

HO CHI MINH UNIVERSITY OF TECHNOLOGY
OFFICE FOR INTERNATIONAL STUDY PROGRAM



COMPUTER ARCHITECTURE
ASSIGNMENT REPORT

Instructor: Prf. Pham Quoc Cuong

Class: CC05

Student name: Lương Thị Minh Oanh

Student ID: 1950031

Table of Contents

1.INTRODUCTION	3
1.1.What is MIPS assembly language?	3
1.2.What is MARS?	3
2.PROBLEM: TIC TAC TOE	4
2.1.What is Tic Tac Toe?	4
2.2.Game Rules	4
2.3.Win cases	4
2.4.Mistake cases	5
3.IDEA	5
3.1.Introduction	5
3.2.Basic flowchart	6
4.SOLUTION	6
4.1.Setting	6
4.2.Introduction of Tic-Tac-Toe game	7
4.3.Input a number to choose favorite sign (X or O)	7
4.4.Playing game	8
4.5.Input a number to choose favorite space	8
4.6.Checking win condition	9
4.7.Printing	9
5.CONCLUSION	10
6.REFERENCE	10

1. INTRODUCTION:

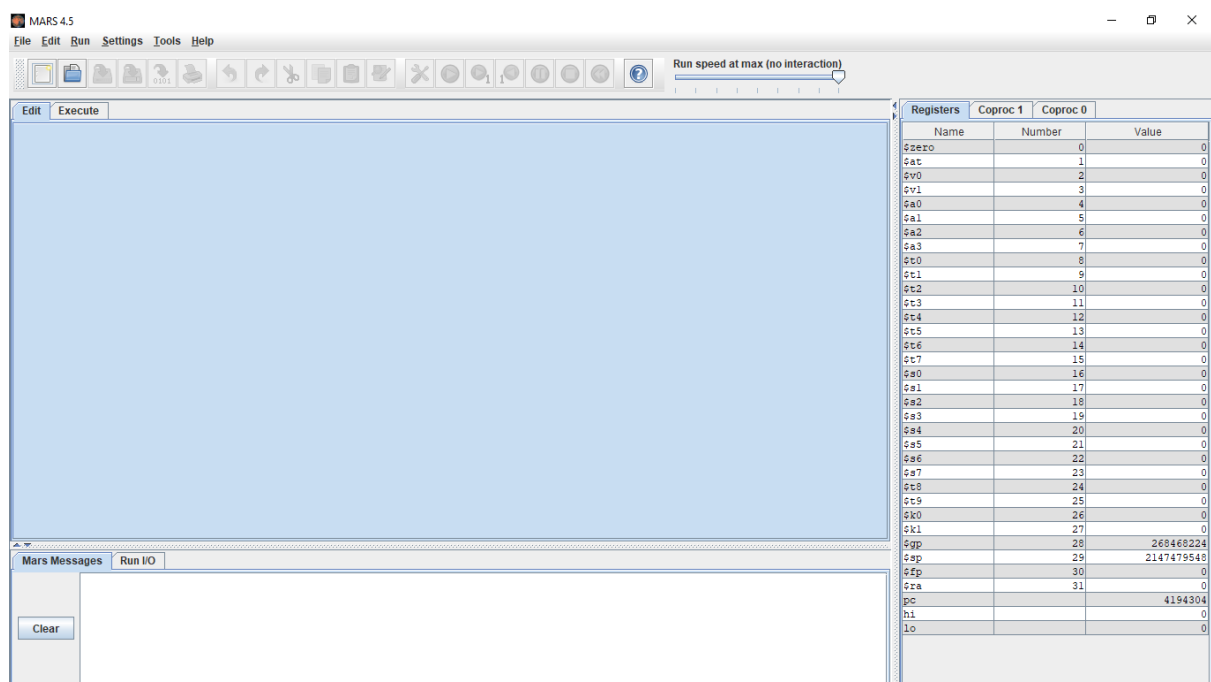
1.1. What is MIPS assembly language?

MIPS assembly language simply refers to the assembly language of the MIPS processor. The term MIPS is an acronym for Microprocessor without Interlocked Pipeline Stages. It is a reduced-instruction set architecture developed by an organization called MIPS Technologies.

The MIPS assembly language is a very useful language to learn because many embedded systems run on the MIPS processor. Knowing how to code in this language brings a deeper understanding of how these systems operate on a lower level.

1.2. What is MARS?

MARS is a lightweight interactive development environment (IDE) for programming in MIPS assembly language, intended for educational-level use with Patterson and Hennessy's *Computer Organization and Design*.



2.PROBLEM: TIC TAC TOE

2.1.What is Tic Tac Toe?

Tic-tac-toe, noughts and crosses, or Xs and Os is a paper-and-pencil game for two players who take turns marking the spaces in a three-by-three grid with X or O. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner. It is a **solved game**, with a forced draw assuming best play from both players

2.2.Game Rule:

The game is played on a grid that's 3 squares by 3 squares.

You are **X**, your friend (or the computer in this case) is **O**. Players take turns putting their marks in empty squares.

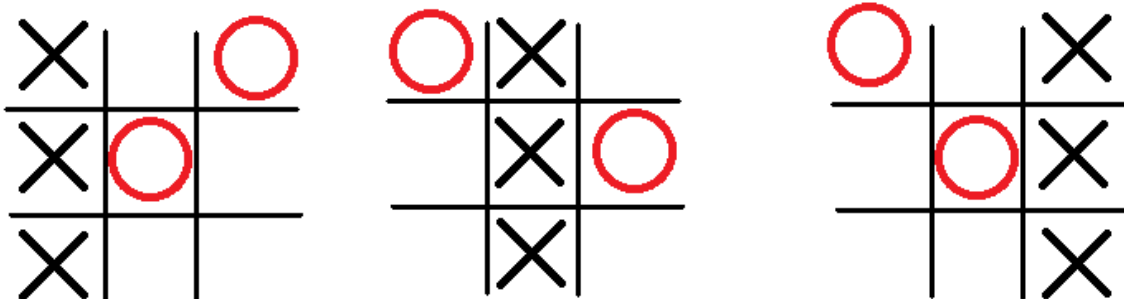
The first player to get 3 of her marks in a row (up, down, across, or diagonally) is the winner.

When all 9 squares are full, the game is over. If no player has 3 marks in a row, the game ends in a tie.

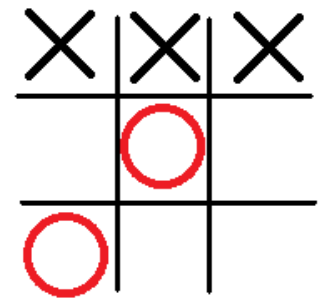
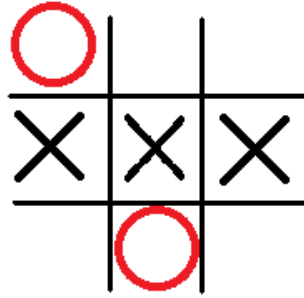
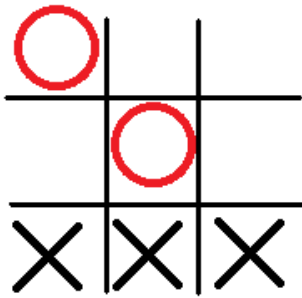
2.3.Win cases:

Assuming the player who choose “X” is win when:

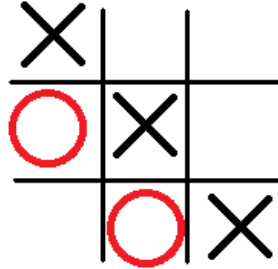
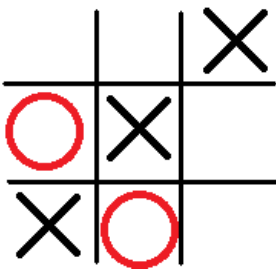
- Finishing a vertical row:



- Finishing a horizontal row:



- Finishing a diagonal row:



2.4. Mistake cases:

- A player draws outside the three-by- three grid.(OutOfRangeException error)
- A player marks on a full space.(Duplication error)

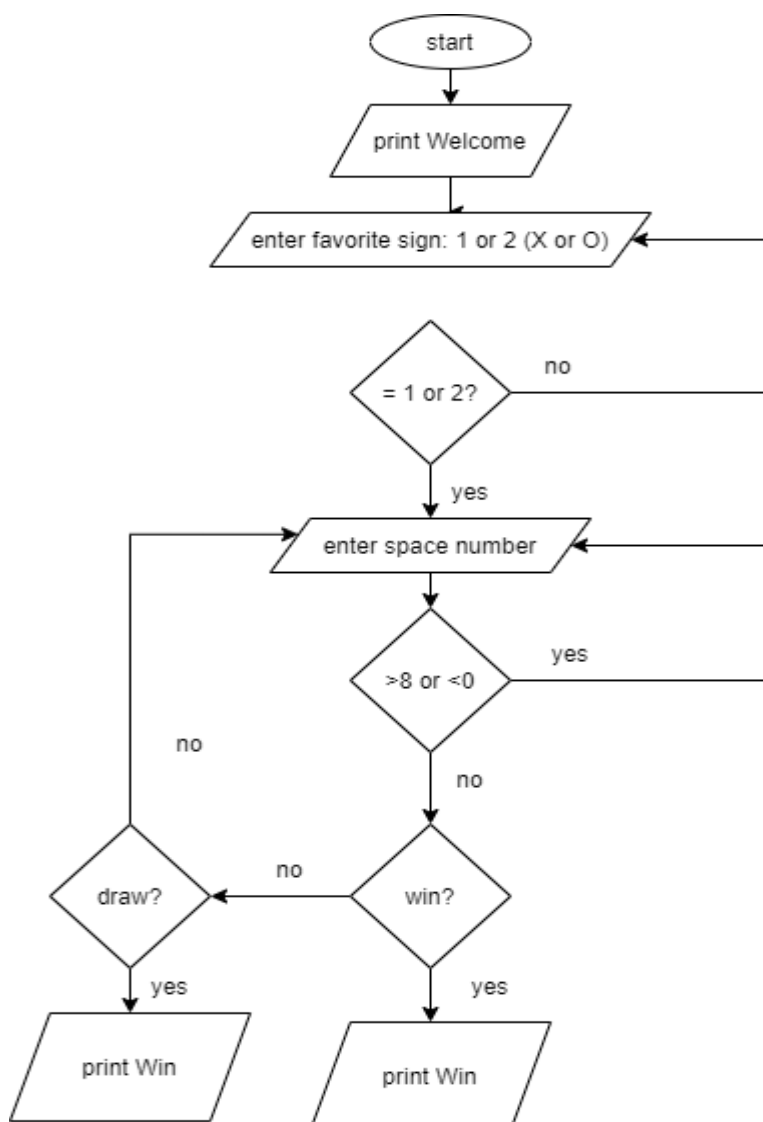
3. IDEA:

3.1. Introduction:

At first, to draw a sign at a wanted space position, the player will give an integer number (from 0 to 9). Every number space stores "0", which includes that it is not full yet. Secondly, if the player draws "X" the number space store "1", and if drawing "O" the number space store "4". Next, the program will check the total of 2 stored numbers at the same row (vertical, horizontal and diagonal). If this is equal to 2 (1+1), player "X" is the winner. And if it is equal to 8 (4+4), player "O" will win this game. After the last turn (turn 9) at the game, in case there is no one of them in enough condition to win, it becomes a drawn play.

0	1	2
3	4	5
6	7	8

3.2. Basic flowchart:



4. SOLUTION:

4.1. Setting:

string: an array to print after every single draw turn.

```
string: .word '0','1','2','3','4','5','6','7','8'
```

Some register use:

Register	Explaining	Storage
\$s0-\$s7, \$t8	Storing who draw at space number from 0 to 8	0,1,4
\$t0	Current play turn	1 (X turn), 2 (O turn)
\$t1	Newest space number (input number)	integer
\$t7	The total turn from starting	from 0 to 9

4.2.Introduction of Tic-Tac-Toe game:

Firstly, the program will give a paragraph to welcome players.

```
Tic-Tac-Toe game
Welcome you and your friend to this game
```

4.3.Input a number to choose favorite sign (X or O):

```
Please choose the sign:
1.X
2.O
```

Next, a player is required to input an integer number (1 or 2) to choose her/his sign (X or O). In case he/she gives a different integer number, the program will require this player to give another integer number until he/she inputs the allowed number (1 or 2). (Figure P.4.3.1)

If the first competitor choose 1 (means her/his sign is “X”), the sign of the second one is “O” and vice versa.(Figure P.4.3.2)

```
Error: OUTOFRANAGE!  
Please input another number.
```

(Figure P.4.3.1)

```
You are X and your friend is O
```

(Figure P.4.3.2)

4.4. Playing game:

```
Let's play now!  
X first!  
|0|1|2|  
|3|4|5|  
|6|7|8|
```

(Figure P.4.4.1)

At beginning of the game, Mars Mips require who choose “X” play first and present original grid (Figure P.4.4.1)

4.5. Input a number to choose favorite space:

```
Please choose your location you want (from 0 to 8):
```

In this step, a small grid was presented, which helps the player to see clearly to choose a number space (from 0 to 8) for the purpose of drawing “X” of “O”. When she/he input a integer number greater than 8 or less than 0, the program will print a messenger (Figure P.4.5.1) and require the player to input a suitable number to continue playing in below steps.

```
Error: OUTOFRANAGE!  
Please input another number. Please choose your location you want (from 0 to 8):
```

(Figure P.4.5.1)

In case, a player chose a space number drawn before. It will send a messenger in the screen and require the player choose another space number: (Figure P.4.5.2)

```
Error: DUPLICATION!  
Please input another number: Please choose your location you want (from 0 to 8):
```

(Figure P.4.5.2)

After that program will help players check win condition at this turn (can see clearly at 4.5)

4.6. Checking win condition:

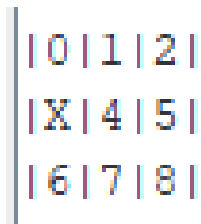
Before checking the step, the program will store “1” (if X turn) or “4” (if O turn) at a suitable register. For example, the player, in X turn, chose space number 4, \$s4 will store 1.

Next, the program will check all couple registers, which could lead to a win case, in order to show if the player will be a winner. For instance, assuming the player chooses space number 3 at X turn. Clearly, the program has to check if the total stored number of \$s0 and \$s6 is 2 or not. Else it continues, check the figure of \$s4 and \$s5. If it is equal to 2, the player wins at this step (*Figure P.4.6.1*).



(*Figure P.4.5.1*)

Unless, the program will print a grid with space number 3 is drawn “X” on (*Figure P.4.6.2*). And the competition could continue with O turn.



(*Figure P.4.6.2*)

4.7. Printing:

Before printing the grid, at the “string” array, the number must be replaced by the player's sign (X or O). Mars Mips will go to the index, which is equal to the newest inputted number, in order to this number by the sign.

It is following this step by printing string so as to show the hoped grid to players. The program prints all elements in the “string” array in terms of a three- by - three grid.

If the turn is the last turn (turn 9), the program will show that it is a draw played by the below messenger(*Figure P.4.7.1*).

|Draw

(Figure P.4.7.1)

Contrast, the next player can start her/his turn to get a chance to win. (Figure P.4.7.2).

|The next turn is O turn

(Figure P.4.7.1)

5. **CONCLUSION:**

In conclusion, the Tic-Tac-Toe game is not only a relaxing game, it is also a useful practise for those who want to study about Mars Mips. By solving this game, I have studied deeply and gotten more knowledge about MIPS programming. Thanks to this assignment, I could have a chance to practice well.

6. **REFERENCE:**

<https://sweetcode.io/building-first-simple-program-mips-assembly-language/>

<http://courses.missouristate.edu/kenvollmar/mars/>

<https://en.wikipedia.org/wiki/Tic-tac-toe>

https://www.exploratorium.edu/brain_explorer/tictactoe.html