CHAPTER 6

LOW LATENCY, HIGH SPEED ARCHITECTURE FOR NON SEPARABLE BLOCK BASED LIFTING DWT

The inherent advantage of the in-place computation of the lifting based DWT over the conventional convolution method makes it suitable for efficient hardware implementation with lower computational complexity. The Non Separable block based method is used to derive the VLSI architecture for the 2-D Lifting DWT is discussed in section 1.5.2 of chapter 1. The 1-D DWT is obtained, by first applying the lifting steps for the given image coefficients row-wise, to produce the L and H sub bands and then in column-wise to give four bands LL, LH, HL, HH. The Separable architectures, proposed by Wu & Chen (2001); Barua et al (2004); Tian et al (2011) and Mohanty et al (2012) requires 8 lifting steps for the 9/7 lossy compression standard of JPEG, and this introduces delay or latency in DWT decomposition. A block based parallel scanning method as discussed in Lai (2009), for the given floating point image coefficients is done, and the order of scanning of the image coefficients is done first column wise and then row-wise, to reduce the memory requirements. The Non separable 2-D convolutional DWT architectures as proposed by Cheng & Parhi (2008), are obtained by first transforming the image coefficients in the column direction and then in a row manner using parallel FIR structures. A non-separable VLSI architecture, by combining the row and column processor, as proposed by Masahiro & Hitoshi (2013), can be used for high speed VLSI implementation of the lifting 2-D DWT. To represent the given real image coefficients in a binary format for
multipliers and adders in the discussed architecture, IEEE 754 single precision floating point format as discussed in IEEE 754-2008 (2008), is used.

The architecture is suitable for handling a parallel block of 4×4 input image coefficients at each level, for achieving higher speed with reduced lifting steps achieving dynamic accuracy. The architecture is coded in Hardware Description Language and implemented in Altera Cyclone II FPGA. The critical path delay of the proposed non-separable architecture obtained is 3.189 ns, with a speed of 313.58 MHz and total thermal power dissipation of 128.59 mW, suitable for low power embedded multimedia applications.

6.1 MATHEMATICS INVOLVED IN THE DERIVATION OF NON SEPARABLE VLSI ARCHITECTURE FOR 9/7 LOSSY FILTERS

This section introduces the mathematical background involved in the derivation of separable and Non separable Lifting steps for efficient hardware implementation. The Non separable architecture is obtained by a combination of row/ column filters. The lossy standard 9/7 filters require six lifting steps, and latency is reduced.

6.1.1 Separable 2-D DWT and its Latency in Lifting Steps for the 5/3 Lossless and 9/7 Lossy Standard

The lifting scheme has been developed as a flexible tool, suitable for constructing the second generation wavelets. It is composed of three basic operation stages: split, predict, and update. These steps are executed sequentially, after obtaining the results of the previous stage. This introduces latency in the conventional Separable method. A Non-separable lossy 9/7 standard for high speed applications of the lifting DWT is proposed, with a
reduced number of lifting steps. The separable method of implementation of the 2-D DWT requires temporal memory, for storing the intermediate levels of coefficients.

i. 5/3 lossless standard

The filter coefficients of the finite causal low pass and high pass for the 5/3 lossless standard is given by,

\[
\begin{bmatrix}
    H_1 \\
    H_2
\end{bmatrix} = \begin{bmatrix}
    H_1(Z_1) \\
    H_2(Z_1)
\end{bmatrix} = \begin{bmatrix}
    -1/2 & 0 \\
    0 & 1/4
\end{bmatrix} \begin{bmatrix}
    1 & 1/Z_1 \\
    1 + Z_1 & 1
\end{bmatrix}
\]

(6.1)

The above equation represents the low pass and high pass filter coefficients for the lossless 1-D (row) lifting separable DWT. The filter coefficients corresponding to the vertical (column) processing are given by,

\[
\begin{bmatrix}
    H_1^* \\
    H_2^*
\end{bmatrix} = \begin{bmatrix}
    H_1(Z_2) \\
    H_2(Z_2)
\end{bmatrix} = \begin{bmatrix}
    -1/2 & 0 \\
    0 & 1/4
\end{bmatrix} \begin{bmatrix}
    (1 + Z_2^*) \\
    (1 + Z_2^*)^{-1}
\end{bmatrix}
\]

(6.2)

The implementation of the 5/3 lossless standard requires four lifting steps in the conventional separable lifting 2-D DWT.

ii. 9/7 lossy standard:

The lossy 9/7 standard requires four lifting steps in row processing and four lifting steps in column processing, for the implementation of the separable 2-D DWT. The low pass and high pass filter coefficients for the 2-D 9/7 row processing are given by,

\[
\begin{bmatrix}
    H_1 & H_2 \\
    H_2 & H_4
\end{bmatrix} = \begin{bmatrix}
    H_1(Z_1) & H_3(Z_1) \\
    H_2(Z_1) & H_4(Z_1)
\end{bmatrix}
\]

(6.3)

The column filters are given by,
\[
\begin{bmatrix}
H_1 & H_3 \\
H_2 & H_4
\end{bmatrix}
= \begin{bmatrix}
\alpha & 0 \\
0 & \beta
\end{bmatrix}
\begin{bmatrix}
1 + Z^{-1} \\
1 + Z^{-1}
\end{bmatrix}
\]

(6.4)

where
\[
\begin{bmatrix}
H_1(Z) \\
H_2(Z)
\end{bmatrix}
= \begin{bmatrix}
\alpha & 0 \\
0 & \beta
\end{bmatrix}
\begin{bmatrix}
1 + Z^{-1} \\
1 + Z^{-1}
\end{bmatrix}
\]

(6.5)

\[
\begin{bmatrix}
H_3(Z) \\
H_4(Z)
\end{bmatrix}
= \begin{bmatrix}
\gamma & 0 \\
0 & \delta
\end{bmatrix}
\begin{bmatrix}
1 + Z^{-1} \\
1 + Z^{-1}
\end{bmatrix}
\]

(6.6)

The computations of the filter coefficients at each level are dependent on the results of the previous stage. The latency exists in separable decomposition, and is equal to eight for the row-column decomposition of the separable lifting DWT. Each of the filters \(H_1(Z), H_2(Z), H_3(Z), H_4(z)\) contains two adders, one multiplier and one delay register, with a critical path delay of \(Tm+2Ta\), where \(Tm\) and \(Ta\) are the delay in the execution times of the multipliers and adders respectively.

### 6.1.2 Mathematics in the Proposed Non-Separable 2-D Lifting DWT and its latency

The Non–separable 2-D DWT can be obtained, by combining the row/column decomposition filters, as discussed in Masahiro & Hitoshi (2013), so that the latency of eight lifting steps is reduced to six in the proposed architecture. The low pass and high pass filter coefficients of 2-D 9/7 DWT for row/column processing are given by,

\[
\begin{bmatrix}
H_1H_1^* & H_3H_3^* \\
H_2H_2^* & H_4H_4^*
\end{bmatrix}
= \begin{bmatrix}
H_1(Z_1)H_1(Z_2) & H_3(Z_1)H_3(Z_2) \\
H_2(Z_1)H_2(Z_2) & H_4(Z_1)H_4(Z_2)
\end{bmatrix}
\]

(6.7)

The filter coefficients \(H_1, H_2, H_3, H_4\) are the row processing filters and \(H_1^*, H_2^*, H_3^*, H_4^*\) are the column filters. The in-place matrix representation of the Non separable Lifting is obtained, by multiplying the down sampled version of the input image coefficients with the corresponding
filter coefficients, to obtain the sub band decomposed DWT values. The output decomposed values LL, LH, HL, and HH are related to the down sampled odd and even coefficients values $X_{11}, X_{12}, X_{21}, X_{22}$ by:

$$
\begin{bmatrix}
LL \\
LH \\
HL \\
HH
\end{bmatrix} = (J_K \cdot \begin{bmatrix} N_{H_4} & H_3 & H_4 & H_2 \\
N_{H_2} & H_1 & H_2 & H_1 \\
N & H & 2 & H & 1
\end{bmatrix}) \begin{bmatrix}
X_{11} \\
X_{12} \\
X_{21} \\
X_{22}
\end{bmatrix}
$$

(6.8)

where $\alpha, \beta, \gamma$ and $\delta$ are the multiplier coefficients, and $J_K$ is the constant and the sum of the diagonal elements represents normalizing the magnitude, for floating point representation. The inputs X odd and X even is the odd and even pixels of the down sampled input image coefficients. The architecture uses four blocks of a 4x4 input image in parallel, at the given clock period. The output sub bands LL, LH, HL, and HH are obtained, based on the input-output relationship of Equation (6.8), and the general block diagram of the Non Separable method is discussed in Section 1.5.2. The total number of lifting steps is reduced from eight (separable) to six in the Non Separable implementation of the 2-D lifting DWT. The latency for the execution steps is reduced, and their percentage of reduction when compared to that of the separable method is shown in Table 6.1.
Table 6.1 Latency comparison of the Separable and Non separable 2-D DWT

<table>
<thead>
<tr>
<th>DWT METHOD</th>
<th>LIFTING STEPS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5/3 Lossless Standard</td>
</tr>
<tr>
<td>Separable 2-D DWT</td>
<td>4(100%)</td>
</tr>
<tr>
<td>Non Separable 2-D DWT</td>
<td>3(75%)</td>
</tr>
</tbody>
</table>

6.2 HARDWARE IMPLEMENTATION OF THE NON SEPARABLE LIFTING 2-D DWT

The Non separable Lifting DWT architecture proposed uses the combination of row/column filters. The floating point arithmetic implementation of the functional units in the architecture is done, for handling the real time implementation of image/video signals with improved accuracy. The latency is reduced to six for 2-D lifting DWT.

6.2.1 Proposed Architecture for the Non Separable Lifting DWT

The conventional Separable Lifting DWT depends on the results of the previous output at each level, as discussed in Sections 3.1 & 5.1. This introduces a latency of four clock cycles for the 5/3 lossless and eight for the 9/7 lossy separable systolic 2-D DWT. The column processing is initiated only after computing the row or horizontal coefficients. An additional memory is required to store the intermediate results.

The Non–separable 2-D DWT can be obtained, by combining the row/column decomposition filters. The filter coefficients $H_1$, $H_2$, $H_3$, $H_4$ are the row processing filters are combined with $H_1^*$, $H_2^*$, $H_3^*$, $H_4^*$ the column filters processing filters in six clock cycles to give the first level sub band.
coefficients. The total number of lifting steps is reduced from eight (separable) to six in the Non Separable implementation of the 2-D lifting DWT with the controlled increase of hardware cost.

The row/ column filter in the proposed architecture in Figure 6.1. is implemented, by using 32 bit floating point multipliers, adders and registers. The down sampler divides the given array of input image coefficients in to odd and even coefficients. The inputs $X_{11}$, $X_{12}$, $X_{21}$ and $X_{22}$ of the proposed architecture are the corresponding outputs of the down sampler. The registers used in the architecture are 32 bit registers to hold the result of the execution at each level. These registers are pipelined to achieve parallel processing of the outputs at each level.

![Proposed Non-Separable 2-D Lifting DWT hardware architecture](image)

**Figure 6.1** Proposed Non-Separable 2-D Lifting DWT hardware architecture

### 6.2.2 Method of Scanning Image Coefficients

The input image coefficients are scanned along the first column of the block of coefficients, and the filtering is done by combining the row—
column filters to generate a detailed HH sub band output. The image coefficients are scanned along the second column, to generate the LH and HL sub band coefficient output. The image coefficients are scanned along the second column, to generate the required approximated low pass (LL) output. The same processing steps are repeated along the 4th, 5th and 6th column and multiplied with the scaling factors to generate the final sub band coefficient output. The general order parallel scanning of the image coefficient is shown in Figure 6.2 below. In this each circle denotes a parallel block of 4×4 image coefficients at each level.

![Figure 6.2](image)

**Figure 6.2 Scanning order of the input image Coefficients in the proposed method**

The floating point multiplier and adder in the architectural design, as shown in Figure 6.1 are being efficiently implemented in the hardware, to make Non separable architecture suitable for high speed and low power image processing applications, with improved dynamic accuracy. In the multimedia
System On Chips, the design of specialized floating point arithmetic units (FPUs) is critical, with respect to the operating speed and area. An efficient implementation of the IEEE 754 single precision floating point multipliers for FPGA devices is discussed by Al-Ashrafy et al (2011). The pipelining of multipliers for a high speed MAC unit is also discussed.

6.2.3 Implementation Steps of Floating Point Multipliers

The Multiplication of the floating point numbers $X_1$ with sign $S_1$, exponent $E_1$ and significand $P_1$ and the second number $X_2$ with sign $S_2$, exponent $E_2$ and significand $P_2$, is shown in the multiplier architecture in Figure 6.3. The preprocessing of the given floating point number is used to detect whether the given number is not a number (NaN), Infinity (INF), and zero (IsZ) to accommodate the given image floating point value in the given word length, as discussed by Kahan (1996). The algorithm of the IEEE 754 single precision floating point multipliers implementation steps in the Non Separable architecture in hardware, are listed below:

i. Exponent addition

Calculate the tentative exponent of the product by adding the biased exponents of the two numbers, subtracting the bias. The bias is 127 for the single precision IEEE data format, and is given by: $E_1 + E_2 - 127$.

ii. Sign calculation

If the sign of the two floating point numbers is the same, set the sign of product to positive, else set it to negative. The multiplication of two floating numbers is negative, if and only if, any one of the multiplied numbers is negative. This is obtained by comparing the sign of $S_1$ and $S_2$, using the XOR gate. $\text{Sign out} = S_1 \oplus S_2$. 
iii. Multiplication of Significant

Multiply the two significands $P_1$ and $P_2$ each of 24 bits. For $P$ bit significand, the product is $2P$ bits wide, i.e., 48 bits. The decimal point is placed between the 47th and 48th bit in the significand multiplier result.

iv. Normalization

The result of the significand multiplication must be normalized, to have a leading 1 to the left of the decimal point. The intermediate product has the leading one bit at the 47th or 48th bit.

i. If the leading one is at bit 47 to the left of the decimal point, then the intermediate product is a normalized number, and no shift is required.

ii. If the leading one is at bit 48, then the intermediate product is shifted to the right, and the exponent is incremented by 1.
v. Rounding operations:

The rounding of the floating point result is obtained by the concept of round to the nearest –even approach. Two guard bits and a sticky bit are used to ensure the computation results within acceptable accuracy and with minimum overhead. The MSB of the product of significant multiplication is taken for rounding. If the last two bits of the significand product are zero, no rounding is required. If the second MSB bit of the significand product is one, then add one to the 24th bit of significand product and the remaining three bits are appended with zero. The Flagger or exceptions handling, is used to detect whether there is an overflow, underflow or inexact result after rounding, based on the normalization effect on the exponent result, as shown in Table 6.2. The flag bit will be set corresponding to the result of the Exponent.

Table 6.2 Normalization Effect on the Exponent result and Overflow/Underflow detection

<table>
<thead>
<tr>
<th>Exponent result</th>
<th>Condition</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{\text{result}} \leq 0$</td>
<td>Underflow</td>
<td>Normalization cannot be done</td>
</tr>
<tr>
<td>$E_{\text{result}} = 0$</td>
<td>Zero</td>
<td>Can be normalized by adding 1 to it.</td>
</tr>
<tr>
<td>$1 &lt; E_{\text{result}} \leq 254$</td>
<td>Normalized number</td>
<td>Results in overflow during normalization</td>
</tr>
<tr>
<td>$255 \leq E_{\text{result}}$</td>
<td>Overflow</td>
<td>Cannot be represented as normalized number</td>
</tr>
</tbody>
</table>
6.2.4 Implementation steps of Floating Point Adders

The addition of two floating point numbers $X_1$ with sign $S_1$, exponent $E_1$ and significand $P_1$ and $X_2$ with sign $S_2$, exponent $E_2$ and significand $P_2$ is shown in the Figure 6.4. The pre-processing step consists of determining whether the given floating number is Not a Number, de-normal or infinity. Leading one anticipation is a well-known method, for implementing high speed floating point adders, and the architecture for the same was given by Ashwini (2012). The hardware architecture for the implemented floating point adder is shown in Figure 6.4.

![Figure 6.4 Hardware Architecture for floating point adder](image)
The algorithm for the implementation of the floating point adder is given in the following steps:

i. Compare the exponents of two numbers $E_1$ and $E_2$, and calculate the absolute value of difference between the two exponents. The larger exponent is taken as the tentative exponent of the result. $E_1 > E_2$, $E_1 \leq E_2$.

ii. Calculate the absolute difference between the two exponent values, which shift the significand of the number with the smaller exponent right through a number of bit positions, that is equal to the exponent difference. $D = |E_1 - E_2|$

iii. In parallel with the right shifter, two of the shifted out bits of the aligned significand are retained as guard (G) and Round (R) bits. So for P-bit significands, the effective width of the aligned significand must be $P + 2$ bits. Append a third bit, namely, the sticky bit (S), at the right end of the aligned significand. The sticky bit is the logical OR of all shifted out bits. These bits are used for rounding in the later stages.

iv. Add/subtract the two signed-magnitude significands based on the sign information. Let the result of this be the SUM of the significands.

V. Check the SUM for carry out (Cout) from the MSB position during addition. Shift SUM right by one bit position if a carry out is detected, and increment the tentative exponent by one. During subtraction, check SUM for leading zeros. Shift SUM to the left until the MSB of the shifted result is a one bit. Subtract the leading zero count from the tentative exponent. Evaluate exception conditions, if any.
vi. The normalization of significands requires a left shift by the variable predicted from Leading one Predictor; no shift if the output is already normalized or right shift by one bit if the carry out when the addition inputs have the same sign.

vii. The exponent is adjusted based on the result of normalization.

viii. Round the result if the logical condition of the P\(^{th}\) and (P + 1)\(^{st}\) bits from the left end of the normalized significand is true, and a 1 is added at the P\(^{th}\) bit from the left side of the normalized significand. If P MSB of the normalized significand is 1, rounding can generate a carry-out. In that case, normalization (step vi) has to be done again updating all the bit positions.

ix. The exceptions handling is used to detect whether there is an overflow, underflow or inexact result after rounding, based on the normalization effect on the exponent and significand sum result.

6.3 IMPLEMENTATION RESULTS OF THE PROPOSED NON SEPARABLE ARCHITECTURE

The developed 2-D DWT non separable lifting architecture comprising of the down sampler, 32 bit floating point registers, adders and multipliers, is coded in Verilog HDL and is implemented in EP2C35F672C6 cyclone II Altera FPGA. The computational complexity of the architecture is increased due to the floating point arithmetic, but this disadvantage can be overcome by achieving reduced critical path latency and low power, suitable for multimedia applications.
6.3.1 Simulation Results of the Non Separable Architecture

The simulation of the proposed architecture is done in Modelsim Altera. The Verilog code as dealt with Palnitkar (2003) and Williams (2008) for the architecture is coded for the functional circuits, and test bench is created for the given array of image coefficients. The test bench is created for 256×256 image coefficients, which are loaded into the architecture, where at each level of the inputs a block of 4×4 image coefficients is given. The image is read from the file in the read mode, and the output coefficients are initialized in the write mode in synchronization with the clock period. From the simulation analysis it is clear, that as soon as the image coefficient file is loaded into the architecture the sub band output, a low pass coefficient is obtained at a latency of six clock cycles.

6.3.2 Implementation Results of the Non Separable Architecture

The performance analysis of the Non separable Lifting 2-D DWT architecture is coded in HDL, and is generic for any square input block of image coefficients. The architecture is implemented using EP2C35F672C6 cyclone II Altera FPGA. The performance of the proposed architecture is compared, in terms of the number of multipliers, number of adders, storage size, computing time, control complexity and hardware utilization. The computing time has been normalized to the internal clocking rate. Every 4×4 block of coefficients at each level generates the sub band output blocks of LL, LH, HL and HH coefficients, with a latency of six clock cycles. The RTL schematic of the Non separable DWT architecture is shown in Figure 6.5; it consists of an input image coefficient X of 32 bit floating point data, along with the down sampler, registers and the required sub band outputs LL, LH, HL and HH coefficients.
Figure 6.5  RTL schematic of the Down sampler, Registers and sub-band outputs as implemented

The sample of the floating point adders, and multipliers, along with the required down sampled input image coefficients and registers are shown in Figure 6.6 below. The implementation results given in Tables 6.3 and 6.4 shows that the hardware implementation of the lifting schemes based on the Non Separable DWT is suitable for real time high speed low power multimedia applications.

Figure 6.6  Sample of the functional units of the proposed architecture as implemented
Table 6.3 Synthesis report of the Non separable 2-D DWT architecture

<table>
<thead>
<tr>
<th>FPGA</th>
<th>Altera Cyclone II EP2C35F672C6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total logic elements</td>
<td>246</td>
</tr>
<tr>
<td>Total Combinational functions</td>
<td>19</td>
</tr>
<tr>
<td>Total registers</td>
<td>236</td>
</tr>
<tr>
<td>Total pins</td>
<td>162</td>
</tr>
<tr>
<td>Total memory bits</td>
<td>160</td>
</tr>
<tr>
<td>Clock Set up period (ns)</td>
<td>3.189 ns</td>
</tr>
<tr>
<td>Maximum Operating frequency(MHz)</td>
<td>313.58 MHz</td>
</tr>
</tbody>
</table>

Table 6.4 Power analysis report of the Non separable 2-D DWT architecture

<table>
<thead>
<tr>
<th>Total Thermal Power dissipation(mW)</th>
<th>Core Static thermal power dissipation (mW)</th>
<th>I/O Thermal power dissipation(mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.59</td>
<td>79.98</td>
<td>48.60</td>
</tr>
</tbody>
</table>

From the implementation results the critical path latency of 3.189 ns is achieved with a 313.58 MHz of frequency of operation. The total power dissipation of the proposed architectures is 128.59 mW, suitable for low power applications.
6.3.3 Performance Calculation of the Compression Ratio in Altera FPGA

The proposed Non separable Lifting DWT for block based scanning of image coefficients is implemented on Altera DE2 115 Cyclone IV FPGA. The details of programming the Altera DE2-115 Cyclone IV FPGA are given by the Altera DE2 user manual (2013). The given image of size 256×256 is initialized into the Flash memory of the FPGA, as Memory Initialization File (MIF). The image coefficients are 32 bit wide with 8 bit indexed locations, for specifying the gray level variations of the given image. The FPGA is programmed using the Quartus II programmer, by downloading the configuration bit stream file in to it. The sample outputs obtained after performing the first level decomposition of the given images like boat, couple and Lena images in JPEG formats, are shown in the Figures 6.7, 6.8 and 6.9.

Figure 6.7 FPGA implementation of the Non Separable Lifting DWT for the Boat image
Figure 6.8  FPGA implementation of the Non Separable Lifting DWT for the Couple image

Figure 6.9  FPGA implementation of the Non Separable Lifting DWT for the Lena image

The given images are displayed by programming the VGA controller for displaying the images in the monitor. To estimate the
performance of the proposed architecture in hardware, compression ratio of the images as shown in the Figures: 6.7, 6.8& 6.9 is calculated.

i. Boat image

The dimension of the original image is 256×256 = 65536 bytes, and the first level decomposed boat image as shown in Figure 6.7 is 23694 bytes. The HDL coded output file is loaded into the MATLAB, for calculating the compression ratio:

\[
\text{Compression ratio in Altera FPGA hardware} = \frac{23694}{65536} \times 100 = 36.154\%
\]

ii. Couple image

The dimension of the original image is 256×256 = 65536 bytes, and the first level decomposed couple image as shown in Figure 6.8, is 24623 bytes.

\[
\text{Compression ratio in Altera FPGA hardware} = \frac{24623}{65536} \times 100 = 37.57\%
\]

Similarly for the Lena image shown in Figure 6.9, the Compression ratio in Altera FPGA hardware is 34.66%. The quality of compression obtained in hardware is higher, when compared to the corresponding compression ratio obtained in MATLAB, as discussed in Section 3.6.1. Hence, the hardware implementation of the Lifting DWT improves the quality of the displayed images for real-time image processing applications.