CCC Visioning Study:

System-Level Cross-Layer Cooperation to Achieve Predictable Systems from Unpredictable Components

www.relxlayer.org

Executive Summary

- Continued scaling → unpredictable components
- Traditional solutions are too expensive
- Cannot solve at any single level
- Opportunities are cross-layer
  - Memory systems offer inspiriation
  - Logic: bigger challenge, but big payoff
- Q: Can we create community consensus on
  - Potential vision
  - Opportunities for cross-layer cooperation
  - Make-a difference research questions that help realize the vision

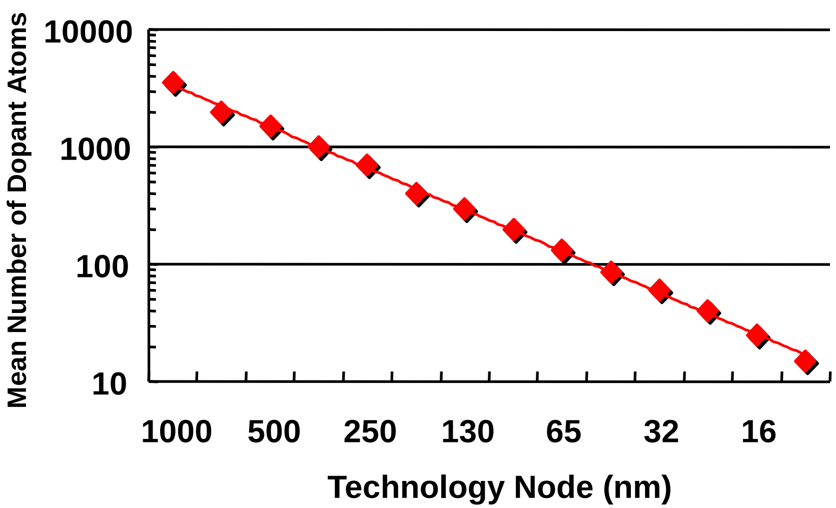
Outline

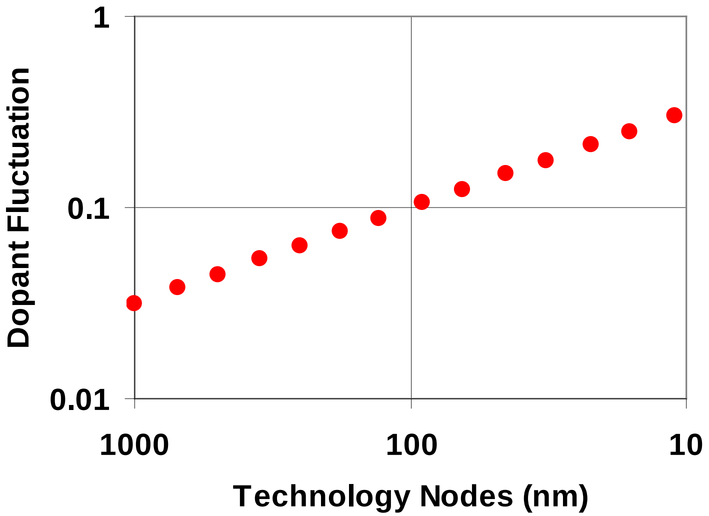
- Scaling Effects
- Weaknesses of traditional approaches
- Single Layer
- Cross-layer Vision
  - Memory Inspiration
  - Principles
  - Example
- Pervasive Demand for Resilience (Carter)
- Early warning from Satellites (Quinn)
- Building community consensus (DeHon)
  - Goal
  - Starting points

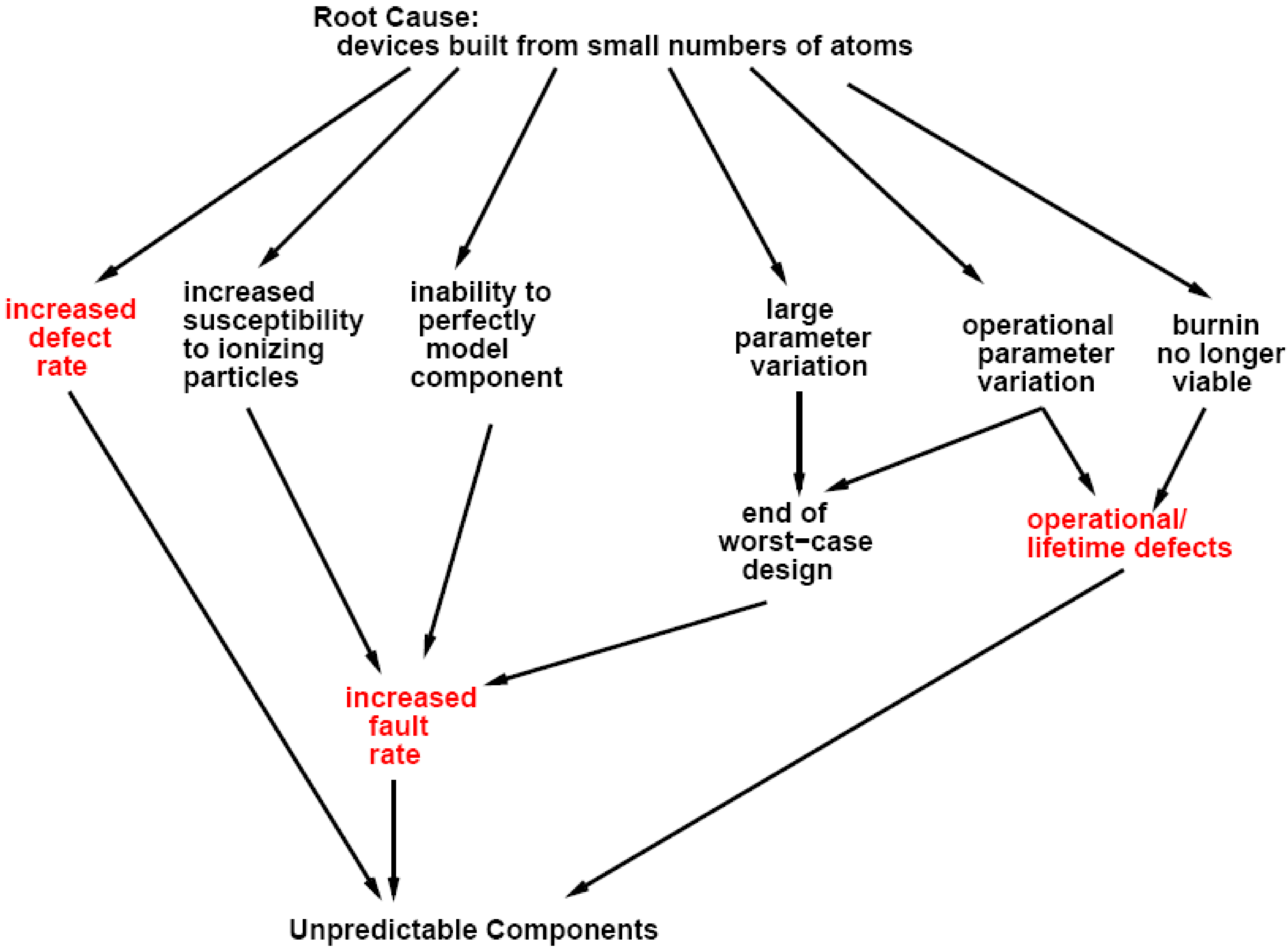
Scaling

Scaling → Unpredictability

- As our devices approach the atomic scale, we must deal with statistical effects governing the placement and behavior of individual atoms and electrons.






Root Cause:
devices built from small numbers of atoms
increased defect rate
increased susceptibility to ionizing particles
inability to perfectly model component
large parameter variation
operational parameter variation
burnin no longer viable
end of worst-case design
operational/ lifetime defects
increased fault rate
Unpredictable Components

Unpredictable Components

1. **Defects:** Manufacturing imperfection
   - Occur before operation; persistent
     - Shorts, breaks, bad contact
2. **Variation:** Continuous case of above
   - Parameters take on variety of values
     - Resistance of wire, on-resistance of switch, threshold voltage of transistor
3. **Transient Faults:**
   - Occur during operation; transient
     - node X value flips: crosstalk, ionizing particles, bad timing, tunneling, thermal noise
4. **Operational/lifetime defects**
   - Parts become bad during operational lifetime
     - Fatigue, electromigration, burnout….
   - …slower
     - NBTI, Hot Carrier Injection

Traditional

Traditional Solutions

- Chip/core-level: sparing, replication, rollback
  - Discard chips with any defects (non-memory)
  - Brute-force, {core,thread}-level replication
    - Integer factors of overhead
    - Not free in power-limited world
  - Expensive, OS-level rollback/reboot
- Worst-Case margining
  - Across billions of devices
  - Expected degradation over lifetime
- Burnin to find weak devices

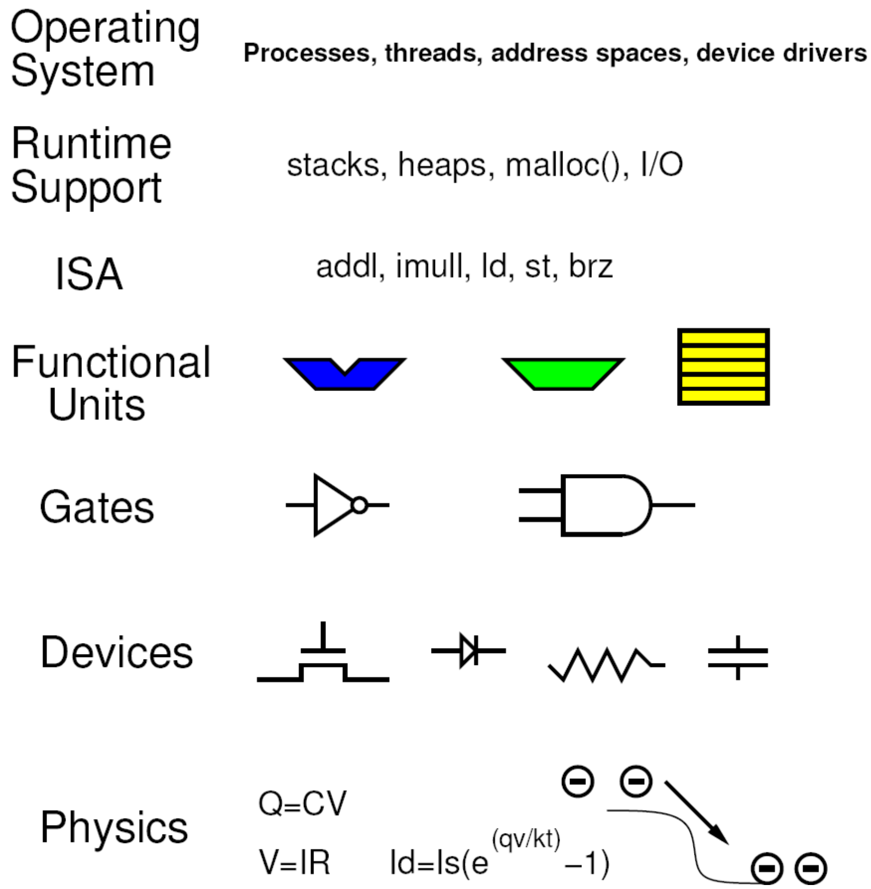
New Reality

- Fabrication will be imperfect
- Devices will wear and fail during operation
- Conservative margining too costly
  - Delay, energy → defeat scaling
- Transient errors impact compute as well as memory
  - Concern for **all** segments
  - Not just Space, Avionics, Financial

Single Layer

Quick, single-level fix?

- Device: small numbers
- Circuit: complete mitigation too expensive
- Architecture: Lacks context information
- OS/Application-level: too late
  - Upset rates too high
  - Cannot run enough logic, long enough



Cross-Layer Vision

Memory Systems

- Deal with defective fabrication
  - row/column sparing
- Accommodate transient upsets
  - Error-Correcting Codes
  - Scrubbing

Memory: Cross-Layer Optimization

- Device
  - Hardening
- Circuit
  - Differential reliability
- Architecture
  - ECC Protection (e.g. SECDEC 12.5% overhead)
- OS
  - Periodic Scrubbing
  - Map out bad blocks
- More expensive if tried to solve at **any** single level

Principles

1. Prepared for repair
2. Errors filtered at multiple levels
3. Multi-level tradeoffs
   – (generalization of hardware/software)
4. Strategic redundancy
5. Differential reliability
6. Scalable/adaptive solutions

Multi-level Vision

- A traditional, ECC-protected memory
  - provides the **reliability** of large feature sizes
  - with the **density** of small memory cells
- Multi-level computational designs
  - provide the **reliability** of large-feature and large-energy devices
  - with the **density** and **energy** consumption of small-feature, low-energy devices

Computational Application

1. Prepared for repair
   - Regular, fine-grained architectures: e.g. FPGA
   - Computational model to abstract defect details
2. Errors filtered at multiple levels
   - Circuit and architecture invariants
   - Application and OS self-checks
3. Multi-level tradeoffs
   - Right-level of filtering, handling at each level
4. Strategic redundancy
   - Invariants, end-to-end consistency, application-specific
5. Differential reliability
   - Management and repair circuitry built from coarser-feature logic
6. Scalable/adaptive solutions
   - Tune to upset rate, criticality of computation
   - Tune level of redundancy in-system, throughout lifetime

Outline

- Scaling Effects
- Weaknesses of traditional approaches
- Single Layer
- Cross-layer Vision
  - Memory Inspiration
  - Principles
  - Example
- **Rethinking Resilience** (Carter)
- Early warning from Satellites (Quinn)
- Building community consensus (DeHon)
  - Goal
  - Starting points

Outline

- Scaling Effects
- Weaknesses of traditional approaches
- Single Layer
- Cross-layer Vision
  - Memory Inspiration
  - Principles
  - Example
- Rethinking Resilience (Carter)
- **Early warning from Satellites** (Quinn)
- Building community consensus (DeHon)
  - Goal
  - Starting points

Outline

- Scaling Effects
- Weaknesses of traditional approaches
- Single Layer
- Cross-layer Vision
  - Memory Inspiration
  - Principles
  - Example
- Rethinking Resilience (Carter)
- Early warning from Satellites (Quinn)
- **Building community consensus** (DeHon)
  - Goal
  - Starting points

Goal Visioning Study

- Build consensus around compelling vision
- Articulate vision
- Build community consensus on key research problems

Strawman Starting Point

- Assembled six questions as initial start
  – Based around 6 principles
  – Need clarification and crystallization
  – Organization for today’s breakout groups
- Miss key areas/opportunities?
  – Open-microphone later today
  – Discuss lunch/dinner
  – Slot for presentations tomorrow

Q1: Repair

- How do we best accommodate repair?
  – Granularity
  – Division of labor across layers
  – Visible interfaces

Q2: Error Filtering

- What is the right level of filtering at each level of the hierarchy?
  - Characterize level provided?
  - Measure/assess/validate solution?
  - Tune?

Q3: Multi-Layer Cooperation

- How do we organize, manage, and analyze layering for cooperative fault mitigation?
  – New layer contracts?
  – What useful to reflect up?
  – What lower-level controls should be exposed?
  – What higher levels pass down?
  – Engineering/analyze adaptation/repair control loops?

Q4: Lightweight Checking

- Can we establish a useful theory and collection of design patterns for lightweight checking?
  - Class of computations amenable?
  - Express checks in computations
    - Programming language
  - Optimize checks

Q5: Differential Reliability

- What would a theory and framework for expressing and reasoning about differential reliability look like?
  - Express allowable noise?
  - Reflecting/exposing noise to application-level?
  - Express or analyze reliability needs of pieces of a computation?

Q6: Adaptation

- Can a scalable theory and architectures that will allow adaptation to various upset rates and system reliability targets be developed?
  - Energy-delay-area-reliability vs. upset rate
    - Like rate-distortion curves
  - Efficiency of arch. tuned across varying range of upset rates, reliability targets

Immediate Questions

- 10:30 Immediate/short questions
- 10:45 Break
- 11:00 Full group discussion
- 12:00 Lunch
- 1:30 Breakout 1

Breakout

Q1	121	Repair	Carter	Adve
Q2	122	Filter	Nassif	Carter
Q3	204	Multilevel	Adve	Quinn
Q4	324	Strategic	Mitra	DeHon
Q5	201	Diffrel	DeHon	Huang
Q6	213	Adapt	Quinn	Savage