Time-recursive Parallel IIR Filter Structures for the IFFT/FFT in DMT Transceiver Systems

K. M. M. Prabhu, A. Madhu Sudana Rao, and P. C. W. Sommen

Abstract—**Discrete multitone (DMT) is a line code which has been adopted as a standard for the asymmetric digital subscriber line (ADSL) systems by the American National Standards Institute (ANSI). The DMT uses the IFFT and FFT for modulation and demodulation, respectively. The large size of the IFFT/FFT results in a heavy computational load on the programmable DSP processors. It also proves that a cost-efficient hardware implementation of radix-2 IFFT/FFT butterfly structure is not feasible, since it requires a large number of multipliers and adders. For cost-efficient hardware implementation, we have proposed two schemes to develop the IFFT/FFT architecture based on the time-recursive approach. In fact, these methods require approximately a maximum of only 12.5% of the multipliers and 15.6% of the adders when compared to the conventional butterfly type of implementation.**

Index Terms—**Discrete multitone (DMT), FFT/ IFFT, IIR structures, Time-recursive approach**

I. Introduction

In this section, we consider the discrete multitone (DMT) [1], [2] which is a multicarrier transmission technique that uses the fast Fourier transform (FFT) and the inverse FFT to allocate the transmitted bits among many narrowband quadrature amplitude modulated (QAM) tones depending on the transport capacity of each tone.

A. DMT Principles and its Computational Complexity

For channels that exhibit high signal attenuation at frequencies within the passband, a valid alternative to carrierless-amplitude-phase (CAP) modulation [3] and quadrature amplitude modulation (QAM) [4] is represented by a modulation technique known as orthogonal frequency division multiplexing (OFDM), or multicarrier modulation (MCM). As the term implies, multicarrier modulation is obtained in principle by modulating several carriers in parallel using blocks of symbols, therefore using a symbol period that is typically much longer than the symbol period of a single carrier system transmitting at the same bit rate. The resulting narrowband signals around the frequencies of the carrier are then added and transmitted over the channel.

The narrowband signals are usually referred to as subchannel signals. The advantage of OFDM with respect to single carrier systems is represented by the lower complexity required for equalization, which under certain conditions can be performed by a filter with a single coefficient per subchannel. A long symbol period also yields a greater immunity of an OFDM system to impulse noise. Another important aspect is represented by the efficient implementation of modulator and demodulator, obtained by sophisticated signal processing algorithms.

Multicarrier modulation has recently found use in many applications in both baseband and passband transmissions. Some of the most well-knowm applications are digital subscriber line (DSL) over wired media using discrete multitone (DMT) [1], [2] and digital audio broadcasting (DAB) or digital video broadcasting (DVB) for wireless communications using OFDM [5].

The DMT, therefore, is a multicarrier modulation scheme for DSL applications with the discrete Fourier transform (DFT) bases as the carriers. DMT [1], [2] has many advantages over the traditional single-carrier modulation system [3], [4] and it has been adopted as the ANSI standard for asymmetric digital subscriber line (ADSL). Since it takes channel signal-to-noise ratios (SNRs) into account, transmission of high speed data over bandlimited channels (sub-channels) can be achieved by allocating more bits to those sub-channels with higher SNRs and fewer or no bits to those sub-channels with lower SNRs. The DFT is a heavily used and well–understood operation in digital signal processing and a variety of structures exist for its very efficient implementation.

The direct computation of the DFT requires $4N^2$ real multiplications and $(4N-2)N$ real additions. Therefore, the number of multiplications and additions is proportional to N^2. Hence, it is of practical interest to develop more efficient or fast algorithms for computing the DFT. The conventional approach to recursive architecture began with the advent of Goertzel's algorithm [6]. This constitutes a set of algorithms proposed to compute the DFT of a given N-point sequence in a recursive manner. The computation of the N-point DFT results in $4N^2$ real multiplications and $4N^2$ real additions. Therefore, this algorithm, in comparison to the direct DFT computation, requires the same number of real multiplications, but $2N$ more real additions. Hence, it is slightly more inefficient than the direct approach. But the advantage of this algorithm is that the N-complex coefficients required to compute the output do not have to be computed or stored in advance, but they are computed recursively as needed. This algorithm can be made more efficient by avoiding the complex multiplications as given in [6]. Then, the number of real multiplications and additions for the scheme works out to be approximately N^2 and $2N^2$, respectively, which are still proportional to N^2. Goertzel's algorithm is attractive

Manuscript received September 1, 2002; revised May 1, 2003.

K. M. M. Prabhu is with the Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai- 600 036, India (e-mail: prabhu_kmm@hotmail.com).

A. Madhu Sudana Rao is with the Sasken Communication Technologies Ltd., Bangalore- 560 075, India (e-mail: amadhu@sasken.com).

P. C. W. Sommen is with the Department of Electrical Engineering, P. O. Box 513, Bldg. Eh 3. 34, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands (e-mail: p.c.w.sommen@tue.nl).

Publisher Item Identifier S 1682-0053(03)0172

1682-0053/03$10 © 2003 JD

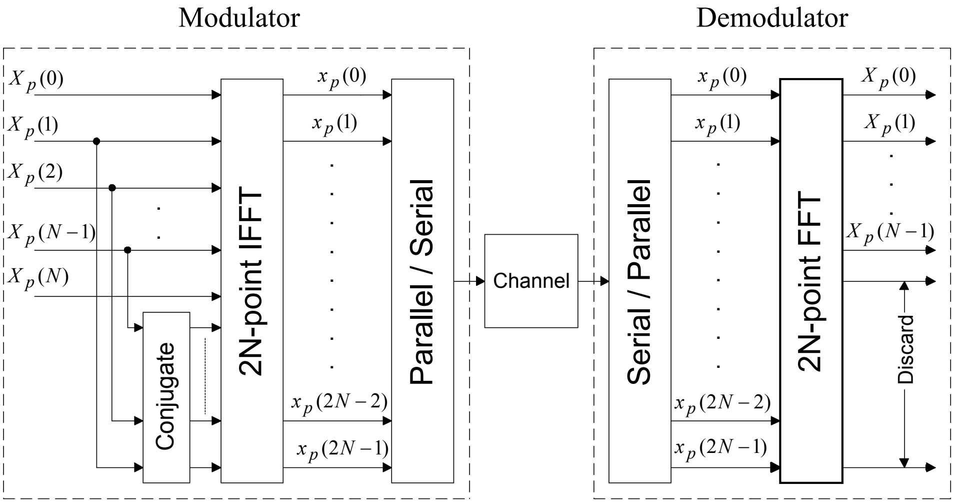

Fig. 1. The IFFT/FFT block diagram of the DMT transceiver system.

only in applications requiring the computation of a few samples of the DFT. One such example is the dual-tone multifrequency (DTMF) signal detection. A variety of structures exist for the very efficient implementation of DFT in $N\log_2 N$ operations and are known as FFT algorithms [6].

The DMT uses the inverse FFT (IFFT) for modulation and the FFT for demodulation. The DMT divides the channel into many sub-channels. The frequency response of the sub-channels can be approximated to have a flat response if they are large in number. The flatness is required in order to reduce the complexity of the equalizer. However, this increases the IFFT/FFT length due to an increase in the number of sub-channels. The large size of the IFFT/FFT results in a heavy computational load on programmable DSP processors. It also proves that a cost-efficient hardware implementation of radix-2 IFFT/FFT butterfly type of structure is still not feasible, since it requires a large number of multipliers and adders, which is of the order of $N\log_2 N$.

We consider the orthogonal transforms from a time-recursive point of view instead of the whole block of data. We do so because in digital signal transmission, data arrive serially. In this paper, we have proposed two methods to derive the hardware-efficient parallel infinite impulse response (IIR) filter structures for the IFFT/FFT implementation based on the time-recursive approach outlined in [7]. The throughput of these schemes is one input sample per clock cycle. In the first method, we decompose the IFFT into an inverse modified discrete cosine transform (IMDCT) and an inverse modified discrete sine transform (IMDST); and the FFT is decomposed into a modified discrete cosine transform (MDCT) and a modified discrete sine transform (MDST). Then, based on the time-recursive approach, the transfer functions of the dually generated pairs of the IMDCT and IMDST, and the MDCT and MDST are realized. Based on these transfer functions, the IMDCT/IMDST and MDCT/MDST modules are proposed. All the structures are of second-order IIR filter types with different multiplier coefficients. We have then proposed the architectures to implement the IFFT using IMDCT and IMDST modules, and the FFT using the MDCT and MDST modules. In the second method, the IMDCT and IMDST are realized using the inverse discrete Hartley transform (IDHT). Similarly, the MDCT and MDST are computed from the discrete Hartley transform (DHT). The advantage of these methods is that they require less number of multipliers and adders. The methods proposed, therefore, require approximately a maximum of only 12.5 % of the multipliers and 15.6 % of the adders, when compared to the butterfly type of implementation. The advantage of the schemes proposed is that they require lesser number of multipliers than those proposed in [8]. Furthermore, the modules used in the IFFT/FFT architecture are regular and modular. The second method proposed has lesser hardware complexity than the first one, but it takes twice the time to get the required output.

B. Notation Convention

The 'notation convention' used in the forthcoming Sections are given below:

$X_p(k)$: p-th symbol of the IFFT input
$X_{p,r}(k)$: real part of $X_p(k)$
$X_{p,i}(k)$: imaginary part of $X_p(k)$
$X_r(k)$: sequence of real parts of all successive symbols $X_p(k)$
$X_i(k)$: sequence of imaginary parts of all successive symbols $X_p(k)$
$X_{ri}(k)$: sequence of interleaved real and imaginary parts of all successive symbols $X_p(k)$
$x_c(n,t)$: IMDCT of $[X_r(t),...,X_r(t+N-1)]]$
$x_c(n,t+1)$: IMDCT of $[X_r(t+1),...,X_r(t+N)]$
$x_s(n,t)$: IMDST of $[X_r(t),...,X_r(t+N-1)]$
$x_s(n,t+1)$: IMDST of $[X_r(t+1),...,X_r(t+N)]$
$x_p(n)$: p-th symbol of IFFT output
$x(n)$: sequence of successive symbols $x_p(n)$

C. Organization of the Paper

The paper is organized as follows: Section II derives the required expressions for the implementation of the IFFT and FFT modules in terms of IMDCT/ IMDST and MDCT/MDST modules, respectively. Section III discusses the realization of IFFT and FFT using IDHT/DHT modules. Section IV compares the hardware complexity involved in implementing the IFFT/FFT blocks in a typical DMT transceiver system and Section V enumerates the conclusions drawn.

$X_{0,r}(0), X_{0,r}(1), ..., X_{0,r}(N-1),$	$X_{1,r}(0), X_{1,r}(1), ..., X_{1,r}(N-1),$	....

(a)

$X_{0,i}(0), X_{0,i}(1), ..., X_{0,i}(N-1),$	$X_{1,i}(0), X_{1,i}(1), ..., X_{1,i}(N-1),$	....

(b)

Fig. 2. (a) $X_r(m)$ sequence, and (b) $X_i(m)$ sequence.

II. Realization of IFFT and FFT Architectures

The conceptual block diagram of a DMT transceiver, involving the IFFT/FFT, is depicted in Fig. 1. The DMT encoder generates the complex sub-symbols $X_p(k)$, $[k=0,1,....,N-1]$ which are fed to the IFFT. We use a subscript p in $X_p(k)$ to denote the p-th symbol. The real and imaginary parts of $X_p(k)$ are denoted by $X_{p,r}(k)$ and $X_{p,i}(k)$, respectively. To ensure that the IFFT generates only real-valued outputs $x_p(n)$, the conjugate symmetry conditions on the input $X_p(k)$ are imposed [1] and hence the encoder output $X_p(k)$ is modified as:

$$X_p(k)=\begin{cases}0, & k=0\\ X_p(k), & k=1,2,...,N-1\\ 0, & k=N\\ X_p^*(2N-k), & k=N+1,N+2,....,2N-1,\end{cases} \quad (1)$$

where $X_p(k)=X_{p,r}(k)+jX_{p,i}(k)$. The 0-th and N-th samples of every symbol are set to zero, since they are not used in the DSL applications. The steps involved in decomposing the IFFT are illustrated below:

As defined in [6], the IFFT of a p-th symbol, $X_p(k)$, of length $2N$ is:

$$x_p(n)=\frac{1}{2N}\sum_{k=0}^{2N-1}X_p(k)\exp(j\frac{2\pi}{2N}kn),\ n=0,1,...,2N-1 \quad (2)$$

By decomposing k into the first half and the second half; and using the facts that $X_p(0)=X_p(N)=0$, (2) becomes

$$\begin{aligned}x_p(n)&=\frac{1}{N}[\sum_{k=1}^{N-1}X_{p,r}(k)\cos(\frac{\pi}{N}nk)-\sum_{k=1}^{N-1}X_{p,i}(k)\sin(\frac{\pi}{N}nk)]\\ &=[x_{p,c}(n)-x_{p,s}(n)],\ n=0,1,...,2N-1\end{aligned} \quad (3)$$

where $x_{p,c}(n)$ is the N-point IMDCT of $X_{p,r}(k)$, $x_{p,s}(n)$ is the N-point IMDST of $X_{p,i}(k)$, respectively, and they are given by:

$$\begin{aligned}&x_{p,c}(n)=\frac{1}{N}[\sum_{k=1}^{N-1}X_{p,r}(k)\cos(\frac{\pi}{N}nk)],\\ &x_{p,s}(n)=\frac{1}{N}[\sum_{k=1}^{N-1}X_{p,i}(k)\sin(\frac{\pi}{N}nk)],\\ &n=0,1,...,2N-1\end{aligned} \quad (4)$$

The IMDCT and IMDST involve only real-valued computations. Furthermore, it can be shown that

$$\begin{aligned}&x_{p,c}(n)=x_{p,c}(2N-n),\ n=1,...,N-1\\ &x_{p,s}(n)=-x_{p,s}(2N-n),\ n=1,...,N-1\end{aligned}. \quad (5)$$

This relationship can help us in saving an additional 50% hardware complexity [8]. At the receiver side of the DMT system, the $2N$-point FFT is used to demodulate the received signal, which is given by:

$$X_p(k)=\sum_{n=0}^{2N-1}x_p(n)\exp(-j\frac{2\pi}{2N}nk),\ k=0,\ 1....,2N\text{-}1 \quad (6)$$

where $x_p(n)$ is real-valued. Therefore, (6) can be written as:

$$\begin{aligned}X_p(k)&=\sum_{n=0}^{2N-1}x_p(n)\cos(\frac{\pi}{N}nk)-j\sum_{n=0}^{2N-1}x_p(n)\sin(\frac{\pi}{N}nk),\\ &=X_{p,r}(k)-jX_{p,i}(k),k=0,1,...,2N-1\end{aligned} \quad (7)$$

where $X_{p,r}(k)$ is the $2N$-point MDCT of $x_p(n)$, and $X_{p,i}(k)$ is the $2N$-point MDST of $x_p(n)$, respectively, and they are given by:

$$\begin{aligned}&X_{p,r}(k)=\sum_{n=0}^{2N-1}x_p(n)\cos(\frac{\pi}{N}nk),\\ &X_{p,i}(k)=\sum_{n=0}^{2N-1}x_p(n)\sin(\frac{\pi}{N}nk),\\ &k=0,1,...,2N-1\end{aligned} \quad (8)$$

Both the MDCT and the MDST use $x_p(n),\ n=0,\ 1,...,2N-1$, as the inputs. Hence, we can employ real-valued kernels to implement the FFT. No complex-valued operations are required in computing the FFT. In addition, in the DMT system, the lower N-point FFT outputs are conjugate symmetric of the upper N-point outputs. Therefore, we can neglect the outputs $X_p(k)$, $k=N,\ N+1,...,2N-1$. Hence, this simple manipulation can save 50% of the hardware complexity [8].

A. Implementation of IFFT Using IMDCT and IMDST Modules

We implement IFFT using the IMDCT and the IMDST modules, the expressions for which are derived in the following subsection.

1) Derivations of Transfer Functions for IMDCT and IMDST

Since the input to the modules is a serial sequence, the serial sequences $X_r(m)$ and $X_i(m)$ are constructed from $X_{p,r}(k)$ and $X_{p,i}(k)$, respectively, and they are illustrated in Fig. 2. So $X_r(0)$ and $X_r(N)$ corresponds to $X_{0,r}(0)$ and $X_{1,r}(0)$, respectively.

We employ the time-recursive approach [7], [9] to implement the dual generation of IMDCT and IMDST modules. The IMDCT and IMDST of a sequential input data vector starting from $X_r(t)$ (or $X_i(t)$) and ending with $X_r(t+N-1)$ (or $X_i(t+N-1)$) are denoted as $x_c(n,t)$ and $x_s(n,t)$, respectively, and defined as:

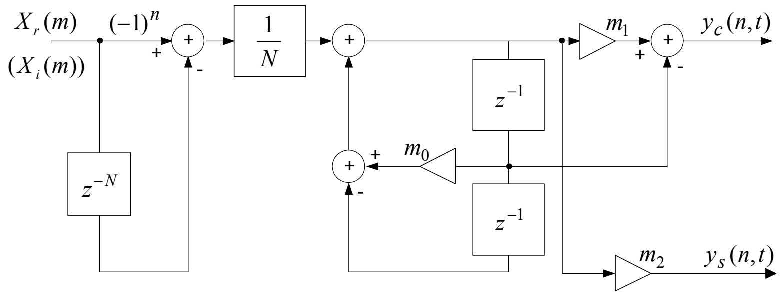

Fig. 3. IIR filter structure for the IMDCT and IMDST.

$$x_c(n,t) = \frac{1}{N}\sum_{m=t}^{t+N-1} X_r(m)\cos(\frac{\pi}{N}(m-t)n),\ n = 0,1,...,2N-1$$
$$x_s(n,t) = \frac{1}{N}\sum_{m=t}^{t+N-1} X_r(m)\sin(\frac{\pi}{N}(m-t)n),\ n = 0,1,...,2N-1 \tag{9}$$

where the time index t in $x_c(n,t)$ as well as in $x_s(n,t)$ denotes that the transform starts from $X_r(t)$. The data arrives serially. We are interested in the IMDCT and IMDST of the next input data vector, $[X_r(t+1), X_r(t+2),..., X_r(t+N)]$, which are denoted by $x_c(n,t+1)$ and $x_s(n,t+1)$, respectively, and is given by:

$$x_c(n,t+1) = \frac{1}{N}\sum_{m=t+1}^{t+N} X_r(m)\cos(\frac{\pi}{N}(m-t-1)n),$$
$$x_s(n,t+1) = \frac{1}{N}\sum_{m=t+1}^{t+N} X_r(m)\sin(\frac{\pi}{N}(m-t-1)n), \tag{10}$$
$$n = 0,1,...,2N-1.$$

We can rewrite (10) as:

$$x_c(n,t+1) = \tilde{x}_c(n,t+1)\cos(\frac{\pi}{N}n) + \tilde{x}_s(n,t+1)\sin(\frac{\pi}{N}n),$$
$$x_s(n,t+1) = \tilde{x}_s(n,t+1)\cos(\frac{\pi}{N}n) - \tilde{x}_c(n,t+1)\sin(\frac{\pi}{N}n),$$
$$n = 0,1,...,2N-1 \tag{11}$$

where

$$\tilde{x}_c(n,t+1) = \frac{1}{N}\sum_{m=t+1}^{t+N} X_r(m)\cos(\frac{\pi}{N}(m-t)n)$$
$$= x_c(n,t) - \frac{1}{N}X_r(t) + \frac{1}{N}(-1)^n X_r(t+N) \tag{12}$$
$$\tilde{x}_s(n,t+1) = \frac{1}{N}\sum_{m=t+1}^{t+N} X_r(m)\sin(\frac{\pi}{N}(m-t)n) = x_s(n,t).$$

The time difference equations for dually generated pairs can be obtained from (11) by considering $x_c(n,t+1)$ and $x_s(n,t+1)$ as $y_c(n,t)$ and $y_s(n,t)$, respectively, which can be written as follows:

$$y_c(n,t) = [y_c(n,t-1) - \frac{1}{N}X_r(t) + \frac{1}{N}(-1)^n X_r(t+N)] \times$$
$$\cos(\frac{\pi}{N}n) + y_s(n,t-1)\sin(\frac{\pi}{N}n)$$
$$\mathrm{y_s}(n,t) = \mathrm{y_s}(n,t-1)\cos(\frac{\pi}{N}n) - \sin(\frac{\pi}{N}n)[y_c(n,t-1) -$$
$$\frac{1}{N}X_r(t) + \frac{1}{N}(-1)^n X_r(t+N)]. \tag{13}$$

The z- transforms of the above difference equations are:

$$Y_c(n,z) = Y_s(n,z)z^{-1}\sin(\frac{\pi}{N}n) +$$
$$[Y_c(n,z)z^{-1} - \frac{1}{N}X_r(z) + \frac{1}{N}(-1)^n X_r(z)z^N]\cos(\frac{\pi}{N}n)$$
$$Y_s(n,z) = Y_s(n,z)z^{-1}\cos(\frac{\pi}{N}n) -$$
$$[Y_c(n,z)z^{-1} + \frac{1}{N}(-1)^n X_r(z)z^N - \frac{1}{N}X_r(z)]\sin(\frac{\pi}{N}n). \tag{14}$$

By solving (14) and by including a delay of N (to make them causal systems), we get the transfer functions of IMDCT and IMDST, which are denoted as $H_{ci}(z)$ and $H_{si}(z)$, respectively, and are given as follows:

$$H_{ci}(z) = \frac{Y_c(n,z)}{X_r(z)} = \frac{1}{N}\frac{[(-1)^n - z^{-N}][\cos(\frac{\pi}{N}n) - z^{-1}]}{1 - 2z^{-1}\cos(\frac{\pi}{N}n) + z^{-2}}$$
$$H_{si}(z) = \frac{Y_s(n,z)}{X_r(z)} = \frac{1}{N}\frac{[(-1)^n - z^{-N}][-\sin(\frac{\pi}{N}n)]}{1 - 2z^{-1}\cos(\frac{\pi}{N}n) + z^{-2}} \tag{15}$$

2) *Parallel IIR Structure for the IFFT*

Based on the transfer functions derived in (15), the IMDCT and the IMDST modules are realized using a single universal filter module consisting of a shift register array of size N, which is initially stored with zeros (need not reset to zero after each symbol is processed); and a second order IIR filter. This structure is depicted in Fig. 3. For the given input $X_r(m)$, every N-th sample of $y_c(n,t)$ and $y_s(n,t)$ gives the IMDCT and the IMDST of $X_{p,r}(k)$, $[k = 0,1,...,N-1]$, respectively, to that index n. The N^{th} sample of $y_c(n,t)$ corresponds to the IMDCT of $X_{0,r}(k)$ and the $2N$-th sample of $y_c(n,t)$ corresponds to the IMDCT of $X_{1,r}(k)$, to that index n. Similarly, IMDCT and IMDST of the other symbols can be obtained for different inputs $X_r(m)$ or $X_i(m)$. A total of N-modules are required to get all the IMDCT and the IMDST of a given input sequence of block of length N simultaneously. The multiplier coefficients m_0, m_1 and m_2 of the n-th module can be obtained by selecting the corresponding n values in the expressions for m_0, m_1 and m_2 which are given below:

$$m_0 = 2\cos(\frac{\pi}{N}n),\ m_1 = \cos(\frac{\pi}{N}n) \text{ and } m_2 = -\sin(\frac{\pi}{N}n).$$

Based on (3), the IFFT of $X_{p,r}(k)$, $[k = 0,1,....,2N\text{-}1]$ are computed using these modules by adding every N-th sample of IMDCT of $X_r(k)$ and IMDST of $X_i(k)$, for each index n. Since IMDCT and IMDST are computed for

$X_{0,i}(0),\ ...,X_{0,i}(N-1),$	$X_{0,r}(0),\ ...,X_{0,r}(N-1),$	$X_{1,i}(0),\ ...,X_{1,i}(N-1),$	$X_{1,r}(0),\ ...,X_{1,r}(N-1),$	....

Fig. 4. $X_{ri}(m)$ sequence.

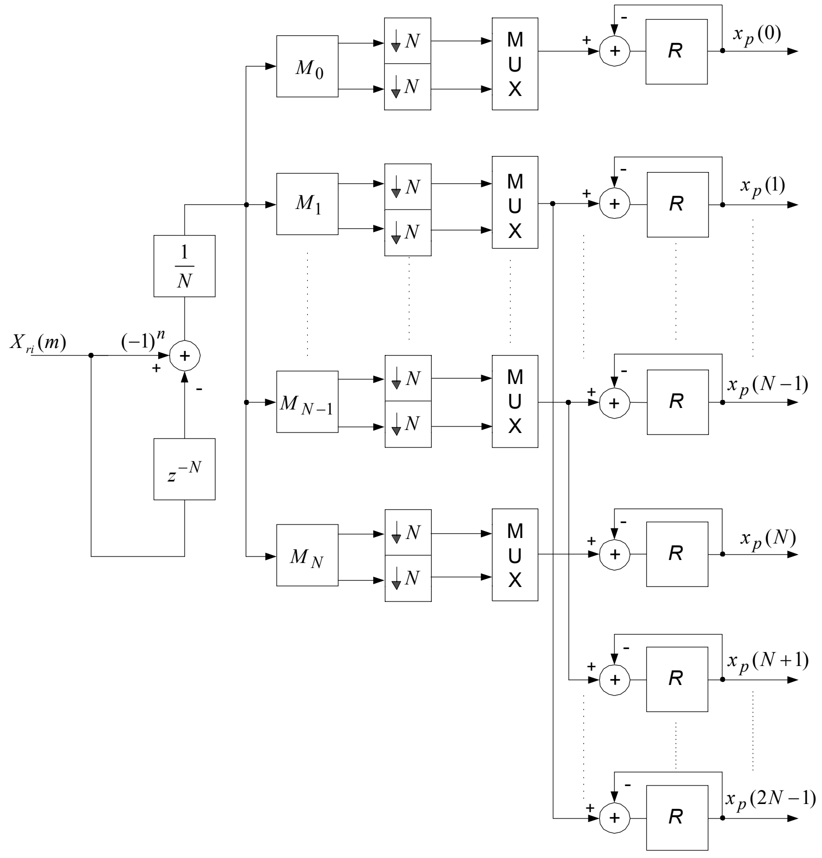

Fig. 5. IFFT architecture using IMDCT and IMDST modules.

$x_0(0), x_0(1), ..., x_0(2N-1),$	$x_1(0), x_1(1), ..., x_1(2N-1),$	....

Fig. 6. $x(n)$ sequence.

different inputs, we require two modules for each n. This redundancy can be avoided by using one module for each n. To achieve this, we construct a new sequence, say $X_{ri}(m)$, by interleaving the symbols $X_{p,r}(k)$ and $X_{p,i}(k)$ as shown in Fig. 4. Since we require IMDCT for the input $X_{p,r}(k)$ and IMDST for the input $X_{p,i}(k)$, multiplexers (MUX) are used to select the required one between the two outputs of each module for that particular input and bypass the other one. The N-fold decimators are used to pick-up every N-th sample from the output of the modules. By using (5) and $N+1$ modules $(M_0, M_1, ..., M_N)$, we construct $2N$-point IFFT. The IFFT architecture based on these modules is illustrated in Fig. 5. First, we pass the imaginary part of the first symbol, $X_{0,\ i}(k)$, $[k=0,1,...,N\text{-}1]$ sequentially through all the modules, and then the decimators select the N-th sample of the outputs of all these modules, which are stored in R-registers. Now, the real part of the first symbol, $X_{0,\ r}(k)$, $[k=0,1,...,N\text{-}1]$ is passed sequentially through all the modules, and then the N-th samples of all the outputs of the modules are added to their respective register (R) contents, which give the IFFT of the first symbol $X_0(k)$, $[k=0,1,...,2N\text{-}1]$. Similarly, the IFFT of the other symbols can be computed from the given input $X_{ri}(m)$.

B. Implementation of FFT using MDCT and MDST Modules

In (8), $X_{p,r}(k)$ and $X_{p,i}(k)$ are defined as the MDCT and the MDST of $x_p(n)$, respectively. Sequence $x(n)$ is obtained from $x_p(n)$ as illustrated in Fig. 6. The MDCT and MDST of a sequential input data vector starting from $x(t)$ and ending with $x(t+2N-1)$ are denoted by $X_r(k,t)$ and $X_i(k,t)$, respectively.

$X_r(k,t+1)$ and $X_i(k,t+1)$ are the MDCT and the MDST of the next input data vector starting from $x(t+1)$ and ending with $x(t+2N)$. Following the derivations given in (9) through (14), we get the transfer functions of the MDCT and the MDST, $H_c(z)$ and $H_s(z)$, respectively, as follows:

$$H_c(z) = \frac{Y_r(k,z)}{X(z)} = \frac{[1-z^{-2N}][\cos(\frac{\pi}{N}k)-z^{-1}]}{1-2z^{-1}\cos(\frac{\pi}{N}K)+z^{-2}}$$

$$H_s(z) = \frac{Y_i(k,z)}{X(z)} = \frac{[1-z^{-2N}][-\sin(\frac{\pi}{N}k)]}{1-2z^{-1}\cos(\frac{\pi}{N}k)+z^{-2}} \quad . \quad (16)$$

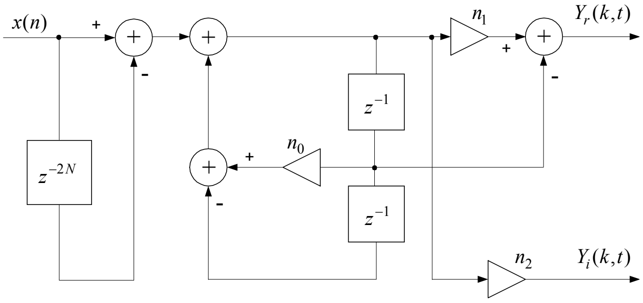

Fig. 7. IIR filter structure for the MDCT and MDST.

The MDCT and MDST IIR filter structure based on the above transfer functions is shown in Fig. 7. The multiplier coefficients n_0, n_1 and n_2 of the k-th module can be obtained by choosing the corresponding k values in the expressions for n_0, n_1 and n_2, which are given below:

$$n_0 = 2\cos(\frac{\pi}{N}k),\ n_1 = \cos(\frac{\pi}{N}k) \text{ and } n_2 = -\sin(\frac{\pi}{N}k).$$

The FFT architecture with the MDCT and MDST modules is shown in Fig. 8. The $2N$-fold decimators select every $2N$-th sample from the output of the modules which gives $X_{p,r}(k)$ and $X_{p,i}(k)$ simultaneously. Since we are interested in $X_{p,r}(k)$ and $X_{p,i}(k)$, for $k = 0,1,....,N-1$, we have used N such modules in the FFT architecture, as shown in Fig. 8.

III. Realization of IFFT/FFT Using IDHT/DHT Modules

In the first method, the IFFT/FFT architectures have been realized from the DCT-like and the DST-like modules. In the second method, which is outlined below, the DCT-like and DST-like functions are computed using the IDHT/DHT functions. The IDHT/DHT transfer functions are derived based on the time-recursive approach from which the required modules are constructed. Finally, using these modules, the IFFT/FFT architectures are realized. The main advantage of this method is that the complexity of the previous method can be reduced even further at the cost of time needed to calculate the result.

A. Implementation of IDHT and DHT Modules

As defined in [10], the IDHT of a $2N$-point data sequence, $X_{p,h}(k)$, is

$$x_{p,h}(n) = \frac{1}{2N}\sum_{k=0}^{2N-1} X_{p,h}(k)\,\text{cas}(\frac{2\pi}{2N}nk), \quad n = 0,1,...,2N-1 \tag{17}$$

and the DHT of a $2N$-point data sequence, $x_{p,h}(n)$, is

$$X_{p,h}(k) = \sum_{n=0}^{2N-1} x_{p,h}(n)\text{cas}(\frac{2\pi}{2N}nk), \quad k = 0,1,...,2N-1 \tag{18}$$

where $\text{cas}(\frac{\pi}{N}nk) = \cos(\frac{\pi}{N}nk) + \sin(\frac{\pi}{N}nk).$

The IDHT and the DHT transfer functions can be obtained in a manner similar to that of the IMDCT/IMDST transfer functions [9], which have been derived in Section II. However, the transfer functions, IDHT and DHT differ by a factor of $1/2N$ only. Following the same procedure followed in Section II, the transfer functions using the IDHT/DHT can be found as given in (19) and (20) below:

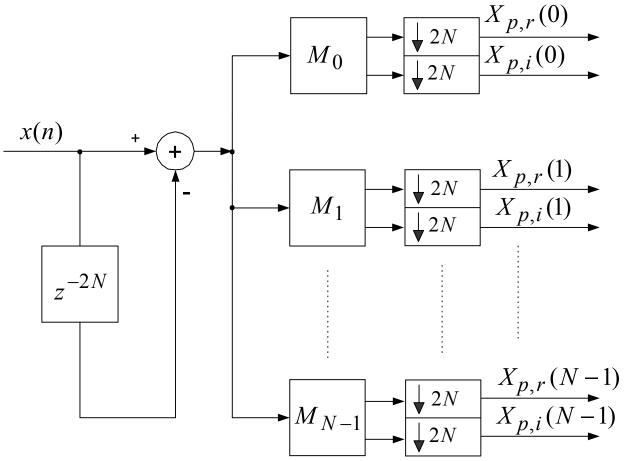

Fig. 8. FFT architecture using the MDCT and MDST modules.

$$H_h(n,z) = \frac{1}{2N}\frac{[1-z^{-2N}][\cos(\frac{\pi}{N}n) - \sin(\frac{\pi}{N}n) - z^{-1}]}{1-2z^{-1}\cos(\frac{\pi}{N}n) + z^{-2}} \tag{19}$$

$$n = 0,1,...,2N-1$$

and

$$H_h(k,z) = \frac{[1-z^{-2N}][\cos(\frac{\pi}{N}k) - \sin(\frac{\pi}{N}k) - z^{-1}]}{1-2z^{-1}\cos(\frac{\pi}{N}k) + z^{-2}}. \tag{20}$$

$$k = 0,1,...,2N-1$$

Based on the above transfer functions, the generalized structure for IDHT/DHT is constructed and is shown in Fig. 9. The multiplication factor C is $1/2N$ for the IDHT module while it is 1 for the DHT module. For IDHT, the multiplier coefficients are $Q_0 = 2\cos(\pi n/N)$ and $Q_1 = \cos(\pi n/N) - \sin(\pi n/N)$ and for the DHT, the multiplier coefficients are $Q_0 = 2\cos(\pi k/N)$ and $Q_1 = \cos(\pi k/N) - \sin(\pi k/N)$. Every $2N$-th sample of the n-th (or k-th) module output $y(t)$ gives the IDHT (or DHT) of the blocks (length of each block is equal to $2N$) of $x(t)$ to that of index n (or k).

B. IMDCT and IMDST from IDHT

In order to get the IMDCT and the IMDST of (4) using the IDHT IIR filter structure, we define two $2N$-point sequences as follows:

$$X_{p,hr}(k) = \begin{cases} X_{p,r}(k), & k = 0,1,...,N\text{-}1 \\ X_{p,r}(2N-k), & k = N,...,2N\text{-}1 \end{cases}$$
$$X_{p,hi}(k) = \begin{cases} X_{p,i}(k), & k = 0,1,...,N\text{-}1 \\ -X_{p,i}(2N-k), & k = N,...,2N\text{-}1 \end{cases} \tag{21}$$

Now, the IDHT of $X_{p,hr}(k)$ and $X_{p,hi}(k)$ give the IMDCT of $2(X_{p,r}(k))$ and IMDST of $2(X_{p,i}(k))$, respectively.

C. IFFT Architecture Using IDHT Modules

Fig. 10 shows the IFFT architecture using the IDHT modules. The common input $X_{hri}(k)$ to the modules is shown in Fig. 11. The operation of IFFT architecture via

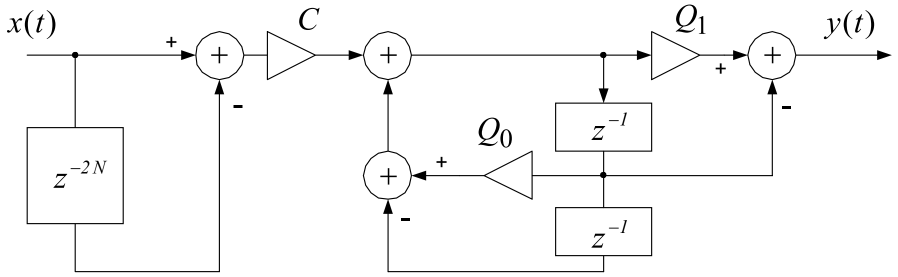

Fig. 9. IIR filter structure for IDHT/DHT.

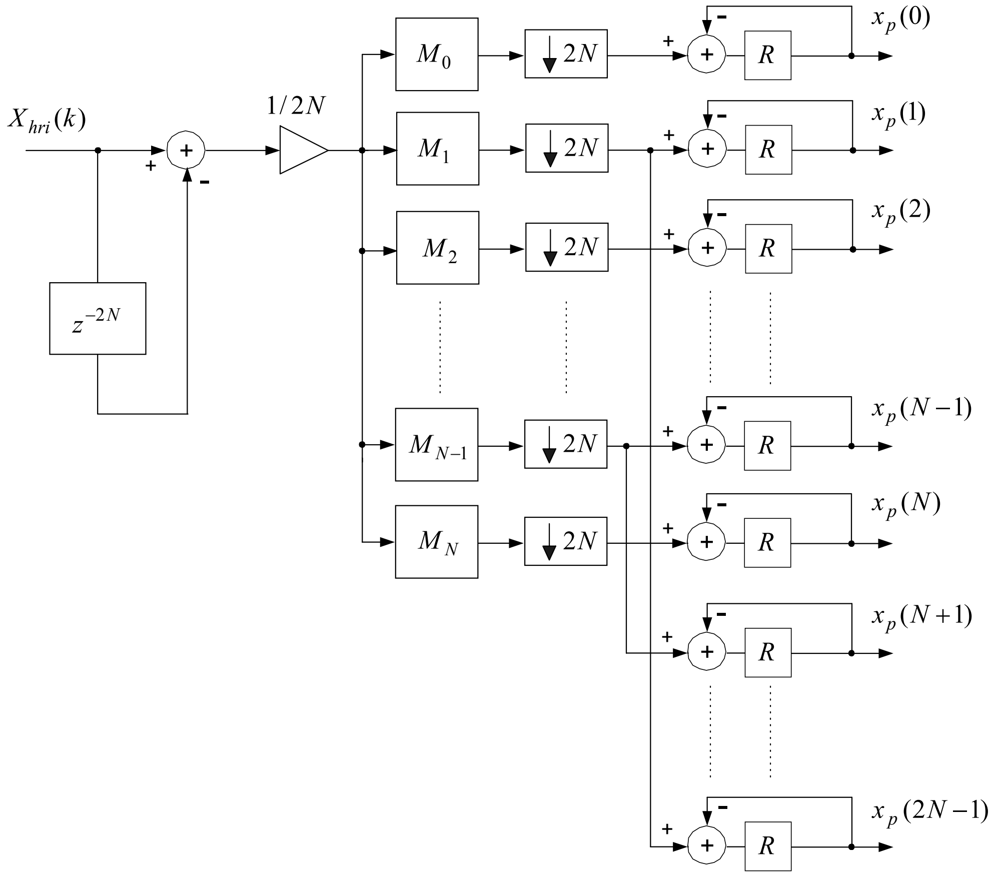

Fig. 10. IFFT architecture using IDHT modules.

$x_{0,hi}(0),..., x_{0,hi}(2N-1),$	$x_{0,hr}(0),..., x_{0,hr}(2N-1),$	$x_{1,hi}(0),..., x_{1,hi}(2N-1),$	....

Fig. 11. $X_{hri}(k)$ sequence

IDHT modules is similar to that explained in Section II for Fig. 5.

D. FFT Architecture Using DHT Modules

We use the DHT module to get $X_{p,r}(k)$ and $X_{p,i}(k)$ of (8). We define a new $2N$-point sequence, $x_{p_1}(n)$, as:

$$x_{p_1}(0) = x_p(0)$$
$$x_{p_1}(n) = x_p(2N-n),\ n = 1,2,...,2N-1\ ^{\cdot}$$

The sum of the DHT of $x_p(n)$ and $x_{p_1}(n)$ produces the MDCT $2X_{p,r}(k)$. Similarly, the sum of the DHT of $x_p(n)$ and $-x_{p_1}(n)$ produces the MDST $2X_{p,i}(k)$. Fig. 12 shows the FFT architecture using the DHT modules. The common input $x_1(n)$ to the modules is defined in Fig. 13.

IV. Hardware Complexity Analysis and Results of Comparison

In this section, we compare and contrast the hardware complexity of our proposed IFFT/ FFT architecture with the direct implementation using the butterfly type of approach [6] and the architecture proposed in [8]. In [8], the authors use the parallel lattice structures derived in [11] to realize the IFFT/ FFT blocks of a DMT. Since it uses N lattice modules, the latency and throughput of this parallel realization are N and 1, respectively. Since there is no global communication and the structure is modular and regular, it is suitable for practical VLSI implementation.

In the present paper, we consider the optimal unified IIR filter structures, based on the time-recursive approach

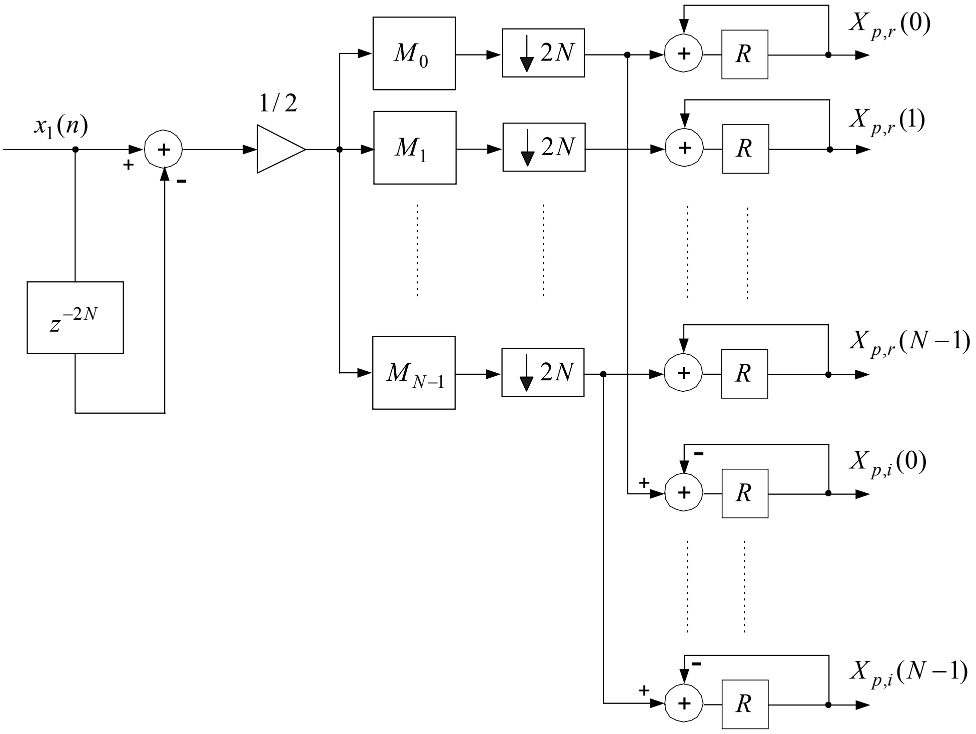

Fig. 12. FFT architecture using DHT modules.

$x_{01}(0), x_{01}(1), ..., x_{01}(2N-1),$	$x_0(0), x_0(1), ..., x_0(2N-1),$	....

Fig. 13. $x_l(n)$ sequence

proposed in [7], for the realization of IFFT/ FFT blocks in a DMT. These filter architectures preserve the advantages of the lattice architecture while reducing the hardware complexity nearly by half. Unlike the FFT, this architecture has only local interconnections and is better suited for VLSI implementation. Also, an area-time complexity analysis provided in [7] shows that the proposed approach is asymptotically optimal in speed and area. Moreover, unlike many fast algorithms for DFT and DHT, there is no constraint on the transform size N.

In this paper, since we have decomposed the IFFT and FFT into real-valued transform kernels (DCT-like and DST-like), they are implemented by modules consisting of real multipliers and real adders. The direct (butterfly-type) implementation of a $2N$-point IFFT block requires $4N\log_2 N+4N$ real multipliers and $6N\log_2 N+6N$ real adders, since the data is complex. In the case of the FFT block, which is used in the demodulator part of the receiver, the direct (butterfly-type) implementation requires only $2N\log_2 N+8N$ real multiplications and $3N\log_2 N+8N$ real additions [12], since the input to the demodulator part of the DMT is real. Therefore, a further reduction of arithmetic operations is possible in the FFT block of the demodulator, while considering the butterfly type of implementation. This factor has not been taken into account by the authors of [8] for the butterfly type of implementation. Consequently, they obtain a better arithmetic complexity ratio (CR), which is defined later in this section, especially for the FFT computation, as can be seen from Table I of [8].

Now, coming back to the two methods presented in this paper (refer to Sections II and III for details), the arithmetic complexity involved can be explained as follows: For the sake of simplicity, let us consider the procedure explained in sub-Sections II-A and II-B as method 1 and that of sub-Sections III-C and III-D as method 2. In method 1, we use $N+1$ modules, each of which requires three real multipliers and three real adders to implement the $2N$-point IFFT [13]. Moreover, the overall IFFT architecture of Fig. 5 needs $2N$ extra adders for adding the IMDST and IMDCT outputs and one adder for the realization of $1-z^{-N}$. The multiplier $1/N$ is common to all the modules. To sum up, we need a total of $3N+4$ real multipliers and $5N+4$ real adders to implement the IFFT architecture of Fig. 5. However, the method proposed in [8] requires $4N-4$ real multipliers and $5N-3$ real adders. Based on similar lines, to implement the FFT architecture presented in Fig. 8, we require $3N$ real multipliers and $3N+1$ real adders, whereas the method proposed in [8] takes $4N-4$ real multipliers and $3N-2$ real adders. If we define the complexity ratio (CR) as $CR = M/B$, where B and M are the number of real multipliers (or real adders) in the direct butterfly type of approach and the two methods proposed, respectively, the result of comparison of method 1 and the method proposed in [8] can be represented as given in Tables I and II, for real multipliers and real adders, respectively.

Similarly, for the second method proposed in this paper (referred to as method 2), we need a total of $2N+3$ real multipliers and $5N+4$ real adders to implement the IFFT architecture of Fig. 10. To implement the corresponding FFT architecture (refer to Fig. 12), we require $2N+1$ real

TABLE I
COMPARISON OF REAL MULTIPLIERS FOR THE $2N$-POINT IFFT/FFT ARCHITECTURE

	IFFT				FFT			
N	Butterfly	Proposed method in [8]	Proposed method 1	CR	Butterfly	Proposed method in [8]	Proposed method 1	CR
256	9216	1020	772	8.38%	6144	1020	768	12.5%
512	20480	2044	1540	7.52%	13312	2044	1536	11.6%
1024	45056	4092	3076	6.83%	28672	4092	3072	10.7%

TABLE II
COMPARISON OF REAL ADDERS FOR THE $2N$-POINT IFFT/FFT ARCHITECTURE

	IFFT				FFT			
N	Butterfly	Proposed method in [8]	Proposed method 1	CR	Butterfly	Proposed method in [8]	Proposed method 1	CR
256	13824	1277	1284	9.29%	8192	766	769	9.4%
512	30720	2557	2564	8.35%	17920	1534	1537	8.6%
1024	67584	5117	5124	7.59%	38912	3070	3073	7.9%

TABLE III
COMPARISON OF REAL MULTIPLIERS FOR THE $2N$-POINT IFFT/FFT ARCHITECTURE

	IFFT				FFT			
N	Butterfly	Proposed method in [8]	Proposed method 2	CR	Butterfly	Proposed method in [8]	Proposed method 2	CR
256	9216	1020	515	5.59%	6144	1020	513	8.35%
512	20480	2044	1027	5.01%	13312	2044	1025	7.7%
1024	45056	4092	2051	4.55%	28672	4092	2049	7.1%

TABLE IV
COMPARISON OF REAL ADDERS FOR THE $2N$-POINT IFFT/FFT ARCHITECTURE

	IFFT				FFT			
N	Butterfly	Proposed method in [8]	Proposed method 2	CR	Butterfly	Proposed method in [8]	Proposed method 2	CR
256	13824	1277	1284	9.29%	8192	766	1281	15.6%
512	30720	2557	2564	8.35%	17920	1534	2561	14.3%
1024	67584	5117	5124	7.59%	38912	3070	5121	13.1%

multipliers and $5N+1$ real adders. The results of comparison of method 2 and the method presented in [8], along with the direct butterfly type of approach, is tabulated in Tables III and IV, for real multipliers and real adders, respectively. The hardware complexity of different methods considered in this paper for a $2N$-point FFT/IFFT is presented in Tables I through IV. From these Tables, it can be seen that for our methods, we require a maximum of only 12.5% of the multipliers and 15.6% of the adders when compared to the direct butterfly type of approach. Moreover, the number of real multipliers required in our methods are much smaller than those required in [8], while there is a very slight increase in the number of real adders.

We have not considered the computational complexity of Goertzel's algorithm, since it is not applicable in the present context [14]. The relevance of IFFT/ FFT in the context of a DMT receiver system has been extensively discussed in [15]. The input/ output operations of the proposed methods are serial in/ parallel out (SIPO), since the modules take the inputs serially and give the outputs in parallel. On the other hand, the operations are parallel in/ parallel out (PIPO) in the direct butterfly type of implementation. The number of modules required for both the methods proposed in this paper depends on the value of the transform length N. An area-time complexity analysis provided in [7] shows that the proposed approach is asymptotically optimal in speed and area.

V. CONCLUSIONS

Discrete multitone (DMT) is a multicarrier transmission technique that uses fast Fourier transform (FFT) and inverse FFT [15]. In this paper, we have derived efficient parallel IIR filter structures based on the time-recursive approach, for the implementation of IFFT/FFT blocks in a DMT system. Based on this, we have proposed two methods for cost-efficient implementation of a large block size IFFT/FFT in the DMT/OFDM transceiver system. The proposed methods require lesser number of multipliers than those required by the architectures described in [8]. The second method proposed requires even lesser number of multipliers than those required by the first method, but takes twice the time to get the required output. The complexity ratios of the multipliers and adders of the proposed methods to the direct butterfly type of approach are approximately a maximum of only 12.5% and 15.6%, respectively.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers for their constructive comments and suggestions, which have greatly enhanced the presentation of the paper.

REFERENCES

[1] J. S. Chow, J. C. Tu, and J. M. Cioffi, "A discrete multitone transceiver system for HDSL applications," *IEEE J. Select. Areas Commun*., vol. 9, no. 6, pp. 895-908, Aug. 1991.

[2] T. N. Zogakis, J. T. Aslanis Jr., and J. M. Cioffi, "A coded and shaped discrete multitone system," *IEEE Trans. on Commun*., vol. 43, no. 12, pp. 2941-2949, Dec. 1995.

[3] G. H. Im, D. D. Harman, G. Huang, A. V. Mandzik, M. H. Nguyen, and J. J. Werner, "51.84 Mb/s 16-CAP ATM LAN standard," *IEEE J. Select. Areas Commun.*, vol. 13, no. 4, pp. 620-632, May 1995.

[4] B. Dameshrad, and H. Samueli, "A 1.6 Mbps digital-QAM system for DSL transmission," *IEEE J. Select. Areas Commun*, vol. 13, no. 9, pp. 1600-1610, Dec. 1995.

[5] W. Y. Zou, and Y. Wu, "COFDM: An overview," *IEEE Trans. on Broadcasting,* vol. 41, no. 1, pp. 1-8, Mar. 1995.

[6] A. V. Oppenheim, and R. W. Schafer, *Discrete-Time Signal Processing,* Prentice-Hall, Englewood Cliffs, N. J., 1989, pp. 514-560.

[7] K. J. R. Liu, C. T. Chiu, K. Kolagotla, and J. F. Jaja, "Optimal unified architectures for the real-time computation of time-recursive discrete sinusoidal transforms," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, no. 2, pp. 168-180, Apr. 1994.

[8] A. N. Wu, and T. S. Chan, "Cost-efficient parallel lattice VLSI architecture for the IFFT/FFT in DMT transceiver technology," in *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP),* pp. 3517-3520, Seattle, May 1998.

[9] A. M. S. Rao, *A Multicarrier System: Discrete Multitone Transceiver*, M. S Thesis, IIT Madras, India, 2001.

[10] R. N. Bracewell, "The fast Hartley transform," *Proc. IEEE*, vol. 72, no. 8, pp. 1010-1018, Aug. 1984.

[11] K. J. R. Liu and C. –T. Chiu, "Unified parallel lattice structures for time-recursive discrete cosine/sine/Hartley transforms," *IEEE Trans. on Signal Processing*, vol. 41, no. 3, pp. 1357-1377, Mar. 1993.

[12] R. G. Lyons, *Understanding Digital Signal Processing*, Addison Wesley Publishing Company, 1997, pp. 412-425.

[13] A. M. S. Rao, K. M. M. Prabhu, and P. C. W. Sommen, "Parallel IIR lattice filter structures for the IFFT/FFT in DMT transceiver," in *Proc. 2nd IEEE Benelux Signal Processing Symposium,* pp. S001-S004, Hilvarenbeek, The Netherlands, Mar. 2000.

[14] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, McGraw-Hill Inc., 1998, pp. 520-523.

[15] T. Starr, J. M. Cioffi, and P. Silverman, *Understanding Digital Subscriber Line Technology*, Prentice-Hall PTR, N.J., 1999, pp. 242-251.

K. M. M. Prabhu obtained the B.Sc. (Eng.) degree in Electronics and Communication Engineering from the Government College of Engineering, Trivandrum, the M.Sc. (Engg.) degree in Applied Electronics Engineering from M.I.T., Madras and the Ph.D. degree in Digital Signal Processing from the Indian Institute of Technology Madras, India, in 1971, 1973 and 1981, respectively. He joined the Department of Electrical Engineering, Indian Institute of Technology (IIT) Madras, in 1981, where he is currently a full Professor.

While at IIT Madras, he has been a Principal Investigator for many projects funded by the Defence Research and Development Organization. He has published more than 95 papers in the International Journals and the International Conference Proceedings. Under his supervision, 17 M. S. (by research) and Ph. D. students have completed their degrees. He has conducted 10 short-term courses in the area of DSP for the benefit of Engineers and Scientists from various organizations. He spent his sabbatical, from August 1999 to August 2000, with the Signal Processing Systems Group, Eindhoven University of Technology, Eindhoven, The Netherlands. He has been a reviewer for a number of IEEE, IEE and other International Journals. His current research interests include DSP algorithms and their implementations, digital filter structures, time-frequency signal analysis and Wavelet-based signal processing. Prof. Prabhu is a Senior Member of the IEEE.

A. Madhu Sudana Rao obtained the Engineering degree (A.M.I.E.) from the Institute of Engineers in 1995 and the M.S. (by research) from the Indian Institute of Technology Madras in 2000. Since then, he has been working with the SASKEN Communication Technologies Limited, Bangalore, India, as a Software Engineer. His research interests include signal processing applied to communications.

P. C. W. Sommen received the Ingenieur degree in Electrical Engineering from Delft University of Technology in 1981 and his Ph.D. from Eindhoven University of Technology in 1992. From 1981 to 1989, he was with Philips Research Laboratories, Eindhoven and since 1989, with the faculty of Electrical Engineering at Eindhoven University of Technology, where he is currently an Associate Professor. Dr. Sommen is involved in internal and external courses, all dealing with different basic and advanced signal processing topics. His main field of research is in adaptive array signal processing, with applications in acoustic communication systems. Dr. Sommen is a member of the ProRISC board, Vice President of the IEEE Benelux Signal Processing Chapter and Officer of the Administrative Committee of EURASIP. Besides, he is Editor of EURASIP's Journal of Applied Signal Processing.