DRAM RELIABILITY

Mahdi Nazm Bojnordi

Assistant Professor

School of Computing

University of Utah

Overview

- Upcoming deadline
  - April 7th (11:59PM): sign up for your student paper presentation

- This lecture
  - Memory errors
  - Error detection vs. correction
  - Memory scrubbing
  - Disturbance errors

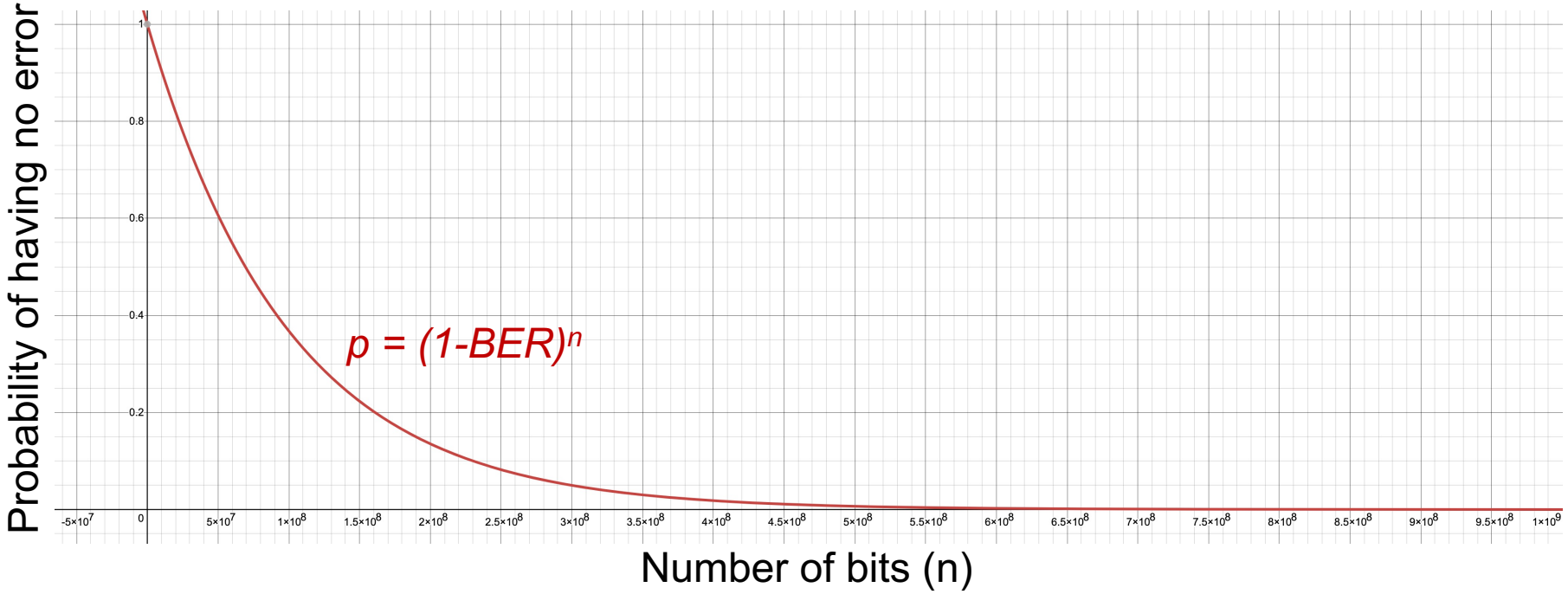
Memory Errors

- Any unwanted data change (bit flip)
  - storage cell
  - sensing circuits
  - wires
- Soft errors are mainly caused by
  - Slight manufacturing defects
  - Gamma rays and alpha particles
  - Electrical interference
  - …

Error Detection and Correction

- Main memory stores a huge number of bits
  - Nontrivial bit flip probability
  - Even worse as the technology scales down

$BER = 10^{-8}$



Error Detection and Correction

- Main memory stores a huge number of bits
  - Nontrivial bit flip probability
  - Even worse as the technology scales down
- Reliable systems must be protected against errors
- Techniques
  - Error detection
    - parity is a rudimentary method of checking the data to see if errors exist
  - Error correction code (ECC)
    - additional bits used for error detection and correction

Error Correction Codes

- Example: add redundant bits to the original data bits
  - SECDED: Hamming distance (0000, 1111) = 4

Power	Correct	#bits	Comments
Nothing	0,1	1	
Single error detection (SED)	00,11	2	01,10 =>errors
Single error correction (SEC)	000,111	3	001,010,100 => 0 110,101,011 => 1
Single error correction double error detection (SECDED)	0000,1111	4	One 1 => 0 Two 1's => error Three 1's => 1

[adopted from Lipasti]

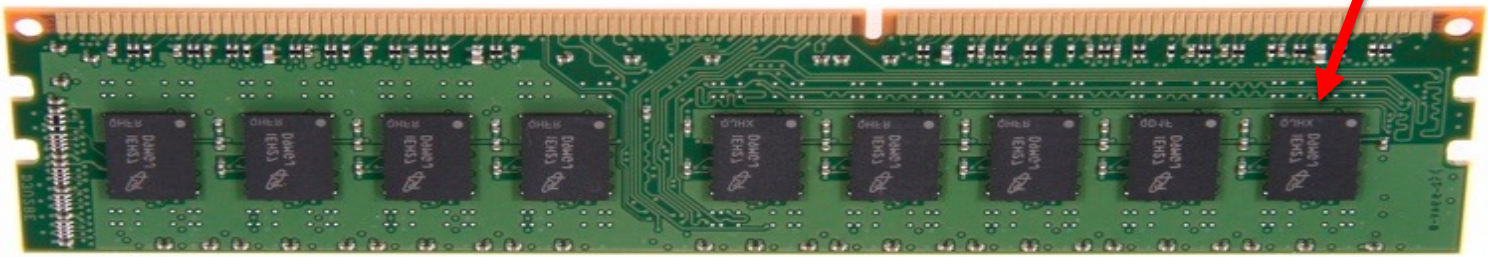
Error Correction Codes

- Reduce the overhead by applying the codes to words instead of bits

# bits	SED overhead	SECDED overhead
1	1 (100%)	3 (300%)
32	1 (3%)	7 (22%)
64	1 (1.6%)	8 (13%)
n	1 (1/n)	1 + $\log_2$ n + a little

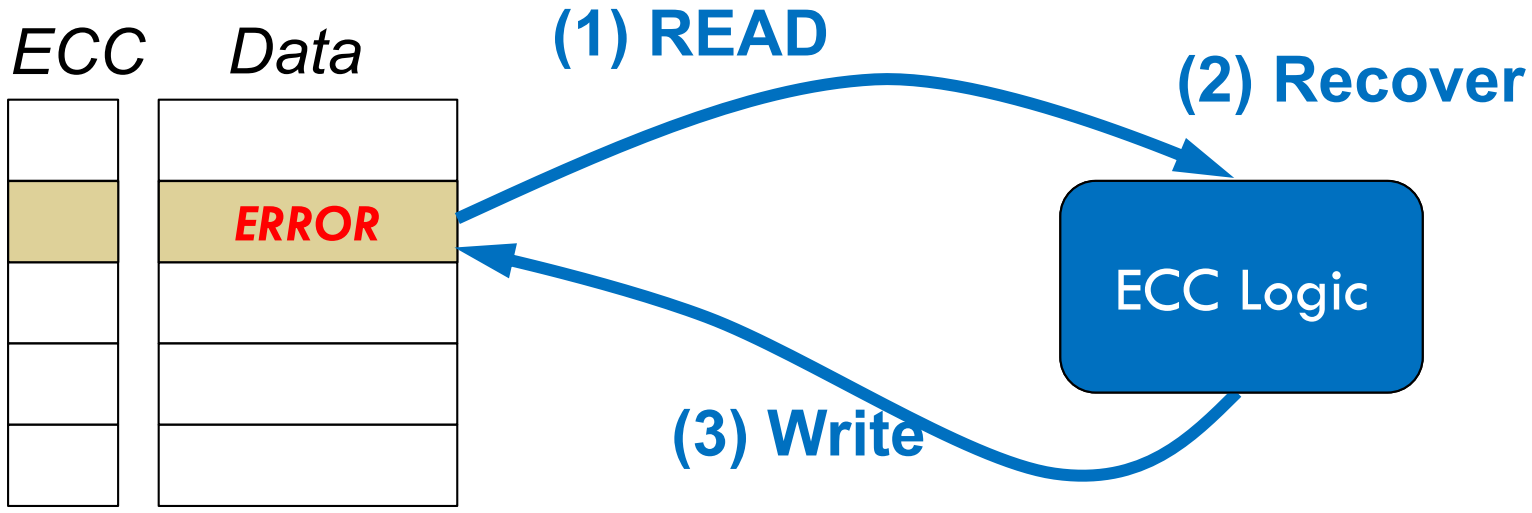
ECC Chip

ECC DIMM (9 x8 chips)

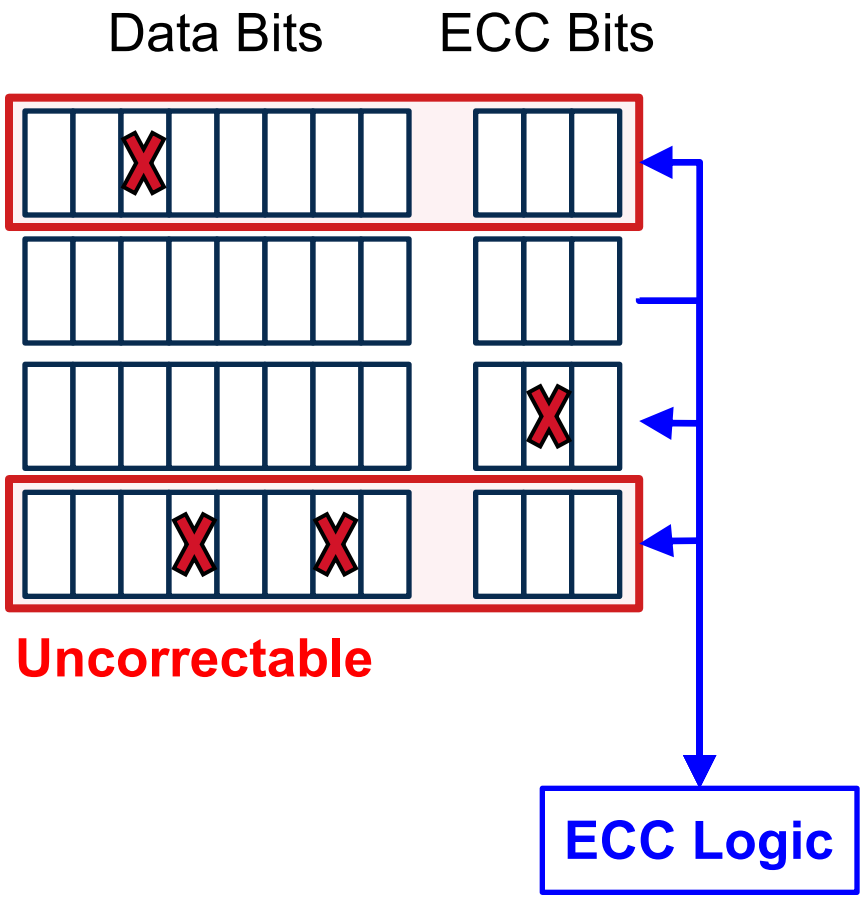


Memory Error Correction

- ECC allows the memory controller to correct cell retention errors and relax memory cell retention requirements.



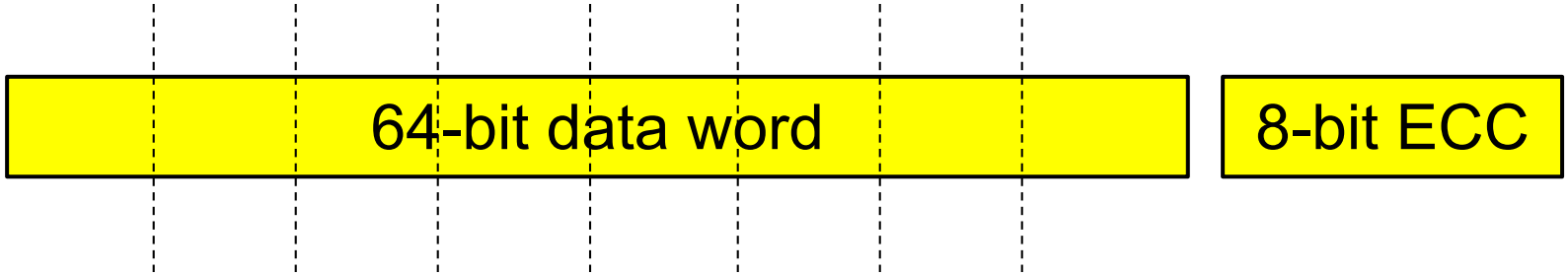
Memory Scrubbing



- ECC can correct a fixed number of errors
- Data can become uncorrectable if ECC is used without scrubbing
- Scrubbing prevents errors from accumulating over time
  - Periodically read all of the memory locations
  - Check ECC, correct errors, and write corrected data back to memory

Stronger Error Corrections

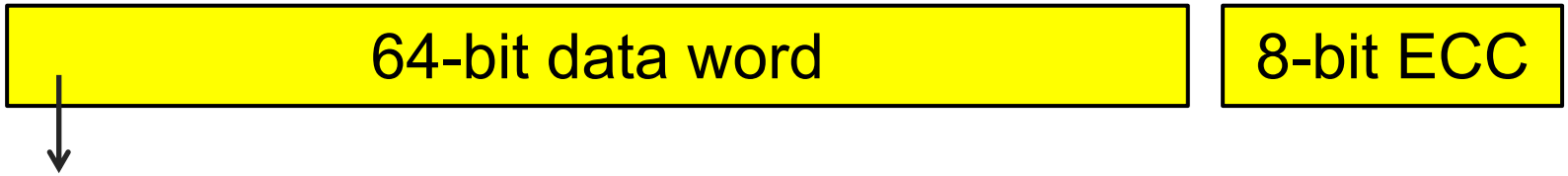
- SECDED Support



- One extra x8 chip per rank
- Storage and energy overhead of 12.5%
- Cannot handle complete failure in one chip

Stronger Error Corrections

- SECDED Support
- Chipkill Support

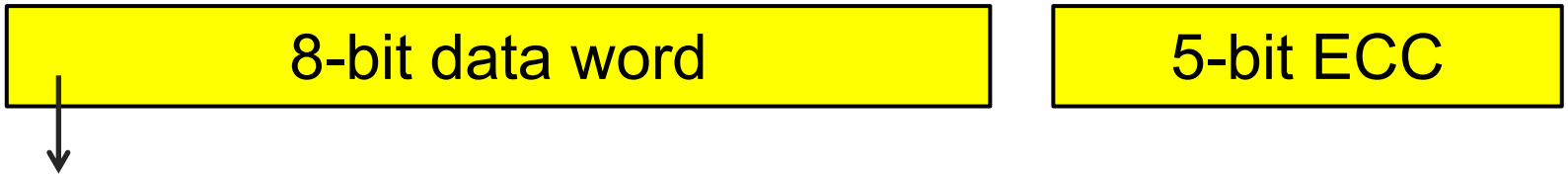


At most one bit from each DRAM chip

- Use 72 DRAM chips to read out 72 bits
- Dramatic increase in activation energy and overfetch
- Storage overhead is still 12.5%

Stronger Error Corrections

- SECDED Support
- Chipkill Support



At most one bit from each DRAM chip

- Use 13 DRAM chips to read out 13 bits
- Storage and energy overhead: 62.5%
- Other options exist; trade-off between energy and storage

Row Hammer Problem

- Repeated row activations can cause bit flips in adjacent rows

Available for: OS X Mountain Lion v10.8.5, OS X Mavericks v10.9.5

Impact: A malicious application may induce memory corruption to escalate privileges

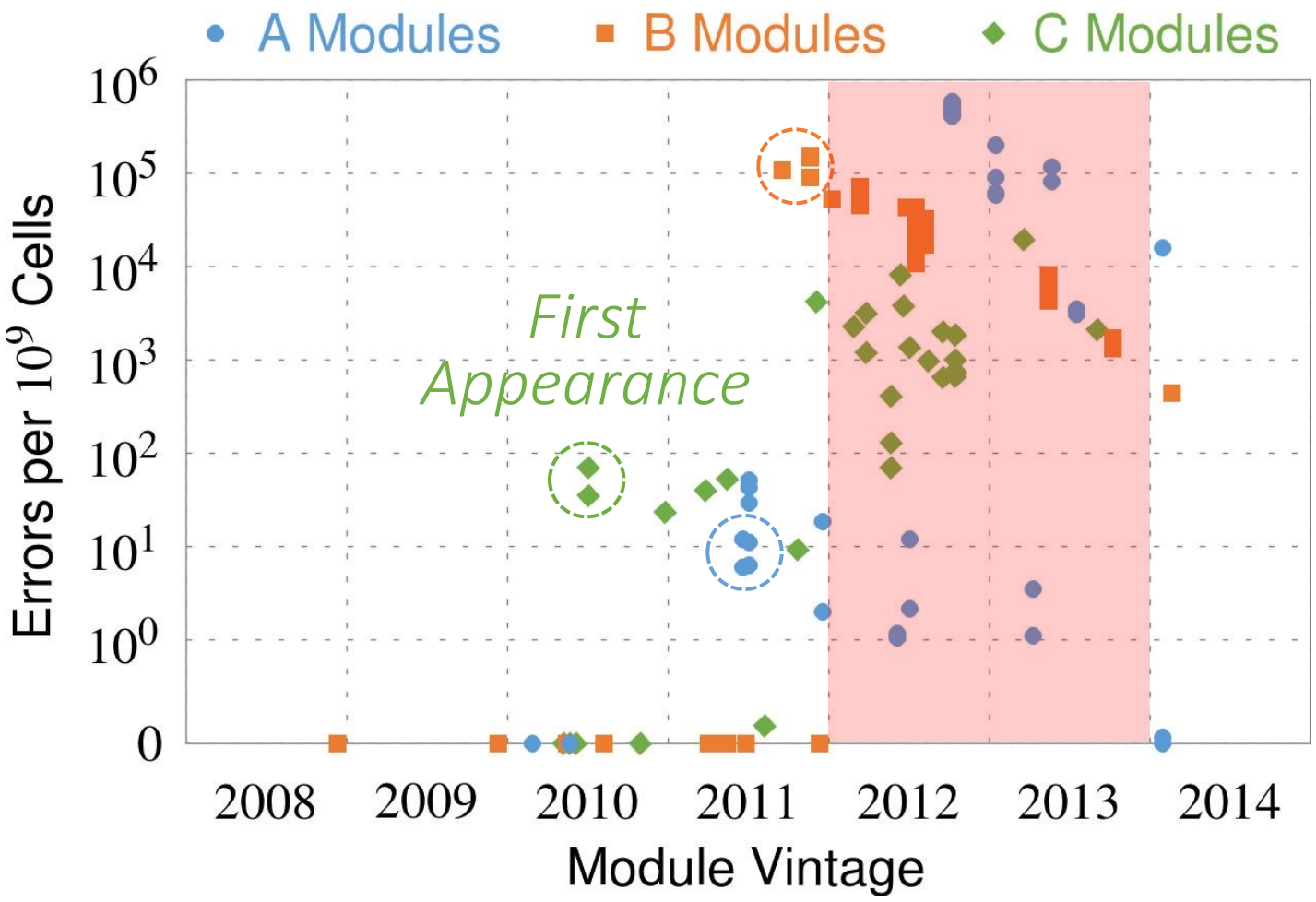
Description: A disturbance error, also known as Rowhammer, exists with some DDR3 RAM that could have led to memory corruption. This issue was mitigated by increasing memory refresh rates.

CVE-ID

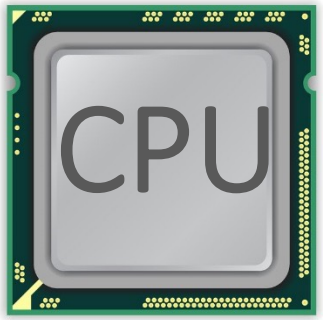
CVE-2015-3693 : Mark Seaborn and Thomas Dullien of Google, working from original research by Yoongu Kim et al (2014)

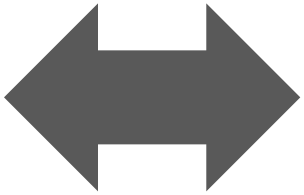
[Apple]

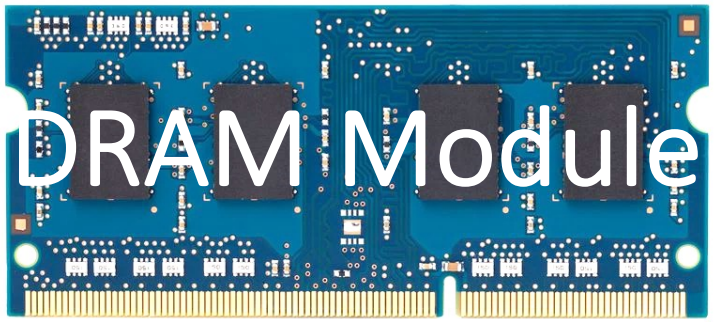
Modern DRAM is Vulnerable



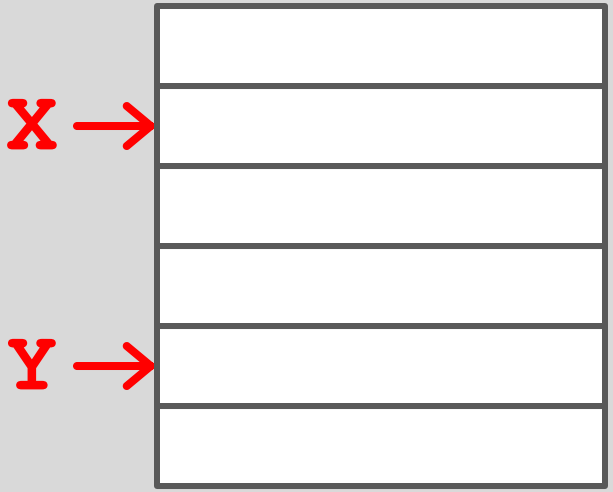
How Program Induces RH Errors?








```
loop:
   mov (X), %eax
   mov (Y), %ebx
   clflush (X)
   clflush (Y)
   mfence
   jmp loop
```



Sources of Disturbance Errors

- *Cause 1: Electromagnetic coupling*
  - Toggling the wordline voltage briefly increases the voltage of adjacent wordlines
  - Slightly opens adjacent rows → Charge leakage
- *Cause 2: Conductive bridges*
- *Cause 3: Hot-carrier injection*

Confirmed by at least one manufacturer

[slide source:Mutlu]

Basic Solutions

- *Throttle accesses to same row*
  - Limit access-interval: $\geq 500\text{ns}$
  - Limit number of accesses: $\leq 128\text{K}$ $(=64\text{ms}/500\text{ns})$

- *Refresh more frequently*
  - Shorten refresh-interval by $\sim 7\text{x}$

Both naive solutions introduce significant overhead in performance and power

[Kim'2014]

Probabilistic Adjacent Row Activation

- Key Idea
  - After closing a row, we activate (i.e., refresh) one of its neighbors with a low probability: $p = 0.005$
- Reliability Guarantee
  - When $p=0.005$, errors in one year: 9.4×10^{-14}
  - By adjusting the value of p, we can vary the strength of protection against errors

[Kim'2014]