Binary multiplication and division

Multiplication in Signed-Magnitude System

Steps:

1. Use XOR function to determine the product sign

$$\text{Sg } X \oplus \text{Sg } Y = \text{Sg } P$$

2. Perform an unsigned multiplication of the magnitudes.

3. Convert the result to two’s complement system (if the sign is negative).

Multiplication, alg.1

$X=-79$ $X_{CC}=1.0110001$ $SgZ=1\oplus 0=1$

$Y=81$ $Y=0.1010001$ $|x|=0.1001111$

$|Y|=0.1010001$

			Y												Adder								
		1	0	1	0	0	0	1		0	0	0	0	0	0	0	0	0	0	0	0	0	0
	+X																1	0	0	1	1	1	1
→4	Y	0	0	0	0	1	0	1		0	0	0	0	0	0	0	1	0	0	1	1	1	1
	+X												1	0	0	1	1	1	1				
→2	Y	0	0	0	0	0	0	1		0	0	0	1	0	1	0	0	1	1	1	1	1	1
	+X										1	0	0	1	1	1	1						
										0	1	1	0	0	0	1	1	1	1	1	1	1	1

$|Z|=0.01100011111111$

$Zcc=1.10011100000001$

Multiplication, alg.2

X=121 X=0.1111001

Y=-98 Ycc=1.0011110

SgZ=1⊕0=1

|x|=0.1111001

|Y|=0.1100010

			Y												Adder								
		1	1	0	0	0	1	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0
	+X										1	1	1	1	0	0	1						
←1	Y	1	0	0	0	1	0	0		0	1	1	1	1	0	0	1	0	0	0	0	0	0
	+X											1	1	1	1	0	0	1					
←4		1	0	0	0	0	0	0		1	0	1	1	0	1	0	1	1	0	0	0	0	0
	+X															1	1	1	1	0	0	1	
										1	0	1	1	1	0	0	1	0	1	0	0	1	0

|Z|=0.10111001010010

Zcc=1.01000110101110

Division with reestablishment of the remainder

Division of signed numbers can be accomplished using unsigned division.

If we have fixed-point fraction we can divide numbers if only **|A|<|B|**, because otherwise we can obtain a integer part of a quotient and this is overflow.

1. Sign bit is computed as XOR of input sign bits.
2. Find absolute values of A and B (operands).
3. First subtraction |A|-|B| (is substituted with addition in two's complemented system).
4. a) If the remainder is positive, then operation is stopped and the pseudo sign bit is 1.

   b) If the remainder is negative, |A|<|B| and the pseudo sign bit is 0.
5. The reestablishment of the dividend is done by addition of the divisor to the remainder.
6. The dividend is shifted left.
7. Subtraction of the divisor.

   a) If the remainder is positive the quotient bit is 1. The remainder is shifted left.

   b) If the remainder is negative, the quotient bit is 0 and the reestablishment of the last positive remainder is done. Then it is shifted left.

- The number of iterations depends on the required precision.
- The algorithm can be stopped when the remainder is 0.

- X=1.01101
- Y=0.11001

Q	Adder	Comm

- $X=0.11011$
- $Y=1.00010$

Q	Adder	Comm

Division without reestablishment of the remainder

- Rule: After first control subtraction, the sign of the remainder is examined.
- If the sign is positive, the remainder is shifted and then subtraction of |B| is done.
- If the sign is negative, the remainder is shifted and the addition of |B| is done.
- The quotient is obtained using the same rules as in first algorithm.

- X=1.01101
- Y=0.11001

Q	Adder	Comm

- $X=0.11011$
- $Y=1.00010$

Q	Adder	Comm