HIGH SPEED COMPLEX VEDIC MULTIPLIER USING HYBRID SQUARE KOGGE STONE ADDER TECHNIQUE

Suraj Chaudhary, M-Tech Scholar, Department of ECE, Rabindranath TAGORE University, Bhopal, csuraj363@gmail.com,
Jitendra Ahir, Assistant Professor, Department of ECE, Rabindranath TAGORE University, Bhopal, jitendraahir888@gmail.com

ABSTRACT—The built architecture for the complex Vedic multiplier is presented in this work by rectifying the problems in the current method and improving speed by using the kogge stone adder with the aid of hybrid square technique. In almost all modern processing units, the addition is a crucial timing procedure. Performance parameters such as the region of implementation, the latency of the adder and the dissipation of power determine the option of adders for various applications. Therefore the design of higher speed and less complicated adder architectures with lower dissipation of power is an extensive research priority. Kogge stone adders are the most frequently used of the several adder topologies available, as they give several design choices to achieve area/power/delay efficiency and they also provide trade-off optimization. The preferred option of the last few decades is the High Speed Complex Vedic Multiplier using Hybrid Square Kogge Stone Adder design and implementation of area-power optimised hybrid parallel-prefix Ling Adder. For even-indexed and Kogge-Stone structures for odd-indexed bits, the hybrid adder topology used in this work uses the Ladner-Fischer approach. The independent measurement of odd and even bits contributes directly to the reduction of the prefix tree fan-out and thus to a reduced delay. By the measurement of the individual carries using updated Ling's equations, the area efficiency is achieved. Based on modified Ling equations using 0.18µm CMOS technology, the proposed adders are implemented with 16 bit and 32 bit word size. The synthesis findings show that up to 26 percent and 36 percent saving of area-power product and power-delay product respectively could be achieved by the proposed adders over the adders based on traditional Ling equations.

Keywords— Kogge Stone Adder (KSA), Vedic Multiplier (VM)

I. INTRODUCTION

In microprocessors, digital signal processors (DSP) and data processing in application-specific integrated circuits, binary addition is one of the most widely used operations (ASIC). The binary adders therefore constitute, to name a few, the basic building blocks of the arithmetic and logic units (ALU), address generation units (AGU) and floating-point blocks. Usually, the high performance large adders use a parallel prefix tree to calculate the generation of the group and the group propagate signals to calculate the carries and the final sum pieces. The Parallel Prefix Adder (PPA) designs are therefore the most preferable for their higher speed of operation among the many adder designs presented in the literature. In the last few decades, various addition algorithms have been proposed aiming at improving the computational efficiency of PPAs by optimizing one or more of the parameters, namely, speed, power, area and regularity of carry graphs. Several parallel prefix adder topologies have been published in the literature, and they also present the comparisons among the parallel tree adders. Different design parameters such as the delay, fan-out, wiring complexity, regularity and the area required for implementation have been used to describe the comparative benefits of various adders. There are several other techniques proposed for the carry computation in the parallel prefix trees. Sklansky (1960) proposes tree-prefix algorithms for adders, wherein a tree structure is used to compute the intermediate signalsThe main problem of the scheme is the large number of gates and the long lateral wires required between the consecutive stages, which increases power dissipation. The minimum depth prefix graph has been introduced by Ladner and Fischer (1980). As the longest lateral fanning wire extend from one node to n/2 other nodes, the capacitive fan-out load becomes larger for later levels in the carry graph. Additionally an appropriate number of buffering inverters are added to drive these large loads at the cost of slightly increased delay parameter. Brent and Kung (1982), propose the prefix-computation graph in an area-optimal way. Here, the lateral fan-out of each node is restricted to unity, similar to the Kogge -Stone graph, however, without using several long wires. In spite of the attractive topological structure, it incurs increased logical depth. Han and Carlson (1987) presented a new prefix tree, which is a hybrid of Brent-Kung and Kogge-Stone adders, with reduced number of computational nodes and slight increase in logical depth. Knowles (2001) has demonstrated how the various adder topologies influence the fan- out and the wiring density, thus, influencing the design decisions and yielding to better area/power trade-offs. Some adders, instead of being fully parallel, use sparseness to reduce the impact of lateral fan-out. Several sparse tree implementations have been published with sparseness of two and four. Sanu Mathew et al (2001) and S. Kao et al (2006) present the prefix tree with sparseness of two. Naffziger (1996) and Shimazaki et al (2004), implements 64 bit adders using sparse four trees to decrease the latency. Ling (1981) has proposed different carry generation equations where one propagate term is factored out to simplify the group generate function, thereby making the first level of prefix tree simpler, which further reduces the critical path delay. Dimitrakopoulos and Nikolos (2005) propose an approach which can save one logic level of implementation compared to the parallel -prefix structures proposed for the traditional definition of carry look-ahead equations and the structure reduces the fan-out requirements of the design. This paper discusses the design and implementation of Kogge and Stone (1973), in their scheme uses the recursive doubling property and the prefix trees were characterized by their minimum logic depth, regular structure, and unity fan -out and they are used when very high performance is needed. The proposed adder employs two tree architectures, Kogge-Stone on odd-numbered bits and Ladner-Fischer on even-numbered bits as optional for user. It uses modified Ling equations to reduce the complexity of the generate function at the first level. Furthermore, as the real carries for higher order bits are computed from the lower order Ling carries, a significant area saving has been achieved along with reduced delay and power. The area, delay and power parameters are computed and compared among the adders for two different word sizes to prove the area-power and power-delay efficiency of the proposed adders.

The rest of the paper is structured as follows: Section 2 revisits the basics of kogge stone addition in section 3. The proposed area-power efficient hybrid prefix adders based on modified Ling equations are presented in Section 4. Section 5 discusses the simulation and synthesis results that validate the area efficiency of the proposed adders. Finally, Section 6 concludes the paper.

II. BACKGROUND

A. Vedic Multiplier

Vedic mathematics in the modern world is based on 16 aphorisms and 12 corollaries. These formulations were selected by Swami Bharati Krishna Tirtha from Atharva Ved (1884-1960). Thereafter, in these selected sutras and sub-sutras, the former Jagadguru Sankaracharya developed and introduced the techniques to change the concepts. For the purpose of multiplication, the Nikhilam Navatashcaramam Dashatah and Urdhva-Tiryagbhyam sutras are used among all these sutras and sub-sutras. When implemented for multiplication, these Vedic mathematical techniques showed very good results in terms of saving computational time. It is therefore concluded that the multiplier design integrated with Vedic mathematical techniques based on "Urdhvatriyagbhyam" (vertical and cross-wise algorithm) sutra[4] improved the speed of operation of multiplication. The methodology for 4x4Vedic mathematics is given below to clarify the procedure:

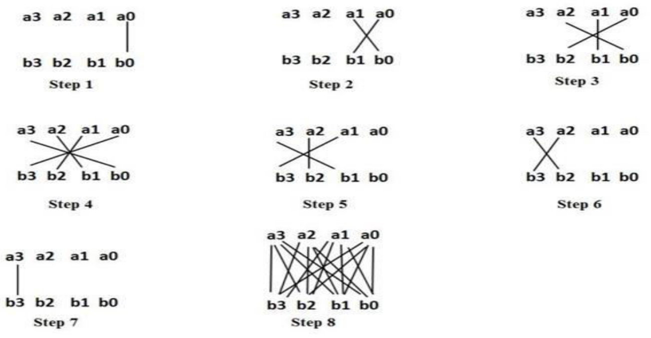

Figure 1. : Steps of vedic multiplication

B. Parallel prefix adders

Parallel prefix adders are most important because of the speed at which they operate. The sum of n-bit number can be computed in time O(log n)[1]. This reduction in time is achieved due to its use of a tree network known as prefix operation graph. The reduction in time helps in addition of wider word lengths. A block diagram for parallel prefix adder is shown in figure 2. Every parallel prefix adder can be designed using three stages as described in the figure 2.

The first stage is simple half adder. The core of the parallel prefix adder is the prefix graph that propagates the carry to the final stages. An example of the graph is show in figure 3 .In the prefix operation graph, each node is a basic logical circuit described as prefix operation. The goal of addition is to compute the sum, S, of two operands A and B, both of which are binary words of length n. For n-bit addition, the first stage of the adder computes the generate (G) and propagate (P) terms for each bit of the operands according to the following equations:

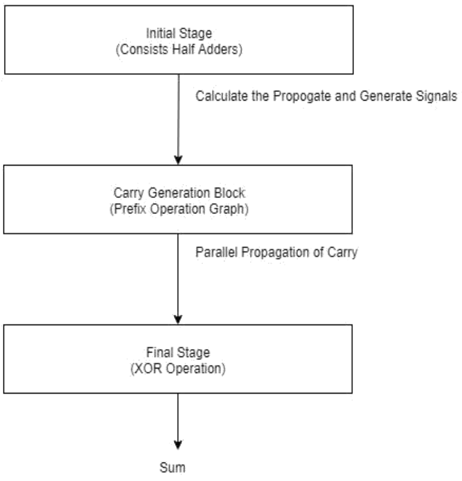

Figure 2: Parallel Prefix Block Diagram.

$$Gi = Ai \text{ AND } Bi$$

$$Pi = Ai \text{ XOR } Bi$$

Stage 2 consists of the basic prefix operation, pref, is defined as follows:

$$(Gi,Pi) \text{ pref } (Gj,Pj) = (Gi + Pi . Gj , Pi.Pj)$$

In the above equation, + refers to logical OR and. refers to logical AND. In the end, the carry is equal to Gi's and sum is calculated by XOR with initial propagate which is the final stage. We have designed two parallel prefix adders:

1. Kogge-Stone Adder
2. Han-Carlson Adder

P.M. Kogge and H.S. Stone were he first to use the property of commutativity and design parallel prefix adders where the computation of the prefixes is consid-ered to be a recurrence that can be performed in parallel[4]. The Kogge-Stone computation uses log2n stages, where n is the number of bits in the operands. Han-Carlson adder is a hybrid of Kogge-Stone and another parallel prefix adder I.e Brent-Kung. Kogge-Stone takes log2n stages and the Brent-Kung construction takes 2log2n-1 stages[3]. Han-Carlson adder (not discussed in details) takes less area for the combinational circuits as compared to Kogge-Stone design Each prefix tree consist of the Some basic building blocks such as prefix_op (Bigger Circle), square Box, Buffer and Diamond (Last stage XOR). Prefix tree graph for 16-bit Kogge stone is shown in figure 2.

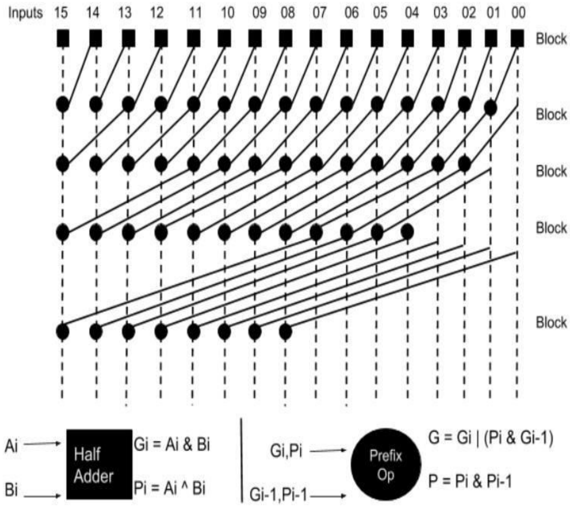

Figure 3: 16-bit Kogge Stone Prefix Graph

Table 1: Statistics: Kogge Stone Adder

Statistics: Kogge Stone Adder				
	8-bit	16-bit	32-bit	64-bit
Delay (ps)	31.87	38.49	49.07	56.01
Number of wires	13	18	17	21
Number of wire bits	97	236	504	1054
Number of wire bits	15	12	17	21
Number of public wire bits	93	206	494	1124
Number of memories	0	0	0	0
Number of memory bits	0	0	0	0
Number of processes	0	0	0	0
Number of cells	36	89	217	516
Buffer	8	15	32	64
Diamond	7	17	29	74
Pref_Operation	15	46	125	315
Square_Operation	9	17	36	65

Table 2: Statistics: Ripple Carry Adder

		Delay(ps)	Number of Cells
Ripple Carry Adder	8-bit	47.59	39
	16-bit	84.84	77
	32-bit	119.79	147
	64-bit	239.47	284
Carry Save Adder	8-bit	60.24	83
Kogge-Stone Adder	16-bit	87.63	185
	32-bit	125.05	446
	64-bit	246.20	882
	8-bit	30.97	57
Han-Carlson Adder	16-bit	38.45	164
	32-bit	47.07	454
	64-bit	56.01	1200
	16-bit	78.28	93
	32-bit	65.67	287

All of these discussed multipliers are using efficiently as per the application requirements. From the discussion above a table [Table 3] is shown below to represent the comparison among the multipliers by taking into the account of some very important parameters such as: time delay, power consumption, circuit complexity and area required for simple understanding.

Table 3. Results: Multiplier

Multiplier	Speed	Area	Power consumption
Array	Low	Small	Most
Booth	Low but better than Array	Small	Less than Array
modified booth	High	Medium	Less
wallace tree	Higher	Larger	More
modified booth Wallace	Highest	Largest	More
Vedic	Higher than Wallace tree	Larger than Wallace tree	More than Wallace tree

Multipliers with higher efficiency are extremely important to increase the performance of the current communication setup. The speed of computation was increased by quick multipliers, which also improved the efficiency of other digital applications such as image processing. Delayed time, power usage, complexity of circuits and area specifications are the key parameters that need to be taken care of in the case of multipliers. In terms of all of these performance parameters, all the multipliers mentioned above are efficient. The array multiplier is, of all the simplest because of its simple circuitry, which leads to less use of space. But with low velocity and maximum power consumption, this multiplier suffers. The fastest multiplier among all is the modified booth Wallace tree multiplier by taking the advantages of both multipliers: modified booth multiplier and Wallace multiplier. In this multiplier the number of partial products is minimized to either half or one by third of the number of multipliers bit by using radix 4 algorithm and radix-8 algorithms respectively.
The Wallace tree multiplier, where the overall speed of the accumulation increases due to using carry save adder (CSA) has occupies the largest area. Here by minimizing the number of partial products and examine more than one partial products at the same time, the speed is further enhanced and these techniques also make the system more accurate. One of the fastest and less power consuming multiplier is Vedic multipliers, which is based on the vedic mathematical formulations. It is proved by several researchers that the vedic multiplier reduces the delay time and power consumption by approximately 45% and 57% as compared to the array multiplier

III. GENRAL DISCUSSION

Kogge stone adder

Kogge stone adder [10] is a parallel prefix type of carry look forward adders. It comprises of four vertical stages, every vertical phase of Kogge stone adder creates an engender and produce bit. It is considered as the quickest adder and it is broadly utilized in businesses for superior of arithmetic circuits. In Kogge stone adder carrys are registered quick by processing them in parallel at the expense of expanded territory. Kogge stone adder is adder which is having low delay.

Vedic multiplier

The utilization of Vedic science [3], [4] lies in the way that it decreases the normal counts in customary arithmetic to straightforward ones. This is so because the Vedic formulae are professed to be founded on the regular standards on which the human personality works. Vedic Mathematics is a philosophy of number juggling decides that permit progressively effective speed usage.

IV. RESULT

In terms of area, capacity, delay and power delay products, the five different types of adders are compared. It is shown that even though there is an increase in area, the Kogge stone adder is considered the fastest adder. The energy for this adder (Power delay product) is also considered to be the smallest of the five comparable adders. In the following tables, the simulation results of five different adders for 4-,8-,16- bit adders are shown.

Table 4 : Different 4-bit adders comparison based on 45nm

Adders(4 bit)	Area	Power (nW)	Delay (nS)	PDP
Carry skip	8	2151.8	956	2051628.4
Carry save	9	2282.5	981	2235556.5
Carry lookahead	15	2416.23	734	1775196.05
Carry select	18	3121.4	985	3079835.8
Kogge stone	13	1934.38	608	11734468.28

Table 5 : Different 4-bit adders comparison based on 90 nm

	90nm			
Adders(4 bit)	Area	Power (nW)	Delay (nS)	PDP
Carry skip	79	5431.7	1132	6148684.4
Carry save	142	5549.1	1170	6492447
Carry lookahead	119	4920.17	802	3945976.34
Carry select	174	6214.9	1164	7234143.6
Kogge stone	117	3710.15	705	2615655.75

Table 6 : Different 4-bit adders comparison based on 90 nm

	90nm			
Adders(8bit)	Area	Power (nW)	Delay (nS)	PDP
Carry skip	369	15668.51	4417	69207786.6
Carry save	802	38386.15	1695	65064520.9
Carry lookahead	389	16926.73	3226	54605637.4
Carry select	602	28020.38	1510	42310767.8
Kogge stone	807	33277.94	1063	35374449.2

Table 7: Comparison of 8-bit Vedic multiplier with Proposed hybrid multiplier based on 45nm, 90nm and 180nm technology

	45nm			
8-bit	Area	Power (nW)	Delay(nS)	PDP
Vedic	269	36676.89	3828	140318127
proposed	286	37022	3468	128348350
	90nm			
Vedic	1847	73576.368	4191	308427939
proposed	1908	79772.47	3667	292195559
	180nm			
Vedic	5581	378889.01	5764	2180115887
proposed	6199	395075.34	4217	1664448212

The comparison of the 4-, 8- and 16- bit Vedic multiplier with the proposed hybrid Vedic multiplier for 45nm, 90nm and 180nm technology with kogge stone adder is shown in Table 4 to 7. When compared to the current Vedic multiplier, the proposed multiplier is quicker. It is also proven that the energy of the proposed multiplier is noteworthy. Tables also display delay comparison of the latest 45nm, 90nm and 180nm technology-based vedic multiplier with kogge stone adder proposed.

References

[1] V. G. Oklobdzija, B. R. Zeydel, and H. Q. Dao, "Comparison ofhigh-performance VLSI adders in the energy-delay space," in IEEETransactions on Very Large-Scale Integration (VLSI) Systems, vol. 13,no. 6, June 2005.

[2] Jasbir Kaur and Lalit Sood, "Comparison between various types of addertopologies," in IJCST, vol. 6, no. 1, Jan.–Mar. 2015.

[3] Maroju SaiKumar and Dr. P. Samundiswary, "Design and performanceanalysis of various adders using verilog," in International Journal ofComputer Science and Mobile Computing, IJCSMC, vol. 2, no. 9,pp. 128–138, Sep. 2013.

[4] Nagendra, C.; Irwin, M.J.; Owens, R.M.,"Area-time-powertradeoffs in parallel adders", Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on Volume 43, Issue10, Page(s): 689 – 702, 1996

[5] Min Cha and Earl E. Swartzlander, Jr, "Modified Carry SkipAdder for reducing first block delay", Proc. 43rd IEEE MidwestSymp. on Circuits and Systems, Lansing MI, Page(s): 346-348,2000

[6] Pak K. Chan, et al, Delay Optimization of Carry-Skip Adders andBlock Carry-Lookahead Adders Using Multidimensional DynamicProgramming, IEEE Transactions on Computers, vol. 41, No. 8,pp. 920-93, 1992

[7] Wang, Y.; Pai, C.; Song, X., "The design of hybrid carrylookahead/carry-selectadders, Circuits and Systems II: Analog andDigital Signal Processing, IEEE Transactions on Volume 49, pp.16-24, 2002.

[8] Jin-Fu Li, Jiunn-Der Yu, Yu-Jen Huang, "A Design Methodologyfor Hybrid Carry-Lookahead/Carry-SelectAdders withReconfigurability", IEEE, 2005

[9] Raminder Preet Pal Singh, Praveen Kumar, and BalwinderSingh,Performance Analysis of 32-Bit ArrayMultiplier with a Carry Save Adder and with a Carry Look Ahead Adder, Letters of International Journalof Recent Trends in Engineering, vol.2, no.6, pp. 83-89, Nov 2009.

[10] P. M. Kogge and H. S. Stone,"A parallel algorithm for the efficientsolution of a general class of recurrence equations,"IEEETransactions on computers, vol. C-22, no. 8, pp. 786–793, Aug. 1973.

[11] R Arun Sekar, Balaji G Naveen, A Gautami, B Sivasankari,"High efficient carry skip adder in various multiplier structures",

[12] Advances in Natural and Applied Sciences, Vol. 10, Issue 14, sep.2016, pp. 193-198.

[13] . T.Prabakaran, R.ArunSekar," High Performance Reversible Vedic Multiplier using Cadence 45nm Technology", International Journal of Research and Advanced Development, Vol. 02, Issue 04, pp. 64-71, Oct. 2018.

[14] B.RatnaRaju, D.V.Satish, "A High Speed 16*16 Multiplier Based On UrdhvaTiryakbhyam Sutra", International Journal of ScienceEngineering and Advance Technology, IJSEAT, Vol 1, Issue 5, Oct. 2013.

[15] KathiAnoosha, SasiKiran, "Design of High Speed Vedic Multiplier Using Carry Select Adder with Brent Kung Adder",International Journal & Magazine of Engineering, Technology, Management and Research, Volume 3, Issue 10, October 2016.

[16] 15. M. Kathirvelu and Manigandan, "Design of Area Optimized, Low power, High Speed Multiplier using Optimized PDP full adder", International Journal of Electrical Engg, ISSN 0974-2158, Volume 6, Number 2, pp. 173-185, June 2013