

The LEA Instruction

The High Level

The **Load Effective Address** instruction is a type of **Data Movement Instruction** that loads the address of a label from the assembler's **Symbol Table** into a register. (For those of you who love pointers, it returns a pointer to a label.)

To understand the content of this tutorial, you should know what a **PCOffset9** is.

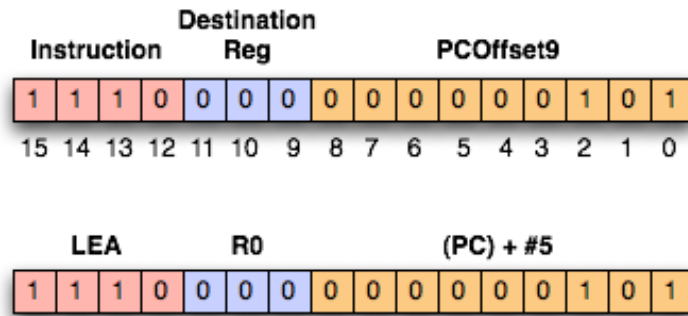


Figure 1: The LEA Instruction (details)

The Breakdown:

- Bits [15-12] specify what instruction to execute (1110 is LEA)
- Bits [11-9] specify which register to store the label's address in
- Bits [8-0] specify a 9-bit PCOffset, which is the number of memory locations (i.e. "spaces") between the PC and the label whose address you want

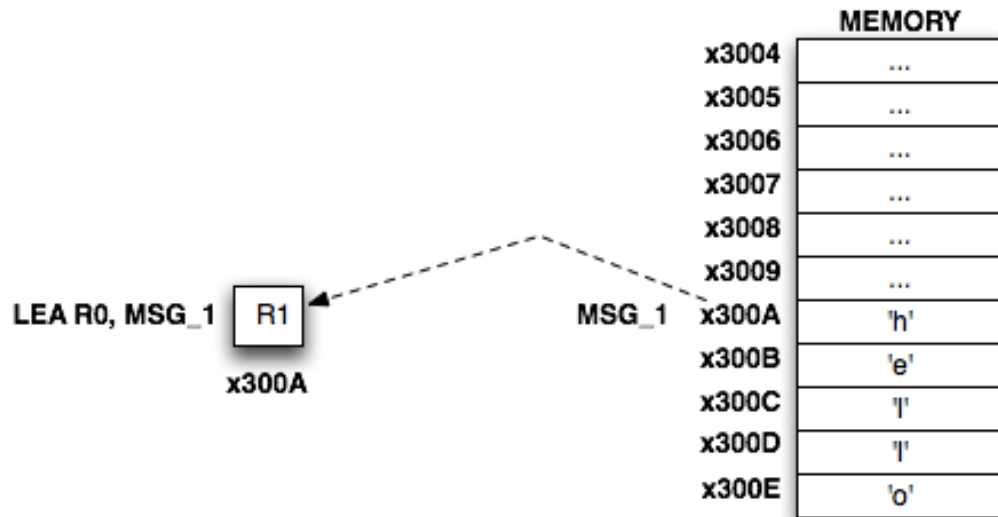


Figure 2: The LEA Instruction – Visual Execution

The Examples!

```
.orig x3000                ; program begins here
;-----
; Instructions
;-----
LEA R0, MSG_1              ; R0 <-- x300A                @Memory: x3000
LEA R1, ARRAY_1            ; R1 <-- x3005                @Memory: x3001
LEA R2, RESERVED_LOCATION ; R2 <-- x3004                @Memory: x3002
HALT                      ; Stop execution of program    @Memory: x3003

;-----
; Data
;-----
RESERVED_LOCATION .BLKW    #1                @Memory: x3004
ARRAY_1           .BLKW    #5                @Memory: x3005 - x3009
MSG_1             .STRINGZ "Hello world!"    @Memory: x300A - x3016
.end
```

Pitfalls... (aka: Erroneous code makes baby turtles cry)

The example below is erroneous. Please do NOT try to code this way!

```
LEA R1, x4000 ; (ERROR: You must use a label, not a literal memory address)
```

```
LEA R1, LabelThatIsReallyFarAway ; (ERROR: Overflows 9-bit PCOffset9 field)
```

The first example pitfall code above is **incorrect** because you have to use a **label** whenever you use the LEA instruction. You cannot give the instruction an address. It's just not built that way.

The second example pitfall code above is **incorrect** and classically causes problems for **many** students. Since this instruction uses a 9-bit **PC Offset**, which means the label whose address you want to load must be no more than [Range of 9 bits] memory locations away from the PC (Note: The range of a Two's Complement 9-bit field is \pm [Range of 8 bits] == [-256, 255]).