Cost-Effective Triple-Mode Reconfigurable Pipeline FFT/IFFT/2-D DCT Processor

Chin-Teng Lin, *Fellow, IEEE*, Yuan-Chu Yu, *Member, IEEE*, and Lan-Da Van, *Member, IEEE*

***Abstract*—This investigation proposes a novel radix-4^2 algorithm with the low computational complexity of a radix-16 algorithm but the lower hardware requirement of a radix-4 algorithm. The proposed pipeline radix-4^2 single delay feedback path ($\mathrm{R4^2SDF}$) architecture adopts a multiplierless radix-4 butterfly structure, based on the specific linear mapping of common factor algorithm (CFA), to support both 256-point fast Fourier transform/inverse fast Fourier transform (FFT/IFFT) and 8 × 8 2-D discrete cosine transform (DCT) modes following with the high efficient feedback shift registers architecture. The segment shift register (SSR) and overturn shift register (OSR) structure are adopted to minimize the register cost for the input re-ordering and post computation operations in the 8 × 8 2-D DCT mode, respectively. Moreover, the retrenched constant multiplier and eight-folded complex multiplier structures are adopted to decrease the multiplier cost and the coefficient ROM size with the complex conjugate symmetry rule and subexpression elimination technology. To further decrease the chip cost, a finite wordlength analysis is provided to indicate that the proposed architecture only requires a 13-bit internal wordlength to achieve 40-dB signal-to-noise ratio (SNR) performance in 256-point FFT/IFFT modes and high digital video (DV) compression quality in 8 × 8 2-D DCT mode. The comprehensive comparison results indicate that the proposed cost effective reconfigurable design has the smallest hardware requirement and largest hardware utilization among the tested architectures for the FFT/IFFT computation, and thus has the highest cost efficiency. The derivation and chip implementation results show that the proposed pipeline 256-point FFT/IFFT/2-D DCT triple-mode chip consumes 22.37 mW at 100 MHz at 1.2-V supply voltage in TSMC 0.13-μm CMOS process, which is very appropriate for the RSoCs IP of next-generation handheld devices.**

***Index Terms*—Computation complexity, cost effective, hardware utilization, next-generation wireless communications, pipeline architecture, $\mathrm{R4^2SDF}$, triple modes.**

I. Introduction

NEXT-GENERATION mobile multimedia applications, including mobile phones and personal digital assistants (PDAs), require much sufficiently high processing power for multimedia applications. Multimedia applications include video/audio codecs, speech recognition, and echo cancellers. The speech recognition requires the speech extraction and autocorrelation coefficient computations [2] in the voice command application. The video codec is the most challenging element of a multimedia application, since it requires much processing power and bandwidth. Hence, a flexible and low cost pipeline processor with the superiority of high processing rate is required to realize necessary computation-intensive algorithms, such as 256-point fast Fourier transform/inverse fast Fourier transform (FFT/IFFT) and 8 × 8 2-D discrete cosine transform (DCT) [3]–[5]. Additionally, a major integration challenge is to design the digital baseband and accompanying control logic. The WiMAX baseband is constructed around orthogonal frequency division multiplexing (OFDM) technology requiring high processing throughput. The fixed, IEEE 802.16e [1], version of WiMAX also needs a 256-point FFT computation. Many researchers have recently concentrated on designing an optimized reconfigurable DSP processor to achieve a high processing rate and low power consumption in next-generation mobile multimedia applications [3], [4]. The software-based architecture such as the coprocessor and dual-MAC designs have been proposed by Chai *et al.* [3] and Kolagotla *et al.* [4], respectively. However, they induce the large chip size because of the high flexibility. Vorbach *et al.* have also presented hardware-based concepts such as the processing element (PE) array [5], which achieves a high processing rate with reasonable flexibility. However, the processing kernel has the flaw of a low utilization rate with a large array memory and muti-MACs, leading to poor cost efficiency. The specific application-specific integrated circuit (ASIC)-based design on a fast computation algorithm provides high cost efficiency [6]–[8]. Tell *et al.* [6] presented the FFT/WALSH/1-D DCT processor for multiple radio standards of the upcoming fourth generation wireless systems. Conversely, some designs [6]–[8] only support 1-D DCT computation, and have no 2-D DCT support. However, 2-D DCT is desirable for the video compression among wireless communication applications. This study not only presents a single reconfigurable architecture for the 256-point FFT/IFFT modes and the 8 × 8 2-D DCT mode, but also achieves high cost-efficiency in portable multimedia applications.

He *et al.* [9] has presented several reliable architectures and the detailed comparisons of the corresponding hardware cost for efficient pipeline FFT processors. The comparison results of these architectures indicate that the $\mathrm{R2^2SDF}$ has the highest butterfly utilization and lowest hardware resource usage in the pipeline FFT/IFFT architecture. However, the radix-2^2

Manuscript received March 16, 2007; revised July 23, 2007. This work was supported in part by the National Science Council of the Republic of China, Taiwan under Contract NSC93-2218-E-009-061 and Contract MOEA-96-EC-17-A-01–S1-048.

C.-T. Lin is with the University Provost and the Chair Professor of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: ctlin@mail.nctu.edu.tw).

Y.-C. Yu is with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: Vincent_yu@emc.com.tw).

L.-D. Van is with the Department of Computer Science, National Chiao-Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: ldvan@cs.nctu.edu.tw).

Digital Object Identifier 10.1109/TVLSI.2008.2000676

algorithm has a higher multiplicative complexity than the high-radix FFT and other mixed-radix algorithms. Results of comprehensive comparison further indicate that the proposed R4^{2}SDF-based pipeline processor achieves a higher utilization with a smaller hardware requirement than R2^{2}SDF-based pipeline processor [9] in the 256-point FFT/IFFT mode, and thus has higher cost efficiency. The proposed R4^{2}SDF-based design also achieves satisfactory performance for the DV encoding standard with the lowest cost in the 8 × 8 2-D DCT mode. Finally, the chip implementation has been applied in the TSMC 0.13-μm 1P8M CMOS process, which only consumes 22.37 mW under 1.2 V at 100 MHz. Thus, the proposed design is very appropriate for reconfigurable system-on-chips (RSoCs) IP in next-generation mobile multimedia applications. The remainder of this study is structured as follows. A new R4^{2}SDF FFT/IFFT and 8 × 8 2-D DCT algorithm is given in Section II. Section III demonstrates the proposed FFT/IFFT/2-D DCT pipeline architecture using the R4^{2}SDF algorithm. The finite wordlength analysis is given in Section IV, and indicates that the proposed architecture achieves the required system performance in both 256-point FFT/IFFT and 8 × 8 2-D DCT modes with the lowest hardwire cost. Section V tabulates the comparison results in terms of hardware utilization and cost to demonstrate the high cost-efficiency of the proposed architecture, and also discusses the chip implementation. The final section draws conclusions.

II. New Radix-4^2-Based FFT/IFFT and 8 × 8 2-D DCT Algorithm

A. Radix-4^2-Based FFT Formula

The FFT of the N-point input $x[n]$ is given by

$$X[k]=\sum_{n=0}^{N-1}x[n]\cdot W_N^{kn} \tag{1}$$

where $W_N=e^{-j2\pi/N}$. Applying a 3-D linear index map

$$\begin{aligned} n&=\frac{N}{4}n_1+\frac{N}{16}n_2+n_3\\ k&=k_1+4k_2+16k_3. \end{aligned} \tag{2}$$

The common factor algorithm (CFA) [10] form can be written as

$$\begin{aligned} &X[k_1+4k_2+16k_3]\\ &\quad=\sum_{n_3=0}^{\frac{N}{16}-1}\sum_{n_2=0}^{3}\sum_{n_1=0}^{3}x\left(\frac{N}{4}n_1+\frac{N}{16}n_2+n_3\right)\\ &\qquad\times W_N^{\left(\frac{N}{4}n_1+\frac{N}{16}n_2+n_3\right)(k_1+4k_2+16k_3)}\\ &\quad=\left\{\sum_{n_3=0}^{\frac{N}{16}-1}\left\{\sum_{n_2=0}^{3}B_{\frac{N}{4}}^{k_1}\left(\frac{N}{16}n_2+n_3\right)W_N^{\frac{N}{16}n_2(k_1+4k_2)}\right\}\right.\\ &\qquad\left.\times W_N^{n_3(k_1+4k_2)}\right\}W_{\frac{N}{16}}^{n_3k_3} \end{aligned} \tag{3}$$

where the butterfly structure of the first stage takes the form

$$\begin{aligned} B_{\frac{N}{4}}^{k_1}\left(\frac{N}{16}n_2+n_3\right)&=x\left(\frac{N}{16}n_2+n_3\right)\\ &\quad+(-j)^{k_1}x\left(\frac{N}{16}n_2+n_3+\frac{N}{4}\right)\\ &\quad+(-1)^{k_1}x\left(\frac{N}{16}n_2+n_3+\frac{2N}{4}\right)\\ &\quad+(j)^{k_1}x\left(\frac{N}{16}n_2+n_3+\frac{3N}{4}\right). \end{aligned} \tag{4}$$

Notably, the radix-4 butterfly structure only requires trivial multiplication, which involves real-imaginary swapping and sign inversion, and which does not require any complex multiplication of the type shown in (4). The structure has only four four-input complex adders and some shuffle circuits. Decomposing the FFT computation with a higher radix, induces increasingly complex multiplications in the single butterfly structure. The radix-4 based is more cost-efficient than higher-radix-based butterfly structures. Following a similar decomposition procedure, (3) can be decomposed as

$$X[k_1+4k_2+16k_3]=\left\{\sum_{n_3=0}^{\frac{N}{16}-1}B_{\frac{N}{16}}^{k_1,k_2}(n_3)W_N^{n_3(k_1+4k_2)}\right\}W_{\frac{N}{16}}^{n_3k_3}. \tag{5}$$

Meanwhile, the butterfly structure of the second stage can be obtained as

$$\begin{aligned} B_{\frac{N}{16}}^{k_1,k_2}(n_3)&=B_{\frac{N}{4}}^{k_1}(n_3)+W_{16}^{k_1}\left[(-j)^{k_2}B_{\frac{N}{4}}^{k_1}\left(n_3+\frac{N}{16}\right)\right]\\ &\quad+W_{16}^{2k_1}\left[(-1)^{k_2}B_{\frac{N}{4}}^{k_1}\left(n_3+\frac{2N}{16}\right)\right]\\ &\quad+W_{16}^{3k_1}\left[(j)^{k_2}B_{\frac{N}{4}}^{k_1}\left(n_3+\frac{3N}{16}\right)\right]. \end{aligned} \tag{6}$$

Clearly, the decomposition creates three multipliers: $W_{16}^{k_1}$, $W_{16}^{k_2}$, and $W_{16}^{k_3}$, as written in (6). The three full complex multipliers from the second butterfly stage can be simplified as one single constant multiplier in the proposed R4^{2}SDF architecture. The constant multiplier cost can be further reduced by applying the subexpression elimination algorithm. The detailed hardware structure is described in Section III. The second radix-4 butterfly structure in (6) is the same as the first radix-4 butterfly structure in (4) after simplification of the common factor of the constant multiplier. The complete radix-4^2 DIF FFT algorithm is obtained by applying the CFA procedure recursively to the remaining FFTs of length N/16 in (5), as illustrated in Fig. 1. Fig. 1 indicate that the proposed radix-4^2 algorithm decomposes the N-points FFT computation by cascading the number of $\log_{16}N$ radix-16-based butterfly (R16-BF) computations, which can be split into two cascading radix-4-based butterfly (R4-BF) computations as depicted in (4) and (6). When the variables of k_1, k_2, and k_3 were treated as constants for each single output $X[k_1+4k_2+16k_3]$ as depicted in (3) and (5), the summation rages indicate that the required computation results of first and second

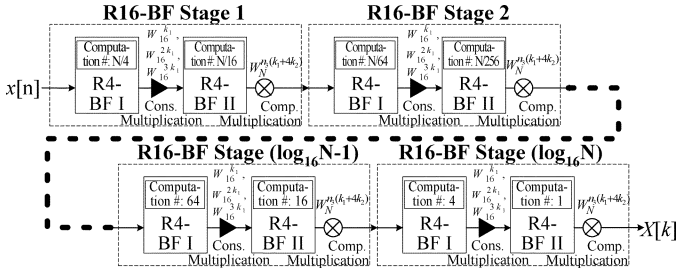


Fig. 1. CFA decomposition procedure of the proposed radix-4^2-based N-point FFT algorithm.

radix-4 butterfly stage were N/4 and N/16, respectively, as depicted in Fig. 1. The radix-4^2 algorithm has the same multiplicative complexity as the radix-16 algorithm, but still retains the radix-4 butterfly structure. Significantly, the radix-16 algorithm clearly has a lower multiplicative complexity than other low-radix algorithm, such as a radix-2^2 algorithm. For instance, the number of complex multiplications of the 256-point FFT computation adopting the radix-2^2 and radix-4^2 algorithms are 1539 and 224, respectively. Thus, the proposed design based on the new radix-4^2 algorithm has a lower multiplication complexity (85.4%) than the R2^2SDF design [9], [12]. Furthermore, as mentioned before, the radix-4^2 algorithm does not require any multiplication in the single butterfly structure.

B. Radix-4^2-Based IFFT Formula

Following the similar procedure, the radix-4^2 IFFT algorithm can be obtained as follows. The IFFT of the N-point input $X[k]$ is given by

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\cdot W_N^{-kn}. \tag{7}$$

In (7), the coefficient $1/N$ can be implemented with the simple right-shift circuit. This study focuses on the 256-point pipeline FFT/IFFT computations, where the eight right-shift is induced. Thus, the IFFT derivation results can be written as

$$\begin{aligned} &x(n_1+4n_2+16n_3)\\ &\quad=\frac{1}{N}\sum_{k_3=0}^{\frac{N}{16}-1}\sum_{k_2=0}^{3}\sum_{k_1=0}^{3} X\left[\frac{N}{4}k_1+\frac{N}{16}k_2+k_3\right]\\ &\qquad\times W_N^{-\left(\frac{N}{4}k_1+\frac{N}{16}k_2+k_3\right)(n_1+4n_2+16n_3)}\\ &\quad=\frac{1}{N}\left\{\sum_{k_3=0}^{\frac{N}{16}-1}\left\{\sum_{k_2=0}^{3} B_{\frac{N}{4}}^{n_1}\left(\frac{N}{16}k_2+k_3\right)W_N^{-\frac{N}{16}k_2(n_1+4n_2)}\right\}\right.\\ &\qquad\left.\times W_N^{-k_3(n_1+4n_2)}\right\}W_{\frac{N}{16}}^{-k_3n_3}\\ &\quad=\frac{1}{N}\left\{\sum_{k_3=0}^{\frac{N}{16}-1} B_{\frac{N}{16}}^{n_1,n_2}(k_3)W_N^{-k_3(n_1+4n_2)}\right\}W_{\frac{N}{16}}^{-k_3n_3} \end{aligned} \tag{8}$$

where the butterfly structure of the first and second stage has the form

$$\begin{aligned} &B_{\frac{N}{4}}^{n_1}\left(\frac{N}{16}k_2+k_3\right)\\ &\quad=x\left(\frac{N}{16}k_2+k_3\right)+(j)^{n_1}x\left(\frac{N}{16}k_2+k_3+\frac{N}{4}\right)\\ &\qquad+(-1)^{n_1}x\left(\frac{N}{16}k_2+k_3+\frac{2N}{4}\right)\\ &\qquad+(-j)^{n_1}x\left(\frac{N}{16}k_2+k_3+\frac{3N}{4}\right) \end{aligned} \tag{9a}$$

and

$$\begin{aligned} B_{\frac{N}{16}}^{n_1,n_2}(k_3) &= B_{\frac{N}{4}}^{n_1}(k_3)+W_{16}^{-n_1}\left[(j)^{n_2}B_{\frac{N}{4}}^{n_1}\left(k_3+\frac{N}{16}\right)\right]\\ &\quad+W_{16}^{-2n_1}\left[(-1)^{n_2}B_{\frac{N}{4}}^{n_1}\left(k_3+\frac{2N}{16}\right)\right]\\ &\quad+W_{16}^{-3n_1}\left[(j)^{n_2}B_{\frac{N}{4}}^{n_1}\left(k_3+\frac{3N}{16}\right)\right]. \end{aligned} \tag{9b}$$

Notably, the only difference between FFT and IFFT algorithm are the sign bits as given in (4), (6), (9a), and (9b). Therefore, the pipeline FFT/IFFT processor can be easily implemented with a single module by controlling the sign coefficient. Additionally, the proposed pipeline IFFT processor has a similar butterfly structure and a single constant multiplier structure with the proposed pipeline FFT processor, which could replace the three multipliers: $W_{16}^{-n_1}$, $W_{16}^{-n_2}$, and $W_{16}^{-n_3}$.

C. 8 × 8 2-D FFT and 8 × 8 2-D DCT Formula

Two concurrent 2-D DCTs can be calculated by the single 2-D shifted FFT (SFFT) algorithm [11] from the input reordering and post computation. This study presented a high-speed pipeline processor to support the triple-mode 256-point FFT/IFFT/8 × 8 2-D DCT with the radix-4^2 algorithm. Two concurrent 2-D DCTs results can be obtained by the proposed radix-4^2-based architecture in the 8 × 8 2-D DCT mode. The 8 × 8 2-D DCT $C[k_1,k_2]$ of the input signal $x(n_1,n_2)$ is given by

$$\begin{aligned} C[k_1,k_2] &= \frac{1}{4}b(k_1)b(k_2)\sum_{n_1=0}^{7}\sum_{n_2=0}^{7}x(n_1,n_2)\\ &\quad\cdot\cos\left(\frac{\pi\left(n_1+\frac{1}{2}\right)k_1}{8}\right)\cdot\cos\left(\frac{\pi\left(n_2+\frac{1}{2}\right)k_2}{8}\right). \end{aligned} \tag{10}$$

This study neglects the post-scaling factor of $(1/4)b(k_1)b(k_2)$ in (10). The input data $x(n_1,n_2)$ could then be reordered as

$$\begin{aligned} y(i_1,i_2) &= x(2i_1,2i_2)\\ y(i_1,7-i_2) &= x(2i_1,2i_2+1)\\ y(7-i_1,i_2) &= x(2i_1+1,2i_2)\\ y(7-i_1,7-i_2) &= x(2i_1+1,2i_2+1) \end{aligned} \tag{11}$$

where $i_1 = i_2 = 0,1,2,3$. After the scaling and input data reordering of (11), (10) can be recast as

$$X[k_1,k_2] = \sum_{n_1=0}^{7}\sum_{n_2=0}^{7} y(n_1,n_2)\cdot\cos\left(\frac{\pi k_1(1+4n_1)}{16}\right)\cdot\cos\left(\frac{\pi k_2(1+4n_2)}{16}\right). \tag{12}$$

The value of $X[k_1,k_2]$ is then calculated with the 8 × 8 2D SFFT with a time-domain shift of 1/4 samples

$$\begin{aligned} Y_s[k_1,k_2] &= \sum_{n_1=0}^{7}\sum_{n_2=0}^{7} y(n_1,n_2)\cdot W_8^{\left(n_1+\frac{1}{4}\right)k_1}\cdot W_8^{\left(n_2+\frac{1}{4}\right)k_2} \\ &= W_8^{\frac{1}{4}k_1}\cdot W_8^{\frac{1}{4}k_2}\cdot\sum_{n_1=0}^{7}\sum_{n_2=0}^{7} y(n_1,n_2)W_8^{n_1k_1}\cdot W_8^{n_2k_2} \\ &= W_8^{\frac{1}{4}k_1}\cdot W_8^{\frac{1}{4}k_2}\cdot Y[k_1,k_2] \end{aligned} \tag{13}$$

where $0 \le k_1,k_2,n_1,n_2 \le 7$. In (13), the 8 × 8 2-D FFT $Y[k_1,k_2]$ of the input signal $y(n_1,n_2)$ is given by

$$Y[k_1,k_2] = \sum_{n_1=0}^{7}\sum_{n_2=0}^{7} y(n_1,n_2)\cdot W_8^{n_1k_1}\cdot W_8^{n_2k_2} \tag{14}$$

where $0 \le k_1,k_2,n_1,n_2 \le 7$. Since the input data $y(n_1,n_2)$ form a real-valued sequence, the second half output can be derived as

$$\begin{aligned} Y_s[8-k_1,k_2] &= \sum_{n_1=0}^{7}\sum_{n_2=0}^{7} y(n_1,n_2)\cdot W_8^{\left(n_1+\frac{1}{4}\right)(8-k_1)} \\ &\quad\cdot W_8^{\left(n_2+\frac{1}{4}\right)k_2} \\ &= (-j)\cdot\sum_{n_1=0}^{7}\sum_{n_2=0}^{7} y(n_1,n_2)\cdot W_8^{-\left(n_1+\frac{1}{4}\right)k_1} \\ &\quad\cdot W_8^{\left(n_2+\frac{1}{4}\right)k_2} \end{aligned} \tag{15}$$

where $0 \le k_1,k_2,n_1,n_2 \le 7$. By combining (13) and (15), the 8 × 8 2-D DCT output can be recast as

$$X[k_1,k_2] = \frac{1}{2}\left\{\mathrm{Re}\left[Y_S(k_1,k_2)\right] - \mathrm{Im}\left[Y_S(8-k_1,k_2)\right]\right\} \tag{16}$$

where $0 \le k_1,k_2 \le 7$. Equation (16) adopts only the real value of $Y_S(k_1,k_2)$ and the imaginary value of $Y_S(8-k_1,k_2)$ to calculate the $X[k_1,k_2]$. By combining two reordered input sequences $\{y_1(n_1,n_2)\},\{y_2(n_1,n_2)\}$ for two independent sequences $\{x_1(n_1,n_2)\},\{x_2(n_1,n_2)\}$, and forming a complex input sequence $\{y(n_1,n_2) = y_1(n_1,n_2) + \mathrm{j}y_2(n_1,n_2)\}$, the double throughput of 2-D 8 × 8 DCT of $\{x_1(n_1,n_2)\},\{x_2(n_1,n_2)\}$ can be derived by single 2-D 8 × 8 SFFT computation. Consequently, two independent 8 × 8 2-D DCTs $X_1[k_1,k_2], X_2[k_1,k_2]$ of $x_1(n_1,n_2), x_2(n_1,n_2)$, respectively, can then be created as

$$\begin{aligned} X_1[k_1,k_2] &= \frac{1}{4}\left\{\mathrm{Re}\left[Y_s(k_1,k_2)\right] - \mathrm{Re}\left[Y_s(8-k_1,8-k_2)\right]\right\} \\ &\quad - \frac{1}{4}\left\{\mathrm{Im}\left[Y_s(8-k_1,k_2)\right] + \mathrm{Im}\left[Y_s(k_1,8-k_2)\right]\right\} \end{aligned} \tag{17a}$$

$$\begin{aligned} X_2[k_1,k_2] &= \frac{1}{4}\left\{\mathrm{Im}\left[Y_s(k_1,k_2)\right] - \mathrm{Im}\left[Y_s(8-k_1,8-k_2)\right]\right\} \\ &\quad + \frac{1}{4}\left\{\mathrm{Re}\left[Y_s(8-k_1,k_2)\right]\right. \\ &\qquad \left. + \mathrm{Re}\left[Y_s(k_1,8-k_2)\right]\right\}. \end{aligned} \tag{17b}$$

To help understand the serial pipeline operation, the 2-D location $X[k_1,k_2]$ and $x(n_1,n_2)$ can be substituted as $X[8k_1+k_2]$ and $x(8_n1+n_2)$, respectively. Then, the specific 2-D linear index map is applied as follows:

$$\begin{aligned} n_1 &= 4n_{11} + n_{12} \\ k_1 &= 2k_{11} + k_{12} \end{aligned}$$

where

$$\begin{aligned} &0 \le n_{11} \le 1, \quad 0 \le n_{12} \le 3, \quad 0 \le n_2 \le 7 \\ &0 \le k_{12} \le 1, \quad 0 \le k_{11} \le 3, \quad 0 \le k_2 \le 7. \end{aligned} \tag{18}$$

The word numbers of the shift registers in the post-computation of the fourth stage can be minimized by following the specific mapping in (18). The 8 × 8 2-D FFT CFA form can then be written as

$$\begin{aligned} &Y[16k_{11}+8k_{12}+k_2] \\ &= \sum_{n_2=0}^{7}\left\{\sum_{n_{12}=0}^{3}\left\{\sum_{n_{11}=0}^{1} y(32n_{11}+8n_{12}+n_2)W_8^{4n_{11}k_{12}}\right\}\right. \\ &\qquad \left.\times W_8^{n_{12}(2k_{11}+k_{12})}\right\}W_8^{n_2k_2} \\ &= \sum_{n_2=0}^{7}\left\{\sum_{n_{12}=0}^{3} B_{32}^{k_{12}}(8n_{12}+n_2)W_8^{n_{12}(2k_{11}+k_{12})}\right\}W_8^{n_2k_2} \\ &= \sum_{n_2=0}^{7} B_8^{k_{12},k_{11}}(n_2)W_8^{n_2k_2} \\ &= B_{\mathrm{even}}^{k_{12},k_{11},k_2} + W_8^{k_2}B_{\mathrm{odd}}^{k_{12},k_{11},k_2}. \end{aligned} \tag{19}$$

The butterfly structures for 8 × 8 2-D DCT, corresponding to (10)–(19), are summarized as follows.

Butterfly stage I:

$$B_{32}^{k_{12}}(8n_{12}+n_2) = y(8n_{12}+n_2) + (-1)^{k_{12}}y(8n_{12}+n_2+32).$$

Butterfly stage II:

$$B_8^{k_{12},k_{11}}(n_2) = B_{32}^{k_{12}}(n_2) + W_8^{k_{12}}\left[(-j)^{k_{11}} B_{32}^{k_{12}}(n_2+8)\right] + W_8^{2k_{12}}\left[(-1)^{k_{11}} B_{32}^{k_{12}}(n_2+16)\right] + W_8^{3k_{12}}\left[(j)^{k_{11}} B_{32}^{k_{12}}(n_2+32)\right].$$

Butterfly stage III:

$$B_{even}^{k_{12},k_{11},k_2} = \left\{B_8^{k_{12},k_{11}}(0) + (-j)^{k_2} B_8^{k_{12},k_{11}}(2) + (-1)^{k_2} B_8^{k_{12},k_{11}}(4) + (j)^{k_2} B_8^{k_{12},k_{11}}(6)\right\}$$

$$B_{\text{odd}}^{k_{12},k_{11},k_2} = \left\{B_8^{k_{12},k_{11}}(1) + (-j)^{k_2} B_8^{k_{12},k_{11}}(3) + (-1)^{k_2} B_8^{k_{12},k_{11}}(5) + (j)^{k_2} B_8^{k_{12},k_{11}}(7)\right\}.$$

The additional stage of 2-D DCT:

$$B_1^{k_{12},k_{11},k_2}(n_2) = B_{\text{even}}^{k_{12},k_{11},k_2} + W_8^{k_2} B_{\text{odd}}^{k_{12},k_{11},k_2}.$$

The time-domain shift stage of 2-D DCT:

$$Y_s[8k_1+k_2] = W_8^{\frac{1}{4}(k_1+k_2)} \cdot Y[8k_1+k_2].$$

Butterfly stage IV:

$$\begin{aligned} X_1[8k_1+k_2] &= \frac{1}{4}\{\text{Re}[Y_s(8k_1+k_2)] - \text{Re}[Y_s(72-8k_1-k_2)]\} \\ &\quad - \frac{1}{4}\{\text{Im}[Y_s(64-8k_1+k_2) - \text{Im}[Y_s(8+8k_1-k_2)]]\} \\ X_2[8k_1+k_2] &= \frac{1}{4}\{\text{Im}[Y_s(8k_1+k_2)] - \text{Im}[Y_s(72-8k_1-k_2)]\} \\ &\quad + \frac{1}{4}\{\text{Re}[Y_s(64-8k_1+k_2) + \text{Re}[Y_s(8+8k_1-k_2)]]\}. \end{aligned} \quad (20)$$

Two 8 × 8 2-D DCT computation results $X_1[8k_1 + k_2]$ and $X_2[8k_1 + k_2]$ are calculated concurrently in the post-computation of the butterfly stage IV. The 8 × 8 2-D IDCT computation can also be obtained following a similar decomposition procedure. Because of the cost-effective constraint in the physical design, this study only considers the triple-mode FFT/IFFT and 2-D DCT computations. The derivation results of the radix-4^2-based FFT/IFFT/2-D DCT algorithm indicate that all butterfly computation can be easily implemented with four four-input complex adders and some shuffle circuits. The radix-4 butterfly structure has no multipliers. Additionally, the regular structure can be easily derived in both the 8 × 8 2-D DCT and 256-point FFT/IFFT pipeline processor architecture.

III. Pipeline 256-Point FFT/IFFT/8 × 8 2-D-DCT Processor Architecture

He *et al.* presented several pipeline FFT/IFFT architectures [9]. The serial delay feedback (SDF)-based architecture is known to have a low hardware cost and high cost-efficiency advantages with the feedback type shift registers architecture [9], [12], [13]. The delay-feedback type shift register approaches are always more efficient than other corresponding approaches in terms of memory utilization since the butterfly output share the same storage with its input [9]. The $\text{R2}^2\text{SDF}$ pipeline architecture has the same computation complexity as the radix-4 algorithm, and few hardware requirements as the radix-2 algorithm. This study presents a $\text{R4}^2\text{SDF}$ reconfigurable pipeline architecture with a low computation complexity as the radix-16 algorithm, and low hardware requirements as the radix-4 algorithm. Significantly, the proposed triple-mode radix-4 butterfly structure, like the radix-2 butterfly structure, does not require a complex multiplier or constant multiplier. Section V presents detailed comparisons between the $\text{R4}^2\text{SDF}$ and $\text{R2}^2\text{SDF}$ architectures. This section describes a novel radix-4^2 single-path delay feedback ($\text{R4}^2\text{SDF}$) architecture to support the three modes, 256-point FFT, 256-point IFFT, and 8 × 8 2-D DCT, based on the radix-4^2 DIF FFT algorithm obtained in Section II. Fig. 2 shows a block diagram of the purposed $\text{R4}^2\text{SDF}$-based 256-point FFT/IFFT and 8 × 8 2-D DCT pipeline processor. Based on the proposed $\text{R4}^2\text{SDF}$ pipeline architecture, the cost-effective 256-point FFT/IFFT processor is first constructed. This processor only requires four butterfly stages with 255-word shift registers, two constant multipliers and one complex multiplier with one coefficient ROM, represented by black solid color in Fig. 2. Fig. 2 indicates that a single data stream passes through two constant multipliers and one complex multiplier to realize different combinations of k_1, k_2, and k_3 of $X[k_1 + 4k_2 + 16k_3]$, as illustrated in (3). Using some control circuits, one additional radix-2 butterfly (R2-BF) with an one-word shift register and additional eight-word shift register, two concurrent 2-D-DCT operations are calculated from the single 2-D-SFFT computation as depicted in (11), (17a), and (17b). These extra circuits were embedded at the first butterfly stage, the additional stage, and the fourth butterfly stage, which is represented by the gray color in Fig. 2. These circuits complete the input reordering, time domain shift, and post-computation in the 8 × 8 2-D DCT computation, respectively. Notably, only one radix-2 butterfly and nine-word shift register are needed to support additional two 8 × 8 2-D DCT computations in the original pipeline SDF-based FFT/IFFT architecture. The proposed architecture has, in total, four radix-4 butterflies, one radix-2 butterfly, two constant multipliers, one

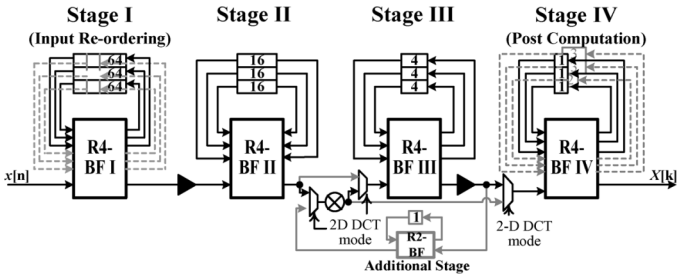


Fig. 2. Block diagram of the $\text{R4}^2\text{SDF}$-based 256-point FFT/IFFT and 8 × 8 2-D-DCT architecture.

TABLE I
CORRESPONDING EQUATION NUMBERS FOR EACH BUILDING BLOCK

Building Blocks	BFI	Cons. Mult. I	BFII	Comp. Mult.	BFIII	Cons. Mult. II	BFIV	R2-BF
FFT Eq. #	(4)	(6)	(6)	(5)	(4)	(6)	(6)	N/A
IFFT Eq. #	(9a)	(9b)	(9b)	(8)	(9a)	(9b)	(9b)	N/A
DCT Eq. #	(11), (19)	(19)	(19)	(13)	(19)	(19)	(17)	(19)

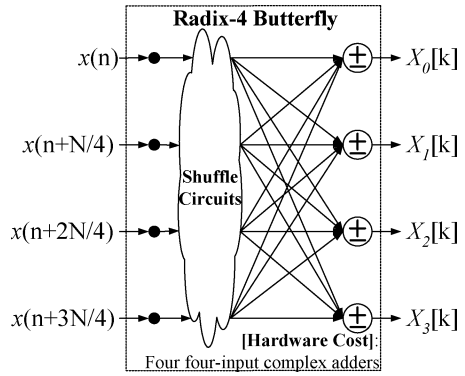


Fig. 3. Block diagram of the radix-4 butterfly architecture.

complex multiplier, one coefficient ROM, and a 264-word shift register. To help understand the corresponding functions of each building block, the respective equation numbers related to each element are shown in Table I. The detailed operations of each element are described as follows.

A. Radix-4 Butterfly and Radix-2 Butterfly

The derivation results of the radix-4^2-based algorithm reveal that both the FFT/IFFT butterfly computation in (4), (6), (9a), (9b), and the 8 × 8 2-D-DCT butterfly computation in (20), can be easily completed with the radix-4-based butterfly architecture. The only difference between the 8 × 8 2-D-DCT and the 256-point FFT/IFFT butterfly computation is the summands and minuends at the butterfly stage one and four, which can be easily realized with the multiplex circuits in the radix-4 butterfly structure. Significantly, the number of the two first stages in the 8 × 8 2-D-DCT computation can be completed concurrently at the first butterfly stage in parallel. Additionally, the input reordering operation at the first stage and the post-computation operation at the fourth stage of the 8 × 8 2-D-DCT mode are described in detail. Fig. 3 illustrates the proposed radix-4 butterfly structure, which only includes four four-input complex adders with no complex multipliers inside. This configuration means that the proposed radix-4 butterfly structure has a low hardware cost of higher-radix butterfly structures. Moreover, the proposed radix-4^2 algorithm has the same complex multiplication complexity as the radix-16 algorithm, so has the high cost efficiency of lower radix architectures. To obtain the additional stage of the 8 × 8 2-D DCT mode in (20), one additional SDF-based radix-2 butterfly structure with one word shift register is required, as illustrated in Fig. 2. The additional stage of the 2-D DCT mode only requires two 2-input complex adders and one shift register, giving it a small hardware penalty.

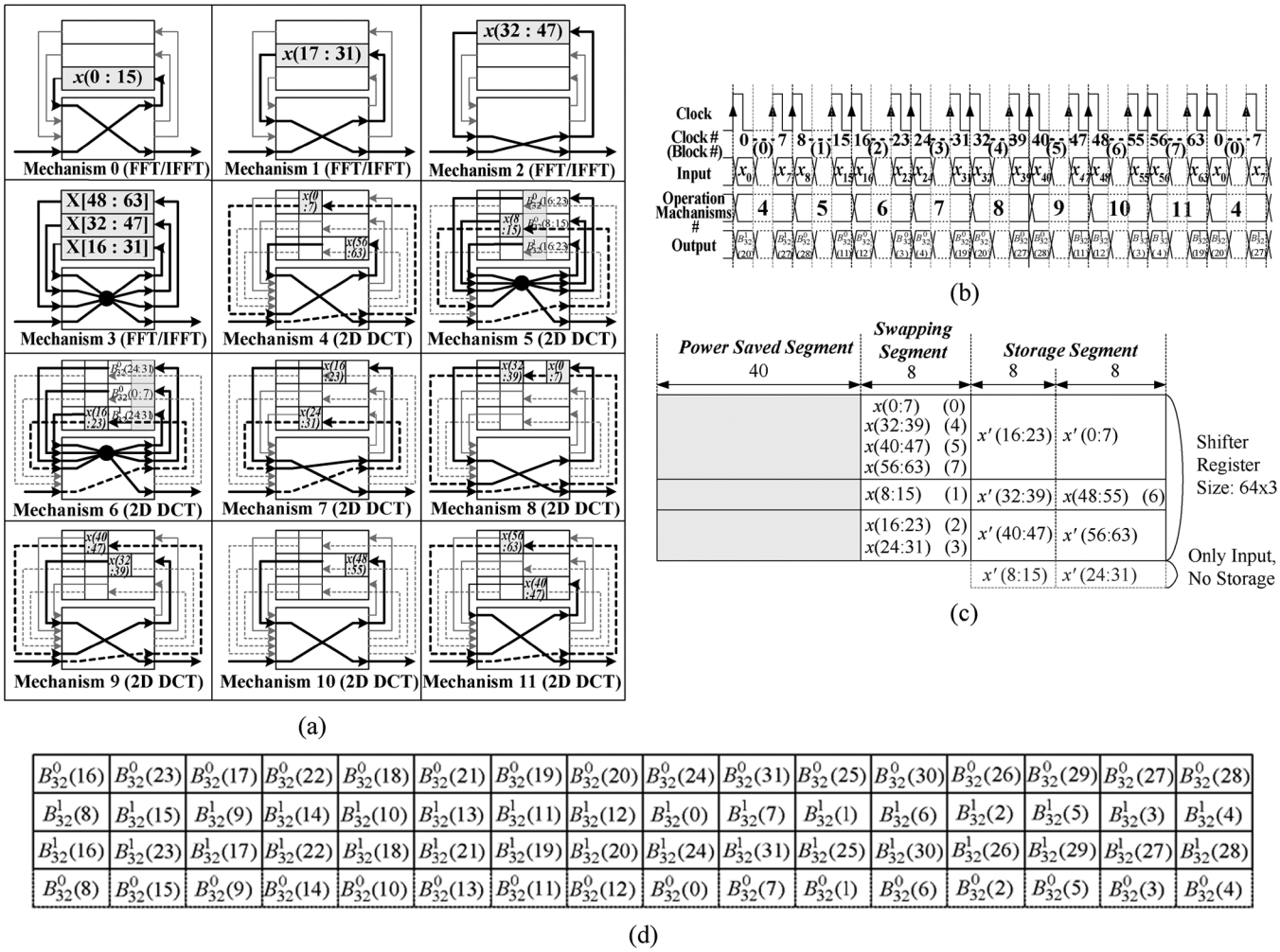
B. Memory Structure

The memory structure of butterfly stage is well known to be an important issue for the high cost-effective FFT/IFFT pipeline processor design. From the exiting researches, there are mainly two different approaches: delay commutator (DC) [9] and delay feedback (DF) [9], [12], [13]. In this study, the DF-based memory structure is adopted and depicted in Fig. 2. In order to compute the radix-4-based butterfly computations, the input data and the intermediate results have to be reordered as four concurrently data streams using memory as shown in Fig. 3. Each radix-4 butterfly unit applies the three parallel memories to store the serial data input and butterfly output in the feedback paths as presented in Fig. 2. The timing sequence of N-point FFT computation can be divided into four stages, each containing N/4 clock cycles. In the first N/4 cycles (i.e., first stage), the butterfly units simply store the input samples into the first feedback memory. Similarly, the second and third feedback memory are filled in the second and third stages. After the 3N/4 cycles, the butterfly units retrieves the $x(n)$, $x(n + N/4)$, and $x(n + 2N/4)$ samples from the feedback memory, performs corresponding operations with the sample $x(n + 3N/4)$ and then feeds the output into the next butterfly units as depicted in Fig. 2. The required number of memory cells for the kth stage is $3 \times N/(4^k)$. Thus, the 256-points FFT/IFFT computations require the 64 × 3, 16 × 3, 4 × 3, and 1 × 3 word shift registers in the first, second, third, and fourth butterfly stages, respectively. Significantly, the SDF-based pipeline FFT/IFFT structure is highly regular, which has the high effective memory structure with the simpler routing complexity [9], [12], [13]. In this study, the shift registers were all realized by the cascaded flip-flops, which are composed of two latch circuits.

C. Input Reordering and First Butterfly Computation

Consider the new radix-4^2 algorithm presented in Section II. The proposed SDF architecture is estimated to need 64 × 3-word shift registers at the first butterfly stage in the 256-point FFT/IFFT mode. Although the 8 × 8 2-D DCT mode only requires 16 × 3-word shift registers at the first butterfly stage, the 8 × 8 2-D DCT mode needs a swapping buffer to complete the input reordering and post-computation in (11) and (17) from the 8 × 8 2-D SFFT computation. Notably, the number of shift registers at the fourth butterfly stage for the post computation in the 8 × 8 2-D DCT mode depends on the sequential order of the input data at the first butterfly stage. Following the specific linear mapping in (18), the number of shift registers can be reduced to only eight words at the fourth butterfly stage, as revealed in Fig. 2. Comparing with the other linear mapping, the proposed architecture could reduce at least 96% shift registers cost. The segmented shift registers (SSR) structure is also proposed to realize both the input reordering and butterfly computation operation at the first stage to support the 256-point FFT/IFFT and 8 × 8 2-D DCT modes.

Fig. 4(a) shows the 12 proposed operation mechanisms of the first butterfly stage to finish the input reordering and the first-stage computation. Operation mechanisms 0–3 are adopted



$B_{32}^0(16)$	$B_{32}^0(23)$	$B_{32}^0(17)$	$B_{32}^0(22)$	$B_{32}^0(18)$	$B_{32}^0(21)$	$B_{32}^0(19)$	$B_{32}^0(20)$	$B_{32}^0(24)$	$B_{32}^0(31)$	$B_{32}^0(25)$	$B_{32}^0(30)$	$B_{32}^0(26)$	$B_{32}^0(29)$	$B_{32}^0(27)$	$B_{32}^0(28)$
$B_{32}^1(8)$	$B_{32}^1(15)$	$B_{32}^1(9)$	$B_{32}^1(14)$	$B_{32}^1(10)$	$B_{32}^1(13)$	$B_{32}^1(11)$	$B_{32}^1(12)$	$B_{32}^1(0)$	$B_{32}^1(7)$	$B_{32}^1(1)$	$B_{32}^1(6)$	$B_{32}^1(2)$	$B_{32}^1(5)$	$B_{32}^1(3)$	$B_{32}^1(4)$
$B_{32}^1(16)$	$B_{32}^1(23)$	$B_{32}^1(17)$	$B_{32}^1(22)$	$B_{32}^1(18)$	$B_{32}^1(21)$	$B_{32}^1(19)$	$B_{32}^1(20)$	$B_{32}^1(24)$	$B_{32}^1(31)$	$B_{32}^1(25)$	$B_{32}^1(30)$	$B_{32}^1(26)$	$B_{32}^1(29)$	$B_{32}^1(27)$	$B_{32}^1(28)$
$B_{32}^0(8)$	$B_{32}^0(15)$	$B_{32}^0(9)$	$B_{32}^0(14)$	$B_{32}^0(10)$	$B_{32}^0(13)$	$B_{32}^0(11)$	$B_{32}^0(12)$	$B_{32}^0(0)$	$B_{32}^0(7)$	$B_{32}^0(1)$	$B_{32}^0(6)$	$B_{32}^0(2)$	$B_{32}^0(5)$	$B_{32}^0(3)$	$B_{32}^0(4)$

(d)

Fig. 4. Block diagram of the proposed first radix-4 butterfly stage in the $\mathrm{R4^2SDF}$-based 256-point FFT/IFFT and 8 × 8 2-D-DCT architecture. (a) The proposed 12 reconfigurable operation mechanisms of the first butterfly stage. (b) The timing sequences of operation mechanism in the first butterfly stage. (c) The storage content in SSR in the 8 × 8 2-D-DCT mode. (d) The content of the 8 × 8 2-D DCT computation result in SSR.

in the FFT/IFFT computation, and operation mechanisms 4–11 are adopted in the 8 × 8 2-D DCT computation. In the 8 × 8 2-D DCT mode, reconfigurable operation mechanisms 5 and 6 are adopted for the butterfly computation, and reconfigurable operation mechanisms 4 and 7–11 are adopted for the input reordering. Fig. 4(b) lists the corresponding timing sequence of the first butterfly stage, which discusses the relationships among the input data, output data, and respective operation mechanisms during each clock (block number) in the 8 × 8 2-D DCT mode. Additionally, Fig. 4(c) and (d) illustrate the data content before and after the butterfly computation in SSR, respectively. The first stage butterfly computation is completed by applying operation mechanisms 5 and 6. Most results of 8 × 8 2-D DCT computation are then pushed back into the SSR, as shown in Fig. 4(d), where $B_{32}^{k_{12}}$ denotes the computation results from (20). Fig. 4(d) presents the complete computation results. The 64 × 3-word shift register is segmented as $(40 + 8 + 16) \times 3$, which is easily realized by three dependent clock domains with a simple 3-bit counter controller, as depicted in Fig. 4(c). These three segments in the SSR are called the power-saving, swapping and storage segments, and their sizes are 40 × 3, 8 × 3, and 16 × 3, respectively. Since 2-D DCT mode as depicted in (19) has a low computation complexity, the first, second, and third butterfly stages have shift registers comprising 40 × 3, 8 × 3, and 2 × 3 words, respectively. These shift registers are set as power-saving segments, and gated to reduce power consumption. To perform the input reordering operation, 64 serial input data words are split into eight blocks of eight words, as shown in Fig. 4(b) and (c). The block numbers, written inside the brackets in Fig. 4(b) and (c), denote the serial input sequential order of eight-word blocks. In Fig. 4(c), the terms x' and x, respectively, represent the 2-D DCT image data in the previous and current frame, which both contain 64 points in each frame. Following the operation mechanisms 4, 7, 8, 9, 10, and 11 in Fig. 4(a), the serial input data of each block adopt the swapping segment as the swapping space to achieve the required storage ordering in the storage segment.

The detail timing sequence of the proposed 8 × 8 2-D DCT computations is given as follows. Operation mechanism 4 pushes the input data $x(0:7)$ into the swapping segment from the clock number 0 to 7 (block number 0). At the same time, the original data $x'(56:63)$ in the swapping segment are

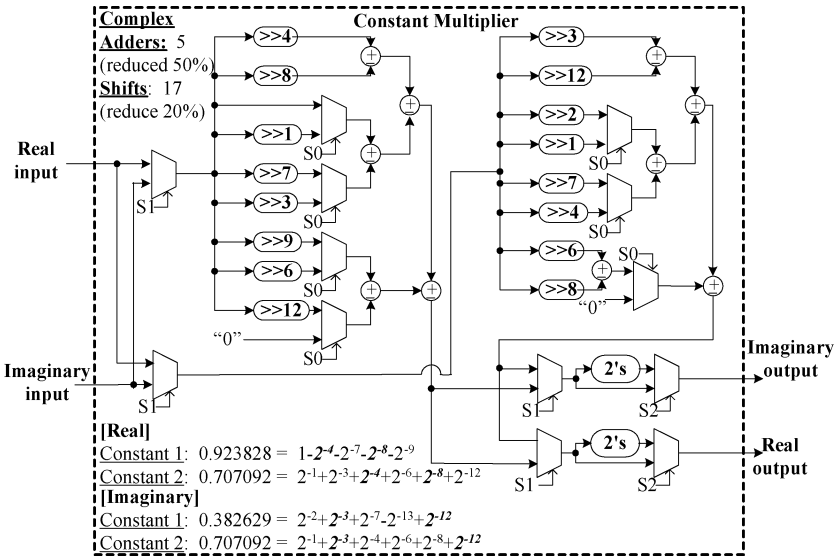


Fig. 5. Block diagram of the proposed constant multiplier architecture.

simultaneously pushed into the storage segment as illustrated in Fig. 4(a) and (c). In the following 16 clock cycles, operation mechanisms 5 and 6 replace the swapping segments with the input data $x(8 : 15)$ and $x(16 : 23)$ (i.e., block number 1 and 2), as presented in Fig. 4(a)–(c). Operation mechanisms 5 and 6 provide the original swapping segment data $x'(8 : 15)$ and $x'(24 : 31)$ for the first butterfly stage computation in (20), along with the data in the storage segment. At the same time, 48 new 8×8 2-D DCT results of $B_{32}^{k_{12}}(8n_{12} + n_2)$, as listed in the top three rows of Fig. 4(d), are pushed into the storage segment by the feedback path. Furthermore, the other 16 new 8×8 2-D DCT results, which are listed in the final row in Fig. 4(d), are pushed directly to the second butterfly stage, as shown in Fig. 4(b). Notably, the 48 different 2-D DCT results in the storage segment are pushed out one-by-one due to the swapping operation by the swapping segment data in the following 48 clock cycles. Following a similar procedure, the serial input data of block numbers 3, 4, 5, and 7 complete their respective swapping operations by operation mechanisms 7, 8, 9, and 11. The block number 6 is stored into the storage segment directly by operation mechanism 10 as illustrated in Figs. 4(a) and (c). The input reordering operation is finished after a period of 64 clock cycles, which includes 7 clock cycles of input swapping latency.

D. Constant Multiplier

Based on the derivation results in Section II, the radix-4^2 algorithm requires some complex multiplications, namely $W_{16}^{k_1}(W_{16}^{-n_1})$, $W_{16}^{2k_1}(W_{16}^{-2n_1})$, $W_{16}^{3k_1}(W_{16}^{-3n_1})$ in the 256-point FFT/IFFT mode in (6), (9), and $W_8^{k_{12}}$, $W_8^{2k_{12}}$, $W_8^{3k_{12}}$, $W_8^{k_2}$ in the 8×8 2-D DCT mode in (20). Due to the finite range of k_1 and n_1 in (6) and (9b), namely 0–3, the three complex multiplications, $W_{16}^{k_1}(W_{16}^{-n_1})$, $W_{16}^{2k_1}(W_{16}^{-2n_1})$, and $W_{16}^{3k_1}(W_{16}^{-3n_1})$ can be written as $\{W_{16}^{0}(W_{16}^{-0}), W_{16}^{1}(W_{16}^{-1}), W_{16}^{2}(W_{16}^{-2}), W_{16}^{3}(W_{16}^{-3})\}$, $\{W_{16}^{0}(W_{16}^{-0}), W_{16}^{2}(W_{16}^{-2}), W_{16}^{4}(W_{16}^{-4}), W_{16}^{6}(W_{16}^{-6})\}$, and $\{W_{16}^{0}(W_{16}^{-0}), W_{16}^{3}(W_{16}^{-3}), W_{16}^{6}(W_{16}^{-6}), W_{16}^{9}(W_{16}^{-9})\}$. Follow -ing the similar procedure, $W_8^{k_{12}}$, $W_8^{2k_{12}}$, $W_8^{3k_{12}}$, and $W_8^{k_2}$ in (20) can be expanded as $\{W_8^0, W_8^1\}$, $\{W_8^0, W_8^2\}$, $\{W_8^0, W_8^3\}$, and $\{W_8^0, W_8^1, W_8^2, W_8^3, W_8^4, W_8^5, W_8^6, W_8^7\}$. The system has in total 38 different twiddle factor values, which could be implemented as 38 different constant multipliers by only shifters and adders. Based on the SDF-based architecture, the proposed design only has to calculate one complex multiplication in (6), (9), and (19) during each clock cycle. The 38 twiddle factor values can thus be reduced to the extension of two different values of W_{16}^1 and W_{16}^2 using the complex conjugate symmetry rule. Accordingly, the other 36 twiddle factor values can be expressed as the real-imaginary swapping or sign inversion of these two constant values. Moreover, the repeated shifters and adders of two constant multipliers could be simplified using the subexpression elimination algorithm [14] as illustrated in Fig. 5. According to our implementation results, the small cost penalty for the multiplexer control (i.e., S0, S1, and S2) could be neglected as shown in Fig. 5.

Following the three steps to reduce the complex multipliers to the most economical constant multipliers are summarized as follows. First, the twiddle factors from (6), (9), and (20) are realized as the constant multipliers, which only contain shifters and adders as shown in Fig. 2. Second, the complex conjugate symmetry rule is applied to decrease the number of complex multiplications (12) to only two constant multiplications per stage with some shuffle circuits as shown in Fig. 5, thus achieving a constant multiplier cost reduction of 94.7%. Finally, the subexpression elimination algorithm [14] is adopted to reduce the number of shift circuits by more than 20%, and the number of complex adders by 50% in the one constant multiplier, as depicted in Fig. 5. The strictest constant multiplier is obtained in the purposed architecture by following these three steps. The cost penalty of the constant multiplier is thus minimized.

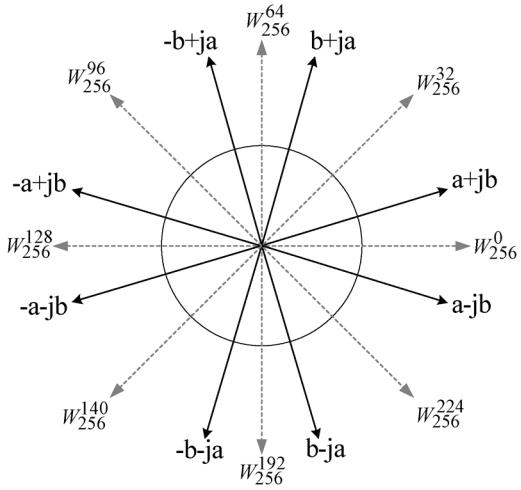


Fig. 6. Block diagram of eight-folded algorithm in the coefficient ROM.

TABLE II
DATA CONTROL OF THE COEFFICIENT ROM

H = $n_3(k_1+4k_2)$	Address Mode (H[5])	ROM address	Data Mode (H[7:5])	ROM data
0~32	0	Two's complement of H[5:0]	0	a+jb
33~63	1	H[5:0]	1	b+ja
64~95	0	Two's complement of H[5:0]	2	-b+ja
96~127	1	H[5:0]	3	-a+jb
128~159	0	Two's complement of H[5:0]	4	-a-jb
160~191	1	H[5:0]	5	-b-ja
192~223	0	Two's complement of H[5:0]	6	b-ja
224~255	1	H[5:0]	7	a-jb

E. Eight-Folded Complex Multiplier

The proposed architecture has only one complex multiplier and one coefficient ROM to realize the complex multiplication of twiddle factors $W_N^{n_3(k_1+4k_2)}$ in (3), $W_N^{-k_3(n_1+4n_2)}$ in (8) and $W_8^{(1/4)(k_1+k_2)}$ in (20). Significantly, the implementation of the time-domain shift for 8 × 8 2-D-DCT computation needs one feedback path. To decrease the ROM size, the complex conjugate symmetry rule and subexpression elimination [14] is applied to devise one eight-folded complex multiplier as shown in Fig. 6. The proposed eight-folded complex multiplier only has to store 32 words in the coefficient ROM, reducing the ROM size by 87.5%. The ROM address and data control circuit are also easily realized by the 8-bit counter controller given in Table II.

F. Post Computation

Clearly, the 256-point FFT/IFFT modes only require 1 × 3 word shift registers at the fourth butterfly stage of the proposed $\mathrm{R4^2SDF}$ architecture. However, the 8 × 8 2-D DCT mode has to implement the post-computation at the fourth butterfly stage in (17a) and (17b). As described in Section III-A, the proposed architecture follows the specific linear mapping in (19) to minimize the number of shift registers at the fourth stage. Fig. 7(a) depicts the analysis of the order of the fourth butterfly results following the specific linear mapping. Notably, the gray solid line in Fig. 7(a) represents the input data order that do not follow the required sequence. For instance, $\{Y_s[17], Y_s[23]\}$, $\{Y_s[18], Y_s[22]\}$, and $\{Y_s[19], Y_s[21]\}$ should be regarded as three groups for the fourth butterfly computation. However, the sequence of the input data at the fourth butterfly stage is $Y_s[17]$, $Y_s[18]$, $Y_s[19]$, $Y_s[21]$, $Y_s[22]$, $Y_s[23]$. Then, $Y_s[23]$ and $Y_s[21]$ should be reordered. Thus, the proposed overturn shift register (OSR) structure at fourth butterfly stage resolves this simple reordering procedure without any performance degradation, as depicted in Fig. 7(b). The desired ordering is obtained with the OSR structure at the fourth butterfly stage, along with the input reordering operation at the first butterfly stage as discussed in Section III-A. The full-pipeline $\mathrm{R4^2SDF}$ architecture can then easily follow the two concurrent 8 × 8 2-D DCT outputs.

IV. FINITE WORDLENGTH ANALYSIS

The next generation mobile-multimedia system can handle high-quality multimedia operations with embedded 256-point FFT/IFFT and 8 × 8 2-D DCT pipeline processor [3]–[5]. The system performance should then satisfy the relative specifications. A higher system performance undoubtedly implies a larger chip cost and greater power consumption, owing to the wider internal wordlength. Since the chip cost and system performance are known to be a tradeoff, this study performed a finite wordlength analysis to estimate the appropriate word-length for both 256-point FFT/IFFT and 8 × 8 2-D DCT system requirements.

A. Pipeline 256-Point FFT/IFFT

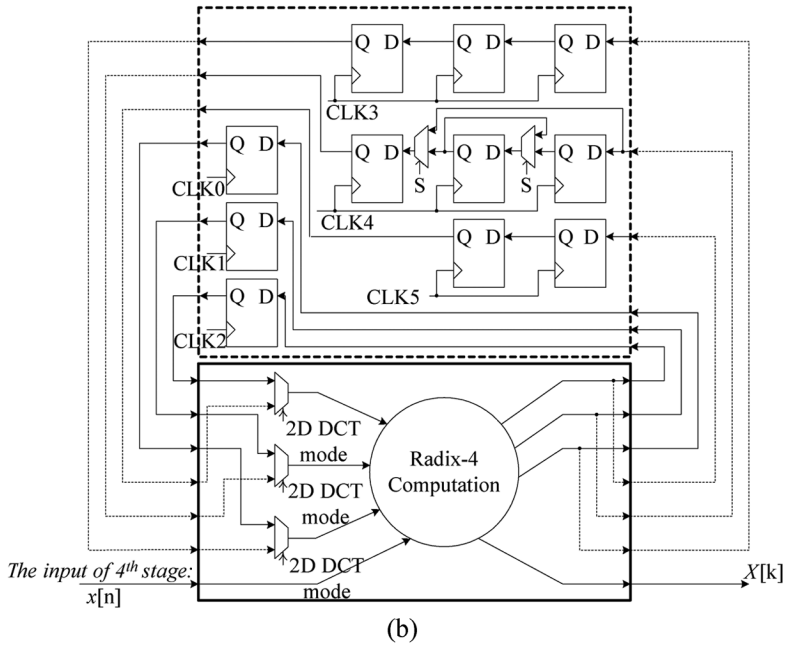
In the 256-point FFT/IFFT modes, the output signal-to-noise ratio (SNR) performance was estimated under different noise environment. First, the input data of the double floating-point precision were generated from the ideal IFFT(FFT) model by passing the additive white Gaussian noise (AWGN) channel model under five noise levels: 20, 40, 60, 80, and 100 dB. The input data with noise were sent into the proposed $\mathrm{R4^2SDF}$ pipeline FFT/IFFT architecture, which was modeled at different fixed-point levels. The output SNR was obtained by comparing the original input data with the fixed-point model output. The results after 100 000 iterations were averaged as depicted in Fig. 8, where the x- and y-axes represent the data word-length and the whole system output SNR, respectively. These analytical results demonstrate that the output SNR saturated as the data word length increased. The output SNR was increased by 20 dB for each additional three bits. The 13-bit internal wordlength for each function units produced satisfactory results under 40-dB noise environments, satisfying the IEEE 802.16e WiMAX [1] standard.

B. Pipeline 8 × 8 2-D DCT

In the 8 × 8 2-D DCT mode, the performance of the proposed $\mathrm{R4^2SDF}$ pipeline architecture was measured in common video compression standards, including the high-quality DV standard [15]. The DV standard defines some tolerances that the 8 × 8 2-D DCT computation maintains the accuracy and

$Y_s[0]$	$Y_s[1]$	$Y_s[2]$	$Y_s[33]$	$Y_s[34]$	$Y_s[32]$	$Y_s[36]$	$Y_s[17]$	$Y_s[18]$	$Y_s[19]$	$Y_s[16]$	$Y_s[9]$	$Y_s[10]$	$Y_s[11]$	$Y_s[8]$	$Y_s[25]$	$Y_s[26]$	$Y_s[27]$	$Y_s[24]$
$Y_s[4]$	$Y_s[7]$	$Y_s[6]$	$Y_s[39]$	$Y_s[38]$			$Y_s[23]$	$Y_s[22]$	$Y_s[21]$	$Y_s[48]$	$Y_s[15]$	$Y_s[14]$	$Y_s[13]$	$Y_s[56]$	$Y_s[31]$	$Y_s[30]$	$Y_s[29]$	$Y_s[49]$
	$Y_s[3]$		$Y_s[35]$				$Y_s[49]$	$Y_s[50]$	$Y_s[51]$	$Y_s[20]$	$Y_s[57]$	$Y_s[58]$	$Y_s[59]$	$Y_s[12]$	$Y_s[41]$	$Y_s[42]$	$Y_s[43]$	$Y_s[28]$
	$Y_s[5]$		$Y_s[37]$				$Y_s[55]$	$Y_s[54]$	$Y_s[53]$	$Y_s[52]$	$Y_s[63]$	$Y_s[62]$	$Y_s[61]$	$Y_s[60]$	$Y_s[47]$	$Y_s[46]$	$Y_s[45]$	$Y_s[44]$

(a)



(b)

Fig. 7. Block diagram of the proposed fourth butterfly stage in the $\mathrm{R4^2SDF}$-based 256-point FFT/IFFT and 8 × 8 2-D-DCT architecture. (a) The data context of the fourth butterfly stage in the 8 × 8 2-D DCT mode. (b) The OSR structure of the fourth butterfly stage.

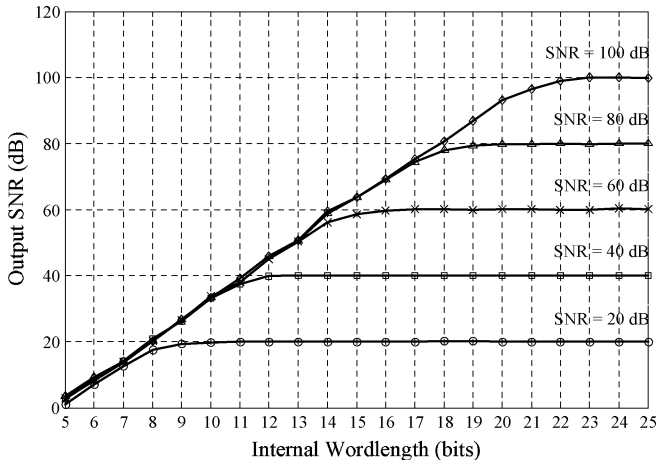


Fig. 8. Finite wordlength analysis of the proposed pipeline $\mathrm{R4^2SDF}$-based 256 points FFT/IFFT architecture.

consequently an acceptable reconstructed video quality [15], [16]. The DV standard applies four measured error criteria, namely the probability of occurrence of error, mean square errors (MSE), peak mean square error (PMSE), and steady AC coefficients [16]. Following the procedure in the preceding subsection, the double floating-point precision is assumed to be precise in comparing with the fixed-point computation. The zero-mean white input sequences are generated by a random-number generator in the range [−128, 127]. After the repeated 100, 000 loops, the probability of occurrence of error, which is greater than 1, is less than 1×10^{-15}. Moreover, the steady AC coefficients of the proposed fixed-point 2-D 8 × 8 DCT model are all zero under the equal-values input. Fig. 9(a) and (b) depict the MSE and PMSE simulation results, respectively. Notably, the proposed architecture could satisfy the limitation of MSE and PMSE of the DV standard, when the internal wordlength is greater than 12 bits. Thus, the 13-bit internal word length for each function units is the qualified internal wordlength for the DV standard. Fig. 9(c) and (d) indicate that the overall mean error (OME) is below 0.01, and the peak signal to noise ratio (PSNR) is close to 60 dB, which has the required video compression quality under the configuration of the 13-bit internal wordlength [17]. According to the finite wordlength analysis of the proposed $\mathrm{R4^2SDF}$ 256-point FFT/IFFT pipeline architecture a 13-bit internal wordlength achieves the satisfactory results under the 40-dB noise quality, thus satisfying the IEEE 802.16e standard. The 13-bit internal wordlength was thus chosen for the proposed $\mathrm{R4^2SDF}$ 256-point FFT/IFFT/2-D DCT RSoC IP to meet the requirements of next-generation handheld applications.

V. Comparison Results and Chip Implementation

A. Comparison Between $R4^2SDF$ and $R2^2SDF$

He *et al.* presented the efficient pipeline FFT processor, several reliable architectures and the detailed comparison of their hardware costs [9]. A comparison of these architectures

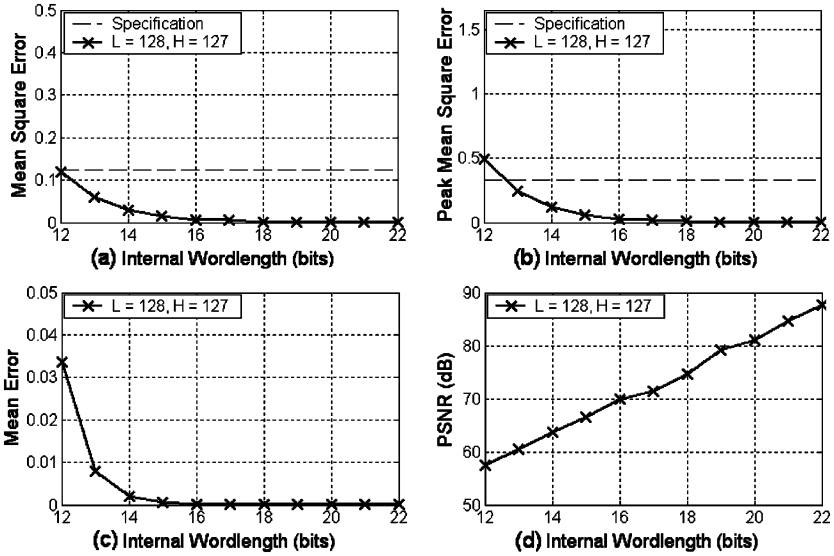


Fig. 9. Finite wordlength analysis of the proposed pipeline $\mathrm{R4^2SDF}$-based 8 × 8 2-D DCT architecture. (a). Overall mean square error analysis. (b) Peak mean square error analysis. (c). Overall mean error analysis. (d). Peak mean error analysis.

indicates that $\mathrm{R2^2SDF}$ has the highest butterfly utilization of 50%, a the highest complex multiplier utilization of 75%, and the lowest hardware resource requirement [9], [12]. Additionally, the SDF-based design has the structural merits of high regularity and modularity with simple wiring complexity, making it very appropriate for the VLSI implementation of the pipeline FFT processor design [9], [12], [13]. This section presents the comprehensive comparison results of several famous pipeline FFT/IFFT architectures to demonstrate the high cost-efficiency of the proposed $\mathrm{R4^2SDF}$ FFT/IFFT architecture. The architectures were compared in two indices, namely cost and utilization, to express the cost efficient of the proposed FFT/IFFT architecture, as listed in Tables III and IV. Table III lists the required hardware resources, where T denotes the number of complex adders required in the implementation of the constant multiplier. Significantly, the proposed constant multiplier is minimized using complex conjugate symmetry rule and subexpression elimination algorithm. The area of the complex multiplier is known to be one dominant cost index in the pipeline FFT/IFFT design. The comparison results in Table IV clearly demonstrate that the proposed $\mathrm{R4^2SDF}$-based-FFT/IFFT architecture has the fewest complex multipliers requirement among other pipeline architectures. The 256-point FFT/IFFT architecture only needs one complex multiplier, which is 67% and 95% below the requirement of the $\mathrm{R2^2SDF}$ and R8MDC FFT/IFFT architectures, respectively. Additionally, the proposed architecture applies the feedback type memory structure to maintain the minimum shift registers requirement. Although the proposed $\mathrm{R4^2SDF}$-based architecture needs slightly more complex adders than the $\mathrm{R2^2SDF}$-based architecture, this small cost penalty is acceptable.

To estimate the total chip cost in the 256-point FFT/IFFT architectures, which includes the number of complex multipliers, complex adders and memory size, the conventional comparative methodology [13], [18] with the unit of equivalent adders was adopted to estimate the cost of each different architecture.

TABLE III
HARDWARE COST COMPARISONS OF THE PIPELINED FFT/IFFT ARCHITECTURE

Pipeline architecture	Mult. complexity	Complex Mult.	Complex adders (including constant mult.)	Complex Memory Size	Equivalent area in 256 points
R2SDF	Radix-2	$\log_2$N-2	2$\log_2$N	N-1	647.5
R4SDF	Radix-4	$\log_4$N-1	8$\log_4$N	N-1	513.5
R8SDF	Radix-8	$\log_8$N-1	(24+2T)$\log_8$N	N-1	617.5
$\mathrm{R2^2SDF}$	Radix-2^2	$\log_4$N-1	4$\log_4$N	N-1	497.5
$\mathrm{R2^3SDF}$	Radix-2^3	2($\log_8$N-1)	6$\log_8$N	N-1	655.5
R2MDC	Radix-2	$\log_2$N-2	2$\log_2$N	1.5N-2	812.6
$\mathrm{R2^2MDC}$	Radix-2^2	$\log_2$N-2	2$\log_2$N	1.5N-2	812.6
R4MDC	Radix-4	3$\log_4$N-3	4$\log_2$N	2.5N-4	1308.8
R8MDC	Radix-8	7$\log_8$N-7	(24+2T)$\log_8$N	4.5N-8	2673.2
Proposed $\mathrm{R4^2SDF}$	Radix-4^2	$\log_{16}$N-1	(8+T)$\log_{16}$N	N-1	415.5

TABLE IV
HARDWARE UTILIZATION RATE COMPARISONS OF THE PIPELINED FFT/IFFT ARCHITECTURE

Pipeline architecture	Utilization rate of complex Mult.	Utilization rate of complex adders (including constant mult.)	Utilization rate of complex memory	Throughput (Output/Cycle)
R2SDF	50%	50%	100%	1
R4SDF	75%	25%	100%	1
R8SDF	87.5%	12.5%	100%	1
$\mathrm{R2^2SDF}$	75%	50%	100%	1
$\mathrm{R2^3SDF}$	87.5%	50%	100%	1
R2MDC	50%	50%	50%	2
$\mathrm{R2^2MDC}$	37.5%	50%	50%	2
R4MDC	25%	25%	25%	4
R8MDC	12.5%	12.5%	12.5%	8
Proposed $\mathrm{R4^2SDF}$	87.5%	56.25%	100%	1

Based on the implementation results in our process, we convert the area of each complex multiplier and complex memory to the 50 and 1.3 complex adder, respectively, when adopting 13-bit precision, and the scheme with three real multiplications

TABLE V
HARDWARE REQUIREMENT COMPARISON OF 8 × 8 2-D DCT ARCHITECTURE

8×8 DCT	Lee et al. [19] (parallel)	Chang & Wang [22] (2D systolic)	Hsiao and Shiue [20] (linear-array)	Ruetz et al. [21] (linear-array)	Madisetti et al. [23] (parallel MAC)	Proposed (R4^{2}SDF)
Real multipliers	28	64	-	-	-	-
Real adders	134	88	-	-	-	-
Complex multipliers	-	-	3	8	14	1
Complex adders	-	-	9	18	32	26
Twiddle factors realization	Hardwired Multiplier	Hardwired Multiplier	ROM based LUT	ROM based LUT	Hardwired Multiplier	Hardwired Multiplier & ROM based LUT
Total transistor count	~ 400 K	~ 340 K	~ 105 K	N/A	~ 67 K	~60 K
Hardware complexity	O(NlogN)	O(N^2)	O(logN)	O(logN)	O(log$_8$N)	O(log$_{16}$N)
Throughput (Output/cycle)	16	8	2	2	4	2
Internal Wordlength	18	16	16	14	22	13
Interconnect complexity	Complex	Simple	Moderate	Moderate	Simple	Moderate
FFT/IFFT/2-D DCT triple modes	No	No	No	No	No	Yes

[1] A gate count was determined and the number of transistors was determined by assuming four transistors per gate.

[2] An unknown gate count was indicated by "N/A".

and five real additions, in the implementation. The right-most column of Table III lists the area indexes of the equivalent adder of the 256-point FFT/IFFT architecture. Clearly, the proposed $\mathrm{R4^2SDF}$-based 256-point FFT/IFFT architecture has the lowest hardware requirements. The $\mathrm{R4^2SDF}$-based 256-point FFT/IFFT architecture has a 16% lower cost than the $\mathrm{R2^2SDF}$-based 256-point FFT/IFFT architecture. Significantly, the cost advantage of our proposed architecture becomes more evident when the transform length is larger. Thus, the proposed $\mathrm{R4^2SDF}$-based architecture has a lower hardware cost than $\mathrm{R2^2SDF}$ and other famous pipeline FFT/IFFT architecture in terms of the number of ROMs, complex multipliers, complex adders, constant multipliers, and shift registers.

Table IV shows the comprehensive comparison of the hardware utilization rate in terms of the utilization rate of complex multipliers, complex adders, and complex memory. Clearly, the proposed architecture achieves the highest complex multiplier utilization rate among pipeline architecture (87.5%). Additionally, the proposed architecture maintains the maximum complex memory utilization rate of 100%. Furthermore, the proposed architecture, including the constant multipliers, has the highest complex adder utilization rate of 56.25%. Thus, the purposed architecture achieves a higher hardware utilization rate than $\mathrm{R2^2SDF}$ and other well-known pipeline FFT/IFFT architecture in terms of the utilization rate of complex multipliers, complex adders, constant multipliers, and complex memory. Although the R2MDC, R4MDC, and R8MDC architectures have the higher throughput rate (output/cycle) of 2, 4, and 8 than SDF-based architecture, these approaches require large hardware requirement, such as complex multipliers, adders, and memory size, as shown in Table III. Therefore, this investigation focuses on the "hardware-oriented" architecture, in which the arithmetic operations can be tightly scheduled for efficient hardware utilization. This study demonstrates that the proposed $\mathrm{R4^2SDF}$-based pipeline FFT/IFFT architecture has the lowest hardware cost and highest hardware utilization. Conversely, the proposed $\mathrm{R4^2SDF}$-based pipeline FFT/IFFT architecture is the most cost-efficient.

B. 8 × 8 2-D DCT Comparison

Many DCT implementations exist spanning a broad spectrum of architectures, focusing on different applications. Lee *et al.* [19] presented a highly parallel approach with high arithmetic cost and high power consumption for the high-performance application. The systolic implementation of Lee *et al.* [19] employs the row-column decomposition to derive the configurable 2-D $N \times N$ DCT in three steps with each step implemented in systolic form. This work concentrates on high-speed FFT/IFFT/2D DCT architectures with a throughput rate of at least one output sample per cycle, targeted for applications in next-generation handheld devices needing a high data-processing rate. Moreover, the proposed architecture has high cost efficiency and low cost in a portable consumer device. This subsection lists the hardware requirement comparison between six different implementations in terms of the number of real (complex) multipliers, real (complex) adders, twiddle factors realization, total transistor count, hardware complexity, throughput, internal wordlength, interconnect complexity, and support for triple-mode, as shown in Table V. Clearly, the proposed pipeline $\mathrm{R4^2SDF}$-based FFT/IFFT/2D-DCT processor has the fewest complex multipliers and lowest hardware complexity, an acceptable throughput rate and moderate interconnect complexity. Although the number of the complex adders in the proposed processor is greater than the designs in [20] and [21], the total area including complex multiplier is still lower than others. The total number of transistors indicates that the proposed design achieves the smallest chip cost among architectures supporting FFT/IFFT mode.

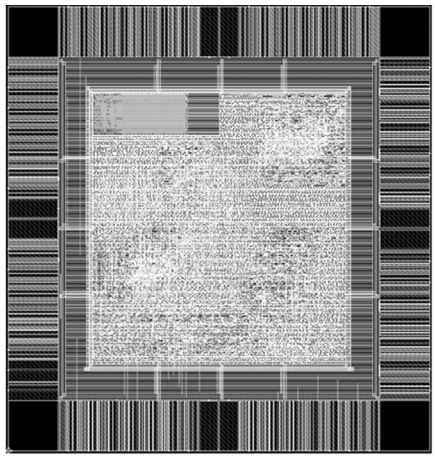
C. Chip Implementation

Following the functional verification in the MATLAB environment, the 256-point FFT/IFFT/2-D DCT architecture in which the internal word length of the entire design is 13-bit was synthesized by the Design Compiler with TSMC 0.13-μm CMOS technology. The floorplan and post-layout were performed by Astro. The post-simulation was issued by NC-Simulator to

TABLE VI
GATE COUNT USAGE OF EACH BUILDING BLOCKS

Categories	Control	Butterfly Cores	Complex Multiplier	Constant Multipliers	Shift Registers
Area	1.3 %	21.74 %	18.9 %	3.48 %	54.58 %

verify the functionality after back-annotation was performed from the Start-RC extractor. The static timing check can be signed-off by PrimeTime. Finally, the power analysis and DRC were conducted using Astro Rail and Dracula, respectively. The core area of the post layout was 0.6 mm^2. The reported equivalent gate count is 60 086 gates, which approaches 60-k gates. The gate count usage for each building block is listed in Table VI. It is obvious that 264 words shift register dominates the chip cost of 54.58%. The implementation result without the 2-D DCT indicates that the total gate count decreased to 55.2 k. The implementation reports in this study reveal that the routing cost penalty incurred by the additional 8 × 8 2-D DCT mode is small. The chip operated at 100 MHz, thus satisfying the high throughput requirement After the conversion, the proposed $R4^2SDF$ design in 8 × 8 2-D DCT mode could provide high frame rates of 505 and 1042 kf/s for frame sizes of 176 × 144 and 128 × 96 ($pixel^2$), respectively. Concerning the speed performance, because the pipelined multiplier operation is easy to design at a clock rate of 100 MHz or even higher, the proposed architecture can achieve a high clock rate by simple pipelining techniques for the involved arithmetic components. The chip properties shown in Fig. 10 demonstrate that the average power dissipation of the 256-point FFT/IFFT/2-D DCT design was 22.37 mW at 100 MHz at 1.2-V supply voltage. The layout view as shown in Fig. 10 has 64 I/O pins, of which eight pins are power supply pins. The proposed $R4^2SDF$-based 256-point FFT/IFFT/2-D DCT implementation not only satisfies the system performance of DV standards in 8 × 8 2-D DCT mode, but also achieves the satisfactory results with 40-dB performance in 256-point FFT/IFFT modes. Additionally, the proposed $R4^2SDF$ based 256-point FFT/IFFT/2-D DCT implementation has a low power consumption (22.37 mW), and the lowest hardware requirement of all pipeline architectures. These findings indicate that the proposed design is suitable for the highly cost-efficient FFT/IFFT/2-D DCT triple-mode RSoCs IP for next-generation handheld devices.



Mode Selection	256-point FFT/IFFT and 8×8 2D-DCT
Architecture	$R4^2SDF$ pipeline
Technology	0.13 μm CMOS
Core Size	807(μ m) x 754(μ m) = 0.6 mm^2
Power Consumption / Freq.	22.37 mW / 100 MHz
Accuracy / internal wordlength	40dB in DV standard / 13-bits
Input/Output/Power Pins #	29 / 27 / 8

Fig. 10. Layout view and design characteristics of proposed pipeline 256-point FFT/IFFT/8 × 8 2-D DCT processor.

VI. CONCLUSION

This investigation develops a triple-mode reconfigurable pipeline $R4^2SDF$ VLSI architecture that supports the 256-point FFT/IFFT and 8 × 8 2-D DCT computations. The comparison results demonstrate that the proposed $R4^2SDF$ pipeline FFT/IFFT architecture has a lower hardware cost and higher utilization than $R2^2SDF$ and other pipeline architectures. Following the fixed-point analysis the proposed 256-point FFT/IFFT/8 × 8 2-D DCT chip design is successfully implemented in 0.13-μm CMOS technology with an internal wordlength of 13 bits. This design has a power consumption of 22.37 mW at 100 MHz at 1.2-V supply voltage. These features ensure that the proposed reconfigurable processor design is certainly amenable to the next-generation mobile communications. The upcoming fourth-generation wireless system requires the simultaneous application of many computing algorithms including MPEG-4 AVC [24] and Walsh transform [25], in the same handheld device. The reconfigurable hardware core for supporting more transforms is a significant topic for future work.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their valuable suggestions to this paper.

REFERENCES

[1] *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE Standard 802.16-2004, 2004.
[2] E. Cornu, N. Destrez, A. Dufaux, H. Sheikhzqadeh, and R. Brennan, "An ultra low power, ultra miniature voice command system based on hidden markov models," in *Proc. IEEE Inter. Conf. Acoust., Speech, Signal Process.*, May 2002, vol. 4, pp. 3800–3803.
[3] S. M. Chai, S. Chiricescu, R. Essick, B. Lucas, P. May, K. Moat, J. M. Norris, and M. Schuette, "Streaming processors for next-generation mobile imaging application," *IEEE Commun. Mag.*, vol. 43, no. 12, pp. 81–89, Dec. 2005.
[4] R. K. Kolagotla, J. Fridman, M. M. Hoffiman, W. C. Anderson, B. C. Aldrich, D. B. Witt, M. S. Allen, R. R. Dunton, and L. A. Booth, "A 333-MHz dual-MAC DSP architecture for next-generation wireless application," in *Proc. IEEE Inter. Conf. Acoust., Speech, Signal Process.*, May 2001, vol. 2, pp. 1013–1016.
[5] M. Vorbach and J. Becker, "Reconfigurable processor architectures for mobile phones," in *Proc. IEEE Inter. Symp. Parallel Distrib. Process.*, Apr. 2003, p. 181.1.

[6] E. Tell, O. Seger, and D. Liu, "A converged hardware solution for FFT, DCT and Walsh transform," in *Proc. IEEE Inter. Symp. Signal Process. Its Appl.*, Jul. 2003, vol. 1, pp. 609–612.
[7] R. Storn, "Efficient input reordering for the DCT based on a real-valued decimation-in-time FFT," *IEEE Signal Process. Lett.*, vol. 3, no. 8, pp. 242–244, Aug. 1996.
[8] C. Diab, M. Oueidat, and R. Prost, "A new IDCT-DFT relationship reducing the IDCT computational cost," *IEEE Trans. Signal Process.*, vol. 50, no. 7, pp. 1681–1684, Jul. 2002.
[9] S. He and M. Torkelson, "Designing pipeline FFT processor for OFDM (de)modulation," in *Proc. URSI Int. Symp. Signals, Syst., Electron.*, 1998, pp. 257–262.
[10] C. S. Burrus, "Index mapping for multidimensional formulation of the DFT and convolution," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-25, no. 3, pp. 239–242, Jun. 1977.
[11] S. F. Hsiao, Y. H. Hu, T. B. Juang, and C. H. Lee, "Efficient VLSI implementations of fast multiplierless approximated DCT using parameterized hardware modules for silicion intellectual property design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 8, pp. 1568–1579, Aug. 2005.
[12] W.-H. Chang and T. Nguyen, "An OFDM-specified lossless FFT architecture," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 6, pp. 1235–1243, Jun. 2006.
[13] W. C. Yeh and C. W. Jen, "High-speed and low-power split-radix FFT," *IEEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 864–874, Mar. 2003.
[14] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. New York: Wiley, 1999.
[15] Sony, Tokyo, Japan, "DVCAM format overview," 2000 [Online]. Available: http://www.sony.ca/dvcm/brochures.htm
[16] A. Silva, P. Gouveia, and A. Navarro, "Fast multiplication-free QWDCT for DV coding standard," *IEEE Trans. Consumer Electron.*, vol. 50, no. 1, pp. 180–187, Feb. 2004.
[17] A. Ichigaya, M. Kurozumi, N. Hara, Y. Nishida, and E. Nakasu, "A method of estimating coding PSNR using quantized DCT coefficients," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 2, pp. 251–259, Feb. 2006.
[18] T. Sansaloni, A. Perez-Pascual, V. Torres, and J. Valls, "Efficient pipeline FFT processors for wireless LAN MIMI-OFDM systems," *Electron. Lett.*, vol. 41, no. 19, pp. 1043–1044, Sep. 2005.
[19] Y. P. Lee, T. H. Chen, L. G. Chen, M. J. Chen, and C. W. Ku, "A cost effective architecture for 8 × 8 two-dimensional DCT/IDCT using direct method," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 459–467, Jun. 1997.
[20] S. F. Hsiao and W. R. Shiue, "A new hardware-efficient algorithm and architecture for computation of 2-D DCTs on a linear array," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 11, pp. 1149–1159, Nov. 2001.
[21] P. A. Ruetz, P. Tong, D. Bailey, D. A. Luthi, and P. H. Ang, "A high performance full-motion video compression chip set," *IEEE Trans. Circuits Syste. Video Technol.*, vol. 2, no. 2, pp. 111–121, Jun. 1992.
[22] Y.-T. Chang and C.-L. Wang, "New systolic array implementation of the 2-D discrete cosine transform and its inverse," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 2, pp. 150–157, Apr. 1995.
[23] A. Madisetii and A. N. Willson, "A 100 MHz 2-D 8 × 8 DCT/IDCT processor for HDTV application," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 2, pp. 158–164, Apr. 1995.
[24] G. A. Jian, C. D. Chien, and J. I. Guo, "A memory-based hardware accelerator for real-time MPEG-4 audio coding and reverberation," in *Proc. IEEE Inter. Symp. Circuit Syst.*, May 2007, pp. 1569–1572.
[25] R. Pandey and M. L. Bushnell, "Architecture for variable-length combined FFT, DCT and MWT transform hardware for multi-modeWireless system," in *Proc. IEEE Inter. Conf. Embedded Syst.*, Jan. 2007, pp. 121–126.

Chin-Teng (CT) Lin (S'88–M'91–SM'99–F'05) received the B.S. degree in control engineering from National Chiao-Tung University (NCTU), Hsinchu, Taiwan, R.O.C., in 1996, and the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1989 and 1992, respectively.

Since August 1992, he has been with the College of Electrical Engineering and Computer Science, NCTU, where he is currently the University Provost and the Chair Professor of Electrical and Control Engineering. He served as the Founding Dean of Computer Science College of NCTU from 2005 to 2007. He is the author of *Neural Fuzzy Systems* (Prentice-Hall, 1996) and *Neural Fuzzy Control Systems with Structure and Parameter Learning* (World Scientific, 1994). He has published over 110 journal papers, including about 80 IEEE transaction papers. His research interests include intelligent technology, soft computing, brain-computer-interface, intelligent transportation systems, robotics and intelligent sensing, and NBIC (nano-bio-information technologies and cognitive science).

Dr. Lin was a recipient of the Outstanding Research Award granted by National Science Council (NSC), Taiwan, since 1997 to present, the Outstanding Professor Award granted by the Chinese Institute of Engineering (CIE) in 2000, and the 2002 Taiwan Outstanding Information-Technology Expert Award. He was prompted to IEEE Fellow in 2005 for contributions to biologically inspired information systems. He is a member of the Board of Governors (BoG) of the IEEE Systems, Man, Cybernetics Society (SMCS) from 2003 to 2005, and a current BoG member of IEEE Circuits and Systems Society (CASS). He has been the IEEE Distinguished Lecturer from 2003 to 2005. He also currently serves as the Deputy Editor-In-Cheif of the IEEE TRANSACTIONS OF CIRCUITS AND SYSTEMS—PART II: EXPRESS BRIEFS. He was the Program Chair of the 2006 IEEE International Conference on Systems, Man, and Cybernetics held in Taipei, Taiwan. He was the president of Board of Government (BoG) of Asia Pacific Neural Networks Assembly (APNNA) from 2004 to 2005. He was also elected to be one of 38th Ten Outstanding Rising Stars in Taiwan (2000). He is a member of Tau Beta Pi, Eta Kappa Nu, and Phi Kappa Phi honorary societies.

Yuan-Chu Yu (M'05) received the B.S. degree in automatic control engineering from the Feng-Chia University, Taichung, Taiwan, R.O.C., in 1995.

From 1999 to 2001, he was working as a Digital Circuit Engineer with the Acer Corporation, where his main field was related to multi-processor server system design. In 2001, he was working as a Digital IC Designer with Elan. Currently, he is an Assistant Manager in the PC peripherals line of digital IC development division. His main field in Elan was related to Microprocessor core design, DSP core design, 32-bit RISC core design, and SOC design. He is currently pursuing the Ph.D. degree with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsin-Chu, Taiwan, R.O.C. His current research interests include biomedical signal processing, digital IC design, and wireless network.

Lan-Da Van (S'98–M'02) was born in Miaoli, Taiwan, R.O.C., in 1972. He received the B.S. (Honors) and the M.S. degree from the Tatung Institute of Technology, Taipei, Taiwan, R.O.C., in 1995 and 1997, respectively, and the Ph.D. degree from the National Taiwan University (NTU), Taipei, Taiwan, R.O.C., in 2001, all in electrical engineering.

From 2001 to 2006, he was an Associate Researcher with the National Chip Implementation Center (CIC), Hsinchu, Taiwan. Since Feb. 2006, he joined the faculty of the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, where he is currently an Assistant Professor. His research interests include VLSI algorithms, architectures, and chips for digital signal processing (DSP), baseband communication systems, and computer applications. This includes the design of high-performance/low-power/area-aware DSP processors, adaptive filters, transform, computer arithmetic, and platform-based system-on-a-chip (SOC) designs. He has published over 20 IEEE journal and conference papers in these areas.

Dr. Van was a recipient of the Chunghwa Picture Tube (CPT) and Motorola fellowships in 1996 and 1997, respectively. He was an elected chairman of IEEE NTU Student Branch in 2000. In 2002, he has received the IEEE award for outstanding leadership and service to the IEEE NTU Student Branch. From 2003 to 2006, he has been listed in Marquis Who's Who in Science and Engineering. In 2005, he was a recipient of the Best Poster Award at iNEER Conference for Engineering Education and Research (iCEER). In 2006, he will be listed in Marquis Who's Who in the World. Presently, he serves as a reviewer for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTION ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE SIGNAL PROCESSING LETTERS, *Elsevier Microelectronics Journal*, *Elsevier Integration-the VLSI Journal*, and *Eurasip Journal on Applied Signal Processing*.