International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Stochastic Computing for Compute-Intensive Sections of JPEG Compression

Minal S. Deshmukh[1,2], Prasad D. Khandekar[1], Ketan J. Raut[2]

Submitted: 30/05/2023 **Revised**: 15/07/2023 **Accepted**: 29/07/2023

Abstract: Embedded real-time image processing applications are subject to strict design constraints concerning size and power. As devices scale down to nanoscale dimensions, circuit reliability becomes an increasingly pressing concern. Stochastic computing (SC) emerges as a cost-effective alternative to conventional deterministic binary computing, achieving this by encoding and processing information through digitized probabilities. This paper delves into the exploration of reconfigurable architectures to develop precise image-processing circuits for the compute-intensive blocks within baseline JPEG compression algorithms. These encompass RGB to YCbCr conversion, Discrete Cosine Transform (DCT), and Quantization, employing both deterministic and stochastic logic. Synthesis trial results underscore that stochastic implementation requires fewer hardware resources, occupies less physical area, and consumes less power when contrasted with deterministic logic. Employing an FPGA (PYNQ-Z2), the proposed designs have been executed, leading to power reductions of 112% and 120%, as well as area utilization improvements of 14% and 67.67%, for RGB to YCbCr and DCT-Quantization stochastic circuits, respectively, in comparison with deterministic circuits

Index Terms *- Stochastic Computing, Deterministic Computing, Image processing applications, FPGA*

1. Introduction

Real-time image processing typically needs a lot of hardware and/or software resources since pixel streams include enormous volumes of data. Due to the computational demands of real-time image processing, there are significant layout limitations in relation to space, power as well as cost. In the context of image representation, an image is stored as a two-dimensional array, akin to a matrix and contemporary image processing programs often require functional modifications to the pixels of an input image [1][2][3][4]. If the hardware is compact, then may be combined with the imaging-sensing circuits to create a so-called" vision chip". These chips act as the preprocessing front end of an image processing system [1]. The implementation of Signal, Image Processing, Machine learning, neural networks, and analytics applications on programmable devices are intriguing considering the current developments in the Internet of Things (IoT) and wearable technology. However, multiple components and extremely intricate connections are frequently needed, which increases the amount of hardware and energy used. Stochastic Computing (SC) is a potential method for handling digital information. SC satisfies rigid IOT device requirements in terms of hardware resources for embedded and mobile platforms [5]. A team led by B. R. Gaines first presented stochastic computing SC in the 1960s where binary data is used as probabilities, SC is an alternative to the conventional approach. SC makes use of very simple arithmetic units, Gaines introduced stochastic modules to carryout fundamental mathematical operations such as addition, multiplication, and division on stochastic numbers [6]. Minimal hardware footprint, low power, and energy-efficient circuits are achieved with SC-based hardware execution while high precision is maintained [5][7]. This study investigates SC as a binary computing option that is low-cost, low-power, and low-complexity. This work addresses the design of image processing circuits for compute-intensive blocks of the JPEG compression algorithm, encompassing both deterministic and stochastic approaches. First, we discuss the mathematical steps involved in the hardware implementation of RGB to YCbCr, DCT, and Quantization for both stochastic and deterministic logic circuits. Subsequently, we provide a quantitative analysis, focusing on area utilization and power. The structure of the paper is as follows. The process for synthesizing digital circuits using stochastic logic is covered in the section, II which also gives an introduction to stochastic logic. Section III outlines the prior endeavors conducted within the domain of Stochastic Computing, along with pertinent investigations into Reconfigurable Stochastic hardware applied to Image Processing applications. The section V and VI provide detailed explanations of the hardware design for the selected applications – RGB to YCbCr,DCT, and Quantization, respectively. These sections also encompass comprehensive descriptions of the implementation processes. In Section VII, you will find the outcomes of the conducted experiments, while

[1] *Dr Vishwanath Karad MIT World Peace University, School of Electronics and Communication Engineering Kothrud, Pune, India*
[2] *Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Information and Technology, Pune, India*
Email:minal.deshmukh@viit.ac.in; khandekar.prasad@gmail.com

Section VIII delves into conclusions and future prospects. The paper concludes with a References section.

2. Stochastic Computing

Future integrated circuits will likely be constructed from developing Nano devices. Reduced integrated circuit footprint makes it more noise sensitive, which could result in unstable hardware implementation. Ionizing radiation-induced soft errors are also a significant problem. Currently used fault-tolerant solutions like error-correcting codes and modular redundancy are used to mitigate these issues. Furthermore, the stochastic implementation in deep submicron devices can be extremely resilient to tiny data variations that may be brought on by the production process or the operational environment. Stochastic circuits and stochastic computing applications create low-power designs with a small physical footprint for the hardware execution [5][6][7].

The fundamental benefit of stochastic computing is the high degree of parallelism made possible by the low complexity of hardware arithmetic units. As seen in Fig. 1(a) logical AND gate can be used to multiply and Fig 1(b) scaled multiplexer is used for addition [7][8]. In some arithmetic-intensive systems, stochastic circuits can replace conventional deterministic hardware components. A distinctive data representation used in stochastic computing relies on the statistics of the information rather than the bit positions of the bits in the stream. Probability is used to express a stochastic number. Consider the binary values "01001011" and "0110110101101110" which represent 0.5 and 0.625, respectively. To illustrate, let's consider an example with a bit-stream X, consisting of 35% ones and 65% zeros. This signifies the value of a number, n = 0.35, representing the probability of encountering a one at a random bit location. It should be emphasized that X's structure and length are not required to be fixed. During n clock cycles, the n-bit inputs, X1 and X2, undergo multiplication through a two-input AND operation. The resultant, Z, will be 1 only if both the inputs are 1 in the similar cycle. To achieve addition stochastic number with the sum constrained within the range [0, 1], a scaled version of a two-input multiplexer is employed [6][7].

Achieving desired precision and accuracy would require careful design of the stochastic number generators. The two popular Probability conversion circuits (PCC) used for binary-to-stochastic conversion are digital comparator CMP and weighted binary generator (WBG) [8]. Employing either a genuinely random or, more probable, a pseudorandom number generator, the conversion procedure comprises creating an m-bit random binary number in each clock cycle and then comparing it to the m-bit input binary number. Within the context of CMP, when the generated random number is smaller than the binary number, the comparator outputs a 1, and when it is not, it outputs a 0. The probability of a 1 occurring at the comparator's output during each clock cycle matches the binary input of the converter when it is translated into a fractional number, presuming that the random numbers are spread evenly over the range [0,1]). WBG circuits function differently. Initially, it transforms the output sequence from an LFSR into a series of binary numbers with assigned weights. Subsequently, it produces the output bit stream by utilizing both the weighted sequence and the input binary number x [7][8][9]. For this investigation, we used the WBG, functions as a PCC proposed by Gupta and Kumaresan [10] shown in figure 2(a) and the an LFSR (Linear Feedback Shift Register) shown in figure 2(b) as the source of random numbers in SC circuits because of its benefits in terms of minimal hardware complexity and fast operation [9]. It is much easier to convert a stochastic number to binary.

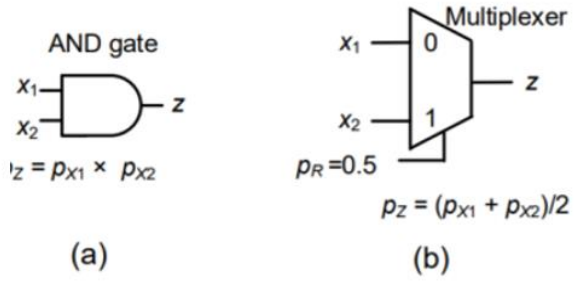

Fig.1. Selection of components for stochastic computing [7]

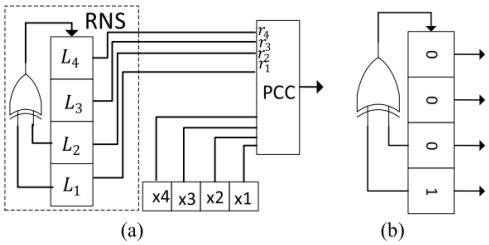

Fig. 2. (a) General structure of an SNG. (b) LFSR is used as RNS [8]

The value of the stochastic number, p, which is carried by the number of 1s in its bit-stream form, must be extracted by counting these ones [7].

3. Relevant Work

Von Neumann and Shannon are credited with developing reliable computing [11] [12]. For fault tolerance, methods like majority voting and modular redundancy are frequently employed. Circuit and system design frequently employ probabilistic approaches. They are typically used to try and characterize uncertainty. For example, statistical timing analysis is utilized in transistor sizing to enhance yield as well as to achieve more stringent performance limits. Numerous variants of probabilistic design have been suggested for integrated circuits. As an example, in [13], a design methodology centered around nanotechnology using Markov random fields is presented. Furthermore, there has been encouraging progress in the design of stochastic processors [15]. The fundamental principles of stochastic computing were initially introduced by Gaines [6], who proposed one of the earliest stochastic computing systems called stochastic computational logic. This system relies on combinational logic operations like multiplication and scaled addition (or subtraction). In 2010, Qian et al. [16] introduced a Reconfigurable Stochastic Computing (ReSc) architecture that relied on combinational logic. The architecture utilized Bernstein basis functions to represent both polynomial and non-polynomial functions. Semiconductor systems designed for moderate power utilization, such as transistors with varying pinch-off voltage and oxide widths, require dedicated manufacturing method. These methods can result in larger integrated circuit sizes and longer processing times. However, they achieve reduced power usage compared to conventional systems [12]. We leverage stochastic circuits and stochastic computing applications to create low-power designs with a small physical footprint for the hardware execution of this research. This paper builds upon the work of Alaghi et al. [1] in real-time Image Processing and Li et al. [17] in Stochastic computing for Image Processing algorithms. The state of art technique paper validates their findings and derives new insights and extrapolations based on their results.

This paper's primary contributions are:

1. This research focuses on the FPGA implementation of traditional binary (deterministic) and stochastic circuits for prevalent image processing tasks, including Color conversion, DCT, and Quantization.
2. Hardware resources have been compared; stochastic circuits are more compact and less power-hungry than their traditional counterparts in terms of area and power.

4. Real-Time hardware Implementation (Zynq)

In the process of converting raw images into a processed format suitable for further analysis, a systematic series of steps is undertaken as shown in Figure 3 [18] [19]. The initial step, referred to as Step 1, involves converting raw images stored in JPEG and BMP formats into HEX files, specifically using the. coe format. This HEX format enables the representation of image data compatible with subsequent processing. Converting image pixels to a. coe file entails encoding pixel values into hexadecimal format, creating a memory initialization file used in FPGA designs for initializing memory blocks

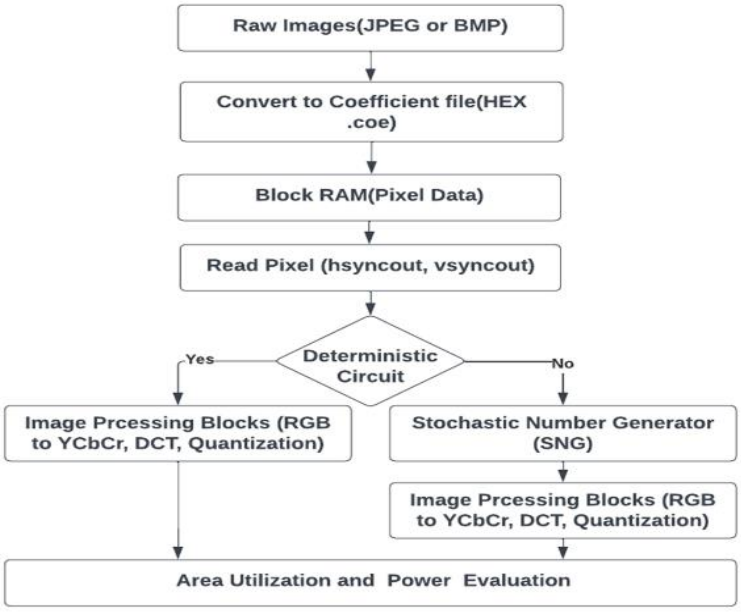

Fig. 3 Flow Depicting Real-Time Hardware Implementation

like Block RAM (BRAM) with image data. Moving to Step 2, the process includes reading a coefficient file containing RGB pixel values. These values are significant as each pixel's color is delineated by a 4-bit resolution. This lower bit resolution balances computational efficiency with visual fidelity. Step 3 involves storing these pixel values within a block RAM, an essential organization and retention step for efficient subsequent access. Step 4 covers synchronization, where pixel values are read with horizontal and vertical synchronous counts (hsyncout and vsyncout). This synchronization ensures accurate pixel data extraction and processing aligned with image spatial orientation. Step 5 introduces image processing blocks including RGB-to-YCrCb conversion, Discrete Cosine Transform (DCT), and quantization for deterministic circuit. For Stochastic circuits convert the processed image pixels into stochastic bit-stream using a Stochastic Number Generator for image processing blocks RGB-to-YCrCb conversion, DCT, and quantization. Finally, Step 6 determines efficiency in terms of area utilization and power consumption, offering insights into computational efficiency and performance trade-offs across the entire processing pipeline.

5. Color Conversion Rgb To Ycbcr

The RGB image undergoes conversion into various formats including YCbCr, CMY, and YUV, especially when color information plays a pivotal role in applications. Notably, the human eye finds it simpler to perceive luminance compared to chrominance components. One of the prominent advantages of transitioning from RGB to YCbCr is the capability to eliminate brightness influence during image processing. Consequently, the conversion to YCbCr takes place in the context of the JPEG compression process. In the case of monochrome imagery, the luma channel, often denoted as Y (or more precisely Y' to signify gamma encoding), closely approximates the image's content. Meanwhile, the two Chroma channels, Cb and Cr, serve as channels for color difference [20] [21]. The RGB to YCbCr conversion formula given in equation 1 is a mathematical transformation used to convert Red, Green, and Blue color values of a pixel into its corresponding luminance (Y) and chrominance (Cb and Cr) components. Mainly for enabling efficient image compression and processing

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ Cb &= -0.169R - 0.331G + 0.500B + 128 \\ Cr &= 0.500R - 0.419G - 0.081*B + 128 \end{aligned} \quad (1)$$

These equations can be directly mapped to the architecture shown in Figure 4 The blue boxes in Figure represent logic blocks, which are always implemented using XtremeDSP slices.

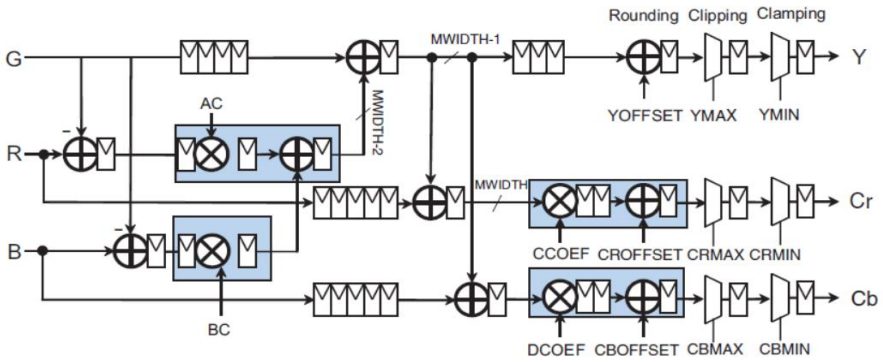

Fig. 4. Application Schematic (Vivado Design Suite) [19]

In this study we have implemented RGB to YCbCr equations on the PYNQ-Z2 board which is built around the Xilinx Zynq-7000 all Programmable System-On-Chip (SoC), which combines both an FPGA (Field-Programmable Gate Array) and ARM Cortex-A9 processors on a single chip. The table below shows utilization summary comprising of area utilization and power on Chip for RGB to YCbCr equation using Deterministic circuit.

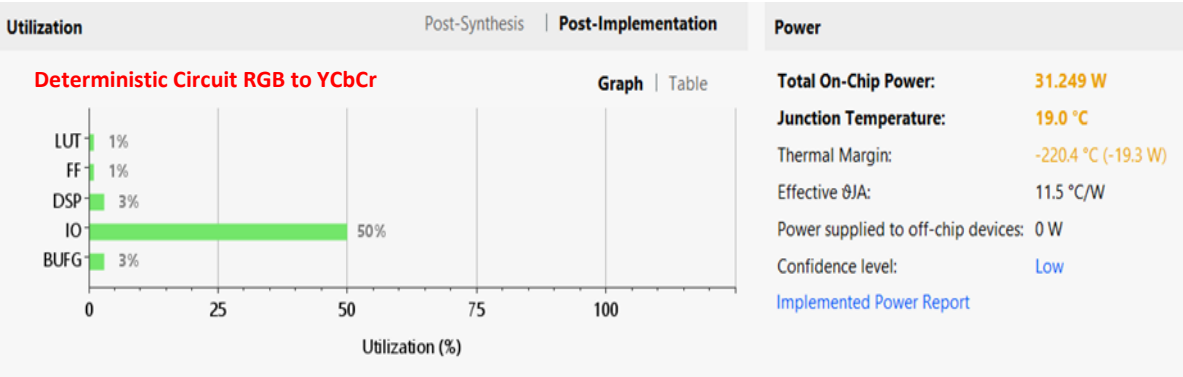

Fig. 5. Area Utilization and Power Analysis for Deterministic Circuit for RGB to YCbCr

The below section shows Stochastic Computing Approach for Optimized Stochastic RGB to YCbCr hardware implementation as shown in Figure 6.

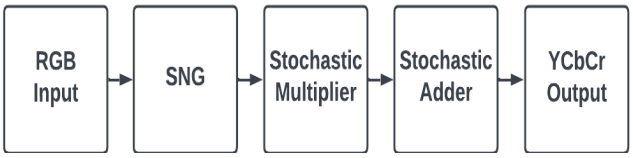

Fig.6. Block Diagram of Stochastic approach for RGB to YCbCr Conversion

In the optimized Stochastic RGB to YCbCr, the process initiates by intricately encoding RGB color values (R, G, B) into HEX files (. coe). These encoded colors smoothly transition to stochastic number generators, precision-engineered tools that produce random bit streams serving as representative probabilities. The following phase sees generated stochastic bit streams converge upon stochastic multipliers (AND logic), pivotal units executing multiplication operations. Leveraging stochastic bit streams as operands introduces calculated randomness into core arithmetic processes. This interplay of structured randomness helps in reducing area and power utilization. Stochastic accumulators orchestrate the culmination of multiplications, seamlessly integrating outcomes from stochastic product calculations. This iterative process facilitates gradual unfolding of cumulative values. Stochastic adders (scaled 2 input MUX) are used to perform the necessary additions for Y, Cb, and Cr components. Figure 7, below shows utilization summary of optimized RBG to YCbCr algorithm using stochastic computing.

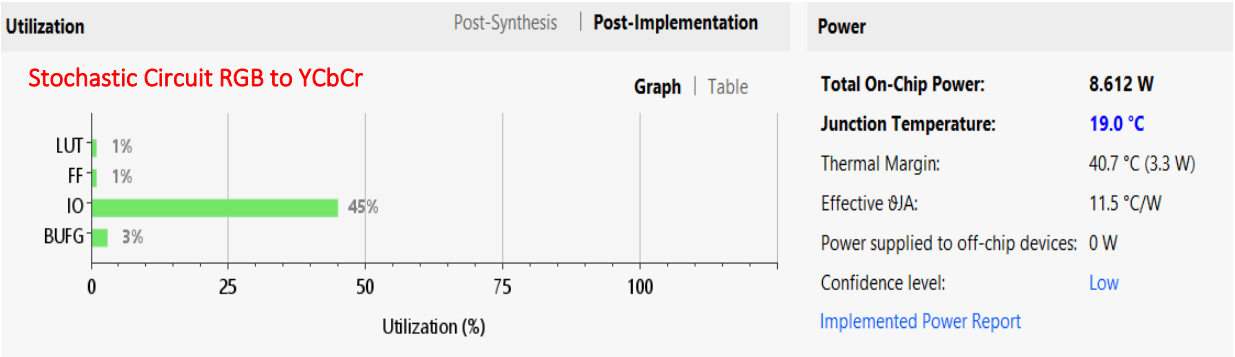

Fig.7. Area Utilization and Power Analysis for Stochastic Circuit RGB to YCbCr

Although multiplication is a fundamental operation in the mathematical transformation from RGB to YCbCr, DSP blocks are not employed in the circuit's implementation which reduces area utilization and total power on chip. Can be seen evidently from above utilization report in Figure 7.

6. Discrete Cosine Transform (Dct) and Quantization

The DCT is a mathematical technique that transforms a 2D array of pixel values into a frequency-domain representation, and quantization is the process of reducing the precision of these transformed coefficients to achieve compression. The DCT formula describes how the input signal is transformed into a set of DCT coefficients by computing the weighted sum of the input samples multiplied by cosine functions of different frequencies [20] [21]. The resulting DCT coefficients provide information about the frequency components present in the original signal, making it useful for various signal processing and compression applications. direct implementation DCT as shown in equation 2 requires 64 multiplications and 56 additions. FPGA platform consists of five main elements: DSP blocks, look-up tables (LUTs), flip-flops, random access memory (RAM) blocks and routing matrix [18] [19]. After applying the DCT, the resulting frequency coefficients typically contain both high-frequency and low-frequency components. Quantization involves dividing these coefficients by certain values (quantization step sizes) and rounding them to a limited number of levels, thereby reducing the precision of the coefficients as shown in equation 3.

DCT Formula (2D):

$$X(u, v) = \alpha(u) * \alpha(v) * \sum [\, x(m, n) * \cos((2m+1)u\pi / 2N) * \cos((2n+1)v\pi / 2N) \,] \quad \text{for } 0 \le u, v < N \qquad (2)$$

Where: X(u, v) is the DCT coefficient at frequency (u, v).

$\alpha(u) = \alpha(v) = 1/\sqrt{2}$ for u = 0 or v = 0, otherwise $\alpha(u) = \alpha(v) = 1$.

(m, n) are pixel coordinates within the block. N is the block size.

Quantization Formula:

$$Q(i, j) = \text{round}(DCT(i, j) / QF) \qquad (3)$$

Where: Q(i, j) is the quantized DCT coefficient.

DCT(i, j) is the DCT coefficient.

QF is the quantization factor, determining the level of compression

The above equations can be directly mapped onto an FPGA for deterministic computing. The DSP blocks, including pre-adders, multipliers, and accumulators, are mainly used for hardware implementations, which require a larger hardware footprint and more power. Figure 8, below shows utilization summary of Deterministic circuit for DCT and Quantization

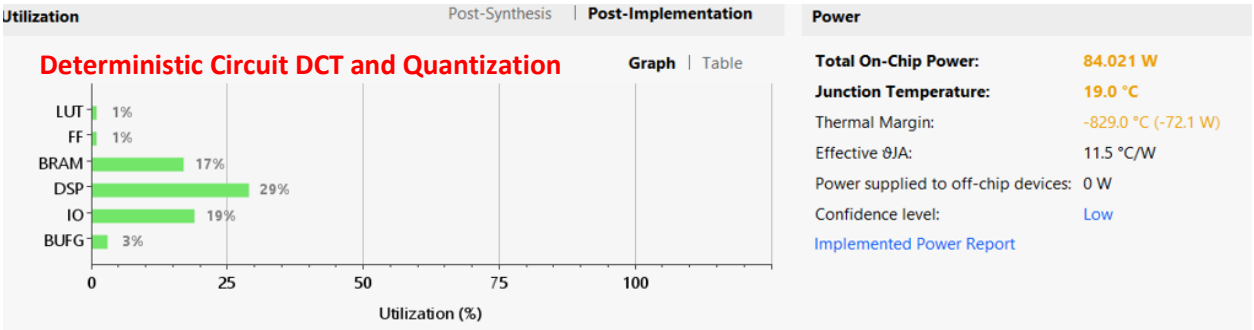

Fig. 8 Area Utilization and Power Analysis for Deterministic DCT and Quantization Circuit

The below section shows Stochastic Computing Approach for DCT and Quantization as shown in Figure 9.

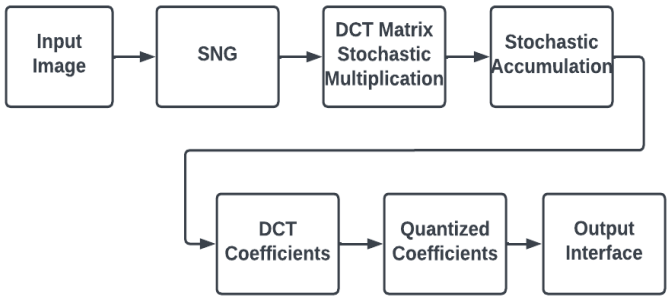

Fig. 9 Block Diagram of a Stochastic Computing Approach for DCT and Quantization

During this approach, the Discrete Cosine Transform (DCT) is applied to the image data using specialized tools called stochastic multipliers (AND gate) and accumulators(MUX). These tools are influenced by random bit sequences from a number generator. The resulting DCT coefficients are then made simpler through a process called quantization, using more stochastic multiplication and accumulation. Next, for additional scaling and simplification, the quantized coefficients undergo stochastic division (XOR) and shifting. Ultimately, we get a set of quantized DCT coefficients. These coefficients are then passed to an output interface, which can lead to further steps like processing, storage, or display of the image.

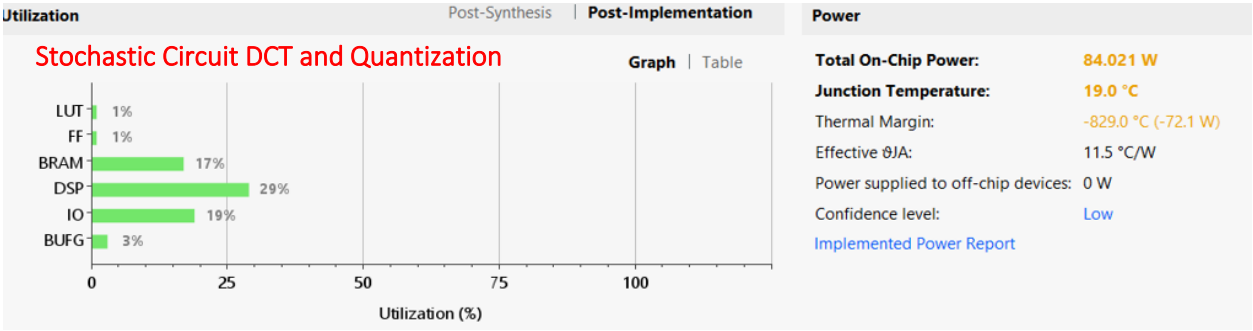

Fig.10. Area Utilization and Power Analysis for Stochastic DCT and Quantization Circuit

7. Result and Discussion

Both image processing applications involve primarily mathematical transformations of pixel values, and these equations can be directly mapped onto the FPGA architecture. The FPGA platform comprises five main elements: DSP blocks, look-up tables (LUTs), flip-flops, random access memory (RAM) blocks, and a routing matrix. DSP blocks are essential for the mathematical transformation of pixel values. From Figure 11 it is evident DSP block are used in hardware implementation of DCT and Quantization block of deterministic circuit. Color conversion block also need DSP blocks for mathematical transformation of formula shown in equation 1.

Utilization report for Deterministic Circuit

Name	Slice LUTs (53200)	Slice Registers (106400)	Slice (13300)	LUT as Logic (53200)	Block RAM Tile (140)	DSPs (220)	Bonded IOB (125)	BUFGCTRL (32)
dct_syn	520	887	246	520	24	64	24	1
dut (fdct)	520	887	246	520	24	64	0	0

Utilization report for Stochastic Circuit

Name	Slice LUTs (53200)	Slice Registers (106400)	Slice (13300)	LUT as Logic (53200)	Block RAM Tile (140)	Bonded IOB (125)	BUFGCTRL (32)
dct_syn	2495	3021	779	2495	24	20	1
dut (fdct)	2495	3021	779	2495	24	0	0

Fig. 11. Area utilization report of DCT and Quantization block.

DSP (Digital Signal Processing) blocks require more power and area due to their inherent complexity and specialized design. These blocks are tailored to perform intricate mathematical operations on digital signals, often involving multiplications, additions, and other computations. To execute these operations efficiently, DSP blocks consist of multiple components, such as multipliers and accumulators, which demand substantial hardware resources. Additionally, DSP blocks are designed to operate at high clock frequencies to handle real-time signal processing tasks. This increased clock speed, coupled with the complexity of operations, leads to higher power consumption. Consequently, the need for these specialized functionalities in DSP applications comes at the cost of increased power consumption and larger physical space on the integrated circuit.

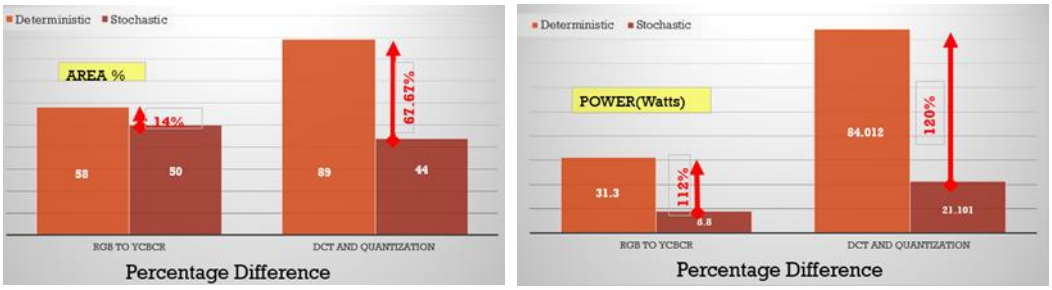

Fig. 12 comparison of area and Power utilization

Upon comparing the Area Utilization and Power Analysis of Stochastic and Deterministic circuits for both image processing applications given in Figure 12, it becomes evident that Deterministic circuits require more power and exhibit higher area utilization due to the utilization of DSP blocks. Moreover, accommodating a deterministic circuit in a single FPGA is also challenging task. On the other hand, stochastic circuits employ basic logic gates to perform arithmetic operations, leading to a significant reduction in both required power and area utilization.

8. Conclusion

The synthesis trials yielded compelling results, underscoring the advantages of stochastic implementation. Stochastic circuits demonstrated superior resource efficiency, occupying less area and consuming less power when compared to their deterministic counterparts. The proposed designs were successfully implemented on an FPGA (PYNQ-Z2), showcasing substantial reductions in power consumption of 112% and 120%, and improvements in area utilization of 14% and 67.67% for RGB to YCbCr and DCT-Quantization stochastic circuits, respectively, in contrast to deterministic circuits. This research not only sheds light on the promising potential of stochastic computing for efficient image processing but also substantiates its practical benefits through concrete implementation and evaluation. These findings have important implications for the development of energy-efficient and compact image processing circuits in the context of contemporary technological constraints.

References

[1] Armin Alaghi, Cheng Li and John P. Hayes, "*Stochastic Circuits for Real-Time Image-Processing Applications*", DAC'13, May 29–June 07, 2013, Austin, TX, USA

[2] Bouridane, D. Crookes, P. Donachy, K. Alotaibi, K. Benkrid, "*A high level FPGA-based abstract machine for image processing*", Journal of Systems Architecture Volume 45, Issue 10, April 1999, Pages 809-824

[3] Sparsh Mittal, Saket Gupta and S. Dasgupta, "*FPGA: An Efficient and Promising Platform for Real-Time Image Processing Applications*," Proceedings of the National Conference on Research and Development in Hardware & Systems (CSI-RDHS), 2008

[4] J Sarkhawas, P D Khandekar and A A Kulkarni, "*Variable Quality Factor JPEG Image Compression Using Dynamic Partial Reconfiguration and Microblaze*" in International Conference ICCUBEA 2015, at PCCOE Pune, pp 620-624, IEEE DOI 10.1109/ICCUBEA.2015.127.

[5] Armin Alaghi , Weikang Qian, John P. Hayes "*The Promise and Challenge of Stochastic Computing*", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 2017.

[6] B. R. Gaines, Stochastic Computing Systems, Department of Electrical Engineering Science University of Essex, Colchester, Essex, U.K

[7] Alaghi, John P. Hayes," *Survey of Stochastic Computing*", ACM Transactions on Embedded Computing Systems, Vol. 12, No. 2s, Article 92, 2013. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73

[8] Sayed Ahmad Salehi "*Low-Cost Stochastic Number Generators for Stochastic Computing*", IEEE Transactions On Very Large Scale Integration (VLSI) Systems, 2020.

[9] Junsangsri, Pilin; Lombardi, Fabrizi, "*Design of a Random Number Generator for Stochastic Computing(SC)*" (2022) TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.21341715.v1

[10] P.K. Gupta; R. Kumaresan, "*Binary multiplication with PN sequences*" IEEE Transactions on Acoustics, Speech, and Signal Processing (Volume: 36, Issue: 4, April 1988) https://doi.org/10.1109/29.1564

[11] Moore, Edward F., and Claude E. Shannon. "*Reliable circuits using less reliable relays*." Journal of the Franklin Institute 262.3 (1956): 191-208.

[12] Von Neumann, John. "*Probabilistic logics and the synthesis of reliable organisms from unreliable components.*" Automata studies 34 (1956): 43-98.

[13] Nepal, Kundan, et al. "*Designing logic circuits for probabilistic computation in the presence of noise.*" Proceedings of the 42nd annual Design Automation Conference. ACM, 2005

[14] Palem, Krishna V. "*Energy aware computing through probabilistic switching: A study of limits*." Computers, IEEE Transactions on 54.9 (2005): 1123 -1137.

[15] Narayanan, Sriram, et al. "*Scalable stochastic processors.*" Proceedings of the Conference on Design, Automation and Test in Europe. European Design and Automation Association, 2010.

[16] Weikang Qian; Riedel, M.D., "*The synthesis of robust polynomial arithmetic with stochastic logic,*" Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE, vol., no., pp.648,653, 8 -13 June 2008

[17] Li, Peng, and David J. Lilja. "*Using stochastic computing to implement digital image processing algorithms."* Computer Design (ICCD), 2011 IEEE 29th International Conference on. IEEE, 2011.

[18] Python pro ductivity for Zynq (Pynq) Documentation, Release 2.7, Xilinx, November, 2021

[19] Vivado Design Suite User Guide, UG910 (v2021.2) October 27, 2021.

[20] R.C. Gonzalez and R.E. Woods, Digital Image Processing, 2nd ed., Prentice Hall, 2002

[21] John C. Russ, F. Brent Neal, "The Image Processing Handbook, 7th Edition", CRC Press ISBN: 9781498740289, September 2018

[22] Rose, J. D. ., R, V. R. ., Lakshmi, D., Saranya, S. ., & Mohanaprakash, T. A. . (2023). Privacy Preserving and Time Series Analysis of Medical Dataset using Deep Feature Selection. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3), 51–57. https://doi.org/10.17762/ijritcc.v11i3.6201

[23] Thakre, B., Thakre, R., Timande, S., & Sarangpure, V. (2021). An Efficient Data Mining Based Automated Learning Model to Predict Heart Diseases. Machine Learning Applications in Engineering Education and Management, 1(2), 27–33. Retrieved from http://yashikajournals.com/index.php/mlaeem/article/view/17

[24] Pandey, J. K., Ahamad, S., Veeraiah, V., Adil, N., Dhabliya, D., Koujalagi, A., & Gupta, A. (2023). Impact of call drop ratio over 5G network. Innovative smart materials used in wireless communication technology (pp. 201-224) doi:10.4018/978-1-6684-7000-8.ch011 Retrieved from www.scopus.com