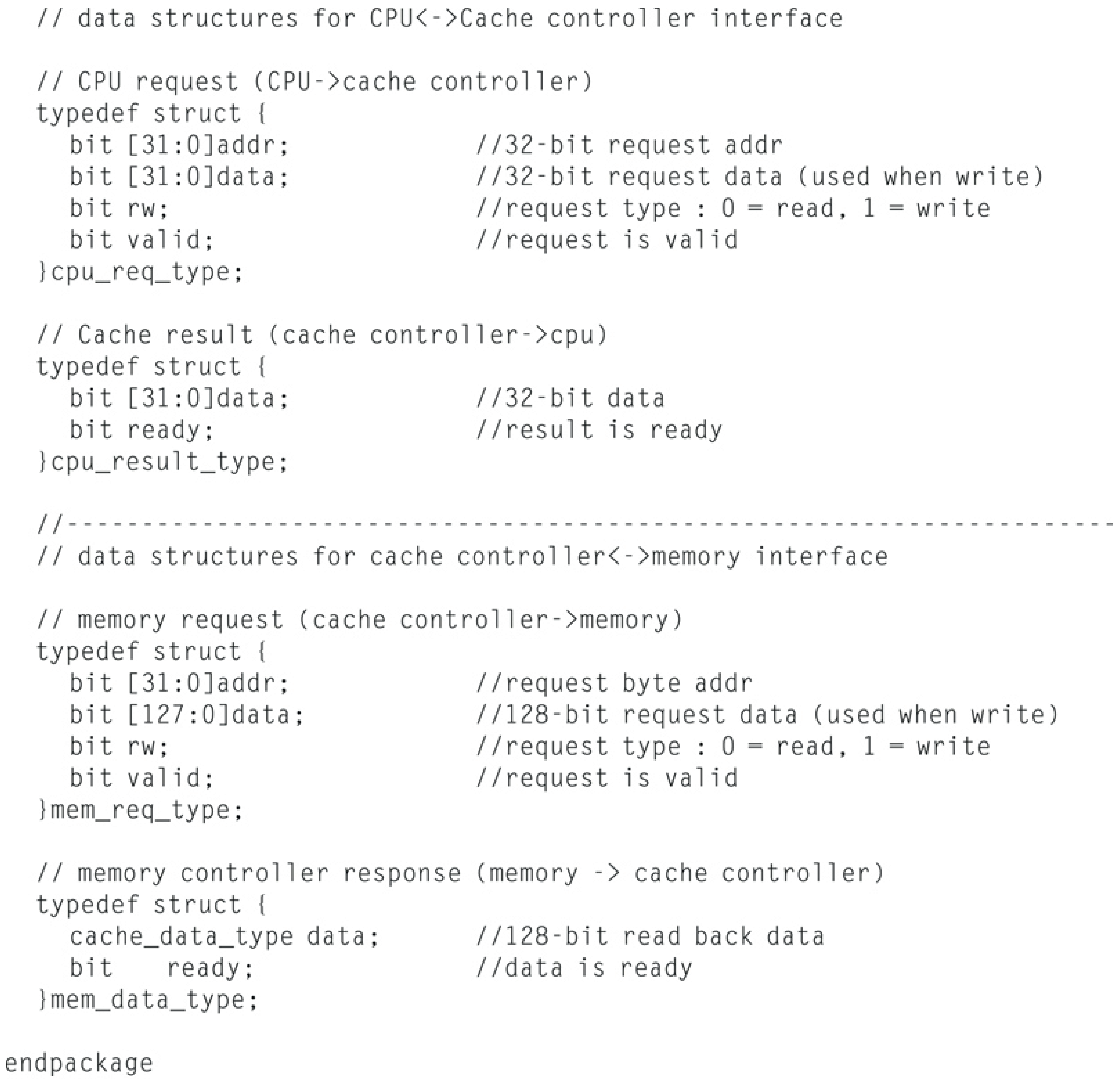

```
  // data structures for CPU<->Cache controller interface

  // CPU request (CPU->cache controller)
  typedef struct {
    bit [31:0]addr;                //32-bit request addr
    bit [31:0]data;                //32-bit request data (used when write)
    bit rw;                        //request type : 0 = read, 1 = write
    bit valid;                     //request is valid
  }cpu_req_type;

  // Cache result (cache controller->cpu)
  typedef struct {
    bit [31:0]data;                //32-bit data
    bit ready;                     //result is ready
  }cpu_result_type;

  //-------------------------------------------------------------------------
  // data structures for cache controller<->memory interface

  // memory request (cache controller->memory)
  typedef struct {
    bit [31:0]addr;                //request byte addr
    bit [127:0]data;               //128-bit request data (used when write)
    bit rw;                        //request type : 0 = read, 1 = write
    bit valid;                     //request is valid
  }mem_req_type;

  // memory controller response (memory -> cache controller)
  typedef struct {
    cache_data_type data;          //128-bit read back data
    bit    ready;                  //data is ready
  }mem_data_type;

endpackage
```

FIGURE e5.12.2 Type declarations in SystemVerilog for the CPU-cache and cache-memory interfaces. These are nearly identical except that the data is 32 bits wide between the CPU and cache and is 128 bits wide between the cache and memory.