

UC Berkeley Computer Science

Subject: Spring Semester Objective
Date: January 17, 2006
From: Ivan Sutherland
UCIES #2006-is15

References:

UCIES# 2005-is14: FLEET – A One-Instruction Computer, Ivan Sutherland, 5 December 2005
Reference to Igor's memos

OBJECTIVE

During the Spring semester I'd like us to simulate a FLEET system. For this purpose we have two tools available: Igor's simulator and the RAMP system. We will use them to build FLEET systems of increasing complexity. It is my hope that by the end of the semester we will be able to run code in a FLEET system.

There are two parts to running code in a FLEET system: the "hardware" and the "software." Our "hardware" will either be simulated by Igor's simulator, or be assembled in the RAMP system. It would be nice if a common description of the "hardware" could serve as input to both these tools. The "software" is, of course, the content of the code bags used to drive the hardware. It will be important that the same source software be able to run on both systems.

Thus we have two tasks before us: a "tool" part and a "substance" part. The tool task seeks to merge the "hardware" descriptions required by two quite different tools. As part of this task I'd like us to understand the structure of both tools. We should strive to simplify them if possible, but especially strive to improve their understandability. I believe that comparison of the two tools may help us distinguish the essential from the irrelevant in describing the structures we wish to build.

The substance task seeks to design and program FLEET systems. In the Fall semester we focused on understanding what to include in FLEET. We wrote a variety of simple programs and critiqued them. As a result of this work, the design of FLEET evolved: we came to understand literals, we rejected indexing source and destination addresses, and we learned about the concurrency possible in a FLEET design. The memo, is14, cited above, attempts to summarize the results of our work. Our Spring semester task is to make programs real by running them. If there's anything we learned in the Fall, it's how hard it is to find bugs in the type of concurrent code used by FLEET.