A High Performance Four-Parallel 128/64-point Radix-\(2^4\) FFT/IFFT Processor for MIMO-OFDM Systems

Hang Liu and Hanho Lee
School of Information and Communication Engineering
Inha University, Incheon, 402-751, Korea
mailiuhang@hotmail.com, hhlee@inha.ac.kr

Abstract—This paper presents a novel high-speed, low-complexity 128/64-point radix-\(2^4\) FFT/IFFT processor for the applications in a high-throughput MIMO-OFDM systems. The high radix radix-\(2^4\) multi-path delay feed-back (MDF) FFT architecture provides a higher throughput rate and low hardware complexity by using a four-parallel data-path scheme. The proposed processor not only supports the operation of FFT/IFFT in 128 points and 64 points but can also provide a high data processing rate by using a four-parallel data-path scheme. Furthermore, the proposed design has a less hardware complexity compared with traditional 128/64-point FFT/IFFT processors. The proposed FFT/IFFT processor was implemented using Xilinx Virtex-4 FPGA. Our proposed processor achieves a considerable performance, which has a high throughput rate of up to 560 M sample/s at 140 MHz.

I. INTRODUCTION

The growing demand of data, multimedia and communication has requested the need for generating many disparate devices into a high speed and efficiency bandwidth network capacity, with seamlessly supporting and integrating each sector’s unique requirements. The multiple-input multiple-output (MIMO) schemes have been widely studied and received great attention by both academy and industry in wireless communications application. The application of multiple antennas at both transmitter and receiver provides enhanced performance over diversity system. This technique can significantly increase the data rates of wireless system without increasing system power or bandwidth. Besides, the cost of increasing data rates is just the additional multiple antennas, multi-dimensional signal processing block and the extra system space. Now, the MIMO scheme has been adopted by the IEEE 802.11n Wireless LAN and IEEE 802.16e WiMAX standards.

Orthogonal Frequency Division Multiplexing (OFDM) is modulation method known for its capability to mitigate multipath [1]. It can provide a flat with the sub-carrier, and achieve high efficiency and ability to deal with frequency selective fading and narrowband interference. Also the flexibility using MIMO with OFDM can support the nonflat fading channels. The transceiver structure of MIMO-OFDM is depicted in Fig. 1. The trend of MIMO design in FPGA implementation demands a low device expenditure and low system clock solution. Thus, we need to concentrate on the optimization of MIMO-OFDM transceiver system algorithm and structure.

The FFT/IFFT processor is one of the kernel modules having high computational complexity in the physical layer of the MIMO-OFDM system, which is used to process multi-carrier modulation. Since the MIMO-OFDM system needs more independent channel operators and processors, the system complexity and hardware cost dramatically increase. For example, a MIMO-OFDM system with \(N\) transceiver inputs and outputs require \(N\) basebands to be operated, and therefore \(N\) FFT/IFFT processors are necessary, also the system complexity increase \(N\) times. Thus, a scheme of high-speed, low-complexity FFT/IFFT processor can apparently reduce the complexity of MIMO-OFDM systems [2][3]. And, as the data transmission rate of MIMO-OFDM systems increases, generating OFDM symbols with high data rate requires high-efficiency FFT algorithm and architecture. This paper proposes a high-speed, low-complexity 128/64-point radix-\(2^4\) FFT/IFFT processor with a four-parallel data-path and a multipath delay feed-back (MDF) structure to deal with the issues of the high-throughput and hardware complexity for MIMO-OFDM applications. It can provide a higher throughput rate and low hardware complexity.
This paper is organized as follows. Section II describes the design issues of the FFT/IFFT processor. Section III describes the proposed 128/64-point radix-2\(^4\) MDF (R2\(^4\)MDF) FFT/IFFT architecture. In Section IV, the hardware cost and throughput rate of the proposed FFT/IFFT architecture is compared with that of the traditional 128/64-point FFT/IFFT architectures. Conclusions are described in Section V.

II. DESIGN ISSUE OF THE FFT/IFFT PROCESSOR

The receiver of MIMO-OFDM system contains four RFs, four analog-to-digital converters (ADCs), four FFTs, a MIMO equalizer, four De-QAM and de-interleaver, a de-spacial parser, a de-puncturer, a channel decoder, a synchronization block, and a channel estimation block [4]. However, the hardware cost is also increased significantly, because more memory and complex multipliers are needed to allow multiple data to be operated simultaneously. Therefore, radix-2\(^4\) SDF FFT/IFFT architecture has been proposed to offers high throughput, low hardware complexity and low power consumption by reduce the number of complex multiplications [5]. In pipelined FFT hardware scheme, the multipath delay commutator (MDC) scheme can achieve higher throughput rate by using multiple data paths, while the single-path delay feed-back (SDF) scheme needs less memory and hardware complexity with the delay feed-back scheme. Therefore, MDF FFT/IFFT architecture has been proposed to provide high throughput, low hardware complexity, and low power consumption [4]. Proposed radix-2\(^4\) MDF (R2\(^4\)MDF) architecture can provide higher throughput rate with minimal hardware cost by combining the features of MDC and SDF.

III. PROPOSED FFT/IFFT ARCHITECTURE

The radix-2\(^4\) algorithm is described in detail in [5]. The algorithm can take complex constant multiplier instead of programmable complex multiplier. The Canonc Signed Digit (CSD) constant multiplier contains the fewest number of non-zero bits, so it can be reduce the area and power consumption [6]. Fig. 2 shows the architecture of proposed four-parallel data-path 128/64-point R2\(^4\)MDF FFT/IFFT processor. The proposed design can support the operation of FFT/IFFT in 128-point or 64-point. It consists of memory units, butterfly units (BF_64, BF_128, BF1, BF2), complex Booth multipliers, CSD complex constant multipliers, multiplexers and adders. Also the four-parallel design requests four inputs and outputs, which operate the real and imaginary data separately. Butterfly unit is the kernel of a FFT processor. For the multiply factors \(j\) in different butterfly steps, we design two kinds of similar butterfly units. The BF1 stores all of N/2-th input data in different butterfly steps, we design two kinds of similar butterfly units. Therefore, radix-2\(^4\) MDF (R2\(^4\)MDF) architecture can provide higher throughput rate with minimal hardware cost by combining the features of MDC and SDF.
by new input data. And then the subtracted output data \(x[n]-x[N/2+n]\) are stored in the location of previous input data \(x[n]\). When the new set of input data are fed to BF1 after the N-th input operation, the new input data are stored in the RAM and \((x[n]-x[N/2+n])\) RAM. The BF2 architecture is almost same with the BF1 architecture except the operation of 3N/4-th input data multiplying of \(-j\). The output results of butterfly units are complex addition and complex subtraction of two input data \(x[n]\) and \(x[N/2+n]\), where \(N\) is the value of point, as shown in Fig. 3 and 4. The BF_64 block, which is MDC hardware design, consists of adders, subtractors, multiplexers and RAMs, which can store complex data. The architecture BF_64 block is shown in Fig. 5. We design the individual processor module for each 64 and 128-point with four-parallel data-path through the different complex data storage. It operates as following steps: at the beginning 16 cycles, the first 32 data are stored in the register file. When the next N/2-th input data are stored in the register file, the previously stored input data \(x[2n-1]\) are read from the register file and generates the outputs. Then these outputs are added and subtracted by \(x[2n-1]\) and the new input data \(x[N/2+2n-1]\) simultaneously. And the input data \(x[N/2+2n]\) written into the module is stored in the location where the previous \(x[2n-1]\) data was. After \(x[N]\) data is stored in the register file, it write out data \(x[2n]\) and \(x[N/2+2n]\), those are added and subtracted in the module at the next cycle simultaneously.

Based on the radix-2^4 FFT algorithm, after the BF_32 block, it will be multiplied with trivial constant multipliers. The radix-2^4 FFT algorithm with four-parallel data-path architecture has fewer multipliers than other schemes of lower radix FFT algorithm. Thus, this algorithm can reduce the degree of multiplicative complexity efficiently [6]. The twiddle factors, \(W(8), W(16), W(24),\) and \(W(48)\) correspond to the \(\cos(\pi/8)\) and \(\sin(\pi/8)\) by using the formula of Trigonometric Functions. Table I shows the twiddle factors, which illustrates the 10-bits coefficients in the decimal representation, the 2’s complement representation, and the CSD representation. To efficiently compensate the quantization error, the truncated bits are divided into two groups (major group and minor group) depending upon their effects on the quantization error. The error compensation bias is first expressed in terms of the truncated bits in the major group. The total CSD complex multiplier block consists of eight CSD constant multipliers, 2’s complement logics, and multiplexers as shown in Fig. 6. When the real and imaginary values of twiddle factors are same, the two CSD constant multipliers are used and theirs two outputs are added to generate the output of the CSD complex multiplier. Otherwise, the CSD constant multipliers are used for the multiplication of input and twiddle factors. If inputs don’t need to multiply with twiddle factor, the output is generated from the input directly.

For the four-parallel approach to implement the radix-2^4 FFT algorithm, we design the four-parallel data path complex Booth multipliers module. This kind of Booth multiplier needs a ROM to store the twiddle factor. The only 1/8 period of cosine and sine twiddle factors are needed, which are stored in ROM [7]. To reduce the truncation error in the fixed-width multiplier, the Dadda reduction network with error-compensation circuit [8][9] was used, as shown in Fig. 7. This method can compensate the truncation error of fixed-width

---

### Table I. CSD Binary Representation of Twiddle Factor

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>Decimal</th>
<th>2’s Comp.</th>
<th>CSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\sin(\pi/8))</td>
<td>0.3827</td>
<td>0011000011</td>
<td>0100000100</td>
</tr>
<tr>
<td>(\cos(\pi/8))</td>
<td>0.9239</td>
<td>0111011001</td>
<td>1000100100</td>
</tr>
</tbody>
</table>

---

![Figure 5. Block diagram of BF_64.](image)

![Figure 6. CSD complex constant multiplier (128-point).](image)

![Figure 7. Dadda reduction network with error-compensation circuit.](image)
IV. IMPLEMENTATION AND PERFORMANCE

The appropriate word length in the proposed processor is determined by a fixed-point simulation before hardware implementation. Based on the simulation results, 10 word lengths of the proposed FFT/IFFT architecture were determined in both real and imaginary parts. In addition, the SQNR of the proposed FFT/IFFT processor is about 30 dB. After the appropriate word length was chosen, the FFT/IFFT processor was implemented using Xilinx Virtex-4 FPGA. Table II shows performance comparisons between the proposed processor and the traditional FFT/IFFT processors.

From the comparison of the implementation results, the four-parallel 128/64-point R2^MDF FFT/IFFT processor consists of 5,633 slices, and the highest throughput rate of the proposed architecture is as high as 560 M sample/s at 140 MHz.

V. CONCLUSION

In this paper, a four-parallel data-path pipelined 128/64-point radix-2^M DDF FFT/IFFT processor has been proposed. In the proposed architecture, high-speed data processing and low hardware complexity can be achieved due to application of a fixed-width Booth multiplier with a Dadda reduction network, which maintains the input and output at 10-bit width at 30dB SQNR. Furthermore, the number of complex Booth multipliers and memories are effectively reduced by using the radix-2^M FFT algorithm. The performance results show that the data processing rate is as high as 560 Msample/s while requiring small hardware complexity. The proposed architecture is expected to be incorporated in MIMO-OFDM systems such as IEEE 802.11n WLAN, IEEE 802.16e mobile WiMAX and 4G.

ACKNOWLEDGMENT

This research was supported by the Ministry of Knowledge Economy, Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment) (IITA-2008-C1090-0801-0019).

REFERENCES