# System Design

The figure below depicted our system hierarchy.



Figure 1 System Hierarchy

With the hierarchy easily seen and understood, now we’d like to address a few design ideas of the LEAF elements (Rom Loader, Memory, ISA, CU, Registers, and ALU) of the computer. The Rom Loader is supposed to actually load the boot program from files, which is used as ROM, to the memory. However, it works now as loading some hard coded instructions from within the Java program temporarily. We will change it soon.

The Memory is now a whole memory system. It has a main memory implemented as an array of 2048 integers and a simple cache. Its size of is designed to be expandable.

Our system is based on certain Instruction Set Architecture (ISA). It is not explicitly “owned” by CPU or part of it, though. A bunch of decoding schemes and instruction definition conventions would be included in this element.

The Control Unit would be in charge of lots of tasks as it is in practice. It needs to be able to direct Data Handling Operations such as load and store, and it is also the one that executes the Instruction Cycle, which is of great significance in our system.

What Processor Registers include INITIALLY has already been put on the figure. Their bit length is not necessarily the same as required in the project description. We packaged them together into the CPU Java class.

ALU would implement some arithmetic and logic operations.

The object design derived from the above figure. In addition, we made use of a few software engineering techniques for the implementation. Examples are Java interface, MVC pattern, etc. Lots of classes are in the source code and the code is arranged as well as possible in order for further development.

# UI Design

The outmost would be a JFrame of BorderLayout, which is described below:



Figure 2 Frame

For the moment we only made use of the center and south part to settle the Register Panel and Control Panel (to form the Operator Console). But in future we want to make some changes so that there would be Operator Console and Field Engineer Console in the Frame (hopefully one in the West and the other in the East).

Now we would mainly focus on the Register Panel since Control Panel would just consist of a few buttons.



Figure 3 Register Panel & SubPanels

There should be some things to be clarified for the components of Left Panel or Right Panel, Register GUI. First, the points of black and white, implemented with radio buttons, represent the value of that register. Second, there would always be a Set button to enable making changes to the value of certain registers at any time. Third, the Name and Index are optional because:

1. For GPR and IX, all would have indices but only the first has name.

2. For others, all would have name but none has index.