

LSP 考试终极突击教材 (Exam Methodology Mastery)

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： - $X = \text{NOR}(A, Y)$ - A 是 Reset 信号 - $Y = \text{NOR}(B \cdot C, X)$ - 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

时 刻	A	B	C	$B \cdot C$	X	Y	分 析
t0							
t1							
...							

2. 确定初始状态 t0

- 先计算 $B \cdot C$
- 如果 $A=1 \rightarrow X=0$ (Reset 优先)
- 如果 $B \cdot C=1$ 且 $A=0 \rightarrow Y=0, X=1$ (Set)
- 如果 $A=0$ 且 $B \cdot C=0 \rightarrow$ 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 $B \cdot C$: $B \cdot C=1$ 且 $A=0$ 导致 $Y=0$
 - 否则保持前一状态

4. 验证公式

$$X(t) = \text{NOR}(A(t), Y(t-1)) \quad // \text{ 注意 } Y \text{ 用的是前一时刻的值}$$

$$Y(t) = \text{NOR}(B(t) \cdot C(t), X(t)) \quad // \text{ } X \text{ 用的是当前时刻的值}$$

经典例题 (Classic Example)

题目 (2025-06-09 官方):

$$\begin{array}{l} A = 1 \mid 1 \mid 1 \mid 0 \mid 1 \\ B = 1 \mid 1 \mid 0 \mid 1 \mid 1 \\ C = 0 \mid 1 \mid 0 \mid 0 \mid 0 \\ t0 \quad t1 \quad t2 \quad t3 \quad t4 \end{array}$$

官方答案:

$$\begin{array}{l} X = 1 \mid 1 \mid 0 \mid 0 \mid 0 \\ Y = 0 \mid 0 \mid 1 \mid 0 \mid 1 \end{array}$$

详细解析:

时 刻	A	B	C	$B \cdot C$	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	$B \cdot C=1$ 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	$A=0$, $B \cdot C=0$, 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 $B \cdot C$

- 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 $B \cdot C=0$ 时, 电路保持前一状态
 4. 记住优先级
 - 通常 Reset ($A=1$) 优先于 Set ($B \cdot C=1$)
-

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:

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

在 LSP 考试中的标准形式:

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

其中: - f = 当 $X=0$ 时, f 的值 (X 的下一个状态) - f = 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

$$\begin{aligned} X &= \text{NOR}(A, Y) = \text{NOT}(A \text{ OR } Y) = \bar{A} \cdot \bar{Y} \\ Y &= \text{NOR}(B \cdot C, X) = \text{NOT}(B \cdot C \text{ OR } X) = (B+C) \cdot \bar{X} \end{aligned}$$

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 和 \bar{f}

- $f = f(A, B, C, 0)$
- $\bar{f} = f(A, B, C, 1)$

4. 画卡诺图 (必须!)

$f_0:$	B		$f_1:$	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	A	0
C	0	0	1
1	1	0	0

C	0	0	1	0	C	0	1	0	1	0
1	1	0	0	1	1	0	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 和 f' 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

例：124 + 125 + 126 + 127 = 502

2. 计算 unsigned = R mod 2^N

8 位 : 502 mod 256 = 246

9 位 : 502 mod 512 = 502 (不溢出)

10 位 : 502 mod 1024 = 502 (不溢出)

3. 计算 signed

如果 unsigned < 2^(N-1) :

signed = unsigned - 2^N

否则 :

signed = unsigned

例 (8 位) : 246 < 128, 所以 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:

步骤 1: 124 + 125 + 126 + 127 = 502

步骤 2: 502 mod 256 = 246 (unsigned)

步骤 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: $1023(10 \text{ 位}) = -1$, $255(8 \text{ 位}) = -1$
-

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 变量 :	4 变量 :
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:

$$\begin{aligned} y2 &= C + ABC + \bar{A}CD \\ &= C(1 + AB + \bar{A}D) \\ &= C \end{aligned}$$

化简 y3:

$$\begin{aligned} y3 &= \bar{A}D + AD + CD \\ &= (A \bar{D}) + CD \end{aligned}$$

化简 y4:

$$\begin{aligned} y4 &= CD + (\bar{A} \bar{D}) \\ &= CD + (A \bar{D}) \quad [\text{因为 } \bar{A} \bar{D} = A \bar{D}] \\ &= (A \bar{D}) + CD \end{aligned}$$

官方答案: y3 y4

必背恒等式 (Essential Identities)

恒等式	说明
$\bar{A}B = A\bar{B}$	取反后的 XOR 等于原 XOR
$A\bar{B} = \bar{A}B + AB$	XOR 的标准形式
$(A+B)(\bar{A}+B) = A\bar{B}$	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
 - $XNOR(A,B) = NOT(A \oplus B) = A \bar{B}$

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 状态转移函数	Finite output alphabet State transition function
	输出函数	Output function
s	初始状态	Initial state

Moore vs Mealy 关键区别

特性	Moore	Mealy
(转移函数)	$S \times X \rightarrow S$	$S \times X \rightarrow S$
(输出函数)	$S \rightarrow Z$	$S \times X \rightarrow Z$
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：

$M = < X, S, Z, \delta, \lambda, s_0 >$

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)

- 状态转移函数 (State transition function)

: $S \times X \rightarrow S$

- 输出函数 (Output function)

Moore: : $S \rightarrow Z$

Mealy: : $S \times X \rightarrow Z$

s - 初始状态 $s_0 \in S$ (Initial state)

避坑指南 (Common Pitfalls)

1. 不要写错 的定义

- Moore: $S \rightarrow Z$ (只看状态)
- Mealy: $S \times X \rightarrow Z$ (看状态和输入)

2. 记忆口诀

- Moore = More states (状态多, 输出简单)
- Mealy = More 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          (最低位取反)
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 (全1时产生进位)

做题”大招”(Step-by-Step Method)

记忆规律： - 第 0 位：直接取反 - 第 n 位： $x \text{ XOR } (\text{前面所有位的 AND})$

电路图绘制：

x [NOT] s

x [XOR] s

x [XOR] s

[AND]

x [XOR] s

[AND] ($x \text{ AND } x \text{ AND } x$)

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):

```

y <= ('0' & x(3 downto 1)) xor x;
-- 等价于: y(i) = x(i) XOR x(i+1)

```

2. 条件选择 (Conditional):

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

3. 位操作:

```

y <= x(0) & x(1) & x(2) & x(3);    -- 位翻转
y <= x(2 downto 0) & x(3);        -- 循环左移
y <= not x;                      -- 取反

```

经典例题 (2025-06-09 官方)

题目：用单条语句描述电路

官方答案：

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

等价写法：

```

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

```

指令      | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
-----|---|---|---|---|---|---|---|
ADD R1,R2 | IF| ID| EX|MEM| WB|   |
SUB R3,R1 |   | IF| ID| ID| EX|MEM| WB| ← 暂停 1 周期 (等 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 锁存器 : $X = \text{NOR}(A, Y), Y = \text{NOR}(B \cdot C, X)$

Shannon 展开 : $X = X \cdot f + X \cdot \bar{f}$

unsigned 转 signed : 如果 $2^{N-1} < \text{unsigned}$, 则减 2^N

XOR 恒等式 : $\bar{A} \oplus B = A \oplus B$

+1 加法器 : $s = \text{NOT } x, s = x \oplus \text{进位链}$

Gray 码 : $y = ('0' \& x(N-1 \text{ downto } 1)) \oplus x$

Moore 输出 : $S \rightarrow Z$

Mealy 输出 : $S \times X \rightarrow Z$

必记数值

8 位 范围 : unsigned 0~255, signed -128~127

9 位 范围 : unsigned 0~511, signed -256~255

10 位 范围 : unsigned 0~1023, signed -512~511

全 1 = -1 (signed)

考试技巧

等价函数必须用卡诺图 !

RS 锁存器逐时刻分析, 不跳步 !

注意题目给的位数 (8位/9位/10位)

检查答案 : signed 应该是负数还是正数 ?

祝考试顺利! Good luck!

最后更新: 2026 年 1 月 13 日 考试当天