

1. Zaškrtnutím označte všechny logické funkce, které mají jinou funkci s nimi shodnou:

$$y_1 \leq (A \text{ and not } D) \text{ or } (\text{not } A \text{ and not } B \text{ and } C) \text{ or } (B \text{ and } D);$$

y1

$$y_2 \leq ((B \text{ or not } C) \text{ and } (\text{not } A \text{ or } C)) \text{ xor } ((A \text{ or not } B \text{ or not } D) \text{ and } (\text{not } A \text{ or not } B \text{ or not } C));$$

y2

$$\text{and } (\text{not } A \text{ or not } B \text{ or not } C) \text{ and } (\text{not } A \text{ or } B \text{ or not } D);$$

y3

$$y_3 \leq (A \text{ or not } B \text{ or } D) \text{ and } (\text{not } A \text{ or } B \text{ or not } D) \text{ and } (A \text{ or } B \text{ or } C);$$

y4

$$y_4 \leq (B \text{ xor not } D) \text{ or } (A \text{ and } B) \text{ or } (\text{not } A \text{ and not } B \text{ and } C);$$

2. Vstupy A, B, C, D měly v časech t_0, t_1, t_2, t_3, t_4 hodnoty uvedené v obrázku. Napište hodnoty X a Y výstupů. Intervaly mezi změnami vstupů jsou dost dlouhé, takže lze zanedbat zpoždění hradel.

$$A = \dots 0..|0..|1..|0..|0..|$$

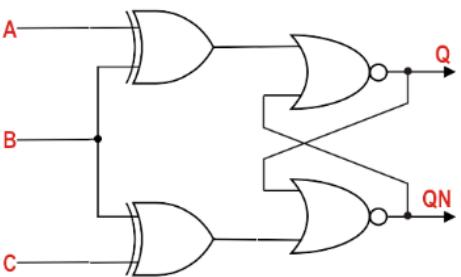
$$t_0..|..t_1..|..t_2..|..t_3..|..t_4..$$

$$B = \dots 0..|..1..|1..|0..|1..|$$

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

$$C = \dots 1..|..0..|..0..|..0..|..1..|$$

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

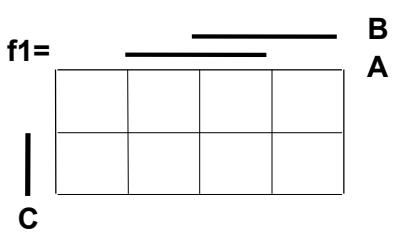
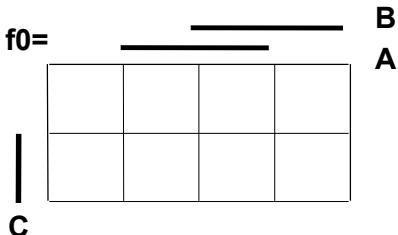


$$Q = \dots \dots | \dots \dots | \dots \dots | \dots \dots |$$

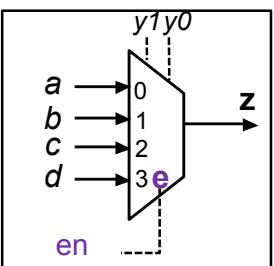
$$t_0..|..t_1..|..t_2..|..t_3..|..t_4..$$

$$QN = \dots \dots | \dots \dots | \dots \dots | \dots \dots |$$

3. Funkci $Q=f(A,B,C, Q)$ obvodu z otázky 2, rozložte na $X= (\text{not } Q \text{ and } f_0(A, B, C)) \text{ or } (Q \text{ and } f_1(A, B, C))$ pomocí Shannonovy expanze. Výsledné funkce f_0 a f_1 napište jako Karnaughovy mapy:



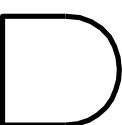
4. Doplňte do schématu minimum prvků potřebných k tomu, aby vznikl obvod na uvedené schematické značce. Vstup en (enable) je aktivní v '1', tj. při en ='0', je Z vždy v '0'.



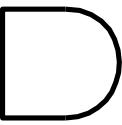
en

$y_1 \downarrow$ $y_0 \downarrow$

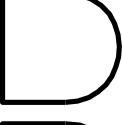
$a \rightarrow$



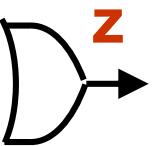
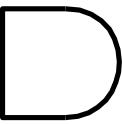
$b \rightarrow$



$c \rightarrow$



$d \rightarrow$



4

-2 body za méně důležitý
chybějící/špatný prvek,
-8 bodů za zničující chybu

8

5

5. Doplňte: Automat Moore (Mealy) je uspořádaná šestice $M = \langle X, S, Z, \omega, \delta, s_0 \in S \rangle$, kde

X

S

Z

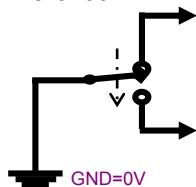
δ : Moore..... Mealy.....

ω : Moore..... Mealy.....

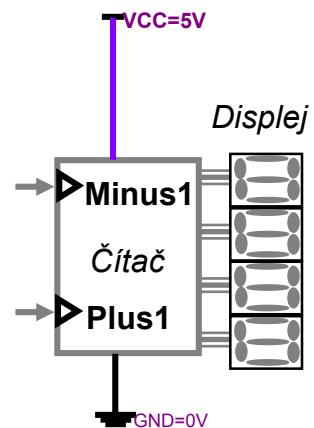
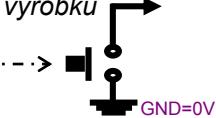
s_0

6. Na výrobní lince je snímač výrobků provedený jako jednopólové tlačítko. Při následné manuální technické kontrole, může být výrobek vyřazen, což se hlásí dvoupólovým tlačítkem korekce. Tlačítka aktivují hodinové vstupy čítače výrobků, ty reagují na náběžné hrany, a snímač ovládá vstup Plus1, korekce vstup Minus1. Čítač musí číst už v okamžiku stisknutí tlačítka. Doplňte 8 potřebných součástek mezi tlačítko a čítač a k tomu propojky.

Dvoupólové
tlačítko
korekce



Jednopólové
tlačítko
snímače
výrobků



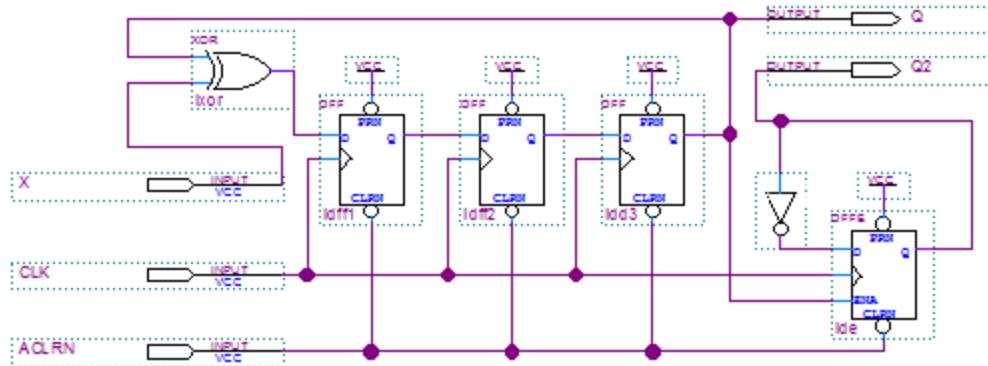
7. Souběžnými příkazy (tj. bez použití sekvenčních) popište obvod z otázky 4:

```
library ieee; use ieee.std_logic_1164.all;
entity Test is port(a,b,c,d,en:in std_logic;y:in std_logic_vector(1 downto 0); Z:out std_logic);
end entity;
architecture dataflow of Test is
```

begin

end architecture;

8. Obvod na obrázku popište VHDL příkazy



```
library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity Test2 is port( ACLRN, CLK, X : in std_logic;Q, Q2 : out std_logic );
end entity;
architecture rtl of Test2 is
```

begin

8

9. Prémiová otázka, u níž se uzná jen plně funkční řešení: Doplňte chybějící části VHDL tak, aby vznikl 3-bitový synchronní čítač pro generátor pily, hodnota výstupu=0,1,2,3,4,5,6,7,6,5,4,3,2,1,0,1.. atd., a to pokud je vstup en (enable) v '1'. Při en='0' drží výstup svou poslední hodnotu. Z té dále pokračuje po en='1'. Volte co nejúspornější kód, který se správně provede i v simulaci.

```
library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity pila is port (clk, en : in std_logic; q: out std_logic_vector(2 downto 0));
end entity;
architecture rtl of pila is
begin
```

```
end architecture;
```