A Cost-Efficient Iterative Truncated Logarithmic Multiplication for Convolutional Neural Networks

HyunJin Kim, Min Soo Kim, Alberto A. Del Barrio, Nader Bagherzadeh

Motivations

Approximate multiplication

- Well applied to inference of simple neural networks.
- But high complex convolutional neural networks (CNNs) require high accurate computation.

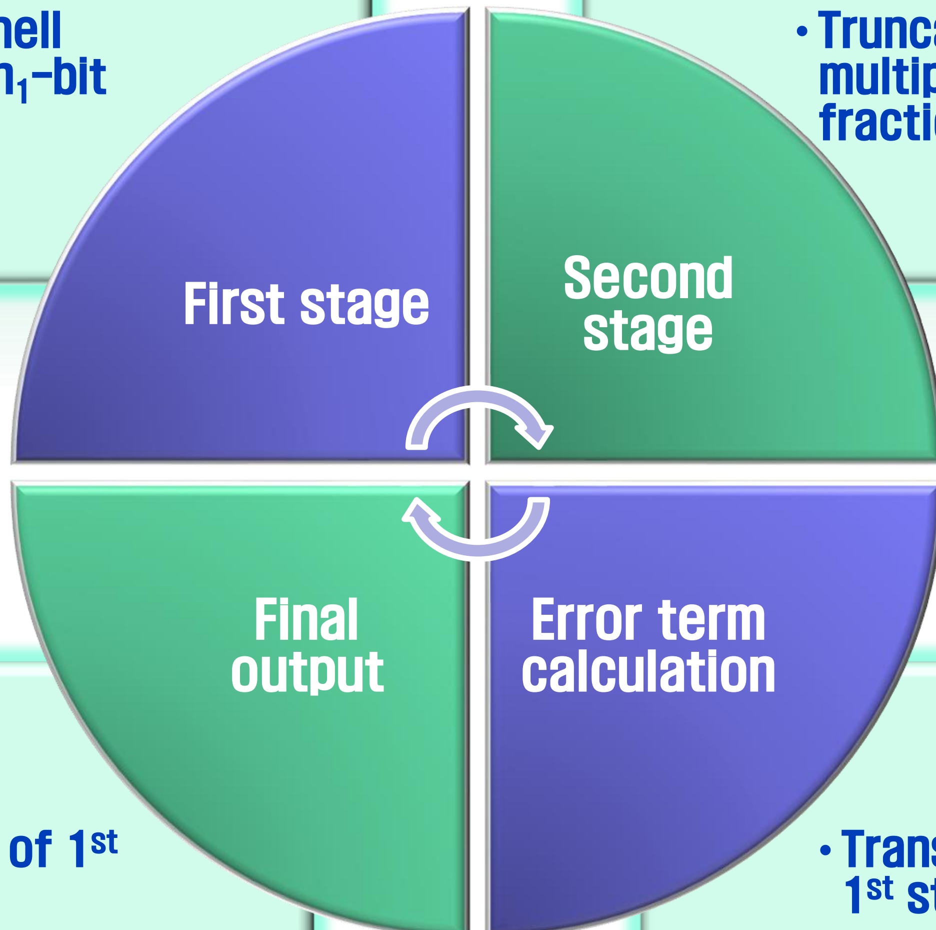
High accurate computation with low cost

- Iterative structure can enhance the accuracy.
- Repeating basic blocks add significant cost.
- Let us reduce cost of basic blocks without degrading performance of CNNs!!

Summary of Proposed Design

- Truncated Mitchell multiplier with n_1-bit fractions

- Truncated Mitchell multiplier with n_2-bit fractions



- Sum of outputs of 1st and 2nd stages

- Transfer error from 1st stage to 2nd stage

Basics of Mitchell Algorithm (Multiplication)

$$A = (1 + x_A) \cdot 2^{k_A},\ B = (1 + x_B) \cdot 2^{k_B}$$

$$log_2(A \cdot B) = k_A + k_B + log_2((1 + x_A) \cdot (1 + x_B))$$

How to approximate it?

$$C = (1 + x_C) \cdot 2^{k_C} \text{ and C = A } \cdot B,$$

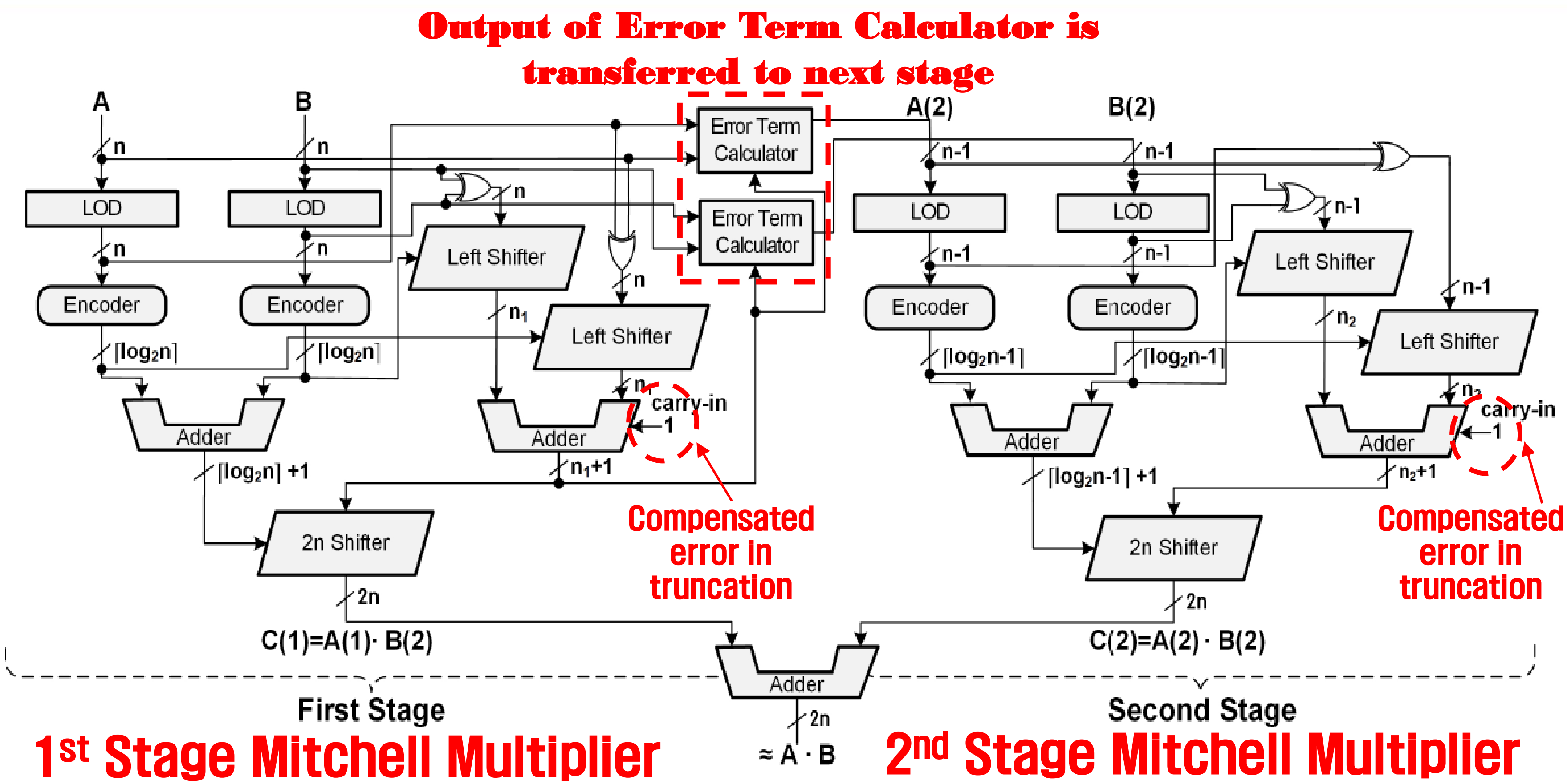
$$\begin{cases} k_C = k_A + k_B + 1,\ x_C = x_A + x_B - 1,\ x_A + x_B \geq 1 \\ k_C = k_A + k_B,\ x_C = x_A + x_B,\ x_A + x_B < 1. \end{cases}$$

$$\therefore rerr = \frac{MUL_{exact} - MUL_{appr}}{MUL_{exact}} = \begin{cases} \frac{(1-x_A)\cdot(1-x_B)}{(1+x_A)\cdot(1+x_B)}, & \text{if } x_A + x_B \geq 1 \\ \frac{x_A \cdot x_B}{(1+x_A)\cdot(1+x_B)}, & \text{if } x_A + x_B < 1. \end{cases}$$

* ***rerr*: relative error**

Error depends on sum of fractions.

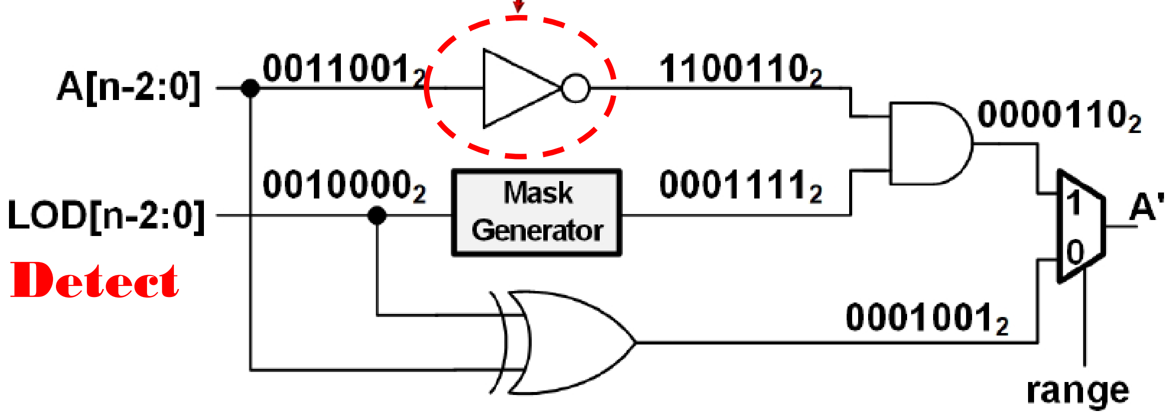
Structure of Proposed Design



Error Term Calculator

$$\begin{cases} A(2) = (1-x_A)\cdot 2^{k_A} - 1, \\ B(2) = (1-x_B)\cdot 2^{k_B} - 1, \\ \quad \text{if } x_{A(1)} + x_{B(1)} + 2^{-n_1} \geq 1 \\ \\ A(2) = x_A \cdot 2^{k_A}, \;\; B(2) = x_B \cdot 2^{k_B}, \\ \quad \text{if } x_{A(1)} + x_{B(1)} + 2^{-n_1} < 1. \end{cases}$$

1's complement



* **LOD: Leading One Detect**

Summary of Error Analysis

n	n_1	$rerr_{max}$	$rerr_{min}$	$rerr_{avg}$	$\|rerr\|_{avg}$
	4	11.1%	-6.25%	-1.09%	1.77%
	5	11.1%	-3.33%	-0.83%	1.11%
8	6	11.1%	-2.50%	-0.58%	0.76%
	7	11.1%	-2.08%	-0.32%	0.57%
	8	11.1%	-1.88%	-0.06%	0.44%
	4	11.1%	-6.25%	0.10%	1.44%
	5	11.1%	-3.33%	0.11%	0.77%
16	6	11.1%	-2.50%	0.11%	0.46%
	7	11.1%	-2.08%	0.12%	0.33%
	8	11.1%	-1.88%	0.12%	0.28%
	4	11.1%	-6.25%	0.11%	1.44%
	5	11.1%	-3.33%	0.12%	0.77%
32	6	11.1%	-2.50%	0.12%	0.46%
	7	11.1%	-2.08%	0.13%	0.33%
	8	11.1%	-1.88%	0.13%	0.28%

11.1% $rerr_{max}$ from error term using 1's complement. (e.g. $A=11_2$, $B=11_2 \rightarrow A(2)=0$, $B(2)=0$)

Comparison of Error and Cost

Better $rerr_{avg}$ compared to other approximate multipliers

Great cost reduction over Booth multiplier when n=16 and n=32

n	design	$rerr_{max}$ (%)	$rerr_{avg}$ (%)	critical path (ns)	area (um^2)	power (uW)
8	Booth[a]	0	0	1.3	613	403
	MM[b]	11.11	3.76	1.3	446	217
	IM[c]	6.25	0.83	1.9	1,133	590
	PROP [d]	11.11	-1.09	2.6	786	370
16	Booth[a]	0	0	2.8	2,507	1,760
	MM[b]	11.11	3.85	2.3	1,168	602
	IM[c]	6.25	0.99	3.7	2,901	1,410
	PROP [e]	11.11	0.11	5.1	1,638	739
32	Booth[a]	0	0	5.4	10,139	6,750
	MM[b]	11.11	3.85	4.2	3,418	1,640
	IM[c]	6.25	0.99	6.5	7,674	3,680
	PROP [e]	11.11	0.12	7.9	3,102	1,370

[a] Radix-4 Booth multiplier [23]
[b] Mitchell multiplier [16]
[c] Two-stage Babic's iterative multiplier [17]
[d] Proposed two-stage multiplier with $\boxed{n_1 = 4, n_2 = 2}$
[e] Proposed two-stage multiplier with $\boxed{n_1 = 6, n_2 = 2}$

Comparison of Accuracy on CNNs

When n_1=6 and n_2=2, there is no significant accuracy drop in CNN models, which outperforms original Mitchell multiplier.

For n_1=8, top-5 accuracy of ResNet-50 reaches up to 90.9%.

Model	Dataset	Multiplier	Top-1 (%)	Top-5 (%)
NiN [11]	CIFAR-10	FLOAT[a]	89.4	-
		FIXED[b]	89.4	-
		MM[c]	88.7	-
		IM[d]	89.4	-
		PROP[e]	89.5	-
AlexNet [12]	ImageNet	FLOAT[a]	57.0	81.3
		FIXED[b]	57.0	81.3
		MM[c]	56.8	80.8
		IM[d]	56.8	81.3
		PROP[e]	56.9	81.3
GoogLeNet [13]	ImageNet	FLOAT[a]	68.3	88.4
		FIXED[b]	68.3	88.4
		MM[d]	67.1	87.5
		IM[d]	68.3	88.2
		PROP[e]	68.3	88.3
ResNet-50 [14]	ImageNet	FLOAT[a]	74.3	90.9
		FIXED[b]	74.2	90.9
		MM[c]	72.4	90.0
		IM[d]	73.9	90.9
		PROP[e]	73.8	90.6

[a] Original Caffe using floating-point multiplications
[b] Fixed-point multiplications
[c] Mitchell multipliers [2]
[d] Two-stage Babic's iterative multipliers [7]
[e] Proposed two-stage multiplier with $n_1 = 6, n_2 = 2$

Conclusion

We proposes the iterative truncated logarithmic multiplication, and error & cost and application of CNNs are analyzed.

Proposed iterative truncated logarithmic multiplication

- Can increase performance with small cost in applications of approximate multiplier

Application of CNNs

- The truncation and error term calculation of our design do not incur substantial impact on inference accuracy on CNN models

A Cost-Efficient Iterative Truncated Logarithmic Multiplication for Convolutional Neural Networks

HyunJin Kim, Min Soo Kim, Alberto A. Del Barrio, Nader Bagherzadeh