FPGA Programing Tutorial
Lab II

1 Sevent-segment-display

A Sevent-segment-display is made upon sevent different segments each of which contains a diode. Each of the segments can be switched on or off independant from the others. The generation of number between 0 and 9 is done by switching on some of the segments as shown in Figure 1.

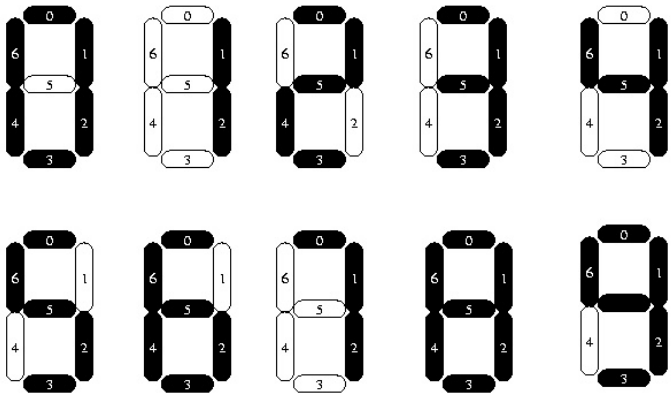

Figure 1: Dezimal number representation using a seven-segment display

The control of the sevent segment display should be specified in VHDL. The controller reacts to its inputs, which consists of a number between 0 and 9 and switch the corresponding segments on.

Preparation

1. Complete the following VHDL code of the the sevent-segment-decoder
   A switching on respectively off if a signal SSG_x ist set to 0 respectively 1 (negative Logique).

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
```

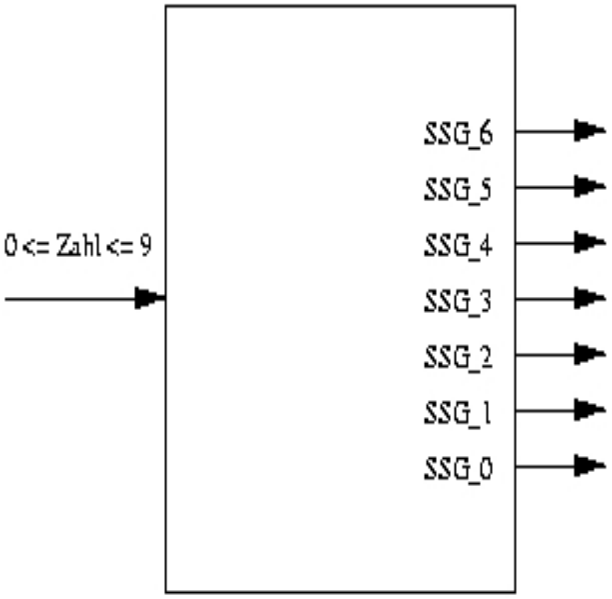

Figure 2: Controller for the sevent-segment-display

```
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity sevensegment is
  port (NUMBER: in STD_LOGIC_VECTOR (3 downto 0);
        HEX0: out STD_LOGIC_VECTOR (6 downto 0) );
end sevensegment;

architecture Behavioral of sevensegment is
  SIGNAL DIGIT_in   : STD_LOGIC_VECTOR (6 DOWNTO 0);

begin
  process(NUMBER)
-- |-0-|
-- 5   1
-- |-6-|
-- 4   2
-- |-3-|

  begin
    case NUMBER is    --  6543210
```

```
        when "0000" => DIGIT_in <= "0111111";
        ....
        ....
      end case;
    end process;
    HEX0 <= not ..... ;

end Behavioral;
```

Lab

2. Start a project in the **ISE** environment and Simulate your implementation.
3. Implement the seven-segment display for one digit (digit 0 on the Nexys 3). Test your implementation by downloading your code on the board, providing the inputs and observing the outputs.

   Assign the pins to your design as described in the Nexys 3 user manual

2 Two-numbers-display

We will now like to display tow digit-numbers. The controller for this will be implemented in a module **Display_2pos** shown in Figure 3. The module **Display_2pos** contains two instances of the entity (**sevensegment**) previously descrived. An additional module **two_number_split** is needed to split a number between 0 and 99 in two digits to be displayed on the two sevent-segment-display.

Preparation

1. Complete the following VHDL-Code for the modul **two_number_split** of Figure 3.

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY two_number_split is
   PORT ( number: in integer range 0 to 99;
          position1, position0: out STD_LOGIC_VECTOR (3 downto 0));
```

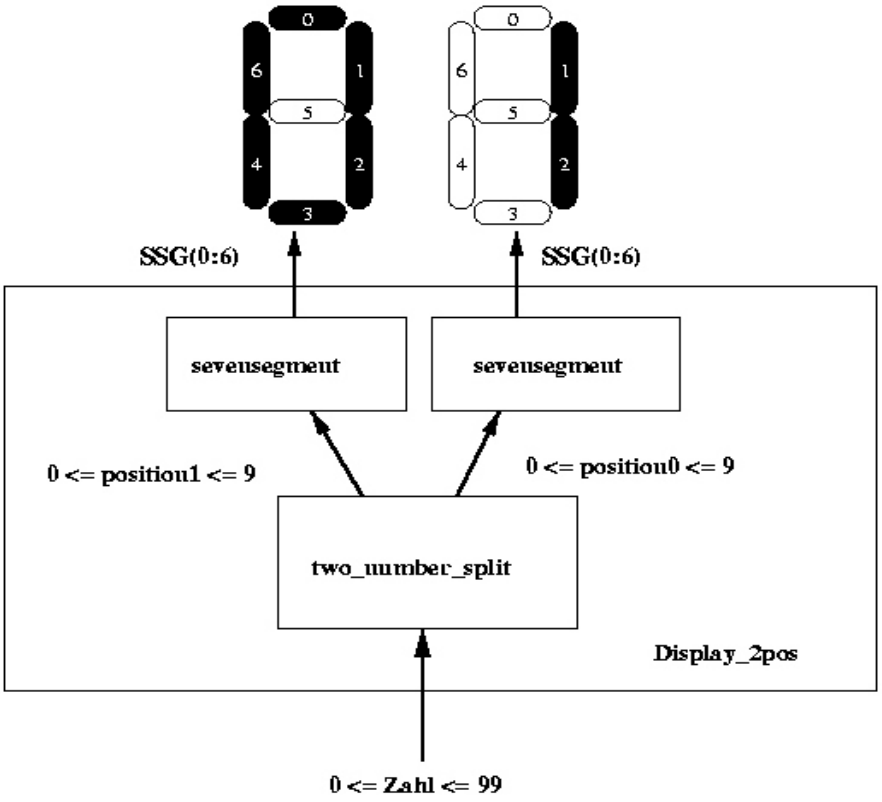

Figure 3: Decimal number display on 2 sevent-segment-display

```
END two_number_split;

ARCHITECTURE Behavioral of two_number_split IS

SIGNAL position1_temp, position0_temp: in STD_LOGIC_VECTOR (3

BEGIN

  CONVERT: process(number,  position0_temp, position1_temp)
  BEGIN
     .......
     .......
     .......

     position0 <= position0_temp;
```

```
     position1 <= position1_temp;

  end process CONVERT;
end Behavioral;
```

2. Complete the following VHDL-Code for the module Display 2pos of figure 3.

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY Display_2pos is
   PORT ( number: in integer range 0 to 99;
          digit1,digit0 : OUT STD_LOGIC_VECTOR (6 downto 0)
END Display_2pos;

ARCHITECTURE Structural of Display_2pos IS
COMPONENT sevensegment IS
   PORT (number: IN  STD_LOGIC_VECTOR (3 downto 0);
         digit : OUT STD_LOGIC_VECTOR (6 downto 0)
 );
END COMPONENT;

...

SIGNAL position1_in, position0_in: in STD_LOGIC_VECTOR (3 downto 0);
SIGNAL number_in: INTEGER RANGE 0 TO 99;

BEGIN
    SSG1: sevensegment

    ...
end Structural;
```

Lab

simulate and implement you design on the Nexys 3 board.