A Low-Complexity 128-Point Mixed-Radix FFT Processor for MB-OFDM UWB Systems

Sang-In Cho and Kyu-Min Kang

In this paper, we present a fast Fourier transform (FFT) processor with four parallel data paths for multiband orthogonal frequency-division multiplexing ultra-wideband systems. The proposed 128-point FFT processor employs both a modified radix-2^4 algorithm and a radix-2^3 algorithm to significantly reduce the numbers of complex constant multipliers and complex booth multipliers. It also employs substructure-sharing multiplication units instead of constant multipliers to efficiently conduct multiplication operations with only addition and shift operations. The proposed FFT processor is implemented and tested using 0.18 µm CMOS technology with a supply voltage of 1.8 V. The hardware- efficient 128-point FFT processor with four data streams can support a data processing rate of up to 1 Gsample/s while consuming 112 mW. The implementation results show that the proposed 128-point mixed-radix FFT architecture significantly reduces the hardware cost and power consumption in comparison to existing 128-point FFT architectures.

Keywords: Fast Fourier transform (FFT), mixed-radix, complex constant multiplier (CCM), substructure-sharing multiplication unit (SMU), ultra-wideband (UWB).

I. Introduction

Ultra-wideband (UWB) systems supporting various data rates from tens of Mb/s to hundreds of Mb/s are very suitable for application to short range wireless communications because they can share the frequency band with existing narrowband systems [1]-[3]. One of the candidate schemes for the high-speed UWB physical layer (PHY) is a multiband orthogonal frequency-division multiplexing (MB-OFDM) scheme. One OFDM symbol in the MB-OFDM UWB system consists of 128 subcarriers and 37 zero samples. The 128 subcarriers are composed of 100 data subcarriers, 12 pilot subcarriers, 10 guard subcarriers, and 6 null subcarriers. Therefore, the fast Fourier transform (FFT) processor of the MB-OFDM UWB system conducts a 128-point FFT operation, where the sampling frequency is 528 MHz and the subcarrier frequency spacing is 4.125 MHz. Although the FFT period is 242.42 ns, the 128-point FFT operation is allowed to be performed within 312.5 ns because a length-37 zero-padded suffix duration (70.08 ns) is added in one OFDM symbol [3].

Many FFT architectures have been developed over the last three decades. Recently, several parallel data-path pipelined FFT processors for UWB applications have been developed [4]-[9]. A 128-point mixed-radix FFT algorithm with a four-data-path approach, including radix-2 and radix-2^3 FFT algorithms, was presented in [4] to reduce the number of complex multiplications. When the 128-point FFT algorithm is broken into three successive FFT algorithms, that is, one radix-2 FFT algorithm and two radix-2^3 FFT algorithms, the hardware cost of complex multipliers in the mixed-radix multipath delay feedback (MRMDF) FFT processor comes to be only 44.8% of that in a split-radix multipath delay commutator (SRMDC) FFT processor [4], [10]. By modifying

Manuscript received Apr. 22, 2009; revised Sept. 8, 2009; accepted Nov. 19, 2009.

This work was supported by the IT R&D program of KCC/KEIT [A Study on the Radio Requirements of Coexistence for Dynamic Spectrum Access].

Sang-In Cho (phone: +82 42 860 3955, email: sicho@etri.re.kr) and Kyu-Min Kang (corresponding author, phone: +82 42 860 6703, email: kmkang@etri.re.kr) are with the Broadcasting & Telecommunications Convergence Research Laboratory, ETRI, Daejeon, Rep. of Korea.

doi:10.4218/etrij.10.0109.0232

the approach proposed by K. Maharatna and others in [11], Y.W. Lin and others in [4] efficiently realized nontrivial complex multipliers, at the fourth stage among seven stages for the 128-point FFT operation, with nine hard-wired constant units. Chakraborty and others proposed a hardware-efficient complex constant multiplier (CCM) structure in [7]. Although alternative FFT architectures for UWB applications have also been discussed in [8] and [9], the hardware cost is still high due to several nontrivial complex multiplications needed at two stages for the 128-point FFT operation.

To further reduce the hardware complexity and power consumption, Cho and others recently presented a four-parallel data-path 128-point mixed-radix decimation-in-frequency (DIF) FFT processor operating at over 132 MHz in [5]. In the proposed FFT processor, nontrivial complex multiplication operations are only needed at the fourth stage by breaking up the 128-point FFT algorithm into two FFT algorithms, namely, radix-2^4 FFT and radix-2^3 FFT algorithms. Because a relatively large number of constant multipliers are required to implement twiddle factors (TFs) at the end of each stage in a conventional radix-2^4 FFT architecture [12], a modified radix-2^4 FFT structure without constant multipliers at the third stage is presented. However, the proposed FFT architecture was not fully analyzed in [5]. There were also mistakes in the figures of [5]. In this paper, we present mathematical formulation and analysis of the proposed 128-point mixed-radix FFT algorithm. Detailed characteristics of the proposed FFT processor are also analyzed. The amended figures of the signal flow graph, butterfly units (BUs), and CCMs of the proposed FFT processor are given. We compare the hardware complexity of the proposed FFT processor and several existing 128-point FFT architectures with four parallel data paths. Multiplication units using a substructure-sharing scheme are additionally suggested to efficiently implement the constant coefficient multipliers with shift operations and additions [13], [14].

The organization of this paper is as follows. The mathematical formulations of the 128-point mixed-radix FFT algorithm are given in section II. In section III, we describe the proposed FFT architecture with four parallel data paths. The hardware complexity of the proposed FFT architecture is compared with that of the existing 128-point FFT architectures for MB-OFDM UWB systems in section IV. Conclusions are given in section V.

II. 128-Point Mixed-Radix FFT Algorithm

Given a length-N complex input sequence $x(n)$, its discrete Fourier transform (DFT) can be described as

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

where $W_N^{nk}=e^{-j(2\pi nk/N)}$ is the TF, k is a frequency index, and n is a time index. As reported in many works [4]-[12], a hardware-efficient mixed-radix FFT algorithm should be employed to reduce the number of complex multiplications because the 128-point FFT is not at a power of 4 or 8. In this section, we present a modified radix-2^4 DIF FFT algorithm for stages 1 to 4 and a radix-2^3 DIF FFT algorithm for stages 5 to 7.

1. Modified Radix-2^4 FFT Algorithm

To derive a modified radix-2^4 DIF FFT algorithm, consider the first 4 steps of the decomposition of an N-point FFT (N=128). By a five-dimensional linear index map, indices k and n are denoted by

$$k=k_1+2k_2+4k_3+8k_4+16\bar{k}_5, \tag{2}$$
$$k_1,k_2,k_3,k_4=0,1;\quad \bar{k}_5=0,\cdots,\frac{N}{16}-1,$$

$$n=\frac{N}{2}n_1+\frac{N}{4}n_2+\frac{N}{8}n_3+\frac{N}{16}n_4+\bar{n}_5, \tag{3}$$
$$n_1,n_2,n_3,n_4=0,1;\quad \bar{n}_5=0,\cdots,\frac{N}{16}-1.$$

Using (2) and (3), (1) can be rewritten as

$$\begin{aligned}&X(k_1+2k_2+4k_3+8k_4+16\bar{k}_5)\\&=\sum_{\bar{n}_5=0}^{\frac{N}{16}-1}\sum_{n_4=0}^{1}\sum_{n_3=0}^{1}\sum_{n_2=0}^{1}\sum_{n_1=0}^{1}x\left(\frac{N}{2}n_1+\frac{N}{4}n_2+\frac{N}{8}n_3+\frac{N}{16}n_4+\bar{n}_5\right)\\&\quad\cdot W_N^{\left(\frac{N}{2}n_1+\frac{N}{4}n_2+\frac{N}{8}n_3+\frac{N}{16}n_4+\bar{n}_5\right)(k_1+2k_2+4k_3+8k_4+16\bar{k}_5)}\\&=\sum_{\bar{n}_5=0}^{\frac{N}{16}-1}H_{N/16}(\bar{n}_5,k_1,k_2,k_3,k_4)W_N^{\bar{n}_5(k_1+2k_2+4k_3+8k_4)}W_{N/16}^{\bar{n}_5\bar{k}_5}.\end{aligned} \tag{4}$$

After some straightforward calculation, we have the fourth butterfly unit as

$$\begin{aligned}H_{N/16}(\bar{n}_5)&=H_{N/16}(\bar{n}_5,k_1,k_2,k_3,k_4)\\&=H_{N/8}(\bar{n}_5)+W_{16}^{(k_1+2k_2+4k_3)}H_{N/8}\left(\bar{n}_5+\frac{N}{16}\right)W_2^{k_4},\end{aligned} \tag{5}$$

where the third butterfly unit $H_{N/8}(n)$, the second butterfly unit $H_{N/4}(n)$, and the first butterfly unit $H_{N/2}(n)$ are obtained by

$$\begin{aligned}H_{N/8}(n)&=H_{N/8}(n,k_1,k_2,k_3)\\&=H_{N/4}(n)+W_8^{(k_1+2k_2)}H_{N/4}\left(n+\frac{N}{8}\right)W_2^{k_3},\end{aligned} \tag{6}$$

$$H_{N/4}(n) = H_{N/4}(n,k_1,k_2)$$
$$= H_{N/2}(n) + \underbrace{W_4^{k_1}}_{\text{TF for stage 1}} \cdot H_{N/2}(n+\frac{N}{4})W_2^{k_2}, \quad (7)$$

$$H_{N/2}(n) = H_{N/2}(n,k_1) = x(n) + x(n+\frac{N}{2})W_2^{k_1}. \quad (8)$$

In the conventional radix-2^4 FFT architecture [12], a relatively large number of multipliers are needed to implement the TFs, $W_{16}^{(k_1+2k_2+4k_3)}$, at the end of the third stage. To effectively eliminate multipliers in the third stage of the conventional radix-2^4 FFT architecture, we move some parts, $W_{16}^{(k_1+2k_2)}$, of the TFs at the end of the third stage to the end of the second stage. Then, the forth butterfly unit becomes

$$H_{N/16}(\bar{n}_5) = \tilde{H}_{N/8}(\bar{n}_5) + \underbrace{W_4^{k_3}}_{\text{TF for stage 3}} \cdot \tilde{H}_{N/8}(\bar{n}_5 + \frac{N}{16})W_2^{k_4}, \quad (9)$$

where the third butterfly unit $\tilde{H}_{N/8}(\bar{n}_5)$ is expressed as

$$\tilde{H}_{N/8}(\bar{n}_5) = \underbrace{W_{16}^{\left\lfloor \frac{n}{N/16} \right\rfloor (k_1+2k_2)}}_{\text{TF for stage 2}}$$
$$\cdot \left\{ H_{N/4}(n) + \underbrace{W_8^{(k_1+2k_2)}}_{\text{TF for stage 2}} H_{N/4}(n+\frac{N}{8})W_2^{k_3} \right\}. \quad (10)$$

Note that $\lfloor \cdot \rfloor$ is the floor function, which returns the largest integer less than or equal to its argument value.

2. Radix-2^3 FFT Algorithm

In this subsection, we further decompose the butterfly of radix-8 into three stages by adopting a radix-2^3 FFT algorithm. Let

$$G_{N/16}(\bar{n}_5) = G_{N/16}(\bar{n}_5,k_1,k_2,k_3,k_4)$$
$$= H_{N/16}(\bar{n}_5) \underbrace{W_N^{\bar{n}_5(k_1+2k_2+4k_3+8k_4)}}_{\text{TF for stage 4}}, \quad (11)$$

and

$$\bar{k}_5 = k_5 + 2k_6 + 4k_7, \quad k_5,k_6,k_7 = 0,1, \quad (12)$$

$$\bar{n}_5 = 4n_5 + 2n_6 + n_7, \quad n_5,n_6,n_7 = 0,1. \quad (13)$$

Using (11)-(13), (4) can be rewritten as

$$X(k_1+2k_2+4k_3+8k_4+16k_5+32k_6+64k_7)$$
$$= \sum_{n_7=0}^{1}\sum_{n_6=0}^{1}\sum_{n_5=0}^{1} G_{N/16}(4n_5+2n_6+n_7)$$
$$\cdot W_8^{(4n_5+2n_6+n_7)(k_5+2k_6+4k_7)}$$
$$= G_{N/64}(0) + \underbrace{W_8^{(k_5+2k_6)}}_{\text{TF for stage 6}} G_{N/64}(1)W_2^{k_7}, \quad (14)$$

where

$$G_{N/64}(n) = G_{N/64}(n,k_5,k_6)$$
$$= G_{N/32}(n) + \underbrace{W_4^{k_5}}_{\text{TF for stage 5}} \cdot G_{N/32}(n+\frac{N}{64})W_2^{k_6}, \quad (15)$$

$$G_{N/32}(n) = G_{N/32}(n,k_5)$$
$$= G_{N/16}(n) + G_{N/16}(n+\frac{N}{32})W_2^{k_5}. \quad (16)$$

We break up the 128-point DFT into a 16-point DFT and an 8-point DFT, where the 16-point and 8-point DFTs are implemented by applying the modified radix-2^4 FFT algorithm and radix-2^3 FFT algorithm, respectively.

Note that the inverse FFT (IFFT) of a length-N complex sequence $x(n)$ can be obtained by

$$x(n) = \frac{1}{N}\left\{ \sum_{k=0}^{N-1} X^*(k)W_N^{nk} \right\}^*. \quad (17)$$

The IFFT can be performed by taking the complex conjugate of the input data first and then the outgoing data without changing any coefficients in the original FFT algorithm [4].

III. Four-Parallel Data-Path FFT Architecture

1. Proposed Four-Parallel Data-Path Mixed-Radix FFT Architecture

Because the sampling rate of the analog-to-digital (A/D) converter is 528 MHz in the MB-OFDM UWB system, it is not easy to design a receiver structure with a single data-path using current CMOS process technologies. A four-parallel data-path receiver structure including an FFT block and a Viterbi decoder can be considered to limit the system clock of the baseband modem core to a maximum of 132 MHz for practical VLSI implementation [15], [16]. In this paper, we propose a hardware-efficient 128-point mixed-radix FFT architecture with four data paths to meet the high-speed requirements. The signal flow graph of the proposed four-parallel data-path 128-point FFT processor is shown in Fig. 1, where the input sequence is broken into four parallel data streams. The order of the four parallel input sequences of the proposed FFT processor is $x(4m)$, $x(4m+1)$, $x(4m+2)$, and $x(4m+3)$, where $m = 0,1,\cdots,31$. The radix-2 butterfly unit is simplified as shown in Fig. 2. Figure 3 shows a block diagram of the proposed four-parallel data-path 128-point FFT processor. The proposed FFT architecture consists of butterfly units (BU1, BU2, and BU3), complex constant multipliers (CCM1, CCM2, and CCM3), complex booth multipliers

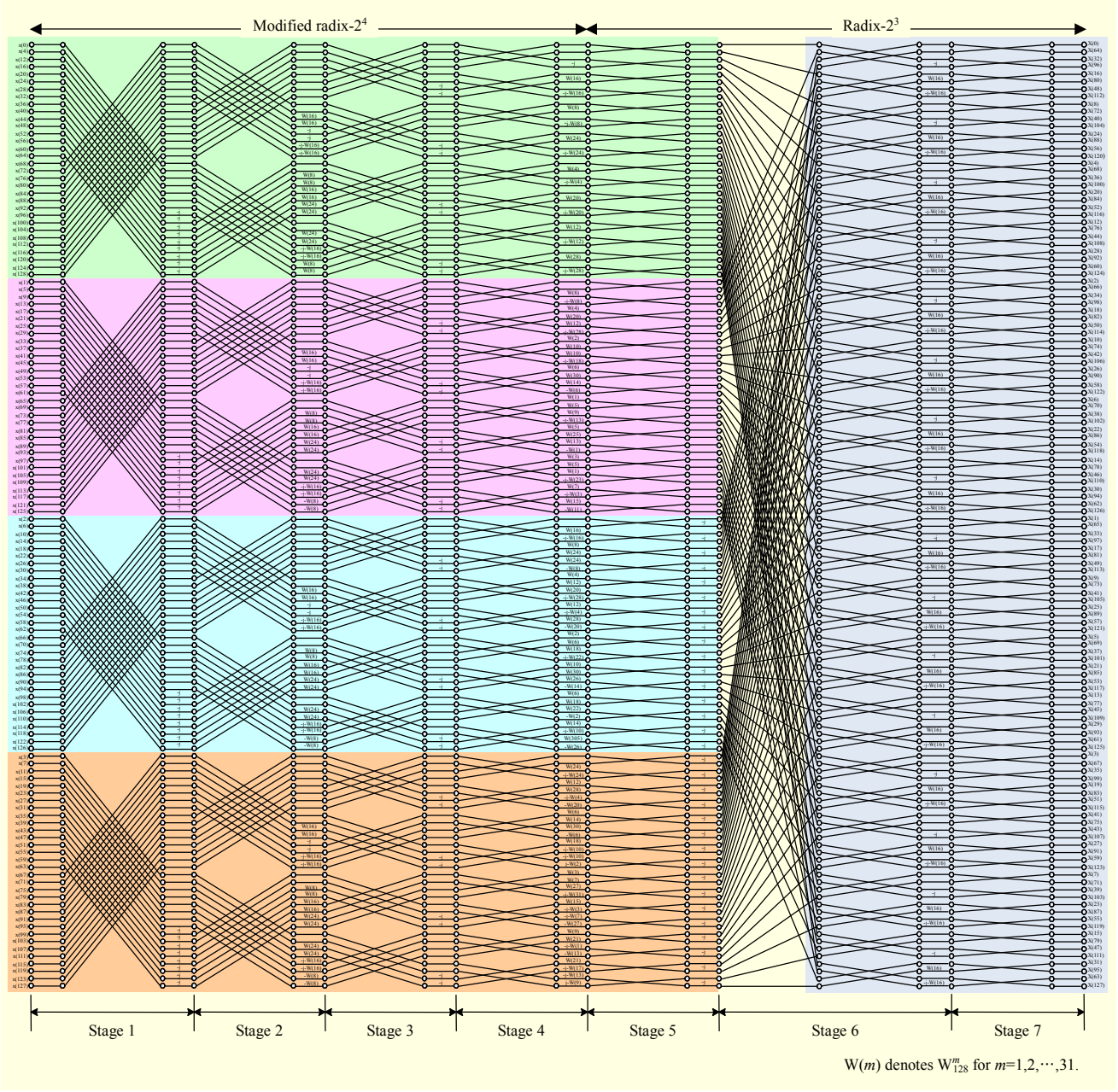


Fig. 1. Signal flow graph of the proposed four-parallel data-path 128-point mixed-radix FFT processor.

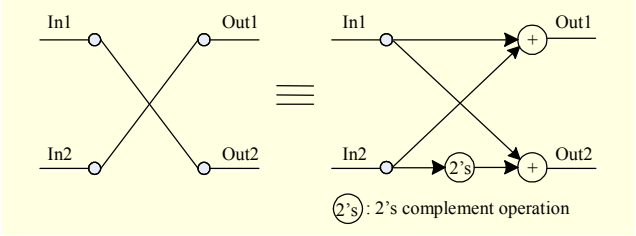


Fig. 2. Block diagram of the radix-2 butterfly unit.

(CBMs), and registers [5], [17]. As discussed in section II, the proposed FFT architecture is based on both the modified radix-2^4 and the radix-2^3 DIF FFT algorithms in order to reduce the number of multipliers. The proposed FFT architecture actually requires multipliers in three stages, namely, stages 2, 4, and 6. The other stages performing $-j$ multiplication arithmetic can be implemented by simply exchanging the imaginary value with the 2's complement of the real value without actual multiplication operation (see Fig. 4(b)).

2. Butterfly Units

The proposed FFT architecture employs three kinds of

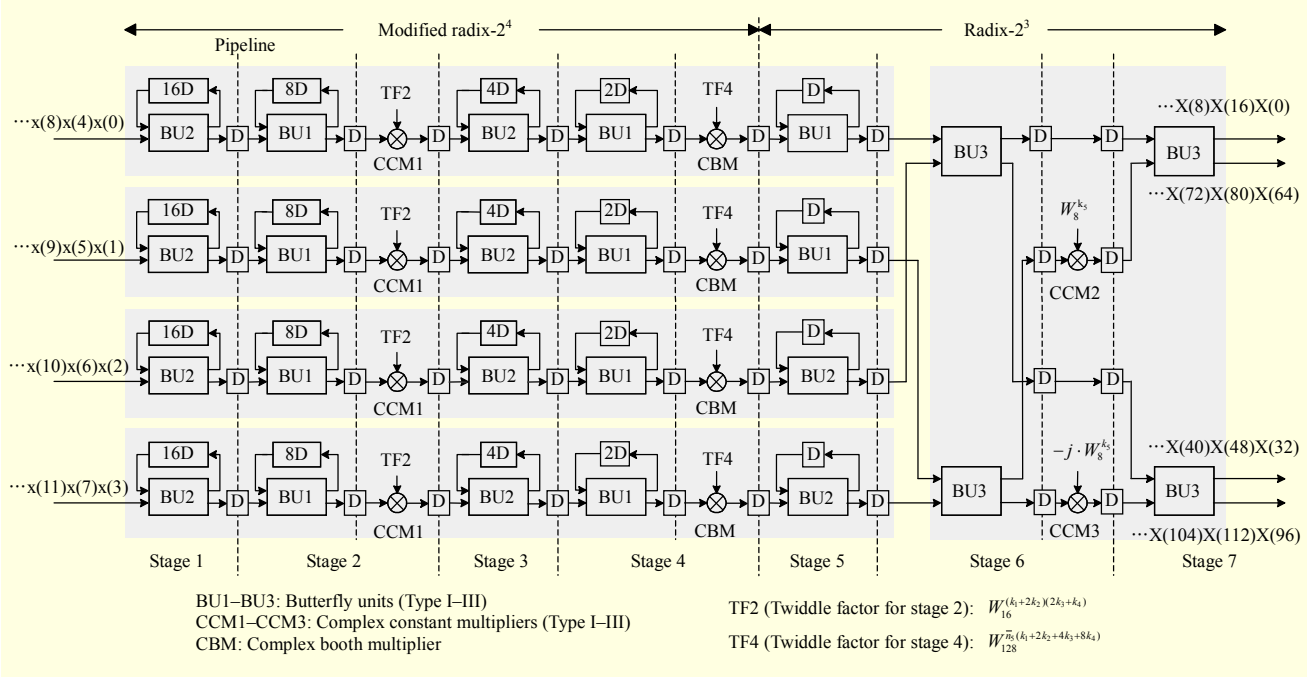


Fig. 3. Block diagram of the proposed four-parallel data-path 128-point mixed-radix FFT processor.

butterfly units (BU1, BU2, and BU3). The butterfly units perform complex addition and complex subtraction with the two complex data inputs as shown in Figs. 4(a) to (c). A complex input from the first-in-first-out (FIFO) buffer and an incoming complex input are utilized to conduct complex addition and complex subtraction in the BU1 of Fig. 4(a). One of the two complex outputs in the BU1 is stored in the FIFO buffer and the other output is passed to the next stage. The BU2 in Fig. 4(b) is constructed by adding a $-j$ multiplication unit at the end of the BU1. The BU3 of Fig. 4(c) is a conventional radix-2 butterfly unit.

3. Complex Constant Multipliers

Figures 5(a) to (c) show three kinds of CCMs used for the proposed FFT architecture. Four CCM1s are employed in stage 2, and one CCM2 and one CCM3 are employed in stage 6 for the proposed FFT architecture, while four nontrivial multipliers (CBMs) are employed in stage 4. Seven kinds of TFs are needed at the end of stage 2 in the proposed FFT architecture. In the CCM1 of stage 2, the multiplication operations of the complex input and the TFs, 1, W_8^1, $-j$, $-jW_8^1$, W_{16}^1, W_{16}^3, and $-W_{16}^1$, are conducted using four control signals. The TF selection methods in CCM1, CCM2, and CCM3 with control signals are given in Table 1. Note that the seven TFs correspond to the trigonometric functions of 1, $-j$, $\cos(\pi/8)$, $\sin(\pi/8)$, and $\cos(\pi/4)$. CCM1 is composed of six real multipliers, three 2's complement logics, two real adders, and ten multiplexers. In Fig. 5(a), when the twiddle factor is $\pm W_{16}^1$ or W_{16}^3, four constant coefficient fixed-width multipliers employing $\cos(\pi/8)$ or $\sin(\pi/8)$ are utilized, whereas two constant coefficient fixed-width multipliers employing $\cos(\pi/4)$ are used when the twiddle factor is W_8^1 or $-jW_8^1$. The multiplication output of CCM2 in Fig. 5(b) is calculated by $\mathrm{In1}\times\{\cos(k_5\pi/4)-j\sin(k_5\pi/4)\}$ with k_5=0 or 1. The multiplication output of CCM3 in Fig. 5(c) is equivalent to the output of the CCM2 multiplied by $-j$. As discussed in [4], CCM2 or CCM3 with 10-bit word length can be implemented by using ten real adders and two multiplexers. In CCM1, six real multipliers can also be implemented using 24 real adders and shift operations. Accordingly, CCM1 can be implemented using 26 real adders and 10 multiplexers. The CCM1 architecture is approximately three times more complex than the CCM2 or CCM3 architecture.

In many FFT processors, multipliers are implemented so that the resultant bit width of the multiplication output remains the same as that of their input. Accordingly, a round-off error may occur by shortening the bit width of the multiplication output. A fixed-width modified booth multiplier in [17] and a fixed-width canonic signed digit multiplier in [18] use error compensation bias schemes to efficiently compensate for the round-off error. Note that the CBM employed in stage 4 of the proposed FFT architecture is composed of two booth encoders, four partial product generators, several adders, and a read-only memory (ROM), which is detailed in [6] and [17].

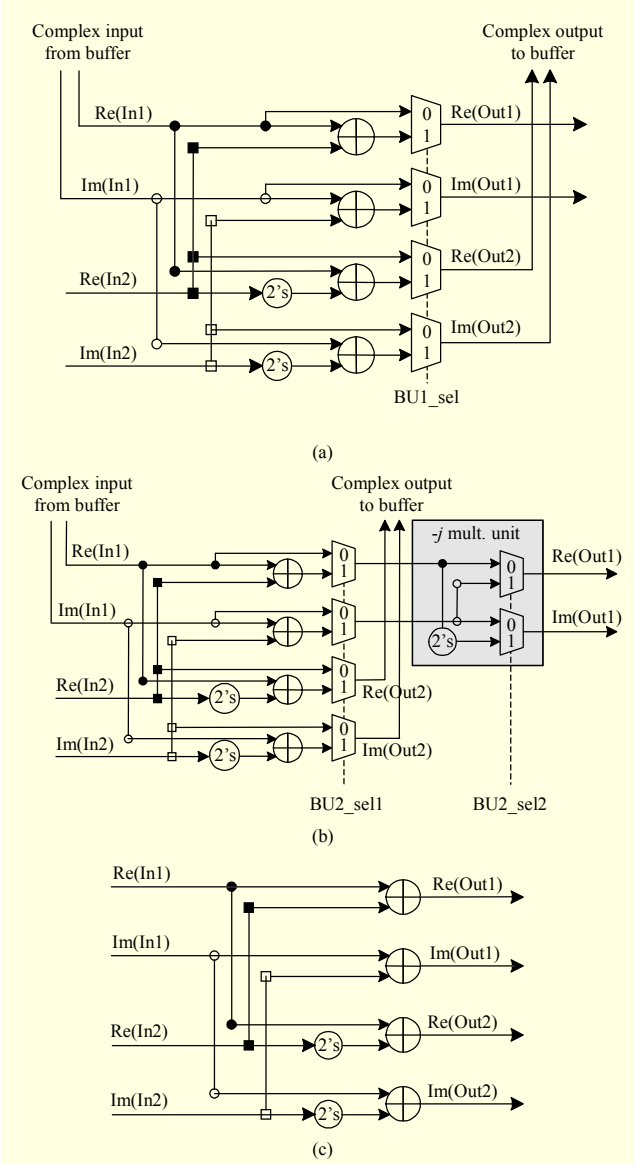


Fig. 4. Butterfly units: (a) type I (BU1), (b) type II (BU2), and (c) type III (BU3).

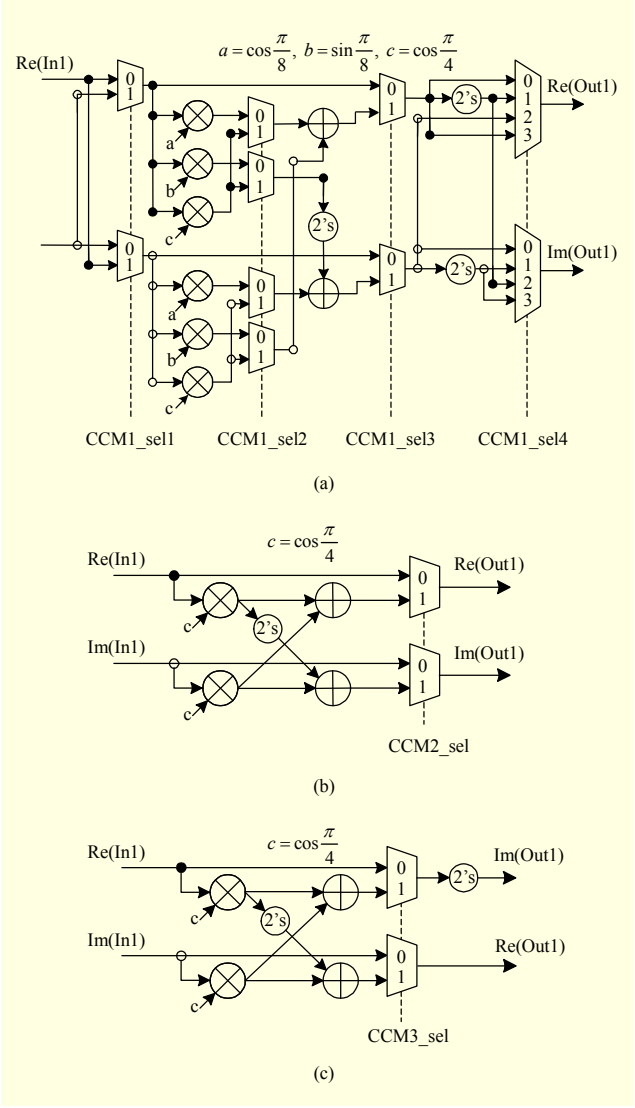


Fig. 5. Complex constant multipliers: (a) type I (CCM1), (b) type II (CCM2), and (c) type III (CCM3).

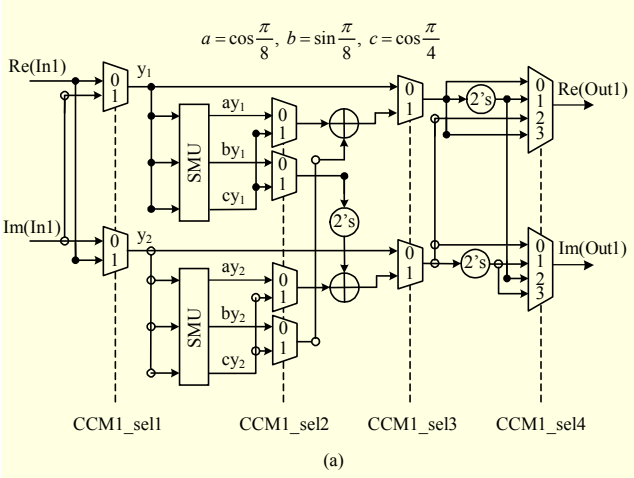
Table 1. Selection of the twiddle factors in CCM1, CCM2, and CCM3.

Twiddle factor	1	W_8^1	$-j$	$-jW_8^1$	W_{16}^1	W_{16}^3	$-W_{16}^1$
CCM1_sel1	0	0	0	0	0	1	0
CCM1_sel2	x†	1	x	1	0	0	0
CCM1_sel3	0	1	0	1	1	1	1
CCM1_sel4	0	0	2	2	0	3	1
CCM2_sel1	0	1	-	-	-	-	-
CCM3_sel1	-	-	0	1	-	-	-

† x denotes don't care value.

4. Substructure-Sharing Multiplication Units

Because six real multipliers are needed to implement CCM1 as shown in Fig. 5(a), the hardware complexity of CCM1 is rather high. In this paper, we propose an enhanced CCM1 with two substructure-sharing multiplication units (SMUs), shown in Fig. 6, to reduce the hardware complexity of CCM1. The SMU of Fig. 6(a) is utilized for the multiplication operations of a real input value and three constant coefficients, $\cos(\pi/8)$, $\sin(\pi/8)$, and $\cos(\pi/4)$. These three multiplication operations can be performed by simply using six additions and eight shift operations as shown in Fig. 6(b) if the proposed FFT processor is implemented with a 10-bit word length. Figure 7 shows an SMU for the enhanced CCM2 and CCM3. In 10-bit word



Coefficients	Decimal	2's complement	2's complement decomposition
$a=\cos\frac{\pi}{8}$	0.9239	0 1 1 1 0 1 1 0 0 1	0 1 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1
$b=\sin\frac{\pi}{8}$	0.3827	0 0 1 1 0 0 0 0 1 1	0 0 1 1 0 0 0 0 1 1
$c=\cos\frac{\pi}{4}$	0.7071	0 1 0 1 1 0 1 0 1 0	0 1 0 0 0 0 1 0 0 0 0 0 1 1 0 0 1 1 0

Fig. 6. Enhanced complex constant multiplier: (a) enhanced CCM1 and (b) substructure-sharing multiplication unit (SMU) for the enhanced CCM1.

Coefficients	Decimal	2's complement
$c=\cos\frac{\pi}{4}$	0.7071	0 1 0 1 1 0 1 0 1 0

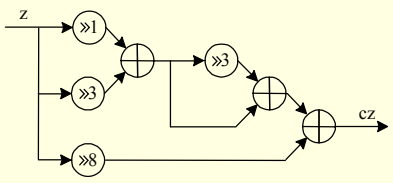


Fig. 7. Substructure-sharing multiplication unit (SMU) for the enhanced CCM2 and CCM3.

length implementation, by employing the SMU scheme, CCM2 or CCM3 can be designed using only eight adders and two multiplexers. As such, the hardware complexity of CCM1, CCM2, and CCM3 can be significantly reduced using the proposed multiplierless multiplication units with the substructure-sharing scheme.

Table 2. Implementation results of the proposed FFT processor.

Word length	8 bits	10 bits	12 bits
SQNR (dB)	24	35	47
No. of gates[1)]	71,250	80,100	88,200
Operating speed (MHz)	272	250	225
Processing rate (Msample/s)	1,088	1,000	900
Power (mW)[2)]	98	112	122

1) Based on 2×1 NAND gates.
2) Power consumption is estimated by Synopsys' Power Compiler.

IV. Implementation Results

We determined the internal word length of the proposed FFT processor using a fixed-point simulation with MATLAB before hardware implementation. After the word length of the proposed FFT processor was chosen, the FFT architecture was modeled in Verilog HDL and functionally verified using a ModelSim simulator. Then, the FFT architecture was synthesized with the appropriate time and area constraints using the Synopsys Design Compiler. Note that the FFT processor was implemented and tested using Samsung 0.18 μm CMOS technology and a standard cell library. Table 2 compares the implementation results of the proposed FFT processor for three internal word lengths. The signal-to-quantization noise ratio (SQNR) of the proposed FFT processor is about 24 dB when the word length is 8 bits, and the SQNR of the proposed FFT processor is about 47 dB when the word length is 12 bits. The hardware cost and power consumption of the proposed FFT processor are increased as the internal word length increases, whereas the operation clock

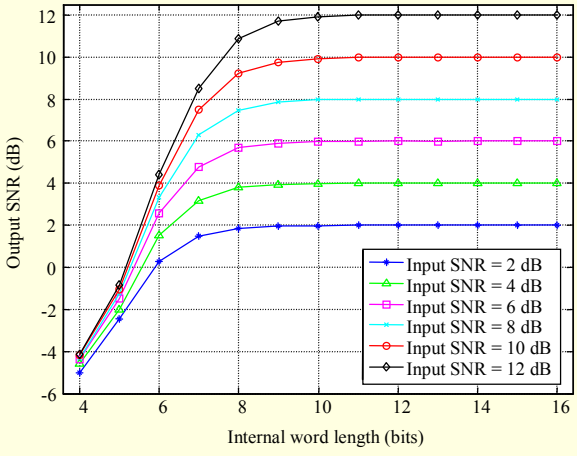


Fig. 8. Output SNR for a fixed input SNR with various internal word lengths in the proposed FFT processor.

Table 3. Comparison of the proposed and existing 128-point FFT architectures.

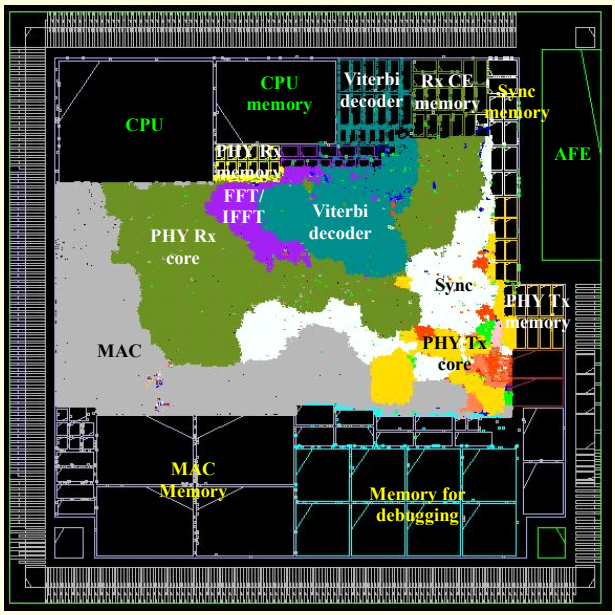
	Proposed FFT processor	Modified C.-P. Fan et al. [12]	Y.W. Lin et al. [4]	Modified Y. Jung et al. [19]	Z. Wang et al. [8]	S. Qiao et al. [9]
Architecture	Modified radix-2^4, radix-2^3	Radix-2^4, radix-2^3	Radix-2, radix-2^3	Radix-2, radix-4	Radix-4, radix-2	Radix-2, radix-8, radix-2^3
No. of complex registers	124 (56.4%)	124 (56.4%)	124 (56.4%)	220 (100%)	220 (100%)	148 (67.3%)
No. of nontrivial multipliers[1)]	4×0.6 (34.3%)	4 (57.1%)	2+4×0.62 (64%)	6 (85.7%)	3+4 (100%)	2+4×0.62 (64%)
No. of trivial multipliers[2)]	4×1.97+2×0.82 (52.9%)	4×3+6 (100%)	6 (33.3%)	6 (33.3%)	4×3+4 (88.9%)	4 (22.2%)
No. of complex adders	48 (100%)	48 (100%)	48 (100%)	28 (58.3%)	48 (100%)	42 (87.5%)
Word length	10 bits	-	10 bits	-	-	10 bits
Throughput rate (R: clock rate)	4R	4R	4R	4R	4R	4R

1) The nontrivial multiplier is the conventional complex variable multiplier [12], [19].

2) In Table 3, the number of trivial multipliers is counted as the number of the complex constant multipliers for the twiddle factor W_8^1 or W_8^3, which is realized by shifters and adders in the existing FFT processors [4], [11].

speed of the FFT processor is decreased as shown in Table 2. Implementation results indicate that the proposed FFT processor with a 10-bit internal word length can support a data processing rate of 1 Gsample/s with a power dissipation of 112 mW at 250 MHz. Note that the throughput rate of the MRMDF FFT processor in [4] is up to 1 Gsample/s, and it consumes 175 mW. The power consumption of the proposed FFT processor is approximately 36% lower than that of the MRMDF FFT processor. Figure 8 shows the output signal-to-noise ratio (SNR) for the fixed input SNR with various internal word lengths in the proposed FFT architecture. As the word length is equal to or greater than 10 bits, the output SNR is almost saturated, and accordingly, the quantization noise can be nearly ignored. Based on the simulation results, the proposed FFT processor is implemented with a 10-bit internal word length.

Table 3 compares the hardware complexity of the proposed FFT processor and the existing 128-point four-parallel data-path FFT architectures. Because the proposed FFT processor employs modified radix-2^4 and radix-2^3 FFT architectures, nontrivial multiplication operations are only needed at stage 4. In the proposed FFT architecture, four nontrivial complex multipliers at stage 4 are implemented with the CBMs presented in [17] with 60% of the hardware cost of conventional complex variable multipliers [12], [19]. In addition, the hardware complexities of CCM1s at stage 2 and CCM2 (or CCM3) at stage 6 are significantly reduced by about 34% and 18%, respectively, by employing the proposed SMU architectures as compared to those of conventional CCMs. Note that the trivial multiplication operations of the proposed FFT processor can be performed with approximately 53% of the hardware cost of the conventional radix-2^4 FFT processor in [12]. The proposed FFT processor reduces the hardware complexity of complex multipliers by about 31% as compared to the MRMDF FFT processor in [4]. Table 3 indicates that the proposed 128-point mixed-radix FFT



MAC: medium access control
CPU: central processing unit
Tx: transmitter
Rx: receiver
CE: channel estimator
PHY: physical layer
FFT: fast Fourier transform
Sync: synchronization block
AFE: analog front-end
IFFT: inverse FFT

Fig. 9. Floor plan of an MB-OFDM UWB SoC.

processor is a hardware-efficient structure and is therefore suitable for high-speed UWB applications.

Figure 9 shows the floor plan of an MB-OFDM UWB system-on-a-chip (SoC) including the proposed low-complexity 128-point mixed-radix FFT processor. The implemented MB-OFDM UWB SoC consists of several modules, namely, a medium access control (MAC), a PHY, an analog front-end (AFE), a central processing unit (CPU), and memory blocks. In our implementation, the 128-point FFT/IFFT block occupies about 5.1% of the silicon area of the PHY module.

V. Conclusion

In this paper, we have proposed a hardware-efficient 128-point mixed-radix DIF FFT processor with four data paths for MB-OFDM UWB systems. We have derived a mixed-radix FFT algorithm composed of modified radix-2^4 FFT and radix-2^3 FFT algorithms. By employing the mixed-radix FFT algorithm in the proposed FFT architecture, we have significantly reduced the number of both CCMs and CBMs. In addition, the hardware complexity of the proposed CCMs for trivial multiplications has been reduced by approximately 32% when compared to that of the existing CCM structures by adopting multiplication units using a substructure-sharing scheme. Implementation results have shown that the proposed mixed-radix FFT processor with 10-bit internal word length and four parallel data paths can support a data processing rate of up to 1.0 Gsample/s with a power dissipation of 112 mW at 250 MHz using 0.18 μm CMOS technology.

References

[1] A. Batra et al., "Design of a Multiband OFDM System for Realistic UWB Channel Environments," *IEEE Trans. Microw. Theory Tech.,* vol. 52, no. 9, Sept. 2004, pp. 2123-2138.

[2] K.M. Kang and S.S. Choi, "Initial Timing Acquisition for Binary Phase-Shift Keying Direct Sequence Ultra-wideband Transmission," *ETRI Journal*, vol. 30, no. 4, Aug. 2008, pp. 495-505.

[3] W. Abbott et al., *Multiband OFDM Physical Layer Specification, Version 1.2 (draft),* WiMedia Alliance, Feb. 2007.

[4] Y.W. Lin, H.Y. Liu, and C.Y. Lee, "A 1-GS/s FFT/IFFT Processor for UWB Applications," *IEEE J. Solid-State Circuits*, vol. 40, no. 8, Aug. 2005, pp. 1726-1735.

[5] S.I. Cho, K.M. Kang, and S.S. Choi, "Implementation of 128-Point Fast Fourier Transform Processor for UWB Systems," *Proc. IEEE IWCMC*, Aug. 2008, pp. 210-213.

[6] J.S. Lee et al. "A High-Speed, Low-Complexity Radix-2^4 FFT Processor for MB-OFDM UWB Systems," *Proc. IEEE ISCAS*, May 2006, pp. 4719-4722.

[7] T.S. Chakraborty and S. Chakrabarti, "A Reduced Area 1 GSPS FFT Design Using MRMDF Architecture for UWB Communication," *Proc. IEEE APCCAS*, Nov. 2008, pp. 1128-1131.

[8] Z. Wang et al., "A Novel FFT Processor for OFDM UWB Systems," *Proc. IEEE APCCAS*, Dec. 2006, pp. 374-377.

[9] S. Qiao et al., "An Area and Power Efficient FFT Processor for UWB Systems," *Proc. IEEE WICOM*, Sept. 2007, pp. 582-585.

[10] J. García, J.A. Michel, and A.M. Burón, "VLSI Configurable Delay Commutator for a Pipeline Split Radix FFT Architecture," *IEEE Trans. Signal Process*., vol. 47, no. 11, Nov. 1999, pp. 3098-3107.

[11] K. Maharatna, E. Grass, and U. Jagdhold, "A 64-Point Fourier Transform Chip for High-Speed Wireless LAN Application Using OFDM," *IEEE J. Solid-State Circuits*, vol. 39, no. 3, Mar. 2004, pp. 484-493.

[12] C.-P. Fan, M.-S. Lee, and G.-A. Su, "A Low Multiplier and Multiplication Costs 256-Point FFT Implementation with Simplified Radix-2^4 SDF Architecture," *Proc. IEEE APCCAS*, Dec. 2006, pp. 1935-1938.

[13] K.K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*, New York; John Wiley & Sons, 1999.

[14] G. Zhong et al., "An Energy-Efficient Reconfigurable Angle-Rotator Architecture," *Proc. IEEE ISCAS*, vol. 3, May 2004, pp. 661-664.

[15] C.H. Shin et al., "A Design and Performance of 4-Parallel MB-OFDM UWB Receiver," *IEICE Trans. Commun*., vol. E90-B, no. 3, Mar. 2007, pp. 672-675.

[16] S.W. Choi, K.M. Kang, and S.S. Choi, "A Two-Stage Radix-4 Viterbi Decoder for Multiband OFDM UWB Systems," *ETRI Journal*, vol. 30, no. 6, Dec. 2008, pp. 850-852.

[17] K.J. Cho et al., "Design of Low-Error Fixed-Width Modified Booth Multiplier," *IEEE Trans. VLSI Syst.*, vol. 12, no. 5, May 2004, pp. 522-531.

[18] S.M. Kim, J.G. Chung, and K.K. Parhi, "Low Error Fixed-Width CSD Multiplier with Efficient Sign Extension," *IEEE Trans. Circuits & Systems II*, vol. 50, no. 12, Dec. 2003, pp. 984-993.

[19] Y. Jung, H. Yoon, and J. Kim, "New Efficient FFT Algorithm and Pipeline Implementation Results for OFDM/DMT Applications," *IEEE Trans. Consumer Elect.*, vol. 49, no. 1, Feb. 2003, pp. 14-20.

Sang-In Cho received the BS and MS degrees in information and telecommunication engineering from Chonbuk National University, Korea, in 1997 and 1999, respectively. Since 1999, he has been with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea. His current research interests include VLSI digital signal processing and digital communications with applications to UWB transmission systems.

Kyu-Min Kang received the BS, MS, and PhD degrees in electronic and electrical engineering from Pohang University of Science and Technology (POSTECH), Gyeongbuk, Korea, in 1997, 1999, and 2003, respectively. Since 2003, he has been with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea. His current research interests include spectrum engineering, digital signal processing, and high-speed digital transmission systems.