J. Vis. Commun. Image R. 38 (2016) 297–306

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Memory-efficient high-speed VLSI implementation of multi-level discrete wavelet transform ☆,☆☆

Yongfei Zhang [a,b,*], Haiheng Cao [a], Hongxu Jiang [a,b], Bo Li [a,b]

[a] Beijing Key Laboratory of Digital Media, School of Computer Science and Engineering, Beihang University, Beijing 100191, China
[b] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China

ARTICLE INFO

Article history:
Received 4 November 2015
Revised 8 March 2016
Accepted 8 March 2016
Available online 9 March 2016

Keywords:
DWT
Multi-level
VLSI
Memory efficient
High speed

ABSTRACT

Memory requirements and critical path are essential for 2-D Discrete Wavelet Transform (DWT). In this paper, we address this problem and develop a memory-efficient high-speed architecture for multi-level two-dimensional DWT. First, dual data scanning technique is first adopted in 2-D 9/7 DWT processing unit to perform lifting operations, which doubles the throughputs per cycle. Second, for 2-D DWT architecture, the proposed Row Transform Unit and Column Transform Unit take advantage of input sample availabilities and provision computing resources accordingly to optimize the processing speed, in which the number of processors is further optimized to significantly reduce the hardware cost. Third, to address the problem of high cost of memory for the immediate computing results from each level and the computation time as resolution level increases, multiple proposed 2-D DWT units were combined to build a parallel multi-level architecture, which can perform up to six levels of 2-D DWT in a resolution level parallel way on any arbitrary image size at competitive hardware cost. Experimental results demonstrated that the proposed scheme achieves improved hardware performance with significantly reduced on-chip memory resource and computational time, which outperforms the-state-of-the-art schemes and makes it desirable in memory-constrained real-time application systems.

1. Introduction

Discrete Wavelet Transform (DWT) is an excellent multi-resolution analysis tool, through which signals can be decomposed into different subbands with both time and frequency information [1]. The coding efficiency and the quality of the recovery image with the DWT are higher than those with the conventional Discrete Cosine Transform (DCT). Moreover, DWT also provides resolution, distortion and spatial scalabilities, which are very difficult to achieve in DCT based compression systems. As a result, DWT has been widely adopted in signal processing and image compression [2–5].

The implementation of DWT of multiple decomposition levels, however, is highly computational intensive, which make it a great challenge to implement it for real-time applications [6].

The computation of DWT can be done either by convolution-based schemes [7,8] or lifting-based schemes [9–11]. Compared with convolution-based ones, lifting-based architectures not only have lower computation complexity but also require less memory [12,13]. The lifting steps are easy to implement and factoring the wavelet filter into lifting steps can reduce the computational complexity of the corresponding DWT by up to 50% [13,14]. Nevertheless, directly mapping the lifting algorithm to hardware [15] leads to a rather long critical path with the delay of $4T_m + 8T_a$, where T_m and T_a are the delay of a multiplier and an adder, respectively. Besides, direct mapping architecture costs an on-chip memory as large as the image, which is unbearable in many memory limited applications.

To overcome these problems, several architectures based on the lifting scheme have been proposed for efficient computation of DWT. Shi et al. [16] proposed a folded architecture, which achieves a low hardware area and a reduced critical path delay of $T_m + T_a$, with however a quite long computational time, since the folded architecture in nature performs computation iteratively. Through optimizing the lifting scheme, Wu and Lin [17] proposed a pipelined architecture to reduce the critical path delay to T_m, which however had only 1 input/output throughput per cycle and failed to exploit the parallelism of the algorithm.

☆ This paper has been recommended for acceptance by M.T. Sun.

☆☆ This work was partially supported by the National Hi-Tech Research and Development Program (863 Program) of China (2014AA015102), and the National Natural Science Foundation of China (61272502, 61272347).

* Corresponding author at: Beijing Key Laboratory of Digital Media, School of Computer Science and Engineering, Beihang University, Beijing 100191, China.
E-mail addresses: yfzhang@buaa.edu.cn, zhangyf.ac@gmail.com (Y. Zhang).

http://dx.doi.org/10.1016/j.jvcir.2016.03.014
1047-3203/

To improve the throughput of the circuit, Lai et al. [18] proposed a parallel 2-D DWT based on [17] with a pipelined 2 input/output throughput per cycle architecture, which however requires complex control circuit to complete the interlaced 1-D DWT and thus more registers in data paths. Designs in [19–21] have proposed block-wise scan technique and corresponding lifting unit, which further increases the 1-D throughput.

The 2-D DWT can be implemented through a 1-D row transform and a 1-D column transform separately, which requires memory buffer since column samples could not be processed till all the row samples are available in memory buffer. For direct mapping architectures, a memory buffer of size $N \times N$ is required, where N represents the width of image being processed, and hence it is usually off-chip to DWT processor. This leads to more power consumption in data accesses with external memory. An alternative approach is to start filtering column as soon as sufficient numbers of rows have been filtered which deals with intelligent memory management in 2-D DWT architectures [17]. Size of internal buffer memory for the filtered lines mainly depends on the length of the filter adopted. In the optimized architectures mentioned above, there still exists a problem that the computational time and memory cost will increase as the transform level increases. Wu et al. [22] and Mei [23] have proposed a line-based scanning scheme and a folded architecture for the computation of multilevel 2-D DWT level-by-level, and the computational time is increased as the level increases by the nature of serial processing. Besides, the level-by-level architecture [22] requires a memory of size $N^2/4$ for the intermediate LL subband to pass to the next level.

The above designs are so called folded architecture, in which multi-level DWT is conducted level by level iteratively. As the highly parallel algorithms and architectures can be used to speed up the signal processing process [24,6,25,26] studied the multi-level parallel architectures. Aziz [25] proposed a multi-level design of integer lossless 5/3 filter by using multiple instances of the level 1 processor operating in parallel, which implements lifting scheme with a single multiplier-free processing element to perform both predict and update operations. However, the multiplier-free processing element could not hold the precision well for the 9/7 DWT filter. Moreover, the throughput of the each level in [25] is limited to 1 input/output per clock cycle. A Pipeline architecture for lifting-based multilevel 2-D DWT is proposed in [6], in which each level of 2-D DWT computation is processed in separate computing blocks in cascaded pipeline structure, for concurrent computation of multilevel DWT without buffering the subband components. In [26], a pipeline VLSI architecture for fast computation of the 2-D discrete wavelet transform is proposed, in which the inter-stage parallelism is enhanced by optimally mapping the computational task of multi decomposition levels to the stages of the pipeline and synchronizing their operations while the intra-stage parallelism is enhanced by dividing the 2-D filtering operation into four subtasks that can be performed independently in parallel and minimizing the delay of the critical path of bit-wise adder networks for performing the filtering operation. Although the architectures in [6,26] achieve better parallelism by consuming more samples per cycle, which is however at the cost of more computing resource. More specifically, the architectures in [6] processes each level of 2-D DWT computation in separate computing blocks in cascaded pipeline structure and the architectures in [26] computes the four decomposed subbands at a level by performing four 2-D convolutions, which involves more computing and memory resources.

To address these problems, in this paper, a memory-efficient high-speed VLSI implementation scheme for multi-level DWT is proposed, which exploits the potential parallelism to a greater extent with higher speed and minimized memory cost. ***First***, to match the dataflow characteristic of 2 input/output throughput per cycle, the dual scanning technique is first adopted in the proposed lifting unit, which could help double the throughput per cycle. ***Second***, the hardware utilization is optimized through sparing the operations of splitting the even and odd samples, which keeps the multiplier in the lifting unit occupied all the time. To eliminate the repeated calculation in column transform, a pipeline buffering scheme is proposed, which facilitates fetch, calculation, update of the results from the intermediate lifting steps. ***Third***, to solve the problem of buffering intermediate LL subband and increasing computation time as the resolution levels increase, multiple instances of the proposed 2-D DWT processor are used to form the multi-level parallel processor, and the number of processors is further optimized to significantly reduce hardware cost with no deterioration in throughput. In all, the proposed architecture achieves parallel operation of processors at various levels, which minimizes the computational time and memory to a lowest level reported to date.

The rest of this paper is organized as follows. Section 2 reviews the lifting-based DWT scheme. Section 3 presents the proposed architecture for the multi-level 2-D DWT, and Section 4 provides implementation results and performance comparisons with the state-of-the-art architectures. Conclusion is drawn in Section 5.

2. Lifting-based DWT

DWT is performed through separating the low frequency information from high frequency counterparts. This operation can be realized in a number of ways, which could be level-by-level, block-based or line-based transformations. The lifting scheme [11] is a popular method to compute DWT and is accepted as a JPEG2000 compliant technique [3,4]. This method factorizes the wavelets into simple lifting steps and then performs the transform, which causes less computational complexity and memory cost, and supports in-place computation [11]. The lifting-based DWT is implemented basically in three phases, which are called split, predict and update, respectively, shown as in Fig. 1 [25].

The split phase separates even samples $x_e[n]$ and odd samples $x_o[n]$ from the input samples $x[n]$, represented in the following equation:

$$\begin{cases} x_e[n] = x[2n] \\ x_o[n] = x[2n+1] \end{cases} \tag{1}$$

The predict phase generates odd samples $x_o[n]$ based on even samples $x_e[n]$, which leads to the error $d[n]$:

$$d[n] = x_o[n] - P(x_e[n]) \tag{2}$$

where $P(\cdot)$ represents predict operator in form of interpolating polynomial. The predict is a reversible process, and $x_o[n]$ can be restored with $x_e[n]$ and $d[n]$, thus $x[n]$ can be reconstructed.

The update phase updates the predicted values $s[n]$ based on even samples as the follows:

$$s[n] = x_e[n] + U(x_o[n]) \tag{3}$$

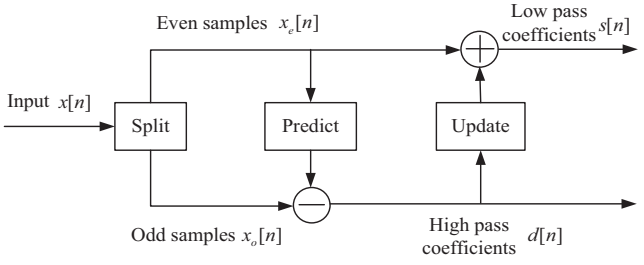

Fig. 1. Lifting-based method.