DESIGN AND IMPLEMENTATION OF NOVEL 32 BIT MAC UNIT FOR DSP APPLICATIONS

[1]V. Katamma, [2]Dr. M Pavitra Jyothi
[1]PG Scholar, M.Tech, Dept of VLSI, Shadan Women's College of Engineering and Technology, Hyderabad, TS.
abrahammarkapudi@gmail.com
[2]Professor, Dept of ECE, Shadan Women's College of Engineering and Technology, Hyderabad, TS.

ABSTRACT
In today's smart and fast computing world, the designing of high speed and low energy consumption based Digital Signal Processors (DSPs) is a realistic and ever embryonic area of research. Conversely, the design of a proficient Digital Signal Processor intended to carry out the complex computations associated with image processing or signal processing involves the design of an efficient Multiply-Accumulate (MAC) unit which is one of the most vital blocks of processor. The multiplier, adder, accumulator are the fundamental construction sub-units for MAC units. Moreover, the computation carried out with the extensive and appropriate usage of Vedic Mathematics is set up to be well proficient and capable as compared to the basic Mathematics. This paper has presented the implementation of novel 32-bit MAC unit consisting of Vedic Multiplier using Urdhva Tiryakbhyam sutra and efficient adder circuit using Modified Weinberger adder technique. From comparative analysis, the MAC unit designed was found to be proficient in terms of delay and energy consumed.

INTRODUCTION

MAC (Multiply Accumulate) units are the one which are used in order to carry out the specific types of arithmetic operations namely multiplication and accumulation. These units are quite handful to perform fundamental and momentous operations in today's smart and emerging technological world. These operations are regularly used in high-speed microprocessors, microcontrollers, all large- and small-scale data computing units and digital signal processors. Designing MAC units for digital signal processors or intended for applications related to digital signal processing (DSP) is one of the ongoing and incredibly effervescent field in the today's smart digital and electronic era. Also, low power and high-speed MAC units are tremendously in demand especially for high end smart applications such as Audio and video signal processing, Data Analytics, Artificial Intelligence (AI), Machine learning, RADAR, Li-DAR, SONAR and further defense and military related applications [1-3].

However, more precisely the Multiply Accumulate (MAC) units are widely used in domain of digital signal processing in order to compute convolution, correlation, signal transformation and communication, calculation of filter coefficient and hence designing various types of digital filter, data and image pre-processing and various transformations etc. In addition, the analog signal processing (continuous domain) techniques are as well as functional and served in digital field along with their widespread practice in order to implement a range of algorithms and techniques such as Discrete Wavelet Transform (DWT), Discrete Fourier Transform (DFT), Laplace Transform (LT), Fast Fourier Transform (FFT), Z-Transform etc. This enables the users to get a hold of a soft cushion over the circuit intricacy and modularity, response time, execution time, computational time and power consumption. Therefore, the performance matrix of the designed MAC unit is going to conclude the performance parameters matrix for the desired application [4].

Hence, designing of proficient and optimized MAC units is one of the key potential areas of current and ongoing trend of research. The standard and functional architecture of MAC unit is comprised of product calculator or multiplier, summer or adder and a data accumulation register. The multiplier unit here is going to carry out the generation of various partial products; cutback and trimming of these generated partial products, propagation of a previous stage carry to next stage and further addition of the propagated carry. Furthermore, the architectural design of an accumulator is a commonly used PIPO (Parallel Input Parallel Output) shift register. These registers facilitate the storage of the generated result which is obtained after data accumulation. However, the contemporary and available result achieved from the multiplier unit is auxiliary added with the preceding stored result of the accumulator unit in the designed adder unit.

The traditional multiplication operation is achieved by performing the practice of recurring addition. Hence, the complication of multiplication increases momentarily with the increase in the count. This traditional process of fundamental arithmetical logics and operations turns out to be further monotonous and tiresome for higher order data. Hence, designing of efficient and prolific multiplier is desirable. A variety of multipliers architectures are surveyed and presented with a brief note on their various pros and cons in [5]. Furthermore, in order to carry out the complex arithmetic computations an alternative approach of Vedic mathematics in incorporated. Vedic Mathematics is the special branch of mathematics dealing with a

fastidious and specific set of rules fundamentally taken and formulated from the great Ancient Indian Scriptures, while expounding and facilitating a widespread use of assorted mathematical results and validation in uncomplicated and realistic forms.

Conversely, the word Vedic is essentially derived from the word 'Veda' which signify the storeroom or abode of the intact knowledge and awareness. Moreover, the Vedic mathematics was further revived and formulated by Sri Bharati Krisna Tirthaji from the great Indian Vedas in early 20th century in the form of standard sixteen sutras [6]. Vedic Mathematics has further demonstrated its supremacy in diversity of applications resulting in swift and smooth calculations such as addition, subtraction, multiplication, division, square, square root, cube, cube roots and so on [7].

These specific and standard Vedic arithmetic units enable researchers in order to propose and implement novel Vedic Multiply-Accumulate (MAC) units which can further be extended to design a wide range of applications.

EXISTING SYSTEM:

The design and implementation of a high speed, low power based 32-bit IEEE 754 multiplier using novel Booth algorithm was done in [8]. This design is consisting of 32-bit adder, subtraction and multiplier circuits. The multiplier implemented here can be further be used in Digital Signal Processing (DSP) applications such as Fast Fourier Transform (FFT), correlation, convolution, filtering etc. The design and implementation of an 8-bit Vedic multiplier was done in [9] with the general use of a special 8-bit barrel shifter. This barrel shifter needs only single clock cycle in order to perform n number of shifts. This design was found to be superior with respect to propagation delay in comparison of other traditional multipliers such as tree multiplier, Braun multiplier, Modified Booth multiplier, and Array multiplier. Another novel design of multiplier was proposed in [10] which results in the reduction of generation of partial products. Various types of adders were used here with comparison of various performance parameters.

However, Kogge Stone adder was selected among all since it was found to have optimized values for power consumption and propagation delay. A proficient and novel scheme for in order to perform squaring operation is implemented using Vedic sutra without multiplication operation in [11]. A novel design for area and delay efficient circuit in order to perform factorial calculation with the extensive use of application of Vedic algorithms was proposed in [12]. A delay efficient multiplier design with the extensive use of Nikhilam sutra of Vedic mathematics was proposed in [13]. This designed multiplier has salient features like reduction of a bigger operation into the smaller operations and enhanced speed of addition during accumulation stage of partial product with the extensive used of carry select adders. A multiplier less squaring technique with superior speed was proposed in [14]. In this algorithm, squaring operation of any n-bit number is performed using a (n−1) bit squaring sub unit along with a 2n bit binary adder and an another bit binary adder/subtractor circuit. This design was found to be delay and area efficient as compared to squaring circuit designed using booth multiplier.

In [15, 16] author has proposed a fanatical and proficient square and cube circuits designs with the extensive use of Vedic mathematics. These designed architectures switch the bigger size square or cube data into a smaller size data by performing certain specific arithmetic operations. Both the techniques work on basic decimal algorithm which can be further effectively broadened to the radix-2 binary number system making an allowance for the digital platforms. These square and cube designs were found to be superior in comparison of a variety of up to date usual and other Vedic architectures.

Design and implementation of a floating point MAC unit with use of basic mathematics was done in [17]. This helps in the cutback of the generated partial products and further enhancement in the speed of accumulation of generated partial products. Few other improvised schemes for MAC designs were discussed in [18]. The comparative performance analysis was prepared with various performance parameters such as power consumption, speed, delay and area. The comparative analysis proves the 8 bit MAC based on Baugh-Wooley Multiplier to be superior in terms of energy consumption at the cost of enlarged delay in comparison of the other schemes. Furthermore, another area and delay efficient design of MAC unit was implemented using modified Low Area Wallace tree multiplier (LAW) in [19]. This designed multiplier was found to be helpful to facilitate high speed applications especially for complex transformation in digital signal processing domain.

PROPOSED SYSTEM

In this research work, author has implemented the novel design of 32-bit MAC unit with 32-bit adder unit using Weinberger adder and hence the multiplier unit based on Urdhva Sutra and extensive use of Weinberger adder as a construction units. Since any real number can be represented in standard IEEE-32 bit format, hence the design was proposed for 32-bit. The design of 32-bit MAC unit starts with design of 4-bit MAC unit to 8-bit MAC unit to 16 bit MAC unit and hence 32-bit MAC. The final design of proposed 32-bit MAC unit will be developed in order to design the digital signal processing related applications which is ultimate objective and area of research of the author. Here, the

comparison of various performance parameters such as power, delay, and area has been carried out in order to compose inroads for auxiliary research.

LITERATURE SURVEY

"Correctly Rounded Architectures for Floating-Point Multi-Operand Addition and Dot-Product Computation", Y. Tao, G. Deyuan, D. Xiaoya, J. Nurmi.

This study presents hardware architectures performing correctly rounded Floating-Point (FP) multioperand addition and dot-product computation, both of which are widely used in various fields, such as scientific computing, digital signal processing, and 3D graphic applications. A novel realignment method is proposed to solve the catastrophic cancellation and multi-sticky bits. Only one rounding operation is performed in both of the proposed FP multi-operand adder and dot-product computation unit. Implementation results show that our architectures not only can produce correctly rounded results, whose errors are less than 0.5 ULP (Unit in the Last Place), but also have reduced delay comparing with the traditional network architecture, which uses 2-operand FP adders and multipliers to perform multi-operand addition and dot-product computation.

"**Floating-Point Sparse Matrix-Vector Multiply for FPGAs**", M. deLorimier, A. DeHon.

Large, high-density FPGAs with high local distributed memory bandwidth surpass the peak floating-point performance of high-end, general-purpose processors. Microprocessors do not deliver near their peak floating-point performance on efficient algorithms that use the Sparse Matrix-Vector Multiply (SMVM) kernel. In fact, it is not uncommon for micro-processors to yield only 10–20% of their peak floating-point performance when computing SMVM. We develop and analyze a scalable SMVM implementation on modern FPGAs and show that it can sustain high throughput, near peak, floating-point performance. For benchmark matrices from the Matrix Market Suite we project 1.5 double precision Gflops/FPGA for a single Virtex II 6000-4 and 12 double precision Gflops for 16 Virtex IIs (750Mflops/FPGA).

"**An FPGA-specific approach to floating-point accumulation and sum-of-products**", F. de Dinechin, B. Pasca, O. Cret, R. Tudoran.

This article studies two common situations where the flexibility of FPGAs allows one to design application-specific floating-point operators which are more efficient and more accurate than those offered by processors and GPUs. First, for applications involving the addition of a large number of floating-point values, an ad-hoc accumulator is proposed. By tailoring its parameters to the numerical requirements of the application, it can be made arbitrarily accurate, at an area cost comparable to that of a standard floating-point adder, and at a higher frequency. The second example is the sum-of-product operation, which is the building block of matrix computations. A novel architecture is proposed that feeds the previous accumulator out of a floating-point multiplier whose rounding logic has been removed, again improving the area/accuracy tradeoff. These architectures are implemented within the FloPoCo generator, freely available under the LGPL.

"Design, Implementation and Performance Analysis of an Integrated Vedic Multiplier Architecture", Ramachandran.S, Kirti.S.Pande.

Fundamental and the core of all the Digital Signal Processors (DSPs) are its multipliers and speed of the DSPs is mainly determined by the speed of its multipliers. Multiplication is the most fundamental operation with intensive arithmetic computations. Two important parameters associated with multiplication algorithms performed in DSP applications are latency and throughput. Latency is the "real delay of computing a function". Throughput is a measure of "how many computations can be performed in a given period of time". The execution time of most DSP algorithms is dependent on its multipliers, and hence need for high-speed multiplier arises. Urdhva tiryakbhyam sutra performs faster for small inputs and Nikhilam sutra for larger inputs. Here a novel Integrated Vedic multiplier architecture, which by itself selects the appropriate multiplication sutra based on the inputs, is proposed. So depending on inputs, whichever sutra is faster, that sutra is selected by the proposed integrated Vedic multiplier architecture. In the simulation results, it can be seen that Urdhva performs faster for small inputs, but Nikhilam performs better for large inputs (more than twice as much for 64 bit multiplicands).

PROPOSED METHODOLOGY

BLOCK DIAGRAM

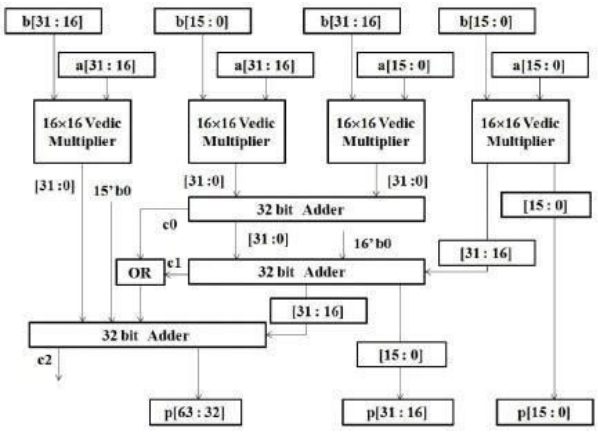


Fig. 32 x 32 Vedic Multiplier

MODULE EXPLANATION:

Urdhva Tiryagbhyam Sutra:

The Urdhva Tiryagbhyam Sutra as the name suggests is based on the vertical and further crosswise multiplication technique since derived from Sanskrit literature where the word 'Urdhva' means 'Vertical' and the word 'Tiryagbhyam' means 'crosswise'. Lets' we need to perform multiplication on two numbers (both are two digit numbers) as (a1x+b1) and (a2x+b2). The simple product is given by $a1a2x^2 + (a1b2+ b1a2) x + b1b2$. Following are the steps involved during multiplication for a 2-digit multiplier.

Step 1: The first value i.e. the coefficient of x^2 is achieved by performing the vertical multiplication of a1 and a2.

Step 2: The second value i.e. the coefficient of x is achieved by performing the crosswise multiplication of a1 and b2 and of b1 and a2 and further the addition of the two obtained partial products.

Step 3: The third value i.e. constant coefficient is achieved by performing the vertical multiplication of constants b1 and b2.

The Urdhva Tiryagbhyam Sutra based algorithm is basically based on the scheme of generation of all partial products along with their additions happening simultaneously in concomitant fashion. Hence, the multiplier is solely independent of processor clock frequency for synchronous circuits, which results in quick calculation of final product and superiority at higher order frequency clock circuits. Furthermore, the above discussed steps can further be extended and generalized for higher order multiplications.

Mux Based Full Adder (Mux-FA)

The traditional full adder in order to design an array multiplier is substituted with a different type of full adder which is designed using multiplexer (MUX) XOR gates. Here, two XOR gates and one 2:1 MUX is used. This design results in decrease of area and delays further. The architecture is shown in fig.

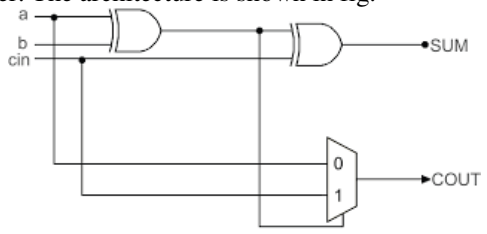


Fig. Mux based Full Adder (Mux-FA)

Modified Weinberger Adder

Weinberger adder improvises the speed of the addition process by performing the parallel computing technique for generation of carry. This works on the recurrence algorithm. The general 8-bit architecture of Weinberger adder is shown in Fig.2 along with all the standard logics carried out which can be extended to design 32-bit architecture. This

architecture lays out forward the universal structure of recurrence of carry generation which is never confined to amount of steps and bits in terms of computation and complexity. As a result, this trim down the delay propagated throughout the adder architecture and hence augments the swiftness of performing the addition.

Similar to the architecture shown in Fig.2, in 32-bit Weinberger architecture the last carry, is result of recurrence. In traditional adders in order to compute, it is mandatory to perform computation of previous carry; while in the case of

Weinberger adder the recurrence carry is adequate in order to perform computation. Here, the rapidity of generation and propagation of carry is mainly due to the extensive utilization of generate and propagate. However, there are bitwise generate (g), set wise generate (G), bitwise propagate (p) and set wise propagate (P) to perform the carry generation. The computation of all these generate and propagate terms is performed as shown in Fig.3. Moreover, the loading effects during computations over the obtained carry seem to be mounting but they merely affect it due the abridged and diminished computations of propagate and generate terms along with the reduced set of carry generation steps. This auxiliary boosts the energy saving and further enables the scope of delay fix.

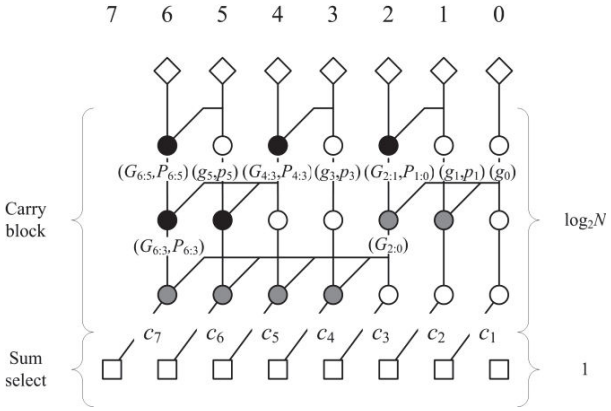


Fig. General Architecture of Weinberger Adder

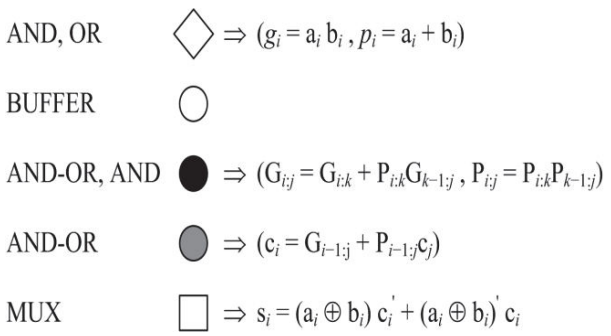


Fig. General Logic Operations Performed for Weinberger Adder

The modified Weinberger is proposed in order to optimize the area, power and delay further. As shown in Fig.4, the modified Weinberger adder is consisting of Ripple carry adder (RCA), Binary to Excees-1 Converter unit (BX1C) and Multiplexer.

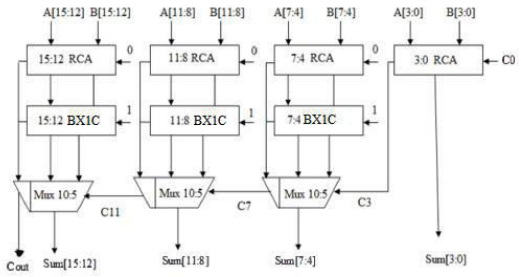


Fig. 4. Proposed Modified Weinberger Adder

Ripple Carry Adder (RCA)

Ripple carry adder is a conventional and basic adder circuit which designed using numerous basic single bit full adders in order to perform addition of n-bit numbers. However, each individual unit of full adder accepts Cin as one of the inputs which is traditionally Cout of preceding adder unit. Since, the carry bit from preceding adder unit is getting rippled to the next adder unit hence circuit is named as ripple carry adder. This circuit is easy to design but the major drawback is propagation delay. This slowness is mainly due to waiting time during preceding carry to get rippled into next adder unit. A simple 4-bit ripple carry adder is consisting of four single bit full adders. This is generally constructed using AOI (AND OR INVERT) logic.

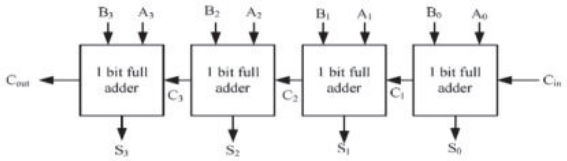


Fig. Ripple Carry Adder (RCA)

Binary to Excess-1 Converter Unit (BX1C)

In order to design power and area efficient structure of RCA with input carry as '1', the basic unit is swapped with Binary excess-1 converter (BX1C). The optimized design result in reduction of the Ex-OR logic gates as shown in Fig.6. This converter unit acts as optimized carry select adder architecture with ripple carry adder along with input carry as '0'.

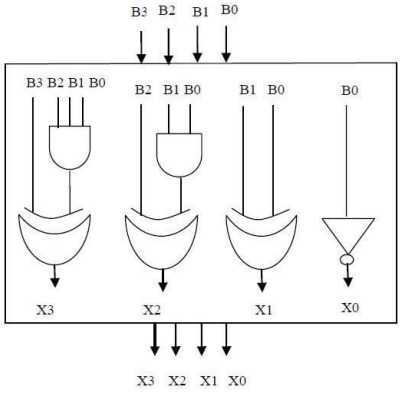


Fig. Binary to Excess-1 Converter Unit (BX1C)

10:5 Multiplexer

Multiplexer is a universal combinational circuit. A traditional multiplexer has of 2^n inputs, 'n' select lines and one output. The working of multiplexer depends upon decision made by the select lines in order to choose appropriate input for desirable output. Five independent 2:1 multiplexers are combined to obtain the desired 10:5 multiplexer here.

32x32 Vedic Multiplier

Design of 32x32 Vedic multiplier starts with the design of 4x4 Vedic multiplier and extension to 8x8 Vedic multiplier, extension to 16x16 Vedic multiplier, and hence extending to 32-bits. The general architecture of a 32x32 Vedic multiplier is shown in Fig.7. Here, total four 16x16 Vedic multipliers along with three 32-bit adder units are used. The very first 32-bit Adder is used in order to perform addition for two 32-bit operands which are achieved after performing the cross multiplication for the two middle 16x16 Vedic multiplier units. The second 32-bit adder is used in order to perform addition for two 32-bit operands which are achieved after concatenation of 32-bit (16 bits of 0's and the other 16 bits as the most significant bits of right hand most of 16x16 Vedic multiplier unit) and another 32-bit operand is achieved as the result of first 32-bit adder. The output carry from this adder is further given as input to the third 32-bit adder unit. Next, the third 32-bit adder unit is used in order to perform addition for two 32-bit operands which is achieved after concatenation of 32bit (Carry output of first adder unit, 15 bits of 0's and other 16 bits as the most significant output sum result bits of second adder) and another 32-bit operand is achieved as the result of left hand most of first 32-bit adder one 32-bit operand we get as the output sum of left hand most of 16x16 Vedic multiplier unit. Depending on the types of adder used in circuit; delay, power, area etc. types of performance parameters can be optimized further.

32-bit MAC Unit

MAC (Multiply Accumulate) units are the one which are used in order to carry out the specific types of arithmetic operations namely multiplication and accumulation. This is a key objective in digital signal processing (DSP) with wide range of applications. The standard design of MAC unit as shown in Fig.8, is comprised of 3 basic sub circuits as a 32x32 Vedic multiplier, Weinberger recurrence algorithm-based adder and accumulator to integrate both the Vedic multiplier and adder units.

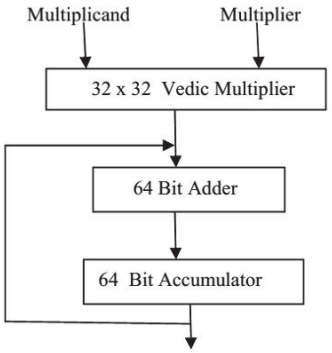
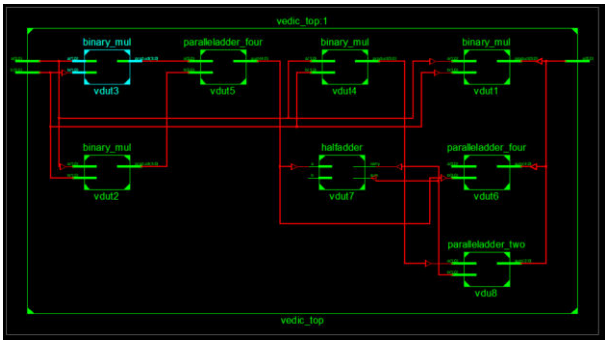


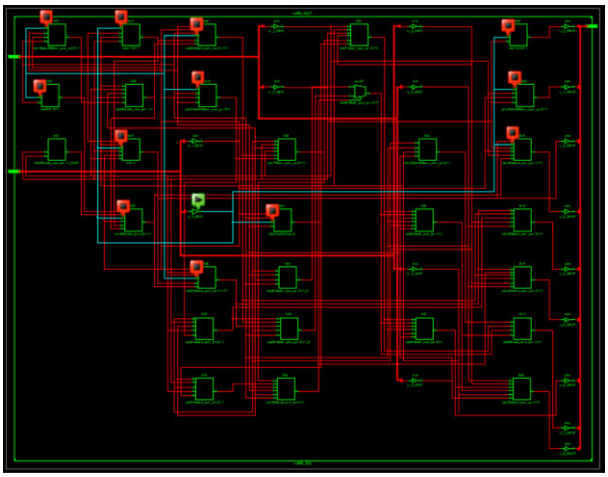
Fig. Proposed MAC Unit based on Vedic Multiplier & Weinberger Recurrence Adder

SIMULATION RESULTS:

vedic_top Project Status (07/03/2022 - 16:03:34)			
Project File:	work.xise	Parser Errors:	No Errors
Module Name:	vedic_top	Implementation State:	Synthesized
Target Device:	xc6slx9-3tqg144	* Errors:	No Errors
Product Version:	ISE 14.7	* Warnings:	No Warnings
Design Goal:	Balanced	* Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	* Timing Constraints:	
Environment:	System Settings	* Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	28	5720	0%
Number of fully used LUT-FF pairs	0	28	0%
Number of bonded IOBs	17	102	16%





Device Utilization Summary				
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	0	11,440	0%	
Number of Slice LUTs	24	5,720	1%	
Number used as logic	24	5,720	1%	
Number using O6 output only	21			
Number using O5 output only	0			
Number using O5 and O6	3			
Number used as ROM	0			
Number used as Memory	0	1,440	0%	
Number of occupied Slices	11	1,430	1%	
Number of MUXCYs used	0	2,860	0%	
Number of LUT Flip Flop pairs used	24			
Number with an unused Flip Flop	24	24	100%	
Number with an unused LUT	0	24	0%	
Number of fully used LUT-FF pairs	0	24	0%	
Number of slice register sites lost to control set restrictions	0	11,440	0%	
Number of bonded IOBs	17	102	16%	
Number of RAMB8BWERs	0	32	0%	
Number of RAMB8BWERs	0	64	0%	
Number of BUFIO2/BUFIO2_2CLKs	0	32	0%	
Number of BUFIO2FB/BUFIO2FB_2CLKs	0	32	0%	
Number of BUFG/BUFGMUXs	0	16	0%	
Number of DCM/DCM_CLKGENs	0	4	0%	

CONCLUSION

This research has presented the implementation and design of a 32-bit MAC unit based on Vedic Mathematics. The multiplier for presented MAC unit is designed Vedic Multiplier using Urdhva Tiryakbhyam sutra. The addition circuit is designed using modified Weinberger adder. From comparative analysis, MAC unit designed with Vedic Multiplier and using modified Weinberger adder was found to be efficient in terms of power and latency at the cost of area. This reduction in delay and energy consumption leverages the scholar in order to design high speed and compact SoC applications intended for DSP related application such as correlation, convolution, digital filter design and digital image processing etc. for further research work. However, there is scope in this research work in order to work upon the area constraint too which is not discussed here which will be another topic of interest of further research.

REFERENCES:

[1] Y. Tao, G. Deyuan, D. Xiaoya, J. Nurmi, "Correctly Rounded Architectures for Floating-Point Multi-Operand Addition and Dot-Product Computation", Proc. 24th IEEE Conf. on Application-Specific Systems, Architectures and Processors (ASAP), pp. 346-355, 2013

[2] M. deLorimier, A. DeHon, "Floating-Point Sparse Matrix-Vector Multiply for FPGAs" Proc. ACM/SIGDA 13th Int. Symp. On Field Programmable Gate Arrays (FPGA), pp. 75-85, 2005

[3] F. de Dinechin, B. Pasca, O. Cret, R. Tudoran, "An FPGA-specific approach to floating-point accumulation and sum-of-products" Proc. 2008 Int. Conf. on Field Programmable Technology (FPT), pp. 33-40, 2008

[4] Mathematical Thought in Vedic India, Chapter-2, Mathematics in India. doi:10.1515/9781400834075.13.

[5] Ramachandran.S, Kirti.S.Pande, Design, Implementation and Performance Analysis of an Integrated Vedic Multiplier Architecture, International Journal of Computational Engineering Research, May-June 2012, Vol. 2. Issue No.3, 697-703

[6] Prabir Saha, Arindam Banerjee, Partha Bhattacharyya, Anup Dandapat, 2011. High Speed ASIC Design of Complex Multiplier Using Vedic Mathematics, IEEE Students' Technology Symposium, IIT Kharagpur

[7] Pavan Kumar U.C.S, Saiprasad Goud A, A.Radhika, FPGA Implementation of High Speed 8-bit Vedic Multiplier Using Barrel Shifter, International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 3, March 2013

[8] S. Deepak and Binsu J Kailath, 2012, Optimized MAC unit design, (2012) IEEE EDSSC 2012, IEEE International Conference on Electron Devices and Solid-State Circuits held from 3-5 Dec. 2012

[9] Mohammed Hasmat Ali, Anil Kumar Sahani, June 2013 Study, Implementation and Comparison of Different Multipliers based on Array, KCM and Vedic Mathematics Using EDA Tools, International Journal of Scientific and Research Publications, Volume 3, Issue 6

[10] M. Ramalatha, K. Deena Dayalan, P. Dharani, S. Deborah Priya, High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques, ACTEA 2009, July 15-17, 2009 Zouk Mosbeh, Lebanon

[11] Ajinkya Kale, Shaunak Vaidya, Ashish Joglekar, A Generalized Recursive Algorithm for Binary Multiplication based on Vedic Mathematics

[12] Meenu S Ravi, R H Khade and Ajit Saraf, "Design of Fast Floating Point Multiply Accumulate Unit using Ancient Mathematics for DSP Applications." European Journal of Advances in Engineering and Technology, pp.55-59, 2015.

[13] Anuja A. Bhat, Mangesh N.Thakare, Review on 32-Bit IEEE 754 Complex Number Multiplier Based on FFT Architecture using BOOTH Algorithm, IJECS, Vol. 6 Issue 2 Feb. 2017, Pg. No. 20308-20312

[14] S. Deepak and Binsu J Kailath, 2012, Optimized MAC unit design, (2012) IEEE EDSSC 2012, IEEE International Conference on Electron Devices and Solid-State Circuits held from 3-5 Dec. 2012

[15] Mohammed Hasmat Ali, Anil Kumar Sahani, June 2013 Study, Implementation and Comparison of Different Multipliers based on Array, KCM and Vedic Mathematics Using EDA Tools, International Journal of Scientific and Research Publications, Volume 3, Issue 6

[16] Kasliwal P.S., Patil B.P., Gautam D.K.: 'Performance evaluation of squaring operation by Vedic mathematics', IETE J. Res., 2011, 57, (1), pp. 39–41

[17] Saha P., Banerjee A., Dandapat A., ET AL.: 'ASIC design of a high-speed low power circuit for factorial calculation using ancient Vedic mathematics', Microelectronics J., 2011, 42, (12), pp. 1343–1352

[18] Pradhan M., Panda R., 'High speed multiplier using Nikhilam Sutra algorithm of Vedic mathematics', Int. J. Electron., 2014, 101, (3), pp. 300–307

[19] Sethi K., Panda R., 'Multiplier less high-speed squaring circuit for binary numbers', Int. J. Electron., 2015, 102, (3), pp. 433–443

[20] Barik R.K., Pradhan M., 'Area-time efficient square architecture', AMSE J., Adv. D, 2015, 20, (1), pp. 21–34

[21] Barik R.K., Pradhan M., 'Efficient ASIC and FPGA implementation of cube architecture', IET Computer Digital Technology, 2017, 11, (1), pp. 43–49

[22] Meenu S Ravi, R H Khade and Ajit Saraf, "Design of Fast Floating Point Multiply Accumulate Unit using Ancient Mathematics for DSP Applications." European Journal of Advances in Engineering and Technology, pp.55-59, 2015.

[23] Ku. Shweta N. Yengade, Associate Prof. P. R. Indurkar, "Review On Design Of Low Power Multiply And Accumulate Unit Using Baugh-Wooley Based Multiplier." International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 02 Feb - 2017.

[24] G.Indira, G. Madhusudhana Rao, P. Jaya Babu, M. Ravi Kiran, "An Efficient Architecture of MAC Unit with LAW Multiplier." International Journal of Advanced Scientific Technologies in
Engineering and Management Sciences (IJASTEMS-ISSN: 2454-356X) Volume.2, Issue.11, November.2016.

[25] Devpura P, Paliwal A, High throughput Vedic multiplier using binary to excess-1 code converter, Int. Journal of Adv Research Electronics & Communication Engineering, 2016.

[26] Zain Shabbir, Anas Razzaq Ghumman & Shabbir Majeed Chaudhry 2015, A reduced-sp-D3Lsum adder-based high frequency 4x4 bit multiplier using Dadda Algorithm, Circuit system and signal processing, vol. 39, pp. 1-22.

[27] Jumbe S, Mandavgane RN, Khatri DM, Review of parallel polynomial multiplier based on FFT using Indian Vedic mathematics, Int. Journal Computer Applications, 2015.
[28] Mogre, SV & Bhalke, DG 2015, Implementation of High-Speed Matrix Multiplier using Vedic Mathematics on FPGA, Proceedings International Conference on Computing Communication Control and Automation, pp. 959-963.
[29] Jithin S, Prabhu E, "Parallel multiplier-accumulator unit based on Vedic Mathematics", ARPN journal of engineering and applied sciences, Volume 10, No.8, pp.3608-3613,2015.
[30] Mittal, A., Nandi, A.: 'Design of 16-bit FIR filter using Vedic multiplier with carry save adder'. Proc. of 44th IRF Int. Conf., November 2015, pp. 57–60.