

# LSP Průvodce zkouškou (CZ)

Autoři: Správci studijních materiálů LSP

## 1. 考前须知

### [HACK] 考试生存法则

记住三句话：

1. 不会的先跳过，把会的分拿稳
2. 写步骤给分，哪怕算错了过程分也有
3. 时间不够就蒙，空着一定 0 分

### 1.1 考试范围 (老师说的)

#### 必考 vs 不考

必考：

- 有符号/无符号数计算
- K-Map 4 变量化简
- RS 锁存器波形
- 香农展开
- 流水线 Hazard

不考： Cache 计算、分支预测器

来源：老师邮件原话“不考分支预测和 Cache，会考流水线”

### 1.2 答题顺序

#### 推荐顺序

1. 数值题 (15min) - 最简单，先拿稳
2. K-Map (20min) - 画图题，细心点
3. 流水线 (20min) - 套模板
4. RS/香农 (25min) - 按步骤来
5. 检查 (10min) - 检查计算

## 2. 数值系统 (计算器流)

### [HACK] 核心思想：十进制算完再转！

别傻乎乎转二进制算了！

用十进制算，最后一步再处理溢出。

### 2.1 2 的幂次表 (背下来！)

$2^4$	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 N 位数范围 (必背)

#### 范围速记

- Unsigned:  $[0, 2^N - 1]$
- Signed:  $[-2^{N-1}, 2^{N-1} - 1]$

常用：

- 8 位 Signed:  $[-128, +127]$
- 10 位 Signed:  $[-512, +511]$
- 16 位 Signed:  $[-32768, +32767]$

## 2.3 无符号溢出

### [HACK] Unsigned 溢出 - 直接取模！

无脑公式：结果 = (十进制结果) %  $2^N$

例：8 位 Unsigned，计算  $200 + 100$

1. 十进制:  $200 + 100 = 300$
2. 取模:  $300 \bmod 256 = 300 - 256 = 44$
3. 答案: 44

就这么简单，不用转二进制！

## 2.4 有符号溢出

### [HACK] Signed 溢出 - 超了就减！

无脑三步：

1. 先用十进制算出结果  $R$
2. 判断:  $R > 2^{N-1} - 1$  (最大正数)?
3. 如果超了:  $R - 2^N =$  最终答案

例：10 位 Signed，计算  $511 + 511$

1. 十进制:  $511 + 511 = 1022$
2. 判断:  $1022 > 511$ ? 是的！溢出了！
3. 操作:  $1022 - 1024 = -2$
4. 答案: -2

反向也一样：如果  $R < -2^{N-1}$ ，就加  $2^N$

## 2.5 负数转补码

### [HACK] 负数补码 - 不用取反加 1！

Pro Way 公式： $-X$  的补码 =  $2^N - X$

例：8 位表示 -5

1. 计算:  $256 - 5 = 251$
2. 转二进制:  $251 = 128 + 64 + 32 + 16 + 8 + 2 + 1$
3. 答案: 11111011

比“取反加 1”快多了！

## 2.6 补码转十进制

### [HACK] 看 MSB 定正负！

看最高位：

- MSB=0? 正数，直接转
- MSB=1? 负数，用公式

负数公式：值 = 二进制值  $-2^N$

例：8位 11110100

1. MSB=1, 是负数
2. 先当正数算： $128 + 64 + 32 + 16 + 4 = 244$
3. 减去  $2^8$ ： $244 - 256 = -12$
4. 答案： $-12$

## 2.7 符号扩展

### [HACK] 扩展位数 - 抄符号位！

**Unsigned:** 高位全补 0

**Signed:** 高位抄 MSB

- 正数 (MSB=0): 补 0
- 负数 (MSB=1): 补 1

例：4位  $\rightarrow$  8位

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

### [!] 送分题别丢分

- 看清题目问的是 Signed 还是 Unsigned
- 看清位数！8位和10位范围不一样
- 计算器按对！ $2^{10} = 1024$  不是 1000

## 3. 卡诺图 (找茬游戏)

### [HACK] K-Map 就是找茬游戏！

三步走：

1. 填图：看到变量就填 1
2. 圈图：把 1 圈起来 (越大越好)
3. 写式：圈里不变的变量写出来

### 3.1 4 变量 K-Map 模板 (照着画！)

		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

格雷码顺序：00, 01, 11, 10 (背下来！)

## 3.2 填图口诀

### [HACK] 看到变量往哪填？

4 变量 ABCD 的位置：

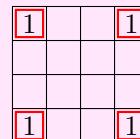
- A = 1: 下两行 (row 11, 10)
- B = 1: 中间两行 (row 01, 11)
- C = 1: 中间两列 (col 01, 11)
- D = 1: 右边两列 (col 11, 10)

Not 变量： $\bar{A}$  就是  $A = 0$  的区域，取反！

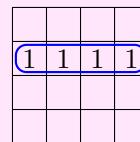
## 3.3 圈图无脑法

### [HACK] 四种必会的圈法

1. 四角圈 =  $\bar{B}\bar{D}$



2. 一整行 = 只有 AB 变



3. 田字格 (4个) = 消两个变量

4. 上下卷/左右卷 = 可以跨边界！

## 3.4 圈图规则

### [!] 圈图三铁律

1. 圈的大小必须是 2 的幂：1,2,4,8,16
2. 圈越大越好 (消去更多变量)
3. 每个 1 至少被圈一次

## 3.5 从圈写表达式

### [HACK] 圈里不变的写出来！

看圈内的变量值：

- 变量全是 1  $\rightarrow$  写变量本身
  - 变量全是 0  $\rightarrow$  写变量的 NOT
  - 变量有 0 有 1  $\rightarrow$  不写 (被消掉了)
- 例：圈在 AB=01, CD= 任意
- A 全是 0  $\rightarrow$  写  $\bar{A}$
  - B 全是 1  $\rightarrow$  写 B
  - CD 有 0 有 1  $\rightarrow$  不写
  - 结果： $\bar{A}B$

## 3.6 异或的秘密

### [HACK] 看到棋盘格 = XOR!

如果 K-Map 里 1 和 0 像国际象棋棋盘一样交错：

1	0
0	1

直接写： $A \oplus B$  (异或)

这种情况没法化简，别浪费时间圈了！

## 3.7 Don't Care

### X 可以当 1 也可以当 0

- 能让圈变大 → 当 1
  - 不能 → 当 0 忽略
- 目标：让圈尽量大！

## 4. RS 锁存器 (交通灯法则)

### [HACK] NOR 门 RS - 高电平有效

红绿灯记忆法：

- $S = 1 \rightarrow Q$  变 1 (Set= 设置)
- $R = 1 \rightarrow Q$  变 0 (Reset= 清零)
- 都是 0 → 抄前一个状态
- 都是 1 → 写“禁止”或“不稳定”

### [HACK] NAND 门 RS - 低电平有效

反过来！

- $\bar{S} = 0 \rightarrow Q$  变 1
- $\bar{R} = 0 \rightarrow Q$  变 0
- 都是 1 → 抄前一个状态
- 都是 0 → 禁止

## 4.1 RS 真值表 (必背)

S	R	Q	怎么写
0	0	Q	抄上一个
0	1	0	写 0
1	0	1	写 1
1	1	?	写“禁止”

## 4.2 波形分析做题法

### [HACK] 波形题三步走

1. 在 S 和 R 波形上画竖线标出每个变化点
2. 每段时间看 S 和 R 的值
3. 按真值表填 Q 的值

口诀：S 高 Q 高，R 高 Q 低，都低抄前面！

## 5. 香农展开 (复制粘贴法)

### [HACK] 香农展开 - 复制粘贴！

不要推导！直接套公式：

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

三步操作：

1.  $F_0$ : 把式子里的 A 全部擦掉换成 0
2.  $F_1$ : 把式子里的 A 全部擦掉换成 1
3. 套公式写答案

## 5.1 香农展开实战

例： $F = AB + \bar{A}C + BC$ , 对 A 展开

Step 1: 算  $F_0$  ( $A=0$ )

把 A 换成 0,  $\bar{A}$  换成 1:

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

Step 2: 算  $F_1$  ( $A=1$ )

把 A 换成 1,  $\bar{A}$  换成 0:

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

Step 3: 套公式

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

### [HACK] 吸收律速记

- $X + XY = X$  (有大的就不要小的)
- $X + \bar{X}Y = X + Y$  (互补相加)

## 5.2 两变量香农展开

### [HACK] 对 AB 同时展开

公式：

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

操作：分别代入  $(A,B)=(0,0), (0,1), (1,0), (1,1)$

## 5.3 香农展开做 MUX

香农 = 2:1 MUX

$F = \bar{A} \cdot F_0 + A \cdot F_1$  直接对应:

- 选择信号 =  $A$
- 输入 0 =  $F_0$
- 输入 1 =  $F_1$

### [!] 香农展开易错点

- $\bar{A}$  代入 0 时变成 1, 别搞反!
- 化简时别忘了吸收律
- MUX 的  $I_0$  接  $F_0$ ,  $I_1$  接  $F_1$ , 顺序别错

## 6.3 三种 Hazard

### Hazard 分类

#### 1. 结构冲突 (Structural):

- 硬件不够用 (如只有一个内存端口)
- 解决: 加硬件

#### 2. 数据冲突 (Data):

- 后面指令要用前面还没算完的数据
- 解决: Forwarding 或 Stall

#### 3. 控制冲突 (Control):

- 分支跳转导致取错指令
- 解决: Flush/预测

## 6. 流水线 (连连看)

### [HACK] 流水线关键词匹配

看到题目关键词, 直接连答案:

题目说	答案写
Data Hazard	Forwarding
Load-Use	Stall (气泡)
Branch/Jump	Flush (冲刷)
Structural	加硬件

## 6.1 5 级流水线 (背下来)

### 五个阶段

IF → ID → EX → MEM → WB

IF	Instruction Fetch	取指令
ID	Instruction Decode	译码/读寄存器
EX	Execute	ALU 计算
MEM	Memory	访存
WB	Write Back	写回

## 6.2 流水线时序图模板

### [HACK] 画图照着抄!

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

规律: 每条指令往右错一格

### [HACK] RAW 冲突检测

看寄存器!

I1: ADD R1, R2, R3 ; 写 R1

I2: SUB R4, R1, R5 ; 读 R1 <- RAW!

判断方法: 后面指令读的寄存器 = 前面指令写的寄存器?

是 → RAW 冲突!

## 6.5 Forwarding 路径

### [HACK] Forwarding 两条路

#### 1. EX/MEM → EX:

- 前一条 ALU 结果直接给后一条用
- 解决 1 周期 RAW

#### 2. MEM/WB → EX:

- 前两条的结果给当前用
- 解决 2 周期 RAW

## 6.6 Load-Use 必须 Stall

### [!] Forwarding 救不了 Load-Use!

LW R1, 0(R2) ; MEM 阶段才有数据

ADD R3, R1, R4 ; EX 阶段就要数据

必须插入 1 个 Stall (气泡)!

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

-- 就是气泡/Stall

## 6.7 CPI 计算

### [HACK] CPI 公式

$$CPI = 1 + \text{Stall 率}$$

例：30% 指令是 Load，其中 50% 造成 Stall

$$\text{Stall 率} = 0.3 \times 0.5 = 0.15$$

$$CPI = 1 + 0.15 = 1.15$$

## 6.8 加速比

### 加速比公式

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

$k$  = 流水线级数,  $n$  = 指令数

当  $n$  很大时, 加速比接近  $k$  倍

### [HACK] 流水线题目套路

1. 画时序图
2. 找冲突（看寄存器）
3. 标 Forwarding 箭头或 Stall
4. 算 CPI

## 7. 作弊条 (考前看这页)

### 7.1 2 的幂次表

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

### 7.2 数值范围

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

### 7.3 数值计算速查

Unsigned 溢出：结果 %  $2^N$

Signed 溢出：结果  $> 2^{N-1} - 1$ ? 减  $2^N$

负数补码： $2^N - |X|$

补码转值：MSB=1? 值  $-2^N$

### 7.4 K-Map 位置

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

顺序：00-01-11-10

圈大小：1,2,4,8,16

四角可以圈！

### 7.5 RS 锁存器

S	R	Q
0	0	保持
0	1	0
1	0	1
1	1	禁止

口诀：S 高 Q 高, R 高 Q 低

### 7.6 香农展开

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

$F_0$ : A 换 0,  $\bar{A}$  换 1

$F_1$ : A 换 1,  $\bar{A}$  换 0

### 7.7 流水线连连看

Data Hazard	Forwarding
Load-Use	Stall
Branch	Flush

CPI = 1 + Stall 率

5 级：IF-ID-EX-MEM-WB

### 7.8 布尔代数

$A + \bar{B} = \bar{A}\bar{B}$	德摩根
$\bar{AB} = \bar{A} + \bar{B}$	德摩根
$A + AB = A$	吸收律
$A + \bar{A}B = A + B$	互补
$A \oplus B = A\bar{B} + \bar{A}B$	异或

Good Luck!

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

兄弟，稳住，你能行的！