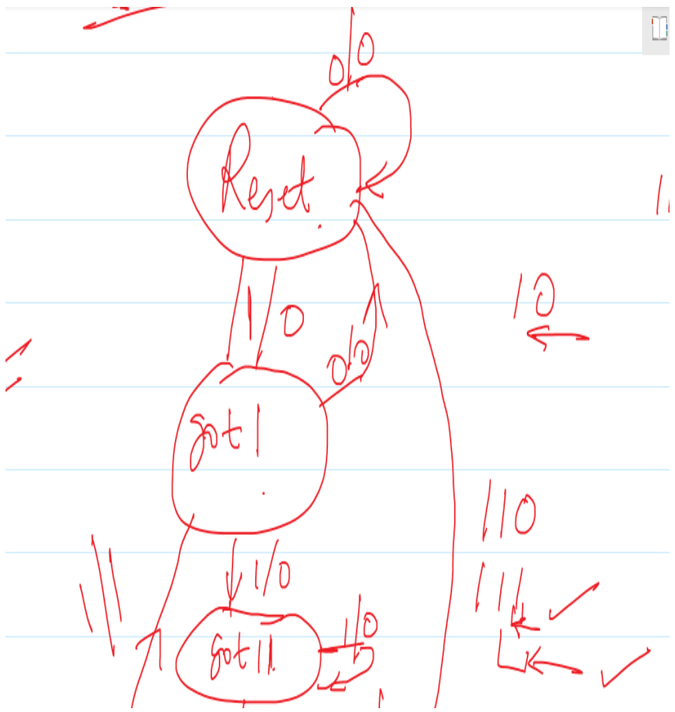
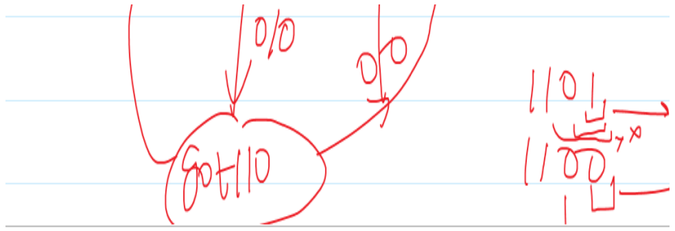
Overlapping Mealy Code

Tuesday, April 20, 2021 1:42 PM

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_arith.all;
use IEEE.STD_LOGIC_unsigned.all;

entity Seq_1101_mealy is
        port(
                clk : in STD_LOGIC;
                rst : in STD_LOGIC;
                X : in STD_LOGIC;
                Z : out STD_LOGIC
                );
end Seq_1101_mealy;

architecture JK_Mealy_arch of Seq_1101_mealy is
        type state is (reset,got1,got11,got110);
        signal pr_state,nx_state : state;

begin
        process (rst,clk)
        begin
                if (rst = '1')then
                        pr_state <= reset;
                elsif (clk'event and clk='0') then
                        pr_state <= nx_state;
                end if;
        end process;

process (pr_state,x)
begin
        case pr_state  is
                when reset =>
                        if (x ='0') then
                                z <='0';
                                nx_state <= reset;
                        else
                                z <='0';
                                nx_state <= got1;
                        end if;

                when got1 =>
                        if (x ='0') then
                                z <='0';
                                nx_state <= reset;
                        else
                                z <='0';
                                nx_state <= got11;
                        end if;

                when got11 =>
                        if (x ='0') then
                                z <='0';
                                nx_state <= got110;
                        else
                                z <='0';
                                nx_state <= got11;
                        end if;
```




```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
---------------------------------------------------
ENTITY <entity_name> IS
   PORT ( input: IN <data_type>;
          reset, clock: IN STD_LOGIC;
          output: OUT <data_type>);
END <entity_name>;
---------------------------------------------------
ARCHITECTURE <arch_name> OF <entity_name> IS
   TYPE state IS (state0, state1, state2, state3, ...);
   SIGNAL pr_state, nx_state: state;
BEGIN
   ---------- Lower section: ------------------------
   PROCESS (reset, clock)
   BEGIN
      IF (reset='1') THEN
         pr_state <= state0;
      ELSIF (clock'EVENT AND clock='1') THEN
         pr_state <= nx_state;
      END IF;
   END PROCESS;
   ---------- Upper section: ------------------------
   PROCESS (input, pr_state)
   BEGIN
      CASE pr_state IS
         WHEN state0 =>
            IF (input = ...) THEN
               output <= <value>;
               nx_state <= state1;
            ELSE ...
            END IF;
         WHEN state1 =>
            IF (input = ...) THEN
               output <= <value>;
               nx_state <= state2;
            ELSE ...
            END IF;
         WHEN state2 =>
            IF (input = ...) THEN
               output <= <value>;
               nx_state <= state3;
            ELSE ...
            END IF;
         ...
      END CASE;
   END PROCESS;
END <arch_name>;
```




```
            when got110 =>
                if (x ='0') then
                    z <='0';
                    nx_state <= reset;
                else
                    z <='0';
                    nx_state <= got1;
                end if;
        end case;
    end process;
end JK_Mealy_arch;
```