

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



| Batch: A3       | Roll No.:16010121051         |
|-----------------|------------------------------|
| Experiment / a  | assignment / tutorial No     |
| Grade: AA / Al  | B / BB / BC / CC / CD /DD    |
| Signature of th | ne Staff In-charge with date |

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

**AIM**: The basis of algorithm is based on paper and pencil approach and the operation involves 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:**

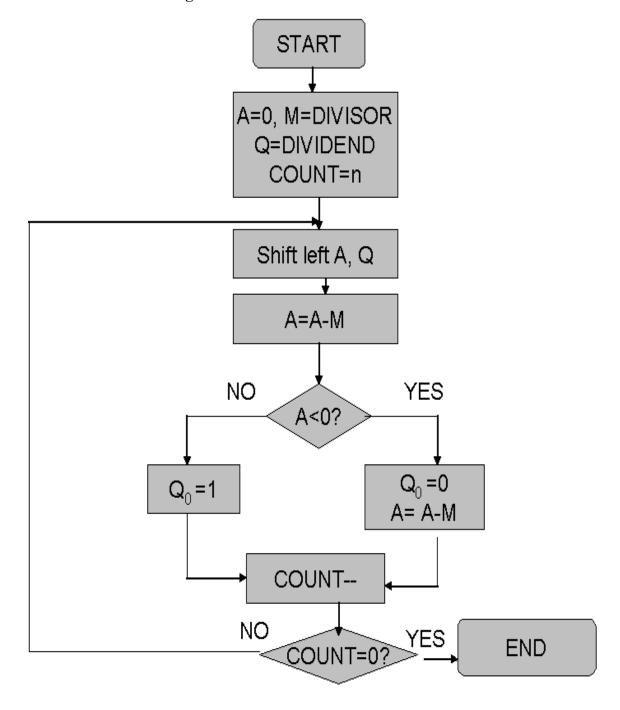
The Restoring algorithm works with any combination of positive and negative numbers.







## Flowchart for Restoring of Division:









## **Design Steps:**

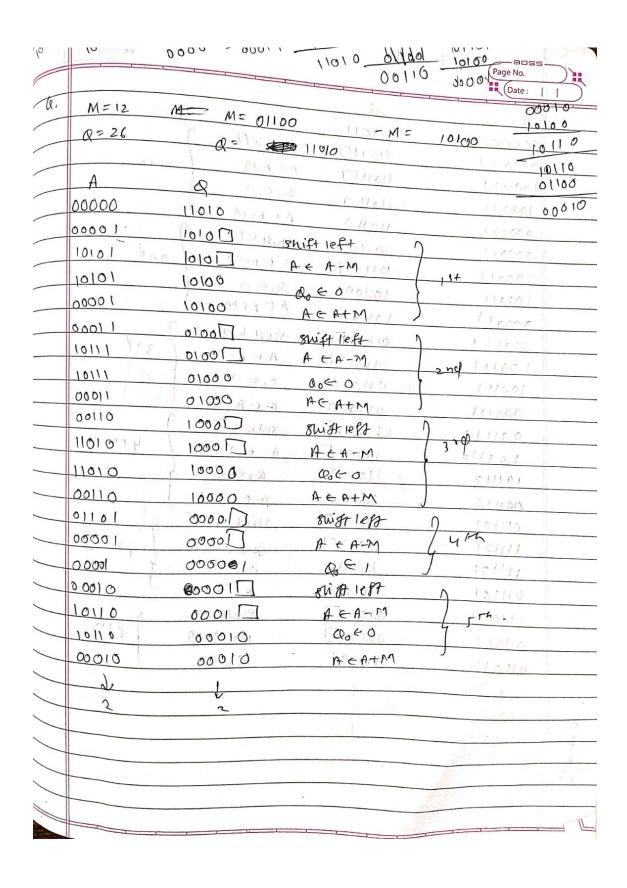
- 1. Start
- 2. Initialize A=0, M=Divisor, Q=Dividend and count=n (no of bits)
- 3. Left shift A, Q
- 4. If MSB of A and M are same
- 5. Then A=A-M
- 6. Else A=A+M
- 7. If MSB of previous A and present A are same
- 8.  $Q_0=0$  & store present A
- 9. Else  $Q_0=0$  & restore previous A
- 10. Decrement count.
- 11. If count=0 go to 11
- 12. Else go to 3
- 13. STOP

## Example:-















| A      | Q            |              | 10         |
|--------|--------------|--------------|------------|
| 000000 | 111011       | DOTTE THE SE | M = 1.3    |
| 000001 | 11011        | shift left   | 35 A       |
| 100001 | 11011        | A C A-M      | +          |
| 100001 | 110110       | 0,0€0        | A          |
| 400001 | 110110       | ACATM J      | A5 30 A    |
| 000011 | 101101       | shift refor  | Lonne      |
| 100011 | 10110        | A CA-M       | 2nd        |
| 100011 | 101100       | Q. E D 000   | 10101      |
| 00001  | 101100       | At At Moore  | 16000      |
| 000111 | 01100        | Shift lett 1 | Linas      |
| 100111 | 011001       | A-CA-M       | 3 80 11101 |
| 100111 | 011000       | 0,00000      | 11101      |
| 000111 | 011000       | AEA+M        | 100        |
| 001110 | 1100017      | swift lett o | 01100      |
| 101110 | 11000 1      | REA-MI       | 4 10 1011  |
| 101110 | 110000       | Q 6 6 0 00 C | 0.1011     |
| 001110 | (1600 0      | A-CA+M       | rura       |
| 011101 | 100001       | sufflet.     | 0          |
| 111101 | 100000       | ACA-M        | ۲۰۲۲ با    |
| 11/10) | 100000       | Q. C 0       | y the      |
| 011101 | 100000       | ACA+M.       | C full of  |
| 111011 | 00000        | Shift left   | <b>n</b>   |
| 011011 | 00000        | M-C-A-M      | C Th       |
| 01/0/1 | 00000        | 0,00         | 5 616.00   |
|        |              |              | + f)       |
| 4      | $\checkmark$ | <i>¥</i>     | Y.         |
| 22     |              |              |            |
|        |              |              |            |
|        |              |              |            |
|        |              |              |            |
|        |              |              |            |







| 12/1  | 1110       | 0001             | 001 1101 11    | 6 G Page No.    |
|-------|------------|------------------|----------------|-----------------|
| 0101  | 00 000     | 1101             | 1110 6660      | 60 Date: 1 1    |
| 00    | 80         |                  |                |                 |
| - EJ. | M = 3      | M= 0011          | -M=1101        | 27 7 1 1/4 1 27 |
| -     | R = 7      | 0=011            | 11 (1)         | A The second    |
|       | 1          |                  |                |                 |
|       | A          | 8                |                | 11              |
|       | 0000       | 0111             | /8/16          | 7 0000          |
|       | 0000       | (11111 + 12) +3: |                | First aycle.    |
|       | 0101 +2    | 1111 M-9-31      |                | 1/2/1           |
|       |            |                  | Q. E 0 / C     | 1500            |
|       | 1101       | M10 MAR 201      |                | lecol           |
|       | 0000       | 1/10 (M-11-) A   | shift left )   | 3 a / F         |
|       | 0001       |                  | A & A-MIO      | and eyele.      |
|       | 1110       | 1100 MIA 3 A     | 0000           | 1000            |
|       | 5001       | 1100 50 117/112  | ACATMO         | 0.100           |
|       | 0016       |                  | swiftleft )    | This d my c/e.  |
|       | 6000       | 100 1 00         | ACA-MOI 9      | THE             |
|       | 5000       | 100 11164 34     | 0071001        | c/ cm           |
|       | 0001       | 00 1/ A 1/12     | shift left     | 1010            |
|       | 1110 - 110 | 001 -A - A       | ACA-NO         | HM cycle.       |
|       | 1110       | 0010             | Q.E.O          | CORRE           |
|       | 000 /      | 0010             | ACA+M          | <u>.</u>        |
|       | 1          | <b>→</b>         | * * *          | *               |
|       | A          | 2                |                |                 |
|       | #          |                  |                |                 |
|       |            |                  |                |                 |
|       |            |                  |                |                 |
|       |            |                  | 15             |                 |
|       |            |                  | and the second |                 |
|       |            |                  | 1              |                 |
|       |            |                  | A.B.           |                 |







| 60 000 000 | 0000 1000 10       | 101 0,010         | Date:                 |
|------------|--------------------|-------------------|-----------------------|
| 0 0        |                    |                   |                       |
| a. M= r m= | 0101 1010 - M = 10 |                   |                       |
|            | 0101               | 111000 -11.       | 3 1                   |
| a di       | 0107               | - 1 1 px y        | 7 7                   |
| A          | d                  |                   |                       |
| 0000       | 0101               |                   |                       |
| 0000       |                    |                   |                       |
| 1011       | 101 Shift          |                   | C, 1 1 1 1            |
| 1011       | 101                |                   | s <del>+</del> 1616 1 |
| 0000       |                    | 60                |                       |
| 0001       |                    | CA+M              | , for part of         |
| 1106       |                    | Friett 1          | 1000                  |
| 1100       |                    | CA-M              | n e)                  |
|            | 0100               | 601               | 5111                  |
| 1000       |                    | CA+M DOIL         | 0.171                 |
| 0010       | 1001               | wiftlest on       | i.c.                  |
| 1101       | 1601 A             | E A-M 37          |                       |
| 11011      | 1000               | 2,60              | C.C. (d)              |
| Ø516       |                    | ACAFM             | 0000                  |
| 0101       | 000                | shift left.       | V C and               |
| 00 00      | 000                | A CA-MICAL        | you.                  |
| 9000       | 0001               | Q E 1 1 1 1 1 1 1 | 6111                  |
|            | 1 +1 -2 -4         | 6 15              | 1 200                 |
|            | <b>-</b>           | 4                 |                       |
|            |                    | 1                 | *                     |
|            | 4                  |                   |                       |
|            |                    |                   |                       |
|            | 7                  |                   |                       |
|            |                    |                   |                       |
|            |                    |                   |                       |
|            |                    |                   |                       |
|            |                    |                   |                       |
|            |                    |                   |                       |



# **K. J. Somaiya College of Engineering, Mumbai-77** (A Constituent College of Somaiya Vidyavihar University)



**Department of Computer Engineering** 

| Implementation:                                                                          |
|------------------------------------------------------------------------------------------|
| /*Program for Restoring Division.*/                                                      |
| #include <stdio.h></stdio.h>                                                             |
| #include <conio.h></conio.h>                                                             |
| #include <math.h></math.h>                                                               |
|                                                                                          |
| int $a=0,b=0,c=0,com[5]=\{1,0,0,0,0\},s=0;$                                              |
| int anum $[5]=\{0\}$ ,anumcp $[5]=\{0\}$ ,bnum $[5]=\{0\}$ ;                             |
| $\underline{int\ acomp[5]=\{0\},bcomp[5]=\{0\},rem[5]=\{0\},quo[5]=\{0\},res[5]=\{0\};}$ |
|                                                                                          |
| <pre>void binary(){</pre>                                                                |
| $\underline{a = fabs(a)}$ ;                                                              |
| b = fabs(b);                                                                             |
| <u>int r, r2, i, temp;</u>                                                               |
| $for(i = 0; i < 5; i++){$                                                                |
| r = a % 2;                                                                               |
| a = a / 2;                                                                               |
| r2 = b % 2;                                                                              |
| b = b / 2;                                                                               |
| <u>anum[i] = r;</u>                                                                      |

anumcp[i] = r;







| $\underline{\qquad \qquad bnum[i] = r2;}$                       |
|-----------------------------------------------------------------|
| $if(r2 == 0)\{$                                                 |
| bcomp[i] = 1;                                                   |
| }                                                               |
| $if(r == 0)\{$                                                  |
| acomp[i] =1;                                                    |
| }                                                               |
| }                                                               |
| //part for two's complementing                                  |
| c = 0;                                                          |
| for $(i = 0; i < 5; i++)$                                       |
| res[i] = com[i] + bcomp[i] + c;                                 |
| <u>if(res[i]&gt;=2){</u>                                        |
| c = 1;                                                          |
| }                                                               |
| else                                                            |
| $\underline{\qquad \qquad c=0;}$                                |
| $\underline{\operatorname{res}[i] = \operatorname{res}[i]\%2;}$ |
| _}                                                              |
| $for(i = 4; i >= 0; i){$                                        |
| bcomp[i] = res[i];                                              |







| _}                                 |
|------------------------------------|
| 1                                  |
| void add(int num[]){               |
| <u>int i;</u>                      |
| c = 0;                             |
| for( $i = 0$ ; $i < 5$ ; $i++$ ){  |
| res[i] = rem[i] + num[i] + c;      |
| $if(res[i] \ge 2){$                |
| <u>c = 1;</u>                      |
| }                                  |
| <u>else</u>                        |
| c = 0;                             |
| res[i] = res[i]%2;                 |
| }                                  |
| $for(i = 4; i \ge 0; i){$          |
| <u>rem[i] = res[i];</u>            |
| <pre>printf("%d",rem[i]);</pre>    |
| }                                  |
| printf(":");                       |
| $for(i = 4; i >= 0; i){$           |
| <pre>printf("%d",anumcp[i]);</pre> |







| }                                                               |
|-----------------------------------------------------------------|
| 1                                                               |
| <pre>void shl(){//for shift left</pre>                          |
| int i;                                                          |
| $for(i = 4; i > 0; i){//shift the remainder}$                   |
| $\underline{\operatorname{rem}[i] = \operatorname{rem}[i-1];}$  |
| }                                                               |
| $\underline{\operatorname{rem}[0] = \operatorname{anumcp}[4];}$ |
| $for(i = 4; i > 0; i){//shift the remtient}$                    |
| <pre>anumcp[i] = anumcp[i-1];</pre>                             |
| }                                                               |
| $\underline{\text{anumcp}[0] = 0;}$                             |
| <pre>printf("\nSHIFT LEFT: ");//display together</pre>          |
| for( $i = 4$ ; $i >= 0$ ; $i)$ {                                |
| <pre>printf("%d",rem[i]);</pre>                                 |
| }                                                               |
| <u>printf(":");</u>                                             |
| $for(i = 4; i >= 0; i){$                                        |
| <pre>printf("%d",anumcp[i]);</pre>                              |
| }                                                               |
| 1                                                               |







| <pre>int main(){</pre>                                  |
|---------------------------------------------------------|
| <del>_</del> ;                                          |
| int i;                                                  |
| <pre>printf("\t\tRESTORING DIVISION ALGORITHM");</pre>  |
| <pre>printf("\nEnter two numbers to multiply: ");</pre> |
| <pre>printf("\nBoth must be less than 16");</pre>       |
| //simulating for two numbers each below 16              |
| do{_                                                    |
| <pre>printf("\nEnter Dividened: ");</pre>               |
| scanf("%d",&a);                                         |
| <pre>printf("Enter Divisor: ");</pre>                   |
| scanf("%d",&b);                                         |
| }while(a>=16    b>=16);                                 |
|                                                         |
| <pre>printf("\nExpected Quotient = %d", a/b);</pre>     |
| <pre>printf("\nExpected Remainder = %d", a%b);</pre>    |
| $if(a*b < 0){$                                          |
| <u>s = 1;</u>                                           |
| }                                                       |







| binary();                                                   |
|-------------------------------------------------------------|
| <pre>printf("\n\nUnsigned Binary Equivalents are: ");</pre> |
| <u>printf("\nA = ");</u>                                    |
| $for(i = 4; i >= 0; i){$                                    |
| <pre>printf("%d",anum[i]);</pre>                            |
| }                                                           |
| <pre>printf("\nB = ");</pre>                                |
| $for(i = 4; i >= 0; i){$                                    |
| <pre>printf("%d",bnum[i]);</pre>                            |
| }                                                           |
| <pre>printf("\nB'+ 1 = ");</pre>                            |
| $for(i = 4; i >= 0; i){$                                    |
| <pre>printf("%d",bcomp[i]);</pre>                           |
| _}                                                          |
| printf("\n\n>");                                            |
| //division part                                             |
| <u>shl();</u>                                               |
| $for(i=0;i<5;i++){}$                                        |
| <pre>printf("\n&gt;"); //start with subtraction</pre>       |
| <pre>printf("\nSUB B: ");</pre>                             |
| add(bcomp);                                                 |







| if(rem[4]==1){//simply add for restoring          |
|---------------------------------------------------|
| <pre>printf("\n&gt;RESTORE");</pre>               |
| <pre>printf("\nADD B: ");</pre>                   |
| $\underline{\qquad \qquad \text{anumcp}[0] = 0;}$ |
| add(bnum);                                        |
| }                                                 |
| else{                                             |
| $\underline{\qquad} anumcp[0] = 1;$               |
| }                                                 |
| if(i<4)                                           |
| shl();                                            |
|                                                   |
| }                                                 |
| printf("\n");                                     |
| <pre>printf("\nSign of the result = %d",s);</pre> |
| <pre>printf("\nRemainder is = ");</pre>           |
| $for(i = 4; i \ge 0; i){$                         |
| <pre>printf("%d",rem[i]);</pre>                   |
| }                                                 |
| <pre>printf("\nQuotient is = ");</pre>            |
| for(i = 4; i >= 0; i)                             |



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



| <pre>printf("%d",anumcp[i]);</pre> |  |
|------------------------------------|--|
| }                                  |  |
| getch();                           |  |
|                                    |  |
|                                    |  |

## Output:

}

## C:\Academics\SY\COA\restoringdevision.exe

```
RESTORING DIVISION ALGORITHM
Enter two numbers to multiply:
Both must be less than 16
Enter Dividened: 10
Enter Divisor: 3
Expected Quotient = 3
Expected Remainder = 1
Unsigned Binary Equivalents are:
Onsigned Bind
A = 01010
B = 00011
B'+ 1 = 11101
SHIFT LEFT: 00000:10100
SUB B: 11101:10100
SUB B: 11101:10100
-->RESTORE
ADD B: 00000:10100
SHIFT LEFT: 00001:01000
SUB B: 11110:01000
-->RESTORE
ADD B: 00001:01000
SHIFT LEFT: 00010:10000
SUB B: 11111:10000
-->RESTORE
ADD B: 00010:10000
SHIFT LEFT: 00101:00000
SUB B: 00010:00000
SHIFT LEFT: 00100:00010
SUB B: 00001:00010
Sign of the result = 0
Remainder is = 00001
Quotient is = 00011_
```



# **K. J. Somaiya College of Engineering, Mumbai-77** (A Constituent College of Somaiya Vidyavihar University)





| Conclusion:                                                                                                                                                                                                                                                                                                                  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Booths restoring division was implemented successfully.                                                                                                                                                                                                                                                                      |
|                                                                                                                                                                                                                                                                                                                              |
|                                                                                                                                                                                                                                                                                                                              |
|                                                                                                                                                                                                                                                                                                                              |
|                                                                                                                                                                                                                                                                                                                              |
| Post Lab Descriptive Questions                                                                                                                                                                                                                                                                                               |
| 1. What are the advantages of restoring division over non 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. The disadvantage, though, is that an extra bit must be maintained in the partial remainder to keep track of the sign. |
|                                                                                                                                                                                                                                                                                                                              |
|                                                                                                                                                                                                                                                                                                                              |
|                                                                                                                                                                                                                                                                                                                              |
| Date: Signature of faculty in-charge                                                                                                                                                                                                                                                                                         |



# **K. J. Somaiya College of Engineering, Mumbai-77** (A Constituent College of Somaiya Vidyavihar University)



Department of Computer Engineering