SIMON FRASER UNIVERSITY
School of Mechatronic Systems Engineering
MSE 352 Digital Logic and Microcontrollers
Quiz III – Fall 2018

Student Name: Solution

1- Consider the following assembly program.

```
start
main        MOV R0,#1
            MOV R1,#2
            CMP R0,R1
            BLE smaller
greater     MOV R2,#2
            B  follow
smaller     MOV R2,#1
follow      MOV R0, #0
            MOV R1, #0
            MSR CPSR_f,#0     ; This line is an instruction to move a register contents or an
                              ; immediate constant to CPSR. The _f specifies that it should only
                              ; apply to the flags bit field (N,Z,C,V).
stop
```

Now, answer the following questions:

- Which branch is executed (greater or smaller), and why? **(1.5 marks)**

  According to the Condition (less than or equal), "smaller" is executed.

- Which flag(s) do the branch operation BLE, use? **(1.5 marks)**

  Flags N (for checking negative) and flag Z (for Zero)

- What values should the flags have in order for the BLE instruction to cause a branch jump? **(1 marks)**

  $N=1$ or $Z=1$

- Fill the following table. **(2 marks)**

	CPSR				Registers	
Instruction	N	Z	C	V	R0(dec)	R1(dec)
MOV R_0,#1	0	0	0	0	1	0
MOV R_1,#2	0	0	0	0	1	2
CMP R_0,R_1	1	0	0	0	1	2
BLE smaller	1	0	0	0	1	2
MOV R_2,#1	1	0	0	0	1	2
MOV R_0,#0	1	0	0	0	0	2
MOV R_1,#0	1	0	0	0	0	0
MSR CPSR_f,#0	0	0	0	0	0	0

SIMON FRASER UNIVERSITY
School of Mechatronic Systems Engineering
MSE 352 Digital Logic and Microcontrollers
Quiz III – Fall 2018

2- In assembly programming one of the ways to create a delay is to elapse time using a downward counter like the example below. every iteration of this function takes 3 cycles. In the following example a 1-second delay is created when count=5333333 (decimal).
what is the microcontroller's frequency rate? **(4 marks)**

```
LDR R1 , =count
LDR R0 , [R1]
…
BL  delay                    ; delay at least (3*R0) cycles

.
.
.
.

delay    SUBS R0, R0, #1
         BNE delay                  ; if count (R0) != 0, skip to 'delay'
         BX  LR                     ; return
```

Each iteration of the "delay" subroutine takes 3 pulses and totally the loop repeats 5,333,333 times.

$$5,333,333 * 3 = 15,999,999 \text{ Pulses}$$

Therefore 15,999,999 pulses has passed.

On the other hand this number of pulses has taken "one second". This means that the frequency of the clock pulse is 15,999,999 HZ ≃ 16 MHZ

SIMON FRASER UNIVERSITY
School of Mechatronic Systems Engineering
MSE 352 Digital Logic and Microcontrollers
Quiz III – Fall 2018

cc: Condition Codes

	Generic		*Unsigned*		*Signed*
CS	Carry Set	HI	Higer Than	GT	Greater Than
CC	Carry Clear	HS	Higer or Same	GE	Greater Than or Equal
EQ	Equal (Zero Set)	LO	Lower Than	LT	Less Than
NE	Not Equal (Zero Clear)	LS	Lower Than or Same	LE	Less Than or Equal
VS	Overflow Set			MI	Minus (Negative)
VC	Overflow Clear			PL	Plus (Positive)

ARM Instructions

Add with Carry	ADC⟨cc⟩⟨S⟩	Rd, Rn, ⟨op1⟩	⟨cc⟩: Rd	← Rn + ⟨op1⟩ + CPSR(C)
Add	ADD⟨cc⟩⟨S⟩	Rd, Rn, ⟨op1⟩	⟨cc⟩: Rd	← Rn + ⟨op1⟩
Bitwise AND	AND⟨cc⟩⟨S⟩	Rd, Rn, ⟨op1⟩	⟨cc⟩: Rd	← Rn & ⟨op1⟩
Branch	B⟨cc⟩	⟨offset⟩	⟨cc⟩: PC	← PC + ⟨offset⟩
Branch and Link	BL⟨cc⟩	⟨offset⟩	⟨cc⟩: LR	← PC + 8
			⟨cc⟩: PC	← PC + ⟨offset⟩
Compare	CMP⟨cc⟩	Rn, ⟨op1⟩	⟨cc⟩: CSPR	← (Rn - ⟨op1⟩)
Exclusive OR	EOR⟨cc⟩⟨S⟩	Rd, Rn, ⟨op1⟩	⟨cc⟩: Rd	← Rn ⊕ ⟨op1⟩
Load Register	LDR⟨cc⟩	Rd, ⟨op2⟩	⟨cc⟩: Rd	← M(⟨op2⟩)
Load Register Byte	LDR⟨cc⟩B	Rd, ⟨op2⟩	⟨cc⟩: Rd(7:0)	← M(⟨op2⟩)
			⟨cc⟩: Rd(31:8)	← 0
Move	MOV⟨cc⟩⟨S⟩	Rd, ⟨op1⟩	⟨cc⟩: Rd	← ⟨op1⟩
Move Negative	MVN⟨cc⟩⟨S⟩	Rd, ⟨op1⟩	⟨cc⟩: Rd	← $\overline{\langle op1 \rangle}$
Bitwise OR	ORR⟨cc⟩⟨S⟩	Rd, Rn, ⟨op1⟩	⟨cc⟩: Rd	← Rn | ⟨op1⟩
Subtract with Carry	SBC⟨cc⟩⟨S⟩	Rd, Rn, ⟨op1⟩	⟨cc⟩: Rd	← Rn - ⟨op1⟩ - $\overline{\text{CPSR(C)}}$
Store Register	STR⟨cc⟩	Rd, ⟨op2⟩	⟨cc⟩: M(⟨op2⟩)	← Rd
Store Register Byte	STR⟨cc⟩⟨S⟩	Rd, ⟨op2⟩	⟨cc⟩: M(⟨op2⟩)	← Rd(7:0)
Subtract	SUB⟨cc⟩⟨S⟩	Rd, Rn, ⟨op1⟩	⟨cc⟩: Rd	← Rn - ⟨op1⟩
Software Interrupt	SWI⟨cc⟩	⟨value⟩		
Swap	SWP⟨cc⟩	Rd, Rm, [Rn]	⟨cc⟩: Rd	← M(Rn)
			⟨cc⟩: M(Rn)	← Rm
Swap Byte	SWP⟨cc⟩B	Rd, Rm, [Rn]	⟨cc⟩: Rd(7:0)	← M(Rn)(7:0)
			⟨cc⟩: M(Rn)(7:0)	← Rm(7:0)