Microprocessor & Computer Architecture (μpCA)

UE19CS252

Dr. D. C. Kiran

Department of
Computer Science and Engineering

Microprocessor & Computer Architecture (μpCA)

Block Transfer Instructions

Procedure Call or Subroutine

Dr. D. C. Kiran

Department of Computer Science and Engineering

Microprocessor & Computer Architecture (μpCA)

Syllabus

Unit 1: Basic Processor Architecture and Design

- ~~Microprocessor Overview~~
- ~~CISC VS RISC~~
- ~~Introduction to ARM Processor & Applications~~
- ~~ARM Architecture Overview~~
- ~~Different ARM processor Modes~~
- ~~Register Bank~~
- ~~ARM Program structure~~
- ~~ARM Instruction Format~~
- **ARM INSTRUCTION SET**

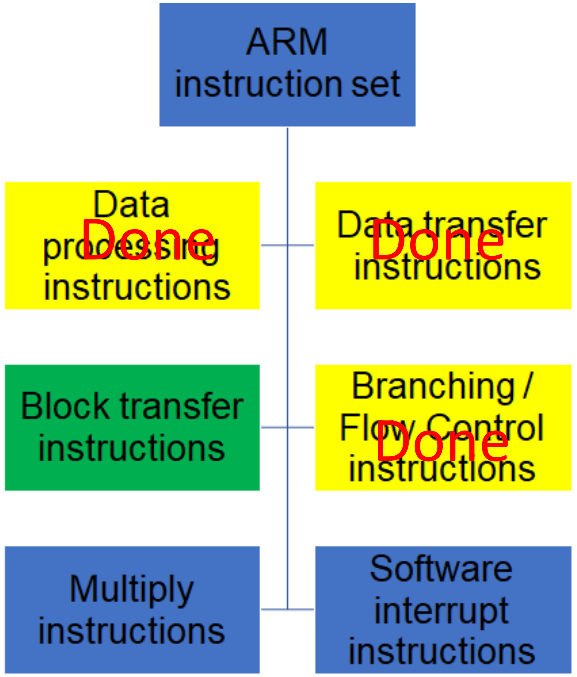
  ~~Data Processing Instructions~~

  ~~Flow Control Instructions~~

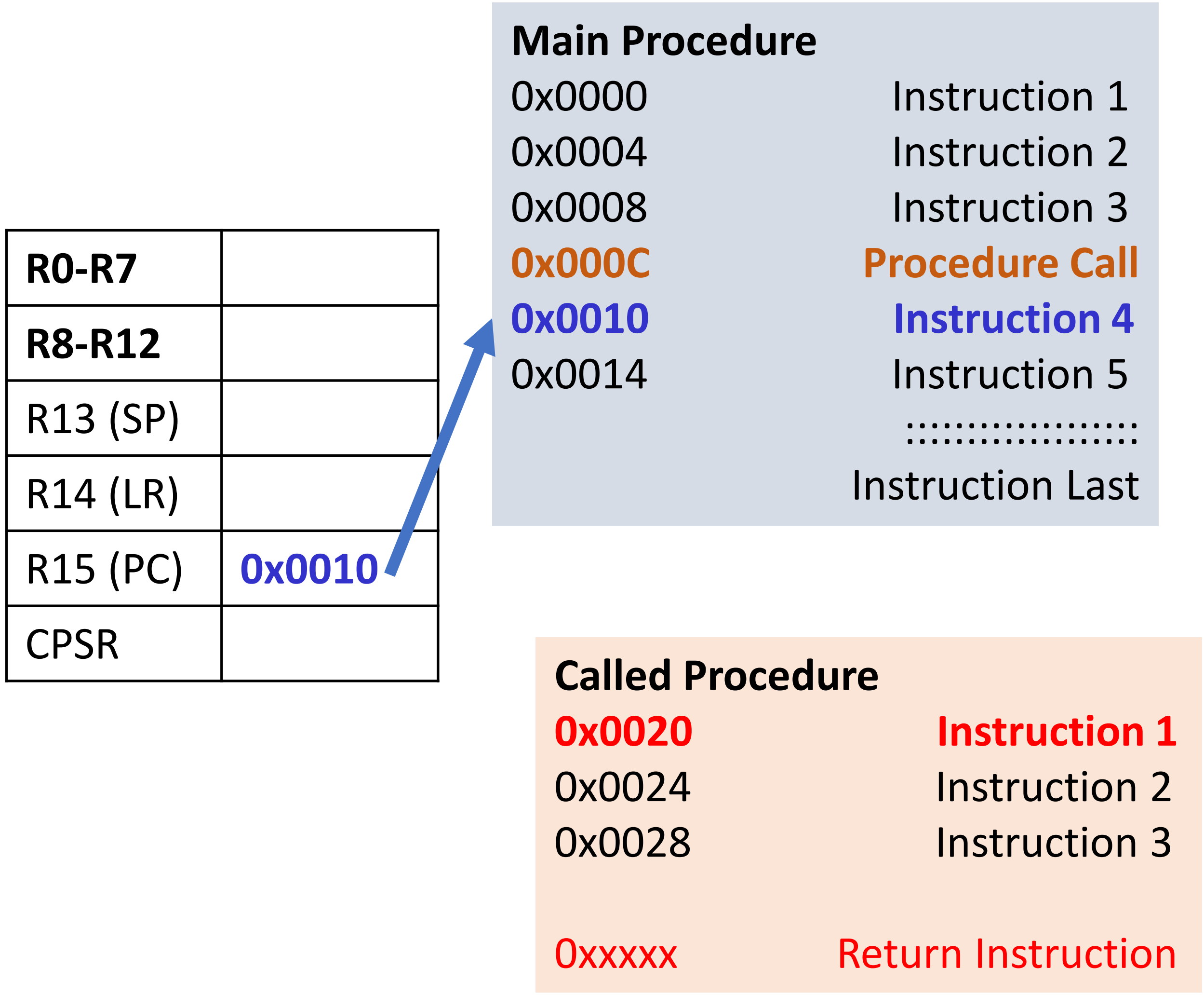
  ~~Data Transfer Instructions~~

  ~~Block Transfer Instructions (Stack)~~

  Procedure Call


ARM instruction set
Data processing instructions
Done
Data transfer instructions
Done
Block transfer instructions
Branching / Flow Control instructions
Done
Multiply instructions
Software interrupt instructions

Procedure Call



R0-R7	
R8-R12	
R13 (SP)	
R14 (LR)	
R15 (PC)	**0x0010**
CPSR	

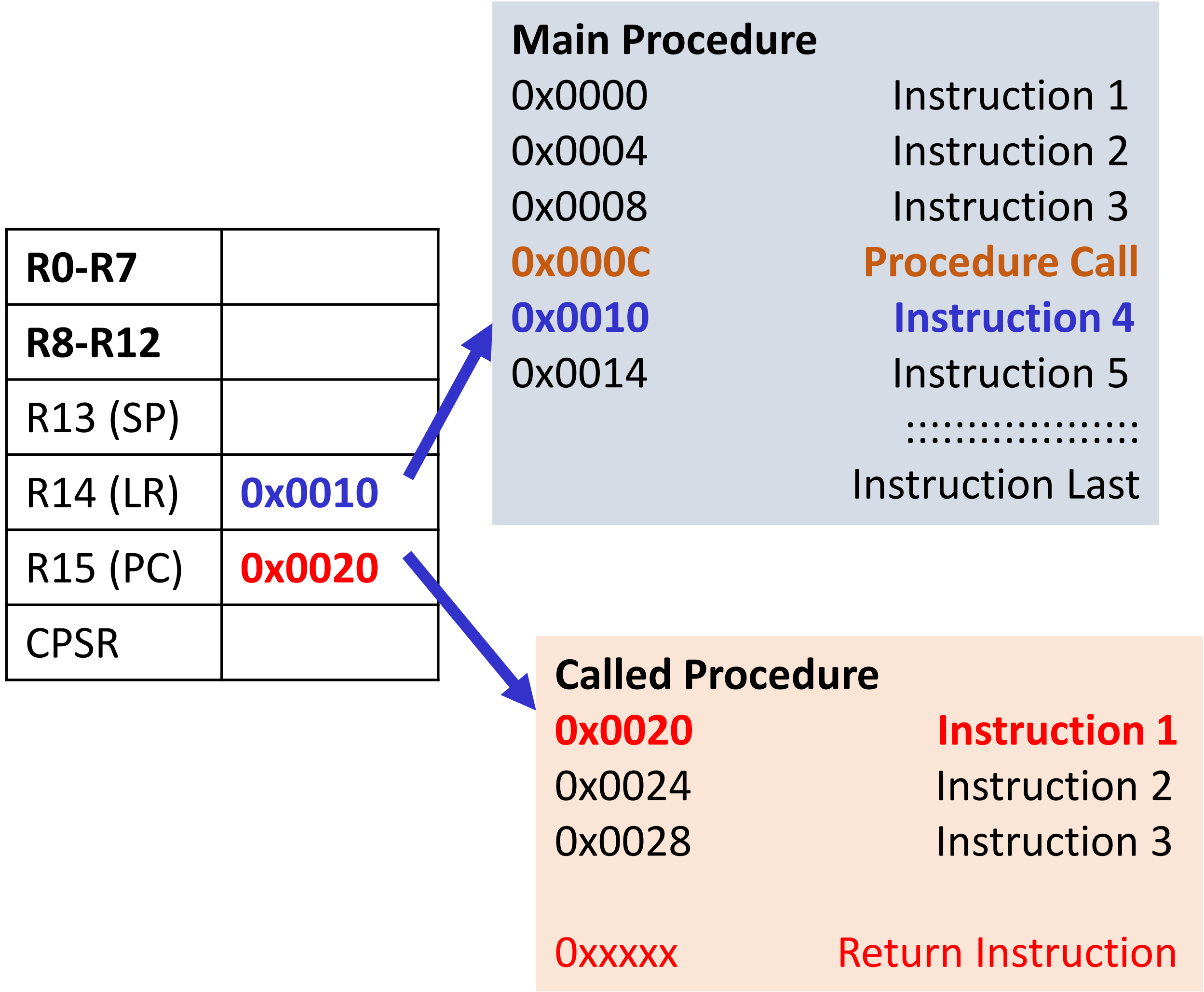
Main Procedure

0x0000	Instruction 1
0x0004	Instruction 2
0x0008	Instruction 3
0x000C	**Procedure Call**
0x0010	**Instruction 4**
0x0014	Instruction 5
	:::::::::::::::::::
	Instruction Last

Called Procedure

0x0020	**Instruction 1**
0x0024	Instruction 2
0x0028	Instruction 3
0xxxxx	Return Instruction

General Structure of Procedure Call:



R0-R7	
R8-R12	
R13 (SP)	
R14 (LR)	**0x0010**
R15 (PC)	**0x0020**
CPSR	

Main Procedure

0x0000	Instruction 1
0x0004	Instruction 2
0x0008	Instruction 3
0x000C	**Procedure Call**
0x0010	**Instruction 4**
0x0014	Instruction 5
	::::::::::::::::::
	Instruction Last

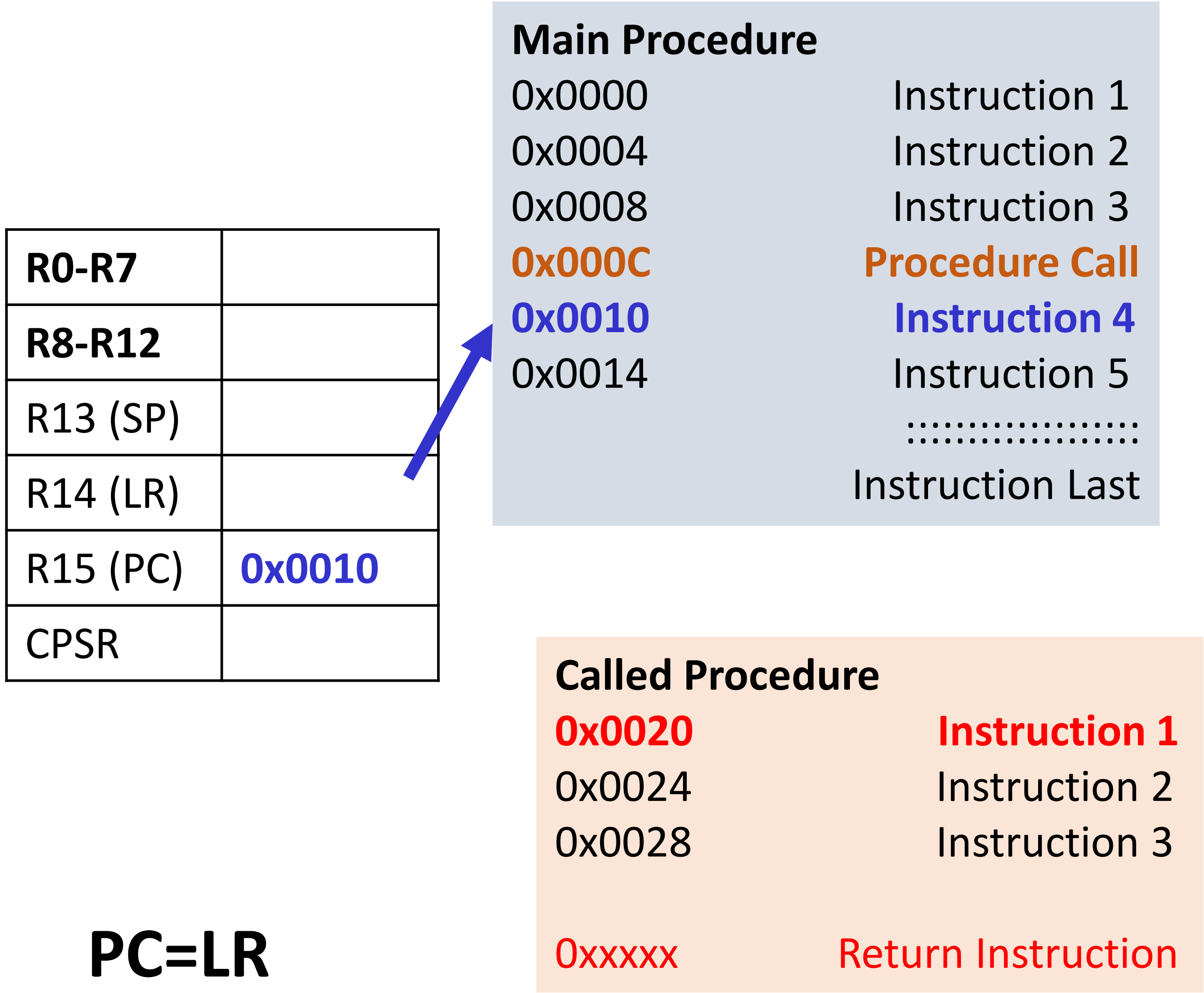
Called Procedure

0x0020	**Instruction 1**
0x0024	Instruction 2
0x0028	Instruction 3
0xxxxx	Return Instruction

LR=PC

PC= Address of the 1st Instruction

General Structure of Procedure Return



R0-R7	
R8-R12	
R13 (SP)	
R14 (LR)	
R15 (PC)	**0x0010**
CPSR	

Main Procedure

0x0000	Instruction 1
0x0004	Instruction 2
0x0008	Instruction 3
0x000C	**Procedure Call**
0x0010	**Instruction 4**
0x0014	Instruction 5
	:::::::::::::::::::
	Instruction Last

Called Procedure

0x0020	**Instruction 1**
0x0024	Instruction 2
0x0028	Instruction 3
0xxxxx	Return Instruction

PC=LR

General Structure of Procedure Call & Return

Main Procedure

```
0x0000                Instruction 1
0x0004                Instruction 2
0x0008                Instruction 3
0x000C                BL Procedure
0x0010                Instruction 4
0x0014                Instruction 5
                         ::::::::::::::::::
                      Instruction Last
```

Called Procedure

```
0x0020        Procedure:  Instruction 1
0x0024                    Instruction 2
0x0028                    Instruction 3
0xxxxx                    MOV PC LR
                             or
                            BX LR
```

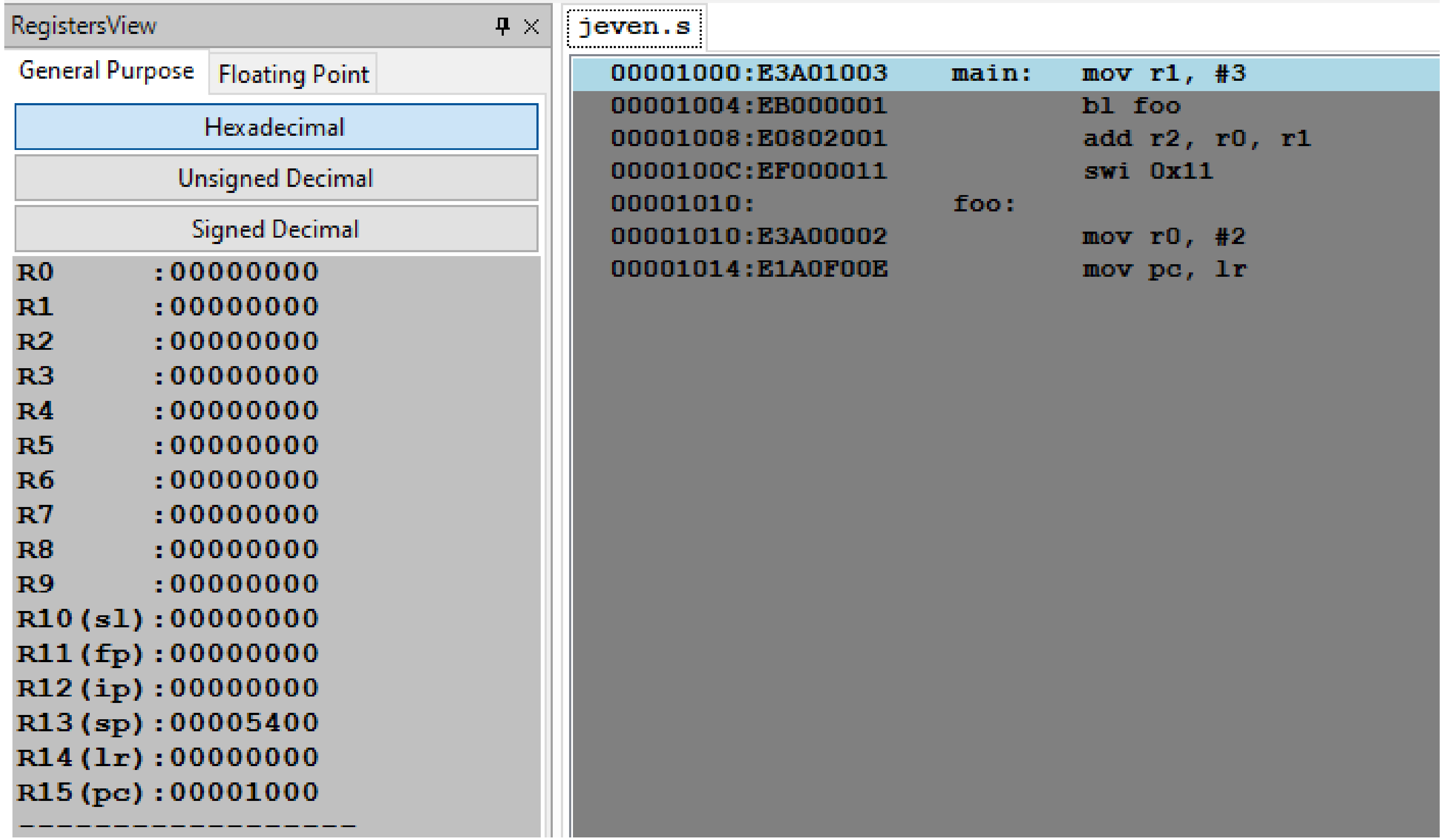
Procedure Call : Example 1

```
main:       mov r1, #3
            bl foo
            add r2, r0, r1
        swi 0x11
foo:
            mov r0, #2
            bx lr
```

```
main:       mov r1, #3
            bl foo
            add r2, r0, r1
        swi 0x11
foo:
            mov r0, #2
        mov pc, lr
```

Microprocessor & Computer Architecture (μpCA)

Procedure Call : Example 1

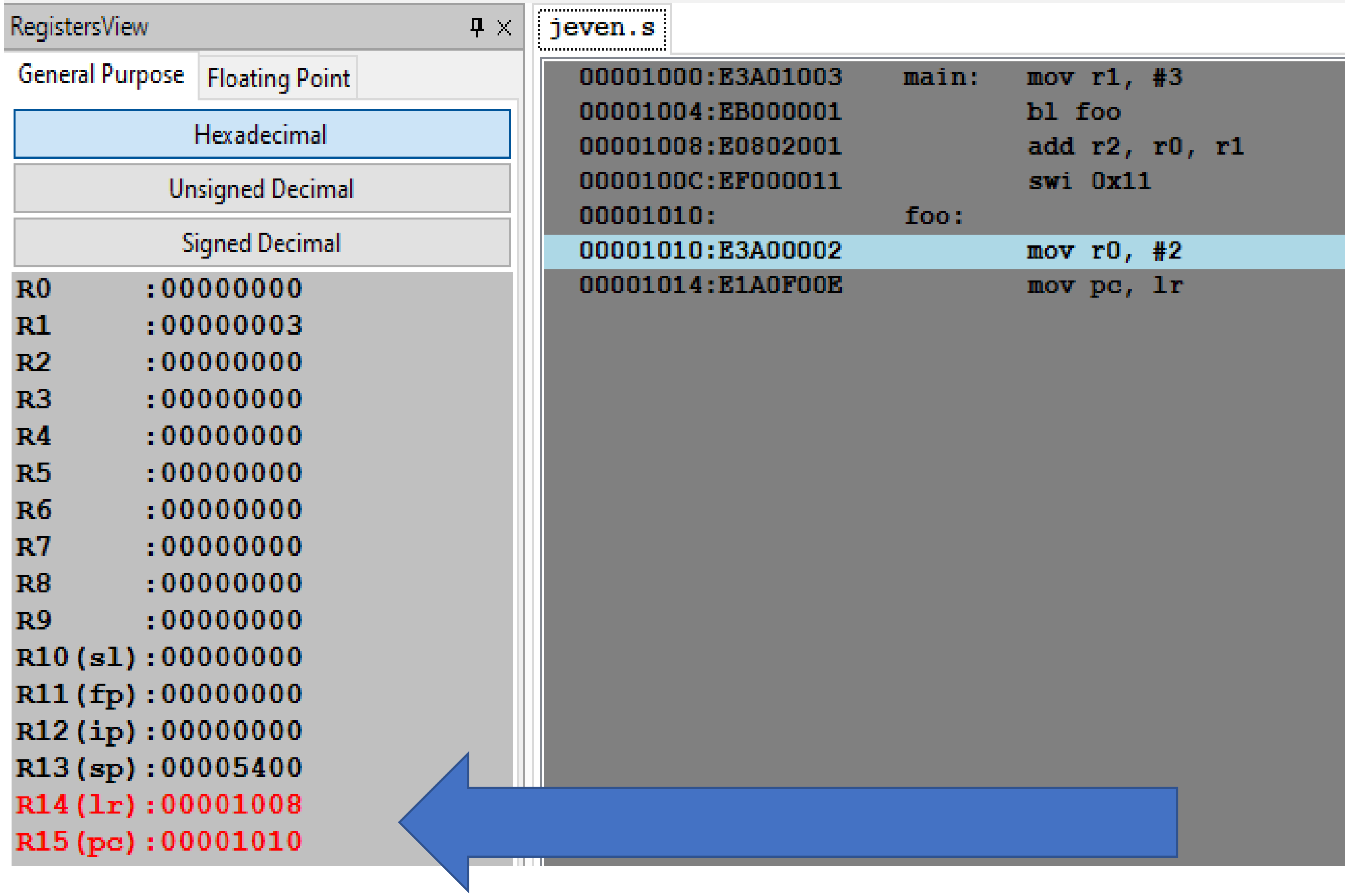


```
R0        :00000000
R1        :00000000
R2        :00000000
R3        :00000000
R4        :00000000
R5        :00000000
R6        :00000000
R7        :00000000
R8        :00000000
R9        :00000000
R10(sl)   :00000000
R11(fp)   :00000000
R12(ip)   :00000000
R13(sp)   :00005400
R14(lr)   :00000000
R15(pc)   :00001000
-----------------
```

```
00001000:E3A01003     main:    mov r1, #3
00001004:EB000001              bl  foo
00001008:E0802001              add r2, r0, r1
0000100C:EF000011              swi 0x11
00001010:             foo:
00001010:E3A00002              mov r0, #2
00001014:E1A0F00E              mov pc, lr
```

Microprocessor & Computer Architecture (μpCA)

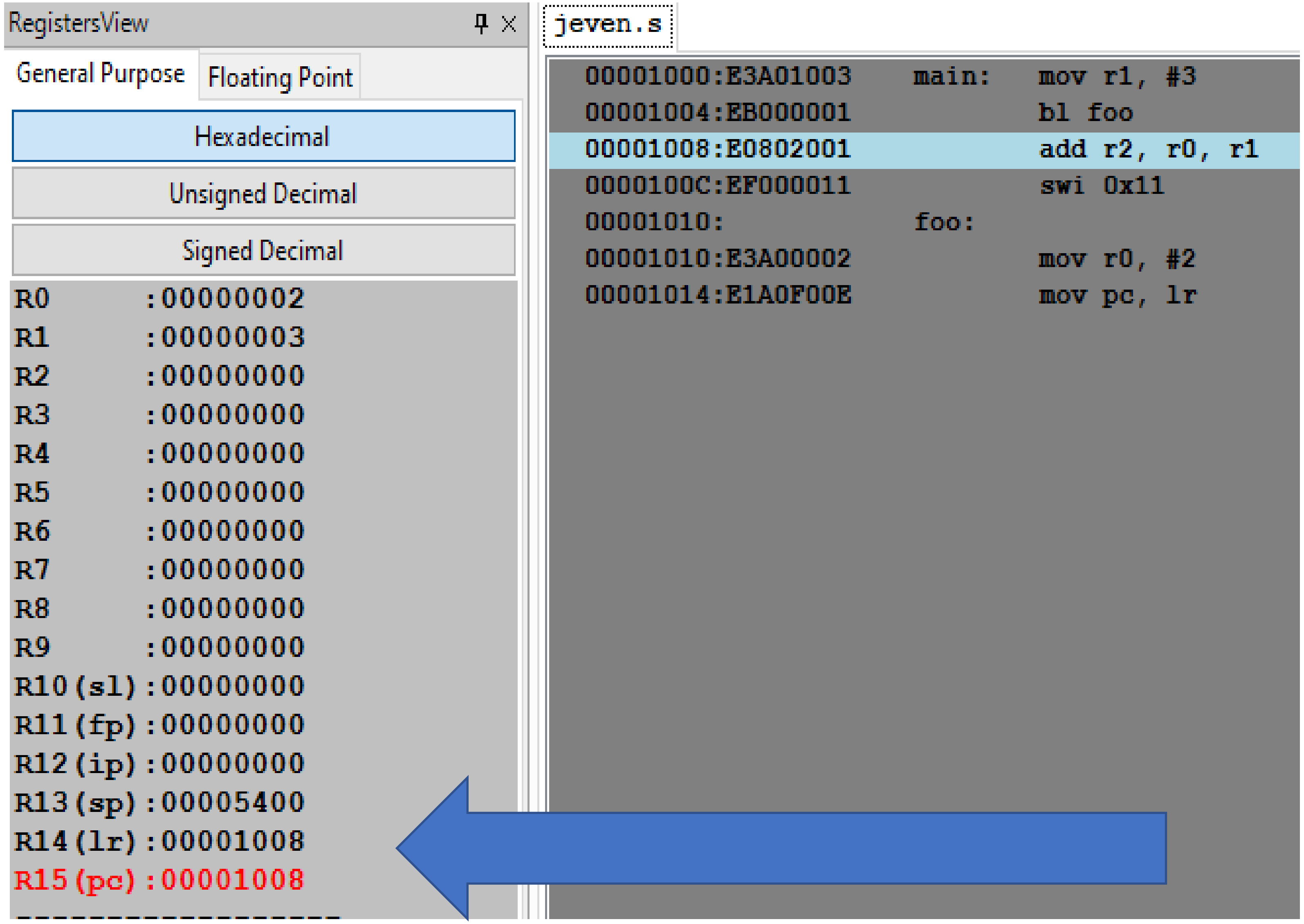
Procedure Call : Example 1



```
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) :00005400
R14(lr) :00001008
R15(pc) :00001010
```

```
00001000:E3A01003    main:    mov r1, #3
00001004:EB000001             bl foo
00001008:E0802001             add r2, r0, r1
0000100C:EF000011             swi 0x11
00001010:            foo:
00001010:E3A00002             mov r0, #2
00001014:E1A0F00E             mov pc, lr
```

Procedure Call : Example 1



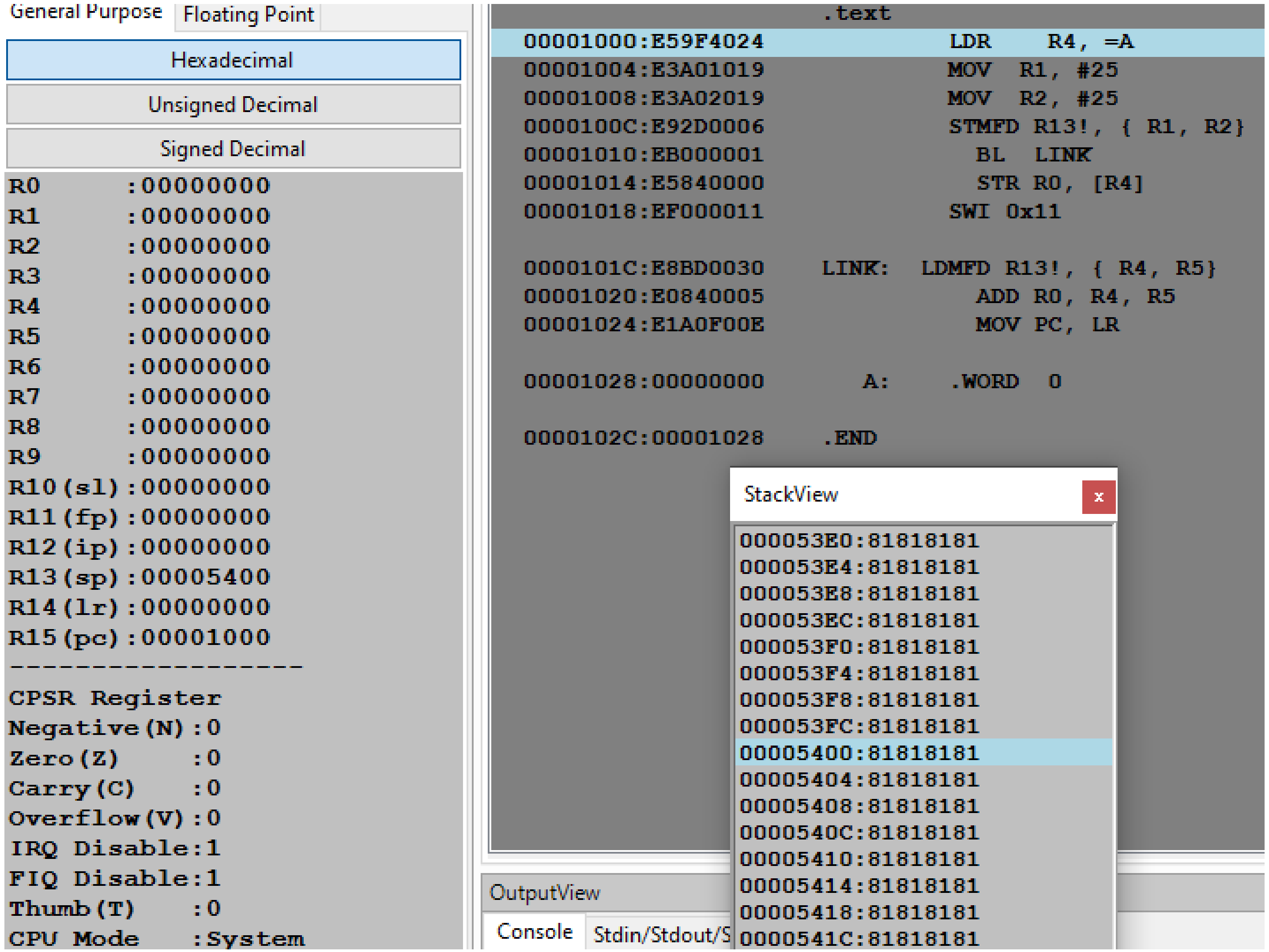
```
R0      :00000002
R1      :00000003
R2      :00000000
R3      :00000000
R4      :00000000
R5      :00000000
R6      :00000000
R7      :00000000
R8      :00000000
R9      :00000000
R10(sl) :00000000
R11(fp) :00000000
R12(ip) :00000000
R13(sp) :00005400
R14(lr) :00001008
R15(pc) :00001008
```

```
00001000:E3A01003    main:    mov r1, #3
00001004:EB000001             bl  foo
00001008:E0802001             add r2, r0, r1
0000100C:EF000011             swi 0x11
00001010:            foo:
00001010:E3A00002             mov r0, #2
00001014:E1A0F00E             mov pc, lr
```

Parameter passing to procedures using stack

```
        LDR    R4, =A
        MOV  R1, #25                      ; parameter1
        MOV  R2, #25                      ; parameter2
        STMFD R13!, { R1, R2}              ; parameters are PUSHed on stack.
        BL  LINK
        STR R0, [R4]                     ; return value in Reg. R0.
        SWI 0x11

LINK:  LDMFD R13!, { R4, R5}           ; parameters are POPed from the  stack
        ADD R0, R4, R5                  ;  Result is in register R0.
        MOV PC, LR

  A:    .WORD  0

.END
```

PARAMETER PASSING TO PROCEDURES USING STACK



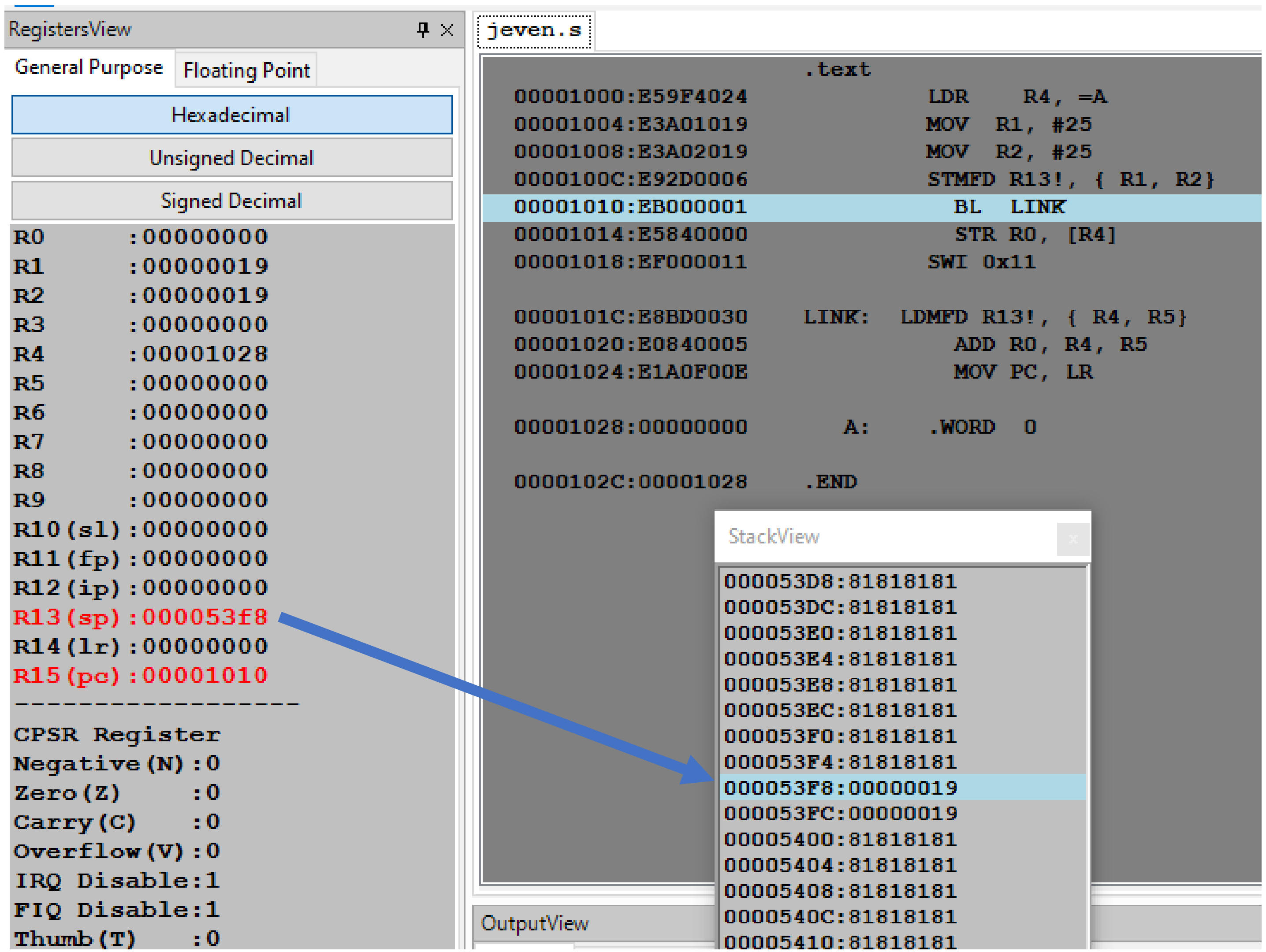
```
R0      :00000000
R1      :00000000
R2      :00000000
R3      :00000000
R4      :00000000
R5      :00000000
R6      :00000000
R7      :00000000
R8      :00000000
R9      :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001000
------------------
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
```

```
                        .text
00001000:E59F4024                LDR     R4, =A
00001004:E3A01019                MOV  R1, #25
00001008:E3A02019                MOV  R2, #25
0000100C:E92D0006                STMFD R13!, { R1, R2}
00001010:EB000001                  BL  LINK
00001014:E5840000                  STR R0, [R4]
00001018:EF000011                SWI 0x11

0000101C:E8BD0030      LINK:  LDMFD R13!, { R4, R5}
00001020:E0840005                  ADD R0, R4, R5
00001024:E1A0F00E                  MOV PC, LR

00001028:00000000          A:     .WORD  0

0000102C:00001028      .END
```

```
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
```

Microprocessor & Computer Architecture (μpCA)

PARAMETER PASSING TO PROCEDURES USING STACK



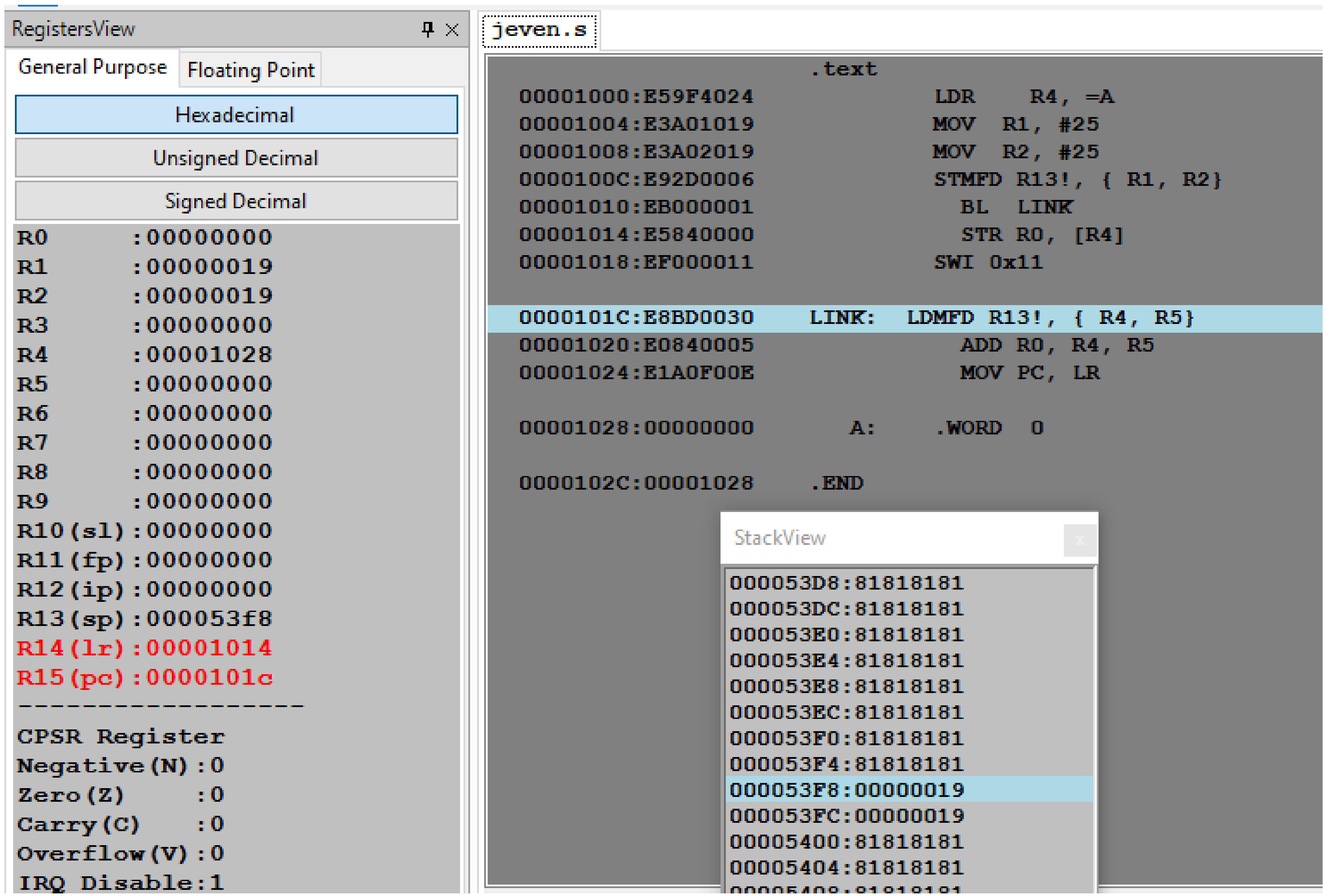
```
R0       :00000000
R1       :00000019
R2       :00000019
R3       :00000000
R4       :00001028
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :000053f8
R14(lr)  :00000000
R15(pc)  :00001010
-------------------
CPSR Register
Negative(N):0
Zero(Z)    :0
Carry(C)   :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
```

```
                          .text
00001000:E59F4024                 LDR    R4, =A
00001004:E3A01019                 MOV  R1, #25
00001008:E3A02019                 MOV  R2, #25
0000100C:E92D0006                 STMFD R13!, { R1, R2}
00001010:EB000001                   BL   LINK
00001014:E5840000                   STR R0, [R4]
00001018:EF000011                 SWI 0x11

0000101C:E8BD0030       LINK:  LDMFD R13!, { R4, R5}
00001020:E0840005                   ADD R0, R4, R5
00001024:E1A0F00E                   MOV PC, LR

00001028:00000000          A:     .WORD  0

0000102C:00001028       .END
```

```
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:00000019
000053FC:00000019
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
```

Microprocessor & Computer Architecture (μpCA)

PARAMETER PASSING TO PROCEDURES USING STACK



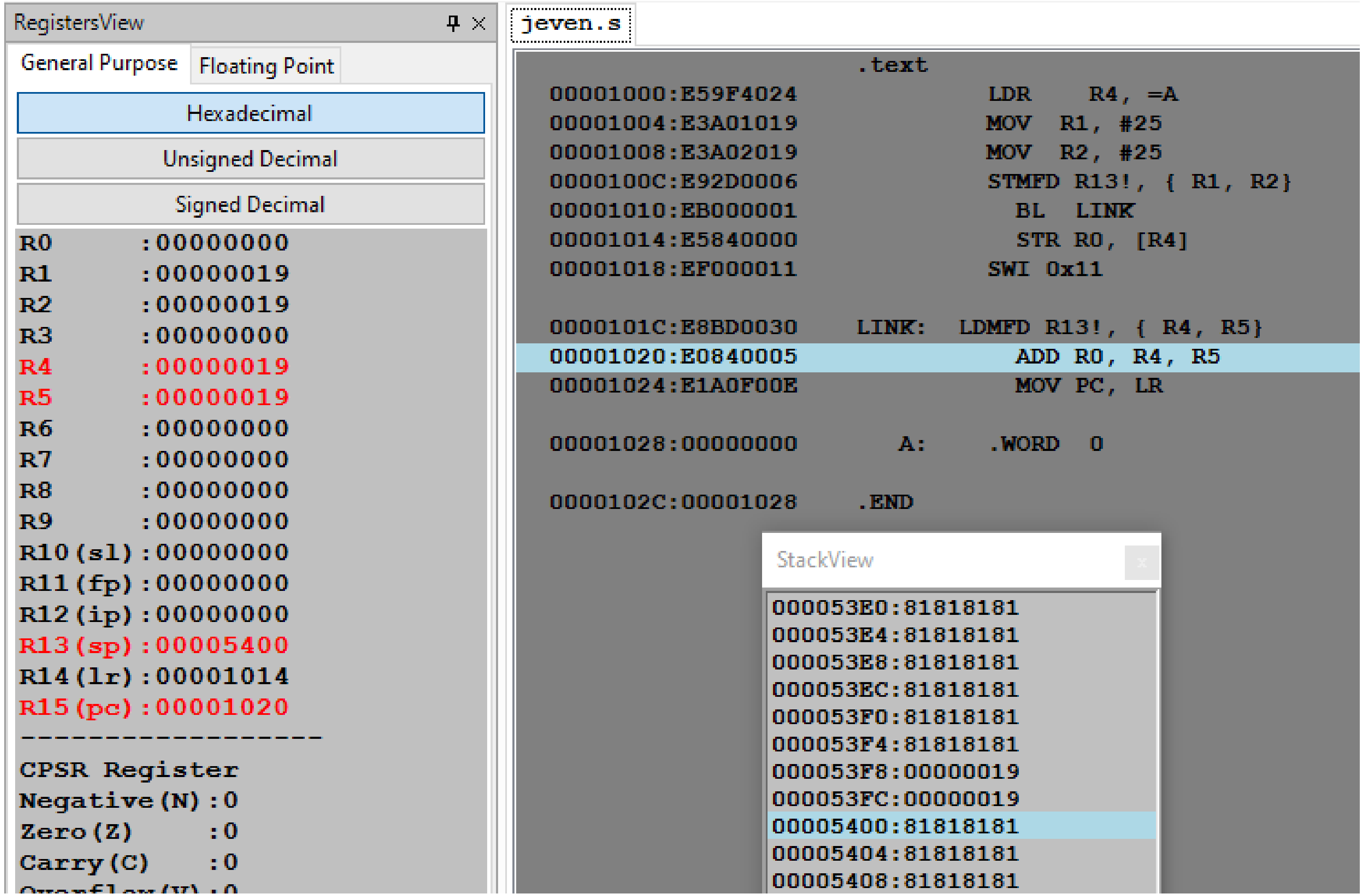
```
R0       :00000000
R1       :00000019
R2       :00000019
R3       :00000000
R4       :00001028
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):000053f8
R14(lr):00001014
R15(pc):0000101c
------------------
CPSR Register
Negative(N):0
Zero(Z)     :0
Carry(C)    :0
Overflow(V):0
IRQ Disable:1
```

```
                         .text
00001000:E59F4024               LDR    R4, =A
00001004:E3A01019               MOV  R1, #25
00001008:E3A02019               MOV  R2, #25
0000100C:E92D0006               STMFD R13!, { R1, R2}
00001010:EB000001                 BL   LINK
00001014:E5840000                 STR R0, [R4]
00001018:EF000011               SWI 0x11

0000101C:E8BD0030        LINK:  LDMFD R13!, { R4, R5}
00001020:E0840005                 ADD R0, R4, R5
00001024:E1A0F00E                 MOV PC, LR

00001028:00000000          A:     .WORD  0

0000102C:00001028        .END
```

```
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:00000019
000053FC:00000019
00005400:81818181
00005404:81818181
00005408:81818181
```

Microprocessor & Computer Architecture (µpCA)

PARAMETER PASSING TO PROCEDURES USING STACK



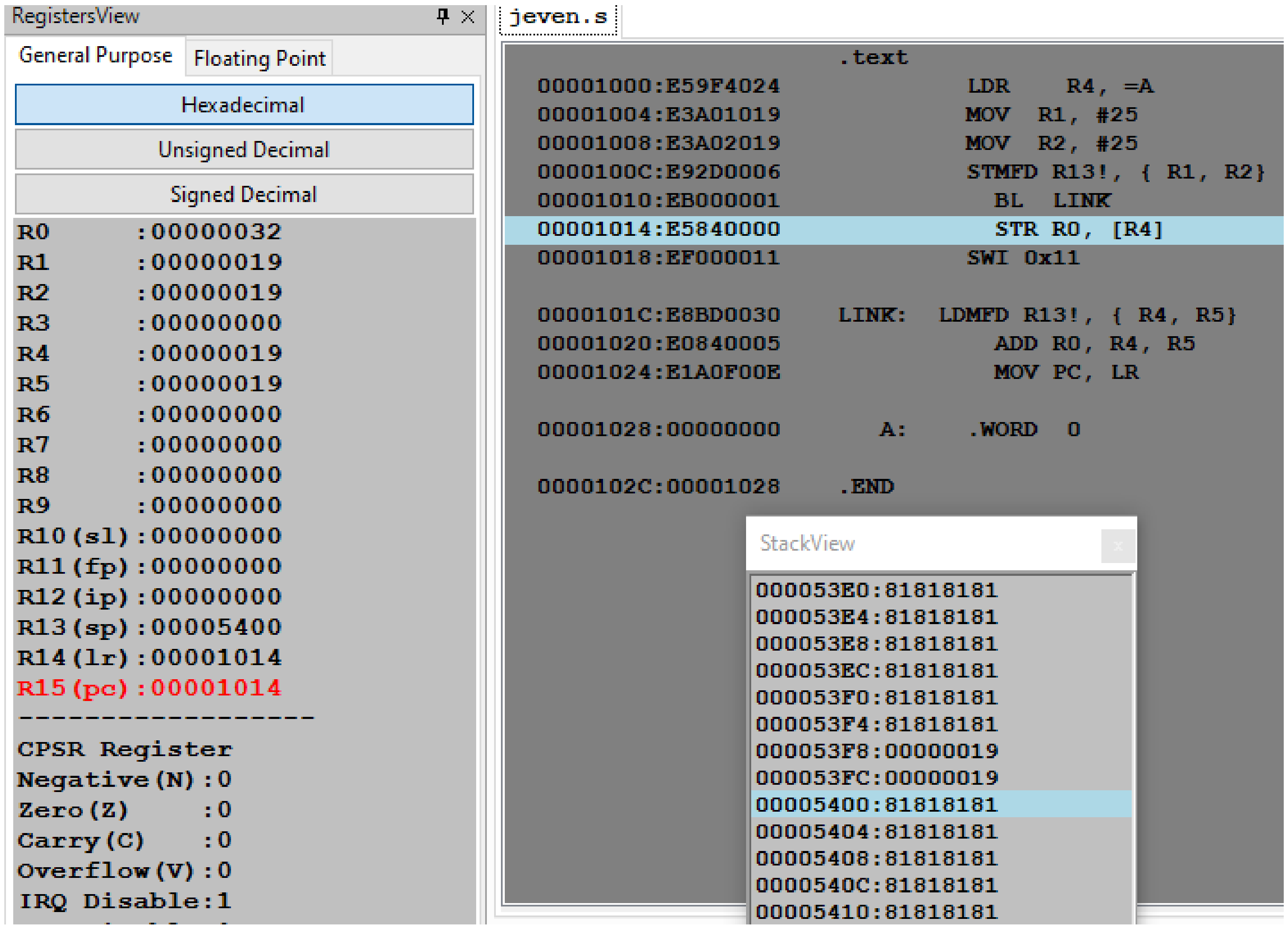
```
R0      :00000000
R1      :00000019
R2      :00000019
R3      :00000000
R4      :00000019
R5      :00000019
R6      :00000000
R7      :00000000
R8      :00000000
R9      :00000000
R10(sl) :00000000
R11(fp) :00000000
R12(ip) :00000000
R13(sp) :00005400
R14(lr) :00001014
R15(pc) :00001020
-------------------
CPSR Register
Negative(N):0
Zero(Z)    :0
Carry(C)   :0
Overflow(V):0
```

```
                            .text
00001000:E59F4024                   LDR    R4, =A
00001004:E3A01019                   MOV   R1, #25
00001008:E3A02019                   MOV   R2, #25
0000100C:E92D0006                   STMFD R13!, { R1, R2}
00001010:EB000001                     BL   LINK
00001014:E5840000                     STR R0, [R4]
00001018:EF000011                   SWI 0x11

0000101C:E8BD0030           LINK:  LDMFD R13!, { R4, R5}
00001020:E0840005                     ADD R0, R4, R5
00001024:E1A0F00E                     MOV PC, LR

00001028:00000000              A:     .WORD  0

0000102C:00001028           .END
```

```
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:00000019
000053FC:00000019
00005400:81818181
00005404:81818181
00005408:81818181
```

Microprocessor & Computer Architecture (μpCA)

Parameter passing to procedures using stack



```
R0        :00000032
R1        :00000019
R2        :00000019
R3        :00000000
R4        :00000019
R5        :00000019
R6        :00000000
R7        :00000000
R8        :00000000
R9        :00000000
R10(sl)   :00000000
R11(fp)   :00000000
R12(ip)   :00000000
R13(sp)   :00005400
R14(lr)   :00001014
R15(pc)   :00001014
------------------
CPSR Register
Negative(N):0
Zero(Z)    :0
Carry(C)   :0
Overflow(V):0
IRQ Disable:1
```

```
                            .text
00001000:E59F4024                    LDR    R4, =A
00001004:E3A01019                    MOV   R1, #25
00001008:E3A02019                    MOV   R2, #25
0000100C:E92D0006                    STMFD R13!, { R1, R2}
00001010:EB000001                      BL   LINK
00001014:E5840000                      STR R0,  [R4]
00001018:EF000011                    SWI 0x11

0000101C:E8BD0030           LINK:  LDMFD R13!, { R4, R5}
00001020:E0840005                      ADD R0, R4, R5
00001024:E1A0F00E                      MOV PC, LR

00001028:00000000              A:      .WORD   0

0000102C:00001028           .END
```

```
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:00000019
000053FC:00000019
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
```

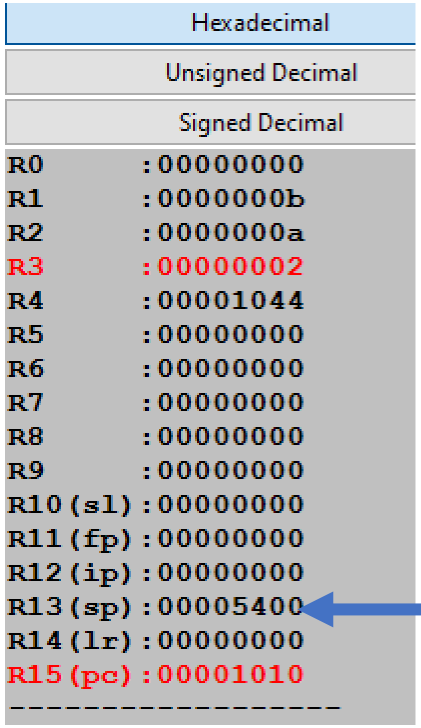
Microprocessor & Computer Architecture (μpCA)

Nested Procedure Call: MUL(ADD(a,b),c)

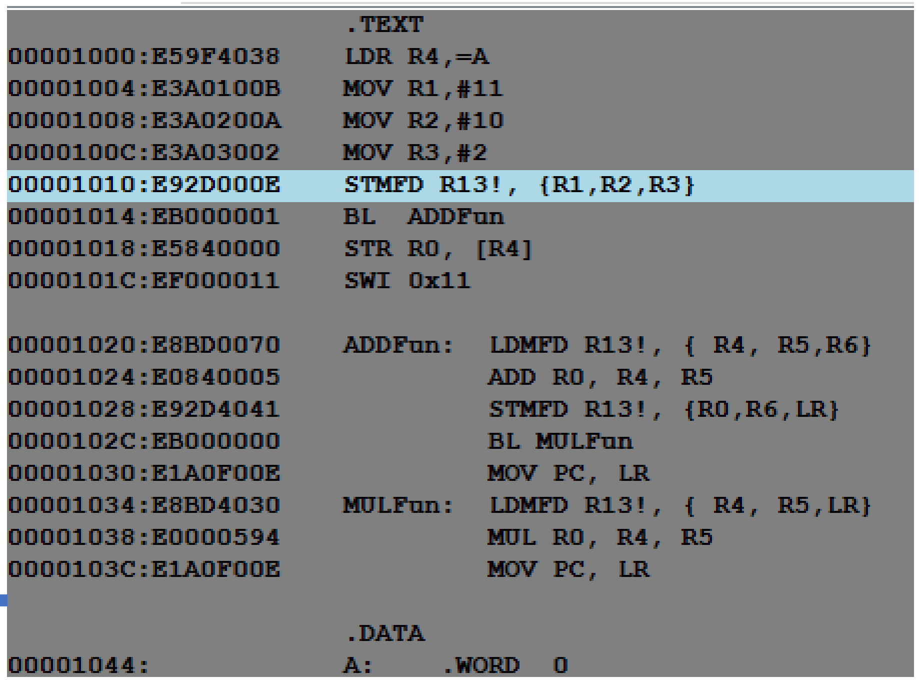
```
.TEXT       ; MAIN Procedure
LDR R4,=A
MOV R1,#11
MOV R2,#10
MOV R3,#2
STMFD R13!, {R1,R2,R3}
BL  ADDFun                              ; Call to ADD Procedure
STR R0, [R4]
SWI 0x11

ADDFun:  LDMFD R13!, { R4, R5,R6}   ; ADD Procedure
     ADD R0, R4, R5
     STMFD R13!, {R0,R6,LR}
     BL MULFun                          ; Call to MUL Procedure
     MOV PC, LR                         ;Return to Main Procedure
MULFun:  LDMFD R13!, { R4, R5,LR}
     MUL R0, R4, R5
     MOV PC, LR                         ; Return to ADD Procedure

.DATA
A:    .WORD  0
```

Microprocessor & Computer Architecture (µpCA)

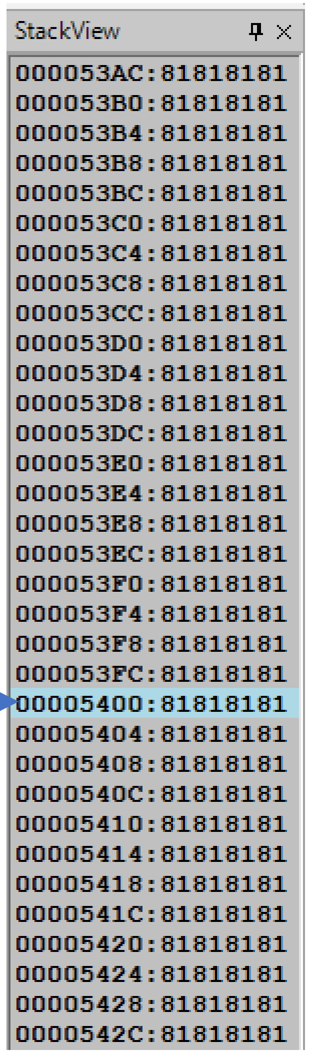
Nested Procedure Call: MUL(ADD(a,b),c)



```
R0      :00000000
R1      :0000000b
R2      :0000000a
R3      :00000002
R4      :00001044
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) :00001010
------------------
```



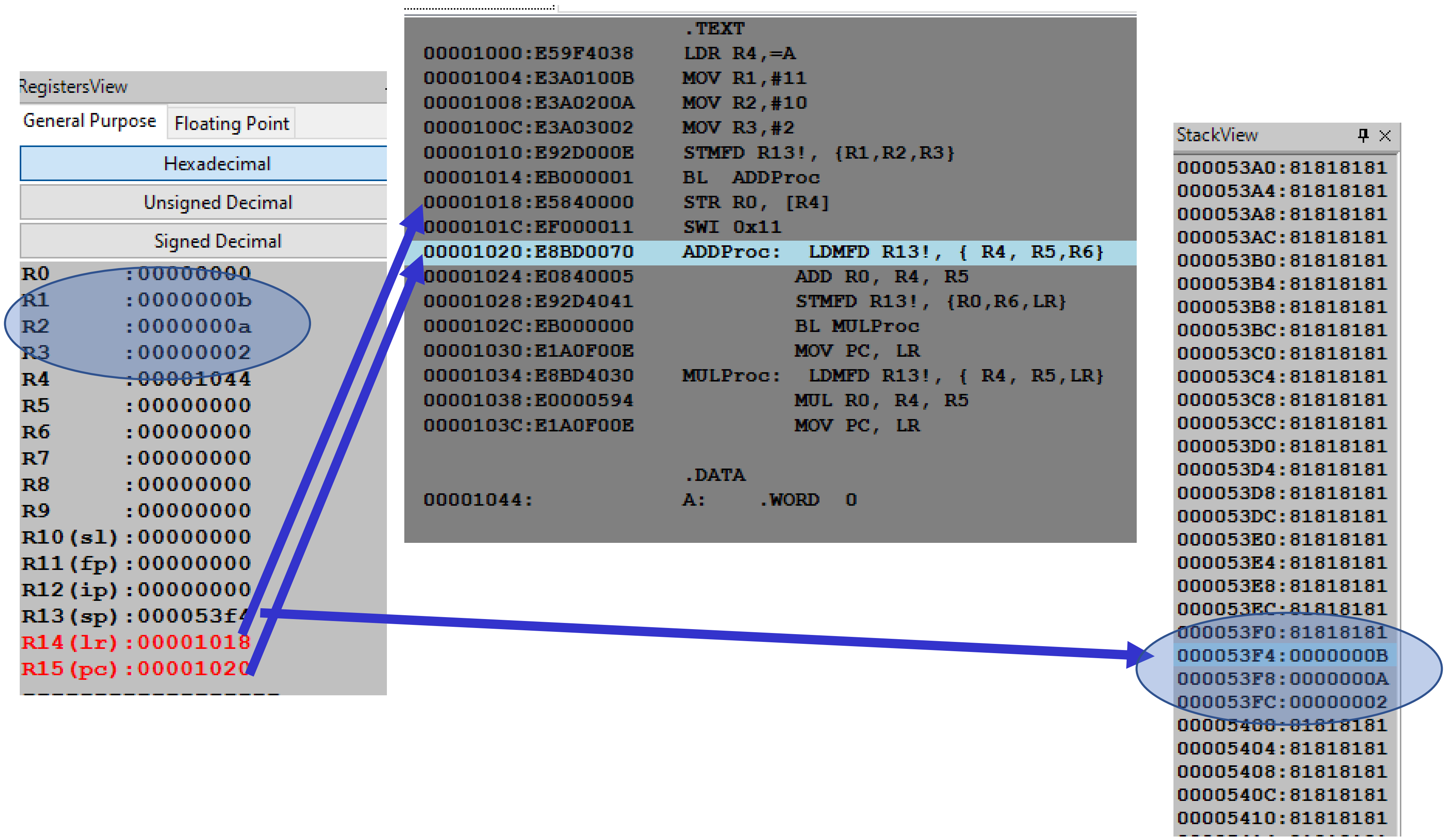
```
                          .TEXT
00001000:E59F4038        LDR R4,=A
00001004:E3A0100B        MOV R1,#11
00001008:E3A0200A        MOV R2,#10
0000100C:E3A03002        MOV R3,#2
00001010:E92D000E        STMFD R13!, {R1,R2,R3}
00001014:EB000001        BL  ADDFun
00001018:E5840000        STR R0, [R4]
0000101C:EF000011        SWI 0x11

00001020:E8BD0070        ADDFun:  LDMFD R13!, { R4, R5,R6}
00001024:E0840005                 ADD R0, R4, R5
00001028:E92D4041                 STMFD R13!, {R0,R6,LR}
0000102C:EB000000                 BL MULFun
00001030:E1A0F00E                 MOV PC, LR
00001034:E8BD4030        MULFun:  LDMFD R13!, { R4, R5,LR}
00001038:E0000594                 MUL R0, R4, R5
0000103C:E1A0F00E                 MOV PC, LR

                          .DATA
00001044:                A:     .WORD  0
```



```
000053AC:81818181
000053B0:81818181
000053B4:81818181
000053B8:81818181
000053BC:81818181
000053C0:81818181
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
00005420:81818181
00005424:81818181
00005428:81818181
0000542C:81818181
```

Microprocessor & Computer Architecture (µpCA)

Nested Procedure Call: MUL(ADD(a,b),c)



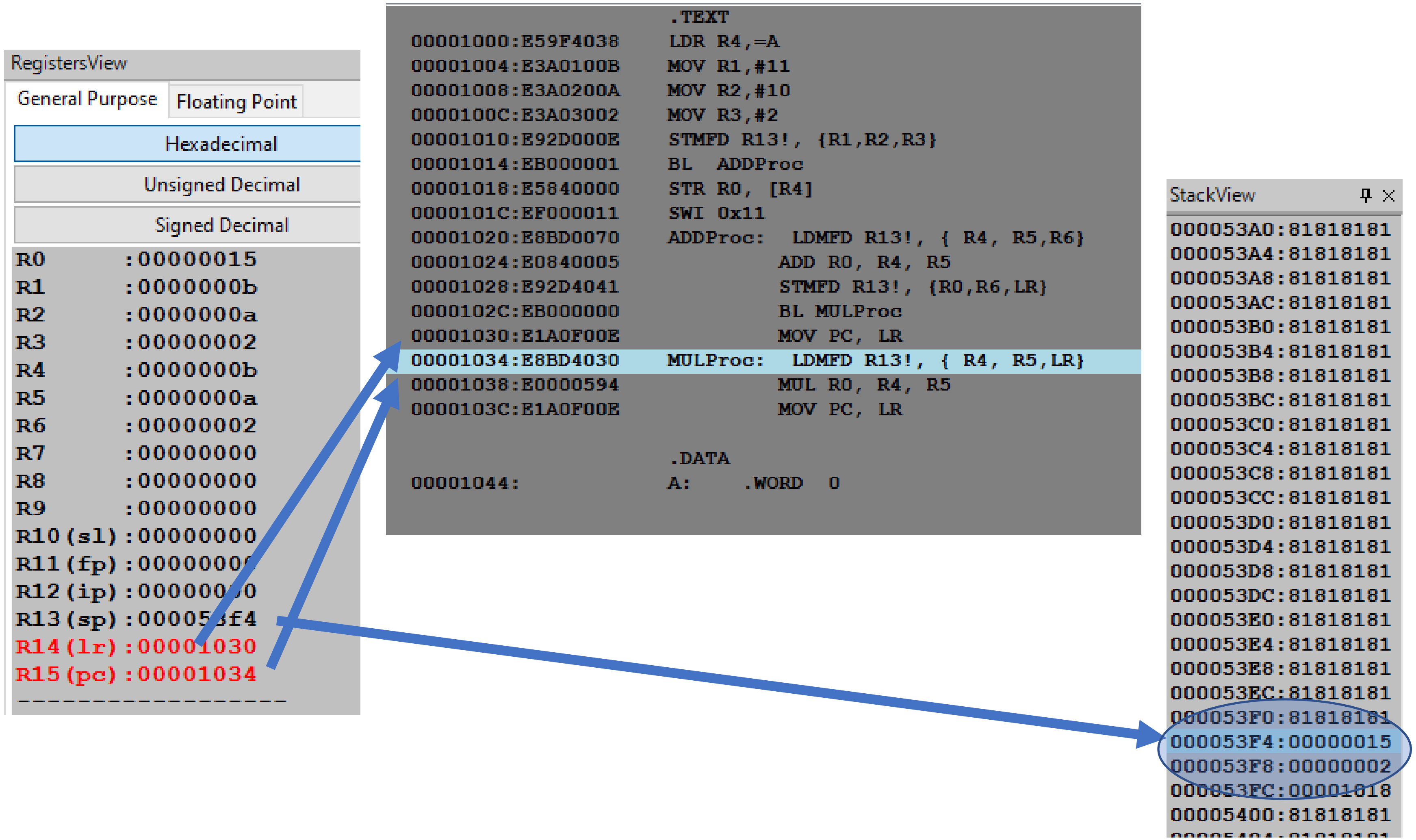
```
R0       :00000000
R1       :0000000b
R2       :0000000a
R3       :00000002
R4       :00001044
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :000053f4
R14(lr)  :00001018
R15(pc)  :00001020
```

```
                        .TEXT
00001000:E59F4038     LDR R4,=A
00001004:E3A0100B     MOV R1,#11
00001008:E3A0200A     MOV R2,#10
0000100C:E3A03002     MOV R3,#2
00001010:E92D000E     STMFD R13!, {R1,R2,R3}
00001014:EB000001     BL  ADDProc
00001018:E5840000     STR R0, [R4]
0000101C:EF000011     SWI 0x11
00001020:E8BD0070     ADDProc:  LDMFD R13!, { R4, R5,R6}
00001024:E0840005               ADD R0, R4, R5
00001028:E92D4041               STMFD R13!, {R0,R6,LR}
0000102C:EB000000               BL MULProc
00001030:E1A0F00E               MOV PC, LR
00001034:E8BD4030     MULProc:  LDMFD R13!, { R4, R5,LR}
00001038:E0000594               MUL R0, R4, R5
0000103C:E1A0F00E               MOV PC, LR

                        .DATA
00001044:             A:     .WORD  0
```

```
000053A0:81818181
000053A4:81818181
000053A8:81818181
000053AC:81818181
000053B0:81818181
000053B4:81818181
000053B8:81818181
000053BC:81818181
000053C0:81818181
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:0000000B
000053F8:0000000A
000053FC:00000002
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
```

Microprocessor & Computer Architecture (μpCA)

Nested Procedure Call: MUL(ADD(a,b),c)



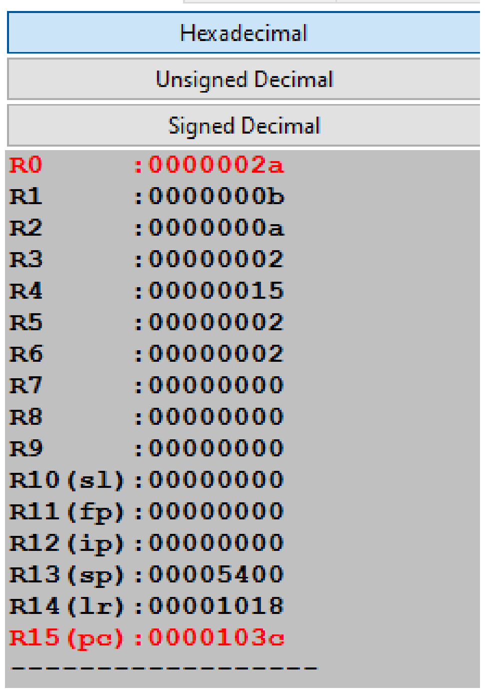
```
R0       :00000015
R1       :0000000b
R2       :0000000a
R3       :00000002
R4       :0000000b
R5       :0000000a
R6       :00000002
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :000053f4
R14(lr)  :00001030
R15(pc)  :00001034
------------------
```

```
                        .TEXT
00001000:E59F4038     LDR R4,=A
00001004:E3A0100B     MOV R1,#11
00001008:E3A0200A     MOV R2,#10
0000100C:E3A03002     MOV R3,#2
00001010:E92D000E     STMFD R13!, {R1,R2,R3}
00001014:EB000001     BL  ADDProc
00001018:E5840000     STR R0, [R4]
0000101C:EF000011     SWI 0x11
00001020:E8BD0070     ADDProc:  LDMFD R13!, { R4, R5,R6}
00001024:E0840005               ADD R0, R4, R5
00001028:E92D4041               STMFD R13!, {R0,R6,LR}
0000102C:EB000000               BL MULProc
00001030:E1A0F00E               MOV PC, LR
00001034:E8BD4030     MULProc:  LDMFD R13!, { R4, R5,LR}
00001038:E0000594               MUL R0, R4, R5
0000103C:E1A0F00E               MOV PC, LR

                        .DATA
00001044:             A:     .WORD  0
```

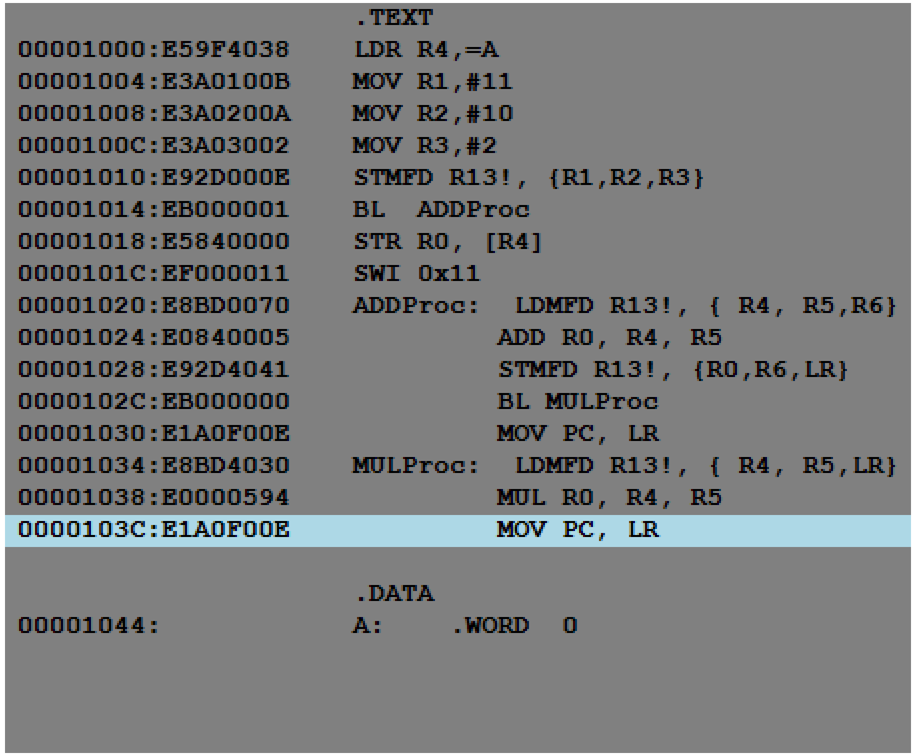
```
000053A0:81818181
000053A4:81818181
000053A8:81818181
000053AC:81818181
000053B0:81818181
000053B4:81818181
000053B8:81818181
000053BC:81818181
000053C0:81818181
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:00000015
000053F8:00000002
000053FC:00001018
00005400:81818181
```

Microprocessor & Computer Architecture (μpCA)

Nested Procedure Call: MUL(ADD(a,b),c)



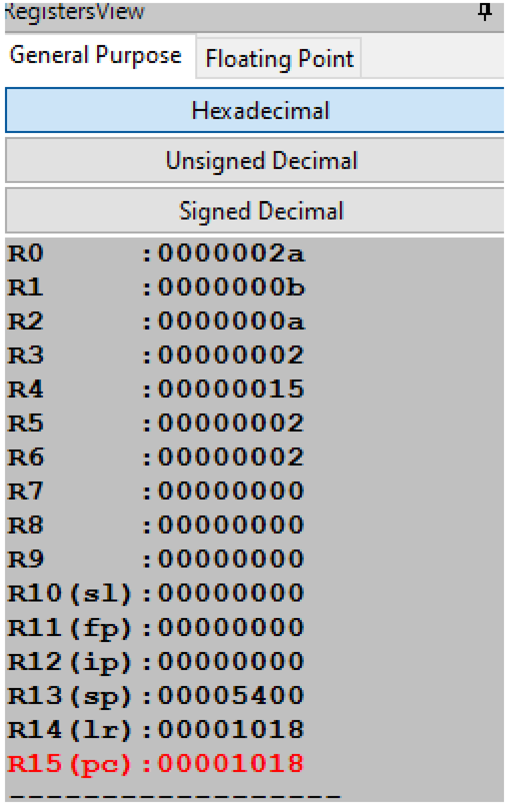
```
R0       :0000002a
R1       :0000000b
R2       :0000000a
R3       :00000002
R4       :00000015
R5       :00000002
R6       :00000002
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00005400
R14(lr)  :00001018
R15(pc)  :0000103c
-------------------
```



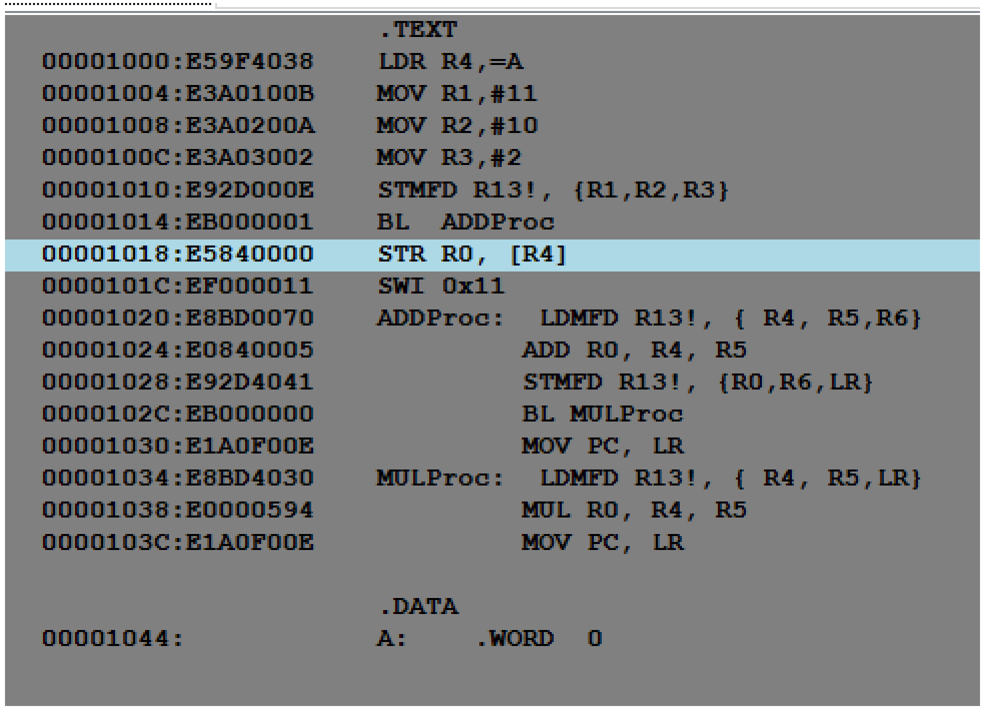
```
                     .TEXT
00001000:E59F4038    LDR R4,=A
00001004:E3A0100B    MOV R1,#11
00001008:E3A0200A    MOV R2,#10
0000100C:E3A03002    MOV R3,#2
00001010:E92D000E    STMFD R13!, {R1,R2,R3}
00001014:EB000001    BL  ADDProc
00001018:E5840000    STR R0, [R4]
0000101C:EF000011    SWI 0x11
00001020:E8BD0070    ADDProc:  LDMFD R13!, { R4, R5,R6}
00001024:E0840005              ADD R0, R4, R5
00001028:E92D4041              STMFD R13!, {R0,R6,LR}
0000102C:EB000000              BL MULProc
00001030:E1A0F00E              MOV PC, LR
00001034:E8BD4030    MULProc:  LDMFD R13!, { R4, R5,LR}
00001038:E0000594              MUL R0, R4, R5
0000103C:E1A0F00E              MOV PC, LR

                     .DATA
00001044:            A:     .WORD  0
```

Microprocessor & Computer Architecture (μpCA)

Nested Procedure Call: MUL(ADD(a,b),c)



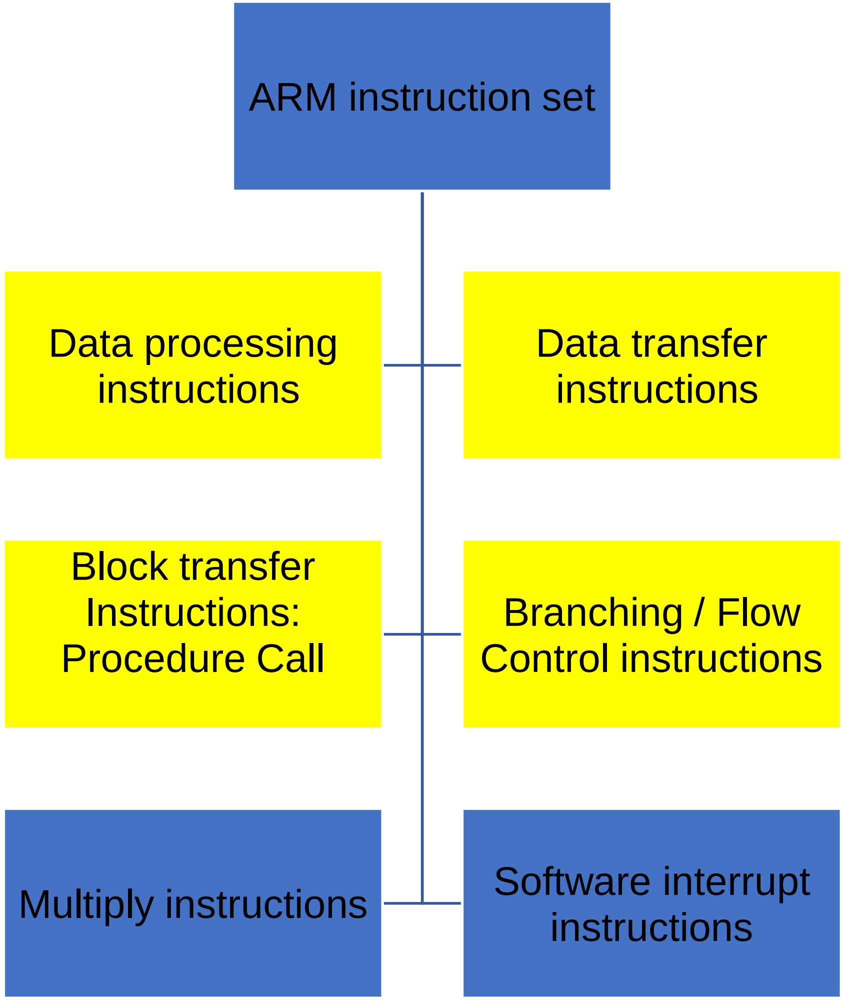
```
R0       :0000002a
R1       :0000000b
R2       :0000000a
R3       :00000002
R4       :00000015
R5       :00000002
R6       :00000002
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00005400
R14(lr)  :00001018
R15(pc)  :00001018
-------------------
```



```
                      .TEXT
00001000:E59F4038     LDR R4,=A
00001004:E3A0100B     MOV R1,#11
00001008:E3A0200A     MOV R2,#10
0000100C:E3A03002     MOV R3,#2
00001010:E92D000E     STMFD R13!, {R1,R2,R3}
00001014:EB000001     BL  ADDProc
00001018:E5840000     STR R0, [R4]
0000101C:EF000011     SWI 0x11
00001020:E8BD0070     ADDProc:  LDMFD R13!, { R4, R5,R6}
00001024:E0840005              ADD R0, R4, R5
00001028:E92D4041              STMFD R13!, {R0,R6,LR}
0000102C:EB000000              BL MULProc
00001030:E1A0F00E              MOV PC, LR
00001034:E8BD4030     MULProc:  LDMFD R13!, { R4, R5,LR}
00001038:E0000594              MUL R0, R4, R5
0000103C:E1A0F00E              MOV PC, LR

                      .DATA
00001044:             A:     .WORD  0
```

Microprocessor & Computer Architecture (μpCA)

NEXT Session : Multiplication


ARM instruction set

Data processing instructions

Data transfer instructions

Block transfer Instructions: Procedure Call

Branching / Flow Control instructions

Multiply instructions

Software interrupt instructions

THANK YOU

Dr. D. C. Kiran

Department of Computer Science and Engineering

dckiran@pes.edu

9829935135