Precision-Aware and Quantization of Lifting Based DWT Hardware Architecture

Rekha. N, Dr. K B Shivakumar, M.Z Kurian
M.Tech. Student, Professor, Dept. of TCE, HOD, Dept. of E&C
Sri Siddhartha Institute of Technology, Tumkur.

Abstract- *This paper presents precision-aware approaches and associated hardware implementations for performing the DWT. By implementing BP architecture and also presents DS design methodologies. These methods enable use of an optimal amount of hardware resources in the DWT computation. Experimental measurements of design performance in terms of area, speed, and power for 90-nm complementary metal–oxide semiconductor implementation are presented. Results indicate that BP designs exhibit inherent speed advantages than DS design.*

Keywords - *Fixed point arithmetic, image coding, very large scale integration (VLSI), wavelet transforms*

I. INTRODUCTION

Discrete wavelet transforms (DWT) decomposes image into multiple sub bands of low and high frequency components. Encoding of sub band components leads to compression of image. DWT along with encoding technique represents image information with less number of bits achieving image compression. Image compression finds application in every field such as entertainment, medical, defense, commercial and industrial domains. The core of image compression unit is DWT. Other image processing techniques such as image enhancement, image restoration and image filtering also requires DWT and Inverse DWT for Transformations. DWT-IDWT is one of the prominent transformation techniques that are widely used in signal processing and communication applications. DWT-IDWT computes or transforms signal into multiple resolution sub bands. DWT is computationally very intensive and consumes power due to large number of mathematical operations.

Latency and throughput are other major limitations of DWT as there are multiple levels of hierarchy. DWT was traditionally implemented by convolution. Digit serial or parallel representation of input data further decides the architecture complexity. Such an implementation demands a large number of computations and a large storage that are not desirable for either high-speed or low-power applications. Recently, a lifting-based scheme that often requires fewer computations has been proposed for the DWT. 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.

The JPEG 2000 standard [1] offers considerable coding efficiency and flexibility advantages over the original block DCT-based JPEG standard. A key element of JPEG 2000 is the discrete wavelet transform (DWT), which recursively decomposes an input image into subbands 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.

II. LITERATURE SURVEY

The JPEG committee began to investigate possibilities for a new still image compression [1] standard to serve current and future applications. This was named JPEG2000, this results in compressive techniques. The JPEG2000 system was not only to provide higher compression efficiency compared to the baseline JPEG system. It was to provide a new image representation with a rich set of features. The low-complexity and memory efficient block DCT of JPEG has been replaced by the full-frame discrete wavelet transform (DWT). The DWT provides a multi-resolution image representation and also improve compression efficiency due to good energy compaction.

There are so many literature on the different hardware implementation of the DWT [2]-[6] and novel DWT algorithm those literature paid much less attention to the precision of the DWT computation and here this is specifically considered as a design goal. The relatively few solutions of this problem include the work of C. Huang *et al.* in [2], C. Xiong,

et al. [6], Barua *et al.* in [8], Spiliotopoulos *et al.* in [9]. The work in [2] provide a verity of hardware implementations and minimize the critical path as well as the memory requirement of the lifting based DWT by flipping conventional lifting structures. The work in [6] presents a novel architecture for 1-d and 2-D DWT by using lifting schemes. This is designed to receive an input and generate an output with the low and high - frequency component of original data being available alternately. The work in [8] 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 [9] conducts a similar analysis with the fixed-point data path fixed to 12 bits of integer and 12 bits of fractional precision, which provides sufficient dynamic range to compute a six-level DWT with over 50-dB PSNR.

These works primarily directed to filter coefficients; but here we address simultaneous optimization of not only the coefficient precision but also the internal data paths used in their computation and for that we present a solution that is fully regard to precision allow the design of a DWT to any desired accuracy. By using this approach, we show that the optimization technique can be used to minimize operand bit widths in a bit-parallel (BP) architecture. This enables the implementations with a significant improvement in hardware resources and/or execution time and also make safe that overflows are avoided and precision requirements are met.

This paper organized as follows. Section III presents top level architecture of the DWT design. Section IV gives an overview of the lifting-based DWT and JPEG2000 quantization. Section V describes the proposed design. Section VI Presents DS-DWT architecture. Section VII gives the Results.

III. TOP LEVEL ARCHITECTURE

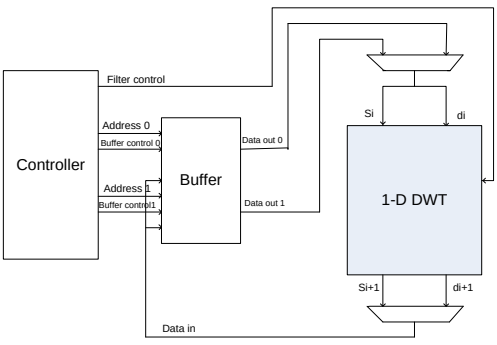

Fig.1**:** Generic high level architecture of the DWT design

Fig.1 shows the generic high level architecture of the DWT design. The 1-D DWT module performs the actual wavelet transform. It can be implemented through a BP. The buffer is dual-port which holds two data frames and is used to store the original raw data, intermediate data, and /or the final transformed data. The controller is used to manage the overall operation of design by generating control signals for the buffer and the filter.

IV. LIFTING BASED DWT

A. *Lifting*

The lifting scheme based DWT has been included in the upcoming JPEG2000 standard because it reduces the arithmetic complexity [8] of the conventional, over the convolution based DWT.

Fig.2 shows the steps for performing a two level DWT on an image. The 1-D DWT perform the actual wavelet transform by first performing rowise operation on the image producing low frequency *L1* and high- frequency *H1* components. After performing 1-D DWT again on the columns of *L1* and *H1*, the first level of decomposition is completed, and *LL1*, *HL1*, *LH1*, and *HH1* are obtained. This process can be repeated on *LL1* to produce the *LL2*, *HL2*, *LH2*, and *HH2* subbands.

The traditionally 9/7 DWT was implemented through convolution-based methods, in which low pass and high pass filters are used. Later Daubenchies and Sweldens showed that DWT can be decomposed into a finite sequence of lifting steps, which provides several advantages like lower computation and memory requirements and easier boundary management [9] over the convolution based .So in this project a lifting based 9/7 DWT is used.

Fig.3 describes the flipping structure of 9/7 DWT by *Haung et.al*. [2] for the lifting-based 1-D DWT. The flipping structure share the same computational complexity with the traditional lifting scheme, this flipping structure reduces the critical path by flipping some computations units with the inverse of multiplier coefficients. Constants *C0……..C5* are given by

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

B. *Quantization*

Quantization involved in image processing is a

lossy compression technique achieved by compressing a range of values to a single quantum value. When the number of discrete symbols in a given stream is reduced, the stream becomes more compressible. For example, reducing the number of colors required to represent a digital image makes it possible to reduce its file size. Specific applications include DCT data quantization in JPEG and DWT data quantization in JPEG 2000

Quantization is a key element for the lossy 9/7 DWT in achievable compression performance. The JPEG 2000 standard supports uniform dead-zone quantization, as well as trellis coded quantization [8]. In this paper Uniform dead-zone quantization is chosen due to its simplicity and hardware efficiency

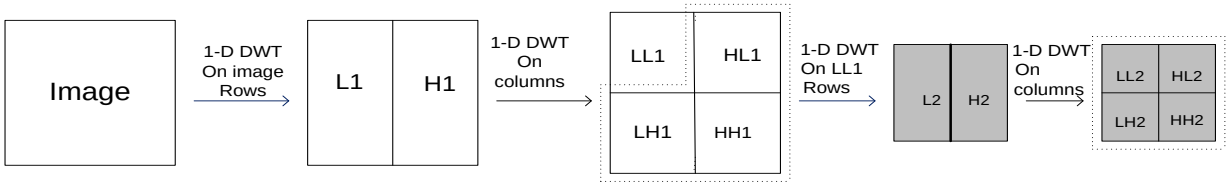

Fig.2: Two level wavelet decomposition. The dotted portions are the final transformed data

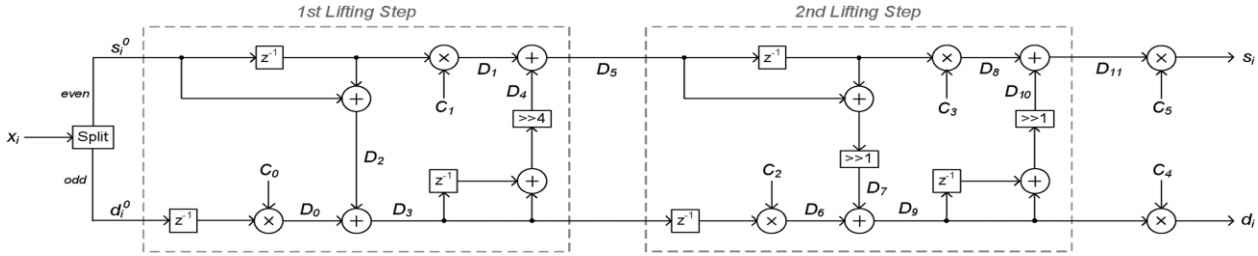

Fig.3: Bit parallel Architecture

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. For example, using this quantization scheme, with reference to Fig. 2 if *HL1, LH1* and *HH1* have a precision of *n* bits, then *LL2, HL2, LH2* and *HH2* should have a precision of *n* + 1 bit [7].

V. PRAPOSED ARCHITECTURE

It is desirable to minimize the bit-widths for all variables in the data paths, leading to size reductions in tables, and operators such as adders and multipliers. We use a bit width minimization scheme which minimizes bit-widths while ensuring that the results meet the one ulp error bound requirement. We split the problem of minimizing fixed-point bit-widths into two parts: range analysis followed by precision analysis. The two parts are performed entirely within MATLAB framework, making use of the finite precision hardware emulation models a numerical approach is taken to tackle the range and precision minimization problems Range analysis involves inspecting the dynamic range and working out the bit-widths of the integer parts. Using insufficient bits for the range can cause overflows or underflows and excessive bits waste valuable hardware resources. The range analysis method uses a simulation-based approach, where each input of the design is supplied with a large set of random numbers, which ranges over the interval of possible values for the particular input, including the extreme values of that interval.

In BP approach computing speed is the primary goal, the design challenge is determining the appropriate number of integer and fractional bits to use in representing all the signals utilized during the computation.. The number of integer bits, fractional bits and the total number of bits of signal z are denoted by IBz, FBz and Bz respectively, where Bz = IBz + FBz.

A. Integer Bit- Width Determination

The IB can be determined by using an approach, 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. Hence, for the 1-D DWT shown in Fig. 3, the following signal pairs need to share the same IB, i.e., (*D0 ,D2*), (*D1, D4*), (*D6, D7*) and (*D8, D10*).Practically, this implies that the IB should be set to the larger IB of the two, e.g. IBD0 = IBD2 = (IBD0 , IBD2). Furthermore si, di are the final output data.

B. Fractional Bit-Width Optimization

The fractional bit-width optimization is executed in two steps, one is static step based on analytical models to obtain the set of widths, and a dynamic step based on simulation that further

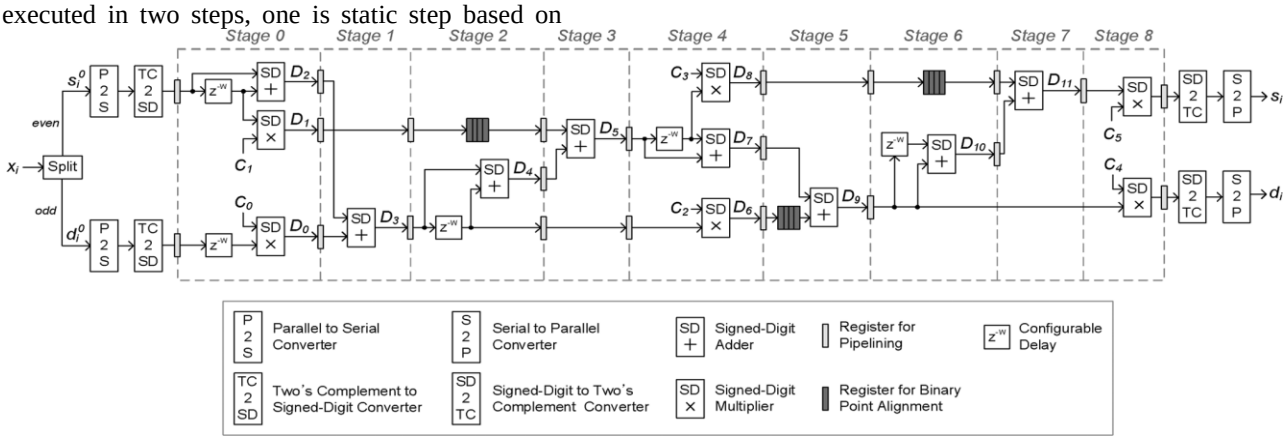

Fig.4: DS 1-D 9/7 DWT

Reduce the bit width using a PSNR delta threshold. The target precision metric (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.

1. *Static Optimization:* The worst case (maximum absolute
error) quantization errors for truncation and round-to-nearest are given by

Truncation: $Ez = \max (0, 2^{-FBz} - 2^{FBz'})$ …… (1)

Round-to-nearest: $Ez = \{0, \text{ if } FBz \geq FBz'\}$ …… (2)

2^{-FBz-1}, otherwise

Where FBz' is the full precision of unquantized

$Es_i = \max (D_{11}) \times 2^{-FBc5-1} + C5 \times E_{D11} + E_{D11} \times 2^{-FB_{C5}-1}$ … (3)

$+ \max (0, 2^{-FB_{si}} - 2^{-FBc5-FB_{D11}})$

$Edi = \max (D9) \times 2^{-FB_{C4}-1} + C4 \times E_{D9} + E_{D9} \times 2^{-FB_{C4}-1}$ …(4)

$+ \max (0, 2^{-FBdi} - 2^{-FBc4-FB_{D9}})$

These error expressions consider the worst case error bounds at each node and can be recursively derived for any number of DWT levels. The bit widths of the internal data paths are found using the error expressions in conjunction with simulated annealing. Since the quantization scheme of JPEG 2000 uses increasing precision with $i=L$

VI. DS DWT DESIGN

Fig. 4 shows the DS 1-D 9/7 DWT design. 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 used for the static step is the unit in the last place.

The DS representations use a radix-2 SD redundant number system. Due to redundancy, SD operations do not propagate carries and hence they are able to run in most significant digit first (MSDF) mode. This MSDF property makes it attractive for the DS DWT approach since it allows for varying the number of iterations to obtain different precision

VII. IMPLEMENTATION RESULTS

The codes are written in verilog language as it is easier than any other HDL language and because of some of its salient features like it allows the descriptions of each module to done mathematically in terms its terminals and external parameters applied to the module.etc. The same has

been simulated using the able simulation tool modelsim 6.2. The results in terms of numbers and waveforms are analyzed to get accurate results. One sample window showing simulation results for compression is shown below in figure 6. Table 1 gives the implemented results.

Designs are synthesized using Synopsys Design Compiler with the standard-cell library of Infineon 90-nm CMOS technology. Power results are obtained via Synopsys Power Compiler, which performs gate-level power simulation using user supplied data (images). Supplied data (images).

TABLE. I

COMPARISONS BETWEEN BP AND DS IMPLEMENTATIONS

Approach	Area	Clock speed	Processing Time	Dynamic power	Static power
Bit-parallel	35228	56	7.4	6.8	3.5
Digit-Serial	18680	435	29,8	23.5	1.7

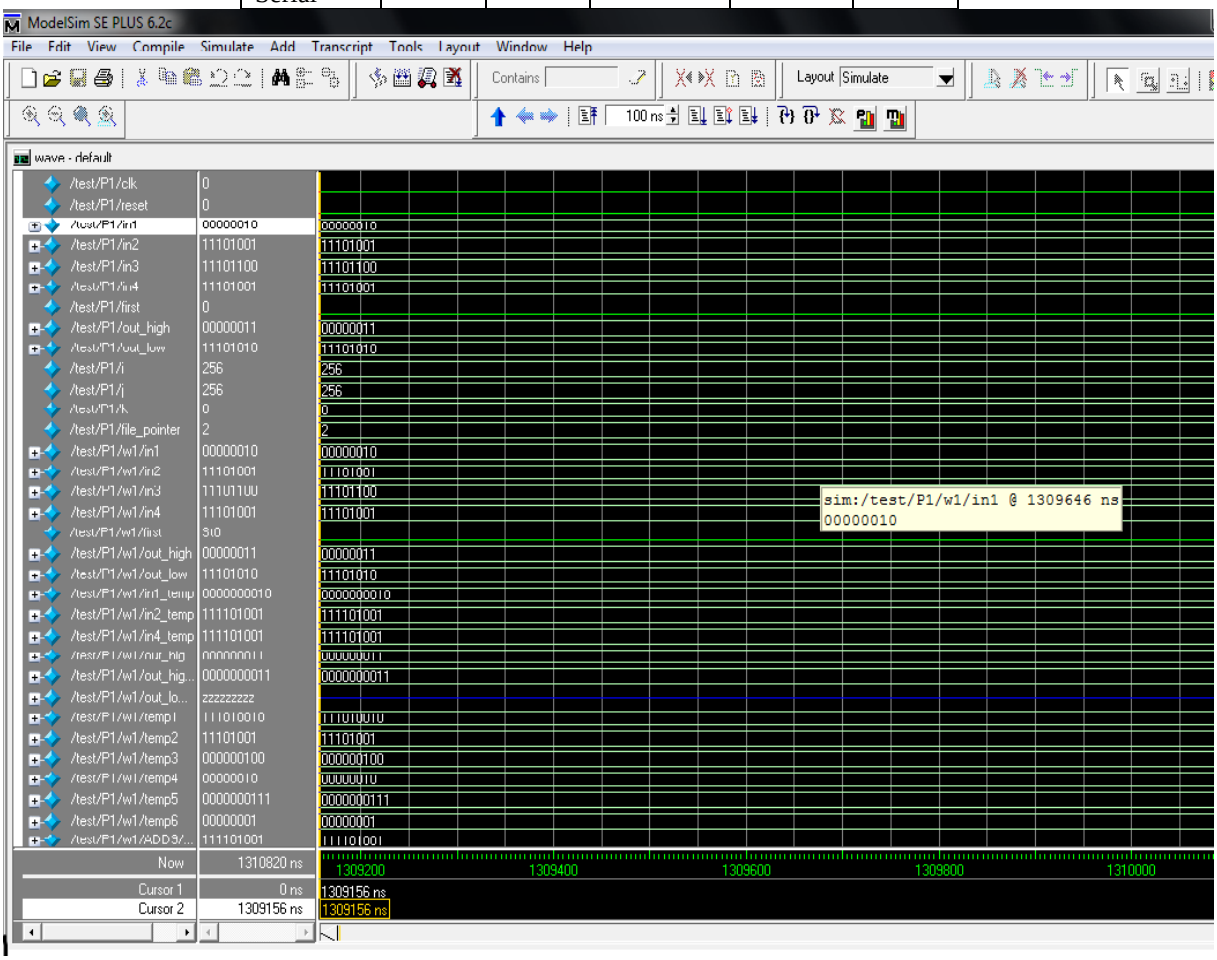

Fig.6: Simulation result

VIII. CONCLUSION

In this work precision-aware approaches and associated hardware implementations for performing the DWT are presented. Both BP and DS design methodologies and results have been presented. These methods enable use of an optimal amount of hardware resources in the DWT computation. Moreover, this framework enables quantization, which is traditionally performed after the DWT in algorithms such as JPEG 2000. We examined the energy and power tradeoffs between the associated BP and DS designs, in particular, highlighting the differing respective roles of static and dynamic power in each. We believe that design methods and architectures such as those presented here play a significant role in the design of future energy- and precision-optimized DWT implementations.

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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] M. Weeks, "Precision for 2-D discrete wavelet transform processors," in *Proc. IEEE Workshop Signal Process. Syst.*, 2000, pp. 80–89.

[8] 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.

[9] 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.

[10] Dong-U Lee, *Member, IEEE*, Lok-Won Kim, *Student Member, IEEE*, and John D. Villasenor, *Senior* Member "precision aware self quantizing hardware architecture for DWT", *in proc IEEE Trans. Image processing* , VOL. 21, NO. 2, February 2012.