DATE 21
DESIGN, AUTOMATION & TEST IN EUROPE
01 – 05 February 2021 · virtual conference
The European Event for Electronic System Design & Test

Improving the energy efficiency of STT-MRAM based approximate cache

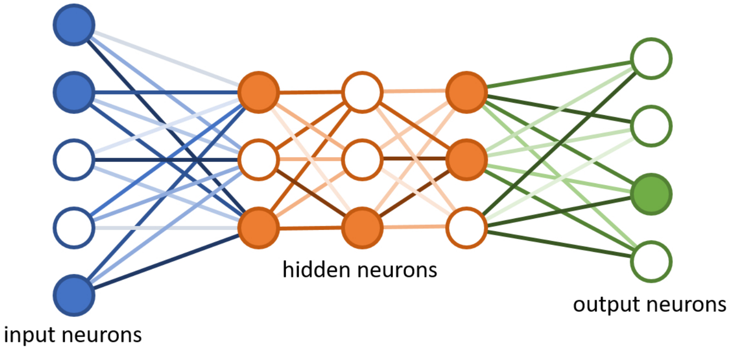
Wei Zhao[1], Wei Tong[1], Dan Feng[1], Jingning Liu[1], Zhangyu Chen[1], Jie Xu[1], Bing Wu[1], Chengning Wang[1], Bo Liu[2]

[1]*Huazhong University of Science and Technology,*
[2]*Hikstor Technology Co., LTD*

Content

- **Background**
- **Motivation**
- **Design**
- **Evaluation**
- **Conclusion**

Background



Neural network

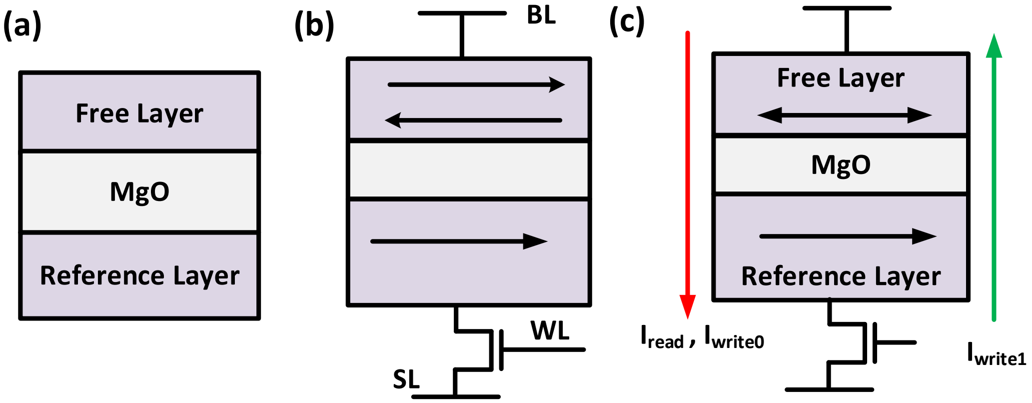
Image processing

① **Current machine learning and image processing , etc. applications *consumes much energy*.**

② **SRAM cache consumes large energy due to *high leakage power*.**

③ **STT-MRAM (Spin Transfer Torque Magnetic RAM) cache has the advantage of *low leakage power*.**

Background



STT-MRAM cell is composed of one MTJ and one transistor (i.e. 1T1M structure).

STT-MRAM cache needs *high write energy* to switch the cell value.

Background

Original

3% error

① Machine learning and image processing applications are ***error-tolerant*****.**

② **Users are *insensitive* to errors in the output data.**

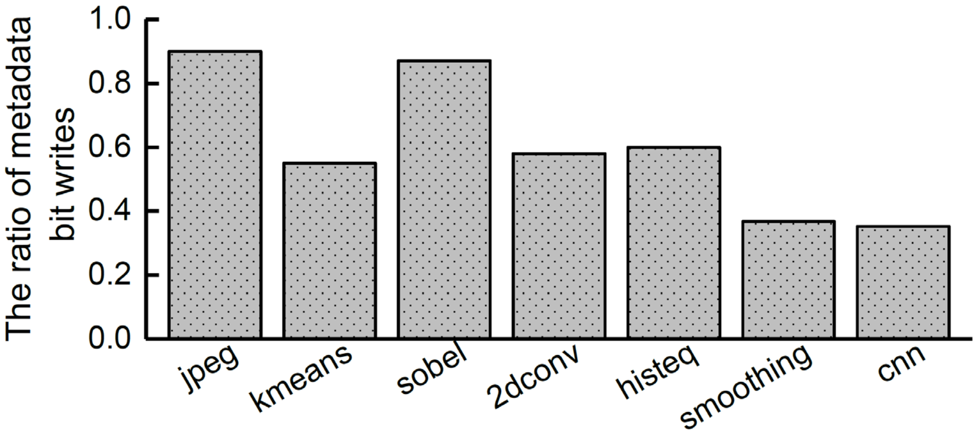
These features show the possibility of reading/writing approximate data to reduce energy.

Motivation

STT-MRAM cache has much lower leakage power, but the *dynamic write energy is high*. → Reduce it!

Some new observations:

① Many bit writes to data are from image raw data.

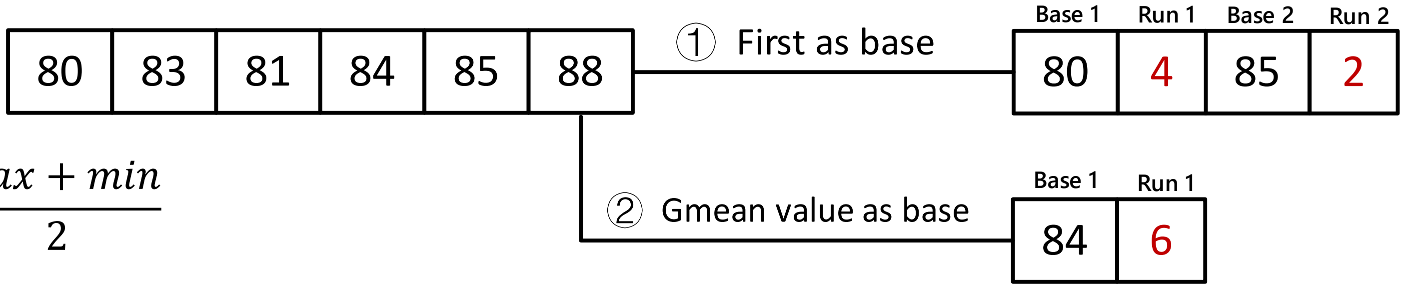


Motivation

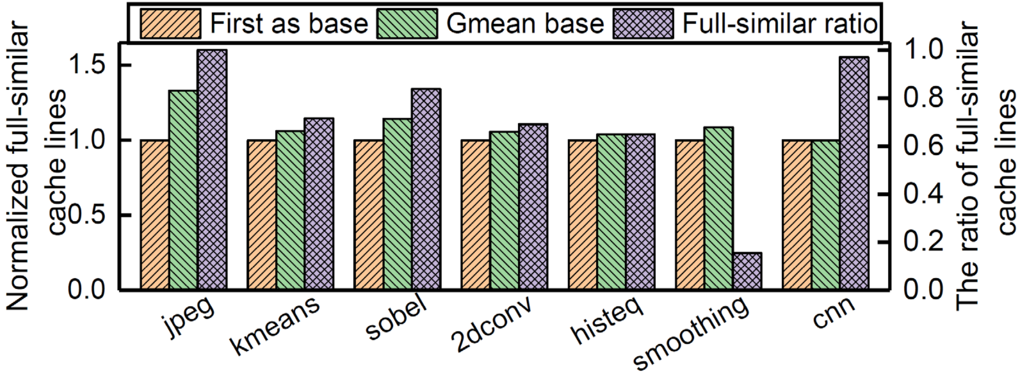
The image raw data shows data similarity/redundancy.

$error = 4$

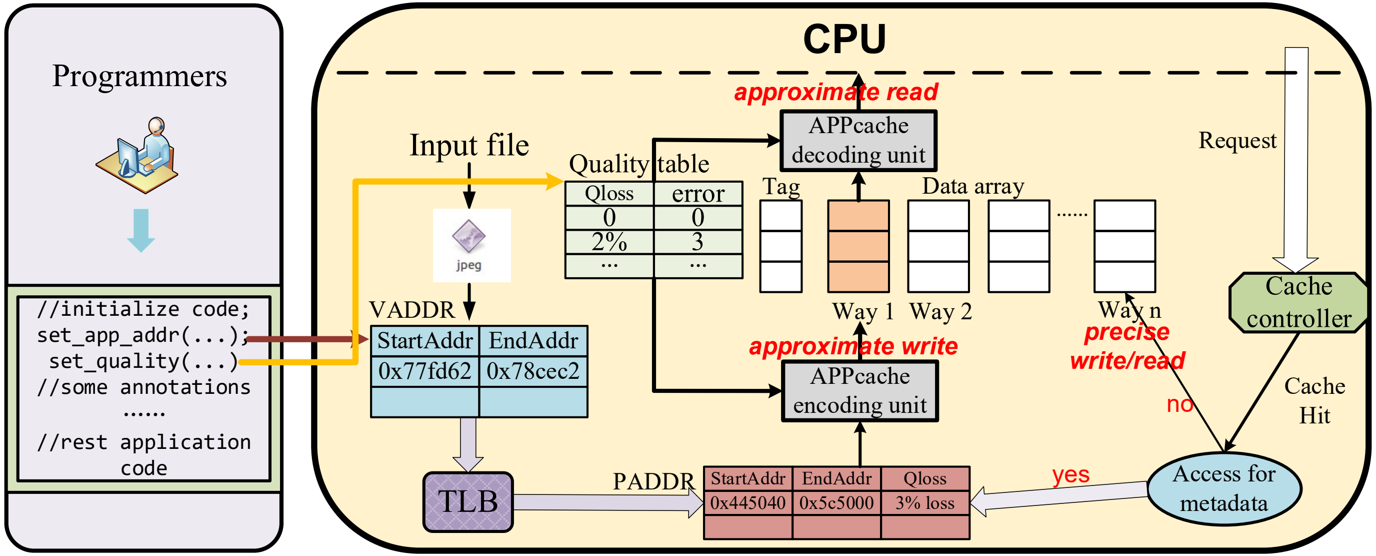
$$Gmean = \frac{max + min}{2}$$



② Gmean base method can reduce more redundant elements.



Design-Overview



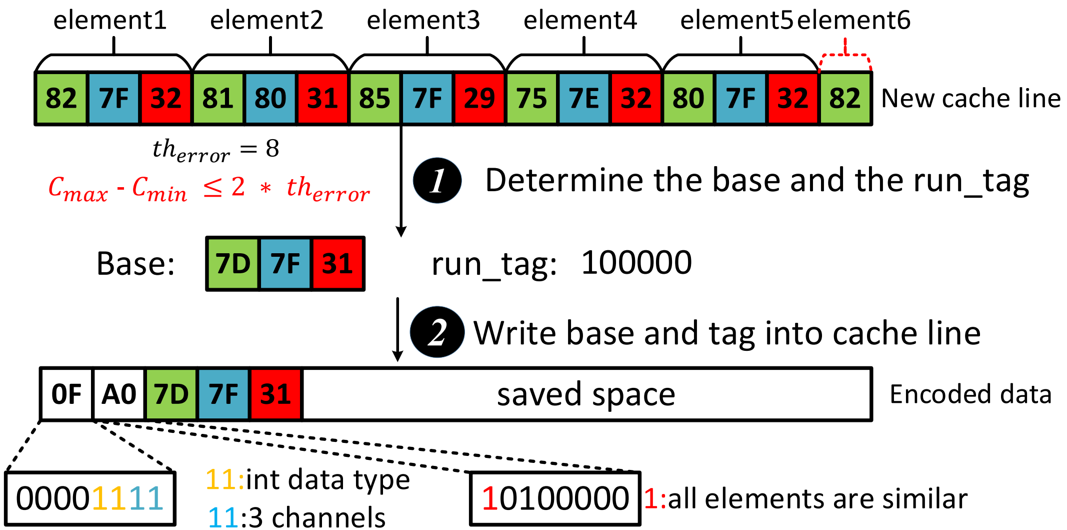
APPcache encoding logic → Approximate write.

APPcache decoding logic → Approximate read.

Software interface → Set output quality.

Design-Similarity based encoding

Encoding：

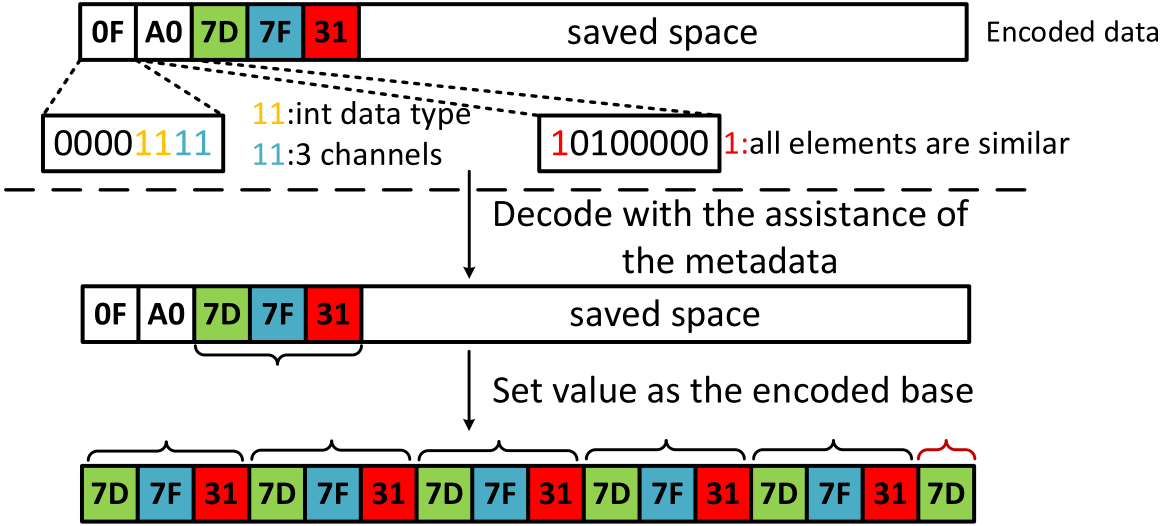


run_tag： '1' means this location starts a new base. '0' means this element is similar to the base.

Encoded data records the metadata, such as data type and channel count.

Design-Similarity based encoding

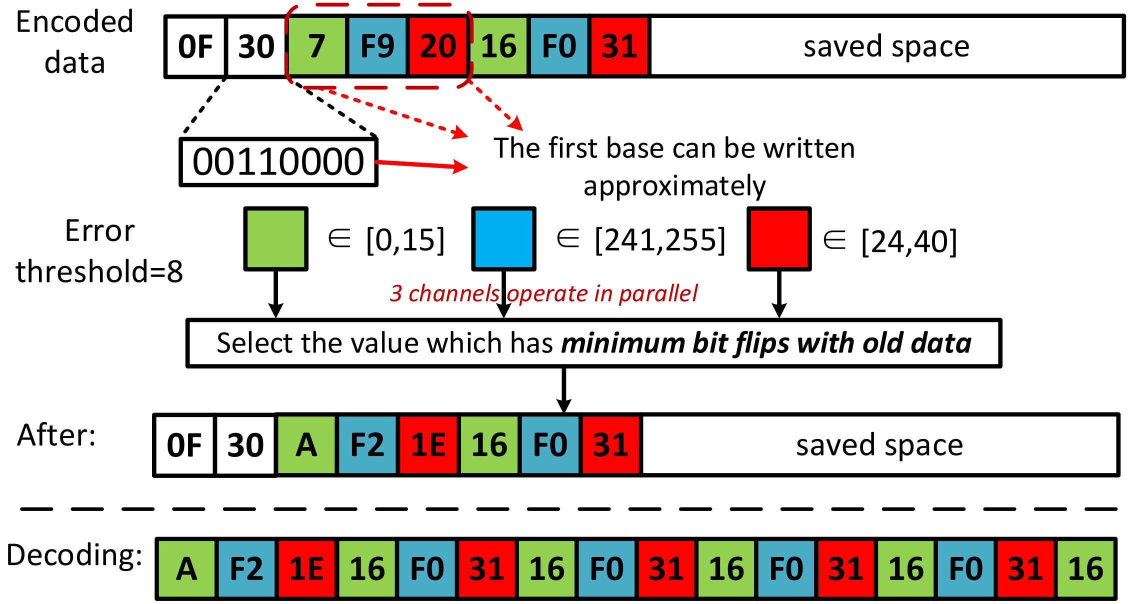
Decoding：



Decoder can get the base and similar elements from metadata.

Each similar element is decoded to its corresponding base.

Design-Selecting the Energy-efficient Base



For bases that do not have similar elements can execute *approximate write/read*, while other bases must execute *precise write/read* .

Design-Software Interface

Software | ***Hardware***

Programmers

```
set_quality(start,end,Q_loss);
```

Quality table

Qloss	error
0	0
2%	3
…	…

`start` **is the start address,** `end` **is the end of approximate region,** `Q_loss` **is the specified quality loss.**

Evaluation-Experiment Setup

We use gem5 to implement our scheme.

TABLE I: System configurations.

Cores	4-Core, 2.0GHz, out-of-order
L1 I/D cache	private, 64KB per core, 2-way; LRU, 2-cycle latency
L2 unified cache	8MB, 64B cache line; 8-way, LRU, 15-cycle latency
Main Memory	4GB, DDR-1600

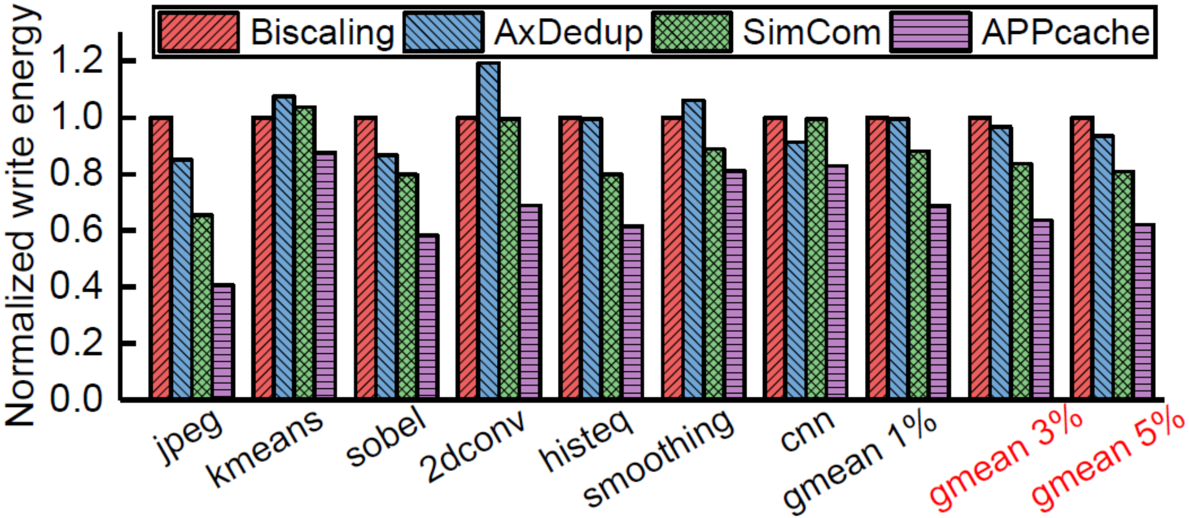
TABLE II: Application benchmarks.

Application	Algorithm	Dataset	Evaluation Metric
Digit Recognition	cnn	MNIST	Classification Accuracy
Image Compression	jpeg	Axbench	Root Mean Square Error
Edge Detection	sobel		
Image Segmentation	kmeans		
Image Processing	2dconv	PERFECT	
Image Enhancement	histeq		
Image Smoothing	smoothing	MiBench	

We make comparisons with following schemes:

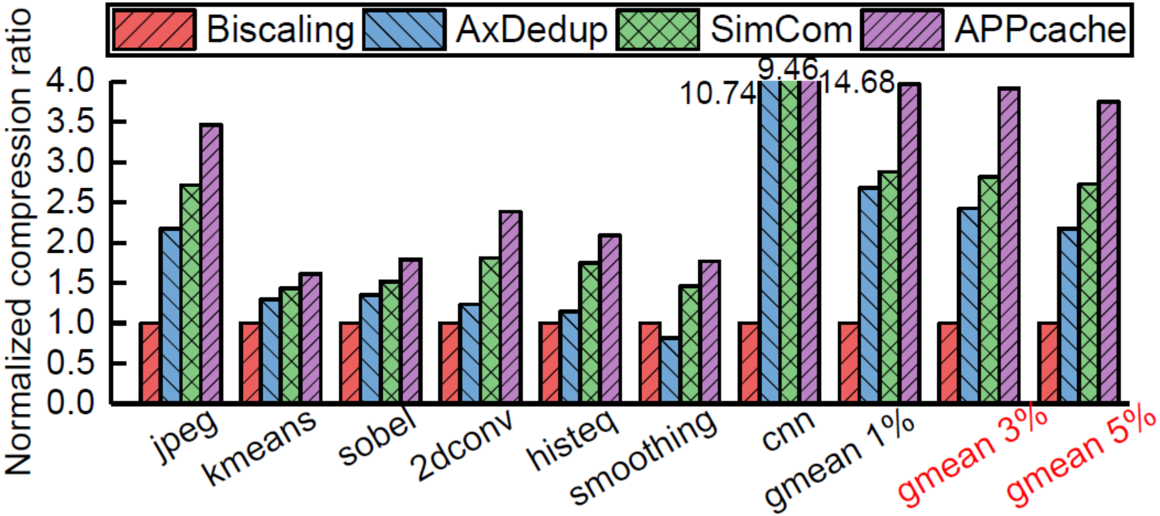
- Biscaling (ISLPED 17)
- AxDedup (ICCAD 18)
- SimCom (DAC20)
- APPcache (Our scheme)

Evaluation-Write Energy



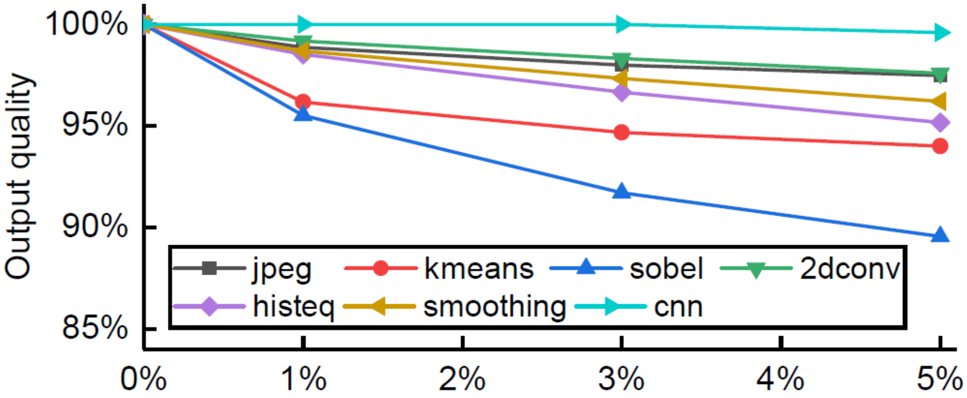
APPcache can reduce write energy by 31.3%/30.8%/21.9%(36.34%/34.1%/24.03%)(38.05%/33.67%/23.42%)compared with Biscaling/AxDedup/SimCom with 1%(3%)(5%)error rate.

Evaluation-Raw Data Compression Ratio



APPcache can improve image raw data compression ratio by 296.9%/48.3%/38.0% (291.48%/61.38%/39.06%) (275.0%/72.16%/37.55%) compared with Biscaling/AxDedup/SimCom with 1% (3%) (5%)error rate.

Evaluation-Output Quality



The quality losses of all benchmarks are within 5% with 1% error rate.

The output images of ***jpeg***:

Conclusion

In this work, we propose APPcache to reduce the write energy of STT-MRAM based approximate cache.

- **Some new observations.**
- **Several techniques to reduce the write energy.**
- **Software support.**
- **Good effectiveness.**

Q&A