ECE319K Fall 2024 Exam1 Version A

UT EID: _____ First: _____ Last: _____

Instructions:

- Closed book and closed notes. No books, no papers, no data sheets (other than the addendum)
- No devices other than pencil, pen, eraser (no calculators, no electronic devices), please turn cell phones off.
- Please be sure that your answers to all questions (and all supporting work that is required) are contained in the space (boxes) provided. *Anything outside the boxes/blanks will be ignored in grading*. You may use the back of the sheets for scratch work.
- You have 75 minutes, so allocate your time accordingly.
- For all questions, unless otherwise stated, find the most efficient (time, resources) solution.
- Unless otherwise stated, make all I/O accesses friendly and all subroutines AAPCS compliant.
- *Please read the entire exam before starting.*

(20) Question 1a.

(6) Part a) Specify the op code for **XXX** to perform the equivalent C operation. **z** is signed and in R3,
The assembly instructions are:

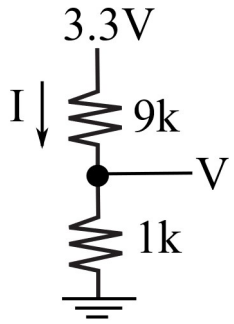
```
MOVS R4,#0x03
XXX  R3,R3,R4
```

XXX	C Operation
	z = z / 8;
	z = z * 3;
	z = z ^ 0x03;
	z = z | 0x03;
	z = z & (~0x03);
	z = z & 0x03;

(3) Part b) The initial value of **z** is 17=0x11=0b10001
What is the resulting value of **z** after all 6 lines of C are executed? …

(5) Part c) What is the value of R1 after these instructions are executed. Give your answer in hexadecimal? ……

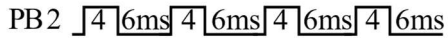
```
        .text
        .align 2
Data:  .byte 0x50,0x60,0x70,0x80,0x90,0xA0,0xB0,0xC0
main:   LDR  R2,=Data
        LDRH R1,[R2,#4]
```



(3) Part d) What is V? …….

(3) Part e) What is I? ……

(5) Question 2a. There is a positive logic LED connected to PB2. The LED voltage is 2V and the current is 5mA. The software creates this output.



How much power, in mW, is delivered to the LED? Show your work.

(5) Question 3a. Fill the missing opcode (**XXX**) that is needed to implement each of the corresponding C if statements. Assume **x** is unsigned.

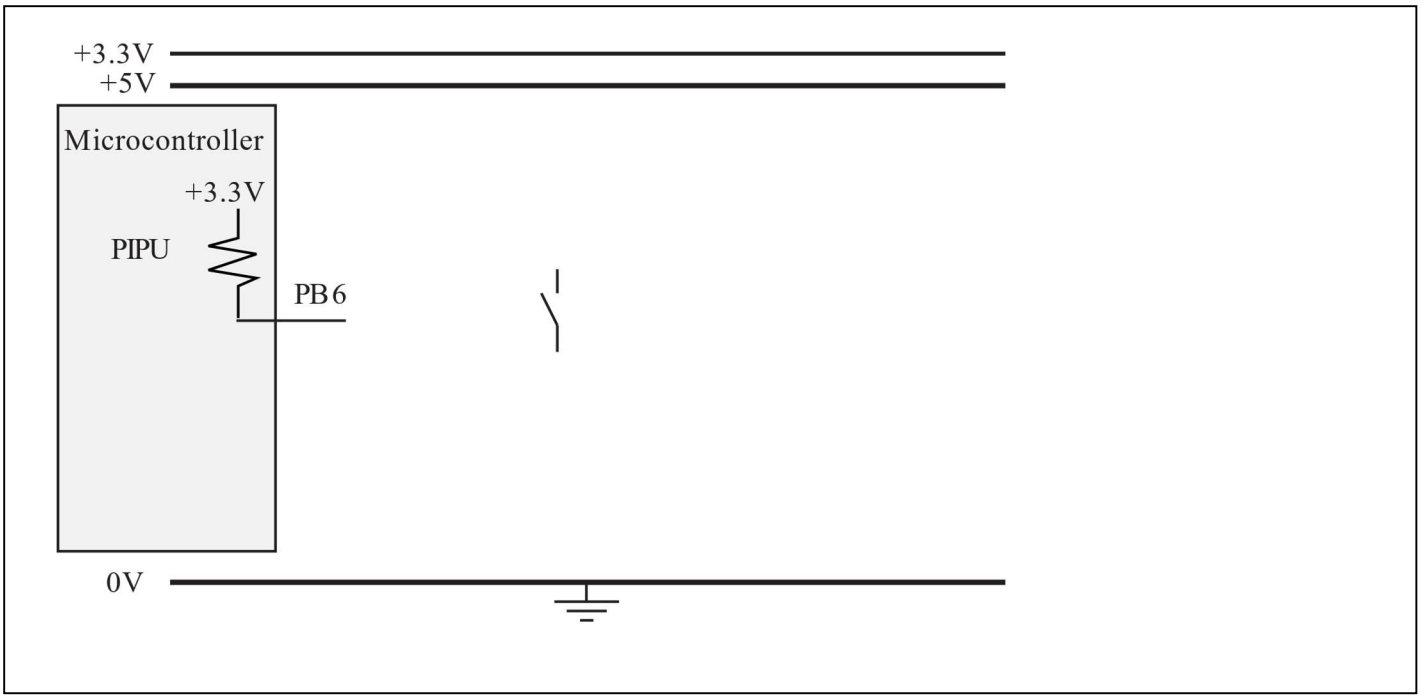
```
   LDR R0,=x
   LDR R1,[R0]
   CMP R1,#10
   XXX Skip
   BL  Thing
Skip:
```

XXX	C Operation
	`if(x==10){Thing();}`
	`if(x>10){Thing();}`
	`if(x<=10){Thing();}`

(15) Question 4a. Convert the assembly to C box by box. All values are 32-bit unsigned. The assembly code follows AAPCS. Give the local variable in R1 the name **m**. Ignore overflow.

`Fun:` `  MOVS R1,#1 // R1 is m`	`uint32_t Fun(uint32_t n){`
`Loop:` `  MULS R1,R1,R0`	
`  SUBS R0,R0,#1`	
`  CMP  R0,#0` `  BNE  Loop`	
`  MOVS R0,R1` `  BX   LR`	

(10) Question 5a. Interface a **switch** to PB6 using NO resistors. To interface the switch without resistors, the software activates internal pull up, **PIPU**. I.e., there is an internal resistor from PB6 to 3.3V. Connect the switch to +5V, +3.3V, 0V, and the microcontroller as needed.



(14) Question 6a. Assume a negative logic LED is interfaced to PB16. PB16 has already been initialized as a GPIO output. Write assembly function that turns the LED on if R0=1, and turns it off if R0=0. Make the code friendly. Follow AAPCS. It is invoked by **`BL LEDout`**

(6) Question 7a Consider debugging the flashing LED in Lab 2. For each debugging procedure, classify as N M or H

N = nonintrusive,
M = minimally intrusive, or
H = highly intrusive

Part a) Logic analyzer connected to all GPIO pins ---------- [ ]

Part b) Breakpoints ---------- [ ]

Part c) Dump GPIO data into a buffer ---------- [ ]

Part d) Scope connected to GPIO output pin ---------- [ ]

Part e) Black box testing: running code and looking at the LED with your eyes -- [ ]

Part f) Single stepping ---------- [ ]

(10) Question 8a. Show the contents of the stack at the marked points respectively in the execution of the following code. The initial stack pointer is **0x20201008**.

```
0x00002000      MOVS R0,#0
0x00002002      MOVS R1,#1
0x00002004      PUSH {R1,R0}
0x00002006      MOVS R4,#4
0x00002008      BL   Func
0x0000200A      POP  {R0,R1}
…
            Func:
0x00002042      PUSH {R4,LR}
0x00002044      MOVS R4,#10
0x00002046      MULS R0,R0,R4 // <----A) draw stack
0x00002048      POP  {R4,PC}
```

Give value of SP and the contents of the stack after execution point A):

0x20200FF8	
0x20200FFC	
0x20201000	
0x20201004	
0x20201008	
0x2020100C	
0x20201010	
0x20201014	
0x20201018	

SP=

Specify a value for each of the 9 boxes on the stack. Place **XXX** *in any box for which you do not know its contents, or is not guaranteed to be valid.*

(15) Question 9a. Write a C function called **Min** that searches an array and returns the minimum value. If the array is empty return 255. Each element is 8-bit unsigned. The arrays are terminated with sentinel 255. The prototype and main program cannot be changed.

```
uint8_t Min(uint8_t buf[]);
uint8_t buffer1[5]={4,3,7,8,255};
uint8_t buffer2[8]={7,6,5,4,3,2,1,255};
uint8_t buffer3[1]={255};
int main(void){
  uint8_t ans1 = Min(buffer1); // answer is 3
  uint8_t ans2 = Min(buffer2); // answer is 1
  uint8_t ans3 = Min(buffer3); // answer is 255
  while(1){}
}
```

```
uint8_t Min(uint8_t buf[]){
```