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

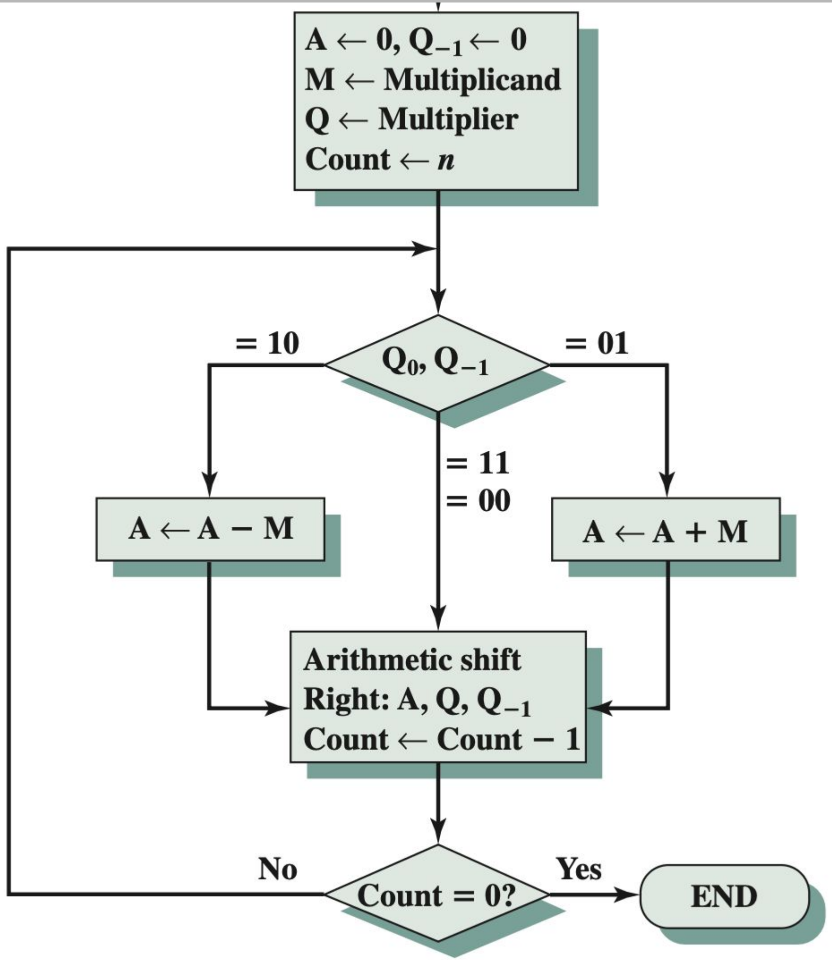
For ***n*** times:

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