FPGA Implementation of Discrete Wavelet Transform for Image Processing

Vineetha Soman

1 Department Of Electronics and Communication
1 Federal Institute of Science and Technology, Mahatma Gandhi University, Kerala, India

Abstract—In recent world, video and image compression have become very essential. There are many applications where we need to use different transform techniques to convert the signal or data in frequency domain. That’s why; availability of powerful software design tools is a fundamental requirement to take advantage of many advanced and specialized resources included in the latest devices. As we move in to the future there is a rising demand for high performance, high capacity and high bit rate wireless communication systems to integrate wide variety of communication services such as high-speed data, video and multimedia traffic. The separation of the sub carriers is theoretically minimal such that there is a very compact spectral utilization. DWT is faster than FFT hence it’s replaced by transform technique for OFDM. In this project we propose a multiplier less structure for FPGA hence cost effective. The main advantage of this improvement is that it can get the same throughput as traditional polyphase methodology. This project presents a novel approach to realize the DA algorithm. We propose to implement the architecture of DWT processor and verify the results on Spartan 3E FPGA. We will use MODEL.SIM for the verification of functional results of our scheme. We will try and compare the results with other proposed designs. We propose to implement DWT using Verilog HDL and it will be synthesized using Xilinx ISE.

Key words: discrete wavelet transforms, distributed arithmetic, field programmable gate arrays, image processing.

I. INTRODUCTION

Nowadays, implementing digital signal processing algorithms on field programmable gate arrays (FPGAs) becomes a growing trend, for the reason that FPGAs have merit on merging digital signal processing algorithms with other control logic. There has been a growing trend to implement DSP algorithms on Field Programmable Gate Arrays (FPGAs), because FPGAs offer higher level of parallelism than traditional processors. A field-programmable gate array (FGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing—hence "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC). The most common analog feature of the FPGA is programmable slew rate and drive strength on each output pin, allowing the engineer to set slow rates on lightly loaded pins that would otherwise ring unacceptably, and to set stronger, faster rates on heavily loaded pins on high-speed channels that would otherwise run too slow. This design style provides a means for fast prototyping and also for cost-effective chip design, especially for low-volume applications. A typical field programmable gate array (FPGA) chip consists of I/O buffers, an array of configurable logic blocks (CLBs), and programmable interconnect structures. The programming of the interconnects is implemented by programming of RAM cells whose output terminals are connected to the gates of MOS pass transistors.

Implementing DWT/IDWT on FPGA can extend FPGA’s application on digital signal processing, and some works about it have been reported. Among these works, most of them focus on DWT implementation of certain specific wavelet. There are two main approaches to implement DWT: convolution based and the lifting methods. Convolution-based methods are mainly based on Mallat’s pyramid algorithm. [5], which decomposes the input signal into frequency sub bands. This structure can be realized as a filter bank built by means of cascaded Quadrature Mirror Filters (QMF). For a ID-DWT, it translates to high and low-pass filter pairs followed by a decimation stage. The basic structure can be cascaded from the low-pass filter to have several levels of transformation. Sweden’s [6] presented an alternative method, known as the lifting scheme, which consists of splitting the signal into two parts and then finding a correlation between both to get rid of redundant operations. Thus, the number of computations is approximately reduced by a factor of 2 in comparison to convolution-based schemes. This is because the polyphase nature of DWT is seamlessly coupled into the lifting scheme. Huang [1] proposes a design to realize Daubechies 8 wavelet, which bases on Mallat algorithm and is synthesized on Altera Cyclone platform, while the maximum clock frequency of DWT and relevant IDWT can respectively achieve 217.72 MHz and 217.58 MHz Chilo [2] designs a frame work for the implementation of the 1D discrete wavelet transform on an FPGA using a polyphase structure along with the calculation unit structure and built-in DSP units. Patrick Longa [3] proposes an 8-tap Daubechies wavelet filter-bank architecture based on Distributed Arithmetic (DA) technique, and synthesizes it on Altera Stratix II FPGA platform, in which the resource cost is 614 LEs and the maximum clock frequency is 149.3MHz.

Comparing to the certain specific wavelet type DWT, universal wavelet type DWT has larger application area, because it can choose different wavelet basis functions to fit different occasions. However, report about universal architecture of DWT is not rich. S.Masud [4] designs a DWT intellectual property core with parameterized wavelet type and word-length, which is synthesized on Xilinx 4052XL FPGA platform, and occupies 614 LEs, while the...
maximum clock frequency is 78.85MHz. This design attempts to give a generic architecture of DWT, but its frequency is not enough to satisfy the demand of major real-time applications.

Proposed is a FPGA implementation of Discrete Wavelet Transform using Distributed Arithmetic algorithm with pipeline adder and add-shift method. DA Algorithm with Pipeline Adder been proposed as a resource complexity-less system for minimal requirement scenarios. Add-Shift been proposed as better resource utilization, better performance system even though the resource complexity is higher compared to Pipeline Adder method. This algorithm increases the resource utilization and reduces the power consumption.

II. DISCRETE WAVELET TRANSFORM ALGORITHM

A. Mallat Algorithm

Mallat algorithm is proposed in 1988[5] as a fast algorithm of discrete wavelet transform. As a leading theory of DWT hardware realization, it is widely used in DWT, just as the fast Fourier transform used in the classical Fourier analysis.

\[
y[n] = -2^k \sum_{i=0}^{\frac{N}{2}} y[i]x_{2^k} + x[n] + 2^{k+1} \sum_{i=0}^{\frac{N}{2}} y[i]x_{2^{k+1}} + x[n]
\]

(2)

Eq. 2 is the DA Mallat algorithm for our hardware implementation of discrete wavelet transform.

DA appeared as a very efficient solution especially suited for LUT-based FPGA architectures. This technique, first proposed by Crosier et al. [7], is a multiplier-less architecture based on an efficient partition of the function in partial terms using 2's complement binary representation of data. The partial terms can be pre-computed and stored in LUTs. The flexibility of this algorithm on FPGAs permits everything from bit-serial implementations to pipelined or fully parallel versions of the scheme.

Distributed Arithmetic structure is used to increase the resource usage while pipeline structure is also used to increase the system speed. In addition, the divided LUT method is also used to decrease the required memory units. However, according to Distributed Arithmetic, we can make a Look-Up-Table (LUT) to conserve the MAC values and callout the values according to the input data if necessary. Therefore, LUT can be created to take the place of MAC units so as to save the hardware resources.

Distributed arithmetic (DA) is a bit serial method of computing the inner product of two vectors with a fixed number of cycles. The original DA architecture stores all the possible binary combinations of the coefficients y[n] of (1) in a memory or lookup table. It is evident that for large values of L, the size of the memory containing the precomputed terms grows exponentially too large to be practical. The memory size can be reduced by dividing the single large memory (2L words) into m multiple smaller sized memories each of size 2n where \( L = m \times n \). The memory size can be further reduced to \( 2L^- \) and \( 2L^- \) by applying offset binary coding and exploiting resultant symmetries found in the contents of the memories. However, for very large values of L, the listed approaches still succumb to the limitations of storing the coefficient combinations in memory due to the exponential dependence of memory size on filter length.

\[
y[n] = \sum_{i=0}^{\frac{N}{2}} h[i]y[n-i]
\]

(1)

According to literature [7], we can get the distributed algorithm.

Fig. 1: schematic diagram of two-level DWT

Figure 1 gives a DWT structure based on Mallat algorithm. Using this method, DWT can be efficiently implemented by using cascaded quadrature mirror filters (QMFs), which are comprised of a high-pass FIR filter, a low-pass FIR filter and two 2-extraction stages, respectively.

In this paper, we propose a novel approach to realize the algorithm. Different from the polyphase filter partition [6], the filter is replaced by distributed arithmetic (DA) algorithm, a multiplier-less structure. This improvement can get the same throughput as traditional polyphase methodology, but the resource cost is further reduced. Inverse discrete wavelet transform is the inverse process of DWT and can be implemented by similar structure.

B. DA Algorithm

DA algorithm has parallel processing structure, and is widely used in the calculation of Multiplication-Accumulation(MAC) structures for its ability of improve calculation efficiency. For a distributed algorithm based FIR filter, the processing speed is decided just by the width of input data, but not the filter order, while the order affects FPGA resource cost.

An N-order FIR filter's output can be derived as linear convolution:

\[
y[n] = \sum_{i=0}^{\frac{N}{2}} h[i]y[n-i]
\]

According to literature [7], we can get the distributed algorithm.

Fig. 2: Basic 4-input LUT DA unit for the function Lrn.
Several observations provide valuable insight into the operation of a DA FIR filter. In a conventional multiply accumulate (MAC) based FIR realization, the sample throughput is to the filter length. With DA architecture the system sample rate is related to the bit precision of the input data samples. Each bit of an input sample must be indexed and processed in turn before a new output sample is available. For B-bit precision input samples, B clock cycles are required to form a new output sample for a non-symmetrical filter, and B+1 clock cycles are needed for a symmetrical filter. The rate at which data bits are indexed occurs at the bit-clock rate. The bit-clock frequency is greater than the filter sample rate s f and is equal to B s f for a non-symmetrical filter and (B+1) s f for a symmetrical filter.

Distributed Arithmetic for inner product generation can be easily implemented in the LUT-based Xilinx Virtex FPGAs. The inner product production basically consists of table lookup operations and additions. Thus RAM or ROM can be employed holding table values, and table lookup operations can be performed, and then a parallel adder usually follows to sum up LUT values provided by ROM or RAMs.

III. PROPOSED ARCHITECTURE OF DA ALGORITHM

A. DA Algorithm with Pipeline-Adder

Fig. 3: Architecture of Pipeline adder DA method
Signals are passed through Shift Registers on which they are multiplied with Filter Coefficients with resultant values stored in Look up Tables. Resultant values are passed through Pipeline Adder. Pipeline Adder method is a Resource complexity-less method and, has better performance compared to basic Convolution method.

B. DA Algorithm with Add-Shift Method

Fig. 4: Architecture of a add/shift DA algorithm
Instead of continuous addition type Pipeline adder, add-shift method is used. Intermediate values are passed through Accumulator and stored in Register until final values are generated. During addition each bit of next input signal is shifted to left and then added with corresponding bit values.

IV. A HIGH-SPEED FPGA ARCHITECTURE OF DWT/IDWT

This paper proposes the FPGA design and implementation of Discrete Wavelet Transform using Distributed Arithmetic algorithm with pipeline adder and add-shift method separately based on resource and performance attributes.

Below is a basic Architecture of High Speed DWT.

![Fig. 5: Architecture of high-speed DWT](image)

Since above calculation need long chain of adders, the system clock frequency is limited. In order to meet the requirement of real-time application, we propose the pipelined adder chain structure based on the improved DA Algorithm. The direct sum of long data list is break into some two-input adders, in which middle registers are involved to storage intermediate results. These two input adders are parallel, so the rate of algorithm calculation is greatly improved and the system clock frequency is improved.

V. SYNTHESIS RESULT

The design is synthesized and implemented on XYLINX SPARTAN 3 400 development board. It occupies 104 flip-flops, 246 registers and 67 IOB.DA algorithm increases resource-utilization on FPGA and reduces the power consumption.

A. Performance Analysis

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Convolution</th>
<th>Pipeline adder</th>
<th>Add shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Delay</td>
<td>16.791ns</td>
<td>42.484 ns</td>
<td>41.183 ns</td>
</tr>
<tr>
<td>Total memory</td>
<td>149440 KB</td>
<td>114040 KB</td>
<td>112184 KB</td>
</tr>
<tr>
<td>No. of Slices</td>
<td>1106</td>
<td>980</td>
<td>939</td>
</tr>
<tr>
<td>flip-flop</td>
<td>543</td>
<td>462</td>
<td>398</td>
</tr>
<tr>
<td>4 ip LUT</td>
<td>1888</td>
<td>1770</td>
<td>1702</td>
</tr>
<tr>
<td>IOBs</td>
<td>67</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td>Global Clock</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>MUX</td>
<td>NA (Multipliers)</td>
<td>1036</td>
<td>976</td>
</tr>
</tbody>
</table>

Table. 1: Performance Analysis
VI. MATLAB RESULT

Output was verified on MATLAB application with "KOALA" image as input. The approximation wavelet coefficient decomposition is the low-frequency output of the original image and it is similar as the original image. Although it has only half pixels of the original image, the original image is well described. In figure, the approximation wavelet coefficient decomposition (the low filter output) is shown in the left, and the high-frequency noise of original image is shown in the right. Thus, our design successfully retains original image’s overview in the low-frequency decomposition output, and extracts the noise in the high-frequency decomposition output.

![Fig. 6: original image](image1)

![Fig. 7: KOALA after first level Decomposition](image2)

VII. CONCLUSIONS

In this paper, high-performance FPGA implementation of discrete wavelet transform is proposed. Considering real-time and universal requirements of discrete wavelet transform, a high-speed architecture is designed. Our designs are realized on Xilinx Spartan 3 FPGA successfully, and verified by hardware tests. Compared with similar literatures, our high-speed FPGA design has maximum resource utilization and low power consumption. Even after image compression quality of the image will be maintained.

REFERENCES