International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Digital Multipliers: A Review

Jyoti Sharma[1], Sachin Kumar[2]

[1]Dept. of Electronics and Communication Engineering, Meri College of Engineering & Technology, Sampla, Bhadurgarh, Haryana, India

[2]Assistant Professor, Department of Electronics and Communication Engineering, Meri College of Engineering & Technology, Sampla, Bhadurgarh, Haryana, India

Abstract: ***Multiplication is one of the basic functions used in digital signal processing (DSP). Hardware resources and processing time required by it are more than addition and subtraction. There are two kinds of multiplication algorithms, serial multiplication algorithms and parallel. Multiplication algorithms. Serial multiplication algorithms use sequential circuits with feedbacks. Parallel multiplication algorithms often use combinational circuits, and do not contain feedback structures. This paper presents various multiplier architectures. Multiplier architectures fall generally into two categories i.e., "tree" multipliers and "array" multipliers. Tree multipliers add as many partial products in parallel as possible and therefore, are very high performance architectures. Multiplication operation involves generation of partial products and their accumulation. The speed of multiplication can be increased by reducing the number of partial products.***

Keywords: Architecture, Digital system, Hardware, Logic functions, Propagation delay, Sequentially.

1. Introduction

Different computer arithmetic techniques can be used to implement a digital multiplier. Out of these most techniques involve computing a set of partial products, and then summing the partial products together. Until the 1970s, most minicomputers did not have a multiply instruction however, Mainframe computers had multiply instructions, but they did the some sorts of shifts and adds as a "multiply routine". Early microprocessors also had no multiply instruction. Then the Motorola 6809, introduced in 1978, was one of the earliest microprocessors with a dedicated hardware multiply instruction. It did the same sorts of shifts and adds as a "multiply routine", but the difference is that implementation was done in the microcode of the MUL instruction.

As more transistors per chip became available due to larger-scale integration, it became possible to put adequate adders on a single chip to sum all the partial products at once, rather than reuse a single adder to handle each partial product one at a time.

Now, because some common digital signal processing algorithms spend most of their time multiplying, digital signal processor designers sacrifice a lot of chip area in order to make the multiply as fast as possible; a single-cycle multiply–accumulate unit often used up most of the chip area of early DSPs. The construction of most digital systems is a large task. Disciplined designers in any field will subdivide the original task into manageable subunits building blocks and will use the standard subunits wherever possible. In digital hardware, the building blocks have such names as adders, registers, and multiplexers.

2. Multipliers

A binary multiplier is an electronic circuit used in digital electronics, such as a computer, to multiply two binary numbers. It is built using binary adders. There are different types of multipliers. Some of these are

- Array Multipliers
- Tree Multipliers

Array Multipliers
These can be classified in to following categories:
a) Braun Multiplier
b) Booth Multiplier
c) Modified Booth Multiplier
d) Baugh -Wooley Multiplier

a) Braun Multiplier
Braun Array multiplier is well known due to its regular structure. It is a simple parallel multiplier that is commonly known as carry save array multiplier. This multiplier is restricted to performing multiplication of two unsigned numbers. It consist of array of AND gates and adders arranged in iterative structure that does not require logic registers. This is also known as the non-additive multiplier since it does not add an additional operand to result of multiplication [3].
Advantages:
1) Regular structure
2) Easy to layout
3) Small size
4) The design time of an array multiplier is much less
5) Ease of design for a pipelined architecture.
Disadvantages:
1) As operand sizes increase, arrays grow in size at a rate equal to the square of the operand size.
2) The hardware is underutilized [2].

b) Booth Multiplier
Conventional array multipliers, like the Braun multiplier and Baugh Woolley multiplier achieve comparatively good performance but they require large area of silicon, unlike the add-shift algorithms, which require less hardware and exhibit poorer performance. The Booth multiplier makes use of Booth encoding algorithm in order to reduce the number of partial products by considering two bits of the multiplier at a time, thereby achieving a speed advantage over other multiplier architectures. This algorithm is valid

for both signed and unsigned numbers. It accepts the number in 2'scomplement form, based on radix-2 computation [3].

c) Modified Booth Multipliers-

The modified Booth encoding (MBE), or modified Booth's algorithm (MBA), was proposed by O. L. Macsorley in 1961 [5]. The recoding method is widely used to generate the partial products for implementation of large parallel multipliers, which adopts the parallel encoding scheme. The original version of Booth algorithm (Radix-2) had two drawbacks:

- The number of add subtract operations and the number of shift operations becomes variable and becomes inconvenient in designing parallel multipliers.
- The algorithm becomes inefficient when there are isolated 1"s.

These problems can be overcome by modified Booth algorithm. MBA process three bits at a time during recoding. Recoding the multiplier in higher radix is a powerful way to speed up standard Booth multiplication algorithm. In each cycle a greater number of bits can be inspected and eliminated therefore, total number of cycles required to obtain products get reduced. Number of bits inspected in radix r is given by $n = 1 + \log_2 r$.

3. Tree Multipliers

The tree multiplication algorithm can reduce the number of partial products by employing multiple input compressors capable of accumulating several partial products concurrently [2]. Tree multiplier can handle the multiplication process for large operands. This is achieved by minimizing the number of partial product bits in a fast and efficient way by means of a CSA tree constructed from 1-bit full adders. These are of following types:

Wallace Tree Multiplier- A fast process for multiplication of two numbers was developed by Wallace. In 1964, C.S. Wallace [6] observed that it is possible to find a structure, which performs the addition operations in parallel, resulting in less delay. Wallace introduced a different way of parallel addition of the partial product bits using a tree of carry save adders, which is known as "Wallace Tree" [4]. A Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integers In order to perform the multiplication of two numbers with the Wallace method, partial product matrix is reduced to a two-row matrix by using a carry save adder and the remaining two rows are summed using a fast carry- propagate adder to form the product Wallace method uses three-steps to process the multiplication operation.

- Formation of bit products.
- The bit product matrix is reduced to a 2-row matrix by using a carry-save adder.
- The remaining two rows are summed using a fast carry-propagate adder to produce the product.

Carry Save Adder- CSA performs the addition of m numbers in lesser duration compared to the simple addition. A carry save adder consist of full adders like ripple adders, but the carry output from each bit is taken out to form second result vector rather than being wired to the next most significant bit [7]. It takes three numbers (a, b, c) to add together and output two numbers, sum (s) and carry (c). It is carried out in one time unit duration. In carry-save adder, the carry (c) is bought until the last step and the ordinary addition is carried out in the very last step. The most important application of a carry-save adder is to calculate the partial products in integer multiplication. This allows for architectures, where a tree of carry- save adders is used to calculate the partial products very fast. One 'normal' adder is then used to add the last set of carry bits to the last partial products to give the final multiplication result. By using carry-save adder the need of carry propagation in the adder is avoided and latency of one addition is equal to gate delay of adder.

Advantages

- Propagation delay is reduced

Disadvantages

- Regularity is absent
- Routing becomes complicated

4. Multiplier Architecture

The multiplier architectures can be generally classified into following categories:

- Serial multiplier
- Parallel multiplier
- Serial-parallel

Serial Multipliar

The simplest method to perform multiplication is to add series of partial products. The serial multipliers use a successive addition algorithm. They are simple in structure because both the operands are entered in a serial manner. Therefore, the physical circuit requires less hardware and a minimum amount of chip area. However, the speed Performance of the serial multiplier is due to the operands entered sequentially.

Parallel Multiplier

Most advanced digital systems incorporate a parallel multiplication unit to carry out high speed mathematical operations. A microprocessor requires multipliers in its arithmetic logic unit and a digital signal processing system requires multipliers to implement algorithms such as convolution and filtering. Parallel multipliers [1] present high-speed performance, but are expensive in terms of silicon area and power consumption because in parallel multipliers both the operands are input to the multiplier in parallel manner. Therefore, the circuitry occupies a much larger area and is more complex as compared to serial multipliers.

Comparison

Generally, it is not possible to say that an exact multiplier yields to greater cost- effectiveness, since trade-off is design and technology dependent. Comparison between different multipliers is done by using various parameters .

These are shown in table 1.

Table:1 Comparison between multipliers[8]

Multiplier Type	*Speed*	*Circuit Complexity*	*Layout*	*Area*
Array	Low/ Medium	Simple	Regular	Smallest
Booth	High	Complex	Irregular	Medium
Wallace	Higher	Medium	More irregular	Large
Booth Wallace	Highest	More complex	Medium regularity	Largest

5. Conclusion

From this review of different multipliers it is concluded that These basic array multipliers like Braun multiplier consume low power and exhibit good performance, however there use is limited to sixteen bits. For operands of sixteen bits and higher, the modified Booth algorithm reduces the partial product number by half. Thus speed of multiplier is increased. Due to circuitry overhead in Booth multiplier its power dissipation is comparable to Braun multiplier. Wallace's strategy for carry save adder trees is to combine the partial product bits as early as possible. This method yields to simpler CSA tree and a wider carry propagate adder and the designs using the Wallace tree method are fast. However a logarithmic depth reduction tree based CSA¨s has an irregular structure that makes the design and layout difficult.

Reference

[1] Purushottam D. Chidgupkar, and Mangesh T. Karad, "The implementation of algorithms in digital signal processing", Global J. of Engineering Education, vol.8, no.2 © 2004 UICEE Published in Australia.
[2] Rabey, Nikolic, and Chandrasekhran, "Digital Integrated Circuits: A Design Perspective", 2nd Edition, Prentice Hall, pp. 586-594, 2003.
[3] Roy, Kaushik, Yeo, and Kiat-Seng, "Low voltage Low-power VLSI Subsystems", McGraw-Hill, pp.124-141.
[4] Weste, Neil H.E. Eshraghian, and Kamran, "CMOS VLSI Design: A Circuits and Systems Perspective", 3rd Edition, Pearson Education, pp. 345-356, 2005.
[5] O. L. MacSorley, "High speed arithmetic in binary computers", Proc.IRE, vol.49,pp. 67-91, 1961.
[6] C.S. Wallace, "A suggestion for fast multipliers", IEEE Trans. Electronics Comput., vol. EC-13, pp.14-17, Feb.1964.
[7] Parhami, Behrooz, "Computer Arithmetic: Algorithms and Hardware Designs", Oxford University Press 2000.
[8] S.Shah, A.J.Kbalili,D.Al-Khabili "Comparison of 32-bit multipliers for various performance measures", The 12th International Conference on Microelectronics Tehran, Oct. 31- Nov. 2, 2000.

Author Profile

Jyoti Sharma received the B. Tech. degree in Electronics and communication Engineering from Matu Ram Institute of Engineering and Management in 2012. Now she is pursuing M.Tech from Meri College of Engineering & Technology, Sampla (Haryana). She got 65th rank in all Haryana Talent ship Competition.