# LSP Exam — January 20, 2025 (Official Answers Included)

**CVUT FEL (CVUT) – České vysoké učení technické v Praze | Czech Technical University in Prague**

中文版 | English | Čeština

**Verified against official PDF answers**

## Exam Information

- Date: 2025年1月20日
- Language: 英语
- Official answers included

---

## Q1 – RS锁存器仿真 (RS Latch Simulation) (5分)

**Problem:** 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

**Input sequence:**

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

**Official answer (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)

**Additional notes:** RS锁存器 (RS Latch) 分析步骤：先看Reset信号A，再看Set信号B·C

---

## Q2 – Shannon展开 (Shannon Expansion) (6分)

**Problem:** 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
```

**Additional notes:** $f_0 = f|_0$（X为0时的函数），$f_1 = f|_1$（X为1时的函数）

---

## Q3 – 等价逻辑函数 (Equivalent Logic Functions) (4分)

**Problem:** 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);
```

**Official answer:** y1 ≡ y4

**Additional notes:** 可用卡诺图或代数化简验证，两者展开后表达式相同

---

## Q4 – 10位加法器运算 (10-bit Adder Arithmetic) (4分)

**Problem:** 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

**Calculation:** – 1023 = 2^10 – 1 = 0x3FF (10位全1) – 4 × 1023 = 4092 – 4092 mod 1024 = 1020

**Official answer:** – a) **unsigned: (1024-1)×4 = 4096-4 ≡ -4 ≡ 1020 (mod 1024)** – b) **signed: 1023在有符号中表示-1，4×(-1) = -4**

**Additional notes (Two's Complement):** – 10位全1 = 1023 (unsigned) = -1 (signed) – 关键：signed解释下 1023 = -1

---

## Q5 – 全加器设计 (Full Adder Design) (6分)

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

**全加器公式 (Full Adder Formulas):**

```
Sum = A   B   Cin
Cout = (A   B)   (Cin   (A   B))
```

**Additional notes:** – Sum用两级XOR门 – Cout = 多数表决器 (Majority Function)

---

## Q6 – 用NOR门实现XOR (XOR using NOR Gates) (5分)

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

**Official answer:**

```
A xor B = (A nor B) nor ((A nor A) nor (B nor B))
```

**Additional notes (De Morgan's Theorem):** – A nor A = NOT A – 需要4个NOR门实现XOR – XOR = (A+B)·(Ā+B̄) = (A+B)·NOT(A·B)

---

## Q7 – Gray码转换器VHDL (Gray Code Converter) (10分)

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

**Official answer:**

```vhdl
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;
```

**Additional notes (Gray Code Conversion):** – 二进制转格雷码：G = B XOR (B ≫ 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

---

## Q8 – 分支预测器和Cache (Branch Prediction & Cache) (10分)

**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。

**Problem:** 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：**

```c
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)

**Problem:** 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

**Official answer:** – **1位预测器（初始NT）**：2（外循环）+ 5×2（内循环）= **12次miss** – **2位预测器（初始WT）**：1（外循环）+ 5×1（内循环）= **6次miss**

**Additional notes:** – 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)

**Problem:** 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

**Official answer:** – 2000个double × 8字节 = 16000字节 < 32KB（可完全放入cache） – 每块4字 × 8字节 = 32字节 = 4个double – Cache miss = 2000 / 4 = **500次miss**

**Additional notes (Cache Calculation Formula):**

```
Miss  =      /
          = 2000 / (  /  )
          = 2000 / (32/8) = 500
```

---

## Key Takeaways

### Quick Answers（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 | **10位运算** | **unsigned:1020, signed:−4** |
| 7 | Gray码VHDL | y <= ('0' & x(3 downto 1)) xor x |
| 8A | 分支预测 | **1位:12, 2位:6** |
| 8B | Cache | **500次miss** |

### Must-Know Formulas

**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）
- 4×1023 mod 1024 = 1020 (unsigned)
- 4×(−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;
//
i = 0;
do {
  body;
  i++;
} while (i < N);  //
```

**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 =    /
           = (   ×   ) / ( /  ×   )
           =    / ( /  ×   /  )
```

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