Design and Analysis of an Approximate 2D Convolver

Ke Chen and Fabrizio Lombardi
Electrical and Computer Engineering Department
Northeastern University
Boston, USA

Jie Han
Electrical and Computer Engineering Department
University of Alberta
Edmonton, Canada

***Abstract*—This paper proposes a two-dimensional (2D) convolver in which both approximate circuit- and algorithm-level techniques are utilized in the design. Truncation is used as circuit techniques, while bit-width reduction is utilized at the algorithm level. These different techniques are related to the configuration of the convolver by which its operation can be configured to meet different and often contrasting figures of merit. Circuit-level simulation (using HSPICE) and an extensive evaluation of different error metrics, generic metrics such as the mean error distance (MED) and the peak signal noise ratio (PSNR) for image convolution are performed. A detailed error analysis is also presented to substantiate the simulation results. Convolution for image processing (Gaussian smoothing) is treated in detail to show the effectiveness of the proposed approach. The design, the analysis and the simulation results show that the approximate techniques utilized in the inexact convolver can operate in synergy.**

Keywords— Approximate design, approximate computing, convolver, image processing, error

I. INTRODUCTION

Image and video processing requires the computation of two-dimensional (2D) convolution for many applications, such as filtering, restoration, feature recognition, object tracking and template matching [1][2]. 2D convolution consists of computing the weighted sum of neighboring pixels. Given an input image, the convolution of the generic pixel P(x,y) is computed by adding the k2 products obtained by multiplying a k×k neighborhood of pixels (centered at P(x,y)) by a k×k convolution kernel. 2D isotropic kernels (such as Gaussian, Laplacian, mean, median, sharpening and smoothing) are frequently used for image and video processing; an isotropic kernel has the property of being equally applicable in all directions of an image, thus with no specific sensitivity or bias towards a particular set of directions. This feature is usually used to reduce the hardware complexity of 2D convolvers.

This paper proposes an approximate 2D convolver in which both circuit and algorithm techniques are utilized in the design. These techniques allow substantial reductions in figures of merit (such as power dissipation and circuit complexity) as well as keeping the loss of accuracy as nearly as desired. Truncation is used as circuit techniques, while bit-width reduction is utilized at the algorithm level. These different techniques are related to the configuration of the convolver by which its operation can be configured to meet different and often contrasting figures of merit. Circuit-level simulation (using HSPICE) and an extensive evaluation of different error metrics (generic metrics such as the mean error distance (MED) and the peak signal noise ratio (PSNR) for image convolution) are performed. A detailed error analysis is also presented to substantiate the simulation results.

II. REVIEW

A. Approximate computing

Energy efficiency has become of paramount concern in the design of today's computing systems. A characteristic of many applications is that often an exact result is not necessary and an approximate or less-than-optimal outcome is also viable. Thus, approximate computing has recently emerged as a promising approach for the energy-efficient design of digital systems [3]. Approximate computing can be accomplished using two types of techniques: circuit-level techniques and algorithm-level techniques. At circuit-level, approximate adders and multiplier are the most commonly used modules. Approximate adders have been proposed by using a reduced number of transistors [4] and by truncating/rearranging the carry propagation chain for a speculation-based operation. Approximate designs achieve good performance in terms of area, power and delay compared to conventional (exact) adders [5][6]. Multiplication can be thought as the repeated sum of partial products; so, inexact adder replacement and truncation techniques can often achieve the desirable approximation for multiplication. At algorithm-level, dynamic bit-width adaptation [7] and power reduction via voltage scaling have been proposed [8][9]. In general, algorithm-level techniques are very flexible, but not always achieving the performance improvements of circuit-level techniques for approximate computing.

B. Convolution

Convolution is one of the basic operations for image and video processing, thus it strongly influences the overall performance of computation for these applications. The computed k×k products are added to generate the output pixel value; the 2D convolution can be expressed as:

$$O_{(x,y)} = \sum_{i=0}^{k-1}\sum_{j=0}^{k-1} I_{\left(x-i+\frac{k-1}{2},\, y-j+\frac{k-1}{2}\right)} \times W_{(i,j)} \qquad (1)$$

In image and video processing, isotropic kernels are commonly used. This feature is generally known as the symmetry property of the computational operations; a kernel with the symmetry property has repetitive weight values in a window arranged in a symmetric pattern [10]. By considering the symmetry property in the kernel, the additions of the pixels in the four quadrants are performed first (thus, reducing the need for three multiplications); for an odd kernel, this is described by

$$O_{(x,y)} = \sum_{i=0}^{\frac{k-1}{2}-1} \sum_{j=0}^{\frac{k-1}{2}-1} I_{\left(x\pm i+\frac{k-1}{2}, y\pm j+\frac{k-1}{2}\right)} \times W_{(i,j)} + I_{(x,y)} \times W_{\left(\frac{k-1}{2},\frac{k-1}{2}\right)} \quad (2)$$

III. INEXACT 2D CONVOLTION

A. General Priciples

For many computing and signal processing applications, one of the most powerful and easily available features for trading off energy and computational complexity is the operand bit-width. This algorithm-level scheme allows a flexible adjustment for approximate computing, however it requires specific data requirements to be met for efficient execution. For the 2D convolution kernel, the coefficients located far from the center are typically small after quantization; so, they do not impact image quality as much as the coefficients near the center. This implies that for inexact computing, a lower bit-width can then be used for operations involving the far coefficients; however, this algorithm-level process requires a detailed analysis, because it impacts computational outcomes and processing quality. A 5×5 kernel is shown in Figure 1.

$$\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \times \frac{1}{273}$$

(a)

w_5	w_4	w_3	w_4	w_5
w_4	w_2	w_1	w_2	w_4
w_3	w_1	w_0	w_1	w_3
w_4	w_2	w_1	w_2	w_4
w_5	w_4	w_3	w_4	w_5

(b)

Figure 1 5×5 kernel

Consider also the Gaussian kernel as an example; the coefficients have different weights. The coefficients that are far from the center, are less significant than those near the center. The center coefficient has the largest weight and the corner coefficients have the least weights. The 2D convolution value is the sum of the products; thus, an approximation can be inserted in the weight multiplication. This manuscript utilizes a bit-width reduction process as an algorithm-based technique for approximate computing. Assume that the input pixel and the coefficient are quantized in n bits; k denotes the reduced bit-width, where $0 \leqslant k \leqslant n$. For example, for a gray scale, n=8; so, the set of permissible bit-widths is given by 8, 6, 4, 2, and 0 bits (where 0 bit means that no calculation is performed).

A greedy process is used in this paper to reduce the bit-width; the corner coefficients must be considered first. Consider an application requiring a target PSNR (given by T). Initially the corner coefficient (w_5) (i.e. the least sensitive) is changed; the bit-width is decreased from 8 to 6 and the image quality is checked for compliance with T. If the quality is still higher than T, the change is confirmed. Two sets of operands are now available: one with 8-bit width operands ($w_0 \ldots w_4$) and the other with 6-bit width operand (w_5). If the previous change is confirmed, then there are two candidates for bit-width reduction: w_4 and w_3. Only one candidate is selected at a time and the bit-width of the selected candidate is reduced by a level. After calculating the PSNR for the two cases, the one with the larger PSNR is selected. The bit-width is reduced by a single level at a time (8 to 6, 6 to 4, 4 to 2, 2 to 0) till the PSNR is just less than T. This process continues until a candidate (if it exists) is found to satisfy the image quality constraint and is applicable to coefficients in a so-called configuration for computing the convolution (as dealt in more detail next).

B. Configurations and error analysis

A configuration is found using the process presented previously for bit-width reduction. Let L_i (i=0, 1, …, 5) denote the bit-width of each coefficient after the bit-width reduction process; the so-called configuration is represented as (L0, L1, L2, L3, L4, L5). The corresponding weights are given by $(w_0, w_1, w_2, w_3, w_4, w_5)$. The error originates from the reduced (truncated) number of bits. Let N represent the total (original) bit-width; the bit-width of the currently considered configuration is denoted by L, i.e. the number of truncated bits is given by N-L. Since the inexact result is always smaller than the exact value, then the error difference is defined as $(exact\ result - inexact\ result)$ or

$$\text{inexact result} = \lfloor \text{exact result}/2^{N-L_i} \rfloor \times 2^{N-L_i} \quad (3)$$

where $\lfloor\ \rfloor$ denotes the maximum integer that is smaller than the value.

Consider then the error introduced by such a process. Let the distance between the exact and the truncated values for a single multiplication be denoted by D(i). The maximum distance (denoted as $D_{max}(i)$) is given by

$$\begin{aligned} D_{max}(i) &= w_i + 2w_i + \cdots + 2^{N-1-L_i} w_i \\ &= (2^{N-L_i} - 1) \times w_i \end{aligned} \quad (4)$$

The maximum error for one-pixel convolution is then

$$\begin{aligned} E_{max} &= \sum_{i=0}^{5} D_{max}(i) \times q(i) \\ &= \sum_{i=0}^{5} (2^{N-L_i} - 1) \times w_i \times q(i) \end{aligned} \quad (5)$$

where q(i) is the number of weights with the same value.

The average error can be estimated to be as nearly half of the maximum error, so

$$E_{avg} = \frac{1}{2} E_{max} \quad (6)$$

The worst case scenario of the PSNR occurs when $E_{max}(i)$, so

$$\begin{aligned} PSNR_{estimated\_worst} &= 20 \log_{10} \frac{2^N - 1}{E_{max}(i)} \\ &= 20 \log_{10} \frac{2^N - 1}{\sum_{i=0}^{5} (2^{N-L_i} - 1) \times w_i \times q(i)} \end{aligned} \quad (7)$$

The average PSNR value is then

$$\mathrm{PSNR}_{\mathrm{estimated\_avg}} = 20\log_{10}\frac{2^{\mathrm{N}}-1}{\mathrm{E}_{\mathrm{avg}}(\mathrm{i})} = 20\log_{10}\frac{2\times(2^{\mathrm{N}}-1)}{\sum_{\mathrm{i}=0}^{5}(2^{\mathrm{N}-\mathrm{L_i}}-1)\times \mathrm{w_i}\times \mathrm{q(i)}} \quad (8)$$

Let the target PSNR be denoted by T (whose value is set a-priori). The approximate configuration is determined based on two parameters: (a) The calculated PSNR of the configuration is greater than the target value T; (b) The total bit cost has the least value, where

$$\text{total bit cost} = \sum \mathrm{L_i}\times \mathrm{q(i)} \quad (9)$$

For the kernel in Figure 1 (a),

$$\text{total bits cost} = \mathrm{L_0} + 4(\mathrm{L_1}+\mathrm{L_2}+\mathrm{L_3}+\mathrm{L_5}) + 8\mathrm{L_4} \quad (10)$$

A figure of merit that considers the above parameters, is introduced to assess a configuration; this figure of merit is referred to as the PSNR/bit cost ratio and relates the accuracy of processing (in this case, the quality of an image) to the bit-width of the inexact implementation.

C. Configuration selection

Selection of the configuration must be accomplished next, i.e. to select the inexact configurations according to the combined figure of merit of the PSNR/bit cost ratio. The worst-case PSNR (as per (7)) and the total bits cost (as per (9)) of each configuration are first calculated; the ratio is then calculated. Those configurations that have a high PSNR and a low total bit cost, are the best candidates, i.e. the configurations with the higher PSNR/bits cost ratio are selected. For example, let the PSNR values be 25 dB, 35 dB and 45 dB for the kernel in Figure 1(b); the candidates in the ranges (24 dB - 26 dB), (34 dB – 36 dB) and (44 dB – 46 dB) are found first. Then, the PSNR/bit cost ratios for each candidate are listed; finally, the inexact configuration that has the highest PSNR/bit cost ratio, is selected. Hereafter they are denoted as configuration 1 (8,8,8,6,6,2), configuration 2 (8,6,6,6,4,2) and configuration 3 (6,6,6,4,2,0).

IV. Inexact Design Considerations

A. Power gating

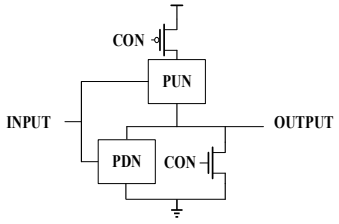

Figure 2 Power gating scheme

In the previous section, different inexact configurations for convolution based on error and pixel bit-width reduction were found. The inexact convolver must be configured in its hardware according to the requirement of image quality as established in the selection process; so, at least some parts of a computational module have to be turned on/off according to the desired configuration and the reduced bit-width. In this paper, power gating is used at circuit-level to accomplish bit-width reduction. In this scheme, two transistors are added to a generalized CMOS gate: a PMOS in series with a pull-up network (PUN) and an NMOS in parallel with a pull-down network (PDN) (Figure 2). The value of the control signal CON is determined by the target PSNR.

B. Inexact multiplier implementation

For convolution, the input data (weight and pixel) is assumed to be in sign-and-magnitude form. As for image convolution, the pixel value is always positive or 0, then, the sign-bit of the multiplication result is the same as the sign-bit of the weight. The signed multiplier is a conventional unsigned multiplier and only the magnitude of the two operands is calculated, because the sign-bit of the product is the same as the sign-bit of the weight. Once all results of the multipliers are calculated, an adder/subtractor is employed to accumulate the final convolution result.

A single multiplier accounts for the largest delay from the input (i.e. the operand) to the generation of the final result; the average values of the total power consumption and delay are shown in TABLE I. for different inexact multipliers (implemented at 32nm technology and simulated using a PTM). The complexity of an inexact multiplier is determined by the number of input bits (i.e. N) and the number of truncated bits (i.e. L). For the 5×5 isotropic kernel, the circuit complexity is dependent on the inexact configuration used; TABLE II. shows the complexity of the 3 inexact configurations as well as the exact convolver.

TABLE I. Average Delay And Power For Different Inexact Multipliers (N=8)

	Exact	***L*=6**	***L*=4**	***L*=2**
Delay	9.7ns	8.1ns	5.3ns	3.6ns
Power	94.05mW	60.54mW	37.70mW	23.80mW

TABLE II. Circuit complexity (in number of transistors) for the inexact configurations and the exact convolver (N=8)

Configuration	**Circuit Complexity**
Exact	4988
configuration 1	4028
configuration 2	3452
configuration 3	2674

V. Inexact Design Evaluation

A. Input error distribution of images.

The error of the proposed scheme depends on the lower bits, i.e. the truncated bits. To evaluate the error in the inputs, the value of the last m bits (truncated bits) is evaluated for all test images (taken from [11]). The mean and the variance for different numbers of truncated bits are listed in TABLE III. for the images of [11]. As m increases both the mean and the variance increase, so leading to larger values in all cases.

B. PSNR and MED.

TABLE IV. shows the estimated and the simulated PSNRs and MED [12] for several inexact configurations. The estimated PSNR value is calculated by using (7) and (8).

TABLE III. MEAN VALUE AND VARIANCE OF VALUES DUE TO TRUNCATED BITS FOR THE TEST IMAGES

Image	m=2		m=4		m=6	
	Var	Mean	Var	Mean	Var	Mean
Image 1	1.12	1.50	4.59	7.54	19.13	29.56
Image 2	1.12	1.48	4.61	7.46	16.77	31.86
Image 3	1.12	1.42	5.01	6.89	16.03	27.12
Image 4	1.11	1.53	4.24	6.93	13.85	32.12
Image 5	0.71	2.78	3.48	13.67	17.77	56.06
Image 6	1.12	1.50	4.62	7.50	19.01	32.99
Average	1.04	1.61	4.49	7.94	17.74	33.12

TABLE IV. ESTIMATED AND SIMULATED PSNR AND MED FOR INEXACT CONFIGURATIONS

	Inexact configuration 1		Inexact configuration 2		Inexact configuration 3	
	PSNR	MED	PSNR	MED	PSNR	MED
Estimated (worst case)	44.14	0.75	34.44	2.37	24.63	7.18
Estimated (avg case)	50.16	0.78	40.46	2.38	30.65	7.61
Image 1	49.22	0.70	40.38	2.19	30.81	7.72
Image 2	49.13	0.80	40.36	2.38	30.38	8.28
Image 3	49.54	1.67	40.87	4.47	30.16	13.27
Image 4	48.99	0.80	40.29	2.42	29.61	7.24
Image 5	44.26	0.78	34.94	2.39	25.45	7.36
Image 6	48.93	0.78	40.22	2.39	30.77	7.27
Average	48.67	0.78	39.87	2.38	29.95	7.61

The input error distribution is used to calculate the MED and compare the estimated with the simulated MED values. The estimate (calculated) value is given by

$$\mathrm{MED}_{\mathrm{calculated}} = \sum \mathrm{w_i} \times \mathrm{Mean_i} \times \mathrm{n} \qquad (11)$$

where *i* denotes the index of the kernel weights; *n* denotes the number of elements in the kernel with the same weight *i*. Figure 3 shows the difference between the calculated and the simulated MEDs for the three inexact configurations; in nearly all 6 cases (except 5), the calculated MED value is almost the same as the value found by simulation.

VI. CONCLUSION

This paper has presented the analysis and design of a convolver whose operation is based on approximate computing; approximate computing reduces circuit complexity and power consumption. Different techniques have been used when implementing approximate computing for a convolver design. Bit-width reduction has been utilized to assess the quality of the convolution results and the power consumption of the required hardware; power gating has been employed at circuit-level to reduce the bit-width. Truncation have also been employed; while they can be utilized in the entire inexact design, these techniques have been selectively used to meet the desired figures of merit in the inexact convolver.

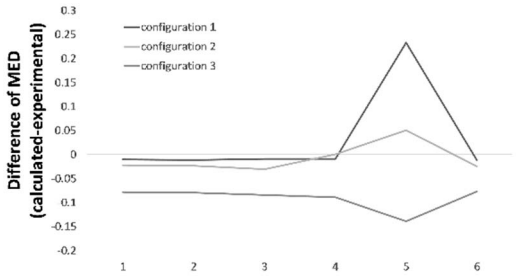

Figure 3 Difference between the estimated (calculated) and the simulated MEDs

An error analysis has been pursued to assess different figures of merit such as PSNR and MED. For image processing, the proposed scheme operates on a pixel basis of an image using different bit widths. In terms of image quality, the inexact configurations 1, 2 and 3 have average PSNRs of 48.68dB, 39.84dB, and 30.56dB respectively. In terms of power consumption, the same configurations achieve 30.6%, 51.5% and 64.3% reduction compared with an exact convolver. The design, the analysis and the simulation results show that the techniques utilized in the inexact processing of a convolver can operate in synergy, thus offering many design alternatives to meet different operational requirements.

REFERENCES

[1] Gonzalez, R., and Woods, R.: 'Digital image processing' (Prentice Hall, 2002, 2nd edn.)

[2] Bosi, B., Bois, G., and Savaria, Y.: 'Reconfigurable pipelined 2-D convolvers for fast digital signal processing', IEEE Trans. VLSI Syst., 1999, 7, (3) pp. 299–308

[3] J. Han; Orshansky, M., "Approximate computing: An emerging paradigm for energy-efficient design," Test Symposium (ETS), 2013 18th IEEE European , pp.1,6, 27-30 May 2013.

[4] Z. Yang, A. Jain, J. Liang, J. Han and F. Lombardi, "Approximate XOR/XNOR-based adders for inexact computing," IEEE Conf. on Nanotechnology, pp. 690-693, Beijing, August 2013.

[5] S.-L. Lu, "Speeding up processing with approximation circuits," Computer, vol. 37, no. 3, pp. 67-73, 2004.

[6] A.K. Verma, P. Brisk and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in Proc. DATE, pp. 1250-1255, 2008.

[7] J. Park, J. Choi, and K. Roy, "Dynamic bit-width adaptation in DCT: an approach to trade off image quality and computation energy," IEEE Trans. VLSI Systems, vol. 18, no. 5, pp. 787-793, May 2011.

[8] R. Hegde and N.R. Shanbhag, "Soft digital signal processing," IEEE Trans. VLSI Systems (TVLSI), 9(6):813–823, December, 2001.

[9] B. Shim and N.R. Shanbhag, "Energy-efficient soft error-tolerant digital signal processing," IEEE Trans. VLSI Systems, 14(4):336-348, 2006.

[10] Perri, S.; Corsonello, P., "VLSI implementations of efficient isotropic flexible 2D convolvers," in Circuits, Devices & Systems, IET , vol.1, no.4, pp.263-269, August 2007.

[11] The USC-SIPI Image Database — A large collection of standard test images: http://sipi.usc.edu/database/.

[12] J. Liang, J. Han and F. Lombardi, "New Metrics for the Reliability of Approximate and Probabilistic Adders," IEEE Transactions on Computers, vol. 62, no. 9, pp. 1760-1771, 2013.