# Maxime Baconnais

# CS 435 HW 6/13

Part 1

**4. Start the Debug Session (Ctrl + F5) and run the program until the opening brace in the main function is highlighted. Open the Registers window (View -> Registers Window). What are the values of the stack pointer (r13), link register (r14) and the program counter (r15)?**

R13 = 0x20018000

R14 = 0x080002A9

R15 = 0x080002B0

**5. Open the Disassembly window (View->Disassembly Window). Which instruction does the yellow arrow point to, and what is its address? How does this address relate to the value of pc?**

The yellow arrow points to PUSH {r4,lr} at 0x080002B0. It’s the value of PC.

**6. Step one machine instruction using the F10 key while the Disassembly window is selected. Which two registers have changed (they should be highlighted in the Registers window), and how do they relate to the instruction just executed?**

The registers R13 (SP) and R15 (PC) changed, it’s the stack pointer and the program counter.

**7. Look at the instructions in the Disassembly window. Do you see any instructions which are four bytes long? If so, what are the first two?**

The two first instructions which are four bytes long are BL.W (mbed\_sdk\_init) at 0x080002B2 and BL.W (mbed\_main) at 0x080002B6.

**8. Continue execution (using F10) until reaching the *BL.W my\_strcpy* instruction. What are the values of the SP, PC and LR?**

SP = 0x20017FF8

PC = 0x080002B6

LR = 0x0800030D

**9. Execute the *BL.W* instruction. What are the values of the SP, PS and LR? What has changed and why? Does the PC value agree with what is shown in the Disassembly window?**

SP = 0x20017FF8

PC = 0x080002BA

LR = 0x080002BB

LR and PC changed, because the instruction was executed. The value of PC is the same than the value in the disassembly window.

**10. What registers hold the arguments to *my\_strcpy*, and what are their contents?**

R0 and R1 hold the arguments to my\_strcpy

R0 = 0x00000000

R1 = 0x00000004

**12. What is the value of “a”?**

The value of “a” is "Hello world!".

**13. What is the value of “b”?**

The value of “b” is uninitialized.

**14. Single step through the assembly code watching “Call Stack + Locals” window to see the string being copied character by character from a to b. What register holds the character?**

R2 holds the caracters.

**15. What are the values of the character, the src pointer, the dst pointer, the link register (R14) and the program counter (R15) when the code reaches the last instruction in the subroutine (*BX lr*)?**

LR: 0x08000213

PC: 0x08000214

**16. Execute the *BX lr* instruction. Now what is the value of PC?**

0x08000214

**17. What is the relationship between the PC value and the previous LR value? Explain.**

PC is the return value.

Part 2.

**1. Write a new subroutine in assembly to convert the upper-case letters to lower-case letter.**

\_\_asm void my\_uppercase(char \*str)

{

cap\_loop

LDRB r1, [r0]

CMP r1, #'a'-1

BLS cap\_skip

CMP r1, #'z'

BHI cap\_skip

SUBS r1,#32

STRB r1, [r0]

cap\_skip

ADDS r0, r0, #1

CMP r1, #0

BNE cap\_loop

BX lr

}