2581-4575

International Journal For Recent Developments in Science & Technology

Crossref

A Peer Reviewed Research Journal

IMPLEMENTATION OF LOW POWER 4 BIT DADDA MULTIPLIER USING DADDA ALGORITHM AND OPTIMIZED FULL ADDER

M.LAVANYA[1] , K.SAIFUDDIN[2]

[1] Assistant Professor, Dept of ECE, Tadipatri Engineering College, Tadipatri,AP, India.

[2] Associate Professor ,Dept of ECE, SRIT, Anantapuramu , AP, India.

Abstract: This paper presents the model of 4-bit multiplier having low power and high speed using Algorithm named Dadda and the basic building block used is optimized Full adder having low power dissipation and minimum propagation delay. Full and half adder blocks have been designed using pass-transistor logic and CMOS process technology to reduce the power dissipation and propagation delay. We have also applied Dadda algorithm to reduce the propagation delay. Multiplication is the basic process which is used in different electronic and in various digital communication applications. Multipliers with low latency and minimum power dissipation are preferred to design an optimized circuit so that maximum throughput can be achieved in minimum response time. Building blocks used in multipliers are a full adder and a half adder.

1. INTRODUCTION

With the rapid advances in multimedia and communication systems, real-time signal processing and large capacity data processing are increasingly being demanded. The multiplier is an essential element of the digital signal processing such as filtering and convolution. Most digital signal processing methods use nonlinear functions such as discrete cosine transform (DCT) or discrete wavelet transform (DWT). As they are basically accomplished by repetitive application of multiplication and addition, their speed becomes a major factor which determines the performance of the entire calculation. Since the multiplier requires the longest delay among the basic operational blocks in digital system, the critical path is determined more by the multiplier. Furthermore, multiplier consumes much area and dissipates more power. Hence designing multipliers which offer either of the following design targets – high speed, low power consumption, less area or even a combination of them is of substantial research interest. Multiplication operation involves generation of partial products and their accumulation. The speed of multiplication can be increased by reducing the number of partial products and/or accelerating the accumulation of partial products. Among the many methods of implementing high speed parallel multipliers, there are two basic approaches namely Booth algorithm and Wallace Tree compressors. This paper describes an efficient implementation of a high speed parallel multiplier using both these approaches. Here two multipliers are proposed. The first multiplier makes use of the Radix-4Booth Algorithm with 3:2 compressors while the second multiplier uses the Radix-8 Booth algorithm with 4:2compressors. The design is structured for m x n multiplication where m and n can reach up to 126 bits. The number of partial products is n/2 in Radix-4 Booth algorithm while it gets reduced to n/3 in Radix-8 Booth algorithm. The Wallace tree uses Carry Save Adders (CSA) to accumulate the partial products. This reduces the time as well as the chip area. To further enhance the speed of operation, carry-look-ahead(CLA) adder is used as the final adder

2. ABOUT DADDA MULTIPLIER

The Dadda multiplier is a hardware multiplier design invented by computer scientist Luigi Dadda in 1965. It is similar to the Wallace multiplier, but it is slightly faster (for all operan sizes) and requires fewer gates (for all but the smallest operand sizes). In fact, Dadda and Wallace multipliers have the same three steps for two bit strings A1 and A2 of lengths L1 and L2 respectively.

1. Multiply (logical) each bit of W1, by each bit of W2, yielding results, grouped by weight in columns reflecting the magnitude of the original bit values in the multiplication. For example, the product of bits *anbm* has weight *n+m*

2. Reduce the number of partial products by stages of full and half adders until we are left with at most two bits of each weight.

3. Add the final result with a conventional adder.

As with the Wallace multiplier, the multiplication products of the first step carry different weights reflecting the magnitude of the original bit values in the multiplication. For example, the product of bits has weight. Unlike Wallace multipliers that reduce as much as possible on each layer, Dadda multipliers attempt to minimize the number of gates used, as well as input/output delay. Because of this, Dadda multipliers have a less expensive reduction phase, but the final numbers may be a few bits longer, thus requiring slightly bigger adders.

Low Power 4×4 Bit Multiplier Design using Dadda Algorithm and Optimized Full Adder Abstract- This paper presents the model of 4-bit multiplier having low power and high speed using Algorithm named Dadda and the basic building block used is optimized Full adder having low power dissipation and minimum propagation delay. Full and half adder blocks have been designed using pass-transistor logic and CMOS process technology to reduce the power dissipation and propagation delay. We have also applied Dadda algorithm to reduce the propagation delay. The model has been designed using Cadence Virtuoso in 90-nm technology. The proposed multiplier starts its operation at the frequency of 3.83 GHz and its average dynamic power is 184.3µW at the supply of 1V.

3. DADDA MULTIPLIER

Multipliers are among the fundamental components of many digital systems and, hence, their power dissipation and speed are of primary concern. For portable applications where the power consumption is the most important parameter, one should reduce the power dissipation as much as possible. One of the best ways to reduce the dynamic power dissipation, henceforth referred to as power dissipation in this paper, is to minimize the total switching activity, i.e., the total number of signal transitions of the system.

Multiplication plays an essential role in computer arithmetic operations for both general purpose and digital signal processors. For computational extensive algorithms required by multimedia functions such as finite impulse response (FIR) filters, infinite impulse response (IIR) filters and fast Fourier transform (FFT), the percentage of power consumption occupied by multiplication shows the importance itself.

In a popular multiplication scheme the array, the summation proceeds in a more regular, but slower manner, to obtaining the summation of the partial products .Using this scheme only one row of bits in the matrix is eliminated at each stage of the summation.

In a parallel multiplier the partial products are generated by using array of AND gates. The main problem is the summation of the partial products, and it is the time taken to perform this summation which determines the maximum speed at which a multiplier may operate. The Dadda scheme essentially minimizes the number of adder stages required to perform the summation of partial products. This is achieved by using full and half adders to reduce the number of rows in the matrix number of bits at each summation stage**.**

Dadda multipliers are a refinement of the parallel multipliers presented by Wallace. Dadda multiplier consists of three stages. The partial product matrix is formed in the first stage by N2 AND stages. In the second stage, the partial product matrix is reduced to a height of two. Dadda replaced Wallace Pseudo adders with parallel (n, m) counters. A Parallel (n, m) counter is a circuit which has n inputs and produce m outputs which provide a binary count of the ONEs present at the inputs. A full adder is an implementation of a (3, 2) counter which takes 3 inputs and produces 2 outputs. Similarly a half adder is an implementation of a (2, 2) counter which takes 2 inputs and produces 2 outputs.

In Dadda multipliers that reduce the number of rows as much as possible on each layer, Dadda multipliers do as few reductions as possible. Because of this, Dadda multipliers have less expensive reduction phase, but the numbers may be a few bits longer, thus requiring slightly bigger adders.

In general, the product, p, of two n-bit unsigned binary numbers x and y may be expressed as follows: (p(2n-1) p(2n-2)……p2 p1 p0)

In a parallel multiplier, the terms yi ^ (xn-1 - . . . x0) are known as the partial products and are generated using an array of AND gates. For a parallel multiplier, the shifting term 2i is inherent in the wiring and does not require any explicit hardware. Thus the main problem is the summation of the partial products, and it is the time taken to perform this summation which determines the maximum speed at which a multiplier may operate. The realization of a parallel multiplier for digital computers has been considered in [7] by C.S. Wallace, who proposed a tree of pseudo-adders (that means adders without carry propagation) producing two numbers, whose sum equals the product. This sum can be obtained by applying the two numbers to a carry-propagating adder. Consider the process of multiplication of two binary numbers, each composed of n bit, as been based on obtaining the sum of v summands. These summands are obtained, in the simplest schemes, by shifting left the multiplicand by 1, 2, 3,….(n-1) places, and multiplying it by the corresponding bits of the multiplier. In this situation v = n. Now the number of summands can be made less than n by using some multiples of the multiplicand, on the basis of two or more multiplier digits. This architecture is based on the use of logical blocks called it as parallel (n, m) counters, these are combinational networks with m outputs and n(≤ 2m) inputs. The m outputs, considered as a binary number, codify the number of « ones» present at the inputs.

To achieve this, the structure of the second step is governed by slightly more complex rules than in the Wallace tree. As in the Wallace tree, a new layer is added if any weight is carried by three or more wires. The reduction rules for the Dadda tree, however, are as follows: Take any three wires with the same weights and input them into a full adder. The result will be an output wire of the same weight and an output wire with a higher weight for each three input wires. If there are two wires of the same weight left, and the current number of output wires with that weight is equal to 2 (modulo 3), input them into a half adder. Otherwise, pass them through to the next layer. If there is just one wire left, connect it to the next layer. This step does only as many adds as necessary, so that the number of output weights stays close to a multiple of 3, which is the ideal number of weights when using full adders as 3:2 compressors. However, when a layer carries at most three input wires for any weight, that layer will be the last one. In this case, the Dadda tree will use half adder more aggressively (but still not as much as in a Wallace multiplier), to ensure that there are only two outputs for any weight. Then, the second rule above changes as follows: If there are two wires of the same weight left, and the current number of output wires with that weight is equal to 1 or 2 (modulo 3), input them into a half adder.

4. BLOCK DIAGRAM

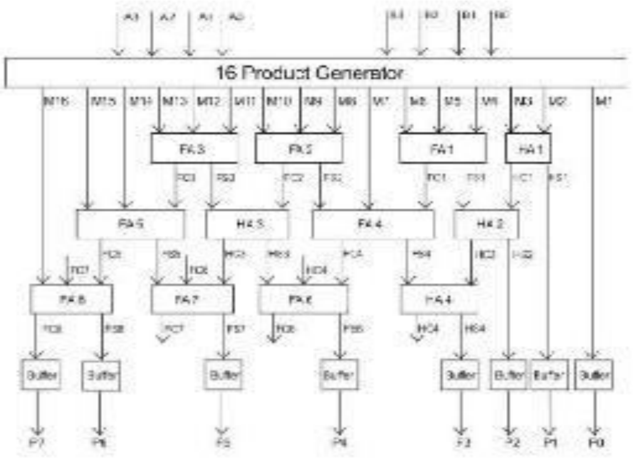


Fig:3.1 Block diagram of partial product generator of 4×4 bit multiplier

5. HALF ADDER AND FULL ADDER

An adder is a digital logic circuit in electronics that implements addition of numbers. In many computers and other types of processors, adders are used to calculate addresses, similar operations and

table indices in the ALU and also in other parts of the processors. These can be built for many numerical representations like excess-3 or binary coded decimal. Adders are classified into two types: half adder and full adder. The half adder circuit has two inputs: A and B, which add two input digits and generate a carry and sum. The full adder circuit has three inputs: A and C, which add the three input numbers and generate a carry and sum.

6. STAGES OF DADDA ALGORITHM

The objective is to reduce the height of the tree from four to two. Therefore, building blocks have been used in such a way to reduce the tree height from four to three after the completion of first Dadda stage and then from three to two after the completion of the second Dadda Stage. Furthermore, the two steps are again again added. Dadda stages are used to reduce this tree height. The stages are explained below:

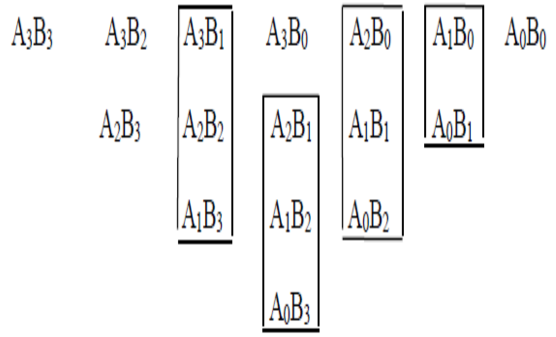
Partial products generation for the existing system at different stages Multiplication of multiplier and multiplicand will generate partial products,

We have considered A3, A2, A1, A0 and B3, B2, B1, B0 are the inputs to the multiplier. The multiplication process as follows:

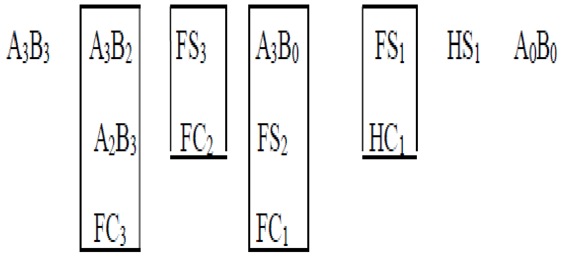
$$\begin{array}{cccccccc}
 & & & & A_3 & A_2 & A_1 & A_0 \\
 & & & & \times B_3 & B_2 & B_1 & B_0 \\
\hline
 & & & & A_3B_0 & A_2B_0 & A1B_0 & A_0B_0 \\
 & & & A_3B_1 & A_2B_1 & A_1B_1 & A_0B_1 & \\
 & & A_3B_2 & A_2B_2 & A_1B_2 & A_0B_2 & & \\
 & A_3B_3 & A_2B_3 & A_1B_3 & A_0B_3 & & & \\
\hline
P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
\end{array}$$

The above generated P7, P6, P5, P4, P3, P2, P1, P0 are the partial products generated from the multiplier

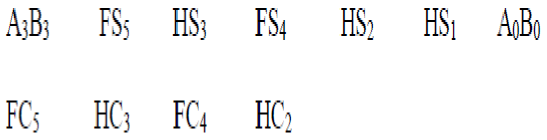
STAGE 1



STAGE 2



STAGE 3



STAGE 4

P_7
P_6
P_5
P_4
P
P_2
P_1
P_0

CONCLUSION & FUTURE SCOPE

The proposed model having high speed, low latency and minimum delay are being designed which has two modified circuits in it. One is hybrid full adder which has been designed by Pass transistor logic and CMOS process technology. Hybrid full adder has low propagation delay by which maximum throughput can be achieved in minimum response time. A highspeed 4*4 multiplier has been designed using the hybrid Full adder as its building block and Dadda Algorithm has been applied to achieve this. The proposed 4*4 multiplier operates at a frequency of 3.84 GHz and has an average dynamic power of 181.8 μW with a delay of 0.09 nS which is higher as compared to existing multiplier designs. Multiplier having low latency, minimum power dissipation and less layout area have been designed by the proposed model.

BIBLIOGRAPHY

[1] Zain Shabbir, Anas Razzaq Ghumman, Shabbir Majeed Chaudhry, A Reduced-sp-D3Lsum Adder-Based High Frequency 4 × 4 Bit Multiplier Using Dadda Algorithm, Springer Science+Business Media New York 2015.

[2]Design of high-speed carry saves adder using carry lookahead adder. Available from: https://www.researchgate.net/publication/301407573_Design_of_ high_speed_carry_save_adder_using_carry_ lookahead_adder [accessed Sep 22, 2017].

[3] S. Z. Naqvi, S. Z. Hassan and T. Kamal, "A power consumption and area improved design of IIR decimation filters via MDT," 2016 International Conference on Intelligent Systems Engineering (ICISE), Islamabad, 2016.

[4] A. Mukhtar, H. Jamal and U. Farooq, "An area efficient interpolation filter for digital audio applications," in IEEE Transactions on Consumer Electronics, vol. 55, no. 2, pp. 768772, May 2009.

[5] Stephen P. Boyd, Seung-Jean Kim, Dinesh D. Patil, Mark A. Horowitz, Digital Circuit Optimization via Geometric Programming, Operations Research, v.53 n.6, p.899-932, November-December 2005.

[6]P. Prem Kumar, K. Duraiswamy, and A. Jose Anand, "An optimized device sizing of analog circuits using genetic algorithm,"European Journal of Scientific Research,vol.69,no.3,pp.441–448, 2012.