Probabilistic Error Analysis of Limited Precision Stochastic Rounding

joint work with El-Mehdi El Arar, Mantas Mikaitis, and Massimiliano Fasi

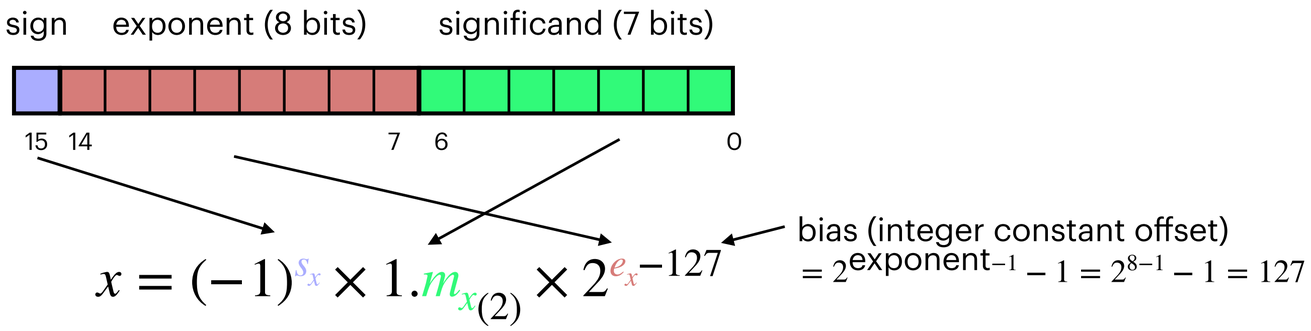
Silviu-Ioan Filip, 1 September 2025

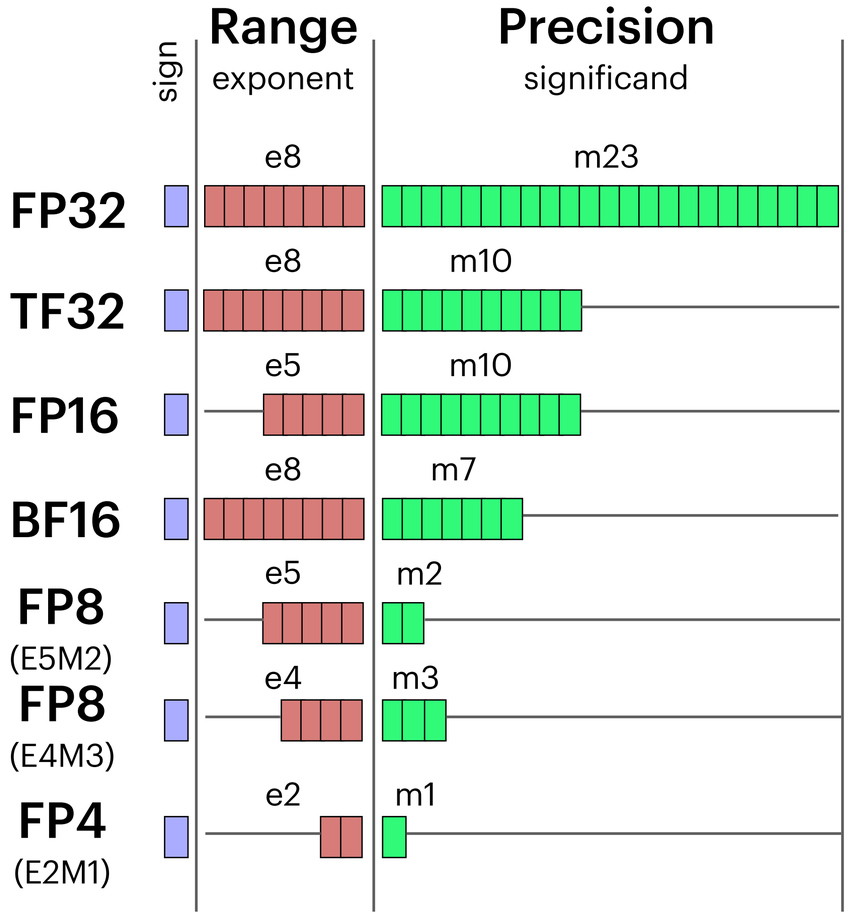
Common and Emerging FP Formats in HPC & AI

➡they offer various tradeoffs in terms of range, precision & performance

FP performance numbers for recent NVIDIA GPU architectures

Peak performance (TFLOPS)							
Device	Year	fp64	fp32	tfloat32	fp16	bfloat16	fp8
P100	2016	5	9	-	19	-	-
V100	2017-2019	8	16	-	125	-	-
A100	2020-2021	19	19	156	312	312	-
H100	2022	48	48	400	800	800	1600



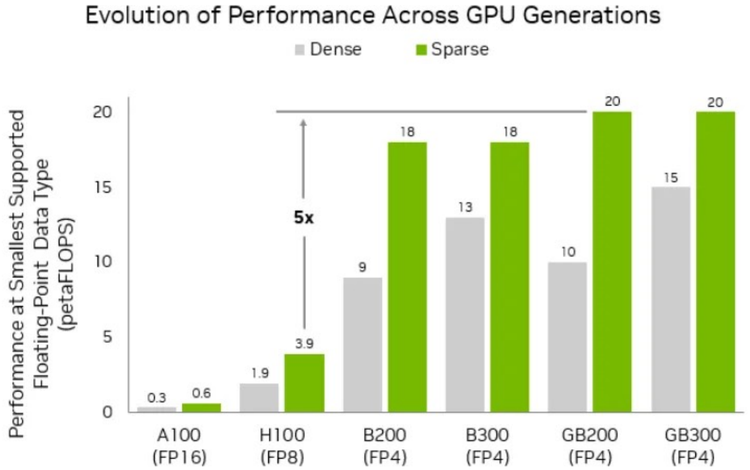


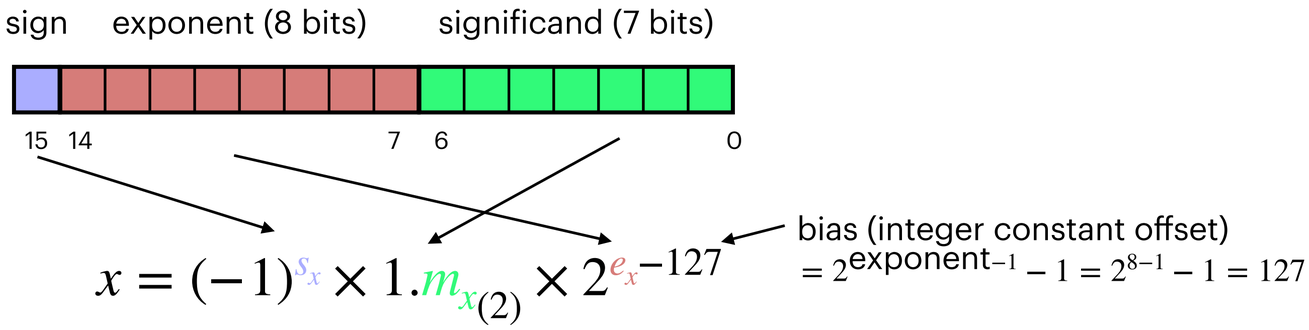
IEEE FP Standardisation for ML: P3109 WG

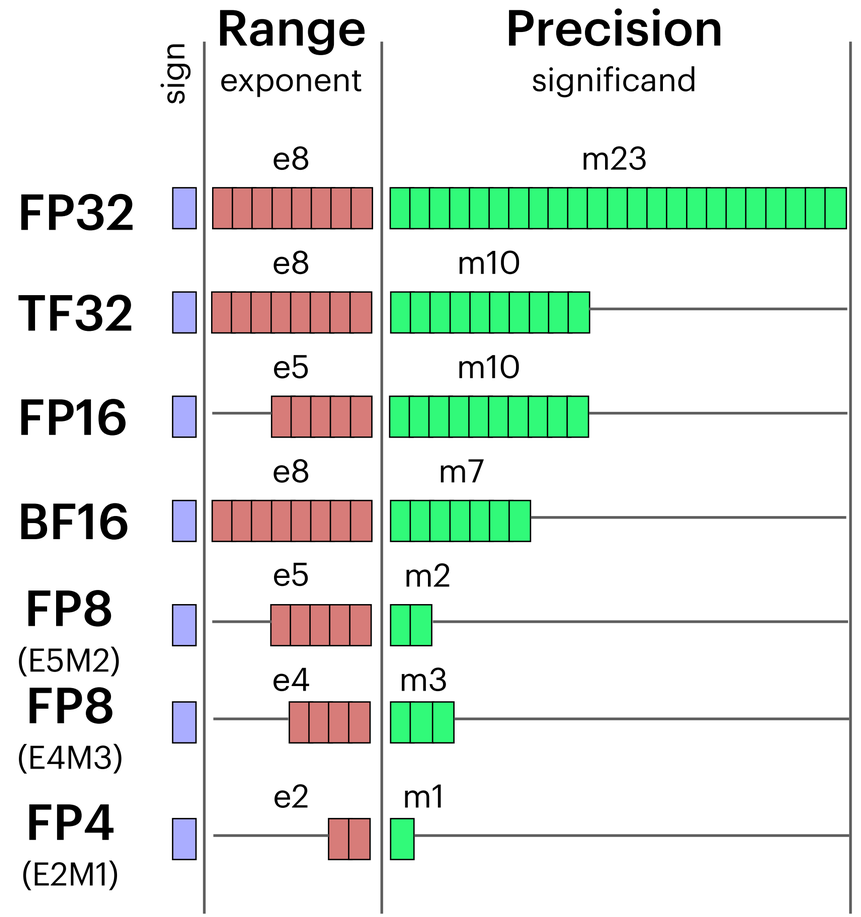
https://github.com/P3109/Public

Common and Emerging FP Formats in HPC & AI

➡they offer various tradeoffs in terms of range, precision & performance





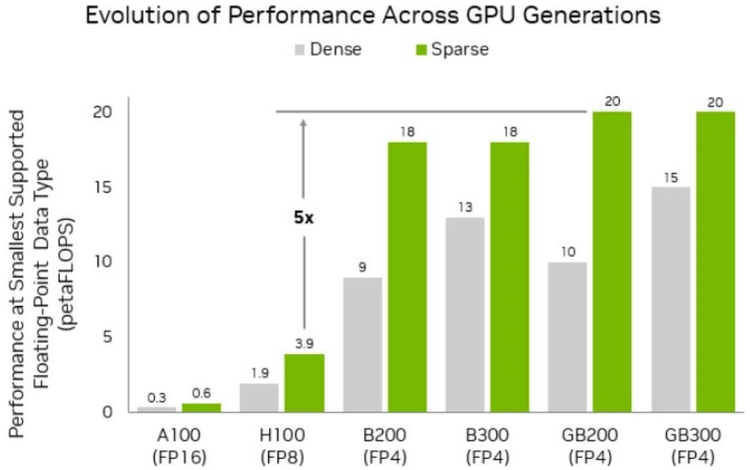


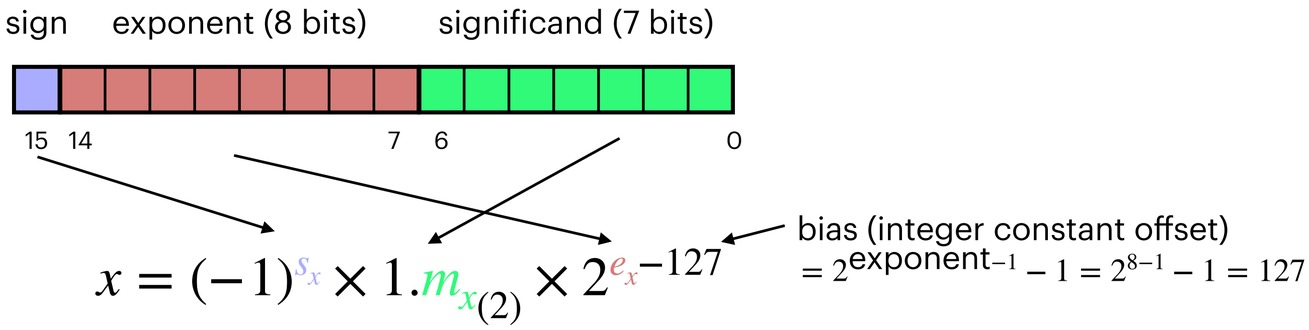
IEEE FP Standardisation for ML: P3109 WG

https://github.com/P3109/Public

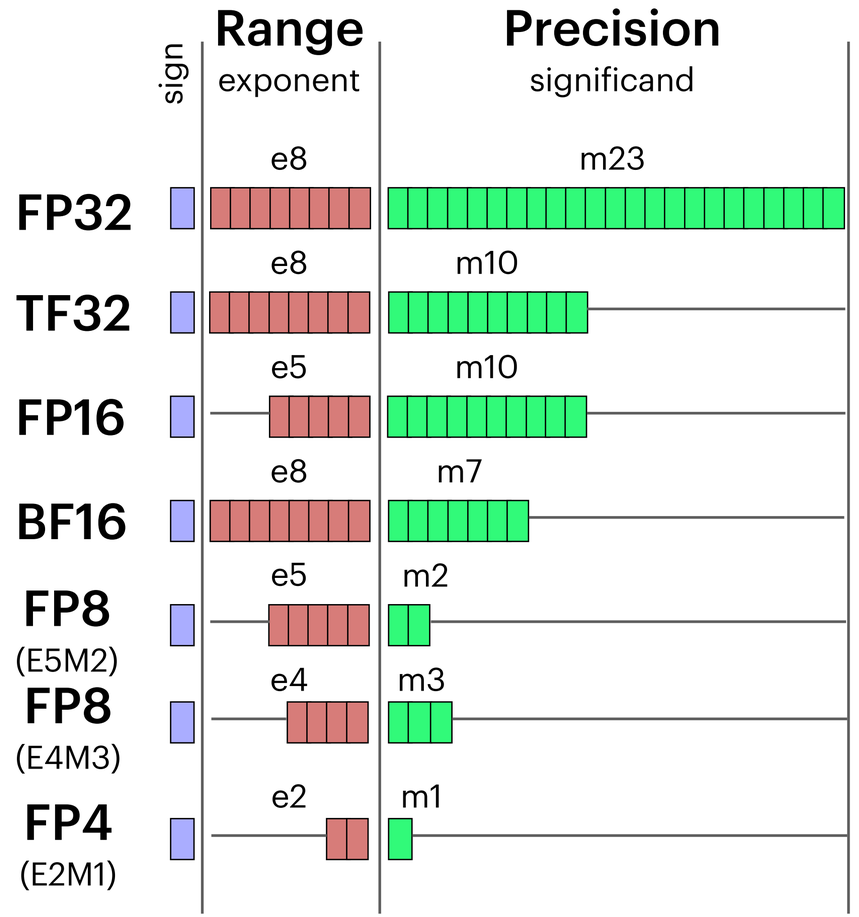
Common and Emerging FP Formats in HPC & AI

➡they offer various tradeoffs in terms of range, precision & performance





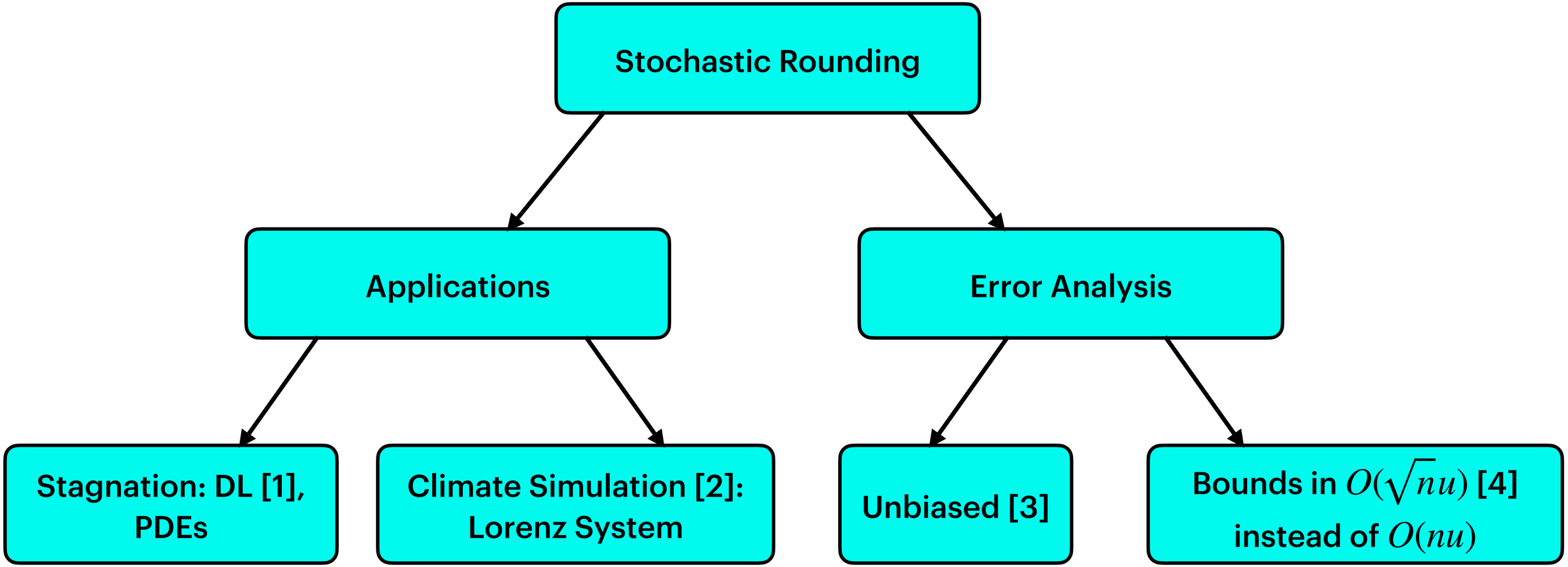
➡effective rounding modes are critical: **stochastic rounding**



IEEE FP Standardisation for ML: P3109 WG

https://github.com/P3109/Public

Motivation



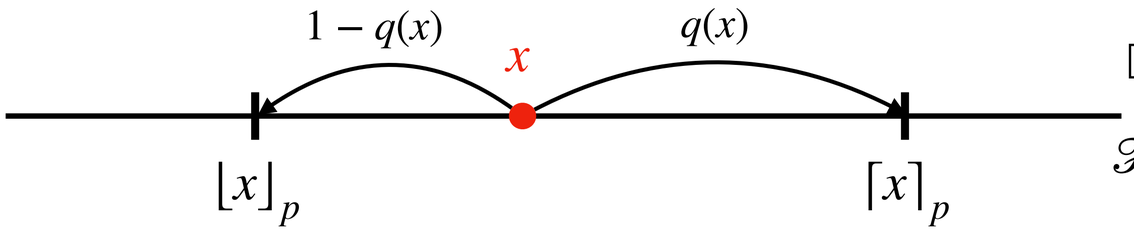
[1] Gupta et al: Deep Learning with Limited Numerical Precision
[2] Paxton et al: Climate Modeling in Low Precision: Effects of Both Deterministic and Stochastic Rounding
[3] Parker: Monte Carlo Arithmetic: Exploiting Randomness in Floating-Point Arithmetic
[4] El Arar: Stochastic Models for the Evaluation of Numerical Errors

Stochastic Rounding

For $x, y \in \mathbb{R}$ and $\text{op} \in \left\{+, -, \times, /, \sqrt{}\right\}$

$$\text{SR}_p(x) = x(1+\delta), \qquad |\delta| \leqslant u_p := 2^{1-p}$$

$$\text{SR}_p(x \text{ op } y) = (x \text{ op } y)(1+\beta), \qquad |\beta| \leqslant u_p$$



$\mathbb{R}$

$\mathscr{F}_p$

$$\text{SR}_p(x) = \begin{cases} \lceil x \rceil_p, & \text{with prob. } q(x) \\ \lfloor x \rfloor_p, & \text{with prob. } 1 - q(x) \end{cases} \qquad q(x) = \frac{x - \lfloor x \rfloor_p}{\lceil x \rceil_p - \lfloor x \rfloor_p}$$

- Introduced by Forsythe in 1950

not exceeding u. The present author proposes a *random round-off*, whereby any real number $u = [u] + v$ is "rounded up" to $[u]+1$ with probability v, and "rounded down" to $[u]$ with probability $1-v$. The error r of random round-off is truly a random variable. Since $E(r) = 0$ and $E(r^2) = v(1-v) \leqq 1/2$, simple probabilistic bounds on the accumulated error can be given without assuming a distribution for v. Tests with

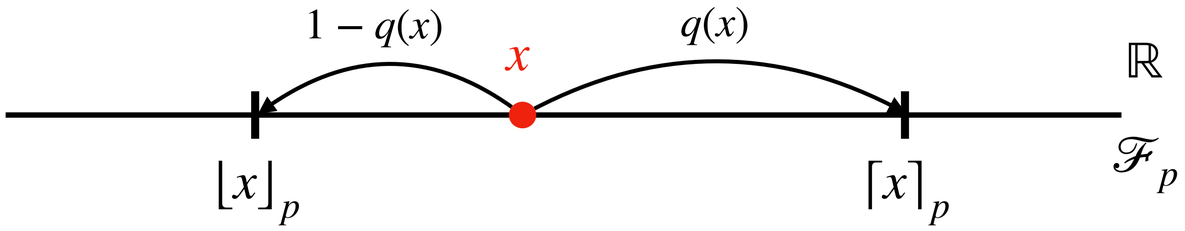
- Of interest in recent years due to use in Deep Learning

Stochastic Rounding

For $x, y \in \mathbb{R}$ and $\text{op} \in \left\{+, -, \times, /, \sqrt{}\right\}$

$$\text{SR}_p(x) = x(1+\delta), \qquad |\delta| \leqslant u_p := 2^{1-p}$$

$$\text{SR}_p(x \text{ op } y) = (x \text{ op } y)(1+\beta), \qquad |\beta| \leqslant u_p$$



$\mathbb{R}$

$\mathscr{F}_p$

$$\text{SR}_p(x) = \begin{cases} \lceil x \rceil_p, & \text{with prob. } q(x) \\ \lfloor x \rfloor_p, & \text{with prob. } 1 - q(x) \end{cases} \qquad q(x) = \frac{x - \lfloor x \rfloor_p}{\lceil x \rceil_p - \lfloor x \rfloor_p}$$

- Introduced by Forsythe in 1950

not exceeding u. The present author proposes a *random round-off*, whereby any real number $u = [u] + v$ is "rounded up" to $[u]+1$ with probability v, and "rounded down" to $[u]$ with probability $1-v$. The error r of random round-off is truly a random variable. Since $E(r) = 0$ and $E(r^2) = v(1-v) \leqq 1/2$, simple probabilistic bounds on the accumulated error can be given without assuming a distribution for v. Tests with

- Of interest in recent years due to use in Deep Learning

Emerging HW support

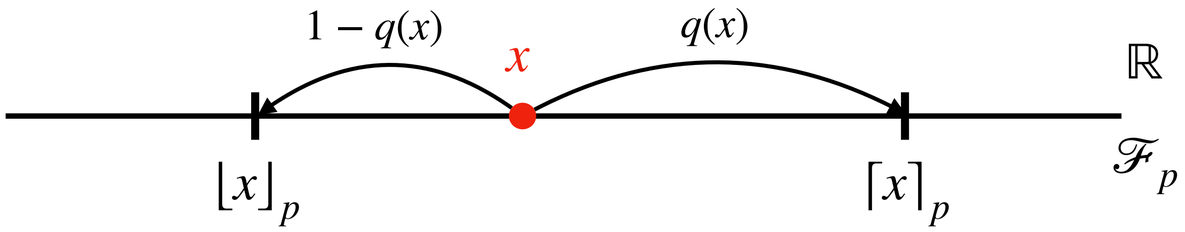
- Graphcore IPUs
- AMD MI300 GPUs
- Tesla D1
- Amazon Trainium
- NVIDIA Blackwell

Stochastic Rounding

For $x, y \in \mathbb{R}$ and op $\in \left\{+, -, \times, /, \sqrt{}\right\}$

$$\mathrm{SR}_p(x) = x(1+\delta), \qquad |\delta| \leqslant u_p := 2^{1-p}$$

$$\mathrm{SR}_p(x \text{ op } y) = (x \text{ op } y)(1+\beta), \qquad |\beta| \leqslant u_p$$



$$\mathrm{SR}_p(x) = \begin{cases} \lceil x \rceil_p, & \text{with prob. } q(x) \\ \lfloor x \rfloor_p, & \text{with prob. } 1 - q(x) \end{cases} \qquad q(x) = \frac{x - \lfloor x \rfloor_p}{\lceil x \rceil_p - \lfloor x \rfloor_p}$$

Advantages

+ roundoff errors with zero mean & error cancellation in long comp. chains
+ tends to avoid stagnation effects
+ total roundoff error grows more slowly with problem size n than with classic deterministic rounding (e.g. round to nearest (RN))

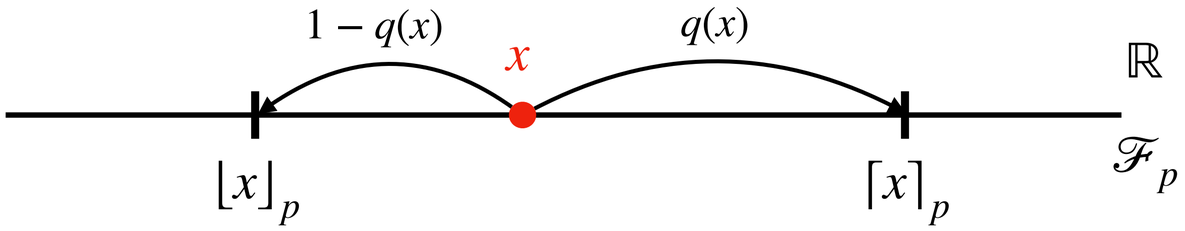
$$O\left(\sqrt{n}u_p\right) \text{vs } O\left(nu_p\right)$$

Stochastic Rounding

For $x, y \in \mathbb{R}$ and op $\in \left\{+, -, \times, /, \sqrt{}\right\}$

$$\mathrm{SR}_p(x) = x(1+\delta), \qquad |\delta| \leqslant u_p := 2^{1-p}$$

$$\mathrm{SR}_p(x \text{ op } y) = (x \text{ op } y)(1+\beta), \qquad |\beta| \leqslant u_p$$



$$\mathrm{SR}_p(x) = \begin{cases} \lceil x \rceil_p, & \text{with prob. } q(x) \\ \lfloor x \rfloor_p, & \text{with prob. } 1 - q(x) \end{cases} \qquad q(x) = \frac{x - \lfloor x \rfloor_p}{\lceil x \rceil_p - \lfloor x \rfloor_p}$$

Advantages

+ roundoff errors with zero mean & error cancellation in long comp. chains
+ tends to avoid stagnation effects
+ total roundoff error grows more slowly with problem size n than with classic deterministic rounding (e.g. round to nearest (RN))

$$O\left(\sqrt{n}u_p\right) \text{vs } O\left(nu_p\right)$$

Limitations

– lack of reproducibility

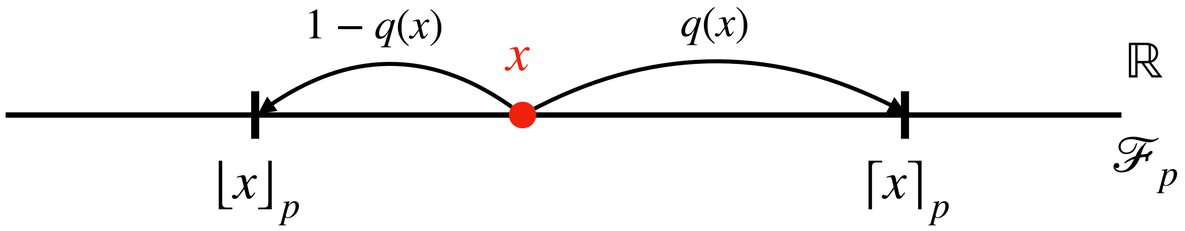
– increased computational overhead

– inability to use true randomness

Stochastic Rounding

For $x, y \in \mathbb{R}$ and $\text{op} \in \left\{+, -, \times, /, \sqrt{}\right\}$

$$\text{SR}_p(x) = x(1+\delta), \qquad |\delta| \leqslant u_p := 2^{1-p}$$
$$\text{SR}_p(x \text{ op } y) = (x \text{ op } y)(1+\beta), \qquad |\beta| \leqslant u_p$$



$$\text{SR}_p(x) = \begin{cases} \lceil x \rceil_p, & \text{with prob. } q(x) \\ \lfloor x \rfloor_p, & \text{with prob. } 1 - q(x) \end{cases} \qquad q(x) = \frac{x - \lfloor x \rfloor_p}{\lceil x \rceil_p - \lfloor x \rfloor_p}$$

Advantages

+ roundoff errors with zero mean & error cancellation in long comp. chains
+ tends to avoid stagnation effects
+ total roundoff error grows more slowly with problem size n than with classic deterministic rounding (e.g. round to nearest (RN))

$O\left(\sqrt{n}u_p\right)$ vs $O\left(nu_p\right)$

Limitations

– lack of reproducibility

– increased computational overhead

– inability to use true randomness

How can we show this?

Ingredients

Definition (*Mean Independence*) A rv X is ***mean independent*** of the rv Y if $\mathbb{E}(X \mid Y) = \mathbb{E}(X)$. The sequence of rvs $X_0, X_1, \ldots$ is ***mean independent*** if $\mathbb{E}(X_k \mid X_0, \ldots, X_{k-1}) = \mathbb{E}(X_k)$ for all k.

Ingredients

Definition (*Mean Independence*) A rv X is ***mean independent*** of the rv Y if $\mathbb{E}(X \mid Y) = \mathbb{E}(X)$. The sequence of rvs $X_0, X_1, \ldots$ is ***mean independent*** if $\mathbb{E}(X_k \mid X_0, \ldots, X_{k-1}) = \mathbb{E}(X_k)$ for all k.

Definition (*Martingale*) A sequence of rvs $M_0, \ldots, M_n$ is a ***martingale*** with respect to the sequence of rvs $X_0, \ldots, X_n$ if, for all k:

(a) M_k is a function of $X_0, \ldots, X_k$;

(b) $\mathbb{E}(|M_k|) < \infty$;

(c) $\mathbb{E}(M_k \mid X_0, \ldots, X_{k-1}) = M_{k-1}$.

Ingredients

Definition (*Mean Independence*) A rv X is ***mean independent*** of the rv Y if $\mathbb{E}(X \mid Y) = \mathbb{E}(X)$. The sequence of rvs $X_0, X_1, \ldots$ is ***mean independent*** if $\mathbb{E}(X_k \mid X_0, \ldots, X_{k-1}) = \mathbb{E}(X_k)$ for all k.

Definition (*Martingale*) A sequence of rvs $M_0, \ldots, M_n$ is a ***martingale*** with respect to the sequence of rvs $X_0, \ldots, X_n$ if, for all k:

(a) M_k is a function of $X_0, \ldots, X_k$;

(b) $\mathbb{E}(|M_k|) < \infty$;

(c) $\mathbb{E}(M_k \mid X_0, \ldots, X_{k-1}) = M_{k-1}$.

Lemma (*Azuma-Hoeffding*) Let $M_0, \ldots, M_n$ be a martingale wrt $X_0, \ldots, X_n$. We assume there exist $a_k > 0$ s.t. $|M_k - M_{k-1}| \leqslant a_k$, for $k = 1, \ldots, n$. Then, for any $\lambda > 0$

$$\mathbb{P}\left(|M_n - M_0| \geqslant \lambda \sqrt{\sum_{k=1}^{n} a_k^2} \right) \leqslant 2 \exp\left(-\lambda^2/2\right).$$

Proof Sketch [1]

Mean independence of rounding errors

The computations of interest generate rounding errors $\delta_1, \delta_2, \ldots$

The δ_k are rvs of mean zero such that

$$\mathbb{E}(\delta_k \,|\, \delta_1, \ldots, \delta_{k-1}) = \mathbb{E}(\delta_k) = 0$$

[1] Stochastic rounding and its probabilistic backward error analysis, M. P. Connolly, N. J. Higham, T. Mary, SISC, vol. 43, No. 1, pp. A565-A585, 2021

Proof Sketch [1]

Mean independence of rounding errors

↓

Obtain a **martingale**

The computations of interest generate rounding errors $\delta_1, \delta_2, \ldots$

The δ_k are rvs of mean zero such that

$$\mathbb{E}(\delta_k \,|\, \delta_1, \ldots, \delta_{k-1}) = \mathbb{E}(\delta_k) = 0$$

$$M_0 = 0, M_k = \sum_{i=1}^{k} \delta_i \text{ for } k \geqslant 1$$

[1] Stochastic rounding and its probabilistic backward error analysis, M. P. Connolly, N. J. Higham, T. Mary, SISC, vol. 43, No. 1, pp. A565-A585, 2021

Proof Sketch [1]

Mean independence of rounding errors

Obtain a **martingale**

Get $O\left(\sqrt{n}u_p\right)$ probabilistic error bound using **Azuma-Hoeffding**

The computations of interest generate rounding errors $\delta_1, \delta_2, \ldots$

The δ_k are rvs of mean zero such that

$$\mathbb{E}(\delta_k \,|\, \delta_1, \ldots, \delta_{k-1}) = \mathbb{E}(\delta_k) = 0$$

$$M_0 = 0, M_k = \sum_{i=1}^{k} \delta_i \text{ for } k \geqslant 1$$

$$\prod_{i=1}^{n}(1+\delta_i) = 1 + \theta_n, |\,\theta_n\,| \leqslant \lambda\sqrt{n}u_p + O(u_p^2)$$ with probability at least $1 - 2\exp\left(-\lambda^2/2\right)$

If $y = a^T b$, where $a, b \in \mathbb{R}^n$, is evaluated using SR_p, then the computed result $\hat{y}$ satisfies

$$\frac{|\,y - \hat{y}\,|}{|\,y\,|} \leqslant \frac{|\,a\,|^T |\,b\,|}{|\,a^T b\,|}\left(\lambda\sqrt{n}u_p + O(u_p^2)\right)$$

with the same probability

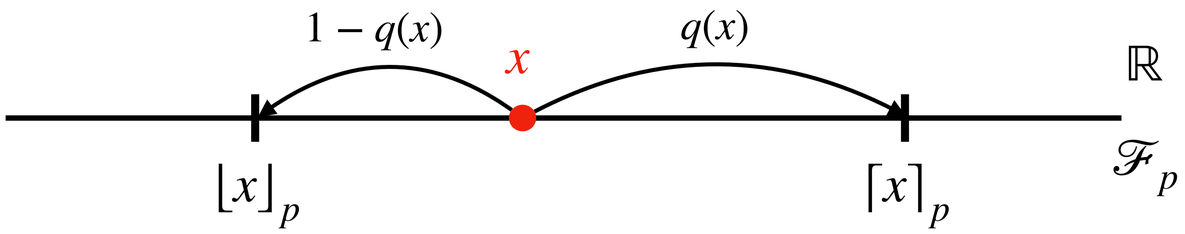
[1] Stochastic rounding and its probabilistic backward error analysis, M. P. Connolly, N. J. Higham, T. Mary, SISC, vol. 43, No. 1, pp. A565-A585, 2021

Stochastic Rounding

For $x, y \in \mathbb{R}$ and $\text{op} \in \left\{+, -, \times, /, \sqrt{}\right\}$

$$\text{SR}_p(x) = x(1+\delta), \qquad |\delta| \leqslant u_p := 2^{1-p}$$

$$\text{SR}_p(x \text{ op } y) = (x \text{ op } y)(1+\beta), \qquad |\beta| \leqslant u_p$$



$$\text{SR}_p(x) = \begin{cases} \lceil x \rceil_p, & \text{with prob. } q(x) \\ \lfloor x \rfloor_p, & \text{with prob. } 1 - q(x) \end{cases} \qquad q(x) = \frac{x - \lfloor x \rfloor_p}{\lceil x \rceil_p - \lfloor x \rfloor_p}$$

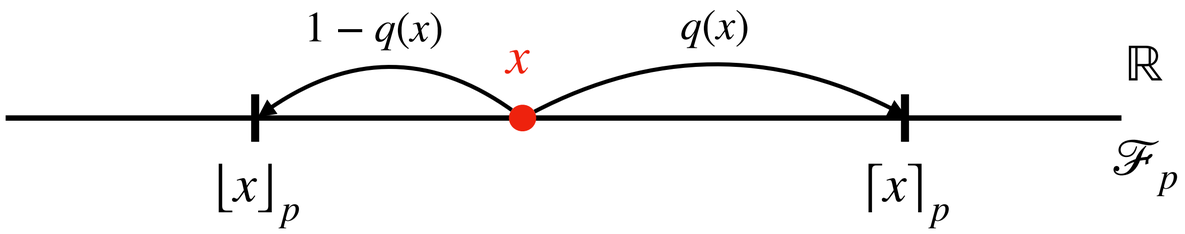
What do we need to implement it?

Stochastic Rounding

For $x, y \in \mathbb{R}$ and op $\in \left\{+, -, \times, /, \sqrt{}\right\}$

$$\mathrm{SR}_p(x) = x(1+\delta), \qquad |\delta| \leqslant u_p := 2^{1-p}$$

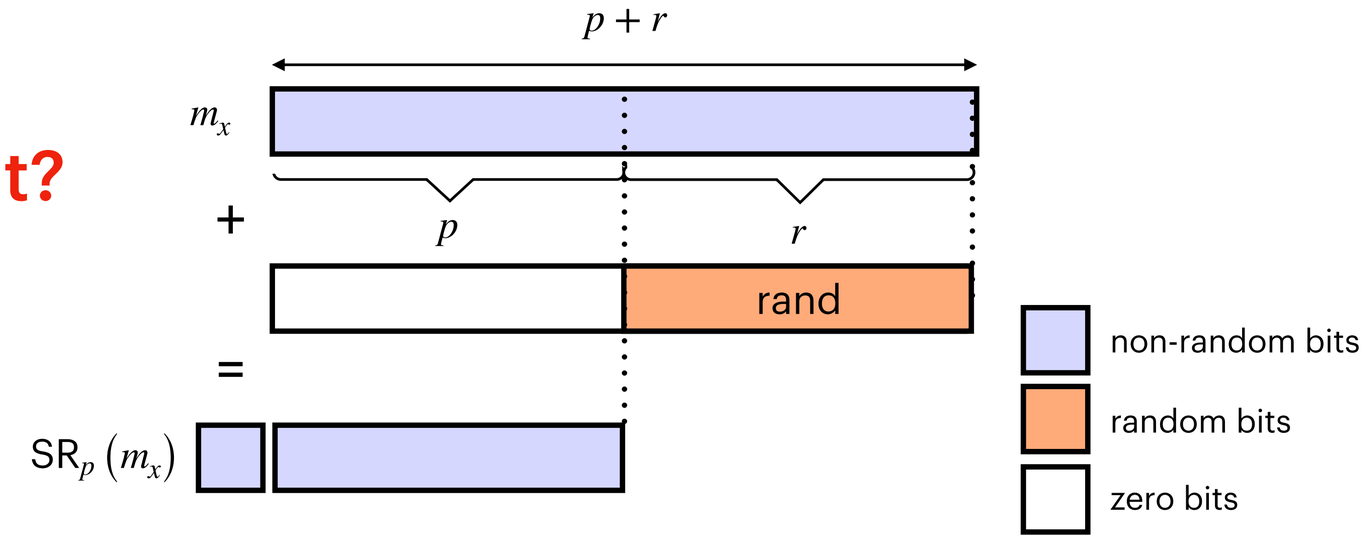
$$\mathrm{SR}_p(x \text{ op } y) = (x \text{ op } y)(1+\beta), \qquad |\beta| \leqslant u_p$$



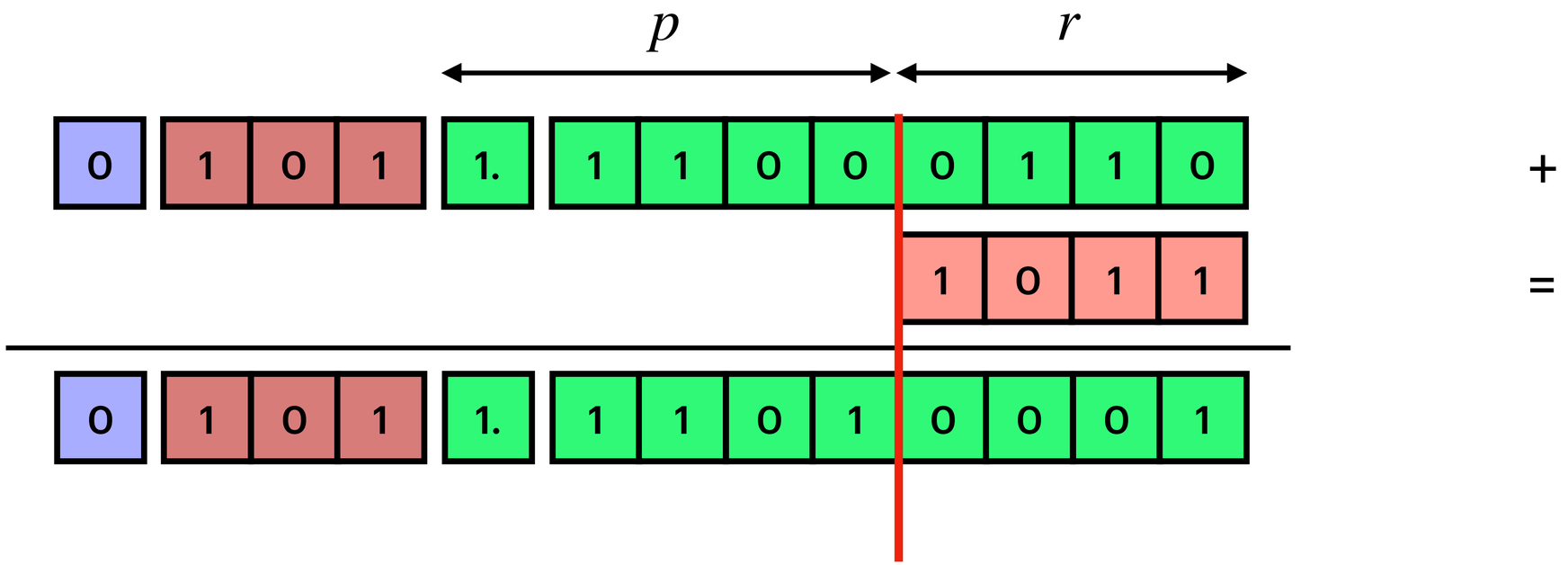
$$\mathrm{SR}_p(x) = \begin{cases} \lceil x \rceil_p, & \text{with prob. } q(x) \\ \lfloor x \rfloor_p, & \text{with prob. } 1 - q(x) \end{cases} \qquad q(x) = \frac{x - \lfloor x \rfloor_p}{\lceil x \rceil_p - \lfloor x \rfloor_p}$$

What do we need to implement it?

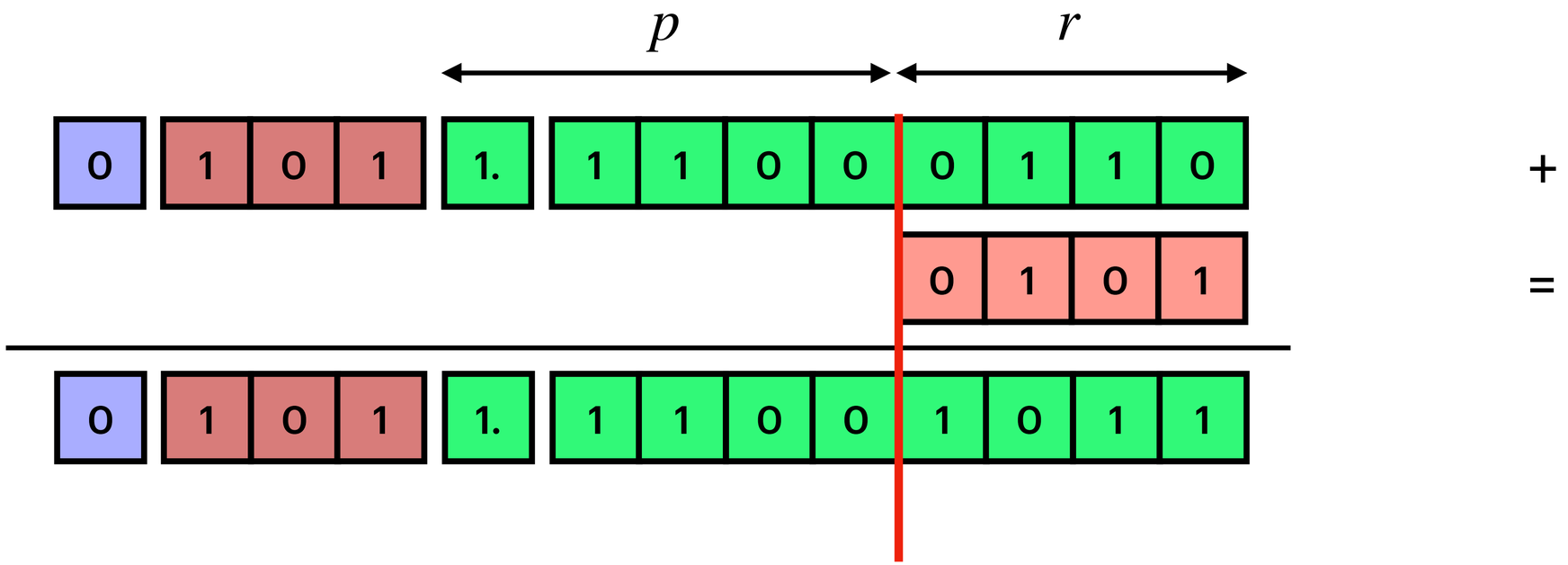
- A stream of random bits
- An accurate view of x (or x op y)



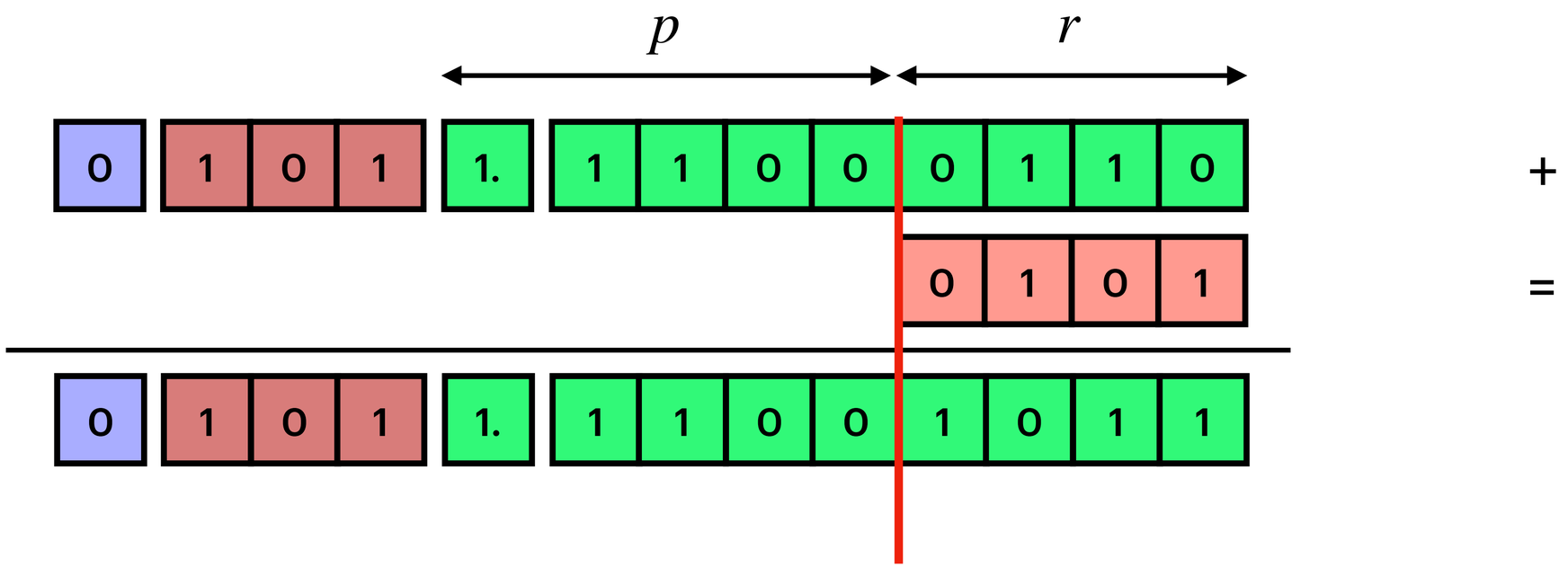
SR Example: E3M4 ($p = 5$)



SR Example: E3M4 ($p = 5$)

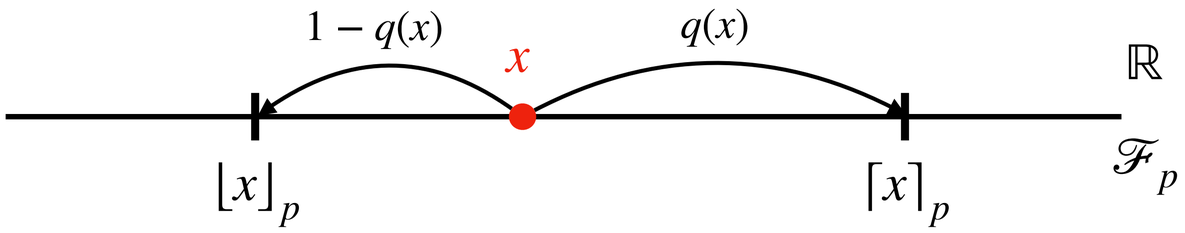


SR Example: E3M4 ($p = 5$)



How many random bits r (i.e., extra precision) are needed?

Limited Precision Stochastic Rounding



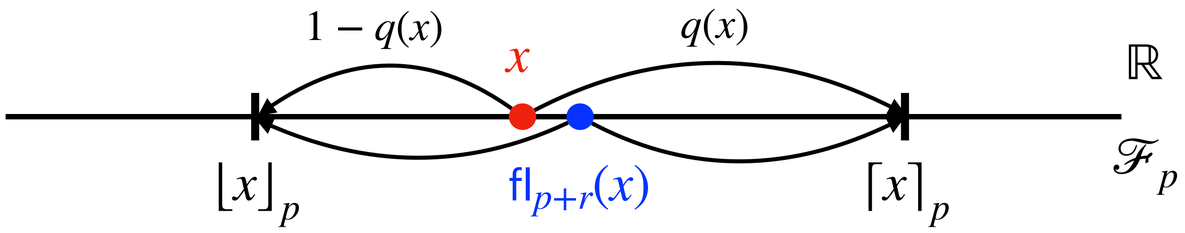
$$\mathrm{SR}_p(x) = \begin{cases} \lceil x \rceil_p, & \text{with prob. } q(x) \\ \lfloor x \rfloor_p, & \text{with prob. } 1 - q(x) \end{cases} \qquad q(x) = \frac{x - \lfloor x \rfloor_p}{\lceil x \rceil_p - \lfloor x \rfloor_p}$$

Exact

$\mathbb{E}(\mathrm{SR}_p(x)) = x$ (unbiased)

$\mathbb{V}(\mathrm{SR}_p(x)) \leqslant x^2 u_p^2 / 4$

Limited Precision Stochastic Rounding



$$\mathrm{SR}_p(x) = \begin{cases} \lceil x \rceil_p, & \text{with prob. } q(x) \\ \lfloor x \rfloor_p, & \text{with prob. } 1-q(x) \end{cases} \qquad q(x) = \frac{x - \lfloor x \rfloor_p}{\lceil x \rceil_p - \lfloor x \rfloor_p}$$

$$\mathrm{SR}_{p,r}(x) = \begin{cases} \lceil x \rceil_p, & \text{with prob. } q_r(x) \\ \lfloor x \rfloor_p, & \text{with prob. } 1-q_r(x) \end{cases} \qquad q_r(x) = \frac{\mathrm{fl}_{p+r}(x) - \lfloor x \rfloor_p}{\lceil x \rceil_p - \lfloor x \rfloor_p}$$

Exact

$\mathbb{E}(\mathrm{SR}_p(x)) = x$ (unbiased)

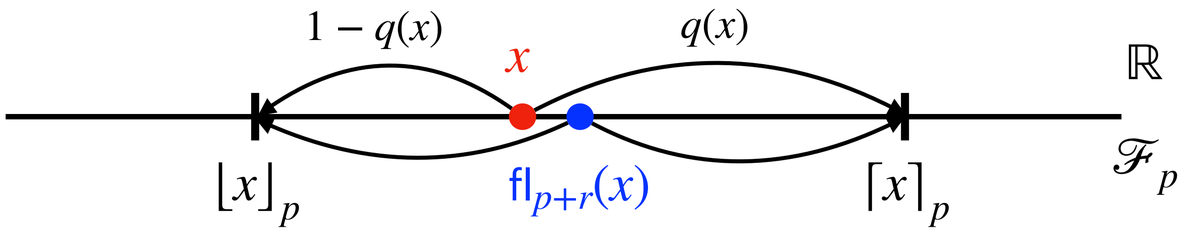
$\mathbb{V}(\mathrm{SR}_p(x)) \leqslant x^2 u_p^2/4$

Limited Precision

$\mathbb{E}(\mathrm{SR}_{p,r}(x)) = \mathrm{fl}_{p+r}(x)$ (biased)

$\mathbb{V}(\mathrm{SR}_{p,r}(x)) \leqslant x^2 u_p^2/4$

Limited Precision Stochastic Rounding



$$\mathrm{SR}_p(x) = \begin{cases} \lceil x \rceil_p, & \text{with prob. } q(x) \\ \lfloor x \rfloor_p, & \text{with prob. } 1-q(x) \end{cases} \qquad q(x) = \frac{x - \lfloor x \rfloor_p}{\lceil x \rceil_p - \lfloor x \rfloor_p}$$

$$\mathrm{SR}_{p,r}(x) = \begin{cases} \lceil x \rceil_p, & \text{with prob. } q_r(x) \\ \lfloor x \rfloor_p, & \text{with prob. } 1-q_r(x) \end{cases} \qquad q_r(x) = \frac{\mathrm{fl}_{p+r}(x) - \lfloor x \rfloor_p}{\lceil x \rceil_p - \lfloor x \rfloor_p}$$

- $\mathrm{fl}_{p+r}(x) = x(1+\beta)$, such that $|\beta| \leqslant u_{p+r}$
- $\mathrm{SR}_{p,r}(x) = x(1+\delta)$, such that $|\delta| \leqslant u_p$
- mean independence property is lost

$$\mathbb{E}(\delta_k \,|\, \delta_1, \ldots, \delta_{k-1}) = \beta_k \neq \mathbb{E}(\delta_k)$$

- β_k is a rv and $\mathbb{E}(\beta_k) = \mathbb{E}(\delta_k)$

Sufficient values for r

• Conversion of precision $t > p$ number	$t - p$
• Addition	$e_{\max} - e_{\min}$
• Multiplication	p
• Division	unbounded

How do smaller r affect the $O(\sqrt{n}u_p)$ bound?

Limited Precision Stochastic Rounding

We can show the following result about mean independence

Lemma [1]. Let $\delta_1, \ldots, \delta_n$ be random errors produced by a sequence of elementary ops using $\mathrm{SR}_{p,r}$ and let $\beta_1, \ldots, \beta_n$ be their corresponding errors incurred by fl_{p+r}.
Then, the rvs $\alpha_k = \delta_k - \beta_k$, for $1 \leqslant k \leqslant n$ are **mean independent**

$$\mathbb{E}(\alpha_k \,|\, \alpha_1, \ldots, \alpha_{k-1}) = \mathbb{E}(\alpha_k) = 0$$

Moreover, for all $1 \leqslant i \leqslant n$,

$$\prod_{k=i}^{n}(1+\delta_k) = \prod_{k=i}^{n}(1+\alpha_k) + \mathscr{B}_i,$$

with $|\mathscr{B}_i| \leqslant \gamma_{n-i+1}(u_p + u_{p+r}) - \gamma_{n-i+1}(u_p)$, where $\gamma_m(x) = (1+x)^m - 1$

[1] Probabilistic error analysis of limited-precision stochastic rounding, E.-M. El Arar, M. Fasi, S.-I. Filip, M. Mikaitis, arXiv 2408.03069, to appear in SISC

Limited Precision Stochastic Rounding

Theorem [1]. If $y = a^T b$, where $a, b \in \mathbb{R}^n$, is evaluated using $\mathrm{SR}_{p,r}$, then the computed result $\widehat{y}$ satisfies

$$\frac{|y - \widehat{y}|}{|y|} \leqslant \frac{|a|^T |b|}{|a^T b|} \left(\lambda \sqrt{n} u_p + n u_{p+r} + O(u_p^2 + u_{p+r}^2) \right)$$

with probability at least $1 - 2\exp(-\lambda^2/2)$.

[1] Probabilistic error analysis of limited-precision stochastic rounding, E.-M. El Arar, M. Fasi, S.-I. Filip, M. Mikaitis, arXiv 2408.03069, to appear in SISC

Limited Precision Stochastic Rounding

Theorem [1]. If $y = a^T b$, where $a, b \in \mathbb{R}^n$, is evaluated using $\mathrm{SR}_{p,r}$, then the computed result $\widehat{y}$ satisfies

$$\frac{|y - \widehat{y}|}{|y|} \leqslant \frac{|a|^T |b|}{|a^T b|} \left(\lambda \sqrt{n} u_p + n u_{p+r} + O(u_p^2 + u_{p+r}^2) \right)$$

with probability at least $1 - 2\exp(-\lambda^2/2)$.

Proof Idea.

$$\prod_{k=i}^{n} (1 + \delta_k) = \prod_{k=i}^{n} (1 + \alpha_k) + \mathscr{B}_i,$$

[1] Probabilistic error analysis of limited-precision stochastic rounding, E.-M. El Arar, M. Fasi, S.-I. Filip, M. Mikaitis, arXiv 2408.03069, to appear in SISC

Limited Precision Stochastic Rounding

Theorem [1]. If $y = a^T b$, where $a, b \in \mathbb{R}^n$, is evaluated using $\mathrm{SR}_{p,r}$, then the computed result $\widehat{y}$ satisfies

$$\frac{|y - \widehat{y}|}{|y|} \leqslant \frac{|a|^T |b|}{|a^T b|} \left(\lambda \sqrt{n} u_p + n u_{p+r} + O(u_p^2 + u_{p+r}^2) \right)$$

with probability at least $1 - 2\exp(-\lambda^2/2)$.

Proof Idea.

$$\prod_{k=i}^{n} (1 + \delta_k) = \prod_{k=i}^{n} (1 + \alpha_k) + \mathscr{B}_i,$$

Martingale argument

$$O(\sqrt{n} u_p)$$

$$|\mathscr{B}_i| \leqslant \gamma_{n-i+1}(u_p + u_{p+r}) - \gamma_{n-i+1}(u_p)$$

$$O(n u_{p+r})$$

$$O(\sqrt{n} u_p + n u_{p+r})$$

[1] Probabilistic error analysis of limited-precision stochastic rounding, E.-M. El Arar, M. Fasi, S.-I. Filip, M. Mikaitis, arXiv 2408.03069, to appear in SISC

Limited Precision Stochastic Rounding

Theorem [1]. If $y = a^T b$, where $a, b \in \mathbb{R}^n$, is evaluated using $\mathrm{SR}_{p,r}$, then the computed result $\widehat{y}$ satisfies

$$\frac{|y - \widehat{y}|}{|y|} \leqslant \frac{|a|^T |b|}{|a^T b|} \left(\lambda \sqrt{n} u_p + n u_{p+r} + O(u_p^2 + u_{p+r}^2) \right)$$

with probability at least $1 - 2\exp(-\lambda^2/2)$.

$$O(\sqrt{n} u_p + n u_{p+r})$$

Good rule of thumb for choosing r in length n summation chains:

$$\sqrt{n} u_p \geqslant n u_{p+r} \Rightarrow r \geqslant \lceil (\log_2 n)/2 \rceil$$

[1] Probabilistic error analysis of limited-precision stochastic rounding, E.-M. El Arar, M. Fasi, S.-I. Filip, M. Mikaitis, arXiv 2408.03069, to appear in SISC

Numerical Experiments

- Recursive summation (srfloat)
- Rosenbrock function minimisation (srfloat)
- Parameter update in DNN training (mptorch)

SR$_{p,r}$ simulation tools:

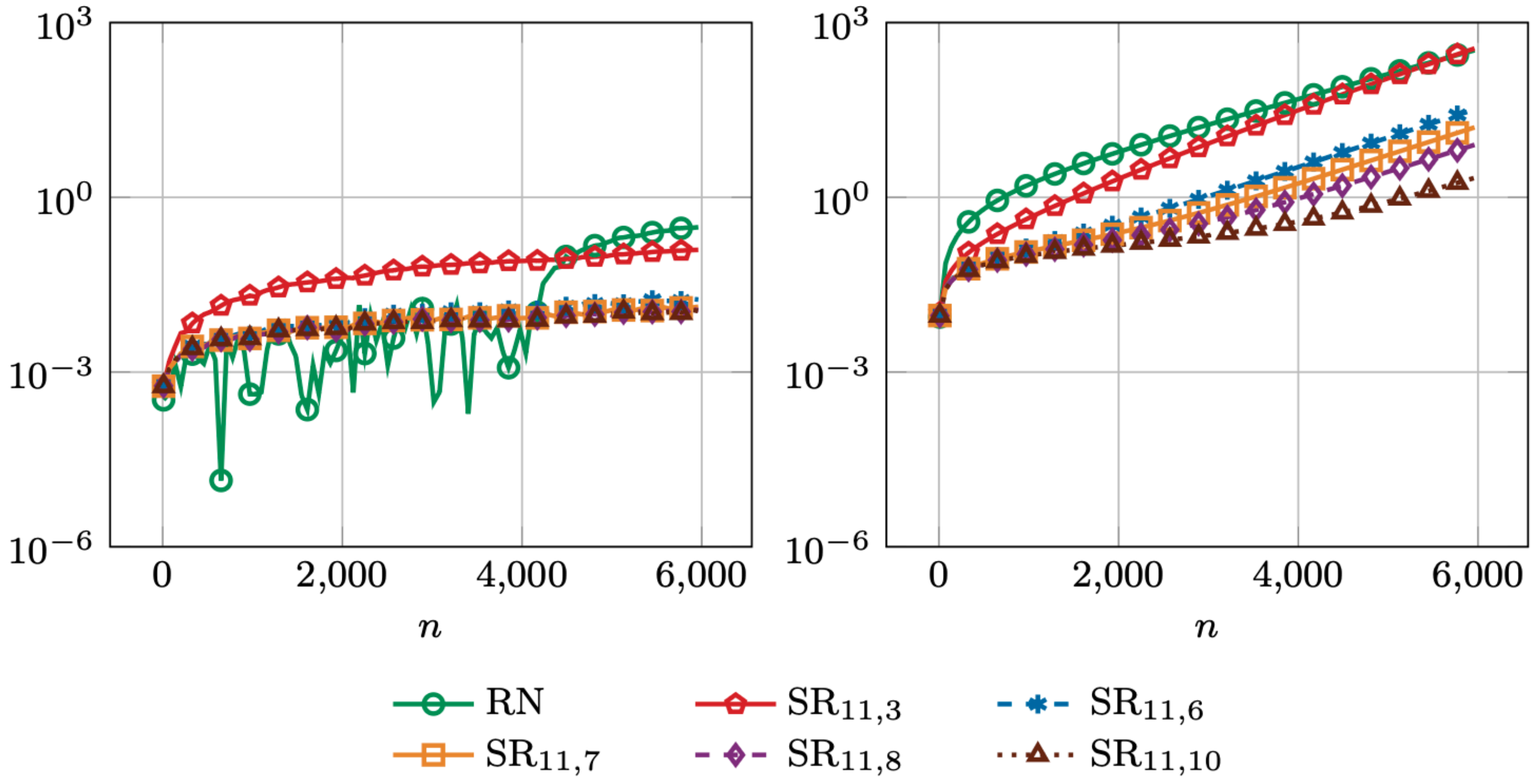
- https://github.com/sfilip/srfloat
- https://github.com/mptorch/mptorch

Recursive Summation

- binary16 ($p = 11$) arithmetic
- elements sampled from $U([0,1])$
- r close to $\lceil (\log_2 6{,}000)/2 \rceil = 7$ works well

Relative forward error

Bounds with $1 - 2\exp(-\lambda^2/2) = 0.9$



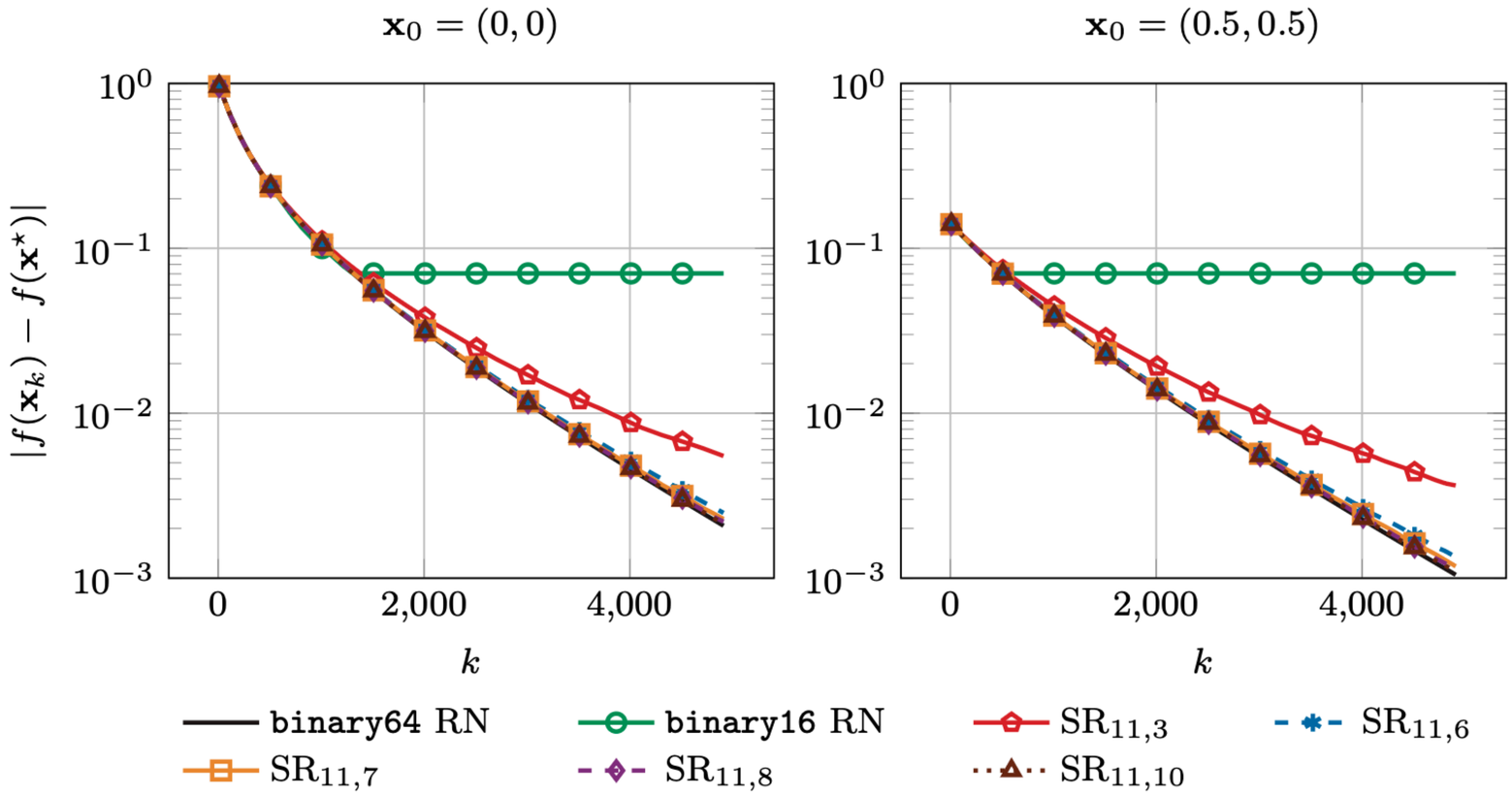
Rosenbrock Function

$$f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2,$$

- GD update rule ($t_k = 0.001$):

$$\mathbf{x}_{k+1} = \mathbf{x}_k - t_k \nabla f(\mathbf{x}_k),$$

- r close to $\lceil (\log_2 6{,}000)/2 \rceil = 7$ works well



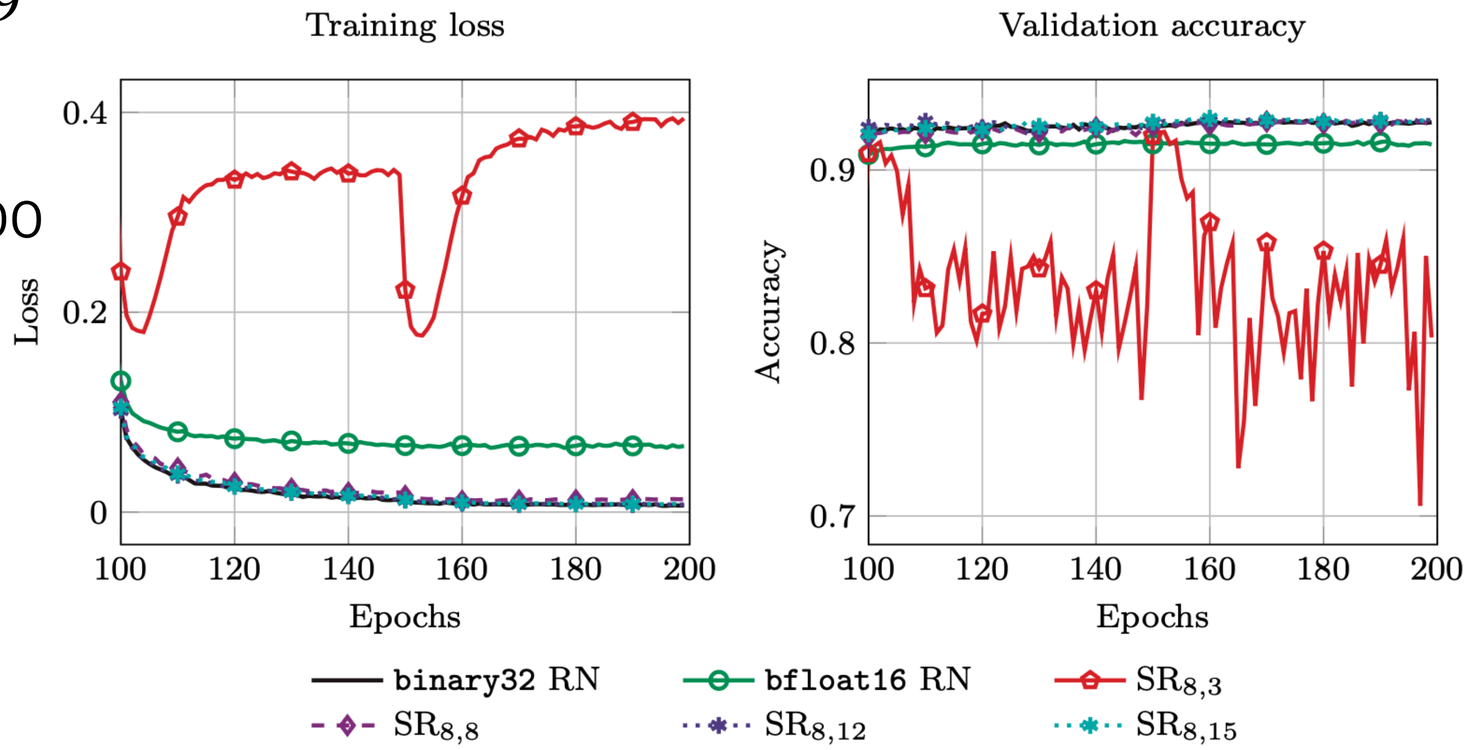
Parameter Updates in DNN Training

- Training a ResNet32 model on CIFAR-10
- Hyperparameters
  - Batch size: 128, momentum: $\mu = 0.9$
  - Training time: 64,000 iterations (200 epochs)
  - LR: $t_k = 0.1$, reduced by 10 at 32,000 and 48,000 iterations

- bfloat16 ($p = 8$) arithmetic
- Update rule:

$$\mathbf{v}_{k+1} = \circ(\mu \mathbf{v}_k + \mathbf{g}_k),$$
$$\mathbf{x}_{k+1} = \circ(\mathbf{x}_k - t_k \mathbf{v}_{k+1}),$$

$$r \geqslant \left\lceil (\log_2 64{,}000)/2 \right\rceil = 8$$



MPTorch: Simulation Support for MP Training

➡library/simulation support for low/mixed precision training is limited

- **QPyTorch** [1]: applies quantization on operands for FP32 arithmetic kernels

Limitation: not an adequate simulation of low precision training hardware

[1] QPyTorch: A Low-Precision Arithmetic Simulation Framework, *Zhang et al.,* arXiv:1910.04540, 2019
[2] MPTorch and MPArchimedes: Open Source Frameworks to Explore Custom Mixed-Precision Operations for DNN Training on Edge Devices, Tatsumi et al., ROAD4NN 2021
[3] MPTorch-FPGA: a Custom Mixed-Precision Framework for FPGA-based DNN Training,*S. Ben Ali, S.-I. Filip, O. Sentieys. G. Lemieux*, IEEE/ACM DATE conference, Mar 2025

MPTorch: Simulation Support for MP Training

➡library/simulation support for low/mixed precision training is limited

- **QPyTorch** [1]: applies quantization on operands for FP32 arithmetic kernels

Limitation: not an adequate simulation of low precision training hardware

MPTorch [2,3]

➡PyTorch-based extension for low/mixed precision training (& inference) simulation

- mix of C++, CUDA C++, HLS, and Python
- **precision and format at the operator level** for **inference** (FWD) and **training** (BWD)
- facilitate research into low/mixed precision DNN training accelerators

➡ support for various number formats and rounding modes:

- floating-point, fixed-point, block floating-point, logarithmic,...
- round to nearest, stochastic rounding,...

➡ also: Transformer-based architectures + IEEE-P3109 WG FP8 formats

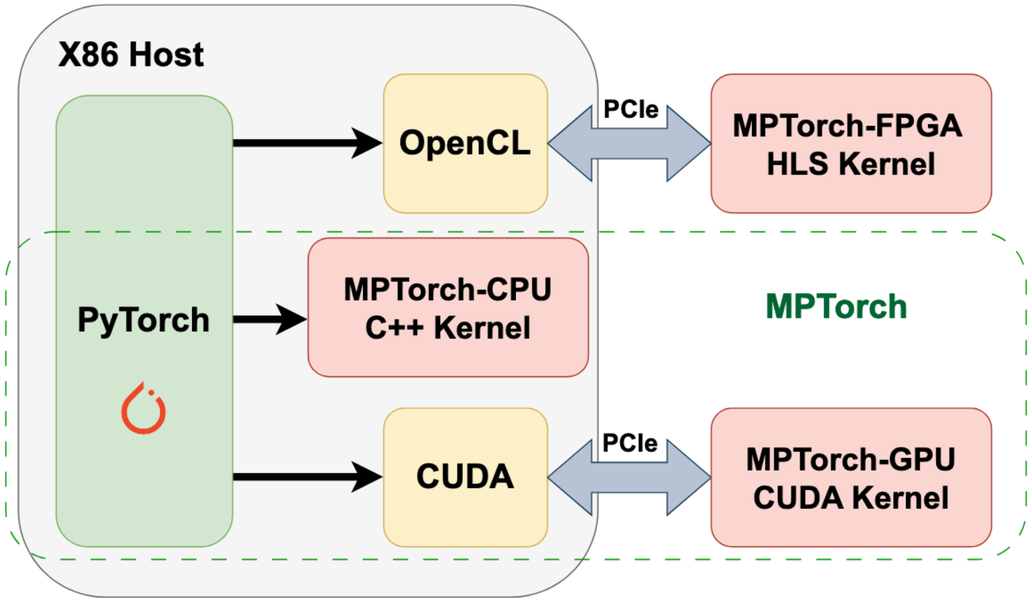
[1] QPyTorch: A Low-Precision Arithmetic Simulation Framework, *Zhang et al.,* arXiv:1910.04540, 2019
[2] MPTorch and MPArchimedes: Open Source Frameworks to Explore Custom Mixed-Precision Operations for DNN Training on Edge Devices, Tatsumi et al., ROAD4NN 2021
[3] MPTorch-FPGA: a Custom Mixed-Precision Framework for FPGA-based DNN Training,*S. Ben Ali, S.-I. Filip, O. Sentieys. G. Lemieux*, IEEE/ACM DATE conference, Mar 2025

MPTorch: Simulation Support for MP Training

➡library/simulation support for low/mixed precision training is limited

- **QPyTorch** [1]: applies quantization on operands for FP32 arithmetic kernels

Limitation: not an adequate simulation of low precision training hardware

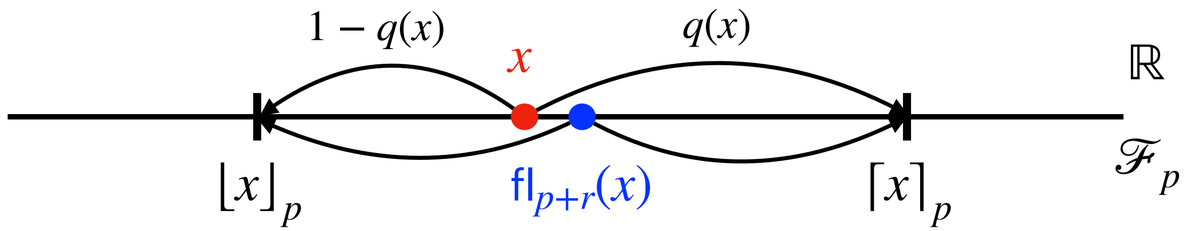


[1] QPyTorch: A Low-Precision Arithmetic Simulation Framework, *Zhang et al.*, arXiv:1910.04540, 2019

[2] MPTorch and MPArchimedes: Open Source Frameworks to Explore Custom Mixed-Precision Operations for DNN Training on Edge Devices, Tatsumi et al., ROAD4NN 2021

[3] MPTorch-FPGA: a Custom Mixed-Precision Framework for FPGA-based DNN Training,*S. Ben Ali, S.-I. Filip, O. Sentieys. G. Lemieux*, IEEE/ACM DATE conference, Mar 2025

Limited Precision Stochastic Rounding [1]



$$\mathrm{SR}_p(x) = \begin{cases} \lceil x \rceil_p, & \text{with prob. } q(x) \\ \lfloor x \rfloor_p, & \text{with prob. } 1-q(x) \end{cases} \qquad q(x) = \frac{x - \lfloor x \rfloor_p}{\lceil x \rceil_p - \lfloor x \rfloor_p}$$

$$\mathrm{SR}_{p,r}(x) = \begin{cases} \lceil x \rceil_p, & \text{with prob. } q_r(x) \\ \lfloor x \rfloor_p, & \text{with prob. } 1-q_r(x) \end{cases} \qquad q_r(x) = \frac{\mathsf{fl}_{p+r}(x) - \lfloor x \rfloor_p}{\lceil x \rceil_p - \lfloor x \rfloor_p}$$

Exact

$\mathbb{E}(\mathrm{SR}_p(x)) = x$ (unbiased)

$\mathbb{V}(\mathrm{SR}_p(x)) \leqslant x^2 u_p^2/4$

$O(\sqrt{n}u_p)$ bounds

Limited Precision

$\mathbb{E}(\mathrm{SR}_{p,r}(x)) = \mathsf{fl}_{p+r}(x)$ (biased)

$\mathbb{V}(\mathrm{SR}_{p,r}(x)) \leqslant x^2 u_p^2/4$

$O(\sqrt{n}u_p + nu_{p+r})$ bounds

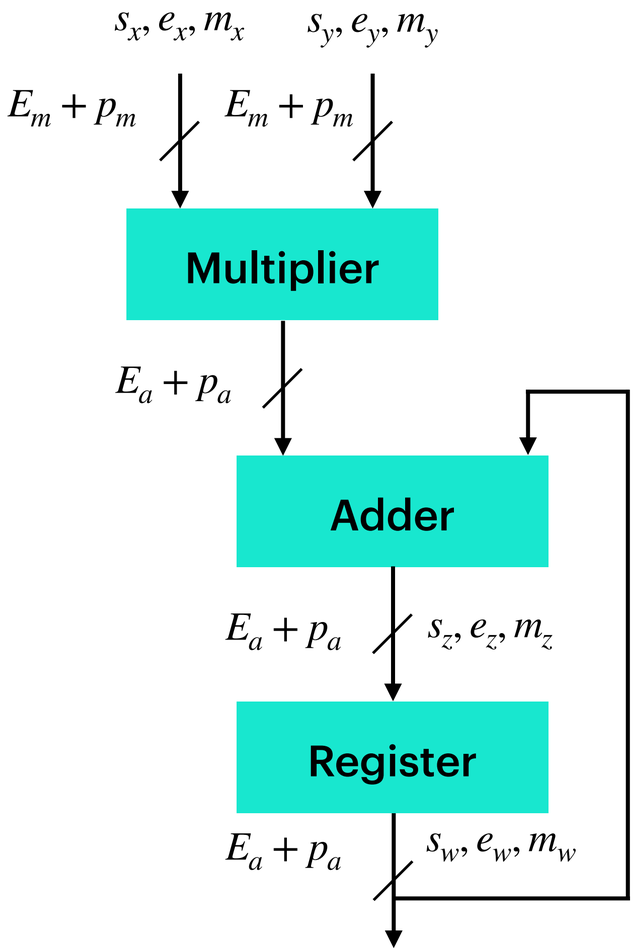
$r \geqslant \lceil (\log_2 n)/2 \rceil$

Thank you! Questions?

[1] Probabilistic error analysis of limited-precision stochastic rounding, E.-M. El Arar, M. Fasi, S.-I. Filip, M. Mikaitis, arXiv 2408.03069

Classic MAC Unit Design

Multiply-Accumulate (MAC) Unit Design

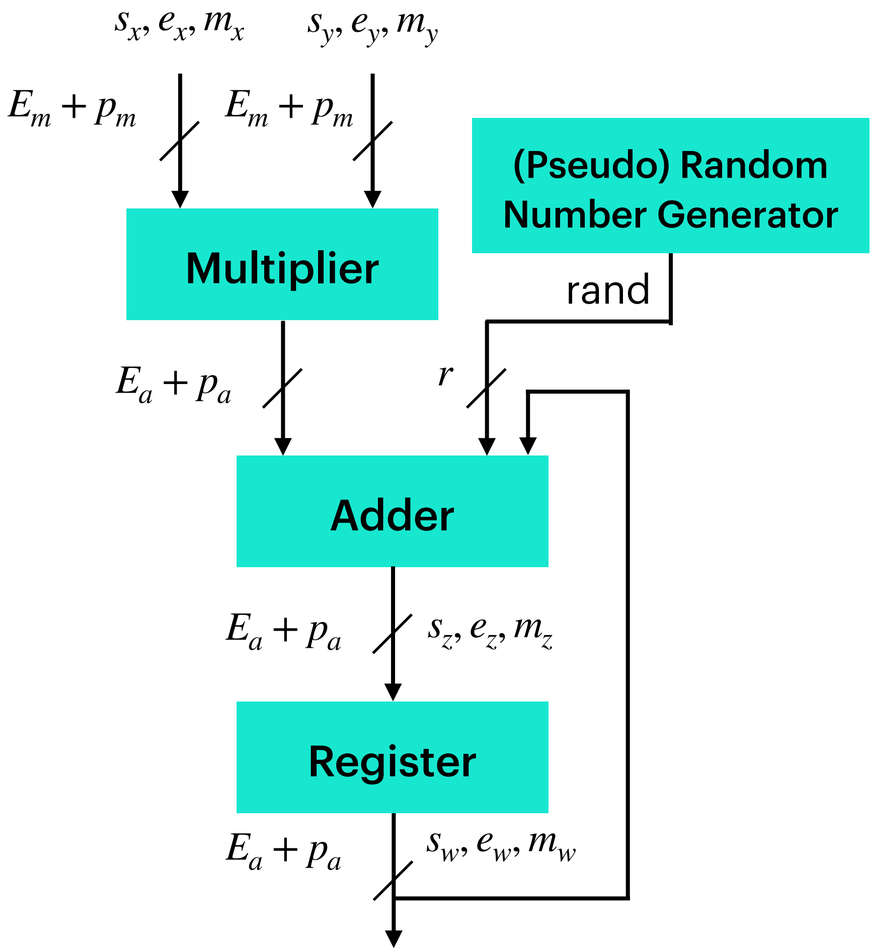


$$z = xy + z$$

Central to performing dot product and matrix multiplication

Classic MAC Unit Design with SR

Multiply-Accumulate (MAC) Unit Design

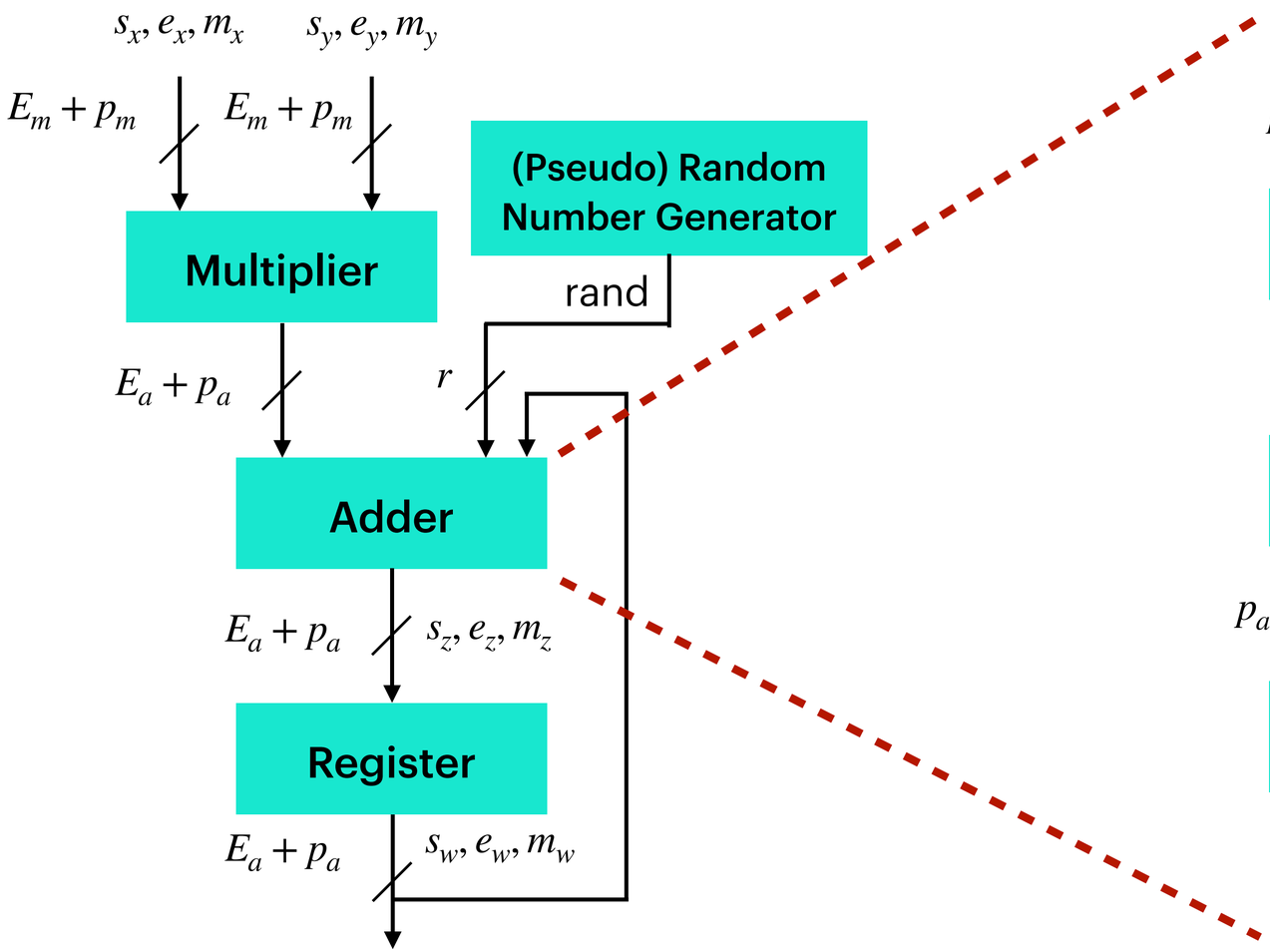


$$z = \mathrm{SR}\left(xy + w\right)$$

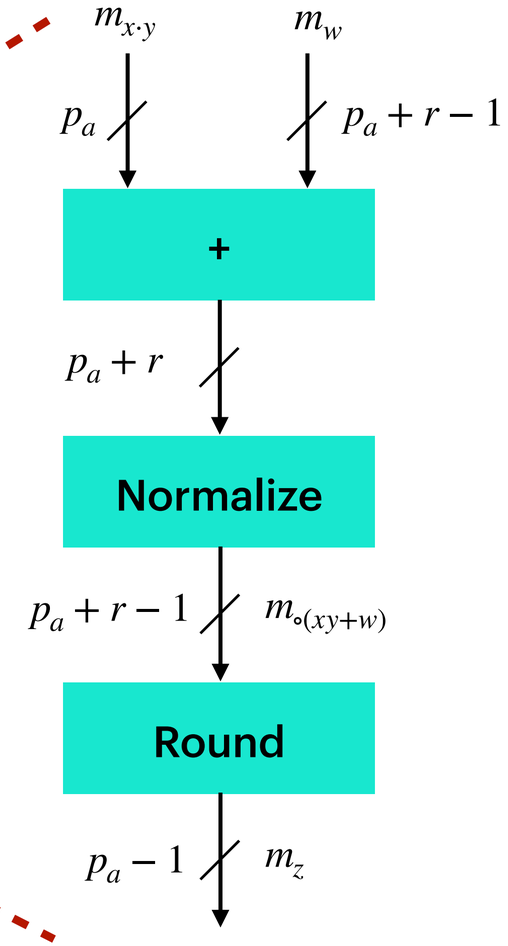
Central to performing dot product and matrix multiplication

Classic MAC Unit Design with SR

Multiply-Accumulate (MAC) Unit Design



Adder Architecture

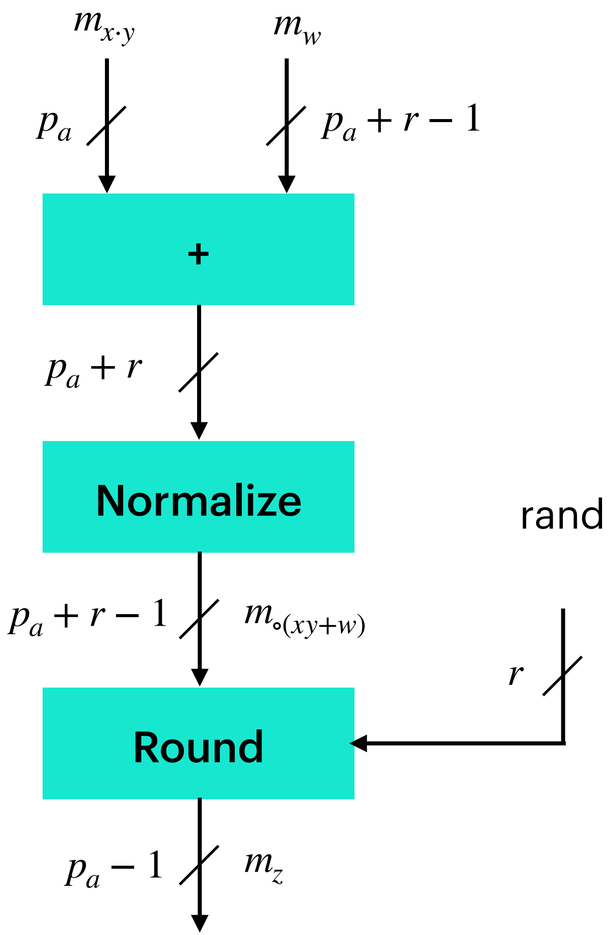


$$z = \mathrm{SR}\left(xy + w\right)$$

Central to performing dot product and matrix multiplication

Classic MAC Unit Design with SR

Classic SR Design



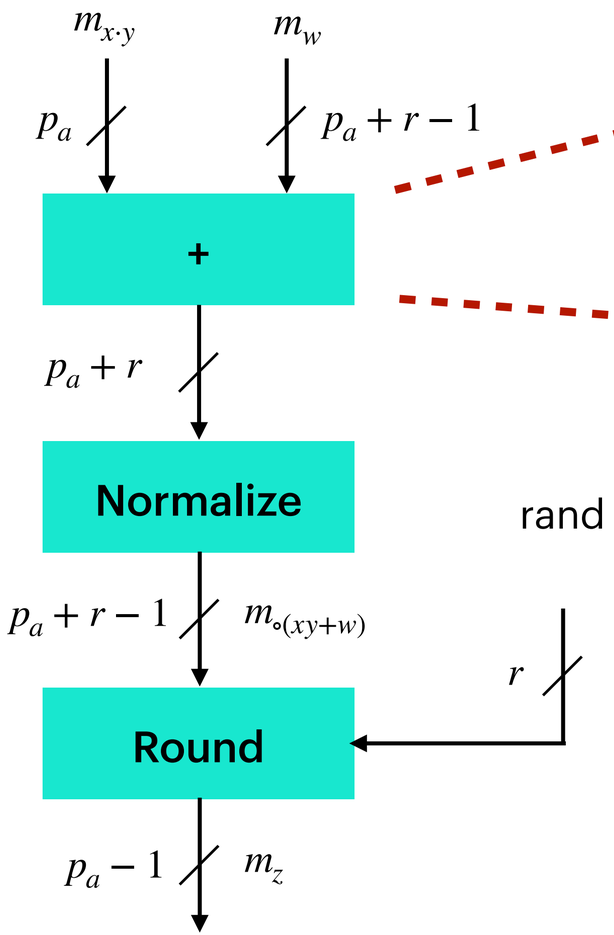
Toy Example: $a + b$

$$a = 1.111_{(2)} \cdot 2^8$$

$$b = 1.101_{(2)} \cdot 2^5$$

Classic MAC Unit Design with SR

Classic SR Design



Toy Example: $a + b$

$$a = 1.111_{(2)} \cdot 2^8 \qquad b = 1.101_{(2)} \cdot 2^5$$

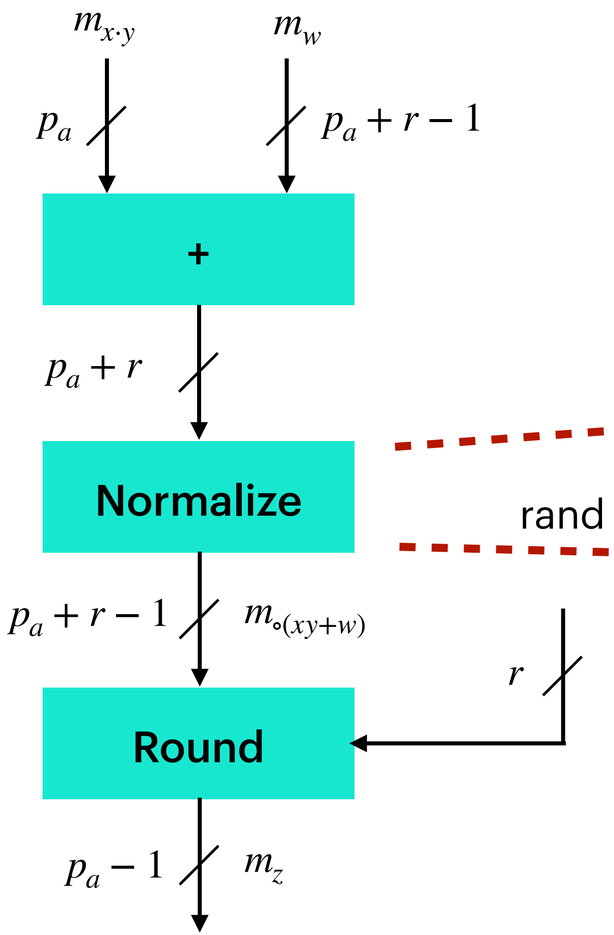
significand alignment →

$$a = 1.111000_{(2)} \cdot 2^8 \qquad b = 0.001101_{(2)} \cdot 2^8$$

significand addition $a + b = 10.000101_{(2)} \cdot 2^8$

Classic MAC Unit Design with SR

Classic SR Design



Toy Example: $a + b$

$$a = 1.111_{(2)} \cdot 2^8 \quad \xrightarrow{\text{significand alignment}} \quad a = 1.111000_{(2)} \cdot 2^8$$

$$b = 1.101_{(2)} \cdot 2^5 \qquad\qquad b = 0.001101_{(2)} \cdot 2^8$$

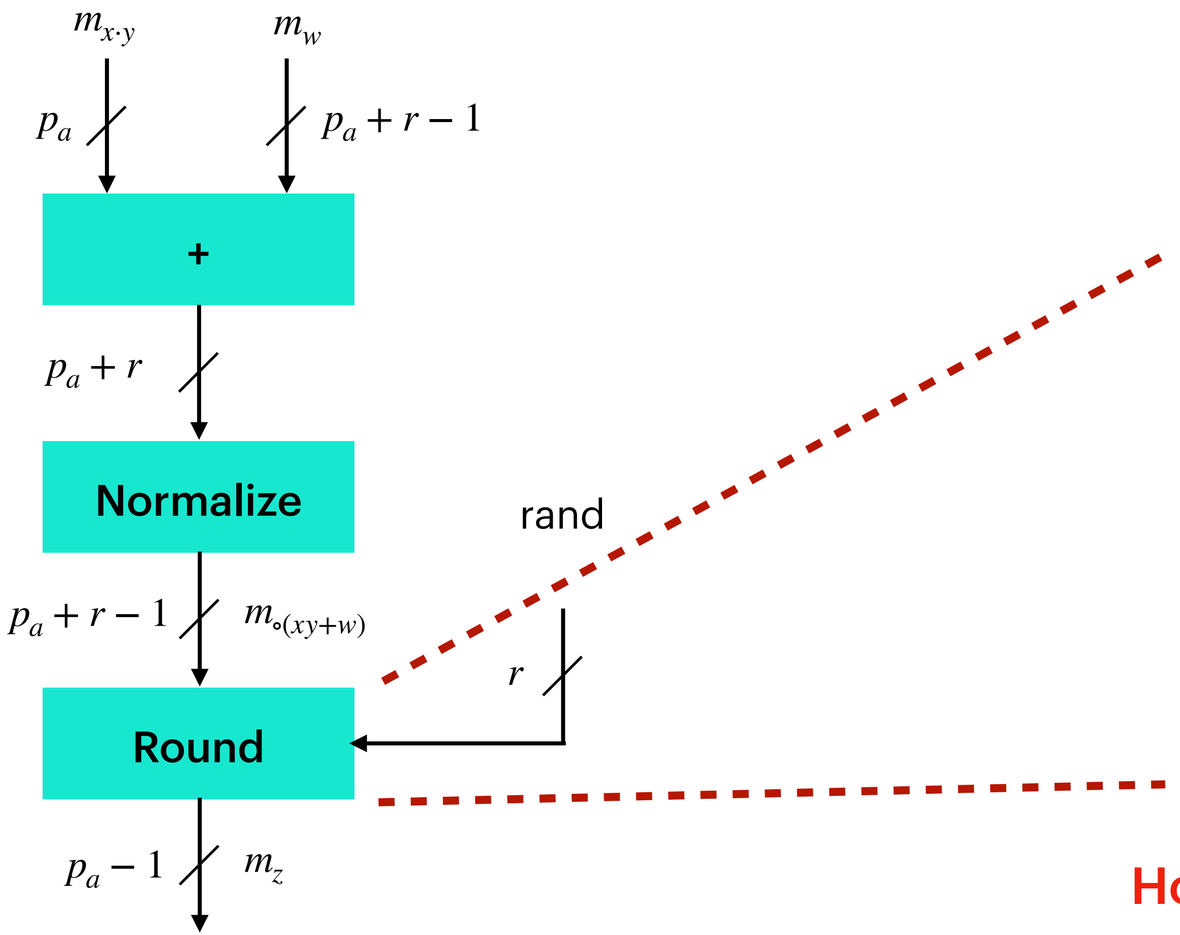
significand addition $\quad a + b = 10.000101_{(2)} \cdot 2^8$

normalization

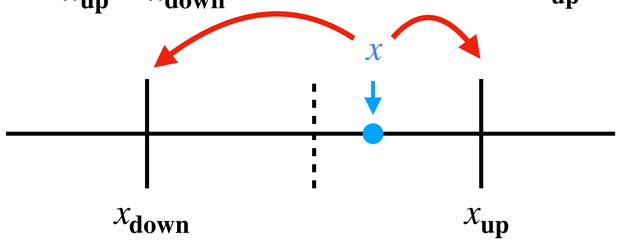
$$a + b = 1.0000101_{(2)} \cdot 2^9$$

Classic MAC Unit Design with SR

Classic SR Design



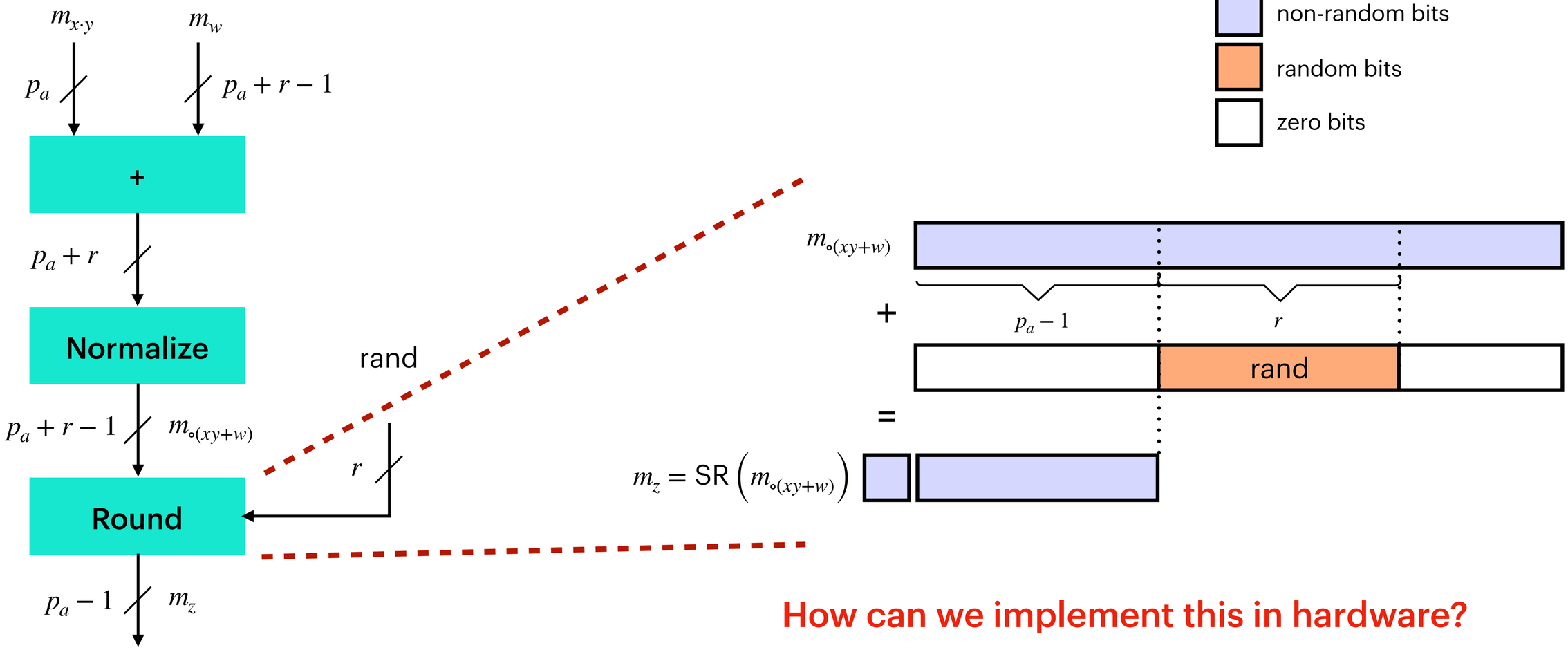
$$p_{\text{down}} = \frac{x_{\text{up}} - x}{x_{\text{up}} - x_{\text{down}}} \qquad p_{\text{up}} = \frac{x - x_{\text{down}}}{x_{\text{up}} - x_{\text{down}}}$$



How can we implement this in hardware?

Classic MAC Unit Design with SR

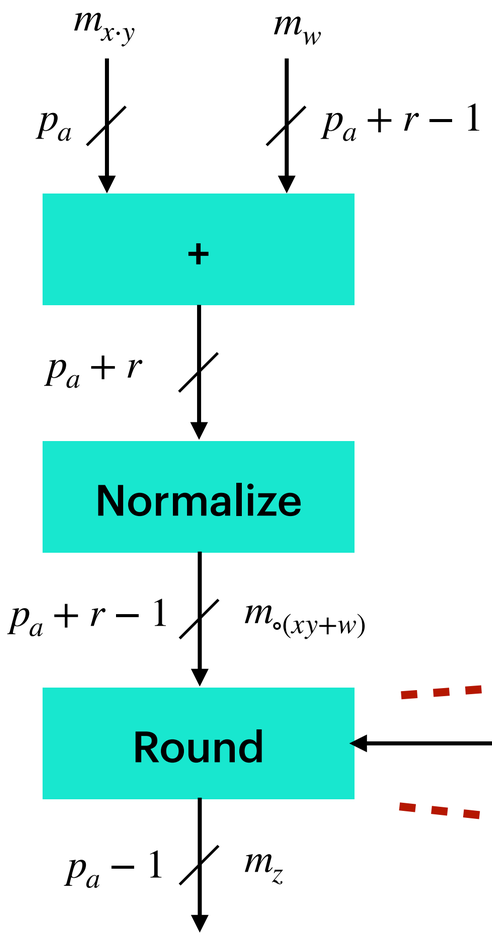
Classic SR Design



How can we implement this in hardware?

Classic MAC Unit Design with SR

Classic SR Design



rand

r

Toy Example: $a + b$

$$a = 1.111_{(2)} \cdot 2^8$$
$$b = 1.101_{(2)} \cdot 2^5$$

significand alignment

$$a = 1.111000_{(2)} \cdot 2^8$$
$$b = 0.001101_{(2)} \cdot 2^8$$

significand addition

$$a + b = 10.000101_{(2)} \cdot 2^8$$

normalization

$$a + b = 1.0000101_{(2)} \cdot 2^9$$

add random bits

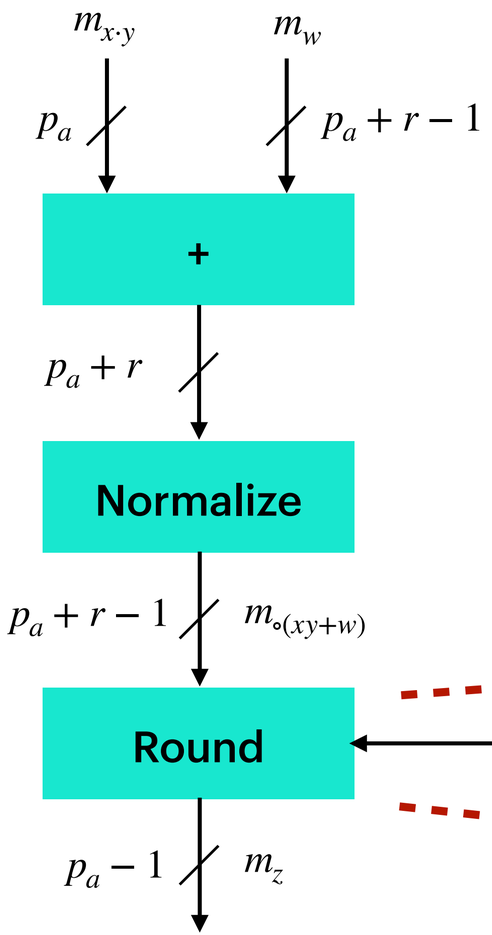
$$a + b = 1.0000101_{(2)} \cdot 2^9$$
$$\textbf{rand} = 0.0001101_{(2)} \cdot 2^9$$

round

$$\textbf{SR}(a + b) = 1.001_{(2)} \cdot 2^9$$

Classic MAC Unit Design with SR

Classic SR Design



rand

r

Toy Example: $a + b$

$$a = 1.111_{(2)} \cdot 2^8$$
$$b = 1.101_{(2)} \cdot 2^5$$

significand alignment

$$a = 1.111000_{(2)} \cdot 2^8$$
$$b = 0.001101_{(2)} \cdot 2^8$$

significand addition

$$a + b = 10.000101_{(2)} \cdot 2^8$$

normalization

$$a + b = 1.0000101_{(2)} \cdot 2^9$$

add random bits

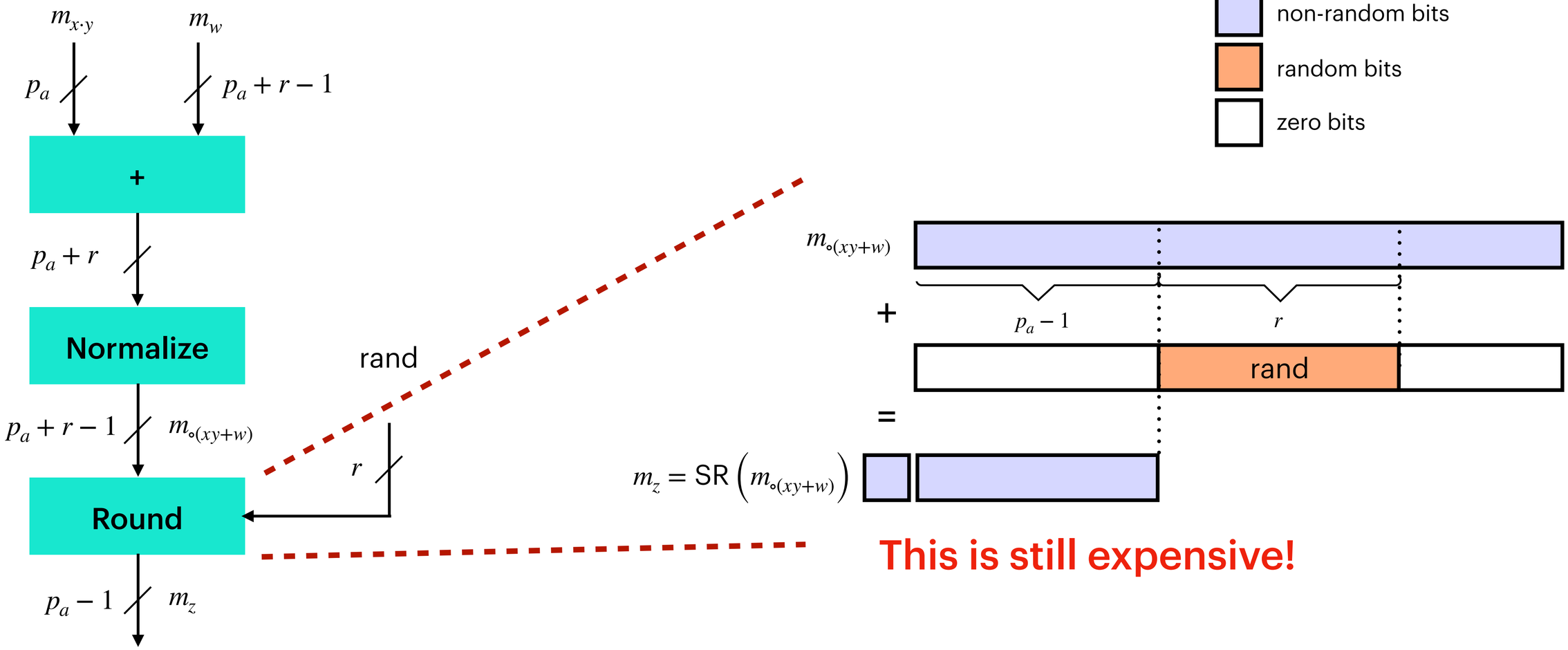
$$a + b = 1.0000101_{(2)} \cdot 2^9$$
$$\textbf{rand} = 0.0000110_{(2)} \cdot 2^9$$

round

$$\textbf{SR}(a + b) = 1.000_{(2)} \cdot 2^9$$

Classic MAC Unit Design with SR

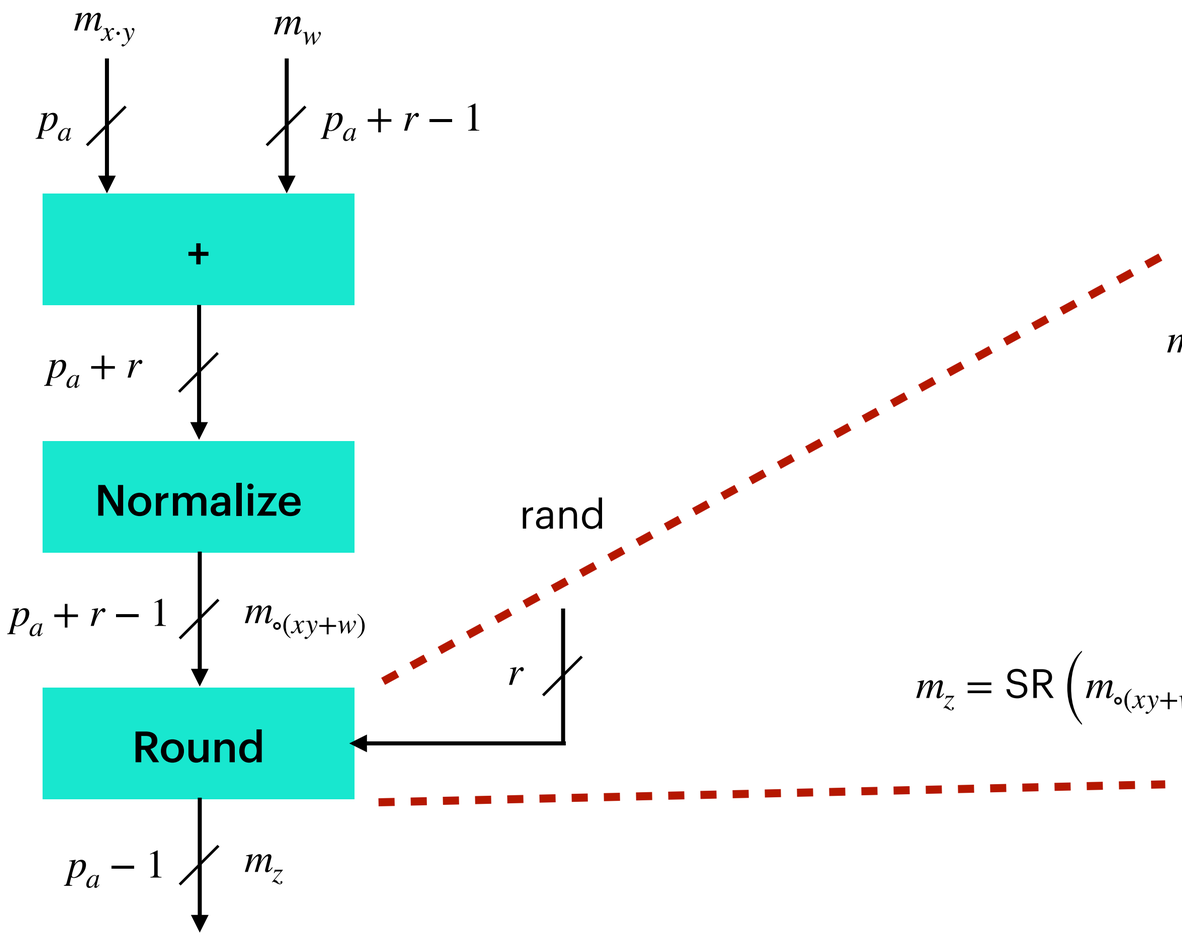
Classic SR Design

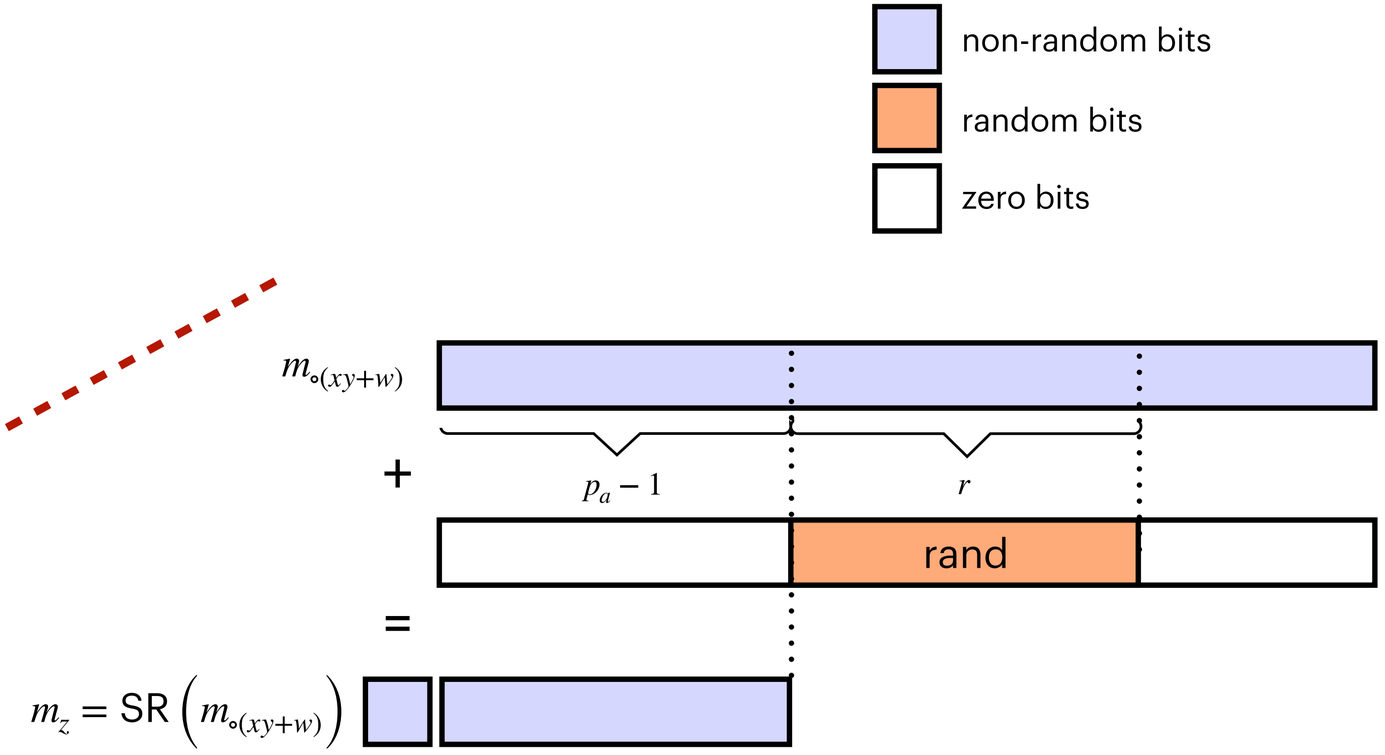


This is still expensive!

Classic MAC Unit Design with SR

Classic SR Design (lazy)





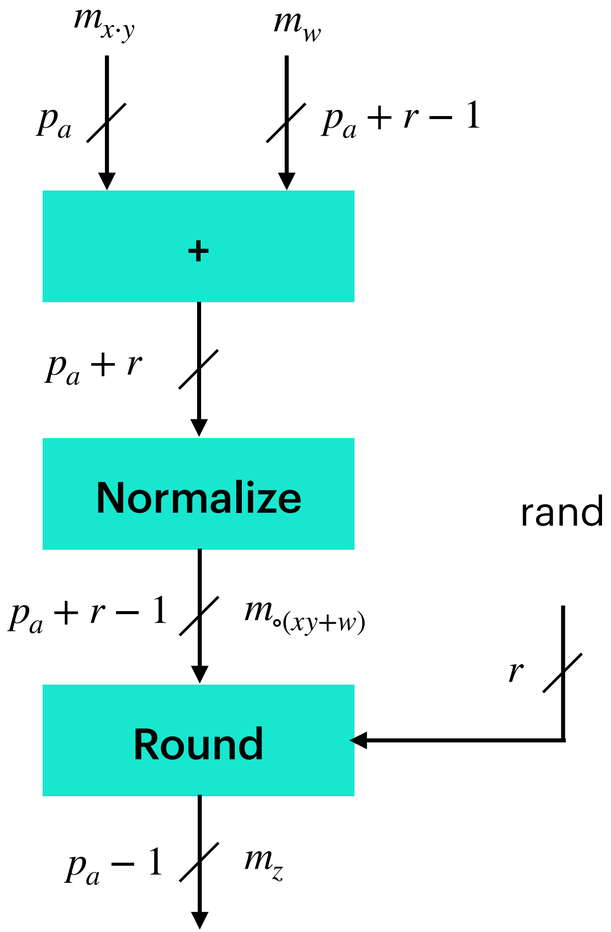
This is still expensive!

Idea: start rounding process early

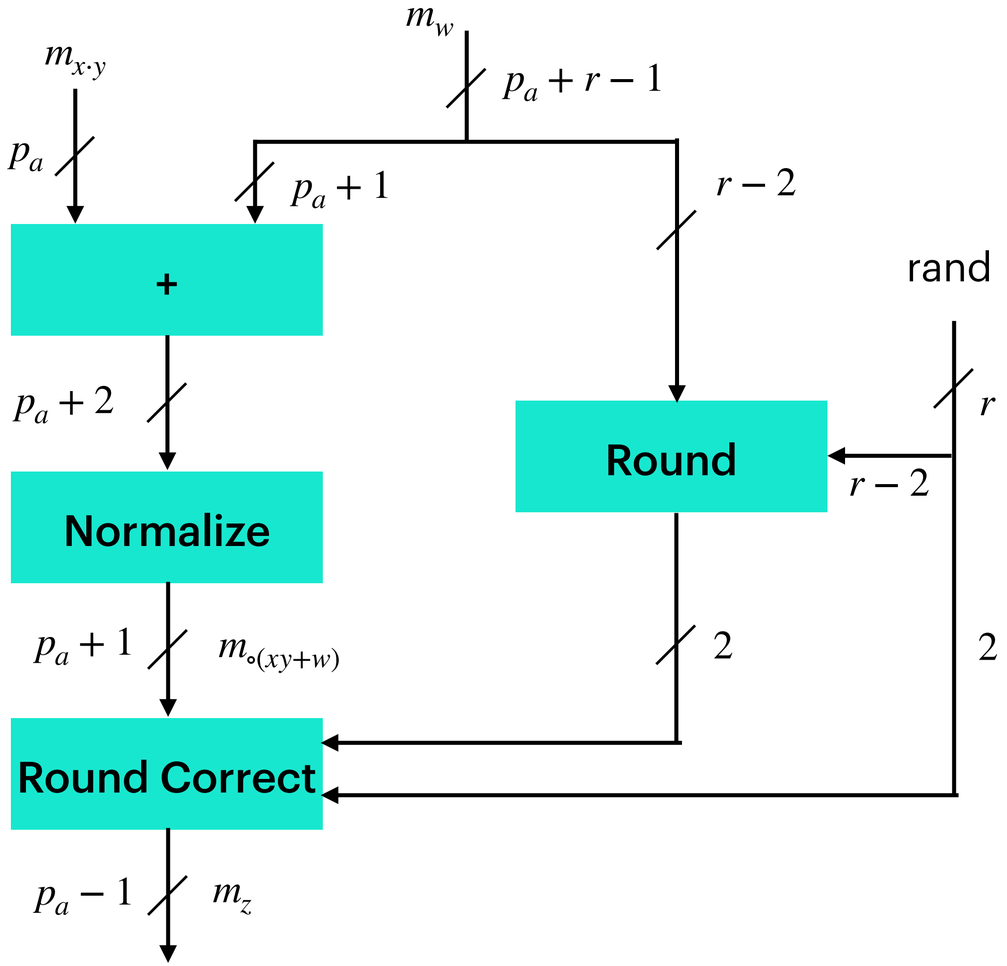
[1] A Stochastic Rounding-Enabled Low-Precision Floating-Point MAC for DNN Training, *S. Ben Ali, S.-I. Filip, O. Sentieys.*, in proceedings of the 2024 IEEE/ACM Design, Automation and Test in Europe (DATE) conference, Mar 2024

Eager MAC Unit Design with SR

Classic SR Design (lazy)



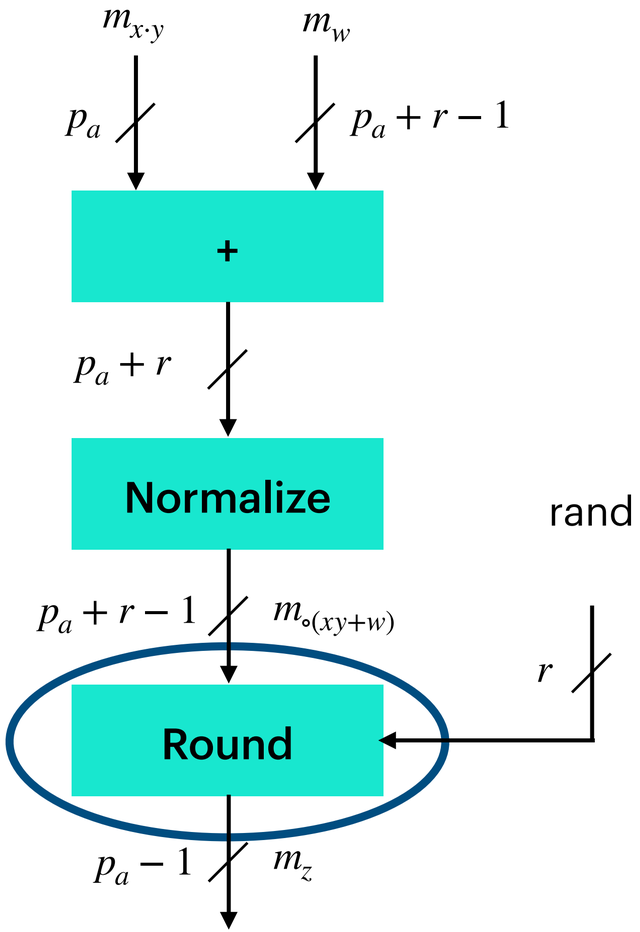
Proposed SR Design (eager)



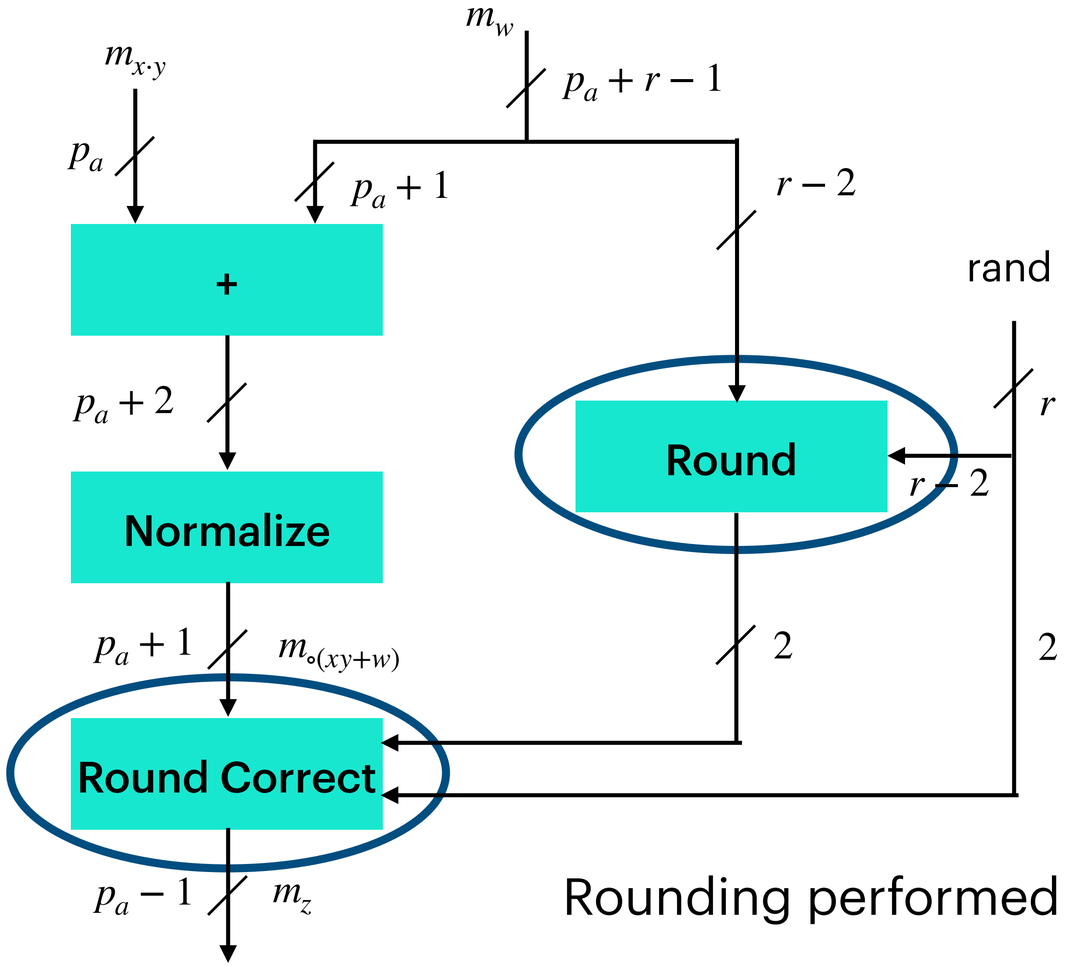
[1] A Stochastic Rounding-Enabled Low-Precision Floating-Point MAC for DNN Training, *S. Ben Ali, S.-I. Filip, O. Sentieys.*, in proceedings of the 2024 IEEE/ACM Design, Automation and Test in Europe (DATE) conference, Mar 2024

Eager MAC Unit Design with SR

Classic SR Design (lazy)



Proposed SR Design (eager)

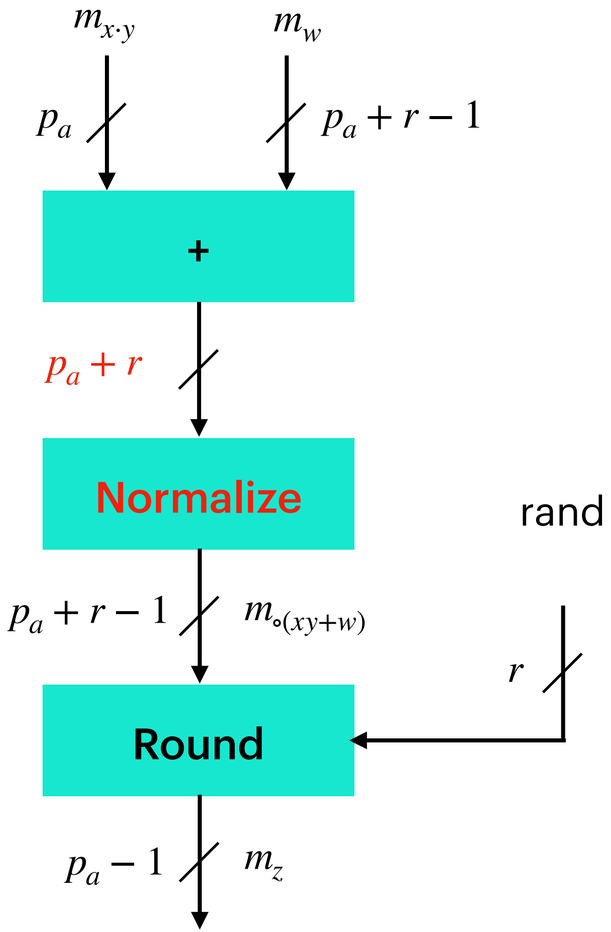


Rounding performed in two stages

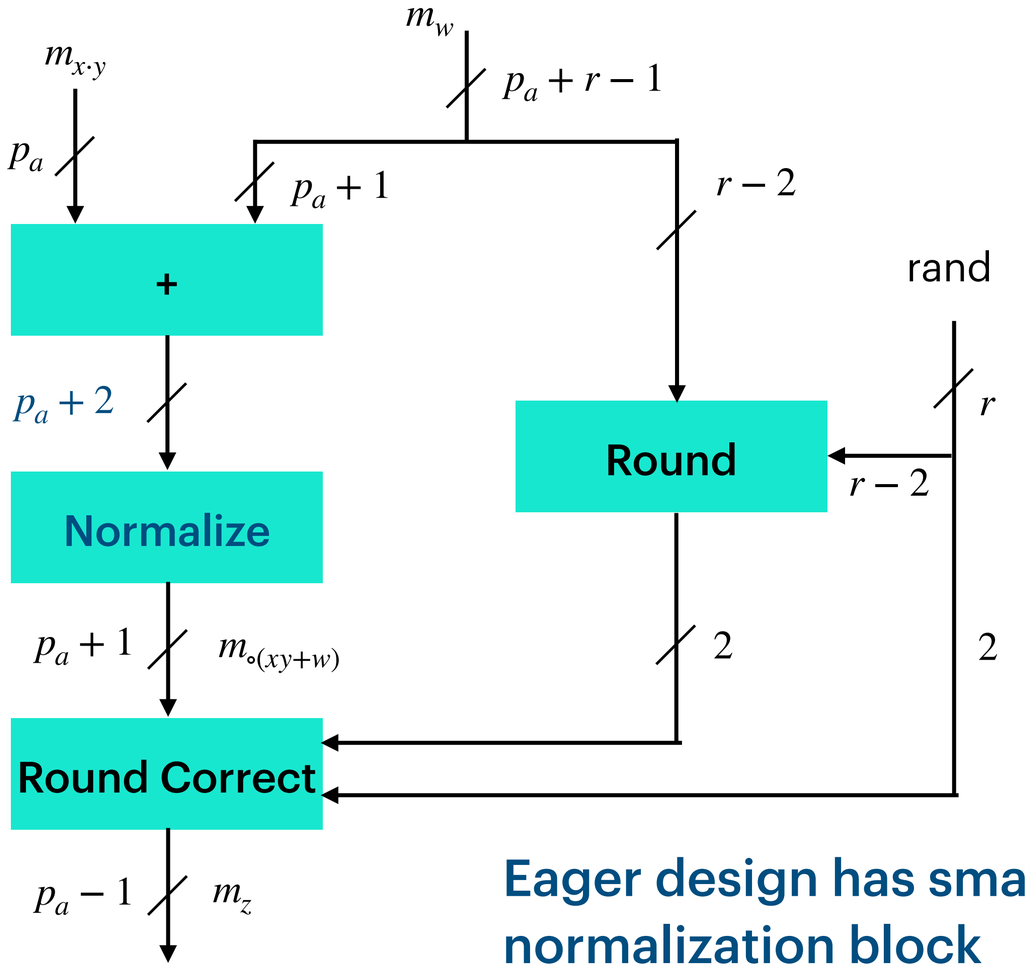
[1] A Stochastic Rounding-Enabled Low-Precision Floating-Point MAC for DNN Training, *S. Ben Ali, S.-I. Filip, O. Sentieys.*, in proceedings of the 2024 IEEE/ACM Design, Automation and Test in Europe (DATE) conference, Mar 2024

Eager MAC Unit Design with SR

Classic SR Design (lazy)



Proposed SR Design (eager)



Eager design has smaller normalization block

[1] A Stochastic Rounding-Enabled Low-Precision Floating-Point MAC for DNN Training, *S. Ben Ali, S.-I. Filip, O. Sentieys.*, in proceedings of the 2024 IEEE/ACM Design, Automation and Test in Europe (DATE) conference, Mar 2024

Hardware Synthesis Results

FP12 (6,5) accumulator with SR

Synthesis results obtained using FDSOI 28nm technology

Achieve up to **18.5%** and **14.2%** savings in area and energy

➡larger savings for larger r (9 vs 18 here)

Can report up to **32.2%** delay reduction