CT1 Exercises Data Transfer

1. What is a load/store architecture?

2. What is the difference between a MOV and a MOVS instruction?

3. Which data transfer instructions should you use if at least one high register is an operand?

4. List different ways of initializing a low register with an immediate value. What are the advantages/disadvantages?

5. What is a pseudo-Instruction? Explain what is done with a <LDR Rn, = literal> pseudo-instruction.

6. Consider the following assembled listing.
   a) For all the LDR instructions, calculate the value of imm and write the instructions in the form LDR <Rt>,[PC,#<imm>]
   b) Replace yyyy with the appropriate machine code
   c) Do you understand what the <B loop> instruction does? If not, look this up in the assembly document on OLAT.
   d) What do you think will happen during the program execution if there was no <B loop> instruction after ADD R0,R1?

```
                CONST_VALUE_X   EQU         123456
00000000
00000000  yyyy  loop    LDR         R0, =0xFF55AAB0
00000002  yyyy          LDR         R1, =CONST_VALUE_X
00000004  yyyy          LDR         R2, myval2
00000006  yyyy          LDR         R3, myval3
00000008  yyyy          LDR         R4, myval4
0000000A  yyyy          LDR         R5, myval5
0000000C  yyyy          LDR         R6, myval6
0000000E  yyyy          LDR         R7, myval7
00000010  4408          ADD          R0, R1
00000012  E7F5          B           loop
00000014
00000014  33445566
                myval3  DCD         0x33445566
00000018  0000FFAA
                myval4  DCD         0x00FFAA
0000001C  00110044
                myval5  DCD         0x110044
00000020  00AABBCC
                myval7  DCD         0xAABBCC
00000024  7788ABCD
                myval6  DCD         0x7788ABCD
00000028  33445555
                myval2  DCD         0x33445555
0000002C
0000002C
0000002C
0000002C
                FF55AAB0
                0001E240
```

7. Consider the following listing.
   a) Compute on the basis of the opcode (marked in blue) the position where data is read to load registers
   b) The opcodes for <LDR R0, lit1> and <LDR R0, =lit1> are the same. But where are the operands? Explain the differences between the 2.
   c) Suppose that the first operation (0x00000000 in the listing) is effectively at address 0x08000008 after linking and loading. Compute the contents of the registers after each instruction is executed once.

```
00000000  4804    again   LDR     R0, lit1
00000002  480A            LDR     R0, =lit1
00000004  4A04            LDR     R2, lit2
00000006  4A0A            LDR     R2, =lit2
00000008  4B04            LDR     R3, lit3
0000000A  4C04            LDR     R4, lit3
0000000C  4E09            LDR     R6, =lit4
0000000E  4F09            LDR     R7, =lit4
00000010  4408            ADD     R0, R1
00000012  E7F5            B       again
00000014
00000014  00000001
          xxx   DCD       0x01
00000018  00000002
          xxx   DCD       0x02
0000001C  00000003
          xxx   DCD       0x03
00000020  00000004
          xxx   DCD       0x04
00000024  00000005
          xxx   DCD       0x05
00000028  00000006
          xxx   DCD       0x06
0000002C
0000002C
0000002C
0000002C
          00000000
          00000000
          00000000
          00000000
```

8. Whenever possible, work out the contents of registers or memory positions that have changed.

```
again   LDR     R0,=0xFF
        LDR     R1, lit1
        LDR     R2, lit2
        LDR     R3,=0x55AA
        MOV     R4, R1
        MOV     R4, R2
        MOV     R6, R3
        MOVS    R7, #04
        LDR     R0,=lit1
        LDR     R1,=lit2
        LDR     R2,=lit3
        LDRB    R5,[R2]
        LDRH    R6,[R2,#2]
        LDR     R2,[R0]
        LDR     R3,[R0,#4]
        STR     R3,[R1]
        STR     R2,[R1,#8]
        STR     R4,[R1,R7]
        LDR     R5,=lit5
        MOVS    R0,#0
        ADDS    R7,R0,#1
        LDRSB   R6,[R5,R0]
        LDRSH   R6,[R5,R7]
        B   again

lit1    DCD  0xEFAA
lit2    DCD  0x12345
lit3    DCD  0x8F1097
lit4    DCD  0xFF76552F
lit5    DCD  0xAA654389
lit6    DCD  0x0165
Var1    DCD  0x23
Var2    DCD  0x24
Var3    DCD  0x23455678
Var4    DCD  0xE4568900
```

9. Write down the assembly instructions to perform the following actions.
   a) Copy contents of R1 in R3 (flags unchanged)

   b) Initialize R0 with 0xAA (flags unchanged)

   c) Initialize R1 with 234 (flags modified)

   d) Initialize R4 with 0x55AACC

   e) Copy contents of R9 in R3.

   f) Initialize R10 with 0x345678

   g) Copy contents of R8 in R9

10. Code the following C programs in assembly

a)

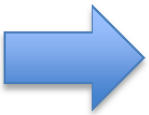
Code this in assembly

```
// C-Code
int x;
int y;
int *xp;

void main(void) {

    x = 3;
    xp = &x;
    y = *xp;

}
```



b)

Code this in assembly

```
// C-Code
char demoArray[2];
char *xp;

void main(void) {
    demoArray[0] = 10;
    demoArray[1] = 11;
    xp = demoArray;
    *xp = 111;
    xp++;
    *xp = 112;
}
```