A LIGHT WEIGHT VLSI FRAME WORK FOR HIGHT CIPHER ON FPGA

Rajesh M S[1]

[1]*Assistant Professor, Dept. of Electronics and Communication Engineering, College of Engineering Adoor, Kerala, India*

---***---

Abstract - *The field of communication has acquired various applications in the digital age. In recent years, the number of low-resource devices has grown significantly. Security and resource constraints are two issues that come up with such communication devices. Low resource systems are the focus of encryption methods known as lightweight cyphers, such as HIGHT cyphers. Flexible and rapid design of light-weight cyphers are used for reconfigurable platforms like field-programmable gate arrays. The HIGHT cypher has easy operations and offers a sufficient level of protection. The implementation of a Hight cipher's lightweight VLSI design on an FPGA is presented in this study. It is appropriate for low-power and low complexity applications. The size, power, and speed of the suggested VLSI design have been optimized. For faster throughput, the encryption and decryption procedures might be carried out simultaneously. Performance results for the suggested architecture's FPGA implementation are given. The outcomes demonstrate the effectiveness of the suggested architecture, with a throughput of 25 Mbps and an encryption and decryption delay of 0.64 ms. Additionally the outcomes show that the suggested architecture is appropriate for low-power applications.*

Key Words: Light weight cryptography, Block ciphers, VLSI architecture, FPGA, Low resource device.

1. INTRODUCTION

A style of computer architecture known as lightweight VLSI focuses on combining lightweight and power-efficient components to minimize the system's overall size and power consumption. This kind of architecture is becoming more and more common as a result of its capacity to lower costs and power usage while maintaining a high degree of performance [1]. Applications for lightweight VLSI designs include embedded systems, mobile phones, and medical equipment. They are also utilized in high-performance computing because they can deliver performance that is superior to that of conventional designs while still using less electricity [2]. Digital signal processor, field programmable gate array (FPGA), application-specific integrated circuit, and system-on-chip designs are the core elements of a lightweight VLSI architecture. The architecture is created to make use of each of these components' strengths, such as FPGAs' low power needs and their capacity to handle several jobs at once [3]. Power efficiency is the most crucial aspect to consider while creating a lightweight VLSI architecture. This requires careful component selection and system design to ensure that overall power consumption is kept to a minimum. Applications based on intelligent low-resource devices have been significantly expanded recently. Internet of Things [4], radio frequency identification [5], smart cards, wireless body area networks, and wireless sensor networks are a few examples of such applications. Numerous design factors including area, security, power, and energy must be taken into account to get the maximum performance out of the implemented system.

For devices with less memory and processing capacity, conventional encryption techniques might not be suitable. This prompted the creation of the "lightweight cryptography" branch of crypto logic reasoning. The HIGHT algorithms were created expressly to fit the requirements of lightweight cryptology [6].

The "HIGHT" cypher, one of the well-known lightweight cyphers designed for low resource systems, must be highlighted before discussing the hardware FPGA implementation of lightweight cyphers. A hardware-oriented, lightweight block cypher is called HIGHT [7]. The HIGHT cipher's structure is a modification of the generalized Feistel network [8] and makes use of simple hardware-oriented transform operations. Key length of HIGHT algorithm is 128bit and block size is 64-bit.

In general, block cyphers can be implemented in hardware or software. Compared to software designs, hardware implementations are faster and use less energy. Field-programmable gate array (FPGA) designs are one approach for implementing hardware designs. The benefits of an FPGA design are flexibility and a short development time at a low cost [9]. Additionally, the FPGA's reconfigurability feature makes it a desirable choice for implementing cryptographic algorithms. Reconfigurability makes it easier to modify, upgrade, and swap cypher algorithms while still in use [10].

The design has been optimized to use the least amount of hardware resources and consume less power and energy. Although other performance indicators, such as speed, are equally crucial, power and energy are the main problems for low-resource systems. The HIGHT encryption was chosen because it has good energy and throughput performance and is designed for low-resource implementation [11].

2. LITERATURE REVIEW

The topic of the article [13] is the design of specialized VLSI intended for use in high-performance computing systems. General-purpose processor (CPU) and graphics accelerator (GPU) operate in the mode of general-purpose computing acceleration. For the design of energy adjustable circuits for open-minded application, approximate computing is gaining strength. A low power diminishing region square root (SQR) circuit is described in [14] and achieves incredible region and energy efficiency while exhibiting insignificant errors in the results. For the purpose of reestablishing cluster-based square-pull circuits for challenging applications, two approximated approaches are put out. By using the Boolean articulations, approximated reestablishing subtractor cells are used in the main configuration to replace specific SQR subtractor cells.

[15] Presents an equipment-improved predicted adder with essentially no typical error and a typical error distribution, or Gaussian error. The suggested adder is referred to by the authors as HOAANED, and it is expanded as equipment updated rough adder with typical error dispersion. For a fair analysis, authors took into account the use of HOAANED for computerized picture handling with the exact adder. In particular, the augmentations used precise and inexact adders separately to carry out rapid Fourier Change and retrograde quick Fourier Change operations to replicate the images. [16] Focuses on designing a Parameterizable SHA-256 algorithm implementation in an FPGA to teach blockchain concepts. The fundamental idea behind Blockchain design that adds security and anonymity to a system is SHA-256. This one-way hash function creates a singular result for a given input, guaranteeing the validity and nonrepudiation of the data. The decentralization of blockchain technology is causing it to become more and more popular online.

[17] Presents a 65-nm CMOS stage-locked circle with established sub stage-locked circle structure 32-GHz frequency-tweaked persistent wave modulator. With the sub stage-locked circle, the low-pass influence in stage space is acknowledged, lowering the impact of the delta sigma modulator's quantization disturbance and spikes. The stage space model and stage clamour model are broken down and imitated in order to achieve excellent solidity and stage commotion execution. These models are discussed and reproduced in light of the trill linearity, which aids in determining the plan boundaries and verifies the linearity improvement. In [18], a precisely stage-controlled transmitter operating in the 76 to 81 GHz band is introduced for the auto radar application. The multi-channel transmitter using this finder achieves a progressive work error of less than 0.6° root-mean-square in the 76 to 81 GHz frequency band. The fundamental functional unit to execute modular arithmetic in the different cryptography and pseudorandom bit generator (PRBG) methods provided in [19] is the three-operand binary adder. The method that is most frequently used to execute three-operand addition is called the carry save adder. The suggested design is synthesized using a 32nm CMOS technology library that is commercially accessible, and it is also implemented on an FPGA device for functional validation. For 32-, 64-, and 128-bit architectures, respectively, the post-synthesis results of the proposed adder reported 3.12, 5.31, and 9.28 times faster than the CS3A.

[20] introduced differential postpone component (DCDE) can achieve a stage shift of 20 ps and a normal goal of 1.25 ps thanks to the two information bits that control its inclination current and resistive burden and the two other pieces that organize its result capacitive burden inside the differential current-mode rationale (CML) DCDE. Using a low-power block-sharing without offset recurrence following circle (FTL) [21] shows how to align the cycle voltage-temperature varieties of the voltage-controlled oscillator (VCO) recurrence).

Many applications built using VLSI architecture have big size components that cause a design fault in the filter during floating point arithmetic stages. To increases design complexity and the time delay effect the architectural paradigm must be changed. The rising number of components, especially delay elements, is a problem in VLSI architectures for finite impulse response (FIR) filters [22]. This article offers a cross-folded shifting implementation of the floating point processing element for the VLSI architecture that has been altered to use less registers. The suggested FIR filter system enhances the complexity and high delay rate of the logical operation by reducing the number of components in the circuit.

For one-layered (1-D) wavelet-based electrocardiography (ECG) pressure frameworks with quality assurance, a speed and power-effective set apportioning in various leveled trees (SPIHT) plan is offered in [23]. To overcome the challenges of low coding speed and complex equipment design the authors first propose a coding-time and calculation effective SPIHT calculation using various types of coding status register record. Earlier SPIHT calculations are then described by the authors for dynamic calculation and course of action in the arranging and refinement handling stage that were necessary. You-only-look-once (YOLO) CNN is implemented in a Tera-OPS streaming hardware accelerator by [24] for real-time object identification with great throughput and power efficiency. FPGA can store the whole network model in its block RAMs thanks to the binary weight, which significantly reduces off-chip accesses and boosts performance. For better hardware consumption, the suggested approach fully pipelines each convolutional layer. The best power efficiency is achieved by this CNN that uses the VC707 FPGA and can handle 1.877 tera operations per second (TOPS) at 200 MHz with batch processing while only requiring 18.29 W of on-chip power.

[25] Makes a proposal for a closed structure articulation for determining the base operating voltage (VDDmin) of D flip-flops (FFs). The base inventory voltage at which the flip flops are functional without errors is referred to as VDDmin. The suggested explanation illustrates that VDDmin of FFs is a direct function of the square root of the logarithm of the number of flip flops, and that both its inclination and catch depend on the compatibility. The reconstruction results as well as the silicon estimations support the planned VDDmin articulation. Finally, the authors discuss VDDmin's reliance on device boundaries. Hybrid floating point (FP) executions

in [26] advance programming FP execution without creating a space overhead from FP units with all of the necessary equipment. The suggested executions are built into tiny fixed-point processor with a RISC-like architecture using 65-nm CMOS technology.

A technique in [27] allows for the effective use of field programmable gate array hardware resources while computing all of the CNN's layers. Compared to the basic GEMM-based design, it exhibits performance improvements in the range of 1.4 to 4.02 with only 13% more FPGA resources. Popular CNN models like AlexNet and VGG-16 have been mapped onto the suggested accelerator by the authors, and the measured performance compares favorably to other recent implementations [28]. The CNN is implemented using the VC707 FPGA, which has the best power efficiency, achieving a throughput of 1.877 tera operations per second. The CPU first divides large circuits into several segments, which are then transmitted one by one to the FPGA for random walking [29]. By directing individual part of the algorithms to the appropriate architecture and building the FPGA bit stream just once, the framework overcomes the problem of limited field programmable gate array on-chip resources and integrates both benefits of FPGA and CPUs. In order to increase the performance of FPGAs, various kernel optimization techniques are employed. In order to obtain the findings in [30], a hamming error detection and correction code has been constructed in Verilogger pro 6.5 after an algorithm for encoding and decoding hamming codes has been explained. Hamming code, which is implemented in Verilog, is an upgraded form of the parity check method that transmits n-bits of information data with redundancy bits. The design procedure for hamming code utilizing VLSI is attempted to be explained in this work because FPGA is a more cost-effective technology than the alternatives.

[31] Offer FPGA implementation of the AES algorithm based on high-level synthesis. The implementation, which makes use key of a 128-bit length, is ideal for telecom applications like 5G. This is due to their reconfigurable and parallel architectures. Because of their quicker speeds, FPGAs are also used as platform for software development in pre-silicon environment. High-Level Synthesis (HLS) tools are also heavily utilized by the design community in VLSI design flows.

Because of their small region, modest power, wide tuning range, and ease of scaling with process innovation, ring oscillators (ROs) are well recognized. However due to ineffective stage clamour and jittery execution, its usage in many applications is constrained. Warm loudness and glittering turbulence causes jitter, which, in contrast, lessens with fluctuating repetition. [32] Depicts a method for reducing jitter in ROs called recurrence assistance. Authors support the interior swaying recurrence in order to maintain the ideal result recurrence, and present a recurrence divider after the oscillator. This technique offers reduced jitter along with the opportunity to trade off yield jitter for power for dynamic execution of the board.

3. METHODOLOGY

A block cypher called HIGHT is appropriate for low-cost, low-power, and ultra-lightweight applications. The major features of HIGHT include block length of 64-bit, key length of 128-bit, 8-bit oriented operations, and a 32-round iterative structure. As a result, the encryption and decryption processes are identical, but the sequence in which the blocks are applied is reversed.

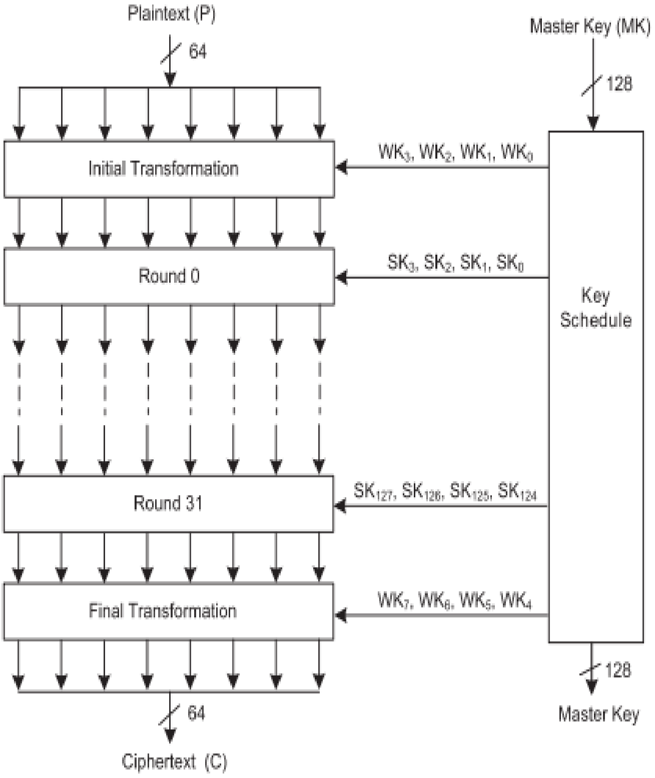


Figure 1: The HIGHT algorithm's encryption step.

HIGHT's encryption step includes a preliminary transformation, followed by transformations of 32 rounds and a final transformation. The simple operation mod 28 addition, bitwise rotations, and exclusive-or make up the majority of each of the 32 rounds. The main scheduling is divided into two parts: Generations of "whitening keys" (WKs) and "sub keys" (SKs). Four WKs are utilized in each of the initial and final transformations, which both use the WKs. The pseudo-random generated constants and mod 28 addition of the MK are utilized to create the SK that are used in the 32 round of transformation. Four SKs are utilized in each of the 32 rounds, totaling 128 SKs, which are created. WKs and SKs are the two key-generation algorithms used in the key schedule. Mapped to WK0, WK1, WK2, and WK3, respectively, are MK12, MK13, MK14, and MK15. The corresponding WK4, WK5, WK6, and WK7 codes for MK0, MK1, MK2, and MK3 are listed below. To maximize the hardware resource, including area, power, and energy of the fundamental design, the FPGA implementations using many rounds are addressed and investigated. The production of -terms and the execution of the number of rounds are the two main parts of the design optimization that are investigated. One of the primary design issues for the former is generating terms, thus we looked at two implementation techniques: the linear feedback shift register (LFSR) method [34] and the lookup-table (LUT) method. For the latter, both single-round and multiple-round implementations of the number of rounds in the hardware are explored.

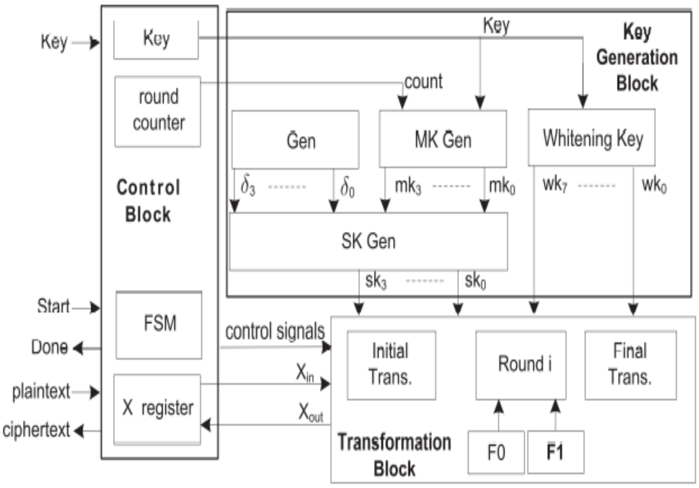


Figure 2: FPGA Design HIGHT

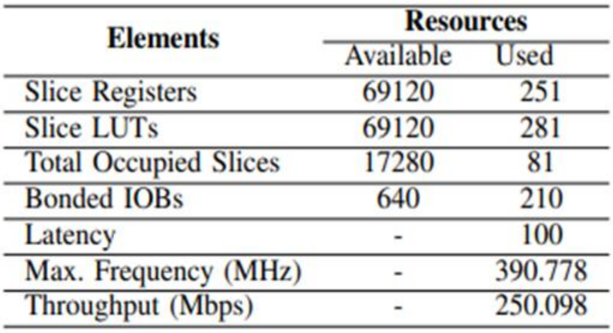
3.1 One round design for basic Field Programmable Gate Array

The architecture of the HIGHT Field programmable gate array implementation is depicts in figure 2 for one round, which is consist of control logic, key generation, and transformation. The interaction with the external system and executes the algorithm's activity sequencing capability managed by the control block. The control logic is instructed to start the encryption process by the start input signal, and the encryption is finished by the done output signal. The key, round, and X-register are the three registers are controlled by the control logic. 32 rounds involved in the round counter. The key generation block generates the WK and ensuing SK. The MK bytes are primarily used to construct the WK, and the terms and MK bytes are combined to create the SKs. The initial transformation, followed by the 32 round transformations, and the final transformation are all put into action by the transformation block. According to the LFSR implementation approach, each round should result in the production of four unique words. Therefore, each transformation round has four LFSR blocks. The usage of FF is often large and power consuming, is the fundamental drawback of the LFSR architecture.

3.2 Experimental Result

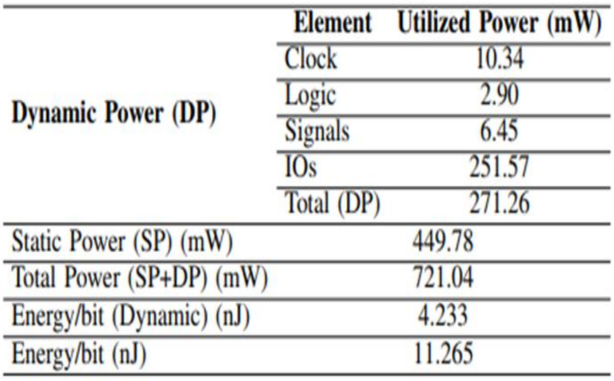
The suggested architecture has been successfully synthesized and implemented in Xilinx ISE Design Suite 14.7 using with the Xilinx Virtex-5 xc5vlx110t-1ff1136 serving as the target FPGA device. Table I includes the FPGA Device Utilization Summary as well as a number of other performance metrics. Results from simulation, synthesis, are presented. The Area Reduction design technique was used to achieve the benchmark results shown in TABLE I, with the speed set to -1 and all other settings are kept. TABLE I makes it very clear that the suggested architecture only employs 0.45 percent of the available Slice, of which 0.36 percent are Slice Register and 0.40 percent are Slice LUTs. When evaluated in various scenarios, the suggested design exhibits even greater qualities. When the speed setting is set to -2, the maximum frequency soars to 459 MHz, indirectly enhancing the throughput. Numerous additional metrics have been noticed. The reporting of results has, however, adopted a consistent methodology.

Table -1: FPGA Device Utilization and Performance Summary.



Elements	Resources	
	Available	Used
Slice Registers	69120	251
Slice LUTs	69120	281
Total Occupied Slices	17280	81
Bonded IOBs	640	210
Latency	-	100
Max. Frequency (MHz)	-	390.778
Throughput (Mbps)	-	250.098

Table -2: ON CHIP Power Summary



	Element	Utilized Power (mW)
Dynamic Power (DP)	Clock	10.34
	Logic	2.90
	Signals	6.45
	IOs	251.57
	Total (DP)	271.26
Static Power (SP) (mW)		449.78
Total Power (SP+DP) (mW)		721.04
Energy/bit (Dynamic) (nJ)		4.233
Energy/bit (nJ)		11.265

4. CONCLUSIONS

On an FPGA, a Lightweight VLSI Architecture for the HIGHT Cipher was successfully implemented. The HIGHT cypher encryption and decryption functionality was built into the architecture. The encryption and decryption throughput of the FPGA implementation was 25 Mbps with 0.64 ms latency. Both the area and the performance both exhibit improvement. A key size of 128 bits is also supported by the FPGA implementation, offering a high level of security. The suggested design offers a good throughput or area trade-off in compared to the current serial and parallel systems. The concept offers a quick, cost-efficient, and secure alternative for secure communication. For embedded systems that need good security and minimal power consumption, the suggested architecture is a suitable solution.

REFERENCES

[1] Yarlagadda, S., Kaza, S., chowdary Tummala, A., Babu, E. V., & Prabhakar, R. (2021). The reduction of Crosstalk in VLSI due to parallel bus structure using Data Compression Bus Encoding technique implemented on Artix 7 FPGA Architecture. Information Technology In Industry,9(1),456-460.

[2] Somashekhar, V. M., & Singh, R. P. (2020). FPGA implementation of fault tolerant adder using verilog for high speed VLSI architectures. International Journal of Engineering and Advanced Technology (IJEAT) ISSN, 2249-8958.

[3] Palchaudhuri, A., & Dhar, A. S. (2021). Speed-area optimized VLSI architecture of multi-bit cellular automaton cell based random number generator on FPGA with testable logic support. Journal of Parallel and Distributed Computing, 151, 13-23.

[4] Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. Computer networks, 54(15), 2787-2805.

[5] Rolfes, C., Poschmann, A., Leander, G., & Paar, C. (2008). Ultra-lightweight implementations for smart devices–security for 1000 gate equivalents. In Smart Card Research and Advanced Applications: 8th IFIP WG 8.8/11.2 International Conference, CARDIS 2008, London, UK, September 8-11, 2008. Proceedings 8 (pp. 89-103).

[6] Hong D, Sung J, Hong S, Lim J, Lee S, Koo BS, Lee C, Chang D, Lee J, Jeong K, Kim H. HIGHT: a new block cipher suitable for low-resource device. In Cryptographic Hardware and Embedded Systems-CHES 2006. Springer, 2006; 46–59.

[7] Knudsen, L. R. (2005, June). Practically secure Feistel ciphers. In Fast Software Encryption: Cambridge Security Workshop Cambridge, UK, December 9–11, 1993 Proceedings (pp. 211-221).

[8] Wollinger, T., Guajardo, J., & Paar, C. (2004). Security on FPGAs: State-of-the-art implementations and attacks. ACM Transactions on Embedded Computing Systems (TECS), 3(3), 534-574.

[9] Uthayakumar, C., Jijina, G. O., Suresh, G., & Nagaraju, V. (2021, July). FPGA Based Approximate Digital VLSI Circuit Validating Focused on Fault Diagnosis. In Journal of Physics: Conference Series (Vol. 1964, No. 6, p. 062079).

[10] Sowmya, N., & Rout, S. S. (2021). A Review on VLSI Implementation in Biomedical Application. In Innovations in Bio-Inspired Computing and Applications: Proceedings of the 10th International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA 2019) held in Gunupur, Odisha, India during December 16-18, 2019 10 (pp. 130-138).

[11] Lian, X., Liu, Z., Song, Z., Dai, J., Zhou, W., & Ji, X. (2019). High-performance FPGA-based CNN accelerator with block-floating-point arithmetic. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 27(8), 1874-1885.

[12] Kim, S., Na, S., Kong, B. Y., Choi, J., & Park, I. C. (2021). Real-time SSDLite object detection on FPGA. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 29(6), 1192-1205.

[13] Tarasov, I. E. (2019, October). Architectures of high-performance VLSI for custom computing systems. In Journal of Physics: Conference Series (Vol. 1333, No. 2, p. 022019)

[14] Arya, N., Soni, T., Pattanaik, M., & Sharma, G. K. (2020, January). Area and energy efficient approximate square rooters for error resilient applications. In 2020 33rd international conference on VLSI design and 2020 19th international conference on embedded systems (VLSID) (pp. 90-95). IEEE.

[15] Nayar, R., Balasubramanian, P., & Maskell, D. L. (2020, July). Hardware optimized approximate adder with normal error distribution. In 2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI) (pp. 84-89). IEEE.

[16] Devika, K. N., & Bhakthavatchalu, R. (2019, April). Parameterizable FPGA implementation of SHA-256 using blockchain concept. In 2019 International Conference on Communication and Signal Processing (ICCSP) (pp. 0370-0374). IEEE.

[17] Fu, Y., Li, L., Liao, Y., Wang, X., Shi, Y., & Wang, D. (2020). A 32-GHz nested-PLL-based FMCW modulator with 2.16-GHz bandwidth in a 65-nm CMOS process. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 28(7), 1600-1609.

[18] Fujibayashi, T., & Takeda, Y. (2019). A 76-GHz to 81-GHz, 0.6 RMS phase error multichannel transmitter with a novel phase detector and compensation technique. IEEE Solid-State Circuits Letters, 2(12), 277-280.

[19] Panda, A. K., Palisetty, R., & Ray, K. C. (2020). High-speed area-efficient VLSI architecture of three-operand binary adder. IEEE Transactions on Circuits and Systems I: Regular Papers, 67(11), 3944-3953.

[20] Rehman, S. U., Khafaji, M. M., Carta, C., & Ellinger, F. (2019). A 10-Gb/s 20-ps delay-range digitally controlled differential delay element in 45-nm SOI CMOS. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 27(5), 1233-1237.

[21] Yang, S., Yin, J., Mak, P. I., & Martins, R. P. (2018). A 0.0056-mm 2− 249-dB-FoM all-digital MDLL using a block-sharing offset-free frequency-tracking loop and dual multiplexed-ring VCOs. IEEE Journal of Solid-State Circuits, 54(1), 88-98.

[22] John, T. M., & Chacko, S. (2021). FPGA-based implementation of floating point processing element for the design of efficient FIR filters. IET Computers & Digital Techniques, 15(4), 296-301.

[23] Hsieh, J. H., Hung, K. C., Lin, Y. L., & Shih, M. J. (2017). A speed-and power-efficient SPIHT design for wearable quality-on-demand ECG applications. IEEE Journal of Biomedical and Health Informatics, 22(5), 1456-1465.

[24] Nguyen, D. T., Nguyen, T. N., Kim, H., & Lee, H. J. (2019). A high-throughput and power-efficient FPGA implementation of YOLO CNN for object detection. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 27(8), 1861-1873.

[25] Fuketa, H., & Matsukawa, T. (2017). A closed-form expression for minimum operating voltage of CMOS D flip-flop. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 25(7), 2007-2016.

[26] Pimentel, J. J., Bohnenstiehl, B., & Baas, B. M. (2016). Hybrid hardware/software floating-point implementations for optimized area and throughput tradeoffs. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 25(1), 100-113.

[27] Kala, S., Jose, B. R., Mathew, J., & Nalesh, S. (2019). High-performance CNN accelerator on FPGA using unified winograd-GEMM architecture. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 27(12), 2816-2828

[28] Sikka, P., Asati, A., & Shekhar, C. (2020, November). Area-optimal FPGA implementation of the YOLO v2 algorithm using High-Level Synthesis. In 2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON) (pp. 1-6). IEEE.

[29] Wei, X., Yan, C., Zhou, H., Zhou, D., & Zeng, X. (2019, March). An efficient FPGA-based floating random walk solver for capacitance extraction using SDAccel. In 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE) (pp. 1040-1045). IEEE.

[30] Shahariar Parvez, A. H. M., Mizanur Rahman, M., Podder, P., Hossain, M., & Islam, M. A. (2019). Design and implementation of hamming encoder and decoder over FPGA. In International Conference on Computer Networks and Communication Technologies: ICCNCT 2018 (pp. 1005-1022).

[31] Sikka, P., Asati, A. R., & Shekhar, C. (2020). Speed optimal FPGA implementation of the encryption algorithms for telecom applications. Microprocessors and Microsystems, 79, 103324.

[32] Lee, T. H., & Abshire, P. A. (2016). Frequency-boost jitter reduction for voltage-controlled ring oscillators. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 24(10), 3156-3168.

[33] Lim, Y. I., Lee, J. H., You, Y., & Cho, K. R. (2009). Implementation of HIGHT cryptic circuit for RFID tag. IEICE Electronics Express, 6(4), 180-186.

[34] Weste, N. H., & Harris, D. (2015). CMOS VLSI design: a circuits and systems perspective. Pearson Education India.