HEALM: Hardware-Efficient Approximate Logarithmic Multiplier with Reduced Error

Shuyuan Yu, **Maliha Tasnim and Sheldon Tan**

VSCLAB

Department of Electrical and Computer Engineering

University of California, Riverside

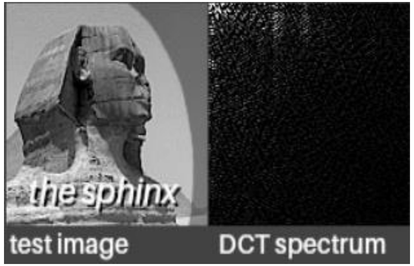
Outline

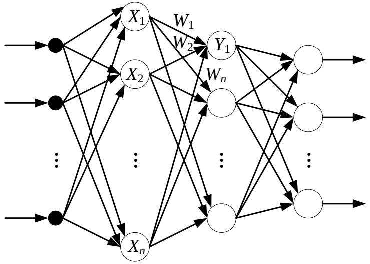
- Background
- Related Works & Motivation
- HEALM Method
- Results & Discussion
    - Error Evaluation
    - Hardware Performance
    - DCT application
- Conclusion

Background

More emerging workloads and application are error tolerant

- Image process and computer vision (DCT, edge detection, contrast stretch, etc.)
- Machine learning (DNN, CNN, Transformer, Language model, etc.)
- Applications interfacing with human being do not need exact values





Approximate computing, which allows the trade-off between area, delay, and power, is more efficient for error tolerant applications.

Background

Approximate Multipliers

- Ad-hoc based
- **Inexact adder based**
- Smaller multiplication with reduced precision
- **Log-based**

Advantages for ALM:

- Scalability (same Rel. error for 8-bit, 16-bit, 32-bit)
- Area/Power/Energy efficient
- Acceptable accuracy: **3.76%** (Mean Rel.) & **11.11%** (Peak Rel.)

Approximate Log-based Muliplier (ALM)

Concept:

$a = 2^{ka} \cdot (1+x)$

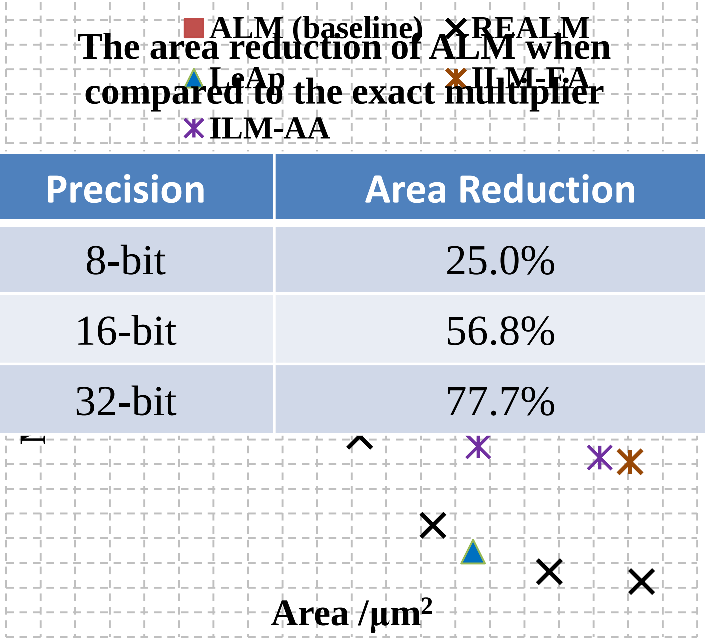
$b = 2^{kb} \cdot (1+y)$

$$C_{Exact} = 2^{k_a+k_b} \cdot (1+x) \cdot (1+y)$$
$$= 2^{k_a+k_b} \cdot (1+x+y+\cancel{xy})$$

$$C_{ALM} = \begin{cases} 2^{k_a+k_b} \cdot (1+x+y), & x+y<1, \\ 2^{k_a+k_b+1} \cdot (x+y), & x+y \geq 1 \end{cases}$$

Related Work & Motivation

State of art works with 8-bit precision compared with ALM baseline (Area vs. Mean Rel.)



The area reduction of ALM when compared to the exact multiplier

Precision	Area Reduction
8-bit	25.0%
16-bit	56.8%
32-bit	77.7%

State of art works

- LeAp [15]
- ILM-EA, ILM-AA [13]
- REALM [10]

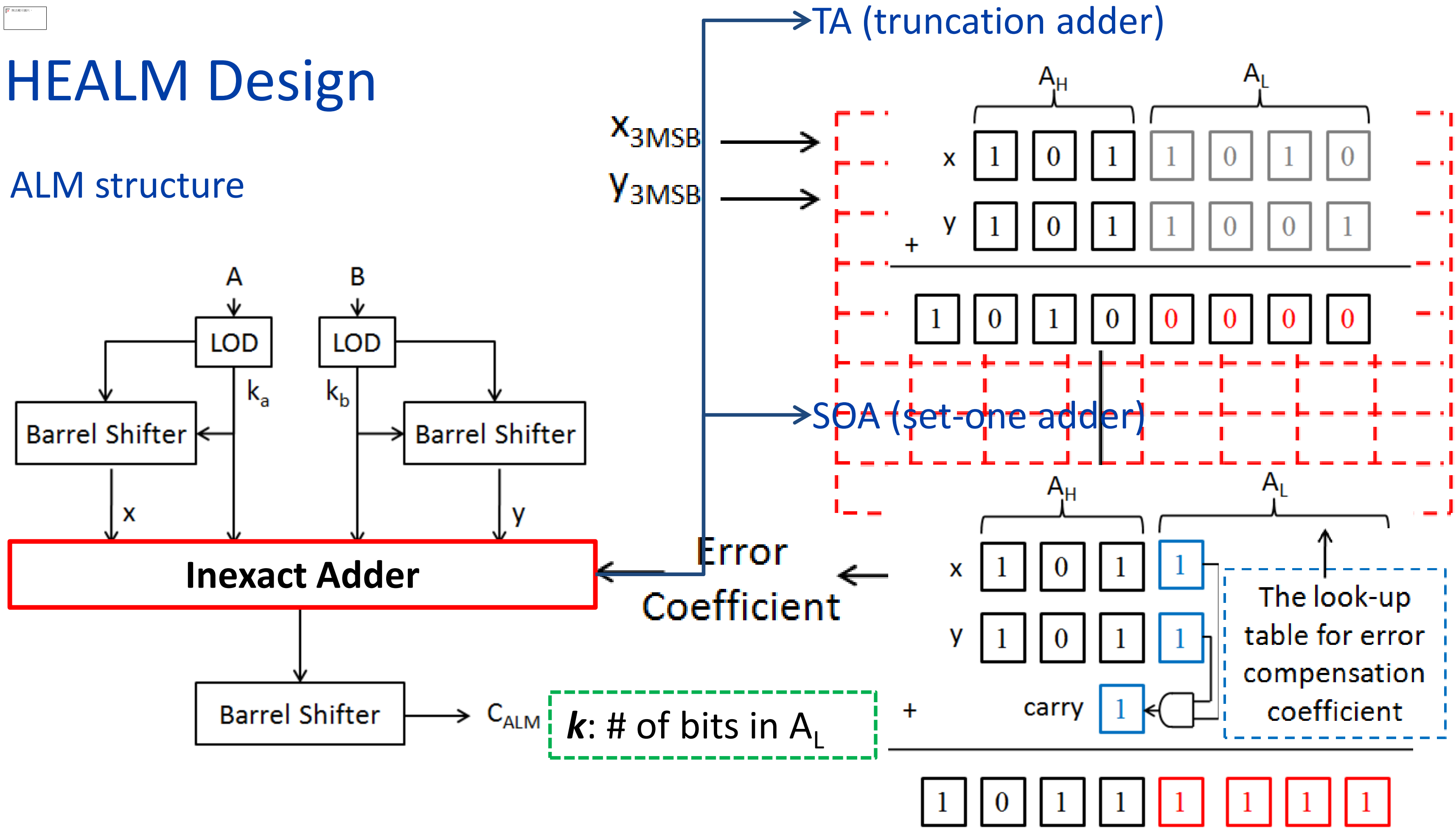
Problems

- Improve accuracy with resource overhead
- Save resource with accuracy loss
- Lower precision, less resource saving

Motivation: propose 8-bit logarithmic-based multiplier with resource saving and accuracy improvement at the same time

HEALM Design

ALM structure

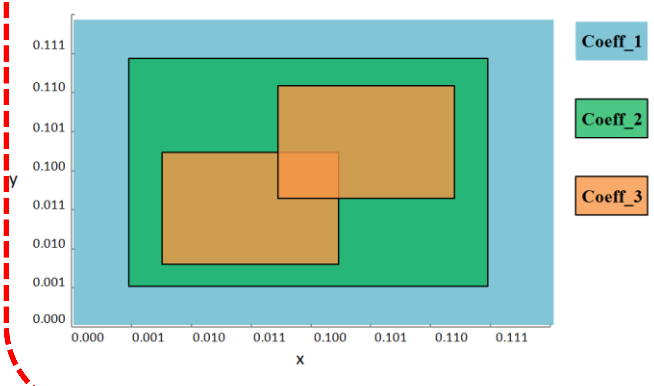


k: # of bits in A_L

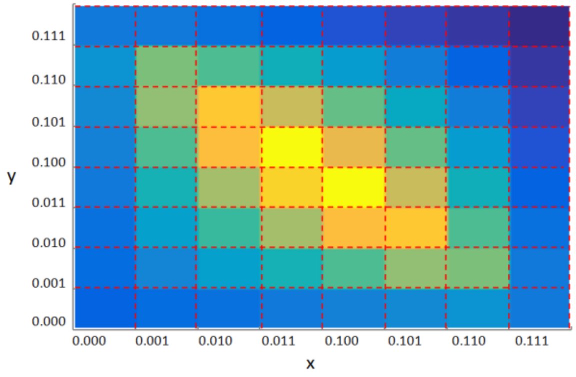
Error Compensation

State of art improved ALMs

LeAp [15]



REALM [10]



ONE Pattern

NOT suitable when we introduce inexact summation unit with 8-bit precision.

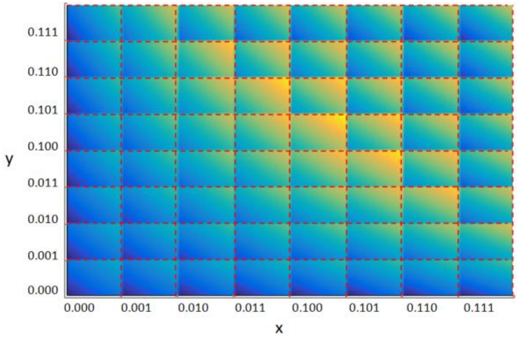
HEALM Method

Generate error coefficient table based on different value of ***k***

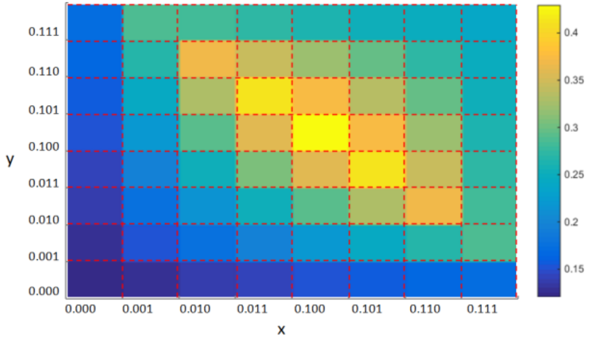
Error profile with inexact adder & error compensation

*****

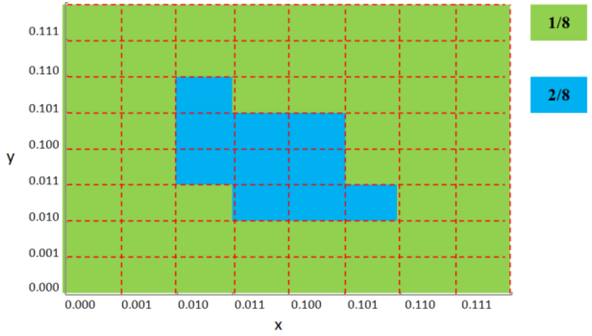
# of bits of coeff
$\leq$
# of bits of A_H



Error profile



Error profile (avg. in block)



Error coefficient

HEALM-TA

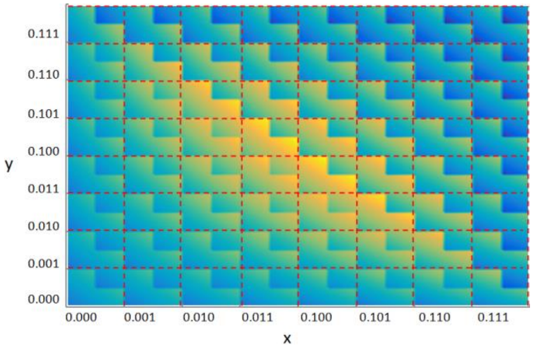
UC RIVERSIDE

HEALM Method

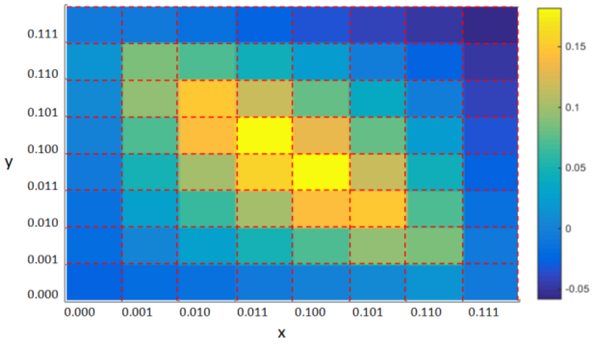
Generate error coefficient table based on different value of ***k***

Error profile with inexact adder & error compensation

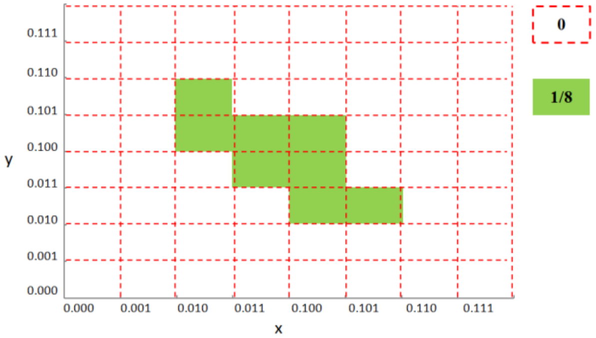
Ex. k =4



Error profile



Error profile (avg. in block)

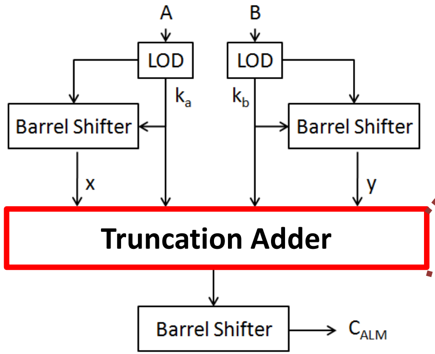


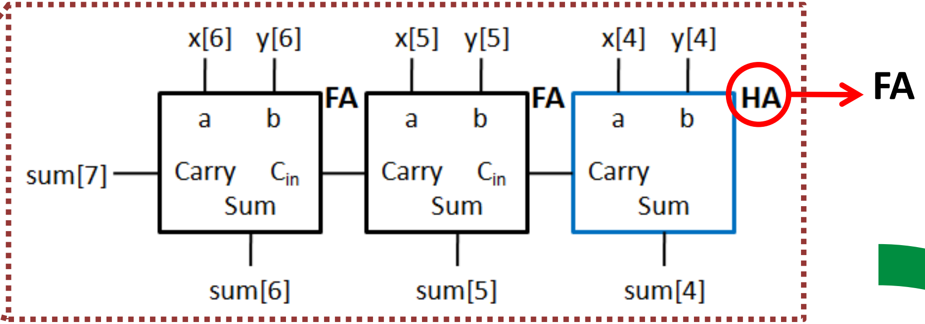
Error coefficient

HEALM-SOA

Single Coefficient Mode (HEALM-TA-S)

Ex. With largest value of *k* (8-bit, k=4)



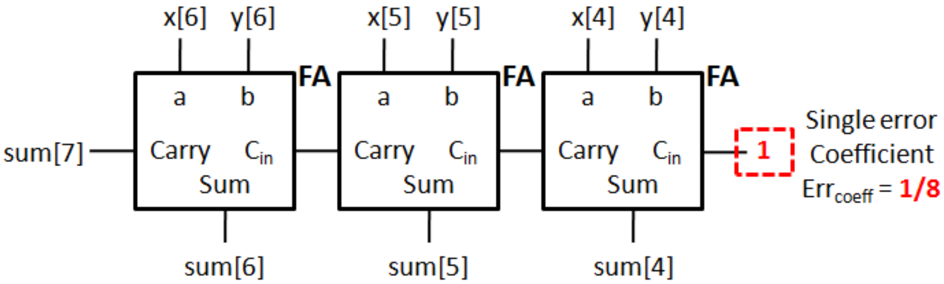


For ALL the inputs, introduce just ONE single error coefficient

$x[6:0] + y[6:0]$

$k = 4$, **truncate 4 bits**

$x[6:4] + y[6:4]$



UC RIVERSIDE

Results & Discussion

Experimental setup

- Baseline: ALM

- EDK 32nm standard cell library (SNPS DC)

- 400MHz working frequency

- 1 million random input pairs (Matlab)

Error Metrics & Hardware Performance

Logarithmic MUL	*k*	Mean /%	Peak /%	Area /μm²	Power /μW
ALM (baseline)	**0**	**3.76**	**11.11**	**820.63**	**72.83**
HELAM-TA	1	1.12	4.86	1216.84	114.50
	2	1.40	8.25	938.05	92.17
	3	**2.17**	**9.75**	**743.63**	**74.14**
	4	3.66	13.77	595.46	51.41
HEALM-SOA	1	1.13	4.71	1175.41	113.96
	2	1.38	5.90	966.76	90.32
	3	**1.78**	**7.65**	**808.94**	**75.05**
	4	3.12	12.17	664.33	59.55

Results & Discussion

Experimental setup

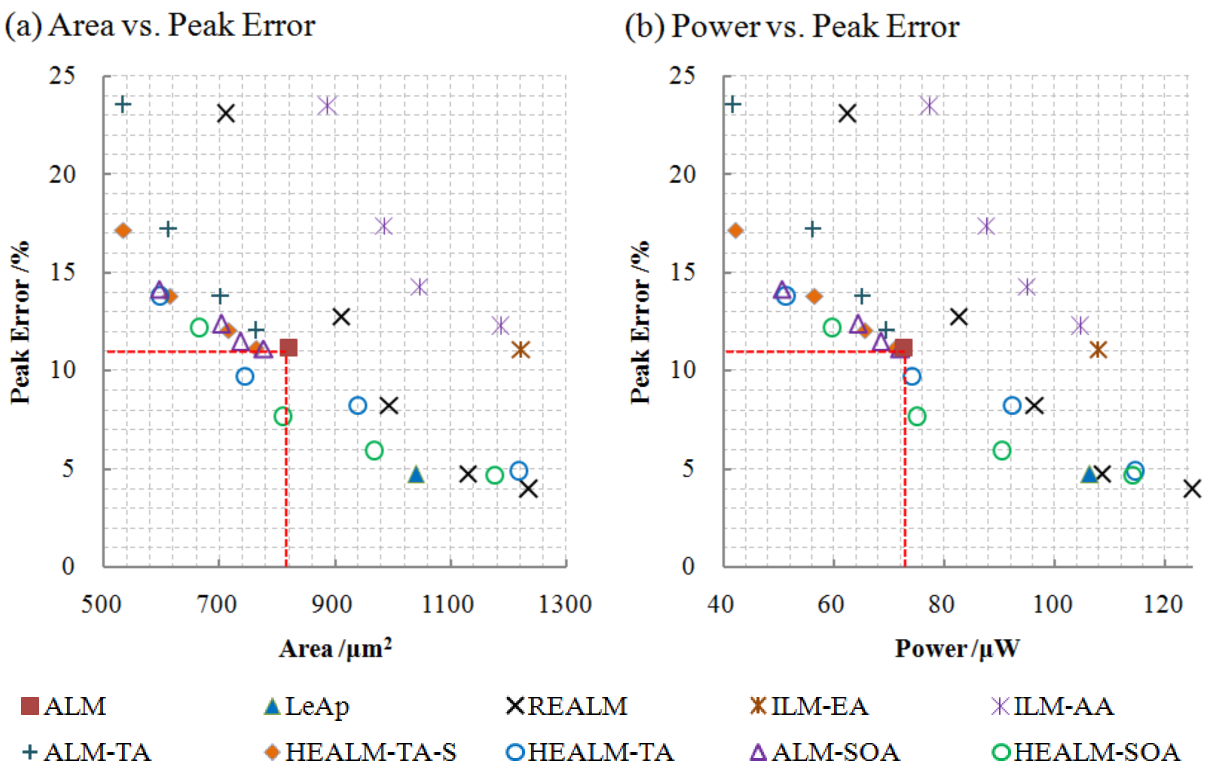
- Baseline: ALM
- EDK 32nm standard cell library (SNPS DC)
- 400MHz working frequency
- 1 million random input pairs (Matlab)

Error Metrics & Hardware Performance

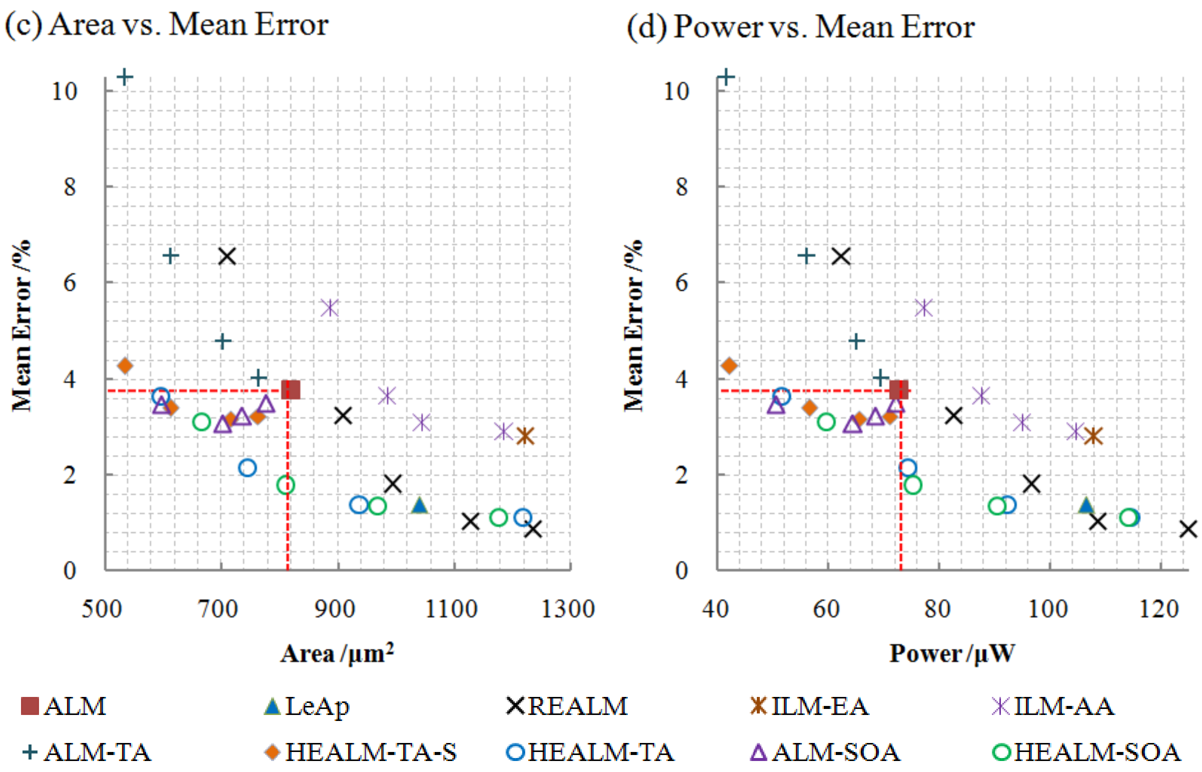
Logarithmic MUL	*k*	Mean /%	Peak /%	Area /µm²	Power /µW
ALM (baseline)	**0**	**3.76**	**11.11**	**820.63**	**72.83**
ALM-TA	1	4.02	12.03	763.45	69.41
	2	4.79	13.83	702.71	65.07
	3	6.58	17.25	612.74	56.20
	4	**10.29**	**23.53**	**533.19**	**41.73**
HEALM-TA-S	1	3.21	11.11	763.70	71.19
	2	3.17	12.02	716.69	65.68
	3	3.39	13.79	614.01	56.62
	4	**4.26**	**17.12**	**534.72**	**42.32**

6.0% **6.4%** **Almost NO overheads**

Comparison with State of Art Works



Comparison with State of Art Works



Complementary Results for **16-bit** Cases

Experimental setup

- Baseline: ALM
- EDK 32nm standard cell library (SNPS DC)
- 200MHz working frequency
- 1 million random input pairs (Matlab)

Error Metrics & Hardware Performance

Logarithmic MUL	*k*	Mean /%	Peak /%	Area /μm²	Power /μW
ALM (baseline)	**0**	**3.76**	**11.11**	**1825.52**	**110.91**
REALM	0	0.75	3.70	2383.36	164.50
	9	**1.06**	**5.27**	**1572.90**	**94.07**
LeAp	0	0.98	4.76	1990.71	128.20
ALM-TA	9	4.88	12.93	1263.86	66.26
HEALM-TA-S	**9**	**4.87**	**12.02**	**1267.16**	**66.45**
HEALM-TA	**9**	**1.64**	**5.83**	**1511.39**	**87.62**
ALM-SOA	9	3.07	12.03	1383.56	74.10
HEALM-SOA	**9**	**1.38**	**5.15**	**1577.47**	**91.89**

Application: Discrete Cosine Transform

Image quality: PSNR (dB)

MUL	Lena	Boat	Barbara	House	Pepper	Avg.
ALM (baseline)	**19.1**	**18.7**	**19.3**	**18.4**	**18.7**	**18.8**
***k* = 4**						
ILM-AA	20.2	18.9	20.1	18.5	19.5	19.5
ALM-TA	14.1	13.7	14.1	13.4	13.7	13.8
ALM-SOA	18.9	18.9	19.6	16.7	18.9	18.6
REALM	19.8	20.0	19.8	17.8	19.5	19.4
HEALM-TA	**29.1**	**28.7**	**29.1**	**25.9**	**28.5**	**28.3**
HEALM-TA-S	**23.1**	**22.0**	**22.6**	**24.3**	**22.3**	**22.9**
HEALM-SOA	**22.2**	**22.1**	**22.6**	**19.4**	**22.4**	**21.7**

Up to 17.2 dB

At least 2.9 dB

Conclusion

- Propose **HEALM** approach: Hardware-Efficient Approximate Logarithmic Multiplier, with reduced error.

- Do improvements upon the ALM baseline in **error metrics** and **hardware performance** aspects **at the same time**.

- Outperform the state of art design with up to **2.9%, 9.3%, 16.1%, 17.6%** improvement in mean error, peak error, area, power consumption, repectively for 8-bit precision.

- The single coefficient mode **HEALM-TA-S** could improve up to **6.0%** and **6.4%** in mean and peak error respectively when compared to ALM with simple TA, with almost **NO** resource overheads.

- For DCT workloads, with k=1, **HEALM design** could achieve up to **17.2dB** improvement upon ALM baseline; with k=4, **HEALM design** could achieve at least **2.9dB** improvement in image quality.

Questions???