

Finite representation of real numbers

Fixed-point arithmetic

Dr. Ing. Rodrigo Gonzalez

`rodrazalez@frm.utn.edu.ar`

`rodrigo.gonzalez@ingenieria.uncuyo.edu.ar`



UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO

- 1 **Addition**
 - Addition in 2's complement
 - Overflow
- 2 **How to avoid overflow**
 - Longer word-length accumulator
 - Saturation
- 3 **Multiplication**
 - Multiplication in 2's complement
 - Four-bit signed integer multiplication
 - Four-bit Q0.3 multiplication
 - Underflow
- 4 **How to avoid underflow**
 - Rounding schemes, Truncation
 - Rounding schemes, Round-off
 - Error in rounding schemes
- 5 **MAC operation** Operación más importante en procesamiento digital de señales
- 6 **Shifts** Desplazamientos
 - Logical and Arithmetic shifts
 - Shifts implementation in DSP processors

Addition

Addition in 2's complement

- 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.

$$\begin{array}{r} \boxed{11}111\ 111\ (\text{carry}) \\ 0000\ 1111\ (15) \\ +\ 1111\ 1011\ (-5) \\ \hline 0000\ 1010\ (10) \end{array}$$

$$\begin{array}{r} \boxed{01}11\ (\text{carry}) \\ 0111\ (7) \\ +\ 0011\ (3) \\ \hline 1010\ (-6)\ \underline{\text{invalid!}} \end{array}$$

catastrófico al dar un negativo

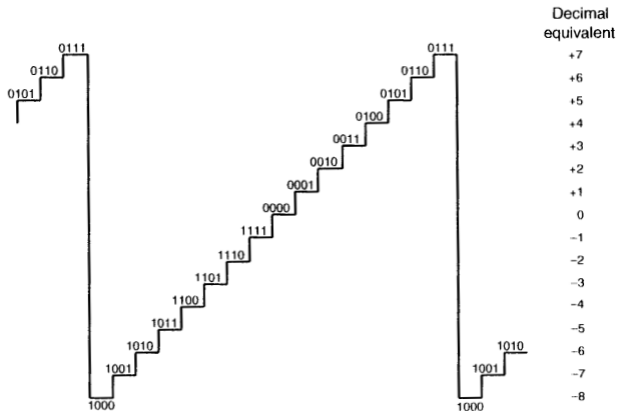
- The last two bits of the carry row show if overflow occurs.

La arquitectura de a computadora detecta overflow leyendo los ultimos 2 bits de acarreo.

Si son iguales no hubo overflow

Si son distintos hubo overflow. Se activa un bit flag

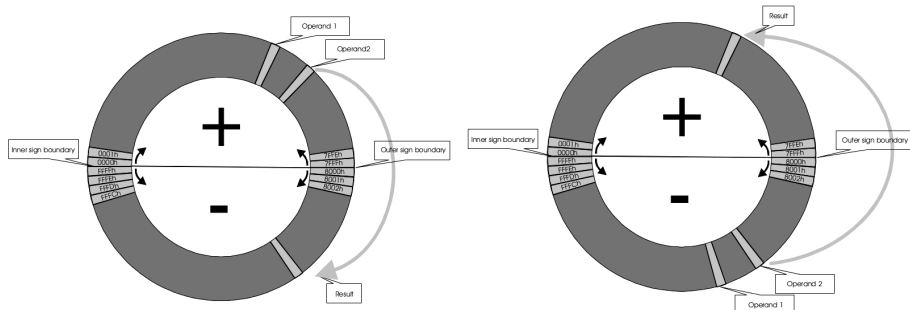
- An **overflow** occurs in a when a result is greater than $2^{N-1} - 1$ or lesser than -2^{N-1} .
- An overflow produces a **roll-over** (wrap).



Addition

Overflow

- 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.



Represento en 16 bit complemento a 2 los numeros

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?
 $16 + \log_2(s) = 40.$ $s=2^{24}$

In C:

C code

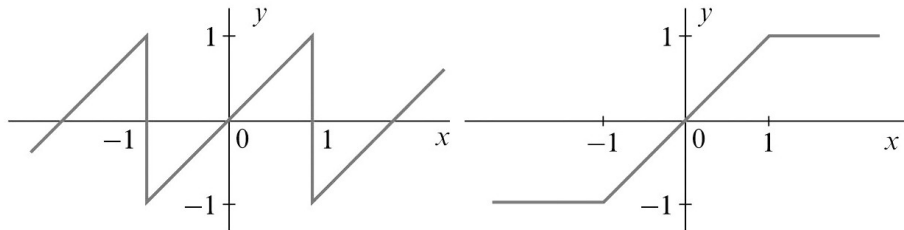
```
1 int32_t a[K] ;  
2 int64_t c = 0 ;  
3 for (i=0; i<K; i++)  
4 { c = c + (int64_t) a[i] };
```

How to avoid overflow

Saturation

- 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).

Típicamente en procesadores y PCs se aplica por software



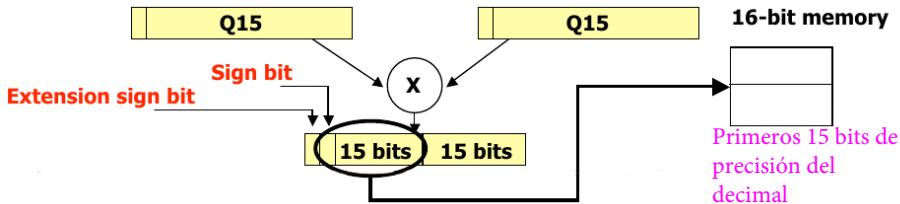
Pasamos de situación catastrófica a situación mala

Be aware of non-linearity!

Multiplication

Multiplication in 2's complement

- The product of 2 **N-bit** numbers requires **$2 \cdot N$ 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.



Multiplication

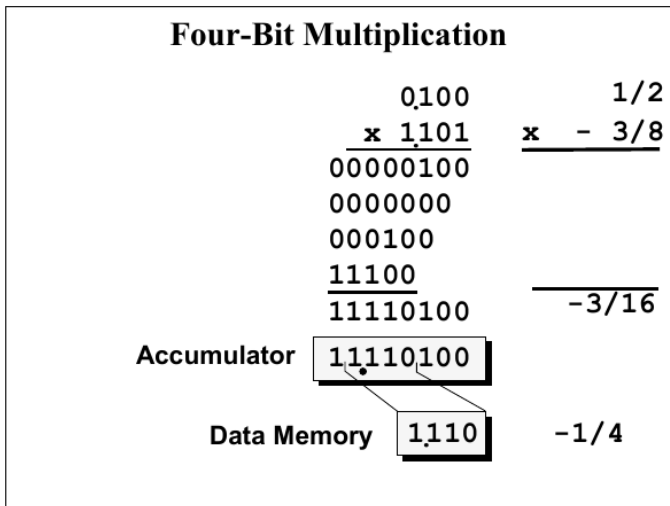
Four-bit signed integer multiplication

Four-Bit Integer Multiplication

	0100	4
	<u>x 1101</u>	<u>x -3</u>
	00000100	
	0000000	
	000100	
	<u>11100</u>	
	11110100	<u>-12</u>
Accumulator	11110100	-12
Data Memory	11110100	-12

Multiplication

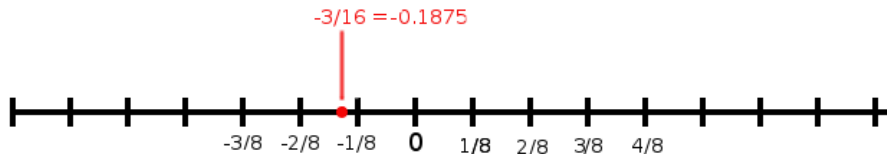
Four-bit Q0.3 multiplication



Multiplication

Underflow

- 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?

Methods in Next slide

How to avoid underflow

Rounding schemes, Truncation

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

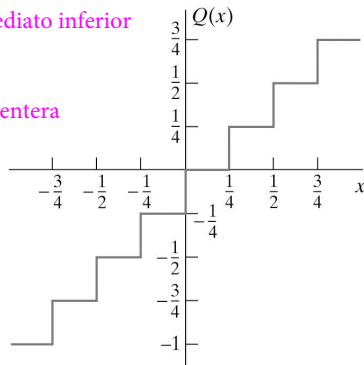
• $x = a \cdot b$, $y = Q(x)$.

Elimina bits extra generados. Devuelve valor inmediato inferior

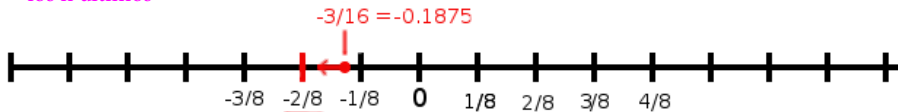
C code

Usaría `int64_t` en result para no perder parte entera

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



Como tenés $2n$ bits de decimales, eliminás los n últimos



How to avoid underflow

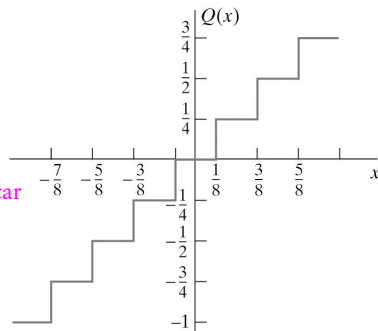
Rounding schemes, Round-off

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

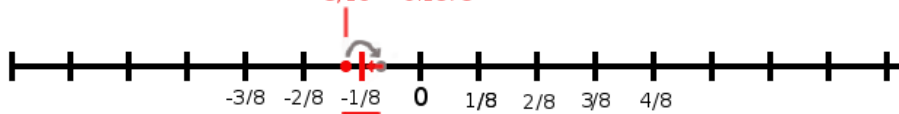
- $x = a \cdot b$, $y = Q\left(x + 2^{-(n+1)}\right)$, where $2^{-(n+1)} = 2^{-n}/2$, is half precision.

C code

```
1 int32_t roundoff(int64_t X)
2 {
3     int32_t a; Suma (2^-n)/2 para truncar
4     a = ( X + (1 « (n - 1) ) );
5     return truncation(a);
6 }
```



$-3/16 = -0.1875$



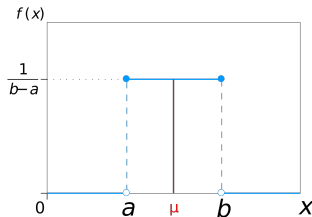
How to avoid underflow

Error in rounding schemes

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: $e = Q(x) - x$, $-2^{-n} \leq e < 0$, $\mu = -\frac{2^{-n}}{2}$, $\sigma^2 = \frac{2^{-n}}{12}$

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

En promedio da menos error

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:

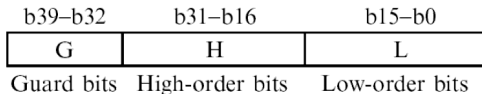
1 ciclo de reloj

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).

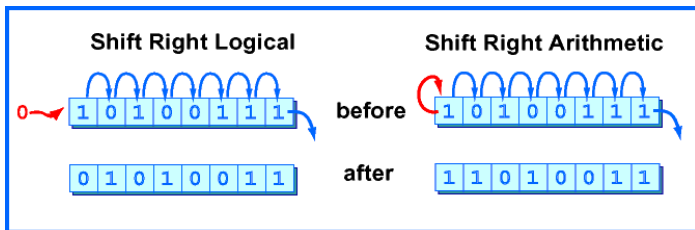
Dijo profe que también
underflows



Shifts

Logical and Arithmetic shifts

- 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? *Cambiaría de signo. MAL*
- The sign bit must be preserved! (**arithmetic shift**).
- Arithmetic shift \neq logical shift.



Se vuelve a colocar el numero inicial

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.

- 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.