

Review Paper on FIR filter based on vedic multiplier and kogge stone adder

Piyush Gour¹ and Santosh Jha²

M. Tech. Scholar, LNCT Bhopal¹

Assistant Professor, LNCT Bhopal²

Abstract

Increase is a critical capacity in number-crunching activities. A CPU (focal handling unit) dedicates a lot of preparing time in performing number juggling tasks. Increase requires generously more hard-product assets and handling time than expansion and sub-footing. Computerized flag processors (DSPs) are the innovation that is ubiquitous in building Discipline. Quick duplication is essential in DSPs for advanced channel, convolution, Fourier changes and so on. In this proposed research work an attempt will make for making a novel FIR filter using Vedic multiplier modified Kogge stone adder. The implementation FIR filter based on Vedic multiplier will have not only fast response but also having less number of component, area and path delay.

Keywords

Vedic multiplier, FIR filter, Modified kogge stone adder.

1.Introduction

Several attempts have, therefore, been made to develop dedicated and reconfigurable architectures for realization of FIR filter in application specific integrated circuits (ASIC) and field-programmable gate arrays (FPGA) platforms. Systolic designs represent an attractive architectural paradigm for efficient hardware implementation of computation-intensive DSP applications, being supported by the features like simplicity, regularity and modularity of structure [1-2]. In addition, they also possess significant potential to yield high-throughput rate by exploiting high-level of concurrency using pipelining or parallel processing or both [3]. To utilize the advantages of systolic processing, several algorithms and architectures have been suggested for systemization of FIR filters [4-5]. However, the multipliers in these structures require a large portion of the chip-area, and consequently enforce limitation on the maximum possible number of processing elements (PEs) that can be accommodated and the highest order of the filter that can be realized. The multiplier-less distributed arithmetic (DA)-based technique has gained substantial popularity, in recent years, for its high-throughput preparing ability and expanded consistency which brings about practical and zone time effective registering structures [6]. The principle activities required for DA-based calculation of internal item are a grouping of query table (LUT)

gets to took after by move collection tasks of the LUT yield. DA-based calculation is appropriate for FPGA acknowledgment, on the grounds that the LUT and in addition the shift-add operations, can be efficiently mapped to the LUT-based FPGA logic structures [7].

In FIR filtering, one of the convolving sequences is derived from the input samples while the other sequence is derived from the fixed impulse response coefficients of the filter. This behavior of the FIR filter makes it possible to use DA-based technique for memory-based realization. It yields faster output compared with the multiplier-accumulator-based designs because it stores the precomputed partial results in the memory elements [8], which can be read out and accumulated to obtain the de-sired result. The memory requirement of DA-based implementation for FIR filters, however, increases exponentially with the filter order. DA was first introducing [9] and was further developed [10] for efficient implementation of digital filters. Attempts are made to use offset-binary coding [11] to reduce the ROM size by a factor of 2. An LUT-less adder-based DA approach has been suggested by Yoo and Anderson, where memory-space is reduced at the cost of additional adders [12]. Memory-partitioning and the multiple memory-bank approach along with flexible multibit data-access mechanisms are suggested for FIR filtering and inner-product.

Multiplication is an important fundamental function in arithmetic operations. Multiplication-based operations such as Multiply and Accumulate (MAC) unit and inner products are some of the frequently used Computation- Intensive Arithmetic Functions currently implemented in many Digital Signal Processing (DSP) applications, for example, convolution, Fast Fourier Transform (FFT), channel circuits and in microchips in its number juggling and rationale unit (ALU). Since duplication rules the execution time of most DSP calculations, so there is a need of fast multiplier. Right now, duplication time is as yet the overwhelming element in deciding the direction process duration of a DSP chip.

In this work we have put into impact a rapid Vedic multiplier utilizing barrel shifter. The sutra was

actualized by altered plan of "Nikhilam Sutra" because of its component of lessening the quantity of fractional items. The barrel shifter is utilized at various levels of outlines to diminish the defer when contrasted with customary multipliers. The equipment execution of Vedic multiplier is utilizing barrel shifter adds to satisfactory change of the speed. In many DSP algorithms, the multiplier lies in the critical delay path and ultimately determines the performance of algorithm. The speed of multiplication operation is great importance in DSP as well as in general processor. In past multiplication was implemented with a sequence of addition, subtraction and shift operations. There have been many algorithms proposals to perform the multiplication, and each offering different advantages and having in terms of speed, circuit complexity, area and power consumption.

The multiplier is a fairly large block of a computing system. For multiplication algorithms performed in DSP applications latency and throughput are the two major concerns from delay perspective. Latency is the real delay of computing a function, a measure of how long the inputs to a device are stable is the final result available on outputs. Throughput is the measure of how many multiplications can be performed in a given period of time multiplier is not only a high delay block but also a major source of power dissipation. That's why if one also aims to minimize power consumption, it is of great interest to reduce the delay by using various delay optimizations.

Advanced multipliers are the center parts of all the computerized signal processors (DSPs) and the rate of the DSP is generally controlled by the velocity of its multipliers. Two most basic duplication calculations followed in the computerized equipment are exhibit increase calculation and Booth augmentation calculation. The calculation time taken by the exhibit multiplier is relatively less on the grounds that the halfway items are ascertained autonomously in parallel. The postponement connected with the exhibit multiplier is the time taken by the signs to spread through the entryways that shape the Multiplication cluster. Corner increase is another vital augmentation calculation. Extensive corner clusters are required for fast duplication and exponential operations which thus require expansive halfway aggregate and incomplete convey registers. Duplication of two n-bit operands utilizing a radix-4 corner recording multiplier requires roughly $n/(2m)$ clock cycles to create the minimum noteworthy

portion of the last item, where m is the quantity of Booth recorder snake stages. Hence, an extensive spread deferral is connected with this case.

2.Literature review

Deepak Kumar Patel et al. [1], Speed and area are now a day's one of the fundamental design issues in digital era. To increase speed, while doing the multiplication or addition operations, has always been a basic requirement of designing of advanced system and application. Carry Select Adder (CSA) is a fastest adder used in many processors to accomplish fast arithmetic function. Many different adder architecture designs have been developed to increase the efficiency of the adder. It is very commonly known that per second any processors performed millions of work functions in semiconductor industry. So when we do designing of multipliers, one of the main standards is performing speed that should be taken in the mind. In this paper, we propose a technique for designing of FIR filter using multiplier based on compressor and carry select adder. Performance of all adder designs is implemented for 16, 32 and 64 bit circuits. These structures are synthesized on Xilinx device family.

B. Madhu Latha et al. [2], a 8-bit Vedic multiplier is enhanced as far as transmission deferral when contrast and the additional unsurprising multipliers. We have utilized 8-bit barrel shifter which desires for stand out clock cycle for "n" measure of movements in our anticipated configuration. The course of action is executed and checked utilizing FPGA and ISE Simulator. The focal part was executed on Xilinx Spartan-6 family xc6s1x75T-3-fgg676 FPGA. The transmission deferral complexity was excerpted from the blend report and static timing report as well. The basic configuration may achieve engendering postponement of 6.781ns by method for barrel shifter in base determination module and multiplier.

A Murali et al. [3], execution of Vedic multiplier is upgraded for spread postponement when contrasted and other ordinary multiplier like exhibit multiplier, Braun multiplier, altered corner multiplier and Wallace tree multiplier. For math duplications diverse Vedic enlargement techniques are used. It has been discovered that Urdhva tiryakbhyam Sutra is most gainful Sutra, giving slightest concede for increment of an extensive variety of numbers, either little numbers or immense numbers. In our arrangement we have utilized 8-bit barrel shifter which requires emerge clock for "n" number of developments. The arrangement is completed and

affirmed using FPGA and Mentor Graphics Simulators. The middle was realized on Xilinx Spartan-3E group. The causing deferment examination was isolated from the amalgamation report and static planning report too. The design could achieve expansion deferral of 6.771ns using barrel shifter as a section of base choice module and multiplier.

Mrs. Toni J. Billore et al. [4], this paper portrays the usage of a 8-bit Vedic multiplier utilizing quick viper improved as a part of terms of proliferation postponement when contrasted and ordinary multiplier. In our outline of 8 bit Vedic multiplier utilizing quick snake, we have used 8-bit barrel shifter which requires one and only clock cycle for "n" number of movements. The configuration of 8 bit Vedic multiplier utilizing barrel shifter is executed and confirmed utilizing FPGA and ISE Simulator. The center utilized here was actualized on Altera Cyclone® II 2C20 FPGA gadget programming. The proliferation postponement between 8 bit Vedic multiplier utilizing barrel shifter utilizing barrel shifter and utilizing quick snake examination was removed from the union report and static timing report too. The configuration which is executed here could accomplish spread deferral of 6.781ns utilizing barrel shifter obstruct as a part of base determination module and multiplier of building design utilized. In our undertaking, we make a correlation between execution investigation of 8 bit Vedic multiplier utilizing barrel shifter and utilizing quick viper.

Pavan Kumar et al. [5], This paper describes the implementation of an 8-bit Vedic multiplier enhanced in terms of propagation delay when compared with conventional multiplier like array multiplier, Braun multiplier, modified booth multiplier and Wallace tree multiplier. In our design we have utilized 8-bit barrel shifter which requires only one clock cycle for 'n' number of shifts. The design is implemented and verified using FPGA and ISE Simulator. The core was implemented on Xilinx Spartan-6 family xc6s1x75T-3-fgg676 FPGA. The propagation delay comparison was extracted from the synthesis report and static timing report as well. The design could achieve propagation delay of 6.781ns using barrel shifter in base selection module and multiplier.

3. Vedic multiplier

As specified prior, Vedic Mathematics can be isolated into 16 unique sutras to perform scientific counts. Among these the Urdhwa Tiryakbhyam Sutra is one of the most exceedingly favored calculations

for performing increase. The calculation is sufficiently able to be employed for the duplication of whole numbers and also binary numbers. The expression "Urdhwa Tiryakbhyam" started from 2 Sanskrit words Urdhwa and Tiryakbhyam which mean "vertically" and "transversely" respectively. It depends on a novel idea through which the era of every single fractional item should be possible with the simultaneous expansion of these halfway items. The calculation can be summed up for n x n bit number. Since the incomplete items and their totals are figured in parallel, the multiplier is free of the clock recurrence of the processor. In this way the multiplier will require the same measure of time to figure the item and henceforth is free of the clock recurrence.

The net advantage is that it reduces the need of microprocessors to operate at increasingly high clock frequencies. While a higher clock frequency generally results in increased processing power, its disadvantage is that it also increases power dissipation which results in higher device operating temperatures. The processing power of multiplier can easily be increased by increasing the input and output data bus widths since it has a quite a regular structure. Due to its regular structure, it can be easily layout in a silicon chip. The Multiplier has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other multipliers. Therefore it is time, space and power efficient.

To illustrate this multiplication scheme, let us consider the multiplication of two decimal numbers (14×12).

Example-14×12

- The right hand most digit of the multiplicand, the first number (14) i.e., 4 is multiplied by the right hand most digit of the multiplier, the second number (12) i.e., 2. The product $4 \times 2 = 8$ forms the right hand most part of the answer.

$$\begin{array}{r} 1 \quad \blacktriangledown \quad 4 \\ 1 \quad 2 \\ \hline 8 \end{array}$$

- Now, diagonally multiply the first digit of the multiplicand (14) i.e., 4 and second digit of the multiplier (12) i.e., 1 (answer $4 \times 1 = 4$); then multiply the second digit of the multiplicand i.e., 1 and first digit of the multiplier i.e., 2 (answer $1 \times 2 = 2$); add these two i.e., $4 + 2 = 6$. It gives the

next, i.e., second digit of the answer. Hence second digit of the answer is 6.

$$\begin{array}{r} 1 \quad 4 \\ 1 \quad 2 \\ \hline 6 \quad 8 \end{array}$$

- Now, multiply the second digit of the multiplicand i.e., 1 and second digit of the multiplier i.e., 1 vertically, i.e., $1 \times 1 = 1$. It gives the left hand most part of the answer. Thus the answer is 16 8.

$$\begin{array}{r} \blacktriangledown \quad 1 \quad 4 \\ \quad 1 \quad 2 \\ \hline \quad 1 \quad 6 \quad 8 \end{array}$$

- Thus the answer is 16 8.

4. Vedic multiplier using ks adder

The duplication of two numbers is finished by utilizing Urdhwa Triyakbhyam. Here first the minimum noteworthy bits of the two digits are duplicated. At that point the moderate digits are cross multi-employed and included. After this the most noteworthy digits are increased. For the 16X16 piece increase little square of 2X2 or 4X4 or 8X8 multiplier were utilized as a part of parallel to make the procedure simple and effective.

In our proposed technique the rapid convey select snake is supplanted by the convey select viper alongside Kogge Stone (KS) snake which cases to give a superior speed and less spread postponement. Here we have used four multiplier of 8 bit to perform 16 bit multiplication. The method used is the addition of all partial product formed by the cross multiplication of one bit with another. The LSB bits of first multiplier P1 (7-0) gives the LSB bits Q (7-0) of the final output.

Another bits of first multiplier P1 (15-8) are added in series with LSB 8 bits of second multiplier to form the 16 bits, which in turn get added with 16 bits of third multiplier by using KS Adder. The LSB bits of the output of KS adder forms the Q (15-8) bits of the final output. The remaining 8 bit P2(15-8) is then added with the left 8 bits of KS output to form 16 bits, which is then added with 16 bits of the fourth multiplier by using KS 2 adder. The output from KS 2 adder forms the Q (31-16) bits. This is how the 32bit output is achieved in the less possible time.

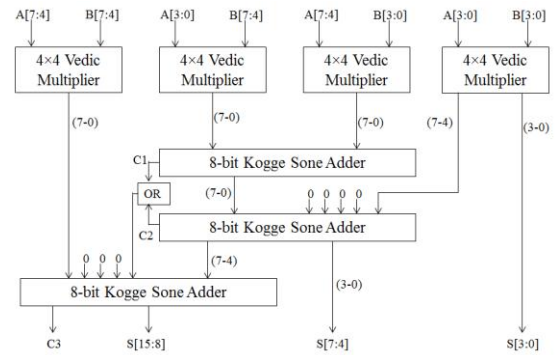


Figure 1 Logic diagram of vedic multiplier using kogge stone adder

5. Simulation result

All the designing and experiment regarding algorithm that we have mentioned in this paper is being developed on Xilinx 14.1i updated version. Xilinx 9.2i has couple of the striking features such as low memory requirement, fast debugging, and low cost. The latest release of ISE™ (Integrated Software Environment) design tool provides the low memory requirement approximate 27 percentage low. ISE 14.1i that provides advanced tools like smart compile technology with better usage of their computing hardware provides faster timing closure and higher quality of results for a better time to designing solution. ISE 14.1i Xilinx tools permits greater flexibility for designs which leverage embedded processors. The ISE 14.1i Design suite is accompanied by the release of chip scope Pro™ 14.1i debug and verification software. By the aid of that software we debug the program easily. Also included is the newest release of the chip scope Pro Serial IO Tool kit, providing simplified debugging of high-speed serial IO designs for Virtex-4 FX and Virtex-5 LXT and SXT FPGAs. With the help of this tool we can develop in the area of communication as well as in the area of signal processing and VLSI low power designing. To simplify multi rate DSP and DHT designs with a large number of clocks typically found in wireless and video applications, ISE 14.1i software features breakthrough advancements in place and route and clock algorithm offering up to a 15 percent performance advantage. Xilinx 14.1i Provides the low memory requirement while providing expanded support for Microsoft windows Vista, Microsoft Windows XP x64, and Red Hat Enterprise WS 5.0 32-bit operating systems.

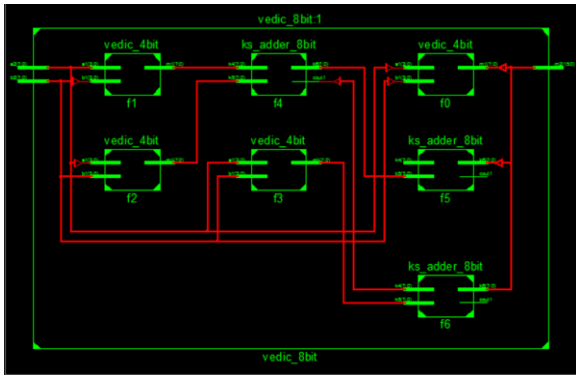


Figure 2 RTL View of 8-bit Vedic Multiplier

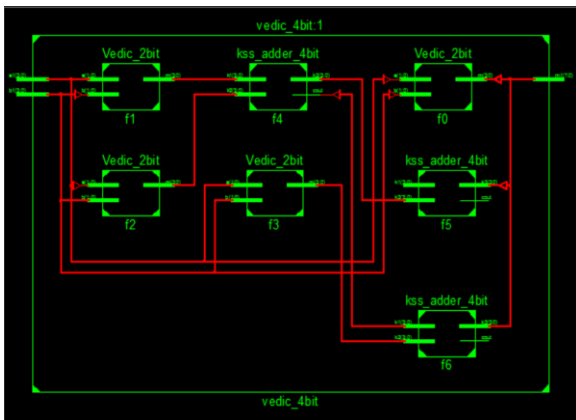


Figure 3 RTL View of 4-bit Vedic Multiplier

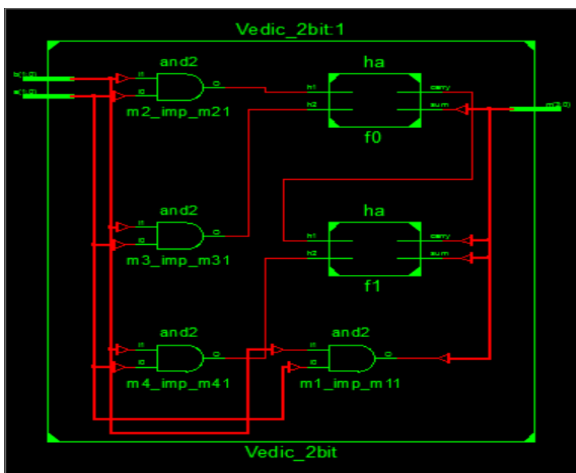


Figure 4 RTL View of 2-bit Vedic Multiplier

6. Conclusion

The high speed implementation of such a multiplier has wide range of applications in image processing, arithmetic logic unit and VLSI signal processing. The proposed 8x8 Vedic multiplier architecture has been designed and synthesized using on Xilinx software. The proposed Vedic Multiplier with modified Kogge

stone adder is compared with the existing Vedic multiplier using Carry select adder along with Common Boolean Logic and can be inferred that proposed architecture is faster compared to existing Vedic multiplier. In future the proposed multiplier performance parameters can be improved by high level pipelining operations and applied in signal processing applications like image processing and video processing

References

- [1] Patel DK, Chouksey R, Saxena M. Design of fast FIR filter using compressor and Carry Select Adder. In signal processing and integrated networks (SPIN), 3rd international conference on 2016 (pp. 460-5). IEEE.
- [2] Gokhale GR, Bahirgonde PD. Design of Vedic-multiplier using area-efficient Carry Select Adder. In advances in computing, communications and informatics (ICACCI), International conference on 2015 (pp. 576-81). IEEE.
- [3] Gokhale GR, Gokhale SR. Design of area and delay efficient Vedic multiplier using Carry Select Adder. In information processing (ICIP), 2015 International conference on 2015 (pp. 295-300). IEEE.
- [4] Kumar UP, Goud AS, Radhika A. FPGA Implementation of high speed 8-bit Vedic multiplier using barrel shifter. In energy efficient technologies for sustainability (ICEETS), International conference on 2013 (pp. 14-7). IEEE.
- [5] KUMAR AA, BASHA SM. Design and implementation of high speed 8 bit Vedic multiplier on FPGA.
- [6] Murali A, Padma GV, Saritha T. An optimized implementation of vedic multiplier using barrel shifter in FPGA technology. Journal of Innovative Engineering. 2014; 2(2):1.
- [7] Khatri S, Jangid G. FPGA Implementation of 64-bit Fast multiplier using barrel shifter. International Journal for Research in Applied Science and Engineering Technology. 2014; 2:344-8.
- [8] Billore TJ, Rotake DR. FPGA implementation of high speed 8 bit Vedic Multiplier using Fast adders. Journal of VLSI and Signal Processing. 2014; 4(3):54-9.
- [9] Kerur SS, Prakash Narchi JC, Kittur HM, Girish VA. Implementation of Vedic multiplier for digital signal processing. In international conference on VLSI, communication & instrumentation (ICVCI) 2011 (pp. 1-6).
- [10] Jindal V, Rizvi MN, Singh DK. VHDL Code of Vedic Multiplier with Minimum Delay Architecture. In national conference on synergetic trends in engineering and technology (STET-2014) international journal of engineering and technical research (pp. 2321-0869).
- [11] Marul BD, Darvadiya A. VHDL Implementation of 8-Bit Vedic Multiplier Using Barrel Shifter. International Journal for Scientific Research & Development. 2014; 2(01):2321-0613.