Memory Footprint Reduction for Power-Efficient Realization of 2-D Finite Impulse Response Filters

Basant K. Mohanty, Senior Member, IEEE, Pramod K. Meher, Senior Member, IEEE, Somaya Al-Maadeed, Senior Member, IEEE and Abbes Amira, Senior Member, IEEE

Abstract—We have analyzed memory footprint and combinational complexity to arrive at a systematic design strategy to derive area-delay-power-efficient architectures for two-dimensional (2-D) finite impulse response (FIR) filter. We have presented novel block-based structures for separable and non-separable filters with less memory footprint by memory sharing and memory-reuse along with appropriate scheduling of computations and design of storage architecture. The proposed structures involve \( L \) times less storage per output (SPO) and nearly \( L \) times less energy consumption per output (EPO) compared to the existing structures, where \( L \) is the input block-size. They involve \( L \) times more arithmetic resources than the best of the corresponding existing structures, and produce \( L \) times more throughput with less memory band-width (MBW) than others, where \( L \) is the input block-size. We have also proposed separate generic structures for separable and non-separable filter-banks, and a unified structure of filter-bank constituting symmetric and general filters. The proposed unified structure for 6 parallel filters involves nearly 3.6\( L \) times more multipliers, 3\( L \) times more adders, \( N^2 \times N_2 + 2 \) less registers than similar existing unified structure, and computes 6\( L \) times more filter outputs per cycle with 6\( L \) times less MBW than the existing design, where \( N \) is FIR filter size in each dimension. ASIC synthesis result shows that, for filter size \((4 \times 4)\), input-block size \( L = 4 \), and image-size \((512 \times 512)\), proposed block-based non-separable and generic non-separable structures, respectively, involve 5.95 times and 11.25 times less area-delay-product (ADP), and 5.81 times and 15.63 times less EPO than the corresponding existing structures. The proposed unified structure involves 4.64 times less ADP and 9.78 times less EPO than the corresponding existing structure.

Index Terms—Block processing, 2-dimensional (2-D) finite impulse response (FIR), Digital Filters, VLSI Architecture

I. INTRODUCTION

Two-dimensional (2-D) digital filters are frequently used in 2-D signal processing as well as the image and video processing applications such as image enhancement, image restoration [1], template matching [2], face recognition, feature extraction for bio-metric systems [3]–[5], and video communication etc. The 2-D FIR filters are more popularly used compared to its infinite impulse response (IIR) counterpart due to their numerical stability and simplicity of design. The system function of 2-D FIR filter is often non-separable, while in a few cases, it is separable when impulse response

\[
H(z_1, z_2) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} h_1(i) \cdot h_2(j) \cdot z_1^{-i} z_2^{-j}
\]

where \([h(l, k)]\) is the impulse response matrix of the non-separable 2-D FIR filter of size \(N \times N\) while \([h_1(i)]\) and \([h_2(j)]\) are the impulse responses of 1-D FIR filters used for row-wise and column-wise processing of 2-D input.

Block diagrams of conventional realization of non-separable and separable 2-D FIR filters are shown in Fig.1. As shown in this figure, both these filter structures consist of two types of hardware components: (i) the combinational component and (ii) the memory or storage component. The combinational component consists mainly of the arithmetic circuits along with some steering logic like multiplexors and demultiplexers, while the storage component consists of transposition buffers and/or shift-registers to provide appropriate data to combinational units. The non-separable structure uses shift-registers to introduce the necessary row-delays for the processing of intermediate data while the separable structure uses shift-registers for transposition of intermediate data. We can find from (1) that, a non-separable 2-D FIR filter of size \((N \times N)\) involves \((N-1)\) shift-registers (SRs) of size \(M\) each, \((N-1)^2\) registers (for row-column processing), and \(N^2\) multipliers and \(N^2\) adders to compute one filter output per cycle. Similarly, we can find from (2) that, the separable 2-D filter of size \((N \times N)\) involves \((N-1)\) SRs of \(M\) words each, \((2N-1)\) registers, \(2N\) multipliers and \(2N\) adders to compute one filter output per cycle. Combinational and memory (register) complexities of full-parallel non-separable and separable filters are given in Table I. Since, the image size \((M)\) is higher than the filter-size \((N)\) by more than an order of magnitude in most of the
image-processing applications, memory becomes the dominant component of hardware complexity of 2-D FIR structures, and consumes major amount of chip-area and total power consumption.

Some systolic architectures have been suggested for VLSI implementation of 2-D FIR filters to achieve high-throughput and low-latency implementation [6]–[11]. Recently, some efficient structures have been proposed for 2-D IIR filters [13]–[17] and shown that non-separable 2-D FIR filters can also be realized efficiently using those structures. In all these existing designs, systolization [18] of the structure is considered as the major issue, and a substantially large number of delay elements are placed in the data-path to avoid global communication. Similarly, the symmetry of impulse response matrix has been exploited to reduce the hardware and time complexities of the structures [12]–[17]. In the last four decades, several design approaches have been suggested for reducing the arithmetic complexity of one-dimensional (1-D) FIR filters [19]–[28].

Keeping that in view, in this paper, we present memory-centric design strategy is discussed in Section II and block formulation of 2-D FIR filter is given in Section III. Proposed structures are presented in Section IV for separable and non-separable filters. Generic structure of filter-bank consisting of different types of filters is presented in Section V. Hardware complexity and performance of the proposed structures are discussed in Section VI. Conclusion is presented in Section VII.

II. PROPOSED DESIGN STRATEGY

To arrive at the proposed design strategy we analyze here memory complexities of possible configurations of 2-D FIR filter. The system function of non-separable 2-D FIR filter (given by (1)) can be written in a split form as:

$$H(\tilde{z}_1, \tilde{z}_2) = \sum_{i=0}^{N-1} \tilde{z}_1^{-i} H_i(\tilde{z}_2)$$  \hspace{1cm} (3a)

$$H_i(\tilde{z}_2) = \sum_{j=0}^{N-1} h(i, j) \cdot \tilde{z}_2^{-j}$$  \hspace{1cm} (3b)

Computations of (3a) and (3b) can be performed by a direct-form or a transposed-form structure to have four possible configurations such as fully-direct (direct-direct), fully-transposed (transpose-transpose), hybrid-1 (direct-transpose), and hybrid-2 (transpose-direct), for the realization of non-separable $H(\tilde{z}_1, \tilde{z}_2)$ as shown in Fig.2 for $N = 4$. All these four structures require the same number of arithmetic components (multiplier and adders) and delay elements ($\tilde{z}_1^{-1}$ and $\tilde{z}_2^{-1}$ corresponding to shift-registers and fixed registers, respectively) except their locations in the data-path. Since, the bit-widths of arithmetic units, buses and registers in the data-path are different for input signals and intermediate signals, the overall memory requirements of different configurations are different in terms of number of storage bits, although all of them have the same number of delay elements.

We have estimated memory complexity of all the four type of structures for an input image size $512 \times 512$ (i.e $M = 512$) and filter length ($N = 4$ and $N = 8$), and listed in Table II for comparison. We find that, fully-direct structure has the lowest memory requirement than others. Interestingly, the memory complexity of fully-direct structure is independent of word-length of intermediate signals since all the delay elements of this structure are placed on the input path only. This is a very useful feature to be exploited for memory footprint reduction in 2-D FIR filter structure.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Combinational</th>
<th>Memory (words)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Multiplier</td>
<td>Adder</td>
</tr>
<tr>
<td>Non-separable</td>
<td>$N^2$</td>
<td>$(M+N)(N-1)$</td>
</tr>
<tr>
<td>Separable</td>
<td>$2N$</td>
<td>$2(N-1)$</td>
</tr>
</tbody>
</table>

$N$: filter size, $M$: image width/height
TABLE II

MEMORY COMPLEXITY FULLY-DIRECT, FULLY-TRANSPOSE, HYBRID-1 AND HYBRID-2 STRUCTURES. N: FILTER-SIZE, M: INPUT IMAGE-SIZE, w: INPUT SIGNAL BIT-WIDTH AND w': INTERMEDIATE SIGNAL BIT-WIDTH.

<table>
<thead>
<tr>
<th>Structure</th>
<th>Shift-Register/Register words</th>
<th>Memory (bits)</th>
<th>Memory (bits), M = 512, w = 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Input signal storage</td>
<td>Intermediate signal storage</td>
<td>N = 4, w = 16</td>
</tr>
<tr>
<td>Fully-Direct</td>
<td>(M + N)(N - 1)</td>
<td>0</td>
<td>12384</td>
</tr>
<tr>
<td>Hybrid-1</td>
<td>N(N - 1)</td>
<td>M(N - 1)</td>
<td>24672</td>
</tr>
<tr>
<td>Hybrid-2</td>
<td>M(N - 1)</td>
<td>N(N - 1)</td>
<td>12480</td>
</tr>
<tr>
<td>Fully-Transpose</td>
<td>0</td>
<td>(M + N)(N - 1)</td>
<td>24768</td>
</tr>
</tbody>
</table>

Fig. 2. Four different configurations for realization of 2-D FIR filter for N = 4. (a) Fully-direct structure, (b) Hybrid-1 structure (c) Hybrid-2 structure, and (d) Fully-transposed structure, where \( z_{-1} \) represents a shift-register of M words and \( z_{-1} \) represents a single register.

A. Exploration of Memory-Reuse Possibilities

To explore the memory reuse possibilities, let us consider the input data-flow of fully-direct structure for the computation of \( m \)-th row of outputs \( \{y(m, n), y(m, n + 1), y(m, n + 2), y(m, n + 3)\} \) as shown in Fig.3 for \( N = 4 \). The samples required to compute a given filter output is shown in a pair of curly braces and the corresponding filter output is shown at its right. Each arrow shows the source (shift-register/register) of samples. To compute each output of \( (4 \times 4) \) filter, 16 input-samples corresponding to 4 rows and 4 columns of 2-D input are required. Out of 4 rows (numbered as \( m, m - 1, m - 2, m - 3 \)), the \( m \)-th row is the current input-row and others are immediate past rows. Memory-unit uses three shift-registers (SR-1, SR-2, SR-3) to buffer 3 past \( (m - 1, m - 2, m - 3) \)-th rows of input samples. The required \( (n - 1, n - 2, n - 3) \)-th columns of samples of a particular-row are...
provided by the serial-in parallel-out (SIPO) shift-register of \((N - 1)\) words. As shown in Fig.3, all the 16 input-samples required to compute the filter output are obtained from the memory-unit and current input. Therefore, memory used by fully-direct structure to compute each output is \((3M + 12)\) words, where SRs-unit provides \(3M\) words and fixed register-unit provides 12 words. Memory band-width (MBW) of the structure is 15 (read operations on SR-unit is 3, and 12 values are obtained from register-unit to compute an output). This can be generalized to find the number of memory words used by fully-direct structure to be \(\left[(N - 1)(M + N)\right]\) and MBW to be \((N^2 - 1)\).

As shown in Fig.3, in total 64 input samples are required to compute the outputs \(y(m, n), y(m, n + 1), y(m, n + 2), y(m, n + 3)\). These 64 samples belong to 4 rows \{m, m - 1, m - 2, m - 3\} and 7 columns \{n + 3, n + 2, n + 1, n - 1, n - 2, n - 3\} of the input image. It can be noticed that out of 64 samples 28 samples are different from each other, while redundancy exists in rest 36 samples. The redundant values corresponding to filter-output \(y(m, n), y(m, n + 1), y(m, n + 2), y(m, n + 3)\) in the blue boxes in Fig.3. These redundant-memory access could be avoided by parallel computation of filter outputs \(y(m, n), y(m, n + 1), y(m, n + 2), y(m, n + 3)\). In that case, four current input samples \(x(m, n), x(m, n + 1), x(m, n + 2), x(m, n + 3)\) corresponding to four outputs need to be available in each cycle. Four input values of each of the past rows \(m - 1, m - 2, m - 3\) are retrieved from respective shift-registers in every cycle. To realize this, each of the shift registers (of \(M\) words) is required to be split into four equal parts of \(M/4\) words. Therefore, 12 input values are obtained from the SR-unit in every cycle. Out of 28 unique samples, 4 are obtained as input samples of current cycle and 12 samples are obtained from the SR-unit. The remaining 12 samples are obtained from four SIPO shift-registers. The memory usage for parallel computation of 4 filter outputs is \(3 \times 4 \times M/4 + 12\) which is the same as the memory usage of the fully-direct structure for one output. MBW for parallel computation of four filter outputs is \((4 \times 3 + 12 = 24)\). The memory words required to compute a block of \(L\) filter outputs per cycle of the 2-D FIR filter of size \((N \times N)\) can therefore found to be \(\left[(N - 1)(M + N)\right]\) and MBW can found to be \(\left[(N - 1)(L + N)\right]\).

Interestingly, the storage space of fully-direct structure of 2-D FIR filter of size \((N \times N)\) is the same as parallel computation of \(L\) filter outputs due to memory reuse during parallel computation. Only arithmetic resources increases proportionately with throughput rate. This is an important feature which can be utilized for area and power saving. The memory reuse efficiency (MRE) of block-based structure can be estimated as \(\text{MRE}=\left|\frac{\text{total input words} - \text{actual memory usage}}{\text{actual memory usage}}\right|\), where total input words is \(L\) times the memory-usage of single-input single-output (SISO) structure. For block-based structure with block size \(L, MRE=\left|\frac{(N - 1)(M + N) - (N - 1)(M + N)}{(N - 1)(M + N)}\right| = L - 1\). Therefore, MRE of block-based parallel structure increases proportionately with the block-size \((L)\). Higher the MRE, lesser is the memory requirement per output. Consequently, the structure is more area-delay efficient compared with the SISO structure. MBW of block-based structure increases by a factor of \(\left|\frac{(L + N)/(N - 1)}\right|\) instead of \(L\) times, where \(L\) is the input block-size. This is mainly due to the number of redundant samples \(\beta = N(N - 1)/(L - 1)\). Higher the \(\beta\), better is the MBW reduction. We have estimated the memory band-width saving (MBS) using the formula \(\text{MBS}=\left|\frac{(L \times \text{MBW of SISO structure}) - \text{MBW of block-based structure of block-size} L}{(L \times \text{MBW of SISO structure})}\right|\). Therefore, we can have \(\text{MBS}=\beta/(L \times (N^2 - 1))\). Since, \(\beta\) varies with \(L\) and \(N\), MBS is higher for larger size filters with higher input block-size.

We have estimated \(\beta\), MBS and MRE of parallel 2-D FIR structure for different filter sizes \((N)\) and input block-sizes \((L)\). The estimated values are listed in Table III to quantify the scope of memory reuse. We can find from Table III that a block-based realization enhances memory-reuse and reduces memory-band width per output. Therefore, we have presented the block formulation and its subsequent implementation of 2-D FIR filters.

B. Memory-Sharing in Generalized 2-D FIR Filter Structures

The fully-direct non-separable structure as well as the conventional separable structure use shift-registers to store \(M\) words each, but shift-registers of fully-direct non-separable structure stores input pixel values while those of separable structure stores intermediate values. Due to the difference in bit-width, a common shift-register unit cannot be shared by these two structures. A different design approach for separable filter is required where the shift-register unit stores the input signal only. For this purpose, we have used an efficient decomposition scheme for 2-D separable filter in the following.
TABLE III
MEMORY REUSE EFFICIENCY AND MEMORY BAND-WIDTH SAVING FOR DIFFERENT SIZE FILTERS (N) AND INPUT-BLOCK SIZE (L)

<table>
<thead>
<tr>
<th>Filter size (N)</th>
<th>Block-size (L)</th>
<th>β</th>
<th>MBS (in %)</th>
<th>MRE (in times)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N = 4</td>
<td>2</td>
<td>12</td>
<td>40</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>36</td>
<td>60</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>84</td>
<td>70</td>
<td>7</td>
</tr>
<tr>
<td>N = 8</td>
<td>2</td>
<td>56</td>
<td>44.4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>168</td>
<td>66.7</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>392</td>
<td>77.8</td>
<td>7</td>
</tr>
<tr>
<td>N = 16</td>
<td>2</td>
<td>240</td>
<td>47</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>720</td>
<td>70.5</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1680</td>
<td>82.4</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3600</td>
<td>88.2</td>
<td>15</td>
</tr>
<tr>
<td>N = 32</td>
<td>2</td>
<td>992</td>
<td>48.4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2976</td>
<td>72.6</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>6944</td>
<td>84.8</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>14880</td>
<td>90.8</td>
<td>15</td>
</tr>
</tbody>
</table>

β = N(N−1)(L−1), MBS = β/[L(N^2−1)], MRE (in times) = L − 1.

The input-output relation of separable 2-D FIR filter, given by (2), can be written as:

\[ y(m, n) = \sum_{i=0}^{N−1} \sum_{j=0}^{N−1} h_1(i) \cdot h_2(j) \cdot x(m − i, n − j) \]  \hspace{1cm} (4)

Computation of (4) can be expressed in split form as:

\[ y(m, n) = \sum_{i=0}^{N−1} h_2(i) \cdot v(m, n − i) \]  \hspace{1cm} (5a)

\[ v(m, n) = \sum_{i=0}^{N−1} h_1(i) \cdot x(m − i, n) \]  \hspace{1cm} (5b)

and can be expressed as inner-products of a pair of N-point vectors \( b_m, n, h_1 \) and \( u_m, n, h_2 \) as

\[ u(m, n) = b_m, n \cdot h_1 \]  \hspace{1cm} (6a)

\[ y(m, n) = u_m, n \cdot h_2 \]  \hspace{1cm} (6b)

and \( b_m, n, u_m, n, h_1 \) and \( h_2 \) are given by

\[ b_m, n = [x(m, n) \ x(m − 1, n) \ \ldots \ x(m − N + 1, n)] \]  \hspace{1cm} (7a)

\[ u_m, n = [u(m, n) \ u(m, n − 1) \ \ldots \ u(m, n − N + 1)] \]  \hspace{1cm} (7b)

\[ h_1 = [h_1(0) \ h_1(1) \ \ldots \ h_1(N − 1)]^T \]  \hspace{1cm} (7c)

\[ h_2 = [h_2(0) \ h_2(1) \ \ldots \ h_2(N − 1)]^T \]  \hspace{1cm} (7d)

According to (5), input-vectors are fed to the row-filter in column overlapped order, and the row-filter generates intermediate values column-wise exactly in the same order as the column-filter consumes intermediate values. Consequently, transposition-unit in this case is comprised of fixed registers instead of shift-registers.

Separable filter structure based on decomposition scheme of (5) is shown in Fig.4 for \( N = 4 \). The input samples are fed as overlapped blocks \( (b_{m,n}) \) for \( 0 \leq m \leq M − 1 \) and \( 0 \leq n \leq N − 1 \), where successive blocks of a column are overlapped by \( N − 1 \) samples. The input-blocks are fed in row-serial order from shift-register unit comprising of \( N − 1 \) shift-registers of \( M \) words each. We can find from Fig.3 and Fig.4 that shift-register unit of fully-direct non-separable structure and the separable structure based on this decomposition scheme has the same number of shift-registers and they are of the same size. Interestingly, the data-input and data-output formats of both these shift-register units are identical. This favours the possible sharing of shift-register units of fully-direct non-separable structure and separable structure. This leads to a generalized structure for both non-separable and separable filters individually or in parallel configuration.

Data-flow of a shared shift-register unit is shown in Fig.5 for \( N = 4 \), where the input-data requirement of both fully-direct non-separable and separable structures is taken care of by the shared shift-register unit. A shared shift-register unit not only offers memory-saving, but also allows parallel realization of both non-separable and separable filters. It is shown in the later Sections that the parallel implementation of generalized structure offers higher area-delay-power efficiency over sequential structure. Keeping these facts in view, we have outlined here a systematic memory-centric design strategy to derive an area-delay-power efficient structure for 2-D FIR filter.

- A fully-direct form structure should be used for non-
separable filter to have less storage-complexity.
• A block implementation of fully-direct structure should be used for MBW reduction.
• Separable structure based on the proposed decomposition algorithm could be derived for shift-register sharing with non-separable filter structure.
• Appropriate algorithm partitioning and scheduling scheme need to be used for separable filter to minimize memory bandwidth and increase register sharing.

The register sharing property of proposed non-separable and separable filter structures are exploited further to derive generic structures. The proposed generic structures can be configured for realization of a single filter of types (separable, non-separable, symmetry (diagonal, four-fold rotational, quadrant) or parallel realization of any combination of these filters.

III. BLOCK FORMULATION OF 2-D FIR FILTERS

A. For Non-separable Filter

Let us consider a non-separable filter which processes a block of \( L \) input samples and generates a block of \( L \) outputs in every cycle. The \( k \)-th block of filter output of the \( m \)-th row \( y_{m,k} \) is computed by relation;

\[
y_{m,k} = \sum_{i=0}^{N-1} v_{i,k}
\]

where \( y_{m,k} \) and \( v_{i,k} \) are defined as

\[
y_{m,k} = \begin{bmatrix} y(m,kL) & y(m,kL-1) & \ldots & y(m,kL-1) \\ \end{bmatrix}^T
\]

\[
v_{i,k} = \begin{bmatrix} v(i,kL) & v(i,kL-1) & \ldots & v(i,kL-1) \\ \end{bmatrix}^T
\]

The intermediate vector \( v_{i,k} \) is computed by product of input-matrix \( A_k^{m-i} \) and impulse-response vector \( h_i \), and given by

\[
v_{i,k} = A_k^{m-i} \cdot h_i
\]

The input matrix \( A_k^{m-i} \) is derived from \((m-i)\)-th row of the input matrix \( \begin{bmatrix} X \end{bmatrix} \) of size \((M \times M)\) and given by

\[
A_k^{m-i} = \begin{bmatrix}
x(m',kL) & x(m',kL-1) & \ldots & x(m',k'L+1) \\
x(m',kL-1) & x(m',kL-2) & \ldots & x(m',k') \\
\vdots & \vdots & \ddots & \vdots \\
x(m',kL-L+1) & x(m',kL-L) & \ldots & x(m',k'-L+2)
\end{bmatrix}
\]

for \( m' = m - i \) and \( k' = kL - M \), and \( h_i \) is given by:

\[
h_i = \begin{bmatrix} h(i,0) & h(i,1) & \ldots & h(i,N-1) \end{bmatrix}^T
\]

From (9) and (11), we find \( v(i,kL-l) \) is the inner-product of \( s_k^{m-i} \) (\(l\)-th row of matrix \( A_k^{m-i} \)) and \( h_i \), given by

\[
v(i,kL-l) = s_k^{m-i} \cdot h_i
\]

B. For Separable 2-D FIR Filter

Let us consider a separable filter which processes a block of \( L \) input samples and generates a block of \( L \) outputs in every cycle. The \( k \)-th block of filter output of the \( m \)-th row \( y_{m,k} \) is computed in this case by two successive matrix-vector products given by

\[
u_{m,k} = B_k^m \cdot h_1
\]

\[
y_{m,k} = U_k^m \cdot h_2
\]

and the input-matrix \( B_k^m \) and intermediate-matrix \( U_k^m \) are given by

\[
B_k^m = \begin{bmatrix} x(m,kL) & x(m-1,kL) & \ldots & x(m',kL) \\
x(m,kL-1) & x(m-1,kL-1) & \ldots & x(m',kL-1) \\
\vdots & \vdots & \ddots & \vdots \\
x(m,k') & x(m-1,k') & \ldots & x(m',k')
\end{bmatrix}
\]

for \( m' = m - M + 1 \) and \( k' = kL - L + 1 \).

\[
U_k^m = \begin{bmatrix} u(m,kL) & u(m,kL-1) & \ldots & u(m,k') \\
u(m,kL-1) & u(m,kL-2) & \ldots & u(m,k'-1) \\
\vdots & \vdots & \ddots & \vdots \\
u(m,kL-L+1) & u(m,kL-L) & \ldots & u(m,k'-L+1)
\end{bmatrix}
\]

for \( k' = kL - N + 1 \).

IV. PROPOSED STRUCTURES

In this Section, we have derived two separate structures for block implementation of non-separable and separable 2-D FIR filters.

A. Block-based Structure for Non-separable 2-D FIR Filter

The computation of (8) and (10) are mapped into a fully-direct \( L \) parallel structure to derive the proposed block-based structure for non-separable 2-D FIR filter. The proposed structure is shown in Fig.6 for filter-size \( N = 8 \) and block-size \( L = 4 \). It consists of one memory-module and one arithmetic module.

1) Memory-Module Design: The memory-module of Fig.6 is comprised of one shift-register array and \( N \) input-register units (IRUs). The shift-register array further consists of \((L-1) = 28\) shift-registers of \( P \) words each, where \( P = M/L \). Proposed structure receives a block of \( L \) input samples and computes a block of \( L \) outputs in each cycle. All the samples of each input-block belong to the same row and the inputs are fed to the structure block-by-block and then row-by-row in serial order. The input-block \( \begin{bmatrix} x_{k,l}^m \end{bmatrix} \) corresponding to the \( m \)-th row of input image \( \begin{bmatrix} X \end{bmatrix} \) is fed to the structure during \((k+1)\)-th cycle of \( m \)-th set of \( P \) cycles, and the entire image is fed in \( MP \) cycles for \( 0 \leq k \leq P - 1 \).
\[ 0 \leq m \leq M - 1, \quad [L(N - 1)] \] shift-registers of the shift-register array are arranged in groups referred to as SR-block and each SR-block has \( L \) shift-registers. As shown in Fig.6, \{SR-1,SR-2,SR-3,SR-4\} constitute SR-block-1 and \{SR-25,SR-26,SR-27,SR-28\} constitute SR-block-7. One SR-block stores one input row and the shift-register array stores \((N - 1)\) rows of input. SRs of SR-block are connected such that a block of samples transfer from one SR-block to the adjacent SR-block to its right after every cycle. Therefore, a block of \( L = 4 \) inputs of a particular row are obtained from each SR-block in every cycle, and \((N - 1)\) input-blocks corresponding to \((N - 1)\) consecutive input rows are obtained from the shift-register unit in every cycle.

\[ 0 \leq m \leq M - 1, \quad [N/L] \] shift-registers of the shift-register array are arranged in groups referred to as SR-block and each SR-block has \( L \) shift-registers. As shown in Fig.6, \{SR-1,SR-2,SR-3,SR-4\} constitute SR-block-1 and \{SR-25,SR-26,SR-27,SR-28\} constitute SR-block-7. One SR-block stores one input row and the shift-register array stores \((N - 1)\) rows of input. SRs of SR-block are connected such that a block of samples transfer from one SR-block to the adjacent SR-block to its right after every cycle. Therefore, a block of \( L = 4 \) inputs of a particular row are obtained from each SR-block in every cycle, and \((N - 1)\) input-blocks corresponding to \((N - 1)\) consecutive input rows are obtained from the shift-register unit in every cycle.

Current input-block and \((N - 1)\) past input-blocks available in the shift-register array are sent to \( N \) IRUs to generate input-matrix \([A_k]\) of size \((L \times N)\). During \( k\)-th cycle of each set of \( P \) cycles, the first IRU receives the current block of input \([x_k^m]\), and the \((i + 1)\)-th IRU receives an input-block from \(i\)-th SR-block. The \((i + 1)\)-th IRU generates the input-matrix \([A_k^i]\), for \(1 \leq i \leq N - 1\). According to (11), a block of \((N + L - 1)\) consecutive samples of \((m - i)\)-th row of input are required to construct the matrix \([A_k^i]\). Each IRU receives \( L \) samples from the corresponding SR-block during \( k\)-th cycle and uses \((N - 1)\) past samples belonging to \([N/L]\) past input-blocks. The internal structure of \((i + 1)\)-th IRU is shown in Fig.7 for \( N = 8 \) and \( L = 4 \). It consists of \((N - 1)\) registers, and produces \( L \) number of \( N\)-point input-vectors \((s_{k,l}^m)\) for \(0 \leq l \leq L - 1\) corresponding to \( L \) rows of \([A_k]\).

\[ 0 \leq m \leq M - 1, \quad [N/L] \] shift-registers of the shift-register array are arranged in groups referred to as SR-block and each SR-block has \( L \) shift-registers. As shown in Fig.6, \{SR-1,SR-2,SR-3,SR-4\} constitute SR-block-1 and \{SR-25,SR-26,SR-27,SR-28\} constitute SR-block-7. One SR-block stores one input row and the shift-register array stores \((N - 1)\) rows of input. SRs of SR-block are connected such that a block of samples transfer from one SR-block to the adjacent SR-block to its right after every cycle. Therefore, a block of \( L = 4 \) inputs of a particular row are obtained from each SR-block in every cycle, and \((N - 1)\) input-blocks corresponding to \((N - 1)\) consecutive input rows are obtained from the shift-register unit in every cycle.

\[ 0 \leq m \leq M - 1, \quad [N/L] \] shift-registers of the shift-register array are arranged in groups referred to as SR-block and each SR-block has \( L \) shift-registers. As shown in Fig.6, \{SR-1,SR-2,SR-3,SR-4\} constitute SR-block-1 and \{SR-25,SR-26,SR-27,SR-28\} constitute SR-block-7. One SR-block stores one input row and the shift-register array stores \((N - 1)\) rows of input. SRs of SR-block are connected such that a block of samples transfer from one SR-block to the adjacent SR-block to its right after every cycle. Therefore, a block of \( L = 4 \) inputs of a particular row are obtained from each SR-block in every cycle, and \((N - 1)\) input-blocks corresponding to \((N - 1)\) consecutive input rows are obtained from the shift-register unit in every cycle.

2) Arithmetic-Module Design: The arithmetic module is comprised of \( N \) functional-units (FUs) and one adder tree (AT). In each cycle, \( N \) FUs of arithmetic module receive \( N \) input-vectors from \( N \) IRUs of storage-module such that \((i + 1)\)-th FU receives \( L \) input-vectors generated by \((i + 1)\)-th IRU and it performs \( L \) separate inner-product computation with the \((i + 1)\)-th row of impulse-response matrix \([h_i]\) to obtain \( L\)-point partial output-vector \([v_i]\) according to (10). Internal structure of the FU is shown in Fig.8(a). It consists of \( L \) inner-product cells (IPCs). Each IPC (shown in Fig.9) performs \( N\)-point inner product of input-vector and weight-vector. Finally, \( N \) partial-output vectors of \( N \) FUs are added together in an adder-block (shown in Fig.8(b)) according to (8) to obtain one block of complete output \([y_k^m]\) corresponding to the \( m\)-th block of input in one cycles and successive output blocks after every cycle thereafter, where one clock period \( T = T_M + T_A + T_{FA}(2\log_2 N + 1), T_M, T_A, \) and \( T_{FA} \) are, respectively, one multiplication time, addition time and one full-adder delay. One complete row of output is obtained in \( P \) cycles and the entire output matrix of size \((M \times M)\) in \( MP \) cycles.

![Fig. 6. Proposed block-based structure for non-separable 2-D FIR filter for block-size \( L = 4 \) and filter-size \( N = 8 \).](image)

![Fig. 7. Internal structure of \((i + 1)\)-th input register unit (IRU) for \( L = 4 \) and \( N = 8 \).](image)

![Fig. 8. (a) Structure of \((i + 1)\)-th functional unit (FU) for \( L = 4 \) and \( N = 8 \). (b) Structure of adder-block for \( L = 4 \) and \( N = 8 \).](image)

![Fig. 9. Internal structure of inner-product cell (IPC) for \( N = 8 \).](image)
B. Block-based Structure for Separable Filter

The proposed block-based structure for separable 2-D FIR filter is shown in Fig.10. It consists of one processing cell (PC) and one transposition-unit (TU), where a PC consists of two FUs. Structure of each FU is the same as the one shown in Fig.8(a) except that a constant vector is stored in each FU. In this case, FU-1 and FU-2 store the constant-vectors \( (h_1) \) and \( (h_2) \), respectively. It processes the \((k+1)\)-th block of input \( [X] \) during the \( k \)-th cycle of a set of \( P \) cycles, and produces an output block \( [Y] \), where \( P = M/L \) and \( L \) is the input block-size. One complete row of \([X]\) is processed in \( P \) cycles and the complete image in \( MP \) cycles.

Proposed structure receives a block of \( L \) input samples through \( L \) number of \( N \)-point input-vectors \( (b_{m,1:k}) \) where each of the input-vectors is overlapped by \((N-1)\) samples. Input-vectors of \( k \)-th and \((k+1)\)-th cycles of the \( m \)-th input cycles are shown in Fig.10 for \( N = 8 \) and \( L = 4 \). Components of each input-vectors are shown in the rectangular box adjacent to its left. The input-vectors of \( k \)-th cycle form the matrix \( (B^m_k) \) as given in (15), where \( b_{m,1:k} \) is the \((l+1)\)-th row of \( B^m_k \), for \( 0 \leq l \leq L-1 \), \( 0 \leq k \leq P-1 \), and \( 0 \leq m \leq M-1 \). \( L \) rows of \( B^m_k \) are fed to IPCs of FU-1 in parallel to perform one matrix-vector multiplication with the constant vector \( (h_1) \) to calculate an \( L \)-point intermediate-vector \( (u_{m,k}) \) according to (14a). From each intermediate-vector, one intermediate-matrix (as given in (15)) is generated, such that \( U^m_k \) is generated from \( (u_{m,k}) \). TU generates the required rows of \( U^m_k \) from \( (u_{m,k}) \). We can find from (11) and (16) that the elements of \( U^m_k \) and \( A^k_{l} \) satisfy similar property. Therefore, TU performs the same function as the IRUs and its structure is identical with that of an IRU (as shown in Fig.7). The TU of separable structure generates \( L \) rows \( (s_{k,l}^m, \) for \( 0 \leq l \leq L-1 \)) of \( U^m_k \) in parallel and feeds those to FU-2 in parallel to perform one matrix-vector product with constant-vector \( (h_2) \) to compute a block of filter output \( (y_{m,k}) \) according to (16b).

V. GENERIC STRUCTURES

In this Section, we derive two separate generic structures for non-separable and separable filter banks. Also we have proposed a unified structure for realization of 2-D FIR filter-bank comprised of non-separable and separable filters.

A. Generic Block-Based Structure for Non-separable Filters

The coefficient matrices of non-separable filters can have varieties of symmetry, e.g., diagonal, centro, four-fold rotational, quadrant etc. These symmetries can be exploited to realize the transfer functions with lesser number of multiplications. The arithmetic module of proposed structure for non-separable filters could be optimized to take advantage of these symmetry property. The storage-module of the structure can be interfaced as a common unit with arithmetic modules of filters with and without symmetry. This results in a generic structure shown in Fig.11. Each sub-module of arithmetic-module of generic structure represents arithmetic module of a constituent filter. The arithmetic-module of each filter is enabled with a select signal \( (E_{Ni}) \) for \( 1 \leq i \leq 4 \) to switch off the arithmetic module to have power saving if the output of a particular filter is not required. The proposed generic structure can be used to realize any of the four types of filters or a parallel combination of filters by selecting the arithmetic modules of respective filters through the select signals. The proposed generic structure has higher MRE and MBS than the non-separable structure for a given block-size and filter-size due to common storage-unit. The area-delay-product (ADP) and power consumed per output \( (PPO) \) of the proposed generic structure in parallel configuration is expected to be significantly less than separate implementation of individual filters.

B. Generic Block-Based Structure for Separable Filters

The proposed generic structure for the realization of separable filters with and without symmetry is shown in Fig.12. The structure is similar to proposed generic non-separable structure except that the shift-register array is only common with PCs of the processing unit. Proposed generic structure can be used to realize a separable filter with and without symmetry in parallel.

![Fig. 10. Proposed block-based structure for separable 2-D FIR filter for \( N = 8 \) and \( L = 4 \).](image)

![Fig. 11. Proposed generic block-based structure for non-separable 2-D FIR filters, \( N \) is the filter size and \( L \) is the input block-size.](image)
C. Unified Structure for 2-D FIR Filter-Bank

The shift-register size and input to the shift-register array are the same in the proposed separable and non-separable generic structures. \(NL\) input samples are obtained from the shift-register array and fed to the IRU-array of generic non-separable structure as \(N\) blocks of \(L\) samples each, while the processing unit of the generic separable structure is fed with \(L\) blocks of \(N\) samples each. Therefore, the input-blocks of both non-separable and separable filters can be obtained from the same shift-register array. Outputs of the common shift-register array need to be rearranged appropriately for the non-separable and separable generic structures.

Data rearrangement of a common shift-register array is shown in Fig.13. The data-flow of an SR-block is shown in blue color. The input-blocks shifting through the SR-block are shown for the \(k\)-th and \((k+1)\)-th cycle. The contents of the first two locations of each SR of the SR-block are shown for the \(k\)-th and \((k+1)\)-th input cycles of the \(m\)-th input row. The rectangular dotted boxes show the content of one cell of the SR-block comprised of 4 SRs corresponding to \(L = 4\). A direct flow of data from the shift-register unit meet the data-flow requirement of non-separable generic structure while shift-register output data are rearranged (shown by the data-distribution block in Fig.13) for the separable generic structure.

The proposed unified structure for 2-D FIR filter is shown in Fig.14. It has a common storage unit for both separable and non-separable sections which can be used for the realization of any one of four types of non-separable or two types of separable filters. It also can be used for parallel realization of any combination of separable or non-separable. It involves \((M + N + 2)(N - 1)\) memory-words and computes \(L\) outputs each of all the six filters in every cycle in its full-parallel configuration. Although we have shown the unified structure for 6 parallel filters, it can be used for realization of more than 6 filters without any additional storage. The processing module complexity (arithmetic-modules and PCs of each filter) only increases proportionately with the number of parallel filters. This is a major advantage for area-delay-power efficient realization of large size filter banks consisting of separable and non-separable filters.

VI. COMPLEXITIES AND PERFORMANCE CONSIDERATIONS

The proposed structure for non-separable filter consists of a storage-module and an arithmetic module, while the proposed separable structure consists of one shift-register array and one processing cell, where the processing cell again consists of both arithmetic and storage components. The storage module of non-separable structure consists of one shift-register array and \(N\) IRUs. The shift-register array of both non-separable and separable structures consists of \((N - 1)\) SRs of \(M\) words each. Each IRU consists of \((N - 1)\) registers. The arithmetic module of non-separable structure consists of \(N\) FUs and one adder-block while the processing cell of separable structure consists of two FUs and one TU (same as the IRU). Each FU consists of \(L\) IPCs, and each IPC consists of \(N\) multipliers and one adder-tree (AT) to add \(N\) words. The adder-block consists of \(L\) such ATs.

A. Complexity of Block-based Structures for Separable and Non-separable Filters

The arithmetic-module of block-based non-separable structure involves \((LN^2)\) multipliers, \([L(N^2 - 1)]\) adders while
each processing cell of separable structure involves $2LN$ multipliers, $2L(N - 1)$ adders and $(N - 1)$ registers. Apart from these, the non-separable structure involves $[(M + N)(N - 1)]$ registers and the separable structure involves $[(M + 1)(N - 1)]$ registers. Both these structures compute $L$ outputs per cycle, where one cycle period is $T = T_M + T_1$ and $T = T_M + T_2$, respectively for non-separable and separable structure, for $T_1 = T_A + 2T_{FA}(\log L N - 1)$, $T_2 = T_A + T_{FA}(\log L N - 1)$, $T_M$, $T_A$ and $T_{FA}$ are, respectively, one multiplier delay, adder delay and full-adder delay$^2$.

### TABLE IV

**General Comparison of MRE and MBWPO of Proposed Structures.** $M$: Input-image Width/Height, $N$: Filter-size, $L$: Input-block Length

<table>
<thead>
<tr>
<th>Structure</th>
<th>MRE</th>
<th>MBWPO</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS-block-based</td>
<td>$L - 1$</td>
<td>$(L + N)(N - 1)/L$</td>
</tr>
<tr>
<td>S-block-based</td>
<td>$L - 1$</td>
<td>$(L + 1)(N - 1)/L$</td>
</tr>
<tr>
<td>NS-Generic</td>
<td>$4L - 1$</td>
<td>$(L + N)(N - 1)/4L$</td>
</tr>
<tr>
<td>S-Generic</td>
<td>$2L - 1$</td>
<td>$(L + 2)(N - 1)/2L$</td>
</tr>
<tr>
<td>Unified</td>
<td>$[M(6L - 1) + N(4L - 1) + 2(L - 1)]/(M + N + 2) \times (N - 1)/6L$</td>
<td>$(L + N + 2)$</td>
</tr>
</tbody>
</table>

**Legend:** NS: non-separable, S: separable.

### B. Complexity of Generic Structures

Arithmetic module of non-separable generic structure comprises four sub-modules corresponding to four types of filters where each sub-module represents arithmetic-module of the corresponding filter. Sub-modules of symmetric filters involve the same number of adders as those of sub-module of general filter which is $[L(N^2 - 1)]$, and only differ by the number of multipliers. Sub-module of diagonal, four-fold and quadrant symmetry filters, respectively, involves $[LN(N + 1)/2]$, $(LN^2/4)$ and $(LN^2/2)$ multipliers. Arithmetic-module of non-separable generic structure, therefore, involves $[LN(9N + 2)/4]$ multipliers and $[4L(N^2 - 1)]$ adders. Processing unit of proposed separable generic structure is comprised of two processing cells (one general and one symmetric filter). It involves $3LN$ multipliers, $4L(N - 1)$ adders and $2(N - 1)$ registers. The proposed non-separable generic structure, therefore, involves $[LN(9N + 2)/4]$ multipliers and $[4L(N^2 - 1)]$ adders and $[(M + N)(N - 1)]$ registers. It computes $L$ outputs of each of the four filters in every cycle. Similarly, the proposed separable generic structure involves $3LN$ multipliers, $4L(N - 1)$ adders and $[(M + 2)(N - 1)]$ registers, and computes $L$ outputs of each pair of filters in every cycle. The proposed unified structure has one storage unit and one processing module which is comprised of one non-separable section and one separable section. The non-separable section represents arithmetic-module of proposed non-separable generic structure whereas the separable section represents one processing unit of separable generic structure. Complexity of storage unit is the same as those of proposed block-based non-separable structure. The proposed unified structure, therefore, involves $LN(9N + 14)/4$ multipliers, $4L(N + 1 - 2)$ adders and $[(M + N + 2)(N - 1)]$ registers, and computes $L$ filter outputs of each of six filters (four non-separable and two separable) in every cycle.

![Memory Reuse Efficiency and Bandwidth Requirement](image)

**Fig. 15.** (a) Memory reuse efficiency (MRE), (b) Memory band-width per output (MBWPO). NS, S, NSG, SG and UNI, respectively, stands for non-separable, separable, non-separable generic, separable generic and unified.

### C. Memory Reuse Efficiency and Bandwidth Requirement

Using the definition of MRE and MBWPO given in Section II, we have calculated MRE and MBWPO of proposed structures according to expressions given in Table IV. Using these expressions, we have estimated MRE and MBWPO of the proposed structure for filter-size $N = 8$ and image-size $M = 512$. The estimated values are shown in graphs of Fig. 15. The MRE of proposed structures increases with block-size. MRE of unified structure is higher than others for a given block-size. This is mainly due to memory-sharing by more filters in the unified structure.
TABLE V
COMPARISON OF HARDWARE- AND TIME- COMPLEXITY OF THE PROPOSED STRUCTURES AND EXISTING STRUCTURES

<table>
<thead>
<tr>
<th>Structures</th>
<th>Filter type</th>
<th>Multiplier</th>
<th>Adder</th>
<th>Register</th>
<th>Filter-output/cc</th>
<th>MBWPO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Van [11]</td>
<td>non-separable</td>
<td>$N^2$</td>
<td>$N^2 - 1$</td>
<td>$3N((N - 1)/3) + M(N - 1)$</td>
<td>$1/[T_M + 2T_A]$</td>
<td>$N^2 - 1$</td>
</tr>
<tr>
<td>Khoo et al [16]</td>
<td>non-separable</td>
<td>$N^2$</td>
<td>$N^2 - 1$</td>
<td>$(M + N)(N - 1)$</td>
<td>$1/[T_M + 2T_A]$</td>
<td>$N^2 - 1$</td>
</tr>
<tr>
<td>Proposed</td>
<td></td>
<td>$LN^2$</td>
<td>$L(N^2 - 1)$</td>
<td>$(M + N)(N - 1)$</td>
<td>$L/[T_M + T_A]$</td>
<td>$(L + N)(N - 1)/L$</td>
</tr>
<tr>
<td>Mohancy et al [29]</td>
<td>separable</td>
<td>$2N$</td>
<td>$2(N - 1)$</td>
<td>$(M + 1)(N - 1)*$</td>
<td>$1/[T_M + T_A]$</td>
<td>$2N - 1$</td>
</tr>
<tr>
<td>Proposed</td>
<td></td>
<td>$2LN$</td>
<td>$2L(N - 1)$</td>
<td>$(M + 1)(N - 1)$</td>
<td>$L/[T_M + T_A]$</td>
<td>$(L + 1)(N - 1)/L$</td>
</tr>
<tr>
<td>Proposed Generic</td>
<td></td>
<td>$3LN$</td>
<td>$4L(N - 1)$</td>
<td>$(M + 2)(N - 1)$</td>
<td>$2L/[T_M + T_A]$</td>
<td>$(L + 2)(N - 1)/2L$</td>
</tr>
</tbody>
</table>

*$(MN - M)$ registers required to feed overlapped input-blocks.
$T_1 = T_A + 2T_{FA}(\log_2 N - 1)$, $T_2 = T_A + T_{FA}(\log_2 N - 1)$, $T_{FA}$: full-adder delay.

TABLE VI
COMPARISON OF HARDWARE- AND TIME-COMPLEXITY OF PROPOSED NON-SEPARABLE GENERIC AND UNIFIED STRUCTURES AND EXISTING UNIFIED STRUCTURE OF [17]

<table>
<thead>
<tr>
<th>Structures</th>
<th>Multiplier</th>
<th>Adder</th>
<th>Register</th>
<th>cycle period</th>
<th>output/cycle</th>
<th>MBWPO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chen [17]</td>
<td>$N(5N + 2)/8$</td>
<td>$N(21N + 8)/16$</td>
<td>$M(N - 1) + 2N^2$</td>
<td>$T_M + 3T_A$</td>
<td>$1$</td>
<td>$N^2 - 1$</td>
</tr>
<tr>
<td>Proposed Generic (NS)</td>
<td>$LN(9N + 2)/4$</td>
<td>$4L(N^2 - 1)$</td>
<td>$(M + N)(N - 1)$</td>
<td>$T_M + T_1$</td>
<td>$4L$</td>
<td>$(L + N)(N - 1)/4L$</td>
</tr>
<tr>
<td>Proposed Unified (NS+S)</td>
<td>$LN(9N + 14)/4$</td>
<td>$4L(N(N + 1) - 2)$</td>
<td>$(M + N + 2)(N - 1)$</td>
<td>$T_M + T_1$</td>
<td>$6L$</td>
<td>$(L + N + 2)(N - 1)/6L$</td>
</tr>
</tbody>
</table>

$T_1 = T_A + 2T_{FA} \log_2 N - 1$, $T_{FA}$: full-adder delay.

D. Performance Comparison

In [16], an efficient 2-D IIR filter (pole-zero) structure is presented which can be modified for realization of 2-D FIR filter by removing the all-pole structure from the pole-zero structure. Accordingly, we have extracted an FIR filter structure from [16] and synthesized that for comparison. Hardware complexity of these extracted structures along with those of proposed structures and structure of [11] and [29] are listed in Table V for comparison of complexities. We find from Table V that proposed non-separable structure involves $L$ times more multipliers and adders than the existing structures and equal number of registers, but it offers $L$ times higher throughput and $N$ times less MBWPO than others. Similarly, proposed separable structure involves $L$ times more multipliers and adders than those of [29] and the same number of registers, but offers $L$ times higher throughput and nearly 2 times less MBWPO than those of [29]. Proposed separable generic structure involves $(3L/2)$ times more multipliers and 2$L$ times more adders than those of [29] and $(N - 1)$ more registers, and offers $L$ times higher throughput with nearly 4 times less MBWPO than those of [29].

We have extracted a unified 2-D FIR filter structure from the multimodal structure of [17] for comparison purpose. The hardware and time complexities of this structure and the proposed non-separable generic and unified structure are listed in Table VI. Compared with structure of [17], proposed non-separable generic structure and unified structure involve nearly $3.6L$ times more multiplier $3L$ times more adders and compute $4L$ and $6L$ times more outputs, respectively. Proposed generic structure involves $N(N + 1)$ less registers than that of [17] and it has nearly $4L$ times less MBWPO than other. Similarly, the proposed unified structure involves $(N^2 - N + 2)$ less registers and nearly $6L$ times less MBWPO.

TABLE VII
HARDWARE AND TIME COMPLEXITIES OF PROPOSED GENERIC AND UNIFIED STRUCTURES AND EXISTING UNIFIED STRUCTURE OF [17] FOR INPUT-IMAGE SIZE $M = 512$, FILTER-SIZE $N = 4$

<table>
<thead>
<tr>
<th>Structure</th>
<th>Filter output/cc</th>
<th>MULT</th>
<th>ADD</th>
<th>REG</th>
<th>MBWPO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chen [17]</td>
<td>1</td>
<td>11</td>
<td>23</td>
<td>1568</td>
<td>15</td>
</tr>
<tr>
<td>Proposed</td>
<td>16</td>
<td>152</td>
<td>240</td>
<td>1548</td>
<td>1.5</td>
</tr>
<tr>
<td>Generic (NS)</td>
<td>32</td>
<td>304</td>
<td>480</td>
<td>1548</td>
<td>1.125</td>
</tr>
<tr>
<td>Proposed Unified</td>
<td>24</td>
<td>200</td>
<td>288</td>
<td>1554</td>
<td>1.25</td>
</tr>
<tr>
<td>Unified (NS+S)</td>
<td>48</td>
<td>400</td>
<td>576</td>
<td>1554</td>
<td>0.875</td>
</tr>
</tbody>
</table>

LEGEND: NS: non-separable, S: separable.

Fig. 16. Normalized storage per filter output (calculated for $N = 4$, $L = 4$ and $M = 512$, NS, S, NSG, SG and UNI, respectively, stands for non-separable, separable, non-separable generic, separable generic and unified.
We have estimated hardware complexity of proposed non-separable generic and unified structures and the unified structure of [17] for $N = 4$, block-size $L = 4$, 8 and for image-size $M = 512$. The estimated values are listed in Table VII. We can find from Table VII that, proposed non-separable generic structure involves 13.6% less normalized multipliers, 34.7% less normalized adders and 90% less MBWPO than those of [17]. Proposed unified structure involves 24.27% less normalized multipliers, 47.82% less normalized adders and 94% less MBWPO than those of [17]. Unlike the existing unified structure, the proposed one does not require any steering logs for signal switching. Note that, the proposed non-separable generic structure is comprised of one general filter and three symmetric filters, while the proposed unified structure is comprised of one general non-separable and one general separable filter, and four symmetric filters. But, unified structure of [17] is comprised of only four symmetric filters. Proposed structures have several advantage over the existing structures, (i) provide filter-banks for non-separable and/or separable filters with symmetry and/or without symmetry which could be used in many image processing applications, (ii) can be configured for implementation of any one filter of the filter-bank or parallel configuration of any of the filters of the filter-bank, and (iii) can be easily scaled for high-throughput.

We have estimated normalized storage per filter output (SPO) of proposed structures and the existing structure of [11], [16], [17], [29] for $N = 4$, $L = 4$ and $M = 512$ and shown in the graph of Fig.16. We find that proposed structures involves less normalized SPO than the existing structure. This is mainly due to memory-reuse efficiency and memory-sharing of the proposed structures.

E. ASIC Synthesis Result

We have coded the proposed designs in VHDL for filter size $N = 4$, block-size $L = 4$ and image-size $(512 \times 512)$, and synthesized using Synopsys tool. We have also synthesized non-separable 2-D FIR filter structures extracted from IIR structure of [16] and unified structure of [17] and separable structure of [29]. We have used Wallace-tree based generic Booth-multiplier of Synopsys DesignWare building blocks library for all the designs. Shift-registers and registers of all designs are synthesized using D-FF (flip-flop). We have considered input signal width $w = 8$-bit and all intermediate signals and output signal width $w' = 16$-bit. We have set switching activity toggle rate 0.25 and static probability is 0.5. The netlist file obtained from the Synopsys Design Compiler is processed in IC Compiler. After place, route and clock synthesis area, time and power (leakage and dynamic) reported by the IC Compiler are listed in Table VIII for comparison. Power consumption of all the synthesized designs are estimated for 20MHz clock. Due to memory footprint reduction, the area, leakage power and dynamic power consumption of the proposed structures do not increase proportionately with the number of outputs. Consequently, proposed structures involve significantly less area-delay-product (ADP$^3$) and consume less energy per output (EPO$^4$). Compared with [16], proposed generic non-separable structure involves 11.25 times less ADP and 15.63 times less EPO. The proposed generic separable structure involves 4.85 times less ADP and 5.53 times less EPO than those of [29]. Compared with [17], proposed unified structure involves 4.64 times less area and 9.78 times less EPO.

VII. Conclusion

We have analyzed memory footprint and combinational complexity of 2-D FIR structures to arrive at a systematic design strategy to derive area-delay-power-efficient architectures. Based on that we have presented novel block-based separable and non-separable structures, generic structures for separable and non-separable filter-banks and unified structure for concurrent realization of both separable and non-separable filter-banks. It is shown that storage requirement of proposed structures does not change with input block-size ($L$). Similarly,
the storage size of generic non-separable structure is indepen-
dent of number parallel filters (P) in a filter-bank. It increases
marginally by \( P(N - 1) \) words in case of generic separable
and unified structures, where \( N \) is the filter size. Proposed
structures, therefore, offer higher memory reuse efficiency and
MBW reduction for higher values of \( L \) and \( P \). This reduces
SPO and PPO of proposed structures.

Compared with the existing structures, proposed block-
based separable and non-separable structures involve the
same number of storage words, and proportionately the same
or less number of arithmetic resources than the corresponding
best of existing structures, and compute \( PL \) times more filter
outputs per cycle with \( PL \) times less MBW. The proposed
unified structure with 4 non-separable filters and 2 separable
filters involves nearly 3.6L times more multipliers, 3L times
more adders, \( (N^2 - N + 2) \) less registers than existing unified
structure, and computes 6L times more filter outputs per cycle
with 6L times less MBW than other. ASIC synthesis result
for filter-size \((4 \times 4), \) input-block size \( L = 4 \) and image-size
\((512 \times 512) \) shows that, proposed block-based non-separable
and generic non-separable structures, respectively, involve
5.95 times and 11.25 times less ADP, and 5.81 times and
15.63 times less EPO than the corresponding existing
structures. The proposed unified structure involves 4.64 times
less ADP and 9.78 times less EPO than the corresponding
existing structure. When the inner-product computation
involving in FIR filtering is realized by multiplier-less designs
to reduce the combinational-complexity that may require
additional memory leading to overall increase in memory
complexity. But, the advantage gained by the proposed
memory footprint reduction technique is not affected by such
change of method of implementation of inner-products.

Acknowledgements: This publication was made possible by
an internal grant from Qatar University (QUUG-CENG-DCS-
11/12 – 7). The statements made herein are solely the respon-
sibility of the authors.

REFERENCES
[3] T. Barbu, Gabor filter based face recognition technique, in Proceeding of
the Romanian Academy, Series A, vol. 11, no.3/2010, pp. 277-283,
2010.
features based on Gabor filters, IEEE Transactions on Image Processing,
vol. 11, no. 10, pp. 1160-1167, OCT. 2002.
banks for efficient texture feature extraction, In Proc. 11th International
Conference on Control, Automation, Robotics and Vision, Singapore,
7-10th December 2010 , pp.1193-1197.
Transaction on Acoustic Speech and Signal Processing, vol. 37, pp.560-
systolic architecture for high-throughput implementation of 2-D FIR
implementation of bit-level systolic architecture for 1-D and 2-D digital
filters, IET Proc. Computer and Digital Technique, vol. 146, No. 2,
ture without global broadcast for 2-D systolic digital filters, in Proc.
broadcast, IEEE Transaction on Very Large Sclae Integration Systems,
filter design applications, IEEE Circuits and System Mag., vol. 3, no. 3,
diagonal-symmetry filter architecture design, in Proc. IEEE APCSAS,
architectures without global broadcast and some symmetry applications,
D fourfold-rotational-symmetry filter architecture design, in Proc. IEEE
formulation of 2-D filter structures without global broadcast for VLSI
implementation, in Proc., IEEE MWSCAS, Seattle, WA, Aug. 2010,
pp.426-529.
Power-efficient cost-effective 2-D symmetry filter architecture, IEEE
Transaction on Circuit ans Systems-I, Regular Papers, vol. 58, no.1,
pp.112-125, Jan. 2011.
subexpression elimination method for synthesizing fixed-point FIR fil-
[20] Y. Voronenko and M. Puschel, Multiplierless multiple constant multipli-
2007.
algorithm for realizing low complexity digital filters, IEEE Transaction
on Computer-Aided Design Integr Circuits Syst., vol. 24, no. 12,
[22] R. Mahesh and A. P. Vinod, A new common subexpression elimination
algorithm for realizing low complexity higher order digital filters, IEEE
Transaction on Computer-Aided Design Integr Circuits Syst., vol. 27,
[23] J. Luis, T.-Xihuitl, R. M.A.-Ponce and M. Bayoumi, Hybrid multiplier-
less FIR filter architecture based on NEDA, in Proc. IFIP International
Conference on Very Large Scale Integration (VLSI-Soc 2007), pp.316-
filters by efficient and flexible systolization using distributed arithmetic,
IEEE Transaction on Signal Processing, vol. 56, no. 7, pp.3009-3017,
July 2008.
[25] P. K. Meher, New approach to look-up table design and memory-
based realization of FIR digital filters, IEEE Transaction on Circuit and
[27] S. A. White, Applications of the distributed arithmetic to digital signal
processing: A tutorial review, IEEE ASSP Magazine, vol. 6, no. 3, pp.5-
19, July 1989.
multipliers, IEEE Transactions on Circuits and Systems II: Analog and
[29] B. K. Mohanty and P. K. Meher, New scan method and pipeline
architecture for VLSI implementation of separable 2-D FIR filters
without using transposition, in Proc. IEEE Region 10 TENCON2008