Let us assume we want to transmit 128 bits simultaneously on $N = 64$ QPSK modulated subcarriers ($n = -32, -31, \dots, +31$). Then for example the subcarrier with index -32 may carry the first pair of bits, the subcarrier with index -31 would carry the second pair, and so on. The last two bits would then be assigned to the subcarrier with index +31.

So, within one OFDM symbol, each subcarrier has its own phase p_n and amplitude a_n . The whole baseband signal looks like this:

$$\sum_{n=n_{\min}}^{n_{\max}} a_n \cdot e^{j p_n} \cdot f_n \quad (7)$$

Where $n_{\min}$ and $n_{\max}$ set the range for n .

We can interpret this as the Discrete Fourier Transform (DFT) of the OFDM symbol to generate. Consequently, instead of really modulating all 64 subcarriers, the discrete time domain signal is calculated by applying to s an inverse DFT of length N . The result is a series of samples that represent one period of a signal that has the same spectrum as the desired OFDM symbol. The sampling frequency is $f_s = N \cdot f_d$. One period of the calculated series yields all amplitude and phase information contained in s . The minimum OFDM Symbol duration is thus

$$T_u = \frac{N}{f_d} \quad (8)$$

If N is chosen to be power of 2, an inverse fast Fourier transform (FFT) can be used instead of the more time consuming DFT. Notice that the DFT is defined for signals that are periodic in the time domain as well as in the frequency domain. The alias spectra are eliminated through (digital) baseband filtering. Therefore, it is necessary to leave some empty space at the high and low end of the DFT spectrum s for the filter to cut in. Thus, only a subset of the N maximum possible subcarriers will actually be modulated, the others have zero amplitude.

A. Twiddle Factor:

A twiddle factor, in fast Fourier transform (FFT) algorithms, is any of the trigonometric constant coefficients that are multiplied by the data in the course of the algorithm "twiddle factors" originally referred to the root-of-unity complex multiplicative constants in the butterfly operations of the Cooley-Tukey FFT algorithm, used to recursively combine smaller discrete Fourier transforms. This remains the term's most common meaning, but it may

also be used for any data-independent multiplicative constant in an FFT[5].

The Prime-factor FFT algorithm is one unusual case in which an FFT can be performed without twiddle factors, only for restricted factorizations of the transform size.

B. The Cost Of Mapping:

Different types balance mapping with sub problem cost

(i) in radix-2

- Sub problems are trivial (only sum and differences)
- Mapping requires twiddle factors (large number of multiplies)

(ii) in prime-factor algorithm

- Sub problems are DFTs with co-prime lengths (costly)
- Mapping trivial (no arithmetic operations)

III. CANONIC SIGNED DIGIT

The canonic signed digit (CSD) representation is used to design the function of complex multiplier, which is the main function block in the FFT processor. The processor of a 16-bit 16-point pipeline FFT is realized on the Xilinx Virtex-4 FPGAs. The achieved maximum clock frequency is 196.8 MHz, utilizing 310 out of 49152 slices and 241 out of 98304 look-up tables. Another 16-bit 64-point pipeline FFT processor is also realized. The achieved maximum clock frequency is 111.2 MHz, utilizing 1303 out of 49152 slices and 2065 out of 98304 look-up tables. Comparing with the conventional complex multiplier, the derived results show the proposed design has improved efficiency on Virtex-4.

The main trends of FFT hardware development are toward high throughput and low power consumption. The pipelined structure is the most common choice for high-throughput FFT processor. The word lengths of various signals are minimized according to their respective signal-to-noise ratio (SNR) requirements. To decide the optimal word length, input waveforms with Gaussian noise are fed to the FFT with fixed-point arithmetic implementation. The frequency domain FFT output signals are obtained and the output signal-to-noise ratio (SNR) is computed. The CORDIC algorithm has been used for the twiddle factor multiplication in FFT processors due to its efficiency in vector rotation. In this sub-Section, we evaluate and compare the performance and complexity of a CORDIC and a complex multiplier in phase rotation. The conventional CORDIC algorithm refers to the radix-2CORDIC, and radix-2=4 CORDIC refers to the work in that enhances operation speed and reduces 25% of the micro-rotation stages. The complex multiplier used in the proposed chip consists of three multiplications and five additions. To make a fair comparison, we set the precision to 16 bits in all algorithms. To avoid rounding error propagation, 19 bits are allocated in the data path of the CORDIC-based architectures.

A. Modified Complex Multiplier:

The conventional complex multiplication this architecture needs four multipliers and two adders.

$$(x + xi)(y + yi) = (x \cdot y - xi \cdot yi) + (xi \cdot y + x \cdot yi)$$

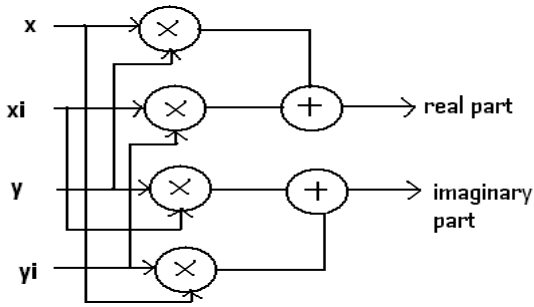


Figure 3.1. Block Diagram of Conventional Complex Multiplier Structure

Conventional complex multiplier structure for the pipeline FFT processors, their performances depend on the arithmetic operation of the complex multipliers. To increase the overall operational speed, reduce area and power consumption, an efficient implementation of complex multipliers is very important. The canonic signed digit (CSD) representation is chosen for realizing complex multipliers because it yields significant hardware reduction comparing with conventional complex multiplier. The CSD unit consists of many shifters and adders. It is used to realize the multiplication when the one factor is constant. The CSD unit is used to construct a complex multiplier. Since CSD representation of a number contains the minimum possible number of non-zero bits for the range $(-4/3, 4/3)$, the CSD blocks for the complex computing of multiplication in Fig. 2 need fewer adders and hence the power consumption and area can be reduced by about 33%.

B. Hardware Realization:

The design of a 16-point pipeline radix-2 SDF FFT is implemented on the Xilinx Virtex-4 xc4vlx100. The achieved maximum clock frequency is 196.8MHz. It utilizes 310 out of 49152 slices and 241 out of 98304 look-up tables. The 64-point pipeline radix-22 SDF FFT is also realized on the Xilinx Virtex-4 xc4vlx100. It achieves the maximum clock frequency of 111.182MHz and utilizes 1303 out of 49152 slices, 2065 out of 98304 look-up tables. Due to the efficiency of Booth multiplier, it is wide applied to realize the conventional complex multiplier. Comparing the realized complex multipliers on the Xilinx FPGA of with our complex multiplier, improved performances of the presented complex multiplier can be obtained. The efficiency of the circuit is quantified by throughput per unit of area (MHz/CLB). The comparison is shown in table.

Table 3.1. Twiddle Factor of Real and Imaginary Part

Twiddle factor	Co-efficient of 16-point	
	Real part	Imaginary part
W_{16}^1	0.923828	0.382629
W_{16}^2	0.707092	0.707092
W_{16}^3	0.382629	0.923828
W_{16}^4	0	-1
W_{16}^6	-0.707092	-0.707092
W_{16}^9	-0.923828	-0.382629

IV. PROPOSED SYSTEM

Memory-based architecture is widely adopted to design an FFT processor, also known as the single processing element (PE) approach. This design style is usually composed of a main PE and several memory units, thus the

hardware cost and the power consumption are both lower than the other architecture style. However, this kind of architecture style has long latency, low throughput, and cannot be parallelized. On the other hand, the pipeline architecture style can get rid-off the disadvantages of the foregoing style, at the cost of an acceptable hardware overhead.

In order to improve the previous works on power reduction, we propose a radix-2 64-point pipeline FFT/IFFT processor with low power consumption. The proposed architecture is composed of three different types of processing elements (PEs), a complex constant multiplier, delay-line (DL) buffers (as shown by a rectangle with a number inside), and some extra processing units for computing IFFT. Here, the conjugate for extra processing units is easy to implement, which only takes the 2's complement of the imaginary part of a complex value. The divided-by-64 module can be substituted with a barrel shifter.

In addition, for a complex constant multiplier, here it is proposed a novel reconfigurable complex constant multiplier to eliminate the twiddle-factor ROM. This new multiplication structure thus becomes the key component in reducing the chip area and power consumption of our proposed FFT/IFFT processor.

A. Processing Elements:

Based on the radix-2 FFT algorithm, three types of processing elements (PE3, PE2, and PE1) used. The functions of these three PE types correspond to each of the butterfly. First, the PE3 stage is used to implement a simple radix-2 butterfly structure only, and serves as the sub-modules of the PE2 and PE1 stages. In the figure, I_{in} and I_{out} are the real parts of the input and output data, respectively. Q_{in} and Q_{out} denote the image parts of the input and output data, respectively. As for the PE2 stage, it is required to compute the multiplication by $-j$ or 1. Note that the multiplication by -1 is practically to take the 2's complement of its input value. In the PE1 stage, the calculation is more complex than the PE2 stage, which is responsible for computing the multiplications by $-j$, $W_N^{N/8}$ and $W_N^{3N/8}$ respectively. Since $W_N^{3N/8} = -j W_N^{N/8}$ it can be given by either the multiplication by $W_N^{N/8}$ first and then the multiplication by $-j$ or the reverse of the previous calculation. Hence, the designed hardware utilizes this kind of cascaded calculation and multiplexers to realize all the necessary calculations of the PE1 stage. This manner can also save a bit-parallel multiplier for computing $W_N^{N/K}$ which further forms a low-cost hardware.

B. Bit-Parallel Multipliers:

The multiplication by $1/\sqrt{2}$ can employ a bit-parallel multiplier to replace the word-length multiplier and square root evaluation for chip area reduction. The bit-parallel operation in terms of power of 2 is given by, $output = in * \sqrt{2}/2 = in * (2^{-1} + 2^{-3} + 2^{-5} + 2^{-7} + \dots)$ (9)

If a straightforward implementation for the above equation is adopted, it will introduce a poor precision due to the truncation error, and will spend more hardware cost. Therefore, to improve the precision and hardware cost the above equation is rewritten as,

$$output = in * \sqrt{2}/2 = in * [1 + (1 + 2^{-2})(2^{-6} - 2^{-2})] \quad (10)$$

The circuit diagram of the bit-parallel multiplier is illustrated in Fig. 7. The resulting circuit uses three additions

and three barrel shift operations. The realization of complex multiplication by $W_N^{N/8}$ using a radix-2 butterfly structure with its both outputs commonly multiplied by $1/\sqrt{2}$ is shown in Fig. 8. This circuit has just been used in the PE1 stage.

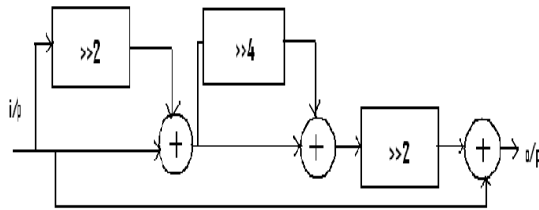


Figure 4.1. Block diagram of bit parallel multiplication by $1/\sqrt{2}$

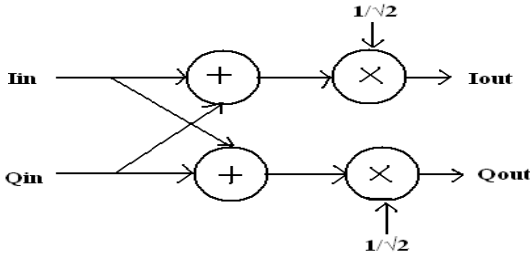


Figure 4.2. Block diagram for multiplication by $W_N^{N/8}$

C. Reconfigurable Complex Constant Multipliers:

A reconfigurable low-complexity complex constant multiplier for computing $W_N^{N/8}$ is proposed, as shown in Fig.9. This structure of this complex multiplier also adopts a cascaded scheme to achieve low-cost hardware. Here, the meaning of two input signals (I_{in} and I_{out}) and two output signals (Q_{in} and Q_{out}) are the same as the signals in the PE1 stage.

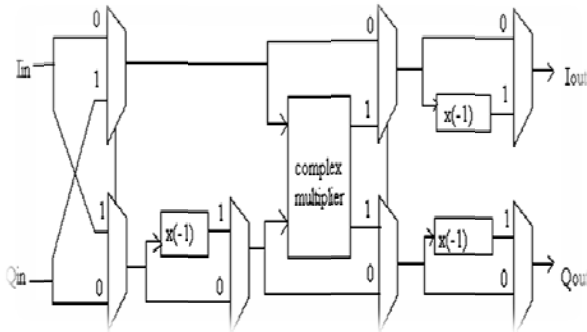


Figure 4.3. Block diagram of complex constant multiplier

This circuit is responsible for the computation of multiplication by a twiddle factor $W_N^{N/8}$, which is also an important circuit of our FFT/IFFT processor. The word-length multiplier used adopts a low-error fixed width modified Booth multiplier for hardware cost reduction. The coefficient values $i1-i8$ and $q1-q8$ are listed in Table 2, which can be used to synthesize the entire twiddle factors required in our proposed 64-point FFT processor.

Table 4.1. Co-efficient values

Co-efficient	Value	Co-efficient	Value
i1	0.7071	q1	0.7071
i2	0.7730	q2	0.6343
i3	0.8314	q3	0.5555
i4	0.8819	q4	0.4713
i5	0.9238	q5	0.3826
i6	0.9569	q6	0.2902
i7	0.9807	q7	0.1950
i8	0.9951	q8	0.0980

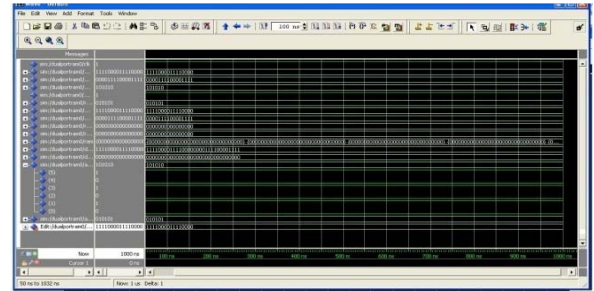


Figure 4.4 Output

V. CONCLUSION

A powerful FFT/IFFT processor which is used in many other wireless communication systems has been designed with ROM less architecture. This result shows that this design owns lower hardware cost and power consumption compared to the existing ones. Of course, the proposed scheme can also be adapted to high-point FFT applications, with a lower size of twiddle-factor ROM's.

VI. REFERENCES

- [1]. IEEE Std 802.11a, 1999, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-Speed Physical Layer in the 5 GHz band.
- [2]. IEEE 802.16, IEEE Standard for Air Interface for Fixed Broadband Wireless Access Systems, the Institute of Electrical and Electronics Engineers, Inc., June 2004.
- [3]. 3GPP LTE, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation" 3GPP TS 36.211 v8.5.0, 2008-12.
- [4]. ETSI, "Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television," ETSI EN 300 744 v1.4.1, 2001.
- [5]. J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comput., vol. 19, pp. 297-301, Apr. 1965
- [6]. Minhyeok Shin and Hanho Lee, "A High-Speed Four-Parallel Radix-24 FFT/IFFT Processor for UWB Applications," in Proc. IEEE Int. Symp. Circuits and Systems, 2008, pp. 960-963
- [7]. Wen-Chang Yeh and Chein-Wei Jen, "High-speed and low-power splitradix FFT," IEEE Transactions on Signal Processing, vol. 51, no. 3, pp. 864- 874, Mar. 2003.