

Continue... from LEC 21...

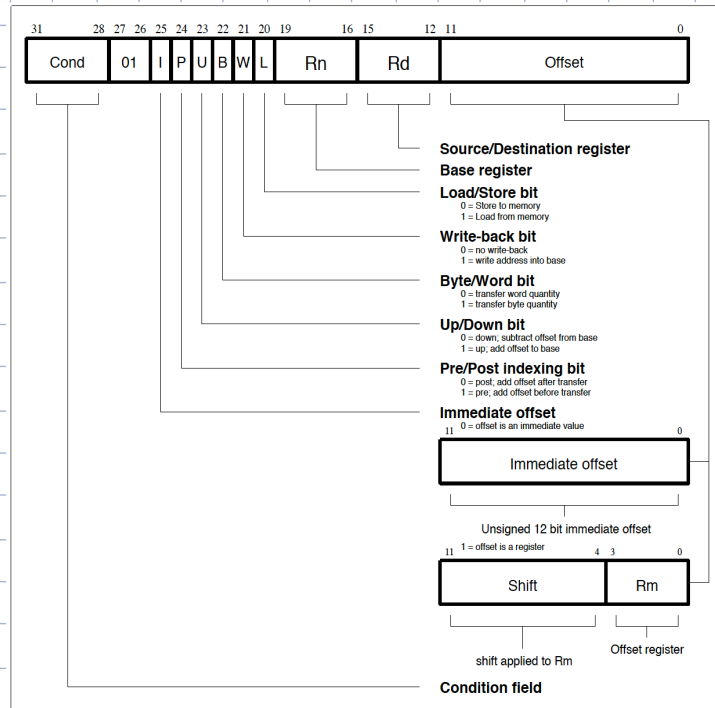
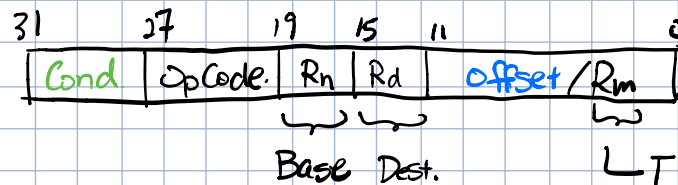


Figure 4-14: Single data transfer instructions

more about Machine Code.

LDR / STR format

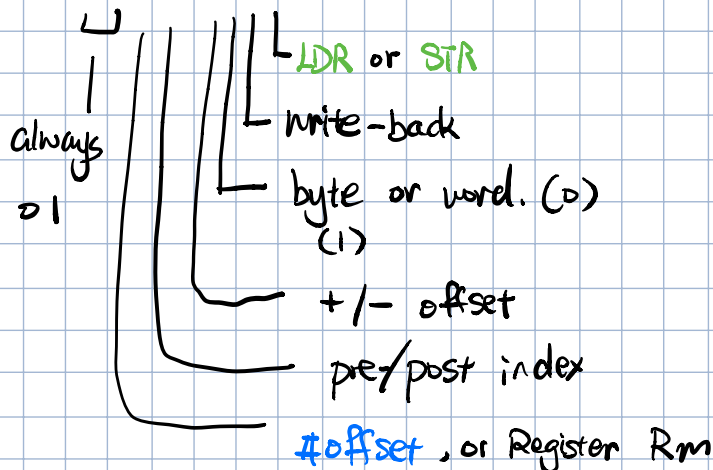


LDR examples

LDR R2, [R1]      1110 0101 1001 0001 0010 00...000

LDR R2, [R1, #4]      1110 0101 1001 0001 0010 00...100

LDR R2, [R1, #4]!      1110 0101 1011 0001 0010 00...100



Question: why not always branch using

mov PC, #label

instead of

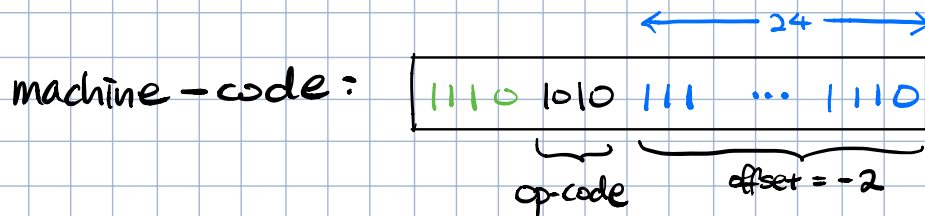
B label

Answer = mov has limited #D range, B has a larger range.

Branch is implemented as  $PC \leftarrow PC + \text{offset}$ , where the offset is the 2's complement value needed to reach the label E.g.:

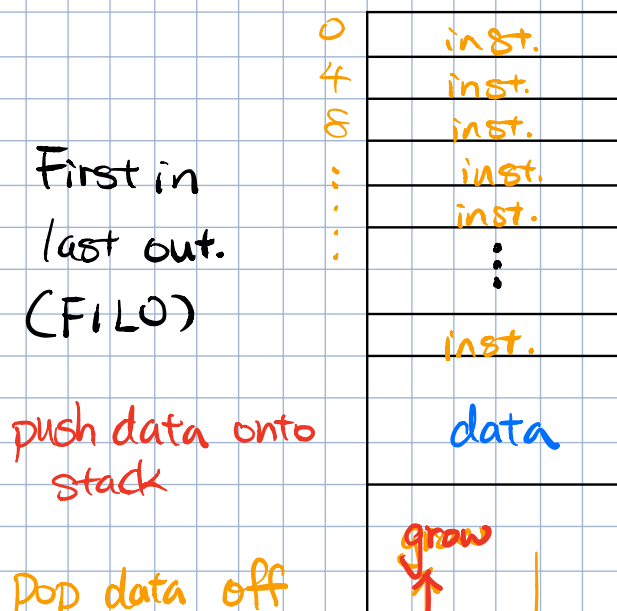
HERE: B HERE

At the time that the ARM processor executes B the PC has already been incremented by 2 words, so, we need an offset of -2 words.



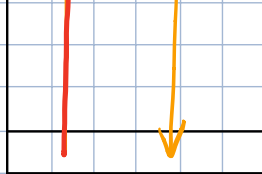
## using the stack

-The stack is an ideal data structure for temp. storage.



stack

bottom of the stack.



top of the stack.



register R13, aka SP, always points to the "top of the stack"

## Example

Note: ARM always push in the order of largest reg. first

PUSH {R4, R5, R6, R9, LR}

.text

.global

- Start

- Start: mov SP, #0x20000

⋮

// code that uses R4, R5, R6, R9 (example)

BL MY\_SUB

⋮

// Subroutine that uses R4, R5, R6, and R9.

MY\_SUB: PUSH {R4, R5, R6, R9, LR} // or {R4-R6, R9, LR}

⋮

POP {R4-R6, R9, PC}

SP = 0x1FFEC  
0x1FFFD  
0x1FFF4  
0x1FFF8  
0x1FFFC  
SP = 0x20000

