Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A novel 3780-point FFT processor scheme for the time domain synchronous OFDM system*

Ji-nan LENG[1], Lei XIE[†‡1,2], Hui-fang CHEN[1,2], Kuang WANG[1,2]

([1]*Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China*)

([2]*Zhejiang Provincial Key Laboratory of Information Network Technology, Hangzhou 310027, China*)

[†]E-mail: xiel@zju.edu.cn

Received Mar. 26, 2011; Revision accepted Aug. 16, 2011; Crosschecked Nov. 4, 2011

Abstract: The 3780-point FFT is a main component of the time domain synchronous OFDM (TDS-OFDM) system and the key technology in the Chinese Digital Multimedia/TV Broadcasting-Terrestrial (DMB-T) national standard. Since 3780 is not a power of 2, the classical radix-2 or radix-4 FFT algorithm cannot be applied directly. Hence, the Winograd Fourier transform algorithm (WFTA) and the Good-Thomas prime factor algorithm (PFA) are used to implement the 3780-point FFT processor. However, the structure based on WFTA and PFA has a large computational complexity and requires many DSPs in hardware implementation. In this paper, a novel 3780-point FFT processor scheme is proposed, in which a 60×63 iterative WFTA architecture with different mapping methods is imported to replace the PFA architecture, and an optimized CoOrdinate Rotation DIgital Computer (CORDIC) module is used for the twiddle factor multiplications. Compared to the traditional scheme, our proposed 3780-point FFT processor scheme reduces the number of multiplications by 45% at the cost of 1% increase in the number of additions. All DSPs are replaced by the optimized CORDIC module and ROM. Simulation results show that the proposed 3780-point FFT processing scheme satisfies the requirement of the DMB-T standard, and is an efficient architecture for the TDS-OFDM system.

Key words: 3780, CoOrdinate Rotation DIgital Computer (CORDIC), Digital Multimedia/TV Broadcasting-Terrestrial (DMB-T), FFT, Time domain synchronous OFDM (TDS-OFDM), Winograd Fourier transform algorithm (WFTA)
doi:10.1631/jzus.C1100071 **Document code:** A **CLC number:** TN915.05

1 Introduction

The digital television (DTV) technology is now being applied worldwide. Three major DTV standards, the Digital Video Broadcasting-Terrestrial (DVB-T), Advanced Television System Committee (ATSC), and Integrated Services Digital Broadcasting-Terrestrial (ISDB-T), are fully developed in Europe, North America, and Japan, respectively. China, with over 400 million TV sets, representing almost 30% among the world's TV users, has developed its own Digital Terrestrial Television Broadcasting (DTTB) standard since 1994. After 12 years of effort, a Chinese DTTB standard, typically referred to as the Digital Terrestrial/Television Multimedia Broadcasting (DTMB) (SAC, 2006), was ratified in August 2006. This standard became a mandatory provision in August 2007.

The DTMB consists of two modulation modes. One of them, Advanced Digital Television Broadcasting-Terrestrial (ADTB-T), is a single carrier modulation (C=1), and the other, Digital Multimedia/TV Broadcasting-Terrestrial (DMB-T), is a multicarrier modulation (C=3780) (Song *et al.*, 2007). Different from the modulation method in the DVB-T system (Yang *et al.*, 2007), time domain synchronous OFDM (TDS-OFDM) is the key technology of the DMB-T system, and it uses 3780 sub-carriers in a

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 61071129) and the Science and Technology Department of Zhejiang Province, China (Nos. 2008C21088, 2011R10035, and 2011R09003-06)

bandwidth of 8 MHz to fit the multicarrier modulation mode. Moreover, instead of traditional power of 2 (2K, 4K, 8K), a 3780-point fast Fourier transform (FFT) processor is used in the TDS-OFDM system, different from other major standards (Liu *et al.*, 2008).

Since the 3780-point FFT processor is an important component of the DMB-T modulation system, much work has been reported on this processor (Yang Z *et al.*, 2002; Camarda *et al.*, 2009; Cheng and Su, 2010; Yang X *et al.*, 2010). A 3780-point inverse FFT (IFFT) processor was presented as a typical design (Yang *et al.*, 2002). First, it uses Winograd Fourier transform algorithm (WFTA) (Winograd, 1976; Cheng and Parhi, 2007) to compute the small-N discrete Fourier transform (DFT). Then the Good-Thomas prime factor algorithm (PFA) (Good, 1958) is used to form 60- and 63-point FFTs combined small-N DFTs (N=3, 4, 5, 7, 9). Finally, the Cooly-Tukey Fourier transform algorithm (Cooley and Tukey, 1965) is used to complete the 3780-point FFT. Compared to the efficient radix-4 split-radix algorithm, the computational complexity of the 3780-point IFFT processor is almost the same, but the arithmetic unit complexity is decreased. However, for the high-speed applications where the multiplications are complex and the ROM size for twiddle factors are large, the hardware cost needs to be further reduced.

A reconfigurable FFT architecture was proposed by Camarda *et al.* (2009) to compute the FFT in sizes of 2048 (2K), 4096 (4K), 8192 (8K), and 3780. This architecture uses a high pipelined structure that guarantees a high throughput, and allows one to rapidly switch from one configuration to another. However, while processing the 3780-point FFT, it still deals with the 60- and 63-point FFTs with the prime factor algorithm, and has to execute the radix mapping algorithm in the last two stages, which leads to the traditional use of the twiddle factors. Furthermore, when implementing the 3780-point FFT, this architecture uses the same data processing flow as the typical design. Thus, its computational complexity and hardware cost are not optimized efficiently.

Two other solutions have recently been brought forward by Yang *et al.* (2010) and Cheng and Su (2010). The former implementation uses a decomposition of 3780 as 27×7×5×4 and adopts the pipeline structure. The number of multiplications is 0.724 times that in the typical design, but the number of additions increases to 1.69 times. The computational complexity can be further optimized. The latter implementation uses shift rules and analyzes the finite word length effect. Its structure is achieved by the composition of 3×4×5×7×9. This implementation also reduces hardware resource consumption and limits the number of embedded multipliers to 24. However, the latency time of two frames lowers the system performance.

To further reduce the computational complexity and the hardware cost of the 3780-point FFT processor, a novel scheme is proposed in this paper. In the proposed scheme, an iterative method with small-N WFTA with different mapping methods (Winograd, 1976) is used to implement the 60- and 63-point FFTs. An improved CoOrdinate Rotation DIgital Computer (CORDIC) module (Volder, 1959; Abdullah *et al.*, 2009) is employed to perform twiddle factor multiplications at the final stage of the 3780-point FFT.

2 WFTA and CORDIC algorithms

2.1 Generalized WFTA algorithm

As an efficient method to compute DFT, the WFTA decomposes N as the multiplication of small prime factors (Winograd, 1976). For each small prime factor, there is small-N DFT computation. WFTA represents N-point DFT in matrix form as

$$\boldsymbol{X} = \boldsymbol{D}_N \boldsymbol{x}, \tag{1}$$

$$\boldsymbol{D}_N = \boldsymbol{S}_N \boldsymbol{C}_N \boldsymbol{T}_N, \tag{2}$$

where $\boldsymbol{x}$ is the input vector and $\boldsymbol{X}$ is the DFT of $\boldsymbol{x}$, $\boldsymbol{S}_N$ and $\boldsymbol{T}_N$ are the rectangular matrices with elements −1, 0, +1, and $\boldsymbol{C}_N$ is a diagonal matrix. A basic WFTA module for small-N DFT costs normal multiplications in obtaining $\boldsymbol{C}_N$. The basic WFTA module also costs additions and subtractions in obtaining $\boldsymbol{S}_N$ and $\boldsymbol{T}_N$.

2.2 Generalized CORDIC algorithm

The CORDIC algorithm embodies a unique method for the rapid computation of sine and cosine (Volder, 1959). There are three choices of linear, circular, and hyperbolic coordinate systems. The twiddle factor multiplications in FFT can be represented in vector form as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \tag{3}$$

where $[x'\ y']^{\mathrm{T}}$ is the objective vector. The objective rotation angle θ is the sum of rotation angles in $\{\alpha_i\}$:

$$\theta = \sum_{i=0}^{b-1} \sigma_i \cdot \alpha_i. \tag{4}$$

$$\alpha_i = \arctan 2^{-i}. \tag{5}$$

In Eq. (4), b steps rotations are needed to reach the objective rotation angle θ. The sign array, in which σ_i=1 or −1, is decided by the distance between the resulting rotation angle and the objective rotation angle at each stage. Combining Eqs. (3)–(5), we have

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \prod_{i=0}^{b-1} \cos\alpha_i \begin{bmatrix} 1 & \sigma_i 2^{-i} \\ -\sigma_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \tag{6}$$

$$\sigma_i = \mathrm{sign}\left(\theta - \sum_{r=0}^{i-1} \alpha_r\right), \tag{7}$$

$$z_{i+1} = z_i + \sigma_i 2^{-i}, \tag{8}$$

where $[x_i\ y_i]^{\mathrm{T}}$ and z_i are the results of the ith stage CORDIC operation. $[x_i\ y_i]^{\mathrm{T}}$ and z_i represent the ith stage vector and the remaining rotation angle, respectively. Finally, the result is adjusted by a scale factor, $\prod_{i=0}^{b-1} \cos\alpha_i$, which is defined as K_{c}. The sign array σ_i and the scale factor K_{c} are computed generally by accompanying modules in recent architectures (Adiono and Purba, 2009). The basic CORDIC operation unit in implementation (Fig. 1) has no multiplication unit but addition/subtraction and shifting units.

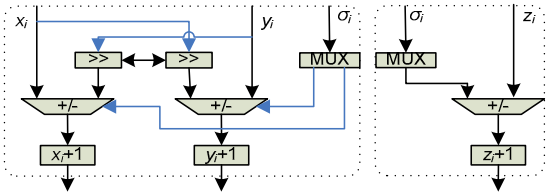

Fig. 1 Basic CORDIC operation unit

3 Novel 3780-point FFT processor scheme

Fig. 2 depicts the conventional 3780-point FFT processing flow. The 3-, 4-, 5-, 7-, and 9-point DFTs are calculated by the small-N WFTAs. Then, a 63-point PFA module and a 60-point PFA module are implemented to combine the small-N WFTAs. At last, a basic Cooly-Tukey algorithm is applied to combine the two modules with the twiddle factor multiplications.

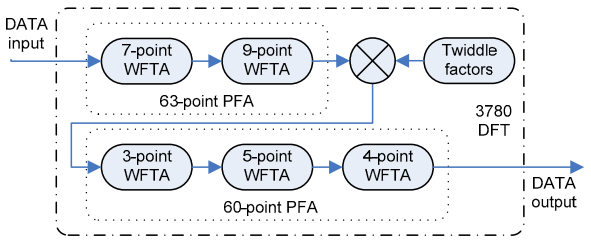

Fig. 2 Processing flow of the conventional 3780-point FFT

To optimize the 3780-point FFT architecture, we introduce a new scheme (Fig. 3) with the iterative WFTA and CORDIC method. In the proposed scheme, a 63-point iterative WFTA module and a 60-point iterative WFTA module replace the conventional PFA modules. Caches I and II are used to execute the matrix index mapping (Wu *et al.*, 1994). A CORDIC module is used to replace the twiddle factor multiplications structure. The input conjugate is connected to a frequency domain interleaver module. The output conjugate connects to the back-end baseband processing module with an index mapping cache.

The ***S***, ***C***, and ***T*** coefficients of the iterative WFTA modules are calculated according to the Kronecker product (Silverman, 1977). The iterative method of WFTA and its implementation (including index mapping) will be discussed in Section 3.1. The CORDIC method and its implementation will be discussed in Section 3.2.

3.1 Mapping and iteration of small-N WFTA and its module implementation

The small-N WFTA algorithm is an efficient way to solve multi-radix FFT. Silverman (1977) had fully analyzed the WFTA based on a matrix Kronecker product and presented some specific methods to execute the small-N algorithm in programming. Xing *et al.* (1996) designed a 1D-to-1D mapping to obtain an iterative WFTA structure, which is the same as the Cooly-Tukey algorithm. The mapping method in Xing *et al.* (1996) was complicated, however, and the reordering process should be carried out both before

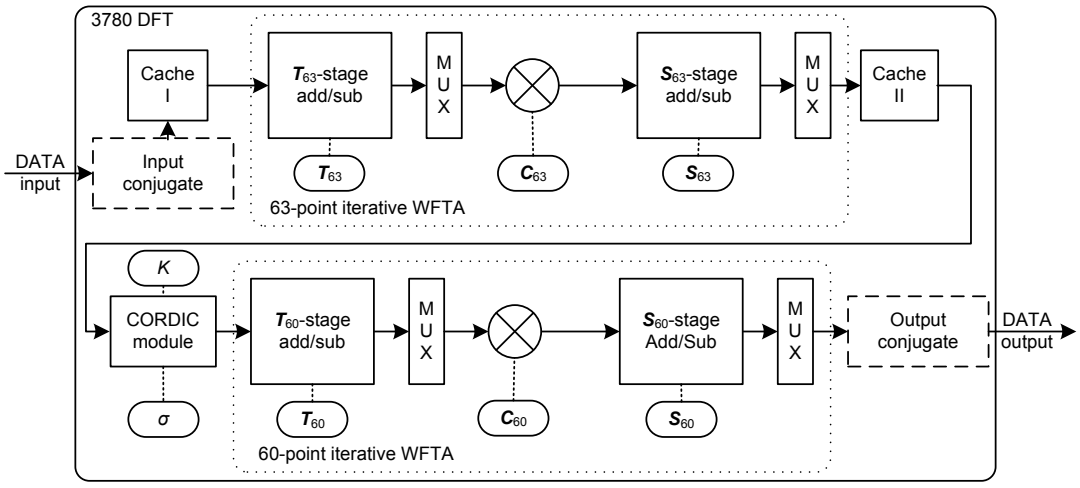

Fig. 3 Data flow of the proposed 3780-point FFT

and after FFT processing. Here we analyze the WFTA algorithm with a nesting method of 2D DFT and use simple mapping and Chinese remainder theorem (CRT) mapping to reorder the input and output data. According to the analysis results and using the mapping method, we implement the iterative WFTA modules for a 3780-point FFT processor. Based on the different mapping methods, an optimization on data reordering is planned for future work.

First, we map from 1D DFT to 2D DFT. The N-point DFT is defined as

$$X_k = \sum_{n=0}^{N-1} x_n W^{nk},\ k = 0,1,\cdots,N-1. \tag{9}$$

If $N=N_1\times N_2$, and N_1 and N_2 are mutually prime, a simple mapping of the subscripts of the input data can be defined as $n\leftrightarrow(n_1, n_2)$. To do this simple mapping, a middle variable group (T_1, T_2) is calculated first:

$$T_1 \cdot N_2 + T_2 \cdot N_1 \equiv 1 \,(\mathrm{mod}\, N). \tag{10}$$

Then the subscript group (n_1, n_2) is derived from T_1, T_2, and n:

$$\begin{aligned} &n_1 \equiv T_1 \cdot n \,(\mathrm{mod}\, N_1),\ \ n_2 \equiv T_2 \cdot n \,(\mathrm{mod}\, N_2), \\ &n_1 = 0,1,\cdots,N_1-1,\ \ n_2 = 0,1,\cdots,N_2-1. \end{aligned} \tag{11}$$

Using the CRT, and combining Eqs. (10) and (11), we deduce

$$n \equiv n_1 N_2 + n_2 N_1 \,(\mathrm{mod}\, N),\ \ n = 0,1,\cdots,N-1. \tag{12}$$

Hence, the subscript group (n_1, n_2) is a complete mapping of n, which is defined as simple mapping, or $n\leftrightarrow(n_1, n_2)$ mapping. The subscripts of the output data follow another CRT mapping, $k\leftrightarrow(k_1, k_2)$. Using the middle variable group (T_1, T_2) in Eq. (10), the subscript group (k_1, k_2) is formed by

$$\begin{aligned} &k_1 \equiv k \,(\mathrm{mod}\, N_1),\ \ k_2 \equiv k \,(\mathrm{mod}\, N_2), \\ &k_1 = 0,1,\cdots,N_1-1,\ \ k_2 = 0,1,\cdots,N_2-1. \end{aligned} \tag{13}$$

Using the CRT, the CRT mapping of $k\leftrightarrow(k_1, k_2)$ can be written as

$$k \equiv k_1 T_1 N_2 + k_2 T_2 N_1 \,(\mathrm{mod}\, N),\ k = 0,1,\cdots,N-1. \tag{14}$$

With these subscripts mapping rules, $X(k_1, k_2)$ and $x(n_1, n_2)$ can be substituted into $X_{k_1T_1N_2+k_2T_2N_1}$ and $x_{N_2n_1+N_1n_2}$, respectively. Considering $W^{N_1N_2}=1$, $W^{N_2}=W_{N_1}$, and $W^{N_1}=W_{N_2}$, Eq. (9) is adapted to an $N_1\times N_2$ 2D DFT as

$$\begin{aligned} &X(k_1,k_2) = \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x(n_1,n_2) W_{N_1}^{n_1k_1} W_{N_2}^{n_2k_2}, \\ &k_1 = 0,1,\cdots,N_1-1,\ \ k_2 = 0,1,\cdots,N_2-1. \end{aligned} \tag{15}$$

The subscripts of the 2D $N_1\times N_2$ DFT are generated using Eqs. (12) and (14). Then the 2D $N_1\times N_2$ DFT in Eq. (15) can be written in matrix form as follows:

$$\boldsymbol{X}=\boldsymbol{W}_{N_1}\boldsymbol{x}\boldsymbol{W}_{N_2},\ \boldsymbol{X}=(X_{k_1,k_2})_{N_1\times N_2},\ \boldsymbol{x}=(x_{n_1,n_2})_{N_1\times N_2},$$

$$\boldsymbol{W}_{N_i}=\begin{bmatrix}1 & 1 & 1 & \cdots & 1\\ 1 & W_{N_i} & W_{N_i}^2 & \cdots & W_{N_i}^{N_i-1}\\ \vdots & \vdots & \vdots & & \vdots\\ 1 & W_{N_i}^{N_i-1} & W_{N_i}^{2(N_i-1)} & \cdots & W_{N_i}^{(N_i-1)^2}\end{bmatrix},\ i=1,2. \tag{16}$$

We define the operation ⊗ as a matrix Kronecker product (Silverman, 1977). Due to the symmetry of $\boldsymbol{W}_{N_i}$, Eq. (16) can be rewritten as

$$\boldsymbol{X}=\boldsymbol{W}_{N_2}\otimes\boldsymbol{W}_{N_1}\boldsymbol{x}. \tag{17}$$

That is,

$$\begin{bmatrix}\boldsymbol{X}_0\\ \boldsymbol{X}_1\\ \vdots\\ \boldsymbol{X}_{N_2-1}\end{bmatrix}=\boldsymbol{W}_{N_2}\begin{bmatrix}\boldsymbol{W}_{N_1}\boldsymbol{x}_0\\ \boldsymbol{W}_{N_1}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{W}_{N_1}\boldsymbol{x}_{N_2-1}\end{bmatrix}. \tag{18}$$

Eqs. (17) and (18) can be described as

$$\boldsymbol{X}_{k_2}=\sum_{n_2=0}^{N_2-1}(W_{N_2}^{n_2k_2}\boldsymbol{W}_{N_1})\boldsymbol{x}_{n_2},\ k_2=0,1,\cdots,N_2-1. \tag{19}$$

This is an N_2-point DFT. Each multiplication with matrix $\boldsymbol{W}_{N_1}$ is an N_1-point DFT. Eqs. (17)–(19) are different forms of the same 2D DFT with the property of the Kronecker product (Silverman, 1977). Adapting $\boldsymbol{D}$ as the $\boldsymbol{W}$, we obtain $\boldsymbol{X}=(\boldsymbol{D}_{N_2}\otimes\boldsymbol{D}_{N_1})\boldsymbol{x}$ from Eq. (17). Considering the Kronecker product property $\boldsymbol{AB}\otimes\boldsymbol{CD}=(\boldsymbol{A}\otimes\boldsymbol{C})(\boldsymbol{B}\otimes\boldsymbol{D})$ and the generalized WFTA's basic Eqs. (1) and (2), the structure of the iterative WFTA module is

$$\boldsymbol{X}=(\boldsymbol{S}_{N_2}\otimes\boldsymbol{S}_{N_1})(\boldsymbol{C}_{N_2}\otimes\boldsymbol{C}_{N_1})(\boldsymbol{T}_{N_2}\otimes\boldsymbol{T}_{N_1})\boldsymbol{x}. \tag{20}$$

Therefore, when the greater N can be decomposed into two mutually prime factors, the smaller-N WFTA calculation can be iterated to achieve a greater N-point DFT with a data index mapping described in Eqs. (12) and (14). With the same approach to mapping and iterating, this calculation method can be extended to L mutually prime factors, where $\boldsymbol{X}$ and $\boldsymbol{x}$ are vectors with N elements, and N can be decomposed into L mutually prime factors $N_L, N_{L-1}, \ldots, N_1$.

The iterative WFTA module is shown in Fig. 4. The input data are reordered with simple mapping described in Eq. (12) and the output data are reordered with the CRT mapping as described in Eq. (14). In Fig. 4, the $\boldsymbol{T}$ stage operation is carried out by L (here L is the size of $\boldsymbol{x}$) groups of addition/subtraction. The input data (real part and imaginary part) stream fills in a row of registers step by step. As an incidence matrix (which has elements with values 0, −1, 1), the $\boldsymbol{T}$ coefficients are used to drive the adders/subtracters. The length of each $\boldsymbol{T}$ coefficient (i.e., the size of $\boldsymbol{C}$) is M. Therefore, each adder/subtracter works M steps. These intermediate results from the $\boldsymbol{T}$ stage operation flow to the $\boldsymbol{C}$ stage multiplication operation. Since $\boldsymbol{C}$ is a diagonal matrix including only pure number elements (i.e., real number or imaginary number), the multiplication operations need only real multipliers and real/imaginary switchers driven by $\boldsymbol{C}$ coefficients.

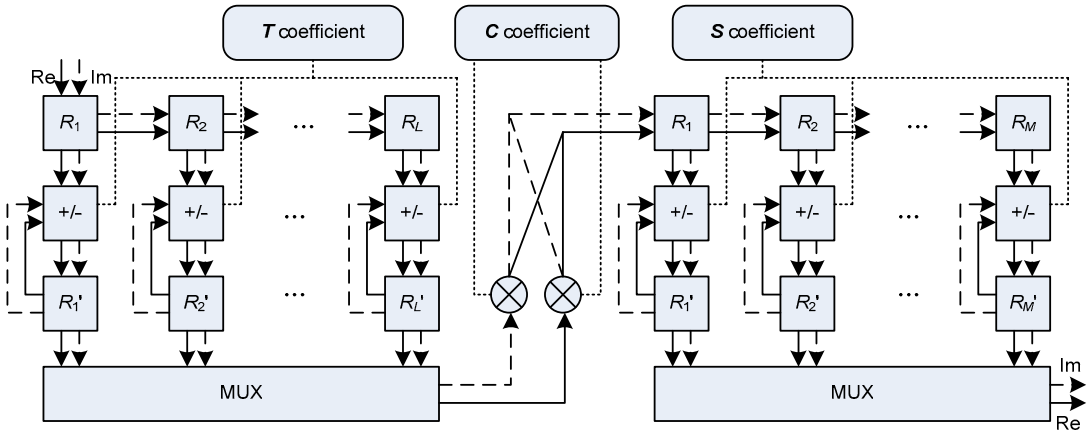

Fig. 4 The iterative Winograd Fourier transform algorithm (WFTA) module

The $\boldsymbol{S}$ stage operation, in which there are M groups of additions/subtractions and every adder/subtracter works L steps, is almost the same as the $\boldsymbol{T}$ stage operation.

For example, for the 63-point iterative WFTA module, the input data and the output data are reordered. With Eqs. (10)–(15), letting N_1=9, N_2=7, and N=63, the intermediate variable group (T_1, T_2)=(4, 4), the input simple mapping is $n\equiv 7n_1+9n_2 \pmod{63}$, n_1=0, 1, …, 8, n_2=0, 1, …, 6, and the output CRT mapping is $k\equiv 28k_1+36k_2 \pmod{63}$, k_1=0, 1, …, 8, k_2=0, 1, …, 6. Then, following the small-N WFTA matrices of 9- and 7-point, incidence matrices $\boldsymbol{S}_9$, $\boldsymbol{T}_9$, $\boldsymbol{S}_7$, $\boldsymbol{T}_7$, and diagonal matrices $\boldsymbol{C}_9$, $\boldsymbol{C}_7$ can be obtained.

With Eq. (20), the matrices $\boldsymbol{S}_{63}$, $\boldsymbol{T}_{63}$, and $\boldsymbol{C}_{63}$ of the 63-point WFTA module can be generated by $\boldsymbol{S}_7\otimes\boldsymbol{S}_9$, $\boldsymbol{T}_7\otimes\boldsymbol{T}_9$, and $\boldsymbol{C}_7\otimes\boldsymbol{C}_9$, respectively. Table 1 lists the sizes of the 9-, 7-, and 63-point WFTA matrices. Here the incidence matrix $\boldsymbol{T}_{63}$ is taken as an example to explain how to obtain $\boldsymbol{T}_{63}$ with $\boldsymbol{T}_7$ and $\boldsymbol{T}_9$. $\boldsymbol{T}_{63}$ can be deduced from $\boldsymbol{T}_7\otimes\boldsymbol{T}_9$. If

$$\boldsymbol{T}_7=\begin{bmatrix}1&1&1&1&0&0&0\\0&1&1&1&1&1&1\\0&1&0&-1&-1&0&1\\0&0&-1&1&1&-1&0\\0&-1&1&0&0&1&-1\\0&1&1&-1&1&-1&-1\\0&1&0&1&-1&0&-1\\0&0&-1&-1&1&1&0\\0&-1&1&0&0&-1&1\end{bmatrix},$$

then

$$\boldsymbol{T}_{63}=\begin{bmatrix}\boldsymbol{T}_9&\boldsymbol{T}_9&\boldsymbol{T}_9&\boldsymbol{T}_9&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{T}_9&\boldsymbol{T}_9&\boldsymbol{T}_9&\boldsymbol{T}_9&\boldsymbol{T}_9&\boldsymbol{T}_9\\\boldsymbol{0}&\boldsymbol{T}_9&\boldsymbol{0}&-\boldsymbol{T}_9&-\boldsymbol{T}_9&\boldsymbol{0}&\boldsymbol{T}_9\\\boldsymbol{0}&\boldsymbol{0}&-\boldsymbol{T}_9&\boldsymbol{T}_9&\boldsymbol{T}_9&-\boldsymbol{T}_9&\boldsymbol{0}\\\boldsymbol{0}&-\boldsymbol{T}_9&\boldsymbol{T}_9&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{T}_9&-\boldsymbol{T}_9\\\boldsymbol{0}&\boldsymbol{T}_9&\boldsymbol{T}_9&-\boldsymbol{T}_9&\boldsymbol{T}_9&-\boldsymbol{T}_9&-\boldsymbol{T}_9\\\boldsymbol{0}&\boldsymbol{T}_9&\boldsymbol{0}&\boldsymbol{T}_9&-\boldsymbol{T}_9&\boldsymbol{0}&-\boldsymbol{T}_9\\\boldsymbol{0}&\boldsymbol{0}&-\boldsymbol{T}_9&-\boldsymbol{T}_9&\boldsymbol{T}_9&\boldsymbol{T}_9&\boldsymbol{0}\\\boldsymbol{0}&-\boldsymbol{T}_9&\boldsymbol{T}_9&\boldsymbol{0}&\boldsymbol{0}&-\boldsymbol{T}_9&\boldsymbol{T}_9\end{bmatrix}.$$

The rows of $\boldsymbol{T}_{63}$ are the coefficients used in the T stage operation of the 63-point iterative WFTA module (Fig. 4). With the same method, the coefficients of $\boldsymbol{S}_{63}$ and $\boldsymbol{C}_{63}$ can be generated. All of these coefficients are stored in ROMs. The first input data stream from a 2D 9×7 DFT described in Eq. (15) is $\boldsymbol{x}_{63}=[x_{0,0}, x_{1,0}, \ldots, x_{8,0}, x_{0,1}, x_{1,1}, \ldots, x_{8,1}, \ldots, x_{0,6}, x_{1,6}, \ldots, x_{8,6}]^{\mathrm{T}}$ and the final output data stream from a 2D matrix is $\boldsymbol{X}_{63}=[X_{0,0}, X_{1,0}, \ldots, X_{8,0}, X_{0,1}, X_{1,1}, \ldots, X_{8,1}, \ldots, X_{0,6}, X_{1,6}, \ldots, X_{8,6}]^{\mathrm{T}}$. The whole 63-point WFTA module is the same as that shown in Fig. 4.

Table 1 The sizes of the 9-, 7-, and 63-point WFTA matrices

Matrix	Size	Matrix	Size	Matrix	Size
$\boldsymbol{S}_9$	9×11	$\boldsymbol{S}_7$	7×9	$\boldsymbol{S}_{63}$	63×99
$\boldsymbol{C}_9$	11×11	$\boldsymbol{C}_7$	9×9	$\boldsymbol{C}_{63}$	99×99
$\boldsymbol{T}_9$	11×9	$\boldsymbol{T}_7$	9×7	$\boldsymbol{T}_{63}$	99×63

3.2 CORDIC module for twiddle factor multiplication

The final stage of the 3780-point FFT processing, the combination of the 63- and 60-point DFTs, is a Cooly-Tukey algorithm. That is,

$$X_{k_1,k_2}=\sum_{n_1=0}^{59}W_{N_1}^{n_1k_1}\left[W^{n_1k_2}\left(\sum_{n_2=0}^{62}x_{n_1,n_2}W_{N_2}^{n_2k_2}\right)\right],\quad n_1,k_1=1,2,\cdots,59,\ \ n_2,k_2=1,2,\cdots,62. \tag{21}$$

The twiddle factor multiplication module contains 3780 complex multiplications, which consume many DSP resources in hardware implementation. A CORDIC module can be used to replace the complex multipliers. The conventional method (Lakshmi and Dhar, 2008) uses feedback architecture to calculate z_i flow. Abdullah *et al.* (2009) proposed a novel method to calculate the micro-rotation direction vector for a radix-2 N-point FFT. This method is used in our 3780-point FFT processor scheme. We calculate the sign array (direction vector) σ_i from the twiddle factors in Eq. (21) and pre-store them in ROM. As the twiddle factors $W_{N_2}^{n_2k_2}$ occur repeatedly, the pre-stored sign array can be reduced to half in size. To realize the design precision, an 18-stage CORDIC module is proposed (Fig. 5). Since the number of stages is fixed, the scale factor K_c is also fixed and stored in the same ROM.

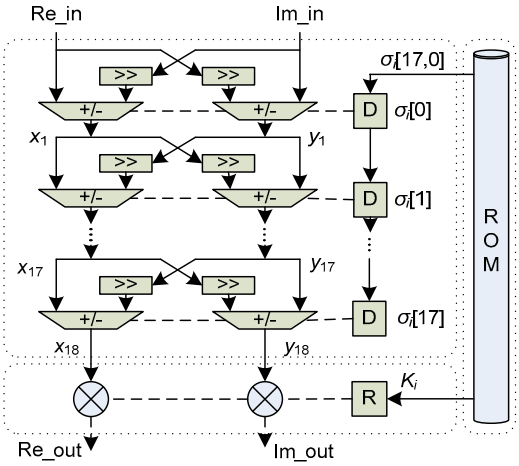

Fig. 5 The CORDIC module for twiddle factor multiplication

4 Computational complexity analysis

The 3780-point FFT contains 60- and 63-point FFT modules, whose basic elements are small-N WFTA units. Define M_i, A_i (i=1, 2) as the multiplications and additions of the N_i-point FFT. The N-point (N=N_1×N_2) FFT in Yang *et al.* (2002) and Camarda *et al.* (2009) is calculated using the Good-Thomas PFA. According to the N_1×N_2 2D DFT described in Eq. (15) and (N_1, N_2)=1, an intermediate value $Y(k_1, n_2)$ is

$$Y(k_1,n_2)=\sum_{n_1=0}^{N_1-1}x(n_1,n_2)W_{N_1}^{n_1k_1},\quad k_1=0,1,\cdots,N_1-1,\ \ n_2=0,1,\cdots,N_2-1. \tag{22}$$

Hence,

$$X(k_1,k_2)=\sum_{n_2=0}^{N_2-1}Y(k_1,n_2)W_{N_2}^{n_2k_2},\quad k_1=0,1,\cdots,N_1-1,\ \ k_2=0,1,\cdots,N_2-1. \tag{23}$$

Eq. (22) represents N_2 groups of N_1-point DFT which contain N_2M_1 multiplications and N_2A_1 additions. Eq. (23) represents N_1 groups of N_2-point DFT which contain N_1M_2 multiplications and N_1A_2 additions. Let M and A be the number of multiplications and additions of the N-point FFT, respectively. Then,

$$\begin{cases}M=N_1M_2+N_2M_1,\\ A=N_1A_2+N_2A_1.\end{cases} \tag{24}$$

In the proposed architecture, the iterative WFTA module is performed based on Eq. (20); that is, $\boldsymbol{X}=(\boldsymbol{S}_{N_2}\otimes\boldsymbol{S}_{N_1})(\boldsymbol{C}_{N_2}\otimes\boldsymbol{C}_{N_1})(\boldsymbol{T}_{N_2}\otimes\boldsymbol{T}_{N_1})\boldsymbol{x}$. As $\boldsymbol{C}_{N_1}$ and $\boldsymbol{C}_{N_2}$ are diagonal matrices, the combined matrix $\boldsymbol{C}_N$ is also a diagonal matrix, and its size is the product of the other two matrices. Hence, the number of multiplications is the size of $\boldsymbol{C}$, i.e., M=M_1M_2.

To calculate the number of the combined additions, A, Eq. (20) is analyzed with the vector formation described in Eq. (19). Eq. (19) is also an N_2-point DFT, whose twiddle factor is $W_{N_2}^{n_2k_2}\boldsymbol{W}_{N_1}$. Since Eq. (19) contains N_2 groups of N_1-point DFT (in vector form), it has M_2 vector multiplications and A_2 vector additions. As vector multiplication $W_{N_2}^{n_2k_2}\boldsymbol{W}_{N_1}\boldsymbol{x}_{n_2}$ is an N_1-point DFT, each vector multiplication has A_1 additions. As vectors $\boldsymbol{X}_{k_2}$ and $\boldsymbol{x}_{k_2}$ both have N_1 elements, each vector addition has N_1 additions. The total number of additions in the iterative WFTA module, A, is A_2N_1+M_2A_1. Hence, the combined computational complexity of the iterative WFTA module is

$$\begin{cases}M=M_1M_2,\\ A=A_2N_1+M_2A_1.\end{cases} \tag{25}$$

To combine the 60- and 63-point FFTs into the final 3780-point FFT, we use a Cooly-Tukey algorithm as Eq. (21), whose computational complexity (Cooley and Tukey, 1965) is fully introduced in the textbook. Then the computational complexity of the 3780-point FFT is

$$\begin{cases}M_{3780}=N_{60}M_{63}+N_{63}M_{60}+N,\\ A=N_{60}A_{63}+N_{63}A_{60}.\end{cases} \tag{26}$$

With the PFA method and based on Eq. (24), the computational complexity of 63-point FFT can be calculated from 7- and 9-point small-N WFTA, M_{63}=158 and A_{63}=632. The computational complexity of 60-point FFT is calculated from 3-, 4-, 5-point small-N WFTAs, M_{60}=192 and A_{60}=444. Then based on Eq. (26), M_{3780}=25 356 and A_{3780}=65 892. Since the multiplicands in the WFTA module are all pure number elements (i.e., real number or imaginary number), the number of real multiplications is double

that of the complex multiplications, and the number of real additions is also double that of the complex additions. The PFA architecture adopted in Yang *et al.* (2002) consumes 50 712 real multiplications and 140 184 real additions. The 5% difference between real additions is caused by the different small-N WFTA matrices chosen.

In the proposed 3780-point FFT architecture, the iterative WFTA method is used. According to Eq. (24), the computational complexity of the 63-point FFT can be calculated from 7- and 9-point small-N WFTA, M_{63}=99 and A_{63}=704. The computational complexity of the 60-pint FFT can be calculated from 3-, 4-, 5-point small-N WFTAs, M_{60}=72 and A_{60}=396. Then based on Eq. (26), M_{3780}=14 256 and A_{3780}=67 188. The numbers of real multiplications and additions are 28 512 and 134 376, respectively. The computational complexities (numbers of complex additions and complex multiplications) of the PFA method and the proposed iterative WFTA method are listed in Table 2. Using the proposed iterative WFTA method, 45% of the multiplication operations are reduced at the cost of 1% increase in the number of additions.

Table 2 Comparison of the computational complexity between the prime factor algorithm (PFA) and the proposed iterative Winograd Fourier transform algorithm (WFTA)

Point of FFT	Number of complex multiplications		Number of complex additions	
	PFA	Iterative	PFA	Iterative
3	3	6	3	6
4	4	8	4	8
5	6	17	6	17
7	9	36	9	36
9	11	44	11	44
63	158	632	99	704
60	192	444	72	396
3780	25 356	65 892	14 256	67 188

For the last stage, the twiddle factor operation, there are 3780 complex multiplications. Adopting the proposed CORDIC module, DSPs can be replaced by an 18-stage shifting and addition, as described in Section 3.2. Furthermore, the ROM for twiddle factors can also be reduced to half in size because of the utilization of the σ_i array.

5 Simulation results and analysis

5.1 System SNR

Since the fixed-point truncation and rounding in the WFTA modules will result in error and precision decrease (Patterson and Mcclellan, 1978), the proposed 3780-point FFT architecture is first simulated on Matlab in fixed- and floating-point calculation modes, respectively. The difference between the results of two calculation modes is defined as the system noise. The signal-to-noise ratio (SNR) of the system is defined as the energy ratio of the floating-point calculation to the system noise. The input data stream is set to be a Gaussian signal with $N(\mu, \sigma^2)$, where mean value μ is zero and the variance σ^2 is determined by the word length b. We simulate the proposed architecture with different word lengths and adjust the stages of the CORDIC module. The results show that the system SNR increases with the increase of the word length of output data. In the proposed architecture, the input data are 5-bit inphase components and 5-bit quadrature components, while the output data are 16-bit inphase components and 16-bit quadrature components. The system SNR is 67.9 dB, which satisfies the requirement of the TDS-OFDM system.

5.2 Latency analysis

In the proposed 3780-point FFT architecture, the latency is introduced by two iterative WFTA modules and their in/out caches. To generate a complete OFDM symbol, the 63-point iterative module described in Section 3.1 requires 3942 clock cycles, including the registers at $\boldsymbol{S}$ and $\boldsymbol{T}$ stages and input reordering RAM. Similarly, the 60-point iterative module requires 3912 clock cycles. The CORDIC module for the twiddle factor multiplication described in Section 3.2 requires 19 clock cycles. For the TDS-OFDM system, the symbol rate is 7.56 Msps (SAC, 2006), and the latency of the total 7873 clock cycles is approximately 1042 μs, which is close to the latency of a typical design (Yang *et al.*, 2002), and also satisfies the requirement of the DTMB standard.

5.3 Computational complexity and resource usage

Table 3 provides the comparison of computational complexity and resource usage of the proposed 3780-point FFT architecture and four existing designs.

Table 3 Comparison of the computational complexity and resource usage

3780-point FFT scheme	Architecture	RealMul	RealAdd	Clock cycles	Logic cells	Memory bits/buffer	DSP blocks
Camarda *et al.*, 2009	7×5×4×3×3×3			30289	4077	5112568	44
Cheng and Su, 2010	7×9×5×3×4				6280	704752	24
Yang X *et al.*, 2010	27×7×5×4	33000	236446				
Yang Z *et al.*, 2002	7×9×4×5×3	50712	140184	8064		141612	
Proposed architecture	63×60	28512	134376	7873	3343	136080	0

Blank means that the property is not mentioned in the literature

The computational complexities of iterative WFTA and PFA have been analyzed in Section 4. As shown in Table 3, the number of complex multiplications in our proposed architecture is 56% of that in the typical design and 86% of that in Yang *et al.* (2010)'s design. The number of complex additions in our proposed architecture is almost the same as that in the typical design (In Section 4, we have calculated the number of complex additions in the typical design, which is 131784, smaller than the original number 140184. This is caused by the different small-N WFTA matrices chosen.) and 57% of that in Yang *et al.* (2010)'s design. Most of the buffers in the proposed architecture are consumed by the reordering RAM of the two iterative WFTA modules. As the proposed processor is placed between a frequency domain interleaver and a frame combiner, the output reordering RAM is integrated into the following module. According to the Chinese DTMB standard, the forward error correction coding data flow needs to be transferred into nQAM (n=64, 32, 16, 4) symbol flow. The input inphase components and quadrature components can be scaled by 2 bits, and 5-bit word length is enough for them. Thus, the input RAM word length is set to 5 bits. The buffer requirement of the 63-point iterative module is 378000 bits. As the word length of the intermediate results before the twiddle factors multiplier is 13 bits, the buffer size required by the 60-point iterative module is 98280 bits. The total buffer requirement is 136080 bits, less than that of the typical design, because three intermediate caches between small-N WFTA modules are removed.

The proposed 3780-point FFT architecture is implemented in Quartus II with Verilog HDL, synthesized and simulated on a Stratix II EP2S90 chip. The proposed architecture can support a maximum clocking frequency of 98.14 MHz, using 3343 logic combinational ALUTs and no DSP blocks. The logic resource consumption in the proposed design is 53% of that in Cheng and Su (2010)'s design and 81% of that in Camarda *et al.* (2009)'s design. We also simulate the proposed architecture with and without the CORDIC method. The results show that the CORDIC module used in the twiddle factor multiplication saves 40 9-bit DSP blocks.

From all the above, we observe that the computational complexity and hardware resource of our proposed 3780-point FFT architecture are optimized from the existing designs.

6 Conclusions

We propose a novel 3780-point FFT processor scheme. To reduce the computational complexity and hardware resource, an iterative WFTA architecture with different mapping methods and an optimized CORDIC module are implemented to replace the conventional PFA architecture and twiddle factor multiplications. Compared with the typical design, the proposed architecture reduces the number of multiplications by 45% at the cost of 1% increase in the number of additions. Simulation and analysis results show that the performance of the proposed 3780-point FFT processor is better than those of the previous designs in terms of latency and recourse usage. It is an efficient architecture for the TDS-OFDM system in the DTMB standard.

The iterative WFTA architecture introduces a complicated reordering procedure, which consumes some ROM resources. Our future work is to optimize the reordering procedure with the advantages of the different mapping rules.

References

Abdullah, S., Haewoon, N., McDermot, M., Abraham, J., 2009. A High Throughput FFT Processor with No Multipliers. IEEE Int. Conf. on Computer Design, p.485-490. [doi:10.1109/ICCD.2009.5413113]

Adiono, T., Purba, R.S., 2009. Scalable Pipelined CORDIC Architecture Design and Implementation in FPGA. Int. Conf. on Electrical Engineering and Informatics, p.646-649. [doi:10.1109/ICEEI.2009.5254736]

Camarda, F., Prevotet, J.C., Nouvel, F., 2009. Implementation of a Reconfigurable Fast Fourier Transform Application to Digital Terrestrial Television Broadcasting. 19th Int. Conf. on Field Programmable Logic and Applications, p.353-358. [doi:10.1109/FPL.2009.5272266]

Cheng, C., Parhi, K.K., 2007. Low-cost fast VLSI algorithm for discrete Fourier transform. *IEEE Trans. Circ. Syst. I*, **54**(4):791-806. [doi:10.1109/TCSI.2006.888772]

Cheng, G., Su, K., 2010. A 3780-point FFT Algorithm and Its FPGA Implementation. Int. Symp. on Next Generation Electronics, p.231-233. [doi:10.1109/ISNE.2010.5669154]

Cooley, J.W., Tukey, J.W., 1965. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.*, **19**(90):297-301. [doi:10.1090/S0025-5718-1965-0178586-1]

Good, I.J., 1958. The interaction algorithm and practical Fourier series. *J. R. Stat. Soc.*, **20**(2):361-372.

Lakshmi, B., Dhar, A.S., 2008. Low Latency VLSI Architecture for the Radix-4 CORDIC Algorithm. IEEE Region 10 and 3rd Int. Conf. on Industrial and Information Systems, p.1-5. [doi:10.1109/ICIINFS.2008.4798377]

Liu, M., Crussiere, M., Helard, J.F., Pasquero, O.P., 2008. Analysis and Performance Comparison of DVB-T and DTMB Systems for Terrestrial Digital TV. 11th IEEE Int. Conf. on Communication Systems, p.1399-1404. [doi:10.1109/ICCS.2008.4737413]

Patterson, R.W., Mcclellan, J.H., 1978. Fixed-point error analysis of Winograd Fourier transform algorithms. *IEEE Trans. Acoust. Speech Signal Process.*, **26**(5):447-455. [doi:10.1109/TASSP.1978.1163134]

SAC (Standardization Administration of the People's Republic of China), 2006. Framing Structure, Channel Coding and Modulation for Digital Television Terrestrial Broadcasting System. Standards Press of China, Beijing, China (in Chinese).

Silverman, H., 1977. An introduction to programming the Winograd Fourier transform algorithm (WFTA). *IEEE Trans. Acoust. Speech Signal Process.*, **25**(2):152-165. [doi:10.1109/TASSP.1977.1162924]

Song, J., Yang, Z., Yang, L., Gong, K., Pan, C., Wang, J., Wu, Y., 2007. Technical review on Chinese Digital Terrestrial Television Broadcasting Standard and measurements on some working modes. *IEEE Trans. Broadcast.*, **53**(1):1-7. [doi:10.1109/TBC.2007.891835]

Volder, J.E., 1959. The CORDIC trigonometric computing technique. *IEEE Trans. Electron. Comput.*, **8**(3):330-334. [doi:10.1109/TEC.1959.5222693]

Winograd, S., 1976. On computing the discrete Fourier transform. *PNAS*, **73**(4):1005-1006. [doi:10.1073/pnas.73.4.1005]

Wu, C., Wang, R., Yen, C., 1994. Adder-based SIMD-systolic architectures and VLSI chip for computing the Winograd small FFT algorithms. *Int. J. Electron.*, **76**(6):1135-1149. [doi:10.1080/00207219408926023]

Xing, Q., Zhu, Q., Sun, S., 1996. Iterative Structure of Winograd FFT Algorithm. Conf. on Precision Electromagnetic Measurements, p.57-58. [doi:10.1109/CPEM.1996.546557]

Yang, J., Wang, K., Zou, Z., 2007. DVB-S2 inner receiver design for broadcasting mode. *J. Zhejiang Univ.-Sci. A*, **8**(1):28-35. [doi:10.1631/jzus.2007.A0028]

Yang, X., Du, W., Yang, Z., Lv, R., Wang, X., 2010. Hardware Design and Implementation of 3780 Points FFT Based on FPGA in DTTB. 4th Int. Conf. on Application of Information and Communication Technologies, p.1-4. [doi:10.1109/ICAICT.2010.5611806]

Yang, Z., Hu, Y., Pan, C., Yang, L., 2002. Design of a 3780-point IFFT processor for TDS-OFDM. *IEEE Trans. Broadcast.*, **48**(1):57-61. [doi:10.1109/11.992857]