A Low-energy Hybrid Radix-4/-8 Multiplier for Portable Multimedia Applications

Gyeonghoon Kim, Seungjin Lee, Junyoung Park, and Hoi-Jun Yoo
Department of Electrical Engineering
KAIST
Daejeon, Republic of Korea
ghoon@einfo.kaist.ac.kr

Abstract—A hybrid radix-4/-8 multiplier is proposed for portable multimedia applications that demand high speed and low energy operation. Depending on the input pattern, the multiplier operates in the radix-8 mode in 56% of the input cases for low power, but reverts to the radix-4 mode in 44% of the slower input cases for high speed. For this, a mode detection circuit determines the mode signal from the input operand in just 2 gate delays. Based on the mode signal, the radix-4/-8 dual Booth encoder generates encoding signals in a hardware efficient way. Moreover, the carry save adder block is selectively activated to reduce power consumption. Compared to a conventional radix-4 multiplier, the proposed hybrid multiplier architecture consumes 33.5% less power at the expense of just 3.3% additional propagation delay, resulting in 31.3% less energy per operation.

I. INTRODUCTION

Multipliers are essential components in signal-processing for multimedia applications. While many previous works focused on implementing high-speed multipliers, recently there have been many attempts to reduce power consumption [2]. This is due to the increased demand for portable multimedia applications which require low power consumption as well as high speed operation.

However low-power multipliers without any consideration for high-speed are not the appropriate solutions of low-energy embedded signal processing for multimedia applications [1][4]. Previously, a hybrid radix-4/-8 modified Booth encoded (MBE) multiplier was proposed for low-power and high-speed operation. This multiplier architecture had separate radix-4 and radix-8 Booth encoders. Both encoders were operated regardless of the input patterns [2], resulting in power and area overhead. Therefore although its power consumption was reduced compared to the radix-4 architecture, its critical path delay was considerably increased. As a result, its energy efficiency was not improved over conventional radix-4 or radix-8 architectures [2].

We propose a hybrid architecture that determines the radix mode before multiplication and adaptively operates the Booth encoder and the Wallace tree in radix-4 mode or radix-8 mode depending on the input pattern. In the majority of the input cases, the radix-8 architecture is as fast as radix-4 architecture while consuming less power. However, in the remaining input cases, the radix-8 architecture is bottlenecked by the generation of the ±3B partial product term, which requires an additional carry propagation adding stage. Therefore, we detect the input cases that require the ±3B term (just 44% of the total inputs) and operate the multiplier as a radix-4 multiplier, which is faster at the cost of power increase. As a result, the critical path of the hybrid multiplier is only 3.3% longer than a radix-4 multiplier, while consuming 33.5% less power thanks to the low power radix-8 mode. The overall structure of the proposed hybrid multiplier is illustrated in figure 1.

The paper is organized as follows. Section II introduces the conventional radix-4 and radix-8 multiplier architecture. In Section III, the proposed hybrid radix-4 and radix-8 multiplier architecture and hardware components are described in detail. Implementation results are summarized in Section IV. Finally, conclusion of this paper is made in Section V.

II. CONVENTIONAL RADIX-4/RADIX-8 MULTIPLIER

The modified Booth encoded multiplier has two types, the radix-4 multiplier architecture and the radix-8 multiplier architecture. The difference between the two architectures, summarized in Table I, results from the difference in the number of bits of the input operand that are encoded. A radix-4 multiplier encodes 2 bit-segments of the input operand while the radix-8 encodes 3 bit-segments of the input operand. As a
result, the radix-8 architecture has a smaller number of partial products and thus lower power consumption. However, it is slower than the radix-4 architecture due to the higher complexity of the partial product generation stage.

In a radix-8 multiplier, the performance bottleneck occurs only when the encoding result contains ±3B terms. Table II shows all possible encoding results. In Table II, the probability that radix-8 Booth encoding multiplier generates ±3B terms is only four out of sixteen cases. If it is possible to encode partial product in a different way when ±3B term occurs, power consumption can be reduced remarkably with only one third of partial products compared to radix-4's partial product generation method. Moreover an adding step to produce ±3B term is unnecessary therefore the propagation delay of proposed hybrid multiplier is as same as a radix-4 multiplier. As a result, we can implement a hybrid radix-4/-8 MBE multiplier which satisfies both the high-speed of radix-4 and the low-power consumption of radix-8.

### III. PROPOSED HYBRID RADIX-4/RADIX-8 MULTIPLIER

A. Determining mode signal depending on input patterns

First of all, when multiplication inputs are provided and encoded with the radix-8 method, determining whether the encoding result contains ±3B terms or not is essential to determine radix mode. In Table II, we can observe that ±3B terms occur in a special pattern. Exploiting this pattern, ±3B terms can be detected using simple logic gates as shown in Figure 2. The mode detection circuit takes 4 bits of the input operand, and outputs a high logic value if a ±3B term is detected. Only two stages of logic consisting of two XOR gates and one AND gate is needed, resulting in just a small time overhead.

Compared to a radix-4 multiplier, the proposed hybrid multiplier has a small additional delay due to this mode detection step. However since the mode signal generation circuit has a delay of only two gate delays, its portion of the critical path delay is just 3.3%. Thus by accepting a slight additional delay, we can significantly reduce the power dissipation of the multiplier.

B. Hardware sharing: radix-4/-8 dual encoder

After detecting the ±3B term, an encoding signal is generated. Considering similarities between radix-4 and radix-8 encoding schemes enabled shared hardware architecture called the radix-4/-8 dual encoder block.

Figure 3 shows the booth encoding segmentation of an 8x8 multiplier for (a) radix-8 architecture and (b) radix-4 architecture [10]. Only the encoding for the last segment (x6, x7) is identical between the two architectures. Therefore, in a naive approach, only one radix-4 encoder block is shared, and two radix-8 encoder blocks and four radix-4 encoder blocks are necessary to provide both radix-4 and radix-8 encoding.

However, there is more possibility for hardware sharing when comparing with radix-4’s and radix-8’s encoding equation. For hardware sharing, we have to generate encoding signals of radix-4 or radix-8 within one block depending on a mode signal value. Figure 4 shows how to arrange bits to share hardware efficiently. Radix-4 mode does not need to use the MSB because it only uses 3-bit segment (including the last bit of the previous segment). On the other hand, radix-8 mode

### Table I. Comparison Between Radix-4 and Radix-8

<table>
<thead>
<tr>
<th>N x N multiplier</th>
<th>Radix-4 multiplier</th>
<th>Radix-8 multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
<td>Fast</td>
<td>Slow</td>
</tr>
<tr>
<td>Number of Partial product</td>
<td>N/2</td>
<td>N/3</td>
</tr>
<tr>
<td>Power consumption</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Utility</td>
<td>Wide</td>
<td>Limited</td>
</tr>
<tr>
<td>Area</td>
<td>Large</td>
<td>Small</td>
</tr>
</tbody>
</table>

### Table II. Radix-8 Partial Product Generation Table (AXB)

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Partial Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xi+2</td>
<td>Xi+1 Xi Xi-1 PPi</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>0</td>
<td>0 1 0 1</td>
</tr>
<tr>
<td>0</td>
<td>1 0 0 2B</td>
</tr>
<tr>
<td>0</td>
<td>1 1 0 3B</td>
</tr>
<tr>
<td>0</td>
<td>1 1 1 4B</td>
</tr>
<tr>
<td>1</td>
<td>0 0 0 -4B</td>
</tr>
<tr>
<td>1</td>
<td>0 1 0 -3B</td>
</tr>
<tr>
<td>1</td>
<td>0 1 1 0</td>
</tr>
<tr>
<td>1</td>
<td>1 0 0 -4B</td>
</tr>
<tr>
<td>1</td>
<td>1 0 1 -3B</td>
</tr>
<tr>
<td>1</td>
<td>1 1 0 -2B</td>
</tr>
<tr>
<td>1</td>
<td>1 1 1 0</td>
</tr>
</tbody>
</table>

![Figure 2. Mode signal generation circuit with input_A](image)

![Figure 3. Multiplier segmentation in two cases, (a) radix-8 and (b) radix-4. (c) Hardware sharing: radix-4/-8 dual encoding](image)

![Figure 4.](image)

1172
The generation circuit has the same architecture as the conventional radix-4 case, are required. Therefore, we can obtain the following four encoding signals except a tri-state buffer at the end. 

\begin{align}
\text{negative} &= \begin{cases} 
X_{n+2} & (\text{mode} = 0) \\
X_{n+1} & (\text{mode} = 1)
\end{cases} \\
\text{one} &= X_0 \oplus X_{n+1} \\
\text{two} &= X_0 X_{n+1} + X_{n+1} X_{n+2} \\
\text{four} &= X_0 X_{n+1} X_{n+2} + X_{n+1} X_{n+2} X_{n+3} \quad (\text{mode} = 0) \\
&= X_{n+1} X_{n+2} X_{n+3} + X_{n+2} X_{n+3} X_{n+4} + X_{n+3} X_{n+4} X_{n+5} + X_{n+4} X_{n+5} X_{n+6} \quad (\text{mode} = 1)
\end{align}

The one signal and the two signal are exactly the same in both radix-4 and radix-8 modes. A circuit which generates the negative signal is composed of one MUX. Also the four signal generation circuit has the same architecture as the two signal generation circuit except a tri-state buffer at the end.

Although radix-4 and radix-8 multipliers have different encoding methods, it is possible to design an encoding signal generation circuit that is available in both of the cases, radix-4 and radix-8 mode, by the proposed bit-arrangement. Without sharing, we need six blocks to encode both radix-4 and radix-8 mode. Now we can merge radix-4 into radix-8 encoding circuit except a tri-state buffer at the end. So expected power reduction in CSAs can be obtained as

\[100 \times \{1 - (0.56 \times 0.5 + 0.44 \times 1)\} = 28\% .\]  

D. Scalability of the proposed hybrid multiplier architecture

The amount of the proposed hybrid multiplier’s energy reduction goes up when the size of the multiplier is large. This is because decrease in the average number of partial product generation block creates one partial product, the number of activated blocks determines the number of partial products. As a result, it results in power reduction in the Wallace tree which will be described in following paragraphs.

C. Energy-efficient 4:2 compressor in Wallace tree

Wallace trees are composed of many CSAs and one fast CPA [7] [8]. Generally, we combine two CSAs into one 4:2 compressor for convenience as shown in Figure 6 (a) [9]. The number of partial products is different depending on the mode signal value. It means that we need additional controls to turn-off or turn-on CSAs based on the mode signal value which is already determined.

When mode signal value is high, four partial products are created as in a radix-4 multiplier so the four 4:2 compressor is fully activated as depicted in Figure 6 (a). On the other hand, when the mode signal is low, three partial products are created as in a radix-8 multiplier so only one CSA in the compressor has to be activated as shown in Figure 6 (b). As a result, half of the CSAs are activated if the mode signal is low (radix-8 mode) compared to the radix-4 mode. Therefore average power consumption of the Wallace tree can be reduced for this reason.

The portion of radix-8 mode can be calculated as follows:

\[1 - \left(\frac{12}{16} \times \frac{4}{16} + \frac{4}{16} \times \frac{12}{16} + \frac{4}{16} \times \frac{4}{16}\right) = 0.56 \]  

So expected power reduction in CSAs can be obtained as

\[100 \times \{1 - (0.56 \times 0.5 + 0.44 \times 1)\} = 28\% .\]  

In this paper, we describe two different hybrid multipliers. One is a hybrid multiplier between radix-4 and radix-8 multipliers. It is a proposed radix-4/-8 dual encoder block except a tri-state buffer at the end. Since this proposed hybrid multiplier is enable to to eliminate ±3B cases, four encoding signals exist (NEG, ONE, TWO, FOUR) except signal THREE. Therefore we can obtain four encoding signals exist (NEG, ONE, TWO, FOUR) except signal THREE. Therefore we can obtain four encoding signals except a tri-state buffer at the end.
A hybrid radix-4/radix-8 multiplier architecture which is proposed in this paper is an appropriate solution for portable multimedia applications because it is both low-power and high-speed. The proposed architecture is operated as radix-8 multiplier for low power consumption. When some input patterns which are the performance bottleneck of radix-8 multiplier architecture are detected, the multiplier turns into radix-4 multiplier architecture which has no performance bottleneck. However it does not imply even tradeoff between the high speed of radix-4 multiplier and the low power of radix-8 multiplier. This proposed hybrid radix-4/radix-8 multiplier architecture dissipates 33.5% less power in an 8x8 multiplier with only a 3.3% increase in delay, compared to the radix-4 multiplier architecture. As a result the hybrid multiplier consumes 31.3% less energy than the radix-4 multiplier architecture.

### REFERENCES