Informatics and Mathematical Modelling
Technical University of Denmark

02207 : Advanced Digital Design Techniques

Lab 1: Exercise on Synthesis

Group *dt07*

Markku Eerola (s053739)

Rajesh Bachani (s061332)

Josep Renard (s071158)

October 16, 2007

Contents

1 Purpose of the Exercise **2**

2 Behavioral Description for 24-bit Adder **2**

3 Behavioral Description for 24-bit Register **2**

4 Top-Level Netlist from 24-bit Adder and 24-bit Register **2**

5 Synthesis Results **2**

1 Purpose of the Exercise

The goal of the exercise was to get familiar with the process of synthesis of digital circuits, using special tools for synthesis such as Design Vision. A register-level netlist containing a 24-bit adder is synthesized in the exercise, for different values of the clock time period. The reports concerning the timing constraints, area, power consumption etc. are documented in the report.

2 Behavioral Description for 24-bit Adder

The VHDL code for a simple 24-bit adder is provided below.

```
library IEEE;
use IEEE.std_logic_1164.all, IEEE.numeric_std.all;

entity NBitAdder is
    port (A, B: in std_logic_vector(23 downto 0);
        Cin: in std_logic;
        Sum: out std_logic_vector(23 downto 0);
        Cout: out std_logic);
end entity NBitAdder;

architecture unsgned of NBitAdder is
    signal result: unsigned(24 downto 0);
    signal carry: unsigned(24 downto 0);
    constant zeros: unsigned(23 downto 0) := (others => '0');

begin
    carry <= (zeros & Cin);
    result <= ('0' & unsigned(A)) + ('0' & unsigned(B)) + carry;
    Sum <= std_logic_vector(result(23 downto 0));
    Cout <= result(24);
end architecture unsgned;
```

The behavior of this adder is verified in Modelsim.

3 Behavioral Description for 24-bit Register

4 Top-Level Netlist from 24-bit Adder and 24-bit Register

5 Synthesis Results