International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 07
May 2017

An Efficient Flexible Dsp Architecture For Error Tolerant Applications Employing Carry Save Arithmetic

PULIPATI SANTHI KUMARI [1], M.Tech SCHOLAR, CHIRALA ENGINEERING COLLEGE, CHIRALA
S.PRABHU DAS [2], ASSOCIATE PROFESSOR, CHIRALA ENGINEERING COLLEGE, CHIRALA

ABSTRACT: In this paper, we present a low power 32-bit multiplier design, by using Carry Save Adder (CSA). The multiplier design shown in this paper is modeled using Verilog language for 32-bit unsigned data. Optimizing speed and power constraints have become challenging tasks in the design of any reliable and efficient integrated circuit. This paper sets up the design of a 32-bit multiplier which is of high speed, and has low power consumption. By reducing the generated partial products the speed of the multiplier can be increased. There are many ways by which we can reduce the number of partial products generated in a multiplication process. Wallace tree multiplier method is one of them.Therefore, minimizing the number of half adders used in a multiplier will reduce the circuit complexity.

I. INTRODUCTION

The major considerations while designing the digital circuits are speed, power and area. Multiplication is a mathematical operation that at its simplest is an abbreviated process of adding an integer a specified number of times. A basic multiplier can be divided in to three parts i) partial product generation ii) partial product addition and iii) final addition. Multiplication plays an important role in Digital Signal Processing (DSP)applications, such as filtering and fast Fourier transform(FFT).

Parallel array multipliers are widely used to achieve high speed execution. But these multipliers consume more power. In today's VLSI system design, Power consumption has become a critical concern. For the design of low-power DSP systems the designers need to concentrate on power efficient multipliers.

The impulse response of the filter can be either finite or infinite. The methods for designing and implementing of these twofilter classes differ considerably. Finite impulse response(FIR) filters are digital filters whose response to a unit impulse (unit sample function) is finite in duration. This is in contrast to infinite impulse response (IIR) filters whose response to a unit impulse (unit sample function) is infinite in duration. FIR and IIR filters each have advantages and disadvantages. In some applications, the FIR filter circuit must be able to operate at high sample rates, while in other applications the FIR filter circuit must be a low power.

Circuit operating at moderate sample rates. However, these algorithms are based on carry-propagation adders with two inputs and one output, but for many high speed applications carry-save arithmetic is preferred [8]. By using carry-save adders the need for carry propagation in the adder is avoided and the latency of one addition is

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 07
May 2017

equal to the gate delay of a full adder. The carry-save adder has three inputs and two outputs, where the two outputs together form the result.

The length of the carry output increases with one bit after each addition, but by using the carry overflow detection proposed in [8] the length can be kept constant. If the input to the multiplier is in carry-save format the previously proposed multipliers can be used by replacing each adder with two carry-save adders. However, if the input is only one binary word the mapping to carry-save adders will be suboptimal.

II. ARRAY MULTIPLICATION

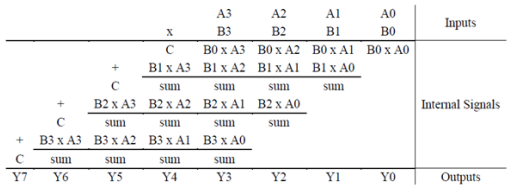


Figure.1.Array Multiplication

Array multiplier is well known due to its regular structure. Multiplier circuit is based on add and shift algorithm. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial product are shifted according to their bit orders and then added. The addition can be performed with normal carry propagate adder. In array multiplication we need to add, as many partial products as there are multiplier bits.

This proposed system incorporates the CS-to-MB recoding unit. We assume 16-bit input operands for all the designs and, without loss of generality; we do not consider any truncation concept during the multiplications. A **carry-save adder** is a type of digital adder, used in computer micro architecture, to compute the sum of three or more *n*-bit numbers in binary a type of digital is used that is carry save adder. It differs from other digital adders in that it outputs two numbers of the same dimensions as the inputs, one which is a sequence of partial sum bits and another which is a sequence of carry bits. Using basic arithmetic. Although we know the last digit of the result at once, we cannot know the first digit until we have gone through every digit in the calculation, passing the carry from each digit to the one on its left. To take a time proportional to *n* add two n digit numbers, even if the machinery we are using would otherwise be capable of performing many calculations simultaneously.

In electronic terms, using bits (binary digits), this means that even if we have *n* one-bit adders at our disposal, to allow a possible carry to propagate from one end of the number to the other a time proportional to n is required. Until we have done this,

1. We do not know the result of the addition.
2. We do not know whether the result of the addition is larger or smaller than a given number (for instance, we do not know whether it is positive or negative).

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 07
May 2017

A carry look-ahead adder can reduce the delay. In principle the delay can be reduced so that it is proportional to log*n*, but for large numbers this is no longer the case, because even when carry look-ahead is implemented, the distances that signals have to travel on the chip increase in proportion to *n*, and propagation delays increase at the same rate.

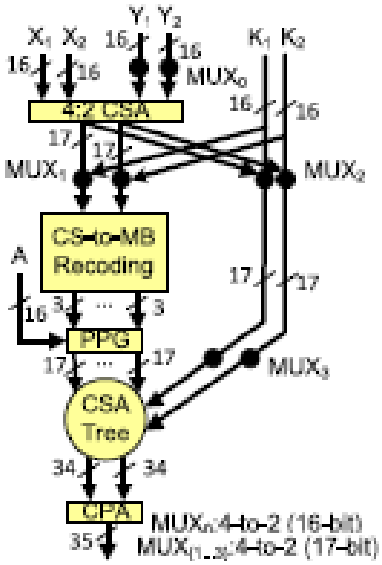


Figure.2.Incorporating the CS-to-MB recoding concept

A carry look-ahead is not of much use in public key cryptography, once we get the 512bit to 2048 bit number sizes that is required.

III.ARCHITECTURE OF MULTIPLIER WITHCARRY SAVE ADDER

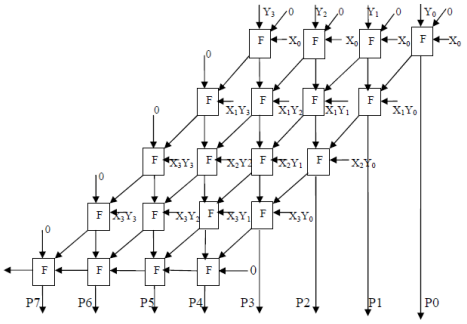


Figure.3.Multiplier graph with Carry saves Adder Architecture

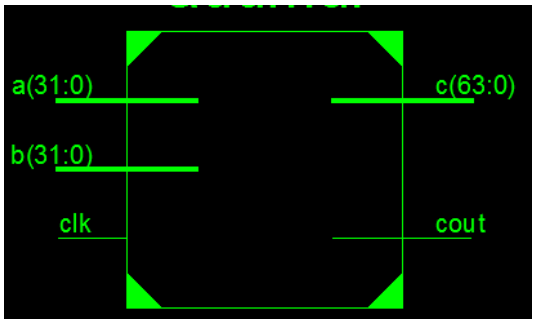
In the Carry Save Addition method, the first row will be either Half-Adders or Full-Adders. If the first row of the partial products is implemented with Full-Adders, Cin will be considered „0". Then the carries of each Full- Adder can be diagonally forwarded to the next row of the adder. The resulting multiplier is said to be Carry Save Multiplier, because the carry bits are not immediately added, but rather are saved for the next stage. In the design if the full adders have two input data the third input is considered as zero. In the final stage, carries and sums are merged in a fast carry-propagate (e.g. ripple carry or carry look ahead) adder stage.

Here we utilize this technique to represent the different topologies for the possible multipliers. The graphs are directed acyclic graphs (DAGs), i.e., there is no feedback and the edges have a direction. Each edge in the graph represents a multiplication with, i.e., a binary shift. The sign can be arbitrarily selected since subtraction is as expensive as addition. The edge can either be a single binary word or in carry-save representation. Each vertex in the graph,

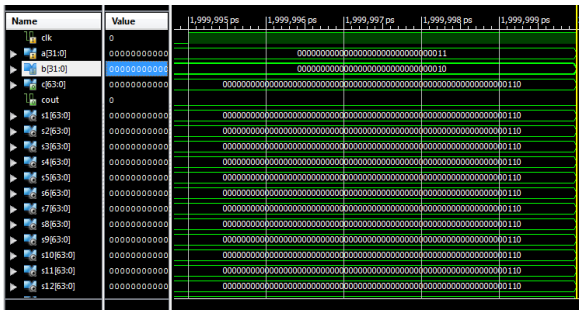
except for the initial, represents an addition. Each vertex can have two or three inputs and an arbitrary number of outputs. If the vertex has three inputs all have to be single binary words. The corresponding hardware is then one carry-save adder. One carry-save adder is also obtained for one single word and one carry-save input. If both inputs are in carry-save format, two carry-save adders are required in the corresponding implementation.

IV.RESULTS

RTL SCHEMATIC



OUTPUT



V.CONCLUSION

In this paper we present a 32-bit multiplier design which is highly efficient and low power consuming. The multiplication operation using carry save adder is faster and efficient than any other adder. From the results, we see how CSA logic is implemented in each partial product line which improves the overall performance of the multiplier unit. This multiplier is very much helpful in larger multiplications as the delay has been reduced. This design is implemented on 32-bit unsigned data. This design can also be extended over signed multiplication as well as it can be used for designing of higher bit multiplier.

VI.REFERENCES

[1] P. Ienne and R. Leupers, *Customizable Embedded Processors:Design Technologies and Applications*. San Francisco, CA, USA: Morgan Kaufmann, 2007.

[2] P. M. Heysters, G. J. M. Smit, and E. Molenkamp, "A flexible and energy-efficient coarse-grained reconfigurable architecture for mobilesystems," *J. Supercomput.*, vol. 26, no. 3, pp. 283–308, 2003.

[3] B. Mei, S. Vernalde, D. Verkest, H. D. Man, and R. Lauwereins,"ADRES: An architecture with tightly coupled VLIW processor andcoarse-grained reconfigurable matrix," in *Proc. 13th Int. Conf. FieldProgram. Logic Appl.*, vol. 2778. 2003, pp. 61–70.

[4] M. D. Galanis, G. Theodoridis, S. Tragoudas, and C. E. Goutis,"A high-performance data path for synthesizing DSP kernels," *IEEETrans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 6, pp. 1154–1162, Jun. 2006.

[5] K. Compton and S. Hauck, "Automatic design of reconfigurable domainspecificflexible cores," *IEEE Trans. Very Large Scale Integr. (VLSI)Syst.*, vol. 16, no. 5, pp. 493–503, May 2008.
[6] S. Xydis, G. Economakos, and K. Pekmestzi, "Designing coarse-grainreconfigurable architectures by inlining flexibility into custom arithmeticdata-paths," *Integr., VLSI J.*, vol. 42, no. 4, pp. 486–503, Sep. 2009.
[7] S. Xydis, G. Economakos, D. Soudris, and K. Pekmestzi, "High performanceand area efficient flexible DSP datapath synthesis," *IEEE Trans.Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 3, pp. 429–442,Mar. 2011.
[8] G. Ansaloni, P. Bonzini, and L. Pozzi, "EGRA: A coarse grainedreconfigurable architectural template," *IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst.*, vol. 19, no. 6, pp. 1062–1074, Jun. 2011.
[9] M. Stojilovic, D. Novo, L. Saranovac, P. Brisk, and P. Ienne, "Selectiveflexibility: Creating domain-specific reconfigurable arrays," *IEEE Trans.Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 5, pp. 681–694,May 2013.

PULIPATI SANTHI KUMARI received my B.Tech degree from Rajamahendri Institute of Engg&Technology,Rajamundry, A.P, India in 2013 and Pursuing M.Tech in Chirala Engineering College,Chirala, India

Mr. S.PRABHU DAS is working as Associate Professor in ECE Department in Chirala Engineering College, Chirala. He has 11 years of Teaching experience.