MIPS Processor Design

Course Name: EG 212, Computer Architecture-processor design
Student Name: Pranay Kelotra,Yash Anand,Areen Patil
Student ID: IMT2023563 ,IMT2023624, IMT2023013

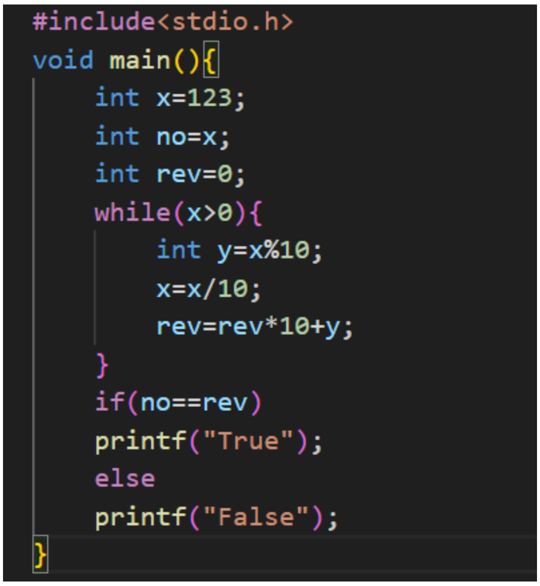
26 February 2024

1 Introduction

To demonstrate working of MIPS processor,we implemented 3 codes: palindrome checker,factorial and Fibonacci of a number. We wrote Assembly on MARS and it converted it into binary file(shortened to hexadecimal). Our processor reads binary file and generates the required output.

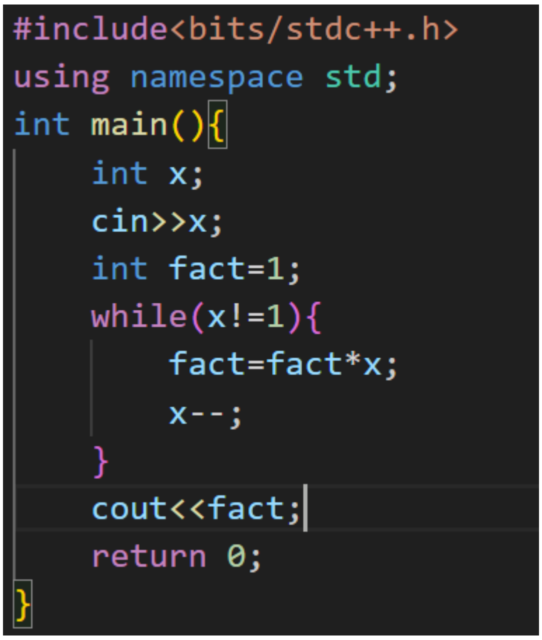
2 C++ Codes

2.1 PALINDROME CHECKER



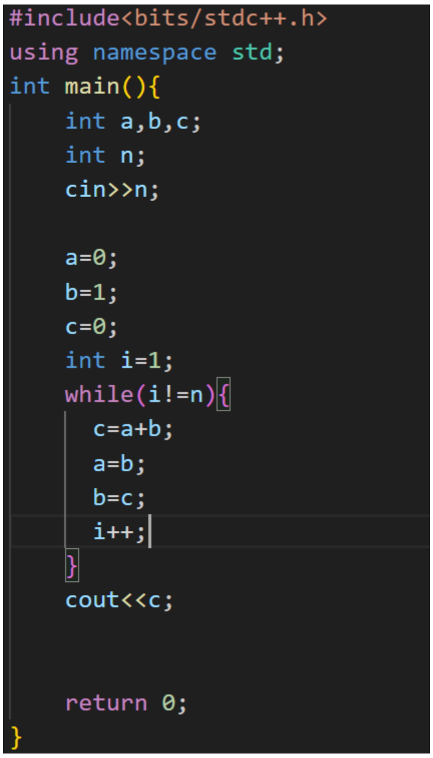
```
#include<stdio.h>
void main(){
    int x=123;
    int no=x;
    int rev=0;
    while(x>0){
        int y=x%10;
        x=x/10;
        rev=rev*10+y;
    }
    if(no==rev)
    printf("True");
    else
    printf("False");
}
```

2.2 FACTORIAL



```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int x;
    cin>>x;
    int fact=1;
    while(x!=1){
        fact=fact*x;
        x--;
    }
    cout<<fact;
    return 0;
}
```

2.3 FIBONACCI



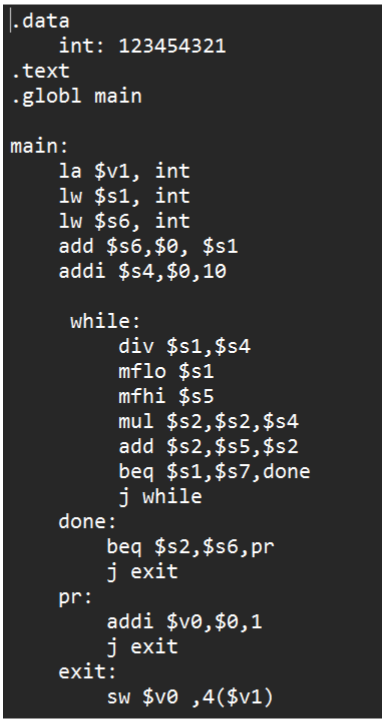
```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int a,b,c;
    int n;
    cin>>n;

    a=0;
    b=1;
    c=0;
    int i=1;
    while(i!=n){
        c=a+b;
        a=b;
        b=c;
        i++;
    }
    cout<<c;


    return 0;
}
```

3 Assembly language

3.1 PALINDROME CHECKER

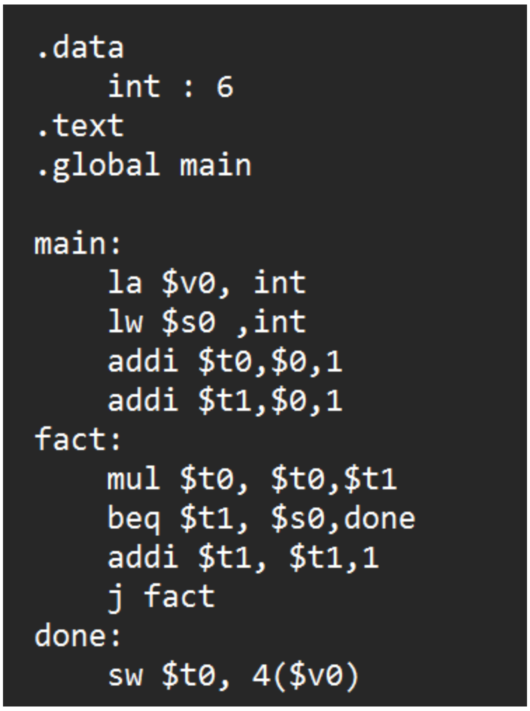


```
.data
    int: 123454321
.text
.globl main

main:
    la $v1, int
    lw $s1, int
    lw $s6, int
    add $s6,$0, $s1
    addi $s4,$0,10

     while:
         div $s1,$s4
         mflo $s1
         mfhi $s5
         mul $s2,$s2,$s4
         add $s2,$s5,$s2
         beq $s1,$s7,done
         j while
    done:
        beq $s2,$s6,pr
        j exit
    pr:
        addi $v0,$0,1
        j exit
    exit:
        sw $v0 ,4($v1)
```

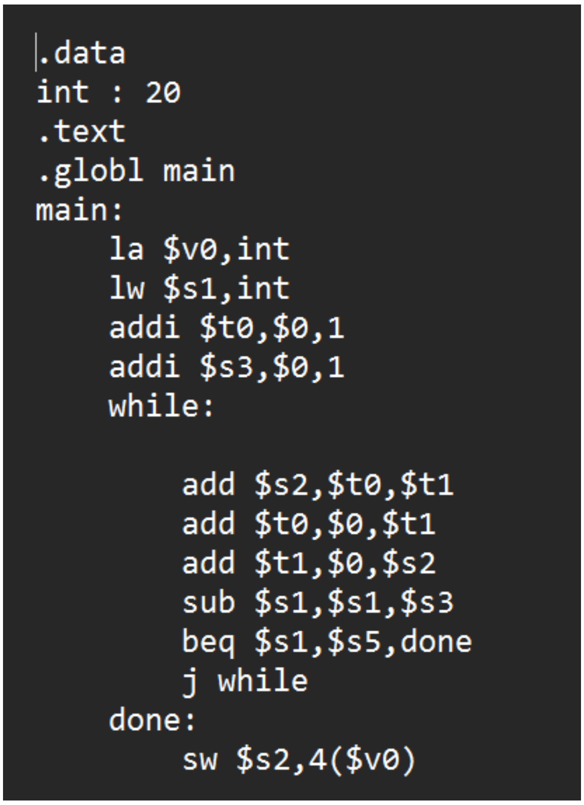
3.2 FACTORIAL



```
.data
    int : 6
.text
.global main

main:
    la $v0, int
    lw $s0 ,int
    addi $t0,$0,1
    addi $t1,$0,1
fact:
    mul $t0, $t0,$t1
    beq $t1, $s0,done
    addi $t1, $t1,1
    j fact
done:
    sw $t0, 4($v0)
```

3.3 FIBONACCI



```
.data
int : 20
.text
.globl main
main:
    la $v0,int
    lw $s1,int
    addi $t0,$0,1
    addi $s3,$0,1
    while:

        add $s2,$t0,$t1
        add $t0,$0,$t1
        add $t1,$0,$s2
        sub $s1,$s1,$s3
        beq $s1,$s5,done
        j while
    done:
        sw $s2,4($v0)
```

4 PROCESSOR

We divided our processor into 6 stages:Control Path,fetch,decode,execute,memory access and write back. Control path reads the opt code and function field if present and set all the select lines of muxes accordingly. In fetch we read the binary file(shortended to hexadecimal) and set instructions in decode phase. In decode we read its opt code and set the control path according to instruction. If required we us sign extend to make the address 32 bits. In execute we perform ALU operations according to opt code and function fields of instruction. If it is jump or BEQ we change the PC counter accordingly. In memory access we write to/ read from the data memory. And finally in write back stage we write back to registers if required.
The output for palindrome,factorial and fibonacci are following respectively.