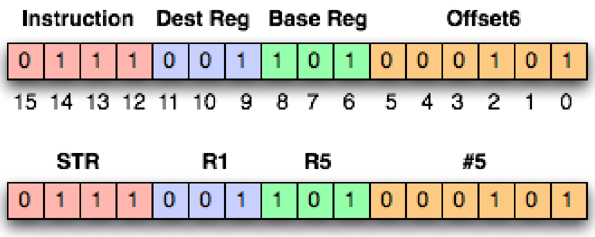
The STR Instruction

The High Level

The **Store Relative** instruction is a type of **Data Movement Instruction** that, given a **Base Register** and an **Offset**, stores the value of a register into MEM[BaseRegister + Offset].

To understand the content of this tutorial, you should know what a **PCOffset9** is, as well as how the **ST** instruction works.



Instruction				Dest Reg			Base Reg			Offset6					
0	1	1	1	0	0	1	1	0	1	0	0	0	1	0	1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

STR				R1			R5			#5					
0	1	1	1	0	0	1	1	0	1	0	0	0	1	0	1

Figure 1: The LDR Instruction (details)

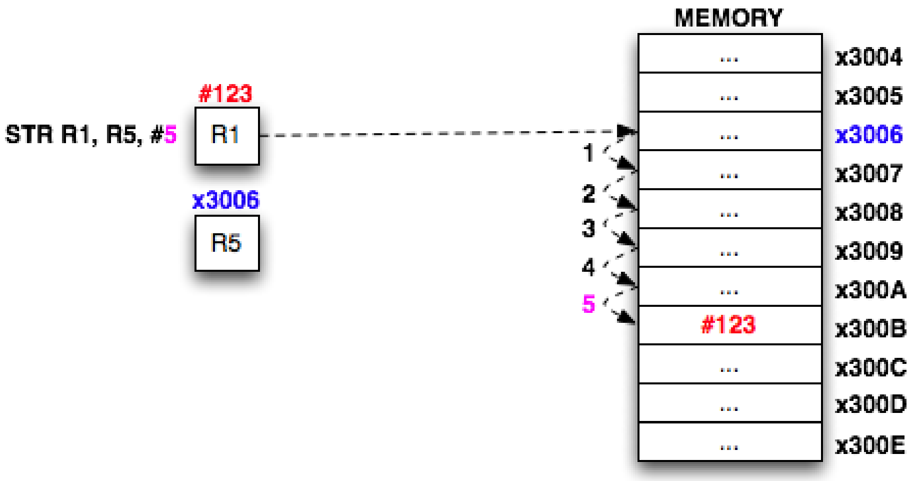


Figure 2: The LDR Instruction – Visual Execution

The Breakdown:

- Bits [15-12] specify what instruction to execute (0111 is STI)
- Bits [11-9] specify which register to store the value in
- Bits [8-6] specify a Base Register
- Bits [5-0] specify a Two's Compliment 6-bit offset.

The Examples!

```
.orig x3000
   ;---------------
   ; Instructions
   ;---------------
   LD R1, DEC_22     ; R1 <-- #22
   LD R5, ADDR_1     ; R5 <-- x4000
   STR R1, R5, #0    ; MEM[R5 + #0] == MEM[R5] == MEM[x4000] <-- R1
   HALT

   ;---------
   ; Data
   ;---------
   DEC_22  .FILL  #22
   ADDR_1  .FILL x4000

.end
```

Result:
The result of this program is that the value #22 is stored in memory at location x4000

Pitfalls... (aka: Erroneous code makes baby Teaching Assistants cry)

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

```
STR #5, R1     ; (ERROR: Order must be: LDR [Base Reg], [Offset])
STR R1, x4000  ; (ERROR: You must use a label,not a literal memory address)
STR R1, #32    ; (ERROR: Overflows Two's Compliment 6-bit field)
```

The first example pitfall code above isi incorrect because the order of operands should have been: STR R1, #5

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

The third example pitfall code above is incorrect because it overflows a Two's Compliment 6-bit field. Since 6 bits can represent only numbers in the range $[-32, 31]$, the number #32 is too big to be expressed with 6 bits. Hence, the error.