

**Department of Computer Science & Engineering**

**Microprocessor & Computer Architecture**

**MPCA-Laboratory/Assignment/Hands-on/Project**

**UE20CS252**

| **Sl. No** | **Programs** |
| --- | --- |
| **Week No.2** | 1. Write a program in ARM7TDMI-ISA to copy a block of N data items from Location A to Location B. { do not use LDM / STM instructions).  a. Use Full word (.word directive)  b. Use Half word(.Hword directive)  c. Use Byte wise (.Byte directive)  **a. Use Full word (.word directive)**  ; Program to transfer a block of data from location A to Location B.  .DATA  A: .WORD 10,20,30,40,50,60,70,80,90,100  B: .WORD 0,0,0,0,0,0,0,0,0,0  .TEXT  LDR R1,=A ; INITIALISATION of the block addresses  LDR R2,=B    MOV R5,#1 ; COUNT register  L1: LDR R3, [R1]  STR R3, [R2]  ADD R1, R1, #4 ; address to the next data  ADD R2, R2, #4    ADD R5, R5, #1 ; increment the count register  CMP R5, #11 ; Check whether all numbers are added  BNE L1 ; Else branch to L1 location  SWI 0X011 ; logical end of the program  **b. ;Program to transfer a block of data from location A to Location B using Half word**  .DATA  A: .HWORD 10,20,30,40,50,60,70,80,90,100  B: .HWORD 0,0,0,0,0,0,0,0,0,0  .TEXT  LDR R1,=A ; INITIALISATION of the block addresses  LDR R2,=B    MOV R5,#1 ; COUNT register  L1: LDRH R3, [R1]  STRH R3, [R2]  ADD R1, R1, #2 ; address to the next data  ADD R2, R2, #2    ADD R5, R5, #1 ; increment the count register  CMP R5, #11 ; Check whether all numbers are added  BNE L1 ; Else branch to L1 location  SWI 0X011 ; logical end of the program    **B**  **C : Program to transfer a block of data from location A to Location using BYTE**  .DATA  A: .byte 10,20,30,40,50,60,70,80,90,100  B: .byte 0,0,0,0,0,0,0,0,0,0  .TEXT  LDR R1,=A ; INITIALISATION of the block addresses  LDR R2,=B    MOV R5,#1 ; COUNT register  L1: LDRB R3, [R1]  STRB R3, [R2]  ADD R1, R1, #1 ; address to the next data  ADD R2, R2, #1    ADD R5, R5, #1 ; increment the count register  CMP R5, #11 ; Check whether all numbers are added  BNE L1 ; Else branch to L1 location    SWI 0X011 ; logical end of the program  2. Write a program in ARM7TDMI-ISA to find the sum of N data items in the memory. Store the result in the memory location.  a. Use Full word (.word directive)  b. Use Half word(.Hword directive)  c. Use Byte wise (.Byte directive)  **a. Program: Use Full word (.word directive)**  **– This program is from the class slide).**  **.DATA**  A: .WORD 10,20,30,40,50,60,70,80,90,100  SUM: .WORD 0  **.TEXT**  LDR R1,=A  LDR R2,=SUM  MOV R4,#0 ; INITIALISATION  MOV R5,#1 ; COUNT register  L1: LDR R3, [R1]  ADD R4,R4,R3 ; Add next element in the array.  ADD R1, R1, #4 ; address to the next data  ADD R5, R5, #1 ; increment the count register  CMP R5, #11 ; Check whether all numbers are added  BNE L1 ; Else branch to L1 location  STR R4,[R2] ; store the result in location SUM.  SWI 0X011 ; logical end of the program.  **b. Use Half word(.Hword directive)**  Program to find the sum of N numbers using **half word**  .DATA  A: **.HWORD** 0x10,0x20,0x30,0x40,0x50,0x60,0x70,  0x80,0x90, 0x0100  SUM: .WORD 00    .TEXT  LDR R1,=A  LDR R2,=SUM  MOV R4,#0 ;INITIALISATION  MOV R5,#1 ;COUNT  L1: **LDRH R3,[R1]**  ADD R4,R4,R3  ADD R1,R1,#2  ADD R5,R5,#1  CMP R5, #11  BNE L1  **STRH R4, [R2]**  SWI 0X011  **C. Use Byte( .byte directive )**  Program to find the sum of N numbers using **Byte Data**  ; SUM OF N NUMBERS  ; DATA GIVEN  .DATA  A: **.BYTE** 1,2,3,4,5,6,7,8,9,10  SUM: .word 0  .TEXT  LDR R1,=A  LDR R2,=SUM  MOV R4,#0 ;INITIALISATION  MOV R5,#1 ;COUNT  L1: **LDRB R3,[R1]**  ADD R4,R4,R3  ADD R1,R1,#1  ADD R5,R5,#1  CMP R5, #11  BNE L1  **STRB R4,[R2]**  SWI 0X011  **3. Write a program in ARM7TDMI-ISA to find the sum of N natural numbers. Store the result in the memory location.**  .TEXT  LDR R2,=SUM    MOV R1,#1  L1: ADD R0,R0,R1  ADD R1,R1,#1  CMP R1,#11  BNE L1  STR R0,[R2]  SWI 0X011    .DATA  SUM: .WORD 0  **4. Write a program in ARM7TDMI-ISA to find the product of two 32bit numbers using barrel shifter.**  .DATA  A: .WORD 0X12345678  B: .WORD 65  C: .WORD 0    .TEXT  LDR R1,=A  LDR R2,=B  LDR R3,=C    LDR R4,[R1]  ADDS R5, R4, R4, LSL #6  STR R5,[R3]  SWI 0X011  .END  **5. Convert the following statement in C language into an ALP using ARM7TDMI – ISA.**  **IF([A]==[B]) then C=[A]+[B];**  **ELSE IF ([B]==[C]) D=[A]-[B];**  **ELSE E=[A]\*[B]**  **Where A,B C, D & E are memory locations.**  .DATA  A: .WORD 10  B: .WORD 20  C: .WORD 0  D: .WORD 0  E: .WORD 0    .TEXT  LDR R1,=A  LDR R2,=B  LDR R3,=C  LDR R4,=D  LDR R5,=E    LDR R6,[R1]  LDR R7,[R2]  LDR R11,[R3]    CMP R6,R7  ADDEQ R8,R6,R7  STREQ R8,[R3]    CMPNE R7,R8  SUBEQ R9,R6,R7    MULNE R10,R6,R7    SWI 0X011    .END  **6. Write a program in ARM7TDMI-ISA to find the factorial of a number.**  .DATA  A: .WORD 5 ; NUMBER  FACT: .WORD 0 ; FINAL RESULT LOCATION  .TEXT  LDR R1,=A  LDR R2,[R1]  LDR R4,=FACT  MOV R3,#1    LOOP: CMP R2,#0  BEQ EXIT  MUL R3,R2,R3  SUB R2,R2,#1  BNE LOOP    EXIT: STR R3,[R4]  SWI 0X011 |