LSP Methodology Mastery Guide / Metodika zkoušky (průvodce) / 考试方法论突击指南

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

CN Version | EN Version | CZ Version

Metodika zkoušky LSP (rychlá příprava)

Poznámka: Tato CZ verze je zatím odvozena z CN masteru; některé názvy sekcí a příklady zůstávají dočasně v čínštině.

2026年1月13日考试专用版
根据老师邮件确认：不考分支预测和Cache，会考流水线！

考试范围确认（来自老师邮件）

必考内容

题型	SEL参考页	重要程度
有符号/无符号数位运算	p.2-3	
等价逻辑函数（用卡诺图！）	p.4-7	
RS锁存器仿真 + Shannon展开	p.11-14	
处理器流水线	课件	

不考内容（老师明确说明）

- 跳转预测器 (Jump Predictors)
- Cache未命中计算 (Cache Misses)
- SEL第17页及之后的内容

核心考点模块

模块1：RS锁存器仿真 (RS Latch Simulation)

核心概念 (Concept)

RS锁存器 (RS Latch / Set-Reset Latch) 是一种基本的时序电路，具有两个稳定状态。

LSP考试中的标准电路结构：

```
A      NOR        X




         NOR        Y
B·C
```

工作原理： – **X = NOR(A, Y)** — A是Reset信号 – **Y = NOR(B·C, X)** — B·C是Set信号 – 当 **A=1** 时：X被强制为0（Reset） – 当 **B=1 且 C=1** 时：Y被强制为0，X变为1（Set）

做题"大招" (Step-by-Step Method)

黄金法则：逐时刻分析，不要跳步！

步骤：

1. **画出时序表格**

   ```
       | A | B | C | B·C | X | Y |
   -----|---|---|---|-----|---|---|------
   t0   |   |   |   |     |   |   |
   t1   |   |   |   |     |   |   |
   ...
   ```

2. **确定初始状态 t0**
   - 先计算 B·C
   - 如果 A=1 → X=0（Reset优先）
   - 如果 B·C=1 且 A=0 → Y=0, X=1（Set）
   - 如果 A=0 且 B·C=0 → 保持状态或根据电路初态

3. **逐时刻迭代**
   - 对于每个时刻：
     - 先看 A：A=1 必然导致 X=0
     - 再看 B·C：B·C=1 且 A=0 导致 Y=0
     - 否则保持前一状态

4. **验证公式**

   ```
   X(t) = NOR(A(t), Y(t-1))  //  Y
   Y(t) = NOR(B(t)·C(t), X(t))  // X
   ```

经典例题 (Classic Example)

题目（2025-06-09官方）：

```
A = 1 | 1 | 1 | 0 | 1
B = 1 | 1 | 0 | 1 | 1
C = 0 | 1 | 0 | 0 | 0
    t0  t1  t2  t3  t4
```

官方答案：

```
X = 1 | 1 | 0 | 0 | 0
Y = 0 | 0 | 1 | 0 | 1
```

详细解析：

时刻	A	B	C	B·C	分析	X	Y
t0	1	1	0	0	A=1但电路初态X=1(根据官方答案)	**1**	**0**
t1	1	1	1	1	B·C=1但A=1也是1, A优先Reset但X保持	**1**	**0**
t2	1	0	0	0	A=1持续，X终于变0	**0**	**1**
t3	0	1	0	0	A=0, B·C=0, 保持	**0**	需算
t4	1	1	0	0	A=1, Reset	**0**	**1**

注意： 实际解题需要看考卷给出的具体电路图！不同考试电路可能略有不同。

避坑指南 (Common Pitfalls)

1. **不要忘记计算 B·C**
   - B和C需要AND在一起才是Set信号
2. **不要混淆NOR和NAND**
   - NOR：任一输入为1，输出为0
   - NAND：全部输入为1，输出才为0
3. **不要忘记”保持”状态**
   - 当A=0且B·C=0时，电路保持前一状态
4. **记住优先级**
   - 通常Reset (A=1) 优先于Set (B·C=1)

模块2：Shannon展开 (Shannon Expansion)

核心概念 (Concept)

Shannon展开定理 (Shannon Expansion Theorem)：任何布尔函数都可以对某个变量展开为：

$$f(x_1, ..., x_n) = \bar{x_i} \cdot f|_{x_i=0} + x_i \cdot f|_{x_i=1}$$

在LSP考试中的标准形式：

$$X = (\overline{X} \cdot f_0(A, B, C)) + (X \cdot f_1(A, B, C))$$

其中： – **f_0** = 当X=0时，f的值（X的下一个状态） – **f_1** = 当X=1时，f的值（X的下一个状态）

做题”大招” (Step-by-Step Method)

步骤：

1. **从电路图推导X的表达式**

```
X = NOR(A, Y) = NOT(A OR Y) = Ā · Y
Y = NOR(B·C, X) = NOT(B·C OR X) = (B+C) · X
```

2. **将Y代入X的表达式**
   - 得到 X = f(A, B, C, X)

3. **分别令X=0和X=1求f_0和f_1**
    - f_0 = f(A, B, C, 0)
    - f_1 = f(A, B, C, 1)
4. **画卡诺图（必须！）**

```
f0:     B                f1:     B
      A  0  1                  A  0  1

C 0                   C 0
  1                     1
```

5. **从卡诺图写出最简表达式**

经典例题 (Classic Example)

题目（2025-06-09官方答案）：

对于第1题的电路，求Shannon展开。

官方答案：

```
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


X = (not C and B and A) or (C and not B and not A) or (X and not B and not A) or (X and B and A)
```

避坑指南 (Common Pitfalls)

1. **不要跳过卡诺图**
    - 老师明确说：必须用卡诺图！代值法是"地狱之路"！
2. **卡诺图的变量顺序**
    - 注意A和B的位置，不要画反
3. **验证方法**
    - 把求得的f_0和f_1代回去，检查是否正确

模块3：有符号/无符号数运算 (Signed/Unsigned Arithmetic)

核心概念 (Concept)

N位寄存器的表示范围： | 类型 | 范围 | 特殊值 | |——|——|——–| | N位 unsigned | 0 ~ 2^N − 1 | – | | N位 signed | −2^(N−1) ~ 2^(N−1) − 1 | 全1 = −1 |

关键公式： – **溢出处理**：result mod 2^N – **signed转换**：如果 unsigned ≥ 2^(N−1)，则 signed = unsigned − 2^N

做题"大招" (Step-by-Step Method)

步骤（万能解法）：

1. **计算原始结果 R**

```
124 + 125 + 126 + 127 = 502
```

2. **计算 unsigned = R mod 2^N**

```
8 502 mod 256 = 246
9 502 mod 512 = 502
10 502 mod 1024 = 502
```

3. **计算 signed**

```
unsigned   2^(N-1)
  signed = unsigned - 2^N

  signed = unsigned

8  246   128   signed = 246 - 256 = -10
```

经典例题 (Classic Example)

题目（2025-06-09官方）： 124+125+126+127 存入8位寄存器

解答:

```
1 124 + 125 + 126 + 127 = 502
2 502 mod 256 = 246 (unsigned)
3 246   128 signed = 246 - 256 = -10
```

官方答案： – a) unsigned: **246** – b) signed: **–10**

常考数值速查表

运算	位数	原值	mod	unsigned	signed
124+125+126+127	8位	502	256	246	–10
254+255+256+257	9位	1022	512	510	–2
4×1023	10位	4092	1024	1020	–4
4×510	9位	2040	512	504	–8
255+253+251	8位	759	256	247	–9

避坑指南 (Common Pitfalls)

1. **不要忘记mod运算**
   - 不同位数的mod值不同：8位=256，9位=512，10位=1024
2. **signed判断阈值**
   - 8位：128，9位：256，10位：512
3. **特殊记忆**
   - 全1在signed中 = –1
   - 例：1023(10位) = –1, 255(8位) = –1

模块4：等价逻辑函数 (Equivalent Logic Functions)

核心概念 (Concept)

等价逻辑函数：两个布尔表达式在所有输入组合下输出相同。

判定方法：1. **卡诺图法**（老师强烈推荐！） 2. 代值法（老师说这是”地狱之路”）

做题”大招” (Step-by-Step Method)

步骤：

1. **为每个函数画卡诺图**

```
3           4
     B                 CD
    A  0  1            AB  00  01  11  10

C  0                 00
   1                 01
                  11
                      10
```

2. **填写每个格子的值**
   - 根据表达式计算每个输入组合的输出
3. **比较卡诺图**
   - 卡诺图完全相同的函数是等价的

经典例题 (Classic Example)

题目（2025-06-09）：

```
y1 <= (A or D) and (not A or C);
y2 <= C or (A and C and B) or (not A and C and D);
y3 <= (not A and D) or (A and not D) or (C and D);
y4 <= (C and D) or (not A xor not D);
```

解答过程：

化简 y2：

```
y2 = C + ABC + ĀCD
   = C(1 + AB + ĀD)
   = C
```

化简 y3：

```
y3 = ĀD + AD + CD
   = (A   D) + CD
```

化简 y4：

```
y4 = CD + (Ā   D)
   = CD + (A   D)     [  Ā D = A D]
   = (A   D) + CD
```

官方答案：y3 ≡ y4

必背恒等式 (Essential Identities)

恒等式	说明
$\bar{A} \oplus \bar{B} = A \oplus B$	取反后的XOR等于原XOR
$A \oplus B = \bar{A}B + A\bar{B}$	XOR的标准形式
$(A+B)(\bar{A}+\bar{B}) = A \oplus B$	XOR的乘积形式
$A + AB = A$	吸收律
$A + \bar{A}B = A + B$	吸收律变体

避坑指南 (Common Pitfalls)

1. **绝对不要代值验证**
   - 4个变量有16种组合，容易出错
2. **优先化简**
   - 看到 `A + AB` 立即用吸收律
3. **注意XNOR**
   - XNOR(A,B) = NOT(A XOR B) = A⊙B

模块5：Moore/Mealy自动机定义 (FSM Definition)

核心概念 (Concept)

有限状态机 (Finite State Machine, FSM) 六元组定义：

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

符号	含义	英文
X	有限输入字母表	Finite input alphabet
S	有限状态集合	Finite set of states
Z	有限输出字母表	Finite output alphabet
δ	状态转移函数	State transition function
ω	输出函数	Output function
s_0	初始状态	Initial state

Moore vs Mealy 关键区别

特性	Moore	Mealy
δ (转移函数)	S × X → S	S × X → S
ω (输出函数)	**S → Z**	**S × X → Z**
输出依赖	仅状态	状态 + 输入
输出时机	状态变化后	输入到达时
状态数量	通常更多	通常更少

考试标准答题模板

题目： 补全定义（必须数学上精确！）

标准答案：

```
M = < X, S, Z,  ,  , s   S >

X -      (Finite input alphabet)
S -     (Finite set of states)
Z -      (Finite output alphabet)
 -      (State transition function)
    : S × X → S
 -    (Output function)
    Moore:  : S → Z
    Mealy:  : S × X → Z
s  -    s   S (Initial state)
```

避坑指南 (Common Pitfalls)

1. **不要写错 ω 的定义**
   - Moore: S → Z（只看状态）
   - Mealy: S × X → Z（看状态和输入）
2. **记忆口诀**
   - **M**oore = **M**ore states（状态多，输出简单）
   - **M**ealy = **M**ore efficient（状态少，输出复杂）

模块6：+1加法器设计 (Incrementer Design)

核心概念 (Concept)

+1加法器 (Incrementer) 是一种特殊的加法器，只加1，比全加器简单。

核心公式（必背！）：

```
s  = NOT x              (   )
s  = x  XOR x
s  = x  XOR (x  AND x )
s  = x  XOR (x  AND x  AND x )
...
s  = x  XOR (x   AND ... AND x )

Carry = x  AND x  AND x  AND x   ( 1   )
```

做题"大招" (Step-by-Step Method)

记忆规律： – 第0位：直接取反 – 第n位：x XOR (前面所有位的AND)

电路图绘制：

```
x   [NOT]            s

x    [XOR]           s
```

```
x     [XOR]         s


   [AND]

x     [XOR]         s


   [AND]  (x  AND x  AND x )
```

避坑指南 (Common Pitfalls)

1. **不要用全加器**
   - 题目要求"不用全加器"，用XOR和AND即可
2. **注意进位链**
   - 进位是所有低位的AND

模块7：VHDL单行并发语句

核心概念 (Concept)

并发语句 (Concurrent Statement)：在architecture中直接写，不需要process。

常考模式速查

1. Gray码转换器 (Binary to Gray Code):

```vhdl
y <= ('0' & x(3 downto 1)) xor x;
--   y(i) = x(i) XOR x(i+1)
```

2. 条件选择 (Conditional):

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

3. 位操作：

```vhdl
y <= x(0) & x(1) & x(2) & x(3);  --
y <= x(2 downto 0) & x(3);       --
y <= not x;                       --
```

经典例题（2025-06-09官方）

题目： 用单条语句描述电路

官方答案：

```vhdl
y <= ('0' & x(3 downto 1)) xor x;
```

等价写法：

```vhdl
y <= x(3) & (x(3 downto 1) xor x(2 downto 0));
--
y <= x(3) & (x(2) xor x(3)) & (x(1) xor x(2)) & (x(0) xor x(1));
```

模块8：处理器流水线 (Processor Pipelines)　新增必考！

核心概念 (Concept)

流水线 (Pipeline)：将指令执行分成多个阶段，允许多条指令同时执行。

经典5阶段RISC流水线： | 阶段 | 英文 | 说明 | |——|——|——| | IF | Instruction Fetch | 取指令 | | ID | Instruction Decode | 译码 | | EX | Execute | 执行 | | MEM | Memory Access | 访存 | | WB | Write Back | 写回 |

关键概念

1. 流水线冒险 (Pipeline Hazards):

类型	英文	原因	解决方法
结构冒险	Structural Hazard	硬件资源冲突	增加硬件
数据冒险	Data Hazard	数据依赖	前递/暂停
控制冒险	Control Hazard	分支指令	预测/延迟槽

2. 数据冒险类型 (Data Hazard Types): – **RAW** (Read After Write)：最常见 – **WAR** (Write After Read)：反依赖 – **WAW** (Write After Write)：输出依赖

3. 解决方法： – **前递/旁路 (Forwarding/Bypassing)**：直接传递结果 – **暂停/气泡 (Stalling/Bubble)**：插入NOP – **重排序 (Reordering)**：编译器优化

流水线时序图示例

```
          | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
----------|---|---|---|---|---|---|---|
ADD R1,R2 | IF| ID| EX|MEM| WB|   |   |
SUB R3,R1 |   | IF| ID| ID| EX|MEM| WB|  ← 1  R1
MUL R4,R5 |   |   | IF| IF| ID| EX|MEM|
```

避坑指南 (Common Pitfalls)

1. **理解CPI (Cycles Per Instruction)**
   - 理想流水线CPI = 1
   - 实际因冒险而增加
2. **流水线加速比**
   - 理想加速比 = 流水线级数
   - 实际受冒险和开销影响

考前最后检查清单

必会公式

```
RS  X = NOR(A, Y), Y = NOR(B·C, X)
```

```
Shannon  X = X̄·f  + X·f
unsigned signed     2^(N-1)   2^N
XOR  Ā   B = A   B
+1   s =NOT x , s =x  XOR (  )
Gray y = ('0' & x(N-1 downto 1)) xor x
Moore   : S → Z
Mealy   : S × X → Z
```

必记数值

```
8  unsigned 0~255, signed -128~127
9  unsigned 0~511, signed -256~255
10  unsigned 0~1023, signed -512~511
 1 = -1 (signed)
```

考试技巧

```
RS
     8 /9 /10
   signed
```

祝考试顺利！Good luck!

最后更新：2026年1月13日 考试当天