```
/*cache finite state machine*/

module dm_cache_fsm(input  bit clk, input  bit rst,
        input  cpu_req_type   cpu_req,     //CPU request input (CPU->cache)
        input  mem_data_type  mem_data,    //memory response (memory->cache)
        output mem_req_type   mem_req,     //memory request (cache->memory)
        output cpu_result_type cpu_res     //cache result (cache->CPU)
     );

  timeunit 1ns;
  timeprecision 1ps;

  /*write clock*/
  typedef enum {idle, compare_tag, allocate, write_back} cache_state_type;

  /*FSM state register*/
  cache_state_typevstate, rstate;

  /*interface signals to tag memory*/
  cache_tag_typetag_read;                  //tag read result
  cache_tag_typetag_write;                 //tag write data
  cache_req_typetag_req;                   //tag request

  /*interface signals to cache data memory*/
  cache_data_typedata_read;                //cache line read data
  cache_data_typedata_write;               //cache line write data
  cache_req_typedata_req;                  //data req


  /*temporary variable for cache controller result*/
  cpu_result_typev_cpu_res;

  /*temporary variable for memory controller request*/
  mem_req_typev_mem_req;

  assign mem_req = v_mem_req;              //connect to output ports
  assign cpu_res = v_cpu_res;
```

FIGURE e5.12.5 FSM in SystemVerilog, part I. These modules instantiate the memories according to the type definitions in the previous figure.

Copyright © 2014 by Mosby, an imprint of Elsevier Inc.