Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2007, Article ID 13754, 5 pages
doi:10.1155/2007/13754

Research Article

Fixed Wordsize Implementation of Lifting Schemes

Tanja Karp

Department of Electrical and Computer Engineering, College of Engineering, Texas Tech University, P.O. Box 43102, Lubbock, TX 79409-3102, USA

Received 16 December 2005; Revised 29 May 2006; Accepted 26 August 2006

Recommended by Soontorn Oraintara

We present a reversible nonlinear discrete wavelet transform with predefined fixed wordsize based on lifting schemes. Restricting the dynamic range of the wavelet domain coefficients due to a fixed wordsize may result in overflow. We show how this overflow has to be handled in order to maintain reversibility of the transform. We also perform an analysis on how large a wordsize of the wavelet coefficients is needed to perform optimal lossless and lossy compressions of images. The scheme is advantageous to well-known integer-to-integer transforms since the wordsize of adders and multipliers can be predefined and does not increase steadily. This also results in significant gains in hardware implementations.

1. INTRODUCTION

Lifting schemes have been introduced by Daubechies and Sweldens as a structure to design and implement the discrete wavelet transform (DWT) [1]. They have gained immense popularity, since they provide an elegant means to maintain perfect reconstruction of the DWT while staying in the field of integer numbers [2], thus allowing for lossless coding. At each lifting step, a quantizer that ensures that intermediate results are rounded to integer numbers is introduced.

Lifting schemes have been applied to a variety of fields. They are used to implement the polyphase filters of modulated filter banks [3], are building blocks of integer-to-integer transforms such as the IntFFT [4], the IntDCT [5], and YCoCg-R color space transform [6]. More recently, the concept of lifting has been extended to multidimensional inputs [7]. Lifting schemes can even be designed in such a way that they preserve signal symmetries, a fact that is important for image compression. The performance of nonlinear reversible DWT using lifting schemes has been studied in [8].

While keeping the wordlength of the wavelet coefficients finite is a necessary condition for lossless coding, the dynamic of the signal and thus its wordlength normally increase. This is not only disadvantageous from a compression point of view, but also means a higher complexity of the circuitry. For example, in an FPGA, the number of gates needed for a multiplier increases significantly with the wordlength. It is therefore of interest to keep the maximum wordsize limited, thus resulting in a fixed-point implementation of the lifting scheme, where, in addition to rounding, overflow occurs if a value exceeds the range of representable numbers.

In this paper, we show that lifting schemes are also robust towards overflow if it is handled as wrap-around and we evaluate the performance of fixed wordsize lifting schemes.

2. FIXED WORDSIZE LIFTING

A typical lifting step and its inverse can be described in a matrix form as

$$\begin{bmatrix} \mathcal{Z}\{y_0(n)\} \\ \mathcal{Z}\{y_1(n)\} \end{bmatrix} = \begin{bmatrix} 1 & A(z) \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathcal{Z}\{x_0(n)\} \\ \mathcal{Z}\{x_1(n)\} \end{bmatrix},$$
$$\begin{bmatrix} \mathcal{Z}\{\hat{x}_0(n)\} \\ \mathcal{Z}\{\hat{x}_1(n)\} \end{bmatrix} = \begin{bmatrix} 1 & -A(z) \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathcal{Z}\{y_0(n)\} \\ \mathcal{Z}\{y_1(n)\} \end{bmatrix}, \quad (1)$$

where $\mathcal{Z}\{x(n)\}$ denotes the z transform of a sequence $x(n)$ and $A(z)$ is an FIR filter. For integer-to-integer lifting, the output sequence of the filter $A(z)$ is rounded to be an integer. For a fixed wordsize implementation, we additionally need to take care of overflow, in case the result exceeds the dynamic range defined by the wordsize. Overflow can occur at each multiplication and each addition during the lifting step. For reasons of simplicity, let us first look at the case where the filter $A(z)$ simplifies to a constant, that is, $A(z) = a$. The rounding and overflow operations for the lifting and inverse lifting steps are shown in Figure 1. Overflow needs to be handled at

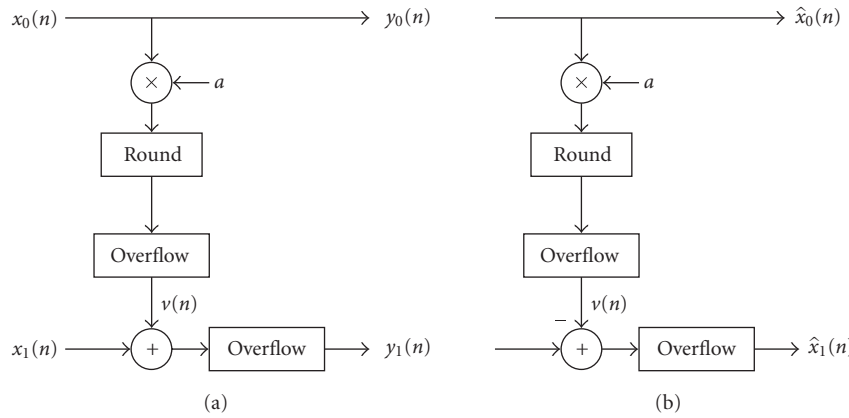


Figure 1: Fixed-point implementation of a lifting step and its inverse.

the resultant of the multiplication as well as at the resultant of the adder.

We assume that $x_0(n)$, $x_1(n)$ are finite resolution numbers representable with a fixed wordsize, and a is an arbitrary real number. It can be easily seen from Figure 1 that $\hat{x}_0(n) = x_0(n)$ always holds true if the same fixed-point number format is chosen for lifting and inverse lifting. The multiplication $a \cdot x_0(n)$ is implemented at both the lifting step and the inverse lifting step since $y_0 = x_0$, resulting in the same sequence $v(n)$ at the lifting and inverse lifting steps as long as the same rounding rules are applied and overflow is handled in the same way (saturation or wrap-around) for both schemes. If no overflow occurs during the addition $y_1(n) = x_1(n) + v(n)$, then no overflow occurs during the subtraction at the inverse lifting step and we obtain $\hat{x}_1(n) = y_1(n) - v(n) = x_1(n)$.

For overflow occurring during the addition $v(n) + x_1(n)$ and the subtraction $y_1(n) - v(n)$, the overflow only cancels, if it is treated as wrap-around and not as saturation. In the following, we assume that we can represent numbers in the range from $x_{\min}$ to $x_{\max}$, including these two margins.

In saturation, numbers that are larger than $x_{\max}$ or smaller than $x_{\min}$ are mapped to $x_{\max}$ or $x_{\min}$, respectively. Assuming that overflow occurs at a certain time index n_0 in such a way that $x_1(n_0) + v(n_0) > x_{\max}$ yields

$$y_1(n_0) = \text{saturate}\,(x_1(n_0) + v(n_0)) = x_{\max}, \tag{2}$$

$$\hat{x}_1(n_0) = y_1(n_0) - v(n_0) = x_{\max} - v(n_0) \neq x_1(n_0). \tag{3}$$

Note that the result of $y_1(n_0) - v(n_0)$ always lies within the range of representable numbers, and therefore no overflow occurs in (3). A similar case can be made for $x_1(n_0) + v(n_0) < x_{\min}$.

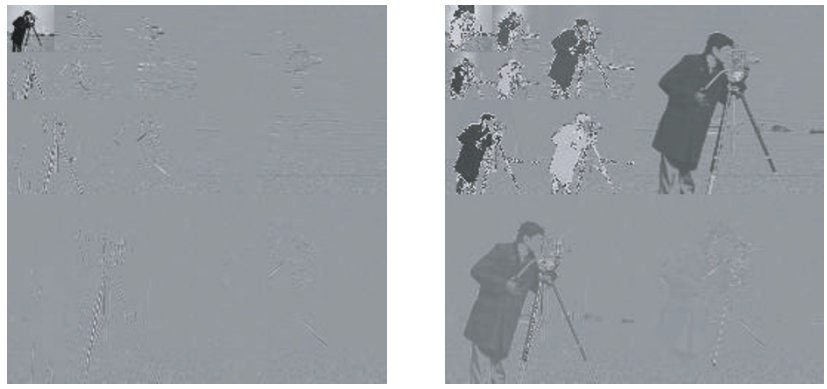
In wrap-around, overflow is handled in such a way that the amount by which a number exceeds $x_{\max}$ will be added to $x_{\min}$ and the amount by which a number is smaller than $x_{\min}$ will be subtracted from $x_{\max}$. Note that wrap-around is very similar to modulus operation and results in $\text{wrap}(x + k(x_{\max} - x_{\min})) = x$, if k is an integer value. While overflow error is larger in wrap-around than in saturation, it has the advantage that the result obtained after adding a number and then subtracting the same number is the original value even if overflow affected the intermediate result. Thus, if $x_1(n_0) + v(n_0) > x_{\max}$, we obtain

$$\begin{aligned} y_1(n_0) &= \text{wrap}\,(x_1(n_0) + v(n_0)) \\ &= x_{\min} + (x_1(n_0) + v(n_0) - x_{\max}) \\ &= x_1(n_0) + v(n_0) - (x_{\max} - x_{\min}), \\ \hat{x}_1(n_0) &= \text{wrap}\,(y_1(n_0) - v(n_0)) \\ &= \text{wrap}\,(x_1(n_0) + v(n_0) - (x_{\max} - x_{\min}) - v(n_0)) \\ &= \text{wrap}\,(x_1(n_0) - (x_{\max} - x_{\min})) = x_1(n_0), \end{aligned} \tag{4}$$

and if $x_1(n_0) + v(n_0) < x_{\min}$, we get

$$\begin{aligned} y_1(n_0) &= \text{wrap}\,(x_1(n_0) + v(n_0)) \\ &= x_{\max} - (x_{\min} - (x_1(n_0) + v(n_0))) \\ &= x_1(n_0) + v(n_0) + (x_{\max} - x_{\min}), \\ \hat{x}_1(n_0) &= \text{wrap}\,(y_1(n_0) - v(n_0)) \\ &= \text{wrap}\,(x_1(n_0) + v(n_0) + (x_{\max} - x_{\min}) - v(n_0)) \\ &= \text{wrap}\,(x_1(n_0) + (x_{\max} - x_{\min})) = x_1(n_0). \end{aligned} \tag{5}$$

From the upper equations, we see that the overflow error introduced at the lifting step is canceled at the inverse lifting step. If we now return to the general form of lifting with $A(z)$ being a filter, we realize that all we have to ensure for overflow errors to cancel is that the filter output signal $v(n)$ is the same for lifting and inverse lifting. Since both filters and their input signals are identical, this means that we have to implement the convolution in an identical way at the lifting and the inverse lifting steps, that is, applying the same wordsize, using the same rounding algorithms, and treating overflow in the same way. Note that for the overflow handling blocks which calculate $v(n)$, it does not matter whether we choose saturation or wrap-around in the case of overflow, as long as we do the same for lifting and inverse lifting.



(a) (b)

FIGURE 2: Fixed-point approximation and detail coefficients in the wavelet domain (3-level decomposition) with wordsize of (a) 13 bits and (b) 9 bits.

3. PERFORMANCE ANALYSIS

To evaluate the performance of the fixed-point lifting scheme, we implemented Daubechies' 9–7 wavelet using the lifting scheme described in [1]. Each lifting step is implemented according to Figure 1, where instead of a constant a we have first-order filters. The dual lifting step is implemented accordingly. We then apply three levels of the fixed-point wavelet transform to the 256×256 pixel 8-bit greyscale image known as "cameraman" using different fixed-point formats and wordsizes. Image boundaries are treated using symmetric extension [9]. Since the final scaling factor in the lifting decomposition of the 9–7 wavelet does not pertain symmetry of the approximation and detail coefficients if it is implemented as a 4-step lifting [1], it has been omitted in our implementation. Reference [9] explains how it could be implemented using a per-lifting-step symmetric extension method.

In a first step, the input image is scaled such that all pixel values lie within the range $(-1, 1)$, and thus can be represented by a sign bit and 7 bits describing the values after the binary point. All lifting coefficients and intermediate signals are realized using an identical fixed-point format. For our analysis, we consider wordsizes with 7 bits after the binary point (same as input) and 9 bits after the binary point to allow for a more accurate representation of intermediate results. The number of bits before the binary point is fixed for each implementation. Depending on the setup, it varies from 1 (sign bit) to 6, allowing for different dynamic ranges for intermediate signals as well as the approximation and detail coefficients. Overflow at the lifting step addition is treated as wrap-around to ensure cancellation of overflow errors at the reconstruction. Overflow within the lifting step is treated either as wrap-around or saturation.

Figure 2 shows the wavelet coefficients of the 3-level decomposition if 7 bits are used after the binary point, saturation is applied for overflow within a lifting step, and wrap-around is used at the adders. For Figure 2(a), we allow 6 bits in front of the binary point (i.e., a total wordsize of 13 bits) and for Figure 2(b), we only allow 2 bits before the binary point (i.e., a total wordsize of 9 bits).

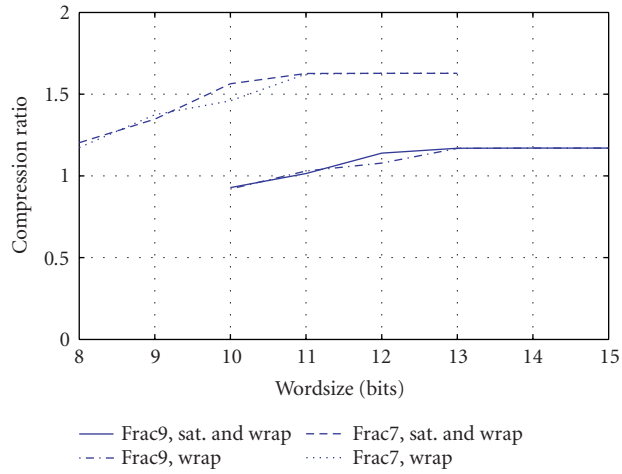


FIGURE 3: Lossless compression using SPIHT. Wavelet coefficients have 7 bits after the binary point (frac7) or 9 bits (frac9); the number of bits before the binary points varies.

While Figure 2(a) looks very similar to what we are used to see for linear, integer-to-floating-point DWT decomposition, namely horizontal, vertical, and diagonal edge information in the detail bands and a scaled copy of the original image in the approximation band, Figure 2(b) shows large coefficients in the detail bands. In fact, most of the detail bands look like an approximation of the original image. This can be explained by the fact that handling overflow as wrap-around turns a large positive number into a small negative number, a nonlinear operation that produces new frequencies, resulting in the large number of high-frequency components seen in the detail bands of Figure 2(b).

To examine the suitability of our fixed-point lifting scheme for lossless compression, we have losslessly encoded the fixed-point wavelet coefficients using SPIHT encoder [10]. Figure 3 shows the obtained compression ratios. As

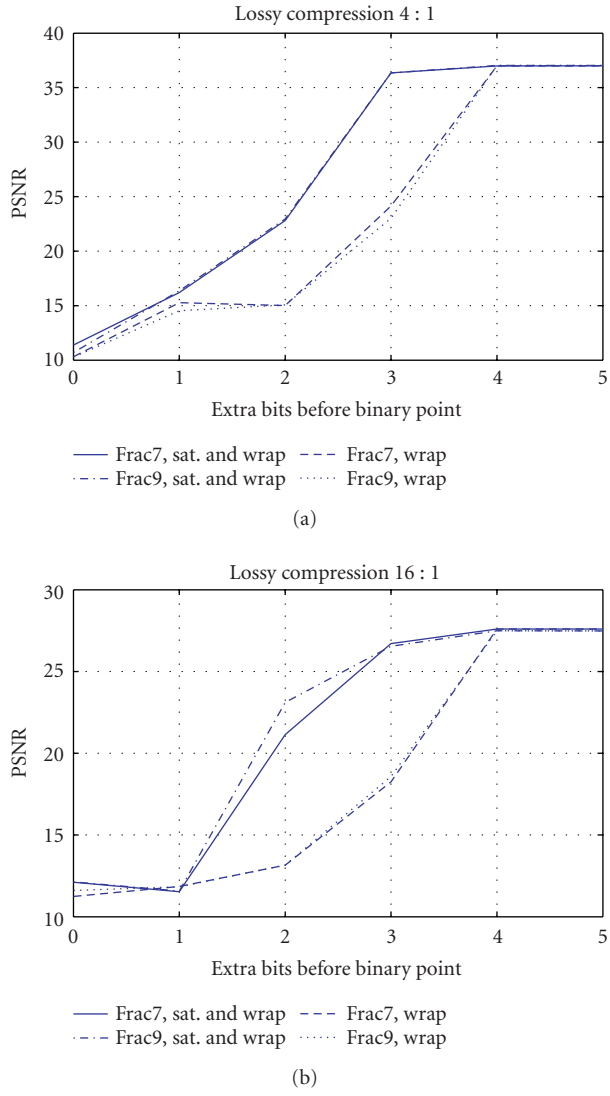


FIGURE 4: Lossy coding using SPIHT. Wavelet coefficients have 7 bits after the binary point (frac7) or 9 bits (frac9); the number of bits before the binary points varies.

expected, the scheme with 7 bits after the binary point achieves a higher compression ratio as the one with 9 bits, since SPIHT can stop encoding 2-bit levels earlier. However, the schemes with the shorter wordlength result in a poorer compression ratio, mainly because of the large number of significant wavelet coefficients in the detail bands. We can observe from Figure 3 that we need a wordsize that is about 3 to 4 bits larger than the 8-bit input wordsize to obtain the highest compression ratio. Having a larger wordsize than that does not improve the performance any further but means increased circuit complexity, since adders and multipliers of larger wordsize have to be used.

Also, we see from Figure 3 that the fixed-point implementation that uses saturation for the lifting filter and wrap-around for the addition outperforms the one that uses wrap-around at both places. This is because saturation introduces the lower overflow error of both schemes.

Figure 4 shows the PSNRs obtained when performing lossy compression of ratios 4 : 1 and 16 : 1 using SPIHT for the fixed-point wavelet coefficients. We notice that the PSNR significantly reduces if less than 4 extra bits are spent before the binary point to cover the increased dynamic of the signal. Again, for short wordsizes, the SPIHT has to spend many bits to encode the large number of significant detail coefficients. For the range of 1 to 3 extra bits, the fixed-point scheme with saturation for overflow handling at the lifting filter again outperforms the one that used wrap-around. Also, we see that the increased precision of the intermediate signals, lifting filter coefficients, and wavelet coefficients maintained in the scheme with 9 bits after the binary point shows only minimal improvement over the scheme with 7 bits, and thus does not justify the increased circuitry since it also has a lower compression ratio in the lossless case. Note that the low compression ratio obtained is due to the small image size.

4. APPROPRIATE CHOICE OF WORDSIZE

The results from the previous section have shown that the performance of the compression scheme highly depends on the wordsize chosen. From Figure 2, we have seen that overflow that happens when calculating the approximation coefficients results in a significant change of the detail coefficients and a reduced ability of SPIHT to compress the image effectively. Since most large wavelet coefficients are in the approximation band, which shows a reduced-size thumbnail of the original image, we have to choose the wordsize such that overflow only happens rarely when calculating these coefficients. For Daubechies' 9–7 wavelet, the lowpass filter amplifies the amplitude by $\sqrt{2}$, resulting in a factor of 2 per 2D decomposition step, and thus 1 additional bit being required per level of decomposition. Since we did not implement the scaling factor of 1.1496 in the lifting decomposition [1], the gain is actually slightly lower. Figures 3 and 4 confirm that we obtain close-to-optimal performance in terms of compression ratio for lossless compression and in terms of PSNR for lossy compression with 3 additional bits before the binary point if we use saturation for overflow within a lifting step. However, from the difference in PSNR at 3 additional bits when using wrap-around or saturation within a lifting step, we conclude that overflow still happens during the calculations since otherwise both results should be identical. Only at 4 additional bits the results for wrap-around and saturation do converge. Thus, in addition to the general scaling of the approximation coefficients, one needs to take intermediate results into account, where overflow can occur even if the final wavelet coefficient lies within the range of realizable numbers. For the lifting implementation of Daubechies' 9–7 wavelet [1], the most critical step in the decomposition is the first lifting step, where neighboring even-indexed pixels are added and then scaled by a factor of -1.5861, thus resulting in values that have a magnitude that is 3.1723 as large as the incoming values if those belong to a smooth region of the image and are identical. Thus, the extra bit (4 additional bits before the binary point instead of 3) avoids major occurrence of overflow at this step. The other lifting steps of

our implementation [1] have scaling factors with magnitudes less than one and are thus of no concern.

5. CONCLUSION

In this paper, we have presented a fixed wordsize implementation of the lifting scheme that maintains perfect reconstruction even in the case of overflow occurrences. It provides a nonlinear, reversible, integer-to-integer discrete wavelet transform with predefined dynamic range of the wavelet coefficients. However, when limiting the dynamic range of the wavelet coefficients too much, a large number of significant detail coefficients occur due to overflow errors and make the scheme less suitable for lossy and even lossless compression using standard wavelet encoding schemes. However, as long as the wordlength is chosen such that overflow occurs only rarely, the scheme provides the advantage that limited wordsize adders and multipliers can be used, which is of particular advantage for FPGA implementations. We have derived how the increase in wordsize depends on the number of levels of wavelet transform that are performed as well as the values of the lifting coefficients.

REFERENCES

[1] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Journal of Fourier Analysis and Applications*, vol. 4, no. 3, pp. 247–269, 1998.

[2] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Applied and Computational Harmonic Analysis*, vol. 5, no. 3, pp. 332–369, 1998.

[3] A. Mertins and T. Karp, "Modulated, perfect reconstruction filterbanks with integer coefficients," *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1398–1408, 2002.

[4] S. Oraintara, Y.-J. Chen, and T. Q. Nguyen, "Integer fast Fourier transform," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 607–618, 2002.

[5] J. Liang and T. D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 3032–3044, 2001.

[6] H. Malvar and G. Sullivan, "YCoCg-R: a color space with RGB reversibility and low dynamic range," ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, July 2003.

[7] R. Geiger, Y. Yokotani, and G. Schuller, "Improved integer transforms for lossless audio coding," in *Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers (ACSSC '03)*, vol. 2, pp. 2119–2123, Pacific Grove, Calif, USA, November 2003.

[8] J. Reichel, G. Menegaz, M. J. Nadenau, and M. Kunt, "Integer wavelet transform for embedded lossy to lossless image compression," *IEEE Transactions on Image Processing*, vol. 10, no. 3, pp. 383–392, 2001.

[9] M. D. Adams and R. K. Ward, "Symmetric-extension-compatible reversible integer-to-integer wavelet transforms," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2624–2636, 2003.

[10] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, 1996.

Tanja Karp received the Dipl.-Ing. degree in electrical engineering (M.S.E.E.) and the Dr.-Ing. degree (Ph.D.) from Hamburg University of Technology, Hamburg, Germany, in 1993 and 1997, respectively. In 1995 and 1996, she spent two months as a Visiting Researcher at the Signal Processing Department of ENST, Paris, France, and at the Mutirate Signal Processing Group, University of Wisconsin at Madison, respectively, working on modulated filter banks. In 1997, she joined the Institute of Computer Engineering at Mannheim University, Germany, as a Senior Research and Teaching Associate. From 1998 to 1999, she has also taught as a Guest Lecturer at the Institute for Microsystem Technology at Freiburg University, Germany. From 2000 to 2006, she was an Assistant Professor in the Department of Electrical and Computer Engineering at Texas Tech University, Lubbock, Texas. She is now an Associate Professor in the same department. Her research interests include multirate signal processing, filter banks, audio coding, multicarrier modulation, and signal processing for communications. She is an IEEE Member and regularly reviews articles for several IEEE and EURASIP transactions.