Abstract (Easy) Machine Instructions

Halt	halt
I/O	write <argument> read_boolean <address> read_integer <address>

Jump	pc := <address> if <argument>
	pc := <address> if not <argument>
	pc := <address>

Move	<register> := <register> contents <address> := contents <address>
Immediate Data *Move*	<argument> := <number> <argument> := - <number>
Load	<register> := contents <address>
Store	contents <address> := <register>
ALU Arithmetic	<register> := <register> = <register> <register> := <register> /= <register> <register> := <register> < <register> <register> := <register> <= <register>
Argument Arithmetic	<register> := <argument> and <argument> <register> := <argument> or <argument> <register> := not <argument> <register> := <argument> + <argument> <register> := <argument> - <argument> <register> := <argument> * <argument> <register> := <argument> / <argument> <register> := - <argument>

Arguments

<register>
contents <address>

Registers

b	*Base (in reality: R0)*
r <number>	*Regular*

Abstract Machine Instructions

Addresses (use combinations)

b	*Base*
r <number>	*Register*
<number>	*Displacement*

Allowed combinations:
b
b, r <number>
b, r <number>, <number>
r <number>
r <number>, <number>
b, <number>
<number>