ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 5, Issue 4, April 2016

Review Paper on an Efficient Processing by Linear Convolution using Vedic Mathematics

Taruna Patil, Dr. Vineeta Saxena Nigam
Electronics & Communication Dept.
UIT, RGPV, Bhopal

***Abstract*— In this Technical era the high speed and low area of VLSI chip are very- very essential factors. Day by day number of transistors and other active and passive elements are drastically growing on VLSI chip. All the processors of the devices adders and multipliers are played an important role. Adder is a striking element for the designing of fast multiplier. Ultimately here need a fast adder for high bit addition. In this paper, the review of linear convolution is based on different types of adder. Proposing Kogge stone adder provides less components, less path delay and better speed compare to other existing Kogge Stone adder and other adders. Here we are comparing the Kogge Stone adders of different-different word size from other adders. The design and experiment can be done by the aid of Xilinx 14.2i Spartan 3E device family.**

Keywords: - Kogge Stone Adder, Ripple Carry Adder Linear Convolution, 14.2i Spartan 3 Device Family

I. Introduction

The processors speed mostly depends on adder design techniques. Adder is the device by which two or more than two bit information can be added. For the high speed processing of the data transfer area must be less of the passive and active element. Adder has two outputs specially sum and carry. For making fast adder carry can be reduced and replaced in different ways. The propagation delay or gate delay of a gate is basically the time interval between the application of the input pulse and the occurrence of the resulting output pulse. The propagation delay is a very important characteristic of logic circuits because it limits the speed at which they can operate. The shorter the propagation delay, the higher the speed of the circuit and vice-versa. Propagation delay should be minimizing as possible as, for high efficient addition. For instance 4 bit addition generally propagation delay is occurred highly. When we add one high bit to another high bit carry is occurred due to normally addition operation. This carry propagates to next bit and now bit addition is performed by 3 bit adder. So carry will propagate to the next bit over and over, this cause propagation delay will be occurred. As we have concerned (see in figure 1) (A_3, A_2, A_1, A_0) bits are added with (B_3, B_2, B_1, B_0) then carry propagation delay bits are occurred notations as(C_2, C_1, C_0). On the other hand propagation delay can be reduced by the aid of suitable structural designing process. For instance full adder can be designed with one XOR gate, three AND gate and one OR gate. That type of designing will provide 8.326 ns propagation delay. On the other hand full adder can be design by using two half adder and one OR gate. This type of designing will provide only 8.036 ns propagation delay. Carry propagation delay can be reduced by using ripple carry adder, fast adder that is also called look a-head carry generator, parallel adder, and specially Kogge stone adder.

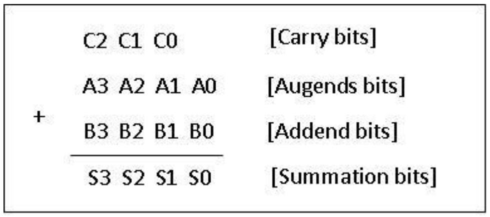

Fig 1: A Propagation delay for four bit

Now a days, time required in multiplication process is still the dominant factor in determining the instruction cycle time of a DSP chip [3]. Traditionally shift and add algorithm is being used for designing. However this is not suitable for VLSI implementation and also from delay point of view. Some of the important algorithms proposed in literature for VLSI implementable fast multiplication are Booth multiplier, array multiplier and Wallace tree multiplier [4]. Although these multiplication techniques have been effective over conventional"shift and add" technique but their disadvantage of time consumption has not been completely removed. Vedic Mathematics provides unique solution for this problem. The Urdhva Triyagbhyam Sutra or vertically and Crosswise Algorithm for multiplication is discussed and then used to develop digital multiplier architecture. For division, different division algorithms are studied, by comparing drawbacks and advantages of each algorithm; Nikhilam Algorithm based on vedic mathematics is modified according to need and then used.

II. Literature Review

Rashmi K. Lomte et al., (2011, [1]), Convolution and Deconvolution has many applications in digital signal processing. Multipliers and dividers are basic blocks in convolution and deconvolution implementation. They consumes much of time. With advances in technology, many researchers have tried and are trying to design multipliers and dividers which offer either of the following- high speed, low power consumption, regularity of layout and hence less area or even combination of them in multiplier and divider. In this paper, direct method is used to find convolution and deconvolution. Discrete linear convolution of two finite length sequences using Urdhva Triyagbhyam algorithm is presented here. Same algorithm is also used for deconvolution to improve speed. This design approach efficiently and accurately speeds up computation without compromising with area.

Prof J M Rudagi et al., (2011, [2]), Multiplication is an important fundamental function in arithmetic operations. Multiplication-based operations such as Multiply and Accumulate(MAC) and inner product are among some of the frequently used computation Intensive Arithmetic Functions(CIAF) currently implemented in many Digital Signal Processing (DSP) applications such as convolution, Fast Fourier Transform(FFT), filtering and in microprocessors in its arithmetic and logic unit . Since multiplication dominates the execution time of most DSP algorithms, so there is a need of high speed multiplier. Currently, multiplication time is still the dominant factor in determining the instruction cycle time of a DSP chip.

Akhalesh K. Itawadiya et al. (2013, [3]), Digital Signal Processing (DSP) operations are very important part of engineering as well as medical discipline. Designing of DSP operations have many approaches. For the designing of DSP operations, multiplication is play important role to perform signal processing operations such as Convolution and Correlation. The new approach of this implementation is mentally and easy to calculate of DSP operations for small length of sequences. In this paper a fast method for DSP operations based on ancient Vedic mathematics is contemplated.

Surabhi Jain et al. (2014, [4]), In Digital Signal Processing, the convolution and deconvolution with a very long sequence is ubiquitous in many application areas. The basic blocks in convolution and deconvolution implementation are multiplier and divider. They consume much of time. This paper presents a direct method of computing the discrete linear convolution, circular convolution and deconvolution. The approach is easy to learn because of the similarities to computing the multiplication of two numbers. The most significant aspect of the proposed method is the development of a multiplier and divider architecture based on Ancient Indian Vedic Mathematics sutras Urdhvatriyagbhyam and Nikhilam algorithm. The results show that the implementation of linear convolution and circular convolution using vedic mathematics is efficient in terms of area and speed compared to their implementation using conventional multiplier & divider architectures. The coding is done in VHDL. Simulation and Synthesis are performed using Xilinx ISE design suit 14.2

III. DIFFERENT TYPES OF ADDER

Ripple carry is a combinational circuit for adding more than two bit information. It is also called parallel adder. Ripple carry adder can be designed by using full adder in cascading form. Carry output of first full adder is connected with input of the next full adder, so carry is rippled from one adder to another adder. That is by it is called ripple-carry adder. Let us take example, for designing n bit RCA inputs are $(A_n \dots A_3, A_2, A_1, A_0)$ and $(B_n \dots B_3, B_2, B_1, B_0)$ then carry bits $(C_n \dots C_3, C_2, C_1)$ and summation bits are$(C_{out} \dots S_3, S_2, S_1, S_0)$.

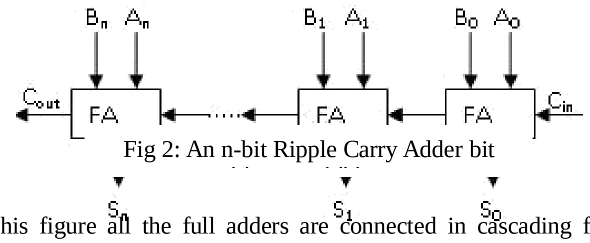

Fig 2: An n-bit Ripple Carry Adder bit

In this figure all the full adders are connected in cascading form. Carry input C_{in} is an extra input which has fixed value0. First full adder gives the carry output C_1 and summation outputS_0. Carry output of the first full adder is connected with second cascading full adder which will be considered as an input bit.

$$S_0 = (A_0 \oplus B_0) \oplus C_{in} \quad (1)$$

$$S_1 = (A_1 \oplus B_1) \oplus C_1 \quad (2)$$

$$C_{out} = (A_n . B_n) + (C_n . B_n) + (A_n . C_n) \quad (3)$$

- **KOGGE STONE ADDER**

Kogge Stone Adder was proposed by Peter M. Kogge and Harold S. Stone. Kogge Stone Adder is an advanced technology of Look a-head Carry Adder. That is also called parallel prefix adder. It has more area than to Brent Kung Adder but less Fan-out. This adder provides the carry signal time $(O_{logn})$and become fastest adder for industrial level.

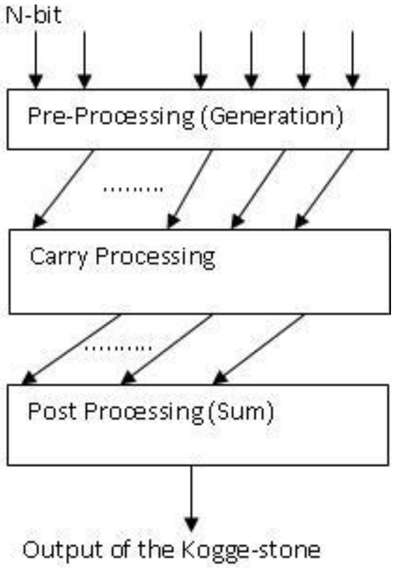

Fig 3: A Block Structure of Kogge Stone

First block of KSA is Pre- Processing that will generate and propagate the carry. Processing of carry will be done over the carry processing area and all the carry signal go through the post processing block. In the pre preprocessing stage we find the, generate and propagate signals from each inputs.

$$P_n = A_n \oplus B_n \qquad (4)$$
$$G_n = A_n . B_n \qquad (5)$$

Carry processing stage provides the carries corresponding to each bit. Execution of these bit operation is carried out from parallel. After finding the carries in parallel they are segmented in to smaller pieces.

$$CP_{n-1} = P_{n-1} \oplus P_n \qquad (6)$$
$$CG_{n-1} = (P_n \oplus G_{n-1}) + G_n \qquad (7)$$

Bottom block is summation block which provides the summation bits. That blocks are comprised with XOR gate. If one input is different from another then output will be high. And if inputs are same then outputs will be low. Kogge Stone provides the less area than to other parallel adder like carry select adder, carry save adder and look ahead adder.

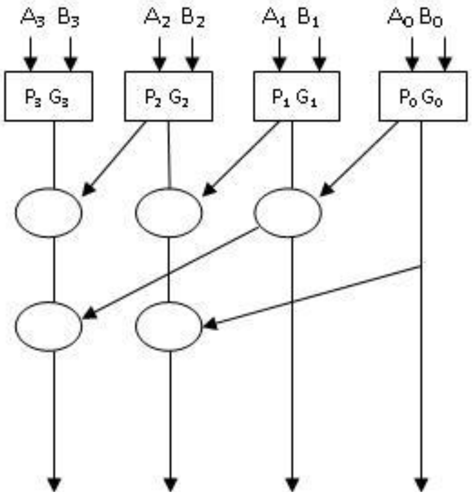

Fig 4: A Functional Diagram of Kogge Stone Adder

Above diagram is a functional diagram of Kogge Stone adder for 4 bit addition. Here elliptically symbol defined as a carry processing stage. The output of the preprocessing stage is fed to next carry stage and post processing as well.

- **MODIFIED KOGGE STONE ADDER**

The main object of this paper is to reduce the route delay and logic delay. As soon as we increase the bit for addition in Kogge Stone adder area will be increased. So, area and propagation delay can be reduced by the aid of modified KS adder. This adder will be designed like as ripple carry adder. Carry output of one KS adder is connected with another KS adder but this method is very beneficiary for high efficient digital devices as per concerning propagation delay.

$$P = A_i \oplus B_i \qquad (8)$$
$$G = A_i . B_i \qquad (9)$$
$$C_i = G_i \qquad (10)$$
$$S_i = P_i \oplus C_{i-1} \qquad (11)$$

Generally, 2 bit KS adder is comprised with two half adder, two XOR gate. Area of the any circuit is played an important role. Area can be calculated with the help of number of primary gates. Primary gates are AND, OR and NOT. For this type of designing one thing has to be kept in mind C_{in} must be assigned with 0 bit.

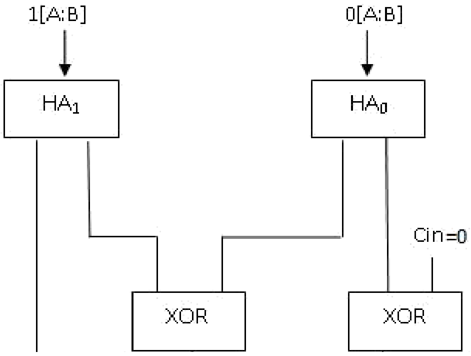

Fig 5: A Modified 2 bit KS Adder

IV. 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 beemployed for the duplication of whole numbers and also binarynumbers. The expression "Urdhwa Tiryakbhyam" started from 2Sanskrit 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 X 2 = 8 forms the right hand most part of the answer.

```
 1  4
 1  2
 ----
    8
```

- 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 X 1=4); then multiply the second digit of the multiplicand i.e., 1 and first digit of the multiplier i.e., 2 (answer 1 X 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.

```
 1      4
 1      2
 ---------
 6      8
```

- Now, multiply the second digit of the multiplicand i.e., 1 and second digit of the multiplier i.e., 1 vertically, i.e., 1 X 1 = 1. It gives the left hand most part of the answer.

```
   1      4
   1      2
 ----------
   1   6  8
```

- Thus the answer is 16 8.

V. LINEAR CONVOLUTION

Complex logical designing can be reduced by the array mathematics calculation which is consisting with 16 sutras. Number of fan in, fan out pin and input output buffers can be minimized by using these array mathematics sutras. For the high speed convolution, multiplier and adder must be high efficient and low area as possible as. For instance (*A*3, *A*2, *A*1, *A*0) and (*B*3, *B*2, *B*1, *B*0) are the finite length sequence.

For the appropriate output we can use the 4 bit array multiplier, 8 and 9 bit ripple carry adder. Multiplication of convolution input sequence is different from ordinary binary multiplication.

VI. SIMULATION ANALYSIS

Simulation of these experiments can be done by using Xilinx 14.2I VHDL tool. In this paper we are focusing on propagation delay. Propagation delay must be less for better performance of digital circuit. Xilinx is an analysis and simulation tools which has many application in research filed. In this tool simulation is divided in to three categories, model, behavioral and structural. Xilinx 14.2i is an updated version which has many merits than other version.

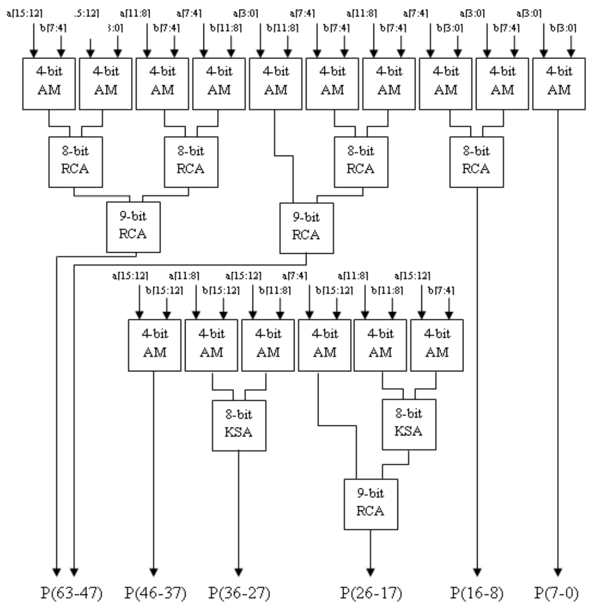

Figure 6: Linear Convolution

Table 1: Device Summary

Bit size	Adder	Area	Delay in ns
8 bit SQRT CSLA	Regular(RCA)	144	11.92
	BEC	132	13.69
	Modified (CBL)	111	11.15
	KS	83	5.776
16 bit SQRT CSLA	Regular (RCA)	348	16.15
	BEC	291	18.77
	Modified (CBL)	276	15.48
	KS	166	10.85
32 bit SQRT CSLA	Regular (RCA)	698	28.97
	BEC	762	34.44
	Modified (CBL)	552	26.23
	KS	332	20.56
64 bit SQRT CSLA	Regular (RCA)	1592	52.82
	BEC	1498	64.61
	Modified (CBL)	1104	47.74
	KS	664	40.25

VII. CONCLUSION

Conclusion of this paper is that, designed a low power and less area or minimum propagation delays Kogge Stone Adder. According above table (see Table 1) ripple carry adder and other parallel adder has more number of slices than to KSA. Proposing high efficient KS adder can be used for multiplication to design high speed processor. Apart from that it can be used in high speed convolution and deconvolution methods all the experiment has done in Spartan 3E, Xilinx 14.2I VHDL package.

VIII. FUTURE SCOPE

Now a day's all the devices need a design with compact and high speed portable components. KS adder can used to design a fast multiplier and multiplier is an important device for high speed processor. These devices can be used in high efficient convolution and de-convolution, FIR filter, ALU etc.

REFERENCES

[1] Jain, S. ; Saini S. "High Speed Convolution and Deconvolution algorithm (Based on Ancient Indian Vedic Mathematics) electrical engineering/electronics, computer, telecommunication and information technology (ecti-con), 2014 11th international conference on doi: 10.1109/ ecticon.2014.6839756 Publication Year: 2014 , Page(s): 1 – 5.IEEE 2014.

[2] Lomte, Rashmi K., and P. C. Bhaskar. "High Speed Convolution and Deconvolution Using Urdhva Triyagbhyam." VLSI (ISVLSI), 2011 IEEE Computer Society Annual Symposium on. IEEE, 2011.

[3] Itawadiya, Akhalesh K., et al. "Design a DSP operations using vedic mathematics." Communications and Signal Processing (ICCSP), 2013 International Conference on. IEEE, 2013.

[4] L. Sriraman, T.N. Prabakar, "Design and Implementation of Two Variable Multiplier Using KCM and Vedic Mathematics," 1st Int. Conf. on Recent Advances in Information Technology, Dhanbad, India, 2012, IEEE Proc., pp. 782-787.

[5] Soma BhanuTej, 2012.Vedic Algorithms to develop green chips for future, International Journal of Systems, Algorithms and Applications, Volume 2, Issue ICAEM12, ISSN Online: 22772677..

[6] Youngjoon Kim and Lee-Sup Kim, 2001.A low power carry select adder with reduced area,IEEE International Symposium on Circuits and Systems, vol.4, pp.218-221. Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[7] Akhilesh Tyagi,1990.A Reduced Area Scheme for Carry- Select Adders,IEEE International Conference on Computer design, pp.255-258..

[8] Belle W.Y.Wei and Clark D.Thompson,1990.Area-Time Optimal Adder Design, IEEE transactions on Computers, vol.39, pp. 666675.

[9] Y. Choi, 2005.Parallel Prefix Adder Design, Proc. 17th IEEE Symposium on Computer Arithmetic, pp 90-98.