LSP Exam 2023–07–06

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, 2023–07–06, Shannon expansion, arithmetic, edge detector, VHDL

CN Version | EN Version | CZ Version

LSP Exam 2023–07–06 (English edition)

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

Exam Info

- Date: 2023–07–06
- Language: Czech / English
- Contains statistical graphs

Q1 – Shannon Expansion (6 pts)

Task: Decompose the circuit functions $X, Y = f(A, B, C)$ using Shannon expansion.

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

Q2 – 8–bit Adder Arithmetic (4 pts)

Task: Perform 96+97+98+99 on an 8–bit adder.

Computation: – 96+97+98+99 = 390 – 390 mod 256 = 134 – Binary of 134: 10000110

Answer: – a) **unsigned**: **134** – b) **signed (two's complement)**: **–122** (because MSB is 1)

Q3 – 1:4 Demultiplexer Design (5 pts)

Task: Draw the logic circuit diagram of the demultiplexer.

Logic equations:

```
x0 = Data and (not y1) and (not y0)
x1 = Data and (not y1) and y0
x2 = Data and y1 and (not y0)
x3 = Data and y1 and y0
```

Q4 – RS Latch Simulation (5 pts)

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

Input sequence:

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

Q5 – VHDL RTL Analysis (10 pts)

Task: Draw the RTL view and name the circuit.

```vhdl
library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity XXX is port(a, b : in std_logic; c : out std_logic); end entity;
architecture beh of XXX is
begin
  process (a,b)
    variable d:std_logic:='0';
    variable e:integer range 0 to 25:=0;
  begin
    if b='0' then
      e:=0; d:='0';
    elsif rising_edge(a) then
      if e<25 then
        e:=e+1;
      else
        e:=0; d:=not d;
      end if;
    end if;
    c<=d;
  end process;
end architecture;
```

Answer: – **Name: divide–by–26 frequency divider** – Output toggles once per 26 clock cycles – b is an asynchronous reset

---

Q6 – Demultiplexer VHDL Description (5 pts)

Task: Describe the demultiplexer in VHDL.

```vhdl
library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity yyy is
  port(Data: in std_logic;
       y: in std_logic_vector(1 downto 0);
       x: out std_logic_vector(3 downto 0));
end entity;
architecture dataflow of yyy is
begin
  process(Data, y)
  begin
    x <= (others => '0');
    x(to_integer(unsigned(y))) <= Data;
  end process;
end architecture;
```

Q7 – Divide–by–16 Counter Design (10 pts)

Task: Complete the circuit to implement synchronous divide–by–16 of CLK.

Design: – 4–bit counter (D flip–flops) – +1 adder – DIV16 output is the MSB

Q8 – Branch Predictor (5 pts)

> **Not on Exam note:** According to the 2026 exam notes, branch–predictor calculation tasks are not tested; you can skip strategically.

Task: A C program finds the maximum in an array:

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

Answer placeholders: – 1–bit predictor (initial NT): misses = ? – 2–bit predictor (initial WT): misses = ?

Q9 – Bonus: Edge Detector (10 pts)

Task: When input D changes value, output Y is '1' for one clock cycle.

```vhdl
library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity Edges is
  port (clk, Reset, D : in std_logic; Y: out std_logic);
end entity;
architecture rtl of Edges is
  signal D_reg : std_logic := '0';
begin
  process(clk)
  begin
    if rising_edge(clk) then
      if Reset = '1' then
        D_reg <= '0';
        Y <= '0';
      else
        Y <= D xor D_reg;
        D_reg <= D;
      end if;
    end if;
  end process;
end architecture;
```

Key Topics Summary

Focus points in this exam

1. Shannon expansion
2. **8–bit signed/unsigned arithmetic**
3. Demultiplexer design
4. RS latch simulation
5. **VHDL RTL analysis** (divider)
6. Demultiplexer in VHDL
7. Divide–by–16 design
8. **Branch predictor**
9. Edge detector

Important answers

- 96+97+98+99 = 390 → unsigned: 134, signed: −122
- Divider recognition: /26