```
      else begin
      /*miss with dirty line*/
         /*write back address*/
         v_mem_req.addr = {tag_read.tag, cpu_req.addr[TAGLSB-1:0]};
         v_mem_req.rw = '1;
         /*wait till write is completed*/
         vstate = write_back;
      end
     end
end
/*wait for allocating a new cache line*/
allocate: begin
     /*memory controller has responded*/
     if (mem_data.ready) begin
        /*re-compare tag for write miss (need modify correct word)*/
        vstate = compare_tag;
        data_write = mem_data.data;
        /*update cache line data*/
        data_req.we = '1;
     end
      end
 /*wait for writing back dirty cache line*/
 write_back : begin
     /*write back is completed*/
     if (mem_data.ready) begin
        /*issue new memory request (allocating a new line)*/
        v_mem_req.valid = '1;
        v_mem_req.rw = '0;

        vstate = allocate;
     end
   end
 endcase
end

always_ff @(posedge(clk)) begin
 if (rst)
   rstate <= idle;        //reset to idle state
 else
   rstate <= vstate;
end
/*connect cache tag/data memory*/
dm_cache_tag  ctag(.*);
dm_cache_data cdata(.*);
endmodule
```

FIGURE e5.12.8 FSM in SystemVerilog, part IV. Actual FSM states via the case statement in the prior figure and this one. This figure has the last part of the Compare Tag state, plus Allocate and Write-Back states.