ISSN 1840-4855
e-ISSN 2233-0046

Original scientific article
http://dx.doi.org/10.70102/afts.2024.1631.220

OPTIMIZING VLSI ARCHITECTURE WITH CARRY LOOK AHEAD TECHNOLOGY BASED HIGH-SPEED, INEXACT SPECULATIVE ADDER

S. Pragadeswaran[1*], M. Vasanthi[2], E. Veera Boopathy[3], S. Suthakaran[4], S. Madhumitha[5], N. Ragupathi[6*], R. Nikesh[7], R. Preethika Devi[8]

[1]Department of Electronics and Communication Engineering, Karpagam Institute of Technology, Coimbatore, India. e-mail: praga1994@gmail.com,*
orcid: https://orcid.org/0009-0009-0987-3039
[2]Department of Electronics and Communication Engineering, Knowledge Institute of Technology, Salem, India. e-mail: mvece@kiot.ac.in,
orcid: https://orcid.org/0009-0003-4198-7833
[3]Department of Electronics and Communication Engineering, Karpagam Institute of Technology, Coimbatore, India. e-mail: boopathy.veera@gmail.com,
orcid: https://orcid.org/0000-0001-8413-840X
[4]Department of Electronics and Communication Engineering, Karpagam Institute of Technology, Coimbatore, India. e-mail: suthakaran138@gmail.com,
orcid: https://orcid.org/0000-0001-8729-6825
[5]Department of Electronics and Communication Engineering, Karpagam Institute of Technology, Coimbatore, India. e-mail: madhumitha.ece@karpagamtech.ac.in,
orcid: https://orcid.org/0009-0006-1505-8418
[6]Department of Electronics and Communication Engineering, Karpagam Institute of Technology, Coimbatore, India. e-mail: ragusty171@gmail.com,*
orcid: https://orcid.org/0009-0001-7004-3175
[7]Department of Electronics and Communication Engineering, Karpagam Institute of Technology, Coimbatore, India. e-mail: nikenikesh142@gmail.com,
orcid: https://orcid.org/0009-0006-8799-5638
[8]Department of Electronics and Communication Engineering, Karpagam Institute of Technology, Coimbatore, India. e-mail: user.preethi@gmail.com,
orcid: https://orcid.org/0009-0009-3067-8078

SUMMARY

This paper describes the design of the Carry Look-ahead Adder (CLA), which uses the Inexact Speculative Adder (ISA). The speculative adder is designed for high-speed VLSI architecture and features advanced compensation techniques and optimized hardware efficiency. Considered to be the adder's critical path, it is finely pipelined to contain a few logic gates along its carry propagation chain. This increases the frequency of operation by employing CLA, which is pipelined with some logic gates. To lower the model's power consumption, a separate planned ISA stage has been clocked gated. The Field Programmable Gate Array (FPGA) framework is used for hardware implementation and punctuality verification. The carry look-ahead adder operates at the clock frequency of 324 MHz. A power and area study of a 32-bit planned ISA is performed using CMOS technology. Our device's power consumption and chip area consumption are lower than those of a conventional speculative adder, at 2 mm and 9.68 mW, respectively.

Key words: *inexact speculative adder, FPGA, ISA, CLA.*

Received: August 09, 2024; Revised: October 14, 2024; Accepted: November 04, 2024; Published: December 24, 2024

INTRODUCTION

These days, high-speed adders are quite desirable, but silicon area and power (or energy) are just as important. A lot of research has been done recently on the use of spectrum sensors in intelligent cognitive-radio contexts [1, 16], and Internet of Everything (IoE) [3] devices with physical interfaces. The primary focus of hardware for these algorithms' is sensing and actuating, and for real-time interfaces, response time is a crucial factor that must be addressed. As a result, our efforts are concentrated on building highly efficient adders in terms of speed, which is crucial in the current day [13-15]. An approximation and incorrect circuit technique can be used to accomplish high-speed, low-power, and area-efficient design with an appropriate loss of accuracy and performance [12][17]. One way to boost power and speed in such circuits is to trade off accuracy with speculation.

More sophisticated signal processing systems are implemented on a VLSI device as integration scope increases [2]. These signal-processing apps just don't need a lot of computational power, but they do consume a lot of energy [11]. While execution and region continue to be the two most significant planning commitments, energy consumption is becoming a crucial consideration in the design of modern VLSI systems [5]. There are two primary reasons why a low-power VLSI system is required. First, large flows need to be conveyed and the intensity caused by high power consumption needs to be reduced by suitable cooling techniques as the working recurrence and handling limit per chip continue to grow steadily. Second, the battery life of multipurpose electronic devices is limited. These handy devices' low power setup just causes a delayed active time. Second, there are only so many portable electronic gadgets available. These mobile devices' working times are directly extended by their low-power design. Two main forces drive the need for low-power VLSI frameworks.

Addition typically has an impact on a crucial arithmetic function as well as the digital system's overall performance. The most widespread use for adders is in electronic applications. Adders are used in conjunction with multipliers and DSP to perform a variety of algorithms, including FFT, FIR, IIR, and others [6]. Adders appear whenever the concept of multiplication is discussed. As we all know microprocessors can carry out millions of instructions in a second and speed is the most critical constraint to consider when developing multipliers. Devices should be highly portable, which means that their power consumption and size should be large [8]. Devices like laptops and mobile phones require additional battery backup.

In very large-scale integration, low power consumption and fast speed are the primary key points [7]. To maximize accuracy while reducing power consumption and increasing speed, we must take into account a much-improved adder. We may increase the pace while maintaining an accurate and appropriate degradation by employing the speculative technique. To label these adders as imprecise speculative adders. Many adders are recommended by the resources. Since accuracy is the primary factor in this case, we may use a speculative adder that produces more pronounced output. Additionally, it increases operating speed while consuming the least amount of electricity.

Designing steps are listed below:

- Manipulating the CLA by using a speculative adder.
- Then CLA is pipelined to minimize the delay in the critical path. Eight-, sixteen-, and thirty-two-bit versions of projected and advised architectures are executed in FPGA.
- Pipelined phases of the ISA architecture will be activated through clock-gated signals.

EXISTING ADDER DESIGNS

Half Adder

A half adder is a type of digital circuit with combination functions that produce the sum (S) bit and carry (C) bit after adding two single-bit binary values. If A and B are the main input bits, the sum bit (S) is obtained by X-ORing A and B, whereas the carry bit (C) is obtained by ANDing A and B. The simplest adder circuit is the half adder, which is used in a variety of applications such as designing complete adders and calculators, ALUs in various processors, calculating addresses, and so on [19]. When input A is at low logic, the total output is a duplicate of input B, however, when input A is at high logic, it is complemented by input B. When control input A is high, the carry output is the same as input B; when low, it is logic '0'.

Full Adder

Full adders, being one of the most fundamental building blocks for circuit applications, have remained a primary research focus over the years [9]. Different logic models, each with its own set of advantages and disadvantages, were researched to develop a more reliable, simpler, and low-power implementation, although the on-chip area needed is often greater than that of its dynamic equivalent [10]. These designs take advantage of the characteristics of several logic types to increase the overall performance of the entire adder.

Ripple Carry Adder

The number of full adders in the Ripple Carry Adder is determined by the number of bits. The initial bits of the two numbers (A(0) and B(0)) and the input offset (C_{in}) are supplied to the first complete adder. The first full adder yields the sum and the initial carry piece, which is then spiraled into the subsequent full adder and so forth. Thus, the name Ripple Carry Adder for the viper. The circuit delay is substantial despite the RCA region's lower usage [16]. Compared to the other combiners described, RCA has a mix of low area usage, high delay time, and greater current consumption. Sum (S) and Carry (C) are the outcomes of the addition operation that we have to execute with the operands A and B. Assuming that the input carrying capacity (C_{in}) is zero, the first full adder receives the very initial bits of the operators A(0) and B(0). The outputs are the output carrying capacity (C_{out}), which ripples into the viper to return to the prior full value, and the sum (initial bit of S(0)). Consequently, the second full adder is allotted the operands' second bits, the third complete adder gets the third bits, and the fourth adder is given the fourth bits. Each full adder produces a total that includes the sum bit and carry bit from the previous full adder; it is transferred to the following complete adder as an input. We get the outcome of the ongoing procedure from the ultimate complete adder.

Carry Look Ahead Adder

By recalculating carries at intermediate stages utilizing carry create and carry propagate, independent of the input carry, the Carry Look Ahead adder expedites the acquisition of the result. It is known as the Carry Look Ahead adder as a result. The two new parts of this adder are carry propagate and carry generate. While carry create generates carry independent of the input carry provided to the first step, carry propagates to succeeding stages. The operand bits A and B, along with the previously mentioned formulae, are used to compute the propagate and generate terms. Then, utilizing the aforementioned equations, the carry and sum bits are created. The generate and sum bits are generated based on the previously built generate and propagate bits.

Carry Skip Adder

This utilizes skip logic to propagate the carry. The main concept employed is that carry is transmitted across the bit position without being altered for varying values of A_1 and B_1. In this situation, we make use of two RCAs for each block, and we can incorporate additional RCAs if necessary. Its logic is implemented from the second block and comprises of AND gates that generate a carry. Here, the operands are A and B, both of which consist of four bits. The operands' first two bits are given to the first RCA, and the next two bits are given to the second RCA. The first RCA is assigned the initial carry C_{in} of '0'. The initial RCA generates the first pair of bits for the total before transferring the carry to the subsequent RCA, which generates the remaining bit pairs. The circuit uses an OR gate in addition to AND logic for its total output carry [18].

Carry Save Adder

An adder called Carry Save is not like the ones that came before it. Instead of passing on intermediate carries to subsequent stages, it keeps the carry and uses an extra full adder to include it in the next stage's sum. The technique of summing binary numbers, usually three sets of four bits each, diminishes the circuit's time delay. The initial step in the addition process is to store the carry bits and add up the pieces before moving on to the next phase. Stage 2 combines the stored carry and sum bits, working similarly to RCA. Here, three things are used: Z, B, and A. Z denotes a 4-bit carry-in. Four bits of A, B, and Z are represented by four complete adders. Every complete adder computes its carry bits and sum. The carry bits are kept and added to the following sum term via a ripple carry adder rather than being sent on to the next full adder.

Kogge Stone Adder

A high-speed parallel variation of the Carry Look Ahead Adder is the Kogge Stone Adder. Stone Adder Kogge. Of all the adders that have been discussed thus far, the Kogge stone adder is the fastest. This adder is extensively used in circuits or businesses that require high-speed operations since it eliminates the time delay in the circuit to create the carry signals. Three phases could be separated the circuit. Stage 1 uses four full adders with input carry 0 to generate P and G. Using the previously stated equations, P_1 and G_1 are developed in the second step, and P_2 and G_2 are generated in the third stage. Propagate and produce are words that are learned to forecast future loads. The terms propagate and generate are acquired to predict future loads. Terms are produced based on the outcomes of the initial phase, creating sum bits.

Carry Select Adder (CSLA)

For each of the two possible inputs of C_{in}, a carry-select adder returns either 0 or 1. The carry and output sum are predetermined by employing two RCAs, and the multiplexer selects between a carry input of 0 and 1. First, four full adders with an input carry of 0 receive the operands X and Y. Next, four more full adders with an input carry of 1 receive the same operands X and Y. That is, we use input carry values of 0 and 1 in advance to construct the sum and output carry bits. Four complete adders each make up an RCA. A 2x1 MUX receives the output sum bits for input carry equal to either 0 or 1 together with user input. The overall sum is calculated by adding the outputs of all four MUXs utilized for sum bits. Two RCAs' worth of output carry are sent to a second 2x1 MUX, where the user's input carry serves as the selection line.

PROPOSED INEXACT SPECULATIVE ARCHITECTURE

About Speculator and Adder: $A=A_0$, A_1, A_2... are the two n-bit operands for addition. $B=A_{n-1}$, $A_n=B_0$, B_1, and B_2...$B_{n-1.}$ $S=S_0$, S_1, S_2, C_{in}, C_{out}... The symbols S_{n-1} represent the carry-in, carry-out, and sum in that order. The input carry for every subsequent adder block is the output carry of each speculator, as seen in Figure 2 [4]. The adder block uses the following formula to determine the total of the two operands and the equations (1-4) is given below.

$$S_i = P_i \oplus C_i \qquad (1)$$

The speculator block evaluates the carry based on the equations given below.

$$G_i = A_i \,.\, B_i \qquad (2)$$

$$P_i = A_i \oplus B_i \qquad (3)$$

$$C_{i+1} = P_i.C_i + G_i \qquad (4)$$

Where P_i represents the carry propagate and G_i represents the carry generate.

During ISA architecture design, carry for each block of adder is speculated. But in the conventional design, speculator, compensator, and adder blocks create delays such as $\partial 4-$, $\partial 4+$, and ∂ severally. Detection of speculation errors can be detected by the inspection of the previous carry. Later on, correction and reconciliation can be performed by the compensator block. Conventional ISA and speculator blocks create h delay. Logical blocks produce (+1) h delay.

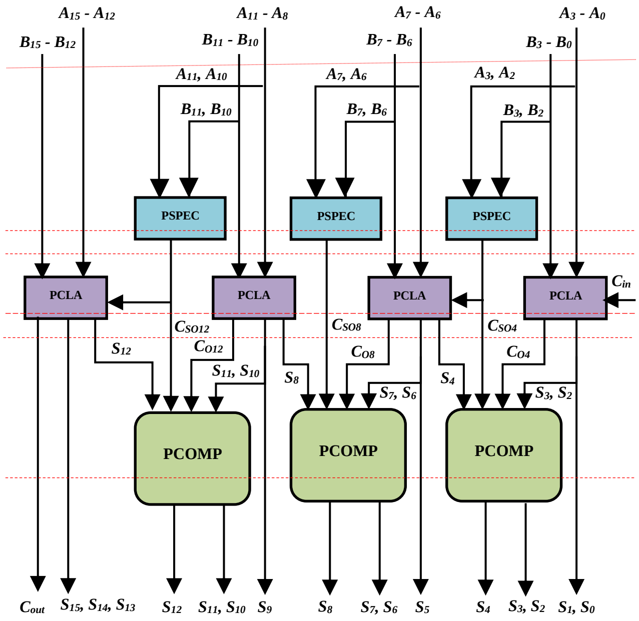

Figure 1. In-exact speculative adder architecture

The proper analysis of the pipelining blocks helps us to minimize the critical path delay and boost the speed shows in Figure 1.

Here x=4, n=16 bit is used for the explanation of the pipelining technique. Raising the ‘n' value, which does not produce any causes in the critical path, the Adder, speculator, and compensator keep stable when the ‘x' value remains uninterrupted shows in Figure 3 and 4. Pipelined speculator, pipelined compensator, and pipelined carry look-ahead adders are outlined by five stages of pipelining and six extent registers.

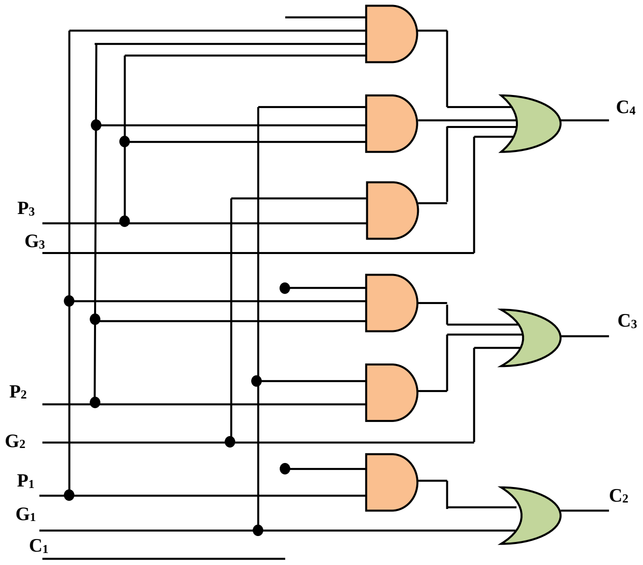

Figure 2. Logic diagram of Carry Look-Ahead Adder

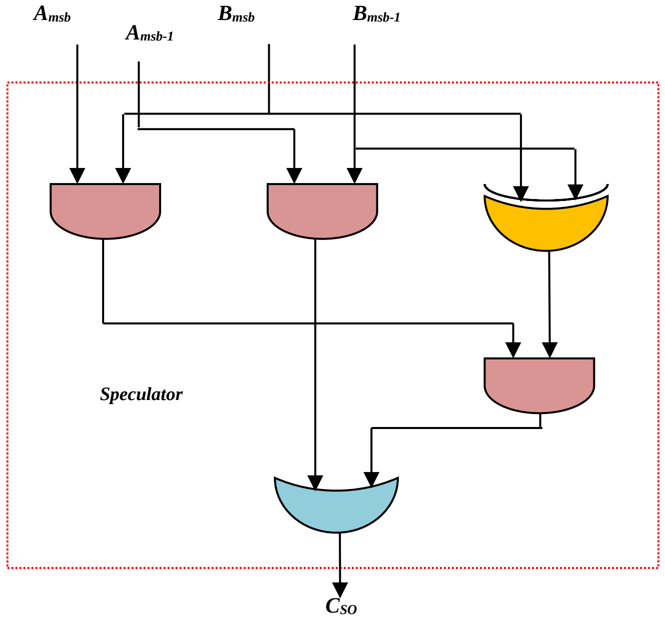

Figure 3. Logic diagram of speculative adder

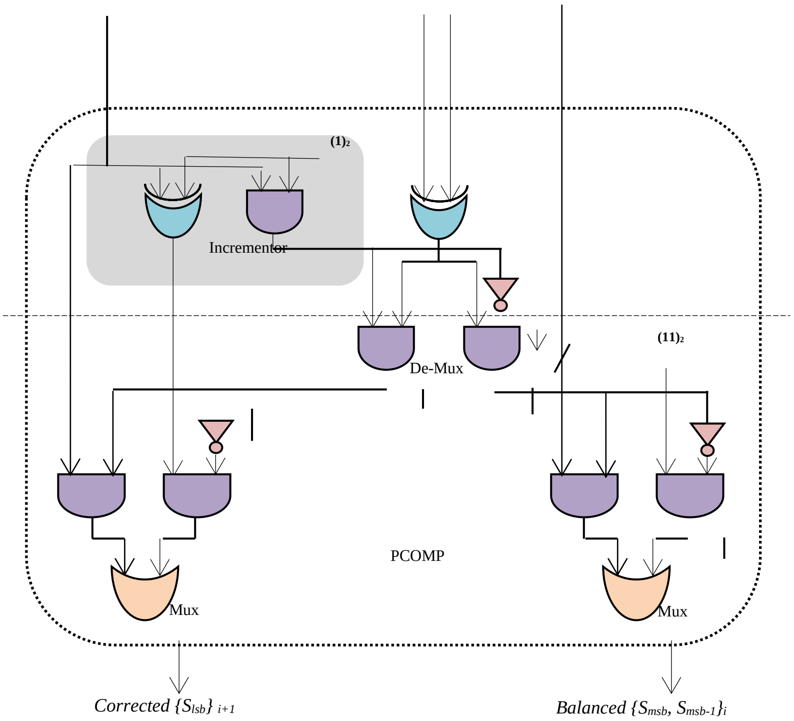

Figure 4. Logic diagram of compensator

SIMULATION OF PROPOSED ARCHITECTURE

The propagation delay of t_{phl}, t_{plh} and t_p in a circuit appears as input and output from a simulation window. The further consumption appears on the right base bit of the window. If the semiconductor unit sizes are modified. Generate the layout once more and run the simulations until you succeed in your target delays. Betting on the input sequence assigned at the input, the output is ascertained within the simulation; the value of power is already given.

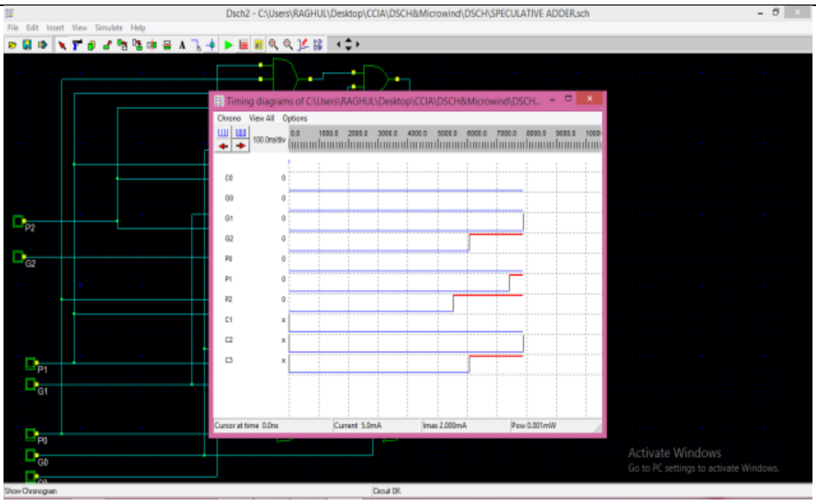
Dsch2 - C:\Users\RAGHUL\Desktop\CCIA\DSCH&Microwind\DSCH\SPECULATIVE ADDER.sch
File Edit Insert View Simulate Help
Timing diagrams of C:\Users\RAGHUL\Desktop\CCIA\DSCH&Microwind\DSCH...
Chrono View All Options
C0
G0
G1
G2
P0
P1
P2
C1
C2
C3
Cursor at time 0.0ns
Current 5.0mA
Imax 2.000mA
Pow 0.001mW
P2
G2
P1
G1
P0
G0
Show Chronogram
Circuit OK
Activate Windows
Go to PC settings to activate Windows.

Figure 5. Timing diagram of proposed gate-level speculative adder design

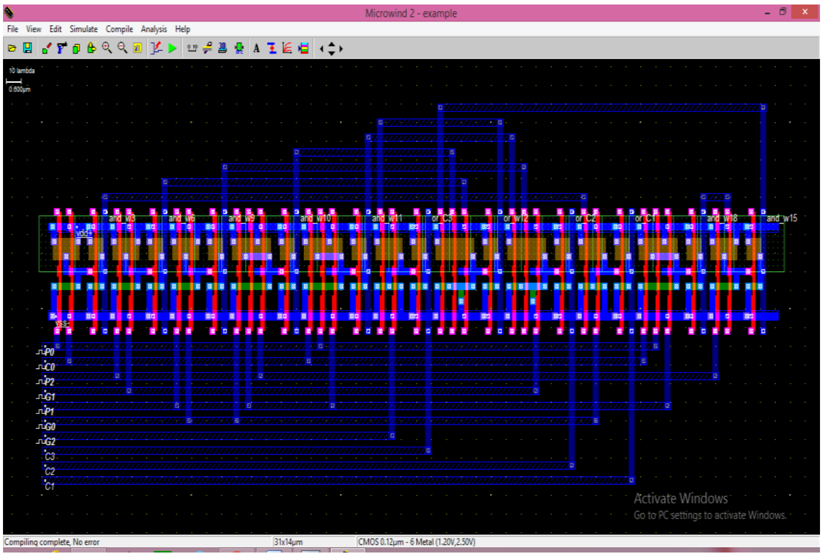

Figure 6. Layout structure of proposed gate-level speculative adder design

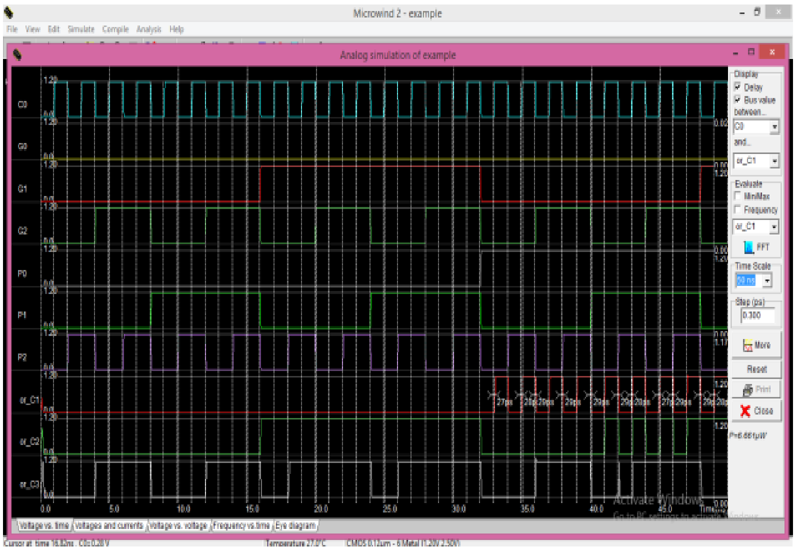

Figure 7. Characteristic curves of proposed gate-level speculative adder design

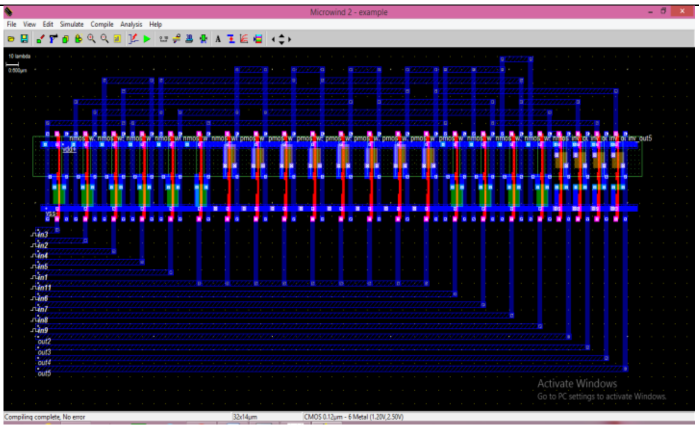
Microwind 2 - example
File View Edit Simulate Compile Analysis Help
in3
in2
in4
in5
in1
in11
in6
in7
in8
in9
out2
out3
out4
out5
Compiling complete, No error
CMOS 0.12µm - 6 Metal (1.20V,2.50V)

Figure 8. Layout structure of proposed transistor-level speculative adder design

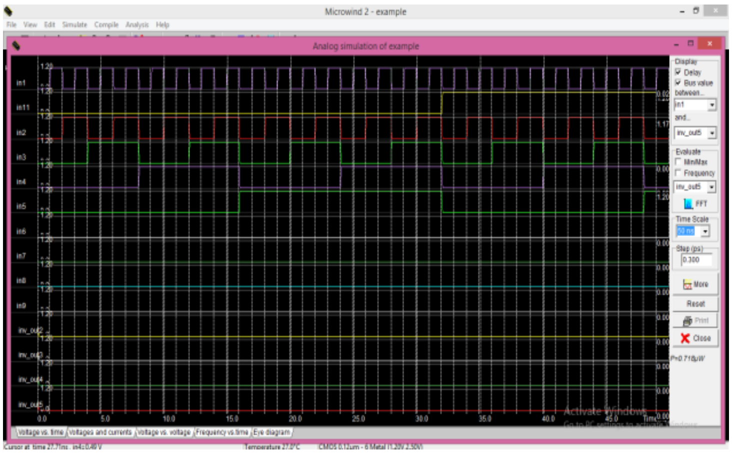

Figure 9. Characteristics curves of proposed transistor-level speculative adder design

The simulation results of the timing diagram, layout structure, and characteristic curves based on the proposed gate-level speculative adder design are shown in Figures 5, 6, and 7, respectively. The simulation results of layout structure and characteristic curves based on the proposed transistor-level speculative adder design are illustrated in Figures 8 and 9, respectively.

RESULTS AND DISCUSSIONS

The important board-level execution and verification of pipelined and non-pipelined ISA are presented in this section. Square computed the findings behind the post-reproduction comparison. This production in the course of the layout and prescribed ISA adder foundation antiquated coded in HDL. So remotely encouraged as incorporated in ISE dash batch. In this design, we incorporated 3 patterns, such as 8, 16, and 32 bits, respectively. The provoke netlists the square-3E interpretation of the Xilinx FPGA board at the time of the winning syntax check and integration of the look adder. After that mortal order data and therefore the individual adders can be enforced and determined by the diversity of devices.

Table 1. Comparison of results of power and ratio in FPGA implementation

IAS Configurations	8-bit NPLA	8-bit PLA	16-bit NPLA	16-bit PLA	32-bit NPLA	32-bit PLA
FPGA family	Spartan -3E	Sparta n-3E	Spartan -3E	Spartan -3E	Spartan -3E	Spartan -3E
FPGA device	Xc- 7a100tc sg 324-1L -1L	Xc- 7a100t csg324	Xc- 7a100tc sg 324-1L	Xc- 7a100tc sg 324-1L	Xc- 7a100tc sg 324-1L	Xc- 7a100tc sg 324-1L
4-bit LUT's	13	31	22	72	46	149
Critical path delay (ns)	11.292	4.7368	13.672	4.7368	13.762	4.7368
Max. Clock frequency (MHz)	88.592	139.44	73.141	139.44	72.66	139.44
Power consumed (W)	5.533	3.96	10.596	7.152	21.865	14.1072

Table 1, the utmost practical oscillation is 139.44 MHz. This value is 57.4%, 90.6%, and 92.1% greater than the clock oscillations accompanying 8, 16, and 32-bit non-pipelined adders. The connection of certain space utilized as far as LUTs and force is additionally required for such adder's square amplitude possible by assembling still as brithing out FPGA for the entire of such adders.

The design of compression of the stall in the decisive avenue at the cost of space is that prevalent by the registers it will not yield pipeline moment inward the style. Accordingly, the advocated ISA model, the draft non-pipelined alone compared with pipelined decisive, and it also required additional registers. In this dissimilar attitude, we need to separate the prescribed ISA design into an entire variety of phases by pipelining it. At this moment, it will compose our architecture desired for clock-gating. In this advanced behavior, we are required to gate the clock signal, a certain delivery into the individual's phase. On this accomplishment, the excellent nodes of our architecture are set back from the clock varies, which remarkably decreases competence dissipation.

In this additional method, the corresponding gating is accurate exclusively all over the start and end conferences. On initiation of expansion following pipeline phases (against the yield side) of the outlining field unit standard, these divisions are clock-gated.

The 32-bit pipelined ISA depleted the value of 14.107 watts, which resolved this phase of 7.7488 watts into lower power compared to non-pipelined ISA. It is generally complete because of the performance of clock gating proficiency and further being of pipelining the essential avenue stall of the non-pipelined (32-bit) adder, which is 13.462 ns section, and the pipelined is 4.7368 ns section. According to that pipelining, the stall is decreased by 8.7 ns. The on-leading of the bench displays the identification of the aggregate area drain in charge of LUT's power drain, the essential avenue stall, and further the almost clock oscillation, which is 8, 16, and 32 bits non-pipelined. In that, the adder bequeath has capacity decline as correlated along the non-pipelined alone. This performance contains the organization and panel layout clone conclusion of three compositions of individual pipelined and non-pipelined adders for the desired correlation to bring about the collection of LUTs required and plenty of power dominated by individual adders.

Along with adequate deterioration in efficiency and measures, it's achievable to assume immense momentum, flat potential, and area cost-effective characteristics. Exploitation imprecise relative circuit capability. The efficiency of corresponding circuits can be indexed off to appreciate the proficiency and velocity by consideration. In that the equivalent adders field system is known as ISA. Despite resistance to equivalent adders to enhance velocities permeable the efficiency and least possible fault. In that, at the time of production field system, our supplements are as pursued. Designate and reasoning of the CLA and fundamentally positioned ISA have contributed. After that, cereal pipelined to extend and reverse the decisive avenue stall. Such several enlarge in the application velocity when the adder is acknowledged.

CONCLUSION

The suggested architecture design generates low power and high speed in the architecture. To decrease the delay and power consumption by using clock gated and also using fine grain pipelined in the field, programmable gate array test results seem to indicate that the changed architecture working at a maximum frequency is 139.44 MHz. Afterward, 14.1072 W of power was consumed by the 32-bit pipeline architecture. So, this design uses many applications and IoE in electronic devices. In the low-power technology, area-related problems may occur.

REFERENCES

[1] Liu T, Lu SL. Performance improvement with circuit-level speculation. In Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture. 2000 Dec 1;348-355.

[2] AL-Nabi NR, Laith AK, Ammar SM, Ahmed RA, Laith AA. Design and Implementation of a Low-cost IoT Smart Weather Station Framework. Journal of Internet Services and Information Security. 2024;14(2): 133-144.

[3] Shrestha R. High-speed and low-power VLSI-architecture for inexact speculative adder. In IEEE International Symposium on VLSI Design, Automation and Test (VLSI-DAT) 2017 Apr 24;1-4.

[4] Surendar A, Saravanakumar V, Sindhu S, Arvinth N. A Bibliometric Study of Publication-Citations in a Range of Journal Articles. Indian Journal of Information Sources and Services. 2024;14(2):97-103. https://doi.org/10.51983/ijiss-2024.14.2.14

[5] Camus V, Schlachter J, Enz C. Energy-efficient inexact speculative adder with high performance and accuracy control. In IEEE international symposium on circuits and systems (ISCAS) 2015 May 24;45-48.

[6] Baranov OV, Smirnov NV, Smirnova TE, Zholobov YV. Design of a quadrocopter with PID-controlled fail-safe algorithm. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications. 2020;11(2):23-33.

[7] Boopathy V, Kalirajan K, Mounitha R. A Review on Evolution of Approximate Truncation Multipliers. In IEEE 9th International Conference on Advanced Computing and Communication Systems (ICACCS) 2023 Mar 17;1:391-394.

[8] Nur G, Barış BN, Levent B, Sazaklıoğlu BS, Ak E. BUSER Transcutaneous Electric Nerve Stimulator Device Design. Natural and Engineering Sciences. 2023 Apr 1;8(1):18-30.

[9] Boopathy EV, Usha M, Swathi R, Usha A, Subhashree N. Design and analysis of high speed low area-power semi-custom standard library cells using 90nm MOCMOS technology. InJournal of Physics: Conference Series 2019 Nov 1;1362(1):012004. IOP Publishing. doi:10.1088/1742-6596/1362/1/012004

[10] Hu J, Qian W. A new approximate adder with low relative error and correct sign calculation. In IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE) 2015 Mar 9;1449-1454.

[11] Stevovic S, Jovanovic J, Djuric D. Energy efficiency in urban areas by innovative permacultural design. Archives for Technical Sciences, 2018;2(19):65–74.

[12] Veera BE, Swadhi PM, Sujitha S, Susmitha R, Sonia SV. Realization of high speed low power MCC adder using dynamic CMOS transistors. In IEEE International Conference on Current Trends towards Converging Technologies (ICCTCT) 2018 Mar 1;1-5.

[13] Kim Y, Zhang Y, Li P. An energy efficient approximate adder with carry skip for error resilient neuromorphic VLSI systems. In IEEE/ACM international conference on computer-aided design (ICCAD) 2013 Nov 18;130-137.

[14] Thakur G, Sohal H, Jain S. FPGA-based parallel prefix speculative adder for fast computation application. In IEEE Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC) 2020 Nov 6;206-210.

[15] Kishore KH, Prasad BK, Teja YM, Akhila D, Sai KN, Kumar PS. Design and comparative analysis of inexact speculative adder and multiplier. International Journal of Engineering and Technology (UAE), ISSN No. 2018:413-426.

[16] Zhu N, Goh WL, Wang G, Yeo KS. Enhanced low-power high-speed adder for error-tolerant application. In IEEE International SoC Design Conference 2010 Nov 22;323-327.

[17] Boopathy V, Raghul G, Karthick K. Low power and high performance MOSFET. In IEEE International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA) 2015 Jan 8;1-5.

[18] Weber M, Putic M, Zhang H, Lach J, Huang J. Balancing adder for error tolerant applications. In IEEE International Symposium on Circuits and Systems (ISCAS) 2013 May 19;3038-3041.

[19] Esposito D, De Caro D, Strollo AG. Variable latency speculative parallel prefix adders for unsigned and signed operands. IEEE Transactions on Circuits and Systems I: Regular Papers. 2016 Jul 9;63(8): 1200-1209.