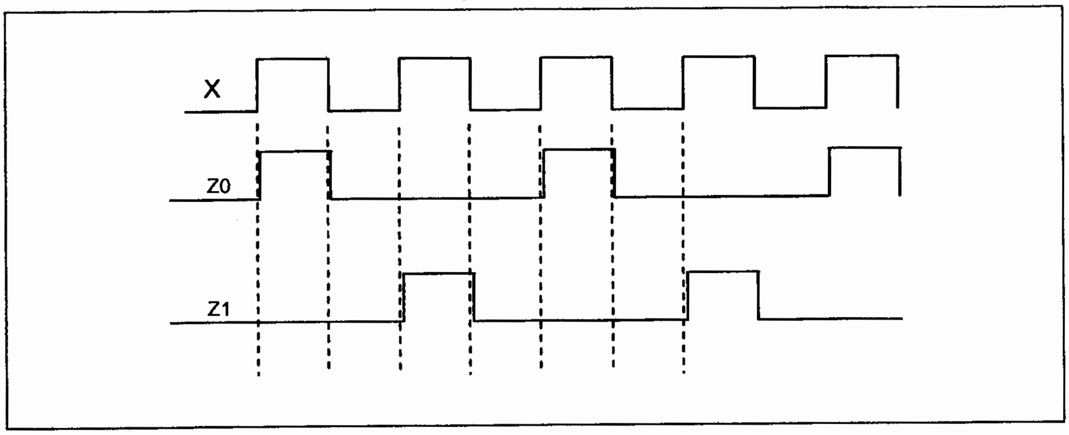
FYSE420 DIGITAL ELECTRONICS

02.07.2010

K.Loberg

1. In picture given below show output sequence of asynchronous circuit. Design circuit and give an **exitation functions, transition table** and **total state table** for an asynchronous circuit. Underscore the stable states in transition table. Avoid hazards and remember a proper binary state assignment. X is input, Z0 and Z1 are outputs. Use only gates.



2. Explain shortly

(a) Mealy state machine. Draw simple block diagram.
(b) Moore state machine. Draw simple block diagram.
(c) One-Hot Encoded state machine.
(d) Registered-Output Finite State Machine
(e) What kind of problems we have in Asynchronous Sequential Circuits.

3. Give a circuit diagram of the VHDL description shown below.

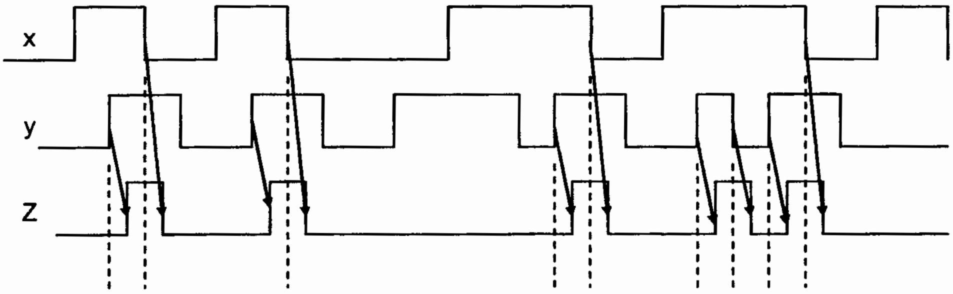
```
entity comblog is port(
                            sel : in integer range 0 to 3;
                            x,y, z : in std_logic;
                            res : out std_logic);
                   end entity comblog;

                   architecture behavior_arch of comblog is
                            begin
                            case sel is
                                     when 0 => res <= x and y;
                                     when 1 => res <= y xor z;
                                     when 2 => res <= x nand z;
                                     when others => res <= x nor z;
                            end case;
                            end process;
                   end architecture behavior_arch;
```

4. Give the circuit diagram of the VHDL description shown below. Assume that you have positive edge triggered D flip-flops, which has also Clock Enable (CE) input. CE is active on level -1-.

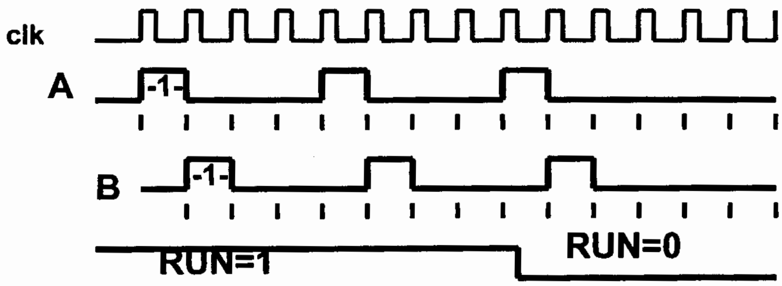
```
library IEEE;
use ieee.std_logic_1164.all;
entity reg is
port (
         ina : in std_logic_vector(7 downto 0);
         enable : in std_logic;
         clk,start: in std_logic;
         com : out std_logic_vector(7 downto 0)
    );
end reg;

architecture reg_arch of reg is
         signal reg_out : std_logic_vector(7 downto 0);
begin
process (clk) is
         begin
         if (clk'event and clk = '1') then
         if enable ='1' then
                  reg_out <= ina;
         end if;
         end if;
end process;

process(reg_out,start) is
         variable varcom : std_logic_vector(7 downto 0);
```

```
begin
        if start = '0' then
                varcom := "00000000";
        else
                varcom := reg_out;
        end if;
        com <= varcom;
end process;
end architecture reg_arch ;
```

5. Design a working asynchronous circuit, which produces same output sequence, given below (use only gates in implementation). Start synthesis from State Diagram and then other tables and K-maps. Give minimized solution without hazards. Inputs are x and y. Output is Z.



6. Design the synchronous **binary encoded** finite state machine (FSM). The external control input is ***RUN***. If ***RUN*** = ***-1-***, the FSM enters to the continuous sequence described below. When ***RUN*** goes to the ***-0-*** level, the **FSM stop to the state where *A* = *B* = *-0-*. If outputs A or B are -1- when RUN goes to the level : RUN=0, the FSM sequence runs normally until *A* = *B* = *-0-* at the same time.** Let the first state be ***a***, when outputs ***A*** = ***-0-*** and ***B*** = ***-0-*** **.** Draw only clock connections between DFFs and give the boolean functions of the exitationfunctions of DFFs. Outputs A and B has own DFFs. The solution is Registered Output FSM, which has binary-encoded states.


clk
A
-1-
B
-1-
RUN=1
RUN=0