High Speed Multiplier Based on the Algorithm of Chinese Abacus

Chien-Hung Lin, Shu-Chung Yi and Jin-Jia Chen

1Department of Electrical Engineering, National Changhua University of Education
2Graduate Institute of Integrated Circuit Design, National Changhua University of Education
The National Chaunghua University of Education
No.1, Jin-De Road, 50007, Changhua, Taiwan
lkhwin2004@yahoo.com.tw

Abstract—A 4x4 bit multiplier is demonstrated based on the Chinese abacus. As comparing the simulation result of this work with the speed and power consumption of the 4x4 bits Braun array multiplier, the delays of the 4-bit abacus multiplier are 19.7% and 10.6% less than that of Braun array multiplier with 0.35µm and 0.18µm technologies, respectively. Meanwhile, the power consumption of the 4-bit abacus multiplier is, respectively, less about 8.7% and 18% also.

Key Word—Braun array multiplier, Chinese abacus multiplier, fast multiplier.

1 Introduction

Multiplication is one of the most critical operations in many computational systems. Among various multiplier techniques, array-based multipliers [1][2] and tree-based multipliers [3] are the most well known techniques and are often used in the VLSI design for implementing fast multipliers. This study presents a multiplier based on the antique Chinese abacus algorithm to achieve an efficient operation with high speed and low power consumption.

The Chinese abacus is a very old and very popular invention, which has been used for centuries in China and other Asian countries to perform arithmetic functions. The basic architecture of the Chinese abacus is depicted in Fig.1, which demonstrates a decimal number of one hundred sixty-eight. Each column element has one higher bead with a weight of five and four lower beads, each with a weight of one. The key feature of the Chinese abacus is the use of one bead with weight five. This allows the user to minimize the transmission of rests. The first multiplier and adders employing the technique of the Chinese abacus are proposed by Gang et al. [4]–[6].

![Fig.1 Basic architecture of Chinese abacus coded with a decimal number of 168.](image-url)

The proposed Chinese abacus multiplier is based on an abacus adder in [9] that each column element, e.g. \((H_2H_1H_0|M_2M_1M_0|L_2L_1L_0)_{\text{abacus}}\), consists of three different weighted bead groups, which represent a decimal number to be

\[(H_2H_1H_0|M_2M_1M_0|L_2L_1L_0)_{\text{abacus}} = (H_2 + H_1 + H_0)16 + (M_2 + M_1 + M_0)4 + (L_2 + L_1 + L_0)\]
For instance, a multiplication operation is demonstrated in Fig. 2, where \( B = (b_3 b_2 b_1 b_0)_2 = (1101)_2 = 13 \) is the multiplicand and \( A = (a_3 a_2 a_1 a_0)_2 = (1110)_2 = 14 \) is the multiplier. This multiplication is first proceeded with two partial products, i.e., 
\[
(a_1 a_0)_2 \times (1101)_2 = (001|011|011)_{\text{abacus}}
\]
and 
\[
(a_3 a_2)_2 \times (1101)_2 = (011|001|111)_{\text{abacus}},
\]
and then the binary products of these two partial products give the product of \( B \times A \), which is denoted by a binary product to abacus (BPA), 
\[
(011|111|001|011)_{\text{abacus}} = 2 \times 4^3 + 3 \times 4^2 + 1 \times 4^1 + 2 \times 4^0 = 182.
\]

![Fig.2 Example of the multiplication based on the proposed algorithm.](image)

The block diagram of the proposed multiplier is depicted in Fig.3. The 4x4 abacus multiplier is divided into three modules. The first one is the BPA(binary product to abacus) module. The second one is the PA (parallel addition) module [9]. The third one is TB (Thermometric to Binary) [9]. These three modules are discussed in the following sections.

![Fig.3 Block diagram of the 4x4 abacus multiplier.](image)

### 2 The Design of the Proposed Multiplier

#### 2.1 The BPA Module

The block diagram of the BPA module is depicted in Fig.4. This module converts each 4x2 binary number, 
\[
(b_3 b_2 b_1 b_0)_2 \times (a_1 a_0)_2 \quad \text{and} \quad (b_3 b_2 b_1 b_0)_2 \times (a_3 a_2)_2,
\]
into an abacus representation (H\(_2\)H\(_1\)H\(_0\)|M\(_2\)M\(_1\)M\(_0\)|L\(_2\)L\(_1\)L\(_0\))\(_{\text{abacus}}\). (H\(_2\)H\(_1\)H\(_0\)) represents the three higher beads each with a weight of sixteen (4\(^2\)). (M\(_2\)M\(_1\)M\(_0\)) represents the three middle beads each with a weight of four (4\(^1\)), and (L\(_2\)L\(_1\)L\(_0\)) represents three lower beads each with a weight of one (4\(^0\)).

![Fig.4 The block diagram of BPA module.](image)

As shown in Fig.4, the BPA module also consists of three different sub-modules. The behavior of the BT module is modeled in equations (1) - (5):

\[
L_0 = (I_1 + I_0)(S_1 \cdot S_0) + (I_0)(S_1 \cdot \overline{S_0}) + (I_1 + I_0)(S_1 \cdot S_0)
\]

(1)

\[
L_1 = (I_1)(S_1 \cdot S_0) + (I_0)(S_1 \cdot \overline{S_0}) + (I_1 \oplus I_0)(S_1 \cdot S_0)
\]

(2)

\[
L_2 = (I_1 \cdot I_0)(S_1 \cdot S_0) + (I_1 \cdot I_0)(S_1 \cdot S_0)
\]

(3)

\[
H_0 = I_1 \cdot S_1
\]

(4)

\[
H_1 = (I_1 \cdot I_0)(S_1 \cdot S_0)
\]

(5)
The PR module adds the beads with the same weight and then transforms the beads into middle beads \( (K_2,K_3,K_4) \). The behavior of the PR module can be modeled in equations (6) - (11).

\[
\begin{align*}
    f_1 &= \overline{X_2} \cdot X_1 \\
    f_2 &= \overline{X_1} \cdot X_0 \\
    C_{out} &= (Y_1) f_1 + (0) f_2 + (0) \overline{X_0} + (Y_0) X_2 \\
    K_0 &= (\overline{Y_1}) f_1 + (1) f_2 + (Y_0) \overline{X_0} + (Y_1 + Y_0) X_2 \\
    K_1 &= (\overline{Y_1}) f_1 + (Y_0) f_2 + (Y_1) \overline{X_0} + (Y_0) X_2 \\
    K_2 &= (\overline{Y_1} \cdot Y_0) f_1 + (Y_1) f_2 + (0) \overline{X_0} + (\overline{Y_0}) X_2
\end{align*}
\]

The detailed circuit of the PR module is depicted in Fig.5. The PS module transfers the previous stage to higher beads. The behavior of the PS module is modeled in equations (12) - (14).

\[
\begin{align*}
    O_0 &= X_0 + C_{in} \\
    O_1 &= X_1 + C_{in} \ X_0 \\
    O_2 &= 0
\end{align*}
\]

An example to demonstrate the algorithm of the BPA is shown below:

\((B_3B_2B_1B_0)_2 = (1101)_2 = 13\) and \((A_{1A_0})_2 = (10)_2 = 2\)

\((1101)_2 * (10)_2 = (001[011][011])_{\text{abacus}} = (0*0+1)*16+(0+1+1)*4+(0+1+1)*1 = 26.\)

### 2.2 The PA (Parallel Addition) Module:

This module acts similarly as a multiplexer, which can count two column elements with the same weight and then transform the sum into a thermometric representation \(K_0 \cdots K_j\), in which \(0 \leq K_i \leq K_j \leq 1\) for \(i > j\).

As shown in Fig. 3, the number \((X_2X_1X_0)_2\) is the input signal of the multiplexer, and \((Y_1Y_0)\) is the selector used to modify the number \((X_2X_1X_0)_2\). The result gives a thermometric sum \((K_2K_3K_4K_5K_0)\). Note that there are only four configurations for each number \((X_2X_1X_0)_2\) or \((Y_1Y_0)\), i.e., 000, 001, 011, and 111.

The behavior of the PA module can be modeled in equations (15) – (22), and the detailed circuits, as shown in Fig.6. The PA module can count all the beads simultaneously.

\[
\begin{align*}
    f_1 &= \overline{X_2} \cdot X_1 \\
    f_2 &= \overline{X_1} \cdot X_0 \\
    K_0 &= (1) f_1 + (1) f_2 + (Y_0) \overline{X_0} + (1) X_2 \\
    K_1 &= (1) f_1 + (Y_0) f_2 + (Y_1) \overline{X_0} + (1) X_2 \\
    K_2 &= (Y_0) f_1 + (Y_1) f_2 + (Y_2) \overline{X_0} + (1) X_2 \\
    K_3 &= (Y_1) f_1 + (Y_2) f_2 + (0) \overline{X_0} + (Y_0) X_2 \\
    K_4 &= (Y_2) f_1 + (0) f_2 + (0) \overline{X_0} + (Y_1) X_2 \\
    K_5 &= (0) f_1 + (0) f_2 + (0) \overline{X_0} + (Y_2) X_2
\end{align*}
\]
2.3 The TB (Thermometric to Binary) Transformation Module

This module transforms the thermometric representation to binary numbers. It can convert the higher part or lower part numbers of $K_5 - K_0$ to binary numbers as shown in Fig.3. The output signals $S_1$, $S_0$ and $C_{out}$ are determined using the following equations:

\[
C_{out} = C_{in}K_2 + K_3
\]

\[
S_0 = C_{in}K_1\overline{K_0} + C_{in}K_3\overline{K_2} + C_{in}K_4\overline{K_5} +
C_{in}\overline{K_0} + C_{in}K_2K_1 + C_{in}K_4K_3 + C_{in}K_5
\]

\[
S_1 = C_{in}\overline{K_3}K_1 + C_{in}\overline{K_2}K_0 + C_{in}K_4 + K_5
\]

The detailed circuits of the TB module are depicted in Fig.7.
3 Simulations and Comparisons

In the previous sections we have designed a multiplier using the methodology of the Chinese abacus. The circuits of the prototype are simulated using HSPICE and the 0.35µm and 0.18µm TSMC CMOS technologies. For the sake of simplicity, all the lengths and widths of the transistors were set to be the smallest value allowed by each technology, and the width size of PMOS transistors was 2.5 times that of NMOS transistors. CMOS inverters were used as loads in each output in the simulations. The simulation was performed at a frequency of 50 MHz, with 0.35µm CMOS technology. All of the output and input waves are shown in Fig.8. The power consumption wave is also shown in Fig.9.

The simulation results are compared with those of the Braun array multiplier, as listed in Table 1. The delay is defined as the longest signal time from input to output with all input patterns. The 4x4 abacus multiplier results a delay of 2.83ns and 1.432ns for 0.35µm and 0.18µm TSMC CMOS technologies, respectively. These data are 19.7% and 10.6% less than those of the Braun array multiplier for the same 0.35µm and 0.18µm technologies, respectively.

The power consumption levels of various multipliers using different methodologies is also listed in Table 1. The average results of the power consumption are different for all input patterns. The simulation was performed at a frequency of 50MHz. For the 4-bit abacus multiplier power consumptions were 8.7 % and 18% less than those of the Braun array multiplier with 0.35µm and 0.18µm technologies, respectively. From these results, we conclude that the abacus multiplier still has competitive with the Braun array multiplier. The chip layout of the 4x4 abacus multiplier is shown in Fig.10.
TABLE 1. Simulation results of the 4x4 array multiplier and abacus multiplier

<table>
<thead>
<tr>
<th>Tech. [µm]</th>
<th>Braun</th>
<th>Abacus</th>
<th>Reduction % (compare to Braun)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay (ns)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.35</td>
<td>3.39</td>
<td>2.83</td>
<td>19.7%</td>
</tr>
<tr>
<td>0.18</td>
<td>1.584</td>
<td>1.432</td>
<td>10.6%</td>
</tr>
<tr>
<td>Power (µw)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.35</td>
<td>313</td>
<td>288</td>
<td>8.7%</td>
</tr>
<tr>
<td>0.18</td>
<td>45.2</td>
<td>38.3</td>
<td>18%</td>
</tr>
<tr>
<td>P<em>D (µw</em>ns)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.35</td>
<td>1061</td>
<td>815</td>
<td>30%</td>
</tr>
<tr>
<td>0.18</td>
<td>577.6</td>
<td>71.6</td>
<td>30%</td>
</tr>
</tbody>
</table>

4 Conclusion

We propose a multiplier based on the Chinese abacus algorithm that all the results are simulated by 0.18µm and 0.35µm TSMC CMOS technologies, respectively. As can be concluded from the simulation results, the delays level of the 4-bit abacus multiplier are 19.7% and 10.6% less than those of Braun array multiplier with 0.35µm and 0.18µm technologies, respectively; furthermore, the power consumption of the 4-bit abacus multiplier is about 8.7% and 18% less than those of the Braun array multiplier with 0.35µm and 0.18µm technologies. Obviously, this proposed abacus multiplier can significantly reduce the delay and the power consumption that it owns the competitive ability with respect to conventional fast multipliers.

Fig.10 The chip layout of 4x4 abacus multiplier in 0.35µm TSMC CMOS technology.

References: