NEW UNIFIED VLSI ARCHITECTURES FOR COMPUTING DFT AND OTHER TRANSFORMS[1]

Shen-Fu Hsiao   and   Chung-Yi Yen

Institute of Computer and Information Engineering
National Sun Yat-Sen University, Taiwan
sfhsiao@cie.nsysu.edu.tw

ABSTRACT

Fast computation of DFT and other popular transforms is essential in high-speed DSP applications. This paper proposes new architectures with low hardware cost and high throughput rate. The new architectures are very suitable for VLSI implementation since they are very regular and require much fewer complex multipliers compared to the recently proposed approaches. Furthermore, the same architectures may be exploited to compute a variety of frequently-used transforms.

1. INTRODUCTION

Signal transform has been one of the fundamental operations in DSP. Among the popular transforms are discrete Fourier transform (DFT), discrete cosine transform (DCT), discrete sine transform (DST), discrete Hartley transform (DHT) and Walsh-Hadamard transform (WHT). There have been many papers on the fast implementation of these transforms in order to achieve real-time processing speed. This paper presents other novel and regular DFT architectures which have the advantages of low hardware cost and high throughput rate. These new FFT architectures require only one-third of complex multipliers compared to the recently proposed systolic DFT architecture in [1]. The cycle time is also reduced to that for one complex multiplication instead of one complex multiplication-addition as required in [1]. The new architectures are also extended to radix-4 versions where only one (two) complex multipliers are needed to compute DFT of length 8 (64). Furthermore, by adding suitable pre-processing unit and post-processing unit, other frequently used transforms such as DCT, DST, DHT, WHT and their inverses can also be realized on the same architectures.

2. NEW SYSTOLIC DFT ARCHITECTURES

The DFT of length $N = 2^n$ defined as

$$X_n = \sum_{k=0}^{N-1} x_k w_N^{nk} \quad , n = 0,1,2,....,N-1$$

$$w_N^{nk} = e^{-j\frac{2\pi}{N}nk} = \cos\frac{2\pi nk}{N} - j\sin\frac{2\pi nk}{N}$$

can be represented as a matrix-vector product form $X = Wx$ where W is the coefficient matrix. Take $N=8$ as an example. By bit-reversing the indices of the elements in X, the 8 x 8 coefficient matrix W can be decomposed into product of thre coefficient submatrices $W = W_3 W_2 W_1$ with each submatrix W_i containing only three non-zero diagonals (one right-diagonal W_i^D, one upper-diagonal W_i^U and one lower-diagonal W_i^L):

$$
\begin{bmatrix} X_0 \\ X_4 \\ X_2 \\ X_6 \\ X_1 \\ X_5 \\ X_3 \\ X_7 \end{bmatrix}
=
\begin{bmatrix}
w_8^0 & w_8^0 & 0 & 0 & 0 & 0 & 0 & 0 \\
w_8^0 & w_8^4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & w_8^0 & w_8^0 & 0 & 0 & 0 & 0 \\
0 & 0 & w_8^0 & w_8^4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & w_8^0 & w_8^0 & 0 & 0 \\
0 & 0 & 0 & 0 & w_8^0 & w_8^4 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & w_8^0 & w_8^0 \\
0 & 0 & 0 & 0 & 0 & 0 & w_8^0 & w_8^4
\end{bmatrix}
\begin{bmatrix}
w_8^0 & 0 & w_8^0 & 0 & 0 & 0 & 0 & 0 \\
0 & w_8^0 & 0 & w_8^0 & 0 & 0 & 0 & 0 \\
w_8^0 & 0 & w_8^4 & 0 & 0 & 0 & 0 & 0 \\
0 & w_8^2 & 0 & w_8^6 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & w_8^0 & 0 & w_8^0 & 0 \\
0 & 0 & 0 & 0 & 0 & w_8^0 & 0 & w_8^0 \\
0 & 0 & 0 & 0 & w_8^0 & 0 & w_8^4 & 0 \\
0 & 0 & 0 & 0 & 0 & w_8^2 & 0 & w_8^6
\end{bmatrix}
$$

(stage 3)   (stage 2)

$$
\begin{bmatrix}
w_8^0 & 0 & 0 & 0 & w_8^0 & 0 & 0 & 0 \\
0 & w_8^0 & 0 & 0 & 0 & w_8^0 & 0 & 0 \\
0 & 0 & w_8^0 & 0 & 0 & 0 & w_8^0 & 0 \\
0 & 0 & 0 & w_8^0 & 0 & 0 & 0 & w_8^0 \\
w_8^0 & 0 & 0 & 0 & w_8^4 & 0 & 0 & 0 \\
0 & w_8^1 & 0 & 0 & 0 & w_8^5 & 0 & 0 \\
0 & 0 & w_8^2 & 0 & 0 & 0 & w_8^6 & 0 \\
0 & 0 & 0 & w_8^3 & 0 & 0 & 0 & w_8^7
\end{bmatrix}
\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}
\qquad (1)
$$

(stage 1)

For example, the three coefficient diagonal vectors for W_1 are

$$W_1^U = [0,0,w_8^0,w_8^0,w_8^0,w_8^0,0,0]$$

$$W_1^D = [w_8^0,w_8^0,w_8^0,w_8^0,w_8^4,w_8^5,w_8^6,w_8^7]$$

$$W_1^U = [0,0,w_8^0,w_8^1,w_8^2,w_8^3,0,0]$$

The signal flow graph for the 8-pt. DFT based on the decomposition in eqn. (1) is shown in Fig. 1.

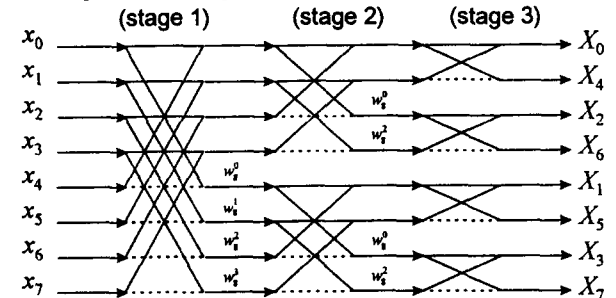Figure 1: Signal flow graph for 8-pt. DFT.

From the above coefficient matrix decomposition, several

615

DFT architectures can be derived including the radix-2 pipelined FFT [5] and the Boriakoff systolic DFT [1]. For example, using the systolic implementation of banded matrix-vector multiplication in [4], the coefficient matrix decomposition in eqn. (1) can be realized by the systolic structure in Fig. 2.
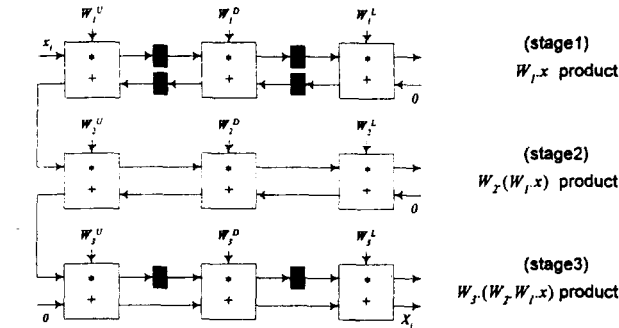
Figure 2: Systolic DFT architecture for N=8.

This structure can be transformed into another recently proposed Boriakoff systolic DFT architecture in [1] by proper retiming.

In fact, the above structure can be further simplified by observing that the elements of W_i^U are either 1 or 0, and that $w_N^k * x_i \pm w_N^{N-k} * x_j = w_N^k * (x_i \mp x_j)$ with elements w_N^k from W_i^L and w_N^{N-k} from W_i^D. The simplified architecture shown in Fig. 3 reduces the number of complex multipliers from $3\log_2 N$ to $\log_2 N - 1$ (N=8 in this case), and thus significantly reduces the total power consumption. The cycle time is also reduced to that for one complex multiplication instead of one complex multiplication-addition.
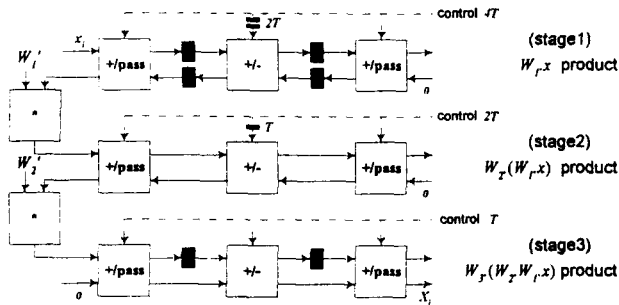
Figure 3: New systolic DFT (N=8) architecture with reduced number of complex multipliers.

3. NEW RADIX-4 SYSTOLIC DFT

The same idea in Sec. 2 can be applied to generate a general radix-4 DFT structure computing DFT of length $N = 2^{2k+1}$ if every two tridiagonal coefficient submatrices are combined into another submatrix with seven nonzero diagonals. For example, by combining the two tridiagonal coefficient submatrices W_2, W_1 into another submatrices $W_{1,2}$ with seven nonzero diagonals, another

decomposition is obtained as for the 8-pt. DFT:

Furthermore, observing that

$$w_N^k x_1 + w_N^{k+\frac{N}{4}} x_2 + w_N^{k+\frac{N}{2}} x_3 + w_N^{k+\frac{3N}{4}} x_4 = w_N^k (x_1 - jx_2 - x_3 + jx_4)$$

, another radix-4 version of the 8-pt DFT architecture is generated in Fig. 4 which requires only one complex multiplier and seven complex adders/subtractors. There are six different operations performed in PEs: ⌐+/-⌐ denotes complex addition/subtraction; ⌐j/-j⌐ denotes multiplication of j/-j followed by complex addition; ⌐p⌐ denotes pure data passing; ⌐*⌐ denotes complex multiplication.
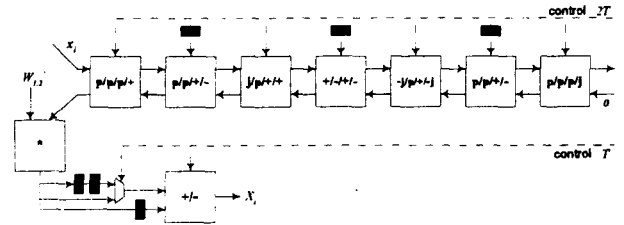
Figure 4: Radix-4 version of the systolic DFT structure for N=8 with only one complex multiplier.

Note that the last coefficient matrix W_{2k+1} can always be implemented by a single PE containing only complex addition/subtraction.

Similarly, radix-4 systolic DFT architectures exist for $N = 2^{2k}$ if every neighboring two submatrices of the $2k$ submatrices are combined into one submatrix with seven diagonals. Fig. 5 shows a radix-4 systolic DFT architecture for N=64 where only two complex multipliers are needed.
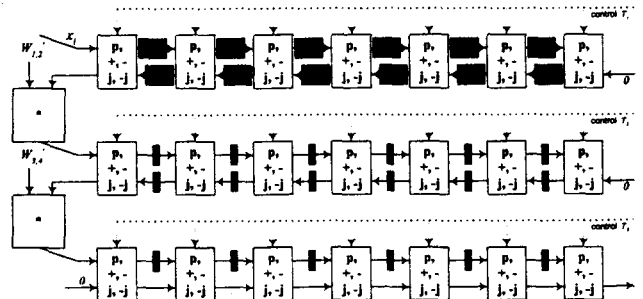
Figure 5: Radix-4 version of the systolic DFT structure for N=64 consisting of only two complex multipliers.

Many systolic implementations, such as those in [2][3], require $O(N)$ arithmetic processors to compute DFT of length N while ours call for only $O(logN)$ processors, a great saving in hardware cost. Tab. 1 compares the speed and hardware performance of our $O(logN)$ DFT architectures (both radix-2 and radix-4 versions) with the Boriakoff systolic DFT structures [1]. Our proposed FFT improves significantly the hardware cost by reducing the number of multipliers by 2/3. Such improvement is due to the observation of the distribution law $c*x_i \pm c*x_j = c*(x_i \pm x_j)$ for entries in rows of the coefficient submatrices.

	# of complex multipliers	# of complex adders	# of registers
Boriakoff FFT	$3\log_2N$	$3\log_2N$	$2N-4\log_2N+2$
proposed new radix-2 FFT	$\log_2N-1$	$3\log_2N-2$	$2N-4\log_2N+2$
proposed new radix-4 FFT	$\log_4N-1$	$7\log_4N$	$2N-12\log_4N+10$

Table 1: Comparison of several $O(logN)$ DFT structures.

4. UNIFIED TRANSFORM STRUCTURE

In this section, we will show that the coefficient matrices of other popular transforms such as DCT, DST, DHT, WHT, and their inverses can be decomposed into product of several tridiagonal coefficient submatrices with the same structure as those in eqn. (1) except that a pre- or post- permutation matrix are needed. Thus, the systolic DFT structures proposed in the previous sections can be directly exploited to implement these transforms.

Take as an example the 8-pt DST which can be expressed as $\bar{X} = \frac{1}{4}S_{post}S_3S_2S_1\bar{x}$:

$$
\begin{bmatrix} X_1 \\ X_5 \\ X_3 \\ X_7 \\ X_2 \\ X_6 \\ X_4 \\ X_8 \end{bmatrix} = \frac{1}{4}
\begin{bmatrix} 1 & 0 & 0 & 0 & & & & \\ 1 & 1 & 1 & 0 & & & & \\ 1 & 0 & 1 & 0 & & & & \\ 1 & 1 & 0 & 1 & & & & \\ & & & & 1 & 0 & 0 & 0 \\ & & & & 1 & 1 & 0 & 0 \\ & & & & 0 & 0 & 1 & 0 \\ & & & & 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} c_4 & c_4 & & & & & & \\ 1 & -1 & & & & & & \\ & & c_4 & c_4 & & & & \\ & & 1 & -1 & & & & \\ & & & & c_4 & c_4 & & \\ & & & & 1 & -1 & & \\ & & & & & & c_4 & c_4 \\ & & & & & & c_4 & -c_4 \end{bmatrix}
$$
$$
\begin{bmatrix} c_2 & & c_2 & & & & & \\ & c_6 & & c_6 & & & & \\ 1 & & -1 & & & & & \\ & 1 & & -1 & & & & \\ & & & & c_2 & & c_2 & \\ & & & & & c_6 & & c_6 \\ & & & & 1 & & -1 & \\ & & & & & 1 & & -1 \end{bmatrix}
\begin{bmatrix} c_1 & & & & c_1 & & & \\ & c_3 & & & & c_3 & & \\ & & c_7 & & & & c_7 & \\ & & & c_5 & & & & c_5 \\ 1 & & & & -1 & & & \\ & 1 & & & & -1 & & \\ & & 1 & & & & -1 & \\ & & & 1 & & & & -1 \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_4 \\ x_3 \\ x_8 \\ x_7 \\ x_5 \\ x_6 \end{bmatrix} \quad (2)
$$

with $c_i = \dfrac{1}{2\cos(i\pi/16)}$. Note that the coefficient

submatrices S_i have the same structure as W_i in eqn. (1). The signal flow graph for the 8-pt DST is shown in Fig. 6. The distribution law $c*x_i \pm c*x_j = c*(x_i \pm x_j)$ can be applied to all the submatrices S_i. Thus, all the systolic structures mentioned above can be used to implement the DST if the coefficient ROMs are changed to that for DST and a final stage implementing the post-scaling submatrix S_{post} is added.
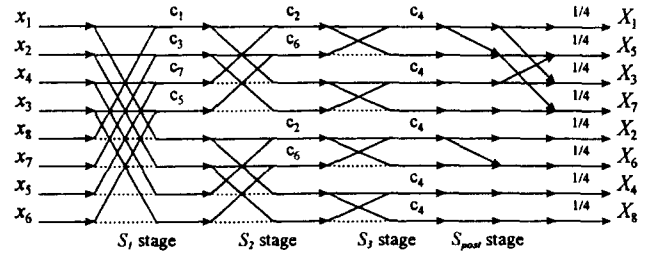

Figure 6: Signal flow graph for the 8-pt DST. Note the similarity of the first three stages to that in Fig. 1.

The inverse DST (IDST) is equivalent to the transpose of the DST and can be written as

$$[IDST] = [DST]^T$$
$$\therefore \bar{x} = ([DST_{Post}][DST_n].....[DST_3][DST_2][DST_1])^T.\bar{X}$$
$$= ([DST_1]^T[DST_2]^T[DST_3]^T.....[DST_n]^T[DST_{Post}]^T).\bar{X}$$

Thus, the 8-pt IDST follows by taking the transpose of the DST decomposition in eqn. (2).

$$
\begin{bmatrix} x_1 \\ x_2 \\ x_4 \\ x_3 \\ x_8 \\ x_7 \\ x_5 \\ x_6 \end{bmatrix} =
\begin{bmatrix} c_1 & & & & 1 & & & \\ & c_3 & & & & 1 & & \\ & & c_7 & & & & 1 & \\ & & & c_5 & & & & 1 \\ c_1 & & & & -1 & & & \\ & c_3 & & & & -1 & & \\ & & c_7 & & & & -1 & \\ & & & c_5 & & & & -1 \end{bmatrix}
\begin{bmatrix} c_2 & & 1 & & & & & \\ & c_6 & & 1 & & & & \\ c_2 & & -1 & & & & & \\ & c_6 & & -1 & & & & \\ & & & & c_2 & & 1 & \\ & & & & & c_6 & & 1 \\ & & & & c_2 & & -1 & \\ & & & & & c_6 & & -1 \end{bmatrix}
$$
$$
\begin{bmatrix} c_4 & 1 & & & & & & \\ c_4 & -1 & & & & & & \\ & & c_4 & 1 & & & & \\ & & c_4 & -1 & & & & \\ & & & & c_4 & 1 & & \\ & & & & c_4 & -1 & & \\ & & & & & & c_4 & c_4 \\ & & & & & & c_4 & -c_4 \end{bmatrix}
\begin{bmatrix} 1 & 1 & 1 & 1 & & & & \\ 0 & 1 & 0 & 1 & & & & \\ 0 & 1 & 1 & 0 & & & & \\ 0 & 0 & 0 & 1 & & & & \\ & & & & 1 & 1 & 0 & 0 \\ & & & & 0 & 1 & 0 & 0 \\ & & & & 0 & 0 & 1 & 0 \\ & & & & 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} X_1 \\ X_5 \\ X_3 \\ X_7 \\ X_2 \\ X_6 \\ X_4 \\ X_8 \end{bmatrix} \quad (3)
$$

However the transposed coefficient submatrices S_i^T do not have the structure as S_i or W_i. Thus the IDST can not be directly implemented using the structures mentioned in the previous sections, and some modifications are necessary. First, through reordering of the output data and data permutation, we obtain

$$\begin{bmatrix} x_1 \\ x_8 \\ x_4 \\ x_5 \\ x_2 \\ x_7 \\ x_3 \\ x_6 \end{bmatrix} = \begin{bmatrix} c_1 & 1 & & & & & & \\ c_1 & -1 & & & & & & \\ & & c_7 & 1 & & & & \\ & & c_7 & -1 & & & & \\ & & & & c_3 & 1 & & \\ & & & & c_3 & -1 & & \\ & & & & & & c_5 & 1 \\ & & & & & & c_5 & -1 \end{bmatrix} \begin{bmatrix} c_2 & & 1 & & & & & \\ & c_2 & & 1 & & & & \\ c_2 & & -1 & & & & & \\ & c_2 & & -1 & & & & \\ & & & & c_6 & & 1 & \\ & & & & & c_6 & & 1 \\ & & & & c_6 & & -1 & \\ & & & & & c_6 & & -1 \end{bmatrix}$$

$$\begin{bmatrix} c_4 & & & 1 & & & & \\ & c_4 & & & 1 & & & \\ & & c_4 & & & 1 & & \\ & & & c_4 & & & c_4 & \\ c_4 & & & -1 & & & & \\ & c_4 & & & -1 & & & \\ & & c_4 & & & -1 & & \\ & & & c_4 & & & -c_4 & \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & & & & \\ & & & & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & & & & \\ & & & & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & & & & \\ & & & & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & & & & \\ & & & & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_5 \\ X_3 \\ X_7 \\ X_2 \\ X_6 \\ X_4 \\ X_8 \end{bmatrix}$$

Then, the combination law $\underline{\mathbf{ABC}} = \mathbf{A}\underline{\mathbf{BC}}$ is applied. For example:

$$\begin{bmatrix} c_1 & 1 & & \\ c_1 & -1 & & \\ & & c_7 & 1 \\ & & c_7 & -1 \end{bmatrix} \begin{bmatrix} c_2 & 1 & & \\ & c_2 & 1 & \\ c_2 & -1 & & \\ & c_2 & -1 \end{bmatrix} \begin{bmatrix} c_4 & & \\ & c_4 & \\ & & c_4 \\ & & & c_4 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & & \\ 1 & -1 & & \\ & & 1 & 1 \\ & & 1 & -1 \end{bmatrix} \begin{bmatrix} c_1 & & \\ & 1 & \\ & & c_7 \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & & \\ 1 & 1 & & \\ 1 & -1 & & \\ 1 & -1 \end{bmatrix} \begin{bmatrix} c_2 & & \\ & c_2 & \\ & & 1 \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \\ & & & 1 \end{bmatrix} \begin{bmatrix} c_4 & & \\ & c_4 & \\ & & c_4 \\ & & & c_4 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & & \\ 1 & -1 & & \\ & & 1 & 1 \\ & & 1 & -1 \end{bmatrix} \begin{bmatrix} c_1 & c_1 & & \\ & 1 & 1 \\ c_7 & -c_7 & \\ 1 & -1 \end{bmatrix} \begin{bmatrix} c_2 & & \\ & c_2 & \\ & & 1 \\ & & & 1 \end{bmatrix} \begin{bmatrix} c_4 & & \\ & c_4 & \\ & & c_4 \\ & & & c_4 \end{bmatrix}$$

After such processing, the IDST can be rewritten as:

$$\begin{bmatrix} x_1 \\ x_8 \\ x_4 \\ x_5 \\ x_2 \\ x_7 \\ x_3 \\ x_6 \end{bmatrix} = \begin{bmatrix} 1 & 1 & & & & & & \\ 1 & -1 & & & & & & \\ & & 1 & 1 & & & & \\ & & 1 & -1 & & & & \\ & & & & 1 & 1 & & \\ & & & & 1 & -1 & & \\ & & & & & & 1 & 1 \\ & & & & & & 1 & -1 \end{bmatrix} \begin{bmatrix} c_1 & c_1 & & & & & & \\ & 1 & 1 & & & & & \\ c_7 & -c_7 & & & & & & \\ & 1 & -1 & & & & & \\ & & & c_3 & c_3 & & & \\ & & & & 1 & 1 & & \\ & & & c_5 & -c_5 & & & \\ & & & & 1 & -1 \end{bmatrix} \begin{bmatrix} c_2 & & c_2 & & & & & \\ & c_2 & & c_2 & & & & \\ & & 1 & & & & 1 & \\ & & & 1 & & & & 1 \\ & & & & c_6 & & -c_6 & \\ & & & & & c_6 & & -c_6 \\ & & & & & & 1 & -1 \\ & & & & & & & 1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} c_4 & & & & & & & \\ & c_4 & & & & & & \\ & & c_4 & & & & & \\ & & & c_4 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & c_4 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & & & & \\ & & & & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & & & & \\ & & & & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & & & & \\ & & & & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & & & & \\ & & & & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_5 \\ X_3 \\ X_7 \\ X_2 \\ X_6 \\ X_4 \\ X_8 \end{bmatrix} \quad (4)$$

where the last two submatrices denote the pre-scaling required in the computation of the IDST. The other three submatrices now have the same structures as W_i and thus can be realized using the systolic architectures mentioned in the previous sections. Fig. 7 shows the signal flow graph for the 8-pt IDST in eqn. (4).
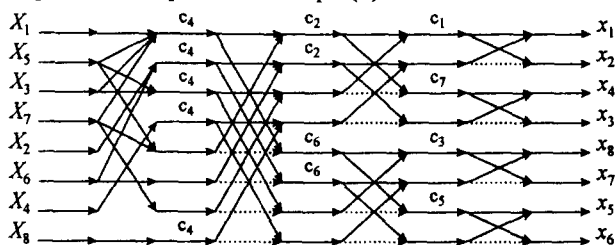

Figure 7: Signal flow graph for the IDST. Note that the last three stages have the same structure as Fig. 1.

Similar decomposition can be derived for coefficient matrices of DCT, DHT, WHT and their inverses. Thus, a unified architecture for computing various transforms are generated as shown in Fig. 8 for $N=8$. The central unit can be any one of the architectures proposed in the previous sections with different coefficient inputs. In this figure, the low hardware-cost systolic DFT in Fig. 3 is used. The data converters, the pre- and post-processor perform the necessary data ordering, pre- and post- operations for different transforms.
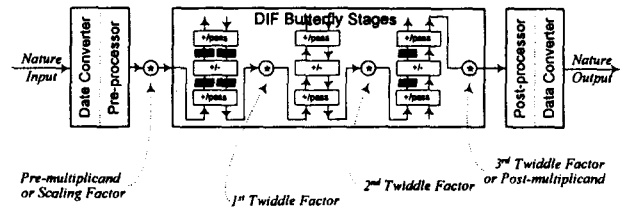

Figure 8: A unified structure for the computation of various popular transforms and their inverses.

5. CONCLUSION

Several new DFT structures have been presented by decomposing the coefficient matrix into submatrices with several nonzero diagonals and by realizing the submatrix-vector product using systolic arrays. Further hardware reduction is possible by applying the distribution law to the data entries in the rows of the coefficient submatrices. Radix-4 version of the systolic DFT structures are also proposed. Moreover, it is possible to implement various popular transforms and their inverses on the same DFT structures if the pre- and post-processing units are added.

Reference:

[1] V. Boriakoff, *"FFT Computation with Systolic Arrays, A New Architecture"*, IEEE Trans. Circuits and Systems, pp. 278-284, Apr. 1994.

[2] J. A. Beraldin *et. al.*, *"Efficient 1-D Systolic Array Realization of the Discrete Fourier Transform"*, IEEE Trans. Circ. and Syst., pp. 95-100, 1989.

[3] G. E. Bridges *et. al.*, *"Dual Systolic Architectures for VLSI Digital Signal Processing Systems"*, IEEE Trans. Comp., pp. 916-923, 1986.

[4] C. Mead and L. Conway, *"Introduction to VLSI Systems"*, Section 8.3, by H. T. Kung and C. E. Leiserson, Addison-Wesley Pub., 1980.

[5] L. R. Rabiner and B. Gold, *"Theory and Application of Digital Signal Processing"*, Prentice-Hall, 1975.