

Zkouska LSP 2022–01–19

Course: B0B35LSP – Logické systémy a procesory | BE5B35LSP – Logic Systems and Processors **University:** CVUT FEL (CVUT) – České vysoké učení technické v Praze | Czech Technical University in Prague **Keywords:** LSP, Exam, Zkouska, 2022–01–19, truth table, De Morgan, RS latch, VHDL

[CN Version](#)

| [EN Version](#)

| [CZ Version](#)

Zkouska LSP 2022–01–19

Al odvozená verze –V PDF nejsou oficiální odpovědi; níže jsou odvozené poznámky/rěšení.

Informace o zkousce

- Datum: 2022–01–19
 - Jazyk: čeština
 - Celkem: 50 bodů (část 1: 25 b ≥ 9 + část 2: 25 b ≥ 9)
-

Úloha 1 –Pravdivostní tabulka (7 b)

Zadání: Podle schématu napište pravdivostní tabulku funkce Y.

Tip: Lze použít Shannonův rozklad pro zjednodusení výpočtu.

Formát Karnaughovy mapy:

		B				
		A	00	01	11	10
DC	00					
	01					
	11					
	10					

Úloha 2 –De Morganovy zákony (3 b)

Zadání: Prěpíšte logický výraz F tak, aby operátor **not** byl pouze přímo před proměnnými.

Původní tvar:

$$F = \text{not} ((\text{not } A \text{ and not } C) \text{ xor } (A \text{ and } B \text{ and } C))$$

Postup: 1. Rozepiste XOR: $A \oplus B = (A \wedge \overline{B}) \vee (\overline{A} \wedge B)$ 2. Použijte De Morgana: $\overline{A \vee B} = \overline{A} \wedge \overline{B}$

Rěšení (jedna z možností):

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

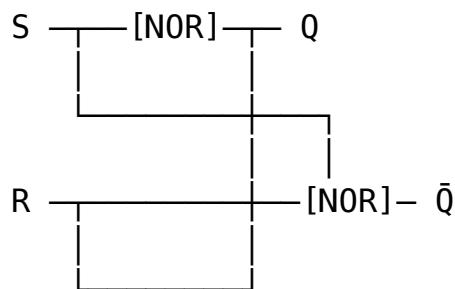
-- zjednodušení:

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

Úloha 3 –Návrh RS záchytu (10 b)

Zadání: Nakreslete RS záchyt pomocí hradel NOR i NAND a doplňte pravdivostní tabulky.

RS záchyt s hradly NOR



Pravdivostní tabulka (NOR): | S | R | Q | Q̄ | | | | | | | 0 | 0 | drž Q | drž Q̄ | 0 | 1 | 0 | 1 | | 1 | 0 | 1 | 1 | 0 | 0 (zakázáno) |

RS záchyt s hradly NAND

Pravdivostní tabulka (NAND): aktivní v nule | S | R | Q | Q̄ | | | | | | | 0 | 0 | 1 | 1 (zakázáno) | | 0 | 1 | 1 | 0 | | 1 | 0 | 0 | 1 | | 1 | 1 | drž Q | drž Q̄ |

Úloha 4 –Definice automatu Moore/Mealy (5 b)

Zadání: Doplňte definici automatu Moore/Mealy.

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

Rěšení: – **X:** konečná vstupní abeceda – **S:** konečná množina stavů – **Z:** konečná výstupní abeceda – **δ:** přechodová funkce (Moore: $S \times X \rightarrow S$, Mealy: $S \times X \rightarrow S$) – **ω:** výstupní funkce (Moore: $S \rightarrow Z$, Mealy: $S \times X \rightarrow Z$) – **s₀:** pocáteční stav

Úloha 5 —Analýza VHDL kódu (10 b)

Zadání: Analyzujte zkrácený VHDL kód a nakreslete schéma.

```
library IEEE; use IEEE.STD_LOGIC_1164.all;
entity zzz is port (a : in std_logic; data : in std_logic; q, qn : out std_
architecture rtl of zzz is
    signal qv : std_logic;
begin
    process(a)
    begin
        if rising_edge(a) then
            qv <= not data;
            q <= qv;
        end if;
        qn <= not qv;
    end process;
end;
```

Funkční analýza: – Spouští se na náběžné hraně a – qv ukládá negovanou hodnotu data – q dává předchozí qv (zpoždění o 1 takt) – qn je negace qv – Chování odpovídá jednoduchému 1bitovému registru/pipeline s inverzí

Úloha 6 —Návrh plného scítacě (3 b)

Zadání: Nakreslete kompletní 1bitový plný scítac se vstupy A, B, Cin a výstupy Y (součet), Cout.

Rovnice:

$$\begin{aligned} \text{Sum} &= A \oplus B \oplus \text{Cin} \\ \text{Cout} &= (A \wedge B) \vee (\text{Cin} \wedge (A \oplus B)) \end{aligned}$$

Úloha 7 —Popis obvodu ve VHDL (7 b)

Zadání: Popisťe dané schéma co nejkratsím VHDL kódem s využitím vektorových operací.

```
library ieee; use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity Zahadum is port(
    -- deklarujte porty dle schématu
);
end entity;
architecture rtl of Zahadum is
```

```

begin
    -- zjednodušení pomocí vektorových operací
end architecture rtl;

```

Úloha 8 –Současný (concurrent) popis ve VHDL (5 b)

Zadání: Analyzujte funkci obvodu z Úlohy 1 a popište ji zjednodusenými concurrent príkazy (ne prímým prepisem rovníc).

```

library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity yyy is port(a,b,c,d: in std_logic; y: out std_logic); end entity;
architecture dataflow of yyy is
begin
    -- zjednodušené concurrent příkazy
end architecture;

```

Úloha 9 –Bonus: Kódovac SOS majáku (10 b)

Zadání: Napište kódovac SOS signálu pomocí pokročilejsích VHDL konstrukcí. Sekvence: 010101000111011101110001010100

```

library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity SOS is
    port (clk : in std_logic;
          X: in std_logic_vector(4 downto 0);
          Morse, STOP: out std_logic);
end entity;
architecture rtl of SOS is
    constant SOS_PATTERN : std_logic_vector(29 downto 0) :=
        "010101000111011101110001010100";
begin
    process(X)
        variable idx : integer;
    begin
        idx := to_integer(unsigned(X));
        if idx < 30 then
            Morse <= SOS_PATTERN(29 - idx);
            STOP <= '0';
        else
            Morse <= '0';
            STOP <= '1';
        end if;
    end process;
end architecture;

```

Shrnutí témat

Hlavní okruhy

1. Pravdivostní tabulky
2. De Morganovy zákony
3. **RS záchyt (NOR a NAND realizace)**
4. Definice automatu Moore/Mealy
5. Analýza VHDL kódu (posuvný registr)
6. Návrh plného scítacě
7. Vektorové operace ve VHDL
8. Zjednodusení obvodu / funkční analýza
9. Kódovac Morse/SOS