CHAPTER 1

INTRODUCTION

The wavelet transform (WT) provides a time-frequency representation of the signal. It was developed to overcome the short coming of the short time Fourier transform (STFT), which can also be used to analyze non-stationary signals. While STFT gives a constant resolution at all frequencies, the WT uses multiresolution technique by which different frequencies are analyzed with different resolutions. The wavelet analysis is done similar to the STFT analysis. The signal to be analyzed is multiplied with a wavelet function just as it is multiplied with a window function in STFT, and then the transform is computed for each segment. A wavelet is a short oscillating function which contains both analysis function and the window function. In WT, time information is obtained by shifting the wavelet over the signal, while the frequencies are changed by contraction and dilatation of the wavelet function. The continuous wavelet transform (CWT) retrieves the time-frequency content information with an improved resolution compared to the STFT [Louis et al. (1997)].

Discrete wavelet transform (DWT) is a mathematical technique that provides a new method for signal processing and decomposes a discrete signal in the time domain by using dilated / contracted and translated versions of a single basis function, named as prototype wavelet [Mallat (1989a) ; Mallat (1989b) ; Daubachies (1992) ; Meyer (1993) ; Vetterli and Kovacevic (1995)]. DWT offers wide variety of useful features over other unitary transforms like discrete Fourier transforms (DFT), discrete cosine transform (DCT) and discrete sine transform (DST). Some of these features are; adaptive time-frequency windows, lower aliasing distortion for signal processing applications, efficient computational complexity and inherent scalability [Grzesczak et al. (1996)]. Due to these features one dimensional (1-D) DWT and two dimensional (2-D) DWT are applied in various application such as numerical analysis [Beylkin et al. (1992)], signal analysis [Akanshu and Haddad (1992)], image coding [Sodagar et al. (1999); Taubman (2000)], pattern recognition [Kronland et al. (1987)], statistics [Stoksik et al. (1994)] and biomedicine [Senhadji et al. (1994)]. Several algorithms and computation schemes have been suggested during last three decades for efficient hardware implementation of 1-D DWT and 2-D DWT. Computation schemes commonly used in hardware implementation are briefly discussed here in the next section.
1.1 Computation Scheme for 1-D DWT

In DWT, the input signal is decomposed into two subbands known as low-pass subband and high-pass subband. The low-pass and high-pass subband components of a particular DWT decomposition level is obtained by filtering the input signal using a pair of low-pass and high-pass filter. The low-pass and high-pass filter pair forms a quadrature mirror filter (QMF) for perfect signal reconstruction. The low-pass and high-pass filters are short length finite impulse response (FIR) filter. As shown in Figure 1.1, the low-pass filter output is down-sampled to obtained the low-pass subband output \( u_l(n) \). Similarly, the high-pass filter output is down-sampled to obtained the high-pass subband output \( u_h(n) \). The 1-D DWT computation is equivalent to a two channel down-sampled FIR filter computation. The filtering unit (FU) of 1-D DWT constitutes a pair of filters (low-pass filter (LPF) and high-pass filter (HPF)), and a pair of down samplers. The low-pass and high-pass filter outputs are calculated using two computation schemes known as (i) convolution scheme and (ii) lifting scheme. These computation schemes are discussed briefly in section 1.1.1 and 1.1.2.

![Filtering Unit (FU)](image)

**Figure 1.1:** Computation of one level 1-D DWT.

1.1.1 Convolution Scheme

In convolution scheme, the low-pass and high-pass filter output of an FU are calculated using the expressions

\[
u_l(n) = \sum_{i=0}^{h-1} h(i)x(2n-i)
\]

(1.1)
\[ u_k(n) = \sum_{i=0}^{k_1-1} g(i) x(2n - i) \] 

where, \( k_1 \) is the length of low-pass filter, \( k_2 \) is the length of high-pass filter, \( x(n) \) is the input signal. \( u_l(n) \) and \( u_h(n) \) are the low-pass and high-pass subband components, respectively. \( h(n) \) and \( g(n) \) are, respectively, low-pass and high-pass filter coefficients of wavelet filter.

Wavelet filters are classified as, orthogonal and biorthogonal wavelets. The wavelet filter coefficients satisfy the orthogonal property is known as orthogonal wavelet, where the biorthogonal wavelet filter coefficients satisfy the orthonormal property in addition to orthogonal property. The orthogonal low-pass and high-pass filters are, asymmetric and have same lengths, where the low-pass and the high-pass filters of biorthogonal wavelet are symmetric and different in length [Rao and Bopardikar (1999)].

### 1.1.2 Lifting Scheme

The lifting scheme was proposed by Sweldens (1996). According to lifting scheme, computation of an FU of 1-D DWT can be factored into lifting steps. The basic principle of lifting scheme is to factorize the polyphase matrix \( (P(z)) \) of wavelet filters into a sequence of alternating upper and lower triangular matrices and a constant diagonal matrix. This leads to wavelet computation by means of banded-matrix multiplications [Daubechies and Sweldens (1998)]. The lifting based DWT has many useful properties such as symmetric forward and inverse transform, in-place computation, integer-to-integer transform and requires less computation than convolution based DWT [Acharya and Chakrabarti (2006)].

Let \( H(z) \) and \( G(z) \) are the system function of low-pass and high-pass wavelet filters. \( H(z) \) can be decomposed into \( H_e(z) \) and \( H_o(z) \), where \( H_e(z) \) and \( H_o(z) \) represents the system function of even and odd part of the impulse response \( h(n) \) of low-pass wavelet filter. Similarly, \( G_e(z) \) and \( G_o(z) \) represents the system function of even and odd part of the impulse response \( g(n) \) of high-pass wavelet filter. The system function of FU can be represented in polyphase matrix as

\[
P(z) = \begin{bmatrix}
H_e(z) & G_e(z) \\
H_o(z) & G_o(z)
\end{bmatrix}
\]  

(1.3)
The matrix $P(z)$ can be factorized into lower, upper and diagonal matrix as [Daubechies and Sweldens (1998)]

$$P(z) = \prod_{i=1}^{i=m} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix}$$ (1.4)

where $K$ is the scaling constant, $s_i(z)$ and $t_i(z)$ are the system function of predict and update unit of $i$-th lifting step. Each predict and update stage represents one lifting step of DWT. For example: lifting computation of an FU using 9/7 biorthogonal wavelet filter is expressed in four lifting step as

$$\begin{bmatrix} u_i(n) \\ u_s(n) \end{bmatrix} = \begin{bmatrix} x(2n) \\ x(2n-1) \end{bmatrix} P(z)$$ (1.5)

and

$$P(z) = \begin{bmatrix} 1 & \alpha(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma(1+z^{-1}) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & K^{-1} \end{bmatrix}$$ (1.6)

where $\alpha$, $\beta$, $\gamma$ and $\delta$ are lifting constants.

Substituting (1.6) in (1.5), the input-output of lifting DWT of 9/7 wavelet filter are expressed in the recursive form as

$$s_1(n) = x(2n-1) + \alpha(x(2n) + x(2n-2))$$ (1.7)

$$s_2(n) = x(2n-2) + \beta(s_1(n) + s_1(n-1))$$ (1.8)

$$v_1(n) = s_1(n-1) + \gamma(s_2(n) + s_2(n-1))$$ (1.9)

$$v_2(n) = s_2(n-1) + \delta(v_1(n) + v_1(n-1))$$ (1.10)

$$u_s(n) = K^{-1}v_1(n)$$ (1.11)

$$u_i(n) = Kv_2(n)$$ (1.12)

for $0 \leq n \leq \left(\frac{N}{2}\right) - 1$ and $N$ is the length of input signal $x(n)$. 

4
1.2 Multilevel 1-D DWT

Multiresolution analysis (MRA) is a characteristic feature of DWT and it is used for better spectral representation of the signal. In MRA, the signal is decomposed for more than one DWT level known as multilevel DWT. It means the low-pass output of first DWT level is further decomposed in a similar manner in order to get the second level of DWT decomposition and the process is repeated for higher DWT levels. Few algorithms have been suggested for computation of multilevel DWT. These algorithms are briefly discussed here.

1.2.1 Pyramid Algorithm

Mallat (1989a) has proposed a pyramid algorithm (PA) for parallel computation of multilevel DWT. PA for 1-D DWT is given by

\[ u^j_l(n) = \sum_{i=0}^{N-1} h(i)u^{j-1}_l(2n-i) \]  
\[ u^j_h(n) = \sum_{i=0}^{N-1} g(i)u^{j-1}_h(2n-i) \]

where \( u^j_l(n) \) is the \( n \)-th low-pass subband component of the \( j \)-th DWT level and \( u^j_h(n) \) is the \( n \)-th high-pass subband component of the \( j \)-th DWT level, for \( n \leq 0 \leq \left( \frac{N}{2} \right) - 1 \), \( u^0_l \) represents zeroth level low-pass subband component which represents the input signal \( x(n) \).

Figure 1.2 shows the computation of three level 1-D DWT using PA. The low-pass and high-pass subband output at each DWT levels are calculated by using the low-pass subband output of the previous DWT level.

![Figure 1.2: Computation of three level 1-D DWT using pyramid algorithm (PA).](image-url)
Due to repeated down sampling of the signal, the amount of computation after every decomposition level decreases steadily by a factor of 2. This results in low hardware utilization when the PA is implemented in hardware. Under utilization of the resource is measured in terms of hardware utilization efficiency (HUE). The HUE is defined as actual computation time to the busy time of the corresponding section, where time is expressed in number of clock cycles [Liao et al. (2004)]. HUE of PA based 1-D DWT structure is given by

$$U = \frac{2N(1-2^{-J})}{JN} \times 100\%$$  \hspace{1cm} (1.15)

HUE of PA-based 1-D DWT structure for $J = 2, 3, 4$ and 5 is found to be, 75%, 58%, 46% and 38%, respectively. HUE is a major concern of PA-based multilevel 1-D DWT structure. To overcome this problem, Vishwanath (1994) has proposed a recursive pyramid algorithm (RPA).

### 1.2.2 Recursive Pyramid Algorithm

RPA is a reformulation of PA which allows DWT computation in a real time fashion using single FU. In RPA, computation of higher DWT levels are time multiplexed with the first level DWT. A general block diagram of 3-level RPA based DWT is shown in Figure 1.3. RPA rearrange the order of $N$ outputs such that an output is scheduled at the “earliest” instance that it can be scheduled. The “earliest” instance is decided based on a strict precedence relation, i.e., if the “earliest” instance of the $j$-th octave clashes with that of the $(j + 1)$-th octave, then the $j$-th octave output is scheduled. The low-pass output schedule generate by the RPA (for $N = 16$ and $J = 3$) is given in Table 1.1, where ‘-’ represents no output is scheduled. The high-pass output schedule is exactly the same as the low-pass output given in Table 1.1. The HUE of RPA-based $J$-th level 1-D DWT structure is calculated using the relation

$$U = \frac{N + N(1-2^{-J})}{2N} \hspace{1cm} (1.16)$$
HUE of RPA based 1-D DWT structure for \( J = 2, 3, 4 \) and 5 is calculated using the formula given in (1.16) and it found to be 75\%, 87\%, 93\% and 96\%, respectively. The HUE of RPA based 1-D DWT structure is much higher than the HUE of PA-based DWT structure for the same decomposition level.

![Figure 1.3: Computation of three level 1-D DWT using recursive pyramid algorithm (RPA).](image)

**Table 1.1:** The low-pass output of RPA based DWT schedule for \( J = 3 \) and \( N = 16 \)

<table>
<thead>
<tr>
<th>Clock cycle</th>
<th>First level output</th>
<th>Second level output</th>
<th>Third level output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( u_1^0(0) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>( u_1^2(0) )</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>( u_1^1(1) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>( u_1^3(0) )</td>
</tr>
<tr>
<td>5</td>
<td>( u_1^1(2) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>( u_1^2(1) )</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>( u_1^1(3) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>( u_1^1(4) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-</td>
<td>( u_1^2(2) )</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>( u_1^1(5) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>-</td>
<td>-</td>
<td>( u_1^3(1) )</td>
</tr>
<tr>
<td>13</td>
<td>( u_1^1(6) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>-</td>
<td>( u_1^2(3) )</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>( u_1^1(7) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
1.3 Computation Scheme for 2-D DWT

Two-dimensional signal, such as images, are analyzed using the 2-D DWT. Currently 2-D DWT is applied in many image processing applications such as image compression and reconstruction [Lewis and Knowles (1992)], pattern recognition [Kronland et al. (1987)], biomedicine [Senhadji et al. (1994)] and computer graphics [Meyer (1993)]. The 2-D DWT is a mathematical technique that decomposes an input image in the multiresolution frequency space. The 2-D DWT decomposes an input image into four subbands known as low-low (LL), low-high (LH), high-low (HL) and high-high (HH) subband. The 2-D DWT uses 2-D wavelet basis function \( \{ h_{l_l}(p,q), h_{l_h}(p,q), h_{h_l}(p,q), h_{h_h}(p,q) \} \) of size \( (L \times L) \) to decompose the input 2-D signal into four subbands \{LL, LH, HL and HH\}.

The 2-D DWT can be computed by two approaches (i) non-separable (direct) and (ii) separable (indirect) [Yu and Chen (1997); Week and Bayoumi (2003)]. The non-separable 2-D DWT computation is performed using non-separable wavelet basis function, where the separable 2-D DWT computation is performed using separable wavelet basis function. Separable wavelet basis function is expressed as a product of 1-D basis function

\[
\begin{align*}
    h_{l_l}(p,q) &= h_1(p)h_2(q) \\
    h_{l_h}(p,q) &= h_1(p)g_2(q) \\
    h_{h_l}(p,q) &= g_1(p)h_2(q) \\
    h_{h_h}(p,q) &= g_1(p)g_2(q)
\end{align*}
\]

where \( 0 \leq (p,q) \leq L-1 \), \( h_1(p) \) and \( h_2(q) \) are, respectively, the low-pass wavelet filter coefficient of row and column separable 2-D DWT. Similarly, \( g_1(p) \) and \( g_2(q) \) are respectively the high-pass wavelet filter coefficient of row and column DWT.

1.3.1 Non-Separable Approach

According to Mallat (1989a) PA, DWT coefficient of any decomposition level can be obtained from the scaling coefficient of its previous level. Therefore, PA of 2-D DWT with non-separable wavelet basis function is given by the following equations

\[
\begin{align*}
    h_{l_l}(p,q) &= h_1(p)h_2(q) \\
    h_{l_h}(p,q) &= h_1(p)g_2(q) \\
    h_{h_l}(p,q) &= g_1(p)h_2(q) \\
    h_{h_h}(p,q) &= g_1(p)g_2(q)
\end{align*}
\]
where $0 \leq m \leq \left(\frac{M}{2^j}\right)^{-1}$, $0 \leq n \leq \left(\frac{N}{2^j}\right)^{-1}$ and $A^j(m, n)$, $B^j(m, n)$, $C^j(m, n)$ and $D^j(m, n)$ are, respectively, the coefficient of the low-low ($A^j$), low-high ($B^j$), high-low ($C^j$) and high-high ($D^j$) subband matrices of $j$-th decomposition level of size $\left(\frac{M}{2^j} \times \frac{N}{2^j}\right)$ each. The zeroth level non-separable low-low subband ($A^0$) represents the input matrix ($X$) of size ($M \times N$).

**Figure 1.4:** General block diagram of non-separable filtering unit (NSFU).
In the non-separable approach, the DWT of 2-D signal is computed by performing four separate 2-D filtering operations using four 2-D filters of wavelet basis function \( h_1(p,q), h_{ih}(p,q), h_{id}(p,q) \) and \( h_{ih}(p,q) \). The output signals of these four filters are then decimated by a factor of two along the row and column direction to obtain four subband matrices. The low-low, low-high, high-low and high-high 2-D wavelet filters constitute a non-separable filtering unit (NSFU). A NSFU transform 2-D input signal into four subband matrices. General block diagram of NSFU is shown in Figure 1.4. According to equation (1.2.1) – (1.2.4), the non-separable 2-D DWT computation involves \( L^2 \) arithmetic operations (multiplication and addition).

### 1.3.2 Separable Approach

PA for separable 2-D DWT with separable wavelet basis function is given by

\[
a^l(m,n) = \sum_{p=0}^{L-1} \sum_{q=0}^{L-1} h_1(p)h_2(q)a^{l-1}(2m - p, 2n - q) \quad (1.25)
\]

\[
b^l(m,n) = \sum_{p=0}^{L-1} \sum_{q=0}^{L-1} h_1(p)g_2(q)a^{l-1}(2m - p, 2n - q) \quad (1.26)
\]

\[
c^l(m,n) = \sum_{p=0}^{L-1} \sum_{q=0}^{L-1} g_1(p)h_2(q)a^{l-1}(2m - p, 2n - q) \quad (1.27)
\]

\[
d^l(m,n) = \sum_{p=0}^{L-1} \sum_{q=0}^{L-1} g_1(p)g_2(q)a^{l-1}(2m - p, 2n - q) \quad (1.28)
\]

where \( 0 \leq (p,q) \leq L - 1 \), \( h_1(p) \) and \( h_2(q) \) are, respectively, the low-pass wavelet filter coefficient of row and column separable 2-D DWT. Similarly, \( g_1(p) \) and \( g_2(q) \) are, respectively, the high-pass wavelet filter coefficient of row and column DWT.

The computation of equation (1.25), (1.26), (1.27) and (1.28) can be decomposed into three distinct stages of computation. In stage-1 computation of 2-D DWT is performed using the pair of equations

\[
u_j^{l-1}(m,n) = \sum_{p=0}^{L-1} h_1(p)a^{l-1}(m, 2n - p) \quad (1.29)
\]
where $u_{l}^{(i)}(m, n)$ and $u_{h}^{(i)}(m, n)$ are low-pass and high-pass component of intermediate matrices $U_{l}^{(i)}$ and $U_{h}^{(i)}$, respectively. The two intermediate matrices $U_{l}^{(i)}$ and $U_{h}^{(i)}$ generated by the stage-1 are transposed in stage-2. The transposed matrices are process in stage-3 to calculate four subband matrices. Stage-3 computation of 2-D DWT is performed using the following set of equations

\[ a^{i}(m, n) = \sum_{q=0}^{L-1} h_{2}(q)u_{l}^{(i-1)}(2m - q, n) \]  
\[ b^{i}(m, n) = \sum_{q=0}^{L-1} g_{2}(q)u_{l}^{(i-1)}(2m - q, n) \]  
\[ c^{i}(m, n) = \sum_{q=0}^{L-1} h_{2}(q)u_{h}^{(i-1)}(2m - q, n) \]  
\[ d^{i}(m, n) = \sum_{q=0}^{L-1} g_{2}(q)u_{h}^{(i-1)}(2m - q, n) \]

The 2-D DWT of any given level of decomposition using separable approach can, therefore, be computed in three distinct stages using separable filtering unit (SFU) as shown in Figure 1.5. In stage-1, 1-D DWT is performed on each of the $M$ rows of the matrix of size $(M \times N)$ using one FU to obtain two intermediate output matrices of size $(M \times N/2)$. In stage-2, the intermediate matrices are transposed and in stage-3, 1-D DWT once again is performed on the rows of the two intermediate output matrices using FU to obtain the four subband output matrices of size $(M/2 \times N/2)$.

![Figure 1.5: General block diagram of separable filtering unit (SFU).](image)
According to equation (1.29) – (1.34), separable 2-D DWT computation involves $2L$ arithmetic operation. Compared with non-separable approach, separable 2-D DWT involves $L/2$ times less computation for same throughput rate implementation. However, separable approach demands a transposition unit of size equal to the size of the input image.

1.4 Multilevel 2-D DWT

In multilevel 2-D DWT, low-low subband matrix ($A^1$) is further decomposed to generates four subband matrices i.e. $A^2$, $B^2$, $C^2$ and $D^2$ of second level DWT. This process is repeated to generate subband matrices of all higher DWT levels. Due to repeated down sampling of 2-D signal by factor of 2 along row direction and column direction, the amount of 2-D DWT computation decreases steadily by a factor of four for higher DWT levels. For example the amount of data to be processed in first level 2-D DWT is $MN$. Then the amount of data to be processed in second level, third level and fourth level 2-D DWT are, respectively, $MN/4$, $MN/16$ and $MN/32$. In general the amount of data to be processed in $j$-th level 2-D DWT is $MN/4^j$.

According to Mallat PA algorithm, separate NSFU or SFU is required for each DWT level to compute multilevel 2-D DWT using non-separable or separable approach. Since, the amount of computation decreases steadily by a factor of four after every decomposition level, the HUE of each NSFU/SFU of PA-based structure is also decreases in the order of one fourth. The HUE of a $J$ level PA-based 2-D DWT structure can be calculated using the formula

$$U = \frac{4(1-4^{-j})}{3J} \times 100\%$$

(1.35)

HUE of 2-level, 3-level and 4-level PA based 2-D DWT are calculated using the formula of (1.35), and found to be 62%, 44% and 33%, respectively. Due to low HUE, PA-based 2-D DWT structure is not suitable for efficient hardware implementation.

To overcome this difficulty, a RPA schedule has been proposed by Vishwanath et al. (1995) for 2-D DWT in the similar form of RPA schedule of 1-D DWT. According to Vishwanath et al. (1995), RPA schedule for 2-D DWT can be prepared by time multiplexing 2-D DWT computations of higher levels at the row-wise or column-wise down sampling
clock cycles of first level. Using the RPA schedule, \( J \) level 2-D DWT of an input 2-D signal can be computed using only one NSFU/SFU. Consequently, the HUE of RPA-based 2-D DWT structure is significantly higher than the PA-based 2-D DWT structure. HUE of RPA-based 2-D DWT structure can be calculated using the formula

\[
U = \frac{2(1-2^{-2J})}{3}
\]  

HUE of RPA-based 2-D DWT structure for \( J = 2, 3, 4 \) and 5, are calculated using the formula of (1.36), and found to be 62.5%, 65.6%, 66.4% and 66.6%, respectively. The maximum HUE achievable in a RPA-based 2-D structure is 66.66%, which is far less than 100%. Although, HUE of RPA-based 2-D DWT structure is significantly higher than that of PA-based 2-D DWT structure, but 100% HUE can not be achieved in RPA-based 2-D DWT structure. Also, the RPA-based structure requires complex control circuitry than the PA-based designs.

### 1.4.1 Folded Scheme

Wu and Chen (2001) have proposed folded scheme to overcome the difficulties of RPA-based 2-D DWT structure. According to folded scheme of Wu and Chen (2001) multilevel 2-D DWT computation is performed serially in level by level using single SFU and a frame buffer. The low-low subband of the current DWT level is stored in the frame buffer to calculate the higher DWT levels.

The general block diagram for computation of multilevel 2-D DWT using folded scheme is shown in Figure 1.6, which includes a SFU, a frame buffer and a multiplexer. The size of the frame buffer is \( MN/4 \) words, where \( (M \times N) \) is the image size. In the first level decomposition, the multiplexer selects data from the input matrix. The SFU decomposes the input matrix into four subbands matrices low-low (\( A^1 \)), low-high (\( B^1 \)), high-low (\( C^1 \)) and high-high (\( D^1 \)), and saves the low-low (\( A^1 \)) subband to the frame buffer. After finishing the first level decomposition, the multiplexer selects data from the frame buffer. The low-low (\( A^1 \)) subband is then sent into the SFU to perform the second level decomposition. The SFU decomposes the low-low (\( A^1 \)) subband matrix to four subband matrices \( A^2 \), \( B^2 \), \( C^2 \) and \( D^2 \), and saves the \( A^2 \) subband matrix to the frame buffer. This procedure repeats until the desired
DWT level is computed. The HUE of folded structure is always 100%. Due to design simplicity and 100% HUE, folded scheme is more popular than PA and RPA for hardware efficient realization of multilevel 2-D DWT.

Figure 1.6: Computation of multilevel level 2-D DWT using folded scheme.

1.5 VLSI System for DWT

The DWT is computationally intensive and most of its application demand real-time processing. One way of achieving high speed performance is to use fast computational algorithm in a general purpose computers. Another way is to exploit the parallelism inherent in the computation for concurrent processing by a set of parallel processor. But, it is not cost effective to use a general purpose computer for a specific application. Also, general purpose computer used for their implementation required more space, large power and more computation time. With the development of very large scale integration (VLSI) technology it facilitates to digital signal processing (DSP) system designer to design a high performance, low cost and low power system in a single chip. The characteristic of VLSI system are that they offer greater potential for large amount of concurrency and offer an enormous amount of computing power within a small area [Weste and Eshraghian (1993)]. The computation is very cheap as the hardware is not an obstacle for VLSI system. But, the non localized global communication is not only expensive but demands high power dissipation. Thus, a high degree of parallelism and a nearest neighbor communication are crucial for realization of high performance VLSI system [Kung (1982)]. Keeping this in view, high performance application specific VLSI systems are rapidly evolving in recent years. The special purpose VLSI systems maximize processing concurrency by parallel / pipeline processing and
provides cost effective alternative for real-time application. Therefore, 2-D DWT is currently implemented in a VLSI system to meet the temporal requirement of real-time application.

Keeping this fact in view, several design schemes have been suggested in the last two decades for efficient implementation of 2-D DWT in a VLSI system. Researchers have adopted different algorithm formulation, mapping scheme, and architectural design methods to reduce the computational time, arithmetic complexity or memory complexity of 2-D DWT structures. However, the area-delay performance of the existing structures changes marginally. This is mainly due to the memory complexity, which forms a major hardware component of folded 2-D DWT structure. A detail study of the existing design methods and a complexity analysis is made in Chapter 2 to find an appropriate design strategy to improve the area-delay performance of 2-D DWT structures.