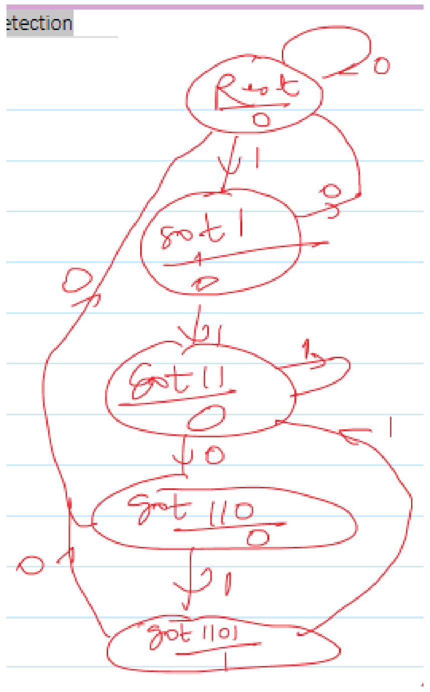
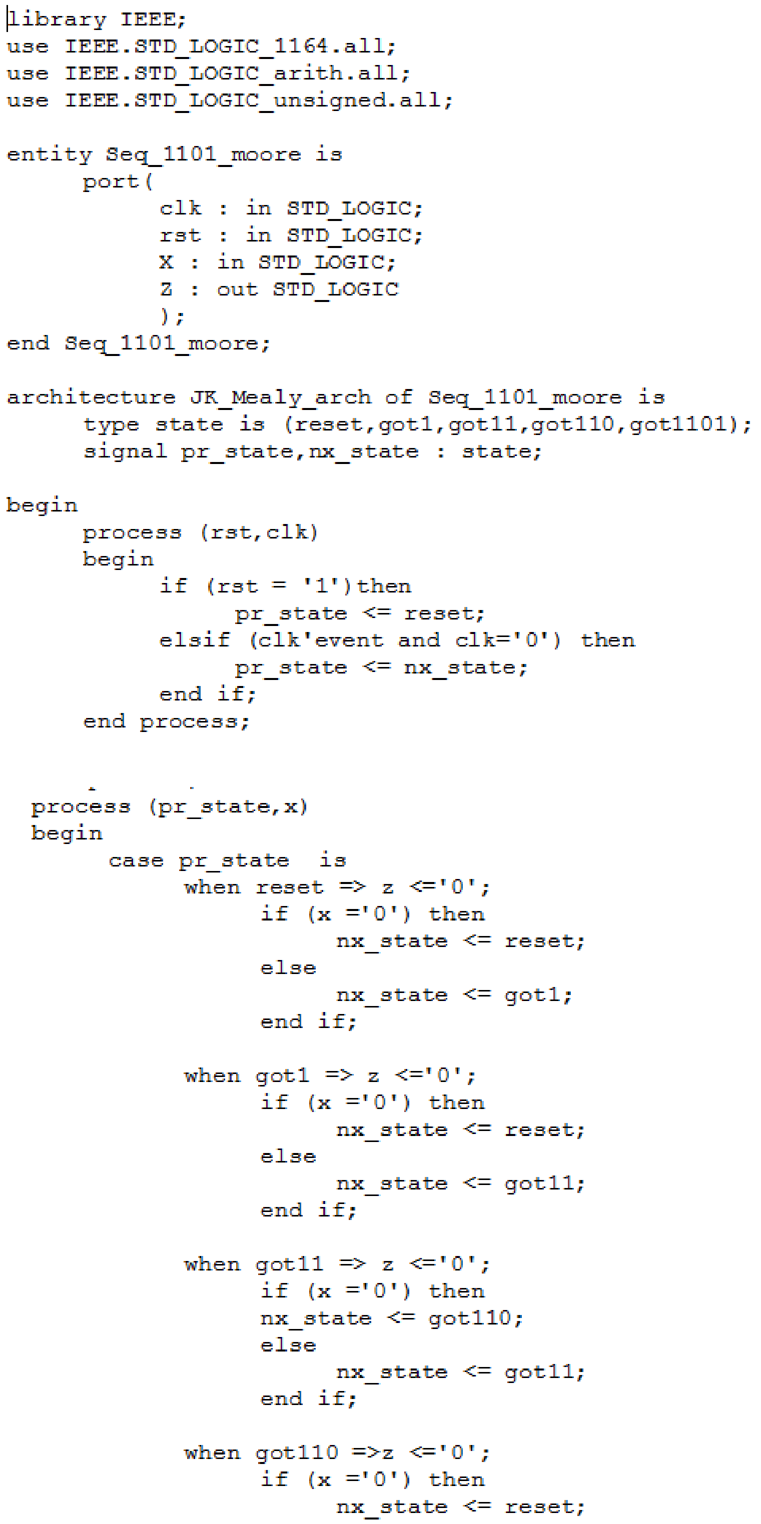
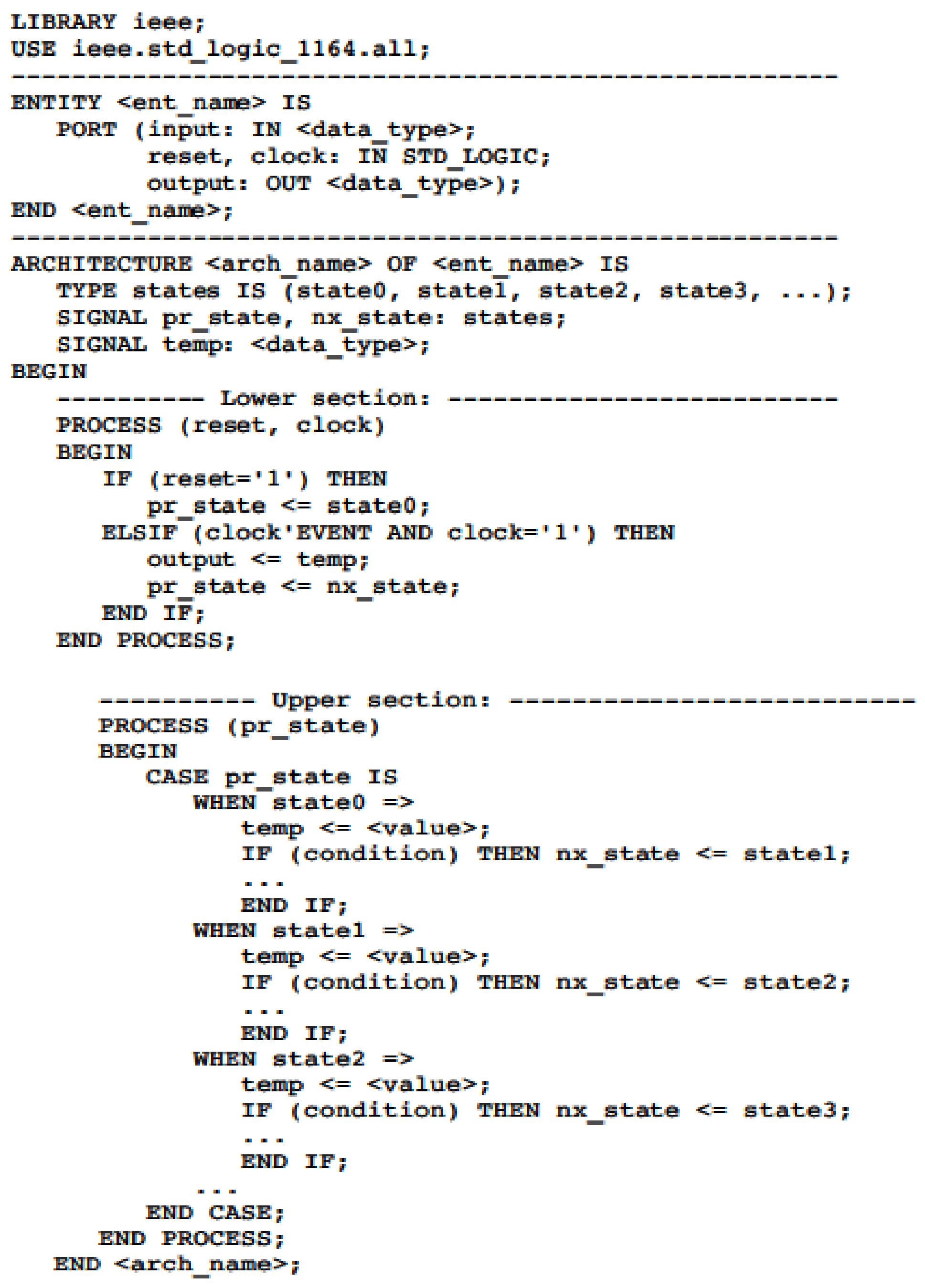
Overlapping Moore code

Tuesday, April 20, 2021 1:49 PM

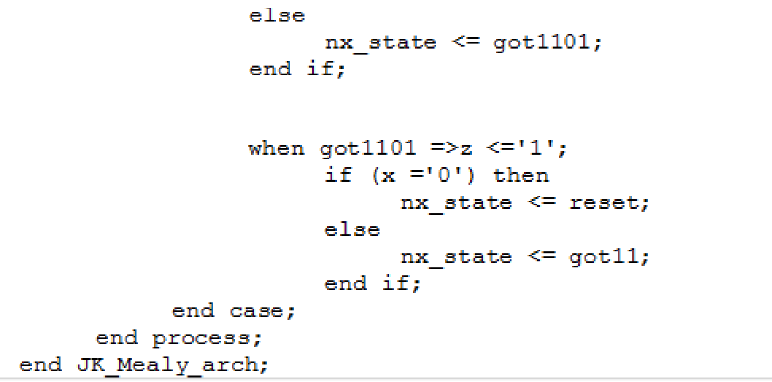




```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_arith.all;
use IEEE.STD_LOGIC_unsigned.all;

entity Seq_1101_moore is
      port(
            clk : in STD_LOGIC;
            rst : in STD_LOGIC;
            X : in STD_LOGIC;
            Z : out STD_LOGIC
            );
end Seq_1101_moore;

architecture JK_Mealy_arch of Seq_1101_moore is
      type state is (reset,got1,got11,got110,got1101);
      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 => z <='0';
                    if (x ='0') then
                          nx_state <= reset;
                    else
                          nx_state <= got1;
                    end if;

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

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

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



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

      ---------- Upper section: --------------------------
      PROCESS (pr_state)
      BEGIN
         CASE pr_state IS
            WHEN state0 =>
               temp <= <value>;
               IF (condition) THEN nx_state <= state1;
               ...
               END IF;
            WHEN state1 =>
               temp <= <value>;
               IF (condition) THEN nx_state <= state2;
               ...
               END IF;
            WHEN state2 =>
               temp <= <value>;
               IF (condition) THEN nx_state <= state3;
               ...
               END IF;
            ...
         END CASE;
      END PROCESS;
   END <arch_name>;
```



```
            else
                nx_state <= got1101;
            end if;

            when got1101 =>z <='1';
                if (x ='0') then
                    nx_state <= reset;
                else
                    nx_state <= got11;
                end if;
        end case;
    end process;
end JK_Mealy_arch;
```