

# UC Berkeley Computer Science

**Subject:** The Files Needed to Simulate FLEET  
**Date:** January 25, 2006  
**From:** Ivan Sutherland  
**UCIES** #2006-is18

## References:

UCIES# 2005-is05: Notation Questions, Ivan Sutherland, 12 September 2005  
UCIES# 2005-is06: Some Ideas About Notation, Ivan Sutherland, 13 September 2005  
UCIES# 2005-is14: FLEET – A One-Instruction Computer, Ivan Sutherland, 5 December 2005  
UCIES# 2005-is15: Spring Semester Objective, Ivan Sutherland, 17 January 2006  
UCIES# 2005-is17: Flow Through a FIFO, Ivan Sutherland, 18 January 2006

## INTRODUCTION

We had a very encouraging class on Wednesday, January 18. I came away feeling that there's enough skill and enthusiasm for us to reach the objective set out in the is15 reference above. I believe we will see running code by semester's end.

There were two presentations during class. Igor Benko presented his simulator as it now stands. Students will initially use Igor's simulator to find answers to the questions posed in is16. More complex uses will follow.

The second presentation was by Adam Megaz. He offered a definition of assembly code syntax that follows the outline of the two papers on notation cited above, is05 and is06. His notation is compact enough to get the entire definition on a single page.

## THE FILES

A main topic of discussion was what files do we need to define and run a FLEET. Because we intend to simulate both with Igor's simulator and with the RAMP equipment at BWRC, we need to spend some time resolving any format conflicts that may exist.

This memo offers two tables of the files I know about. Table 1 is for use with Igor's FLEET simulator. Table 2, less well defined, is for running on RAMP. I seek additions and corrections to the tables.

---

This document is a product of a collaboration between Sun Microsystems and the University of California at Berkeley.. The ideas contained herein are freely available for any academic purpose.

<b>Table 1: Files required for running code on a simulated FLEET</b>					
<b>#</b>	<b>Purpose of file</b>	<b>Name</b>	<b>Language</b>	<b>Used by</b>	<b>Written by</b>
1	Define the FLEET parts available to assemble a fleet	The Library or Catalog	XML (format defined in DTD file)	Igor's simulator	SHIP Designers. Defined as we build up the library. There may be several separate files.
2	Define format of Library file in DTD	LibFormat	XML	Igor's simulator	Defined by Igor
3	Semantics of the FLEET parts	LibContent	Java	Igor's simulator	SHIP designers. Simple parts now exist. More to come.
4	Define the topology of a FLEET	Topology	XML	Igor's simulator	FLEET designer.
5	Define Topology file format in DTD	TopFormat	XML	Igor's simulator	Defined by Igor
6	Give binary addresses of Sources and Destinations	Address File	It's a table that maps names to numbers	A compiler. Adam's?	Format to be defined. Derived from Topology file
7	Programming FLEET	Source Program	Assembly code	A compiler. Adam's?	Format defined in memos and AssySyntax
8	Define syntax of Assembly code	AssySyntax	Adam's special	A compiler. Adam's?	Written by Adam
9	Tell FLEET what to do	Binary Program	FLEET machine code	FLEET	Produced by compiler. Adam's?
10	Give names to binary sources and destinations	Name File	It's a table that maps numbers to names	Debugging	The inverse of the Address File. Derived from Topology file

<b>Table 2: Files required for running FLEET code on RAMP</b>					
<b>#</b>	<b>Purpose of file</b>	<b>Name</b>	<b>Language</b>	<b>Used by</b>	<b>Written by</b>
1	Define the FLEET parts available to assemble a fleet.	The Library or Catalog	Use RDL	RAMP	SHIP Designers. Defined as we build up the library. There may be several separate files.
2	Syntax of Library file	LibFormat	UNKNOWN	RAMP	May be hard coded.
3	Semantics of the FLEET parts in Verilog or Java	LibContent	UNKNOWN	RAMP	Ship designers.
4	Define the topology of a FLEET	Topology	Use RDL Can easily be converted from XML file of table 1.	RAMP	FLEET designer.
5	Syntax of Topology file	TopFormat	UNKNOWN	RAMP	May be hard coded.
6	Give binary addresses of Sources and Destinations	Address File	It's a table that maps names to numbers	A compiler. Adam's?	Format to be defined. Derived from Topology file
7	Programming FLEET	Source Program	Assembly code	A compiler. Adam's?	Format defined in memos and Assembly Syntax
8	Define syntax of Assembly code	Assembly Syntax	Adam's special	A compiler. Adam's?	Written by Adam
9	Tell FLEET what to do	Binary Program	FLEET machine code	FLEET	Produced by compiler. Adam's?
10	Give names to binary sources and destinations	Name File	It's a table that maps numbers to names	Debugging	The inverse of the Address File. Derived from Topology file

<b>Purpose of file</b>	<b>Name</b>	<b>Language</b>	<b>Used by</b>	<b>Written by</b>
Define the FLEET parts available to assemble a fleet	The Library or Catalog	Format To Be Decided. Greg Gibeling may know	RAMP	SHIP Designers. Defined as we build up the library. There may be several separate files.
Define format of Library file	LibFormat	UNKNOWN	RAMP	May be hard coded.
Semantics of the FLEET parts	LibContent	UNKNOWN	RAMP	May be hard coded.
Define the topology of a FLEET	Topology	Format To Be Decided. Greg Gebling may know	RAMP	FLEET designer.
Define Topology file format	TopFormat	XML	Igor's simulator	Defined by Igor
Give binary addresses of Sources and Destinations	Address File	It's a table that maps names to numbers	A compiler. Adam's?	Format to be defined. Derived from Topology file
Programming FLEET	Source Program	Assembly code	A compiler. Adam's?	Format defined in memos and AssySyntax
Define syntax of Assembly code	AssySyntax	Adam's special	A compiler. Adam's?	Written by Adam
Tell FLEET what to do	Binary Program	FLEET machine code	FLEET	Produced by compiler. Adam's?
Give names to binary sources and destinations	Name File	It's a table that maps numbers to names	Debugging	The inverse of the Address File. Derived from Topology file

