Computer Organization Project – Documentation

# Simulator

## General Purpose

# Assembler

## General Purpose

This executable is used to translate instructions in assembly syntax into machine code readable by a CPU (in this context, the Simulator). Calling the program via CLI interface is as follows:

asm.exe program.asm imemin.txt dmemin.txt

where program.asm contains the input assembly code, imemin.txt and dmemin.txt will contain the instructions memory map and the data memory map, respectively, at the end of the program.

## Implementation

The program flow is linear, and will be described with a simple list:

1. First the program iterates over program.asm line by line, recording every label it runs into and conducting each “.word” instruction present within the code, writing to dmemin.txt the instructed values to desired addresses.
2. Using the previously mentioned label record, the program iterates over program.asm line by line again, this time each line (if applicable) is parsed into an Instruction struct with each textual field parsed into a numeric counterpart, including label values in imm1 and imm2.
3. For each Instruction instance created, the program calls encode\_instruction(…) to write the “binary” data (actually a string representation of the hexadecimal value) to imemin.txt.
4. At the end of the program all dynamically allocated memory is freed, and the program terminates.

## Functions