***Advanced System on Chip Course***

**QUIZ 4**

**Issue 1.0**

# MODULE 3 and 4: Armv7-A/R ISA Overview

**Question 1:**

You execute the following instruction:

STR R1, [R2]

This may cause

1. The contents of R1 to be modified.
2. The contents of R2 to be modified.
3. The contents of both R1 and R2 to be modified.
4. None of the above.

**Question 2:**

You execute the following two instructions:

MOVS R1, 0

MUL R1, R2

This may cause

1. The contents of R1 and R2 to be modified.
2. The contents of R2 and R14 to be modified.
3. The contents of R1 and R15 to be modified.
4. All of the above.

**Question 3:**

What does the execution of following assembly code achieve?

MUL R2, R4

STR R2, [R5]

1. It multiplies the values stored in R2 and R4 and stores the result in R5.
2. It multiplies the values stored in R2 and R4 and stores the result in the memory location whose address is stored in R5.
3. It first multiplies the values stored in R2 and R4, then copies the value stored in R5 into R2.
4. It changes the contents of R2, and R5.

**Question 4:**

Which of the following branch instructions is NOT conditional?

1. BEQ R2
2. BHI R1
3. BLE R1
4. BLX R0

**Question 5:**

Which of the following branch instructions will change the value of the link register?

1. BLT R2
2. BLE R2
3. BLX R2
4. None of the above.

**Question 6:**

Endianness refers to the order of bytes stored in memory. Which of the following instructions would you use for conversion of 32-bit data items from a big-endian to a little-endian format?

1. REV R0 R0
2. RRX R0 R0
3. ROR R0 R0 16
4. REV16 R0 R0

**Question 7:**

You execute the following two instructions:

LDR R0, [R1, #8]!

LDR R2, [R1], #8

After that

1. R0 contains the value stored in memory address R1+8 and R2 contains the value stored in memory address R1+16.
2. both R0 and R2 contain the same value, which was stored in memory address R1+8.
3. both R0 and R2 contain the same value, which was stored in memory address R1+16.
4. None of the above.

**Question 8:**

You execute the following instruction:

RSBS R0, R1, #1024

The result is

1. R0 = R1 - (1024+ R1)
2. R0 = R1 - (R0 + 1024)
3. R0 = R1 - 1024
4. R0 = 1024 - R1

**Question 9:**

You execute the following instruction:

ADD R0, R1, R1, LSR #1

The result is

1. R0 = R1 \* 3
2. R0 = R1 \* 1.5
3. R0 = R1 / 2
4. R0 = R1 \*2

**Question 10:**

You execute the following instruction:

ADR R0, label

1. it gets the address (relative to the PC) pointed by label and loads the value stored in that memory address in R0.
2. it gets the address (relative to the PC) pointed by label and adds the value stored in that memory address to the value in R0.; then stores the result in R0.
3. it gets the address (relative to the PC) pointed by label and loads the address value in R0.
4. None of the above.

**Question 11:**

There are two ways of coding flow control in the Arm state:

1. Append a two letter conditional suffix to the mnemonic or use the IT (If-Then) instruction.
2. Use a conditional branch instruction or use the IT (If-Then) instruction.
3. Append a two letter conditional suffix to the mnemonic or use a conditional branch instruction.
4. None of the above.

**Question 12:**

Which instruction would you use to move a value from a coprocessor register to an Arm register?

1. MCR.
2. MRC.
3. MVT.
4. It is not allowed moving values between Arm registers and coprocessor registers. That causes an Undefined Instruction exception.

**Question 13:**

Suppose the contents of registers R0 and R1 are 0x37DAFF07 and 0x25DDFD2A, and you execute the following instruction:

QADD16 R0, R1

The result would be

1. R0 = 0x37DAFFFF
2. R0 = 0x5DB8FC31
3. R0 = 0xFFFFFC31
4. R0 = 0x37DAFC31

**Question 14:**

Suppose the content of register R0 is 0x003B05DD, and you execute the following instruction:

CLZ R1, R0

The result would be

1. R1 = 0x00000000
2. R1 = 0x00000008
3. R1 = 0x0000000F
4. R1 = 0x0000000A

**Question 15:**

Suppose the content of register R1 is 0x00002AD3, and you execute the following instruction:

SSAT16 R0, #14,

The result would be

1. R0 = 0xFFFF2AD3 and Q =1
2. R0 = 0xFFFFFFFF and Q=1
3. R0 = 0x00002AD3 and Q=0
4. R0 = 0x00002FFF and Q=1

**Answers**

Q1)4

Q2)3

Q3)2

Q4)4

Q5)4

Q6)1

Q7)2

Q8)4

Q9)2

Q10)3

Q11)3

Q12)2

Q13)1

Q14)4

Q15)3