LAB NO : 2 DATE : 15/01/2025

Title : PROGRAMS ON DATA TRANSFER INSTRUCTIONS

# Lab Exercise 1: Write an ARM assembly language program to store data into General Purpose Registers

## Code :

AREA RESET, DATA, READONLY

EXPORT \_\_Vectors

\_\_Vectors

DCD 0X10001000 ;stack ptr value when stack is empty

DCD Reset\_Handler ;reset vector. The program linker requires Reset\_Handler

ALIGN

AREA mycode, CODE, READONLY

ENTRY

EXPORT Reset\_Handler

Reset\_Handler

MOV R0, #20

STOP

B STOP

END

## Output :

# Lab Exercise 2: Write an ARM assembly language program to transfer a 32-bit number from one location in the data memory to another location in the data memory.

Source : SRC = 0X00000004 at location pointed by R0

Desination : DST = 0X00000002 at location pointed by R1 initally and DST = 0X00000004 after execution

## Code :

AREA RESET, DATA, READONLY

EXPORT \_\_Vectors

\_\_Vectors

DCD 0X10001000 ;stack ptr value when stack is empty

DCD Reset\_Handler ;reset vector. The program linker requires Reset\_Handler

ALIGN

AREA mycode, CODE, READONLY

ENTRY

EXPORT Reset\_Handler

Reset\_Handler

LDR R0, =SRC

LDR R1, =DST

LDR R2, [R0]

STR R2, [R1]

STOP

B STOP

SRC DCD 4

DST DCD 2

END

## Output :

# Lab Exercise 3: Write an ARM assembly language program to transfer block of ten 32-bit numbers code memory to data memory when the source and destination blocks are non-overlapping.

## Code :

AREA RESET, DATA, READONLY

EXPORT \_\_Vectors

\_\_Vectors

DCD 0x10001000 ; stack pointer value when stack is empty

DCD Reset\_Handler ; reset vector

ALIGN

AREA mycode, CODE, READONLY

ENTRY

EXPORT Reset\_Handler

Reset\_Handler

LDR R0, =SRC

LDR R1, =DST

MOV R2, #10

loop

LDR R3, [R0], #4

STR R3, [R1], #4

SUBS R2, #1

BNE loop

STOP

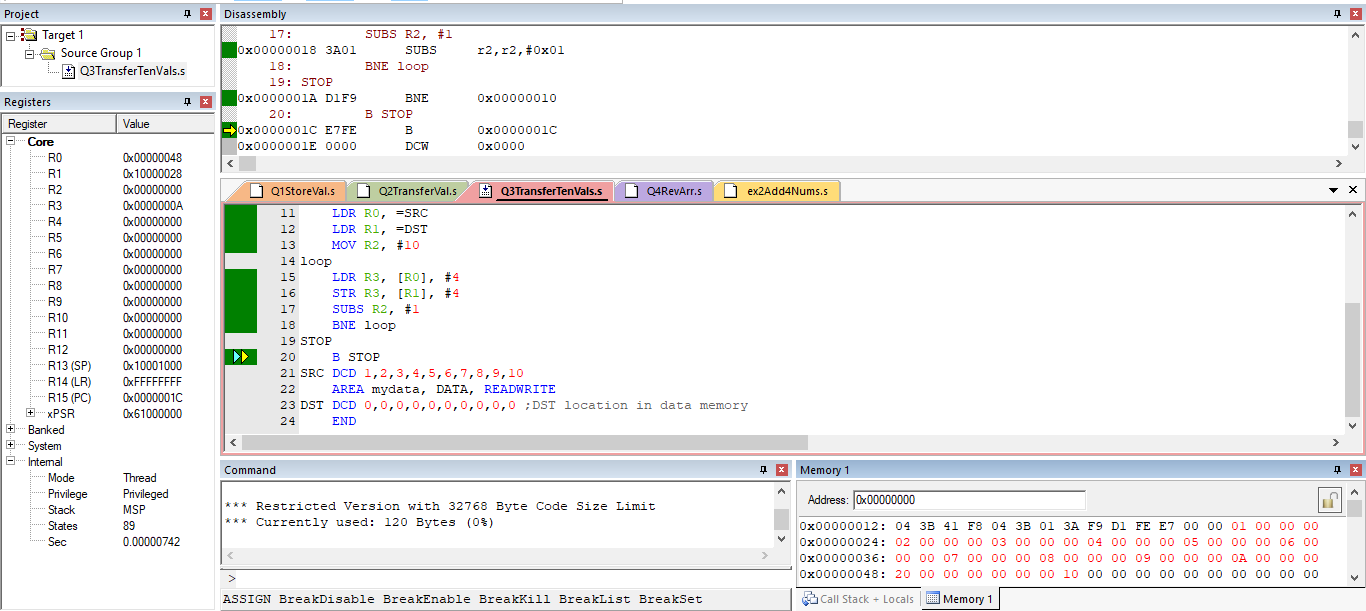
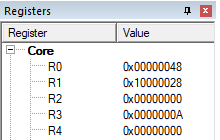
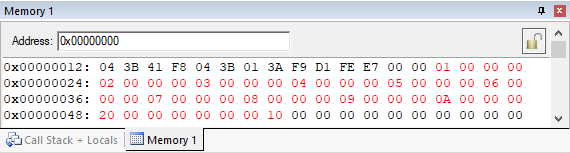
B STOP

SRC DCD 1,2,3,4,5,6,7,8,9,10

AREA mydata, DATA, READWRITE

DST DCD 0,0,0,0,0,0,0,0,0,0 ;DST location in data memory

END

Output :   

# Additional Exercise 1 : Reverse an array of ten 32-bit numbers in the memory

## Code :

AREA RESET, DATA, READONLY

EXPORT \_\_Vectors

\_\_Vectors

DCD 0x10001000 ; stack pointer value when stack is empty

DCD Reset\_Handler ; reset vector

ALIGN

AREA mycode, CODE, READONLY

ENTRY

EXPORT Reset\_Handler

Reset\_Handler

LDR R0, =ARR ;Beginning pointer to First/0th Element

ADD R1, R0, #36 ;Ending pointer to Last Element

MOV R2, #10 ;Counter

MOV R5, #1 ;Initalizing and Storing the array

loop1 STR R5, [R0], #4

ADD R5, #1

SUBS R2, #1

BNE loop1

MOV R2, #5 ;Counter reset to N/2 as 0 and n-1 are swapped together

LDR R0, =ARR

loop2 LDR R3, [R0] ;Loading elements from beginning

LDR R4, [R1] ;Loading elements from ending

STR R4, [R0], #4 ;Storing nth element and Incrementing the beginning pointer

STR R3, [R1], #-4 ;Storing 0th element and Decrementing the ending pointer

SUBS R2, #1 ;Decrementing Count

BNE loop2

STOP

B STOP

AREA mydata, DATA, READWRITE

ARR DCD 0,0,0,0,0,0,0,0,0,0 ;DST location in data memory

END

Output : 