**Experiment 3**

**Aim:**

To write an ARM Assembly Language to copy words from consecutive location and store it in consecutive destination locations.

1. Multiple register transfer instructions.
2. Load and Store instruction in a loop.

**Tool Used:**

Keil uVision4

**Theory:**

LDM load multiple register locations with starting address mentioned. ! is used in LDM for updating pointer, else same value will be updated in all registers. STM load the value into consecutive memory locations with starting address mentioned.

**a) Multiple register transfer instructions.**

**Code:**

AREA PROGRAM, CODE, READONLY

ENTRY

MAIN

  LDR R0, =0x00000000; SOURCE MEMORY LOCATION

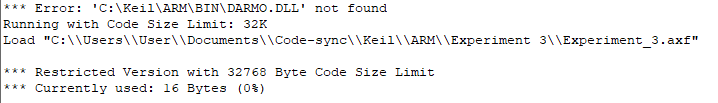
  LDR R1, =0x00000100; DESTINATION MEMORY LOCATION

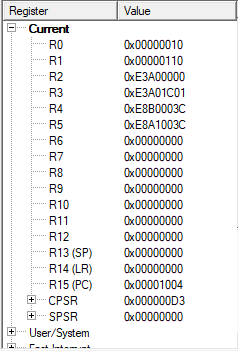
  LDM R0!, {R2-R5}; COPIES CONTINUOSLY FROM EXTERNAL TO R2 TO R5

  STM R1!, {R2-R5}; COPIES CONTINUOSLY FROM R2 TO R5 TO DESTINATION

  END

**Output:**

****

****

The input data to be copied

****

The data copied after running

****

4 words starting at 0x…000 is copied at destination address 0x…100.

**Using IA, IB, DA, DB**

**Code:**

  AREA PROGRAM, CODE, READONLY

ENTRY

MAIN

  LDR R0, =0x00000000; SOURCE MEMORY LOCATION

  LDR R1, =0x00000100; DESTINATION MEMORY LOCATION

  LDMIA R0!, {R2-R5}; COPIES CONTINUOSLY FROM EXTERNAL TO R2 TO R5

  STMIA R1!, {R2-R5}; COPIES CONTINUOSLY FROM R2 TO R5 TO DESTINATION

  END

**Output:**

For Increment after [IA]:

Input at 0x00000000



Input at 0x00000000



Value starts at 00 and 100 indicating that the increment is done after copying

For Increment before [IB]:

LDR R0, =0x00000000; SOURCE MEMORY LOCATION

  LDR R1, =0x000000F0; DESTINATION MEMORY LOCATION

LDMIB R0!, {R2-R5}; COPIES CONTINUOSLY FROM EXTERNAL TO R2 TO R5

  STMIB R1!, {R2-R5}; COPIES CONTINUOSLY FROM R2 TO R5 TO DESTINATION

Input at 0x00000000



Output at 0x00000000

****

Value starts at 04 and F4 instead of F0 indicating that the increment is done before copying

For Decrement After [DA]:

LDMDA R0!, {R2-R5}; COPIES CONTINUOSLY FROM EXTERNAL TO R2 TO R5

  STMDA R1!, {R2-R5}; COPIES CONTINUOSLY FROM R2 TO R5 TO DESTINATION

Input at 0x00000000



Output at 0x00000000

****

Value starts at 10 and F0 indicating that the decrement is done after copying

For Decrement Before [DB]:

LDMDB R0!, {R2-R5}; COPIES CONTINUOSLY FROM EXTERNAL TO R2 TO R5

  STMDB R1!, {R2-R5}; COPIES CONTINUOSLY FROM R2 TO R5 TO DESTINATION

Input at 0x00000000



Output at 0x000000E8

****

Value starts at 00 and ends at EC indicating that the decrement is done before copying.

**b) Load and Store Instructions in a loop**

**Code:**

  AREA PROGRAM, CODE,READONLY

    ENTRY

MAIN

    LDR R0, =0x00000000; SOURCE MEMORY LOCATION

    LDR R1, =0x00000100; DESTINATION MEMORY LOCATION

    MOV R2, #4; 4 LOCATIONS FOR 4 WORDS

FOR LDR R3, [R0], #4; MOVE THE RESPECTIVE VALUE TO R3

    STR R3, [R1], #4; STORE R3 TO THE RESPECTIVE LOCATION

    SUBS R2,R2,#1; DECREMENT THE COUNTER

    BNE FOR; IF NOT ZERO REPEAT THE LOOP

    END

**Output:**

Before Execution Input at 0x00000000



After Execution Output at 0x00000100



**Result:**

The experiments on shift operations have been performed and verified to be correct.