

Design And Implementation of an Multiplier Using Hybrid Carry Technique

L.Soujanya¹, P.Sirisha²

1M.Tech Student, Sahasra College of Engineering for Women, Warangal

2Assistant Professor, Sahasra College of Engineering for Women, Warangal

ABSTRACT: In a typical processor, Multiplication is one of the basic arithmetic operations and it requires substantially more hardware resources and processing time. In computers S-MB, a typical central processing unit devotes a considerable amount of processing time in implementing arithmetic operations, particularly multiplication operations. In this paper, multiplier is done for low power requirement and high speed with Hybrid Carry Technique to improve the speed, area parameters of multiplier.

INTRODUCTION

Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore low-power multiplier design has been an important part in low-power VLSI system design. There has been extensive work on low-power multipliers at technology, physical, circuit and logic levels. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system.

We evaluated the performance of the proposed S-MB technique by comparing its three different schemes with the state-of-the-art recoding techniques. Industrial tools for RTL synthesis and

power estimation have been used to provide accurate measurements of area utilization, critical path delay and power dissipation regarding various bit-widths of the input numbers. We show that the adoption of the proposed recoding technique delivers optimized solutions for the FAM design enabling the targeted operator to be timing functional (no timing violations) for a larger range of frequencies. Also, under the same timing constraints, the proposed designs deliver improvements in both area occupation and power consumption, thus outperforming the existing recoding solutions.

Existing system

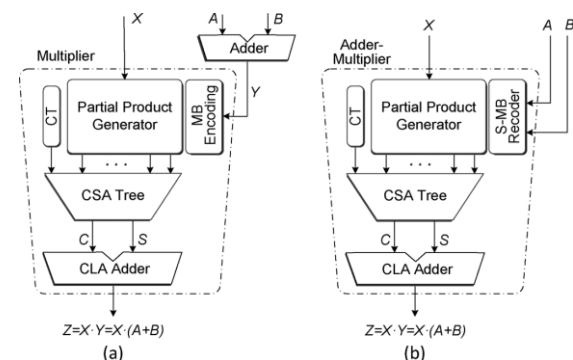


Fig. 1 Existing System

The conventional design of the AM operator requires that its inputs and are first driven to an adder and then the input and the sum are driven to a multiplier in order to get output. The drawback of using an adder is that it inserts a

significant delay in the critical path of the AM. As there are carry signals to be propagated inside the adder, the critical path depends on the bit-width of the inputs. In order to decrease this delay, a Carry-Look-Ahead (CLA) adder can be used which, however, increases the area occupation and power dissipation. An optimized design of the AM operator is based on the fusion of the adder and the MB encoding unit into a single datapath block.

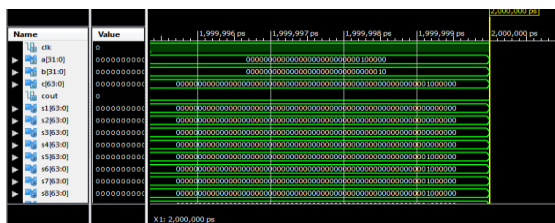


Fig. 2 Existing System Waveform

Above figure 2 shows the simulation results of multiplier using Carry look ahead. The timing diagrams for two 32-bit inputs and one 64-bit output. a[31:0],b[31:0] are the input patterns of 32-bit and c[63:0] is the final product value.

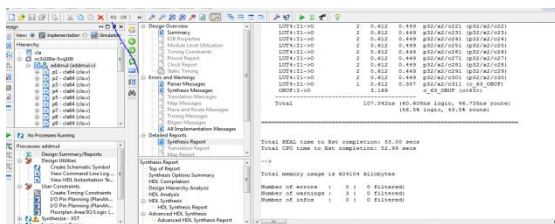


Fig.3 Synthesis report of existing system

The above figure 3 shows synthesis report after the process of synthesis is completed.

Time delay: 107.342ns.

Memory used: 604104 kilobytes.

Proposed system

If we want to multiply two binary number (multiplicand A has n bits and multiplier B has m bits) using single n bit adder, we can built a Hybrid Carry circuit that processes a single partial product at a time and then cycle the circuit m times. This type of circuit is called Hybrid Carry multiplier. Sequential multipliers are attractive for their low area requirement.

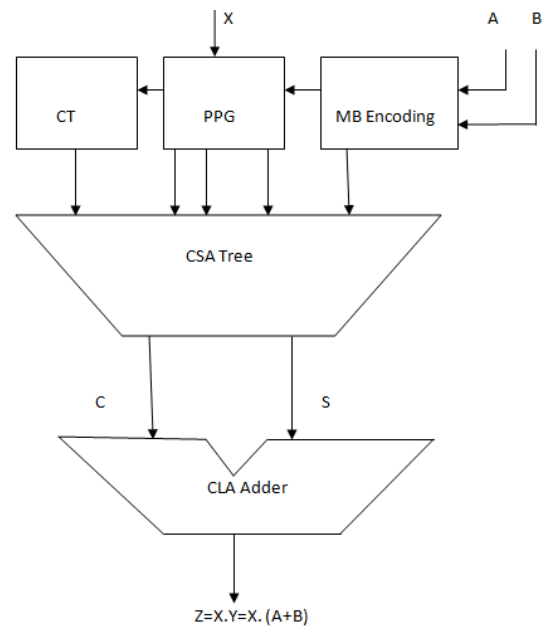


Fig. 4 Proposed System

In a Hybrid Carry multiplier, the multiplication process is divided into some sequential steps. In each step some partial products will be generated, added to an accumulated partial sum and partial sum will be shifted to align the accumulated sum with partial product of next steps. Holding outputs in accumulator register can reduce additional instruction. An accumulator should be fast in response so it can be implemented with one of fastest adder like carry-look-ahead adder.

The pipelining is a popular technique to increase throughput of a high speed system which divides total system into several small cascade stages and add some registers to synchronize output of each stage. As the no. of stages increases, the power consumption and area gets increased. So, most of the times pipelining technique can be introduced in CSA tree in order to improve the performance. Also, when arithmetic throughput is more important than latency, pipelined multipliers are useful because the introduction of registers along the array reduces the unnecessary activity.

Pipelining is a concept to reduce the delay in the critical path. It is done by adding registers or latches in the data path. By eliminating the delay in the critical path the speed and throughput is increased. Pipelining block is constructed using registers. Registers consists of latches (flip-flops). Pipelining is a popular technique to increase throughput of a high speed system, which divides total system into several small cascade stages and adds some register to synchronize outputs of each stages. Also parallel pipeline architecture is considered to be most suitable for low voltage and low power system. In a pipelining system, the maximum operating frequency is limited by the slowest stage which has the longest delay time.

The output waveform result is shown in below figure 5.

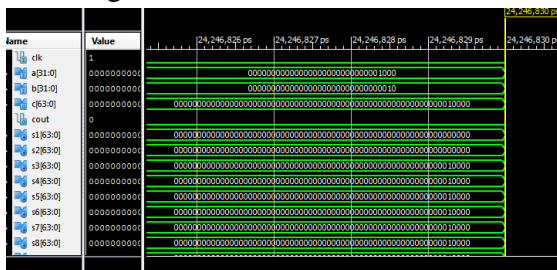


Fig. 5 Proposed system waveform results of multiplier using Hybrid carry technique. The timing diagrams for two 32-bit inputs and one 64-bit output. a[31:0], b[31:0] are the two input patterns of 32-bit and c[63:0] is the final product value. Clock is the clock signal supplied to the circuit and remaining timing waveforms represents the partial products of multiplier.

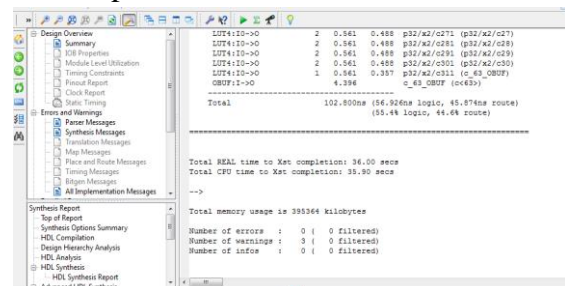


Fig.6 Synthesis report of proposed system

The above figure 6 shows synthesis report after the process of synthesis is completed.

Time delay: 102.800ns.

Memory used: 395364 kilobytes

Table1: Comparison of Existing and Proposed Systems Results

parameters	Existing	Proposed
Time delay	107.342ns.	102.800ns.
Memory Used	604104 Kilobytes	395364 kilobytes

CONCLUSION

This paper, presented a Multiplier is the very important hardware block in digital systems. By increasing the performance of the multiplier, the performance of entire

circuit will increase. In existing system Carry-look-ahead adder is used to reduce the delay and area. The proposed system Hybrid carry technique yields considerable reduction in the delay and area compared to the existing system. To overcome the problem of power and area, the CSA design makes use of efficient full adders. Among the tested three efficient full adders N-10T is found to be the most efficient full adder.

REFERENCES

- [1] Amaricai, M. Vladutiu, and O. Boncalo, "Design issues and implementation for floating-point divide-add fused," *IEEE Trans. Circuits Syst. II-Exp. Briefs*, vol. 57, no. 4, pp. 295–299, Apr. 2010.
- [2] E. E. Swartzlander and H. H. M. Saleh, "FFT implementation with fused floating-point operations," *IEEE Trans. Comput.*, vol. 61, no. 2, pp. 284–288, Feb. 2012.
- [3] J.J.F. Cavanagh, *Digital Computer Arithmetic*. New York: McGraw-Hill, 1984.
- [4] S. Nikolaidis, E. Karaolis, and E. D. Kyriakis-Bitaros, "Estimation of signal transition activity in FIR filters implemented by a MAC architecture," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 19, no. 1, pp. 164–169, Jan. 2000.
- [5] O. Kwon, K. Nowka, and E. E. Swartzlander, "A 16-bit by 16-bit MAC design using fast 5:3 compressor cells," *J. VLSI Signal Process. Syst.*, vol. 31, no. 2, pp. 77–89, Jun. 2002.
- [6] L.-H. Chen, O. T.-C. Chen, T.-Y. Wang, and Y.-C. Ma, "A multiplication-Accumulation computation unit with optimized compressors and minimized switching activities," in *Proc. IEEE Int.*

Symp. Circuits and Syst., Kobe, Japan, 2005, vol. 6, pp. 6118–6121.

[7] Y.-H. Seo and D.-W. Kim, "A new VLSI architecture of parallel multiplier-accumulator based on Radix-2 modified Booth algorithm," *IEEE Trans. Very Large Scale Integer. (VLSI) Syst.*, vol. 18, no. 2, pp. 201–208, Feb. 2010.

[8] A. Peymandoust and G. de Micheli, "Using symbolic algebra in algorithmic level DSP synthesis," in *Proc. Design Automation Conf.*, Las Vegas, NV, 2001, pp. 277–282.

[9] W.C. Yeh and C.-W. Jen, "High-speed and low-power split-radix FFT," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 864–874, Mar. 2003.

[10] C.N. Lyu and D. W. Matula, "Redundant binary Booth recoding," in *Proc. 12th Symp. Comput. Arithmetic*, 1995, pp. 50–57.

Authors:



L. Soujanya pursuing M.Tech in VLSI from Sahasra College of Engineering for Women, Warangal.

P. Sirisha working as Assistant professor, Department of ECE in Sahasra College of Engineering for Women, Warangal.