# Final Project Proposal
## Multiple CPU Communication

Sparsh Bansal | Thomas Jagielski | Dhara Patel

Olin College of Engineering
[sbansal, tjagielski, dpatel]@olin.edu

## 1 Project Overview

We want to try implementing a device consisting 2 CPUs that can perform operations in parallel for efficient computation and reduced runtime.

According to an article posted to EDN, "The future of computers – Part 1: Multi-core and the Memory Wall," "MapReduce is one of the most commercially popular massively parallel applications. It divides huge databases across many microprocessors in order to perform rapid searches in parallel."

Some of the challenges we will potentially tackle include:

- Shared/Standalone hardware
- Programming the hardware using single or multiple assembly programs
- Compiling a program
- Interrupt handling for communication between the 2 CPU's

## 2 Deliverables

### 2.1 Minimum Viable Product

A multi-core CPU (2 cores) that uses a shared data memory between the cores send and receive data. Written in Verilog, each core will have its own assembly code and instruction memory. We cannot use multiprocessor framework, so we must use co-processor structure.

### 2.2 Planned Goal

We will iterate on our MVP to implement a system in which we have a single assembly program and single instruction memory. And, the instructions are divided up among the two cores somehow. We are looking into how we can implement this using MIPS in the following ways:

1. This can be implemented in hardware possibly using a MUX, which we will explore more.
2. We could also assign "IDs" to each processor so that the instructions can be assigned to the correct processor by the assembly code itself.

### 2.3 Stretch Goal

We will iterate on our planned goal, but create a heterogeneous CPU architecture. In addition, we can add more cores to allow for more complicated operations. We can consider cores which are optimized for specific things such as multiplication, power usage, or physical size.

## 3 Milestones

Week 1: Block Diagram of System for each of our goals (shared memory vs dedicated instruction memory for each core)
Week 2: Complete the MVP
Week 3: Complete Planned Goal
Week 4: Planned goal buffer and write-up rough draft
Week 4.5: Complete Write-Up for submission

## 4 References


1. The future of computers Part 1: Multicore and the Memory Wall - EDN
2. Hardware Accelerated Mappers for Hadoop MapReduce Streaming - Katayoun Neshatpour, Maria Malik, Avesta Sasan, Setareh Rafatirad, Houman Homayoun
3. What Is a GPU? - Modern GPU technology powers traditional graphics applications, Intel
4. https://www.mentor.com/embedded-software/events/cpu-to-cpu-com-mulitcore-webinar
5. https://www.jontse.com/courses/cs5220.html
6. http://15418.courses.cs.cmu.edu/fall2016content/lectures/