Exercise 3: Sequential Logic

Prof. Dr. Sven Simon

1 Shift register

1. Given the entity below, implement an 8 bit shift register that is shifting the internal value one to the left every clock cycle when `load` is '0'. If `load` is '1', `data_in` is stored in the internal register. Use a synchronous active high reset.

```
entity lsr is
  port (
    clk      : in  std_logic;
    rst      : in  std_logic;
    load     : in  std_logic;
    data_in  : in  std_logic_vector(7 downto 0);
    data_out : out std_logic_vector(7 downto 0));
end entity lsr;
```

2. Add a clock enable to the entity. When clock enable is asserted, the circuit behaves normally. When not, the internal register is unchanged.

2 Mod-9 counter

Implement a mod-9 counter.

3 Johnson counter

Implement a 4-bit Johnson counter.

4 Gray counter

Implement a 4-bit Gray counter.

5 Universal shift register

A universal shift register is a shift register that can shift right, shift left, or perform a parallel load based on a control signal. Implement one.