Assignment #4

Started: Jul 17 at 4:46pm

Quiz Instructions

Question 1 — 2 pts

1. Find out what the following assembly code calculates. what is the final value in r0 (**in decimal**)?

```
    AREA my code, CODE
     EXPORT __main
     ALIGN
     ENTRY
__main PROC

     MOVS  r0,#0
     MOVS  r1,#15
     MOVS  r2,#0

loop
     CMP   r2,r1
     BGT   stop
     MLA   r0,r2,r2,r0
     ADDS  r2,r2,#1
     B     loop

     ; The final result is saved in register r0
```

```
stop
    ENDP
    END
```

Question 2

2 pts

1. Translate the following C program into an assembly program. The C program finds the minimal value of three signed integers. Assume *a*, *b*, and *c* is stored in register r0, r1, and r2, respectively. The result min is saved in register r3.

```
if (a < b && a < c) {
    min = a;
} else if (b < a && b < c){
    min = b;
} else {
    min = c;
}
```

Edit View Insert Format Tools Table

12pt Paragraph B I U A T²

p 0 words </>

Question 3 1 pts

1. Suppose r0 = 0x20008000. From which address will r7 be stored in the following instructions? What is the value of r0 after executing each instruction? Assume each instruction is being executed separately, i.e., they are not part of a program.

Please enter the result in Hex format starting with 0x.

1. STMIA r0!, {r3, r9, r7, r1, r2}

Hint: value of r2 = **0x20008004**

r0 = _____ **and r7 =** _____ **in Hex**

2. STMIB r0!, {r3, r9, r7, r1, r2}

r0 = _____ **and r7 =** _____ **in Hex**

Question 4 2 pts

Write an assembly program that converts a 32-bit integer stored in the memory from little endian to big endian, without using the REV instruction. Make sure the result is saved back to the memory.

For example:

If, Little-endian: **0x01020304**

Then, Big-endian: **0x04030201**

; Using REV instruction

REV r2, r0

; Write your code without using REV instruction

Edit View Insert Format Tools Table

12pt Paragraph

p 0 words

No new data to save. Last checked at 5:48pm Submit Quiz