HO CHI MINH UNIVERSITY OF TECHNOLOGYFACULTY OF

COMPUTER SCIENCE AND ENGINEERING

COMPUTER ARCHITECTURE (LAB)

REPORT

ASSIGNMENT: CALCULATOR BY MIPS

CLASS CC01 --- SEMESTER 232

Lecturer: Mr. Nguyễn Thiên Ân

Full Name: Bùi Quốc Bảo
Student ID: 2152411

Ho Chi Minh City, May 2024

TABLE OF CONTENTS

I. SUMMARISE INFORMATION

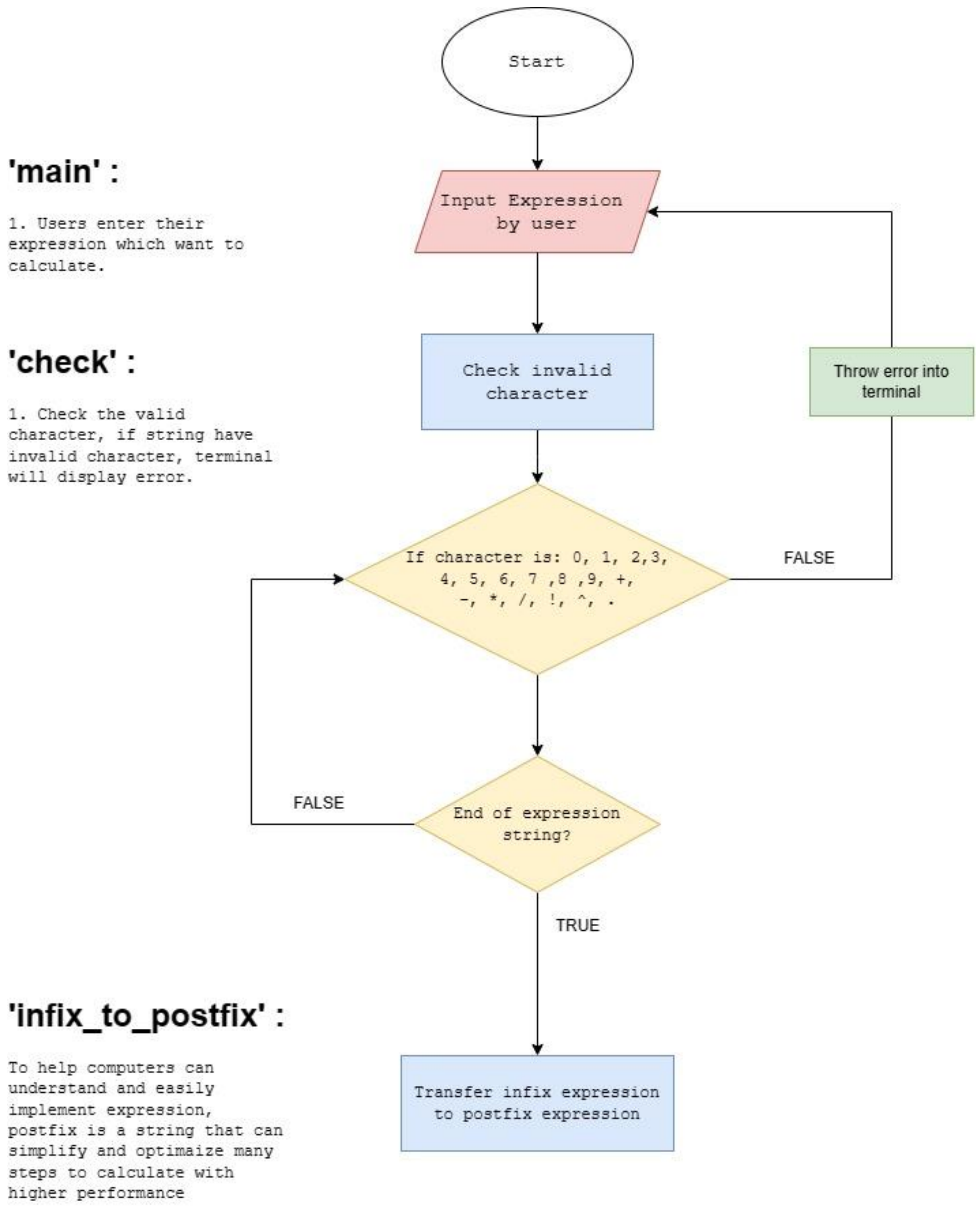
According to assignment, I decide to divide this my MIPS program into 5 main models to complete all requirements:

- **Main:** users must enter their expression to calculate, and I am going to check all character that guarantee all of them are valid.
- **Infix_to_Postfix:** human usually use infix expression to calculate by hand, which cause many difficult to computer can understand and calculate because of the different priority of math operators. Therefore, I decide to transfer the infix expression to postfix expression to simplify number of calculation steps, optimize and enhance the performance, which can help reduce time and error occuring in executed time.
- **Implement calculation:** reading the postfix from left to right bit by bit, and transfer data type from character to double (this will be explained further in following parts). After that, I use stack which is data structure to implement calculation all expression.
- **Display result (terminal & log file):** only using for displaying the result on terminal and treat some problems to write this result into calc_log.txt
- **Reset data:** because the requirements of this assignment are MIPS program which can run continuosly until user enters ‘quit’ to exit the program. Thus, we need to reset all registers which were used before unless only a f-register to store the result that can be reused for the next expression by M.

II. EXPLAINATION

2.1. Main

The flowchart below will demonstrate how we can input string and check the invalid character:


Start
Input Expression by user
Check invalid character
If character is: 0, 1, 2,3, 4, 5, 6, 7 ,8 ,9, +, -, *, /, !, ^, .
FALSE
Throw error into terminal
End of expression string?
FALSE
TRUE
Transfer infix expression to postfix expression

'main' :

1. Users enter their expression which want to calculate.

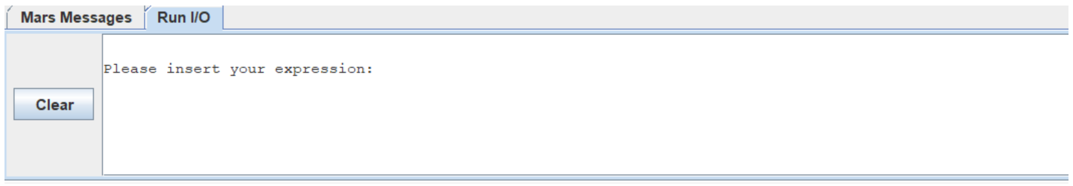
'check' :

1. Check the valid character, if string have invalid character, terminal will display error.

'infix_to_postfix' :

To help computers can understand and easily implement expression, postfix is a string that can simplify and optimaize many steps to calculate with higher performance

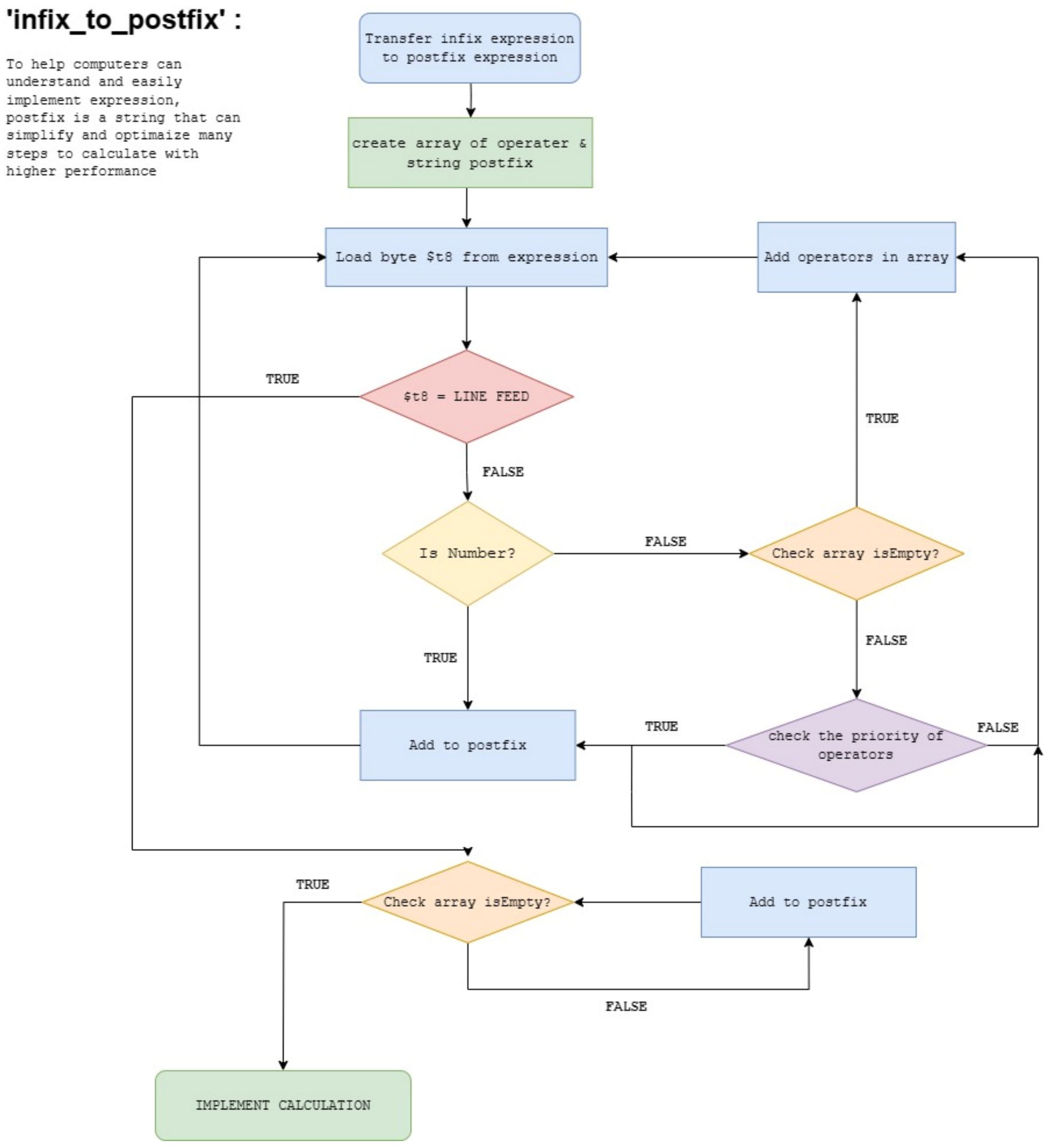
1. Firstly, when we run the program, the terminal will display promt: *“Please insert your expression: “* to announce for user who enter their expression as a string.



```
Please insert your expression:
```

2. After users had entered, the program will load each byte to comparision with given valid character in assignment file. If there is a invalid character, terminal will display promt: *"You inserted an invalid character in your expression"* and give users another announcement to enter expression again.
3. The process will loop until it meets “LINE FEED”, and the program will jump the next section.

2.2. Infix to Postfix


'infix_to_postfix' :

To help computers can understand and easily implement expression, postfix is a string that can simplify and optimaize many steps to calculate with higher performance

The most difficult problem in this module is how to transfer infix to postfix, so I will take an example to explain these problems:

The expression input: 5+((1+2)*4)+3

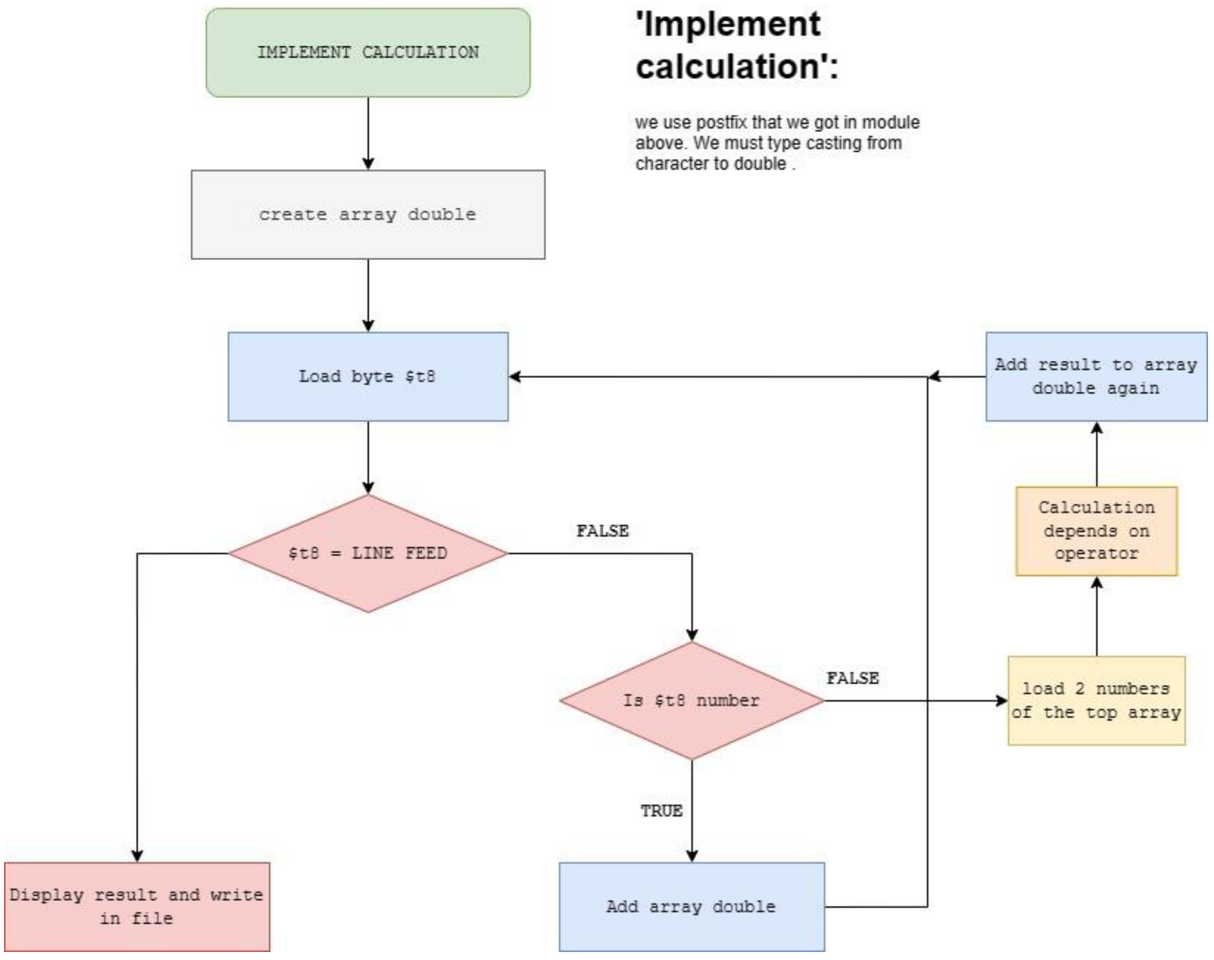
We can implement like the table below:

Character	Implement	Array of operator	Postfix String
5	Push into postfix string		5
+	Push into array	+	5
(	Push into array	+((	5
(	Push into array	+((	5
1	Push into postfix string	+((	5 1
+	Push into array	+((+	5 1
2	Push into postfix string	+((+	5 1 2
)	When meets ')' , we pop the operator until meeting '(' and delete it	+(	5 1 2 +
*	Push into array	+(*	5 1 2 +
4	Push into postfix string		5 1 2 + 4
)	Do it again like above	+	5 1 2 + 4 *
+	In this step, we need to comparision the priority of operators. Because '+' in array has the same priority so pop '+' to push into postfix string, then push the left '+' into array	+	5 1 2 + 4 * +
3	Push into postfix string	+	5 1 2 + 4 * + 3
	The final step, we must check whether array is empty or not. If it empty, move on; but if it is not empty, we must push all operator into postfix string		5 1 2 + 4 * + 3 +

We must notice about priority of operators: if we read the operator which differ '(' and ')', we have 2 case:

1. The operator which we are going to add is more priority the operator of the top of array, we just push the new operator into array
2. The operator which we are going to add is less than or equal priority the operator of the top of array, we must pop operator of the top of array and push it into postfix string until we meet the operator of the top of array which is less priotity than the new operator, then we push the new operator into array.

2.3. Implement calculation

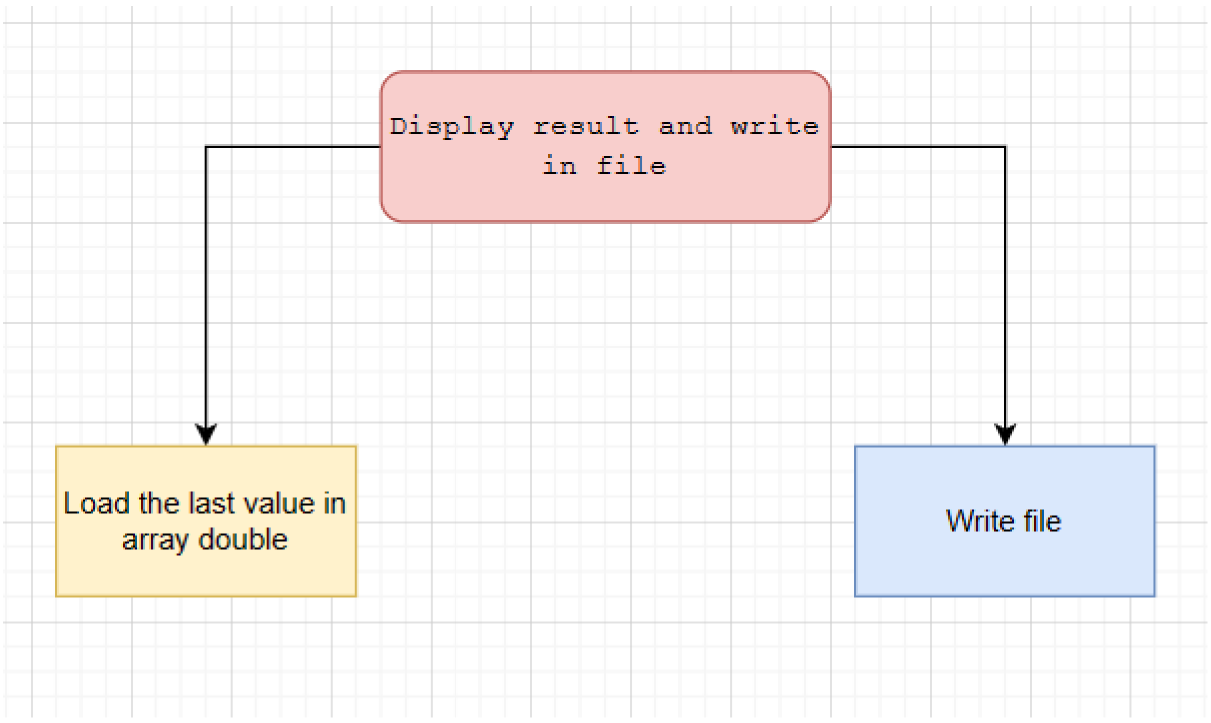

IMPLEMENT CALCULATION
'Implement calculation':
we use postfix that we got in module above. We must type casting from character to double .
create array double
Load byte $t8
Add result to array double again
Calculation depends on operator
$t8 = LINE FEED
FALSE
Is $t8 number
FALSE
load 2 numbers of the top array
TRUE
Display result and write in file
Add array double

The method sovle that how to type casting that turn character into double in

MIPS:

1. When we load each byte to $t8 from postfix string, I use $t6 register to keep the value ($t6 = $t8 - ‘0’). After a loop, if we do not meet the white space, I will multiple $t6 with 10 and plus with byte we just load on the top of loop. Therefore, if we meet the white space, I turn $t6 into double by ‘mtc1’ instruction and add it to array double.
2. If $t8 is operator, we move on the calculation module that depends on operators. Next, we load 2 numbers of the top array double and calculate. Then we put the results into the array again.
3. The last value in array will be the result for the expressio. If $t8 is LINE FEED, we move on to display result module.

2.4. Display result


Display result and write in file
Load the last value in array double
Write file

The last value in the double array wil be the result

The method to write result in calc_log.txt: transfer the result from double to string.

2.5. Reset data

In this section, I will load all the register that I used for these section above as 0, but I need to keep the result as M for the next time if user haven’t entered ‘quit’ yet. Therefore, I only keep the register $f30 to keep the last value for M value.

III. REFERENCE

1. Phạm Quốc Cường. (2017). Giáo trình Kiến Trúc Máy Tính. Đại học Bách Khoa: NXB Đại học Quốc Gia Tp.HCM
2. Davida. Pattern – John L. Hennessy. (2014). Computer Organization And Design.