CornellX: ENGRI1210x The Computing Technology Inside Your Smartphone

KarenWest ▼

Courseware Course Info Discussion Wiki Progress Discussion Guidelines Resources Exploring Engineering Syllabus How to Use Jade

Help

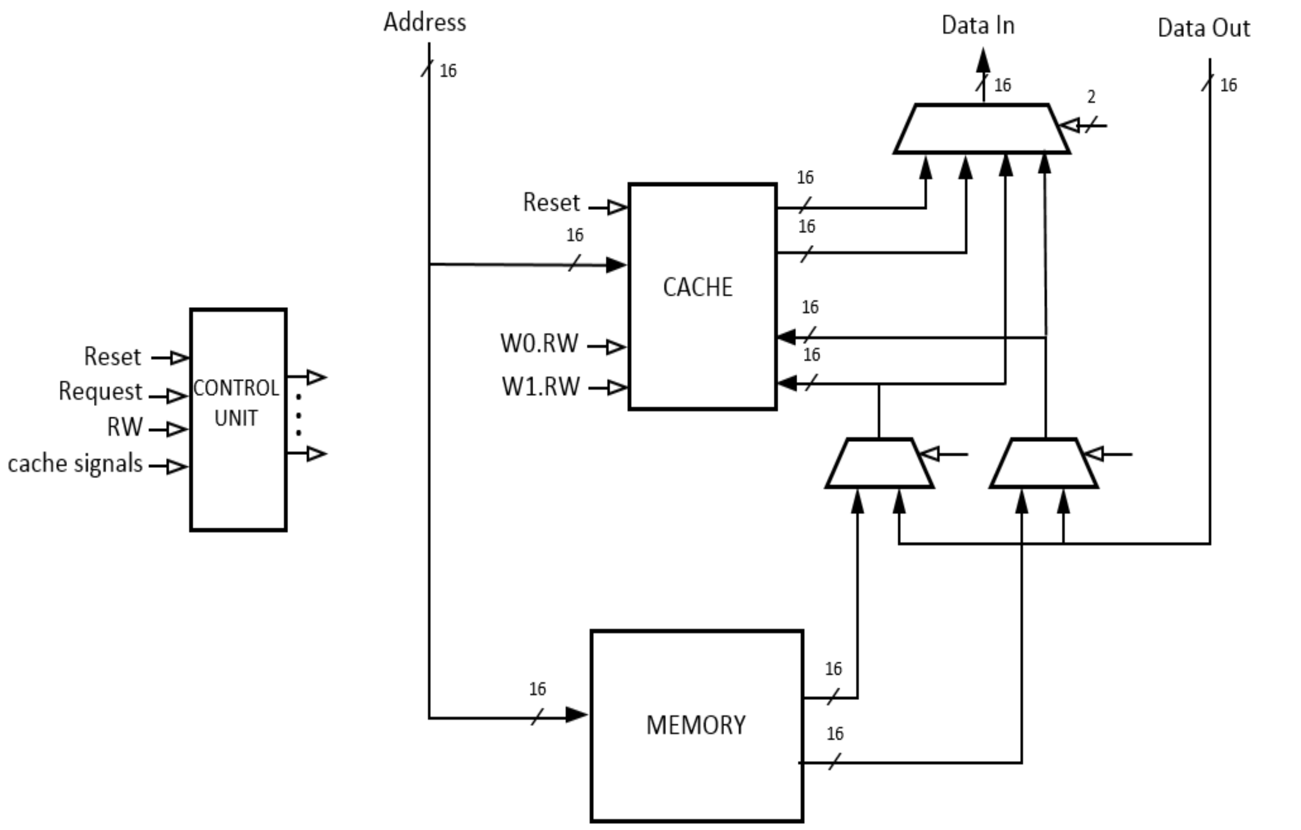
This **LC-3 cache hackathon** involves the design of a data cache. Collectively, you'll create working hardware and write Jade test programs to verify that your design works. You'll receive a badge in recognition of your participation in the hackathon. (See the last page for directions.)

Help

The process is as follows. We'll provide a little technical information (below), work windows, and discussion forums. The rest is up to you in terms of how you work together to figure out the design subtleties, to develop working designs, and to create Jade tests to verify design correctness. Use what we provide, or group collaborative tools that you find online. Hold face-to-face meetings with students in your geographical area, or use the discussion forum and collaborative tools to work together online. Unlike graded work, you are encouraged to share!

LC-3 Data Cache Diagram

Below is a diagram of an LC-3 data cache and memory:



Processor/Cache Interface

The processor/cache interface consists of the following signals:

- Request: When 1, the processor has made a data cache request.

- RW: When 0, the request is a read, and a write when a 1.
- Address: Memory address.
- Data In: Data into the processor (for Load instructions).
- Data Out: Data out of the processor (for Store Instructions).

Help

You may assume that the processor maintains the values of Request, RW, Address, and Data Out until the operation is completed.

Cache Design

The cache is direct mapped and holds 32 blocks, requiring 5 index bits. The block size is two 16-bit words (32 bits), which means that we have 1 word offset bit. The remaining 10 bits are the tag.

The cache should be constructed from four 32-entry memories: Valid bit memory (32 x 1), Tag memory (32 x 10), Data Word 0 memory (32 x 16), and Data Word 1 memory (32 x 16).

Memory Design

The Memory holds 32,768 32-bit blocks (32,768 x 32). For simplicity, you might assume that both the cache and Memory require one cycle to access, even though in a real design the Memory would be much slower than the cache.

Operation

When Request = 1, the cache is first checked for the block using the Address. The Valid and Tag memories are first read. On a hit for a Load instruction (RW = 0), both Data Word memories are then read, and the desired word is passed to the processor on Data In. On a hit for a Store instruction (RW = 1), the word on Data Out is then written to the desired word location in the cache.

On a miss, the block is retrieved from Memory and written into the cache. For a Load, the desired word is returned on Data In. For a Store, the word on Data Out is written to the desired word location in the cache.

The above design is one suggestion. You may choose different sizes for your cache and Memory. Moreover, the design assumes for simplicity that the Memory is only read and never written. You may choose to create a more complex design where the Memory is written along with the cache on a Store.

Use the following discussion forum to discuss how the design works, and whether you want to deviate from the above design parameters:

Show Discussion New Post

Control Unit

Help

The Control Unit orchestrates the flow of data in and out of the cache and, on a miss, from Memory to the cache. It receives the Request and RW signals from the processor, and signals from the cache, and generates all of the cache and Memory control signals to handle Loads and Stores, and cache hits and misses.

You may decide to implement the Control Unit as an FSM, or as an FSM and microinstruction memory as was done with the LC-3 processor control.

Use the following discussion forum to discuss the Control Unit design:

Show Discussion New Post

Tasks

The design will require three major tasks:

- Datapath design, including memories for the data cache and Memory, cache hit logic, and muxes.
- Control Unit design.
- Jade tests that verify that the hardware operates properly.

You will need to coordinate among these three tasks. For instance, those writing the Jade tests will need to know the module and signal names used by the hardware designers. This may require creating a design specification with agreed-upon signal names.

Stay patient! It will take time to get organized and figure everything out!

Use the following discussion forum to discuss how to organize yourselves, including face-to-face meetings and online collaboration tools:

Show Discussion New Post

About edX Follow Us