Table of Contents

1. **Objective ………………………………………………………………………3**
2. **Theory …………………………………………………………………………3**
3. **Program Flow Chart …………………………………………………………4**
4. **Single Stepping ……………………………………………………………… 5**
5. **Mnemonics ……………………………………………………………………5**
6. **Questions/Answers ………………………………………………………. 6 - 7**
7. **Conclusion …………………………………………………………………….7**

Objective

* Get a handle for manipulating data using the SDK-85 (Student Development Kit)
* Use Appendix F (8085 Instruction Set), specifically the Data Transfer Group, to transfer data between the registers and memory locations of the SDK-85.

Theory

* The SDK-85 (Student Development Kit) is a single board microcomputer system kit using the 8085 processor. It is made by Intel and is now used to teach students about the concepts of microprocessors. Contains the following
  + **Microprocessor**
  + **Memory Element** – This describes both ROM (Read Only Memory) and RAM (Random Access Memory)
    - ROM (Read Only Memory) – Contains system boot up instructions
    - RAM (Random Access Memory) – Has Read/Write capabilities
  + **I/O Unit** – Handles input from user and provides output
* Microprocessors are computer processors that incorporate the functions of a central processing unit on a single integrated circuit (IC) or at most a few integrated circuits. They contain the following:
  + **Combinational logic Unit** ­– are logic circuits implemented by Boolean (logic gates) circuits, where the output is a pure function of the present input only. Think Half-Adders, Full-Adders, Encoders, and Decoders.
  + **Sequential logic Unit** – this is a type of logic circuit whose output depends on previous inputs as well as on the present inputs.
    - Contains Memory
    - Contains a clock
* This lab focuses on **Appendix F (the 8085 Instruction Set)**, specifically the **Data Transfer Group**. This is the set of assembly instructions that perform the moving, storing, loading, and exchanging of data.

Program Flow Chart

[2051] ← [A]

[A] ← [CC]

Single Stepping

|  |  |  |  |
| --- | --- | --- | --- |
| **Move** | **Data** | **A** | **2051** |
| 2000 | 3E | ? | ? |
| 2002 | 32 | CC | ? |
| 2005 | CF | CC | CC |

Mnemonics

|  |
| --- |
| **MVI A, CC** |
| **STA 2051** |
| **RST 1** |

Questions/Answers

**Part A:**

* We verified that the data FF was placed into the memory location 2051
* We also verified that an error was obtained when attempting to put FF in ROM location 0018
* We then verified that the data FF was placed into the register B

**Part B:**

* The number of bytes used for the program shown in the *Mnemonics Section* took 6 bytes
  + 2 for MV1 A, CC
  + 3 for STA 2051
  + 1 for RST 1
* Additionally, the about of seconds it takes is given by the following:
  + (7 + 13 + 12) = (32 cycles) × (330 × 10-9 seconds/cycle) = **1.056 × 10-6 seconds**

**Part C:**

* What is the meaning of the following displays?
  + **-8085** – Indicates the reset of the SDK 85
  + **E** ­– Indicates the program is in an infinite loop
  + **Err ­**– Indicates there is an error

**Part D/Part E:**

|  |  |  |
| --- | --- | --- |
| A | 0A | 0A |
| B | FF | 01 |
| C | 00 | 00 |
| D | 00 | 00 |
| E | 99 | 00 |
| F | A0 | A0 |
| H | 20 | 20 |
| L | FE | FE |
| I | 07 | 07 |
| SPH | 20 | 20 |
| SPL | E9 | E3 |
| PCH | 00 | 02 |
| PCL | 75 | 35 |
| 0150 | F6 | F6 |
| 2050 | 21 | 21 |

* After pressing reset the program the values of the register are shown above in the second column. When reset is pressed and we check those registers again the values are different except for the memory locations, 0150 and 2050.

**Part F:**

* The program shown in the mnemonics section above does the following:
  + Places the data CC into register A
  + Stores the value of register A into memory location 2051
  + We end the program by using the RST 1 which stops/freezes the program
* The two displays for the SDK-85 are the address and data fields.

Conclusion

* We were successful in completing this lab. We had brief issues with running the program. We had the order wrong. We first were supposed to press GO, then go to the register which is the first line of the program, then press EXEC. Once we had that sorted we were able to successfully get the results: having the data CC in register A and memory location 2051.