SEERAD: A High Speed yet Energy-Efficient Rounding-based Approximate Divider

Reza Zendegani[1], Mehdi Kamal[1], Arash Fayyazi[1], Ali Afzali-Kusha[1], Saeed Safari[1], Massoud Pedram[2]
[1]School of Electrical and Computer Engineering, University of Tehran
[2]Department of Electrical Engineering, University of Southern California
{zendegani_reza, mehdikamal, a.fayyazi, afzali, saeed}@ut.ac.ir, pedram@usc.edu

***Abstract*— In this paper, a high speed yet energy-efficient approximate divider for error resilient applications is proposed. For the division operation, the divisor is rounded to a value with a specific form resulting in the transformation of the division operation to the multiplication one. The proposed approximate divider enjoys the flexibility of increasing the accuracy at the price of higher delay and hardware usage. The efficacy of the proposed approximate divider is evaluated in comparison to three different implementations of the SRT divider. The results show that the delay and energy consumption of the proposed approximate divider are, on average, 14 and 300 times smaller than those of the Radix-2 SRT with the carry-save reminder computation. Additionally, the effectiveness of the proposed approximate divider is studied in an image division operation performed in image processing applications. The results suggest the appropriateness of the proposed approximate divider for digital signal processing applications.**

I. Introduction

Demands for more computational speeds of digital systems are continuously increasing. These demands for mobile devices (as well as other non-mobile ones) are accompanied by the requirement of low power/energy consumptions. In computational systems including those performing digital signal processing, the speed and energy consumption of the arithmetic units have determining roles in the speed and energy consumption of the whole system. Among the four basic operations of Add, Sub, Mul, and Div, the last one has been used less frequently compared to the others [1]. This may be attributed to its low speed and high energy consumption features [1]. There are different techniques for performing the division operation where among them, the digit recurrence algorithms, represent the most common class. In this technique a single quotient-digit is produced at each computation step. In modern processors, the technique is commonly implemented using SRT division [1][2].

One approach to increase the performance and also reduce the power consumption of the arithmetic units is approximate computing. The idea of the approximate computing comes from the tolerance of the application to imprecise results. In other words, the approximated computing may be used in applications which are resilient to errors during the execution [3]. Examples of these applications belong to the categories of multimedia, artificial intelligence, wireless communication, and digital signal processing. In the approximate computing approach, performance and power/energy consumptions are improved at the price of accuracy providing a tradeoff between these parameters [3][4]. The approximation may applied in different design abstraction levels, from circuit to algorithm and software layers [5].

Many works focusing on the field of approximating arithmetic building blocks have been reported in the literature (see *e.g.*, [6][7]). They have mainly concentrated on approximate adders and multipliers with a few works on approximate dividers. In [8], two approximate dividers in two different levels of design abstraction (cell and array levels) were proposed. In the cell level, first, by exploiting the logic complexity reduction method, an approximate subtractor was proposed. Next, this subtractor was utilized in non-restoring division to obtain an approximate division operation. At the array level, by modifying the arrangements of the exact cells and also, changing the truncation scheme, an approximate array divider was realized. An approximate integer division by nine for image processing applications has been suggested in [9]. It worked based on dividing by eight instead, due to the fact that dividing by eight was translated to a simple shift right operation by 3 at the expense of some error.

In this paper, we propose a high-speed yet low-power/energy and area-efficient approximate divider. It has a high accuracy making it appropriate for efficient digital signal processing applications. The proposed divider is formed based on rounding the divisor to a specific form leading to transforming the division operation to some Shift and Add operations. This simplification provides us with reductions in delay and energy consumption of the divider at the cost of small accuracy loss. Also, by modifying the structure of the divider during the design time, the accuracy of the proposed divider may be improved which requires more hardware and leads to higher delay. The proposed divider is denoted by high Speed yet Energy-Efficient Rounding-based Approximate Divider (SEERAD). The rest of the paper is organized as follows. In Section II, the proposed division algorithm as well as its hardware implementation and accuracy are discussed. The SEERAD design parameters compared to the conventional dividers along with its efficacy in image processing applications are studied in Section III. Finally, the paper is concluded in Section IV.

II. Proposed Approximate Divider

A. SEERAD Algorithm

In the SEERAD algorithm, the dividing operation (*i.e.*, A/B) is simplified by rounding the divisor to its almost nearest number (*i.e.*, B_r) in the form of $2^{(K+L)}/D$ where K shows the position of the most significant 1 in the binary representation of B, and L and D are predefined constant integer numbers. Therefore, 2^K is the floor of logarithm of B (*i.e.*, $\lfloor \log B \rfloor$)

denoted by B_f. Hence, in this approach, the division operation is approximated by

$$\frac{A}{B} \simeq \frac{A}{B_r} = \frac{A}{\frac{2^{K+L}}{D}} = \frac{D \times A}{2^{K+L}} = \frac{D \times A}{2^{L} \times B_f} \quad (1)$$

The parameters D and L play the determining roles in the accuracy of the proposed division approach. To determine D and L, for each input bit length, an exhaustive search for the proper tuple of (L, D), should be performed. During the search, the value of A is changed from 0 to the maximum value and then the error of dividing A to B_r is compared to that of dividing by B. Then, the tuple which leads to the lowest mean relative error (MRE) is selected as the best tuple for this input bit length. If the lengths of both inputs are n, the D (L) is changed from 0 (0) to $2^n - 1$ ($n - 1$). Our study shows that in the cases where the input bit length is larger than 8, the tuple (D, L) found in the case of 8-bit leads to small error. Using this tuple for all the input bit lengths obviously yields a simpler hardware implementation with smaller delay at the cost of slightly larger inaccuracy for the SEERAD.

One approach to increase the accuracy of SEERAD is to consider different tuples of (L,D) for a given the input bit length. For this purpose, based on different combination of higher bits, we form different groups for the values of B. This approach, however, results in higher hardware complexity and delay for the SEERAD. The accuracy and corresponding overhead increase with increasing the number of the groups. Here, we suggest to determine the group that each divisor value belongs to it based on the stream of one and zero bits (from left to right) in the binary representation of the divisor. The length of the considered stream depends on the considered groups count and the first bit position of the stream is indicated based on the leftmost one in the binary representation of the divisor. Table I shows the groups for the B value and its corresponding D and L under different groups count where the accuracy level is equal to $log_2 |Group| + 1$.

TABLE I. B VALUES IN EACH GROUP AND ITS CORRESPONDING D AND L UNDER DIFFERENT GROUPS COUNT.

|Group|	Accuracy Level	Index	B	D	L
1	1	0	0 ... 01X X	5	3
2	2	0	0 ... 010X X	12	4
		1	0 ... 011X X	9	
4	3	0	0 ... 0100X X	28	5
		1	0 ... 0101X X	24	
		2	0 ... 0110X X	20	
		3	0 ... 0111X X	17	
8	4	0	0 ... 01000X X	120	7
		1	0 ... 01001X X	108	
		2	0 ... 01010X X	97	
		3	0 ... 01011X X	88	
		4	0 ... 01100X X	82	
		5	0 ... 01101X X	76	
		6	0 ... 01110X X	70	
		7	0 ... 01111X X	66	

In each group, during the exhaustive search, the value for D is selected such that the MRE is minimum and has the lowest nonzero terms in the polynomial representation of $\sum a_i 2^i$ such that $a_i \in \{-1,0,1\}$. On the other hand, to reach a simpler hardware implementation, and hence, smaller delay, we select one L value for all the groups in each accuracy level. It should be noted that this approach does not increase the error of approximate division operation using B_r. To see the accuracy of the approximate divider, as an example, the MRE value of the SEERAD in the case of the 4th accuracy level for different bit lengths has been plotted in Figure 1. As the figure reveals, the error is almost constant for the input lengths larger than 8 bits. The accuracy of the SEERAD will be studied in more details in Subsection II.C.

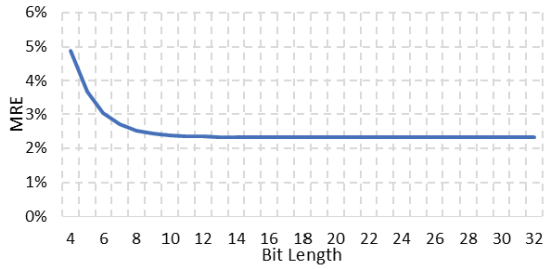

Figure 1. MRE of the SEERAD in the case of the 4th accuracy level under different bit lengths using D and L values given in Table I.

B. Hardware Implementation of SEERAD

Based on the above discussion, the block diagram of the hardware implementation of the SEERAD is given in Figure 2. In this work, we assume that n-bit inputs of the SEERAD are in 2's complement format. Hence, in the first stage (*Sign Detector* block), the signs of the dividend and divisor are determined. In the next stage, B_f is determined in the *Rounding* block which calculates the floor of B in the form of 2^n. Figure 3 shows the gate level implementation of this block.

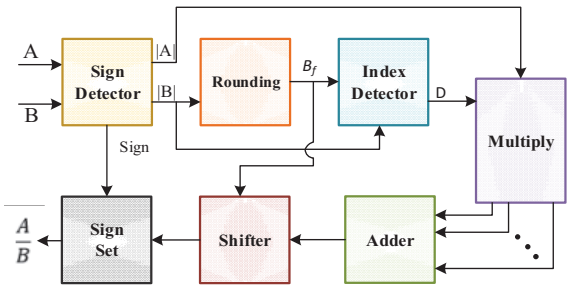

Figure 2. The block diagram of the hardware implementation of SEERAD.

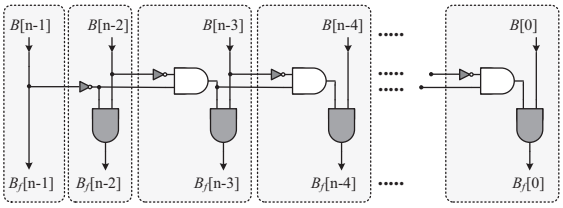

Figure 3. The gate level implementation of the *Rounding* block.

The value of B_f is used in the *Index Detector* block to find the index of the input B in the considered accuracy level to determine the D value. Using the B values in Table I, as an

example, the following equation may be utilized to obtain the index of the group that the B belongs to in the case of 3th accuracy level.

$$\begin{aligned} Index_0 &= \sum_{i=2}^{n-1} B_r[i].\overline{B[i-1]}.\overline{B[i-2]} \\ Index_1 &= \sum_{i=2}^{n-1} B_r[i].B[i-1].\overline{B[i-2]} \\ Index_2 &= \sum_{i=2}^{n-1} B_r[i].\overline{B[i-1]}.B[i-2] \\ Index_3 &= \sum_{i=2}^{n-1} B_r[i].B[i-1].B[i-2] \end{aligned} \quad (2)$$

It should be noted, as mentioned before, to simplify the hardware of the SEERAD, the L value is considered the same for all the indices.

Having determined the parameter D, in the block *Multiply*, $|A| \times D$ is calculated through shift to left operation. Based on the D values given in Table I, the number of shift units in the first, second, third and fourth accuracy levels are 2, 2, 2, and 3, respectively. The outputs of these shifter units are summed in the block *Adder*. It is worth to mention that the outputs bit length of the *Multiply* and *Adder* blocks are $2n$ bits.

Next, the output of the *Adder* block is utilized by the block *Shifter* to calculate $(D \times A)/(2^L \times B_f)$. This block shifts to right its input by $K + L$ bits. The shift to right, for implementing the division by 2^K, is implemented by a barrel shifter. In this case, no bits are shift out while the least significant bits are considered as the fractional part of the result. Hence, the output width of this block is $2n + L$ bits where its n bits belong to the integer part and $n + L$ bits represent the fractional part of the result.

Finally, the sign of the result is determined based on the sign of the inputs. For this, only in the case where one of the inputs is negative, the *Sign Set* block should complement the result. Also, in the case of unsigned division operation, the *Sign Detector* and *Sign Set* blocks do not need to be utilized and may be eliminated from the SEERAD.

C. Accuracy of SEERAD

The error of the SEERAD approach originates from the divisor rounding, and hence, may be expressed as

$$\begin{aligned} \text{error}(\text{A},\text{B}) &= 1 - \frac{\left(\frac{A}{B}\right)_{\text{approximate}}}{\left(\frac{A}{B}\right)_{exact}} \\ &= 1 - \frac{B}{B_r} = 1 - \frac{2^k + \sum_{i=0}^{k-1} a_i 2^i}{\frac{2^k 2^L}{D}} \\ &= 1 - \tfrac{D}{2^L}\left(1 + \textstyle\sum_{i=0}^{k-1} \frac{a_i 2^i}{2^k}\right) \end{aligned} \quad (3)$$

Hence, as we expected, the error of SEERAD depends on D and L. Now, by deriving the error function and extracting the stationary points, the maximum of the error is obtained from

$$Max\{\text{error}(\text{A},\text{B})\} = 1 - \frac{D_{max}}{2^L} \quad (4)$$

where D_{max} is the maximum value of D in the considered accuracy level. Using this discussion, the maximum error and the mean relative error (MRE) of the SEERAD under different bit lengths calculated and reported in Table II. It should be noted that since we consider the same D and L values for all the bit lengths, the maximum error is almost independent from the bit length. As is expected, by increasing the accuracy level, the error of the SEERAD is decreased. Also, as the bit length enlarges, the error either remains the same or becomes slightly smaller.

TABLE II. MAXIMUM ERROR AND THE MRE OF THE SEERAD UNDER DIFFERENT BIT LENGTHS

		MRE		
Accuracy Level	Maximum error	8 bits	16 bits	32 bits
1	37.5%	16.55%	16.25%	16.25%
2	25.0%	9.15%	8.77%	8.77%
4	12.5%	4.66%	4.55%	4.55%
8	6.25%	2.42%	2.20%	2.20%

III. RESULTS AND DISCUSSION

A. Hardware Implementation

To assess the efficacy of the proposed divider in different accuracy levels compared to those of the conventional exact dividers, we implemented 32-bit signed and unsinged SEERAD in four accuracy levels, and also three different implementations of the SRT division. The three implementations of the SRT division were Radix-2, Radix-2 with Carry-Save computation of the reminder (*i.e.*, Radix-2 CS), and Radix-4 [10]. It should be noted that all of the considered dividers were implemented using combinational circuits. The circuits were synthesized by Synopsys Design Compiler in a 45 nm technology [11]. The delay, power, area, energy, EDP (Energy-Delay-Product) and PDA (Power-Delay-Area product) of the dividers are reported in Table III. The results reveal that all the design parameters of the proposed divider in all the accuracy levels outperform compared to those of the SRT dividers. It should be noted that compared with the other basic operations, the delay, power, and area of the unsigned SEERAD, on average, are about 50%, 81%, and 0.58% smaller than those of the exact Wallace-tree multiplier.

To simultaneously evaluate the accuracy and design parameters of the SEERAD, we have plotted the normalized (based on the maximum value in each series) of the inverse of multiplying the MRE by different design parameters (*i.e.*, delay, power, area, energy, EDP, PDA) of 32-bit unsigned SEERAD in Figure 4. Note that in this figure, the larger the value is, the better the design is. As the results show, for the cases where the delay is important, the SEERAD implementation in the 4th level accuracy is the best. If either power or area is the important design parameter, the 3rd accuracy level is the better design, while if energy, EDP, or PDA, the 1st accuracy level implementation of the SEERAD is preferred.

TABLE III. DELAY, POWER, AREA, ENERGY, EDP AND PDA OF IMPLEMENTED DIVIDERS.

Architecture	Accuracy Level	Delay (ns)	Power (mw)	Area (μm^2)	Energy (pJ)	EDP (ns × pJ)	PDA ($mW \times ns \times \mu m^2$)
Unsigned SEERAD	1	0.61	0.64	1343	0.39	0.24	524
	2	0.7	1.18	2445	0.83	0.58	2020
	3	0.8	2.117	4378	1.69	1.35	7415
	4	1.07	7.53	12451	8.06	8.62	100319
Signed SEERAD	1	0.76	0.99	1537	0.75	0.57	1156
	2	0.87	1.69	2684	1.47	1.28	3946
	3	0.92	3.27	4868	3.01	2.77	14645
	4	1.27	9.72	12840	12.34	15.68	158502
SRT Radix-2		19.61	60.88	28691	1193.86	23411.53	34252945
SRT Radix-2 CS		11.3	31.07	15863	351.09	3967.33	5569357
SRT Radix-4		17.63	54.47	25051	960.31	16930.20	24056628

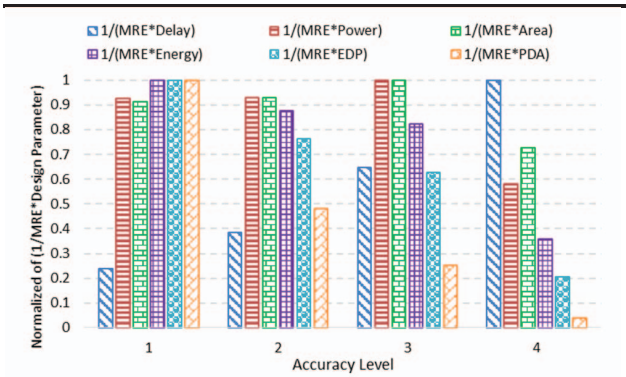

Figure 4. The inverse of multiplying MRE by different design parameters (normalized) for the 32-bit unsigned SEERAD versus the accuracy level.

B. Image Processing Application

To study the effectiveness of the proposed divider in image processing applications, we have utilized the SEERAD in image division operation. In this operation, each pixel of the output image is the quotient between the two corresponding pixel values from the input images [12]. This operation is used in detecting changes in a sequence of images. For this study, we performed the operation on each two frame sequence of Bus, Miss America, and Football video benchmarks. Only 40 frames from each benchmark were considered. For each benchmark, the division operation was performed by utilizing both the exact division and the SEERAD under four accuracy levels. The results which are given in Table IV include the maximum, minimum, and average of the PSNR of the output of the division operation. The PSNR calculated based on comparing the result of the SEERAD compared to that of the exact division. As the accuracy level is increased, the quality of the division operation improves. In the worst-case, the minimum PSNR belongs to the Bus benchmark in the 1st accuracy level of the SEERAD where the PSNR is 56dB. On the other hand, in case of Football benchmark, the highest PSRN of 92dB occurred when the SEERAD at the 4th level accuracy was utilized.

TABLE IV. MAXIMUM (MAX), MINIMUM (MIN), AND AVERAGE (AVG) OF THE PSNR (DB) OF THE SEERAD IN DIVISION OPERATION FOR DIFFERENT BENCHMARKS UNDER DIFFERENT ACCURACY LEVELS.

	Accuracy Level											
	1			2			3			4		
	max	min	avg	max	min	avg	max	min	avg	max	min	avg
Football	76	66	75	81	69	79	86	75	85	92	81	91
Miss America	73	73	72	76	76	76	82	82	82	88	88	88
Bus	66	56	62	70	59	66	77	65	72	82	71	78

IV. CONCLUSION

In this paper, a high-speed and energy efficient approximate divider called SEERAD was proposed. It was based on rounding the divisor to the nearest number in a specific form. This transformation simplified the division operation to some add and shift operations. Also, the proposed divider provided the flexibility for the designer to increase the accuracy at the price of higher delay and hardware usage. The efficiencies of different implementations of the SEERAD were assessed by comparing their delays, powers, and areas with those of accurate SRT dividers. The comparison showed that the proposed divider reduced delay, power (energy) metrics considerably compared to those of the SRT dividers. More specifically, the delay and energy consumption of the proposed approximate divider were, on average, 14 and 300 times smaller than those of the Radix-2 SRT with the carry-save reminder computation. Finally, the efficiency of the proposed divider was demonstrated in one image processing application.

REFERENCES

[1] W. Liu and A. Nannarelli, "Power Efficient Division and Square Root Unit," *IEEE Transactions on Computers*, vol. 61, no. 8, pp. 1059-1070, 2012.

[2] E. Antelo, T. Lang, P. Montuschi, and A. Nannarelli, "Digit-Recurrence Dividers with Reduced Logical Depth," *IEEE Transactions on Computers*, vol. 54, no. 7, pp. 837-851, 2005.

[3] L. Leem, H. Cho, J. Bau, Q.A. Jacobson, and S. Mitra. "ERSA: Error resilient system architecture for probabilistic applications," *Proceedings of Design, Automation and Test in Europe (DATE)*, 2010, pp. 1560 –1565.

[4] H. Esmaeilzadeh, A. Sampson, L. Ceze, D. Burger, "Architecture Support for Disciplined Approximate Programming," *Proceedings of International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2012, pp.301-312.

[5] A. Yazdanbakhsh, et al., "Axilog : Language Support for Approximate Hardware Design," *Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2015, pp.812-817.

[6] Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IMPACT: Imprecise adders for low-power Approximate Computing, "*Proceedings of IEEE/ACM international symposium on Low-power electronics and design (ISLPED)*, 2011, pp. 409-414.

[7] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, March 2014, pp. 1-4.

[8] L. Chen, J. Han, W. Liu, and F. Lombardi, "Design of Approximate Unsigned Integer Non-restoring Divider for Inexact Computing", *Proceedings of the 25th edition on Great Lakes Symposium on VLSI(GLSVLSI)*, 2015, pp. 51-56.

[9] N. Venkateswaran and Y.V. Ramana Rao, "A Numerically Approximate High-speed Divider for Image Processing Applications," *Information Technology Journal*, vol. 5, no. 3, pp. 445-447, 2006.

[10] J.P. Deschamps, G. J.A. Bioul, and G. D. Sutter, Synthesis of Arithmetic Circuits: FPGA, ASIC and Embedded Systems, Wiley, 2006.

[11] Nangate 45nm Open Cell Library. http://www.nangate.com/.

[12] U. Qidwai and C.H. Chen, Digital Image Processing: An Algorithmic Approach with MATLAB, CRC Press, 2009.