





Batch: A3 Roll No.:16010121051

Experiment / assignment / tutorial No.4

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

**TITLE:** To study and implement Non Restoring method of division

**AIM**: The basis of algorithm is based on paper and pencil approach and the operation involve repetitive shifting with addition and subtraction. So the main aim is to depict the usual process in the form of an algorithm.

Expected OUTCOME of Experiment: (Mention CO/CO's attained here)

\_\_\_\_\_

#### **Books/ Journals/ Websites referred:**

- **1.** Carl Hamacher, Zvonko Vranesic and Safwat Zaky, "Computer Organization", Fifth Edition, TataMcGraw-Hill.
- **2.** William Stallings, "Computer Organization and Architecture: Designing for Performance", Eighth Edition, Pearson.
- **3**. Dr. M. Usha, T. S. Srikanth, "Computer System Architecture and Organization", First Edition, Wiley-India.

### **Pre Lab/ Prior Concepts:**

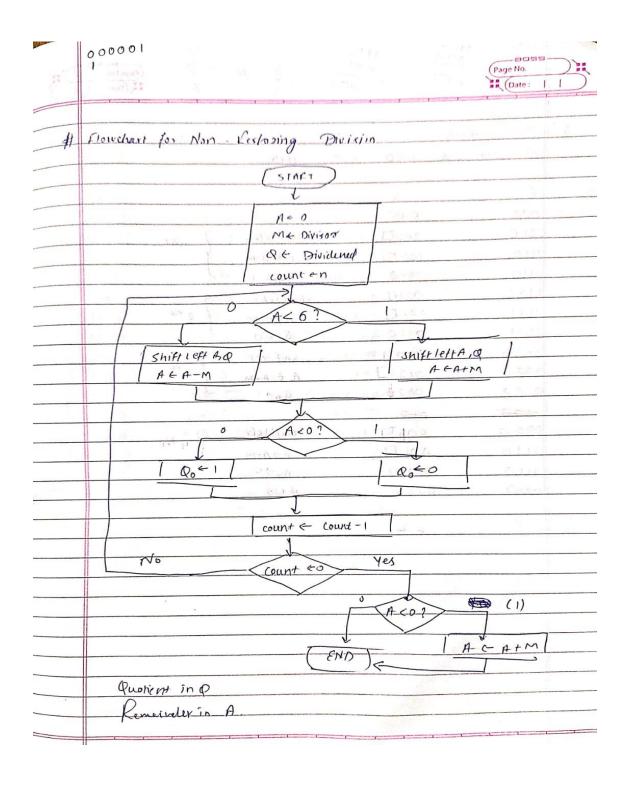
The Non Restoring algorithm works with any combination of positive and negative numbers.



(A Constituent College of Somaiya Vidyavihar University) **Department of Computer Engineering** 



### Flowchart for Non Restoring of Division:









## **Example:**

		110111 001100	001100	00010 110100 (Page No.	ss
-	- N		0,00001 -	110110 Date:	-
a.	M=12	N = 21			011'0
	W=001100	R= 26	11 -	001	
		0= 011010	- M = 110100	000	01
	A				
	600000	Ф			
	000000	011010	HASE	0	
	7.53	110101	shift left	1	
	110100	11010	n- +1	15+	-
	110100	110100	0000	15	
	101001	10100	Shift left (	1.1:01	11
	110101	101001	A+A+N1	2 nd	-
	110101	101000 60	(Po-) 0	1.64	
	101011	61 000 T		0.001	_
	11011	010001	A+M 4	3 rd	
	110111	010000	Q0-)000 J	solf colf	1
	101110	100000	swifle ft	(12 ×11)	
	111010	10000	7.	th	-
	111010	100000	A+M 64	2.0011	
	110101	00000	Sufflest 7	1181	
	000001	00000	A+M	545	-
	00001	000001	9031	Licen	
	000010	000017	snift up 1	2000	
	110110	000017		em	1
	110110	000010	Q6-) 0		
	020010	000010	A+ M		
				31.10	
	2	2		- L	
				11 1	
					-
_					-





(A Constituent College of Somaiya Vidyavihar University) **Department of Computer Engineering** 

	(	1101	3000	134		Page No.	
Q.	M=F					Date:	1
	,	9=5	M = 0	101			
		t i laye		4 - 010	-M	- 1011	
	A	9	Con Medica	0.1011/2 2			_
	0000						-
	0000	0101					
	1011	1010	0.0	shift let	10	Internal	
	1011	101 0	+3371 11 11	A JA-M	15+		
	R	1010	1-1-1-07-A	<b>?</b> ₀→ 0	9	10000	
	0111	01017	0 - 0 - 0/a	Snift left	0	- Actori	
	1100	0101	t pal limite		200	(5/7)	
	1100	0100	1 1-1 1 2 1-40	The Horizontal Control of the Contro	1	Ligitai	
	1000	1007	0 6,9	0, +0 Time	<u> </u>	1-101	
	1101	1007		shift 1017 101	7 3 00	1607	
	1101	1000	VH31-1FUIA	ATATM		41,5	
	1011	500 [	- Krintha	10,000		0.1011	
	0500		1	Swift left		57671	
		000 []	145 113.34	ATAM	/ 4th	Carri	
	000	0001	MINE IN	(00-31 CM	J	Legan	
_	0		1500	TO STORY		Norte	
8		# 1 P	gal if USA	TI OF	N.	Ofern	
	1 crases en		17-A1+A	- Flice		Juliant	
		X	0 402	0/00/		o trail	
	7	1	16-5-6-0	alloni		A WAS	
	CLASSIC.		The second	. A.A			
	4.5						
			3.5	-1.4			1
							h
	-						-
				and the second s			1







					_
R1)	M=2,	Q = 4	En College Park	Table of Carlo Hall	110
	M = 0010	8-0100 -	M= 1110		
			Y 161		
	A	Q	- V		
	0000	0100		7 15+	
	0000	1000	shift left.	4 - 13.1	
	1110	100 17	A e A-M		
	1110	1000	Q0€1		
	1101	0017	Shiff left	1 2 nd	
	1111	001 🗆	ACATM	-	1
	1//1	0010	Q0 +0		
	1110	0000	shif left	7 3 24	
	0000	001	AtAM	1	-
	0000	000 0	9071		
	0000	<del>010</del>	. <u> </u>		
	0000	0000	<i>withlest</i>	747	1 1
	1110	001 D	AIAM	<u> </u>	-
	1110	0010	$Q_{\sigma} \rightarrow 0$		
	000	0010	A+M	1	
	A 3.		<u> </u>		
	ی	£ 2	2000 7 Mario		1
				-	
		Los Y	as to tour be	0V;	
		en la			
	111 &		1		
		3. 1			
	765 E D =	2	¥.		
			MAR 1		
				i A	
				के पर अनुसरकान	
				161 e.j. 1911	







10000	√ ۲	16 100011	100000	001110	
		10	100111	161110	Page No.
18=	M= 32			2	Date:
		Q=59		18	011101
	W= 100000	Q= 111011			100000
	A		- W = 100	000	11110
		P	And the same of the latest and the same of		111011
	00000	111011		AND ADDRESS OF THE PARTY OF THE	100000
	000001	11011		de la companya del companya del companya de la comp	511011
	100001	The second secon	5.1	1 11	p 3 and 2 and
	100001	110117	A-M	4 "	
	11000	110110	0000	J	
	100011	101101	1.1	Jary	
	100011	10110 [	AtM	contractor	4 (1) 1
	000111	101100	0000	)	
	Loons	01100	5- 6	( 3 40	
	100111	01100D	A+M	1,	
	00 1110	011000	060	J	1 15.0
	101110	//0001]	<b>♣</b> 5.		EH.
3	101110	110001	The state of the s	71 4 4 14	Land of the second
- 1	01)101	110000	G0+0		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
41 12	211161	100001	5. L	) (1	
		100007	A+N1		
	111101	100000	60+0	J	
77	11101 12	00000	5. L	) . =	
	Ollott	00000 [	A+M	(674	
	011011	0 00001	9071	J	
		e o duti			
	27				
12					
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				
				HE STATE OF THE ST	
	1 時				
	131 +				
21				and a second second	







```
Implementation:
#include <math.h>
#include <stdio.h>
//NON RESTORING DIVISION
int main()
{
int a[50],a1[50],b[50],d=0,i,j;
 int n1,n2, c, k1,k2,n,k,quo=0,rem=0;
  printf("Enter the number of bits\n");
  scanf("%d",&n);
 printf("Enter the divisor and dividend\n");
 scanf("%d %d", &n1,&n2);
 for (c = n-1; c \ge 0; c--)//converting the 2 nos to binary
  k1 = n1 >> c;
  if (k1 & 1)
   a[n-1-c]=1;//M
  else
   a[n-1-c]=0;
   k2 = n2 >> c;
```







```
if (k2 & 1)
  b[2*n-1-c]=1;//Q
 else
 b[2*n-1-c]=0;
}
for(i=0;i<n;i++)//making complement
  if(a[i]==0)
   a1[i]=1;
  else
   a1[i]=0;
}
a1[n-1]+=1;//twos complement ie -M
if(a1[n-1]==2)
    for(i=n-1;i>0;i--)
  {
      if(a1[i]==2)
       a1[i-1]+=1;
```







```
a1[i]=0;
      }
 if(a1[0]==2)
  a1[0]=0;
 for( i=0;i<n;i++)// putting A in the same array as Q
   b[i]=0;
 }
printf("A\tQ\tPROCESS\n");
 for(i=0;i<2*n;i++)
  if(i==n)
    printf("\t");
  printf("%d",b[i]);
}
printf("\n");
```







```
for(k=0;k< n;k++)//n iterations
{
  for(j=0;j<2*n-1;j++)//left shift
    {
     b[j]=b[j+1];
    }
   for(i=0;i<2*n-1;i++)
    {
      if(i==n)
        printf("\t");
      printf("%d",b[i]);
    }printf("_");
   printf("\tLEFT SHIFT\n");
      if(b[0]==0)
      {
              for(i=n-1;i>=0;i--)//A=A-M
              {
                b[i]+=a1[i];
                   if(i!=0)
```





(A Constituent College of Somaiya Vidyavihar University) **Department of Computer Engineering** 

```
{
     if(b[i]==2)
             b[i-1]+=1;
            b[i]=0;
          }
     if(b[i]==3)
          {
            b[i-1]+=1;
            b[i]=1;
          }
         // printf("%d",b[i]);
   }
}
     if(b[0]==2)
       b[0]=0;
     if(b[0]==3)
       b[0]=1;
for(i=0;i<2*n-1;i++)
{
  if(i==n)
     printf("\t");
```







```
printf("%d",b[i]);
       }printf("_");
       printf("\tA-M\n");
}
else
{
       for(j=n-1;j>=0;j--)//A=A+M
          {
            b[j]+=a[j];
            if(j!=0)
         {
            if(b[j]==2)
                  {
                    b[j-1]+=1;
                    b[j]=0;
                  }
            if(b[j]==3)
                  {
```





(A Constituent College of Somaiya Vidyavihar University) **Department of Computer Engineering** 

```
b[j-1]+=1;
            b[j]=1;
          }
  }
     if(b[0]==2)
       b[0]=0;
     if(b[0]==3)
       b[0]=1;
  }
  for(i=0;i<2*n-1;i++)
{
  if(i==n)
     printf("\t");
  printf("%d",b[i]);
}printf("_");
printf("\tA+M\n");
```



}





```
if(b[0]==0)//A==0?
{
  b[2*n-1]=1;
  for(i=0;i<2*n;i++)
    {
      if(i==n)
         printf("\t");
      printf("%d",b[i]);
    }
    printf("tQ0=1\n");
}
if(b[0]==1)//A==1?
{
```







```
b[2*n-1]=0;
            for(i=0;i<2*n;i++)
              {
                if(i==n)
                  printf("\t");
                printf("%d",b[i]);
              }
             printf("\tQ0=0\n");
         }
}}
```



(A Constituent College of Somaiya Vidyavihar University) **Department of Computer Engineering** 



#### Output:

```
C:\Academics\SY\COA\nonrestoringdivison.exe
Enter the number of bits
Enter the divisor and dividend
15
               PROCESS
       Q
000000 001111
000000 01111_ LEFT SHIFT
111011 01111_
               A-M
111011 011110 Q0=0
110110 11110 LEFT SHIFT
111011 11110_
               A+M
111011 111100 Q0=0
110111 11100_ LEFT SHIFT
111100 11100
               A+M
111100 111000 Q0=0
111001 11000_
               LEFT SHIFT
111110 11000_
               A+M
111110 110000 Q0=0
111101 10000_ LEFT SHIFT
000010 10000
               A+M
000010 100001 Q0=1
000101
       00001_ LEFT SHIFT
000000
       00001_
               A-M
000000 000011 Q0=1
Process exited after 14.44 seconds with return value
```

#### **Conclusion:**

Non restoring divison was implemented successfully.







# **Post Lab Descriptive Questions**

Date: \_\_\_\_\_

1. What are the advantages of non restoring division over restoring division? The advantage of using non-restoring arithmetic over the standard restoring division is that a test subtraction is not required; the sign bit
determines whether an addition or subtraction is used.

Signature of faculty in-charge