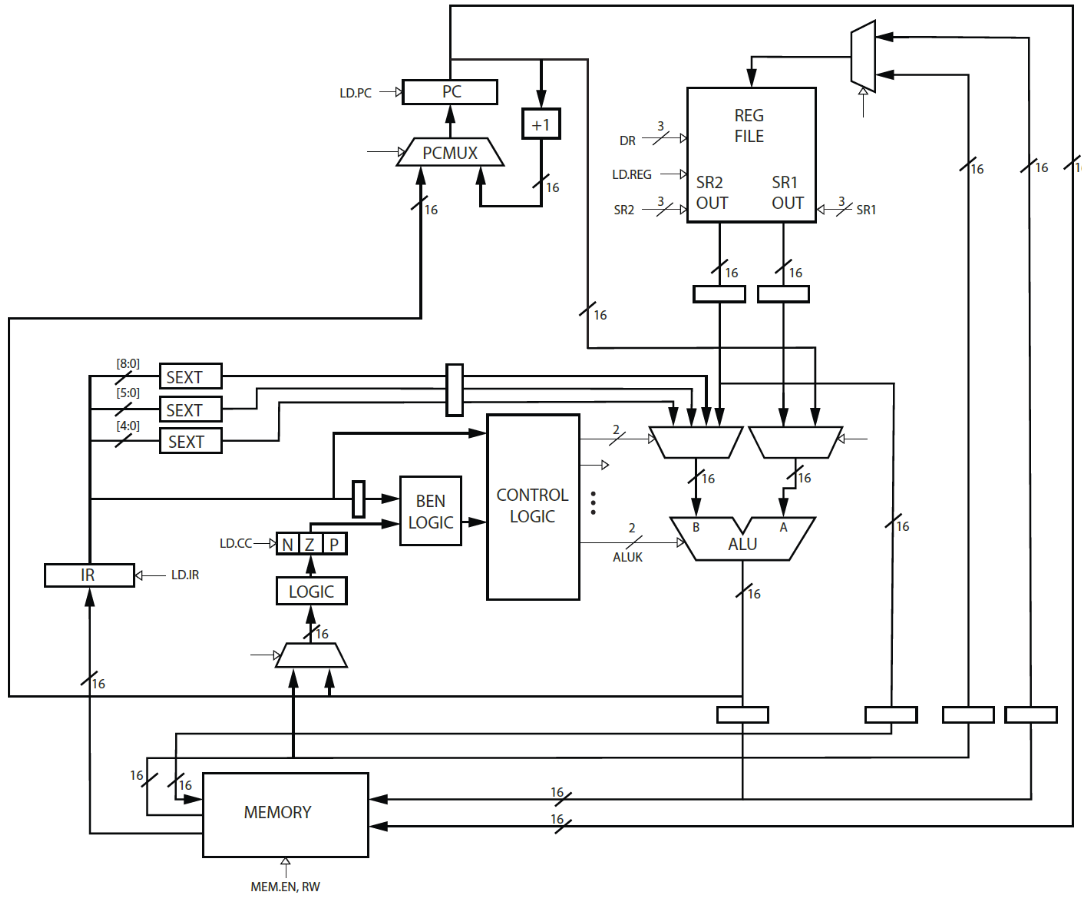
INTRODUCTION

Are you up to the challenge of creating a pipelined LC-3 Lite with your fellow students? Then participate in this **LC-3 Lite Pipelining Hackathon**! Collectively, you'll not only create working hardware, but also write Jade test programs and assembly language code to verify that your design works, and compare its performance against the non-pipelined LC-3 Lite. You'll also receive a badge from Cornell in recognition of your participation in the hackathon. (See the last page for directions.)

Here's how this works. 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, to create Jade tests to verify design correctness, and to design assembly language programs to measure performance. 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!

PIPELINED LC-3 LITE BLOCK DIAGRAM

Below is the LC-3 Lite pipeline diagram without forwarding:



Here's a bit of information about the design:

- The unnamed rectangles represent portions of the pipeline registers.
- The memory includes two ports, one for IF and a second for MEM. IF consists of an address input and data output (for instructions). MEM comprises an address input, a data output (for LDR), and a data input (for STR).
- In ID for STR, the data to be stored is read from SR2 OUT, since SR1 OUT is used to read the base register. This differs from the non-pipelined LC-3 where the data to be stored was read from SR1 OUT.
- The TRAP-Halt instruction must allow all prior instructions in the pipelined to complete before stopping the machine. You may choose to not implement TRAP-Halt, but rather to use it as a marker of the completion of the program in the Jade timing window.

Use the following discussion forum to discuss how the design works for each of the seven LC-3 Lite instructions:

Show Discussion

New Post

Help

CONTROL LOGIC

Because each stage holds a different instruction, the Control Logic is not a centralized state machine as with the non-pipelined LC-3 Lite. Rather, each stage generates its own control based on the instruction. (Thus, the single Control Logic block in the diagram should be visualized as five blocks, one for each stage.) In order to accomplish this stage-by-stage control, the contents of IR, which are loaded during IF, are passed along with the instruction through every successive stage. You will design the hardware that uses this instruction information to generate the control.

Use the following discussion forum to discuss the control logic design for each stage:

Show Discussion

New Post

PIPELINED LC-3 LITE DESIGNS

You will create two designs. The first will not support forwarding or discarding instructions following a taken BR. The code written for this version will require NOP instructions in the assembly code (using the .NOP assembly directive) in order to provide enough spacing between dependent instructions, and after a BR instruction. The second, high performance version will support both of these features.

TASKS

Both versions will require three major tasks:

- Hardware design for every stage, and for the complete design.
- Jade tests for every stage, and for the complete design, which verify that the hardware operates properly.
- Assembly language programs to compare the performance of the different designs.

You will need to coordinate among these three tasks. For instance, those writing the Jade tests will need to know the names used by the hardware designers. The assembly language programmers will need to understand the limitations of the designs in terms of required NOPs.

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

‹ ›

Help

EdX offers interactive online classes and MOOCs from the world's best universities. Online courses from MITx, HarvardX, BerkeleyX, UTx and many other universities. Topics include biology, business, chemistry, computer science, economics, finance, electronics, engineering, food and nutrition, history, humanities, law, literature, math, medicine, music, philosophy, physics, science, statistics and more. EdX is a non-profit online initiative created by founding partners Harvard and MIT.

Terms of Service and Honor Code

Privacy Policy (Revised 10/22/2014)

POWERED BY OPENedX

About edX

- About
- News
- Contact
- FAQ
- edX Blog
- Donate to edX
- Jobs at edX

Follow Us

- Facebook
- Twitter
- LinkedIn
- Google+
- Tumblr
- Meetup
- Reddit
- Youtube