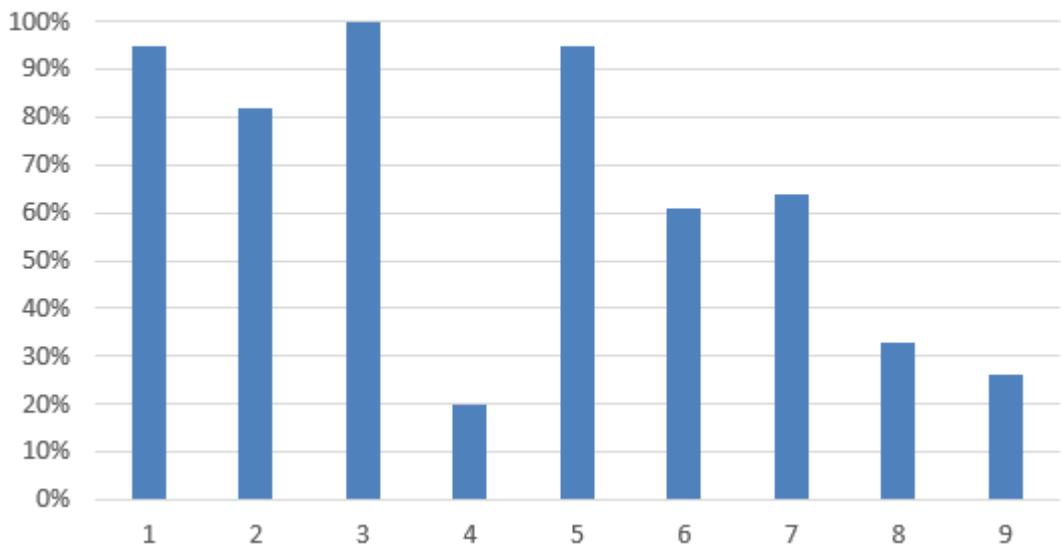


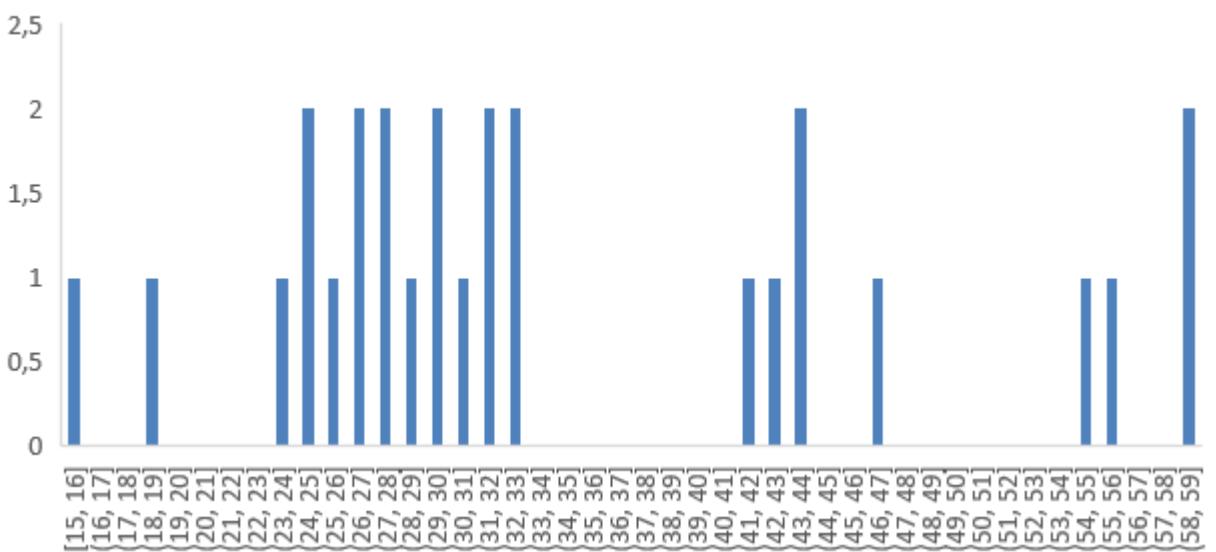
Statistika zkoušky 23. června 2023

Úspěšnost u otázek 1 až 9



Procenta udávají průměrný počet bodů vztažený k maximu z otázky

Četnost bodů z písemky



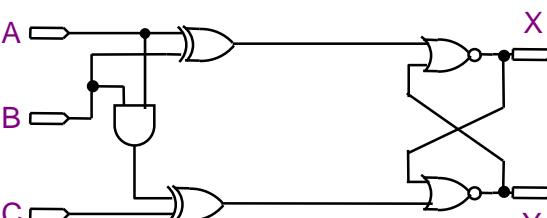
Příjmení: Jméno:

. Jméno:.....

Odpovědní arch písemky LSP ze dne 23.6.2023 - pište sem jen Vaše odpovědi

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.

$$\begin{aligned}
 A &= \dots 1 \dots | 1 \dots | 0 \dots | 0 \dots | 0 \dots | \\
 &\quad t_0 \dots t_1 \dots t_2 \dots t_3 \dots t_4 \dots | \dots \\
 B &= \dots 0 \dots | 1 \dots | 1 \dots | 1 \dots | 0 \dots | \\
 &\quad t_0 \dots t_1 \dots t_2 \dots t_3 \dots t_4 \dots | \dots \\
 C &= \dots 0 \dots | 0 \dots | 0 \dots | 1 \dots | 1 \dots |
 \end{aligned}$$



$X \equiv$ | | | | |

$t_0 \sqcup t_1 \sqcup \dots \sqcup t_n \sqcup t_{n+1} \sqcup t_{n+2}$

$\mathbf{Y} \equiv$ | | | | |

.....|.....|.....|.....|.....|.....|.....|

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:**

$f_0 =$

C

$f_1 =$ **B**
A

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

$y_1 \leq B$ or $(\neg A \text{ and } B \text{ and } D)$ or $(A \text{ and } B \text{ and } C)$;
 $y_2 \leq (B \text{ and } D)$ or $(D \text{ and } \neg A)$ or $(A \text{ and } \neg D)$;
 $y_3 \leq (A \text{ or } D) \text{ and } (B \text{ or } D \text{ or } \neg A)$;
 $y_4 \leq (\neg A \text{ xor } \neg D) \text{ or } (B \text{ and } D)$;

y1
y2
y3
y4

4. Dopište VHDL funkci, která stanoví zaokrouhlenou celočíselnou střední hodnotu údajů uložených v poli typu AS7. **Žádá se zde maximálně rychlý výpočet v obvodu.** Dosáhne se toho po vyřešení: vhodného sčítání a výpočtu zaokrouhleného průměru.

type AS7 is array(0 to 6) of integer range -215 to 2**15-1;**

```
function meanAS7( x: AS7) return integer is
```

hepiste

4

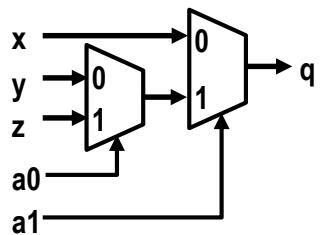
6

4

6

end function;

5. Použijte pouze hradla AND, NAND, OR, NOR a invertory NOT. Zapojte s nimi zapojte schéma vlevo.



8

6. Obvod z otázky 5 nahoře vytvořte ve VHDL nejjednodušším možným způsobem.

```
library ieee; use ieee.std_logic_1164.all;use ieee.numeric_std.all;
entity yyy is port( x,y,z, a1,a0: in std_logic; q: out std_logic);
end entity;
architecture dataflow of yyy is
```

end architecture:

7. 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 konstantnímimezemi 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=.....

6

8. 32-bitový procesor čte data z adres 0x14, 0x94, 0x14, 0x94, 0x114, 0x14

Nechť jeho cache má jenom **128 byte** a je organizovaná jako 2-cestná částečně asociativní o délce řádku **1 slovo**. Používá se v ní nahrazovací algoritmus LRU.

Napište cache miss a nakreslete finální stav ovlivněných setů cache a jaké adresy budou v nich načtené spolu s jejich tagy.

10

9. Prémiová otázka, u níž se uzná jen plně funkční řešení: Vytvořte 3bitový synchronní Johnsonův čítač, který s každou náběžnou hranou clk provede operaci: při RESET='1' má Q="000".
Při RESET='0' a DN=0, čítá vpřed, tedy má výstupy Q = 000,001, 011,111,110, 100, 000, 001... atd.
Je-li RESET='0' a DN=1, čítá stejnou posloupnost opačným směrem.

```
library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity JC3 is port (clk, DN, RESET: in std_logic;
                     Q:out std_logic_vector(2 downto 0));
end entity;
architecture rtl of JC3 is
```

```
end architecture;
```