Vidyavardhini's College of Engineering and Technology
Department of Artificial Intelligence & Data Science

Experiment No. 8
Implement Restoring algorithm using c-programming
Name: Patil Pranjal Keshav
Roll Number: 45
Date of Performance:
Date of Submission:

Aim: To implement Restoring division algorithm using c-programming.

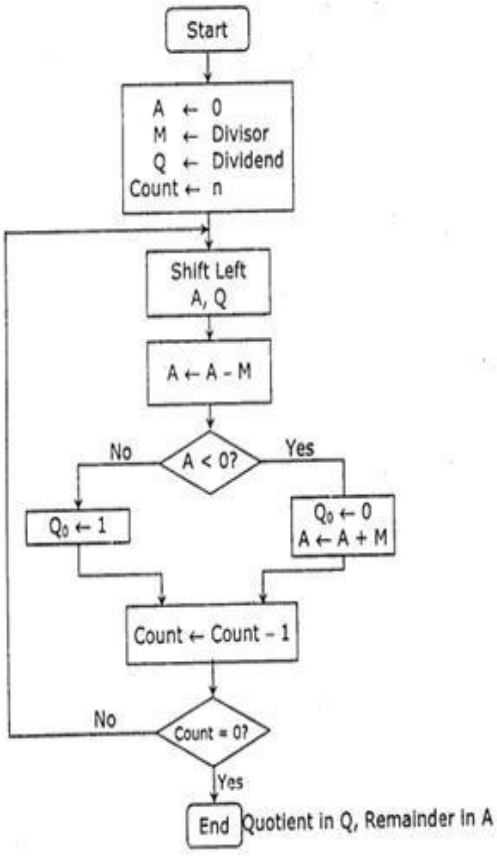
Objective -

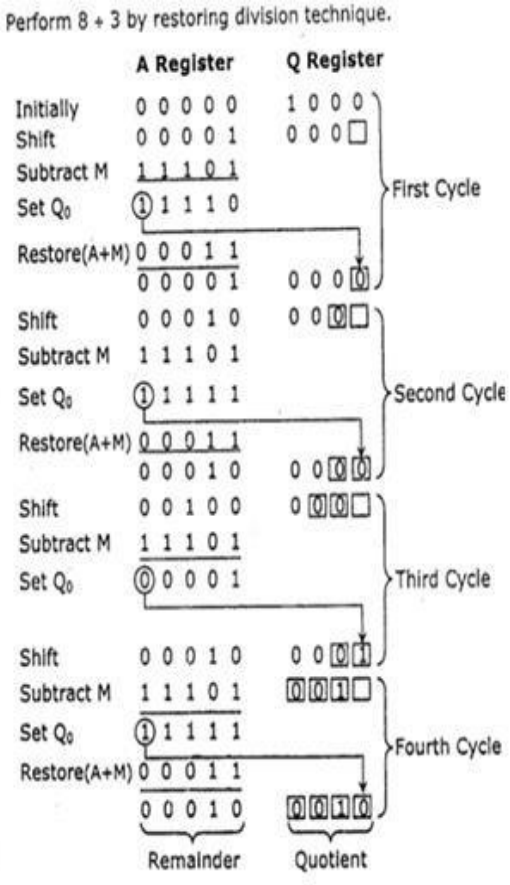
1. To understand the working of Restoring division algorithm.
2. To understand how to implement Restoring division algorithm using c-programming.

Theory:

1) The divisor is placed in M register, the dividend placed in Q register.
2) At every step, the A and Q registers together are shifted to the left by 1-bit
3) M is subtracted from A to determine whether A divides the partial remainder. If it does, then Q0 set to 1-bit. Otherwise, Q0 gets a 0 bit and M must be added back to A to restore the previous value.
4) The count is then decremented and the process continues for n steps. At the end, the quotient is in the Q register and the remainder is in the A register.

Flowchart





Program-

```
#include <stdio.h>

void binaryPrint(int num, int bits) {
   for (int i = bits - 1; i >= 0; i--) {
      printf("%d", (num >> i) & 1);
   }
}

void restoringDivision(int dividend, int divisor) {
   int quotient = 0;
   int remainder = 0;
   int bits = 5; // Adjust bits for binary representation (5 bits for max dividend 31)

   // Shift left the quotient and remainder for the number of bits in dividend
   for (int i = bits - 1; i >= 0; i--) {
      // Left shift the remainder and add the next bit of the dividend
      remainder = (remainder << 1) | ((dividend >> i) & 1);
      quotient <<= 1; // Shift left quotient

      // Subtract divisor from remainder
      if (remainder >= divisor) {
         remainder -= divisor;
         quotient |= 1; // Set the least significant bit of the quotient
      }
   }

   // Print the results
   printf("Quotient: ");
   binaryPrint(quotient, bits);
   printf(" Remainder: ");
   binaryPrint(remainder, bits);
   printf("\n");
}

int main() {
   int dividend, divisor;

   // Input from user
   printf("Enter the Number (dividend and divisor separated by space): ");
```

```
scanf("%d %d", &dividend, &divisor);

    restoringDivision(dividend, divisor);

    return 0;
}
```

Output -

```
Enter the Number (dividend and divisor separated by space): 15 7
Quotient: 00010 Remainder: 00001
```

Conclusion -

The Restoring Division algorithm is an effective method for performing division in binary arithmetic. It simulates the long division process, repeatedly subtracting the divisor from a portion of the dividend while managing the remainder.