

COST-EFFICIENT PARALLEL LATTICE VLSI ARCHITECTURE FOR THE IFFT/FFT IN DMT TRANSCIVER TECHNOLOGY

An-Yeu Wu and Tsun-Shan Chan

Department of Electrical Engineering
National Central University
Chung-Li, 32054, Taiwan, ROC

ABSTRACT

The discrete multitone (DMT) modulation/demodulation scheme is the standard transmission technique in the application of *asymmetric digital subscriber lines (ADSL)*. Although the DMT can achieve higher data rate compared with other modulation/demodulation schemes, its computational complexity is too high for cost-efficient implementations. For example, it requires 512-point IFFT/FFT as the modulation/demodulation kernel. The large block size results in heavy computational load in running programmable DSP processors. It also makes the VLSI implementation not feasible. In this paper, we derive the parallel lattice structure for the IFFT/FFT based on the time-recursive approach. The resulting architectures are regular, modular, and without global communications so that they are very suitable for VLSI implementation. Also, the proposed structure requires only 11% number of multipliers and 9% number of adders compared with the direct implementation approach.

1. INTRODUCTION

Recent progress of Internet access has a strong demand on high-speed data transmission. To overcome the transmission bottleneck over the conventional twisted-pair telephone lines, several sophisticated modulation/demodulation schemes have been proposed, including carrierless-amplitude-phase (CAP) modulation [1], discrete multitone modulation (DMT) [2], [3], and QAM technology [4]. Among these advanced modulation schemes, the DMT can achieve highest transmission rate since it incorporates lots of advanced DSP techniques such as dynamic bit allocation, multi-dimensional tone encoding, frequency-domain equalization, *etc.* As a consequence, the DMT has been chosen as the physical-layer transmission standard by the ADSL standardization committee.

One major disadvantage of the DMT scheme is its high computational complexity. In particular, the large block size of the IFFT/FFT (512-point) makes VLSI implementation very cost-expensive. In this paper, we propose cost-efficient implementations of the IFFT/FFT modules based on the time-recursive transform coding lattice structures [6], [7]. We first decompose the IFFT/FFT into a combination of a DCT-like and a DST-like transform functions. By making use of the symmetric property of the Fourier transform as well as the fast algorithms of the DCT and DST, we derive new schemes to replace the complex-domain IFFT/FFT. The new schemes can avoid redundant operations of the IFFT/FFT, and involve only real-valued operators. Then we employ the time-recursive parallel lattice structures [6] to implement our

derivation. The structure computes the transformed data from sequential inputs in a time-recursive way and requires only $O(N)$ hardware complexity. Due to the linear hardware complexity and real-valued operations, we will need only 11% multipliers and 9% adders compared with direct butterfly implementation of the IFFT/FFT.

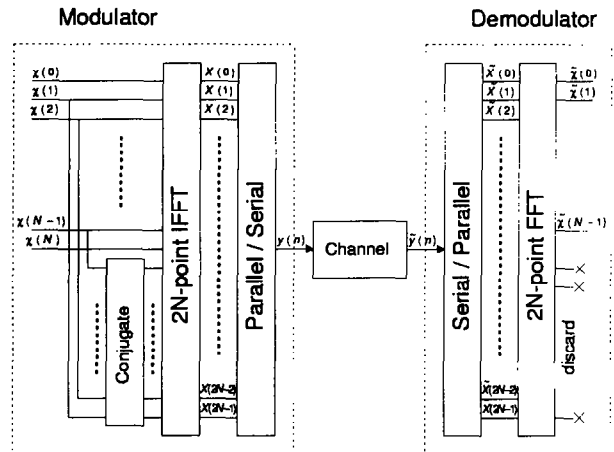


Figure 1: The IFFT/FFT block diagram in the DMT system.

2. IFFT MODULE

2.1. The IFFT Derivation

The IFFT/FFT block diagram in the DMT system is shown in Fig. 1, where $N=256$ in the current standard. At the transmitter side, to ensure the IFFT generates only real-valued outputs, the inputs of the IFFT have the constraint [2]

$$x(n) = x^*(2N - n), \quad \text{for } n = 0, 1, \dots, N - 1, \quad (1)$$

where $x(n) = x_r(n) + jx_i(n)$ are encoded complex symbols with $x(0) = x(N) = 0$. As defined in [5], the IFFT of a data sequence of length $2N$ is

$$X(k) = \sum_{n=0}^{2N-1} x(n) W_{2N}^{nk}, \quad \text{for } k = 0, 1, \dots, 2N - 1, \quad (2)$$

where $W_{2N}^{nk} \triangleq \exp(-j \frac{2\pi nk}{2N})$. By decomposing n into the first half and the second half and using the facts that $x(0) = x(N) = 0$, (2) becomes

$$X(k) = \sum_{n=1}^{N-1} x(n) W_{2N}^{nk} + \sum_{n=N+1}^{2N-1} x(n) W_{2N}^{nk}. \quad (3)$$

Next, by applying the conjugate-symmetric property of the input data, we can simplify (2) as

$$\begin{aligned} X(k) &= 2 \left[\sum_{n=1}^{N-1} x_r(n) \cos \frac{2\pi nk}{2N} + \sum_{n=1}^{N-1} x_i(n) \sin \frac{2\pi nk}{2N} \right] \\ &= 2[MDCT(k) + MDST(k)], \end{aligned} \quad (4)$$

for $k = 0, 1, \dots, 2N - 1$. From (4), we can also see that the computation of the IFFT is decomposed into two real-valued operations. One is a *discrete cosine transform (DCT)*-like operation with $x_r(n)$, $n = 1, 2, \dots, N - 1$, as the inputs. The other is a *discrete sine transform (DST)*-like operation with $x_i(n)$, $n = 1, 2, \dots, N - 1$, as the inputs. We shall name first term *modified DCT (MDCT)*, and second term *modified DST (MDST)*. Note that MDCT and MDST involve only real-valued computations. Furthermore, it can be shown that

$$\begin{aligned} MDCT(k) &= MDCT(2N - k), \\ MDST(k) &= -MDST(2N - k), \end{aligned} \quad (5)$$

for $k = 1, 2, \dots, N - 1$. So, we can focus on computing $MDCT(k)$ and $MDST(k)$ for $k = 1, 2, \dots, N - 1$. Then, expand the results for $k = N, N + 1, \dots, 2N - 1$. This simple relationship can help us to save an additional 50% hardware complexity.

2.2. Parallel Lattice Structure for the IFFT

Next, we derive the lattice VLSI architecture of the IFFT module. We employ the time-recursive approach in [6] to implement the IFFT/FFT modules. It is an efficient implementation of the dual generation of the MDCT and MDST. First, we consider the lattice module of the MDCT. We follow the derivation in [6] and define the MDCT of a sequential input data starting from $x_r(t)$ and ending with $x_r(t + N - 1)$ as follows:

$$X_c(k, t) = \sum_{n=t}^{t+N-1} x_r(n) \cos \frac{2\pi(n-t)k}{2N}, \quad (6)$$

for $k = 0, 1, \dots, 2N - 1$. After the new datum $x_r(t + N)$ arrives, the MDCT is updated as

$$X_c(k, t + 1) = \sum_{n=t+1}^{t+N} x_r(n) \cos \frac{2\pi(n-t-1)k}{2N}, \quad (7)$$

for $k = 0, 1, \dots, 2N - 1$. We can rewrite (7) as

$$X_c(k, t + 1) = \bar{X}_c(k, t + 1) \cos \frac{\pi k}{N} + \bar{X}_s(k, t + 1) \sin \frac{\pi k}{N}, \quad (8)$$

where

$$\bar{X}_c(k, t + 1) \triangleq \sum_{n=t+1}^{t+N} x_r(n) \cos \frac{2\pi(n-t)k}{2N} \quad (9)$$

and

$$\bar{X}_s(k, t + 1) \triangleq \sum_{n=t+1}^{t+N} x_r(n) \sin \frac{2\pi(n-t)k}{2N}. \quad (10)$$

On the other hand, we can relate $\bar{X}_c(k, t + 1)$ to $X_c(k, t)$ by deleting the old datum $x_r(t)$ and updating the new datum $x_r(t + N)$ as follows:

$$\bar{X}_c(k, t + 1) = X_c(k, t) - x_r(t) + (-1)^k x_r(t + N). \quad (11)$$

For block processing, the initial states are zeros. Hence, $\bar{X}_c(k, t + 1)$ can be simplified as

$$\bar{X}_c(k, t + 1) = X_c(k, t) + (-1)^k x_r(t + N). \quad (12)$$

Similarly, we can relate $\bar{X}_s(k, t + 1)$ to $X_s(k, t)$ as

$$\bar{X}_s(k, t + 1) = X_s(k, t). \quad (13)$$

Based on (8), (12), and (13), we can derive the complete time-recursive lattice module as shown in Fig. 2.

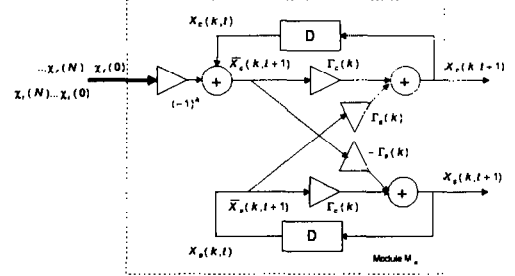


Figure 2: The lattice module (M_k , $k = 1, 2, \dots, N - 1$) for the dual generation of the MDCT and the MDST of the IFFT, where $\Gamma_c(k) \triangleq \cos \frac{\pi k}{N}$, $\Gamma_s(k) \triangleq \sin \frac{\pi k}{N}$.

Next, we want to consider the implementation of the MDST. As with the derivation of the MDCT, we define the MDST as

$$X_s(k, t) = \sum_{n=t}^{t+N-1} x_i(n) \sin \frac{2\pi(n-t)k}{2N}, \quad (14)$$

for $k = 0, 1, \dots, 2N - 1$. Following the derivations in (7) - (10), we can find

$$X_s(k, t + 1) = \bar{X}_c(k, t + 1) \cos \frac{\pi k}{N} - \bar{X}_s(k, t + 1) \sin \frac{\pi k}{N}, \quad (15)$$

where

$$\bar{X}_c(k, t + 1) \triangleq \sum_{n=t+1}^{t+N} x_i(n) \cos \frac{2\pi(n-t)k}{2N} \quad (16)$$

and

$$\bar{X}_s(k, t + 1) \triangleq \sum_{n=t+1}^{t+N} x_i(n) \sin \frac{2\pi(n-t)k}{2N}. \quad (17)$$

Note that (16) and (17) are similar to (9) and (10) except that the inputs are $x_i(n)$. Hence, the updating equations in (12) and (13), can be applied to (16) and (17) except that the inputs become $x_i(n)$. As a result, the lattice module in Fig. 2 can be used to generate MDCT and MDST outputs simultaneously. For $k = 0$ and N , the IFFT outputs become $X(0) = \sum_{n=0}^{N-1} x_r(n)$ and $X(N) = \sum_{n=0}^{N-1} (-1)^n x_r(n)$, respectively. These two special cases can be implemented by using accumulators. We denote this special module as *SC1*. The complete architecture of the IFFT is shown in Fig. 3. The *Module Array* consists of $N - 1$ lattice modules in Fig. 2 and a special case module (*SC1*). The expanding circuit consists of N multiplexers and $N - 1$ adders. The multiplexers bypass MDCT outputs when $x_r(n)$ are used as inputs

and bypass MDST outputs when $x_i(n)$ are used as inputs. Then, based on Eq. (5), the MDCT outputs and the MDST outputs are expanded to $2N$ outputs. The R -registers are used to store intermediate results. From Fig. 3, we can see that the architecture is modular, regular, and without global communications.

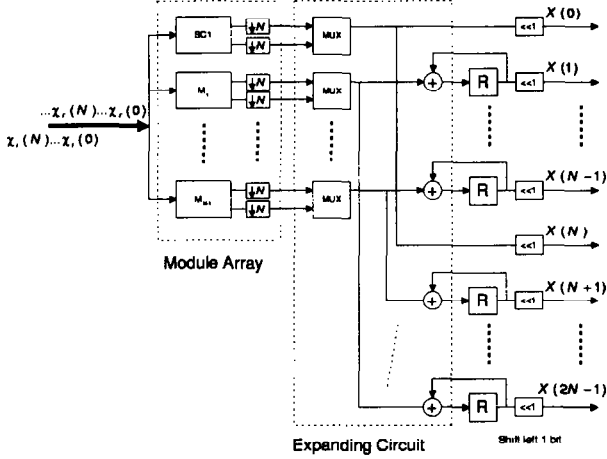


Figure 3: Overall IFFT architecture based on the lattice modules, where $\downarrow N$ denotes that we pick up the MDCT/MDST results at the time $x_r(N)$ and $x_i(N)$ finish the updating in Module Array.

The operation of the IFFT architecture is as follows: First, $x_r(n)$, $n = 0, 1, \dots, N-1$, are fed into the module array to compute the MDCT results, and the MDCT results are stored in the R -registers. Second, $x_i(n)$, $n = 0, 1, \dots, N-1$, are used as the inputs of the module array to compute the MDST results. Third, we combine the MDCT and MDST results to obtain the IFFT data according to (4). Then the scaling operation is achieved by shifting left by 1 bit.

3. FFT MODULE

3.1. The FFT Derivation

At the receiver side (see Fig. 1), the 512-point FFT is used to demodulate the received signals, which is given by

$$\tilde{x}(n) = \frac{1}{2N} \sum_{k=0}^{2N-1} \tilde{X}(k) W_{2N}^{-nk}, \quad (18)$$

for $n = 0, 1, \dots, 2N-1$. Note that $\tilde{X}(k)$, $k = 0, 1, \dots, 2N-1$, are real-valued numbers. Hence, (18) can be rewritten as

$$\begin{aligned} \tilde{x}(n) &= \frac{1}{2N} \left[\sum_{k=0}^{2N-1} \tilde{X}(k) \cos \frac{2\pi nk}{2N} + j \sum_{k=0}^{2N-1} \tilde{X}(k) \sin \frac{2\pi nk}{2N} \right] \\ &= MDCT(n) + j MDST(n), \end{aligned} \quad (19)$$

for $n = 0, 1, \dots, 2N-1$. Eq. (19) shows that the computation of the FFT is decomposed into a combination of an MDCT and an MDST. Both MDCT and MDST use $\tilde{X}(k)$, $k = 0, 1, \dots, 2N-1$, as the inputs. Hence, we can employ two real-valued kernels (MDCT and MDST) followed by a scaling operation of $\frac{1}{2N}$ 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 upper N -point outputs. Hence, we can neglect the outputs $\tilde{x}(n)$, $n = N, N+1, \dots, 2N-1$, which can save 50% of the complexity.

3.2. Parallel Lattice structure for the FFT

In this section, we derive the lattice structure for the FFT at the receiver side. In [6], [7], it has been shown that the lattice structure can perform the FFT for real-valued input data. The difference is that, in the DMT system, the input block size is $2N$ while we need only N FFT outputs. In what follows, we briefly derive the FFT lattice structure for completeness of this paper. First, we consider the MDCT. Following the derivation in Eqs. (6) - (10), the time-recursive equation of the MDCT becomes

$$\tilde{x}_r(n, t+1) = \tilde{x}_r(n, t+1) \cos \frac{\pi n}{N} + \tilde{x}_i(n, t+1) \sin \frac{\pi n}{N}. \quad (20)$$

Similarly, the MDST can be rewritten as

$$\tilde{x}_i(n, t+1) = \tilde{x}_i(n, t+1) \cos \frac{\pi n}{N} - \tilde{x}_r(n, t+1) \sin \frac{\pi n}{N}, \quad (21)$$

where

$$\tilde{x}_r(n, t+1) \triangleq \sum_{k=t+1}^{t+2N} \tilde{X}(k) \cos \frac{2\pi n(k-t)}{2N} \quad (22)$$

and

$$\tilde{x}_i(n, t+1) \triangleq \sum_{k=t+1}^{t+2N} \tilde{X}(k) \sin \frac{2\pi n(k-t)}{2N} \quad (23)$$

Similar to (11) - (12), we can find the updating equations for the $\tilde{x}_r(n, t+1)$ and $\tilde{x}_i(n, t+1)$ as

$$\tilde{\tilde{x}}_r(n, t+1) = \tilde{x}_r(n, t) + x_r(t+2N) \quad (24)$$

and

$$\tilde{\tilde{x}}_i(n, t+1) = \tilde{x}_i(n, t). \quad (25)$$

The complete time-recursive lattice module is shown in Fig. 4, which consists of 4 multipliers and 3 adders. Its operation is similar to the IFFT in Fig. 2 except that $\tilde{X}(k)$, $k = 0, 1, \dots, 2N-1$ are used as inputs. For $n = 0$, the FFT can be simplified as $\tilde{x}(0) = \sum_{k=0}^{2N-1} \tilde{X}(k)$. The special case is implemented using an accumulator and we denote it as $SC2$.

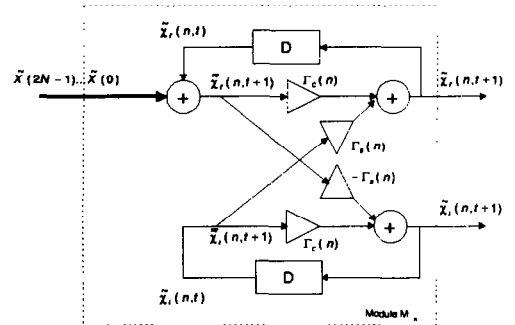


Figure 4: The lattice module (M_n , $n = 1, 2, \dots, N-1$) for the dual generation of the MDCT and MDST of the FFT, where $\Gamma_c(n) \triangleq \cos \frac{\pi n}{N}$, $\Gamma_s(n) \triangleq \sin \frac{\pi n}{N}$.

The complete FFT architecture based on lattice structure is shown in Fig. 5. The operations of the FFT architecture is as follows: First, $\tilde{X}(k)$, $k = 0, 1, \dots, 2N-1$, are fed into the array module to compute the MDCT and MDST outputs. Second, perform the scaling of $\frac{1}{2N}$ by shifting right by $\log_2(2N)$ bits. Third, combine the real part (MDCT outputs) and the imaginary part (MDST outputs) to form the complex-valued FFT results according to (19).

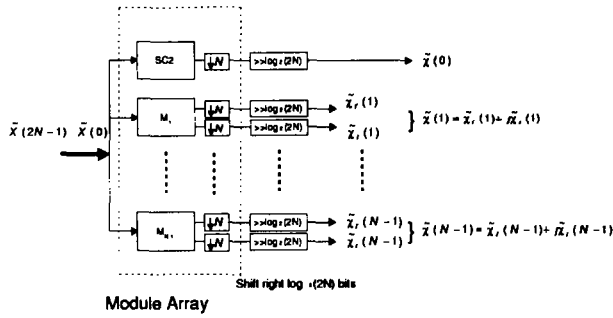


Figure 5: Overall FFT architecture based on the lattice modules.

4. COMPLEXITY ANALYSIS

In this section, we compare the hardware complexity of the proposed lattice structure with the direct implementation using butterfly approach [5]. In the direct implementation using butterfly approach, it consists $\log_2(2N)$ stages in the $2N$ -point IFFT/FFT. Each stage consists of N multipliers and $2N$ adders. Because input sequences are complex data, the IFFT/FFT kernels are complex in nature. Note that it requires 4 real valued multipliers and 2 real-valued adders for 1 complex multiplier. Also, it takes 2 real adders to realize a complex adder. Hence, the direct approach requires a total of $4N \log_2(2N)$ real multipliers and $6N \log_2(2N)$ real adders.

In our approach, we decompose the IFFT into the real-valued transform kernels (MDCT and MDST). As a result, we only need real operators to realize the IFFT structure in Fig. 3. There are one SC1 that consists of 2 adders, and $N - 1$ lattice modules (M_k). Each M_k consists of 4 multipliers and 3 adders. In addition, the overall architecture of the IFFT needs extra $2(N - 1)$ adders, N multiplexers, and $2(N - 1)$ registers for combining the MDCT and MDST outputs. Therefore, we need a total of $4(N - 1)$ real multipliers and $5(N - 1) + 2$ real adders for the implementation of the IFFT.

For the FFT module in Fig. 5, we need one SC2 which is an accumulator, and $(N - 1)$ lattice modules (M_n). The overall architecture of the FFT needs a total of $4(N - 1)$ real multipliers and $3(N - 1) + 1$ real adders to compute the MDCT (real part) and MDST (imaginary part) outputs.

The complexity comparison for $2N$ -point IFFT/FFT is listed in Table 1. The complexity ratio (CR) is defined as

$$CR \triangleq \frac{O_2}{O_1} \quad (26)$$

where O_1 and O_2 are the number of multipliers (or adders) in the butterfly approach and our approach, respectively. We can see that the complexity ratio of the multiplier is only 11% for the $N = 256$ case. Table 1 also shows that, our approach can gain more hardware savings compared with the direct implementation as N increases. This yields a very efficient solution for the applications of digital audio broadcasting (DAB) and digital TV terrestrial broadcasting, where the problem of larger block size N of the IFFT/FFT needs to be handled in the Orthogonal Frequency Division Multiplexing (OFDM) scheme [8]. In addition to the reduction of adders and multipliers, our approach also has lots of advantages in VLSI implementation. We illustrate these major features by comparing with the butterfly approach. The architectural comparison is listed in Table 2.

Table 1: Hardware comparison for $2N$ -point IFFT/FFT.

N	IFFT			FFT		
	Butterfly (O_1)	Our (O_2)	CR	Butterfly (O_1)	Our (O_2)	CR
256	9216	1020	0.111	9216	1020	0.111
512	20480	2044	0.100	20480	2044	0.100
1024	45056	4092	0.091	45056	4092	0.091
2048	98304	8188	0.083	98304	8188	0.083
4096	212992	16380	0.077	212992	16380	0.077
8192	458752	32764	0.071	458752	32764	0.071

(a) Number of 2-input multipliers

N	IFFT			FFT		
	Butterfly (O_1)	Our (O_2)	CR	Butterfly (O_1)	Our (O_2)	CR
256	13824	1277	0.092	13824	766	0.055
512	30720	2557	0.083	30720	1534	0.050
1024	67584	5117	0.076	67584	3070	0.045
2048	147456	10237	0.069	147456	6142	0.042
4096	319488	20477	0.064	319488	12286	0.038
8192	688128	40957	0.060	688128	24574	0.036

(b) Number of 2-input adders

5. CONCLUSIONS

In this paper, we derived cost-effective VLSI architectures for the implementations of the IFFT/FFT kernel in the DMT system. We reformulate the IFFT/FFT functions so as to avoid complex-domain operations. We also employ the time-recursive lattice structures to realize the IFFT/FFT in $O(N)$ hardware complexity. The complexity ratio of the multipliers and adders is only 11% and 9%, respectively, compared with the direct implementation approach. The proposed architecture provides a good solution for the cost-efficient implementation of larger block size IFFT/FFT in the DMT/OFDM transceiver system.

Table 2: Architectural comparison.

	Butterfly	Our approach
I/O Operation	PIPO	SIPO
Architecture	Modular	Modular and Regular
Communication	Global	Local
Hardware cost	Large	Small
Latency	$\log_2(2N)$	$2N$

References

- [1] 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, pp. 620-632, May 1995.
- [2] 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, pp. 895-908, Aug. 1991.
- [3] K. Sistanizadeh, P. Chow, and J. M. Cioffi, "Multi-tone transmission for asymmetric digital subscriber lines (ADSL)," in *Proc. ICC'93*, pp. II. 756-760, 1993.
- [4] B. Daneshmand and H. Samuelli, "A 1.6 Mbps digital-QAM system for DSL transmission," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 1600-1610, Dec. 1995.
- [5] A. V. Oppenheim and R. W. Schaffer, "Discrete-Time Signal Processing," Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [6] K. J. Ray Liu, C. T. Chiu, R. K. Kolagotla, and J. F. J'aj'a, "Optimal unified architectures for the real-time computation of time-recursive discrete sinusoidal transforms," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 4, no. 2, pp. 168-180, April 1994.
- [7] A.-Y. Wu and K. J. R. Liu, "Algorithm-based low-power transform coding architectures," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, (Detroit), pp. 3267-3270, May 1995.
- [8] W. Y. Zou and Y. Wu, "COFDM: An Overview," *IEEE Trans. on Broadcasting*, vol. 41, no. 1, pp. 1-8, Mar. 1995.