2025-06-19 LSP Exam Solution / Řešení zkoušky / 考试解析

Course: B0B35LSP – Logické systémy a procesory | BE5B35LSP – Logic Systems and Processors **University**: CTU 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 Exam — 2025-06-19

Exam info

- Date: 2025-06-19
- Language: Czech (source)
- Type: standard exam
- Official answers verified from PDF

Q1 — RS latch simulation (5)

Task: Inputs A, B, C have the values shown at times $t_0..t_4$. Write the values of outputs X and Y.

Input sequence:

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

Official answer:

```
X = 0 | 0 | 0 | 0 | 1
Y = 1 | 0 | 1 | 1 | 0
```

Q2 — Shannon expansion (6)

Task: Decompose the feedback function $X = f(A, B, C, X)$ into:

$$X = (\neg X \wedge f_0(A, B, C)) \ \vee \ (X \wedge f_1(A, B, C)).$$

Official answer (K-map shown on sheet):

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

Shannon form (as in CN master): - $f_0(A, B, C) = \neg A \wedge B \wedge C$ - $f_1(A, B, C) = \neg A$

So:

```
X = (not X and (not A and B and C)) or (X and (not A))
```

Note: This derivation assumes the same latch equations as on the exam sheet; always follow the official K–map on the paper.

Q3 — Equivalent logic functions (4)

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

Answer: $y3 \equiv y4$.

Q4 — 8–bit register arithmetic (2)

The CN master provides the generic method, but the concrete operands are not included in the text. Use: – Unsigned: R mod 256 – Signed: if $(R \text{ mod } 256) \geq 128$ then subtract 256

Q5 — Moore/Mealy automaton definition (3)

Automat Moore (Mealy) is an ordered 6–tuple:

$$M = \langle X, S, Z, \omega, \delta, s_0 \in S \rangle.$$

- X is a finite input alphabet
- S is a finite set of states
- Z is a finite output alphabet
- δ is the state transition function:
  - Moore: $\delta : S \times X \to S$
  - Mealy: $\delta : S \times X \to S$
- ω is the output function:
  - Moore: $\omega : S \to Z$
  - Mealy: $\omega : S \times X \to Z$
- s_0 is the initial state

Q6 — Incrementer (+1) using gates (7)

For an n–bit incrementer, no full adders are needed; use XOR and AND chains:

- $s_0 = \neg x_0$
- $s_1 = x_1 \oplus x_0$
- $s_2 = x_2 \oplus (x_1 \wedge x_0)$
- $s_3 = x_3 \oplus (x_2 \wedge x_1 \wedge x_0)$
- $C_{out} = x_3 \wedge x_2 \wedge x_1 \wedge x_0$

Q7 — VHDL bit manipulation (7)

The CN master lists common one–line patterns (Gray code, rotate, invert, etc.), but the exact required statement depends on the circuit image on the exam sheet.

Q8 — VHDL circuit analysis (7+1)

The CN master provides a mapping guide (DFF for rising_edge, MUX for selects, incrementer for +1, comparator for < or =). The concrete block diagram depends on the actual VHDL code on the sheet.

Q9 — Branch prediction (8)

The CN master contains a generic branch predictor explanation but no finalized miss counts for this particular instance. If you want, I can compute exact misses once you provide the branch outcome sequence (or the data array and predictor initialization).