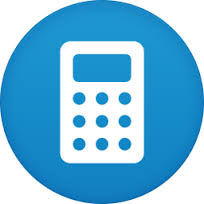
**“*Calculator Using Booth Algorithm” ***

*Computer Architecture & Organization-*

*BSE 3A*

*“Software Engineering”*

***Semester: Fall – 2021***

***Teacher Name:***

*“SIR SAMAR YAZDANI”* ***Lab Engineer:***

*“SIR REHAN BAIG”*

## *Group Members:*

## *SADIA AMBREEN (02-131192-056)*

## *DANIA AHMED(079)*

## *RIDA AFZAL (008)*

## 

***ACKNOWLEDGEMENT:*** *This project consumed huge amount of work, research and dedication. Still, implementation would not have been possible if we did not have a support of our teachers. Therefore we would like to extend our sincere gratitude to all of them. Also our Lab task help us.*

*We are also grateful to [Sir REHAN BAIG] for technical support in the implementation. Without their superior knowledge and experience, the Project would like in quality of outcomes, and thus their support has been essential.*

*Nevertheless, we express our gratitude toward our families and colleagues for their kind co-operation and encouragement which help us in completion of this project.*

***ABSTRACT:*** *The basic abstract was from our labs. This project consumed huge amount of work, research and dedication. Still, implementation would not have been possible if we did not have a support of our teachers. Therefore we would like to extend our sincere gratitude to all of them.*

*The Program is basically a calculator with 4 functions, i.e addition, subtraction, multiplication and division.. The addition and subtraction is simply done by their respective commands, but multiplication and division is implemented by booth algorithm.*

*The study was basically all in class and lab, those were all separate labs for each and everything. Addition and subtraction from the beginning of labs, switch statement from the last labs, booth algorithm learned in class and implemented with a lot of tries again and again up to the point we succeeded.*

***INTRODUCTION:*** *The Program is basically a calculator with 4 functions, i.e:-*

*Addition*

*Subtraction*

*Multiplication*

*Division*

*The addition and subtraction is simply done by their respective commands, but multiplication and division is implemented by booth algorithm.*

***OBJECTIVE AND SCOPE:*** *The both algorithm is useful for new students as a sample program .*

***ANALYSIS AND DESIGN:***

*Program is basically a calculator with 4 functions, i.e addition, subtraction, multiplication and division.. The addition and subtraction is simply done by their respective commands, but multiplication and division is implemented by booth algorithm.*

***IMPLEMENTATION:***

*Program is basically a calculator with 4 functions, i.e addition, subtraction, multiplication and division..*

*varword: .word main,case1,case2,case3,case4*

*"Enter multiplicand M :"*

*Enter multiplier Q :"*

*"Quotient is :"*

*Remainder is :"*

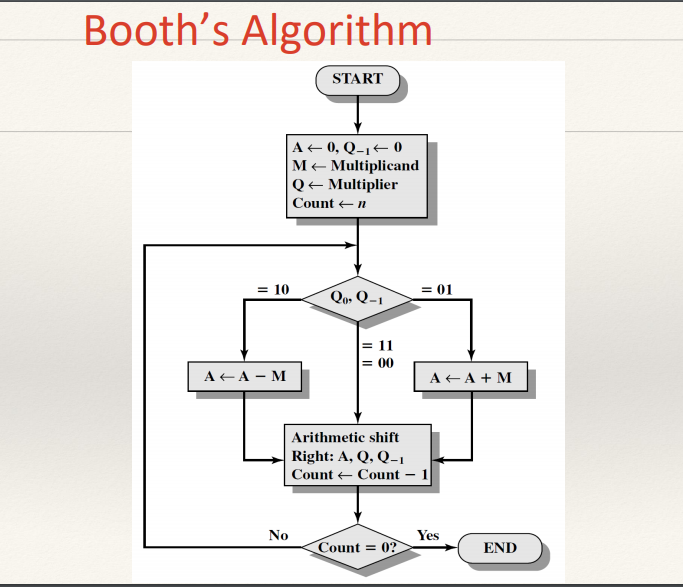
*a value from 1 to 4 "*

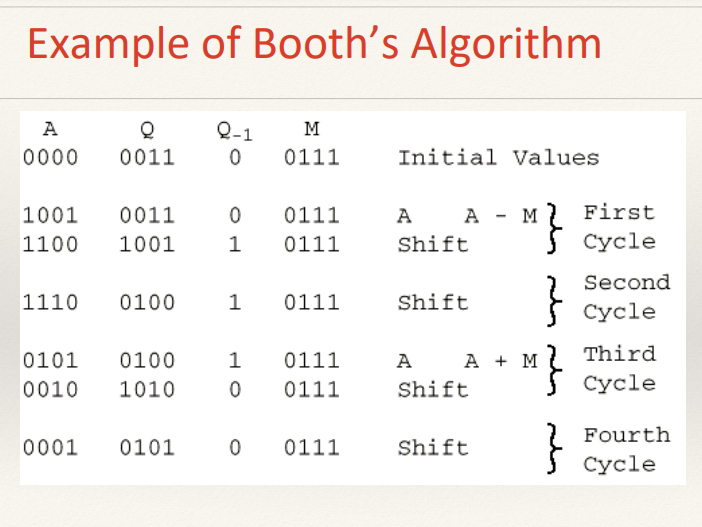
*you are in addition"*

*you are in subtraction"*

*you are in multiplication"*

*you are in division*





***PROGRAM CODING:***

***################# Data segment #####################***

***.data***

***.align 2***

***varword: .word main,case1,case2,case3,case4***

***mult1: .asciiz "Enter Multiplicand M : "***

***mult2: .asciiz "\nEnter Multiplier Q : "***

***div1: .asciiz "Quotient is :"***

***div2: .asciiz "\nRemainder is :"***

***input1: .asciiz "\n=========== CALCULATOR ON BOOTH ALGORITHM======== : "***

***input: .asciiz "\n=========== Type a value from 1 to 4 ======== : "***

***msg\_1: .asciiz "\n You are in Addition : "***

***msg\_2: .asciiz "\n You are in Subtraction : "***

***msg\_3: .asciiz "\n You are in Multiplication : "***

***msg\_4: .asciiz "\n you are in Division : "***

***################# Code segment #####################***

***.text***

***.globl main***

***main:***

***la $a0,input1***

***li $v0,4 # print input1 message***

***syscall***

***la $a0,input***

***li $v0,4 # print input message***

***syscall***

***li $v0,5 # read integer***

***syscall***

***blez $v0,main # default for less than 1***

***li $t3,4***

***bgt $v0,$t3,main # default for greater than 3***

***la $a1,varword # load address of varword***

***sll $t0,$v0,2 # compute word offset***

***add $t1,$a1,$t0 # form a pointer into variable***

***lw $t2,0($t1) # load an address from varword***

***jr $t2 # jump specific case "switch"***

***case1:***

***li $v0,4***

***la $a0,msg\_1***

***syscall***

***li $v0,5***

***syscall***

***move $t0,$v0***

***li $v0,5***

***syscall***

***move $t1,$v0***

***add $t5,$t1,$t0***

***move $a0,$t5***

***li $v0,1***

***syscall***

***b end***

***case2:***

***li $v0,4***

***la $a0,msg\_2***

***syscall***

***li $v0,5***

***syscall***

***move $t0,$v0***

***li $v0,5***

***syscall***

***move $t1,$v0***

***sub $t5,$t1,$t0***

***move $a0,$t5***

***li $v0,1***

***syscall***

***b end***

***case3:***

***li $t0,0***

***li $t1,0***

***li $t2,0***

***li $t3,0***

***li $v0,4***

***la $a0,mult1***

***syscall***

***li $v0,5***

***syscall***

***move $t1,$v0 # M***

***li $v0,4***

***la $a0,mult2***

***syscall***

***li $v0,5***

***syscall***

***move $t2,$v0 # Q***

***li $t0,32***

***loop:***

***andi $t5,$t2,1***

***beqz $t5,SRL***

***AplusM:***

***add $t3,$t3,$t1 # A+M where $t3=A***

***move $t4,$t3 # saving A***

***srl $t3,$t3,1 # shift right logical A***

***srl $t2,$t2,1 # shift right logical Q***

***andi $t4,1***

***beqz $t4,nochange***

***changto1:***

***ori $t2,2147483648***

***nochange:***

***b nochang***

***SRL:***

***move $t4,$t3 # saving A***

***srl $t3,$t3,1 # shift right logical A***

***srl $t2,$t2,1 # shift right logical Q***

***andi $t4,1***

***beqz $t4,nochang***

***ori $t2,2147483648***

***nochang:***

***li $t4,0***

***li $t5,0***

***addi $t0,$t0,-1***

***bgtz $t0,loop***

***li $v0,1***

***move $a0,$t2***

***syscall***

***b end***

***case4:***

***li $v0,4***

***la $a0,msg\_4***

***syscall***

***li $v0,5***

***syscall***

***move $t0,$v0***

***li $v0,5***

***syscall***

***move $t1,$v0***

***div $t0,$t1***

***mflo $t2***

***mfhi $t3***

***move $a0,$t2***

***li $v0,4***

***la $a0,div1***

***syscall***

***move $a0,$t2***

***li $v0,1***

***syscall***

***li $v0,4***

***la $a0,div2***

***syscall***

***move $a0,$t3***

***li $v0,1***

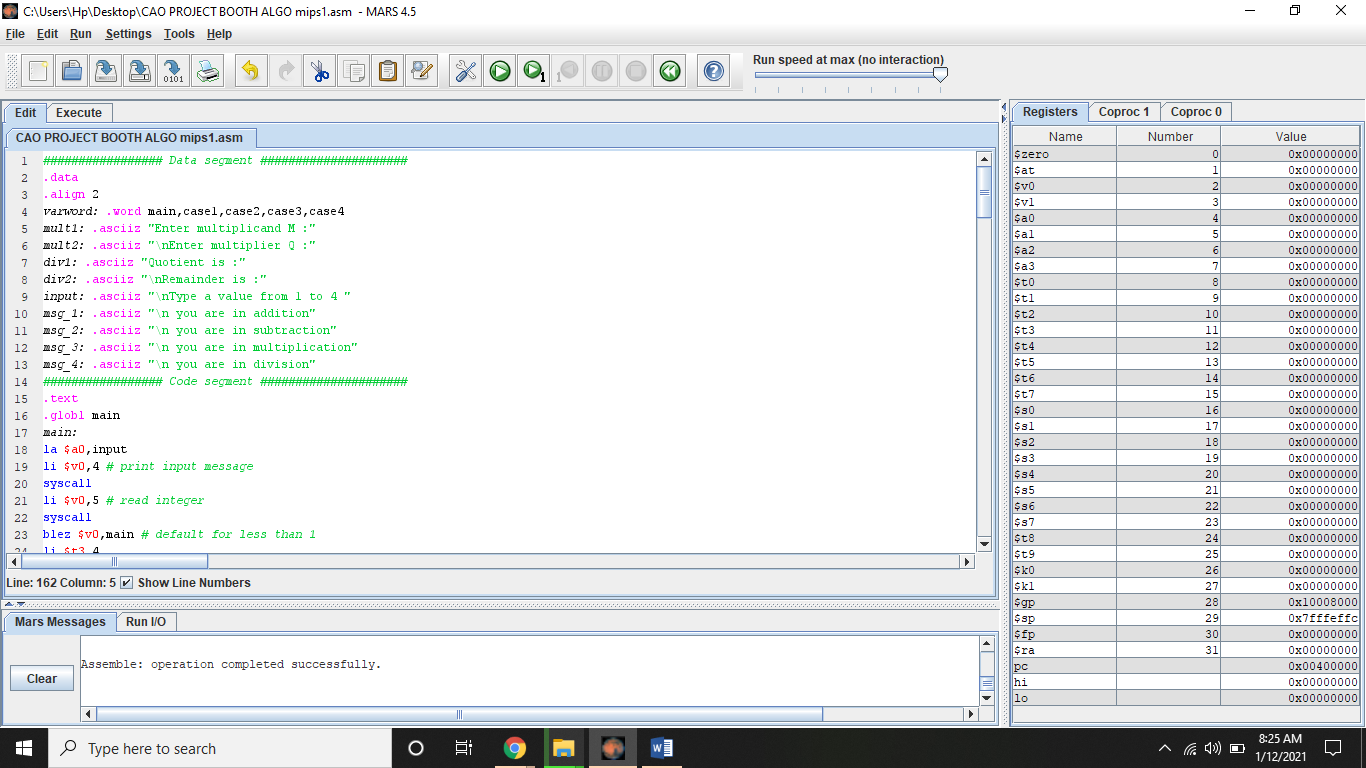
***syscall***

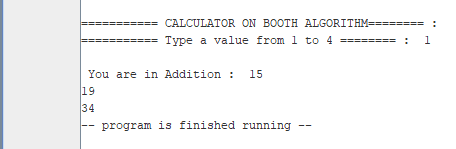
***end:***

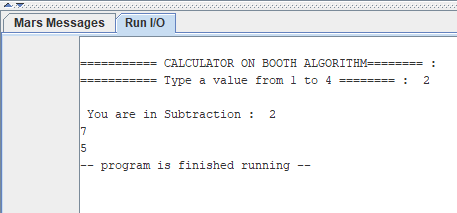
***li $v0,10***

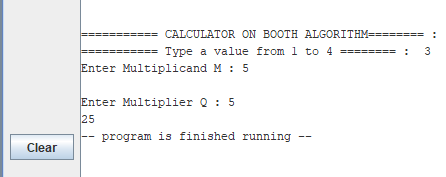
***syscall***

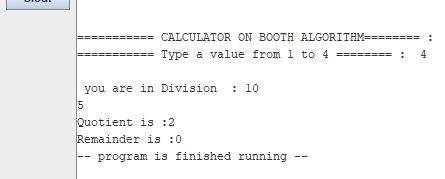
***OUTPUT:***

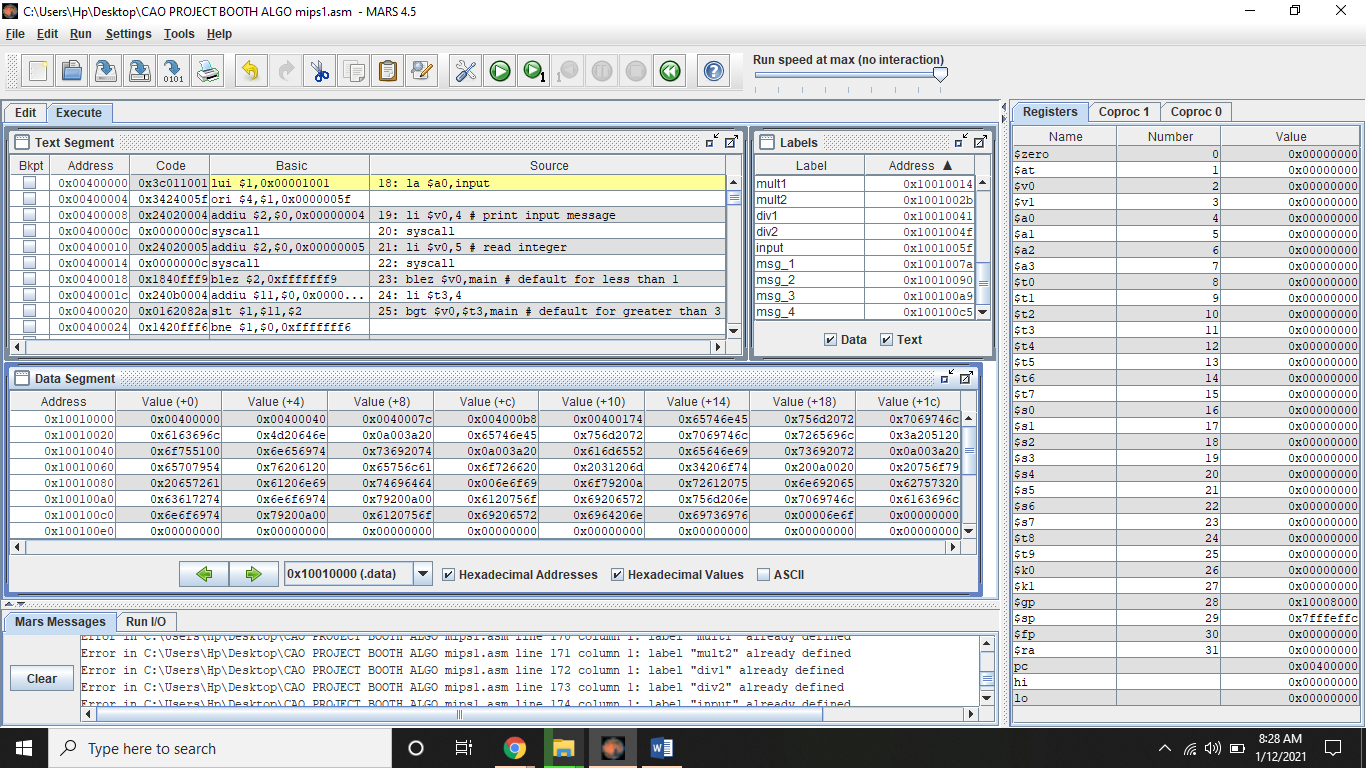
******









******

***RESULTS AND DISCUSSIONS:***

*Result is all as normal calculator done. Program is basically a calculator with 4 functions, i.e addition, subtraction, multiplication and division.. The addition and subtraction is simply done by their respective commands, but multiplication and division is implemented by booth algorithm.*

***CONCLUSION:***  *Result is all as normal calculator done. The Program is basically a calculator with 4 functions, i.e addition, subtraction, multiplication and division.. The addition and subtraction is simply done by their respective commands, but multiplication and division is implemented by booth algorithm.*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** ***. THE END***

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**