e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:08/August-2023 Impact Factor- 7.868 www.irjmets.com

LEAST MEAN SQUARES LOW POWER DISTRIBUTED ARITHMETIC ADAPTIVE FILTER DESIGN

Komal Rai*1, Chandrahas Sahu*2

*1Master Of Technology (M.Tech) Research Scholar Electronics And Telecommunications (VLSI Design) Shri Shankaracharya Technical Campus Bhilai, India.

*2Asst. Professor, Electronics And Telecommunication (VLSI Design), Shri Shankarcharya Technical Campus Bhilai, India.

ABSTRACT

The essential objective of each and every channel is to remove important data from a boisterous sign. A decent typical channel is made utilizing the measurements of two signs: data and commotion. A versatile channel naturally acclimates to changes in the climate utilizing recursive calculations and is utilized when the sign's measurements and fleeting changes are obscure. A versatile discrete channel gets an info x (n) and yields y(n) because of convolution with the channel weight w(k). At the result, a reference wanted signal d (n) is contrasted with get a mistake gauge e (n). This blunder signal is used to progressively adjust the channel weight for the accompanying time stretch. A few calculations are utilized. The LMS (Least Mean Square) and the RLS (Recursive Least Square) are two techniques for changing the weight. The LMS (least mean squares) procedure is presented in this task for the structure of versatile crossing channels utilizing a recently proposed high velocity Vedic multiplier. The LMS versatile channel is written in Verilog HDL, the plan is combined with Xilinx XST, and recreation is performed with Modelsim 6.5. This undertaking's application side contains distinguishing an obscure framework in different correspondence and computerized signal handling applications like Channel Leveling, Versatile Commotion Wiping out, and Versatile Reverberation Undoing.

Keywords: Design Of Vedic Multiplier, FIR Filter, Adaptive Traversal Filters, LMS (Least Mean Squares) Algorithm VLSI.

I. INTRODUCTION

There are different duplication calculations; one, the Egyptian increase, comes to us from times long past, and the Vedic strategy is probably going to be something very similar. Another technique, known as cross section augmentation, was acquainted with Europe in the mid-1200s, while the fourth, known as Russian Worker duplication, was probably imagined significantly later and first showed up in exceptionally contemporary times. An exceptional increase model purportedly planned by a Chinese teacher that has been getting out and about on the Web is reasonable not more established than the actual Web. Except for the Vedic assortment, none cases divine beginning. The essential objective of this study is to create and develop a quick multiplier in light of Vedic math that might be used in any processor application. This work looks at the investigation, plan, and execution of Vedic multipliers going from 44 to 64x64 bits. The subject of Vedic duplication has likewise been examined in this paper exertion. What's more, Vedic multiplier engineering with less doors and a fast determination is worked here. This multiplier's blend and execution were finished utilizing Xilinx XST, and reproduction was finished with ModelSim 6.5s Device.

1.1 URDHVA TIRYAGBHYAM

The "Urdhva Tiryagbhyam Sutra" is a nonexclusive increase equation that might be utilized to any duplication issue. The expressions "Urdhva" and "Tiryagbhyam" are acquired from Sanskrit writing. " Urdhva" signifies "in an upward direction", while "Tiryagbhyam" signifies "transversely". The multiplier developed the LMS strategy, which depends on a Urdhva Tiryakbhyam (Vertical and Across) calculation of old Indian Vedic Arithmetic. The Urdhva Tiryakbhyam Sutra is a nonexclusive increase recipe that might be utilized for a duplication. " In an upward direction and across" is the exacting interpretation. In view of a progressive idea permits the development of all fractional items while simultaneously adding these halfway items. Urdhava Triyakbhyam is utilized to accomplish parallelism in the making of halfway items and their summation. The Vedic Multiplier depends on a book Urdhava. The Triyakbhyam philosophy of computerized duplication is particular from the customary techniques for augmentation like add and shift. Where more modest blocks are used to make a bigger one. The Vedic Multiplier is made in Verilog HDL since it takes full advantage of the underlying procedure of demonstrating. The Verilog equipment depiction language is utilized to execute every individual block. ModelSim and ISE reenactment instruments are utilized to approve the working of each block.

1.2 Verilog Hdl

Outline of Verilog and its set of experiences, as well as a portion of its elements and advantages. Verilog is for sure an equipment depiction language (HDL) that is broadly utilized in the field of electronic plan for displaying and portraying computerized frameworks. It's utilized for different purposes, including plan, confirmation, and execution of electronic circuits. Here is a rundown of the central issues you've referenced. Beginning and Advancement: Verilog was initially evolved by Passage Plan Mechanization in 1984. Afterward, Rhythm Plan Frameworks gained Door and kept creating Verilog. It acquired prominence as an equipment portrayal language for demonstrating computerized frameworks.

Standardization: The specifications for Verilog-HDL were released by Cadence and were accepted as an IEEE standard (IEEE 1364) in 1995. The standardization included support for the Programming Language Interface (PLI) version 1.0 and later PLI 2.0 (VPI).

Verilog 2001: In 2001, the IEEE updated the Verilog standard, often referred to as Verilog 2001. This version introduced enhancements to the language and its capabilities.

Syntax and Abstraction Levels: Verilog's sentence structure is frequently contrasted with the C programming language, making it moderately simple for those with C programming experience to learn. Verilog permits various degrees of reflection to be blended inside similar model, empowering originators to portray circuits at different levels, from doors to undeniable level social code.

Abstraction Flexibility: Designers can model hardware using different abstraction levels such as switches, gates, Register Transfer Level (RTL), or behavioral descriptions. This flexibility allows designers to choose the appropriate level of abstraction for their specific needs.

Conversion from VHDL: You mentioned converting an IOP (Input-Output Processor) VHDL design to Verilog. This could be for various reasons, such as compatibility with specific tools or platforms that prefer Verilog, or the familiarity of the design team with Verilog.

Comparison with VHDL: You briefly compared Verilog with VHDL. VHDL is another widely used HDL for describing hardware systems. While VHDL offers more programming constructs and supports 9-value logic, Verilog is often noted for its C-like programming style and closeness to hardware representation.

Synthesis and Tool Support: Verilog is well-supported by logic synthesis tools, which convert high-level descriptions into gate-level representations suitable for hardware implementation. Overall, your description provides a comprehensive overview of Verilog, its history, features, and advantages, as well as its role in the field of electronic design. It's worth noting that as of my last update in September 2021, these details are accurate, but there might have been further developments or changes in the field since then.

II. LITERATURE REVIEW

This chapter essentially exposes the relevant literature evaluation based on research papers for the advancement of research work done. In recent years, the increasing complexity of algorithms in all areas of electronics, along with the rapid reduction of integrated circuits, has begun to alter the situation in which hardware and IC designers develop their implementations. At that rate, with the existing and expected complexity level, and with a non-declining consumer market demanding cheaper but also more advanced products, designers and businesses cannot afford to create any design from scratch. At this stage, reusability and, more especially, "macro" design become extremely important. High-speed multipliers, filter structures with specialised functions designed to be adaptable in not just one but two ways When corporations compete to position their goods sooner, more competitively, and at lower costs in the present exceptionally aggressive commercial centers, not one, yet various applications build up forward movement. As PC and sign handling applications have extended, so has the interest for high velocity handling. In some ongoing sign and picture handling applications, higher throughput number juggling activities are expected to accomplish the required presentation. Duplication is a basic numerical activity in such applications, and the improvement of quick multiplier circuits has been a well-established center. Time deferral and power utilization decrease are basic standards for some applications. Different multiplier designs are displayed in this paper. One of the quickest and most reduced power multipliers depends on Vedic science.

Poornima M, et al. (2019), proposes the development of a rapid Vedic Multiplier in light of Vedic Math moves toward that have been refreshed to help execution. A high velocity processor is vigorously dependent on the multiplier, which is a basic equipment part in most computerized signal handling frameworks and universally useful processors. Vedic Science utilizes an exceptional working out strategy in light of 16 Sutras. This paper examines research on a rapid 8x8 piece Vedic multiplier engineering that varies from customary techniques for increase like add and shift. Moreover, Verilog HDL coding of the Urdhva tiryakbhyam Sutra for 8x8 piece increase and FPGA execution on Simple 3 unit utilizing Xilinx Amalgamation Apparatus have been finished, and yield has been displayed on LEDs. Spartan3 gear.

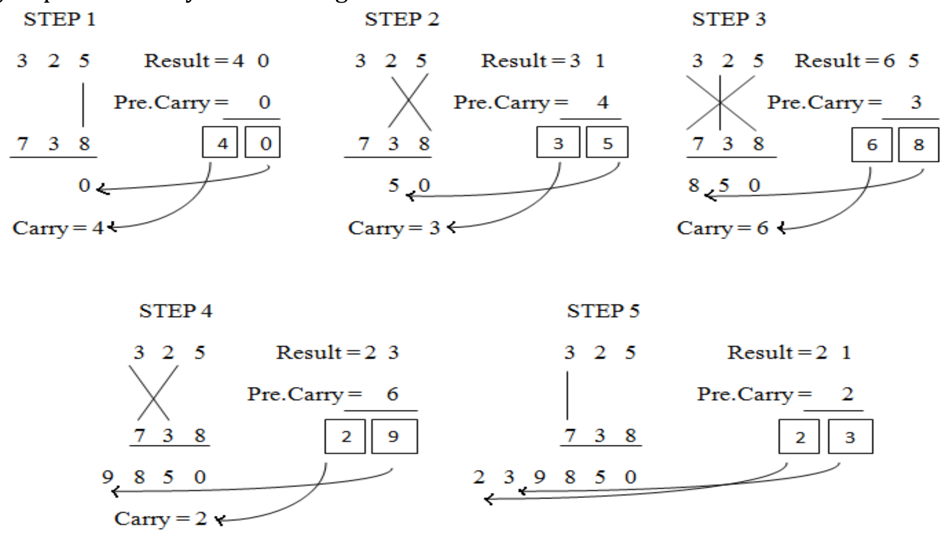
Premananda et al. (2017), the multipliers' speed is confined by the speed of the adders utilized for fractional item expansion. In this review, they proposed a 8-cycle multiplier for making fractional items in light of Vedic Science (Urdhva Tiryagbhyam sutra). The convey skip procedure is utilized to accomplish fractional item expansion in the Vedic multiplier. A 4-bit multiplier and changed swell convey adders are utilized to make a 8-digit multiplier. We brought down the quantity of rationale levels in the proposed design, which diminished the rationale idleness. The design is mimicked with Xilinx ISIM and incorporated with Xilinx XST. When contrasted with the standard Vedic multiplier, the outcomes show a 13.65% lift in speed.

Sheshavali et al (2016), It is proposed to develop a progressive multiplier design in view of the ROM technique using Vedic Science. The development of this multiplier is like that of a Steady Coefficient Multiplier (KCM). Nonetheless, KCM requires a decent info, though the recommended multiplier may different two factors. The recommended multiplier is carried out on a Tornado III FPGA and tried against the Cluster Multiplier and Urdhava Multiplier in both 8 cycle and 16 bit circumstances, with the outcomes detailed. The recommended multiplier is 1.5 times speedier than different multipliers in the 16x16 case and takes up only 76% of the space of the 8x8 multiplier and 42% of the region of the 16x16 multiplier.

III. METHODOLOGY

3.1 Duplication of Two Decimal Numbers in Vedic Strategy 325*738.

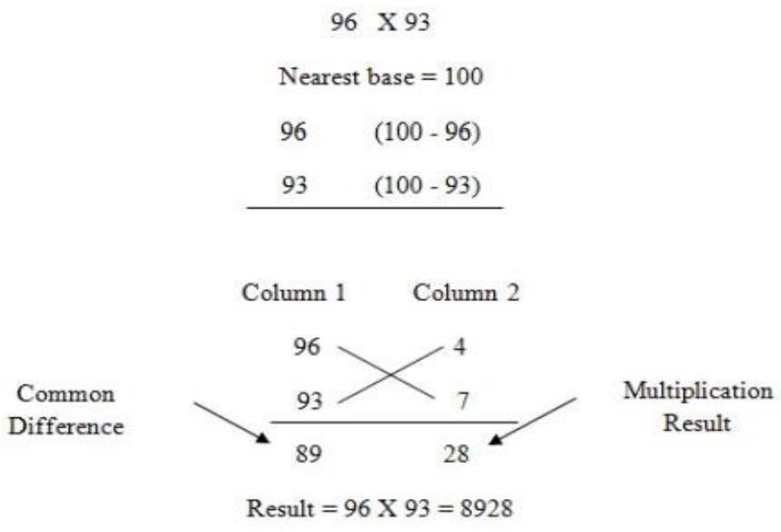
The convey from the past step is duplicated by the digits on the two sides of the line. This produces one of the outcome's pieces as well as a convey. This convey is included the accompanying stage, and the interaction proceeds. In the event that there are many lines in a solitary step, the outcomes are added to the first convey. In each step, the most un-critical piece fills in as the outcome bit, while the excess pieces act as the convey for the accompanying step. The convey is first thought to be zero.



325 X 738 = 239850

3.2 Nikhilam Sutra

Nikhilam Sutra in a real sense deciphers as "all from 9 and last from 10". Despite the fact that it is material to all increase circumstances, it is more proficient when the numbers included are colossal. Since it acquires the supplement of the huge number from its closest base to direct the duplication procedure on it, the bigger the underlying number, the less complex the increase. We start by concentrating on the increase of two decimal whole numbers (96 * 93) where the chose base is 100, which is both closest to and greater than both of these qualities.



3.3 The LMS Adaptation Algorithm

Widrow and Hoff fostered the LMS calculation in 1960 for use in brain network preparing. It look through the ideal weight vector utilizing a rough inclination estimation. This technique is utilized to decide the weight vectors for preparing the ALC (Adaline). The learning standards might be coordinated into a similar gadget, permitting it to auto-adjust when the ideal data sources and results are provided. As each information yield blend is handled, the weight vectors' qualities are modified. This interaction is rehashed until the ALC produces the ideal outcomes. This is a genuine preparation system on the grounds that the weight vector esteem needn't bother with to be determined expressly. Where is a union component that decides how enormous or small the sign is going through the channel; its worth is basic for combination speed and LMS calculation unwavering quality. It acknowledges values somewhere in the range of 0 and 2/MSmáx, where M is the quantity of channel steps and Smáx is the best worth of the info u step's power ghastly thickness. The LMS (least mean squares) approach is a steepest plunge guess that utilizes a prompt gauge of the slope vector of an expense capability. The inclination is assessed utilizing test values from the tap-input vector and a mistake signal. The strategy repeats over the channel, changing every coefficient in the course of the determined angle [1]. A reference is expected for the LMS calculation. The planned channel yield is addressed by signal d[n]. The mistake signal is the contrast between the reference signal and the genuine result of the cross-over channel.

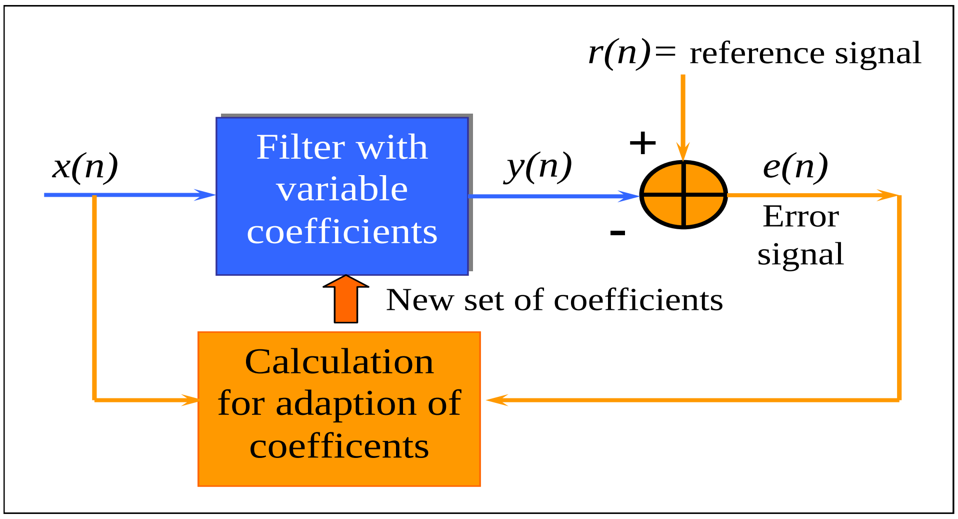

r(n)= reference signal
x(n)
Filter with variable coefficients
y(n)
+
-
e(n)
Error signal
New set of coefficients
Calculation for adaption of coefficents

Figure 3.1: Block diagram of LMS filter

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:08/August-2023 Impact Factor- 7.868 www.irjmets.com

LMS Algorithm Steps

- **Filter output** $y[n] = \sum_{k=0}^{M-1} u[n-k] w_k^*[n]$
- **Estimation error** $e[n] = d[n] - y[n]$
- **Tap-weight adaptation** $w_k[n+1] = w_k[n] + \mu u[n-k] e^*[n]$

$$\begin{pmatrix}\text{update value}\\ \text{of tap - weigth}\\ \text{vector}\end{pmatrix} = \begin{pmatrix}\text{old value}\\ \text{of tap - weight}\\ \text{vector}\end{pmatrix} + \begin{pmatrix}\text{learning-}\\ \text{rate}\\ \text{parameter}\end{pmatrix}\begin{pmatrix}\text{tap -}\\ \text{input}\\ \text{vector}\end{pmatrix}\begin{pmatrix}\text{error}\\ \text{signal}\end{pmatrix}$$

3.4 The Multiplier Architecture

Vedic Multiplication Technique and Its Application to Designing Multiplier Architecture. The Vedic multiplication technique is an ancient Indian method of multiplication that involves breaking down numbers into smaller parts and performing parallel calculations to arrive at the final product. This technique is based on various sutras (aphorisms) from Vedic mathematics. In the context of multiplier architecture, the Vedic multiplication technique can be utilized to design a parallel multiplier that computes partial products and their sums simultaneously, thus potentially improving the speed of multiplication operations. The advantage of this approach is that it can lead to a more efficient and streamlined design compared to other multiplication algorithms. In the specific example you provided, where two 64-bit numbers are being multiplied, the multiplier and multiplicand are divided into smaller blocks to enable parallel processing. The numbers are separated into 32-bit blocks, and afterward further isolated into 16-cycle blocks, etc, until you arrive at the littlest unit of computation, which for this situation is by all accounts two pieces. Every one of these blocks is then duplicated utilizing the Vedic increase strategy, and the outcomes are consolidated in an equal way to get the end result of the full 64-bit duplication. The critical benefit of this approach is that it can prompt a multiplier plan that requires less rationale entryways contrasted with some customary duplication calculations. This decrease in the quantity of doors can mean quicker augmentation activities and possibly more effective equipment executions. It's critical to take note of that the subtleties of the engineering would rely upon the particular Vedic duplication method being involved and the plan contemplations for the equipment execution. Moreover, while the Vedic duplication strategy can offer benefits in specific cases, present day processors likewise utilize an assortment of increase calculations upgraded for speed and effectiveness.

64 Bit Vedic Multiplier

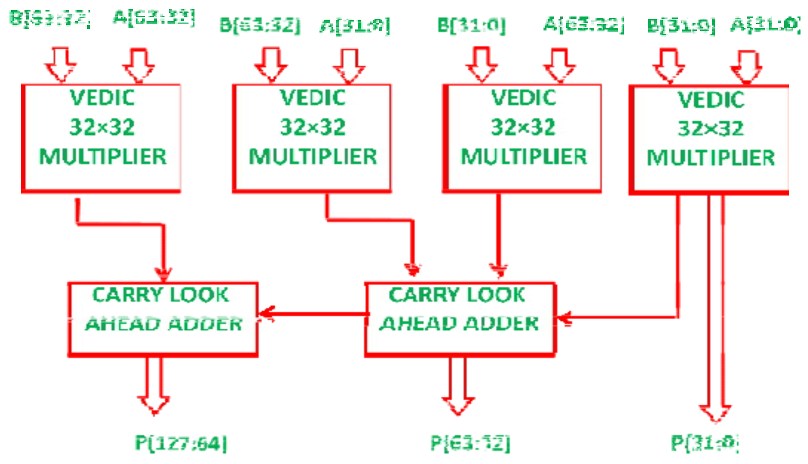


Figure 3.2: Vedic Multiplier 64x64 Bit Design Proposed

(P127-P64) & (P63-P32) & (P31-P0) Result

32 Bit Vedic Multiplier

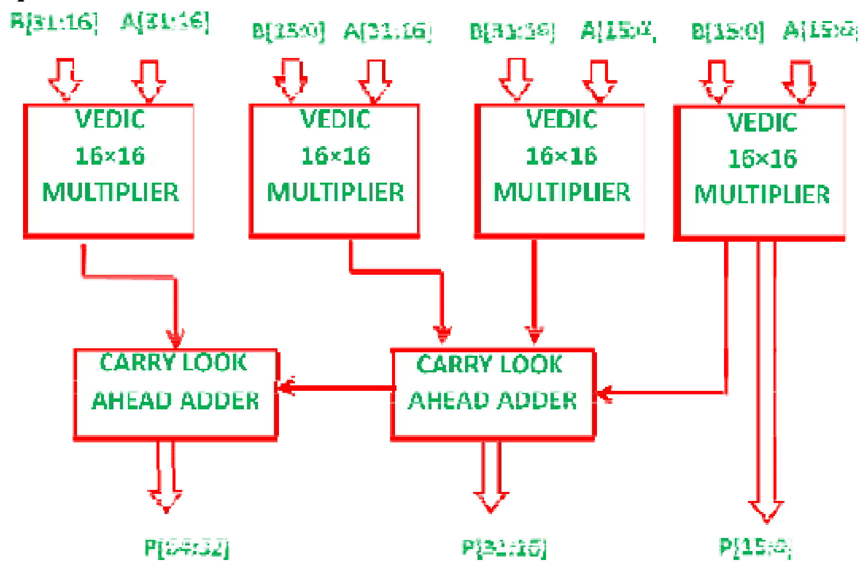


Figure 3.3: Vedic Multiplier Proposed 32X32 Bit.

(P63-P32) & (P31-P16) & (P15-P0) = Result

16 Bit Vedic Multiplier

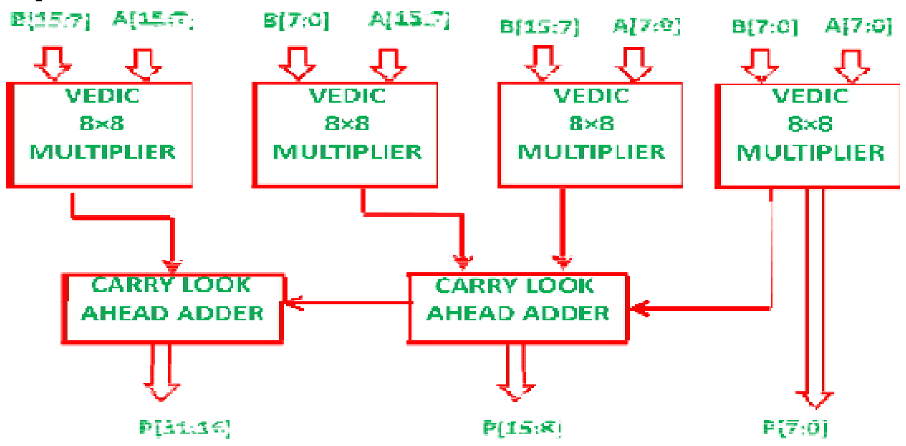


Figure 3.4: Disintegrated 16x16 Bit Vedic Multiplier

(P31-P16) and (P15-P8) and (P7-P0) = Result

8 BIT VEDIC MULTIPLIER

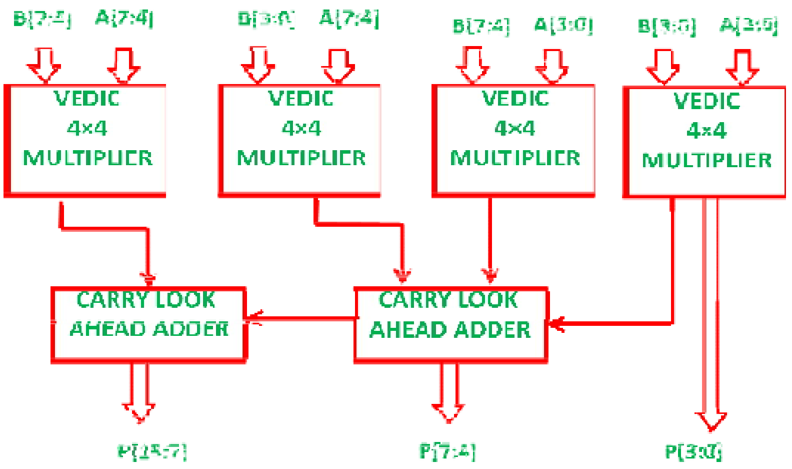


Figure 3.5: 8X8 Bit Decayed Vedic Multiplier

(P15-P8) and (P7-P4) and (P3-P0)= Result

IV. SIMULATION RESULT

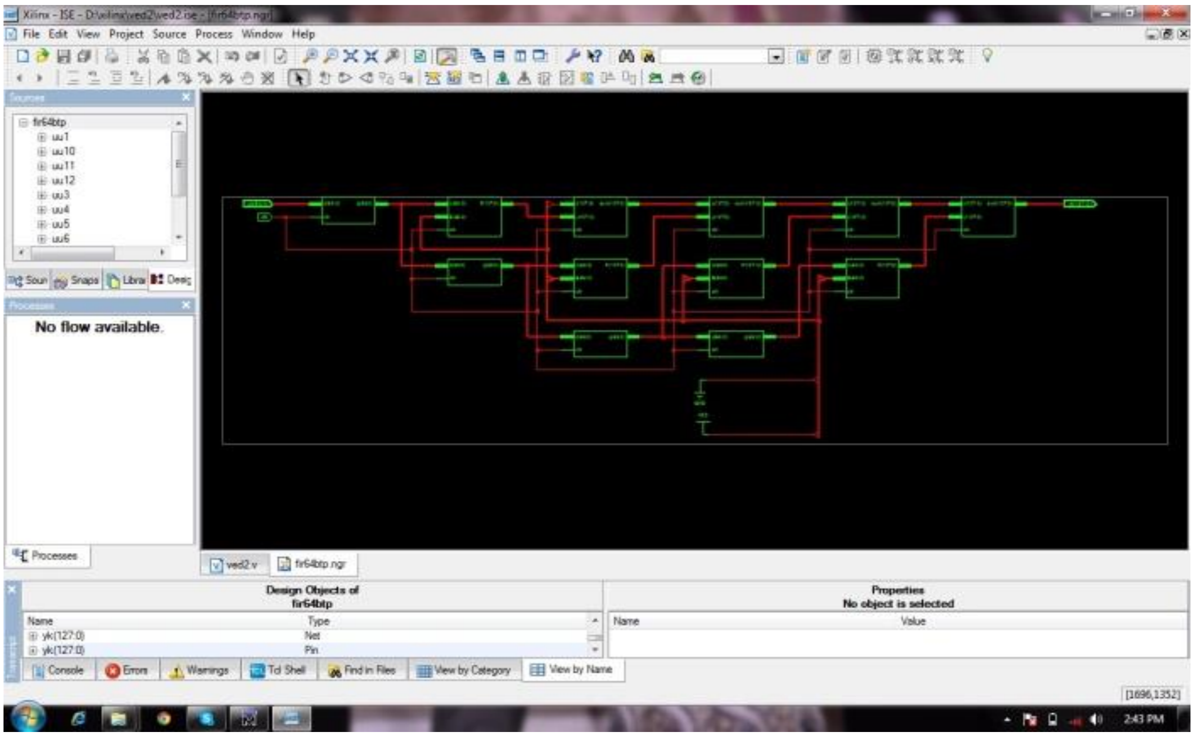


Figure 4.1: Diagram of an LMS filter

The Macro Statistics from the HDL synthesis report for a specific design, presumably a 64x64-bit Vedic Multiplier implementation. These statistics give insight into the resources used and the structure of the synthesized design. Let's break down the information you've provided:

Adders/Subtractors: There is a total of 1 adder/subtractor used in the design.

128-bit adder: There's one 128-bit adder in the design.

Registers: The design uses a total of 8199 registers.

128-bit register: There's one 128-bit register.

4-bit registers: There are 8192 4-bit registers.

64-bit registers: There are 6 64-bit registers.

XOR Gates: The design uses a total of 24582 XOR gates.

1-bit XOR2: There are 16384 1-bit XOR gates.

128-bit XOR2: There are 12 128-bit XOR gates.

16-bit XOR2: There are 768 16-bit XOR gates.

32-bit XOR2: There are 192 32-bit XOR gates.

4-bit XOR2: There are 4096 4-bit XOR gates.

64-bit XOR2: There are 58 64-bit XOR gates.

These statistics provide insight into the utilization of various resources in the design, such as adders, registers, and XOR gates. The 128-bit adder is likely used for the final accumulation stage, where the fractional items are added to deliver the eventual outcome. The registers are used for storing intermediate results and operands during various stages of computation. XOR gates are likely used in the Vedic multiplier architecture for various computations involving partial products and accumulations.

Overall, this information gives you an idea of the complexity and resource utilization of the synthesized Vedic multiplier design. It seems like a relatively efficient and compact design, considering the number of resources used for a 64x64-bit multiplication operation.8-bit xor2: 3072

Synthesis Report Of Adaptive LMS Filter

Discharge 7.1i - xst H.38

Copyright (c) 1995-2005 Xilinx, Inc. Protected by copyright law.

--> Parameter TMPDIR set to __projnav

Computer chip: 0.00/1.97 s | Passed : 0.00/1.00 s

--> Parameter xsthdpdir set to ./xst

Computer chip: 0.00/1.97 s | passed: 0.00/1.00 s

--> Understanding plan: adaptive_filter64.prj

List of chapters

1) Synthesis Choices Synopsis

2) HDL Compilation

3) HDL Analysis

4) HDL Synthesis

5) High level HDL Synthesis

5.1) HDL Synthesis Report

6) Low Level Synthesis

7) Final Report

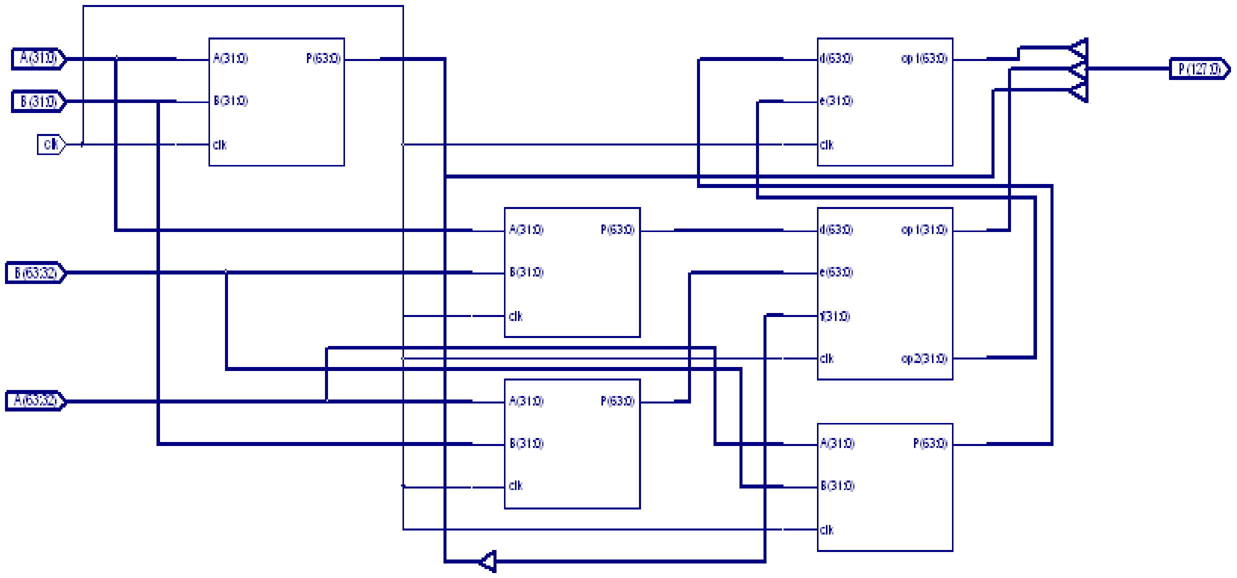


Figure 4.2: Syntheses Result 64×64 Bit Vedic Multiplier.

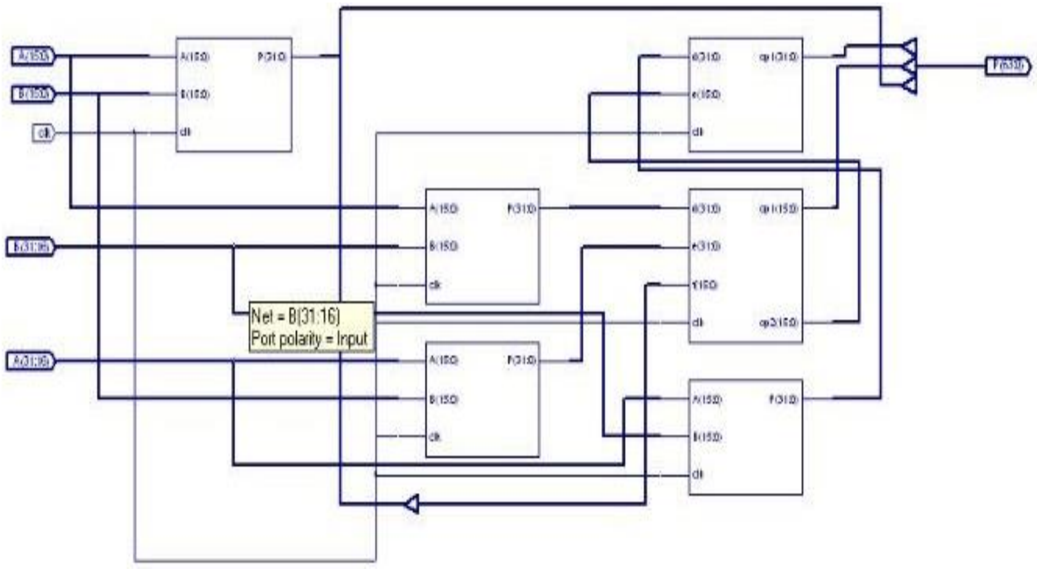


Figure 4.3: Syntheses Result 32×32 Bit Vedic Multiplier.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:08/August-2023 Impact Factor- 7.868 www.irjmets.com

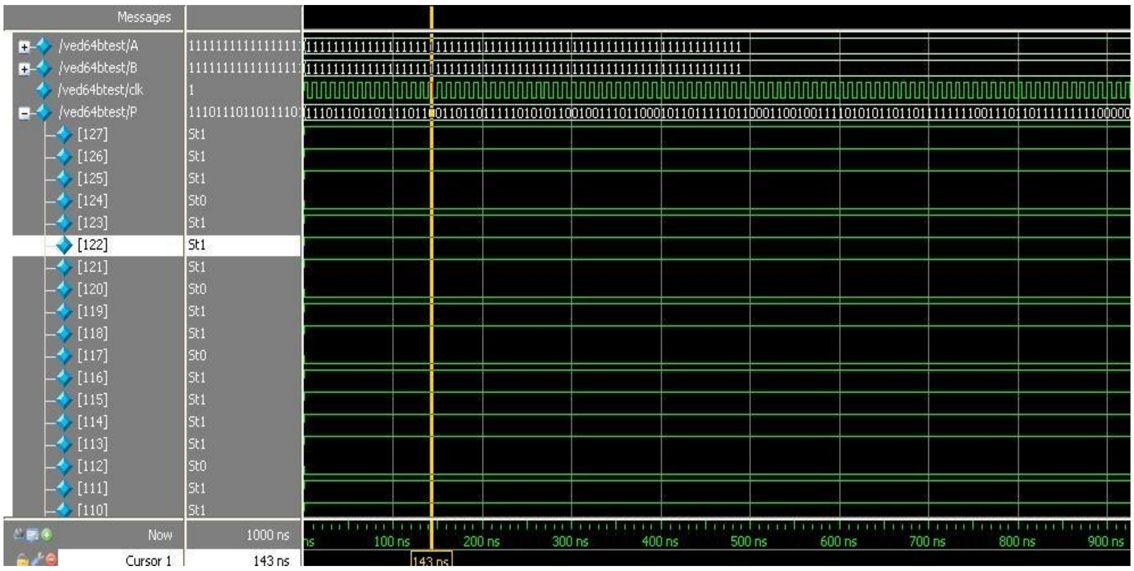
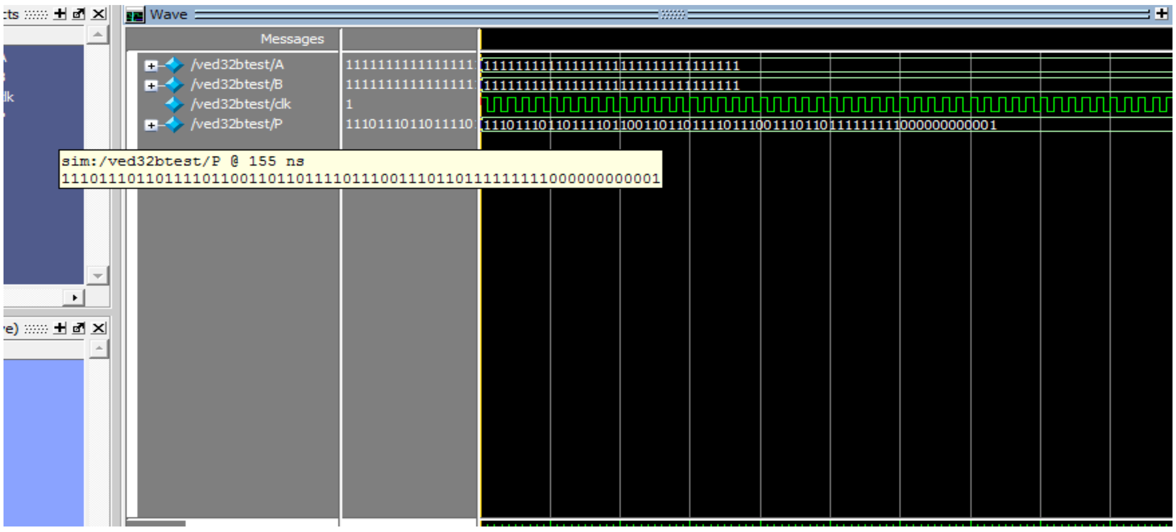


Figure 4.4: Simulation Result 64×64 Bit Multiplier.



```
sim:/ved32btest/P @ 155 ns
1110111011011110110011011011110111001110110111111111000000000001
```

Figure 4.5: Simulation Result 32×32 Bit Multiplier.

An HDL synthesis report for a 64-bit multiplier design. This report provides information about various aspects of the synthesized design, like the quantity of registers, goes back and forth, and XOR entryways utilized. Allow me to separate the data for you:

# Registers: This indicates the total number of registers used in the design. In this case, there are 4096 registers.

Flip-Flops: Flip-flops are a type of digital circuit element used for storing binary information (0 or 1). In your design, there are also 4096 flip-flops, which is consistent with the number of registers.

# Xors: This shows the all-out number of XOR doors utilized in the plan. In your plan, there is 1022 XOR doors altogether.

16-bit xor2: This likely refers to 16-bit XOR gates, and there are 96 of them in your design. These gates could be used for performing XOR operations on 16-bit data.

32-bit xor2: Similarly, there are 24 instances of 32-bit XOR gates, which could be used for XOR operations on 32-bit data.

4-bit xor2: There are 512 4-bit XOR gates. These gates might be used for performing XOR operations on 4-bit data.

64-bit xor2: There are 6 instances of 64-bit XOR gates. These gates could be used for XOR operations on 64-bit data.

8-bit xor2: Lastly, there are 384 8-bit XOR gates, likely used for XOR operations on 8-bit data.

The information you've provided gives insights into the structure of the synthesized multiplier design, particularly in terms of registers, flip-flops, and XOR gates of various sizes. This type of information is essential for assessing the efficiency and complexity of a digital design, and it's commonly used by hardware engineers during the design and optimization process.

Output. The mistake signal is the contrast between the reference signal and the genuine result of the cross-over channel.

V. CONCLUSION

The recommended Vedic multiplier engineering beats the multiplier design gave in the 64x64 Vedic multiplier, as well as other lower bit multipliers in view of the Urdhva Tiryakbhyam Sutra, as far as speed and intricacy at the door level plan engineering. This technique might be reasonable for increase of numbers bigger than 64 pieces. This work portrays a versatile FIR channel framework that utilizes the most un-Mean-Square Calculation and gives benefits with regards to adaptability, power productivity, and design time. This undertaking contains a recently proposed Vedic multiplier thought distributed in IETE diaries that is extremely proficient with regards to speed and can be promptly acknowledged on silicon because of its standard and equal construction, bringing about a lessening in equipment cost. The Movable The Most un-Mean Square FIR channel is written in Verilog HDL, then orchestrated with Xilinx7.1i and recreated with Modelsim 6.5.The introduced Versatile Least Mean Square FIR channel framework is liable for giving the best answer for FIR channel acknowledgment and independent transformation, and can be utilized in different correspondence and computerized signal handling applications like Channel Leveling, Versatile commotion dropping, Versatile reverberation retraction, Framework distinguishing proof, and numerous others.

VI. FUTURE WORK

Urdhava Tiryakbhyam (In an upward direction and transversely) is one of the Sixteen Vedic Sutras that arrangements with mathematical duplication. The sutra is portrayed before in the postulation. Two decimal qualities (31 x 35) are duplicated in this model. The numerals at the line's finishes are increased, and the outcome is added to the previous convey. At the point when there are at least three lines, the outcomes are added to the previous convey. The most un-huge digit of the subsequent number fills in as one of the outcome digits, while the rest of as the convey for the accompanying step. The convey is first thought to be zero. What's in store depends on a ROM procedure, albeit both multiplier data sources can be adaptable. In this proposed arrangement, a ROM is used to store the squares of numbers, rather than KCM, which stores the products. To get (a x b), we should initially decide whether the contrast among 'a' and 'b' is odd or even. The item is resolved utilizing (1) and (2) in light of the distinction.

VII. REFERENCES

[1] Poornima M, Shivaraj Kumar Patil, Shivukumar , Shridhar K P , Sanjay H, "Implementation of Multiplier using Vedic Algorithm", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-6, May 2013.

[2] Premananda B.S. , Samarth S. Pai, Shashank B., Shashank S. Bhat "Design and Implementation of 8-Bit Vedic Multiplier " International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization) Vol. 2, Issue 12, December 2013

[3] C.Sheshavali M.Tech1, K.Niranjan kumar Asst. professor "Design and Implementation of Vedic Multiplier" International Journal of Engineering Research and Development e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com Volume 8, Issue 6 (September 2013), PP.23-28

[4] G.Ganesh Kumar, V.Charishma "Design of High Speed Vedic Multiplier using Vedic Mathematics techniques " International Journal of Scientific and Research Publications, Volume 2, Issue 3, March 2012 1 ISSN 2250-3153

[5] Haykin, Simon, Adaptive Filter Theory, Prentice Hall, Upper Saddle River, New Jersey, 1996.

[6] Bernard Widrow and Samuel D. Stearns: "Adaptive Signal Processing", Prentice-Hall, Inc., Upper Saddle River, NJ, 1985.

[7] Honig, Michael L., Messerschmitt, David G., Adaptive Filters, Structures, Algorithms and Applications, Kluwer Academic Publishers, Boston, 1984.

[8] Jenkins, W. Kenneth, Hull, Andrew W ,Strait, Jeffrey C., Schnaufer, Bernard A., Li, Xiaohui, Advanced Concepts in Adaptive Signal Processing, Kluwer Academic Publishers, oston, 1996.

[9] Purushottam D. Chidgupkar and Mangesh T. Karad, "The Implementation of Vedic Algorithms in Digital Signal Processing", Global J. of Engng. Educ., Vol.8, No.2 © 2004 UICEE Published in Australia.

[10] Himanshu Thapliyal and Hamid R. Arabnia, "A Time-Area- Power Efficient Multiplier and Square Architecture Based On Ancient Indian Vedic Mathematics", Department of Computer Science, The University of Georgia, 415 Graduate Studies Research Center Athens, Georgia 30602-7404, U.S.A.

[11] E. Abu-Shama, M. B. Maaz, M. A. Bayoumi, "A Fast and Low Power Multiplier Architecture", The Center for Advanced Computer Studies, The University of Southwestern Louisiana Lafayette, LA 70504.