|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FAKULTI TEKNOLOGI KEJURUTERAAN**  **ELEKTRIK DAN ELEKTRONIK**  **UNIVERSITI TEKNIKAL MALAYSIA MELAKA** | | | | | |
| **COMPUTER ORGANIZATION AND ARCHITECHTURE** | | | | | |
| BEEC 2373 | | | SEMESTER 2 | SESI 2019/2020 | |
| LAB 2: DATA TRANSFER AND MEMORY ACCESS | | | | | |
| **NO.** | **STUDENTS' NAME** | | | | **MATRIC. NO.** |
| **1.** |  | | | |  |
| **2.** |  | | | |  |
| **3.** |  | | | |  |
| **PROGRAMME** | |  | | | |
| **SECTION / GROUP** | |  | | | |
| **DATE** | |  | | | |
| **NAME OF INSTRUCTOR(S)** | |  | | | |
|  | | | |
| **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 opertion

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

module

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

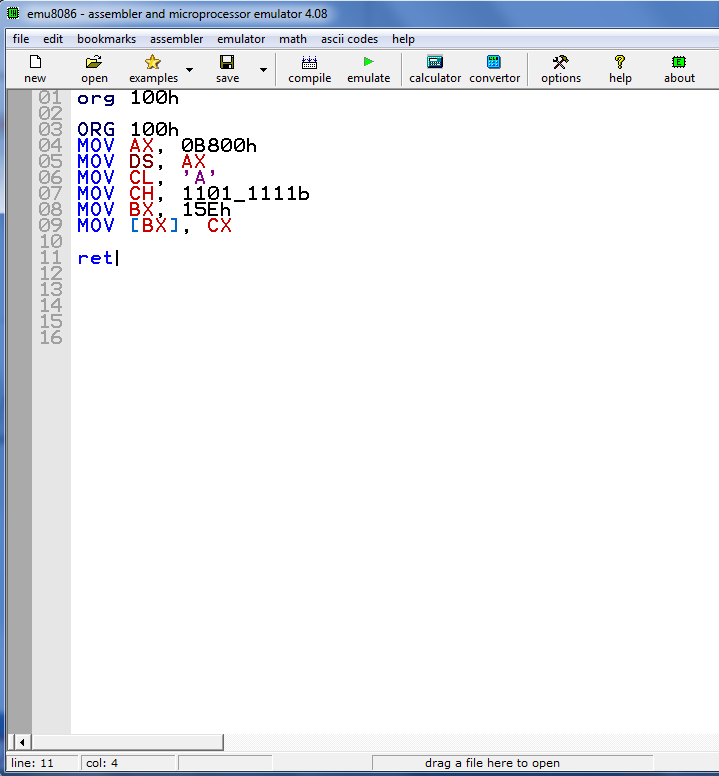
**Conversion** May involve special logic Translate, Convert

to perform conversion

1. **PROCEDURE**

## Data Transfer

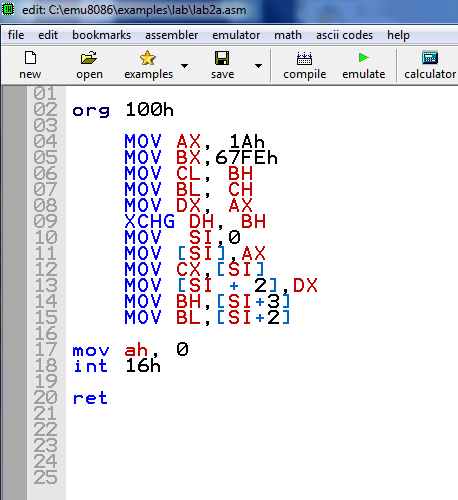
1. Run the emulator8086.
2. Consider compiling below program to execute on various type of data transfer.



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

**4.2 Memory Access**

1. Assemble the given codes with the instructions below:



1. Observe the contents of the specified registers below after each instruction has been executed and record your result.
2. For number 8 and 10, what are the contents of the memory locations DS:0000 to DS:0003 after each instruction has been executed. Use debug,exe emulation.

DS:0000 =

DS:0002 = \_\_\_\_\_\_

## EXPERIMENT RESULTS

**5.1 Memory Access**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instruction | Register Content | | | | | Flag Register |
| AX | BX | CX | DX | IP |
| org 100h |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| mov AX, 0b800h |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| mov DS, AX |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| mov CL, 'A' |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| mov CH, 1101\_1111b |  |  |  |  |  | NV UP DI PL NZ AC PE NC |
| mov BX, 15Eh |  |  |  |  |  | NV UP DI PL NZ NA PE NC |
| mov [BX], CX |  |  |  |  |  | NV UP DI NG NZ NA PO NC |
| ret |  |  |  |  |  | NV UP DI NG NZ NA PO NC |

**Table 1** : Registers value

**5.2 Memory Access**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instruction | Register Content | | | | | Flag Register |
| AX | BX | CX | DX | IP |
| Prior to execution |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| 1. MOV AX, 1Ah |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| 1. MOV BX,67FEh |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| 1. MOV CL, BH |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| 1. MOV BL, CH |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| 1. MOV DX, AX |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| 1. XCHG DH, BH |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| 1. MOV SI,0 |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| 1. MOV [SI],AX |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| 1. MOV CX,[SI] |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| 1. MOV [SI + 2],DX |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| 1. MOV BH,[SI+3] |  |  |  |  |  | NV UP DI PL NZ NA PO NC |
| 1. MOV BL,[SI+2] |  |  |  |  |  | NV UP DI PL NZ NA PO NC |

**Table 2** : Registers value

## DISCUSSION

## CONCLUSION