

Implementation of Arithmetic Logic Subsystem in a Sliced Processor

K.Rajeshwaran¹, K.Sudhakar², A.Kavitha³, M.Geethalakshmi⁴, C.Palaniappan⁵

¹Assistant Professor, Sri Ramakrishna Engineering College, Coimbatore, India

²Assistant Professor, M.Kumarasamy College of Engineering, Karur, India

³Professor, K.Ramakrishnan College of Technology, Trichy, Tamilnadu, India

⁴Assistant Professor, Kongunadu College of Engineering and Technology, Trichy, Tamilnadu, India

⁵Assistant Professor, Sri Bharathi Engineering College for Women, Pudukkottai, Tamilnadu, India

ABSTRACT

The designed estimated CLA (carry look-ahead adder) is swift and power (RAP-CLA). This adder will move between preliminary and precise modes of operation, making it appropriate for both error-tolerant and exact implementations. The framework, which is more space and power effective than current transportable estimated ripple carry adder, is accomplished by certain modifications. The findings show that, in the estimated mode of operation, the conceptual 32-bit adder achieves greater delay and energy reduction than the identical CLA, while maintaining a low error rate. It also has a smaller value and energy consumption than the other estimated adders investigated in this paper. Eventually, the suggested adder's usefulness is shown in two computer vision applications: smoothing and polishing. The CLA is then incorporated into the ALU, and the entire module is then used to create a module. Finally, these components are incorporated into a sliced processor to minimize area, energy, and delay.

Keywords: Ripple Carry Adder, Carry Look Ahead Adder, Delay, carry and Power

Introduction

Glorious results, i.e. reliable computing results, may not be needed in practical uses such as communication systems, electronic communication, artificial intelligence, computer animation, machine vision, big data, data mining, cloud services, biometric data, neural network computing, and so on. Alternatively, it might be acceptable to consider roughly accurate calculation outcomes that are contained within a given error bound. This is accomplished by using the limitations of human vision.

An online verifiable complete adder and an online verifiable n-bit ripple carry adder were proposed by Bose et al. To build a lightweight online verifiable complete adder and also a ripple carry adder that can be tested online [1]. The authors identify the CFTFA gate, a parity-preserving adder gate that optimizes the overall amount of gates, trash outputs, quantum cost, and essential inputs of the circuitry reversible online verifiable complete adder [2]. The CFTFA gate improves the number of gates by 25%, quantitative cost by 42.30%, and the amount of constant inputs by 50% over the current best. Due to the longest carry delay time, the Ripple carry adder conducts slower addition.

The asynchronous power sensitive of the carry save adder is created by Benet et al. VLSI power architecture restrictions have slowly increased over the last decade. Power consumption is a major consideration in VLSI architecture [3]. The dissipation is the source of the energy losses. It was once overlooked, but now it absorbs half of the overall power used by standard high VLSI chips. Certain power restrictions have been greatly alleviated by the complications that have been shrunk into the central submicron region, and the immobile certain adder now has the highest overall depravity [4-6]. The main benefit of the carry save adder is that it has less replication

interference, and it is then motionless, which is directed at the power degeneracy and peak power leakage that occurs with the carry save circuit. It takes up more space and uses more energy than Ripple Carry [13]. The Parallel Simultaneous Self Timing Adder (PASTA) is created using a recursive formulation and a grid structure. A multiplexer is being used in the configuration to prevent interconnection issues [14]. To stop carry chain propagation, the process is performed in parallel. Due to carry core feature, there is an issue with high fan-in and fan-out in the carry-look ahead adder [15]. To prevent high fan-out, a PASTA complete detection unit is present to bind all carries. For simultaneous logic, where transistors are linked in parallel, a high fan-in is inevitable.

Design of Sliced Processor

The most significant procedure a processor performs is conditional addition. Furthermore, some digital devices, such as the Special Purpose Processor (GPP), can do both estimated and exact calculations [7]. This will not necessitate the use of an additional correction device.

Carry Look Ahead Adder Circuit

A CLA adder is a power efficient adder that decreases propagation delay by using more renowned technology. As a result, it is more expensive. The carry concept of the adder over the fixed group of bits is simplified to the two level logics in this system, which is nothing more than a conversion of RCA model [8].

Table1: Truth Table for CLA

A	B	CR _i	CR _{i+1}	DETAILS
0	0	0	0	Carry Generation Not Available
0	0	1	0	
0	1	0	0	
0	1	1	1	Carry Propagation Not Available
1	0	0	0	
1	0	1	1	
1	1	0	1	Generation of Carry
1	1	1	1	

This approach uses logic gates to examine the lower order bits of the augend and multiplicand to determine not whether a higher cognitive carry should be produced. Let's take a closer look.

Include the complete adder circuit with accompanying truth table shown above. If two factors are described as carry create M_i and carry propagation N_i ,

$$M_i = A_i \oplus B_i$$

$$N_i = A_i B_i$$

The sum output and carry output is

$$S_i = M_i \oplus CR_i$$

$$CR_{i+1} = N_i + M_i CR_i$$

Where M_i is a carry generate that generates the carry independent of the input carry while both A_i and B_i are one. P_i is a carry propagate that is connected to the carry propagation from CR_i to CR_{i+1} . The carry output Boolean function of each stage in a 4 stage carry-Look ahead adder can

be expressed as

$$CR1 = N0 + M0 CRin$$

$$CR2 = N1 + M1 CR1 = N1 + M1 N0 + M1 M0 CRin$$

$$CR3 = N2 + M2 CR2 = N2 + M2 N1 + M2 M1 N0 + M2 M1 M0 CRin$$

$$CR4 = N3 + M3 CR3 = N3 + M3 N2 + M3 M2 N1 + M3 M2 M1 N0 + M3 M2 M1 M0 CRin$$

It can be seen from the above Boolean calculations that CR4 does not even have to wait for CR3 and CR2 to propagate; in reality, CR4 is promulgated simultaneously with CR3 and CR2. Since the sum of components is the Boolean expression for each carry output, this can be executed with one set of AND gates accompanied by such an OR gate.

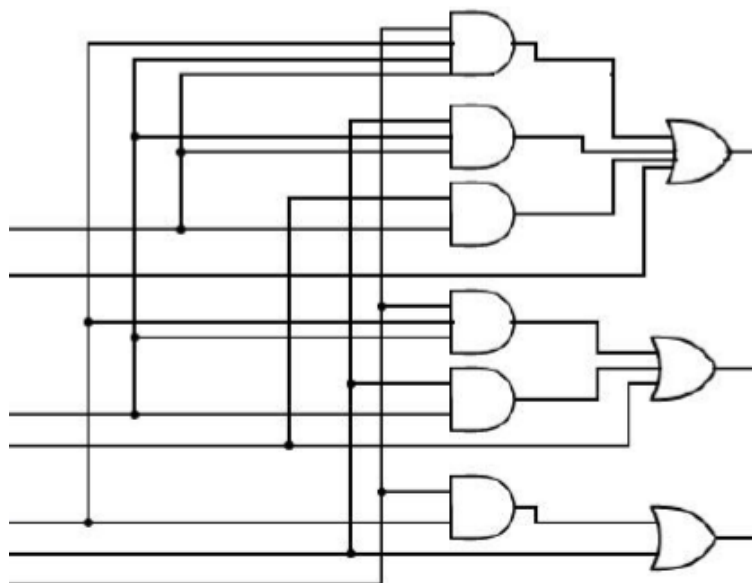


Figure1.Logic Diagram for CLA

As a result, as seen in the diagram below, a 4 bit parallel adder can be introduced with the carry-Look ahead system to speed up assessment and ensure. Every sum output requires two Ex-OR gates in this case. The first Ex-OR gate produces the Pi variable, while the AND gate produces the Gi parameter.

As a result, all of these N's and M's are created in two gates stages. The CLA generator enable all of these N and M signals to spread once they have reached their steady-state values, and they generate the output carriers with a two-level gate delay. As a result, the propagation latency times of the sum outputs S2 to S4 are equal.

By cascading the number of 4 bit adders with carry logic, 16 bit and 32 bit parallel adders can also be installed. A 16-bit CLA is made by cascading four bit adders with following two gate delays, while a 32-bit carry-Look ahead adder is made by rippling two 16-bit adders. The gates delays for CR16 and SI15 in a 16 bit CLA are 5 and 8, however, which are less than the 9 and 10 gate slows for CR16 and S15 in cascaded four bit CLA adder frames.

Likewise, in a 32 bit adder, CR32 and S31 require 7 and 10 gate waits, respectively, which are less than the 18 and 17 gate delays needed for the same outputs if the 32 bit adder is configured using eight 4 bit adders

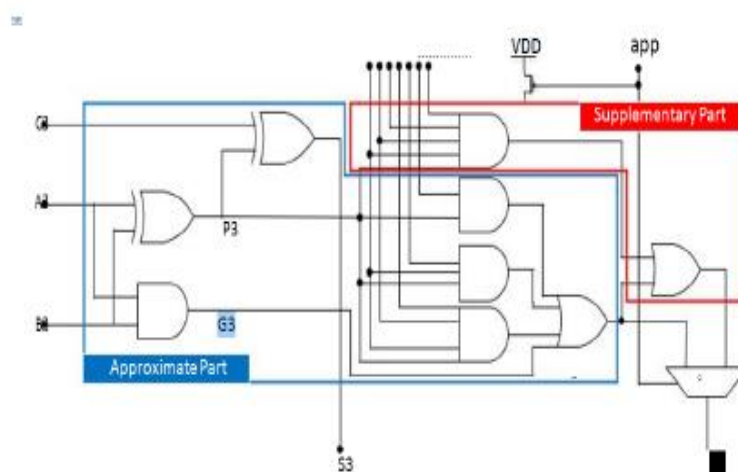


Figure2.Gate level net list of determining c4 in the RAP-CLA structure

The architecture of a bit-slice microprocessor has three major merits. As a arithmetic logic units can be connected in horizontal arrangements to produce machines that can process vast amounts of data at once. The *AMD 2901* is an example of a bit-slice processor. The *2901* is the arithmetic logic units and the *AMD 2910* is the CU in the *AMD 2901* bit-slice.

The Intel 8080 was a competitor, but the *8080* could only accommodate 8 bits of data at a time. The *2901* was a 4-bit ALU, but four of them could be linked together just to make a 16-bit machine, eight could be linked together just to make a 32-bit computer, and so on. While the *8080* will have to process 16 or 32 bits in several loops, the acceptable *2901* design could do so in a single loop, allowing the machine considerably more power than that of the *8080*.

The Bit-slice architecture has the benefit of allowing the processors to use bipolar chip technology due to the two chip design (example: the Intel 3002). Bipolar is extremely fast, but it absorbs a lot of energy and generates a lot of heat. Bipolar chips could not be as dense (in terms of the number of transistors per area) as PMOS or NMOS chips due to the heat dissipation problem.

Bipolar technologies could not be used to make single-chip CPUs. As a result, in addition to the broader data paths that bit-slice devices could achieve, the bipolar hardware used to create the chips made them intrinsically quicker.

Arithmetic Logic Unit

The following functions are performed by an Arithmetic unit: Addition, Adding withhold, Subtraction, Simple arithmetic with borrow, Decrement, Increment, and Transfer. We get 32-bit output since the data is 32-bit [16]. Since the arithmetic can only display one output at a time, a selector is needed to choose one of the operators.

The aim of reconfigurable CLAs is to boost the delay efficiency of the adders. The suggested adders are intended for use in a SLICED processor with a tweaked ALU. The proposed simulation outcome is described in this part.

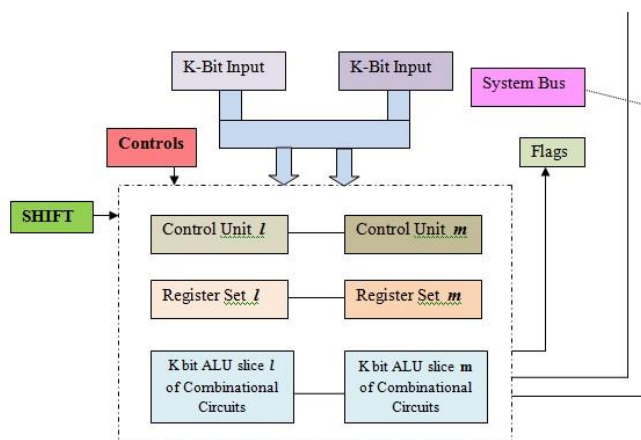


Figure3. Sliced Processor Blocks

Logic Unit

The following operations are performed by a logic unit: logical Plus, logic *OR*, logical *XOR*, and logical *NOT*. We'll create a logic unit that can execute the four simple logical micro operations *OR*, *AND*, *XOR*, and Supplement, so all other logic micro-operations can be deduced from these four.

Shift Unit

A Shift Register is a type of storage system that stores or transfers binary data. Consider the following two registers.

Left shift

Right shift

Rotate Bits of A Number

Bit rotation (also known as circular shift) is a similar process to shift, with the exception that bits that break off at one end are returned to the other. The pieces that come off at the left end are inserted back at the right end of left rotation. The pieces that come off at the right end are returned to the left end of right rotation.

Arithmetic Logic Unit

Split Arithmetic Logic Unit into three different three modules, Arithmetic, Logic, and Change, is the technique used in these. The earlier described arithmetic, logic, and shift are combined into an Arithmetic Logic Unit with a same selection line. The micro shift operation is often performed in a separate unit, although the shift unit is also integrated into the overall Arithmetic Logic Unit.

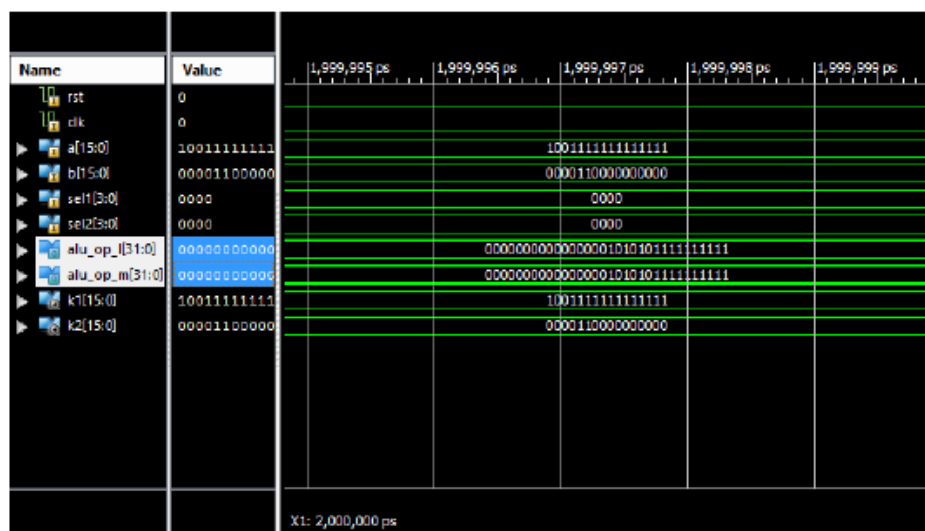


Figure4 Output of Sum Operation

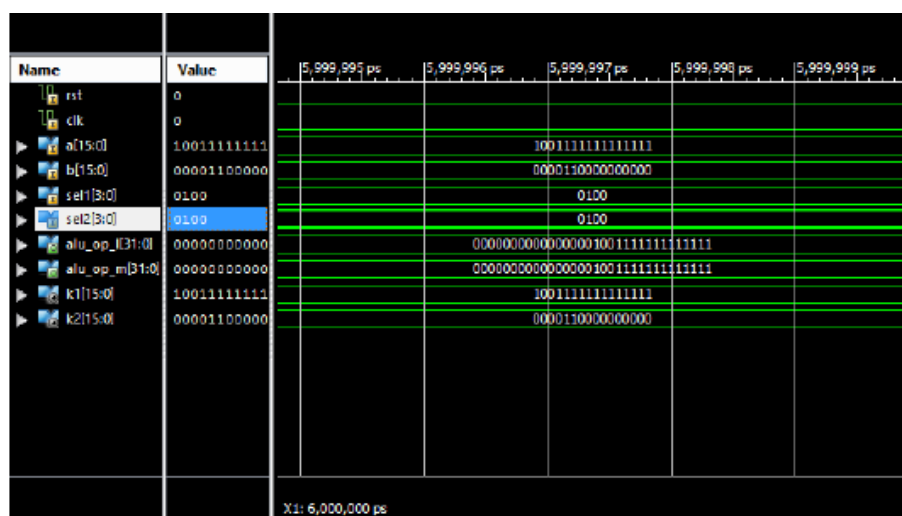


Figure5.Output of Logic OR Operation

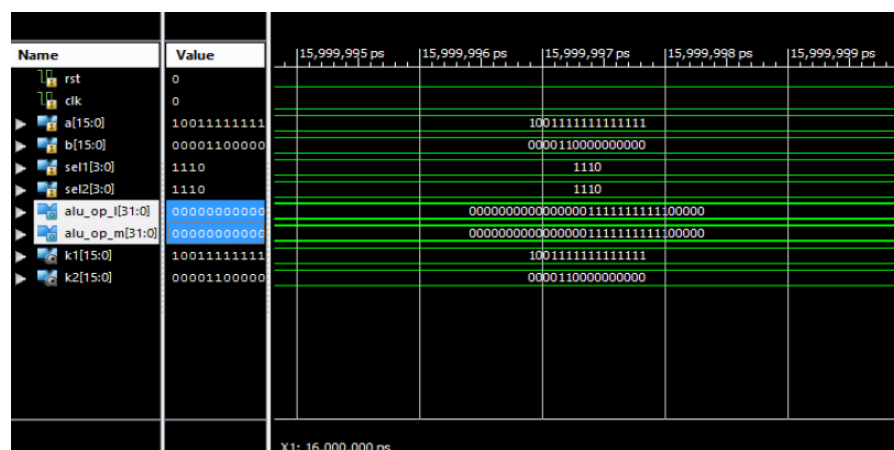


Figure6.Output of Shift Right Operation

Conclusion

This project designed and implemented in ALU a swift yet energy efficient reconfigurable approximate carry look ahead adder with the ability to transform between approximate and exact modes, resulting in a 32-bit sliced processor with low area and power consumption.. The sliced processor solution could be used in signal processing and other image processing applications in the future. It can be found in the Internet of Things and its implementations due to the reduced latency.

References

- [1] B. K. Mohanty and S. K. Patel,(2014) "Area–Delay–Power Efficient Carry-Select Adder," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, no. 6, pp. 418-422, June 2014.
- [2] B. Shao and P. Li(2015), "Array-Based Approximate Arithmetic Computing: A General Model and Applications to Multiplier and Squarer Design," IEEE Transactions on Circuits and Systems Regular Papers, vol. 62, no. 4, pp. 1081-1090.
- [3] A. Raha, H. Jayakumar, and V. Raghunathan, (2015) "Input-Based Dynamic Reconfiguration of Approximate Arithmetic Units for Video Encoding," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 99, pp. 1-1,
- [4] M. S. Khairy, A. Khajeh, A. M. Eltawil and F. J. Kurdahi, "Equi-Noise: A Statistical Model That Combines Embedded Memory Failures and Channel Noise," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 61, no. 2, pp. 407-419, Feb. 2014.
- [5] R. Ye, T. Wang, F. Yuan, R. Kumar and Q. Xu, "On reconfiguration-oriented approximate adder design and its application," Proceedings of IEEE/ACM International Conference on Computer Aided Design (ICCAD), 2013, pp. 48-54.
- [6] A. K. Verma, P. Brisk and P. Ienne, "Variable Latency Speculative Addition: A New Paradigm for Arithmetic Circuit Design," Proceedings of Design, Automation and Test in Europe (DATE), 2008, pp. 1250-1255.
- [7]Sudhakar, K., Selvakumar, T., Jayasingh, T “Design and implementation of adaptive clock gating technique with double edge triggered flip flops” ICIIECS 2015 - 2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems, 2015, 7193249

- [8] Raghu, G., Sudhakar, K., Devi, M.G. “ Design and implementation of encoding techniques for wireless applications” IEEE International Conference on Circuit, Power and Computing Technologies, ICCPCT 2015, 2015, 7159313
- [10] Arunprathap, S., Sudhakar, K. “Printed circuit board design of compact CAN to ethernet converter” International Journal of Scientific and Technology Research, 2020, 9(2), pp. 591–595
- [11] Sakthimani, S., Kalaiarasan, R. “Investigation on analysis of power efficient 15/16 prescaler” International Journal of Scientific and Technology Research, 2020, 9(3), pp. 757–760
- [12] Dhamodaran, M., Jegadeesan, S., Murugan, A., Ramasubramanian, B. “Modeling and simulation of the flyback converter using SPICE model” International Journal of Recent Technology Engineering, 2019, 8(3), pp. 946–952
- [13] C Bhuvaneshwari, A Manjunathan, “Reimbursement of sensor nodes and path optimization”, Materials Today: Proceedings, 2020.
- [14] Bhuvaneshwari C, Manjunathan A, “Advanced gesture recognition system using long-term recurrent convolution network”, Materials Today: Proceedings, vol. 21, pp.731-733, 2020.
- [15] M Ramkumar, C Ganesh Babu, K Vinoth Kumar, D Hepsiba, A Manjunathan, R Sarath Kumar, “ECG Cardiac arrhythmias Classification using DWT, ICA and MLP Neural Network”, Journal of Physics: Conference Series, vol.1831, issue.1, pp.012015, 2021
- [16] K Balachander, G Suresh Kumaar, M Mathankumar, A Manjunathan, S Chinnapparaj, “Optimization in design of hybrid electric power network using HOMER”, Materials Today: Proceedings, 2020.