HIGH PERFORMANCE FUSED ADD MULTIPLY OPERATOR

1Assistant Professor, 2,3,4,5Student Members
Department of Electronics and Communication Engineering
Coimbatore Institute of Engineering and Technology

Abstract: - In some Digital Signal Processing (DSP) applications, there are no such simple arithmetic operations are used to process. To increase the total output and to reduce the complexity of arithmetic operations, we have designed an Add-Multiply operator that is fused in a single module that directly recodes the sum of two numbers in its modified booth (MB) form and uses Wallace Carry save adder for the partial product addition. The technique focuses on FAM design by using prefix adders in the last stage of partial product addition that reduces both power consumption and delay thus increasing the efficiency and making it adaptable for a low power consumption application.

Keywords: Modified Booth algorithm, Wallace Carry Save adder, Parallel Prefix adders.

1. Introduction
Due to the rapid advancements in the modern consumer electronics that makes use of Digital Signal Processing (DSP). Low power usage and high performance are the main requisites to meet the requirements of various applications. The performance of these systems is mainly based on arithmetic operations as their implementation depends on Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), Finite Impulse Response (FIR) filters and signals convolution. To agitate the functioning of these arithmetic operations, faster and smaller multipliers are required. Modified Booth recoding is used for this purpose to make it more efficient differing from other simple methods.

In this work, we use a combined technique that performs simultaneously both addition and multiplication operation at the same instant of time. For this, a specially designed signed bit adders are used. Here, we use a radix-4 modified booth’s algorithm to increase the speed of multiplication operation. As the radix size increases, the number of partial products are drastically reduced, which in turn uses small adder tree for the addition of partial products, which in turn increases speed and reduces area.

The fused op adder. In the existing system carry look ahead or ripple carry adder is used at the last stage. Here, we can replace it with parallel prefix adder that reduces delay and the power that is consumed during the operation as the addition is performed in a parallel fashion.

2. Existing Design
In conventional design of AM operator addition and multiplication are performed separately. First addition is done by using any conventional adder and then the input and the result of adder are driven to a multiplier to get the output. The disadvantage of using a separate conventional adder is that it inserts a significant critical path delay. As the bit width increases critical path delay also increases, because the carry signals are to be propagated inside the adder. To overcome this we used a Carry Look Ahead (CLA) adder, which however
increases area and power dissipation. In order to reduce this area and power dissipation, an optimized design of the AM operator which combines the adder and MB encoding unit into a single data path block by direct recoding the sum of two numbers into its modified Booth form as in figure 1.

The fused Add Multiply (FAM) operator uses only one adder at the last stage of addition; as a result it reduces both area and critical path delay. A way to increase the speed of multipliers is through the use of Modified Booth algorithm to generate the partial products. The generated partial products are added by using Wallace tree instead of normal full adders. The carry look ahead adder is performed for the sum and carry bit generated from the Wallace CSA at the last stage for the final result. CLA calculates the carry in advance based on input bits. The carry is obtained in two cases: In first case when both inputs are 1 and in the second case when either of the input is 1 and the carry from the previous stage is 1. It generates two internal signals: carry propagate and generate signals. The carry propagate signal denotes whether the carry is passed to the next stage or not. It happens when either of the input is 1. The carry generate signal is 1 when both inputs are 1.

3. Proposed Design

This paper focuses on the efficient design of the fused AM unit which implements the operation \( Z = X. (A + B) \). The parallel prefix adder is used at the last stage of fused Add-Multiply (FAM) as shown in figure 2.

3.1 S-MB recoding

In this we recode the sum of two consecutive bits of the input A with two consecutive bits of the input B into one MB digit \( y_j^{MB} \). S-MB recoding is performed in 3 different schemes for this a set of bit level Half Adders (HAs) and Full Adders (FAs) are developed.

3.1.1 S-MB1 Recoding

In S-MB1 recoding technique, we used conventional and signed FA for both odd and even number of bit width of input numbers. In order to form the MB digit \( y_j^{MB} \), \( 0 \leq j \leq k-1 \) we need 3 bits \( (s_{2j+1}, s_{2j}, c_{2j}) \) which are the outputs of jth recoding cell having the inputs \( a_{2j}, a_{2j+1} \) and \( b_{2j}, b_{2j+1} \). The sum bits \( s_{2j+1} \) and \( s_{2j} \) are extracted from the jth recoding cell and \( c_{2j} \) bit is extracted from the conventional FA having the inputs \( a_{2j-1}, b_{2j-1} \) and \( b_{2j-2} \).

3.1.2 S-MB2 Recoding

In S-MB2 recoding technique, we used signed FA and HA for both odd and even number of bit width of input numbers. Initially we consider \( c_0 = c_1 = 0 \), \( 0 = 2 \). In order to form the MB digit \( y_j^{MB} \), \( 0 \leq j \leq k-1 \) we need 3 bits \( (s_{2j+1}, s_{2j}, c_{2j}, 2) \) which are the outputs of jth recoding cell. As in the S-MB1 recoding scheme, we use a FA to produce the sum \( s_{2j} \) and the carry \( c_{2j+1} \) with \( a_{2j}, b_{2j}, c_{2j}, 1 \) as inputs. The bit \( c_{2j}, 1 \) is the output carry of the conventional HA of the previous recoding cell and has the bits \( a_{2j-1}, b_{2j-1} \) as inputs. The output bit \( s_{2j+1} \) is produced from the HA* which is negatively signed bit.
3.1.3 S-MB3 Recoding

The third recoding scheme is S-MB3. In this scheme, we use a conventional FA, signed HA and FA. Initially we consider c0, 1 = c0, 2 = 0. As same in the previous recoding schemes, we use a FA to produce sum s2j and the carry c2j+1 with a2j, b2j, c2j, 1 as inputs. The bit c2j, 1 is the output carry of the signed HA (HA*) of the previous recoding cell and the output bit s2j+1 is produced from HA** which is negatively signed.

![Adder-Multiplier Diagram](image)

Figure 2 Proposed Block Diagram

3.2 Radix-4 Modified Booth Recoding

Booth uses radix recoding to achieve high speed. As the radix size increases the numbers of partial products are reduced resulting in high speed. It is possible to reduce the number of partial products by half, using radix-4 booth recoding. It performs the process of recoding the multiplicand based on the multiplier bits. As we use radix-4, it will compare three consecutive bits at a time with overlapping technique.

4. Wallace CSA

Wallace CSA tree is implemented in two ways. One way among them is considering all bits in each column at a time and compresses them into two bits (a sum and a carry). Another way is to consider all bits in each four rows at a time and compresses them into two bits using 4:2 compressors, 3:2 compressors, full adders and half adders. The inputs to a column are the bits of the partial products and the carry bits from one column to the right and the sum bits that are generated within the same column. The outputs from a column are the carry bits to the column one to the left and the last two sum bits in that column that are passed to the prefix adder. Finally, the carry-save output of the Wallace CSA is leaded to a Parallel prefix adder to form the final result Z = X. (A+B).

5. Parallel Prefix adder

Parallel prefix adder differs in the implementation of carry generation block with that of CLA. There are many prefix...
adders. In this paper we use Brent-Kung adder, the structure is shown in figure 4. It is done in three steps: The fundamental step is to pre-calculate the propagation and generation signals.

\[ p_0, i = a_i \oplus b_i \]
\[ g_0, i = a_i \land b_i \]

In the later stages, these propagation and generation signals are calculated by taking the previous stage signals as inputs. The equations to generate these signals:
\[ p_j, i = p_{j-1}, i \land p_{j-1}, i-1 \]
\[ g_j, i = g_{j-1}, i \lor (p_{j-1}, i \land g_{j-1}, i-1) \]

In the second step prefix graphs are used to calculate the carries, and it describes the structure that performs this part. The third step is to generate the sum. The equations to generate the sum and carries are as below:
\[ c_i = g_j, i \]
\[ s_i = p_0, i \oplus c_i-1 \]

6. Conclusion

The design of Add-Multiply operator is used to implement the direct recoding of the sum of two numbers to its MB form. When compared to the existing schemes, the proposed recoding schemes deliver considerable improvements in both delay and power consumption. Regarding the overall performance S-MB2, S-MB3 based schemes will suit for low power applications where the power consumption is less compared to S-MB1.

REFERENCES


