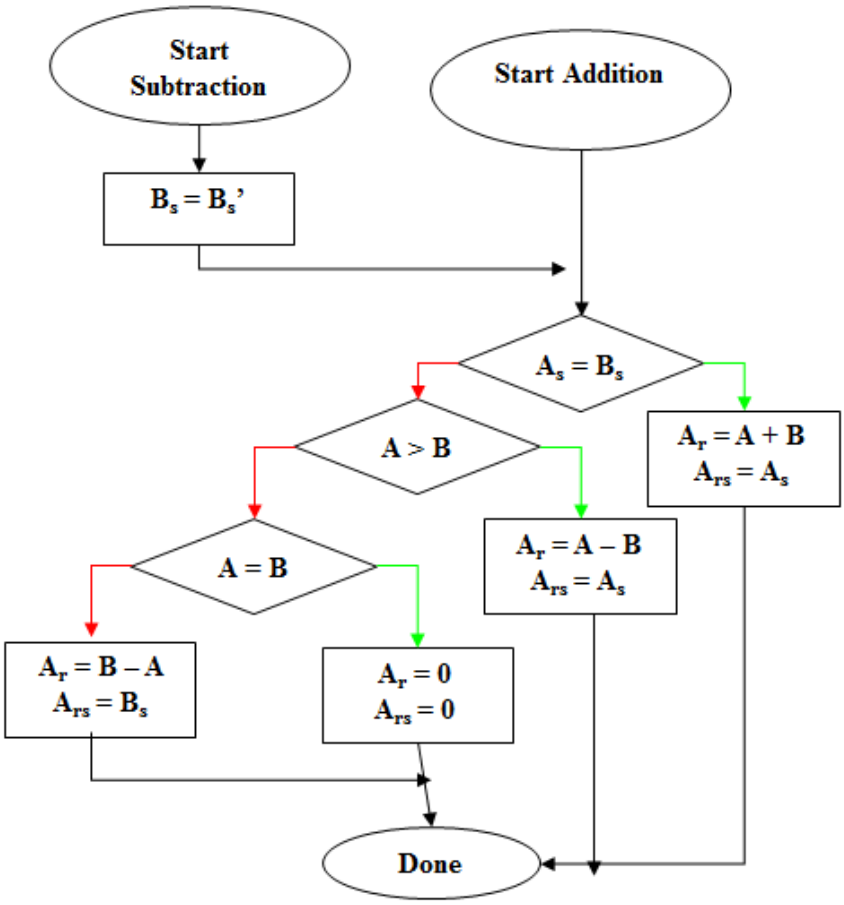
Các phép toán số học trên các số lượng dấu, bù 1, Bias, BCD

1. Số lượng dấu

Operation	ADD Magnitudes	SUBTRACT Magnitudes		
		A > B	A < B	A = B
(+A) + (+B)	+ (A + B)			
(+A) + (-B)		+ (A – B)	- (B – A)	+ (A – B)
(-A) + (+B)		- (A – B)	+ (B – A)	+ (A – B)
(-A) + (-B)	- (A + B)			
(+A) - (+B)		+ (A – B)	- (B – A)	+ (A – B)
(+A) - (-B)	+ (A + B)			
(-A) - (+B)	- (A + B)			
(-A) - (-B)		- (A – B)	+ (B – A)	+ (A – B)

Từ đó ta có thuật toán sau:



Ví dụ:

```
78 – (-5)

 As                 As

 0100 1110  =>      0100 1100

Bs                 Bs

-1000 0101         +0000 0101

                    0101 0011
```

2. Số bù 1

Tham khảo thuật toán dưới đây:

Adding two values is straight forward. Simply align the values on the least significant bit and add, propagating any carry to the bit one position left. If the carry extends past the end of the word it is said to have "wrapped" around, a condition called an "end-around carry". When this occurs, the bit must be added back in at the right-most bit. This phenomenon does not occur in two's complement arithmetic.

```
  0001 0110     22
+ 0000 0011      3
===========   ====
  0001 1001     25
```

Subtraction is similar, except that borrows are propagated to the left instead of carries. If the borrow extends past the end of the word it is said to have "wrapped" around, a condition called an "end-around borrow". When this occurs, the bit must be subtracted back in at the right-most bit. This phenomenon does not occur in two's complement arithmetic.

```
  0000 0110      6
- 0001 0011     19
===========   ====
1 1111 0011    -12    —An end-around borrow is produced,
- 0000 0001      1    —Subtract the end-around borrow back into the result.
===========   ====
  1111 0010    -13    —The correct result (6 – 19 = -13)
```

It is easy to demonstrate that the bit complement of a positive value is the negative magnitude of the positive value. The computation of 19 + 3 produces the same result as 19 − (−3).

Add 3 to 19.

```
  0001 0011     19
+ 0000 0011      3
===========   ====
  0001 0110     22
```

Subtract −3 from 19.

```
  0001 0011     19
- 1111 1100     -3
===========   ====
1 0001 0111     23    —An end-around borrow is produced.
- 0000 0001      1    —Subtract the end-around borrow back into the result.
===========   ====
  0001 0110     22    —The correct result (19 - (-3) = 22).
```

Ví dụ:

78 – (-5)

```
0100 1110

-1111 1010

1 0101 0100

- 0000 0001

  0101 0011
```

3. Số Bias

Bias Addition

Assume a, b, c are three values:

$$(a+b)=c \qquad \text{(values)}$$

$$\begin{aligned}&(a+k)+(b+k) \quad \text{(representations)}\\ &=(a+b)+2k\\ &=(c+k)+k\end{aligned}$$

Rewriting A, B, C in *Excess-K* representations:

$$\begin{aligned}A+B&=C+k\\ C&=(A+B)-k\end{aligned}$$

Ví dụ:

```
78 - (-5)

 1100 1101

-0111 1010

 0101 0011

+0111 1111

 1101 0010
```

4. Số BCD

Tham khảo thuật toán dưới đây:

BCD Addition

Either packed or unpacked BCD numbers can be summed. BCD addition follows the same rules as binary addition. However, if the addition produces a carry and/or creates an invalid BCD number, an adjustment is required to correct the sum. The correction method is to add 6 to the sum in any digit position that has caused an error.

Ví dụ:

```
24 + 13 = 37
  0010 0100    = 24
+ 0001 0011    = 13
-----------    ----
  0011 0111    = 37
```

```
15 + 9 = 24
  0001 0101    = 15
+ 0000 1001    =  9
-----------    ----
  0001 1110    = 1? (invalid)

  0001 1110    = 1? (invalid)
+ 0000 0110    =  6 (adjustment)
-----------    ----
  0010 0100    = 24
```

```
19 + 28 = 47
  0001 1001    = 19
+ 0010 1000    = 28
-----------    ----
  0100 0001    = 41 (error)

  0100 0001    = 41 (error)
+ 0000 0110    =  6 (adjustment)
-----------    ----
  0100 0111    = 47
```

BCD Subtraction

Either packed or unpacked BCD numbers can be subtracted. BCD subtraction follows the same rules as binary subtraction. However, if the subtraction causes a borrow and/or creates an invalid BCD number, an adjustment is required to correct the answer. The correction method is to subtract 6 from the difference in any digit position that has caused an error.

Ví dụ:

```
37 - 12 = 25
  0011 0111  =  37
- 0001 0010  =  12
-----------    ----
  0010 0101  =  25
```

```
65 - 19 = 46
  0110 0101    = 65
- 0001 1001    = 19
-----------    ----
  0100 1100    = 4? (invalid)

  0100 1100    = 4? (invalid)
- 0000 0110    =  6 (adjustment)
-----------    ----
  0100 0110    = 46
```

```
41 - 18 = 23
  0100 0001    = 41
- 0001 1000    = 18
-----------    ----
  0010 1001    = 29 (error)

0010 1001      = 29 (error)
- 0000 0110    =  6 (adjustment)
-----------    ----
  0010 0011    = 23
```

Ví dụ:

```
78 – (–5)

 0000 0111 1000

–1001 1001 0101

0110 1110 0011

–0110 0110 0000

 0000 1000 0011
```