

---

Name: \_\_\_\_\_ ID: \_\_\_\_\_

EL6483 Real Time Embedded Systems

QUIZ 1

March 13, 2019

---

1. Consider the Hailstone sequence that is defined by the following rules:
  - a. Start off with a positive number,  $n$ .
  - b. If the  $n$  is 1 then the sequence ends.
  - c. If  $n$  is even then the next  $n$  of the sequence =  $n/2$ .
  - d. If  $n$  is odd then the next  $n$  of the sequence =  $(3 * n) + 1$ .

Write a segment of ARM assembly (see below) that will calculate the Hailstone sequence for  $n=3$ . Comment each line and specify the purpose of each register.

```
mov R1, #3 ; initialize n = 3
...
...
halt B halt
```

1. The algorithm for encoding an RGB color triplet to a single value is as follows:

$$(\text{long})\text{rgb} = (\text{blue} * 256 * 256) + (\text{green} * 256) + \text{red}$$

Write the C code to extract the red part, green part and blue part of the long rgb.

```
(unsigned char)red = ???;
(unsigned char)green = ???;
(unsigned char)blue = ???;
```

2. Consider the following ARM Assembly. Notice there are 2 functions, **\_\_main** and **MyFunc**. Descriptively comment each line between “\_\_main PROC” and “END” of the code and state what this program does. Also comment on the function of each register used.

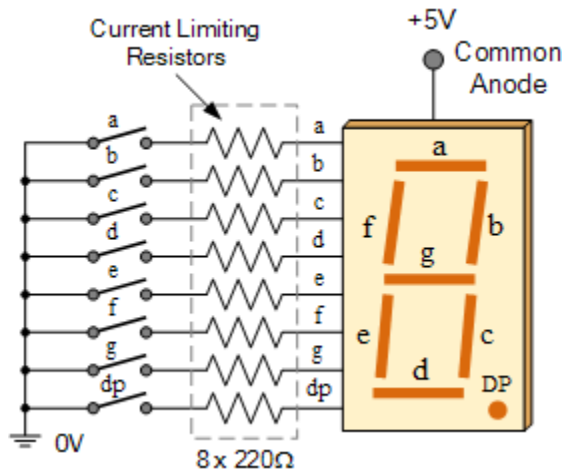
```
                AREA myData, DATA
str             DCB  "Less is more", 0

                AREA MyCode, CODE
                EXPORT __main
                ALIGN
                ENTRY

__main          PROC
                LDR      r0, =str
                MOVS     r1, #'s'
                BL       MyFunc ; r0 and r1 are arguments
stop            B        stop
                ENDP

MyFunc          PROC
                MOV      r2, r0
loop            LDRB     r3, [r0]
                CBZ      r3, exit
                CMP      r3, r1
                LDRBNE    r3, [r0]
                STRBNE    r3, [r2]
                ADDNE     r2, r2, #1
                ADD      r0, r0, #1
                B         loop
exit            STRB     r3, [r2]
                BX       lr
                ENDP
                END
```

3. Suppose you would like to use PORTC of our controller to drive a 7 segment LED display. A typical **common anode** hardware setup is shown below.



Write the C code to setup PORTx\_PCRx for each pin and GPIOC\_DDR registers to enable this functionality. Be sure to specify the pins used for each LED segment (a-g and DP). Also assume the LEDs need high drive strength.

Write a C function named SevenSeg() that takes in an int from 0 to 9 (ignore the DP) and illuminates the display accordingly by modifying the GPIOC\_Pxxx registers.

Extra Credit: Write a C function named SevenSeg() that takes in a short returned from an ADC channel (10 bit,  $0 - 2^{10} - 1$ ) and illuminates the display to represent the nearest rounded integer corresponding to the voltage level, assuming the ADC is set up for measuring between 0 and +5V.