Exercises - TOC

Lesson 1 – HW Modeling and Design with VHDL

1.- Using VHDL, code the following simple elements:

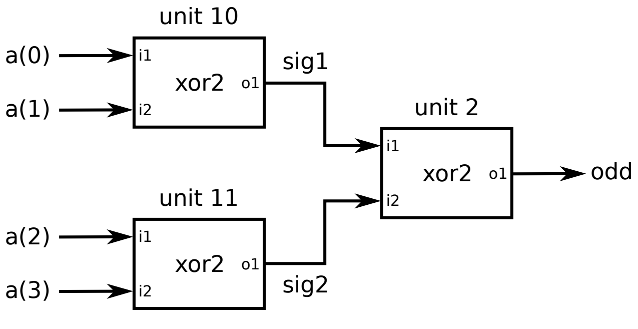
a. An XOR for two 4-bit inputs.
b. A 4:1 multiplexer.
c. A 3:8 decoder.
d. An 8:3 encoder.
e. A comparator for two 4-bit numbers that returns 1 if both of them are the same.

2.- Using VHDL, code the following combinatorial system:

a. Its input is a positive number whose possible values are in the range [0-15].
b. Its output Z is 1 if the input fulfills one of the following conditions, 0 otherwise:
   – It is a prime number.
   – It is lower than 4 and an even number (consider the 0 as even).
   – It is greater than 8 and an odd number.

3.- Using VHDL, code a combinatorial system whose aim is to multiply by 3 a positive number X whose possible values are in the range [0-7]. Its output Z is a positive integer number whose possible values are in the range [0-15]. An additional output D indicates if the multiplication has overflow.

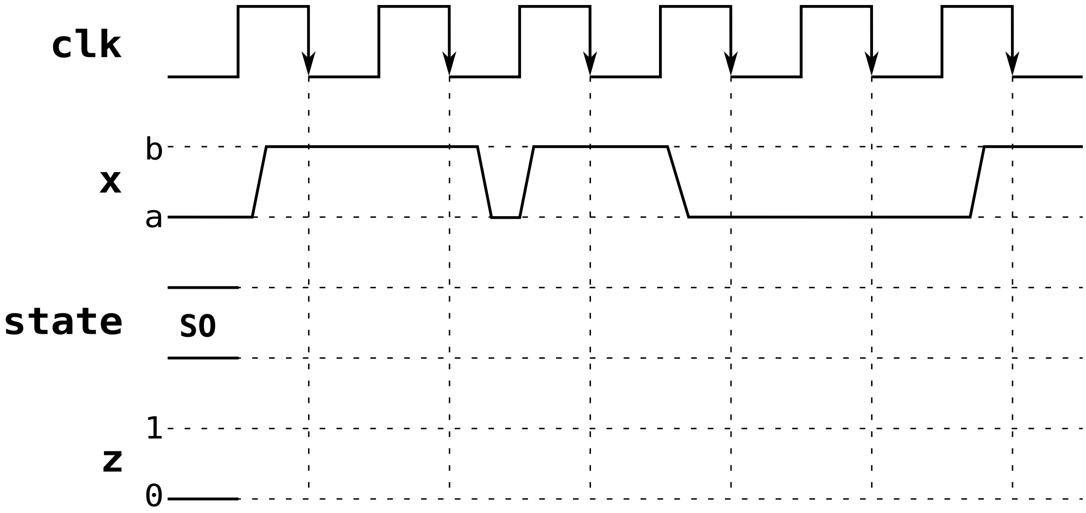
4.- The following RTL scheme shows a circuit composed of XOR gates. Using VHDL, code a structural architecture of this circuit.



5.- Given the following sequential system:

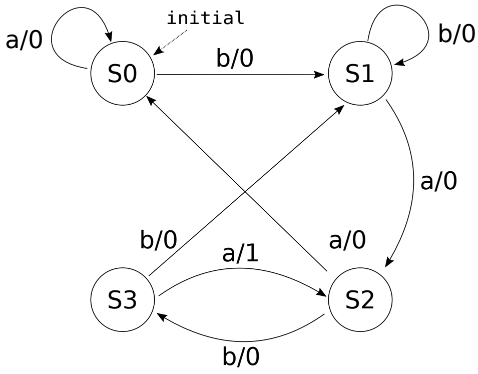
$$z(t) = \begin{cases} \text{'1'} & if\ x(t-2, t-1, t) = \text{'aaa' or 'bbb'}, \\ \text{'0'} & otherwise \end{cases}$$

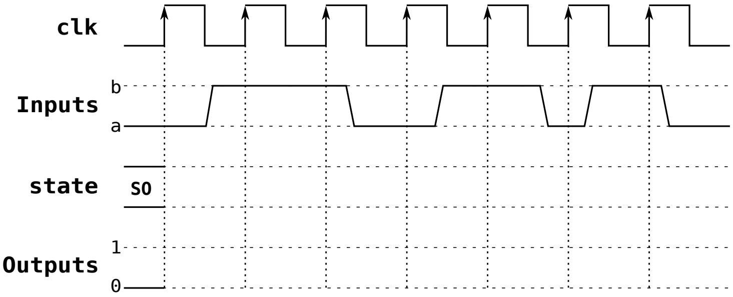
a. Using VHDL, design that system as a Mealy machine.
b. Complete the following schedule:
c. Create a testbench based on this schedule and compare the results. Assume that the clk period is 100 ns.



6.- The following state diagram represents a pattern recognizer.

a. Which pattern does it recognize?
b. Implement this circuit with VHDL, .
c. Complete the schedule. Code a testbench that corresponds to this schedule, assuming that the `clk` frequency is 5 MHz.





7.- Using VHDL, code an 8-bit left shift register with 3 control bits that indicate the number of bits that must be shifted to the left. Assume that the shift must be carried out in just one clock cycle (independently of how many bits must be shifted). NOTE: It is forbidden to use the VHDL instruction SLL.

8.- The following structural VHDL code describes the operation of a circuit. Draw an equivalent RTL description of it.

```
library ieee;
use ieee.std_logic_1164.all;
    entity hundred_counter is
        port (
            clk, rst        : in std_logic;
            en              : in std_logic;
            q_ten, q_one    : out std_logic_vector(3 downto 0);
            p_ten           : out std_logic
        );
end hundred_counter;
```

```
architecture str_arch of hundred_counter is
    component dec_counter
        port (
            clk, rst : in std_logic;
            en       : in std_logic;
            q        : out std_logic_vector(3 downto 0);
            pulse    : out std_logic
        );
    end component;

    signal p_one, p_ten_aux: std_logic;
begin
    one_digit: dec_counter
        port map (clk=>clk, rst=>rst, en=>en,
                  pulse=>p_one, q=>q_one);

    ten_digit: dec_counter
        port map (clk=>clk, rst=>rst, en=>p_one,
                  pulse=>p_ten_aux, q=>q_ten);

    p_ten<=p_ten_aux;
end str_arch;
```

9.- The following VHDL code calculates the parity of the std_logic_vector A. However, it is not synthesizable. Code a synthesizable FSM whose operation is exactly the same:

```
architecture beh2_arch of even_detector is
begin
    process(A)
        variable sum, r: integer;
    begin
        sum := 0;
        for i in 2 downto 0 loop
            if A(i)='1' then
                sum := sum +1;
            end if ;
        end loop ;
        r : = sum mod 2;
        if (r=0) then
            even <='1';
        else
            even <='0';
        end if;
    end process;
end beh2_arch;
```