MULTIPLIERLESS LIFTING BASED FFT VIA FAST HARTLEY TRANSFORM

Taizo Suzuki[1]*, Seisuke Kyochi*[2]*, Yuichi Tanaka*[3]*, Masaaki Ikehara*[4]*, and Hirotomo Aso*[5]

[1]Faculty of EIS, Univ. of Tsukuba, [2]IME Dept., The Univ. of Kitakyushu,
[3]Graduate School of BASE, TUAT, [4]EEE Dept., Keio Univ., [5]EEE Dept., Nihon Univ., Japan
Email: [1]taizo@cs.tsukuba.ac.jp, [2]s-kyochi@kitakyu-u.ac.jp,
[3]ytnk@cc.tuat.ac.jp, [4]ikehara@tkhm.elec.keio.ac.jp, [5]aso@ee.ce.nihon-u.ac.jp

ABSTRACT

The multiplierless fast Fourier transform (FFT) with dyadic-valued (rational) coefficients is important for many signal processing tools. The proposed lifting based FFT (L-FFT) based on fast Hartley transform (FHT) has a simpler structure than existing ones because fewer lifting steps need to be approximated. In addition, it has a structure of real-valued calculation followed by complex-valued parts, thereby it requires fewer memories for the internal implementation than the conventional FFTs.

Index Terms— Dyadic-valued coefficient, fast Fourier transform (FFT), fast Hartley transform (FHT), lifting structure

1. INTRODUCTION

The fast Fourier transform (FFT) [1], which is a fast algorithm of discrete Fourier transform (DFT) [2], is one of the most fundamental and common tools in signal processing [3–5]. Generally, the FFT consists of floating-point coefficients and yields floating-point output signals even if the input signals are integer-valued. In this calculation, floating point multipliers waste computational cost and power consumption which cannot be neglected. To efficiently implement the FFT, floating-point coefficients are often approximated to dyadic-valued (rational) coefficients. It is the most important issue to reduce its power consumption. Ideally, such approximation should be carried out without losing perfect reconstruction (PR) or near PR property.

Lifting structure [6] is a key technology to efficiently approximate floating-point coefficients by dyadic-valued coefficients while keeping near PR [7]. Oraintara et al. introduced the lifting factorization of a traditional (split-radix based) FFT and realized a multiplierless lifting based FFT (L-FFT) with dyadic-valued coefficients [8]. However, the structure does NOT consider minimizing the number of lifting steps for fewer adders and bit shifters, and thus it has redundant operations. Moreover, it requires many memories for the internal implementation because there are many complex-valued operations everywhere.

In this paper, we propose a multiplierless L-FFT based on the fast Hartley transform (FHT) [9] for one and two dimensional (1-D and 2-D) signals. The FHT is a fast algorithm of discrete Hartley transform (DHT) [10] which is a DFT like real-valued transform. The conventional and proposed multiplierless FFTs are named sprit-radix based L-FFT (L-‘S’FFT) and FHT based L-FFT (L-‘H’FFT), respectively, to distinguish them throughout this paper. The L-HFFT, which has a different structure from the traditional radix based FFTs, has a simpler structure than that of the L-SFFT. Specifically, it can reduce the number of lifting steps which need to be approximated. In addition, since it has a structure of real-valued calculation followed by complex-valued parts, it requires fewer memories for the internal implementation than the conventional ones. In the simulations, we evaluate the L-HFFT and numerically show its comparable performance to the L-SFFT in spite of fewer operations.

Notations: $\mathbf{I}$, $\mathbf{J}$, $\mathbf{M}^T$, $\mathbf{M}^\dagger$, $\mathbf{M}_N$ and j are an identity matrix, a reversal identity matrix, transpose of a matrix $\mathbf{M}$, conjugate transpose of a matrix $\mathbf{M}$, a matrix $\mathbf{M}$ with $N \times N$ size and an imaginary number defined as $j^2 = -1$, respectively.

2. REVIEW

2.1. DFT and DHT

The (m, n)-element of M-channel DFT matrix $\mathcal{F}_M$ is defined as [2]

$$[\mathcal{F}_M]_{m,n} = \frac{1}{\sqrt{M}} \exp\left(\frac{-j2mn\pi}{M}\right)$$

where $0 \leq m, n \leq M-1$. The inverse DFT (IDFT) is the conjugate transpose of the DFT matrix, i.e., $\mathcal{F}_M^{-1} = \mathcal{F}_M^\dagger$. For simplicity, let M be defined as $M = 2^k$ ($k \in \mathbb{N}$).

The DHT [10] is a DFT like real-valued transform. The (m, n)-element of M-channel DHT matrix $\mathcal{H}_M$, which is the symmetric orthogonal matrix as $\mathcal{H}_M^{-1} = \mathcal{H}_M^T = \mathcal{H}_M$, is defined as

$$[\mathcal{H}_M]_{m,n} = \frac{1}{\sqrt{M}} \left(\cos\left(\frac{2mn\pi}{M}\right) + \sin\left(\frac{2mn\pi}{M}\right)\right). \quad (1)$$

These DFT and DHT have many fast algorithms, i.e., FFT [1] and FHT [9], respectively, which are usually used in practical implementation.

2.2. Multiplierless Lifting Structure

The lifting structure [6], also known as the ladder structure, is a special type of lattice structure. It is implemented by cascading elementary matrices — identity matrices with a single nonzero off-diagonal element.

Fig. 1 shows a basic lifting step. It is expressed by

$$y_j(n) = x_j(n),\ y_i(n) = x_i(n) + \mathrm{round}\{Tx_j(n)\}$$
$$z_j(n) = y_j(n),\ z_i(n) = y_i(n) - \mathrm{round}\{Ty_j(n)\}$$

where $\mathrm{round}\{\cdot\}$ and T are a rounding operation and a lifting coefficient, respectively. Thus the lifting structure with the rounding

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

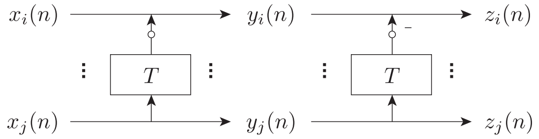

Fig. 1. A basic lifting step (white circles: rounding operations)

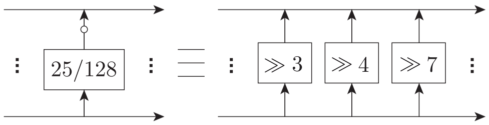

Fig. 2. An approximation from multiplier with dyadic-valued coefficients to bit-shifter and adder in a lifting structure ($\gg \beta$ means β bit-shifter).

operation can achieve integer-to-integer transform. Also, the lifting and its inverse matrices in this case are represented as

$$\begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix} \quad \text{and} \quad \begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix}^{-1} = \begin{bmatrix}1 & -T\\ 0 & 1\end{bmatrix},$$

respectively.

For a high-speed and circuit-size saving implementation, lifting coefficients are required to approximate floating-point coefficients to software/hardware-friendly dyadic-valued coefficients such as $\alpha/2^{\beta}$ ($\alpha, \beta \in \mathbb{N}$) which can be implemented by only adders and bit-shifters [7]. The dyadic-valued representation enables to perform fast implementation in a software and reduce the circuit size. For example, a coefficient $25/128$ can be operated as

$$\frac{25}{128} = \frac{2^4 + 2^3 + 2^0}{2^7} = \frac{1}{2^3} + \frac{1}{2^4} + \frac{1}{2^7}.$$

Hence, the lifting with its coefficient $25/128$ and a rounding operation is replaced to the summation of 3, 4 and 7 bit-shifters illustrated in Fig. 2. It is clear that the PR in lifting is always kept even if floating-point coefficients are approximated to dyadic-valued coefficients. When a floating-point coefficient is allocated β bit, the dyadic-valued coefficient is presented as $\alpha/2^{\beta}$.

3. MULTIPLIERLESS L-FFT VIA FHT (L-HFFT)

In this section, we derive a realization of multiplierless FFT for various signal processing tools by the following procedures.

1. An FFT factorization is derived based on FHT.
2. Several obtained rotation matrices are approximated to multiplierless lifting structures, and the scaling factors are moved to the last part.

(For 2-D signals, e.g., images)

3. It is completed by using 2-D separable block transform.

3.1. FFT based on FHT

FHT in (1) is simplified by rotation matrices as the following recursive algorithm and its 16-channel case is shown in Fig. 3.

$$\hat{\mathcal{H}}_M = \begin{bmatrix}\hat{\mathcal{H}}_{M/2} & \mathbf{0}\\ \mathbf{0} & \hat{\mathcal{H}}_{M/2}\end{bmatrix}\mathbf{R}_M\mathbf{W}_M$$

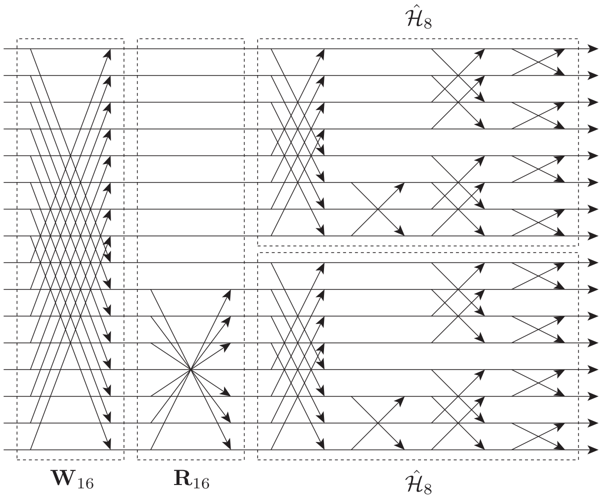

Fig. 3. The 16-channel FHT $\hat{\mathcal{H}}_{16}$

where $\hat{\mathcal{H}}_M$ is an FHT permutated frequency band order of $\mathcal{H}_M$, i.e., $\mathcal{H}_M = \mathbf{P}_M^T\hat{\mathcal{H}}_M = \hat{\mathcal{H}}_M^T\mathbf{P}_M$ where $\mathbf{P}_M$ is a permutation matrix. Also, $\mathbf{R}_M$ and $\mathbf{W}_M$ are expressed as

$$\mathbf{R}_M = \begin{bmatrix}\mathbf{I}_{M_1} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{C}_{M_2} & \mathbf{0} & \mathbf{J}_{M_2}\mathbf{S}_{M_2}\\ \mathbf{0} & \mathbf{0} & 1 & \mathbf{0}\\ \mathbf{0} & \mathbf{S}_{M_2}\mathbf{J}_{M_2} & \mathbf{0} & -\mathbf{J}_{M_2}\mathbf{C}_{M_2}\mathbf{J}_{M_2}\end{bmatrix}$$

for $M_1 = M/2 + 1$ and $M_2 = M/4 - 1$, and

$$\mathbf{W}_M = \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{I}_{M/2} & \mathbf{I}_{M/2}\\ \mathbf{I}_{M/2} & -\mathbf{I}_{M/2}\end{bmatrix},$$

respectively. The (l, l)-elements of $\mathbf{C}_{M_2}$ and $\mathbf{S}_{M_2}$ are presented by

$$[\mathbf{C}_{M_2}]_{l,l} = \cos\left(\frac{2(l+1)\pi}{M}\right) \quad \text{and} \quad [\mathbf{S}_{M_2}]_{l,l} = \sin\left(\frac{2(l+1)\pi}{M}\right),$$

respectively, for $0 \leq l \leq M/4 - 2$.

A representation of M-channel FFT $\mathcal{F}_M$ is constructed by only one FHT $\mathcal{H}_M$ and the matrix $\mathbf{G}_M$ with real- and complex-valued coefficients as follows [9]:

$$\mathcal{F}_M = \mathbf{Q}_M\mathbf{G}_M\mathcal{H}_M = \mathbf{Q}_M\mathbf{G}_M\hat{\mathcal{H}}_M^T\mathbf{P}_M \tag{2}$$

where

$$\mathbf{G}_M = \begin{bmatrix}1 & \mathbf{0} & 0 & \mathbf{0}\\ \mathbf{0} & \frac{1-j}{2}\mathbf{I}_{M/2-1} & \mathbf{0} & \frac{1+j}{2}\mathbf{J}_{M/2-1}\\ \mathbf{0} & \mathbf{0} & 1 & \mathbf{0}\\ \mathbf{0} & \frac{1+j}{2}\mathbf{J}_{M/2-1} & \mathbf{0} & \frac{1-j}{2}\mathbf{I}_{M/2-1}\end{bmatrix}$$

and $\mathbf{Q}_M$ is a permutation matrix. Also, the inverse of $\mathbf{G}_M$ is presented by the conjugate transpose of $\mathbf{G}_M$, i.e., $\mathbf{G}_M^{-1} = \mathbf{G}_M^{\dagger}$.

3.2. Multiplierless Lifting Factorization and Moving Scaling Factors to The Last Part

First, the rotation matrices in $\mathbf{R}_M$ and the complex matrices in $\mathbf{G}_M$ are factorized into the lifting structures as

$$\begin{bmatrix}\cos\theta & \sin\theta\\ \sin\theta & -\cos\theta\end{bmatrix} = \begin{bmatrix}1 & 0\\ \frac{1-\cos\theta}{\sin\theta} & -1\end{bmatrix}\begin{bmatrix}1 & \sin\theta\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ \frac{\cos\theta-1}{\sin\theta} & 1\end{bmatrix}$$

and

$$\frac{1-j}{2}\begin{bmatrix}1 & j\\ j & 1\end{bmatrix} = \begin{bmatrix}1 & 0\\ 1 & -j\end{bmatrix}\begin{bmatrix}1 & \frac{1+j}{2}\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ -1 & 1\end{bmatrix},$$

respectively. Next, floating-point coefficients in the lifting rotation matrices in $\mathbf{R}_M$ are approximated to dyadic-valued coefficients. Note that the lifting steps in $\mathbf{G}_M$ already have only dyadic-valued coefficients.

On the other hand, the rotation matrices with the angles $\pi/4$ in FHT part $\hat{\mathcal{H}}_M^T$ except for $\mathbf{R}_M$ are presented by

$$\begin{bmatrix}\cos\frac{\pi}{4} & \sin\frac{\pi}{4}\\ \sin\frac{\pi}{4} & -\cos\frac{\pi}{4}\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}.$$

Since all of the scaling factors $1/\sqrt{2}$ can be moved to the last part of FFT, the matrices can be implemented by only ± 1. Let $\tilde{\mathcal{H}}_M$ be $\hat{\mathcal{H}}_M$ removed its scaling factors. Then, (2) is rewritten as

$$\mathcal{F}_M = \mathbf{Q}_M\mathbf{D}_M\mathbf{G}_M\tilde{\mathcal{H}}_M^T\mathbf{P}_M \tag{3}$$

where $\mathbf{D}_M$ is the scaling factor matrix. The resulting FFT is shown in Fig. 4.

3.3. Use of 2-D Separable Block Transform

Note that the structure obtained in the above subsection products the non-dyadic-valued (floating-point) scaling factor $1/\sqrt{2}$ when $\log_2 M$ is odd. Moreover, if the structure in Fig. 4 is directly applied to image, the complex-valued signals produced after the first dimensional (i.e., column or row) implementation will be input to the second dimensional (i.e., row or column) operation. Hence, the structure requires many memories for the internal implementation. To solve the problem, we change the order of the 2-D separable block transform [7] for the proposed FFT. When we apply a block transform matrix $\mathcal{F}_M$ into a 2-D input signal $\mathbf{x}$ in column- and row-wise separately, the 2-D output signal $\mathbf{y}$ is expressed by

$$\mathbf{y} = (\mathcal{F}_M(\mathcal{F}_M\mathbf{x})^T)^T = \mathcal{F}_M\mathbf{x}\mathcal{F}_M^T \tag{4}$$

Since $\mathcal{F}_M$ is factorized as $\mathbf{D}_M\mathbf{G}_M\tilde{\mathcal{H}}_M^T\mathbf{P}_M$ as shown in (3), (4) is represented by

$$\mathbf{y} = \mathbf{Q}_M\mathbf{D}_M\mathbf{G}_M\tilde{\mathcal{H}}_M^T\mathbf{P}_M\mathbf{x}\mathbf{P}_M^T\tilde{\mathcal{H}}_M\mathbf{G}_M^T\mathbf{D}_M^T\mathbf{Q}_M^T.$$

This equation means that the 2-D block transform by $\hat{\mathcal{H}}_M^T$ is firstly applied after the 2-D block transform by $\mathbf{P}_M$, the 2-D block transform by $\mathbf{G}_M$ is secondly applied, the 2-D block transform by $\mathbf{D}_M$ is thirdly applied, and the 2-D block transform by $\mathbf{Q}_M$ is finally applied. Therefore, when $\log_2 M$ is odd, the non-dyadic-valued scaling factor $1/\sqrt{2}$ in 1-D transform can be implemented as the dyadic-valued scaling factor $1/2$ in 2-D transform. Moreover, it is clear that the FFT requires fewer memories for the internal implementation because it can be completely separated to real- and complex-valued parts.

Table 1. The FFTs for their transform performance comparison (The split-radix based FFT with dyadic-valued coefficients without lifting factorization is named 'S'FFT).

Name	Basement	Lifting	Dyadic-Value
(ideal)	Split-Radix	NA	NA
SFFT	Split-Radix	NA	✓
L-SFFT	Split-Radix	✓	✓
L-HFFT	FHT	✓	✓

Table 2. The numbers of operations in FFTs with 8 bit word length allocated coefficients (Lift., Add. and Shift. mean the numbers of lifting steps which require finite word length allocation, adders and bit-shifters, respectively).

Channel	L-SFFT			L-HFFT		
M	Lift.	Add.	Shift.	Lift.	Add.	Shift.
16	24	178	106	15	151	73
32	78	518	332	51	424	234
64	216	1366	910	147	1098	652
128	486	3356	2298	387	2689	1667
256	1296	8114	5634	963	6395	4093
512	2430	18710	13292	2307	14836	9718
1024	6552	43116	30692	5379	33765	22503

*For simplicity, the scaling parts are NOT included in this table.

4. EXPERIMENTAL RESULTS

The FFTs for their transform performance comparison are shown in Table 1. In Sec. 4.2 and 4.3, we used 256-channel FFTs and 256×256 8-bit grayscale images *Camera* and *Girl* as test images.

4.1. Number of Operations in FFTs

Table 2 shows the number of lifting steps which must be approximated, adders and bit-shifters in the L-SFFT and the L-HFFT approximated by 8 bit allocation. It is clear that the L-HFFT has not only no multiplier but fewer adders and bit-shifters than the L-SFFT. It is no wonder that the L-HFFT is constructed by fewer lifting steps than the L-SFFT. Let the numbers of operations in M-channel L-SFFT and L-HFFT be defined as S_M and H_M, respectively. They are recursively determined as follows:

$$\begin{aligned} S_M &= \frac{3(M-4)}{2} + S_{\frac{M}{2}} + 2S_{\frac{M}{4}} \\ H_M &= \frac{3(M-4)}{4} + 2H_{\frac{M}{2}} \end{aligned}$$

where $S_2 = S_4 = H_2 = H_4 = 0$.

4.2. Accuracy of FFT

The images *Camera* and *Girl*, which are reconstructed by the ideal inverse FFT (IFFT) after they are transformed by each of SFFT, L-SFFT and L-HFFT, are compared with peak-to-noise ratio (PSNR) as

$$\text{PSNR[dB]} = 10\log_{10}\left(\frac{255^2}{\text{MSE}}\right)$$

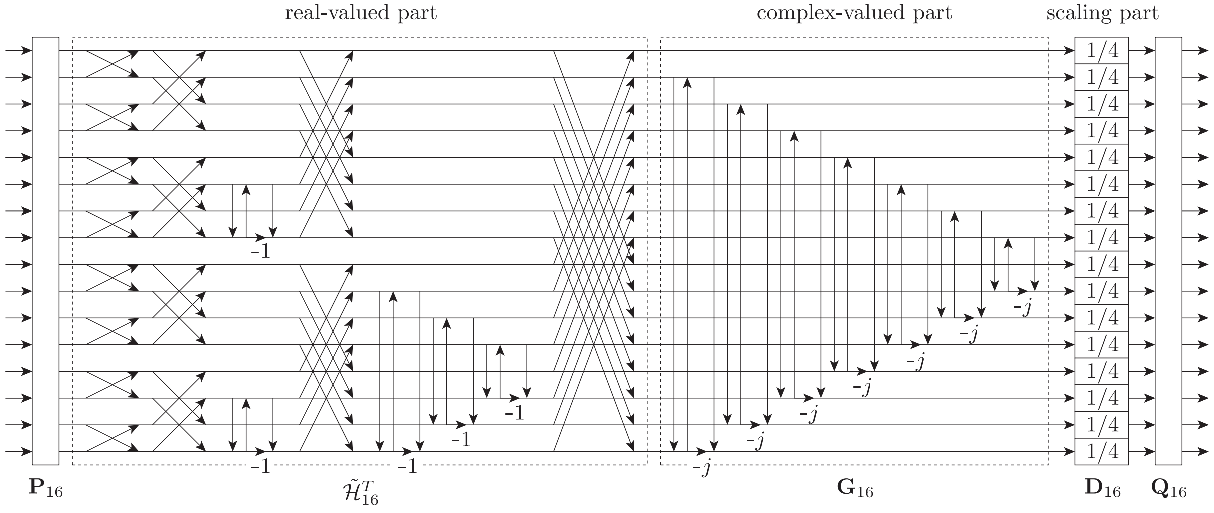

Fig. 4. The 16-channel L-HFFT

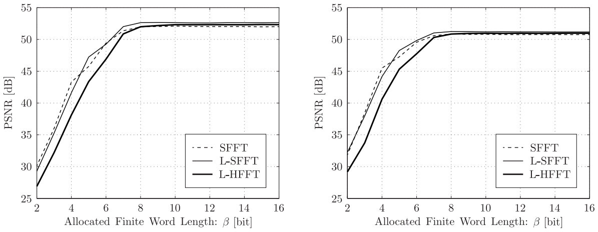

Fig. 5. F-D curves of the images reconstructed by ideal IFFT after they are transformed by each of SFFT, L-SFFT and L-HFFT: (left) *Camera*, (right) *Girl*.

Table 3. The comparison examples in the images reconstructed by ideal IFFT after they are transformed by each of L-SFFT and L-HFFT when $M = 256$.

	Numbers		PSNR[dB]	
	Adders	Shifters	*Camera*	*Girl*
L-SFFT (4 bit alloc.)	5,138	3,090	41.69	44.58
L-HFFT (5 bit alloc.)	**5,031**	**2,983**	**43.64**	**46.10**

where MSE is the mean squared error. This simulation clarifies an approximation accuracy of the dyadic-valued FFTs to the ideal FFT, under the rounding operation. Fig. 5 shows F-D (finite word length —distortion) curves at the case. Although the L-HFFT seemingly shows lower PSNRs, the L-HFFT shows comparable performance to the L-SFFT in spite of fewer operations in practice as shown in Table 3. Also, the L-HFFT requires fewer memories for the internal implementation than the L-SFFT due to real-valued calculation followed by complex-valued parts.

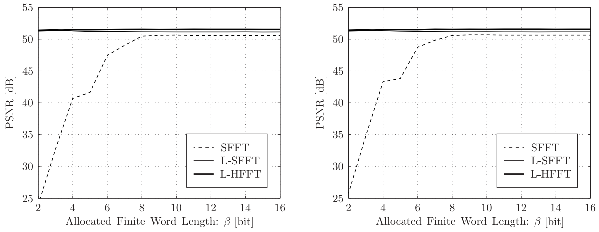

Fig. 6. F-D curves of the images reconstructed by each of the inverse transforms of SFFT, L-SFFT and L-HFFT after they are transformed by each of SFFT, L-SFFT and L-HFFT: (left) *Camera*, (right) *Girl*

4.3. Accuracy of Near PR

Test images *Camera* and *Girl*, which are reconstructed by each of the inverse transforms of SFFT, L-SFFT and L-HFFT after they are transformed by each of SFFT, L-SFFT and L-HFFT, are compared with PSNR. This simulation clarifies an accuracy of reconstruction of each dyadic-valued FFT because the complete PR is lost due to error generated by bit-shifts in the last scaling part. Fig. 6 shows F-D curves at the case. Although the SFFT degrades its PSNRs in lower bit allocation, such degrading is NOT shown in the L-SFFT and the L-HFFT. Hence, the FFTs with lifting structures always preserve near PR.

5. CONCLUSION

This paper has proposed a realization of multiplierless fast Fourier transform (FFT). It is composed of fast Hartley transform (FHT) and multiplierless lifting structures, and has only adders and bit-shifters, i.e., no multipliers. Moreover, it has fewer operations and requires fewer memories for the internal implementation than the conventional FFTs. In spite of such facts, the proposed FFT verified its comparable performance to the conventional ones in the simulations.

6. REFERENCES

[1] J. W. Cooley and J. W. Tukey, *An algorithm for the machine computation of complex Fourier series*, Math. Comput., 1965.

[2] N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*, Berlin, Germany: Springer-Verlag, 1975.

[3] R. Hardie, "A fast image super-resolution algorithm using an adaptive Wiener filter," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2953–2964, Dec. 2007.

[4] M. Ghazal, A. Amer, and A. Ghrayeb, "Structure-oriented multidirectional Wiener filter for denoising of image and video signals," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 12, pp. 1797–1802, Dec. 2008.

[5] D. Humphrey and D. Taubman, "A filtering approach to edge preserving MAP estimation of images," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1234–1248, May 2011.

[6] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, vol. 3, no. 2, pp. 186–200, Apr. 1996.

[7] T. Suzuki and M. Ikehara, "Integer discrete cosine transform via lossless Walsh-Hadamard transform with structural regularity for low-bit-word-length," *IEICE Trans. Fundamentals*, vol. E93-A, no. 4, pp. 734–741, Apr. 2010.

[8] S. Oraintara, Y. J. Chen, and T. Q. Nguyen, "Integer fast Fourier transform," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 607–618, Mar. 2002.

[9] R. N. Bracewel, "The fast Hartley transform," *Proc. of the IEEE*, vol. 72, no. 8, pp. 1010–1018, Aug. 1984.

[10] R. N. Bracewel, "Discrete Hartley transform," *J. Opt. Soc. Am.*, vol. 73, no. 12, pp. 1832–1835, Dec. 1983.