International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Design of an Energy-Efficient Carry Select Adder for High-Performance Digital Systems

Pilli Charles Finny [1], T. Ravikanth [2]

[1] PG Student, Dept.of ECE, Kallam Haranadhareddy Institute of Technology, Guntur, A.P, India
[2] Associate Professor, Dept.of ECE, Kallam Haranadhareddy Institute of Technology, Guntur, A.P, India

Abstract
Modern mathematical and logical units aren't worth their salt unless they can do basic algorithmic operations like multiplication and addition quickly and accurately. Digital signal processing, AI, neural networks, and ML all rely on multiplication as their primary function. Filters have new potential as data storage components in the digital age. Full adders are the most popular choice for digital filter implementations. The digital filter is implemented using PPA in this project. The sklansky adder outperforms other PPAs in terms of efficiency. Kogge Stone Adder encounters more hardware complexity during implementation. Because of this, simplifying the sklansky adder's design is essential. A comparison was undertaken between several characteristics, including latency and the number of logic gates. The suggested design above uses a number of characteristics to simplify the hardware. A new logic structure and RCA construction for the BTA are proposed based on the analysis's conclusions. The comparison shows that the proposed RCA architecture is better than the current RCA in terms of space, latency, and energy efficiency. Based on this RCA architecture, we propose the BTA structure. Based on the synthesis findings, the proposed 32-operand BTA decreases consumption of energy by 28.7 percent end size by 22.5 percent compared to the best dual-operand adder that is presently available. Along with that, the authors have evaluated the proposed BTA in novel multiplier architectures. The effectiveness of multiplier designs was significantly improved upon by implementing the proposed BTA, as shown by the synthesis outcome. This is why the proposed BTA architecture could be the best option for developing energy-, space-, and latency-efficient digital computers for processing images and signals.

Keywords: Binary tree adders, kogge stone adder, Carry Select Adder, Self Adaptive Reconfigurable Array and carry choose array adder.

1. Introduction

It is now possible to include a more intricate signal processing system onto a VLSI chip. These processing of signals applications have a great computational capability but also a significant energy consumption. Modern very large scale integration (VLSI) systems are increasingly concerned with power consumption, but performance and space remain the two primary design goals [1]. There are two main reasons why low-power VLSI systems need to be developed. To begin with, keeping up with the ever-increasing frequency of operation and computation capacity per chip requires the provision of huge currents and the removal of heat from excessive consumption of power via the use of suitable cooling technologies. The second problem is that portable electronics have a short battery life. The extended runtime of these portable devices is a result of their low energy design. The performance of digital systems and a crucial

mathematical function are both profoundly impacted by addition. When it comes to adders, electrical applications reign supreme. Algorithm implementations in digital signal processing and multipliers use them, including Fast Fourier Transform (FFT), Fast Infrared (FIR), and IIR. Whenever the concept of multiplication is broached, adders are inevitably involved. That microprocessors are capable of carrying out millions of instructions/second is common knowledge. Consequently, the essential limitation while developing multipliers should be the operating speed. To make it portable, the gadget should be small and have low power consumption. Electronic equipment such as mobile phones, computers, and others use more power while they are not in use. Accordingly, these represent the three aspects of design that a VLSI engineer has to maximise. Depending on the demand or intended usage, a compromise between all of these characteristics may be required since they are not easily satisfied. Despite being the slowest, ripple carry arithmetic are also the smallest. Carry look forward, the fastest choice, does consume more space, however. Mediating between the two ends of the adder spectrum are carry select adders. To speed up the adding process, Wang et al. (2002) proposed a new concept for hybrid adders called hybrid carrier look-ahead/carry select adders. Low power multipliers were based on new hybrid complete adders that were launched in 2008.The development of space-and power-efficient extremely fast data route logic computers is a primary goal of research into the architecture of VLSI (creative &large scale integration) systems. When it comes to digital adders, the speed of adding is limited by the time required for spreading a carry through. After adding the bits at the previous place and carrying the result to the next, a basic adder creates the total per each bit location in a sequential fashion. By generating several carries independently and then selecting one to add up, the CSLA eliminates carrying propagation delay problem, which affects a large number of computing systems.

2. Literature Survey

Design of a Compressor- and Carry-Select-Based Fast Fourier Transform Infrared Filter One of the primary concerns with digital design nowadays is space and speed. An essential need in the design of sophisticated systems and applications has always been the need to enhance the speed of performing arithmetic operations. Many processors employ the Carry Select Adder (CSA), the fastest adder, to perform quick arithmetic operations. In an effort to make adders more efficient, several architectural forms have been proposed. In the semiconductor industry, it is well-known that any processor may execute millions of work operations every second. Therefore, performance speed should be considered as one of the primary criteria while constructing multipliers. We provide a method for creating FIR filters with multipliers based on compressors and carry select adders in this work. For 16, 32, and 64 bit circuits, all adder designs are implemented with performance. The Xilinx device family is used to synthesise these structures.

A Signal Processing Application Using the Kogge Stone Adder Designing systems with minimal power consumption has emerged as a key performance objective. Digital signal processing often makes use of the efficient Finite Impulse Response Filter. The first stage of implementing a FIR filter relies heavily on adders and multipliers. The Kogge stone addition tool and booth multiplier were used in the creation of the suggested FIR filter. A high-performance circuit designer may use the Kogge stone adder, a fast adder, and a multiplication method called the Booth multiplier, which uses 2's complement notation to execute multiplication. Both the FIR filter's latency and power consumption may be decreased with the help of the suggested design. The suggested FIR filters were developed using the Xilinx ISE 12.4 tool and the verilog Hardware Description Language. Building a dadda multiplier and a parallel prefix adder into a

programmed fir filter Digital filters are very useful in many signal processing applications. What really propelled DSP to the forefront was the efficiency of its filters. Typically, one manipulates the input data by filtering in order to get a desired result. In order to alter the data that contributed to the development of various apps that benefit society, several filters are used. The multiplication block is the most important component of any filter design since it determines the filter's performance. The Dadda multiplication is one of many available multipliers; its unique selling point is the small number of gates it uses to accomplish multiplication. With this in mind, we designed and built a customisable FIR filter utilising the MAC (Multiply & Accumulate) unit [8]. Here, PPA is used to do addition and the Dadda multiplier is utilised to conduct multiplication. In conclusion, this effort contributes to the development of a highly effective FIR filter that has several potential uses in digital signal processing. Pradeep Kumar Patel, Architect Raksha Chouksey, and Dr. Minal Saxena's Design of a Fast FIR Filtering Using a Compressor and a Carry Select Adder These days, in the digital age, one of the primary design concerns is speed and space. An essential need in the design of sophisticated systems and applications has always been the need to enhance the speed of performing arithmetic operations. Many processors employ the Carry Select Adder (CSA), the fastest adder, to perform quick arithmetic operations. In an effort to make adders more efficient, several architectural forms have been proposed. In the semiconductor industry, it is well-known that any processor may execute millions of work operations every second. Therefore, performance speed should be considered as one of the primary criteria while constructing multipliers. We provide a method for creating FIR filters with multipliers based on compressors and carry select adders in this work. For 16, 32, and 64 bit circuits, all adder designs are implemented with performance. The Xilinx device family is used to synthesise these structures. Low power system design has become an important performance aim.

A. Abinaya and M. Maheswari developed the Kogge Stone amplification tool for signal processing applications. Digital signal processing often makes use of the efficient Finite Impulse Response Filter. The operation of the FIR filter relies heavily on adders and multipliers. The Kogge stone adder with booth multiplier were used in the creation of the suggested FIR filter. A high-performance circuit designer may use the Kogge stone adder, a fast adder, and a multiplication method called the Booth multiplier, which uses 2's complement notation to execute multiplication. Both the FIR filter's latency and power consumption may be decreased with the help of the suggested design. We used the Xilinx ISE 12.4 tool to synthesise the suggested FIR filters after designing them in verilog Hardware Description Language.

A programmable fir filter has been developed by S. Madhavi along with K. Rasagna, an N. Kavya, and M. Sindhu utilising the dadda multiplier and a parallel prefix. Digital filters are very useful in many signal processing applications. What really propelled DSP to the forefront was the efficiency of its filters. Typically, one manipulates the input data by filtering in order to get a desired result. In order to alter the data that contributed to the development of various apps that benefit society, several filters are used. The multiplication block is the most important component of any filter design since it determines the filter's performance. The Dadda multiplication is one of many available multipliers; its unique selling point is the small number of gates it uses to accomplish multiplication. With this in mind, we designed and built a programmed FIR filter utilising the MAC (Multiply & Accumulate) unit [8]. Here, PPA is used to do addition and the Dadda multiplier is utilised to conduct multiplication. In conclusion, this effort contributes to the development of a highly effective FIR filter that has several potential uses in digital signal processing.

3. Proposed Method

3.1 Carry-Look ahead Adder

One sort of adder in digital logic is a carry-over ahead adder (CLA). The speed is improved by a carry-look forward adder because it reduces the time required to detect carry bits. One such approach is the simpler but usually slower ripple carry adder, which computes the carry bit and the sum bit at the same time and requires each bit to wait for the preceding carry to be calculated before it can compute its own result & carry bits. The carry-look ahead addition function calculates multiple carry bits before adding them, which reduces the time it takes for calculating the result when using the bits with bigger values. This class of adders includes, among others, the Brent-Kung adder and the Kogge-Stone adder.A ripple-carry adder works in a way that is analogous to adding by hand. Starting with the most logical (least significant) digit position, we add the two matching digits to obtain the result. Just as in "9+5=4, carry 1" in a scenario of pencil-and-paper systems, this digit position could also be carried out.

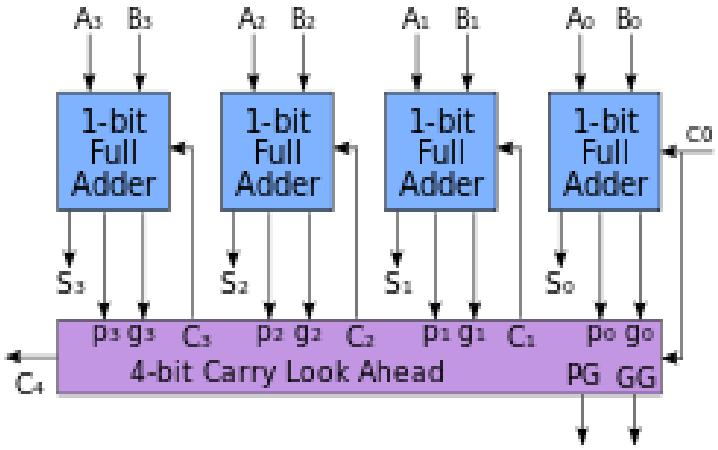

Figure 1: 4 Bit Carry Look Ahead adder

As bits are added to a binary order, the Carry Look Forward Logic determines whether the matching pair of bits will generate or disperse a carry. Consequently, the circuit could "pre-process" the two integers being added to determine the carry ahead of time. Because of this, the time it takes for carrying from one creative Full Adder to connect to another, also called the ripple carrying effect, is no longer an issue, and the addition may proceed immediately. We can integrate it with this simple 4-bit generalised carry look-ahead circuit after modifying the 4-bit ripple carrier adder we previously used. The examples' generate (g) and disseminate (p) parameters are based on the following rationale. A numerical value, ranging from 0 on the left to 3 on the right, regulates the signal generated by the system up top:

3.2 Carry Select Adder

The development of space-and power-efficient extremely fast data route logical systems is a primary goal of research into the architecture of VLSI (creative &large scale integration) systems. When applied to digital adders, the speed process addition is limited by the time required to spread through a carry through. To construct the sum for each position in a simple adder, carrying a value from one bit position is carried over for the next spot, and the process continues sequentially.By generating several carries independently and then selecting one to add up, the CSLA eliminates the propagation of carry delay problem, which affects a large number of computing systems. The CSLA is not space efficient because

the multiplexers (mux) choose the final sums and carry, but it takes into consideration carry inputs of 0 and 1 and employs a number of Ripple Carry Adder Adapters (RCA) to construct partial sums and carry.The fundamental idea of this research is to reduce space and electricity via a Basic to Excess-1 Conversion (BEC) instead of a Standard CSLA with cin=1. The principal advantage of this BEC logic over the n-bit The complete adder (FA) architecture is the decreased number of logic gates.In the class of conditional sum adders, the carry select adder is one kind of adder. A conditionally sum adder can't do its job unless certain conditions are met. Before the input carry comes, we suppose it is 1 and 0, that is, in order to determine the total and carry. When the actual extend input arrives, a multiplexer is employed to choose the calculated sum and carry values. The k/2 bit adder traditionally stores the smallest parts (LSBs), whereas the two k/bit adders traditionally store the most important bits (MSBs). When two multiple-state-by-bit (MSB) adders are used to add two integers, one of the adders will take the input as a one and the other will take it as a zero. The carry out calculated in the last step, the bit with the lowest significance stage, is used to determine the precise values of the result, which is total and carry. The selection is accomplished with the help of a multiplexer. Segmenting the adder in this way makes better use of space and speeds up adding.

3.3 Modified Regular CSLA Using BEC

Using BEC rather of the RCA and Cin=1 is the main idea of this study to reduce the space & energy use of the normal CSLA. The n-bit RCA may be replaced with an n+1 bit BEC. The functions of an a four-bit BEC are detailed in Table II, and its arrangement is shown in figure.

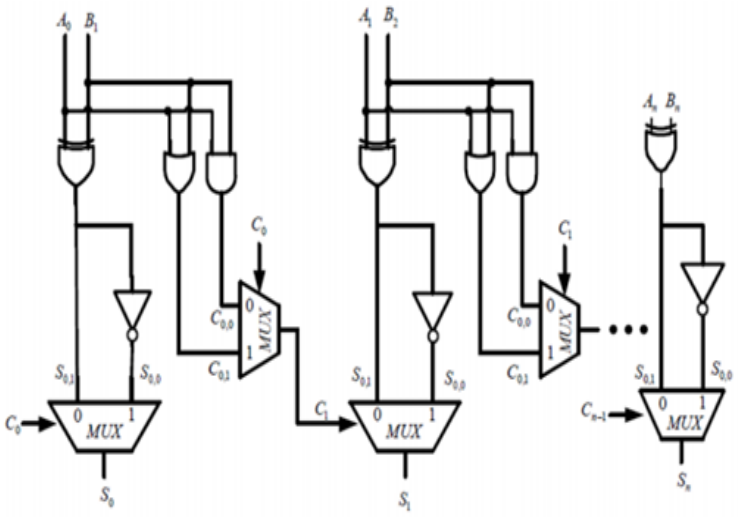

Figure 2: Internal Structure of the proposed area efficient carry select adder

As shown in Figure, the basic CSLA function may be obtained by combining 4-bit BEC with the mux. In order to operate, the 8:4 mux requires two inputs: the BEC output & (B3, B2, B1, & B0). The mux selects one of the inputs that are direct or BEC output in reaction to controlling signal Cin, resulting in the simultaneous generation of the two possible partial results. The importance of BEC logic grows as a result of the drastic reduction in semiconductor area that happens when designing CSLAs with a large number of bits. This is the 4-bit BEC's Expressions that are either true or false (keep an eye out for the useful characters NOT and XOR)

3.4 SQRT CSLA Using Common Boolean Logic

By reusing a typical Boolean Logic (CBL) term, an area economical SQRT CSLA is introduced, which eliminates the need for duplicate adder cells in the conventional CSLA. Results from analysing the truth table of a solitary-bit full adder show that as long as carrying-in a message is logic "0," the summation signal comes out as logic "1," this is the inverse message of itself. The red circles in Table II indicate this. The ubiquitous boolean logic term, the summation pair, can be produced with just one INV gate and one XOR gate. The carry pair cannot be created without also implementing an AND gate and an OR gate. Keeping the synthesis and carry circuits parallel is therefore feasible. It is well knowledge that power limitations are a challenge for advanced VLSI technology. Many studies investigating low-power alternatives to the conventional exact computing paradigm have already been conducted. Reducing power usage by intentionally tolerating errors is the goal of approximate computing, a technique that has emerged recently. Many situations, such as those involving machine learning, haptic processing, video, and audio, really tolerate occasional, minor errors. Such error-tolerant programming is heavily used by a plethora of innovative applications and technologies. Most studies on approximation computing have concentrated on mathematical circuits, the building blocks of computers. The creation of some approximations adder designs is particularly noteworthy [14].

One such method cuts power usage for an independent cosine transform (DCT) calculation by 60% without substantially affecting the produced images. Actually, different applications could have different requirements for accuracy. The precision of a mobile computer might be severely limited by its power mode, even when used for the same application. Changing the mathematical correctness during runtime using techniques like dynamic voltage as well as frequency scaling may help achieve the best accuracy-power tradeoff. Voltage overscaling brings errors (often severe ones) into the critical timing path connected to the most significant bits (MSBs), but they come near in [3] and demonstrate the benefit of improving runtime accuracy. Proper designs have been devised to reduce total error by intentionally allowing mistakes in lower bits having shorter carrying chains in addition operations.

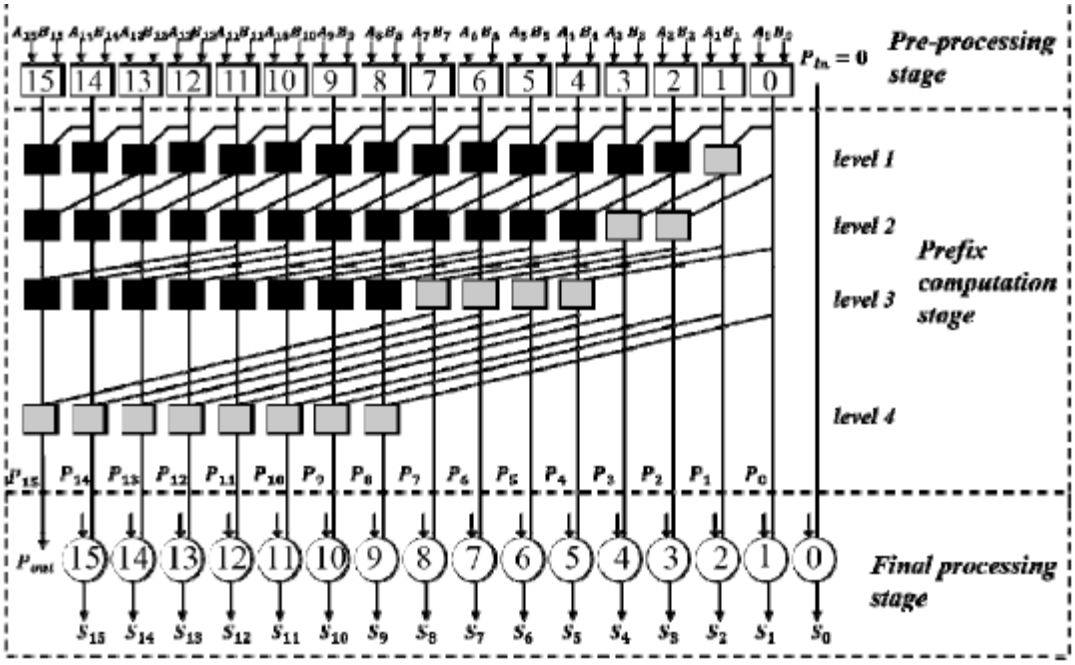

Figure 3: Proposed SQRT CSLA using Common Boolean Logic

These schemes begin with an exact adder design that is made by chaining a set of subadders. Every subadder comes with its own estimated carry prediction circuit. On the other hand, it's fast. By deciding between carry projection and carry generated by a subadder, you may modify the total accuracy to varying degrees. With this approach, error detection and correction circuitry is superfluous. The sequence of higher bits and lower bits is unrelated. Consequently, configurations often converge rapidly or deteriorate gently. The subadders described in GDA [18] resemble CRA designs, and the plus-prediction circuit is similar to CLA's carry lookahead component. Additionally, it may adjust the level of accuracy for its carry prediction. Nevertheless, the complicated carry prediction causes a significant rise in area overhead.

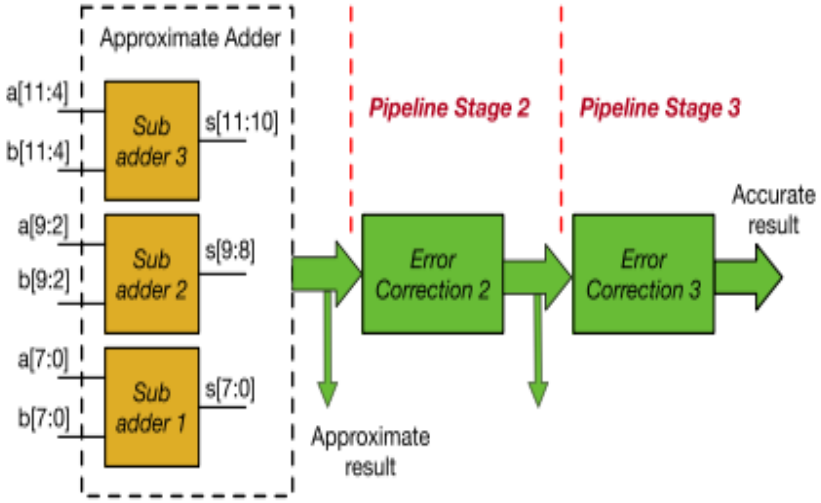

Figure 4: Error-correction-based configurable adder

4. Simple Accuracy-Reconfigurable Adder

An further improvement to this strategy is suggested, which is seen in Figure 3(c), where si+1 is reliant on ci instead of cˆi. This approach could help reduce the result's error rate when the incorrect carry is conveyed. Because the sum stays accurate and the carry does not spread when the addends are equal. Further, out of the four potential configurations for sum/carry calculation by approximate/exact carry-in, the most practical one is to allow customisation of the carry bit and calculate the sum bit by accurate carry.

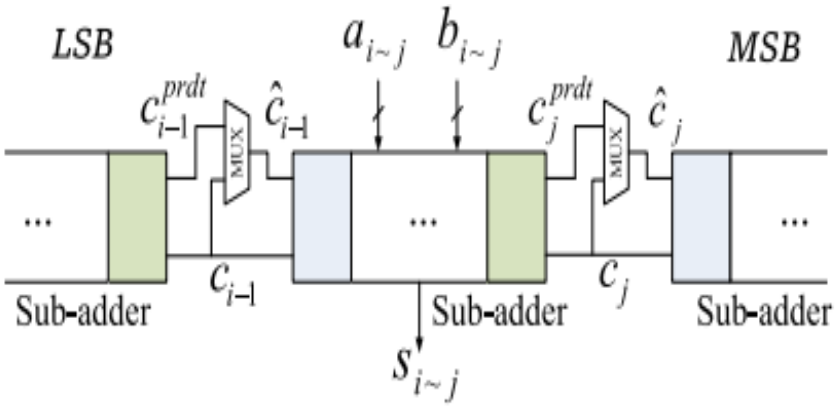

Figure 5: Design of SARA

This means that the proper carry ci ignores the choice of c prdti and immediately calculates sum si+1. Although it still restricts the critical path to be within c prdt i−1 and sj+1, calculating sj+1 from cc j in the approximate technique of SARA yields more accurate results than computing sj+1 from c¸j, as seen in Figure 4. When contrasted with sum computation in RAP-CLA and GDA, this technique improves accuracy with little additional overhead. While CRA has almost no cost for configurable adders, SARA's MUXes are the only source of extra work. The latency of sj+1 could still be minimised by approximated carry in the lower subadder, even if it is determined by accurate carry c j. Sum bit delay in a multibit adder is defined by the propagation of the carry chain of the lower bits. Even if the proper carry c j is broadcast at bit j, our SARA structure permits the potential that approximate carry in later bits may shorten the carry chain.

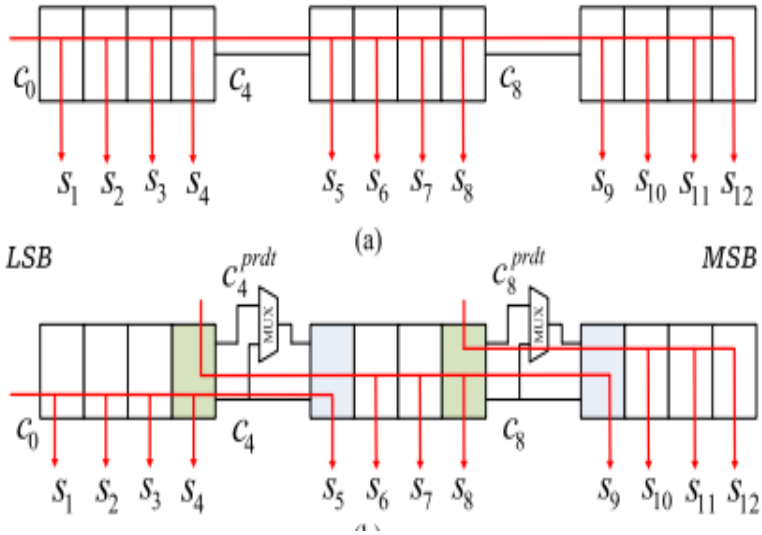

Figure 6: Six stage reconfiguration of SARA

A six-stage delay is seen for s9 in SARA, as shown in the image. In contrast to the identical bit in CRA, the three-stage decrease in being tardy of summing bit s9 in SARA is rather impressive. There is a comparable reduction in latency for bits 6–12, which constitute the remaining amount. For sums at bits 1–5, their latency is identical to CRA since they use an exact carry c0 from LSB. Because a multibit adder's maximum delay is proportional to the length of its longest critical path, 12-bit SARA achieves a reduced maximum latency. Application for SARA (C) The critical route in SARA is the same as in the CRA when all K creative subadders have cˇi set to ci, and the critical path runs along N-bit full additions. If all cˆi are selected to be c prdti, the critical path is simplified to about L-bit complete adders. Raising the supply voltage is one strategy for reducing power usage; doing so drastically reduces delay. Voltage scaling, which entails lowering the supply voltage, causes digital circuits to experience an increase in delay. As a result, reducing SARA's supply voltage may reduce its critical latencies down to the same level as CRA's under normal voltage. A decrease in the supply voltage allows for a reduction in power consumption. There may be 2K−1 different arrangements. When two configurations have the same critical pathway length, only the one with the higher level of accuracy will be considered. For this reason, viable designs include K full adders with critical channel lengths ranging from 2 bits to K·L. Delays in this kind of configurable design fluctuate in relation to the set accuracy, resulting in power savings based on variable voltage scaling.

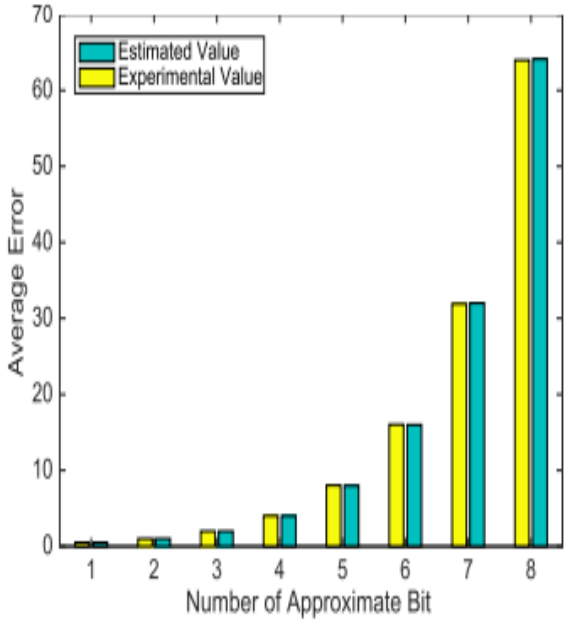

Figure 7: Average Error Vs number of Bits

5. Delay-Adaptive Reconfiguration of SARA

Almost all previous ACA research [15–19] reasonably assumes that architectural or system level applications determine the accuracy setting. We offer a specialised self-configuration approach for situations when decisions at the system or architectural level are difficult or unclear. Several previous static approximations adder solutions were beaten in simulations by SARA's self-configuration. The true worst-case route delay is determined by the addend settings, which is the essential notion underpinning self-configuration. Real route delay is only significant in the rare event of a carry that propagates over many consecutive bits. Any incorrect propagate bit created by the addend will result in a shorter carry propagate chain. Since the actual carry transmission chain is already short, there's no need to use the approximation configuration to make it shorter. When we detect a potentially long carry chain, our DAR approach simply entails turning the output of a SARA MUX to approximation mode. Compared to the continuously approximation setup, there are a number of benefits, such as avoiding specific errors for genuine small carry chains, shortening the actual extended carry chain, and accomplishing delay/power reduction. The SARA-DAR layout and detection of lengthy carry chains are shown in Figure 7(a). The actual carry chain is affected by the placement of the false propagation bit, but any false propagation bit in an identification window may switch MUX to accurate mode. Precise mode allows for a reduced carry chain by beginning the MUX detection window at bit i in the MSB versus bit i + 1. Figure 7 shows that a 2-bit detecting window (pi+1 and pi+2) may be used to find out whether a carry propagating between two subadders is happening. If this is really the case, we may adjust the MUX in order that cˆi → c prdti; otherwise, ci is the correct value. (22) The blue line in Figure 7(a) shows a successful carry chain while operating in approximation mode; this chain can only be as long as L + 1 bits. When operating in accurate mode, the MUX may restrict the lengths of the carry chain—represented by the red bars in Fig. 7(b)—to about L + 2 bits. Higher bits may repurpose propagating bits because they only utilise local main inputs, which helps keep costs down. With just one NAND gate required to set up each MUX, the detection overhead is almost nonexistent in this case. The 2-bit detecting window shown in Figure 7 may be expanded to a W-bit one. After that, when pi+j equals 1, the ERdari for bit i of MUX is equivalent to

ERprdti · W j=1. For precise mode, the size of the detection window (W) determines the tradeoff between correctness and the successful outcome carry length of the chain (L + W). In accurate mode, the critical length of the route increases as W increases, but the error rate decreases.

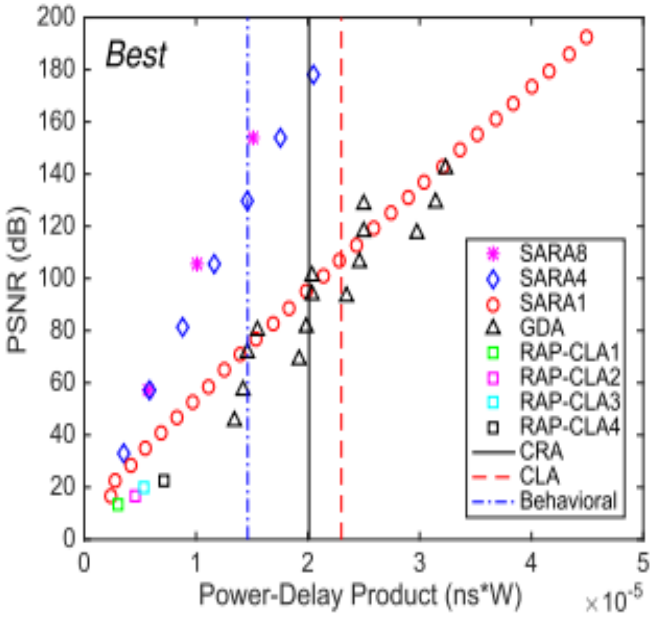

Figure 8: SARA: PSNR versus PDP

5.1 Results of Tradeoff for Different Configurations

The following are the main designs of precision adjustable adders that we compare in this area. 1) GDA [18]: Our design is in perfect harmony with [18], using four bits for every subadder. You have the option to choose either the actual or predicted carry-out depending on the subadder component when configuring this design. By modifying the quantity of lower bit addends, you may modify the accuracy of the payload prediction at each segment. The four separate designs used by RAP-CLA [19] utilise carry prediction bit-widths varying from one to four bits, as the name implies. For instance, RAP-CLA2 shows that for each with its carry predictions, it uses the two lower bits. One may choose between an exact mode and a single approximate mode for each design, as was done in [19].

Our third proposed architecture, SARA, evaluates subadders with 1, 4, and 8 bits of width, thus the designations SARA1, SARA4, & SARA8, respectively. Each dot in Figure 8 represents a different design arrangement, and together they form the main utcome. While PSNR assesses the precision of computations, PDP specifies the usual design objectives. If the picture's northwest quadrant shows a low PDP and a high PSNR, it means the design and setup are top-notch. Two well-known proper designs, CLA and CRA, have their PDPs represented by the two vertical lines, which approach infinity as their PSNR approaches. Since SARA's infinite PSNR isn't displayed as a single dot, the image cannot portray the consequence of SARA functioning in completely accurate mode. It is obvious that SARA4 & SARA8 provide the best choices. The PDP the SARA4 & SARA8, respectively, at a 100 decibel PSNR makes up nearly 50% of GDA or CRA. The results of the most current previous research, RAP-CLA, are overshadowed by SARA in the PSNR-PDP trading off. An interesting case in point is Sara1.

5.2 Tradeoff for Delay-Adaptive Reconfiguration

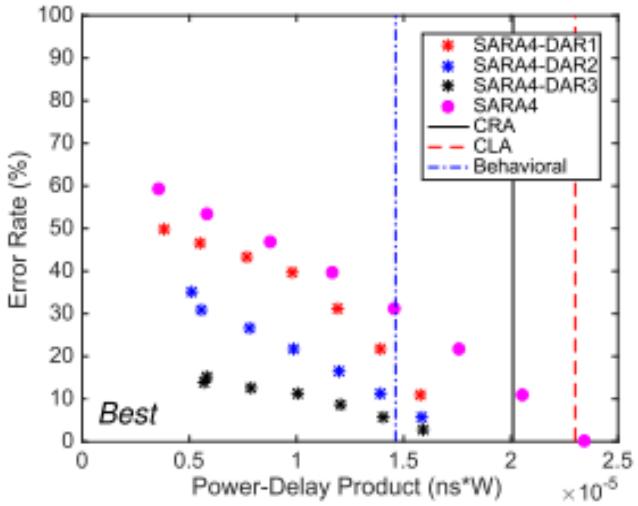

Figure 9: Error rate of SARA4-DAR with a different detection window

We may now examine the SARA-DAR design because the configuration has been chosen. Consequently, it is wise to use static approximation adders for further comparisons, since they do not require setup. Our goal in this study is to create designs for static approximation adders, such ETAII[8], FICTS[13], and AFICTS [13]. In addition, there is CRA-trunci, a simple but helpful comparison baseline that is based on estimates from CRA but does not take into account the minimum i bits contained in the supplementary files. These designs are generated using seven distinct SARA4 setups, and the average increase in PSNR of SARA4-DAR is around 3 dB. Figures 1 and 2 further show that the PDP difference among SARA4-DAR2 and SARA4-DAR1 varies with different parameters. The bulk of configurations exhibit very little change in PDP among SARA4-DAR2 and SARA4-DAR1, except for the first configuration (which is the initial dot of SARA4-DAR to the left of the photos), which exhibits a larger change. The main reason for this is the fact that different configurations have different levels of latency fluctuation.

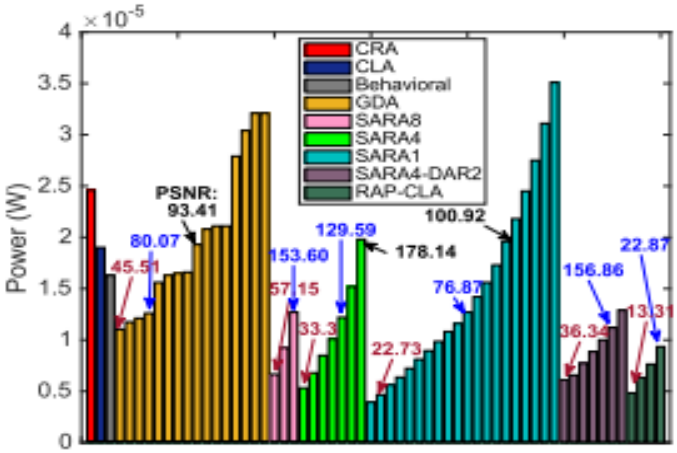

Figure 10: ISO - delay power comparison

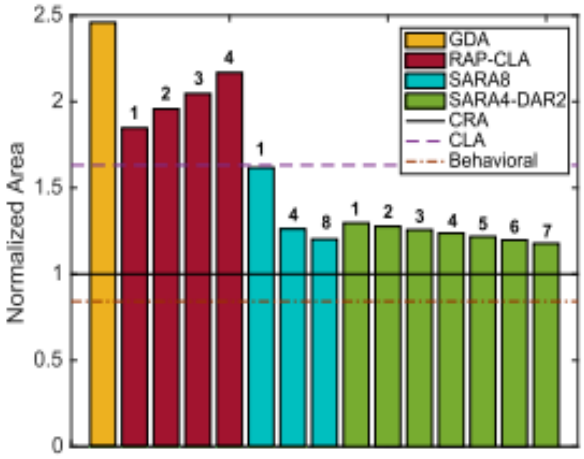

Figure 11: Area comparison

6. Result Analysis

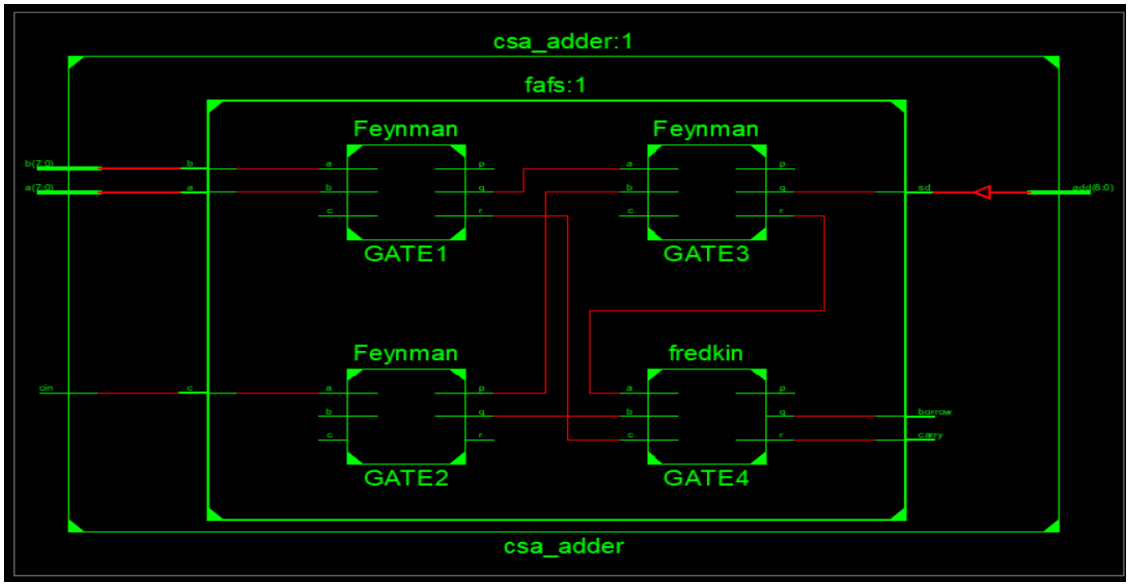

Figure 12: Internal block diagram

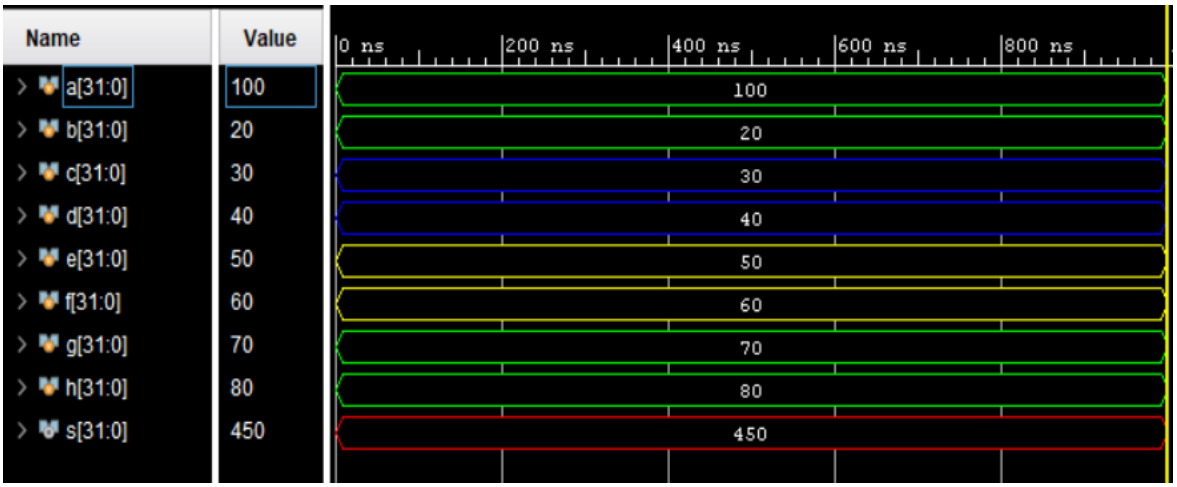

Figure 13: Simulation Results

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 14: Area Utilization Summary

7. Conclusion

This study introduced a high-speed, low-power variation of the existing ISA design. This design has been fine-grain pipelined and clock-gated to boost speed and reduce power consumption, respectively. According to the results of the experiments, the suggested ISA may operate at a maximum clock frequency of 444.64 Ghz on a 90 nm-CMOS ASIC system and 324.57 MHz for an FPGA platform. At 400 MHz, it employs 9.68 mW of overall energy and occupied 5111 μm2 of storage at this specialised node thereafter. Therefore, the proposed ISA may have a speed advantage of 52%, a power advantage of 52.38%, and a space advantage of 40.7% over the current state-of-the-art ISA design. Consequently, this design will definitely be useful for current and future electronic devices that are part of the World Wide Web of Everything, also called IoE, and for a myriad of other contemporary applications. One possible solution to the area issue is to include lower-tech node into the design process.

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, Design challenges of technology scaling, IEEE Micro 19 (4) 23–29, July–August.
3. A.P. Chandrakasan, R.W. Brodersen, "Minimizing power consumption in digital CMOS circuits," Proceedings of the IEEE 83 (4), April 2024.
4. Kaushik Roy, Sharat Prasad, Low-Power CMOS VLSI Circuit Design, 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 0.13 µm technology, in: 15th International Conference on Advanced Computing and Communications (ADCOM 2023), pp. 18–21.
6. S. Monisha, M. Priya, A. Upendra Raju, V. Uma, Efficient reduction of leakage power in low power VLSI circuits using Sleepy Keeper Approach, Int. J. Eng. Appl. Sci., 2(1), January 2023, pp. 62–65.
7. Ashesh Rastogi, Wei Chen, Alodeep Sanyal, Sandip Kundu, An efficient technique for leakage current estimation in sub-65 nm scaled CMOS circuits based on loading effect, in: 20th International Conference on VLSI Design Held Jointly with 6th 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. Integr. Circ. Syst., 41(8), August 2022, pp. 2518–2531.
10. Nan Zhuang, Haomin Wu, A new design of the CMOS full adder, IEEE J. Solid-State Circuits, 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. 6–8, 2021.
12. Reto Zimmermann, Wolfgang Fichtner, Low-power logic styles: CMOS versus pass-transistor logic, IEEE J. Solid-State Circuits, 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, in: IEEE International Symposium on Circuits and Systems, May 2021, pp. 563–565.
14. K.T. Lau, Digital IC functions with differential Cascode voltage switch circuits, IEEE Trans. Consum. Electron., 34.
15. Fang-shi Lai, Wei Hwang, Design and implementation of differential Cascode voltage switch with pass-gate (DCVSPG) logic for high-performance digital systems, IEEE J. Solid-State Circuits, 32(4), April 2020, pp. 563–573.
16. Lixin Gao, High performance complementary pass transistor logic full adder, in: 2021 International Conference on Electronic & Mechanical Engineering and Information Technology, August 2021, pp. 4306–4309.
17. M. Suzuki, N. Ohkubo, T. Shinbo, T. Yamanaka, A. Shimizu, K. Sasaki, Y. Nakagome, A 1.5-ns 32-bit CMOS ALU in double pass-transistor logic, IEEE J. Solid-State Circuits, 28(11), November 2020, pp. 1145–1151.
18. A. Parameswar, H. Hara, T. Sakurai, A swing-restored pass-transistor logic-based multiply and accumulate circuit for multimedia applications, IEEE J. Solid-State Circuits, 31(6), June 2019, pp. 804–809.
19. T. Bhagyalaxmi, S. Rajendar, S. Srinivas, Power-aware alternative adder cell structure using swing-restored complementary pass transistor logic at 45 nm technology, in: International Conference on Nanomaterials and Technologies, vol. 10, Procedia Materials Science, 2019, pp. 789–792.
20. Pankaj Kumar, Rajender Kumar Sharma, Low voltage high performance hybrid full adder, Engineering Science and Technology, an International Journal, 19(1), March 2019, pp. 559–565.
21. Minkyu Song, Geunsoon Kang, Seongwon Kim, Euro Joe, Bongsoon Kang, Design of a low power 7-bit serial counter with Energy Economized Pass-Transistor Logic (EEPL), in: Proceedings of the Third International Conference on Electronics, Circuits and Systems (ICECS '96), 2018.
22. Yingtao Jiang, Abdulkarim Al-Sheraidah, Yuke Wang, Edwin Sha, Jin-Gyun Chung, A novel multiplexer-based low-power full adder, IEEE Trans. Circuits Syst. II: Express Briefs, 51(7), July 2018, pp. 345–348.

23. Hamed Naseri, Somayeh Timarchi, Low-power and fast full adder by exploring new XOR and XNOR gates, IEEE Trans. Very Large Scale Integr. Syst., 26(8), August 2018, pp. 1481–1493.
24. Mehedi Hasan, Md Shahbaz Hussain, Mainul Hossain, Mohd Hasan, Hasan U. Zaman, Sharnali Islam, A high-speed and scalable XOR-XNOR-based hybrid full adder design, Comput. Electr. Eng., 93, July 2018.
25. Jyoti Kandpal, Abhishek Tomar, Mayur Agarwal, Design and implementation of 20-T hybrid full adder for high-performance arithmetic applications, Microelectron. J., 115, September 2021.
26. S. Sharmila Devi, V. Bhanumathi, Design of reversible logic based full adder in current-mode logic circuits, Microprocess. Microsyst., 76, July 2020.
27. Azeem Mohammed Abdul, N.V.S.S. Prasanna, K.V. Harish Kumar, K. Banu Teja, 14-T hybrid full adder design and its implementation for high-performance arithmetic applications, Mater. Today: Proc., 69(2), 2022, pp. 454–458.
28. C. Kalamani, V.S. Nishok, A. Asha, S. Saravanakumar, Design of adiabatic circuits with reversible logic based full adder and multiplier in current-mode logic circuits for efficient power dissipation, Optik, December 2022, Article 170438. https://doi.org/10.1016/j.ijleo.2022.170438.
29. Parisa Rahimi, Myasartabany, Seyedalipourmoafi, "A novel low power and high speed 9-transistors dynamic full-adder cell simulation and design," in: 2023 IEEE Symposium on Computers and Communications (ISCC), 2023, pp. 1287–1292.
30. M. Aguirre, M. Linares, An alternative logic approach to implement high-speed low-power full adder cells, in: Proc. SBCCI, Florianopolis, Brazil, Sep. 2005, pp. 166–171.
31. Kala Bharathan, Seshasayanan Ramachandran, An optimized embedded adder for digital signal processing applications, Turk. J. Electr. Eng. Comput. Sci., 24(6), 2016, pp. 5224–5237.