```
//-------------------------------------Cache FSM-------------------------
case(rstate)
/*idle state*/
idle : begin
    /*If there is a CPU request, then compare cache tag*/
    if (cpu_req.valid)
       vstate = compare_tag;
       end
/*compare_tag state*/
compare_tag : begin
          /*cache hit (tag match and cache entry is valid)*/
    if (cpu_req.addr[TAGMSB:TAGLSB] == tag_read.tag && tag_read.valid) begin
          v_cpu_res.ready = '1;
    /*write hit*/
    if (cpu_req.rw) begin
       /*read/modify cache line*/
       tag_req.we = '1; data_req.we = '1;

       /*no change in tag*/
       tag_write.tag = tag_read.tag;
       tag_write.valid = '1;
       /*cache line is dirty*/
       tag_write.dirty = '1;
    end

       /*xaction is finished*/
       vstate = idle;
end
/*cache miss*/
else begin
  /*generate new tag*/
  tag_req.we = '1;
  tag_write.valid = '1;
  /*new tag*/
  tag_write.tag = cpu_req.addr[TAGMSB:TAGLSB];
/*cache line is dirty if write*/
tag_write.dirty = cpu_req.rw;

/*generate memory request on miss*/
v_mem_req.valid = '1;
/*compulsory miss or miss with clean block*/
if (tag_read.valid == 1'b0 || tag_read.dirty == 1'b0)
   /*wait till a new block is allocated*/
   vstate = allocate;
```

FIGURE e5.12.7 FSM in SystemVerilog, part III. Actual FSM states via case statement in this figure and the next. This figure has the Idle state and most of the Compare Tag state.