LSP Exam - January 25, 2016

> **CVUT FEL (ČVUT) - České vysoké učení technické v Praze | Czech Technical University in Prague**
>
> [ ](2016-01-25_Exam_CN.md) | [English](2016-01-25_Exam_EN.md) | [Čeština](2016-01-25_Exam_CZ.m

> **AI-Generated Solution** - Reference analysis below

Question 5: RS Latch Drawing Frequently Tested

Problem: Draw RS latch using only NOR gates, and draw RS latch using only NAND gates.

NOR-type RS Latch

```
S      [>NOR]      Q
             ↑





R      [>NOR]      Q
             ↑


```

Characteristics:
- S=1, R=0 → Q=1 (Set)
- S=0, R=1 → Q=0 (Reset)
- S=0, R=0 → Hold
- S=1, R=1 → Forbidden state

NAND-type RS Latch

```
S      [>NAND]      Q
             ↑





R      [>NAND]      Q
             ↑


```

Characteristics (active-low):
- S=0, R=1 → Q=1 (Set)

- S=1, R=0 → Q=0 (Reset)
- S=1, R=1 → Hold
- S=0, R=0 → Forbidden state

Question 7: Motor Control Circuit Design

Problem: A student came to the exam just to try. Complete his unfinished circuit - pressing the butt

Hint: Need to add 7 additional components to complete the circuit.

Circuit Components
- VCC = 5V
- GND = 0V
- Relay
- 24V Battery
- Push-button
- D Flip-Flop (DFF)
- Motor

Design Approach
1. Use D flip-flop to implement T flip-flop functionality (toggle state on each button press)
2. Feed Q back to D input
3. Connect button to CLK (rising edge triggered)
4. Connect CLRN to VCC to ensure power-on reset
5. Q output controls the relay

Components to Add
1. Wire from Q to D
2. VCC to PRN connection
3. Button debounce circuit
4. Relay driver circuit
5. Protection diode, etc.

Question 8: VHDL Code Analysis

Problem: Analyze the poorly formatted VHDL code, draw the corresponding logic circuit diagram and g

```vhdl
library IEEE; use IEEE.STD_LOGIC_1164.all;
entity test20140214 is port (a, b, c, d : in std_logic; e : out std_logic); end;
architecture rtl of test20140214 is begin
process(a, b) variable z:std_logic_vector(0 to 3); begin
if b = '0' then z:=(others=>'0'); elsif rising_edge(a) then
if c='1' then z:=d & z(0 to 2); else z:=z(3) & z(0 to 2); end if; end if; e<=z(3); end process; end rtl
```

Formatted Code

```vhdl
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.all;

entity test20140214 is
    port (
        a, b, c, d : in std_logic;
        e : out std_logic
    );
end;

architecture rtl of test20140214 is
begin
    process(a, b)
        variable z: std_logic_vector(0 to 3);
    begin
        if b = '0' then
            z := (others => '0');        -- Async clear
        elsif rising_edge(a) then        -- Rising edge triggered
            if c = '1' then
                z := d & z(0 to 2);      -- Mode 1: Serial input
            else
                z := z(3) & z(0 to 2);   -- Mode 2: Circular shift
            end if;
        end if;
        e <= z(3);                       -- Output MSB
    end process;
end rtl;
```

Circuit Diagram Description

Circuit Name: 4-bit Controllable Dual-Mode Shift Register

```

    d    MUX   → [DFF] → [DFF] → [DFF] → [DFF]   → e
                    z(0)    z(1)    z(2)    z(3)
    c    sel


                    (circular feedback)

    a    CLK (all DFFs)
    b    CLRN (all DFFs)
```

Functional Description

Signal	Function
a	Clock signal (rising edge)
b	Async clear (b='0' clears)
c	Mode select
d	Serial data input
e	Output (z(3))

c value	Mode
'1'	Serial input: d → z(0) → z(1) → z(2) → z(3)
'0'	Circular shift: z(3) → z(0) → z(1) → z(2) → z(3)

Knowledge Summary

Question Type	Key Points
RS Latch	NOR/NAND gate implementation, understanding set/reset logic
Motor Control	D flip-flop application, T flip-flop design
VHDL Analysis	Code formatting, shift register understanding