# Machine Code

# Machine Language Programming

## Definition

**Machine code** is code written in machine language instructions.

- Machine code can be loaded directly into memory for execution.
  1. Load the binary machine instructions into memory.
  2. Set the PC to the address of the first instruction.
- Machine code programs need not be interpreted or compiled.

## Example

In modern computers, the **boot loader** is responsible for loading the first program into memory when the computer is turned on. The **operating system** then handles loading of user programs.

# Machine Language Programming

- When loading a file, the LC-3 simulator assumes that the first $16$-bit value is the desired initial address.
- By convention, the initial address is typically `0x3000`.

## Example

Consider the following machine code program to increment the value stored at memory location `0x3010`:

```
1 | 0011 0000 0000 0000
2 | 0010 0000 0000 1111
3 | 0001 0000 0010 0001
4 | 0011 0000 0000 1101
```

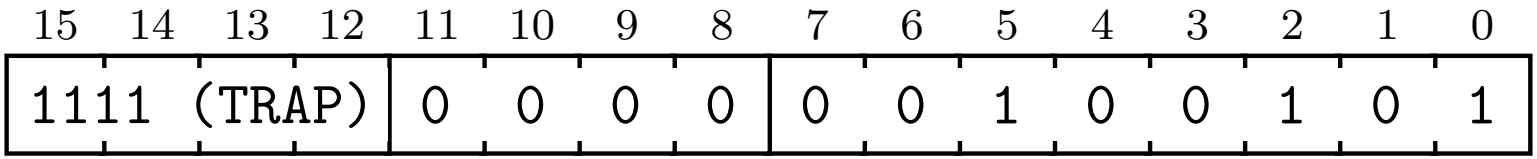
# Machine Language Programming

- Note that a machine code program is a binary file.
  - Instructions are encoded as '0' and '1' bits.
  - Instructions are *not* written with '0' and '1' characters.
- The LC-3 simulator includes a text-to-binary converter.
  - All whitespace is ignored.
  - Anything following a ';' is ignored.

## Example *(cont.)*

```
1 ; Increments the value at 0x3010.
2
3 0011 0000 0000 0000      ; Start at 0x3000.
4 0010 000 000001111       ; Load 0x3010 into R0.
5 0001 000 000 1 00001     ; Increment R0.
6 0011 000 000001101       ; Store R0 at 0x3010.
```

# HALT (System Call, Halt)




1. Request that the instruction cycle be stopped.

## Example *(cont.)*

```
1 ; Increments the value at 0x3010.
2
3 0011 0000 0000 0000      ; Start at 0x3000.
4 0010 000 000001111       ; Load 0x3010 into R0.
5 0001 000 000 1 00001     ; Increment R0.
6 0011 000 000001101       ; Store R0 at 0x3010.
7 1111 0000 00100101       ; Halt.
```

...the above is a complete LC-3 program.

# "if" Statements

Consider the following problem:

- Given an integer at 0x3010, find the sign of that integer.

## Example

```
1  0011 0000 0000 0000     ; Start at 0x3000.
2  0010 000 000001111      ; Load 0x3010 into R0.
3  0000 011 000000011      ; If negative...
4  0101 000 000 1 00000    ; ...clear R0...
5  0001 000 000 1 11111    ; ...set R0 to #-1.
6  0000 111 000000011      ; ...else...
7  0000 010 000000010      ; ...if positive...
8  0101 000 000 1 00000    ; ...clear R0...
9  0001 000 000 1 00001    ; ...set R0 to #1.
10 0011 000 000000111      ; Store R0 at 0x3010.
11 1111 0000 00100101      ; Halt.
```

# "while" Loops

Consider the following problem:

- Given positive integers in R0 and R1, compute R2 = R0 * R1.

## Example

```
1 | 0011 0000 0000 0000      ; Start at 0x3000.
2 | 0101 010 010 1 00000     ; Clear R2.
3 | 0001 011 001 1 00000     ; Copy R1 into R3.
4 | 0000 110 000000011       ; While positive...
5 | 0001 010 010 000 000     ; ...add R0 to R2...
6 | 0001 011 011 1 11111     ; ...decrement R3...
7 | 0000 111 111111100       ; ...loop back.
8 | 1111 0000 00100101       ; Halt.
```

- Note that HALT may alter the values of registers.