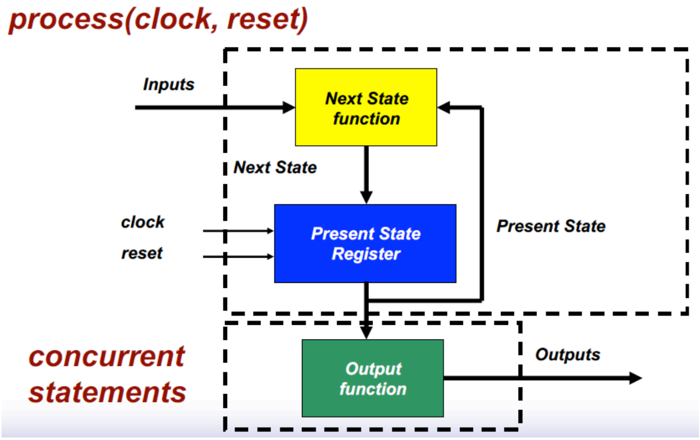
EE4620/6620 FSM (Exercise 2) *Prof. Henry Chen*

1. Let's construct the sequence detector for the sequence ***3 or more consecutive 1's*** using **Moore** state machine. The Output of the State machine depends only on present state. The output of state machine are only updated at the clock edge. ***The FSM uses an " asynchronous" positive reset which has a higher priority than the clock.***



The Moore state machine require **four states *a, b, c,* and *d*** to detect the ***3 or more consecutive 1's*** sequence.

a) Draw the state transition graph (STG).

b) Complete the entity and architecture pair of VHDL description for the Moore machine.

```
library ieee;
use ieee.std_logic_1164.all;
entity seq_detect is
        port (clk, reset, x : in std_logic;
                y : out std_logic);
end;

architecture enum of seq_detect is
type state is (state_a, state_b, state_c, state_d);

signal present_state, next_state : state;

begin
        nx_state: process (present_state, x) -- next state process
        begin
        -- use CASE and then IF statement
```

```
    state_reg: process (clk, reset_bar)     -- state register process
    begin



    end process state_reg;

    -- Concurrent VHDL for output assignment


end enum;
```

c) Construct **state table** for the Moore machine. There are 4 states, a, b, c, and d. You need 2 flip-flops, A and B for implementation. Use state assignment (A B) = (0 0) for state **a**, (0 1) for state **b**, (1 0) for state **c**, and (1 1) for state **d**. Note: In your state table, use A_P for the present state A, B_P for the present state B, A_N for the next state A, B_N for the next state B, x for the input, and y for the output. **Derive Boolean equation for the next state A_N, B_N, and the output y.**