Digit Recurrence Floating-point Division under HUB Format
Talk Outline

• Introduction
• HUB format
  – Definition
  – Floating point HUB numbers
  – Advantages and drawbacks
• Division under HUB
  – Digit recurrence algorithm
  – Data path bit-width and number of iterations
  – On-the-fly conversion
  – Unbiased round to nearest
• Summary and conclusion
HUB format
HUB format

- **HUB** = **Half-Unit Biased format**
  
  Digit-vector
  
  \[ X = (X_{n-1}, X_{n-2}, \ldots X_1, X_0, \hat{X}_1, \ldots, X_{-f}) \]

  Conventional number in radix \( \beta \)
  
  \[ X = \left[ \sum_{i=-f}^{n-1} X_i \cdot \beta^i \right] \]

  HUB number in radix \( \beta \)
  
  \[ X = \left[ \sum_{i=-f}^{n-1} X_i \cdot \beta^i \right] + \frac{\beta - f - 1}{2} \]
HUB format

- Example (4 bits, radix 2)
HUB format

- **The extra LSB → ILSB**  Implicit Least Significant Bit
  
  - Value: 1
    - Implicit bit (ILSB)
      - Not stored
      - Not transmitted
      - Not needed for representation
    - Required when operating
HUB format

- Exactly representable numbers (ERN)

- The set of ERN is different for both representations
- $\text{Set}(\text{HUB}) \cap \text{Set}(\text{Conventional}) = \emptyset$ (Disjoint sets)
- Distance between two consecutive numbers is the same
- The amount of ERN is the same
  - Both representations have the same precision
HUB format

- Floating point HUB number in radix-2 ($\beta=2$)

\[(S_x, M_x, E_x)\]
\[x = (-1)^{S_x} M_x 2^{E_x}\]

- Normalized HUB significand:

Digit-vector
\[M_x = (M_{x_0}, M_{x_{-1}}, M_{x_{-2}}, \ldots, M_{x_{-f}})\]

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

- Exponent (conventional): $E_x$\n- Significand (HUB magnitude): $M_x$\n- Sign (conventional): $S_x$\n
$1 < M_x < 2$
HUB format

- Normalized Floating point HUB number in radix-2

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

Representative form

\[ M_x = 1.M_{x-1}M_{x-2} \cdots M_{x-f} \]

Operational form

\[ M_x = 1.M_{x-1}M_{x-2} \cdots M_{x-f}1 \]

Digit-vector

To operate

ILSB
HUB format

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

![Diagram showing the format of single precision IEEE-754 and its HUB counterpart.]

- Stored form:
  - SP IEEE 754 & SP HUB format
- Operational form:
  - SP IEEE 754
  - SP HUB format

Example:

- Stored (for both)
  - SP IEEE 754 (Operat.)
  - SP HUB format (Operat.)
HUB format

• Advantages
  – Two’s complement → bit-wise
  – Round to nearest → by truncation
  – No double rounding error
  – Simplicity

• Drawbacks
  – Not valid for integers
  – Other rounding modes require 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
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

1.1101100111

1.110111

1.110101

1.111000

1.111001

1.110110

1.110101

1.111000

1.111001
**HUB format**

- **The tie case** *(0 for all bits starting at ILSB)*

Formally: \( M'_{x}[0:m-2] = M[0:m-3,0] \)

\( \text{LSB}^* = \text{LSB of the representative form} \)
HUB format

• The tie case, example

\[ M_x \begin{array}{c}
1.0101100000
\end{array} \quad M_x \begin{array}{c}
1.0101000000
\end{array} \]

\[ \downarrow \quad \downarrow \]

\[ \text{LSB}^* \quad \text{ILSB} \]

\[ M_x \begin{array}{c}
1.01010
\end{array} \quad M_x \begin{array}{c}
1.01010
\end{array} \]

Effective rounding down

Effective rounding up

Truncation & Forcing \( \text{LSB}^* = 0 \)
HUB format

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

**Conventional**

1.001 0111 1001

1st rounding: 1.001 1000
2nd rounding: 1.010

Direct rounding:

1.001

**HUB**

1.001 0111 1001

1st rounding: 1.001 0111
ILSB
2nd rounding: 1.001
ILSB

Direct rounding:

1.001
ILSB
HUB format

- Efficiency of HUB format
  - Fixed-point
    - [6]: Optimization of FIR filters
    - [7]: QR decomposition
  - Floating-point
    - [9]: High dynamic range image and video systems
    - [8]: Quantitative analysis of HUB for floating point adders, multipliers and converters
Floating point \([5,8]\)

- **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)

[8] *Measuring the improvement when using HUB formats to implement floating point systems under round to nearest*, IEEE Transactions on VLSI 2015, DOI: 10.1109/TVLSI.2015.2502318
Floating point [5,8]

- Efficiency of HUB format
  - Floating-point
    - Division:
      - Not studied yet

Objective: extend the HUB format to Floating-point Division
Floating-point division under HUB
Floating-point division for HUB

- Division of two FP HUB numbers

\[ x \rightarrow (S_x, M_x, E_x) \]
\[ d \rightarrow (S_d, M_d, E_d) \]

\[ q = \frac{x}{q} \]

\[ q \rightarrow (S_q, M_q, E_q) \]

\( M \rightarrow \text{magnitud & normalized} \)
Floating-point division for HUB

- Digit recurrence algorithm [10]

\[ q(i) = q(0) + \sum_{j=1}^{i} q_j r^{-j} \]

\[ q_i \in [-a, a] \quad a \geq \lceil r/2 \rceil \]

\[ \rho = \frac{a}{r - 1}, \quad \frac{1}{2} < \rho \leq 1 \]

Residual

\[ w(i) = r^i (x - dq(i)) \]

Bound

\[ |w(i)| \leq \rho \cdot d \]

Recurrence

\[ w(i + 1) = rw(i) - dq_{i+1} \]

Selection function

\[ q_{i+1} = \begin{cases} 
1 & \text{if } 0 \leq rw(i) \leq 3/2 \\
0 & \text{if } rw(i) = 3/2 \\
-1 & \text{if } -5/2 \leq rw(i) \leq -1 
\end{cases} \]
Floating-point division for HUB

- Digit recurrence algorithm for HUB

\[ M_x = 1.xxx...xxx1 \quad \text{if} \quad 1 < M_x < 2 \]
\[ M_d = 1.ddd...ddd1 \quad \text{if} \quad 1 < M_d < 2 \]

\[ \Rightarrow \quad M_q \in (\frac{1}{2}, 2) \quad \Rightarrow \quad \text{Normalization required} \]

- Initialization step
  - \( w(0) = x/2 \) or \( w(0) = x-d \) if \( \rho = 1 \) (maximum redundancy)
  - \( w(0) = x/4 \) if \( \rho < 1 \)

- Termination step
  - Correction of the initialization (1 or 2 positions left shift)
  - Correction if final negative residue
  - Normalization if quotient \( \epsilon (\frac{1}{2} < 1) \)
  - **Rounding-to-nearest: by truncation / adding 1 ulp (conventional)**
  - Check zero condition if exact quotient is needed
Floating-point division for HUB

- Data path bit-width (h)
  - m (operational form of HUB number) plus
    » 1 guard bits due to normalization (bit $G$)
    » 1 bit if $\rho = 1$ or 2 bits if $\rho < 1$ (bit $S$, scaling of the intitilization step)

\[ h = m + 1 + (2 - \lfloor \rho \rfloor) = m + 3 - \lfloor \rho \rfloor \]
Floating-point division for HUB

- Data path bit-width (h)

$h = m + 1 + (2 - \lfloor \rho \rfloor) = m + 3 - \lfloor \rho \rfloor$

Conclusion: same data path bit-width for HUB and for its conventional counterpart

Example for $\rho=1$
Floating-point division for HUB

- Operation inside the divisor

\[
d = 1. \text{xxx} \ldots \text{xxx}1 \quad \text{HUB}
\]

\[
x = 1.\text{xxx} \ldots \text{xxx}1 \quad \text{HUB}
\]

**RECURRENCE**

\[
w(0) = 0.1\text{xxx} \ldots \text{xxx}100 \quad \text{(no HUB)}
\]

\[
w(1) = X.\text{xxxx} \ldots \text{xxxxxxx} \quad \text{(no HUB)}
\]

\[
w(2) = X.\text{xxxx} \ldots \text{xxxxxxx} \quad \text{(no HUB)}
\]

\[\ldots \ldots \ldots \ldots \]

\[
w(N) = X.\text{xxxx} \ldots \text{xxxxxxx} \quad \text{(no HUB)}
\]

\[
w(N) = \frac{X.\text{xxxx} \ldots \text{xxx}}{X.\text{xxxx} \ldots \text{xxx}} \quad \text{(HUB op.)}
\]

\[
w(N) = \frac{X.\text{xxxx} \ldots \text{xxx}1}{X.\text{xxxx} \ldots \text{xxx}} \quad \text{(HUB rep.)}
\]

**QUOTIENT**

\[
q_1 \quad \text{(no HUB)}
\]

\[
q_1 q_2 \quad \text{(no HUB)}
\]

\[
q_1 q_2 q_3 \quad \text{(no HUB)}
\]

\[\ldots \ldots \ldots \ldots \]

\[
q_1 q_2 q_3 \ldots q_N \quad \text{(no HUB)}
\]

\[
M_q = \frac{1.\text{xxx} \ldots \text{xxx}1}{1.\text{xxx} \ldots \text{xxx}} \quad \text{(HUB, op.)}
\]

\[
M_q = \frac{1.\text{xxx} \ldots \text{xxx}1}{1.\text{xxx} \ldots \text{xxx}} \quad \text{(HUB, rep.)}
\]

*Quotient (significand) after correction, normalization and rounding by truncation*
Floating-point division for HUB

- Number of iterations
  - Depends on the number of bits to be computed by the iterations \( (h) \) and the radix \( (r) \)

\[
N = \left\lceil \frac{h}{\log_2 r} \right\rceil
\]

**Conclusion**: same number of iterations for HUB and for its conventional counterpart
Floating-point division for HUB

Conventional with round to nearest

Conventional recurrence

N-iterations

Termination step

Mx = 1. xxx ... xxx000
Md = 1. xxx ... xxx000

HUB

Conventional

Mx = 1. xxx ... xxx100
Md = 1. xxx ... xxx100

Termination step *

Conventional recurrence

N-iterations

HUB
On-the-fly conversion and rounding

– On the fly conversion
  – Overlapping the final conversion from redundant to conventional with the iterations
  – Rounding, correction and normalization is integrated in the on-the-fly conversion of the last digit
On-the-fly conversion and rounding

- On the fly conversion [10,12]

\[ Q(i) \rightarrow i \text{ MSD of the converted quotient (digit vector)} \]

\[ Q(i) = \sum_{j=1}^{i} q_{i} r^{-i} \]

To avoid carry propagation:

\[ QD(i) = Q(i) - r^{-i} \]

On-the-fly algorithm:

\[
Q(i + 1) = \begin{cases} 
(Q(i) \parallel q_{i+1}) & \text{if } q_{i+1} \geq 0 \\
(QD(i) \parallel (r - |q_{i+1}|)) & \text{if } q_{i+1} < 0 
\end{cases}
\]

\[
QD(i + 1) = \begin{cases} 
(Q(i) \parallel q_{i+1} - 1) & \text{if } q_{i+1} > 0 \\
(QD(i) \parallel (r - 1 - |q_{i+1}|)) & \text{if } q_{i+1} \leq 0 
\end{cases}
\]
On-the-fly conversion and rounding

- On the fly conversion [10,12]
  - Round to nearest → addition of 1 ulp after regular iterations
    » Last digit can be out of range of the digit set \([-a,a]\) if \(a \geq r-2\)
    » \(q_N = a \rightarrow q_{N+1} \notin [-a,a]\)
  - A new form \(QR(i)\) is defined to combine the correction, normalization and rounding:

\[
QR(i) = Q(i) + r^{-i}
\]

\[
QR(i + 1) = \begin{cases} 
(QR(i) \parallel 0) & \text{if } q_{i+1} = r - 1 \\
(Q(i) \parallel q_{i+1} + 1) & \text{if } -1 \leq q_{i+1} \leq r - 2 \\
(QD(i) \parallel (r + 1 - |q_{i+1}|)) & \text{if } q_{i+1} < -1 
\end{cases}
\]
On-the-fly conversion and rounding

- On the fly conversion [10,12]

Rounded significand before normalization and truncation

\[
MMq = \begin{cases} 
(QR(N - 1) \| u) & \text{if } q^*_N \geq r \\
(Q(N - 1) \| u) & \text{if } 0 \leq q^*_N \leq r - 1 \\
(QD(N - 1) \| u) & \text{if } q^*_N < r 
\end{cases}
\]

\[
q^*_N \in \{-a, a + 1\}
\]

\[
u = q^*_N \mod r
\]

Final normalized significand:

\[
M_q[0 : m - 2] = \begin{cases} 
MMq[0 : m - 2] & \text{if } MMq[0] = 1 \\
MMq[1 : m - 1] & \text{if } MMq[0] = 0 
\end{cases}
\]
On-the-fly conversion and rounding

– On the fly conversion for HUB

\[
Q(i + 1) = \begin{cases} 
(Q(i) \parallel q_{i+1}) & \text{if } q_{i+1} \geq 0 \\
(QD(i) \parallel (r - |q_{i+1}|)) & \text{if } q_{i+1} < 0
\end{cases}
\]

\[
QD(i + 1) = \begin{cases} 
(Q(i) \parallel q_{i+1} - 1) & \text{if } q_{i+1} > 0 \\
(QD(i) \parallel (r - 1 - |q_{i+1}|)) & \text{if } q_{i+1} \leq 0
\end{cases}
\]

(Same as [10,12])

– Round to nearest is carried out by truncation \(\rightarrow\) No carry propagation \(\rightarrow\) no addition of 1 ulp after regular iterations
  » Last digit is always inside the range of the digit set \([-a,a]\)

– The form \(QR(i)\) is not required any more

\[
QR(i) = Q(i) + r^{-i}
\]

\[
QR(i + 1) = \begin{cases} 
(QR(i) \parallel 0) & \text{if } q_{i+1} = r - 1 \\
(Q(i) \parallel q_{i+1} - 1) & \text{if } -1 \leq q_{i+1} \leq r - 2 \\
(QD(i) \parallel (r + 1 - |q_{i+1}|)) & \text{if } q_{i+1} < -1
\end{cases}
\]
On-the-fly conversion and rounding

On the fly conversion for HUB

\[ Q(i + 1) = \begin{cases} 
(Q(i) || q_{i+1}) & \text{if } q_{i+1} \geq 0 \\
(QD(i) || (r - |q_{i+1}|)) & \text{if } q_{i+1} < 0 
\end{cases} \]

\[ QD(i + 1) = \begin{cases} 
(Q(i) || q_{i+1} - 1) & \text{if } q_{i+1} > 0 \\
(QD(i) || (r - 1 - |q_{i+1}|)) & \text{if } q_{i+1} \leq 0 
\end{cases} \]

On the fly conversion for conventional

\[ Q(i + 1) = \begin{cases} 
(Q(i) || q_{i+1}) & \text{if } q_{i+1} \geq 0 \\
(QD(i) || (r - |q_{i+1}|)) & \text{if } q_{i+1} < 0 
\end{cases} \]

\[ QD(i + 1) = \begin{cases} 
(Q(i) || q_{i+1} - 1) & \text{if } q_{i+1} > 0 \\
(QD(i) || (r - 1 - |q_{i+1}|)) & \text{if } q_{i+1} \leq 0 
\end{cases} \]

\[ QR(i + 1) = \begin{cases} 
(QR(i) || 0) & \text{if } q_{i+1} = r - 1 \\
(Q(i) || q_{i+1} + 1) & \text{if } -1 \leq q_{i+1} \leq r - 2 \\
(QD(i) || (r + 1 - |q_{i+1}|)) & \text{if } q_{i+1} < -1 
\end{cases} \]
On-the-fly conversion and rounding

– On the fly conversion for HUB

We define QC as a function of the sign of w(N):

\[
QC = \begin{cases} 
Q(N) & \text{if } \text{sign} = 0 \\
QD(N) & \text{if } \text{sign} = 1 
\end{cases}
\]

Final normalized and round to nearest HUB quotient (representative form):

\[
M_q[0 : m-2] = \begin{cases} 
QC[0 : m-2] & \text{if } QC[0] = 1 \\
QC[1 : m-1] & \text{if } QC[0] = 0 
\end{cases}
\]
On-the-fly conversion and rounding

– Optimizing the on-the-fly conversion

– QC can be obtained from $Q(N-1)$ and $QD(N-1)$

Last cycle:

\[
QC = \begin{cases} 
Q(N) & \text{if } sign = 0 \\
QD(N) & \text{if } sign = 1 
\end{cases}
\]

\[
Q(N) = \begin{cases} 
(Q(N-1) \parallel q_N) & \text{if } q_N \geq 0 \\
(QD(N-1) \parallel (r - |q_{i+1}|)) & \text{if } q_N < 0 
\end{cases}
\]

\[
QD(N) = \begin{cases} 
(Q(N-1) \parallel q_N - 1) & \text{if } q_N > 0 \\
(QD(N-1) \parallel (r - 1 - |q_N|)) & \text{if } q_N \leq 0 
\end{cases}
\]
On-the-fly conversion and rounding

HUB

Conventional
On-the-fly conversion and rounding

- Unbiased round to nearest
  - In conventional IEEE standard the tie case never happens [10]
  - In HUB, it can happen
- The tie case in HUB:
  - Remainder is zero & LSB of the quotient is zero

![Diagram showing tie case and HUB ERN](image)
On-the-fly conversion and rounding

- Unbiased round to nearest

Normalized Quotient before rounding

1.xxx...xxxKL

z KL --> K'L'
0 xy x 1 (not tie)
1 x1 x 1 (not tie)
1 00 0 1 (tie, rounding up)
1 10 0 1 (tie, rounding down)

Normalized Quotient before rounding (tie case) 1.xxx...xxxK0 (operational form)

Normalized Quotient after rounding (tie case) 1.xxx...xxx01 (operational form)

1.xxx...xxx0 (representative form)
Summary and conclusion

- **HUB format**
  - Simplify some important operations
  - Represented with the same number of bits as its conventional counterpart
  - In general, it reduces the complexity of the hardware

- **Digit recurrence division under HUB**
  - The same data path bit-width as conventional (counterpart)
  - The same number of iterations
  - The same selection function
  - The same hardware than conventional format for the iterations
  - On-the-fly conversion for \( a \geq r-2 \)
    - Less hardware requirements
    - The selection of the last digit is simplified
THANK YOU!

QUESTIONS?