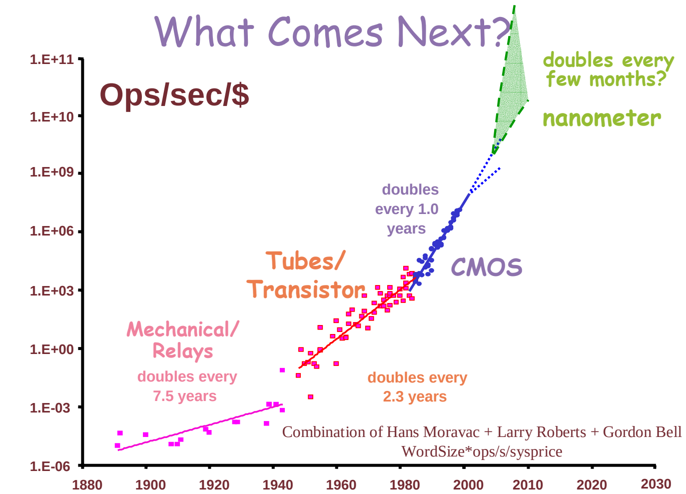
From Molecules to Computers

Seth Copen Goldstein
Carnegie Mellon University

Istanbul, Turkey
November 19, 2002

What Comes Next?



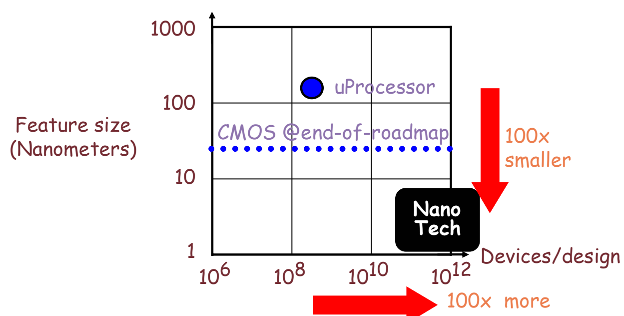
From Gray Turing Award Lecture

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

On the cusp of Major Technology Change

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

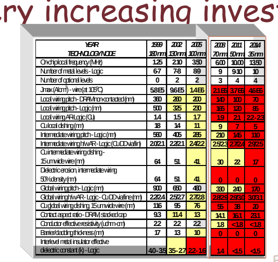


Outline

- The current path
- Molecular Electronic Devices
- Fabrication
- Implications
- Circuits
- Architectures
- Compilation
- Defect Tolerance

Current Approach

- Today's Approach
  - CMOS
  - Photolithography (Top-down manufacturing)
  - ITRS "roadmap"
- Requires every increasing investment of
  - Time
  - Power
  - Money

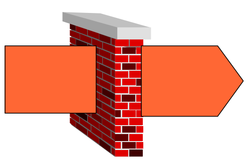


YEAR TECHNOLOGY NODE	1999 180nm	2002 130nm	2005 100nm	2008 70nm	2011 50nm	2014 35nm
On-chip local frequency (MHz)	125	210	350	600	1000	1350
Number of metal levels - Logic	6-7	7-8	8-9	9	9-10	10
Number of optional levels	0	2	2	3	4	4
Jmax (A/cm²) - wire (at 105°C)	5.8E5	9.6E5	1.4E6	2.1E6	3.7E6	4.6E6
Local wiring pitch - DRAM non-contacted (nm)	360	260	200	140	100	70
Local wiring pitch - Logic (nm)	500	325	230	165	120	85
Local wiring AR-Logic (Cu)	1.4	1.5	1.7	1.9	2.1	2.2-2.3
Cu local dishing (nm)	18	14	11	9	7	5
Intermediate wiring pitch - Logic (nm)	560	405	285	210	145	110
Intermediate wiring h/w AR - Logic (Cu DD via/lin)	2.0/2.1	2.2/2.1	2.4/2.2	2.5/2.3	2.7/2.4	2.9/2.5
Cu intermediate wiring dishing - 15 um wide wire (nm)	64	51	41	30	22	17
Dielectric erosion, intermediate wiring 50% density (nm)	64	51	41	0	0	0
Global wiring pitch - Logic (nm)	900	650	460	330	240	170
Global wiring h/w AR - Logic - Cu DD via/line (nm)	2.2/2.4	2.5/2.7	2.7/2.8	2.8/2.9	2.9/3.0	3.0/3.1
Cu global wiring dishing, 15 um wide wire (nm)	116	95	76	55	38	20
Contact aspect ratio - DRAM, stacked cap	9.3	11.4	13	14.1	16.1	23.1
Conductor effective resistivity (uohm-cm)	2.2	2.2	2.2	1.8	<1.8	<1.8
Barrier/cladding thickness (nm)	17	13	10	0	0	0
Interlevel metal insulator effective dielectric constant (k) - Logic	4.0-3.5	3.5-2.7	2.2-1.6	1.4	<1.5	<1.5

From ITRS Roadmap

Scaling Is Getting Harder

- For the first time, scaling appears to have reached fundamental limits in several areas.
- Current ITRS contains major barriers with no known solutions.
- Red Brick Wall is getting closer!
- Overcoming these barriers requires:

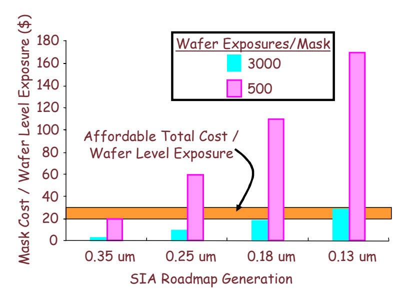


- ***New*** materials
- ***New*** types of devices
- ***New*** physical models
- ***New*** types of processing
- ***New*** design tools & methodology

George Bourionoff, Intel

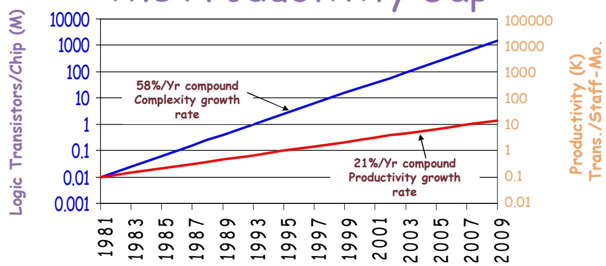
Cost Is a Driving Force

- Costs are increasing
  - Development
  - Fabrication plants
  - Wafers
  - Tools
  - Masks
  - Design



Karen Brown, NIST

The Productivity Gap

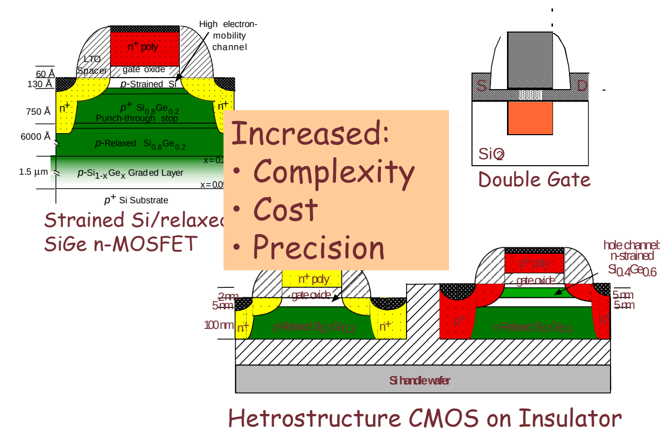


Year	Feature Size (nm)	Logic trans/chip (M)	3 Yr. Design Staff Size	Staff Costs
1997	350	6	210	$90 M
1998	250	9.5	270	$120 M
1999	180	15	360	$160 M
2002	130	59.5	800	$360 M

* @ $ 150K/Staff Yr. (in 1997 Dollars) Source: SEMATECH

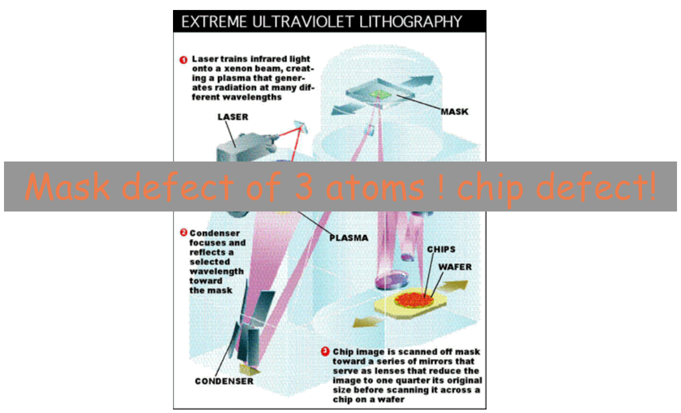
Micro tutorial © 2002 Seth Copen Goldstein 9

Example Current Directions



Micro tutorial © 2002 Seth Copen Goldstein 10

EUV?



Scientific American, 2001

Micro tutorial © 2002 Seth Copen Goldstein 11

Should We Worry?

- Basic technology is 40 years old
- "Improvements" take a long time
- Cost is becoming prohibitive
- Red Brick wall is drawing closer
- Yet
  - electrical parameters getting worse
  - Efficiency is going down (E.g., SpecInt/Mhz)

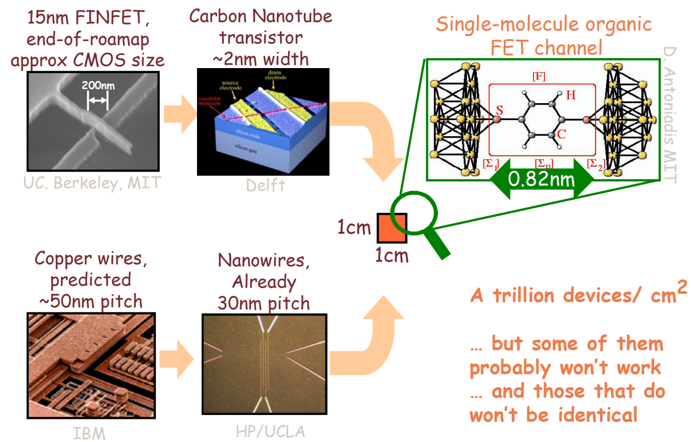
Bottom Line: Size Matters!

Micro tutorial © 2002 Seth Copen Goldstein 12

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

Size Matters

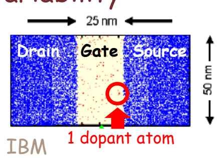


A trillion devices/ cm^2

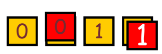
… but some of them probably won't work
… and those that do won't be identical

Consequences of Smallness

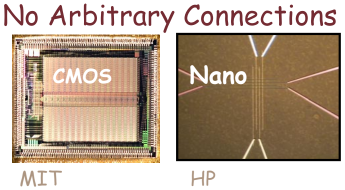
- Variability



- Single-event upsets



- No Arbitrary Connections



- Solutions:
  - post-fabrication reconfiguration
  - Computation
  - redundancy

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

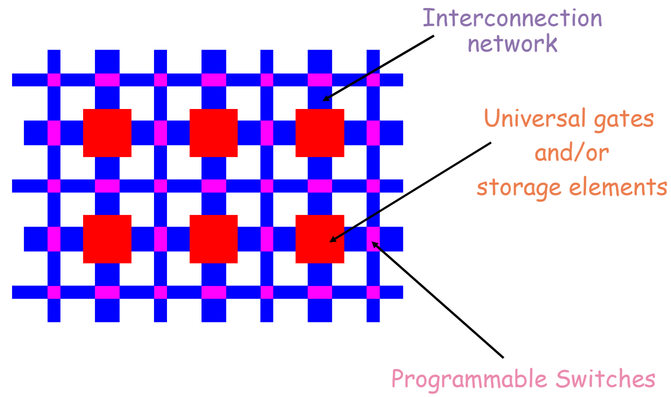
Requires:
- Defect tolerant architectures
- Higher level specification
- Universal substrate

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

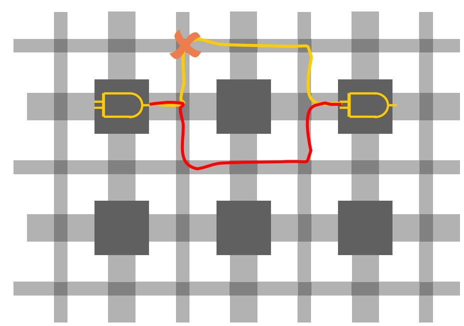
Island-Style FPGA



Dealing With Defects

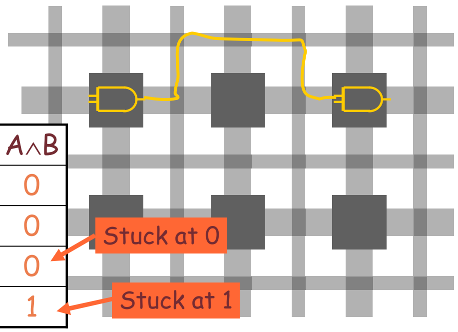
- Assume there exists a map of the defects.
  - Broken connections (stuck-open)
  - Shorted wires (stuck-closed)
  - Bad devices
- Only use the good devices
  - Route around defects
  - Use good devices
- Sometime, no loss in logic density

Routing Around Defects



- Choose new channels
- Choose new wires in the same channels
- Re-place destination or source

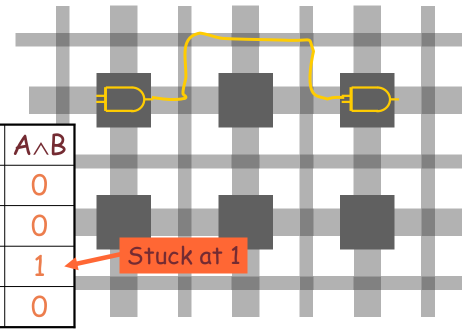
Configure Around Defects



A	B	A∧B
0	0	0
0	1	0
1	0	0
1	1	1

- See if stuck-open or closed fits function

Configure Around Defects

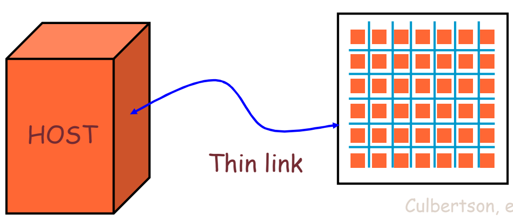


A	!B	A∧B
0	1	0
0	0	0
1	1	1
1	0	0

- See if stuck-open or closed fits function
- See if complement is available

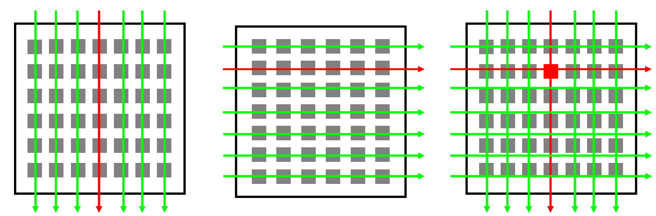
Finding Defects

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



Culbertson, et al, FCCM'97

Built-In Self-Test



Configure testers vertically

Configure testers horizontally

Identify fault!

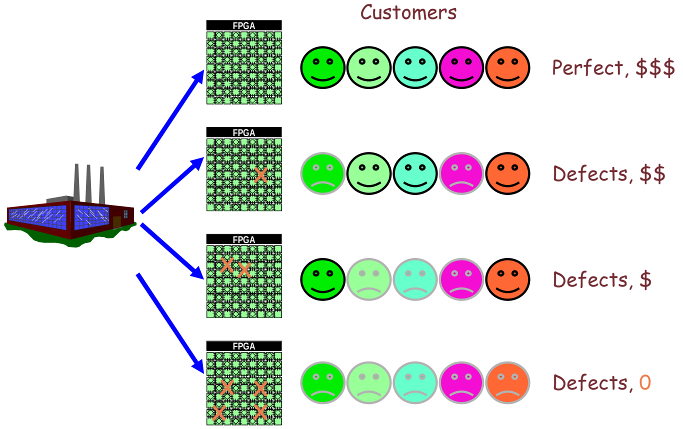
Configure the device to test itself!

Defect Tolerance Realized: Teramac

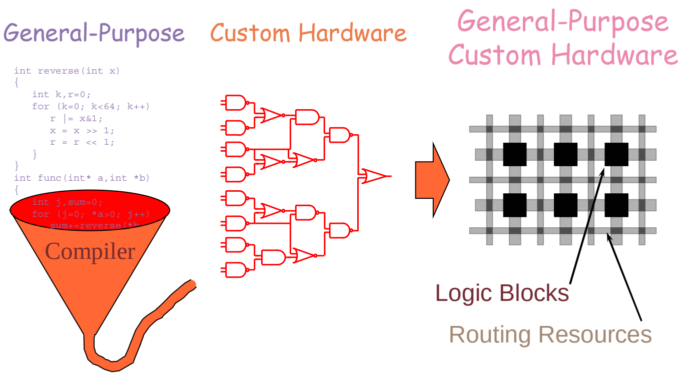
- Largest defect-tolerant system ever built
  - 8 PCBs, 4 MCMs each, 27 FPGAs each
  - 8 million total resources, 3% defective
- Compile around the defects
  - 10% of all logic cells
  - 10% all interchip wires
- Demonstrated years of reliable operation of large-scale defect-tolerant programmable hardware

HP

Coping With Defects

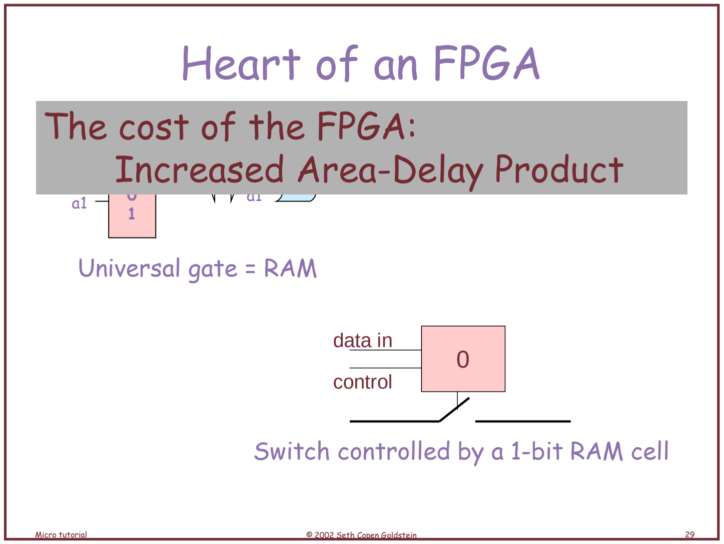


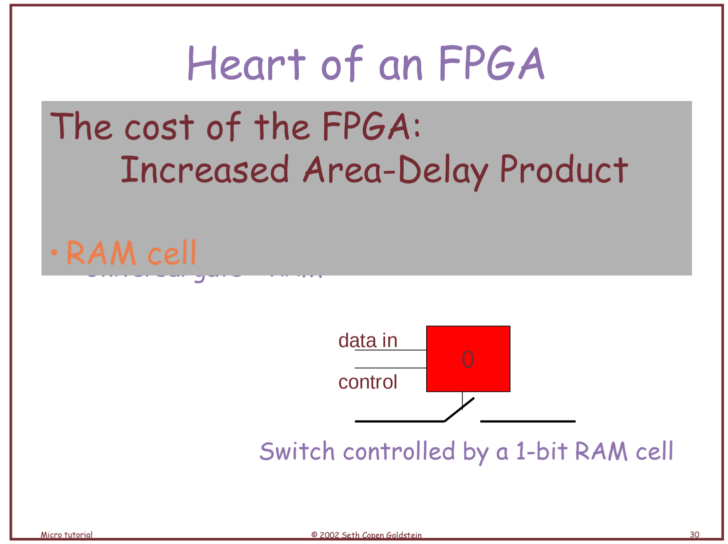
Reconfigurable Computing

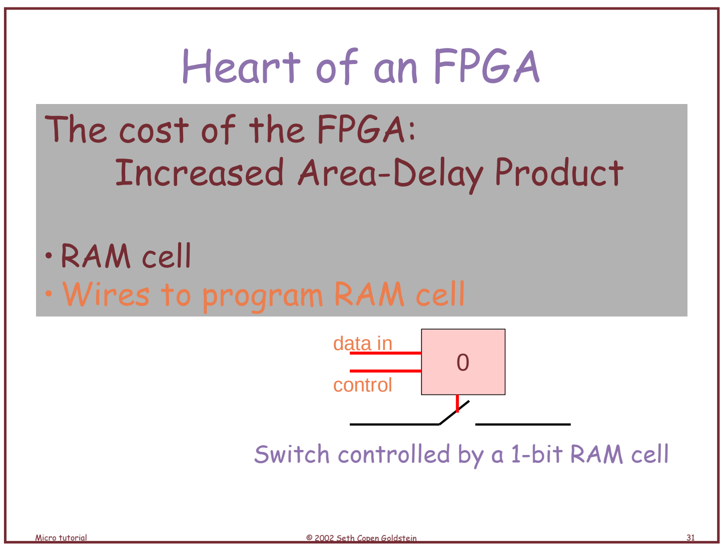


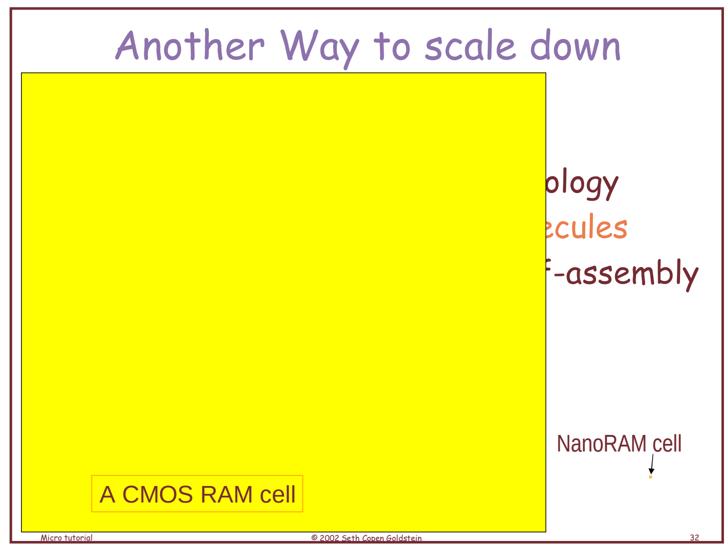
Advantages of Reconfigurable

- Flexibility of a processor
- Performance of custom hardware
  - Reduce time to market
  - Reduce design cost
  - Built-in
    - Dynamic fault tolerance
    - Self-test
    - Defect tolerance
  - Low power

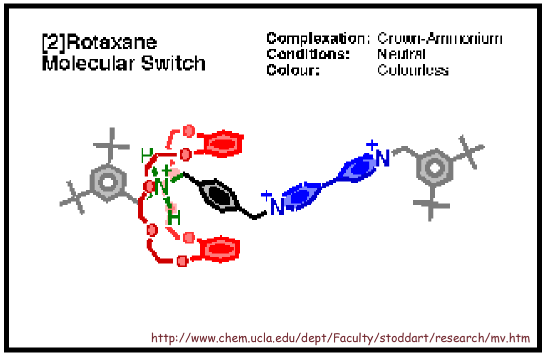

Heart of an FPGA
The cost of the FPGA:
Increased Area-Delay Product
Universal gate = RAM
data in
control
0
Switch controlled by a 1-bit RAM cell
Micro tutorial
© 2002 Seth Copen Goldstein
29


Heart of an FPGA
The cost of the FPGA:
Increased Area-Delay Product
• RAM cell
data in
control
0
Switch controlled by a 1-bit RAM cell
Micro tutorial
© 2002 Seth Copen Goldstein
30

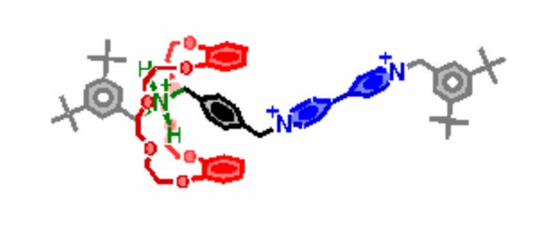

Heart of an FPGA
The cost of the FPGA:
Increased Area-Delay Product
• RAM cell
• Wires to program RAM cell
data in
control
0
Switch controlled by a 1-bit RAM cell
Micro tutorial
© 2002 Seth Copen Goldstein
31

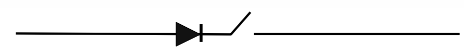

Another Way to scale down
ology
ecules
-assembly
NanoRAM cell
A CMOS RAM cell
Micro tutorial
© 2002 Seth Copen Goldstein
32

Key Component: Reconfigurable Switch

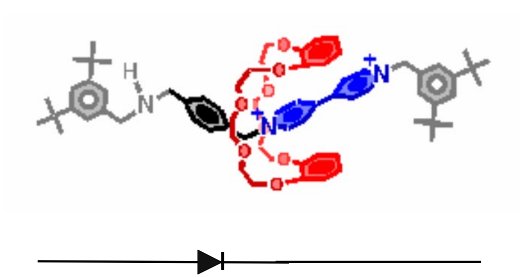


Key Component: Reconfigurable Switch

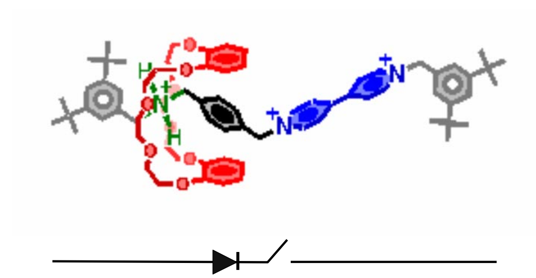




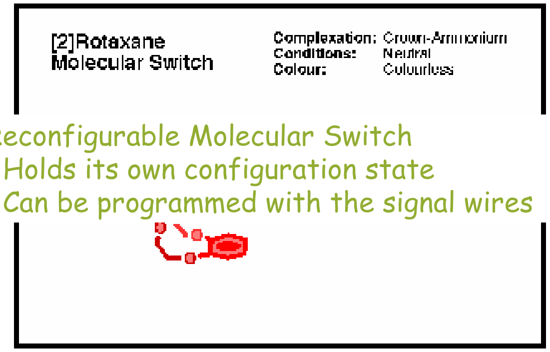
Key Component: Reconfigurable Switch



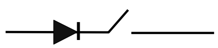
Key Component: Reconfigurable Switch



Key Component: Reconfigurable Switch

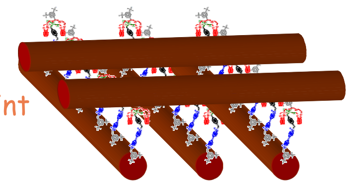


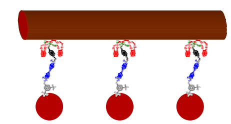
Reconfigurable Molecular Switch
- Holds its own configuration state
- Can be programmed with the signal wires

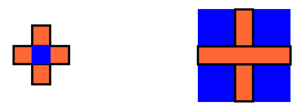


Using The Reconfigurable Switch

- Create a crossbar (perfect for FPGA)
- Programmable crosspoint fits in wire crossing!
- Compare to CMOS crosspoint (even with same dimension devices)





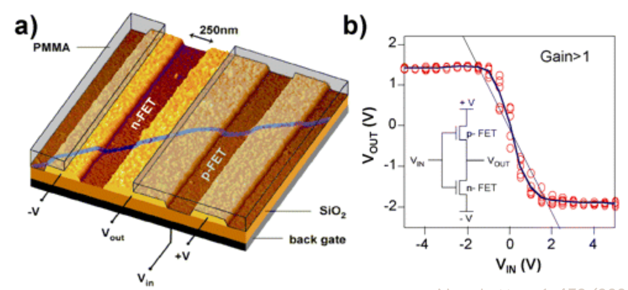


(to scale)

Molecular Device Characteristics

- Have useful and novel features
  - I-V curves
  - Devices with integrated state
  - Huge variety possible
  - Amenable to thermodynamic-assembly
- State of the art is changing rapidly!
- Requires new circuit methodologies

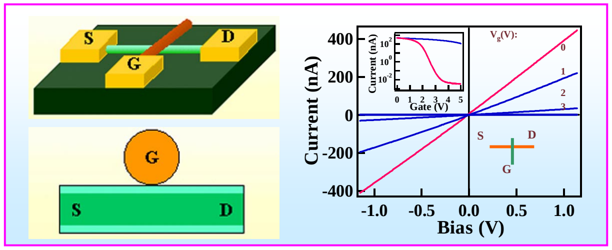
Carbon Nanotube Inverter



Nano Letters 1, 453 (2001)

- Two Carbon nanotube FETs
- Doped on substrate
- Not all dimensions are nanoscale

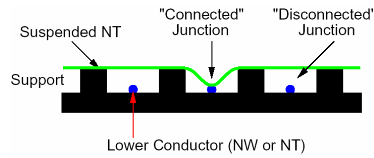
Crossed Nanowire FETs



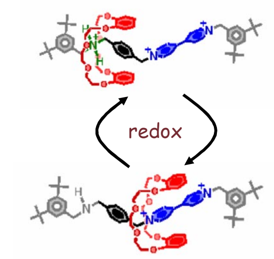
Science **294**, 1313 (2001)

- A crossed nanowire FET
  - channel width by the active nanowire diameter (to 2 nm)
  - channel length by the gate nanowire diameter (to 1-2 nm)
  - gate dielectric oxide coating on the nanowires (to 1 atomic layer)
- The conductance of cNW-FETs can be changed by more than 10^5-times with less than 0.1 V variation in the nano-gate.

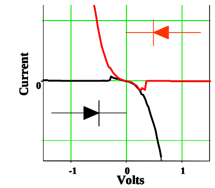
Molecular Switches



Science, 200, 289(5476), 94-97

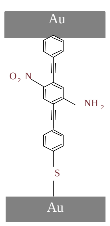


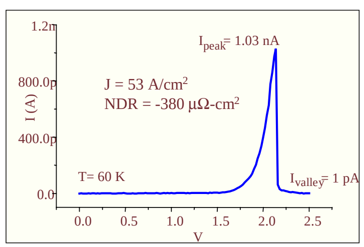
- Devices with state
- Programmed with signal wires
- Perfect for FPGA-like systems
- Also, Mono/di-nitros



Angew. Chem. Int. Ed. Engl., 1997, 36, 1904-190

Negative Differential Resistor (NDR)

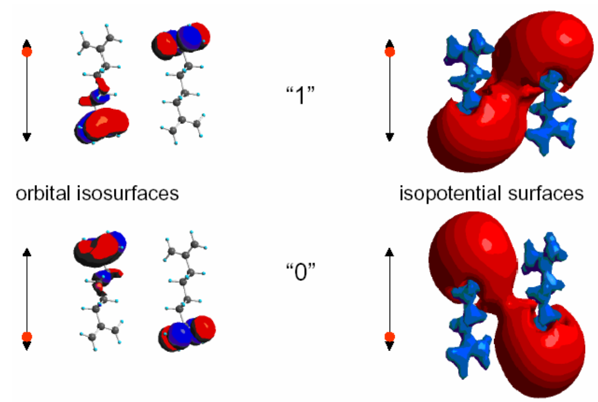




Mark Reed

- NDR with 1000:1 Peak-to-valley ratio
- Similar to RTDs
- Heavily explored in the 70s!

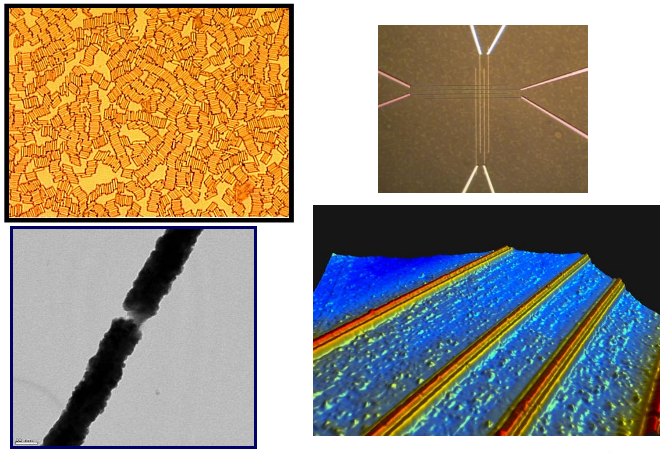
QCA Cell



Information carrier is NOT current or voltage

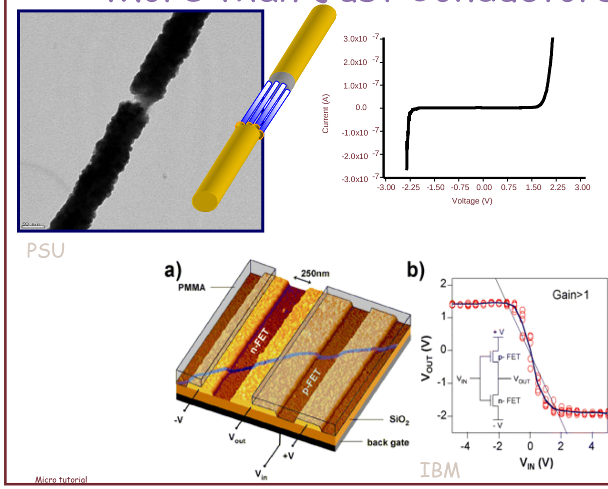
Notre Dame, Lieberman

Some Nanoscale Wires



Penn State, HP, UCLA

More than Just Conductors

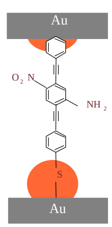


PSU

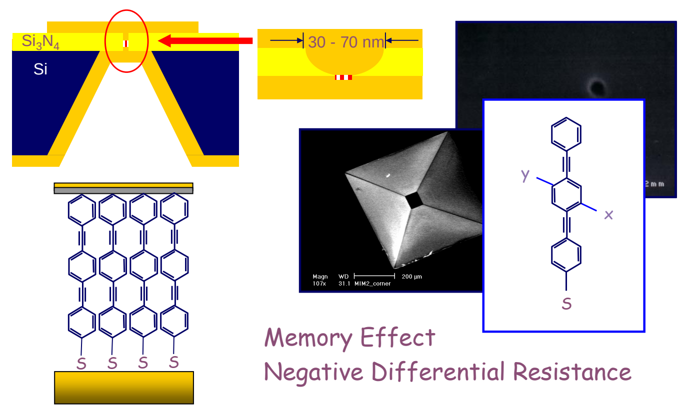
IBM

Contacts

- Contacts are still a problem
- Ohmic contacts are just part of the solution.
- Molecular devices and contacts cannot be analyzed in isolation
- E.g, DNA measured as
  - Conductor
  - Insulator
  - Semi-conductor
  - Proximity-induced superconductor



NanoPore



Memory Effect

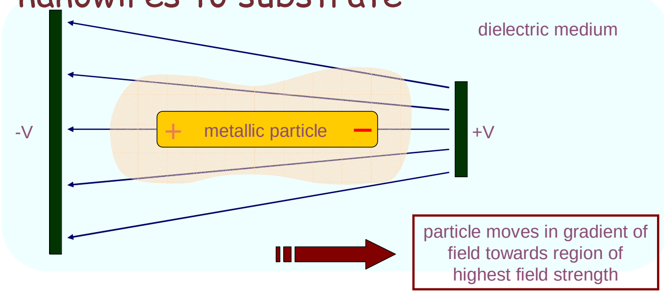
Negative Differential Resistance

Zhou, C., M. R. Despande, M. A. Reed, L. Jones, and J. M. Tour, *Appl. Phys. Lett.*, 71 (5) 1997

Theresa Mayer, PSU

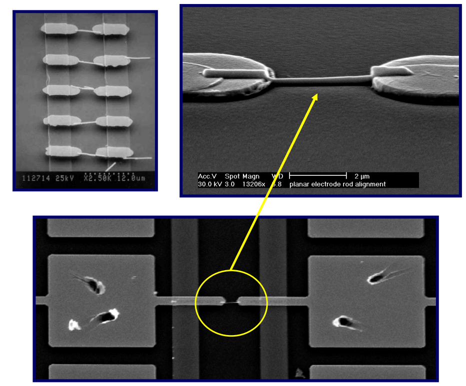
Electrofluidic Assembly of Nanowires

- Long-range forces used to attract nanowires to substrate



Theresa Mayer, PSU

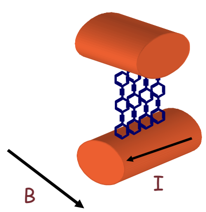
Characterizing Nanowires



Theresa Mayer, PSU

Measuring Devices

- Break Junction (Reed, Yale)
- Individual Molecules with STM (Weiss, PSU)
- Crossed-Wire Tunnel Junction (Kushmerick, NRL)

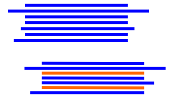


Fabrication Is Different!

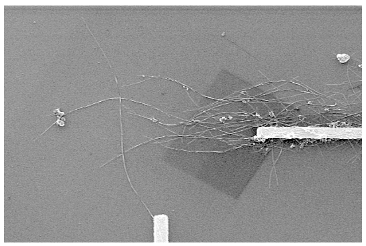
- Electronic nanotechnology separates:
  - Manufacturing of devices
  - Fabrication of the circuit
- Manufacture wires and devices
- Somehow glue them together

Assembly, Assembly, Assembly

Parallel Wires



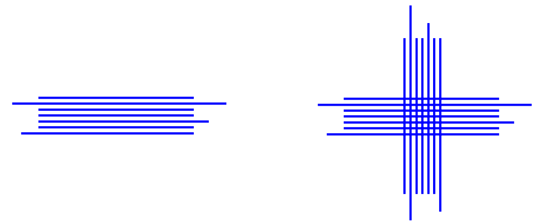
- Can align wires
- Can mix types of wires
- Not precise



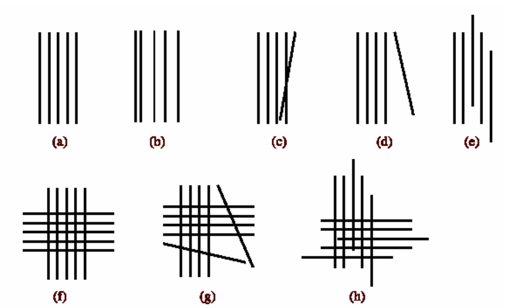
UCLA (2000); PSU(2000)

Fabrication Primitives

- Parallel wires
- 2-D grids

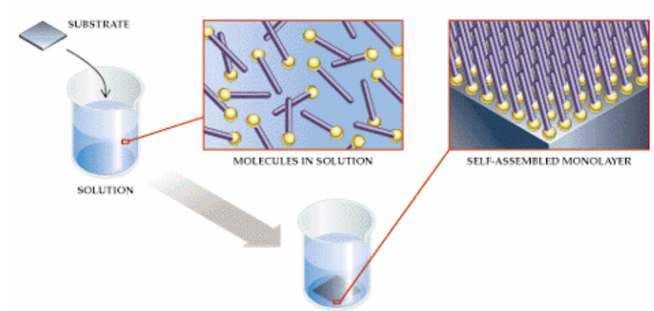


Some Possible Outcomes



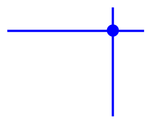
Fabrication Primitives

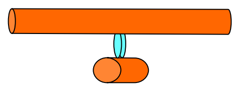
- Parallel wires
- 2-D grids
- Self-assembled monolayers



Fabrication Primitives

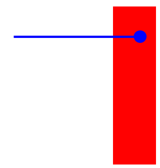
- Parallel wires
- 2-D grids
- Self-assembled monolayers
- Connections between orthogonal wires





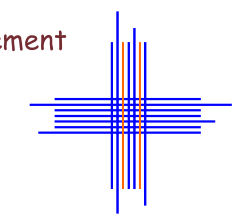
Fabrication Primitives

- Parallel wires
- 2-D grids
- Self-assembled monolayers
- Connections between orthogonal wires
- Connections to CMOS



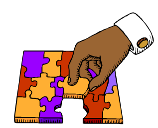
Fabrication Primitives

- Parallel wires
- 2-D grids
- Self-assembled monolayers
- Connections between orthogonal wires
- Connections to CMOS
- Non-deterministic placement



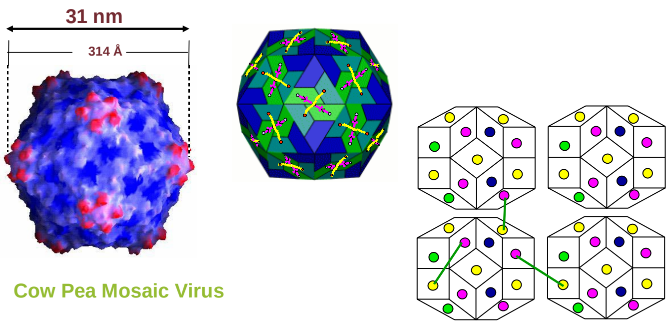
Fabrication Primitives

- Parallel wires
- 2-D grids
- Self-assembled monolayers
- Connections between orthogonal wires
- Connections to CMOS
- Non-deterministic placement
- Selective variability on a lithographic scale



Viral Crystals

Directed self-assembly in 3-D

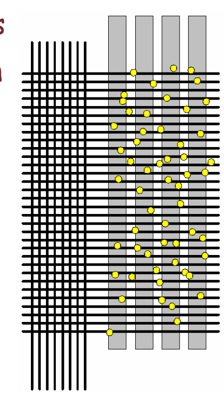


- Scaffolding for 3-D assembly

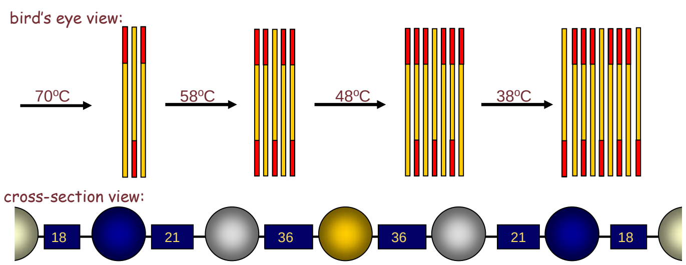
Paul Franzon

Harnessing Randomness

- Fab primitives prefer low information content structures
- Can we design circuits that can exploit this?
- Example: Scale Matching
  - Address 1 nanoscale wire with 5log(N) micronscale wires
  - Based on random placement of molecules



Temp-programmed Raft Assembly



Deterministic rafts will be assembled at the aq/aq interface via sequential assembly of nanowires harboring decreasing lengths of oligonucleotides A and A´ as the sample is cooled.

5' HS-$C_{12}H_{24}$-TTG AGA CCG TTA AGA CGA GGC AAT CAT GCA ATC CTG 3'

Length	T_m
36-mer	75ºC
21-mer	61ºC
18-mer	51ºC
15-mer	41ºC
9-mer	28ºC

Keating, PSU

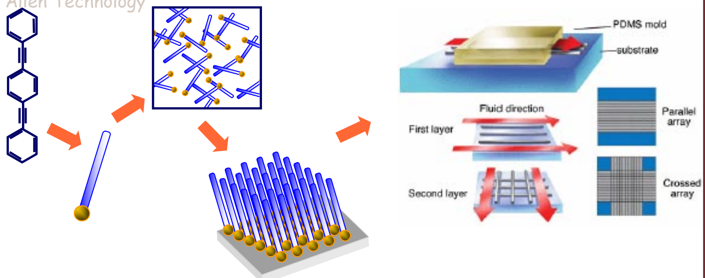
Nanoimprinting

- Create master
  - Ebeam
  - CVD & then break
- Stamp into "soft" material
- Stamp and leave behind molecules

Assembly, Assembly, Assembly

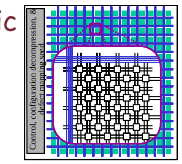
Alien Technology

ires alone are not useful
scale computing is bottom-



Computing Crystals

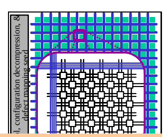
- A reasonable medium-term position
  - No end-to-end connections
  - Maybe multi-terminal devices
  - No complex predetermined patterns
  - Quasi-regular & non-deterministic
  - Hierarchical assembly



Assembly ⇒ Computing Crystals

Computing Crystals

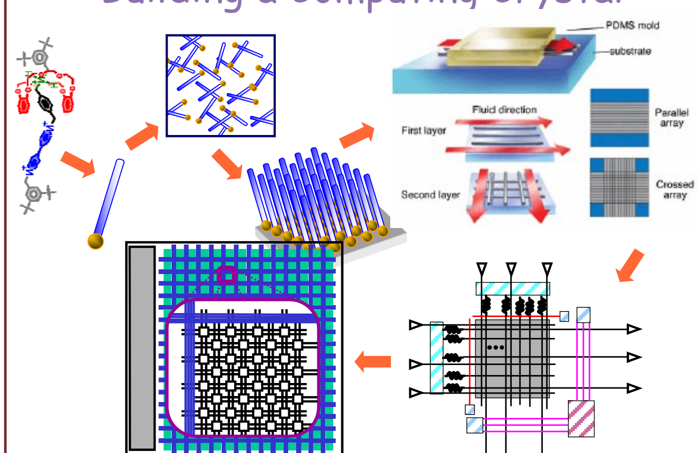
- A reasonable medium-term position
  - No end-to-end connections
  - Maybe multi-terminal devices
  - No complex predetermined patterns
  - Quasi-regular & non-deterministic
  - Hierarchical assembly



With defects

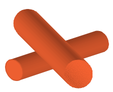
Assembly ⇒ Computing Crystals

Building a Computing Crystal



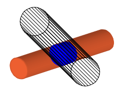
Some Implications

- Restricted set of manufacturable patterns
  - Current uncompressed single layer mask data is 70GB, (90nm → 220GB)
  - Can we even do 3-terminal devices?
- Increased defect densities
- Increased fault rates
- Efficient Programmable crosspoints



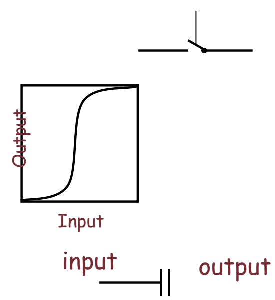
Some Implications

- Restricted set of manufacturable patterns
  - Current uncompressed single layer mask data is 70GB, (90nm → 220GB)
  - Can we even do 3-terminal devices?
- Increased defect densities
- Increased fault rates
- Efficient Programmable crosspoints



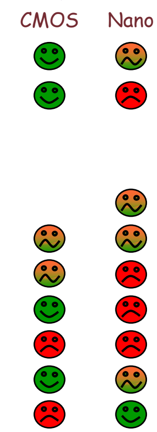
Good Computing Devices

- Requirements:
  - Complete Logic Family
  - Restoring Logic
  - Fan-out
  - I/O-isolation
  - Gain
  - Inexpensive
  - Expend power only when computing
- Lithographically created CMOS MOSFET practically perfect

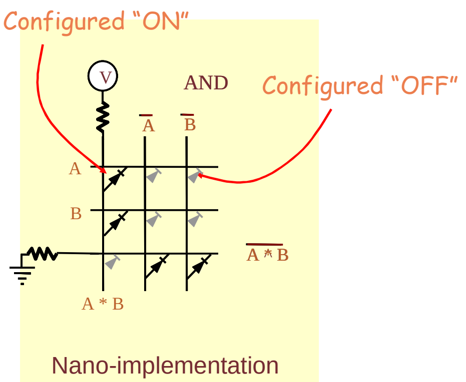


Requirements: Nano-Circuits

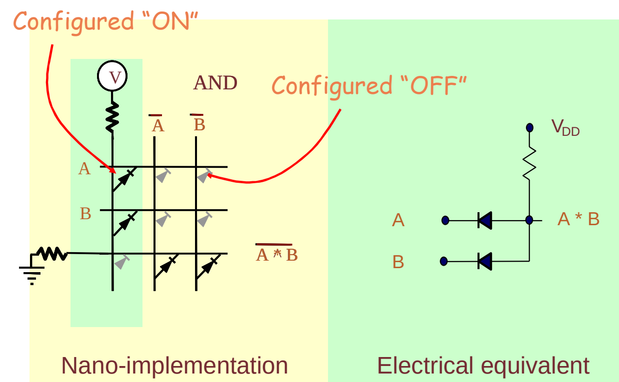
- Complete logic family
- Composable Circuits
  - Gain
  - Input/Output isolation
  - Fan-out
- Fabrication friendly
- Low variability
- Device/Wire ratio
- Energy-Delay product
- Harness Novel IV curves



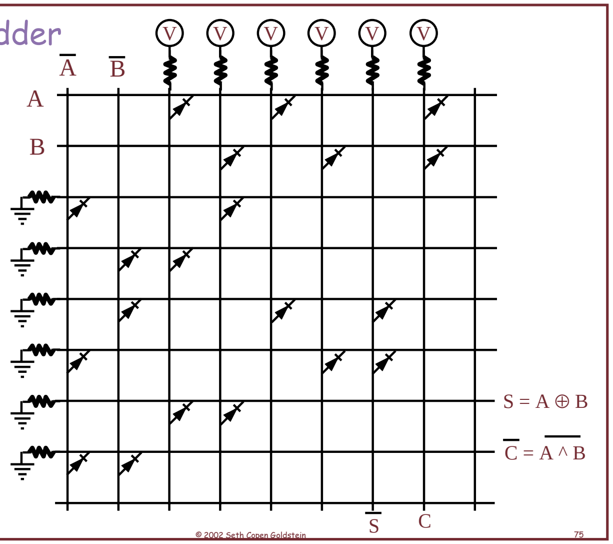
Diode-resistor Logic



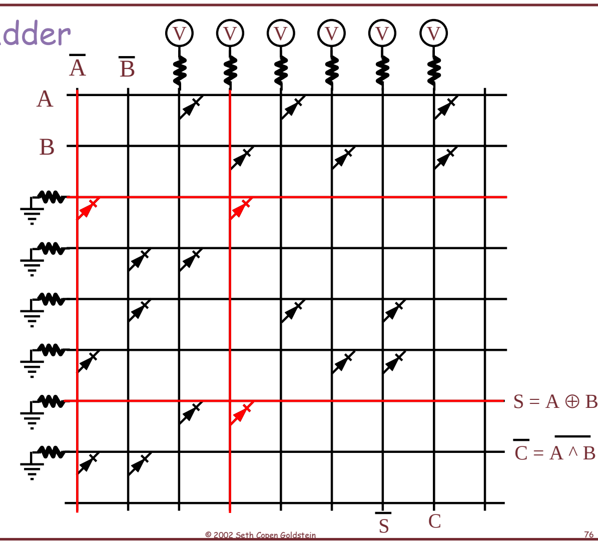
Diode-resistor Logic



Half adder



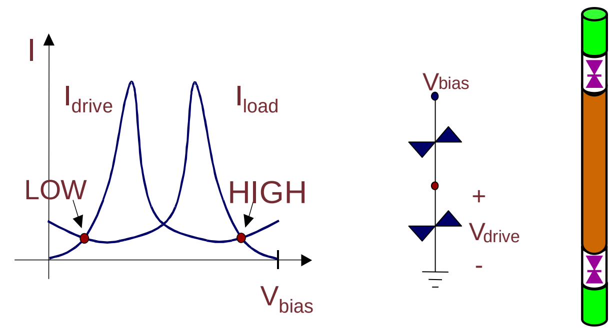
Half adder



But, Wait

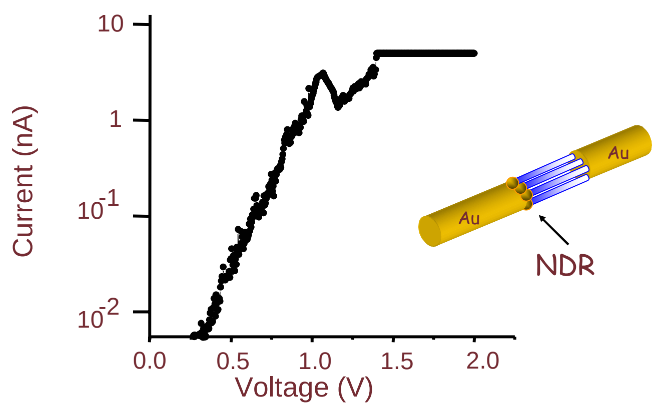
- Noise immunity?
- Signal restoration?
- Restoring logic?
- I/O isolation?
- Gain?
- Latches?

Molecular Latch Using NDRs



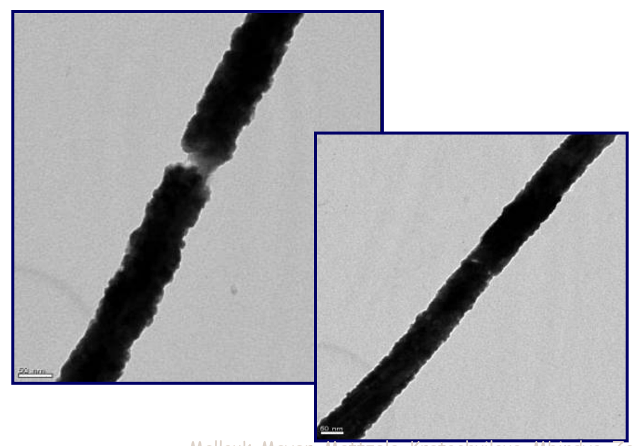
High and Low States of Bistable Latch

Recent in-wire measurments



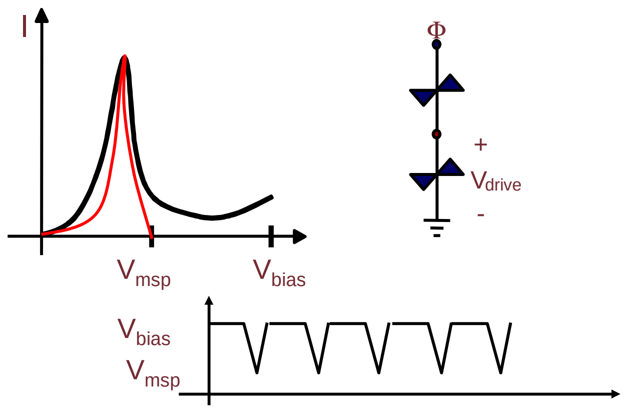
Mallouk· Mayer, Mattzela, Kratochvilova, Mbindyo, Zambova

Inwire devices

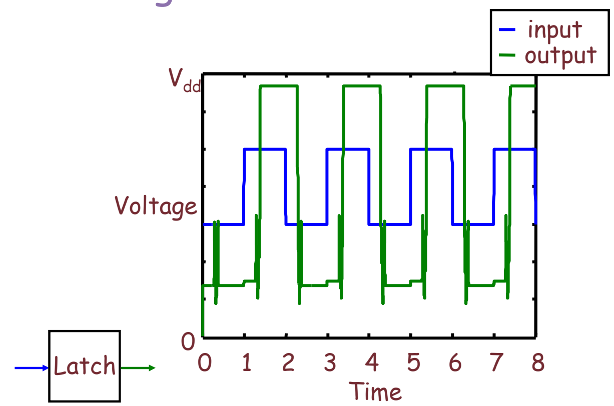


Mallouk· Mayer, Mattzela, Kratochvilova, Mbindyo, Zambova

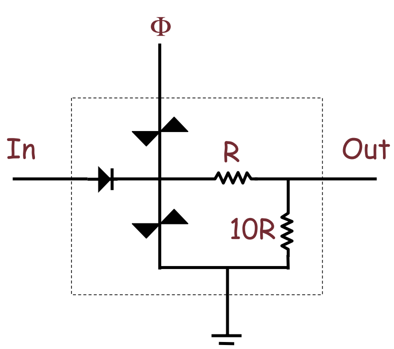
Clocking Methodology



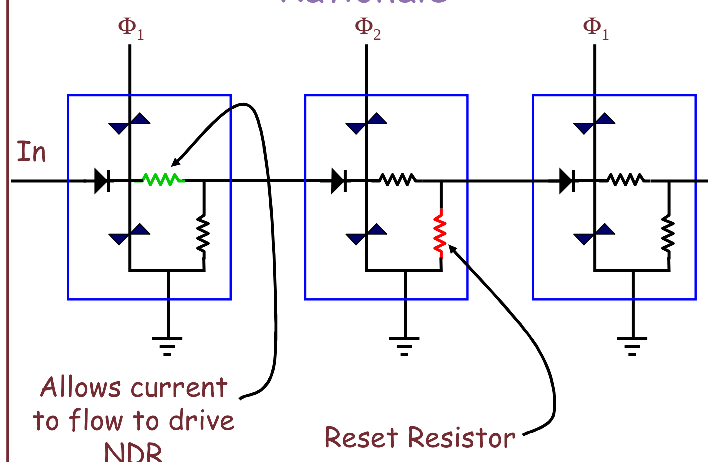
Logic Restoration Effect



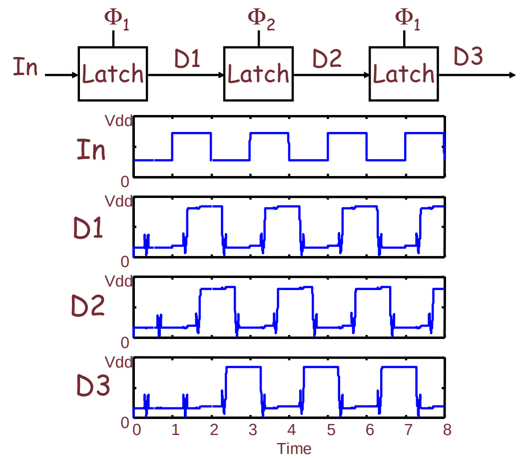
Making It Work



Rationale

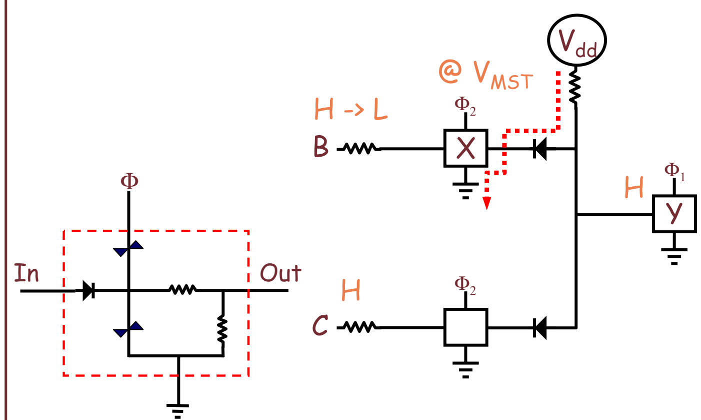


Delay Line



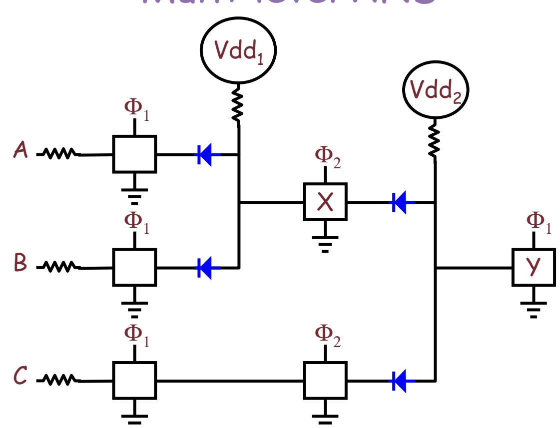
Micro tutorial © 2002 Seth Copen Goldstein 85

V_{dd} Must Be Clocked



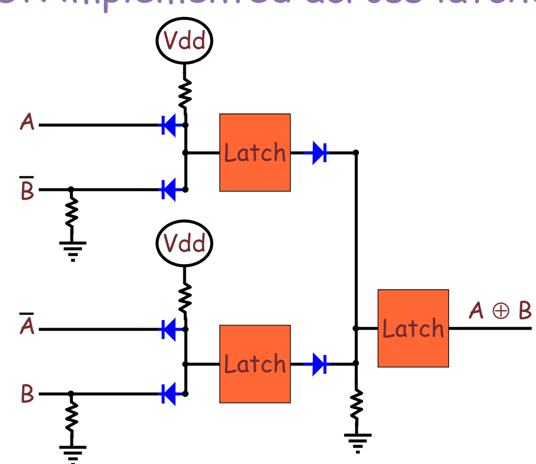
Micro tutorial © 2002 Seth Copen Goldstein 86

Multi-level AND



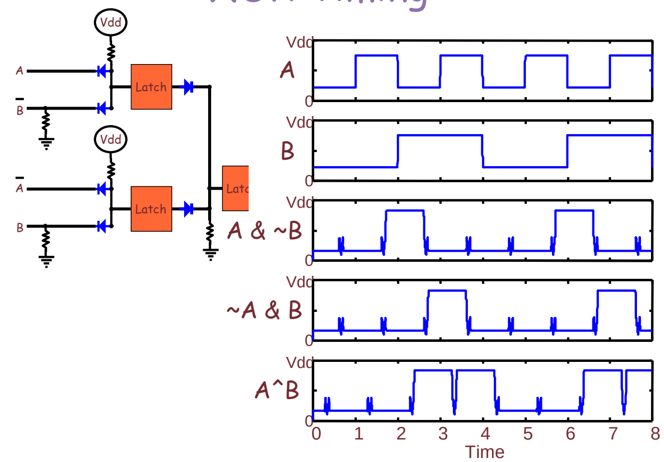
Micro tutorial © 2002 Seth Copen Goldstein 87

XOR implemented across latches



Micro tutorial © 2002 Seth Copen Goldstein 88

XOR Timing



Necessary Device Abstraction

- Transistor provides it all
  - Switch
  - Fan-out
  - I/O isolation
  - Signal restoration

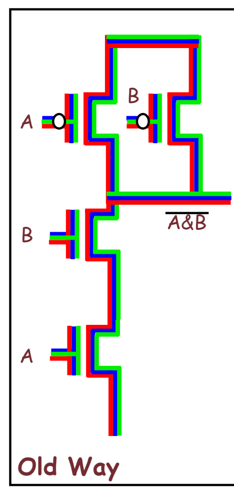
  (Switch, Fan-out: Necessary for logic; I/O isolation, Signal restoration: Necessary for large designs)
- Replace single abstraction with multiple abstractions: SIRM
- Enables an incremental path to direct nano-circuits

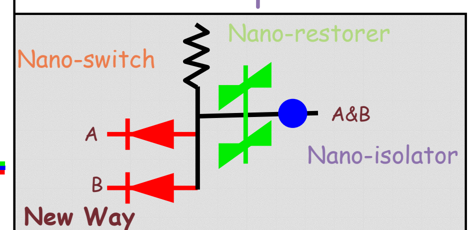
SIRM

- SIRM allows multiple devices to replace a single transistor
  - Switch
  - Isolator
  - Restorer
  - Memory

  (Switch, Isolator: Necessary for logic; Isolator, Restorer, Memory: Necessary for large designs)
- SIRM already in use
- Allows circuit designers to explicitly manage the different components of a scalable circuit technology

SIRM Example

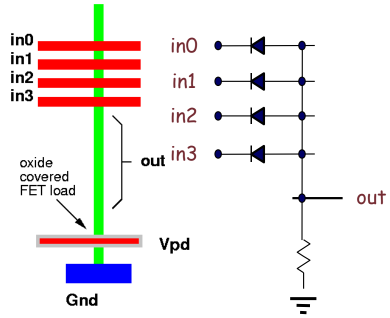




- Breaks up transistor abstraction into its components
- Allows creation of library of composable components
- Increases freedom of design

Diode Logic From Crossing NWs

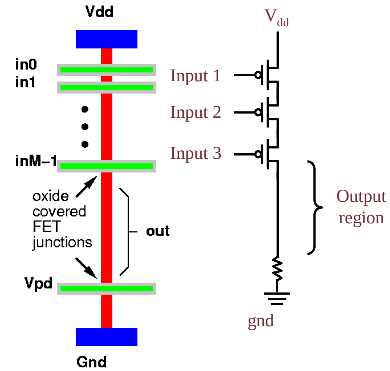
- Another example of DR-logic
- Or gate shown
- The S from SirM



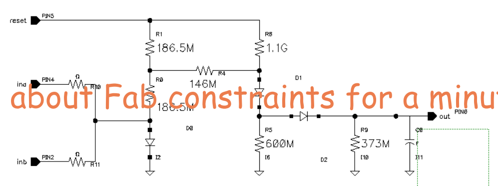
Dehon, NSC-1

PMOS-like Restoring FET Logic

- Crossing Doped NWs to build NOR
- Similar to N-MOS logic of old. (Static load)
- Sir of SirM



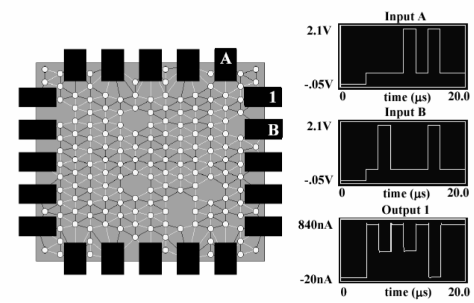
NDR-based Circuits - 1



Forget about Fab constraints for a minute.

- 2-Input Current-mode NAND gate
- Gate is first reset to place the NDR diodes in the low impedance state
- If both input currents reach threshold, the output transitions from a low impedance state to a high impedance state
- Complex, low noise immunity, poor power-delay product, little manufacturing tolerance, poor isolation

NDR-based Circuits - 2

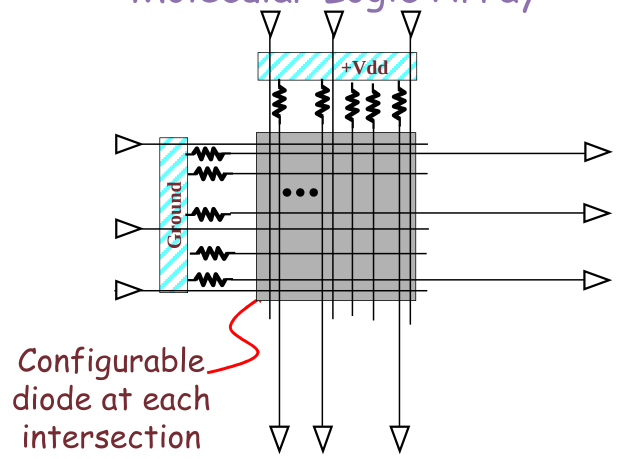


- Nanocell (Husband, Tour, Reed, et. Al.) Husband, 2002
- Easy to fabricate
- S of SirM
- Will come back to this

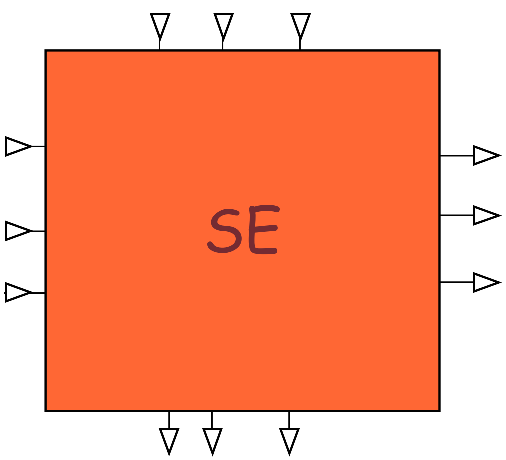
Review

- Devices
  - Diodes, resistors, programmable switches, memory devices, wires, NDRs, latches
- Fabrication Constraints
  - Regular
  - Low-information content structures
  - Defects
- Need post-manufacturing modification
- Crosspoints are cheap.

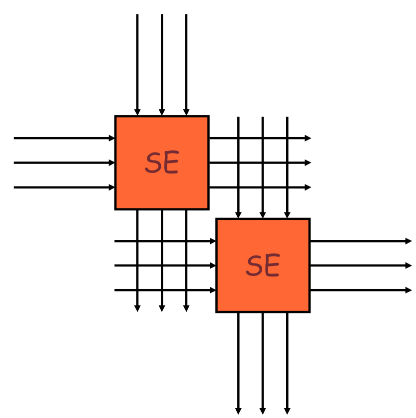
Molecular Logic Array

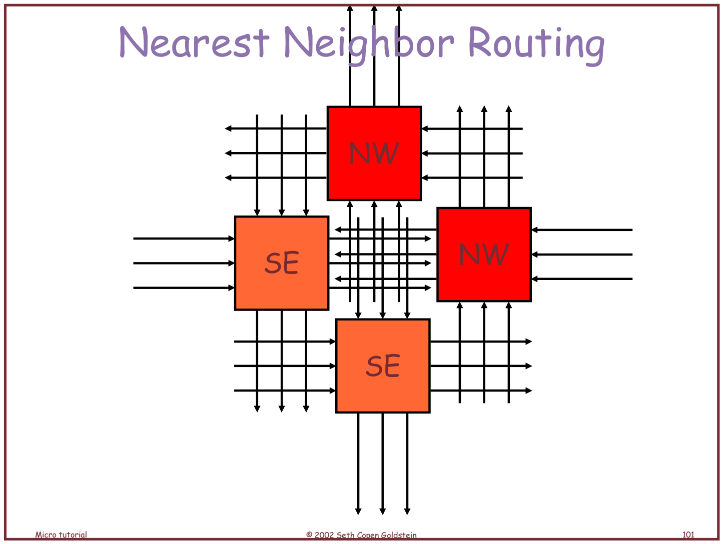


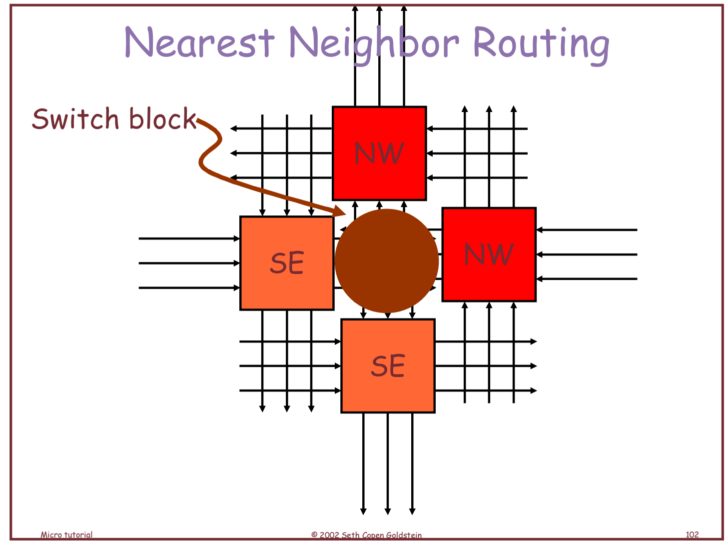
Abstract

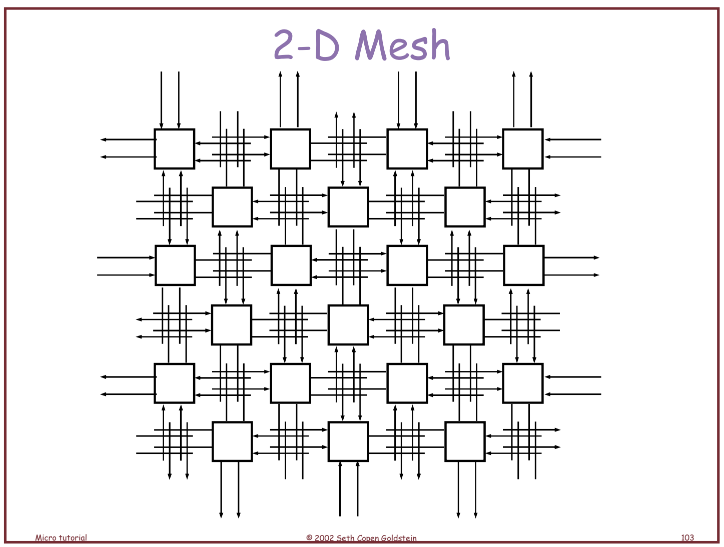


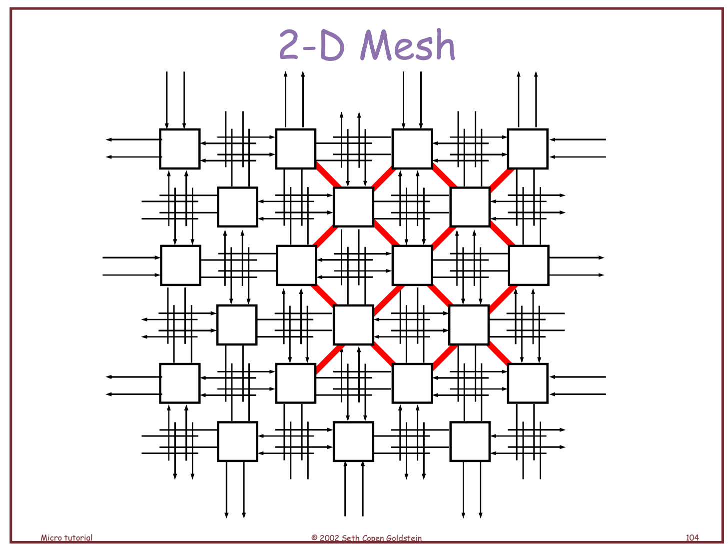
Nearest Neighbor Routing



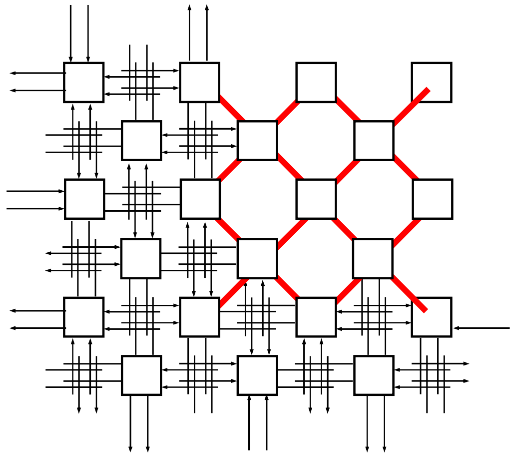

Nearest Neighbor Routing
NW
SE
NW
SE
Micro tutorial
© 2002 Seth Copen Goldstein
101


Nearest Neighbor Routing
Switch block
NW
SE
NW
SE
Micro tutorial
© 2002 Seth Copen Goldstein
102


2-D Mesh
Micro tutorial
© 2002 Seth Copen Goldstein
103

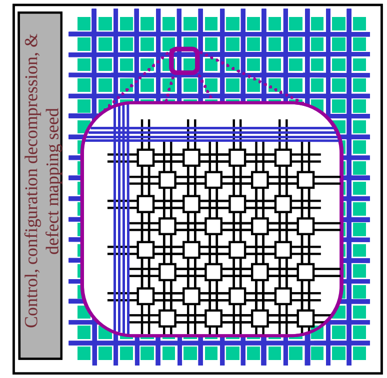

2-D Mesh
Micro tutorial
© 2002 Seth Copen Goldstein
104

2-D Mesh



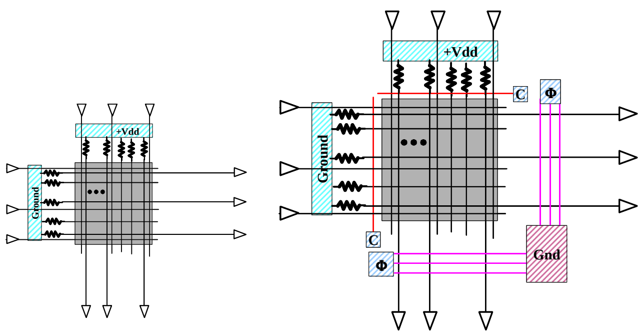
The NanoFabric

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



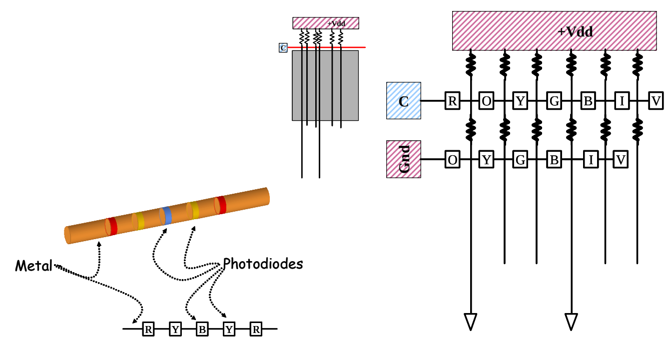
Lets Fill in some details

- Need to be able to configure crosspoints
- Need to restore logic signals



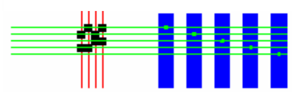
Interfacing Micro & Nano

- Nanoblock strategy uses single-wire demux
- Enables small nanoscale array

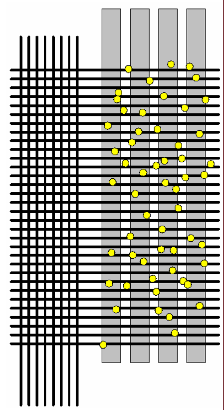


Log(N) Demuxes

- Use Log(N) Micronscale wires to address N nanoscale wires.
- Significantly raises overhead



Deterministic Dehon

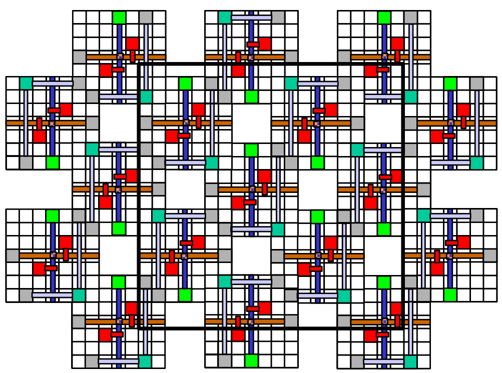


Random

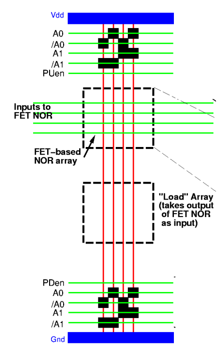
Choosing an array size

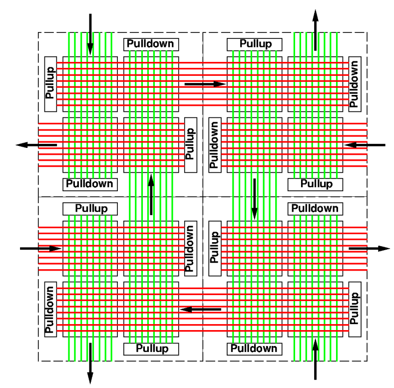
- Prefer large array
  - To amortize overhead
- Prefer small array
  - To reduce hit from defects
  - To isolate defects
  - To increase speed
  - To increase utilization

Putting Nanoblocks on CMOS



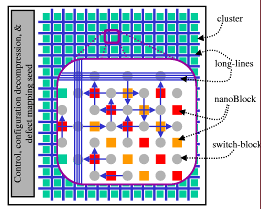
Another Array-based Architecture



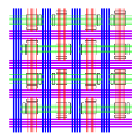


Dehon, NSC-1

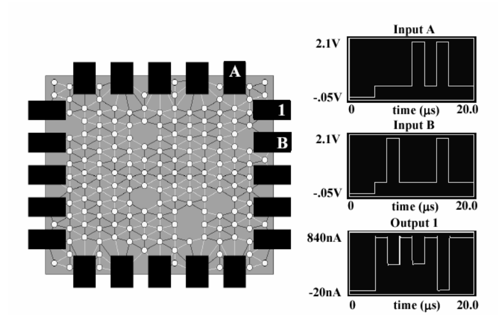
Array-based Architectures



- Both
  - Regular
  - Built on CMOS
  - Based on reconfigurable non-volatile switch
- Nanofabric
  - Does not require precise nanoscale placement
  - Uses light for programming
- ABAME
  - Requires precise placement for decoders
  - Can program electrically



Nanocell architecture



Husband, 2002

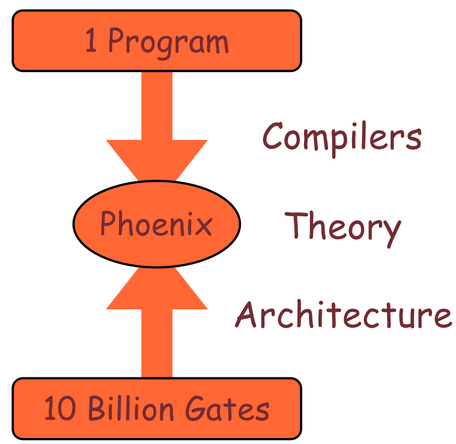
- Random deposition of particles
- Determine functionality after manufacture
- Requires significant CMOS

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

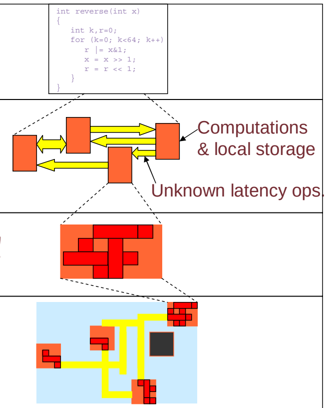
How do we use it?

Spanning 10-orders of Magnitude

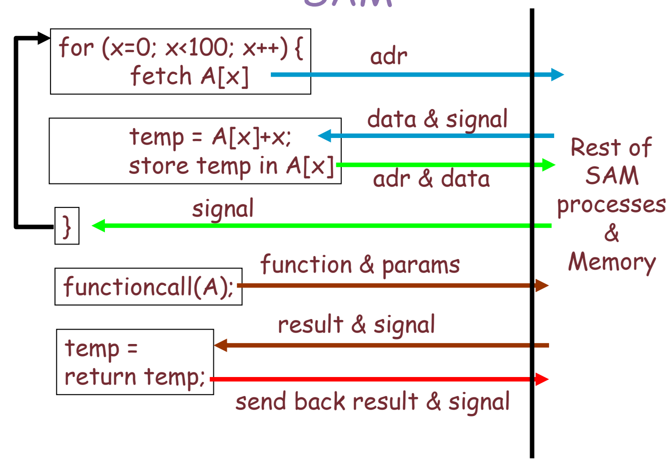


Compilation/CAD

1. Program/Circuit
2. Split-phase Abstract Machines
3. Configurations placed independently
4. Placement on chip



SAM



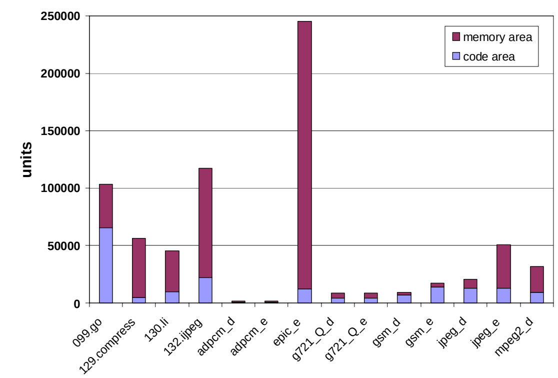
Limit Study

- We investigate using the nanoFabric as a computing substrate
- Mapping

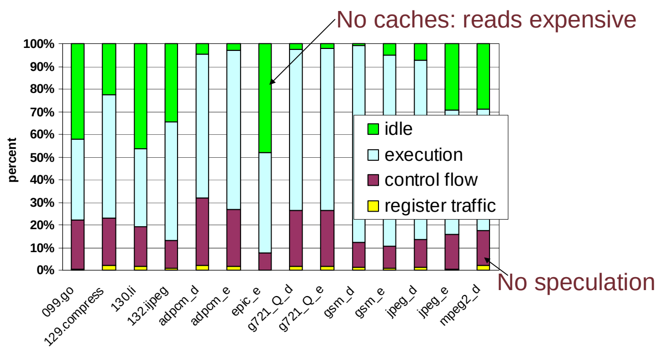
  Programs ⇒ SAM ⇒ nanoFabric
- We discover:
  - Programs and data fit on 1 cm^2 (From SpecInt95 & MediaBench)
  - Single-threaded performance is fair
  - Spatial computing requires different tradeoffs and compilation techniques

Area

(10^6 units/cm^2 available)

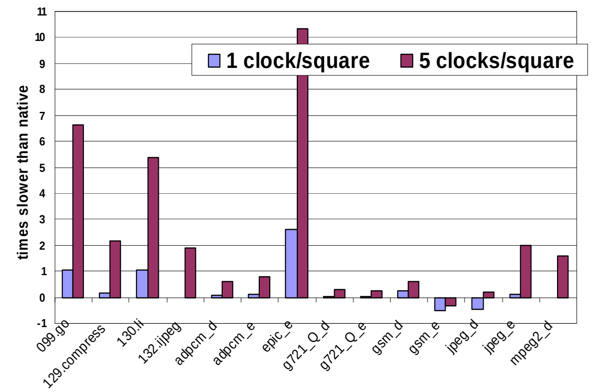


How Time Is Spent



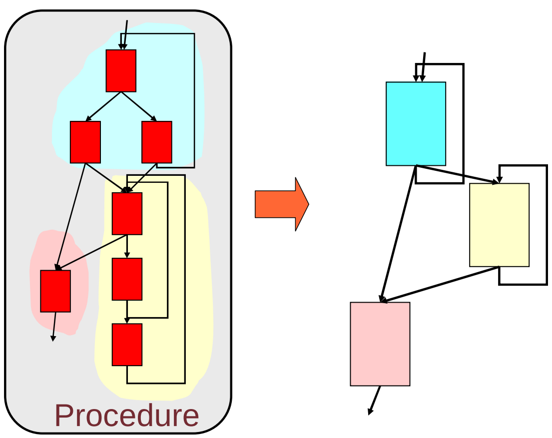
Micro tutorial © 2002 Seth Copen Goldstein 121

Application Slowdown



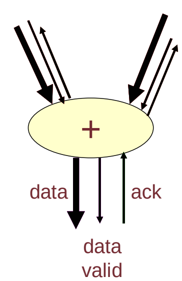
Micro tutorial © 2002 Seth Copen Goldstein 122

Hyperblocks



Micro tutorial © 2002 Seth Copen Goldstein 123

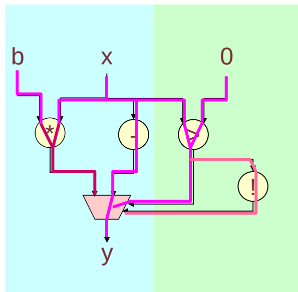
Asynchronous Computation



Micro tutorial © 2002 Seth Copen Goldstein 124

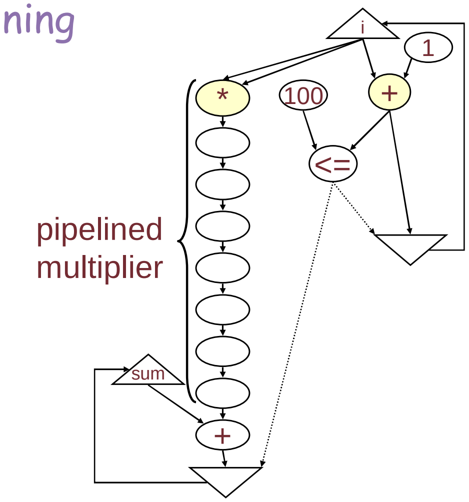
Lenient Operations

```
if (x > 0)
        y = -x;
else
        y = b*x;
```

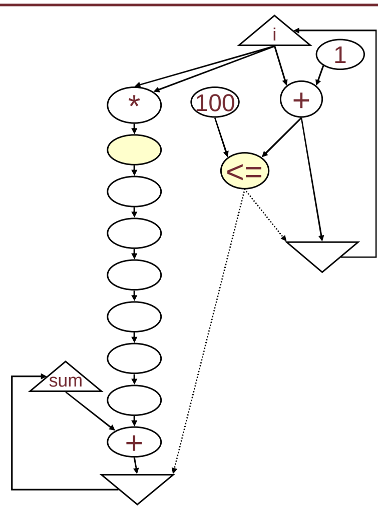


Solve the problem of unbalanced paths

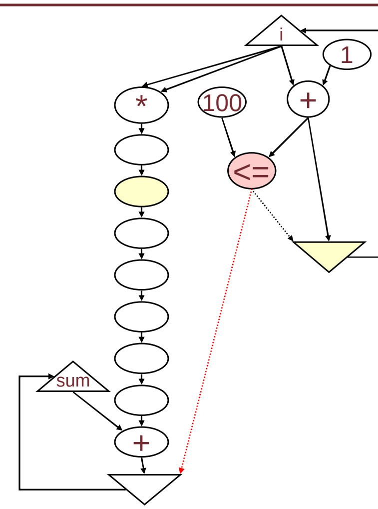
Pipelining

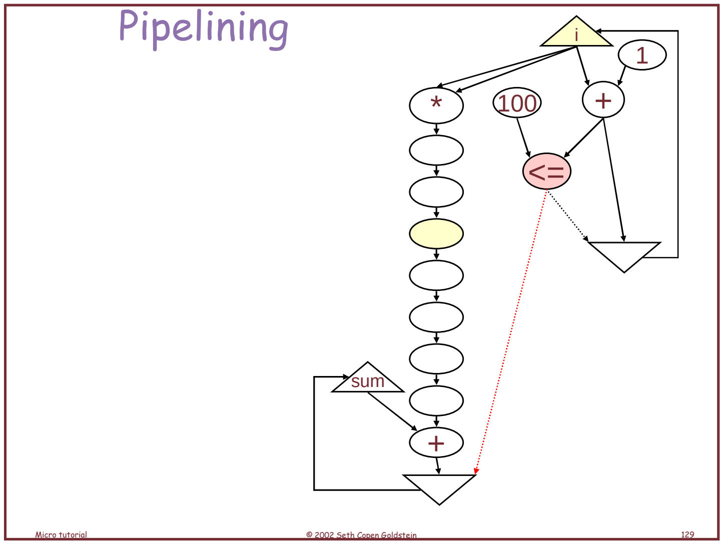


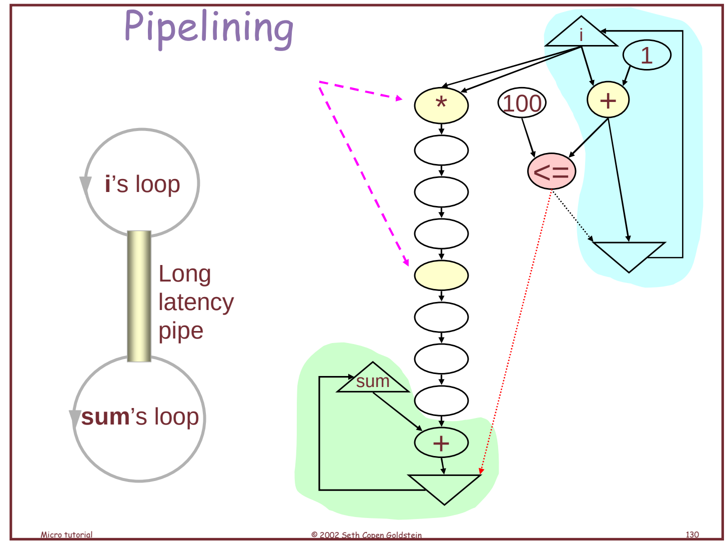
Pipelining

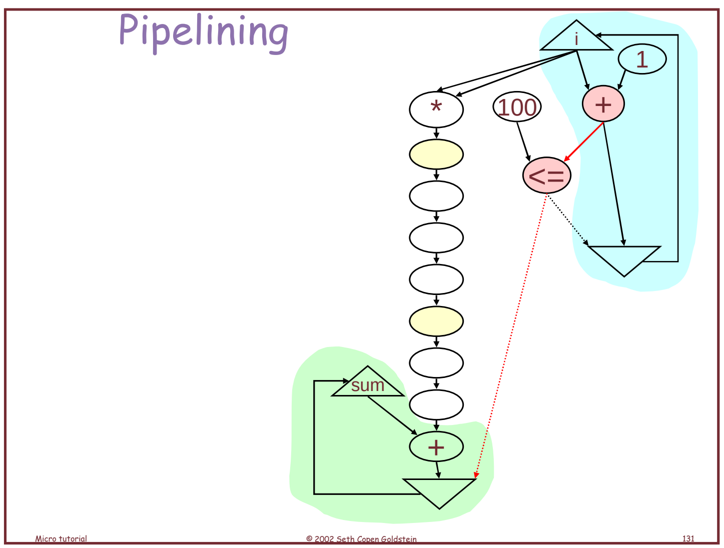


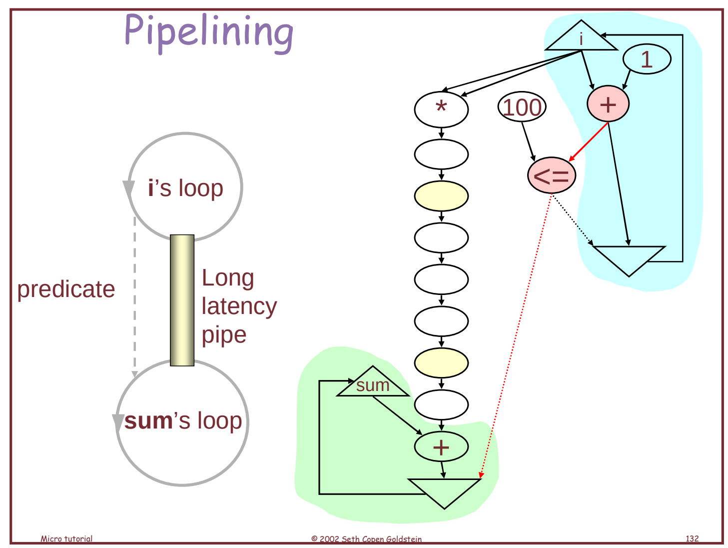
Pipelining



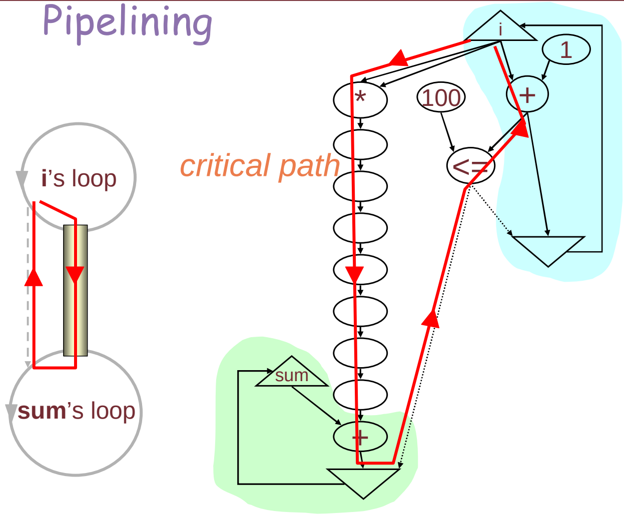

Pipelining
i
1
*
100
+
<=
sum
+
Micro tutorial
© 2002 Seth Copen Goldstein
129


Pipelining
i's loop
Long
latency
pipe
sum's loop
i
1
*
100
+
<=
sum
+
Micro tutorial
© 2002 Seth Copen Goldstein
130

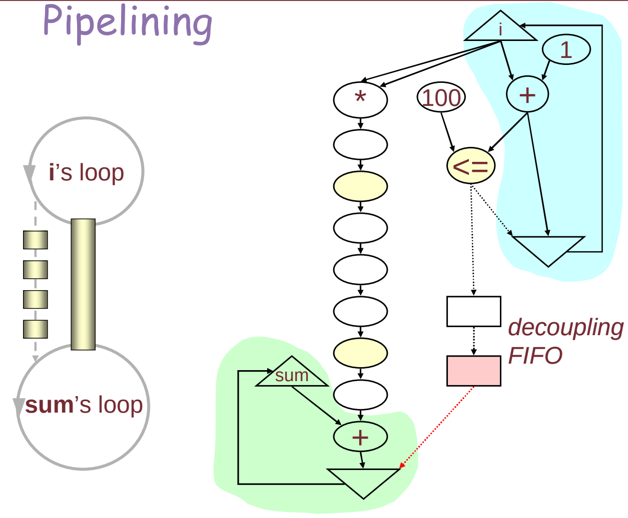

Pipelining
i
1
*
100
+
<=
sum
+
Micro tutorial
© 2002 Seth Copen Goldstein
131


Pipelining
i's loop
predicate
Long
latency
pipe
sum's loop
i
1
*
100
+
<=
sum
+
Micro tutorial
© 2002 Seth Copen Goldstein
132

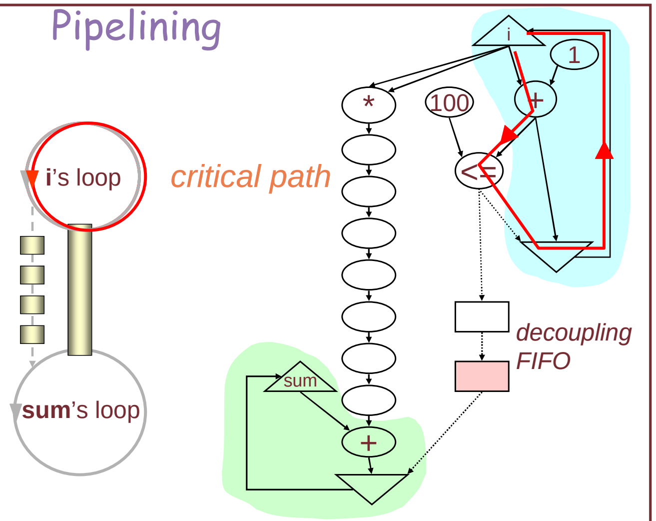
Pipelining



Pipelining

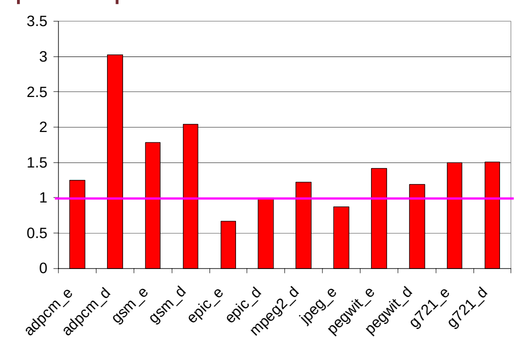


Pipelining

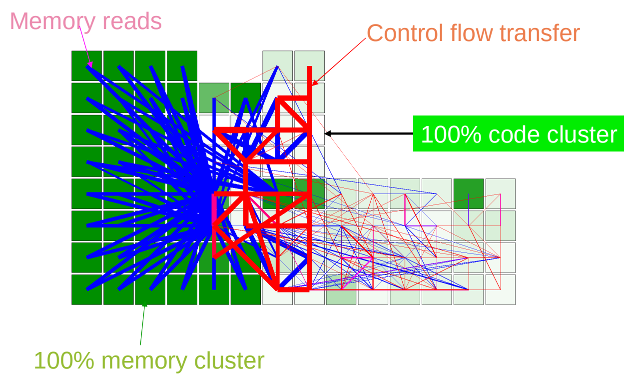


Performance

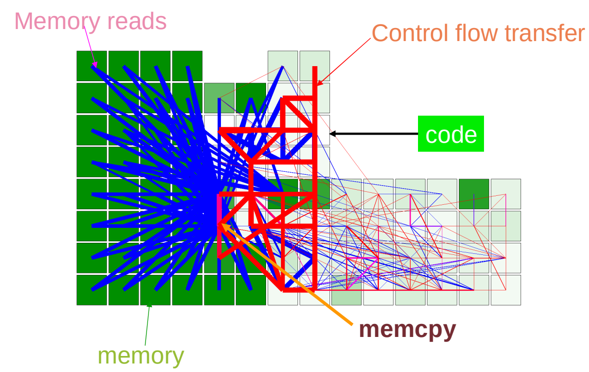
- Preliminary comparison with 4-wide OOO
- Assumed same FU latencies
- Speed-up on kernels from Mediabench



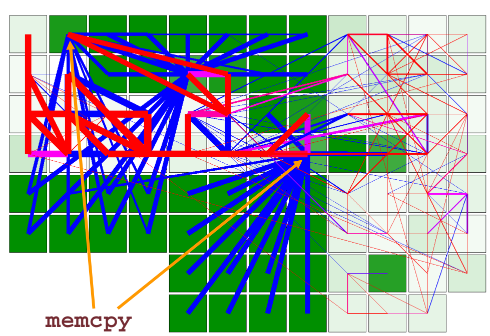
Typical Program Graph (g721_e)



Typical Program Graph (g721_e)



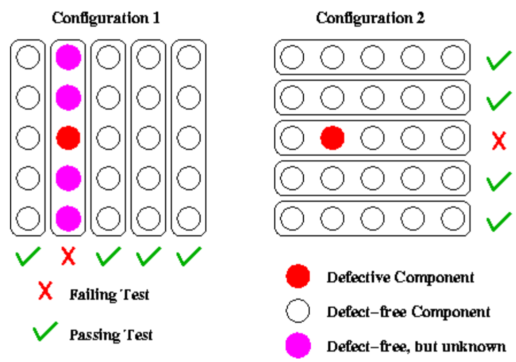
Program Graph After Inlining `memcpy`



Return to Defect Tolerance

- Expect faulty components
  - Broken wires
  - Shorted wires
  - Bad contacts
  - unprogrammable configuration bits
- Strategy
  - Locate defects
    - Must scale with defect densities
    - Must scale with number of devices
    - Cannot probe individual components
  - Place&route around them

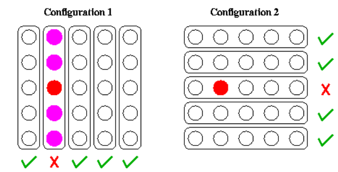
Test-circuits in action



Some terminology

- *n* components being tested
- Probability of defect *p*
- Each test circuit has *k* components
- Circuits arranged in various orientations, or *tilings*
- % of good components recovered: *yield*

In the example,
- *n*=25
- *k*=5
- 2 tilings
- yield is 100%.



Assumptions

- Permanent defects
  - defective component always displays faulty behavior
  - defect in one component does not affect others
    - **i.e., no short-circuits between wires**
    - **manufacturing process biased to ensure this**
  - no Byzantine failures

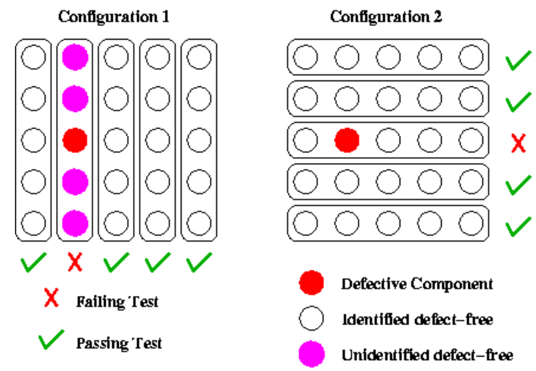
Assumptions (cont.)

- Arbitrary, unlimited connectivity
  - any component can be connected to any other, including non-adjacent ones
  - makes large number of tilings possible
- Above assumption: to simplify analysis

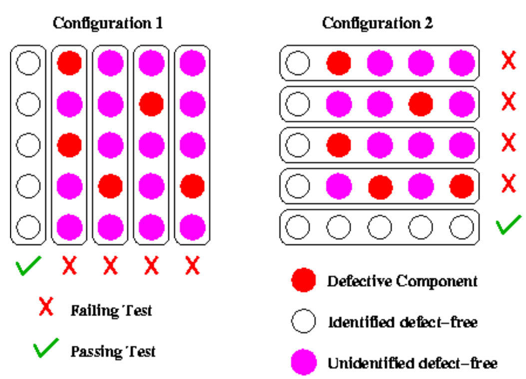
Scaling with defect density

- Expected *k*p defects/test-circuit*
- Fewer defects/circuit:
  easier to locate
- We examine the following 3 cases:
  - $k*p \ll 1$
  - $k*p \approx 1$
  - $k*p \gg 1$
- Remember, *k cannot be too small*

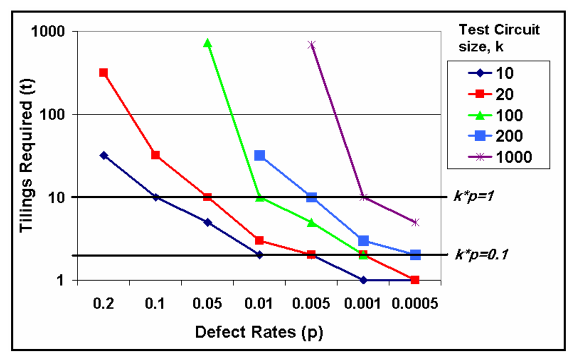
Example with very low defect rate



Example with higher defect rate



Tilings required for low defect rates



Desired yield = 99%

High defect rates: $k*p$ » 1

- Many defects/test-circuit
- Finding a defect free circuit is extremely unlikely
  - e.g., for k=100, p=0.1, probability of finding a defect-free circuit = $1.76*10^{-5}$
- The previous approach does not work: something new is needed

Making test circuits more powerful

- Use test-circuits which *count* defects
  - error in output depends directly on number of defects
  - e.g., use error-correcting, fault-tolerant circuit designs
- These can return correct counts only upto a certain threshold
  - must indicate when threshold is crossed
  - use two different test circuits simultaneously!

New testing methodology

- Split into two phases:
  - *probability-assignment* phase
  - *defect-location* phase
- First phase: identifies components with high probability of being defect-free
- Second phase: tests these components further to pin-point defects
  - each phase: uses many different tilings

Probability-assignment phase

- Each component made a part of many different test circuits and defect counts are obtained
- Find probability of each component being good using Bayesian probabilistic analysis
- Discard components with low probability of being good

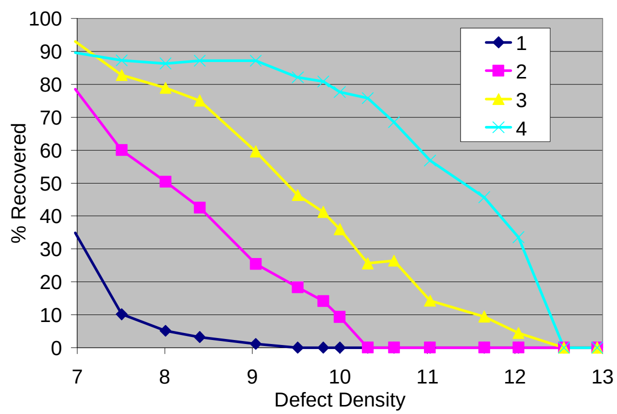
This works, but why?

- Intuitively, a defective component increases defect counts of all circuits it is a part of
- If a component is part of many circuits with a high defect count, our analysis assigns it a low probability of being good

Defect location phase

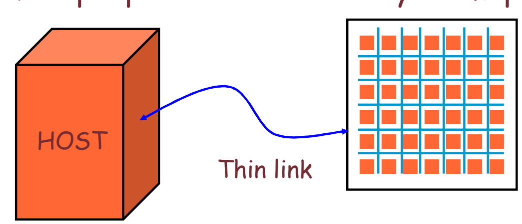
- Remaining components have low defect rate
- Configure into test circuits, mark all the components good if circuit has no defects
- Repeat for many different tilings
- Everything left is marked bad

Perfect Counting Circuits



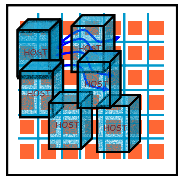
Scaling to 10^{11}

- 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}

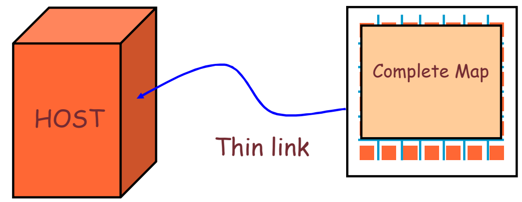
- Eliminate Host
- After initial area is found, configure chip to act as host
- At end of complete test, upload entire defect map.
- Goal: <1 day to map



Micro tutorial © 2002 Seth Copen Goldstein 157

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.
- Goal: <1 day to map



Micro tutorial © 2002 Seth Copen Goldstein 158

Predicting the Future

Imagine Computers:

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

Computers from atomic scale components

Micro tutorial © 2002 Seth Copen Goldstein 159

Summary

- Significant progress in nanoscale
  - Devices
  - Wires
  - Fabrication
- Independent of Technology
  - Loss of arbitrary patterns at fab time
  - Increase in defects and faults
  - Reconfigurable fabrics look good
- Molecular electronics is reconfigurable friendly

Micro tutorial © 2002 Seth Copen Goldstein 160

Plenty of Interesting Research

- Devices Abstractions
- Circuits
- Architectures
- Cad Tools
- Compilers
- Defect Tolerance
- Fault Tolerance