

# Final Exam S3

## Computer Architecture

Duration: 1 hr 30 min

Write answers only on the answer sheet.

**Exercise 1 (4 points)**

Complete the table shown on the [answer sheet](#). Write down the new values of the registers (except the PC) and memory that are modified by the instructions. **Use the hexadecimal representation. Memory and registers are reset to their initial values for each instruction.**

Initial values:      D0 = \$FFFF0020    A0 = \$00005000    PC = \$00006000  
                           D1 = \$12340004    A1 = \$00005008  
                           D2 = \$FFFFFFF7    A2 = \$00005010

\$005000	54 AF 18 B9 E7 21 48 C0
\$005008	C9 10 11 C8 D4 36 1F 88
\$005010	13 79 01 80 42 1A 2D 49

**Exercise 2 (3 points)**

Complete the table shown on the [answer sheet](#). Give the result of the additions and the values of the N, Z, V and C flags.

**Exercise 3 (4 points)**

Let us consider the following program. Complete the table shown on the [answer sheet](#).

```

Main      move.l  #$A9500,d7
next1     moveq.l #1,d1
          tst.b   d7
          beq   next2
          moveq.l #2,d1
next2     moveq.l #1,d2
          cmpi.w #$95,d7
          bgt   next3
          moveq.l #2,d2
next3     clr.l   d3
          move.l #$512,d0
loop3     addq.l  #1,d3
          subq.b  #1,d0
          bne   loop3
next4     clr.l   d4
          move.l  #64,d0
loop4     addq.l  #1,d4
          dbra   d0,loop4      ; DBRA = DBF
quit     illegal
  
```

**Exercise 4 (9 points)**

All the questions in this exercise are independent. **Except for the output registers, none of the data or address registers must be modified when the subroutine returns.** For the whole exercise, a pair of ASCII characters is made up of two characters representing an integer between 0 and 99 (in base 10). For instance, the characters “08” (‘0’ and ‘8’) represent the value 8; and the characters “42” (‘4’ and ‘2’) represent the value 42. As a reminder, the ASCII code of the ‘0’ character is equal to \$30.

**Be careful. All the subroutines must contain 15 lines of instructions at the most.**

1. Write the **ToAscii** subroutine that converts an integer into a pair of ASCII characters.

Input: **D0.L** holds an integer between 0 and 99 (in base 10).

**A0.L** points to a 2-byte buffer where the pair of ASCII characters must be written.

Example: If **D0** = \$0000002A (42 in base 10) and **A0.L** = \$00005000, then the address \$5000 will hold the value \$34 (ASCII code of ‘4’) and the address \$5001 will hold the value \$32 (ASCII code of ‘2’).

2. Write the **ToInt** subroutine that converts a pair of ASCII characters into an integer.

Input: **A0.L** points to a pair of ASCII characters to convert.

Output: **D0.B** holds the integer that is represented by the pair of ASCII characters.

Be careful, only **D0.B** must be modified (bits from 8 to 31 must not be modified).

Example:

```

Main      movea.l #ascii,a0
          jsr      ToInt          ; D0.B = $2A (42 in base 10)
          illegal
ascii     dc.b      "42"

```

3. By using the **ToInt** subroutine, write the **GetSum** subroutine that returns the sum of two pairs of ASCII characters.

Input: **A0.L** points to a first pair of ASCII characters.

**A1.L** points to a second pair of ASCII characters.

Output: **D0.B** holds an integer that is the sum of the two pairs of ASCII characters.

Be careful, only **D0.B** must be modified (bits from 8 to 31 must not be modified).

Example:

```

Main      movea.l #ascii1,a0
          movea.l #ascii2,a1
          jsr      GetSum        ; D0.B = $32 (50 in base 10)
          illegal
ascii1    dc.b      "42"
ascii2    dc.b      "08"

```





Last name: ..... First name: ..... Group: .....

**ANSWER SHEET TO BE HANDED IN**

**Exercise 1**

Instruction	Memory	Register
Example	\$005000 54 AF <span style="border: 1px solid black; padding: 2px;">00 40</span> E7 21 48 C0	A0 = \$00005004 A1 = \$0000500C
Example	\$005008 C9 10 11 C8 D4 36 <span style="border: 1px solid black; padding: 2px;">FF</span> 88	No change
MOVE.W #5000, -4(A2)		
MOVE.L \$500A, -(A2)		
MOVE.B 1(A0), -7(A1,D1.W)		
MOVE.L -12(A2), \$13(A1,D2.W)		

**Exercise 2**

Operation	Size (bits)	Result (hexadecimal)	N	Z	V	C
\$42 + \$2A	8					
\$7FF0 + \$11	16					
\$FFFFFFFF0 + \$11	32					

**Exercise 3**

Values of registers after the execution of the program. Use the 32-bit hexadecimal representation.	
D1 = \$	D3 = \$
D2 = \$	D4 = \$

**Exercise 4**

ToAscii

ToInt

GetSum