

## CS220: Assignment#11

**1. [50 marks]** Write a module to implement the following slow version of the Booth's multiplication algorithm. The module takes a 32-bit multiplicand and a 32-bit multiplier in 2's complement format and a clock as inputs. On each rising edge of the clock, it examines the current multiplier bit and updates the product according to Booth's algorithm. To detect early termination, if the current multiplier bit being examined has value  $k \in \{0, 1\}$ , it also checks if the remaining bits of the multiplier are all  $k$ ; if yes, it updates the product appropriately and emits a done signal to the environment. The module also keeps track of the number of addition and subtraction operations done. The top module on a falling edge of the clock displays the multiplicand, multiplier, product, and the number of addition and subtraction operations if the done bit is set. On a falling edge of the clock, the top module also sends a new set of inputs to the multiplier provided done is one (done needs to be initialized to 1). To indicate a new set of inputs, the top module sends a bit to the multiplier as an additional input. This bit is reset to zero by the top module on each falling edge of the clock when it is not sending a new input. The multiplier resets the done bit when it starts working on a new set of inputs provided the multiplication doesn't terminate in the same cycle, in which case the done bit will stay one. Test your implementation with ten different input sets.

**2. [50 marks]** Write a module to implement the non-restoring division algorithm for unsigned 32-bit dividend and divisor. The module takes a 32-bit dividend, a 32-bit divisor, and a clock as inputs. To simplify the implementation, the lengths of the dividend and the divisor are also taken as inputs. On each rising edge of the clock, it executes one iteration of the algorithm. The first clock cycle is spent in aligning the divisor with the dividend. When the division completes, the module emits a done signal to the environment. The module also keeps track of the number of addition and subtraction operations done. The top module on a falling edge of the clock displays the dividend, divisor, quotient, remainder, and the number of addition and subtraction operations if the done bit is set. On a falling edge of the clock, the top module also sends a new set of inputs to the divider provided done is one (done needs to be initialized to 1). To indicate a new set of inputs, the top module sends a bit to the divider as an additional input. This bit is reset to zero by the top module on each falling edge of the clock when it is not sending a new input. The divider resets the done bit when it starts working on a new set of inputs provided the division doesn't terminate in the same cycle, in which case the done bit will stay one. Test your implementation with ten different input sets.

**Submission:** Please prefix the names of the Verilog files for Q1 with A11Q1\_ and for Q2 with A11Q2\_. Email the files as attachment to cs220spring2021submit@gmail.com. The subject line of the email MUST be the following (replace N by your group number): Assignment#11 submission for Group#N