Laboration i digital konstruktion I: Lab 1

1.0 First design : Adderare

1.1 Purpose

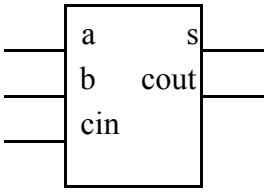
This lab-exercise shows how VHDL can be used in different ways to describe the same functionality and how the different descriptions can be verified in the same testbench.

1.2 Preparations

```
cd ~/VHDL
mkdir lab1
cd lab1
qhlib work
cp ~kurser/digkon1/lab/VHDL_kod/add8_tb.vhdl .
```

1.3 To do

1.3.1 Write a synthesizable model for an 1-bit full-adder (fa) in the text-editor *emacs*. Name the file *fa.vhdl*



The modul-interface should be like:

```
entity fa is
port ( a, b, cin : in std_logic;
       s, cout : out std_logic);
end fa;
```

The data type `std_logic` is defined in `IEEE.STD_LOGIC_1164`.

1.3.2 Simulate the full-adder

Compile the source-code (fa.vhdl)

`qvhcom fa.vhdl`

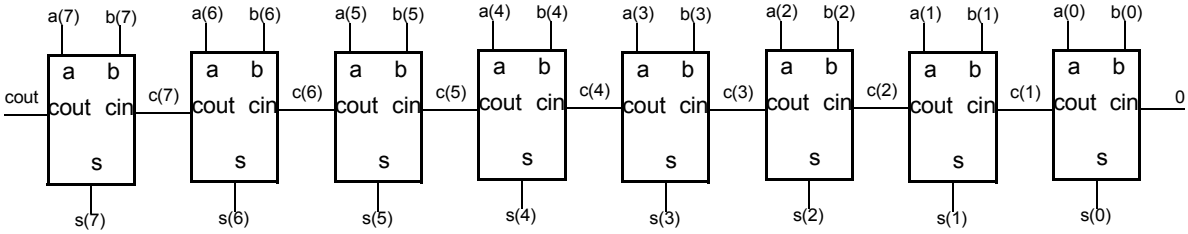
start the simulator

`qhsim`

Verify the full-adder by simulating all combinations on the inputs.

1.3.3 Structural description of an 8-bit adder

Construct a structural description of an 8-bit adder (add8) by using the previously designed full-adder. By connecting the eight 1-bit adders we build an adder capable of adding two 8-bit numbers.



1.3.4 The modul-interface should be like:

```
entity add8 is
port ( a, b : in std_logic_vector(7 downto 0);
       cin : in std_logic;
       s : out std_logic_vector(7 downto 0);
       cout : out std_logic);
end add8;
```

Since we are going to write two descriptions of *add8* where the module-interface shall be identical (i.e. the entity declaration) we put the `entity` declaration in a separate file. Name the file r *add8_entity.vhdl*. Write the structural description of add8, name the architecture *struct*, and save it in a file called *add8_struct.vhdl*.

1.3.5 Verify the structural description

Compile *add8_struct.vhdl* and read it into the simulator to test a couple of combinations of the inputs, just to do a quick test to see if it works at all.

1.3.6 Write a synthesizable behavioural description of *add8.*

Name the architecture *rtl* and save the file in *add8_rtl.vhdl*. The addition is made by the + *sign*.

1.3.7 Verify the behavioural description

Compile *add8_rtl.vhdl* and read it into the simulator to test a couple of combinations of the inputs, just to do a quick test to see if it works at all.

1.3.8 Verify that the structural description and the behavioural description are equivalent.

Open the testbench-file *add8_tb.vhdl* in *emacs*. Explain what is does.

Compile *add8_tb.vhdl* and read it into the simulator. Simulate. What happens ?

1.4 Report

Answer the questaions, include VHDL-code results from simulations, demonstrate your designs for the lab. assistant.