

2025-06-04 LSP Exam Solution / Řešení zkoušky / 考试解析

Course: B0B35LSP – Logické systémy a procesory | BE5B35LSP – Logic Systems and Processors **University:** ČVUT FEL (CTU) – České vysoké učení technické v Praze **Keywords:** Zkouška, Exam, Test, Solutions, Vysledky, Answers, K-Map, RS Latch, Pipeline

CN Version | EN Version | CZ Version

Informace o zkoušce

- Datum: 2025-06-04
 - Jazyk: CZ (zdroj), oficiální odpovědi ověřeny z PDF
-

Úloha 1 — Simulace RS klopného obvodu (5)

Vstupy v časech $t_0..t_4$:

A	=	1		0		0		1		0
B	=	0		0		0		1		0
C	=	0		0		1		0		1
		t ₀		t ₁		t ₂		t ₃		t ₄

Oficiální odpověď: – X = 01101 (t₀=0, t₁=1, t₂=1, t₃=0, t₄=1) – Y = 11000 (t₀=1, t₁=1, t₂=0, t₃=0, t₄=0)

Rychlá metoda: určete, který vstup řídí **Reset** (zde: A) a který řídí **Set** (zde: B·C), a postupně projděte časové okamžiky (zpoždění hradel zanedbáme).

Úloha 2 — Shannonův rozklad (8)

Rozklad funkce se zpětnou vazbou $X = f(A, B, C, X)$:

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

Oficiální odpověď (K-map na zadání):

f ₀ :		B				f ₁ :		B			
		A	0	1				A	0	1	
C	0	0	1	1	1	C	0	1	1	1	1
	1	0	1	1	1		1	0	1	1	1

Zjednodušený výraz (uvedený ve výsledcích):

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

Postup: spočítejte $f_0 = f(A, B, C, 0)$ a $f_1 = f(A, B, C, 1)$ a minimalizujte pomocí Karnaughových map.

Úloha 3 — Ekvivalentní logické funkce (4)

```
x1 <= (B and not A) or (A and not B);
x2 <= (A and not C) xor (C and A);
x3 <= (B or A) and (not B or not A);
x4 <= (C xor A) or (B and not A);
```

Oficiální odpověď: $x1 \equiv x3$ (obě jsou $A \oplus B$).

Úloha 4 — Aritmetika v 9 bitech (2)

Uložte dolní bity z 4×510 do 9bitového registru.

Oficiální výsledek: – Unsigned: 504 – Se znaménkem (two's complement): –8

Důvod: $-4 \cdot 510 = 4 \cdot (2^9 - 2) = 2^{11} - 8$. – Dolních 9 bitů $\equiv 512 - 8 = 504$. – Se znaménkem: $504 - 512 = -8$.

Úloha 5 — Jednobitový plný scítač (6)

Správné rovnice:

```
Sum = A XOR B XOR Carry_in
Carry_out = (A AND B) OR (Carry_in AND (A XOR B))
```

Úloha 6 — 4:1 multiplexer ve VHDL (4+4)

Současný (concurrent) zápis:

```
qcon <= z when a1='1' else y when a0='1' else x;
```

Sekvenční (process) zápis:

```
process(all)
begin
    if a1='1' then qseq <= z;
    elsif a0='1' then qseq <= y;
    else qseq <= x;
    end if;
end process;
```

Úloha 7 — Analýza posuvného registru (8+2)

```
library ieee; use ieee.std_logic_1164.all;
entity Test20250604q7 is
    port (A, B : in std_logic;
          C : in std_logic_vector(3 downto 0);
          D : out std_logic);
end entity;
```

```

architecture rtl of Test20250604q7 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(2 downto 0) & not rg(3);
      end if;
    end if;
    D <= rg(0);
  end process;
end architecture;

```

Správný název: posuvný registr s invertovanou zpětnou vazbou — **Johnsonův čítač**.

Úloha 8 — Direct-mapped cache (mimo rozsah 2026)

Téma cache je podle rozsahu pro 2026–01 možné přeskocit; ponecháno pouze pro úplnost.
(Podrobná tabulka mapování a hit/miss stopa je v CN verzi.)