

# 2025-06-09 LSP Exam Solution / Řešení zkoušky / 考试解析

**Course:** B0B35LSP – Logické systémy a procesory | BE5B35LSP – Logic Systems and Processors **University:** ČVUT FEL (CTU) – Czech Technical University in Prague **Keywords:** Zkouška, Exam, Test, Solutions, Vysledky, Answers, K-Map, RS Latch, Pipeline

CN Version | EN Version | CZ Version

---

## Zkouška LSP — 2025-06-09

### Informace o zkoušce

- Datum: 2025-06-09
  - Jazyk: čeština (zdroj)
  - Typ: standardní zkouška
  - Oficiální odpovědi ověřeny podle PDF
- 

### Úloha 1 — simulace RS záchytného obvodu (5)

**Zadání:** Vstupy A, B, C mají v časech  $t_0..t_4$  hodnoty podle obrázku. Určete hodnoty výstupů X a Y.

**Sekvence vstupů:**

A	=	1		1		1		0		1
B	=	1		1		0		1		1
C	=	0		1		0		0		0
		$t_0$		$t_1$		$t_2$		$t_3$		$t_4$

**Oficiální odpověď:**

X = 0 | 0 | 1 | 1 | 0 (binárně: 00110)  
Y = 1 | 0 | 0 | 0 | 1 (binárně: 10011)

Poznámka: postupujte krok po kroku přes  $t_0..t_4$  dle RS chování a priority reset/set v konkrétním zapojení na zadání.

---

### Úloha 2 — Shannonův rozvoj (6)

**Zadání:** Rozložte zpětnovazební funkci  $X = f(A, B, C, X)$  z úlohy 1 do tvaru:

$$X = (\neg X \wedge f_0(A, B, C)) \vee (X \wedge f_1(A, B, C)).$$

**Oficiální odpověď (K-map na testu):**

f0:		B				f1:		B			
		A	0	1				A	0	1	
C	0	0	0	1	0	C	0	1	0	1	0
	1	1	0	0	0		1	1	0	1	0

**Oficiální výraz (jak je uvedeno):**

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

---

### Úloha 3 — ekvivalentní logické funkce (4)

**Zadání:** Zakroužkujte všechny logické funkce, které mají v seznamu jinou ekvivalentní funkci.

$y1 \leq (A \text{ or } D) \text{ and } (\text{not } A \text{ or } C);$   
 $y2 \leq C \text{ or } (A \text{ and } C \text{ and } B) \text{ or } (\text{not } A \text{ and } C \text{ and } D);$   
 $y3 \leq (\text{not } A \text{ and } D) \text{ or } (A \text{ and not } D) \text{ or } (C \text{ and } D);$   
 $y4 \leq (C \text{ and } D) \text{ or } (\text{not } A \text{ xor not } D);$

Oficiální odpověď:  $y3 \equiv y4$ .

---

### Úloha 4 — aritmetika v 8bit registru (2)

**Zadání:** Uložte nižší bity součtu  $124 + 125 + 126 + 127$  do 1-bajtového registru. Jaká je decimální hodnota jako 8bitové číslo?

Výpočet:

$124 + 125 + 126 + 127 = 502$   
 $502 \bmod 256 = 246$

Odpověď: – Unsigned: 246 – Signed: -10 (protože  $246 - 256 = -10$ )

---

### Úloha 5 — definice Moore/Mealy automatu (3)

**Zadání:** Doplňte definici (matematicky přesně).

Mooreův (Mealyho) automat je uspořádaná šestice:

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

- $X$  je konečná vstupní abeceda
  - $S$  je konečná množina stavů
  - $Z$  je konečná výstupní abeceda
  - $\delta$  je přechodová funkce:
    - Moore:  $\delta : S \times X \rightarrow S$
    - Mealy:  $\delta : S \times X \rightarrow S$
  - $\omega$  je výstupní funkce:
    - Moore:  $\omega : S \rightarrow Z$
    - Mealy:  $\omega : S \times X \rightarrow Z$
  - $s_0$  je počáteční stav
-

## Úloha 6 — realizace +1 scítačky (7)

**Zadání:** Navrhněte +1 inkrementor pomocí hradel (bez plného scítače).

Typický 4bit inkrementor:  $-s_0 = \neg x_0$   $-s_1 = x_1 \oplus x_0$   $-s_2 = x_2 \oplus (x_0 \wedge x_1)$   $-s_3 = x_3 \oplus (x_0 \wedge x_1 \wedge x_2)$   $-carry = x_0 \wedge x_1 \wedge x_2 \wedge x_3$

---

## Úloha 7 — VHDL bitové operace (7)

**Zadání:** Popište obvod optimálně pomocí konkurenčního VHDL (bez sekvencních příkazů). Plný počet bodů za jediný příkaz v architektuře.

```
library ieee; use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity Test20250609q7 is
    port(x: in std_logic_vector(3 downto 0);
          y: out std_logic_vector(3 downto 0));
end entity;
architecture rtl of Test20250609q7 is
begin
    y <= x and (x(2 downto 0) & '1');
end architecture rtl;
```

---

## Úloha 8 — analýza VHDL obvodu (7+1)

**Zadání:** Nakreslete blokové schéma pro obvod popsany následujícím kódem (hradla, MUX, +1 scítačka, komparátor, DFF).

```
library ieee; use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity Ex20250609q7 is
    port (CLK : in std_logic; Q : out std_logic);
end entity;
architecture rtl1 of Ex20250609q7 is
begin
    iproc: process(CLK)
        constant M: integer := 9;
        variable cntr : integer range 0 to M := 0;
        variable x : std_logic := '0';
    begin
        if rising_edge(CLK) then
            if cntr < M then
                cntr := cntr + 1;
            else
                cntr := 0;
                x := not x;
            end if;
        end if;
        Q <= x;
    end process;
end architecture;
```

**Interpretace:** dělička frekvence / modulo-10 čítač s překlápěním výstupu.

---

## Úloha 9 — predikce skoků (8)

Pozn.: V CN verzi je poznámka, že v kontextu 2026-01 to „není na zkoušku“, ale původní úloha je zde ponechána pro úplnost.

**Zadání:** Pro následující kód (předpoklad: `for` je přeložen jako `do-while`) spočítejte počet chybných predikcí pro uvedené prediktory.

```
int data[] = { 0, 1, -2, -3, 4, -5, 6, -7, -8, 9 };
int min = INT_MAX;
for (int i = 0; i < 10; i++) {
    if (data[i] < min) min = data[i];
}
```

**Oficiální odpověď (podle CN master):** – 1bit prediktor (pocáteční NT): misses = 9 (if 7 + smyčka 2) – 2bit prediktor (pocáteční WT): misses = 7 (if 6 + smyčka 1)

Náčrt průběhu (poznámky):

Branch:	N	T	N	N	T	N	T	N	N	T
Data:	0	1	-2	-3	4	-5	6	-7	-8	9