HDL Week 2

Jochem Arends (495637)

Self Test Chapter 3

```
ARCHITECTURE x OF gate_network IS
BEGIN
    -- concurrent assignment statements operate in parallel
    -- D(1) selects bit 1 of standard logic vector D
    X <= A AND NOT(B OR C) AND (D(1) XOR D(2));
    -- Process must declare a sensitivity list
    -- Sensitivity list includes all signals which can change the outputs
    PROCESS (A,B,C,D)
    BEGIN
        -- Statements inside a process execute in sequential order
        Y <= A AND NOT(B OR C) AND (D(1) XOR D(2));
    END PROCESS;
END x;
```

1. *Is this a structural, behavioral, or RTL description?*

   This is a RTL description cause the circuit is described in the relationship between signals.

2. *What is the purpose of a generic?*

   The purpose of generics is to create flexible components.

3. *What is the purpose of a configuration statement?*

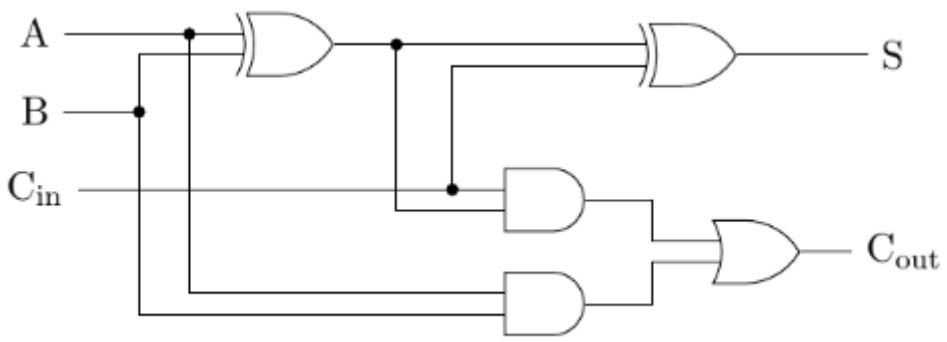
   -

4. *In our tutorial there is an adder, is this a structural, behavioral, or RTL description?*

   The adder from the tutorial is defined using a RTL description. The behaviour is described in terms of relationships between signals.

5. *Draw the schematics of a full adder and make a structural VHDL description*

   +



Self Test Chapter 4

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity adder is
port (
    a, b:  in  std_logic_vector(2 downto 0);
    sum:   out std_logic_vector(2 downto 0);
    carry: out std_logic
);
end adder;

architecture arch of adder is
variable cin: std_logic := '0';
begin
    process (a, b)
    begin
        for i in 0 to a'length loop
            -- sum for current bit
            if (a(i) = '0' and b(i) = '0' and cin = '1')
            or (a(i) = '0' and b(i) = '1' and cin = '0')
            or (a(i) = '1' and b(i) = '0' and cin = '0')
            or (a(i) = '1' and b(i) = '1' and cin = '1') then
                sum(i) <= '1';
            else
                sum(i) <= '0';
            end if;

            -- carry for next
            if a(i) = b(i) then
                cin := '1';
            else
                cin := '0';
            end if;
        end loop;

        -- the final carry
        carry <= cin;
    end process;
end arch;
```