

Tutorial 08: ARM Block Move

Computer Science Department

CS2208b: Introduction to Computer Organization and Architecture

Winter 2019

Instructor: Mahmoud R. El-Sakka

Office: MC-419

Email: elsakka@csd.uwo.ca

Phone: 519-661-2111 x86996

ARM Block Move

```

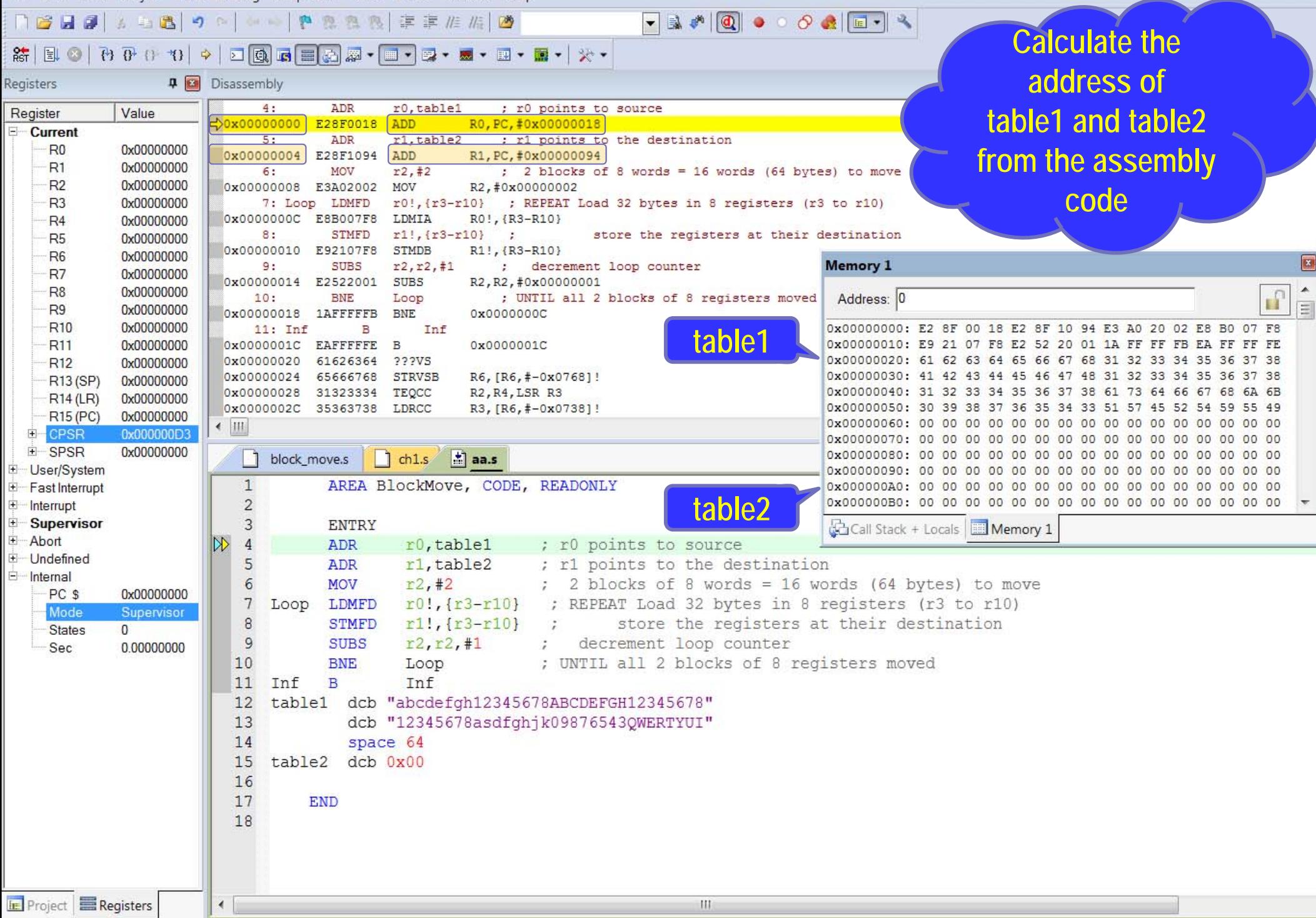
ADR      r0,table1      ;r0 points to source
                  ;(note pseudo-op ADR)
ADR      r1,table2      ;r1 points to the destination
MOV      r2,#2          ;2 blocks of 8 = 16 words (64 bytes)to move
Loop    LDMFD  r0!,{r3-r10} ;REPEAT Load 8 registers (r3 to r10)
        STMFD  r1!,{r3-r10} ;store the registers at their destination
        SUBS   r2,r2,#1      ;decrement loop counter
        BNE    Loop          ;UNTIL all 2 blocks of 8 registers moved
Inf     B                Inf

```

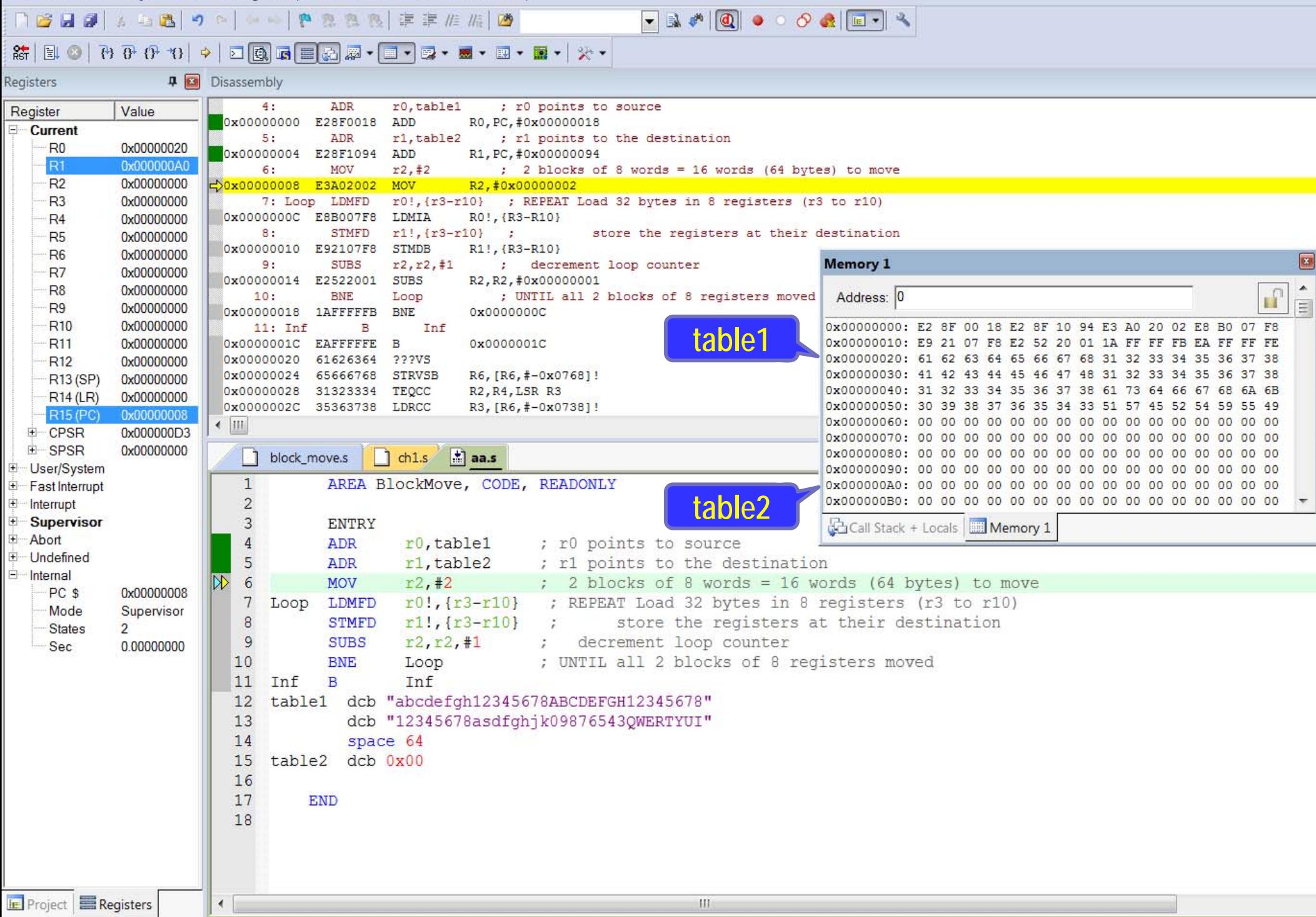
```

table1   DCB  "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNO"
          DCB  "12345678asdfghjk12345678QRSTUVWXYZUI"
          SPACE 64
table2   DCB  0x00

```



Calculate the address of table1 and table2 from the assembly code



C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\aa.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

RST | Disassembly | Registers | Stack | Call Stack + Locals | Memory 1 | Project | Registers

Registers

Register	Value
Current	
R0	0x00000020
R1	0x000000A0
R2	0x00000002
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000000C
CPSR	0x000000D3
SPSR	0x00000000

User/System

- Fast Interrupt
- Interrupt
- Supervisor**
- Abort
- Undefined

Internal

- PC \$ 0x0000000C
- Mode Supervisor
- States 3
- Sec 0.00000000

Disassembly

```

4:     ADR    r0,table1      ; r0 points to source
      0x00000000 E28F0018 ADD    R0,PC,#0x00000018
5:     ADR    r1,table2      ; r1 points to the destination
      0x00000004 E28F1094 ADD    R1,PC,#0x00000094
6:     MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
      0x00000008 E3A02002 MOV    R2,#0x00000002
7: Loop   LDMFD r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
      0x0000000C E8B007F8 LDmia  R0!,{R3-R10}
8:           STMFD r1!,{r3-r10} ; store the registers at their destination
      0x00000010 E92107F8 STMDB R1!,{R3-R10}
9:           SUBS  r2,r2,#1      ; decrement loop counter
      0x00000014 E2522001 SUBS  R2,R2,#0x00000001
10:          BNE   Loop        ; UNTIL all 2 blocks of 8 registers moved
      0x00000018 1AFFFFF BNE   0x0000000C
11:          Inf   B           Inf
      0x0000001C EAFFFFFE B    0x0000001C
0x00000020 61626364 ???VS
0x00000024 65666768 STRVSB R6,[R6,-0x0768]!
0x00000028 31323334 TEQCC R2,R4,LSR R3
0x0000002C 35363738 LDRCC R3,[R6,-0x0738]!

```

Memory 1

Address: 0

0x00000000: E2 8F 00 18 E2 8F 10 94 E3 A0 20 02 E8 B0 07 F8
0x00000010: E9 21 07 F8 E2 52 20 01 1A FF FF FB EA FF FF FE
0x00000020: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000030: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000040: 31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B
0x00000050: 30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49
0x00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

table1

block_moves.s **ch1.s** **aa.s**

```

1      AREA BlockMove, CODE, READONLY
2
3      ENTRY
4      ADR    r0,table1      ; r0 points to source
5      ADR    r1,table2      ; r1 points to the destination
6      MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
7 Loop   LDMFD r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
8      STMFD r1!,{r3-r10} ; store the registers at their destination
9      SUBS  r2,r2,#1      ; decrement loop counter
10     BNE   Loop        ; UNTIL all 2 blocks of 8 registers moved
11     Inf   B           Inf
12     table1 dcb "abcdefghijkl12345678ABCDEFGH12345678"
13     dcb "12345678asdfghjk09876543QWERTYUI"
14     space 64
15     table2 dcb 0x00
16
17     END
18

```

table2

Call Stack + Locals **Memory 1**

Simulation t1: 0.00000000 sec

Instruction Encoding/Decoding

ARM Instruction: : **LDMIA r0!, {r3-R10}**

Condition = 1110 (always – unconditional)

P = 0 (IA: use pointer then adjust)

U = 1 (IA: increment)

S = 0 (user mode)

W = 1 (write-back adjusted pointer)

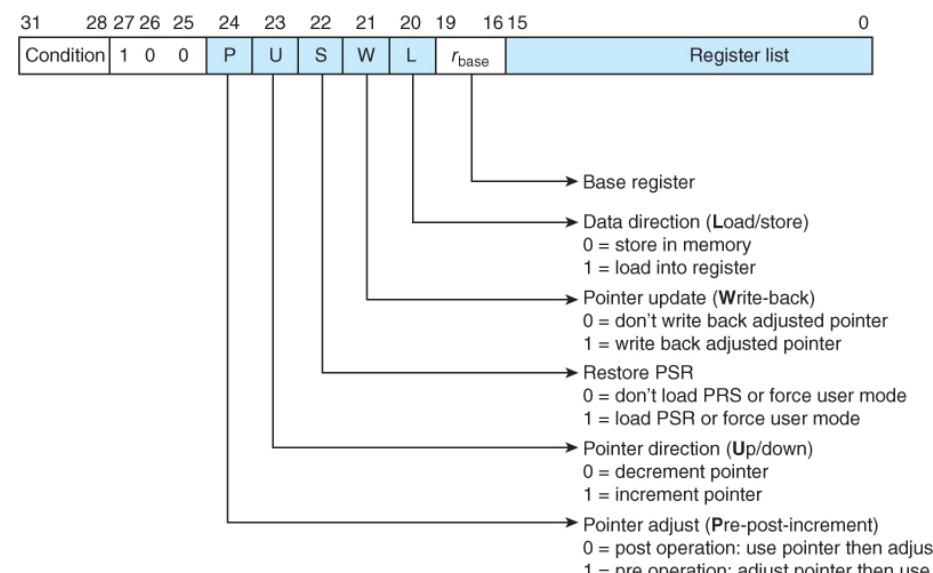
L = 1 (load)

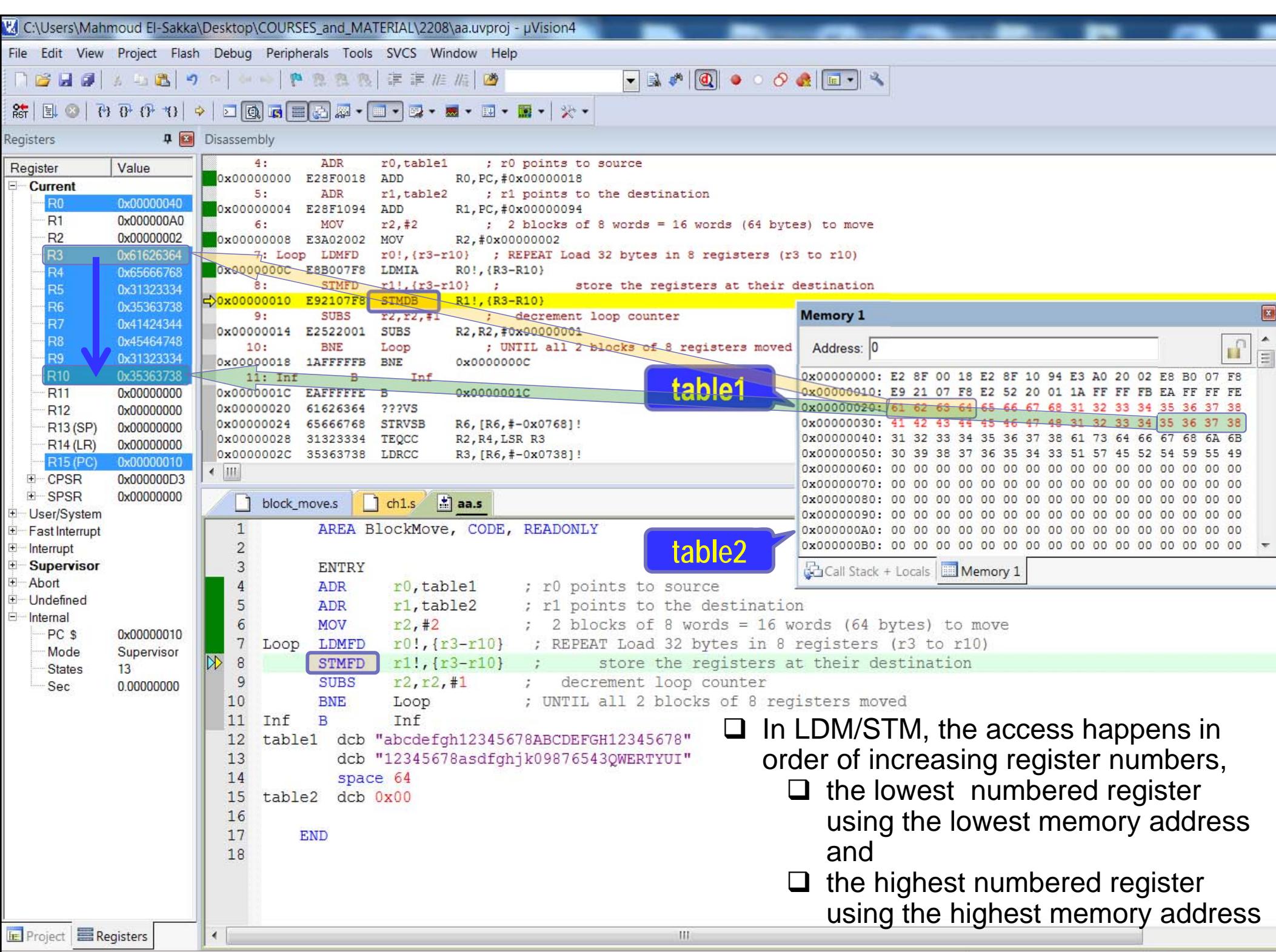
$r_{base} = 0000$ (r0)

Register list (r15, r14,, r2, r1, r0) = 0000 0111 1111 1000
1110 1000 1011 0000 0000 0111 1111 1000

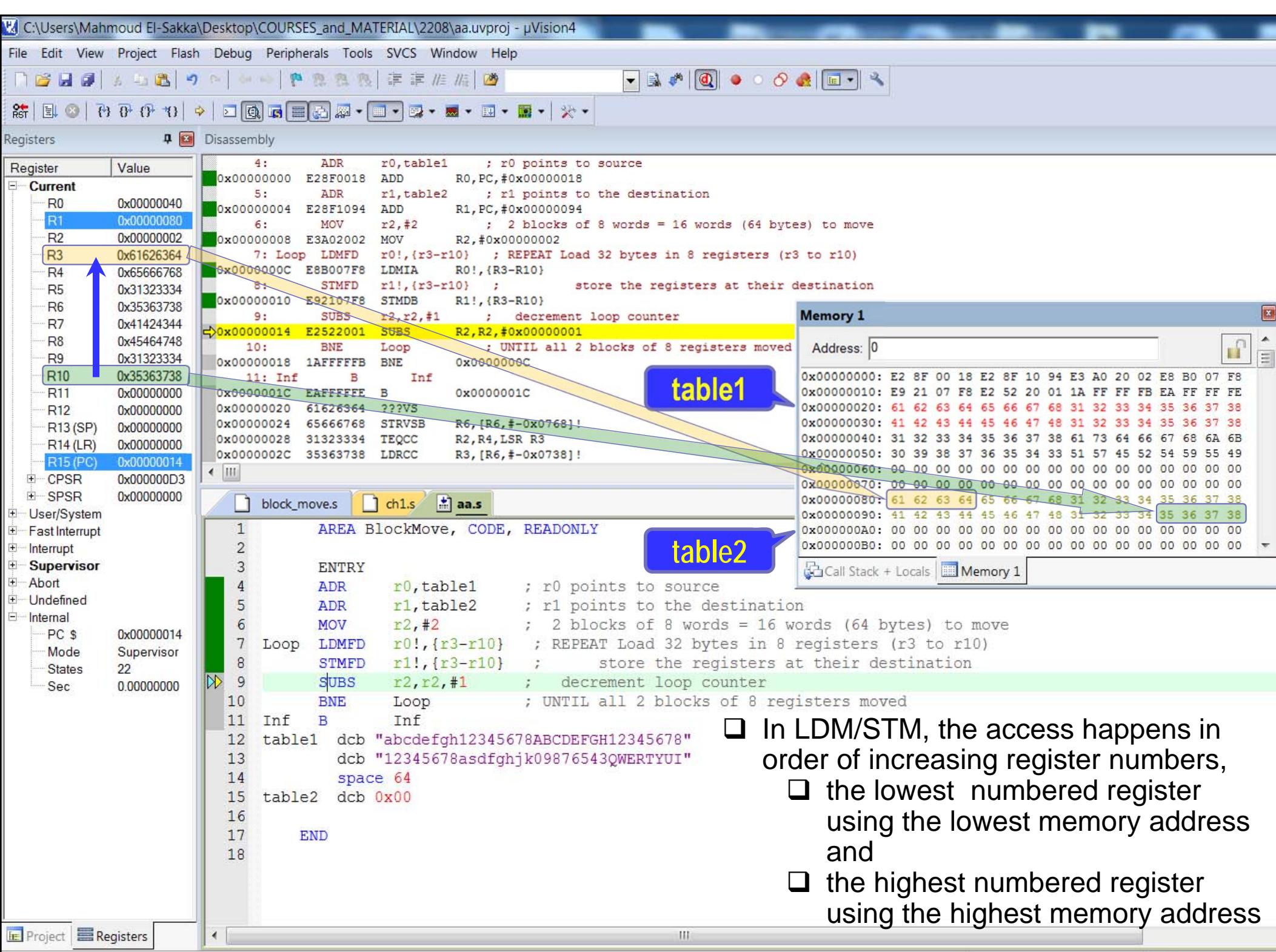
0xE8B007F8

FIGURE 3.58 Encoding ARM's block move instructions





- In LDM/STM, the access happens in order of increasing register numbers,
 - the lowest numbered register using the lowest memory address and
 - the highest numbered register using the highest memory address



- In LDM/STM, the access happens in order of increasing register numbers,
 - the lowest numbered register using the lowest memory address and
 - the highest numbered register using the highest memory address

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\aa.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

RST | Disassembly | Registers | Stack | Call Stack + Locals | Memory 1 | Simulation | t1: 0.00000000 sec

Registers

Register	Value
Current	
R0	0x00000040
R1	0x00000080
R2	0x00000001
R3	0x61626364
R4	0x65666768
R5	0x31323334
R6	0x35363738
R7	0x41424344
R8	0x45464748
R9	0x31323334
R10	0x35363738
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000018
CPSR	0x200000D3
SPSR	0x00000000

User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC \$ 0x00000018
Mode Supervisor
States 23
Sec 0.00000000

Disassembly

```

4:     ADR    r0,table1      ; r0 points to source
      0x00000000 E28F0018 ADD    R0,PC,#0x00000018
5:     ADR    r1,table2      ; r1 points to the destination
      0x00000004 E28F1094 ADD    R1,PC,#0x00000094
6:     MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
      0x00000008 E3A02002 MOV    R2,#0x00000002
7: Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
      0x0000000C E8B007F8 LDmia  R0!,{R3-R10}
8:       STMFD  r1!,{r3-r10} ; store the registers at their destination
      0x00000010 E92107F8 STMDB  R1!,{R3-R10}
9:       SUBS   r2,r2,#1      ; decrement loop counter
      0x00000014 E2522001 SUBS   R2,R2,#0x00000001
10:    BNE   Loop            ; UNTIL all 2 blocks of 8 registers moved
      0x00000018 1AFFFFF BNE   0x0000000C
11:    Inf   B               Inf
      0x0000001C EAFFFFFE B    0x0000001C
0x00000020: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000030: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000040: 31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B
0x00000050: 30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49
0x00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000080: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000090: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

table1

Memory 1

Address: 0

block_moves.s **ch1.s** **aa.s**

table2

```

1 AREA BlockMove, CODE, READONLY
2
3 ENTRY
4     ADR    r0,table1      ; r0 points to source
5     ADR    r1,table2      ; r1 points to the destination
6     MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
7 Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
8       STMFD  r1!,{r3-r10} ; store the registers at their destination
9       SUBS   r2,r2,#1      ; decrement loop counter
10      BNE   Loop            ; UNTIL all 2 blocks of 8 registers moved
11     Inf   B               Inf
12     table1 dcb "abcdefg12345678ABCDEFGH12345678"
13     dcb "12345678asdfghjk09876543QWERTYUI"
14     space 64
15     table2 dcb 0x00
16
17 END
18

```

PC location

PC location + pipelining

Branch backward
5 instructions

Project Registers

Instruction Encoding/Decoding

ARM Instruction: : BNE Loop

Condition = 0001 (NE)

L = 0 (Not BL)

Loop address = 0x0000000C

Current address + pipelining effect

$$= 0x00000018 + 0x00000008$$

$$= 0x00000020$$

Byte-offset

Difference = Loop address

$$- (\text{Current address} + \text{pipelining effect})$$

$$= 0xC - 0x20 = -0x14 = -2_0001\ 0100$$

Difference $\gg 2 = -(2_0001\ 0100 \gg 2) = -2_0000\ 0101$

Word-offset

$$= 2_{1111\ 1011}$$

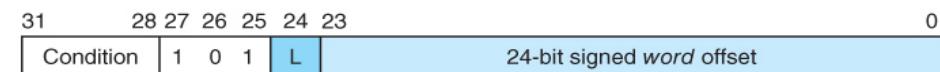
$$= 2_{1111\ 1111\ 1111\ 1111\ 1111\ 1011}$$

0001 1010 1111 1111 1111 1111 1111 1011

TABLE 3.2 ARM's Conditional Execution and Branch Control Mnemonics

Encoding	Mnemonic	Branch on Flag Status	Execute on condition
0000	EQ	Z set	Equal (i.e., zero)
0001	NE	Z clear	Not equal (i.e., not zero)
0010	CS	C set	Unsigned higher or same
0011	CC	C clear	Unsigned lower
0100	MI	N set	Negative
0101	PL	N clear	Positive or zero
0110	VS	V set	Overflow
0111	VC	V clear	No overflow
1000	HI	C set and Z clear	Unsigned higher
1001	LS	C clear or Z set	Unsigned lower or same
1010	GE	N set and V set, or N clear and V clear	Greater or equal
1011	LT	N set and V clear, or N clear and V set	Less than
1100	GT	Z clear, and either N set and V set, or N clear and V clear	Greater than
1101	LE	Z set, or N set and V clear, or N clear and V set	Less than or equal
1110	AL		Always (default)
1111	NV		Never (reserved)

FIGURE 3.41 Encoding ARM's branch and branch-with-link instructions



The L-bit is 0 for a branch instruction and 1 for a branch with link instruction.

The 24-bit word offset is shifted left twice to create a 26-bit byte offset.

0xAFFFFFB

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\aa.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

RST | Disassembly | Registers | Stack | Call Stack + Locals | Memory 1 | Simulation | t1: 0.00000000 sec

Registers

Register	Value
Current	
R0	0x00000040
R1	0x00000080
R2	0x00000001
R3	0x61626364
R4	0x65666768
R5	0x31323334
R6	0x35363738
R7	0x41424344
R8	0x45464748
R9	0x31323334
R10	0x35363738
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000000C
CPSR	0x200000D3
SPSR	0x00000000

User/System

- Fast Interrupt
- Interrupt
- Supervisor**
- Abort
- Undefined

Internal

- PC \$ 0x0000000C
- Mode Supervisor
- States 26
- Sec 0.00000000

Disassembly

```

4:     ADR    r0,table1      ; r0 points to source
      0x00000000 E28F0018 ADD    R0,PC,#0x00000018
5:     ADR    r1,table2      ; r1 points to the destination
      0x00000004 E28F1094 ADD    R1,PC,#0x00000094
6:     MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
      0x00000008 E3A02002 MOV    R2,#0x00000002
7: Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
      0x0000000C E8B007F8 LDMIA  R0!,{R3-R10}
8:     STMFD  r1!,{r3-r10}  ; store the registers at their destination
      0x00000010 E92107F8 STMDB  R1!,{R3-R10}
9:     SUBS   r2,r2,#1       ; decrement loop counter
      0x00000014 E2522001 SUBS   R2,R2,#0x00000001
10:    BNE   Loop            ; UNTIL all 2 blocks of 8 registers moved
      0x00000018 1AFFFFF8 BNE    0x0000000C
11:    Inf   B Inf           ; Inf
      0x0000001C EAFFFFFE B    0x0000001C
0x00000020 61626364 ???VS
0x00000024 65666768 STRVSB  R6,[R6,#-0x0768]!
0x00000028 31323334 TEQCC  R2,R4,LSR R3
0x0000002C 35363738 LDRCC  R3,[R6,#-0x0738]!

```

Memory 1

Address: 0

0x00000000: E2 8F 00 18 E2 8F 10 94 E3 A0 20 02 E8 B0 07 F8
0x00000010: E9 21 07 F8 E2 52 20 01 1A FF FF FB EA FF FF FE
0x00000020: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000030: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000040: 31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B
0x00000050: 30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49
0x00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000080: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000090: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

table1

table2

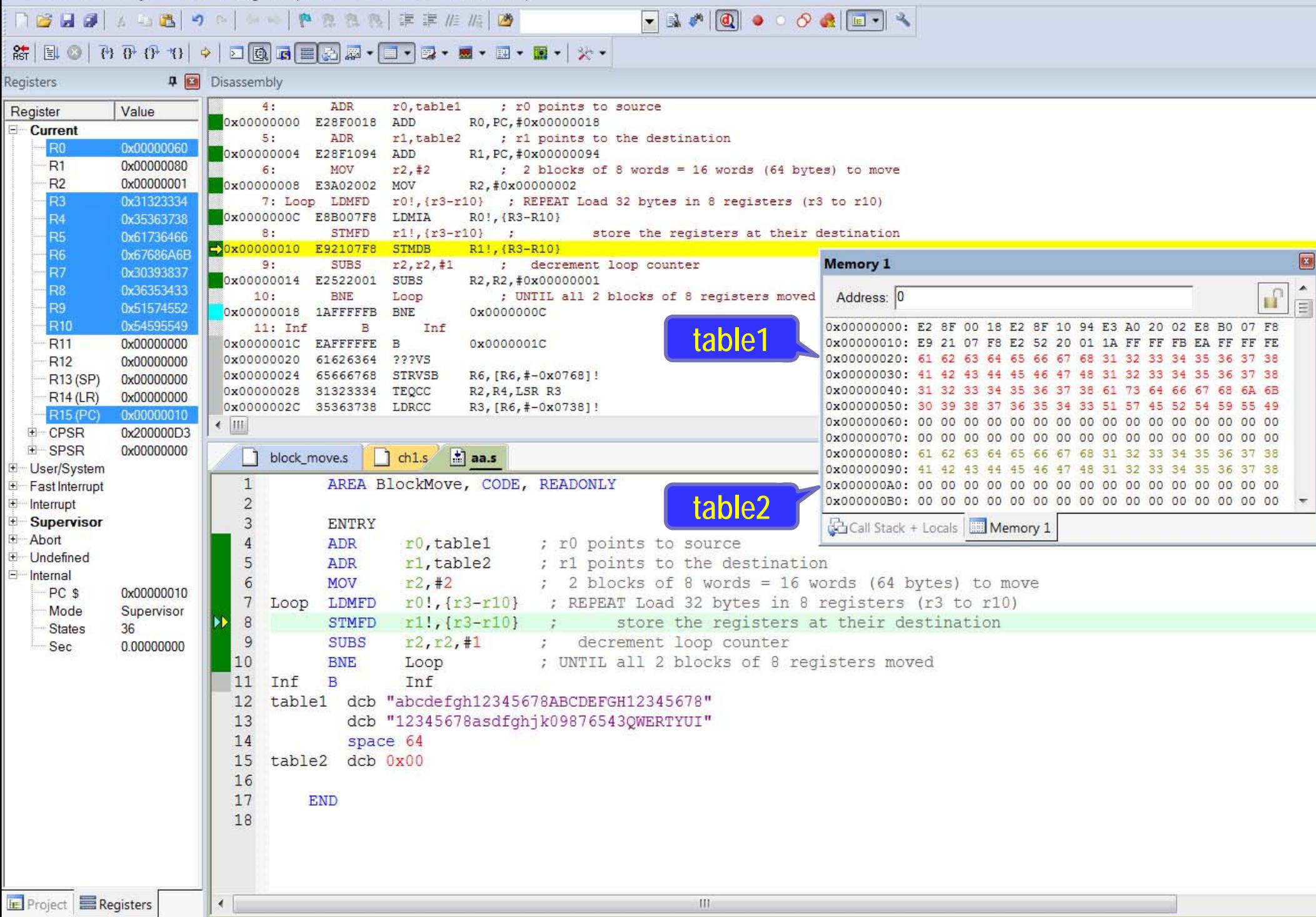
Code Editor

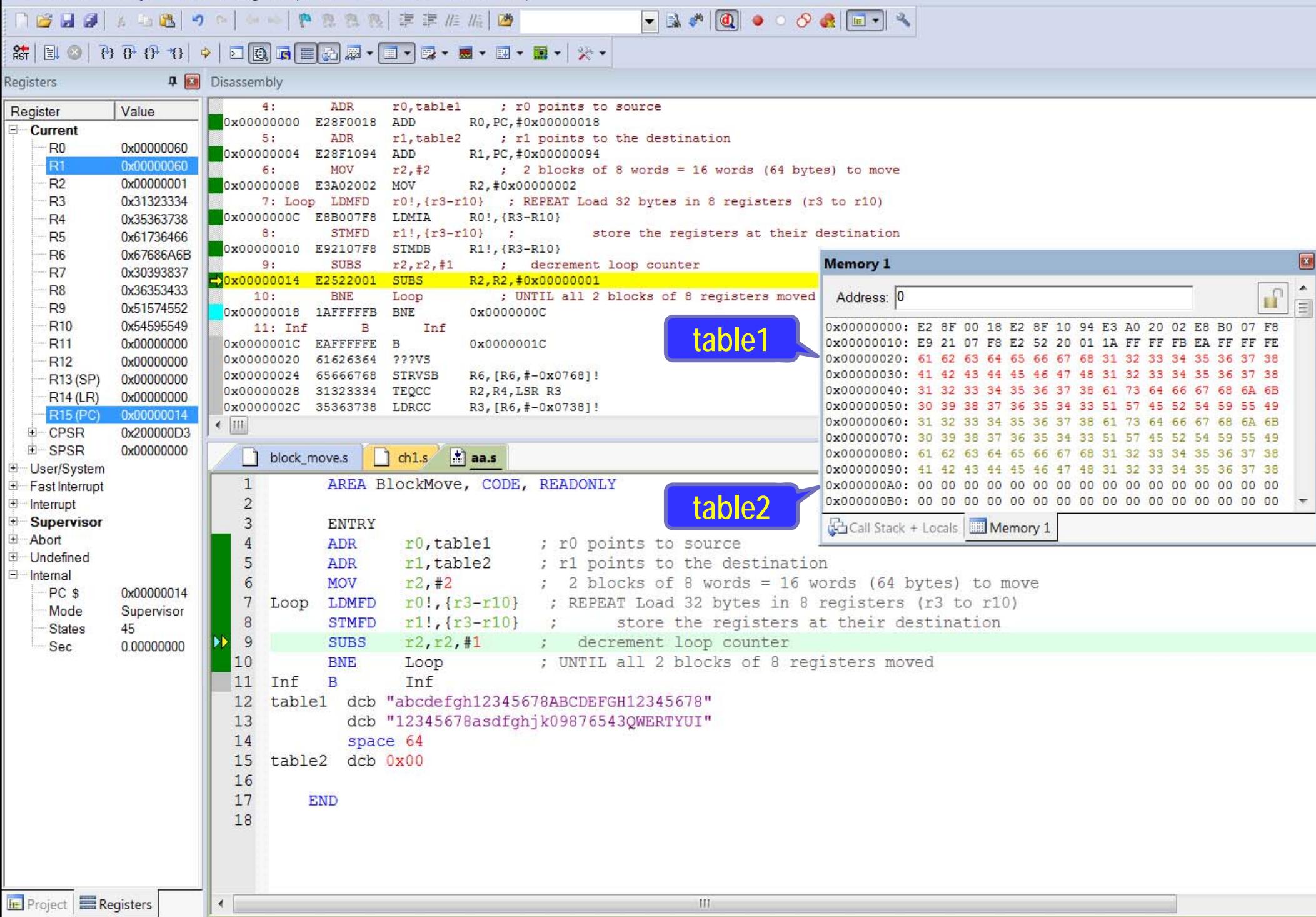
```

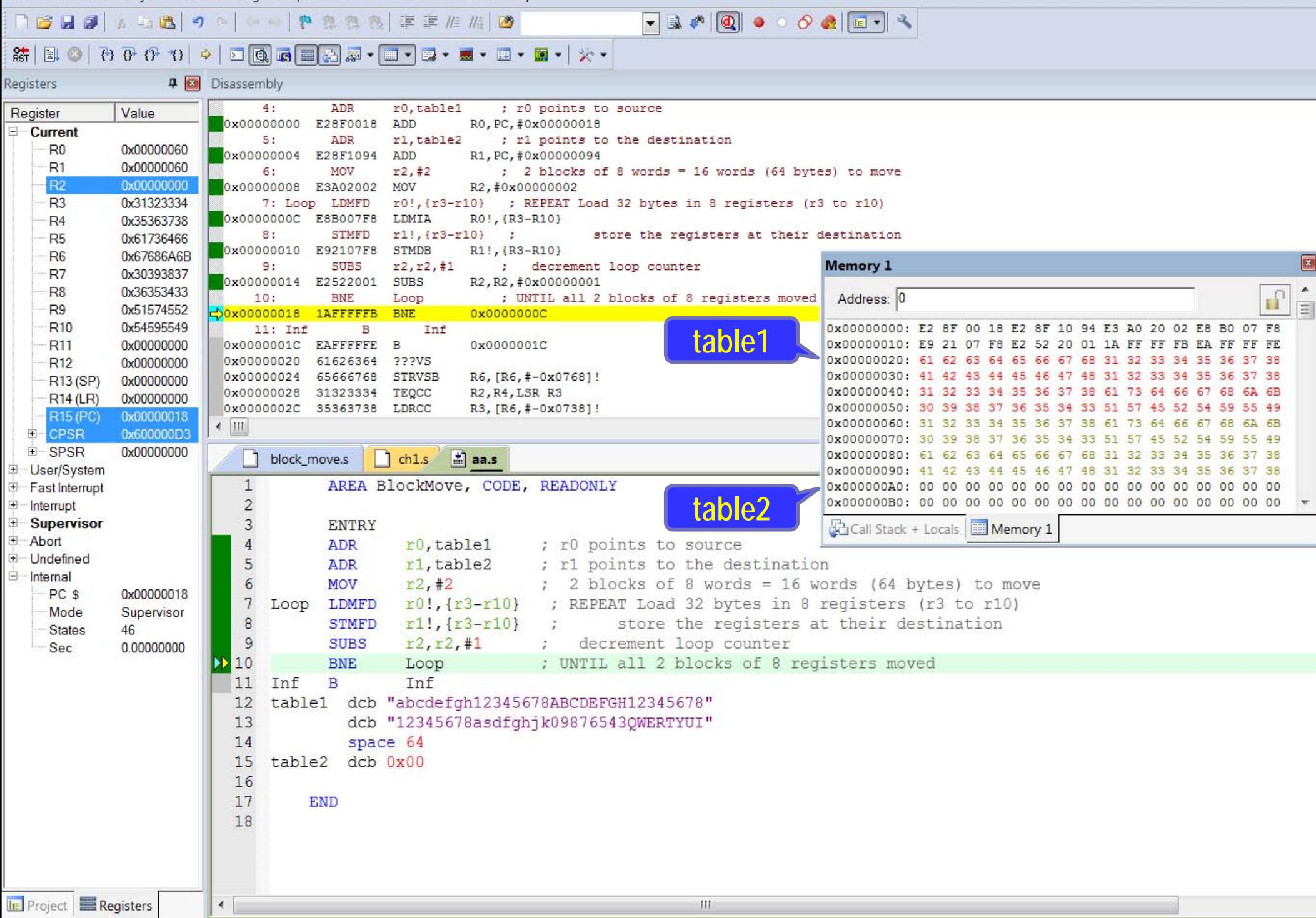
block_moves.s ch1.s aa.s
1 AREA BlockMove, CODE, READONLY
2
3 ENTRY
4 ADR r0,table1      ; r0 points to source
5 ADR r1,table2      ; r1 points to the destination
6 MOV r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
7 Loop LDMFD r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
8 STMFD r1!,{r3-r10} ; store the registers at their destination
9 SUBS r2,r2,#1       ; decrement loop counter
10 BNE Loop           ; UNTIL all 2 blocks of 8 registers moved
11 Inf B Inf           ; Inf
12 table1 dcb "abcdefghijkl2345678ABCDEFGHijkl2345678"
13 dcb "12345678asdfghjk09876543QWERTYUI"
14 space 64
15 table2 dcb 0x00
16
17 END
18

```

Project Registers







C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\aa.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

RST | Disassembly | Registers | Stack | Call Stack + Locals | Memory 1 | Simulation | t1: 0.00000000 sec

Registers

Register	Value
Current	
R0	0x00000060
R1	0x00000060
R2	0x00000000
R3	0x31323334
R4	0x35363738
R5	0x61736466
R6	0x67686A6B
R7	0x30393837
R8	0x36353433
R9	0x51574552
R10	0x54595549
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000001C
CPSR	0x600000D3
SPSR	0x00000000

User/System

- Fast Interrupt
- Interrupt
- Supervisor**
- Abort
- Undefined

Internal

- PC \$ 0x0000001C
- Mode Supervisor
- States 47
- Sec 0.00000000

Disassembly

```

4:     ADR    r0,table1      ; r0 points to source
      0x00000000 E28F0018 ADD    R0,PC,#0x00000018
5:     ADR    r1,table2      ; r1 points to the destination
      0x00000004 E28F1094 ADD    R1,PC,#0x00000094
6:     MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
      0x00000008 E3A02002 MOV    R2,#0x00000002
7: Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
      0x0000000C E8B007F8 LDMIA  R0!,{R3-R10}
8:     STMFD  r1!,{r3-r10}  ; store the registers at their destination
      0x00000010 E92107F8 STMDB  R1!,{R3-R10}
9:     SUBS   r2,r2,#1       ; decrement loop counter
      0x00000014 E2522001 SUBS   R2,R2,#0x00000001
10:    BNE   Loop            ; UNTIL all 2 blocks of 8 registers moved
      0x00000018 1AFFFFF8 BNE    0x0000000C
11:    Inf   B Inf           ; Inf
      0x0000001C EAFFFFFE B    0x0000001C
      0x00000020 61626364 ???VS
      0x00000024 65666768 STRVSB R6,[R6,#-0x0768]!
      0x00000028 31323334 TEQCC R2,R4,LSR R3
      0x0000002C 35363738 LDRCC R3,[R6,#-0x0738]!

```

Memory 1

Address: 0

```

0x00000000: E2 8F 00 18 E2 8F 10 94 A0 20 02 E8 B0 07 F8
0x00000010: F9 21 07 F8 F2 52 20 01 17 FF FF FB FA FF FE FE
0x00000020: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000030: 41 42 43 44 45 46 47 48 32 33 34 35 36 37 38
0x00000040: 31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B
0x00000050: 30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49
0x00000060: 31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B
0x00000070: 30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49
0x00000080: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000090: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

table1

table2

Should use STMEA and fix table2 label



Registers

Register	Value
Current	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000000
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000000
Mode	Supervisor
States	0
Sec	0.00000000

```

4:     ADR    r0,table1 ; r0 points to source
      ↳ 0x00000000 E28F0018 ADD    R0,PC,#0x00000018
5:     ADR    r1,table2 ; r1 points to the destination
      ↳ 0x00000004 E28F1054 ADD    R1,PC,#0x00000054
6:     MOV    r2,#2 ; 2 blocks of 8 words = 16 words (64 bytes) to move
      ↳ 0x00000008 E3A02002 MOV    R2,#0x00000002
7: Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
      ↳ 0x0000000C E6B007F8 LDMIA  R0!,{R3-R10}
8:       STMIA  r1!,{r3-r10} ; store the registers at their destination
      ↳ 0x00000010 E8A107F8 STMIA  R1!,{R3-R10}
9:       SUBS   r2,r2,#1 ; decrement loop counter
      ↳ 0x00000014 E2522001 SUBS   R2,R2,#0x00000001
10:      BNE   Loop    ; UNTIL all 2 blocks of 8 registers moved
      ↳ 0x00000018 1AFFFFF8 BNE    0x0000000C
11:      Inf   B     Inf
      ↳ 0x0000001C EAFFFFFE B     0x0000001C
0x00000020: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000030: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000040: 31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B
0x00000050: 30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49
0x00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

table1

table2

Memory 1

Address: 0

0x00000000:	E2 8F 00 18 E2 8F 10 54 E3 A0 20 02 E8 B0 07 F8
0x00000010:	E8 A1 07 F8 E2 52 20 01 1A FF FF FB EA FF FF FE
0x00000020:	61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000030:	41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000040:	31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B
0x00000050:	30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49
0x00000060:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000070:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000080:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000090:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000A0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000B0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Call Stack + Locals

Memory 1

```

block_moves.s ch1.s aa.s
1      AREA BlockMove, CODE, READONLY
2
3      ENTRY
4      ADR    r0,table1 ; r0 points to source
5      ADR    r1,table2 ; r1 points to the destination
6      MOV    r2,#2 ; 2 blocks of 8 words = 16 words (64 bytes) to move
7 Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
8       STMEA  r1!,{r3-r10} ; store the registers at their destination
9       SUBS   r2,r2,#1 ; decrement loop counter
10      BNE   Loop    ; UNTIL all 2 blocks of 8 registers moved
11      Inf   B     Inf
12      table1 dcb "abcdefghijkl12345678ABCDEFGHI12345678"
13      dcb "12345678asdfghjk09876543QWERTYUI"
14      table2 space 64
15
16      END
17

```

Calculate the address of table2 from the assembly code



Registers

Register	Value
Current	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13(SP)	0x00000000
R14(LR)	0x00000000
R15(PC)	0x00000004
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000004
Mode	Supervisor
States	1
Sec	0.00000000

Disassembly

```

4:     ADR    r0,table1      ; r0 points to source
      0x00000000 E28F0018 ADD    R0,PC,#0x00000018
5:     ADR    r1,table2      ; r1 points to the destination
      0x00000004 E28F1054 ADD    R1,PC,#0x00000054
6:     MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
      0x00000008 E3A02002 MOV    R2,#0x00000002
7: Loop   LDMFD r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
      0x0000000C E6B007F8 LDMIA R0!,{R3-R10}
8:     STMEA r1!,{r3-r10}  ; store the registers at their destination
      0x00000010 E8A107F8 STMIA R1!,{R3-R10}
9:     SUBS   r2,r2,#1      ; decrement loop counter
      0x00000014 E2522001 SUBS   R2,R2,#0x00000001
10:    BNE   Loop           ; UNTIL all 2 blocks of 8 registers moved
      0x00000018 1AFFFFF8 BNE    0x0000000C
11:    Inf   B Inf          ; Inf
      0x0000001C EAFFFFFE B    0x0000001C
      0x00000020 61626364 ???VS
      0x00000024 65666768 STRVS B R6,[R6,#-0x0768]!
      0x00000028 31323334 TEQCC R2,R4,LSR R3
      0x0000002C 35363738 LDRCC R3,[R6,#-0x0738]!

```

table1

table2

block_moves.s ch1.s aa.s

```

1      AREA BlockMove, CODE, READONLY
2
3      ENTRY
4      ADR    r0,table1      ; r0 points to source
5      ADR    r1,table2      ; r1 points to the destination
6      MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
7 Loop   LDMFD r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
8      STMEA r1!,{r3-r10}  ; store the registers at their destination
9      SUBS   r2,r2,#1      ; decrement loop counter
10     BNE   Loop           ; UNTIL all 2 blocks of 8 registers moved
11     Inf   B Inf          ; Inf
12 table1 dcb "abcdefghijklmnopqrstuvwxyz"
13             dcb "12345678asdfghjk09876543QWERTYUI"
14 table2 space 64
15
16     END
17

```

Memory 1

Address: 0

0x00000000:	E2 8F 00 18 E2 8F 10 54 E3 A0 20 02 E8 B0 07 F8
0x00000010:	E8 A1 07 F8 E2 52 20 01 1A FF FF FB EA FF FF FE
0x00000020:	61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000030:	41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000040:	31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B
0x00000050:	30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49
0x00000060:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000070:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000080:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000090:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000A0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000B0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Call Stack + Locals Memory 1

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\aa.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

RST | Disassembly | Registers | Stack | Call Stack + Locals | Memory 1 | Simulation | t1: 0.00000000 sec

Registers

Register	Value
Current	
R0	0x00000020
R1	0x00000060
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000008
CPSR	0x000000D3
SPSR	0x00000000

User/System

- Fast Interrupt
- Interrupt
- Supervisor**
- Abort
- Undefined

Internal

- PC \$ 0x00000008
- Mode Supervisor
- States 2
- Sec 0.00000000

Disassembly

```

4:     ADR    r0,table1      ; r0 points to source
      0x00000000 E28F0018 ADD    R0,PC,#0x00000018
5:     ADR    r1,table2      ; r1 points to the destination
      0x00000004 E28F1054 ADD    R1,PC,#0x00000054
6:     MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
      ↳ 0x00000008 E3A02002 MOV    R2,#0x00000002
7: Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
      0x0000000C E8B007F8 LDMIA  R0!,{R3-R10}
8:     STMEA r1!,{r3-r10}   ; store the registers at their destination
      0x00000010 E8A107F8 STMIA  R1!,{R3-R10}
9:     SUBS  r2,r2,#1       ; decrement loop counter
      0x00000014 E2522001 SUBS   R2,R2,#0x00000001
10:    BNE   Loop            ; UNTIL all 2 blocks of 8 registers moved
      0x00000018 1AFFFFF8 BNE    0x0000000C
11:    Inf   B Inf           ; Inf
      0x0000001C EAFFFFFE B    0x0000001C
      0x00000020 61626364 ???VS
      0x00000024 65666768 STRVS  R6,[R6,#-0x0768]!
      0x00000028 31323334 TEQCC  R2,R4,LSR R3
      0x0000002C 35363738 LDRCC  R3,[R6,#-0x0738]!

```

Memory 1

Address: 0

```

0x00000000: E2 8F 00 18 E2 8F 10 54 E3 A0 20 02 E8 B0 07 F8
0x00000010: E8 A1 07 F8 E2 52 20 01 1A FF FF FB EA FF FF FE
0x00000020: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000030: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000040: 31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B
0x00000050: 30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49
0x00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

table1

table2

aa.s

```

1      AREA BlockMove, CODE, READONLY
2
3      ENTRY
4      ADR    r0,table1      ; r0 points to source
5      ADR    r1,table2      ; r1 points to the destination
6      MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
7 Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
8      STMEA r1!,{r3-r10}   ; store the registers at their destination
9      SUBS  r2,r2,#1       ; decrement loop counter
10     BNE   Loop            ; UNTIL all 2 blocks of 8 registers moved
11     Inf   B Inf           ; Inf
12 table1 dcb "abcdefghijkl12345678ABCDEFGH12345678"
13             dcb "12345678asdfghjk09876543QWERTYUI"
14 table2 space 64
15
16     END
17

```

Call Stack + Locals

Registers



Registers

Register	Value
Current	
R0	0x00000020
R1	0x00000060
R2	0x00000002
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000000C
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x0000000C
Mode	Supervisor
States	3
Sec	0.00000000

Disassembly

```

4:     ADR    r0,table1      ; r0 points to source
      0x00000000 E28F0018 ADD    R0,PC,#0x00000018
5:     ADR    r1,table2      ; r1 points to the destination
      0x00000004 E28F1054 ADD    R1,PC,#0x00000054
6:     MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
      0x00000008 E3A02002 MOV    R2,#0x00000002
7: Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
      0x0000000C E8B007F8 LDMIA  R0!,{R3-R10}
8:     STMEA r1!,{r3-r10}   ; store the registers at their destination
      0x00000010 E8A107F8 STMIA  R1!,{R3-R10}
9:     SUBS  r2,r2,#1       ; decrement loop counter
      0x00000014 E2522001 SUBS   R2,R2,#0x00000001
10:    BNE   Loop           ; UNTIL all 2 blocks of 8 registers moved
      0x00000018 1AFFFFF8 BNE    0x0000000C
11:    Inf   B   Inf
      0x0000001C EAFFFFFE B    0x0000001C
0x00000020 61626364 ???VS
0x00000024 65666768 STRVS B R6,[R6,#-0x0768]!
0x00000028 31323334 TEQCC R2,R4,LSR R3
0x0000002C 35363738 LDRCC R3,[R6,#-0x0738]!

```

table1

table2

block_moves.s ch1.s aa.s

```

1      AREA BlockMove, CODE, READONLY
2
3      ENTRY
4      ADR    r0,table1      ; r0 points to source
5      ADR    r1,table2      ; r1 points to the destination
6      MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
7 Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
8      STMEA r1!,{r3-r10}   ; store the registers at their destination
9      SUBS  r2,r2,#1       ; decrement loop counter
10     BNE   Loop           ; UNTIL all 2 blocks of 8 registers moved
11     Inf   B   Inf
12 table1 dcb "abcdefghijklmnopqrstuvwxyz"
13             dcb "12345678asdfghjk09876543QWERTYUI"
14 table2 space 64
15
16     END
17

```

Memory 1

Address: 0

0x00000000: E2 8F 00 18 E2 8F 10 54 E3 A0 20 02 E8 B0 07 F8
0x00000010: E8 A1 07 F8 E2 52 20 01 1A FF FF FB EA FF FF FE
0x00000020: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000030: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000040: 31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B
0x00000050: 30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49
0x00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Call Stack + Locals

Memory 1

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\aa.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

RST | Disassembly | Registers | Stack | Call Stack + Locals | Memory 1 | Simulation | t1: 0.00000000 sec

Registers

Register	Value
Current	
R0	0x00000040
R1	0x00000060
R2	0x00000002
R3	0x61626364
R4	0x65666768
R5	0x31323334
R6	0x35363738
R7	0x41424344
R8	0x45464748
R9	0x31323334
R10	0x35363738
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000

User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC \$ 0x00000010
Mode Supervisor
States 13
Sec 0.00000000

Disassembly

```

4:     ADR    r0,table1      ; r0 points to source
      0x00000000 E28F0018 ADD    R0,PC,#0x00000018
5:     ADR    r1,table2      ; r1 points to the destination
      0x00000004 E28F1054 ADD    R1,PC,#0x00000054
6:     MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
      0x00000008 E3A02002 MOV    R2,#0x00000002
7: Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
      0x0000000C E8B007F8 LDMIA  R0!,{R3-R10}
8:     STMEA r1!,{r3-r10}   ; store the registers at their destination
      0x00000010 E8A107F8 STMIA  R1!,{R3-R10}
9:     SUBS  r2,r2,#1       ; decrement loop counter
      0x00000014 E2522001 SUBS   R2,R2,#0x00000001
10:    BNE   Loop           ; UNTIL all 2 blocks of 8 registers moved
      0x00000018 1AFFFFF8 BNE    0x0000000C
11:    Inf   B   Inf
      0x0000001C EAFFFFFE B    0x0000001C
0x00000020: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000030: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000040: 31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B
0x00000050: 30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49
0x00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

table1

table2

Memory 1

Address: 0

Call Stack + Locals | Memory 1

Code Editor

```

block_moves.s ch1.s aa.s
1 AREA BlockMove, CODE, READONLY
2
3 ENTRY
4 ADR r0,table1      ; r0 points to source
5 ADR r1,table2      ; r1 points to the destination
6 MOV r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
7 Loop LDMFD r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
8 STMEA r1!,{r3-r10} ; store the registers at their destination
9 SUBS r2,r2,#1       ; decrement loop counter
10 BNE Loop           ; UNTIL all 2 blocks of 8 registers moved
11 Inf B Inf
12 table1 dcb "abcdefghijklmnopqrstuvwxyz"
13 dcb "12345678asdfghjk09876543QWERTYUI"
14 table2 space 64
15
16 END
17

```

Project | Registers | Stack | Call Stack + Locals | Memory 1 | Simulation | t1: 0.00000000 sec

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\aa.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

RST | Disassembly | Registers | Stack | Call Stack + Locals | Memory 1 | Simulation | t1: 0.00000000 sec

Registers

Register	Value
Current	
R0	0x00000040
R1	0x00000080
R2	0x00000002
R3	0x61626364
R4	0x65666768
R5	0x31323334
R6	0x35363738
R7	0x41424344
R8	0x45464748
R9	0x31323334
R10	0x35363738
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000014
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000014
Mode	Supervisor
States	22
Sec	0.00000000

Disassembly

```

4:     ADR    r0,table1      ; r0 points to source
      0x00000000 E28F0018 ADD    R0,PC,#0x00000018
5:     ADR    r1,table2      ; r1 points to the destination
      0x00000004 E28F1054 ADD    R1,PC,#0x00000054
6:     MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
      0x00000008 E3A02002 MOV    R2,#0x00000002
7: Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
      0x0000000C E8B007F8 LDMIA  R0!,{R3-R10}
8:       STMEA r1!,{r3-r10}  ; store the registers at their destination
      0x00000010 E8A107F8 STMIA  R1!,{R3-R10}
9:       SUBS  r2,r2,#1      ; decrement loop counter
      0x00000014 E2522001 SUBS  R2,R2,#0x00000001
10:    BNE   Loop           ; UNTIL all 2 blocks of 8 registers moved
      0x00000018 1AFFFFF8 BNE   0x0000009C
11:    Inf   B              Inf
      0x0000001C EAFFFFFE B    0x0000001C
0x00000020 61626364 ??VS
0x00000024 65666768 STRVSB R6,[R6,#-0x0768]!
0x00000028 31323334 TEQCC R2,R4,LSR R3
0x0000002C 35363738 LDRCC R3,[R6,#-0x0738]!

```

Memory 1

Address: 0

```

0x00000000: E2 8F 00 18 E2 8F 10 54 E3 A0 20 02 E8 B0 07 F8
0x00000010: E8 A1 07 F8 E2 52 20 01 1A FF FF FB EA FF FF FE
0x00000020: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000030: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000040: 31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B
0x00000050: 30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49
0x00000060: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000070: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

table1

table2

aa.s

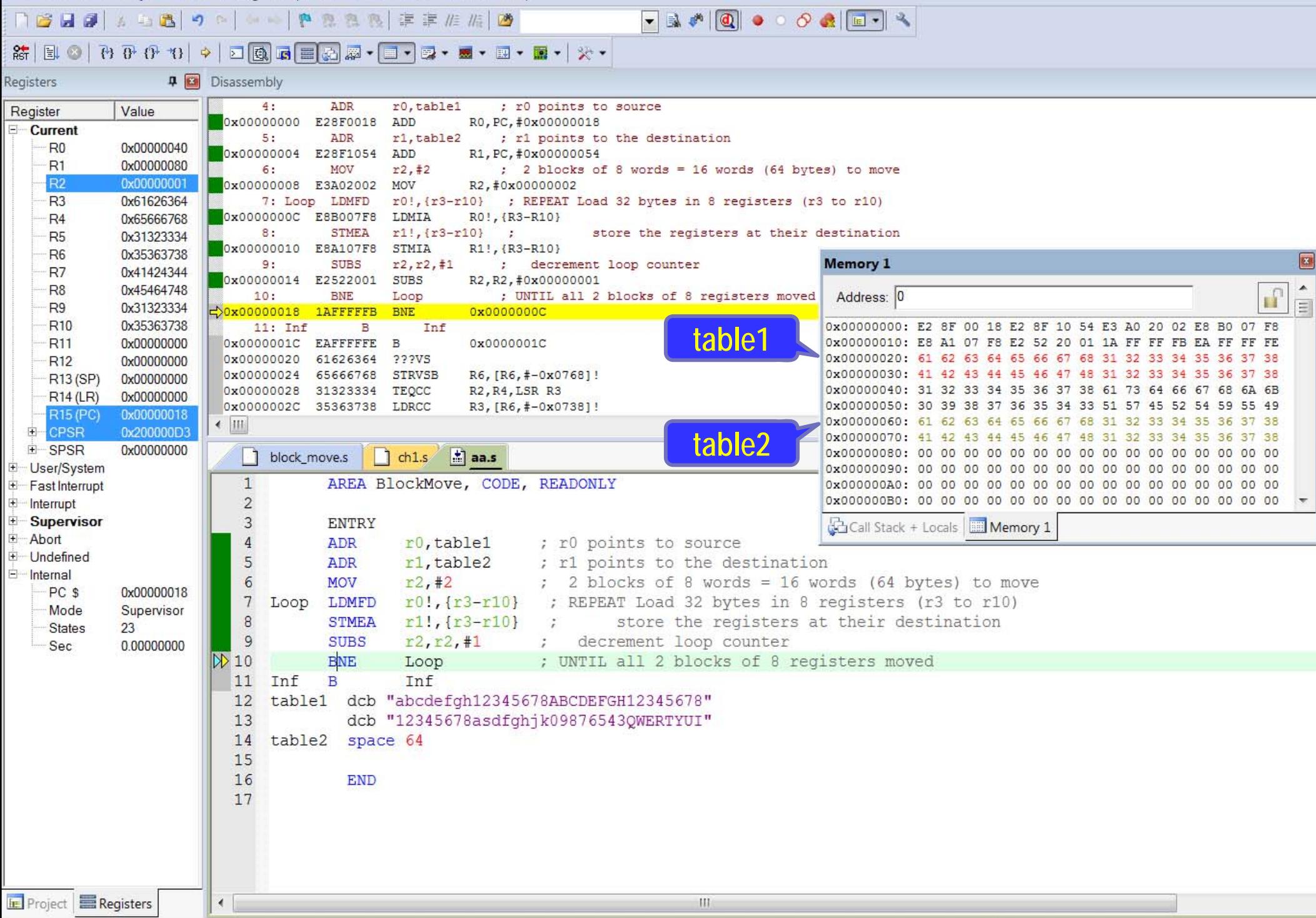
```

AREA BlockMove, CODE, READONLY
ENTRY
    ADR    r0,table1      ; r0 points to source
    ADR    r1,table2      ; r1 points to the destination
    MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
       STMEA r1!,{r3-r10}  ; store the registers at their destination
       SUBS  r2,r2,#1      ; decrement loop counter
    BNE   Loop           ; UNTIL all 2 blocks of 8 registers moved
    Inf   B              Inf
table1 dcb "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOQRSTUVWXYZ"
table2 space 64
END

```

In LDM/STM, the access happens in order of increasing register numbers,

- the lowest numbered register using the lowest memory address and**
- the highest numbered register using the highest memory address**





Registers

Register	Value
Current	
R0	0x00000040
R1	0x00000080
R2	0x00000001
R3	0x61626364
R4	0x65666768
R5	0x31323334
R6	0x35363738
R7	0x41424344
R8	0x45464748
R9	0x31323334
R10	0x35363738
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000000C
CPSR	0x200000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x0000000C
Mode	Supervisor
States	26
Sec	0.00000000

```

4:     ADR    r0,table1      ; r0 points to source
      0x00000000 E28F0018 ADD    R0,PC,#0x00000018
5:     ADR    r1,table2      ; r1 points to the destination
      0x00000004 E28F1054 ADD    R1,PC,#0x00000054
6:     MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
      0x00000008 E3A02002 MOV    R2,#0x00000002
7: Loop   LDMFD r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
      0x0000000C E8B007F8 LDMIA R0!,{R3-R10}
8:     STMEA r1!,{r3-r10}   ; store the registers at their destination
      0x00000010 E8A107F8 STMIA R1!,{R3-R10}
9:     SUBS  r2,r2,#1       ; decrement loop counter
      0x00000014 E2522001 SUBS  R2,R2,#0x00000001
10:    BNE   Loop           ; UNTIL all 2 blocks of 8 registers moved
      0x00000018 1AFFFFF8 BNE   0x0000000C
11:    Inf   B               Inf
      0x0000001C EAFFFFFE B    0x0000001C
0x00000020 61626364 ??VS
0x00000024 65666768 STRVSB R6,[R6,#-0x0768]!
0x00000028 31323334 TEQCC R2,R4,LSR R3
0x0000002C 35363738 LDRCC R3,[R6,#-0x0738]!

```

table1

table2

```

block_moves aa.s
1      AREA BlockMove, CODE, READONLY
2
3      ENTRY
4      ADR    r0,table1      ; r0 points to source
5      ADR    r1,table2      ; r1 points to the destination
6      MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
7 Loop   LDMFD r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
8      STMEA r1!,{r3-r10}   ; store the registers at their destination
9      SUBS  r2,r2,#1       ; decrement loop counter
10     BNE   Loop           ; UNTIL all 2 blocks of 8 registers moved
11     Inf   B               Inf
12     table1 dcb "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNO"
13     dcb "12345678asdfghjk09876543QWERTYUI"
14     table2 space 64
15
16     END
17

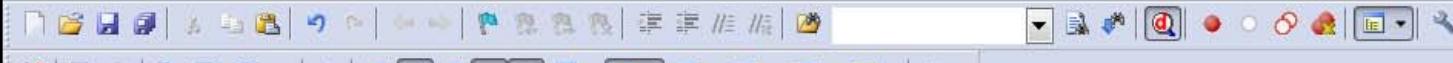
```

Memory 1

Address: 0

0x00000000: E2 8F 00 18 E2 8F 10 54 E3 A0 20 02 E8 B0 07 F8
0x00000010: E8 A1 07 F8 E2 52 20 01 1A FF FF FB EA FF FF FE
0x00000020: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000030: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000040: 31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B
0x00000050: 30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49
0x00000060: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000070: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Call Stack + Locals | Memory 1



Registers

Register	Value
Current	
R0	0x00000060
R1	0x00000080
R2	0x00000001
R3	0x31323334
R4	0x35363738
R5	0x61736466
R6	0x67686A6B
R7	0x30393837
R8	0x36353433
R9	0x51574552
R10	0x54595549
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x200000D3
SPSR	0x00000000

User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000010
Mode	Supervisor
States	36
Sec	0.00000000

Disassembly

```

4:     ADR    r0,table1      ; r0 points to source
      0x00000000 E28F0018 ADD    R0,PC,#0x00000018
5:     ADR    r1,table2      ; r1 points to the destination
      0x00000004 E28F1054 ADD    R1,PC,#0x00000054
6:     MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
      0x00000008 E3A02002 MOV    R2,#0x00000002
7: Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
      0x0000000C E8B007F8 LDMIA  R0!,{R3-R10}
8:     STMEA r1!,{r3-r10}   ; store the registers at their destination
      0x00000010 E8A107F8 STMIA  R1!,{R3-R10}
9:     SUBS  r2,r2,#1       ; decrement loop counter
      0x00000014 E2522001 SUBS   R2,R2,#0x00000001
10:    BNE   Loop           ; UNTIL all 2 blocks of 8 registers moved
      0x00000018 1AFFFFFF BNE    0x0000000C
11:    Inf   B              Inf
      0x0000001C EAFFFFFE B    0x0000001C
0x00000020: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000024: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000028: 31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B
0x00000030: 30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49
0x00000034: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000038: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000044: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000052: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000056: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000064: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000068: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000072: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000076: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000084: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000088: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000092: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

table1

table2

block_moves aa.s

```

1      AREA BlockMove, CODE, READONLY
2
3      ENTRY
4      ADR    r0,table1      ; r0 points to source
5      ADR    r1,table2      ; r1 points to the destination
6      MOV    r2,#2          ; 2 blocks of 8 words = 16 words (64 bytes) to move
7 Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
8      STMEA r1!,{r3-r10}   ; store the registers at their destination
9      SUBS  r2,r2,#1       ; decrement loop counter
10     BNE   Loop           ; UNTIL all 2 blocks of 8 registers moved
11     Inf   B              Inf
12 table1 dcb "abcdefghijkl12345678ABCDEFGHI12345678"
13             dcb "12345678asdfghjk09876543QWERTYUI"
14 table2 space 64
15
16     END
17

```

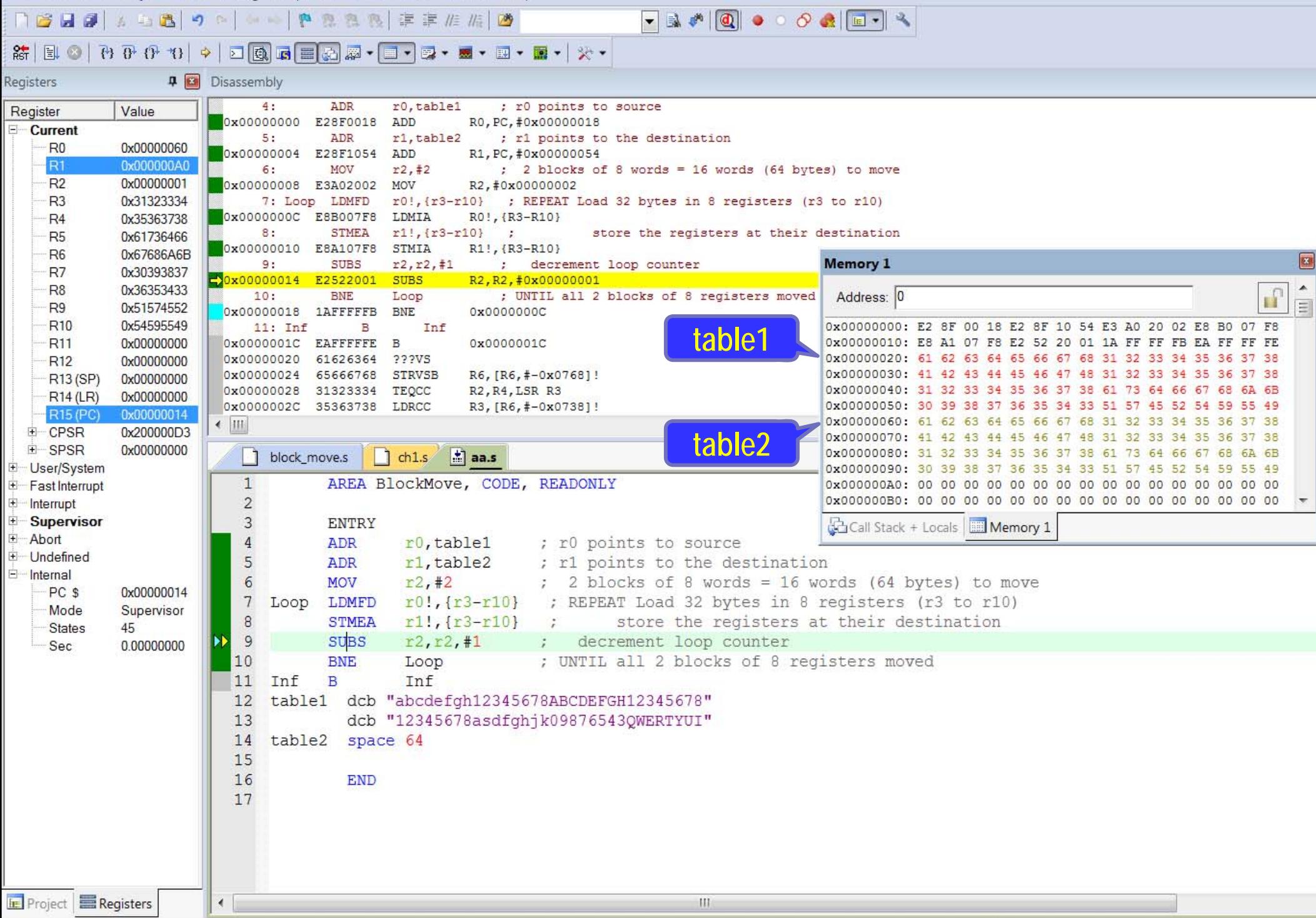
Memory 1

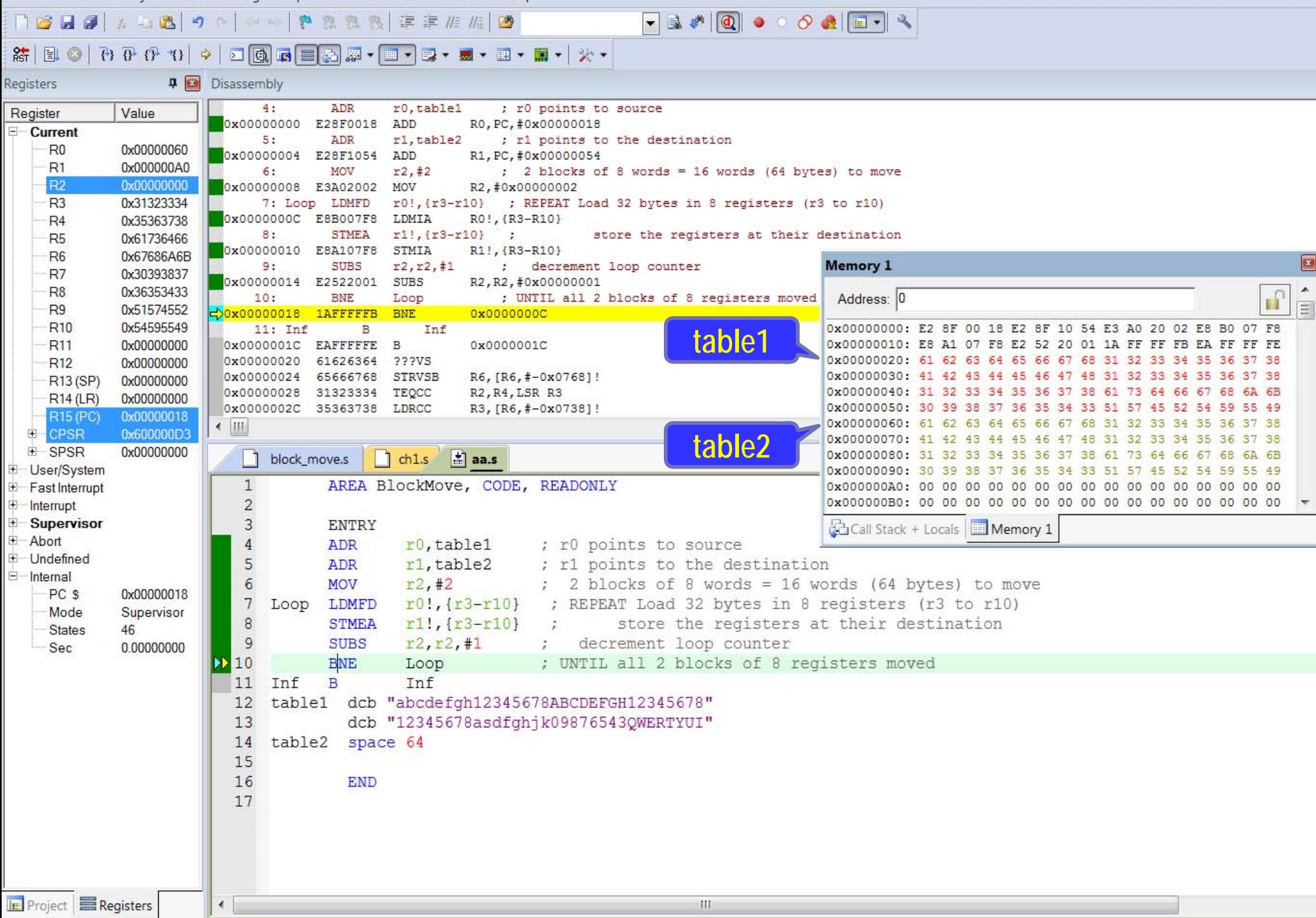
Address: 0

0x00000000: E2 8F 00 18 E2 8F 10 54 E3 A0 20 02 E8 B0 07 F8
0x00000010: E8 A1 07 F8 E2 52 20 01 1A FF FF FB EA FF FF FE
0x00000020: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000030: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000040: 31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B
0x00000050: 30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49
0x00000060: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38
0x00000070: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38
0x00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Call Stack + Locals

Memory 1







Registers

Register	Value
Current	
R0	0x00000060
R1	0x000000A0
R2	0x00000000
R3	0x31323334
R4	0x35363738
R5	0x61736466
R6	0x67686A6B
R7	0x30393837
R8	0x36353433
R9	0x51574552
R10	0x54595549
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000001C
CPSR	0x600000D3
SPSR	0x00000000

User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC \$ 0x0000001C
Mode Supervisor
States 47
Sec 0.00000000

```

4:      ADR    r0,table1 ; r0 points to source
5:      ADR    r1,table2 ; r1 points to the destination
6:      MOV    r2,#2      ; 2 blocks of 8 words = 16 words (64 bytes) to move
7: Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
8:      STMEA r1!,{r3-r10} ; store the registers at their destination
9:      SUBS   r2,r2,#1   ; decrement loop counter
10:     BNE   Loop    ; UNTIL all 2 blocks of 8 registers moved
11:     Inf   B      Inf
⇒ 0x0000001C EAFFFFE B 0x0000001C
12:     0x00000020 61626364 ??VS
13:     0x00000024 65666768 STRVSB R6,[R6,-0x0768]!
14:     0x00000028 31323334 TEQCC R2,R4,LSR R3
15:     0x0000002C 35363738 LDRCC R3,[R6,-0x0738]!

```

```

block_moves aa.s
1      AREA BlockMove, CODE, READONLY
2
3      ENTRY
4      ADR    r0,table1 ; r0 points to source
5      ADR    r1,table2 ; r1 points to the destination
6      MOV    r2,#2      ; 2 blocks of 8 words = 16 words (64 bytes) to move
7 Loop   LDMFD  r0!,{r3-r10} ; REPEAT Load 32 bytes in 8 registers (r3 to r10)
8      STMEA r1!,{r3-r10} ; store the registers at their destination
9      SUBS   r2,r2,#1   ; decrement loop counter
10     BNE   Loop    ; UNTIL all 2 blocks of 8 registers moved
11     Inf   B      Inf
12     table1 dcb "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
13     dcb "12345678asdfghjk09876543QWERTYUI"
14     table2 space 64
15
16     END
17

```

Order was
kept

table1

table2

Memory 1

Address: 0

0x00000000: E2 8F 00 18 E2 8F 10 54	A0 20 02 E8 B0 07 F8
0x00000010: F8 A1 07 F8 F2 52 20 01 1A FF FF FB FA FF FE FE	
0x00000020: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38	
0x00000030: 41 42 43 44 45 46 47 48 32 33 34 35 36 37 38	
0x00000040: 31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B	
0x00000050: 30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49	
0x00000060: 61 62 63 64 65 66 67 68 31 32 33 34 35 36 37 38	
0x00000070: 41 42 43 44 45 46 47 48 31 32 33 34 35 36 37 38	
0x00000080: 31 32 33 34 35 36 37 38 61 73 64 66 67 68 6A 6B	
0x00000090: 30 39 38 37 36 35 34 33 51 57 45 52 54 59 55 49	
0x000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0x000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	

Call Stack + Locals | Memory 1