Type: MCQ

Q1. _____ instruction perform bitwise operations (0.5)

1. Arithmetic
2. **Logical
3. Compare
4. all

Q2. The value of Y, If X = 1111 0100 and Y = (X Arithmetic Shift Right by 2 bits) is ____ (0.5)

1. 1111 1010
2. 1110 1000
3. **1111 1101
4. 1111 0010

Q3. The values of C and Z flags after executing the following code snippet are ____, _____respectively.

```
LDR R0, =0xFFFFFFFD
ADDS R0, R0, #3
ADC R1, R0, #0x0
```
(0.5)

1. ** C:1, Z:1
2. C:1, Z:0
3. C:0, Z:1
4. C:0, Z:0

Q4. The BEQ instruction is used _____ (0.5)

1. To check the equality condition between operand and then branch
2. To check if the operand is greater than the condition value and then branch
3. **To check if the flag Z is set to 1 and then causes branch
4. None of the above

Q5. _____ is used to rotate the R0 register contents to left by 24 (0.5)

1. ROL R0, R0, #24
2. ROR R0, R0, #24
3. **ROR R0, R0, #8
4. ROL R0, R0,

Q6. The content of register R0 in decimal after the execution of the instruction, **MOV R0, R0, LSR 2** if R0 = 0x00000400 is _____

(0.5)

1. 512
2. **256
3. 2048
4. 128

Q7. The alternative name for STMED is _____ (0.5)

1. STMIB
2. STMIA
3. **STMDA
4. STMDB

Q8. _____ instruction is used to branch unconditionally (0.5)

5. **BAL
6. BA
7. BB
8. BL

Q9. The value of R0 after the execution of the instruction, **RSB R0, R2, R1** if R1=0 and R2= 0X00000033 is ____ (0.5)

1. **0XFFFFFFCD
2. 0XFFFFFECD
3. 0X00000033
4. 0X000000CD

Q10. The content of Stack pointer after the execution of the instruction, **STMFD R13!, {R1, R4}** if R13= 0X00080014 is ____ (0.5)

1. 0X00080018
2. **0X0008000C
3. 0X00080010
   0X0080001C

Type: DES

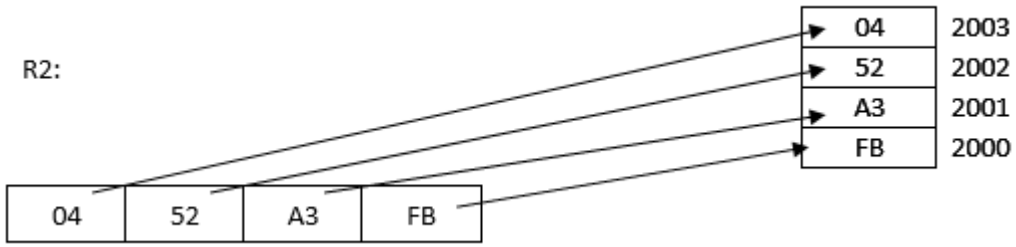
Q11. Analyze the following code snippet and draw the contents of the register R4 and the 32 bit content of the memory location pointed by R1 after executing the following code snippet. (2)

LDR R1, =0x2000

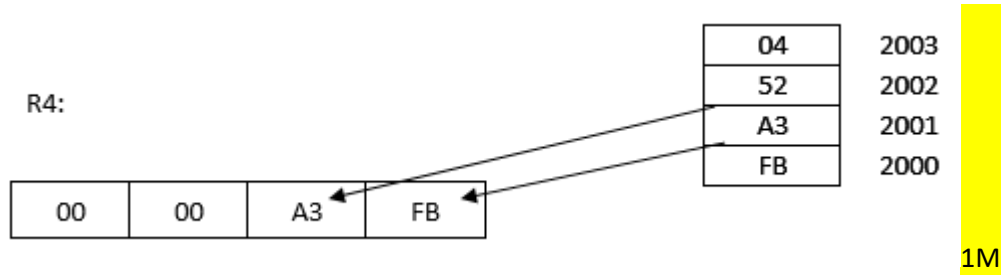
LDR R2, =0x452A3FB

LDR R3, =0xFF55AA7

STR R2, [R1]

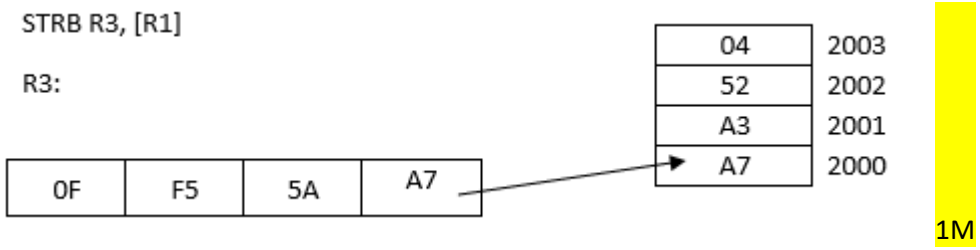


LDRH R4, [R1]



1M

STRB R3, [R1]



1M

Q12. Write a code snippet using ARM CortexM3 instruction set to count the number of even elements in an array of ten 32 bit numbers (2)

```
        LDR R0, = data   ; Pointing to array
        LDR R6, = Rslt   ; Pointing to result
        MOV R2, #0       ; Initialize even no
        MOV R3,#0X09 ; Initialize counter                    (0.5M)
L1      LDR R1, [R0], #4; Loading array elements
        ANDS R1, #01     ; 'AND'ing to check whether number is odd or even (0.5M)
        BEQ JUMP         ; If odd branch to JUMP
        ADD R2,#01       ; else increment R2 to count the even number of elements (0.5M)
JUMP    SUBS R3,#1       ; Decrement counter for array
        BNE L1           ; Repeat till all elements are done
        STR R2, [R6]     ; Store the result                  (0.5M)
STOP B STOP
```

Q13. Identify the special function registers from the register set R0-R15 in ARM Cortex M3 processor and explain their functionalities. (3)

Special function registers are R13, R14 and R15.

R13: Stack pointer - Always points to the top of stack. Holds the address of the top of stack. Top of stack is the last filled location or the empty location where the next data will be stored depending on the type of stack-full or empty respectively.

R14: Link register - Holds the return address during subroutine calls. When the instruction for call to a subroutine is executed, the address of the next instruction (PC content) is copied to link register.

R15: Program counter – Holds the address of the next instruction to be executed. When an instruction is fetched for execution, it is incremented such that it points to the next instruction.

1 mark for each register

Q14. Identify the stack type used by PUSH, POP instructions and explain that type of stack. What is the content of R13, R1, R2 and R3 after the execution of each of the following instructions? Refer Table P14 for the initial data.

1. LDMIA R13, {R1,R2,R3}
2. LDMIB R13, {R1,R2,R3}
3. LDMDA R13, {R1,R2,R3}
4. LDMDB R13, {R1,R2,R3}

Table P14

	Address	Data
	0x010	10
	0x014	20
	0x018	30
	0x01C	40
R13 →	0x020	50
	0x024	60
	0x028	70
	0x02C	80
	0x030	90
	0x034	100

(3)

The stack structure used by ARM for PUSH, POP instructions is a **Full Descending stack**. To store data the stack pointer should be decremented first and then the data should be pushed. While retrieving, data should be popped first and then the stack pointer should be incremented. STMDB and LDMIA pair of instructions can be used. 1M

1. R13= 0X020, R1=50, R2=60, R3=70 0.5M
2. R13= 0X020, R1=60, R2=70, R3=80 0.5M
3. R13= 0X020, R1=30, R2=40, R3=50 0.5M
4. R13= 0X020, R1=20, R2=30, R3=40 0.5M