

# ArmSim Clone User Manual

Welcome to the **ArmSim Clone**, a web-based ARM assembly simulator. This guide will help you get started with writing, assembling, and executing ARM assembly code.

## Interface Overview

The interface is divided into four main sections to help you code and debug efficiently.



## Legend

1. **Code Editor:** This is where you write your ARM assembly code. You can also load preset examples from the dropdown menu at the top right of this section.
2. **Controls:** Use these buttons to control execution.
  - **Run:** Execute the program continuously.
  - **Step:** Execute one instruction at a time (useful for debugging).

- **Reset:** Stop execution and reset the CPU and Memory to their initial state.
- 3. **Registers:** View the current values of all CPU registers (R0-R15). Values update in real-time as you step through the code.
- 4. **Memory:** Inspect the contents of the system memory. You can scroll through addresses or jump to a specific address using the input field.

## Detailed Controls Guide

### Using the "Step" Button

The **Step** button allows you to execute your program one instruction at a time. This is the best way to understand how your code works and find errors.

#### How to use:

1. Load or write your code. The first line will be highlighted.
2. Click **Step**.
3. **Action:** The simulator executes the *currently highlighted* instruction.
4. **Result:**
  - The **Registers** panel updates immediately if the instruction modified a register (e.g., MOV, ADD).
  - The **Memory** panel updates if the instruction wrote to memory (STR).
  - The highlighter moves to the *next* instruction to be executed.
5. Repeat to trace the entire program flow.

## Using the "Reset" Button

The **Reset** button wipes the slate clean.

### When to use:

- You want to restart your program from the beginning.
- Your program entered an infinite loop.
- You want to clear modifications made to Memory or Registers.

### Effect:

- **Registers:** All registers (R0-R15) are set back to 0.
- **Memory:** All memory values are cleared to 0.
- **Program Counter (PC):** Reset to 0 (Line 1).
- **Editor:** The highlighter returns to the first line, ready for execution.

## Using the "Run" Button

The **Run** button executes the program continuously until it finishes or hits a breakpoint.

### How to use:

1. Click **Run**.
2. The program executes rapidly. The "Step" button is disabled while running.
3. To stop execution manually, click **Pause** (the button toggles).

## Common Usage Scenarios

### Scenario 1: Debugging a Calculation

**Goal:** Verify that ADD R0, R1, R2 produces the correct result.

1. Write your code.
2. Click **Step** repeatedly to reach the ADD line.
3. Check the "Registers" panel to verify R1 and R2 have the expected values *before* execution.
4. Click **Step** one more time to execute the ADD.
5. Check "Registers" again. R0 should now contain the sum.

## Scenario 2: Verifying Memory Writes

**Goal:** Check if value 0xFF is written to address 0x0010.

1. Write code: MOV R0, #255 followed by STR R0, [R1] (assuming R1 is 0x10).
2. **Step** through the code until past the STR instruction.
3. Look at the **Memory** panel (Section 4).
4. Find the row for 0x0010.
5. Verify the cell contains FF (hex).

## Supported Instructions

The simulator supports a subset of ARM instructions including:

- **Data Processing:** MOV, ADD, SUB, MUL, AND, ORR, EOR
- **Shifts:** LSL, LSR
- **Memory:** LDR, STR
- **Branching:** B, BEQ, BNE, CMP

Enjoy coding!