Abstract—In this paper, we present an efficient poly-phase decomposition scheme for implementation of 2-D non-separable filter bank. Poly-phase decomposition scheme offers multiplexing of filter bank computations or/and reduce the data clocking without affecting the overall throughput rate. Both these features can be used conveniently depending on resources availability or processor-technology. Time-multiplexing could be the choice for resource-constrained applications. Slower clocking rate could be chosen if processor-technology is the constraint. In that case, the design could be realized with cheaper and slower processor-technology. Time-multiplexed design needs proper data scheduling to perform filter bank computation interleavingly without data overlapping. Keeping this in mind, we have derived a systolic architecture for hardware realization of time-multiplexed filter bank where we have used novel data buffering scheme for the filter coefficients of the filter bank. Comparison result show that, the proposed structure involves almost J times less hardware resource than the non poly-phase filter bank structure and it provides the same throughput rate as the other, where J is the filter bank size. The hardware saving is significant for large size filter banks like Gabor. The proposed structure could be a good candidate for efficient hardware implementation of non-separable filter bank used in various image processing applications such as biometrics systems.

I. INTRODUCTION

Two (2-D) digital filters are widely used in many image processing applications like face detection, feature extraction, image enhancement and segmentation, edge detection and pattern recognition etc. [1]. 2-D finite impulse response (FIR) filters computation can be performed by separable or non-separable method depending on the 2-D filter impulse response matrix. Most of the image processing applications use 2-D FIR filters with non-separable impulse response matrix. Non-separable 2-D FIR filter is computationally intensive and most of its applications demand high-processing speed to deliver better performance. Keeping this in view, several architectures have been suggested for VLSI implementation of non-separable 2-D FIR filters [5], [6].

Non-separable filter banks are used frequently in image processing applications like face recognition, finger-print enhancement and feature extraction [2]. For example, Gabor filter bank is most common in biometric systems for detecting face [3], [4]. The size of 2-D filter and filter bank size used in biometric system is large. Consequently, huge amount of computation is performed by the filter bank and it involves large computation time which is not suitable for real-time application. On the other hand, there is large demand of prototype biometric systems from both civilian and military applications. These prototype systems can be easily deployed in remote locations. Most of the prototype equipments are constrained with resource and power. Efficient implementation of non-separable 2-D FIR filter bank in resource-constrained hand-held devices with capability for real-time processing is, therefore, a necessary challenge.

From the literature, we have not found any systematic approach to derive an efficient hardware structure for non-separable 2-D FIR filter banks of large sizes. Though few designs have been suggested for implementation of non-separable 2-D discrete wavelet transform (DWT) where filter bank size is 4 and small size impulse response matrix is used [7]. But the size of 2-D filter and the filter bank used in feature extraction is significantly higher than that of non-separable 2-D DWT. Realization of a bank of large number of non-separable 2-D FIR filter of higher size in a single chip is a huge challenging task. In this paper we have explored poly-phase decomposition scheme to derive an efficient structure for non-separable filter bank. We found that, poly-phase decomposition scheme not only helps for efficient realization of non-separable filter bank, it also offers a trade-off between hardware resource and processor technology without affecting the throughput rate of the filter bank. This has a potential design advantage to match the input sampling rate with slower processor technology and/or hardware resource.

The remainder of the paper is organized as follows: Necessary mathematical formulation using poly-phase decomposition scheme is presented in Section II. The proposed systolic architecture is presented in Section III. Hardware and time-complexity of the proposed structure is discussed in Section IV. Conclusion is presented in Section V.

II. MATHEMATICAL FORMULATION

In this Section, we develop the necessary mathematical formulation to derive the proposed architecture. The non-separable 2-D FIR filter algorithm [1] in z-domain is given by:

\[
H(z_1, z_2) = \sum_{l=0}^{L-1} \sum_{k=0}^{K-1} h(l, k) z_1^{-l} z_2^{-k}
\]  

(1)
where \( h(l, k) \) is the impulse response of the non-separable 2-D FIR filter of size \((L \times K)\). The input-output relation of the non-separable 2-D FIR filter in \( z \)-domain is:

\[
Y(z_1, z_2) = H(z_1, z_2)X(z_1, z_2)
\]

(2)\n
\( X(z_1, z_2) \) and \( Y(z_1, z_2) \) are, respectively, the \( z \)-transform of the input and output matrix each of size \((M \times N)\).

Suppose we have \( J \)-channel filter-bank as shown in Fig.1, where \( J \) number of non-separable 2-D FIR filters are connected in parallel. Each 2-D filter of the filter-bank generates one sub-band matrix of size \((M/D_1 \times N/D_2)\), and \( J \) filters generate \( J \) sub-band matrices \( Y^j \), for \( 0 \leq j \leq J - 1 \), form the 2-D input matrix \( X \), where \( D_1 \) and \( D_2 \) are, respectively, row and column down-sampling factors and \( J = D_1 D_2 \) [2].

One shift-register (SR) of \( N \) words. As shown in Fig.2, the direct form-1 structure consists of \( L \) identical sections, \((L-1)\) SRs and \((L-1)\) adders, where each section is comprised of \( K \) multipliers corresponding to one row of impulse response matrix \( \{h^j(l, k)\} \), for \( 0 \leq l \leq L - 1 \) and \( 0 \leq k \leq K - 1 \), \((K - 1)\) adders and equal number of registers. Each 2-D FIR filter of the filter bank, therefore, involves \( LK \) multipliers, \( L(K - 1) \) adders and \( N(L - 1) + L(K - 1) \) registers, and computes one output in every cycle. The filter outputs are column-wise downsampled by factor \( D \) (only one filter output is retained out of every set of \( D \) outputs of each row of filter output). The resulted 2-D filter output matrix again downsampled along row-wise with factor \( D \) for the sub-band matrix. Due to column and row down-sampling, one filter output out of a block of \( D^2 \) outputs corresponding to \( D \) rows and \( D \) columns is only retained for the sub-band matrix. Each filter of the filter bank, therefore, generates one components of the sub-band matrix in every \( D^2 \) cycles and the entire sub-band matrix \( [Y^j] \) in \( N^2 \) cycles.

The 2-D filter bank with \( J \) filters, therefore, involves \( JKL \) multipliers, \( JL(K - 1) \) adders and \( J[N(L - 1) + L(K - 1)] \) registers. All \( J \) filters of the filter bank perform computation in parallel and require \( N^2 \) cycles to generate \( J \) sub-band matrices of the input matrix. Since, rows and columns down-sampling operations are applied post-filtering operation, this results a wastage of resource and time. For example if \( D = 2 \) or \( J = 4 \), then each filter retains only 25% of the computed filter outputs for the sub-band. In other words, 75% of the input power, time and resource are wasted to generate 25% of the useful outputs. The case is even worst for \( D = 4 \) or \( J = 16 \), where nearly 95% of the power, resource and time is wasted. In many image processing applications like facial feature extraction in biometric systems, filter banks of size 32 and 40 are used. In those cases, almost 99% of input power, resource and time is

\[
Y^j(z_1, z_2) = X(z_1, z_2)H^j(z_1, z_2)
\]

(3)

where, \( H^j(z_1, z_2) \) is the system function of \((j + 1)\)-th filter of the filter-bank and its system function is defined as;

\[
H^j(z_1, z_2) = \sum_{l=0}^{L-1} \sum_{k=0}^{K-1} h^j(l, k) \cdot z_1^{-l} \cdot z_2^{-k}
\]

(4)

A. Complexity of 2-D Filter Bank

The direct form-1 structure for realization of system function \( H^j(z_1, z_2) \) of \((j + 1)\)-th 2-D FIR filter of filter bank is shown in Fig.2, where \( z_1^{-1} \) and \( z_2^{-1} \), respectively, represent row and column delay of the filter. For row-by-row data input, one column delay is equal to one cycle delay and one-row delay is equal to \( N \) cycles delay. The column delay is, therefore, realized using one register and the row-delay using

\[1\] Square input matrix is used in most of the practical application.

\[2\] This notation signifies that the input matrix is square.

\[3\] The down-sampling factors are also equal (\( D_1 = D_2 = D \)) as the combined size of all the sub-band matrices is the same as the input image size. Input-output relation of \((j + 1)\)-th filter of the filter-bank is given by:

\[
Y^j(z_1, z_2) = X(z_1, z_2)H^j(z_1, z_2)
\]

The block diagram of \( J \)-channel 2-D filter-bank is shown in Fig.1. Without loss of generality, we assume square input matrix\(^1\) \( i.e. \ M = N \). The down-sampling factors are also equal (\( D_1 = D_2 = D \)) as the combined size of all the sub-band matrices is the same as the input image size. Input-output relation of \((j + 1)\)-th filter of the filter-bank is given by:

\[
Y^j(z_1, z_2) = X(z_1, z_2)H^j(z_1, z_2)
\]

(3)
wasted. The direct form filter-bank structure is not good for hardware implementation of filter banks where speed, power and space are the performance parameters.

A poly-phase type decomposition [2] could be considered to derive structures of 2-D filters of filter bank where the down-sampling operations (row and column) could be performed prior to the filter computation (per-filter operation) similar to 1-D poly-phase structures for interpolator and decimators [2]. This has two fold advantages (i) down-sampled outputs can be skipped and filter computation only performed on the inputs whose outputs are required by the sub-band matrix, (ii) to save wastage of power, the data clocking of the filter could be reduced and equal to output sampling rate which is less by 1/D² times than the input sampling rate, and (iii) it favors multiplexing the computation of different filters of filter bank to reduce the wastage of resource. Poly-phase approach is, therefore, most appropriate method to derive an efficient structure for hardware implementation of 2-D filter bank. In the next section we have developed a mathematical formulation for the poly-phase decomposition of non-separable 2-D FIR filters.

B. Poly-phase Decomposition of 2-D FIR Filter

Equation (4) can be rewritten as

$$H^j(z_1, z_2) = \sum_{l=0}^{L-1} z_1^{-l} \cdot H^j_l(z_2)$$

(5)

where,

$$H^j_l(z_2) = \sum_{k=0}^{K-1} h^j(l, k)z_2^k$$

(6)

Equation (6) can be decomposed and expressed as;

$$H^j_l(z_2) = \sum_{p=0}^{D-1} z_2^{-p} H^j_{l,p}(z_2)$$

(7)

where,

$$H^j_{l,p}(z_2) = \sum_{q=0}^{Q-1} h^j(l, pD + q)z_2^{-(pD+q)}$$

(8)

where Q is an integer factor of K and Q = K/D. If K is not an integer multiple of Q, then appropriate zeros to be padded with the impulse response matrix.

Using (5), each section of Fig.2 is decomposed and realized using D identical sub-filters. The structure for realization of system function $H^j_l(z_2)$ of (l + 1)-th section of j-th 2-D filter is shown in Fig.3(a). Direct form-1 structure of (p + 1)-th sub-filter of $H^j_l(z_2)$ is shown in Fig.3(b). As shown in Fig.3(b), each delay element is clocked at D times slower rate than the input sampling rate. Each sub-filter, therefore, operate at D times slower rate than the input sampling frequency. This helps to move the column down-sampling factor from the output side of Fig.2 to the input side as shown in Fig.4.

The filter bank structure using the poly-phase 2-D FIR filters is shown in Fig.5, where both the row and the column down-sampling operations are moved to the input side of the 2-D filter. In poly-phase structure, input samples of those down-
sampled filter outputs are skipped. Since, both the row and the column down-sampling are performed on the input sample, each filter remains ideal during the down-sampling cycles as the filter is clocked at the input sampling frequency. Due to column down-sampling factor \((D)\), each filter skip filter computation of \((D−1)\) input samples after every sample of each row. Further, \((D−1)\) successive input rows are skipped by the filter after every one row of input. Each filter, therefore, remain ideal for \([N^2−(N/D)^2]\) cycles in a set of \(N^2\) cycles. On the other hand, each filter requires only \((N/D)^2\) cycles to compute the sub-band matrix. Down-sampled filter computation of \(D^2\) filters can easily be time-multiplexed and performed using one filter without any data overlapping. This results fully utilization of the row and column down-sampling cycles of one filter. In other word, all the \(J = D^2\) filters can be realized using the hardware resource of one filter unit. Alternately, one can eliminate the ideal clock cycles by reducing the operating frequency of each filter by a factor of \(D^2\). In this case all the filters of the filter bank are realized separately and the filter bank involves \(J\) times more resources than the time-multiplexed design. But it operates at \(D^2\) slower clock frequency. This design is suitable to implement the filter bank structure using slower processor technology without affecting the throughput rate of the filter bank. One can have partly time-multiplexed and partly parallel filter bank structure to meet the processor technology which is critical to a design. Therefore, using the poly-phase design scheme, one can have the flexibility to trade processor technology with area without affecting the throughput rate of the filter bank. This property is very useful to implement the design in different platforms delivering the required throughput rate. Since, time-multiplexed design needs a proper data scheduling scheme, we discus this issue in the next section and presented a fully systolic architecture for the filter bank.

III. PROPOSED ARCHITECTURE

Row-column computation of a set of \(D\) filter are time-multiplexed and performed interleavingly to utilize the column down-sampling cycles. Similarly, row-column computation of \(D\) such sets of filters are time-multiplexed during the row down-sampling cycles and performed interleavingly. Row-column computation of all the \(J = D^2\) non-separable \(2\)-\(D\) FIR filters are time-multiplexed and performed interleavingly in one filter. In this section we have derived a systolic architecture to perform computations of time-multiplexed filter bank.

Proposed structure for \(2\)-\(D\) non-separable filter bank of size \(J\) is shown in Fig.6(a). It consists of \(L\) identical sections. Each section is further consists of \(D\) sub-sections. Structure of \((l+1)\)-th section is shown in Fig.6(b) and structure of its \((p+1)\)-th subsection is shown in Fig.7. Computations pertaining to system functions \(H_{l,j}^{(p)}\) for \(0 \leq j \leq J−1\), corresponding to \((p+1)\)-th subsection (sub-filter) of \((l+1)\)-th section of \(j\)-th filter are time-multiplexed and mapped to the \((p+1)\)-th subsection of the proposed structure. the \((p+1)\)-th subsection computes sub-filter outputs of all the \(J\) filters interleavingly. Each buffer (BUF) of the subsection stores the impulse response of a particular row and column of all the \(J\) filters. As shown in Fig.7, each subsection comprises of \(Q\) multipliers, \(Q\) BUFS and one pipelined adder-tree (PAT). The \((q+1)\)-th BUF (for \(0 \leq q \leq Q−1\)) stores the \((pD+q+1)\)-th column of \((l+1)\)-th row of impulse response matrices \(h_{l,k}^{(p)}\), for \(0 \leq l \leq L−1\), and \(0 \leq k \leq K−1\) and \(K = DQ\). Structure of \((q+1)\)-th BUF of \((p+1)\)-th subsection of \((l+1)\)-th section of the proposed structure is shown in Fig.8. It consists of \(J = D^2\) registers arranged in \(2\)-\(D\) square array. Each row of registers store filter constants of a set of \(D\) filters, such that \((r+1)\)-th row of registers store filter constants \(\{h_{r,D,l}(l,pD+q), h_{r,D+1,l}(l,pD+q), \ldots, h_{r,D+s−1,l}(l,pD+q)\}\) of \((r+1)\)-th set of \(D\) filters (for \(0 \leq r \leq D−1\) and \(0 \leq s \leq D−1\)). Each row of registers of the BUF forms a circulating buffer, where the values stored in a particular row of registers circulate anti-clockwise direction. Each register transfer the stored value to the adjacent register to its right after a gap of every \(D\) cycle. Values stored in the right-most register of each row are selected by one \(D\)-to-1 line multiplexer (MUX). MUX selects input lines after every clock cycle. Suppose, during a particular set of \(D\) cycles, MUX selects filter constants of filter-1 to filter-\(D\) from the BUF, then in the next set of \(D\) cycles, it selects filter constants of filter-(\(D+1\)) to filter-\(2D\) from the same BUF. Like this, in \(D^2\) successive cycles MUX selects one filter constant (of a particular row and column) of each \(J\) filters of the filter bank.

During each cycle, \(Q\) filter constants of a particular row of
All the input rows of SR array in $DN$ cycles. Note that, during successive sets of $N$ cycles, $(L-1)$ input rows are overlapped. Consequently, the same $L$ input rows are repeated after every sets of $N$ cycles. During first $N$ cycles, $L$ consecutive row of input are obtained from the SR-DSU, for first set of $D$ filters, and the same set of $L$ input rows are obtained once again from the SR-DSU during next set of $N$ cycles for the second set of $D$ filters (filter-$D+1$ to filter-$2D$). This process is repeated and same set of $L$ input rows are obtained from the SR-DSU during $D$-th set of $N$ cycles for the $D$ set of filters (filter-$D(D-1)+1$ to filter-$D^2$). In other words, SR-DSU functions like a repeater which repeat $L$ input rows for $D$ successive sets of $N$ cycles.

All the $L$ sections of proposed structure receive $L$ input rows from the SR-DSU in parallel and generate $L$ partial outputs corresponding to $L$ output rows of 2-D filter. These partial outputs are added using a separate PAT for obtaining one complete 2-D filter output. The proposed structure calculates one output of a particular row and column of one sub-band in every cycle and it calculates one component each of $J$ sub-bands of same row and same column in successive $J$ cycles. It calculates one complete row if size $N/D$ of $J$ sub-bands in $NJ/D$ cycles. The entire sub-band matrix of all the $J$ filters of each size $(N/D)$ times $N/D$ are obtained in $N^2$ cycles. The time-multiplexed sub-band outputs are demultiplexed using an output data-selector unit (ODSU) shown in Fig.9(b). It consists of $(D+1)$ 1-to-$D$ line de-multiplexes (DMUXes). DMUX-1 split the sub-band components of $D$ sets of filters where DMUX-2 to DMUX-9 are used to split sub-band components of individual filter in each set of $D$ filters.

IV. HARDWARE AND TIME COMPLEXITIES

In this section we have calculated hardware and time complexities of the proposed structure and the non poly-phase structure for comparison.

A. Proposed Structure

Proposed structure consists of $L$ sections, one PAT of size $L$ words, one SR-DSU and one ODSU. Each section is further comprised of $D$ subsections, one PAT of size $D$ words and
(D – 1) registers, and each subsection comprises of Q BUFs, Q multipliers, (Q – 1) registers and one PAT of size Q words. Each BUF comprises of D^2 registers and D MUXes. A PAT of size P words involves (P – 1) two-input adders and (P – 2) pipeline registers. Each subsection, therefore, comprises of Q multipliers, (Q – 1) adders, [D(QD + 1) + Q – 3 registers and QD MUXes. Each section of the structure involves DQ multipliers, (DQ – 2) adders, \{D[D(QD + 1) + Q – 1] – 3\} registers and QD MUXes.

### B. Non Poly-Phase Filter Bank Structure

The non poly-phase filter bank structure consists of J identical blocks corresponding to J filters. Each block consists of L sections, one PAT of size L words and one SR-array. Each section is further comprised of D subsections, one PAT of size D words and (D – 1) registers. Each subsection comprises of Q multipliers, (Q – 1) adders and (D + Q – 3) registers. Each section, therefore, comprises DQ multipliers, (DQ – 2) adders and \{D[D + Q – 1] – 3\} registers. Each block of the filter bank involves LDQ multipliers, L(DQ + 1) – 3 adders and \{L[D + Q – 1] + Q(N – 1) – 2\} registers.

### C. Performance Comparison

Hardware and time complexities of the proposed structure and the non-poly-phase filter bank structure are listed in Table I for comparison. We find that proposed structure involves almost J times less hardware resource than the non poly-phase filter bank structure and it provides the same throughput rate as the other, where J is the filter bank size. The hardware saving is huge if the proposed structure in used to realize large size filter banks like Gabor. The proposed structure could be a good choice for efficient hardware implementation of non-separable filter bank used in various image processing applications.

### 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 responsibility of the authors.

### References


[5] B. K.Mohanty and P. K. Meher, Cost-effective novel flexible cell-vlevel symbolic architecture for hardware realization of time-multiplexed filter bank where we have used novel data buffering scheme for accessing the filter coefficients of the filter bank. Comparison result shows that, the proposed structure involves almost J times less hardware resource than the non poly-phase filter bank structure and it provides the same throughput rate as the other, where J is the filter bank size. The hardware saving is huge if the proposed structure in used to realize large size filter banks like Gabor.

### Table I

<table>
<thead>
<tr>
<th>Structures</th>
<th>Multiplier</th>
<th>Adder</th>
<th>Registers</th>
<th>MUX/DMUX</th>
<th>Computation time (in cc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Poly-phase Structure</td>
<td>LDQ</td>
<td>L[(DQ + 1) – 3]</td>
<td>D[L(DQ + 1) + Q – 1] (+ N(L + D + 2) - 2L - 2]</td>
<td>L(QD^2 + 1) + D + 1</td>
<td>(N^2)</td>
</tr>
<tr>
<td>Non Poly-phase Structure</td>
<td>JLDQ</td>
<td>JL(DQ + 1) – 3</td>
<td>J[L[D + Q – 1] – 2] (+ N(L – 1) – 2]</td>
<td>0</td>
<td>(N^2)</td>
</tr>
</tbody>
</table>

We have presented an efficient poly-phase decomposition scheme for implementation of 2-D non-separable filter bank. Poly-phase decomposition scheme offers multiplexing of filter bank computations or/and reduce the data clocking without affecting the overall throughput rate. Both these features can be used conveniently depending on availability of resources or processor-technology. Time-multiplexing could be the choice for resource-constrained applications. Slower clocking rate could be chosen if processor-technology is the constraint. In that case, the design could be realized with cheaper and slower processor-technology. Time-multiplexed design needs proper data scheduling to perform filter bank computation interleavingly without data overlapping. Keeping this in mind, we have derived a systolic architecture for hardware realization of time-multiplexed filter bank where we have used novel data buffering scheme for accessing the filter coefficients of the filter bank. Comparison result shows that, the proposed structure involves almost J times less hardware resource than the non poly-phase filter bank structure and it provides the same throughput rate as the other, where J is the filter bank size. The hardware saving is huge if the proposed structure in used to realize large size filter banks like Gabor.

### References


[5] B. K.Mohanty and P. K. Meher, Cost-effective novel flexible cell-vlevel symbolic architecture for hardware realization of time-multiplexed filter bank where we have used novel data buffering scheme for accessing the filter coefficients of the filter bank. Comparison result shows that, the proposed structure involves almost J times less hardware resource than the non poly-phase filter bank structure and it provides the same throughput rate as the other, where J is the filter bank size. The hardware saving is huge if the proposed structure in used to realize large size filter banks like Gabor.

### 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 responsibility of the authors.

### References


[5] B. K.Mohanty and P. K. Meher, Cost-effective novel flexible cell-vlevel symbolic architecture for hardware realization of time-multiplexed filter bank where we have used novel data buffering scheme for accessing the filter coefficients of the filter bank. Comparison result shows that, the proposed structure involves almost J times less hardware resource than the non poly-phase filter bank structure and it provides the same throughput rate as the other, where J is the filter bank size. The hardware saving is huge if the proposed structure in used to realize large size filter banks like Gabor.