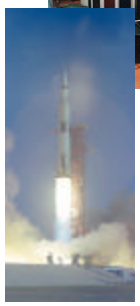
Reconfiguring The Future

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

IEE FPGA Developers Forum
10/22/03

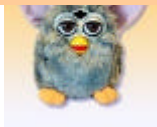
Moore's Law

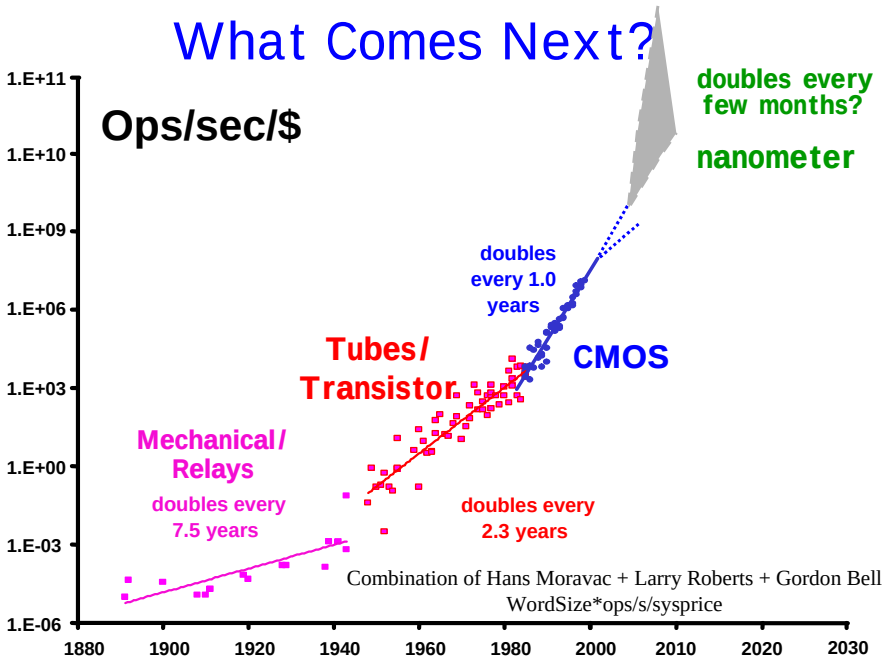

Moore's Law

Nanoscale components ⇒ Computers that are

- *Small* enough to fit inside cells
- *Cheap* enough to be disposable
- *Dense* enough to embed a supercomputer
- *Smart* enough to assemble themselves



What Comes Next?



Combination of Hans Moravac + Larry Roberts + Gordon Bell
WordSize*ops/s/sysprice

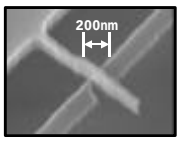
From Gray Turing Award Lecture

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

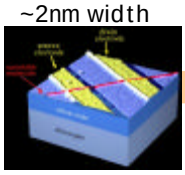
Size Matters

15nm FINFET, end-of-roamap approx CMOS size

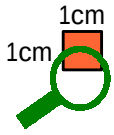


UC. Berkeley, MIT

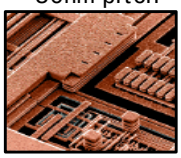
Carbon Nanotube transistor ~2nm width



Delft

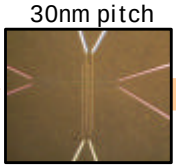


Copper wires, predicted ~50nm pitch



IBM

Nanowires, Already 30nm pitch



HP/UCLA

A trillion devices/cm²

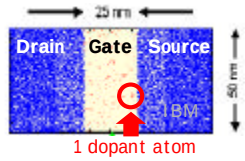
… but many of them probably won't work

… and those that do won't be identical

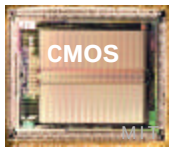
Independent of Technology

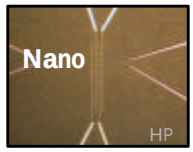
As we scale down:

- Devices become
  - more variable
  - more faulty (defects & faults)
  - numerous
- Fabrication becomes
  - More expensive
  - More constrained
- Design becomes
  - More complicated
  - More expensive



1 dopant atom





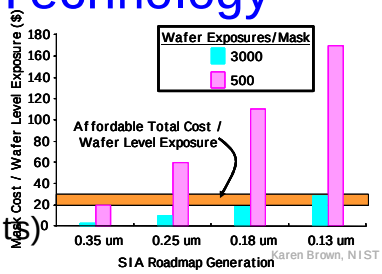
Requires:

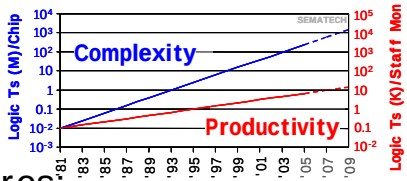
- Defect tolerant architectures
- Higher level specification
- Universal substrate

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





Requires:

- Defect tolerant architectures
- Higher level specification
- Universal substrate

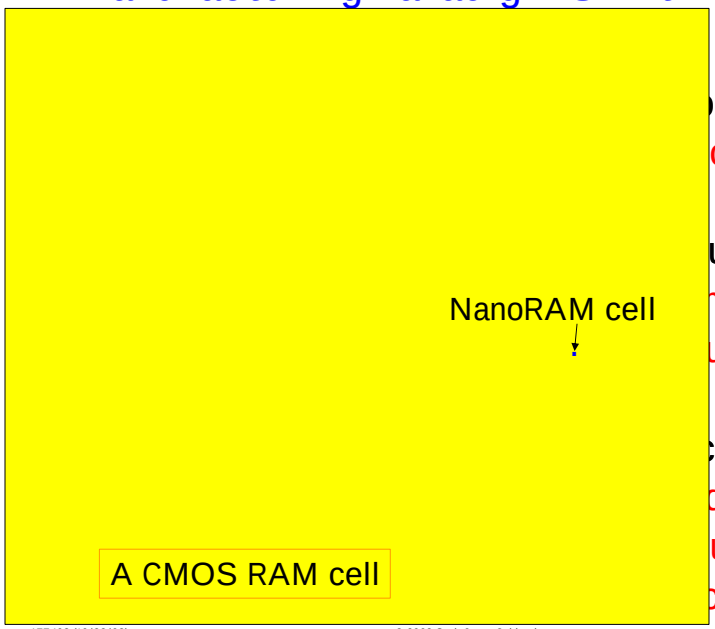
CMOS Too …

- Precision is expensive
- Current Abstractions have a major impact on cost
  - Requires process engineers to deliver robust devices
  - Requires perfect devices
- Process rules and mask costs are already eliminating arbitrary patterns (e.g., forbidden pitches)
- CMOS: *just another nanoscale component*

Manufacturing Paradigm Shift Required

Today	Future
• 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

Manufacturing Paradigm Shift Required



onents
components

ufacture
nanufacture
up assembly

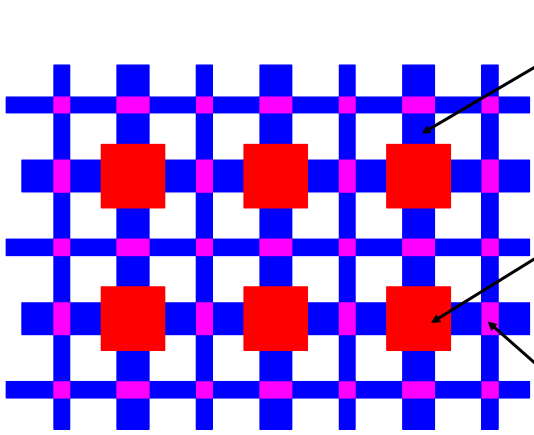
cales down
d variability
unctionality
connectivity

Defect Tolerant Architectures

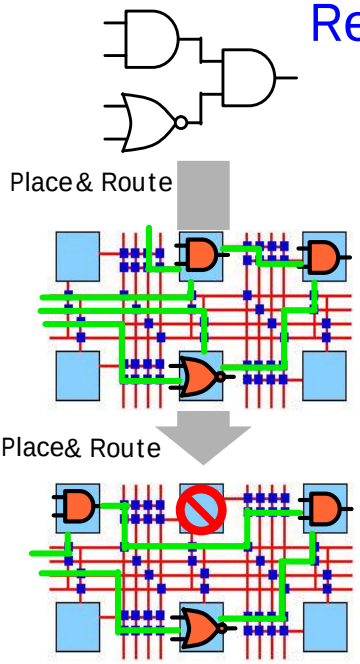
- Features:
  - Regular topology
  - Homogenous resources
  - Fine-grained?
  - Post-fabrication modification
- Example from today: DRAM
  - Requires external device for testing
  - Requires external device for repair
- Logic? FPGA

Key is redundancy

FPGA



Reconfigurability & DFT



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

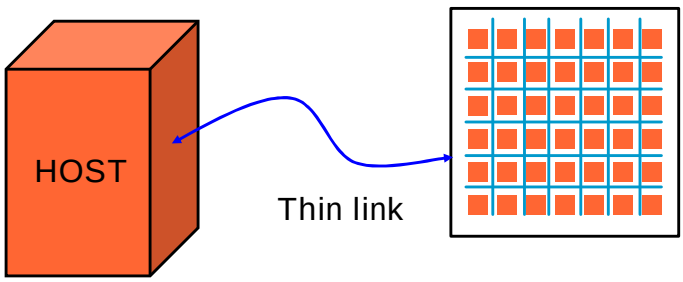
Defect Tolerance

Teramac works with mostly broken parts:

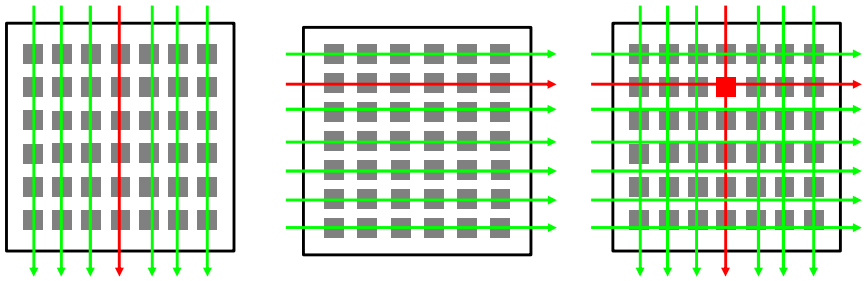
Defect mapping
+
Fine-grained configurability
⇒
Working device

Finding The 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

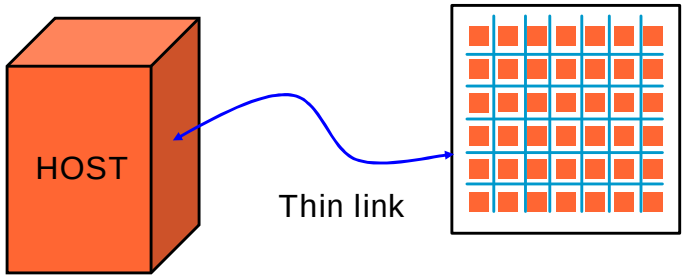
Configure testers horizontally

Identify fault!

Configure the device to test itself!

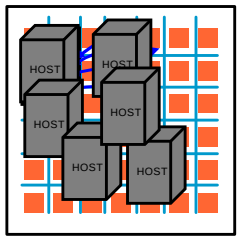
Scaling to 10^{11}

- Eliminate Host
- After initial area is found, configure chip to act as host
- At end of complete test, upload entire defect map.
- For proposed device <1 day to map



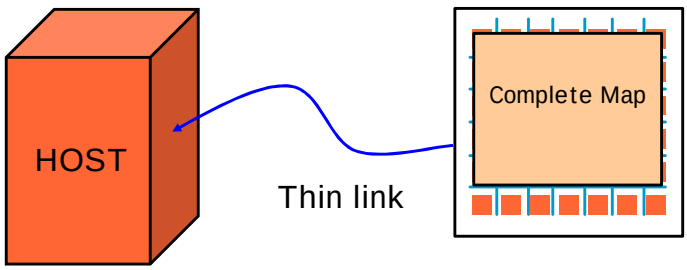
Scaling to 10^{11}

- Eliminate Host
- After initial area is found, configure chip to act as host
- At end of complete test, upload entire defect map.
- For proposed device <1 day to map

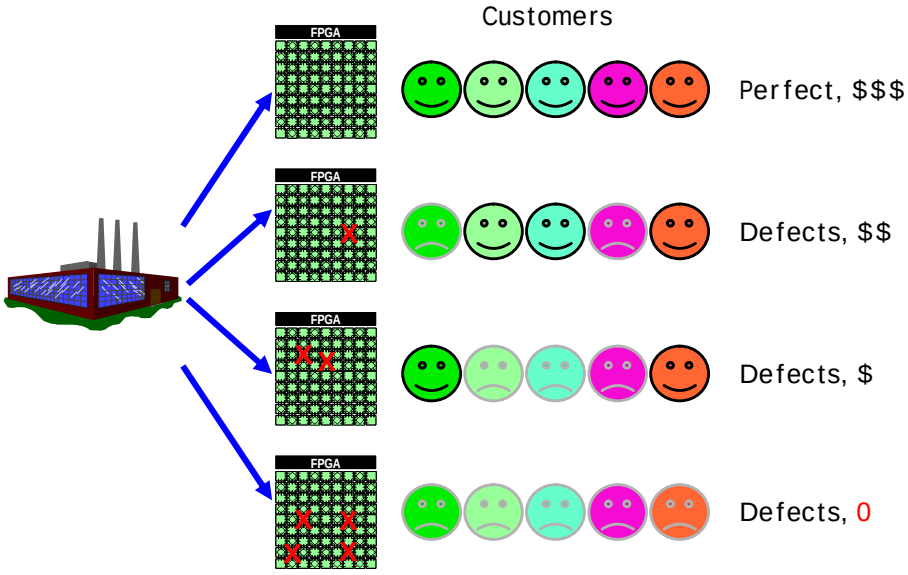


Scaling to 10^{11}

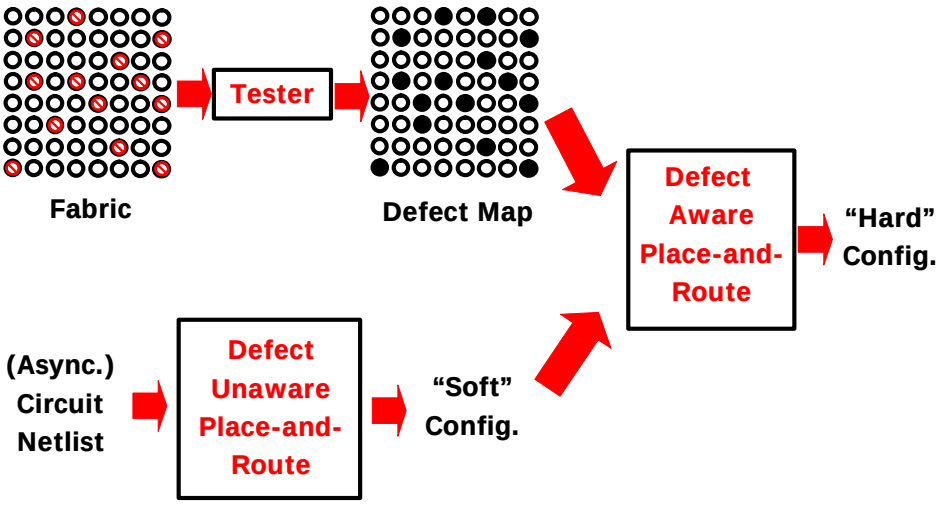
- Eliminate Host
- After initial area is found, configure chip to act as host
- At end of complete test, upload entire defect map.
- For proposed device <1 day to map



Coping With Defects



DFT tool flow



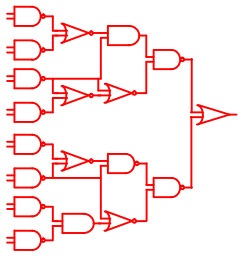
Reconfigurable Computing

General-Purpose

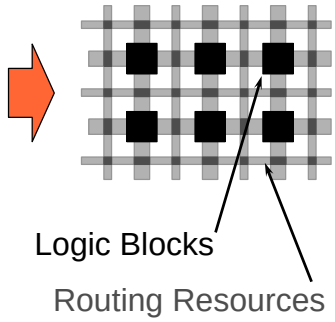
```
int reverse(int x)
{
   int k,r=0;
   for (k=0; k<64; k++)
      r |= x&1;
      x = x >> 1;
      r = r << 1;
   }
}
int func(int* a,int *b)
{
   int j,sum=0;
   for (j=0; *a>0; j++)
      sum+=reverse(*b
```

Compiler

Custom Hardware



General-Purpose
Custom Hardware



Advantages of Reconfigurable

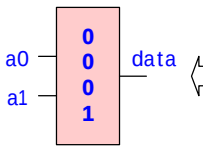
- Flexibility of a processor
- Performance of custom hardware

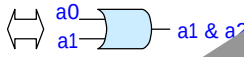
Near

You have to
- Store and
- Address

the configuration

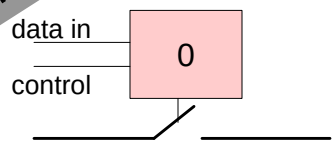
Heart of an FPGA





Universal gate =

The cost of the FPGA:
Increased Area -Delay Product



Switch controlled by a 1-bit RAM cell

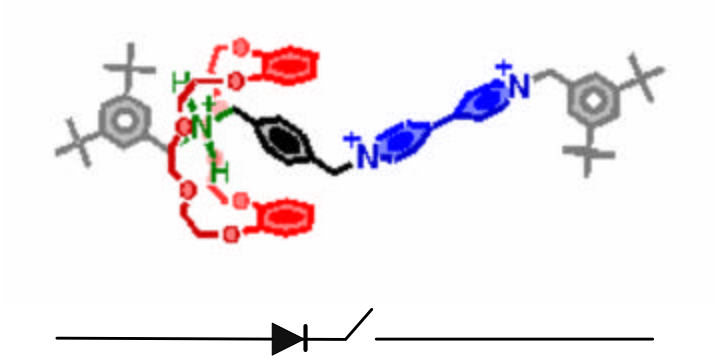
Key Component: Reconfigurable Switch

[2]Rotaxane
Molecular Switch

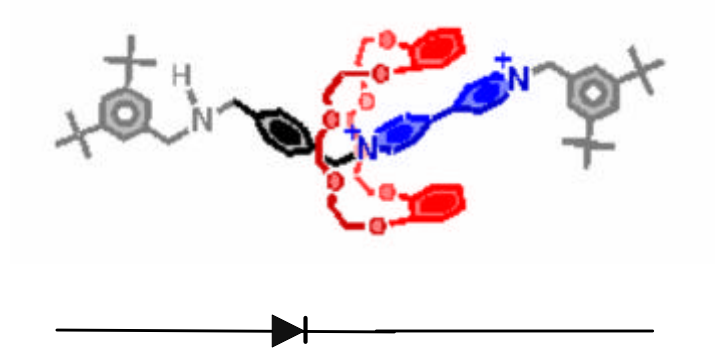
Complexation: Crown-Ammonium
Conditions: Neutral
Colour: Colourless

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

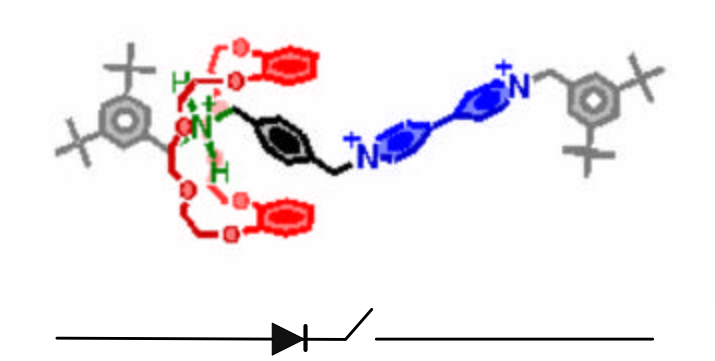
Key Component: Reconfigurable Switch



Key Component: Reconfigurable Switch

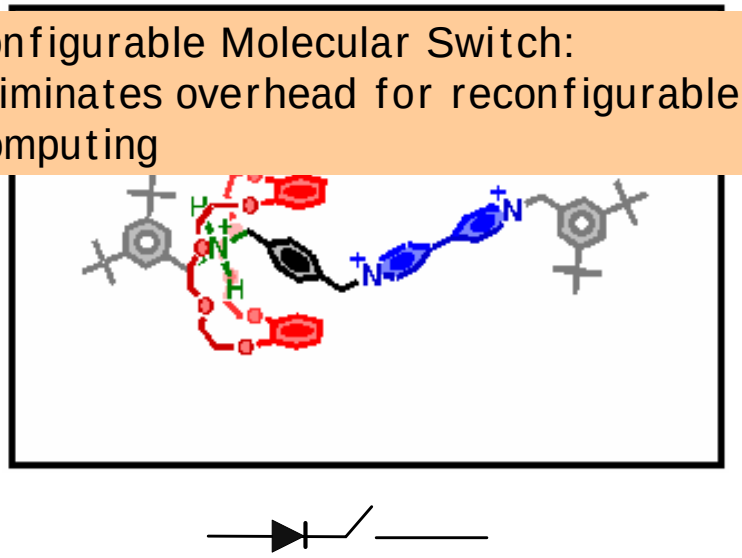


Key Component: Reconfigurable Switch

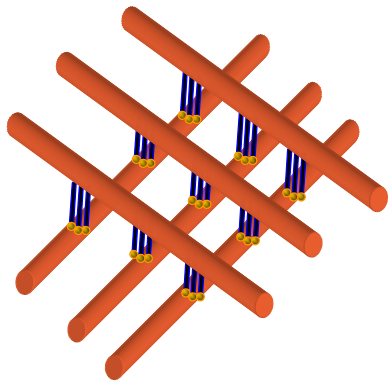


Key Component: Reconfigurable Switch

Reconfigurable Molecular Switch:
Eliminates overhead for reconfigurable computing



The Molecular Electronics Advantage: A Reconfigurable Switch

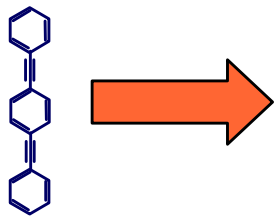


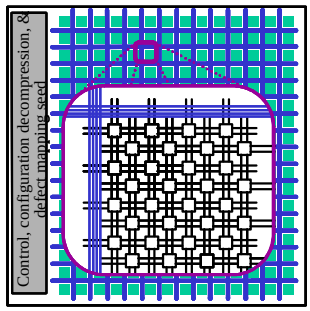
- Each crosspoint is a reconfigurable switch
- Can be programmed using the signal wires

Eliminates overhead found in CMOS FPGAs!

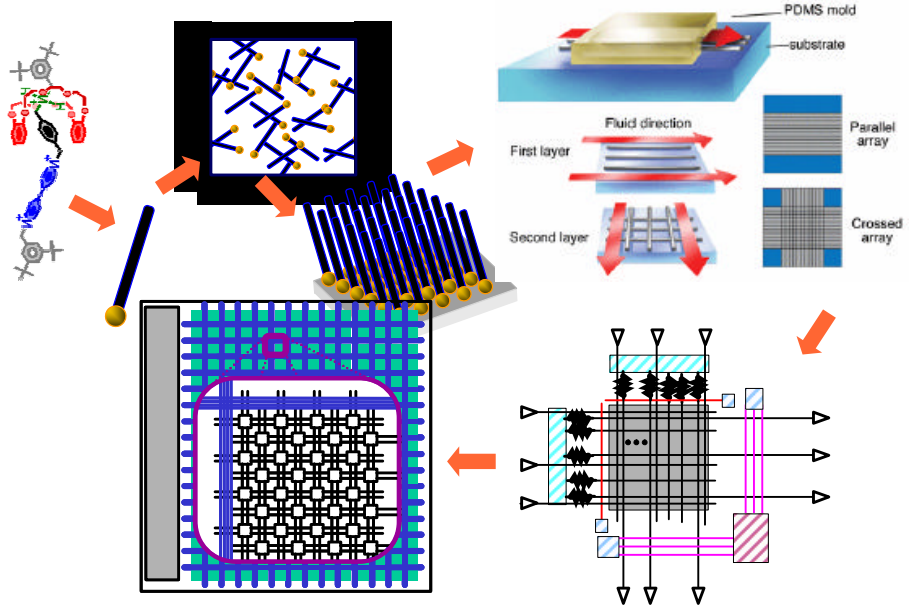
Fabrication Is Different

- Devices & wires alone are not useful
- Key to nanoscale computing is bottom-up assembly

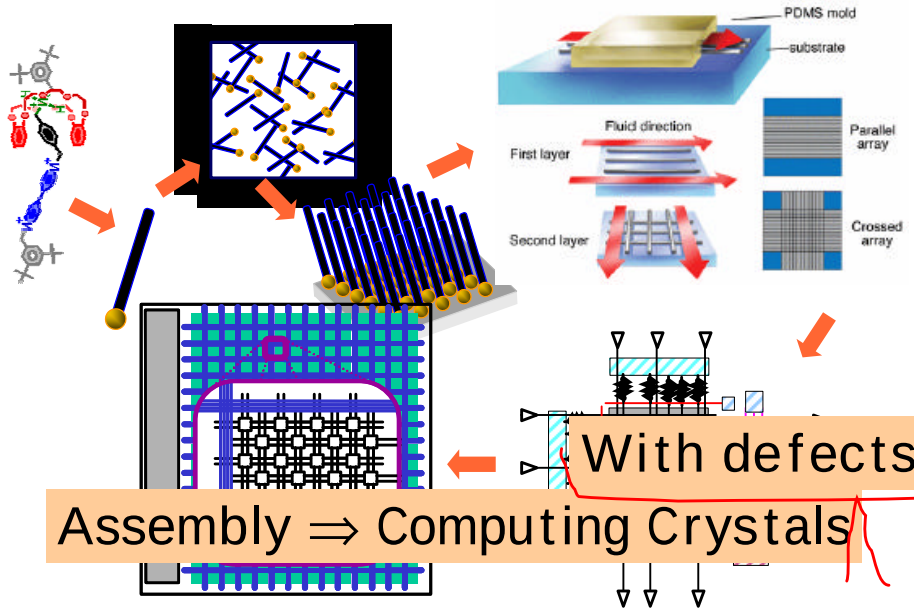




Building a Computing Crystal



Building a Computing Crystal



Implications

- Use only 2 terminal devices?
- Only regular structures
- Defect-tolerance required
- Functionality must be added post-fabrication

→ Reconfigurable!

Remember, this all applies to future CMOS generations too!

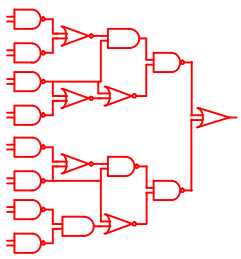
Reconfigurable Computing

General-Purpose

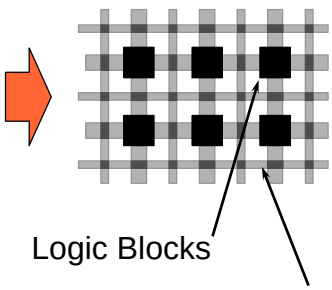
```
int reverse(int x)
{
   int k,r=0;
   for (k=0; k<64; k++)
      r |= x&1;
      x = x >> 1;
      r = r << 1;
   }
}
int func(int* a,int *b)
{
   int j,sum=0;
   for (j=0; *a>0; j++)
      sum+=reverse(*b
```

Compiler

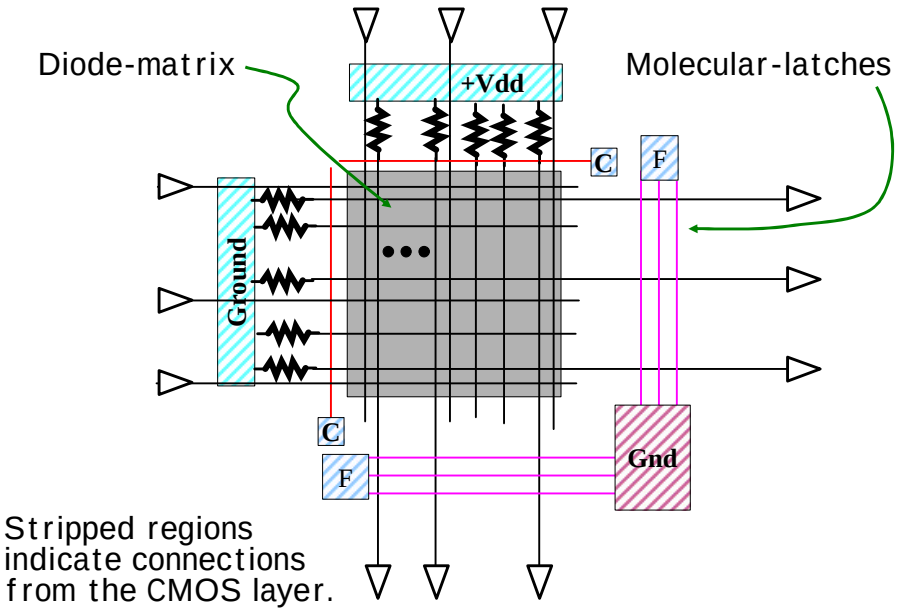
Custom Hardware



General-Purpose Custom Hardware

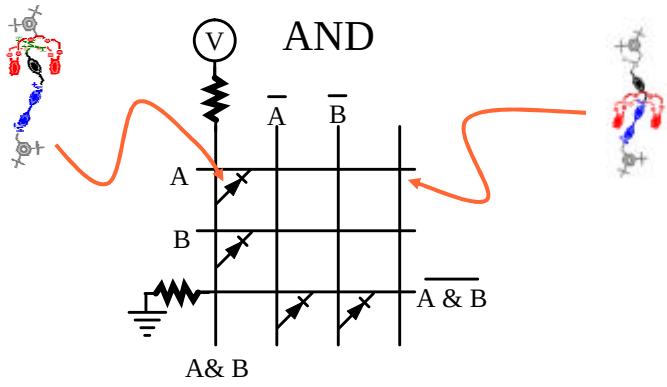


NanoBlock

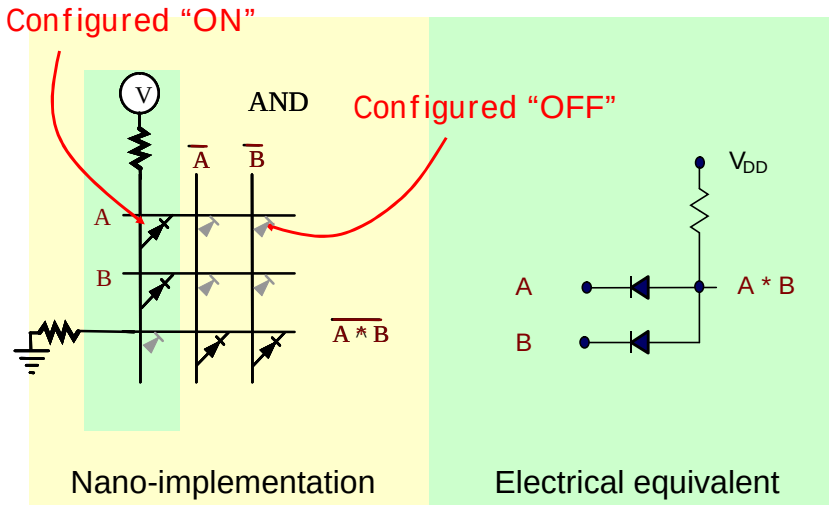


Computing with a nanoBlock

- The active component is a diode.
- Use diode-resistor logic



Diode-resistor Logic



How Do We Get There?

Abstractions Are The Key

- Transistor
  - An ideal switch and isolator
  - Identical all across the chip
- Circuit
  - Can be laid out on the chip
  - Manufactured without defects
- Instruction Set Architecture
  - Hides all details of the internal architecture
  - Mechanism for specifying programs

As Fabrication goes, so …

Necessary Device Abstraction

- Transistor provides it all
  - Switch
  - Fan-out

  } Necessary for logic
  - I/O isolation
  - Signal restoration

  } Necessary for large designs
- Replace single abstraction with multiple abstractions: SIRM
- Enables an incremental path to direct nano-circuits

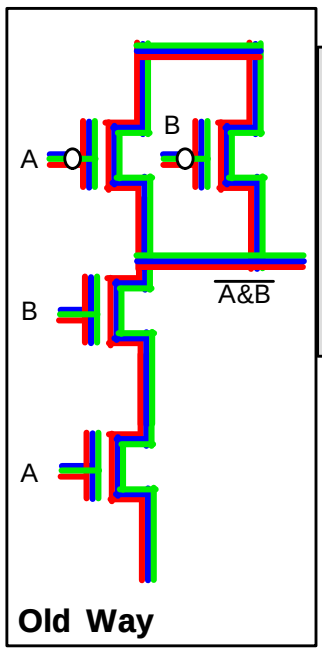
SIRM

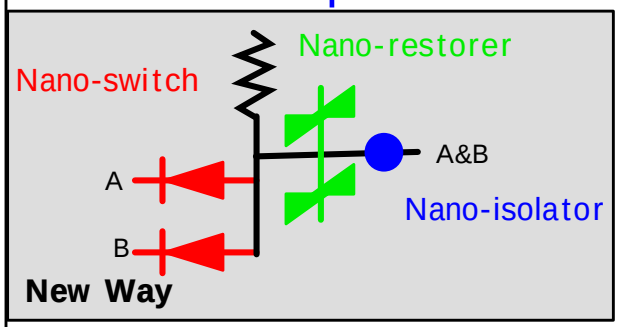
- SIRM decomposes the abstraction into its fundumental components.
  - Switch
  - Isolation
  - Restoration
  - Memory

  Necessary for logic

  Necessary for large designs
- SIRM already in use
- Allows circuit designers to explicitly manage the different components of a scalable circuit technology

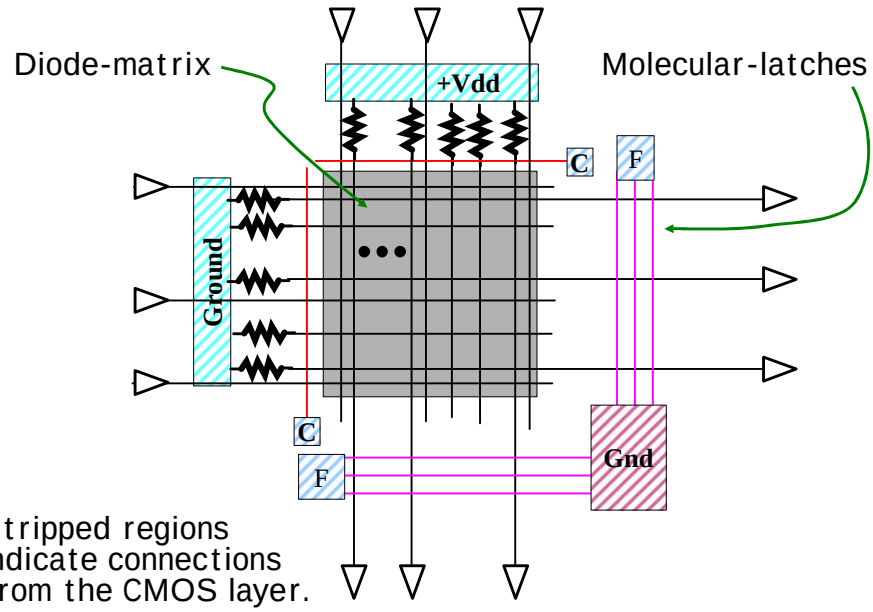
SIRM Example



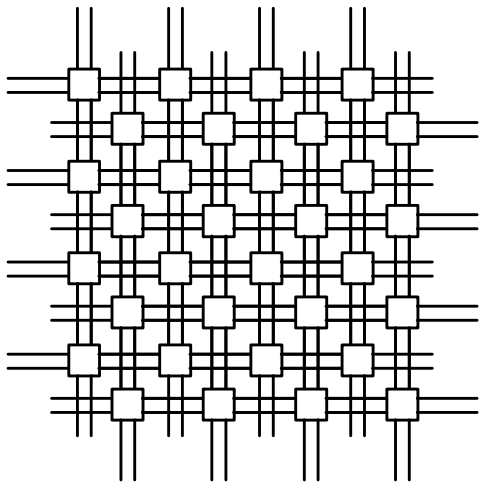


- Breaks up transistor abstraction into its components
- Allows creation of library of composable components
- Increases freedom of design

NanoBlock

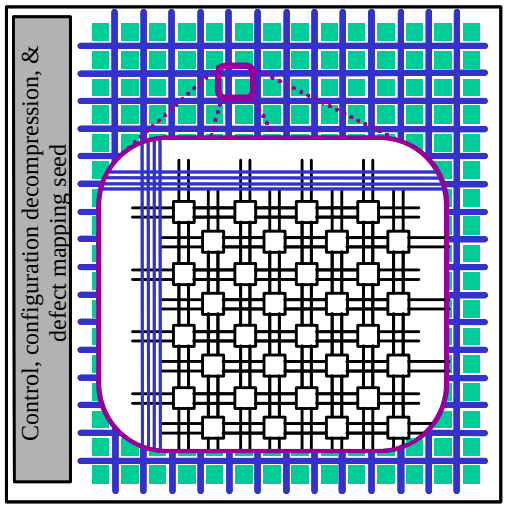


Cluster of NanoBlocks



The NanoFabric

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

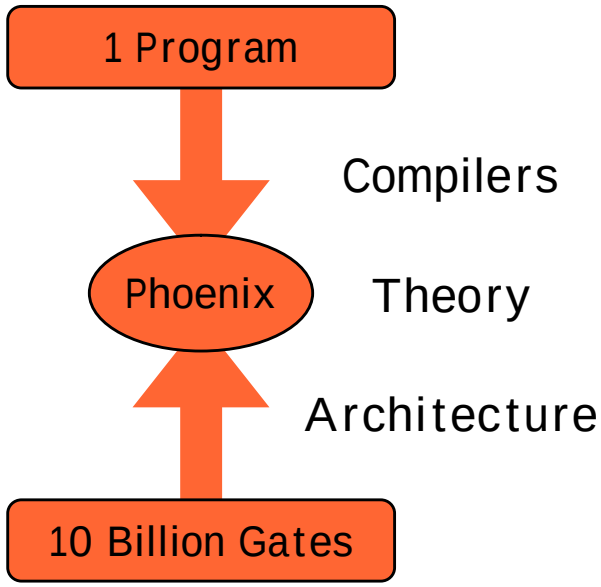


NanoFabric Attributes

- 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?

Spanning 10-orders of Magnitude



Bryant's Law

- Processor verification is always 10 years behind
- What to do?

Don't use processors!

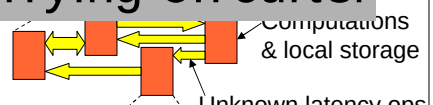
Circuits From Compilers

1. Program

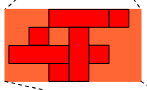
```
int reverse(int x)
{
   int k,r=0;
   for (k=0; k<64; k++)
      r |= x&1;
      x = x >> 1;
      r = r << 1;
```

Certifying Circuits!

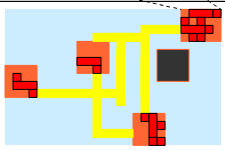
2. Split-phase Abstract Machines



3. Configurations placed independently



4. Placement on chip



Phoenix

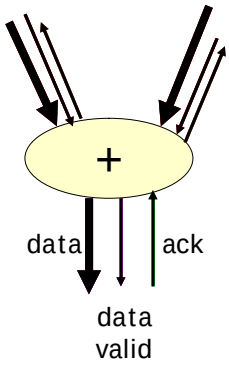
General: applicable to today's software
- programming languages
- applications

Automatic: compiler-driven

Scalable: - run-time: with clock, hardware
- compile-time: with program size

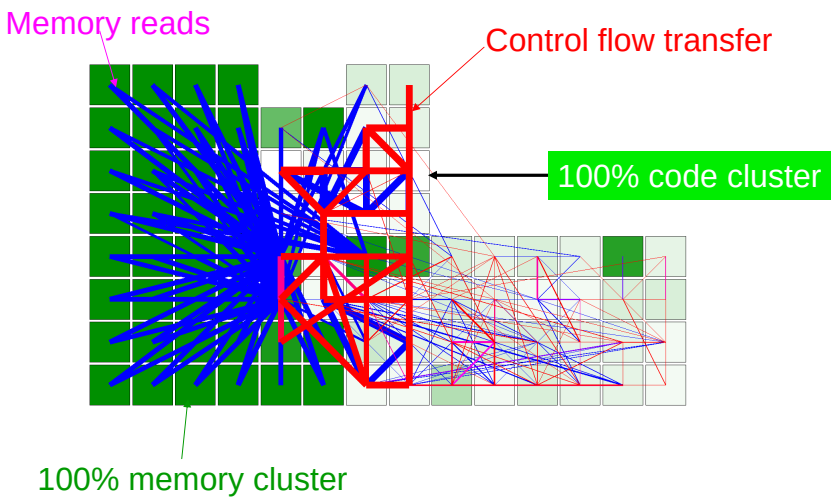
Parallelism: exploit application parallelism

Asynchronous Computation

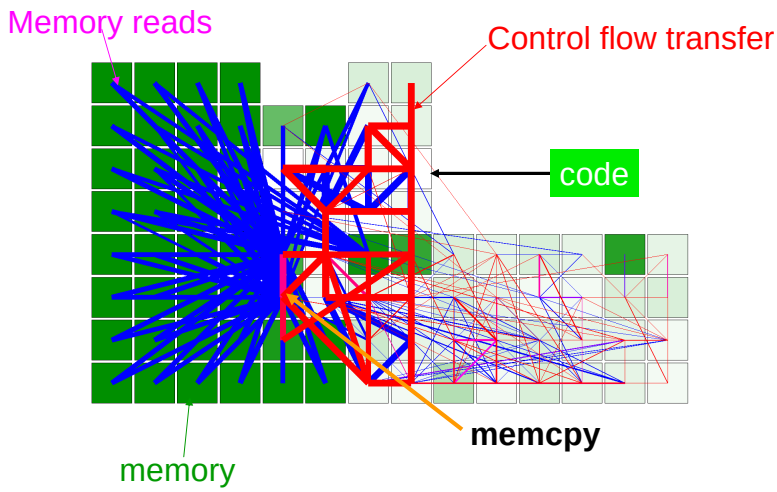


- Tolerant of
  - Layout
  - Parametric variation
  - Variable Latency Operations
- Low-power
- Natural implementation for dataflow

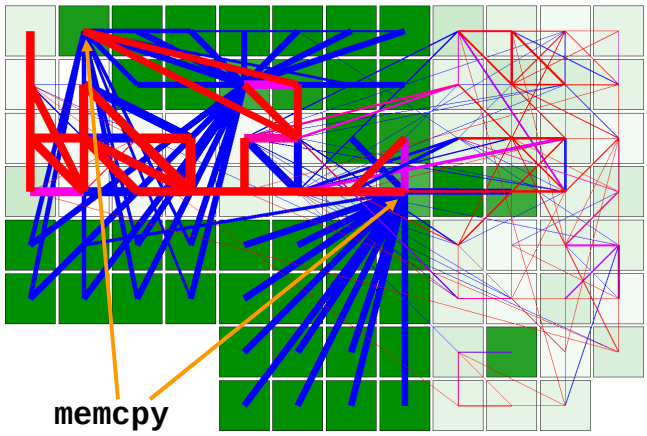
Typical Program Graph (g721_e)



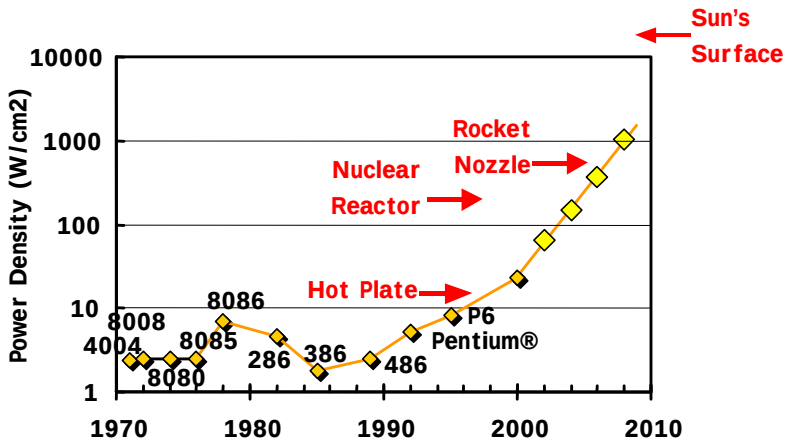
Typical Program Graph (g721_e)



Program Graph After Inlining **memcpy**



Another Consideration: Power



- Power densities too high to keep junctions at low temps

Source: Irwin & Borkar, De Intel â

Power α 1/Parallel

- $P \alpha CV^2F$
- $F \alpha \frac{(V-V^{th})^{5/4}}{V}$, for CMOS
- In relevant range: $F \alpha V$
- So, $P \alpha CF^3$
- Also, $1/A \alpha T^n$, early VLSI theory result $(1 \leq n \leq 2)$
- If we fix T, then $F^{-1} \alpha T$, $F \alpha A^{-1/n}$
- If n=2, $P \alpha CA^{-3/2}$
- If $C \alpha A$, $P \alpha A^{-1}$

Conclusions

- Reconfigurable Computing is inevitable
  - Cost of manufacturing
  - Defect tolerance
  - Fabrication constraints
- Molecular switches are ideal for reconfigurable device
- EN: New constraints, but huge potential
  - Billions of devices per cm^2
  - Ultra-low power
  - Faster design time
  - Easier verification