NanoComputing Architectures

Seth Copen Goldstein
Carnegie Mellon University

CMU
October 17, 2002

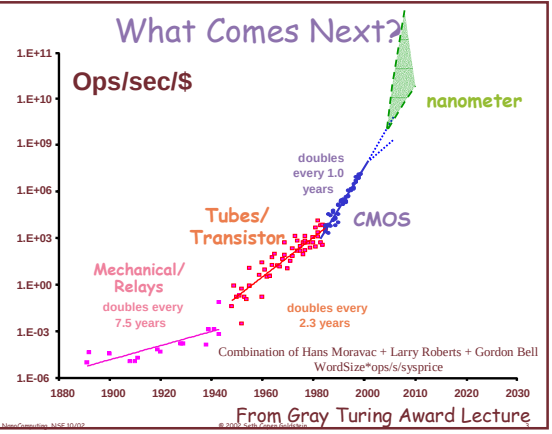
 © 2002 Seth Copen Goldstein

Outline

- Manufacturing Paradigm Shift
  - Dealing with the atomic scale
- Research Directions
  - Defect/Fault Tolerance
  - An Example architecture
  - Abstractions
    - Devices
    - Circuits
    - Architectures
- Summary

 © 2002 Seth Copen Goldstein

What Comes Next?



From Gray Turing Award Lecture

 © 2002 Seth Copen Goldstein

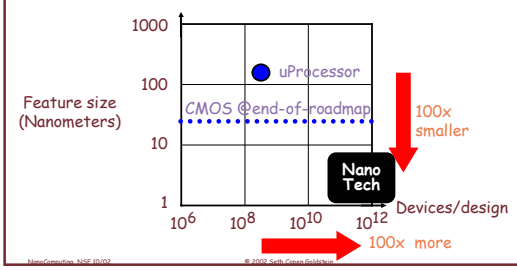
Technology Shifts

- Size of Devices
  ⇒ Inches to Microns to Nanometers
- Type of Interconnect
  ⇒ Rods to Lithowires to Nanowires
- Method of Fabrication
  ⇒ Hammers to Light to Self-Assembly
- Largest Sustainable System
  ⇒ 10^1 to 10^8 to 10^{12}
- Reliability
  ⇒ Bad to Excellent to Unknown

 © 2002 Seth Copen Goldstein

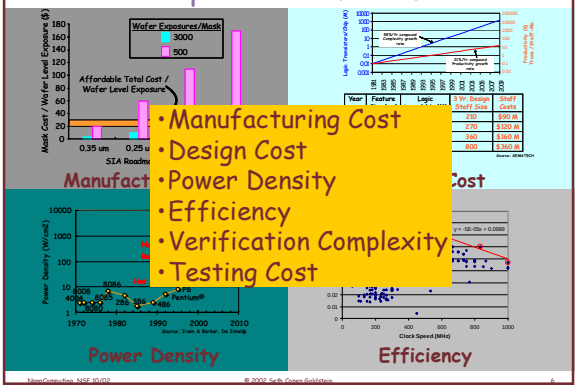
On the cusp of Major Technology Change

- Devices will be *very* small & numerous
- Scale is new: 100x smaller, 100x more



 © 2002 Seth Copen Goldstein

Independent of Shift?



 © 2002 Seth Copen Goldstein

© 2002 Seth Copen Goldstein

The Atomic Scale

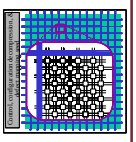
- Very small devices/wires
  - Increased variability
  - Increased single-event upsets
  - New behaviors/mechanisms
  - No predetermined arbitrary patterns
  - Increased defect densities
- Lots of devices/wires
  - Scalable .* required

CMOS Too …

- Precision is expensive
- Current Abstractions have a major impact on cost
  - Requires process engineers to deliver robust devices
  - Requires perfect devices
- CMOS is becoming just another nanoscale component
- Process rules and mask costs are already eliminating arbitrary patterns

Computing Crystals

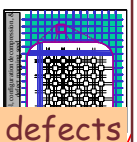
- A reasonable medium-term position
  - No end-to-end connections
  - Maybe multi-terminal devices
  - No complex predetermined patterns
  - Quasi-regular & non-deterministic
  - Hierarchical assembly



Assembly ⇒ Computing Crystals

Computing Crystals

- A reasonable medium-term position
  - No end-to-end connections
  - Maybe multi-terminal devices
  - No complex predetermined patterns
  - Quasi-regular & non-deterministic
  - Hierarchical assembly



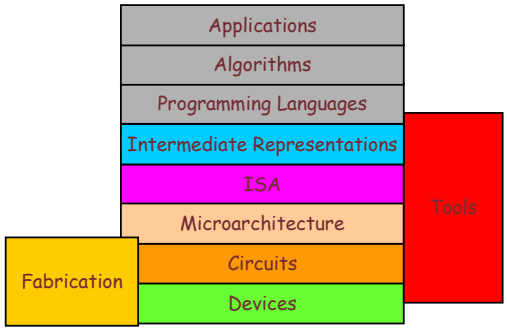
With defects

Assembly ⇒ Computing Crystals

How to Make Forward Progress

- New Abstractions
  - Defect/fault tolerance
  - Devices
  - Circuits
  - Architectures
- Close the loop between designers and device engineers

Abstraction Layers



Defect/Fault Tolerance

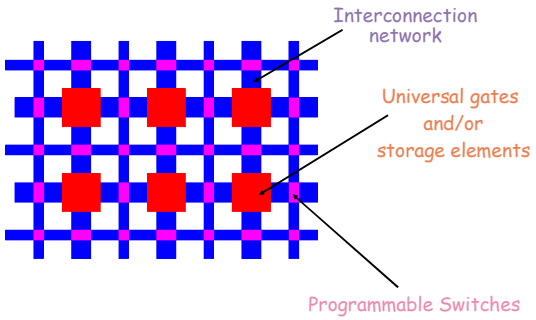
- Vast body of knowledge, but nanocomputing is different:
  - Massive integration and diverse function
  - High density of defects and faults
  - Constrained fabrication
- Nanoscale engineering *requires* that we build systems that work from devices that don't – this is fundamental due to atomic scale
- Fault and Defect tolerance is key to reducing cost and supporting novel devices

Use redundancy and computation to compensate for lack of robustness and precision

Hierarchical Solution

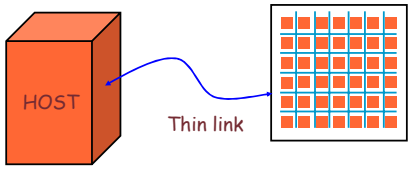
- Must attack this at every level
  - Device engineering 1
  - Fabrication 2
  - Circuits 3
  - Testing methods 6
  - Architectures 8
  - Algorithms infinite?

Island-Style FPGA

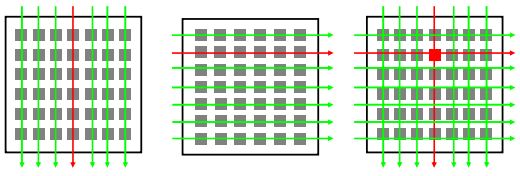


Dealing With Defects

- Need to discover the characteristics of the individual components, but
- Can't selectively stimulate or probe the components
- Download test machines



Built-In Self-Test

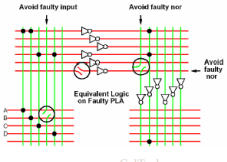


Configure testers vertically | Configure testers horizontally | Identify fault!

Configure the device to test itself!

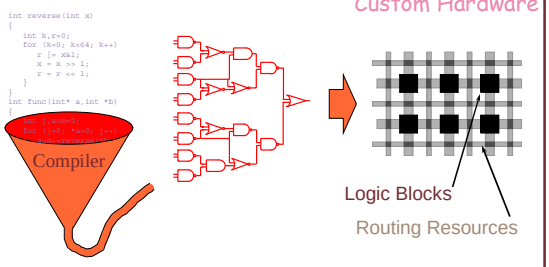
Defect Tolerant Architecture

- Extra devices, wires, and post-processing to route around defective elements
- Multi-layered approach required
- Radical architectures to exploit
  - Randomness
  - Reconfigurability
  - Plentiful resources
- New testing approaches



Reconfigurable Computing

General-Purpose | Custom Hardware | General-Purpose Custom Hardware

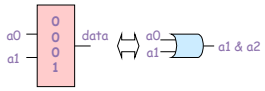


Advantages of Reconfigurable

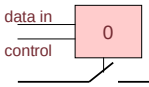
- Flexibility of a processor
- Performance of custom hardware
- Defect Tolerant

(Molecular give back area-delay lost in CMOS)

Heart of an FPGA



Universal gate = RAM

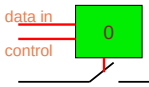


Switch controlled by a 1-bit RAM cell

Heart of an FPGA

a0
a1
Unive

The cost of the FPGA:
Increased Area-Delay Product
RAM cell
- Wires to program RAM cell



Switch controlled by a 1-bit RAM cell

Key Component: Reconfigurable Switch

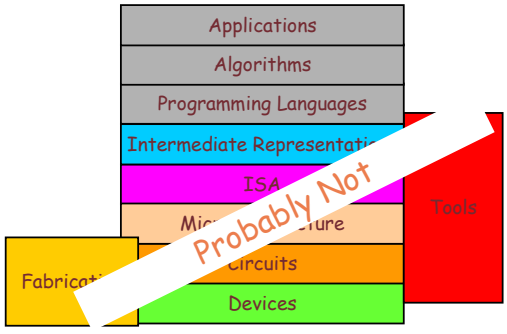
[2]Rotaxane Molecular Switch

Complexation: Crown-Ammonium
Conditions: Neutral
Colour: Colourless

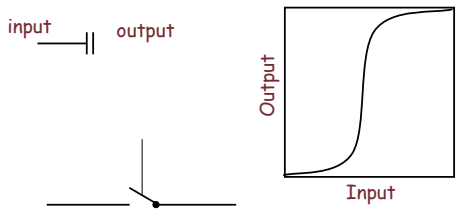
Reconfigurable Molecular Switch
- Holds its own configuration state
- Can be programmed with the signal wires

http://www.chem.ucla.edu/dept/Faculty/stoddart/research/mv.htm

Abstraction Layers Still OK?

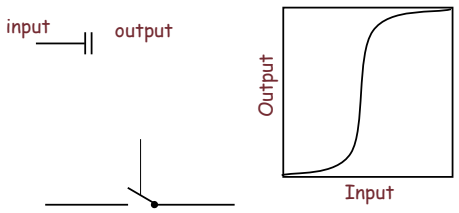


Transistor Abstractions



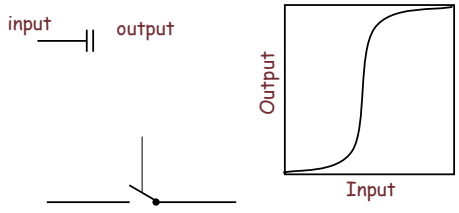
Old model:
- One device: restores, isolates, …
- All turn on at same point
- All use same power

Transistor Abstractions



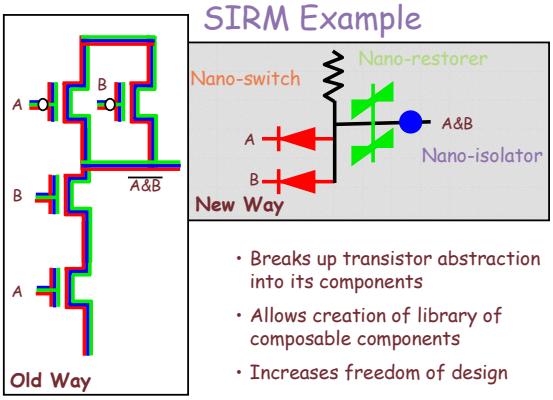
Current model:
- Many different kinds of Ts: ram, logic
- different turn-ons, power usage, speed

Transistor Abstractions



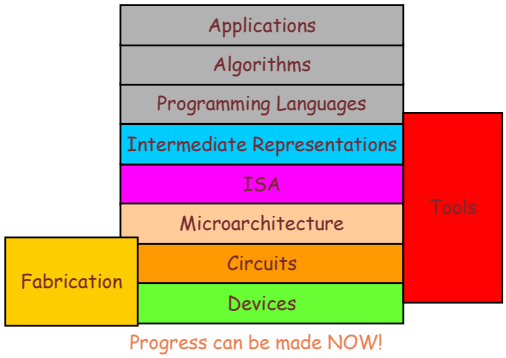
New model: SirM
- Some devices are switches
- Some devices are isolators
- Some devices are restorers

SIRM Example



- Breaks up transistor abstraction into its components
- Allows creation of library of composable components
- Increases freedom of design

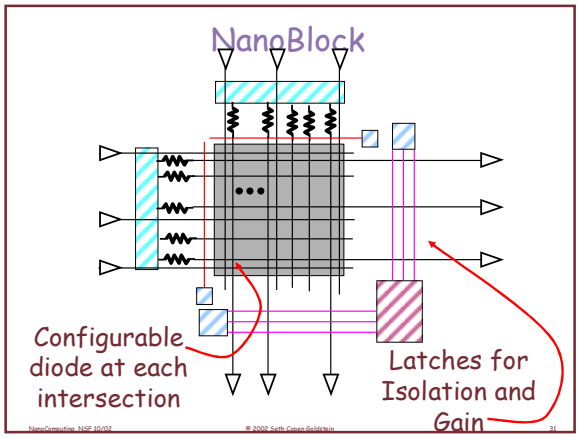
New Layer

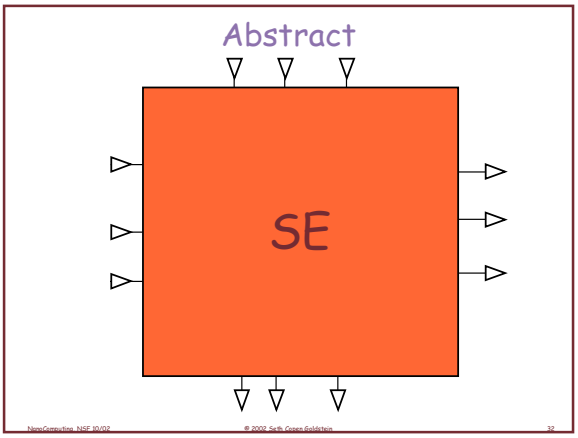


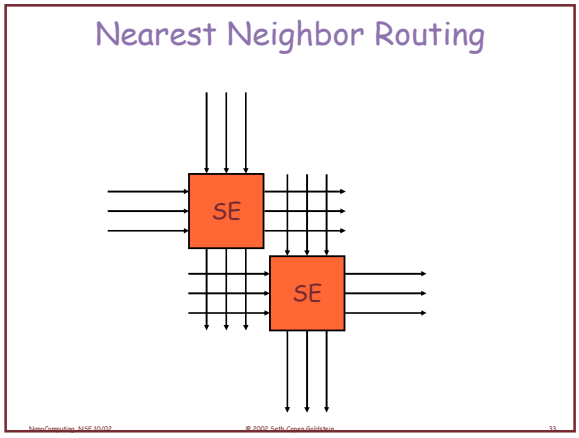
Progress can be made NOW!

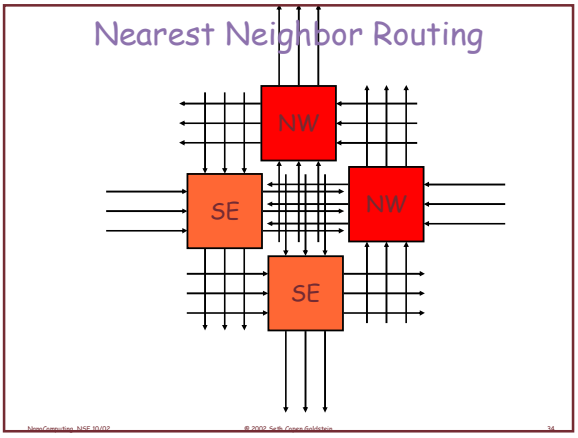
Circuit Abstractions

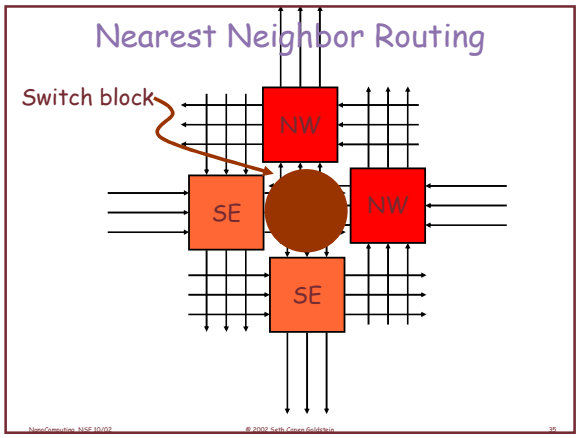
- Compute with "1" and "0" — Multivalued logic? Analog?
- Logical operations, AND, OR, etc.
- Connect up devices as nee… — Look-up tables?
  Point-to-poin… — Meshes? Switch-blocks
- Synchronize as necessary
  - Combinational logic
  - register… — Asynchronous? Forced to latch?
- Unlimited fan-out
- wires are "perfect" (maybe RC) — What is a wire?

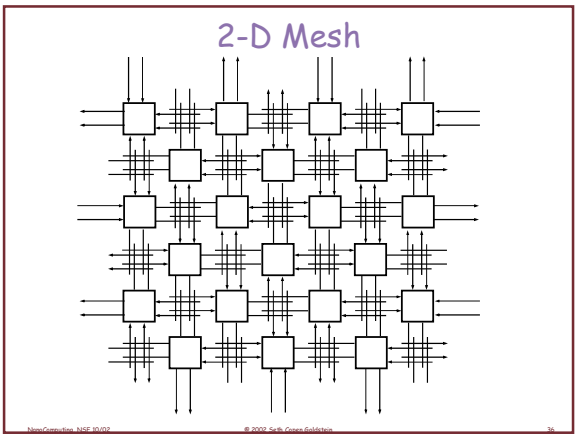

NanoBlock
Configurable diode at each intersection
Latches for Isolation and Gain


Abstract
SE

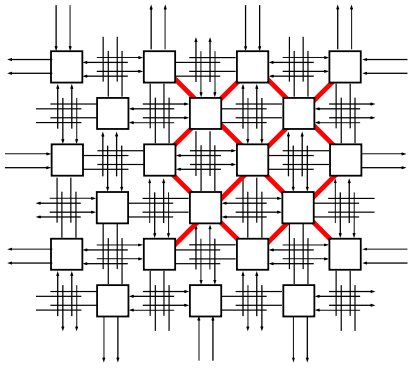

Nearest Neighbor Routing
SE
SE


Nearest Neighbor Routing
NW
SE
NW
SE

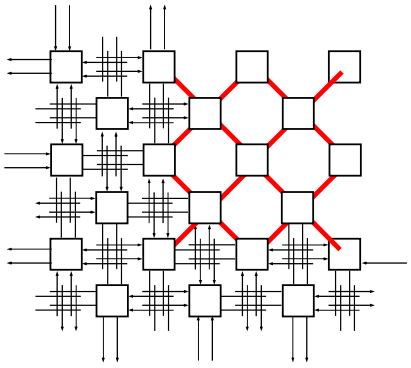

Nearest Neighbor Routing
Switch block
NW
SE
NW
SE


2-D Mesh

2-D Mesh

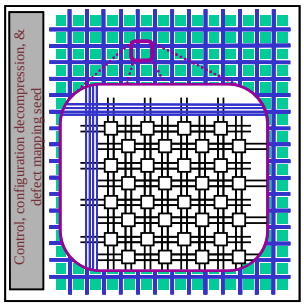


2-D Mesh



The NanoFabric

- Nanoscale layer put **deterministically** on top of CMOS
- Highly regular
- ~10^8 long lines
- ~10^6 clusters
  - Cluster has 128 blocks



NanoFabric Attributes

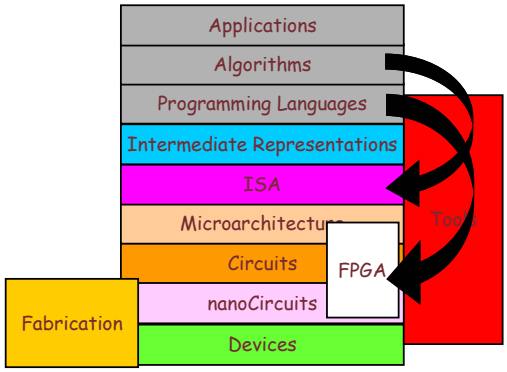
- Hierarchical fabrication
  1. Manufacture devices
  2. Non-deterministically align wires
  3. Self-assemble monolayers onto wires
  4. Create meshes
  5. Deterministically align w/CMOS
- Reconfigurable
- Defect tolerant

How do we use it?

ISA has to go?

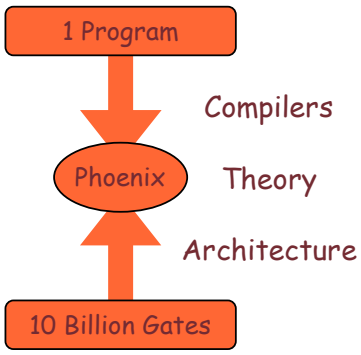
- Current ISA hides to much
  - Good for
    - forward compatibility
    - human oriented assembly
    - ad hock additions
  - Bad for
    - removing constraints
    - exploiting compiler
    - verification
- What can replace ISA?

Breaking (And Adding) Abstractions

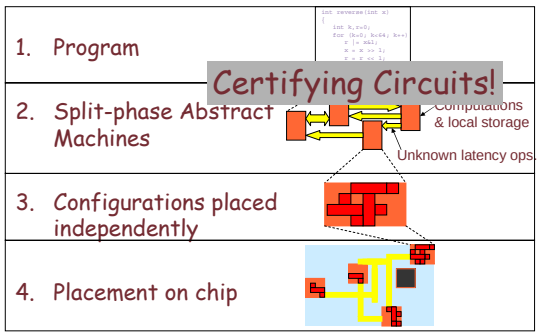


© 2002 Seth Copen Goldstein

Spanning 10-orders of Magnitude



Circuits From Compilers

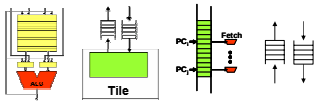


Direct Mapping: App's → Gates

- Likely to be difficult because tool must
  - Build in error recovery
  - Synthesize, place, and route 10^{12} devices
  - Virtualize communication resources
  - Allow movement of computation away from failing components (redo p&r)
- Compiler doesn't see everything
  - separately compiled applications
  - Operating System interactions

New Abstraction Layer: Nano-Core

- Provides parameterized abstractions for
  - datapaths
  - memory
  - control
  - communication



- Compiler/OS target Nano-Core
  Can build any computing device – on the fly
- Abstractions hide defects and faults
  - at cost of additional area, power, delay
  - This allows forward progress now
  - Can pierce this layer over time

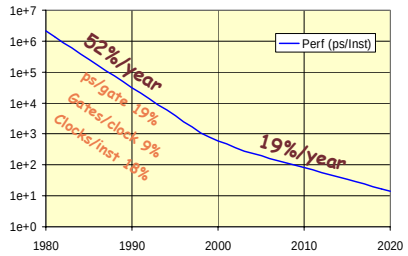
Conclusions

- Investment in nanodevices and materials is bearing fruit
- A new manufacturing approach is necessary to fabricate large-scale circuits (>10^{12} devices/cm^2) from nano-scale components.
  - Derivatives of "self-assembly" have the best chance to succeed.
  - All known approaches result in statistically imperfect macro-scale circuit structures.
- Dealing with imperfections, defects, and outright faults is a new requirement
  - Hardware and software approaches
  - Redundancy and computation compensate for defects/faults
- Abstraction is the solution
  - Enables simultaneous progress at all levels
  - Dependent only on size and scale

Conclusions - 2

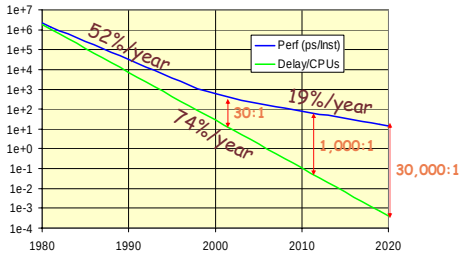
- Abstractions need to be:
  tool friendly not human friendly
- Focus moves from
  - ISA → IR
  - Von Neumann → Parallel fine-grained fabric
- Harness reconfigurable computing!

Conventional Processors No Longer Scale Performance by 50% each year



Bill Dally

Future potential of novel architecture is large (1000 vs 30)



Bill Dally