ABSTRACT

Multiplication is one of the most common arithmetic operations employed in digital systems, but multipliers are the most time, area, and power consuming circuits. Improvement in any of these parameters can be advantageous for improving the efficiency of the circuit. The ever increasing need for development of efficient and high speed multipliers has motivated several researchers to go a step ahead and present some novel approach. This paper presents an approach towards the reduction of delay in the Wallace tree multipliers by using 4:2 compressors along with full-adders and half-adders, in the partial product reduction stage; and employing Kogge-Stone adder for the final addition. The proposed multiplier has been designed using Xilinx ISE Design Suite 14.7 and implemented for Spartan 3 FPGA.

INTRODUCTION

Multipliers are an inevitable part of systems like ALUs, DSPs and other processors. They are often the slowest component of the circuit which limits the overall performance of the system. Hence, reduction of processing delay has been a major concern in research. Wallace tree multipliers are quite fast among the available multipliers [1], as they use the carry save addition algorithm [2], but with the ever-increasing demand for faster operations, attempts are being made to make it even faster.

The basic process of multiplication involves three essential steps: generation of partial products, reduction of the partial products, and summation of the reduced partial products in order to produce the final product [1], [3]. The overall delay of the multiplication process may be reduced by applying delay-reduction techniques in any or all of the three stages.

PROPOSED WALLACE TREE MULTIPLIER

The proposed scheme for delay reduction in multipliers focuses on minimising the delays occurring in the partial product reduction and final addition step of the multiplication process. The partial products for N-bit by N-bit multiplication have been generated by performing bit-by-bit AND operation of the two operands using N² AND gates [1], the partial product reduction phase utilises the combination of 4:2 compressors, full-adders and half-adders, and the final addition of the last two rows, to obtain the product, is achieved using a high speed Kogge-Stone adder. Figure 1 shows the multiplication of two 8-bit numbers. The multiplication has been extended for 16-bit and 32-bits operands using the decomposition algorithm presented in [4],[5], where four units of 8-bit by 8-bit multiplier have been combined to produce a 16-bit by 16-bit multiplier, and four units of the resulting 16-bit by 16-bit multiplier have been combined to give 32-bit by 32-bit multiplication result.
Compressor

The 4:2 compressor in [6] accepts four inputs - \( \text{IN}_1, \text{IN}_2, \text{IN}_3, \text{IN}_4 \), and a carry-in bit, \( C_{\text{IN}} \) and produces two outputs - \( \text{SUM} \) and \( \text{CARRY} \), and a carry-out bit, \( C_{\text{OUT}} \). Here, the circuit of compressor has been used as a column adder which adds five bits at a time giving three outputs - a sum (LSB) and two carry outputs (MSBs) of equal weight. Figure 2 [6] shows the 4:2 compressor, also called as 5:3 counters in [7].

The equations of output signals in [6], [8] are given as follows:

\[
\text{SUM} = \text{IN}_1 \oplus \text{IN}_2 \oplus \text{IN}_3 \oplus \text{IN}_4 \oplus C_{\text{IN}} \quad (1)
\]

\[
\text{CARRY} = (\text{IN}_1 \oplus \text{IN}_2 \oplus \text{IN}_3 \oplus \text{IN}_4) \cdot C_{\text{IN}} + \left(\text{IN}_1 \oplus \text{IN}_2 \oplus \text{IN}_3 \oplus \text{IN}_4\right) \cdot \text{IN}_4 \quad (2)
\]

\[
C_{\text{OUT}} = (\text{IN}_1 \oplus \text{IN}_2) \cdot \text{IN}_3 + (\text{IN}_1 \oplus \text{IN}_2) \cdot \text{IN}_1 \quad (3)
\]

The use of compressors in the partial product reduction stage has aided in minimising the number of stages needed to reduce the partial products to two, as compared with the conventional Wallace tree multiplier design described in [1]. For 8-bit multiplication, the conventional Wallace multiplication approach needs 4 stages for reduction of partial products as described in [1], [9], whereas the proposed approach achieves the target in 3 stages only.

http://www.ijesrt.com
Kogge-Stone Adder

The Kogge-Stone adder is a parallel prefix adder, known for its efficient and performance-oriented design. The diagram of the prefix stage of a 12-bit Kogge-Stone adder has been shown in figure 3 [10]. This adder generates all the carry signals in parallel using the generate signal $G_i$, and propagate signal, $P_i$, calculated using the $i^{th}$ bit of input operands; thus reducing the computation delay. From the carry tree or prefix stage of the KSA we can calculate the carry signals using the relation in [10],

$$C_i = G_{i:0}$$ (4)

and the sum output will follow the relation in [10],

$$S_i = P_i \oplus G_{i-1:0}$$ (5)

Figure 3: Generation of Carry Signals in 12-bit Kogge-Stone Adder

RESULT

The design entry of the proposed multiplier has been done in VHDL, using Xilinx ISE Design Suite 14.7; synthesis reports have been generated using XST for XC3S1500FG676-4 FPGA, and simulation results have been obtained by ISim simulator.

The delay analysis, and device utilization summary (after implementation) of 8-bit by 8-bit, 16-bit by 16-bit, and 32-bit by 32-bit multipliers have been described in Table 1 and Table 2, respectively.

<table>
<thead>
<tr>
<th>Delay (in ns)</th>
<th>8-bit by Multiplier</th>
<th>8-bit Multiplier</th>
<th>16-bit by Multiplier</th>
<th>16-bit Multiplier</th>
<th>32-bit by Multiplier</th>
<th>32-bit Multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Combinational Path Delay</td>
<td>28.601</td>
<td>51.090</td>
<td>88.114</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logic Delay</td>
<td>13.797</td>
<td>21.511</td>
<td>34.375</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Route Delay</td>
<td>14.804</td>
<td>29.579</td>
<td>53.739</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Device Utilization Summary of Multipliers

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>8-bit by Multiplier</th>
<th>8-bit Multiplier</th>
<th>16-bit by Multiplier</th>
<th>16-bit Multiplier</th>
<th>32-bit by Multiplier</th>
<th>32-bit Multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of 4-input LUTs</td>
<td>181</td>
<td>819</td>
<td>3765</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of Occupied Slices</td>
<td>97</td>
<td>444</td>
<td>2021</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of Bonded IOBs</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average Fanout of Non-Clock Nets</td>
<td>3.45</td>
<td>3.44</td>
<td>3.53</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The simulation results of the 8-bit by 8-bit, 16-bit by 16-bit, and 32-bit by 32-bit multipliers have been shown in figure 4, 5, and 6, respectively.
CONCLUSION

This paper proposes improvement in the design of Wallace multiplier when considered with respect to the processing delay. Using 4:2 compressors in the partial product reduction stage along with full-adders and half-adders has proved to be advantageous in reducing the number of stages when compared with the conventional Wallace tree multiplier. Also, the delay occurring at the final summation stage has been minimised by the use of Kogge-Stone adder (a fast parallel prefix adder).

Further improvements in this design can be achieved by using modified Booth algorithm in generation of partial product (first stage).

REFERENCES


