

Signals, Variables and Process Sensitivity

IN3160 Oblig 3, Spring 2022

Martin Mihle Nygaard (martimn@ifi.uio.no)

- a) The output data signal changes at 450 ns. Despite `indata` being changed at 200 ns, the change takes time to propagate through the `data1`–`data5` signals and variables. The signals `data1`, `data3`, and `data5` are only changed on a rising clock edge *after* the process defined in `delay.vhd` terminates; this means the variable assignments of `data2` and `data4` are delayed by a clock cycle each for a total of 200 ns. Also, the `indata` is not changed on a rising edge, delaying the process by a further 50 ns.
- b) The `outdata` changes at 750 ns and 850 ns to `F0` and `0F`, respectively. Now, the data uses a clock cycle for each update to the `data1`–`data5` signals, for a 500 ns total. At time 50 ns, the output data is still uninitialised, dependent on the previous state of the circuit, which is unknown.
- c) Signals in a process are set to their last assignment within that process; `output(7 downto 6)` is always equal `output(3 downto 2)` since both are signal assignments, both defined by the same signals (`sig1` and `sig2`), and not reassigned elsewhere within the same process.

In contrast, `outdata(1 downto 0)` and `outdata(5 downto 4)` are defined by *variables* `var1` and `var2`, whose values have been altered between assignments in the process.

- d) Now, the process is evaluated on changes to only the `indata` signal. We set this to 0 at time 0, and thus the signals `sig1` and `sig2` are assigned to `not var1` and `not indata`, respectively, *at the process end*. The process is not evaluated again until time 100 ns, when we change the `indata` signal again, consequently the state of `outdata(1 downto 0)` and `outdata(3 downto 2)` are unknown before this (because their assignments are dependent on `sig1` and `sig2`).