

VHDL course

Lab Session 1

E.M. Aboulhamid

1 Objectives

The objective of this exercise is to give some practical experience of the use of VHDL in simple modeling situations and to gain some familiarity with your favorite set of simulation tools

2 Tutorial

If you are using ModelSim environment, run the tutorial starting page 97 of the tutorial (help::EE documentation::EE Tutorial)

Examine the code you are using and try to understand it.

3 Running a behavioral model

Re-run the dataflow model of the full adder with the testbench. Examine the log of the list of signals. Understand the delta delay mechanism.

4 Development of a structural model

4.1 Development steps

- Create a folder for a project.
- Create a library work once.
- Develop and compile all your design units.
- Load and simulate your design.

4.2 Work to accomplish

The FA is an instance of a full_adder and is a component of the entity TB. The Full_adder has three instances of components: U1 and U2 are two instances of a half_adder and U3 is an instance of an OR2 gate. U1 has a delay of 3 ns, U2 has a delay of 8 ns. The behavioral descriptions of the half adder and the OR2 gate are given.

- a) Develop the structural description of the full_adder
- b) Develop a test_bench where you define 5 signals in1, in2, Cinitial, CoutF and SumF. They should receive the following waveforms
in1 <= '0' , '1' after 60 ns;
in2 <= '0' , '1' after 30 ns, '0' after 60 ns, '1' after 90 ns;
cinitial <= '0' , '1' after 15 ns, '0' after 30 ns, '1' after 45 ns, '0' after 60 ns, '1' after 75 ns, '0' after 90 ns, '1' after 105 ns;

4.3 VHDL descriptions of the components

```
entity OR2 is  
port (A:in bit; B:in bit; C: out bit);  
end OR2;
```

architecture behavioral of OR2 is

```
begin
```

```
    C<= (A or B);
```

```
end;
```

```
    generic (delay: TIME := 0 ns);
```

```
    PORT (I1:in bit;
```

```
          I2: in bit;
```

```
          carry: out bit;
```

```
          sum: out bit);
```

```
end half_adder;
```

architecture behavioral of half_adder is

```
begin
```

```
    sum <= (I1 xor I2) after delay;
```

```
    carry <= (I1 and I2) after delay;
```

```
end behavioral;
```

4.4 View of the design

TB

