Binary to Octal, and Hexadecimal

Binary	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Binary to Octal, and Hexadecimal

Going from octal or hexadecimal to binary is a straightforward process. We just replace the octal or hexadecimal digit with its binary equivalent.

347646_8 => $011\ 100\ 111\ 110\ 100\ 110_2$
$B1FF74_{16}$ => $1011\ 0001\ 1111\ 1111\ 0111\ 0100_2$

When going from binary to octal or hexadecimal, we have to ensure the number of bits is a multiple of the base. If it is not, we pad the number until it is.

$1\ 101\ 011\ 110\ 000\ 110_2$ = $111\ 101\ 011\ 110\ 000\ 110_2$ => 753606_8
$101\ 0101\ 0001\ 0111\ 0100_2$ = $1101\ 0101\ 0001\ 0111\ 0100_2$ = $D5174_{16}$

Signed Binary Multiplication

For the multiplication of two binary numbers we use Booth's algorithm.

Before starting, ensure both numbers are of the same length. The length of the numbers we call ***n*** and the length of the product will be at most **2*n***.

Create 4 variables:

- **A** = 0… (padded to a bit length of ***n***)
- **M** = Multiplicand (first number)
- **Q** = Multiplier (second number)
- $\mathbf{Q_{-1}}$ = 0, (bit length of 1)

We will also be referring to a variable $\mathbf{Q_0}$ that represents the least significant bit or the right most bit of the binary number in **Q**.

Let variables separated by commas be considered a concatenation of the variables: $\mathbf{A,Q,Q_{-1}}$ (length of **2*n* + 1**) and $\mathbf{Q_0,Q_{-1}}$ (length of **2**)

Signed Binary Multiplication

Booth's Algorithm

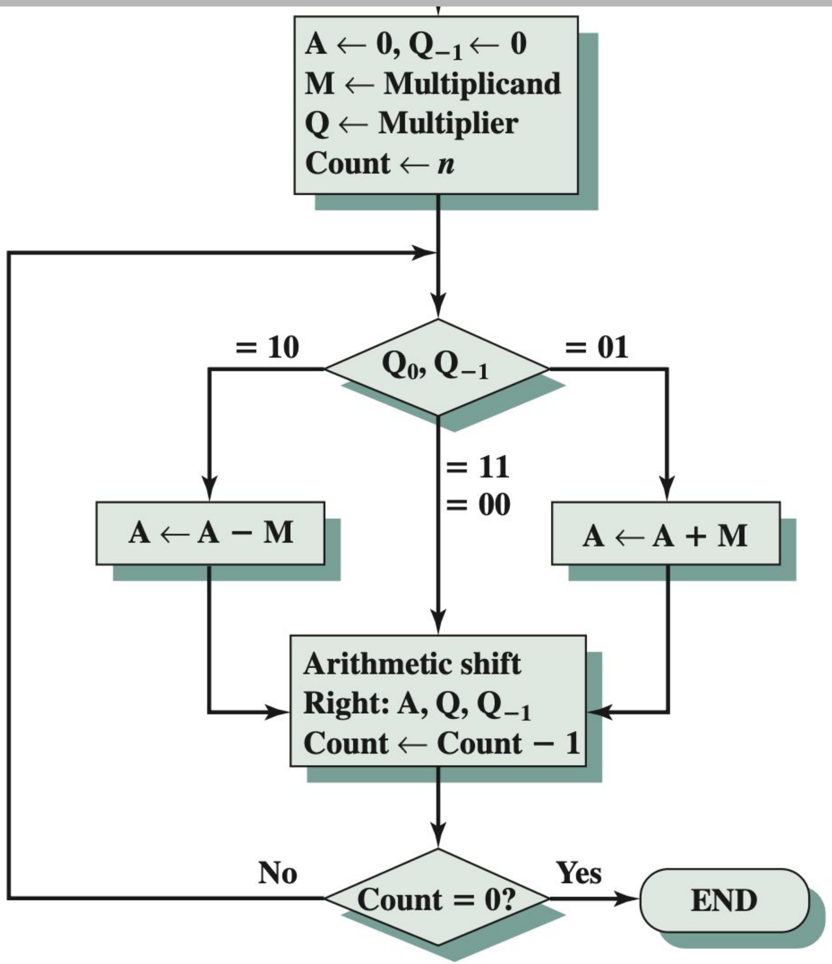
For ***n*** times:

1. If $\mathbf{Q_0,Q_{-1}}$ **= 01**: **A <= A + M**

   Else if $\mathbf{Q_0,Q_{-1}}$ **= 10**: **A <= A - M**

2. $\mathbf{A,Q,Q_{-1}}$ >> 1 (arithmetic shift)

product = **A,Q**



Signed Binary Multiplication

Lets test **0111 x 0011**. We know 0111 = 7 and 0011 = 3. Since 7 x 3 = 21, our binary product should be **00010101** untrimmed.

A	Q	Q_{-1}	M		
0000	0011	0	0111	**Initial values**	
1001	0011	0	0111	**A←A – M**	**First cycle**
1100	1001	1	0111	**Shift**	
1110	0100	1	0111	**Shift**	**Second cycle**
0101	0100	1	0111	**A←A + M**	**Third cycle**
0010	1010	0	0111	**Shift**	
0001	0101	0	0111	**Shift**	**Fourth cycle**

Signed Binary Multiplication

Lets try: 01011 x 1011… 11 x -5 = -55… 1001001

A	Q	Q_{-1}	M		
00000	11011	0	01011	Initial values	
10101 11010	11011 11101	0 1	01011 01011	A = A - M Shift	First Cycle
11101	01110	1	01011	Shift	Second Cycle
01000 00100	01110 00111	1 0	01011 01011	A = A + M Shift	Third Cycle
11001 11100	00111 10011	0 1	01011 01011	A = A - M Shift	Fourth Cycle
11110	01001	1	01011	Shift	Fifth Cycle

Signed Binary Multiplication

Lets try: 1010 x 1010… -6 x -6 = 36… 0100100

A	Q	Q_{-1}	M		
0000	1010	0	1010	Initial values	
					First Cycle
					Second Cycle
					Third Cycle
					Fourth Cycle

Signed Binary Multiplication

Lets try: 01111 x 01010… 15 x 10 = 150… 0010010110

A	Q	Q_{-1}	M		
00000	01010	0	01111	Initial values	
00000	00101	0	01111	Shift	First Cycle
10001 11000	00101 10010	0 1	01111 01111	A = A - M Shift	Second Cycle
00111 00011	10010 11001	1 0	01111 01111	A = A + M Shift	Third Cycle
10100 11010	11001 01100	0 1	01111 01111	A = A - M Shift	Fourth Cycle
01001 00100	01100 10110	1 0	01111 01111	A = A + M Shift	Fifth Cycle

Signed Binary Division

For signed binary division, we need to treat the operands as positive. This requires us to convert negative numbers to their two's complement. We must also keep track of the signs of the original numbers. The result of our division algorithm will produce the quotient as well as the remainder. The sign of the quotient and remainder will depend on the signs of the original numbers.

Let S_D = sign of the dividend

Let S_V = sign of the divisor

If $S_D \neq S_V$: the sign of the quotient is negative (two's complement)

If $S_D < 0$: the sign of the remainder is negative (two's complement)

$D \div V = Q, R$ $\quad$ $D \div -V = -Q, R$ $\quad$ $-D \div V = -Q, -R$ $\quad$ $-D \div -V = Q, -R$

Signed Binary Division

Before starting, ensure the length of the dividend is greater than or equal to the length of the divisor. The length of the dividend we call ***n***. The length of the divisor we call ***m***.

Create 3 variables:

- **A** = 0… (padded to a bit length ***m***)
- **Q** = Dividend (first number)
- **M** = Divisor (second number)

We will also be referring to a variable $\mathbf{Q_0}$ that represents the least significant bit or the right most bit of the binary number in **Q**.

Let variables separated by commas be considered a concatenation of the variables: **A,Q** (length of ***m + n***)

Signed Binary Division

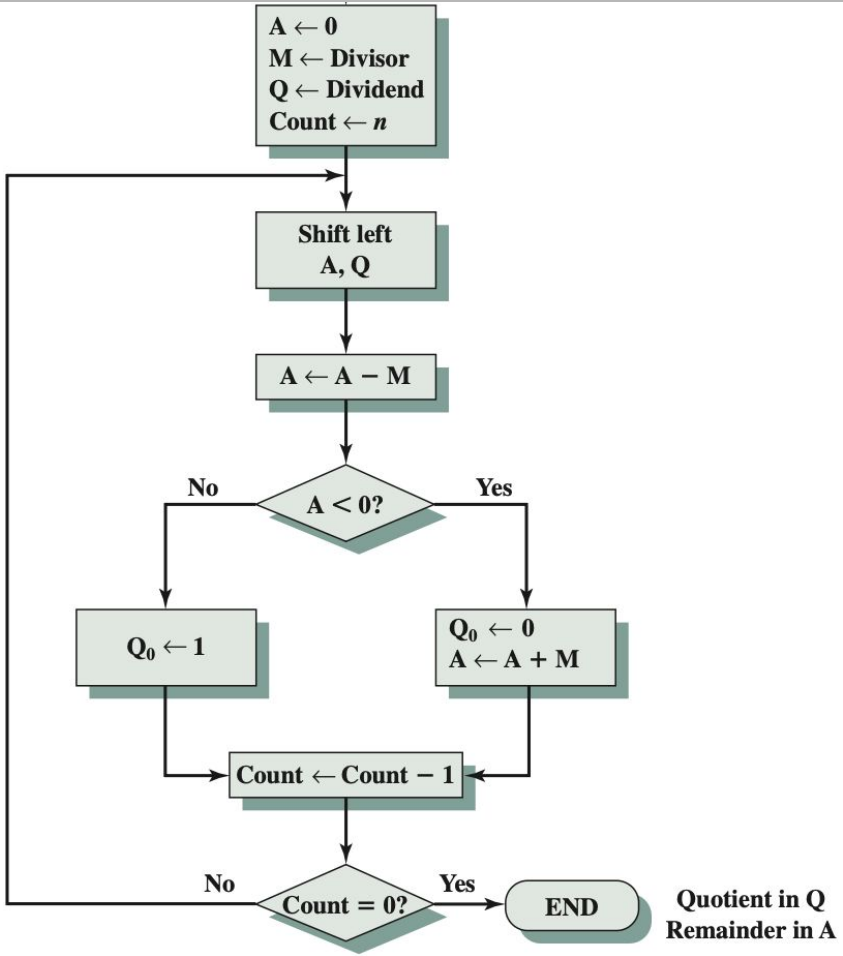
For ***n*** times:

1. **A,Q** << 1 (logical shift)
2. **A** <= **A** - **M**
3. If **A** is negative: $\mathbf{Q_0}$ <= 0

   A <= **A** + **M**

   Else: $\mathbf{Q_0}$ <= 1

Quotient = **Q**
Remainder = **A**



Signed Binary Division

Lets try:

01010 ÷ 010 = 00000101

10 ÷ 2 = 5

Quotient = 00101
Remainder = 000

$S_D == S_V$: no change

$S_D >= 0$: no change

A	Q	M		
000	01010	010	Initial values	
000 110 110 000	10100 10100 10100 10100	010 010 010 010	A,Q << 1 A = A - M Q_0 = 0 A = A + M	1st Cycle
001 111 111 001	01000 01000 01000 01000	010 010 010 010	A,Q << 1 A = A - M Q_0 = 0 A = A + M	2nd Cycle
010 000 000	10000 10000 10001	010 010 010	A,Q << 1 A = A - M Q_0 = 1	3rd Cycle
001 111 111 001	00010 00010 00010 00010	010 010 010 010	A,Q << 1 A = A - M Q_0 = 0 A = A + M	4th Cycle
010 000 000	00100 00100 00101	010 010 010	A,Q << 1 A = A - M Q_0 = 1	5th Cycle

Signed Binary Division

Lets try:

0111 ÷ 101 = 0011110

7 ÷ -3 = -2 remainder 1

Q = 0010
A = 001

S_D != S_V : two's comp Q

S_D >= 0 : no change

Quotient = 1110
Remainder = 001

A	Q	M		
000	0111	011	Initial values	
000 101 101 000	1110 1110 1110 1110	011 011 011 011	A,Q << 1 A = A - M $Q_0 = 0$ A = A + M	1st Cycle
001 110 110 001	1100 1100 1100 1100	011 011 011 011	A,Q << 1 A = A - M $Q_0 = 0$ A = A + M	2nd Cycle
011 000 000	1000 1000 1001	011 011 011	A,Q << 1 A = A - M $Q_0 = 1$	3rd Cycle
001 110 110 001	0010 0010 0010 0010	011 011 011 011	A,Q << 1 A = A - M $Q_0 = 0$ A = A + M	4th Cycle

Signed Binary Division

Lets try:

$1100 \div 011 = 1111111$

$-4 \div 3 = -1$ remainder -1

Q = 0001
A = 001

$S_D \mathrel{!}= S_V$: two's comp Q

$S_D < 0$: two's comp A

Quotient = 1111
Remainder = 111

A	Q	M		
000	0100	011	Initial values	
000 101 101 000	1000 1000 1000 1000	011 011 011 011	A,Q << 1 A = A - M $Q_0 = 0$ A = A + M	1st Cycle
001 110 110 001	0000 0000 0000 0000	011 011 011 011	A,Q << 1 A = A - M $Q_0 = 0$ A = A + M	2nd Cycle
010 111 111 010	0000 0000 0000 0000	011 011 011 011	A,Q << 1 A = A - M $Q_0 = 0$ A = A + M	3rd Cycle
100 1 001 1 001	0000 0000 0001	011 011 011	A,Q << 1 A = A - M $Q_0 = 1$	4th Cycle

Signed Binary Division

Lets try:

1000 ÷ 0101 = 11011111

-8 ÷ 5 = -1 remainder -3

Quotient = 0001
Remainder = 0011

S_D != S_V : two's comp Q

S_D < 0 : two's comp A

Quotient = 1111
Remainder = 1101

A	Q	M		
0000	1000	0101	Initial values	
		0101 0101 0101 0101		1st Cycle
		0101 0101 0101 0101		2nd Cycle
		0101 0101 0101 0101		3rd Cycle
		0101 0101 0101		4th Cycle