

# Low Power Multiplier by Effective Capacitance Reduction

<sup>1</sup>Nageshwar Reddy Peddamgari and <sup>2</sup>Damu Radhakrishnan

<sup>1</sup>Design Verification, Xilinx, Hyderabad, India

<sup>2</sup>Division of Engineering Programs, State University of New York, New Paltz, USA

## Article history

Received: 29-11-2016

Revised: 23-12-2016

Accepted: 31-12-2016

Corresponding Author:

Damu Radhakrishnan

Division of Engineering

Programs, State University of

New York, New Paltz, USA

Email: damu@enr.newpaltz.edu

**Abstract:** In this study we present an energy efficient multiplier design based on effective capacitance minimization. Only the partial product reduction stage in the multiplier is considered in this research. The effective capacitance at a node is defined as the product of capacitance and switching activity at that node. Hence to minimize the effective capacitance, we decided to ensure that the switching activity of nodes with higher capacitance is kept to a minimum. This is achieved by wiring the higher switching activity signals to nodes with lower capacitance and vice versa, for the 4:2 compressor and adder cells. This reduced the overall switching capacitance, thereby reducing the total power consumption of the multiplier. Power analysis was done by synthesizing our design on Spartan-3E FPGA. The dynamic power for our 16×16 multiplier was measured as 360.74 mW and the total power 443.31 mW. This is 17.4% less compared to the most recent design. Also, we noticed that our design has the lowest power-delay product compared to the multipliers presented in literature.

**Keywords:** Booth Multiplier, Effective Capacitance, 4:2 Compressor

## Introduction

With the exponential growth of portable devices operated on batteries, the demand for more and more processing power is increasing steadily, while keeping their power consumption to a minimum. Many such devices incorporate a hardware multiplier for performing fast arithmetic computations. In this context, power minimization of the multiplier plays a significant role. In the present era of CMOS technology, the three major sources of power dissipation are dynamic, short circuit and leakage (Soudris *et al.*, 2002). Generally, power reduction techniques aim at minimizing all the above mentioned power dissipation sources, but our emphasis here is on dynamic power dissipation as it dominates other power dissipation sources in digital CMOS circuits. The switching or dynamic power dissipation occurs due to charging and discharging of capacitors at different nodes in a circuit (Benini and Micheli, 1998).

The average dynamic power consumption of a digital circuit with N nodes is given by (Najm, 1994):

$$P_{dyn} = \frac{1}{2} \left( \sum_{i=1}^N \alpha_i C_i \right) V_{dd}^2 f_c \quad (1)$$

where,  $V_{DD}$  is the supply voltage,  $C_i$  is the load capacitance at node  $i$ ,  $f_c$  is the clock frequency and  $\alpha_i$  is the switching activity at node  $i$ . The product of switching

activity and load capacitance at a node is called effective capacitance. Assuming only one logic change per clock cycle, the switching activity at a node  $i$  can be defined as the probability that the logic value at the node changes from either 0 to 1 or from 1 to 0 between two consecutive clock cycles. For a given logic element, the switching activity at its output(s) can be computed using the static probability on its inputs and is given by:

$$a_i = 2P_i\bar{P}_i = 2P_i(1 - P_i)$$

where,  $P_i$  and  $\bar{P}_i$  denote the static probability of occurrence of a “one” and “zero” at node  $i$  respectively. When  $P_i = 0.5$ , the switching activity at a node is maximum and it decreases as it goes towards the two extremes (i.e., both from 0.5 to 0 and 0.5 to 1).

The two main low power design strategies used for dynamic power reduction are based on (i) supply voltage reduction and (ii) effective capacitance minimization. The reduction of supply voltage is the most aggressive technique because the power savings are significant due to the quadratic dependence on  $V_{DD}$ . Although such reduction is usually very effective, it increases leakage current in the transistors and also decreases circuit speed. The minimization of effective capacitance involves reducing switching activity or node capacitance. The node capacitance depends on the integration technology

used. To reduce switching activity only requires a detailed analysis of signal transition probabilities and implementation of various circuit level design techniques, such as logic synthesis optimization and balanced paths. It is independent of technology used and less expensive. Admiring the advantages of switching activity reduction, this paper focuses on switching activity reduction techniques in a multiplier.

Many different types of multipliers are available in literature. The one we are concerned in this study is a multiplier using modified Booth algorithm. In a Booth coded multiplier, multiplication is done in three separate computation steps. The first step is to generate all partial products in parallel using Booth recoding. In the second step these partial products are reduced to two operands using a number of reduction stages. These stages follow one after the other, feeding the output of one stage to the next. The final step is adding the two operands using a Carry Propagate Adder (CPA) to get the final sum. Power reduction can be applied in all three stages of the multiplication process. But our main focus is the second step, power reduction in the partial product reduction stage.

This paper is organized as follows: We start with an introduction to power consumption in multipliers. Related research on power reduction in multipliers is given in Section 2. Section 3 elaborates our proposed method for power reduction. Section 4 gives details of actual wiring patterns used in the multiplier. Simulation results are given in Section 5 and conclusion in Section 6.

## Related Research

Many researchers have proposed low power multiplier architectures by reducing power consumption in the partial product reduction stage (Oskuii, 2007; Ohban, 2002; Ito *et al.*, 2003; Chen *et al.*, 2003). Historically the partial product reduction stage was implemented using carry save adders based on Wallace or Dadda rules (Parhami, 2010). The carry save adders used are either Full Adders (FA) or Half Adders (HA). To illustrate this a 6×6 unsigned multiplier using a modified Dadda reduction tree is shown in Fig. 1 (Oskuii, 2007). Stage 1 is the rearranged 6×6 unsigned partial product array obtained by partial product generator of a multiplier. At every stage the number of bits with the same order (bits in a column) are grouped and connected to adder cells using Dadda's rules. Each column represents partial product bits of a certain magnitude. A sum output of a FA or HA at one stage will place a dot in the same column at the next stage and a carry output in the column to the left on the next stage (i.e., one order of magnitude higher).

The use of FAs and HAs in the reduction stages, in general, produce irregular layout and increase wiring

complexity. Wiring complexity results in more power. Weinberger (1981) proposed a new module called 4:2 compressor to overcome this, which can add 4 bits together with a carry (Weinberger, 1981). The majority of multiplier designs today make use of 4:2 compressors in the partial product reduction stage to increase the performance of the multiplier. The use of 4:2 compressors decrease the wiring capacitance due to a more regular layout, thereby contributing to fewer transitions in the reduction tree which results in reduced power. Hsiao *et al.* (1998) proposed a modified design of the 4:2 compressor that claimed improvements in both delay and power dissipation compared to earlier designs (Hsiao *et al.*, 1998). Several logic and circuit level optimizations are possible for reducing the number of transitions in the partial product reduction stage using higher order compressors instead of simple FA cells and 4:2 compressors.

Ohban (2002) proposed a low power multiplier using bypassing technique (Ohban, 2002). The main idea of his approach was to minimize the signal transitions while adding zero valued partial products. This is done by bypassing the adder stage whenever the multiplier bit is zero. Ito *et al.* (2003) proposed an algorithm using operand decomposition technique (Ito *et al.*, 2003). They decomposed multiplicand and multiplier into four operands, which result in twice the number of partial products compared to a conventional multiplier. By doing this, they reduced the one probability of each partial product bit to  $1/8$  while it is  $1/4$  in the conventional multipliers. This in turn decreases the switching probability. Chen *et al.* (2003) proposed a multiplier based on effective dynamic range of the input data (Chen *et al.*, 2003). If the data with smaller effective dynamic range is Booth coded, then the partial products have greater chance to be zero and decreases the switching activities of partial products. Fujino and Moshnyaga (2003) proposed a multiply accumulate design using dynamic operand transformation technique in which current values of the input are compared with previous values (Fujino and Moshnyaga, 2003). If more than half of the bits in an operand change, then it is dynamically transformed to its two's complement in order to decrease the transition activity during multiplication. Chen and Chu (2007) proposed a low power multiplier, which uses Spurious Power Suppression Technique (SPST) equipped Booth encoder (Chen and Chu, 2007). The SPST uses a detection logic circuit to detect whether the Booth encoder is performing redundant computations which results in zero partial products and stops the partial product generation process. To implement the proposed techniques in all the above mentioned multiplier architectures not only increase hardware complexity but also introduce additional delay in the operation. Also, the extra circuitry consumes additional power.

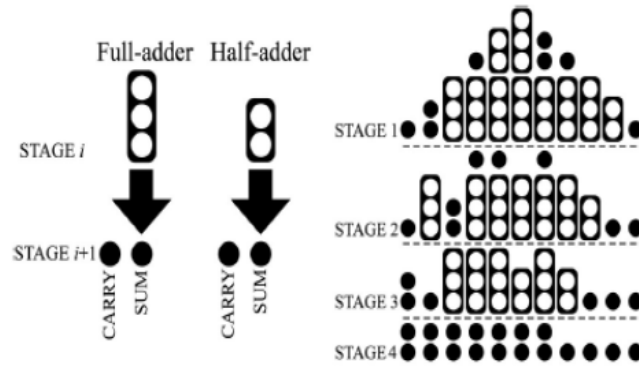


Fig. 1. Modified Dadda reduction tree for 6×6 unsigned multiplier (Oskuii, 2007)

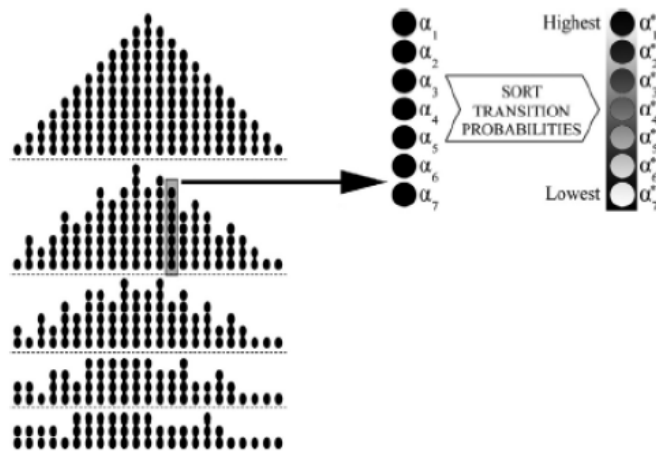


Fig. 2. Example of the reduction tree design and assignment to full adders (Oskuii, 2007)

Oskuii (2007) proposed a heuristic algorithm to reduce power consumption in the partial product reduction stage based on static probabilities on primary inputs (Oskuii, 2007). At every reduction stage, the number of bits with the same order of magnitude (bits in a column) are grouped together and connected to the adder cells in a Dadda tree. The selection of these bits and their grouping influences the overall switching activity of the multiplier. This was illustrated in Oskuii's paper which is described below.

- Only one column per stage is considered. As the generated carry bits from adders propagate from LSB towards MSB, optimization of columns is performed from LSB to MSB and from first stage to last stage. Thus it can be ensured that the optimization of columns and stages that has already been performed will still be valid when later optimizations are being performed
- Glitches and spurious transitions spread in the reduction stages after a few layers of combinational logic. To avoid them is not feasible in most cases.

Therefore, it seems beneficial to assign short paths to partial products having high switching activity

The goal of Oskuii (2007) was to reduce power in Dadda trees. The one probabilities for sum and carry of the FA and HA were calculated from their functional behavior. According to Oskuii's algorithm, the switching probabilities of partial products in a particular stage are calculated using the previous stage one probabilities in each column and they arranged these partial product bits in ascending order. The lower switching probability bits are used to feed full and half adders in the same stage and the higher switching probability bits are moved to the next stage. From the set of bits to feed adders they connected the highest switching probability signal to the carry input of the full adder as its path in a full adder is shorter than the other two inputs.

Figure 2 gives an example where 7 bits with the same order of magnitude are to be added (Oskuii, 2007). This is shown in Fig. 2 as the shaded box in the 2nd group of bits from the top. According to Dadda rules of reducing partial products, two full adders must be used and one bit

will be passed to the next stage together with the sum and carry bits generated by the full adders.  $\alpha_i$ 's denote the switching probabilities of the seven bits for  $i$  from 1 to 7. These are sorted in ascending order and listed as  $a_i^*$ , with the highest one as  $a_1^*$ . According to Oskuii's approach, the bit with the highest switching activity is kept for the next stage, i.e.,  $a_1^*$  in Fig. 2 and assign  $a_2^*$  and  $a_3^*$  to the carry inputs of the two FAs as their path is shorter and other bits to remaining inputs of FAs in any order. In this way the partial product tree was reduced by bringing the highest transition probability bits more closer to the output such that it reduces the total power in the multiplier without any additional hardware cost. Oskuii (2007) claimed that power reduction varying from 4 to 17% in multiplier designs could be achieved using their approach (Oskuii, 2007). On careful analysis of Oskuii's work we noticed that further reduction in power can be achieved by using 4:2 compressors. This will be achieved without introducing any additional delay or additional hardware.

## Proposed Design

Our design also uses a Partial Product Generator (PPG) for the  $n \times n$  multiplier based on radix-4 Booth encoder and generates all partial products. These partial products are then reduced to two operands by employing several Partial Product Reduction Stages (PPR).

We used a combination of 4:2 compressors, FAs and HAs in reduction stages. At each stage modified Dadda rules are applied to obtain operands for the next stage. While minimizing the partial product bits in each column, emphasis was given on higher speed and lower power. Higher speed is achieved by allowing the partial product bits to pass through a minimum number of reduction stages, while minimizing the final carry propagate adder length to the minimum.

Figure 3 illustrates the proposed partial product reduction scheme for a  $16 \times 16$  multiplier. Nine partial products obtained by PPG are reduced to two operands using three reduction stages. The vertical green boxes in each column represent 4:2 compressors. It takes five bits and reduce them into three output bits, one sum in the same column and two carry bits in the next higher significant column (one bit left) of next stage. The vertical red boxes represent full adder cells that reduce three partial product bits in a column to two, the sum and carry. Similarly, the vertical blue boxes represent half adder cells and add two partial product bits and generate two output bits, sum and carry. The order in which the inputs are fed to 4:2 compressor, full adder and half adder is discussed in Section 4. In Fig. 3 the maximum number of partial products in a column is 8 (columns 14 to 17). Since we are using 4:2 compressors that can take up to 5 input bits, when we reduce the partial products in

the first stage, we want to make sure that the maximum number of partial products in the next stage is only 5. This way we can reduce the bits in each column in stage 2 using one level of 4:2 compressors and in the third stage, we want to ensure that the maximum number of bits in any column is 3, so that 3:2 compressors (FA) can be used to add them. This will permit the whole reduction process to be achieved in 3 stages. The half adder in column 2 in reduction stage 1 and the full adder in column 3 in reduction stage 2 are used so as to minimize the size of the final carry propagate adder.

## Power Reduction

Once the maximum number of reduction stages is established for a design, the next criterion is to minimize power consumption. This is achieved by delayed passing and reducing the effective capacitance at every node in the reduction stages following Oskuii's rules (discussed in Section 2). Hence to minimize power, the design must ensure that the switching activity of nodes with higher capacitance must be kept to a minimum. This is achieved by the special interconnection pattern in our design. The higher switching activity signals are wired to nodes with lower capacitance and vice versa. This selective interconnection of signals to the inputs of 4:2 compressors, FAs and HAs minimizes the overall power consumption.

The logic diagram and input capacitances for a full adder are shown in Fig. 4a. In the following discussion we will assume that each and every input lead to a logic gate is considered as one unit load. Hence if a signal is connected to the inputs of two logic gates, then the load is two units. From the logic diagram of the full adder in Fig. 4a, input B is connected only to an XOR gate, whereas inputs A and C are connected to both an XOR and a MUX. Hence the input capacitance seen by B input is smaller than the other two inputs. The load presented at the B input is one unit load, while the loads presented at A and C are 2 unit loads. This is represented by the capacitance value C1 (1 unit load) and C2 (2 unit loads) as shown in Fig. 4a. Hence a transition on input B will result in less effective capacitance. Again by comparing the three inputs, the C input goes through only one logic device (XOR or MUX) before it reaches the output, whereas both A and B go through two logic devices before reaching the output. Hence a transition on any of the inputs A or B could result in output transitions on all three logic devices. But a transition on input C will affect only two of these logic devices. Therefore, we can conclude that even though the inputs A and C represent the same load, the overall effective capacitance on the full adder due to C input will be less than that due to A input. Hence, as a rule of thumb, the first two higher transition inputs among a set of three inputs that are given to a full adder should be connected to the B and C inputs and the least one to A.

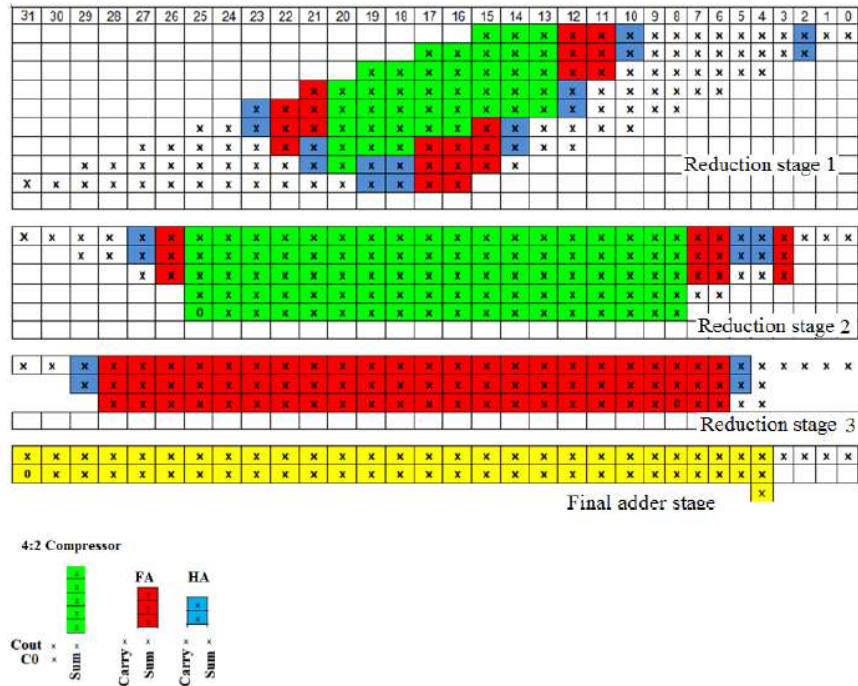


Fig. 3. Proposed PPG scheme for a 16x16 multiplier

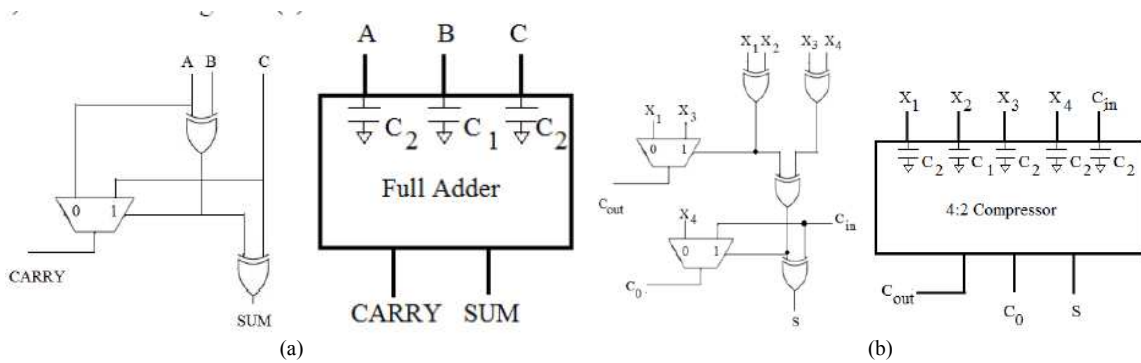


Fig. 4. Full adder and 4:2 compressor (a) Full adder logic and input capacitances (b) 4:2 compressor logic and input capacitances

Similarly, the logic diagram of a 4:2 compressor and its input capacitances are shown in Fig. 4b. The input capacitances seen by  $X_1$ ,  $X_3$ ,  $X_4$  and  $C_{in}$  are twice that seen by  $X_2$ . Hence the highest transition probability signal must be connected to  $X_2$  input. Again, by using a similar argument as in the full adder, the second highest transition probability signal must be given to  $C_{in}$ . The remaining inputs are given to  $X_1$ ,  $X_3$  and  $X_4$  in any order. This minimizes the overall effective capacitance in the 4:2 compressor.

The probability of a logic one at the output of any block is a function of the probability of a logic one at its inputs (Parker and McCluskey, 1975; Cirit, 1987). From the logic functions of 4:2 compressor, FA and HA we can calculate their output probabilities knowing their input probabilities. Table 1 shows the algebraic equations for

calculating the output probabilities for a full adder and half adder, where  $P_A$ ,  $P_B$  and  $P_C$  represent the static 1 probabilities of inputs A, B and C respectively. Similarly, Table 2 shows the equations for a 4:2 compressor. By comparing Table 1 and Table 2 we can conclude that the statistical probabilities of the output signals of basic elements (4:2 compressors, full adders and half adders) used in partial product reduction stages vary.

Table 3 shows the output signal probabilities of 4:2 compressor, full adder and half adder, assuming equal static '1' probabilities of 0.25 for all inputs. In each partial product reduction stage the signals in a particular column have different switching probabilities. The output signal of one stage becomes inputs to the next stage. So the switching probabilities of the outputs diverge more as we move down the partial product reduction stage.



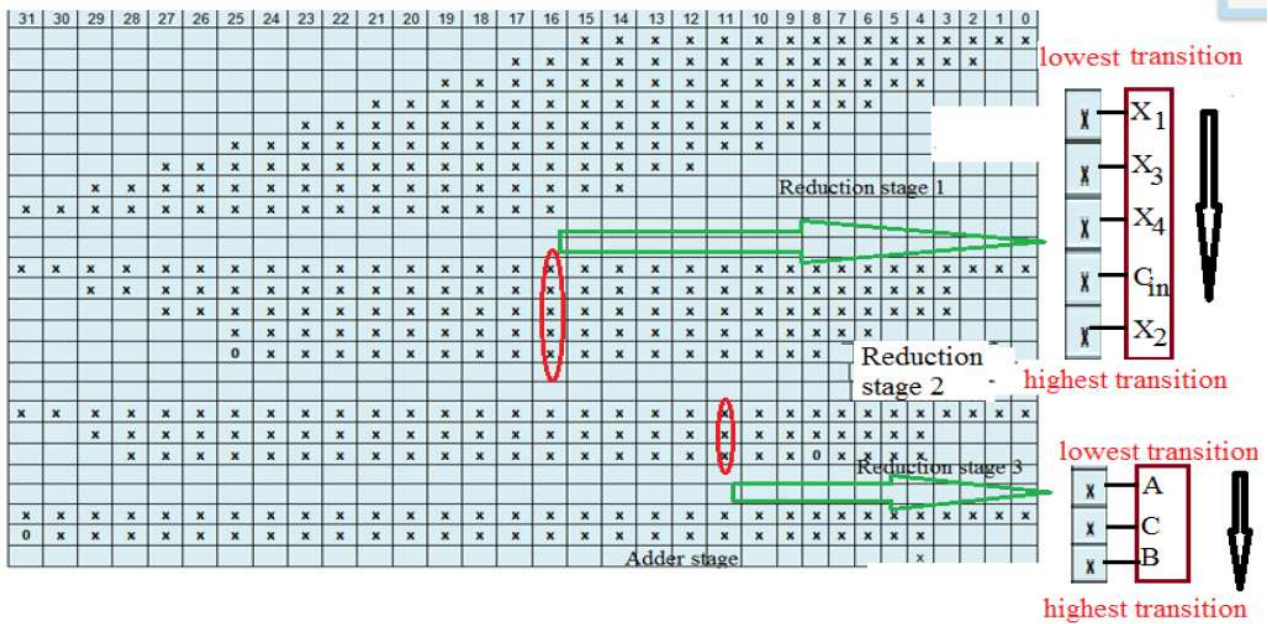


Fig. 5. Wiring patterns for 4:2 compressors and full adders

Table 1. Probability equations for full adder and half adder

	Full adder	Half Adder
Sum	$A \oplus B \oplus C$	$A \oplus B$
Carry	$(A \oplus B).C + \overline{(A \oplus B)}.A$	$A.B$
$P_{Sum}$	$P_A + P_B + P_C + 4P_A.P_B.P_C - 2.(P_A.P_B + P_B.P_C + P_C.P_A)$	$P_A + P_B - 2.P_A.P_B$
$P_{Carry}$	$P_A.P_B + P_B.P_C + P_C.P_A - 2.P_A.P_B.P_C$	$P_A.P_B$

Table 2. Probability equations for 4:2 Compressor

	4:2 Compressor
Sum	$X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus C_{in}$
$C_{Out}$	$(X_1 \oplus X_2).X_3 + \overline{(X_1 \oplus X_2)}.X_1$
$C_O$	$(X_1 \oplus X_2 \oplus X_3 \oplus X_4).C_{in} + \overline{(X_1 \oplus X_2 \oplus X_3 \oplus X_4)}.X_4$
$P_{Sum}$	$P_{X_1} \oplus P_{X_2} \oplus P_{X_3} \oplus P_{X_4} \oplus P_{C_{in}}$
$P_{C_{Out}}$	$(P_{X_1} \oplus P_{X_2}).P_{X_3} + \overline{(P_{X_1} \oplus P_{X_2})}.P_{X_1}$
$P_{C_O}$	$(P_{X_1} \oplus P_{X_2} \oplus P_{X_3} \oplus P_{X_4}).P_{C_{in}} + \overline{(P_{X_1} \oplus P_{X_2} \oplus P_{X_3} \oplus P_{X_4})}.P_{X_4}$

Table 3. Output probabilities of 4:2 compressor and adder cells

Input signal probabilities = 0.25

4:2 Compressor	Full adder	Half adder
Sum 0.4844	Sum 0.4375	Sum 0.375
$C_{Out}$ 0.1563	Carry 0.1563	Carry 0.0625
$C_O$ 0.2266		

Table 4. Power reports from simulation

Design	Quiescent power (mW)	Dynamic power (mW)	Total power (mW)
Ours	82.57	360.74	443.31
Oskuii	82.57	454.06	536.63
Wallace	82.67	475.08	557.75

Table 5. Power delay products of different multipliers

Design	Total Delay (nS)	Power (mW)	Power-Delay Product (nJ)
Ours	30.88	443.31	13.69
Oskuii	31.21	536.63	16.75
Wallace	35.27	557.05	19.65

Several reduction stages are required to reduce the partial products generated in a parallel multiplier. As shown in Fig. 3, at each stage a number of bits with the same order of magnitude are grouped together and connected to the 4:2 compressors and adder cells. As mentioned earlier, the selection of these bits and their grouping influences the overall switching activity of the multiplier. This is what we exploited to reduce the overall switching activity of the multiplier.

Figure 5 shows the array structure of the proposed partial product reduction tree. In the following we assumed that the one probabilities of all 9 partial product bits are the same and are equal to 0.25. These 9 partial products are reduced to 2 operands in three stages. In stage 1, we used 4:2 compressors, full and half adders and reduced the number of operands to 5. The bits in these 5 operands will have different one probabilities. Using their one probabilities we can find their switching probability. If we look at each column, all the bits in a column have the same weight but different one probabilities. So we have enough freedom to choose any of these signals and connect them to any of the inputs of the basic logic devices in the next stage. The way these signals are wired to the logic devices to achieve power reduction will affect the total power consumption of the multiplier.

Figure 5 also shows how we wired the input signals to 4:2 compressors and full adders in the proposed design. In column 16 of reduction stage 2, we have five bits with the same order of magnitude, which are to be added. From the set of 5 inputs that are fed to the 4:2 compressor, the first higher transition bit is fed to X2 input and the next higher transition bit is fed to C<sub>in</sub>, as they provide lowest switching activity when compared to others. The remaining bits are fed to X1, X3 and X4 in any order.

Similarly, Fig. 5 also shows column 11 in reduction stage 3, in which three bits of the same order are to be added. The highest transition bit is given to B input of the full adder and the next higher transition bit is fed to C input. The third one is fed to A input. With this type of reordering of the inputs, we can decrease the output switching probabilities of compressors and adders. By applying the same technique at every stage we reduced the overall switching power of the multiplier.

## Simulation

Power analysis was done by synthesizing our 16×16 multiplier on Spartan-3E FPGA and using XPOWER

Analyzer tool. We evaluated the performance of our 16×16 multiplier by comparing with the conventional Wallace and Oskuii's multipliers.

Table 4 shows the quiescent and dynamic powers of different multipliers obtained by simulation. The quiescent power is almost the same for all multipliers. The dynamic power for our design is 360.74 mW, where as Oskuii's and Wallace's multipliers consume 454.06 and 475.08 mW respectively. The total power consumption for our multiplier is 443.31 mW, which is less by 17.39 and 20.51%, compared to Oskuii's and Wallace multipliers. Table 5 shows the Power Delay Products (PDP) of different multipliers. For our design it is 13.69 nJ as compared to Oskuii's and Wallace's designs with 16.75 and 19.65 nJ respectively. Thus our design has the lowest power delay product compared to both Oskuii's and Wallace multipliers.

## Conclusion

We did an investigation of the power consumption on multipliers, along with some techniques for the minimization of power. Our main contribution is directed towards reducing switching power in multipliers, especially in the partial product reduction stage using 4:2 compressors, full adders and half adders. The switching probabilities of different bits of the same order of magnitude vary as we move down the tree. Hence a reordering of the partial product bits to the inputs of logic modules was done based on their switching probabilities, which resulted in reduced power. We achieved the lowest power consumption of 443.31 mW and a PDP of 13.69nJ for a 16×16 multiplier implemented on Spartan-3E FPGA, as compared to two other designs in the literature. Further research could evaluate extending the proposed interconnection technique to the partial product reduction stage by employing higher order compressors such as 5:2, 9:2, 28:2, etc. In this manner, different architectures using various combinations of compressors in the partial product reduction stage can be compared so as to select the best one with the lowest power dissipation for any multiplier.

## Acknowledgement

The authors acknowledge the support provided by the State University of New York, New Paltz, USA in completing this study.

## Author's Contributions

**Nageshwar Reddy Peddamgari:** Made considerable contributions to design, analysis and interpretation of the proposed design of multipliers. Contributed in all simulations, data analysis and writing of this manuscript.

**Damu Radhakrishnan:** Made considerable contribution to conception and design. Analysis of multiplier operation, verifying design and reviewing the article critically for significant intellectual content. Gave final approval of the paper to be submitted.

## Ethics

This article is original and contains unpublished material. The corresponding author confirms that the other author has read and approved the manuscript and no ethical issues are involved.

## References

- Benini, L. and G.D. Micheli, 1998. *Dynamic Power Management: Design Techniques and CAD Tools*. Kluwer Academic Publishers, MA., ISBN: 079238086X.
- Chen, K.H. and Y.S. Chu, 2007. A low power multiplier with spurious power suppression technique. *IEEE Trans. Very Large Scale Integrat. Syst.*, 15: 846-850.
- Chen, O.T., S. Wang and W. Yi-Wen, 2003. Minimization of switching activities of partial products for designing low-power multipliers. *IEEE Trans. VLSI Syst.*, 11: 418-433.
- Cirit, M., 1987. Estimating dynamic power consumption of CMOS circuits. *Proceedings of International Conference on Computer Aided Design*, Nov. 9-12, IEEE Computer Society Press, USA, pp: 534-537.
- Fujino, M. and V.G. Moshnyaga, 2003. Dynamic operand transformation for low-power multiplier-accumulator design. *Proceedings of the International Symposium on Circuits and Systems*, May 25-28, IEEE Xplore Press, Bangkok, pp: 345-348.
- Hsiao, S.F., M.R. Jiang and J.S. Yeh, 1998. Design of high-speed low-power 3-2 counter and 4-2 compressor for fast multipliers. *Electron. Lett.*, 34: 341-342.
- Ito, M., D. Chinnery and K. Keutzer, 2003. Low power multiplication algorithm for switching activity reduction through operand decomposition. *Proceedings of 21st International Conference on Computer Design*, Oct. 13-15, IEEE Xplore Press, pp: 21-26.
- Najm, F., 1994. A Survey of power estimation techniques in VLSI circuits. *IEEE Trans. VLSI Syst.*, 2: 446-455.
- Ohban, J., 2002. Multiplier energy reduction through bypassing of partial products. *Proceedings of the Asia-Pacific Conference on Circuits and Systems*, Oct. 28-31, IEEE Xplore Press, pp: 13-17.
- Oskuii, S.T., 2007. Transition-activity aware design of reduction-stages for parallel multipliers. *Proceedings of the 17th ACM Great Lakes symposium on VLSI*, Mar. 11-13, ACM, Stresa-Lago Maggiore, Italy, pp: 120-125. DOI: 10.1145/1228784.1228817
- Parhami, B., 2010. *Computer Arithmetic Algorithms and Hardware Designs*, 2nd Ed. Oxford Univ. Press, New York, ISBN-10: 978-0-19-532848-6.
- Parker, K. and E.J. McCluskey, 1975. Probabilistic treatment of general combinational networks. *IEEE Trans. Comput.*, 24: 668-670.
- Soudris, D., C. Piguet and C. Goutiset, 2002. *Designing CMOS Circuits for Low Power*. 1st Edn., Kluwer Academic Press, The Netherlands, ISBN10: 1-4020-7234-1.
- Weinberger, A., 1981. 4:2 Carry save adder module. *IBM Tech. Disclosure Bull.*, 23: 3811-3814.