

Lab 4: Hardware and Software for Interrupt-driven I/O

In this lab, you will upgrade the low-level hardware and software support for input and output for the HERA processor. Continue to work in the same project, but make a copy of the CPU and start new files for the new software.

Divide into groups of 2-3 people each, and work on *one* of the following projects. Coordinate with a group of people doing the other project, and produce a working hardware/software combination:

HARDWARE. Modify your HERA processor so that it can handle interrupt signals, and change the connections to the terminal you added in Lab 2 so that it generates interrupt signals when it is ready to receive or send the next character. Note that, for the purposes of I/O, having interrupts happen only at the *start* of a clock cycle is fine, so you can use a flip-flop or latch to make sure you don't get interrupted right in the middle of an operation (this is not o.k. for interrupts for virtual memory, but we will **not** be doing this in the lab). Coordinate with the software group to set standards as necessary.

SOFTWARE. Write interrupt routines to handle interrupts for input and output. When the terminal is ready to send a character to the processor, the interrupt routine should load that character and save it in a buffer; when the terminal is ready to receive a character, the processor should send it one if one is ready in the output buffer, or simply record the fact that the terminal is ready. Update your functions to print and read tiger strings so that they simply access the buffers; `print` should now just add characters to the buffer and start transferring if the terminal is waiting; `getchar` and `getline` should transfer one character (or line) from the buffer if one is there, or (for now) busy-wait until enough data arrives. Coordinate with the hardware group to set standards as necessary.

Remember to account for the possibility that an interrupt could occur while the interrupt processing code is running, or during a print or read function.

Continue to keep backup copies and exchange work via CVS.