United International Journal of Engineering and Sciences
(UIJES – A Peer-Reviewed Journal); ISSN:2582-5887 | Impact Factor:8.075(SJIF)
Volume 5 | Special Issue 1 | 2025 Edition
National Level Conference on "Advanced Trends in Engineering Science & Technology" – Organized by RKCE

AN ENERGY EFFICIENT HIGH SPEED MULTI OPERAND BINARY TREE ADDER

Dr. R DIVYA[1]
Assistant Professor[1], [234]UG scholar
M. MADHUALA[2], K. SAI SHIVANI[3], T. PUJITHA[4], K. PAVANI[5]
Department of Electronics Communication Engineering R K College of Engineering
Vijayawada, India

Abstract:
This project examines the critical path of a binary tree adder (BTA) that is based on a ripple carry adder (RCA) in order to identify potential ways to minimize delays. The analysis's results inform the proposal of a new logic formulation and an associated RCA design for the BTA. In terms of area, delay, and energy efficiency, the comparative result reveals that the suggested RCA design outperforms the present RCA. The BTA structure is suggested using this RCA design. In comparison to the best multi-operand adder currently available, the suggested32-operand BTA reduces energy consumption by 28.7 percent and area by 22.5 percent, according to the synthesis results. Additionally, the authors have tested the suggested BTA in new multiplier designs to see how well it works. Applying the suggested BTA considerably enhanced the performance of multiplier designs, as seen by the synthesis result. For this reason, thesuggestedBTAarchitecturemaybetheway togofordesigningdigitalsystemsforsignaland image processing that are efficient in terms of space, latency, and energy consumption.

I.INTRODUCTION

A more complex signal processing system may now be integrated into a very large scale integration (VLSI) chip. Both the computational capacity and the energy consumption of these signal processing applications are high. Power consumption is becoming a key issue in thedesign of modern VLSI systems, but performance and area are still the two fundamental design objectives [1]. Two primary factors need the development of low-power VLSI systems. First, itis necessary to provide big currents and remove heat from excessive power consumption using appropriate cooling methods in order to keep up with the increasing operating frequency and processing capacity per chip.

These cond issue is the limited duration of battery life in hand held electronic gadgets. These portable gadgets have a longer runtime because to their low power design.

A key mathematical operation and digital system performance are both significantly affected by addition. The most common usage for adders is in electrical applications. Implementations of algorithms such as FFT, FIR, and IIR make use of them in digital signal processors and multipliers. Adders are involved each time the idea of multiplication is brought up. We all know that microprocessors can execute millions of instructions per second. Therefore, in designing multipliers, the speed of operation should be the primary restriction. The device's miniaturisation and power consumption should be minimal to ensure

portability. Mobile phones, laptops, and other electronic devices have larger battery backup needs. As a result, these are the three design parameters that a VLSI designer must optimise. Due to the difficulty of satisfying all of these requirements,a trade-off betweenthem maybenecessarydependingon the natureof thedemand orthe intendeduse. Although ripple carryaddersare the slowest,theyarethe mostcompact.The quickest option, carry gaze forward, uses more space, however. Carry choose adders mediate between the two extremes of adders. Wang et al. (2002) introduced a novel idea for hybrid adders—hybrid carry look-ahead/carry select adders—to expedite the adding process. In 2008, new hybrid full adders were introduced as the basis for low power multipliers.Research into the design of very large scale integration (VLSI) systems focusses heavily on creating high-speed data route logic systems that are both power- and area-efficient. Time needed to propagate acarry through an adder is the limiting factor in digital adders when it comes to the speed of addition. In an elementary adder, the total for each bit location is created sequentially, after the addition of the previous bit position and the propagation of a carry into the next position.Many computing systems use the CSLA to solve the carry propagation delay issue; it does this by creating many carries separately and then picking one to add up.

II.LITERATURE SURVEY

When working with digital signal processors, microprocessors, and computers, the arithmetic operation most often performed is addition. All other mathematical operations rely on itas afoundationalbuilding component. Thus, binary adder structures establish themselves as an essential hardware componentfor theeffective implementation of an arithmeticunit. There are a plethora of circuit designs, each with its own set of performance characteristics and practical applications, as any reader of computer mathematics will attest. There has been a plethora of study on binary adder topologies, but very little that compares and contrasts their performance.A digital circuit is the subject here. See Electronic mixer for a circuit that processes analogue signals.Digital circuits that add numbers are known as adders or summers in the electronics industry. In addition to their use in the arithmetic logic unit(s), adders find other applications in various processors, such as those that compute addresses, table indices, and so on.Most adders work with binary integers, although they can be built for other forms like excess-3 or binary- codeddecimal.Makingan adder into an adder-subtractor isapieceof cake whennegativevalues are represented by two's or ones' complement. Using a more complicated adder is necessary for other signed number formats. Here are several different types of adders:

Design of 64-bit low power parallel prefix VLSI adder for high speed arithmetic circuits by Nehru, K., A. Shanmugam, and S. Vadivel.

The addition of two binary numbers is the basic and most often used arithmetic operation on microprocessors, digital signal processors and data processing application specific integrated circuits.Parallelprefix adder is ageneraltechniqueforspeedingup binaryaddition. This method implements logic functions which determine whether groups of bits will generate or propagate a carry. The proposed 64-bit adder is designed using four different types prefix cell operators, even-dot cells, odd-dot cells, even-semi-dot cells and odd-semi-dot cells; it offers robust adder solutions typically used for low power and high-performance design application needs. The comparison can be made with various input ranges of Parallel Prefix adders in

terms power, number of transistor, number of nodes. Tanner EDA tool was used for simulating the parallel prefix adder designs in the 250nm technologies.

A comprehensive review on the VLSI design performance of different Parallel Prefix Adders by Rakesh.S,K.S.Vijula Grace,

Adders are an important part of digital systems. In VLSI digital circuits, such adders should satisfy certain design constraints like low power and high speed. In this paper we present a review of the performance of some conventional adders and parallel adders. Parallel Prefix Adders (PPA) are considered to be one of the fastest adders that had been designed and developed. Parallel Prefix Adders were established as the most efficient circuits for binary addition. These adders which are also called Carry Tree Adders were found to have better performance in VLSI designs. This paper investigates the performance of four different Parallel Prefix Adders namely Kogge Stone Adder (KSA), Brent Kung Adder (BKA), Han CarlsonAdder (HCA) and Hybrid Han Carlson Adder (HHCA). In this paper the key contribution is the information about the structure of the Parallel Prefix Adders and their performance parameters. This paper can serve as a reference to the beginners in the digital electronics and VLSI area to gain more knowledge on the Carry Tree Adders.

Low power and area efficient Wallace tree multiplier using carry select adder with binary to excess-1 converter by Kesava, R. Bala Sai, B. Lingeswara Rao, K. Bala Sindhuri, and N. Udaya Kumar

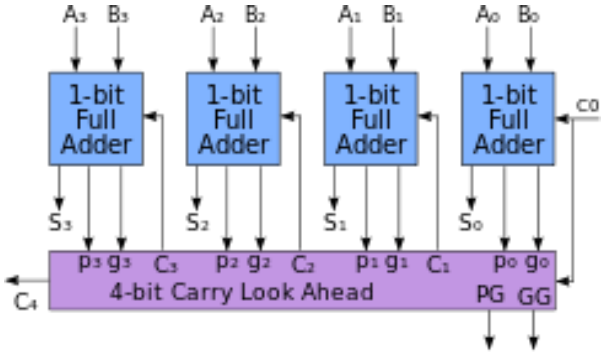
Multipliers are major blocks in the most of the digital and high performance systems such as Microprocessors, Signal processing Circuits, FIR filters etc. In the present scenario, Fast multipliers with less power consumption are leading with their performance. Wallace tree multiplier with carry select adder (CSLA) is one of the fastest multiplier but utilizes more area. To improve the performance of this multiplier, CSLA is replaced by binary excess-1 counter(BEC) which not only reduces the area at gate level but also reduces power consumption. Area and power calculations for the Wallace tree multiplier using CSLA with BEC are giving good results compared to regular Wallace tree multiplier.

III. EXISTING METHOD

In digitallogic, acarry-look ahead adder (CLA) is onekind of adder. By cutting down on the time needed to identify carry bits, a carry-look ahead adder enhances speed. The simpler, but typically slower,ripplecarryadderisagoodexampleofthistypeofalgorithm;inthisadder,the

carry bit is computed simultaneously with the sum bit, and each bit must wait for the previous carry to be computed before it can begin computing its own result and carry bits. To shorten the time it takes to compute the result of the bits with greater values, the carry-look ahead adder computes one or more carry bits prior to adding them. Examples of this sort of adder include the Kogge-Stoneadder andtheBrent-Kungadder.Like addingwithapencilandpaper,a ripple-carry adder uses a similar mechanism. We get the result by adding the two matching digits, starting at the rightmost (least significant) digit position. For similar reasons, this digit position might also be carried out (as in "9+5=4, carry 1" in the case of pencil-and-paper systems). Therefore, it is necessary to consider the potential of adding an additional 1 from a carry that has come in from the position to the right of the rightmost digit for all locations other than the rightmost digit.So, until it is known whether a

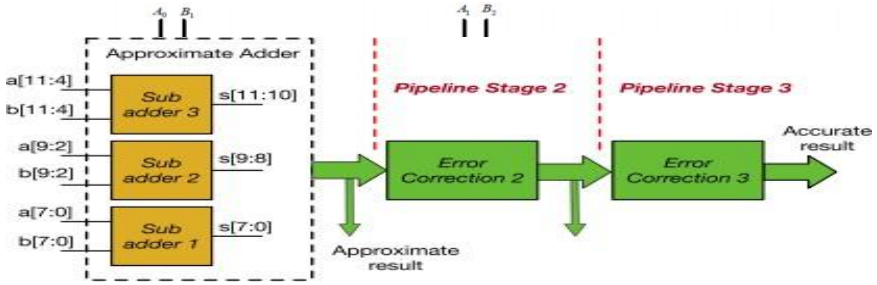
carry is coming in from the right or not, no digit location may have a final value.



When adding bits in a binary sequence, the Carry Look Ahead Logic checks to see whether the corresponding pair of bits will create or spread a carry. Because of this, the circuit may "pre-process" the two numbers being added to find the carry in advance. This eliminates the need to wait for the ripple carry effect, often known as the amount of time it takes for a carry from one Full Adder to reach another, before doing the actual addition. After making a few tweaks to the 4-bit Ripple Carry Adder we used before, we can combine it with this basic 4-bit generalised Carry Look Ahead circuit:The following is the reasoning behind the example's generate (g) and propagate (p) parameters.

The reduced number of logic gates compared to the n-bit Full Adder (FA) structure is theprimary benefit of this BEC logic.One kind of adder that falls within the conditional sum adder category is the carry choose adder. In order to function, a conditional sum adder must be satisfied. To get the total and carry, we first assume that the input carry is 1 and 0, respectively, before it arrives. A multiplexer is used to choose the real computed sum and carry values when the actual carry input comes. In a traditional carry select adder, the least significant bits (LSBs) are stored in a k/2 bit adder, while the most significant bits (MSBs) are stored in two k/bitadders. When adding numbers in a multiple-state-by-bit (MSB) adder, one adder will treat the input as a one while the other will treat it as a zero. The exact values of the output carry and total aredeterminedusingthecarryoutcomputedinthefinalstep,theleastsignificantbitstage.A multiplexer is used to accomplish the selection. This method of segmenting the adder improves area utilisation while speeding up the adding process.

Fig.8Internalstructureoftheproposedarea-efficientcarryselectadderis constructed by sharing the common Boolean logic term.



To obtain a reduced power delay product, the proposed SQRT CSLA trades off speed for transistor count. In light of this, the suggested SQRT CSLA employing CBL outperforms all

of the previously developed adders. You can see the proposed SQRT CSLA block diagram. It is well-known that sophisticated VLSI technology face the difficulty of power limits. The standard precise computing paradigm has previously been the subject of much research into low-power approaches. Approximate computing is a relatively recent trend that aims to reduce power consumption by deliberately allowing mistakes. Small mistakes here and there are really acceptable in many contexts, including video, audio, haptic processing, and machine learning. Numerous new applications and technologies make extensive use of such error-tolerant code. Mathematical circuits, the fundamental units of most computer hardware, have been the focus of most approximation computing research. Some approximation adder designs in particular have been created [1]-[14].

Fig.1.Error-correction-basedconfigurable adder

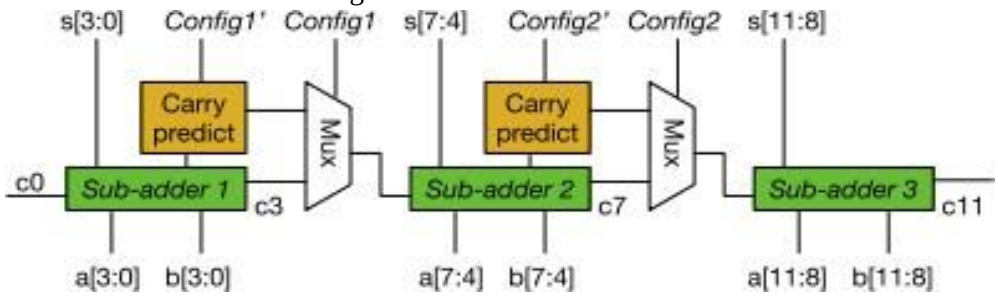


Fig.2.Carry-prediction-basedconfigurable adder

An precise adder design, created by chaining a collection of subadders, is the starting point for these schemes. An estimated carry prediction circuit is included with each subadder. It is quick, however. You may adjust the overall accuracy to various degrees by choosing between carry prediction and carry out from subadder. Error detection and repair circuitry is unnecessary for this method. There is no correlation between the arrangement of lower bits and higher bits. As a result, setups tend to converge quickly or degrade gracefully. The carry-prediction circuit is comparable to the carry lookahead component of CLA, and the sub adders in GDA [18] are CRA designs. It also has the option to tune its carry prediction to various degrees of accuracy. However, there is a significant increase in area over head due to the complex carry prediction.

3.1 SIMPLEACCURACY-RECONFIGURABLE ADDER

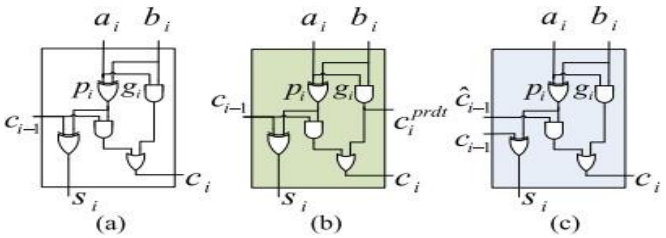


Fig.3.(a)Conventionalful ladder.(b)Carry-out selectablefull adder.(c)Carry-inconfigurable full adder.

B = (bN, bN−1,..., bi,..., b1) and A = (aN, aN−1,..., ai,..., a1) are the two addends that an N-bit adder processes. The carry-in value for bit i is ci−1 and its carry-out value is ci. The typical full adder calculates the sum si and carry ci according to the following formulas: si =

pi⊕ci−1 (1)ci=gi+pi·ci−1.The carry generate bit gi isdefinedasai·bi,thepropagatebitpiasai⊕bi, and the kill bit ki as ¯ai · b¯i. two (2) Figure 3(a) shows a schematic of a traditional complete adder at the gate level. To connect N bits of regular full adders, a CRA is used.One may get ci = gi + pi gi−1 +···+ g1 i k=2 pk + c0 i k=1 pk by recursively applying (2). (3) It follows from this equation that ci may be calculated immediately from all bits' g and p, without having to wait for thelowerbits'ctobecomputed.TheCLAadderisbasedonthisfinding. prdti=gi isthe criticalpoint.The carry-out ci+1for thelower-leftbitof thehigher-bitsubadder, biti+1,may becalculatedintwo ways:eitherby usingthestandardformulaci+1=gi+1+pi+1 ·ci,orbyusingthecarryprediction,whichis ci+1=gi+1+pi+1·cprdti=gi+1+pi+1·gi.(5)

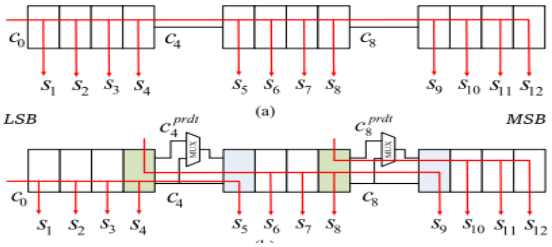
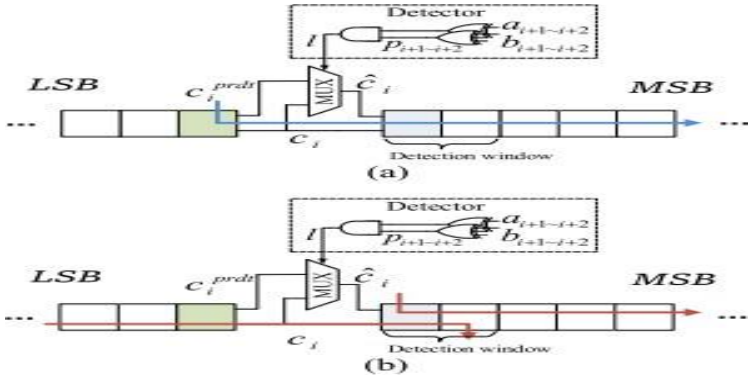


Fig.4.Implementationof12-bitadderin(a)CRA and(b)SARA

IV. PROPOSEDMETHOD

In reaction to the dwindling advantages of technological scalability, stochastic computing has started to take shape. Instead of burying differences beneath costly guard bands, designers are letting looseconventional accuracy requirements and intentionally exposing hardwarevariability to higher layers of the computing stack [1]. One potential method for reducing power, area, and latency in very large scale integration (VLSI) design is approximation computing, whichinvolves rebuilding a system's logic circuit [2]. It achieves various optimisations by taking advantage of the discrepancy between the precision needed by applications and the accuracy offered by the computer system. One of the essential parts of arithmetic circuits, adders have received a lot of interest from academics in the area of approximation computing. A surprising amount of approximate adders have been suggested in the literature, ranging from three to ten. These include segmented adders, which involve dividing an n-bit adder into k-bit subadders, carry selectadders, which usemultiple submodules, approximatefulladders, which approximate the full adder, and speculative adders, which are based on the observation that traditional adders rarely activate the critical path. Given the existing state of affairs, it is very difficult to conduct even a fair comparison of approximation adders [14], [15]. There is a commonality across the structures, despite their conceptual differences: they were all derived using an ad hoc and nonsystematic technique. Using the concept of a template, the impressive but suboptimal generic accuracyconfigurableadder(GeAr)standsout.Ofalltheapproximationaddersthatare

completely combinatorial, the lowerpartORadder(LOA)[9]hasthe best tradeoffbetween error two smaller adders. The subadder at the top is an exact adder with bits ranging from nh to 1, whereas the one at the bottom is built using nl OR gates (bits 0 to nl − 1). An additional AND gate is used to combine the adder inputs of bit position nl, i.e., anl and bnl, in order to



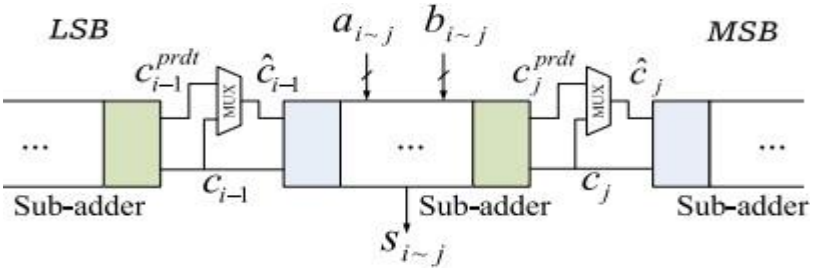
produce the carry-in signal for the precise adder. Limiting the approximation to the least significant bits limits the amount of the mistakes, which is the main benefit of LOA compared to alternative designssuchasequalsegmentationadder(ESA)[4],errortolerantadder(ETAII)[5],orvirtually accurate adder [3]. Systematically enhancing LOA is the objective of this brief. Initially, we create a generic architectural template based on the LOA design. Subsequently, we investigateallpotential implementationsofthis templateandfindtheone thatminimisesmean squarederror (MSE). Our term for it is optimised LOCA, or OLOCA. Given that LOA is the best approximation adder currently available, our optimised design beats all of the other approximate adders when weighing the relative merits of hardware cost and accuracy. The data presented in this brief from the experiments back up this claim.

Fig.5.Design ofDAR forSARA operatingin (a)approximatemodeand (b)accuratemode.

V. ARCHITECTURE

We use a three-step process to methodically develop an optimal1 approximation adder: 1) We detail the hardware cost and error metrics that measure the architecture's quality; 2) We abstract the LOA design into a more generic template; and 3) We optimise the template with respect to MSE to bring forth OLOCA. Part A. KPIs Approximate adders are not all equal; there are a variety of measures that may be used to measure the tradeoff between hardware cost and inaccuracy.Thediscrepancybetweentheadder'sestimatedandcorrectoutputvalues iscalledthe error, and its formula is ε = S˜ − S (1).

S represents the correct outcome, whereas S˜ denotes the approximative (incorrect) output of the adder. The most used metrics for determining the mistake's size are the average error (μ), standard deviation (STD or σ), mean absolute error (MAE), and minimum significant error



(MSE).Theformulafor calculating themis μ = E[ε] (2).σ=E[(ε −μ)2](3) MSE=E[ε2]=μ2+ σ 2 (4) The expectation operator, denoted as E, is used in the equation MAE = E[|v|] (5).Notably, 2n, the normalised form of the aforementioned metrics divided by the adder's range, is another

popular choice. Design area and latency must be taken into account in order to assess the architectures' hardware efficiency. The remainder of this summary uses the letters A and D to indicate the hardware area and delay, correspondingly. A basic monotonic two-input gate (AND, OR, NAND, etc.) has an area and delay cost of one, whereas a simple nonmonotonic two-input gate (XOR and XNOR) has an area and delay cost of two, according to the unit-gate model [16]. The experimental findings clearly take into account the real circuit area and delay.

4.1 GeneralTemplateArchitecture Per the terms of the LOA Among all the available approximation adders, experimental data demonstrate that LOA is the best design [14], [15], as mentioned in Section I, when the error against hardware cost tradeoff is considered. Upon close examination of LOA's architecture, it may be generalised as shown in Figure 2: A 2-to-2 logic block and nl 2-to-1 logic blocks (bits 0 to nl−1) may beused to separatethelowestsignificantsubadder. Theadder's inputs arereceived by this subsequent block at bit nl, and it uses an AND gate to provide the input carry for the precise portion; nevertheless, its total signal may be generated inaccurately. Lastly, an exact adder is the more significant subadder. Using the above generic template, LOA's architecture may be defined by inserting OR gates into the lower significant subadder bits and substitutingthe first bit of the higher significant subadder with approximation OR_AND circuitry. In theory, the blocks may be selected using any appropriately sized Boolean function. A constant function that is either one (Cte-1) or zero (Cte-0) is still a viable choice

4.2 OptimizedArchitecture

A variety ofoptimisationoutcomesarepossible,dependingon theselected errormeasures.Since the MSE is useful in data processing applications, we've chosen it as the error metric to illustrate ourpoint.Findingthebest designfrom thegeneric templaterequirescheckingeveryconceivable
permutation of the 2-to-1 and 2-to-2 logic blocks in Tables I and II. For now, let's just go with our gut and see where it takes us. The significance of mistakes in the top pieces is greater than that in the lower ones, as can be seen in (8). As a result, investing in the 2-to-2 block yieldsbetter returns than the lower 2-to-1 blocks.

V. RESULTANALYSIS

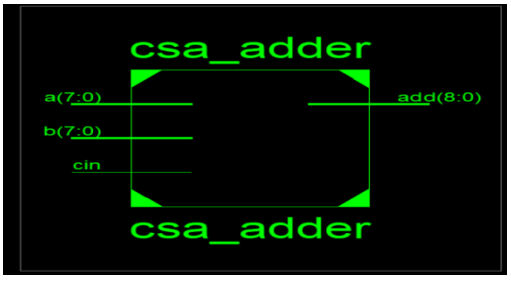


Figure:RTLSchematicDiagram

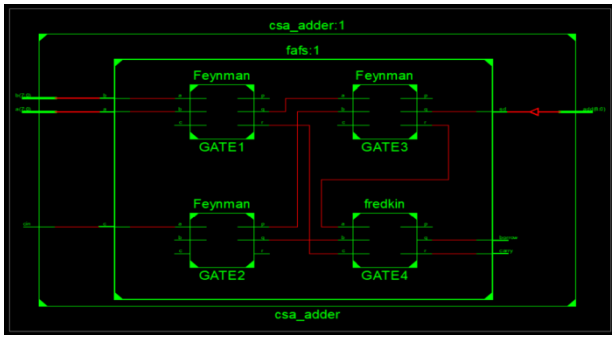


Figure:Internalblockdiagram

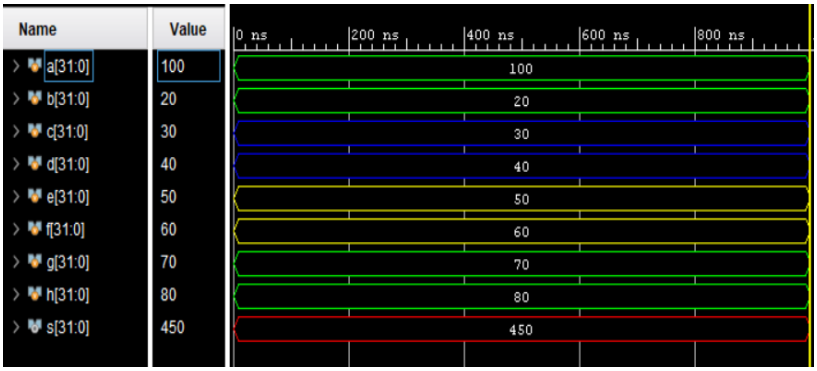


Figure:SimulationResults

Device Utilization Summary			
Slice Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	0	249,600	0%
Number of Slice LUTs	228	124,800	1%
Number used as logic	228	124,800	1%
Number using O6 output only	93		
Number using O5 output only	0		
Number using O5 and O6	135		
Number used as ROM	0		
Number used as Memory	0	48,640	0%
Number used exclusively as route-thrus	0		
Number of occupied Slices	145	31,200	1%
Number of LUT Flip Flop pairs used	228		
Number with an unused Flip Flop	228	228	100%
Number with an unused LUT	0	228	0%

Figure:AreaUtilizationSummary

VI . CONCLUSION

The authors of this work presented a low-power, high-speed variant of the current ISA architecture. In order to increase speed and decrease power consumption, this architecture has been clock gated and fine grain pipelined, respectively. In terms of highest clock frequency, experimental findings demonstrated that the proposed ISA could function at 324.57 MHz on an FPGAplatform and at 444.64MHzon a 90nm-CMOSASICplatform.Afterwards,at 400MHz, it used 9.68 mW of total power and took up 5111 μm2 of space at this technological node. Thus, compared to the state-of-the-art ISA design, the suggested ISA may run at 52% faster speed, use 52.38% less power, and take up 40.7% more space. Therefore, existing and future electronic gadgets for the Internet of Everything (IoE) and a plethora of other modern uses willundoubtedly benefit from such a design. However, by including lower-tech nodes into the design process, the area problem may be partially addressed.

VII. REFERENCES

[1] Sumeer Goel, Ashok Kumar, Magdy A. Bayoumi, Design of robust, energy-efficient full adders for deep-submicrometer design using hybrid-CMOS logic style, IEEE Trans. Very Large Scale Integr. Syst. 14 (12) 1309–1321. 2024

[2] S.Borkar,Designchallenges oftechnologyscaling,IEEEMicro19(4)23–29. July-Aug.

[3] A.P. Chandrakasan, R.W. Brodersen, "Minimizing power consumption in digital CMOS circuits" Proceedings of the IEEE 83 (4) (April 2024).

[4] KaushikRoy,SharatPrasad,"Low-PowerCMOSVLSICircuitDesign",John Wiley&Sons, INC., USA, 2024.

[5] Sundeepkumar Agarwal, P. Ramanathan, P.T. Vanathi, Comparative analysis of low power high performance flip – flops in the 0.13μm technology, in: 15thInternational Conference on Advanced Computing and Communications, ADCOM 2023, pp. 18–21.

[6] S.Monisha,M.Priya,A. UpendraRaju,V.Uma, Efficient reductionof leakage powerin low power VLSI circuits using Sleepy Keeper Approach, Int. J. Eng. Appl. Sci. 2 (Issue-1) (January 2023) 62–65.

[7] Ashesh Rastogi, Wei Chen, Alodeep Sanyal, Sandip Kundu, An efficient technique for leakage current estimation in sub 65nm scaled CMOS circuits based on loading effect, in: 20 International Conference on VLSI Design Held Jointly with 6 International Conference on Embedded Systems, VLSID'07, Bangalore, India, 2023.

[8] Sufia Banu, Shweta Gupta, The sub-threshold leakage reduction techniques in CMOS circuits, in: 2022 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, October 2020, pp. 223–228.

[9] Matthias Függer, Attila Kinali, Christoph Lenzen, Wiederhake Ben, Fast all-digital clock frequency adaptation circuit for voltage droop tolerance, IEEE Trans. Comput. Aided Des. Integrated Circ. Syst. 41 (8) (August 2022) 2518–2531.

[10] NanZhuang,Haomin Wu,AnewdesignoftheCMOSfulladder,IEEEJ.SolidState Circ. 27 (5) (May 2022).

[11] Zarin Tabassum, Meem Shahrin, Aniqa Ibnat, Tawfiq Amin, Comparative analysis and simulation of different CMOS full adders using cadence in 90 nm technology, in: 2021 International Conference for Convergence in Technology (I2CT), Pune, India, Apr 06-08, 2021.

[12] Reto Zimmermann, Wolfgang Fichtner, Low-power logic styles: CMOS versus pass-transistor logic, IEEE J. Solid State Circ. 32 (7).

[13] Fang Lu, Samueli Henry, A high-speed CMOS full-adder cell using a new circuit design technique – adaptively-biased pseudo-nmos logic, IEEE International Symposium on Circuits and Systems (May 2021) 563–565.