ARM Assembly Language & Programming

ARM7

Clicker question

Have you installed ARMSim#?

A. Yes
B. No

http://armsim.cs.uvic.ca/

B1.5 GNU Assembler Quick Reference

This section summarizes the more useful commands and expressions available with the GNU assembler, *gas*, when you target this assembler for ARM. Each assembly line has the format

```
{<label>:} {<instruction or directive>} @ comment
```

Unlike the ARM assembler, you needn't indent instructions and directives. Labels are recognized by the following colon rather than their position at the start of the line. The following example shows a simple assembly file defining a function *add* that returns the sum of the two input arguments:

```
.section     .text, "x"

.global     add                   @ give the symbol add external linkage

add:
            ADD      r0, r0, r1   @ add input arguments
            MOV      pc, lr       @ return from subroutine
```

GNU Assembler Directives

Here is an alphabetical list of the more common *gas* directives.

Assembler directives

.align	n
.ascii	“<string>”
.asciiz	“<string>”
.byte	<byte1> {, <byte2>, …}
.data	{<addr>}
.global	<symbol>
.text	{<addr>}
.word	<word1> {, <word2>, …}

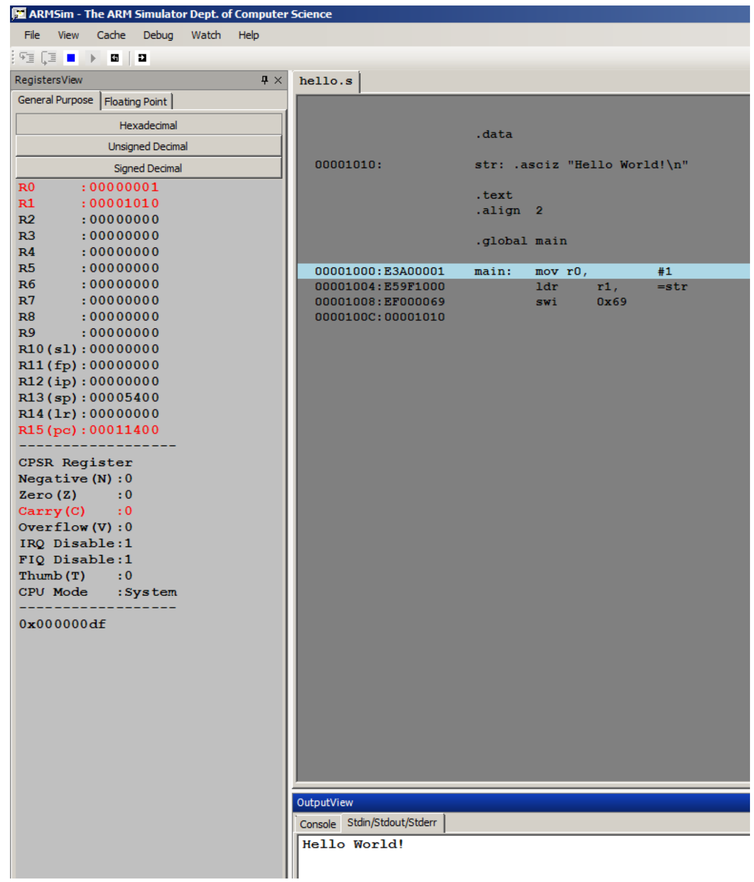
Table 4. SWI I/O operations (0x00 - 0xFF)

Opcode	Description and Action	Inputs	Outputs	EQU
`swi 0x00`	Display Character on Stdout	r0: the character		`SWI_PrChr`
`swi 0x02`	Display String on Stdout	r0: address of a null terminated ASCII string	(see also 0x69 below)	
`swi 0x11`	Halt Execution			`SWI_Exit`
`swi 0x12`	Allocate Block of Memory on Heap	r0: block size in bytes	r0:address of block	`SWI_MeAlloc`
`swi 0x13`	Deallocate All Heap Blocks			`SWI_DAlloc`
`swi 0x66`	Open File (mode values in r1 are: 0 for input, 1 for output, 2 for appending)	r0: file name, i.e. address of a null terminated ASCII string containing the name r1: mode	r0:file handle If the file does not open, a result of -1 is returned	`SWI_Open`
`swi 0x68`	Close File	r0: file handle		`SWI_Close`
`swi 0x69`	Write String to a File or to Stdout	r0: file handleor Stdout r1: address of a null terminated ASCII string		`SWI_PrStr`

Table 4. SWI I/O operations (0x00 - 0xFF)

Opcode	Description and Action	Inputs	Outputs	EQU
swi 0x6a	Read String from a File	r0: file handle r1: destination address r2: max bytes to store	r0: number of bytes stored	SWI_RdStr
swi 0x6b	Write Integer to a File	r0: file handle r1: integer		SWI_PrInt
swi 0x6c	Read Integer from a File	r0: file handle	r0: the integer	SWI_RdInt
swi 0x6d	Get the current time (ticks)		r0: the number of ticks (milliseconds)	SWI_Timer

```
.data

str: .asciz "Hello World!\n"

.text
.align 2

.global main

main: mov   r0,   #1
      ldr   r1,   =str
      swi   0x69
```



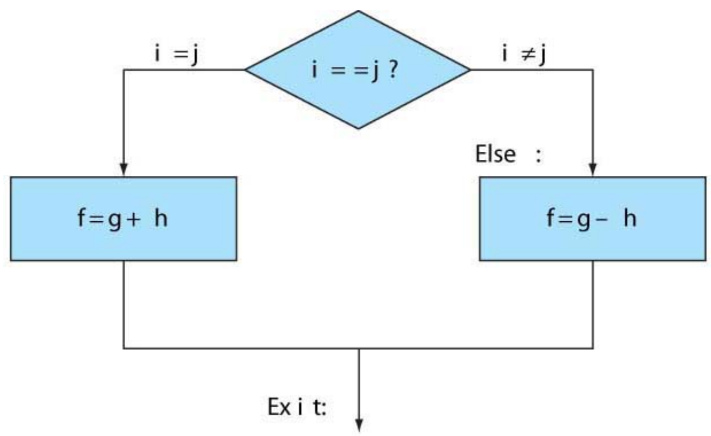
ARM Programming

➢Branch (cmp, beq, bne)

♠ “If-then-else”

Example:

```
if(i==j)   f = g +h;
else       f = g – h;
```



assume f, g, h, i, and j are in r0, r1, r2, r3, and r4 respectively

```
         CMP    r3, r4
         BNE    Else
         ADD    r0, r1, r2
         B      Exit
Else:    sub    r0, r1, r2
Exit:
```

A more compact and efficient version:

```
CMP r3,r4
ADDEQ r0,r1,r2 ; f = g + h (skipped if i ≠ j)
SUBNE r0,r1,r2 ; f = g - h (skipped if i = j)
```

signed versus unsigned comparison

Suppose

```
r0 = 1111 1111 1111 1111 1111 1111 1111 1111two
r1 = 0000 0000 0000 0000 0000 0000 0000 0001two
```

If the following instructions are executed

0x0000 1000: CMP r0, r1
0x0000 1004: BLO L1
0x0000 1008: BLT L2

Where will be the PC at?

A: L1
B: L2
C: 0x0000 100C
D: 0x0000 1010

Value	Meaning	Value	Meaning
0	EQ (EQual)	8	HI (unsigned HIgher)
1	NE (Not Equal)	9	LS (unsigned Lower or Same)
2	HS (unsigned Higher or Same)	10	GE (signed Greater than or Equal)
3	LO (unsigned LOwer)	11	LT (signed Less Than)
4	MI (MInus, <0)	12	GT (signed Greater Than)
5	PL - (PLus, >=0)	13	LE (signed Less Than or Equal)
6	VS (oVerflow Set, overflow)	14	AL (Always)
7	VC (oVerflow Clear, no overflow)	15	NV (reserved)

FIGURE 2.9.5 Encodings of Options for Cond field.

This example is in page 108 - 109, of Chapter 02_COD 4e ARM.pdf).
r0 = 1111 1111 1111 1111 1111 1111 1111 1111
r1 = 0000 0000 0000 0000 0000 0000 0000 0001

So, the result of r0 - r1 is computed (**<u>see page B1-15 in Appendix_B1.pdf for details</u>**)
as r0 + (-r1) = r0 + (~r1 +1)

```
     1111 1111 1111 1111 1111 1111 1111 1111     (r0)
 +)  1111 1111 1111 1111 1111 1111 1111 1111     (~r1 + 1)
--------------------------------------------------------------------------------
    11111 1111 1111 1111 1111 1111 1111 1110
```

Based on this result,
N = 1 (the result is negative, treated as two's complement)
Z = 0 (the result is not zero)
C = 1 (there is carry out of the left-most bit)
V = 0 (there is no overflow)

Therefore, the instruction "BLO" is not taken because of suffix
"LO" indicates unsigned lower which is taken when carry bit is
clear (**<u>See Table B1.2, in Appendix_B1.pdf for details</u>**).
Instead instruction "BLT" is taken when N != V, which is the case as
shown above.

Condition Mnemoics

cond	Mnemonic	Name	CondEx
0000	EQ	Equal	Z
0001	NE	Not equal	$\bar{Z}$
0010	CS / HS	Carry set / Unsigned higher or same	C
0011	CC / LO	Carry clear / Unsigned lower	$\bar{C}$
0100	MI	Minus / Negative	N
0101	PL	Plus / Positive of zero	$\bar{N}$
0110	VS	Overflow / Overflow set	V
0111	VC	No overflow / Overflow clear	$\bar{V}$
1000	HI	Unsigned higher	$\bar{Z}C$
1001	LS	Unsigned lower or same	$Z\ OR\ \bar{C}$
1010	GE	Signed greater than or equal	$\overline{N \oplus V}$
1011	LT	Signed less than	$N \oplus V$
1100	GT	Signed greater than	$\bar{Z}(\overline{N \oplus V})$
1101	LE	Signed less than or equal	$Z\ OR\ (N \oplus V)$
1110	AL (or none)	Always / unconditional	ignored

Value	Meaning	Value	Meaning
0	EQ (EQual)	8	HI (unsigned HIgher)
1	NE (Not Equal)	9	LS (unsigned Lower or Same)
2	HS (unsigned Higher or Same)	10	GE (signed Greater than or Equal)
3	LO (unsigned LOwer)	11	LT (signed Less Than)
4	MI (MInus, <0)	12	GT (signed Greater Than)
5	PL - (PLus, >=0)	13	LE (signed Less Than or Equal)
6	VS (oVerflow Set, overflow)	14	AL (Always)
7	VC (oVerflow Clear, no overflow)	15	NV (reserved)

FIGURE 2.9.5 Encodings of Options for Cond field.

TABLE B1.2 ARM condition mnemonics.

<cond>	Instruction is executed when	cpsr condition
{|AL}	ALways	TRUE
EQ	EQual (last result zero)	Z==1
NE	Not Equal (last result nonzero)	Z==0
{CS|HS}	Carry Set, unsigned Higher or Same (following a compare)	C==1
{CC|LO}	Carry Clear, unsigned LOwer (following a comparison)	C==0
MI	MInus (last result negative)	N==1
PL	PLus (last result greater than or equal to zero)	N==0
VS	V flag Set (signed overflow on last result)	V==1
VC	V flag Clear (no signed overflow on last result)	V==0
HI	unsigned HIgher (following a comparison)	C==1 && Z==0
LS	unsigned Lower or Same (following a comparison)	C==0 || Z==1
GE	signed Greater than or Equal	N==V
LT	signed Less Than	N!=V
GT	signed Greater Than	N==V && Z==0
LE	signed Less than or Equal	N!=V || Z==1
NV	NeVer—ARMv1 and ARMv2 only—*DO NOT USE*	FALSE

CMP Compare two 32-bit integers

```
1. CMP<cond>     Rn, #<rotated_immed>     ARMv1
2. CMP<cond>     Rn, Rm {, <shift>}       ARMv1
3. CMP           Ln, #<immed8>            THUMBv1
4. CMP           Rn, Rm                   THUMBv1
```

Action

```
1. cpsr   flags   set on the result of     (Rn - <rotated_immed>)
2. cpsr   flags   set on the result of     (Rn - <shifted_Rm>)
3. cpsr   flags   set on the result of     (Ln - <immed8>)
4. cpsr   flags   set on the result of     (Rn - Rm)
```

Notes

- In the *cpsr*: N = <*Negative*>, Z = <*Zero*>, C = <*NoUnsigned-Overflow*>, V = <*SignedOverflow*>. The carry flag is set this way because the subtract $x - y$ is implemented as the add $x + \sim y + 1$. The carry flag is one if $x + \sim y + 1$ overflows. This happens when $x \geq y$ (equivalently when $x - \hat{A}y$ doesn't overflow).
- If *Rn* or *Rm* is *pc*, then the value used is the address of the instruction plus eight bytes for ARM instructions, or plus four bytes for Thumb instructions.

Example

```
CMP    r0, r1, LSR#2   ; compare r0 with (r1/4)
BHS    label           ; if (r0 >= (r1/4)) goto label;
```

> For 2.16.1-2, the values of r0 have a one as their left most
> bit. Should we assume these values are signed integers or
> unsigned?

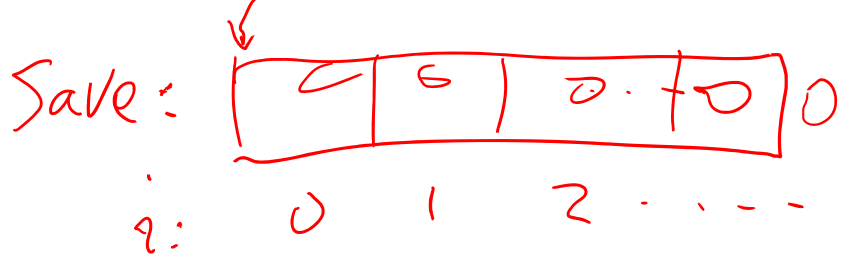
Yes, you assume they are signed integers. The instruction "CMP r0, r1" subtracts value in r0 by value in r1. What matters is the subtraction result, though not saved, based on which the high 4 flag bits NZCV of the current program status register (CPSR) are set accordingly. It is these four bits that determine the behavior of the next conditional instruction.

Loop

Example:

```
while (save[i] == k)
        i += 1;
```

assume i is in r3, k is in r5, and the base of the array is in r6.




```
Loop:   ADD     r12, r6, r3, LSL #2
        LDR     r0, [r12, #0]
        CMP     r0, r5
        BNE     Exit
        ADD     r3, r3, #1
        B       Loop
Exit:
```

➢ Procedure call and stack

motivation: abstraction and code reusability

e.g., a * b

C code:

```c
while(b != 0) {
    sum = sum + a;
    b = b -1;
}
```

ARM Assembly code:

```
@ assume a is in r0, b is in r1

        sub    r7, r7, r7      @ initialize
Loop:   cmp    r1, #0
        beq    Done            @if b == 0
        add    r7, r7, r0      @sum += b a
        sub    r1, r1, 1       @a = a-1
        b      Loop
Done:
```

How do we make use of that piece of code (let's give it a name MUL)?

E.g., a*b + c

solution 1: ("*cut-and-paste*": embed the code of MUL to where it is needed)
(a valid approach under certain circumstances, inline function in C++)
@ assume a, b, c, are in r0, r1, r7

```
        mov     r2, r7   @ move c out of r7, as it is to be used by MUL

        sub     r7, r7, r7      @initialize
Loop:   cmp     r1, #0
        beq     Done            @ if b == 0
        add     r7, r7, r0      @ sum += a
        sub     r1, r1, #1      @b = b-1
        b       Loop
Done:
        add     r7, r7, r2      @add c to a*b
```

Can we reuse the routine "MUL" multiple times?
We don't want to write the same piece of code wherever it is used; we may not know how many times it is to be used when we write the code, e.g., multiply integer *a* by itself for *b* times where b is an integer to be read from the keyboard.

So we want to write the MUL code once and reuse it.

E.g., a*a + b*b + c*c

Solution 2: ("call-by-name": Jump to the MUL code and jump back when it is done)

```
@ assume  a, b, c, are in r2, r3, r4

        mov     r0, r2              @ move a to r0
        move    r1, r3              @ move b to r1
        b        MUL                @ call MUL, product a*b will be put in r7
Done:   add     r7, r7, r4          @add c to a*b

MUL:
        sub     r7, r7, r7          @initialize
Loop:   cmp     r1, #0
        beq     Done                @if b == 0
        add     r7, r7, r0   # sum += a
        sub     r1, r1, #1      # b = b-1
        b       Loop
```

We want "MUL" to return to the next instruction where it is called. In this example, the return address "Done" is hard coded -- so it can only return to one place.

@ For example, let's compute a*a + b*b
@ assume a, b, c, are in r2 and r3

```
        mov r0, r2          @ copy a to r0
        mov  r0, r2         @ copy a to r1
        b MUL               @ call MUL, product a*a will be in r7
Done:   mov r4, r7          @ move a*a to r4

        mov r0, r3          @ copy b to r0
        mov r1, r3          @ copy b to r1
        b MUL               @ call MUL, product b*b will be in r7
Done?:  add r7, r7, r4      @ a*a + b*b is put to r7

MUL:
        sub     $v0, $v0, $v0   # initialize
Loop:
        beq     $0, $a1, Done   # if a == 0
        add     $v0, $v0, $a0   # sum += a
        subi    $a1, $a1, 1     # b = b-1
        j       Loop
```

X

A hard-coded return address "Done" is not working well -- it can only return to one place.

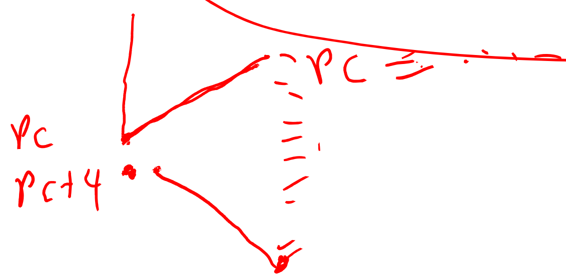
Solution:

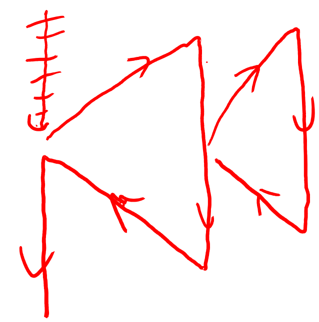
1. save the return address to a register before calling the subroutine
2. jump to the saved return address when the subroutine is done.

Hardware support:

bl ProcedureAddress @save the return address (PC+4) into a designated @register r14 and then jump to the address of the callee.

mov pc, r14 @ reset PC to the return address stored in register r14.





@ For example, let’s compute a*a + b*b
@ assume a and b are in r3 and r4

```
        mov     r0, r3          @ copy a to r0
        mov     r1, r3          @ copy ~~b~~ a to r1
        bl      MUL             @ call MUL,  address of next instruction is put in r14
        mov     r5, r7          @ move a*a to r5

        mov     r0, r4          @ copy b to r0
        mov     r1, r4          @ copy b to r1
        bl      MUL             @call MUL,   address of next instruction is put in r14
        add     r7, r7, r5      @a*a + b*b is put to r7

MUL:
        sub     r7, r7, r7      @ initialize
Loop:   cmp     r1, #0
        beq     Done            @ if a == 0
        add     r7, r7, r0      @ sum += a
        sub     r1, r1, #1      @ b = b-1
        b       Loop
Done:   mov     pc, r14         @ a procedure always ends with this instruction
```

Six steps

1. (caller) place parameters in a location where the procedure (callee) can access them (r0, r1, r2, r3)
2. transfer control to the procedure. (**BL**)

caller

---

3. acquire the storage resources needed for the procedure.
4. perform the desired task
5. place the result value in a place where the caller can access it.
6. return control to the point next to where the program is called. (**MOV pc, r14**)

callee

ARM conventions:
- r0 - r3, r12: registers for storing arguments or scratch registers to used by the callee (not preserved)
- r4-r11: registers that need to be preserved, if used by callee
- lr: register storing the return address (r14)
- sp: stack pointer (r13)

Clicker question:

What is a leaf-procedure?

A. It is a routine you do during the fall to clean up the falling leaves.
B. It is a procedure that does not call any other procedures
C. It is a procedure that is not called by any other procedures
D. It is a procedure that calls other procedures but itself
E. It is a procedure that calls only itself.

What if a procedure calls another procedure?

.data

Spill

Invocation

```
…
proc1(a);
…
…
…
.
proc1(A) {
   …
   …
    proc2(A);
   …
}

proc2(B) {
   …
   …
   …
}
```

PC →

```
Addr:
…
.
*
.
.
.
.
.
.
.
**
.
.
.
.
.
.
.
```

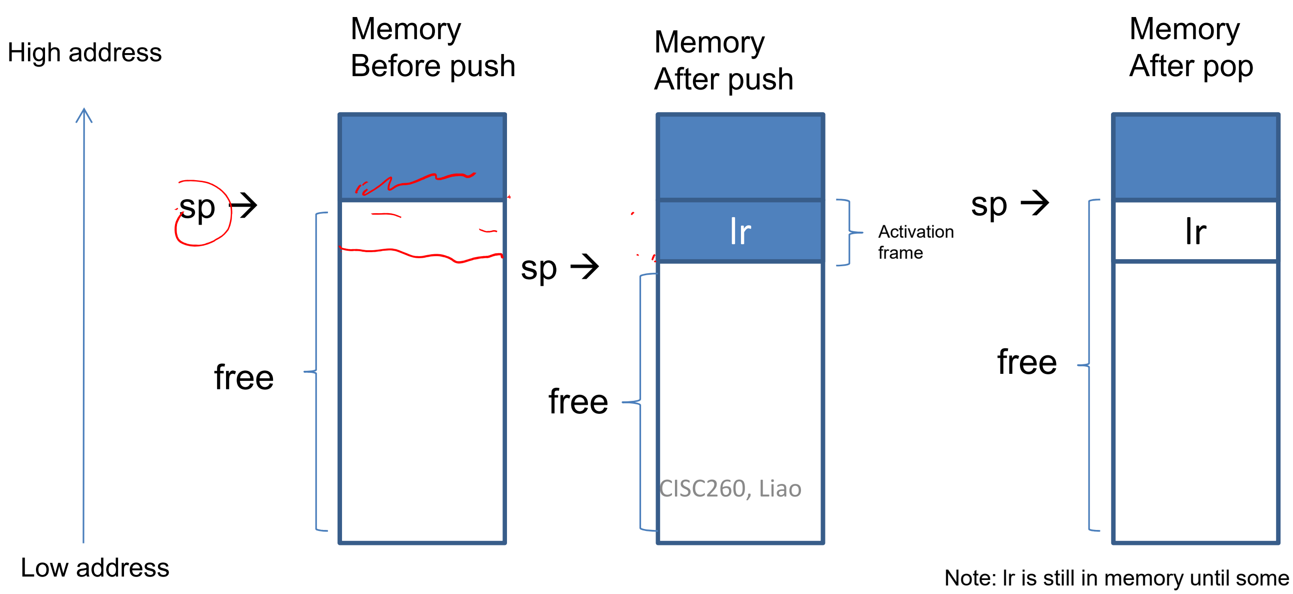
```
        bl      proc1       @ r14 = *
        …
        …
        …
        …
proc1:
        …
        …       STR   Proc1  r14
        bl      proc2       @r14 = * *
        …       LDR   r14
        mov     pc, r14     @r14 has wrong
                             return address
proc2:
        …
        …
        …
        mov     pc, r14
```

Follow the PC and monitor the value change in r14

Solution: Stack (LIFO) Last In First Out

Acquire storage space in main memory, the acquired space is called activation frame

```
SUB     sp, sp, #4      @ push
STR     r14, [sp]       @ push
…
…
…
LDR     r14, [sp]       @ pop
ADD     sp, sp, #4      @ pop
MOV     pc, r14
```

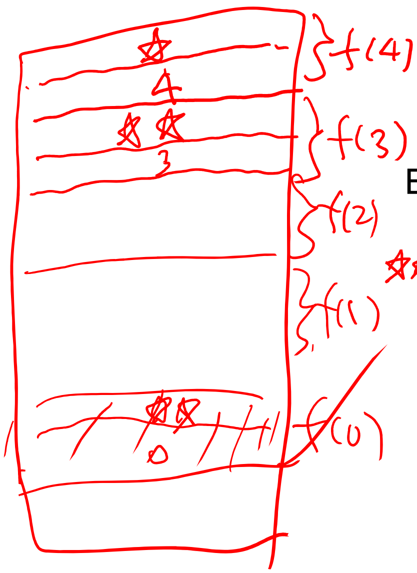


Note: lr is still in memory until some new value is written in the same location. Potential security loophole.

```
Main() {
  int n = 4;
  int res = fact(n);
  printf(res);
}

int fact (int n) {
  if(n < 1) return (1);
  else  return (n * fact (n-1) );
}
```

fact(4)

```c
int fact (int n) {
  if(n < 1) return (1);
  else  return (n * fact (n-1) );
}
```



Overhead cost

```
fact:
    SUB sp, sp, #8            @move stack pointer downwards to accommodate 2 items
    STR  lr, [sp, #0]         @store the return address
    STR  r1, [sp,#4]          @store the argument n

    CMP r1, #1                @check if n < 1
    BGE   Else                @ branch to Else  if n >= 1

    MOV r1, #1                @ make a value of 1, and save it to register $r1
    ADD sp, sp, #8            @ POP 2 items off the stack; lr and r1 haven't been overwritten,
                              @ so need not to restore
    MOV pc, lr                @ return

Else:  SUB r1, r1, #1         @ n = (n-1)
    BL  fact                  @ call fact with argument (n-1); lr and r1 are overwritten
    MOV r2, r1
    LDR   r1, [sp, #4]        @ restore r1
    LDR   lr,  [sp, #0]       @ restore r14
    ADD  sp,  sp, #8          @ move stack pointer upwards

    MUL  r1, r1, r2           @ multiply fact(n-1) in r2 with n in r1 and write the result to r1

    MOV pc, lr                @ return to the caller
```

Q: The instruction MOV r1, #1 will overwrite what's already in r1. Is that a problem?

A. Yes
B. No

F(4)
= 4*F(3)
= 4*3*F(2)
= 4*3*2*F(1)
= 4*3*2*1*F(0)
= 4*3*2*1*1

Address:

```
0x 0000 1000    fact: SUB sp, sp, #8
0x 0000 1004          STR  lr, [sp, #0]
0x 0000 1008          STR  r1, [sp,#4]

0x 0000 100C          CMP r1, #1
0x 0000 1010          BGE   Else
0x 0000 1014          MOV r1, #1
0x 0000 1018          ADD sp, sp, #8
0x 0000 101C          MOV pc, lr

0x 0000 1020    Else: SUB r1, r1, #1
0x 0000 1024          BL  fact
0x 0000 1028          MOV r2, r1
0x 0000 102C          LDR   r1, [sp, #4]
0x 0000 1030          LDR   lr,  [sp, #0]
0x 0000 1034          ADD  sp,  sp, #8

0x 0000 1038          MUL  r1, r1, r2
0x 0000 103C          MOV pc, lr
```

Note: The following is for the 32-bit ARM7, see Chapter 02_COD 4e ARM

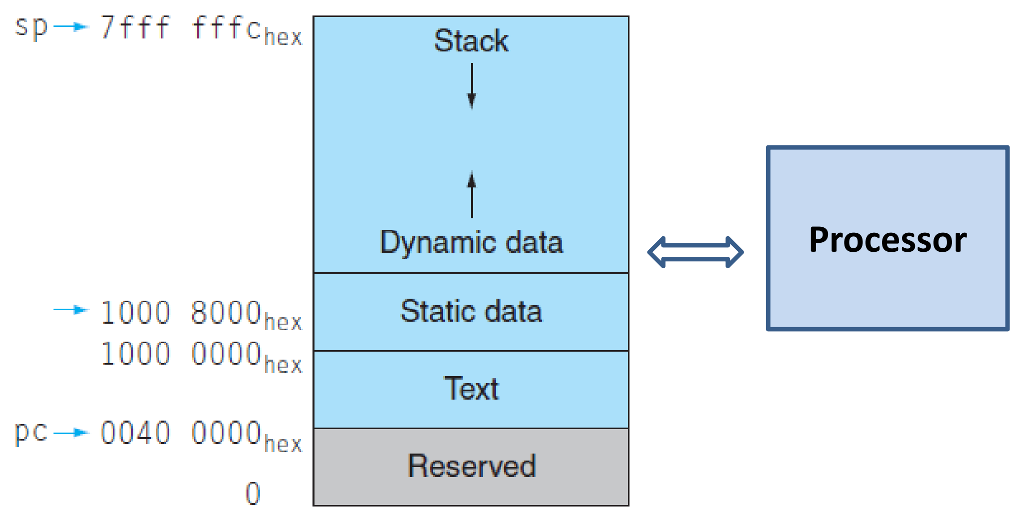
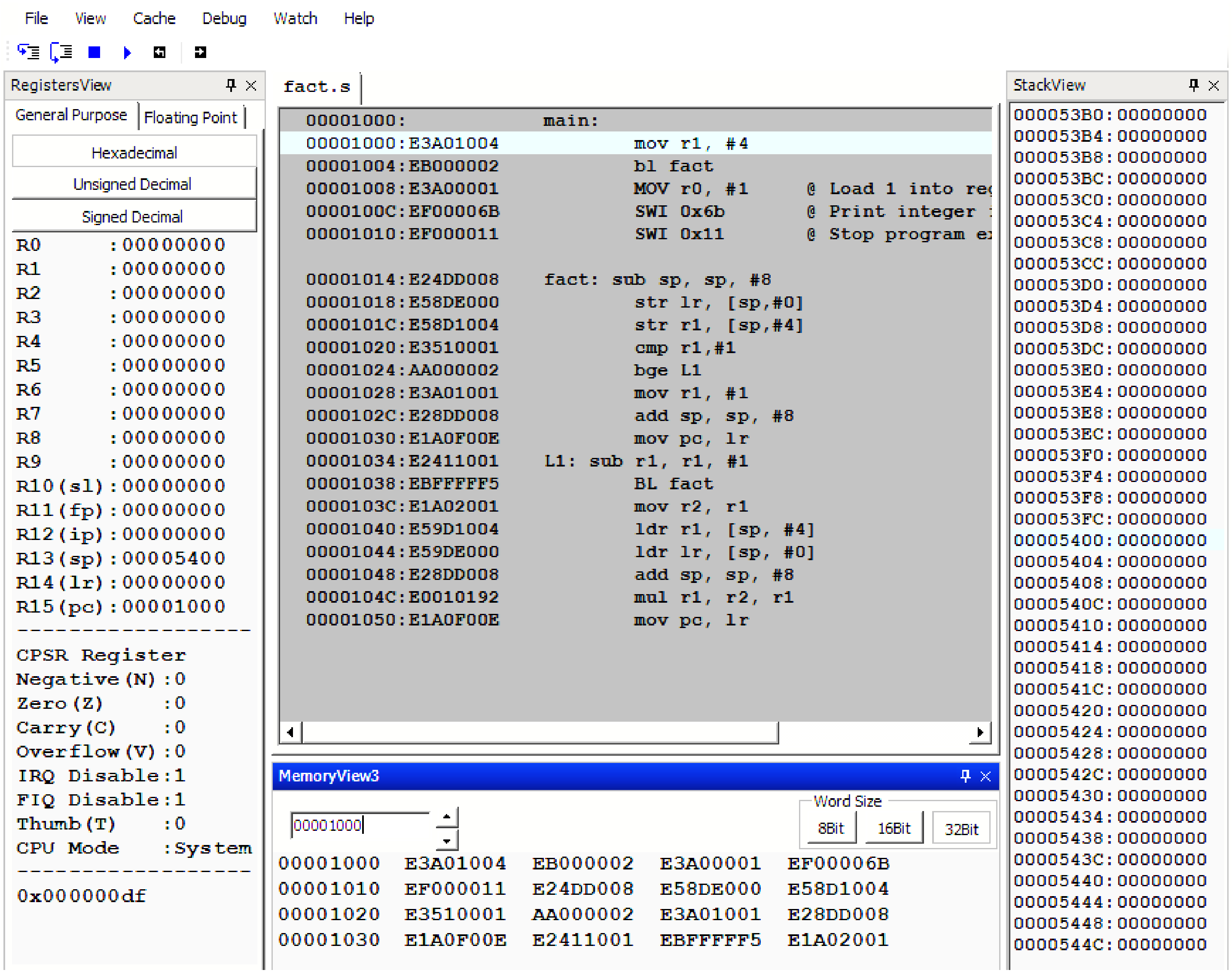
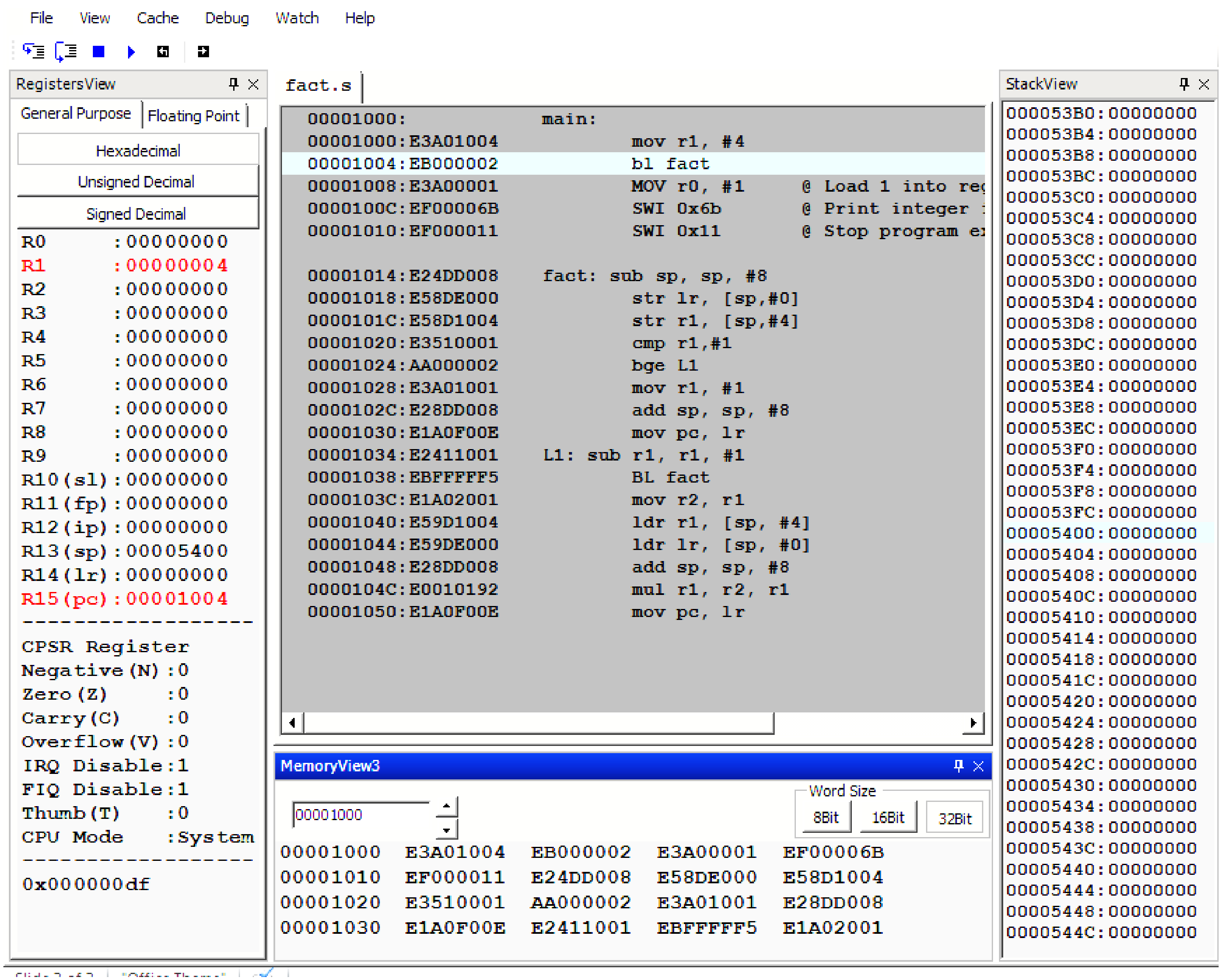
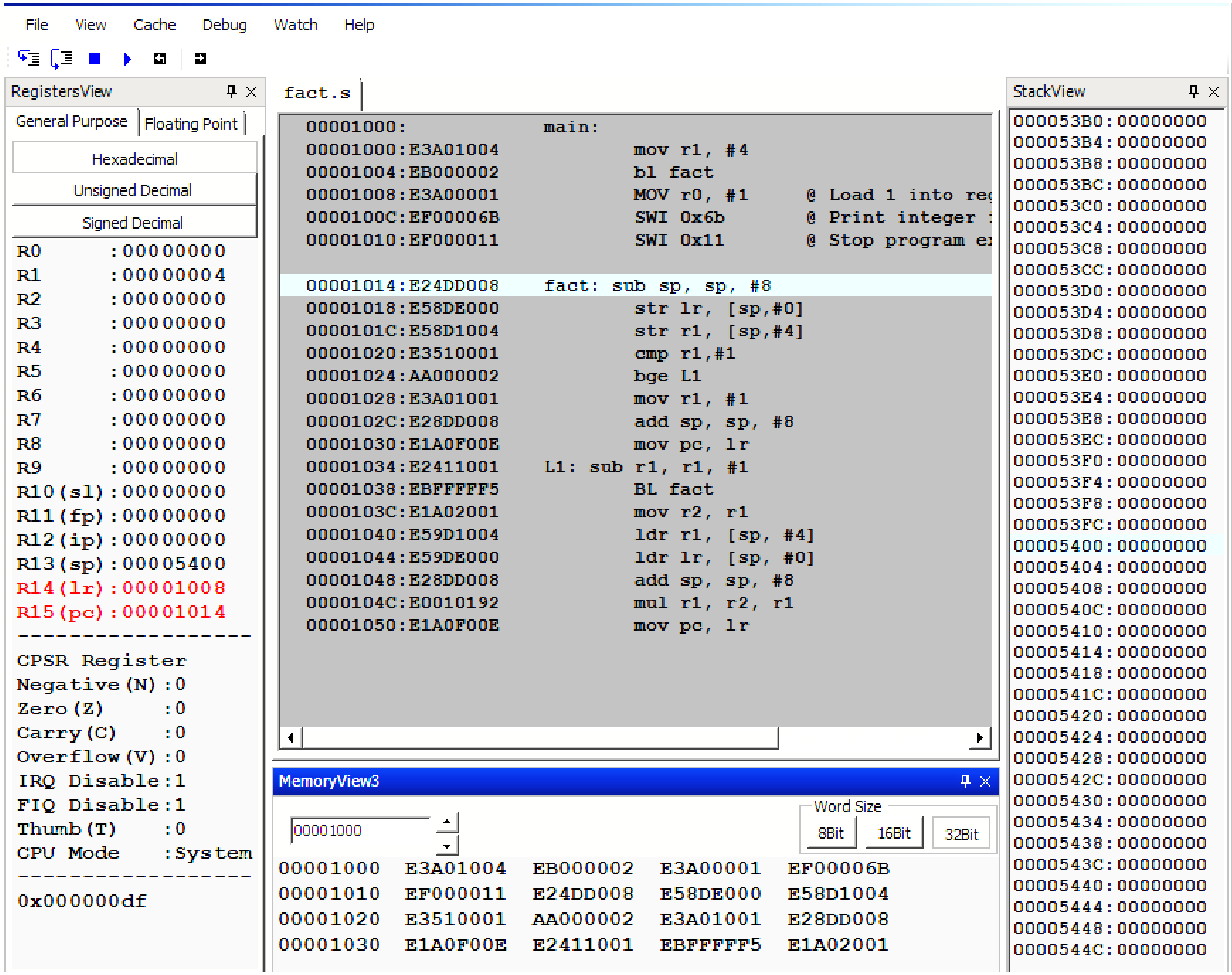
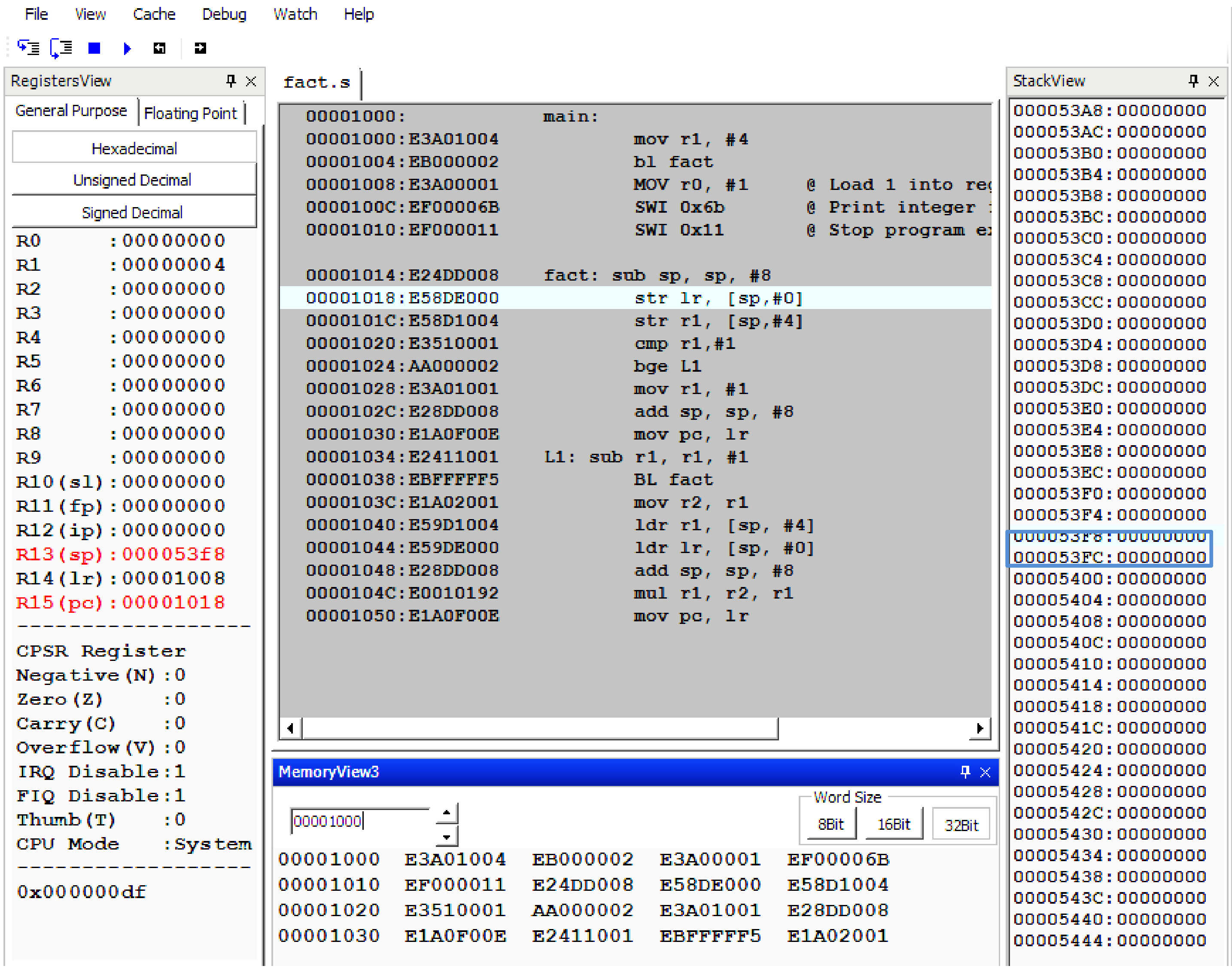


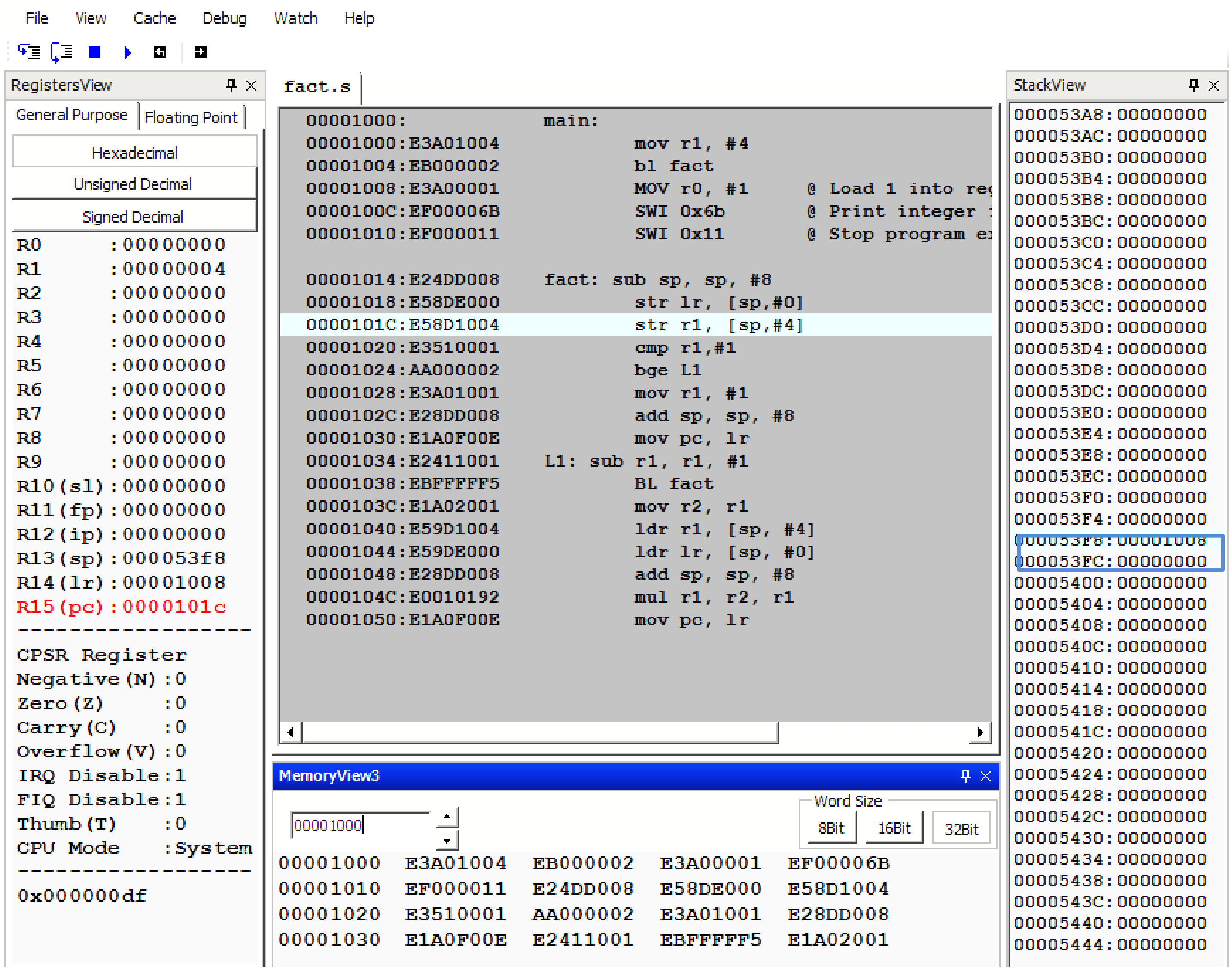
FIGURE 2.13 Typical ARM memory allocation for program and data. These addresses are only a software convention, and not part of the ARM architecture. The stack pointer is initialized to `7fff fffc`$_{hex}$ and grows down toward the data segment. At the other end, the program code ("text") starts at `0040 0000`$_{hex}$. The static data starts at `1000 0000`$_{hex}$. Dynamic data, allocated by `malloc` in C and by `new` in Java, is next. It grows up toward the stack in an area called the heap.

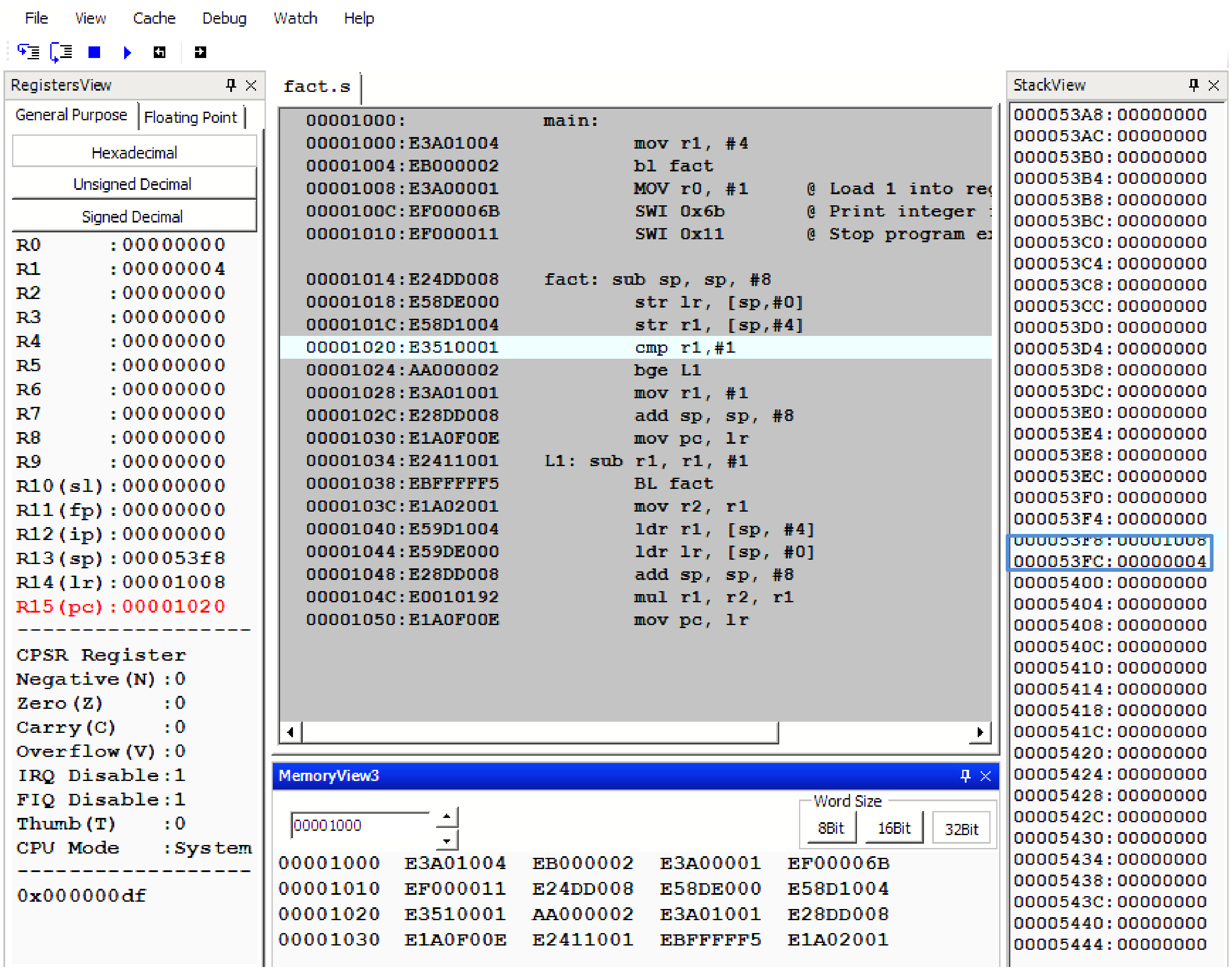

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000000
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001000
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x000000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into reg
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program ex
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit
16Bit
32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053EC:00000000
000053F0:00000000
000053F4:00000000
000053F8:00000000
000053FC:00000000
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000
00005448:00000000
0000544C:00000000

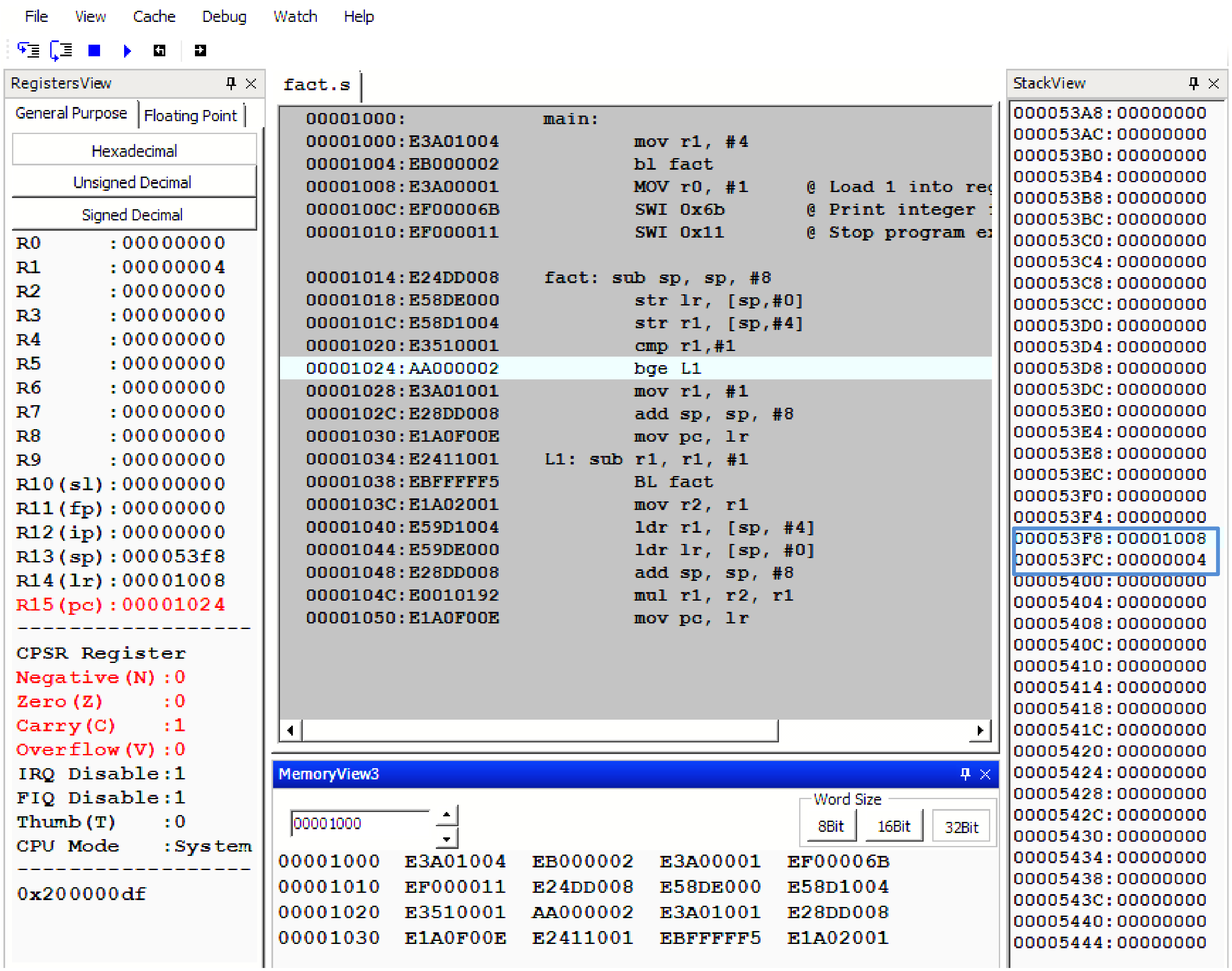

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000004
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001004
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x000000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053EC:00000000
000053F0:00000000
000053F4:00000000
000053F8:00000000
000053FC:00000000
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000
00005448:00000000
0000544C:00000000

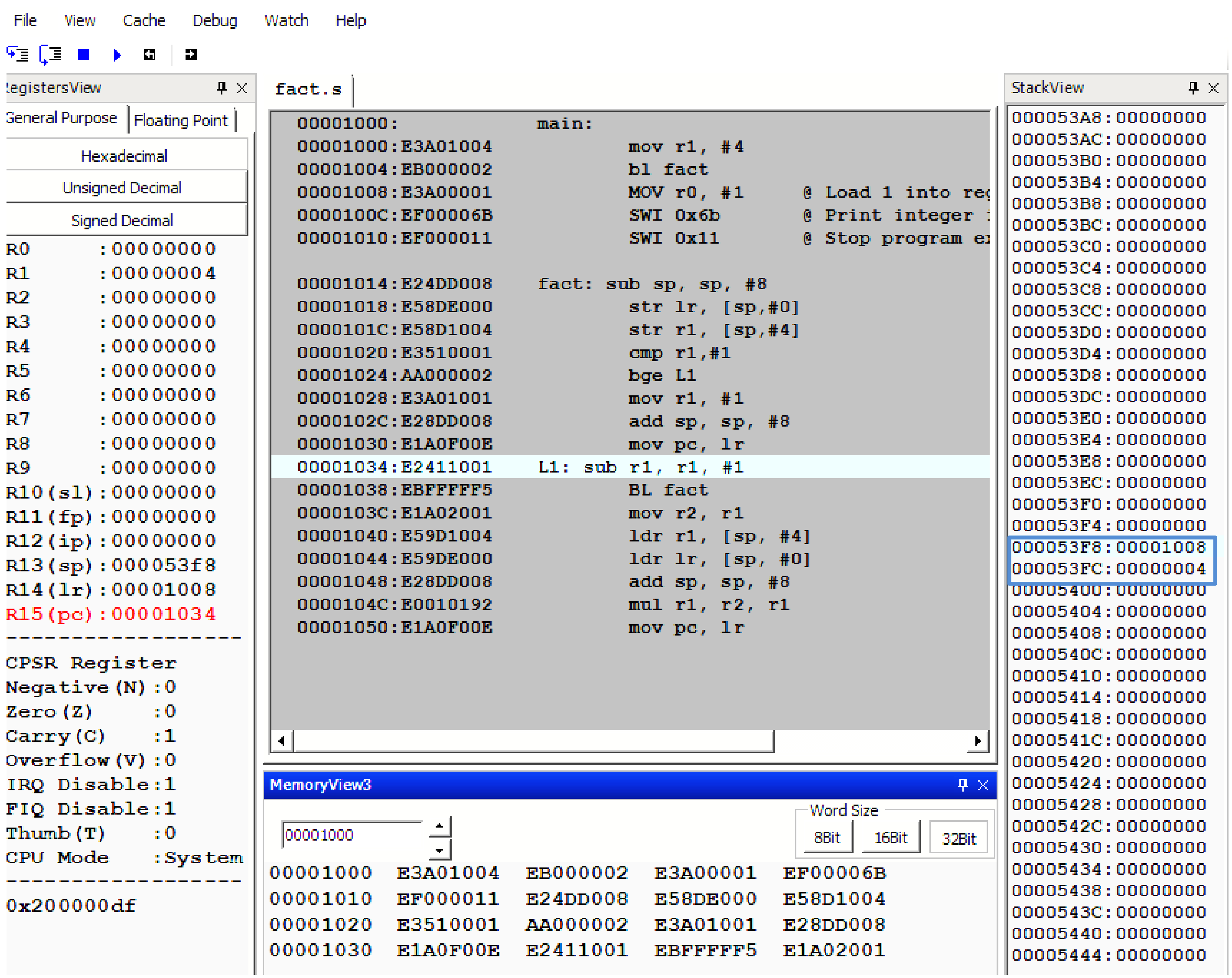

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000004
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00001008
R15(pc):00001014
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x000000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053EC:00000000
000053F0:00000000
000053F4:00000000
000053F8:00000000
000053FC:00000000
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000
00005448:00000000
0000544C:00000000

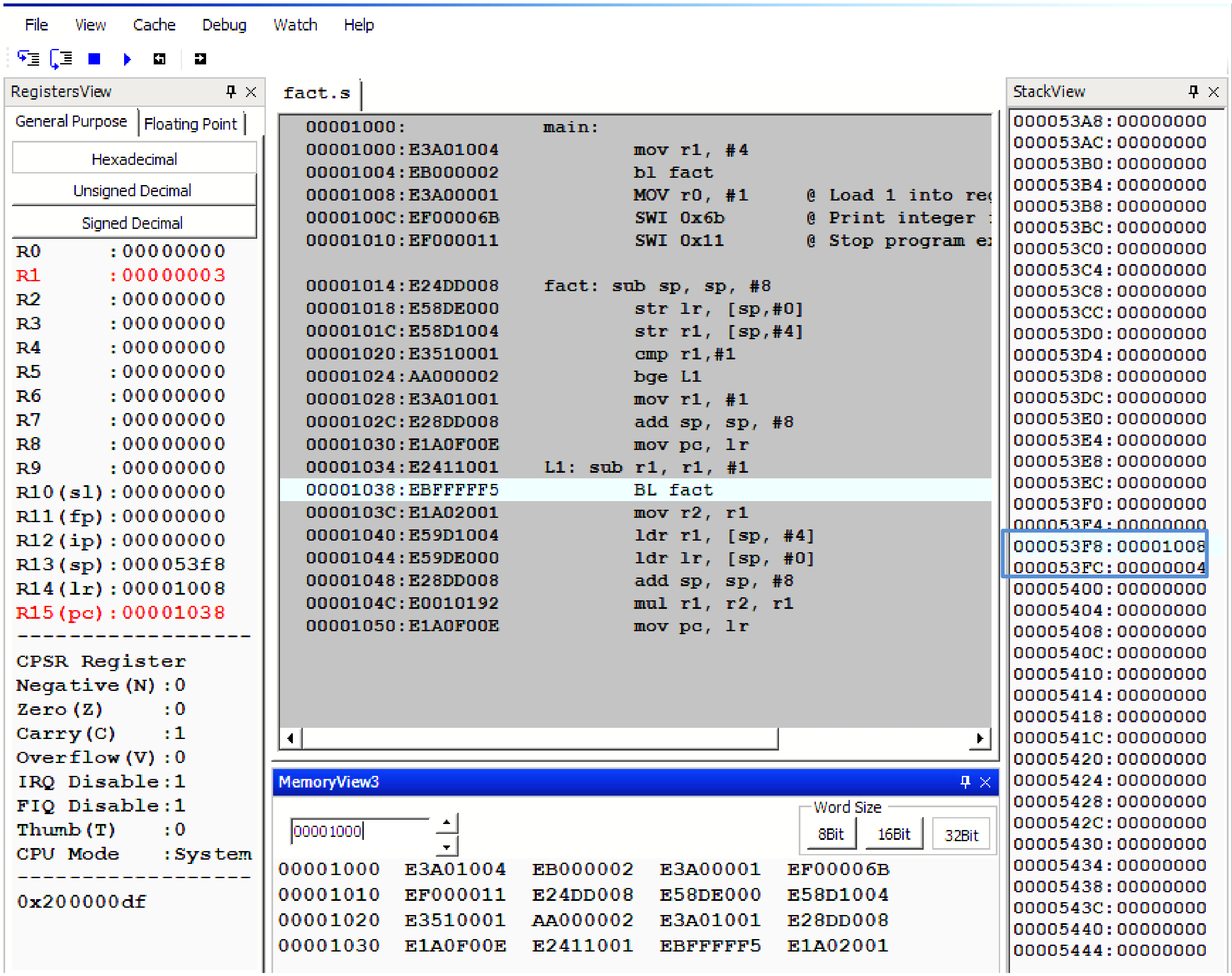

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000004
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f8
R14(lr):00001008
R15(pc):00001018
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x000000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053EC:00000000
000053F0:00000000
000053F4:00000000
000053FC:00000000
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000

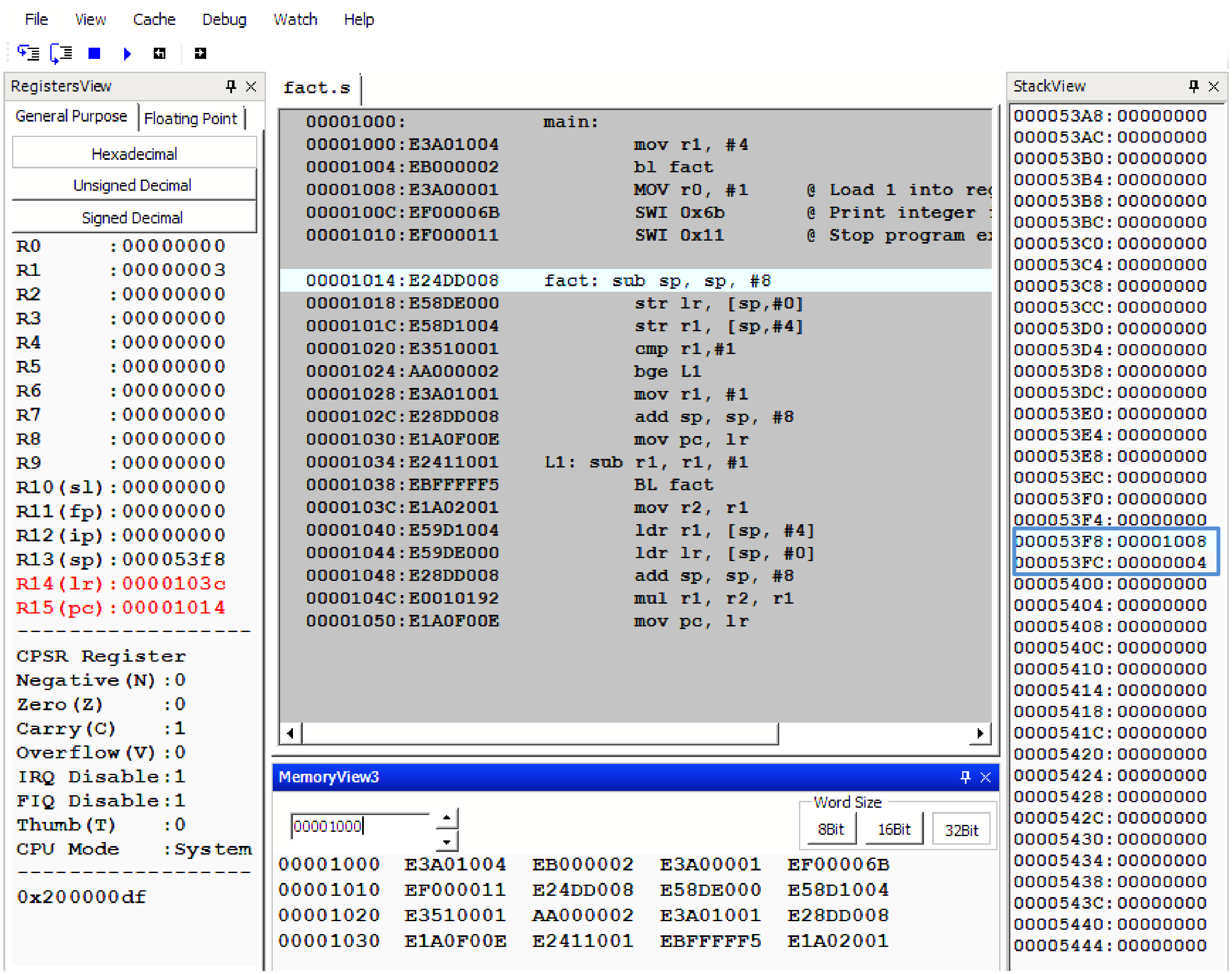

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000004
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f8
R14(lr):00001008
R15(pc):0000101c
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x000000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053EC:00000000
000053F0:00000000
000053F4:00000000
000053F8:00001008
000053FC:00000000
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000

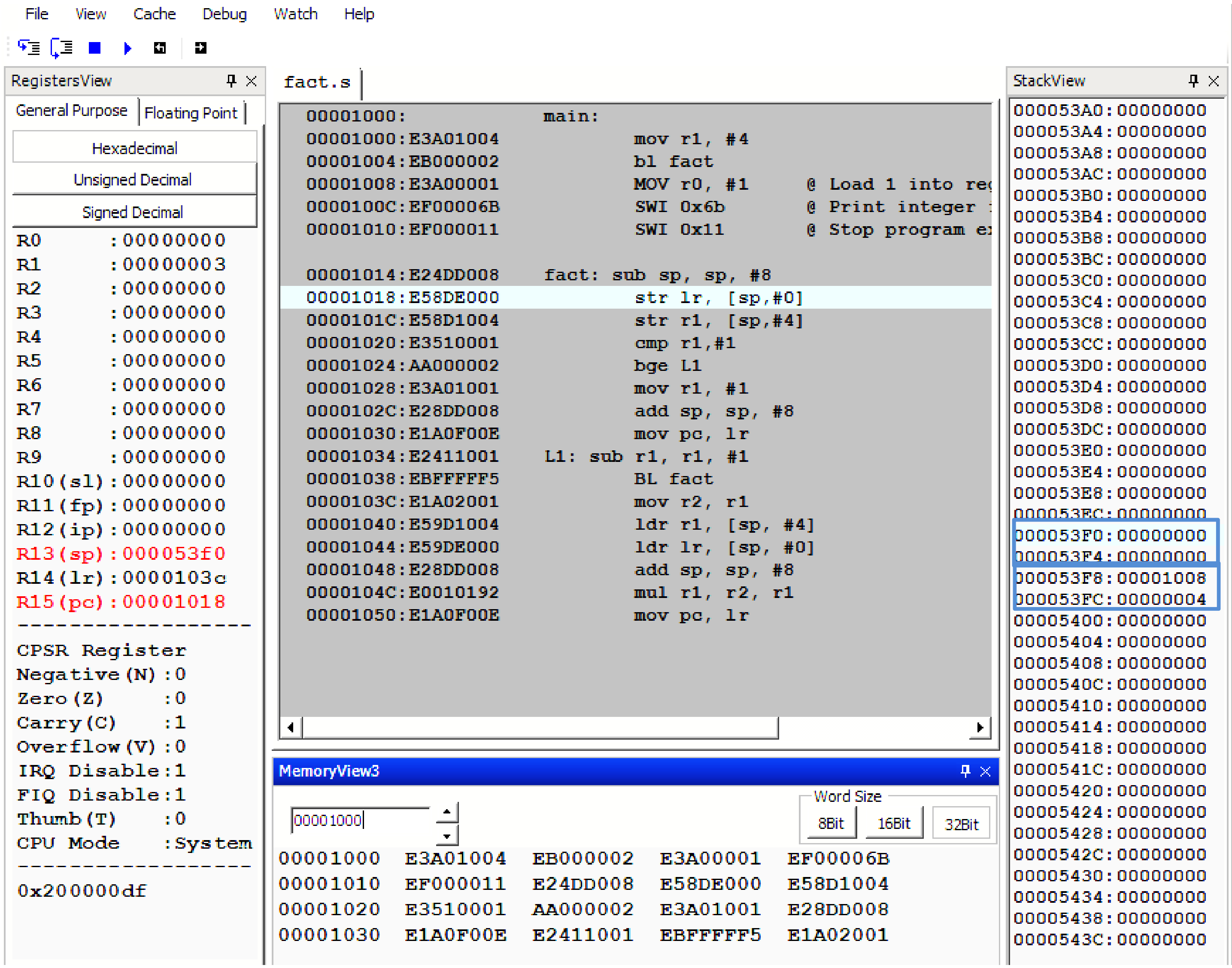

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000004
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f8
R14(lr):00001008
R15(pc):00001020
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x000000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053EC:00000000
000053F0:00000000
000053F4:00000000
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000

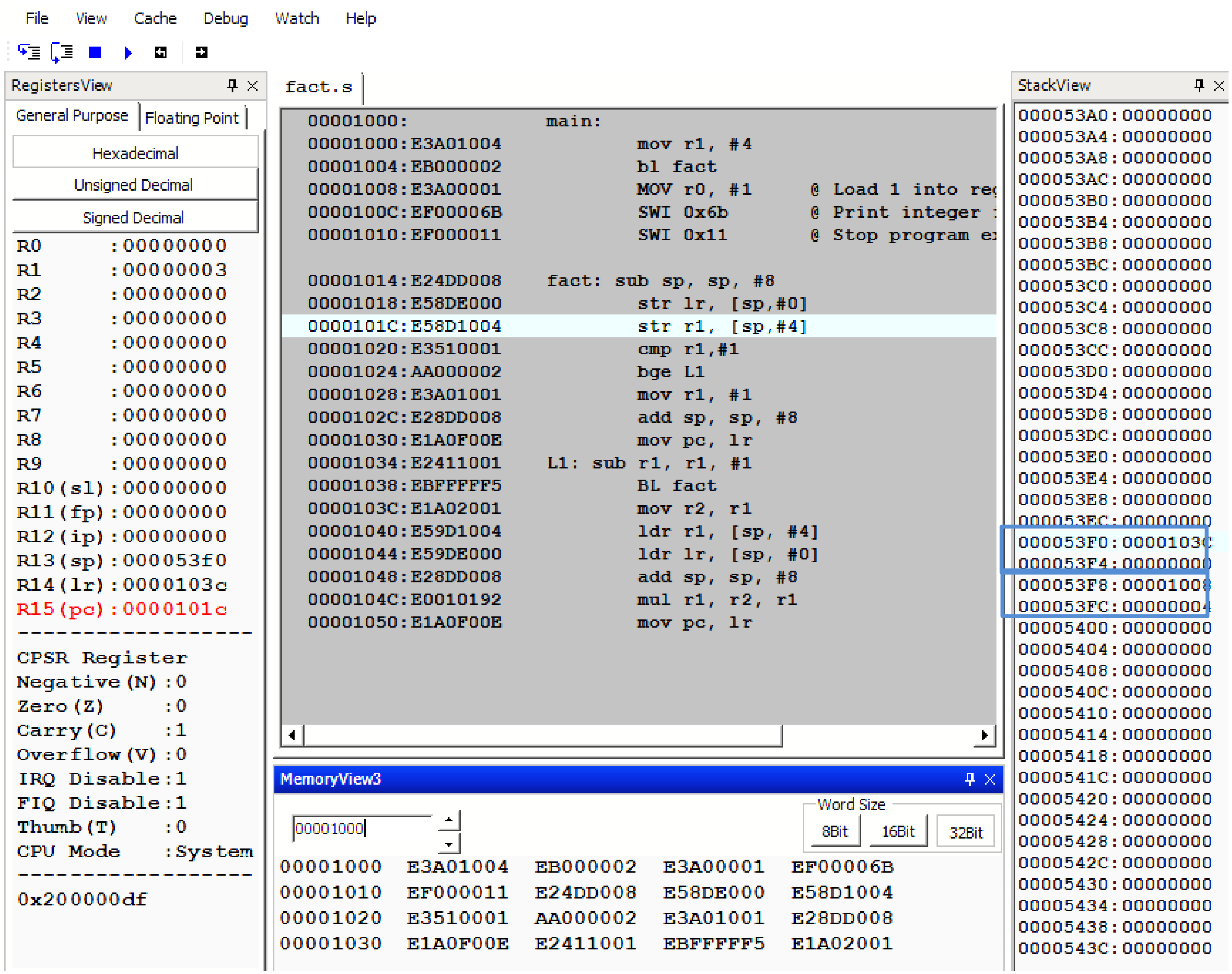

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000004
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f8
R14(lr):00001008
R15(pc):00001024
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x200000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into reg
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit
16Bit
32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053EC:00000000
000053F0:00000000
000053F4:00000000
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000

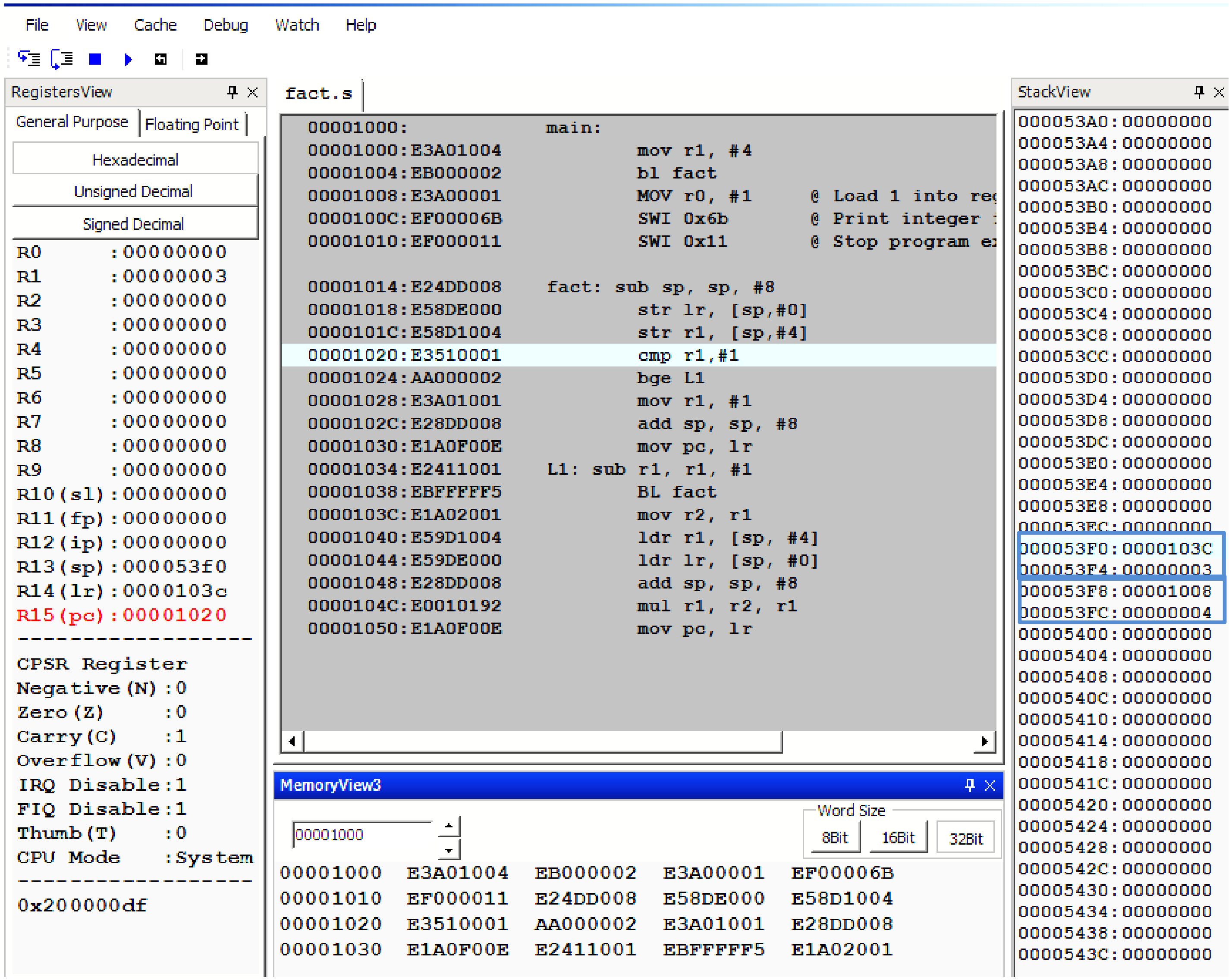

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000004
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f8
R14(lr):00001008
R15(pc):00001034
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x200000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053EC:00000000
000053F0:00000000
000053F4:00000000
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000


File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000003
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f8
R14(lr):00001008
R15(pc):00001038
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x200000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053EC:00000000
000053F0:00000000
000053F4:00000000
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000

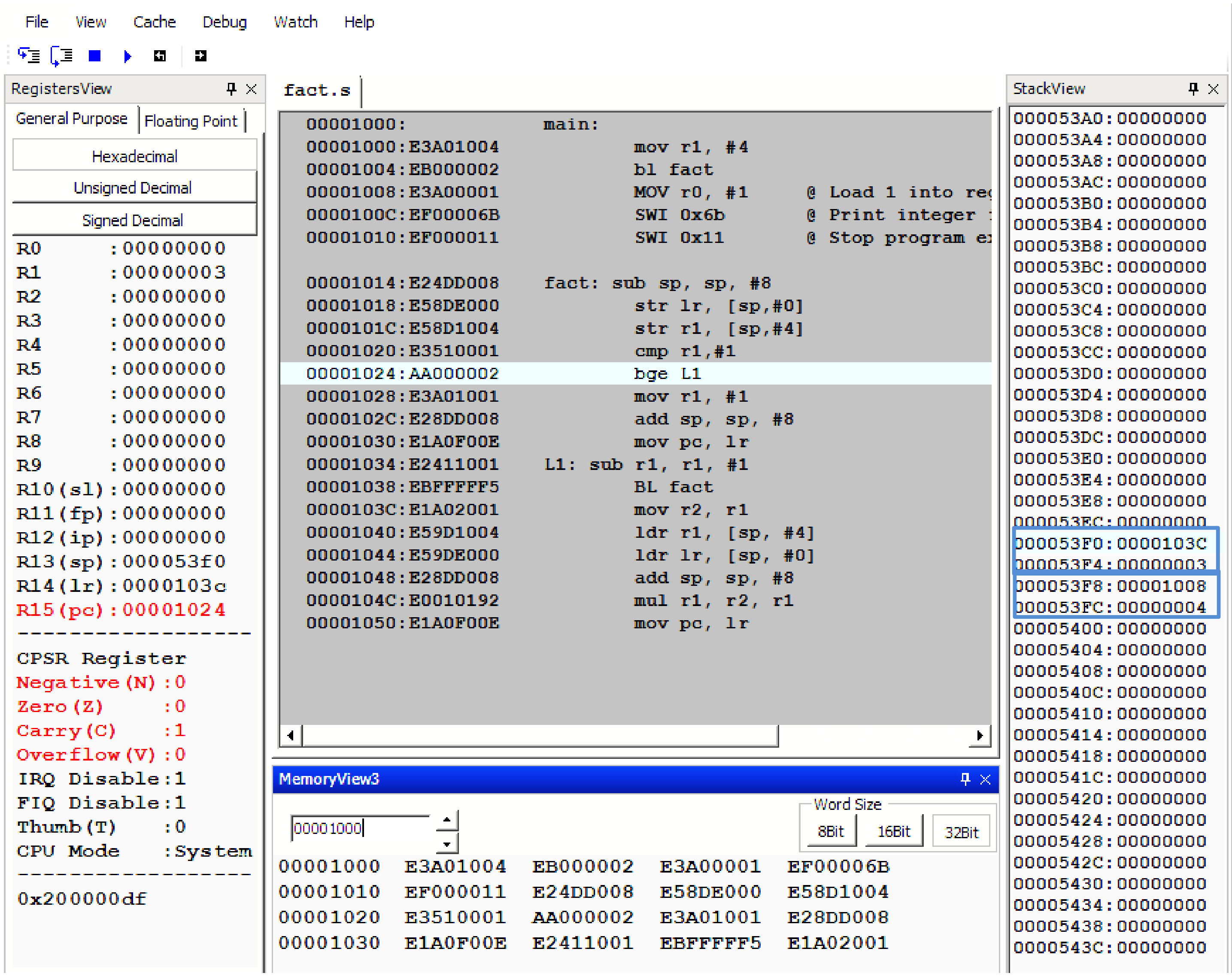

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000003
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f8
R14(lr):0000103c
R15(pc):00001014
CPSR Register
Negative(N) :0
Zero(Z) :0
Carry(C) :1
Overflow(V) :0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x200000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053EC:00000000
000053F0:00000000
000053F4:00000000
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000

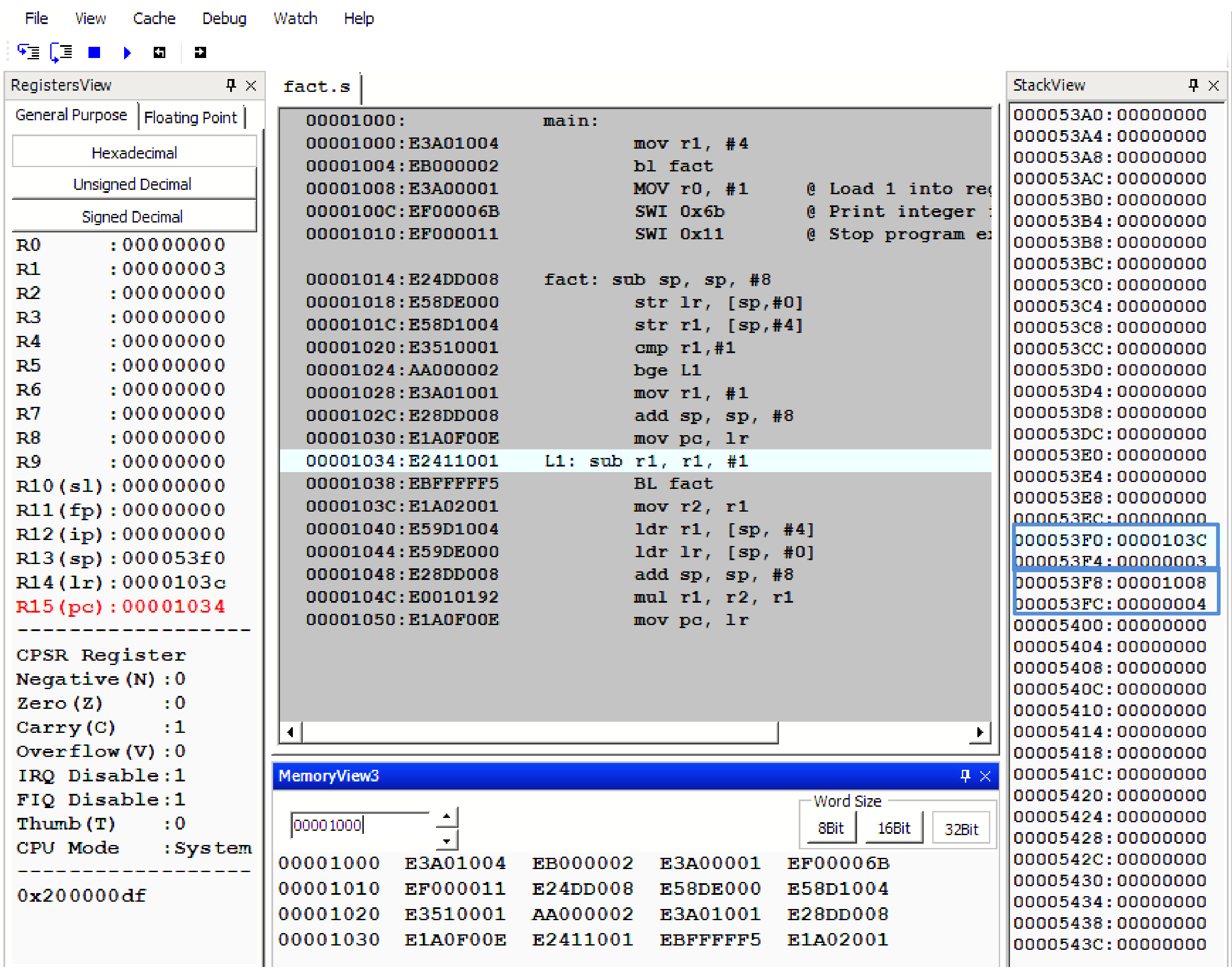

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000003
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f0
R14(lr):0000103c
R15(pc):00001018
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x200000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
Word Size
00001000
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053EC:00000000
000053F0:00000000
000053F4:00000000
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000


File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000003
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f0
R14(lr):0000103c
R15(pc):0000101c
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x200000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053F0:0000103C
000053F4:00000000
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000

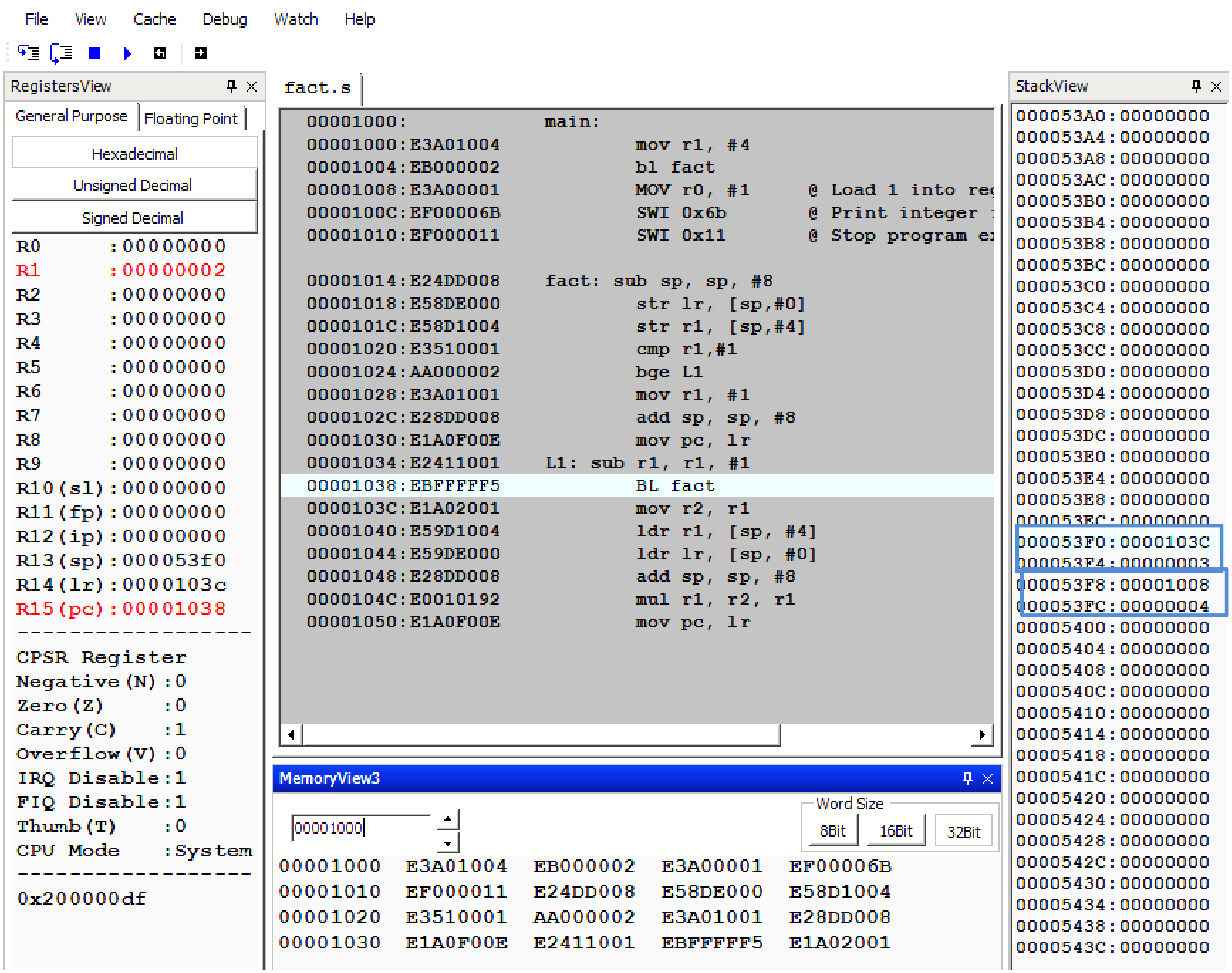


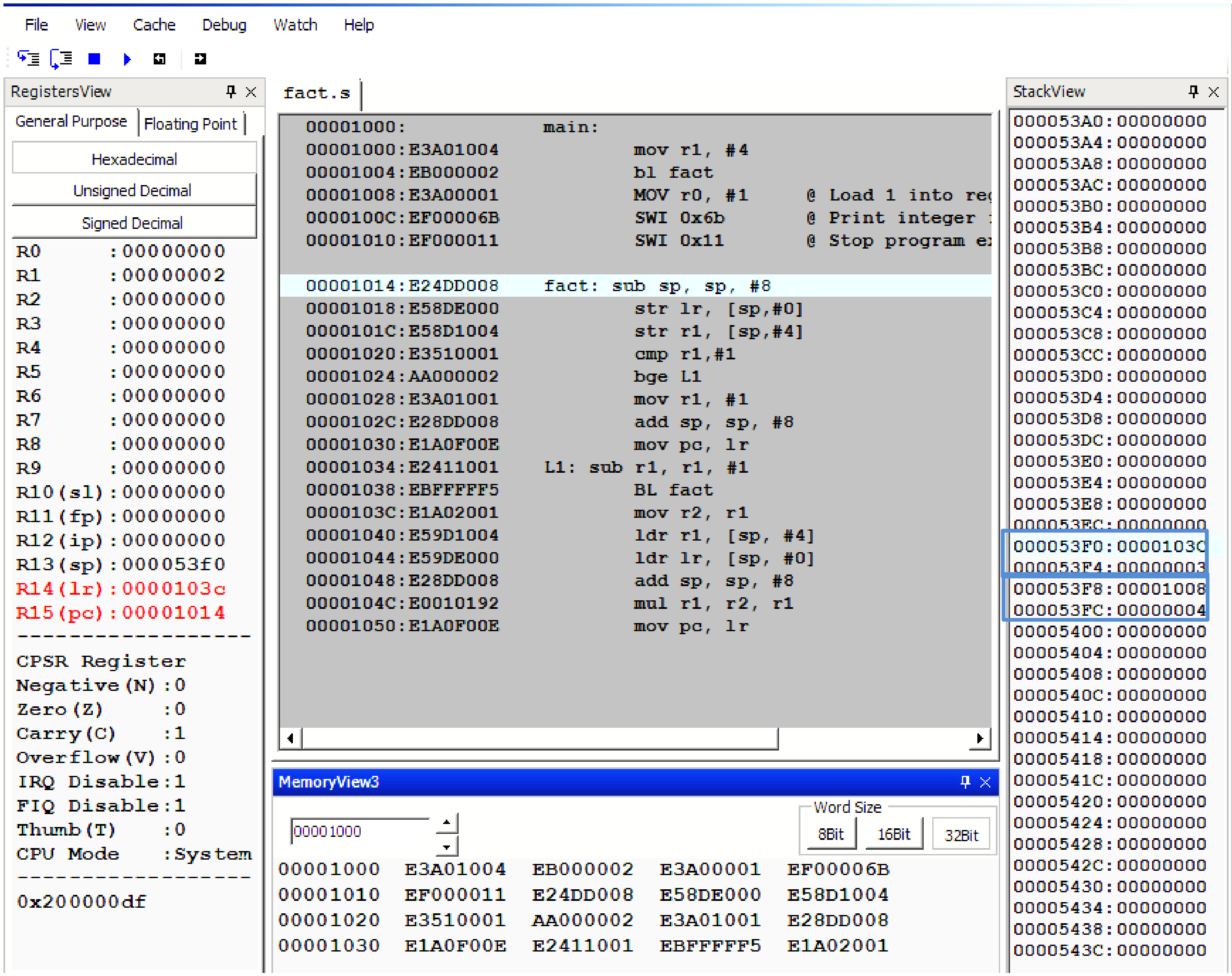
File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000003
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f0
R14(lr):0000103c
R15(pc):00001020
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x200000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into reg
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program ex
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053EC:00000000
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000

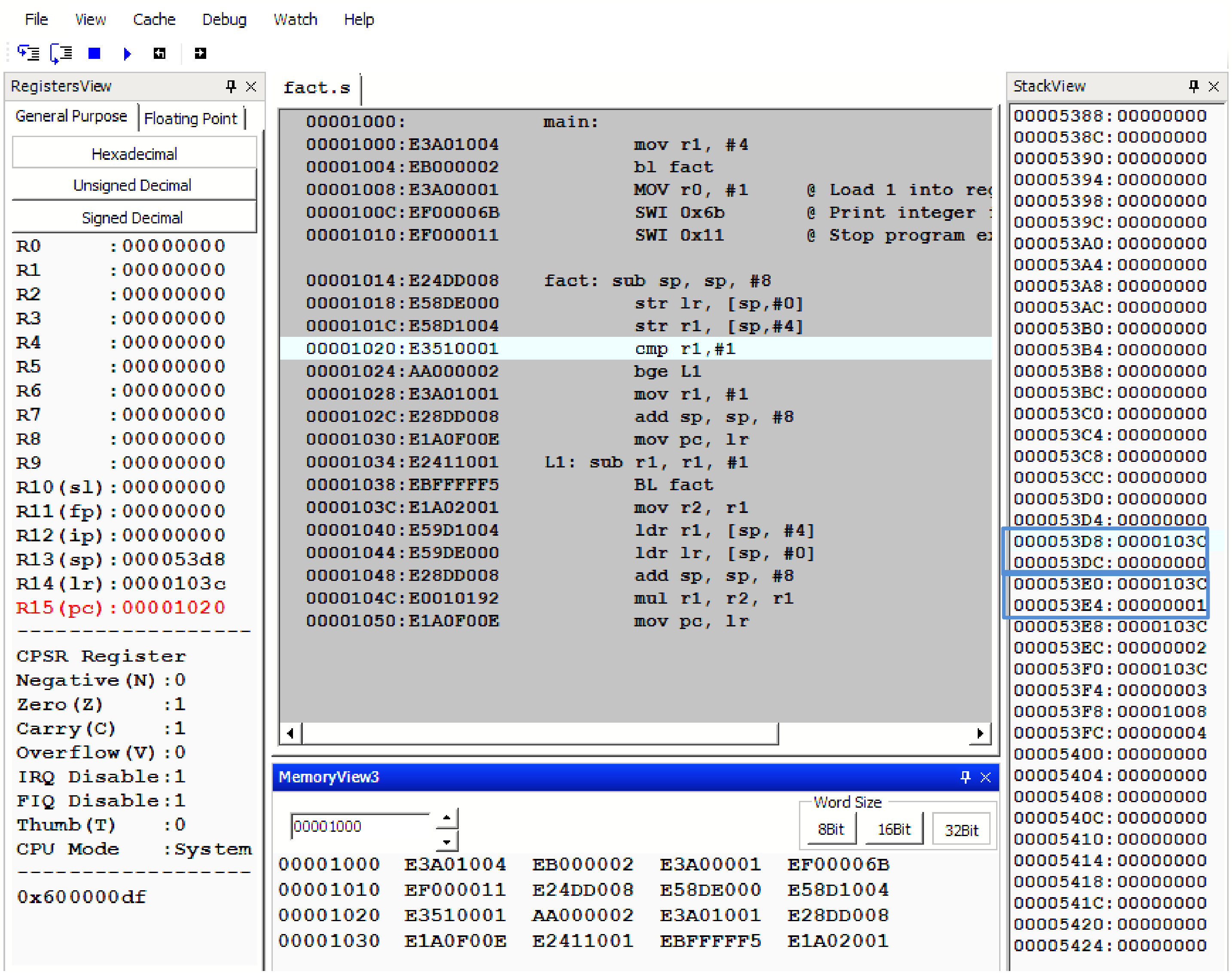

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000003
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f0
R14(lr):0000103c
R15(pc):00001024
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x200000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
Word Size
00001000
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053EC:00000000
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000



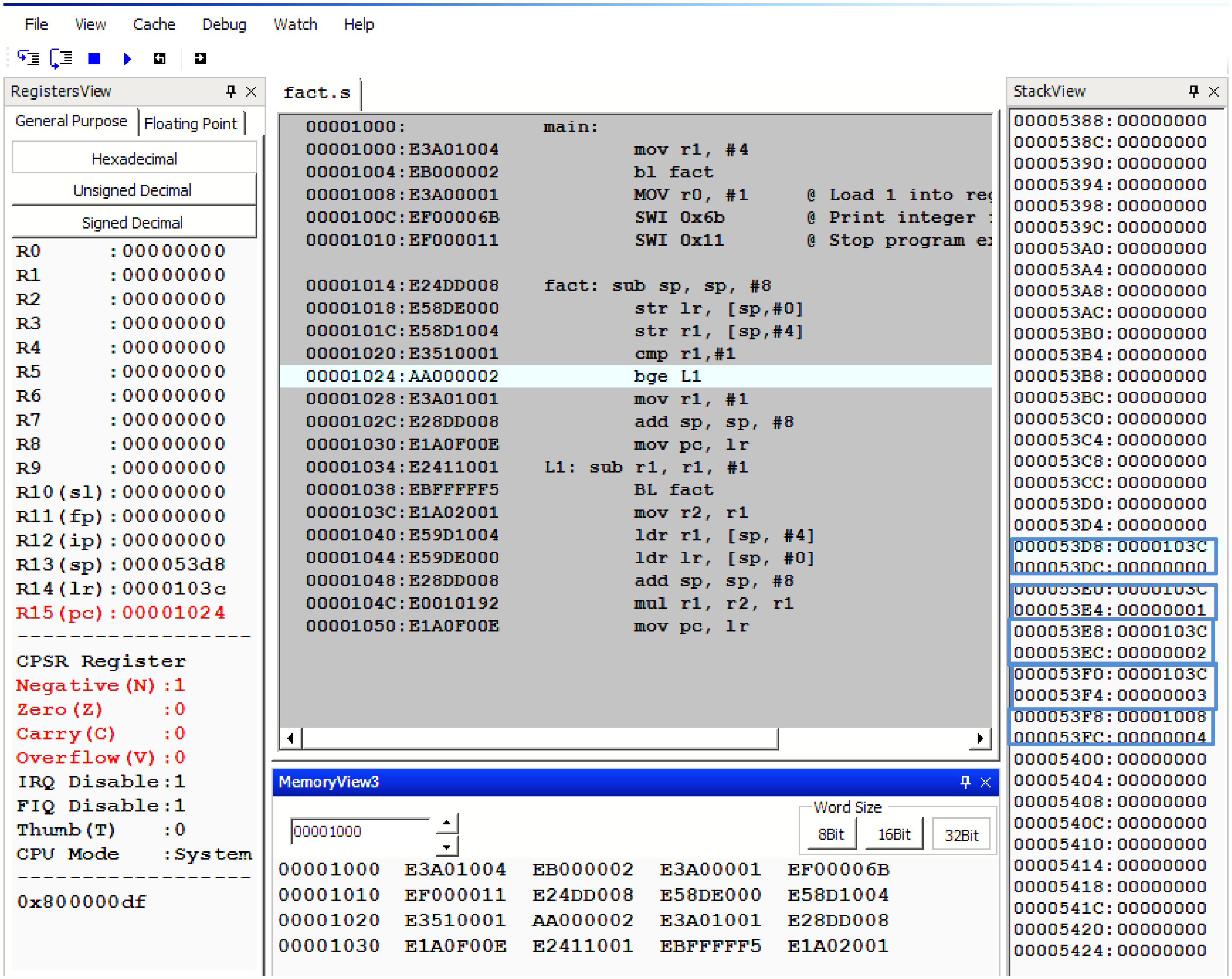
File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000003
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f0
R14(lr):0000103c
R15(pc):00001034
CPSR Register
Negative(N) :0
Zero(Z) :0
Carry(C) :1
Overflow(V) :0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x200000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053EC:00000000
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000

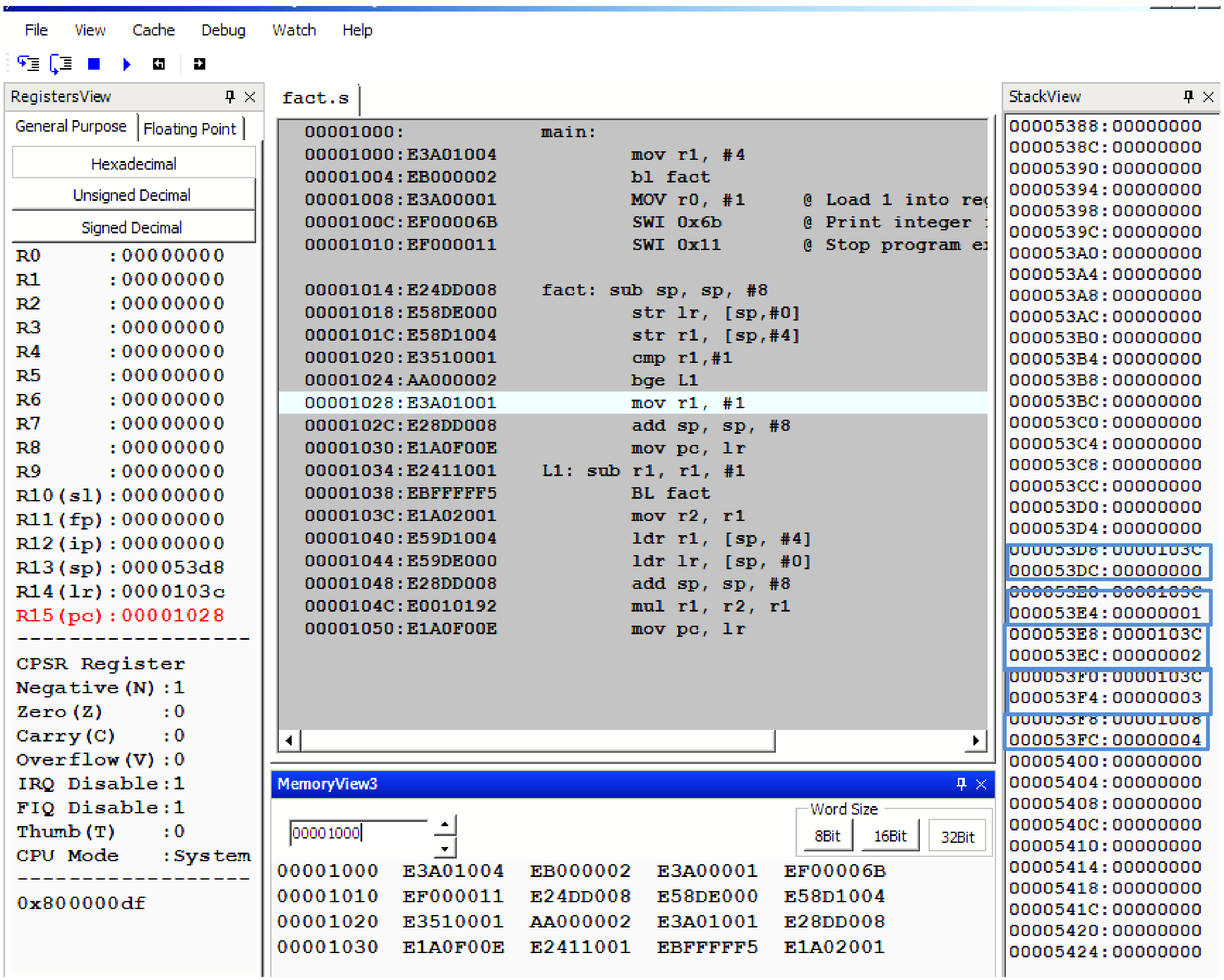

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000002
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f0
R14(lr):0000103c
R15(pc):00001038
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x200000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053EC:00000000
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000

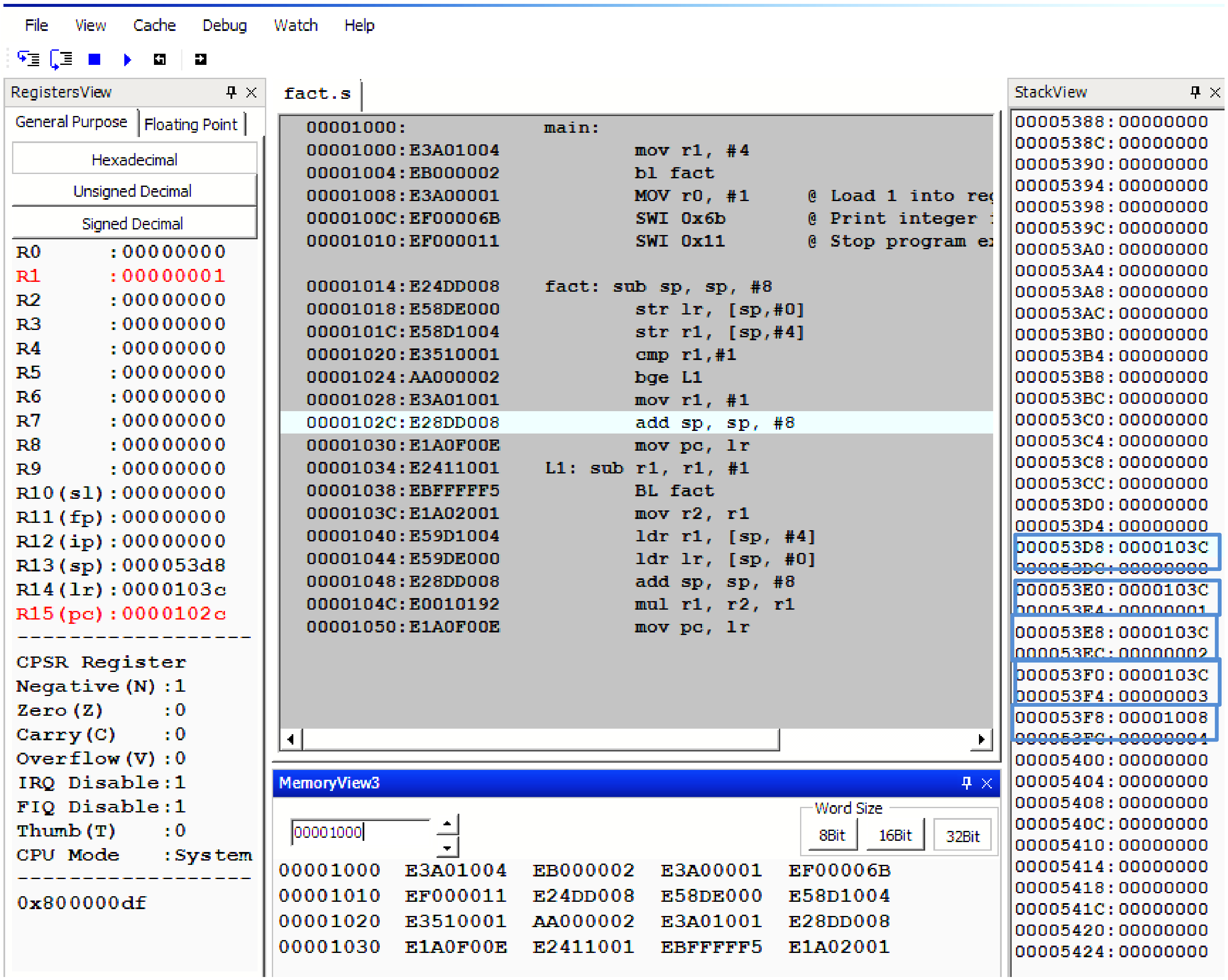

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000002
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f0
R14(lr):0000103c
R15(pc):00001014
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x200000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit
16Bit
32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:00000000
000053DC:00000000
000053E0:00000000
000053E4:00000000
000053E8:00000000
000053F0:0000103C
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000

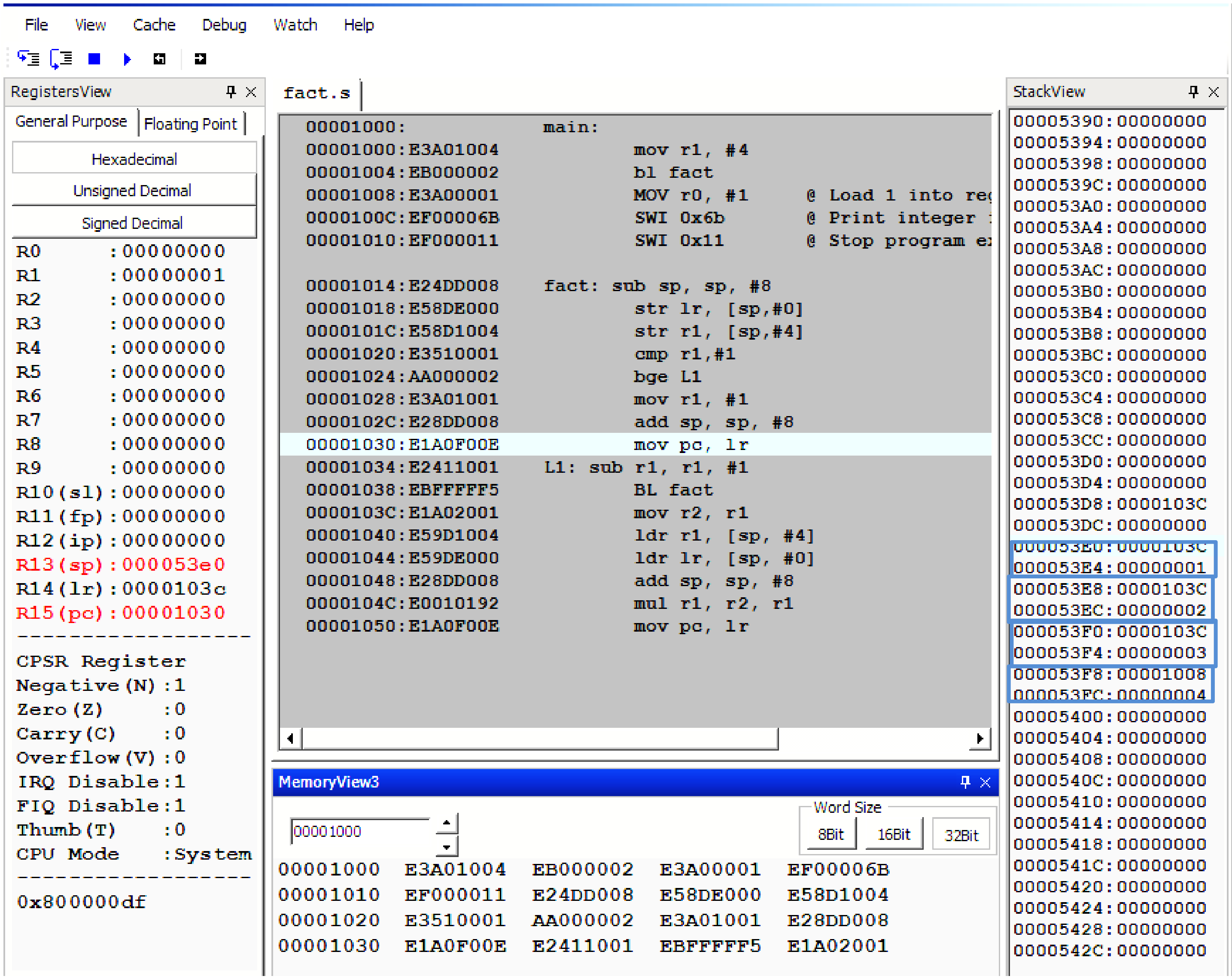


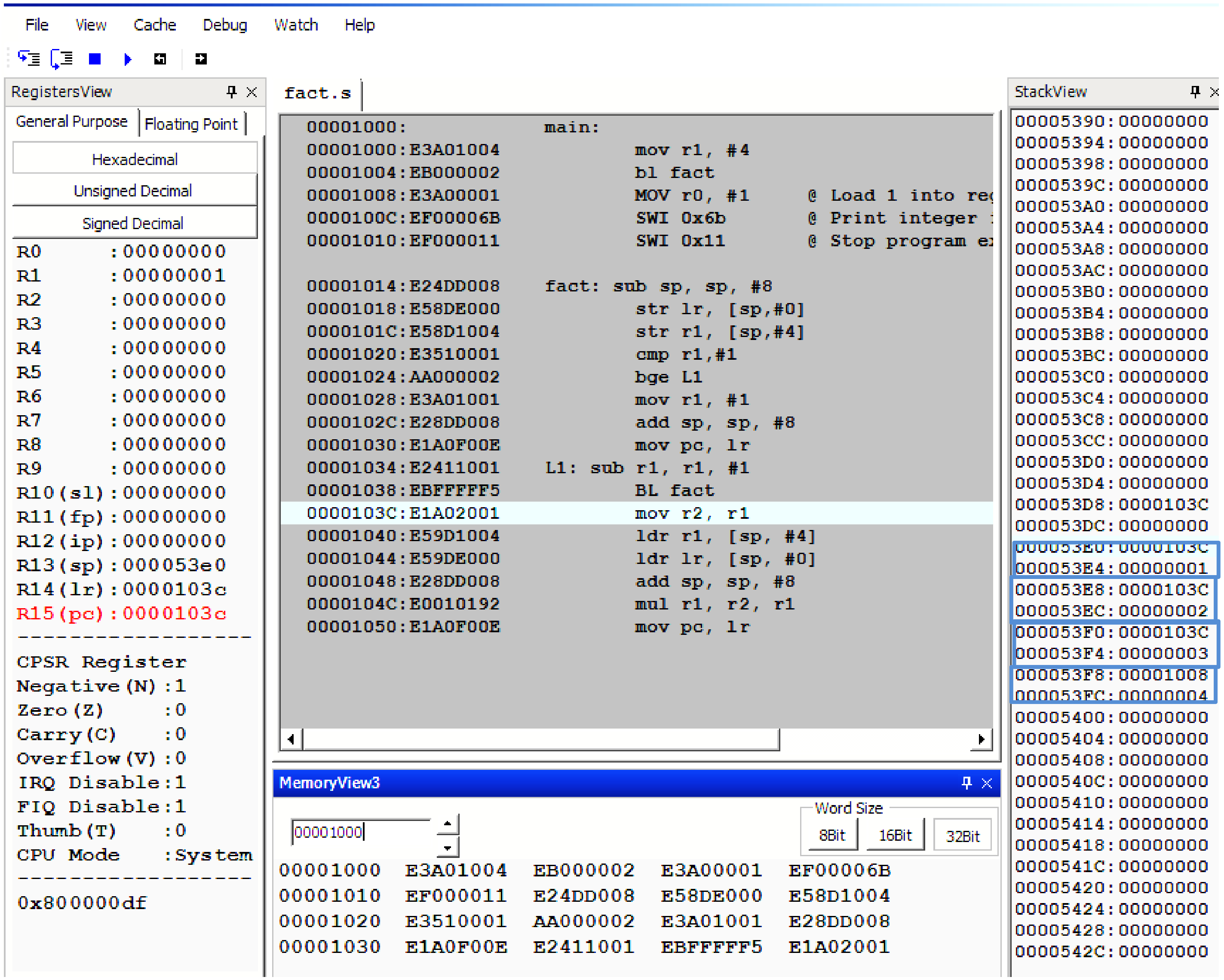
File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000000
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053d8
R14(lr):0000103c
R15(pc):00001020
CPSR Register
Negative(N):0
Zero(Z) :1
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x600000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
00005388:00000000
0000538C:00000000
00005390:00000000
00005394:00000000
00005398:00000000
0000539C:00000000
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000

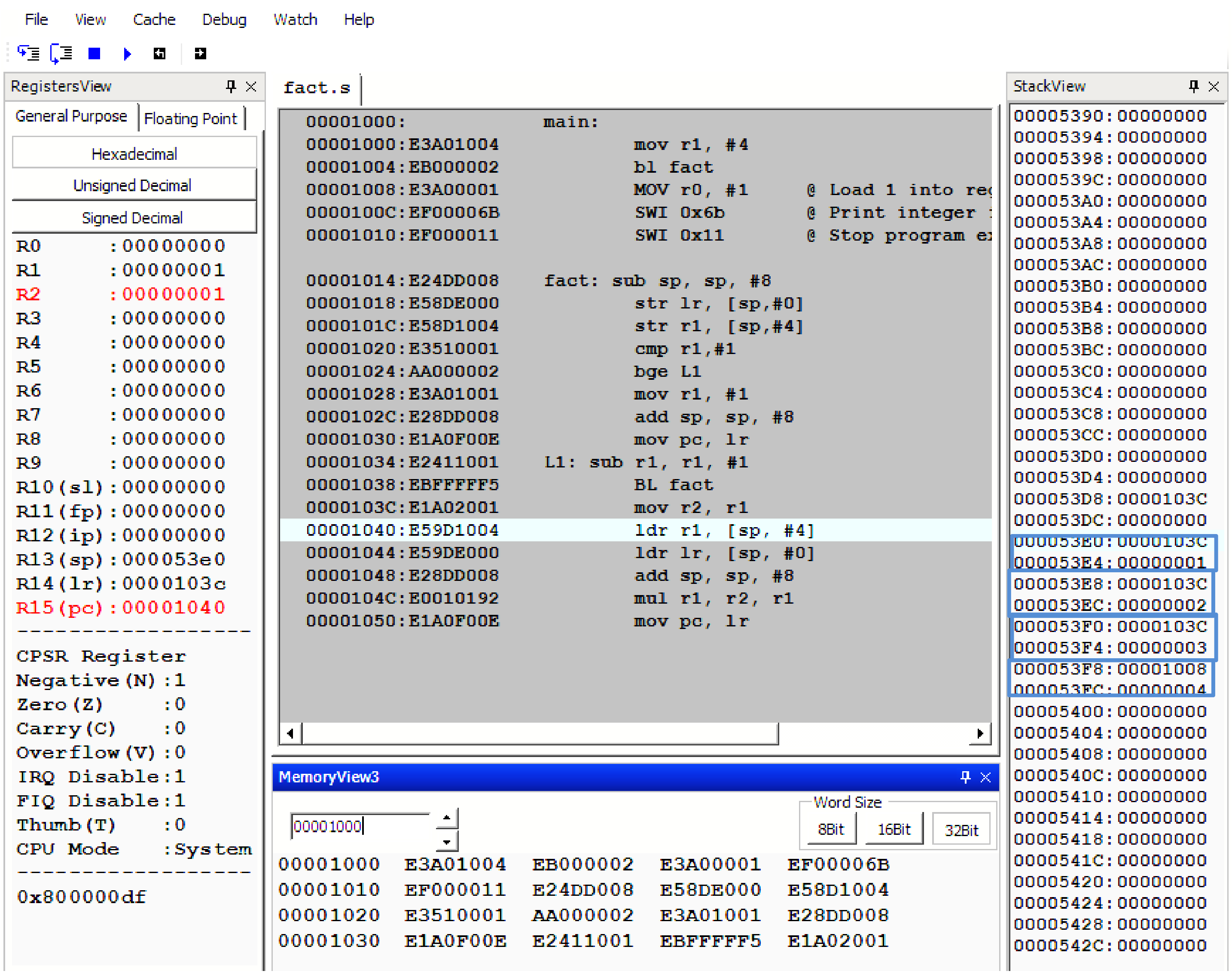

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000000
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053d8
R14(lr):0000103c
R15(pc):00001024
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
00005388:00000000
0000538C:00000000
00005390:00000000
00005394:00000000
00005398:00000000
0000539C:00000000
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000

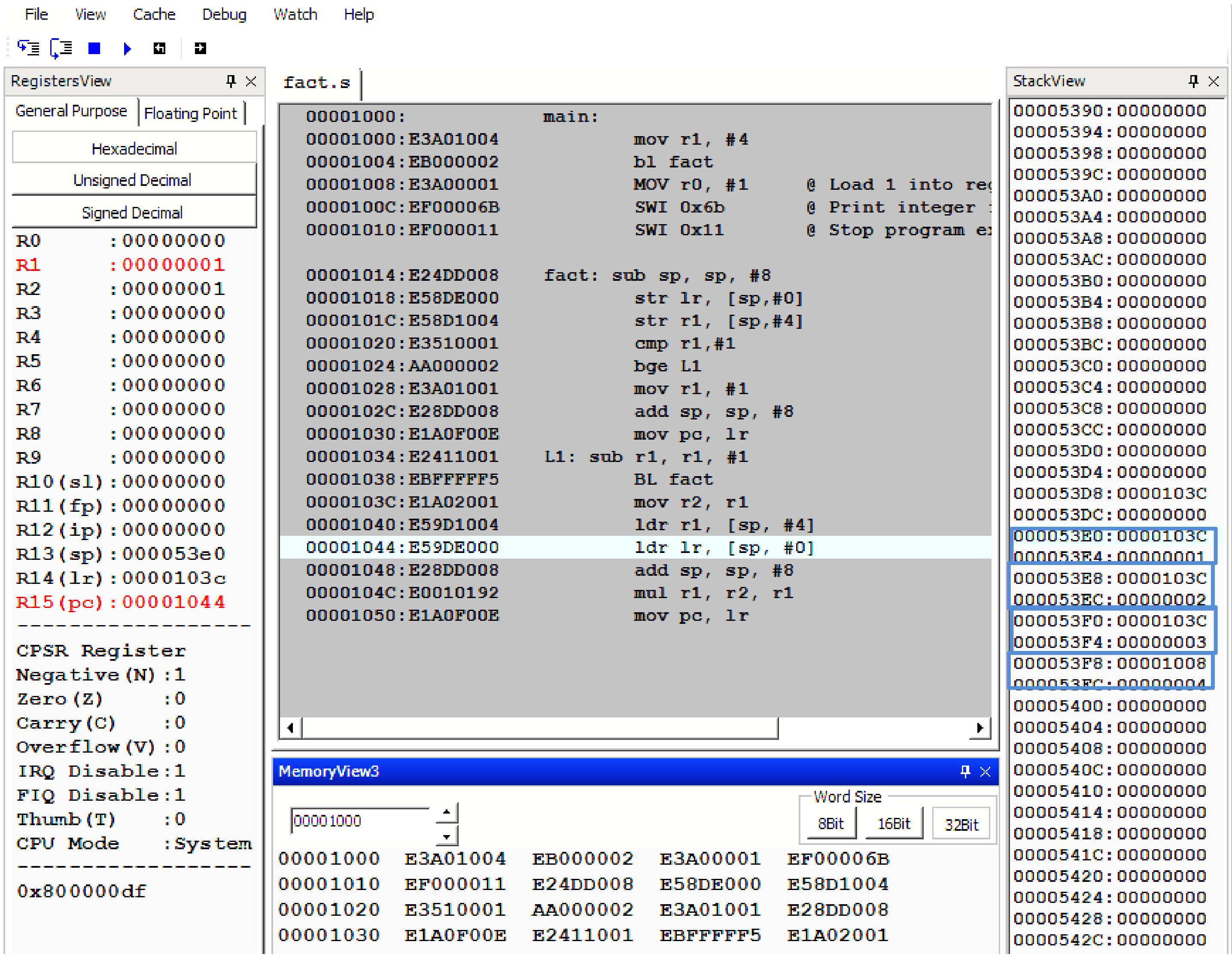

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000000
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053d8
R14(lr):0000103c
R15(pc):00001028
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
00005388:00000000
0000538C:00000000
00005390:00000000
00005394:00000000
00005398:00000000
0000539C:00000000
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053DC:00000000
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F4:00000003
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000

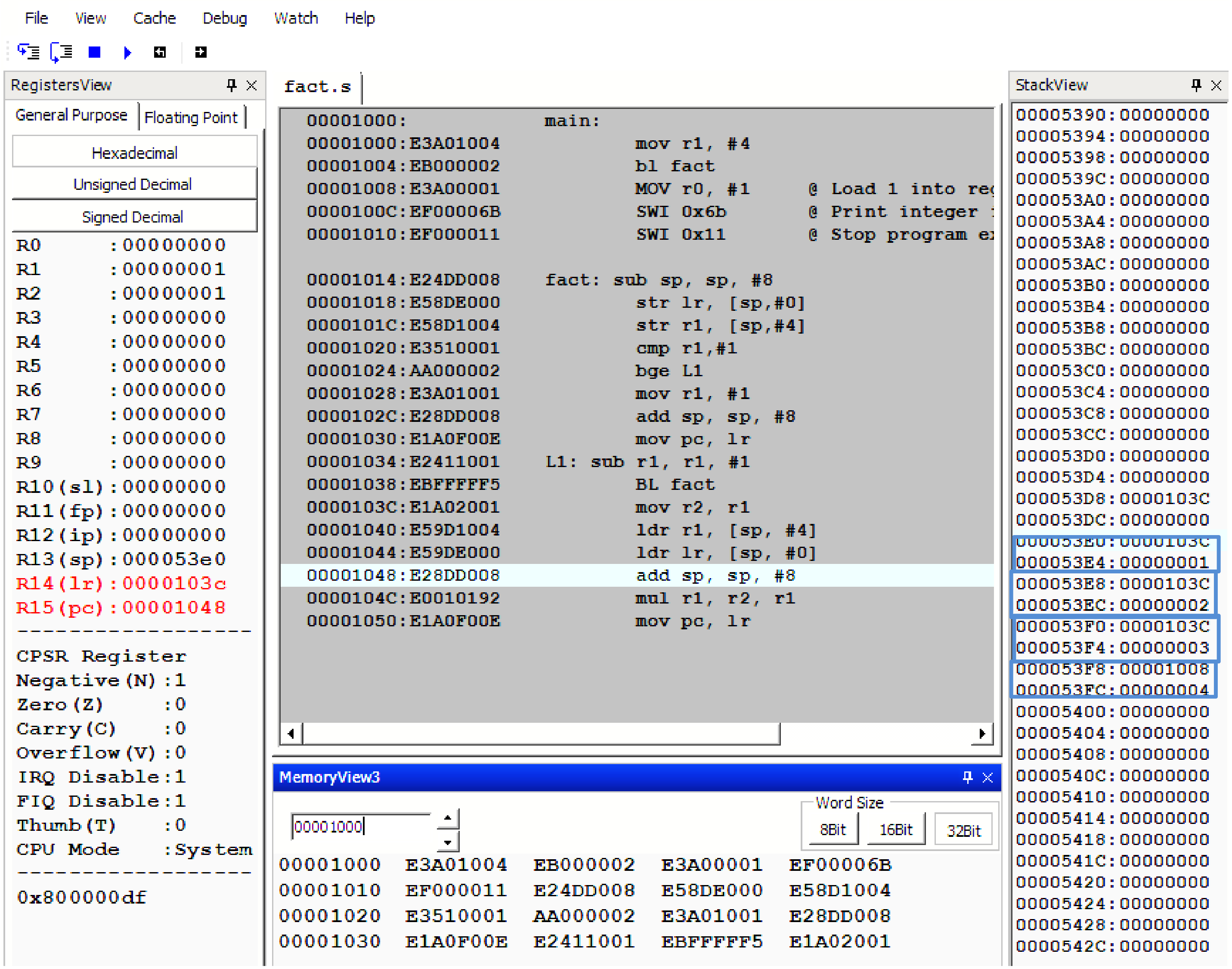

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000001
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053d8
R14(lr):0000103c
R15(pc):0000102c
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
Word Size
00001000
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
00005388:00000000
0000538C:00000000
00005390:00000000
00005394:00000000
00005398:00000000
0000539C:00000000
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000

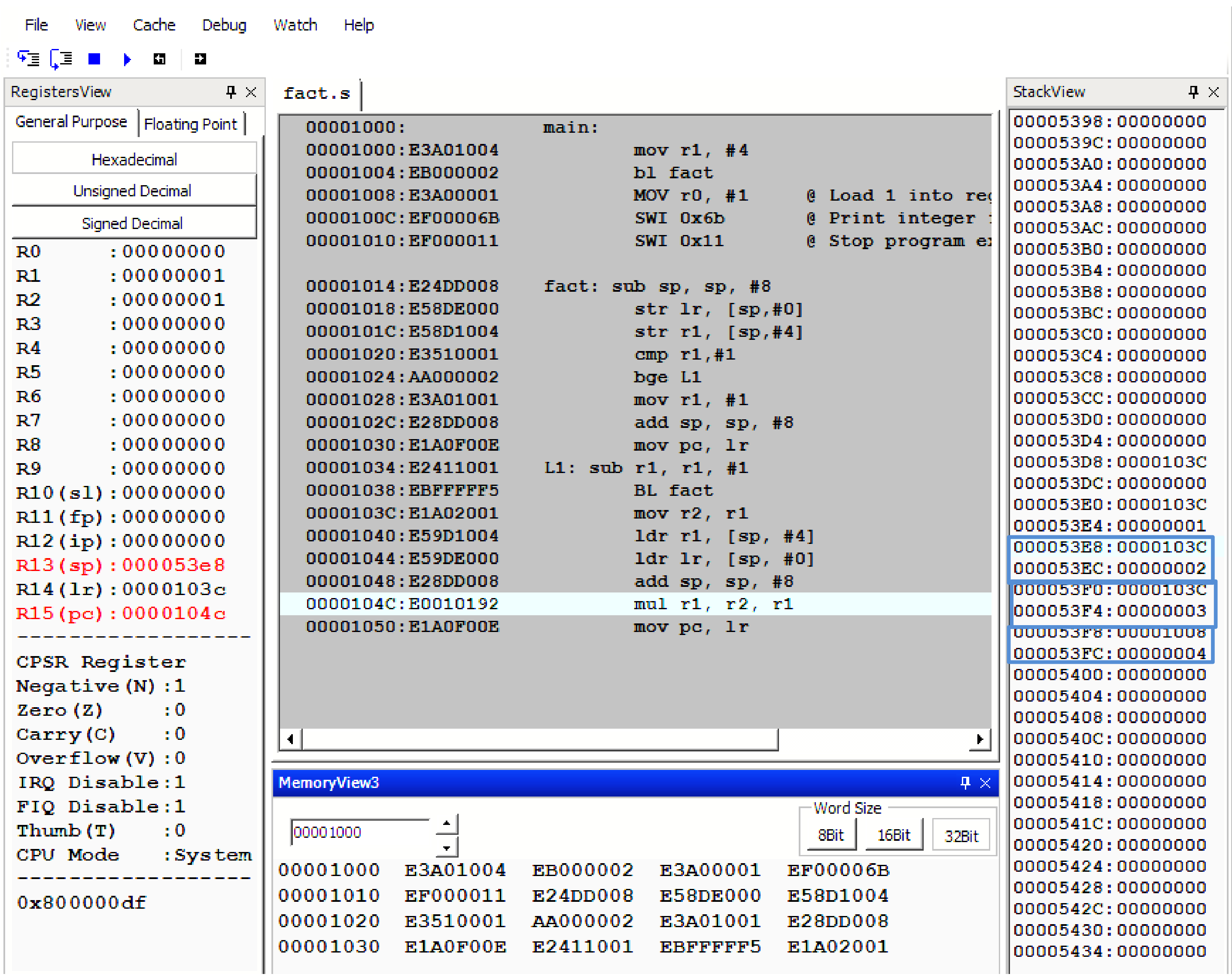

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000001
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053e0
R14(lr):0000103c
R15(pc):00001030
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
Word Size
00001000
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
00005390:00000000
00005394:00000000
00005398:00000000
0000539C:00000000
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000

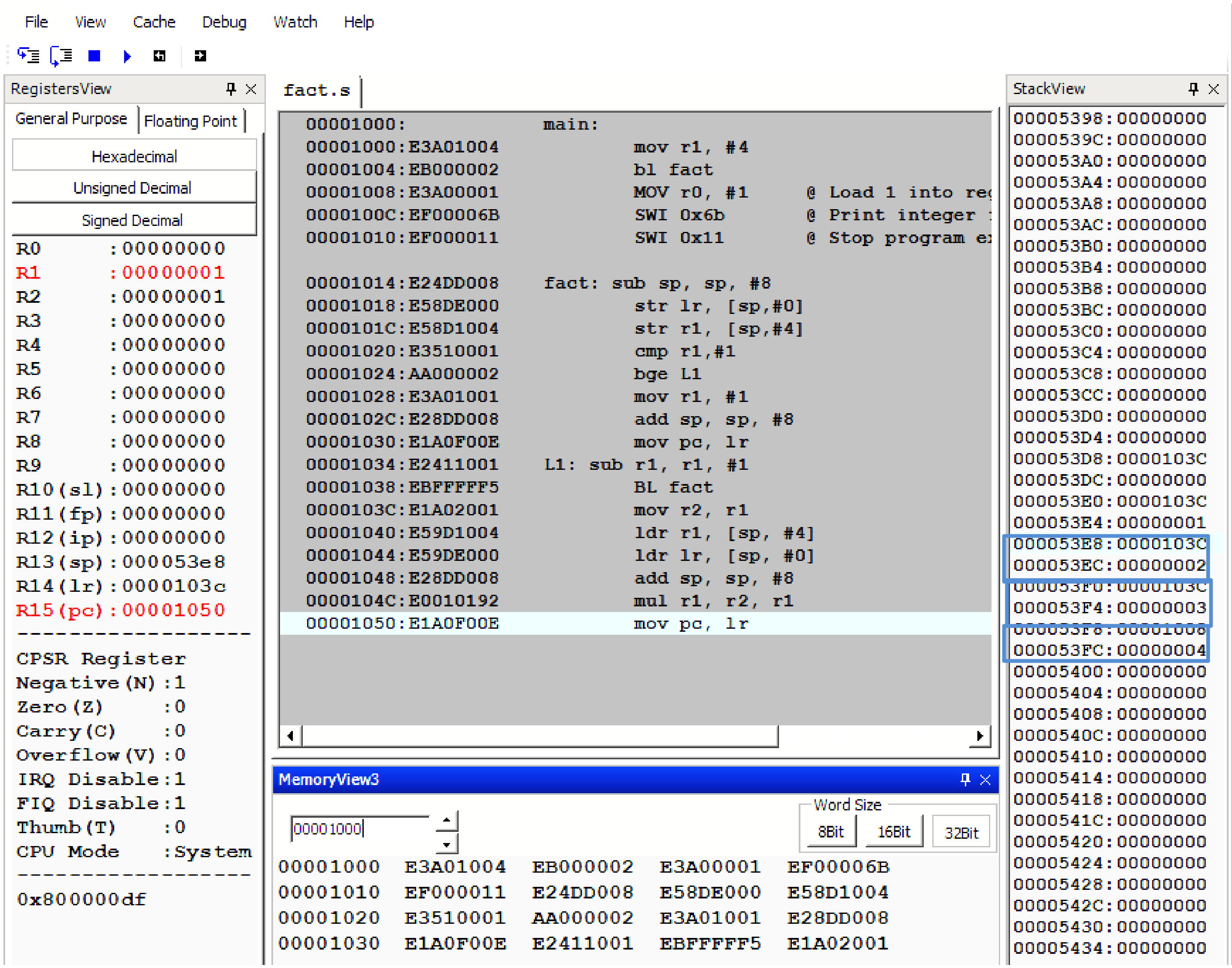

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000001
R2 :00000000
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053e0
R14(lr):0000103c
R15(pc):0000103c
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
00005390:00000000
00005394:00000000
00005398:00000000
0000539C:00000000
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000

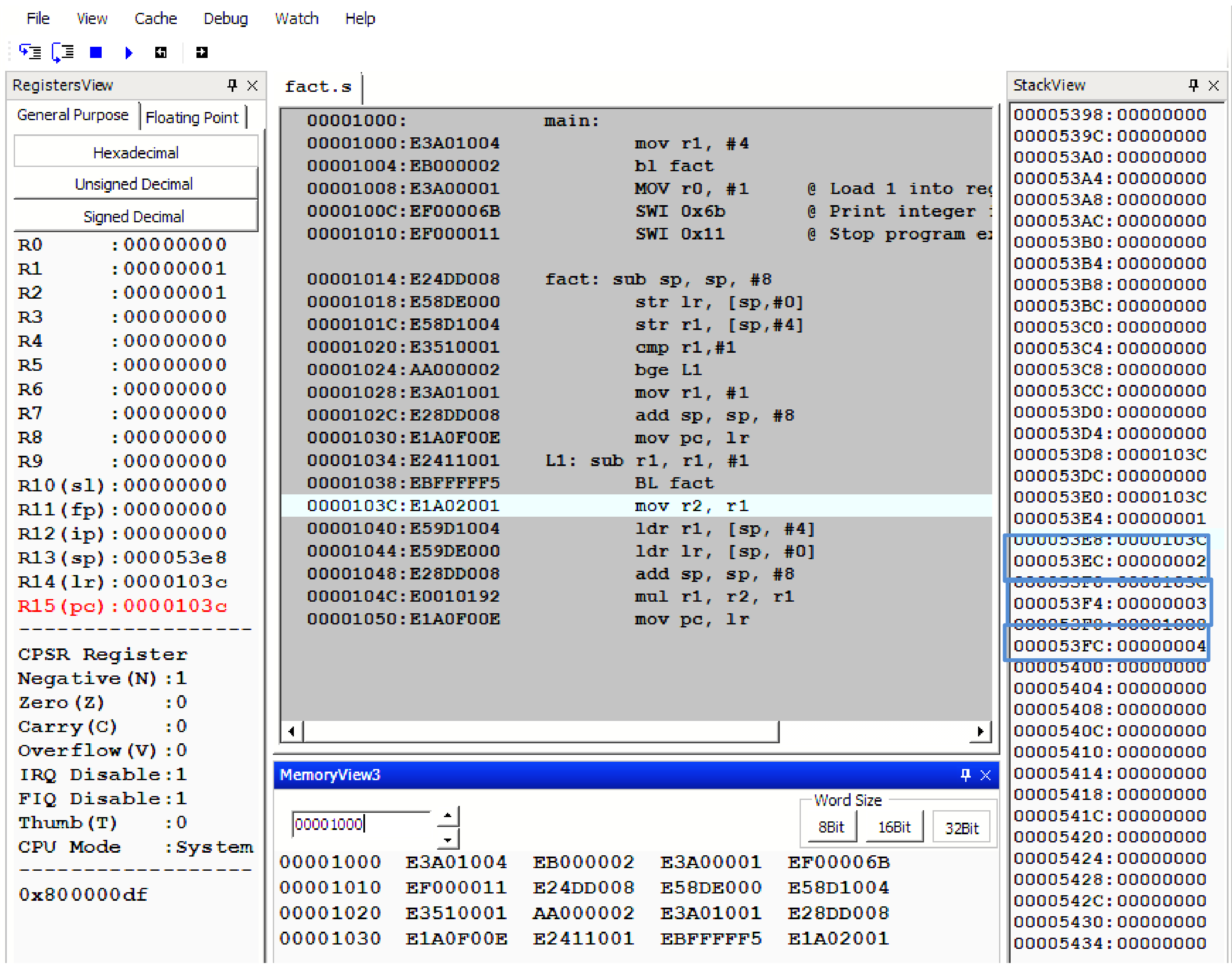

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000001
R2 :00000001
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053e0
R14(lr):0000103c
R15(pc):00001040
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
Word Size
8Bit 16Bit 32Bit
00001000
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
00005390:00000000
00005394:00000000
00005398:00000000
0000539C:00000000
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000

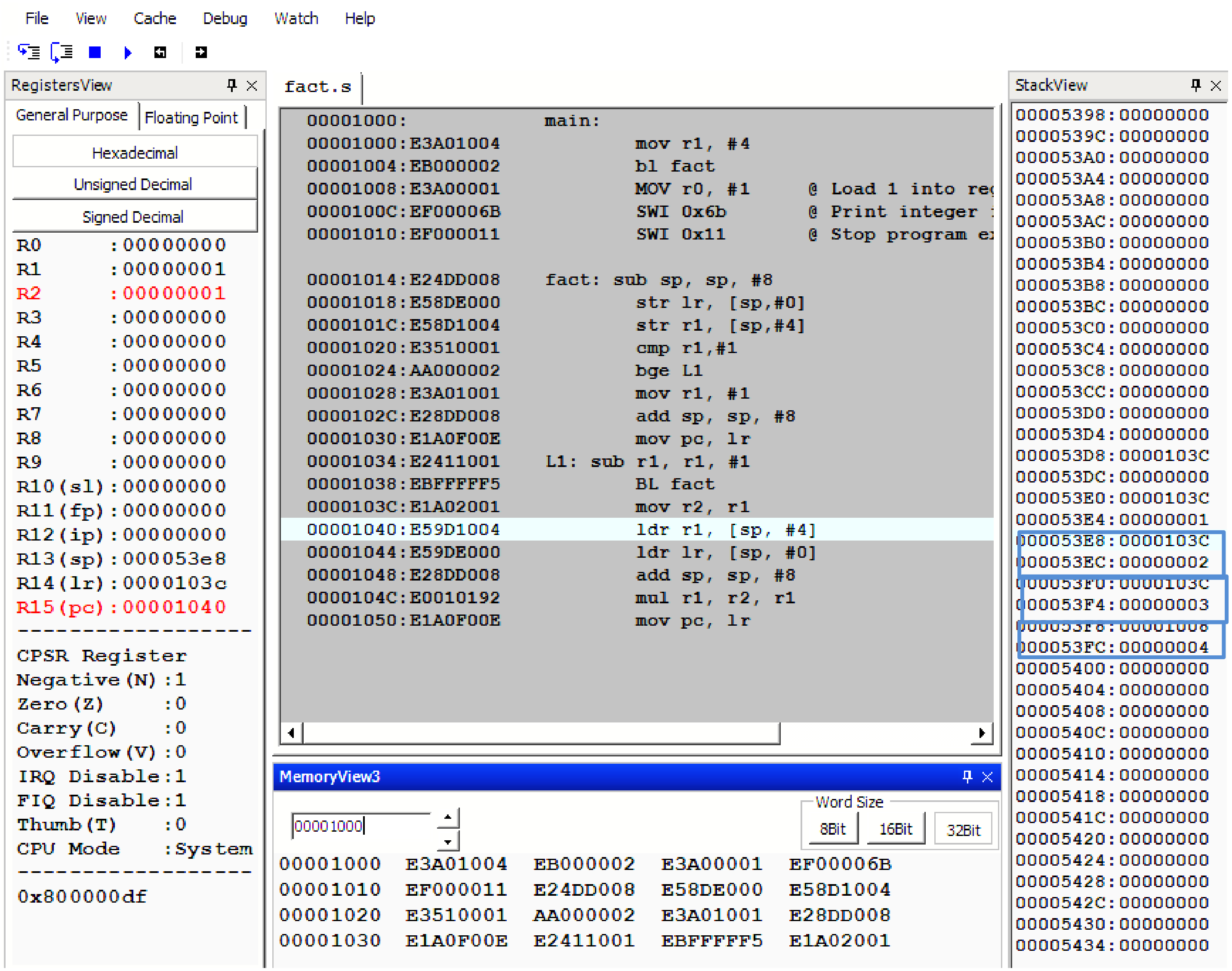

File
View
Cache
Debug
Watch
Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000001
R2 :00000001
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053e0
R14(lr):0000103c
R15(pc):00001044
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit
16Bit
32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
00005390:00000000
00005394:00000000
00005398:00000000
0000539C:00000000
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000

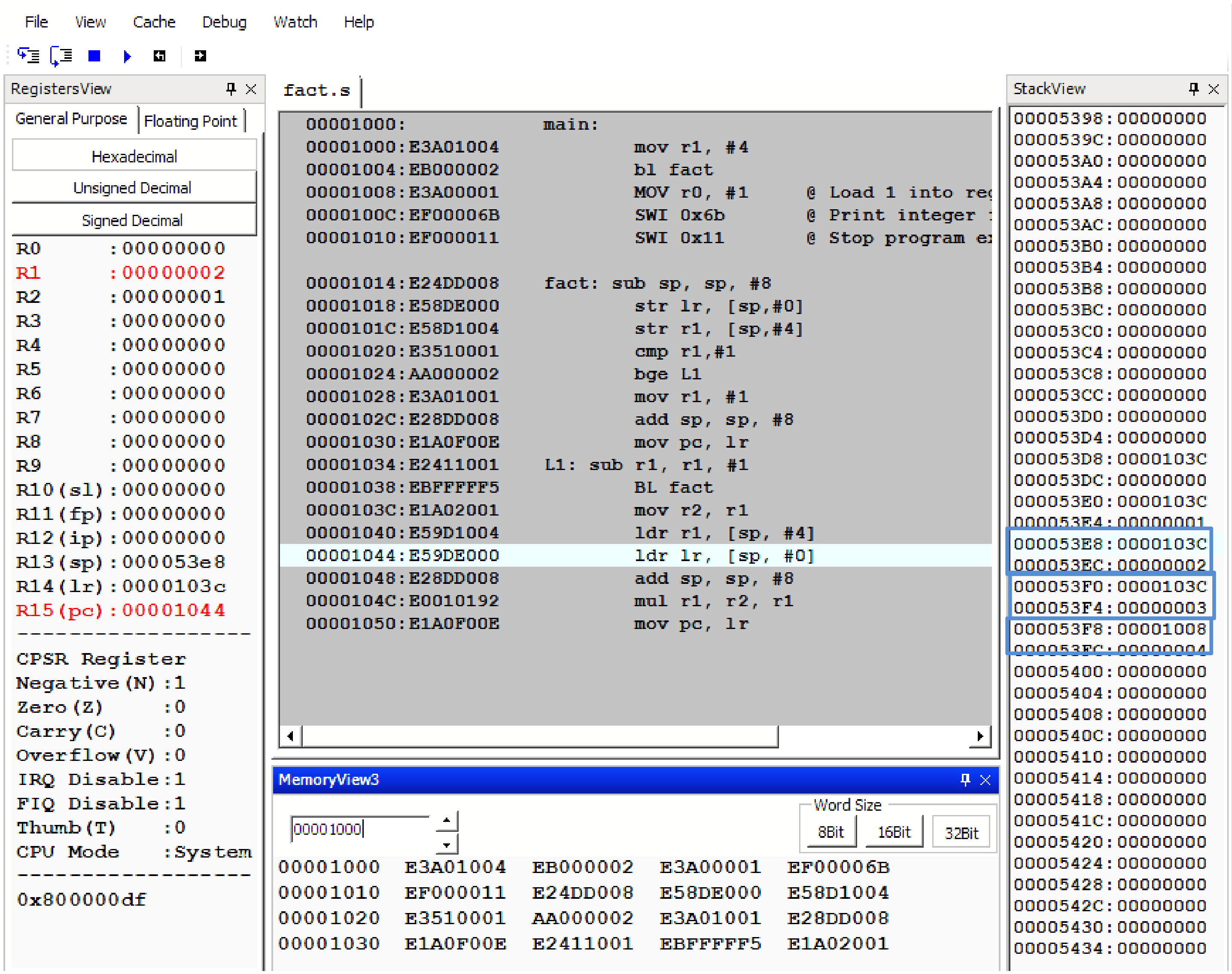

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000001
R2 :00000001
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053e0
R14(lr):0000103c
R15(pc):00001048
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
Word Size
00001000
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
00005390:00000000
00005394:00000000
00005398:00000000
0000539C:00000000
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000

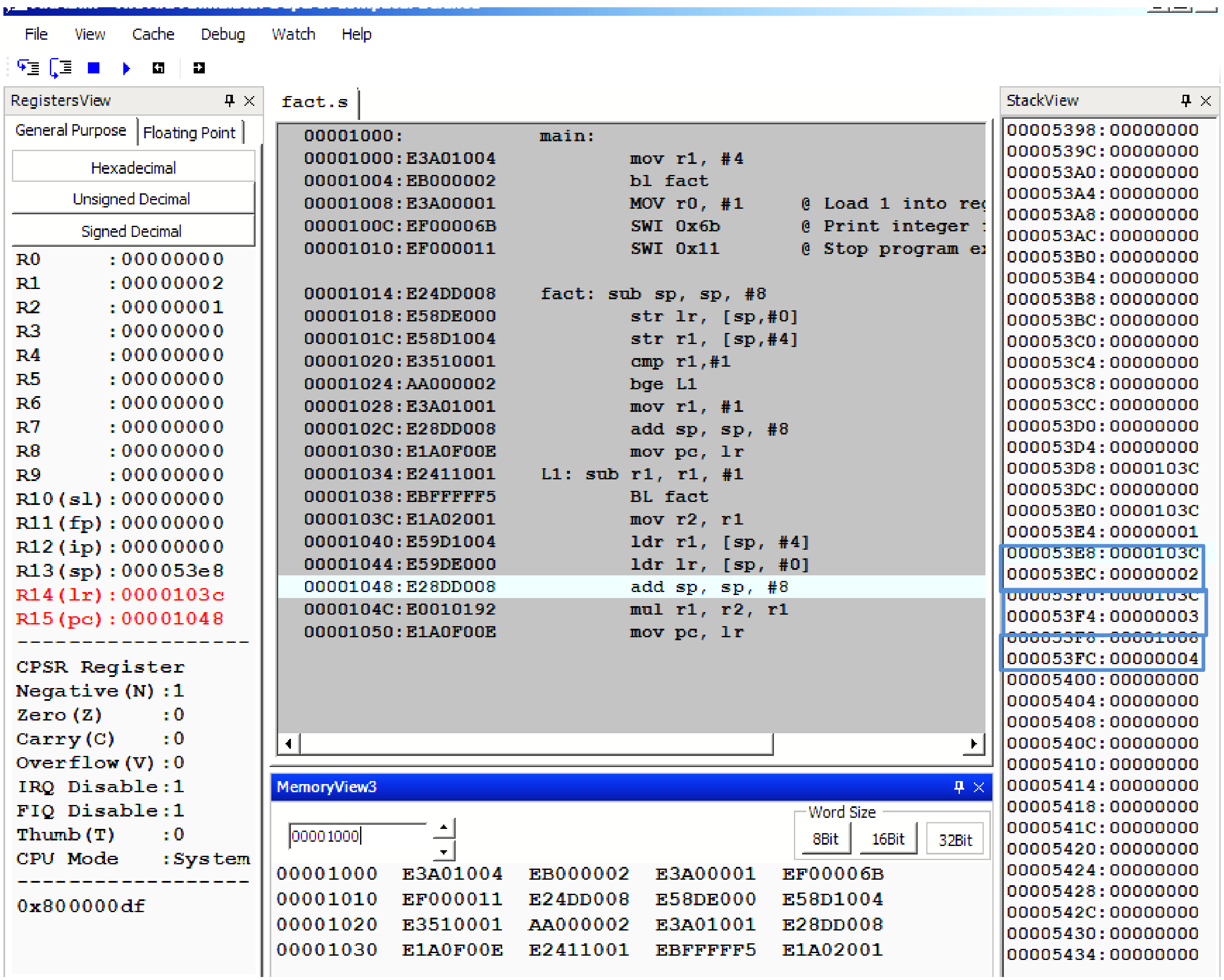

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000001
R2 :00000001
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053e8
R14(lr):0000103c
R15(pc):0000104c
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
00005398:00000000
0000539C:00000000
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000

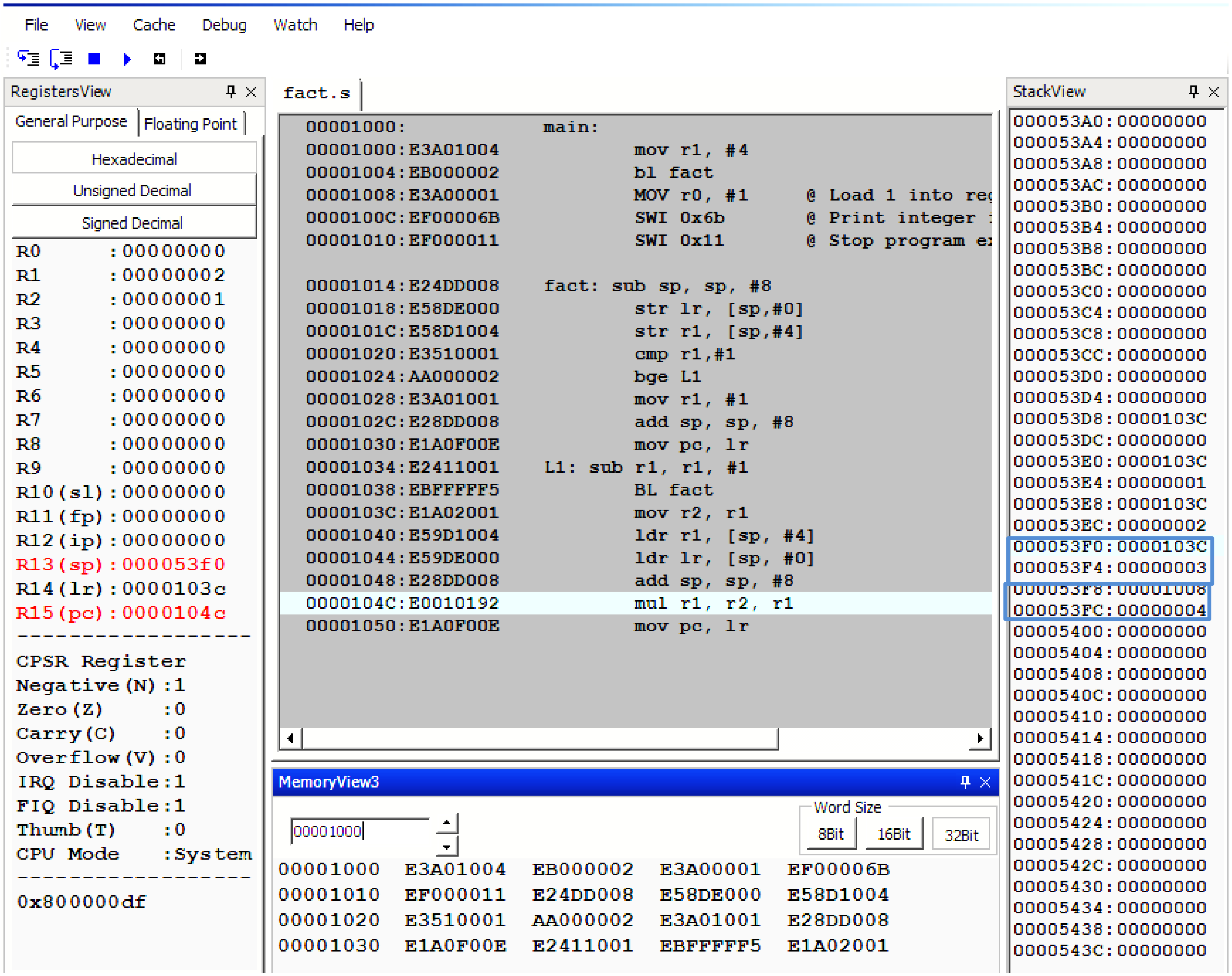

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000001
R2 :00000001
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053e8
R14(lr):0000103c
R15(pc):00001050
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
Word Size
00001000
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
00005398:00000000
0000539C:00000000
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000

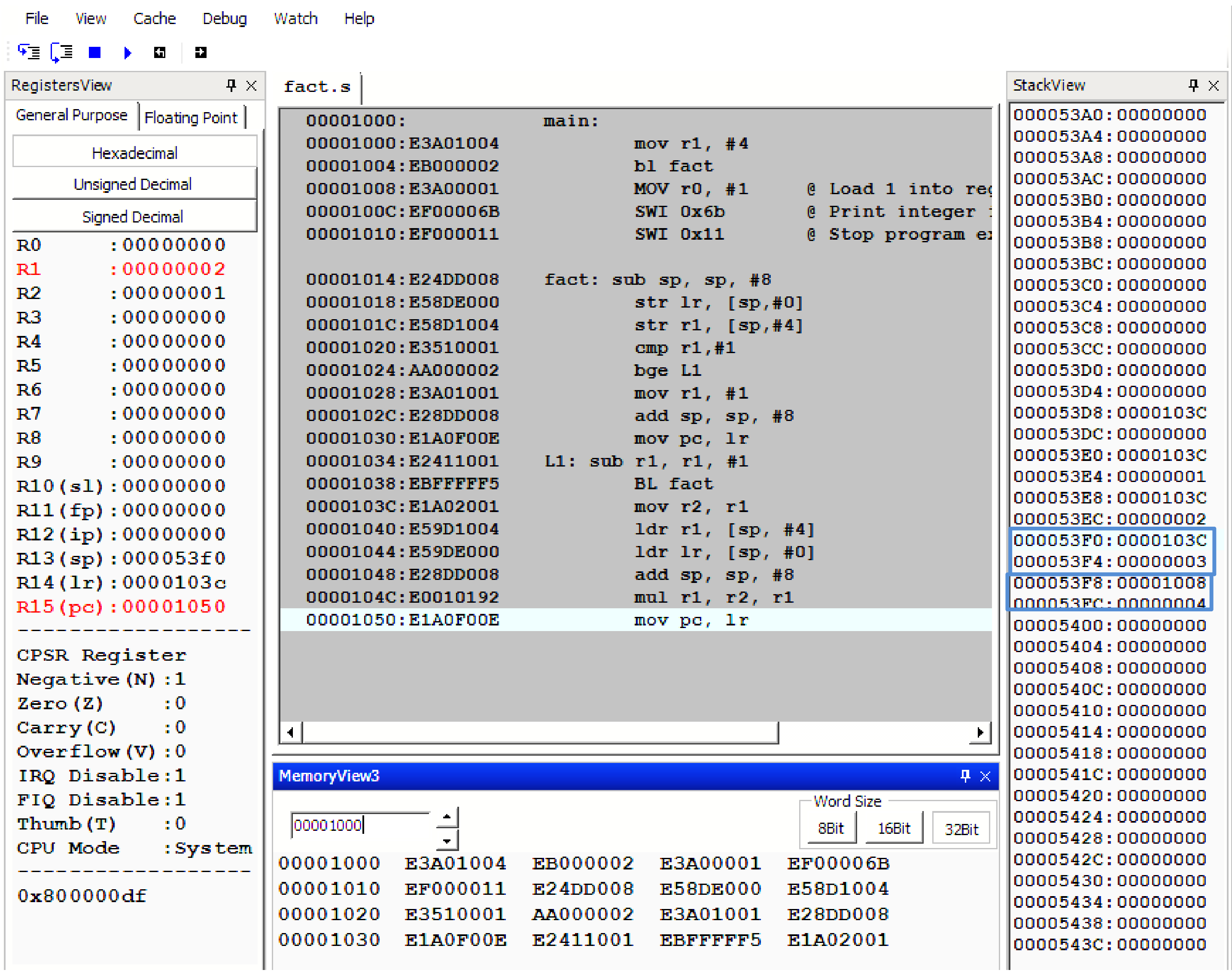

File
View
Cache
Debug
Watch
Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000001
R2 :00000001
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053e8
R14(lr):0000103c
R15(pc):0000103c
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit
16Bit
32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
00005398:00000000
0000539C:00000000
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053EC:00000002
000053F4:00000003
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000


File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000001
R2 :00000001
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053e8
R14(lr):0000103c
R15(pc):00001040
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
00005398:00000000
0000539C:00000000
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000

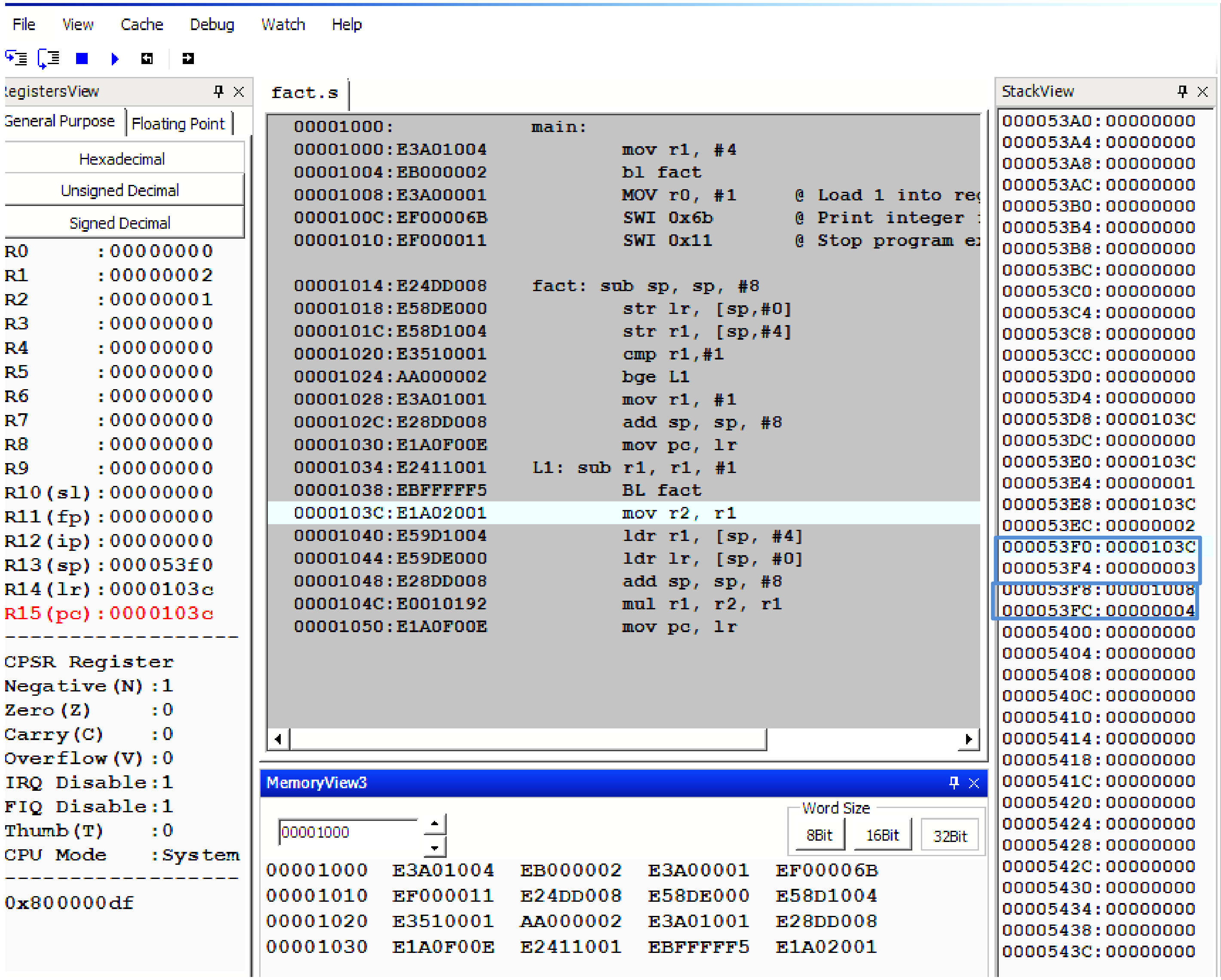

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000002
R2 :00000001
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053e8
R14(lr):0000103c
R15(pc):00001044
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into reg
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program ex
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
00005398:00000000
0000539C:00000000
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000


File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000002
R2 :00000001
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053e8
R14(lr):0000103c
R15(pc):00001048
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
00005398:00000000
0000539C:00000000
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F4:00000003
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000

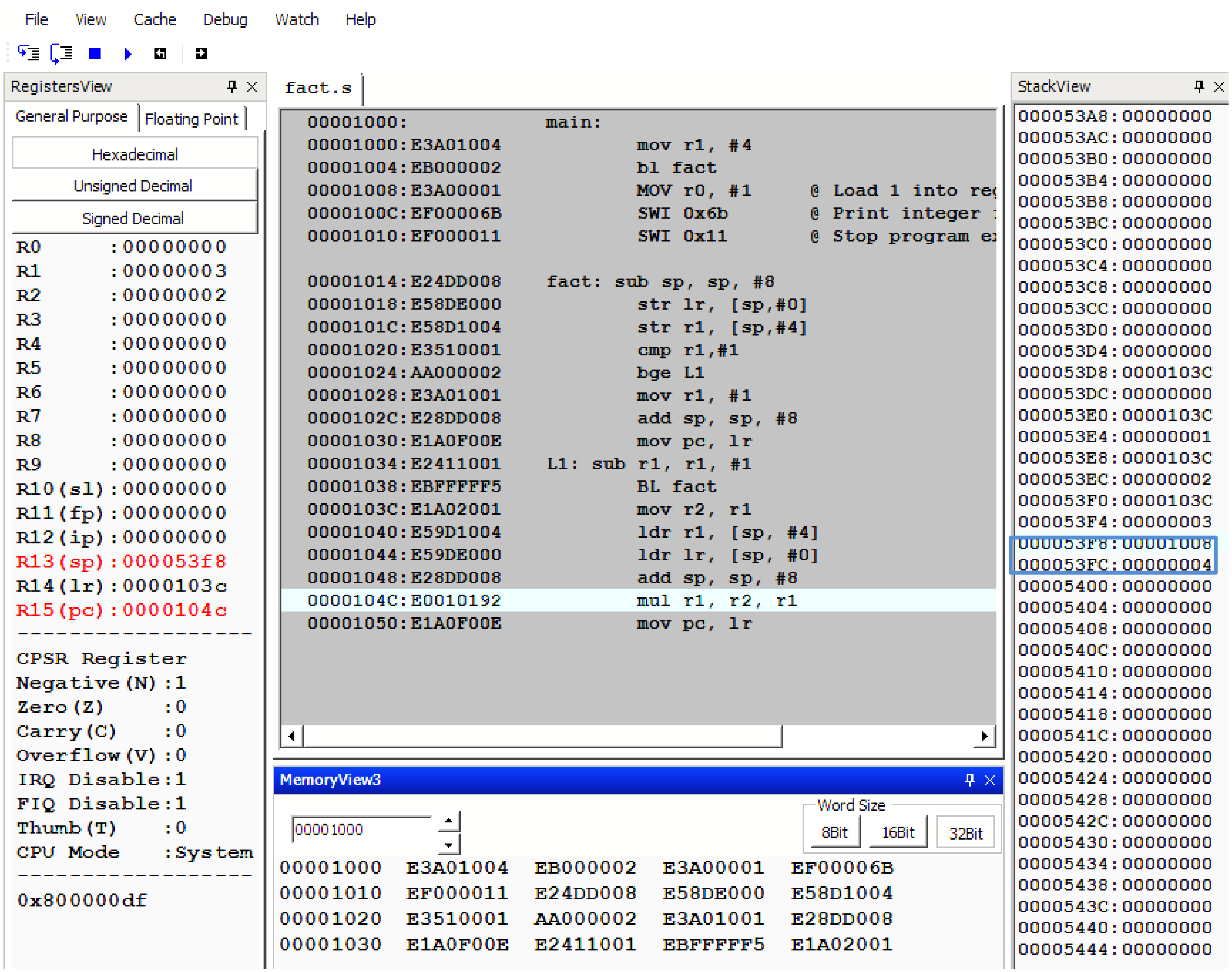

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000002
R2 :00000001
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f0
R14(lr):0000103c
R15(pc):0000104c
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000

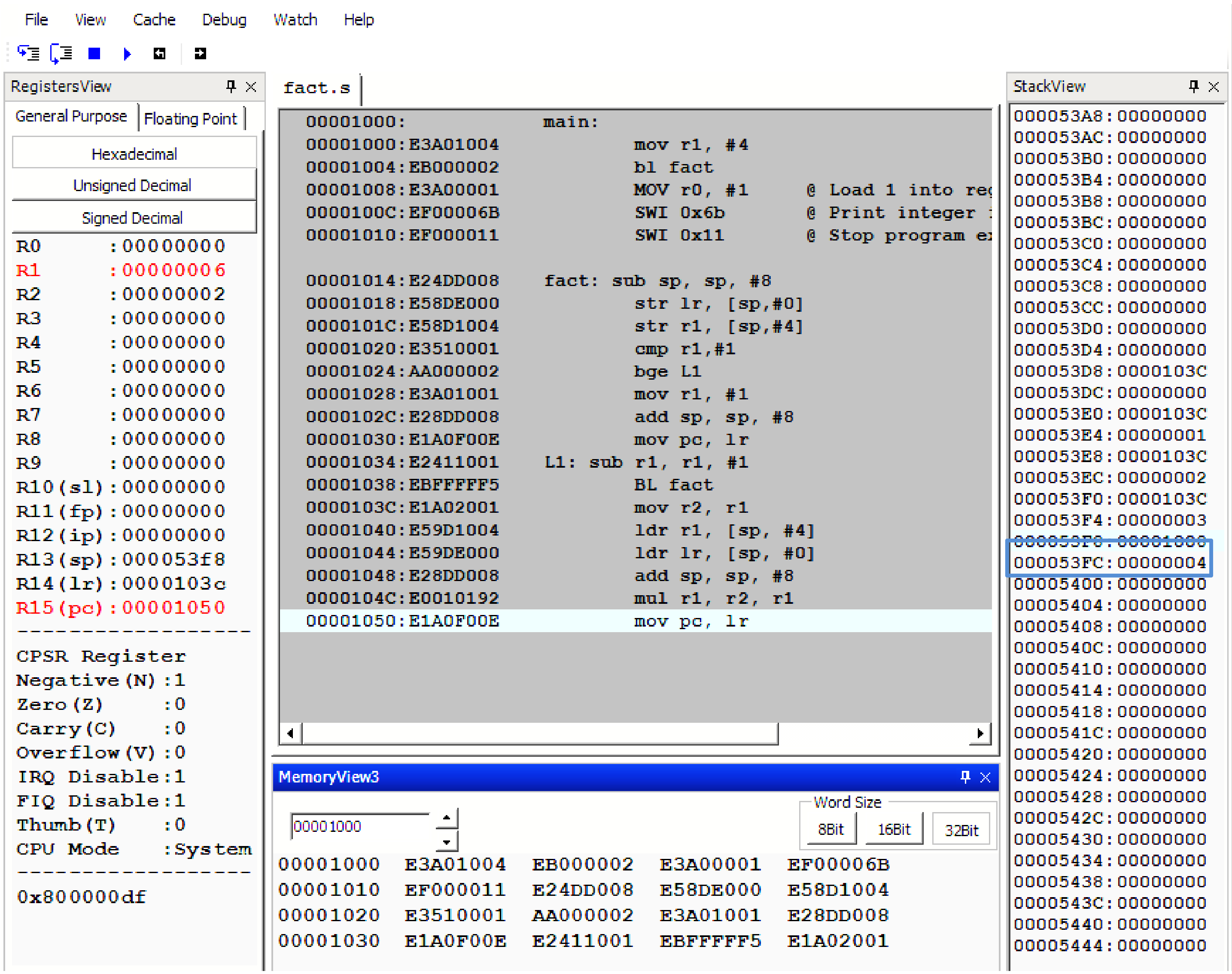


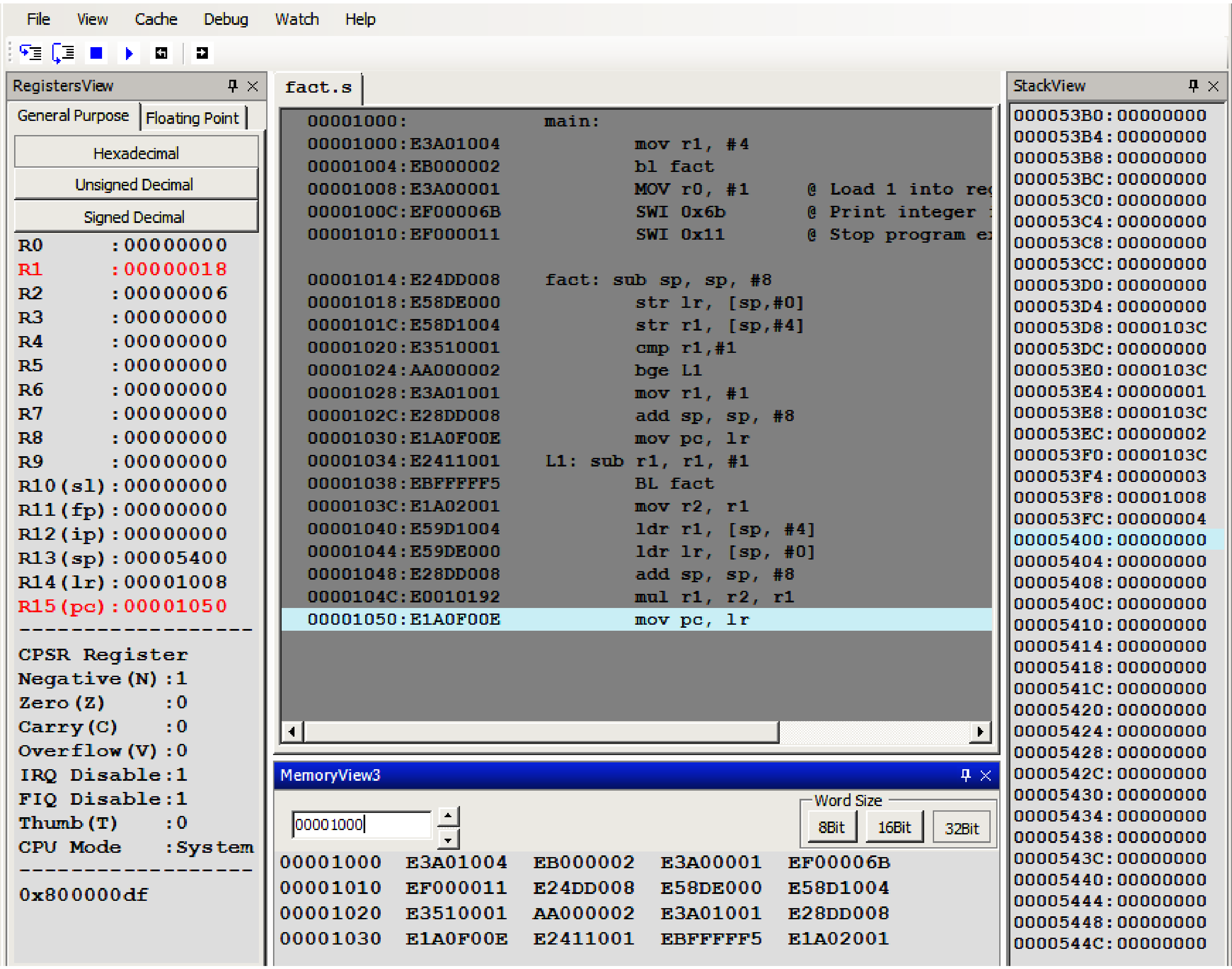
File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000002
R2 :00000001
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f0
R14(lr):0000103c
R15(pc):00001050
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000

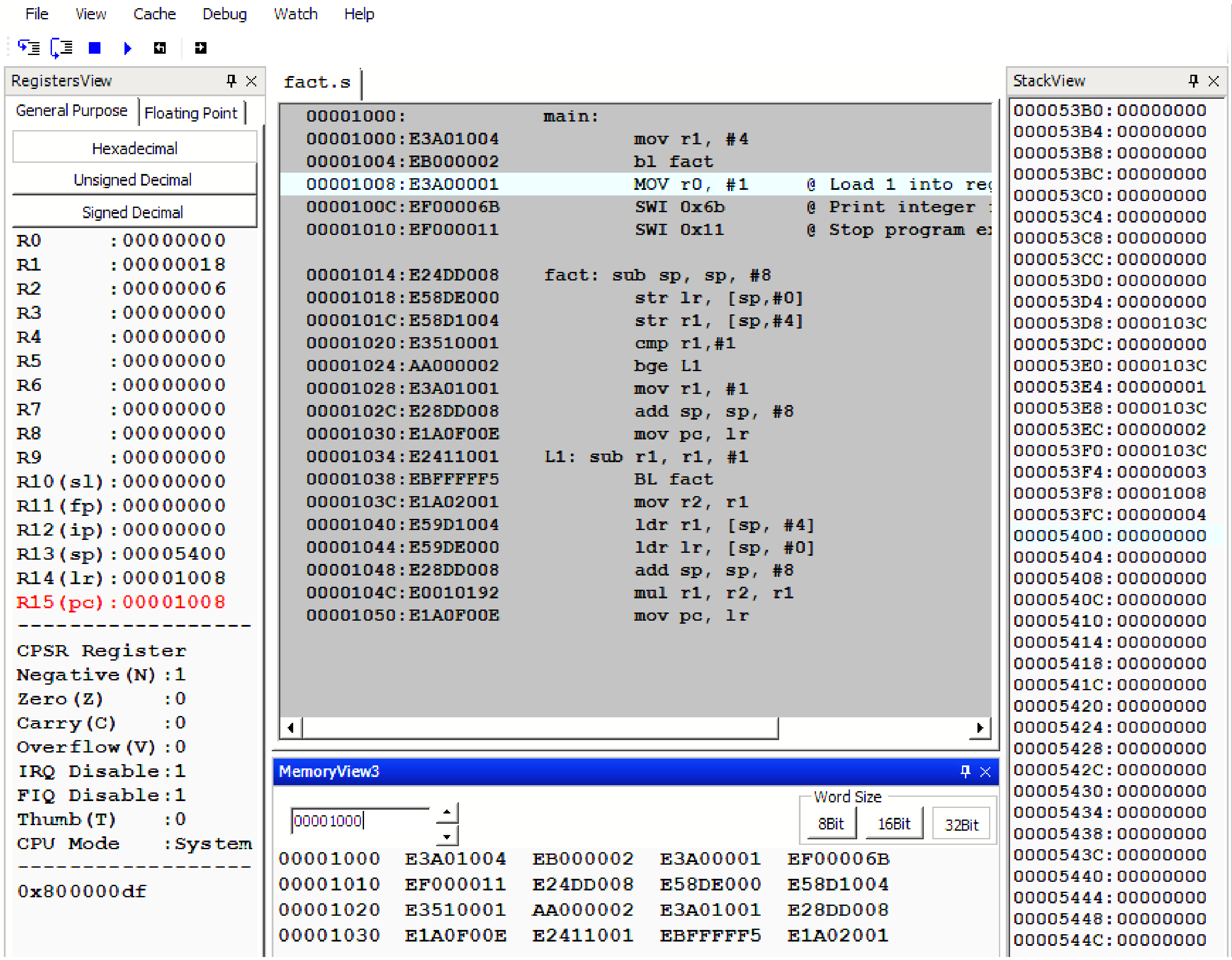


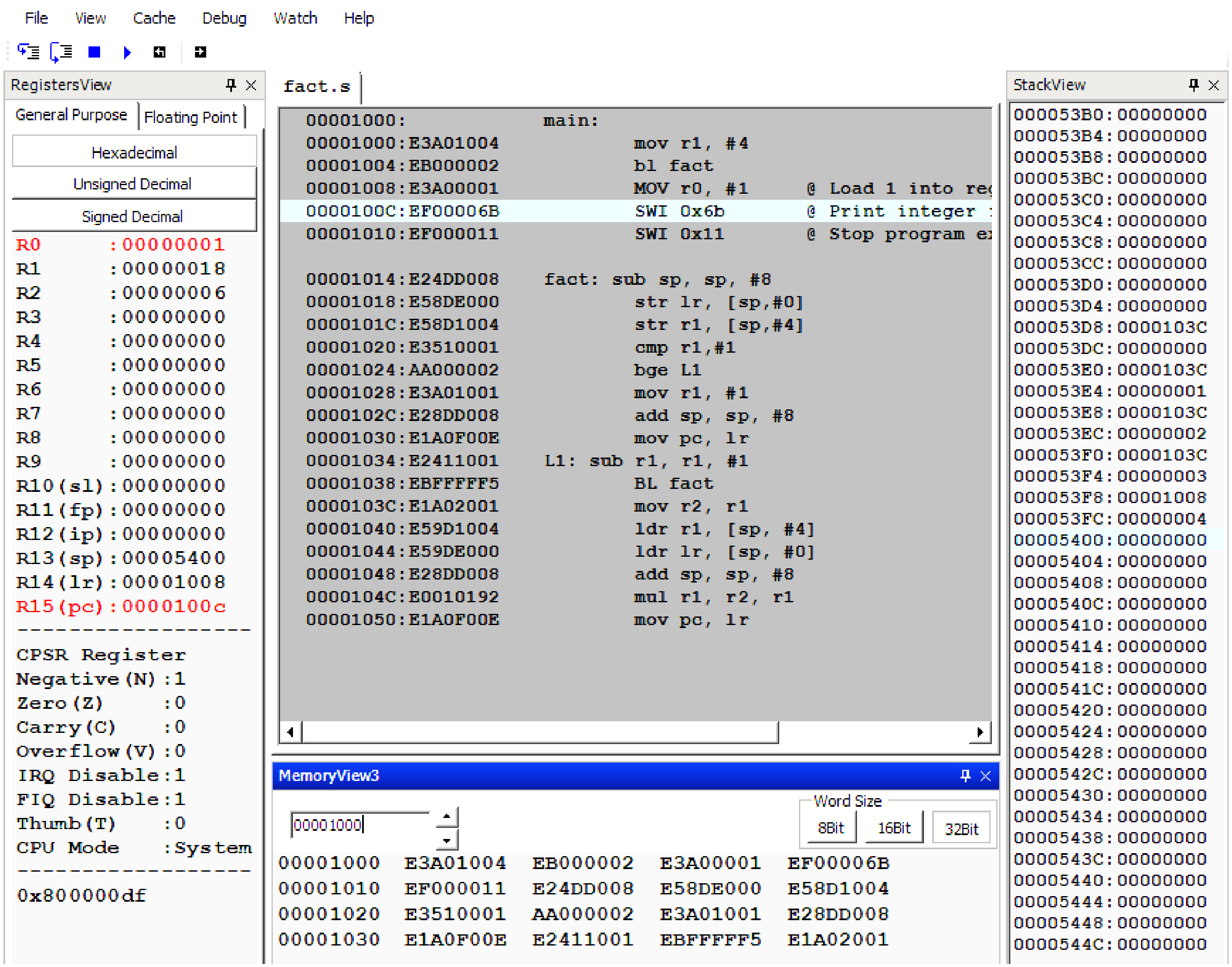
File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000002
R2 :00000001
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f0
R14(lr):0000103c
R15(pc):0000103c
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit
16Bit
32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A0:00000000
000053A4:00000000
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000

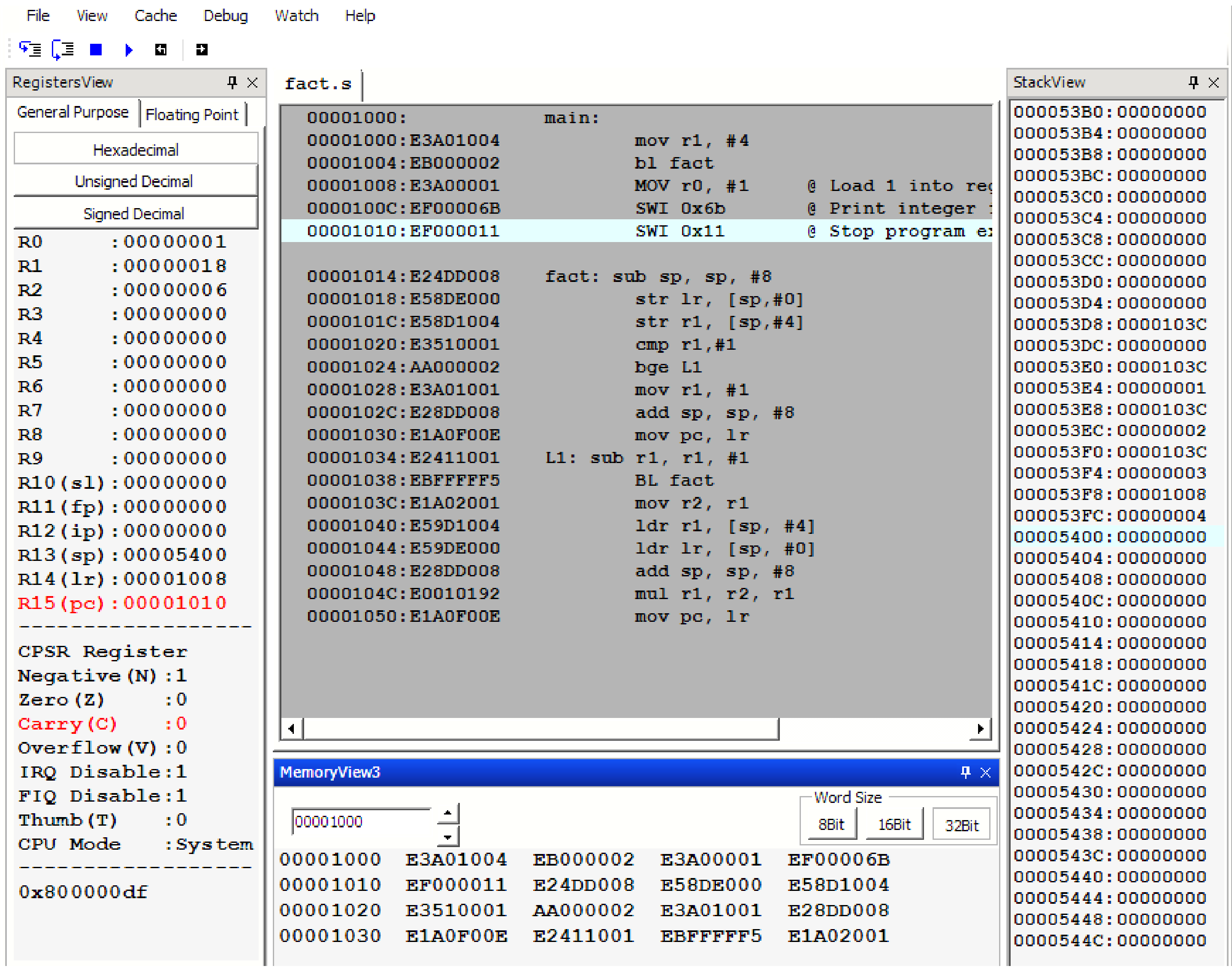

File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000003
R2 :00000002
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f8
R14(lr):0000103c
R15(pc):0000104c
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000


File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000006
R2 :00000002
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f8
R14(lr):0000103c
R15(pc):00001050
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053A8:00000000
000053AC:00000000
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000


File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000018
R2 :00000006
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00001008
R15(pc):00001050
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000
00005448:00000000
0000544C:00000000


File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000018
R2 :00000006
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00001008
R15(pc):00001008
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000
00005448:00000000
0000544C:00000000


File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000001
R1 :00000018
R2 :00000006
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00001008
R15(pc):0000100c
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000
00005448:00000000
0000544C:00000000


File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000001
R1 :00000018
R2 :00000006
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00001008
R15(pc):00001010
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x800000df
fact.s
00001000: main:
00001000:E3A01004 mov r1, #4
00001004:EB000002 bl fact
00001008:E3A00001 MOV r0, #1 @ Load 1 into re
0000100C:EF00006B SWI 0x6b @ Print integer
00001010:EF000011 SWI 0x11 @ Stop program e
00001014:E24DD008 fact: sub sp, sp, #8
00001018:E58DE000 str lr, [sp,#0]
0000101C:E58D1004 str r1, [sp,#4]
00001020:E3510001 cmp r1,#1
00001024:AA000002 bge L1
00001028:E3A01001 mov r1, #1
0000102C:E28DD008 add sp, sp, #8
00001030:E1A0F00E mov pc, lr
00001034:E2411001 L1: sub r1, r1, #1
00001038:EBFFFFF5 BL fact
0000103C:E1A02001 mov r2, r1
00001040:E59D1004 ldr r1, [sp, #4]
00001044:E59DE000 ldr lr, [sp, #0]
00001048:E28DD008 add sp, sp, #8
0000104C:E0010192 mul r1, r2, r1
00001050:E1A0F00E mov pc, lr
MemoryView3
00001000
Word Size
8Bit 16Bit 32Bit
00001000 E3A01004 EB000002 E3A00001 EF00006B
00001010 EF000011 E24DD008 E58DE000 E58D1004
00001020 E3510001 AA000002 E3A01001 E28DD008
00001030 E1A0F00E E2411001 EBFFFFF5 E1A02001
StackView
000053B0:00000000
000053B4:00000000
000053B8:00000000
000053BC:00000000
000053C0:00000000
000053C4:00000000
000053C8:00000000
000053CC:00000000
000053D0:00000000
000053D4:00000000
000053D8:0000103C
000053DC:00000000
000053E0:0000103C
000053E4:00000001
000053E8:0000103C
000053EC:00000002
000053F0:0000103C
000053F4:00000003
000053F8:00001008
000053FC:00000004
00005400:00000000
00005404:00000000
00005408:00000000
0000540C:00000000
00005410:00000000
00005414:00000000
00005418:00000000
0000541C:00000000
00005420:00000000
00005424:00000000
00005428:00000000
0000542C:00000000
00005430:00000000
00005434:00000000
00005438:00000000
0000543C:00000000
00005440:00000000
00005444:00000000
00005448:00000000
0000544C:00000000

Note: The following is for the 32-bit ARM7, see Chapter 02_COD 4e ARM

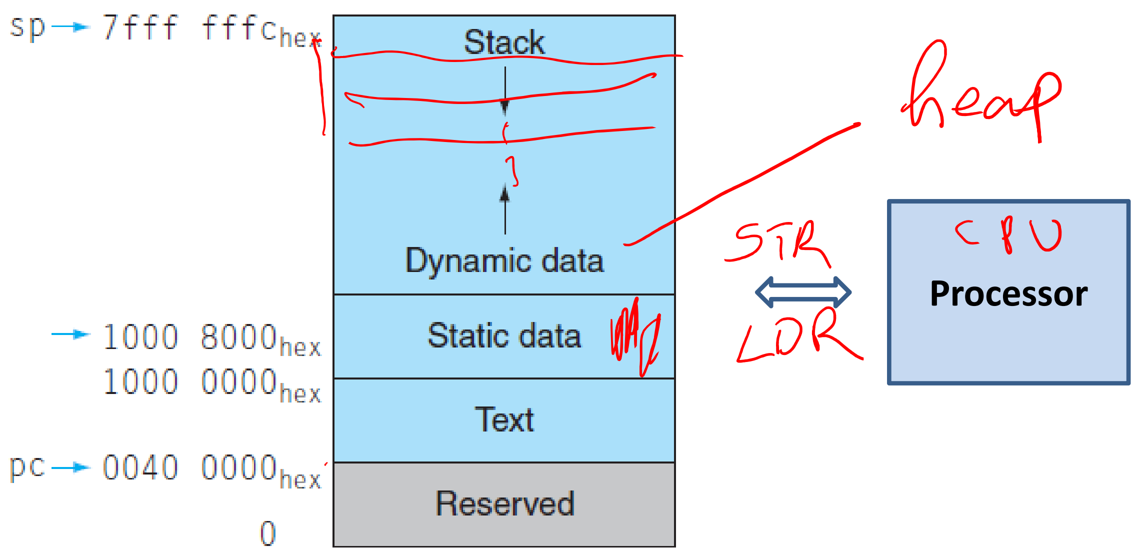


FIGURE 2.13 Typical ARM memory allocation for program and data. These addresses are only a software convention, and not part of the ARM architecture. The stack pointer is initialized to `7fff fffc`$_{hex}$ and grows down toward the data segment. At the other end, the program code ("text") starts at `0040 0000`$_{hex}$. The static data starts at `1000 0000`$_{hex}$. Dynamic data, allocated by `malloc` in C and by `new` in Java, is next. It grows up toward the stack in an area called the heap.

ARM assembly program

label	operation	operand	comments
.text			
main:			
	LDR	R1, value	@ load value
	STR	R1, result	
	SWI	#11	
.data			
value:	.word	0x0000C123	
result:	.word	0	

pseudo instruction

```
gcd(a, b) {
   if(a==b)          return a;
   else if (a>b)   return gcd(a-b, b);
   else               return gcd(a, b-a);
}
```

gcd:

sub sp, sp, #4 @ push

str lr, [sp, #0] @push

cmp r0, r1

beq Return

If: ble Else

sub r0, r0, r1

~~b Rec~~ bl gcd

Else: sub r1, r1, r0

~~Rec:~~ bl gcd @ recursive call

Return: ldr lr, [sp, #0] @ pop

add sp, sp, #4 @ pop

move r7, r0

mov pc, lr

LIFO

```
while (a!=b) {

    if (a > b) {
        a = a - b;
    } else {
        b = b – a;
    }
}
return a;
```

"Normal" Assembler

```
gcd     cmp r0, r1       ;reached the end?
        beq stop
        blt less         ;if r0 > r1
        sub r0, r0, r1   ;subtract r1 from r0
        bal gcd
less    sub r1, r1, r0   ;subtract r0 from r1
        bal gcd
stop
```

ARM Conditional Assembler

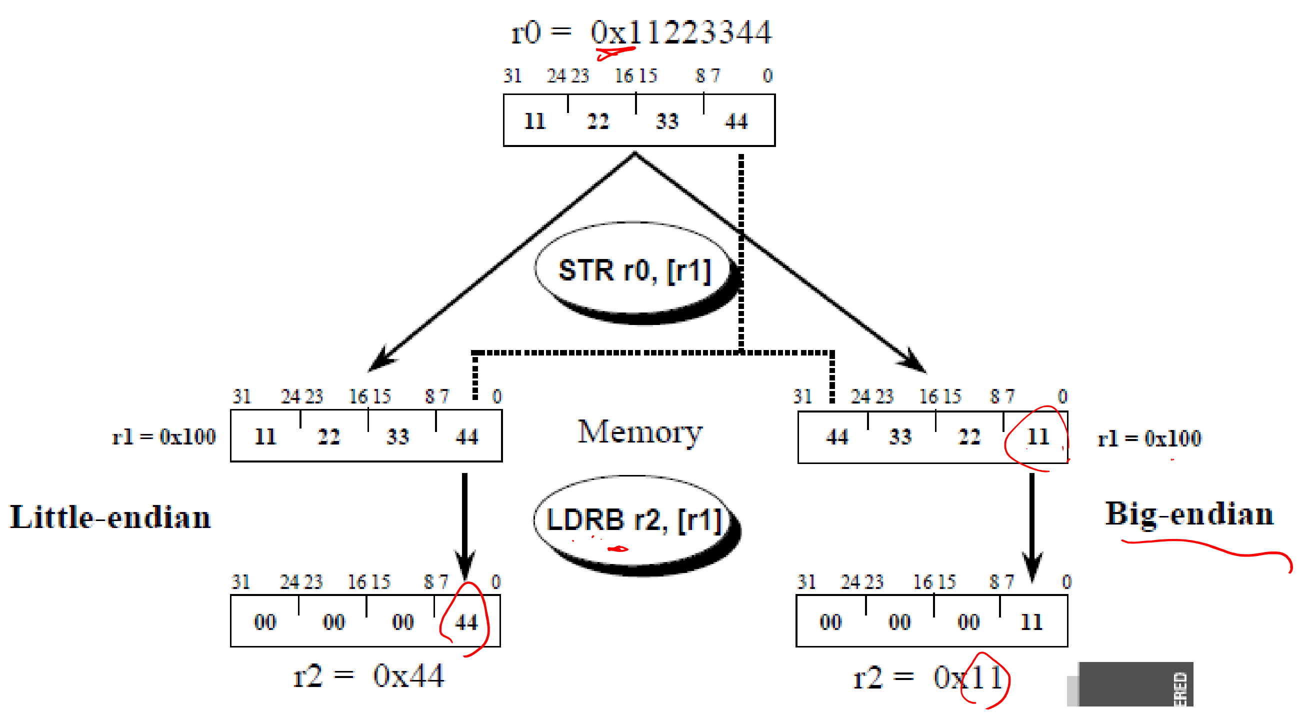
```
gcd     cmp   r0, r1       ;if r0 > r1
        subgt r0, r0, r1   ;subtract r1 from r0
        sublt r1, r1, r0   ;else subtract r0 from r1
        bne   gcd          ;reached the end?
```

Handling large immediate values, label addresses, words, and bytes, …

```
.text
  @mov r0, #345          @ see this number cannot be used as immediate value

  ldr r0, =0x12345678    @ the way to load a large number to register
                         @ see where the number is and pc-relative addressing

  ldr r1, =myByte        @ the way to load address of a label to register

  ldr r2, [r1]           @ see the order of these 4 bytes in memory and in register

  str r0, [r1]           @ see the 4 bytes in a word are stored in memory (little endian)

  ldrb r4, [r1]          @ see which byte in 0x12345678 is loaded back

.data
myByte: .byte 1, 2, 3, 4
```

Little-endian byte ordering – least-significant byte of word or half-word stored at lower address in memory

Big-endian byte ordering – most-significant byte of word or half-word stored at lower address in memory



- The ARM architecture supports both little-endian and big-endian access to memory. The ARM-Sim# supports only the little-endian format (the same as the Intel architecture which hosts the ARMSim#).

- This is only relevant when data is stored by words and is then accessed by bytes.

Etymology [edit]

In 1726, Jonathan Swift described in his satirical novel *Gulliver's Travels* tensions in Lilliput and Blefuscu: whereas royal edict in Lilliput requires cracking open one's soft-boiled egg at the small end, inhabitants of the rival kingdom of Blefuscu crack theirs at the big end (giving them the moniker *Big-endians*).[2][3] The terms *little-endian* and *endianness* have a similar intent.[4]

Wikisource has original text related to this article:
Gulliver's Travels (Part I, Chapter IV)

Danny Cohen's "On Holy Wars and a Plea for Peace" published in 1980[3] ends with: "Swift's point is that the difference between breaking the egg at the little-end and breaking it at the big-end is trivial. Therefore, he suggests, that everyone does it in his own preferred way. We agree that the difference between sending eggs with the little- or the big-end first is trivial, but we insist that everyone must do it in the same way, to avoid anarchy. Since the difference is trivial we may choose either way, but a decision must be made."

This trivial difference was the reason for a hundred-years war between the fictional kingdoms. It is widely assumed that Swift was either alluding to the historic War of the Roses or – more likely – parodying through oversimplification the religious discord in England, Ireland and Scotland brought about by the conflicts between the Roman Catholics (Big Endians) on the one side and the Anglicans and Presbyterians (Little Endians) on the other.

Credit: http://en.wikipedia.org/wiki/Endianness