

## **ArbSimulator: The Arbadell Simulator**

This is a document describing how to use the Arbadell simulator. The source code is free and open to anyone who wishes to use it. The code was written in Java and was developed on a Compaq Deskpro using an editor named TextPad that came on the CD included in the Core Java book.

The main purpose of having this simulator is to debug code. I wanted the simulator to be a GUI and extremely portable. Thus, I chose to implement the simulator in Java. The simulator will simulate all actions of the CPU, allow users to change memory values, issue interrupts, and determine what the chipset would return given an IORECV command.

All values must be entered with the maximum number of significant digits. This is a limitation of the simulator but must be dealt with. For example address 0 must be entered into the text boxes as 0000, address 12f as 012f, value c as 0c, and so on. All addresses are two bytes and all data values are one byte. Also, all values are in hex.

The simulator expects a text file that the Arbadell assembler creates.

### **Button Descriptions and Actions**

Starting at the top the first two buttons on the left are SET INT0 and SET INT1. These buttons will act as the PIC chipset sending interrupts to the CPU. Once clicked the interrupt is set and will be cleared by an RETI command. Next is a button that will change the value of the PC. The value that the PC will be changed to is the value in the text box to the right of the button. To the right of that is the W button that will change the value of W. Again, the value that the W register will be changed to is the value in the text field to the right of the button. The Edit Memory button will change the value of any memory location. The first text field is the address to be changed and the second text field is the value to be changed to.

The next row of buttons just beneath the first contains the text field iorecv to the left. The value in this text field is what will be placed in the W register when an IORECV command is executed. In terms of simulation this would be the value the PIC chipset would give to the CPU. To the right of that is a button called Memory starting value. This button will change the starting value of the text memory dump just beneath the button the value in the text box to the right of the button. Lastly, two buttons up and down will move the text memory dump up or down one line.

At the very bottom are buttons that execute the actual simulation. The first on the far left is the step button which will execute the next instruction. The button to the right labeled Execute until will execute instructions until the PC equals the value in the text box to the right of the button. This will allow the user to simulate many instructions in row instead of hitting the step button many times. Lastly, to the right is the reset button which will reset the CPU.

### **Text Descriptions**

To the far left the CPU registers are displayed. They will be updated after every instruction is executed. Below that is a list of the entire stack. It also, will be updated after every instruction. Also, to the left there is the interrupt status. Once the interrupt buttons are clicked the text will change to reflect that an interrupt is pending.

In the middle is a memory dump that will display the memory. To the right is a list of the instructions pointed to by the PC. Notice that they will not change if the Memory starting value is changed.

## Menu Descriptions

The menu is ridiculously simple. There is only FILE. FILE contains EXIT and OPEN. EXIT exits the program. OPEN opens a file. The simulator takes in the text file that the Arbadell assembler creates.

## Execution

To execute the Simulator you will need a Java Development Kit (JDK). This can be found at [www.sun.com](http://www.sun.com). To compile the code type `javac Arbadell.java`. To run the code type `java Arbadell`. This code was developed using java 1.3.0. That's it your in business. Enjoy!!!