```
module fsm (
clock      , // clock
reset      , // Active high, syn reset
req_0      , // Request 0
req_1      , // Request 1
gnt_0      , // Grant 0
gnt_1
);
//inputs
input   clock,reset,req_0,req_1;
//outputs
output  gnt_0,gnt_1;
reg     gnt_0,gnt_1;
 //FSM variables
parameter SIZE = 3          ;
parameter IDLE  = 3'b001,GNT0 = 3'b010,GNT1 = 3'b100 ;
//-------------Internal Variables----------------------------
reg   [SIZE-1:0]          state       ;
reg   [SIZE-1:0]          next_state   ;
//----------Code startes Here------------------------
always @ (posedge clock)
begin : fsm_proc
if (reset == 1'b1) begin
  state <= #1 IDLE;
end else
 case(state)
   IDLE : if (req_0 == 1'b1) begin
              ….
   GNT0 : if (req_0 == 1'b1) begin
              ……
   GNT1 : if (req_1 == 1'b1) begin
              …
   default : state <= #1 IDLE;
endcase
end

endmodule
```

Use ANSI-style port declarations

Use new **enum** type for FSM state variables

Qualify always process as '**always_ff**'

Case statement can be qualified with '**unique**' keyword