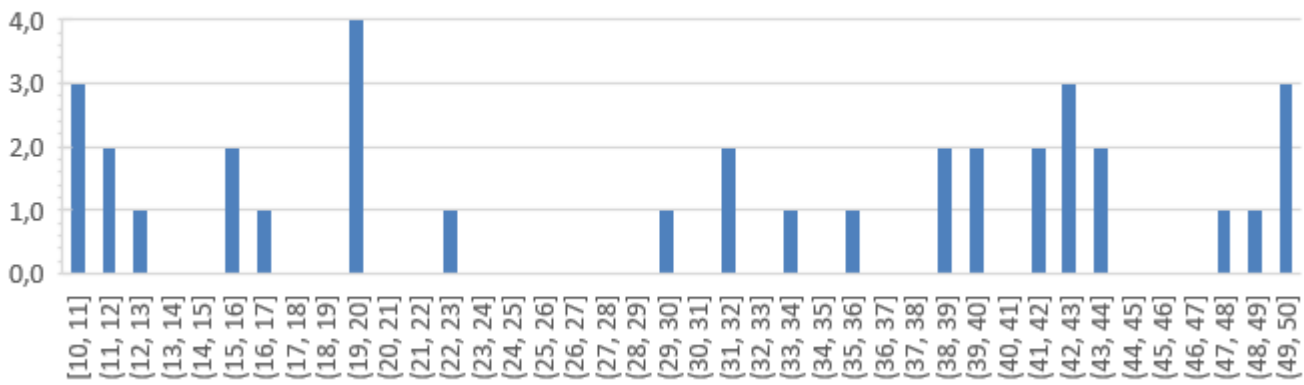
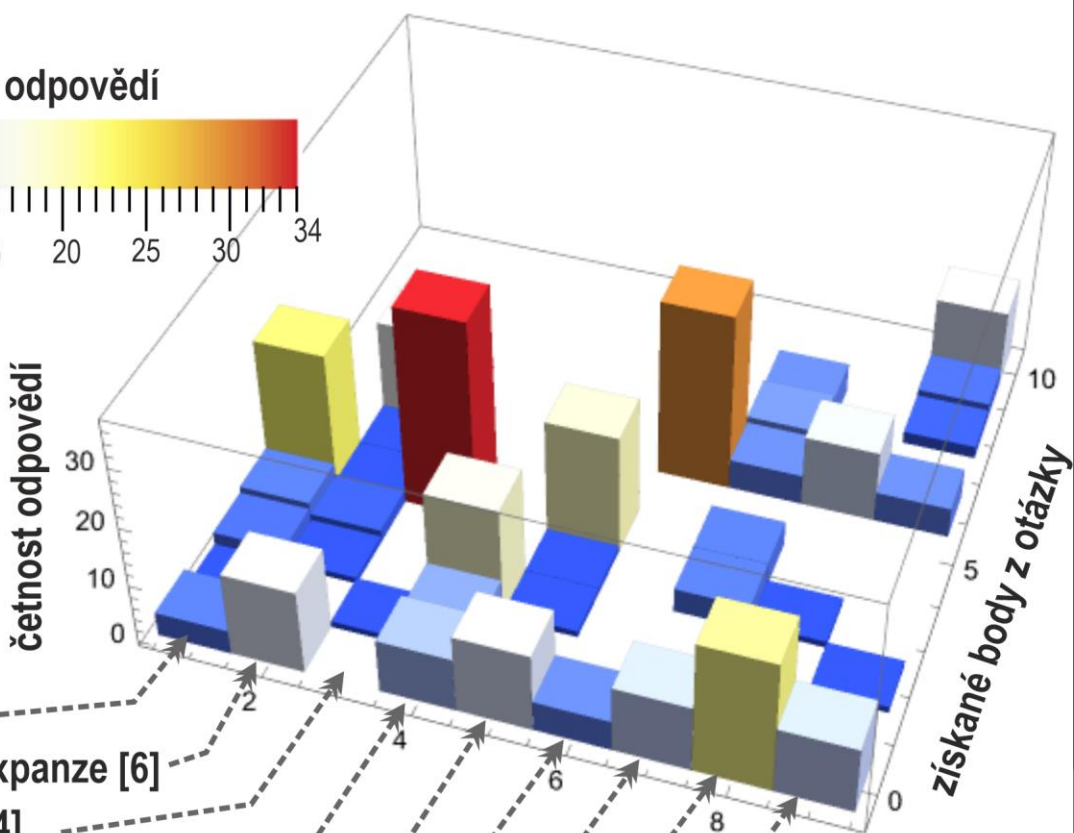
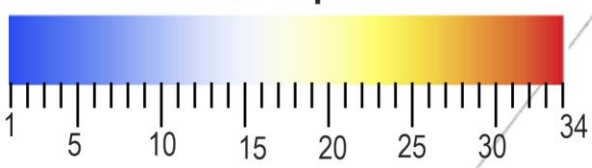


# Statistické výsledky zkoušky LSP v úterý 28. května 2024

Četnost bodů z písemky



četnost odpovědí

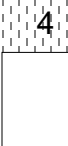


Otázka

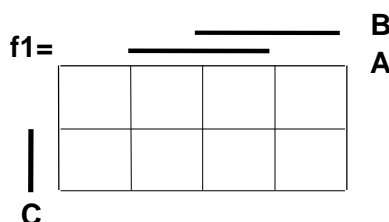
- 1. Simulace [4]
- 2. Shannonova expanze [6]
- 3. Shoda funkcí [4]
- 4. Signed/Unsigned [2]
- 5. Definice automatu [4]
- 6. Kaskáda MUX [6]
- 7. Kaskáda ve VHDL [8]
- 8. Predikce skoků [6]
- 9. VHDL kód demultiplexoru [10]

Zde  
nepište

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.



6



y4 ☐

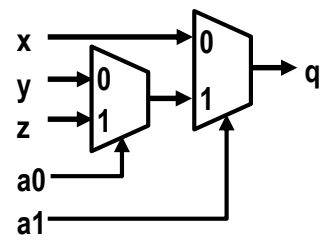
2

b) se znaménkem v zobrazení ve druhém doplňku (signed).....

s0 je .....

4

6. Použijte jen hradla AND, NAND, OR, NOR a inventory NOT a pomocí nich zapojte schéma vlevo.



Zde  
nepíše

6

7. Obvod z otázky 6 nahoře popište ve VHDL jak souběžným, tak sekvenčním příkazem

```
library ieee; use ieee.std_logic_1164.all;use ieee.numeric_std.all;
entity Test20240528q6 is port( x, y ,z, a1,a0: in std_logic:= '0';
                                qcon, qseq: out std_logic := '0')
```

```
end entity;
architecture rtl of Test20240528q6 is
begin
```

**-- qcon souběžným (concurrent) příkazem.....**

qcon  
4

**-- qseq sekvenčním příkazem.....**

iseq: process.....

.....begin.....

qseq  
4

.....end process;.....

end architecture;

```
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];
}
```

1bitové prediktory, ktoré mali svoj výchozí stav Not-Taken, NT, misses=.....

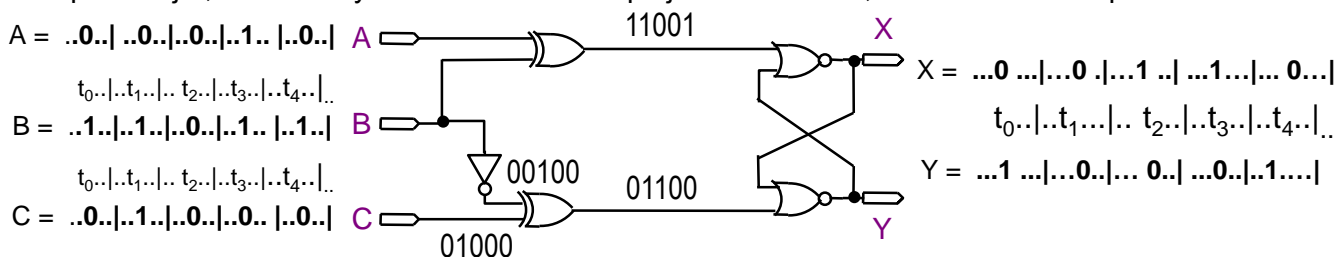
2bitové prediktory, ktoré mali svoj výchozí stav WT, Weakly Taken, misses=.....

b) Nakreslete i jeho vnitřní zapojení, v němž použijete jen hradla OR, AND a NOT.

Symbol  
4

## Schema

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  $f_1$  napište jako Karnaughovy mapy:**

$f_0 =$

	$B$			
	$A$			
$C$	1	0	0	0
	0	0	1	0

$f_1 =$

	$B$			
	$A$			
$C$	1		1	
	1		1	

3.  $y_2 = y_4$

4. (unsigned).....510..... (signed).....-2.....

5. Automat Moore (Mealy): přednáška 8, snímek 19

6. Dvouvstupové multiplexory byly opakovaně v přednáškách, jinak též Logické obvody na FPGA str. 78 až 79

7. správně takto, ale tolerovaly se i vnořené if a rozumně napsaný select a case

```

library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity Test20240528q6 is port( x, y, z, a1, a0: in std_logic:= '0';
                                qcon, qseq: out std_logic := '0');

```

```

end entity;
architecture rtl of Test20240528q6 is
begin
-- souběžnými příkazy
    Qcon<= x when a1='0' else y when a0='0' else z;

```

```

-- sekvencní popis
    iseq: process(x,y,z,a1,a0)
    begin
        if a1='0' then Qseq<=x;
        elsif a0='0' then Qseq<=y; else Qseq<=z;
        end if;
    end process;
end architecture;

```

8. součet chybných predikcí for smyčky a vnitřního if příkazu je u obou prediktorů 7.

9. Kód popisuje demultiplexor, který se mnohokrát objevil v přednáškách. Podrobně je popsán též v Logické obvody na FPGA str. 76 až 77.