Name: **A**

Roll Number:

IT 209: Computer Organization
31st August, 2018 (1st Mid-Term)

Total Marks: 105
Time: 2 hours

Note: ANSWER ALL THE QUESTIONS IN THE SPACE PROVIDED IN THE QUESTION PAPER ITSELF.
DO ALL YOUR ROUGH WORK IN THE SUPPLEMENTARY SHEET PROVIDED.
THE SUPPLEMENTARY SHEET WILL NOT BE EVALUATED.
Write only answers. Give justification/explanation only if asked in the question.
All questions are compulsory.
Question no. 1 to 32 carry 1 mark each. Question no. 33 to 53 carry 3 marks each. Questions 54 and 55 carry 5 marks each.

1 For the below ARM assembly code, trace the values as it executes that will be placed into the registers PC, R0, R1, and R2. (1 Mark)

```
MOV R0, #0      [Address 0x0000]
MOV R1, #5
ADD R0, R0, R1
SUB R1, R1, #1
CMP R0, #10
MOVGT PC, #28
MOV PC, #8
MOV R2, #1
HALT  B HALT        [Address 0x0020]
```

2 For given below program, write an optimized ARM Assembly Language Program by reducing the number of instructions. (1 Mark)

```
…..
CMP r0,#5
BEQ BYPASS
ADD R1,R1,R0
SUB R1,R1,R2
BYPASS…..
…..
```

3 What will be the value of the overflow bit V after executing the following set of instructions? (1 Mark)

```
LDR R0,= -0x60000000
LDR R1,= -0x70000000
ADDS R2,R1, R0
```

4 What will be the contents of register R0 after the execution of the following set of instructions? (1 Mark)

```
MOV R0, #0X05
LSL R0,R0,#1
LDR R1,=0xA0000000
ORR R0,R0,R1
ASR R0,R0,#1
```

5 What will be the contents of register R0 after the execution of the following set of instructions? (1 Mark)

```
LDR R0, =0x11111111
LDR R1, =0x01100101
BIC R2,R0,R1
```

6 What will be the contents of registers R0 & R1 after the execution of the following set of instructions? (1 Mark)

```
MOV R0, #5
MVN R1, #5
```

7 What will be the contents of registers R0 & R1 after the execution of the following set of instructions? (1 Mark)

```
MOV R0, #0X00000031
MOV R1, R0, ROR #2
MOV R2, R0, RRX
```

8 What will be the contents at location 0x00004000 after the execution of the following set of instrucutions? (1 Mark)

```
LDR R5, =0X4000
MOV R1, #0X9
MOV R2, #0X10
MOV R3, #0X2
ADD R0, R1, R2, LSL R3
STR R0, [R5]
```

9 What will be the contents at register R0 & R1 after the execution of the following set of instrucutions? (1 Mark)

```
        MOV R0, #7
        MOV R1, #45
AGAIN   CMP R0, R1
        SUBGT R0, R0, R1
        SUBLT R1, R1, R0
        BNE AGAIN
HALT    B HALT
```

10 What will be the value of R1 and R2 after the following code runs? (1 mark)

```
AREA DOT, CODE, READONLY
 MOV R0, #40     ; R0 is a
 MOV R1, #25     ; R1 is b
again   CMP R0, R1
 BLT isLess
 SUB R0, R0, R1
 B stop
isLess  SUB R1, R1, R0
stop
 END
```

11 Fill in the blank with proper mnemonics in the following code such that "02" is stored at 0x2000 and "1" at 0x2004. (1 mark)

```
area prog6,code,readonly
 entry
 mov r0,#5
 mov r1,#2
 mov r2,#-1
 mov r3,#0x2000
 ldr r4,=0x2004
loop add r2,r2,#1
 _______  r0,r0,r1
 bpl loop
 add r0,r0,r1
 str r0,[r4]
 str r2,[r3]
 end
```

12 Force the program to end WHEN A ZERO IS FETCHED by filling the blank with appropriate instruction. (1 mark)

```
area prog7,code,readonly
entry
mov r4,#0x1000
datadcd 1,0,3,1,2,3,1,2,3,1

ldr r0,=data
loop ldr r2,[r0],#4
 cmp r2,#0
 beq done
 add r1,r1,#1
 bal loop
done str r1,[r4]
_____________________
 End
```

13 What will be the value in register r1. (1 mark)

```
MOV r0,#0x2
 MOV r1,r0,ROR r0.
```

14 What will be the effective address and final address after the following instruction is executed. (1 mark)

```
Ldr r2,[r1,r0,lsl #2]!
```

15 For the following instruction, what will be the address of memory from which value of r1will be taken. (1 mark)

```
Ldr r1,[r0],r2
```

16 In the code below, fill in the blanks by using appropriate indexing to swap the two numbers stored at a particular location. (1 mark)

```
ttl swap 2 numbers stored in memory
area prog1,code,readonly
 entry
 ldr r1, =data
 ldr r2, data
 ldr r3,_____
 str r2,[r1]
 str r3,_____

datadcd 2,4
 end
```

17 After execution of the following code, will the memory address 0X5000 have value 0xFFF1 at or not? If not, then replace the wrong instruction with the correct instruction in the code. (1 Mark)

```
AREA PROGRAM,CODE,READONLY
ENTRY
LDR R1,=0X5000
MOV R2,#0XFFF1
STR R2,[R1]
END
```

18 After executing the following code,will the value in Register R3 be 0X1 or not? If the value in Register R3 is not 0x1, then correct the instruction in the code which is producing wrong result.
NOTE: Do not change any opcode in the code. (Eg. For LDR R0,R1 don't replace LDR with some other opcode) (1 Mark)

```
AREA PROGRAM,CODE,READONLY
      ENTRY
            MOV R1,#64
            MOV R2,#0X64
MOV R3,#0
            CMP R2,R1
      BEQ  B1
      B  STOP
B1       MOV R3,#1
STOP
      END
```

19 After executing the following code, check whether output at Register R1 is 0x21212121 or not? If Not, then what is the correct output? (1 Mark)

```
AREA PROGRAM,CODE,READONLY
      ENTRY
      LDR R1,=0x42424242
      LSL R1,R1,#1
      END
```

20 After executing the following code, check whether output at Register R1 is 0xDCA9BFFF or not? If not, then what is the correct output? (1 Mark)

```
 AREA PROGRAM,CODE,READONLY
      ENTRY
      LDR R1,=0X235648AB
      LDR R2,=0XFFFFFFFF
      EOR R1,R1,R2
      END
```

21 After executing the following code, check whether value at Register R3 is 0x120 or not? If not, then what is the correct value. (1 Mark)

```
MUL R3,R3,R1
    SUBS R1,R1,#1
    BNE B1
STOP
LABEL DCD 5
    END
```

22 After executing the following code, check whether value at Register R3 is 0x429A9A9A or not ? If not, then what is the correct value? (1 Mark)

```
AREA program,CODE,READONLY
    ENTRY
    LDR R0,=0X85353535
    ASR R3,R0,#1
    END
```

23 Is there any correction needed in the following program to store numbers 1 to 10 in 10 consecutive memory locations starting from address 0x4000? If yes, then rectify. NOTE: Do Not change any instruction opcode. (1 Mark)

```
AREA PROG, CODE, READONLY
    ENTRY
    MOV R0,#0X10
    MOV R1,#0X1
    MOV R2,#0X4000
LOOP    STR R1,[R2]
    ADD R1,R1,#1
    ADD R2,R2,#4
    SUBS R0,R0,#0X1
    BNE  LOOP
    END
```

24 Is there any correction needed in the following program to store numbers declared in 'data' in consecutive memory locations starting from address 0x4000? If yes, then rectify. (1 Mark)

```
AREA PROG, CODE, READONLY
    ENTRY
DATA    DCD 0X1,0X2,0X3
    MOV R0,#0X3
    LDR R1,=DATA
    MOV R2,#0X4000
LOOP    LDR R3,[R1]
    STR R3,[R2]
    ADD R1,R1,#4
    ADD R2,R2,#4
    SUBS R0,R0,#0X1
    BNE LOOP
    END
```

25 Is there any correction required in the following code which stores numbers declared in 'data' in consecutive memory locations starting from address 0x4000? If yes, then rectify. NOTE: Do not make changes in any opcode of the instructions. (1 Mark)

```
 AREA PROG,CODE,READONLY
      ENTRY
      MOV R0,#0X4
      LDR R1,=DATA
      MOV R2,#0X4000
LOOP  LDR R3,[R1,#4]
      STR R3,[R2]
      ADD R2,R2,#4
      SUBS R0,R0,#0X1
      BNE LOOP
DATA  DCD 0XA,0XB,0XC,0XD
      END
```

26 Is there any correction needed in the following program which stores the multiplication of 5 and 4 in R2? If yes, then rectify. (1 Mark)

```
 AREA PROG, CODE, READONLY
ENTRY
      MOV R0,#5
      MOV R1,#4
      MOV R2,#0
LOOP  ADD R2,R2,R0
      SUB R1,R1,#1
      BNE LOOP
      END
```

27 Is there any correction needed in following program which stores numbers declared at odd places in DATA(DCD) in consecutive memory locations starting from address 0x4000?
If yes, then rectify the particular instruction.
NOTE: 1.Index of 'Data' starts from 1.
2. Do not change any opcode in the code. (Eg. For LDR R0,R1 , don't replace LDR with some other opcode) (1 Mark)

```
AREA PROG, CODE, READONLY
      ENTRY
      MOV R0,#0X3
      LDR R1,=DATA
      MOV R2,#0X4000
LOOP  LDR R3,[R1],#4
      STR R3,[R2]
      ADD R2,R2,#4
      SUBS R0,R0,#0X1
      BNE LOOP
DATA  DCD 0X0,0X1,0X2,0X3,0X4,0X5
      END
```

28 Is there any correction needed in the following program which finds the greatest between two numbers and stores the greater number at 0x4000? If yes, then rectify the particular instruction. (1 Mark)

```
AREA PROG, CODE, READONLY
    ENTRY
    MOV R0,#0X5
    MOV R1,#0X4
    MOV R2,#0X4000
    CMP R0,R1
    BEQ DONE
    MOV R0,R1
DONE   STR R0,[R2]
    END
```

29 Is there any correction needed in the program below which finds the number of 1's and stores the result in R3? If yes, then rectify the particular instruction. (1 Mark)

```
AREA PROGRAM,CODE,READONLY
    ENTRY
    LDR R0,=0XF
    LDR R3,=0X00
    MOV R1,#0X20
LABEL MOVS R0,R0,RRX
    ADDCC R3,R3,#1
    SUB R1,R1,#1
    CMP R1,#00
    BNE LABEL
    END
```

30 Is there any correction needed in the following program which counts the length of a string and stores the result in R1? If yes, then rectify the particular instruction.
NOTE: Do not change any opcode in the code. (Eg. For LDR R0,R1 , don't replace LDR with some other opcode. (1 Mark)

```
AREA PROGRAM,CODE,READONLY
    ENTRY
    LDR R0,=DATA
    MOV R1,#0
LOOP
    LDRB R2,[R0],#1
    CMP R2,#0X0D
    BEQ STOP
    ADD R1,#1
    B LOOP
STOP B STOP
DATA DCB "DAIICT"
    END
```

31 Is there any correction needed in the following program which transfers 10 numbers (32 bit) from 0x5000 to 0x8000? (Assume at 0x5000 already some numbers are present). If yes, then rectify the particular instruction. (1 Mark)

```
AREA PROGRAM,CODE,READONLY
    ENTRY
    LDR R0,=0X5000
    LDR R11,=0X8000
    LDMIA R0,{R1-R10}
    STMDB R0,{R1-R10}
    END
```

32 Is there any correction needed in the following piece of code for calculating the half adder sum(A'B'+AB) which takes input in R1 and R2 and stores the sum in R5? If yes, then rectify. (1 Mark)

```
AND R3,R1,R2
    MVN R1,R1
    MVN R2,R2
    AND R4,R1,R2
    ROR R5,R3,R4
```

33 Fill in the blanks in the following piece of code to modify the content of CPSR by changing the control field. (3 marks)

```
area mode,code,readonly
 entry
 mov r2,_____________      ; move immediate data 0x10 to r2
 _______ r1,cpsr           ; use appropriate op-code
 _______   _______,r2      ; use appropriate opcode and status register field suffix
 End
```

34 Fill in the blanks by using branching with appropriate conditional suffixes to store the number of positive numbers at location 0x6000. (3 marks)

```
area prog7,code,readonly
 entry
 mov r0,#1
 ldr r1,=0xF00F00F
 ldr r2,=0XEFEFFFA
 ldr r3,=0xFFFFFCD
 ldr r4,=0x0FF0FF0
 mov r5,#0x5000
 mov r6,#0x4
 mov r8,#0x0
 stm r5,{r1,r2,r3,r4}
 ldr r10,=0x6000

look  subs r6,r6,#1
 ________list
 ldr r7,[r5],#4
 cmp r7,r8
 ________loop
 ________look
 ________look
```

35 Perform the operation as indicated by the code below, without using the PUSH and POP. (3 marks)

```
area prog2,code,readonly
 entry
 ldr r13,=label
 pop {r1-r7}     ……(1)
 ldrsp,=0x4000
 push {r1-r7}    ……(2)
label dcd 1,2,3,4,5,6,7
 end
```

36 Fill in the blanks with proper instructions so that first function 1 (func 1) is executed <u>without</u> entering the stop loop and then function 2 (func 2) is executed followed by stop loop at the end. (3 marks)

```
AREA    Loadcon, CODE, READONLY
    ENTRY                   ; Mark first instruction to execute
start
 _____________________
    _____________________
stop
    MOV    r0, #0x18
    LDR    r1, =0x20026
    SWI    #0x123456        ; ARM semihosting
func1
    LDR    r0, =42
 LDR    r1, =0x55555555
    LDR    r2, =0xFFFFFFFF
    _____________________
    LTORG
func2
    LDR    r3, =0x55555555
    ; LDR r4, =0x66666666
```

37 Create a stack using <u>ldm/stm with proper suffix</u>, such that, data 1 is stored at location 0x6004 and data 6 at respective higher address, with stack pointer(SP) pointing at the latest data i.e., 6. (3 marks)

6
5
4
3
2
1

38 Use any mnemonics from the LSL,LSR,ASR,ROR or RRX with suitable suffix to first multiply the number 0x9 by 8 and then determine the number of 0's in the final 32 bit number. (3 mark)

```
area prog3,code,readonly
 entry
 mov r4, #32
 mov r0,#0X9
 mov r1,#0x5000
 mov r2,#0x4000
 mov r3,#0
 mov r5,#0
 str r0,[r1]
 LSL R0,R0,#3          ;……(i)
list rrxs r0,r0        ;……(ii)
 bcs loop
 subs r4,r4,#1
 cmp r4,#0
 bne list
 b stop
loop add r3,r3,#1
 subs r4,r4,#1
 cmp r4,#0
 bne list
```

39 Which instruction can replace all the four instructions? (3 marks)

```
Ldr r0,[r9]
Ldr r1,[r9,#4]
Ldr r2,[r9,#8]
Ldr r3,[r9,#c]
```

40 For the following code, show the registers placement in the memory using IA for load and DA for store, and similarly IB for load and DB for store.
(Assume any initial address in R10 e.g. 0x0C). (3 marks)

```
Ldr r10,=data
Ldmxx r10,{r0,r1,r4}
Stmxx r10,{r0,r1,r4}.
Data dcd 1,2,3,4,5,6
```

41 Describe the function of following subroutine. What is the maximum possible size of the stack? Justify. (3 Marks)

```
SUB STR LR,[SP,#-4]!
STR R0,[SP,#-4]!
MOVS R0,R0,LSR #1
ADDCS R1,R1,#1
BLNE SUB
LDR R0,[SP],#4
LDR PC,[SP],#4
```

42 Following memory is given, write the contents of each register (R0,R1 and R2) after program execution completes, assume R1=0x0001000 and R2=0x00000004 (use Little Endian) (3 Marks)

```
23   13   56   00   45   11   22   88   03   08   35   89   44   93
1000                 1004                 1008                 100C
```

```
LDR R0, [R1]
LDR R0, [R1, R2]
LDR R0, [R1], #4
LDR R0, [R1, R2]!
```

43 Assume that the Stack Pointer is initialized to 0X00002408. What will be the content of R0 & SP after execution of program? (3 Marks)

```
LDR R0,=DATA
LDR SP,=0X00002408
LDMIA R0!,{R2-R9}
STMDB SP!,{R2,R5,R4}
STMIA SP,{R6,R7,R8}
PUSH {R9}
DATA  DCD 1,2,3,4,5,6,7,8
```

44 What are the condition code bits after each of the following ARM instructions are executed in sequence? (3 Marks)

INSTRUCTION	*N*	*Z*	*V*	*C*
MOV R0 , #1				
LSRS R0 ,#30				
SUBS R0 , #1				
CMP R0 , #4				
ADD R0 , #1				
CMP R0 , #3				

45 What will be the contents of R1, R2, R9, R10, R11 &R12 after execution of the following program? (3 Marks)

```
MOV R0, #0
LDR R1,=DATA
LDR R2,=STCK
LDMIA R2!,{R6,R8}
LDR R9, [R1, #12]
LDRB R10, [R1], #4
LDRH R11, [R1], #8
LDR R12, [R1, #8]!
DATA  DCD 0X11223344,0X22334455,0X33445566,0X44556677,0X55667788,0x66778899
STCK  DCD 0X1234,0X5678
```

46 Consider the following code snippet. What values will be stored at memory addresses 0x701C and 0x7010? Justify. (answer in big endian format) (3 marks)

```
  ldr sp,=0x7020
  ldr r1,=8
  ldr r7,=0
 lp push {r7}
  add r7,r7,#1
  subs r1,r1,#1
  pop {r9}
  beq chk
  b lp
chk b chk
```

47 For the code below, what values are stored at addresses 0x3000 and 0x3004? (answer in big endian format) (3 marks)

```
 ldr r1,=1
 ldr r2,data
 ldr r3,=0x10
 ldr r7,=0x3000
 ldr r8,=0x3004
 ldr r5,=0
 ldr r6,=0
loop  ands r4,r3,r1
 beq pt1
 bne pt2
lp add r1,r1,#1
 subs r2,r2,#1
 bne loop
 beq chk
pt1 add r5,r5,r4
 b lp
pt2 add r6,r6,r4
 b lp
chk str r5,[r7]
 str r6,[r8]
data dcd 30
```

48 In the following code snippet, what is the value stored at address 0x4000? What is the code trying to do (what does the value in register r4 signify)? (answer in big endian format) (3 marks)

```
  ldr r0,=0x4000
  ldr r1,=data
  ldr r2,=0x7
  ldr r3,=0x3
  ldr r4,=0
  ldr r6,=6
  ldr r5,[r1],#4
loop  sub r5,r5,r3
  sub r5,r5,r2
  subs r5,r5,r3
  beq pt1
  bpl loop
pt2 ldr r5,[r1],#4
  subs r6,r6,#1
  bne loop
  beq chk
pt1 add r4,r4,#1
  b pt2
chk str r4,[r0]
data dcd 26,25,93,65,78,90
```

49 In the code snippet below, What is the value stored at address 0x5000? What is the value at 0x5000 if value of register r2 is set as 3? (answer in big endian format) (3 marks)

```
  Ldr r0,data
  Ldr r1,=1
  ldr r2,=4
ldr r4,=0x5000
loop  add r5,r2,r1
lsr r0,r0,r5
add r1,r2,r1
subs r2,r2,#1
 bne loop
str r0,[r4]
Data dcd 0xabfcd95e
```

50 In the code snippet below, what is the value stored at address 0x5020? (answer in big endian format) (3 marks)

```
 ldr r0,=data
 ldr r1,=0x5000
 ldmia r0,{r5-r9}
 add r8,r9,r8
 add r7,r8,r7
 add r6,r6,r7
 add r5,r5,r6
 stmia r1!,{r5-r9}

ldr r8,=5
 ldr r9,=0x5020
loop ldrb r7,[r1],#-3
 subs r8,r8,#1
 bne loop
 str r7,[r9]

data dcd 3,5,7,2,1
```

51 After running following code , pair up the registers (out of r0,r1,r2,r3,r5,r7) which have the same output.

Pair 1:---------- , ----------
Pair 2:---------- , ----------
Pair 3:---------- , ----------

```
AREA DOT, CODE, READONLY
LDR r0,=num
LDR r1,num
ADD r2,r2,#num
LDR r3,[r0]
LDR r4,num
LDR r5,[r4,#4]
LDR r6,[r4,#4]!
LDR R7,[R4],#4
num DCD 10
END
```

52 Consider the below program and check whether output at Register R3 is 0x00303030 or not? If not then what is the correct output. (3 Mark)

```
AREA program,CODE,READONLY
    ENTRY
    LDR  R0,=0X353535
    LDR  R1,=0x555555
    BIC  R3,R0,R1
    END
```

53 Consider the below program which counts the number of times the "I" alphabet occurs in the string "DAIICT" and stores the result in R1. Is there any error in code?
If any then correct the code. (ASCII value of "I" is 0x6C) (3 Mark)

```
AREA PROGRAM,CODE,READONLY
    ENTRY
    LDR R0,=DATA
    LDR R3,=0x6C
    MOV R4,#0x49
LOOP
    LDRB R2,[R0],#1
    CMP R2,#0X0D
    BEQ STOP
    CMP R2,R4
    BEQ CHECK
    B LOOP
CHECK
    ADD R1,#1
STOP B STOP
DATA DCB "DAIICT\r"
    END
```

54 Consider a five stage pipeline (F, D, Ex, M, WB) with full data forwarding. Assume that each stage takes 10ns to finish. Assume that you are executing a program where a fraction "f" of all instructions immediately follow a load upon which they are dependent. Based on this information, answer following question. You are required to show detailed calculation. (5 Marks)

With forwarding enabled and ignoring the cycles required to initially fill the pipeline, what would be the total execution time for N instruction in terms of "f"?

55 Given four instructions, how many unique comparisons (between register sources and destinations) are necessary to find all of the RAW, WAR, and WAW dependences? Assume that all four instructions have two source and one destination registers. You are required to show detailed calculation. (5 Marks)