26th IEEE Symposium on Computer Arithmetic

ARITH 26

Kyoto, Japan June 10th-12th, 2019

Reproducible Summation under HUB Format

Prof. Dr. Julio Villalba-Moreno
Dept. Computer Architecture
University of Malaga
SPAIN

Talk Outline

- Posing the problem
- HUB format
  - Definition
  - Floating point HUB numbers
  - Advantages and drawbacks
- Reproducible summation under HUB
  - Error free vector transformation
  - Splitting a HUB number
  - Architectures
- Summary and conclusion

Posing the problem

Posing the problem

- Floating point addition reproducibility
  - Associative law is not met
    - Result can depend on the order of the operands

      $A+B+C+D \neq B+A+D+C$

      - Dynamic scheduling on parallel computing processors

- Solutions
  - Deterministic parallel tree scheme, extra precision
    - High computing time, communication overhead
  - Pre-rounding technique (Rump, Ogita, Oishi 2008[*])
    - Minimize

[*] S.M. Rump, T. Ogita, and S. Oishi. Accurate floating-point summation part I: Faithful rounding. SIAM J. Sci. Comput., 31(1):189–224, 2008.

Posing the problem

- Pre-rounding technique (Rump, Ogita, Oishi 2008[*])
  - Error-free vector transformation (Rump 2008)
    - Accurate floating-point summation
  - Pre-rounding of the data to a common base according to a specific boundary
    - Accurate sum
    - No rounding error

- ***We adapt this method to HUB representation***

  →***Reformulation of the fundamentals***

  → ***New hardware architecture***

[*] S.M. Rump, T. Ogita, and S. Oishi. Accurate floating-point summation part I: Faithful rounding. SIAM J. Sci. Comput., 31(1):189–224, 2008.

HUB format

HUB format

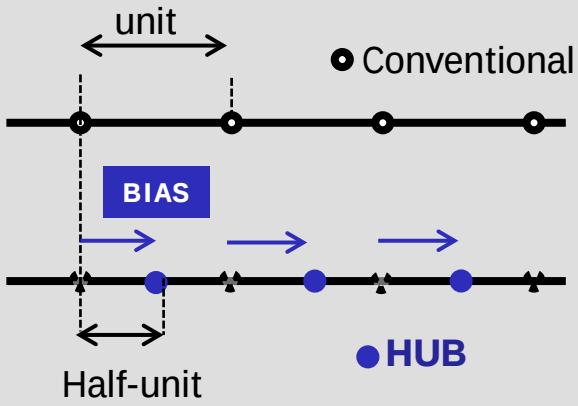
- **HUB** = **H**alf-**U**nit **B**iased format

Digit-vector

$$X = (X_{n-1}, X_{n-2}, \cdots X_1, X_0, X_{-1}, \cdots, X_{-f})$$

Conventional number in radix β

$$X = \left[\sum_{i=-f}^{n-1} X_i \cdot \beta^i \right]$$

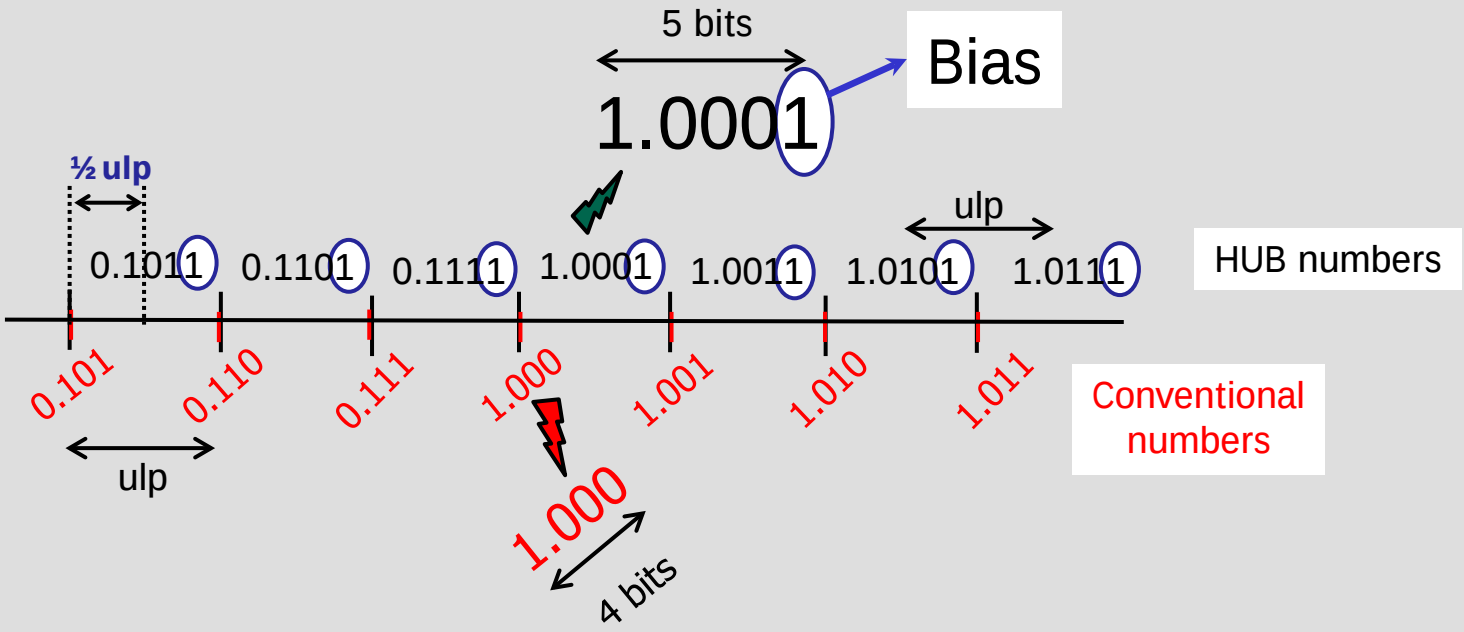


HUB number in radix β

$$X = \left[\sum_{i=-f}^{n-1} X_i \cdot \beta^i \right] + \underbrace{\frac{\beta}{2} \cdot \beta^{-f-1}}_{\text{BIAS}}$$

HUB format

- Example (4 bits, radix 2)



HUB format

- The extra LSB → ILSB Implicit Least Significant Bit

x x x . x x x x x x x x x x x 1

ILSB

  - Value: 1
    - ➢ Implicit bit (ILSB)
      - Not stored
      - Not transmitted
      - Not needed for representation
    - ➢ Required when operating

HUB format

- Floating point HUB number in radix-2 *(β=2)*

(S_x, M_x, E_x)

E_x → Exponent (conventional)

M_x → Significand (HUB magnitude)

S_x → Sign (conventional)

– Normalized HUB significand: $1<M_x<2$

Digit-vector

$$M_x = (M_{x_0}, M_{x_{-1}}, M_{x_{-2}}, \cdots, M_{x_{-f}})$$

$$M_x = \left[\sum_{i=0}^{f} M_{x_i} \cdot 2^{-i}\right] + 2^{-f-1}$$

BIAS

$$M_x = 1.M_{x_{-1}} M_{x_{-2}} \cdots M_{x_{-f}} 1$$

HUB format

- Normalized Floating point HUB number in radix-2

$$M_x = \left[\sum_{i=0}^{f} M_{x_i} \cdot 2^{-i} \right] + 2^{-f-1}$$

ILSB

$M_x = 1.M_{x_{-1}} M_{x_{-2}} \cdots M_{x_{-f}}$

Representative form

Digit-vector

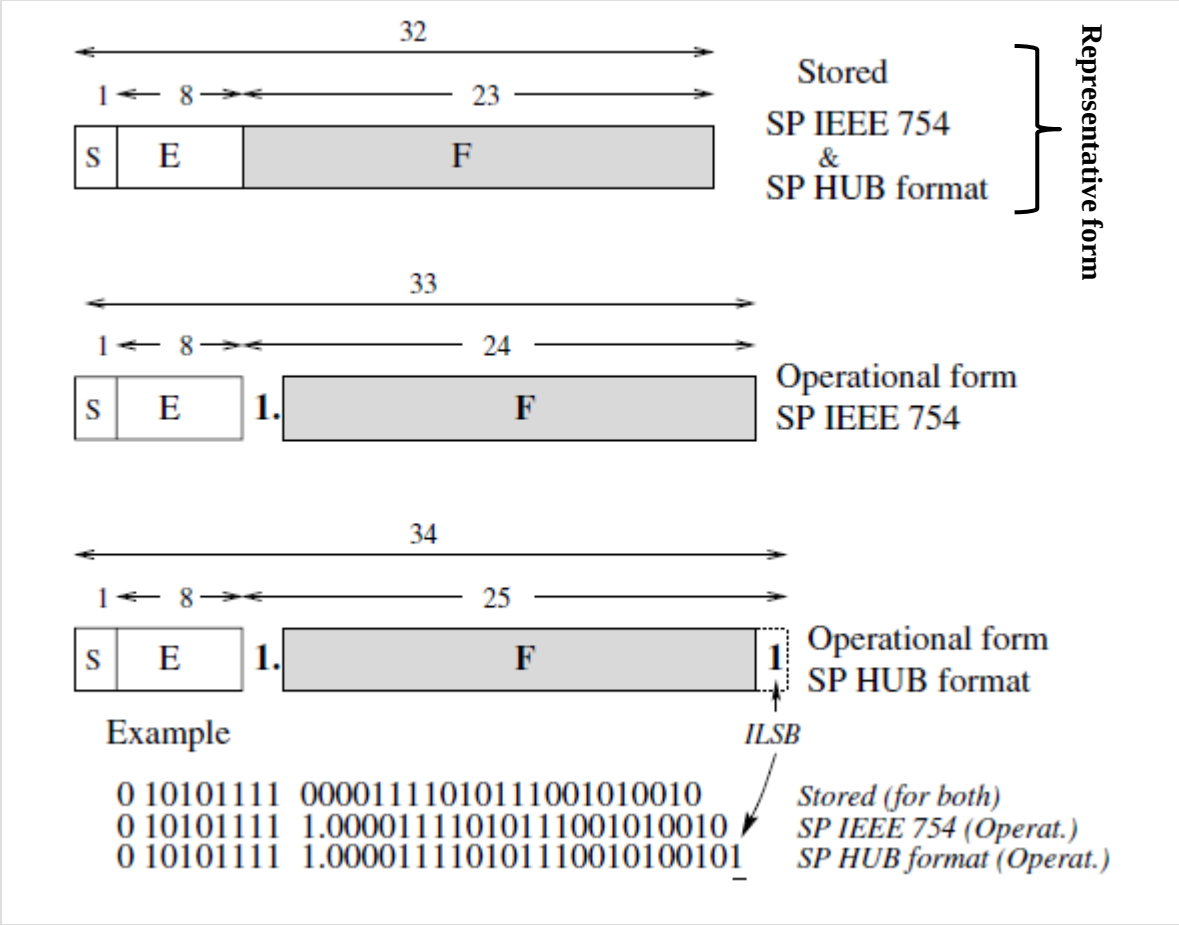
$M_x = 1.M_{x_{-1}} M_{x_{-2}} \cdots M_{x_{-f}} 1$

Operational form

To operate

HUB format

Single precision (SP) IEEE-754 and its HUB counterpart



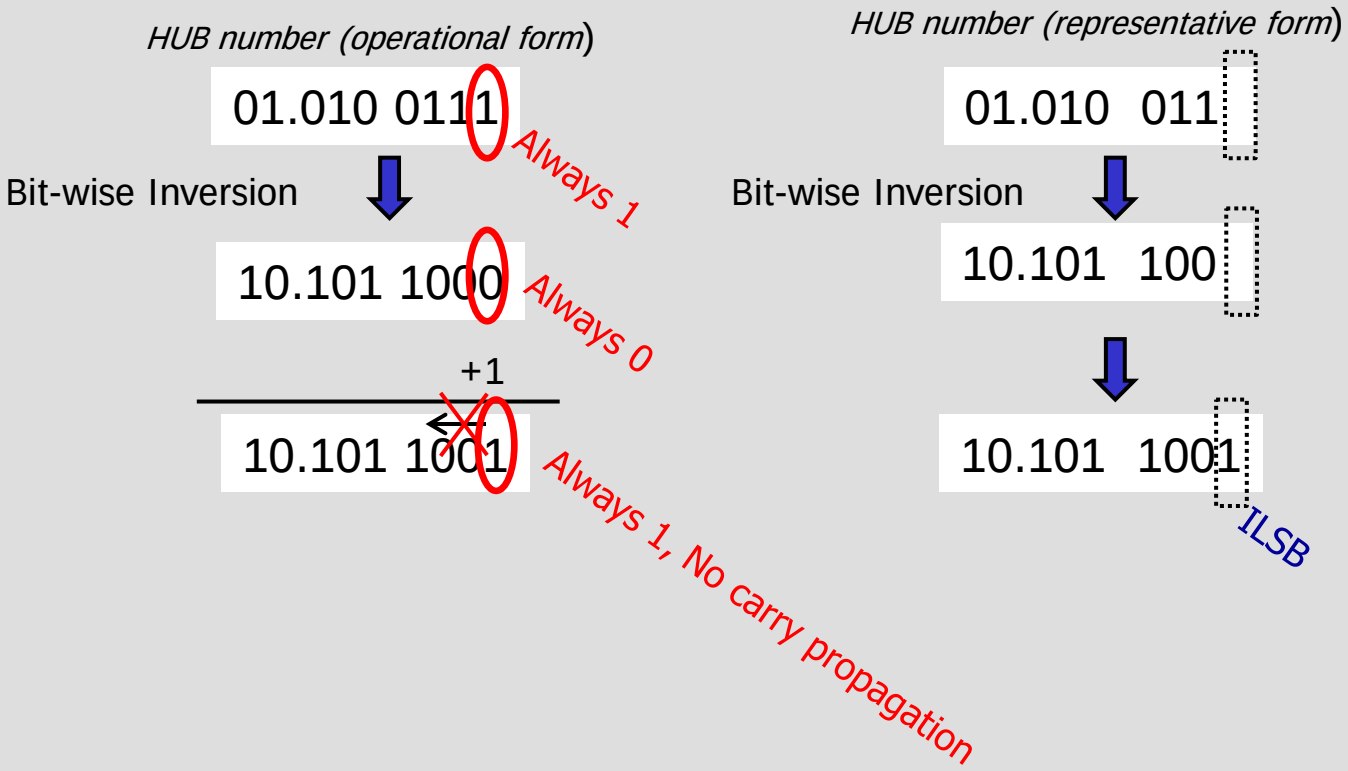
HUB format

- Advantages
  - Two's complement → bit-wise
  - Round to nearest → by truncation
  - No double rounding error
  - Rounding bit not required (round to nearest)
  - In general, it simplifies the underlying hardware

- Drawbacks
  - Not valid for integers
  - Other rounding modes involve carry propagation
  - Not IEEE compliance

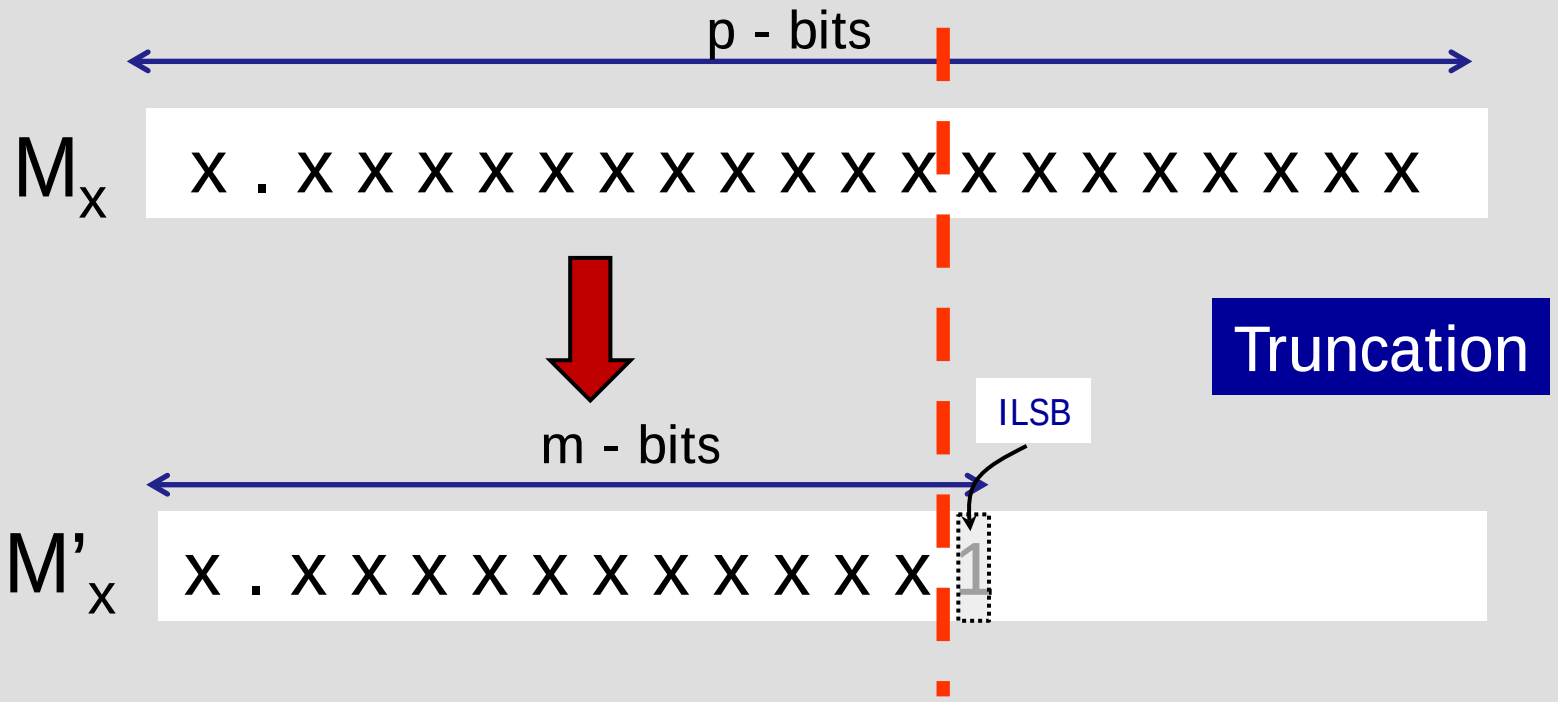
HUB format

- Two's complement of a HUB number
  - Invert the bits of the representative form (bit-wise)
    - » The ILSB=1 → No carry propagation



HUB format

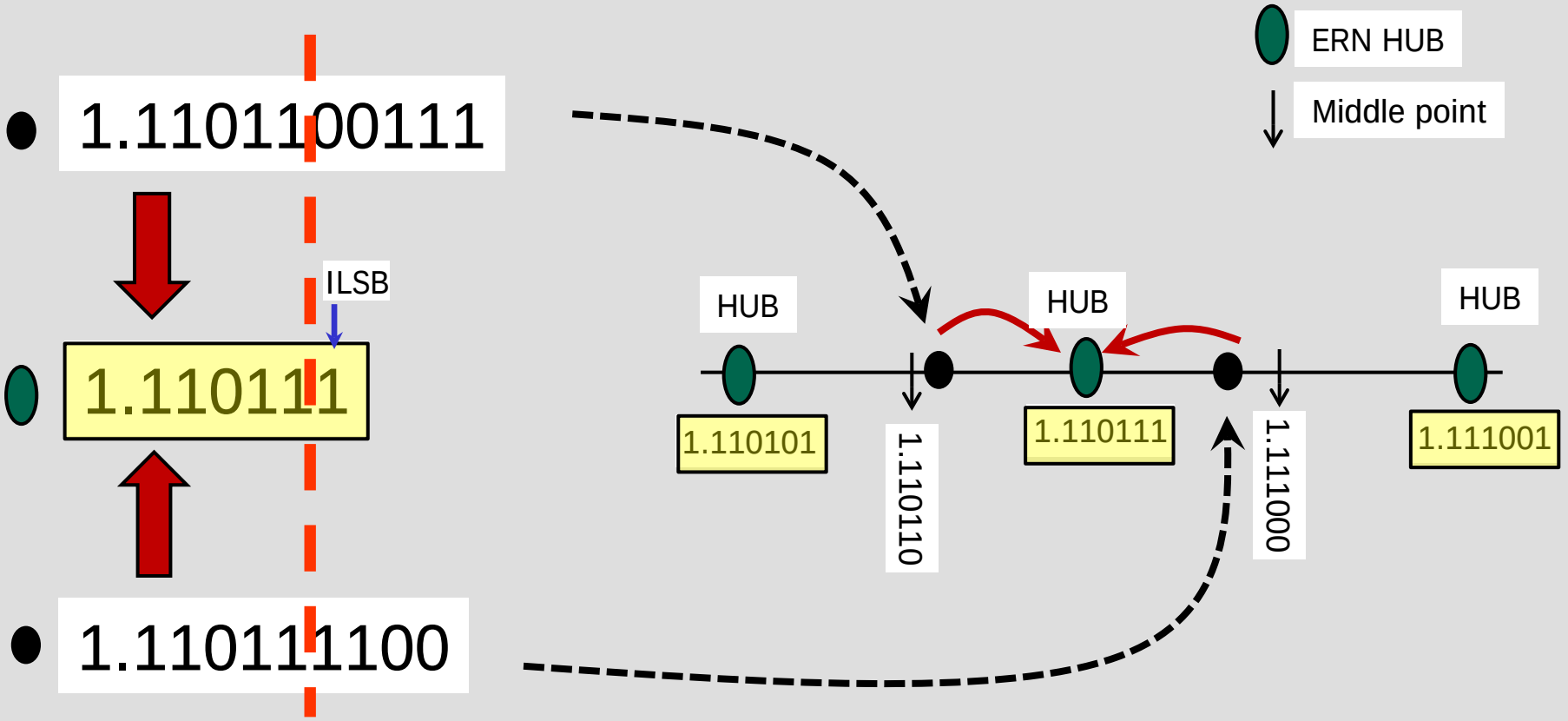
- Round to nearest of a HUB number: by truncation



Formally: M’[0:m-2]=M[0:m-2]

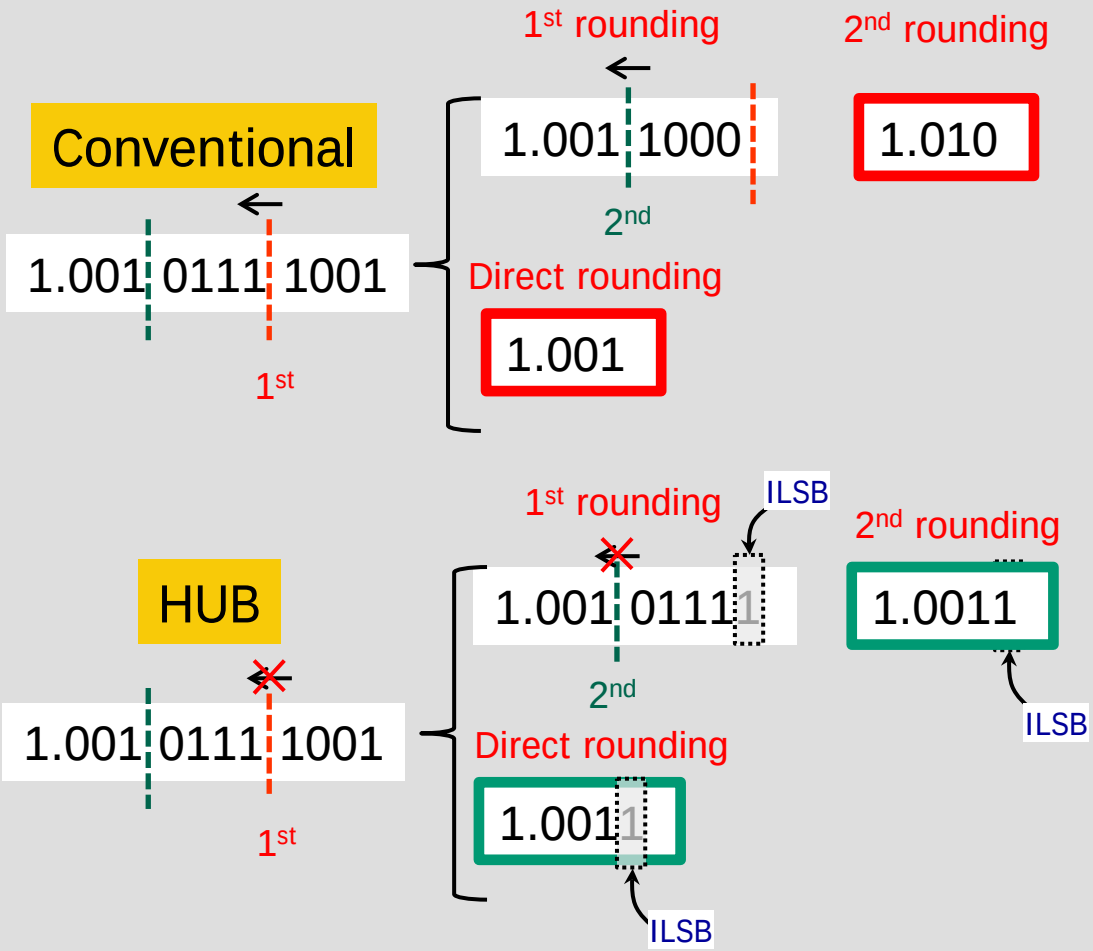
HUB format

- Proposed rounding: round to nearest by truncation



HUB format

- No double rounding error for HUB numbers
  - Truncation avoids the double rounding error



Floating point

- Efficiency of HUB format
  - Floating-point
    - Adder :
      - » Speed-up: 14%
      - » Area reduction: 38%
      - » Power reduction: 25% (single), 15% (double)
    - Multiplier:
      - » Speedup: 17%
      - » Area reduction: 22%
      - » Power reduction: 2% (single), -2.6% (double)
    - Division & Square root:
      - » Same hardware and number of iterations as conventional
      - » Less complex On-the-fly conversion

Resproducible summation under HUB

Reproducible summation

- Representing a normalized FP HUB number

Normalized HUB number, p-bit precision

$x^{hub}=(s,m_{hub},e)$

$$x^{hub} = (-1)^s m_{hub} 2^e$$

$1<m_{hub}<2$

$$m_{hub} = \left[\mu + \frac{1}{2}\right] 2^{-(p-1)}$$

$$2^{p-1} \leq \mu < 2^p$$

μ → integer

Reproducible summation

- Representing a normalized FP HUB number

$$x^{hub} = (-1)^s \left[\mu + \frac{1}{2}\right] 2^{e-(p-1)} = (-1)^s \left[\sum_{i=0}^{p-1} \mu_i 2^i + \frac{1}{2}\right] 2^{e-(p-1)}$$

$$x^{hub} \rightarrow \underbrace{1\ \mu_{p-2}\mu_{p-3}\cdots\ \mu_1\ \mu_0}.\underbrace{1} \times 2^{-(p-1)} \times 2^e$$

μ

$\frac{1}{2}$ *(ILSB)*

Reproducible summation

- Rounding to nearest a normalized non HUB number

$$y = (s, \mu, e) \quad \textit{non HUB number}$$

$$\textit{HUB rounding} \downarrow$$

$$R^{hub}(y) = (s, \lfloor \mu \rfloor, e) \quad \textit{HUB number}$$

$$2^{p-1} \leq \mu < 2^{p}$$

$$1\ \mu_{p-2}\mu_{p-3} \cdots \mu_1\ \mu_0 . \mu_{-1}\mu_{-2} \cdots \mu_{-f} \quad \times 2^{-(p-1)} \leftarrow m_y$$

$$\textit{HUB rounding} \downarrow$$

$$1\ \mu_{p-2}\mu_{p-3} \cdots \mu_1\ \mu_0 . 1 \quad \times 2^{-(p-1)} \leftarrow m_{y'hub}$$

Reproducible summation

– Error-free vector transformation (Rump 2008 [*], used by Demel 2015[**])

$v = \{v_1, v_1, ..., v_n\}$ (n floating point numbers, p bit precision, $\epsilon = 2^{-p}$)

$$
\begin{aligned}
&1: \ m = max(|v_i|) \\
&2: \ \delta = fl(n * m/(1 - 2n\epsilon)) \\
&3: \ M = 2^{\log_2(\delta)} \\
&4: \ T = 0 \\
&5: \ \textit{for } i = 1 \textit{ to } n \textit{ in any order do} \\
&6: \quad q_i = fl(fl(M + v_i) - M) \\
&7: \quad T = fl(T + q_i) \\
&8: \ \textit{end for}
\end{aligned}
$$

← Upper part

[*] S.M. Rump, T. Ogita, and S. Oishi. Accurate floating-point summation part I: Faithful rounding. SIAM J. Sci. Comput., 31(1):189–224, 2008.

[**] J. Demmel and H. D. Nguyen, Parallel Reproducible Summation," IEEE Transactions on Computers, vol. 64, no. 7, pp. 2060–2070, July 2015

Reproducible summation

- Pre-rounding for HUB fundamentals

Lemma I. *Let Q, x, y be three standard floating point numbers with a significand of p bits such that x and y are multiple of $\frac{1}{2}ulp(Q)$. If $|x+y| < \epsilon^{-1}\frac{1}{2}ulp(Q)$ then $x+y$ can be represented as an exact p-bit signficand floating point number and $x+y$ can be exactly computed.*

Q, x, y → standard p-bit FP

x = k·½ulp(Q)

y = k'·½ulp(Q)

$|x+y| < \epsilon^{-1}\cdot$½ulp(Q)

→ x+y → Exact FP, Exactly computed

Reproducible summation

– Pre-rounding for HUB fundamentals

Lemma I. *Let Q, x, y be three standard floating point numbers with a significand of p bits such that x and y are multiple of $\frac{1}{2}ulp(Q)$. If $|x+y| < \epsilon^{-1}\frac{1}{2}ulp(Q)$ then x+y can be represented as an exact p-bit signficand floating point number and $x+y$ can be exactly computed.*

Example for 24 bit precision

Q, x, y → standard 24-bit FP

Q=1, ulp(Q)=2^{-23}, $\epsilon=2^{-24}$

x = k·2^{-24}

y = k·2^{-24}

|x+y| < 1

→ x+y → Exact FP, Exactly computed

Reproducible summation

– Splitting the HUB number in 2 independent upper and lower parts

Theorem I *Let $Q = 2^k$ and consider its standard normalized floating point representation with a significand of p bits. Let $ulp(Q)$ denote the unit in the last place of the standard floating point representation of Q, that is $ulp(Q) = 2^{k-(p-1)}$. Let x^{hub} be a normalized HUB number $(x^{hub} \equiv (s, \mu, e))$, with $Q > x^{hub}$. If we define q and r as the upper and lower part respectively of x^{hub} such that:*

$$x^{hub} = q + r \tag{13}$$

where q is

$$q = \left(\lfloor \mu 2^{e-k} \rfloor + \frac{1}{2} \right) 2^{k-(p-1)} \tag{14}$$

Upper part

$$q = l\frac{1}{2}ulp(Q) \quad l \in \mathbb{N}$$

$$r < \frac{1}{2}ulp(Q)$$

Lower part

- Upper (***q***) and lower (***r***) parts are disjoint
- No overlap between them

{q_i} Fulfil Lemma 1 →

The summation of the upper parts of a set of HUB numbers is reproducible

Reproducible summation

– Splitting the HUB number in 2 independent upper and lower parts

$$v^{hub} = [v_1^{hub}, ..., v_n^{hub}] \xrightarrow{\text{Theorem I}} v_i^{hub} = q_i + r_i, \forall i \longrightarrow \{q_1, ..., q_n\}$$

$$Q = 2^k \quad Q > q_i, \forall i$$

Lemma I:

$$|\sum_{i=1}^{n} q_i| < \epsilon^{-1} \frac{1}{2} ulp(Q)$$

$$k = \left\lceil \log_2 \frac{n|v_i^{hub}|_{max}}{1 - n\epsilon} \right\rceil$$

- This value of k ensures that Lemma I is accomplished and then the summation is reproducible

Reproducible summation

– Algorithm for HUB representation

$$v^{hub} = [v_1^{hub}, \ldots, v_n^{hub}]$$ (n floating point numbers, p bit precision)

1: Calculate $|v_i^{hub}|_{max}$

2: $k = \lceil \log_2(n\,|v_i^{hub}|_{max}/(1-n\epsilon)) \rceil$

3: $Q = 2^k$

4: $T = 0$

5: for *i=0* to *n* in any order do

6: $q_i = (\lfloor \mu 2^{e-k} \rfloor + ½)2^{k-(p-1)}$ ← Upper part

7: $T = T + q_i$

Reproducible summation

– Accuracy

- Depends on the commom base Q
- The smaller Q → the more accurate final sum
- Maximum absolute error:

$$A\_error < n \cdot \frac{1}{2} ulp(Q)$$

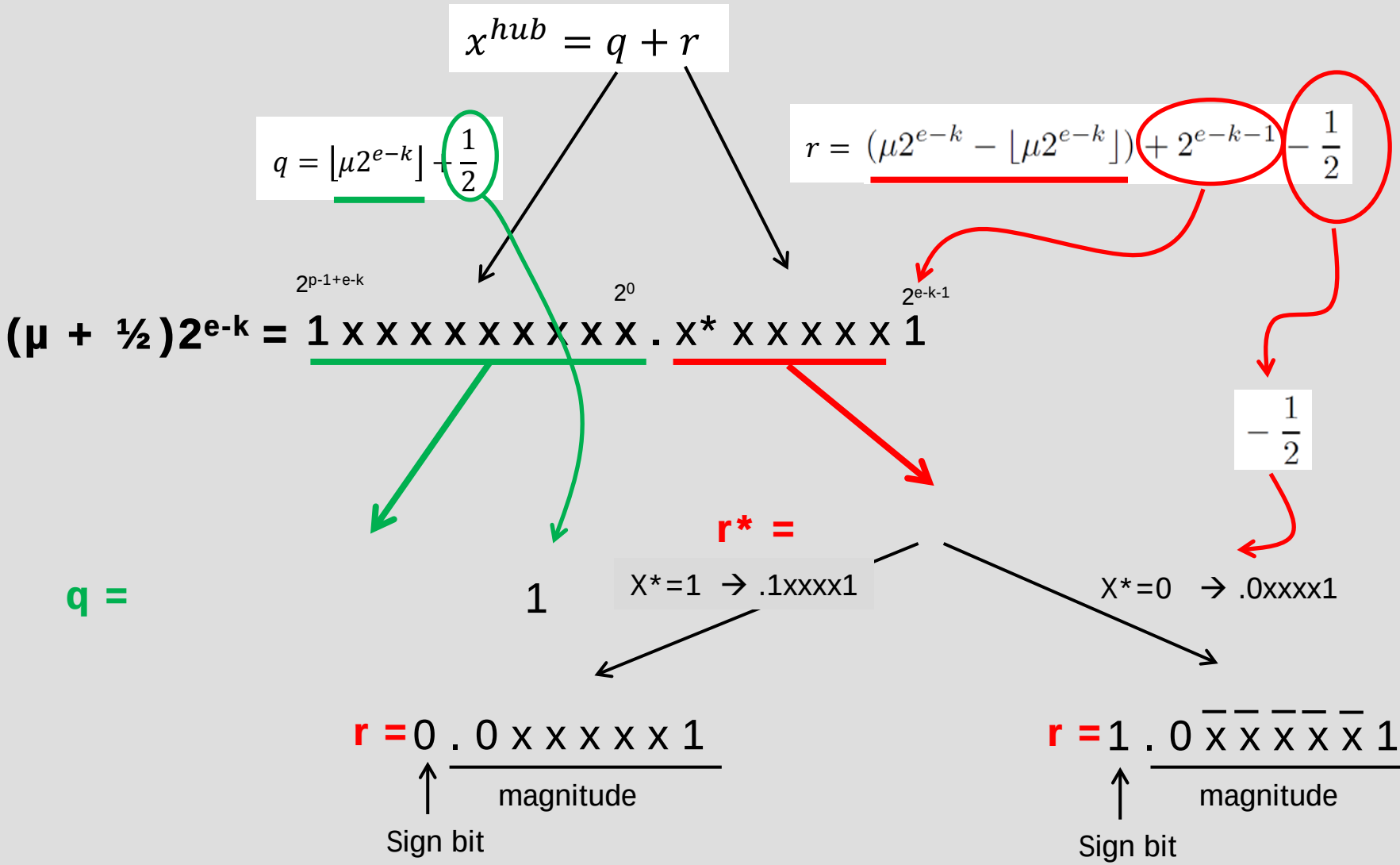
Taking into account that $ulp(Q) = 2^{k-(p-1)}$ we have

$$A\_error < n \cdot 2^{k-p}$$

Architecture

- Architectures for splitting a HUB number
  - Direct manipulation of the significand (pre-rounding unit)
  - Using HUB/Standard adders

Architecture



Architecture

– Pre-rounding Unit (direct manipulation of the significands)

$x_1^{hub} = 1100\ 1101\ 0011\ 1001\ 1 \cdot 2^{-5}$ $x_2^{hub} = 1000\ 0011\ 1001\ 1110\ 1 \cdot 2^{-4}$

$x_3^{hub} = 1001\ 0000\ 1100\ 1101\ 1 \cdot 2^{-7}$ $x_4^{hub} = 1001\ 1011\ 0001\ 0000\ 1 \cdot 2^{-4}$

$Q \rightarrow$ 1000 0000 0000 0000

$\mu_1^{hub} + ½ \rightarrow$ 1100 1101 0011 . 1001 1

$\mu_2^{hub} + ½ \rightarrow$ 1 0000 01110 011 . 1101

$\mu_3^{hub} + ½ \rightarrow$ 10 0100 0011 . 0011 01 1

$\mu_4^{hub} + ½ \rightarrow$ 1 0011 01100 010 . 0001

q_i r_i

Architecture

– Pre-rounding Unit (direct manipulation of the significands)

$x_1^{hub} = 1100\ 1101\ 0011\ 1001\ 1 \cdot 2^{-5}$ $x_2^{hub} = 1000\ 0011\ 1001\ 1110\ 1 \cdot 2^{-4}$

$x_3^{hub} = 1001\ 0000\ 1100\ 1101\ 1 \cdot 2^{-7}$ $x_4^{hub} = 1001\ 1011\ 0001\ 0000\ 1 \cdot 2^{-4}$

Q →	1000 0000 0000 0000	
$\lfloor \mu_1^{hub} \rfloor + ½$ →	1100 1101 0011	.1001 1
$\lfloor \mu_1^{hub} \rfloor + ½$ →	1 0000 01110 011	.1101
$\lfloor \mu_1^{hub} \rfloor + ½$ →	10 0100 0011	.0011 01 1
$\lfloor \mu_1^{hub} \rfloor + ½$ →	1 0011 01100 010	.0001
	q_i	r_i

Architecture

– Pre-rounding Unit (direct manipulation of the significands)

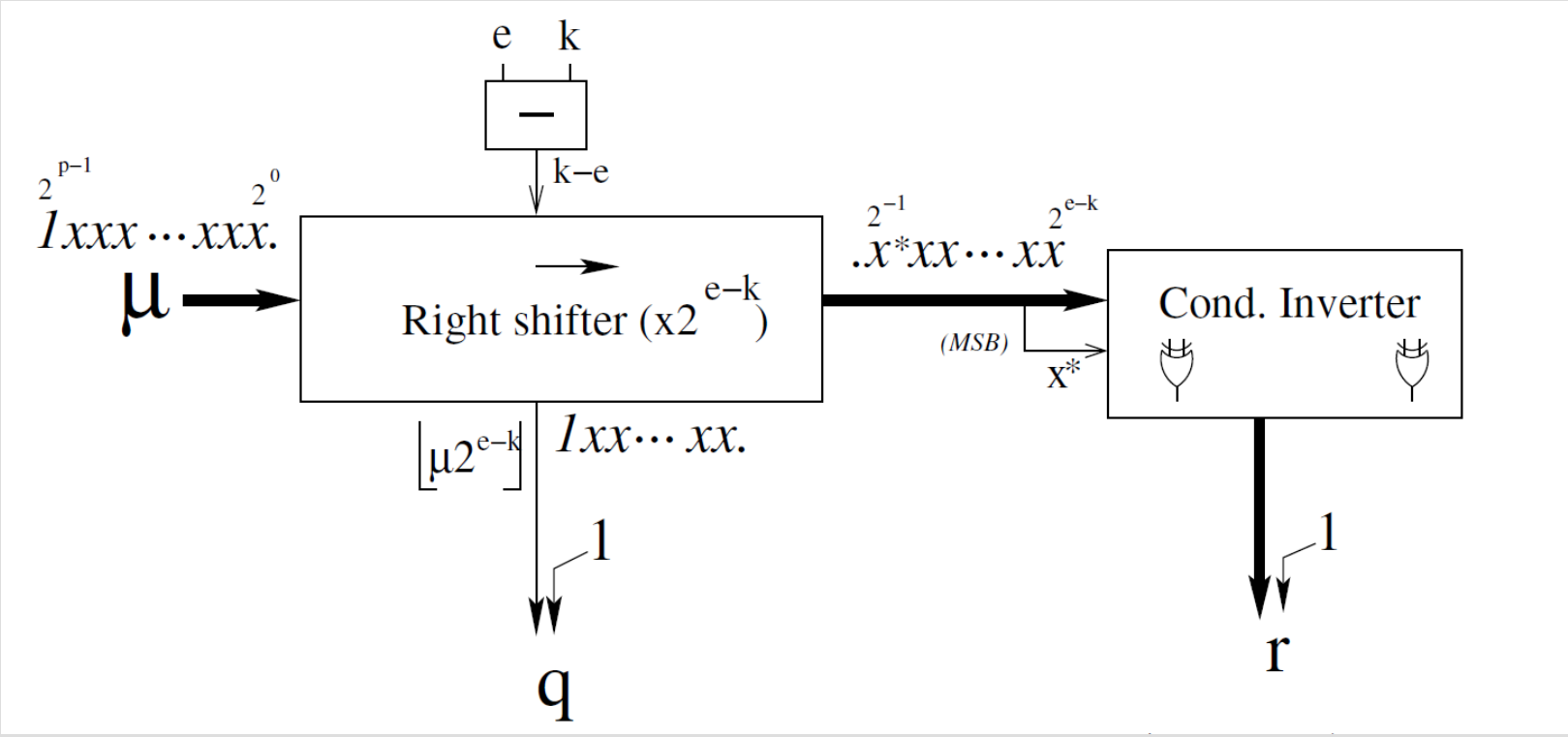
$x_1^{hub} = 1100\ 1101\ 0011\ 1001\ 1 \cdot 2^{-5}$ $x_2^{hub} = 1000\ 0011\ 1001\ 1110\ 1 \cdot 2^{-4}$

$x_3^{hub} = 1001\ 0000\ 1100\ 1101\ 1 \cdot 2^{-7}$ $x_4^{hub} = 1001\ 1011\ 0001\ 0000\ 1 \cdot 2^{-4}$

		sign ↓
Q →	1000 0000 0000 0000	
$\lfloor \mu_1^{hub} \rfloor + ½$ →	1100 1101 0011 .1	0.0001 1
$\lfloor \mu_1^{hub} \rfloor + ½$ →	1 0000 01110 011 .1	0.0101
$\lfloor \mu_1^{hub} \rfloor + ½$ →	10 0100 0011 .1	1.0100 10 1
$\lfloor \mu_1^{hub} \rfloor + ½$ →	1 0011 01100 010 .1	1.0111
	↑	↑
	q_i	r_i

Architecture

– Pre-rounding Unit (direct manipulation of the significands)



Architecture

– Combined HUB/Standard adder (with rounding by truncation)

$$s_i^{hub} = Q + v_i^{hub}$$

$$q_i = s_i^{hub} - Q$$

$$r_i = v_i^{hub} - q_i$$

Q = 1000 0000 0000 0000

v_i^{hub} = 1100 1101 0011 .1001 1

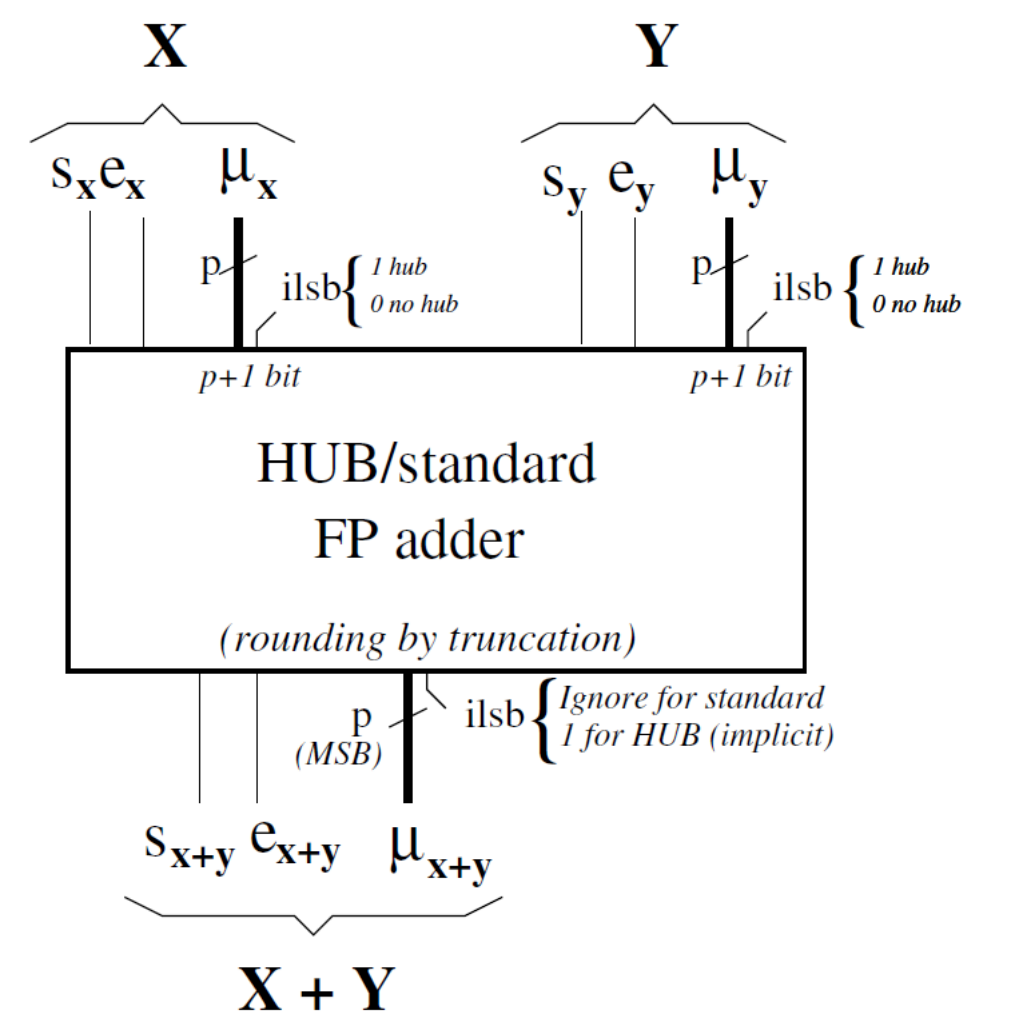
1000 1100 1101 0011 .1

1100 1101 0011 .1

0 .0001 1

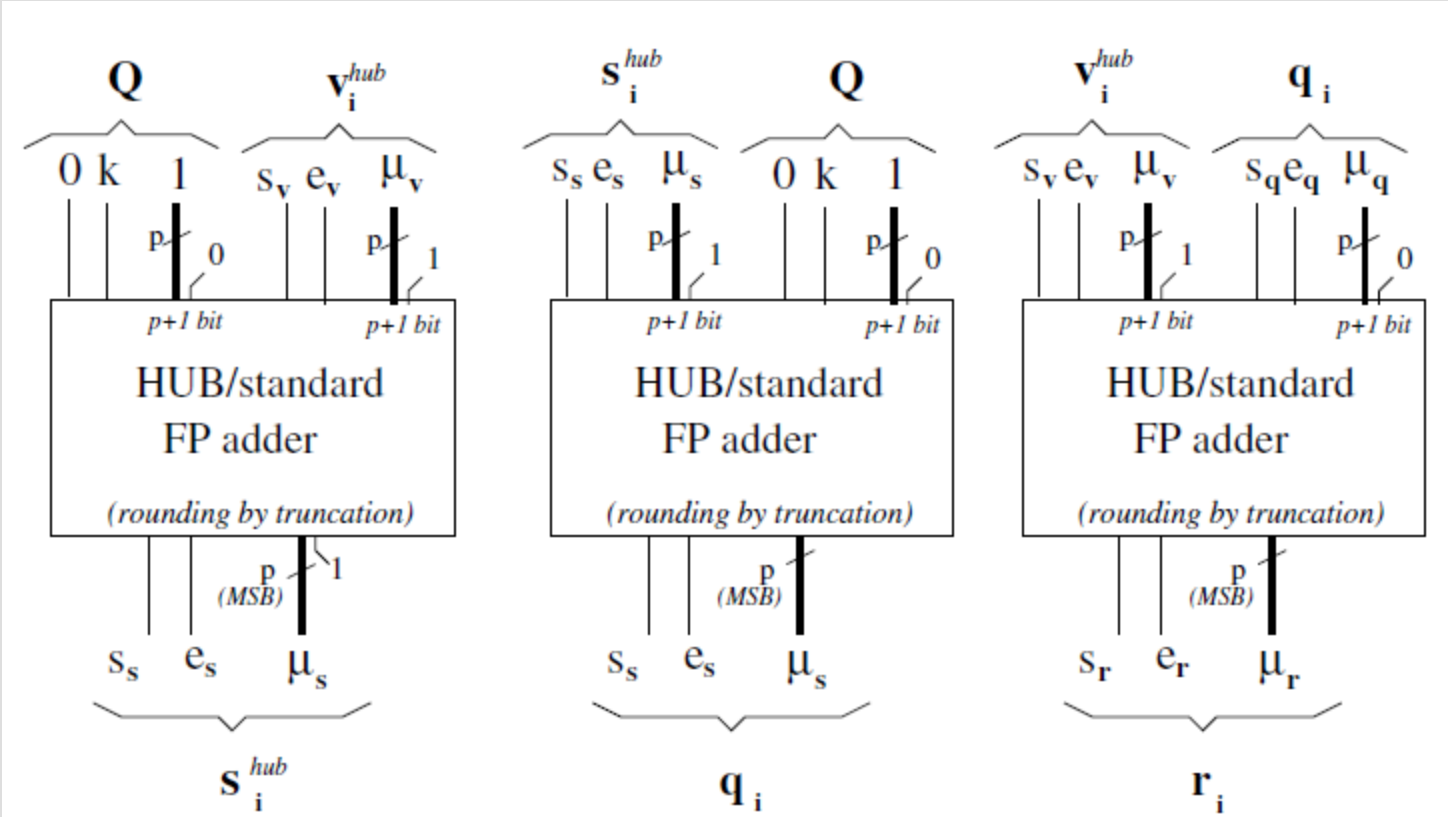
Architecture

– Combined HUB/Standard adder (with rounding by truncation)



Architecture

– Combined HUB/Standard adder (with rounding by truncation)



$$s_i^{hub} = Q + v_i^{hub}$$

$$q_i = s_i^{hub} - Q$$

$$r_i = v_i^{hub} - q_i$$

Summary and conclusion

- HUB format
  - Simplify some important operations
  - Same size as its conventional counterpart for representing
  - In general, reduces the complexity of the underlying hardware
- Reproducible summation in HUB
  - Based on error-free vector transformation (Rump, Ogita, Oishi 2008)
  - Re-formulate of the fundamentals (Lemma I and Theorem I)
  - Split the HUB number (upper&lower) with no carry propagation
  - Architectures for splitting the operands
    - » Specific architecture (pre-rounding unit, no carry propagation)
    - » Combined HUB/Standard adder (rounding by truncation)
  - Future work: Increase the accuracy by exploring the K-folder technique and 1-Reduction

THANK YOU!

QUESTIONS?