Efficient Booth Array Multiplier for Xilinx FPGAs

Martin Kumm

October 17, 2017

1 Introduction

This IP core provides a resource efficient implementation of a Booth Array Multiplier for Xilinx FPGAs. For details about the architecture see [KAZ15].

2 Interface

The top level entity is found in `mult_booth_array.vhd`. The generics as well as the portare described in Table 1 and Table 2, respectively.

Table 1: Description of the generics

Generic	Type	Default	Description
`word_size_a`	integer	8	Input word size operand A
`word_size_b`	integer	8	Input word size operand B
`sync_in_out`	boolean	false	If true, registers are placed at inputs and outputs (for timing results)
`use_pipelining`	boolean	true	If true, the multiplier is internally pipelined (highly recommented)

3 Simulation & Test

For simulation and test, the testbench (`tb_mult_booth_array.vhd`) was created which uses a random number generator together with assert statements to verify the designs (against a naive VHDL multiplication).

References

[KAZ15] M Kumm, Shahid Abbas, and Peter Zipf. An Efficient Softcore Multiplier Architecture for Xilinx FPGAs. In *IEEE Symposium on Computer Arithmetic (ARITH)*, pages 18–25, 2015.

Table 2: Description of the port

Generic	Direction	Type	Word Size	Description
clk_i	in	sl	1	Clock input (used when use_pipelining=true or sync_in_out=true)
rst_i	in	sl	1	Reset input (used when use_pipelining=true or sync_in_out=true)
ce_i	in	sl	1	Clock enable input (used when use_pipelining=true or sync_in_out=true)
a_i	in	slv	word_size_a	Input operand A
b_i	in	slv	word_size_b	Input operand B
p_o	out	slv	word_size_a+word_size_b	Product output $P = A \times B$