ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)

https://doi.org/10.17559/TV-20230404000502
Original scientific paper

Internet of Things Based Reconfigurable SIMD Processor for High-Speed End Devices in FPGA

Subathradevi SAMINATHAN, Ramadevi PONNIAH, Kavitha MUTHURATHINAM, Karthikeyan SOMASUNDARAM*

Abstract: This research article proposed the reconfigurable Single Instruction Multi Data (SIMD) processor design to speed up the accelerated computing task in IoT operations. Single Instruction Multi Data models leverage the parallel real source to speed up computing accelerated tasks. It proposes the utilization of reconfigurable Kogge Stone-dependent hybrid adder structures, now referred to as KS-CPA, in which reconfiguration occurs during the addition operation. The Least Significant Bits (LSB) are processed using a carry propagate adder, while the Most Significant Bits (MSB) are computed using the Kogge Stone adder. Depending on the data width and device-accessible energy resources, the hybrid configuration of the adder offers the 4-bit, 8-bit, and 16-bit addition. The adder form is identified by a shift in the configuration of its Carry Look-ahead and then by a Kogge Stone Adder (KSA). Throughout the activity, the KS-CLA crossbreed configuration is used to attain the fastest speed and low energy usage. The effectiveness, including its proposed hybrid adder, is evaluated by looking at the speed, energy, and area parameters, including a suitable area use during rapid applications in which both less delay and low power adders are required. Considering these, we are structuring an IoT processor that can be reconfigured to gain from SIMD. We have demonstrated that our hybrid adder-enhanced processor saves energy up to 13% and reduces 27% latency. The proposed 16 and 32-bit adders will boost time, power, and Area Delay Product (ADP) by almost 18-24% and 13-19% respectively.

Keywords: computational capabilities; hybrid adder; IoT; low power; reconfigurable; SIMD

1 INTRODUCTION

The emergence of the Internet of Things (IoT) is attributed to the strides made in sensor networks and ubiquitous computing, and it refers to intelligent technologies and the service environment that enable autonomous communications [1, 2]. In addition, multiple accessible sensors or flexible instruments can calculate human biological measurements directly by simple contact, including the heartbeat (HR), the breath rate (RR), and circulatory strain (BP). Even though it is always in the preliminary stage, the IoT intensity has been immediately received by organizations and enterprises, and updates have been produced along with client encounters [3]. However, the coordination of IoT innovation in medical services brings a few difficulties, including information storing, information management, information sharing between devices, safety and protection, combined and daunting exposure. Cloud Computing Technology is a potential solution to these problems. It demonstrates a standard medical services platform, which integrates IoT and cloud networking, to include general and clear information on a routine basis, existing on-demand resources across the network and the execution of tasks that are in problems [4].

IoT-based sensitive devices generate a great deal of information, which will be ready for measurements and applications for processing algorithms. Vital productivity is needed for increased performance, accuracy, and less expense for IoT-based smart devices. Such massive data can be stored, created and imaged from IoT. There are extremely helpful details and various possible properties in the sensor material. The system will have around 30 billion smart devices [5, 6]. However, the microcontroller or processor requires computing capacity to upgrade an essential device [7, 8]. This promotes better economic battery reinforcing tools for Field Programmable Gate Array (FPGA) [9]. The sensor hubs rely on the microcontrollers as the preparation capacity is limited. The FPGAs provide optimized phases and targets to chip away at economic power and vitality use. The FPGA Design has the productivity to upgrade all logic functions or delicate processors of very good quality, make the IoT devices power effective, track incongruous interfaces and help learn how to cope with the imminent growth of the execution of devices. An FPGA-based design solution may be used for this enormous complexity in usage [10, 11]. A significant amount of IoT devices are remote because the devices cannot be wired repeatedly. Some IoT devices have dimensional limitations and are built exclusive of a power cord, so the battery space is restricted. Strengthening this device is a challenging beggar.

Now, the significant components of the IoT systems are defined because it is necessary to consider the elevated sensor location under hazards concerning IoT devices and applications. An IoT system can be described as a network of equipment, scheduling, availability arrangements, and sensors in the grid of deviating things. Therefore, IoT Hardware Design comprises four essential segments: detection, network, information handling, and application phase [12, 13].

The chapter 2 discusses the survey based on existing SIMD processor designs. Section 3 discusses the function of various hybrid adders. Section 4 discusses the suggested hybrid adders-based SIMD processor's working function. Section five discusses the suggested system's simulation results and performance analysis. Finally, section six discusses the work's conclusion and future scope.

2 LITERATURE REVIEW

The computational complexity and power consumption collectively emphasize the growing interest in a straightforward and short-term evaluation strategy. Using Application Specific Processors (ASP), where a design processor becomes unique to the successful implementation of a particular program or for other purposes, may provide a conceivable scenario for achieving the systemic objectives listed above. It is carried out with different specific instructions or with dedicated hardware co-processors by extending the baseline processor. The successive ASP may be specifically degraded for the execution and equipment assets [14].

In carrying out the bulk of the numerical operations, for example, the adders, which are among the arithmetic and logic units (ALUs), are used [15]. The delay from such systems often has a fundamental task of deciding the replication of the operating clock by the processors. Likewise, their power arrangement is a significant part of the power usage in the processor's device centres because of their strong uses. Therefore, preparing competent adders may result in the improvement of the presentation and a decrease in the power of processors. Several concurrent and equivalent structure-dependent systems have often been suggested to enhance different parameters [16, 17].

Those three initial actions only generate a 'Sum', which is single digit long, but the pairing amount comprises two digits as both the bits and the numbers applied are equal to 1. The most critical element of the test is regarded as the carry'. A hybrid circuit that extends the spread of two bits is known as a half-adder, while one that extends the spread of three bits is known as a full-adder [18].

2.1 Ripple Carry Adder

The Full Adder (FA) has become a combination of two operand bits, and a transmitting bit is independently A_{in}, B_{in}, and C_{in} as data outlets and gives (*Sum*) and Carry bit (C_{out}) as performance. This Carry Bit C_{out} returns as the Carry Bit details for the progressive full adder. It demonstrates the sum and carry, including its combinational circuit acquired under Boolean conditions 1 and a 4-bit ripple carry adder from the full adder. For the low operand adder, the extremely easy usage of two operands, A_{in} and B_{in}, is through by falling n of the primary full adder components and has been considered a ripple carry adder. The adder configuration is fast and simple to use but has real issues with delays since for Carry bit from the previous stage of the full adder, the next stage must be done [19, 20].

$$\begin{aligned} Sum &= (A_{\text{in}} \oplus B_{\text{in}}) \oplus C_{\text{in}} \\ C_{\text{out}} &= (A_{\text{in}} \bullet B_{\text{in}}) + (C_{\text{in}} \bullet (A_{\text{in}} \oplus B_{\text{in}})) \end{aligned} \tag{1}$$

2.2 Carry Select Adder (CSA)

For the input data carry (C_{in}), the Ripple carry adder holds tight and then stretches the 'sum' and the 'C_{out}' sequence. The carry select adder is added to decrease the delay, where the 'sum' and 'carry' are used for the two different instances, for example, $C_{\text{in}} = 0$ and $C_{\text{in}} = 1$. A multiplexer choosing the correct output based on the C_{in} from the last stage would be provided with their determined sum. Such pre-calculation in sum decreases a period delayed by Carry ripple to just one multiplexer from every stage. There is a ripple carry adder in each 4-bit adder. The carry select adder has more hardware but provides less delay than the ripple carry adder [21, 22]. Therefore, an important analytical among area, power, and delay of various adders.

2.3 Carry-Skip Adder

The adder is extracted from a selection of the adder with a by-pass and using a ripple carry adder for the adder [23]. By reducing the most optimistic delay, the creation of the carry skip adder building block is well-known. It illustrates the construction of a 16-bit carry skip adder. The extension of the carry skip adder is done at various levels. There is an RCA block, a multiplexer, and a carry estimation device at any point. RCA has been used for every stage to explore [24] which requires the binary number A_{in} and B_{in} data bits, and condition 2 produces the whole sum.

$$\begin{aligned} p_i &= (A_{\text{in}} \oplus B_{\text{in}}) \\ g_i &= (A_{\text{in}} \bullet B_{\text{in}}) \end{aligned} \tag{2}$$

In the generated signal is propagate p_i, and distribute g_i to the i-th adder cell.

It is expressed as 3 from the adder unit.

$$\begin{aligned} Sum &= (p_i \oplus C_{\text{in}}) \\ C_{\text{in}+1} &= g_i + p_i C_{\text{in}} \end{aligned} \tag{3}$$

where C_{in} is the carry input to the i-th cell.

2.4 Carry Look-Ahead Adder

Each once in a while, various approaches were proposed to reduce the specific delay of parallel addition [25]. Yet another method of this is determining the "*Sum*," "C_{out}," and the terms "generate (g_i)" and "propagate" [5, 6] by using conservative conditions. For example, if the carry-out is, the term "1" is compatible because all the data sources A_{in} and B_{in} are '1' and therefore $g_i = (A_{\text{in}} \bullet B_{\text{in}})$. Having a term provides free carry-out of a carry-in. The word shifts the input carry bit to the output carry, when neither of the data sources is large, but rather, the phrase propagate is defined as $p_i = (A_{\text{in}} \oplus B_{\text{in}})$ in expression 2. The model is also distinctly described below, indicating the principle of "Generates and Propagate". Throughout the propagate scenario, the "C_{out}" is dependent on the "C_{in}," e.g. when "C_{in}" is 0 "C_{out}," and when "C_{in}" is 1 "C_{out}" is 1, whatever the "C_{in}" becomes consistently 1. The output "*Sum*" and "C_{out}" of the full adder can be seen as far as p_i and g_i.

The logic of carry look-ahead specifies whether the bit pair adds or separates a carry. This helps the circuit to add two numbers before processing, which evaluates the progress carried ahead early. There is no postponement from tight holding for the ripple carry impact in these lines when the genuine expansion is carried out [26].

The carry look-ahead added is a sort of a ripple carry adder. To speed up an adder, this stage must be updated to quickly carry propagation. This helps the circuit to add two numbers before processing to determine the carry propagation early.

2.5 Ripple Carry Adder

A prefix adder occupies 3 phases, that is to say, the pre-calculation, prefix network, and a post-calculation.

The Pre-Calculation workouts for each input pair, as provided in Condition 2, with 'Propagate' and 'Generate' bits. The network prefix phase is the last to carry bits

'Propagate' and 'Generate. The computation of the carry-out is modified to a prefix problem using the associated operator '○'.

The sets of 'Generate' and 'Propagate' bits are given below:

$$(g_i, p_i)\circ(g_i^{'}, p_i^{'})=(g_i+p_i\bullet g_i^{'},\ p_i\bullet p_i^{'}) \tag{4}$$

where the terms 'Generate' represent g_i and g_i' and p_i and p_i' correspond to the terms 'Propagate'. The operator '○' may be configured back to back 'propagate' and 'generate' sets for carry which follows,

$$C_{in}=(g_i, p_i)\circ(g_{i-1}, p_{i-1})\circ\cdots(g_1, p_1)\circ(g_0, p_0) \tag{5}$$

The parallel prefix of carry calculation can be shown as graphs, which represent the operator '○' as hub '●', and signal sets (g_i, p_i) as the edges of a graph. The green shading node in the map refers to an excessive product through a hub with no logic. Last post-calculation process calculates the last sum from the prefix network step generated carry. In comparison to the Kogge stone of the carry look-ahead, these techniques are extremely skilled in delay and power. The growth in operand size (16-bit or more) contributes to the unpredictability of the prefix network related to an expansion of various logic cells and wiring [27, 29].

With the development of Internet of Things (IoT), low-cost, monitored resource devices need to be more skilled than previous embedded systems, which follow strict power constraints. The power consumption planning must be changed to integrate new capabilities such as learning. Approximate calculations are promising examples for reducing power consumption while introducing errors into calculations. In this work, a Single Instruction Multiple Data (SIMD) processor array for migrating machine learning applications into Field Programmable Gate Array (FPGA) IoT edge devices is presented for medical application.

3 PROPOSED HYBRID ADDERS

We use the reconfigurable hybrid adder in an ALU to promote clarification of our methodology. Note that the hybrid adder of every Sum-of-Product component is protected by our technique. In this arrangement, the partial product tree decreases produce two n-bit Dout0 and Dout1 wide output vectors which contribute to the last CPA propagation. The incomplete element decreases a distorted timing profile at the performance. Because of this skewed appearance pattern for the propagated adder, some traditional, independently-designed expansion plans may not work excellently in regard to a sum of products. There are four adder blocks in our proposed synthesized hybrid adder module. Through LSB to MSB, they are:

- Adder 1 & 2: a carry southeast adder [7:0] near the LSB.
- Adder 3 & 4: A rapid Kogge Stone adder close to the MSB for the next s_{out}.

When a combined adder has the most basic output, it is T_{a} = Max (T_{a2sout}, T_{a3sout}, T_{a4sout}, T_{a4cout}). If the bit width of Adder3 is $(n - w_1 - w_2)/2$ bits and the bit-width of Adder4 $(n - w_1 - w_2)/2$ and the value of T represents the SOP block's most basic output signal will be prepared. Currently, two combined expansion methods related to above are available in Fig. 1 to generate a further updated specification for addition. We used both the Carry look-ahead and Kogge Stone Attachment approaches and provided an alternative solution that is greatly faster than the others.

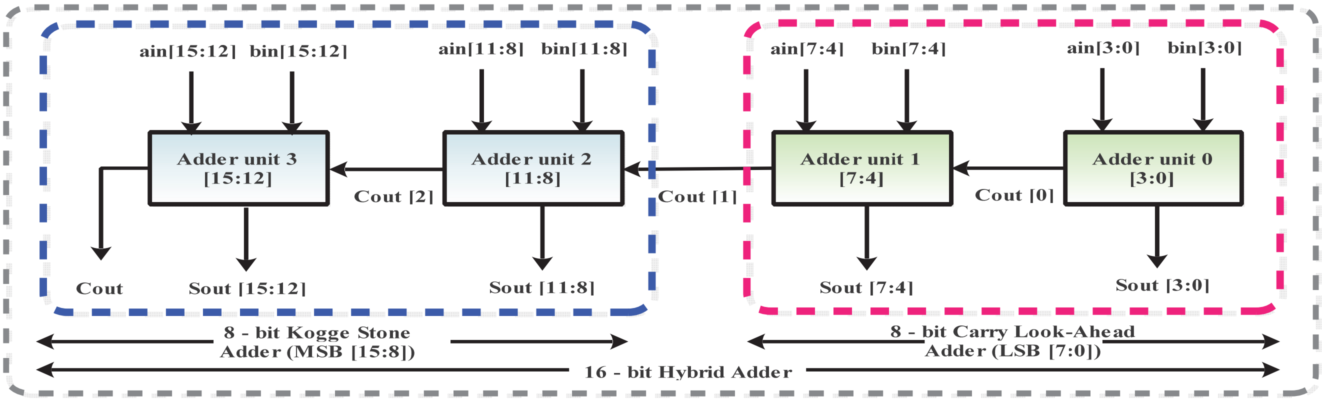

Figure 1 Fast four-stage hybrid adder

4 RECONFIGURABLE PROCESSOR FOR IOT

In order to speed up computing accelerated tasks, multi-data single instruction (SIMD) models leverage the parallel real source. Similar procedures are used for multiple information on different execution units that are strongly influenced by one directive at the same time. A Fast Fourier Transform (FFT) Vector Function Unit (VFU) is scheduled to be used in various remote correspondence instructions. VFU is capable of concurrently running sixteen 16-bit MAC or eight 32-bit MACs. The memory capacity becomes extended to allow efficient exchange of data among that vector file register as well as the main memory. The load-store SIMD instructions logically and arithmetically carry out data processing in parallel on vectors of 8 or 16 elements, as shown in the size of the pixel block used in one picture when the Advanced Video Coding (AVS) format is used. The research towards hardcore for measurement, such as motion compensation, deblocking, etc., is greatly improved by the SIMD instructions. With the addition of customized functional units, respectively low power consumption and superior efficiency will increase. SIMD designs are carried out to speed-up the broad process strategies of fine-grained parallelism. Along with the multicenter framework, a similar quad-core Intel Core i7 3820 CPU consumes 25 times less power [30].

The 16-bit ALU core of the SIMD processor. In this Reconfigurable ALU (RALU), 2's complement calculations for Radix2 are performed. Two clocks are required for RALU operation. In order to load values in registers, the primary clock cycle is used. The next move is to complete the activities.

To determine the capacities, 6-bit opcodes are used. There is an 18-bit instruction set, including the opcode. In the RALU, the delay of each point will be one cycle, which corresponds to the delay of the RALU. The 5 traditional phases without a pipeline are IF, ID, EX, MEM, and WB. A 10-bit address is sent to a Block-RAM instruction (BRAM) for 18-bit instruction in stage IF. The stage ID decodes the commands, and part from the control registers is configured to verify the corresponding steps. RALU manages data in registries, through stage EX, which carries out other control instructions, such as loop or jump. Throughout the MEM stage, the information will be read from / written in contact with BRAM information by instruction and address, when the instruction is store or load. The information is eventually written in the stage WB for registration. On the interface of the processor, uncover the pins for a clock, reset, address, details, and BRAM control. The processor architecture is shown in Fig. 2.

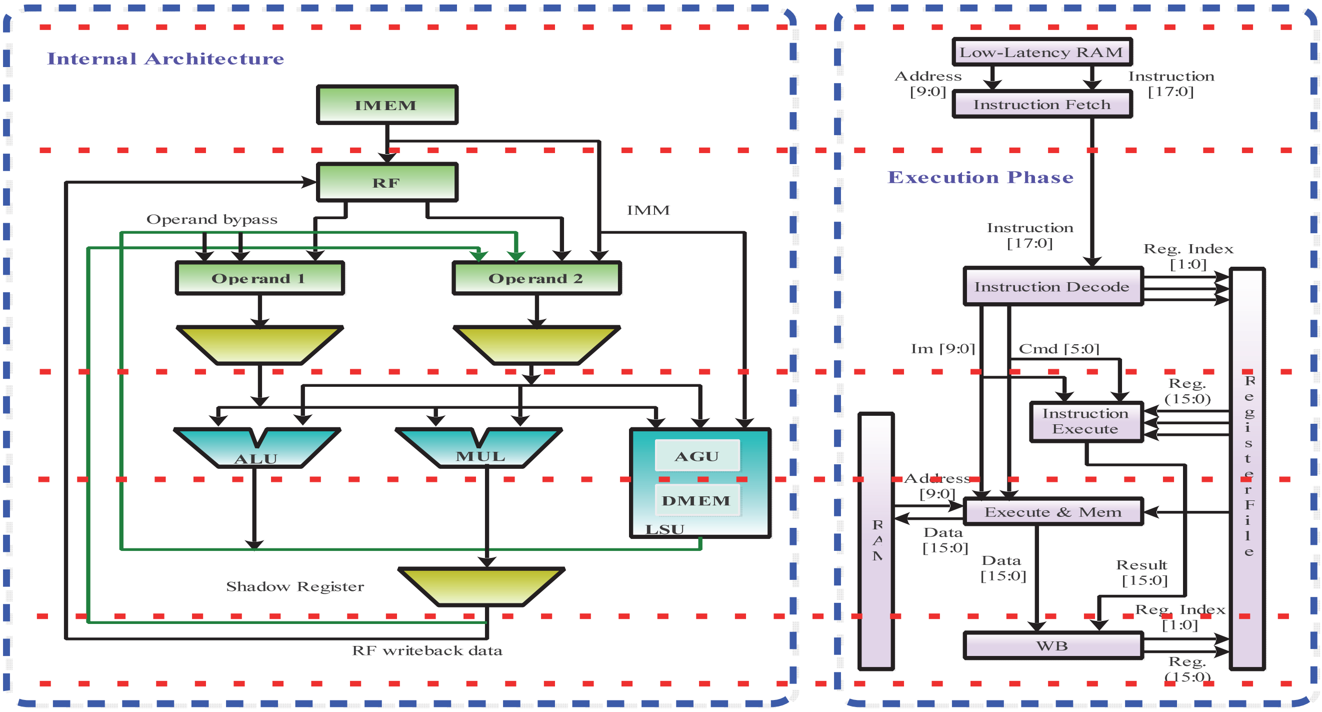

Figure 2 SIMD processor internal blocks and pipeline execution architecture

The SIMD Hybrid Adder, the multiplier and the shifter are three basic processing units in the ALU. These three segments can be replicated in various widths (4, 8, and 16-bits) for computing data and can answer all the instructions throughout the design. All the ports are bounded by three input ports: Q (4-bit), O (8-bit), and H (16-bit), demonstrating a type of information to be used appropriately by the unit.

5 RESULTS AND DISCUSSION

The proposed Hybrid Adders were modeled using the terminology of Verilog hardware definition language and using Xilinx 14.7 software for the pre-simulation evaluation.

The adder has been implemented throughout the hardware processors listed in Tab. 1 (Spartan 6 xc6slx-3csg324). Implemented and in contrast with the proposed hybrid adders are the Ripple Carry Adder (RCA), Carry Select Adders (CSA), Carry Skip Adders (CSkA), Carry Look-Ahead Adders (CLA), and Kogge Stone Adders (KSA). The relation involved various input measurements. The delay analysis of different existing adders is shown in Fig. 3.

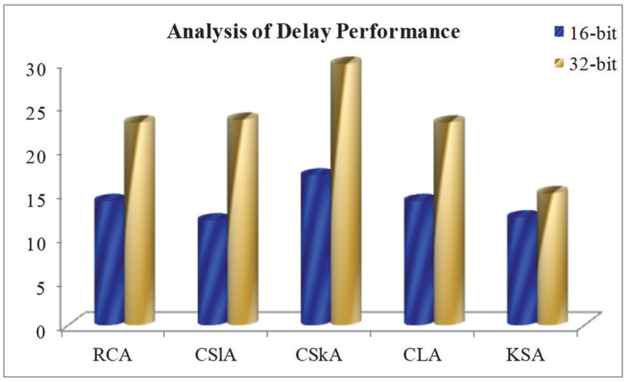

Figure 3 Delay analysis of different existing adders (16 & 32-bit)

Within this article, four adders along with the adder Kogge Stone were suggested, including the configurations KS-RCA, KS-CSlA, KS-CSkA, and KS-CLA,applied in 2 data widths of 16 and 32 bits between this setup. In comparison to different arrangements, KS-CLA offers superior performance during this investigation.

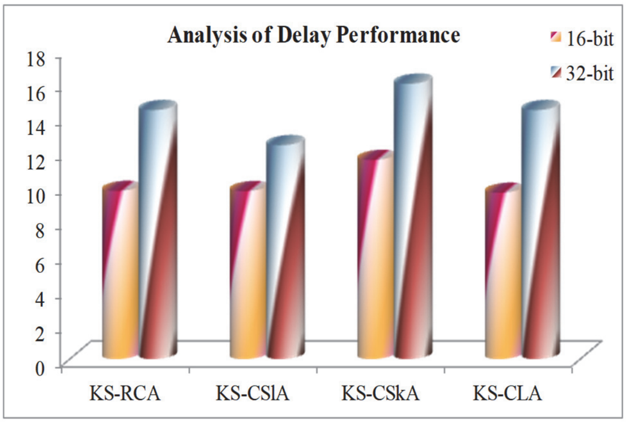

Figure 4 Delay analysis of different hybrid adders (16 & 32-bit inputs)

The results of the post-reactivation investigation introduced in this area are achieved with the helpful standard CMOS library, CADENCE Genus, with 90 nm gpdk. Without restricting the delay and area, test results were obtained. To quantify total power like static and dynamic control, the Cadence Encounter intensity advertisement tool was used.

Tab. 3 presents the strong analytic for different existing adders and the hybrid 16-bit adder proposed. The new 16-bit adder improves power usage by approximately 14.5%, and the latency is reduced by 53% according to the performance. In comparison with the additional current adders and hybrid vendors, improvements to the Area-Delay product (ADP) are about 796.49 and 0.0324 to the Power Delay Product (PDP).

The 32-bit adder and associated expertise results are listed in Tab. 1. The proposed 32-bit adder, as shown by performance, improves energy usage by up to 13.2% and reduces the latency by 33%. The Area-Delay Product (ADP) updates are about 3216 while the Power Delay Product (PDP) is around 0.031 as opposed to other current adder and hybrid Kogge-Stone adders.

Table 1 Performance analysis of different hybrid adders (16-bit inputs)

16-bit inputs	Gates	Area / µm^2	Power / nW			Delay / ps	ADP	PDP / pJ
			Static	Dynamic	Total			
Typical Adders								
RCA	21	457	1384	13518.578	14072.683	4006	1779	0.0654
CSlA	51	752	3133	26787.813	29120.376	2010	1003	0.0523
CSkA	33	519	2059	15199.569	16768.505	5102	2123	0.0792
CLA	65	539	3156	15688.602	17026.732	4186	1894	0.0656
KSA	97	832	6225	25406.227	29830.952	895	812	0.0351
Hybrid Adders								
KSA-RCA	50	489	4275	16562.185	19237.54	2123	799	0.0422
KSA-CSlA	58	578	4598	12049.696	23347.507	2002	762	0.0422
KSA-CSkA	52	507	4232	17381.24	19288.689	3154	1003	0.0521
KSA-CLA	72	657	5053	18196.911	21133.915	1800	816	0.0434

Since a detailed examination, Hybrid KS-CLA, KS-CSlA, HC Adder [27], Hybrid HC Adder [27], Hybrid Sum Producer [17], Kogge-Stone [b], PPF/CSSA4_4 [17], and PPF/CSSA6-3 [28] are also considered. In addition, a comprehensive evaluation will be performed. The configuration of the Hybrid KS-CLA is built with this gate and some essential NOT gates in carry generation because of Kogge-Stone emphasis on 2-input NAND gates. With regard to performance in Tab. 2, there is a strong explanation that, in comparison with [27, 17, 28], the proposed 32-bit KS-CLS adder will improve the energy by approximately 43%. The improvements of the Area-Delay Product (ADP) are about 1952.43, and 53% when compared to [17], and [28], respectively. The proposed 32-bit adder not only has less delay but also enhances the energy consumption and the circuit area.

Table 2 Performance analysis of different hybrid adders (32-bit input)

Adders	N-bits	Technology	Gates	Area / µm^2	Power / mW	Delay / ns	ADP	PDP
KSA-CSlA	32	GPDK 90 nm	139	1293	0.0595	1.51	1952.43	0.0898
KSA-CLA	32	GPDK 90 nm	157	1118	0.0491	3.216	3595.488	0.158
HC adder [a]	32	Free PDK 45 nm	940	1299.02	0.4459	-	-	-
Hybrid HC Adder [a]	32	Free PDK 45 nm	855	1147.44	0.4052	-	-	-
Hybrid Sum-Product [b]	64	TSMC 65 nm	-	10383	6.391	0.561	5824.863	-
Kogge-Stone [b]	64	TSMC 65 nm	-	14424	9.621	0.598	8625.552	-
PPF/CSSA4-4 [c]	64	Free PDK 45 nm	-	2016	7.508	0.638	1286.208	0.418
PPF/CSSA6-3 [c]	64	Free PDK 45 nm	-	2017	714.9	0.63	1270.71	0.45

Tab. 3 offers observational findings and results with relevant experience for the reconfigurable processor suggested. The findings indicate a 13% improvement of the energy benefits of the new processor and a 10.7% decrease in the delay of contrast with the current system.

The delay in the proposed RALU is reduced to the traditional structure for this architecture while its delay and power are as low. In consideration of the others, the key benefit of the new processor is the decreased expense, which results in less latency and less power usage.

Table 3 Performance analysis (16-bit inputs)

Processor	Gates	Area / µm^2	Power / nW			Delay / ps
			Static	Dynamic	Total	
Conventional Adder	5020	28958	156260	11064626.5	11220425.64	4802
Hybrid Adder	4897	29459	152660	10885895.8	11038556.49	4562

6 CONCLUSIONS

The growing demands for system efficiency throughout this rapid computer world were a constant pattern. This is the measure with any integrated processor architecture that consistently corresponds to such a pattern. This evaluation has designed and developed the integrated adder appropriate for the SIMD system. Significant parameters were determined and analyzed, particularly the delay reduction and dynamic power utilization. In contrast, the different current adder architectures and the proposed hybrid adder are applied and evaluated. Around a 53% decrease in delay and 14.5%, lower power utilization was accomplished using the hybrid adder. The adder of the carry bits is generated and propagated to enhance their execution in the proposed reconfigurable structure. Research findings suggest that, in combination with the relevant best of the rest, the proposed 16 and 32-bit adders will boost time, power, and ADP by almost 18-24%, and 13-19% separately. The possibilities have been demonstrated to decrease delays and power with smart creativity. This demonstrates that reconfigurable processor methods are successful with the capabilities for IoT applications to be enforced. In future reversible adder and multiplier logic were designed for IoT operations

7 REFERENCES

[1] Hyun-Jong, C., Ho-Kyung, Y., & You-Jin, S. (2021). The Detection Data Processing Mechanism for Vehicular Cyber Physical System in IoT Environment. *Technical Gazette*, *28*(3), 963-973. https://doi.org/10.17559/TV-20201209230220

[2] Elhoseny, M., Ramírez-González, G., Abu-Elnasr, O. M., Shawkat, S. A., Arunkumar, N., & Farouk, A. (2018). Secure medical data transmission model for IoT-based healthcare systems. *IEEE Access*, *6*, 20596-20608. https://doi.org/10.1109/ACCESS.2018.2817615

[3] Wu, T., Wu, F., Redoute, J. M., & Yuce, M. R. (2017). An autonomous wireless body area network implementation towards IoT connected healthcare applications. *IEEE Access*, *5*, 11413-11422. https://doi.org/10.1109/ACCESS.2017.2716344

[4] Sungwon, L., Hyeonkyu, J., Gihyun, P., & Jonghee Y. (2021). Design of Automation Environment for Analyzing Various IoT Malware. *Technical Gazette*, *28*(3), 827-835. https://doi.org/10.17559/TV-20210202131602

[5] Rahimi Moosavi, S., Nguyen Gia, T., Rahmani, A. M., Nigussie, E., Virtanen, S., Isoaho, J., & Tenhunen, H. (2015). SEA: a secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways. *Procedia Computer Science, Elsevier*, *52*, 452-459. https://doi.org/10.1016/j.procs.2015.05.013

[6] Zhang, J., Zong, Y., Yang, C., Miao, Y., & Guo, J. (2019). LBOA: Location-Based Secure Outsourced Aggregation in IoT. *IEEE Access*, *7*, 43869-43883. https://doi.org/10.1109/ACCESS.2019.2908429

[7] Tresanchez, M. T., Pujol, A. P., Pallejà, T. P., Martínez, D. M., Clotet, E. C., & Palacín, J. P. (2019). An inexpensive wireless smart camera system for IoT applications based on an ARM CORTEX-M7 microcontroller. *Journal of Ubiquitous Systems and Pervasive Networks*, *11*(2), 1-8. https://doi.org/10.5383/JUSPN.11.02.001

[8] Afsarimanesh, N., Mukhopadhyay, S. C., & Kruger, M. (2019). IoT-Enabled Microcontroller-Based System. *Electrochemical Biosensor: Point-of-Care for Early Detection of Bone Loss*, *Springer, Cham*, *30*, 93-103. https://doi.org/10.1007/978-3-030-03706-2

[9] Elnawawy, M., Farhan, A., Al Nabulsi, A., Al-Ali, A. R., & Sagahyroon, A. (2019). Role of FPGA in Internet of Things Applications. *2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), IEEE*, 1-6. https://doi.org/10.1109/ISSPIT47144.2019.9001747

[10] Liu, Y., Briones, J., Zhou, R., & Magotra, N. (2017). Study of secure boot with a FPGA-based IoT device. *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), IEEE*, 1053-1056.

[11] Huang, S. Z. & Chen, R. Q. (2018). FPGA-Based IoT Sensor HUB. *2018 International Conference on Sensor Networks and Signal Processing (SNSP), IEEE*, 139-144. https://doi.org/10.1109/SNSP.2018.00034

[12] Zaidan, A. A., Zaidan, B. B., Qahtan, M. Y., Albahri, O. S., Albahri, A. S., Alaa, M., & Lim, C. K. (2018). A survey on communication components for IoT-based technologies in smart homes. *Telecommunication Systems*, *69*(1), 1-25. https://doi.org/10.1007/s11235-018-0430-8

[13] Tran, C. & Misra, S. (2019). The Technical Foundations of IoT. *IEEE Wireless Communications*, *26*(3), 8-8. https://doi.org/10.1109/MWC.2019.8752474

[14] Sivanandam, K. & Kumar, P. (2018). Low-Power High-Performance Multitransform Architecture Using Run-Time Reconfigurable Adder for FPGA and ASIC Implementation. *System and Architecture*, 63-72.

[15] Reddy, K. S. & Suresh, H. N. (2018). A low-power VLSI implementation of reconfigurable FIR filter using Carry Bypass adder. *International Journal of Intelligent Engineering and Systems*, *11*(2), 225-36. https://doi.org/10.22266/ijies2018.0430.25

[16] Kabra, N. K. & Patel, Z. M. (2019). Low-power and high-speed configurable arithmetic and logic unit. *Innovations in Electronics and Communication Engineering*, 355-363.

[17] Jafarzadehpour, F., Molahosseini, A. S., Zarandi, A. A. E., & Sousa, L. (2019). New energy-efficient hybrid wide-operand adder architecture. *IET Circuits, Devices and Systems*, *13*(8), 1221-1231. https://doi.org/10.1049/iet-cds.2019.0084

[18] Mewada, M. & Zaveri, M. (2016). An input test pattern for characterization of a full-adder and n-bit ripple carry adder. *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE*, 250-255. https://doi.org/10.1109/ICACCI.2016.7732055

[19] Bais, K. & Ali, Z. (2016). Comparison of various adder designs in terms of delay and area. *International journal of science and research (IJSR)*, *5*(5), 1292-1295. https://doi.org/10.21275/v5i5.nov163657

[20] Devi, A. B., Kumar, M., & Laishram, R. (2016). Design and Implementation of an improved carry increment adder. *arXiv preprint arXiv:1603.04094*, *7*(1), 21-27. https://doi.org/10.48550/arXiv.1603.04094

[21] Saxena, P., Purohit, U., & Joshi, P. (2013). Analysis of low power, area-efficient and high speed fast adder. *International Journal of Advanced Research in Computer and Communication Engineering*, *2*(9), 3705-3710. https://doi.org/10.5281/ZENODO.32567

[22] Ykuntam, Y. D., Rao, M. N., & Locharla, G. R. (2013). Design of 32-bit carry select adder with reduced area. *International journal of computer applications*, *75*(2), 47-52. https://doi.org/10.5120/13086-0353

[23] Bahadori, M., Kamal, M., Afzali-Kusha, A., & Pedram, M. (2015). High-speed and energy-efficient carry skip adder operating under a wide range of supply voltage levels. *IEEE Transactions on very large scale integration (VLSI) systems*, *24*(2), 421-433. https://doi.org/10.1109/TVLSI.2015.2405133

[24] Hamdi, B., Khalifa, K. B., & Fradi, A. (2013). Hybrid-CMOS logic style design for fast self-checking adders data paths. *International journal of computers and technology*, *10*(6), 1771-1778.

[25] Yang, T., Sato, T., & Ukezono, T. (2020). An Accuracy-Configurable Adder for Low-Power Applications. *IEICE Transactions on Electronics*, *103*(3), 68-76. https://doi.org/10.1587/transele.2019LHP0002

[26] Mitra, A., Bakshi, A., Sharma, B., & Didwania, N. (2015). Design of a High-Speed Adder. *International Journal of Scientific and Engineering Research*, *6*(4). https://doi.org/10.5281/zenodo.33243

[27] Rakesh, S. & Grace, K. V. (2019). A Comprehensive review on the VLSI design performance of different Parallel Prefix Adders. *Materials Today: Proceedings*, *11*, 1001-1009. https://doi.org/10.1016/j.matpr.2018.12.030

[28] Cui, X., Liu, W., Wang, S., Swartzlander, E. E., & Lombardi, F. (2018). Design of high-speed wide-word hybrid parallel-prefix/carry-select and skip adders. *Journal of Signal Processing Systems*, *90*(8), 409-419.

[29] Pinto, R. & Shama, K. (2019). Low-Power Modified Shift-Add Multiplier Design Using Parallel Prefix Adder. *Journal of Circuits, Systems and Computers*, *28*(2), 1950019. https://doi.org/10.1142/S0218126619500191

[30] Vinoth Kumar, K., Karthikeyan, S., Bhavani, S., & Vennila, C. (2022). Performance Improvement of SIMD Processor for High-Speed end Devices in IoT Operation Based on Reversible Logic with Hybrid Adder Configuration. *Technical Gazette*, *29*(1), 252-258. https://doi.org/10.17559/TV-20210719025814

Contact information:

Subathradevi SAMINATHAN, Assistant Professor
Department of Electronics and Communication Engineering,
BIT Campus, Anna University, Tiruchirappalli

Ramadevi PONNIAH, Associate Professor
Department of Electronics and Communication Engineering,
BIT Campus, Anna University, Tiruchirappalli

Kavitha MUTHURATHINAM, Assistant Professor
Department of Computer Science and Engineering,
K. Ramakrishnan College of Engineering, Tiruchirappalli

Karthikeyan SOMASUNDARAM, Research Scholar
(Corresponding author)
Department of Electronics and Communication Engineering,
SSM Institute of Engineering and Technology,
Dindigul, Tamil Nadu, India-624002
E-mail: thiyan0786@gmail.com