Rounded off unsigned constant division using add-shift in Verilog

Abstract

Even when sophisticated synthesis strategies are already being used to optimise the delay, area and power dissipation of Asic implementation, the quality of the results still heavily depend on the quality of the Register Transfer Level (RTL). In RTL design, multiplication and division by a constant number that is a power of two (e.g. 2, 4) can be done using the left shift (multiplication) and the right shift (division). Yet systems commonly multiply and divide by another constant number, such as by 3 or 7. It is also discovered that the implementation of division in hardware is expensive in term of area. This however can be overcomed by replacing the division with a cheaper adder and shifter (add-shift) that produces the same result. This paper presents the logic synthesis result of the add-shift scheme that was modified from existing algorithm and was described in Verilog code. The constant denominators (deno) were 3, 5, 6, 7 and 9 and the input variables (n) were of 13 bits. The modifications were to eliminate the integer multiplication, round off the unsigned result and maximise the sharing of common partial quotients for the five divisors. The logic synthesis was performed using Synopsis Design Compiler on two different technology libraries. Both 0.18µm Siltera and MIMOS 0.35µm technology libraries showed a significant optimization on power dissipation compared to normal division. However, the area was not optimized neither on Siltera nor MIMOS technology library.