International Journal of Computer Applications Technology and Research
Volume 3– Issue 4, 263 - 266, 2014, ISSN: 2319–8656

Design of Serial-Serial Multiplier based on the Asynchronous Counter Accumulation

Srikanth.P
Vardhaman college of Engineering
Hyderabad, India

Latha Kodam
Vardhaman college of Engineering
Hyderabad, India

Abstract: A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest clement in the system. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. In this project we design a serial –serial multiplier based on the asynchronous counter accumulation. In this serial – serial multiplier the number of sampling cycles is reduced to n from 2n by the reduction of partial products which is based on the asynchronous counters. It achieves high bit sampling rate by replacing conventional fulladder and highest 5:3 counters by simple asynchronous 1's counters so that the critical path is limited to only AND gate and D Flip-flops.

Keywords: Partial Products (PPs), Serial-Serial Multiplier, Ripple carry adder (RCA), CSA, Accumulation.

1. INTRODUCTION

Multipliers are the basic and essential building blocks of VLSI systems. The basic factors which are taken into consideration for a design of a multiplier are speed, area and power consumption. In general a low power and a high speed multiplier circuits are preferred which are not possible to achieve to achieve both the criteria simultaneously. So a good multiplier design requires some tradeoff between speed and power consumption. Hardware implementation of multipliers consists of three stages they are the generation of partial products (PPs), the reduction of partial products (PPs), and the final carry propagation addition. Based on the input data the partial products (PPs) are being generated either serially or parallel.

2. REVIEW OF SERIAL MULTIPLIERS

In a serial-serial multiplier both the operands are loaded in a bit-serial fashion, reducing the data input pads to two whereas a serial-parallel multiplier loads one operand in a bit-serial fashion and the other is always available for parallel operation.

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 [2]. 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.

The 1-bit slice of a typical serial-serial multiplier, called a bit-cell (BC), is excerpted from is shown. 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 shows the unsigned and 2's complement serial-parallel multiplier based on the CSAS structure. It can be observed that the critical path consists of an FA, a DFF, and an AND gate for the unsigned multiplier

3. 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 is presented in the following section; it takes only n cycles to complete the entire partial product generation and accumulation process for an nxn multiplication. The Partial products generation of the Conventional Serial –Serial multiplier is shown in the Fig 1.
The theoretical explanation of the proposed serial-serial multiplier which is based on the asynchronous counters accumulation explained as follows.

The product P of two -bit unsigned binary numbers X and Y can be expressed as

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

Where, x_i and y_j are the ith and jth bits of X and Y respectively, with bit 0 being the LSB. By reversing the sequence of index i and rearranging the above equation can be written as

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

On rearranging the above equation,

$$P = \sum_{}^{n-1} PPr$$

r=0 $PPr=PPr^{L}+PPr^{C}+PPr^{R}$

Where,

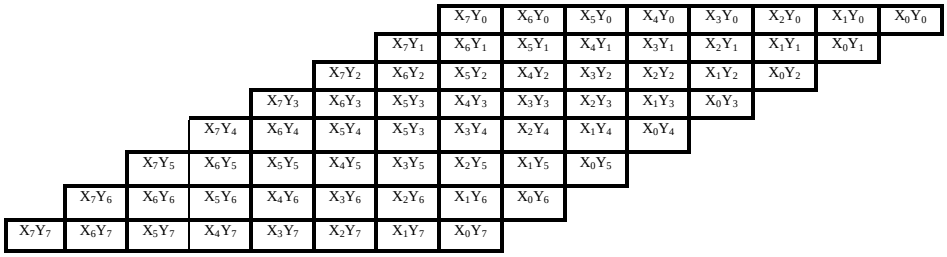

Fig 1 Partial Product generation scheme for an 8x8 Conventional serial-serial multiplier

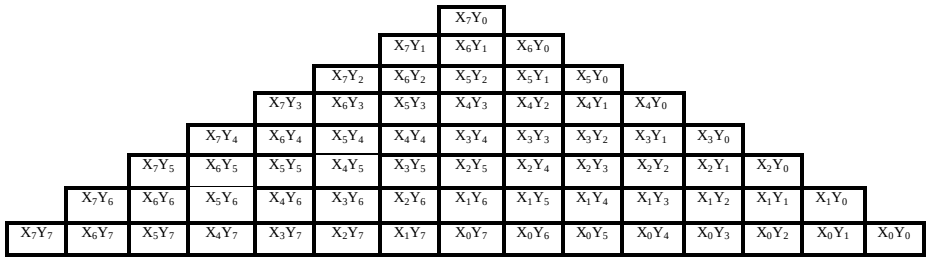

Fig 2 Partial Product generation scheme for an 8x8 Proposed serial-serial multiplier

The partial product row PPr can be generated in r cycles if X is fed from MSB (bit n-1) first and Y is fed from LSB first (bit 0), then in the r^{th} cycle ,PPr^{C} is a partial product bit generated by the current input bits x_{n-r-1} and y_r, PPr^{L} are partial product bits of the current input bit, y_r and each of the preceding input bits of X, i.e x_{n-k-1},for k=0,1,….r-1.and PPr^{R} are the partial product bits of the input bit x_{n-r-1},and each of the preceding input bits of Y, i.e Yr-k-1 for k=0,1,2,….r-1.By appropriately sequencing the input bits of X and Y into a shift register, one PP(PPr)in each cycle can be generated. As a result P can be obtained in n cycles

3.1 Counter -based Accumulation Technique

Accumulation of the partial products is done by using the asynchronous 1's counters. A typical Accumulator is an adder which 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).

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

of all the 1's in the respective columns and their outputs are latched to the second stage for summation. The latching

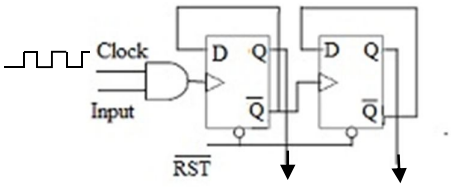

Fig 3.Hardware Architecture of 2bit 1's counter

incremented. The counters can be clocked at high frequency and all the operands will be accumulated at the end of the 'n'th 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 . A carry propagate adder is then used to obtain the final sum.

3.2 Proposed Architecture

The complete architecture for the proposed serial - serial multiplier (8x8) which is based on the asynchronous counter accumulation is shown in the figure 1.3. A PP bit corresponding to the middle column of the PP is produced by the center AND gate when a new pair of input bits (x_{n-i-1} and y_i) is latched by the two DFFs (top-middle) in each clock cycle. In the next cycle, x_{n-i-1} and y_i are shifted to the left and right, respectively, to produce the partial product bits with another pair of input bits and by the array of AND gates. The outputs of the AND gates are given as the inputs to the respective asynchronous counters .To ensure that the outputs of the AND gates which are used to drive the counters are being gated by a Clock 1(Clk1).If the output of the driving AND gate is '1'then the state of the counter takes place at the rising edge of the clock. After the completion of 'n' cycles for an nxn serial –serial multiplier, all the counters hold the sums

register which is present in between the counter stage and the adder stage not only makes it possible to pipeline the serial

data accumulation and the CSA tree reduction, but also it avoids the unnecessary transitions from propagating into the adder tree. To synchronize the data flow between the counter and adder stages two clocks namely Clock1 (Clk1) and Clock 2(Clk2) are employed.Clock2 is derived from Clock1 in order to drive the latching register.

According to the Positional weights of the output bits produced by the counters, the latched outputs are being wired to the correct Fulladders (FAs) and Halfadders (HAs) so that the column height has been reduced from 8 to 4 and the final product P is obtained with in two stages of CSA tree and final RCA as shown in the Fig 4.

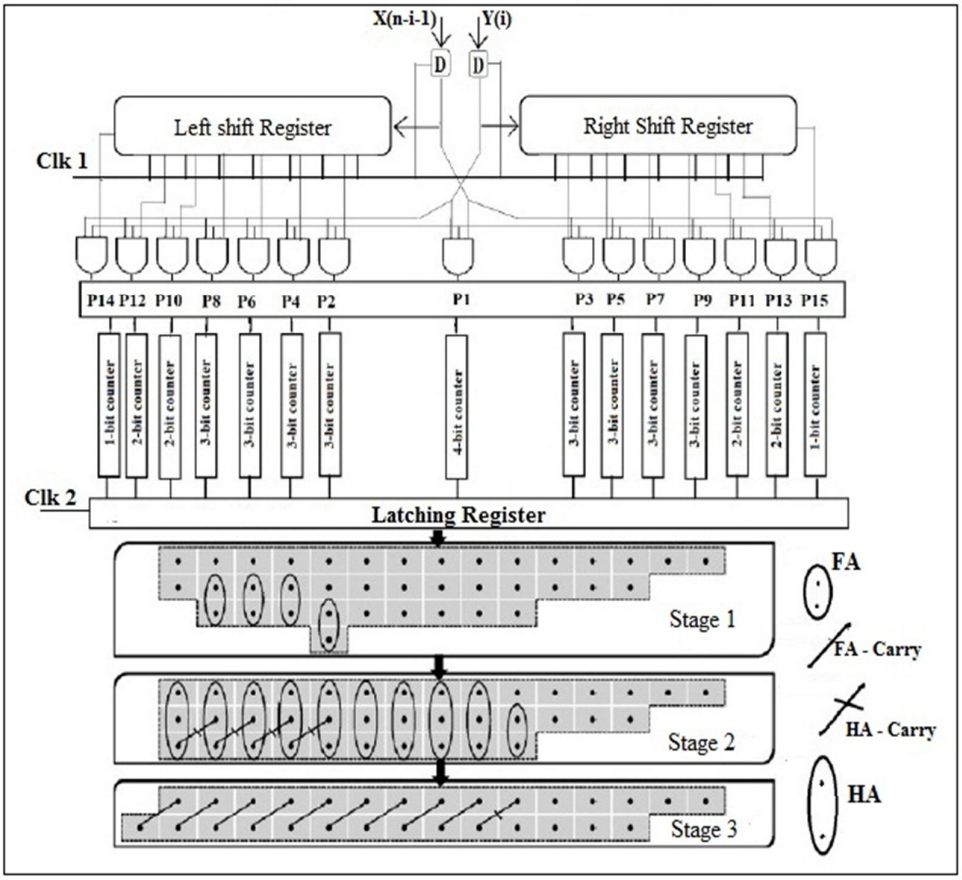

Fig 4 Proposed Architecture for 8x8 serial – serial multiplier

4. RESULTS OBTAINED

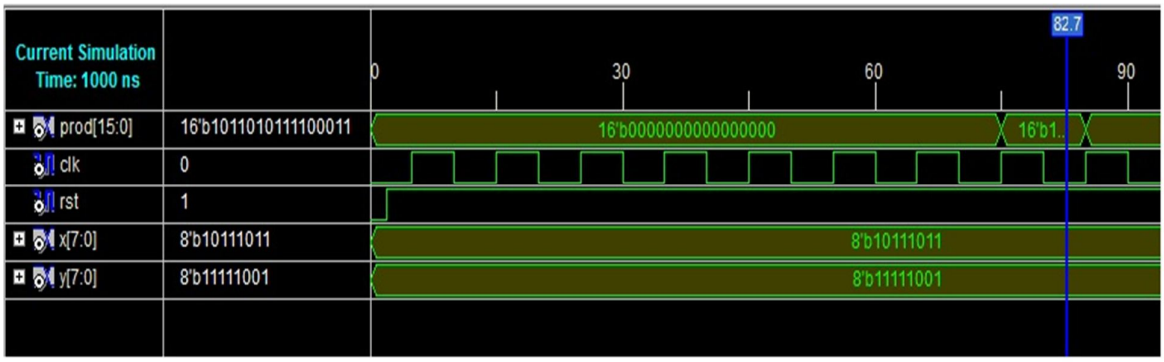

Fig 6 Proposed Unsigned Serial-Serial Multiplier output in Decimal format

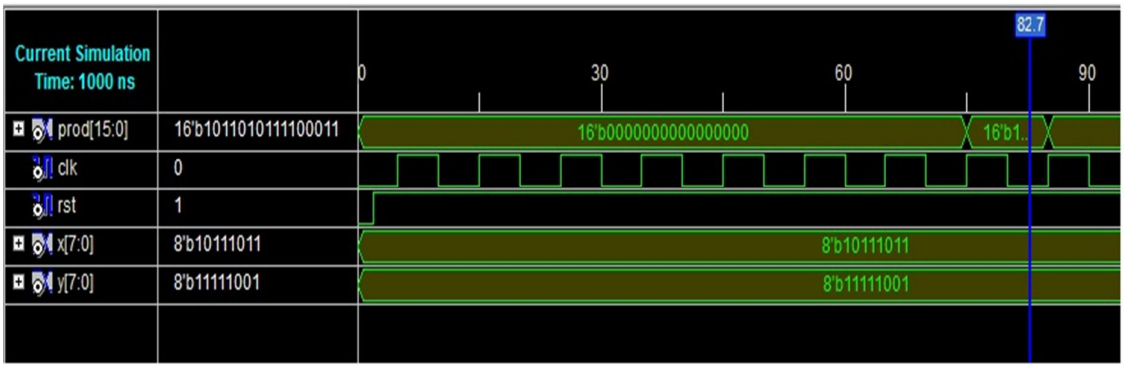

Fig 6 Proposed Unsigned Serial-Serial Multiplier output in Binary format

5. CONCLUSION

In this paper a new method of 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 n cycles for an n x n multiplication. Employing counters to count no of ones in each column allows the partial products 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 partial product height logarithmically and makes it possible to achieve an effective reduction rate. The proposed method outperforms many serial-serial and serial-parallel multipliers in speed. This approach has clear advantage of low I/O requirement and hence is most suitable for complex SOCs, advanced FPGAs and high speed bit serial application

6 REFERENCES

[1] R. F. Lyon, "Two's complement pipeline multipliers," IEEE Commun.Lett., vol. 24, no. 4, pp. 418–425, Apr. 1976.

[2] 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.

[3] P. Ienne and M. A Viredaz, "Bit-serial multipliers and squarers," IEEE Trans. Comput., vol. 43, no. 12, pp. 1445–1450, Dec. 1994.

[4] B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs. New York: Oxford Univ. Press, 2009.

[5] R. Gnanasekaran, "On a bit-serial input and bit-serial output multiplier," IEEE Trans. Comput., vol. C-32, no. 9, pp. 878–880, 1983.

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

[7] D. Bailey, E. Soenen, P. Gupta, P. Villarrubia, and D. Sang, "Challenges at 45 nm and beyond," in Proc. IEEE/ACM Int. Conf. Comput.- Aided Des. (ICCAD), San Jose, CA, 2008, pp. 11–18.

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

[9] M. Koga, "Serial/parallel converter," U.S. Patent 6 339 387 B1, Jan. 15, 2002.

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

[11] R. Dobkin, A. Morgenshtein, A. Kolodny, and R. Ginosar, "Parallel vs. serial on-chip communication," in Proc. ACMInt.Workshop Syst. Level Interconnect Prediction (SLIP), Newcastle, U.K., 2008, pp. 43–50.

[12] O. Nibouche, A. Bouridarie, and M. Nibouche, "New architectures for serial-serial multiplication," in Proc. IEEE Conf. Circuits Syst. (ISCAS), Sydney, Australia, 2001, vol. 2, pp. 705–708.