Architecture and Performance Analysis of Lossless FFT in OFDM Systems

Wei-Hsin Chang and Truong Nguyen Department of Electrical and Computer Engineering, UCSD La Jolla, California 92093-0407, Contact Email: whchang@ucsd.edu

Abstract—The design of Fast Fourier Transform (FFT) architecture is one of the bottlenecks in the implementation of OFDM systems. With the recent progress on the development of lossless transform, its possible applications in communication systems have received more and more attentions. In this paper, a lossless integer FFT (IntFFT) architecture based on radix- 2^2 FFT algorithm is analyzed and implemented. By exploring the symmetric property, the overall memory usage is reduced by 27.4% for a 64-point FFT design. The variance of quantization loss for both IntFFT and conventional fixed-point FFT (FxpFFT) is derived. The signal to quantization loss ratio (SQNR) and the bit error rate (BER) performance in system level is also simulated to test the accuracy of IntFFT. Based on the simulation results, IntFFT can yield comparative performance while using less memory usage than FxpFFT designs.

I. INTRODUCTION

Transform-based signal processing is one of the most widely used techniques in current multimedia systems and communication systems such as JPEG, MPEG, orthogonal frequency demodulation multiplexing (OFDM) systems, asymmetric digital subscribe line (ADSL), digital video broadcasting (DVB) and digital audio broadcasting (DAB). During the past decade, the rapid development of waveletbased signal processing [4] and lifting decomposition [1] make discrete wavelet transform (DWT) an great advance in image processing applications [9]. Especially with lifting scheme (LS), it is possible to realize a lossless transform in actual hardware implementations. The same idea of lifting scheme also inspires different approaches of lossless transforms that eliminate quantization loss for FFT [6] and discrete cosine transform (DCT). In previous research [5], the lossless DCT has been demonstrated to yield comparative performances compared to conventional implementations. However, there is always a doubt to use lossless architectures in communication systems because the presence of noisy channel. Consider OFDM system shown in Fig. 1, the inverse discrete fourier transform (IDFT) and discrete fourier transform (DFT) pair are used to modulate and demodulate the data constellation on the sub-carriers. The input of IDFT at transmitter side is digitally modulated signals. The output of the IDFT consists of the time-domain samples to be transmitted over the real channel. Accordingly, at the receiver side, the DFT is performed. When the channel effects such as channel noise and inter-symbol interference are taken

¹This work is supported by CWC and UC matching fund from UC Discovery



Fig. 1. A simplified scheme for system using OFDM modulation.

into consideration, the received samples may be corrupted. Therefore, in this paper we will analyze the quantization loss of IntFFT and FxpFFT and simulate both designs in an OFDM system to verify the accuracy of IntFFT.

The paper is organized as follows. The previous work of IntFFT algorithm with memory reduction is reviewed in section II. In section III, the proposed architecture is described. In section IV, the analysis of quantization loss between conventional complex multipliers and LS-based multipliers are derived and compared. Section VI has two parts: the system simulation results of FxpFFT and IntFFT are illustrated first, and the hardware resource usage comparison is listed.

II. INTRODUCTION TO INTEGER FFT

IntFFT [6] is an integer-to-integer mapping transform where the concept of lifting scheme used in wavelet transform is applied to FFT algorithms. Since it only involves integer operations, lossless transform is possible. The main idea of IntFFT is to utilize lifting scheme in the conventional FFT by decomposing the original complex number multiplications into three lifting steps [6]. All the original twiddle factors in the FFT computation are presented in lifting coefficient format. Recall that no matter what kind of operation is performed in the lifting path, if the same operation is performed again in the corresponding reverse lifting path, the original input will be reconstructed without any distortion. Hence, IntFFT is able to preserve the perfect reconstruction property. Let t = c + isbe a complex number with magnitude one (i.e. |t| = 1) and $x = x_r + jx_i$, the original complex multiplication can be represented as the following matrix form:

$$tx = (cx_r - sx_i) + j(cx_i + sx_r)$$

= $\begin{bmatrix} 1 & j \end{bmatrix} \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} x_r \\ x_i \end{bmatrix} = \begin{bmatrix} 1 & j \end{bmatrix} R \begin{bmatrix} x_r \\ x_i \end{bmatrix}$ (1)



Fig. 2. Represent original twiddle factors as three lifting steps.

Furthermore, the R matrix shown in (1) can be decomposed into three lifting steps [1].

$$R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} 1 & \frac{c-1}{s} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ s & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{c-1}{s} \\ 0 & 1 \end{bmatrix}$$
(2)

Recall the definition of the twiddle factors, s and c represent $\sin \theta$ and $\cos \theta$ respectively, where θ is the angle of twiddle factors. Since the coefficients are scalar numbers with magnitude one, its inverse is simply the complex conjugates. Compared to common complex multiplier designs, the advantage of lifting decomposition toward hardware implementation is apparent: the number of real multiplications is reduced from four to three. The original complex multiplications of multiplying twiddle factors of its reverse can be represented in lifting scheme form as shown in Fig. 2.

III. PROPOSED ARCHITECTURE DESIGN

The study of high-performance VLSI architecture for FFT is an important issue for wireless LAN design since it is the critical component in the whole system. Several different FFT algorithms have been proposed during the past decades. In general, these fast algorithms divide the FFT computation into odd-half part and even-half part recursively and extract the common twiddle factors as many as possible. The number of required real additions and multiplications is usually used to compare the efficiency of different FFT algorithms. In terms of the multiplicative comparison [10], split-radix FFT is better than all the others because it has most trivial multiplications. However, since the non-trivial multiplications could appear in two successive stages, the requirement of multipliers increases when the pipeline stage design is considered. Take 16-point FFT as example, the signal dataflow graphs (DFG) of radix- 2^2 FFT is illustrated in Fig. 3. Obviously we can find, for designing of pipeline architecture, radix-22 FFT only needs one set of complex multiplier, nevertheless split-radix FFT needs two sets of complex multipliers. Although some efforts have been done [10] to share the complex multipliers between adjacent BF stages so as to reduce the usage of multipliers, it inevitably results in additional overhead on control logic. Hence, in this paper, the proposed architecture is based on radix-2² FFT.

A. Memory Strategy

The memory strategy of BF stage is an important design issue for FFT architecture. There are mainly two different



Fig. 3. DFG of 16-point radix-2² FFT.



Fig. 4. Feedback buffering strategy for BF core.

approaches: delay commutator (DC) [7] and delay feedback (DF) [3]. In our architecture, the latter is adopted and depicted in Fig. 4. In the first half cycles, the BF_i core will simply store the input samples into feedback memory. After the first N/2 cycles, the BF_i core retrieves x(n) sample from feedback memory, performs corresponding operations with the sample x(n + N/2) and then feeds the output into next BF_{i+1} core. The necessary number of memory cells is N/2, N/4 ... etc for the first, second stages ... etc. Therefore, by mathematical induction, the total required memory is N - 1.

B. Memory Reduction in OFDM system

In the widely used OFDM systems, by exploring the symmetric property, with the perfect reconstruction property of IntFFT, it is possible to reduce the memory usage of FFT at the receiver. Conventionally, the wordlength will increase by 1 bit after every BF stages to maintain enough precisions as that of input signals. However, since IntFFT only involves reversible operations, it simply needs the same wordlength of input signals as shown in Fig. 5. Therefore, in the proposed IntFFT design, the wordlength is decreased by 1 bit after every BF stages instead. Assume $\alpha = \log_2 N$ and the input samples are represented as 12-bit vectors. The following equations show the number of memory cells required in conventional designs and proposed IntFFT, respectively.

Conventional:
$$M_c = \sum_{i=1}^{\alpha} \frac{N}{2^i} (2\alpha + i)$$
 (3)

$$Proposed: \ M_p = \sum_{i=1}^{\alpha} \frac{N}{2^i} (2\alpha - i)$$
(4)

The advantages of the idea are twofold. First, obviously the overall memory usage is decreased. Second, since the wordlength involved in the operations is shorter, the critical path is also reduced since the bottleneck of most FFT designs is the delay of multipliers.

IV. ACCURACY ANALYSIS OF INTEGER FFT

In this section, the accuracy of IntFFT is theoretically analyzed and experimentally compared to FxpFFT. Since the round-off effect will only occur in multiplier stages, we put the emphasis on comparing the conventional complex multiplier to the LS-based multiplier. By modeling the quantization loss as an additive uniform distribution random variable, the variance of quantization loss of both conventional complex multipliers and its equivalent LS-based multipliers can be derived. First, for a conventional complex multiplier, assume that $x = x_r + jx_i$ represents the input sample, and t = c + jsrepresents the complex twiddle factor, the quantized output is represented as follows:

$$Q(D) = [Q(cx_r) - Q(sx_i)] + j[Q(cx_i) + Q(sx_r)].$$
 (5)

Consequently, the overall quantization loss and the variance of quantization loss can be expressed as (8), where e_i represents quantization loss introduced by multipliers. For the LS-based complex multipliers depicted in Fig. 6, the lifting coefficients are generated from corresponding twiddle factor by p[n] and q[n] where both p[n] and q[n] are real-valued discrete functions and can be expressed as follows:

$$p[n] = \frac{\cos(\theta) - 1}{\sin(\theta)} \tag{6}$$

$$q[n] = \sin(\theta) \tag{7}$$

where $\theta = \frac{-2\pi n}{N}$, and n is the index of twiddle factors. Again, we assume all random variables are all statistically independent and uniformly distributed. The variance of quantization loss, $f_L[n]$, is derived in (9). For the 64-points radix-2² FFT, $f_L[n]$ versus different decomposition types is plotted in Fig. 7. The selection criteria of decomposition are twofold: First, $f_L[n]$ should be less than 4 so as to keep the quantization loss lower than that of conventional complex multipliers. Second, for the ease of hardware implementation, the position of j term should be the same. Hence it will not lead to design overhead. In the proposed architecture, the combination of Type (a) and Type (b) is chosen. For $0 \le \theta \le \frac{\pi}{2}$, Type (a) is adopted. For $\frac{\pi}{2} < \theta < \pi$, Type (b) is used.

V. SIMULATION & COMPARISON

In order to verify the accuracy of IntFFT against conventional FxpFFT, both algorithms are implemented in Matlab for comparison. The system test flow is as follows: First, a function library of binary operators is constructed as the fundamental components. Second, the Matlab version IntFFT is implemented to verify that the PR property is preserved and the output agrees with the Verilog version design. Third, the FxpFFT is implemented and the SQNR of transformed output of both algorithms is calculated. The wordlength of



Fig. 5. Balanced memory usage of reversible transforms.



Fig. 6. The equivalent quantization loss model of LS-based complex multiplier

coefficients is set to 12 bits. 10,000 trials are made and the average SQNR is plotted in Fig. 8. Moreover, an OFDM-based wireless LAN system is constructed in Simulink for system level simulation. Multipath frequency-selective fading channel is used to evaluate the BER versus difference channel SNR values. Various modulation schemes including QPSK (1/2) and 16QAM (1/2) and 64QAM (1/2) are simulated. From Fig. 9, IntFFT performs as well as conventional FxpFFT even if the noisy channel is present.

The hardware resource comparisons of several classical FFT architecture designs are highlighted in Table I. Although the



Fig. 7. $f_L[n]$ of four equivalent decompositions.

$$E_C[|Q(D) - D|^2] = E[(e_1 - e_2)^2 + (e_3 + e_4)^2] = E[e_1^2] + E[e_1^2] + E[e_1^2] + E[e_1^2] = 4\sigma^2$$
(8)

$$E_{LS}[|Q(D) - D|^2] = E[[(1 + p[n]q[n])e_1 + p[n]e_2 + e_3]^2 + (q[n]e_1 + e_2)^2] = [(\frac{\cos(\omega n) - 1}{\sin(\omega n)})^2 + 3] \cdot \sigma^2 = f_L[n]\sigma^2 \quad (9)$$

number of complex adders and multipliers is the same to that of previous designs, proposed IntFFT architecture utilizes the symmetry of transform pairs to reduce the usage of DF memory. For a 64-pt FFT design, the DF memory usage is reduced by 27.4% due to the reduced memory scheme. After Matlab verification, the design is implemented in Verilog HDL and simulated by Verilog-XL. Then it's synthesized by Buildgates and the AP&R flow is performed by SOC Encounter with TSMC 0.18-µm 1P6M process. After AP&R process, the generated SDF (standard delay format) file is back annotated to Verilog-XL simulator to verify the correctness of proposed design. Then, the power consumption analysis is performed by VoltageStorm. In the design, delay feedback memory blocks are implemented with pre-generated memory hard macros. The final design is a pad limited design. The core size is $500\mu m \ge 500\mu m$, with core utilization is 80%. The whole chip size is $975\mu m \ge 977\mu m$, with 39 data pins, 8 power pins and 1 filler pin. The reported equivalent gate count is 17,983 gates. The estimated core power consumption is 83.56 mW.

VI. CONCLUSIONS

In this paper, based on IntFFT, a radix- 2^2 IntFFT architecture is proposed and verified. The most important feature of IntFFT is that it can yield lossless samples as well as obtain comparative accuracy. The required number of real multipliers is also reduced because the lifting scheme saves one multiplier than general complex multipliers. Moreover, compared to FxpFFT designs, the system simulations show that IntFFT-based architecture can also be adopted in a real OFDM system and yield comparative BER performance in noisy channel case.

VII. ACKNOWLEDGEMENT

The authors would like to thank their colleague, Carson K.S Pun, for his many fruitful discussions.

REFERENCES

- I. Daubechies and W. Sweldens, "Factoring Wavelet Transforms into Lifting Steps," J. Fourier Anal. Appl., Vol. 4, Nr. 3, pp. 247-269, 1998.
- [2] A. M. Despain, "Fourier transform computer using CORDIC iterations," IEEE Trans. Comput., vol. C-23, pp. 993-1001, Oct. 1974.
- [3] S. He and M. Torkelson, "Designing Pipeline FFT processor for OFDM (De)Modulation," in Proc. IEEE URSI Int. Symp. Signals, Syst., Electron., 1998, pp. 257-262.
- [4] S. Mallat, "A Wavelet Tour of Signal Processing", Second Edition, Academic Press.
- [5] T. D. Tran, "The BinDCT: Fast Multiplierless Approximation of the DCT", IEEE Signal Processing Letters, Vol. 7, No. 6, June 2000.
- [6] S. Oraintara, Y-J Chen and T. Nguyen, "Integer Fast Fourier Transform," IEEE Trans. on Signal Processing, Vol. 50, NO. 3, March 2002.
- [7] L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing. Englewood Cliffs, NJ: Prentice-Hall, 1975.

TABLE I Hardware resource usage comparisons

Design	Complex Multiplier	Complex Adder	Memory Usage (bit) ²
			000000
R2MDC [7]	$2(\log_4 N - 1)$	$4 log_4 N$	$2\alpha N + \sum_{i=1}^{\alpha} \frac{N}{2^i}(2\alpha + i)$
			i=2
R4SDF [2]	$\log_4 N - 1$	$8log_4N$	Eq(3)
R2 ² SDF [3]	$\log_4 N - 1$	$4log_4N$	Eq(3)
SRSDF [10]	$\log_4 N - 1$	$4log_4N$	Eq(3)
Proposed ³	$\log_4 N - 1$	$4log_4N$	Eq(4)

- [8] G. Strang and T. Nguyen, "Wavelets and Filter Banks", Wellesley-Cambridge Press.
- [9] D. S. Taubman and M. W. Marcellin, "JPEG2000: Image Compression Fundamentals, Standards, and Practice," Kluwer Academic Publishers, 2002.
- [10] W.C Yeh and C.W Jen, "High-Speed and Low-Power Split-Radix FFT,"IEEE Trans. on Signal Processing, March 2003



Fig. 8. The SQNR comparison between IntFFT and FxpFFT, Nc = 12.



Fig. 9. BER performance analysis of IntFFT in 802.11a wireless LAN system.

²Assume that the wordlength is increased by 1 bit for all previous designs. ³LS-based multiplier is adopted.