Exam 30 January 30.1.15, questions and answers

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

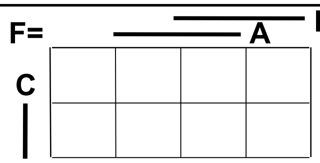
Surname and first name:...

Exam 30.1.2015 - write here only your answers

Zde nepište

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 =..

3

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.

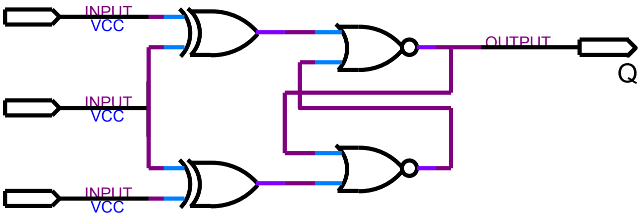
A = ..0..|..1..|..1..|..1..|

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..|

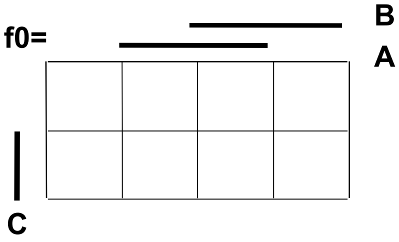


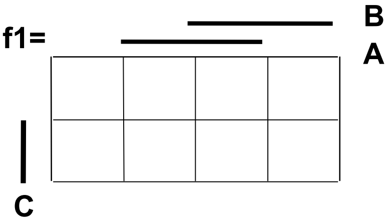
Q = ...|.......|.......|.......|

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

5

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:**





f0/3

f1/3

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); f1 □

f2<=(B or C) and (not A or B or C); f2 □

f3<=((C and not B) or (B and A)); f3 □

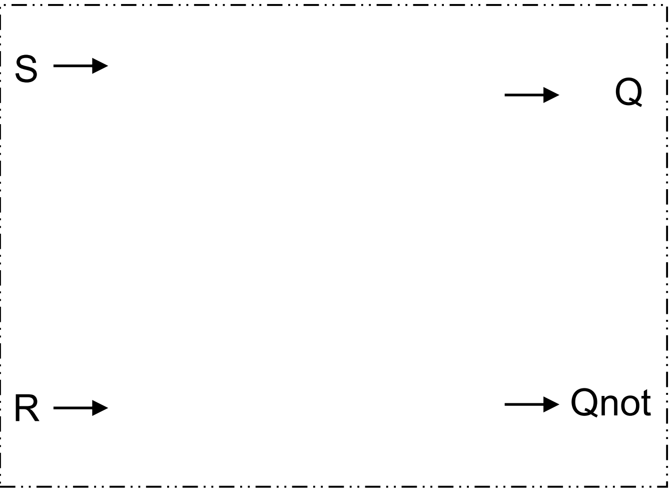
f4<=(A or C) and (not A or not C); f4 □

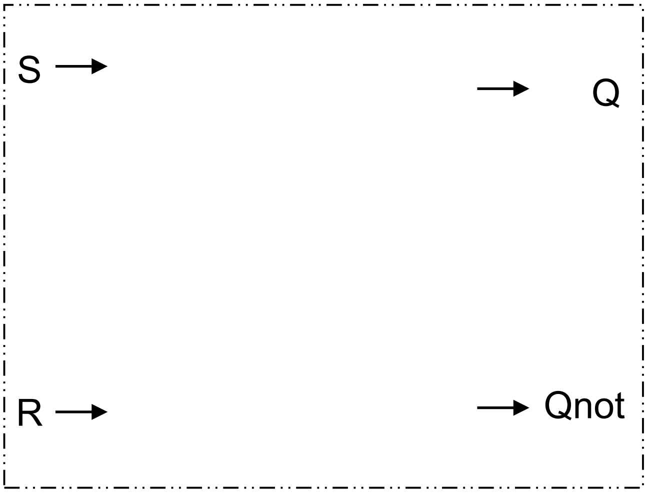
f5<=(A and not B) xor (A and C); f5 □

f6<=(A and not C) or (C and not A); f6 □

4

5. Draw R-S latch only from NOR gates and the same only from NAND gates





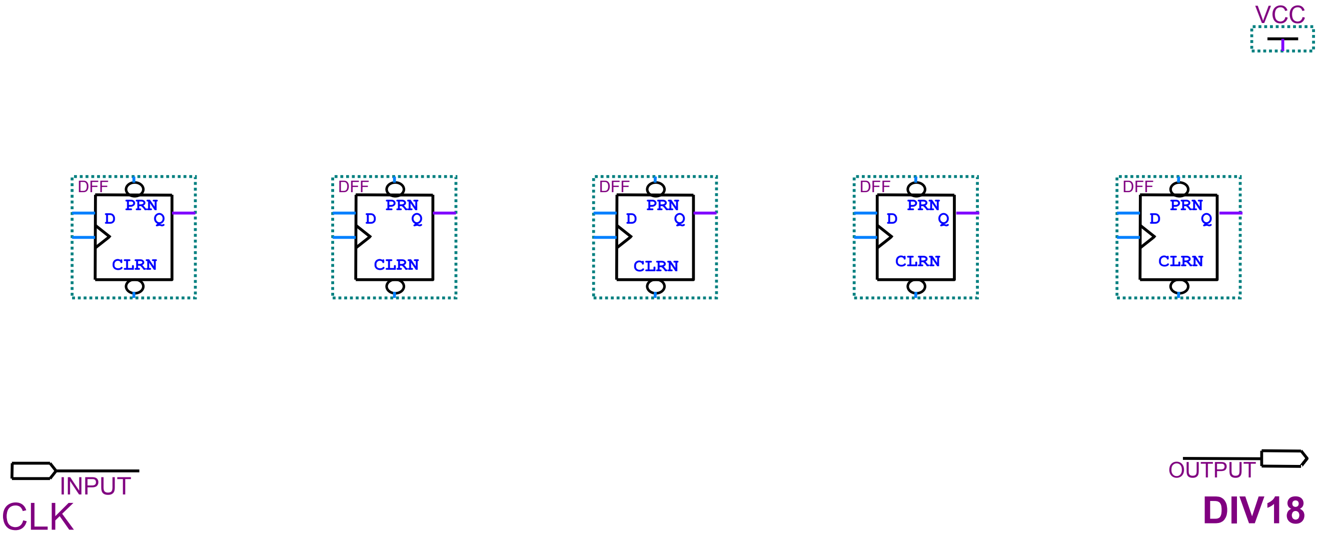
3/Nand

3/nor

Part 1
26

Zde nepište

7. Add gates and wires to the unfinished diagram below, you do not need to use everything in picture, to create a asynchronous divider by 18 of CLK clock signal with asynchronous reset ACLRN.



12

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

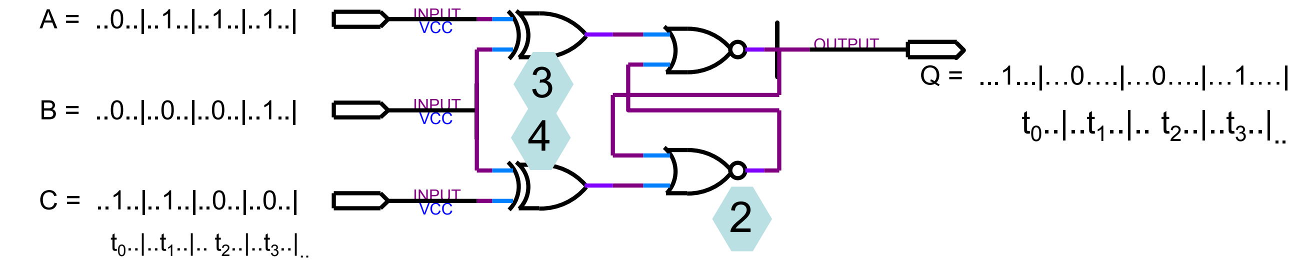
```
library IEEE; use IEEE.STD_LOGIC_1164.all;
entity test20140214 is port (a, b, c, d : in std_logic; e : out std_logic); end;
architecture rtl of test20140214 is begin
   process(a, b) variable z:std_logic_vector(0 to 3); begin
   if b = '0' then z:=(others=>'0'); elsif rising_edge(a) then
   if c='1'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

12

2. part
24

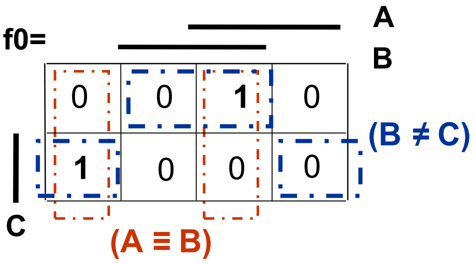
Zde nepište


A = ..0..|..1..|..1..|..1..|
B = ..0..|..0..|..0..|..1..|
C = ..1..|..1..|..0..|..0..|
t0..|..t1..|.. t2..|..t3..|..
INPUT
VCC
INPUT
VCC
INPUT
VCC
3
4
2
OUTPUT
Q = ...1...|...0....|...0....|...1....|
t0..|..t1..|.. t2..|..t3..|..

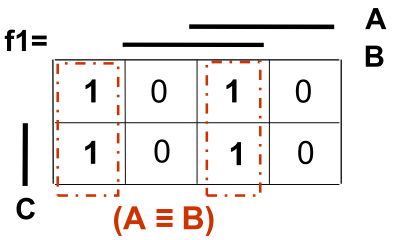
f0 := f(A,B,C,'0') := (A ≡ B) . ('0' + (B ≠ C)) := (A ≡ B) . ('0' + (B ≠ C)):= **(A ≡ B)** **.** **(B ≠ C)**

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 Karnoughovy mapy*



0	0	**1**	0
1	0	0	0



1	0	**1**	0
1	0	**1**	0

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 ■