MEMORY EFFICIENT HIGH SPEED LIFTING BASED VLSI ARCHITECTURE FOR MULTI-LEVEL 2D-DWT

1M.SRAVANTHI, 2T.PRASAD

1P.G. Student in VLSI, Department of E.C.E, SIETK, Tirupati
2Assistant Professor, Department of E.C.E, SIETK, Tirupati
E-Mail to: mekala.sravanthi@gmail.com, Prasad.gmt@gmail.com

Abstract: This paper proposes an improved version of lifting-based Discrete Wavelet Transform (DWT). The lifting based DWT architecture has the advantage of lower computational complexities transforming signals with extension and regular data flow. The main feature of the lifting based DWT scheme is to break up the high pass and low pass filters into a sequence of upper and lower triangular matrices and convert the filter implementation into banded matrix multiplications. Such a scheme has several advantages, including “in-place” computation of the DWT, integer-to-integer wavelet transform (IWT), symmetric forward and inverse transform, etc.

In order to assess the feasibility and the efficiency of the proposed scheme, the architecture thus designed is simulated and implemented on a field-programmable gate-array board. It is therefore a challenging problem to design an efficient VLSI architecture to implement the DWT computation for real-time applications.

Keywords: DWT, IWT, Lifting.

I. INTRODUCTION

Discrete Cosine Transform (DCT) is a technique for converting a signal into elementary frequency components. It is widely used in image compression. Here we develop some simple functions to compute the DCT and to compress images. These functions illustrate the power of mathematic a in the prototyping of image processing algorithms. The discrete wavelet transform (DWT) is a multi resolution analysis tool with excellent characteristics in the time and frequency domains. Through the DWT, signals can be decomposed into different sub-bands with both time and frequency information. The coding efficiency and the quality of image restoration with the DWT are higher than those with the traditional discrete cosine transform. Moreover, it is easy to obtain a high compression ratio. As a result, the DWT is widely used in signal processing and image compression, such as MPEG-4, JPEG2000, and so on. Traditional DWT architectures are based on convolutions.

Then, the second-generation DWTs, which are based on lifting algorithms, are proposed. Compared with convolution-based ones, lifting-based architectures not only have lower computation complexity but also require less memory. Nevertheless, directly mapping these algorithms to hardware leads to relatively long data path and low efficiency. Recently, several novel architectures based on the lifting scheme have been proposed achieved an efficient folded architecture (EFA) with low hardware complexity.

However, its critical path delay is $T_m + T_a$, where $T_m$ and $T_a$ are the delay of a multiplier and an adder, respectively, and the computation time of EFA is quite long. Through optimizing the lifting scheme, Wu and Lin proposed a pipelined architecture to reduce the critical path to one multiplier and limit the size of the temporal buffer to $4N$, but it has one input and one output and cannot achieve high processing speed. Based on Wu and Lin’s design, Lai implemented the parallel 2-D DWT. The design is a pipelined two-input/two output architecture, and a $2 \times 2$ transposing module with four registers was developed. In addition, the critical path delay is one $T_m$. Nevertheless, it needs eight pipelining stages to complete the 1-D DWT and makes the total number of registers reach 22. The flipping structure is another important DWT architecture that was proposed by Huang. With a five-stage pipeline, the critical path can be also reduced to one multiplier.

However, the flipping structure has a large temporal buffer, and fewer pipelining stages lead to longer critical path delay. In this brief, further optimization on the lifting scheme is proposed to overcome shortages in previous works and minimize sizes of the logic units and the memory without loss of the throughput. By recombining the intermediate results of the row and column transforms, the number of pipelining stages and registers is reduced, while keeping the critical path delay as $T_m$. In addition, a novel architecture is developed to implement the 2-D DWT based on the above modified scheme. The parallel scanning method is employed to reduce the size of the transposing buffer. As a result, our design achieves higher efficiency.

The lifting scheme was first proposed by Daubechies and Sweldens in 1996. It shows that every finite-impulse response wavelet or filter bank can be factored into a cascade of lifting steps. That means the polyphase matrices for the wavelet filters can be decomposed into a sequence of alternating upper and lower triangular matrices multiplied by a
diagonal normalization matrix. The whole lifting scheme of the 9/7 filter has two lifting steps.

This paper proposes an improved version of lifting-based Discrete Wavelet Transform (DWT). The lifting based DWT architecture has the advantage of lower computational complexities transforming signals with extension and regular data flow. The main feature of the lifting based DWT scheme is to break up the high pass and low pass filters into a sequence of upper and lower triangular matrices and convert the filter implementation into banded matrix multiplications. Such a scheme has several advantages, including “in-place” computation of the DWT, integer-to-integer wavelet transform (IWT), symmetric forward and inverse transform, etc.

In order to assess the feasibility and the efficiency of the proposed scheme, the architecture thus designed is simulated and implemented on a field-programmable gate-array board. It is therefore a challenging problem to design an efficient VLSI architecture to implement the DWT computation for real-time applications.

Easily can extend the 1D Discrete Wavelet Decomposition and Reconstruction to 2D signal processing at each level, this Vi implements the 1D DWT on each row signal then, this Vi implements the inverse transfer with the reverse option. The following instructions shows the filter bank implementation for a 2D Discrete Wavelet.

Notice that the maxima(White Points) and the minima(Black Points) of Low-High are located around the column edges, the row edges and maxima and the minima of High-High are near the diagonal edges.

Basically wavelet transform (WT) is used to analyze non-stationary signals, i.e., signals whose frequency response varies in time, as Fourier transform (FT) is not suitable for such signals. To overcome the limitation of FT, short time Fourier transform (STFT) was proposed. There is only a minor difference between STFT and FT. In STFT, the signal is divided into small segments, where these segments (portions) of the signal can be assumed to be stationary. For this purpose, a window function "w" is chosen. The width of this window in time must be equal to the segment of the signal where it’s still be considered stationary. By STFT, one can get time-frequency response of a signal simultaneously, which can’t be obtained by FT. The short time Fourier transform for a real continuous signal is defined as:

\[ X(f,t) = \int_{-\infty}^{\infty} [x(t)w(t-\tau)] e^{-2\pi j ft} dt \]

Where the length of the window is \((t-\tau)\) in time such that we can shift the window by changing value of \( t \), and by varying the value \( \tau \) we get different frequency response of the signal segments.
The Heisenberg uncertainty principle explains the problem with STFT. This principle states that one cannot know the exact time-frequency representation of a signal, i.e., one cannot know what spectral components exist at what instances of times. What one can know are the time intervals in which certain band of frequencies exists and is called resolution problem. This problem has to do with the width of the window function that is used, known as the support of the window. If the window function is narrow, then it is known as compactly supported. The narrower we make the window, the better the time resolution, and better the assumption of the signal to be stationary, but poorer the frequency resolution:

Narrow window ===> good time resolution, poor frequency resolution.
Wide window ===> good frequency resolution, poor time resolution.

The wavelet transform (WT) has been developed as an alternate approach to STFT to overcome the resolution problem. The wavelet analysis is done such that the signal is multiplied with the wavelet function, similar to the window function in the STFT, and the transform is computed separately for different segments of the time-domain signal at different frequencies. This approach is called multiresolution analysis (MRA), as it analyzes the signal at different frequencies giving different resolutions.

MRA is designed to give good time resolution and poor frequency resolution at high frequencies and good frequency resolution and poor time resolution at low frequencies. This approach is good especially when the signal has high frequency components for short durations and low frequency components for long durations, e.g., images and video frames.

One Dimensional Discrete Wavelet Transform

This Module computes the transform for 1D Data

- Data is fetched from the memory module one by one
- Buffers included holds the temporary data
- After completion the result will be stored in the memory from where it will be dumped to file
- The architecture can be reconfigured for multiple levels of transform by redefining constants in this entity

![1-D Discrete Wavelet Transform(DWT)](image)

Two Dimensional Discrete Wavelet Transform

- This module is responsible for extending the 1D DWT module to images(2D)
- 2D transform is implemented by two passes of 1D transform
  - Horizontal Pass (Row wise), Vertical Pass (Column Wise)
- The buffers used here is twice the size of picture Data

Line-Based Folded Structure:

The line-based folded structure referred as (LBF) is shown in Fig. 1. It consists of one processor (for one-level 2-DWT) along with transposition memory (TN-MEM). Temporal-memory (TL-MEM) and FB. The 2-D DWT processor is comprised of one row-processor (RP) and one column-processor (CP). RP operates on input data matrix of size \((M \times N)\). It performs row-wise processing and generates a pair of intermediate matrices \([UL, UH]\) of size \((M \times N/2)\) each. From TN-MEM, CP receives the components of \([UL]\) and \([UH]\) columnwise and processes them to generate four subband matrices \([LL, LH, HL, HH]\) of size \((M/2\times N/2)\) each. CP requires a TL-MEM for temporary storage of partial results of filtering operation. Sizes of TN-MEM and TL-MEM are \((2Q - 1)N/2\) words and \(Q_{-}N\) words, respectively,

where \(Q\) represents the minimum number of rows of the intermediate components to be buffered to initiate column-processing and \(Q_{-}\) represents the number of lifting steps for lifting-based filter computation. For convolution-based computation of DWT, \(Q = K\) and \(Q_{-} = 0\). For lifting-based computation of DWT, \(Q = 2\) and \(Q_{-} = Kl\), where \(K\) is the lifting steps of the biorthogonal filter (for example, \(Kl = 2\) for 5/3 filter and \(Kl = 4\) for 9/7 filter). The \(LL\) subband is stored in FB. The processor starts receiving data from FB soon after it completes the first-level computation. FB is comprised of two memory units (Buffer-1 and Buffer-2), where Buffer-1 stores LL subband of odd-numbered levels and Buffer-2 stores LL subband of...
even-numbered levels. Buffer-1 and Buffer-2 are of size $MN/4$ and $MN/16$ words, respectively.

**Folded Design Using Parallel Data-Access Technique:**

The folded structure with parallel data access referred as(PDAF) is shown in Fig. 2. A block of $P$ samples is fed to the processor in every cycle. Input blocks are prepared from a set of $P$ consecutive columns of the data matrix such that adjacent sets of columns are overlapped by $(P - 2)$ columns, where $P = 3$ for lifting-based DWT and $P = K$ for convolution-based DWT. RP generates intermediate components column wise and these components are directly fed to the CP without transposing. Using parallel data access, TN-MEM is avoided but that introduces some additional complexity to the FB of the folded 2-D DWT, since the FB receives LL sub bands from the processor in serial order and feeds those components block-by-block back to the processor. Each buffer (Buffer-1 and Buffer-2) of the FB is designed accordingly.

The folded structure with parallel data access referred as(PDAF) is shown in Fig. 2. A block of $P$ samples is fed to the processor in every cycle. Input blocks are prepared from a set of $P$ consecutive columns of the data matrix such that adjacent sets of columns are overlapped by $(P - 2)$ columns, where $P = 3$ for lifting-based DWT and $P = K$ for convolution-based DWT. RP generates intermediate components column wise and these components are directly fed to the CP without transposing. Using parallel data access, TN-MEM is avoided but that introduces some additional complexity to the FB of the folded 2-D DWT, since the FB receives LL sub bands from the processor in serial order and feeds those components block-by-block back to the processor. Each buffer (Buffer-1 and Buffer-2) of the FB is designed accordingly.

The structure of Buffer-1 is shown in Fig. 3. It consists of two memory blocks (MBs) of size $(MN/8)$ words each and $(P - 2)$ shift registers (SRs) of size $M/2$ words. SRs are used to introduce down-sampling and necessary column overlapping in the data blocks. LL subband is split into even and odd columns by the DMUX. Even-numbered columns are buffered in MB-2, while the odd-numbered columns are buffered in MB-1. But the columns are retrieved simultaneously from MB-1 and MB-2 in the same column serial order as they were stored.

The data-block from these buffers alternatively during alternate decomposition levels. Each SR can be implemented using a pair of single-port RAM of size $M/2$ words and one MUX [shown in Fig. 3(b)]. Due to different size of MBs used in Buffer-1 and Buffer-2, different address signals are required for read/write operation. The control circuit required for the operation of the FB is, therefore, quite complex.
where on-chip storage represents the sum of registers, SRs, and RAM words required by the core. As shown in Table, the structure of requires less on-chip memory than the existing folded lifting-based structures. However, the structure of appears to be the most efficient structure since it requires nearly twice the arithmetic (multiplier and adder) components than those of and offers twice the throughput rate.

But it involves relatively less on-chip memory and FB. Compared with, the proposed structure requires nearly $2.625(K0/K1)$ times more multipliers and adders, $\approx (3K1M/4K/N)$ times more on-chip storage words and (16/5) times less ACT where $K0 = K1 + K2$. Compared with the parallel structure of [13], the proposed structure requires nearly $(PKI/8K0)$ times less multipliers and adders, $\approx (52N3K1M)$ times less on-chip storage and offers $16/P$ times less throughput rate.

The structure of is the most efficient among the existing convolution-based structures but computes only one-level DWT. Compared with the proposed structure requires 10.5 times more multipliers and adders, $\approx (3M/4)$ times more on-chip storage words and it has nearly 11 times less ACT. It is interesting to note that, the proposed structure does not involves FB unlike the existing structures where the size of the FB is $O(MN)$ for folded structure and $O(N)$ for parallel structure.

This is a major advantage for the proposed structure since the size of image is very often as large as $512 \times 512$. We have estimated the area and time complexities of the proposed structure and the existing structures for an image $(512 \times 512)$ excluding the input buffer of size $(M \times N)$ which is common for all competing designs. The memory overhead of the proposed structure is small compared to the size of the input buffer

II. SIMULATION RESULTS

We have synthesized the proposed design for three-level DWT based on Daub-4 and 9/7-filters along with the existing convolution-based designs of, and lifting-based designs by Synopsys Design Compiler using TSMC 90 nm CMOS library. We have synthesized only the core, since the FB is external to the chip. We have considered block size $Q = 16$ for the proposed design as well as the design. Besides, we have assumed 8-bit input pixels, 12-bit intermediate signals and image size $512 \times 512$ for all the designs. We have used the Synopsys Design Ware building blocks library for Wallacetree-based generic Booth-multiplier for all the designs. The netlist file is processed in Synopsys IC Compiler. After place and route the area, time and power, reported by the IC Compiler are listed in Table V. Since DRAM is conventionally used for the implementation of FB and 1-Mb DRAM requires approximately 138 671.9 $\mu$m2 in 90 nm process.2 Using this data, we have estimated the approximate area of the FB for comparison. We have estimated area-delay-product (ADP), 3

where $ACT =$ number of clock cycles required by the design to complete three-level 2-D DWT.

Synthesis results conforming the theoretical estimation are shown in Table. The lifting-based structure has the lowest DAT. As shown in Table V, the proposed structure using Daub-4 involves 1.28 and 16.19 times more core area and offers nearly 10.5 times less ACT in cycles than, respectively. But, it does not involve FB. As a result, proposed structure has 9.17 and 1.7 times less ADP than, respectively. Proposed structure using 9/7-filter involves 2.26 times more core area and offers 10.5 times less ACT than. The parallel structure and the proposed structure of has the same ACT, but involves 1.45 times less core area. Compared with, proposed structure involves 1.51 times less core area and it has 1.31 times less ACT. Proposed structure involves 3.3, 2.6, and 3.4 times less ADP than those of, respectively.

Memory Efficient High Speed Lifting Based VLSI Architecture For Multi-Level 2D-DWT

24
CONCLUSION

The Discrete Wavelet Transform provides a multi resolution representation of images. The transform has been implemented using filter banks. For the design, based on the constraints the area, power and timing performance were obtained. Based on the application and the constraints imposed, the appropriate architecture can be chosen. The latency of the proposed architecture is 44 clock cycles and throughput is 4 clock cycles, and hence is twice faster than the reference design. It is seen that, in applications, which require low area, power consumption, and high throughput, e.g., real-time applications, the poly-phase with DA architecture is more suitable. The biorthogonal wavelets, with different number of coefficients in the low pass and high pass filters, increase the number of operations and the complexity of the design, but they have better SNR than the orthogonal filters. First, the code was written in VHDL and implemented on the FPGA using a 32 x 32 random image. Then, the code was taken through the ASIC design flow. For the ASIC design flow, 8x8 memory considered to store the image. This architecture enables fast computation of DWT with parallel processing. It has low memory requirements and consumes low power. By using the same concepts which are mentioned above are useful in designing the Inverse Discrete Wavelet Transform (IDWT).

REFERENCES