Exam 15 January 15.1.18, questions and answers

Logic Systems And Processors (České Vysoké Učení Technické v Praze)

Surname and first name:.....

Exam 15.1.2018- write here only your answers

1. Rewrite the logical expression F so that not operators were only before variables - check your solution in K-map!

F = not ((A or not B or not C) or ((A or B) and not (A or not C)))

F =.....

F= B / A / C

4

2. Inputs A, B, C have values shown in the figure in times t_0, t_1, t_2, t_3 . Write values of X and Y outputs. Assume that the intervals between changes in the inputs are so long so we can neglect delays of gates.

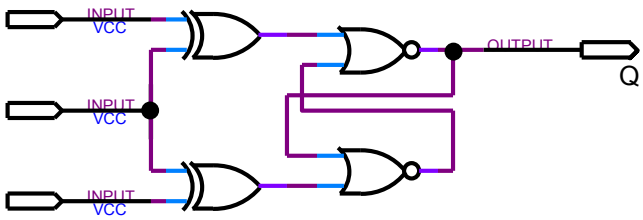
A = ..0..|..1..|..1..|..1..|

t_0..|..t_1..|.. t_2..|..t_3..|..

B = ..0..|..0..|..0..|..1..|

t_0..|..t_1..|.. t_2..|..t_3..|..

C = ..1..|..1..|..0..|..0..|



Q = ...|.....|.....|.....|

t_0..|..t_1..|..t_2..|..t_3..|..

4

3. **Function Q=f(A,B,C, Q) from question 2, decompose into** X= (not X and f0(A, B,C)) or (X and f1(A,B,C)) with the aid of **Shannon's expansion. Write f0 and f1 functions as Karnaugh maps:**

f0/5

f0= B / A / C

f1/5

f1= B / A / C

4. What decimal value is 10-bit number 10 0000 1111 if we interpret it as an integer

a) unsigned ..... b) signed in two's-complement..... ...

2

5. Mark all logic functions that have another equivalent logic function here *:*

f1<=(A xor C) or (A and not C);
f2<=(B or C) and (not A or B or C);
f3<=((C and not B) or (B and A));
f4<=(A or C) and (not A or not C);
f5<=(A and not B) xor (A and C);
f6<=(A and not C) or (C and not A);

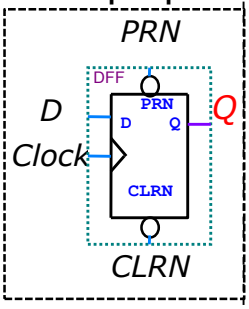
f1 ☐
f2 ☐
f3 ☐
f4 ☐
f5 ☐
f6 ☐

6

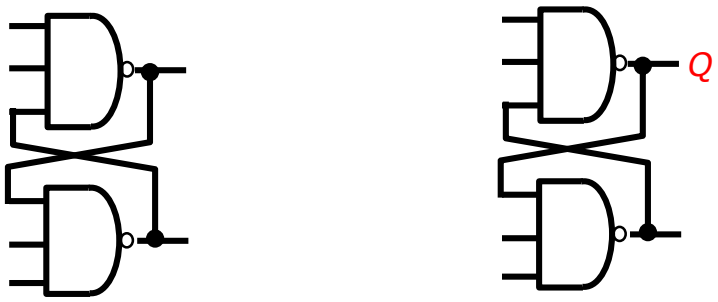
Part 1
26

Zde nepište

6. By adding gates (4 NAND and 3 NOT) and wires, create internal schematic of master-slave flip-flop DFF.



12



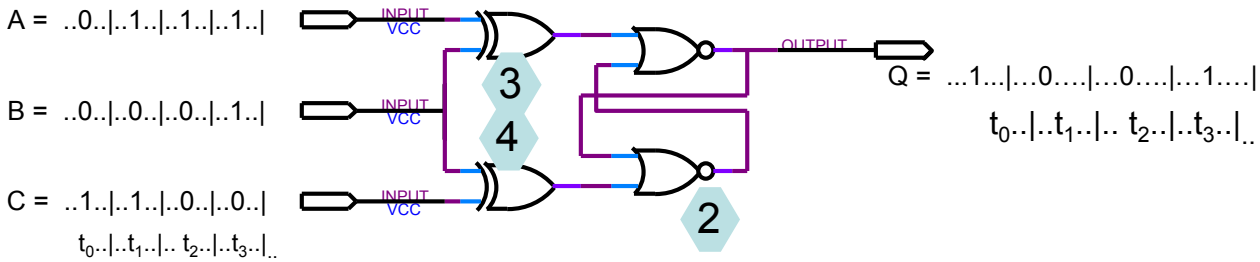
*7. An editor **ask you if you could analyze wrong formatted VHDL code:***

12

```
library IEEE; use IEEE.STD_LOGIC_1164.all;
entity test20140214 is port (a, b, c, d : in std_logic; e : out std_logic); end;
architecture rtl of test20140214 is begin
  process(a, b) variable z:std_logic_vector(0 to 3); begin
  if b = '0' then z:=(others=>'0'); elsif rising_edge(a) then
  if c='1' then z:=d & & z(0 to 2); else z:=z(3) & z(0 to 2); end if; end if; e<=z(3); end process; end rtl;
```

Draw a logical circuit diagram corresponding to this VHDL code and give it appropriate title that describes its function. Guide: First, rewrite this program in the correct formatting.

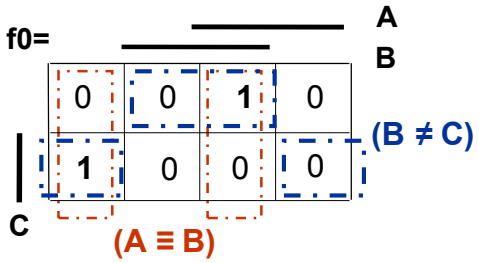
2. part
24

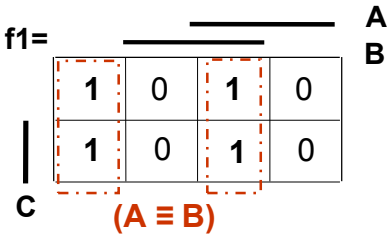


f0 := f(A,B,C,'0') := (A ≡ B) . ('0' + (B ≠ C)) := (A ≡ B) . ('0' + (B ≠ C)):= **(A ≡ B) . (B ≠ C)**

f1 := f(A,B,C,'1') := (A ≡ B) . ('1' + (B ≠ C)) := (A ≡ B) . '1' := **(A ≡ B)**

*Obě nalezené funkce f0:=***(A ≡ B) . (B ≠ C)** a *f1:=***(A ≡ B)** *zapíšeme jako Karnaughovy mapy*





5. Mark all logic functions that have another equivalent logic function here *:*

f1<=(A xor C) or (A and not C);
f2<=(B or C) and (not A or B or C);
f3<=((C and not B) or (B and A));
f4<=(A or C) and (not A or not C);
f5<=(A and not B) xor (A and C);
f6<=(A and not C) or (C and not A);

f1 ■
f2 □
f3 □
f4 ■
f5 □
f6 ■