

LSP Exam 2023–06–23 / Zkouska LSP 2023–06–23 / LSP 考试 2023 年 6 月 23 日

Course: B0B35LSP – Logické systémy a procesory | BE5B35LSP – Logic Systems and Processors **University:** CVUT FEL (ČVUT) – České vysoké učení technické v Praze | Czech Technical University in Prague **Keywords:** LSP, Exam, Zkouska, 2023–06–23, Shannon expansion, branch predictor, cache, VHDL

[CN Version](#)

| [EN Version](#)

| [CZ Version](#)

LSP 考试 2023 年 6 月 23 日

AI 推演版本 – 本试卷 PDF 无官方答案, 以下为推演解析

考试信息

- 日期: 2023 年 6 月 23 日
 - 语言: 捷克语
 - 包含统计数据图表
-

第 1 题 – RS 锁存器仿真 (4 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 =	1		1		0		0		0
B =	0		1		1		1		0
C =	0		0		0		1		1

t0 t1 t2 t3 t4

AI 推演答案 (AI Generated): – t0: A=1 → Reset, X=0, Y=1 – t1: A=1, B·C=0 → Reset 保持, X=0, Y=1 – t2: A=0, B·C=0 → 保持 – t3: A=0, B·C=1 → Set, X=1, Y=0 – t4: A=0, B·C=0 → 保持 – 推测: X=00011, Y=11100 (需根据实际电路验证)

第 2 题 – Shannon 展开 (6 分)

题目: 将 $X=f(A,B,C,X)$ 分解为: [English] Decompose $X=f(A,B,C,X)$ into:

$$X = (\overline{X} \wedge f_0(A, B, C)) \vee (X \wedge f_1(A, B, C))$$

第 3 题 – 等价逻辑函数 (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 <= B or (not A and B and D) or (A and B and C);  
y2 <= (B and D) or (D and not A) or (A and not D);  
y3 <= (A or D) and (B or D or not A);  
y4 <= (not A xor not D) or (B and D);
```

第 4 题 – VHDL 函数编写 (6 分)

题目：编写 VHDL 函数，计算 AS7 类型数组的四舍五入整数平均值。要求最快的硬件计算速度。[English] Write a VHDL function to calculate the rounded integer average of an AS7 type array. Require fastest hardware calculation speed.

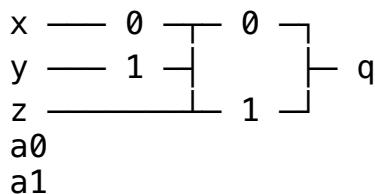
```
type AS7 is array(0 to 6) of integer range -2**15 to 2**15-1;  
  
function meanAS7(x: AS7) return integer is  
    variable sum : integer;  
begin  
    sum := x(0) + x(1) + x(2) + x(3) + x(4) + x(5) + x(6);  
    return (sum + 3) / 7; -- +3 实现四舍五入  
end function;
```

优化技巧: – 并行加法 (树形结构) – 避免除法 (使用乘法和移位近似) – $7 \approx 2^3 - 1$, 可用特殊技巧

第 5 题 – 多路复用器电路实现 (8 分)

题目：仅使用 AND、NAND、OR、NOR 门和反相器实现给定的多路复用器电路 [English] Implement the given multiplexer circuit using only AND, NAND, OR, NOR gates and inverters

符号:



逻辑方程:

$$q = (\text{not } a_1 \text{ and not } a_0 \text{ and } x) \text{ or} \\ (\text{not } a_1 \text{ and } a_0 \text{ and } y) \text{ or} \\ (a_1 \text{ and } z)$$

简化为：

```
q = (not a1 and ((not a0 and x) or (a0 and y))) or (a1 and z)
```

第 6 题 – 多路复用器 VHDL 描述 (6 分)

题目：用最简单的 VHDL 描述第 5 题电路 [English] Describe the circuit from question 5 in simplest VHDL

```
library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity yyy is
    port(x, y, z, a1, a0: in std_logic;
          q: out std_logic);
end entity;
architecture dataflow of yyy is
begin
    q <= z when a1='1' else
        y when a0='1' else
            x;
end architecture;
```

第 7 题 – 分支预测器 (6 分)

非考点提示 (Not on Exam): 根据 2026 年考试说明，分支预测器计算题本次不考，可战略性跳过。

题目：C 程序在数组中查找最小值，假设 for 循环编译为 do-while，计算错误预测次数：[English] C program finds minimum in array, assuming for-loop compiled as do-while, calculate misprediction count:

```
int data[] = { 0, 1, 2, 3, 4, -5, -6, -7, 8, 9 };
int min = INT_MAX;
for (int i = 0; i < sizeof(data)/sizeof(int); i++)
{
    if (data[i] < min) min = data[i];
}
```

答案：– 1 位预测器（初始 Not-Taken）：misses = ? – 2 位预测器（初始 Weakly Taken）：misses = ?

第 8 题 – Cache 计算 (10 分)

非考点提示 (Not on Exam): 根据 2026 年考试说明，Cache 未命中计算题本次不考，可战略性跳过。

题目：32 位处理器，cache 仅 128 字节，2 路组相联，行长 1 字，使用 LRU 替换算法 [English]
32-bit processor, cache only 128 bytes, 2-way set associative, line size 1 word, using LRU replacement

访问序列：0x14, 0x94, 0x14, 0x94, 0x114, 0x14

Cache 参数： – 32 位处理器：字 = 4 字节 – 行长 1 字 = 4 字节 – 128 字节 / 4 字节 = 32 行 –
2 路组相联：32/2 = 16 组 – 地址分解：– Offset: 2 位 (字节偏移) – Set Index: 4 位 (16 组) –
Tag: 剩余位

第 9 题 – 附加题：3 位 Johnson 计数器 (10 分)

题目：实现 3 位同步 Johnson 计数器 [English] Implement a 3-bit synchronous Johnson counter
– RESET='1' 时 Q="000" – DN=0 时正向计数：000,001,011,111,110,100,000... – DN=1 时反向计数

```
library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity JC3 is
  port (clk, DN, RESET: in std_logic;
        Q: out std_logic_vector(2 downto 0));
end entity;
architecture rtl of JC3 is
  signal q_int : std_logic_vector(2 downto 0) := "000";
begin
  process(clk)
  begin
    if rising_edge(clk) then
      if RESET = '1' then
        q_int <= "000";
      elsif DN = '0' then
        q_int <= q_int(1 downto 0) & (not q_int(2));
      else
        q_int <= (not q_int(0)) & q_int(2 downto 1);
      end if;
    end if;
  end process;
  Q <= q_int;
end architecture;
```

知识点总结

本次考试重点

1. RS 锁存器仿真
2. Shannon 展开
3. 等价逻辑函数

4. VHDL 函数编写 (快速平均值)
5. 多路复用器电路实现
6. 多路复用器 VHDL 描述
7. 分支预测器计算 (重要!)
8. Cache 计算 (2 路组相联)
9. Johnson 计数器设计