|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FAKULTI TEKNOLOGI KEJURUTERAAN**  **ELEKTRIK DAN ELEKTRONIK**  **UNIVERSITI TEKNIKAL MALAYSIA MELAKA** | | | | | |
| **COMPUTER ORGANIZATION AND ARCHITECHTURE** | | | | | |
| BEEC 2373 | | | SEMESTER 2 | SESI 2019/2020 | |
| LAB 3: DATA TRANSFER AND ARITHMETIC | | | | | |
| **NO.** | **STUDENTS' NAME** | | | | **MATRIC. NO.** |
| **1.** | **AHMAD IRFAN BIN HARMAN** | | | | **B081910068** |
| **2.** | **CHE IZZATI AYUNI BINTI CHE ROHIM** | | | | **B081910021** |
| **3.** |  | | | |  |
| **PROGRAMME** | | **1BEEC** | | | |
| **SECTION / GROUP** | | **1/1** | | | |
| **DATE** | | **10 MARCH 2020** | | | |
| **NAME OF INSTRUCTOR(S)** | | 1. **EN. NOOR MOHD ARIFF BIN BRAHIN** | | | |
|  | | | |
| **EXAMINER’S COMMENT(S)** | | | | **TOTAL MARKS** | |

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. No. | Date | Author(s) | Description |
| 1.0 | 12 FEB 2020 | 1. Noor Mohd Ariff 2. Ahmad Nizamudin | 1. Update to new UTeM logo 2. Update faculty's name 3. Change "course" to "programme" 4. Remove verification stamp |
|  |  |  |  |
|  |  |  |  |

## OBJECTIVES

* To construct and test programs using the following categories of 8086 Instruction Set:

• Data Movement

• Arithmetic Operations

* To understand 8086 microprocessor architecture.

## EQUIPMENT/COMPONENTS

1. Personal computer installed with 8086 Software.

## SYNOPSIS & THEORY

The table below summarizes the different categories of 8086 Instruction Set. In this experiment you will be creating and testing codes written in assembly language programs using 8086 Instruction Set. Observe how the 8086 internal registers are affected by the different instructions.

**TABLE 1: Categories of 8086 Instruction**

**Set**

**Type Description Operation Name**

**Data Transfer** Transfer data from one location to another

Move, Store, Load, Exchange, Clear, Set, Push, Pop

**Arithmetic** Perform arithmetic function in ALU

Add, Subtract, Multiply, Divide, Absolute, Negate, Increment, Decrement

**Logical** Perform logic function in ALU AND, OR, NOT, XOR, Test, Compare, Shift, Rotate

**Transfer of Control** Update program counter Jump, Jump Conditional, Jump to Subroutine, Return, Skip, Skip Conditional, Halt, Wait, No operation

**Input/Output** Issue command to I/O

module

Input, Output, Start I/O, Test I/O

**Conversion** May involve special logic Translate, Convert

to perform conversion

1. **PROCEDURE**

## Real Mode Programming

1. Run the emulator8086.

2. Assemble the given codes with the mathematical instructions below:

ORG 100H

MOV AX, 27

ADD AX, 15

MOV BX, 625

SUB BX, 250

MUL BX

ADD AX, 191

RET

## Construct Arithmetic Programming

1. Given an arithmetic operation:

-710 \* (6410 - 4910) + 5310

1. Develop your own codes in 8086 assembly language.
2. Make sure your registers are well defined.
3. Explain the output with the registers involved.
4. Run these codes and find out the registers output.

mov ah, 7Fh

mov ax,1234d

mov bh, al

mov bl, ah

mov al, 81h

add al, 0FEh



mov al, 0FEh

sub al, 2

mov bl, 8Ch

mov bh, 2Dh

mov bx, ax

## RESULT

**5.1 Arithmetic**

1. Observe the contents of the specified registers below after each instruction has been executed and record your result in Table 1.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instruction | Register Content | | | | | Flag Register |
| AX | BX | CX | DX | IP |
| 1. ORG 100H | 0000 | 0000 | 0013 | 0000 | 0100 | NV UP EI PL NZ NA PO NC |
| 2. MOV AX, 27 | 001B | 0000 | 0013 | 0000 | 0103 | NV UP EI PL NZ NA PO NC |
| 3. ADD AX, 15 | 002A | 0000 | 0013 | 0000 | 0106 | NV UP EI PL NZ NA PO NC |
| 4. MOV BX, 625 | 002A | 0271 | 0013 | 0000 | 0109 | NV UP EI PL NZ NA PO NC |
| 5. SUB BX, 250 | 002A | 0177 | 0013 | 0000 | 010D | NV UP EI PL NZ AC PE NC |
| 6. MUL BX | 3D86 | 0177 | 0013 | 0000 | 010F | NV UP EI PL NZ NA PE NC |
| 7. ADD AX, 191 | 3E45 | 0177 | 0013 | 0000 | 0112 | NV UP EI NG NZ NA PO NC |
| 8. RET | 3E45 | 0177 | 0013 | 0000 | 0000 | NV UP EI NG NZ NA PO NC |

**Table 1**: Registers value

1. Write the final answer and the register location of the answer.

* **15941(dec)/3E45(hex)**
* register location:AX

1. Write down the mathematical equation from the codes above.

((27 + 15) \* (625 - 250)) + 191

**5.2 Arithmetic**

1. Write your own instruction in below table and record your result.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instruction | Register Content | | | | | Flag Register |
| AX | BX | CX | DX | IP |
| ORG 100H | 0000 | 0000 | 000F | 0000 | 0100 | NV UP EI PL NZ NA PO NC |
| MOV AX, -7 | FFF9 | 0000 | 000F | 0000 | 0103 | NV UP EI PL NZ NA PO NC |
| MOV BX, 64 | FFF9 | 0040 | 000F | 0000 | 0106 | NV UP EI PL NZ NA PO NC |
| SUB BX, 49 | FFF9 | 000F | 000F | 0000 | 0109 | NV UP EI PL NZ NA PO NC |
| MUL BX | FF97 | 000F | 000F | 000E | 010B | NV UP EI PL NZ AC PE NC |
| ADD AX,53 | FFCC | 000F | 000F | 000E | 010E | NV UP EI PL NZ NA PE NC |
| RET | FFCC | 000F | 000F | 000E | 0000 | NV UP EI NG NZ NA PO NC |

**Table 2**: Registers value

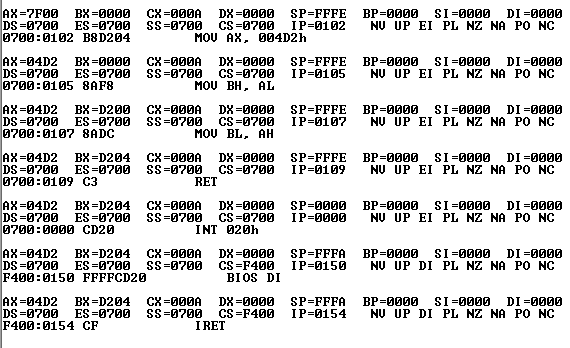
(iii). **Explanation**

* First, move -7(dec) to AX register.

* Then move 64(dec) to BX register.
* After that, the value in BX register substract with 49 which is 64 - 49 = 15(dec) and store in BX register.
* The value in BX register multiply with value in AX register which is 15 \* -7 = -105(dec) and store in AX register.
* Lastly, the value in AX register add with 53(dec) which is -105 + 53 = -52(dec)/FFCC(hex) and store in AX register.

1. Write down the register output after you execute the code in procedure 2 i, ii, and iii.

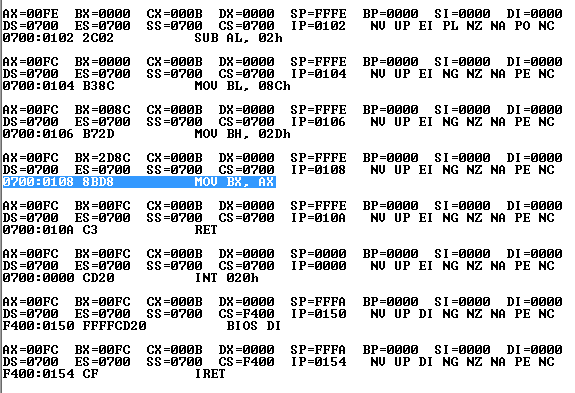
**i)**



**ii)**



**iii)**



## DISCUSSION

* Data transfer instructions move data between memory and the general-purpose and segment registers, and perform operations such as conditional moves, stack access, and data conversion.
* Arithmetic instructions are the instructions which perform basic arithmetic operations such as addition, subtraction, multiplication and few more.
* Arithmetic operations, ADD, SUB, MUL and DIV were used to add, subtract, multiply and divide.

## CONCLUSION

In conclusion, we are able to construct and test programs using the data movement and arithmetic operations of 8086 Instruction Set by the end of this lab. Besides, we also able to understand 8086 microprocessor architecture.