LSP Exam 2023-06-13

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, Zkouška, 2023-06-13, logic equivalence, arithmetic, RS latch, VHDL

CN Version | EN Version | CZ Version

LSP Exam 2023-06-13

AI-derived version – No official answers in the PDF; the following is an inferred walkthrough/notes.

Exam Info

- Date: 2023-06-13
- Language: Czech
- Contains statistical graphs
- Note: in this exam the RS latch is a 4-input variant (A,B,C,D)

Q1 – Equivalent Logic Functions (4 pts)

Task: Check all logic functions that have an equivalent function:

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

Method: Build Karnaugh maps and compare.

Q2 – 12-bit Adder Arithmetic (2 pts)

Task: Perform 510+512+514+516 on a 12-bit binary adder, result as a 12-bit number.

Computation: – 510+512+514+516 = 2052 – 2052 mod 4096 = 2052 (no overflow) – 2052 = 0x804 = 0b100000000100

Answer: – a) **unsigned**: 2052 – b) **signed (two's complement)**: 2052 (positive, within range)

Q3 – RS Latch Simulation (4 pts)

Task: Given inputs A, B, C, D values at times t0—t4, write the values of outputs X and Y.

Input sequence:

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

Q4 – Shannon Expansion (8 pts)

Task: Decompose $X = f(A, B, C, D, X)$ from question 3 into:

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

Q5 – Moore/Mealy Automaton Definition (4 pts)

Task: Complete the definition.

$M =< X, S, Z, \omega, \delta, s_0 \in S >$

Answer: – **X**: Finite input alphabet – **S**: Finite set of states – **Z**: Finite output alphabet – **δ**: State transition function – Moore: $S \times X \to S$ – Mealy: $S \times X \to S$ – **ω**: Output function – Moore: $S \to Z$ – Mealy: $S \times X \to Z$ – $\mathbf{s_0}$: Initial state

---

Q6 – RS Latch Design (8 pts)

Task: Draw RS latch using NOR gates and NAND gates and complete the truth tables.

Truth table (NOR)

S	R	Q	QN
0	0	hold	hold
0	1	0	1
1	0	1	0
1	1	0	0

Truth table (NAND)

S	R	Q	QN
0	0	1	1
0	1	1	0
1	0	0	1
1	1	hold	hold

Q7 – VHDL Code Analysis (10 pts)

Task: Analyze the code and draw the circuit diagram.

```vhdl
library IEEE; use IEEE.STD_LOGIC_1164.all;
entity xxx is port (a, b : in std_logic; q : out std_logic); end entity;
architecture rtl of xxx is
begin
  process(a,b)
    variable z:std_logic_vector(0 to 3);
  begin
    if rising_edge(b) then
      if a= '0' then
        z:="0001";
      else
        z:=z(1 to 3) & z(0);
      end if;
    end if;
    q<=z(3);
  end process;
end rtl;
```

Functional analysis: – Triggered on rising edge of `b` – When `a='0'` it resets to "0001" – When `a='1'` it rotates left cyclically – Output is `z(3)` – **Name: ring shift register with synchronous reset**

Q8 – VHDL Circuit Description (10 pts)

Task: Describe the given circuit diagram in VHDL.

```vhdl
library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity Test2 is
  port(ACLRN, CLK, X : in std_logic;
       Q, Q2 : out std_logic);
end entity;
```

```vhdl
architecture rtl of Test2 is
  signal q_int : std_logic;
begin
  process(ACLRN, CLK)
  begin
    if ACLRN = '0' then
      q_int <= '0';
    elsif rising_edge(CLK) then
      q_int <= X;
    end if;
  end process;
  Q <= q_int;
  Q2 <= not q_int;
end architecture rtl;
```

Q9 – Bonus: Sawtooth Generator (10 pts)

Task: Implement a 3–bit synchronous counter: 0,1,2,3,4,5,6,7,6,5,4,3,2,1,0,1,…

```vhdl
library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity pila is
  port (clk : in std_logic;
        q: out std_logic_vector(2 downto 0));
end entity;
architecture rtl of pila is
  signal cnt : unsigned(2 downto 0) := "000";
  signal dir : std_logic := '1';
begin
  process(clk)
  begin
    if rising_edge(clk) then
      if dir = '1' then
        if cnt = 7 then
          dir <= '0';
          cnt <= "110";
        else
          cnt <= cnt + 1;
        end if;
      else
        if cnt = 0 then
          dir <= '1';
          cnt <= "001";
        else
          cnt <= cnt - 1;
        end if;
```

```
      end if;
    end if;
  end process;
  q <= std_logic_vector(cnt);
end architecture;
```

Key Topics Summary

Focus points in this exam

1. Equivalent logic functions
2. 12–bit adder arithmetic
3. RS latch simulation (4–input)
4. Shannon expansion
5. **Moore/Mealy automaton definition**
6. RS latch design (NOR and NAND)
7. VHDL code analysis (shift register)
8. VHDL circuit description
9. Sawtooth generator