An area- and power-efficient FIFO with error-reduced data compression for image/video processing

Zeinolabedin, Seyed Mohammad Ali; Kim, Tony T.; Zhou, Jun; Liu, Xin

2014

Zeinolabedin, S.M.A., Zhou, J., Liu, X., & Kim, T. T. (2014). An area- and power-efficient FIFO with error-reduced data compression for image/video processing. 2014 IEEE International Symposium on Circuits and Systems (ISCAS), 2277-2280.

<https://hdl.handle.net/10356/79393>

<https://doi.org/10.1109/ISCAS.2014.6865625>

© 2015 Institute of Electrical and Electronics Engineers (IEEE).

Downloaded on 26 Apr 2023 20:40:37 SGT

An Area- and Power-Efficient FIFO with Error-Reduced Data Compression for Image/Video Processing

Seyed Mohammad Ali Zeinolabedin¹, Jun Zhou², Xin Liu² and Tony T. Kim¹

¹VIRTUS, School of EEE, Nanyang Technological University, Singapore 639798

²Institute of Microelectronics, A*STAR, Singapore 117685

Email: seyedmoh001@e.ntu.edu.sg

Abstract— Filtering is a key component of many digital image/video processing algorithms. It often requires FIFO to temporarily buffer the pixels data for later usage. The FIFO size is proportional to the length of the filters and input data width, causing large area and power consumption. This paper presents a technique named FIFO with error-reduced data compression (FERDC) to reduce the FIFO size for various filters. The proposed FERDC significantly reduces the area and power consumption while keeping the error metrics such as mean square error (MSE) and peak signal to noise ratio (PSNR) in the acceptable range. Simulation results of a two dimensional wavelet filter shows that the proposed FERDC technique achieves the FIFO size reduction of up to 44.44% with PSNR values larger than 39 dB, which leads to the reduction of at least 31.6% in the dynamic power and 44.44% in the leakage power.

Keywords—image/video signal processing; FIFO; low power

I. INTRODUCTION

Rapid advancement in VLSI systems has enabled the real-time hardware implementation of high computational complexity image/video processing algorithms. Some recent works requiring fast hardware implementation are video codec [1], computational photography [2], wavelet transform [3], laplacian pyramid [4] and biomedical applications [5]. Early image/video processors focused on improving the speed/throughput of the systems while the contemporary state-of-the-art image/video processors confront additional challenges such as low power consumption and real-time processing.

Many image/video processing algorithms require digital filtering like image enhancement, image restoration, image compression, image segmentation, and video coding [6], [7] in both spatial and frequency domains. Image processing algorithms such as edge detection algorithms manipulate image pixels in the spatial domain while frequency domain is employed in fourier-transform-based algorithms. Regarding the hardware implementation of image/video signal processing algorithms, many filtering operations require storing part of input image or part of previous stage results temporarily. However, it has not been comprehensively investigated to reduce the size of the temporary pixel buffers (i.e. FIFO). The size of the FIFO significantly affects the total area and power of a design. It has been reported that FIFOs consume more than

75% of the route area and power in [8] and more than 80% of power consumption in [4]. Various approaches have been developed to reduce the power consumption of FIFOs. However, most of them are focusing on the circuit-level techniques and memory-splitting schemes along with power gating [8], [9]. In this paper, we propose a new architecture-level technique to reduce the size of FIFO only with negligible degradation in the error metrics such as mean square error (MSE) and peak signal to noise ratio (PSNR). The reduced FIFO size through the proposed method also improves the dynamic and leakage power consumptions.

II. FIFO FOR IMAGE/VIDEO SIGNAL PROCESSING

Filtering is an inevitable function of image/video processing algorithms, which can be generalized as:

$$g(x, y) = T_{xy} * f(x, y) \quad (1)$$

where $f(x, y)$ is an input image, T_{xy} is the filter transfer function and $g(x, y)$ is the output. The output is defined as the convolution of the pixel values located in a given filter window (T). The window is usually a square with $W \times W$ pixels where W is an odd number to have an exact center pixel. Obviously, T is a two dimensional (2D) filter where T_{xy} represents the filter coefficient. The 2D filter can be converted into two one dimensional (1D) filters:

$$T_{xy} = t_{xy} \times (t_{xy})' \quad (2)$$

Here, t_{xy} is the coefficient of the $1 \times W$ filter, t . In this scheme, t is first applied to the horizontal rows of the input, followed by the vertical columns or vice versa. In either case, hardware implementation of the filter needs to be done by stream processing. For the 2D $W \times W$ filter, it requires $W - 1$ FIFOs to store $(W - 1)$ rows of N pixels (Fig. 1.(a)) where N is the input image width. So the number of reads is W pixels during each clock cycle. Otherwise, $W \times W$ pixels have to be read every clock cycle to fill the filter window. Fig. 1 depicts the two general filter configurations where FIFOs are placed in parallel with the filters for implementing 2D filtering [10].

FIFOs have significant effects in the circuit area and power consumption, particularly when the size of the filter, W , is large. Therefore, it is highly required to reduce the size of FIFOs. Several research works report FPGA cores whose

Fig. 1. Figure 1. Sample filter configurations: (a) 2D filter and (b) two 1D filters.

memories are configured as FIFOs to reduce the area and power cost. Dual-port RAMs have also been exploited to further improve area and power efficiency while achieving the same functionality [10]. However, as mentioned earlier there is still a gap in the literature to reduce the size of FIFO leading to area and power consumption reduction.

III. PROPOSED FIFO ARCHITECTURE

In this section, we propose a new technique that can be used to reduce the data width of FIFO in image/video processing compared to the conventional method shown in Fig. 1. Fig. 2 illustrates the proposed method named as FIFO with error-reduced data compression (FERDC). It employs a concept of pixel prediction where every pixel can be predicted utilizing adjacent pixels. For encoding, input data, x_i is given to a differential predictor which calculates the difference between the consecutive inputs denoted as y_i . The differential predictor is to remove horizontal correlations between consecutive inputs. In that case, it ensures the energy of input correlated data, x_i (Fig. 3(a)), is compacted around zero, y_i (Fig. 3(b)) and only a few large numbers spread throughout the entire B-bit integer value range. Now the difference values (e.g. Fig. 3(b)) can be quantized to b-bit integer values to reduce the data width and thus the FIFO size significantly. Before sending y_i to quantizer, the energy-compacted difference is updated by the quantization error of previous difference (E_{i-1}) to prevent the quantization errors from being accumulated at the output, x_{iq} . Eventually, b-bit output (y_{iq}) of the quantizer block (Q.) is stored in the FIFO. For decoding, the read data from the FIFO are sent to the inverse quantizer (I.Q.) to retrieve the b-bit data corresponding to the original input data. Finally, the inverse differential predictor generates the output (x_{iq}) using the retrieved b-bit values. This proposed architecture reduces the FIFO width from B bits to b bits.

Two figure-of-merits (FOM) are employed to measure the distortion introduced by quantizers [11]. They are mean square error (MSE) and peak signal to noise ratio (PSNR). MSE represents the difference between an ideal output and the quantized one. PSNR shows how a reconstructed image

Figure 2. Architecture of the proposed FERDC technique.

Figure 3. Histogram of "Barbara" (First Row) for (a) x_i and (b) y_i .

Figure 4. Differential and Inverse Differential predictors architectures.

differs from the original one. Following are the general expressions for MSE and PSNR.

$$MSE = \frac{1}{n} \sum_{i=1}^n (P_i - Q_i)^2 \quad (3)$$

$$PSNR = 10 \log_{10} \frac{(\max(P_i))^2}{MSE} \quad (4)$$

Here, P_i is the original pixel value or ideal output coefficient Q_i is the reconstructed pixel value or quantized output coefficient and n is the number of total pixels or coefficients. In general cases, an acceptable PSNR value is larger than 35 resulting in MSE less than 20 [11]. Higher PSNR values indicate that reconstructed images are closer to original images.

In order to reduce the overhead circuitry, a B-bit subtractor and adder along with a B-bit register is used for both differential predictor and inverse differential predictor as shown in Fig. 4. Besides the error update block is a (B+1)-bit subtractor. To implement a power- and complexity-effective quantizer, a non-uniformly distributed symmetric quantizer is proposed following the same distribution of its input data (Fig. 3(b)). The quantizer comprises two parts named as Q_p and Q_n which are applied to positive and negative numbers. Q_p and Q_n have four segments determined by fL , sL and tL as depicted in Fig. 5. The first three segments have the step sizes of 1, 2 and 4 respectively while the last segment has a constant value. For example, the positive number, i , between zero and fL are quantized into $[i] + 0.5$, the positive number i between fL and sL are quantized into $2[(i - fL)/2] + 1 + fL$ and the positive number between sL and tL are quantized into $4[(i - sL)/4] + 2 + sL$ and finally all other positive numbers are quantized into a constant value, $tL + 13$. The same quantization happens for the negative numbers.

Figure 5. Quantizer used for positive (Q_p) and negative (Q_n) numbers.

Figure 6. Direct-form Implementation of horizontal filters.

Fig. 5 facilitates the calculation of the new FIFO cells width, b , by applying the total number of quantized levels denoted as L to (5). For instance, if L is 256, b becomes 8 bits. Now for a certain value of b , the quantizer parameters fL , sL and tL can be selected by monitoring the best values of MSE and PSNR. Besides, since many image processing algorithms are implemented with fixed-point arithmetic, B can be measured at the output of the horizontal filter (Fig. 1(b)) performed by the direct-form implementation as shown in Fig. 6. In Fig. 6, if the input data is represented by K -bit values and the filter coefficients (tap weight), t_{ij} , are quantized into M integer bits, the worst-case value of B is calculated as (7), where $+1$ is for the sign bit. In order to compensate the M -bit integer representation of the filter coefficients, a M -bit right shifter is added in Fig. 6.

$$L = fL + \frac{sL+tL}{2} + 2, \quad L = 2^b \quad (5)$$

$$C = \left\lceil \log_2 \left(\sum_{j=1}^W |t_{1j}| \right) \right\rceil + \left\lceil \log_2 (2^k - 1) \right\rceil \quad (6)$$

$$B = C - M + 1 \quad (7)$$

IV. EXPERIMENTAL RESULTS

To verify the proposed technique, we have applied the proposed FERDC technique to the FIFOs exploited in the column filtering of the 2D wavelet transform described in Fig. 7. The utilized filters are 1D Cohen-Daubechies-Feauveau (CDF) 9/7 biorthogonal filters and wavelet Daubechies D4 filters (db4) [12] that are both implemented by the architecture explained in Fig. 1(b). Various standard test images in grayscale are utilized to explore the efficiency of the proposed method in terms of FIFO size, MSEs and PSNRs.

Figure 7. Ideal 2D wavelet transform

We developed two wavelet transform architectures for comparison. First, an ideal wavelet transform is implemented by 9/7 biorthogonal filters in MATLAB and examined to estimate the size of FIFOs for filtering, which are indicated by BL and BH in Fig. 7. All the standard test images summarized in Fig. 8 have BL of 10 bits and BH of 9 bits except “Peppers” and “Lena” that have BH of 8 bits. MSEs of LL (Low-Low), LH (Low-High), HL (High-Low) and HH (High-High) subbands are 0.09, 0.08, 0.09 and 0.08 respectively. For all the images, PSNR is larger than 58 dB.

Second, the FIFOs in Fig. 7 are replaced by the proposed FERDC-based FIFOs. The quantization parameters, fL , sL and tL , can be obtained by (5) for a certain value of L . We chose the values of fL , sL and tL leading to MSEs less than 20 and PSNRs greater than 35 dB as general accepted values [11]. In this work, we select $L = 128$ and $L = 64$ (corresponding to 7- and 6-bit FIFO cell) for the upper and lower parts for investigating fL , sL and tL . The results are shown in Fig. 8. In this case, the FIFO sizes are reduced by 30 % and 33.33% in the upper and lower parts, respectively. Further area reduction can be achieved by lowering L in the upper and lower parts (corresponding to 6- and 5-bit FIFO cell), whose results are summarized in Fig. 9. By lowering L to 64 and 32, the proposed technique accomplishes the FIFO size reduction of 40 % and 44.44 % in the upper and lower parts, respectively. This maintains MSEs less than 13.74 and PSNRs larger than 39 dB. The same simulations with Daubechies D4 filters (db4) are invoked and the results are illustrated in Fig. 8 and Fig. 9 as well. In Fig. 8, the worst case MSE is 9.29 with PSNR of 42.30 (Fig. 10) from ‘Babara’ due to relatively large errors introduced by the quantizer. In Fig. 9, ‘Peppers’ shows the worst case MSE of 16.28 with PSNR of 39.52 (Fig. 9). Notice that the same quantizer parameters, fL , sL and tL , are employed for both filters.

Two main components of dynamic power are an activity factor, α , and load capacitance, C as described in (8). Considering the transitions of data and the load capacitance reduction in the FIFO, which is shown in (9), the dynamic power improvement of the FIFO is illustrated in Fig. 11. It shows that the dynamic power reduction related to activity factor and load capacitance is at least 54.6% and 31.6% for upper and lower parts respectively. Also the leakage power reduction of 30 to 44.4 % is expectable because total leakage is proportional to a circuit size [13].

$$P_{dynamic} \sim \alpha CV^2 f \quad (8)$$

$$C_{new\ design} \cong 0.6\ C_{original\ design} \quad for\ L = 64, 32 \quad (9)$$

Figure 8. FERDC-based 2D wavelet transform with $L=128$ in the upper part and $L=64$ in the Lower part ($fL=28$ $sL=34$ $tL=154$ used in upper part FIFOs and $fL=18$ $sL=29$ $tL=51$ used in lower part FIFOs).

Figure 10. PSNR of Reconstructed Images implemented by 9/7 and db4 filters with $L=128,64$ and $L=64,32$.

V. CONCLUSION

We have presented an architectural-level technique to reduce the FIFO size leading to dynamic and leakage power reduction significantly. It utilizes the spatial correlation between neighboring pixels and performs error-reduced data compression together with quantization to keep MSEs and PSNRs minimized. The proposed method has been implemented in a 2D wavelet with 9/7 biorthogonal filters as well as Daubechies D4 filters for demonstration. The experimental results show that up to 44.44% reduction in FIFO size is achieved, which leads to at least 31.6% reduction in dynamic power and 44.44% reduction in leakage power. The proposed FIFO architecture can be directly employed in other image/video processing algorithms.

REFERENCES

- [1] C. T. Huang, M. Tikekar, C. Juvekar, V. Sze, and A. Chandrakasan, "A 249Mpixel/s HEVC Video-Decoder Chip for Quad Full HD Applications," in IEEE International Solid-State Circuits Conference, Digest of Technical Papers, pp.162-163, Feb 2013.
- [2] R. Rithe, P. Raina, N. Ickes, S. V. Tenneti, and A. P. Chandrakasan "Reconfigurable Processor for Energy-scalable Computational Photography," in IEEE International Solid-State Circuits Conference, Digest of Technical Papers, pp.164-165, Feb 2013.
- [3] C. Zhang, C. Wang, and M. O. Ahmed, "A Pipeline VLSI Architecture for Fast Computation of the 2-D Discrete Wavelet Transform," IEEE Trans. Circuits and Systems I: Regular Papers, vol. 59, pp. 1775-1785, Aug. 2012.

Figure 9. FERDC-based 2D wavelet transform with $L=64$ in the upper part and $L=32$ in the Lower part ($fL=6$ $sL=27$ $tL=77$ used in upper part FIFOs and $fL=1$ $sL=4$ $tL=46$ used in lower part FIFOs).

Figure 11. Dynamic Power Improvement for FIFOs in 2D wavelet transform implemented by 9/7 and db4 filters with $L=64,32$.

- [4] S. M. A. Zeinolabedin, N. Karimi, and S. Samavi, "Low Computational Complexity Hardware Implementation of Laplacian Pyramid," in 18th Int. Iranian Conf. Electrical Engineering, Isfahan, Iran, May 11-13, 2011.
- [5] W. Al-Atabany, B. McGovern, K. Mehran, R. Berlinguer-Palmini, and P. Degenaar, "A Processing Platform for Optoelectronic/Optogenetic Retinal Prosthesis," IEEE Trans. Biomedical Engineering, vol. 60, no. 3, pp. 781-791, Mar 2013.
- [6] W. K. Pratt, Digital image processing : PIKS Scientific inside, 4th ed., Wiley-Interscience: Hoboken, N.J., 2007, ch. 10.
- [7] R. Rithe, C. Chih-Chi, and A. P. Chandrakasan, "Quad Full-HD Transform Engine for Dual-Standard Low-Power Video Coding," IEEE Journal of Solid-State Circuits, vol. 47, pp. 2724-2736, Nov 2012.
- [8] O. Peng, et al., "A 65nm 39GOPS/W 24-core Processor with 11Tb/s/W Packet-controlled Circuit-switched Double-layer Network-on-chip and Heterogeneous Execution Array," in IEEE International Solid-State Circuits Conference, Digest of Technical Papers, pp. 56-5, Feb 2013.
- [9] T.-H. Kim, J. Liu, J. Keane, and C. Kim, "Circuit Techniques for Ultra-Low Power Sub-threshold SRAMs," in IEEE International Symposium on Circuits and Systems, pp. 2574-2577, May 2008
- [10] D. G Bailey, Design for Embedded Image Processing on FPGAs, John Wiley & Sons: Singapore, 2011, ch. 8.
- [11] D. Salomon, Data Compression: The Complete reference, 4th ed., Springer e-books:London, 2007, ch. 1-4 .
- [12] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," IEEE Trans. Image Processing, vol. 1, pp. 205-220, Apr. 1992.
- [13] V. Vasanth, and F. Michael, "Power reduction techniques for microprocessor systems," ACM Computing Survey, vol. 37, no. 3, pp. 195-237, Sep. 2005.