Abstract—A reduced-complexity algorithm is presented for computation of the discrete Fourier transform, where \( N \)-point transform is computed from eight number of nearly \((N/8)\)-point circular-convolution-like operations. A systolic architecture is also derived for VLSI implementation of the proposed algorithm. The proposed architecture is fully-pipelined and contains regular and simple locally-connected processing elements. It is devoid of complex control structure and scalable for higher transform lengths. It is observed that the proposed systolic structure involves either less or nearly the same hardware-complexity compared with the corresponding existing systolic structures. Besides, it offers eight times more throughput and significantly low latency compared with the others.

Index Terms—discrete Fourier transform, systolic array, VLSI, digital signal processing chip.

I. INTRODUCTION

T
he discrete Fourier transform (DFT) plays a key role in various digital signal processing and image processing applications [1]. Not only it is frequently encountered in many different applications, but also it is computation-intensive. Several algorithms are, therefore, suggested by researchers for fast computation of the DFT in general-purpose-programmable computers [2-3]. The general-purpose-computers, however, very often do not meet the speed-requirement of real-time signal- and image processing applications, and the size-constraints of a wide range of embedded systems. Algorithms and architectures are, thus, suggested for fast computation of one- and two-dimensional DFT in dedicated VLSI [4, 5]. Systolic designs represent a popular architectural paradigm for efficient VLSI implementation of computation-intensive digital signal processing (DSP) applications not only due to the simplicity of their design using repetitive identical processing elements (PE) having regular and local interconnections, but also for their potential of using high level of pipelining in small chip-area with low power-consumption inherent with the structure [6]. The radix-2 fast Fourier transform (FFT)-based structures [7, 8] have time-complexity \( O(N) \) and involve \( O(\log N) \) multiply-accumulators along with \( O(N + \log_2 N) \) delay-elements for \( N \)-point transform, but they can be used only for power-of-two transform-lengths. The radix-4 FFT-based structure of [9] offers still lower hardware-complexity over the radix-2 FFT-based structures, but can be used for power-of-4 transform-lengths only. The FFT-based structures [7-9] need to perform less number of arithmetic operations, but due to their irregular signal-flow graphs, they involve complicated routing, which leads to lower VLSI packing density, more power consumption and more design-time. It is also observed further that the algorithms optimized for software-implementation, in general, are not well-suited for dedicated hardware-implementation.

Prime-factor approach [10-13] using linear systolic arrays offers attractive solutions with hardware- and time-complexity \( O(N^{1/2}) \) for composite transform-length \( N = N_1 \times N_2 \), where \( N_1 \) and \( N_2 \) are relatively prime and \( N_1 \approx N_2 \). The structures of [14-16] also offer reduced-hardware and high-throughput solutions by decomposition of the computation for implementation in linear systolic arrays for DFT of smaller transform-lengths. It is, therefore, important to reduce the hardware- and time-complexities of the linear systolic arrays for computing the DFT so as to use them as optimal building blocks for the structures based on various decomposition schemes. Many different linear systolic arrays are reported in the literature [17-21], where the architectures differ mainly in terms of their area-complexity, time-complexity and I/O requirement. Some of the existing systolic arrays [22-27] for the DFT and other similar transforms are based on circular-convolution structure, which offer remarkable advantage over the others, particularly for efficient input/output and data transfer operations. In this paper, we present a high-throughput low-latency reduced-hardware systolization of the DFT using a low-complexity convolution-like formulation [21].

In the next Section, we have derived the proposed convolutional form of the algorithm for systolic realization of the DFT. An example of such formulation of the algorithm is discussed in Section-II, and development of the proposed systolic structure is described in Section-III. The hardware- and time-complexities of the proposed structure is estimated and compared with the corresponding existing structures in Section-IV. Conclusion along with the scope of future work is presented at the end.

II. FORMULATION OF THE PROPOSED ALGORITHM

In this Section, we derive a reduced-complexity algorithm for computation of the DFT, and then we convert the summation terms of the algorithm in to circular-convolution-like form.

A. Derivation of a Reduced Complexity Algorithm

The DFT of a sequence \( \{y(n), \text{ for } n = 0, 1, 2, \ldots, N-1\} \) may be given by

\[
X(k) = A(k) - jB(k) \text{ for } 0 \leq k \leq N - 1, \tag{1}
\]

where

\[
A(k) = \sum_{n=0}^{N-1} y(n) \cos[4\pi k n/N], \tag{2a}
\]

Copyright (c) 2006 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions.org.
When $N$ is even, one can reduce (2) into sum of $\{(N/2)\}$ terms as:

$$A(k) = \sum_{n=0}^{N/2} a(n) \cos[4\alpha_N kn], \quad \text{and} \quad B(k) = \sum_{n=0}^{N/2} b(n) \sin[4\alpha_N kn], \quad \text{for } 0 \leq k \leq N - 1,$$

where

$$a(n) = y(n) + y(N - n), \quad \text{and} \quad b(n) = y(n) - y(N - n) \quad \text{for } 1 \leq n \leq (N/2) - 1.$$

From (8), it is easy to find that all the $N$ DFT-components can be obtained from $(N/4+1)$ components of $A_i(k)$, $B_i(k)$, $A_{5i}(k)$, and $B_{5i}(k)$ for $0 \leq k \leq (N/4)$.

### B. Conversion into Circular-Convolution-like Form

To obtain the desired pair of circular-convolution-like form of $A_i(k)$ for $1 \leq k \leq (M - 1)$, when $M$ is prime, let us write the even and odd components in (6a), respectively, as

$$A_i(2k) = a_i(0) + a_i(M) + S(k), \quad \text{and} \quad A_i(2k - 1) = a_i(0) - a_i(M) + D(k),$$

for $1 \leq k \leq (M - 1)/2$.

$$S(\zeta) = \sum_{n=1}^{(M-1)/2} \{ \text{sign}_S(\zeta, n) \} s(n) \cos[\pi \nu_M(\zeta, n)],$$

$$D(\zeta) = \sum_{n=1}^{(M-1)/2} \{ \text{sign}_D(\zeta, n) \} d(n) \cos[\pi \nu_M(\zeta, n)].$$

$$\nu_M(\zeta, n) = \left\lfloor \frac{(2\zeta - 1)n}{M} \right\rfloor + \frac{1}{2} \quad \text{if } (2\zeta - 1)n < M/2,$$

$$\left\lfloor \frac{(2\zeta - 1)n}{M} \right\rfloor + \frac{1}{2} \quad \text{if } (2\zeta - 1)n > M/2$$

for $1 \leq \zeta, n \leq (M - 1)/2$.

When $M$ is prime, each of the sequences $\{S(\zeta)\}$ and $\{D(\zeta)\}$ for $1 \leq \zeta, \zeta \leq (M - 1)/2$ in (10a) and (10b), respectively, can be converted in to two $((M - 1)/2) - \pi$ circular-convolution structures (except the sign-factor inside the curly brackets) by suitable permutations achieved by mappings the index $n$ to $m$; and the indices $\zeta$ and $\xi$ to $l$, respectively, according to the following equations:

$$n = \left\lfloor \frac{(\eta^m - m)}{M} \right\rfloor \quad \text{if } (\eta^m - m) < M/2,$$

$$n = \left\lfloor \frac{(\eta^m - m)}{M} \right\rfloor + \frac{1}{2} \quad \text{if } (\eta^m - m) > M/2$$

for $m = 0, 1, 2, \ldots, [(M - 1)/2]-1$, and

$$k = (\eta^l), \quad \text{for } l = 0, 1, 2, \ldots, M-2,$$

for even values of $k$, and

$$k = (\xi + 1)/2 \quad \text{for odd values of } \xi.$$
\( \eta \) is the \((M-1)\)-th primitive root of unity, such that
\[
(\eta^{M-1})_M = 1
\]
and \((\eta^j)_M \neq 1\) for \(0 < j < (M-1)\).

Similarly, \(B_1(k)\) for \(1 \leq k \leq (M - 1)/2\) of (6c) can also be converted in to two circular-convolution-like structures. It can be noticed, further, that \(A_2(k)\), and \(B_2(k)\) of (6b) and (6d) are, respectively, the discrete cosine transform (DCT) and the discrete sine transform (DST) of \(M\)–point sequences \(\{a_2(n)\}\) and \(\{b_2(n)\}\). Following the methods in [24-26] the DCT and the DST can be also converted in to four number of \([(M-1)/2]\)-point circular-convolution-like structures. The core computation of the DFT, given by 4 matrix-vector multiplications of (6a)-(6d) can, thus, be converted into 8 number of circular-convolution-like structures of length \([(M-1)/2]\)–1.

III. EXAMPLE OF CONVERSION INTO CONVOLUTION FORM

For simple illustration of the conversion of \(A_1(k), B_1(k), A_2(k),\) and \(B_2(k)\) in to the desired convolutional-form, we show the conversion of \(A_1(k)\) for \(M=7\). We can write (10a) and (10b), respectively, for \(M=7\) as
\[
\begin{bmatrix}
S(1)
\end{bmatrix} = \begin{bmatrix}
\cos(2 \beta) & -\cos(3 \beta) & -\cos(\beta)
\end{bmatrix}
\begin{bmatrix}
s(1)
\end{bmatrix}
\]
\[
\begin{bmatrix}
S(2)
\end{bmatrix} = \begin{bmatrix}
-\cos(3 \beta) & -\cos(\beta) & \cos(2 \beta)
\end{bmatrix}
\begin{bmatrix}
s(2)
\end{bmatrix}
\]
\[
\begin{bmatrix}
S(3)
\end{bmatrix} = \begin{bmatrix}
-\cos(\beta) & \cos(2 \beta) & -\cos(3 \beta)
\end{bmatrix}
\begin{bmatrix}
s(3)
\end{bmatrix}
\]
\]
and
\[
\begin{bmatrix}
D(1)
\end{bmatrix} = \begin{bmatrix}
\cos(\beta) & \cos(2 \beta) & \cos(3 \beta)
\end{bmatrix}
\begin{bmatrix}
d(1)
\end{bmatrix}
\]
\[
\begin{bmatrix}
D(2)
\end{bmatrix} = \begin{bmatrix}
\cos(3 \beta) & -\cos(\beta) & -\cos(2 \beta)
\end{bmatrix}
\begin{bmatrix}
d(2)
\end{bmatrix}
\]
\[
\begin{bmatrix}
D(3)
\end{bmatrix} = \begin{bmatrix}
-\cos(2 \beta) & -\cos(3 \beta) & \cos(\beta)
\end{bmatrix}
\begin{bmatrix}
d(3)
\end{bmatrix}
\]
\]

where, \(\beta = \pi / 7\), and
\[
s(1) = a_1(1)+a_1(6),
\]
\[
s(2) = a_1(2)+a_1(5),
\]
\[
s(3) = a_1(3)+a_1(4),
\]
\[
d(1) = a_1(1)-a_1(6),
\]
\[
d(2) = a_1(2)-a_1(5),
\]
\[
d(3) = a_1(3)-a_1(4).
\]

To convert (13) and (14) in to the convolution-like form, we can find the primitive root of unity \(\eta\) to be 3 (for \(M=7\)), and can map the indices according to (11) as shown in Tables I and II. Using the mapping of indices according to Tables I and II, (13) can be expressed as
\[
\begin{bmatrix}
S(1)
\end{bmatrix} = \begin{bmatrix}
\cos(2 \beta) & -\cos(3 \beta) & -\cos(\beta)
\end{bmatrix}
\begin{bmatrix}
s(1)
\end{bmatrix}
\]
\[
\begin{bmatrix}
S(2)
\end{bmatrix} = \begin{bmatrix}
-\cos(3 \beta) & -\cos(\beta) & \cos(2 \beta)
\end{bmatrix}
\begin{bmatrix}
s(2)
\end{bmatrix}
\]
\[
\begin{bmatrix}
S(3)
\end{bmatrix} = \begin{bmatrix}
-\cos(\beta) & \cos(2 \beta) & -\cos(3 \beta)
\end{bmatrix}
\begin{bmatrix}
s(3)
\end{bmatrix}
\]
\]

Equation (16) is now in a form which identical to 3-point circular-convolution, except the sign-factors which is specified in (10e) and (10f). In this example (for \(M = 7\)), equation (14) does not require any reordering as given by Tables I and II, and it can be seen that it is in the desired circular-convolution-like form. Proceeding in the similar manner, one can represent \(B_1(k)\) for \(1 \leq k \leq 6\) also in terms of two 3-point circular-convolution structures.

**TABLE I: ORDERING OF INDEX \(n\)**

<table>
<thead>
<tr>
<th>(m)</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n)</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

**TABLE II: ORDERING OF INDEX \(\zeta\) AND \(\xi\)**

<table>
<thead>
<tr>
<th>(l)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k)</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>(\zeta)</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>(\xi)</td>
<td>1</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3</td>
</tr>
</tbody>
</table>

**Fig. 1:** The DGs for computation of the convolution like structures for computation of \((D\zeta)\) and \((S\zeta)\) for \(1 \leq \zeta \leq 3\), given by (14) and (16), respectively. (a) The DG for \((S\zeta)\). (b) The DG for \((D\zeta)\). (c) Function of each node of the DG.
IV. DERIVATION OF THE PROPOSED STRUCTURE

The data dependence, computations, and necessary control signals pertaining to the sign of the kernel coefficients for calculation of \( \{S(\xi)\} \) and \( \{D(\xi)\} \) for \( 1 \leq \xi, \zeta \leq 3 \) are depicted in the dependence graphs (DG) of Figs. 1(a) and 1(b), respectively. Each of the DGs, consists of 9 nodes arranged in 3 rows and 3 columns. The function of each node is depicted in Fig. 1(c). Both the DGs can be merged together and projected along \( j \)-direction with a systolic schedule to obtain a linear systolic array consisting of 3 processing elements (PEs) in \( i \)-direction.

![Diagram of systolic array](image)

Fig. 2. The proposed linear systolic array for the computing \( \{S(\xi)\} \) and \( \{D(\xi)\} \) for \( 1 \leq \xi, \zeta \leq 3 \). (a) The linear array. (b) Function of each PE. For the \( i \)-th PE, \( U_{in}=s(i) \) and \( V_{in}=d(i) \). (c) \( C(i)=\cos(i \beta) \) for \( i=1, 2 \) and 3.

As per requirement of systolic mapping, the input values available to the nodes of the DG from \([0 \ 1]^T \) direction stay in the PEs, while the multiplying coefficients available from \([1 \ 1]^T \) direction are transferred to the next PE with 2 delays and the partial result available to each of the nodes from \([1 \ 0]^T \) direction is moved to the next PEs in the subsequent cycles. The proposed structure to realize the desired functionality for systolization of a pair of convolution-like operations is shown in Fig. 2(a). Function of each PE of the structure is shown in Fig. 2(b). The input values and the tag-bits are fed to the individual PEs in parallel and the multiplying coefficients \( C(i) \) are fed through a circularly-right-shift buffer. The input values to a PE are staggered by one cycle-period with respect to the preceding PE to maintain the data dependency requirement. The right-most PE of the structure yields its first output 3 cycles after the first input arrives at its left-most PE, and produces its subsequent output in every cycle period thereafter. It delivers one set of output in every 3 cycle periods once the pipeline is filled in the first 3 cycles. The inputs \( X_{1in} \) and \( X_{2in} \) of PE-1 can be initialized with \( 'a(I_0)+a(M)' \) and \( 'a(I_0)-a(M)' \) to obtain \( A(I)(k) \) for \( 1 \leq k \leq 6 \) from the proposed linear array instead of \( \{S(\xi)\} \) and \( \{D(\xi)\} \) for \( 1 \leq \xi, \zeta \leq 3 \). \( B(I)(k) \), \( B(I)(k) \) and \( A(I)(k) \) for \( 1 \leq k \leq 6 \) can be computed from three such linear arrays and two accumulators can be used for calculation of \( A(I)(0) \) and \( A(I)(0) \) according to (6e) and (6f), respectively.

V. HARDWARE- AND TIME-COMPLEXITIES

In the previous Section, we described a linear systolic array for computing the convolution-like cores for 28-point DFT. For computing \( N \)-point DFT, in general, it would be required to use \( 4 \) such linear arrays where each such array would compute 2 number of \([M-1]/2\)-point convolution-like operations \( (M = N/4) \). Each of the proposed linear arrays, in general, will consist of \([N/4]-1\)/2 number of PEs, and two accumulators can be used for calculation of \( A(I)(0) \) and \( A(I)(0) \). It will yield the first component after \([N/4]-1\)/2 cycle-periods, and will produce the successive output in every cycle-period thereafter. Four adders will be required to add/subtract \( A(I)(k) \) with/from \( A(I)(k) \), and \( B(I)(k) \) with/from \( B(I)(k) \) for computing the DFT components according to (5) and (8). Four adders are required for the additions of (1) to obtain the final DFT output. Similarly, four adders will be required to compute \( a(n), b_1(n), a_2(n) \) and \( b_3(n) \) and four more adders are also required to compute the above values of \( s(n) \) and \( d(n) \). The proposed structure would deliver a complete set of \( N \)-DFT components in every \([N/4]-1\)/2 cycle-periods after a latency of \([N/4]-1\)/2 cycle-periods. During every cycle-period each of the PEs performs 4 real-multiplications and 4 real-addition operations. If each PE contains 4 multipliers and 4 adders, then the duration of cycle-period would be equal to \( T = T_M + T_A \); where \( T_M \) and \( T_A \) are, respectively, the time required to perform a multiplication and an addition in a PE. The proposed structure for computation of the DFT will, thus, require \( 2N-8 \) multipliers and \( 2(N+5) \) number of adders. Apart from that the proposed structure requires two registers of \( N \) words each for ordering of data during input generation. The overhead of input mapping for convolution operation, however, can be avoided by suitable relocation of the rows and columns of the coefficient matrices (equations (14) and (16)). Most of the hardware of the structure is, therefore, contributed by the multipliers and the adders. Hardware- and time-complexities of the proposed structure are computed accordingly and listed along with those of the existing structures [17-20] in Table III. It is observed that the proposed structure involves less number of multipliers compared with the corresponding existing structures. It requires either nearly the same or less hardware-complexity compared with the others. Besides, it offers 8 times more throughput, and nearly (1/8) times the latency compared with the others.

VI. CONCLUSION

A reduced-complexity algorithm is presented for computation of the DFT, where \( N \)-point transform is computed from eight number of nearly \( (N/8) \)-point circular-convolution-like operations. A systolic structure is designed further for high-throughput concurrent implementation of the proposed convolution-like-cores. The proposed structure is found to have 8 times more throughput and nearly (1/8) times of the minimum latency compared with the corresponding existing DFT structures. Besides, it involves nearly the same or less hardware-complexity compared with the existing structures. Apart from simplicity, regularity and
modularity of structure, it can have serial I/O, and does not involve complex control structure. The proposed structure can also be used as building blocks for deriving efficient systolic solution by prime-factor mapping [10-13], and other decomposition schemes [14-16] for computing the DFT of larger size. Moreover, the proposed scheme can also be extended further for computation other orthogonal transforms like DCT, DST, discrete Hartley transform (DHT), and 2-dimensional DFT [6] as well. Due to its high-throughput and low latency, this architecture will be useful for hard-real-time applications with stringent timing conditions.

REFERENCES


TABLE III: HARDWARE- AND TIME-COMPLEXITIES OF THE PROPOSED STRUCTURE AND THE EXISTING DFT STRUCTURES [17-20].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>number of multipliers</td>
<td>(4(N+1))</td>
<td>(4N)</td>
<td>(2N)</td>
<td>(2(N+1))</td>
<td>(2N-8)</td>
</tr>
<tr>
<td>number of adders</td>
<td>(4(N+1))</td>
<td>(2N)</td>
<td>(2N)</td>
<td>(2(N+1))</td>
<td>(2(N+5))</td>
</tr>
<tr>
<td>latency in cycles</td>
<td>((N+1))</td>
<td>(N)</td>
<td>((3N-4))</td>
<td>((N+1))</td>
<td>((N/4+7)/2)</td>
</tr>
<tr>
<td>throughput</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>cycle period</td>
<td>(T_M+T_A)</td>
<td>(T_M+T_A)</td>
<td>(T_M+T_A)</td>
<td>(T_M+T_A)</td>
<td>(T_M+T_A)</td>
</tr>
<tr>
<td>computation time (cycles)</td>
<td>(N)</td>
<td>(N)</td>
<td>(N-1)</td>
<td>(N)</td>
<td>((N/4-1)/2)</td>
</tr>
</tbody>
</table>

(One complex multiplication is assumed to be implemented by 4 real multiplications and 2 real additions)