

Tutorial 9: ARM Stack Frame

Computer Science Department

CS2208b: Introduction to Computer Organization and Architecture

Winter 2019

Instructor: Mahmoud R. El-Sakka

Office: MC-419

Email: elsakka@csd.uwo.ca

Phone: 519-661-2111 x86996

ARM Stack Frame

```
AREA TestProg, CODE, READONLY
```

```
ENTRY                ;This is the calling environment
```

```
Main  ADR    sp,Stack        ;set up r13 as the stack pointer
```

```
      MOV    r0,#124          ;set up a dummy parameter in r0
```

```
      MOV    fp,#123          ;set up dummy frame pointer
```

```
      STR    r0,[sp,#-4]!    ;push the parameter
```

```
      BL     Sub                ;call the subroutine
```

```
      LDR    r1,[sp],#4      ;pop the parameter
```

```
Loop  B      Loop              ;wait here (endless loop)
```

ARM Stack Frame

Sub STMFD **sp!**, {fp,lr} ;push frame-pointer and link-register
 MOV **fp**,sp ;frame pointer at the bottom of the frame
 SUB **sp**,sp,#4 ;create the stack frame (one word)

 LDR **r2**,[fp,#8] ;get the pushed parameter
 ADD **r2**,r2,#120 ;do a dummy operation on the parameter
 STR r2,[**fp**,#-4] ;store it in the stack frame

 ADD **sp**,sp,#4 ;clean up the stack frame
 LDMFD **sp!**, {**fp**,**pc**} ;restore frame pointer and return

The
body
of an
FD
stack

DCD	0x0000	;clear memory
DCD	0x0000	
DCD	0x0000	
DCD	0x0000	
Stack DCD	0x0000	;start of the stack

To be used as a local variable

To be used to push fp (i.e., R11)

To be used to push lr (i.e., R14)

To be used to push the parameter

END

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\ARM examples\chapter 1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

outNxt

Registers

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

Disassembly

```

3: Main  ADR  sp,Stack      ;set up r13 as the stack pointer
0x00000000 E28FD044 ADD     R13,PC,#0x00000044
4:      MOV  r0,#124        ;set up a dummy parameter
0x00000004 E3A0007C MOV     R0,#0x0000007C
5:      MOV  fp,#123        ;dummy frame pointer
0x00000008 E3A0B07B MOV     R11,#0x0000007B
6:      STR  r0,[sp,#-4]!    ;push the parameter
0x0000000C E52D0004 STR     R0,[R13,#-0x0004]!
7:      BL   Sub            ;call the subroutine
0x00000010 EB000001 BL      0x0000001C
8:      LDR  r1,[sp],#4      ;retrieve the data
0x00000014 E49D1004 LDR     R1,[R13],#0x0004
9: Loop  B    Loop          ;wait here (endless loop)
10:
0x00000018 EAffFFFF B      0x00000018
11: Sub  STMFD sp!,{fp,lr}   ;push frame-pointer and link-register
0x0000001C E92D4800 STMDB   R13!,{R11,R14}
12:      MOV  fp,sp          ;frame pointer at the bottom of the frame
0x00000020 E1A0B00D MOV     R11,R13
13:      SUB  sp,sp,#4        ;create the stack frame (one word)
0x00000024 E24DD004 SUB     R13,R13,#0x00000004
14:      LDR  r2,[fp,#8]      ;get the pushed parameter

```

Memory 1

Address: 0

0x00000000:	E2 8F D0 44 E3 A0 00 7C E3 A0 B0 7B E5 2D 00 04
0x00000010:	EB 00 00 01 E4 9D 10 04 EA FF FF FE E9 2D 48 00
0x00000020:	E1 A0 B0 0D E2 4D D0 04 E5 9B 20 08 E2 82 20 78
0x00000030:	E5 0B 20 04 E2 8D D0 04 E8 BD 88 00 00 00 00 00
0x00000040:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000050:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Call Stack + Locals Memory 1

ch1.s

```

1  AREA TestProg, CODE, READONLY
2  ENTRY
3  Main ADR  sp,Stack      ;set up r13 as the stack pointer
4      MOV  r0,#124        ;set up a dummy parameter
5      MOV  fp,#123        ;dummy frame pointer
6      STR  r0,[sp,#-4]!    ;push the parameter
7      BL   Sub            ;call the subroutine
8      LDR  r1,[sp],#4      ;retrieve the data
9  Loop B    Loop          ;wait here (endless loop)
10
11 Sub  STMFD sp!,{fp,lr}   ;push frame-pointer and link-register
12      MOV  fp,sp          ;frame pointer at the bottom of the frame
13      SUB  sp,sp,#4        ;create the stack frame (one word)
14      LDR  r2,[fp,#8]      ;get the pushed parameter
15      ADD  r2,r2,#120      ;do a dummy operation on the parameter
16      STR  r2,[fp,#-4]    ;store it in the stack frame
17      ADD  sp,sp,#4        ;clean up the stack frame
18      LDMFD sp!,{fp,pc}   ;restore frame pointer and return
19
20 DCD  0x0000             ;clear memory

```

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\ARM examples\chapter 1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Disassembly

Register	Value
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x0000004C
R14 (LR)	0x00000000
R15 (PC)	0x00000004
CPSR	0x000000D3
SPSR	0x00000000

3: Main ADR sp,Stack ;set up r13 as the stack pointer
 0x00000000 E28FD044 ADD R13,PC,#0x00000044
 4: MOV r0,#124 ;set up a dummy parameter
 0x00000004 E3A0007C MOV R0,#0x0000007C
 5: MOV fp,#123 ;dummy frame pointer
 0x00000008 E3A0B07B MOV R11,#0x0000007B
 6: STR r0,[sp,#-4]! ;push the parameter
 0x0000000C E52D0004 STR R0,[R13,#-0x0004]!
 7: BL Sub ;call the subroutine
 0x00000010 EB000001 BL 0x0000001C
 8: LDR r1,[sp],#4 ;retrieve the data
 0x00000014 E49D1004 LDR R1,[R13],#0x0004
 9: Loop B Loop ;wait here (endless loop)
 10:
 0x00000018 EAffFFFF B 0x00000018
 11: Sub STMFd sp!,{fp,lr} ;push frame-pointer and link-register
 0x0000001C E92D4800 STMDB R13!,{R11,R14}
 12: MOV fp,sp ;frame pointer at the bottom of the frame
 0x00000020 E1A0B00D MOV R11,R13
 13: SUB sp,sp,#4 ;create the stack frame (one word)
 0x00000024 E24DD004 SUB R13,R13,#0x00000004
 14: LDR r2,[fp,#8] ;get the pushed parameter

Memory 1

Address: 0

0x00000000:	E2 8F D0 44 E3 A0 00 7C E3 A0 B0 7B E5 2D 00 04
0x00000010:	EB 00 00 01 E4 9D 10 04 EA FF FF FE E9 2D 48 00
0x00000020:	E1 A0 B0 0D E2 4D D0 04 E5 9B 20 08 E2 82 20 78
0x00000030:	E5 0B 20 04 E2 8D D0 04 E8 BD 88 00 00 00 00 00
0x00000040:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000050:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

ch1.s

```

1 AREA TestProg, CODE, READONLY
2 ENTRY
3 Main ADR sp,Stack ;set up r13 as the stack pointer
4 MOV r0,#124 ;set up a dummy parameter
5 MOV fp,#123 ;dummy frame pointer
6 STR r0,[sp,#-4]! ;push the parameter
7 BL Sub ;call the subroutine
8 LDR r1,[sp],#4 ;retrieve the data
9 Loop B Loop ;wait here (endless loop)
10
11 Sub STMFd sp!,{fp,lr} ;push frame-pointer and link-register
12 MOV fp,sp ;frame pointer at the bottom of the frame
13 SUB sp,sp,#4 ;create the stack frame (one word)
14 LDR r2,[fp,#8] ;get the pushed parameter
15 ADD r2,r2,#120 ;do a dummy operation on the parameter
16 STR r2,[fp,#-4] ;store it in the stack frame
17 ADD sp,sp,#4 ;clean up the stack frame
18 LDMFD sp!,{fp,pc} ;restore frame pointer and return
19
20 DCD 0x0000 ;clear memory
  
```

3rd word in the stack

2nd word in the stack

1st word in the stack

4th word in the stack

location 0x4C, the top of the stack

5

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\ARM examples\chapter 1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

outNxt

Registers Disassembly

Register	Value
Current	
R0	0x0000007C
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x0000004C
R14 (LR)	0x00000000
R15 (PC)	0x00000008
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000008
Mode	Supervisor
States	2
Sec	0.00000000

```

3: Main  ADR  sp,Stack      ;set up r13 as the stack pointer
0x00000000 E28FD044 ADD     R13,PC,#0x00000044
4:      MOV  r0,#124        ;set up a dummy parameter
0x00000004 E3A0007C MOV     R0,#0x0000007C
5:      MOV  fp,#123        ;dummy frame pointer
0x00000008 E3A0B07B MOV     R11,#0x0000007B
6:      STR  r0,[sp,#-4]!    ;push the parameter
0x0000000C E52D0004 STR     R0,[R13,#-0x0004]!
7:      BL   Sub             ;call the subroutine
0x00000010 EB000001 BL      0x0000001C
8:      LDR  r1,[sp],#4      ;retrieve the data
0x00000014 E49D1004 LDR     R1,[R13],#0x0004
9: Loop  B     Loop         ;wait here (endless loop)
10:
0x00000018 EAffFFFF B      0x00000018
11: Sub  STMFD sp!,{fp,lr}   ;push frame-pointer and link-register
0x0000001C E92D4800 STMDB   R13!,{R11,R14}
12:      MOV  fp,sp          ;frame pointer at the bottom of the frame
0x00000020 E1A0B00D MOV     R11,R13
13:      SUB  sp,sp,#4        ;create the stack frame (one word)
0x00000024 E24DD004 SUB     R13,R13,#0x00000004
14:      LDR  r2,[fp,#8]      ;get the pushed parameter

```

Memory 1

Address: 0

```

0x00000000: E2 8F D0 44 E3 A0 00 7C E3 A0 B0 7B E5 2D 00 04
0x00000010: EB 00 00 01 E4 9D 10 04 EA FF FF FE E9 2D 48 00
0x00000020: E1 A0 B0 0D E2 4D D0 04 E5 9B 20 08 E2 82 20 78
0x00000030: E5 0B 20 04 E2 8D D0 04 E8 BD 88 00 00 00 00 00
0x00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

Call Stack + Locals Memory 1

ch1.s

```

1  AREA TestProg, CODE, READONLY
2  ENTRY
3  Main  ADR  sp,Stack      ;set up r13 as the stack pointer
4      MOV  r0,#124        ;set up a dummy parameter
5      MOV  fp,#123        ;dummy frame pointer
6      STR  r0,[sp,#-4]!    ;push the parameter
7      BL   Sub             ;call the subroutine
8      LDR  r1,[sp],#4      ;retrieve the data
9  Loop  B     Loop         ;wait here (endless loop)
10
11  Sub  STMFD sp!,{fp,lr}   ;push frame-pointer and link-register
12      MOV  fp,sp          ;frame pointer at the bottom of the frame
13      SUB  sp,sp,#4        ;create the stack frame (one word)
14      LDR  r2,[fp,#8]      ;get the pushed parameter
15      ADD  r2,r2,#120      ;do a dummy operation on the parameter
16      STR  r2,[fp,#-4]    ;store it in the stack frame
17      ADD  sp,sp,#4        ;clean up the stack frame
18      LDMFD sp!,{fp,pc}   ;restore frame pointer and return
19
20      DCD  0x0000         ;clear memory

```


C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\ARM examples\chapter 1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

outNxt

Registers Disassembly

Register	Value
R0	0x0000007C
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x0000007B
R12	0x00000000
R13 (SP)	0x0000004C
R14 (LR)	0x00000000
R15 (PC)	0x0000000C
CPSR	0x000000D3
SPSR	0x00000000

User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC \$ 0x0000000C
Mode Supervisor
States 3
Sec 0.00000000

```

3: Main  ADR  sp,Stack      ;set up r13 as the stack pointer
0x00000000 E28FD044 ADD    R13,PC,#0x00000044
4:      MOV  r0,#124        ;set up a dummy parameter
0x00000004 E3A0007C MOV    R0,#0x0000007C
5:      MOV  fp,#123        ;dummy frame pointer
0x00000008 E3A0B07B MOV    R11,#0x0000007B
6:      STR  r0,[sp,#-4]!    ;push the parameter
0x0000000C E52D0004 STR    R0,[R13,#-0x00004]!
7:      BL   Sub            ;call the subroutine
0x00000010 EB000001 BL     0x0000001C
8:      LDR  r1,[sp],#4      ;retrieve the data
0x00000014 E49D1004 LDR    R1,[R13],#0x00004
9: Loop  B    Loop          ;wait here (endless loop)
10:
0x00000018 EAffffff B     0x00000018
11: Sub  STMFD sp!,{fp,lr}   ;push frame-pointer and link-register
0x0000001C E92D4800 STMDB  R13!,{R11,R14}
12:      MOV  fp,sp          ;frame pointer at the bottom of the frame
0x00000020 E1A0B00D MOV    R11,R13
13:      SUB  sp,sp,#4        ;create the stack frame (one word)
0x00000024 E24DD004 SUB    R13,R13,#0x00000004
14:      LDR  r2,[fp,#8]     ;get the pushed parameter

```

Memory 1

Address: 0

0x00000000:	E2 8F D0 44 E3 A0 00 7C E3 A0 B0 7B E5 2D 00 04
0x00000010:	EB 00 00 01 E4 9D 10 04 EA FF FF FE E9 2D 48 00
0x00000020:	E1 A0 B0 0D E2 4D D0 04 E5 9B 20 08 E2 82 20 78
0x00000030:	E5 0B 20 04 E2 8D D0 04 E8 BD 88 00 00 00 00 00
0x00000040:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000050:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Call Stack + Locals Memory 1

ch1.s

```

1  AREA TestProg, CODE, READONLY
2  ENTRY
3  Main  ADR  sp,Stack      ;set up r13 as the stack pointer
4        MOV  r0,#124        ;set up a dummy parameter
5        MOV  fp,#123        ;dummy frame pointer
6        STR  r0,[sp,#-4]!    ;push the parameter
7        BL   Sub            ;call the subroutine
8        LDR  r1,[sp],#4      ;retrieve the data
9  Loop  B    Loop          ;wait here (endless loop)
10
11  Sub  STMFD sp!,{fp,lr}   ;push frame-pointer and link-register
12      MOV  fp,sp          ;frame pointer at the bottom of the frame
13      SUB  sp,sp,#4        ;create the stack frame (one word)
14      LDR  r2,[fp,#8]     ;get the pushed parameter
15      ADD  r2,r2,#120      ;do a dummy operation on the parameter
16      STR  r2,[fp,#-4]    ;store it in the stack frame
17      ADD  sp,sp,#4        ;clean up the stack frame
18      LDMFD sp!,{fp,pc}   ;restore frame pointer and return
19
20      DCD  0x0000         ;clear memory

```

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\ARM examples\chapter 1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

outNxt

Registers Disassembly

Register	Value
R0	0x0000007C
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x0000007B
R12	0x00000000
R13 (SP)	0x00000048
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000

User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC \$ 0x00000010
Mode Supervisor
States 5
Sec 0.00000000

```

3: Main  ADR  sp,Stack      ;set up r13 as the stack pointer
0x00000000 E28FD044 ADD    R13,PC,#0x00000044
4:      MOV  r0,#124        ;set up a dummy parameter
0x00000004 E3A0007C MOV    R0,#0x0000007C
5:      MOV  fp,#123        ;dummy frame pointer
0x00000008 E3A0B07B MOV    R11,#0x0000007B
6:      STR  r0,[sp,#-4]!    ;push the parameter
0x0000000C E52D0004 STR    R0,[R13,#-0x0004]!
7:      BL   Sub            ;call the subroutine
0x00000010 EB000001 BL     0x0000001C
8:      LDR  r1,[sp],#4      ;retrieve the data
0x00000014 E49D1004 LDR    R1,[R13],#0x0004
9: Loop  B    Loop          ;wait here (endless loop)
10:
0x00000018 EAffffff B     0x00000018
11: Sub  STMFD sp!,{fp,lr}   ;push frame-pointer and link-register
0x0000001C E92D4800 STMDB  R13!,{R11,R14}
12:      MOV  fp,sp          ;frame pointer at the bottom of the frame
0x00000020 E1A0B00D MOV    R11,R13
13:      SUB  sp,sp,#4        ;create the stack frame (one word)
0x00000024 E24DD004 SUB    R13,R13,#0x00000004
14:      LDR  r2,[fp,#8]      ;get the pushed parameter

```

Memory 1

Address: 0

0x00000000:	E2 8F D0 44 E3 A0 00 7C E3 A0 B0 7B E5 2D 00 04
0x00000010:	EB 00 00 01 E4 9D 10 04 EA FF FF FE E9 2D 48 00
0x00000020:	E1 A0 B0 0D E2 4D D0 04 E5 9B 20 08 E2 82 20 78
0x00000030:	E5 0B 20 04 E2 8D D0 04 E8 BD 88 00 00 00 00 00
0x00000040:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000050:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Call Stack + Locals Memory 1

ch1.s

```

1  AREA TestProg, CODE, READONLY
2  ENTRY
3  Main  ADR  sp,Stack      ;set up r13 as the stack pointer
4        MOV  r0,#124        ;set up a dummy parameter
5        MOV  fp,#123        ;dummy frame pointer
6        STR  r0,[sp,#-4]!    ;push the parameter
7        BL   Sub            ;call the subroutine
8        LDR  r1,[sp],#4      ;retrieve the data
9  Loop  B    Loop          ;wait here (endless loop)
10
11  Sub  STMFD sp!,{fp,lr}   ;push frame-pointer and link-register
12      MOV  fp,sp          ;frame pointer at the bottom of the frame
13      SUB  sp,sp,#4        ;create the stack frame (one word)
14      LDR  r2,[fp,#8]      ;get the pushed parameter
15      ADD  r2,r2,#120      ;do a dummy operation on the parameter
16      STR  r2,[fp,#-4]    ;store it in the stack frame
17      ADD  sp,sp,#4        ;clean up the stack frame
18      LDMFD sp!,{fp,pc}   ;restore frame pointer and return
19
20      DCD  0x0000          ;clear memory

```


C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\ARM examples\chapter 1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Disassembly

Register	Value
Current	
R0	0x0000007C
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x0000007B
R12	0x00000000
R13 (SP)	0x00000048
R14 (LR)	0x00000014
R15 (PC)	0x0000001C
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x0000001C
Mode	Supervisor
States	8
Sec	0.00000000

3: Main ADR sp,Stack ;set up r13 as the stack pointer
 0x00000000 E28FD044 ADD R13,PC,#0x00000044
 4: MOV r0,#124 ;set up a dummy parameter
 0x00000004 E3A0007C MOV R0,#0x0000007C
 5: MOV fp,#123 ;dummy frame pointer
 0x00000008 E3A0B07B MOV R11,#0x0000007B
 6: STR r0,[sp,#-4]! ;push the parameter
 0x0000000C E52D0004 STR R0,[R13,#-0x0004]!
 7: BL Sub ;call the subroutine
 0x00000010 EB000001 BL 0x0000001C
 8: LDR r1,[sp],#4 ;retrieve the data
 0x00000014 E49D1004 LDR R1,[R13],#0x0004
 9: Loop B Loop ;wait here (endless loop)
 10:
 0x00000018 EAffffff B 0x00000018
 11: Sub STMFDP sp!,{fp,lr} ;push frame-pointer and link-register
 0x0000001C E92D4800 STMDB R13!,{R11,R14}
 12: MOV fp,sp ;frame pointer at the bottom of the frame
 0x00000020 E1A0B00D MOV R11,R13
 13: SUB sp,sp,#4 ;create the stack frame (one word)
 0x00000024 E24DD004 SUB R13,R13,#0x00000004
 14: LDR r2,[fp,#8] ;get the pushed parameter

Memory 1
 Address: 0
 0x00000000: E2 8F D0 44 E3 A0 00 7C E3 A0 B0 7B E5 2D 00 04
 0x00000010: EB 00 00 01 E4 9D 10 04 EA FF FF FE E9 2D 48 00
 0x00000020: E1 A0 B0 0D E2 4D D0 04 E5 9B 20 08 E2 82 20 78
 0x00000030: E5 0B 20 04 E2 8D D0 04 E8 BD 88 00 00 00 00 00
 0x00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

ch1.s

```

1 AREA TestProg, CODE, READONLY
2 ENTRY
3 Main ADR sp,Stack ;set up r13 as the stack pointer
4 MOV r0,#124 ;set up a dummy parameter
5 MOV fp,#123 ;dummy frame pointer
6 STR r0,[sp,#-4]! ;push the parameter
7 BL Sub ;call the subroutine
8 LDR r1,[sp],#4 ;retrieve the data
9 Loop B Loop ;wait here (endless loop)
10
11 Sub STMFDP sp!,{fp,lr} ;push frame-pointer and link-register
12 MOV fp,sp ;frame pointer at the bottom of the frame
13 SUB sp,sp,#4 ;create the stack frame (one word)
14 LDR r2,[fp,#8] ;get the pushed parameter
15 ADD r2,r2,#120 ;do a dummy operation on the parameter
16 STR r2,[fp,#-4] ;store it in the stack frame
17 ADD sp,sp,#4 ;clean up the stack frame
18 LDMFD sp!,{fp,pc} ;restore frame pointer and return
19
20 DCD 0x0000 ;clear memory
  
```

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\ARM examples\chapter 1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Disassembly

Register	Value
R0	0x0000007C
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x0000007B
R12	0x00000000
R13 (SP)	0x00000040
R14 (LR)	0x00000014
R15 (PC)	0x00000020
CPSR	0x000000D3
SPSR	0x00000000

3: Main ADR sp,Stack ;set up r13 as the stack pointer
 0x00000000 E28FD044 ADD R13,PC,#0x00000044
 4: MOV r0,#124 ;set up a dummy parameter
 0x00000004 E3A0007C MOV R0,#0x0000007C
 5: MOV fp,#123 ;dummy frame pointer
 0x00000008 E3A0B07B MOV R11,#0x0000007B
 6: STR r0,[sp,#-4]! ;push the parameter
 0x0000000C E52D0004 STR R0,[R13,#-0x0004]!
 7: BL Sub ;call the subroutine
 0x00000010 EB000001 BL 0x0000001C
 8: LDR r1,[sp],#4 ;retrieve the data
 0x00000014 E49D1004 LDR R1,[R13],#0x0004
 9: Loop B Loop ;wait here (endless loop)
 10:
 0x00000018 EAffFFFF B 0x00000018
 11: Sub STMFd sp!,{fp,lr} ;push frame-pointer and link-register
 0x0000001C E92D4800 STMDB R13!,{R11,R14}
 12: MOV fp,sp ;frame pointer at the bottom of the frame
 0x00000020 E1A0B00D MOV R11,R13
 13: SUB sp,sp,#4 ;create the stack frame (one word)
 0x00000024 E24DD004 SUB R13,R13,#0x00000004
 14: LDR r2,[fp,#8] ;get the pushed parameter

Memory 1
 Address: 0
 0x00000000: E2 8F D0 44 E3 A0 00 7C E3 A0 B0 7B E5 2D 00 04
 0x00000010: EB 00 00 01 E4 9D 10 04 EA FF FF FE E9 2D 48 00
 0x00000020: E1 A0 B0 0D E2 4D D0 04 E5 9B 20 08 E2 82 20 78
 0x00000030: E5 0B 20 04 E2 8D D0 04 E8 BD 88 00 00 00 00 00
 0x00000040: 00 00 00 7E 00 00 00 14 00 00 00 7C 00 00 00 00
 0x00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

ch1.s
 1 AREA TestProg, CODE, READONLY
 2 ENTRY
 3 Main ADR sp,Stack ;set up r13 as the stack pointer
 4 MOV r0,#124 ;set up a dummy parameter
 5 MOV fp,#123 ;dummy frame pointer
 6 STR r0,[sp,#-4]! ;push the parameter
 7 BL Sub ;call the subroutine
 8 LDR r1,[sp],#4 ;retrieve the data
 9 Loop B Loop ;wait here (endless loop)
 10
 11 Sub STMFd sp!,{fp,lr} ;push frame-pointer and link-register
 12 MOV fp,sp ;frame pointer at the bottom of the frame
 13 SUB sp,sp,#4 ;create the stack frame (one word)
 14 LDR r2,[fp,#8] ;get the pushed parameter
 15 ADD r2,r2,#120 ;do a dummy operation on the parameter
 16 STR r2,[fp,#-4] ;store it in the stack frame
 17 ADD sp,sp,#4 ;clean up the stack frame
 18 LDMFD sp!,{fp,pc} ;restore frame pointer and return
 19
 20 DCD 0x0000 ;clear memory

location 0x40, the current SP value

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\ARM examples\chapter 1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Disassembly

Register	Value
R0	0x0000007C
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000040
R12	0x00000000
R13 (SP)	0x00000040
R14 (LR)	0x00000014
R15 (PC)	0x00000024
CPSR	0x000000D3
SPSR	0x00000000

User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC \$ 0x00000024
Mode Supervisor
States 12
Sec 0.00000000

```

0x00000000 E28FD044 ADD R13,PC,#0x00000044
4: MOV r0,#124 ;set up a dummy parameter
0x00000004 E3A0007C MOV R0,#0x0000007C
5: MOV fp,#123 ;dummy frame pointer
0x00000008 E3A0B07B MOV R11,#0x0000007B
6: STR r0,[sp,#-4]! ;push the parameter
0x0000000C E52D0004 STR R0,[R13,#-0x0004]!
7: BL Sub ;call the subroutine
0x00000010 EB000001 BL 0x0000001C
8: LDR r1,[sp],#4 ;retrieve the data
0x00000014 E49D1004 LDR R1,[R13],#0x0004
9: Loop B Loop ;wait here (endless loop)
10:
0x00000018 EAffFFFF B 0x00000018
11: Sub STMFD sp!,{fp,lr} ;push frame-pointer and link-register
0x0000001C E92D4800 STMDB R13!,{R11,R14}
12: MOV fp,sp ;frame pointer at the bottom of the frame
0x00000020 E1A0B00D MOV R11,R13
13: SUB sp,sp,#4 ;create the stack frame (one word)
0x00000024 E24DD004 SUB R13,R13,#0x00000004
14: LDR r2,[fp,#8] ;get the pushed parameter
0x00000028 E59B2008 LDR R2,[R11,#0x0008]

```

Memory 1

Address: 0

0x00000000:	E2 8F D0 44 E3 A0 00 7C E3 A0 B0 7B E5 2D 00 04
0x00000010:	EB 00 00 01 E4 9D 10 04 EA FF FF FE E9 2D 48 00
0x00000020:	E1 A0 B0 0D E2 4D D0 04 E5 9B 20 08 E2 82 20 78
0x00000030:	E5 0B 20 04 E2 8D D0 04 E8 BD 88 00 00 00 00 00
0x00000040:	00 00 00 7E 00 00 00 14 00 00 00 7C 00 00 00 00
0x00000050:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Call Stack + Locals

ch1.s

```

1 AREA TestProg, CODE, READONLY
2 ENTRY
3 Main ADR sp,Stack ;set up r13 as the stack pointer
4 MOV r0,#124 ;set up a dummy parameter
5 MOV fp,#123 ;dummy frame pointer
6 STR r0,[sp,#-4]! ;push the parameter
7 BL Sub ;call the subroutine
8 LDR r1,[sp],#4 ;retrieve the data
9 Loop B Loop ;wait here (endless loop)
10
11 Sub STMFD sp!,{fp,lr} ;push frame-pointer and link-register
12 MOV fp,sp ;frame pointer at the bottom of the frame
13 SUB sp,sp,#4 ;create the stack frame (one word)
14 LDR r2,[fp,#8] ;get the pushed parameter
15 ADD r2,r2,#120 ;do a dummy operation on the parameter
16 STR r2,[fp,#-4] ;store it in the stack frame
17 ADD sp,sp,#4 ;clean up the stack frame
18 LDMFD sp!,{fp,pc} ;restore frame pointer and return
19
20 DCD 0x0000 ;clear memory

```

location 0x40, the current FP value

location 0x40, the current SP value

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\ARM examples\chapter 1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

outNxt

Registers

Register	Value
R0	0x0000007C
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000040
R12	0x00000000
R13 (SP)	0x0000003C
R14 (LR)	0x00000014
R15 (PC)	0x00000028
CPSR	0x000000D3
SPSR	0x00000000

Disassembly

```

0x00000004 E3A0007C MOV R0,#0x0000007C
5: MOV fp,#123 ;dummy frame pointer
0x00000008 E3A0B07B MOV R11,#0x0000007B
6: STR r0,[sp,#-4]! ;push the parameter
0x0000000C E52D0004 STR R0,[R13,#-0x0004]!
7: BL Sub ;call the subroutine
0x00000010 EB000001 BL 0x0000001C
8: LDR r1,[sp],#4 ;retrieve the data
0x00000014 E49D1004 LDR R1,[R13],#0x0004
9: Loop B Loop ;wait here (endless loop)
10:
0x00000018 EAF00000 B 0x00000018
11: Sub STMFDP sp!,{fp,lr} ;push frame-pointer and link-register
0x0000001C E92D4800 STMDB R13!,{R11,R14}
12: MOV fp,sp ;frame pointer at the bottom of the frame
0x00000020 E1A0B00D MOV R11,R13
13: SUB sp,sp,#4 ;create the stack frame (one word)
0x00000024 E24DD004 SUB R13,R13,#0x00000004
14: LDR r2,[fp,#8] ;get the pushed parameter
0x00000028 E59B2008 LDR R2,[R11,#0x0008]
15: ADD r2,r2,#120 ;do a dummy operation on the parameter
0x0000002C E2822078 ADD R2,R2,#0x00000078

```

Memory 1

Address: 0

0x00000000:	E2 8F D0 44 E3 A0 00 7C E3 A0 B0 7B E5 2D 04
0x00000010:	EB 00 00 01 E4 9D 10 04 EA FF FF FE E9 7 48 00
0x00000020:	E1 A0 B0 0D E2 4D D0 04 E5 9B 20 08 E2 82 20 78
0x00000030:	E5 0B 20 04 E2 8D D0 04 E8 BD 88 00 00 00 00 00
0x00000040:	00 00 00 7B 00 00 00 14 00 00 00 7C 00 00 00 00
0x00000050:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

ch1.s

```

1 AREA TestProg, CODE, READONLY
2 ENTRY
3 Main ADR sp,Stack ;set up r13 as the stack pointer
4 MOV r0,#124 ;set up a dummy parameter
5 MOV fp,#123 ;dummy frame pointer
6 STR r0,[sp,#-4]! ;push the parameter
7 BL Sub ;call the subroutine
8 LDR r1,[sp],#4 ;retrieve the data
9 Loop B Loop ;wait here (endless loop)
10
11 Sub STMFDP sp!,{fp,lr} ;push frame-pointer and link-register
12 MOV fp,sp ;frame pointer at the bottom of the frame
13 SUB sp,sp,#4 ;create the stack frame (one word)
14 LDR r2,[fp,#8] ;get the pushed parameter
15 ADD r2,r2,#120 ;do a dummy operation on the parameter
16 STR r2,[fp,#-4] ;store it in the stack frame
17 ADD sp,sp,#4 ;clean up the stack frame
18 LDMFD sp!,{fp,pc} ;restore frame pointer and return
19
20 DCD 0x0000 ;clear memory

```

location
0x3C, the
current
SP value

location
0x40, the
current
FP value

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\ARM examples\chapter 1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

outNxt

Registers

Register	Value
Current	
R0	0x0000007C
R1	0x00000000
R2	0x0000007C
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000040
R12	0x00000000
R13 (SP)	0x0000003C
R14 (LR)	0x00000014
R15 (PC)	0x0000002C
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x0000002C
Mode	Supervisor
States	16
Sec	0.00000000

Disassembly

```

0x00000008 E3A0B07B MOV R11,#0x0000007B
6: STR r0,[sp,#-4]! ;push the parameter
0x0000000C E52D0004 STR R0,[R13,#-0x0004]!
7: BL Sub ;call the subroutine
0x00000010 EB000001 BL 0x0000001C
8: LDR r1,[sp],#4 ;retrieve the data
0x00000014 E49D1004 LDR R1,[R13],#0x0004
9: Loop B Loop ;wait here (endless loop)
10:
0x00000018 EAF0FFFE B 0x00000018
11: Sub STMFD sp!,{fp,lr} ;push frame-pointer and link-register
0x0000001C E92D4800 STMDB R13!,{R11,R14}
12: MOV fp,sp ;frame pointer at the bottom of the frame
0x00000020 E1A0B00D MOV R11,R13
13: SUB sp,sp,#4 ;create the stack frame (one word)
0x00000024 E24DD004 SUB R13,R13,#0x00000004
14: LDR r2,[fp,#8] ;get the pushed parameter
0x00000028 E59B2008 LDR R2,[R11,#0x0008]
15: ADD r2,r2,#120 ;do a dummy operation on the parameter
0x0000002C E2822078 ADD R2,R2,#0x00000078
16: STR r2,[fp,#-4] ;store it in the stack frame
0x00000030 E50B2004 STR R2,[R11,#-0x0004]

```

Memory 1

Address: 0

0x00000000:	E2 8F D0 44 E3 A0 00 7C E3 A0 B0 7B E5 2D 04
0x00000010:	EB 00 00 01 E4 9D 10 04 EA FF FF FE E9 7C 48 00
0x00000020:	E1 A0 B0 0D E2 4D D0 04 E5 9B 20 08 E2 82 20 78
0x00000030:	E5 0B 20 04 E2 8D D0 04 E8 BD 88 00 00 00 00 00
0x00000040:	00 00 00 7B 00 00 00 14 00 00 00 7C 00 00 00 00
0x00000050:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Call Stack + Locals

ch1.s

```

1 AREA TestProg, CODE, READONLY
2 ENTRY
3 Main ADR sp,Stack ;set up r13 as the stack pointer
4 MOV r0,#124 ;set up a dummy parameter
5 MOV fp,#123 ;dummy frame pointer
6 STR r0,[sp,#-4]! ;push the parameter
7 BL Sub ;call the subroutine
8 LDR r1,[sp],#4 ;retrieve the data
9 Loop B Loop ;wait here (endless loop)
10
11 Sub STMFD sp!,{fp,lr} ;push frame-pointer and link-register
12 MOV fp,sp ;frame pointer at the bottom of the frame
13 SUB sp,sp,#4 ;create the stack frame (one word)
14 LDR r2,[fp,#8] ;get the pushed parameter
15 ADD r2,r2,#120 ;do a dummy operation on the parameter
16 STR r2,[fp,#-4] ;store it in the stack frame
17 ADD sp,sp,#4 ;clean up the stack frame
18 LDMFD sp!,{fp,pc} ;restore frame pointer and return
19
20 DCD 0x0000 ;clear memory

```

location
0x3C, the
current
SP value

location
0x40, the
current
FP value

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\ARM examples\chapter 1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Disassembly

Register	Value
Current	
R0	0x0000007C
R1	0x00000000
R2	0x000000F4
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000040
R12	0x00000000
R13 (SP)	0x0000003C
R14 (LR)	0x00000014
R15 (PC)	0x00000030
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000030
Mode	Supervisor
States	17
Sec	0.00000000

```

0x0000000C E52D0004 STR R0,[R13,#-0x0004]!
7: BL Sub ;call the subroutine
0x00000010 EB000001 BL 0x0000001C
8: LDR r1,[sp],#4 ;retrieve the data
0x00000014 E49D1004 LDR R1,[R13],#0x0004
9: Loop B Loop ;wait here (endless loop)
10:
0x00000018 EAF00000 B 0x00000018
11: Sub STMFD sp!,{fp,lr} ;push frame-pointer and link-register
0x0000001C E92D4800 STMDB R13!,{R11,R14}
12: MOV fp,sp ;frame pointer at the bottom of the frame
0x00000020 E1A0B00D MOV R11,R13
13: SUB sp,sp,#4 ;create the stack frame (one word)
0x00000024 E24DD004 SUB R13,R13,#0x00000004
14: LDR r2,[fp,#8] ;get the pushed parameter
0x00000028 E59B2008 LDR R2,[R11,#0x0008]
15: ADD r2,r2,#120 ;do a dummy operation on the parameter
0x0000002C E2822078 ADD R2,R2,#0x00000078
16: STR r2,[fp,#-4] ;store it in the stack frame
0x00000030 E50B2004 STR R2,[R11,#-0x0004]
17: ADD sp,sp,#4 ;clean up the stack frame
0x00000034 E28DD004 ADD R13,R13,#0x00000004

```

Memory 1

Address: 0

0x00000000:	E2 8F D0 44 E3 A0 00 7C E3 A0 B0 7B E5 2D 04
0x00000010:	EB 00 00 01 E4 9D 10 04 EA FF FF FE E9 7C 48 00
0x00000020:	E1 A0 B0 0D E2 4D D0 04 E5 9B 20 08 E2 82 20 78
0x00000030:	E5 0B 20 04 E2 8D D0 04 E8 BD 88 00 00 00 00 00
0x00000040:	00 00 00 7B 00 00 00 14 00 00 00 7C 00 00 00 00
0x00000050:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Call Stack + Locals

ch1.s

```

1 AREA TestProg, CODE, READONLY
2 ENTRY
3 Main ADR sp,Stack ;set up r13 as the stack pointer
4 MOV r0,#124 ;set up a dummy parameter
5 MOV fp,#123 ;dummy frame pointer
6 STR r0,[sp,#-4]! ;push the parameter
7 BL Sub ;call the subroutine
8 LDR r1,[sp],#4 ;retrieve the data
9 Loop B Loop ;wait here (endless loop)
10
11 Sub STMFD sp!,{fp,lr} ;push frame-pointer and link-register
12 MOV fp,sp ;frame pointer at the bottom of the frame
13 SUB sp,sp,#4 ;create the stack frame (one word)
14 LDR r2,[fp,#8] ;get the pushed parameter
15 ADD r2,r2,#120 ;do a dummy operation on the parameter
16 STR r2,[fp,#-4] ;store it in the stack frame
17 ADD sp,sp,#4 ;clean up the stack frame
18 LDMFD sp!,{fp,pc} ;restore frame pointer and return
19
20 DCD 0x0000 ;clear memory

```

location
0x3C, the
current
SP value

location
0x40, the
current
FP value

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\ARM examples\chapter 1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Disassembly

Register	Value
Current	
R0	0x0000007C
R1	0x00000000
R2	0x000000F4
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000040
R12	0x00000000
R13 (SP)	0x0000003C
R14 (LR)	0x00000014
R15 (PC)	0x00000034
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000034
Mode	Supervisor
States	19
Sec	0.00000000

```

0x00000010 EB000001 BL 0x0000001C
8: LDR r1,[sp],#4 ;retrieve the data
0x00000014 E49D1004 LDR R1,[R13],#0x0004
9: Loop B Loop ;wait here (endless loop)
10:
0x00000018 EAffffff B 0x00000018
11: Sub STMFd sp!,{fp,lr} ;push frame-pointer and link-register
0x0000001C E92D4800 STMDB R13!,{R11,R14}
12: MOV fp,sp ;frame pointer at the bottom of the frame
0x00000020 E1A0B00D MOV R11,R13
13: SUB sp,sp,#4 ;create the stack frame (one word)
0x00000024 E24DD004 SUB R13,R13,#0x00000004
14: LDR r2,[fp,#8] ;get the pushed parameter
0x00000028 E59B2008 LDR R2,[R11,#0x00000008]
15: ADD r2,r2,#120 ;do a dummy operation on the parameter
0x0000002C E2822078 ADD R2,R2,#0x00000078
16: STR r2,[fp,#-4] ;store it in the stack frame
0x00000030 E50B2004 STR R2,[R11,#-0x00000004]
17: ADD sp,sp,#4 ;clean up the stack frame
0x00000034 E28DD004 ADD R13,R13,#0x00000004
18: LDMFD sp!,{fp,pc} ;restore frame pointer and return
0x00000038 E8BD8800 LDMIA R13!,{R11,PC}
  
```

Memory 1

Address: 0

0x00000000:	E2 8F D0 44 E3 A0 00 7C E3 A0 B0 7B E5 2D 04
0x00000010:	EB 00 00 01 E4 9D 10 04 EA FF FF FE E9 7 48 00
0x00000020:	E1 A0 B0 0D E2 4D D0 04 E5 9B 20 08 E2 82 20 78
0x00000030:	E5 0B 20 04 E2 8D D0 04 E8 BD 88 00 00 00 00 F4
0x00000040:	00 00 00 7B 00 00 00 14 00 00 00 7C 00 00 00 00
0x00000050:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Call Stack + Locals

ch1.s

```

1 AREA TestProg, CODE, READONLY
2 ENTRY
3 Main ADR sp,Stack ;set up r13 as the stack pointer
4 MOV r0,#124 ;set up a dummy parameter
5 MOV fp,#123 ;dummy frame pointer
6 STR r0,[sp,#-4]! ;push the parameter
7 BL Sub ;call the subroutine
8 LDR r1,[sp],#4 ;retrieve the data
9 Loop B Loop ;wait here (endless loop)
10
11 Sub STMFd sp!,{fp,lr} ;push frame-pointer and link-register
12 MOV fp,sp ;frame pointer at the bottom of the frame
13 SUB sp,sp,#4 ;create the stack frame (one word)
14 LDR r2,[fp,#8] ;get the pushed parameter
15 ADD r2,r2,#120 ;do a dummy operation on the parameter
16 STR r2,[fp,#-4] ;store it in the stack frame
17 ADD sp,sp,#4 ;clean up the stack frame
18 LDMFD sp!,{fp,pc} ;restore frame pointer and return
19
20 DCD 0x0000 ;clear memory
  
```

location
0x3C, the
current
SP value

location
0x40, the
current
FP value

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\ARM examples\chapter 1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

outNxt

Registers

Register	Value
R0	0x0000007C
R1	0x00000000
R2	0x000000F4
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000040
R12	0x00000000
R13 (SP)	0x00000040
R14 (LR)	0x00000014
R15 (PC)	0x00000038
CPSR	0x000000D3
SPSR	0x00000000

User/System

Fast Interrupt

Interrupt

Supervisor

Abort

Undefined

Internal

PC \$ 0x00000038

Mode Supervisor

States 20

Sec 0.00000000

Disassembly

```

0x00000014 E49D1004 LDR R1,[R13],#0x0004
9: Loop B Loop ;wait here (endless loop)
10:
0x00000018 EAEFFFE B 0x00000018
11: Sub STMFDP sp!,{fp,lr} ;push frame-pointer and link-register
0x0000001C E92D4800 STMDB R13!,{R11,R14}
12: MOV fp,sp ;frame pointer at the bottom of the frame
0x00000020 E1A0B00D MOV R11,R13
13: SUB sp,sp,#4 ;create the stack frame (one word)
0x00000024 E24DD004 SUB R13,R13,#0x00000004
14: LDR r2,[fp,#8] ;get the pushed parameter
0x00000028 E59B2008 LDR R2,[R11,#0x000008]
15: ADD r2,r2,#120 ;do a dummy operation on the parameter
0x0000002C E2822078 ADD R2,R2,#0x00000078
16: STR r2,[fp,#-4] ;store it in the stack frame
0x00000030 E50B2004 STR R2,[R11,#-0x0004]
17: ADD sp,sp,#4 ;clean up the stack frame
0x00000034 E28DD004 ADD R13,R13,#0x00000004
18: LDMFDP sp!,{fp,pc} ;restore frame pointer and return
0x00000038 E8BD8800 LDMIA R13!,{R11,PC}
0x0000003C 000000F4 ???EQ
0x00000040 0000007B ANDEO R0,R0,R11,ROR R0

```

Memory 1

Address: 0

0x00000000:	E2 8F D0 44 E3 A0 00 00 00 00 00 00 00 00 00 00
0x00000010:	EB 00 00 01 EA FF FF FE E9 2D 48 00
0x00000020:	E1 A0 B0 0D 4D D0 04 E5 9B 20 08 E2 82 20 78
0x00000030:	F5 12 20 04 E2 8D D0 04 E8 BD 88 00 00 00 00 F4
0x00000040:	00 00 00 7B 00 00 00 14 00 00 00 7C 00 00 00 00
0x00000050:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Call Stack + Locals

ch1.s

```

1 AREA TestProg, CODE, READONLY
2 ENTRY
3 Main ADR sp,Stack ;set up r13 as the stack pointer
4 MOV r0,#124 ;set up a dummy parameter
5 MOV fp,#123 ;dummy frame pointer
6 STR r0,[sp,#-4]! ;push the parameter
7 BL Sub ;call the subroutine
8 LDR r1,[sp],#4 ;retrieve the data
9 Loop B Loop ;wait here (endless loop)
10
11 Sub STMFDP sp!,{fp,lr} ;push frame-pointer and link-register
12 MOV fp,sp ;frame pointer at the bottom of the frame
13 SUB sp,sp,#4 ;create the stack frame (one word)
14 LDR r2,[fp,#8] ;get the pushed parameter
15 ADD r2,r2,#120 ;do a dummy operation on the parameter
16 STR r2,[fp,#-4] ;store it in the stack frame
17 ADD sp,sp,#4 ;clean up the stack frame
18 LDMFDP sp!,{fp,pc} ;restore frame pointer and return
19
20 DCD 0x0000 ;clear memory

```

location
0x40, the
current
SP value

location
0x40, the
current
FP value

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\ARM examples\chapter 1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Disassembly

Register	Value
Current	
R0	0x0000007C
R1	0x00000000
R2	0x000000F4
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x0000007B
R12	0x00000000
R13 (SP)	0x00000048
R14 (LR)	0x00000014
R15 (PC)	0x00000014
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000014
Mode	Supervisor
States	26
Sec	0.00000000

Disassembly

```

0x00000014 E49D1004 LDR R1,[R13],#0x0004
9: Loop B Loop ;wait here (endless loop)
10:
0x00000018 EAffFFFFE B 0x00000018
11: Sub STMFDP sp!,{fp,lr} ;push frame-pointer and link-register
0x0000001C E92D4800 STMDB R13!,{R11,R14}
12: MOV fp,sp ;frame pointer at the bottom of the frame
0x00000020 E1A0B00D MOV R11,R13
13: SUB sp,sp,#4 ;create the stack frame (one word)
0x00000024 E24DD004 SUB R13,R13,#0x00000004
14: LDR r2,[fp,#8] ;get the pushed parameter
0x00000028 E59B2008 LDR R2,[R11,#0x0008]
15: ADD r2,r2,#120 ;do a dummy operation on the parameter
0x0000002C E2822078 ADD R2,R2,#0x00000078
16: STR r2,[fp,#-4] ;store it in the stack frame
0x00000030 E50B2004 STR R2,[R11,#-0x0004]
17: ADD sp,sp,#4 ;clean up the stack frame
0x00000034 E28DD004 ADD R13,R13,#0x00000004
18: LDMFDP sp!,{fp,pc} ;restore frame pointer and return
0x00000038 E8BD8800 LDMIA R13!,{R11,PC}
0x0000003C 000000F4 ???EQ
0x00000040 0000007B ANDEQ R0,R0,R11,ROR R0
  
```

Memory 1

Address: 0

0x00000000:	E2 8F D0 44 E3 A0 00 7C E3 A0 B0 7C E3 A0 B0 7C E3 A0 B0 7C
0x00000010:	EB 00 00 01 E4 9D 10 04 EA FF F4 E9 2D 48 00
0x00000020:	E1 A0 B0 0D E2 4D D0 04 E5 92 20 08 E2 82 20 78
0x00000030:	E5 0B 20 04 E2 8D D0 04 E5 0B 20 04 E2 8D D0 04
0x00000040:	00 00 00 7B 00 00 00 14 00 00 00 7C 00 00 00 00
0x00000050:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Call Stack + Locals Memory 1

ch1.s

```

1 AREA TestProg, CODE, READONLY
2 ENTRY
3 Main ADR sp,Stack ;set up r13 as the stack pointer
4 MOV r0,#124 ;set up a dummy parameter
5 MOV fp,#123 ;dummy frame pointer
6 STR r0,[sp,#-4]! ;push the parameter
7 BL Sub ;call the subroutine
8 LDR r1,[sp],#4 ;retrieve the data
9 Loop B Loop ;wait here (endless loop)
10
11 Sub STMFDP sp!,{fp,lr} ;push frame-pointer and link-register
12 MOV fp,sp ;frame pointer at the bottom of the frame
13 SUB sp,sp,#4 ;create the stack frame (one word)
14 LDR r2,[fp,#8] ;get the pushed parameter
15 ADD r2,r2,#120 ;do a dummy operation on the parameter
16 STR r2,[fp,#-4] ;store it in the stack frame
17 ADD sp,sp,#4 ;clean up the stack frame
18 LDMFDP sp!,{fp,pc} ;restore frame pointer and return
19
20 DCD 0x0000 ;clear memory
  
```

location
0x48, the
current
SP value

C:\Users\Mahmoud El-Sakka\Desktop\COURSES_and_MATERIAL\2208\ARM examples\chapter 1.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

outNxt

Registers

Register	Value
R0	0x0000007C
R1	0x0000007C
R2	0x000000F4
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x0000007B
R12	0x00000000
R13 (SP)	0x0000004C
R14 (LR)	0x00000014
R15 (PC)	0x00000018
CPSR	0x000000D3
SPSR	0x00000000

User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC \$ 0x00000018
Mode Supervisor
States 29
Sec 0.00000000

Disassembly

```

0x00000014 E49D1004 LDR R1,[R13],#0x0004
9: Loop B Loop ;wait here (endless loop)
10:
0x00000018 EAEFFFE B 0x00000018
11: Sub STMFD sp!,{fp,lr} ;push frame-pointer and link-register
0x0000001C E92D4800 STMDB R13!,{R11,R14} ;frame pointer at the bottom of the frame
12: MOV fp,sp ;frame pointer at the bottom of the frame
0x00000020 E1A0B00D MOV R11,R13
13: SUB sp,sp,#4 ;create the stack frame (one word)
0x00000024 E24DD004 SUB R13,R13,#0x00000004 ;create the stack frame (one word)
14: LDR r2,[fp,#8] ;get the pushed parameter
0x00000028 E59B2008 LDR R2,[R11,#0x00000008]
15: ADD r2,r2,#120 ;do a dummy operation on the parameter
0x0000002C E2822078 ADD R2,R2,#0x00000078
16: STR r2,[fp,#-4] ;store it in the stack frame
0x00000030 E50B2004 STR R2,[R11,#-0x0004]
17: ADD sp,sp,#4 ;clean up the stack frame
0x00000034 E28DD004 ADD R13,R13,#0x00000004
18: LDMFD sp!,{fp,pc} ;restore frame pointer and return
0x00000038 E8BD8800 LDMIA R13!,{R11,PC}
0x0000003C 000000F4 ???EQ
0x00000040 0000007B ANDEQ R0,R0,R11,ROR R0

```

Memory 1

Address: 0

0x00000000:	E2 8F D0 44 E3 A0 00 7C E3 A0 B0 7B E5 2D 04
0x00000010:	EB 00 00 01 E4 9D 10 04 EA FF FF FE E9 20 08 00
0x00000020:	E1 A0 B0 0D E2 4D D0 04 E5 9B 20 08 E2 20 78
0x00000030:	E5 0B 20 04 E2 8D D0 04 E8 BD 88 00 00 00 F4
0x00000040:	00 00 00 7B 00 00 00 14 00 00 00 7C 00 00 00 00
0x00000050:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Call Stack + Locals Memory 1

ch1.s

```

1 AREA TestProg, CODE, READONLY
2 ENTRY
3 Main ADR sp,Stack ;set up r13 as the stack pointer
4 MOV r0,#124 ;set up a dummy parameter
5 MOV fp,#123 ;dummy frame pointer
6 STR r0,[sp,#-4]! ;push the parameter
7 BL Sub ;call the subroutine
8 LDR r1,[sp],#4 ;retrieve the data
9 Loop B Loop ;wait here (endless loop)
10
11 Sub STMFD sp!,{fp,lr} ;push frame-pointer and link-register
12 MOV fp,sp ;frame pointer at the bottom of the frame
13 SUB sp,sp,#4 ;create the stack frame (one word)
14 LDR r2,[fp,#8] ;get the pushed parameter
15 ADD r2,r2,#120 ;do a dummy operation on the parameter
16 STR r2,[fp,#-4] ;store it in the stack frame
17 ADD sp,sp,#4 ;clean up the stack frame
18 LDMFD sp!,{fp,pc} ;restore frame pointer and return
19
20 DCD 0x0000 ;clear memory

```

location
0x4C, the
current
SP value

SP and FP values are
correctly restored after
returning from the
function.