

# MITTUNIVERSITETET

*Tentamen – ET061G– 16 Mars 2015*

## **Digitalteknik med VHDL**

Tid: fem timmar

Tillåtna hjälpmedel: inga

Antal uppgifter: 9

Antal sidor: 5

Maxpoäng: 100 (50 poäng krävs för godkänt)

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

Anvisningar för inlämnade lösningar:

- Resonemang och motiveringar får ej vara så knapphändiga att de blir svåra att följa.
- Tankegången bakom uppställda ekvationer skall förklaras.
- Uträkningarna skall vara tillräckligt fullständiga för att visa hur slutresultatet har erhållits.
- Varje problemlösning skall avslutas med ett klart formulerat svar.

# Uppgifter

## Uppgift 1.

8 Poäng

Konvertera följande binärtal till hexadecimal och oktal form:

11001001001

## Uppgift 2.

8 Poäng

Konvertera det decimala talet -51 till ett 8-bitars binärt tall på 2-komplementform.

## Uppgift 3.

10 Poäng

Minimera följande Boolesk uttryck:  $a \cdot \bar{b} + \overline{a \cdot \bar{c} \cdot d} + a \cdot c + d \cdot c$

## Uppgift 4.

10 Poäng

Avgör om följande likhet gäller:  $x \cdot \bar{y} \cdot z + \bar{x} \cdot \bar{z} = (\bar{x} + \bar{y})(\bar{x} + z)(x + \bar{z})$

## Uppgift 5.

10 Poäng

Minimera funktionen  $f(a, b, c, d) = \sum(0, 5, 7, 10, 13, 15) + d(2, 8)$  genom att använda ett Karnaugh-diagram. Skriv resultatet som ett minimalt Summa-av-Produkt (SoP) uttryck.

## Uppgift 6.

12 Poäng

Designa en funktion som tar de binära talen  $X = (x_1, x_0)$  och  $Y = (y_1, y_0)$  som inputs och jämför talen. Utgången  $Z = (z_2, z_1, z_0)$  ska visa om X är mindre ( $z_2 = 1$ ), lika med ( $z_1 = 1$ ) eller större än ( $z_0 = 1$ ) Y. Svara med boolesk uttryck för outputbitarna.

## Uppgift 7.

12 Poäng

Skriv VHDL-kod för funktionen beskrivit i uppgift 6.

## Uppgift 8.

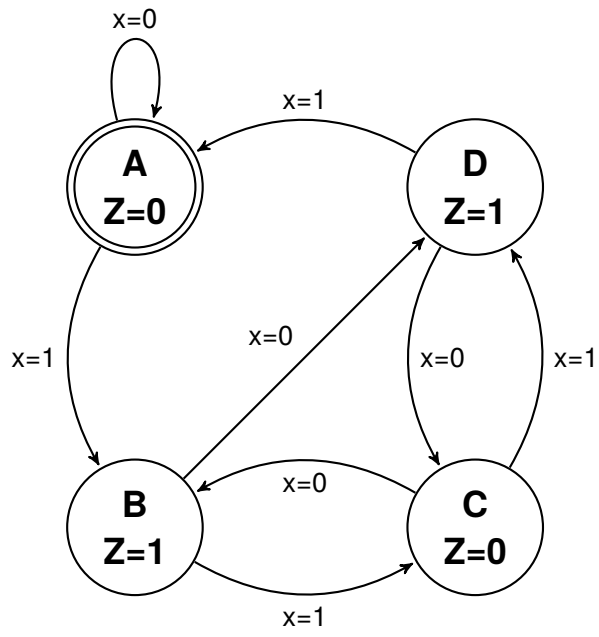
14 Poäng

Rita tillståndsdigrammet för VHDL-koden i Appendix 1. Är detta en Moore- eller en Mealy-maskin? Motivera ditt svar.

## Uppgift 9.

16 Poäng

Utveckla ett kretsschema för tillståndsdigrammet nedan. Använd graykodning och T-vippor för tillståndsdigrammet.



state diagram

## Solution

### Solution 1.

$$\text{IOCT} \quad \underline{0..11} \underline{001} \underline{001} \underline{001} \quad (1)$$

3      1      1      1

$$\text{HEX} \quad \underline{0..110} \underline{0100} \underline{1001} \quad (2)$$

6      4      9

### Solution 2.

11001101

### Solution 3.

solve with deMorgan, then complements, then distributive

$$a + cd + \bar{d} = a + (\bar{d} + c)(\bar{d} + d)$$

$$a + c + \bar{d}$$

### Solution 4.

true! DNF vs CNS from (WA: boolean function 37)

$$x \cdot \bar{y} \cdot z + \bar{x} \cdot \bar{z} = (\bar{x} + \bar{y})(\bar{x} + z)(x + \bar{z})$$

### Solution 5.

		cd			
		00	01	11	10
ab	00	1	0	0	X
	01	0	1	1	0
	11	0	1	1	0
	10	X	0	0	1

$$f = \bar{b} \cdot \bar{d} + b \cdot d$$

**Solution 6.**

Truth table:

no	x1	x2	y1	y0	z2	z1	z0
0	0	0	0	0	0	1	0
1	0	0	0	1	1	0	0
2	0	0	1	0	1	0	0
3	0	0	1	1	1	0	0
4	0	1	0	0	0	0	1
5	0	1	0	1	0	1	0
6	0	1	1	0	1	0	0
7	0	1	1	1	1	0	0
8	1	0	0	0	0	0	1
9	1	0	0	1	0	0	1
10	1	0	1	0	0	1	0
11	1	0	1	1	1	0	0
12	1	1	0	0	0	0	1
13	1	1	0	1	0	0	1
14	1	1	1	0	0	0	1
15	1	1	1	1	0	1	0

X smaller than Y

$$z_2 = \bar{a}\bar{b}d + \bar{a}c + \bar{b}cd$$

ab \ cd	cd			
	00	01	11	10
00		1	1	1
01			1	1
11				
10			1	

X equal Y

$$z_1 = \bar{a}\bar{b}\bar{c}\bar{c} + \bar{a}b\bar{c}d + abcd + a\bar{b}c\bar{d}$$

ab \ cd	cd			
	00	01	11	10
00	1			
01		1		
11			1	
10				1

X larger than Y

$$z_0 = \bar{b}\bar{c}\bar{d} + ab\bar{d} + a\bar{c}$$

		cd			
		00	01	11	10
ab	00				
	01	1			
	11	1	1		1
	10	1	1		

**Solution 7.**

```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity comp is
    port( x in : std_logic_vector (2 downto 0);
          y in : std_logic_vector (2 downto 0);
          z out: std_logic_vector (3 downto 0));
end comp;

architecture comp of comp is
    begin

        process (x,y)
            begin

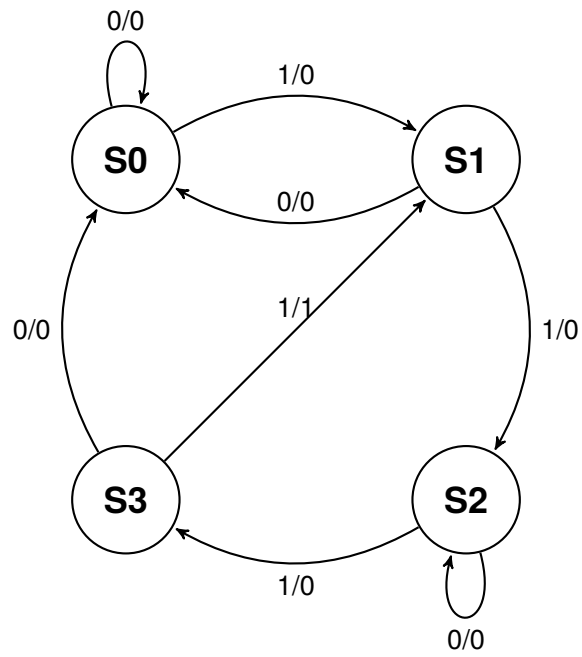
                if x < y then
                    z <= "100";
                elsif x = y then
                    z <= "010";
                else
                    z <= "001";
                end if;
            end process;
        end architecture comp;

```

Alternative: boolean expression from 6. Other alternative: case select when "x& y"...

**Solution 8.**

This is a mealy machine because state and output is set at the same time (in different processes, synchronized by clk).



**Solution 9.**

S	X			Z
	0	1		
A	A	B		0
B	D	C		1
C	B	D		0
D	C	A		1

S	gray
A	00
B	01
C	11
D	10

S	X			Z
	0	1		
00	00	01		0
01	10	11		1
11	01	10		0
10	11	00		1
Q+				

q	q+	T
0	0	0
0	1	1
1	1	0
1	0	1

S	E			Z
	0	1		
00	00	01		0
01	11	10		1
11	10	01		0
10	01	10		1
T				

t0:

E	q <sup>1</sup> ,q <sup>0</sup>			
	00	01	11	10
0		1		1
1	1		1	

t1:

		q <sup>1</sup> ,q <sup>0</sup>			
		00	01	11	10
E	0		1	1	
	1		1		1

z:

		q <sup>1</sup> ,q <sup>2</sup>			
		00	01	11	10
	0		1		
	1				1

## Appendix

```

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;

```