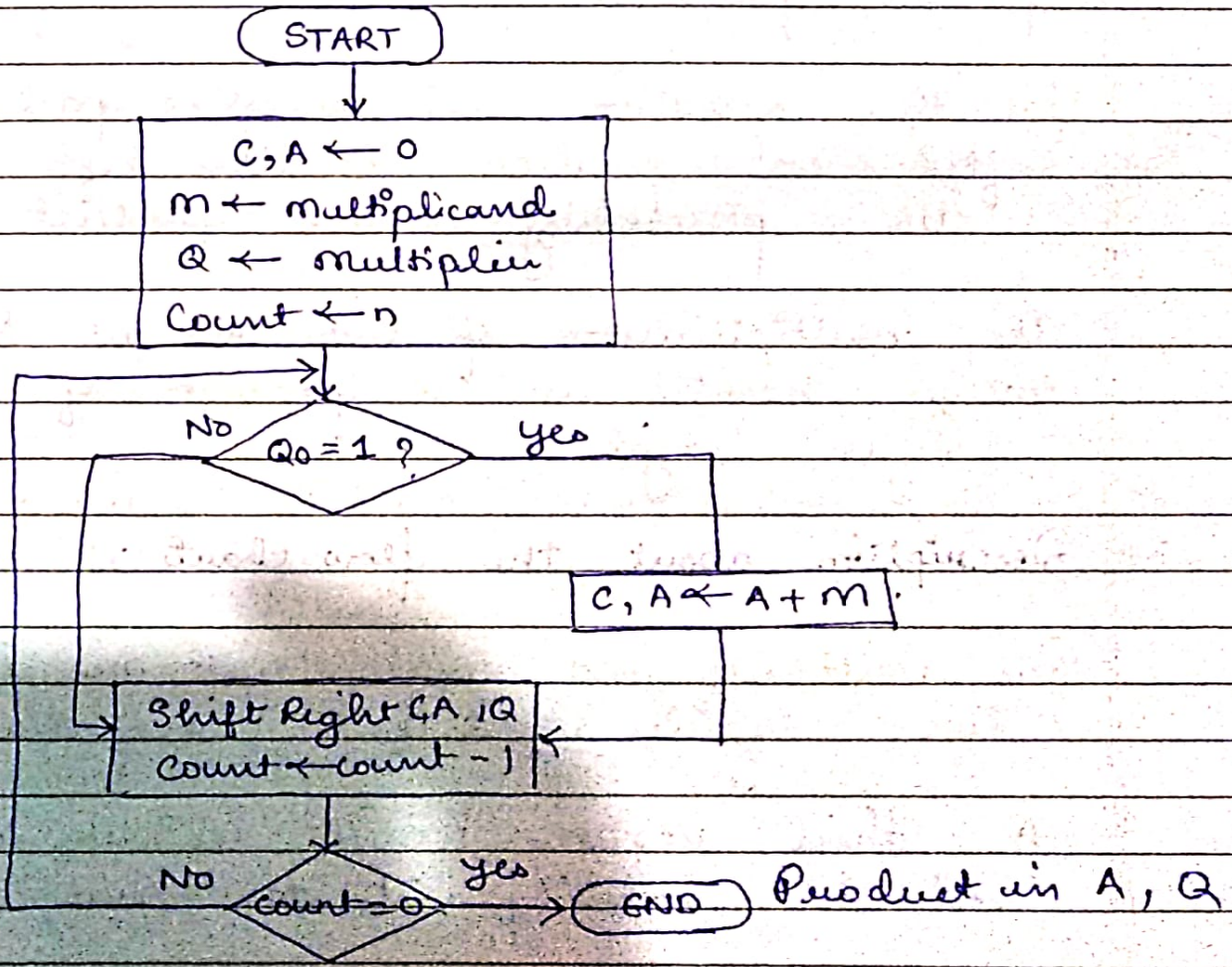
Multiplication (unsigned)

Multiplicand (11) — 1 0 1 1
Multiplier (13) — × 1 1 0 1

1 0 1 1
0 0 0 0 +
1 0 1 1 + +
1 0 1 1 + + +

} Partial product

(143) → 1 0 0 0 1 1 1 1
Actual Product.



Multiplication involves the generation of partial products one for each digit in the multiplier this partial products are the sum are done summation to produce the final product.

When the multiplier bit is zero, the partial product is zero.
When the multiplier is 1, the is multiplicant.

The total product is produced by summing the partial products.

For this operation, each successive partial product is shifted one position to the left relative to the preceding partial product.

The multiplication of two N bit binary integer results in a product of upto 2N bit in length.

⇒ Description about the flow chart.

The multiplier and multiplicant are loaded into two registers Q & M respectively.

A third register i.e 'A' register is also needed and is initially set to zero.

This is also a 1 bit register initialize to zero which holds a potential carry bit resulting

from addition.

Control logic reads the bits of the multiplier one at a time.

If $Q_0 = 1$, then the multiplicant is added to the A register and the result is stored in the A register with the C bit used to overflow.

The all of the bits of C, A & Q registers are shifted to the right 1 bit so that the C bit goes into A_{n-1}, A_0 goes into Q_{n-1} and Q_0 is lost

If Q_0 is zero. then no addition is performed just shift C, A and Q.

This process is repeated for each bit of the original multiplier.

The resulting 2N bit product is contained in the A and Q registers.

Multiplicand - 11 (M) - 1011
Multiplier - 13 (Q) - 1101

C	A	Q	M	
0	0000	1101	1011	← Initial values.
0	1011	1101	1011	Add (A ← A+M) } 1st cycle.
0	0101	1110	1011	Shift Right }
0	0010	1111	1011	Shift Right } 2nd cycle.
0	1101	1111	1011	Add (A ← A+M) } 3rd cycle
0	0110	1111	1011	Shift Right }
1	0001	1111	1011	Add (A ← A+M) } 4th cycle.
0	1000	1111	1011	Shift Right }

Product in A, Q.

1000 1111 = (143).

1st cycle
A = 0000
m = 1011
+
A = 1011

3rd cycle
A = 0010
m = 1011
+
A = 1101

When $Q_0 = 1$ then Add
$Q_0 = 0$, Shift right

Multiplicand = 5 . (M) = 0101
Multiplier = 3 (Q) = 0011

	C	A	Q	M	
	0	0000	0011	0101	← Initial values.
	0	0101	0011	0101	(Add A←A+M)
	0	0010	1001	0101	Shift right
4th cycle.					
A = 0110	0	0111	1001	0101	
m = 1011	0	0011	1100	0101	(Shift righ)
A = 1 0001	0	0001	1110	0101	Shift right
	0	0000	1111	0101	Shift right

Product is in A, Q.

15 = 1111

1st cycle.
A = 0000
m = 0101
A = 0101

3rd cycle
0101
0001

48 32 16 8 4 2 1

Multiplicand = 6 = M 0110

Multiplier = 8 = Q 1000

C	A	Q	M	
0	0000	1000	0110	Initial value.
0	0000	0100	0110	Shift right
0	0000	0010	0110	Shift right
0	0000	0001	0110	Shift right
0	0110	0001	0110	Add (A = A + m)
0	0011	0000	0110	

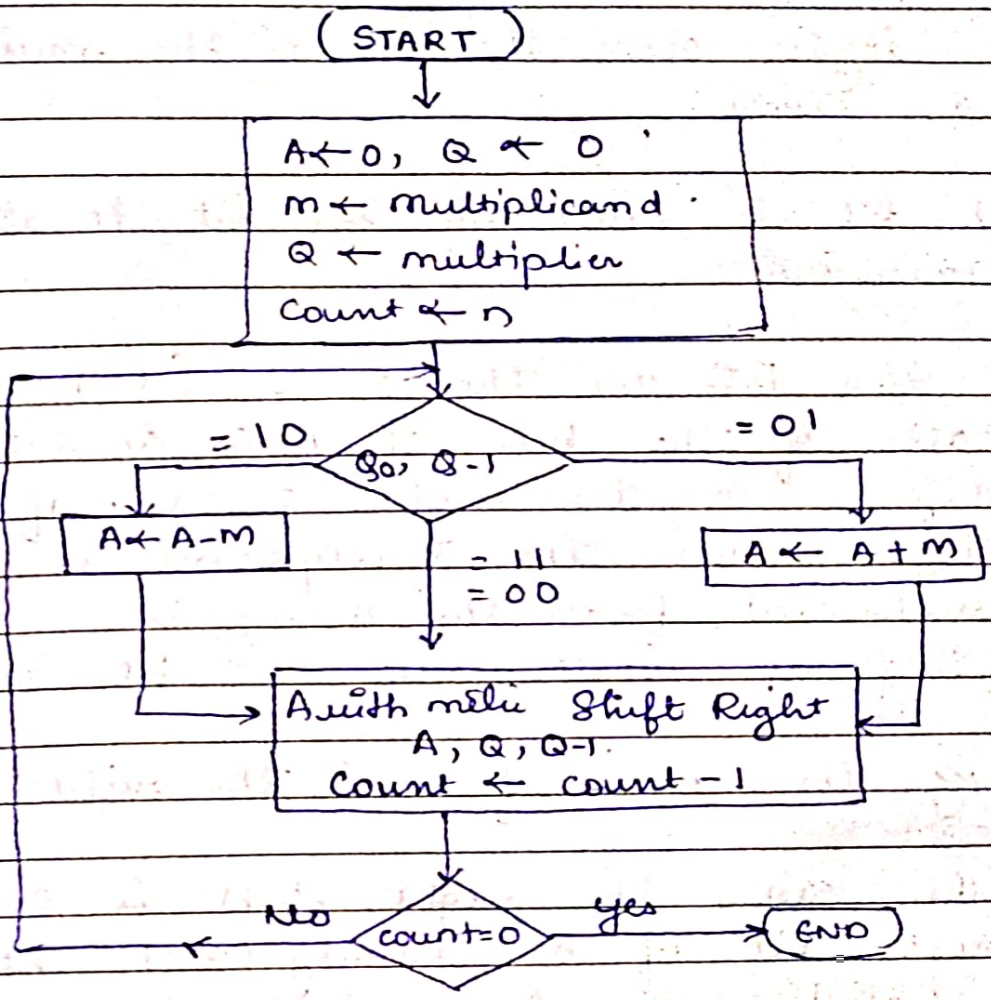
A = 0000
+ M = 0110
A = 0110

Product is in A, Q

∴ 6 × 8 = 48 = 00110000

48

Booth's Algorithm for 2's compliment multiplication.



The multiplier and multiplicand are placed in the Q and m registers respectively.

There is also a 1 bit register placed logically to the right of the least significant bit Q_0 of the Q register and Q-1.

The results of multiplication will appear in A and Q registers. A and Q-1 are initialized to 0.

Control logic uses the bit of the multiplier 1 or 2 times.

The rest bit is examined the bit to its right is also examined.

If the two bits are the same (1-1 or 0-0) then all of the bits of A, Q & Q-1 register are shifted to the right 1 bit. if the two bits differ then the multiplicand is added to or subtracted from the A register depending on whether the 2 bits are 01 or 10.

Following the add/sub the right shift occurs.

In either case, the right shift is such that the left most bit of A namely An-1 not only is shifted into An-2 but also remain An-1

This is required to preserve the sign of the number in A and Q.

It is known as an arithmatic shift because it preserves the signed bit.

10 = A = A − M.   D − 1
11 = Shift.   A ← A + M.
00

Que

Multiplicand = 7 = 0111 (M)
Multiplier = 3 = 0011 (Q)

A	Q_0	Q_{-1}	M	
0000	0011	0	0111	Initial value
1001	0011	0	0111	A ← A − M. } 1st
1100	1001	1	0111	Shift right } 1st
1110	0100	1	0111	Shift right } 2nd.
0101	0100	1	0111	A ← A + M. } 3rd
0010	1010	0	0111	Shift right } 3rd
0001	0101	0	0111	Shift right } 4th.

Product in A, Q.

21 = 0001 0101

1st.

A = 0000   A = 1110
M = 0111   M = 0111
[1] 1001   [1] 0101

X	Y	O/P	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Que

$-7 \leftarrow$ Multiplicand (M)

$+3 \leftarrow$ Multiplier (Q)

$7 \rightarrow 0111$

$1's \rightarrow 1000$

Add 1 = $1001 \leftarrow 2's$.

$\therefore (7) \rightarrow 1001$ M

$(3) \rightarrow 0011$ Q

A	Q_0	Q_{-1}	M	
0000	0011	0	1001	Initial Step
0111	0011	0	1001	$A \leftarrow A - M$. } 1's
0011	1001	1	1001	Shift right }
0001	1100	1	1001	Shift right
1010				
~~0000~~	1100	1	1001	$A \leftarrow A + M$.
~~0~~1101	0110	0	1001	Shift right.
1110	1011	0	1001	Shift right.

Product is in A, Q.

Take 1's = 1110 1011

0001 0100

Add 1

0001 0101

A 0000
M 1001
0111

A 0001
1001
1010

Qu

Ta

A 0000
M 1010
1
1 0110

A ← A+M
1
0011
1010
1101

1110
1010
1 0000

0000
1010
1110

Multiplicand M = -6
Multiplier Q = 5 = 0101

0110
Take 1's Comp - 1001
Add 1 - 1001
+ 1
1010

10
A ← A-M

01
A ← A+M

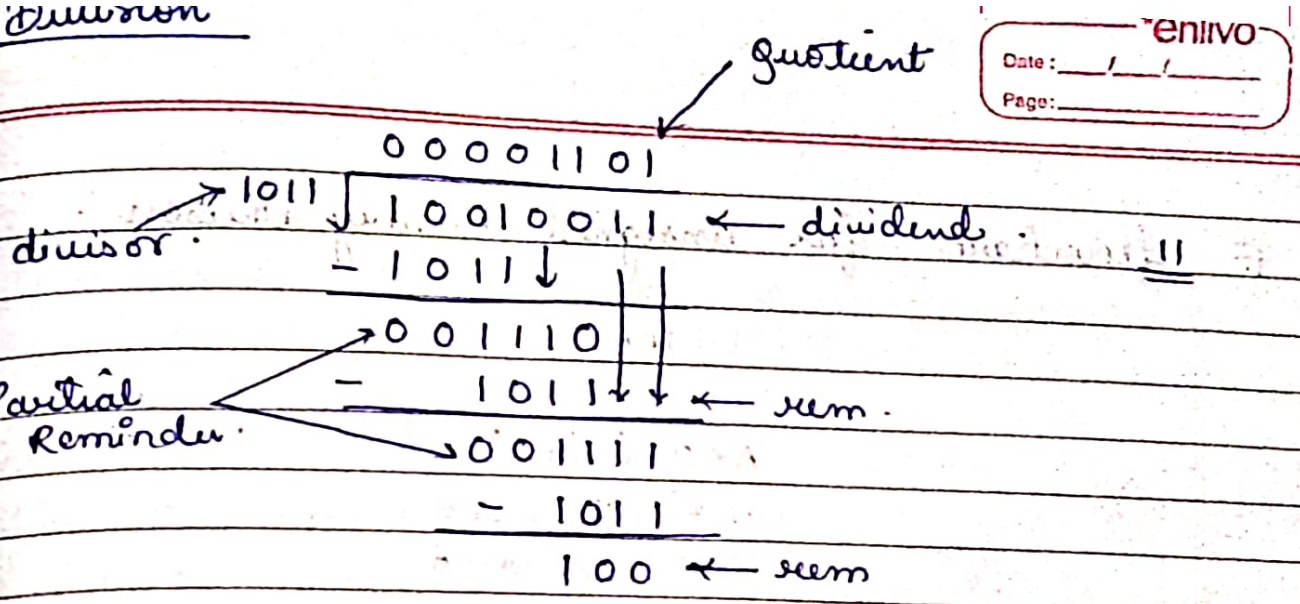
11, 00 Shift

A	Q	Q-1	M	
0000	0101	0	1010	Initially
0110 →	0101 →	0	1010	First cycle
0011	0010	1	1010	Shift right
1101 →	0010 →	1	1010	Second cycle
1110	1001	0	1010	Shift right
0100	1001	0	1010	Third cycle
0010	0100	1	1010	Shift right
1100 →	0100	1	1010	Fourth cycle
1110	0010	0	1010	Shift right

1110 0010
Take 1's 0000 0010
Add 1 0000 ~~00010~~ 1101
+ 1
1111

1110 0010
+0001 1101
1
0001 1110

Division



Division of unsigned binary Integer.

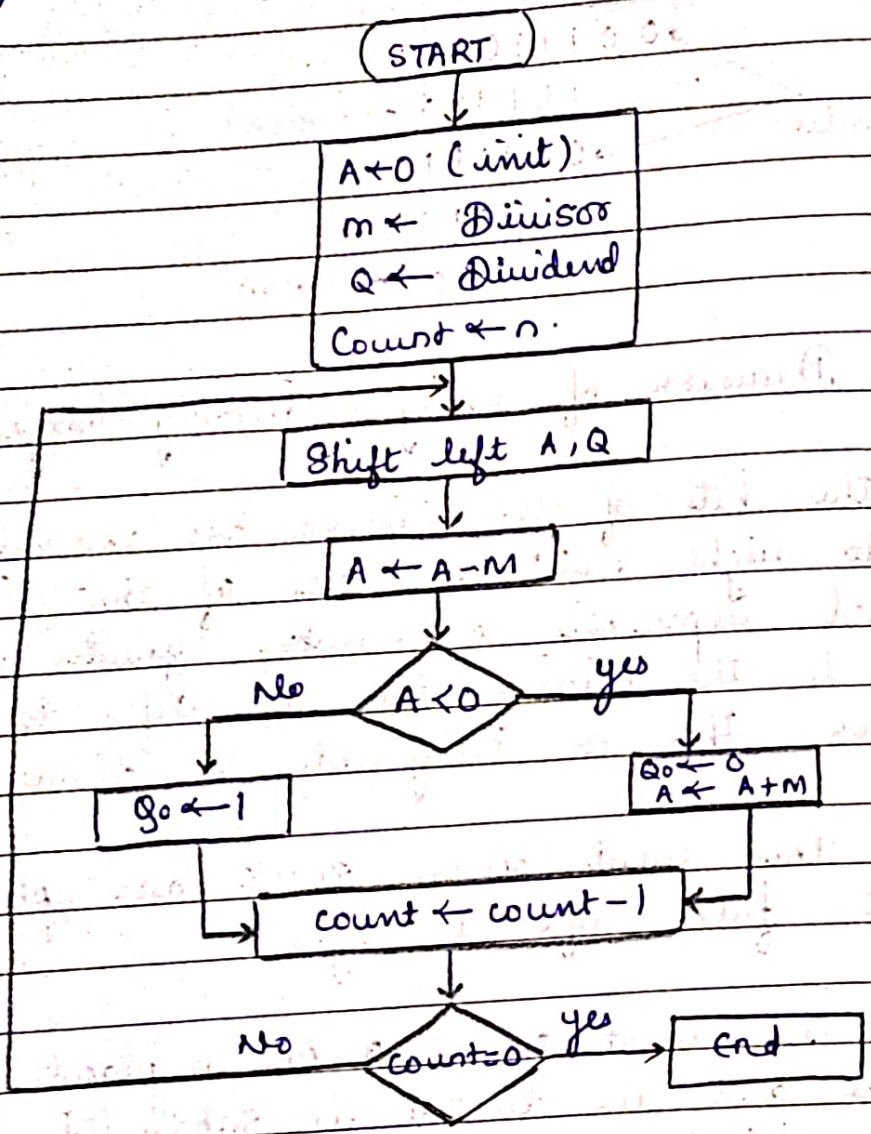
First the bits of the dividend are examined for left to right until the sets of the bit examined represents a number greater than or equal to the divisor This is refer to as the divisor ~~this is~~ being able to divide the no.

Until this event occurs zero's are placed in the quotient from left to right.

When the event occurs a one is placed in the quotient and the divisor is subtracted from the partial dividend.

The result is referred as partial remainder the process continues until all the bits of the dividend are exhausted.

Flowchart for unsigned binary Division.



Quotient is Q
Remainder in A.

1. Figure shows, a machine algorithm that corresponds to the long division process.

2. The divisor is placed in the M register, the dividend in the Q register. At each step, the A and Q registers together are shifted to left by 1 bit.

3. M is subtracted from A to determine wheather A divides the partial reminder.

4. If it does then Qo gets a one bit. Otherwise Qo gets a zero bit and M must be added back to A to restore the previous value.

5. The count is then decremented and the process continous for N step.

6. At the end the quotient is the Q register and the reminder is in the A register.

Algorithm for division of signed Integers.

1. load the divisor into the M register and the dividend into A and Q register.

   The dividend must be represented as a 2N bit 2's complement no.

2. Shift A, Q left one bit position

3. If m and A have the same signs perform
   $A \leftarrow A - M$
   Otherwise $A \leftarrow A + M$

4. The preceeding operation is successfull if the sign of A is the same before and after the operation.

   a) If the operation is successfull or $A = 0$, then set $Q_0 \leftarrow 1$.

   b) If the operation is unsuccessfull and $A \neq 0$, then set $Q_0 \leftarrow 0$ and restore the previous value of A.

5. Repeat steps 2 through 4 as many times as their are bit positions in Q.

6. The remainder is in A, if the signs of the divisor and dividend where the same

then the Quotient is in Q otherwise the correct quotient is the 2's complement of Q.

Que ÷7 ÷ (-3) or 7 ÷ (-3)

D = Q × V + R
↑ divisor.

Q = 7 (dividend)
m = 3 (divisor)

eg. 7 ÷ 3.

	A	Q	M	
	0000	0111	0011	Initial value.
1st cycle.	0000	1110	0011	Shift left
A = 0000	1101			Substract } 1st cycle
m = 0011	0000	1110	0011	Restore
1101				
2nd cycle	0001	1100	0011	Shift left
A = 0001	1110			Substract } 2nd cycle
m = 0011	0001	1100		Restore
1110				
	0011	1000	0011	Shift left
	0000			Subtract } 3rd cycle
3rd cycle	0000	1001	0011	Set Q0 = 1
A = 0011				
m = 0011	0001	0010	0011	Shift left
0000	1110			Subtract } 4th cycle
4th cycle	0001	0010		restore
A = 0001				
m = 0011				
1110				

Remainder = 0001
Quotient = 0010.