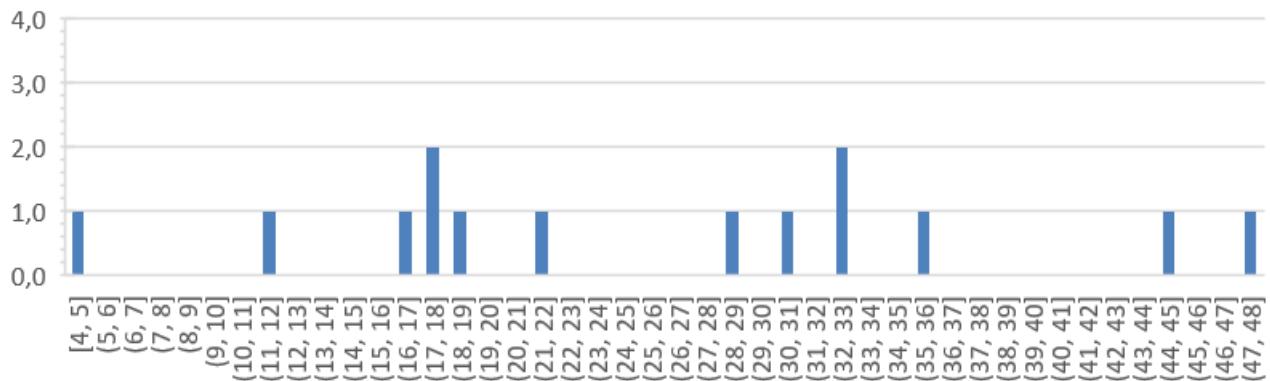
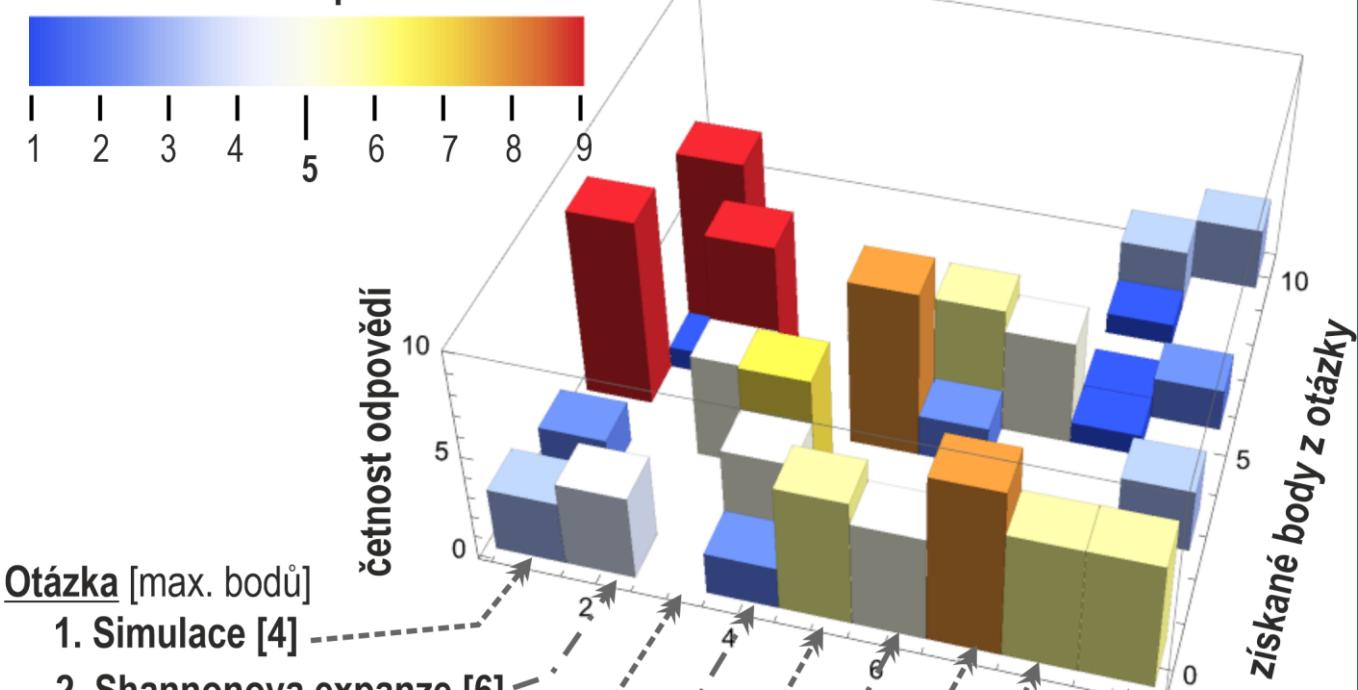


Statistické výsledky zkoušky LSP ve středu 5. června 2024

Četnost bodů z písemky



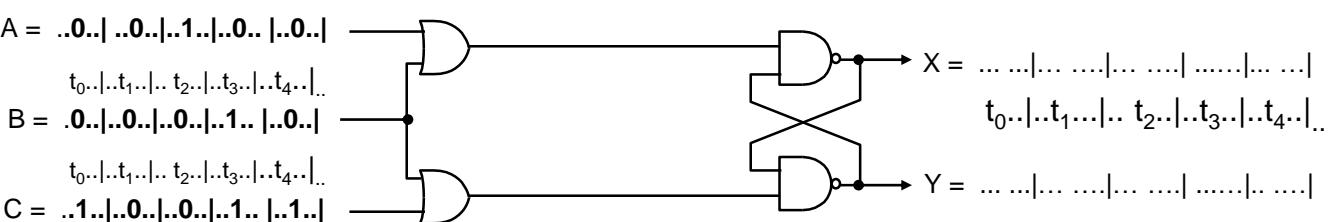
četnost odpovědí



Otázka [max. bodů]

1. Simulace [4]
2. Shannonova expanze [6]
3. Shoda funkcí [4]
4. Signed/Unsigned [2]
5. Úplná sčítáčka [4]
6. Sčítáčka +1 [5]
7. Převodník na Grayův kód ve VHDL [5]
8. Kruhový posuvný registr [10]
9. Paměť cache [10]

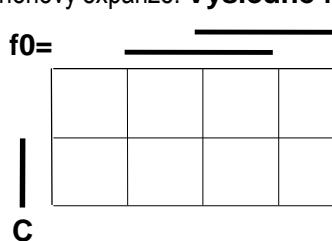
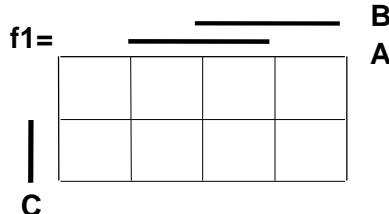
1. Vstupy A, B, C měly v časech t_0, t_1, t_2, t_3 hodnoty uvedené v obrázku. Napište hodnoty X a Y výstupů. Předpokládejte, že intervaly mezi změnami vstupů jsou tak dlouhé, že lze zanedbat zpoždění hradel.



4

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

2. Funkci $X=f(A,B,C)$ 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:**

B
AB
A

6

4

2

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

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

y1

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

y2

y3<=(not A and not C) xor (A and not D)

y3

y4<=(A xnor C) and (not C or not D)

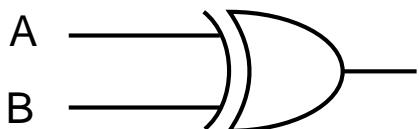
y4

4. Uložíme-li dolní bity výsledku násobení dekadických čísel **4*1023** do 10bitového registru,

jaká hodnota v něm bude, bereme-li ji uvažovat jako 10bitové dekadické číslo

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

5. Student neuměl a nedokončil schéma úplné jednobitové sčítacky. Dokreslete její zapojení.

Carry_{in} —

→ S

→ Carry_{out}

4

5

5

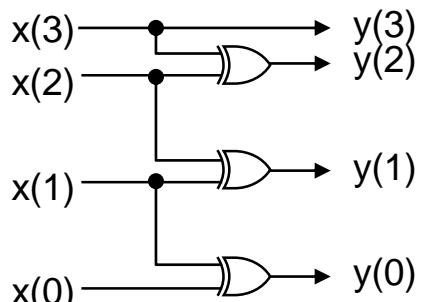
6. Sčítáčka +1 se dá zapojit bez užití úplné sčítáčky mnohem jednodušej pomocí hradel.

Nakreslete její schéma:



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.

```
library ieee; use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity Test20240605q7 is port(
```



```
end entity;  
architecture rtl of Test20240605q7 is
```

begin

end architecture rtl;

Příjmení a jméno:.....

Odpovědní arch písemky LPS dne 5. června 2024 - pište sem jen Vaše odpovědi

8. Nakreslete schéma obvodu, který je popsáný následujícím kódem, a uveďte, jak se nazývá:

```
library ieee; use ieee.std_logic_1164.all;
entity Test20240605q8 is port (A, B : in std_logic; C : in std_logic_vector(3 downto 0); D: out std_logic);
end entity;
architecture rtl of Test20240605q8 is
begin process(A)
variable rg:std_logic_vector(C'RANGE);
begin
if rising_edge(A) then
    if B='1' then rg:=C; else rg:= rg(0) & rg(3 downto 1); end if;
    end if;
    D<=rg(0);
end process;
end architecture;
```

schéma

8

název

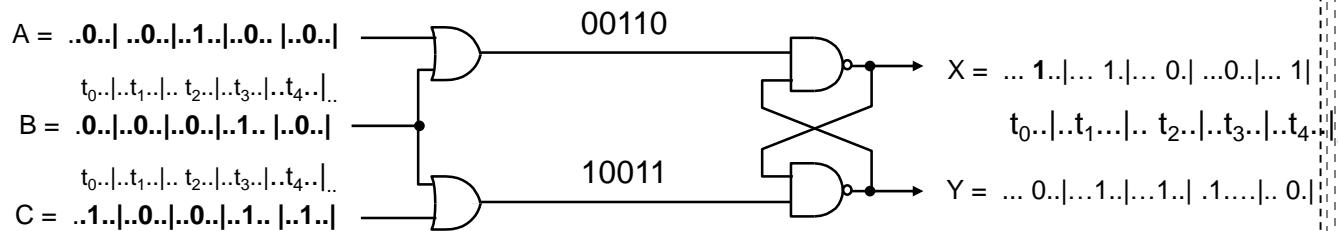
2

9. Malý 32-bitový procesor má cache dlouhou jen **256 bytů a** organizovanou jako **přímo mapovanou** o délce řádku **4 slova**. Označte přístupy do paměti, které budou mít cache hit, budou-li provedené přesně v pořadí uvedeném v tabulce dole:

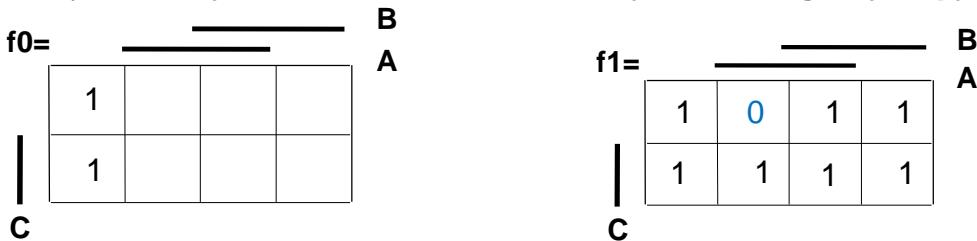
hex. adresa	cache hit
0x14	<input type="checkbox"/>
0x18	<input type="checkbox"/>
0x2C	<input type="checkbox"/>
0x24	<input type="checkbox"/>
0x118	<input type="checkbox"/>
0x10	<input type="checkbox"/>
0x11C	<input type="checkbox"/>
0x110	<input type="checkbox"/>

Řešení

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.

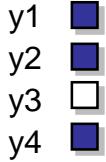


2. Funkci $X=f(A,B,C)$ obvodu z otázky 1, rozložte na tvar $X= (\text{not } X \text{ and } f_0(A,B,C)) \text{ or } (X \text{ and } f_1(A,B,C))$ pomocí Shannonovy expanze. **Výsledné funkce f_0 a f_1 zapište jako Karnaughovy mapy:**



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

- y1<=(not A and not C) or (A and C and not D)
- y2<=(not A or C) and (not A or not D) and (A or not C)
- y3<=(not A and not C) xor (A and not D)
- y4<=(A xnor C) and (not C or not D)



4. Uložíme-li dolní bity výsledku operace s čísly **4*1023** do 10bitového registru,

jaká hodnota v něm bude, bereme-li ji jako 10bitové dekadické číslo

a) bez znaménka (*unsigned*). $4*(2^{10}-1)=(2^{12}-4) \rightarrow$ uloží se jen dolní bity jako $(2^{10}-4)=1020$

b) se znaménkem ve druhém doplňku (*signed*) 1023 je -1, a tak.. $4^*-1= -4$

5. Plná sčítáčka Logické obvody na FPGA str. 92 či přednáška 11

6. Sčítáčka +1 Logické obvody na FPGA str. 97 a určitě i někde v přednáškách

7. Správné řešení

```
library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity Test20240605q7 is port( x: in std_logic_vector(3 downto 0); y:out std_logic_vector(3 downto 0)); end entity;
architecture rtl of Test20240605q7 is
begin
    y<= ('0'& x(3 downto 1)) xor x;
end architecture rtl;
```

Tolerované varianty, pokud měly i správné pořadí operací, u sloučení & na něm záleží:

$$\begin{aligned} y &\leq x(3) \& (x(3) \text{ downto } 1) \text{ xor } x(2 \text{ downto } 0); \\ y &\leq x(3) \& (x(3) \text{ xor } x(2)) \& (x(2) \text{ xor } x(1)) \& (x(1) \text{ xor } x(0)); \end{aligned}$$

8. Posuvný registr - či přednáška 11 snímek 29 a 30

Řešení

9. Malý 32-bitový procesor má cache dlouhou jen **256 bytů a** organizovanou jako **přímo mapovaná** o délce řádku **4 slova**. Označte přístupy do paměti, které budou mít cache hit, budou-li provedené přesně v pořadí uvedeném v tabulce

hex. adresa	cache hit	hex. adresa	cache hit
0x14	<input type="checkbox"/>	0x14	miss-první v setu 1 - načte se řádek, tag=0
0x18	<input checked="" type="checkbox"/>	0x18	hit
0x2C	<input type="checkbox"/>	0x2C	miss - první v setu 2, načte se řádek, tag=0
0x24	<input checked="" type="checkbox"/>	0x24	hit - blok v setu 2
0x118	<input type="checkbox"/>	0x118	miss - nesouhlasí tag, naplněn set 1, tag=1
0x10	<input type="checkbox"/>	0x10	miss - opět přepsaný set 1, tag=0
0x11C	<input type="checkbox"/>	0x11C	miss - zase přepsaný set 1, tag=1
0x110	<input checked="" type="checkbox"/>	0x110	hit - čtení ze setu 1 data, tag=1

Vysvětlení: Slovo 32bitového procesoru má 4 byty.

Rádek načítá (dle zadání) 4 slova, má tedy 16 bytů a vždy se naplní celý.

$256/16 = 16$, cache má tedy 16 setů. U přímo mapované cache má každý set právě 1 řádek.

Adresy se mapují takto: Dolní 2 bity (1 a 0) indexují byte ve slově.

Další **2 bity** (3 a 2) patří indexu slova v řádku. Následující **4 bity** (7 až 4) definují index setu.

Zbylé horní bity označují tag rozlišující obsah setu.

Metodika hodnocení: Za správné označení +3 body, za chybné -3 body.

Kdo neoznačil nic, má 0. Při všech 3 správných +1 prémiový;

Mapování adres:		přímo mapovaná		
			3-bity	1 bit
		tag	set	blok
0x10	00...0001 0000	0	1	0
0x14	00...0001 0100	0	1	1
0x18	00...0001 1000	0	1	2
0x24	00...0010 1100	0	2	3
0x2C	00...0010 0100	0	2	1
0x110	001 0001 0000	1	1	0
0x118	001 0001 1000	1	1	2
0x11C	001 0001 1100	1	1	3