Precision Aware Self-Quantizing Hardware Architecture for the Discrete Wavelet Transform.

C. Bhaskar[1], P.Raja Pirian[2].
Assistant Professor/Odaiyappa College of Engineering and Technology, Theni, Tamil Nadu, India[1].
Assistant Professor/ECE Kings College of Engineering, Thanjavur, Tamil Nadu, India[2].
bhaseceme@gmail.com[1] rajapirian@gmail.com[2]

*AB*STRACT—**In this work I present a design for both bit-parallel(BP) and digit-serial(DS) precision-optimized implementations of the discrete wavelet transform(DWT), with specific consideration given to the impact of depth(the number of levels of DWT) on the overall computational accuracy. These methods allow customizing the precision of a multilevel DWT to a given error tolerance requirement and ensuring an energy-minimal implementation, which increases the applicability of DWT-based algorithms such as JPEG 2000 to energy-constrained platforms and environments. Additionally, quantization of DWT coefficients to a specific target step size is performed as an inherent part of the DWT computation, thereby eliminating the need to have a separate downstream quantization step in applications such as JPEG 2000 .Results indicate that while BP designs exhibit inherent speed advantages, DS designs require significantly fewer hardware resources with increasing precision and DWT level. A four-level DWT with medium precision, for example, while the BP design is four times faster than the digital- serial design, occupies twice the area.**

***Index Terms*— Fixed point arithmetic, image coding, very large scale integration (VLSI), wavelet transforms.**

I. INTRODUCTION

The JPEG 2000 standard offers considerable coding efficiency and flexibility advantages over the original block DCT-based JPEG standard, it has yet to be widely adopted for several years since the standardization was completed. A key element of JPEG 2000 is the discrete wavelet transform (DWT), which recursively decomposes an input image into sub bands with different spatial frequency and orientation. The most commonly used DWT filters in JPEG 2000 are the biorthogonal lossless 5/3 integer and lossy 9/7 floating-point filter banks. In this paper, we focus on the DWT using 9/7 filter, which provides very good compression quality but is particularly challenging to implement with high efficiency due to the irrational nature of the filter coefficients. Although there is a rich literature on different hardware implementations of the DWT and novel DWT algorithms, there has been much less attention directed to approaches in which the precision of the DWT computation is specifically considered as a design goal. The work in considers the effects of quantizing the lifting coefficients of the 9/7 DWT. The number of canonical signed digit (SD) terms for the coefficients are varied, and their effects on the peak signal-to-noise ratio (PSNR) and hardware area/speed are evaluated .The work in examines the effect on PSNR when quantizing filter coefficients for a convolution- based 9/7 DWT, and focuses on analyzing dynamic range requirements of the DWT across different sub bands and decomposition levels. In contrast to the previous work, which has been primarily directed to filter coefficients, we address simultaneous optimization of not only the coefficient precision but also the internal data paths used in their computation and present a solution that is fully generalized with regard to precision, allowing design of a DWT to any desired accuracy. Using this approach, we show that the optimization technique can be used to minimize operand bit widths in a bit-parallel (BP) architecture and to minimize iterations in a digit-serial(DS)architecture. This enables implementations with a significant improvement in hardware resources and/or execution time while also ensuring that overflows are avoided and precision requirements are met.

In addition, we describe a highly flexible overall DWT architecture in which the target precision and number of DWT levels are configurable at run time. In the approach here, the quantization of the DWT coefficients to a target step size is inherently performed through the process of computing the DWT; thereby eliminating the need for a separate quantization step after the DWT is completed. While any hardware DWT implementation will of course result in DWT coefficients that are quantized in accordance with the precision used to compute and represent the DWT coefficients, traditionally, in JPEG 2000, quantization has been thought as a separate downstream processing step occurring after the DWT. While there are some environments in which the approach of taking a high-precision DWT and then lowering the precision through a subsequent quantization step will be still appropriate, there are also many applications, including those based on configurable hardware, in which it is more optimal to jointly address the DWT computation and coefficient quantization.To examine the specific hardware performance and tradeoffs associated with the solutions presented here, design implementations targeting a 90-

nm CMOS process are described, and the quantitative area, speed, and energy characteristics are presented. The rest of this paper is organized as follows: gives an overview of the lifting-based DWT and JPEG 2000 quantization.

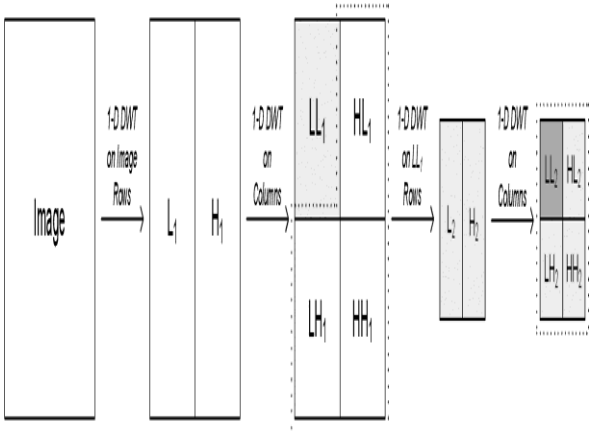
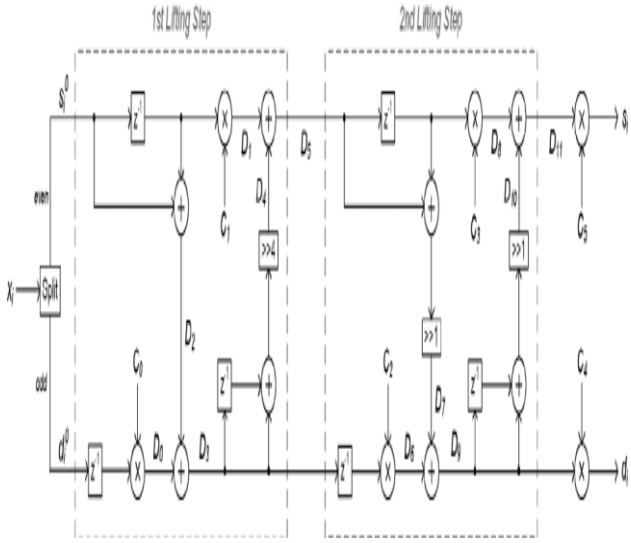


Fig. 1. Illustration of two-level wavelet decomposition. The dotted portions are the final wavelet transformed data.



and HH1 are obtained. This process can be recursively applied on LL1 to produce the LL2, HL2, LH2, and HH2 sub bands. The 9/7 DWT was originally implemented via convolution based methods, in which low-pass and high-pass FIR filters are employed. DWT can be decomposed into a finite sequence of lifting steps, which provides several advantages including lower computation and memory requirements and easier boundary management. When lifting is used, the 9/7 filter can be expressed using the following steps:

$$P(z) = \begin{bmatrix} 1 & \alpha(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & \gamma(1+z^{-1}) \\ 0 & 1 \end{bmatrix} X \begin{bmatrix} 1 & 0 \\ \delta(1+z) & 1 \end{bmatrix} \begin{bmatrix} \zeta & 0 \\ 0 & 1/\zeta \end{bmatrix}$$

Where $\alpha = -1.586134342$, $\beta = -0.05298011854$, $\gamma = 0.8829110762$, $\delta = 0.4435068522$ and $\zeta = 1.149604398$

Fig. 2 illustrates the flipping structure for the lifting-based 1-D 9/7 DWT. Although the flipping structure shares the same computational complexity with the traditional lifting scheme, it reduces the critical path considerably by flipping computation units with the inverses of multiplier coefficients. Constants are given by

$C0 = 1 / \alpha = -0.6304636206$
$C1 = 1 / (\alpha\beta) = 0.7437502472$
$C2 = 1 / (\beta\gamma) = -0.6680671710$
$C3 = 1/(\gamma\delta) = 0.6384438531$
$C4 = 1/(\alpha\beta\delta/\zeta) = 2.065244244$
$C5 = \alpha\beta\gamma\delta\zeta = 2.421021152$

Fig.2. Flipping structure for the lifting-based 1-D 9/7 DWT.

II. DISCRETE WAVELET TRANSFORM

A. Lifting Approach

There are of course many references describing the DWT. For clarity, we briefly describe DWT aspects that are directly relevant to the subsequent design discussion. Fig.1 illustrates the steps for performing a two-level DWT on an image. The 1-D DWT is first performed on the rows of the image producing low-frequency L1 and high-frequency H1components. After performing a 1-D DWT again on the columns of L1 and H1, the first level of decomposition is completed, and LL1, HL1, LH1, The core is the 1-D DWT module, which performs the actual wavelet transform. The controller manages the overall operation of design by generating control signals for the buffer and the filter.

B. Quantization

III. BP DWT DESIGN

Quantization is a key element for the lossy 9/7 DWT in governing achievable compression performance. The JPEG 2000 standard supports uniform dead-zone quantization, as well as trellis coded quantization.

Uniform dead-zone quantization is chosen in this paper due to its simplicity and hardware efficiency. This quantization approach uses equally sized bins, except for a quantizer "dead zone" centered at zero containing a Bin double the size of the others. In typical implementations, the quantization step size is specified for the highest resolution sub band and is decreased by a factor of two for each subsequent decomposition [1], [14], [15], thereby quantizing the sub bands in approximate accordance to their MSE contributions. *Static Optimization:* The worst case (maximum absolute error) quantization errors for truncation and round-to-nearest are given by

Truncation : $Ez = \max(0, 2^{-FBz} - 2^{-FBz'})$

$$\begin{cases} 0, & \text{if } FBz >= FBz' \\ 2^{-FBz-1}, & \text{otherwise} \end{cases}$$

Round to nearest : $Ez =$

We first consider a BP approach, which is appropriate when computing speed is the primary goal. Given, the lifting framework is described earlier, the design challenge lies in determining the appropriate number of integer and fractional bits to use in representing all the signals utilized during the computation. In the discussions, that follow two's complement fixed-point representation is used for all the signals.

A. Integer Bit-Width Determination

It can be implemented via a BP, DS architecture, or a run-time configurable architecture. The dual-port buffer is large enough to hold two data and is used to store the original raw data, intermediate data, and/or the final transformed data.

For IB determination, we use the approach described in which is based on computing the roots of the derivatives of each signal. Since the binary point needs to be aligned for additions, the two addition operands need to share the same IB.

B. Fractional Bit-Width Optimization

The fractional bit-width optimization is executed in two steps a static step based on analytical models to obtain the initial set of bit widths , followed by a dynamic step based on simulation that further reduces the bit widths using a PSNR delta threshold. The target precision metric used for the static step is the unit in the last place (ulp) error criterion, which is a way of specifying the worst case (maximum absolute) error. The static step finds the set of bits that guarantee less than 2- ulp error at the final quantized DWT outputs. The internal data paths are quantized using standard truncation toward infinity by chopping off least significant bits, whereas the final DWT outputs are quantized toward zero. The bit widths of the internal data paths are found using the error expressions above in conjunction with simulated annealing. Since the quantization scheme of JPEG 2000 uses increasing precision with, the bit widths are dominated by the precision requirements of L.

1) Dynamic Optimization: The analytical optimization scheme is conservative in the sense that it assumes that the worst case error can concurrently occur at all nodes, which is extremely likely to occur in practice. As a result, the computed FBs will be in general larger than required. Moreover, it is PSNR, not internal arithmetic accuracy, which is used in practice for numerical assessment of images represented by inverse transforming a quantized DWT representation. In order to tune the precision decisions to more closely relate to PSNR, we perform a secondary simulation-based bit-width refinement step to further reduce the FBs. Using binary search, the FBs are uniformly reduced until either 1) the PSNR loss of a set of test images fall above 0.1 dB compared with the statically optimized set the error of any of the DWT outputs exceed 2 ulp. This process typically reduces the FBs by approximately 20%.

IV. DS DWT DESIGN

While DS arithmetic has a significant advantage over BP in terms of circuit area, a key challenge in DS design involves minimizing number of iterations. For the DS representations used here, we use a radix-2 SD redundant number system. Due to redundancy SD operations do not propagate carriers and hence they are able to run in most significant digit first MSDF. Mode (also known as online arithmetic). This MSDF property makes it attractive for the DS DWT approach since it allows for varying the number of iterations to obtain different precision. In radix-2 SD, the following set is used to represent a digit: {-1, 0, 1}. we use binary bits b'10, b'00, and b'01 to indicate 1, 0, and 1, respectively.

The incoming two's complement data is first serialized and converted into SD representation. The serial SDs is then passed into the DS DWT, which is partitioned into nine pipeline stages that run in parallel. After the last stage, the DWT-transformed data is converted back into two's complement representation and parallelized into words. This approach reduces the memory requirement since two's complement occupies half the area of the Equivalent SD representation. Both SD addition and SD multiplication produce one digit per cycle, starting from the most significant digit.

B. Integer Width Determination

As in the BP approach, the goal here is to use the minimum number of integer digits for each signal while avoiding overflow. Moreover, the number of integer digits of the addition operands need to be identical for binary point alignment. The binary point of a digit can be adjusted via increasing or decreasing the number of integer digits. This is easily achievable for the BP case by simple shifting. In MSDF, however, the number of

V. SYSTEM MODEL AND DESCRIPTION

integer digits needs to be adjusted by inserting and removing delay elements, e.g., registers. We conduct the integer digit analysis for i=0. The analysis ensures that the number of integer digits for all paths increase by one with, ensuring that variable shifters (which are expensive in hardware) are not required.

C. Minimizing the Number of DS Iterations

In a DS implementation, increasing the number of iterations gives more precision but costs more execution time. The goal of iteration optimization is thus to use the minimum number of iterations while meeting the specified error requirement. This is analogous to determining the minimum number of fractional bits with the direct approach. The worst case error for the DS addition z=x+y is given by

$$E_z = E_x + E_y + \max(0, 2^{-FDz} - 2^{-FDz}) + \max(0, 2^{-FDz} - 2^{-Fdy})$$

Where the last two terms are quantization errors due to using a subset of digits, which is a function of the number of iterations.

The BP and digit-serial architectures enable optimized computation of a single level of the DWT at a single precision requirement. However, many DWT applications involve multilevel DWT decompositions. Thus, it is of high interest to have a single reconfigurable DWT processor that supports different DWT levels and precision at run time. Varying these parameters provides the ability to vary compression ratios, image quality, and processing time. Adding this flexibility to the BP approach would mean that all of the operators would need to be large e ough to support the highest level and precision. When performing DWT at a low level and/or precision, this would involve significant hardware inefficiency.

As a function of the DWT level. In order to make the DS approach configurable, the f llowing changes are required.

1) A table containing the number of iterations required for each operator for the range of target combinations of DWT levels and precision is generated. The entries of this table are determined using the techniques.

2) Shift registers that need to dela by aword (such as the configurable delay elements in Fig. 4) need to be large enough to support the widest possible (which will most likely be the highest level and precision).

3) These shift registers need to be configurable to support different amounts of delays. This is achieved by utilizing a multiplexer, as illustrated in Fig. 5. The multiplexer taps off various stages of the delay chain, effectively serving as a run-time configurable shift register.

VI. SIMULATION RESULT

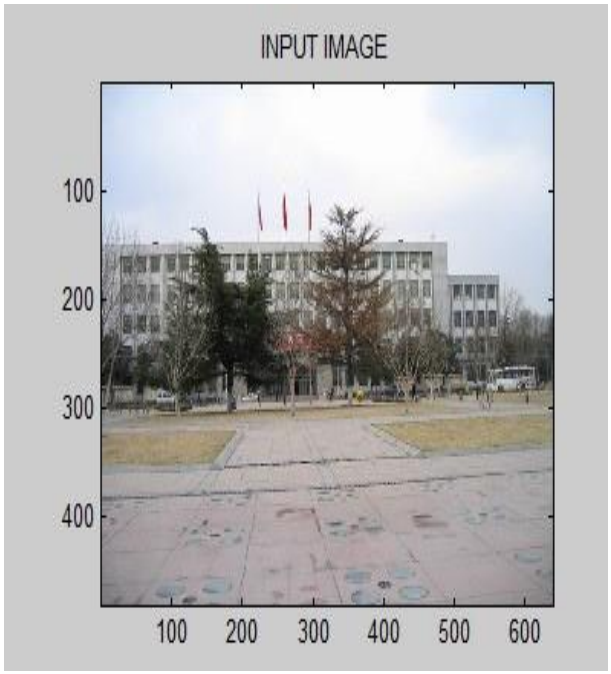


Fig.1. Input Image

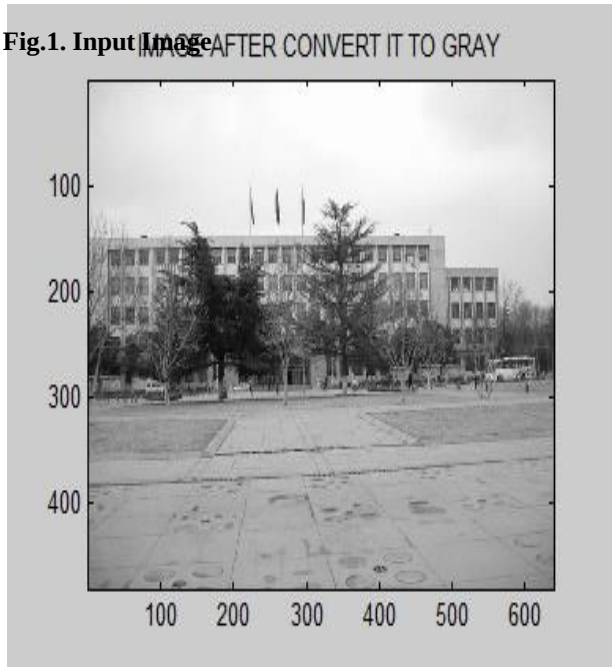


Fig.2. Image after convert it to gray

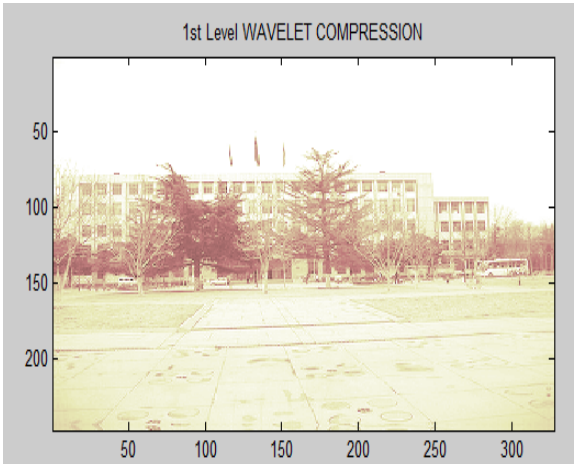


Fig.3. 1st Level wavelet compression

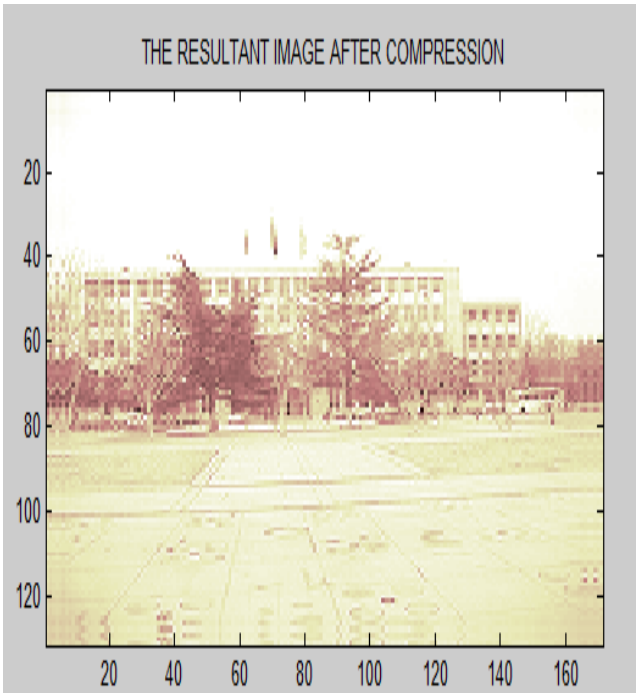


Fig.4. The resultant image after compression

VII. CONCLUSION

In this work, I have completed the image compression using DWT technique to reduce the size of the image without lossing the quality and information of image by JPEG 2000 standard.

REFERENCES:

[1] M. Rabbani and R. Joshi, "An overview of the JPEG 2000 still image compression standard," *Signal Process.: Image Commun., vol. 17, no. 1, pp. 3–48, Jan. 2002.*

[2] C. Huang, P. Tseng, and L. Chen, "Flipping structure: An efficient VLSI architecture for lifting-based discrete wavelet transform," *IEEE Trans. Signal Process., vol. 52, no. 4, pp. 1080–1089, Apr. 2004.*

[3] K. Kotteri, S. Barua, A. Bell, and J. Carletta, "A comparison of hardware implementations of the biorthogonal 9/7 DWT: Convolution versus lifting," *IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 52, no. 5, pp. 256–260, May 2005.*

[4] C. Cheng and K. Parhi, "High-speed VLSI implementation of 2-D discrete wavelet transform," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 393–403, Jan. 2008.

[5] B.Wu and C. Lin, "A high-performance and memory efficient pipeline architecture for the 5/3 and 9/7 discretewavelet transform of JPEG2000 codec," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 12, pp. 1615–1628, Dec. 2005.

[6] C. Xiong, J. Tian, and J. Liu, "Efficient architectures for two-dimensional discrete wavelet transform using lifting scheme," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 607–614, Mar. 2007.

[7] N. Mehrseresht and D. Taubman, "An efficient content-adaptive motion- compensated 3-D DWT with enhanced spatial and temporal scalability," *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1397–1412, Jun. 2006.

[8] S. Barua, K. Kotteri, A. Bell, and J. Carletta, "Optimal quantized lifting coefficients for the 9/7 wavelet," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2004, vol. 5, pp. 193–196.

[9] V. Spiliotopoulos, N. Zervas, Y. Andreopoulos, G. Anagnostopoulos, and C. Goutis, "Quantization effect on VLSI implementations for the 9/7 DWT filters," in

Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 2001, vol. 2, pp. 1197–1200.

[10] K. Kotteri, A. Bell, and J. Carletta, "Design of multiplierless, high performance, wavelet filter banks with image compression applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 3, pp. 483–494, Mar. 2004.

[11] A. Benkrid, K. Benkrid, and D. Crookes, "Optimal wordlength calculation for forward and inverse discrete wavelet transform architectures," *Opt. Eng.*, vol. 43, no. 2, pp. 455–463, Feb. 2004.

[12] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 247–269, May 1998.

[13] T. Acharya and C. Chakrabarti, "A survey on lifting-based discrete wavelet transform architectures," *J. VLSI Signal Process.* vol. 42, no. 3, pp. 321–339, Mar. 2006.

[14] M. Marcellin, M. Lepley, A. Bilgin, T. Flohr, T. Chinen, and J. Kasner, "An overview of quantization in JPEG 2000," *Signal Process.: Image Commun.*, vol. 17, no. 1, pp. 73–84, Jan. 2002.

[15] K. Varma and A. Bell, "JPEG2000—Choices and tradeoffs for encoders," *IEEE Signal Process. Mag.*, vol. 21, no. 6, pp. 70–75, Nov. 2004.[16] M. Weeks, "Precision for 2-D discrete wavelet transform processors," in *Proc. IEEE Workshop Signal Process. Syst.*, 2000, pp. 80–89.

Biography

C.Bhaskar, Assistant professor in ECE Department in Odaiyappa College of Engineering and Technology, Theni, Tamil Nadu, India. He has more than 09 years of teaching experience with expertise in VLSI. He has completed post graduate in VLSI Design.

P.Rajapirian, Assistant professor in ECE Department in Kings College of Engineering, Thanjavur, Tamil Nadu, India. He has more than 09 years of teaching experience with expertise in VLSI. He has completed post graduate in VLSI Design.