# Universidade de Aveiro

**Mestrado Integrado em Engenharia de Computadores e Telemática**
## Arquitectura de Computadores Avançada

# Introduction to VHDL Simulation

Academic year 2020/2021

1. The `flipFlopD.vhd` file (available at the course elearning site) contains the VHDL description of an asynchronous digital circuit which represents a 1 bit storage device and is to be simulated using the Quartus Prime Lite Edition software package.

   1.1. Open the file with a text editor, read it carefully, draw a schematics that depicts its internal organization and explain what it does.
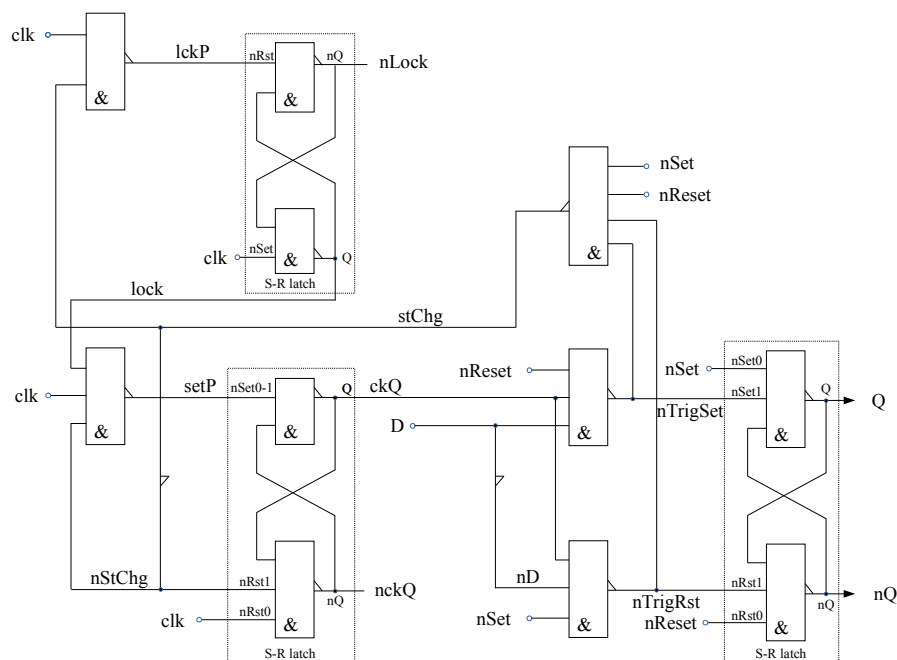


**Transition state table**

| nSet | nReset | Ck | D | lckP | lock | setP | ckQ | stChg | trigSet | trigRst | Q | nQ | Comment |
|------|--------|----|----|------|------|------|-----|-------|---------|---------|---|----|---------|
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | Q | nQ | hold state |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | transition state |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | Q | nQ | transition state |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | Q | nQ | hold state |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | Q | nQ | hold state |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | transition state |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | Q | nQ | transition state |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | Q | nQ | hold state |
| 0 | 1 | 0 | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | hold state |
| 0 | 1 | 1 | X | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | hold state |
| 1 | 0 | 0 | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | hold state |
| 1 | 0 | 1 | X | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | hold state |

The digital circuit works as a positive edge-triggered D-type flip flop with asynchronous *set* and *reset* inputs. The design is a bit more complex that should have been to allow for zero value propagation time delays of the components.

1.2. Create a Quartus project and simulate its operation.
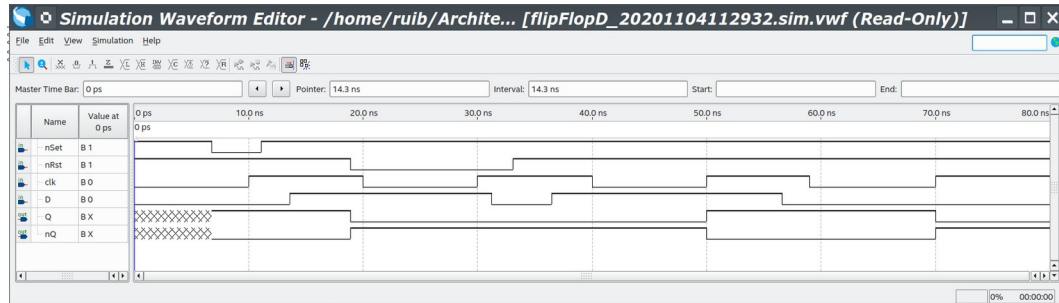


2. The `lRot_8bit.vhd` file (available at the course elearning site) contains the VHDL description of a combinatorial digital circuit which is to be simulated using the Quartus Prime Lite Edition software package.

2.1. Open the file with a text editor, read it carefully, draw a schematics that depicts its internal organization and explain what it does.
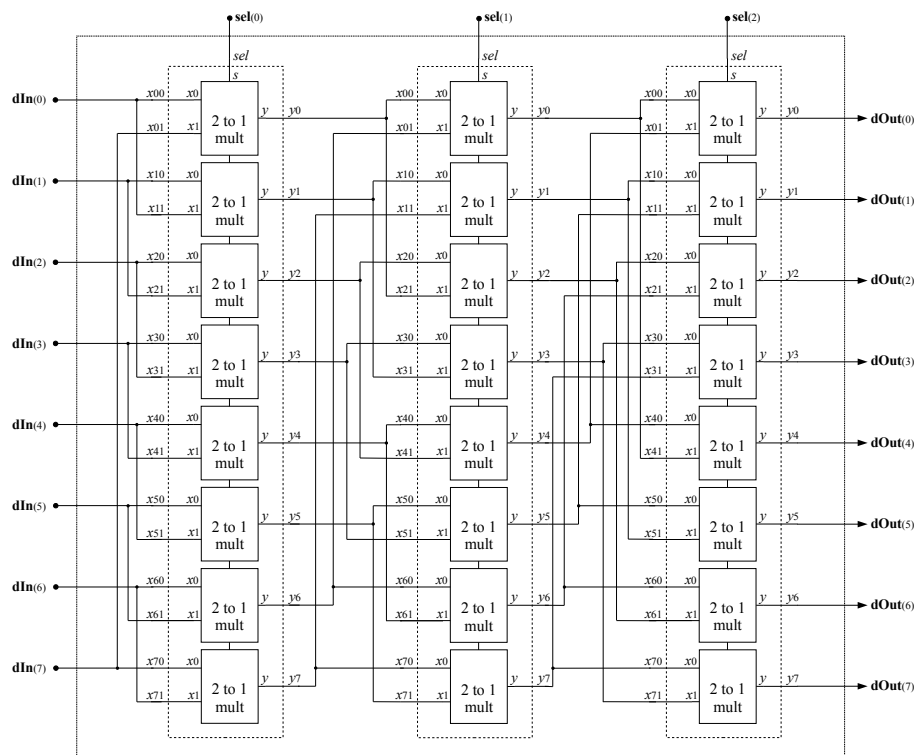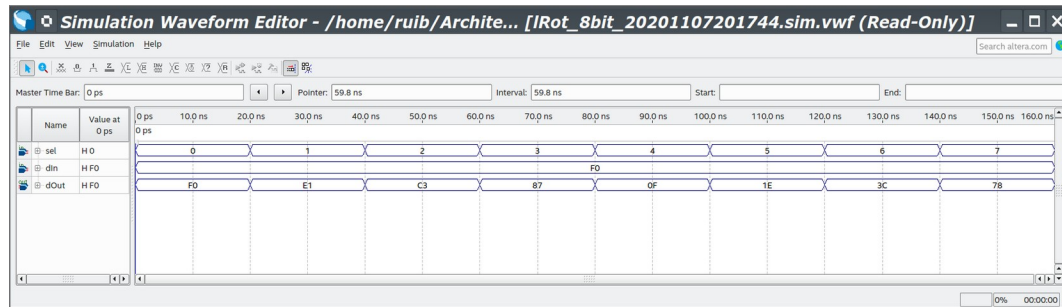


The digital circuit works as a 8-bit left barrel rotator as can be seen from the following set of equations

$$\mathrm{dOut}(i) \;=\; \mathrm{dIn}\left[\left[i + 4 \cdot \mathrm{sel}(2) + 2 \cdot \mathrm{sel}(1) + \mathrm{sel}(0)\right] \bmod 8\right] \;,$$

$$\text{with } i = 0, 1, \cdots, 7 \;.$$

2.2. Create a Quartus project and simulate its operation.



2.3. The circuit as it is described has a privileged direction of operation. Modify the design so that it can operate equally well in both directions. Start by specifying its interface, draw a schematics of its internal organization and only then write the VHDL code that describes it. Create a new Quartus project and simulate its operation.

Using the well-known fact that rotating $k$ bits to the right on a 8-bit word is the same as rotating $(8-k)$ mod 8, or $k$ *8s complement*, bits to the left on the same word, it becomes straightforward to design a 8-bit bidirectional barrel rotator. Notice that since 8 is the 3th power of 2, $k$ *8s complement* can be implemented as the *2s complement* of a 3-bit word.

**3 bit 2s complementor**



**1 bit half adder**