J Sign Process Syst (2010) 59:245–253
DOI 10.1007/s11265-009-0375-y

An Efficient Pipeline Architecture and Memory Bit-Width Analysis for Discrete Wavelet Transform of the 9/7 Filter for JPEG 2000

Chung-Fu Lin · Pei-Kung Huang · Bing-Fei Wu

Received: 29 August 2006 / Accepted: 6 May 2009 / Published online: 20 June 2009

Abstract In this paper, we propose an efficient pipeline architecture for the DWT 9/7 filter defined in JPEG 2000. The proposed architecture is composed of column and row processors to perform the separable 2-D DWT. Based on the rescheduling DWT algorithm, we derive a new data flow graph to shorten the critical path. The proposed 1-D column processor requires less pipeline registers to achieve about the same critical path compared with other lifting-based architectures. For the row processor, the data dependency of each lifting step is reduced to only two computation nodes and therefore more pipeline registers can be applied to achieve higher processing speed without increasing the internal memory size in the 2-D case. That is, for an N×N image, it only requires 4N internal memory to perform the row-wise transform. For the memory bit-width analysis, we use software simulation to reduce the memory bit-width for various compression ratios. Since a portion of information from least significant bits of DWT coefficients would be discarded after EBCOT-tier2 processing, one can decrease the data width of internal memory to perform various compression ratios of JPEG 2000 coding, especially at the low-bit rates. Our simulation results suggest that it is practically possible to design the energy-aware memory architecture to further reduce the power consumption in the future work.

Keywords JPEG 2000 · DWT · Critical path · Memory bit-width · Energy-aware architecture

1 Introduction

JPEG 2000 is a new still image compression standard, which adopts Discrete Wavelet Transform (DWT) as the transform kernel [1]. This is because wavelet transform can decompose the raw image into different sub-bands with both spatial and frequency information to achieve high compression ratios. As for the hardware implementation, the high throughput, low memory, and efficient power consumption are the critical issues for many multimedia applications.

Recently, a less computation lifting-based DWT has been proposed by Daubenchies and Sweldens [2]. However, the long critical path and internal memory size are still the critical points for the 2-D DWT implementation. In general, using pipeline architectures can increase the processing speed of 1-D column processor, but more pipeline registers also increase the internal memory size of row processor for 2-D DWT [3–6]. In this paper, we

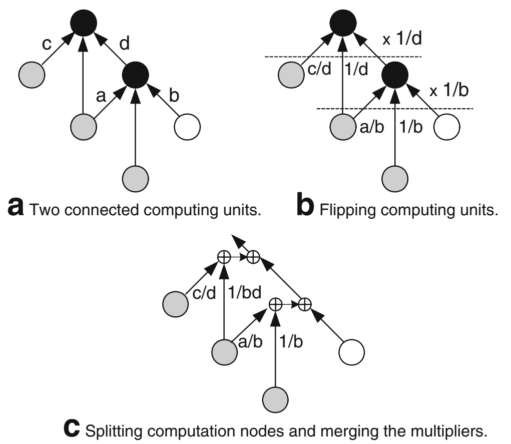
This research was supported by National Science Council under Grant no. NSC 97-2221-E-009-072.

C.-F. Lin (✉)
Core Technology Development Division,
Faraday Technology Corporation,
No. 5, Li-Hsin Rd. III, Hsinchu Science Park,
Hsinchu City, Taiwan 300, R.O.C.
e-mail: cflin.ece90g@nctu.edu.tw

P.-K. Huang · B.-F. Wu
Department of Electrical and Control Engineering,
Chaotic Systems and Signal Processing Lab, CSSP Lab.,
National Chiao Tung University,
1001 Ta Hsheh Road,
Hsinchu 30050, Taiwan

P.-K. Huang
e-mail: nowki.ece92g@nctu.edu.tw

B.-F. Wu
e-mail: bwu@cc.nctu.edu.tw



a Two connected computing units. **b** Flipping computing units.

c Splitting computation nodes and merging the multipliers.

Figure 1 Flipping structure of the computation nodes (i.e. Fig. 7 in [5]).

explore a new data flow graph (DFG) for the rescheduling DWT algorithm [4] to shorten the critical path with less pipeline registers. It only requires 10 registers to achieve one multiplier and two full-adders delay for the 1-D column processor (i.e. T_M+2$T_{1\text{-bit FA}}$). Moreover, the data dependency of 2-D data path for one lifting step is reduced to only two computation nodes. Thus, the tradeoff between the number of pipeline registers and memory size can be eased [7]. That is, for an N×N image, it only requires internal memory with 4N size to perform row-wise transform of the 9/7 filter and more pipeline registers can be even applied to achieve higher processing speed without increasing the internal memory size. Besides, since the 9/7 filter is commonly used for the lossy compression, a portion of compressed data would be discarded after the rate-distortion optimization (RDO), especially that from the low bit-planes [8, 9]. For the power issue, the large internal memory would dominate the power consumption of overall 2-D DWT implementation [10]. Thus, we use software simulation to evaluate the image quality by reducing the bit-width of internal memory. From the experimental results, the data width of internal memory can be properly decreased to make efficient power consumption for different JPEG 2000 compression ratios.

The paper is organized as follows. Section 2 briefly introduces previous architectures of the lifting-based DWT for the 9/7 filter. In Section 3, we explore the data flow graph and propose the one-level 2-D DWT architecture. Then, the comparisons of 1-D and 2-D architectures are addressed. In Section 4, the bit-width of internal memory is simulated for various JPEG 2000 compression ratios to reduce the power consumption. Finally, a brief summary is given in Section 5.

2 Prior Work for Lifting-Based DWT Implementation

The lifting-based DWT requires less computation and lower memory compared with the convolution-based DWT [2]. In general, the pipeline architecture is used to shorten the long computation path and achieve the high processing speed. However, the number of pipeline registers for 1-D DWT would dominate the internal memory size of 2-D architecture. Thus, it is reasonable to reduce the pipeline registers for the 1-D implementation. In the following, we briefly introduce several architectures for the lifting-based DWT [3–6].

The primitive lifting process steps for the 9/7 filter defined in JPEG2000 [1] are described from Eq. (1) to Eq. (8). First, the input sequences x_i are split into even and odd parts, s_i^0 and d_i^0. The two split sequences then perform two lifting steps respectively and the outputs are denoted as s_i^n and d_i^n (n=1, 2). Finally, through the scaling factors K_2 and K_1, we can obtain the low-pass and high-pass wavelet coefficients s_i and d_i.

1. Splitting Step:

$$d_i^0 = x_{2i+1}; \tag{1}$$

$$s_i^0 = x_{2i}; \tag{2}$$

2. Two Lifting Steps:

$$d_i^1 = d_i^0 + \alpha \times \left(s_i^0 + s_{i+1}^0\right); \text{ (Predictor)} \tag{3}$$

$$s_i^1 = s_i^0 + \beta \times \left(d_{i-1}^0 + d_i^1\right); \text{ (Updater)} \tag{4}$$

$$d_i^2 = d_i^1 + \gamma \times \left(s_i^0 + s_{i+1}^1\right); \text{ (Predictor)} \tag{5}$$

$$s_i^2 = s_i^1 + \delta \times \left(d_{i-1}^2 + d_i^2\right); \text{ (Updater)} \tag{6}$$

Table 1 Several architectures for 1-D column processors of the 9/7 filter (T_M: a multiplier delay, T_A: an adder delay).

Architecture	Multi-plier	Adder	Register	Critical path	Bandwidth
Direct + 4 stages [3]	4	8	16	T_M+2T_A	2 I/O
Direct + 32 stages	4	8	32	T_M	2 I/O
Rescheduling [4]	4	8	19	T_M	2 I/O
Flipping + 5 stages [5]	4	8	11	T_M	2 I/O

3. Scaling Step:

$$d_i = K_2 \times d_i^2; \tag{7}$$

$$s_i = K_1 \times s_i^2. \tag{8}$$

For the hardware implementation, the computation of these lifting steps can be mapped by a direct way [3, 6]. Thus, the delay time is restricted by the predictor or updater and the critical path can be reduced by using more pipeline registers. Furthermore, to shorten the critical path with less pipeline registers, a rescheduling algorithm [4] was proposed to merge the computation of each lifting step such that the flattened equation can be calculated by one addition or one multiplication at each operation. The rescheduling algorithm is described as follows: In the first lifting step, Eq. (3) is substituted into Eq. (4) to merge the predictor and updater into one equation.

$$\begin{aligned} s_i^1 &= s_i^0 + \beta \times \left(d_{i-1}^1 + d_i^1\right) \\ &= s_i^0 + \beta \times \left\{\left[d_{i-1}^0 + \alpha \times \left(s_{i-1}^0 + s_i^0\right)\right] + \left[d_i^0 + \alpha \times \left(s_i^0 + s_{i+1}^0\right)\right]\right\} \\ &= s_i^0 + \left(\beta \times d_{i-1}^0 + \beta\alpha \times s_{i-1}^0 + \beta\alpha \times s_i^0\right) \\ &\qquad + \left(\beta \times d_i^0 + \beta\alpha \times s_i^0 + \beta\alpha \times s_{i+1}^0\right). \end{aligned} \tag{9}$$

Similarly, in the second lifting step, Eq. (5) is substituted into Eq. (6) as the first lifting step to get the new equation.

$$\begin{aligned} s_i^2 &= s_i^1 + \delta \times \left(d_{i-1}^2 + d_i^2\right) \\ &= s_i^1 + \delta \times \left\{\left[d_{i-1}^1 + \gamma \times \left(s_{i-1}^1 + s_i^1\right)\right] + \left[d_i^1 + \gamma \times \left(s_i^1 + s_{i+1}^1\right)\right]\right\} \\ &= s_i^1 + \frac{\delta}{\beta} \times \left\{\left[\beta d_{i-1}^1 + \beta\gamma \times \left(s_{i-1}^1 + s_i^1\right)\right] + \left[\beta d_i^1 + \beta\gamma \times \left(s_i^1 + s_{i+1}^1\right)\right]\right\} \\ &= s_i^1 + \left(\frac{\delta}{\beta} \times \beta d_{i-1}^1 + \delta\gamma \times s_{i-1}^1 + \delta\gamma \times s_i^1\right) \\ &\quad + \left(\frac{\delta}{\beta} \times \beta d_i^1 + \delta\gamma \times s_i^1 + \delta\gamma \times s_{i+1}^1\right). \end{aligned} \tag{10}$$

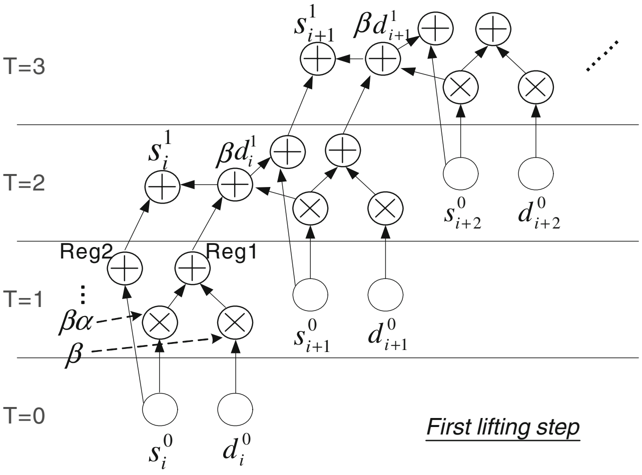


Figure 2 The DFG of the first lifting step of the 1-D 9/7 filter.

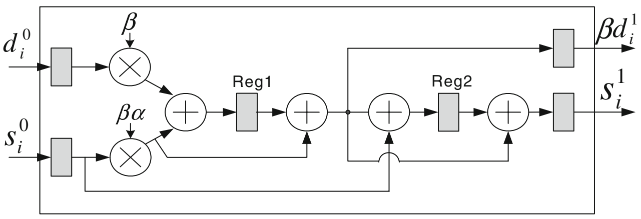


Figure 3 Column-processing element (Column-PE) of the first lifting step of the 9/7 filter.

Finally, the scaling step is represented as follows:

$$d_i = \frac{K_2}{\delta} \times \left(\delta d_i^2\right) \tag{11}$$

$$s_i = K_1 \times s_i^2. \tag{12}$$

Based on the rescheduling algorithm, the critical path can be shortened to one multiplier delay with less pipeline registers compared with direct mapping architecture. Moreover, the flipping structure [5] eliminates the multipliers on the path from input node to the computation node to ease the critical path. As shown in Fig. 1(a), the serious timing accumulation problem can be released by multiplying the inverse coefficients shown in Fig. 1(b). Thus, less pipeline registers are required to achieve the same delay time. Compared with the data path of primitive structure in Fig. 1(a) and flipped structure in Fig. 1(c), the flipping structure releases the critical path of serious timing accumulation problem and preserves the data dependency of primitive lifting-based DWT. Table 1 summaries the three architectures of lifting-based DWT for the 9/7 filter. Direct mapping architecture requires more pipeline registers to archive one multiplier delay (T_M) for the 1-D column processor. Flipping structure eases the critical path without any hardware overhead. Thus, it uses only 11 pipeline registers to achieve one multiplier delay. Since the memory size of 2-D architecture is highly related to the 1-D architecture, it is a crucial issue to shorten the critical path with less pipeline registers and memory size for the 2-D DWT design.

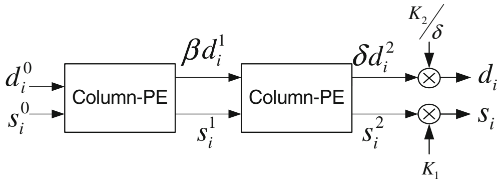


Figure 4 Proposed architecture of the 1-D DWT for the 9/7 filter.

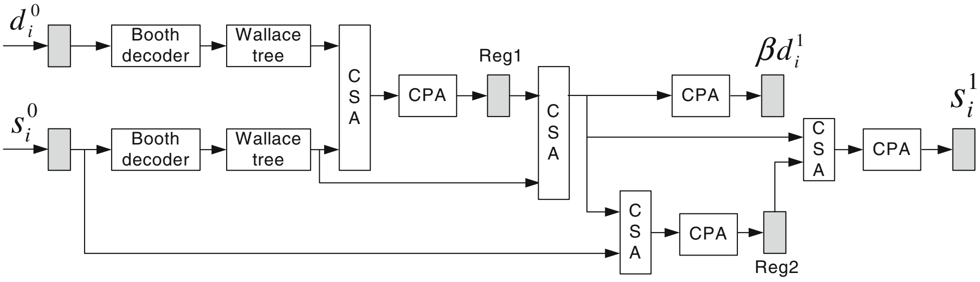


Figure 5 Optimized architecture of the processing element.

3 Proposed 2-D DWT Architecture for the 9/7 Filter

To further reduce the critical path with less pipeline registers, we explore the data flow graph (DFG) of rescheduling DWT algorithm from Eq. (9) to Eq. (12). Fig. 2 shows the DFG of the first lifting step described in Eq. (9). Two input data are entered to the system per clock cycle, and 2 multipliers and 4 adders are required to perform one lifting step computation. Reg1 and Reg2 represent the pipeline registers of the proposed architecture with one multiplier and two adders delay (i.e. T_M+2T_A). Figure 3 presents the column-processing element (Column-PE) for the first lifting step. Moreover, Since Eq. (10) has the same computation form compared to Eq. (9), the 9/7 filter can be realized by cascading two PEs and two scaling multipliers, which perform the rescheduling DWT algorithm. The overall architecture of 1-D column processor is shown in Fig. 4. To optimize the delay time and share the carry propagation adder (CPA) of each multiplier, several carry-save-adder (CSA) trees are applied to calculate the addition in Fig. 3. As shown in Fig. 5, the critical path can be reduced to T_M+2T_{CSA} delay (i.e. $T_{CSA}\approx T_{1\text{-bit FA}}$).

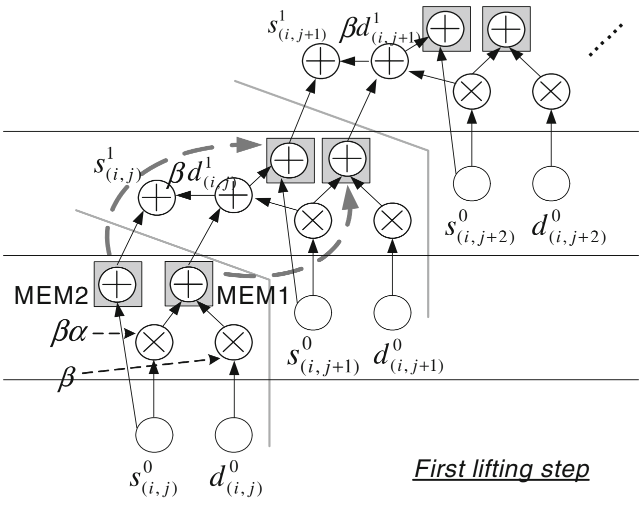


Figure 6 The data path of the i-th row-wise transform for 2-D DWT (i.e. MEM1 and MEM2 are the N-size memories).

Although more pipeline registers can be applied to achieve higher processing speed, it also increases the internal memory size of row processor for 2-D DWT [5, 6]. This is because more pipeline stages would prolong the data dependency of each line transform leading to the larger memory requirement for 2-D architecture. Based on the proposed data path, it only requires 2N size memory to perform the computation of one lifting step for row-wise transform. This is because the data dependency of each lifting step is reduced to only two computation nodes. As shown in Fig. 6, MEM1 and MEM2 are used to save the temporal results of each row-wise transform. For example of the i-th row-wise transform, if two new input data, $s^0_{(i,j+1)}$ and $d^0_{(i,j+1)}$, are obtained, MEM1 and MEM2 are firstly read to partially perform the i-th row-wise transform and the new temporal results are then updated to the next nodes. The outputs, $s^1_{(i,j)}$ and $\beta d^1_{(i,j)}$, are used to perform the second lifting step computation. Similar to the 1-D case, the row-processing element can be derived by substituting Reg1 and Reg2 in Fig. 5 to MEM1 and MEM2, as shown in Fig. 7. Finally, the overall 2-D architecture can be realized by cascading the column processor, transposing buffer [11], row processor and scaling multipliers, as shown in Fig. 8. It uses $N/2\times N/2$ size external memory to store the LL-band output coefficients.

Table 2 compares several architectures for the 1-D DWT. Based on the 2 input samples per cycle, 4 multipliers and 8 adders are required to perform two lifting steps. Direct mapping architecture [3] requires 32 registers to achieve one multiplier delay time. Based on the rescheduling algorithm [4], it decreases the number of pipeline registers to 19. By releasing multiplications of the critical path, flipping structure [5] uses only 11 pipeline registers to perform one multiplier delay time. The architecture for modified algorithm [7] requires 20 registers with one input sample per cycle. Based on the proposed data path, the proposed architecture can further decrease the register number to 10 with one multiplier and two one-bit full-adder delay time.

Table 3 compares several row processors of one-level 2-D DWT architecture for the 9/7 filter. Based on the direct

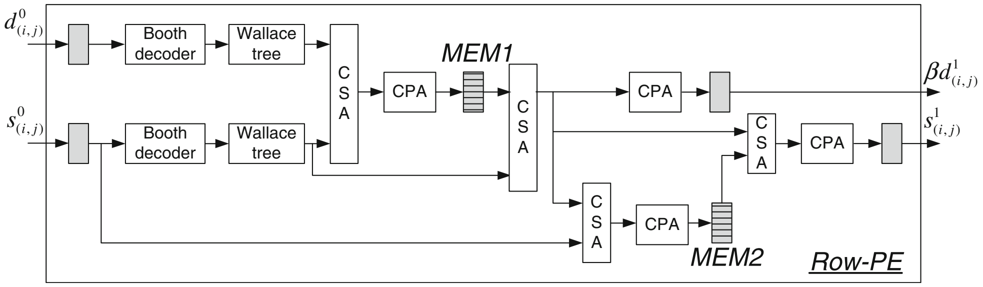


Figure 7 The architecture of row-processing element (Row-PE).

mapping implementation, DSA architecture [6] uses less pipeline registers with $4T_M+8T_A$ delay to achieve 4N size memory. However, the temporal buffer size would become large while the critical path is shortened. Flipping structure [5] eases the critical path without adding pipeline registers. Thus, it uses 4N size memory to perform T_M+5T_A delay. The modified algorithm [7] can achieve one multiplier delay with 4N size memory with one input sample per cycle. Since the proposed data path reduces the data dependency of each lifting step to only two computation nodes, the proposed architecture only requires 4N size memory to perform the computation of two lifting steps with $T_M+2T_{1\text{-bit FA}}$ delay. Besides, more pipeline registers can be further applied to shorten the critical path (i.e. $T_M+2T_{1\text{-bit FA}}$) without increasing the internal memory of 4N. Thus, the tradeoff between high speed and low memory can be solved.

4 Memory-Bit Width Simulations

In this section, we show some simulation results of JPEG 2000 coding with different memory bit-widths in 2-D DWT. Since the 9/7 filter is commonly used for the lossy compression of JPEG2000, a portion of compressed data would be discarded after performing EBCOT Tier-2 process, especially that from low bit-planes. Several methods have been proposed to reduce the computation of EBCOT [8, 9]. For high compression ratios, the precision of DWT can be further reduced to save the power consumption. As for the 2-D DWT implementation, different memory architectures would dominate the result of power consumption [10]. It also implies that controlling the active memory banks under various compression ratios is an efficient way to reduce the power consumption [12]. To illustrate the relation between the image quality and memory bit-width, we perform four levels of DWT and IDWT with different memory bit-widths on several 512×512 test images. In the simulation, the fixed-point operation is applied and realized by the 16-bit adder and 16×12 bit multiplier; the constant coefficient of each lifting step is quantized by 12-bit binary representation. Finally, the 16-bit internal data bus is divided into 11 integer bits and 5 fraction bits [7].

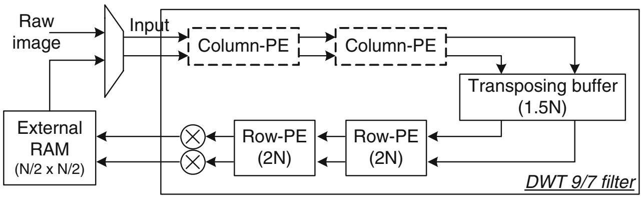


Figure 8 The overall 2-D DWT architecture (Image size: N×N).

Now, we show the simulation results of transposing and temporal buffers with different bit-widths. Throughout the simulation, we choose 40 dB of DWT process as a minimal PSNR value to perform JPEG 2000 coding. We show, in Table 4, the PSNR value with variable data width of transposing buffer. It is found that 12-bit precision of transposing buffer can approach the quality of 16-bit precision operation. Therefore, we pick up the transposing buffer with 16, 12, 11 and 10 bits to perform four levels of DWT and IDWT with variable data width of temporal buffer, as shown in Fig. 9. The simulation results imply that the point with 10 bit-width transposing buffer and 14 bit-width temporal buffer has the least DWT precision (i.e. 41.85 dB) among the working modes considered here. Table 5 summarizes the data width of memory for the five chosen modes.

In Figs. 10 and 11, the JPEG 2000 rate-distortion (R-D) curves with five DWT precisions are compared with the software model [13]. The software model is based on the computation of 32-bit precision and is treated practically as the maximal precision of all operation modes considered above. From Fig. 10, the curve of "Trans16_Temp16" mode is close to the software model. Moreover, "Trans12_Temp16" and "Trans16_Temp16" modes perform about the same R-D curve (i.e. less than 0.2 dB). Although "Trans12_Temp14" and "Trans10_Temp14" modes suffer more image degradation at low compression ratios, the R-D curves would approach the software model when increasing the compression ratio (i.e. less than 0.5 and 0.7 dB while

Table 2 Comparisons of several 1-D column processors for the 9/7 filter.

Architecture	Multi-plier	Adder	Register	Critical path	Band-width
Direct + 4 stages [3]	4	8	16	T_M+$2T_A$	2 I/O
Direct + 32 stages	4	8	32	T_M	2 I/O
Rescheduling [4]	4	8	19	T_M	2 I/O
Flipping + 5 stages [5]	4	8	11	T_M	2 I/O
Modified [7]	2	4	20	T_M	1 I/O
Proposed (optimized)	4	8	10	T_M+$2T_{1\text{-bit FA}}$	2 I/O

Table 3 Comparisons of several row processor architectures for the single-level 2-D DWT architecture (N: image height).

Architecture	Multi-plier	Adder	Critical path	Temporal buffer	Band-width
DSA [6]	4	8	$4T_M$+$8T_A$	4N	2 I/O
Direct + 4 stages	4	8	T_M+$2T_A$	10N	2 I/O
Direct + fully pipe.	4	8	T_M	32N	2 I/O
Flipping + no pipe.[5]	4	8	T_M+$5T_A$	4N	2 I/O
Flipping + 5 stages[5]	4	8	T_M	11N	2 I/O
Modified [7]	2	4	T_M	4N	1 I/O
Proposed (optimized)	4	8	T_M+$2T_{1\text{-bit FA}}$	4N	2 I/O

Table 4 PSNR of several test images with different data bits for transposing buffer (i.e. performing 4-level DWT/IDWT).

Transposing Buffer (Bit-Width)	16	15	14	13	12	11	10	9	8
Lenna512	49.76	50.14	49.89	49.90	49.33	47.76	44.26	39.45	34.07
Baboon512	50.13	50.36	50.30	50.25	49.57	47.91	44.33	39.18	33.51
Pepper512	49.27	49.54	49.41	49.30	48.94	47.38	44.06	39.16	33.82
Airplane512	49.01	49.38	49.23	48.98	48.66	47.22	44.08	39.29	33.93
Boat512	49.82	49.99	49.97	49.83	49.24	47.47	44.23	39.22	33.55
Avg. PSNR (dB)	49.60	49.88	49.76	49.65	49.15	47.55	44.19	39.26	33.78

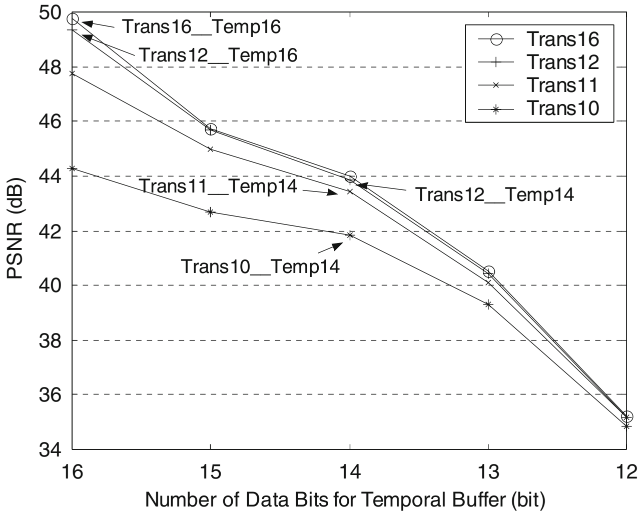


Figure 9 PSNR with different data bits of temporal buffer and transposing buffer (i.e. performing 4-level DWT/IDWT).

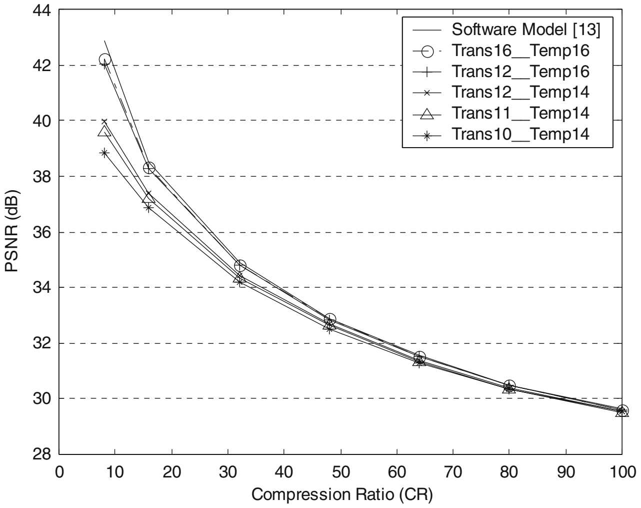


Figure 10 R-D curves of JPEG 2000 for 512×512 Lenna image with five operation modes.

CR is larger than 32). For the Baboon image shown in Fig. 11, all five modes achieve the R-D curve of the software model (i.e. less than 0.5 dB). Therefore, from the simulation results, we conclude that under different compression ratios, the data width of memory can be properly reduced with slight image degradation. In the modes considered here, “Trans12_Temp16” mode is very close to the software model. We emphasize that when the compression ratio becomes high, the memory data width can be further reduced (i.e. “Trans10_Temp14” mode). Therefore, under various compression ratios and applications, it is practical to determine the working operation modes first and then design the corresponding energy-aware memory architecture to reduce the power consumption [14].

5 Conclusion

In this paper, we propose a high-speed and low-memory pipelined architecture for the lifting-based 2-D DWT of the 9/7 filter defined in JPEG 2000. The critical path of 1-D architecture can be shortened by using less pipeline registers compared with other architectures. Besides, more pipeline registers can be used to enhance the processing speed without increasing the internal memory size for the 2-D case. Thus, the tradeoff between high-speed and low-memory can be eased. Moreover, since a portion of DWT coefficients from least significant bits would be discarded after the rate-distortion optimization, one can reduce the bit-width of internal memory for various compression ratios and applications to make efficient power consumption.

Table 5 Five operation modes with specified memory bit-width.

Operation mode	Trans16_ Temp16	Trans12_ Temp16	Trans12_ Temp14	Trans11_ Temp14	Trans10_ Temp14
Transposing buffer (bit)	16	12	12	11	10
Temporal buffer (bit)	16	16	14	14	14
Multiplier (bit × bit)	16×12	16×12	16×12	16×12	16×12
Adder/ Register (bit)	16	16	16	16	16

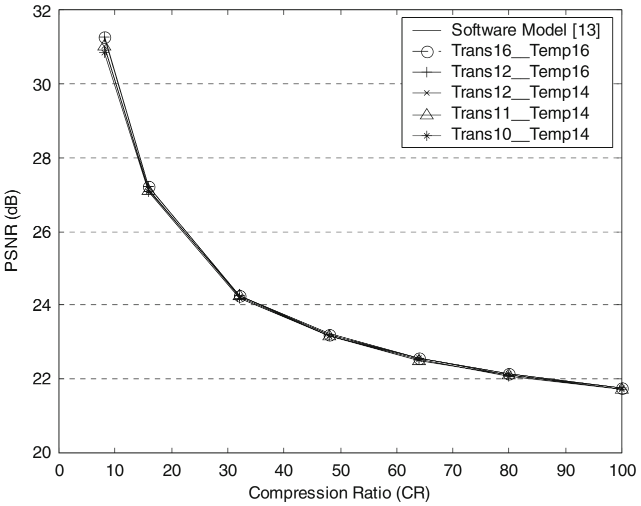


Figure 11 R-D curves of JPEG 2000 for 512×512 Baboon image with five operation modes.

References

1. ISO/IEC. ISO/IEC 15444-1. Information technology–JPEG 2000 image coding system, 2000.
2. Daubechies, I., & Sweldens, W. (1998). Factoring wavelet transform into lifting steps. *The J of Fourier Analysis and Applications, 4*, 247–269.
3. Jou, J. M., Shiau, Y. H., & Liu, C. C. (2001). Efficient VLSI architectures for the biorthogonal wavelet transform by filter bank and lifting scheme. *in Proc. IEEE International Symposium on Circuits and Systems, 2*, 529–532.
4. Wu, B. F., & Lin, C. F. (2003). A rescheduling and fast pipeline VLSI architecture for lifting-based discrete wavelet transforms. *in Proc. IEEE International Symposium on Circuits and Systems, 2*, 732–735.
5. Huang, C. T., Tseng, P. C., & Chen, L. G. (2004). Flipping structure: an efficient VLSI architecture for lifting-based discrete wavelet transform. *IEEE Transactions on Signal Processing, 52* (4), 1080–1089.
6. Liao, H., Mandal, M. K., & Cockburn, B. F. (2004). Efficient architectures for 1-D and 2-D lifting-based wavelet transforms. *IEEE Transactions on Signal Processing, 52*(5), 1315–1326.
7. Wu, B. F., & Lin, C. F. (2005). A high-performance and memory-efficient pipeline architecture for the 5/3 and 9/7 Discrete wavelet Transform of JPEG2000 codec. *IEEE Transactions on Circuits and Systems for Video Technology*, 15, 1615–1628.
8. Chang, T. H., Lian, C. J., Chen, H. H., Chang, J. Y., & Chen, L. G. (2003). Effective hardware-oriented technique for the rate control of JPEG2000 encoding. *in Proc IEEE International Symposium on Circuits and Systems, 2*, 684–687.
9. W. Yu (2004). Integrated rate control and entropy coding for JPEG 2000. in *Proc. IEEE Data Compression Conference*, 152–161.
10. Huang, C. T., Tseng, P. C., & Chen, L. G. (2004). Memory analysis and architecture for two-dimensional discrete wavelet transform. *in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, 5*, 13–16.
11. Huang, C. T., Tseng, P. C., & Chen, L. G. (2005). Generic RAM-based architectures for two-dimensional discrete wavelet transform with line-based method. *IEEE Transactions on Circuits and Systems for Video Technology, 15*(7), 910–920.
12. Rabaey J. M., & Pedram, M. (2002). Power Aware Design Methodologies: Power Optimization by Datapath Width Adjustment (Chapter 7). Kluwer Academic Publishers.
13. JPEG 2000 Software model [Online]. Available: http://www.ece.uvic.ca/~mdadams/jasper/
14. Bhardwaj, M., Min, R., & Chandrakasan, A. P. (2001). Quantifying and enhancing power awareness of VLSI systems. *IEEE Transactions on VLSI Systems, 9*(6), 757–772.

Chung-Fu Lin was born in Taipei, Taiwan in 1977. He received the B.S. degree in mechanical engineering from National Central University in 1999 and M.S. degree in mechanical engineering from National Chiao Tung University in 2001. In 2006, he received the Ph.D. in the electrical and control engineering from National Chiao Tung University. He received the Macronix Golden Silicon Honorable Mention in the 3rd Semiconductor and Application Competition sponsored by Macronix International Co., Ltd., in 2003. His research interests include multimedia application, VLSI design and implementation. Currently, he is working at Faraday Technology Corporation.

Pei-Kung Huang was born in Kaohsiung, Taiwan in 1981. He received the B.S. degree in electronics engineering from Chang Gung University in 2003 and M.S. degree in electrical and control engineering from National Chiao Tung University in 2005. He received the Macronix Golden Silicon Honorable Mention in the 5th Semiconductor and Application Competition sponsored by Macronix International Co., Ltd., in 2005. His research interests include signal processing, VLSI implementation and low-power design. Currently, he is working at PixArt Imaging Inc.

Bing-Fei Wu was born in Taipei, Taiwan in 1959. He received the B.S. and M.S. degrees in control engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1981 and 1983, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1992. Since 1992, he has been with the Department of Electrical Engineering and Control Engineering, where he is currently a Professor. He has been involved in the research of Intelligent Transportation Systems for many years and leads a research team to develop the first Taiwan smart car, TAIWAN*i*TS-1, with autonomous driving and active safety system. His current research interests include vision-based vehicle driving safety, intelligent vehicle control, multimedia signal analysis, embedded systems and chip design.

Prof. Wu founded and served as the Chair of the IEEE Systems, Man and Cybernetics Society Taipei Chapter in Taiwan, 2003. He has been the Director of The Research Group of Control Technology of Consumer Electronics in the Automatic Control Section of National Science Council (NSC), Taiwan, from 1999 to 2000. As an active industry consultant, he also involves in the chip design and applications of the flash memory controller and 3C consumer electronics in multimedia systems. The research has been honored by the Ministry of Education as the Best Industry-Academics Cooperation Research Award in 2003. He received the Fuji/Xerox Academic Research Award in 2007; the Distinguished Engineering Professor Award from Chinese Institute of Engineers in 2002; the Outstanding Information Technology Elite Award from Taiwan Government in 2003; the First Prize Award of TI China-Taiwan DSP Design Contest in 2006; the Outstanding Research Award in 2004 from NCTU; the Golden Acer Dragon Thesis Award sponsored by the Acer Foundation in 1998 and 2003, respectively; the First Prize Award of the We Win (Win by Entrepreneurship and Work with Innovation & Networking) Competition hosted by the Industrial Bank of Taiwan in 2003; and the Silver Award of Technology Innovation Competition sponsored by the Advantech Foundation in 2003.