Malak Hamed Mokhtar **Yahia Ahmed Hatem Mostafa**
Omar Yasser Youssef **Youssef Mohamed Fayed**

Microprocessors Systems Project

Project Description:

This project contains two different scenarios. A main menu screen is provided to help navigate through the different scenarios.

1. Part A

The first scenario is a **"Two-player Processor Simulation game"** with a small scale gun game to increase your points. Before the game, each player enters the suggested initial points to begin with in the game, the lower number will be used for both of them as initial points. Players take turns writing assembly commands, the aim of the game is to pass the hex value [105E] in any of the opponent's registers. Commands can be executed on registers or on the memory. Additionally, the game will end if one of the players lost all his points and the other player is automatically considered the winner. Moreover, before the game starts, each user is asked to enter a forbidden alphanumeric character that his/her opponent cannot use for any command.

i. The **commands** provided are: [MOV-ADD-ADC-PUSH-POP-INC-DEC-DIV-MUL].

ii. The **addressing modes** that could be used are:

a. Direct addressing mode
b. Immediate addressing mode
c. Register addressing mode
d. Register indirect addressing mode

iii. There are different types of **error detections** covered when the user writes his/her command. If the user made an error, the error type will be shown on the screen, the user will lose one of his points, and the user should write another command instead. The error detections covered are:

a. Size mismatch
b. Memory to memory operation
c. Invalid register name
d. Invalid command name

e. Pushing 8-bits

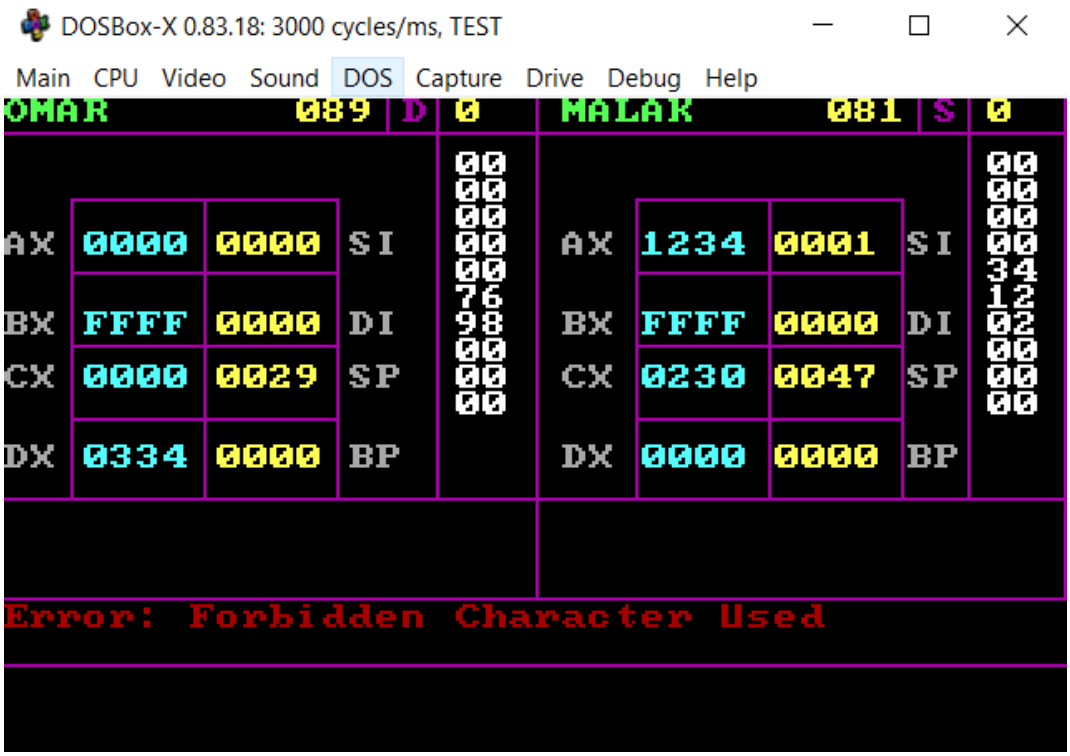
f. Popping 8-bits

g. Incorrect addressing mode

h. Operand 1 cannot be immediate value

i. Forbidden character used

Sample screen showing an error detection: (Note: forbidden character displayed in purple at top)

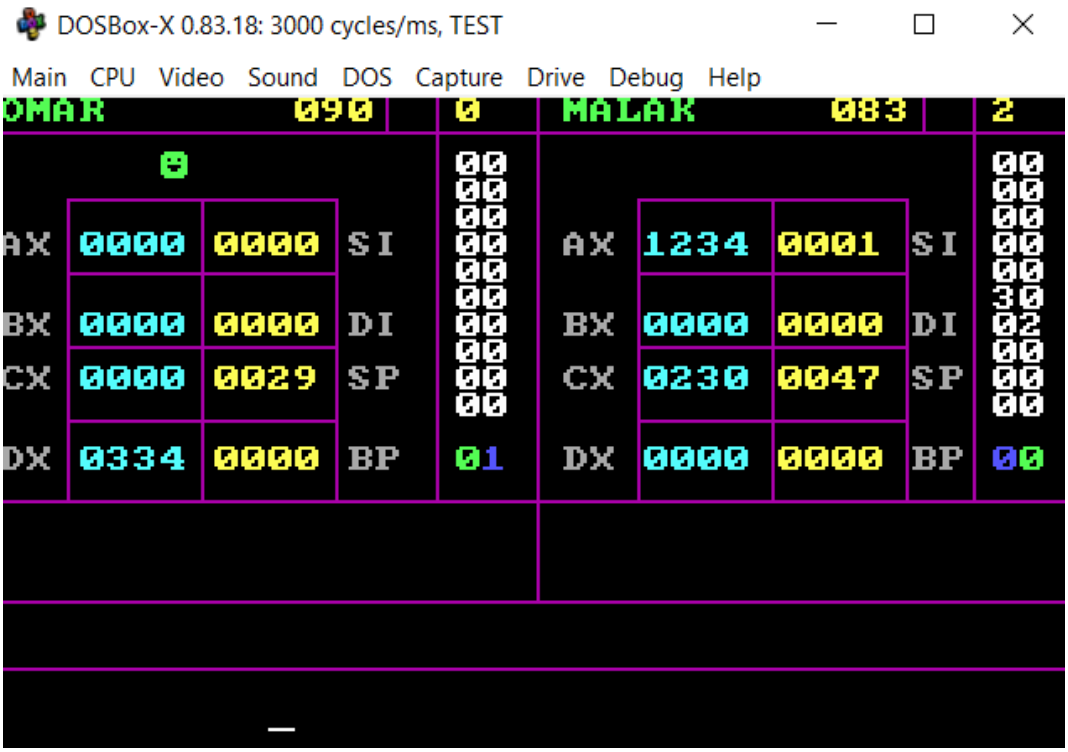


iv. There are **<u>two different levels</u>** for the game, each level comes with its own power-ups and features. Power-ups can be chosen at the beginning of each round and they consume some points from the player depending on their type:

a. Level 1:

The small scale gun game is displayed. Each player has 10 seconds to try and shoot the object (space bar to shoot and keyboard arrows to move within the borders), depending on the object color the player is awarded either 1 or 2 points if he managed to shoot it. The game is displayed every 5 rounds.

Sample screen of the gun game: (Note the green “0” and blue “1” indicate the number of hit objects of each color)



Before the user writes his command he should enter a number from 0-3 to choose his **power-up preference**:

0: No power-up

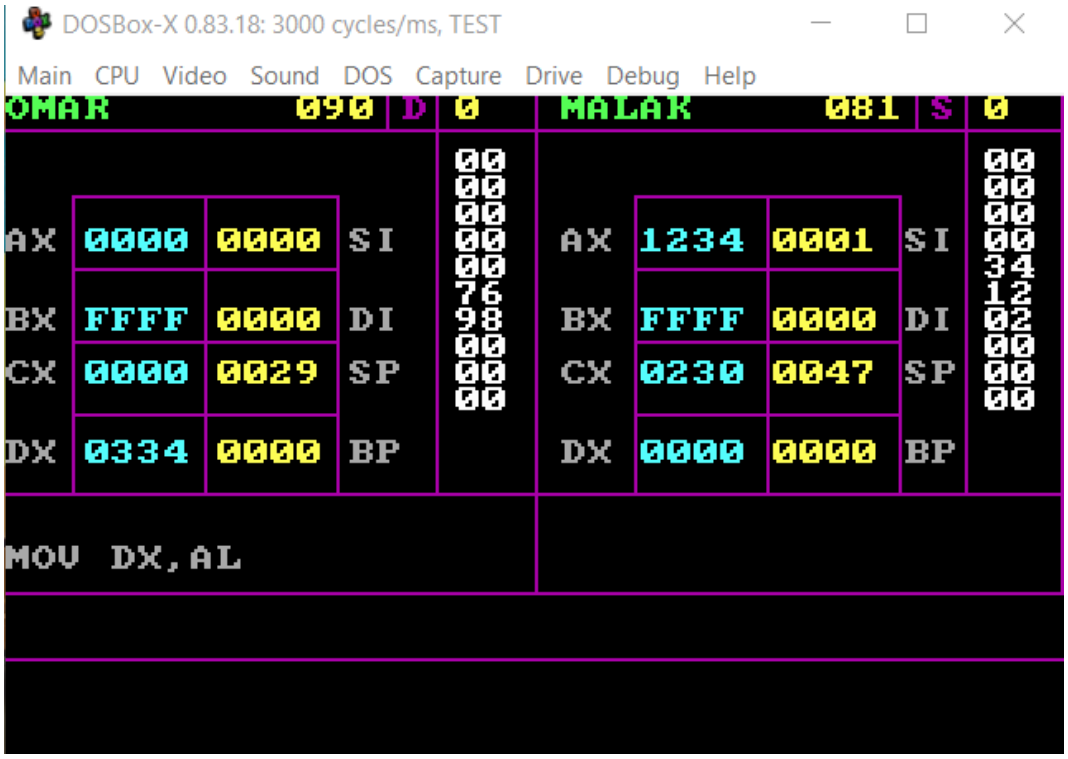
1: Changing his own forbidden character for only one command (consumes 8 points)

2: Executing the command on his/her own processor instead of their opponent’s (consumes 5 points)

3: Executing the command on both processors at the same time (consumes 3 points)

4: Clearing all registers at once and then expect and execute a command from the user (consumes 30 points)

Screen sample after a few turns: (Notice the loaded values in the memory (in white))



b. Level 2:

Level 2 is very similar to Level 1 but with a few differences. First, the user enters power-up preference (same as above), then user enters:

0: Execute command on his opponent's registers

1: Execute command on his own register

After that, the user should write his command. Additionally, the forbidden character is hidden, any command including the forbidden character will display an error message, not be executed and the user will lose a point. Each player should deduce the forbidden character through non executed commands.

At the end of the game, if the user wishes to return to the main-menu screen he should press the "Enter" key; else the program will end.

2. Part B

The second scenario is **“The Chat Mode”**. A screen is displayed and is divided in half; each user has one half of the screen to write on. When the user is done with his message he presses the enter key so that the cursor can move to the other half and expect a message from the other user and so on. The scrolling functionality is available (watch the video to see it) and to return to the main-menu the user can press “F3”.