Q1) Write a program to evaluate the written statement:

$$X = A + B/C * (D + E) - F$$

a. Using memory type computer with three address instructions.

b. Using a general register computer with two address instructions.

c. Using an accumulator type computer with one address instructions.

Sol. →

$$X = A + B/C * (D + E) - F$$

(a) Three Address instruction

DIV $R_1, B, C$ = $R_1 \leftarrow M[B] / M[C]$

Add $R_2, D, E$ = $R_2 \leftarrow M[D] + M[E]$

MUL $R_1, R_1, R_2$ = $R_1 \leftarrow R_1 * R_2$

Add $R_1, R_1, A$ = $R_1 \leftarrow R_1 + M[A]$

Sub $X, R_1, F$ = $M[X] \leftarrow R_1 - M[F]$

(b) Two Address instruction

Mov $R_1, B$ = $R_1 \leftarrow M[B]$

DIV $R_1, C$ = $R_1 \leftarrow R_1 / M[C]$

MOV $R_2, D$ = $R_2 \leftarrow M[D]$

Add $R_2, E$ = $R_2 \leftarrow R_2 + M[E]$

Mul $R_1, R_2$ = $R_1 \leftarrow R_1 * R_2$

Add $R_1, A$ = $R_1 \leftarrow R_1 + M[A]$

Sub $R_1, F$ = $R_1 \leftarrow R_1 - M[F]$

Mov $X, R_1$ = $M[X] \leftarrow R_1$

(c) One address instruction

Load B = $AC \leftarrow M[B]$

DIV C = $AC \leftarrow AC / M[C]$

STORE T = $M[T] \leftarrow AC$

LOAD D = $AC \leftarrow M[D]$

Add E = $AC \leftarrow AC + M[E]$

MUL T = $AC \leftarrow AC * M[T]$

Add A = $AC \leftarrow AC + M[A]$

Sub F = $AC \leftarrow AC - M[F]$

STORE X = $M[X] \leftarrow AC$

Q5 Describe all methods of obtaining 2's complement of a given number. Express −39 in 8-bit 2's complement form.

Sol:-

Methods of obtaining the 2's complement of a number:

The 2's complement of a number can be obtained in three ways as given below:-

1. By obtaining the 1's complement of the given number (by changing all 0's to 1's and vice versa) and then adding 1.

2. By subtracting the given n-bit numbers N from $2^n$.

3. Starting at the LSB, copying down each bit up to and including the first 1 bit encountered, and complementing remaining bits.

Now;

2's complement of 39 = −39.

∴ 39 = 00100111

1's complement = 11011000

(9-7-07)

Adding 1:-

$-39 = \underline{11011001}$ = 2's complement of 39

Also;

$2^8 - 39$

$= \begin{array}{r} 100000000 \\ 00100111 \\ \hline 11011001 \end{array} = -39$

$39 = 00100111$

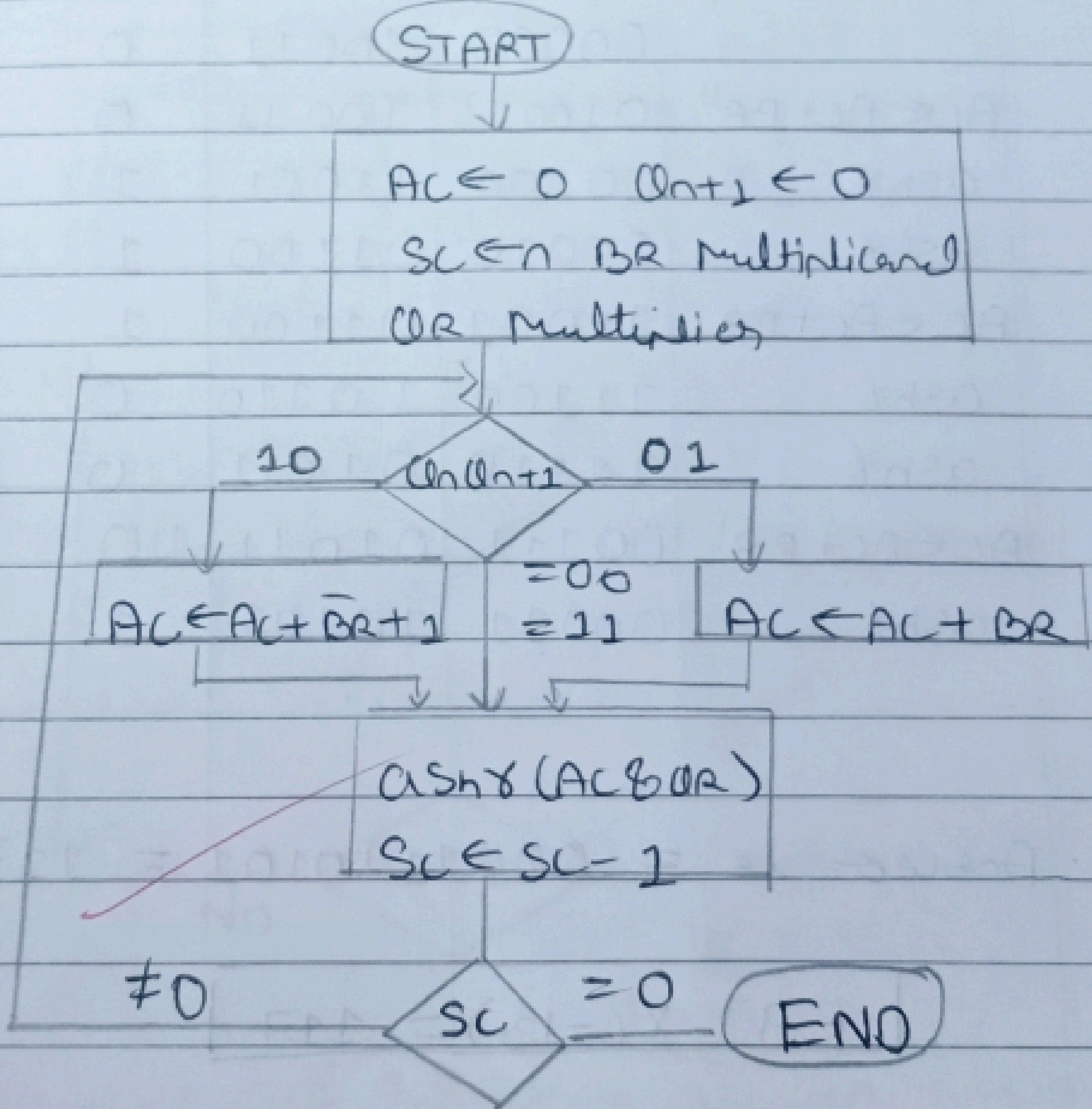
← (third method

∴ 2's complement $= \underline{\underline{11011001}}$

Q.3 Design the flowchart for signed binary multiplication or Booth algorithm using 2's complement numbers.

(-9) x (-13)

Solution:- The flowchart for signed binary multiplication or Booth algorithm is -



Multiplication is –

BR = (−9) , Binary (9) = 01001
2's complement = 10111

QR = (−13) , Binary(13) = 01101
2's Complement = 10011

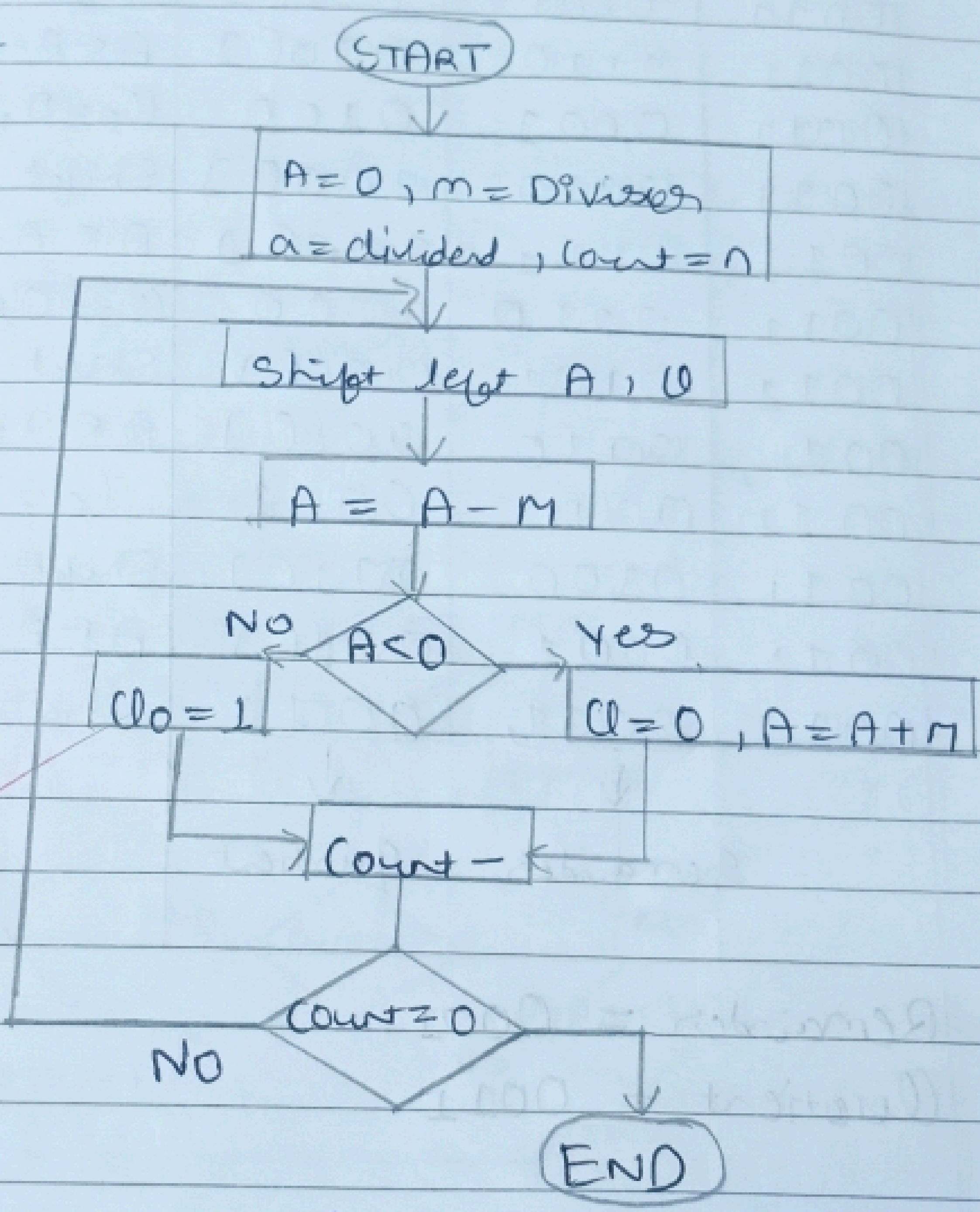
| Operation | AC | $Q_n$ | $Q_{n+1}$ | SC |
|---|---|---|---|---|
| | 00000 | 10011 | 0 | 5 |
| AC←AC+BR'' | 01001 | 10011 | 0 | 5 |
| ashr | 00100 | 11001 | 1 | 4 |
| ashr | 00010 | 01100 | 1 | 3 |
| AC←AC+BR | 11001 | 01100 | 1 | 3 |
| ashr | 11100 | 10110 | 0 | 2 |
| ashr | 11110 | 01011 | 0 | 1 |
| AC←AC+BR'' | 00111 | 01011 | 0 | 1 |
| ashr | 00011 | 10101 | 1 | 0 |

Answer is = 0001110101 = 117

$\boxed{(-9) \times (-13) = 117}$

Q. 4 Draw neat flowchart for restoring division method with the evaluation —
Dividend = 1010 Divisor = 0011
Find remainder and quotient ?

Solution:-




Flowchart for restoring division algorithm

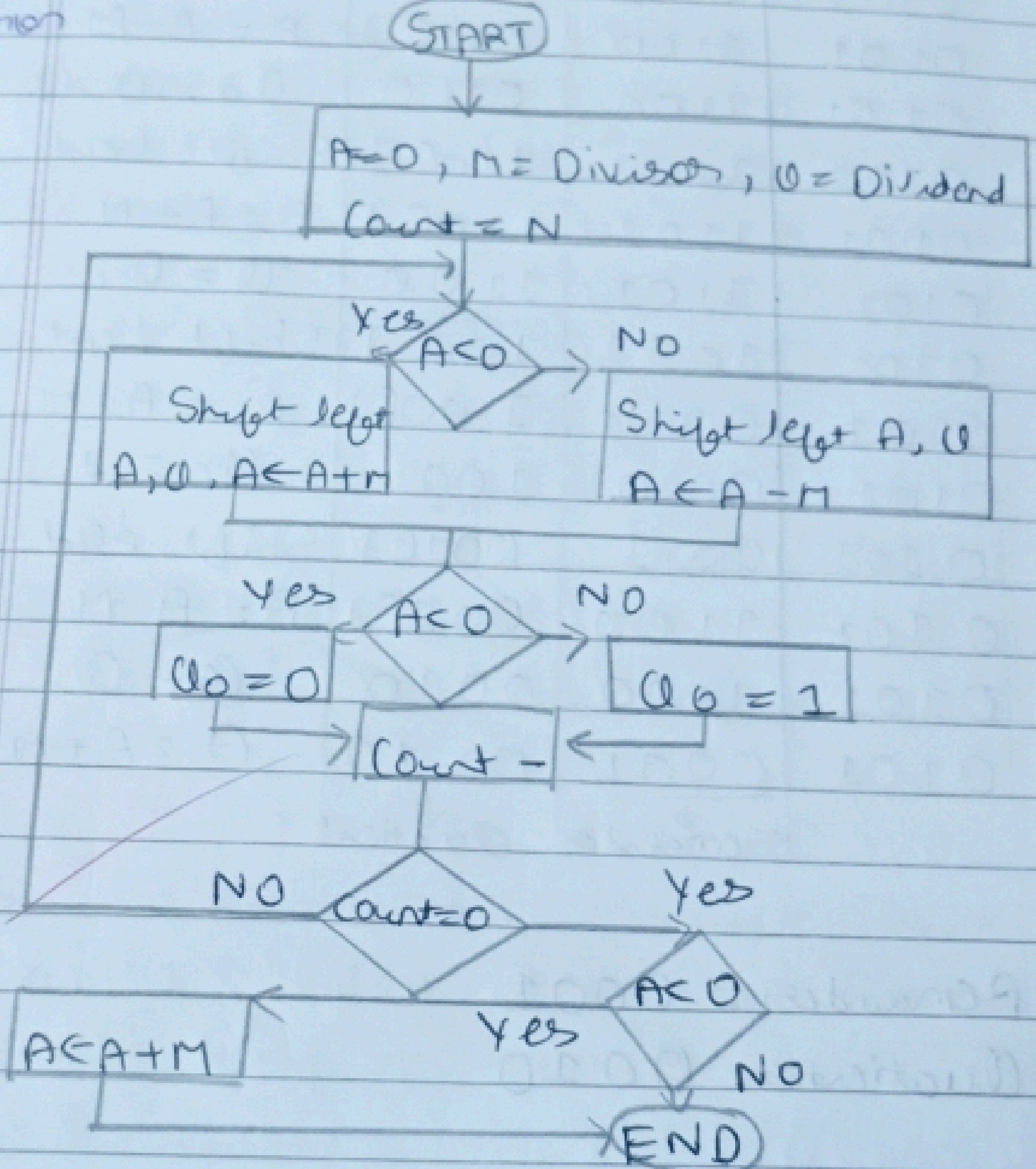
Here, M = 0011 , M'' = 1101
Q = 1010

| Count | M | A | Q | Operation |
|---|---|---|---|---|
| 4 | 0011 | 0000 | 1010 | |
| | 0011 | 0001 | 010[ ] | Shift left |
| | 0011 | 1110 | 010[ ] | A←A-M |
| 3 | 0011 | 0001 | 0100 | $Q_0=0$, A←A+M |
| | 0011 | 0010 | 100[ ] | Shift left |
| | 0011 | 1111 | 100[ ] | A←A-M |
| 2 | 0011 | 0010 | 1000 | $Q_0=0$, A←A+M |
| | 0011 | 0101 | 000[ ] | Shift left |
| | 0011 | 0010 | 000[ ] | A←A-M |
| 1 | 0011 | 0010 | 0001 | $Q_0=1$ |
| | 0011 | 0100 | 001[ ] | Shift left |
| | 0011 | 0001 | 001[ ] | A←A-M |
| 0 | 0011 | 0001 | 0011 | $Q_0=1$ |
| | | ↓ remainder | ↓ quotient | |

Remainder = 0001
Quotient = 0011

Q.5 Draw neat flowchart for non-restoring division method with the evaluation -
Dividend = 1011 Divisor = 0101
Find remainder and quotient?

Solution




Flowchart for non-restoring division algorithm

Here, M = 0101 , M'' = 1011
Q = 1011

| Count | M | A | Q | Operation |
|---|---|---|---|---|
| 4 | 0101 | 0000 | 1011 | |
| | 0101 | 0001 | 011[ ] | Shift Left |
| | 0101 | 1100 | 011[ ] | A = A-M |
| 3 | 0101 | 1100 | 0110 | $Q_0 = 0$ |
| | 0101 | 1000 | 110[ ] | Shift Left |
| | 0101 | 1101 | 110[ ] | A = A+M |
| 2 | 0101 | 1101 | 1100 | Q = 0 |
| | 0101 | 1011 | 100[ ] | Shift Left |
| | 0101 | 0000 | 100[ ] | A = A+M |
| 1 | 0101 | 0000 | 1001 | $Q_0 = 1$ |
| | 0101 | 0001 | 001[ ] | Shift left |
| | 0101 | 1100 | 001[ ] | A = A-M |
| 0 | 0101 | 1100 | 0010 | $Q_0 = 0$ |
| | 0101 | 0001 | 0010 | A = A+M |
| | | ↓ remainder | ↓ Quotient | |

Remainder = 0001
Quotient = 0010