

Booth multiplier



**ASSIGN
BUSTER**

Low Power Booth Multiplier by Effective Capacitance Minimization P. Nageshwar Reddy Dr. Damu Radhakrishnan Stu. in SUNY, New Paltz, NY Prof. in SUNY, New Paltz, NY Abstract: In this paper we present an energy efficient parallel multiplier design based on effective capacitance minimization. Only the partial product reduction stage in the multiplier is considered in our research. The effective capacitance is the product of capacitance and switching activity. Hence to minimize the effective capacitance in our design, we decided to ensure that the switching activity of nodes with higher capacitances is kept to a minimum.

This is achieved in our design by wiring the higher switching activity signals to nodes with lower capacitance and vice versa for the 4: 2 compressor and full adder cells, assuming the initial probability of each partial product bit as 0. 25. This reduced the overall switching capacitance, thereby reducing the total power consumption in the multiplier. Power analysis is done by synthesizing our design on Spartan-3E FPGA and used XPower Analyzer tool that is provided in ISE Xilinx 10. 1. The dynamic power for our 16? 16 multiplier was measured as 360. 4mW, and the total power 443. 31mW. 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 multiplier presented in the literature. Index Terms- Booth multiplier, Effective capacitance, 4: 2 compressor. 1. Introduction A multiplier is the most frequently used fundamental arithmetic unit in various digital systems such as computers, process controllers and signal processors. Thus it has become a major source of power dissipation in these digital systems.

With the exponential growth of portable systems that are operated on batteries, power reduction has become one of the primary design constraints in recent years. In the present era, each and every electronic device is implemented using CMOS technology. The three major sources of power dissipation in digital CMOS circuits are dynamic, short circuit and leakage [1]. Generally, power reduction techniques aim at minimizing all the above mentioned power dissipation sources but our emphasis 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 the charging and discharging of capacitors at different nodes in a circuit [2]. The average dynamic power consumption of a digital circuit with N nodes is given by: where V_{DD} is the supply voltage, C_i is the load capacitance at node i , f_{CLK} is the clock frequency and α_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 ($0 \rightarrow 1$ or $1 \rightarrow 0$) between two consecutive clock cycles. For a given logic element, the switching activity at its output(s) can be computed using the probability of its inputs and is given by: where P_i and $P_{\bar{i}}$ denote the 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 extreme values (i. e. both from 0.5 to 0 and 0.5 to 1). The two main low power design strategies for dynamic power reduction are based on (i) supply voltage reduction and (ii) the effective capacitance

minimization. The reduction of supply voltage is one of the most aggressive techniques because the power savings are significant due to the quadratic dependence on VDD. Although such reduction is usually very effective, it increases leakage current in the transistors and also decreases circuit speed. The minimization of effective switching 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 the technology used and is less expensive. Admiring the advantages of switching activity reduction, this paper focuses on switching activity reduction techniques in a multiplier. Digital Multiplication is done in three steps in a Booth coded multiplier.

The first step is to generate all the partial products in parallel using Booth recoding. In the second step these partial products are reduced to 2 operands in several stages by applying Wallace/Dadda rules. 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 to produce the final sum. Our main focus in this paper is the second step, partial product reduction. Fig. 1 shows the modified Dadda reduction tree for a 6? 6 unsigned multiplier, which uses full adders (FA) and half adders (HA) as basic elements.

Stage 1 is the rearranged 6? 6 unsigned partial product array obtained using the partial product generator. At every partial product reduction(PPR) stage

the number of bits with the same order (bits in a column) are grouped together and connected to adder cells following Dadda's rules. Each column represents partial products of a certain magnitude. The sum output of a FA or HA at one stage will place a dot in the same column at the next stage and an output carry in the column to the left in the next stage (i. e. one order of magnitude higher). Fig. 1. Modified Dadda reduction tree for 6? unsigned multiplication The Wallace and Dadda designs use only FAs and HAs in the reduction stages, which form an irregular layout and increases wiring complexity. Wiring complexity is a measure of power. Since then Weinberger [3] has proposed a 4: 2 compressor, the majority of the multiplier designs today make use of 4: 2 compressors to increase the performance of the multiplier. They also contribute to power reduction as they decrease the wiring capacitance due to a more regular layout, contributing to fewer transitions in the partial product reduction tree. It also reduces hardware cost.

The design of the 4: 2 compressor got improved in time, and modified design presented by Jiang et al. claimed improvements in both delay and power dissipation compared to earlier designs [4]. Several logic and circuit level optimizations are possible by using higher order compressors instead of simple FA cells for reducing the number of transitions in the partial product reduction stage. Because of this we used 4: 2 compressors, FA (3: 2 compressor) and HA cells in our partial product reduction stages. We reduced the switching activity by minimizing the effective capacitance at every node in the circuit.

This stands as the main focus of this paper. This paper is organized as follows: related research in section 2 and 2. Related Research Many researchers have elucidated different low power multiplier architectures by using different techniques to reduce the total switching activity in a multiplier [1]-[4]. Ohban, et al. proposed a low power multiplier using the so called bypassing technique [5]. The main idea of their approach is 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.

Masayuki, et al. proposed an algorithm using operand decomposition technique [6]. They decomposed the multiplicand and the multiplier into 4 operands and using them they generated twice the number of partial products compared to the 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. proposed a multiplier based on effective dynamic range of the input data [7].

If the data with smaller effective dynamic range is Booth coded then the partial products have greater chances to be zero, which decreases the switching activities of partial products. Fujino, et al. proposed a multiply accumulate design using dynamic operand transformation technique in which current values of the input is compared with previous values [8]. 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, et al. proposed a low power multiplier,

which uses spurious power suppression technique (SPST) equipped Booth encoder [9]. The SPST uses a detection logic circuit to detect whether the Booth encoder is calculating redundant computations which yield in Zero partial product and stops such PP generation process. To implement the basic principles used in all the above mentioned multiplier architectures not only increase hardware intensity but also introduce delay in the operation. Also the extra circuitry employed to implement them consumes power.

So our research interest is focused on techniques which decrease power without introducing any delay and additional hardware. Oskui et al. proposed an algorithm based on static probabilities at the primary inputs [10]. At every PP 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 Oskui's paper by referring to an early work, which is described below. Only one column per stage is considered here. 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 stage 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.

Oskuii's goal was to reduce the power in Dadda trees. The one probability for sum and carry of the FA and HA can be calculated from their functional behavior [10]. According to Oskuii's algorithm, assuming the switching probabilities of partial products in a particular stage are calculated using the previous stage one probabilities and in each column and they arranged these partial product bits in ascending order. They first use the lower switching probability bits to feed full and half adders and transfer the higher switching probability bits to the next stage.

From the set of bits to feed adders they tried to feed the highest switching probability signal to the carry input of the full adder as its path in full-adder is shorter than the other two inputs. Fig. 2. Example to illustrate Oskuii's approach [10] Fig. 2 gives an example where 7 bits with the same order of magnitude are to be added. This is shown as the shaded box in the 2nd group of bits from top in Fig. 2. According to Dadda rules of reducing a partial product tree, 2 FAs 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. s_i for i varying from 1 to 7 represent the switching probabilities of the seven bits. These are sorted in ascending order and listed as s_i^* with the highest one as s_1^* . According to their approach, the bit with highest switching activity is kept for the next stage i. e. in Fig. 3. 2, and assign and to the carry inputs of the two FAs as their path is shorter and the other bits to the remaining inputs of FAs in any order. In this way they reduced the partial product tree by bringing the highest transition probability bits more closer to the output such that it reduces the total power in the multiplier without any extra hardware cost.

Oskuii claimed that power reduction varying from 4% to 17% in multiplier designs could be achieved using their approach. On careful analysis of Oskuii's work we notice that further reduction in power can be achieved. This is elaborated in our design presented in the next section.

3. Proposed Work

By using a partial product generator (PPG) for the $n \times n$ multiplier employing radix-4 Booth encoder we obtained the required partial products. These partial products are then reduced to 2 operands employing several partial product reduction (PPR) stages. 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 using 4:2/3:2 compressors and HA cells, 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.

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

Fig. 3 shows the proposed partial product reduction scheme for a 16×16 parallel multiplier.

Nine partial products obtained by PPG are reduced to 2 operands using 3 reduction stages. The vertical green boxes in each column represent 4:2 compressors. It takes five bits and reduces them into 3 output bits, one sum bit in the same column position and two carry bits in the next higher significant column (one bit left) of next stage. The vertical red boxes represent full adder cells, which reduce three partial product bits in a column and generate the sum and carry bits. Similarly, the vertical blue boxes

represent half adders and add two partial product bits to reduce it to 2 output bits.

The order in which the inputs are fed to 4: 2 compressor, full and half adders is discussed in the next section. 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, to 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 only 3, so that full adders 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. 4. Power Reduction Once the minimum number of reduction stages is established for a design, the next criterion is to minimize power consumption. This is achieved by delay passing and reducing the effective capacitance at every node in the reduction stages also following Oskui's rules (discussed in Section 2).

To minimize the effective switching activity, the design must ensure that the switching activity of nodes with higher capacitance value must be kept to a minimum. This is achieved by a special interconnection pattern used in our design. The higher switching activity signals are wired to nodes with lower capacitance and vice versa. Our multiplier design uses the above idea to minimize power. This paper therefore focuses on selective interconnection of

signals to the inputs of 4: 2 compressors and FAs and HAs using the above concept.

The logic diagram and the input capacitances for a full adder are shown in Fig. 4(a). For the following we will assume that each and every input lead to a logic gate is considered as one unit load ($C1$). Hence if a signal is connected to the inputs of two logic gates, then the load is two units ($C2$). From the logic diagram of the full adder in Fig. 4(a), 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 of the B-input is smaller than the other two inputs.

The load presented by the B input is one unit load, while the loads presented by A and C are 2 unit loads. Hence a transition on input B will result in less effective capacitance. This is represented by the capacitance values $C1$ (1 unit load) and $C2$ (2 unit loads) as shown in Fig. 4. 9. Again by comparing the three inputs, the C input goes through only one logic device (XOR gate or Mux) before it reaches the output, whereas both A and B goes 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 the 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 switching effect 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 last one to A. (a) (b) Fig. 4. a) FA logic diagram and input capacitances (b) 4: 2 compressor

logic diagram and input capacitances. Similarly, the logic diagram of a 4: 2 compressor and its input capacitances are shown in Fig. 4. (b). The input capacitances presented by X_1 , X_3 , X_4 and C_{in} are twice that presented by X_2 . Hence, the highest transition probability signal must be connected to the X_2 input. Again by using a similar argument as in the full adder, the second highest transition probability signal must be given to the 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 a 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 [11] [12]. From the logic functions of 4: 2 compressor, FA and HA we can calculate their output probabilities knowing their input probabilities. Table 2: Probability equations for 4: 2 Compressor | 4: 2 Compressor| PSUM| | PCout| | PC0| | Table 1 shows the probability expression for the sum and carry outputs for the full adder and half adder in terms of their input signal probabilities. The 4: 2 compressor output probabilities are shown in Table 2. By comparing

Tables 1 and 2 we can say 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 '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 signals of one stage become inputs to the next stage. So the switching probabilities of the outputs diverge more as we move down the partial production reduction stages.

Table 3. 1: Output Signal Probabilities of FAs and HAs | Full-adder| Half adder| SUM| | | CARRY| | A. B| PSUM| | | PCARRY| | | Table 3: Output probabilities of 4: 2 compressor and adder cells Input signal probabilities = 0. 25| 4: 2 compressor| Full adder| Half adder| SumCoutC0| 0. 48440. 15630. 2266| SumCarry| 0. 43750. 1563| SumCarry| 0. 3750. 0625| 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.

The selection of these bits and their grouping influences the overall switching activity of the multiplier. This is what we will exploit to reduce the overall switching activity of the multiplier. Fig. 5 shows the array structure of the proposed partial product reduction scheme for a 16? 16 multiplier. In the following we assumed that the one probability of all the 9 partial product bits are same and is equal to 0. 25 (as discussed in Section 3. 26). These 9 partial product bits are fed to 4: 2 compressors, full and half adders and are reduced to 5 operands. The bits in these 5 operands will have different one probabilities.

From these one probabilities we can calculate their switching probability. If we look at each column all the bits in that column have the same weight but different one probability. So we have enough freedom to choose any of these signals which can be connected to any of the inputs of the basic elements. The way these signals are wired to basic elements to achieve reduction will affect the total power consumption in a multiplier. Show an example Fig 5 shows how we wired the input signals to 4: 2 compressors and full adders in

the proposed design. To illustrate the principle consider column 16 of reduction stage 2 in Fig. , where we have five bits with the same order of magnitude, which are to be wired to the inputs of a 4: 2 compressor. The first higher transition bit is fed to X2 input and next higher transition bit is fed to Cin, as they provide lower switching activity when compared to others. The remaining three bits can be fed to X1, X3 and X4 in any order. Similarly on column 11 in reduction stage 3, three bits of the same order are to be added. The highest transition bit is given to B input of the adder and the next higher transition bit is fed to C input. The third bit is fed to A input.

This way of feeding the inputs, we can decrease the output switching probabilities of compressors and adders. By applying the same technique to every stage we can reduce the overall switching capacitance of the multiplier, thereby reducing power. Fig. 5. Wiring patterns for 4: 2 compressors and full adders

5. Simulation Power analysis was done by synthesizing our 16? 16 multiplier design on Spartan-3E FPGA and using XPower Analyzer tool provided in ISE Xilinx 10. 1. We evaluated the performance of our 16? 16 multiplier by comparing with the conventional Wallace and Oskui'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 only 360. 74 mW, where as Oskui's and Wallace multipliers consume 454. 06mW and 475. 08 mW respectively. Hence the total power consumption is only 443. 31mW for our multiplier, which is less by 17. 39% and 20. 51%, compared to Oskui's and Wallace multipliers. Table 4: Power reports from simulation for a 16? 16

Multiplier Design| QuiescentPower (mW)| DynamicPower (mW)| TotalPower (mW)| Our Design| 82. 7| 360. 74| 443. 31| Oskuii’s Design| 82. 57| 454. 06| 536. 63| WallaceMultiplier| 82. 67| 475. 08| 557. 75| Table 5 Power-Delay products of 16? 16 multipliers Design| Total Delay (ns)| Power (mW)| Power-Delay Product| Our Design| 30. 889| 443. 31| 13.693×10^{-9} | Oskuii’s Design| 31. 219| 536. 63| 16.753×10^{-9} | WallaceMultiplier| 35. 278| 557. 05| 19.651×10^{-9} | Table 5 shows the power-delay products of different multipliers. Smaller the power delay product of a multiplier the higher is its performance. Our design has the shortest delay of 30. 889ns, compared to 31. 219ns and 35. 78ns for Oskuii’s design and Wallace’s design respectively. Hence our design has the lowest power-delay product compared to both Oskuii’s and Wallace multipliers.

6. Conclusions We have presented an investigation of multiplier power dissipation, along with some techniques which allow reductions in power consumption for this circuit. Given the importance of multipliers, it is essential that further research efforts are to be directed in the following ways. * In this thesis the switching activity criteria for the interconnection pattern in 4: 2 compressors was used only for two of the inputs of the 4: 2 compressor.

The interconnections of signals on the other three inputs are made without any importance given to their switching activity. This is because at the gate level, the load capacitance at a node is measured simply based on the number of connections made at that node. In the 4: 2 compressor, three of the inputs are feeding two inputs each (except the carry input). Hence, we consider them with the same load capacitance. In reality, this is not true. To get an accurate estimate on capacitance, an actual layout of the cell has to

be made using VLSI layout tools and then their capacitances are to be extracted.

Hence further research could focus on the above so as to find an ordering for these inputs based on their capacitance values. Also, different implementations of 4: 2 compressors may be compared so as to select the one with the lowest capacitance values. * 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.

References 1] D. Soudris, C. Piguet, and C. Goutiset , Designing CMOS Circuits for Low Power. KluwerAcademicPress, 2002. [2] L. Benini, G. D. Micheli, et al. , Dynamic Power Management Design Techniques & CAD Tools. Norwell, MA: Kluwer Academic Publishers, 1998. [3] A. Weinberger, “ 4: 2 Carry Save Adder Module,” IBM Technical Disclosure Bulletin, vol. 23, 1981. [4] S. F. Hsiao, M. R. Jjiang, and J. S. Yeh, “ Design of High-Speed Low-Power 3-2 Counter and 4-2 Compressor for Fast Multipliers,” Electronics Let. , vol. 34, no. 4, pp. 341-342, 1998. [5] J. Ohban, “ Multiplier Energy Reduction Through Bypassing of Partial Products” in Proc. Asia-Pacific Conf. on Circuits and Systems, vol. 2, pp. 13-17, 2002. [6] M. Ito, D. Chinnery, and K. Keutzer, “ Low Power Multiplication Algorithm for Switching Activity Reduction Through Operand Decomposition,” 21st Int. Conf. on Computer Design, 2003. [7] O. T. Chen, S. Wang, and Yi-Wen Wu, “ Minimization of Switching Activities of Partial Products for Designing Low-Power Multipliers,” IEEE

Trans. on VLSI Syst. , vol. 11, pp. 418 – 433, 2003. [8] M. Fujino, and V. G. Moshnyaga, “ Dynamic Operand Transformation for Low-Power Multiplier-Accumulator Design,” in Proc. of the Int. symp. n circuits and systems, 2003. [9] K. H. Chen and Y. S. Chu, “ A Low Power Multiplier with Spurious Power Suppression Technique,” IEEE Trans. VLSI Syst. , vol. 15, no. 7, pp. 846-850, 2007. [10] S. T. Oskuii, “ Transition-Activity Aware Design of Reduction-Stages for Parallel Multipliers,” in Proc. of Great Lakes Symp. on VLSI, 2007. [11] K. Parker and E. J. McCluskey, “ Probabilistic Treatment of General Combinational Networks,” IEEE Trans. on Computers, C-24: 668-670, June 1975. [12] M. Cirit, “ Estimating Dynamic Power Consumption of CMOS Circuits” in Proc. of ICCAD, pp. 534-537, 1987.