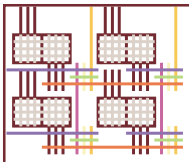


Array-Based Architecture for Molecular Electronics

André DeHon
<andre@cs.caltech.edu>

Molecular Scale Building Blocks

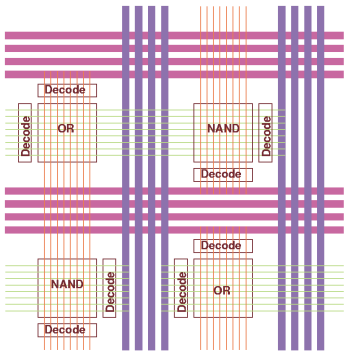
- Chemists starting to demonstrate
  - Molecular-scale switching devices
  - Molecular-scale wires
  - Regular assembly techniques for these components

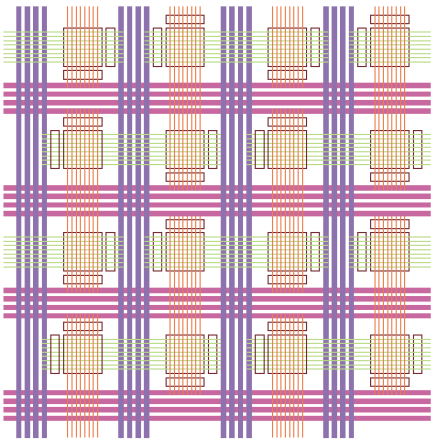
Computing?

- **Problem:** How can we build computing devices from this technology?
  - Organize into useful connectivity?
  - Provide signal restoration?
  - Personalize/Engineer to perform specific computation?
  - Accommodate defects?

Architecture Goal

- This talk:
  - Starting with these building blocks
  - Show how to build universal, programmable computing array.



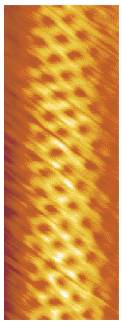


Outline

- Review Components
- Restoring Logic Style
- Bootstrap Programming
- Routing
- Defect Tolerance
- Summary

Wires

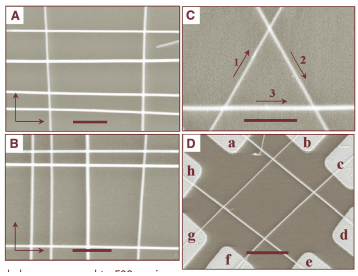
SWNT (Single Wall Carbon Nanotubes)



- Nanometer(s) diameter
- microns long
- good conductors

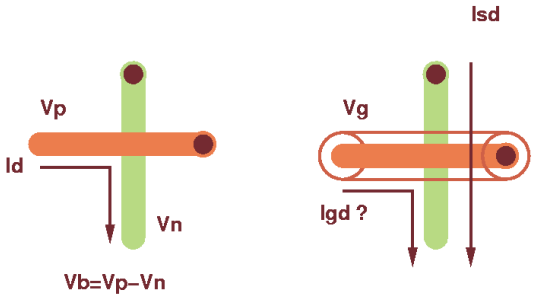
SiNW (Silicon Nanowires)

- Dope to control electrical properties

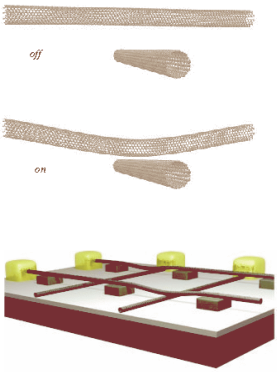


Devices

Diode and FET Junctions



Electrostatic Switches



DeHon February 2002

Unique Characteristics

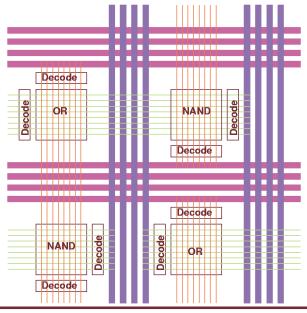
- Can only build very regular structures at nanoscale
  - Arrays of crossed tubes / wires
- Will have many defects
- Can store state of switch in wire crossing
  - Contrast with VLSI where switch >> wire xing
- Switching occurs at tube/wire crossing
  - Not at substrate…long term 3D opportunity

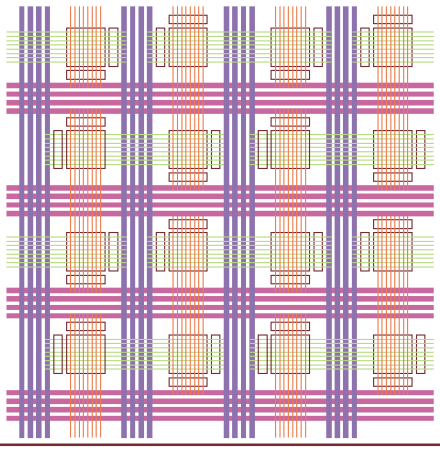
DeHon February 2002

Strategy

Strategy

- Arrays of FETs for gain
- FET decode for bootstrap program
- PLA logic in arrays
- Overlapping wires for interconnect

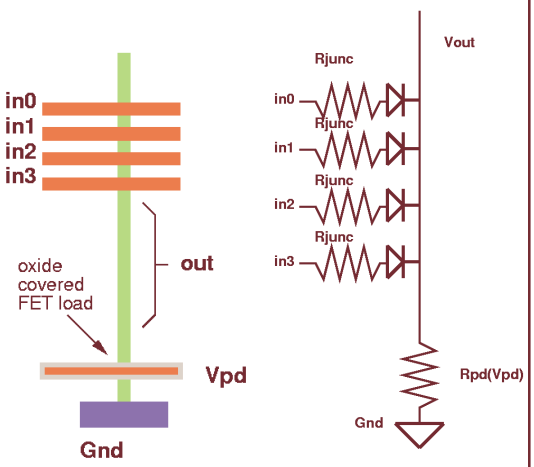




Logic Discipline

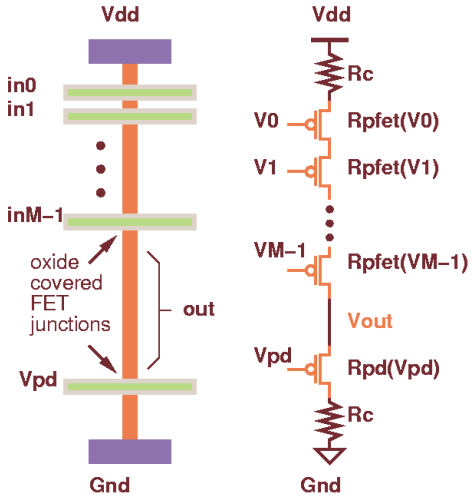
Diode Logic

- Arise directly from touching NW/NTs
- Passive logic
- Non-restoring



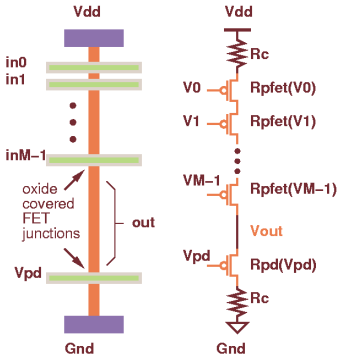
PMOS-like Restoring FET Logic

- Use FET connections to build restoring gates
- Static load
  - Like NMOS (PMOS)



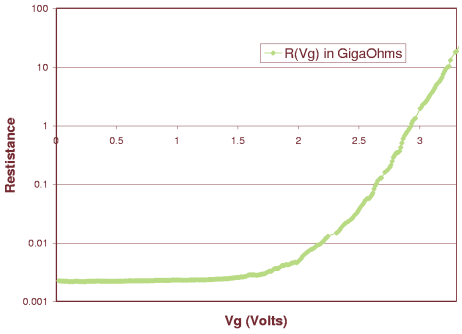
PMOS-like Logic

$$V_{out} = V_{dd}\left(\frac{R_{pd}+R_c}{R_c+\sum_{i=0}^{M-1}\left(R_{pfet}(V_i)\right)+R_{pd}+R_c}\right)$$



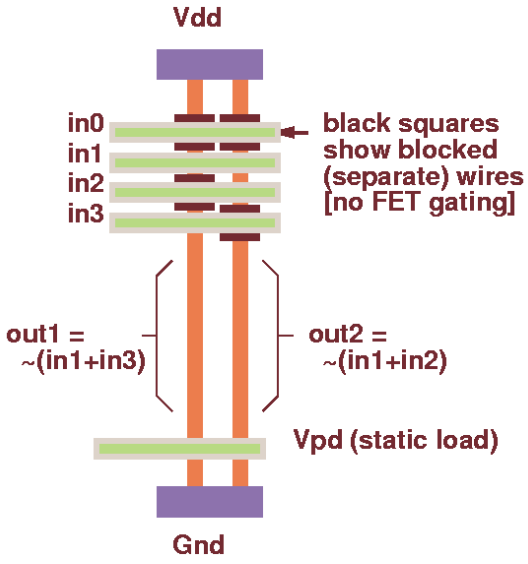
Operating Point

$$V_{out} = V_{dd}\left(\frac{R_{pd}+R_c}{R_c+\sum_{i=0}^{M-1}\left(R_{pfet}(V_i)\right)+R_{pd}+R_c}\right)$$



V_{dd}	3.3V
V_{oh}	3.0V
V_{ih}	2.8V
V_{il}	0.5V
V_{ol}	0.15V
V_{pd}	2.4V

Programmed FET Arrays

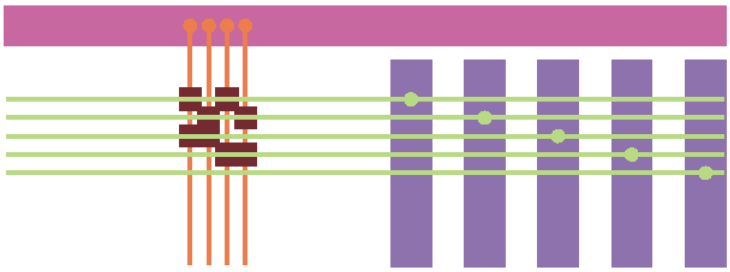


Interfacing and Programming

Micro→nanoscale

FET Decoders

- FETs also ideal for decoding/drive at nano-micro interface

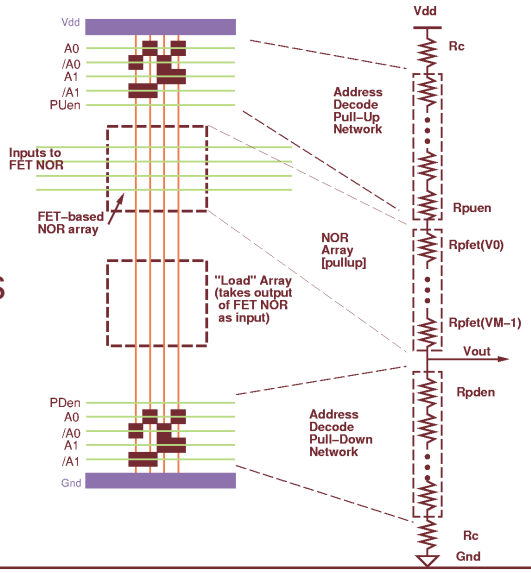


Program Decoder

- Will need to program the decoder
  - Different scheme; in fabrication
  - Imprinting/stamping

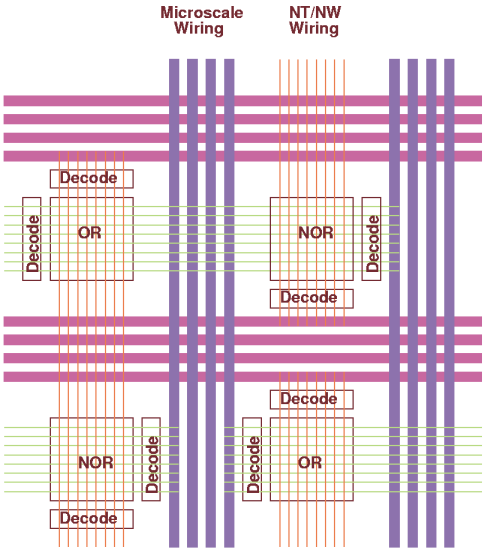
Operating Array

- Decoders allow program array
  - OR, NOR
- Isolatable
- Dual role of loads during operation
- Output used directly by consumer



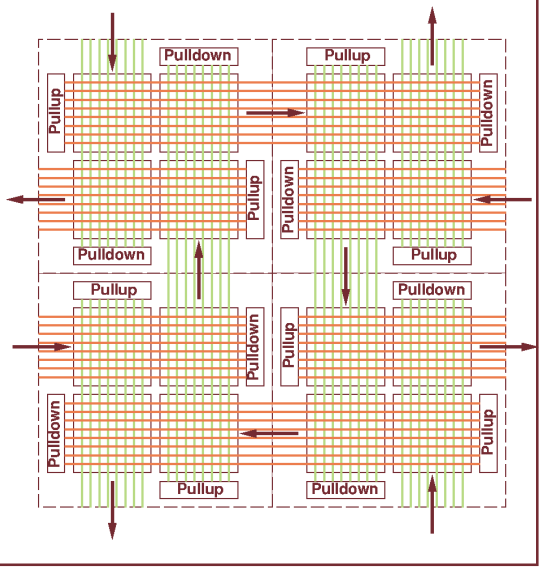
Assembly

- Overlapping NW/NT between arrays provide interconnect
- NOR only sufficient
- Alternate:
  - Programmable (non-restoring) OR
  - followed by fixed (restoring) NOR

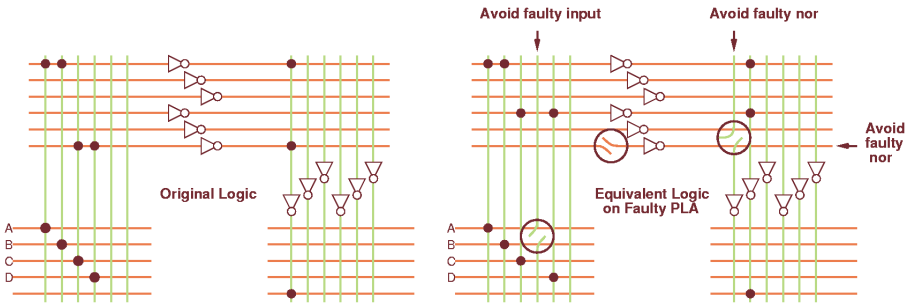


Routing

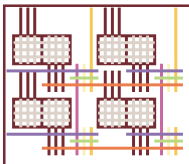
- X-Y, mesh routing with appropriate tile overlap



Defect Tolerance



All components (PLA, routing) interchangeable;
Allows local programming around faults



Summary

- Universal, Programmable Architecture
  - Built entirely from large arrays of crossed NT/NWs
- Provides restoration and inversion entirely at nanoscale
- Support nanoscale bootstrap programming
- Designed to tolerate defective components