Mid Sweden University

Exam – ET061G– March 16 2015

Digital Electronics with VHDL

Time: five hours
Allowed aids: None
Number of tasks: 9
Number of pages: 5
Maximum points: 100 (50 are required to pass)

David Krapohl, tel: 060-148755, email: david.krapohl@miun.se

Instructions for submitted solutions:

- Rationale and justifications may not be so scarce that they become difficult to follow.
- The reasoning behind used equations should be explained.
- The calculations shall be sufficiently complete to show how the final result was obtained.
- Each task must be concluded with a clearly written answer.

Tasks

Exercise 1. 8 P.

Covert the following binary number to hexadecimal and ocatal form:
11001001001

Exercise 2. 8 P.

Convert the decimal number -51 to an 8-bit binary number in two's complement form.

Exercise 3. 10 P.

Minimize the following Boolean expression: $a \cdot \bar{b} + \overline{\overline{a \cdot \bar{c}} \cdot d} + a \cdot c + d \cdot c$

Exercise 4. 10 P.

Determine if the following equality is true: $x \cdot \overline{y} \cdot z + \overline{x} \cdot \overline{z} = (\overline{x} + \overline{y})(\overline{x} + z)(x + \overline{z})$

Exercise 5. 10 P.

Minimize the function $f(a, b, c, d) = \sum(0, 5, 7, 10, 13, 15) + d(2, 8)$ by using a Karnaugh-diagram. Write the result as a minimal Sum-of-Product (SoP) expression.

Exercise 6. 12 P.

Design a function that takes the binary numbers $X = (x_1, x_0)$ and $Y = (y_1, y_0)$ as inputs and compares the numbers. The output $Z = (x_2, x_1, x_0)$ should show if X is smaller ($z_2 = 1$), equal to ($z_1 = 1$) or larger than ($z_0 = 1$) Y. Give your answer as Boolean expressions for the output bits.

Exercise 7. 12 P.

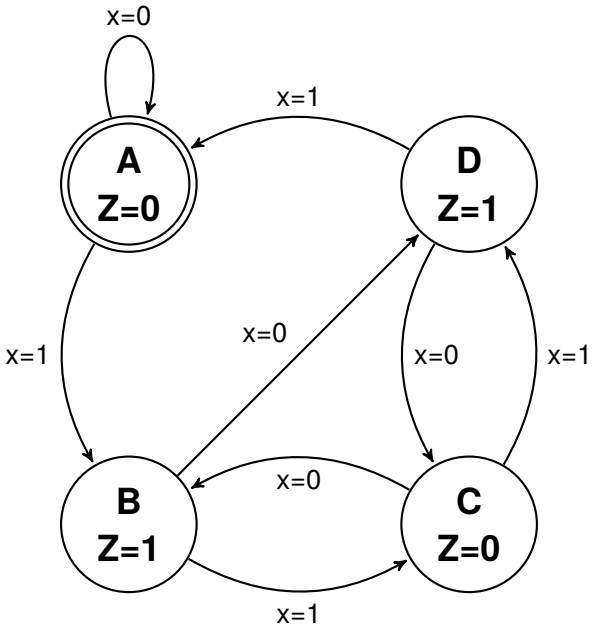
Write VHDL-code for the function described in task 6.

Exercise 8. 14 P.

Draw the state transition graph for the VHDL code given in Appendix 1. Is this a Moore or a Mealy machine? Explain your reasoning.

Exercise 9. 16 P.

Develop a circuit diagram for the state transition graph shown below. Use gray code and T-flip-flops for the state memory.


x=0
x=1
A
Z=0
D
Z=1
x=1
x=0
x=0
x=1
x=0
B
Z=1
C
Z=0
x=1

Appendix

```
\begin{verbatim}
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity seqdet is
    port (clk: in std_logic;
          clr: in std_logic;
          din: in std_logic;
          dout: out std_logic);
end seqdet;

architecture seqdet of seqdet is

type state_type is (s0,s1,s2,s3,s4);
signal present_state, next_state: state_type;

begin
    sreg: process(clk, clr)
    begin
        if clr = '1' then
            present_state <= '0';
        elsif clk'event and clk = '1' then
            present_state <= next_state;
        end if;
    end process;

    C1: process(present_state, din)
    begin
        case present_state is
            when s0 =>
                if din = '1' then
                    next_state <= s1
                else
                    next_state <= s0;
                end if;
            when s1 =>
                if din = '1' then
                    next_state <= s2
                else
                    next_state <= s0;
                end if;
```

```
            when s2 =>
                if din = '1' then
                    next_state <= s3
                else
                    next_state <= s2;
                end if;
            when s3 =>
                if din = '1' then
                    next_state <= s1
                else
                    next_state <= s0;
                end if;
            when others =>
                null;
        end case;
    end process;

    seq2: process(clk, clr)
    begin
        if clr = '1' then
            dout <= '0';
        elsif clk'event and clk='1' then
            if present_state = s3 and din = '1' then
                dout <= '1';
            else
                dout <= '0';
            end if;
        end if;
    end process;
end seqdet;
\end{verbatim}
```