ARMLite Documentation Executive Summary

The purpose of this document is to provide a brief and high-level overview of ARMLite, the instruction set developed in the attached program. The attached program creates an environment that simulates the running of the ASM code.

As the name would suggest, ARMLite is a limited and modest assembly language that is inspired by ARM in terms of register architecture and naming conventions but also has some design choices that separate it from ARM and other languages. Below is a high-level overview of the instructions ARMLite can accept and the syntax used when writing the assembly code.

ARMLite Insturction Set

MOV - Takes an immediate value (#I) and stores the stores it in a register (R0).
Ex) MOV, R0, #10

ADDR - Adds two registers together and then stores it back in the register that is the second argument (R1).
Ex) ADDR, R1, R2

ADDI - Adds the contents of a register (R1) and an immediate value (#I), and stores it back into the same register.
Ex) ADDI, R1, #10

CMP - Compares the contents of two registers together and stores the result of that compare elsewhere. This will be referenced by BEQ below.
Ex) CMP, R1, R2

BEQ - The syntax for BEQ requires the second argument to reference the instruction that the program should branch to if it does so. If no branch happens, the instruction below BEQ will be executed.
Ex) BEQ, 3 (If the CMP was true, branch to instruction #3)

B - Immediate branch to some instruction in the program. The second argument should reference what instruction to branch to.
Ex) B, 3 (Branch to instruction #3)

SWI - Software Interrupt. Currently, SWI is simply used to exit the program and should be jumped to when the program is complete. The second argument should always be 1111
Ex) SWI, 1111