

Week-3 MPCA Lab

Program 1A : move block of data from one memory to another memory location

Code :

```
.DATA
    A : .WORD 10,20,30,40,50,60,70,80
    B : .WORD 0,0,0,0,0,0,0,0

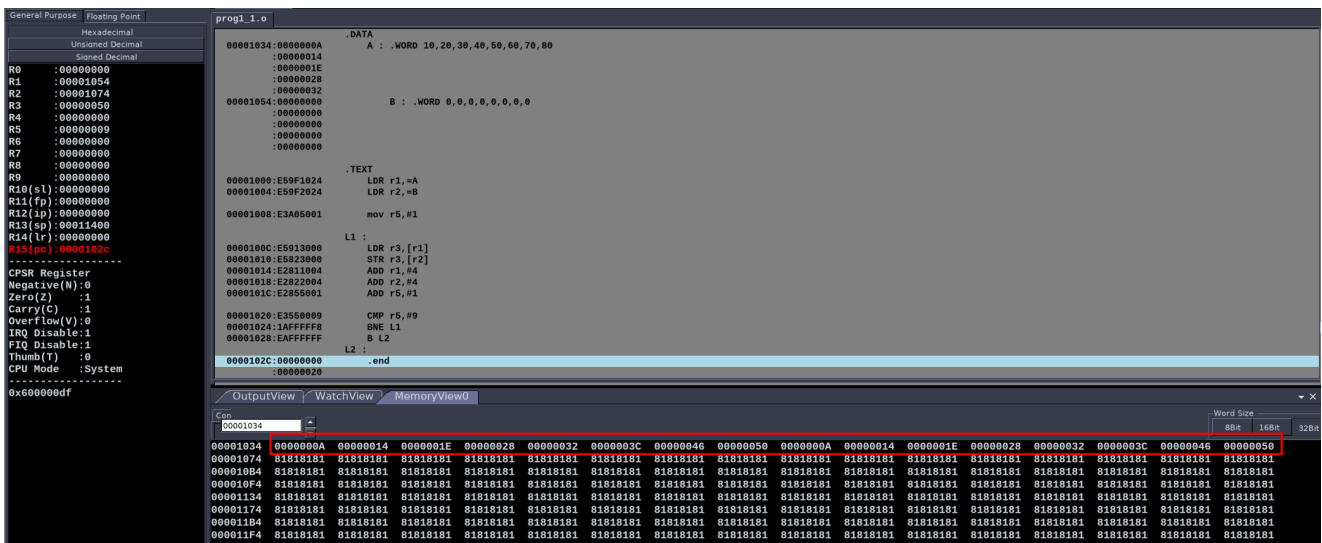
.TEXT
    LDR r1,=A
    LDR r2,=B

    mov r5,#1

L1 :
    LDR r3,[r1]
    STR r3,[r2]
    ADD r1,#4
    ADD r2,#4
    ADD r5,#1

    CMP r5,#9
    BNE L1
    B L2
L2 :
    .end
```

Screenshots :



Program 2 : Write a program to find sun of N data items in the memory

- Store the result in memory
- Use full word
- Use half word
- Use byte words

Note: I have implemented all three in a single program.

Code :

```
.DATA
    A : .WORD 10,20,30,40,50
    B : .HWORD 10,20,30,40,50
    C : .byte 10,20,30,40,50

.TEXT
    LDR r1,=A
    LDR r2,=B
    LDR r3,=C
    LDR r9,=C

    ADD r9,r9,#6
```

```
mov r4,#0
mov r5,#0
mov r6,#0
mov r7,#0
```

L1 :

```
LDR r8,[r1]
ADD r5,r5,r8
ADD r1,#4
```

```
ADD r4,#1
CMP r4,#5
BNE L1
STR r5,[r9]
ADD r9,#4
mov r4,#0
```

L2 :

```
LDRH r8,[r2]
ADD r6,r6,r8
ADD r2,#2
```

```
ADD r4,#1
CMP r4,#5
BNE L2
STR r6,[r9]
ADD r9,#4
mov r4,#0
```

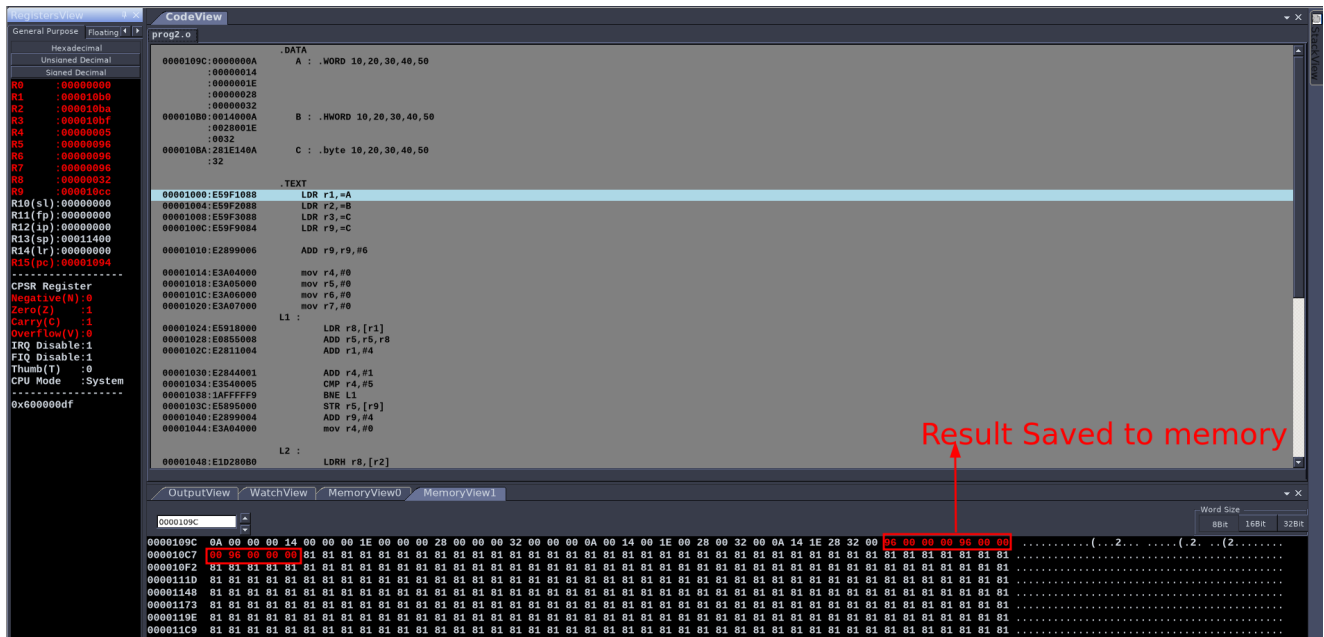
L3 :

```
LDRB r8,[r3]
ADD r7,r7,r8
ADD r3,#1
```

```
ADD r4,#1
CMP r4,#5
BNE L3
STR r7,[r9]
ADD r9,#4
B L4
```

L4 : .end

Screenshots :



Program 3 : Write a program to find sum of N numbers

- Store the result in any memory location.

Code :

```
.DATA
    A : .WORD 16

.TEXT
    LDR r5,=A
    LDR r0,[r5]
    mov r1,#1
    mov r2,#0

L1 :
    ADD r2,r2,r1
    ADD r1,#1
```

```

CMP r1,r0
BNE L1
ADD r5,#4
STR r2,[r5]
B L2
L2 : .end

```

Screenshots :

The screenshot shows the ARM7TDMI emulator interface. The **CodeView** window displays the assembly code for `prog3.o`. The **RegistersView** window shows the state of the registers, with `R5` containing the value `00001034`. The **MemoryView** window shows the memory contents, with the value `00000078` highlighted in red, indicating the result of the program. A red arrow points to this value with the text "Result written to memory".

RegistersView:

Register	Value
R0	00000000
R1	00000000
R2	00000078
R3	00000000
R4	00000000
R5	00001034
R6	00000000
R7	00000000
R8	00000000
R9	00000000
R10 (sl)	00000000
R11 (fp)	00000000
R12 (ip)	00000000
R13 (sp)	00011400
R14 (lr)	00000000
R15 (pc)	00011400

CodeView:

```

00001030:00000010      A : .WORD 16
00001000:E59F5024      LDR r5,=A
00001004:E5950000      LDR r0,[r5]
00001008:E3A01001      mov r1,#1
0000100C:E3A02000      mov r2,#0
00001010:E0822001      L1 :
                                ADD r2,r2,r1
00001014:E2811001      ADD r1,#1
00001018:E1510000      CMP r1,r0
0000101C:1AFFFFF8      BNE L1
00001020:E2855004      ADD r5,#4
00001024:E5852000      STR r2,[r5]
00001028:EAffFFFF      B L2
0000102C:00000000      L2 : .end

```

MemoryView:

Address	Value
00001030	00000010
00001074	81818181
000010B8	81818181
000010FC	81818181
00001140	81818181
00001184	81818181
000011C8	81818181
0000120C	81818181

Program 5 : Convert the following C code to ARM7TDMI asm

C code :

```
if(A==B) C=A+B;
else if(B == C) D=A-B
else E = A*B
```

where A,B,C are memory locations.

Code :

```
.DATA
    A : .WORD 30
    B : .WORD 40
    C : .WORD 50
    D : .WORD 0
    E : .WORD 0

.TEXT
    LDR r0,=A
    LDR r1,=B
    LDR r2,=C
    LDR r11,=C
    LDR r3,=D
    LDR r4,=E

    LDR r5,[r0]
    LDR r6,[r1]
    LDR r7,[r2]

    CMP r5,r6
    BEQ L2
    CMP r6,r7
    BEQ L3
    B L4

L2 :
    ADD r10,r5,r6
    STR r10,[r11]
    B L5
```

```

L3 :
    SUB r10,r5,r6
    STR r10,[r3]
    B L5

L4 :
    MUL r10,r5,r6
    STR r10,[r4]
    B L5

L5 : .end

```

Screenshots :

The screenshot displays a debugger interface with the following components:

- RegistersView:** Shows the state of registers R0 through R15. R10 (SL) contains 000004B0, which is the result of the multiplication in L4.
- CodeView:** Shows the assembly code for prog5.o. The code includes labels L2, L3, and L4. L3 contains instructions to subtract r5 from r6 and store the result in memory at [r3], then branch to L5. L4 contains instructions to multiply r5 and r6, store the result in memory at [r4], and branch to L5.
- MemoryView:** Shows a memory dump starting at address 00001070. The value 000004B0 is highlighted in red at address 00001070, indicating the result of the multiplication in L4.

result written to memory of E