Matrix Multiplication by an Inexact Systolic Array

Ke Chen, Fabrizio Lombardi
Department of Electrical and Computer Engineering
Northeastern University
Boston, MA USA 02115
chen.ke1@husky.neu.edu, lombardi@ece.neu.edu

Jie Han
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Canada
jhan8@ualberta.ca

Abstract— **Different schemes for approximate computing of matrix multiplication (MM) in systolic arrays are presented in this manuscript. Inexact full adder cells are utilized in a processing element (PE) for the Baugh-Wooley multiplier and/or the final adder as circuits implementing the two computational steps required for MM. An extensive analysis and simulation-based assessment of three inexact schemes for the PE are pursued with respect to circuit level performance (such as delay, power consumption and number of transistors) and figures of merit of approximate computing (such as the error distance). The execution of MM in each PE results in an inexact computation affecting only the outputs of the same columns, so the extension of inexact computation to a systolic array can also be performed with very limited error. The discrete cosine transform as application of the proposed inexact systolic arrays, is evaluated; simulation results show that the proposed inexact array is very effective, incurring in a small error.**

Index Terms— **Approximate computing, inexact design, systolic array, matrix multiplication**

I. INTRODUCTION

Power dissipation has become of paramount concern when designing high performance computing systems. Computing systems are increasingly embedded and mobile, so a growing set of applications, related to recognition, data mining and synthesis (RMS), requires energy efficiency in processing. For many RMS applications, the exact result may not always be required; hence, an approximate (or inexact) result is often viable [1]. Approximate computing has been recently advocated as a possible framework by which accuracy of results can be traded off for improvement in other metrics, such as power dissipation, delay and circuit complexity.

Matrix multiplication (MM) is one of the essential operations in various fields of science, engineering and technology, such as signal and image processing, systems theory, statistical and numerical analysis. It has a high complexity, but its computational regularity can be utilized to reduce it when real time constraints must be met [2]. So, the design and implementation of a high speed, area efficient matrix multiplier is highly desirable [3]. [4] has proposed a systolic array (SA) scheme for square matrix multiplication.

A SA consists of modular processing elements (PEs) with homogeneous interconnections, hence high parallelism is achieved by replication [5]. A systolic array is controlled and synchronized by a global clock with a cycle of fixed length, such that data is periodically computed and passed through the array. A complexity of $O(N^3)$ operations is required to perform an N × N MM on a sequential processor i.e. N^3 multiplications, $N\times2(N - 1)$ additions and N^2 operational cycles. SA reduces the time complexity of MM, because only $(2N + 1)$ operational cycles are required using an N × N systolic array [3].

This paper presents several schemes for approximate computing of matrix multiplication in a systolic array. In this paper, the processing element (PE) consists of a Baugh-Wooley multiplier and a final adder; inexact full adder cells are utilized in both, or either of these circuits for the two computational steps required in MM. An extensive analysis and simulation-based assessment of three inexact schemes for the PE are pursued with respect to circuit level performance (such as delay, power consumption and number of transistors) and approximate computing figures of merit (such as the error distance). The execution of MM in a PE results in an inexact computation affecting only the outputs in the same column. The extension of inexact computation to the SA is presented. The discrete cosine transform as application of the proposed inexact systolic arrays is evaluated; simulation results show that the proposed inexact array is very effective, because it has a small error.

II. SYSTOLIC MATRIX MULTIPLICATION

In the SA of [6] [7], the PEs are interconnected in a two-dimensional array (Figure 1) [6] [7] with communication between PEs occurring in cycles; so, the ith column incurs in a delay of i-1 cycles. Each PE stores the accumulated sum P(i,j). After 2N+1 operational cycles, the stored P(i,j) value forms the resulting product matrix P.

III. EXACT PROCESSING ELEMENT (PE)

Consider the multiplication of two N×N matrices A and B, so

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1N} \\ P_{21} & P_{22} & \cdots & P_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1} & P_{N2} & \cdots & P_{NN} \end{bmatrix} = A \times B$$

$$= \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NN} \end{bmatrix} \times \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1N} \\ B_{21} & B_{22} & \cdots & B_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N1} & B_{N2} & \cdots & B_{NN} \end{bmatrix} \quad (1)$$

Where

$$P_{ij} = \sum_{k=1}^{N} A_{ik} \cdot B_{kj} \quad (2)$$

Thus, the cumulative sum (i.e. $\sum_{k=1}^{N} A_{ik} \cdot B_{kj}$) needs to be calculated; so, a PE needs to have two functions: (1) calculate the product of the two input binary operands (A_{ik} and B_{kj}); (2) calculate the cumulative sum of the products. Therefore in the execution of this algorithm, a PE consists of an adder and a multiplier with signed operands.

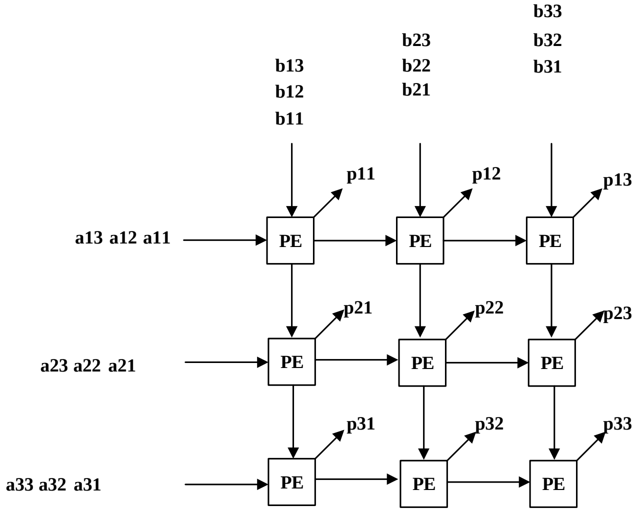


Figure 1 Systolic array for Matrix Multiplication

Consider first the signed multiplication. The Baugh-Wooley algorithm is utilized in this paper [8], i.e. the multiplier for calculating the product of the two n-bit binary input operands A_{ik} = (a_{n-1}, . . . , a_1,a_0) and B_{kj} = (b_{n-1}, . . . ,b_1,b_0). The result is represented by a 2n-bit output value M =$A_{ik} \cdot B_{kj}$ =(m_{2n-1}, . . . ,m_1,m_0). Two 2's complement numbers are used in the signed multiplication [9]; so, a_{n-1} and b_{n-1} are the sign bits.

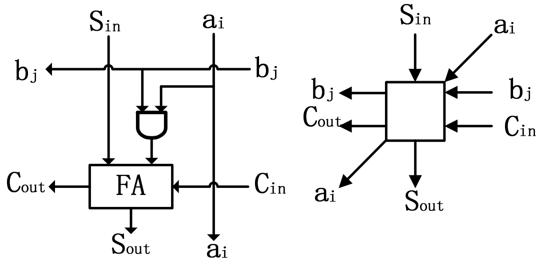
$$A = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i \quad (3)$$

$$B = -b_{n-1}2^{n-1} + \sum_{j=0}^{n-2} b_j 2^j \quad (4)$$

$$M = A \times B = a_{n-1}b_{n-1}2^{2n-2} + \sum_{i=0}^{n-2}\sum_{j=0}^{n-2} a_i\, b_j 2^{i+j}$$

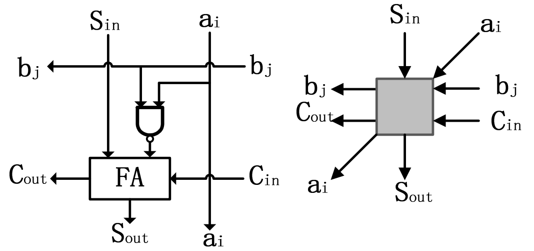
$$+2^{n-1}\sum_{i=0}^{n-2} \overline{b_{n-1}a_i}2^i + 2^{n-1}\sum_{j=0}^{n-2} \overline{a_{n-1}b_j}2^j + -2^{2n-1} + 2^n \quad (5)$$

In the computation in a 3-bit Baugh-Wooley multiplier, each bit of the product m_i is equal to the cumulative sum of the partial products ($a_k b_{i-k}$, i.e. $m_i = \sum_{k=0}^{i} a_k b_{i-k}$) for any $i = 0, 1, 2, 3, 4$. The most significant bit m_5 is the carry out bit.



(a) (b)

Figure 2 Exact AND partial product cell (PPC): (a) Logic circuit; (b) symbol



(a) (b)

Figure 3 Exact NAND partial product cell: (a) Logic circuit; (b) symbol

There are two types of basic operation in the partial product: AND and NAND. Thus, two types of a partial product cell (PPC) are needed. Figure 2 shows the first type of PPC; it computes a single bit multiplication of a_i and b_j using an AND-gate. The result is then added using a full adder (FA) cell with the previous sum output S_{in} from a cell located above, and the previous carry output C_{in} from a cell to the right to generate the final S_{out} and C_{out}..The second type of PPC is shown in Figure 3; it differs only in the gate (now a NAND) from the scheme of Figure 2. In this paper, the final sum is computed by a ripple carry adder (as an exact full adder (FA) cell, the circuit of [11] consisting of 10 transistors is used in this manuscript).

In the matrix multiplication algorithm, a PE calculates $P_{ij} = \sum_{k=1}^{N} A_{ik} \cdot B_{kj}$. An exact PE for matrix multiplication with 3-bit input operands is shown in Figure 4. It utilizes a Baugh-Wooley multiplier and a ripple carry adder. m_{in} is the calculated result of the last cycle. m_{out} is the current calculated result. The 3 bit-operands are A_{ik} (a2, a1, a0) and B_{kj} (b2, b1, b0), where a0 and b0 are the least significant bits. In this case, $A_{ik} \times B_{kj}$ could lead to a 6 bit result, shown by the dotted box in Figure 4. P_{ij} is the cumulative sum of the products. So in general for an N × N matrix multiplication with 3-bit input operands, the maximum value of P_{ij} is a 6-bit product and few additional upper bits (as shown in the dotted box in Figure 4) are required. For example in a 2×2 matrix multiplication, N =2 and P_{ij} requires 8 bits.

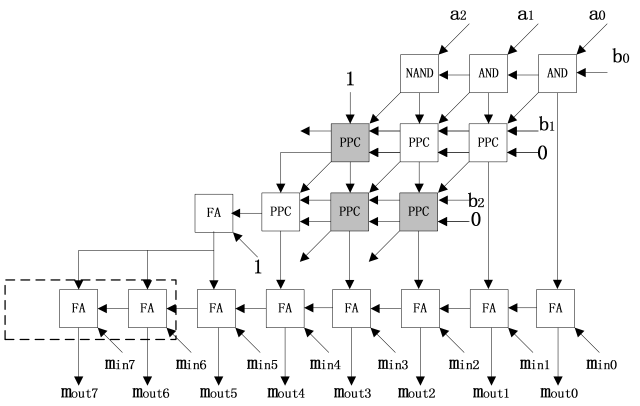


Figure 4 Exact PE for systolic array matrix multiplication with 3-bit input operands

Therefore, a PE is made of two types of PPCs, AND/NAND gates and a ripple carry adder; its complexity in the number of transistors is determined by all of its circuits (Table I). There are n AND gates and the number of full adder cells is at least 2n+1.

Table I Circuit complexity of the Exact PE

Cell type	Number of transistors per cell	Number of cells
AND gate	6	n-1
NAND gate	4	1
AND PPC	12	n^2-3n+3
NAND PPC	14	2n-3
FA	10	>2n+1

Hence, the total number of transistors in an exact PE is given by

$$T \geq 12n^2 + 20n \qquad (6)$$

IV. Inexact Processing Element (PE)

In this paper, the approximate operation is introduced in the adder cells; the so-called AXA inexact design proposed in [12] is used in place of an exact full adder cell. Figure 5 shows the AXA circuit design resulting in an inexact Sum; for this cell, the Sum output is accurate for 6 out of the 8 possible combinations, while C_{out} is accurate for all 8 input combinations. The transistor count is now 7; the total error distance of this design is 2.

$$\mathrm{Sum} = \overline{((X \oplus Y)\mathrm{Cin} + XY)} \qquad (7)$$

$$\mathrm{Cout} = (X \oplus Y)\mathrm{Cin} + XY \qquad (8)$$

So, an AXA cell can be used to replace an exact full adder cell (denoted by FA); exact full adder cells are utilized in two circuits of a PE for systolic matrix multiplication: (a) In a NAND/AND PPC cell as shown previously in Figure 2 and Figure 3. (b) In the final full adder (of width 2N). Let an inexact PPC be denoted as a black shaded box and shown in Figure 6. In this inexact PE, inexact circuits for the adder cells are utilized for processing the k least significant bits, so inexact computation affects only the outputs in the same columns.

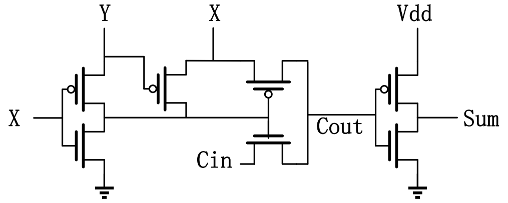


Figure 5 Approximate XOR based adder cell (AXA)

Three schemes are investigated in this paper to introduce inexact computation in a PE.

- Inexact circuits in PPC(s) and adder cell(s) (Scheme A)
- Inexact circuits in PPC(s) only (Scheme B)
- Inexact circuits in the adder cell(s) only (Scheme C)

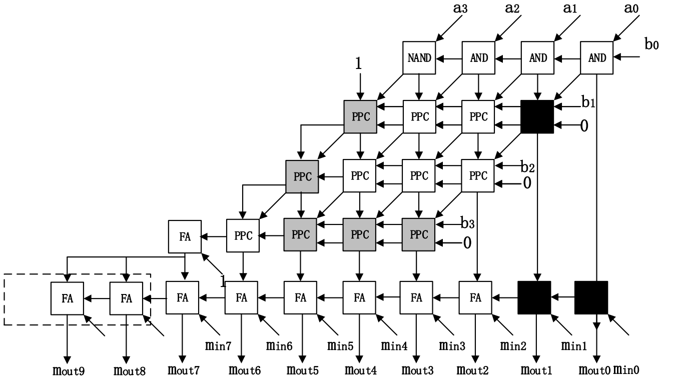


Figure 6 Inexact PE for systolic array matrix multiplication for 4-bit operands with 2 inexact columns (Scheme A)

Figure 6 shows the proposed inexact schemes A for a PE; 4-bit operands and 2 inexact columns are assumed. Shaded blocks identify the circuits with inexact adder cells, i.e. either PPC, or the final adder (identified by the FA cells) as affecting the multiplication or sum operations.

In Scheme B (C) inexact computation occurs only in the multiplier (adder). When inexact computation is considered only for a PPC cell, the number of inexact columns (i.e. k) cannot be equal to 1, because the last bit for the product does not include a PPC cell, i.e. it utilizes an AND gate only.

V. Simulation Results

In this paper all circuits are simulated and evaluated using HSPICE at the 32nm technology node as found in the corresponding Predictive Technology Model (PTM).

Delay: The delay of a single PE unit is defined as the latency from the operands as inputs to the generation of the final result; an exhaustive simulation is performed and the average delays are shown in Figure 7 for different numbers of bits in the input operands and inexact columns. The exact PE has the largest delay. It also has the highest complexity; the delay decreases by decreasing the number of inexact bits.

Power: The total power dissipation (average value) for a single PE is plotted in Figure 8. Scheme A (C) has the lowest (highest) average power dissipation)

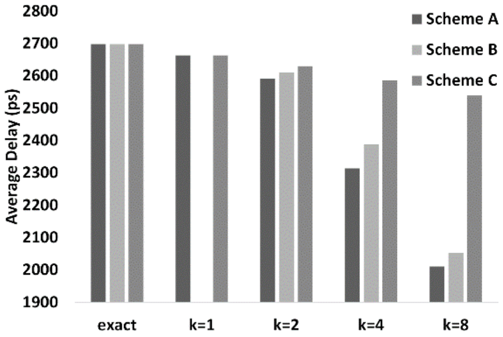


Figure 7 Average Delay of inexact Schemes

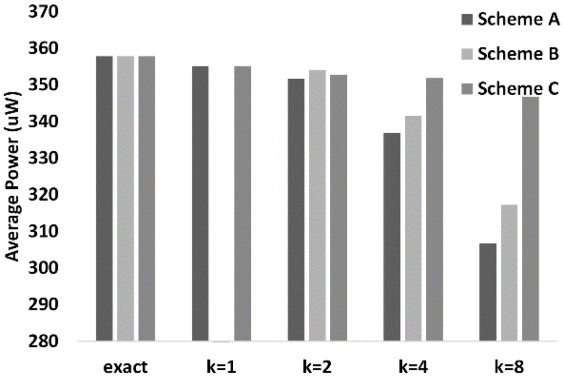
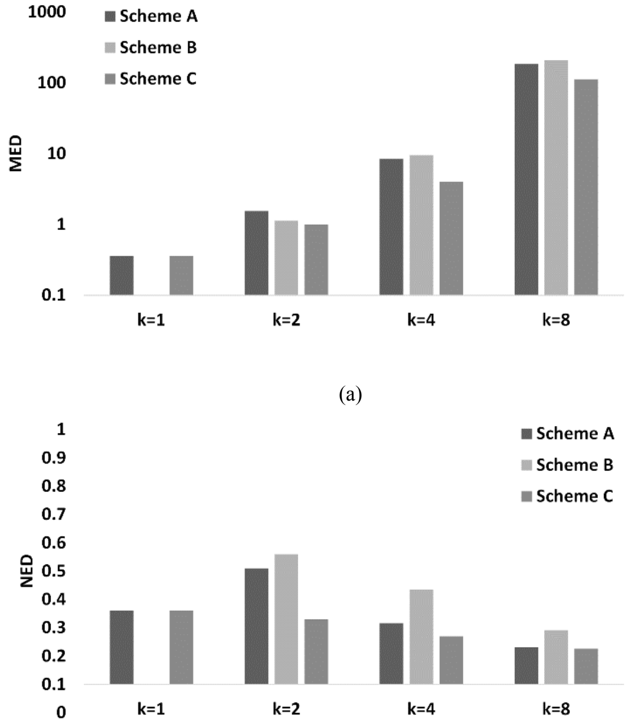


Figure 8 Average Power of inexact Schemes

MED and NED: [13] has compared adders and proposed several new metrics for evaluating approximate and probabilistic adders with respect to unified figures of merit for design assessment in inexact computing applications. For each input to a circuit, the error distance (ED) is defined as the arithmetic distance between an erroneous and the correct outputs [13]. The mean error distance (MED) and normalized error distance (NED) are also proposed by considering the averaging effect of multiple inputs and the normalization of multiple-bit adders. The NED is nearly invariant with the size of an implementation and is therefore useful in the reliability assessment of a specific design. Exhaustive simulation is executed to find the MED and NED for the 3 bit and 4 bit cases; for the 8 bit cases, 1 million input combinations are generated for calculating the MED and NED. From the simulation results (Figure 9), several conclusions can be drawn. (1) The MED increases with an increase of k. (2) For the proposed inexact schemes, Scheme C (A) has the smallest (largest) values of MED and NED. As the multiplier accumulates an error by adding the partial products, the multiplier is the largest contributor to the error distance.

From the simulation results (Figure 9), several conclusions can be drawn. (1) The MED increases with an increase of k. (2) For the proposed inexact schemes, Scheme C (A) has the smallest (largest) values of MED and NED. As the multiplier accumulates an error by adding the partial products, the multiplier is the largest contributor to the error distance.

Complexity: The complexity of an inexact PE employing AXA is determined by the number of input bits (i.e. n) and the number of inexact columns (i.e. k); the number of exact FA cells is determined by the matrix size. Table II shows the results; 2n+1-k is the least number for N=2.



(a)

(b)

Figure 9 Error distance of inexact PE for different number of inexact columns: (a) MED; (b) NED

.

Table II Circuit complexity of inexact PE with AXA (Scheme A)

Cell type	Number of transistors per cell	Number of cells
AND gate	6	n-1
NAND gate	4	1
Exact AND PPC	12	n^2-3n+3-$\sum_{i=1}^{k-1} i$
Exact NAND PPC	14	2n-3
Inexact PPC	9	$\sum_{i=1}^{k-1} i$
Exact FA	10	>2n+1-k
Inexact FA	7	k

The total number of transistors in an inexact PE under Scheme A is given by

$$T_A \geq 12n^2 + 20n - 3\sum_{i=1}^{k} i. \qquad (9)$$

For the other two proposed schemes, the total transistor counts are given as follows

$$T_B \geq 12n^2 + 20n - 3\sum_{i=1}^{k-1} i, \qquad (10)$$

$$T_C \geq 12n^2 + 20n - 3k. \qquad (11)$$

1) *Inexact Systolic Array Simulation :* Next, systolic arrays are simulated for 32 × 32 matrix multiplication. A different number of inexact columns (as discussed previously) is present in each PE and each PE generates a final output as an element of the resulting product matrix. The delay of the array is the same as the single PE, because all PEs are enabled simultaneously after each cycle begins. The average MED and power consumption are established for the systolic array and evaluated in this section. A difference matrix is calculated for measuring the average MED; each element in the difference matrix is given by the difference of the corresponding elements in the exact and the inexact matrices. The average MED is the average value of every element in the difference matrix.i.e.

$$\mathrm{MED_{avg}} = \frac{\sum[|P_{eaxct}(i,j) - P_{ineaxct}(i,j)|]}{N \times N} \qquad (12)$$

The results for the average MED are presented in Figure 10 using PEs of 8 bits as input operands for matrix multiplication. There is no substantial difference in MED in the proposed schemes, however at a higher value of matrix size N, the MED yields a higher value too. The results for power dissipation under the same conditions are reported in Figure 11.

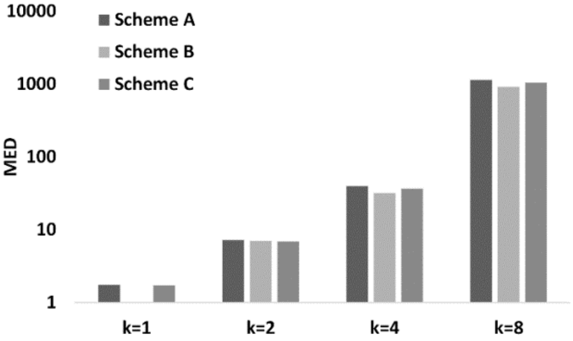


Figure 10 Average MED for matrix multiplication

VI. Application: DCT

Next, the application of the proposed schemes of an inexact systolic array to Discrete Cosine Transforms (DCT) is presented (note that all electrical based performance metrics have already been presented in previous sections, so only the results as related to the DCT computation are presented). DCT removes the correlation of image elements in the transform domain [14]; it is considered a quasi-optimal transform and has been widely applied in the fields of image and video compression coding. The three proposed inexact schemes are evaluated by considering different numbers of columns with respect to the transform of images and the loss of accuracy compared to an exact array. DCT matrix elements can be expressed in floating point format for which a significant number of multiplication and addition operations are required [15]. Under the condition of fixed word length, its accuracy is not very high, floating-point operations incur in a truncation error, and the drifting between encoding and decoding data causes a data mismatch in the decoder. An integer DCT uses integer instead of floating point numbers to fill the transformation matrix; the transformation core is an integer transform involving no floating-point calculation and achieving a high accuracy. The core transformation can be completed only with simple addition and shift operations, such that its computing complexity is significantly reduced [16] [17].

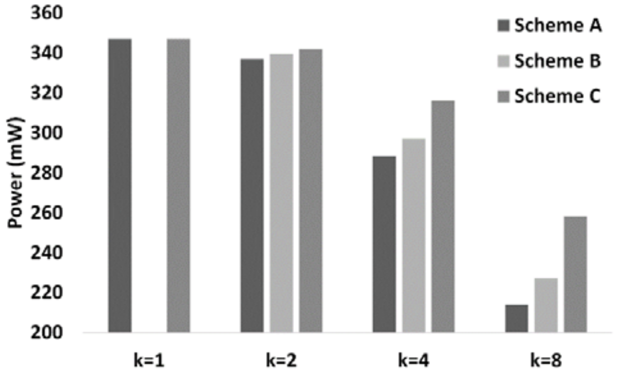


Figure 11 Average Power for matrix multiplication

In matrix notation, the discrete two-dimensional radix-8 DCT is given by Y = P·X·P^T , where X is the 8x8 input image frame and Y is the transformed matrix. The transformation matrix P is given as

$$P = \begin{bmatrix} a & a & a & a & a & a & a & a \\ b & d & e & g & -g & -e & -d & -b \\ c & f & -f & -c & -c & -f & f & c \\ d & -g & -b & -e & e & b & g & -d \\ a & -a & -a & a & a & -a & -a & a \\ e & -b & g & d & -d & -g & b & -e \\ f & -c & c & -f & -f & c & -c & f \\ g & -e & d & -b & b & -d & e & -g \end{bmatrix}$$

As described in [17], one of the most common transform matrix is

$$P1 = \begin{bmatrix} 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\ 89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 \\ 83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 \\ 75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 \\ 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 \\ 50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 \\ 36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 \\ 18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 \end{bmatrix}$$

The sample input images (in integer format) are shown in Figure 12; the Peak Signal to Noise Ratio (PSNR) is used as a metric. The simulation is performed using a 32 × 32 matrix multiplication; the results for this application (Tables III, IV and V) yield the following conclusions. 1) At the same k, Scheme C has the best PSNR, because the error is only introduced in the adder. (2) In all cases, the PSNR decreases when k increases; therefore, an increase in the number of inexact columns causes a more inexact final result for the approximate multiplier.

VII. Conclusion

This paper has presented the analysis of matrix multiplication using systolic arrays whose operation is based on approximate computing. Approximate computing reduces circuit complexity, delay and power consumption. Inexact cell circuits have been included in the design of a processing element (PE) of the array and three inexact schemes have been proposed.

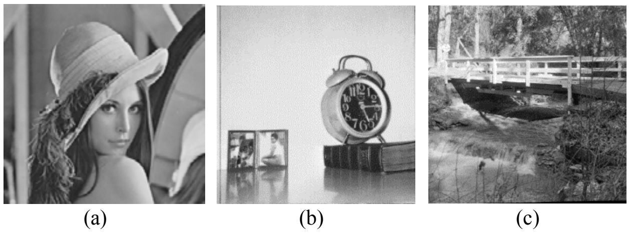


Figure 12 Integer DCT sample image: (a) Lena; (b) Clock; (c) Bridge.

Table III PSNRs for DCT of array with inexact PEs (Scheme A)

	Lena	Clock	Bridge
Exact DCT	61.34 dB	60.72dB	59.84dB
Inexact DCT,k=1	61.34 dB	60.72dB	59.84dB
Inexact DCT,k=2	61.31 dB	60.67dB	59.82dB
Inexact DCT,k=4	60.31 dB	60.14dB	58.89dB
Inexact DCT,k=8	42.99 dB	43.87dB	42.48dB

Table IV PSNRs for DCT of array with inexact PEs (Scheme B)

	Lena	Clock	Bridge
Exact DCT	61.34 dB	60.72dB	59.84dB
Inexact DCT,k=1	-	-	-
Inexact DCT,k=2	61.31 dB	60.68dB	59.82dB
Inexact DCT,k=4	60.42 dB	60.38dB	59.03dB
Inexact DCT,k=8	44.65 dB	44.78dB	44.28dB

Table V PSNRs for DCT of array with inexact PEs (Scheme C)

	Lena	Clock	Bridge
Exact DCT	61.34 dB	60.72dB	59.84dB
Inexact DCT,k=1	61.34 dB	60.72dB	59.84dB
Inexact DCT,k=2	61.33 dB	60.70dB	59.83dB
Inexact DCT,k=4	60.68 dB	60.57dB	59.34dB
Inexact DCT,k=8	46.36 dB	45.67dB	46.74dB

These schemes (A, B, C) introduce inexact computation in a PE cell(s) for either the multiplier, and/or the final adder. The PEs have been designed at nanometric feature sizes and simulated using HSPICE to assess different figures of merit such as delay, power dissipation and circuit complexity. n particular, the following conclusions are applicable. (i) An increase in the number of inexact columns (k) results in a decrease of power and delay; however, the error distance increases too. (ii) Scheme A (with approximate cells in both the multiplier and the final adder) has the smallest delay and consumes the least amount of power; however, the error distance is the largest. (iii) Scheme C (with approximate cells in the final adder only) has the shortest error distance, but it is the worst for the delay and power consumption.

Current research deals with new approximate designs of the multiplier and the adder of a PE [18] [19] as well as additional image processing applications.

REFERENCES

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

[2] Stojčev, M.K.; Milovanović, E.I; Marković, S.R.; Milovanović, IZ., "Synthesis of orthogonal systolic arrays for fault-tolerant matrix multiplication," Microelectronics Proceedings (MIEL), 2010 27th International Conference on , vol., no., pp.327,334, 16-19 May 2010

[3] Saha, P.; Banerjee, A; Bhattacharyya, P.; Dandapat, A, "Improved matrix multiplier design for high-speed digital signal processing applications," Circuits, Devices & Systems, IET , vol.8, no.1, pp.27,37, Jan. 2014

[4] H. T. Kung, C. E. Leiserson: Algorithms for VLSI processor arrays; in: C. Mead, L. Conway (eds.): Introduction to VLSI Systems; Addison-Wesley, 1979.

[5] Hsin-Chou Chi; Hsi-Che Tseng; Kun-Lin Tsai, "Design of timing-error-resilient systolic arrays for matrix multiplication," SoC Design Conference (ISOCC), 2011 International , vol., no., pp.219,222, 17-18 Nov. 2011

[6] J-C Tsay; P-Y Chang, "Some new designs of 2-D array for matrix multiplication and transitive closure," Parallel and Distributed Systems, IEEE Transactions on , vol.6, no.4, pp.351,362, Apr 1995

[7] Kumar, V.K.P.; Tsai, Y.-C., "Synthesizing optimal family of linear systolic arrays for matrix computations," Systolic Arrays, 1988., Proceedings of the International Conference on , vol., no., pp.51,60, 25-27 May 1988

[8] Baugh, Charles R.; Wooley, B.A, "A Two's Complement Parallel Array Multiplication Algorithm," Computers, IEEE Transactions on , vol.C-22, no.12, pp.1045,1047, Dec. 1973

[9] Weste, N. and Harris, D., “CMOS VLSI Design: A Circuits and Systems Perspective (4th Edition)”, 4th Edition Addison-Wesley, MA, 2011.

[10] Hatamian, M.; Cash, G.L., "A 70-MHz 8-bit×8-bit parallel pipelined multiplier in 2.5-μm CMOS," Solid-State Circuits, IEEE Journal of , vol.21, no.4, pp.505,513, Aug 1986

[11] J.-F Lin, Y.-T. Hwang, M.-H. Sheu and C.-C Ho, “A novel high-speed and energy efficient 10-transistor full adder design”, IEEE Trans. on Circuits and Systems-I: Regular Papers, Vol. 54, No.5, May 2007

[12] Z. Yang; Jain, A.; J. Liang; J. Han; Lombardi, F., "Approximate XOR/XNOR-based adders for inexact computing," Nanotechnology (IEEE-NANO), 2013 13th IEEE Conference on , vol., no., pp.690,693, 5-8 Aug. 2013

[13] 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. 1694 - 1704, 2013.

[14] Ahmed, N.; Natarajan, T.; Rao, K.R., "Discrete Cosine Transform," Computers, IEEE Transactions on , vol.C-23, no.1, pp.90,93, Jan. 1974

[15] Yeh, H.-G.; Yeh, H.-Y., "Implementation of the discrete Fourier transform on 2-dimensional systolic processors," Electronic Circuits and Systems, IEE Proceedings G , vol.134, no.4, pp.181,186, August 1987

[16] J. Wu; Y. Li, "A new type of integer DCT transform radix and its rapid algorithm," Electric Information and Control Engineering (ICEICE), 2011 International Conference on , vol., no., pp.1063,1066, 15-17 April 2011

[17] Meher, P.K.; S. Y. Park; Mohanty, B.K.; K. S. Lim; C. Yeo, "Efficient Integer DCT Architectures for HEVC," Circuits and Systems for Video Technology, IEEE Transactions on , vol.24, no.1, pp.168,178, Jan. 2014.

[18] A. Momeni, J. Han, P. Montusci and F. Lombardi, “Design and Analysis of Approximate Compressors for Multiplication,” IEEE Transactions on Computers, vol. 64, no 4, pp. 984-994, 2015.

[19] H. Jiang, J. Han and F. Lombardi, ”A Comparative Review and Evaluation of Approximate Adders,” Proc. ACM/IEEE Great Lakes Symposium on VLSI, pp. 343-348, Pittsburgh, May 2015 (with H. Jiang and J. Han)