Report 04

Filippo Nevi

November 2020

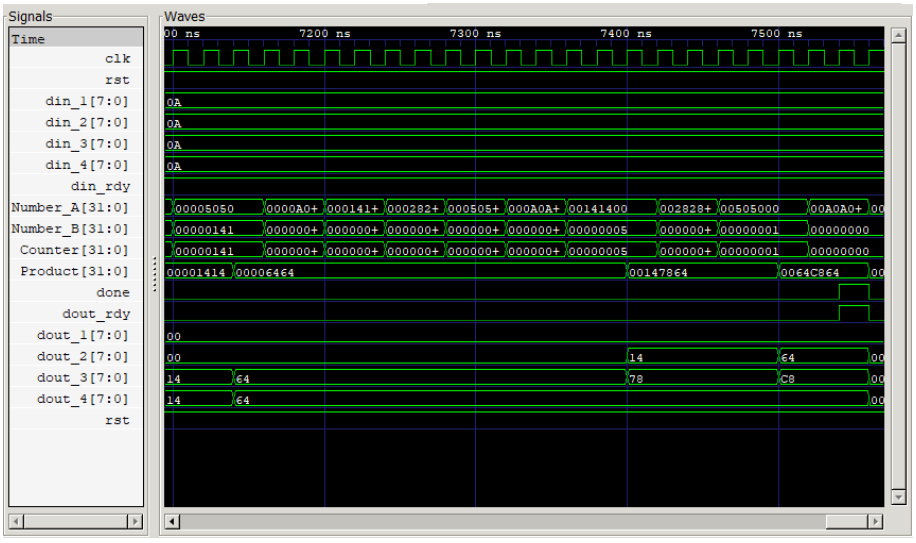
1 Design choices

All of the datatypes used in this implementation are synthesizable: `module`, `wire`, `reg`, `parameter`, `assign`, `always`, `case`, `if`.

The internal data is stored as **reg**s without considering the decimal representation (as for the respective implementations in *SystemC* and *VHDL*) because, being the numbers in Fixed-point notation, the integer and decimal parts can be statically split when printing the output results in the simulation scripts.

2 Waveforms

2.1 First diagram

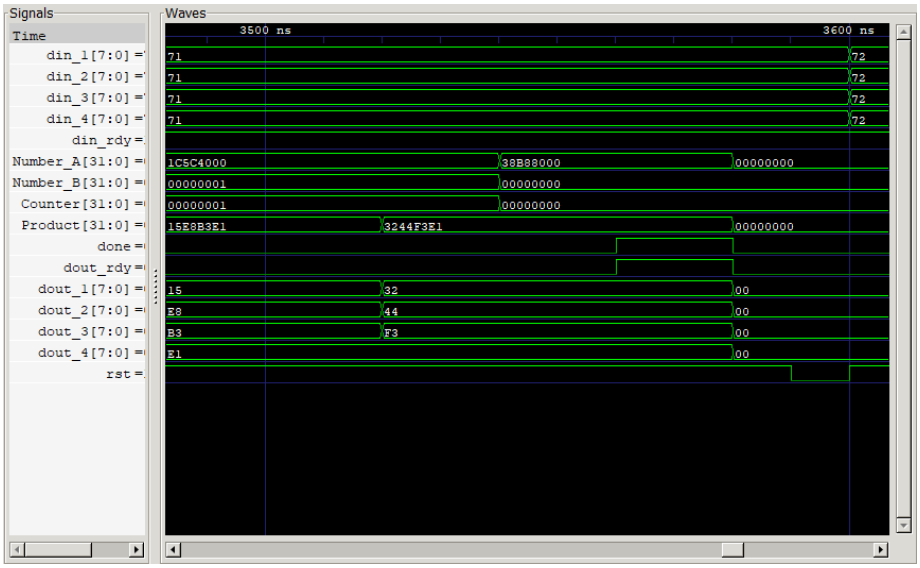


These waveforms show that the hexadecimal values of the input operands are `0A0A` and `0A0A`, and the product is `0064C864`. Translating these data in the fixed-point notation, we will obtain the following operation is:

$$10.0390625 \cdot 10.0390625 = 100.78277587890625$$

We can also notice that the register **done** and the wire **dout_rdy** have high value at the same time, because the signal on the wire is driven by the register using the `assign` statement.

2.2 Second diagram



This image shows another operation (in this case the hexadecimal operands are `7171` and `7171`), but it also shows that when the result is calculated and the FSM returns to the initial state, the registers are set to 0. Then, the execution of *stimuli.tcl* puts the **rst** signal to zero, and finally it requests another computation.