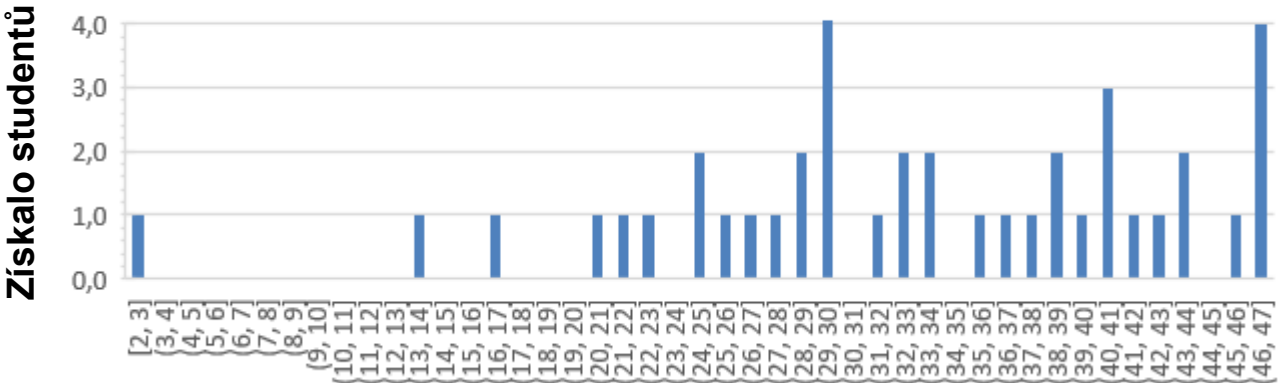
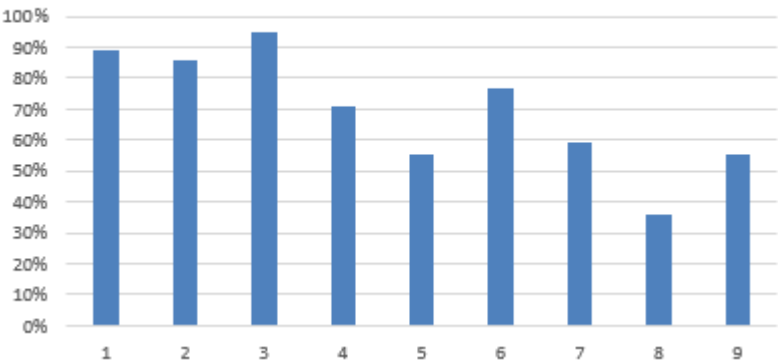


Statistické výsledky zkoušky LSP ve středu 9. června 2025

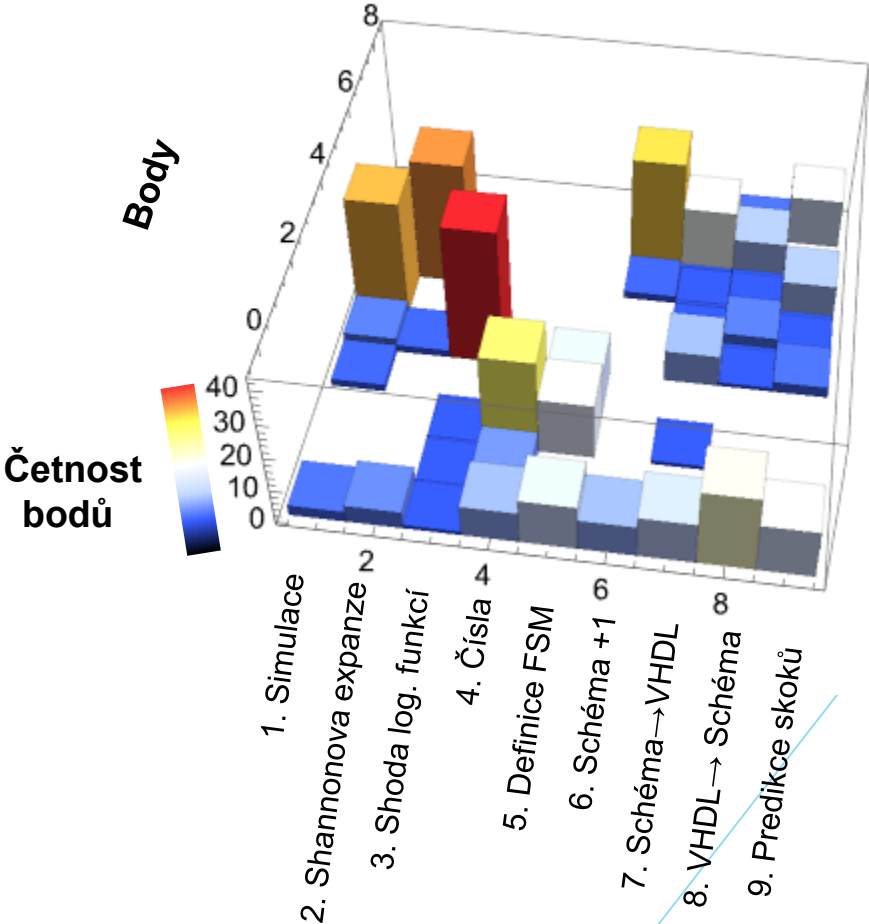
Četnost bodů z písemky



Úspěšnost u otázek 1 až 9



Body

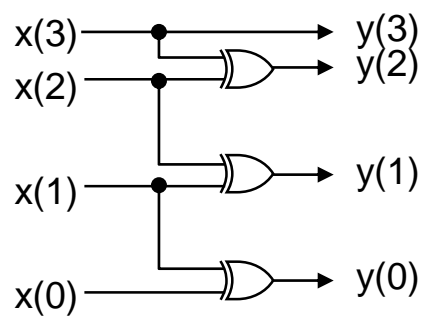


6. Sčítačka +1 se dá zapojit bez užití úplné sčítačky mnohem jednodušeji pomocí hradel.
Nakreslete její schéma složené z hradel:



7

7. Obvod na obrázku optimálně popište **souběžným** (concurrent) VHDL kódem bez užití sekvenčních příkazů. Plný počet bodů dostanete jen tehdy, popíšete-li celé zapojení v bloku architektury pouze jediným příkazem. Za každý další v ní napsaný Vám bude stržený 1 bod.
Nápověda: Použijte logické operace s std_logic_vector.



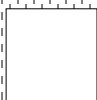
```
library ieee; use ieee.std_logic_1164.all;  
use ieee.numeric_std.all;  
entity Test20250609q7 is port(  
.....  
.....);  
end entity;  
architecture rtl of Test20250609q7 is  
begin  
.....  
.....  
.....  
.....  
.....  
end architecture rtl;
```

7

8. Pro obvod popsáný následujícím kódem nakreslete jeho **blokové schéma** s využitím značek hradel, multiplexoru, sčítačky+1, komparátoru a registrů DFF.

A obvod správně pojmenujte!

```
library ieee; use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity Ex20250609q7 is port ( CLK : in std_logic; Q : out std_logic);
end entity;
architecture rtl1 of Ex20250609q7 is
begin
iproc: process(CLK)
    constant M: integer:=9;
    variable cntr : integer range 0 to M:=0;
    variable x : std_logic:='0';
    begin
        if rising_edge(CLK) then
            if cntr < M then cntr:=cntr+1; else cntr:=0; x:=not x;
            end if;
        end if;
        Q<=x;
    end process;
end architecture;
```



9. C program hledá **minimum** v poli **data**.

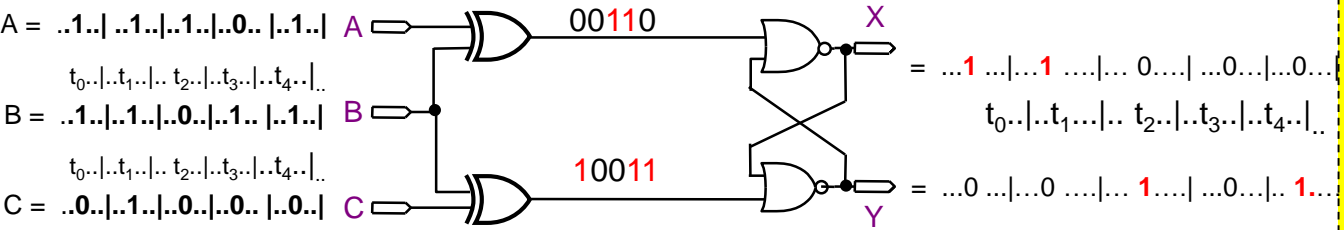
```
int data[] = { 0, 1, -2, -3, 4, -5, 6, -7, -8, 9 };  
int min = INT_MAX; // =2147483647 (v <limits.h>)  
for (int i = 0; i < sizeof(data)/sizeof(int); i++) // i<10  
{ if (data[i] < min) min = data[i];  
}
```

Kolik v něm bude chybných skokových predikcí za předpokladu, že for-loop s konstantními mezemi se úsporně přeložila cyklem do-while a procesor používá pouze:

1bitové prediktory, které měly svůj výchozí stav Not-Taken, NT, misses=.....

2bitové prediktory, které měly svůj výchozí stav WT, Weakly Taken, misses=.....

1. Vstupy A, B, C měly v časech t_0, t_1, t_2, t_3 hodnoty uvedené v obrázku. Napište hodnoty X a Y výstupů. Předpokládejte, že intervaly mezi změnami vstupů jsou tak dlouhé, že lze zanedbat zpoždění hradel.



2. Funkci $X=f(A,B,C, X)$ obvodu z otázky 1, rozložte na tvar $X = (\text{not } X \text{ and } f_0(A, B, C)) \text{ or } (X \text{ and } f_1(A, B, C))$ pomocí Shannonovy expanze. **Výsledné funkce f_0 a f_1 zapište jako Karnaughovy mapy:**

$f_0 =$

	B			
	A			
	0	0	1	0
C	1	0	0	0

$f_1 =$

	B			
	A			
	1	0	1	0
C	1	0	1	0

$$X = (\text{not } C \text{ and } B \text{ and } A) \text{ or } (C \text{ and not } B \text{ and not } A) \text{ or } (X \text{ and not } B \text{ and not } A) \text{ or } (X \text{ and } B \text{ and } A)$$

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

- $y_1 \leq (A \text{ or } D) \text{ and } (\text{not } A \text{ or } C);$
- $y_2 \leq C \text{ or } (A \text{ and } C \text{ and } B) \text{ or } (\text{not } A \text{ and } C \text{ and } D);$
- $y_3 \leq (\text{not } A \text{ and } D) \text{ or } (A \text{ and not } D) \text{ or } (C \text{ and } D);$
- $y_4 \leq (C \text{ and } D) \text{ or } (\text{not } A \text{ xor not } D);$

☐

☐

☒

☒

4. Uložíme-li dolní bity výsledku operace s čísly $124+125+126+127$ do 8-bitového registru,

jaká hodnota v něm bude, bereme-li ji jako 9bitové dekadické číslo

$$2^{**7} - 4 + 2^{**7} - 3 + 2^{**7} - 2 + 2^{**7} - 1 = 4 * (2^{**7}) - 10 = 2^{**9} - 10 \text{ (byte)} \equiv 2^{**8} - 10 = 246 \mid -10$$

a) bez znaménka (unsigned).....246..... b) se znaménkem ve druhém doplňku (signed).....-10.....

5. *Doplňte*: Automat Moore (Mealy) je... přednáška 9 snímek 20

Uspořádaná šestice

$$M = \langle X, S, Z, \omega, \delta, s_0 \in S \rangle$$

- X - konečná množina všech vstupních vektorů
- Z - konečná množina všech výstupních vektorů
- S - konečná množina všech vnitřních stavů

δ - přechodová funkce - zobrazení $\delta: X \times S \rightarrow S$

ω - výstupní funkce - zobrazení ω :

$$\omega: S \rightarrow Z$$

$$\omega: X \times S \rightarrow Z$$

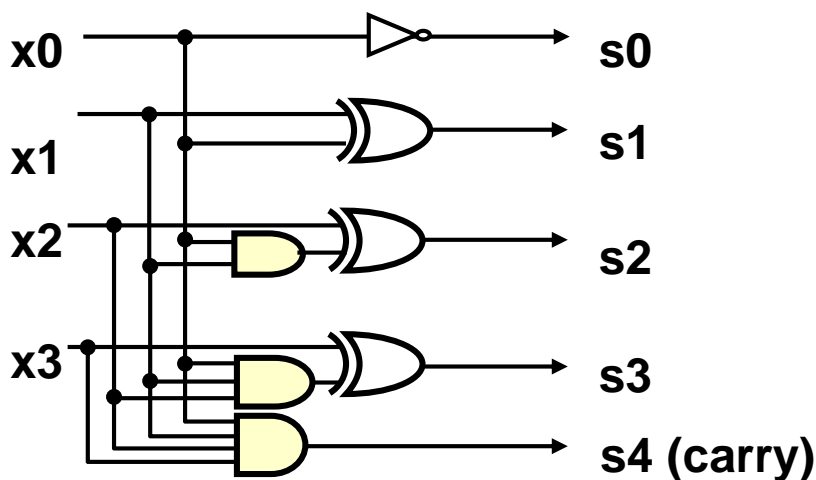
Moore

Mealy

- s_0 - počáteční stav $s_0 \in S$

6. *Doplňte*: Sčítačka +1 - přednáška 1 snímky 35 až 3

+Logické obvody na FPGA - str. 104 + obrázek 120



7. *obvod - přednáška 2, snímek 28*

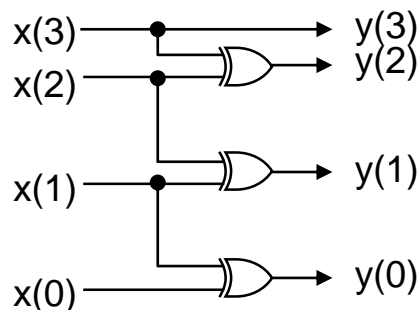
```
library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity Test20250609q7 is port( x: in std_logic_vector(3 downto 0);
                               y: out std_logic_vector(3 downto 0)); end entity;
architecture rtl of Test20250609q7 is
begin
    y<= ('0' & x(3 downto 1)) xor x; -- přednáška 2
end architecture rtl;
```

alternativy

$y <= x(3) \& (x(2) \text{ xor } x(3)) \& (x(1) \text{ xor } x(2)) \& (x(0) \text{ xor } x(1));$

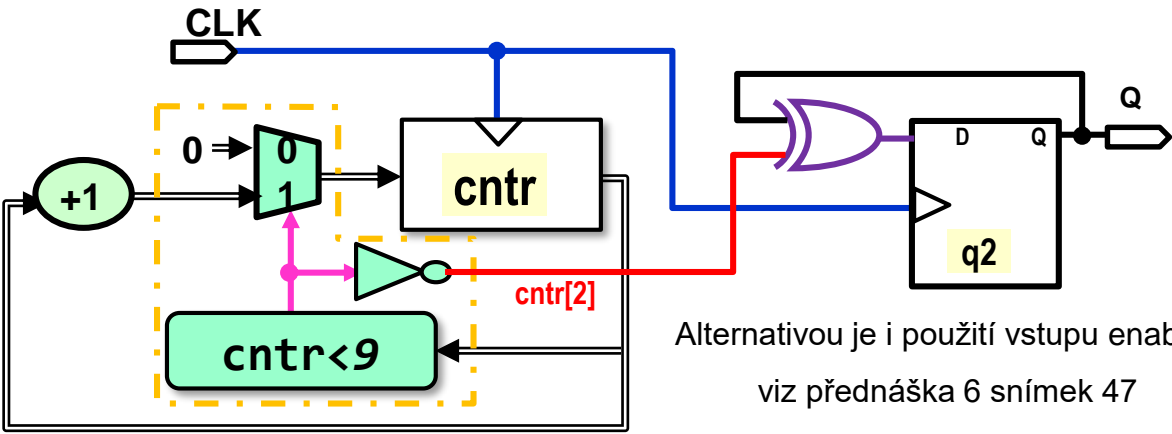
nebo

$y <= x(3) \& (x(3 \text{ downto } 1) \text{ xor } x(2 \text{ downto } 0));$

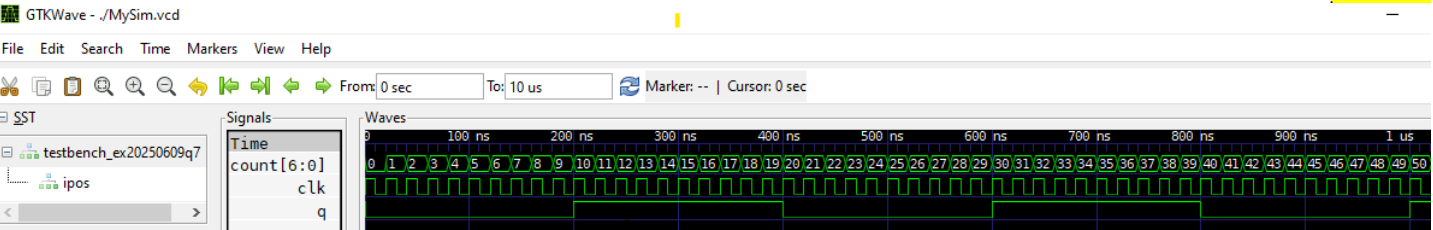


Zde
nepište

8. Pro obvod popsaný následujícím kódem nakreslete jeho **blokové schéma** s využitím značek hradel, multiplexoru, sčítačky, komparátoru a registrů DFF. **A obvod správně pojmenujte! - přednáška 6 snímky 38 až 47**



Alternativou je i použití vstupu enable, viz přednáška 6 snímek 47



Dělič frekvence 20 se symetrickým výstupem Q, tj. mající pracovní cyklus 50 %

9. C program hledá **minimum** v poli **data**.

```
int data[] = { 0, 1, -2, -3, 4, -5, 6, -7, -8, 9 };;
int min = INT_MAX; // =2147483647 (v <limits.h>)
for (int i = 0; i < sizeof(data)/sizeof(int); i++) // i<10
{ if (data[i] < min) min = data[i];
}
```

Kolik v něm bude chybných skokových predikcí za předpokladu, že for-loop s konstantními mezemi se úsporně přeložila cyklem do-while a procesor používá pouze:

jednobitové prediktory, které měly svůj výchozí stav Not-taken, NT,
misses=7 + 2(for-loop)=9.....

dvou bitové prediktory, které měly svůj výchozí stav WT, Weakly Taken,
misses=6 + 1 (for-loop)=7.....

Branch	N	T	N	N	T	N	T	N	N	T
Data	0	1	-2	-3	4	-5	6	-7	-8	9
1 bit	NT	NT>T	T>N	N	NT>T	T>NT	NT>T	T>NT	NT	NT>T
Trefil se	hit	miss	miss	hit	miss	miss	miss	miss	hit	miss
2 bit	WT>WNT	WNT>WT	WT>WNT	WNT>NT	NT>WNT	WNT>NT	NT>WNT	NT>WNT	WNT>NT	NT>WNT
Trefil se	miss	miss	miss	hit	miss	hit	miss	hit	hit	miss