THE UNIVERSITY OF TEXAS AT AUSTIN

Toward Exascale Resilience

Part 3: Fault and error modes and models

Mattan Erez

The University of Texas at Austin

July 2015

Why care about models?

Why care about models?

- Resilience isn't free
- Paying too much overhead means losing out

Modeling techniques

- “First-principles” (simulation and analytical)
  - How detailed?
  - What parameters?
- Empirical
  - Accelerated testing
  - Field studies
- Extrapolations

Different components – different techniques

- Academic are not in the best position here

Field studies

- Collect fault, error, failure reports from actual systems
- Analyze
- Draw conclusions
- Extrapolate

Field studies are hard

- Limited reporting
- Limited information
- Limited access
- Sensitive outcomes

But, that’s the actual world

Some excellent examples in recent years

- In particular for DRAM
  - Starting from Sridharan and Liberty, SC'12
- Good sources for other components too
  - Recent analysis on supercomputers and datacenters
  - Some other specific components too

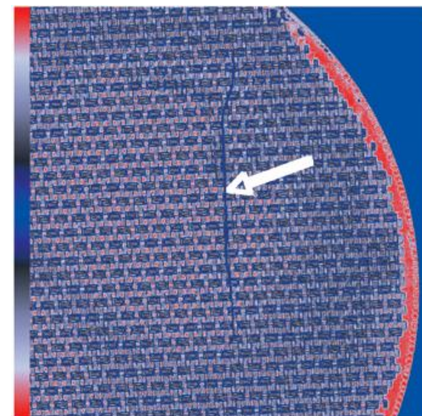
Fault vs. error/failure analysis

- Accelerated testing
  - Baking
  - Beam testing
- Excellent, but
  - Special facilities
  - Special equipment

Back to basics

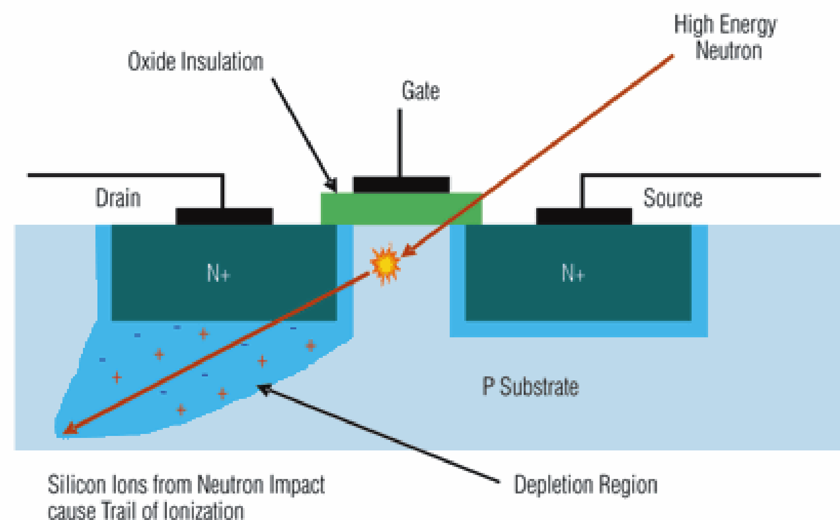
Hard faults and intermittent errors

- Design flaws
- Fabrication defects
- Process variation
- Mechanical stress
- Gradual wearout



Soft faults

- Alpha particles
- Cosmic rays
- Power supply variation
- Voltage droops
- Crosstalk



Hardware summary

- Gates, wires, and memory devices
- Memories and processors (sockets)
- Modules (w/ VRU)
- Shared power, cooling, and links (w/ VRU)
- Cabinets (w/ VRU)
- Storage separated out (for now)

- Hard faults in processors
  - Manufacturing
  - Mechanical
  - Wearout
- Industry probably has these under control
  - **ALL** customers need this to work
  - Several recent studies suggest it is indeed the case

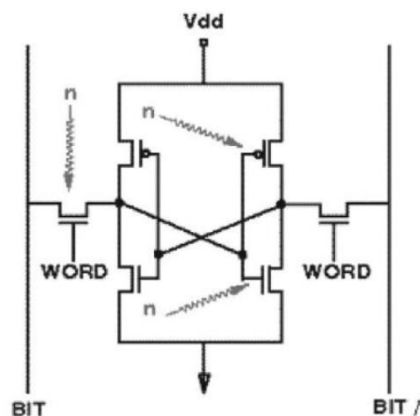
But not clear what is sacrificed

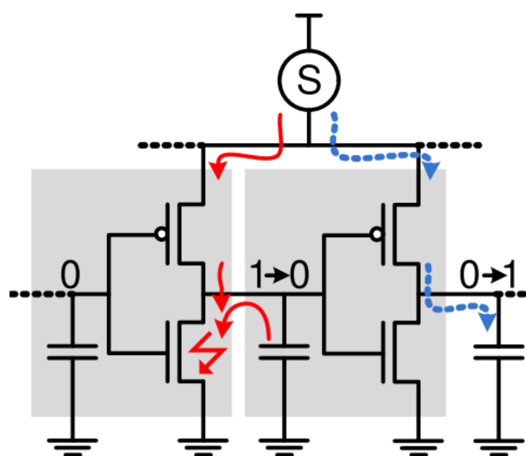
- How much better could we do if system exposed to processor hard faults?

Example: gradual wearout

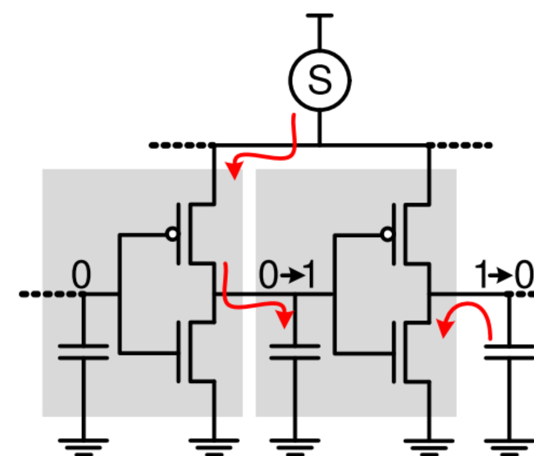
Particle-strike soft errors

- Ionizing particle frees electron-hole pairs in transistor active region
- Charge collected at source/drain creating current in an off device
- Current can flip a bit in a storage device or change value of a logic computation





(a) Stage 1 : strike and flip



(b) Stage 2 : revert

Why are particle strikes problematic?

- Two main sources
  - Alpha particles
  - Atmospheric neutrons
- Alpha particles from packaging and solder
  - Can be reduced but not eliminated completely
- Neutrons are highly penetrating and flux can be high
  - Neutrons randomly strike an atom and create secondary ionizing particles

Difficult to stop, but rare

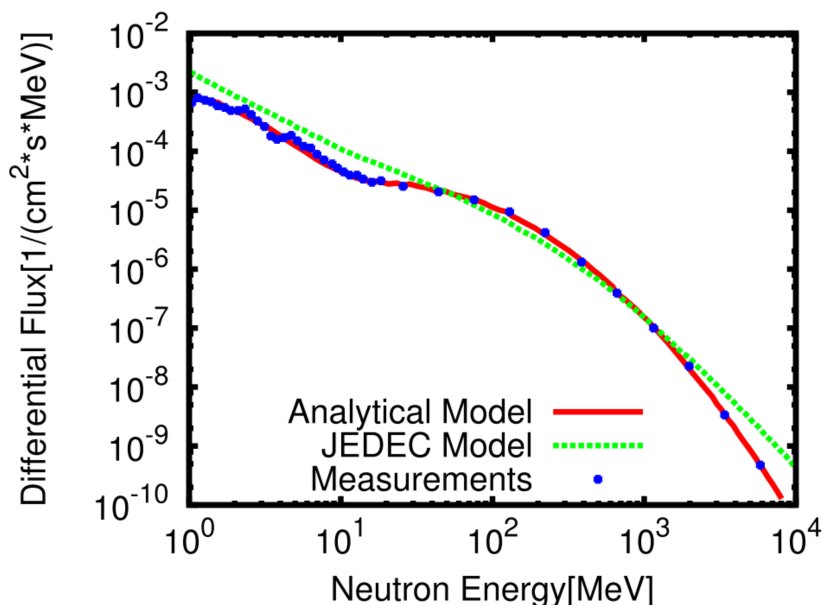
- Protection overhead is high and generally “wasted”
- Minimize waste by understanding vulnerability and trends

Need models for:

- Particle flux, charge collection, and component vulnerability

Likelihood of Strike Directly Proportional to Particle Flux

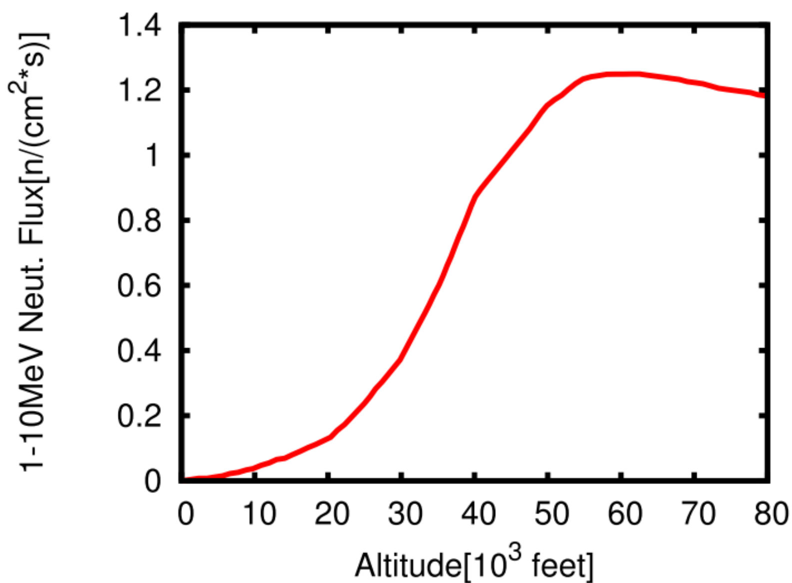
- Alpha particles are low and constant flux
- Neutrons more interesting and problematic
- Significant number of high-energy neutrons
- Even higher flux at lower energies
  - Lower energies may affect future technologies more

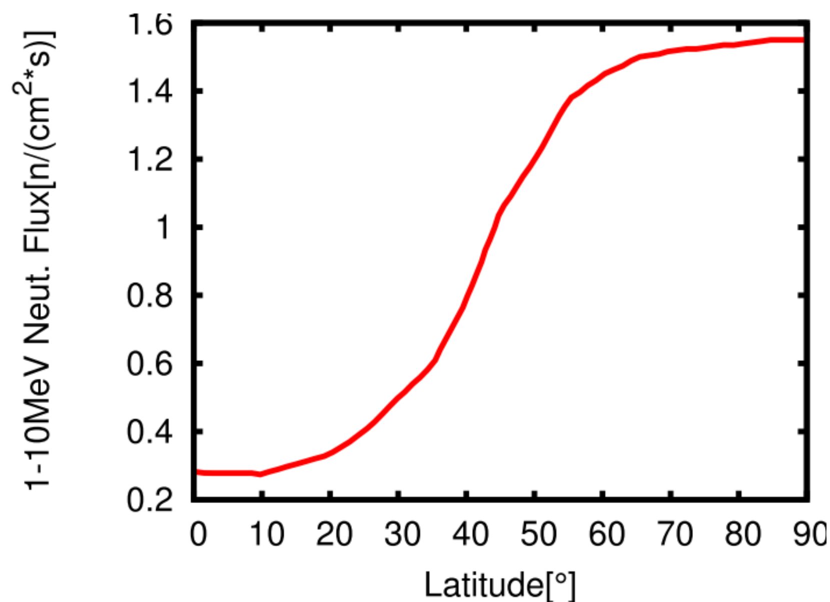


Flux Varies Significantly with Altitude and Latitude

Flux at 50,000’ is 1000 times higher than at sea level

Flux towards poles 5 times higher than equator





Impact of a Particle Strike

Amount of freed charge

- Energy of particle (nature)
- Angle of incidence (generally random)
- Properties of material (hard to gauge)

Efficiency of charge collection

- Volume of depletion zone (shrinks with tech. scaling)
- Supply voltage (small linear impact)
- Temperature (minimal impact)

Required current/charge to cause an error

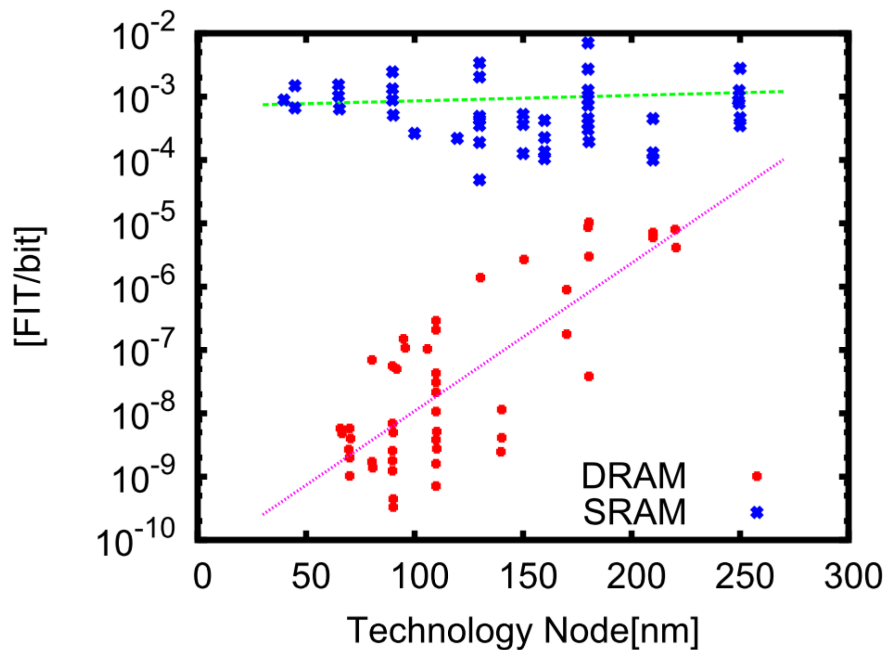
- Q_{crit} is minimal charge to flip an SRAM bit
- Similar notion for latches
- More complex for logic

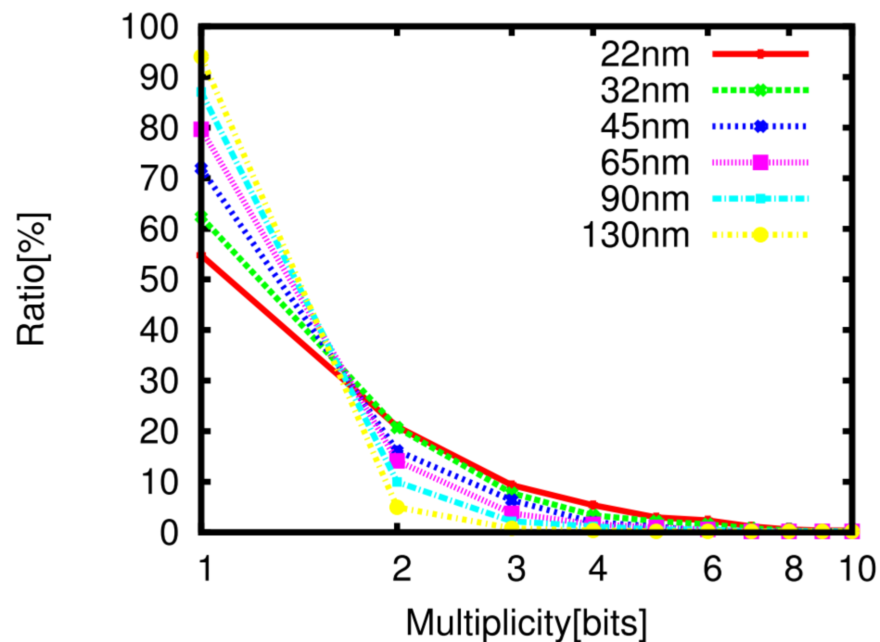
SRAM Trends

- Q_{crit} decreases with technology scaling
  - Lower capacitance to overcome for a flip (linear)
  - Lower voltage lowers Q_{crit} (linear and small)
- Collected charge also decreases somewhat
- Process variation increases error propensity
  - Variation increases as technology scales
- New transistor technology reduces error propensity
  - Also helps limit variation
- FIT per bit expected roughly constant
- Denser devices increase number of bits and also likelihood of multi-bit upsets

SRAM Trends

Multi-bit errors significant concern





Latch Trends

Similar to SRAM

Fewer bits, but large enough to be a problem

Mitigation techniques differ

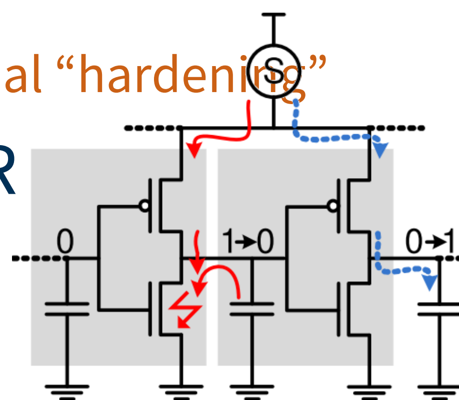
- Latches often synthesized rather than in arrays
- Hardened latches are common, but expensive, solution

MBUs also a problem

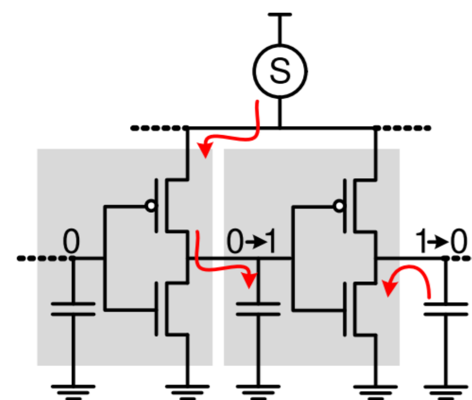
- Accesses mostly narrow

Logic Trends

- Particle strike to off transistor "turns" it on
- Charge translates to current pulse
  - Current flows as long as charge being collected
- Current pulse may not result in an error
  - Logical masking (unaffected by scaling)
  - Temporal masking (related to frequency, expected fixed)
  - Electrical masking
    - Decreases with scale
    - Mitigation possible with minimal "hardening"
- Overall expected 0.5% SER compared to SRAM/latch



(a) Stage 1 : strike and flip



(b) Stage 2 : revert

Other soft faults

Shrinking margins and growing variations →
can lead to timing violations

Our research matches results from IBM research:

- Gain is ~20%
- Almost all this gain possible with reasonable (though dynamic) margins

But what about errors?

- Logic faults/errors very frequently masked
- Error disappears before being latched (timing)
- Error diminishes and disappears because of signal rectification (electrical masking)
- Values lead to logical masking

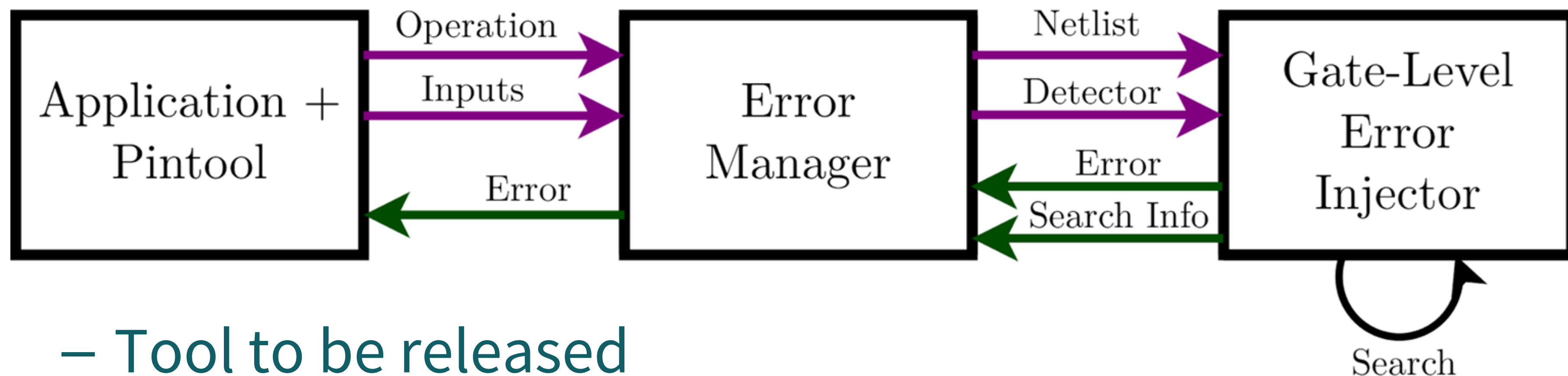
Error modeling?

- Logic simulation
- Multi-mode simulation

THE UNIVERSITY OF TEXAS AT AUSTIN

Quick(ish) way to search the error space

- Multi-mode simulation
- Skip over detectable errors



- Tool to be released
  - Uses only public tools

DRAM faults very problematic

- Dozens of DRAM chips per processor
- Millions per system
- New DRAM fault every couple of hours
- Modules are expensive

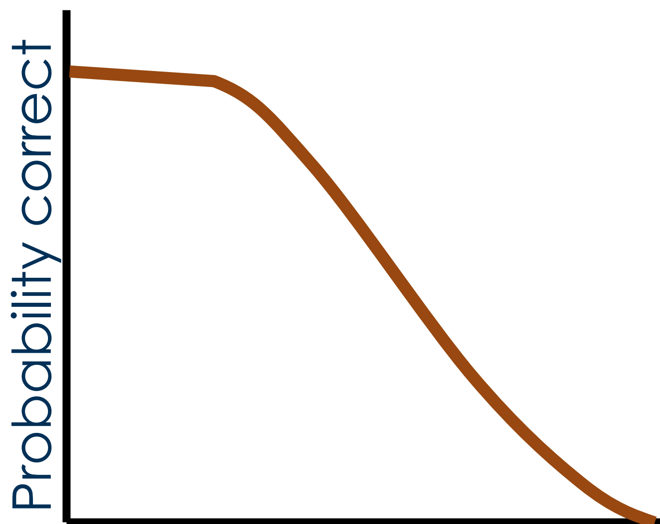
Memory must be **reliable**

- Written data must be recalled “correctly”

Reliability **challenges**

- “Natural” change in cell value
- Induced change in cell value
- Read errors
- Write errors
- Wearout and defects

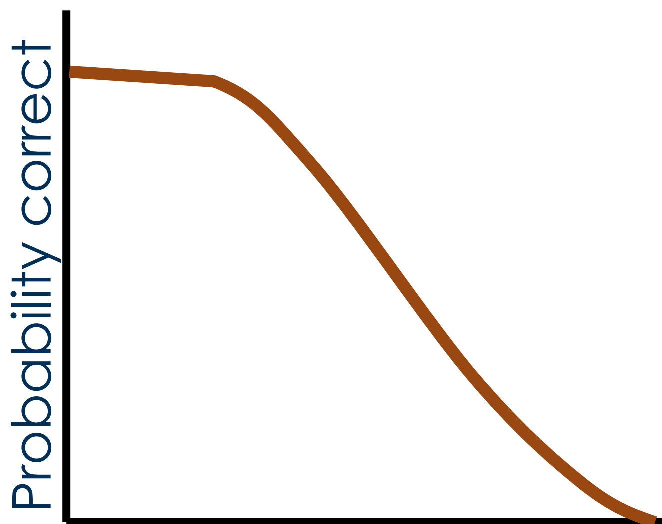
Natural causes of value change

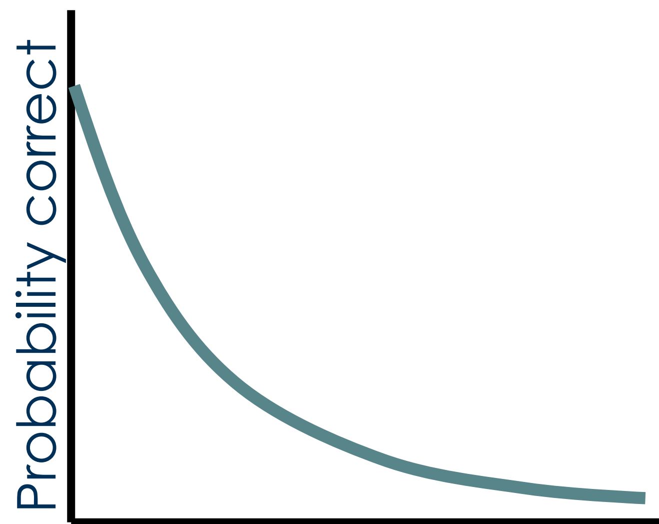


Decay

– Refresh

Natural causes of value change





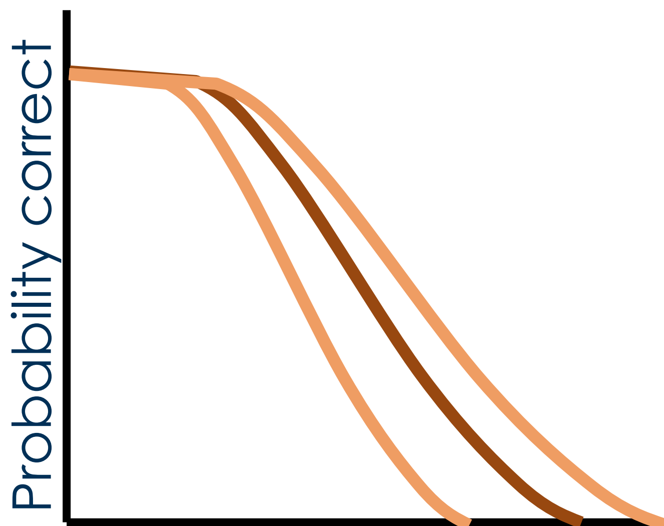
Decay

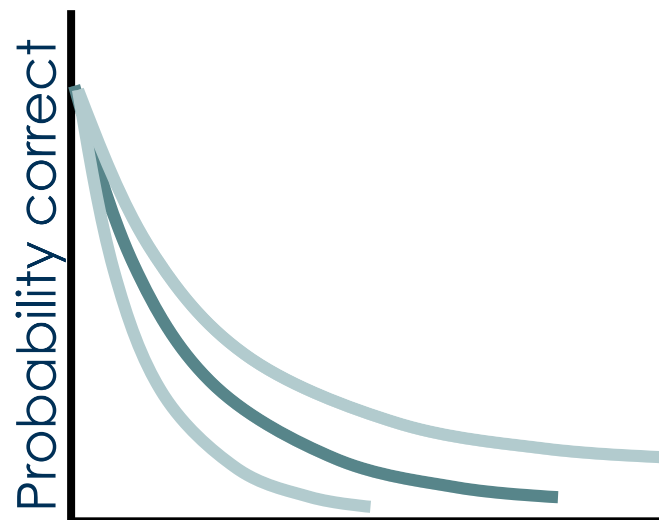
- Refresh

Flip

- ECC + scrub

Natural causes of value change





Time

Time

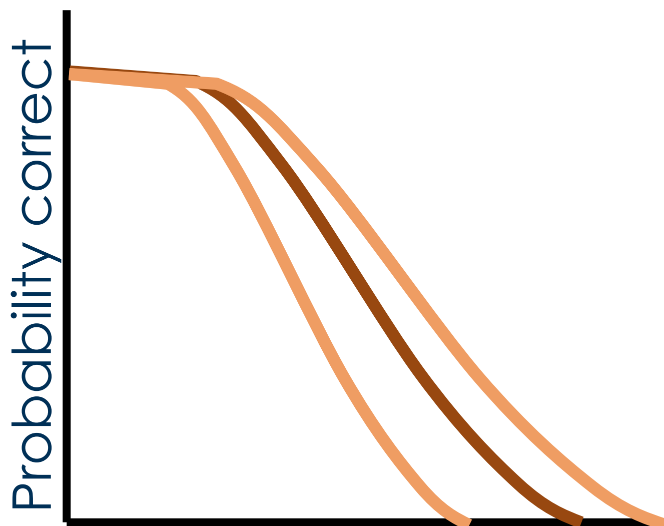
Decay

– Refresh / scrub

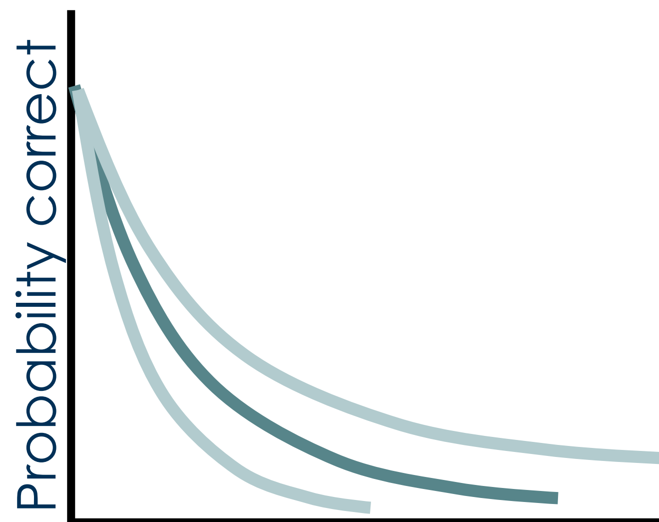
Flip

– ECC + scrub

Natural causes of value change



Time



Time

Decay

– Refresh / scrub

Flip

– ECC + scrub

– **Retention control**

- Less dense
- Writes slower/higher-energy
- Can use different devices

Induced change in value

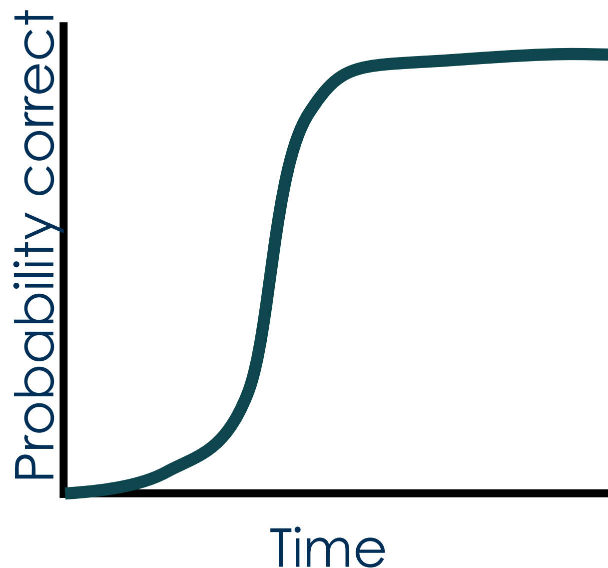
- Writing or reading disturbs nearby cells
- Worse for writes
- Technology and design dependent

Read errors

– Because of small margins for efficiency

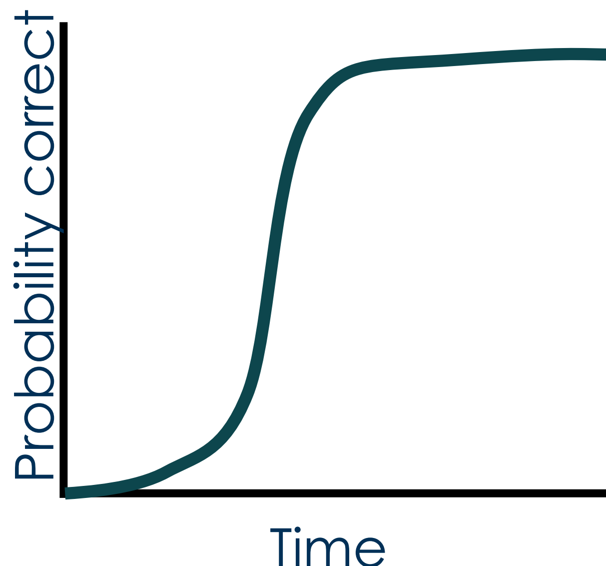
Write errors

- Writing implies changing a state or value
- Stochastic



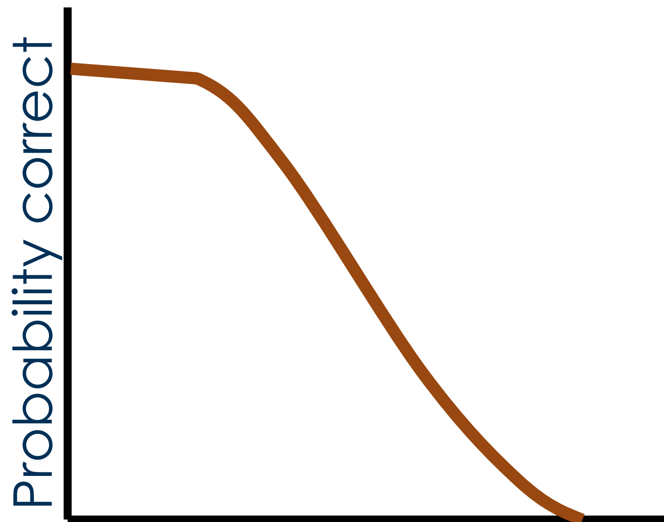
Write errors

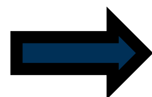
- Writing implies changing a state or value
- Stochastic

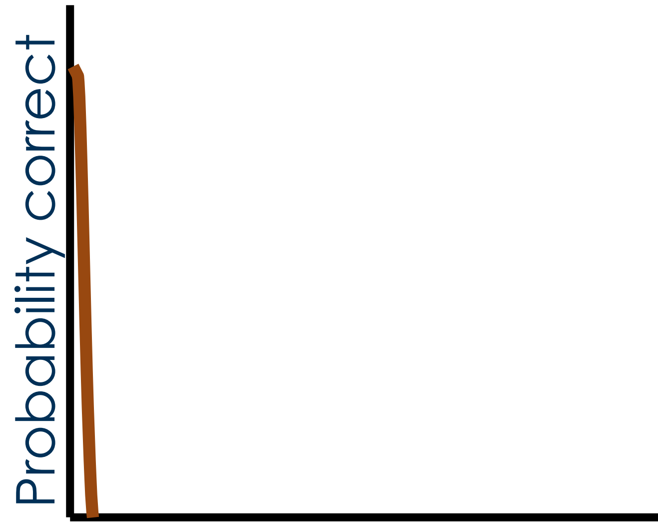


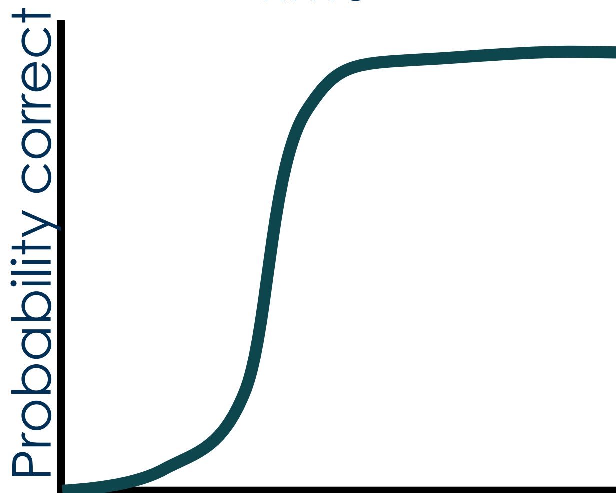
- Waiting inefficient and slow
- Write error control conflicts w/ other errors

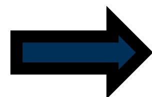
Wearout and defects

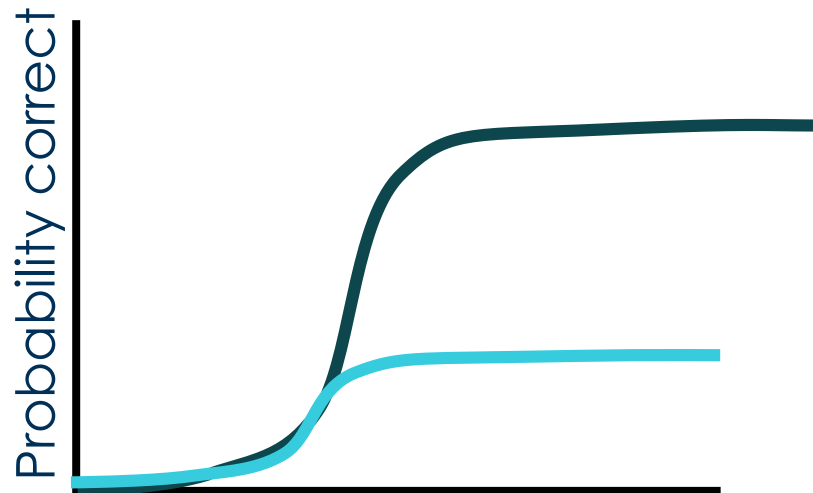












Wearout and defects

- Sparing
- Extra margins
- Retirement
- Compensation (ECC)

- Summary: no “good” memory
  – It’s all about tradeoffs

But, those tradeoffs barely up to us

- Good empirical models for DRAM
- So far, vendors maintaining targets

Interestingly, peripheral circuits very vulnerable

But at the end, good empirical models exist and are currently maintained

Rules of thumb (more later)

- 50/50 soft/hard
- 50/50 single/multi bit

Sub-wordline Driver
Bitline Sense-amp
Cell Array
(e.g. 512 x 512)
Mat

Rank
Chip
Bank A
Bank B
Bank H
M M M
M M M
Switch
Serialize Serialize Serialize Serialize
Chip
Data Pin
Chip
Data Pin
Time
Burst length (8)
Chip width (4-bit)
Channel width (64-bit)

DRAM Errors?

– Highly dependent on ECC scheme
– Probably pretty random

Other hard faults

- Corrosion
- Mechanical stress
- Accidents
- Current stress

Mechanical → power → connectors

- Lots of redundancy can be put in
- Recently, scaling in the chip faster than outside

Network

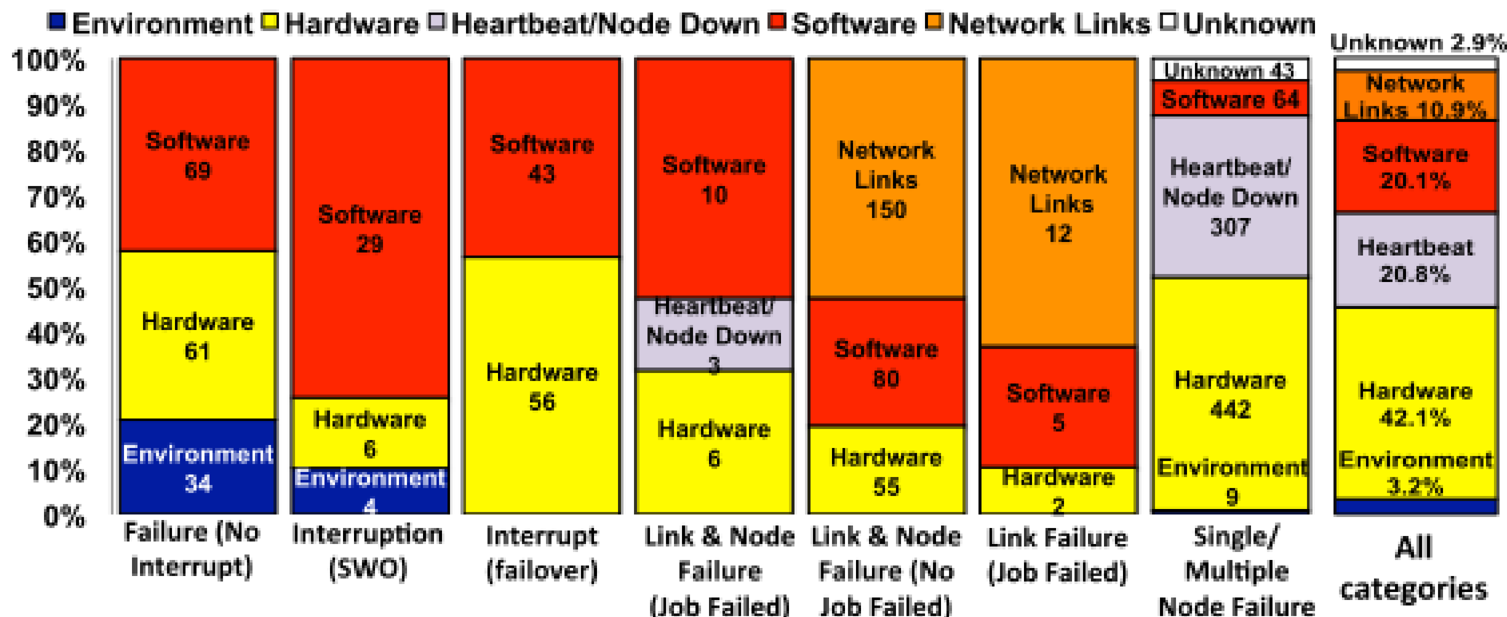
- Processor + memory + links
- Few models available
- Generally very resilient
  - Strong error protection
  - Routing adapts to hard faults

An example: Blue Waters

From Di Marino et al., DSN'14

TABLE III: Failure Statistics. The last row refers to the statistics calculated across all the failure categories.

Failure Category	count	%	MTBF [h]	MTTR [h]	σ_{TBF} [h]	σ_{TTR} [h]
1) Failure (No Interrupt)	164	11%	35.17	13.5	70.8	35.3
2) Interrupt (Failover)	99	6.6%	58	14.7	92	42.2
3) Link & Node Failure (Job Failed)	19	1.3%	297.7	6.1	427.3	5.4
4) Link Failure (No Job Failed)	285	19.1%	19.9	32.7	51.9	91.2
5) Link Failure (Job Failed)	19	1.3%	291.6	16	444	26.7
6) Single/Multiple Node Failure	868	58.2%	6.7	26.7	6.3	72
7) Interruption (system-wide outage)	**39**	**2.62%**	**159.2**	**5.16**	**174.2**	**8.1**
ALL	**1490**	100%	4.2	34.5	13.3	50.5



THE UNIVERSITY OF TEXAS AT AUSTIN

An example: Blue Waters

From Di Marino et al., DSN’14

TABLE III: Failure Statistics. The last row refers to the statistics calculated across all the failure categories.

Failure Category	count	%	MTBF [h]	MTTR [h]	σ_{TBF} [h]	σ_{TTR} [h]
1) Failure (No Interrupt)	164	11%	35.17	13.5	70.8	35.3
2) Interrupt (Failover)	99	6.6%	58	14.7	92	42.2
3) Link & Node Failure (Job Failed)	19	1.3%	297.7	6.1	427.3	5.4
4) Link Failure (No Job Failed)	285	19.1%	19.9	32.7	51.9	91.2
5) Link Failure (Job Failed)	19	1.3%	291.6	16	444	26.7
6) Single/Multiple Node Failure	868	58.2%	6.7	26.7	6.3	72
7) Interruption (system-wide outage)	**39**	**2.62%**	**159.2**	**5.16**	**174.2**	**8.1**
ALL	**1490**	100%	4.2	34.5	13.3	50.5

THE UNIVERSITY OF TEXAS AT AUSTIN

An example: Blue Waters

From Di Marino et al., DSN'14

■ Environment □ Hardware □ Heartbeat/Node Down ■ Software ■ Network Links □ Unknown

100% 90% 80% 70% 60% 50% 40% 30% 20% 10% 0%

Failure (No Interrupt): Software 69, Hardware 61, Environment 34

Interruption (SWO): Software 29, Hardware 6, Environment 4

Interrupt (failover): Software 43, Hardware 56

Link & Node Failure (Job Failed): Software 10, Heartbeat/ Node Down 3, Hardware 6

Link & Node Failure (No Job Failed): Network Links 150, Software 80, Hardware 55

Link Failure (Job Failed): Network Links 12, Software 5, Hardware 2

Single/ Multiple Node Failure: Unknown 43, Software 64, Heartbeat/ Node Down 307, Hardware 442, Environment 9

All categories: Unknown 2.9%, Network Links 10.9%, Software 20.1%, Heartbeat 20.8%, Hardware 42.1%, Environment 3.2%