Design of a Novel Radix-4 Booth Multiplier

Hsin-Lei Lin, Robert C. Chang, Ming-Tsai Chan
Department of Electrical Engineering,
National Chung Hsing University, Taichung, Taiwan

ABSTRACT
This paper presents a novel radix-4 Booth multiplier. A conventional Booth multiplier consists of the Booth encoder, the partial-product summation tree, and the carry-propagate adder. Different schemes are addressed to improve the area and circuit speed effectively. A novel modified Booth encoder/decoder is proposed and the summation column is compressed by the proposed MFAr. The proposed design is simulated by Synopsys and Apollo. It results 20% area reduction, 17%–24% power decrease, and 15% reduction of the delay time of the critical path.

1. INTRODUCTION
The multiplier using the Booth algorithm is a well-know technique for high-speed and low-cost multipliers. There are many researches on high-speed Booth multipliers, and the main technique is the radix-4 Booth encoding[1-6]. Although radix-4 Booth can reduce the input bits and the output bits to half, it also increases the time of compression. In order to get a better system performance, we have improved the circuit of the radix-4 Booth multiplier in this paper.

In the CPU and DSP processor design, we use the modified multiplier scheme widely and commonly. There are several types of multiplier such as series, parallel, array, and encoding. The property of these multipliers as we know that the series multiplier is the simplest structure, the parallel scheme is higher-speed, the matrix one is more difficult when it is used on symbol operation, and the encoding one is much more efficient when it is used on symbol operation. Therefore we modify the encoder and the decoder in order to reduce area and increase the whole speed.

This paper is organized as follows. Section II discusses the proposed radix-4 Booth multiplier which this paper is proposed. Section III compares the proposed radix-4 Booth multiplier structure with a standard one. Section IV is the conclusion.

2. Radix-4 Booth Multiplier
In this section, we present a novel scheme using the modified Booth encoder/decoder (MBE) and the re-modified full-adder (MFAr). It is improved from the Yen’s MBE [1] and the original 4-to-2 compressor to reduce the critical path and area. Figure 1 shows the proposed radix-4 Booth multiplier, which consists of the 3-bit Booth encoder/decoders, the compressors, and the carry-propagate adder [7-11]. The Booth encoder/decoder is the first part of the multiplier when we start to calculate the value of multiplicand and multiplier. Instead of the partial-product summation tree (PPST), the Booth encoder/decoder makes the calculation faster. The radix-4 MBE is useful for the parallel multiplier by 3-bit encoding if the bit number of the operation is not incredible large. The n-bit multiplicator input, denoted as X, is divided into 3-bit groups for the Booth encoder. The encoded information is for the n-bit multiplicand input, which is represented as Y, to get the n/2 rows partial product value after the decoding. After the decoder, we get the (n+3) bits of output at the first row, and the (n+2) bits at the others. There would be 2 bits left and shifted between each row, not including the first row, after the decoder. It still needs the compressor to simplify the counter for calculating the binary number equal to the number of logic-I inputs. Compressor calculates on n-bit input (n>2) into a 2-bit output [4]. The bit number of the multiplier becomes (2n+2) after the process of the compressors. The carry-propagate adder is used for the two outputs of the n+1 compressors, carry out and sum values, to result the final value of the product of X and Y.

![Fig. 1 The proposed radix-4 Booth multiplier](image-url)
2.1. Novel Radix-4 Booth Encoder/Decoder

The conventional radix-4 encoder algorithm is derived from the equations:

\[ P_1 = X_{2n+1}(X_{2n} \oplus X_{2n-1}) \]
\[ P_2 = X_{2n+1}X_{2n}X_{2n-1} \]
\[ Z = X_{2n+1}X_{2n} + X_{2n+1}X_{2n}X_{2n-1} \]
\[ M_1 = X_{2n+1}(X_{2n} \oplus X_{2n-1}) \]
\[ M_2 = X_{2n+1}X_{2n}X_{2n-1} \]

There are five outputs after the conventional radix-4 encoder. The following equation is the decoding algorithm.

\[ PPI_{i} = Z(M_1Y_{i} + M_2Y_{i-1} + P1Y_{i} + P2Y_{i-1}) \] (2)

Some of the radix-4 Booth encoder/decoders have beneficial property for area and timing [1]. Here, the new encoding circuit is designed by rebuilding the Booth encoder truth table into a new one shown in Table 1.

Table 1 Truth table of the new MBE scheme

<table>
<thead>
<tr>
<th>X_{2n+1}</th>
<th>X_{2n}</th>
<th>X_{2n-1}</th>
<th>X_{1b}</th>
<th>X_{2b}</th>
<th>Neg</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The novel encoding circuit is shown in Figure 2. It has a 3-bit input and generates a 3-bit output for the decoding circuit. Figure 3 shows the decoding circuit which receives the signals from the encoding circuit and generates the partial product results.

![Fig2. The novel encoding circuit](image1)

![Fig3. The novel decoding circuit](image2)

2.2. The Efficient Compressor

The compressor simplifies the multi-row partial-product decoded by the Booth decoder into two rows. Figure 4 shows the compressor structure for an 8-bit input. The number of bits in each column is different such that each column has different compression ratio. We construct a compressor with three 3-to-2 compressors and three 4-to-2 compressors. In Figure 4(a), the first column has no delay time that does not need to be compressed, the sixth column data with 3-bit input is compressed by a 3-to-2 compressor, and the eighth column is compressed by a 4-to-2 compressor. The sum of the output will be generated simultaneously if the numbers of compression bits are the same.

![Fig4 Compressors](image3)

Figure 5 shows a standard compressor. It is composed of 8 NAND gates and 4 XOR gates, so that there is a long delay time for a multiplier.

![Fig5 Compressors](image4)
Let \( U = I_2 \oplus I_1 \oplus I_0 \); \( X = \overline{U} \), we get the carry by \( U \), \( I_1 \), and \( Cin \).

\[
\text{Carry} = \overline{\overline{X}}(I_1 + Cin) + X(I_0,Cin)
\]
\[
= U(I_1 + Cin) + \overline{U}(I_0,Cin)
\]
\[
= UI_1 + UCin + I_0,Cin
\]

(3)

In order to shorten the delay time, the algorithm is rewritten as follows.

Let \( U = I_2 \oplus I_1 \oplus I_0 \), and \( T = \overline{I_1} \oplus Cin \)

\[
\text{Carry} = \overline{T}U + TI_1
\]
\[
= (\overline{T} Cin + I_0,Cin)U + (\overline{T} Cin + I_0,Cin)I_1
\]
\[
= UI_1 + CinU + I_0,Cin
\]

(4)

It demonstrates that equation (4) is equal to equation (3).

We restructure the compressor into the proposed one that we name it "MFAr". Here, we merge one NAND-gate and two NOR-gates into one XOR-gate, as shown in Fig.6. It not only decreases the delay time but also lowers the cost.

### 3. COMPARISION AND ANALYSIS

The proposed multiplier is implemented by Synopsys and Apollo library. Table 2 gives the comparison results between the new multiplier and the other four different kinds of the radix-4 Booth multiplier. Obviously, the numbers of the transistor of the proposed circuit is less than MBE-III and MBE-IV. From the delay time calculated by Synopsys, we can see that the proposed circuit is faster than MBE-I and MBE-II.

<table>
<thead>
<tr>
<th>Types of the modified Booth encoder</th>
<th>Transistor Delay(ns)</th>
<th>Delay(ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Apollo)</td>
<td>(Synopsys)</td>
</tr>
<tr>
<td>MBE-I [8]</td>
<td>10</td>
<td>3.0</td>
</tr>
<tr>
<td>MBE-II [9]</td>
<td>16</td>
<td>3.5</td>
</tr>
<tr>
<td>MBE-III [10]</td>
<td>20</td>
<td>2.0</td>
</tr>
<tr>
<td>MBE-IV [11]</td>
<td>18</td>
<td>2.0</td>
</tr>
<tr>
<td>Proposed</td>
<td>16</td>
<td>2.5</td>
</tr>
</tbody>
</table>

The row number of the partial product and the column bits are decreased after the encoding of the new modified Booth encoder/decoder, and the compression ratio is decreased, too. Figure 7 shows the column bits between the radix-4 Booth multipliers and the matrix multipliers. Because Booth encoder's number of bits are decreased, and thus the speed increases. Table 3 gives the comparison results between various compressors. The MFAr's performance is better than the others, no matter in delay time, area, and power. Regarding the delay time of the critical path, it can reduce about 15% than the others. Regarding the area cost, it can economize about 20%. Regarding the power consumption, it can decrease about 17%-24%. From the above, we can know that MFAr indeed can improve the circuit performance of the compressor.

### Table 3. Comparison results of various compressor.

<table>
<thead>
<tr>
<th>Performance</th>
<th>Delay(ns)</th>
<th>Area</th>
<th>Power(mw)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOI</td>
<td>1.34</td>
<td>21.4</td>
<td>2.75</td>
</tr>
<tr>
<td>MFA</td>
<td>1.16</td>
<td>18.4</td>
<td>2.99</td>
</tr>
<tr>
<td>MFAr</td>
<td>1.16</td>
<td>17.39</td>
<td>2.29</td>
</tr>
<tr>
<td></td>
<td>15%</td>
<td>20%</td>
<td>17%-24%</td>
</tr>
</tbody>
</table>

Note: AOI(tradition); MFA(modify full-adder into compressor); MFAr(re-modified full-adder tree)
The area of the radix-4 Booth multiplier is compared with the array multiplier by gate-count. The area of the radix-4 Booth multiplier is about \((N^{1/2} \text{ full adders}) + (N^{1/2} \text{ decoders})\), and the area of the array multiplier is about \((N \text{ full adders}) + (N^{1/2} \text{ AND gates})\). Assume the input has 16 bit. If the novel modified radix-4 Booth multiplier and the array multiplier both use the same compression method, then the array multiplier uses only 108 FAs. Therefore, the area of the matrix multiplier is about 1.5 times of that of the Radix-4 Booth multiplier. It can not only reduce the power consumption but also reduce the circuit complexity.

4. CONCLUSION

We have shown in this paper that the use of the new Booth encoder/decoder and the proposed compressor can truly decrease the circuit area and the delay time of the critical path. Table 2 shows that the proposed design has smaller area than MBE_III and MBE_IV and is faster than MBE_I and MBE_II. Table 3 gives that the area of the compressor is reduced to 80%, the delay time of the critical path to 85% reduction, and the power to 76%-83%.

5. ACKNOWLEDGMENT

This work was supported by the National Science Council of Taiwan under grant NSC 92-2220-E-005-004, and Meng Yao Chip Center.

6. REFERENCES