McGill University Department of Electrical and Computer Engineering

ECSE 324 Mid-Term Exam Winter 2020

Version A

Name: ____

ID:

Closed Book Exam. Department approved calculators allowed. Answer all questions directly on the question paper. You can do rough work on the back of the paper and on the blank pages. Rough work will not be used for grading.

Part 1_______________/10

Part 2 ______________/05

Part 3 ______________/10

Part 1) (10 Marks) Short Questions 1 mark each (General Knowledge): Circle your choice, if more than one is circled you will get zero for that question.

1.1 The number of bytes in a gigabyte is:

a)	b)	c)	d)	e)
2^{32}	2^{30}	2^{20}	2^{40}	None of These

1.2 The binary representation of the number -5 in 2's complement is:

a)	b)	c)	d)	e)
1010	1011	0101	1001	None of These

1.3 Assume a word-aligned machine. Given the instruction "LDR R1 [R2]", which value of R2 is legal:

a)	b)	c)	d)	e)
0x1000	0x00ff	0x0003	0x0002	None of These

1.4 For RISC architectures the:

a)	b)	c)	d)	e)
ALU only gets data from registers	Instructions are of different lengths	Program counter is incremented by one byte after each instruction.	All of these	None of These

1.5 The ARM register CPSR (Current program Status register) contains:

a)	b)	c)	d)	e)
Next Address for PC	The Return Value of the stack	The Zero flag	The relative address of the branch	None of These

1.6 Consider the following instruction: "LDR R2, [R6, #4]!" . After the instruction executes:

a)	b)	c)	d)	e)
R2 is unmodified	R6 is unmodified	R6 is incremented by 16	R2 is incremented by 4	None of These

1.7 The stack pointer register:

a)	b)	c)	d)	e)
Is used to implement function calls	Cannot be modified by an instruction	Contains a return address	All of these	None of These

1.8 Assuming the callee-save convention is used, when a function is called in assembly:

a)	b)	c)	d)	e)
The callee pushes all the registers it has used on the stack upon exit	The callee pushes all the registers it will use on the stack upon entrance	The caller pushes all the registers it has used on the stack before the call	The return address is placed in the frame pointer register upon entrance	None of These

1.9 The part of the software tool chain that cleans up the memory after the program is finished is the:

a)	b)	c)	d)	e)
Compiler	Assembler	Linker	Operating System	None of These

1.10 The symbol table in an object file:

a)	b)	c)	d)	e)
Contains the addresses of the labels found in the assembly program	Contains the addresses of the instructions found in the assembly program	Can only be determined after the second pass has run in the assembler.	All of These	None of These

Part 2) (5 Marks) Problem Analysis.

Assume a 16-bits RISC CPU with 8 general purpose registers (including a stack pointer register), a PC (Program Counter) register, and a CPSR (Current program Status register). Assume that the only instructions available in the instruction set are:

- ADD Rd, Rsource1, Rsource2 // Rd <- [Rsource1] + [Rsource2]
- LD Rd, Rbase, Roffset // Rd <- MEM[[Rbase]+[Roffset]]
- ST Rd, Rbase, Roffset // MEM[Rbase+Roffset] <- [Rd]
- CMP Rsource1, Rsource2 // Set Z flag to 1 if Rsource1==Rsource2
- BREQ addr // Branches to the address *addr* if Z flag is 1
- MOV Rd, #imm // Rd <- #imm (immediate value)

2.1 What is the minimum number of bits required to encode the opcode of an instruction?

a)	b)	c)	d)	e)
8 bits	3 bits	4 bits	6 bits	None of These

2.2 Assuming numbers are encoded using 2's complement. Which instruction would you add to the instruction set in order to enable the substraction of two numbers found in two registers?

a)	b)	c)	d)	e)

AND Rd Rs1 Rs2
Rd <- [Rs1]&[Rs2]
	LSL Rd Rs #imm
	Rd <- [Rs] <<#imm
		OR Rd Rs1 Rs2
		Rd <- [Rs1] | [Rs2]
			MVN Rd Rs
			Rd <- not[Rs]
				None of These

2.3 What is the minimum number of instructions you would require to emulate a PUSH stack operation?

a)	b)	c)	d)	e)
1	2	3	4	None of These

2.4 What is the total amount of memory addressable by this CPU?

a)	b)	c)	d)	e)
4GB	1GB	64KB	512KB	None of These

2.5 Assume that you would like to read 1KB of continuous data from memory from address 0x4000. What is the minimum number of LD instructions executed required?

a)	b)	c)	d)	e)
256	1024	128	512	None of These

Part 3) (10 Marks) Assembly Code. These problems require you to fill in the blanks. Unclear answers will be graded as zero.

3.1 (**5 Marks**)

```
_start:        LDR    R0, =AVECTOR
               LDR    R1, =BVECTOR
               LDR    R2, N
               MOV    R3, #0

TASK1:         LDR    R4, [R0,#4]!
               LDR    R5, [R1,#4]!
               CMP    R4, R5
               MUL    R6, R5, R4
               STR    R6, [R1], #4
               BEQ    STOP
               B      TASK4
TASK2:         STR    R6, [R0], #4
TASK4:         SUBS R2,  R2,  #1

               BGT    TASK1
STOP:          B       STOP

N:            .word     3
AVECTOR: .word     3 ,   9 ,   4 , -5 ,   5
BVECTOR: .word   -5 ,   2 ,   4 ,   4 , -3
```

Write out the contents of the memory locations AVECTOR and BVECTOR after the instruction at line STOP is executed in the space provided below.

AVECTOR					
BVECTOR					

3.2 (**5 Marks**) Convert the high-level program into ARM Assembly code by filling out the blank parts. Note that the labels for the assembly code are provided in **bold** (**Line 1** is the label for the first line of the assembly code)

```
// a, b, c integer data
//  convert into assembly the C code code below

  if ( a <= 2)  {
     b = foo(c);
  } else {
     b = c;
  }
```

```
Line1:        LDR   R0, [a]       // a represents location of variable a in memory
Line2:        LDR   R1, [b]       // b represents location of variable b in memory
Line3:        LDR   R2, [c]       // c represents location of variable c in memory

Line4:        _______________

Line5:        BGT _______________

Line6:        ADD R0  _______________

Line7:        BL foo

Line8:        B _______________

Line9:        MOV _______________

Line10:       ST  R0,  [b]
```