

# LSP 考试 2025 年 1 月 20 日 (英文版, 含答案)

本文件已核对 PDF 官方答案 (Official Answers Verified from PDF)

## 考试信息

- 日期: 2025 年 1 月 20 日
- 语言: 英语
- 包含官方答案

## 第 1 题 - RS 锁存器仿真 (RS Latch Simulation) (5 分)

题目: 给定输入 A, B, C 在时间 t0-t4 的值, 写出 X 和 Y 输出的值 [English] Given inputs A, B, C values at times t0-t4, write the values of X and Y outputs

输入序列:

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

官方参考答案 (Official Answer): - X = 11001 (t0=1, t1=1, t2=0, t3=0, t4=1) - Y = 01100 (t0=0, t1=1, t2=1, t3=0, t4=0)

补充解析: RS 锁存器 (RS Latch) 分析步骤: 先看 Reset 信号 A, 再看 Set 信号 B·C

## 第 2 题 - Shannon 展开 (Shannon Expansion) (6 分)

题目: 将  $X=f(A,B,C,X)$  分解为 Shannon 展开形式 [English] Decompose  $X=f(A,B,C,X)$  using Shannon expansion

官方答案 (卡诺图 Karnaugh Map):

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

补充解析:  $f = f|_{X=0}$  (X 为 0 时的函数),  $f = f|_{X=1}$  (X 为 1 时的函数)

## 第 3 题 - 等价逻辑函数 (Equivalent Logic Functions) (4 分)

题目: 勾选所有具有等价函数的逻辑函数 [English] Check all logic functions that have an equivalent function

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

官方答案: y1 y4

补充解析: 可用卡诺图或代数化简验证, 两者展开后表达式相同

#### 第 4 题 - 10 位加法器运算 (10-bit Adder Arithmetic) (4 分)

题目:  $1023+1023+1023+1023$  在 10 位加法器上的结果 [English] Result of  $1023+1023+1023+1023$  on a 10-bit adder

计算:  $-1023 = 2^{10} - 1 = 0x3FF$  (10 位全 1) -  $4 \times 1023 = 4092 - 4092 \bmod 1024 = 1020$

官方答案: - a) unsigned:  $(1024-1) \times 4 = 4096-4 = 1020 \pmod{1024}$  - b) signed: 1023 在有符号中表示 -1,  $4 \times (-1) = -4$

补充解析 (Two's Complement): - 10 位全 1 = 1023 (unsigned) = -1 (signed) - 关键: signed 解释下  $1023 = -1$

---

#### 第 5 题 - 全加器设计 (Full Adder Design) (6 分)

题目: 设计一个全加器电路 [English] Design a full adder circuit

全加器公式 (Full Adder Formulas):

$$\text{Sum} = A \oplus B \oplus \text{Cin}$$

$$\text{Cout} = (A \oplus B) \oplus (\text{Cin} \oplus (A \oplus B))$$

补充解析: - Sum 用两级 XOR 门 - Cout = 多数表决器 (Majority Function)

---

#### 第 6 题 - 用 NOR 门实现 XOR (XOR using NOR Gates) (5 分)

题目: 仅使用 NOR 门实现 XOR 功能 [English] Implement XOR using only NOR gates

官方答案:

$$A \text{ xor } B = (A \text{ nor } B) \text{ nor } ((A \text{ nor } A) \text{ nor } (B \text{ nor } B))$$

补充解析 (De Morgan's Theorem): -  $A \text{ nor } A = \text{NOT } A$  - 需要 4 个 NOR 门实现 XOR -  $\text{XOR} = (A+B) \cdot (\bar{A}+\bar{B}) = (A+B) \cdot \text{NOT}(A \cdot B)$

---

#### 第 7 题 - Gray 码转换器 VHDL (Gray Code Converter) (10 分)

题目: 用单个并发语句描述电路 [English] Describe the circuit using a single concurrent statement

官方答案:

```
library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity Test20250120q7 is
    port(x: in std_logic_vector(3 downto 0);
          y: out std_logic_vector(3 downto 0));
end entity;
architecture rtl of Test20250120q7 is
begin
    y <= ('0' & x(3 downto 1)) xor x; -- Binary to Gray Code
end architecture rtl;
```

补充解析 (Gray Code Conversion): - 二进制转格雷码:  $G = B \text{ XOR } (B \gg 1)$  - '0' & x(3 downto 1) = 右移 1 位并在高位补 0 - 格雷码特性: 相邻值只有 1 位不同

---

## 第 8 题 - 分支预测器和 Cache (Branch Prediction & Cache) (10 分)

非考点提示 (Not on Exam): 根据 2026 年 1 月考试说明, 分支预测器和 Cache 相关内容本次不考, 整题可战略性跳过。

题目: 分析程序的分支预测和 Cache miss [English] Analyze the branch prediction and cache miss of the program

程序:

```
int i, j; double arr[2000];
for (i=0; i<5; i++) {
    for (j=0; j<2000; j++) arr[j]++;
}
```

### A) 分支预测器 (Branch Predictor)

题目: 计算 1 位和 2 位分支预测器的 miss 次数 [English] Calculate the number of misses for 1-bit and 2-bit branch predictors

官方答案: - 1 位预测器 (初始 NT): 2 (外循环) +  $5 \times 2$  (内循环) = **12 次 miss** - 2 位预测器 (初始 WT): 1 (外循环) +  $5 \times 1$  (内循环) = **6 次 miss**

补充解析: - 1 位: 每个循环入口 miss + 出口 miss = 2 次 - 2 位: 容忍一次预测错误, 仅出口 miss = 1 次

### B) Cache Miss 计算 (Cache Miss Calculation)

题目: 64 位处理器, 直接映射 (Direct-Mapped), 32KB cache, 块长 4 字 (4-word block) [English] 64-bit processor, direct-mapped, 32KB cache, 4-word block

官方答案: -  $2000 \text{ 个 double} \times 8 \text{ 字节} = 16000 \text{ 字节} < 32\text{KB}$  (可完全放入 cache) - 每块  $4 \text{ 字} \times 8 \text{ 字节} = 32 \text{ 字节} = 4 \text{ 个 double}$  - Cache miss =  $2000 / 4 =$  **500 次 miss**

补充解析 (Cache Calculation Formula):

Miss 次数 = 数组元素数 / 每块容纳的元素数  
=  $2000 / (\text{块大小} / \text{元素大小})$   
=  $2000 / (32/8) = 500$

## 知识点总结

重要答案速记 (高频考点!)

题号	类型	答案
1	RS 仿真	X=00110, Y=10001
3	等价函数	y1 y4
4	<b>10 位运算</b>	<b>unsigned:1020, signed:-4</b>
7	Gray 码 VHDL	$y \leq ('0' \& x(3 \text{ downto } 1)) \text{ xor } x$
8A	分支预测	<b>1 位:12, 2 位:6</b>
8B	Cache	<b>500 次 miss</b>

必背公式

### 10 位有符号/无符号运算

- 1023 在 10 位 unsigned = 1023
- 1023 在 10 位 signed = -1 (因为是全 1)
- $4 \times 1023 \bmod 1024 = 1020$  (unsigned)
- $4 \times (-1) = -4$  (signed)

分支预测 miss 计算 For 循环编译为 do-while:

```
// for (i=0; i<N; i++) body;
// 编译为:
i = 0;
do {
    body;
    i++;
} while (i < N); // 这里是分支
```

**1 位预测器 (初始 NT):** - 循环开始: 预测 NT, 实际 T → **miss**, 变为 T - 循环中: 预测 T, 实际 T → hit - 循环结束: 预测 T, 实际 NT → **miss** - 每个循环 2 次 miss

**2 位预测器 (初始 WT):** - 循环开始: 预测 T, 实际 T → hit - 循环结束: 预测 ST, 实际 NT → **miss** - 每个循环 1 次 miss

Cache Miss 计算

Cache Miss = 数据大小 / 块大小  
= (元素数 × 元素大小) / (字数/块 × 字大小)  
= 元素数 / (字数/块 × 字大小/元素大小)

对于 64 位处理器: - 字大小 = 8 字节 - double = 8 字节 - 每块 4 字 = 32 字节 = 4 个 double - 2000 个 double → 2000/4  
= 500 次 miss