Mittuniversitetet
MID SWEDEN UNIVERSITY

Försättsblad Prov Original

Kurskod	Provkod	Tentamensdatum
ET061G	T102	2018-06-14

Kursnamn	Elektroteknik GR (A), Digitalteknik med VHDL
Provnamn	Skriftlig tentamen
Ort	Sundsvall
Termin	V18
Ämne	Elektroteknik

Arkiv

MID SWEDEN UNIVERSITY, SWEDEN

Examination – ET061G– 14 June 2018

Digital Electronics with VHDL

Time:	**Five hours**
Permitted help material:	**None**
Total number of questions:	**4**
Total number of pages sidor:	**4**
Max point:	**55 (22 points are required to pass)**

Najeem Lawal, Tel: 010-142 8561, E-post: najeem.lawal@miun.se

Instructions:

- Please provide the solutions, explanations and arguements in English.
- Please submit a question per sheet.
- Reasons and motivations must not be so few that it will be difficult to follow.
- The thinking behind equations and axioms set must be explained.
- The deductions must be complete to clearly show how results has been obtained.
- Each solution must be completed with a clearly marked answer.

Useful axioms for Boolean algebra:

1. $A + 0 = A$	5. $A + A = A$	9. $\overline{\overline{A}} = A$
2. $A + 1 = 1$	6. $A + \overline{A} = 1$	10. $A + AB = A$
3. $A \cdot 0 = 0$	7. $A \cdot A = A$	11. $A + \overline{A}B = A + B$
4. $A \cdot 1 = A$	8. $A \cdot \overline{A} = 0$	12. $(A + B)(A + C) = A + BC$

Questions

Questions 1. 10 P.

a) Convert **11010101** from binary to decimal number (2 P.).

b) Multiply the binary number in **1a)** by 4 by using the left-shift operator and write the result as a decimal number (2 P.).

c) Divide the binary number in **1a)** by 2 by using the right-shift operator and write the result as a decimal number (2 P.)

d) Convert the hexadecimal number **DAD** into binary. Then convert the obtained binary number into octal (2 P.)

e) Calculate **-6+5** by using 2's complement (2 P.)

Questions 2. 15 P.

a) Determine if the following equality is true using Boolean Algebra (5 P.):

$$a \cdot b + \overline{a} \cdot c = a \cdot b + b \cdot c + \overline{a} \cdot c$$

b) Minimize the following Boolean expression (5 P.):

$$(x + y \cdot z)(x + \overline{y} + z)$$

c) Minimize the following function by using a Karnaugh-diagram.

$$f(x, y, z, w) = \sum(0, 2, 5, 7, 8, 10, 13, 15)$$

Write the result as a minimal Sum-of-Products (SoP) expression and draw the circuit diagram for the minimized expression (5 P.).

Questions 3. 15 P.

a) If the following VHDL code snippet is correct and free from latches, what is the data type of signal **sel**? (2 P.)

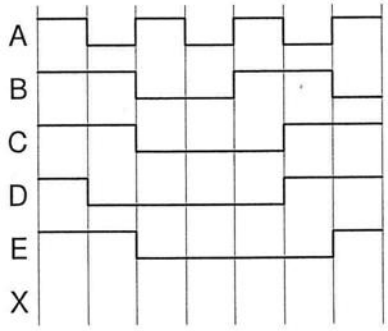
```
case sel is
    when '0' => z <= a;
    when '1' => z <= b;
end case;
```

Re-write the code snippet using **if-then** statements and datatype **std_logic** for signal **sel**. Ensure there are no avoid latches (3 P.)

b) Write a function in VHDL that takes a 5-bit input vector (1 bit for odd parity + 4 bit data), and then verifies if the parity bit matches the data or not (i.e. checks if the data has been correctly transmitted). (5 P.)

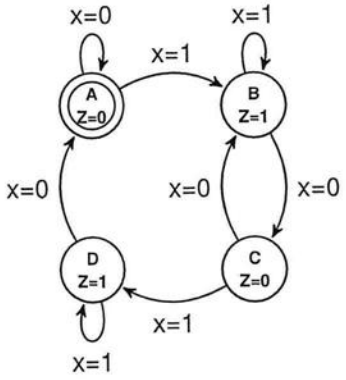
c) Draw the circuit diagram for the expression below and complete the signal for the output X. Point out how each member of the expression contributes to the X-waveform (5 P.).

$$X = AC + \overline{A}B + ED + \overline{C} + \overline{E}$$



Questions 4. 15 P.

a) Write a complete synthesizable VHDL code to implement the state machine in the diagram below. The code must have three different processes. The processes should include combinatorial process for state transistion, clocked process for state transfer and process for output estimation. Expected input signal are **clk, reset** and **x** whereas the output is **Z**. Use Appendix 1 as inspiration. (6 P.)



b) Develop Boolesk expressions for the state transition graph in Question 4a. Use gray coding and T-flipflops (6 P.).

c) Draw the state machine diagram for the VHDL code in Appendix 1. Is this a Moore or Mealy state machine? Motivate your answer (3 P.)

Appendix 1

```
library ieee;
use ieee.std_logic_1164.all;
ENTITY q4c IS
PORT (
      clk, reset, in1     : IN  STD_LOGIC;
      out1                : OUT STD_LOGIC_VECTOR(1 DOWNTO 0) );
END q4c;
ARCHITECTURE fsm OF q4c IS
    TYPE  state_type IS (s0, s1, s2, s3);  -- state declaration
    SIGNAL current_state, next_state : state_type;
BEGIN
    p0 : PROCESS(current_state, in1)  --state transistion combinational process
    BEGIN
        CASE current_state IS
            WHEN s0 => IF in1 ='1' THEN
                           next_state<=s1;
                       END IF;
            WHEN s1 => IF in1 ='0' THEN
                           next_state<=s3;
                       END IF;
            WHEN s2 => IF in1 ='0' THEN
                           next_state<=s0;
                       END IF;
            WHEN s3 => IF in1 ='1' THEN
                           next_state<=s2;
                       END IF;
            WHEN OTHERS => null;
        END CASE;
    END PROCESS p0;

    p1 : PROCESS(clk, reset)  --state transfer clocked process
    BEGIN
        IF reset ='1' THEN
            current_state <= s0;
        ELSIF clk'EVENT AND clk='1' THEN
            current_state <= next_state;
        END IF;
    END PROCESS p1;

    p3 : PROCESS (current_state)   --output estimation combinational process
    BEGIN
        CASE current_state IS
            WHEN s0 => out1 <= "00";
            WHEN s1 => out1 <= "01";
            WHEN s2 => out1 <= "11";
            WHEN s3 => out1 <= "10";
            WHEN OTHERS => null;
        END CASE;
    END PROCESS p3;
END fsm;
```