ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 10, Issue 5,May 2024

Efficient Binary Counters and Compressors and Bitonic Sorting

D.Gowthami[1], Tummalapenta Sai Sarath Saketh[2],
Sunkara Narmada[3], Gundu Naga Sowmya[4], Venna Hemanth Kumar Reddy[5]
[1]Assistant Professor,[2],[3],[4],[5]Undergraduate. Department Of ECE,
PSCMR College Of Engineering And Technology, Vijayawada, Andhra Pradesh, India
dgowthami@pscmr.ac.in[1], sarathsaketh2415@gmail.com[2], narmadanarmada33@gmail.com[3],
nagasowmya1102@gmail.com[4], hemanthkumarvenna07@gmail.com[5]

Abstract—Parallel counters play a pivotal role in many arith-metic circuits, especially high-speed multiplication operations. A key function in most digital signal processing units involves the summation of multiple operands simultaneously. Achieving fast compression requires the use of compressors and compressors with a high compression ratio. This article presents fast saturated binary counters and precision/approximation (4:2) compressors integrated into the sorting network. Sorting grids make it easy for counters to sort input sequences, especially those represented by only one hotcode sequence. In particular, three separate Boolean equations greatly simplify the calculation results when switching between modified sequences and single hotcode sequences. In addition, this research uses a parallel sorting algorithm to identify and sort the largest M values from N inputs, thus allowing the construction of scalable structures based on the proposed methodology. Various sorting methods have been devised to efficiently determine the highest price. In particular, BITONIC SORTING can be effectively implemented through parameter optimization, increasing computational efficiency.

Index Terms—Parallel counters,Arithmetic circuits,Digital sig-nal processing,Compression ratio,Sorting networks, Boolean equations, Parallel sorting algorithm,Scalable structures, Param-eter optimization, Bitonic sorting

I. Introduction

Energy reduction is a critical issue for modern electronic devices, especially those that are portable, such as smartphones and tablets. Achieving this size reduction without sacrificing performance speed is highly desirable. Digital signal process-ing (DSP) blocks play a key role in multimedia applications, with multiplication and addition making up most of the calcu-lations in these blocks. However, the primary function of these processing components, especially multiplication, can lead to significant power and energy consumption. Many DSP cores produce outputs such as images or videos that are intended for human consumption. Because human perception of such media has limitations, approximations can be used to improve the speed and energy efficiency of arithmetic circuits without degrading the quality of the output. In applications where arithmetic precision is not critical, approximate calculations offer advantages in terms of speed, power consumption, and energy efficiency. Different approximation techniques can be used at different design levels, including circuits, logic and architecture, as well as algorithms and software layers. These approaches include a series of arithmetic building blocks, such as adders and multipliers, each offering trade-offs between accuracy, speed, and power consumption. In fields such as cryptography, efficient computations in finite fields, which often require hardware implementations, are key. Different representations of array elements, such as polynomial bases, normal bases, and redundant bases, significantly affect the performance of an arithmetic circuit. For example, redundant base (RB) multipliers have gained attention due to their computational efficiency and regular computational structures. In the field of binary multiplication, various techniques such as Modified Booth method and Wallace Tree method have been proposed to optimize multiplier performance in terms of speed, power consumption and layout regularity. The choice of radix in the encoding of binary operands also affects the efficiency of multiplication operations, with a higher radix encoding potentially reducing the number of subproducts generated. Improvements in radix encoding and reduction tree design can lead to more energy-efficient and high-performance multipliers, especially in daisy-chained architectures where latency and power consumption are critical factors. Choosing radix values such as radix-16 presents challenges but offers op-portunities for optimization in concatenated multiplier designs. Ultimately, optimization of multipliers and arithmetic units is essential to achieve high-speed, low-power, and compact VLSI implementations that are consistent with the evolving requirements of modern electronic devices.

II. Literature Survey

In [3], the NBTI-sensing sleep transistor improves the lifetime stability of the investigated voltage circuit. Another method. The detection of functional symmetry and tran-sistor packing effects is the basis of the combined logic reconfiguration/pin-arrangement approach [6].[7] proposed an NBTI-aware technology mapping approach that guarantees chain lifetime.An old way to slow down aging is to use overkill techniques such as guardrails and over-extending gates. This method can be wasteful in terms of space and energy [8]. Path sensitivity was considered when developing this NBTI optimization method.] Some of these approaches require circuit modification or do not optimize a particular

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 10, Issue 5,May 2024

circuit. All VLSI circuits have their own delay, which degrades chip performance. In conventional designs, the clock cycle is determined by a factor known as critical path delay. In most worst-case designs, the probability of pMRSA activation of the critical pathway is low. Reducing the worst case can lead to a useful design in some cases. If the critical path delay is considered as the total cycle time of the non-critical path, there is a significant residual time. Because conventional circuits waste too much time, variable delay architecture has been proposed.]] with Booth's algorithm, the design of the multiplier pipe is shown [18]. In [19], where the variation of the tolerant design process for the arithmetic unit is presented, the effect of the variation of the process to increase the yield of the circuit is investigated. Worse, the clock cycle to the upper delay of the two-loop line will cause the system to fail if one is delayed. The researchers were able to reduce the waste of time in the conventional circuit to increase productivity, but they did not take into account the effects of aging and it could not adapt to the operating time. [20] and [21] developed a variable delay ad architecture that takes into account the effect of aging. The Booth radix-4 technique was used by Chen et al. (2003) make low-power 2-additive multipliers by reducing the switching activity of partial products. Before the calculation, cabinet code is generated for data with a more effective dynamic range, increasing the probability of zero partial products. A multiplier with such column-based city compressors or counters is created by using a dynamic range-determining unit to control the input data path. Both line-based trees and hybrids are used to implement two multipliers. further reduce power consumption. They can save the previous input state for the inefficient dynamic data range and thus reduce the number of switching operations. Theoretical analysis of the switching behavior of fractional products 27 obtained from the proposed multipliers with minimum power. 16 bits / spl times / sequential multiplication saves 31.2% 19.1% and 33.0% of the power used by the standard multiplier. In-line, additional hybrid-based trees are also reduced by 35.3 percent, 25.3 percent, and 39.6 percent for propagators proposed with in-line and hybrid-based trees. The Radix-4 Booth technique was used by Oskal et al. (2003) make low-power 2-additive multipliers by reducing the switching activity of partial products. Before calculating the two input data, a vertical code is generated for data with a narrow effective dynamic range, increasing the probability that the partial product will be zero. "Multiplier, which is a column-based propagation tree of compressors or counters, is created by using the dynamic range determination unit to manage the input data path. The line-based and hybrid additive tree is used to apply two multipliers. Then reduce energy consumption. Dynamic data range is not efficient. save the previous input state and reduce the number of switching operations for . 16 / spl times / 16 bit multiplier proposed by column-based splicing tree is stored. Hybrid-based splicing tree also reduces energy consumption by 35.3 percent, 25.3 percent and 39.6 percent.

III. ETHODOLOGY

This methodology describes Bitonic sort which is a parallel sorting method that relies on comparisons. It is known for its extensive use of comparisons compared to other sorting algorithms.

A. Bitonic Sorting

Bitonic sort is a parallel sorting algorithm based on comparison. Compared to other known sorting algorithms, Bitonic sort performs more comparisons. In parallel execution, this type is superior since the user compares data-independent components in a fixed order. Therefore, it is a good choice for equipping bit tools. First, we need to know what the Bitonic series is and how biton is produced before we can understand the Bitonic format. As one of the fastest sorting networks, Bitonic sort uses bitonic sorting algorithm. The order of comparison in the selection system does not depend on the data. For this reason, hardware or parallel processor arrays can be used to build a scheduling network. Bitonic sort is a sorting network with n • log(n)2 comparators. A typical joint and shell model has the same exponential complexity as this sorting algorithm. Although there is a sorting network with only O(n • log(n)) comparators [AKS 83], biton form is faster for all real-world sizes due to large volatility.

B. Bitonic Sequence

Based on the 0-1 principle, biton shapes are created here. Each sequence of zeros and series is sorted according to the 0-1 principle, forming a matching network that makes sequences of similar values equal. Bitonic mergesort algorithm is used for parallel sorting. It can also be used to design a point sorting system. This algorithm was developed by Ken Jagal. In this case, the sorting network consists of comparators, and the number of items to be sorted is delayed.

A monotonically decreasing (or increasing) sequence is a "sequence". A bitonic sequence can be considered to have variations in one direction and not the other.

A total of 16 numbers are fed into the system on the left and travel down every 16 wires before exiting the right end. This grid is built to accommodate the maximum number of items below the list.

They serve as pointers. It is necessary to compare the values at both ends of the arrow to ensure that the arrow always points to the larger number. If it is out of sequence, it is replaced. For clarity, colored boxes have no effect on the algorithm.

All red circles have the same structure, with all arrows pointing down (dark red) or up (light red) for each entry in the top half (light red). If the inputs are all in bitonic order, two sequences of bitons will be produced as output. As a result, each element of the upper half is less than or equal to the lower half (for dark red) or vice versa (for light red), the half will be bitonic. If you consider all possible inputs, you can use the null principle to verify this theorem, which is not immediately obvious.

The blue and green squares are formed by merging the red squares. A red box is applied to the entire input sequence, then to each half of the output, and so on. There are two possible outcomes: either all arrows point down (blue) or up (red) (green). The butterfly system is the name given to this organization. A bitonic input will produce an output sorted

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 10, Issue 5,May 2024

in ascending (blue) or descending (red) (green) order. If the first red box appears in a blue or green box, it will sort the numbers into the corresponding half of the list. A small red box will then divide the right quarter of the list. This process is repeated until everything is in place. The green or blue box output will be completely sorted as a result of this process.

The entire sorting system consists of green and blue bins. It can sort a set of inputs into the correct order with the largest value at the bottom. If two neighboring lists are both sorted, the result will be bitonic, since the top list is blue and the bottom list is green. Paired N-type sequences are performed in each column of blue and green boxes to create an N / 2 biton sequence. The boxes in that column then sort the order in that column. An ordered list of all inputs is created starting from each column in the unordered form.

Let us see in practical How Bitonic sort is done, following the above algorithm,Bitonic sequence: 2, 3, 4, 7, 9, 8, 6, 5

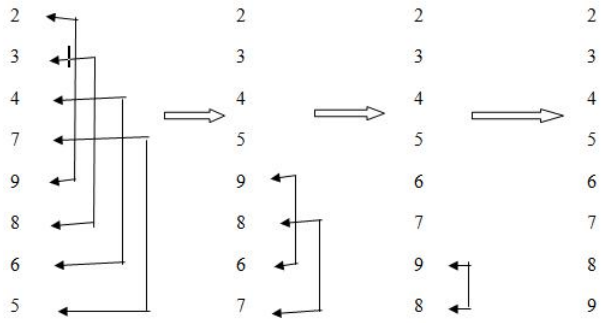

Fig. 1. Practical Representation of Bitonic Sort

Bitonic sorting may now be completed in this manner. A random sequence was applied to a Bitonic sequence above, and then the Bitonic sequence was used as an input for Bitonic sorting.. Beginning of first-half and beginning of second-half Therefore, there is no change. In the second half of the first half, there is a corresponding second half element. 3 x 8 = 0, hence there is no difference. Three elements of the first-half and three elements of second-half are summarised in this way: 4 6, therefore there is no change Due to the fact that 7 ¿ 5, the values of 5 and 7 have been switched. So far, so good, with the first half having all of its components arranged in ascending order. For the time being, I'll just sort the second half of the series. As a result, the first half-element ————- the first half-element 9 ¿ 6 are switched. As a result, the second element of the first half is exchanged with the second element of the second half, as 8 ¿ 7, and so on. As you can see, there is only one set of components left to compare, and that is 9¿ 8, which is why they were switched. 2, 3, 4, 5, 6, 7, 8, 9 are therefore the final Bitonic sorted results.

An efficient sorting system can be built by building a comparison network that can sort all biton sequences that either monotonically increase, then decrease, or monotonically decrease, then increase. The biton sequence contains, for example, 1, 4, 6, 8, 3, 2, 4, 6, and 9, 8, 3, 2, 4, 6. In the zero biton

We will build a comparison grid to sort the sequence of zeros and bits.

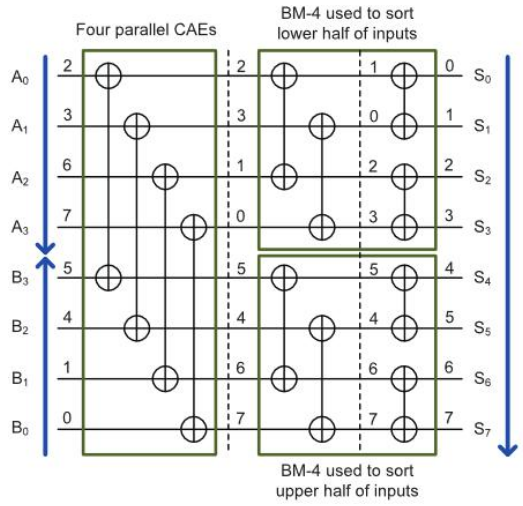

Fig. 2. An increasing 8-input bitonic merging unit

sequence, the structure is straightforward. For some, j and k have the form 0i1j0k or 1i1j1k for 0. Monotonically increasing or decreasing sequences are bitonic and monotonic.

As a blue box, each box goes through the same procedure, but the sorting is different for each box. After each green box is replaced by a blue box, a crossover can be used to run all the wires to the other side of the board. The arrows will all follow the same pattern, but the horizontal lines will be curved. Since the reverse of the binary search sequence is still bitonic, the crossover can be placed to the right of the bottom half of the output from each red block and the form will still work correctly. The internal reorganization of the red box can delete the two events that occurred before and after it. Since each green box becomes a blue plus and a cross, and the orange box becomes a red box obtained by two crossovers, the graph below is the same as before.

C. Half-cleaner

A bitonic sorter has several steps called semi-purifiers. In each semi-purifier, I compare the input string with the input string I + n / 2 for the value I = 1, 2, ..., n / 2. If n is an even number, we continue as before. A HALF Purifier [8] with eight inputs and 8 outputs is a HALF-Purifier [8]. Using a bitonic sequence of 0/1 as input, the semi-filter produces an output sequence with a small value at the top and a larger value at the bottom, both bitonic parts. This is called a semi-purifier. The moniker "semi-clear" comes from the fact that at least half-clear is always 0 or 1. (It should be noted that the whole sequence of bitons is pure.) The properties of these semipurifiers are shown in the following lemma.

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 10, Issue 5,May 2024

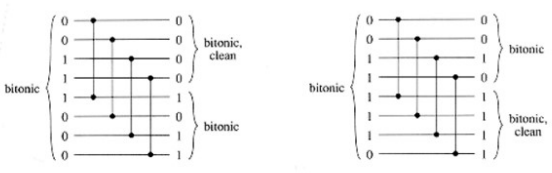

Fig. 3. The comparison network of Half-cleaner

semipure; both bitonic and each element in the upper half is at least as small as its lower half. The HALF-CLEANER[n] comparison system is used to compare two sets of inputs: a set of inputs (i + 1) and a set of inputs (i + n). Without losing generality, assume that the input is in this format:
0000000000000000000000000000000000000000000000000000

Fig. 5. Post-Synthesis Functional simulation of proposed (7,3) counter

. The picture shows four scenarios. Limma is preserved in each instance.

IV. Results and Discussions

The section examines the implications of the proposed sys-tem specifically. In the current system, Counters (6,3) and (7,3) have symmetrical input numbers. This design has full and half attachments. With longer delays and more XOR gates, more circuit area is used. As the number of XOR gates increases, the hardware complexity and power consumption of the circuit are also high. proposed counters (7,3) and (15,4) use a sorting system to organize the numbers in a symmetrical order. Using this new design will reduce the amount completely and half extra. Since these counters do not have XOR gates, latency is lower and the amount of circuit space usage is reduced. The device uses a small amount of electricity. The result provides more reliable and low complexity, low density and low cost, and minimum combination path delay and high throughput design is built by schematic representation of counters and measured by the number of field LUTs.

The Fig.4. is the Schematic Representation of the proposed (7,3) counter showing internal connections with all the I/p and O/p pins.This Synthesis converts a simple RTL design into a gated netlist with all constraints specified by the designer. In simple terms, synthesis is the process of converting an abstract design form into a properly implemented chip in terms of logic gates.Synthesis is a very important process for designers because it allows them to see what the design will actually look like after the design.All parameters such as area, time, power can only be reported and checked by the designer.

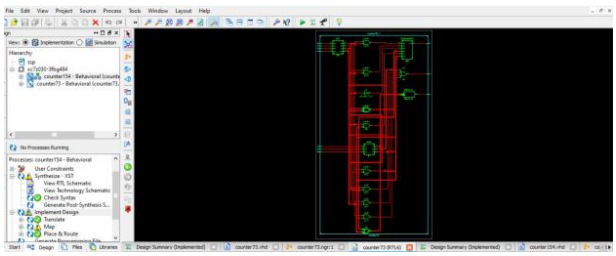

Fig. 4. Schematic Representation of (7,3) Counter

Fig. 5. is the post-synthesis Functional Simulation of pro-posed counter which shows the inputs and outputs of the designed counter and no.of 1's that are in the output.

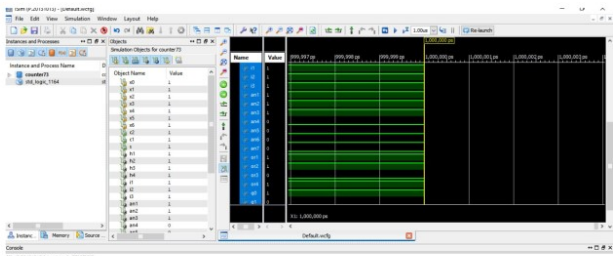

Fig. 6.illustrates the space utilized by the proposed (7,3) counter in LUT count.Here the no.of LUTs is 6.This demon-strates the efficiency gains in area utilization compared to the previous method.

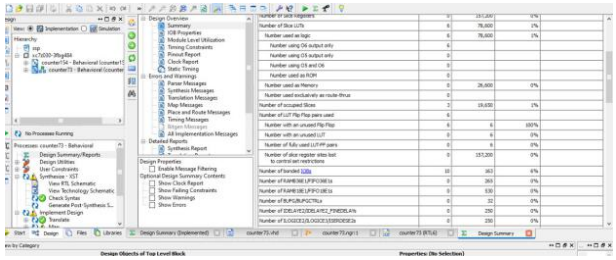

Fig. 6. Area(no. of LUTs) of (7,3) counter

Fig. 7. illustrates the Timing Analysis of (7,3) counter.It includes the Time Delay and Total time taken by the proposed counter.The Delay time is 1.355ns,in which the enhancement is shown.It is important parameter.It calculates the worst possible delay through each logic element,but not the logic operation of the circuit.It covers all steps of VLSI Physical Design like allocation, chip planning, placement, routing and finally clock routing. Because digital circuit timing is critical, three weeks will be fully devoted to static timing analysis (STA).

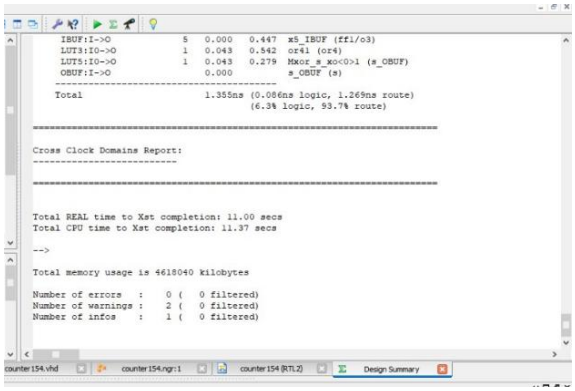

Fig. 7. Timing Analysis of (7,3) counter

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE)
Vol. 10, Issu

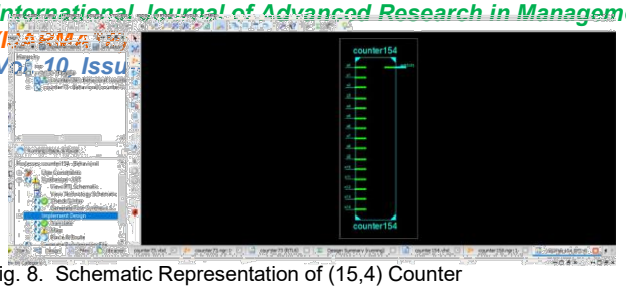

Fig. 8. Schematic Representation of (15,4) Counter

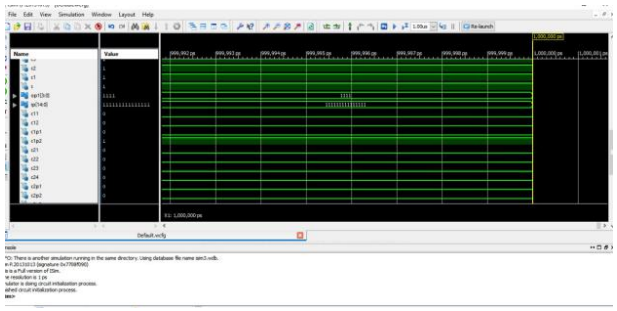

Fig. 9. Post-Synthesis Functional simulation of proposed (15,4) counter

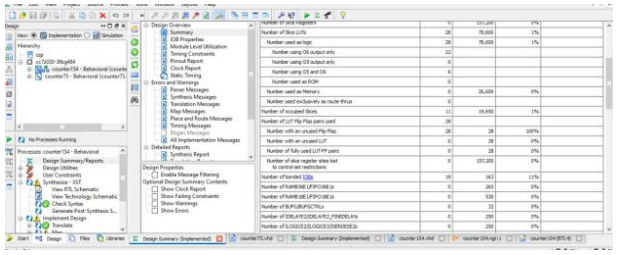

Fig. 10. Area(no. of LUTs)(15,4) counter

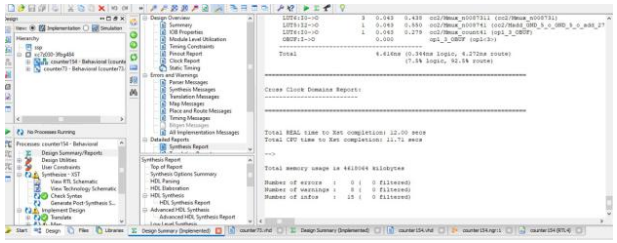

Fig. 11. Timing Analysis of proposed (15,4) counter

S/no	Design	Area (no. of LUTs)	Power (W)	Delay (ns)
1	Parallel (7,3) counter	12	0.081	2.050
2	Proposed (7,3) counter	06	0.042	1.355
3	Parallel (15,4) counter	67	0.081	4.764
4	Proposed (15,4) counter	28	0.042	4.616

Fig. 12. Comparison of the Previous method with Proposed method

Fig. 8. is a schematic representation of the proposed counter (15.4) showing internal connections with all I / p and O / p points. Fig. 9. shows the post-synthesis simulation results (15.4), No. of 1's after the sorting of the given inputs.Fig. 10. gives area consumed by the proposed (15,4) counter LUTs. Fig. 11. proposed (15,4) shows the delay time and the total time taken by the proposed counter.Fig. 12.gives comparison of existing method with the proposed method

V. Conclusion

In conclusion, the importance of parallel counters in various circuits, especially in fast multipliers, cannot be overstated. Digital signal processors rely heavily on parallel summation capabilities for many tasks, requiring counters and compressors with high compression ratios to operate efficiently. This paper presents the implementation of a fast saturated binary counter with an exact/approximate compressor (4:2) in a sorting network, demonstrating its effectiveness in reconstructing the input sequence and simplifying the output using Boolean equations. In addition, a parallel sorting algorithm is used to efficiently find and sort the largest values of the income set, which allows creating an enlarged structure based on the provided method. Although there are other sorting methods to determine the largest value, the efficient implementation of BITONIC SORTING through parameter optimization further demonstrates the importance of optimizing algorithms for high-performance computing.

Overall, the research presented in this paper contributes to the field of digital circuit design by emphasizing the impor-tance of parallel processing, efficient scheduling algorithms, and parameter optimizations to improve performance and scalability in various applications.

Here, an effective but potentially counterintuitive approach has been developed and implemented in the region. Symmetric Fractions: A Double Counter A method of summing and performing calculations is shown. Based on this novel inverse design methodology, we build counters like (7,3), (15,4) based on biotonic sorting. Because ADP has lower latency and higher throughput, the proposed counter is more adaptable than the current design.

References

[1] Wen-Chang Yeh and Chein-Wei Jen, "High-speed Booth encoded parallel multiplier design," IEEE Trans. on Computers, vol. 49, isseu 7, pp. 692-701, July 2000.
[2] Jung-Yup Kang and Jean-Luc Gaudiot, "A simple high-speed multiplier design," IEEE Trans. on Computers, vol. 55, issue 10, Oct. pp. 1253-1258, 2006.
[3] Shiann-Rong Kuang, Jiun-Ping Wang and Cang-Yuan Guo, "Modified Booth multipliers with a regular partial product array," IEEE Trans. onCircuit and Systems, vol.56, Issue 5, pp. 404-408, May 2009.
[4] Li-rong Wang, Shyh-Jye Jou and Chung-Len Lee, "A well-structured modified Booth multiplier design," Proc. of IEEE VLSI-DAT, pp. 85-88, April 2008.
[5] A. A. Khatibzadeh, K. Raahemifar and M. Ahmadi, "A 1.8V 1.1GHz Novel Digital Multiplier," Proc. of IEEE CCECE, pp. 686-689, May 2005.
[6] S. Hus, V. Venkatraman, S. Mathew, H. Kaul, M. Anders, S. Dighe, W. Burleson and R. Krishnamurthy, "A 2GHZ 13.6mW 12x9b mutiplier for energy efficient FFT accelerators," Proc. of IEEE ESSCIRC, pp. 199-202, Sept. 2005.
[7] Hwang-Cherng Chow and I-Chyn Wey, "A 3.3V 1GHz high speed pipelined Booth multiplier," Proc. of IEEE ISCAS, vol. 1, pp. 457-460, May 2002.

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 10, Issue 5,May 2024

[8] M. Aguirre-Hernandez and M. Linarse-Aranda, “Energy-efficient high-speed CMOS pipelined multiplier,” Proc. of IEEE CCE, pp. 460-464, Nov. 2008.
[9] Yung-chin Liang, Ching-ji Huang and Wei-bin Yang, “A 320-MHz 8bit x 8bit pipelined multiplier in ultra-low supply voltage,” Proc. of IEEE A-SSCC, pp. 73-76, Nov. 2008..
[10] S. B. Tatapudi and J. G. Delgado-Frias, “Designing pipelined systems with a clock period approaching pipeline register delay,” Proc. of IEEE MWSCAS, vol. 1, pp. 871-874, Aug. 2005.
[11] A. D. Booth, “A signed binary multiplication technique,” Quarterly J. Mechanical and Applied Math, vol. 4, pp.236-240, 1951.
[12] M. D. Ercegovac and T. Lang, Digital Arithmetic, Morgan Kaufmann Publishers, Los Altos, CA 94022, USA, 2003.
[13] C. S. Wallace, “A suggestion for a fast multiplier,” IEEE Trans. On Computers, vol. BC13, pp. 14-17, Feb. 1964.
[14] M.D. Ercegovac et al., ªFast Multiplication without Carry- Propagate Addition,º IEEE Trans. Computers, vol. 39, no. 11, Nov. 1990.
[15] R.K. Kolagotla et al., ªVLSI Implementation of a 200-Mhz 16,16 Left-to-Right Carry-Free Multiplier in 0.35m CMOS Technology for Next-Generation DSPs,º Proc. IEEE 1997 Custom Integrated Circuits Conf., pp. 469-472, 1997.
[16] P.F. Stelling and V.G. Oklobdzija, ªOptimal Designs for Multipliers and Multiply-Accumulators,º Proc. 15th IMACS WorldCongress Scientific Computation, Modeling, and Applied Math., A. Sydow, ed., pp. 739-744, Aug. 1997.
[17] Passport 0.35 micron, 3.3 volt, Optimum Silicon SC Library, CB35OS142, Avant! Corporation, Mar. 1998.
[18] G. Goto et al., ªA 4.1ns compact 54 54-b Multiplier UtilizingSign-Select Booth Encoders,º IEEE J. Solid-State Circuits, vol. 32, no. 11, pp. 1,676-1,682, Nov. 1997.
[19] G. Goto et al., ªA 54 54-b Regularly Structured Tree Multiplier,º IEEE J. Solid-State Circuits, vol. 27, no. 9, Sept. 1992.
[20] R. Fried, ªMinimizing Energy Dissipation in High-Speed Multipliers, º Proc. 1997 Int’l Symp. Low Power Electronics and Design, pp. 214-219, 1997.
[21] N.H.E. Weste and K. Eshraghian, Principles of CMOS VLSI Design: A Systems Perspective, second ed., chapter 8, p. 520. Addison Wesley, 1993.
[22] J. Fadavi-Ardekani, ªMN Booth Encoded Multiplier Generator Using Optimized Wallace Trees,º IEEE Trans. VLSI Systems, vol. 1, no. 2, June 1993.
[23] A.A. Farooqui et al., ªGeneral Data-Path Organization of a MAC Unit for VLSI Implementation of DSP Processors,º Proc. 1998 IEEE Int’l Symp. Circuits and Systems, vol. 2, pp. 260-263, 1998.
[24] J. Fadavi-Ardekani, ªMN Booth Encoded Multiplier Generator Using Optimized Wallace Trees,º IEEE Trans. VLSI Systems, vol. 1, no. 2, June 1993.
[25] K. Hwang, Computer Arithmetic: Principles, Architecture, and Design, chapter 3, p. 81. John Wiley Sons, 1976.