

# LSP Zkouška - 15. ledna 2018

```
> **CVUT FEL (ČVUT) - České vysoké učení technické v Praze | Czech Technical University in Prague**
>
> [ ](2018-01-15_Exam_CN.md) | [English](2018-01-15_Exam_EN.md) | [Čeština](2018-01-15_Exam_CZ.md)
> **AI-generované řešení** - Referenční analýza níže
```

---

## Úloha 4: Hodnota se znaménkem/bez znaménka Často testováno

**Úloha:** Jaká je desítková hodnota 10bitového binárního čísla `10 0000 1111`?

a) **Bez znaménka (unsigned):** \_\_\_\_\_

b) **Se znaménkem (two's-complement):** \_\_\_\_\_

### Podrobné řešení

...

Binární: 10 0000 1111  
Pozice: 9 8765 4321 0  
...

**a) Výpočet bez znaménka:**

...

$$= 1 \times 2^9 + 0 \times 2^8 + 0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$
$$= 512 + 0 + 0 + 0 + 0 + 0 + 8 + 4 + 2 + 1$$
$$= 527$$

...

**b) Výpočet se znaménkem (doplňkový kód):**

Metoda 1 - Přímý výpočet:

...

Váha MSB je záporná:  $-2^9 + (\text{hodnota zbývajících bitů})$   
$$= -512 + (0 \times 2^8 + \dots + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0)$$
$$= -512 + 15$$
$$= -497$$

...

Metoda 2 - Inverze a přičtení 1:

...

Původní: 10 0000 1111  
Invertované: 01 1111 0000  
Přičíst 1: 01 1111 0001 = 497  
Tedy původní = -497

...

---

## Úloha 5: Ekvivalentní logické funkce Často testováno

**\*\*Úloha\*\*:** Označte všechny logické funkce, které jsou ekvivalentní s jinými funkcemi:

```
```vhdl
f1 <= (A xor C) or (A and not C);
f2 <= (B or C) and (not A or B or C);
f3 <= ((C and not B) or (B and A));
f4 <= (A or C) and (not A or not C);
f5 <= (A and not B) xor (A and C);
f6 <= (A and not C) or (C and not A);
```
```

### Postup řešení: Karnaughova mapa pro každou funkci

#### f1 = (A xor C) or (A and not C)

| f1  | C=0 | C=1 |
|-----|-----|-----|
| A=0 | 0   | 1   |
| A=1 | 1   | 1   |

f1 = A + C

#### f2 = (B or C) and (not A or B or C)

| f2  | BC=00 | BC=01 | BC=11 | BC=10 |
|-----|-------|-------|-------|-------|
| A=0 | 0     | 1     | 1     | 1     |
| A=1 | 0     | 1     | 1     | 0     |

#### f3 = (C and not B) or (B and A)

| f3  | BC=00 | BC=01 | BC=11 | BC=10 |
|-----|-------|-------|-------|-------|
| A=0 | 0     | 1     | 0     | 0     |
| A=1 | 0     | 1     | 1     | 0     |

#### f4 = (A or C) and (not A or not C)

| f4  | C=0 | C=1 |
|-----|-----|-----|
| A=0 | 0   | 1   |
| A=1 | 1   | 0   |

**\*\*f4 = A C (XOR)\*\***

#### f5 = (A and not B) xor (A and C)

Vyžaduje podrobné rozepsání...

#### f6 = (A and not C) or (C and not A)

| f6  | C=0 | C=1 |
|-----|-----|-----|
| A=0 | 0   | 1   |

```

| A=1 | 1 | 0 |

**f6 = A C (XOR)**

### Závěr
**f4 f6** (obě jsou rovny A XOR C)

---

## Úloha: Simulace RS klopného obvodu Často testováno

**Úloha**: Pro daný obvod napište hodnotu výstupu Q pro vstupy A, B, C v časech t0, t1, t2, t3 jak je u

...
A = ..0../..1../..1../..1../
B = ..0../..0../..0../..1../
C = ..1../..1../..0../..0../

t0  t1  t2  t3
...

### Odpověď
...
Q = ...1.../...0.../...0.../...1.../
...

---

## Úloha: Shannonův rozklad Často testováno

Rozložte funkci na:
...
Q = (not Q and f0(A,B,C)) or (Q and f1(A,B,C))
...

### Postup řešení
...
f0 := f(A,B,C, '0') := (A B) · (B C)
f1 := f(A,B,C, '1') := (A B)
...

### Karnaughova mapa pro f0

f0	C=0	C=1
AB=00	0	1
AB=01	0	0
AB=11	1	0
AB=10	0	0

### Karnaughova mapa pro f1

| f1 | C=0 | C=1 |

```

| AB | 00 | 01 | 11 | 10 |
|----|----|----|----|----|
| 1  | 1  | 0  | 1  | 0  |
| 0  | 0  | 1  | 1  | 0  |

---

## ## Rychlý přehled

### ### Vzorec pro výpočet čísel se znaménkem

- Rozsah n-bitového doplňkového kódu:  $[-2^{n-1}, 2^{n-1}-1]$
- Výpočet záporného čísla:  $-2^{n-1} + (\text{hodnota} \setminus \text{zbývající} \setminus \text{bitů})$

### ### Metoda identifikace ekvivalentních funkcí

1. Nakreslete Karnaughovu mapu pro každou funkci
2. Porovnejte, zda jsou Karnaughovy mapy identické
3. Identické funkce jsou ekvivalentní

### ### Vzorec Shannonova rozkladu

$$f(x_1, x_2, \dots, x_n, Q) = \bar{Q} \cdot f_0 + Q \cdot f_1$$

Kde:

- $f_0 = f(x_1, x_2, \dots, x_n, 0)$
- $f_1 = f(x_1, x_2, \dots, x_n, 1)$