Finite representation of real numbers

Fixed-point arithmetic

Dr. Ing. Rodrigo Gonzalez

rodralez@frm.utn.edu.ar
rodrigo.gonzalez@ingenieria.uncuyo.edu.ar

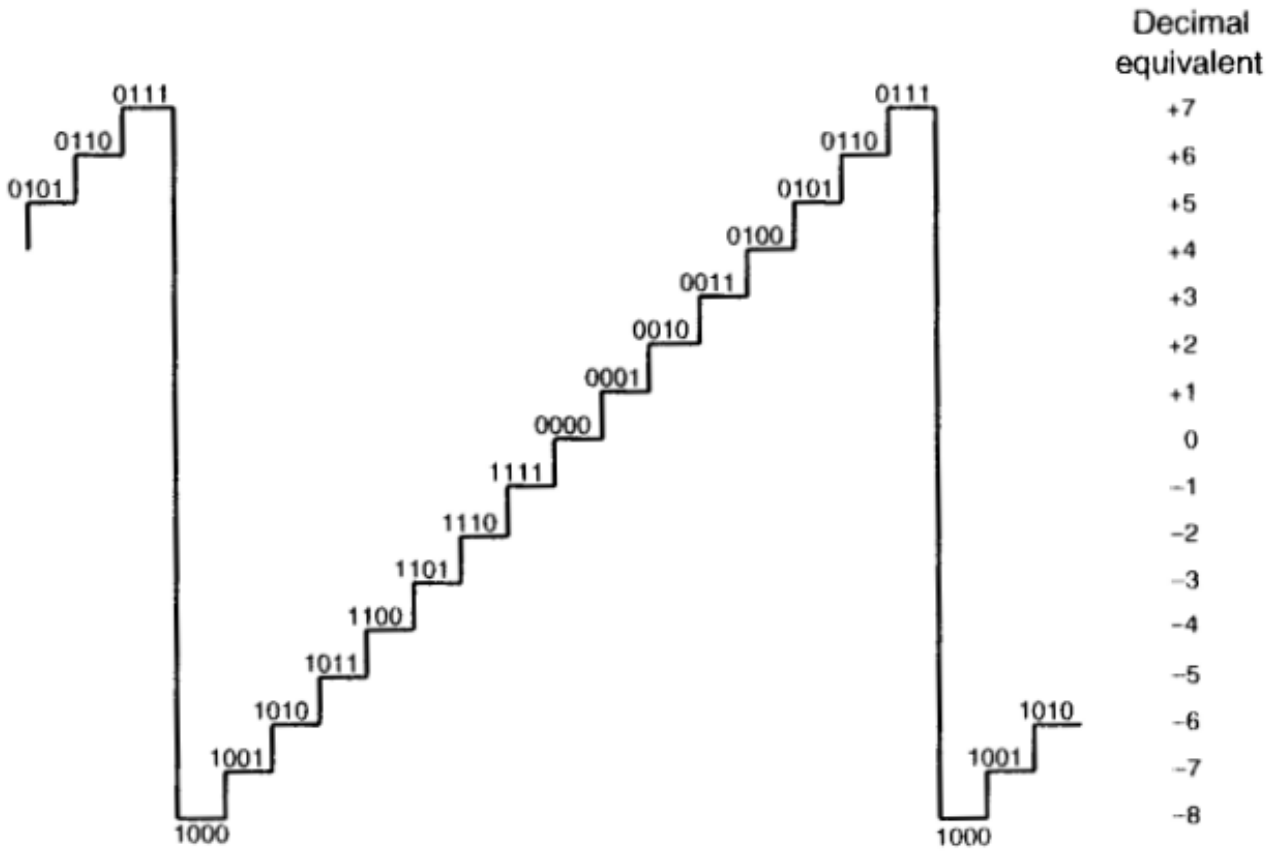
Summary

- Adding two N-bits numbers can produce a N+1 bits result.
- The result will have the same numbers of fractional bits.
- Only the integer part can grow.

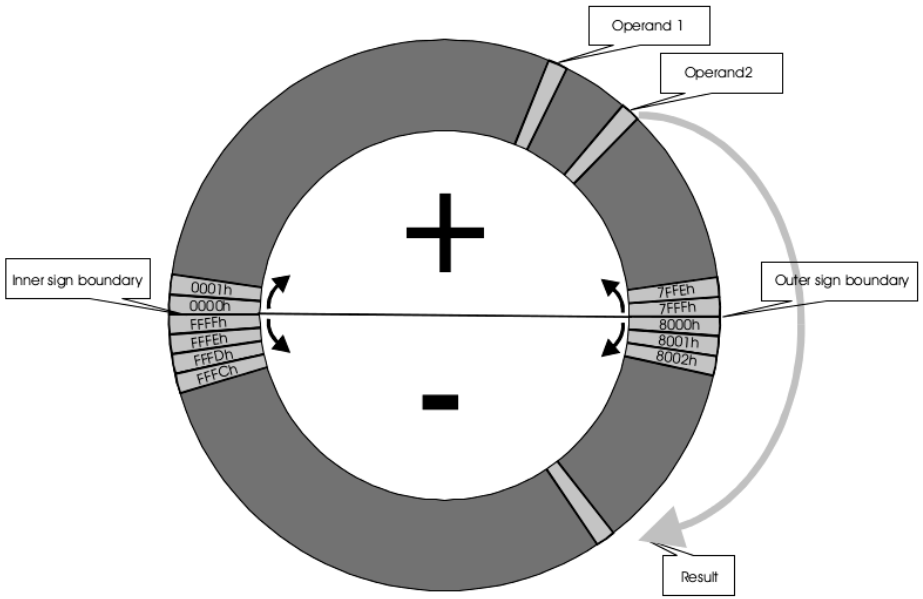
```
 11111 111   (carry)        0111   (carry)
  0000 1111  (15)             0111  (7)
+ 1111 1011  (-5)           + 0011  (3)
=================          ============
  0000 1010  (10)             1010  (-6)  invalid!
```

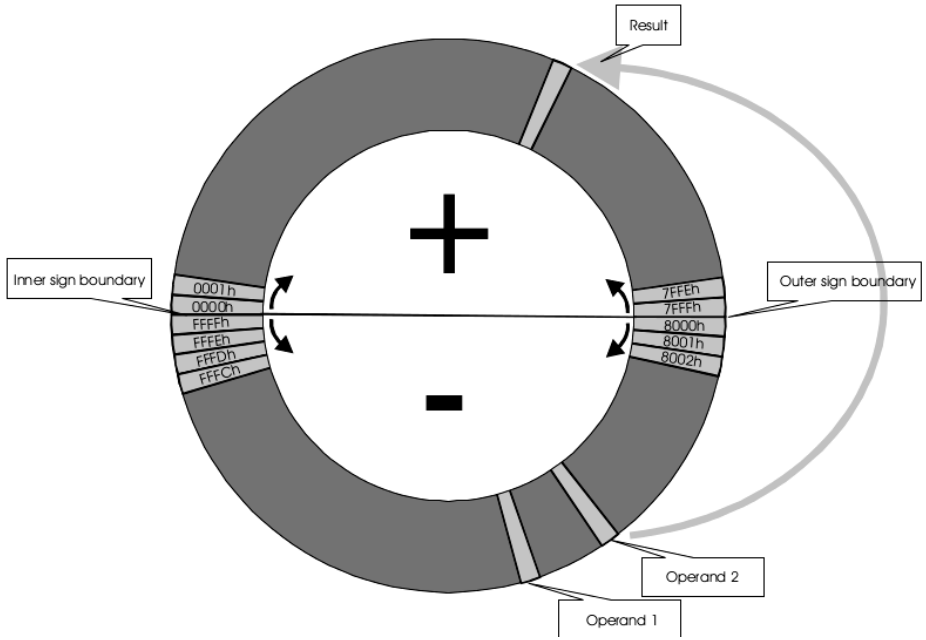
- The last two bits of the carry row show if overflow occurs.

- An **overflow** occurs when a result is greater than $(2^{N-1}-1)$,or lesser than -2^{N-1}.
- An overflow produces a **roll-over** (wrap).



- A roll-over usually has catastrophic consequences on a process.
- It only happens when two very large positive operands, or two very large negative operands, are added.
- It can never happen during the addition of a positive operand and a negative operand, whatever their magnitude.





How to avoid overflow

Longer word-length accumulator

- Saving the result in a N+1 word avoids overflows.
- The general rule is the sum of s individual m-bit can require an accumulator of as many as $m + log_2(s)$ bits.
- **Example:** 256 8-bits words requires an accumulator whose word length is $8 + log_2(256) = 16$.
- A 16-bits DSP processor usually have a 40-bit ALU accumulator.
- How many sums are supported by a 40-bits accumulator for 16-bits numbers?

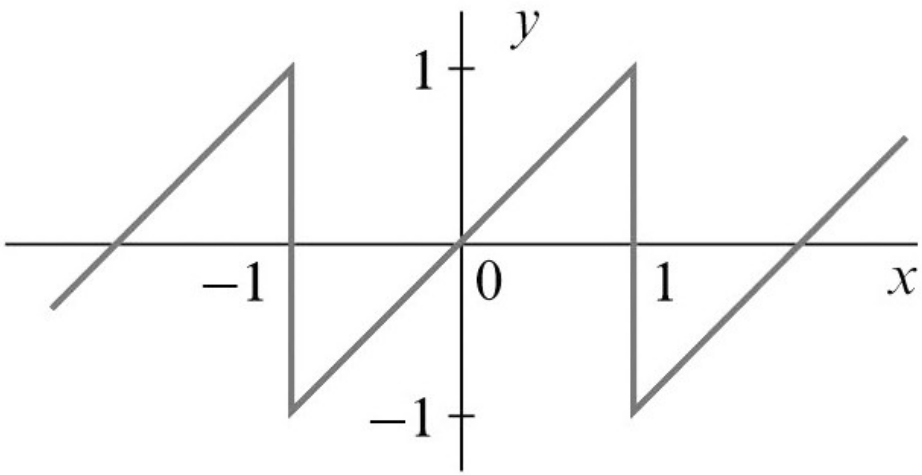
  $16 + log_2(s) = 40$.

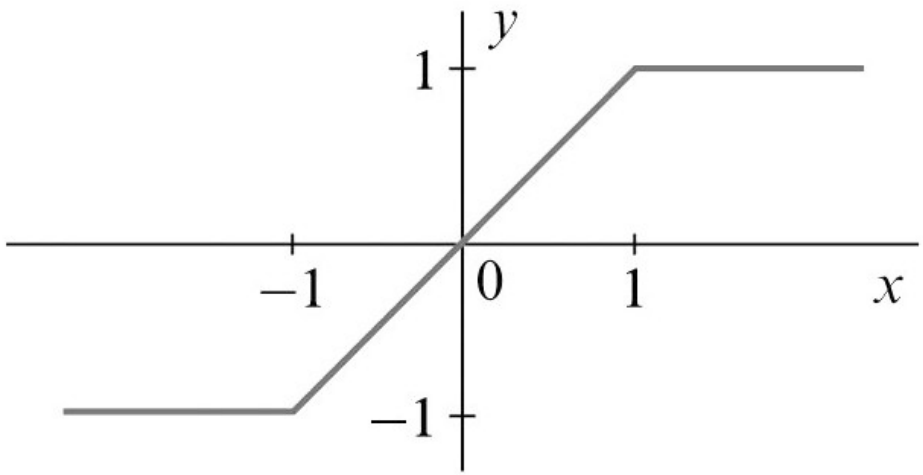
In C:

C code

```
1 int32_t a[K] ;
2 int64_t c = 0 ;
3 for (i=0; i<K; i++)
4  { c = c + (int64_t) a[i] };
```

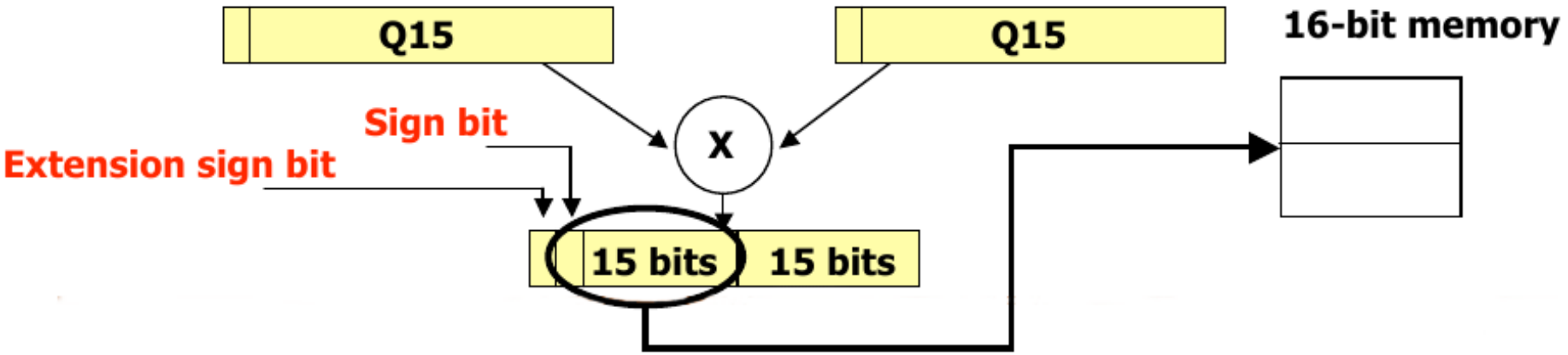
- To avoid a rollover, overflow is detected and the result is saturated to the most positive or most negative value that can be represented.
- This procedure is called **saturation arithmetic**.
- DSP processors allows the results to be saturated automatically in hardware (In TI DSP C5505, SATD Bit at ST1_55 register).





Be aware of non-linearity!

- The product of 2 N-bit numbers requires $2 \cdot N$ bits bits to contain all possible values.
- The 2 Most Significant Bits (MSB) are always equal (extension sign bit).
- Therefore, 2N-1 bits are enough to store the result.
- A Q15 multiplication produces Q1.30 result (extension sign bit).
- To transform the result into Q31 notation, it must be left-shifted by one bit.
- DSP processors have a special mode that allows its ALU to automatically perform the left shift when Q15xQ15.

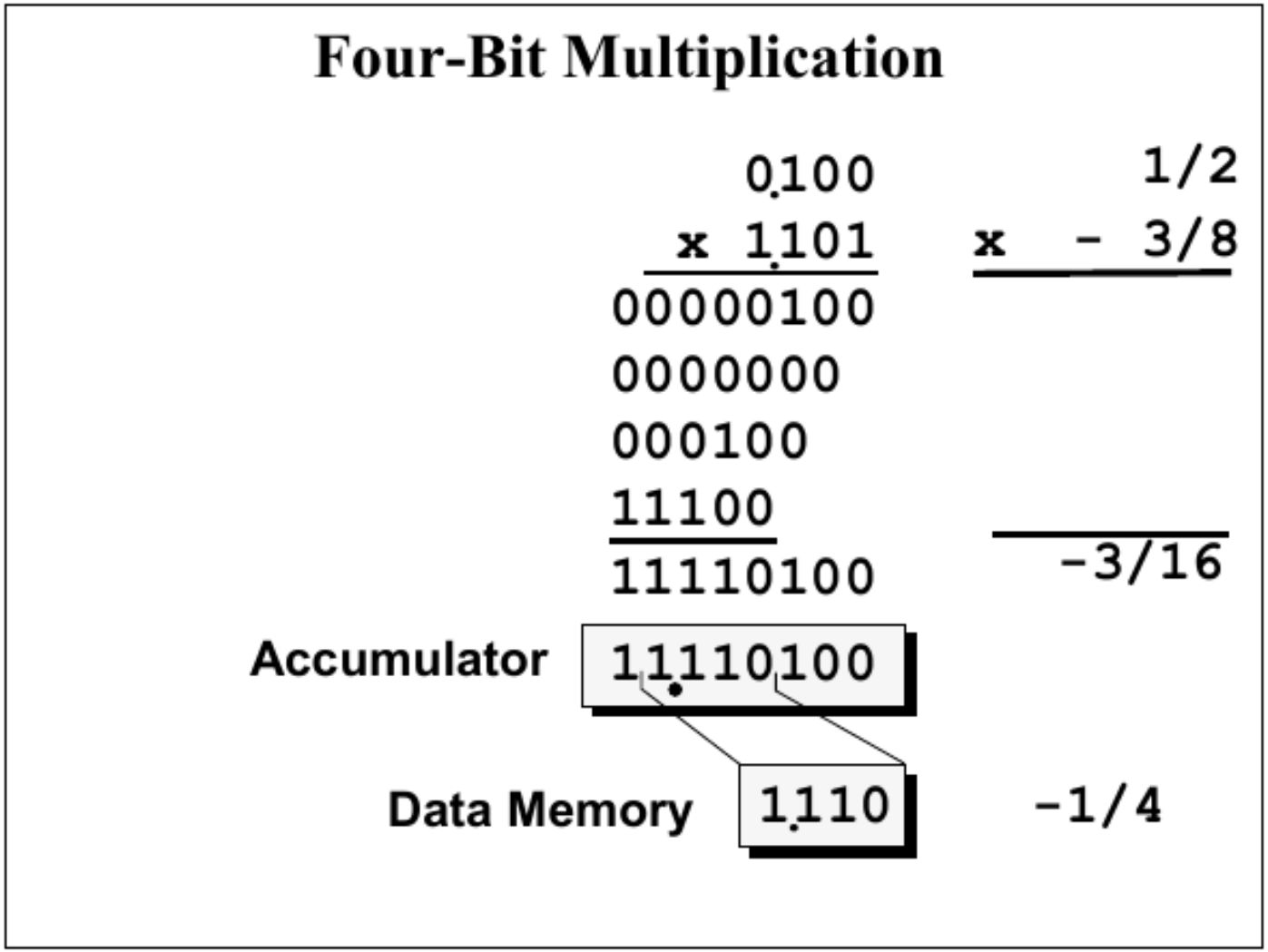


Four-Bit Integer Multiplication

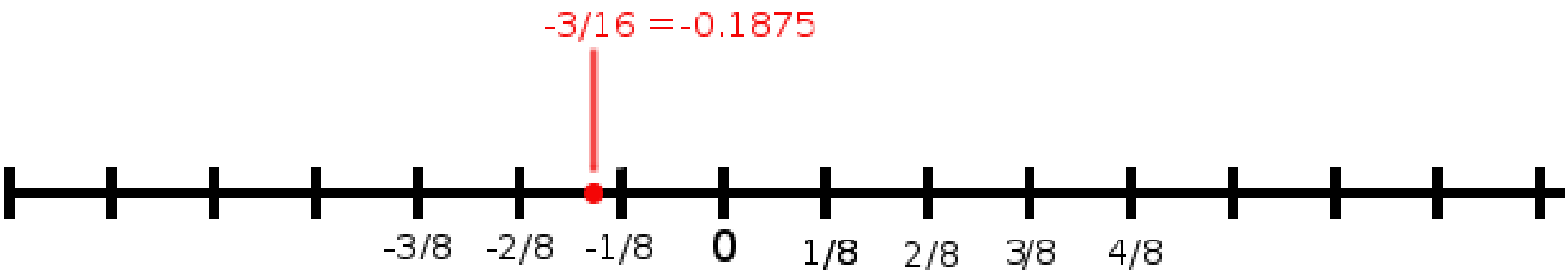
```
          0100          4
        x 1101        x -3
        --------      ----
        00000100
        0000000
        000100
        11100
        --------      ----
        11110100       -12
```

Accumulator **11110100** -12

Data Memory **11110100** -12


Four-Bit Multiplication
0.100
x 1.101
00000100
0000000
000100
11100
11110100
1/2
x - 3/8
-3/16
Accumulator
1.1110100
Data Memory
1.110
-1/4

- After multiplication, 2N bits must be stored in a memory of N-bits word.
- An **underflow** occurs if the result is less than 2^{-n}.
- **Example:** Q0.3 precision is $2^{-3} = \frac{1}{8}$.



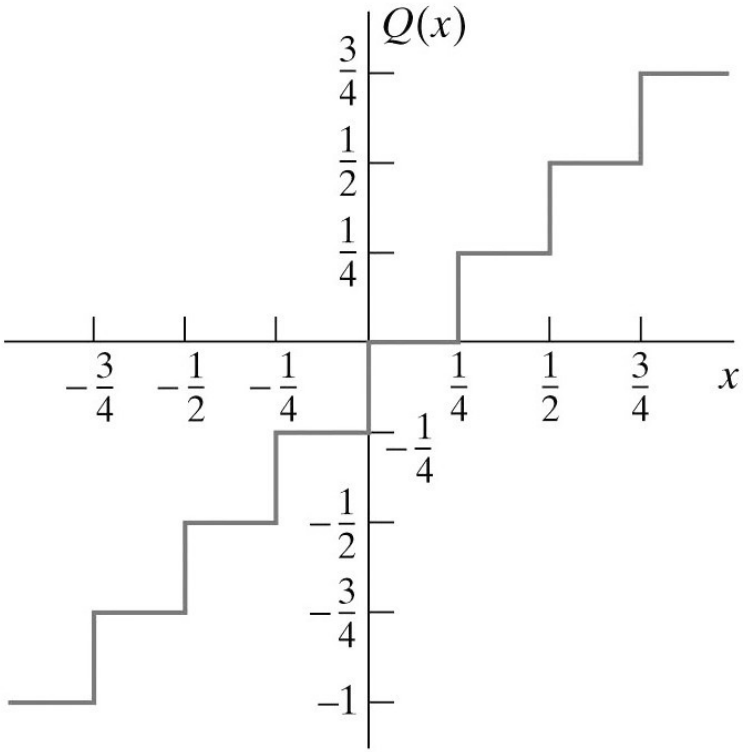
- What value should the multiplication result take? -1/8 or -2/8?
- In other words, what bits should be discarded from the multiplication result?

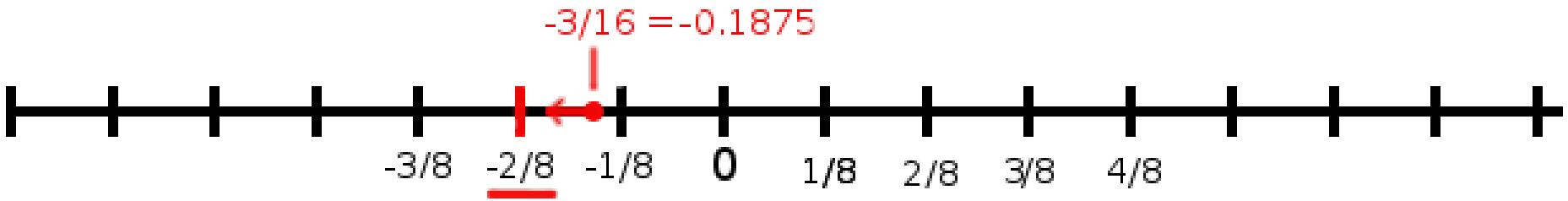
Truncation, also known as *round to minus infinity*.

- $x = a \cdot b$, $y = Q(x)$.

C code

```
1 int32_t truncation(int64_t X)
2 {
3   int32_t a;
4   a = (int32_t) (X » n);
5    return a;
6 }
```



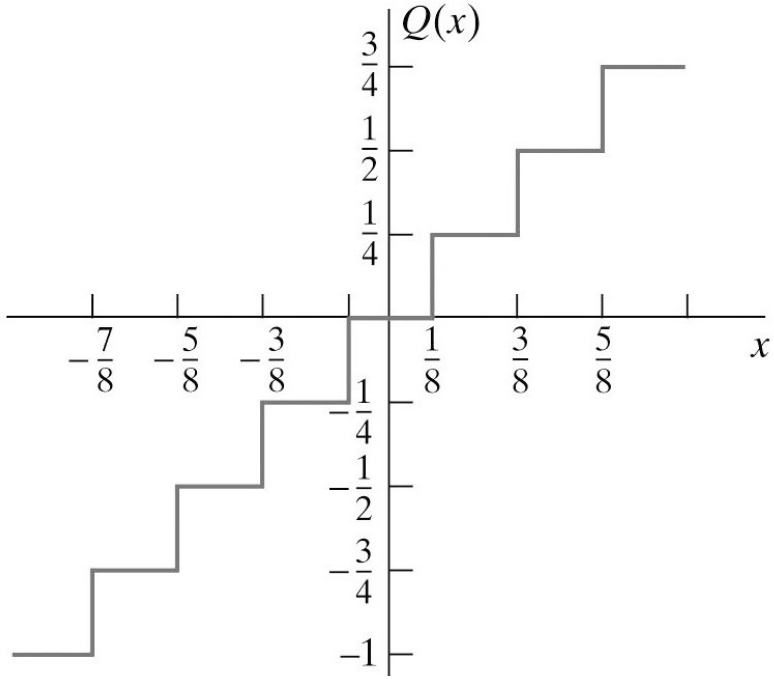


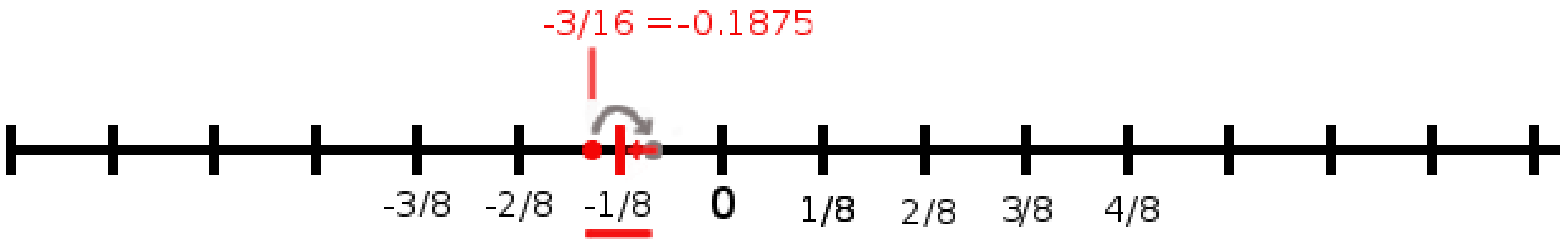
Round-off, also known as *round to the nearest*.

- $x = a \cdot b$, $y = Q\left(x + 2^{-(n+1)}\right)$, where $2^{-(n+1)} = 2^{-n}/2$, is half precision.

C code

```
1 int32_t roundoff(int64_t X)
2 {
3   int32_t a;
4   a = ( X + (1 << (n - 1) ) );
5   return truncation(a);
6 }
```

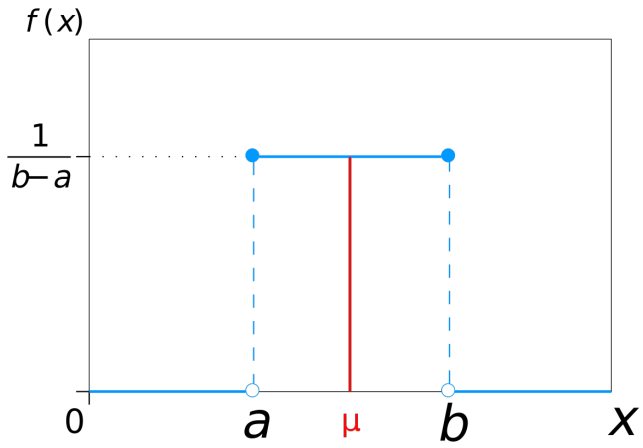




Error in rounding schemes is modeled as a uniform probability distribution.

$$\text{mean, } \mu = \frac{a+b}{2}$$

$$\text{variance, } \sigma^2 = \frac{(b-a)^2}{12}$$



- Truncation: $e = Q(x) - x, \quad -2^{-n} \leq e < 0,$ $\boxed{\mu = -\frac{2^{-n}}{2}}$ $\boxed{\sigma^2 = \frac{2^{-n}}{12}}$
- Round-off: $e = Q\left(x + 2^{-(n+1)}\right) - x, \quad -\frac{2^{-n}}{2} < e \leq \frac{2^{-n}}{2},$ $\boxed{\mu = 0}$ $\boxed{\sigma^2 = \frac{2^{-n}}{12}}$

DSP processors manage truncation and round-off automatically.

MAC operation

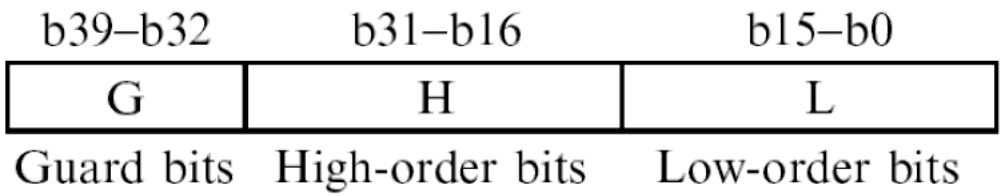
- MAC stands for **M**ultiply and **AC**cumulate.
- Since it represents the convolution operation, it is THE basic arithmetic operation in DSP.

In C:

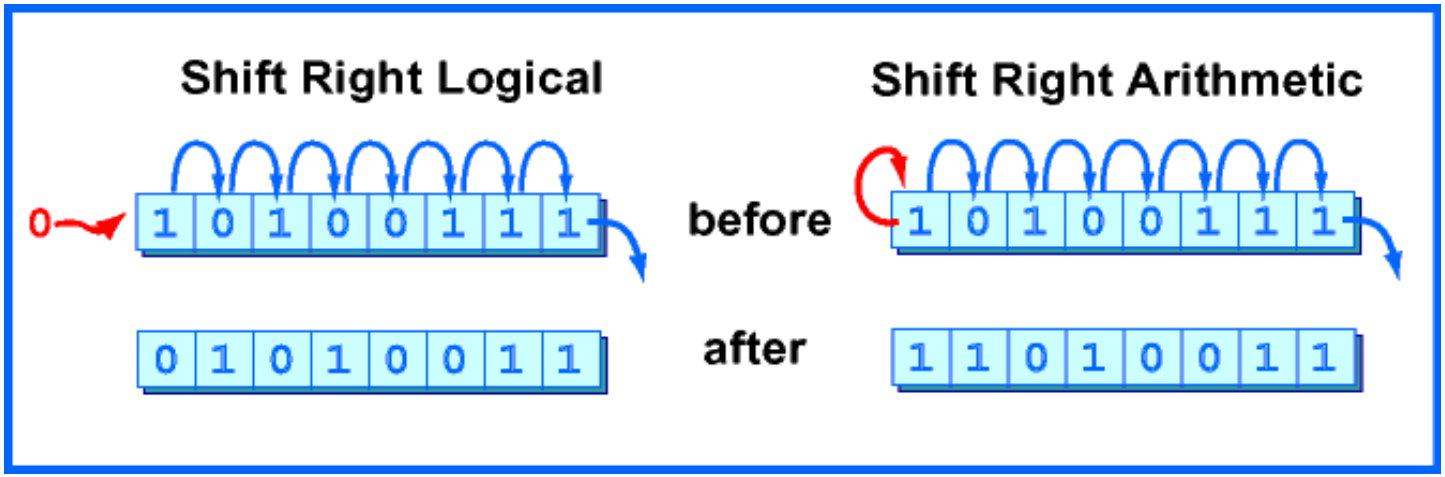
C code

```
1 int32_t a[K] ;
2 int32_t b[K] ;
3 int64_t c = 0 ;
4 for (i=0; i<K; i++)
5  { c = c + ( (int64_t) a[i] * (int64_t) b[i] ) };
```

- A MAC operation summarizes the addition and multiplication problems, i.e., overflows and underflows.
- DSP processors have an accumulator with extra bits (guard bits) to avoid overflow during internal calculations (in TI DSP C5505, 40-bits accumulator).



- Multiplication by 2: all bits are shifted left by one position.
- Division by 2: all bits are shifted right by one position (**logical shift**).
- What happens with 2-complement numbers?
- The sign bit must be preserved! (**arithmetic shift**).
- Arithmetic shift $\neq$ logical shift.



In DSP processors:

- ALU can perform logical shifts of 32-bit operands in one cycle, from 16 bits to the right, to 15 bits to the left.
- Sign extension is performed during shifts to the right, if the Sign Extension Mode control bit (in C5505, SXM) is set.
- Result is saturated during shifts to the left if an overflow is detected, and Overflow bit (in C5505, OVM) is set.

Bibliography

1. Richard G. Lyons. *Understanding Digital Signal Processing, 3rd Ed*. Prentice Hill. 2010. Chapter 12.
2. Bruno Paillard. An Introduction To Digital Signal Processors, Chapter 5.