

Lab 7
Owen Monus
CS 301
200482797
Mar 7, 2024

Registers at the end of execution

Registers		Disass
Register	Value	
Core		
R0	0x00000000	0x
R1	0x0000000E	0x
R2	0x0000001E	0x
R3	0x08000034	0x
R4	0x00000000	
R5	0x00000000	
R6	0x00000000	
R7	0x00000000	
R8	0x00000000	
R9	0x00000000	
R10	0x00000000	
R11	0x00000000	
R12	0x00000000	
R13 (SP)	0x20001000	
R14 (LR)	0xFFFFFFFF	
R15 (PC)	0x08000082	
xPSR	0x61000000	
Banked		
System		
Internal		
Mode	Thread	
Privilege	Privileged	
Stack	MSP	
States	1405	
Sec	0.00011708	

Successful build

The screenshot displays the uVision IDE interface. The main window shows the assembly code for `CountVowelsOne.s`. The code includes comments for each instruction, such as `; Compare with 'a'` for `cmp R0, #'a'`. The program logic involves comparing a character in `R0` with vowels 'a', 'e', 'i', 'o', and 'u'. If it's a vowel, it increments `R3` (labeled `increment_char`). If it's not a vowel, it increments `R2` (labeled `not_vowel`). The program ends with `count_done` and `END ;End of the program`.

The `Build Output` window at the bottom shows the following text:

```
*** Using Compiler 'V5.06 update 3 (build 300)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'Target 1'
linking...
Program Size: Code=80 RO-data=56 RW-data=0 ZI-data=0
".\Objects\lab7.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:00
```

The status bar at the bottom indicates the simulation is running, with registers `L:102 C:1` and memory locations `CAP NUM SCRL OVR R/W`.

Code

```
; Count how many vowels and how many non-vowels are in the following string
; "ARM assembly language is important to learn!",0

MOV R1, #0      ; Initialize vowel count to 0
MOV R2, #0      ; Initialize non-vowel count to 0
LDR R3, = string1 ; Load the address of string1 into the register R3

vowel_or_consonant

    LDRB R0, [R3] ; Load the character from string1

    CMP R0, #0 ; check if value is 0
    beq count_done ; null terminated, exit

; uppercase
    cmp R0, #'A' ; Compare with 'A'
    beq is_vowel ; If equal to 'A', it's a vowel
    cmp R0, #'E' ; Compare with 'E'
    beq is_vowel ; If equal to 'E', it's a vowel
    cmp R0, #'I' ; Compare with 'I'
    beq is_vowel ; If equal to 'I', it's a vowel
    cmp R0, #'O' ; Compare with 'O'
    beq is_vowel ; If equal to 'O', it's a vowel
    cmp R0, #'U' ; Compare with 'U'

; lowercase
    cmp R0, #'a' ; Compare with 'a'
    beq is_vowel ; If equal to 'a', it's a vowel
    cmp R0, #'e' ; Compare with 'e'
    beq is_vowel ; If equal to 'e', it's a vowel
    cmp R0, #'i' ; Compare with 'i'
    beq is_vowel ; If equal to 'i', it's a vowel
    cmp R0, #'o' ; Compare with 'o'
    beq is_vowel ; If equal to 'o', it's a vowel
    cmp R0, #'u' ; Compare with 'u'
    beq is_vowel ; If equal to 'u', it's a vowel

    B not_vowel

increment_char
    ADD R3, #1 ; increment index to the next character
    B vowel_or_consonant

not_vowel
    ADD R2, #1
    B increment_char

is_vowel
    ADD R1, #1
    B increment_char

count_done

    | | END ;End of the program
```