IMPLEMENTATION OF EFFICIENT DIVISION OPERATION IN ARITHMETIC LOGIC UNIT BY IMPLEMENTING VEDIC MATHEMATICAL ALGORITHMS TO IMPROVE THE PERFORMANCE OF THE PROCESSOR

Dr. Prasannakumar Mani, Assistant Professor, Department of Electronics, Sri Ramakrishna College of Arts & Science, Coimbatore -641006 :prasannakumarphd16@gmail.com

ABSTRACT:
The Arithmetic Logic Unit (ALU) is critical to processor performance, as division operations are computationally expensive compared to addition and multiplication. This study describes an efficient division operation in the ALU that uses Vedic mathematical techniques, notably the Nikhilam Sutra and the Paravartya Sutra, to optimize division speed and accuracy. Traditional division methods, such as restoring and non-restoring algorithms, necessitate several repetitions, which increases computing latency. The proposed Vedic division approach shortens execution time by breaking difficult division into smaller, more manageable parts, improving overall processing efficiency. Implementation in hardware description languages (HDL), such as Verilog or VHDL, shows reduced delay and resource consumption, making it appropriate for high-speed processors and embedded systems. Simulation findings show a significant boost in ALU performance, making this approach suitable for real-time computing applications. The incorporation of Vedic mathematics into ALU architecture improves computational efficiency, providing a faster and less power-intensive alternative to traditional division algorithms.
Key words: **ALU, Divider, Nikhilam Sutra, Paravartya Sutra, Hardware Description Languages**

INTRODUCTION:

The Arithmetic Logic Unit (ALU)'s division operation is a critical component of computational systems, allowing for fast arithmetic instruction processing. Traditional division techniques can entail difficult and time-consuming procedures. However, the division procedure can be considerably improved by incorporating Vedic mathematical techniques. Vedic mathematics, which is based on ancient Indian scriptures, provides straightforward and fast arithmetic operations. When used with ALU architecture, these techniques improve computing performance, reduce latency, and reduce hardware complexity. This novel technique has significant promise for boosting processor performance, particularly in applications demanding rapid and reliable arithmetic operations. Using division in high-precision, high-speed digital signal processing (DSP) units leads to increased latency and sophisticated hardware implementation because division is a sequential operation.

EXISTING DIVISION ALGORITHMS:

In the ALU of a CPU, the division operation is one of the most resource-intensive calculations. The graphics processing unit (GPU) plays a crucial role in the successful operation of the flag handling unit. In this way, the division process is sequential, making it more computationally complex than other numerical operations.There are two primary types of division algorithms: slow division (which includes restoring division and non-restoring division) and fast division (which includes the Newton Raphson and Goldschmidt methods). When performing a digital division operation, the divider is often subtracted from the current integer and then added back based on the outcome of the subtraction stage in traditional processing methods. After the deduction stage in a non-restoring system, the decision to do addition or subtraction is made. As a result, both restoring and non-restoring division algorithms require a significant amount of logic and supporting circuitry to be implemented.

PROPOSED DIVISION ALGORITHM:

Vedic mathematics is a distinct technique for conducting mathematical computations with origins in ancient Indian mathematics. Vedic Mathematics is an ancient Indian science with a remarkable

estimation technique founded on the 16 Sutras. Among the initially available sutras, Nikhilam Navatahsearamam Dasatah, Dhwajank, and Paravartya Yojaset are the ones that primarily focus on division. A divider is a crucial hardware component used in a variety of applications, such as digital signal processing, encryption and decryption algorithms in cryptography and various mathematical computations. With the advancement of technology, numerous researchers have attempted to design dividers with either high speed or low power consumption. Regularity of arrangement, resulting in less space, or even combining them in dividers. We believe that the Vedic partition best meets our requirements.

Three sutras based on Vedic mathematics can be used to perform arithmetic division

- **Dasatah, Nikhilam Navatahsearamam (All from nine, last from 10)**
- **Yojayet Paravartya (Transpose and Apply)**

The traditional division includes four terms:

(1) $Dividend\ (E)$ (2) $Quotient\ (Q)$ (3) $Divisor\ (D)$ (4) $Remainder\ (R)$

The following is the relationship between these four terms:

$$\boldsymbol{Dividend\ (E) = Quotient\ (Q) \times Divisor\ (D) + Remainder\ (R)}$$

PARAVARTYA YOJAYET ALGORITHM:

When the divisor is near to and greater than the base, sutra is used. This sutra will be used when the divisor numerals are substantial. The Paravartya Sutra aids in minimising computation, optimising processes, and maintaining accuracy despite a decrease in iterations. The Paravartya Sutra algorithm is only described for decimal division in Ancient Mathematics literature. Consequently, a fundamental knowledge of decimal number division utilising the Paravartya Sutra is required. The Nikhilam method is inapplicable to problems with larger divisor numbers . In such situations, the Paravartya Yojayet method is a generalised algorithm that can encompass most situations. The transposal rule requires an unaltered sign change for each side change. Consequently, '+' becomes '-' and vice versa. The current method of calculation admits this law only for the swapping of terms from right to left and vice versa, as well as from denominator to numerator and vice versa.The widespread use of binary numerals in digital design necessitates the conversion from decimal to binary. Consequently, the same algorithm used for decimal numbers is used for binary numbers in this paper.

The Paravartya Sutra mandates the complementing of bits and the use of bits -1, 0 and 1 to signify the divisor and dividend.

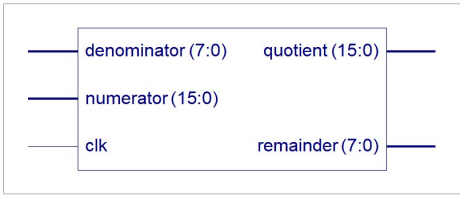

Fig 1. Entity of Paravartya Yojayet Division algorithm.

Divisor				Dividend								
1	0	1	1	1	0	0	1	0	0	1	1	
	0	-1	-1		0	-1	-1					
						0	0	0				
							0	1	1			
								0	0	0		
									0	-1	-1	
				1	0	-1	0	1	1	0	0	
				Q=10001-00100								
				Quotient=01101					Remainder= 100			

Fig 2. Paravartya Yojayet Division algorithm

PARAVARTYA YOJAYET'S ALGORITHM STEPS FOR CARRYING OUT VEDIC DIVISION:

Step 1: First, we have the dividend and divisor. Determine if the divisor is larger or less than the dividend. If the difference is greater, the result will be a positive difference; otherwise, the result will be a negative difference, which will serve as the new divisor for subsequent stages.

Step 2: Subtract the initial digit of the dividend. Afterwards, divide by the difference. Write the result on the next line of the dividend, relocating it slightly to the right.

Step 3: Add the second portion and record the result. Repeat the same procedure until the final digit of the dividend is reached.

Step 4: Obtain the remainder and quotient.

MODIFIED PARAVARTYA YOJAYET BASED VEDIC DIVIDER MODULE:

The performance of numerous division algorithms is compared when designing the divider block. In comparison, the modified Paravartya Yojayet based Vedic division algorithm is more efficient. The algorithm is based on the "transpose and apply" principle, wherein all divisors are inverted except for the most significant bit.

Multiplying these complemented divisor inputs yields additional results for each column of the dividend. First, the MSB dividend bit is partially multiplied with the divisor's complemented digits, followed by the addition of the multiplication results to the next dividend input. The quotient and remainder are obtained by adding columns successively. This algorithm concept is used to determine the utilisation of resources and the time required to perform division operations. The blocks are programmed with the VHDL programming language, synthesised with the Xilinx Integrated Synthesis tool, and simulated with the Modelsim simulator. The outcomes are compared to the multiplier's Lookup Table, Combinational Path delay, Number of paths, and Logical Level Utilisation. As a result, the Paravartya Yojayet-based Vedic division concept occupies less space, consumes less path delay, and employs less logic than the adder-based Vedic Multiplier. Consequently, the proposed Modified Paravartya Yojayet algorithm is used to improve the efficacy of the Arithmetic Logic Unit when designing a divider block for the Arithmetic Logical Unit.

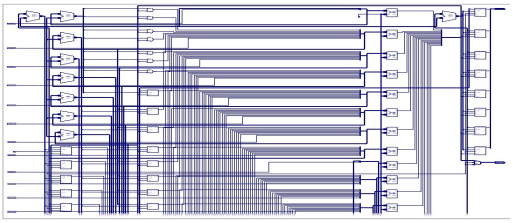

Fig 3. RTL view of 16 by 8 modified Paravartya Yojayet Vedic Divider

The division block is designed using various division algorithms, and the results are compared to determine that the modified Paravartya Yojayet-based Vedic division algorithm achieves the best performance in terms of logical utilisation, distribution of logical slices, total gate count consumption, and power consumption.

COMPARATIVE ANALYSIS REPORT OF VARIOUS DIVIDERS:

The result of non-restoring divider, restoring divider, procedure-based divider, modified Paravartya Yojayet based Vedic divider on two distinct targeted FPGA families includes Logical Utilization, Total Equivalent Gate Count, Combinational Path Delay, Logical Level and Total anticipated power consumptions, which are listed in tables 1 and 2 sequentially.

The Column chart report on LUT & Gate count of Spartan 2E are shown in fig. 4. The line chart report on Delay and Power consumption are shown in fig. 5. The Column chart report on LUT & Gate count of Virtex FPGA are shown in fig. 6. The line chart report on Delay and Power consumption are shown in fig. 7.

ISSN: 2278-4632
Vol-15, Issue-04, No.02, April: 2025

S.No	Divider Variants	Logical Utilization (LUT)	Total Equivalent Gate Count	Combinational Path Delay	Level of Logic	Additional Gate Count	Total Estimated Power Consumption P(mW)
1	Non- Restoring divider	173/2400	4569	46.27ns	18	2112	31
2	Restoring divider	117/2400	1762	31.59ns	35	6384	30
3	Nikhilam procedure based	532/2400	3467	88.105ns	138	2160	35
4	Modified Paravartya Yojayet Vedic Divider	71/2400	426	25.272ns	31	1200	20

Table 1 Comparative report of several dividers on 2s100etq144 Spartan2E FPGA

S.No	Divider Variants	Logical Utilization (LUT)	Total Equivalent Gate Count	Combinational Path Delay	Level of Logic	Additional Gate Count	Total Estimated Power Consumption P(mW)
1	Non- Restoring divider	171/10944	3981	36.27ns	10	2112	163
2	Restoring divider	117/10944	1760	30.45ns	35	6384	169
3	Nikhilam procedure based	390/10944	3467	40.980ns	138	2160	161
4	Modified Paravartya Yojayet Vedic Divider	71/10944	426	28.628ns	31	1200	105

Table 2 Comparative report of several divider

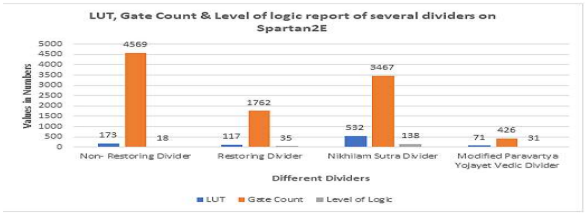

Fig 4. LUT, Gate Count & Level of logic report of several dividers on Spartan2E FPGA device

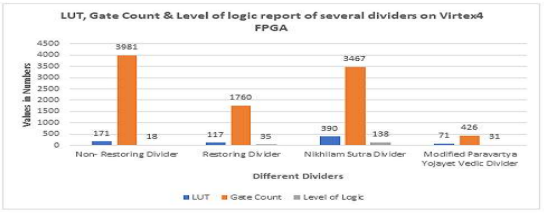

Fig 6. LUT, Gate Count & Level of logic report of several dividers on Virtex4 FPGA device

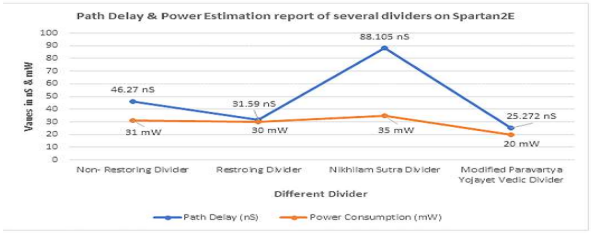

Fig 5. Path Delay & Power Estimation report of several dividers on Spartan2E FPGA device

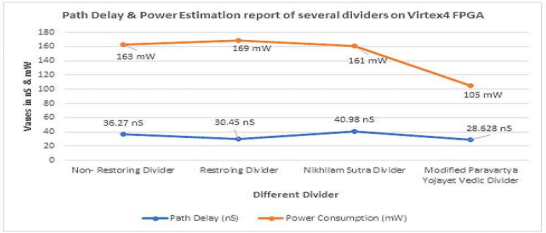

Fig 7. Path Delay & Power Estimation report several dividers on Virtex4 FPGA device

s

on 4vfx12sf363-12 Virtex4 FPG

CONCULSION:

In this work, a modified Paravartya Yojayet Vedic mathematical algorithm is implemented for division operation within the Arithmetic Logic Unit (ALU) to enhance processor performance. The proposed approach simplifies complex division steps, reduces computation time, and minimizes hardware usage compared to conventional methods. By leveraging the parallelism and simplicity of Vedic mathematics, the design achieves faster execution and improved efficiency. Simulation results demonstrate significant gains in speed and resource optimization. This advancement holds promise for high-speed computing applications, particularly in embedded and real-time systems, where performance and power efficiency are critical. Future work can explore integration with full Vedic ALU design.

REFERENCES:

1. Nitish Aggarwal, et.al, "An improvement in the restoring division algorithm (needy restoring division algorithm)", 2nd IEEE International Conference on Computer Science and Information Technology. IEEE, 2009.
2. Prabir Saha, et.al, "Vedic divider: Novel architecture (ASIC) for high speed VLSI applications", International Symposium on Electronic System Design. IEEE, 2011.
3. Alireza Khosropour, et.al, "Quantum division circuit based on restoring division algorithm", Eighth International Conference on Information Technology: New Generations. IEEE, 2011.
4. Jubin Hazra, et.al, "An efficient design technique of circular convolution circuit using Vedic Mathematics and McCMOS technique", International Conference on Computer Communication and Informatics. IEEE, 2012.
5. Diganta Sengupta, et.al, "An algorithm facilitating fast BCD division on low end processors using Ancient Indian Vedic Mathematics Sutras", International Conference on Communications, Devices and Intelligent Systems (CODIS). IEEE, 2012.
6. Kihwan Jun, et.al, "Modified non-restoring division algorithm with improved delay profile and error correction", Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR). IEEE, 2012.
7. Wei Liu, et.al, "Power efficient division and square root unit", IEEE Transactions on Computers 61.8 (2012): 1059-1070.
8. Soma BhanuTej, et.l, "Vedic divider-A high performance computing algorithm for VLSI applications", International conference on Circuits, Controls and Communications (CCUBE). IEEE, 2013.
9. Sukhmeet Kaur, et.al, "Vhdl implementation of non - restoring division algorithm using high speed adder/Subtractor", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering 2.7 (2013): 3317-3324.
10. Ratiranjan Senapati, et.al, "Novel binary divider architecture for high speed VLSI applications", IEEE Conference on Information & Communication Technologies. IEEE, 2013.
11. Kihwan Jun, et.al, "Improved non-restoring division algorithm with dual path calculation", IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE, 2013.
12. Surabhi Jain, et.al, "Binary division algorithm and high speed deconvolution algorithm (Based on Ancient Indian Vedic Mathematics)", 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). IEEE, 2014.
13. Lohita S. Niwal, et.al, "Design of radix 4 divider circuit using SRT algorithm", 2015 International Conference on Communications and Signal Processing (ICCSP). IEEE, 2015.
14. Saba Akram, et.al, "Newton raphson method", International Journal of Scientific & Engineering Research 6.7 (2015): 1748-1752.

15. Peter Malik, et.al, “High throughput floating-point dividers implemented in FPGA”, IEEE 18th International Symposium on Design and Diagnostics of Electronic Circuits & Systems. IEEE, 2015.
16. Bhavana Mehta, “High speed SRT divider for intelligent embedded system”, International Conference on Soft Computing and its Engineering Applications (icSoftComp). IEEE, 2017.
17. Aakanksha Singh, et.al, “Vedic Maths Calculator using VLSI Implementation”, International Journal of research in Engineering, Science and Management”, Vol.2, Issue -5, 2019.
18. Modern Design and Testing of High-Speed Vedic ALU Controller Using Vedic Algorithms 2023.

