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 multi resolution 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 and image compression [Sodagar *et al*. (1999); speech and video-compression, 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. The commonly used computation schemes for DWT 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.

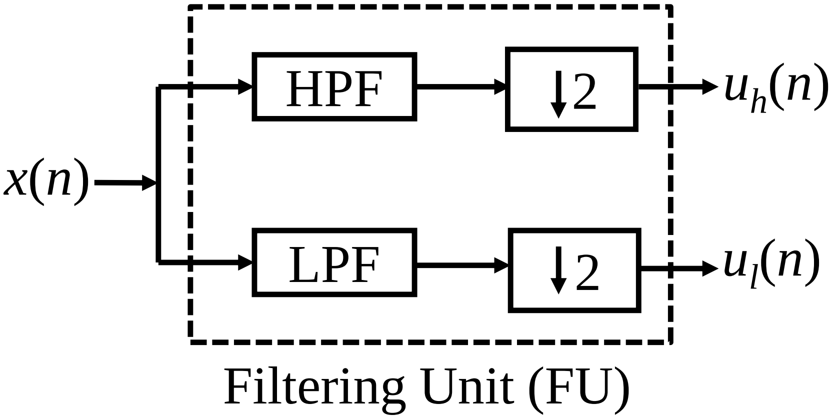

Figure 1.1: Computation of one level 1-D DWT.

1.1.1 CONVOLUTION SCHEME

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

$$u_l(n) = \sum_{i=0}^{k_1-1} h(i)x(2n-i) \tag{1.1}$$

$$u_h(n) = \sum_{i=0}^{k_2-1} g(i)x(2n-i) \tag{1.2}$$

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 ($\mathbf{P}(z)$) 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

$$\mathbf{P}(z) = \begin{bmatrix} H_e(z) & G_e(z) \\ H_o(z) & G_o(z) \end{bmatrix} \tag{1.2}$$

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

$$\mathbf{P}(z) = \left\{ \prod_{i=1}^{i=m} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \right\} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \tag{1.3}$$

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

$$\begin{bmatrix} u_l(n) & u_h(n) \end{bmatrix} = \begin{bmatrix} x(2n) & x(2n-1) \end{bmatrix} \mathbf{P}(z) \tag{1.4}$$

and

$$\mathbf{P}(z) = \begin{bmatrix} 1 & \alpha(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & \gamma(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \delta(1+z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & K^{-1} \end{bmatrix} \tag{1.5}$$

where α, β, γ and δ 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

$$s_1(n) = x(2n-1) + \alpha(x(2n) + x(2n-2)) \quad (1.6)$$

$$s_2(n) = x(2n-2) + \beta(s_1(n) + s_1(n-1)) \quad (1.7)$$

$$v_1(n) = s_1(n-1) + \gamma(s_2(n) + s_2(n-1)) \quad (1.8)$$

$$v_2(n) = s_2(n-1) + \delta(v_1(n) + v_1(n-1)) \quad (1.9)$$

$$u_h(n) = K^{-1}v_1(n) \quad (1.10)$$

$$u_l(n) = Kv_2(n) \quad (1.11)$$

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

1.1.3 FLIPPING SCHEME

Lifting scheme involves a longer critical path delay (CPD) than the design based on convolution scheme. To address this issue, Huang *et al*. (2002) have proposed a modified lifting scheme known as *flipping scheme*. The flipping scheme is inherently pipelined and it has shorter critical path than the lifting-scheme. The lifting computations of (1.7)-(1.12) are expressed in a modified form as:

$$r_1(n) = \alpha^{-1}x(2n-1) + x(2n) + x(2n-2) \quad (1.13)$$

$$r_2(n) = (\alpha\beta)^{-1}x(2n-2) + r_1(n) + r_1(n-1) \quad (1.14)$$

$$r_3(n) = (\beta\gamma)^{-1}r_1(n-1) + r_2(n) + r_2(n-1) \quad (1.15)$$

$$r_4(n) = (\gamma\delta)^{-1}r_2(n-1) + r_3(n) - r_3(n-1) \quad (1.16)$$

$$v_h(n) = \alpha\beta\gamma/K\ r_3(n) \quad (1.17)$$

$$v_l(n) = \alpha\beta\gamma\delta\ K\ r_4(n) \quad (1.18)$$

1.1.4 MULTI-LEVEL 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 Multi-level 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 Multi-level DWT. These algorithms are briefly discussed here. Mallat (1989a) has proposed a pyramid algorithm (PA) for parallel computation of Multi-level DWT. PA for 1-D DWT is given by

$$u_l^j(n) = \sum_{i=0}^{k-1} h(i) u_l^{j-1}(2n-i) \tag{1.19}$$

$$u_h^j(n) = \sum_{i=0}^{k-1} g(i) u_h^{j-1}(2n-i) \tag{1.20}$$

where $u_l^j(n)$ is the *n*-th low-pass sub band component of the *j*-th DWT level and $u_h^j(n)$ is the *n*-th high-pass sub band component of the *j*-th DWT level, for $n \leq 0 \leq \left(\frac{N}{2^j}\right) - 1$, u_l^0 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.

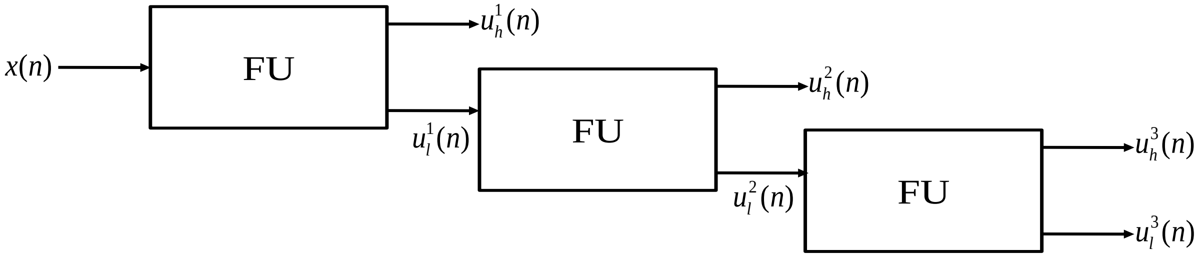

Figure 1.2: Computation of three level 1-D DWT using pyramid algorithm (PA).

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

$$U = \frac{2N(1-2^{-J})}{JN} \times 100\% \tag{1.21}$$

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 Multi-level 1-D DWT structure. To overcome this problem, Vishwanath (1994) has proposed a recursive pyramid algorithm (RPA). In RPA, the multi-level DWT computations are rescheduled such that computation of higher DWT levels are time multiplexed with the computation of first level DWT. Using RPA one or two filtering units are required to perform the computation of multi-level DWT. The HUE of RPA-based J level 1-D DWT structure can be calculated using the formula

$$U = \frac{N + N(1-2^{1-J})}{2N} \tag{1.22}$$

HUE of RPA based 1-D DWT structure is 96% for J=5 which is much higher than the PA-based multi-level 1-D DWT structure.

1.2 COMPUTATION SCHEME OF 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 $\{h_{ll}(p,q)\},\{h_{lh}(p,q)\},\{h_{hl}(p,q)\}$ and $\{h_{hh}(p,q)\}$ of size (L × 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

$$h_{ll}(p,q) = h_1(p)h_2(q) \tag{1.23}$$

$$h_{lh}(p,q) = h_1(p)g_2(q) \tag{1.24}$$

$$h_{hl}(p,q) = g_1(p)h_2(q) \tag{1.25}$$

$$h_{hh}(p,q) = g_1(p)g_2(q) \tag{1.26}$$

where $0 \le (p,q) \le L-1$, $h_1(p)$ 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. PA for separable 2-D DWT with separable wavelet basis function is given by

$$a^j(m,n) = \sum_{p=0}^{L-1}\sum_{q=0}^{L-1} h_1(p)h_2(q)a^{j-1}(2m-p,2n-q) \tag{1.27}$$

$$b^j(m,n) = \sum_{p=0}^{L-1}\sum_{q=0}^{L-1} h_1(p)g_2(q)a^{j-1}(2m-p,2n-q) \tag{1.28}$$

$$c^j(m,n) = \sum_{p=0}^{L-1}\sum_{q=0}^{L-1} g_1(p)h_2(q)a^{j-1}(2m-p,2n-q) \tag{1.29}$$

$$d^{j}(m,n)=\sum_{p=0}^{L-1}\sum_{q=0}^{L-1}g_{1}(p)g_{2}(q)a^{j-1}(2m-p,2n-q) \tag{1.30}$$

where $0\leq\left(p,q\right)\leq L-1$, $h_1(p)$ 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 (1.27), (1.28), (1.29) and (1.30) are decomposed into three distinct stages. In stage-1 computation of 2-D DWT is performed using the following pair of equations

$$u_{l}^{j-1}(m,n)=\sum_{p=0}^{L-1}h_{1}(p)a^{j-1}(m,2n-p) \tag{1.31}$$

$$u_{h}^{j-1}(m,n)=\sum_{p=0}^{L-1}g_{1}(p)a^{j-1}(m,2n-p) \tag{1.32}$$

where $u_{l}^{j-1}(m,n)$ and $u_{h}^{j-1}(m,n)$ are low-pass and high-pass component of intermediate matrices $\mathbf{U}_{l}^{j-1}$ and $\mathbf{U}_{h}^{j-1}$, respectively. The two intermediate matrices $\mathbf{U}_{l}^{j-1}$ and $\mathbf{U}_{h}^{j-1}$ 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.

$$a^{j}(m,n)=\sum_{q=0}^{L-1}h_{2}(q)u_{l}^{j-1}(2m-q,n) \tag{1.33}$$

$$b^{j}(m,n)=\sum_{q=0}^{L-1}g_{2}(q)u_{l}^{j-1}(2m-q,n) \tag{1.34}$$

$$c^{j}(m,n)=\sum_{q=0}^{L-1}h_{2}(q)u_{h}^{j-1}(2m-q,n) \tag{1.35}$$

$$d^{j}(m,n)=\sum_{q=0}^{L-1}g_{2}(q)u_{h}^{j-1}(2m-q,n) \tag{1.36}$$

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 1.3. 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 ($M/2 \times N/2$).

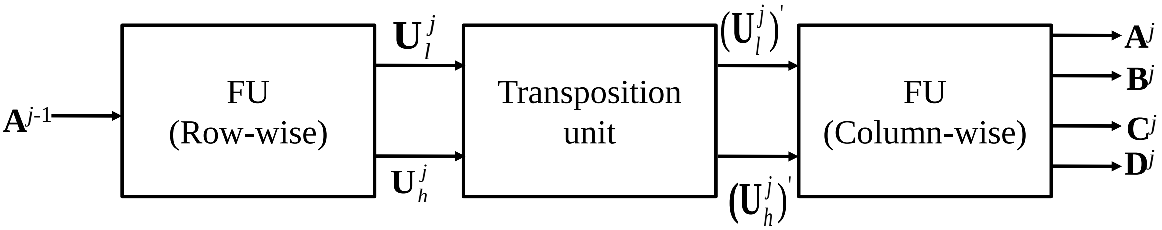

Figure 1.3.: General block diagram of separable filtering unit (SFU).

According to equation (1.31)-(1.36), 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.

1.3 MULTI-LEVEL 2-D DWT

In Multi-level 2-D DWT, low-low sub band matrix ($\mathbf{A}^1$) is further decomposed to generates four sub band matrices i.e. $\mathbf{A}^2$, $\mathbf{B}^2$, $\mathbf{C}^2$ and $\mathbf{D}^2$ of second level DWT. This process is repeated to generate sub band matrices of all higher DWT levels. The block diagram of three level decomposition of 2-D DWT is shown in Figure 1.4. Three SFU are required to perform the computation of three-level 2-D DWT. 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 $MN/4^j$. The HUE of a *J* level PA-based 2-D DWT structure can be calculated using the formula

$$U = \frac{4(1-4^{-J})}{3J} \times 100\% \tag{1.37}$$

HUE of 2-level, 3-level and 4-level PA based 2-D DWT are calculated using the formula of (1.37), 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

$$U = \frac{2(1-2^{-2J)})}{3} \tag{1.38}$$

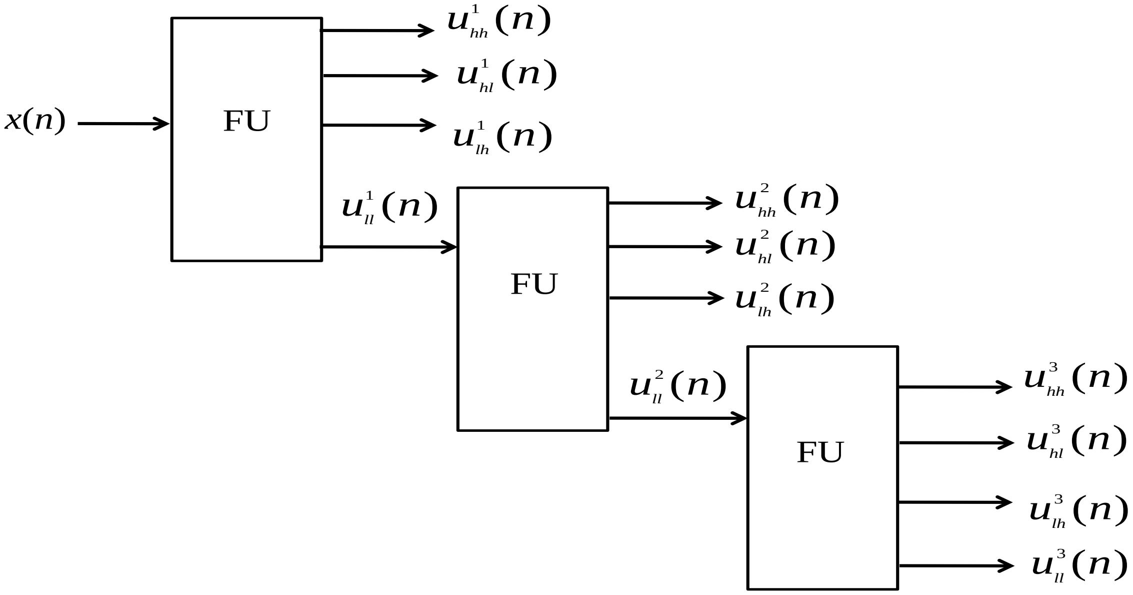

Figure 1.4: Computation of three level 2-D DWT using pyramid algorithm (PA)

HUE of RPA-based 2-D DWT structure for J = 2, 3, 4 and 5 are calculated using the formula of (1.38), 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 cannot be achieved in RPA-based 2-D DWT structure.

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) Multi-level 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 Multi-level 2-D DWT using folded scheme is shown in Figure 1.5, 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 ($\mathbf{A}^1$), low-high ($\mathbf{B}^1$), high-low ($\mathbf{C}^1$) and high-high ($\mathbf{D}^1$), and saves the low-low ($\mathbf{A}^1$) subband to the frame buffer. After finishing the first level decomposition, the multiplexer selects data from the frame buffer. The low-low ($\mathbf{A}^1$) subband is then sent into the SFU to perform the second level decomposition. The SFU decomposes the low-low ($\mathbf{A}^1$) subband matrix to four subband matrices $\mathbf{A}^2$, $\mathbf{B}^2$, $\mathbf{C}^2$ and $\mathbf{D}^2$, and saves the $\mathbf{A}^2$ subband matrix to the frame buffer. This procedure repeats until the desired DWT level is computed. The HUE of folded structure is always 100%.

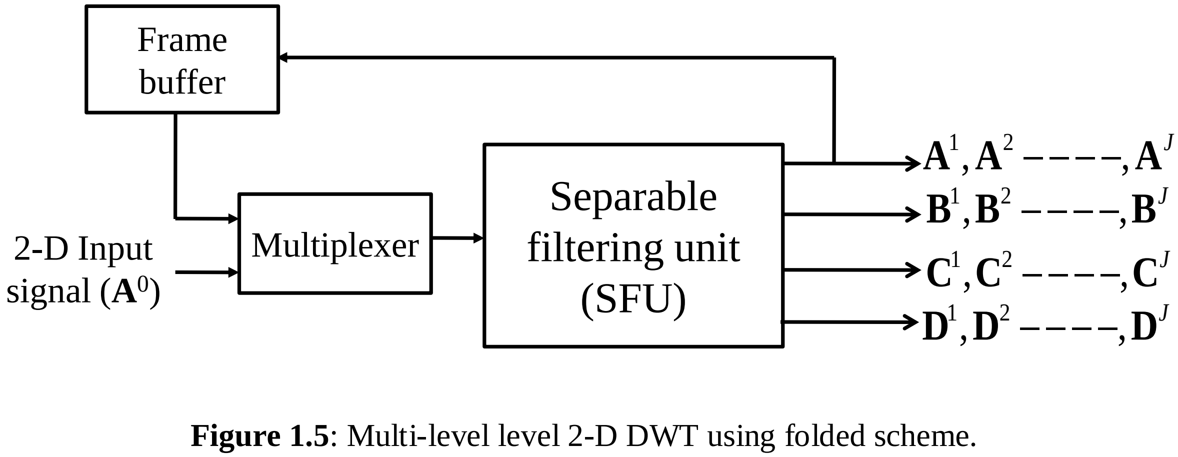
Frame buffer
2-D Input signal ($\mathbf{A}^0$)
Multiplexer
Separable filtering unit (SFU)
$\mathbf{A}^1, \mathbf{A}^2$ ----, $\mathbf{A}^J$
$\mathbf{B}^1, \mathbf{B}^2$ ----, $\mathbf{B}^J$
$\mathbf{C}^1, \mathbf{C}^2$ ----, $\mathbf{C}^J$
$\mathbf{D}^1, \mathbf{D}^2$ ----, $\mathbf{D}^J$

Figure 1.5: Multi-level level 2-D DWT using folded scheme.

Mohanty and Meher (2011) have proposed a parallel scheme for computation of 2-D DWT which is identical to the PA scheme. But, in this scheme the intermediate data-blocks are folded and scheduled such and the 2-D DWT computation of each higher DWT level are performed concurrently in different filtering unit with 100% utilization. However, the parallel computation involves overhead complexity for resizing intermediate data-blocks.

1.4 VLSI SYSTEM

In recent years, there is tremendous growth of portable and wireless devices in various applications. The portable and mobile devices are enabled with audio and video applications, wireless communication and internet applications. Mobile devices need to support increasingly diverse and more sophisticated function in the coming years. It requires high speed, low power and low-complexity digital hardware to deliver superior performance under resource constrained environment. On the other hand portable and wireless devices are resource and power constrained, and uses complex signal processing algorithms such as DWT. High-throughput rate, low-area and low-power are the key requirements of systems implementing digital signal processing algorithms suitable for portable and wireless devices which are battery operated. Therefore, realization of DWT in resource and power constrained environment and delivering computational requirement of ever increasing portable and wireless applications is remain a challenging area of research.

Along with the development of very large scale integration (VLSI) technology, high performance, low-area and low-power dedicated hardware systems are fabricated 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.

With the advancement of VLSI technology, high density and low-cost memory chips are rapidly evolving in recent years. Field programmable gate array (FPGA) is a programmable logic device. FPGA offers high capacity programmable devices for realization of complex DSP algorithms. FPGA uses a fixed architecture and offers specific types of memory and logic resources for realization of digital systems. On the other hand, modern synthesis tools offer a wide range of logic and memory components for realization of application specific integrated circuit (ASIC) systems. Therefore, ASIC system offers higher-performance and consumes less-power compared to FPGA systems. However, ASIC system does not allow reusing its resources through programming unlike the FPGA. Therefore, FPGA offers rapid prototyping the complex DSP algorithms with lesser performance than the ASIC. In general ASIC implementation is preferred for computation intensive algorithms and high volume applications. Several design schemes have been suggested in the last two decades for efficient implementation of 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 DWT.

1.5 THESIS ORGANIZATION

In Chapter 2, literature survey on DWT is presented to study the research progress on architecture design for efficient implementation 2-D DWT. Chapter 3 presents a new data scheme and block formulation to design parallel structures without using data-selectors which most common in lifting 2-D DWT. A concurrent design also presented in this Chapter for computation multi-level lifting 2-D DWT. In Chapter 4, the proposed radix-8 multiplier design is presented. Chapter 5 presents the fixed-width constant radix-8 Booth multiplier design to develop an efficient hardware design for lifting 2-D DWT. Chapter 6 presents look-up-table (LUT) based signed multiplier design for the use of lifting DWT structures. Thesis summary and future scope is presented in Chapter 7.