Study Questions (Part 1)
Sunday February 24, 2019

Covering:

- Introduction to ARM assembly
- Arithmetic instructions (addition and subtraction)
- Pseudo instructions

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

1. Question 3.2 at page 224: Explain the function of the following registers in a CPU
   a. PC
   b. MAR
   c. MBR
   d. IR
2. Question 3.3 at page 224: For each of the following 6-bit operations calculate the values of the C, Z, V, and N flags.
   a. 001011 + 001101
   b. 111111 + 000001
   c. 000000 - 111111
   d. 101101 + 011011
   e. 000000 – 000001
   f. 111110 + 111111
3. Question 3.10 at page 224: Why does the ARM provide a reverse subtract instruction RSB **r0**,r1,r2, that implements [r0] = [r2] - [r1], when the normal subtraction instruction SUB **r0**,r2,r1, will do exactly the same job?
4. Question 3.30 at page 225: What is the meaning of *sign-extension* in the context of copying data from one location to another?
5. Question 3.31 at page 225: Why is sign-extension an important issue when we use a LDR instruction to load a register from memory, but of no importance when we use an STR to store a register in memory?
6. Question 3.48 at page 226: What, in the context of assembly language, is a *pseudo-operation*?
7. Question 3.49 at page 226: Explain the meaning of the following two ARM pseudo-operations. What do they do and why have they been implemented?
   a. ADR **r0**,table
   b. LDR **r0**,=0x1234FEDC
8. Question 3.50 at page 226: Suppose you execute LDR,**r0**=0x12345678 on an ARM machine followed by STR r0,**[r1]** where r1 =0x1000. Now, suppose you do a byte read to the same address with LDRB **r2**,[r1]. What would the value be in r2 if (a) the ARM was big-endian configured and (b) it was little-endian configured?
9. Question 3.56 at page 226: The following is a loop expressed in ARM code. The code is wrong. Why?

```
     MOV r0,#10   ;Loop counter – ten times
     MOV r1,#0    ;initialize the total to zero
Next ADD r1,r1,r0 ;add loop counter to total
     SUB r0,r0,#1 ;decrement loop counter
     BEQ Next     ;continue until all done
```

10. Question 3.61 at page 227: Register r15 is the program counter. You can use it with certain instructions such as a MOV (e.g., MOV **pc**,r14). However, r15 cannot be used in conjunction with most data processing instructions. Why?

11. What do the following terms stand for?
    a. MAR
    b. MBR
    c. IR
    d. PC
    e. SP
    f. LR
    g. ISA
    h. RTL
    i. RISC
    j. CISC
    k. ARM
    l. ALU
    m. CCR
    n. CPSR
12. Describe, using RTL notation, how RISC processors perform the fetch phase of the processing cycle.
13. What does the term *word aligned address* mean?
14. What does the term *half-word aligned address* mean?
15. Is the address 0x123468AA a *word aligned* address?
16. Is the address 0x123468AA a *half-word aligned* address?
17. Is the address 0x123468AB a *word aligned* address?
18. Is the address 0x123468AB a *half-word aligned* address?
19. Is the address 0x123468AC a *word aligned* address?
20. Is the address 0x123468AC a *half-word aligned* address?
21. What is the main function of *program counter* and how it is differ than *location counter*?
22. What is the main function of *location counter* and how it is differ than *program counter*?
23. Write only ONE ARM instruction to copy into `r0` the value `0x12`
24. Write only ONE ARM instruction to copy into `r0` the value `-0x12`
25. Write only ONE ARM instruction to copy into `r0` the value `0x12345678`
26. Write only ONE ARM instruction to copy into `r0` the value `-0x12345678`

27. Write only ONE ARM instruction to copy into `r0` the value `0xFFFFFFEE`

28. Write only ONE ARM instruction to copy into `r0` the value `-0xFFFFFFEE`

29. Write only THREE ARM instructions to copy `0xFF` into `r1`, copy the value in `r1` into `r2`, add the values in `r1` and `r2`, and finally store the result in `r0`.
    How do you enforce updating the condition flags after the addition operation?

30. Write only THREE ARM instructions to copy `0xFF` into `r1`, copy the negative of the value in `r1` into `r2`, add the values in `r1` and `r2`, and finally store the result in `r0`.
    How do you enforce updating the condition flags after the addition operation?

31. Write only THREE ARM instructions to copy `0xFF` into `r1`, copy the complement of each bit of the value in `r1` into `r2`, add the values in `r1` and `r2`, and finally store the result in `r0`.
    How do you enforce updating the condition flags after the addition operation?

32. Write only THREE ARM instructions to copy `0xFF` into `r1`, copy `0xEE` into `r2`, add the values in `r1` and `r2`, and finally store the result in `r0`.
    How do you enforce updating the condition flags after the addition operation?

33. Write only THREE ARM instructions to copy `0xFF` into `r1`, copy `0xFEDCBA98` into `r2`, add the values in `r1` and `r2`, and finally store the result in `r0`.
    How do you enforce updating the condition flags after the addition operation?

34. Write only TWO ARM instructions to copy `0xFF` into `r1`, add `0xEE` to the values in `r1`, and finally store the result in `r0`.
    How do you enforce updating the condition flags after the addition operation?

35. Write only THREE ARM instructions to copy `0xFF` into `r1`, copy `0xFEDCBA98` into `r2`, subtract the value in `r1` from the value in `r2`, and finally store the result in `r0`.
    How do you enforce updating the condition flags after the subtraction operation?

36. Write only TWO ARM instructions to copy `0xFF` into `r1`, subtract `0xEE` from the value in `r1`, and finally store the result in `r0`.
    How do you enforce updating the condition flags after the subtraction operation?

37. Write only TWO ARM instructions to copy `0xFF` into `r1`, subtract the value in `r1` from `0xEE`, and finally store the result in `r0`.
    How do you enforce updating the condition flags after the subtraction operation?

38. How to extend ARM arithmetic capabilities to make it able to add two extended precisions integer values (64-bits each). Give an example. Write down all assumptions you will make.

39. How to extend ARM arithmetic capabilities to make it able to add two extended precisions integer values (96-bits each). Give an example. Write down all assumptions you will make.

40. What are the main differences between ADD, ADDS, ADC, and ADCS?
    Provide numeric examples to demonstrate the differences.

41. What are the main differences between SUB, SUBS, SBC, and SBCS?
    Provide numeric examples to demonstrate the differences.

42. What are the main differences between RSB, RSBS, RSC, and RSCS?
    Provide numeric examples to demonstrate the differences.

43. What are the main differences between SUB, SUBS, RSB, and RSBS?
    Provide numeric examples to demonstrate the differences.

44. What are the main differences between LDR, and ADR?
    Provide numeric examples to demonstrate the differences.

45. After executing the following ARM instructions

```
LDR  r0,=0x87654321
LDR  r1,=0x87654321
ADDS r2,r1,r0
ADC  r3,r1,r0
```

    What will be the values of the NZCV flags, as well as the values of r0, r1, r2, and r3?

46. After executing the following ARM instructions

```
LDR  r0,=0xFEDCBA98
LDR  r1,=0x87654321
ADDS r2,r1,r0
ADC  r3,r1,r0
```

    What will be the values of the NZCV flags, as well as the values of r0, r1, r2, and r3?

47. After executing the following ARM instructions

```
LDR  r0,=0xFEDCBA98
LDR  r1,=0xFEDCBA98
ADDS r2,r1,r0
ADC  r3,r1,r0
```

    What will be the values of the NZCV flags, as well as the values of r0, r1, r2, and r3?

48. After executing the following ARM instructions

```
LDR  r0,=0x87654321
LDR  r1,=0x87654321
SUBS r2,r1,r0
ADC  r3,r1,r0
```

    What will be the values of the NZCV flags, as well as the values of r0, r1 r2, and r3?

49. After executing the following ARM instructions

```
LDR  r0,=0xFEDCBA98
LDR  r1,=0x87654321
SUBS r2,r1,r0
ADC  r3,r1,r0
```

    What will be the values of the NZCV flags, as well as the values of r0, r1, r2, and r3?

50. After executing the following ARM instructions

```
LDR  r0,=0xFEDCBA98
LDR  r1,=0xFEDCBA98
SUBS r2,r1,r0
ADC  r3,r1,r0
```

    What will be the values of the NZCV flags, as well as the values of r0, r1, r2, and r3?

51. After executing the following ARM instructions

```
LDR  r0,=0x87654321
LDR  r1,=0x87654321
RSBS r2,r1,r0
ADC  r3,r1,r0
```

What will be the values of the NZCV flags, as well as the values of r0, r1, r2, and r3?

52. After executing the following ARM instructions

```
LDR  r0,=0xFEDCBA98
LDR  r1,=0x87654321
RSBS r2,r1,r0
ADC  r3,r1,r0
```

What will be the values of the NZCV flags, as well as the values of r0, r1, r2, and r3?

53. After executing the following ARM instructions

```
LDR  r0,=0xFEDCBA98
LDR  r1,=0xFEDCBA98
RSBS r2,r1,r0
ADC  r3,r1,r0
```

What will be the values of the NZCV flags, as well as the values of r0, r1, r2, and r3?

54. After executing the following ARM instructions

```
LDR  r0,=0x1234567
MOV  r1,r0
```

What will be the values of the NZCV flags, as well as the values of r0 and r1?

55. After executing the following ARM instructions

```
LDR  r0,=0x1234567
MOVS r1,r0
```

What will be the values of the NZCV flags, as well as the values of r0 and r1?

56. After executing the following ARM instructions

```
LDR  r0,=0xFEDCBA98
MOV  r1,r0
```

What will be the values of the NZCV flags, as well as the values of r0 and r1?

57. After executing the following ARM instructions

```
LDR  r0,=0xFEDCBA98
MOVS r1,r0
```

What will be the values of the NZCV flags, as well as the values of r0 and r1?

58. After executing the following ARM instructions

```
LDR  r0,= 0x1234567
NEG  r1,r0
```

What will be the values of the NZCV flags, as well as the values of r0 and r1?

59. After executing the following ARM instructions

```
LDR  r0,=0x1234567
NEGS r1,r0
```

What will be the values of the NZCV flags, as well as the values of r0 and r1?

60. After executing the following ARM instructions
```
LDR  r0,=0xFEDCBA98
NEG  r1,r0
```
What will be the values of the NZCV flags, as well as the values of r0 and r1?

61. After executing the following ARM instructions
```
LDR  r0,=0xFEDCBA98
NEGS r1,r0
```
What will be the values of the NZCV flags, as well as the values of r0 and r1?

62. After executing the following ARM instructions
```
LDR  r0,=-0x123
NEG  r1,r0
```
What will be the values of the NZCV flags, as well as the values of r0 and r1?

63. After executing the following ARM instructions
```
LDR  r0,=-0x123
NEGS r1,r0
```
What will be the values of the NZCV flags, as well as the values of r0 and r1?

64. After executing the following ARM instructions
```
LDR  r0,=-0xFFFFF123
NEG  r1,r0
```
What will be the values of the NZCV flags, as well as the values of r0 and r1?

65. After executing the following ARM instructions
```
LDR  r0,=-0xFFFFF123
NEGS r1,r0
```
What will be the values of the NZCV flags, as well as the values of r0 and r1?

66. After executing the following ARM instructions
```
LDR  r0,=-0xFFFFFFFF
NEG  r1,r0
```
What will be the values of the NZCV flags, as well as the values of r0 and r1?

67. After executing the following ARM instructions
```
LDR  r0,=-0xFFFFFFFF
NEGS r1,r0
```
What will be the values of the NZCV flags, as well as the values of r0 and r1?

68. After executing the following ARM instructions
```
MOV  r0,#0
NEG  r1,r0
```
What will be the values of the NZCV flags, as well as the values of r0 and r1?

69. After executing the following ARM instructions
```
MOV  r0,#0
NEGS r1,r0
```
What will be the values of the NZCV flags, as well as the values of r0 and r1?

70. After executing the following ARM instructions
```
MOV  r0,#1
NEG  r1,r0
```
What will be the values of the NZCV flags, as well as the values of r0 and r1?

71. After executing the following ARM instructions

```
        MOV  r0,#1
        NEGS r1,r0
```

What will be the values of the NZCV flags, as well as the values of r0 and r1?

72. In the following program, how the pseudo instructions are implemented?

```
        AREA my_First_Example, CODE, READONLY
        ENTRY
        LDR r0, =0x12
        LDR r1, =-0x12
        NEG r2, r1
p       B   p
        END
```

What will be the values of r0, r1, r2, and r3 after executing this program?

73. In the following program, how the pseudo instructions are implemented?

```
        AREA my_First_Example, CODE, READONLY
        ENTRY
        LDR r0, =0xFFFFFFEE
        LDR r1, =-0xFFFFFFEE
        NEG r2, r1
p       B   p
        END
```

What will be the values of r0, r1, r2, and r3 after executing this program?

74. In the following program, how the pseudo instructions are implemented?

```
        AREA my_First_Example, CODE, READONLY
        ENTRY
        LDR r0, =0x12345678
        LDR r1, =ppp
        LDR r2, ppp
        ADR r3, ppp
p       B   p
ppp     DCD 0xFFFF
        END
```

What will be the values of r0, r1, r2, and r3 after executing this program?

75. the following program, how the pseudo instructions are implemented?

```
        AREA my_First_Example, CODE, READONLY
        ENTRY
        LDR r0, =0x12345678
        LDR r1, =ppp
        LDR r2, ppp
        ADR r3, ppp
p       B   p

        AREA my_First_Example, DATA, READONLY
ppp     DCD 0xFFFF
        END
```

What will be the values of r0, r1, r2, and r3 after executing this program?

76. Consider the following program,

```
        AREA my_First_Example, CODE, READONLY
        ENTRY
        LDR  r1, ppp
        LDRB r2, ppp
p       B    p

ppp     DCD 0x12345678
        END
```

What will be the values of r1 and r1 after executing this program?

77. Consider the following program,

```
        AREA my_First_Example, CODE, READONLY
        ENTRY
        LDR  r1, ppp
        LDRB r2, ppp
p       B    p

ppp     DCD 0xabcde
        END
```

What will be the values of r1 and r1 after executing this program?

78. Consider the following program,

```
        AREA my_First_Example, CODE, READONLY
        ENTRY
        LDR  r1, ppp
        STR  r1, qqq
        STRB r1, rrr
        STRB r1, sss
p       B    p

ppp     DCD  0xabcde
qqq     DCD  0
rrr     DCD  0
sss     DCB  0
        END
```

What will be the values of r1 and the memory locations qqq, rrr, and sss after executing this program?

79. Consider the following program,

```
        AREA my_First_Example, CODE, READONLY
        ENTRY
p       B     p

ppp     SPACE 4
        DCB   0xFF
qqq     SPACE 4
        DCB   0xFF
        END
```

What will be the addresses ppp and qqq?

80. Consider the following program,

```
        AREA my_First_Example, CODE, READONLY
        ENTRY
p       B     p
ppp     SPACE 4
        DCB   0xFF
        ALIGN
qqq     SPACE 4
        DCB   0xFF
        END
```

What will be the addresses ppp and qqq?

81. Consider the following program,

```
        AREA my_First_Example, CODE, READONLY
        ENTRY
p       B     p
ppp     SPACE 4
        DCB   0xFF
        DCD   0x1111
qqq     SPACE 4
        DCB   0xFF
        END
```

What will be the addresses ppp and qqq?