The Impact of the Nanoscale on Computing Systems

Seth Copen Goldstein
seth@cs.cmu.edu
Carnegie Mellon University

What Comes Next?
Combination of Hans Moravac + Larry Roberts + Gordon Bell

Word Size * ops/s/sysprice

From Gray Turing Award Lecture

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

Independent of Technology

As we scale down:
- Devices become
  - more variable
  - more faulty (defects & faults)
  - numerous
- Fabrication becomes
  - More expensive
  - More constrained
- Design becomes
  - More complicated
  - More expensive
- Market pressures remain

Drain   Gate   Source

1 dopant atom

25 nm

IBM NanoCMOS

MIT HP
Independent of Technology

As we scale down:
- Devices become more variable
- More faulty (defects & fault)
- Numerous
- Fabrication becomes more expensive
- More constrained
- Design becomes more complicated
- More expensive
- Market pressures remain

Requires:
- Defect tolerance
- Higher level specification
- Universal substrate
- Asynchronous circuits
- Spatial computing

Size Matters

Challenges arise from:
- Small size: changes in physical process
- Many devices: increased complexity

Opportunities too!
### Manufacturing Paradigm Shift Required

**Today**
- Reliable Systems from reliable components
  - Reliable systems from unreliable components
- Functionality invested at time of manufacture
  - Functionality modified after manufacture
  - New manufacturing: Bottom-up assembly
- Behavior remains same as features scales down
  - Expect increased variability
  - Changes in functionality
  - Restrictions on connectivity

**Future**
- Reliable Systems from reliable components
- Functionality invested at time of manufacture
- New manufacturing: Bottom-up assembly
- Behavior remains same as features scales down

### Limited Patterns

- **Top-Down**
  - Sub wavelength lithography
    - OPC, RET, CPM, ...
  - Nanoimprint lithography
  - DPN
- **Bottom-Up**
  - Self-assembly

### Balance

- Nanoscale makes things harder
- Nanoscale makes things easier
- Challenge: Use devices to
  - Ease restrictions
  - Reduce complexity
  - Reduce power
- How: change abstractions and tools
The Clock

- Design for worst case arrival
  - Parametric variation
  - Timing closure
  - Power
- Asynchronous circuits
  - No global controllers
  - No global clock
  - No timing closure
  - Tolerant of parametric variation
- Use more devices to
  - Reduce power
  - Support device scaling
  - Support defect tolerance

Reconfigurable Computing

- General-Purpose
- Custom Hardware
  - Compiler
  - Logic Blocks
  - Routing Resources

Reconfigurable Rationale

- Reconfigurable Architectures address roadblocks
  - Yield with defect tolerance
  - Cost single substrate eliminates NRE
  - Manufacturability Crystaline architecture reduces fab complexity
  - Power Power ∝ Area(3-σ)/σ where σ is algorithm dependent; typically 2 < σ < 3.
- However, must change computing approach

Reconfigurability & DFT

- FPGA computing fabric
  - Regular
  - Periodic
  - Fine-grained
  - Homogenous
- programs ⇒ circuits
- Aides defect tolerance

Aside:
Molecular Scale Electronics increases fabric density
Reconfigurable Computing

General-Purpose Custom Hardware

Compiler

Logic Blocks
Routing Resources

General-Purpose Custom Hardware

Reconfigurable Computing

Design Pressure
Mean time to chip: 46 weeks

User Requirements
Spec 
HW Design 
Verification 
Masks

Design Crisis: By 2010, 1000 Man-years/chip

Mask costs soar

Spec written in C: used to verify HW and check user reqts

Other issues:
- Yield
- Parametric variation
- Power

Change in Spec, or bug in chip → must respin chip

Spanning 10-orders of Magnitude

1 Program

Compilers
Phoenix
Theory
Architecture

10 Billion Gates

Performance: Ops/Clk * Clks/Sec

Specint2000

Horowitz
ISA has to go?

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

Breaking Abstractions

Spatial Computing

- Use available devices to:
  - Map circuits in space, not time
  - Reduce virtualization
  - Decrease clock frequency
- Eliminate
  - Global control
  - Global structures
- Use different devices/architecture
  - Hybrid approach: CMOS+MSE
  - Hybrid approach: match task to devices
    - Stochastic approaches
Spatial Computing: $C \rightarrow$ hardware

- Compile ALL of ANSI-C
  - No pragmas or hardware directives needed
- Uses new intermediate representation
  - Pegasus has precise semantics
  - Correspondence between pegasus and linear logic
- Produces asynch circuits

Program
\[ x = a \& 7; \]
\[ y = x >> 2; \]

IR
\[ \begin{align*}
  a & 7 \\
  \& & >> 2 \\
\end{align*} \]

Circuits
\[ \begin{align*}
  & \text{Dedicated hardware: CASH circuits} \\
  & \text{Asynchronous \( \mu \)P: FPGAs} \\
  & \text{General-purpose DSP: Microprocessors} \\
\end{align*} \]

Automatic Verification

- Support Three levels of verification
  - Model Checking
    - Check for attributes of $C$ program
    - Verify specification
  - Translation Verification
    - Prove translation is equivalent to original $C$ code
  - Self-Certification
    - Allow safe and secure downloading of hardware
- Linear-logic $\Leftrightarrow$ IR correspondence
  - Gives rise to typed-hardware
  - Eliminates MANY design bugs early
  - Prove useful runtime properties

Energy Efficiency [Operations/nJ]

General-purpose DSP
Dedicated hardware
CASH circuits
FPGA
Microprocessors

Using Area to Reduce Power

- Power in CMOS has four components
  - Dynamic switching
  - Short-circuit
  - Subthreshold leakage
  - Gate leakage
- Early VLSI result: $AT^\sigma = \text{constant}$
- Thus, $T^\sigma \propto A^{-1}$, or $T \propto A^{-1/\sigma}$
- Since $T \propto F^{-1}$, we get: $F \propto A^{-1/\sigma}$
- Thesis: Use more devices ($A$) to reduce $F$ and in turn reduce $P$.

Dynamic Switching Power

- $P_{\text{dyn}} = \alpha CV^2F$
- $F \propto V$ [Chen97,Flynn99]
  - If $C$ per node remains the same
  - If threshold voltage remains fixed
- $C \propto A$
- Using $F \propto A^{-1/\sigma}$ $\Rightarrow$ $P_{\text{dyn}} \propto A^{A^{-3/\sigma}}$
- If $\sigma \leq 3$, power can be reduced by using more area!

Joint work with Paul Beckett
What is $\sigma$

- $\sigma$ is a measure of a circuit's inherent sequentialness
- Lower values of $\sigma$ mean a circuit is more parallelizable
- Many important circuits have $\sigma \leq 2$
  - FFT/DFT
  - Adders
  - Multipliers
  - Sorting
  - …

Subthreshold Power

- $I_{\text{sub}} = I_{\text{SO}} \left(1 - e^{-\frac{-V_{DS}}{V_{T_S}}} \right)$
- Worst case:
  - $V_{GS}=0$, $V_{DS}=V_{DD}$
- $I_{\text{OFF}} \propto e^{-40V_{TH}}$
- Change how we set $V_{TH}$:
  - $V_{TH} = a - bV_{DD}$
- $I_{\text{OFF}} \propto e^{40bV_{DD}}$
- $P_{\text{sub}} \propto A(\sigma - 3)/\sigma$

Power/Area Tradeoff

- All components of power can be reduced by using more transistors such that:
  $$P \propto A(\sigma-k)/\sigma, \ k \geq 3$$
- Constraints/Comments:
  - $V_{DD}$ must scale with $F$
  - Must set $V_{TH}$ properly
  - This improves energy-delay!
  - Algorithm must be parallel enough, i.e.,
    $$\sigma_{\text{alg}} < 3$$

Summary

- Nanoscale imposes new constraints
  - power, cost, defects, regularity, …
  - regular, homogenous architectures
- It's not about technology, but size
- Reconfigurable Computing is inevitable
- Harness scaling
  - Use the massive numbers of devices available at the nanoscale
- Tools are key
  - Make abstractions tool friendly
  - Get human out of the loop
Summary

- Nanoscale imposes new constraints
  - power, cost, defects, regularity, ...
  - regular, homogenous architectures
- It's not about technology, but size
- Reconfigurable computing is inevitable
- Harness scaling
  - Use the massive numbers of devices available at the nanoscale
- Tools are key
  - Make abstractions tool friendly
  - Get human out of the loop

Reconfigurable fabrics
- Reduce manufacturing costs
- Improve time-to-market
- Improve defect tolerance

Asynchronous circuits
- Reduce timing issues
- Aid defect tolerance
- Reduce power

Very high-level synthesis
- Reduce design time
- Reduce verification time

Spatial Computing
- Reduce power
- Reduce wire delay problem

Continuing the Trend

Tradeoff complexity (and precision) at manufacturing time for complexity at compilation time.

Complex fixed chip + Program
Regular, tileable structures + Configuration

What is Nanotechnology?

- Fundamental Misunderstanding?
  Nanotechnology ≈ 10^{-9} meters
- Maybe true for nanomaterials?
- My personal view:
  Nanotechnology ≈ 10^9 components
CS & Nano

- Computer science: The science of controlling complexity through abstraction.
- Nanotechnology: Technology for constructing and manipulating billions of nanoscale items.
- For example, Manage:
  - Randomness/regularity of bottom-up assembly
  - Build in defect-tolerance
  - Complexity of manufacturing

Challenges

Nanoscale regime ⇒ Billions of components

- Use CS to control processes; eliminating need for precise molecular manufacturing yet yielding interesting and valuable products
- CS contributions to nanotechnology:
  - Concurrency
  - Interfaces
  - Hierarchical assembly
  - Distributed control
  - ...

Aka: How do we deal with complexity?