Improving the Energy/Power Consumption of Parallel Decimal Multipliers

Amin Malekpour and Alireza Ejlali

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran;
Malekpour@ce.sharif.edu; Ejlali@sharif.edu

Abstract

Decimal arithmetic has gained intensive attention in the last decade. Most commercial, financial, scientific, and internet-based applications need their data to be precise, while binary number system loses preciseness in some cases. The latency and area are two major factors in existing research works on decimal multiplication. However, energy/power consumption is another important factor in today’s digital systems. Hence, in this paper we proposed a new low power decimal adder based on prediction technique for decreasing the energy/power consumption of parallel decimal multiplication and show its impacts on one of the well-known parallel decimal multipliers architecture. Our observations show 11.5% improvement in terms of total power consumption and 10.13% improvement in terms of energy consumption.

Keywords: Decimal Adder, Decimal Multiplication, Energy Consumption, Parallel Multiplier, Power Consumption

1. Introduction

Decimal multiplication is a popular operation in computational systems. There is an intensive request for decimal computations in various applications such as commercial, scientific, financial and internet-based applications1.

A disadvantage of binary arithmetic is the lack of preciseness. Conventional binary algorithms should be changed to store data in decimal format and some numbers such as 0.2, cannot be accurately represented in binary number system. Therefore, implementation and design of decimal arithmetic algorithms draws attention of researchers. IEEE has added new decimal feature to the IEEE 754 standard for Floating-Point Arithmetic (IEEE 754-2008) which shows the importance of decimal arithmetic computations2.

Decimal multiplication is one of the widely used algorithms throughout the arithmetic systems3. It is more complicated than binary multiplication because in decimal multiplication, a considerably bigger digit set is being dealt4. Decimal multiplication is a complex, power hungry, and time consuming task. Therefore, in some recent processors (e.g. IBM power6 5, IBM z10 6 and IBM z900 7), it has been performed by iterative algorithms. However, there are several promising points for exploiting of parallel decimal multiplication algorithms in hardware. The hardware complexity of an iterative multiplier is much lower than parallel multiplier. However, since decimal operations are computationally intensive, the lower latency of a parallel multiplier makes it more attractive for decimal computations.

Conventional arithmetic circuits focus on delay and area as two parameters to be improved but energy/power consumption is another important factor because of limited power budget in embedded or portable systems and the issue of heat generation by fast and complex circuits8. For each pound of weight, conventional nickel-cadmium battery technology produces 20 W.h of energy9.

Even in those systems that do not have severe energy constraints, low-power design is critical10. As the chips density increase, more power dissipation leads to use of more complex cooling systems, which are costly to build, operate, and maintain8.

*Author for correspondence
Many algorithms have been presented for decimal multiplications. They mostly iterate through the multiplier and add the multiplicand digits to a register. Moreover, there are several algorithms proposed for parallel decimal multiplication, but they just focused on delay and area as the most important factors and did not study the energy/power consumption of parallel decimal multipliers.

In this paper, we propose a new low-power decimal adder based on prediction technique for decreasing the energy/power consumption of parallel decimal multipliers and show its impact on multiplier architecture proposed by Jaberipur and Kaivani\textsuperscript{13}.

The rest of this paper is organized as follows: In Section 2, we briefly describe some well-known parallel decimal multipliers. A theoretical foundation of leakage and dynamic power are provided in Section 3. In Section 4, we talk about our low-power addition technique. Our flow of implementation and synthesis of designs, and a comparison between proposed design and referred algorithm are provided and discussed in Section 5. Finally, Section 6 concludes the paper.

2. Parallel Decimal Multipliers

Various techniques have been proposed for implementing parallel decimal multiplication. In this section, we will describe some well-known parallel decimal multipliers. The algorithm of parallel decimal multiplication is constructed from three parts: Partial Product Generation (PPG), Partial Product Reduction (PPR), and redundant to non-redundant conversion. The PPG part generates a set of partial products consisting of the redundant and BCD digits. This set is sent as input to the PPR part. Decimal adders are widely used arithmetic modules in the PPR section of parallel decimal multipliers to reduce partial products and the PPR is the most complex and time-consuming part of a parallel decimal multipliers\textsuperscript{13,15}.

A new multiplier\textsuperscript{11} is presented which was the first implementation of parallel decimal multiplication. In the PPG part, only multiples \(-X, 2X, -2X, 5X\) and 10 are generated, where \(X\) is multiplicand. In order to select appropriate multiples, two multiplexer have been used. Finally in the PPG part of multiplier\textsuperscript{11}, two BCD digits and one carry bit have been added by means of a decimal adder. In the PPR part of multiplier\textsuperscript{11}, 6-level tree of decimal adders and counters have been used to add a carry-save operands with BCD operands.

A new PPG algorithm\textsuperscript{13} is proposed which only produces \(2X, 5X, (8X + \text{\(8\)}X), (9X + 9\times)\) to compose other multiples. In the partial product reduction part of multiplier\textsuperscript{13}, reduction tree with a 6-level is used. Figure 1 shows an abstract view of the PPR.

As Figure 1 shows, the 16 decimal adders in the first level of the PPR do not use any redundant digit (carry-in) and the other decimal adders use only half of the carries. A 9:4 counter and a 6:3 counter are used in this design to convert the unused carries to BCD digits. The last part of this multiplier is a 32-digit conditional speculative BCD adder.

In the PPG part of multiplier\textsuperscript{14}, the unsigned multiples of multiplicand \(X, 2X, 4X\) and \(5X\) are used. For the PPR part, two reduction schemes are proposed: delay-optimized and area-optimized. In both of these two schemes decimal adders are widely used to reduce partial product. Finally, for both architectures, in order to convert ODDS products to BCD digits, a redundant to non-redundant converter is used.

Two parallel decimal multipliers\textsuperscript{15} are presented which are improved designs of architecture\textsuperscript{13}. In the PPG part of radix-5 design, multiples \(-X, 2X, 2X, 5X\) and 10X are generated. Then a mixed \((4221/5211)\) partial products are generated. In the PPG part of radix-10 design, multiples \(X, 2X, 3X, 4X\) and 5X and \(d + 1\) partial products are generated. In order to reduce these decimal partial products in both designs, decimal adder trees have been used. Finally, both architectures\textsuperscript{15} have used a quaternary tree adder (proposed for redundant to non-redundant converting\textsuperscript{16}).

3. Energy/Power Consumption

Since clock frequencies and transistor counts have increased, power consumption is an important design constraint\textsuperscript{17}. Static and dynamic dissipations are two major source of the power consumption in CMOS-based circuits. The leakage through reverse-biased diodes is the

![Figure 1. Partial product reduction of multiplier\textsuperscript{13}.](image)
main reason of static dissipation\(^{18}\). The sub-threshold conduction through OFF transistors, contention current in ratioed circuits, and tunneling current through gate oxide are the other sources of static dissipation\(^{17}\).

Sources of dynamic dissipation are the short-circuit current during the switching transient and charging and discharging of total output capacitance\(^{17}\). Dynamic power can be determined by the following equation:

\[
P_{\text{dynamic}} = \alpha \cdot C_L \cdot V_{DD}^2 \cdot f_{\text{clk}}
\]

where \(\alpha\) is the probability of switching the signal, \(C_L\) is the capacitance, \(V_{DD}\) is the voltage, and \(f_{\text{clk}}\) is the clock frequency.

As can be seen in Eq. (2), total power dissipation is the sum of the leakage and dynamic power dissipation.

\[
P_{\text{total}} = P_{\text{dynamic}} + P_{\text{static}}
\]

Our observations show that, dynamic power is the main source of power consumption in parallel decimal multipliers and the amount of static power is small. In VLSI circuits, it will be possible to reduce the total power consumption of the design by reducing the switching activity.

Commonly used metrics in design of low-power circuits are power, the power-delay product, and the energy-delay product\(^{17}\). The power alone is not a good metric to compare designs because reducing the operational frequency would result in lower power consumption. So, in this paper we compare designs from the viewpoint of power-delay product -known as energy consumption- as well as power consumption, delay, and area.

4. Proposed Decimal Addition Technique

In order to decrease the energy/power consumption of parallel decimal multiplication in this paper, we focus on \(\alpha\) as the key factor in Eq. (1), since \(V_{DD}\) and \(C_L\) are related to synthesis technology, and operational frequency reduction leads to reducing the performance. Many papers work on the switching activity reduction as a key variable in reducing power consumption\(^{19–21}\).

Figure 1 is an example of PPR part in parallel decimal multipliers. As we demonstrated in Figure 1, decimal adders are the basic building blocks of the partial product reduction tree in parallel decimal multipliers. Consequently, reducing the power consumption of these decimal adders can significantly impact the total power consumption of parallel decimal multipliers.

A decimal adder is an arithmetic module with three inputs and two outputs. Figure 2 shows our proposed decimal addition technique. If \(S_1 = 0\) then the output values can be predicted; therefore, the original logic circuit of decimal adder has no switching activity which will lead to substantially reduce dynamic power consumption of the PPR part in parallel decimal multipliers.
As Figure 2 illustrates, the operations should be done for adding $a$, $b$ and $c$ by decimal adder\textsuperscript{13} are complex. These operations require 8 NOT gates and 55 two-input gates including AND, OR and XOR gates, which are complex and costly, while our prediction unit consists of 8 two-input OR gates and one two-input AND gate.

In this paper we have proposed a low power technique based on prediction for reducing the power consumption of decimal adders. In the proposed technique, if $C_{in}$ and one of inputs ($a$ or $b$) are zero then we will skip all the operations which should be done by original logic circuit of decimal adder. In this case, $Sum$ will be equal to third input and $C_{out}$ will be equal to zero. In this proposed method, we simply predict the outputs of a decimal adder based on its inputs.

As we mentioned in Section 2, since decimal adders are widely used arithmetic modules (mostly in the PPR part) and the PPR is the most complex, time consuming and energy/power consuming part of a parallel decimal multiplier, this technique can lead to significant reduction of the energy/power consumption in parallel decimal multipliers. The impact of this technique on the energy/power consumption reduction of a decimal multiplier\textsuperscript{13} would be discussed in the next section.

### 5. Implementation and Experimental Results

In this section, we initially propose our flow of implementation and synthesis of parallel decimal multipliers. After that we talk about logic synthesis and our experimental results.

#### 5.1 VHDL Description

All the designs in this paper are modeled with RTL VHDL. In order to manage the complexity of hardware, we used top-down methodology. In this scheme, unknown blocks of the designs are described with behavioral description and other blocks are modeled in RTL at each level. In the next step, we describe unknown blocks in detail until all designs are described in RTL. This scheme allows simulating the designs in each level of the hierarchy.

After the hardware description, we did pre-synthesize simulation. In the next step, in order to find our design's synthesis problems, we synthesized all designs using the Leonardo Spectrum synthesis software tool. Finally, in order to check the correctness of all the designs, we did post-synthesize simulation in which we simulated the synthesized designs with a large number of test vectors (using a random test generator).

#### 5.2 Logic Synthesis

During logic synthesis step, we have used Design-Compiler toolbox with TSMC 45 nm technology file. For achieving more precise power reports, all designs have been synthesized with three scripts. First of all, we synthesize the multipliers in order to make a SAIF file and a net-list. The second script is a .DO file in Modelsim that uses generated net-list from first script, and simulates the multipliers with large uniform random test vector—which is same for all multipliers- and dumps signals activity in the SAIF file. Finally, we use the last script to synthesize designs with new activity files. The synthesis parameters during logic synthesis have been shown in Table 1.

#### 5.3 Experimental Results

We have estimated the energy/power consumption, delay, and area of parallel decimal multipliers for 16-digit (64bit) BCD input operands. All of these parameters were obtained from the synthesis of verified RTL models coded in VHDL by Design-Compiler toolbox using TSMC 45nm and uniform random test vector, when the designs were optimized for power.

Table 2 shows the estimated delay, area and power consumption of implemented parallel decimal multipliers. As Table 2 shows, our prediction technique makes multiplier\textsuperscript{13} to perform 1.56% slower with 1.55% more hardware complexity.

As mentioned previously, power/energy consumption in today's digital systems, have recently gained more importance and many digital systems need low-power and low-energy sub systems. Table 3 shows the estimated total power and energy consumption of implemented parallel decimal multipliers. In this table, the power ratio and energy ratio are reported which are the total power or

#### Table 1. Synthesis parameters

<table>
<thead>
<tr>
<th>Wire Load Model</th>
<th>5K_hvratio_1_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Operating Voltage</td>
<td>1.1 V</td>
</tr>
<tr>
<td>Voltage Units</td>
<td>1 V</td>
</tr>
<tr>
<td>Capacitance Units</td>
<td>1.00 pF</td>
</tr>
<tr>
<td>Dynamic Power Units</td>
<td>1mW (derived from V,C,T units)</td>
</tr>
<tr>
<td>Leakage Power Units</td>
<td>1 pW</td>
</tr>
</tbody>
</table>
6. Conclusion

The power and energy consumption are two important features of the portable and non-portable digital systems and multiplication is one of the most widely used arithmetic modules. All previous works on parallel decimal multiplication focus on delay and area of these arithmetic modules as two important factors and propose some methods to improve these factors.

In this paper we have proposed a low power addition technique based on prediction to improving the energy/power consumption of decimal multiplication and showed its impact on multiplier architecture. Our experimental results show 11.5% improvement in terms of total power consumption and 10.13% improvement in terms of energy consumption.

7. References