

LSP Průvodce zkouškou (CZ)

Autoři: Správci studijních materiálů LSP

1. Před zkouškou

[HACK] Pravidla přežití u zkoušky

Pamatuj si tři pravidla:

- 1. **Co neumíš, nejdřív přeskoč** a zajisti si snadné body
- 2. **Piš postup pro dílčí body**; i když se spleteš ve výsledku, body za postup často zůstanou
- 3. **Když dochází čas, tipuj**; nechat prázdné je jistá 0

1.1 Rozsah zkoušky (podle učitele)

Povinné vs Nezkoušené

Povinné:

- Výpočty se znaménkem / bez znaménka
- Zjednodušení K-map pro 4 proměnné
- Průběhy RS záchytu
- Shannonův rozvoj
- Hazard v pipeline

Nezkoušené: výpočty cache, prediktor skoků

Zdroj: Email od učitele: “Predikce skoků a cache se nezkouší; bude pipeline.”

1.2 Doporučené pořadí řešení

Doporučené pořadí

- 1. **Číselné úlohy** (15min) - nejjednodušší, vezmi body hned
- 2. **K-Map** (20min) - kreslení, buď pečlivý
- 3. **Pipeline** (20min) - podle šablony
- 4. **RS/Shannon** (25min) - krok za krokem
- 5. **Kontrola** (10min) - ověř výpočty

2. Číselné soustavy (počítej v desítkové)

[HACK] Hlavní myšlenka: spočítej v desítkové, na konci převed'

Nedělej celé výpočty v binární soustavě. Počítej v desítkové a přetečení řeš až v posledním kroku.

2.1 Mocniny dvou (nauč se)

2 ⁴	16	2 ⁸	256	2 ¹²	4096
2 ⁵	32	2 ⁹	512	2 ¹⁴	16384
2 ⁶	64	2 ¹⁰	1024	2 ¹⁶	65536
2 ⁷	128	2 ¹¹	2048	2 ²⁰	1M

2.2 Rozsahy pro N bitů (povinně)

Rychlé poznámky k rozsahu

- **Unsigned:** $[0, 2^N - 1]$
- **Signed:** $[-2^{N-1}, 2^{N-1} - 1]$
- Často používané:**
 - 8-bit signed: $[-128, +127]$
 - 10-bit signed: $[-512, +511]$
 - 16-bit signed: $[-32768, +32767]$

2.3 Přetečení bez znaménka (unsigned)

[HACK] Unsigned přetečení - vezmi modulo

Pravidlo: výsledek = (desítkový výsledek) % 2^N

Příklad: 8-bit unsigned, spočti 200 + 100

- 1. Desítkově: 200 + 100 = 300
- 2. Modulo: 300 mod 256 = 300 - 256 = 44
- 3. **Výsledek: 44**

Hotovo. Není potřeba převádět do binární.

2.4 Přetečení se znaménkem (signed)

[HACK] Signed přetečení - když je to nad maximum, odečti 2^N

Postup ve třech krocích:

- 1. Spočítej desítkový výsledek R
- 2. Ověř: $R > 2^{N-1} - 1$ (max kladné číslo)?
- 3. Pokud ano: $R - 2^N$ je finální výsledek

Příklad: 10-bit signed, spočti 511 + 511

- 1. Desítkově: 511 + 511 = 1022
- 2. Kontrola: 1022 > 511? Ano, přetečení.
- 3. Oprava: 1022 - 1024 = -2
- 4. **Výsledek: -2**

Opačný směr: pokud $R < -2^{N-1}$, přičti 2^N.

2.5 Záporné číslo do dvojkového doplňku

[HACK] Dvojkový doplněk pro záporná čísla - bez invertování + 1

Vzorec: dvojkový doplněk čísla $-X$ je $2^N - X$

Příklad: 8-bit reprezentace -5

- 1. Spočti: 256 - 5 = 251
- 2. Převod do binární: 251 = 128+64+32+16+8+2+1

3. Výsledek: 11111011

Je to rychlejší než "invertuj bity + 1".

2.6 Dvojkový doplněk na desítkovou hodnotu**[HACK] Znaménko urči podle MSB****Podívej se na nejvyšší bit:**

- MSB=0: kladné, převod normálně
- MSB=1: záporné, použij vzorec

Vzorec pro záporné: hodnota = (binární hodnota jako unsigned) -2^N **Příklad:** 8-bit 11110100

1. MSB=1, tedy záporné
2. Jako unsigned: $128 + 64 + 32 + 16 + 4 = 244$
3. Odečti 2^8 : $244 - 256 = -12$
4. **Výsledek:** -12

2.7 Znaménkové rozšíření**[HACK] Rozšíření šířky - zkopíruj znaménkový bit****Unsigned:** doplň 0**Signed:** zopakuj MSB

- Kladné (MSB=0): doplň 0
- Záporné (MSB=1): doplň 1

Příklad: 4-bit \rightarrow 8-bit

- 0110 (+6) \rightarrow 0000 0110
- 1010 (-6) \rightarrow 1111 1010

[!] Neztrať snadné body

- Pozor, zda je zadání signed nebo unsigned
- Pozor na počet bitů (8-bit a 10-bit mají jiné rozsahy)
- Správné mocniny: $2^{10} = 1024$, ne 1000

3. Karnaughova mapa (hledání rozdílů)**[HACK] K-Mapa je jen hra na hledání rozdílů****Tři kroky:**

1. **Vyplň mapu:** podle mintermů vyznač 1
2. **Seskup 1:** zakroužkuj je (čím větší skupina, tím lépe)
3. **Napiš člen:** zapiš proměnné, které se ve skupině nemění

3.1 Šablona pro 4 proměnné (okopíruj)

		CD			
		00	01	11	10
AB	00	0	1	3	2
	01	4	5	7	6
	11	12	13	15	14
	10	8	9	11	10

Pořadí Grayova kódu: 00, 01, 11, 10 (nauč se)

3.2 Jak vyplňovat mapu**[HACK] Kam patří která proměnná?****Oblasti pro ABCD ve 4-proměnné mapě:**

- $A = 1$: spodní dva řádky (řádky 11, 10)
- $B = 1$: prostřední dva řádky (řádky 01, 11)
- $C = 1$: prostřední dva sloupce (sloupce 01, 11)
- $D = 1$: pravé dva sloupce (sloupce 11, 10)

Negace: \bar{A} znamená oblast $A = 0$.**3.3 Seskupování bez přemýšlení****[HACK] Čtyři povinné vzory seskupování****1. Čtyři rohy = $\bar{B}\bar{D}$**

1			1
1			1

2. Celý řádek = mění se jen AB

1	1	1	1

3. Blok 2x2 (4 buňky) = odstraní dvě proměnné**4. Přes okraj (nahoru/dolů nebo vlevo/vpravo) = skupiny mohou přecházet hranice****3.4 Pravidla seskupování****[!] Tři železná pravidla**

1. Velikost skupiny musí být **mocnina dvou**: 1, 2, 4, 8, 16
2. **Čím větší, tím lepší** (odstraní více proměnných)
3. Každá 1 musí být **pokryta alespoň jednou**

3.5 Jak psát výraz ze skupiny

[HACK] Zapiš to, co se ve skupině nemění

Sleduj hodnoty proměnných ve skupině:

- vždy 1 → napiš proměnnou
- vždy 0 → napiš negovanou proměnnou
- někdy 0 někdy 1 → vynech (zruší se)

Příklad: skupina v $AB=01$, $CD=\text{libovolně}$

- A je vždy 0 → napiš \bar{A}
- B je vždy 1 → napiš B
- C a D se mění → vynech
- Výsledek:** $\bar{A}B$

3.6 Tajemství XOR

[HACK] Šachovnice = XOR

Pokud se v K-map střídají 1 a 0 jako na šachovnici:

1	0
0	1

Napiš rovnou: $A \oplus B$ (XOR)

Tady to **nejde dál zjednodušit**; neztrácej čas seskupováním.

3.7 Don't care

X můžeš brát jako 1 nebo 0

- Pokud zvětší skupinu → ber jako 1
- Jinak → ber jako 0 a ignoruj

Cíl: udělat skupiny co největší.

4. RS záchyt (pravidlo semaforu)

[HACK] RS s NOR - aktivní v 1

Pomůcka se semaforem:

- $S = 1 \rightarrow Q$ je 1 (Set)
- $R = 1 \rightarrow Q$ je 0 (Reset)
- Obě 0 → **opiš předchozí stav**
- Obě 1 → napiš “zakázáno” / “nestabilní”

[HACK] RS s NAND - aktivní v 0

Je to obráceně!

- $\bar{S} = 0 \rightarrow Q$ je 1
- $\bar{R} = 0 \rightarrow Q$ je 0
- Obě 1 → **opiš předchozí stav**
- Obě 0 → zakázáno

4.1 Tabulka RS (nutno umět)

S	R	Q	Co napsat
0	0	Q	opiš předchozí
0	1	0	napiš 0
1	0	1	napiš 1
1	1	?	napiš “zakázáno”

4.2 Postup na časové průběhy

[HACK] Tři kroky

- Na průbězích S a R **nakresli svislé čáry** v každém bodě změny
- Pro každý interval přečti hodnoty S a R
- Doplň Q podle tabulky

Pomůcka: S vysoko \Rightarrow Q vysoko, R vysoko \Rightarrow Q nízko, obě nízko \Rightarrow opiš!

5. Shannonův rozklad (metoda copy-paste)

[HACK] Shannon - jen dosad' do vzorce

Neodvozuj. Použij šablonu:

$$F = \bar{A} \cdot F_0 + A \cdot F_1$$

Tři kroky:

- F_0 : ve výrazu dej všude $A = 0$
- F_1 : ve výrazu dej všude $A = 1$
- Dosadíš do vzorce

5.1 Praktický příklad

Příklad: $F = AB + \bar{A}C + BC$, rozklad podle A

Krok 1: F_0 ($A=0$)

Dosadíš $A \rightarrow 0$ a $\bar{A} \rightarrow 1$:

$$\begin{aligned} F_0 &= (0)B + (1)C + BC \\ &= 0 + C + BC \\ &= C \quad (\text{absorpce}) \end{aligned}$$

Krok 2: F_1 ($A=1$)

Dosadíš $A \rightarrow 1$ a $\bar{A} \rightarrow 0$:

$$\begin{aligned} F_1 &= (1)B + (0)C + BC \\ &= B + 0 + BC \\ &= B \quad (\text{absorpce}) \end{aligned}$$

Krok 3: dosazení do vzorce

$$F = \bar{A} \cdot C + A \cdot B$$

[HACK] Rychlá absorpce

- $X + XY = X$ (větší člen stačí)
- $X + \bar{X}Y = X + Y$ (doplňkové pravidlo)

5.2 Shannon pro dvě proměnné**[HACK] Rozklad podle AB současně**

Vzorec:

$$F = \bar{A}\bar{B}F_{00} + \bar{A}BF_{01} + A\bar{B}F_{10} + ABF_{11}$$

Postup: dosad $(A,B)=(0,0), (0,1), (1,0), (1,1)$.**5.3 Shannon jako MUX****Shannon = 2:1 multiplexer** $F = \bar{A} \cdot F_0 + A \cdot F_1$ odpovídá:

- výběr = A
- vstup 0 = F_0
- vstup 1 = F_1

[!] Časté chyby

- Když dáš $A = 0$, tak $\bar{A} = 1$ (nepoplést)
- Při zjednodušování nezapomeň na absorpci
- U MUXu platí $I_0 = F_0$ a $I_1 = F_1$ (pořadí je důležité)

6. Pipeline (přirazovačka)**[HACK] Přihrad' klíčová slova k odpovědím**

Když uvidíš klíčové slovo, rovnou napiš odpověď:

Zadání říká	Napiš
Data Hazard	Forwarding
Load-Use	Stall (bublina)
Branch/Jump	Flush (propláchnutí)
Structural	Přidej hardware

6.1 5-stupňová pipeline (nauč se)

Pět fází		
IF → ID → EX → MEM → WB		
IF	Instruction Fetch	načtení instrukce
ID	Instruction Decode	dekódování / čtení registrů
EX	Execute	ALU výpočet
MEM	Memory	přístup do paměti
WB	Write Back	zápis zpět

6.2 Šablona časového diagramu**[HACK] Kresli podle šablony**

C1	C2	C3	C4	C5	C6	C7
I1	IF	ID	EX	MEM	WB	
I2		IF	ID	EX	MEM	WB
I3			IF	ID	EX	MEM
I4				IF	ID	EX

Pravidlo: každá další instrukce je posunutá o jedno políčko doprava

6.3 Tři typy hazardů**Klasifikace hazardů****1. Strukturální hazard:**

- konflikt o HW zdroj (např. jeden port do paměti)
- řešení: přidat hardware

2. Datový hazard:

- pozdější instrukce potřebuje data, která ještě nejsou hotová
- řešení: forwarding nebo stall

3. Řídicí hazard:

- skok/větvení způsobí načtení špatné instrukce
- řešení: flush / predikce

6.4 RAW hazard**[HACK] Detekce RAW**

Sleduj registry:

I1: ADD R1, R2, R3 ; zapisuje R1

I2: SUB R4, R1, R5 ; čte R1 <- RAW!

Test: čte pozdější instrukce registr, který dřívější zapisuje?

Ano → RAW hazard.

6.5 Cesty forwardingu**[HACK] Dvě cesty forwardingu****1. EX/MEM → EX:**

- ALU výsledek z předchozí instrukce rovnou do EX
- řeší 1-cyklový RAW

2. MEM/WB → EX:

- výsledek z instrukce před dvěma cykly do EX
- řeší 2-cyklový RAW

6.6 Load-use musí stallovat

[!] Forwarding nepomůže u load-use

LW R1, 0(R2) ; data jsou až po MEM
ADD R3, R1, R4 ; potřebuje data v EX

Musíš vložit 1 stall (bublinu):

	C1	C2	C3	C4	C5	C6
LW	IF	ID	EX	MEM	WB	
ADD		IF	ID	--	EX	MEM

“--” je bublina/stall.

6.7 Výpočet CPI

[HACK] Vzorec pro CPI

$$\text{CPI} = 1 + \text{míra stallů}$$

Příklad: 30% instrukcí je Load, z nich 50% způsobí stall

$$\text{míra stallů} = 0.3 \times 0.5 = 0.15$$

$$\text{CPI} = 1 + 0.15 = 1.15$$

6.8 Zrychlení

Vzorec zrychlení

$$\text{Speedup} = \frac{nk}{k+n-1} \rightarrow k$$

k = počet stupňů, n = počet instrukcí

Pro velké n se zrychlení blíží k .

[HACK] Typický postup řešení

1. Nakresli časový diagram
2. Najdi hazardy (kontrola registrů)
3. Zakresli forwarding nebo stall
4. Spočítej CPI

7. Tahák (před zkouškou si přečti tuhle stránku)

7.1 Mocniny dvou

2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{16}
16	32	64	128	256	512	1024	65536

7.2 Rozsahy hodnot

Počet bitů	Unsigned	Signed
8	0~255	-128 ~ +127
10	0~1023	-512 ~ +511
16	0~65535	-32768 ~ +32767

7.3 Rychlá kontrola výpočtů

Unsigned přetečení: výsledek $\geq 2^N$

Signed přetečení: pokud výsledek $> 2^{N-1} - 1$, odečti 2^N

Dvojkový doplněk záporného: $2^N - |X|$

Dvojkový doplněk na hodnotu: MSB=1? hodnota -2^N

7.4 Pozice v K-map

	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

Pořadí: 00-01-11-10

Velikosti skupin: 1,2,4,8,16

Rohy se mohou seskupovat!

7.5 RS záchyt

S	R	Q
0	0	Drž
0	1	0
1	0	1
1	1	Zakázáno

Pomůcka: S vysoko \Rightarrow Q vysoko, R vysoko \Rightarrow Q nízko

7.6 Shannonův rozklad

$$F = \bar{A}F_0 + AF_1$$

F_0 : dosad $A = 0$, $\bar{A} = 1$

F_1 : dosad $A = 1$, $\bar{A} = 0$

7.7 Pipeline přiřazovačka

Data Hazard	Forwarding
Load-Use	Stall
Branch	Flush

$\text{CPI} = 1 + \text{míra stallů}$

5 stupňů: IF-ID-EX-MEM-WB

7.8 Booleovská algebra

$\overline{A+B} = \bar{A}\bar{B}$	De Morgan
$\overline{AB} = \bar{A} + \bar{B}$	De Morgan
$A + AB = A$	Absorpce
$A + \bar{A}B = A + B$	Doplňkové pravidlo
$A \oplus B = A\bar{B} + \bar{A}B$	XOR

Hodně štěstí!

2026-01-13 10:00 | KN-A-310

V klidu. Zvládneš to.