EMMA – A Suggestion for an Embedded Multi-Precision Multiplier Array for FPGAs

Oliver A. Pfänder and Hans-Jörg Pflieger
Institute for Microelectronics, University of Ulm, Germany

Abstract—A non-monolithic top-down reconfigurable multiplier suitable for embedding in an FPGA structure called EMMA is presented. It is constructed of four individual partitions that can operate as separate multipliers but also concatenate to form a superior multiplier with increased precision and sign handling ability. The number of possible operation modes is limited in order to keep the reconfiguration overhead low. Control signals determine behavior and mode selection, while inactive partitions are disconnected from the supply to save power. EMMA can compute signed two’s complement numbers at up to 32×18-bit precision when all partitions are active and concatenated, or up to four separate 16×8-bit multiplications running simultaneously.

Introduction

- FPGA-based DSP designs require addition, multiplication, and MAC.
- Implementing multiplication in FPGAs using CLBs is area-inefficient and creates bottlenecks.
- More and more hard-wired multipliers are embedded into FPGA structures.

However, only coarse-grain word-length steps are possible with monolithic multiplier blocks.

Our objectives: Run-time application-sensitive word-length decision and power trade-off with customized and configurable embedded multiplier blocks.

EMMA is a non-monolithic multiplier array with four independent and combinable partitions, reconfigurable multi-precision and sign handling.

Basic Multiplication Scheme

\[
\begin{align*}
(a_1 &  a_2  a_3  a_4) \\
\times & (b_1  b_2  b_3  b_4) \\
\Rightarrow & (a_1b_1 + 2a_1b_2 + 4a_1b_3 + 8a_1b_4) \\
& + 2(a_2b_1 + 2a_2b_2 + 4a_2b_3 + 8a_2b_4) \\
& + 4(a_3b_1 + 2a_3b_2 + 4a_3b_3 + 8a_3b_4) \\
& + 8(a_4b_1 + 2a_4b_2 + 4a_4b_3 + 8a_4b_4) \\
& + 1 \times 0 \times 0 \times 1
\end{align*}
\]

Fig. 1: Partial products scheme with partial product inversion and correction term, shown for 4×4-bit numbers.

- Modified Baugh-Wooley scheme for two’s complement fixed-point signed numbers.
- Parallel/parallel building blocks.
- Multiplexer-based exchange of intermediate data.
- Sign extension column replaced by partial product negation in periphery cells and correction term.
- Grey shaded areas in Fig. 2 indicate overhead over conventional implementation (Multiplexers, XOR gates and controllable inverter). Blocks can be configured to handle unsigned or signed numbers.
- Concatenation to form superior multiplexer arrays is possible.
- Easy run-time reconfiguration with control signals only.

Control signals for multiplexers and special cells not shown. In basic cell is partial product inversion by CTRL. Same as above but with additional XOR for two CTRLs.

Fig. 2: Modified Baugh-Wooley two’s complement multiplier structure.

Partitioning Concept

Fig. 3: Schematic view of EMMA

- Adder expansion for complex multiplication.
- Radix-4 Booth-encoded multiplier scheme.
- Expand multipliers to accept fixed input values from look-up table.
- Include one-hot encoder for shift operations.

Input Interface

- Input operands A and B split up in MSB and LSB sub-words.

Fig. 4: EMMA is constructed from four individual partitions.

Table 1: Configuration modes for EMMA with respective precision and features.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Active Partitions(s)</th>
<th>Input Words</th>
<th>Result</th>
<th>Precision</th>
<th>Sign</th>
<th>Reg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>00.111,10.111,00.111</td>
<td>A × B, A × B</td>
<td>32×16-bit</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>B</td>
<td>01.111,10.111,00.111</td>
<td>A × B, A × B</td>
<td>32×8-bit</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>C</td>
<td>01.111,10.111,00.111</td>
<td>A × B, A × B</td>
<td>16×8-bit</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>D</td>
<td>01.111,10.111,00.111</td>
<td>A × B, A × B</td>
<td>16×8-bit</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>E</td>
<td>01.111,10.111,00.111</td>
<td>A × B, A × B</td>
<td>16×8-bit</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>F</td>
<td>01.111,10.111,00.111</td>
<td>A × B, A × B</td>
<td>16×8-bit</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>G</td>
<td>01.111,10.111,00.111</td>
<td>A × B, A × B</td>
<td>16×8-bit</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>H</td>
<td>01.111,10.111,00.111</td>
<td>A × B, A × B</td>
<td>16×8-bit</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>I</td>
<td>01.111,10.111,00.111</td>
<td>A × B, A × B</td>
<td>16×8-bit</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Conclusion

Compared to previous bottom-up reconfigurable multiplier arrays, our new approach called EMMA is a medium-grain embedded multiplier block for FPGAs that can be partitioned in four separate sub-blocks that are fully functional multipliers themselves. These partitions form various superior multipliers in different configurations using a small set of control signals, providing two’s complement sign handling and output buffering, respectively. Thus, EMMA can help to realize multi-precision multiplication with parallel operation, and power can be saved by turning off unused partitions. Potential applications that can benefit from this reconfigurable embedded multi-precision multiplier array include complex multiplication.