|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Microprocessors and Peripherals (2UXC404)** | **Semester:** | **IV** |
| **Date of Performance:** | 24-2-2021 | **Batch No:** | B2 |
| **Faculty Name:** | KCS | **Roll No:** | 1912052 |
| **Faculty Sign & Date:** |  | **Grade/Marks:** | \_\_\_/25 |

**Experiment No: 2**

**Title:** Block Transfer

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| **Aim:** Write an 8085 based ALP to   1. Move a block of 6 data bytes from ML F100H onwards to F200H. 2. Exchange a block of 16 data bytes in ML F500H onwards with a block in ML F600H onwards   **Objectives:**  To study basic instructions and addressing modes of 8085.  This experiment covers following instructions groups.   1. Data transfer 2. Logical (Rotate and compare instructions) 3. Branch |

|  |
| --- |
| **COs to be achieved:** |
| **CO 1.** Describe basic operation of 8085 microprocessor system and explain its timing diagrams. |

|  |
| --- |
| **Useful links** |
| Virtual Lab:  [http://vlabs.iitb.ac.in/vlabs-dev/labs\_local/microprocessor/labs/explist.php](%20http://vlabs.iitb.ac.in/vlabs-dev/labs_local/microprocessor/labs/explist.php)  Simulator:  <https://www.sim8085.com/> |

|  |
| --- |
| **Work to be done** |
| 1. Upload codes and screenshots of virtual lab implementation. 2. Upload scanned image of handwritten algorithm/flowchart and code and results for post lab questions. |

|  |
| --- |
| **Post Lab Subjective/Objective type Questions:** |
| Q1. Write an 8085 based ALP to reverse a block of data at ML F500H and store the reverse block at ML F600H onwards    Q2. Write on 8085 based ALP to count the number of zeros, positive numbers and negative numbers in an array. The length of the array is at memory C100H and tha array starts at memory location C101H. Store the no. of zeros at Location C300H,no. of positive nos at memory C301H and no. of negative numbers at memory C302H.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Address** | | **Opcode** | **Label** | | **Mnemonics** | | **Comments** | | | 3000 | | C3 3 0 |  | | jmp start | |  | | | 3002 | | 21 0 193 | start | | LXI H,C100H | | Contents are assigned to H | | | 3006 | | 4E |  | | MOV C,M | | Content is moved from memory to C | | | 3007 | | 6 0 |  | | MVI B,00H | | B is initialized at 0 | | | 3009 | | 1E 0 |  | | MVI E,00H | | E is initialized at 0 | | | 3011 | | 16 0 |  | | MVI D,00H | | D is initialized at 0 | | | 3013 | | 23 | up | | INX H | | Increment in H location | | | 301A | | 7E |  | | MOV A,M | | Content is moved from memory to accumulator | | | 3015 | | FE 0 |  | | CPI 00H | | Compare 0 with A | | | 300B | | C2 28 0 |  | | JNZ down | | If not zero then jump to down | | | 3018 | | 4 |  | | INR B | | Increment in B | | | 3019 | | 78 |  | | MOVA,B | | Content is moved from B to accumulator | | | 3020 | | 32 0 195 |  | | STA C300H | | Content in A is stored at C300H | | | 3023 | | C3 46 0 |  | | JMP j | | Jump to j | | | 3026 | | E6 128 | down | | ANI 80H | | AND with A | | | 3029 | | CA 41 0 |  | | JZ k | | If zero jump to k | | | 3030 | | 1C |  | | INR E | | Increment in E | | | 300C | | 7B |  | | MOV A,E | | Content is moved from E to accumulator | | | 3031 | | 32 2 195 |  | | STA C302H | | Content in A is stored at C302H | | | 3034 | | C3 46 0 |  | | JMP j | | Jump to j | | | 3037 | | 14 | k | | INR D | | Increment in D | | | 300D | | 7A |  | | MOV A,D | | Content is moved from D to accumulator | | | 3038 | | 32 1 195 |  | | STA C301H | | Content in A is stored at C301H | | | 3041 | | D | j | | DCR C | | Decrement in C | | | 300E | | C2 13 0 |  | | JNZ up | | Jump to up if C not zero | | | 3044 | | 76 |  | | HLT | | Stop | | | **Memory Location** | | **Contents** | | | **Memory Location** | | **Contents** | | | Before execution | | | | | After execution | | | | | C100 | | 04 | | | C300 | | 01 | | | C101 | | 12 | | | C301 | | 01 | | | C102 | | 00 | | | C302 | | 02 | | | C103 | | 8A | | |  | |  | | | C104 | | DD | | |  | |  | |     Before execution:    Q3. Explain various branch instructions of 8085.  There are main 3 types of branch instructions:  JUMP: The jump instruction transfers the program sequence to the memory address given in the operand based on the specified flag  CALL: The call instruction transfers the program sequence to the memory address given in the operand. Before transferring, the address of the next instruction after CALL is pushed onto the stack.  Return:  The return instruction transfers the program sequence from the subroutine to the calling program. |

**Conclusion:**

Wrote 8085 programs to move a block of 6 data bytes from ML F100H onwards to F200H. and exchange a block of 16 data bytes in ML F500H onwards with a block in ML F600H onwards

**Signature of faculty in-charge with Date:**