2024-12-20 LSP Exam Solution / Řěšení 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

Exam info

- Date: 2024-12-20
- Language: English on the sheet; official answers verified from PDF

Q1 — RS latch simulation (5)

Inputs:

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

Official answer (as noted in the CN source): – X = 00011 (t0=0, t1=0, t2=0, t3=1, t4=1) (the sheet also indicates an alternative depending on the exact latch polarity) – Y = 10110 (t0=1, t1=0, t2=1, t3=1, t4=0) (alternative exists accordingly)

Quick note: A=1 behaves as **Reset**, while B·C=1 behaves as **Set** for this circuit.

Q2 — Shannon expansion (6)

Decompose $X = f(A, B, C, X)$ as:

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

Method: derive f from the circuit, then compute $f_0 = f(\cdot, 0)$ and $f_1 = f(\cdot, 1)$ and minimize via Karnaugh maps.

Q3 — Equivalent logic functions (4)

```
x1 <= (B and not A) or (A and not B);
x2 <= (A and not C) xor (C and A);
x3 <= (B or A) and (not B or not A);
x4 <= (C xor A) or (B and not A);
```

Official answer: $x1 \equiv x3$ (both implement $A \oplus B$).

Q4 — 9-bit adder arithmetic (4)

Compute: 254 + 255 + 256 + 257 = 1022.

- Unsigned (9-bit): $1022 \bmod 512 = 510$
- Signed (two's complement, 9-bit): $510 - 512 = -2$

Official answer: unsigned 510, signed −2.

Q5 — Moore/Mealy FSM definition (6)

For $M = \langle X, S, Z, \omega, \delta, s_0 \rangle$: – X: finite input alphabet – S: finite set of states – Z: finite output alphabet – δ: state transition function – ω: output function – s_0: initial state

Moore vs Mealy output mapping: – Moore: $\omega : S \to Z$ – Mealy: $\omega : S \times X \to Z$

Q6 — Multiplexer gate implementation (5)

Use the standard 2:1 MUX equation:

$$Y = (\neg S \wedge A) \vee (S \wedge B).$$

Then realize it using the allowed gate set (AND/NAND/OR/NOR/NOT) using De Morgan transformations.

Q7 — MUX in VHDL (10)

Official concurrent form:

```
Qcon <= z when a1='1' else y when a0='1' else x;
```

Interpretation: priority-style selection with a1 having higher priority than a0.

Q8 — Direct-mapped cache (10)

The 2026-01 scope notes say cache-miss computations may be skipped; kept here for reference.

Given hit/miss trace (as in CN): – miss: 0x10, 0x28, 0x94, 0xA8 – hit: 0x14, 0x2C, 0x10, 0xAC

Address split (as derived in CN): `[tag][5-bit set][3-bit offset]`.