INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

*Corresponding author.

prasad7research@gmail.com

Pipelined and Wave Pipelined Approach Based Comparative Analysis for 16x16 Vedic Multiplier

J Prasad[1]*, M Vasim Babu[2], M Kasiselvanathan[3], K B Gurumoorthy[4]

1 Assistant Professor (Sl.G), ECE, KPR Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India
2 Assistant Professor, Electronics Engineering, Madras Institute of Technology Campus, Chennai, Tamil Nadu, India
3 Assistant Professor (Sl.G), ECE, Sri Ramakrishna Engineering College, Coimbatore, Tamil Nadu, India
4 Associate Professor, ECE, KPR Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India

Abstract

Objectives: This work objective is to construct an FPGA-based 16x16 Vedic multiplier and assess the performance of the multiplier using three distinct architectures: pipeline, wave pipeline, and modified wave pipeline in terms of delay and clock skew. **Methods:** The 16×16 Vedic multiplier was constructed and designed through four numbers of an 8x8 Vedic multiplier. For the 16x16 Vedic multiplier, the 3-stage pipeline and wave pipeline techniques are applied, and the delay performances of the Vedic multiplier are compared. Delay optimization: In the wave pipeline Vedic multiplier architecture, the delay is decreased by inserting the known delay on the longest path delay between the multiplier and adder. Clock skew optimization: The clock skew issue of the wave pipeline Vedic multiplier architecture is minimized by adjusting the setup time violation of the clock signal that is connected to the input and output registers. **Findings:** The delay performance of the Vedic multiplier was evaluated by the synthesis tools Xilinx 12.1, Xilinx ISE 14.2, and Altera, and based on the synthesis report, the Xilinx synthesis tool offers 73.71% delay performance for the pipeline approach and 53.39% for the wave pipeline approach compared to the Altera tool. Further delay is reduced by the proposed modified wave pipeline approach, which saves 2.122 ns of delay compared to the wave pipeline architecture. The clock skew performance was analyzed using the Time Quest timing analyzer tool, and it was minimized to 0.035 from 0.048 compared to the wave pipeline approach. **Novelty:** In this work, the modified wave pipeline approach has been applied to the existing Vedic multiplier architecture, and it offers less delay as well as less clock skew compared to the existing method. Hence, the performance of the Vedic multiplier with a modified wave pipelined approach was evaluated through a 3-tap FIR filter by applying a vibroarthrography signal.

1 Introduction

The Vedic multiplier is a specific multiplier utilized in rapidly developing technology, particularly in the vast fields of image processing, DSP (digital signal processing), and real- time applications due to its efficiency in computation time, area, latency, and speed. Recently, many approaches have been used to improve the efficiency of a Vedic multiplier in terms of less computation time, high speed, and minimum area. Using a 32-bit full adder and a 42-bit full adder to create a Vedic multiplier is one method[1]. In addition to consuming the maximum area, this type of technique causes delays as the multiplier and multiplicand bit sizes increase. Another method for developing a Vedic multiplier is a 4:2 compressor that utilizes multiplexers and exclusive OR logic. A significant route delay results from the cascading of two full adders with four EX-OR gates in this approach[2]. In any type of multiplier, an adder is the primary building element. The multiplier's total efficiency will be defined by the adder's performance. Ripple carry adders are a special type of adder used to design a Vedic multiplier[3]. The ripple carry adder does not permit simultaneous usage of all full adders. Every full adder has to wait for the carry bit from the adjacent full adder to become available. Propagation takes longer as a result. This causes the ripple carry adder to become extremely slow and leads to maximum computation time. Implementation of a Vedic multiplier using a low power 13T hybrid full adder is one of the techniques[4]. Although the performance of the hybrid logic types is promising, the majority of these adders have poor driving capabilities, and if the appropriately designed buffers are not included, their performance significantly deteriorates in the cascaded mode of operation. To enhance the performance of the Vedic multiplier, the encoding technique of Radix-2 and Radix-4 encoding is the best method[5]. The primary drawback of the Radix method is that it requires complex multiplication and complex addition. As the number of inputs increases, this leads to increased computation time. Choosing an adder is crucial in multiplier design since adder design consumes more space and consumes maximum path delay in the architecture, which is related to the multiplication process. Vedic multiplier design using basic adders, half adders, and full adders offered the longest path delay as well as more space[6]. Carry save adder, ripple carry adder, and carry skip adder are utilized for FPGA-based multiplier implementation[7]. These adders slow down the multiplication operation when the width of the inputs increases.

In this work, we presented a carry-look-ahead adder for the Vedic multiplication process, evaluated the multiplier performance with three different architecture pipelines, wave pipelines, and modified wave pipeline techniques, and investigated Vedic multipliers using these methods based on the Altera tools, Xilinx 12.1, and Xilinx ISE 14.2 synthesis reports. In terms of area and delay, the Vedic multiplier performance was improved by the modified wave pipeline approach.

2 Motivation

Most important DSP (Digital Signal Processing) algorithms, such as digital finite and infinite impulse response filters, DFT, FFT, convolution, and other methods, use multiply-accumulate computations[8]. For any DSP method, the number of multiplications is the primary factor that defines the entire processing time because the multiplication time is typically considerably longer than the addition time; hence, multiplier design and implementation are more concentrated in DSP techniques[9]. Applications for digital filtering use a variety of multipliers. Due to its performance in terms of calculation time, area, latency, and speed, the Vedic multiplier is a specific

multiplier utilized in rapidly developing technology, particularly in the enormous arenas of image processing, DSP, and real-time applications.

Vedic mathematics is the foundation of the Vedic multiplier[8]. The 16 sutras of Urdhva Tiryagbhyam provide the foundation of Vedic mathematics. It denotes that multiplication is performed "vertically and crosswise" to use the fewest possible calculations, take up the least amount of space, and speed up computing[3]. This work focused on designing a Vedic multiplier with a size of 16×16 in a wave-pipelined approach, and its performance was improved through a modified wave-pipelined architecture in terms of delay and clock skew minimization.

3 Methodalogy

The main objective of this work is to apply three different approaches to improving the area and delay performance of the Vedic multiplier. The above goal is accomplished by the methods shown in Figure 1.
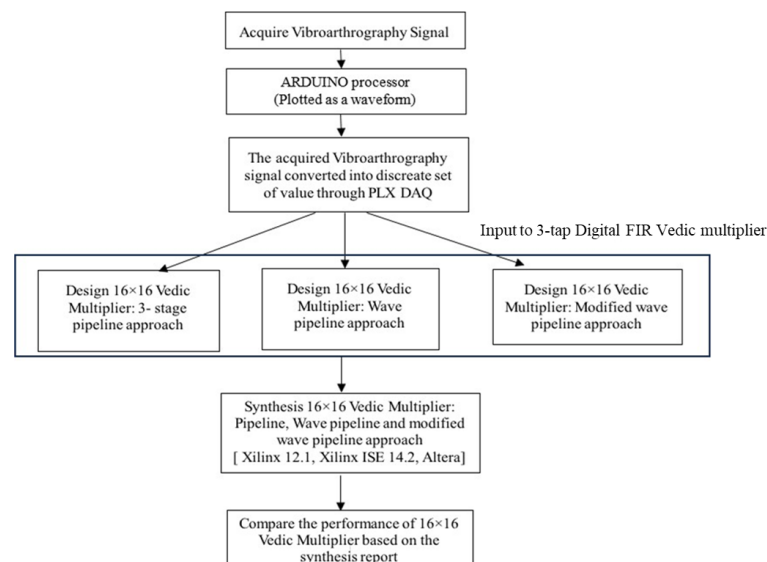

Fig 1. Methodology: Vedic Multiplier design

3.1 Experimental setup for Vibroarthrography signal acquisition

Vibroarthrography (VAG) means joining vibrations." It is the process of recording knee vibration sounds due to flexion[10]. The most often injured joint in the human body is the knee. There is a wide range of variation in the types of causes of knee joint injuries, which can be broadly categorized into three subgroups: autoimmune illnesses, cartilage deterioration caused over time, and trauma-related injuries. The VAG is the vibration signal that is recorded from a joint during movement of the joints. Normal joint surfaces are smooth and produce little or no sound, whereas joints affected by arthritis and other degenerative diseases, which may have suffered cartilage loss, produce a grinding sound. Detection of knee joint problems via analysis of VAG signals could help avoid unnecessary exploratory surgery and aid in the better selection of patients who may benefit from surgery. A normal VAG signal has a frequency range of 10–120 Hz; otherwise, it is called abnormal, and the person has knee joint diseases. The characteristics of Vibroarthrography signals are listed below:

- Vibroarthrography signals are non-stationary because they may vary from one angular position to another articulation joint.
- The generated VAG signal from leg movement contains background noise and random noise based on the sensor placed at the knee.

- Identification of amplitude and frequency range from the measured VAG signal indicates whether the knee is normal or abnormal.

The experimental setup for vibroarthrography signal acquisition is shown in Figure 2. The vibroarthrography signal is acquired by an accelerometer sensor, an Arduino board (plotted as a waveform), and PLX-DAQ (a discrete set of values), as shown in Figure 2. An accelerometer is one of the most common inertial sensors, a dynamic sensor capable of a vast range of sensing. Accelerometers are available that can measure acceleration on one, two, or three orthogonal axes. The output of the Arduino block is shown in Figure 3, where more vibrations occur during extension and flexion, and this waveform is converted to discrete values through the PLX-DAQ unit. The output of PLA-DAQ unit is connected to the input of 3-tap FIR filter. The performance of the Vedic multiplier is evaluated by integrating it into the FIR filter with three different approaches: the wave pipeline and the modified wave pipeline.
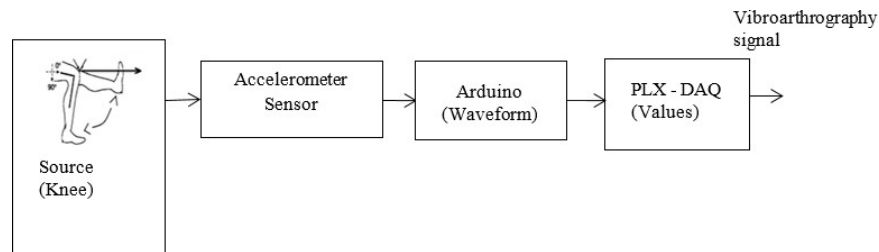

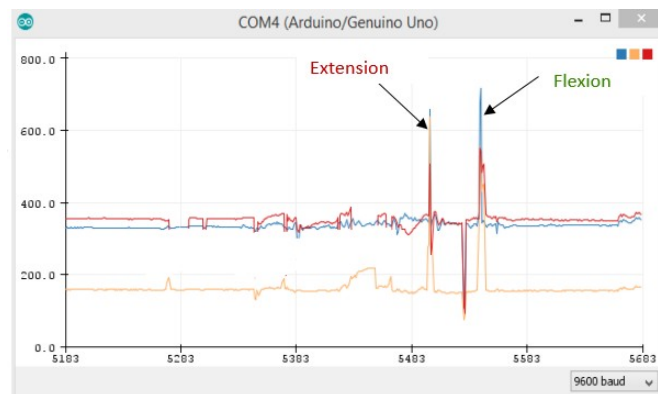**Fig 2. Experimental setup for Vibroarthrography signal acquisition**

Fig 3. Vibroarthrography signal acquired at the output of Arduino block

3.2 Wave Pipeline Vedic Multiplier

In VLSI design, pipeline architectures are required to improve throughput. Pipelining is a very popular concept and provides parallelism. Because of this, it will increase the computation rate or throughput of the system, but the area of the overall hardware module is linearly increased when the number of pipelined stages is increased [11]. In addition, setup time, hold time, and clock skew also increased since all the registers were clocked by the common clock signal. Therefore, the wave pipelining technique can be an alternate solution for reducing the area, delay, and clock skew. To achieve this parameter, in wave pipelining and modified wave pipeline approaches, the intermediate latches are removed between input and output registers [12]. The Vedic multiplier with a 16×16 wave-pipeline architecture is shown in Figure 4, where the 16×16 Vedic multiplier is developed by four 8×8 Vedic multipliers. Table 1 is the comparison performance of the Vedic multiplier with a size of 16×16 by pipeline and wave pipeline approaches.

3.3 Proposed architecture

Reducing the wave-pipeline Vedic multiplier's clock and delay is the goal of the "modified wave-pipeline" architecture that is being suggested. Figure 5 illustrates the updated wave pipeline technique architecture for the 16×16 Vedic multiplier. To balance
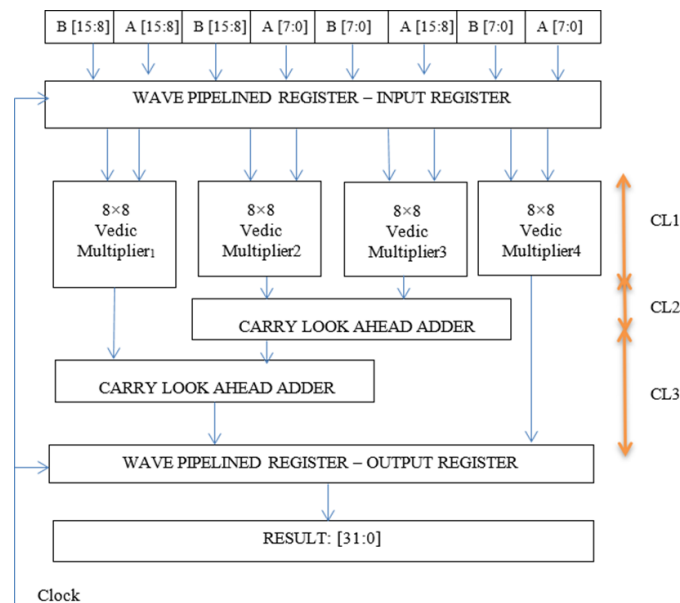
Fig 4. Vedic multiplier architecture with size 16 × 16 using Wave pipelined approach

or shorten the delay of the Vedic multiplier architecture, a unique register called the "wave pipeline register (known delay) is introduced between the multiplier and adder unit, as shown in Figure 5.
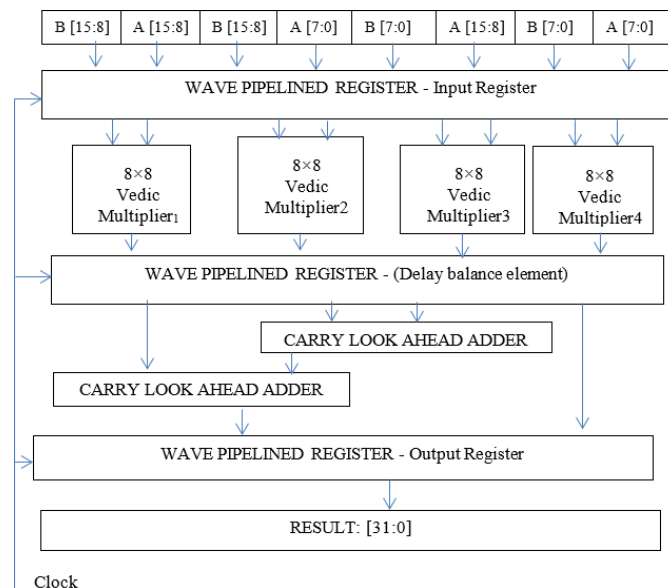

Fig 5. Vedic multiplier architecture with size 16 × 16 using modified wave pipelined approach

3.3.1 Delay Minimization

Introducing the known discrete delay in the hardware module is one of the best approaches to balancing the path between the combinational logic, which reduces the delay of the overall logic module. The overall delay gets varied based on inserting the known delay between the two logics in the overall logic module. The known delay element (wave pipeline register) is placed between the multiplier and adder unit because, according to the synthesis report given by Xilinx 12.1, the worst path delay occurs between the 8×8 Vedic multiplier and the ripple carry adder.

3.3.2 Clock skew Minimization

Clock skew is, in the simplest terms, the interval of time between the arrival of a clock signal's identical edge at the clock pin of the capture flop and launch flop. Throughout all phases of chip design, clock signals play a critical role in the design of VLSI-based synchronous logic circuits [13]. Verifications like static timing analysis and gate-level simulations that are all part of the simulation depend on these clock signals, simply because the state transition happens at a clock transition. The manner in which digital systems function is significantly influenced by their clocking techniques [14]. The fundamental block architecture for the wave pipeline gadgets, wherein the source and destination registers are synchronized by a common clock signal and intermediate latches are removed [9]. The logic depth-timing diagram in Figure 6 illustrates this fundamental idea of wave pipelining, in which the clock frequency is lowered as much as feasible when region R1 rises [15]. The discrepancy between the tmin and tmax data delays over the combinational chunk, propagating times, register setup time and hold time, and unintentional clock skew are the issues currently limiting the clock period.
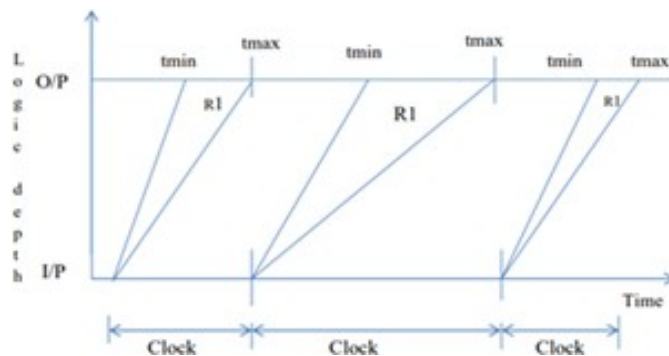

Fig 6. Temporal and spatial diagram of Wave pipelined system

The following variables affect the clock period of the wave-pipelined system [13]:

1. The longest and smallest path delays, respectively.
2. The difference in delay between T_{min} and T_{max}.
3. Setup time (T_{setup}).
4. Hold time (T_{hold}).
5. Clock skew (T_{skew}).

The overall clock signal of a wave-pipelined clock signal depends on the sum of setup time, hold time, skew time, and the difference in delay between T_{min} and T_{max}. The clock period of the wave pipeline technique is defined in Equation (1).

$$(T_{max} - T_{min}) < T_{clk} + (T_{setup} + (2T_{skew}) + T_{hold})) \tag{1}$$

It is clear from Equation (1) that the clock period of a wave-pipelined system's is improved by the delay variance among the T_{max} and T_{min} paths, and that clock overhead consists of setup, hold, and skew times. In this work, the clock skew performance is focused. The clock skew is minimized by adjusting the setup time violation of the clock signal [16]. The clock skew has been developed since the setup time and hold time violations of the clock signal.

The clock skew has been developed since the setup time and hold time violations of the clock signal [17]. The wave pipeline will eliminate the hold time, which is a more serious issue. In this work, setup time violations are concentrated upon. The setup time is the minimum time essential to sample the input, so the input should be steady at that time. Hold time is described as the least quantity of period in time, and the information must be constant subsequently the active edge clock.

In this work, the clock skew performance of the Vedic Multiplier with size 16×16 using Wave Pipelined and Modified Wave Pipelined architectures is analyzed through the Altera Quartus-II Time Quest Timing Analyzer FPGA tool. The Modified Wave Pipelined 16×16 Vedic multiplier architecture gives better performance in terms of clock skew. This can be achieved by adjusting the setup time violation of the clock signal.

4 Results and Discussion

4.1 Comparison between Pipeline, Wave Pipeline and Modified Wave pipeline approach for 16×16 Vedic multiplier architecture

Table 1. Comparison Table: Vedic multiplier: Pipeline Vs Wave Pipeline Vs Modified Wave pipeline approach

Comparison parameters	Pipeline approach			Wave Pipeline approach			Modified Wave Pipeline approach		
	Xilinx 12.1	Xilinx ISE 14.2	Altera Quartus II	Xilinx 12.1	Xilinx ISE 14.2	Altera Quartus II	Xilinx 12.1	Xilinx ISE 14.2	Altera Quartus II
Area (Slice LUT)	188	348	467	156	293	393	146	246	380
Area (Slice registers/Flip flop) utilized	1%	1%	2%	1%	1%	3%	3%	2%	2%
IOB utilized	9%	1%	21%	10%	20%	24%	26%	20%	25%
Delay in ns	1.918	2.102	7.298	4.432	3.818	8.193	3.103	3.403	6.183
Frequency in MHz	5.2	4.7	1.37	2.25	2.61	1.22	3.22	2.93	1.61

Target device for Xilinx: Virtex – 4
Target device Altea: Stratix – II

Based on the synthesis report generated from the synthesis tools Xilinx 12.1, Xilinx ISE 14.2, and Altera Quartus-II with target devices Vitex-4 for Xilinx and Stratix-II for Altera, respectively, Table 1 indicates the comparison results for the implementation of Vedic multipliers through different architecture pipelined, wave pipelined, and modified wave pipelined approaches. When comparing the three synthesis tools' performances, Altera's performance was worse compared to that of the Xilinx synthesis tools for both pipelined and wave-pipelined techniques. All three synthesis tools offered the greatest delay for the Vedic multiplier computation but occupied less area as compared to the pipeline approach over the wave pipeline approach. Hence, a novel method called modified wave pipelined is used with the Vedic multiplier to enhance its performance in terms of area and delay. The modified wave-pipelined approach outperformed the wave-pipelined approach in terms of area and delay, according to the Xilinx Synthesis Tool report. Based on the comparison, Table 1 concludes that the modified wave pipeline strategy performs better than the wave pipeline approach in terms of area and delay.

4.2 Delay Minimization

Table 2 summarizes the comparison between the existing method over the proposed method of wave pipeline and the modified wave pipeline Vedic multiplier performance in terms of delay and area. It can be concluded that the proposed modified wave pipeline approach for the Vedic multiplier gives improved delay performance as well as area compared to the existing method.

Table 2. Performance comparison of existing method vs Proposed method

Existing Method	Parameters	
	Delay in ns	Area Slice LUT utilization
Akshay Savji et. al [1]	54.037	6%
Swati Shetkar et.al [2]	43.65	4%
Nuriddin Safoev et.al [3]	4.25	5%
Mansi Jhamb et.al [4]	13.921	3%
C. M. Kalaiselvi et.al [5]	15.83	4%
VijendraBairwa et.al [6]	11.198	4%
Proposed Method		
Wave Pipelined Vedic Multiplier	4.432	3%

Continued on next page

Table 2 continued		
Modified Wave Pipelined Vedic Multiplier	3.403	3%

4.3 Clock Minimization: Wave Pipeline approach for 16×16 Vedic Multiplier

The detailed timing report of the Vedic Multiplier with a size of 16×16 using the Wave Pipelined technique is shown in Figure 7. The timing analysis report was generated from the Altera Quartus Time Quest timing analyzer tool[17]. The first two waveforms, launching clock and latch clock, indicate the clock signal at the input wave pipelined register (source register) and the output wave pipelined register (destination register), respectively. A dark line indicates the clock edges of the launch and latch. The time it will take for data to go from the input wave pipelined register to the output wave pipelined register is displayed in the next two waveforms, called information arrival and data delay[17]. Observe that the input wave pipeline register's positive edge of the Launch Clock is where the data delay is monitored. The time at which the data should have reached its destination—that is, the Wave Pipeline output register to be accurately saved, including the setup time (uTsu) displayed in the final line-is indicated by the second-to-last waveform, which is referred to as "data required."
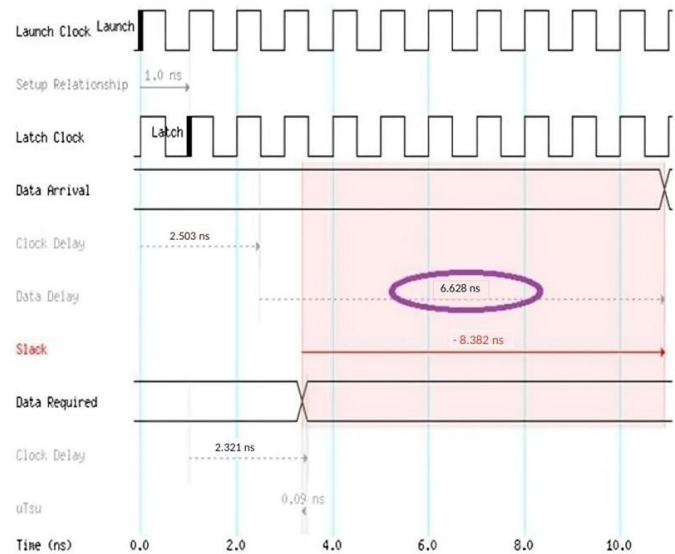

Fig 7. Altera Quartus II-time quest timing analyzer timing report for 16×16 Wave Pipeline Vedic Multiplier

According to the Altera Quartus II-Time Quest timing analyzer report shown in Figure 7, the Vedic Multiplier architecture with a size of 16×16 using the Wave Pipelined technique has a clock skew of 0.048 ns (between the wave pipeline source (input) register and the destination (output) register) and a data pathway delay from the source wave pipeline register to the output register of around 6.628 ns. The clock skew performance of wave-pipelined architecture for Vedic multipliers improved by modified architecture.

4.4 Clock Minimization: Modified Wave Pipeline approach for 16×16 Vedic Multiplier

The detailed timing report for modified wave pipelined 16×16 Vedic Multiplier obtained through the static timing analyzer tool is shown in Figure 8. Here the slack value, i.e., setup time violation, is adjusted to -5.128 ns from -8.832 ns. According to the timing analyzer report revealed in Figure 9, the modified Vedic multiplier with a 16×16 architecture has a clock skew of 0.035 (between the wave pipeline input register and output register), and the data path delay from the source wave pipeline register to the destination register is around 4.506 ns.

From Figures 7 and 8, it is noted that the clock frequency for both wave-pipelined and modified wave-pipelined 16×16 Vedic multipliers remains the same, i.e., 1 ns, where the setup time violation is adjusted in the modified wave-pipelined 16×16 Vedic multiplier clock to -5.128 ns and -8.382 ns for the wave-pipelined 16×16 Vedic multiplier. Due to adjusting or reducing the setup violation time of the clock signal in the modified wave pipelined architecture, the clock skew was reduced to 0.035 from 0.048 ns compared to the wave pipelined 16×16 Vedic Multiplier. Also, the modified wave pipelined requires less delay, i.e., the delay compensation has reduced the setup time violation by about 2.122 ns (6.628–4.506) for the Vedic multiplier with size 16×16
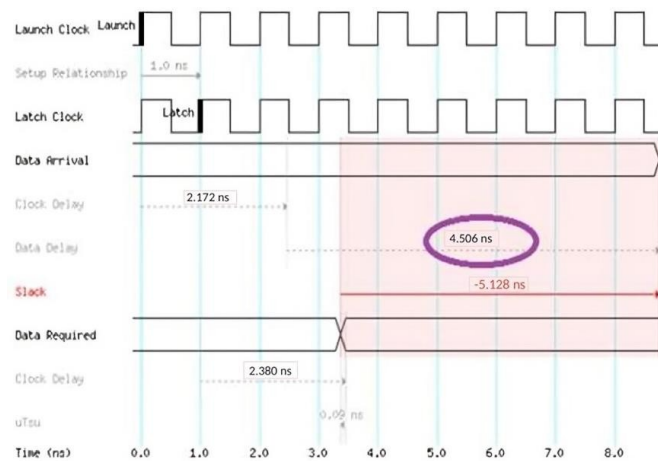
Fig 8. Altera Quartus II-time quest timing analyzer timing report for 16×16 modified Wave Pipeline Vedic Multiplier

modified wave pipelined from the wave pipelined architecture. Hence, the modified wave-pipelined 16×16 Vedic multiplier provides better performance in terms of clock skew and delay compared to the wave-pipelined approach. Xilinx and Altera FPGA software are used to implement digital circuits for image signal processing and digital signal processing applications. Xilinx lags in terms of design tools but performs marginally better. Altera offers significantly less performance in exchange for having superior design tools. Additionally, Altera offers specific timing analyzer tools. Hence, for delay optimization, we have chosen Xilinx tools, and for timing analysis (clock skew) performance, we have used the Altera Time Quest timing analyzer synthesis tool.

4.5 Delay and clock skew comparison

The delay and clock skew comparison between wave pipeline and modified wave pipeline approach for 16×16 Vedic multiplier implementation through Altera Quartus-II tool is shown in Figure 9.
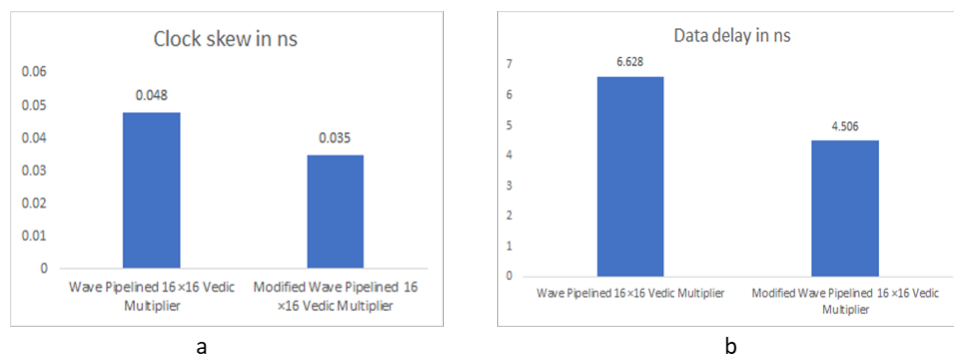

Fig 9. a) Clock skew comparison, b) data delay comparison

5 Conclusion

In this work, the comparative analysis of a Vedic multiplier with a size of 16×16 using a pipelined, wave-pipelined, and modified wave-pipelined approach is discussed with a vibroarthrography signal as input. The performance is evaluated by three synthesis tools: Xilinx 12.1, Xilinx ISE 14.2, and Altera Quartus. The synthesis report generated from the Xilinx synthesis tool shows that the wave pipeline approach for Vedic multipliers consumed more delay compared to the three-stage pipeline approach. To resolve this problem, a new approach called modified wave pipelines was introduced on the Vedic multiplier architecture with the addition of a known delay between the multiplier and adder unit. According to the synthesis report, the modified wave

pipeline approach saves 1.029 ns in delay compared to the wave pipeline approach. The clock skew problem with respect to setup time violation was analyzed through the Altera time quest analyzer tool, where only modified wave pipelined architecture has a clock skew of 0.035 ns compared to wave pipelined architecture 0.048 ns, i.e., modified wave pipelined provide 32.01% improved delay performance and 27.08% clock skew performance for Vedic multiplier computation.

Hence, the modified wave pipelined approach offers better delay and clock skew performance for the Vedic multiplier. As the modified wave-pipelined approach offered better results for the Vedic multiplier computation, further performance was assessed through a 3-tap digital FIR filter to extract the information presented in the input vibroarthrography signal. The vibroarthrography signal is acquired from the 31-year-old male candidate and connected to the input for a 3-tap digital FIR filter. Based on the frequency response of the FIR filter, the acquired Vibroarthrography signal is normal, and the person has no osteoarthritis diseases.

References

1) Savji A, Oza S. Design and Implementation of Vedic Multiplier. *International Journal of Recent Technology and Engineering (IJRTE)*. 2020;8(6):3142–3145. Available from: https://www.ijrte.org/wp-content/uploads/papers/v8i6/F8808038620.pdf.

2) Shetkar S, Koli S. Area, power efficient Vedic multiplier architecture using novel 4:2 compressor. *Sādhanā*. 2023;48(4):1–7. Available from: https://doi.org/10.1007/s12046-023-02274-1.

3) Safoev N, Jeon JC. Design and Evaluation of Cell Interaction Based Vedic Multiplier Using Quantum-Dot Cellular Automata. *Electronics*. 2020;9(6):1–15. Available from: https://doi.org/10.3390/electronics9061036.

4) Jhamb M, Kumar M. Optimized vedic multiplier using low power 13T hybrid full adder. *Journal of Information and Optimization Sciences*. 2023;44(4):675–687. Available from: https://doi.org/10.47974/JIOS-1222.

5) Kalaiselvi CM, Sabeenian RS. A modular technique of Booth encoding and Vedic multiplier for low-area and high-speed applications. *Scientific Reports*. 2023;13(1):1–21. Available from: https://doi.org/10.1038/s41598-023-49913-5.

6) Bairwa V, Jindal P. Analysis of 16x16 Vedic and 16 Bit Floating Point Multiplier -A Comparative Study. In: Proceedings of the 3rd International Conference on Contents, Computing & Communication (ICCCC-2022);vol. 2022. Elsevier BV. ;p. 1–8. Available from: https://doi.org/10.2139/ssrn.4043575.

7) Khubnani R, Sharma T, Subramanyam C. Applications of Vedic multiplier - A Review. In: Virtual National Conference on Materials for Electronics Applications 2021 ;vol. 2225 of Journal of Physics: Conference Series. IOP Publishing. 2022;p. 1–10. Available from: https://iopscience.iop.org/article/10.1088/1742-6596/2225/1/012003/pdf.

8) Immareddy S, Sundaramoorthy A. A survey paper on design and implementation of multipliers for digital system applications. *Artificial Intelligence Review*. 2022;55(6):4575–4603. Available from: https://doi.org/10.1007/s10462-021-10113-0.

9) Prasad J, Geetha DM, Srinivasan K. Experimental setup of stretchable arid dry pad sensors for the signal acquisition fir filter design using Vedic approach. *Measurement*. 2019;141:209–216. Available from: https://doi.org/10.1016/j.measurement.2019.02.083.

10) Łysiak A, Froń A, Bączkowicz D, Szmajda M. Vibroarthrographic Signal Spectral Features in 5-Class Knee Joint Classification. *Sensors* . 2020;20(17):1–15. Available from: https://doi.org/10.3390/s20175015.

11) Algredo-Badillo I, Ramírez-Gutiérrez KA, Morales-Rosales LA, Bautista DP, Feregrino-Uribe C. Hybrid Pipeline Hardware Architecture Based on Error Detection and Correction for AES. *Sensors* . 2021;21(16):1–26. Available from: https://doi.org/10.3390/s21165655.

12) Christilda VD, Milton A. Speed, power and area efficient 2D fir digital filter using Vedic multiplier with predictor and reusable logic. *Analog Integrated Circuits and Signal Processing*. 2021;108:323–333. Available from: https://doi.org/10.1007/s10470-021-01853-8.

13) Krylov G, Friedman EG. Wave Pipelining in DSFQ Circuits. *IEEE Transactions on Applied Superconductivity*. 2022;32(8). Available from: https://doi.org/10.1109/TASC.2021.3135956.

14) Gorawski M, Grochla K, Marjasz R, Frankiewicz A. Energy Minimization Algorithm for Estimation of Clock Skew and Reception Window Selection in Wireless Networks. *Sensors*. 2021;21(5):1–17. Available from: https://doi.org/10.3390/s21051768.

15) Zhou J, Xie G, Yu S, Li R. Clock-Based Sender Identification and Attack Detection for Automotive CAN Network. *IEEE Access*. 2020;9:2665–2679. Available from: https://doi.org/10.1109/ACCESS.2020.3046862.

16) Najafi A, Najafi A, Garcia-Ortiz A. Stochastic Wave-Pipelined On-Chip Interconnect. *IEEE Transactions on Circuits and Systems II: Express Briefs*. 2020;67(5):841–845. Available from: https://doi.org/10.1109/TCSII.2020.2984194.

17) Quartus -II handbook version 9.0;vol. 3. Altera corporation. 2019. Available from: http://ridl.cfd.rit.edu/products/manuals/Altera/User%20Guides%20and%20AppNotes/External%20Memory/qts_qii53018.pdf.