Jawahar Education Societys Annasaheb Chudaman Patil College of Engineering, Kharghar, Navi Mumbai

NAME: PRIYUSH BHIMRAO KHOBRAGADE

PRN NO: 211112018

SUBJECT: DIGITAL LOGIC & COMPUTER ORGANIZATION AND ARCHITECTURE LAB

03

AIM: To implement restoring division algorithm

Practical No. 3

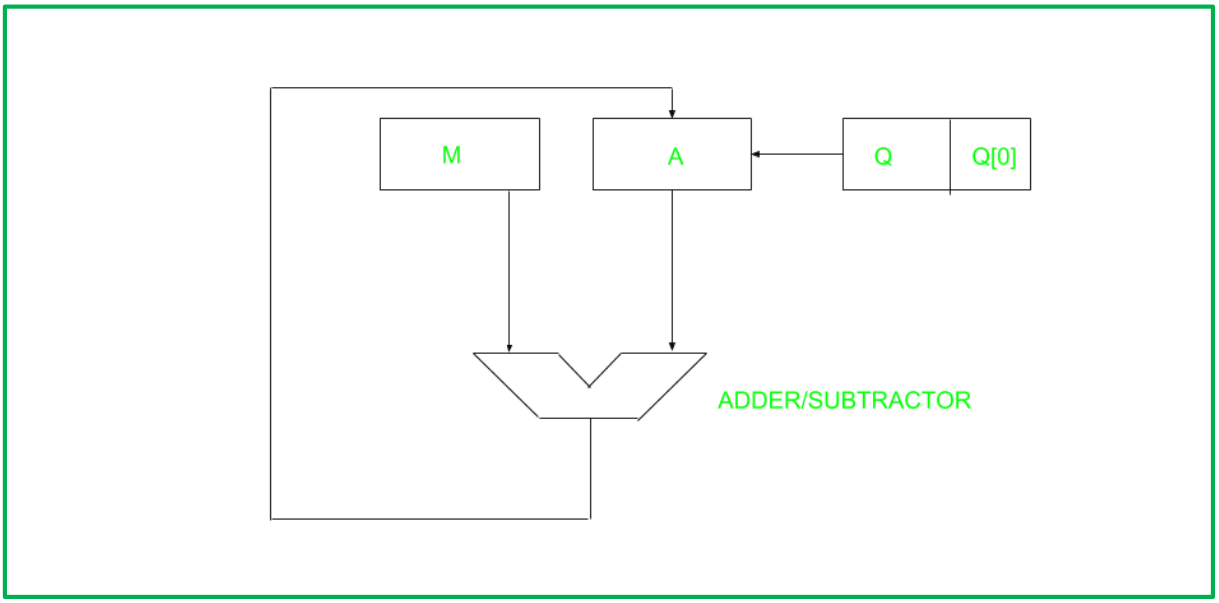
- **Aim**: To implement restoring division algorithm.
- **Objectives**: To implement the operation of the arithmetic unit using the restoring division algorithm.
- **Outcomes**: Learner will able to understand the implementation and working of restoring division algorithm.
- **Hardware / Software Required:** Any programming language C, Java etc.

- **Theory**:

Restoring Division Algorithm:

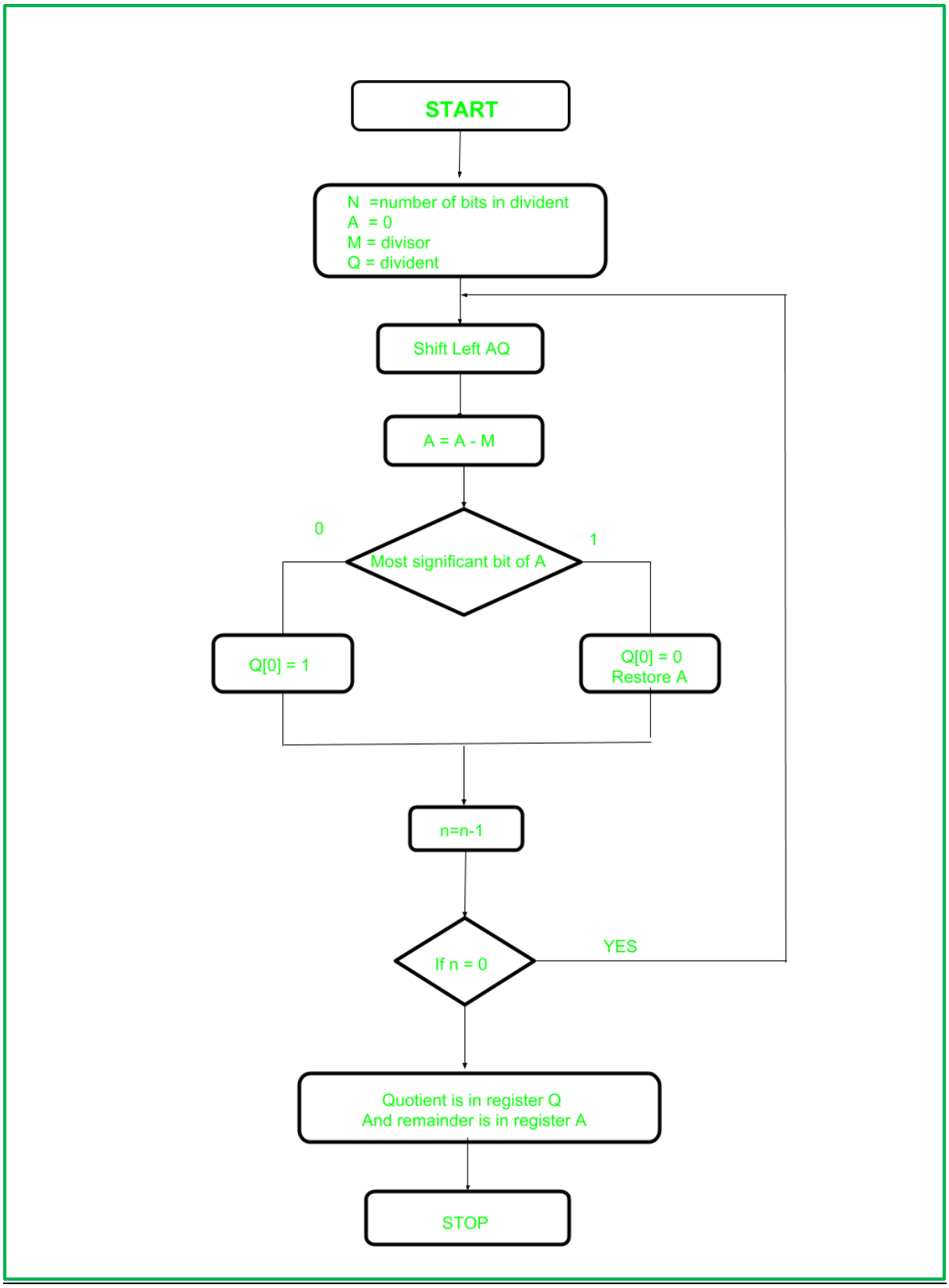
A division algorithm provides a quotient and a remainder when we divide two number. They are generally of two type slow algorithm and fast algorithm. Slow division algorithm are restoring, non-restoring, non-performing restoring, SRT algorithm and under fast comes Newton–Rap son and Goldschmidt.

In this article, will be performing restoring algorithm for unsigned integer. Restoring term is due to fact that value of register A is restored after each iteration.



AIM: To implement restoring division algorithm

Here, register Q contain quotient and register A contain remainder. Here, n-bit dividend is loaded in Q and divisor is loaded in M. Value of Register is initially kept 0 and this is the register whose value is restored during iteration due to which it is named Restoring.



AIM: To implement restoring division algorithm

Let's pick the step involved:

- Step-1: First the registers are initialized with corresponding values (Q = Dividend, M = Divisor, A = 0, n = number of bits in dividend).
- Step-2: Then the content of register A and Q is shifted left as if they are a single unit.
- Step-3: Then content of register M is subtracted from A and result is stored in A.
- Step-4: Then the most significant bit of the A is checked if it is 0 the least significant bit of Q is set to 1 otherwise if it is 1 the least significant bit of Q is set to 0 and value of register A is restored i.e the value of A before the subtraction with M.
- Step-5: The value of counter n is decremented.
- Step-6: If the value of n becomes zero we get of the loop otherwise we repeat from step 2.
- Step-7: Finally, the register Q contain the quotient and A contain remainder.

AIM: To implement restoring division algorithm

Program Input:

```
import java.util.*;
class RESTORING
{
public static void lshift(int a[],int q[])
{
for(int i=0;i<3;i++)
{
a[i]=a[i+1];
}
a[3]=q[0];
for(int i=0;i<3;i++)
{
q[i]=q[i+1];
}
q[3]=0;
}
public static int[] add(int a[],int m1[])
{
int carry =0;
int sum[]=new int [4];
for(int i=3;i>=0;i--)
{
sum[i]=(a[i]+m1[i]+carry)%2;
carry=(a[i]+m1[i]+carry)/2;
}
return sum;
}
public static int [] comp2(int m1[])
```

AIM: To implement restoring division algorithm

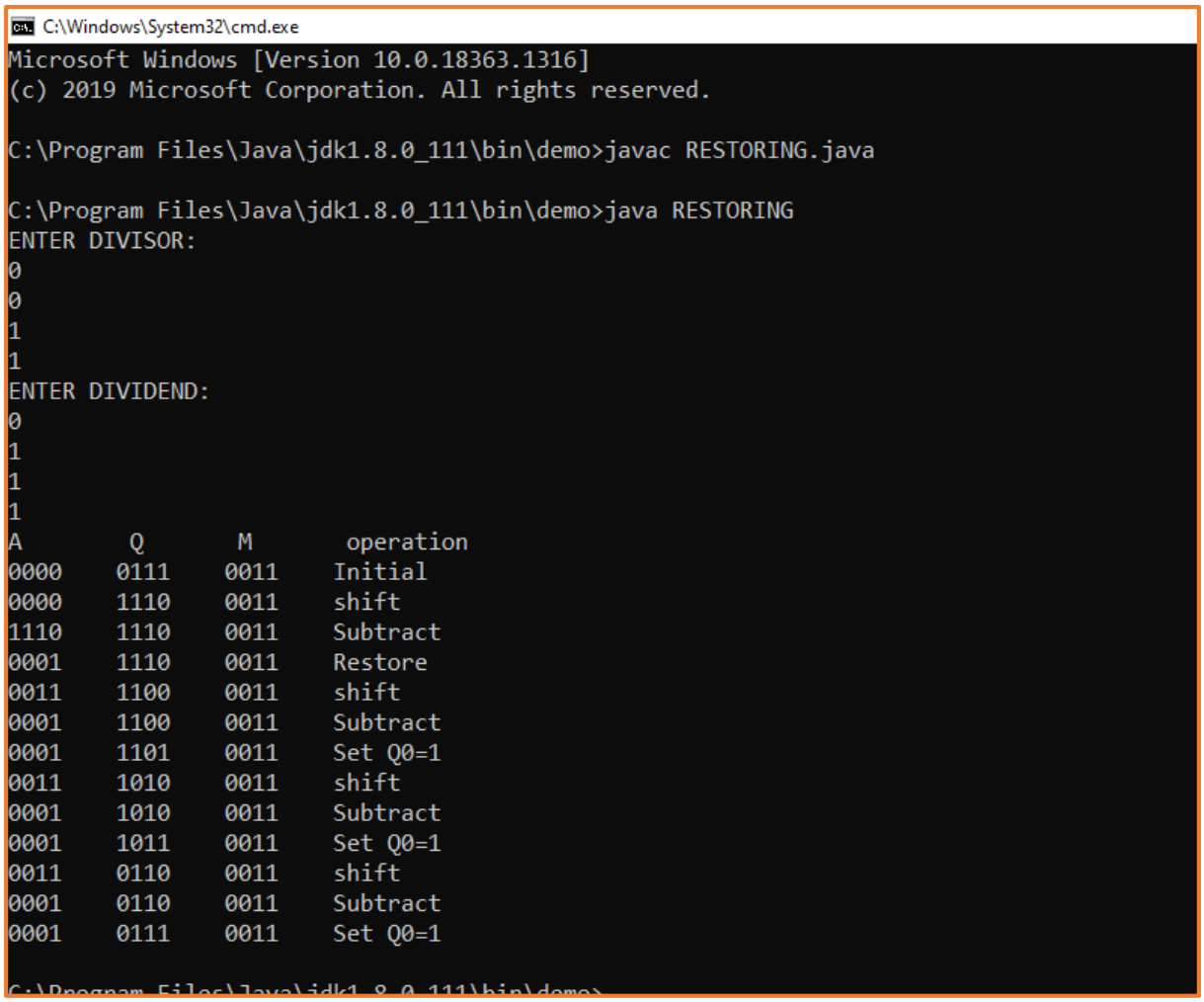
```
{
int z[]={0,0,0,1};
for(int i=0;i<3;i++)
{
if(m1[i]==0)
m1[i]=1;
else
m1[i]=0;
}
m1=add(m1,z);
return m1;
}
public static void display(int a[],int q[],int m[])
{
for(int i=0;i<4;i++)
{
System.out.print(a[i]);
}
System.out.print("\t");
for(int i=0;i<4;i++)
{
System.out.print(q[i]);
}
System.out.print("\t");
for(int i=0;i<4;i++)
{
System.out.print(m[i]);
}
System.out.print("\t");
```

```
}
public static void main(String args[])
{
Scanner sc=new Scanner(System.in);
int a[]={0,0,0,0};
int m[]=new int[4];
int q[]=new int[4];
int count=4;
int m1[]=new int[4];
System.out.println("ENTER DIVISOR:");
for(int i=0;i<=3;i++)
{
m[i]=sc.nextInt();
}
System.out.println("ENTER DIVIDEND:");
for(int i=0;i<=3;i++)
{
q[i]=sc.nextInt();
}
System.out.println("A \t Q \t M \t operation");
display(a,q,m);
System.out.print("Initial\n");
for(int i=count;i>0;i--)
{
for(int j=0;j<4;j++)
{
m1[j]=m[j];
}
lshift(a,q);
```

```
display(a,q,m);
System.out.print("shift\n");
int c[]=new int[4];
c=comp2(m1);
a=add(a,c);
display(a,q,m);
System.out.print("Subtract\n");
if(a[0]==1)
{
q[3]=0;
a=add(a,m);
display(a,q,m);
System.out.print("Restore\n");
}
else
{
q[3]=1;
display(a,q,m);
System.out.print("Set Q0=1\n");
}
}
}
}
```

AIM: To implement restoring division algorithm

Output:



● **Conclusion:** Hence, It is to be concluded that this presentation deals with the design **approach of restoring division algorithm**.