CONCORDIA UNIVERSITY

Electrical & Computer Engineering

COEN 311: Midterm, October 25, 2018 (2:45 – 4:00pm)

Notes: - ONLY calculators allowed (closed book!)
- Hand in question paper with answer booklet, or no marks will be given!
- Do all FOUR problems

Problem 1 (10 marks)

What is the content of the registers involved and the **memory locations** affected after execution of the following two instructions:

```
move.w   (a1), d2

move.w   d2, (a2)
```

a1 | 0000 4000

a2 | 0000 4004

d2 | 5678 CDAB

$4000	12D3 A0BC
$4004	8432 1006

move.w (a1),d2 ; d2 = 5678 12D3 //word at address $00004000 in memory moved to least word of d2

move.w d2,(a2) ; $4004 = 12D3 1006 //least word of d2 moved to memory address $00004004

Problem 2 (30 marks – 5+20+5)

For the following assembly language instruction

move $16(a2), d0

a) What is the corresponding machine language instruction?

0011 000 000 101 010 (destination mode 0, register 0) (source mode 5, register 2)
0000 0000 0001 0110 (displacement)
→ $302A 0016

b) Provide its micro-instructions (fetch, decode, and execute cycles).

```
Fetch: MAR ← PC                (1 cc)
       MDR ← M[MAR]            (10 cc)
       IR ← MDR                (1 cc)
Decode                         (1 cc)
Execute: PC ← PC + 2           (2 cc)
         MAR ← PC              (1 cc)
         MDR ← M[MAR]          (10 cc)
         MAR ← MDR + A2        (2 + 1 cc)
         MDR ← M[MAR]          (10 cc)
         D0 ← MDR              (1 cc)
```

PC ← PC + 2 (2 cc)

c) Assuming register transfers cost 1 clock cycle (cc), decoding costs 1 cc, arithmetic operations cost 2 cc and memory accesses cost 10 cc, compute the total execution time for the above instruction.

1 + 10 + 1 + 1 + 2 + 1 + 10 + 2+1 + 10 + 1 + 2 = 42 clock cycles

Problem 3 (30 marks)

Write the complete assembly program starting from ORG to END for the following pseudo-code:

begin

result := 100
i := 1

while i <= 10 **do**
    result := result – i
    i := i + 1
end while

```
        ORG $1000
        Sub d0,d0
        Move d0,a0
        Move one(a0),d0      ;d0 = 1
        Move d0,d1           ;d1 is i
        Move ten(a0),d2      ;d2 = 10
        Move hundred(a0),d3; d3 is result
loop    cmp d2,d1
        bgt     exit    ;if i becomes larger than 10 → exit
        sub     d1,d3   ; subtract i from result
        add     d0,d1   ; add one to i
        bra     loop
exit    move    d3,result(a0)  ;save final result in memory
        move three(a0),d0
        trap #0
one     dc 1
ten     dc 10
hundred dc 100
three   dc 3
result  ds 1

        end
```

Problem 4 (30 marks)

Suppose that the content of the memory is defined below

000000: 9040 3040 3228 0034 3428 0036 9643 3628
000010: 003C B440 6700 000E C7C1 3028 003C 9440
000020: 9040 60EE 3143 0038 4843 3143 003A 3028
000030: 003E 4E40 0005 0003
00003C: 0001 0003

Complete the assembly language code from the address $000012 onwards.

```
                ORG $0000
000000          SUB D0,D0
000002          MOVE D0,A0
000004          MOVE X(A0),D1
000008          MOVE Y(A0),D2
00000C          SUB D3,D3
00000E          MOVE ONE(A0),D3
000012  LOOP    CMP D0,D2
000014          BEQ DONE        ;target = disp + PC + 2 = $000E + $0014 + 2 = $0024 (DONE)
000018          MULS D1,D3
00001A          MOVE ONE(A0),D0
00001E          SUB D0,D2
000020          SUB D0,D0
000022          BRA.B LOOP  ;target = PC+2+disp = $0022 + 2 + $EE (sign ext.) = $0012 (LOOP)
000024  DONE    MOVE D3,R(A0)
000028          SWAP D3
00002A          MOVE D3,R+2(A0)
00002E          MOVE THREE(A0),D0
000032          TRAP #0
000034    X     DC 5
000036    Y     DC 3
000038    R     DS 2
00003C   ONE    DC 1
00003E  THREE   DC 3
                END
```

Add	1101	dDn_3	001000	sDn_3
	1101	dAn_3	011000	sDn_3
Subtract	1001	dDn_3	001000	sDn_3
Multiply	1100	dDn_3	111000	sDn_3
Divide	1000	dDn_3	111000	sDn_3
Compare	1011	dDn_3	001000	sDn_3
Swap	0100100001000			dDn_3
Move (register to register)	0011	dDn_3	000000	sDn_3
	0011	dAn_3	001000	sDn_3
	0011	dDn_3	000001	sAn_3
	0011	dAn_3	001001	sAn_3
Move (register to memory)	0011	An_3	101001	sAn_3
	Displacement$_{16}$			
	0011	An_3	101000	sDn_3
	Displacement$_{16}$			
Move (memory to register)	0011	dAn_3	001101	sAn_3
	Displacement$_{16}$			
	0011	dDn_3	000101	An_3
	Displacement$_{16}$			
Move (memory to memory)	0011	An_3	101101	An_3
	s-Displacement$_{16}$			
	d-Displacement$_{16}$			
Branch (unconditional)	01100000	Displacement$_8$		
Branch on equal	01100111	Displacement$_8$		
Branch on greater	01101110	Displacement$_8$		
Branch on less	01101101	Displacement$_8$		
Stop, dump	010011100100			0000