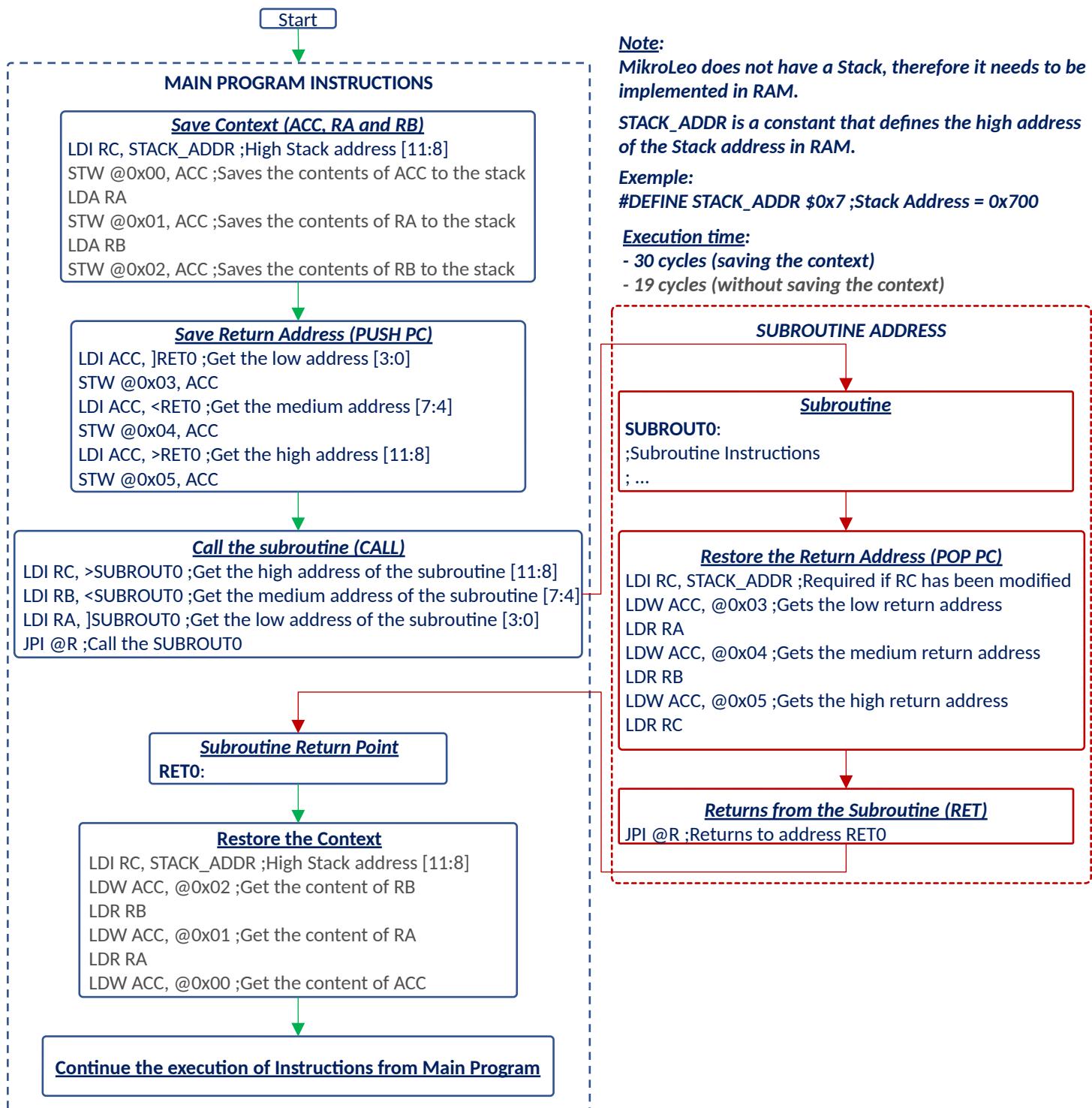


# Where is the stack in the MikroLeo?

## Flowchart and code example for implementing a simple subroutine (1-level Stack) using indirect addressing



### Note:

MikroLeo does not have a Stack, therefore it needs to be implemented in RAM.

STACK\_ADDR is a constant that defines the high address of the Stack address in RAM.

### Exemple:

#DEFINE STACK\_ADDR \$0x7 ;Stack Address = 0x700

### Execution time:

- 30 cycles (saving the context)
- 19 cycles (without saving the context)

SUBROUTINE ADDRESS

Subroutine

SUBROUT0:  
;Subroutine Instructions  
; ...

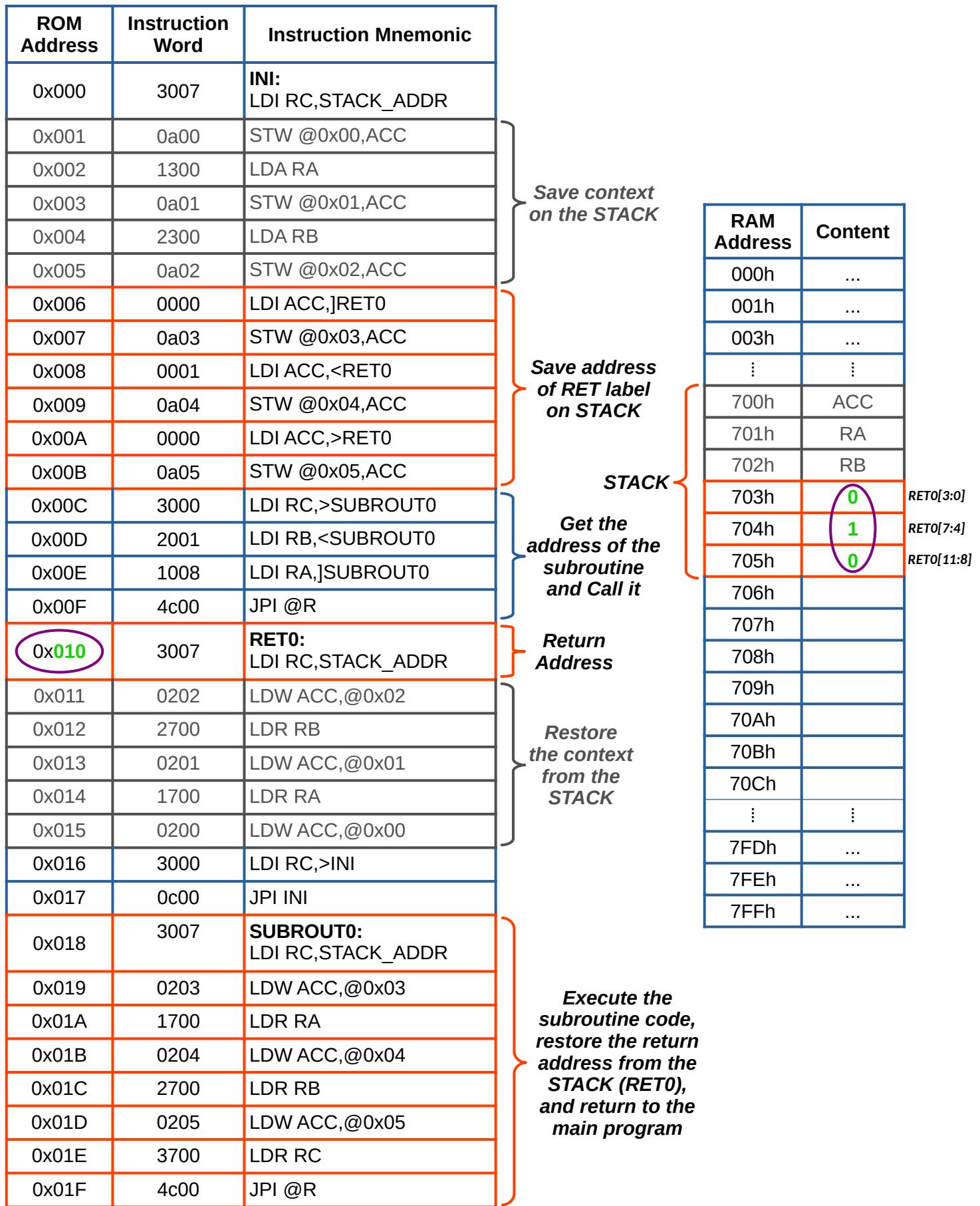
Restore the Return Address (POP PC)

LDI RC, STACK\_ADDR ;Required if RC has been modified  
LDW ACC, @0x03 ;Gets the low return address  
LDR RA  
LDW ACC, @0x04 ;Gets the medium return address  
LDR RB  
LDW ACC, @0x05 ;Gets the high return address  
LDR RC

Returns from the Subroutine (RET)

JPI @R ;Returns to address RET0

## Example of how code is stored in program memory and how a subroutine is manipulated using indirect addressing



Note: Note that in this example, for simplicity, there is no subroutine code. The subroutine code must begin at the address SUBROUT0.