1. Sample of written test with the solutions:

1. Rewrite logic function in VHDL syntax by the way that final function will contain only operations "or", "and" and "not" and "not" operation will never be in front of any round bracket.

F1<= A xor (B or C);

F1<=.....

F2<= **not** (A and (B **xor** C));

F2<=.....

2. Inputs A, B, C have values shown in the figure in times t_0, t_1, t_2, t_3, t_4 . 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.

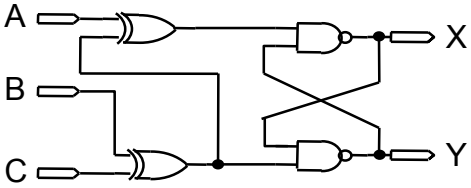
A = **..1..| 1..|..0..|..0.. |..1..|**

t_0..|..t_1..|.. t_2..|..t_3..|..t_4..|..

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

t_0..|..t_1..|.. t_2..|..t_3..|..t_4..|..

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

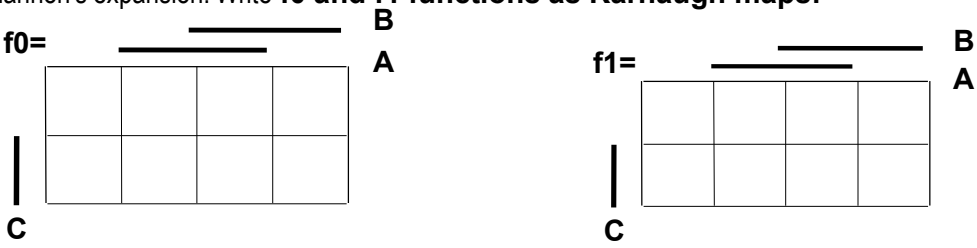


X = ...|... ....|... ....| .....|... ...|

t_0..|..t_1...|.. t_2..|..t_3..|..t_4..|..

Y = ...|... ....|... ....| .....|... ...|

3. Function X=f(A,B,C, X) 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:**



4. What decimal value is 12-bit number 1000 0000 0111 if we interpret it as an integer

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

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

f1<=((C and not B) or (C and B and A)); f1 ☐

f2<=(A xor C) or (A and not C); f2 ☐

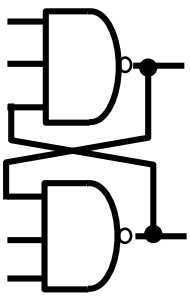
f3<=(A or B) and (not A or B or C); f3 ☐

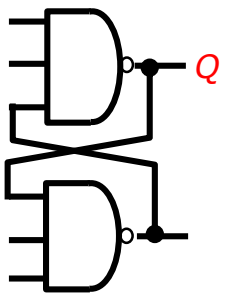
f4<=(not A or not C) and (C or A); f4 ☐

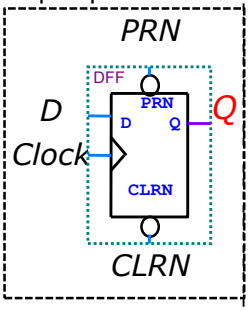
f5<=(not C and A) or (not A and C); f5 ☐

f6<=(A and not B) xor (A and and not B and C); f6 ☐

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







Do not write here

4

5

f0/3

f1/3

2

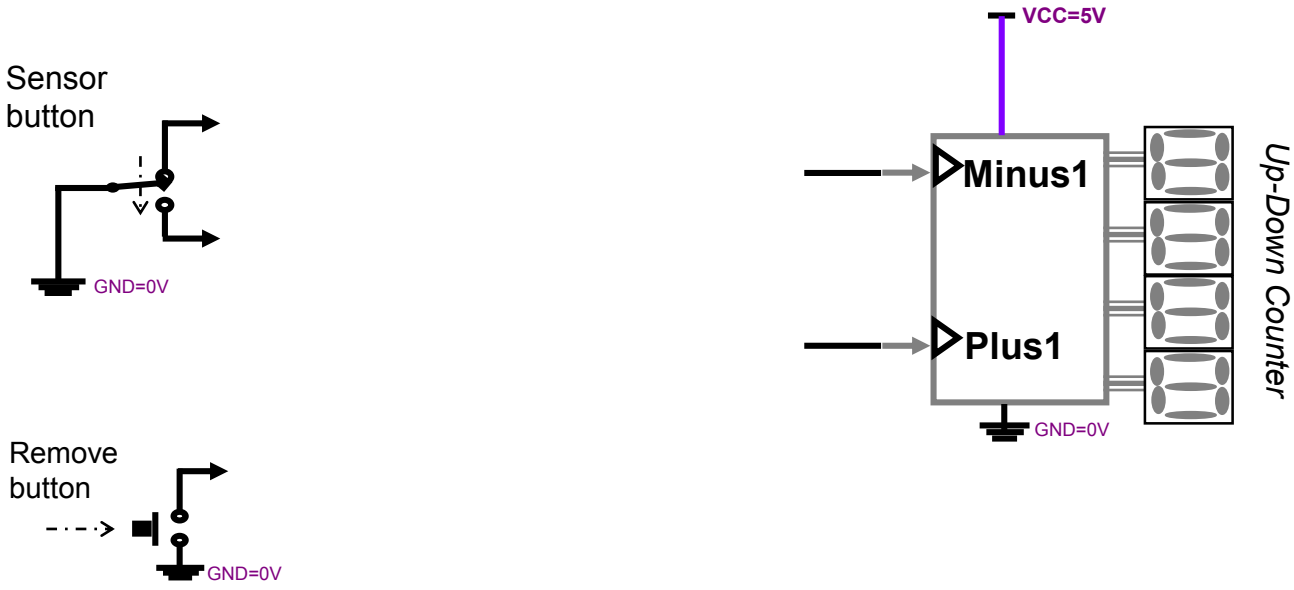
6

7

Part 1

30

7. On an assembly line, there is a mechanical sensor created as two pole button. The sensor detects passes of products. During a next manual check operation, any product can be removed, which is announced by one pole button Remove. The buttons activate clock inputs of Up-Down counter: Plus1 clock input is driven by the sensor, Minus1 clock input by the remove button.
Add 7 necessary elements and wires.



15

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

```
library ieee; use ieee.std_logic_1164.all;
entity XXX is port(  A, B, C, D : in std_logic; Q : out std_logic ); end;
architecture rtl of XXX is  constant E:std_logic:='0'; begin  process (A, D)
variable qv:std_logic;  begin  if D='0' then qv:=E;  elsif rising_edge(A) then
if C='1' then qv:=B; else  qv:=not qv; end if;   end if;  Q<=qv;  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 a better formatting.

15

B1. F1<= A xor (B or C);

F1<=(A and not B and not C) or (not A and (B or C));.....

F2<= **not** (A and (B **xor** C));

F2<=not A or ((not B or C) and (B or not C));.....

Do not write here

SOLUTION

2. Inputs A, B, C have values shown in the figure in times t_0, t_1, t_2, t_3, t_4 . 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.

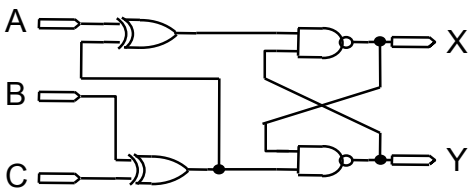
A = ..**1..| 1..|..0..|..0.. |..1..|**

t_0..|..t_1..|.. t_2..|..t_3..|..t_4..|..

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

t_0..|..t_1..|.. t_2..|..t_3..|..t_4..|..

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



X = ... 0..|…1...|…1..| ...1...|...0…|

t_0..|..t_1...|.. t_2..|..t_3..|..t_4..|..

Y = ...1...|…0….|…0….| ..1…|.. 1…|

SOLUTION

3. Function X=f(A,B,C, X) 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=

C \ B A	00	01	11	10
0	1	0	1	0
1	0	1	0	1

f1=

C \ B A	00	01	11	10
0	1	0	1	1
1	1	1	0	1

4. What decimal value is 12-bit number 1000 0000 0111 if we interpret it as an integer

a) As unsigned....2055..... *b)* signed in two's-complement*.....-2041.*

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

f1<=((C and not B) or (C and B and A));

f2<=(A xor C) or (A and not C);

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

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

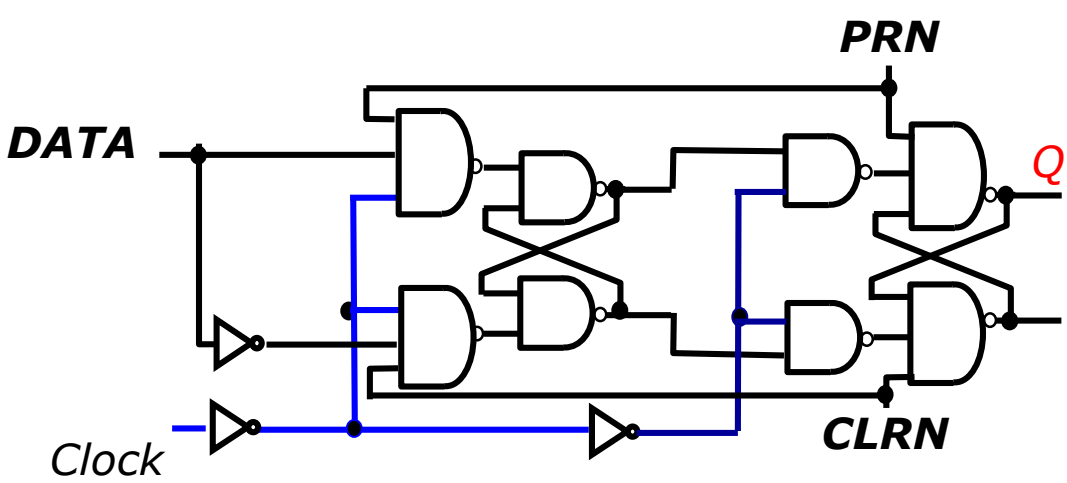
f5<=(not C and A) or (not A and C);

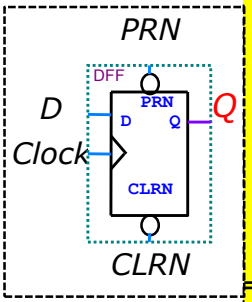
f6<=(A and not B) xor (A and and not B and C);

f1 ☐
f2 ■
f3 ☐
f4 ■
f5 ■
f6 ☐

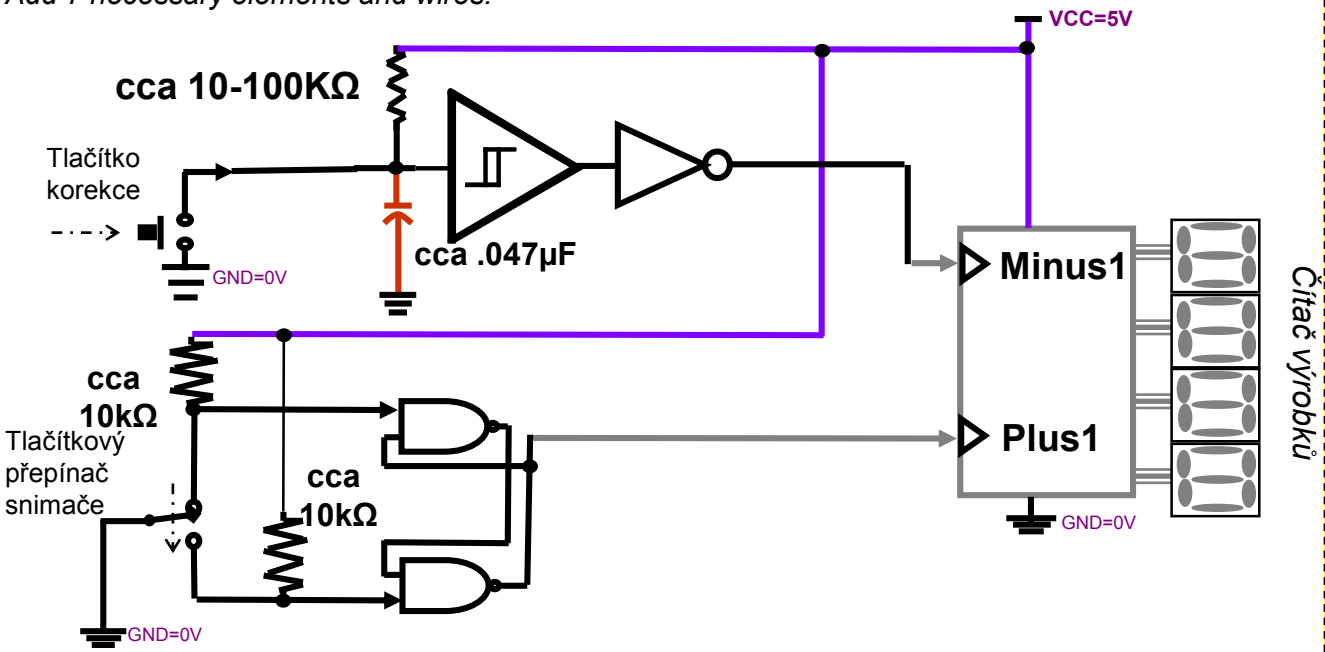
SOLUTION

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





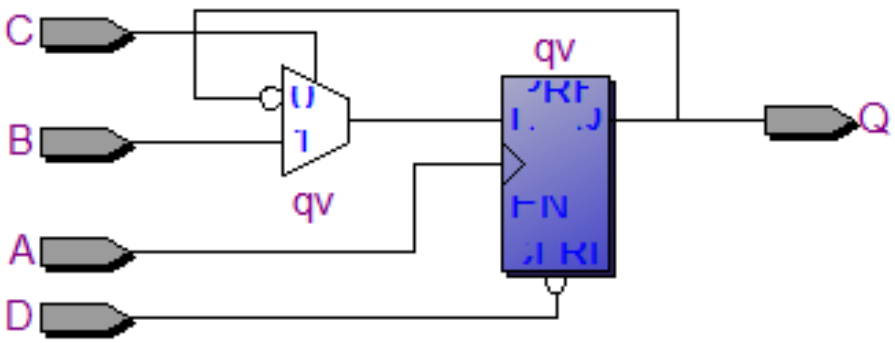
7. On an assembly line, there is a mechanical sensor created as two pole button. The sensor detects passes of products. During a next manual check operation, any product can be removed, which is announced by one pole button Remove. The buttons activate clock inputs of Up-Down counter: Plus1 clock input is driven by the sensor, Minus1 clock input by the remove button.
Add 7 necessary elements and wires.



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

```
library ieee; use ieee.std_logic_1164.all;
entity XXX is port(  A, B, C, D : in std_logic; Q : out std_logic ); end;
architecture rtl of XXX is  constant E:std_logic:='0'; begin  process (A, D)
variable qv:std_logic;  begin  if D='0' then qv:=E;  elsif rising_edge(A) then
if C='1' then qv:=B; else  qv:=not qv; end if;   end if;  Q<=qv;  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 a better formatting.



Do not write here

SOLUTION

SOLUTION

Surname First name:.....

Answer sheet of the LPS test on 21.12.2021 - write only your answers here

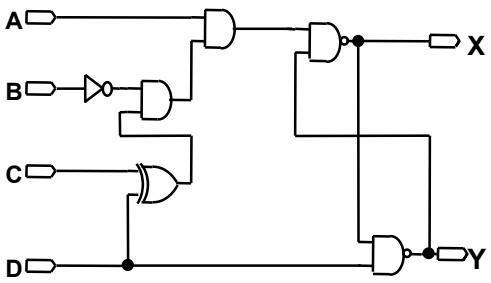
1. *Check all logical functions that have another function identical to them here:*

y1 <= (D and not C) or (not C and A) or (D and B); y1 ☐
y2 <= (D and C) xor (B and A); y2 ☐
y3 <= (D or A) and (not C or B) and (D or not C); y3 ☐
y4 <= (D and B) or (D and not C and not B) or ((D xor A) and not C); y4 ☐

Do not write here
-2 for each error
8

Inputs A,B,C,D have values shown in the figure in times t_0, t_1, t_2, t_3, t_4 . 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.

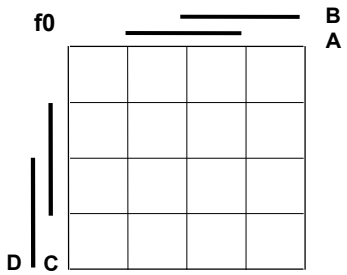
A = ..**0..| 1..|..1..|..1..|..1..|**
t_0..|..t_1..|.. t_2..|..t_3..|..t_4..
B = ..**0..|..0..|..1..|..0..|..0..|**
t_0..|..t_1..|.. t_2..|..t_3..|..t_4..
C = ..**0..|..0..|..1..|..1..|..0..|**
t_0..|..t_1..|.. t_2..|..t_3..|..t_4..
D = ..**1..|..1..|..0..|..0..|..1.|**

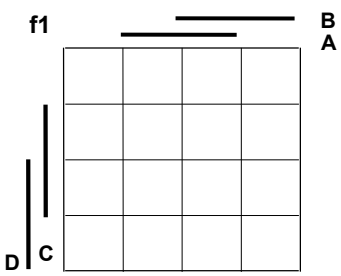


X = ...|… ….|… ….| .....|.....|
t_0..|..t_1...|..t_2...|..t_3...|..t_4..
Y = ...|… ….|… ….| .....|.....|

5

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





4+4
8

4. You only have 2-inputs NAND gates available, nothing else! Use them and the wires to create an XOR gate, ie Y <= A xor B.

A →

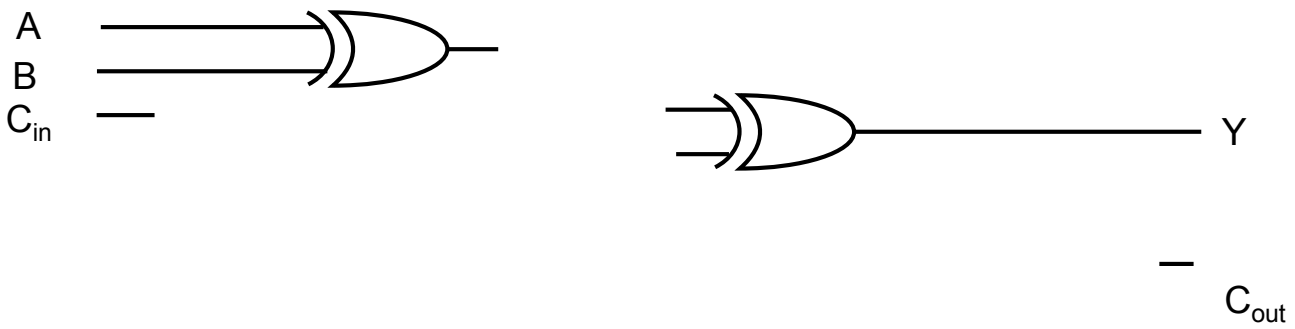
→ Y

B →

3

Do not write here

5. The student came to the exam just to try it and did not complete the scheme of a complete single-bit adder. Finish the figure.



3

6. Using 6 logic gates, create a 4-bit binary adder +1, ie Y <= A + 1.
A0, Y0 are the lowest bits, A3 and Y3 the highest bits.

3

A0 → → **Y0**

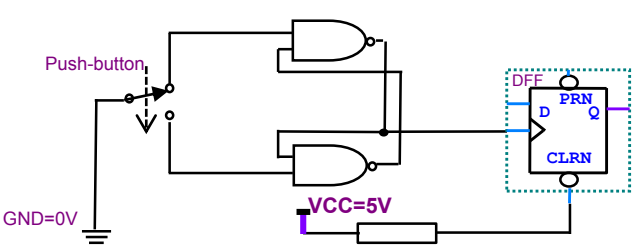
A1 → → **Y1**

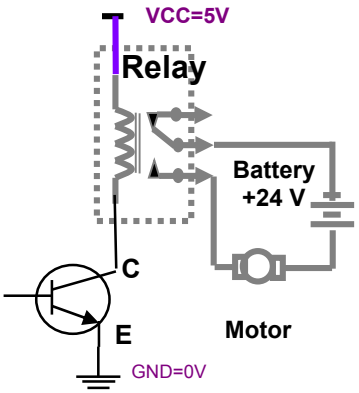
A2 → → **Y2**

A3 → → **Y3**

1. part
30

7. Complete the malfunctioning CMOS circuit so that one press of the two-pole button turns on the motor and the next press of the same button turns the motor off. The motor must always be **switched off** when the power supply is **switched on**! *Hint: A total of seven elements must be added correctly.*





-2 points for a less important missing / bad element, -10 points for a devastating mistake

10

8. *The printer asked you to decipher the following badly formatted code.*

```
library IEEE; use IEEE.STD_LOGIC_1164.all;
entity xxx is port (a, b, c : in std_logic; d : out std_logic); end;
architecture rtl of xxx is begin   process(a,b,c) variable z:std_logic_vector(0 to 3);
begin if a = '0' then z:="0001"; elsif rising_edge(b) then  z:=z(1 to 3) & z(0) ;
        if c='1' then d<=not z(3); else d<=z(3); end if;
        end if; end process; end rtl;
```

Draw a concise diagram of the logic circuit corresponding to the VHDL code and add a label below the figure describing the operation of the circuit. Instructions: rewrite the program first in the correct format.

9. **Premium question** for which only a fully functional solution is accepted:
Fill in the missing parts of the VHDL program so that you create the divider of clk frequency by 12 with a symmetric output q12, i.e. q12 has duty cycle 50 %. Choose such code that will be also executed correctly in the simulation.

```
library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity div12 is port (clk : in std_logic; q12: out std_logic);
end entity;
architecture rtl of div12 is
begin
.....
.....
. .....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
end architecture;
```

Do not write here

10

2. part
30