An Efficient 2-D DWT Architecture via Resource Cycling

Tay-Jyi Lin, Chein-Wei Jen

Department of Electronics Engineering
National Chiao Tung University, Hsinchu, Taiwan
Email: {tjlin, cwjen}@ee.nctu.edu.tw

ABSTRACT

Most DWT architectures reuse the filterbank modules for all decomposition levels to improve the hardware utilization but they should be implemented for the worst-case storage and computational precision requirements. We propose a novel reconfigurable 2-D DWT architecture in this paper that dynamically recycles the unused storage for computation resources with increasing precision as the decomposition level goes higher. We have implemented and verified this cost-effective architecture with PDA (Parallel Distributed Arithmetic) filterbank modules on the Xilinx XCV300-PQ240-6 FPGA and shown great improvement on PSNR/area ratio.

1. INTRODUCTION

Wavelet transform decomposes a signal into a family of wavelet coefficients of a unique mother wavelet Ψ translated by u and dilated by s, which can be written as a convolution product:

$$Wf(u,s)=\int_{-\infty}^{\infty} f(t)\frac{1}{\sqrt{s}}\Psi^{*}\left(\frac{t-u}{s}\right)dt = f\otimes\overline{\Psi_{s}}(u),\ \overline{\Psi_{s}}(t)=\frac{1}{\sqrt{s}}\Psi^{*}\left(\frac{-t}{s}\right)$$

The convolution computes wavelet coefficients with dilated band-pass filters. Discrete wavelet transform (DWT) has better spatial and spectral locality than the short-time Fourier transform (STFT) and discrete cosine transform (DCT), especially at boundary with discontinuities or signal spikes. It attracts more and more attention in various digital signal processing applications, such as audio/image analysis, compression, computer vision, pattern recognition and so on [1]. DWT-based image compression proves high quality at very low bit-rate and recent international standards, such as JPEG-2000 and MPEG-4, have chosen DWT as a primary tool for image de-correlation.

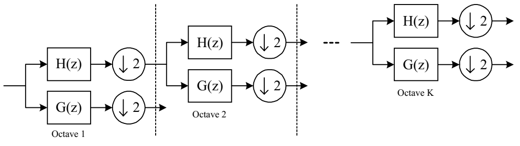

Figure 1. Pyramid Algorithm (1-D DWT)

Mallat proposed the pyramid algorithm (PA) [2], which is much more efficient than computing the full set of inner products. It applies a two-band subband transform with identical filterbank modules in an iterative fashion and computes the wavelet coefficients bottom up as shown in Fig.1. Along the decomposition tree, the dynamic range and precision requirements increase rapidly and the number of bits for intermediate variables needs increasing to guarantee acceptable quality in fixed-point implementations. Inversely, the storage in the blocking-scheduled, inter-octave folded DWT architectures described later are halved every octave. In this paper, the multiplying unused storage resources are incrementally recycled for wider arithmetic units to process the intermediate variables with growing wordlength. Simulation results show our approach improves the PSNR of the 8-bit fixed-precision 2-D DWT by 25dB with 20% area overhead. This paper is organized as follows. First, background material on DWT architectures and dynamic reconfiguration is provided in section 2. Section 3 describes the underlying 2-D DWT architecture that supports resource cycling (redistribution) discussed in section 4. FPGA implementation with simulation results is available in section 5. Section 6 concludes this work.

2. BACKGROUND

2.1 VLSI Architectures for DWT

Directly PA-mapped hardware with identical filterbank modules is wasteful because data rate of higher octaves is much smaller. Two techniques can increase the hardware utilization – ***intra-octave folding*** and ***inter-octave folding*** as shown in Fig.2. Intra-octave folding scales down high-octave hardware for reduced data rates. Fig.3 shows a 2-folded decimation filterbank in transpose form that introduces no additional registers. Unfortunately, hardware scaling for higher octaves is not trivial and bi-orthogonal banks in most standards with different filter lengths for low- & high-pass filters complicate this kind of folding transformation [3]. Digit-serial architectures with halved digit sizes for higher octaves are proposed in [4]. Extra data format converters with additional storage elements, complex routing and control are required to convert among the various digit sizes.

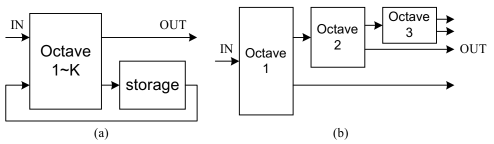

Figure 2. (a) Inter- (b) Intra- Octave Folding

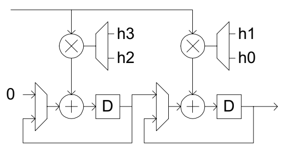

Figure 3. 2-folded Decimation Filter with Polyphase Decomposition

Inter-octave folding schedules the wavelet computations on one filterbank hardware. ***Blocking scheduling*** with the worst-case storage of *O(N)* computes the wavelet coefficients from the lowest to the *K*th octave in order, where *K* is the number of decomposition levels (i.e. the number of octaves). New arrival input signals are blocked during decomposing the high octaves (2~*K*th). The data processing rate of the 1-D DWT is

$$N+\frac{N}{2}+\cdots+\frac{N}{2^{K-1}}=\sum_{i=1}^{K}\frac{N}{2^{i-1}}=2\left(1-2^{-K}\right)N,$$

which approximates *2N* if there exist sufficient decomposition levels. This suggests ***non-blocking*** (***online*** or ***interleaving***) ***scheduling*** interleaves all higher-octave computations into the lowest octave to achieve 100% hardware utilization and reduces the storage size to *O(KL)*, independent of input size *N*, where *L* is the number of filter taps. Fig.4 shows the modified recursive pyramid algorithm (MRPA), a widely used non-blocking scheduler with a generalized hardware architecture shown in Fig.5. Various implementations, such as MUX-based, semi-systolic, systolic and RAM-based routing have been summarized in the literature [5].

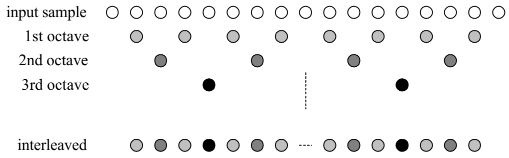

Figure 4. Modified Recursive Pyramid Algorithm

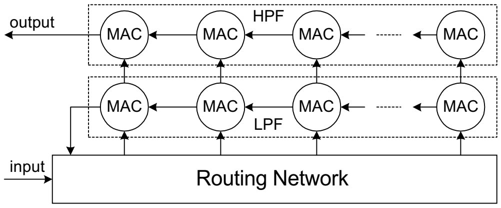

Figure 5. General MRPA-based 1-D DWT Architecture

In this paper, we only discuss separable 2-D DWT as shown in Fig.6, which significantly reduces the computations in non-separable 2-D DWT by introducing a transpose memory. Various combinations of folding techniques for 1-D DWT can be conducted to optimize the 2-D DWT architecture, which will be discussed in next section.

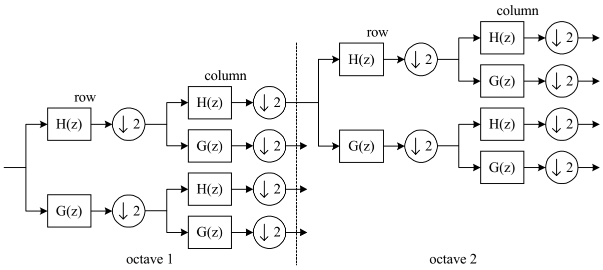

Figure 6. Separable 2-D DWT

2.2 Dynamic Reconfiguration

Dynamic reconfiguration can modify the system functionality and/or interconnection during execution. It is either data-driven or statically scheduled. SRAM-based FPGA [6], a distributed parallel RAM computational model with programmable routing resources is a widely known underlying infrastructure to support dynamic reconfiguration. The adaptation of functionality/interconnection involves a processor, a memory system, even with the support of a compiler and OS together. Dynamic reconfiguration can redistribute the resources among storage, computation and interconnection [7] in a computing machine at run time to prevent worst-case design for varying resource requirements in different situations.

3. PROPOSED 2-D DWT ARCHITECTURE

Intra-octave folding requires $2^{5\sim11}$-folded filters for 3 to 6 decomposition levels and is very impractical for 2-D DWT, even with digit-serial architectures. To schedule all 2-D wavelet coefficient computations on a single filterbank module would complicate the dataflow control due to the embedded transpose memory. Fig.7 shows our proposed alternative architecture of inter-octave folding. Blocking scheduling is chosen because the data rate of the 2-D DWT –

$$N^2+\frac{N^2}{4}+\cdots+\frac{N^2}{4^{K-1}}=\sum_{i=1}^{K}\frac{N^2}{4^{i-1}}=\frac{4}{3}\left(1-4^{-K}\right)N^2,$$

prevents non-blocking scheduling such as MRPA from full hardware utilization. The following subsections explain the major components respectively – two ***filterbank modules***, two ***stream interface units*** (SIU) and one ***external interface unit*** (EIU).

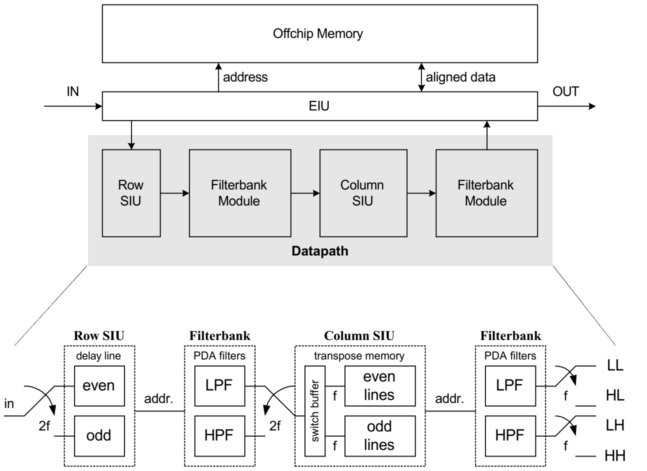

Figure 7. Proposed 2-D DWT Architecture

3.1 Filterbank Module

An *L*-tap FIR filter performs the inner product of two vectors of length *L*, which can be represented as

$$y=A\cdot X=\sum_{i=1}^{L}A_iX_i=\sum_{i=1}^{L}A_i\left(\sum_{k=1}^{N}X_{i,k}2^{-k}\right)=\sum_{k=1}^{N}\left(\sum_{i=1}^{L}A_iX_{i,k}\right)2^{-k},$$

where A_i and X_i are coefficients and inputs respectively, both in unsigned fraction for simplicity and N and L represent wordlength and filter-length respectively. Distributed arithmetic (DA) [3] filters store pre-computed results (bracketed rightmost) in a table and access it with N L-aligned bit-level input. The N lookup values are accumulated to produce an output. In brief, DA performs L N-bit MAC operations in N $2^L \times N$-bit table lookup with only one N-bit multiplication (accumulation) and the input P/S (parallel to serial) conversion overhead. Parallel DA (PDA) with word-level interfacing implements N independent ROMs with a reduction adder tree (identical to the partial product reduction tree in general multipliers) as shown in Fig. 8(a) to remove the bit-level P/S conversion.

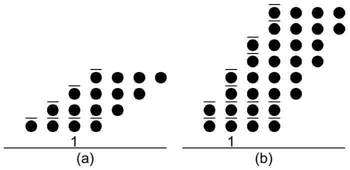

Figure 8. 4-bit Baugh-Wooley Reduction Adder Tree Example (a) without (b) with Polyphase Decomposition

Polyphase decomposition [8] can eliminate all unnecessary computations in decimation filters. In the DA implementation of polyphase filterbank modules, the lookup table size is decreased from 2^L to $2 \times 2^{L/2}$ with a double-sized reduction adder tree as shown in Fig. 8(b) and modified routing to split even and odd samples in the SIU described in the next subsection.

3.2 Stream Interface Unit (SIU)

The two PDA-based filterbank modules in Fig.7 are identical. SIUs are used to generate proper data sequences required by parallel functional units with very high bandwidth [9]. Both row and column SIUs buffer the data samples to be filtered and generate the data addresses to access ROM tables for pre-computed values in DA. The SIU splits even and odd samples to access individual ROMs (generated by even and odd filter coefficients separately) for the PDA-based filters with polyphase decomposition in the proposed filterbank modules. The transpose memory of size $O(LN)$, where L represents the longer tap in biorthogonal filters, is embedded in the column SIU.

The decomposition inside one octave is a binary (symmetric) tree, instead of an asymmetric tree between octaves (i.e. only L-band is further decomposed). The column SIU takes the responsibility to interleave the two outputs from the row filterbank module with the same total data rate into the column filterbank module to achieve 100% utilization without folding the column filters such as Fig.3, which does not work for DA-based filters.

3.3 External Interface Unit (EIU)

The EIU receives the four-band outputs (LL, HL, LH, HH) from the column filterbank module and aligns them in words to be stored in off-chip memories. LL bands would be read back for further decomposition. Video and streaming images could be supported with our proposed 2-D DWT architecture with an additional ping-pong buffer embedded in the EIU for our blocking scheduler.

4. RESOURCE CYCLING VIA DYNAMIC RECONFIGURATION

Designers always try to maximize hardware utilization and prevent functional units from being idle to boost the performance on the minimum hardware. Multiple tasks/functions can share common resources extracted via similarity exploitation with tolerable routing complexity, such as fused DCT/IDCT modules and fused DWT/IDWT modules [10]. Various task/function scheduling heuristics such as reservation-table based [11] can support more complex resource sharing and reduce conflicts. Dynamic reconfiguration, which is able to change the system functionality and/or interconnection at run time, introduces a new dimension in the similarity exploitation for the common resources known as the ***reconfigurable fabric*** or the ***metamer***. Hardware resources can be cycling among more distinct tasks or functions.

Computing machines are composed of logic, storage and routing elements. RAM is a generic computing model that functions as either one of the three elements and can be regarded as the fundamental metamer for general-purpose computing. FPGA, which consists of distributed RAMs (LUT) and programmable routing, is an efficient platform for various applications. Various descendants of FPGA are proposed with some specialization or enhancements to reduce the configuration overheads (i.e. the time required to adapt the functionality and/or interconnection) significantly.

Table 1 depicts that the storage resources for transpose memory (in column SIU) are halved every octave while the number of bits to keep full precision shown in the last column grows dramatically in fixed-point implementation. The actual increase of the wordlength is to keep the computation error under some acceptable level and prevent overflow when the dynamic range grows. We propose a novel approach with ambitious resource cycling in finer granularity for our proposed 2-D DWT architecture – the storage resources are recycled for logic to support wider arithmetic units. It is rather straightforward for memory-based PDA filters, which support wider computations just by incorporating more identical ROMs that store pre-computed values and expanding the reduction adder tree. The data width of the SIU and the EIU needs widening, too.

Table 1. Opportunity for Resource Cycling in 2-D DWT

	Data Samples	Transpose Memory	Full-Precision Multiplication
1st Octave	$N \times N$	LN	$n \times m$
			$(n+m+p) \times m$
2nd Octave	$(N/2) \times (N/2)$	$LN/2$	$(n+2m+2p) \times m$
			$(n+3m+3p) \times m$
3rd Octave	$(N/4) \times (N/4)$	$LN/4$	$(n+4m+4p) \times m$
			$(n+5m+5p) \times m$

$N \times N$: image size; L: number of taps; p = log2 (L)
n: input wordlength; m: filter wordlength;

Assume the PDA filterbank module is composed of D-bit ROM per input bit and the wordlength increases l bits per filtering operation, the hardware resources in the first octave can be approximated as $Dn+NL(n+l)+D(n+l)$, where n is the wordlength of input signals. The three terms represent hardware resources for the row filters, the transpose memory and the column filters respectively and the delay lines and reduction trees are ignored. Similarly, the hardware for the second octave can be approximated as $D(n+2l)+(N/2)L(n+3l)+D(n+3l)$. For fully recycling, the increasing wordlength per filtering operation is

$$l = \frac{NL \cdot n}{NL + 8D}.$$

5. SIMULATION RESULTS

We choose an image size of 512×512 and the CRF (13,7) coefficients, the default integer filter for lossy compression in JPEG2000 [12]. The analytic number of increasing bits per filtering operation (l) for 100% hardware recycling is 2.4 (bits). We have chosen 8-bit row- and 10-bit column-filterbank modules for the first octave in our implementation. These two modules are reconfigured to process 12- and 14-bit samples respectively for the second octave, 16- and 18-bit for the third octave, and so on. The radix point is adjusted for each filtering operation to prevent overflow without saturating arithmetic for simplicity. Bit-true simulation shows our approach has better performance/cost ratio than the fixed 8-bit and 16-bit versions as shown in table 2. The central two columns are PSNR of the reconstruction images from 3-level wavelet coefficients. Hardware cost represented in area is normalized to the fully 8-bit version as shown in the fourth column, where the 20% hardware increase in the proposed architecture is the wider column filterbank module and some overheads.

Table 2. Comparison of Fixed-point 2-D DWT

	PSNR (dB)		Area
	Lena	Baboon	
8-bit	39.2899	39.3210	1.0
16-bit	87.4878	87.4892	2.0
Proposed	64.9259	64.9253	1.2

We have an FPGA prototype on the Xilinx XCV300PQ240-6 – one of the Virtex-series FPGA [13], which is capable of column-based partial reconfiguration. The lookup tables (ROM) that store pre-computed values for the PDA-based filterbank modules can be efficiently mapped onto the LUT, which is conventionally used to implement logic functions in FPGA. We have a manual floorplan to place distinct blocks among octaves (i.e. blocks that required to be reconfigured) on same columns to reduce the reconfiguration overheads. This prototype can be reconfigured at 50MHz in the SelectMAP mode [14] under a host processor. No operation is allowed during reconfiguration. This prototype operates at 40.85MHz for normal operations and can decompose up to 60.89 512×512 8-bit grayscale images per second (including configuration time) into three decomposition levels.

6. CONCLUSION

This paper presents a reconfigurable 2-D DWT architecture to avoid worst-case storage and functional unit designs by resource cycling. Analyses on wordlength increasing rate to reduce error are also given. FPGA, a general-purpose metamer (reconfigurable fabric), serves as the underlying infrastructure of our proposed reconfigurable 2-D DWT architecture in this prototype. Layout is optimized via manual floorplan tuning for rapid column-based partial reconfiguration on Xilinx Virtex architectures. We will continue on a more specific metamer design in ASIC with fast ROM cloning mechanisms for PDA while overlapping reconfiguration times with normal operations. An embedded configuration controller will be considered in the future with minimal on-chip configuration bit-streams.

ACKNOWLEDGEMENT

This work was supported by the National Science Council, Taiwan, ROC under Grant NSC89-2215-E009-078

REFERENCE

[1] S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd Edition, Academic Press, 1999

[2] S. Mallat, “Multifrequency Channel Decompositions of Images and Wavelet Models,” *IEEE Transactions on Acoustics Speech and Signal Processing*, Dec 1989

[3] K. K. Parhi, *VLSI Digital Signal Processing Systems – Design and Implementation*, John Wiley & Sons, 1999

[4] K. K. Parhi, T. Nishitani, “VLSI Architectures for Discrete Wavelet Transforms,” *IEEE Transactions on VLSI Systems*, June 1993

[5] C. Chakrabarti, M. Vishwanath, R. M. Owens, “Architectures for Wavelet Transforms – A Survey,” *Journal of VLSI Signal Processing*, Nov 1996

[6] J. V. Oldfield, R. C. Dorf, *Filed Programmable Gate Arrays*, John Wiley & Sons, 1995

[7] T. Fujii, et al, “A Dynamically Reconfigurable Logic Engine with a Multi-Context/Multi-Mode Unified-Cell Architecture,” *International Solid State Circuits Conference (ISSCC’99)*, 1999

[8] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, 1992

[9] T. J. Lin, C. W. Jen, “Data Stream Generation for Concurrent Computation in VLSI Signal Processors,” *International Conference on Signal Processing (ICSP’00)*, Aug 2000

[10] T. Acharya, “A High Speed Reconfigurable Integrated Architecture for DWT,” *IEEE Global Telecommunications Conference*, 1997

[11] V. K. Madisetti, *VLSI Digital Signal Processors*, IEEE Press, 1995

[12] ISO/IEC JTC1/SC29/WG1 N1135, *JPEG 2000 Verification Model Version 3.0 (B)*, Dec 1998

[13] *The Programmable Logic Data Book*, Xilinx, 2000

[14] XAPP 151, *Virtex Configuration Architecture – Advanced User’s Guide*, Xilinx, Sep 1999

[15] P. C. Wu, L. G. Chen, “An Efficient Architecture for Two-dimensional Discrete Wavelet Transform,” *International Symposium on VLSI Technology, Systems and Applications*, 1999.

[16] R. Tessier, W. Burleson, “Reconfigurable Computing for Digital Signal Processing – A Survey,” *Journal of VLSI Signal Processing*, 2000