

Algorithm:-

- 1) Initialize registers to hold
 - a) The starting addresses of source block.
 - b) Starting address of destination block.
 - c) The count , the number of data bytes to be moved .
- 2) Get data from source location into accumulator
- 3) Transfer accumulator content to the destination location .
- 4) Decrement the count register
- 5) Update data pointers .
- 6) Check whether all the bytes are transferred or not ie, if count is not zero repeat step 2 through .

Manual calculation

$2 \rightarrow 00$	$2 \quad 2$
$4 \rightarrow 00$	$4 \quad 4$
$6 \rightarrow 00$	$6 \quad 6$
$8 \rightarrow 00$	$8 \quad 8$

Date : 20/02/2020.....
Exp. No.: 1a.....

Expt. Title: Move a block of data
within the internal RAM

Page No. 1

WRITE A PROGRAM TO MOVE A BLOCK OF DATA
WITHIN THE INTERNAL RAM.

Aim :- Write a program to move a block of
data within the internal RAM

Program Code

Label	Mnemonics	Comments
	ORG OH	
START:	MOV R0, #40H	R0 pointed to internal RAM 40H
	MOV R1, #30H	R1 pointed to internal RAM 30H
	MOV R2, #5	R2 loaded with No. of elements in array
START:	MOV A, @R0	Data Transfer
	MOV @R1, A	
	INC R0	
	INC R1	
	DJM Z R2, START	Decrement R2 If not equal to 0 continue with Data Transfer
HERE:	SJMP HERE	
	FEND	

Result

Before Execution :- Fill 5 location at I:0040H with data by as 5 location at I:0030H are blank

Address I:0030H

I:0x30H : 00 00 00 00 00

I:0x40H : 10 20 30 40 50

After execution :- 5 location at I:0040H are filled up with data. These are copied to filled 5 locations at I:0030H

Address I:0030H

I:0x30H : 10 20 30 40 50

I:0x40H : 10 20 30 40 50

Sri Venkateshwara College of Engineering

Date :
Expt. No. :

Expt. Title :

Page No.

2

Result:-

To move a block of data program within the internal RAM is performed

D. MVR

Algorithm:-

- 1) Initialize registers to hold:
 - a) the count, the number of data bytes to be moved.
 - b) The starting addresses of source block.
 - c) Starting addresses of destination block.
- 2) Get data from source location into accumulator and transfer to the destination location.
- 3) Update data pointers.
- 4) Decrement the count register, repeat step 2-3 if count is not zero.

Manual Calculation :-

30	31	32	33	34
00	00	00	00	00



30	31	32	33	34
30	31	32	33	3 3 4

Date : 20/2/2020
Exp. No.: 16Expt. Title : Transfer n-bytes of data
From location 8035H to 8041H

Page No.

3

Write an assembly language program to transfer $n=5$ bytes of data from location 8035H to location 8041H Within External RAM [Without overlap].

Aim:- Write an assembly language program to transfer $n=5$ bytes of data from location 8035H to locations 8041H within External RAM [without overlap] Write the code at $C=030H$

Program code

Label	Mnemonics	Comments
	ORG 0000H	
	SJMP 30H	
	CRG 30H	
	MOV DPH, #80H	: source / Desti address higher order Byte
	MOV R0, #35H	: // source address low Byte
	MOV R1, #41H	: // destination address low Byte
	MOV R3, #0AH	: // count
BACK	MOV DPL, R0	
	MOVX A, @DPTR	
	MOV DPL, R1	
	MOVX @DPTR, A	
	INC R0	
	INC R1	

Result :-

Before Execution :- 5 location at $x:8035h$ are to be filled up with data

Address : $x:8035h$

$x:0x008035$: 30 31 32 33 34

$x:0x008041$: 00 00 00 00 00

After Execution :- 5 location at $x:8035h$ are filled up with data from $8035h$

Address : $x:8035h$

$y:0x008035$: 30 31 32 33 34

$x:0x008041$: 30 31 32 33 34

Sri Venkateshwara College of Engineering

Date :	Expt. Title :	Page No.
Exp. No. :		4

DJNZ R3, BACK
HERE SJMP HERE
END

Result:-

To move a block of data program with
in the external RAM is performed

Algorithm:-

Initial step:-
Input data will be fed into program
to calculate sum of all numbers.

Date : 27/02/2020
Exp. No. : 1C

Expt. Title : MOVE A BLOCK OF DATA BYTES STORED IN INTERNAL RAM

Page No.

5

WRITE ALP to Move A Block Of DATA BYTES STORED IN INTERNAL RAM STARTING AT LOCATION 10H TO LOCATION AT 13H .

Aim :- Write a assembly level program to move a block of data bytes stored in internal RAM starting at location 10H to location at 13H .

Program Code

Label	Mnemonics	Comments
	CR0 R0, #10H	Set source address
	MOV R0, #10H	// source address
	MOV R1, #13H	// destination address
	MOV R2, #6H	// count
	DEC R2	
	MOV A, R0	
	ADD A, R2	
	MOV R0, A	
	MOV A, R1	
	ADD A, R2	
	MOV R1, A	
	INC R2	
Start:	MOV A, @R0	
	MOV @R1, A	
	DEC R0	

Result :- ~~After Execution~~ Before Execution

Address :- i : 010H
i : 010H 10 20 30 10 20 30
After Execution

address : i : 010H
i : 010H 10 20 30 10 20 30

Sri Venkateshwara College of Engineering

Date :	Expt. Title :	Page No.
Exp. No. :		6

	DEC R, DJ M2 R2, Start	
HERE:	SJ MP HERE	
	END	

Result :-

To move a block of data bytes stored
in internal RAM

Algorithm

- 1) Initialize bank registers to hold the area size (count), the block - 1 address & the block - 2 addresses
- 2) Get data from block - 1 location into accumulator and save in a register temporarily (say I use R2).
- 3) Get data from the block - 2 location into accumulator and store in block - 1 location
- 4) Get the data from temporary register (R2) and store in block - 1 locations.
- 5) Decrement the count register and repeat from step 2 to 4 till count is zero

Manual Calculation

10	11	12	13	14
20	21	22	23	24
↓				
20	21	22	23	24
10	11	12	13	14

Date : 20/2/2020

Expt. Title :

Page No.

7

Exp. No. : 1d

Exchange Blocks of Data

WRITE AN ALP TO EXCHANGE 5 BYTES OF DATA STORED IN LOCATIONS STARTING FROM 0027h WITH 5 BYTES IN LOCATIONS STARTING FROM 0041h OF EXTERNAL MEMORY. USE BANK REGISTERS TO HOLD MEMORY ADDRESSES

Aim :- a) Write the program without using XCH instruction

Program code

Label	Mnemonics	Comments
	ORG 00H	
	MOV R0, #27H	// source address
	MOV R1, #41H	// destination address
	MOV R3, #05H	// count
BACK:	MOV X A, @R0	
	MOV R2, A	
	MOV X A, @R1	
	MOV X @R0, A	
	MOV A, R2	
	MOV X @R1, A	
	INC R0	
	INC R1	
	DJNZ R3, BACK	
HERE	SJMP HERE	
	END	

Result

Before Execution

Address :- x:0027H

x:0x000027 : 10 11 12 13 14
x:0x000034 : 20 21 22 23 24

After Execution

Address :- x:0027H

x:0x000027 : 20 21 22 23 24
x:0x000034 : 10 11 12 13 14

Manual calculation

01 02 03 04 05

10 20 30 40 50

↓

10 20 30 40 50

01 02 03 04 05

Date :

Expt. Title :

Page No.

8

Exp. No. :

Aim :- b) write the program using XCH instruction

address = 0000000000000000

L1: STX, SJMP L1

Program Code

Label	Mnemonics	Comments
	ORCR 0000H	Initialising R0
	SJMP 30H	Initialising R1
	ORCR 30H	
	MOV R0, #27H	source address
	MOV R1, #41H	destination address
	MOV R3, #05H	count
BACK	MOV X A, @R0	
	MOV R2, A	
	MOV X A, @R1	
	XCH A, R2	
	MOV X @R1, A	
	XCH A, R2	
	MOV X @R0, A	
	INC R0	
	INC R1	
	DJNZ R3, BACK	
HERE	SJMP HERE	
	END	

Result:-

Before Execution

Address :- x: 027H

x: 0x000027 : 10 20 30 40 50

x: 0x 0000 41 : 11 22 33 44 55

After Execution :- The data at x: 0027H & x: 00

Address :- x: 027H

x: 0x000027 : 11 22 33 44 55

x: 0x 0000 41 : 10 20 30 40 50

00 00 00 00 00

Exch

Sri Venkateshwara College of Engineering

Date :

Expt. Title :

Exp. No. :

Page No.

9

Result :-

To exchange 5 bytes data stored in location starting from 0027H with 5 bytes in location starting from 0041H of external memory.