Design for Power and Area Efficient Approximate Multipliers

SANDHI NARESH
MTech student, Dept of ECE, Siddhartha Institute of Engineering And Technology, Hyderabad, TS, India.

MALATHI NADDUNOORO
Assistant Professor, Dept of ECE, Siddhartha Institute of Engineering and Technology, Hyderabad, TS, India.

Abstract: **Multimedia and image processing applications, may tolerate errors in calculations but still generate meaningful and beneficial results. This work deals with a high speed approximate multiplier with TDM tree and carry prediction circuit. The modified multiplier utilizes an optimised TDM carry save tree which reduces the device utilization on FPGA as well as the combinational path delay and power consumption. The proposed design is analyzed using the simulation and implementation results on Xilinx Spartan 3E family.**

Keywords: **Approximate Carry Adder; Three Dimensional Reduction Method; Approximate Multiplier**

1. Introduction

Exact and precise models and algorithms are not always appropriate for proficient use in multimedia and image processing operations. The model of approximate calculation relies on entirely relaxing fully exact and completely deterministic building blocks while, designing energy-efficient systems. In digital designs, integer multiplication is one of the fundamental building blocks, which deeply affects the microprocessor and DSP performance. A faster digital circuit is obtained by implementing a speculative (prediction) approach. Speculative digital circuits are based on faster operation by employing a speculative functional unit, which is an arithmetic unit that employs a predictor for the carry signal, without actually waiting for the carry propagation. The speculative unit predicts the carry of the one or more cells used in the digital circuit without waiting for the actual carry propagation to take place. This is similar to a predictor in the microprocessor. Here we have considered a speculative multiplier which consists of a predictive carry-save reduction tree using three steps: partial products recoding, partial product partitioning and speculative compression. The speculative tree utilize (m: 2) counters, and are faster than traditional compressors based on half adders and full adders. The tree is further comprised of a fast carry-propagate adder and an error rectification circuit. Speculative multipliers have higher speed compared to their conventional counterparts.

2. Previous Works

It Show that approximate circuits have higher performance as compared to precise logic circuits. Many inexact multipliers have been proposed in the literature [4] [6] [7] [13]. These designs employ a truncated multiplication method. In [6], an inexact array multiplier is used, by ignoring selected least significant bits in partial products. A inexact multiplier with correction constant has been proposed in [13]. A variable correction constant inexact multiplier is proposed in [4].This method modifies the correction term according to column n-k-1. If partial products in column n-k-1 are one, then correction factor is increased and, if all partial products in the above column are zero, the correction factor is decreased. In [7], a basic 2x2 multiplier block is suggested for constructing larger multiplier arrays. In all these designs the area was found to be very high. In [11] another approximate multiplier with two approximate [4:2] compressor has been proposed. This multiplier requires lesser area as compared to multipliers using truncation technique however the error percentage was found to be very high.

[12] Describes another approximate multiplier design which utilizes prediction units for the carry signal and also has lesser error percentage as compared to [11]. SFUs (Speculative Functional Units) are prediction circuits that can be considered as black box entities which are faster than their non -speculative counterparts, independently of the particular implementation [8]. Hence approximate multipliers using SFUs also aim to achieve delay improvements, at the same time introducing less power and area overheads. This multiplier utilises Carry Save Adder (CSA) tree [14] for partial product reduction, wherein the carry outputs are propagated rather than being preserved thereby reduces the delay. Popular CSA schemes include Wallace tree and Dadda multiplier. Wallace tree [1] [9] result in long and irregular wires along the columns to connect to the CSA. The wire capacitance in turn increases the delay and energy of the multiplier and the wires are difficult to layout. Dadda refined Wallace's method by introducing a counter placement strategy that requires few numbers of counters in the reduction stage but at the cost of larger Carry propagate Adder (CPA) [2] [9]. The delay from an input to an output in a full adder is not the same. This delay is dependent on a particular transition (0-to-1, 1-to-0). Therefore it is also possible to come up with different realizations of a full adder wherein a specific signal path is favored with respect to the others and has been designed in such a way that a signal propagation of this path takes a minimal amount of time [3]. The CSA scheme which takes

care of this delay in transition is Three Dimensional Scheme (TDM) [3], where partial product array is represented in space and time. This is followed by a speculative adder [5].

3. Proposed method

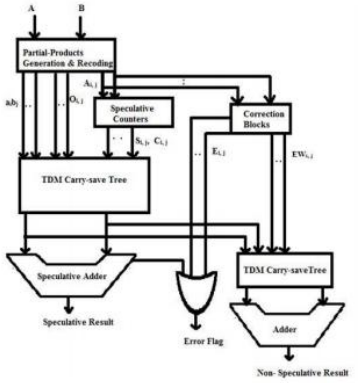

Fig. 1. Architecture of Approximate Multiplier

3.1 Partial Product Recoding

Consider two partial products aibj and ajbi of the i+j-th column of the PPM. Now we will define two modified partial products:

Ai, j = aibj AND ajbi (1)

Oi, j = aibj OR ajbi (2)

Thus couple of partial products aibj and ajbi can be replaced with modified partial products Ai, j and Oi, j. The advantage of introducing such a recoding technique is the introduction of lower probability terms in the PPM. The probability of Ai, j is given by (.25)2 = 0.0625, much lower than the probability of the original partial product (i.e. 0.25). Alternatively the probability of Oi, j is 7/16. From the above two observations it can be concluded that speculative carry tree utilizes lower probability terms, to minimize the probability of misprediction. The introduction of recoded terms does not modify the total number of partial products, but introduces an additional very small delay for the recoded partial products. The figure below shows a 16 X 16 Partial Product Matrix (PPM) after being recoded.

3.2 **Partial Product Partitioning**Only the lower probability terms Ai, j is has been added in the speculation carry-save tree. Partial products that belong to the largest columns of PPM are singly recoded. In the figure given below the partial products in the columns 11, 12……22 are recoded.

3.3 Speculative Compression Although the probability Ai, j has been decreased with respect to the actual partial products, simple removal of Ai, j terms would bring about a large misprediction error probability. Thus, instead of omitting these terms we sum them in an approximate manner by using speculative compressors. A (m: 2) speculative counter has m inputs (x0….xm-1) and only two outputs Sum (S) and Carry (C). The speculation compressor counts the number of input bits and determines the output bits, on the supposition that not than more than three inputs are high. Analogously to full adders and half adders, the output C has a doubled weight with respect to S, so that 2C + S = x0 +x1……..+.xm-1 for: x0 +x1……..+.xm-1 3, it is not possible to represent sum x0 +x1……..+.xm-1, by using only C and S signals for all likely input configurations. The speculation counter computes the outputs based on the supposition that not more than three inputs are high: If this criterion is not met, an error occurs; the multiplication result is wrong and must be corrected.

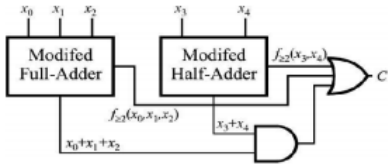

Fig. 2.Speculative Compressor

4. Synthesis and Simulation Results

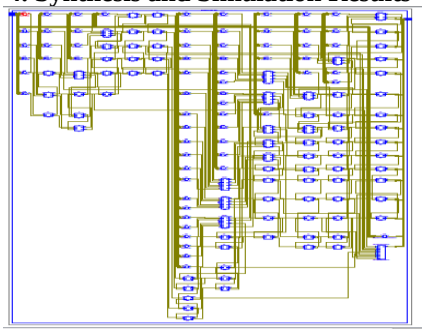

Fig. 3.RTL Schematic

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	88	960	9%
Number of 4 input LUTs	155	1920	8%
Number of bonded IOBs	32	108	29%

Fig. 4.Design Summary

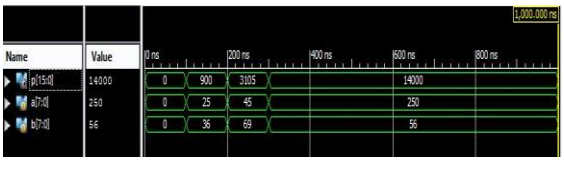

Fig. 5.Approximate Multiplier output

5.Conclusion

Here a high speed approximate multiplier design has been proposed. Proposed design utilizes an optimised TDM tree. The circuit utilizes some of the partial product as well as a speculative compression tree to sum the recoded partial products. A speculative adder is used in the final carry propagate addition. The designs functionality

have been verified using Xilinx ISE design suite 14.5 (web-edition). A comparison of the proposed design with conventional approximate multiplier showed that it has faster operation. The synthesis and simulation results showed that the proposed multiplier design gives 45.4% improvement in delay, lesser resource utilization and lesser power consumption as compared to multiplier without optimisation. In cases where multiplier speed is not critical, the use of speculative units remains unjustified. The performance of the approximate multiplier can further be improved by considering don't- care conditions and further by using variable latency adder instead of almost correct adder.

References

[1]. L. S. Wallace, ―A suggestion for fast multipliers,‖ IEEE Trans.Comput., vol. EC-13, Feb. 1964, pp. 14–17.

[2]. L. Dadda, ―Some schemes for parallel multipliers,‖ Alta Frequenza, vol. 34, Mar. 1965.

[3]. V. G. Oklobdzija, D. Villeger, and S. S. Liu, ―A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach,‖ IEEE Trans. Comput., vol. 45, no. 3, Mar. 1996, pp. 294–306.

[4]. E. J. King and E. E. Swartzlander, Jr., ―Data dependent truncated scheme for parallel multiplication,‖ in Proceedings of the Thirty First Asilomar Conference on Signals, Circuits and Systems, 1998, pp. 1178–1182.

[5]. A. Cilardo, ―A new speculative addition architecture suitable for two's complement operations,‖ in Proc. Design, Autom., Test Eur. Conf. Exhib., Apr. 2009, pp. 664–669.

[6]. H.R. Mahdiani, A. Ahmadi, S.M. Fakhraie, C. Lucas, ―Bio-Inspired Imprecise Computational Blocks for Efficient VLSI Implementation of Soft-Computing Applications,‖ IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 4, April 2010, pp. 850-862.

[7]. P. Kulkarni, P. Gupta, and MD Ercegovac, ―Trading accuracy for power in a multiplier architecture‖, Journal of Low Power Electronics, vol. 7, no. 4, 2011, pp. 490—501.

[8]. A. A. Del Barrio, S. O. Memik, M. C. Molina, J.M.Mendias, and R. Hermida, ―A distributed controller for managing speculative functional units in high level synthesis,‖ IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 30, no. 3, Mar. 2011, pp. 350– 363.

ABOUT AUTHORS

S NARESH, studying M TECH final year of ECE branch, VLSI& EMBEDDED SYSTEMS stream in Siddhartha Institute of Engineering And Technology, Hyderabad, TS, India.

MALATHI NADDUNOORO working as assistant professor in ECE branch, Siddhartha Institute of Engineering And Technology, Hyderabad, TS, India.

