



EAST WEST UNIVERSITY

“Computer Architecture”

CSE360

Section: 03

Mini Project on

Stack machine ISA: Design a stack machine, its instruction set must be stack oriented. (No Register)

Submitted to

Dr. Md. Nawab Yousuf Ali

Professor

Department of Computer Science & Engineering

Faculty of Science and Engineering

Submitted by

Student ID	Student Name
2019-1-60-022	Shuvo Kumar Das
2019-1-60-024	Adri Saha
2018-2-60-009	Khadija Akter

Submission Date: 08th May 2022

Table of Contents

Introduction	3
Problem Definition	3
Objectives	3
Methodology	3
Algorithm	4
Flow Chart	4
Code(.cpp file)	5
Implementation Method	5
System Requirements	5
Debugging Test	5
Result Analysis	6-7
Time Complexity	8
Conclusion	8
Future Improvement	8
Acknowledgement	8
References	9

Introduction

In computer science, computer engineering and programming language implementations, a stack machine is a computer processor or a virtual machine in which the primary interaction is moving short-lived temporary values to and from a push down stack. In the case of a hardware processor, a hardware stack is used. The use of a stack significantly reduces the required number of processor registers. Stack machines extend push-down automaton with additional load/store operations or multiple stacks and hence are Turing-complete.

Problem Definition

Design a stack machine, its instruction set must be stack oriented (no register!)

Objectives

Main goal of this project is to build a stack machine, which won't use any type of register/accumulator. The system will be totally based on stack. Stack is a data structure, which basically use the LIFO (Last In First Out) process. As this system is not allowed to use any register, so the value will be directly inserted into the stack and by the rule of stack machine, it will complete its operations like, sum, subtraction, multiplication etc.

Methodology

Push(), Pop(), top() are the main operations to add values and remove values. Push() is used for adding any value and pop() is used for deleting any value from the top. Top() function is used for fetching the first element of the stack. This stack machine will calculate the values based on the sign between the given numbers. It will work first for the first coming values and so on. Finally, only a single value will be stored in the stack and that will be the result. For example: $7 + 5 = ?$

Steps:

1. Push 7
2. Got the sign (+)
3. Push 5
4. Fetch the top 5(Stored in a variable)
5. Pop 5
6. Fetch the top 7(Stored in a variable)
7. Add $5+7 = 12$
8. Push 12 (Which is the required result)

Algorithm

stackMachine(String[], arraySize n)

Step-1: Start.

Step 2: Take user input until input = stop.

Step 3: Loop for iteration from the first index of the string to the last index n.

Step 4: if the element of the string is operand push it into stack

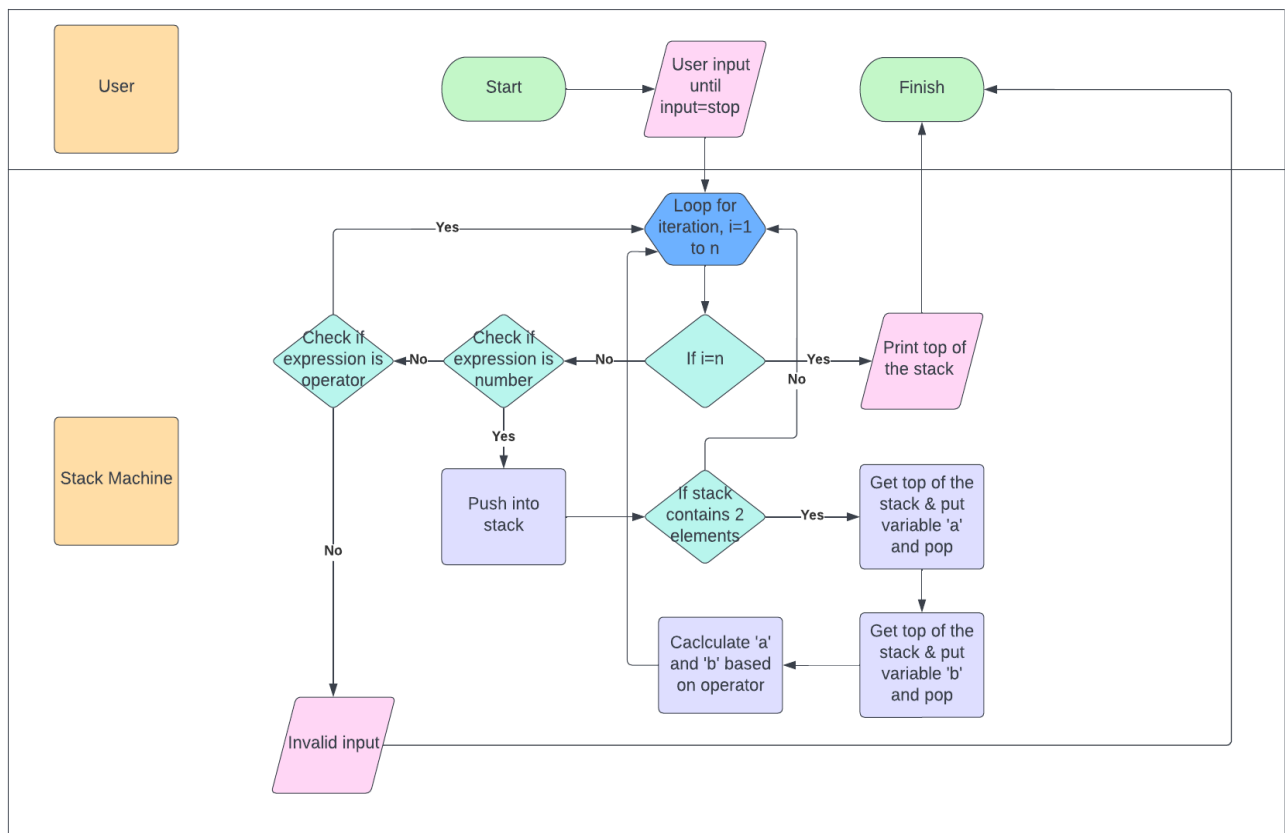
Step 5: If the element is operator, then go for the next element to check the operand and then push it into the stack.

Step 6: If there are 2 elements successfully pushed on the stack, then calculate them with the operator.

Step 7: Then pop the top of the element which is result and print top of the stack

Step 9: Finish.

Flow Chart



Code(.cpp file)



Stack Machine.cpp

https://drive.google.com/file/d/10PHrBSqDLS0SfKJ_rEPe9opaIgPz8942/view?usp=sharing

Implementation:

Functional Modules of Code:

We have implemented a function named stackMachine. This is used for the main calculation of the Stack Machine. We didn't use the built in stack in the code. We have created the Stack functions like- top(), pop(), push() etc. by using data structure.

Built in Documentation:

As our program can take multi digit numbers as input, that's why we have taken the input as String. To handle the string we have use a built in function, which is stod(). It converts the String to double. So that, we can calculate the values.

System Requirements

- ✓ Compiler: CodeBlocks, or any other compiler which can execute C++.
- ✓ RAM: 4 GB
- ✓ Operating System: Windows 7/8/10

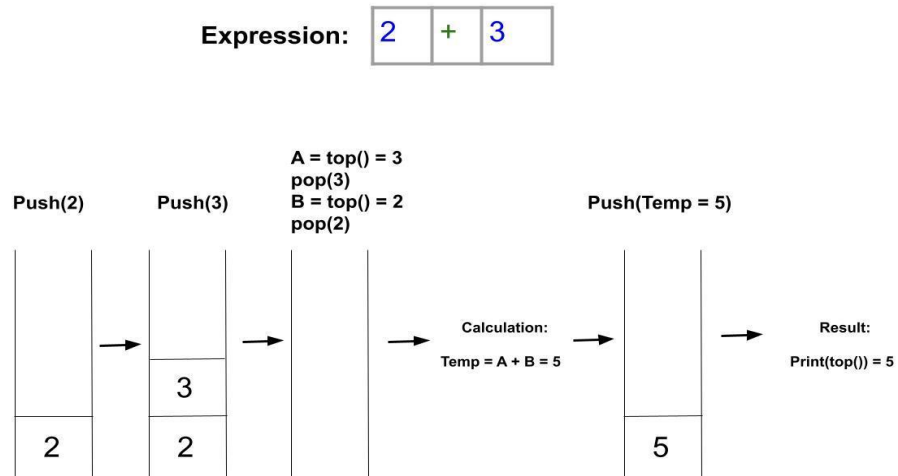
Debugging Test

We have noticed that in many of the online sources, they make the program only for the single digit numbers. But we have done something better from this. Our program can deal with the multi

digit numbers like- 12, 100 and could be any real number. Our Program has been debugged very nicely and it can detect wrong expressions/invalid expression.

Result Analysis

Example 1:



Output:

```

***** 'Stack Machine' *****

1. Push
2. Pop
3. Add
4. Subtract
5. Multiplication
6. Division
7. Enter any expression
8. Exit

7

Enter 'stop' to stop taking input

2 + 3 stop

Stack after push: 2

Stack after push: 3 2

Stack after pop: 2

Stack after pop: Stack is empty

Stack after addition: 5

Stack after pop: Stack is empty

Result: 5

```

Example 2:

<u>Instruction</u>	<u>Stack</u>
PUSH 3	3
PUSH 8	8, 3
ADD	11
PUSH 10	10, 11
<u>MUL</u>	110

Output:

1. Push	1. Push
2. Pop	2. Pop
3. Add	3. Add
4. Subtract	4. Subtract
5. Multiplication	5. Multiplication
6. Division	6. Division
7. Enter any expression	7. Enter any expression
8. Exit	8. Exit
	3
	ADD
	Stack: 11
1	
Push: 3	1. Push
Stack: 3	2. Pop
	3. Add
1. Push	4. Subtract
2. Pop	5. Multiplication
3. Add	6. Division
4. Subtract	7. Enter any expression
5. Multiplication	8. Exit
6. Division	
7. Enter any expression	1
8. Exit	Push: 10
	Stack: 10 11
1	
Push: 8	1. Push
Stack: 8 3	2. Pop
	3. Add
	4. Subtract
	5. Multiplication
	6. Division
	7. Enter any expression
	8. Exit
	5
	MUL
	Stack: 110

Time Complexity

Time Complexity of Stack machine is $O(n)$. Because we used single loop all over the code, and there is no nested loop present in the stackMachine function.

The array size is n , so the iteration will be active from 1 to n . We have implemented the push(), pop(), top() etc functions by own self.

There is no loop present in those mentioned functions except print(). So the time complexity for these functions will be $O(1)$, which is constant.

Conclusion

As per the title of the project, we didn't use any register to design the stack machine. This was little bit challenging. But we have learned a lot from here that, how the stack machine works and what is the working process of it.

Future Improvement

- ✓ This program can't understand that; which operator should be work first. As it works with the system of Stack, it works sequentially. In future we will try to solve it.

- ✓ This program can't handle with special character like comma, percentage sign, dollar sign and others. So, we will try to debug it more efficiently, which will be able to detect those.

- ✓ It can't deal with alphabets. In the future we will make this program comfortable with alphabets also.

Acknowledgement

We would like to sincerely thank our sir Dr. Md Nawab Yousuf Ali for guiding us throughout this project.

Reference

1. Tiwari, Ayushman. (2019). What is the time complexity of the push and pop operation of an array-based stack?. *Quora*. Available at: <https://www.quora.com/What-is-the-time-complexity-of-the-push-and-pop-operation-of-an-array-based-stack>

2. Year Published: 2020. Stack machine in Computer Organisation. *GeeksforGeeks*. Available at: <https://www.geeksforgeeks.org/stack-machine-in-computer-organisation/>

3. Year Published: 2021. Stack machine. *Wikipedia*. Available at: https://en.wikipedia.org/wiki/Stack_machine

4. Year Published: 2020. Compiler Design Module 51 : Stack Machine. *YouTube*. Available at: <https://www.youtube.com/watch?v=eYk0EennUrA>
5. https://en.wikipedia.org/wiki/Stack_machine#:~:text=In%20computer%20science%2C%20computer%20engineering,a%20hardware%20stack%20is%20used.

Thank You