2025-06-09 LSP Exam Solution / Řešení zkoušky / 考试解析

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

CN Version | EN Version | CZ Version

LSP考试 2025年6月9日

考试信息

- 日期：2025年6月9日
- 语言：捷克语
- 类型：标准考试
- **已核对PDF官方答案**

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

题目： 输入A、B、C在时间t0到t4的值如图所示。写出X和Y输出的值。**[English]** Inputs A, B, C had values shown in the figure at times t0 to t4. Write the values of X and Y outputs.

输入序列:

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

官方参考答案 (Official Answer):

```
X = 0 | 0 | 1 | 1 | 0   (  : 00110)
Y = 1 | 0 | 0 | 0 | 1   (  : 10011)
```

详细解析： – **t0**: A=1 → Reset, X=0; B·C=0 → Y=1 – **t1**: A=1, B·C=1 → Reset优先, X=0, Y=0 – **t2**: A=1, B·C=0 → X变1? (需看电路) – **t3**: A=0, B·C=0 → 保持 – **t4**: A=1 → Reset

---

第2题 – Shannon展开 (Shannon Expansion) (6分)

题目： 将第1题电路的函数X=f(A,B,C,X)用Shannon展开分解为：X=(not X and f0(A,B,C)) or (X and f1(A,B,C))。**[English]** Decompose the function X=f(A,B,C,X) from question 1 into the form X=(not X and f0(A,B,C)) or (X and f1(A,B,C)) using Shannon expansion.

官方参考答案（卡诺图 Karnaugh Map）：

```
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
```

官方表达式：

```
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)
```

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

题目： 勾选所有与其他函数等价的逻辑函数。 **[English]** Check all logic functions that have another equivalent function here.

```
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);
```

官方参考答案：y3 ≡ y4

详细解析：

$$y1 = (A + D)(\bar{A} + C)$$
$$= A\bar{A} + AC + D\bar{A} + DC$$
$$= AC + \bar{A}D + CD$$

y2：

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

y3：

$$y3 = \bar{A}D + AD + CD$$
$$= (A \quad D) + CD \qquad \leftarrow$$

y4：

$$y4 = CD + (\bar{A} \quad D)$$
$$= CD + (A \quad D) \qquad \bar{A} \quad D = A \quad D$$
$$= (A \quad D) + CD$$

比较结果： – y1 = AC + ĀD + CD – y2 = C – y3 = (A ⊕ D) + CD – y4 = (A ⊕ D) + CD

答案：y3 ≡ y4

关键公式： $\bar{A} \quad B = A \quad B$（取反后的异或等于原异或）

第4题 – 8位寄存器运算 (2分)

题目： 将124+125+126+127运算结果的低位存入1字节寄存器，作为8位数的十进制值是多少？ **[English]** If we store the lower bits of 124+125+126+127 operation into a 1–byte register, what decimal value will it hold as an 8–bit number?

计算：

```
124 + 125 + 126 + 127 = 502
502 mod 256 = 246
```

答案： – a) unsigned: **246** – b) signed: **–10** (246 – 256)

第5题 – Moore/Mealy自动机定义 (3分)

题目： 补全定义——必须数学上精确！ **[English]** Complete the definition – it must be mathematically precise!

定义： Automat Moore (Mealy) je usporădaná sěstice M = < X, S, Z, ω, δ, s_0∈S >

- **X** 是有限输入字母表 / is a finite input alphabet
- **S** 是有限状态集合 / is a finite set of states
- **Z** 是有限输出字母表 / is a finite output alphabet
- **δ** 是状态转移函数 / is the state transition function：Moore: S×X→S，Mealy: S×X→S
- **ω** 是输出函数 / is the output function：Moore: S→Z，Mealy: S×X→Z
- **s_0** 是初始状态 / is the initial state

第6题 – +1加法器设计 (7分)

题目： +1加法器可以不用全加器，用门电路更简单地实现。画出由门组成的电路图。 **[English]** The +1 adder can be implemented much simpler using gates without a full adder. Draw its schematic composed of gates.

电路结构： – s0 = x0 XOR 1 = NOT x0 – s1 = x1 XOR (x0) – s2 = x2 XOR (x0 AND x1) – s3 = x3 XOR (x0 AND x1 AND x2) – carry = x0 AND x1 AND x2 AND x3

第7题 – VHDL位操作 (7分)

题目： 用并发（concurrent）VHDL代码最优地描述图中电路，不使用顺序语句。整个架构只用一条语句可得满分，每多一条 **[English]** Optimally describe the circuit in the figure using concurrent VHDL code without sequential statements. Full points for using only one statement in the architecture block; one point deducted for each additional statement.

```
library ieee; use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity Test20250609q7 is
  port(x: in std_logic_vector(3 downto 0);
       y: out std_logic_vector(3 downto 0));
end entity;
architecture rtl of Test20250609q7 is
begin
  y <= x and (x(2 downto 0) & '1');
end architecture rtl;
```

第8题 – VHDL电路分析 (7+1分)

题目： 根据以下代码，使用门、多路选择器、+1加法器、比较器和DFF寄存器的符号画出方框图。 **[English]** Draw the block diagram for the circuit described by the following code, using symbols for gates, multiplexor, +1 adder, comparator, and DFF registers.

```vhdl
library ieee; use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity Ex20250609q7 is
  port (CLK : in std_logic; Q : out std_logic);
end entity;
architecture rtl1 of Ex20250609q7 is
begin
  iproc: process(CLK)
    constant M: integer := 9;
    variable cntr : integer range 0 to M := 0;
    variable x : std_logic := '0';
  begin
    if rising_edge(CLK) then
      if cntr < M then
        cntr := cntr + 1;
      else
        cntr := 0;
        x := not x;
      end if;
    end if;
    Q <= x;
  end process;
end architecture;
```

电路名称： 分频器（Frequency Divider）/ 模10计数器带输出翻转

第9题 – 分支预测 (Branch Prediction) (8分)

非考点提示 (Not on Exam): 根据2026年1月考试说明，分支预测器本次不考，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形式，处理器使用以下预测器，计算分支预测错误次数。
[English] A C program finds the minimum in an array. Assuming the for–loop is compiled as do–while and the processor uses the following predictors, calculate the number of branch mispredictions.

```vhdl
int data[] = { 0, 1, -2, -3, 4, -5, 6, -7, -8, 9 };
int min = INT_MAX;
for (int i = 0; i < 10; i++) {
  if (data[i] < min) min = data[i];
}
```

官方参考答案 (Official Answer): – 1位预测器（初始NT）：misses = **9**（if分支7次 + for循环2次）– 2位预测器（初始WT）：misses = **7**（if分支6次 + for循环1次）

官方分支序列分析：

```
Branch: N  T  N  N  T  N  T  N  N  T
Data:   0  1 -2 -3  4 -5  6 -7 -8  9
```

注意：今年考试不考分支预测！（老师邮件确认）　**注意：** 实际答案可能因编译器优化和分支预测器具体实现而略