

HOMEWORK 3

CMPE 102, Assembly Language Programming, Çokgör

Total points towards grade: 5/100

Homework are submitted electronically on the Canvas. E-mail submissions will not be accepted.

Assembly programming part must be submitted in text format.

Add comments to your code to explain your work. Work that is not commented will not earn full credit.

Use Keil uVision to test your assembly program.

1) Write an assembly program that stores R0, which should have its contents as 0xAABBCCDD, at address 0x20004000. You will notice that by default the contents of R0 will be stored in memory in little endian format (see example on the left below). Now, your program should convert this 32-bit number stored in the memory from little endian to big endian. Make sure the result is saved back in memory (see example on the right below). The figure below shows the outcome before and after conversion.

| Little Endian | |
|----------------|----------------|
| Memory Address | Memory Content |
| 0x20004003 | AA |
| 0x20004002 | BB |
| 0x20004001 | CC |
| 0x20004000 | DD |

| After conversion: Big Endian | |
|---------------------------------|----------------|
| Memory Address | Memory Content |
| 0x20004003 | DD |
| 0x20004002 | CC |
| 0x20004001 | BB |
| 0x20004000 | AA |

Submit your assembly code in text format.

(2.5 points)

2) An array of ten 32-bit numbers is stored in registers R1 to R10 (store numbers 1 to 10 in registers R1 to R10; e.g. R1: 0x1; R2: 0x2, etc.)). Store this array to memory starting from address 0x20004000 using a single STM (store multiple registers) instruction. Write an assembly program to add all elements of the array and store the result in R0. Use post-index addressing to access the array elements in memory, i.e., the array elements should be retrieved from memory using post-index addressing.

Submit your assembly code in text format.

(2.5 points)