Efficient Serial Multiplier based on Ripple Counters

Vinoth.A and Deepa.R

Abstract- A design of serial-serial multiplier addresses the high data sampling rate. To reduce the partial product tree (PPT) height in order to reduce the delay in the multiplication process. The proposed technique is effectively considered the entire partial product matrix with n data sampling cycle for n×n multiplication function instead of 2n cycles and hence delay is reduced. The multiplication of partial product considering in two series input among which one is starting from LSB and from MSB. In proposed method number of computational cycles will be reduced. Here the 5:3 counters are replaced by asynchronous counters. So the critical path is limited to DFF and an AND gate. The proposed method occupies less silicon area compared to the conventional serial-serial multiplier

Index Terms-serial-serial multiplier, partial product, Ripple counter, Ripple carry adder (RCA), Carry save adder (CSA).

I. INTRODUCTION

Multipliers are the fundamental and essential building blocks of VLSI systems. The speed of multiply operation is of great importance in digital signal processing as well as in the general purpose processors.The design and implementation approaches of multipliers contribute substantially to the area, speed and power consumption of computational intensive VLSI systems. Hardware implementation of a multiplication operation consists of three stages, specifically the generation of partial products (PPs), the reduction of partial products (PPs), and the final carry propagation addition. Multiplication can be performed in two ways. One is serial multiplier and other is parallel multiplier. While the serial multiplier have two input operands and are given serially, whereas the parallel multipliers are the multiplier this has parallel operands. Serial multipliers are popular for their low area and power and are more suitable for bit-serial signal processing applications with I/O constraints and on-chip serial-link bus architectures. Multipliers are generally the slowest element in the system and the performance of the system or a chip is generally determined by the performance the multiplier. Often, the delay of multipliers dominates the critical path of these systems and due to issues concerning reliability and portability, power consumption is a critical criterion for applications that demand low-power as its primary metric.

Vinoth.A is a PG Scholar, Dept. of EEE, K.S.R. College of Engineering, Tiruchengode and Deepa.R is working as Assistant Professor, Dept. of EEE, K.S.R. College of Engineering, Tiruchengode, Email: Vinoth.mandi@gmail.com, infodeepaeee@gmail.com

While low power and high speed multiplier circuits are highly demanded, it is not always possible to achieve both criteria simultaneously. Therefore, a good multiplier design requires some trade-off between speed and power consumption The partial products are reduced by carry-save adders (CSAS)[2]using an array or tree structure. The number of partial products is reduced to two rows when Carry propagation addition is inevitable.

The partial product (PP) of tree height will be increased linearly with the word length of the multiplier it aggravates the area, delay and power dissipation of the two subsequent stages. A serial multiplier and a squarer with no latency cycles are presented in [4]. Algorithms for serial squarer's and Serial/serial multipliers were derived in [5] and Error Propagation in two's complement operation has been investigated to minimize the number of D flip-flops that need to be cleared between operations.

Unlike the CSAS architecture, the critical path is found only along the AND gates [12]. The partial product formation is done in n clock cycles instead of 2n clock cycles and hence delay is reduced. Thus, the numbers of computational cycles are reduced in the proposed method. Moreover, the counters change states only when input is '1', which leads to low switching power. Therefore, it is highly desirable to reduce the number of partial products before the CSA Stage. In the proposed method the partial product formation is revamped using an algorithm named as serial-serial algorithm which is explained in the following sections. The generated partial products are passed to a group of asynchronous 1's counters for accumulation. The counters will count the number of ones in the partial products which is used for addition. In the following sections an approach to the design of serial multiplier that is capable of processing input data without input buffering and with reduced total number of computational cycles is proposed. A serial-parallel multiplier loads one operand in a bit-serial manner and the other is always available for parallel operation.2n clock cycles are required to complete the process of multiplication. n clock cycles are used for addition of partial products and another n clock cycles for the propagation of carry [4]. The delay due to storage elements is eliminated. Stenzel (11) proposed the partial-partial matrix generation-reduction schemes of Wallace and Dada may be enhanced through the use of multiplier modules larger than 1*1 bit and counters larger and more

general than (3, 2) counters. The larger multiplier modules permit the generation of a partial- partial matrix containing fewer total bits and having a maximum height less than generated by 1*1 multipliers and, with current implementations, require fewer IC packages to do so. Several large counters and multiplier modules have been realized.

II. REVIEW OF SERIAL MULTIPLIERS

Serial multipliers are classified into two categories namely serial-serial and serial-parallel multiplier. In a serial [12] multiplier both the operands are loaded in a bit serial fashion, reducing the data input pads to two. Serial multiplier designs which are particularly suitable for applications where input data are sequentially presented .The operating speeds are determined mainly by the propagation delays along the critical path within the processing elements. It is highly desirable to reduce the number of partial products before the carry-save adder's stage. The drawback is that and higher order compressors are slower and consumes more power than the full adders. An accumulator is an adder which successively adds the current input with the value stored in its internal register.

DOB kin (6) proposed the novel serial links provide better performance than parallel links for long range communications, beyond several millimeters. We analyze the technology dependence of link performance. An example for 65 nm technology is presented, and compare wave pipelined and register-pipelined parallel links to a high performance serial link in terms of bit-rate, power, area and latency. Multiplication of two n bit unsigned numbers requires 2n clock cycles to complete the process out of which n clocks are used for n-row carry-save additions, and the other n clocks are utilized only to propagate the remaining carries. This CSAS structure is modified so that it operates as a CSAS unit for the first n clocks and reconfigures itself as an n bit ripple-carry parallel adder at the (n + 1)st clock, thus allowing the carries to ripple through, eliminating the delay due to storage elements during the last n clocks. Figure 3.2a shows the product matrix where the matrix is added by counting the number of ones in the individual columns.

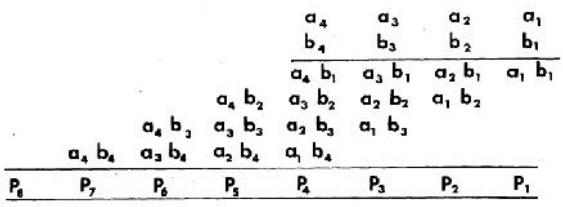

Fig 1. Product matrix

Both signed and unsigned multiplication involves AND gates, full adders and D flip flops. The critical path is along the D flip flop and the AND gates.

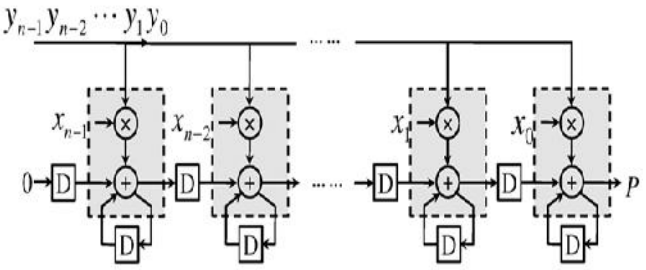

Fig 2. CSAS unsigned multiplier

A simple CSAS multiplier implementation for positive numbers is shown in Figure. 2. For n bit multiplication it takes 2n clock pulses to complete the process. Both multiplier and multiplicand bits are loaded serially AND gates are used to multiply the serially loaded data. Full adders are employed for addition

D flip flops at the side of the adder unit are used for shifting and those that are below are used for storing the carry bit. A 0 at the MSB side is used for reset. P is the result or product of the multiplication process.

A simple CSAS multiplier implementation for signed numbers is shown in Figure.3 during the first (n - 1) clock pulses, Q is zero and the structure acts like a CSAS unit. The operation is similar to unsigned multiplication except that it involves a bit Q and another EX-OR gate. Q stores the MSB of the operands which is the signed bit and is EX-ORed with the multiplicand bit. The result from EX-OR is ANDed with y_0. Then the product P is obtained as an unsigned multiplication unit.

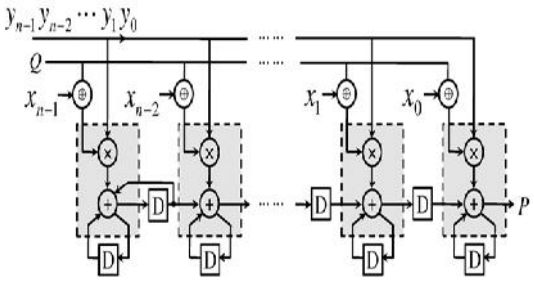

Fig 3. CSAS multipliers implementation for signed number

A Fast CSAS Multiplier capable of producing 2n-bit output in n clock cycles at the expense of an extra RCA. It's a low area×time^2 (AT^2) complexity 2's complement serial-parallel multiplier. It used the baugh-wooley algorithm to avoid the sign extension problem an designed

many systolic and non- systolic multiplier with low AT^2 complexity based on the booth's multi-bit recoding.

III. IMPLEMENTED SERIAL - SERIAL MULTIPLIER

This multiplier will generate all partial product rows and their accumulation will be done in only n cycles for an n×n multiplication. Accumulation is an integral part of serial multiplier design. A typical accumulator is simply an added that successively adds the current input with the value stored in its internal register. Here in this section we will propose a technique for the generation of individual rows of partial products by considering two serial input operands. In which one operand is starting from LSB and another starts from MSB. According to this algorithm the partial product row and column structure is revamped.

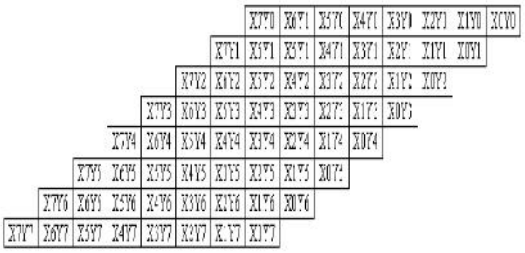

Fig 4. Conventional PP Formation

Fig 4 shows the conventional partial product formation of an 8*8 multiplier. Fig.5 shows the proposed partial product formation of an 8*8 multiplier

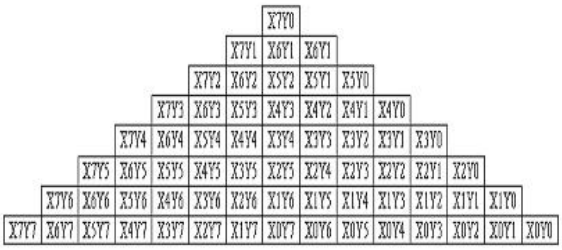

Fig 5. Proposed PP Formation

To form the PP matrix, the multiplier and the multiplicand are ANDed. To reduce the height of the PP matrix, column to row transformation is performed using 1's counters. The length of the counters varies according to the column height of the PP matrix which gets incremented from 1 to n and then reduces to 1. One row of PPs is generated in each cycle.

The product of two unsigned numbers X and Y can be written as

$$P = X \cdot Y = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} x_i y_j 2^{i+j}$$

Where x_i and y_j are the i^{th} and j^{th} bits of X and Y with bit 0 being the LSB

By decomposing and rearranging

$$P = \sum_{r=0}^{n-1} PP_r$$

Where

$$PPr = PP_r^L + PP_r^C + PP_r^R$$

$$PP_r^L \begin{cases} 0, & r = 0 \\ \sum_{K=0}^{r-1} x_{n-k-1}.y_r.2^{n+r-k-1}, & r = 1,2,\ldots,n-1 \end{cases}$$

$$PP_r^C = x_{n-r-1}.y_r.2^{n-1}, r = 0,1,\ldots n-1,$$

$$PP_r^R \begin{cases} 0, & r = 0 \\ \sum_{K=0}^{r-1} x_{n-k-1}.y_{r-k-1}.2^{n+r-k-2}, & r = 1,2,\ldots.n-1 \end{cases}$$

The complete architecture of the proposed counter-based serial-serial multiplier (8× 8) is shown in Figure.7

X and Y are the operands to be multiplied and are loaded serially and PP generation is same as in conventional method.1's counting is carried out using asynchronous counters. Shifting left or right is done using D-flip flops. Two clocks are employed to synchronize the data flow between the two stages. The counters and shift registers are reset to "0" in every cycle to allow a new set of operands to be loaded.clk2 is derived from clk1 to drive the latching register. Each counter changes state at the rising edge of the clock line only if a "1" is produced by its driving AND gate.

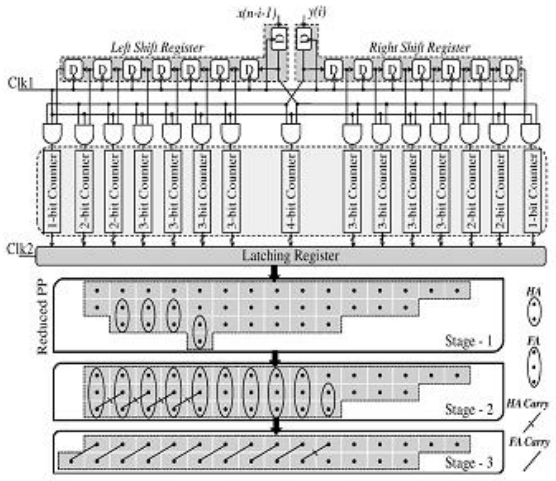

Fig 6. Implemented architecture for serial unsigned multiplication

After n cycles, the counters hold the sums of all the 1's in the respective columns and their outputs are latched to the second stage for summation. The latched outputs are wired to the correct FAs and HAs according to the positional weights of the output bits produced by the counters. The latching register between the counter and the adder stages not only makes it possible to pipeline the serial data accumulation and the CSA tree reduction, but also prevents the spurious transitions from propagating into the adder tree.

There are three stages in addition 2 stages of carry save (CSA) adder using half adder and Full adder, Ripple carry adder (RCA).The column height has been reduced from 8 to 4 and the final product, can be obtained with two stages of CSA tree and a final RCA.

IV. 8 AND 16 BIT WORD LENTH FOR 2's COMPLEMENT NUMBERS

The Most digital systems operate on signed numbers commonly represented in 2's complement. The 2'scomplement numbers are using the Baugh–Woolley algorithm. The architecture of the proposed 2'scomplement serial-serial multiplier is depicted below structure. The addition of the term raises the height of CSA tree by only two bits regardless of the word length of the operands.

In using the 2's complement technique, it is very important to understand the impact of the fixed number of bits, which needs to be decided at the beginning. For example, in 8 bits, one can represent a total of 256 different binary arrangements. If there were no provision (convention) for negative numbers, these 256 arrangements would logically represent the decimal numbers 0 to 255 (%00000000 to %11111111). However, if the 2's Complement convention is specified, the 256 arrangements have the following meanings.

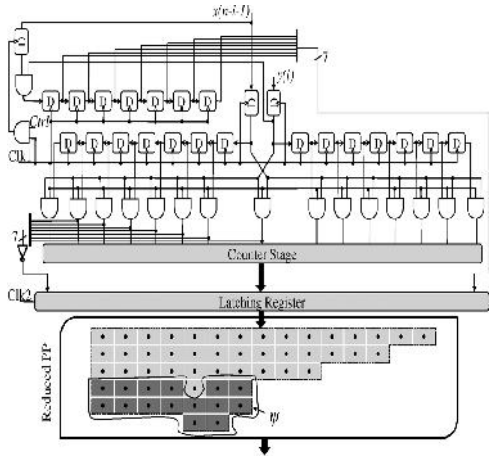

Fig 7. Implemented architecture for 8×8 serial-serial signed multiplication

The operation is similar to unsigned multiplication. PPT height reduction is performed using Baugh- Woolley algorithm. In this algorithm, the PP matrix is formed like in unsigned multiplication. The MSB of each row in the PP matrix is complemented except the last row.

All other bits of the last row are complemented. A 1 is added to the column that is to the left of the middle column. The counter outputs are latched to the adder stage. Finally, addition of the PP bits is carried out by the adder units. It's simpler to design.

Baugh-Woolley Two's compliment Signed multipliers is the best known algorithm for signed multiplication because it maximizes the regularity of the multiplier and allow all the partial products to have positive sign bits [3].Baugh–Woolley technique was enveloped to design direct multipliers for Two's compliment numbers [9].When multiplying two's compliment numbers directly, each of the partial products to be added is a signed numbers. Thus each partial product has to be sign extended to the width of the final product in order to form a correct sum by the Carry Save Adder (CSA) tree. According to Baugh-Woolley approach, an efficient method of adding extra entries to the bit matrix suggested to avoid having deal with the negatively weighted bits in the partial product matrix.

V IMPLEMENTATION AND RESULTS

The 8 bit CSAS multiplier can be designed using MODELSIM 10.0a with VHDL code.

Fig 8. Simulation waveform of CSAS unsigned multiplication

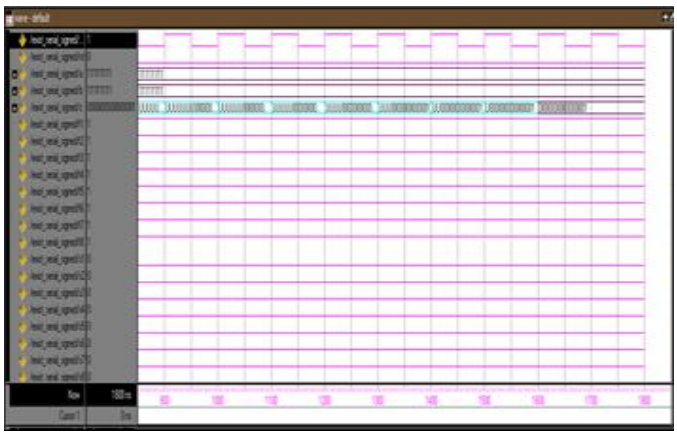

Fig 9. Simulation waveform for CSAS signed multiplication

The design can be implemented and verified using FPGA trainer kit. The simulation result of unsigned multiplication is shown in Figure 8 and signed multiplication is shown in Figure 9.

a. Simulation results of counter based multiplier

The 8 bit proposed multiplier can be designed using MODELSIM 10.0a with VHDL code. The design can be implemented and verified using FPGA trainer kit. The simulation result of unsigned multiplication is shown in Figure 10 and signed multiplication is shown in Figure 11

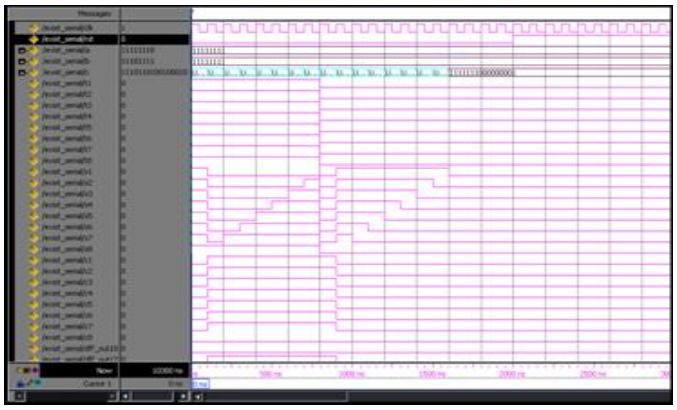

Fig 10. The simulation result of counter-based unsigned multiplication

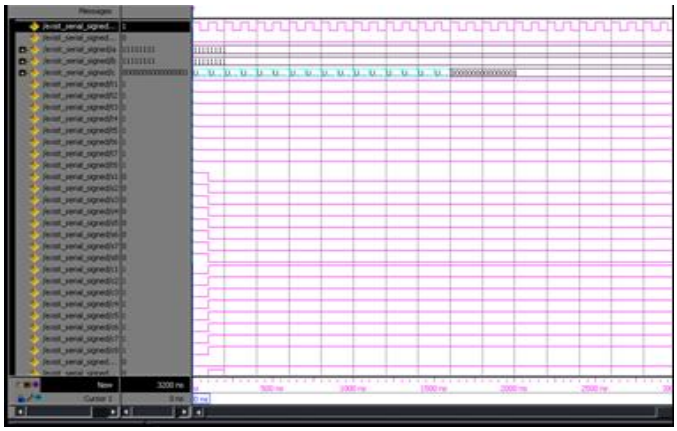

Fig 11. The simulation result of counter-based signed multiplication

V. CONCLUSION

In the CSAS multiplier, the inputs has to pass through AND gate, Full adder and DFF. The critical path delay exists along these units. This technique requires 2n clock cycles for computation. In the counter-based multiplier, the critical path delay exists in only one AND gate of the counter. Hence it requires n clock cycles for computation. 8*8 multiplication has been performed for both the techniques. The simulation results have been compared using Modelsim 10.0b

REFERENCES

[1]. Booth A.D. (1951) “A signed binary multiplication technique,” Quarterly .J Mechan. Appl. Math., Vol. 4 No. 2 pp. 236–240

[2]. Harpreet Singh Dhillon and Abhijit Mitra “A Reduced-Bit Multiplication Algorithm for Digital Arithmetic” International Journal of Computational and Mathematical Sciences 2:2 2008

[3]. R. Gnanasekaran, “A fast serial-parallel binary multiplier,” IEEE Trans. Comput., vol. C-34, no. 8, pp. 741– 744,1985

[4]. P. Ienne and M. A Viredaz, “Bit-serial multipliers and Squarers,” IEEE Trans. Comput., vol. 43, no. 12, pp. 1445–1450, 1994.

[5]. Mark Vesterbacka, Kent Palmkvist, and Lars Wanhammar,“Serial squarers and serial/serial multipliers

[6]. DOB kin R, Morgenstern A, Kolodny A. and Ginosar.R (2008), “Parallel vs. serial on-chip c communication,” in Proc. ACMInt.Workshop Syst. Level Interconnect Prediction (SLIP), Newcastle, U.K., pp. 43–50.

[7]. Manas Ranjan Meher, Ching Chuen Jong, and Chip-Hong Chang (2011),”A High Bit Rate Serial Multiplier With On-the-Fly Accumulation by Asynchronous Counters,” IEEE Trans. Very Large Scale Integer. (VLSI) Syst., Vol.19 No.10.

[8]. Morales-Sandoval M, Feregrino-Uribe C, Kitsos P, “Bit-Serial and Digit-Serial GF(2m) Montgomery Multipliers using Linear Feedback Shift Registers.

[9]. Dudha C., Schopenhauer T. and Wallner P, (2008) “Parallelization of serial digital input signals,” U.S. Patent 2008/0 055 126 A1.

[10]. DOB kin R, Moyal M, Kolodn A. and Ginosar R. (2010), “Asynchronous current mode serial communication,” IEEE Trans. Very Large Scale Integer. (VLSI) Syst., Vol. 18 No. 7 pp. 1107–1117.

[11]. William J. Stenzel, William J. Kumitz, Member IEEE, Gilles H.Garcia(1977), “Compact High- Speed Parallel Multiplication Scheme” IEEE Trans. Compute. C-26:948-957

[12]. A. K. Lenstra and E. Verheul, “Selecting cryptographic key sizes,”J.Cryptology, vol. 14, no. 4, pp. 255–293, 2001

[13]. N. R. Strader and V. T. Rhyne, “A canonical bit sequential multiplier,” IEEE Trans. Comput., vol. C-31, no. 8,pp. 791–795, Aug. 1982

[14]. J. Sklansky, "Conditional-sum addition logic," IRE Transactions on Electronic Computers, vol. 9, pp. 226–231,1960.

[15]. R.P.Brentand H.T.Kung,"ARegular Layout for Parallel Adders,"IEEE Transactions on Computers,vol. C-31,no.3,pp.260–264, 1982.

[16]. Y.Kim and L.S .Kim, "64-bit carry-select adder with reduced area," Electron. Lett., vol.37, no.10,pp.614-615,May 2001.

[17]. G. Bi and E. V. Jones, "High-performance bit-serial adders and multipliers,"IEEProc G-Circuits Devices Systs, vol.139, no. 1, pp.109–113, Feb. 1992

[18]. R. Menon and D. Radhakrishnan, "High performance 5:2 compressor architectures,"IEEProc-Circuits Devices Syst., vol. 153, no. 5, pp.447–452, Oct. 2006

[19]. ManasRanjanMeher, Ching-ChuenJong,Chip-Hong Chang "High-Speed and Low-Power Serial Accumulator for Serial/Parallel Multiplier" IEEE Xplore,2010.

[20]. Nibouche.O, A. Bouridarie, M. Nibouche.O "New Architectures for Serial-Serial Multiplication" 0-7803-6685-9101/$10.0002000 IEEE.

[21]. Ravi Nirlakalla, Thota Subba Rao, Talari Jayachandra Prasad, "Performance Evaluation of High Speed Compressors for High Speed Multipliers" Vol. 8, No. 3, November 2006, 293-306

[22]. H. I. Saleh, A. H. Khalil, M. A. Ashour, and A. E. Salama, "Novel serial-parallel multipliers," IEE Proc-Circuits Devices Syst., vol. 148, no. 4, pp. 183–189, Aug. 2001

[23]. P. Ndai et al., "Fine-grained redundancy in adders," ISQED, 2007.