

Article

Approximate Square Root Circuits with Low Latency and Power Dissipation

Duhwan Kim  and Sunggu Lee 

Department of Electrical Engineering, Pohang University of Science and Technology, Pohang 37673, Korea;
kdh2016@postech.ac.kr

* Correspondence: slee@postech.ac.kr

Abstract: This paper proposes a series of approximate square root circuit designs with high accuracy, low latency, low area, and low power dissipation requirements. The proposed designs are constructed using an array of controlled add–subtract cell elements with both exact and approximate versions. The utility of the proposed designs are evaluated by utilizing them in an example image contrast enhancement application with demonstrably satisfactory results and large peak signal-to-noise ratios and structural similarity values. The accuracy and hardware characteristics of the proposed square root designs are also analyzed and compared with previously proposed state-of-the-art approximate square root designs. When applied to a 16-bit radicand (the number under the square root symbol), the proposed designs have the lowest error rates, normalized mean error distances, and mean relative error distances by at least 1.8x when compared to all previous methods using the same number of approximate cells. When the designs were synthesized using Synopsys Design Compiler with a 28 nm bulk CMOS process, the delay, area, power, and power-delay-product characteristics outperform all previous designs in all but a few cases. These results demonstrate that the proposed designs permit the use of a flexible range of approximate designs with varying accuracy and hardware overhead characteristics, and a suitable design can be selected based on the user design requirements.



Citation: Kim, D.; Lee, S.

Approximate Square Root Circuits with Low Latency and Power Dissipation. *Electronics* **2022**, *11*, 46.
<https://doi.org/10.3390/electronics11010046>

Academic Editors: Leonardo Pantoli, Egidio Ragonese, Paris Kitsos and Gaetano Palumbo

Received: 30 November 2021

Accepted: 21 December 2021

Published: 24 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: square root; approximate computing; error resilient; low power dissipation; low latency

1. Introduction

Artificial intelligence applications using the latest deep neural network designs typically involve massive amounts of time critical computations. However, such applications are also *error resilient* [1], which means that they can tolerate errors in computations without significant overall accuracy loss. Power, latency, and hardware area overhead are important considerations in circuit design. Thus, approximate circuits that can provide the required massive computations with low latency, low power usage, and low hardware area overhead are required [2]. Mobile systems, where battery issues are critical, can be particularly affected. Various circuits such as adders, subtractors, multipliers, and dividers have been approximated using various types of circuits, and it has been confirmed that such circuits can exhibit sufficient levels of performance.

Square root is a time-consuming but essential operation that is occasionally required in specific applications, including error-resilient applications such as those described above. However, because it typically requires a large amount of hardware resources, if it is used as an essential operation for a specific application, it can become a part of the critical path of a circuit and occupy a large proportion of the total operation time. Thus, this paper proposes a sequence of array-based square root designs that are suitable for a variety of error-resilient applications with varying accuracy and hardware resource requirements.

The remainder of this paper is organized as follows. Section 2 provides background material and an overview of related research. The proposed approximate square root circuit design is described in Section 3. In Section 4, the proposed designs are evaluated in terms of accuracy and circuit characteristics and compared to previous state-of-the-art research

work. An analysis of an application utilizing the proposed approximate square root designs is presented in Section 5, which is followed by concluding remarks in Section 6.

2. Background and Related Work

The square root of a number A , called the *radicand*, is the square root Q such that $Q * Q = A$. Since the square of a negative number and a positive number, with the same magnitude, are the same, a positive radicand will have two square roots. The unique nonnegative square root (either 0 or a positive number) of a nonnegative radicand is referred to as the *principle square root*.

2.1. Assumptions and Basic Circuit

Since this paper primarily targets arithmetic circuits for error-resilient applications that work with fixed point numbers such as image pixels, it is assumed that only nonnegative numbers are used for the radicand and square root. Thus, for simplicity, the term *square root* is used to refer to the principle square root of a radicand. Since only nonnegative integers are used, the square root of a radicand A is the square root Q with remainder R such that $Q * Q + R = A$, where $Q = \lfloor \sqrt{X} \rfloor$ and R is a nonnegative integer.

When depicted in the above manner, the square root operation can be viewed as similar to division. As in division, computation of the square root typically involves computation of the bits of the square root in an iterative *trial-and-error* manner. Thus, as in division, the square root can be computed using a *restoring* or *non-restoring* iterative approach as in long division (the primary school pencil-and-paper method) of integers written in decimal notation.

In particular, a non-restoring iterative square root circuit can be efficiently implemented in digital logic hardware using an array of *Controlled Add-Subtract* (CAS) cells, as shown in Figure 1. The detailed structure of a CAS cell is shown in Figure 2.

2.2. Related Work

Unlike other, more common arithmetic operations such as addition or multiplication, there are a relatively small number of research works that have specifically addressed circuits for the approximate square root operation. In the recent 2020 survey of approximate arithmetic circuits by Jiang et al. [3], there are only two references for square root circuits, and of those, only one [4] is for an *approximate* square root. However, another recent work by Arya et al. [5] proposes an alternative approximate square root design, and the approximate subtractor cells proposed by Chen et al. [6,7] can be appropriated for use in an approximate square root design.

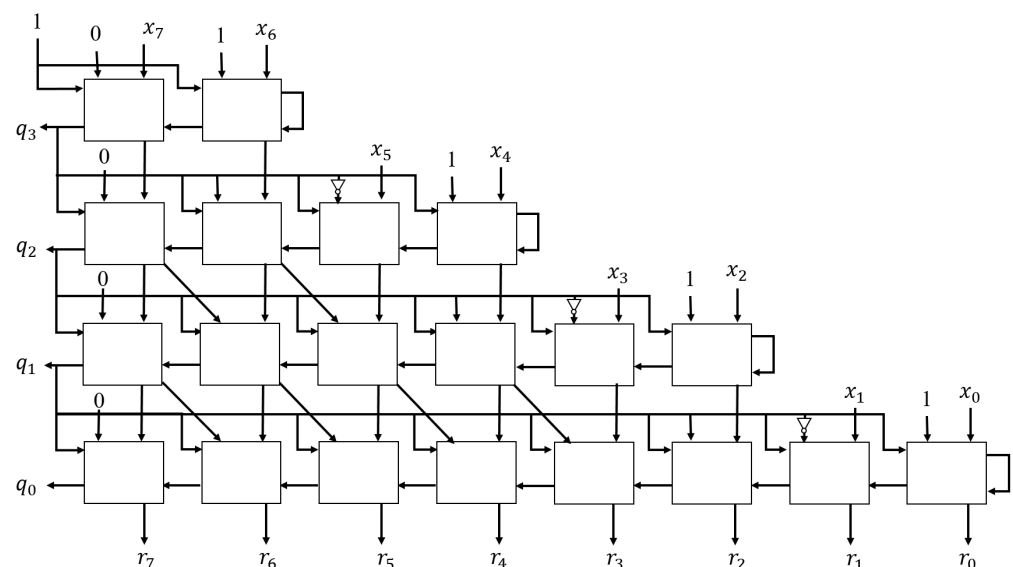


Figure 1. An example of an 8-bit non-restoring square root array circuit.

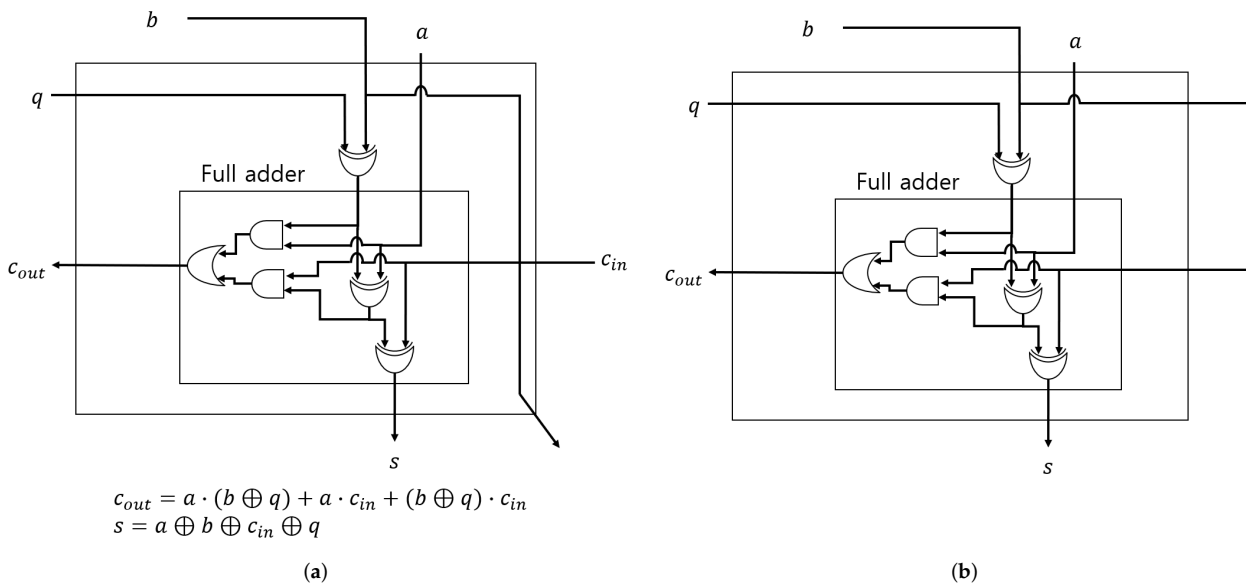


Figure 2. Controlled Add–Subtract (CAS) cells used in a square root array circuit: (a) a cell in the middle portion of the array, (b) a cell in the rightmost position of each row.

The approximate square root circuit proposed by Jiang et al. [4] is based on removing the most significant bits of the radicand A down to the first nonzero bit, truncating the least significant bits of A so that $2k$ bits remain, with k used as an approximation degree parameter, and then using an exact circuit for the remaining $2k$ bits. This is an interesting design that leads to considerable savings in hardware, but it can compromise accuracy greatly for large nonnegative radicands.

Recently, Arya et al. have proposed alternative approximate square root circuits [5] based on square root arrays with cells designed for area reduction and least significant bit truncation. These are simple designs in which the approximation cell used is simple wire fall-through connections for the horizontal and vertical input wires in the square root array of Figure 1. Thus, the resulting designs cannot be flexibly adjusted to achieve varying rates of accuracy or hardware overhead.

Chen et al. proposed AXDr, which is an approximate subtractor cell [6,7]. Although their cell design is used in a divider, the same cell design can be used in the square root array design of Figure 1. Since only a small fraction of the cells are approximated to maintain high accuracy, the advantage in terms of hardware overhead is small.

3. Proposed Approximate Square Root Designs

3.1. Approximate Controlled Add–Subtract (CAS) Cells

The proposed square root array design consists of $n(n + 1)$ CAS cells, and the cells can be classified into five types, as shown in Figure 3, according to the amount of digital logic in each cell. The most extreme cell design considered, named ASC0, uses simple fall-through wire connections. Next, ASC1 uses one inverter in the path from the right upper input to the vertical output that connects to the next row. Then, ASC2 uses one OR gate, ASC3 uses one inverter and one OR gate, and ASC4 uses a tree of three exclusive-OR gates. All of these designs are simpler than the exact CAS cell design shown in Figure 2, which has three exclusive-OR gates, two AND gates, and one OR gate.

A truth table can be constructed for the proposed ASC0 through ASC4 cell designs, as shown in Table 1. The exact results and correct outputs are shown using normal font, and erroneous results are shown using bold font. As can be seen, the designs ASC0 through ASC4 result in successively fewer incorrect outputs. In addition, even the simplest ASC0 design produces correct c_{out} and s outputs for half of the input combinations.

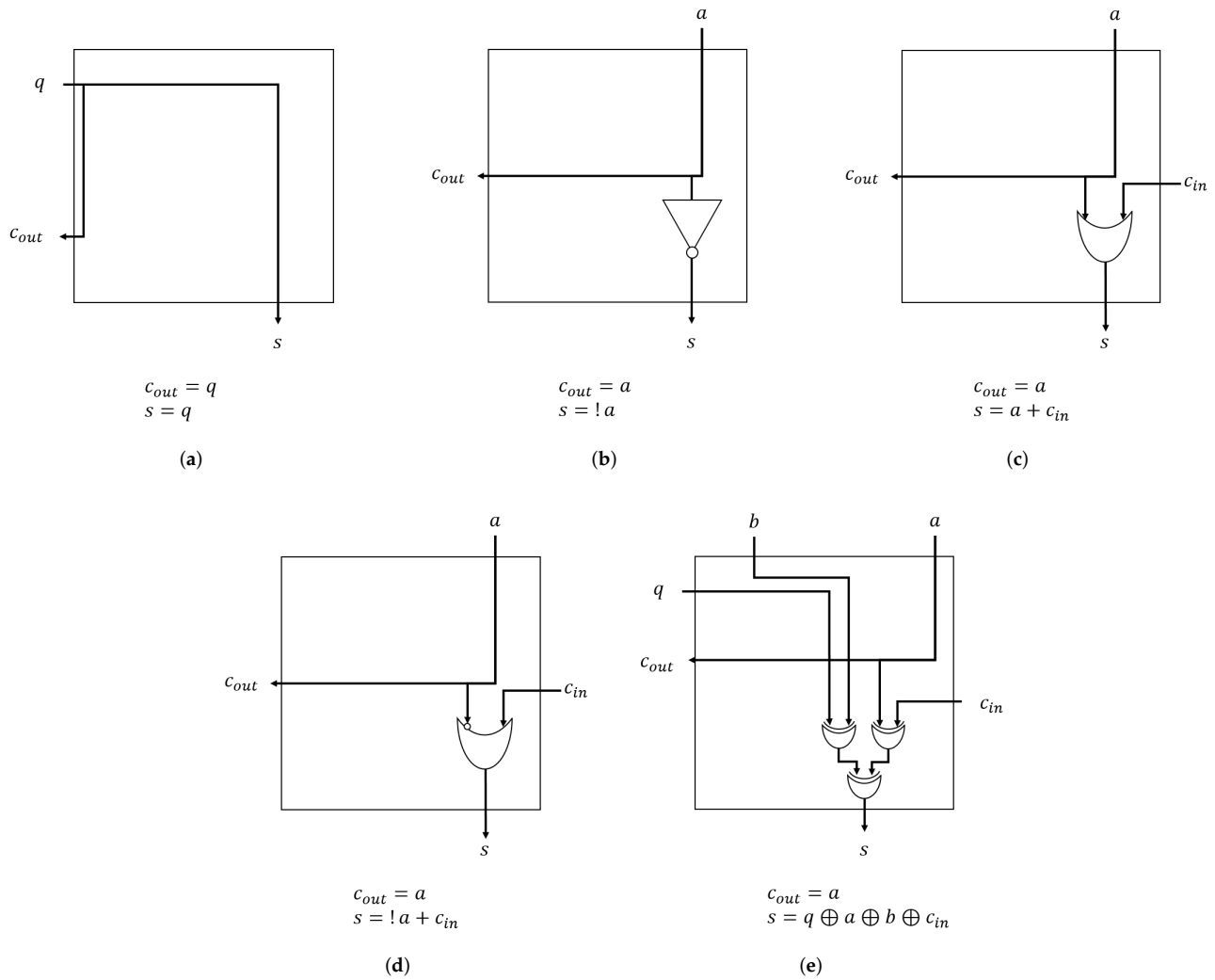


Figure 3. Approximate Controlled Add-Subtract (CAS) cells: (a) ASC0, (b) ASC1, (c) ASC2, (d) ASC3, and (e) ASC4.

Table 1. Truth table for CAS designs (**bold numbers** show incorrect outputs).

q	a	b	c_{in}	Exact		ASC0		ASC1		ASC2		ASC3		ASC4	
				C_{out}	s	C_{out}	s	C_{out}	s	C_{out}	s	C_{out}	s	C_{out}	s
0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	1	0	1	0	0	0	1	0	1	0	1	0	1
0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	1
0	0	1	1	1	0	0	0	0	1	0	1	0	1	0	0
0	1	0	0	0	1	0	0	1	0	1	1	1	0	1	1
0	1	0	1	1	0	0	0	1	0	1	1	1	1	1	0
0	1	1	0	1	0	0	0	1	0	1	1	1	0	1	0
0	1	1	1	1	1	0	0	1	0	1	1	1	1	1	1
1	0	0	0	0	1	1	1	0	1	0	0	0	1	0	1
1	0	0	1	1	0	1	1	0	1	0	1	0	1	0	0
1	0	1	0	0	0	1	1	0	1	0	0	0	1	0	0
1	0	1	1	0	1	1	1	0	1	0	1	0	1	0	1
1	1	0	0	1	0	1	1	1	0	1	1	1	0	1	0
1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1
1	1	1	0	0	1	1	1	1	0	1	1	1	0	1	1
1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	0

3.2. Replacement Methods

A square root array circuit consists of many cells and can be composed of approximate cells in various combinations in each row and column. When considering each of the eight columns in Figure 1, it is clear that the cells in the rightmost columns are less important than the cells in the leftmost columns, as the former and latter produce the least significant and most significant bits, respectively, of the final remainder R . Likewise, when considering the four rows in Figure 1, the cells in the lower rows are less important than the cells in the upper rows, as the former and latter produce the least significant and most significant bits, respectively, of the final quotient Q .

Using the above logic, two methods for replacing the exact CAS cells with approximate CAS cells are considered and shown in Figure 4. In the Stepwise Refinement (SR) method, exact CAS cells in the rightmost columns of the square root array are replaced with approximate CAS cells one column at a time. The variable p is used to denote the number of columns that are replaced with approximate CAS cells. Due to the right-triangle shape of the square root array in Figure 1, higher p values result in successively worse quotient Q and remainder R approximations.

In the Horizontal Refinement (HR) method, exact CAS cells in the lowermost rows of the square root array are replaced with approximate CAS cells one row at a time. Using the same variable p as in SR, there will be situations in which an entire row cannot be replaced with approximate CAS cells. In that case, the CAS cells are replaced in order starting from the rightmost column within that row. This type of row-based replacement method will again affect the accuracies of both the quotient Q and remainder R , but in a different manner from the SR method.

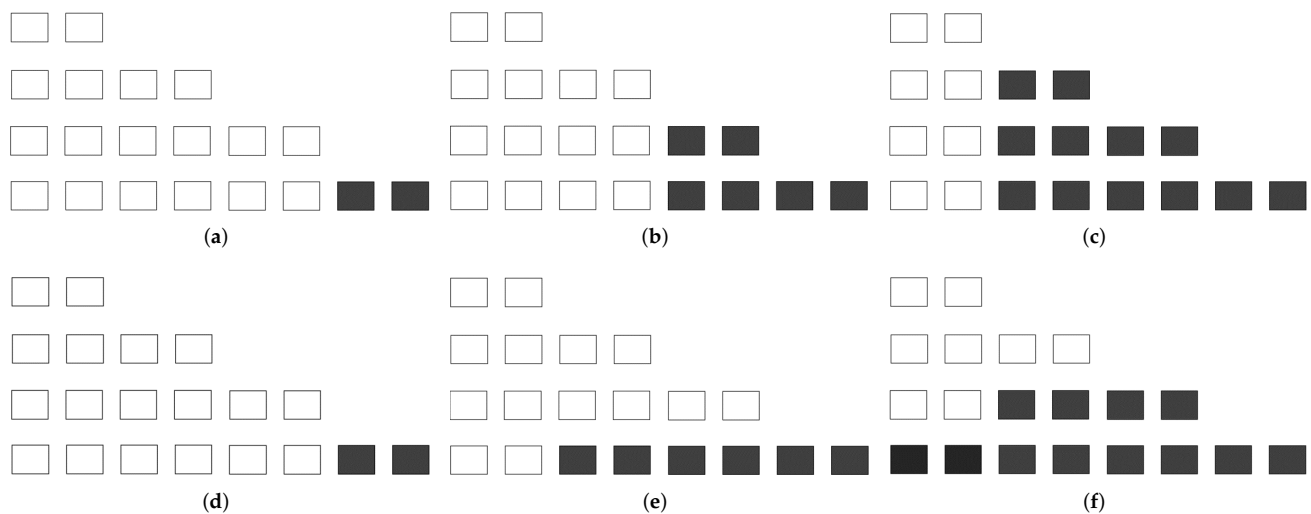


Figure 4. Proposed CAS cell replacement methods for an 8-bit square root circuit: (a) SR with $p = 2$, (b) SR with $p = 4$, (c) SR with $p = 6$, (d) HR with $p = 2$, (e) HR with $p = 4$, and (f) HR with $p = 6$.

4. Results

4.1. Accuracy Analysis

In order to analyze the accuracy, all operations for the circuits presented in this paper have been coded in C and simulated. The results are shown in Table 2. Only the quotient Q output is considered since this is the value that is most often used in image processing applications. For easy analysis, the proposed method and the best accuracy results for each value of p are marked using bold font in Table 2.

The metrics used for analysis are Error Rate, Normalized Mean Error Distance (NMED), and Mean Relative Error Distance (MRED). The Error Rate is the number of input combinations that result in incorrect outputs divided by the total number of input combinations. NMED is defined as the average of the error distances (differences between correct and

actual outputs) normalized by the maximum possible accurate output value [8]. MRED is the average of the relative error distances, and relative error distance is the absolute error distance divided by the correct result.

Table 2. Accuracy evaluation results (proposed designs and best accuracies shown in **bold font**).

Design	p	ER (%)	NMED (10^{-2})	MRED (10^{-2})
ASC-SR	4	1.16	0.004	0.024
	6	9.97	0.040	0.161
	8	43.33	0.197	0.611
ASC-HR	4	3.03	0.012	0.049
	6	50.00	0.195	0.396
	8	75.00	0.586	1.187
A-SQR	4	18.28	0.077	0.407
	6	84.32	1.369	5.074
	8	99.61	19.646	51.978
T-SQR	4	2.13	0.008	0.511
	6	14.04	0.058	0.290
	8	62.55	0.338	1.288
AASR	4	65.82	0.478	0.686
	6	82.30	1.154	1.715
	8	91.14	2.520	3.798

4.2. Hardware Overhead Evaluation

All circuits presented in this paper have also been evaluated for their circuit characteristics. The circuits to be compared were implemented in Verilog and Synopsys Design Compiler was used for circuit evaluation [9]. A Samsung 28 nm CMOS process, 1.1 V supply voltage, 200 MHz clock frequency, and a temperature of 25 °C were used for the synthesis and simulation settings.

Table 3 shows the hardware evaluation results. The metrics used in this evaluation are area, power dissipation, delay, and the Power Delay Product (PDP), which is a commonly used combination metric. For ease of analysis, the proposed methods and the best results for each value of p are shown using **bold font**.

Table 3. Hardware evaluation results (proposed designs and best results shown in **bold font**).

Design	p	Delay (ns)	Area (μm^2)	Power (μW)	PDP (fJ)
ASC-SR	4	1.56	105.30	12.0	18.7
	6	1.42	92.08	10.3	14.6
	8	1.18	72.77	7.65	9.03
ASC-HR	4	1.48	97.70	11.1	16.4
	6	1.19	80.85	8.25	9.82
	8	0.91	62.24	5.97	5.43
A-SQR	4	2.50	141.92	13.1	32.8
	6	1.83	109.86	8.79	16.1
	8	0.81	53.00	2.60	2.11
T-SQR	4	2.75	152.45	14.8	40.7
	6	2.39	132.80	12.3	29.4
	8	1.87	105.07	8.72	16.3
AASR	4	1.67	147.07	9.36	15.6
	6	1.32	117.59	6.45	8.51
	8	0.94	90.68	4.36	4.10

The proposed ASC-HR designs have the best delay and area characteristics for $p = 4$ and $p = 6$, while the ASC-HR delay and area values for $p = 8$ are only 12.3% and 17.4%

worse than the best values. Although the power usage and PDP values for the proposed ASC-SR and ASC-HR designs are somewhat worse than the best values, the differences are not extreme. Overall, the proposed ASC-SR design has the best accuracy, and both the ASC-SR and ASC-HR designs have hardware characteristics that are the best or close to the best for all values of p .

5. Application Analysis

Contrast Enhancement

The approximate square root presented in this paper is evaluated using an example error-resilient application. The targeted application is contrast enhancement, which is an image processing technique used to make the contrast of light and dark in black-and-white photos easier to recognize. It is widely used to make it easier to identify breast cancers caught on X-rays [10].

Figure 5 shows photos of the before and after images, and Table 4 shows PSNR and SSIM values for this contrast enhancement application for several representative versions of the proposed approximate square root designs. The proposed method and the best PSNR and SSIM for each value of p are marked using bold font in Table 4. The square root was calculated after each 8-bit pixel value in the image was multiplied by a factor of 128 for brightness. The application is written in C code, and the metrics used for evaluation are Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM). As can be seen from these results, the proposed designs all produce extremely accurate results with very high PSNR and SSIM values. Compared with other designs, it is shown that the PSNR and SSIM of ASC-SR are the highest in all values of p .

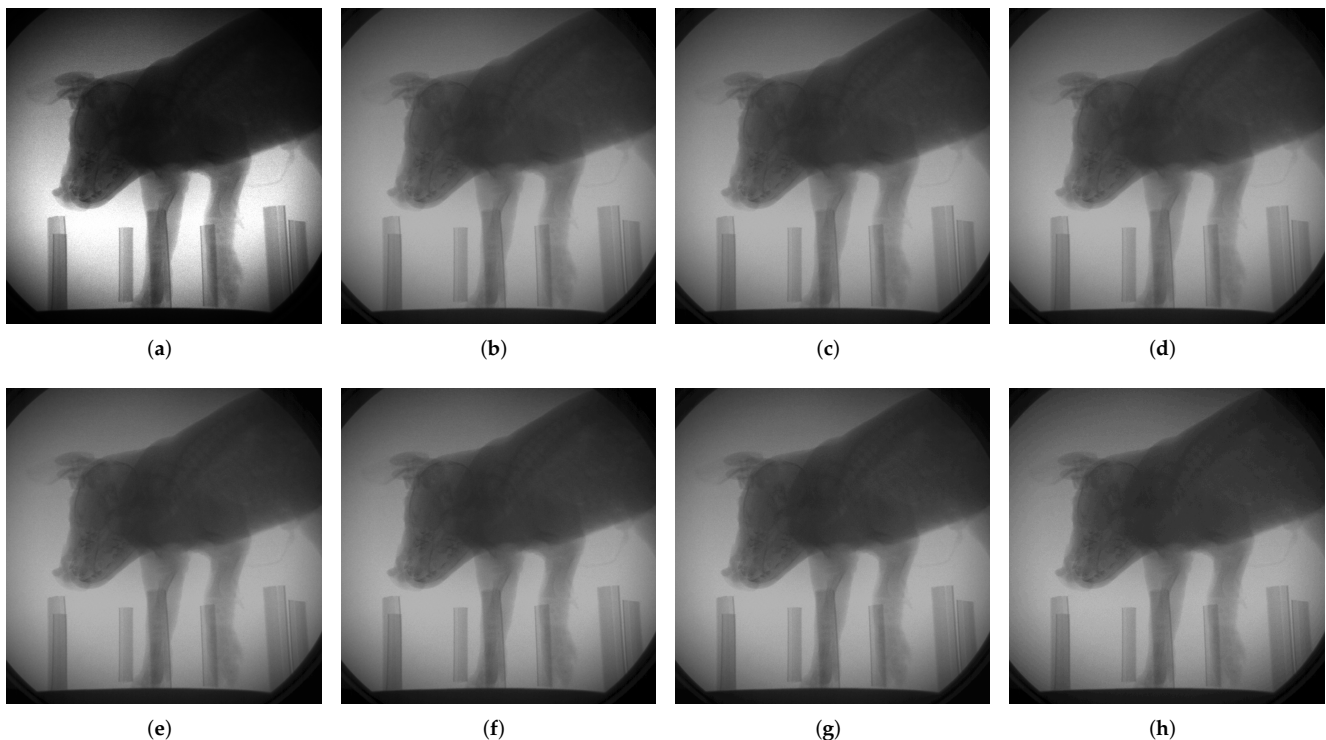


Figure 5. Contrast Enhancement. (a) Figure, (b) accurate, (c) ASC-SR with p 4, (d) ASC-HR with p 4, (e) ASC-SR with p 6, (f) ASC-HR with p 6, (g) ASC-SR with p 8, (h) ASC-HR with p 8.

Table 4. Truth table of cells.

Design	p	Contrast Enhancement	
		PSNR (dB)	SSIM
ASC-SR	4	62.33	0.99999
	6	55.37	0.99996
	8	46.43	0.99970
ASC-HR	4	58.11	0.99998
	6	51.33	0.99990
	8	42.65	0.99927
A-SQR	4	45.12	0.99964
	6	28.55	0.98751
	8	10.25	0.80462
T-SQR	4	53.68	0.99995
	6	45.65	0.99970
	8	37.76	0.99844
AASR	4	47.11	0.99980
	6	40.14	0.99912
	8	33.77	0.99626

6. Conclusions

This paper has proposed an approximate non-restoring square root array circuit that uses approximate Controlled Add–Subtract (CAS) cell designs that take into account the locations of the CAS cells in the array. The proposed designs are shown to produce extremely accurate square root computation results with very low latencies, area overhead, and power dissipation. When compared to previous state-of-the-art designs, the accuracy of the proposed ASC-SR designs are the best for each level of approximation used. In addition, both the proposed ASC-SR and ASC-HR designs have the best, or close to the best, hardware characteristics, in terms of latency, area, power dissipation, and power-delay product, when compared to previous state-of-the-art designs.

Author Contributions: Conceptualization, D.K. and S.L.; methodology, D.K.; software, D.K.; validation, D.K. and S.L.; formal analysis, D.K.; investigation, D.K.; resources, D.K.; data curation, D.K.; writing—original draft preparation, D.K.; writing—review and editing, D.K. and S.L.; visualization, D.K.; supervision, S.L.; project administration, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: The EDA tool was supported by the IC Design Education Center(IDECE), Korea.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Han, J.; Orshansky, M. Approximate computing: An emerging paradigm for energy-efficient design. In Proceedings of the 2013 18th IEEE European Test Symposium (ETS), Avignon, France, 27–30 May 2013; pp. 1–6.
2. Chippa, V.K.; Venkataramani, S.; Chakradhar, S.T.; Roy, K.; Raghunathan, A. Approximate computing: An integrated hardware approach. In Proceedings of the 2013 Asilomar conference on signals, systems and computers, Pacific Grove, CA, USA, 3–6 November 2013; pp. 111–117.
3. Jiang, H.; Santiago, F.J.H.; Mo, H.; Liu, L.; Han, J. Approximate arithmetic circuits: A survey, characterization, and recent applications. *Proc. IEEE* **2020**, *108*, 2108–2135. [\[CrossRef\]](#)
4. Jiang, H.; Liu, L.; Lombardi, F.; Han, J. Low-Power Unsigned Divider and Square Root Circuit Designs Using Adaptive Approximation. *IEEE Trans. Comput.* **2019**, *68*, 1635–1646. [\[CrossRef\]](#)
5. Arya, N.; Soni, T.; Pattanaik, M.; Sharma, G.K. Area and Energy Efficient Approximate Square Rooters for Error Resilient Applications. In Proceedings of the 2020 33rd International Conference on VLSI Design and 2020 19th International Conference on Embedded Systems (VLSID), Bengaluru, India, 4–8 January 2020; pp. 90–95.
6. Chen, L.; Han, J.; Liu, W.; Lombardi, F. Design of approximate unsigned integer non-restoring divider for inexact computing. In Proceedings of the 25th edition on Great Lakes Symposium on VLSI, Pittsburgh, PA, USA, 20–22 May 2015; pp. 51–56.

7. Chen, L.; Han, J.; Liu, W.; Lombardi, F. On the design of approximate restoring dividers for error-tolerant applications. *IEEE Trans. Comput.* **2016**, *65*, 2522–2533. [[CrossRef](#)]
8. Liang, J.; Han, J.; Lombardi, F.; Han, J. New metrics for the reliability of approximate and probabilistic adders. *IEEE Trans. Comput.* **2013**, *62*, 1760–1771. [[CrossRef](#)]
9. Synopsys Co. RTL Synthesis. Available online: <https://www.synopsys.com/support/training/rtl-synthesis.html> (accessed on 16 December 2021).
10. Dhawan, A.P.; Buelloni, G.; Gordon, R. Enhancement of mammographic features by optimal adaptive neighborhood image processing. *IEEE Trans. Med. Imaging* **1986**, *5*, 8–15. [[CrossRef](#)] [[PubMed](#)]