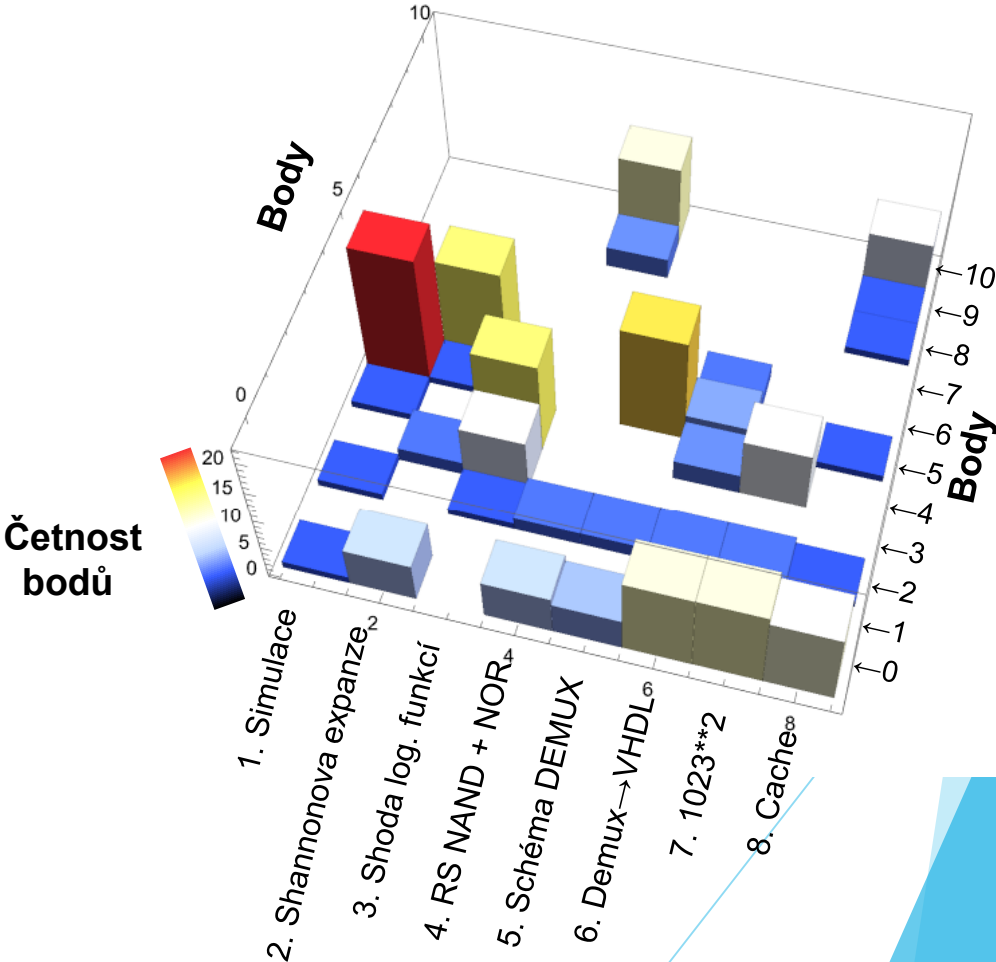
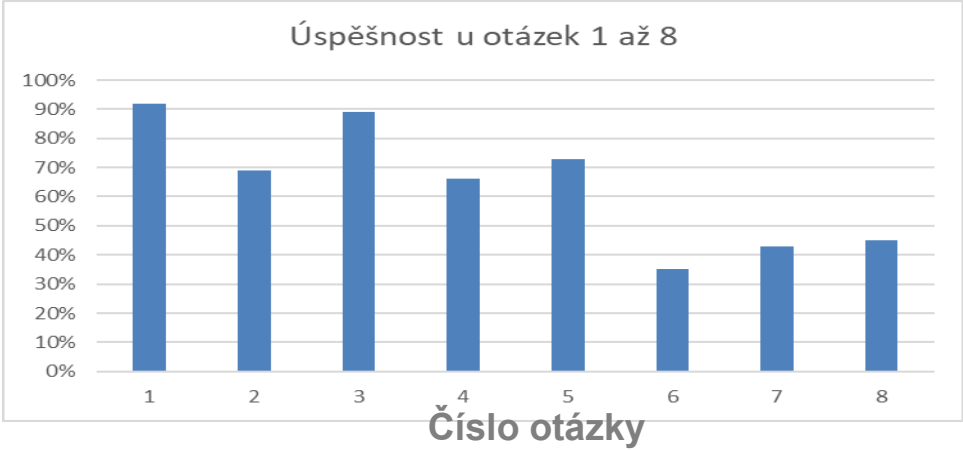
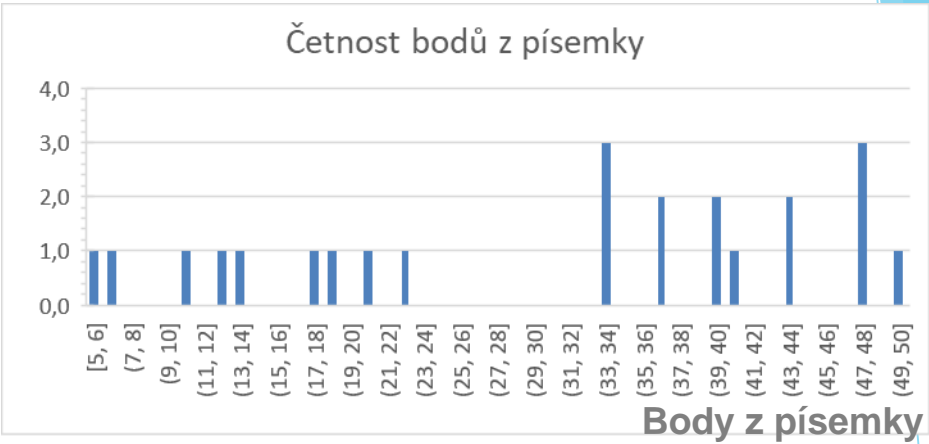


Statistické výsledky zkoušky LSP ve středu 19. června 2025

Získalo studentů



1.Vstupy A, B, C měly v časech t_0 až t_4 hodnoty uvedené v obrázku. Napište hodnoty X a Y výstupů. Předpokládejte, že intervaly mezi změnami vstupů jsou tak dlouhé, že lze zanedbat zpoždění hradel.

A =

0	1	0	1	1
t_0	t_1	t_2	t_3	t_4

B =

1	0	1	1	1
t_0	t_1	t_2	t_3	t_4

C =

1	0	1	1	0
t_0	t_1	t_2	t_3	t_4

X =

...
t_0	t_1	t_2	t_3	t_4

Y =

...
t_0	t_1	t_2	t_3	t_4

5

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

$f_1=$

B

A

6

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

y1 <= (not B or D) and (A or B or not D) and (B or C or not D)

y2 <= (not B and not D) or (B and D) or (A and C and D);

y3 <= (A and not B) or (B and C and D) or (not A and C)

y4 <= (B xnor D) or (A and C and D)

y1 ☐

y2 ☐

y3 ☐

y4 ☐

4

4. Nakreslete vlevo úrovnový klopný obvod typu R-S složený jen z hradel NAND a vpravo jen z hradel NOR tak, aby se v jejich pravdivostních tabulkách nezměnil předepsaný řádek. Ty ostatní doplňte.

S →

→ Q

→ QN

R →

S →

→ Q

→ QN

R →

5

Nand

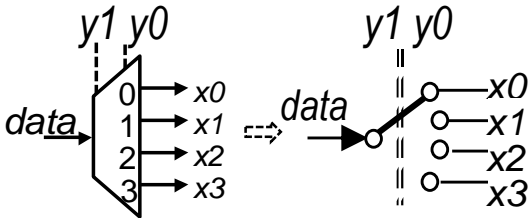
5

nor

Vstupy		Výstupy	
S	R	Q	QN
0	0		
0	1		
1	0	1	0
1	1		

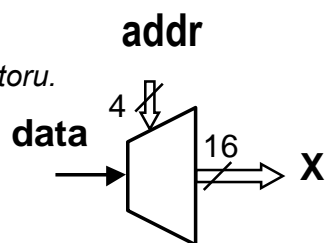
Vstupy		Výstupy	
S	R	Q	QN
0	0		
0	1		
1	0	1	0
1	1		

5. Nakreslete optimální schéma demultiplexoru 1 na 4, které obsahuje jen čtyři logická hradla AND a dva invertory.



addr	y1	y0	x ₀	x ₁	x ₂	x ₃
0	0	0	data	0	0	0
1	0	1	0	data	0	0
2	1	0	0	0	data	0
3	1	1	0	0	0	data

Nápověda: Vzpomeňte si, že můžete indexovat bity v logickém vektoru.



6

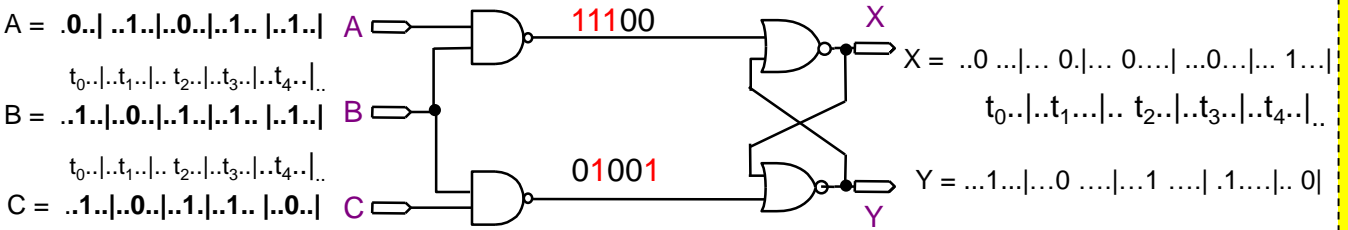
8. 32-bitový procesor má cache dlouhou **4096 bytů a** organizovanou jako **přímo mapovanou** o délce řádku **4 slova**. Vyplňte tabulku mapování adres.

binárně	tag	set	blok-offset
0x0010			
0x0014			
0x0138			
0x013C			
0x101C			
0x1130			
0x113C			

Předpokládejme, že cache je prázdná při spuštění programu.
Označte přístupy do paměti, které budou mít cache hit,
provedou-li se přesně v pořadí uvedeném v tabulce dole.

hex. adresa	cache hit
0x0010	<input type="checkbox"/>
0x0138	<input type="checkbox"/>
0x0014	<input type="checkbox"/>
0x013C	<input type="checkbox"/>
0x101C	<input type="checkbox"/>
0x1130	<input type="checkbox"/>
0x0014	<input type="checkbox"/>
0x113C	<input type="checkbox"/>

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			
C	0	0	1	0
	0	0	0	0

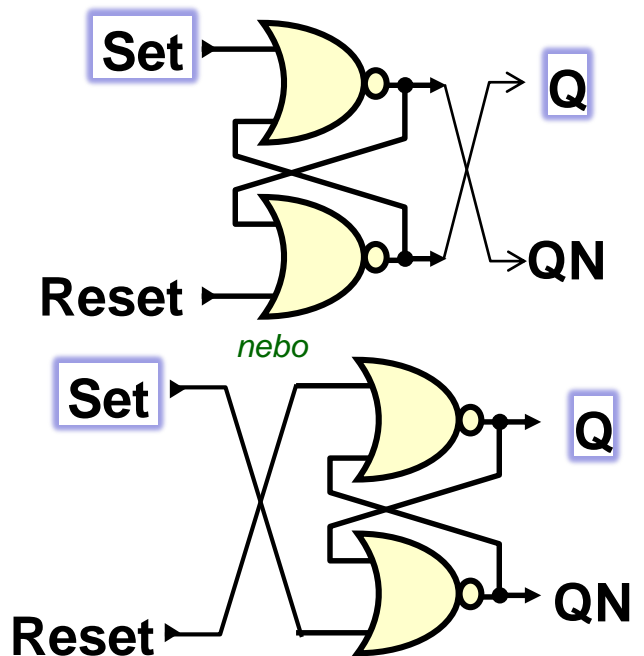
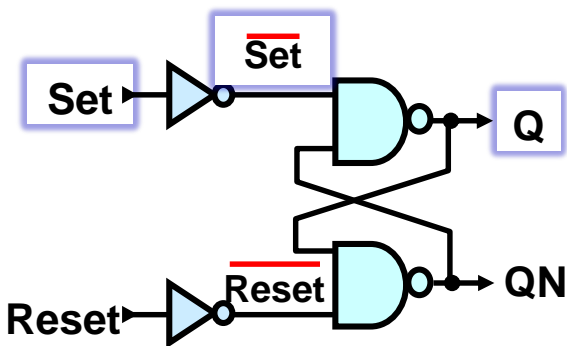
$f_1 =$

	B			
	A			
C	0	0	1	0
	0	0	1	0

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

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

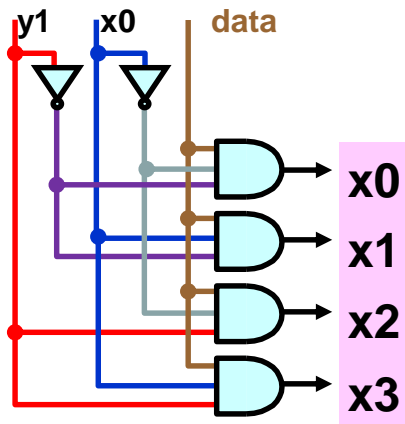
- $y_1 \leq (\text{not } B \text{ or } D) \text{ and } (A \text{ or } B \text{ or } \text{not } D) \text{ and } (B \text{ or } C \text{ or } \text{not } D)$
- $y_2 \leq (\text{not } B \text{ and } \text{not } D) \text{ or } (B \text{ and } D) \text{ or } (A \text{ and } C \text{ and } D);$
- $y_3 \leq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



Vstupy		Výstupy	
S	R	Q	QN
0	0	mem	mem
0	1	0	1
1	0	1	0
1	1	1	1

Vstupy		Výstupy	
S	R	Q	QN
0	0	mem	mem
0	1	0	1
1	0	1	0
1	1	0	0

5. Nakreslete optimální schéma demultiplexoru 1 na 4, které obsahuje jen čtyři logická hradla AND a dva invertory.



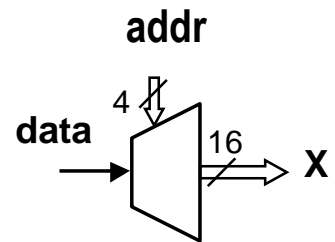
addr	y1	y0	x ₀	x ₁	x ₂	x ₃
0	0	0	data	0	0	0
1	0	1	0	data	0	0
2	1	0	0	0	data	0
3	1	1	0	0	0	data

6. Popište efektivně demultiplexor z předchozí úlohy, avšak rozšířený na 16 výstupů.
Nápověda: Vzpomeňte si, že můžete indexovat bity v logickém vektoru.

```

library ieee; use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity Test20250619q6 is
  port( Data: in std_logic;
        y: in std_logic_vector(3 downto 0);
        x: out std_logic_vector(0 to 15));
end entity;
architecture beh1 of Test20250619q6 is
begin
  process (Data, y)
  begin x<=(others=>'0'); -- initialize all bits
        x(to_integer(unsigned(y))) <= Data; --assign the bit that is selected by y address
  end process;
end architecture;

```



7. Uložíme-li dolní bity výsledku operace $1023^{**}2$ (tj. druhá mocnina 1023) do 12-bitového registru, jaká dekadická hodnota v něm bude, bereme-li ji jako 12bitové číslo?

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

$$\begin{aligned}
 1023^{**}2 &= (2^{**}10 - 1)^{**}2 = (2^{**}20 - 2^{**}10 + 1) = (2^{**}20 - 2^{**}11 + 1) \equiv 12\text{-bitů} \equiv 2^{**}12 - 2^{**}11 + 1 \\
 &= 0x8001 : \text{unsigned } 0x8001 = 2049, \text{ signed } -2^{**}11 + 1 = -2048 + 1 = -2047
 \end{aligned}$$

8. 32-bitový procesor má cache dlouhou **4096 bytů** a organizovanou jako **přímo mapovanou** o délce řádku **4 slova**.
4096 bytů/(4bytové slovo * 4 slova v řádku) = 256 setů
Vyplňte tabulku mapování adres.

binárně		tag	set	blok- offset slova bity 3 a 2	alternat. offset bytu bity 3-0
0x0010	0b0000_0000_0001_0000	0	1	0	0
0x0014	0b0000_0000_0001_0100	0	1	1	4
0x0138	0b0000_0001_0011_1000	0	0x13	2	8
0x013C	0b0000_0001_0011_1100	0	0x13	3	12
0x101C	0b0001_0001_0001_1100	1	1	3	12
0x1130	0b0001_0001_0011_0000	1	0x13	0	0
0x113C	0b0001_0001_0011_1100	1	0x13	0	12

Předpokládejme, že cache je prázdná při spuštění programu.
Označte přístupy do paměti, které budou mít cache hit,
provedou-li se přesně v pořadí uvedeném v tabulce dole.

hex. adresa	cache hit
0x0010	<input type="checkbox"/>
0x0138	<input type="checkbox"/>
0x0014	<input checked="" type="checkbox"/>
0x013C	<input checked="" type="checkbox"/>
0x101C	<input type="checkbox"/>
0x1130	<input type="checkbox"/>
0x0014	<input type="checkbox"/>
0x113C	<input checked="" type="checkbox"/>

