Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, May : 2024

DESIGN HIGH THROUGHPUT AND DENSITY OPTIMIZED MODULAR ADDITION USING CARRY BYPASS LOGIC

M. BALAJI REDDY[1], Dr.J.VK. RATNAM[2,] Dr.S.SURYANARAYANA[3]

[1]PG Student, [2]Professor, [3]Professor & Head

Department of Electronics and Communication Engineering,
Kallam Haranadhareddy Institute of Technology (Autonomous),
Guntur, Andhra Pradesh, India.

ABSTRACT:

Many applications in digital signal processing rely on online residue-based calculations, and a modular adder is a crucial arithmetic component for this. Realisations of modular multipliers and residue-to-binary converters rely on this fundamental building block. As a result, developing a modular adder with high throughput and low footprint is crucial. We provide a novel modular adder architecture in this study. The foundation of this system is the application of ideas created to implement binary adders.The thermometer coding (TC) and one hot coding (OHC) may be used to reduce the power consumption and improve the energy efficiency of modulo addition in compared to normal binary representations, making them suitable for use in low-power embedded and edge devices with small and medium dynamic range needs. The suggested modular adder improves upon this idea by employing the carry bypass adder approach to cut down on space and execution time.

Keywords: TC, OHC, Carry bypass adder, low power, FPGA.

I INTRODUCTION

Besides technological scaling, advances in the field of computer architecture have also contributed to the exponential growth in performance of digital computer hardware. The flip-side of the rising processor performance is an unprecedented increase in hardware and software complexity. Increasing complexity leads to high development costs, difficulty with testability and verifiability, and less adaptability. The challenge in front of computer designers is therefore to opt for simpler, robust, and easily certifiable circuits. Computer arithmetic, here plays a key role aiding computer architects with this challenge. It is one of the oldest sub-fields of computer architecture. The bulk of

hardware in earlier computers resided in the accumulator and other arithmetic/logic circuits. Successful operation of computer arithmetic circuits was taken for granted and high performance of these circuits has been routinely expected. This context has been changing due to various reasons. First, at very high clock rates, the interfaces between arithmetic circuits and the rest of the processor become critical. Arithmetic circuits can no longer be designed and verified in isolation. Rather an integrated design optimization is required. Second, optimizing arithmetic circuits to meet the design goals by taking advantage of the strengths of new technologies, and making them tolerant to the weakness, requires a reexamination of existing design paradigms. Finally, incorporation of higher-level arithmetic primitives into hardware makes the design, optimization and verification efforts highly complex and interrelated. The core of every microprocessor, digital signal processor (DSP), and data processing application-specific integrated circuit (ASIC) is its data path. With respect to the most important design criteria; critical delay, chip size, and power dissipation, the data path is a crucial circuit component. The data path comprises various arithmetic units, such as comparators, adders, and multipliers [4]. The basis of every complex arithmetic operation is binary addition. Hence, it can be concluded that binary addition is one of the most important arithmetic operations. The hardware implementation of an adder becomes even more critical due to the expensive carry-propagation step, the evaluation time of which is dependent on the operand word length. The efficient implementation of the addition operation in an integrated circuit is a key problem in VLSI design [8]. Productivity in ASIC design is constantly improved by the use of cell-based design techniques – such as standard cells, gate arrays, and field programmable gate arrays (FPGA), and low-level and high-level hardware synthesis [10]. This asks for adder architectures which result in efficient cell-based circuit realizations which can easily be synthesized. Furthermore, they should provide enough flexibility in order to accommodate custom timing and area constraints as well as to allow the implementation of customized adders. The tasks of a VLSI chip are the processing of data and the control of internal or external system components. This is typically done by algorithms which are based on logic and arithmetic

operations on data items [10]. Applications of arithmetic operations in integrated circuits are manifold. Microprocessors and DSPs typically contain adders and multipliers in their data path.. Adders, incrementers/decrementers, and comparators are often used for address calculation and flag generation purposes. ASICs use arithmetic units for the same purposes. Depending on their application, they may even require dedicated circuit components for special arithmetic operators, such as for finite field arithmetic used in cryptography, error correction coding, and signal processing. This paper is organized as follows. Section I deals with the introduction of the paper. Section II deals with the Main Aim of the paper. Section III deals with survey research of the paper. Existing system is given in section IV. Proposed system and methodology explanation in section V. Results are explained in section VI. Finally Conclusion of the paper is given in section VII.

II MAIN AIM

The major contributions if this work include:

1. The designed modular addition circuit capable of processing addition operations at a high rate, ensuring efficient operation in real-time .
2. The core design principle of the project is the utilization of carry bypass logic to optimize carry propagation in the addition circuit. Carry bypass logic allows for efficient handling of carry signals, reducing propagation delays and improving overall throughput.
3. By minimizing the hardware resources and optimizing the logic design, the modular addition circuit potentially consumes less power, making it suitable for use in energy-constrained environments or battery-powered devices.
4. The addition circuit supports modular arithmetic operations, allowing for addition operations to be performed within a specified modulus or range. This is crucial for applications such as cryptography, where modular arithmetic is commonly used for operations like encryption and decryption.

III LITERATURE SURVEY

First a brief survey on modular adders were made and discussed the designing techniques of random number generators based on modular adders. Modular adders can be classified into two types: the general modular adder and the special modular adder and it is based on the form of modulus. In the former adder design the two values A+B and A+B+T should be computed first and one of them is

selected as the final output. Bayoumi and Miller [2] proposed a general modular adder for arbitrary modulus by using 2 cascaded binary adders and its delay is the sum of two binary adders. Several modular adders with two binary adders to calculate A+B and A+B+T were proposed subsequently. However this approach offers better delay performance the area consumption is relatively larger and it is twice the binary adder. Reused binary adder configuration [3] is the another type of general modular adder design and was proposed by Dugdale. The drawback of this type of adder is that it will use two operation cycles to perform one modular addition. Subsequently many studies on modular adders were done that have better area and delay performance. A high speed and reduced area modular adder structure for RNS were proposed by Hiasat[6] where any regular carry lookahead based binary adder can be used inthe final stage. This structure needs an extra CLA unit to get the carry out bit of A+B+T before the final CLA addition and as a result delay is not reduced significantly. ELMMA [9]algorithm is another popular modular adder design proposed by R.A. Patel. In this adder two carry computation modules for A+B and A+B+T were used and some carry computation units were shared. But in the worst cases almost two independent carry generation modules were used.

IV EXISTING TECHNIQUE

Thermometer Coding With TC , the value of each number is expressed as the number of ones in a string of bits. Since, in this coding, no weight is assigned to bit positions, it is not important where the ones are placed. However, for simplicity, these 1s are usually placed at one end of the string . As an example, the numbers between 0 and 7 are represented as shown in Fig .4

I. Shows that by increasing the range of numbers, the amount of required bits in a TC representation grows at a fast pace. Therefore, the TC is not suitable to represent large numbers. Nevertheless, the decomposition of numbers in a set of small residues makes TC−based fast circuits suitable for RNS. Therefore, combining TC with RNS improves the performance and efficiency of arithmetic systems. The application of

the thermometer code to RNS residues is herein designated TC for RNS (TCR). In, it is stated that the number bits required to represent all the remainders modulo mi are mi . However, it should be mentioned that, since the leftmost bit is always zero, these numbers can, in fact, be represented with only mi − 1 bits. In order to perform an addition in TC, one needs to count the number of ones in the two operands. If this number is greater than or equal to mi , the sum should be decreased by mi .

V PROPOSED SYSTEM

The structure is based on a combination of integration and expansion schemes [9] with the Conv-CSKA structure, therefore, defined by the CICSKA. It gives us the ability to use simple bypass logic. Logic replaces the 2:1 multiplier with AOI / OAI composite gates. Gates, consisting of fewer transistors, have lower latency, space, and lower power consumption compared to those of the 2:1 multiplexer [5]. Note that, in this structure, as the load is propagated in bypass logics, it is perfected. Therefore, at the bypass logic output of the equal sections, a carrying case is produced. The structure has a very low distribution delay in a small area compared to the conventional one. Note that although AOI (or OAI) power consumption is lower than that of a multiplexer, the power consumption of the proposed CI-CSKA is slightly higher than normal. This is due to the increase in the number of gates, which sets the wiring capacitance higher (in less important ways). Now, we describe the internal structure of the proposed CI-CSKA shown in Fig.1 in detail. The adder contains two N inputs, sections A and B, and Q. Each stage contains an RCA block with Mj size (j = 1,..,Q).

ONE-HOT CODING:

The OHC is usually used to address lookup tables (LUTs) and at the output of some linear circuits such as FIR filters [7]. K +1 bits are required to represent numbers between 0 and K in this coding. With OHC, only one bit takes the value of one and the others are zero. The value of the number in this coding is defined by the relative position of the bit with value "1."Shows the numbers between 0 and 7 encoded in an OHC. The OHC requires one more bit

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, May : 2024

than the TC. When the OHC is used to represent residues in RNS, it is named OHR [1]. The modular addition of two numbers A and B in this type of coding can be computed with shifts. To perform an addition, one operand should be circularly shifted a number of positions defined by the other operand. To perform (A − B) mod m, the complement of B should be added to A.

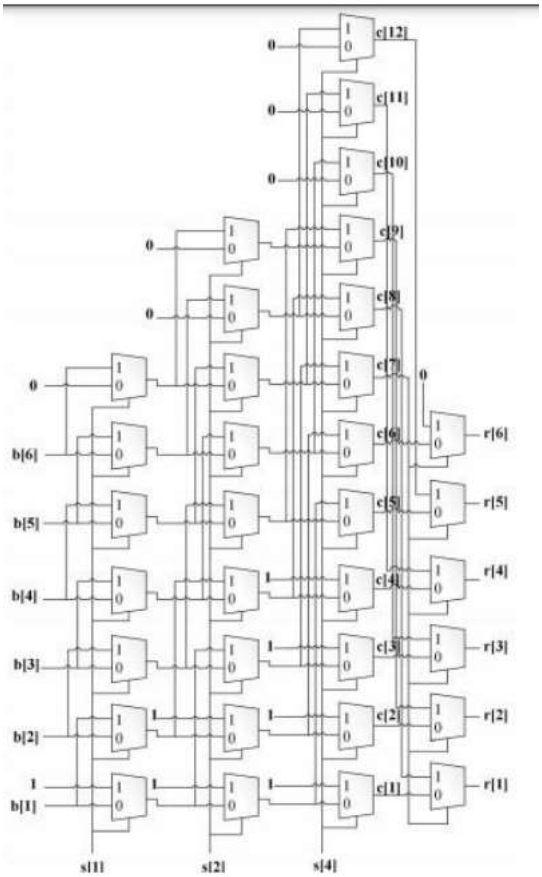

Fig.1. Modulo-7 TCR-based adder .

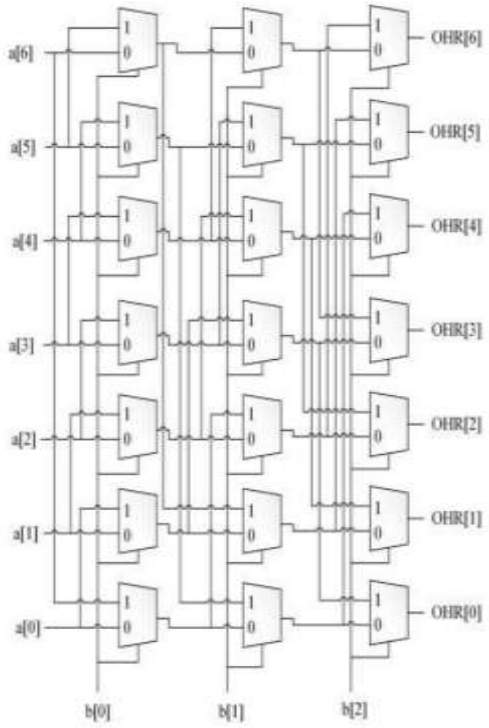

Fig.2. OHR-based modulo-7 adder

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, May : 2024

A + B < m, the result is placed in SUM0. As mentioned before, if at least one of the ANDs' output bits in the first level gets the value 1, the result of the modular addition of A and B is equal to or greater than m. Otherwise, the result is less than m. L0 signals are connected to the NOR gate with six inputs. Based on the output of this gate(sel), SUM0 or SUM1 is selected (sel is the complement ofcl). It should be noted that some multi-input gates in Fig.2 can be implemented using the tree structures of 2- input gates without impacting the delay. Let us analyze the operation of the circuit to compute SUM0 and SUM1. 1) SUM0 Circuit: As observed in Fig.3, with the bits in the reverse order, and A are the inputs of the NOR and AND gates in the first level. When A + B < m, the output of all the AND gates in the first level becomes 0, and the number of output bits of the NOR gates with value "1" is used to identify the number of zeros in SUM0. Therefore, if at least one of the Z0 signals becomes one, the number of zeros in the result is also at least one. Since, with TC, 0s are placed in the bits located on the left-hand side, the value of the left bit of SUM0 is equal to the T0 signal observed in Fig.5.

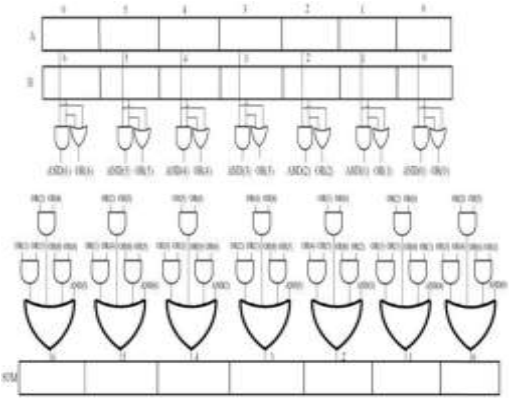

Fig 3 . Proposed OHR modulo adder for m = 7

VI SIMULATION RESULTS:

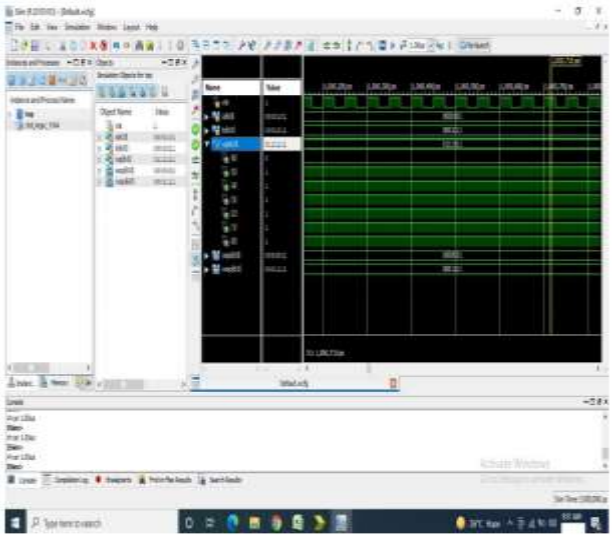

Fig.4. Existing system results.

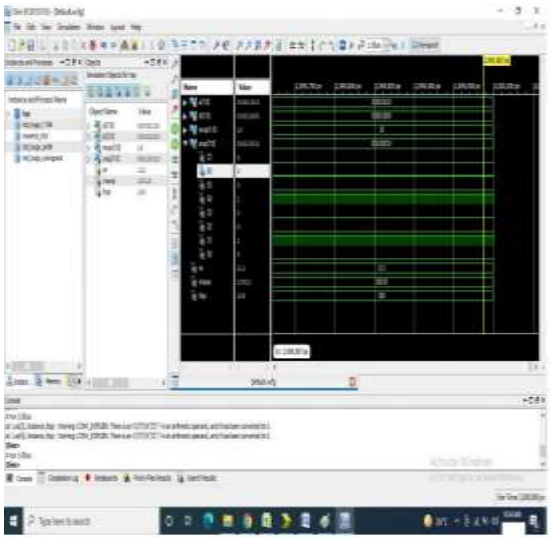

Fig.5. Proposed system results

VII CONCLUSION

OHC modular adders do not require carry propagation, their structures for small moduli become simpler and more efficient and has lower delay than binary modular adders. Finally designed Carry bypass adder based modulo adder occupies less area and has low latency. Proposed designed WOKS for any type of modulo addition with greater efficiency.

FUTURE SCOPE:

The future effort that can be foreseen as a result of the work done so far can be encapsulated as follows.

While possible applications for the shared moduli architectures have been presented, an evaluation of the effect of shared moduli architectures in these applications using VLSI metrics need to be done. This will complete the work on shared moduli architectures for moduli selectable from the set {2 n −1, 2 n , 2 n + 1 } by providing detailed information on the effects of using shared moduli structures. Such a study will thus set out the range and type of applications for which shared moduli structures provide the best value.

REFERANCES

[1] M. Bayoumi, G. Jullien, and W. Miller, “A VLSI implementation of residue adders,” IEEE Trans. Circuits Syst., vol. 34, pp. 284–288, March 1987.

[2] M. Dugdale, “VLSI implementation of residue adders based on binary adders,” IEEE Trans. Circuits Syst. II: Analog and Digital Signal Processing, vol. 39, pp. 325–329, May 1992.

[3] K. M. Elleithy and M. A. Bayoumi, “A θ (1) algorithm for modulo addition,” IEEE Trans. Circuits Syst., vol. 37, pp. 628–631, May 1990.

[4] A. Hiasat, “High-speed and reduced-area modular adder structures for RNS,” IEEE Trans. Comput., vol. 51, pp. 84–90, January 2002.

[5] L.Kalampoukas, D.Nikolos, C.Efstathiou, H.T.Vergos, and J.Kalamantianos, “High speed

parallel-prefix modulo 2n − 1 adders," IEEE Trans. Comput., vol. 49, pp. 673–680, July 2000.

[6] H.T.Vergos, C.Efstathiou, and D.Nikolos, "Diminished-one modulo 2n + 1 adder design," IEEE Trans. Comput., vol. 51, pp. 1389–1399, December 2002.

[7] C. Efstathiou, H. Vergos, and D. Nikolos, "Handling zero in diminished one modulo 2n + 1 adders," International Journal of Electronics, vol. 90, pp. 133–144, February 2003.

[8] G. Alia and E. Martinelli, "A VLSI modulo m multiplier," IEEE Trans. Comput., vol. 40, pp. 873–878, July 1991.

[9] T. Stouraitis, S. W. Kim, and A. Skavantzos, "Full-adder based arithmetic units for finite integer rings," IEEE Trans. Circuits and Syst. - II, vol. 40, pp. 740–745, Nov 1993.

[10] A. A. Hiasat, "New efficient structure of a modular multiplier for RNS," IEEE Trans. Comput., vol. 49, pp. 170–174, Feb 2000.