

# Approximate Storage in Solid-State Memories

Adrian Sampson

Jacob Nelson

Karin Strauss

Luis Ceze

University of Washington

Microsoft Research & UW  
University of Washington

MICRO 2013

Microsoft  
Research



sa|||pa

Compiler

Runtime

CPU



Compiler

Runtime

Vector  
Processor

CPU

GPU



Network



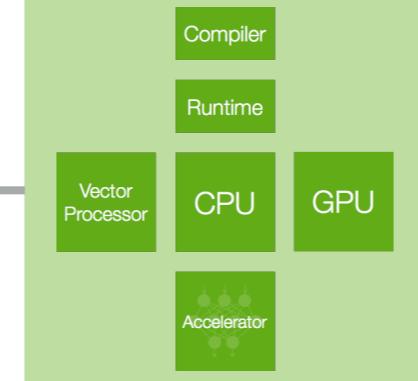
Display



I/O

**Compute**

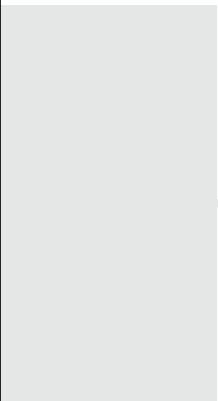
**Storage**



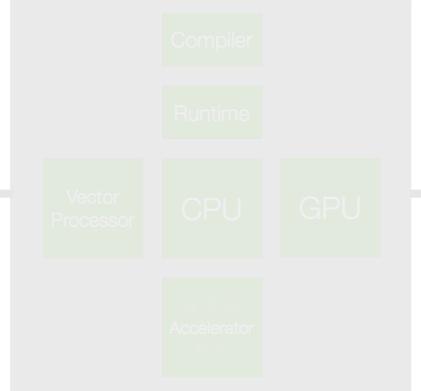
Disk



Memory



Compute



Storage

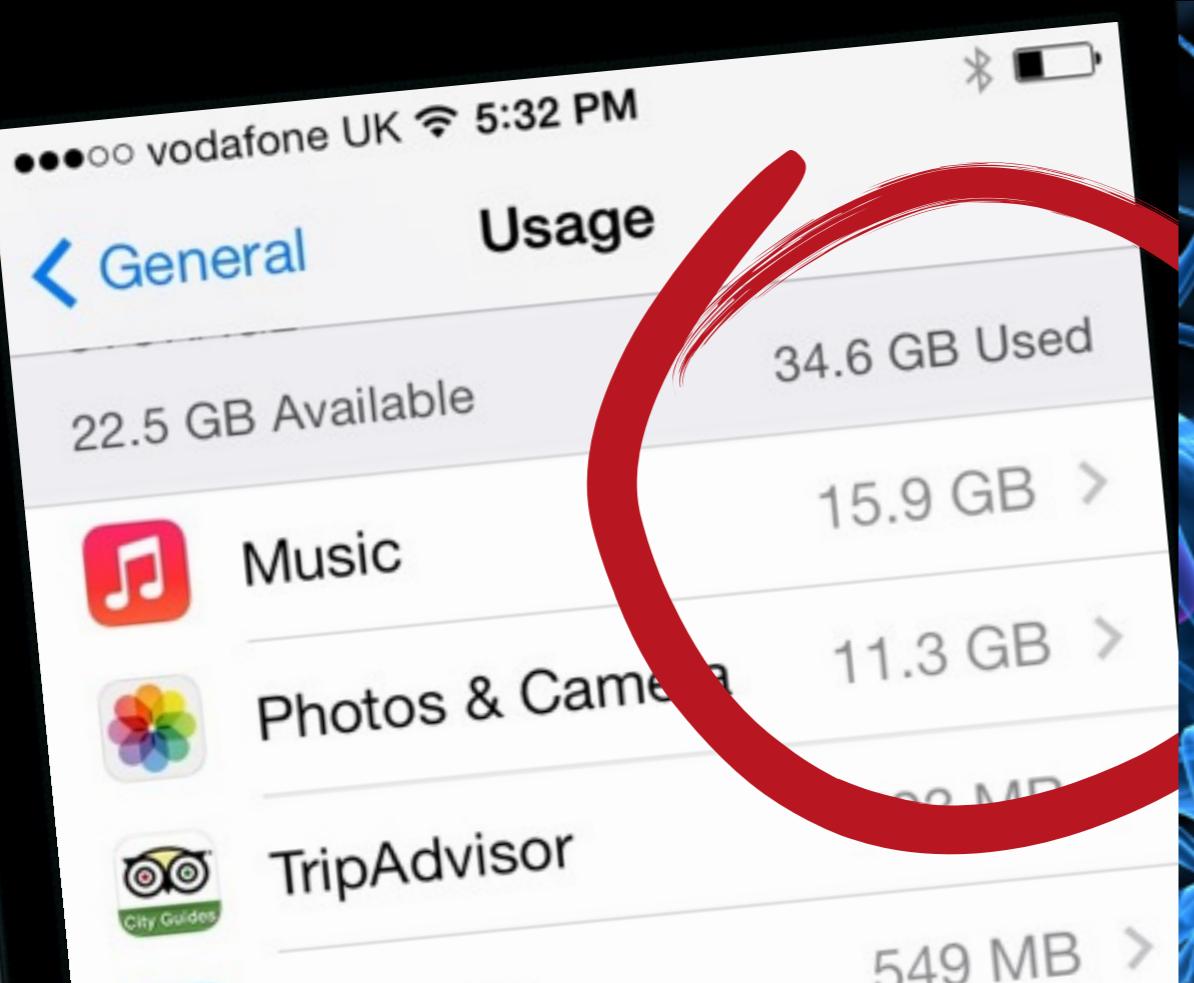


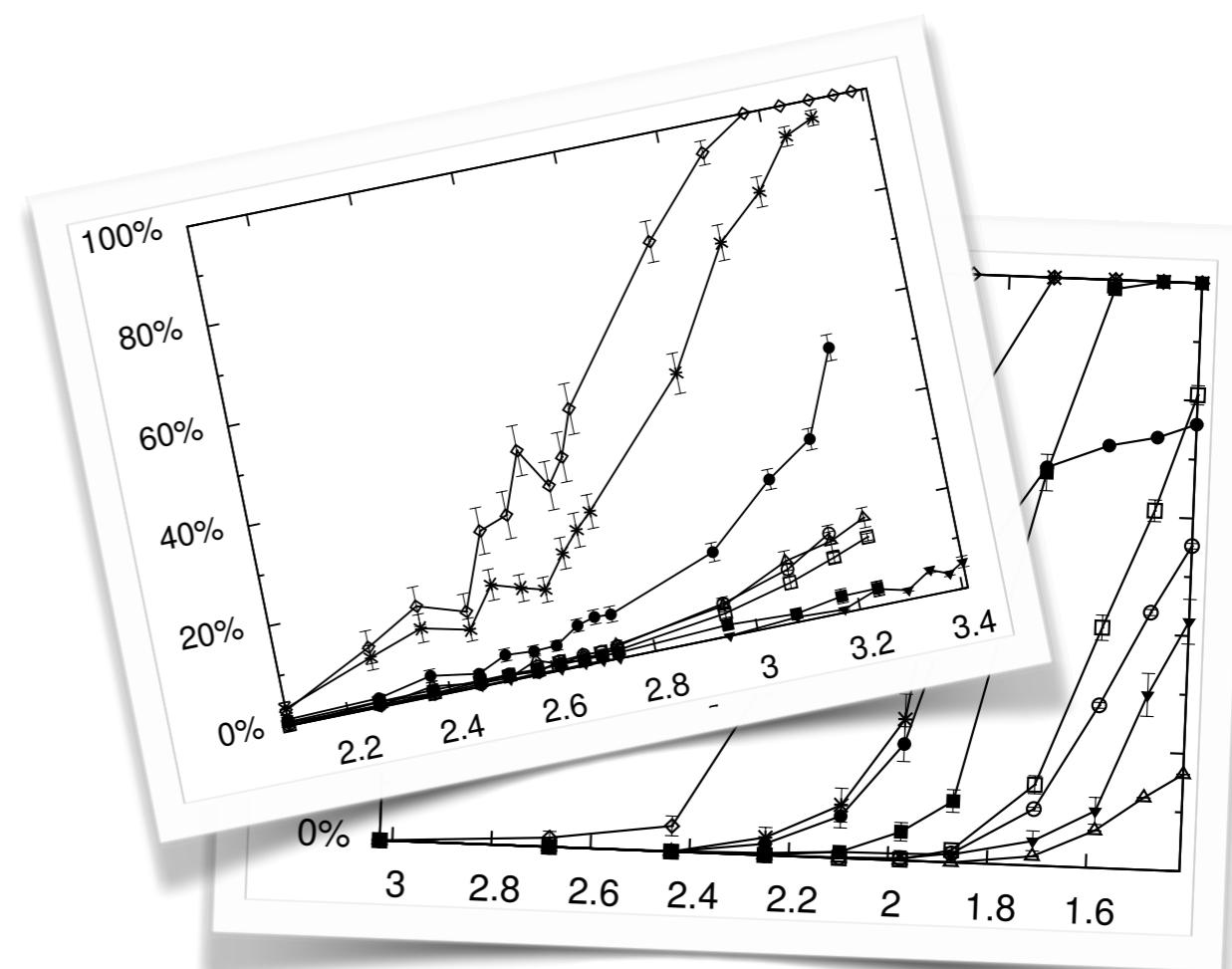
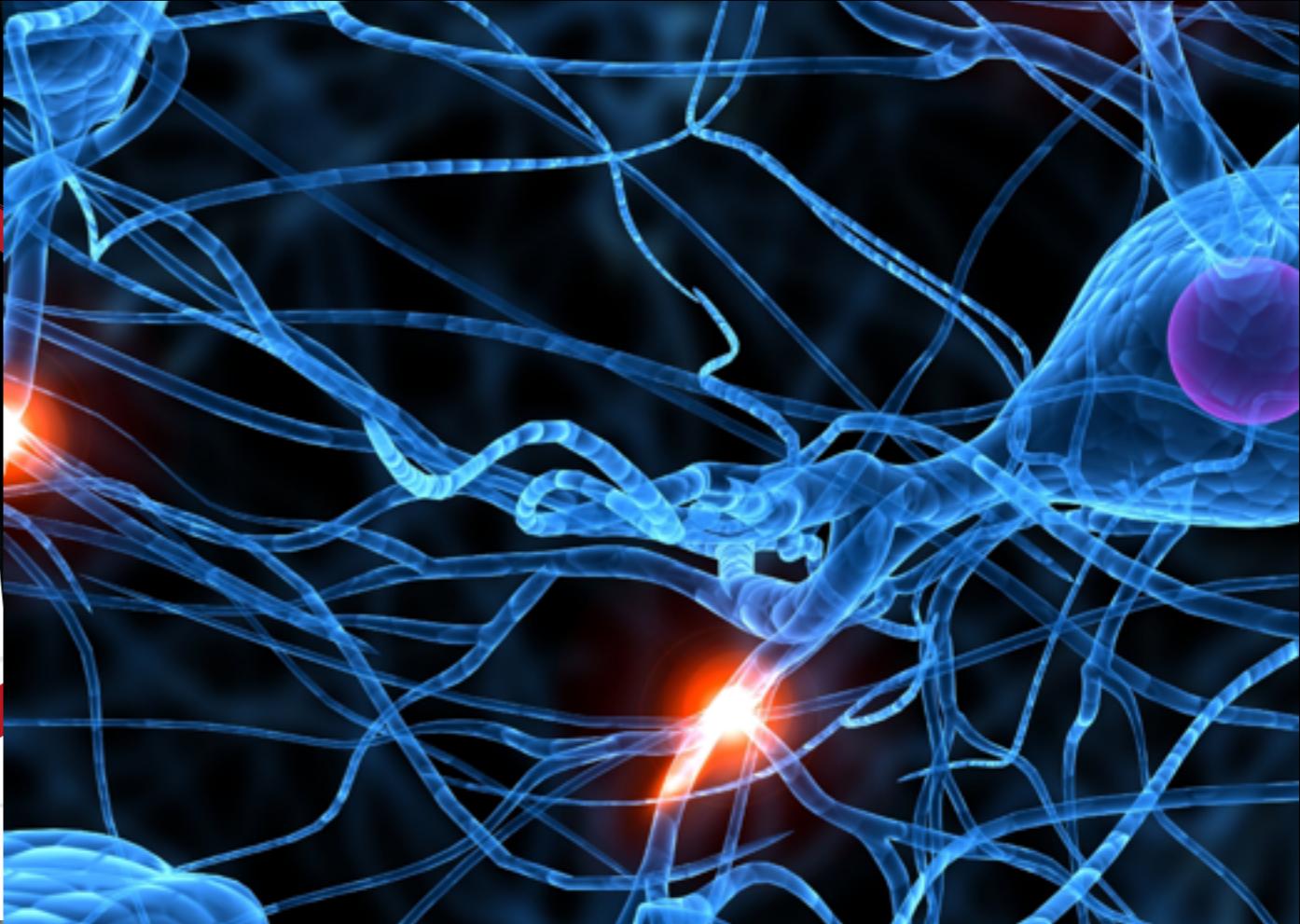
Disk

Memory

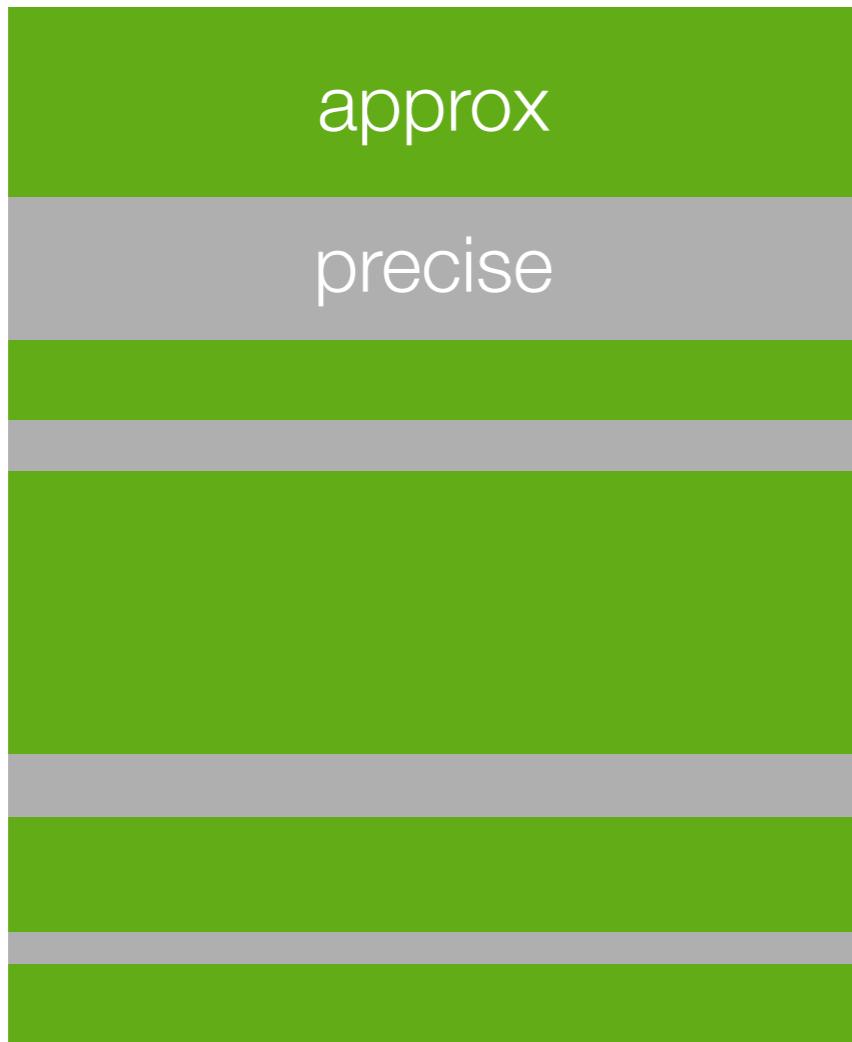




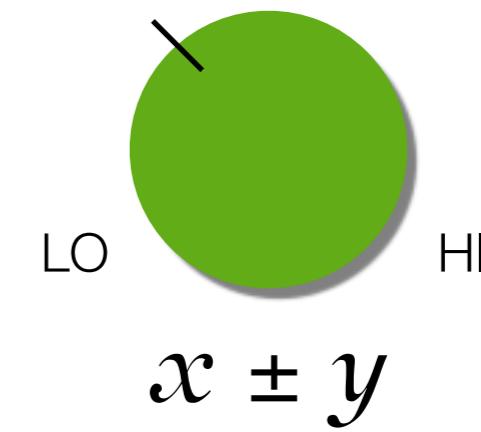




# Themes in approximate computing

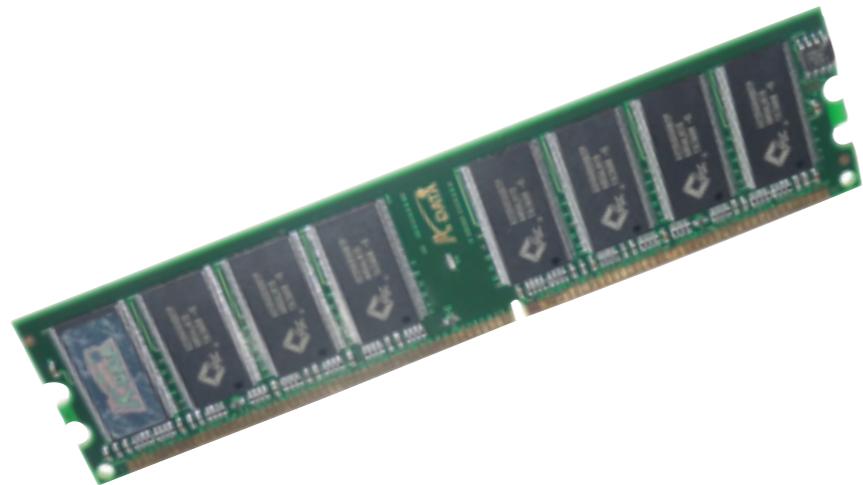


**Interleaving:**  
Programs are both  
approximate & precise



**Error mitigation:**  
Exploit the hardware  
to minimize error

# Phase-change memory (PCM) :)

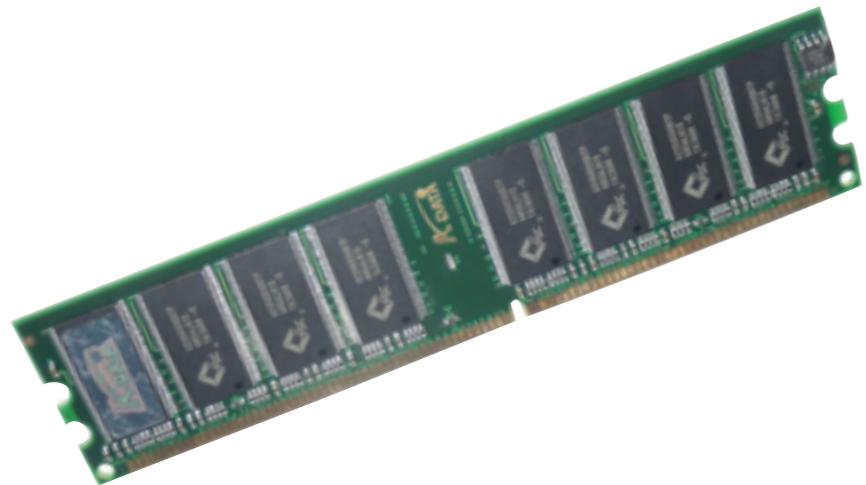


Surpass DRAM's scaling limits

“Almost” as fast as DRAM

Non-volatile  
Faster than flash

# Phase-change memory (PCM) :(



Write speed  
& energy

Cells wear out  
over time

# Phase-change memory (PCM) :(

**Multi-level cells** are denser  
but need more time and energy.

Cells **wear out** over time  
and can no longer be used.

# Phase-change memory (PCM) :(

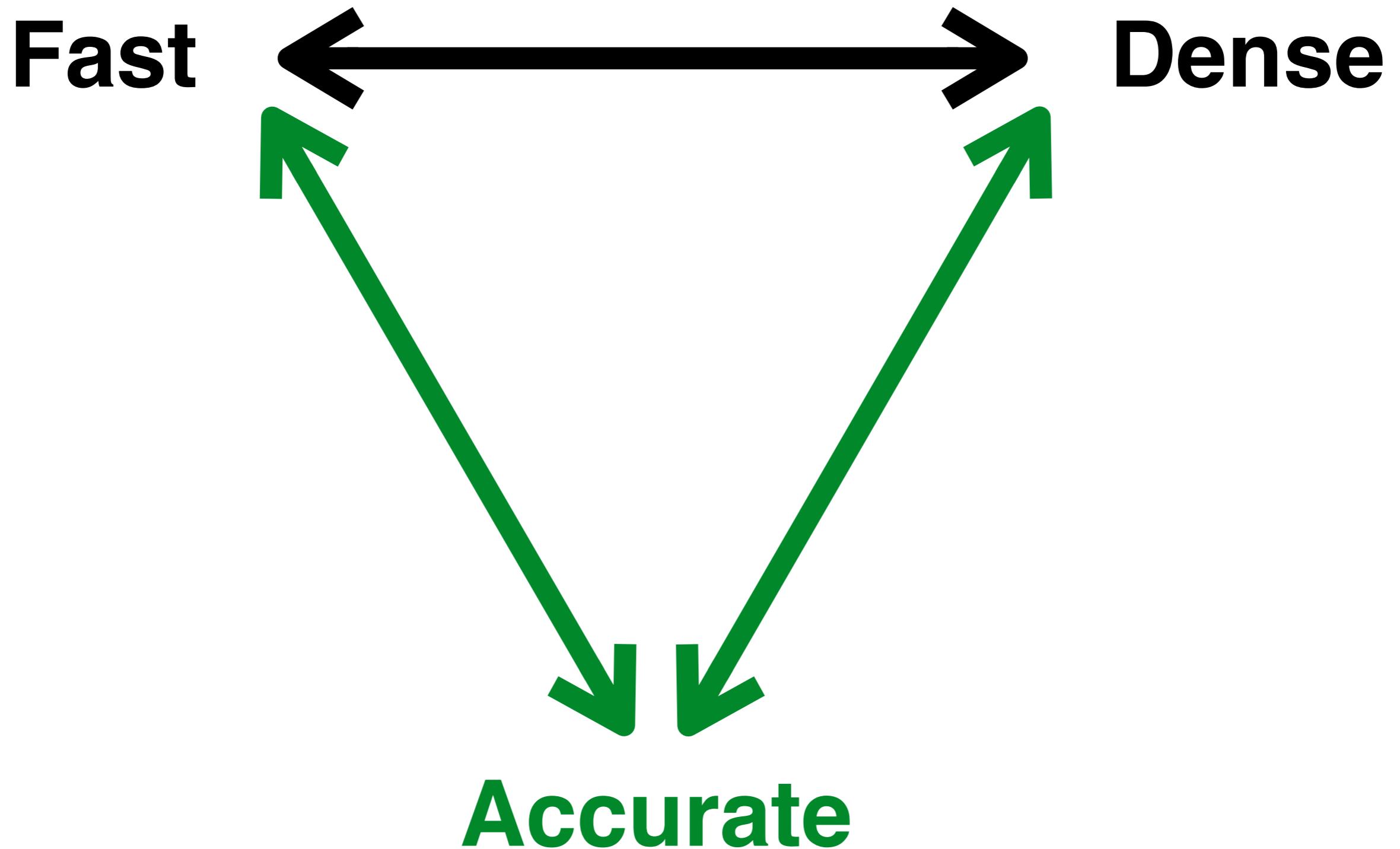
**Multi-level cells** are denser  
but need more time and energy  
**to protect against errors.**

Cells **wear out** over time  
and can no longer be used  
**for precise data storage.**

# Phase-change memory (PCM) :)

Fast  Dense

# Phase-change memory (PCM) :)



# Approximate storage in PCM

Trade off **accuracy** for performance in **multi-level cell** accesses.

Use **worn-out** memory for **approximate** data instead of throwing it away.

# Approximate storage in PCM

1

Trade off **accuracy** for performance in **multi-level cell** accesses.

2

Use **worn-out** memory for **approximate** data instead of throwing it away.

# Approximate storage in PCM

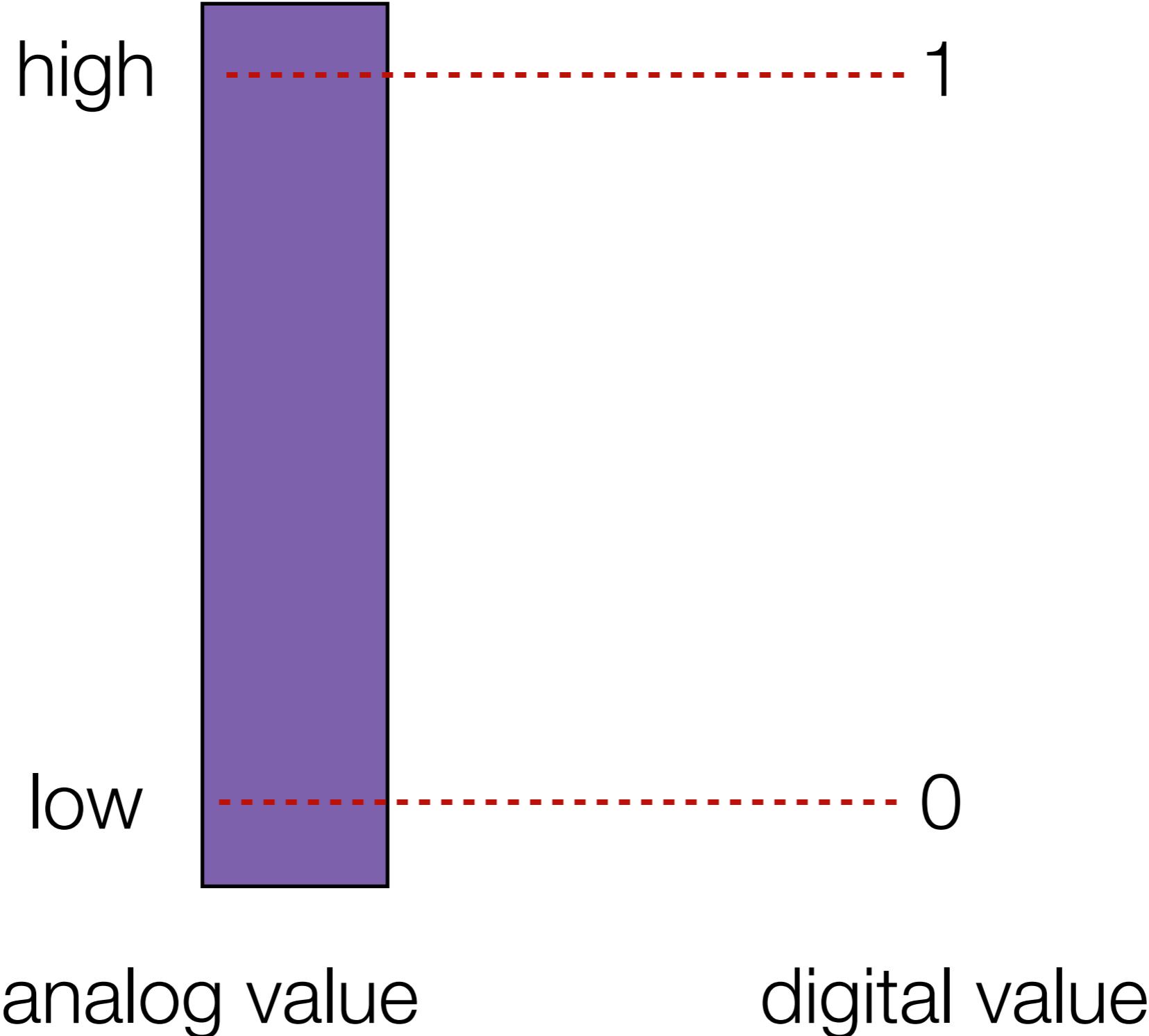
1

Trade off accuracy for performance in **multi-level cell** accesses.

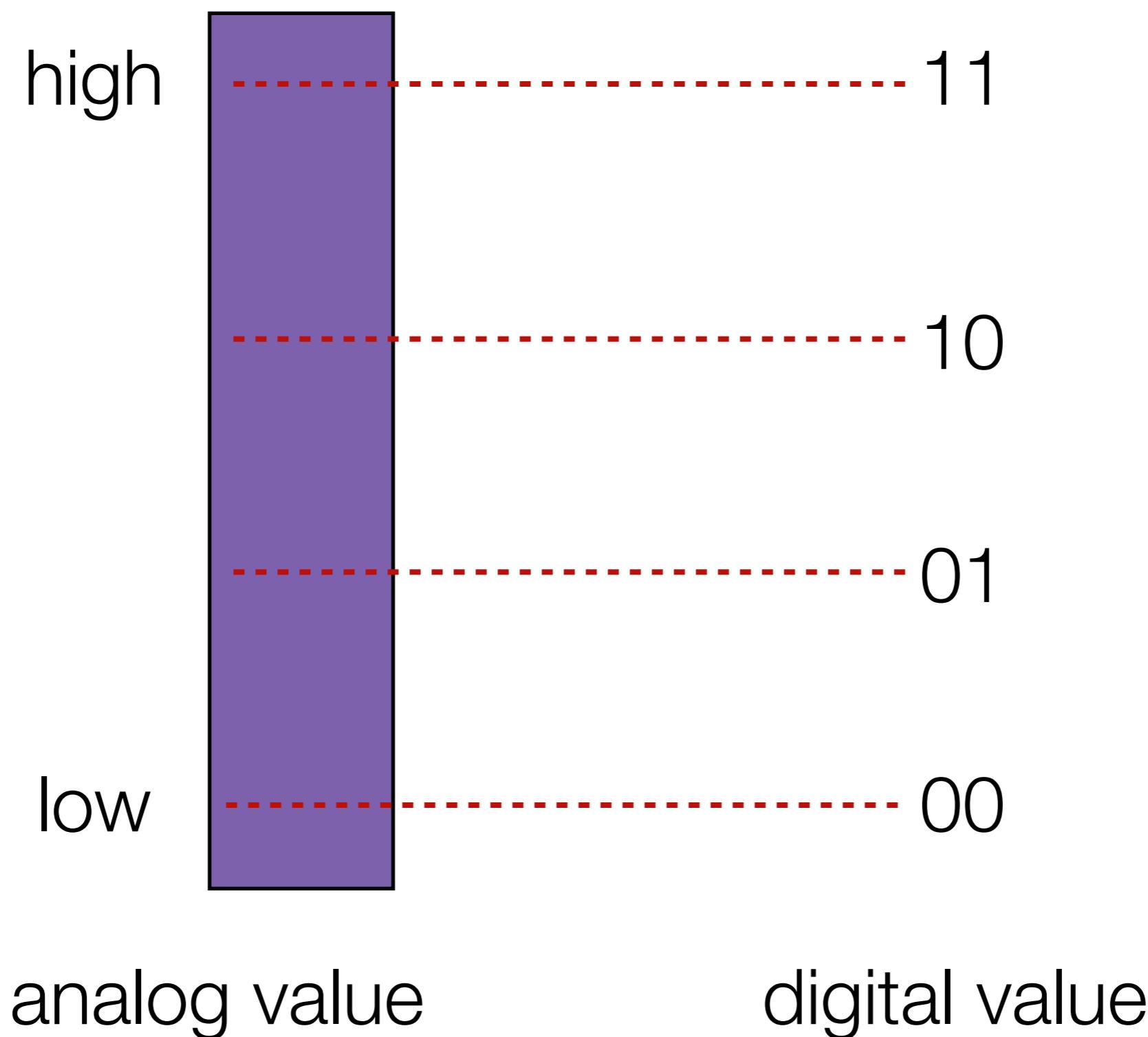
2

Use approximate throwing it away.

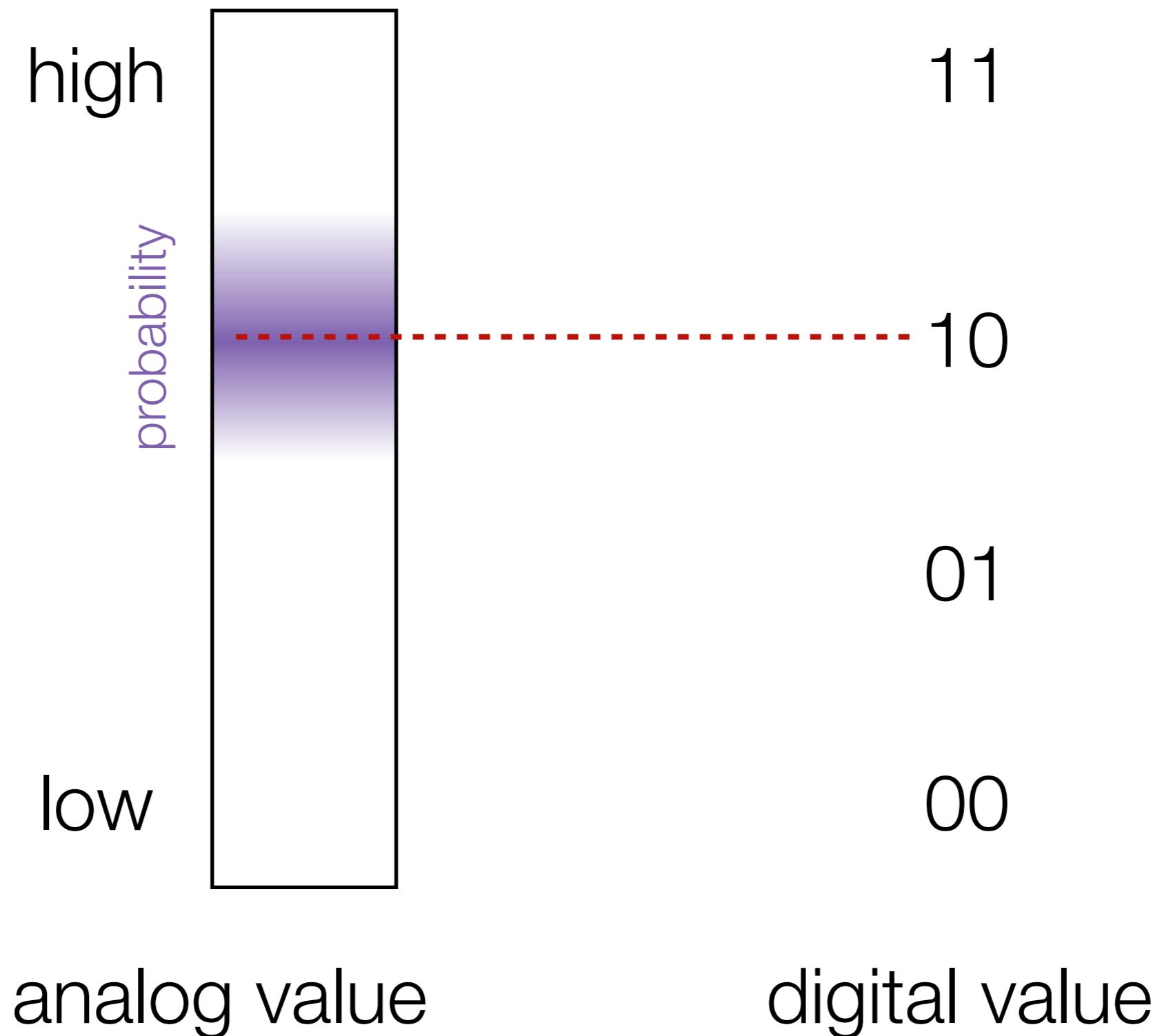
# Single-level cells



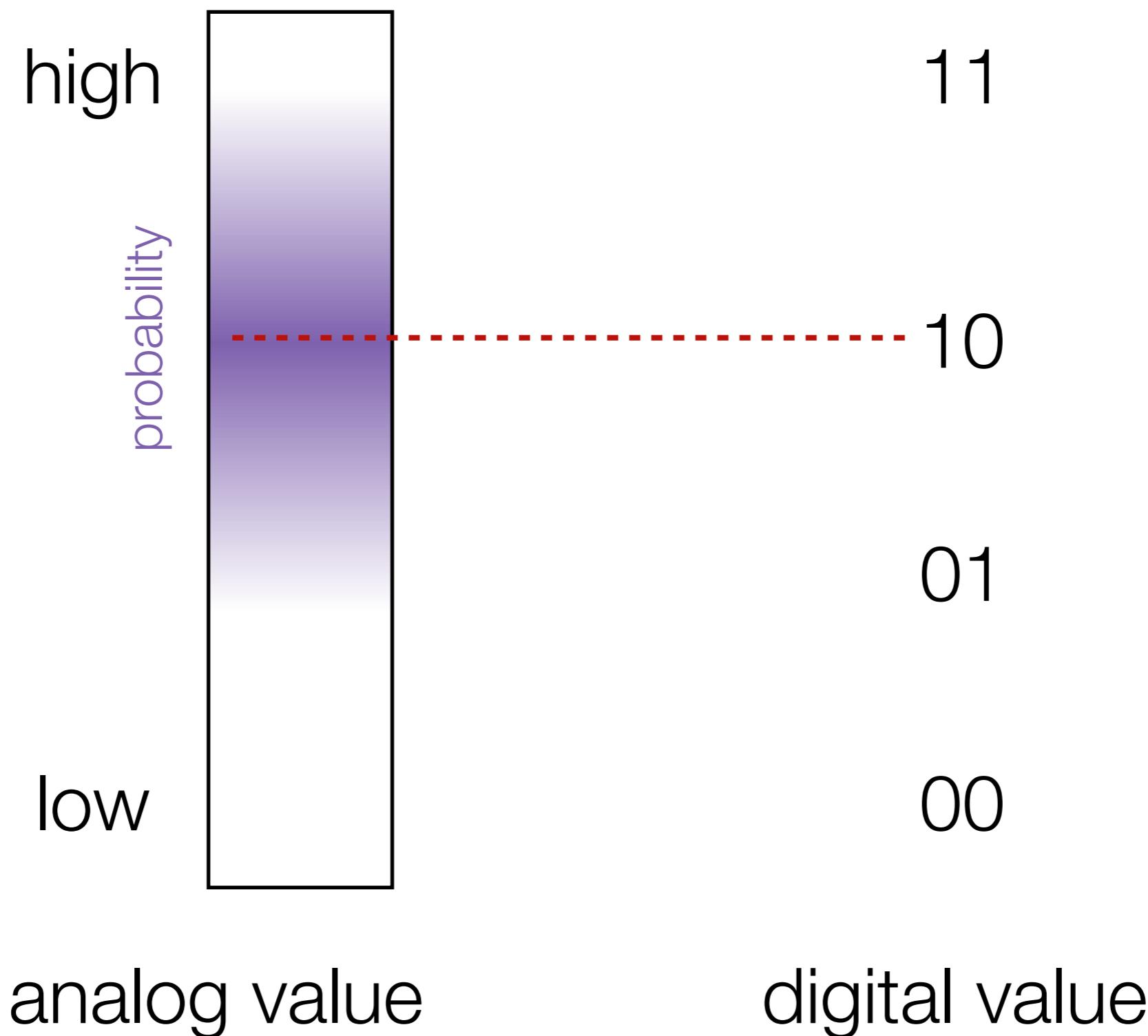
# Multi-level cells



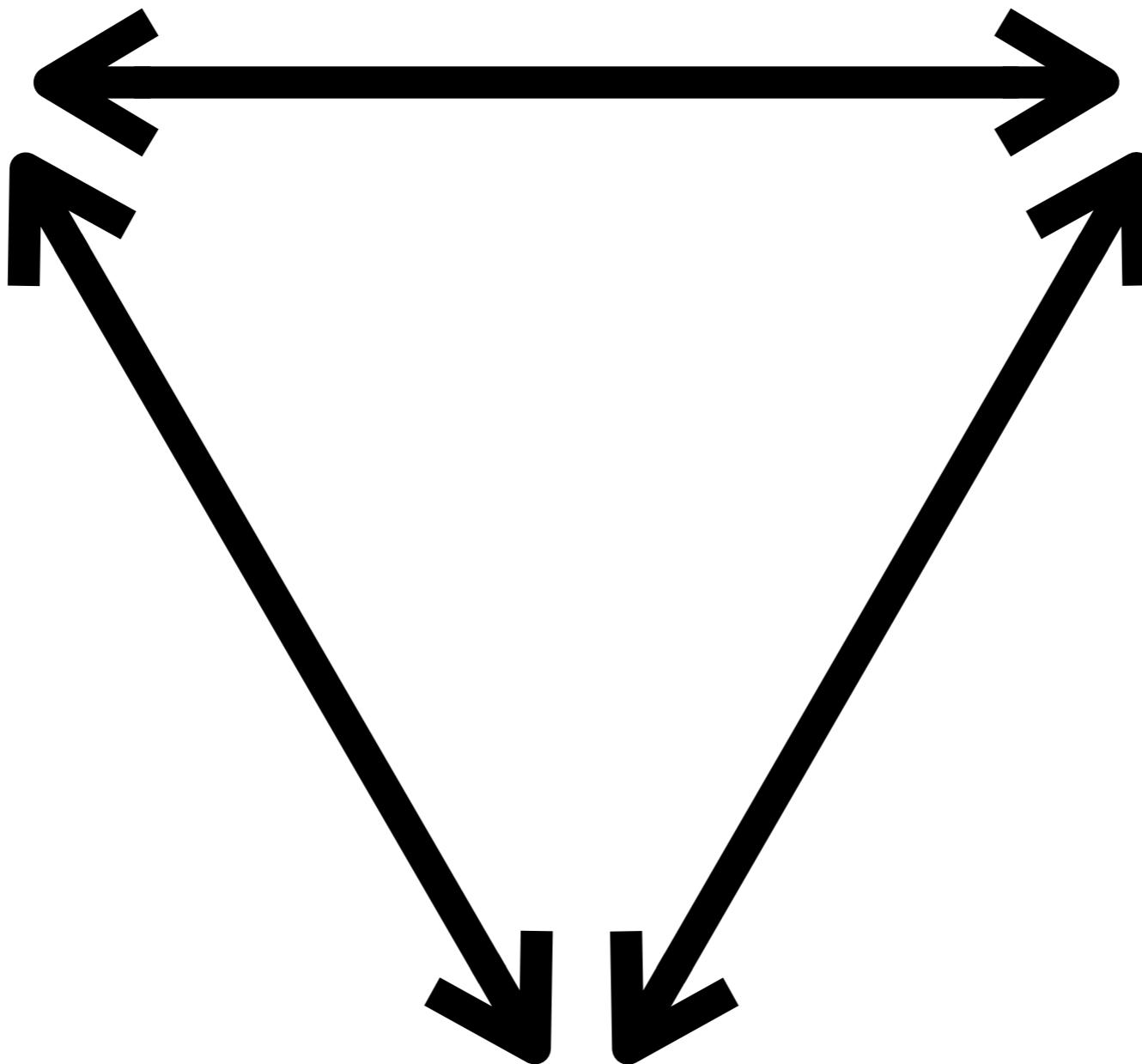
# Writing to multi-level cells



# Writing to multi-level cells, approximately

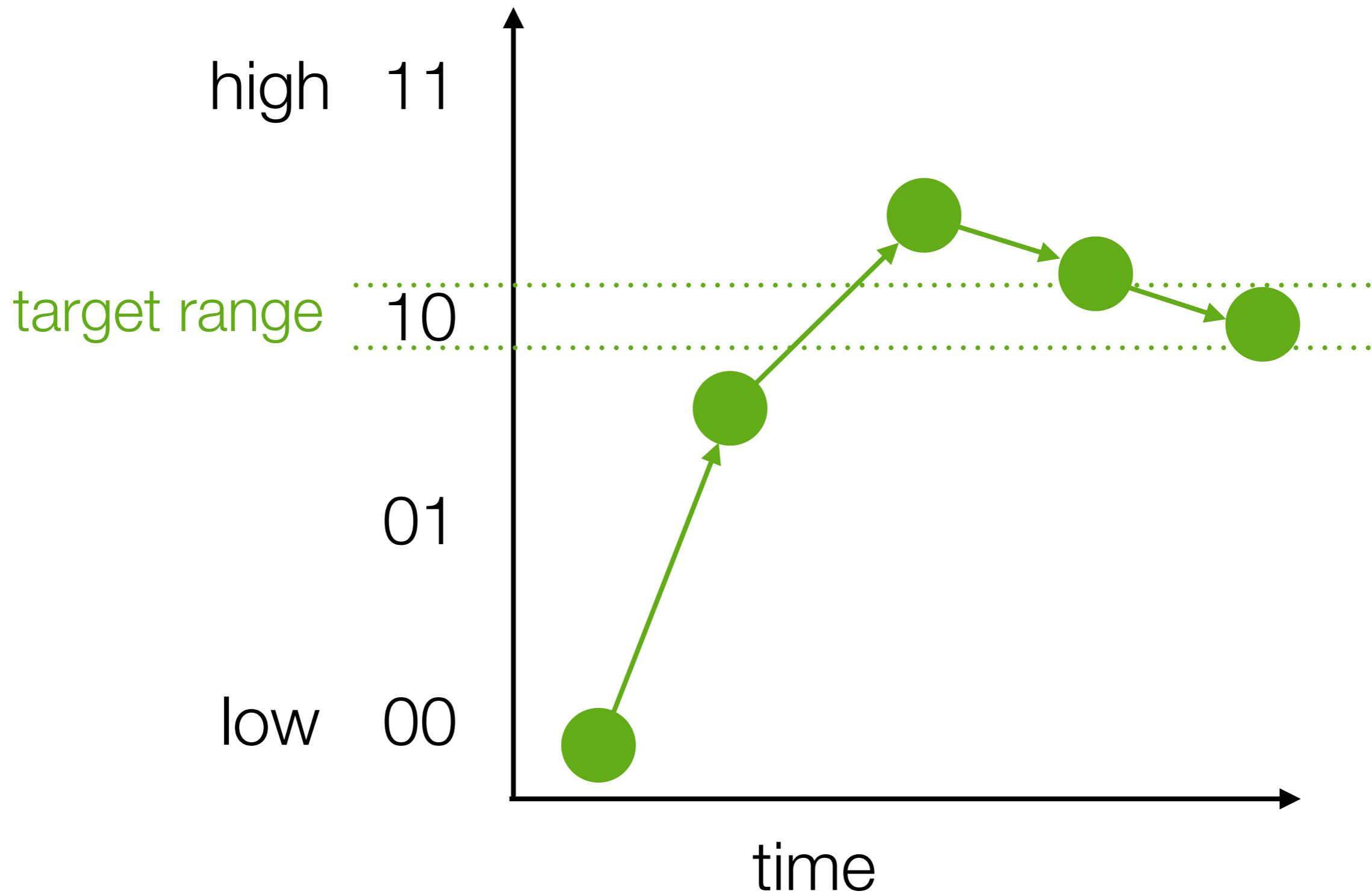


**Speed**      **Density**

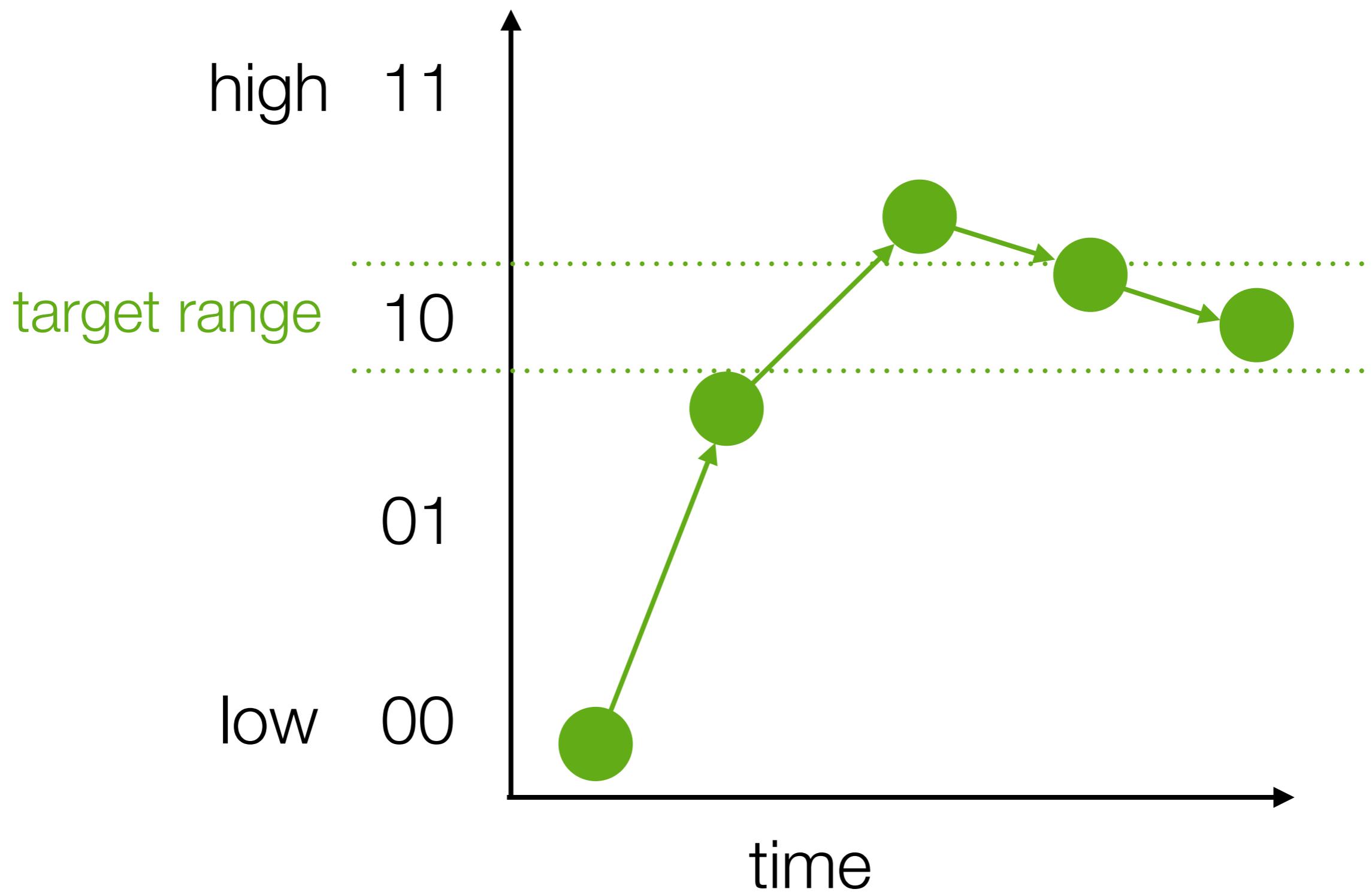


**Accuracy**

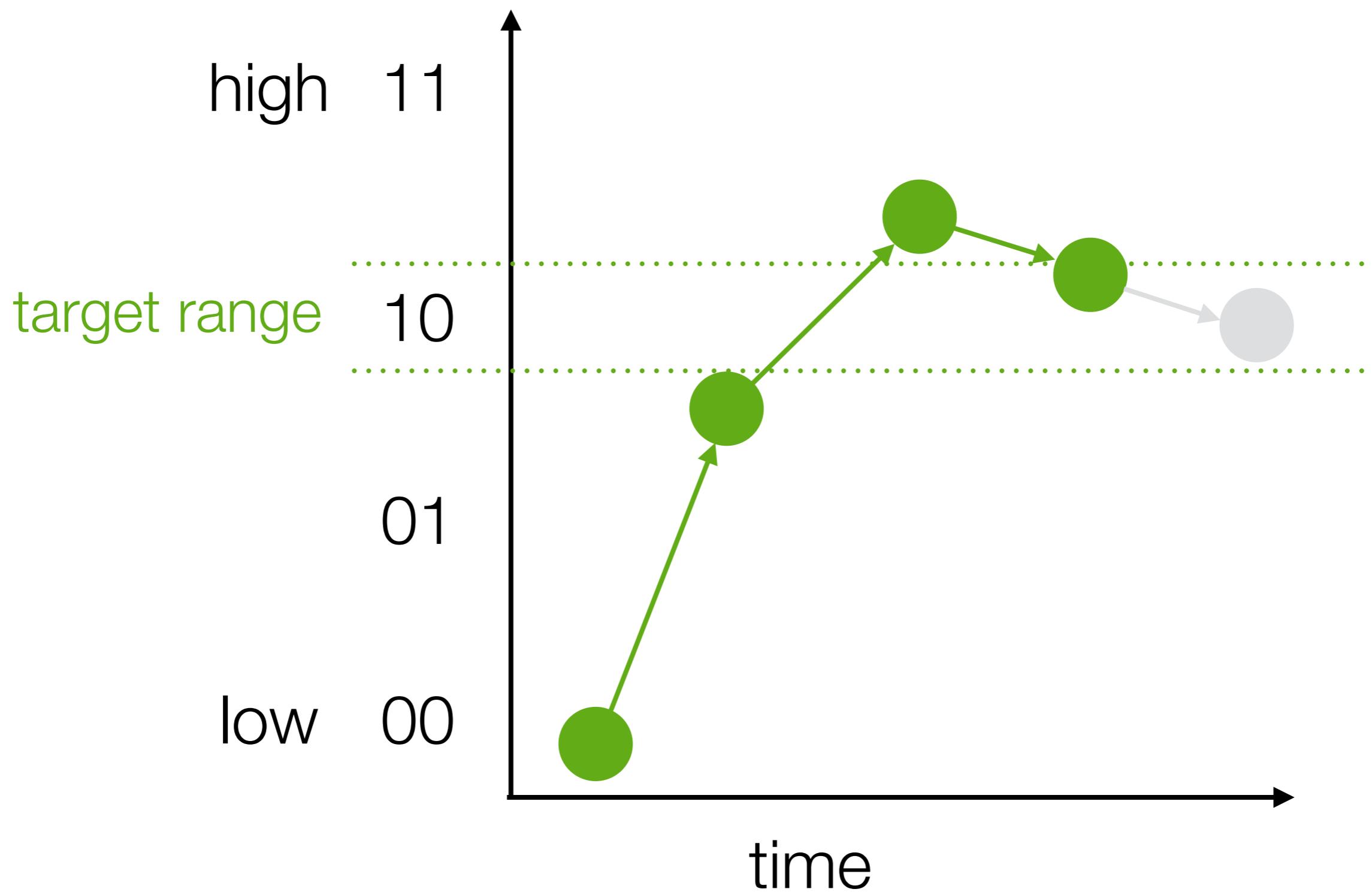
# Iterative writes



# Iterative writes, approximately



# Iterative writes, approximately



**wider target range**



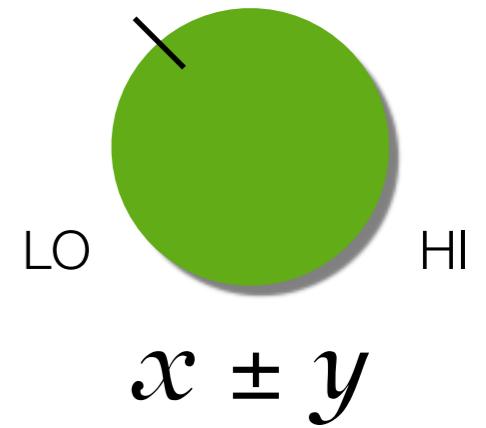
**fewer iterations to converge**



**faster writes**

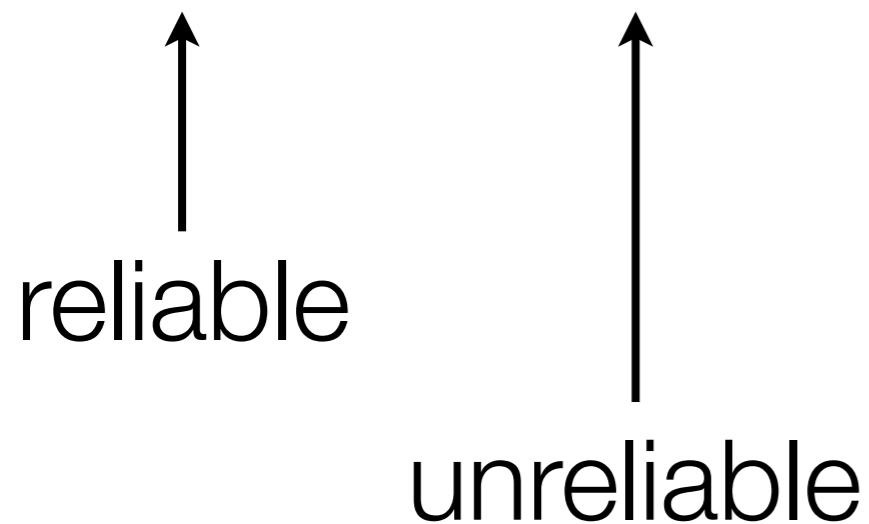
(or better density at the same speed)

# Encoding to minimize error in approximate MLC

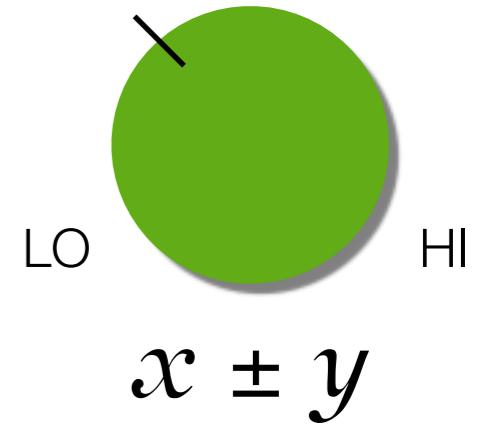


1 cell, 4 bits

|   |   |   |   |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
|---|---|---|---|



# Encoding to minimize error in approximate MLC



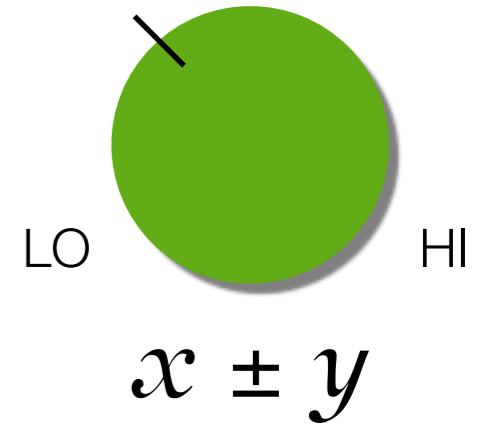
4 cells, 16 bits

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|



lots of errors

# Encoding to minimize error in approximate MLC



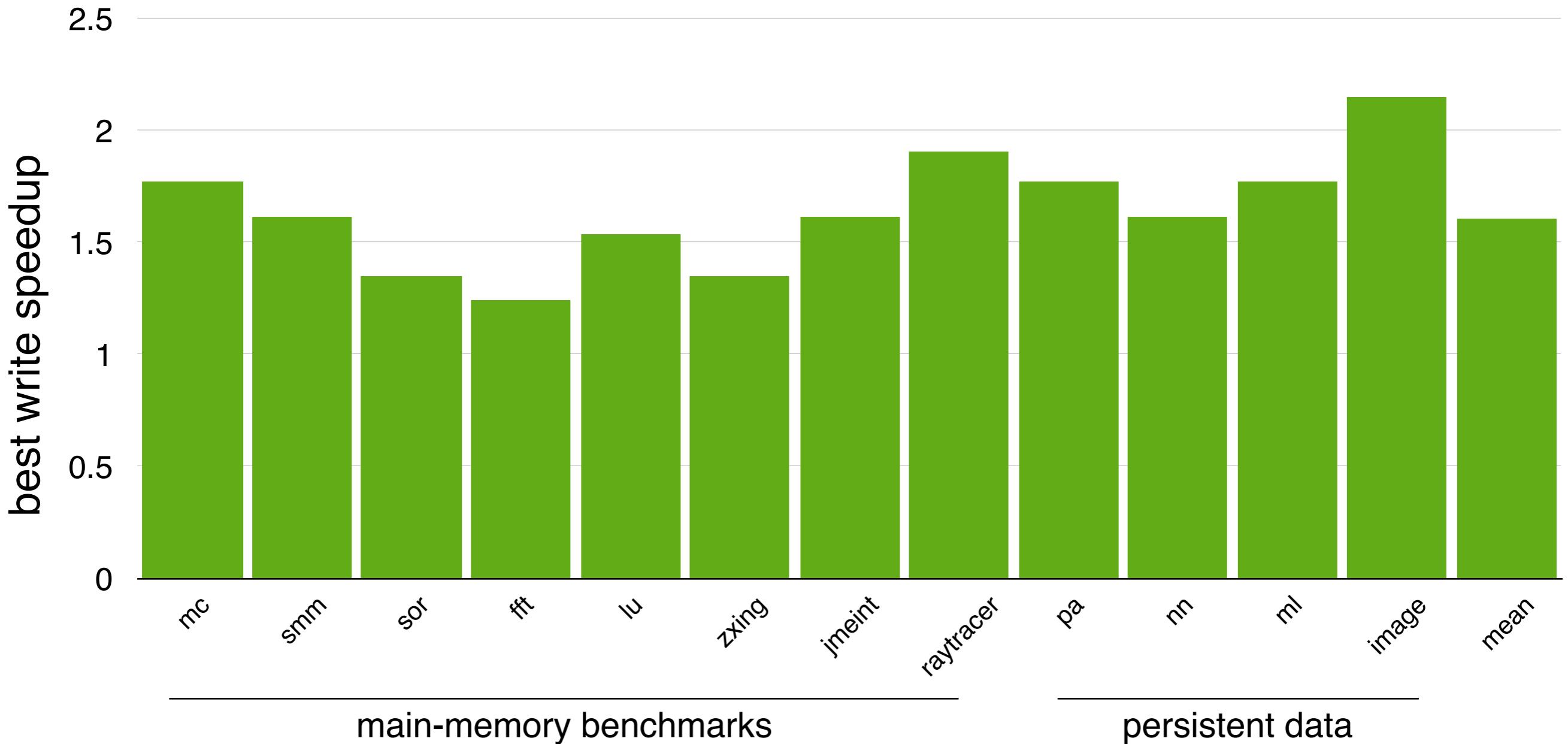
4 cells, 16 bits

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|



lots of errors

# Write speedup for approximate MLC



Writes are 1.7× faster on average  
with quality loss under 10%

# Approximate storage in PCM

Trade off  
performance in  
accesses.

Use **worn-out** memory for  
approximate data instead of  
throwing it away.

# Failed cells are a fact of life

```
0 1 1 0 1 0 0 1 1 1 0 1 0 0 1 0 1 1 1 0 0 0 1 1 0 0 1 0 1 1 0 0 1 0 0 0 1
```

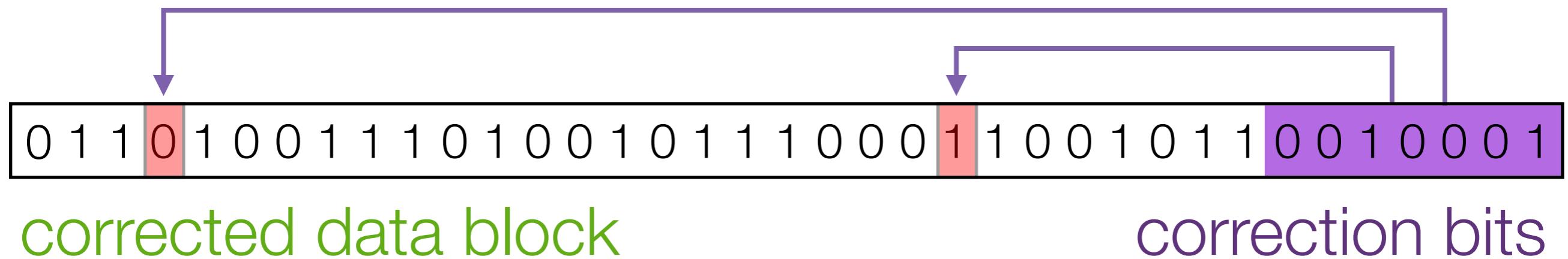
a good block

# Failed cells are a fact of life

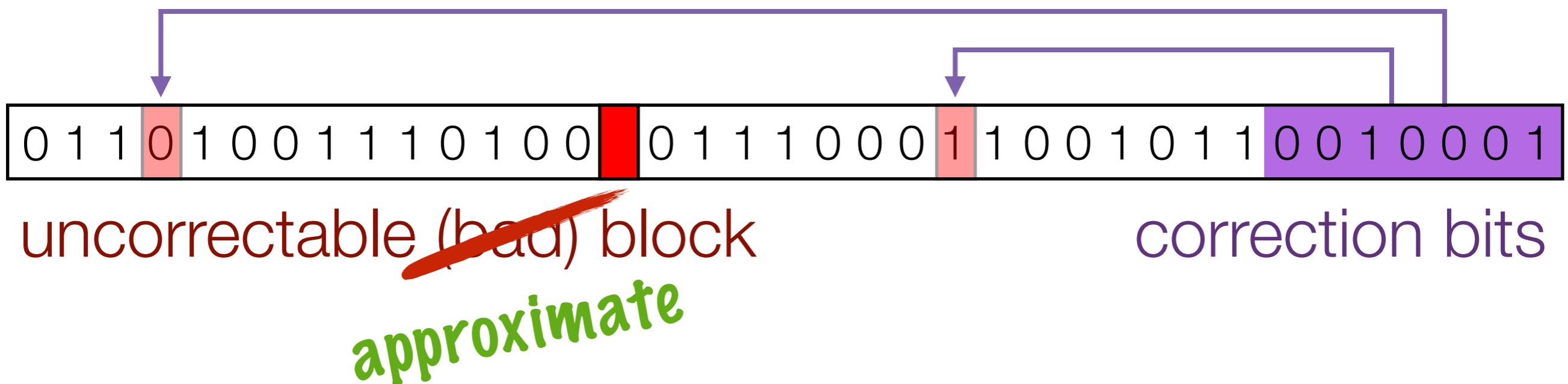
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

a (tragically) failed block

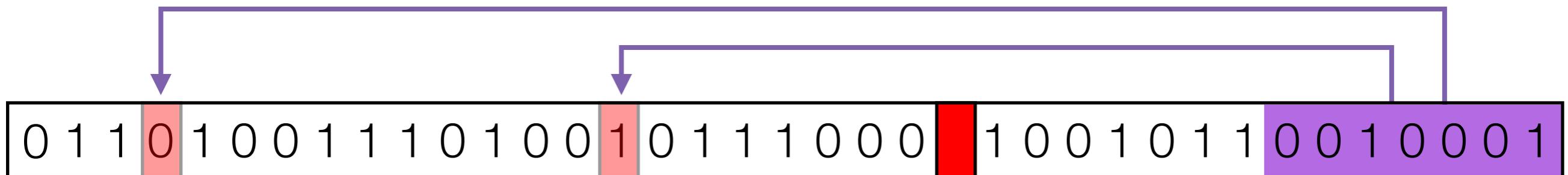
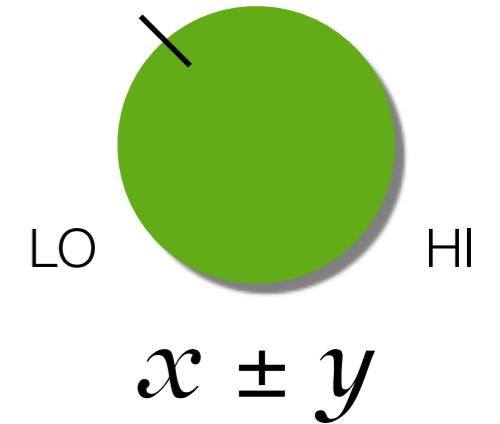
# Traditional error correction



# Correction resources are exhaustible



# Prioritized error correction



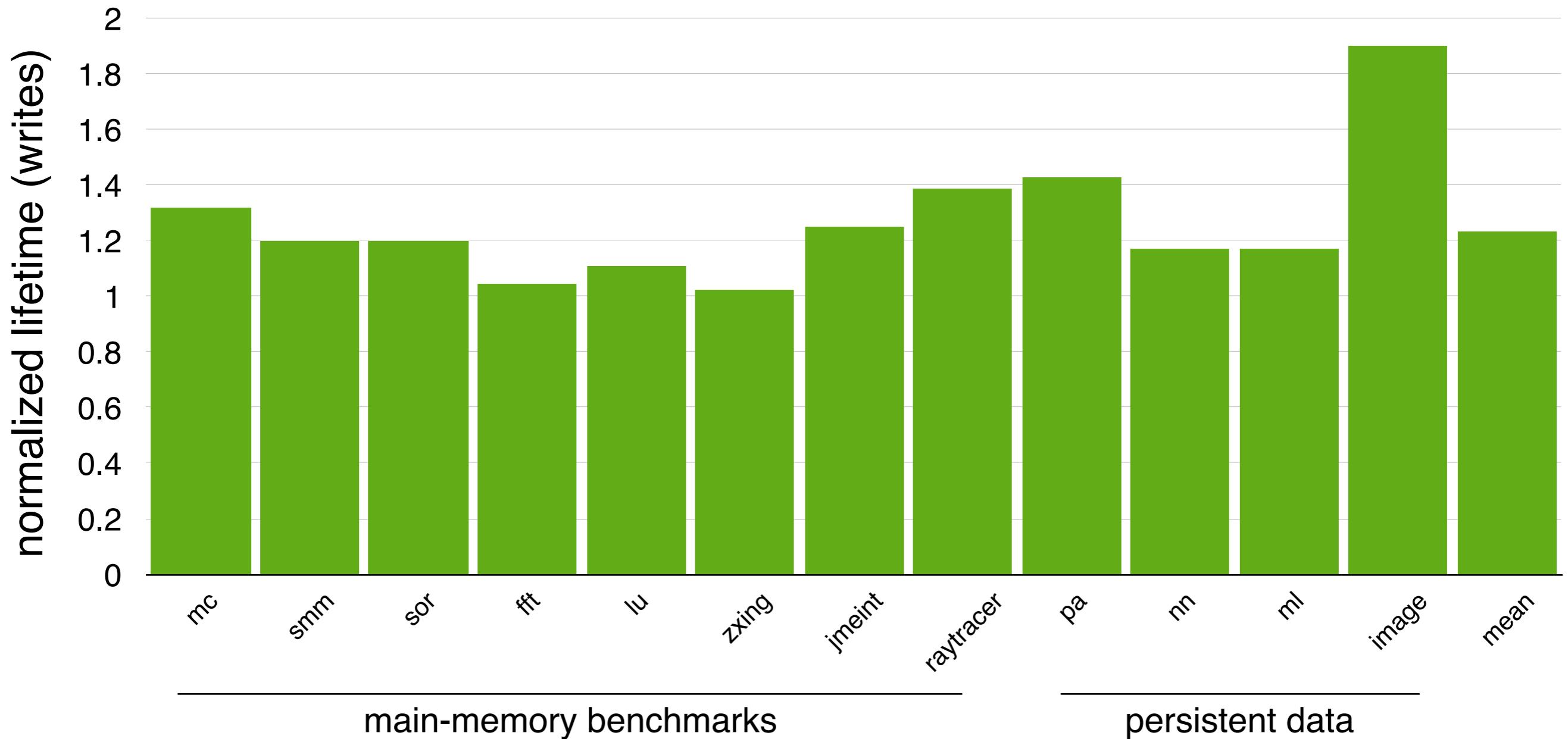
uncorrectable (bad) block

approximate

correction bits

error exposed  
where it does the least harm

# Lifetime extension with block recycling



Lifetime extended by 23% on average  
or from about 5.2 to 6.5 years

Network



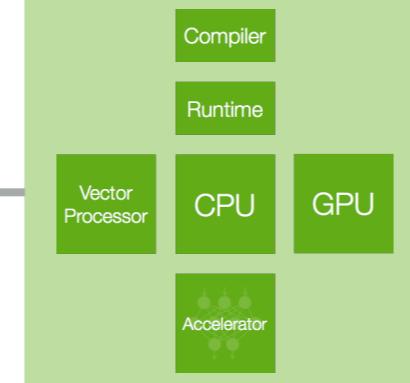
Display



I/O

Compute

Storage



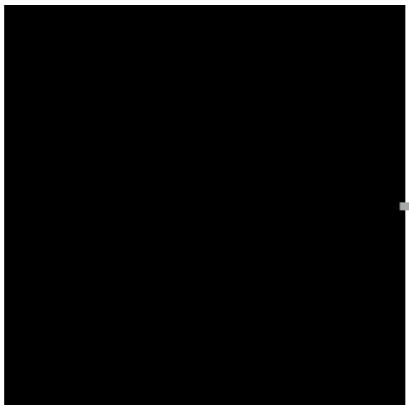
Disk

Memory

Network



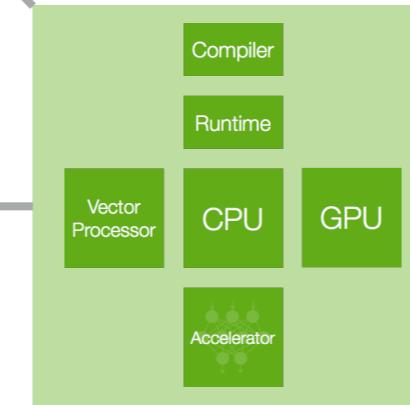
Display



I/O

Compute

Storage



Disk

Memory