Matrix Multiplication in 8085

MINI PROJECT REPORT

Submitted by,

M ASEENA SULTHANA [Reg No: RA2211030010317]

K CHARANYA [Reg No: RA2211030010304]

Under the guidance of

Dr. R Nithya Paranthaman

Associate Professor, Department of Computing Technologies

In partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTING TECHNOLOGIES

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

NOVEMBER 2023

DEPARTMENT OF
SCHOOL OF COMPUTING

College of Engineering and Technology

SRM Institute of Science and Technology

MINI PROJECT REPORT

ODD Semester, 2023-2024

Lab code & Sub Name : 21CSS201T & Computer Organization and Architecture

Year & Semester : II & III

Project Title : **Matrix Multiplication in 8085**

Lab Supervisor **: Dr.R. Nithya Paranthaman**

Team Members : 1. M Aseena Sulthana (Reg.No:RA2211030010317)

2. K Charanya (Reg.No:RA2211030010304)

PARTICULARS	**MAX. MARKS**	**MARKS OBTAINED**
		Name:
		Reg no:
Program and Execution	20	
Demo verification & Viva	15	
Project Report	5	
Total	**40**	

Date:

Staff Name:

Signature:

Matrix Multiplication in 8085

OBJECTIVE

The objective of this project is to integrate a matrix multiplication algorithm onto the Intel 8085 microprocessor, optimizing the algorithm to ensure compatibility with the 8085 architecture. The project aims to evaluate the performance of the implemented algorithm in terms of execution time, memory usage, and overall efficiency. Through comprehensive documentation, including code, flowcharts, and any necessary modifications, the project seeks to demonstrate a deep understanding of both the matrix multiplication algorithm and the Intel 8085 microprocessor. Additionally, practical applications of matrix multiplication on the 8085 will be explored, with the overall goal of contributing valuable insights to the field of integrating mathematical algorithms on microprocessor architectures.

ABSTRACT

Two matrices can only be multiplied if their orders are of the form $m \times n$ and $n \times p$ where $m,n,p \in Z+$. In this project we intend to mulitply matrices of order $1 \times n$ & $n \times 1$.Later on, we may implement for general orders.

INTRODUCTION

Multiplying two matrices of order $m \times n$ and $n \times p$ where $m,n,p \in Z+$ is an $O(n^3)$ where n is the maximum of m,n,p. The project seeks to implement matrix multiplication for smaller order matrices on an Intel 8085 Microprocessor. As you compile the program step by step using GNUSim 8085 Microprocessor you could visualize each row of the product matrix being filled.

As there is no direct multiplication operation available in 8085 Instructions, we intend to multiply numbers through repeated addition method using a loop.

In order to traverse through a row in Matrix 1 & a column in Matrix 2, we first load the starting address of row and column in stack and HL pair respectively. For traversing through row and column we swap the values in HL register pair and top of stack and increment them. We call multiplication sub-routine as and when we require multiplication of 2 numbers.

IMPLEMENTATION

Algorithm for Matrix Multipication:

```
for ( int i = 0 ; i < rowNo ; i++ ){
for ( int j = 0 ; j < colNo ; j++ ){
        for ( int k = 0 ; k < p ; k++ ){
                result[i][j] = result[i][j] + first[c][k]*second[k][d];
            }
        }
}
```

Algorithm for Multiplication

```
int number1, number2;
while( number2 != 0 ){
      number1 = number1 + number2; number2-1;
}
```

Matrix Multiplication Algorithm for 8085 for $1 \times n$ **&** $n \times 1$

Load HL pair with Address of 1st row and 1st column of Matrix1
Load Stack with Address of 1st row and 1st column of Matrix2
MVI E, 00H
Method : Load value in HL memory location in A register
Load value of stack in B register
Call multiply subroutine to multiply two numbers
ADD E
STA E
INX H
XCHG
INX H
JMP Method
Store the value of E in specified memory Location

Matrix Multiplication Algorithm for 8085 for 2×2 **&** 2×2

Load C with 2
Load D with 2
Method1: DCR C
Method: Multiply row 1 vector with column 1 vector using algo defined above
DCR D
if D != 0:
if C != 0: Load HL pair with add. of Matrix1[1][1]
Call Method if C == 0: Load HL pair with add. of Matrix2[2][1]
Call Method if D == 0:
Load HL pair with add. of Matrix1[2][1] MVI D,002H if C == 0: HLT
if C!= 0 : Call Method1

FLOWCHARTS

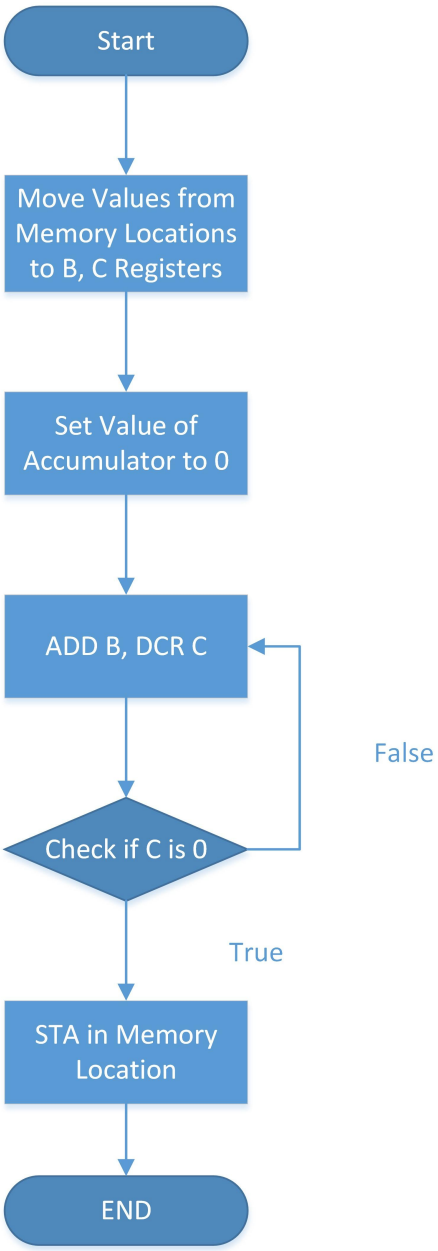


Figure 1: Multiplication of 2 numbers with column

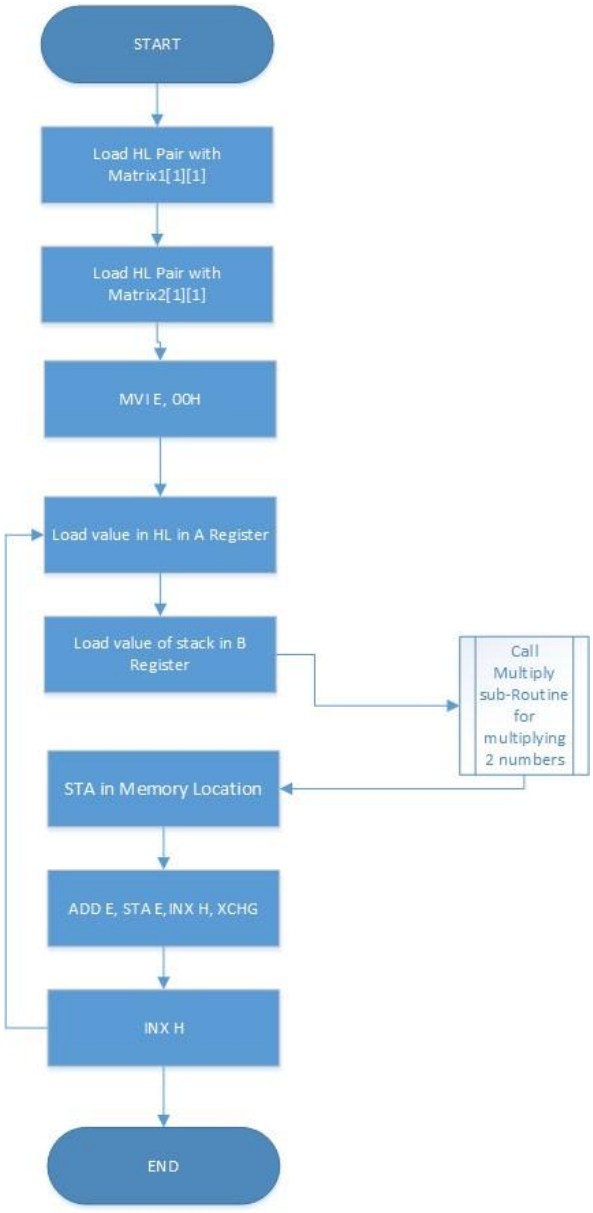


Figure 2: Mutliplying row vector

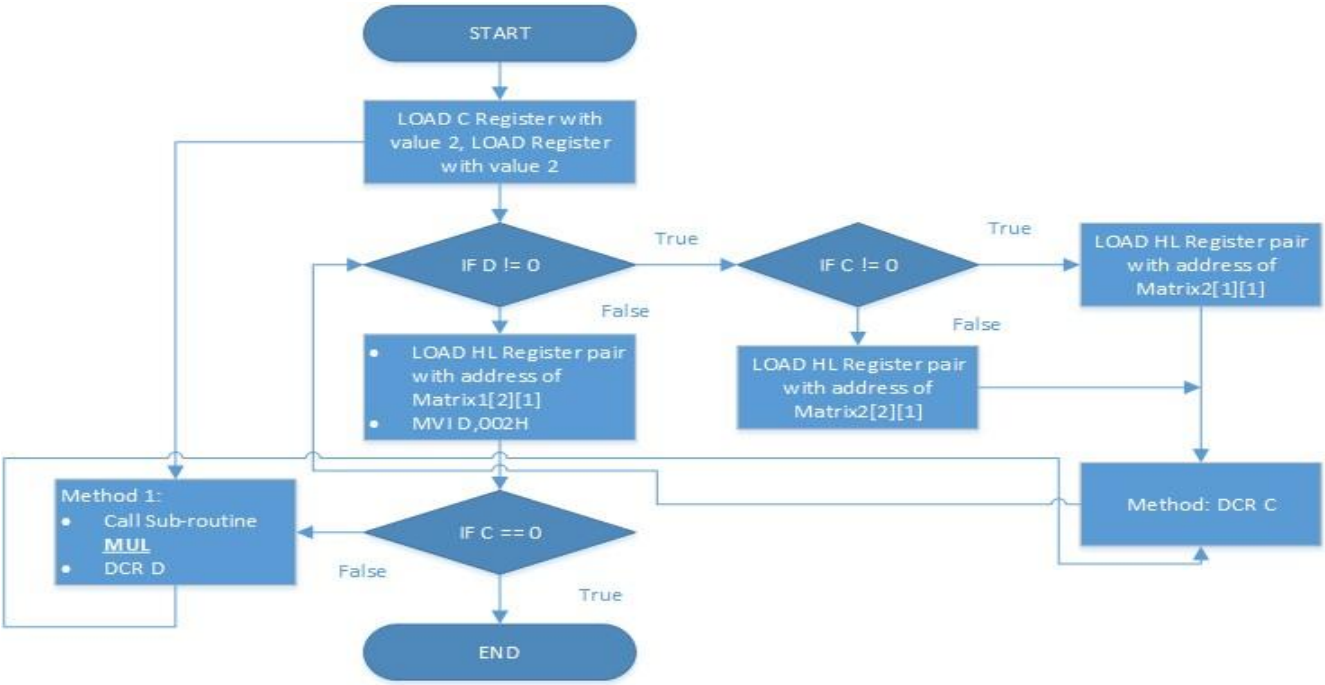


Figure 3: Matrix Multiplication

CODING : [In GNUSim8085]

Code for multiplication :

```
; code for multiplication of
; two numbers by repeated
; addition
; two numbers to be multiplied
; are stored in 0002H and
; 0003H,
; output is stored in 0004H
        MOV B,0002H
        MOV C,0003H
        MVI A,00H
LOOP: ADD B DCR C
        JNZ LOOP
        STA 0004H
```

Multiplying row vector with colum vector :

```
LXI H, 8500H
PUSH 8508H
Method: MOV M, A
XCHG
MOV M,B
CALL MUL
STA 8516H
INX H
XCHG
INX H
JMP Method
```

Matrix Multiplication :

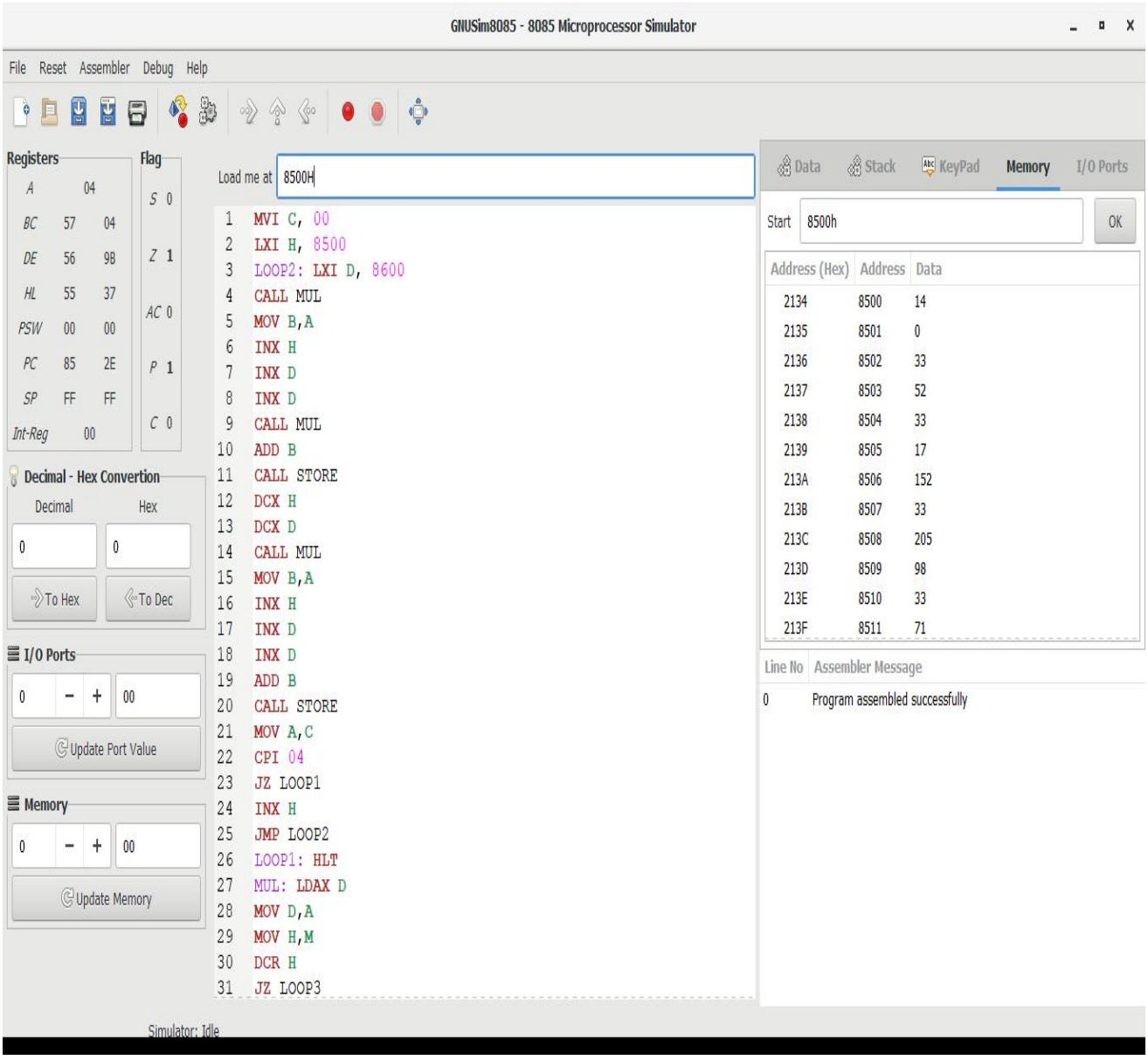
```
MVI C, 002H
MVI D, 002H
```

```
Method2: DCR C
Method3: CALL MRC
         DCR D
         JNZ Method4
Method4: ORI C, 00H
         JNZ Method5
Method5: LXI H, 8500H
         JMP Method3
         ORI C, 00H JZ
Method6: LXI H, 8508H
         JMP Method3
         ORI D, 00H
         JZ Method7:
 Method7: INX H, 8508H
     MVI D, 002H ORI C, 00H
     JNZ Method3
     ORI C, 00H JZ Method8
Method8: HLT
```

FINAL CODE :

```
MVI C, 00
LXI H, 8500
LOOP2: LXI D, 8600
CALL MUL
MOV B,A
INX H
INX D
INX D
CALL   MUL
ADD B
CALL STORE
DCX H
DCX D
CALL MUL
MOV B,A
INX H
INX D
INX D
ADD B
CALL STORE
MOV A,C
CPI 04
JZ LOOP1
INX H
JMP LOOP2
LOOP1: HLT
MUL: LDAX D
MOV D,A
MOV H,M
DCR H
JZ LOOP3
LOOP4: ADD D
DCR H
JNZ LOOP4
LOOP3: MVI H,85
MVI D,86
RET
STORE: MVI B,87
STAX B INR C
RET
```

OUTPUT



RESULT

We have successfully computed Matrix multiplication of orders $1 \times n$ & $n \times 1$ and 2×2 & 2×2 and stored them in memory locations.

PROBLEM

Provided we had 4 more registers it would have easier to generalized matrix multiplication for $m \times n$ & $n \times p$. The need for extra registers could have been overcome by the use of stack but there is a problem. After pushing the values in the stack, if we wish to access them in any order it is not possible. Moreover, if we pop the values of stack , it would alter HL register pair values which we do not wish to do so.

CONCLUSION

At present we have been successively in computing matrices of order $1 \times n$ & $n \times 1$ and 2×2 & 2×2. In conclusion, while it is possible to implement matrix multiplication using the 8085 microprocessor, it is not a straightforward or efficient task due to the processor's general-purpose nature and limited capabilities. Specialized hardware or more powerful processors are better suited for handling complex mathematical operations like matrix multiplication.

REFERENCES

[1] [PDF] Microprocessor Architecture, Programming and Applications with the 8085 By Ramesh Gaonkar Free Download – Learnengineering.in

[2] Intel 8080 Assembly Language Programming Manual Rev.B (1975) : Intel Corporation : Free Download, Borrow, and Streaming : Internet Archive

[3] Matrix multiplication - Wikipedia