

Study Questions (Part 6)

Friday March 31, 2019

Covering:

- Block move instructions and
- Various ways to call a function and return from it

As a good start, you need to answer the question at the end of Chapter 3 of the textbook (pages 224-227).

1. Question 3.34 at page 225: What is the effect of executing STMIB **r13!**, {r0-r2,r4}? Draw a picture of the state of the stack pointed at by r13 before and after this operation.
2. Question 3.35 at page 225: The two pairs of instructions LDMIA, STMDB and LDMFD, STMFD do exactly the same things. Why do these two pairs have different or *alternative* mnemonics? Why does the first pair have different suffixes IA and DB? Why does the second pair have the same suffix FD?
3. Question 3.57 at page 226: We need to swap the following registers. Do this using *block moves*.

Before	After
r1	r3
r2	r4
r3	r5
r4	r6
r5	r7
r6	r1
r7	r2

4. There are 4 stack suffixes (**FD**, **FA**, **ED**, and **EA**) that are used in ARM assembly language. Write the equivalent translation of the following operation using **IA**, **IB**, **DA**, and **DB** suffixes.
 - a. LDMFD →
 - b. STMFD →
 - c. LDMFA →
 - d. STMFA →
 - e. LDMED →
 - f. STMED →
 - g. LDMEA →
 - h. STMEA →

5. What are the values of ABC, DEF, and GHI after executing the following program?

```
AREA Load, CODE, READONLY
ENTRY
MOV r1, PC
STR r1, ABC
STR PC, DEF
ADR r2, GHI
STMDA r2, {PC}

Loop B Loop
ABC DCD 0x00
DEF DCD 0x00
GHI DCD 0x00
END
```

6. What are the values of r0, r1, r2, and r3 (*in hexadecimal*) after executing the following program?

```
AREA prog, CODE, READWRITE
ENTRY
ADR    r3,AAA
MOV    r0,PC
STMEA  r3!,{PC}
STR    PC,[r3]
LDMFA  r3!,{r1,r2}
Loop   B      Loop
DCD    1,2
AAA    DCD    3
DCD    4,5
END
```

7. What are the values of r0, r1, r2, and r3 (*in hexadecimal*) after executing the following program?

```
AREA prog, CODE, READWRITE
ENTRY
ADR    r3,AAA
MOV    r0,PC
STMEA  r3!,{PC}
STR    PC,[r3]
LDMFA  r3,{r1,r2}
Loop   B      Loop
DCD    1,2
AAA    DCD    3
DCD    4,5
END
```

8. What are the values of r0, r1, r2, and r3 (*in hexadecimal*) after executing the following program?

```
AREA prog, CODE, READWRITE
ENTRY
ADR    r3,AAA
MOV    r0,PC
STMEA  r3,{PC}
STR    PC,[r3]
LDMFA  r3!,{r1,r2}
Loop   B      Loop
DCD    1,2
AAA    DCD    3
DCD    4,5
END
```

9. What are the values of r0, r1, r2, and r3 (*in hexadecimal*) after executing the following program?

```
AREA prog, CODE, READWRITE
ENTRY
ADR    r3,AAA
MOV    r0,PC
STMEA  r3,{PC}
STR    PC,[r3]
LDMFA  r3,{r1,r2}
Loop   B      Loop
DCD    1,2
AAA    DCD    3
DCD    4,5
END
```

10. What are the values of r1, r2, r3, r4, and sp (in hexadecimal) after executing the following program?

```
AREA StackTest1, CODE, READONLY
ENTRY
ADR    sp,  STACK
MOV    r1,  #0x11
MOV    r2,  #0x22
MOV    r3,  #0x33
MOV    r4,  #0x44
STMDB  sp!, {r2,r4,r1}
MOV    r1,  #0xAA
MOV    r2,  #0xBB
MOV    r3,  #0xCC
MOV    r4,  #0xDD
LDMIA  sp!, {r1-r2,r4}
LOOP   B     LOOP
AREA StackTest1, DATA, READWRITE
DCD    0, 0, 0, 0
STACK  DCD    0xFFFFFFFF
DCD    0, 0, 0, 0
END
```

Draw a sketch to show the **address** and the **content** of each cell in the DATA area in the above code:

before executing the STMDB ARM instruction,

after executing the STMDB ARM instruction, and

after executing the LDMIA ARM instruction.

Draw an arrow to a stack cell to indicate the value of sp in each case (i.e., the location of stack pointer).

11. What are the values of r1, r2, r3, r4, and sp (in hexadecimal) after executing the following program?

```
AREA StackTest2, CODE, READONLY
ENTRY
ADR    sp,  STACK
MOV    r1,  #0x11
MOV    r2,  #0x22
MOV    r3,  #0x33
MOV    r4,  #0x44
STMIA  sp!, {r2,r4,r1}
MOV    r1,  #0xAA
MOV    r2,  #0xBB
MOV    r3,  #0xCC
MOV    r4,  #0xDD
LDMDB  sp!, {r1-r2,r4}
LOOP   B     LOOP
AREA StackTest2, DATA, READWRITE
DCD    0, 0, 0, 0
STACK  DCD    0xFFFFFFFF
DCD    0, 0, 0, 0
END
```

Draw a sketch to show the **address** and the **content** of each cell in the DATA area in the above code:

before executing the STMIA ARM instruction,

after executing the STMIA ARM instruction, and

after executing the LDMDB ARM instruction.

Draw an arrow to a stack cell to indicate the value of sp in each case (i.e., the location of stack pointer).

12. What are the values of r1, r2, r3, r4, and sp (in hexadecimal) after executing the following program?

```

AREA StackTest3, CODE, READONLY
ENTRY
ADR    sp,  STACK
MOV    r1,  #0x11
MOV    r2,  #0x22
MOV    r3,  #0x33
MOV    r4,  #0x44
STMDB  sp!, {r2,r4,r1}
MOV    r1,  #0xAA
MOV    r2,  #0xBB
MOV    r3,  #0xCC
MOV    r4,  #0xDD
LDMIB  sp!, {r1-r2,r4}
LOOP   B     LOOP
AREA StackTest3, DATA, READWRITE
DCD    0, 0, 0, 0
STACK  DCD    0xFFFFFFFF
DCD    0, 0, 0, 0
END

```

Draw a sketch to show the **address** and the **content** of each cell in the DATA area in the above code:

before executing the STMDB ARM instruction,

after executing the STMDB ARM instruction, and

after executing the LDMIB ARM instruction.

Draw an arrow to a stack cell to indicate the value of sp in each case (i.e., the location of stack pointer).

13. What are the values of r1, r2, r3, r4, and sp (in hexadecimal) after executing the following program?

```

AREA StackTest4, CODE, READONLY
ENTRY
ADR    sp,  STACK
MOV    r1,  #0x11
MOV    r2,  #0x22
MOV    r3,  #0x33
MOV    r4,  #0x44
STMIA  sp!, {r2,r4,r1}
MOV    r1,  #0xAA
MOV    r2,  #0xBB
MOV    r3,  #0xCC
MOV    r4,  #0xDD
LDMDA  sp!, {r1-r2,r4}
LOOP   B     LOOP
AREA StackTest4, DATA, READWRITE
DCD    0, 0, 0, 0
STACK  DCD    0xFFFFFFFF
DCD    0, 0, 0, 0
END

```

Draw a sketch to show the **address** and the **content** of each cell in the DATA area in the above code:

before executing the STMIA ARM instruction,

after executing the STMIA ARM instruction, and

after executing the LDMDA ARM instruction.

Draw an arrow to a stack cell to indicate the value of sp in each case (i.e., the location of stack pointer).

14. In the following code, write an ARM assembly instruction (at location `return`) to return from the function `fun1` to the main program.

What are the values of `r0`, `r1`, and `r2` at the end of the program?

```

        AREA FunctionCall, CODE, READONLY
        ENTRY
main     ADR     sp,   STACK
        ADR     r1,   input
        LDR     r0, [r1], #4
call     BL      fun1
        STR     r0, [r1]
LOOP     B       LOOP

fun1     STMDB   sp!, {r1, r2, LR}
        MOV     r1, r0
        MUL     r2, r1, r0
        MUL     r0, r1, r2
return   .....
        AREA FunctionCall, DATA, READWRITE
        DCD     0, 0, 0, 0
STACK    DCD     0xFFFFFFFF
        DCD     0, 0, 0, 0
input    DCD     0x11
output   DCD     0x00
        END

```

Draw a sketch to show the **address** and the **content** of each cell in the DATA area in the above code: **before** executing the `STMDB` ARM instruction, **after** executing the `STMDB` ARM instruction, and **at the end of the program**.

Draw an arrow to a stack cell to indicate the value of `sp` in each case (i.e., the location of stack pointer).

15. In the following code, write an ARM assembly instruction (at location `return`) to return from the function `fun1` to the main program.

What are the values of `r0`, `r1`, and `r2` at the end of the program?

```

        AREA FunctionCall, CODE, READONLY
        ENTRY
main     ADR     sp,   STACK
        ADR     r1,   input
        LDR     r0, [r1], #4
call     BL      fun1
        STR     r0, [r1]
LOOP     B       LOOP

fun1     STMIB   sp!, {r1, r2, LR}
        MOV     r1, r0
        MUL     r2, r1, r0
        MUL     r0, r1, r2
return   .....
        AREA FunctionCall, DATA, READWRITE
        DCD     0, 0, 0, 0
STACK    DCD     0xFFFFFFFF
        DCD     0, 0, 0, 0
input    DCD     0x11
output   DCD     0x00
        END

```

Draw a sketch to show the **address** and the **content** of each cell in the DATA area in the above code: **before** executing the `STMIB` ARM instruction, **after** executing the `STMIB` ARM instruction, and **at the end of the program**.

Draw an arrow to a stack cell to indicate the value of `sp` in each case (i.e., the location of stack pointer).

16. In the following code, write an ARM assembly instruction (at location `return`) to return from the function `fun1` to the main program.

What are the values of `r0`, `r1`, and `r2` at the end of the program?

```

        AREA FunctionCall, CODE, READONLY
        ENTRY
main     ADR     sp,   STACK
        ADR     r1,   input
        LDR     r0, [r1], #4
call     BL      fun1
        STR     r0, [r1]
LOOP     B       LOOP

fun1     STMDA   sp!, {r1, r2, LR}
        MOV     r1, r0
        MUL     r2, r1, r0
        MUL     r0, r1, r2
return   .....
        AREA FunctionCall, DATA, READWRITE
        DCD     0, 0, 0, 0
STACK    DCD     0xFFFFFFFF
        DCD     0, 0, 0, 0
input    DCD     0x11
output   DCD     0x00
        END

```

Draw a sketch to show the **address** and the **content** of each cell in the DATA area in the above code: **before** executing the `STMDA` ARM instruction, **after** executing the `STMDA` ARM instruction, and **at the end of the program**.

Draw an arrow to a stack cell to indicate the value of `sp` in each case (i.e., the location of stack pointer).

17. In the following code, write an ARM assembly instruction (at location `return`) to return from the function `fun1` to the main program.

What are the values of `r0`, `r1`, and `r2` at the end of the program?

```

        AREA FunctionCall, CODE, READONLY
        ENTRY
main     ADR     sp,   STACK
        ADR     r1,   input
        LDR     r0, [r1], #4
call     BL      fun1
        STR     r0, [r1]
LOOP     B       LOOP

fun1     STMIA   sp!, {r1, r2, LR}
        MOV     r1, r0
        MUL     r2, r1, r0
        MUL     r0, r1, r2
return   .....
        AREA FunctionCall, DATA, READWRITE
        DCD     0, 0, 0, 0
STACK    DCD     0xFFFFFFFF
        DCD     0, 0, 0, 0
input    DCD     0x11
output   DCD     0x00
        END

```

Draw a sketch to show the **address** and the **content** of each cell in the DATA area in the above code: **before** executing the `STMIA` ARM instruction, **after** executing the `STMIA` ARM instruction, and **at the end of the program**.

Draw an arrow to a stack cell to indicate the value of `sp` in each case (i.e., the location of stack pointer).

18. In the following code, write ARM assembly instructions (at location `return`) to return from the function `fun1` to the main program.

What are the values of `r0`, `r1`, and `r2` at the end of the program?

```

        AREA FunctionCall, CODE, READONLY
        ENTRY
main     ADR     sp,   STACK
        ADR     r1,   input
        LDR     r0,[r1],#4
        STR     pc,[sp,#-4]!
call     B       fun1
        STR     r0,[r1]
LOOP     B       LOOP
fun1     STMFD   sp!,{r1,r2}
        MOV     r1,r0
        MUL     r2,r1,r0
        MUL     r0,r1,r2
return   .....
        .....
        .....
        AREA FunctionCall, DATA, READWRITE
STACK    DCD     0, 0, 0, 0
        DCD     0xFFFFFFFF
        DCD     0, 0, 0, 0
input    DCD     0x11
output   DCD     0x00
        END

```

Draw a sketch to show the address and the content of each cell in the DATA area in the above code:
before executing the STMFD ARM instruction,
after executing the STMFD ARM instruction, and
at the end of the program.

Draw an arrow to a stack cell to indicate the value of `sp` in each case (i.e., the location of stack pointer).

19. In the following code, write ARM assembly instructions (at location `return`) to return from the function `fun1` to the main program.

What are the values of `r0`, `r1`, and `r2` at the end of the program?

```

        AREA FunctionCall, CODE, READONLY
        ENTRY
main     ADR     sp,   STACK
        ADR     r1,   input
        LDR     r0,[r1],#4
        STR     pc,[sp,#-4]!
        NOP
call     B       fun1
        STR     r0,[r1]
LOOP     B       LOOP
fun1     STMFD   sp!,{r1,r2}
        MOV     r1,r0
        MUL     r2,r1,r0
        MUL     r0,r1,r2
return   .....
        .....
        .....
        AREA FunctionCall, DATA, READWRITE
STACK    DCD     0, 0, 0, 0
        DCD     0xFFFFFFFF
        DCD     0, 0, 0, 0
input    DCD     0x11
output   DCD     0x00
        END

```

Draw a sketch to show the address and the content of each cell in the DATA area in the above code:
before executing the STMFD ARM instruction,
after executing the STMFD ARM instruction, and
at the end of the program.

Draw an arrow to a stack cell to indicate the value of `sp` in each case (i.e., the location of stack pointer).

20. In the following code, write ARM assembly instructions (at location `return`) to return from the function `fun1` to the main program.

What are the values of `r0`, `r1`, and `r2` at the end of the program?

```

        AREA FunctionCall, CODE, READONLY
        ENTRY
main     ADR     sp,   STACK
        ADR     r1,   input
        LDR     r0,[r1],#4
        STR     pc,[sp,#4]!
call     B       fun1
        STR     r0, [r1]
LOOP     B       LOOP
fun1     STMFA   sp!, {r1,r2}
        MOV     r1,r0
        MUL     r2,r1,r0
        MUL     r0,r1,r2
return   .....
        .....
        .....
        AREA FunctionCall, DATA, READWRITE
STACK    DCD     0, 0, 0, 0
        DCD     0xFFFFFFFF
        DCD     0, 0, 0, 0
input    DCD     0x11
output   DCD     0x00
        END

```

Draw a sketch to show the address and the content of each cell in the DATA area in the above code:
before executing the STMFA ARM instruction,
after executing the STMFA ARM instruction, and
at the end of the program.

Draw an arrow to a stack cell to indicate the value of `sp` in each case (i.e., the location of stack pointer).

21. In the following code, write ARM assembly instructions (at location `return`) to return from the function `fun1` to the main program.

What are the values of `r0`, `r1`, and `r2` at the end of the program?

```

        AREA FunctionCall, CODE, READONLY
        ENTRY
main     ADR     sp,   STACK
        ADR     r1,   input
        LDR     r0,[r1],#4
        STR     pc,[sp,#4]!
        NOP
call     B       fun1
        STR     r0, [r1]
LOOP     B       LOOP
fun1     STMFA   sp!, {r1,r2}
        MOV     r1,r0
        MUL     r2,r1,r0
        MUL     r0,r1,r2
return   .....
        .....
        .....
        AREA FunctionCall, DATA, READWRITE
STACK    DCD     0, 0, 0, 0
        DCD     0xFFFFFFFF
        DCD     0, 0, 0, 0
input    DCD     0x11
output   DCD     0x00
        END

```

Draw a sketch to show the address and the content of each cell in the DATA area in the above code:
before executing the STMFA ARM instruction,
after executing the STMFA ARM instruction, and
at the end of the program.

Draw an arrow to a stack cell to indicate the value of `sp` in each case (i.e., the location of stack pointer).

22. In the following code, write ARM assembly instructions (at location `return`) to return from the function `fun1` to the main program.

What are the values of `r0`, `r1`, and `r2` at the end of the program?

```

        AREA FunctionCall, CODE, READONLY
        ENTRY
main     ADR     sp,   STACK
        ADR     r1,   input
        LDR     r0,[r1],#4
        STR     pc,[sp],#-4
call     B       fun1
        STR     r0, [r1]
LOOP    B       LOOP
fun1     STMED   sp!, {r1,r2}
        MOV     r1,r0
        MUL     r2,r1,r0
        MUL     r0,r1,r2
return   .....
        .....
        .....
        AREA FunctionCall, DATA, READWRITE
STACK   DCD     0, 0, 0, 0
        DCD     0xFFFFFFFF
        DCD     0, 0, 0, 0
input   DCD     0x11
output  DCD     0x00
        END

```

Draw a sketch to show the address and the content of each cell in the DATA area in the above code:
before executing the STMED ARM instruction,
after executing the STMED ARM instruction, and
at the end of the program.

Draw an arrow to a stack cell to indicate the value of `sp` in each case (i.e., the location of stack pointer).

23. In the following code, write ARM assembly instructions (at location `return`) to return from the function `fun1` to the main program.

What are the values of `r0`, `r1`, and `r2` at the end of the program?

```

        AREA FunctionCall, CODE, READONLY
        ENTRY
main     ADR     sp,   STACK
        ADR     r1,   input
        LDR     r0,[r1],#4
        STR     pc,[sp],#-4
        NOP
call     B       fun1
        STR     r0, [r1]
LOOP    B       LOOP
fun1     STMED   sp!, {r1,r2}
        MOV     r1,r0
        MUL     r2,r1,r0
        MUL     r0,r1,r2
return   .....
        .....
        .....
        AREA FunctionCall, DATA, READWRITE
STACK   DCD     0, 0, 0, 0
        DCD     0xFFFFFFFF
        DCD     0, 0, 0, 0
input   DCD     0x11
output  DCD     0x00
        END

```

Draw a sketch to show the address and the content of each cell in the DATA area in the above code:
before executing the STMED ARM instruction,
after executing the STMED ARM instruction, and
at the end of the program.

Draw an arrow to a stack cell to indicate the value of `sp` in each case (i.e., the location of stack pointer).

24. In the following code, write ARM assembly instructions (at location `return`) to return from the function `fun1` to the main program.

What are the values of `r0`, `r1`, and `r2` at the end of the program?

```

        AREA FunctionCall, CODE, READONLY
        ENTRY
main     ADR     sp,   STACK
        ADR     r1,   input
        LDR     r0,[r1],#4
        STR     pc,[sp],#4
call     B       fun1
        STR     r0, [r1]
LOOP     B       LOOP
fun1     STMEA   sp!, {r1,r2}
        MOV     r1,r0
        MUL     r2,r1,r0
        MUL     r0,r1,r2
return   .....
        .....
        .....
        AREA FunctionCall, DATA, READWRITE
STACK    DCD     0, 0, 0, 0
        DCD     0xFFFFFFFF
        DCD     0, 0, 0, 0
input    DCD     0x11
output   DCD     0x00
        END

```

Draw a sketch to show the address and the content of each cell in the DATA area in the above code:
before executing the `STMEA` ARM instruction,
after executing the `STMEA` ARM instruction, and
at the end of the program.

Draw an arrow to a stack cell to indicate the value of `sp` in each case (i.e., the location of stack pointer).

25. In the following code, write an ARM assembly instruction (at location `return`) to return from the function `fun1` to the main program.

What are the values of `r0`, `r1`, and `r2` at the end of the program?

```

        AREA FunctionCall, CODE, READONLY
        ENTRY
main     ADR     sp,   STACK
        ADR     r1,   input
        LDR     r0,[r1],#4
        STR     pc,[sp],#4
        NOP
call     B       fun1
        STR     r0, [r1]
LOOP     B       LOOP
fun1     STMEA   sp!, {r1,r2}
        MOV     r1,r0
        MUL     r2,r1,r0
        MUL     r0,r1,r2
return   .....
        .....
        .....
        AREA FunctionCall, DATA, READWRITE
STACK    DCD     0, 0, 0, 0
        DCD     0xFFFFFFFF
        DCD     0, 0, 0, 0
input    DCD     0x11
output   DCD     0x00
        END

```

Draw a sketch to show the address and the content of each cell in the DATA area in the above code:
before executing the `STMEA` ARM instruction,
after executing the `STMEA` ARM instruction, and
at the end of the program.

Draw an arrow to a stack cell to indicate the value of `sp` in each case (i.e., the location of stack pointer).

26. Encode the following ARM assembly instruction to ARM machine language code.
STMFDVS **sp!**, {r1-r4,r7}
27. Decode the following ARM machine language code to ARM assembly instruction.
0x692D009E
28. Encode the following ARM assembly instruction to ARM machine language code.
STMVCFA **sp**, {r1-r4,r7}
29. Decode the following ARM machine language code to ARM assembly instruction.
0x798D009E
30. Encode the following ARM assembly instruction to ARM machine language code.
STMEDHI **r1!**, {r2-r10,r14}
31. Decode the following ARM machine language code to ARM assembly instruction.
0x882147FC
32. Encode the following ARM assembly instruction to ARM machine language code.
STMLSEA **r1**, {r2-r10,r14}
33. Decode the following ARM machine language code to ARM assembly instruction.
0x988147FC
34. Encode the following ARM assembly instruction to ARM machine language code.
LDMFDGE **sp!**, {r1-r4,r7}
35. Decode the following ARM machine language code to ARM assembly instruction.
0xA8BD009E
36. Encode the following ARM assembly instruction to ARM machine language code.
LDMLTFA **sp**, {r1-r4,r7}
37. Decode the following ARM machine language code to ARM assembly instruction.
0xB81D009E
38. Encode the following ARM assembly instruction to ARM machine language code.
LDMEDGT **r1!**, {r2-r10,r14}
39. Decode the following ARM machine language code to ARM assembly instruction.
0xC9B147FC
40. Encode the following ARM assembly instruction to ARM machine language code.
LDMLEEA **r1**, {r2-r10,r14}
41. Decode the following ARM machine language code to ARM assembly instruction.
0xD91147FC