**A COMBINED ARITHMETIC-HIGH-LEVEL SYNTHESIS SOLUTION TO DEPLOY PARTIAL CARRY-SAVE RADIX-8 BOOTH MULTIPLIERS IN DATAPATHS**

**Abstract:**

While partial carry-save adders are easily designed by splitting them into several fragments working in parallel, the design of partial carry-save multipliers is more challenging. Prior approaches have proposed several solutions based on the radix-4 Booth recoding. This technique makes it possible to diminish the height of a multiplier by half, this being the most widespread option when designing multipliers, as only easy multiples are required. Larger radices provide further reductions at the expense of the appearance of hard multiples. Such is the case of radix-8 Booth multipliers, whose critical path is located at the generation of the 3X multiple. In order to mitigate this delay, in our prior works, we proposed to ﬁrst decouple the 3X computation and introduce it in the dataﬂow graph, leveraging the available slack. Considering this, we then present a partial carry-save radix-8 Booth multiplier that receives three inputs in this format, namely, the multiplicand, the multiplier, and the 3X multiple. Moreover, the rest of the data path is adapted to work in partial carry-save. In comparison with conventional radix-4 and radix-8 Booth-based data paths, the proposal is able to diminish the execution time and energy consumption while beneﬁts from the area reduction provided by the selection of radix 8. Furthermore, it outperforms priorstate-of-the-art partial carry-save multipliers based on radix 4.

**Index Terms**—Multipliers, Booth, radix 8, partial carry-save, slack.

**TOOLS:**

1. **XilinxISE 14.7**

**LANGUAGE:**

1. **VerilogHDL**