Vivekanand Education Society's
Institute of Technology
Approved by AICTE & Affiliated to University of Mumbai

COMPUTER ENGINEERING

DLCA ODD SEM 2021-22/EXPERIMENT 9

NAME:- GAURAV AMARNANI (D7A, 67)

EXPERIMENT NO 9:

AIM: Write a C program for non-restoring algorithm.

SOFTWARE USED: Turbo C IDE

THEORY:

Instead of the quotient digit set {0, 1}, the set {-1, 1} is used by the non- restoring division. The non-restoring division algorithm is more complex as compared to the restoring division algorithm. But when we implement this algorithm in hardware, it has an advantage, i.e., it contains only one decisionand addition/subtraction per quotient bit. After performing the subtraction operation, there will not be any restoring steps. Due to this, the numbers of operations basically cut down up to half. Because of the less operation, the execution of this algorithm will be fast. This algorithm basically performs simple operations such as addition, subtraction.

ALGORITHM:

Step 1: In this step, the corresponding value will be initialized to the registers, i.e., register A will contain value 0, register M will contain Divisor,register Q will contain Dividend, and N is used to specify the number of bits in dividend.

Step 2: In this step, we will check the sign bit of A.

Step 3: If this bit of register A is 1, then shift the value of AQ through left, and perform A = A + M. If this bit is 0, then shift the value of AQ into left and perform A = A - M. That means in case of 0, the 2's complement of M isadded into register A, and the result is stored into A.

Step 4: Now, we will check the sign bit of A again.

Step 5: If this bit of register A is 1, then Q[0] will become 0. If this bit is 0,then Q[0] will become 1. Here Q[0] indicates the least significant bit of Q.

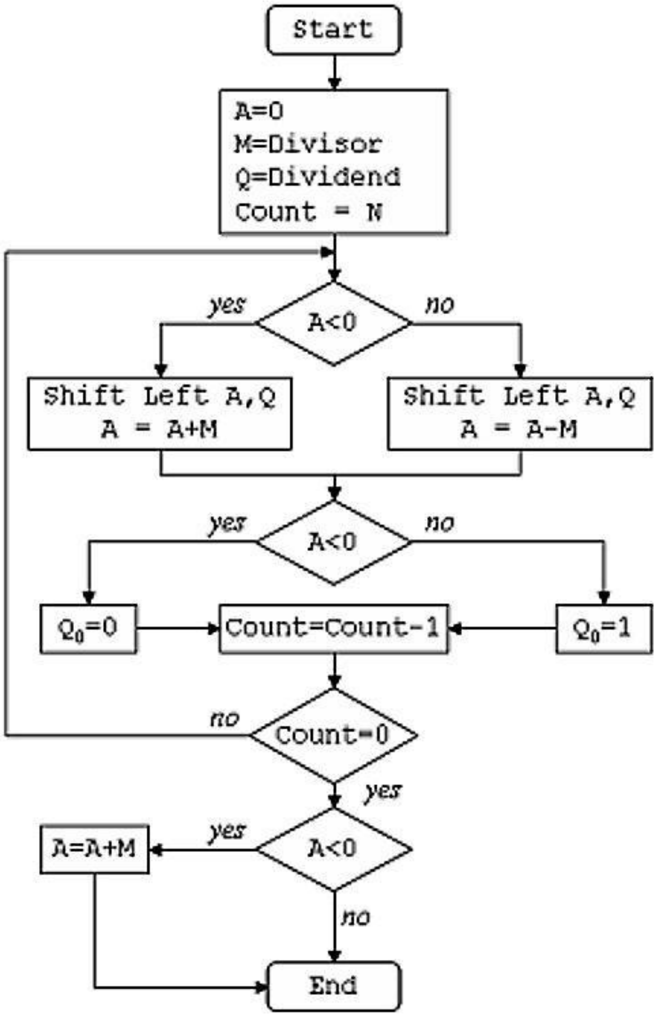
Step 6: After that, the value of N will be decremented. Here N is used as acounter.

Step 7: If the value of N = 0, then we will go to the next step. Otherwise, wehave to again go to step 2.

Step 8: We will perform A = A + M if the sign bit of register A is 1.

Step 9: This is the last step. In this step, register A contains the remainder,and register Q contains the quotient.

FLOWCHART:

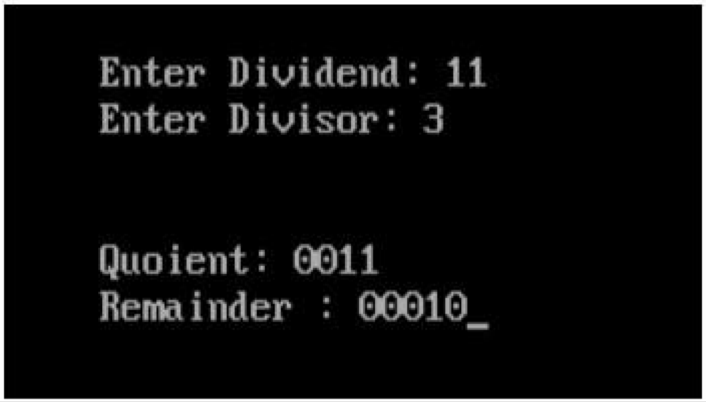


PROGRAM:

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
int acum[100]={0};
void add(int acum[],int b[],int n);
int q[100],b[100],l;
void main() {
int x,y;
int i = 0;
int n = i;
int bc[50];
int j;
clrscr();
printf("\nEnter Dividend: ");
scanf("%d",&x);
printf("Enter Divisor: ");
scanf("%d",&y);
while(x>0||y>0)
{
if(x>0)
{
q[i]=x%2;
x=x/2;
}
else
{
q[i]=0;
}
if(y>0)
{
b[i]=y%2;
y=y/2;
}
else
{
b[i]=0;
}
i++;
}
printf("\n");
for(i=0;i<n;i++)
{
if(b[i]==0)
{
bc[i]=1;
}
else
```

```
{
bc[i]=0;
}
}
bc[n]=1;
for(i=0;i<=n;i++)
{
if(bc[i]==0)
{
bc[i]=1;
i=n+2;
}
else
{
bc[i]=0;
}
}
b[n]=0;
for(i=n;i!=0;i--)
{
if(acum[n]==0)
{
for(j=n;j>0;j--)
{
acum[j]=acum[j-1]; }
acum[0]=q[n-1];
for(j=n-1;j>0;j--)
{
q[j]=q[j-1];
}
add(acum,bc,n+1);
}
Else
{
for(j=n;j>0;j--)
{
acum[j]=acum[j-1];
}
acum[0]=q[n-1];
for(j=n-1;j>0;j--)
{
q[j]=q[j-1];
}
add(acum,b,n+1);
}
if(acum[n]==1)
{
q[0]=0;
}
else
{ q[0]=1;
}
}
if(acum[n]==1)
{
add(acum,b,n+1);
}
printf("\nQuoient: ");
for( l=n-1;l>=0;l--)
{
printf("%d",q[l]);
}
printf("\nRemainder : ");
for( l=n;l>=0;l--)
{
printf("%d",acum[l]);
}
getch();
}
void add(int acum[],int bo[],int n) {
int i=0,temp=0,sum=0;
for(i=0;i<n;i++)
{
sum=0;
sum=acum[i]+bo[i]+temp;
if(sum==0)
{
acum[i]=0; temp=0;
}
else if(sum==2)
{
acum[i]=0; temp=1;
}
else if(sum==1)
{
acum[i]=1;
temp=0;
}
else if(sum==3)
{
acum[i]=1;
temp=1;
}
}
}
```

OUTPUT:



```
Enter Dividend: 11
Enter Divisor: 3


Quoient: 0011
Remainder : 00010_
```

CONCLUSION:

Implemented non-restoring algorithm using C program.