IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 13, Issue 4, Ver. I (Jul.-Aug. 2018), PP 31-33
www.iosrjournals.org

High Speed Arithmetic: Interleaved Modular Multipliers Using Radix 8 Booth Encoder

Deepthi R Shetty[1], Ashwath Rao[2], Praveen Kumar M[3]
[1](M.Tech in VLSI Design and Embedded Systems, Sahyadri College of Engineering and Management, Adyar, Mangaluru, India)
[2] (Head of the Department, Dept of Electronics and Communication Engg, Sahyadri College of Engineering and Management, Adyar, Mangaluru, India)
[3] (Assistant Professor, Dept of Electronics and Communication Engg, Sahyadri College of Engineering and Management, Adyar, Mangaluru, India)
Corresponding Author: Deepthi R Shetty

Abstract: *Booth multiplication algorithm leads to faster performance compared to lot of other algorithms. Various encoding styles are available depending upon number of bits in the group such as radix-2, radix-4, radix-8, radix-16, etc. In this paper radix-8 Booth encoded modular multipliers based on interleaved multiplication algorithm are discussed. We can reduce the number of partial products by using radix-8 in the multiplier encoding, thereby obtaining a simpler CSA tree .This results in less delay and a smaller area size. This multiplication operation can be performed for both signed and unsigned numbers as a result the cost of the system can also be reduced. The carry save adder(CSA) tree implementation can speed up the operation of multiplier. Kogge Stone adders are implemented to improve the efficiency. Kogge Stone adders is a parallel prefix form carry look ahead adder and it can be determined by replacing carry save adder(CSA) tree and the final two operand parallel prefix adder with a parallel prefix adders of Kogge Stone algorithm. Booth encoding is an encoding process used to minimize the number of partial products in a multiplication process.*

Keywords*: Booth algorithm, Radix-8, carry save adder, Koggestone adder, Interleaved modular multipliers.*

Date of Submission: 09-07-2018 Date of acceptance: 23-07-2018

I. Introduction

Multiplier is an important part in digital signal processing (DSP) systems. The operation in any multiplication algorithm is reduced to a partial product summation. Every partial product denotes a multiple of the multiplicand which must be added to the final result [1]. In radix-2 algorithm, a series of products is formed in between the multiplicand, and each and every bit of the multiplier resulting in partial products [2]. All these partial products are added to get the final result. Redundant arithmetic is performed to get the additions as fast as possible. The speed can be increased by a Wallace reduction tree. The Wallace tree structure is a version of the carry-save adders (CSA). In the conventional CSA tree, partial product bits with many inputs residing at the same bit position are reduced to a final sum and carry pair with the help of a series of full adders which are single bit each. At the output, sum and carry which has to be added by a carry-propagate adder (CPA) will be left. Whereas, a radix-8 recoding provides gain in time while adding the partial products since for n bits of multiplier and multiplicand, partial products are reduced to n/3 when compared to n/2 in radix-4[2]. Radix-8 recoding use less number of transistors resulting in a reduced area size and power dissipation compared to radix-4[3]. The number of partial products can also be reduced by using a higher radix-8 booth multiplication technique in the multiplier encoding. A lesser delay can be obtained by replacing the CSA tree with Kogge Stone adder, which is a parallel prefix form of carry look-ahead adder. During binary addition, the propagation delay in the carry chain is a major concern. Length of carry chain increases when the input bit or operand increases. To avoid this carry chain propagation, parallel prefix adder architecture is used. Prefix adder architecture includes pre-processing, carry generate and post- processing stages.

II. Interleaved Modular Multiplication

In interleaved modular multiplication, the multiplication and the calculation of the remainder of the division are interleaved. The interleaved modular multiplication is performed by multiplying the first operand with the second operand bitwise and then added to the intermediate result. Two subtractions per iteration are required in order to reduce the intermediate result with respect to the modulus. The advantage of interleaved modular multiplication is that the length of the intermediate result is only one or two bits larger than the operands and the disadvantage is the use of subtractions for reducing the intermediate results.

III. Design Of Kogge Stone Adder

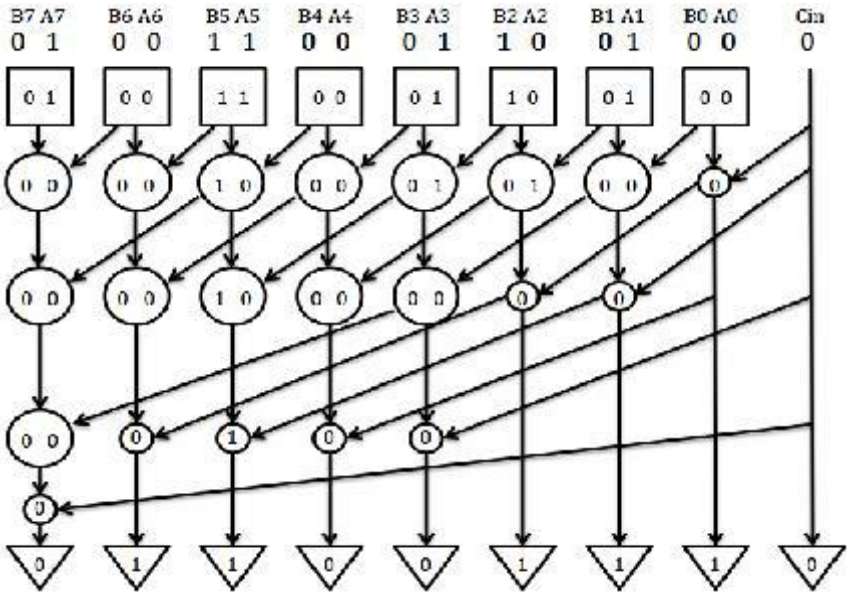

Figure 1: 8 bit Kogge Stone adder

The Kogge Stone adder is a parallel prefix form of carry look-ahead adder. It is considered as the fastest adder design and is widely used in the industry. Carries are generated fast by computing them in parallel, speed of operation is very high due to the low depth of node and operation is performed in parallel. The outcome of the adder is dependent upon the initial inputs. The construction of a Kogge Stone parallel prefix adder is a three step process which includes a pre-processing block, carry generation block and a post-processing block.

i. Pre-processing block: Pre-processing stage is the intial stage of the parallel prefix adder, which is used to generate the propagated signal and generated signal and for a given inputs this signal are computed by using following equations.
Pi= Ai XOR Bi (1)
Gi= Ai AND Bi (2)

ii. Carry Generation Block: Carry generation stage is an important block in this adder design. It consists of two components namely Block Cell and Gray Cell. Block Cell is used to produce the generated signal and propagated signal, needed for the calculation of the next stage. Gray Cell is used to produce only the generated signal and this signal is utilized or needed in the calculation of the sum in the next stage.
Black Cell: The black cell operator receives two set of generate and propagate signals (Gi, Pi) and (Gj, Pj) compute one set of generate and propagate signals (G, P).
Gray Cell: The Gray operator receives two set of generate and propagate signals (Gi, Pi) and (Gj, Pj) compute one set of generate signals (G).

iii. Post Processing Block: This is the final computation stage of the adder. Sum and carry are the final outcomes of the adder.

IV. Results

Figure 2 shows the RTL view of a radix-8 Booth encoded modular multiplier. Figure 3 shows the delay report for the multiplier. Delay has been further reduced by replacing CSA with Kogge Stone parallel prefix adder in the summation stage. Because of this the overall multiplication time required has been reduced with radix-8 architecture.

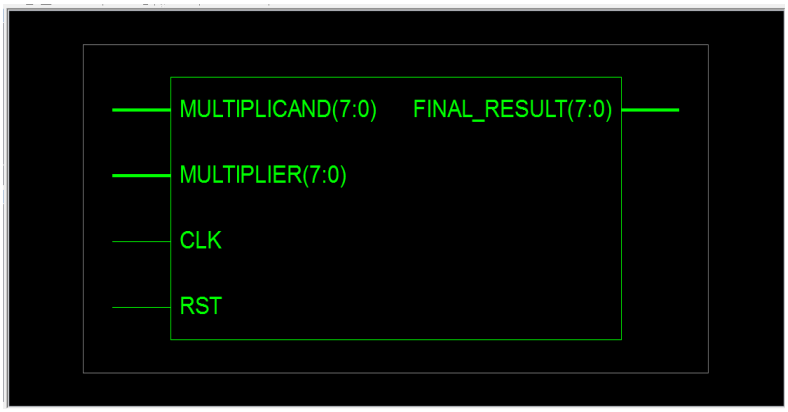

Figure 2: RTL Schematic

```
--------------------------------

Generating "PAR" statistics.

**************************
Generating Clock Report
**************************

+---------------------+--------------+------+------+-------------+-------------+
|        Clock Net    |   Resource   |Locked|Fanout|Net Skew(ns)|Max Delay(ns)|
+---------------------+--------------+------+------+-------------+-------------+
|           CLK_BUFGP |      BUFGMUX2| No   |    8 |  0.000      |  0.936      |
+---------------------+--------------+------+------+-------------+-------------+

* Net Skew is the difference between the minimum and maximum routing
only delays for the net. Note this is different from Clock Skew which
is reported in TRCE timing report. Clock Skew is the difference between
the minimum and maximum path delays which includes logic delays.

   The Delay Summary Report

The NUMBER OF SIGNALS NOT COMPLETELY ROUTED for this design is: 0

   The AVERAGE CONNECTION DELAY for this design is:         1.214
   The MAXIMUM PIN DELAY IS:                                3.695
   The AVERAGE CONNECTION DELAY on the 10 WORST NETS is:    2.952

   Listing Pin Delays by value: (nsec)
```

Figure 3 : Delay Report

A radix -8 booth encoded modular multiplier is implemented using Xilinx and cadence. A total power (mW) of 56.01 is obtained using Xilinx .Table 1 shows the obtained power values when implemented using cadence

.

Leakage Power(nW)	10232.014
Dynamic Power(nW)	127839.464
Total Power(nW)	138071.478

Table 1: Power values obtained using cadence

References

[1]. J.A. Hidalgo, V. Moreno-Vergara, O. Oballe, A. Daza, M.J. Martín-Vázquez, A.Gago ,"A Radix-8 multiplier design for specific purpose"@2011.
[2]. lakshmanan, m. othman, m.a.m. ali, "design and characterization of parallel prefix adders using fpgas," journal of computers, vol. 5, no. 10, october 2012.
[3]. L.P. Rubinfield, "A Proof of the Modified Booth's Algorithm for Multiplication," IEEE Transaction on computers, vol.39.
[4]. P. Montgomery, "Modular Multiplication without Trial Division," Mathematics of Computation, vol. 44, pp. 519–521, 1985.
[5]. G. Blakley, "A Computer Algorithm for Calculating the Product A_B moduloM," IEEE Transactions on Computers, vol. C-32, no. 5, pp. 497–500, May 1983.
[6]. C.S. Wallace, "A suggestion for fast multipliers," IEEE Trans. Electron. Comput., Feb. 1964.
[7]. A.D. Booth, "A signed binary multiplication technique," Quarterly J. Mechan. Appl. Math., vol IV. Part 2, 1951.

Deepthi R Shetty " High Speed Arithmetic: Interleaved Modular Multipliers Using Radix 8 Booth Encoder ." IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) 13.4 (2018): 31-33.