High Bit Rate Column Compressed Serial-Serial Multiplier

Santhosh Kiran Peruri [1], P.P.Priyadarshini [2]

[1]*Department of ECE, Narasaraopet Engineering College, Narasaraopet, Ap, India.*

[2]*Assistant.proffesor Department of ECE, Narasaraopet Engineering College, Narasaraopet, Ap, India.*

***Abstract*— A design of serial-serial hybrid multiplier is proposed for applications with high data rate. Here the proposed technique effectively forms the entire partial product rows in just n cycles where as conventional serial-serial multipliers take 2n cycles to form all partial products. The conventional way of partial product formation is rearranged here. Here the proposed architecture achieves high data rate by replacing full adders with asynchronous 1's counter so that critical path is limited to only DFF and an AND gate. The use of asynchronous counter reduces the height of the partial product rows from n to $[\log_2 n]+1$, resulting in reduction of complex adder tree. The proposed multiplier consists of serial-serial data accumulation unit followed by a dadda multiplier which reduces the average power dissipation. It has a small delay penalty to complete a multiplication when compared to a conventional parallel array multiplier.**

Keywords - Binary multiplication, parallel multipliers, serial multipliers, partial products, asynchronous counter.

I. INTRODUCTION

MULTIPLIERS are the fundamental and essential building blocks of VLSI systems. Area, speed and power consumption are major concerns in design and implementation of multipliers. LOW POWER AND HIGH SPEED multiplier circuits are highly demanded due to issues concerning reliability and portability. But it is not always possible to achieve both criteria simultaneously. Therefore, a good multiplier design requires some tradeoff between speed and power consumption. So,to optimize the area, delay and power consumption of arithmetic computations in battery powered VLSI circuits we explore alternative architectural concepts for the design of digital multipliers.Hardware implementation of a multiplication operation consists of three stages, specifically the generation of partial products (PPs), the reduction of PPs and the final carry-propagation addition [1].The partial products can be generated either in parallel or serially, depending on the target application and the availability of input data. The partial products are generally reduced by carry-save adders (CSAs).Carry propagation addition is done by a simple ripple carry adder (RCA) for low power or a carry look-ahead adder (CLA) for high speed [1]. As the height of PP tree increases linearly with the word length of the multiplier, its area, delay and power dissipation increase.Therefore reduction of PPs before CSA is needed. This can be achieved by Modified Booth algorithm to reduce the height of the PP matrix [2]. Another approach is to use high order column compressors instead of full adders (FAs) to increase the PP reduction ratio of the CSA stage [3], [4]. The drawback is that Modified Booth encoder adds both area and delay overheads to the simple partial product generation process, and higher order compressors are slower and consume more power than the full adders. Hence a hybrid combination of both techniques is often considered.

II. REVIEW OF SERIAL MULTIPLIERS

In a serial-serial multiplier both the operands are loaded in a bit-serial fashion. On the other hand, a serial-parallel multiplier loads one operand in a bit-serial fashion and the other is always available for parallel operation [7]. Lyon [1] proposed a bit-serial input output multiplier in 1976 which features high throughput at the expense of truncated output. A full precision bit-serial multiplier was introduced by Strader et al. for unsigned numbers [3]. The rudimentary cell consists of a 5:3 counter and some DFFs. Later, Gnanasekaran [5] extended the work in [2] and developed the first bit-serial multiplier that directly handles the negative weight of the most significant bit (MSB) in 2's complement representation. This method needs only cells for an n-bit multiplication but it introduces an XOR gate in the critical path, which ends up with a more complicated overall design. Lenne et al. [3] designed a bit-serial-serial multiplier that is modular in structure and can operate on both signed and unsigned numbers.A typical serial-serial multiplier consists of a

cell, called a bit-cell (BC).Such cells are interconnected to produce the output in a bit-serial manner for an serial-serial multiplier. The operands bits and are loaded serially in each cycle and added with the far carry, local carry, and the partial sum in the5:3 counter.To reduce the number of computational cycles from 2n to n in an nxn serial multiplier, several serial-parallel multipliers have been developed over the years. Most of them are based on a carry save add shift (CSAS) structure. It can be observed that the critical path consists of an FA, a DFF, and an AND gate for the previous unsigned multiplier.

III. PROPOSED SERIAL ACCUMULATOR

Accumulation is an integral part of serial multiplier design. A typical accumulator is simply an adder that successively adds the current input with the value stored in its internal register. Generally, the adder can be a simple RCA but the speed of accumulation is limited by the carry propagation chain. The accumulation can be speed up by using a CSA with two registers to store the intermediate sum and carry vectors, but a more complex fast vector merged adder is needed to add the final outputs of these registers. In either case, the basic functional unit is an FA cell. A new approach to serial accumulation of data by using ASYNCHRONOUS COUNTERS is suggested here which essentially count the number of 1's in respective input sequences (columns). An accumulation of n integers x(i) for i= 0,1,……,n --1can be mathematically expressed as

$$S = \sum_{i=0}^{n-1} x(i) \qquad (1)$$

For ease of exposition, let x(i) be an unsigned integer represented by m binary weighted bits. Hence, (1) can be rewritten as

$$S = \sum_{i=0}^{n-1} \left[\sum_{j=0}^{m-1} x_j(i) \cdot 2^j \right] \qquad (2)$$

Where is $x_j(i)$ is the j th bit of the i th operand and is associated with a positional weight 2^j . By changing the order of accumulation of n and m, (2) becomes

$$S = \sum_{j=0}^{m-1} \left[\sum_{i=0}^{n-1} x_j(i) \cdot 2^j \right] \qquad (3)$$

The inner-summation of (3), represents the sum of all the 1's presented in the j th column and it can easily be accomplished by a simple serial 1's counter of width $h=[\log_2 n]+1$. Thus, an m-bit accumulator can be realized with m such dedicated counters to concurrently accumulate the 1's in each bit position. The dependency graph (DG) of such a scheme with m=8and n=7 is shown in Fig. 1. The nodes in the DG perform $y'=y+x_j(i)$, which is equivalent to an increment of y if $x_j(i)=1$.The architecture of the accumulator corresponding to Fig.1 is shown in Fig. 2. The bits of the input operands are serially fed into their corresponding counters from column 0 (right-most in Fig. 1) to column 7. These counters execute independently and concurrently. In each cycle of accumulation, a new operand is loaded and the counters corresponding to the columns that have a 1 input are incremented. The counters can be clocked at high frequency and all the operands will be accumulated at the end of the nth clock. The final outputs of the counters need to be further reduced to only two rows of partial products by a CSA tree, such as a Wallace's or Dadda's tree [1]. A carry propagate adder is then used to obtain the final sum. In Fig. 2, the counters C are used to count the number of 1's in a column. Each of them is a simple DFF-based ripple counter. The clock is provided to the first DFF and all the other DFFs are triggered by the preceding DFF outputs. The clock input is synchronized with the input data rate and thus the operands can be accumulated with a high frequency defined by the setup time and propagation delay of a DFF. Moreover, the counters change states only when the input is "1," which leads to low switching power. This simple and efficient bit accumulation technique is used to design the proposed serial-serial multiplier.

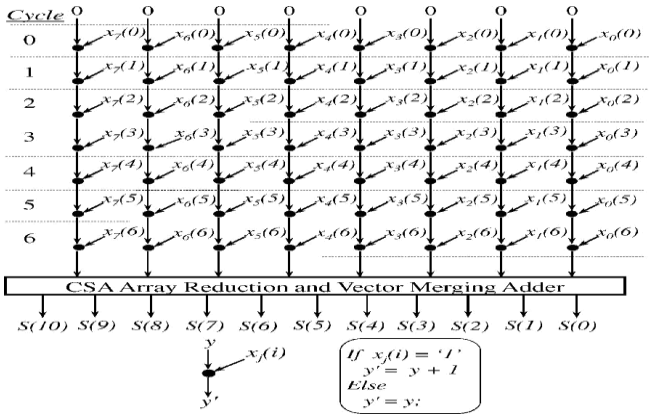

Fig. 1. Dependency graph of the proposed accumulator for 7 8-bit operands

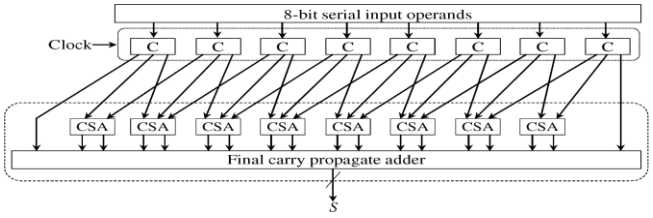

Fig. 2. Architecture of an accumulator with m=8 and n=7

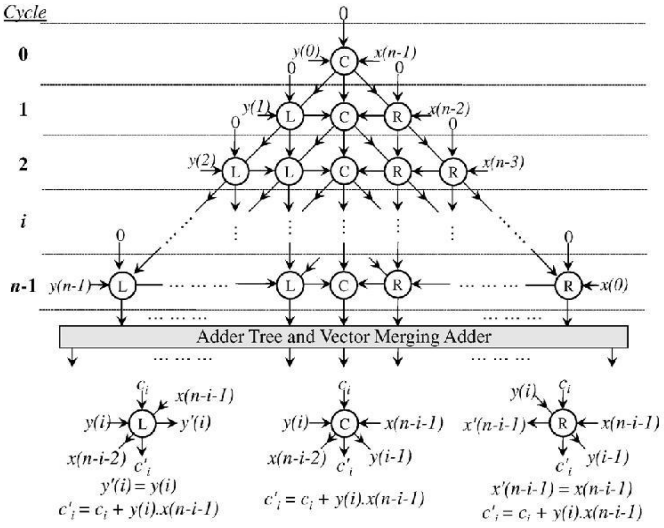

Fig.3.Dependency graph of nxn serial-serial partial product generation and accumulation.

IV.PROPOSED SERIAL-SERIAL MULTIPLIER

This section proposes a new technique of generating the individual row of partial products by considering two serial inputs, one starting from the LSB and the other from MSB. Using this feeding sequence and the proposed counter-based accumulation technique presented in Section III, it takes only n cycles to complete the entire partial product generation and accumulation process for an nxn multiplication. The theoretical underpinning of this design is elaborated as follows. The product P of two n-bit unsigned binary numbers X and Y can be expressed as

$$P = X \cdot Y = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} x_i y_j 2^{i+j} \qquad (4)$$

Where x_i and y_j are the i th and j th bits of X and Y , respectively, with bit 0 being the LSB. By reversing the sequence of index i of (4)

$$P = \sum_{i=n-1}^{0}\sum_{j=0}^{n-1} x_i y_j 2^{i+j} \qquad (5)$$

By appropriately sequencing the input bits of X and Y into a shift register, one PP(PPr) in each cycle(Cycler) can be generated.

$$PP_r = PP_r^L + PP_r^C + PP_r^R \qquad (6)$$

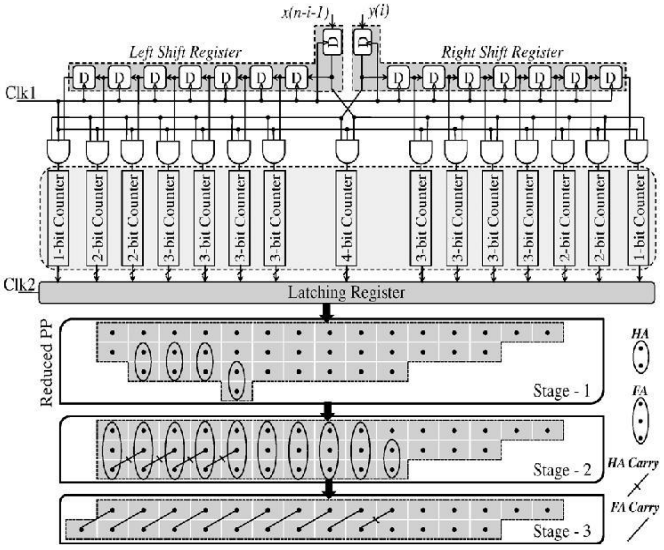

Fig.4. Proposed architecture for 8x8 serial-serial unsigned multiplication.

Consequently, P can be obtained in n cycles. Fig.5 illustrates the PP generation of an 8 × 8 multiplier for two unsigned numbers X and Y. Fig. 5(a) shows the conventional partial product formation and Fig. 5(b) shows the generation sequence of the PPs according to (6). The complete architecture of the proposed counter-based serial-serial multiplier (8 × 8) is shown in Fig. 4.

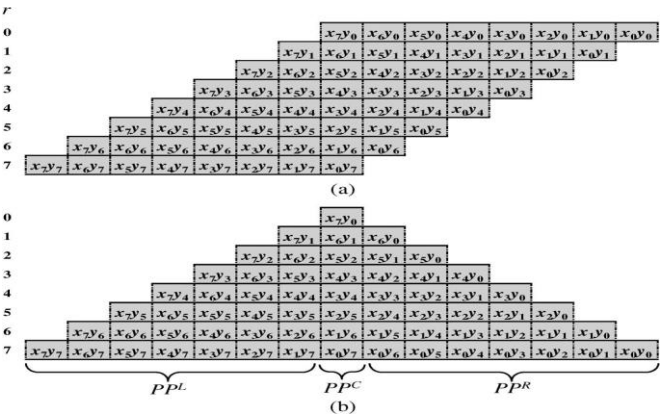

Fig.5 Partial product generation schemes for an 8x8 serial-serial multiplication: (a) conventional and (b) proposed.

Simulation Results:

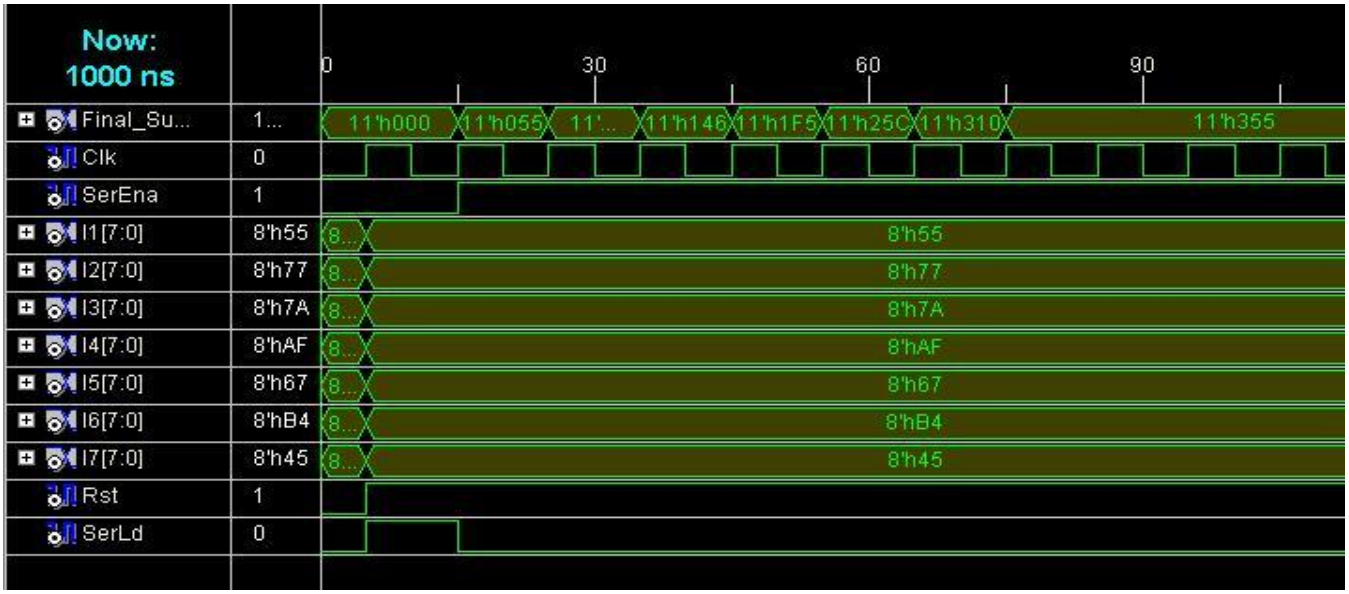

Fig.6 Simulation waveform of Serial Accumulator.

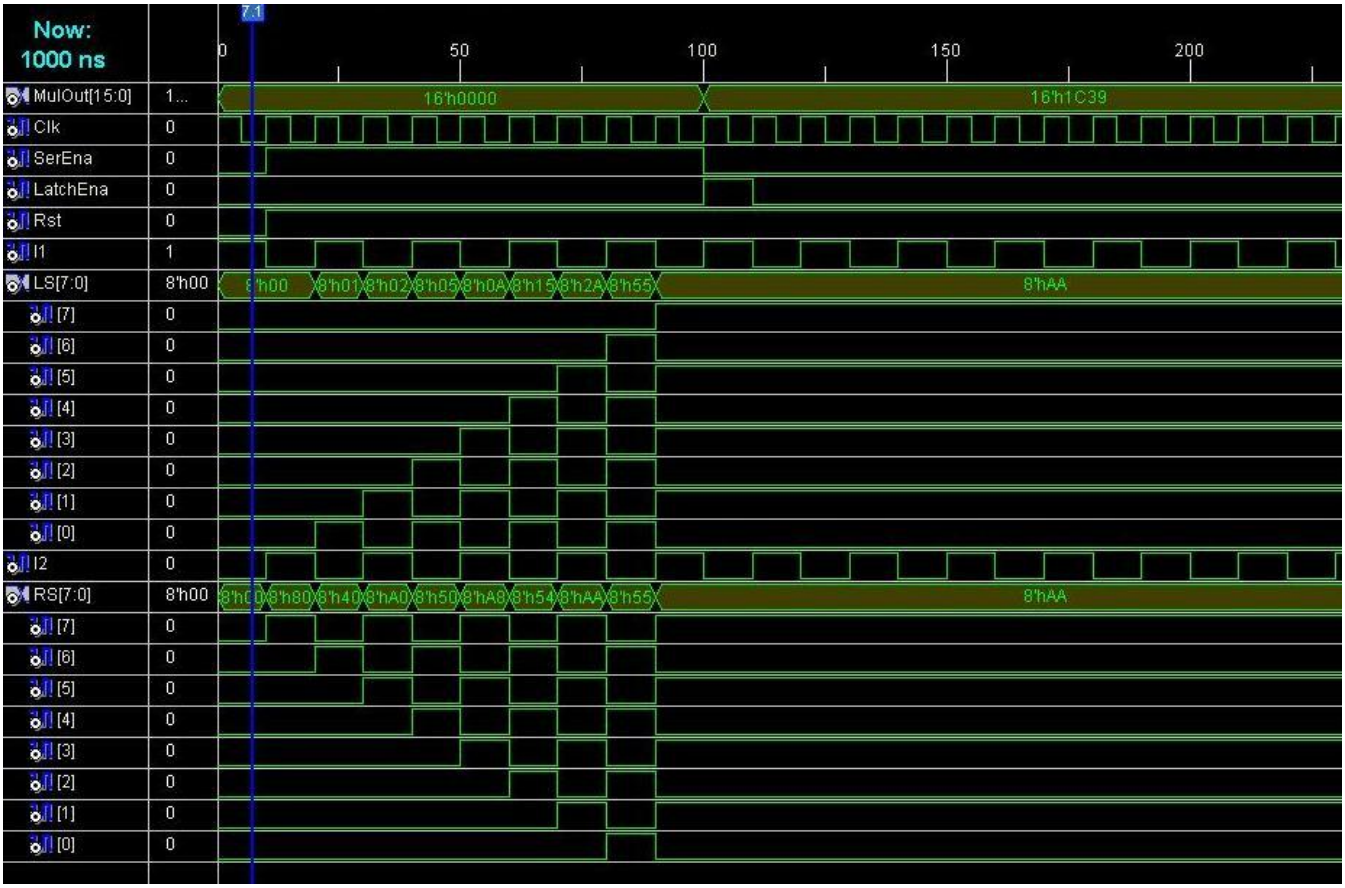

Fig.7 Simulation waveform of Unsigned 8x8 Serial- Serial Multiplier.

VI. CONCLUSION

In this paper, a new method for computing serial-serial multiplication is introduced by using low complexity asynchronous counters. By exploiting the relationship among the bits of a partial product matrix, it is possible to generate all the rows serially in just cycles for an multiplication. Employing counters to count the number of 1's in each column allows the partial product bits to be generated on-the-fly and partially accumulated in place with a critical path delay of only an AND gate and a DFF. The counter-based accumulation reduces the PP height logarithmically and makes it possible to achieve an effective reduction rate of using an FA-based CSA tree.Therefore; low-cost serial multipliers are widely adopted in hardware cryptography and SoC application.

REFERENCES

[1] Manas Ranjan Meher, Ching Chuen Jong, and Chip-Hong Chang, A High Bit Rate Serial-Serial Multiplier With On-the-Fly Accumulation by Asynchronous, Digital Object Identifier 10.1109/TVLSI.2010.2060374.

[2] M. R. Meher, C. C. Jong, and C. H. Chang, "High-speed and lowpowerserial accumulator for serial/parallel multiplier," in Proc. IEEE Asia-Pacific Conf. Circuits Syst. (APCCAS), Macau, China, 2008, pp. 176–179.

[3] R.Menon "High performance 5:2 compressor architecture " IEE proc-Circuits Devices Syst., Vol. 153, no.5, pp. 447-452, oct. 2006.

[4] Ron S. Waters, Earl E.Swartzlander "A Reduced Complexity Wallace multiplier reduction" in IEEE transactions on Computers, Aug.

[5] Whytney J. Townsend, Earl E. Swartz, "A Comparison of Dadda and Wallace multiplier delays". Computer Engineering Research Center, The University of Texas.

[6] A. D. Booth, "A signed binary multiplication technique," Quarterly J. Mechan. Appl. Math., vol. 4, no. 2, pp. 236–240, Aug. 1951.

[7] M. Ghoneima, Y. Ismail, M. Khellah, J. Tschanz, and V. De, "Seriallink bus: A low-power on-chip bus architecture," IEEE Trans. Circuit.Syst. I, Reg. Papers, vol. 56, no. 9, pp. 2020–2032, Sep. 2009.

[8] R. Dobkin, M. Moyal, A. Kolodny, and R. Ginosar, "Asynchronous current mode serial communication," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 7, pp. 1107–1117, Jul. 2010.

[9] M.Burzio and P,.pellegriono,"seriailizer-paralleliing circuit for high speed digital signals",U.S Patent,Aug.4,1998.

[10] "Xilinx13.4, Synthesis and Simulation Design Guide", UG626 (v13.4) January 19, 2012.