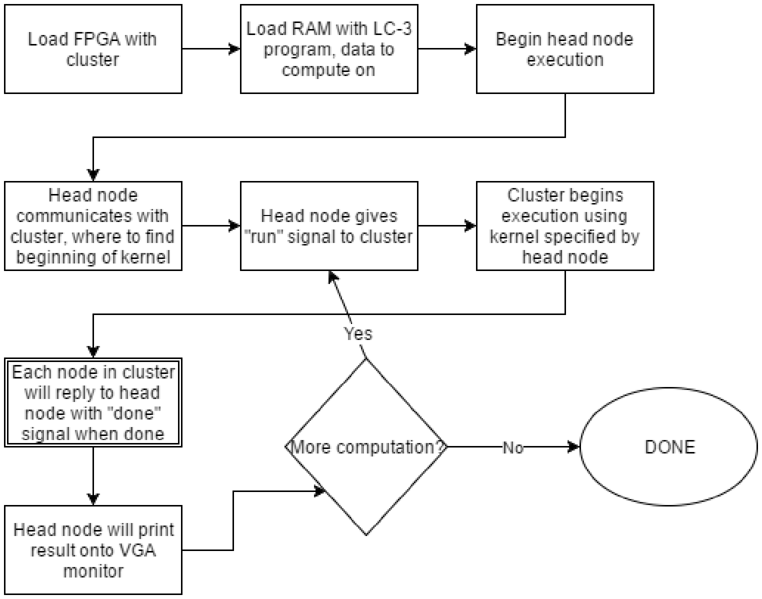
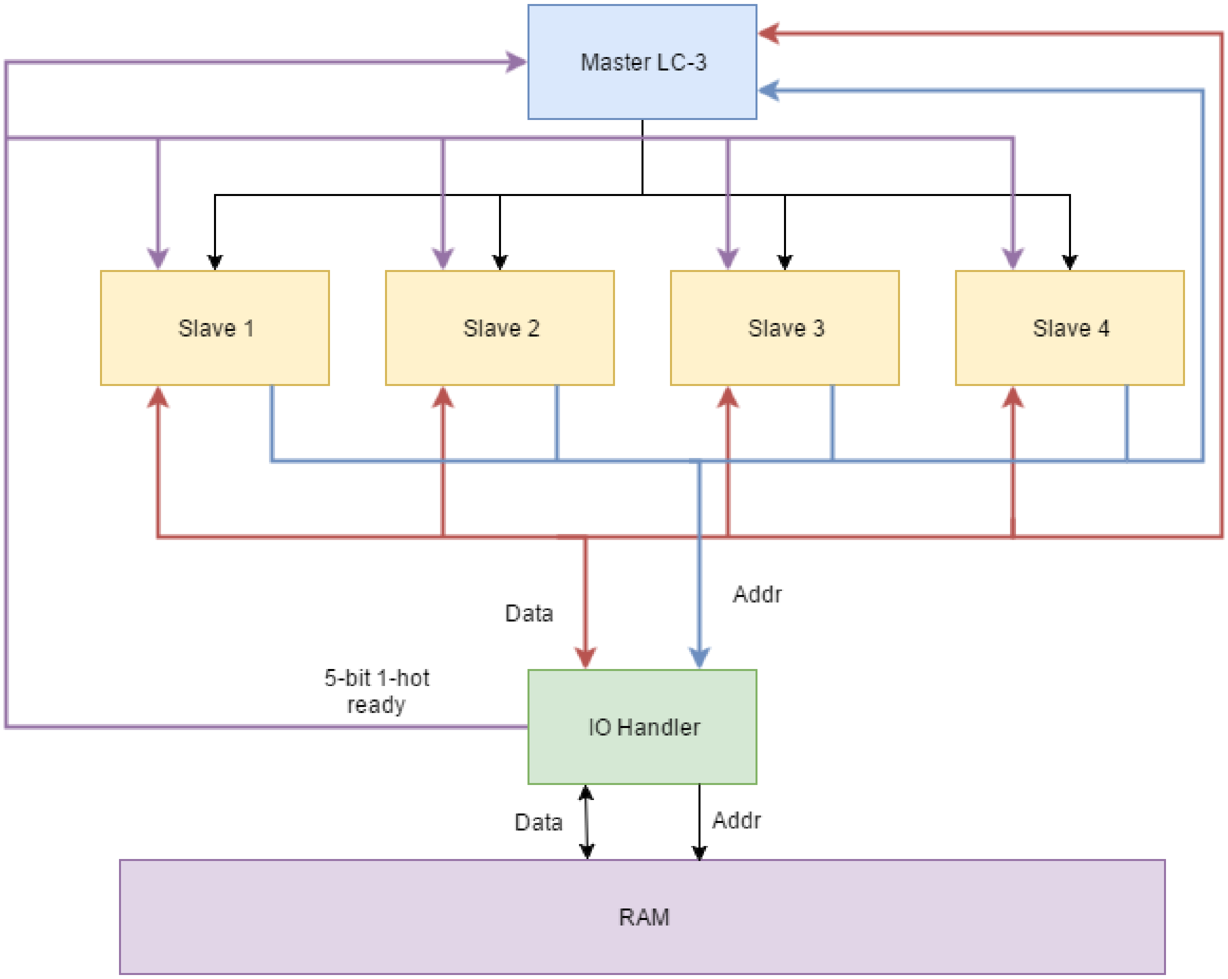
Landon Clipp -- clipp2
Jack Hu -- yhu49

ECE 385 Final Project Proposal

For our final project, we propose to create a parallel LC-3 compute cluster. The parallel compute cluster will act similar to how modern GPUs work with the exception of some distinct differences. In modern GPUs, each core is a specific-purpose processor designed and optimized for video processing. These cores are generally much simpler and less versatile than the CPU. In our project, we will use 4 LC-3 cores for our parallel compute unit. These 4 cores, which will be henceforth called the compute cluster, will be controlled by a “head node” LC-3 CPU that will initialize the RAM with the necessary **compute kernel** (program that the compute cluster cores will execute) and the data that the cluster will perform operations on much in the same way that modern CPU-GPU combinations interact. The main program executed by the head node, as well as the compute kernel, will both need to be written in LC-3 assembly. In order to prevent memory access collisions between all of the cores, an IO queue module will need to be written that handles memory access. An SD card will need to be used to initialize all RAM on the FPGA development board.

The goal of this project is to serve as a proof-of-concept for parallel computation. We intend on benchmarking our compute cluster on a computationally intensive task on some large array of data and compare the benchmark against the same computation performed by a single LC-3 CPU (perhaps by simply tasking the head node to perform the computation). The nature of the computation will be arbitrary, perhaps simply incrementing each cell of the array some number of times. This is what is promised by our project proposal. If time permits, we intend on expanding our project by implementing Conway’s Game of Life, a perfect candidate for parallel computation. Conway’s Game of Life is a famous deterministic non-interactive cellular automata simulation that demonstrates one of the key points of chaos theory, the butterfly effect. This game will be displayed on the VGA monitor. For both assembly programs, the arbitrary computation promised by this report and Conway’s Game of Life, the LC-3 ISA will need to be expanded to allow for some common parallel programming concepts like variable locking, core synchronization, and other various constructs to help prevent race conditions. We will use the SLC-3.2 ISA as a base, or perhaps use a previously-completed expansion of the LC-3 ISA done by another final project if it can be found (due citations will be given if we use another final project as a base). The execution workflow for our system is described in the block diagram below.


Load FPGA with cluster
Load RAM with LC-3 program, data to compute on
Begin head node execution
Head node communicates with cluster, where to find beginning of kernel
Head node gives "run" signal to cluster
Cluster begins execution using kernel specified by head node
Each node in cluster will reply to head node with "done" signal when done
Head node will print result onto VGA monitor
Yes
More computation?
No
DONE



List of Features:

1. Base feature: Implement a parallel computation cluster capable of performing a simple, 1-to-1 task on a set of data. Turn on an on-board LED when computation is finished (for purposes of benchmarking). Modified RAM data should be viewed using the DE2 control panel. Benchmarked time will be compared against single LC-3 core running same computation on all data.
2. Extra feature: Implement the LC-3 TRAP function for printing to VGA screen and print the results of the computation onto the screen when computation is finished.
3. Extra feature: Implement Conway's Game of Life using the VGA screen.

Expected Difficulty

Significant changes will be made to the SLC-3 ISA due to the need to account for memory ready signals, run signals, done signals, TRAP subroutines, and various other instructions that will need to be made. A memory IO handler will need to be implemented in hardware to prevent memory access collisions between all of the cores. For the base feature, we expect a difficulty of **5** to be appropriate.

If extra feature 2 is implemented, a difficulty of **8** would be expected due to the large amount of work getting text sprites to be implemented on the VGA screen.

If extra feature 3 is implemented, a difficult of **10** is expected. 10 is appropriate because significant work will need to be done to implement variable locking, a common parallel programming construct. Also, other considerations will need to be taken into account in the assembly program to prevent race conditions. These considerations will most likely involve an even further expansion to the SLC-3 ISA.

Proposed Timeline

Week of 4/4: Design phase, come up with the details of new instructions, memory interaction, and datapath.

Week of 4/11: Work on / finish the base features. Base features will be used for mid-checkpoint demo.

Week of 4/18 (mid demo): Work on additional features as time permits.

Week of 4/24 (last week to work on final project): Debug, wrap-up.