## Binary Rounding

## Rounding

- Situation:

Number lies between two representable values

- Question:

What rounding choices do we have?

## Rounding - Choices



## Rounding - Choices

### 1.001...001|0011...


1.001...001
binary

## Rounding - Choices

just cut this off

## $1.001 \ldots . .001 \left\lvert\, \begin{array}{ll}1011 \\ \text { binina }\end{array}\right.$


1.001...001
binary

## Rounding - Choices

### 1.001...001|0011...


$\underbrace{\substack{\text {. } \\ \hline}}_{\substack{\text { Rounting } \\ \text { Up }}}$
1.001...001
1.001...010|

## Rounding - Choices

### 1.001...001|0011...



Rounding
Up
1.001...010|

## Rounding - Choices

cut this off
1.001...001|登11


## Rounding - Choices

## AND add +1 here <br> cut this off <br> $1.001 \ldots 001 \mid{ }^{0011_{\text {binand }}}$



## Which when?

- Question:

When does the computer round up/down?

- Rule:

Computer rounds to the closest representable number.

## Which when?

- Case 1:
next bit is 0
$\rightarrow$ round down


## Case 1 - Example

### 1.001...001|0011..

## Case 1 - Example

next bit is 0
1.001...0010011...

## Case 1 - Example



## Which when?

- Case 1:
next bit is 0
$\rightarrow$ round down
- Case 2:
next bit is 1 AND
at least one later bit is $\mathbf{1}$ again
$\rightarrow$ round up


## Case 2 - Example

### 1.001...001|1000010...

## Case 2 - Example

next bit is 1


## Case 2 - Example



## Which when?

- Case 1:
next bit is 0

```
round down
```

- Case 2:
next bit is 1 AND
at least one later bit is $\mathbf{1}$ again $\rightarrow$ round up
- Case 3:
next bit is 1 AND
all the following bits are $\mathbf{0}$ $\rightarrow$ round so that last bit is $0\left(^{*}\right)$
(*) Round-to-Even Rule


## Case 3 - Example 1

### 1.001...001|1000...

binary

## Case 3 - Example 1



## Case 3 - Example 1


this should be 0 after rounding

## Case 3 - Example 1



## Case 3 - Example 2

### 1.001...000|1000...

binary

## Case 3 - Example 2



## Case 3 - Example 2


this should be 0 after rounding

## Case 3 - Example 2



## Why Round-to-Even?

- Reason:

Round down in approx. 50\% of all times and round up in the remaining $50 \%$.

## Why Round-to-Even?

-Reason:
Round down in approx. 50\% of all times and round up in the remaining $50 \%$.

- Imagine:

Sum of 1'000'000 numbers...

## Why Round-to-Even?

- Reason:

Round down in approx. 50\% of all times and round up in the remaining $50 \%$.

- Imagine:

Sum of 1'000'000 numbers...
Always round down in Case 3
$\rightarrow$ underestimation

## Why Round-to-Even?

- Reason:

Round down in approx. 50\% of all times and round up in the remaining $50 \%$.

- Imagine:

Sum of 1'000'000 numbers...
Always round down in Case 3
$\rightarrow$ underestimation
Always round up in Case 3
$\rightarrow$ overestimation

