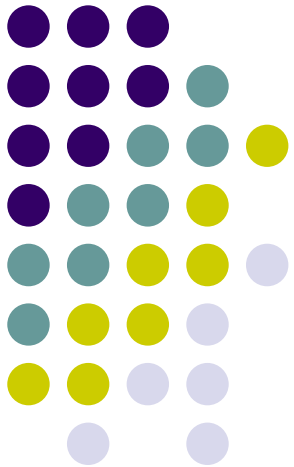
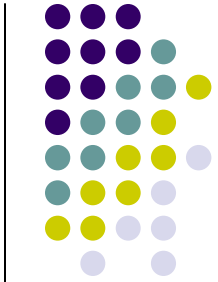
CSCI 2570 Introduction to Nanocomputing

Reconfigurable Computing

John E Savage



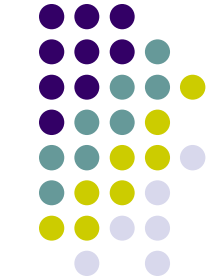
Overview



- Introduction to reconfigurable computing.
- Reconfiguration at the nanoscale.

Figs. On pgs. 4,16-18, 24, 25, 27 were created by A. Dehon.

The Current Hardware Landscape

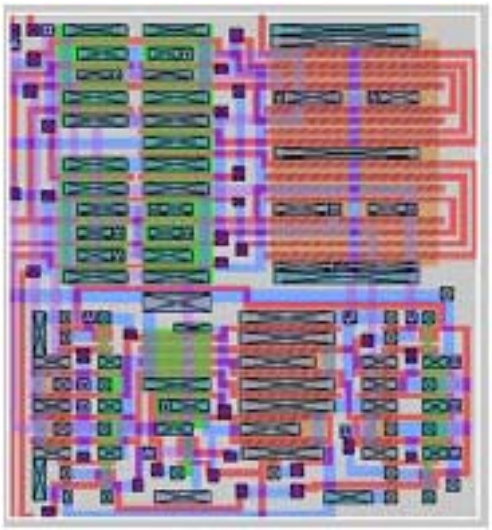


- Very large scale integration (VLSI) yields chips with lots of components

- The role of configurable chips is increasing.
  - Application specific integrated circuits (ASICs) allow engineers to design chips for manufacture from standard parts.
  - Field-programmable gate arrays (FGPAs) allow engineers to program a chip after manufacture.

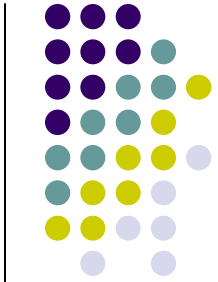
Very Large Scale Integration (VLSI)

- > 100,000,000 transistors/chip today



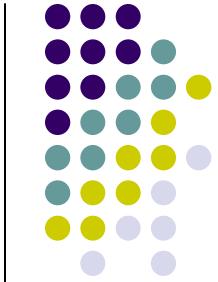
- 2.5×10^{11} λ^2 units of area available on a chip.
  - λ = feature size (minimal wire width and separation) (λ = half-pitch)
  - 3D not fully exploited
- Designing, verifying, making, and testing a big chip is very

CPUs and ASICs



- CPUs are expensive, carefully designed, general-purpose computing elements.
  - CPU speed has stopped doubling every 2-3 years
  - Future growth will be spatial and 3D
- ASICs are
  - Inflexible but efficient in area use
  - Require long turn-around times
  - Get designed once and then are manufactured

Reconfigurable Computers



- Computers in which the hardware can be re-programmed during or just before execution.

- This 1960s concept was not used extensively until the amount of re-programmable hardware on a chip was large enough.

Why Use Reconfigurable Computing?

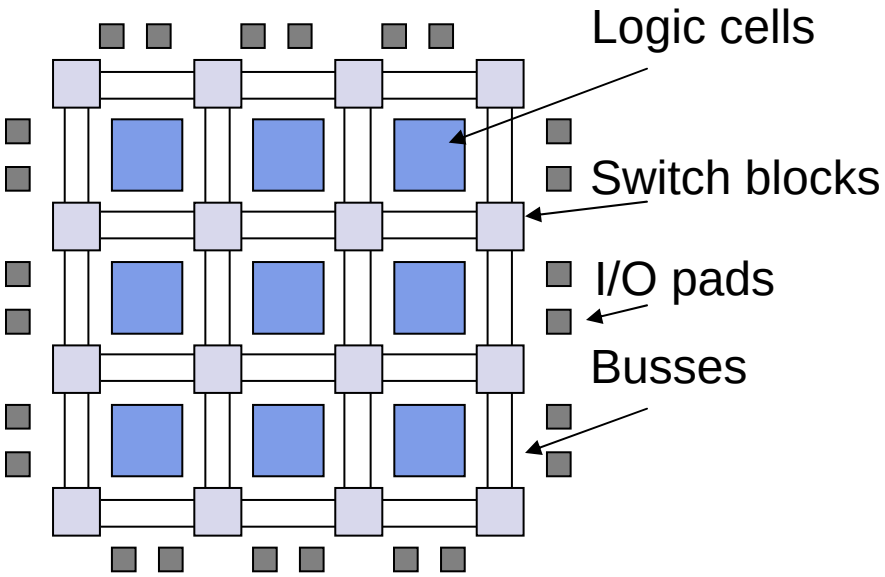
- CPUs have lots of general-purpose hardware
- Some hardware can be bypassed through reconfiguration.
  - Note: Reconfiguration introduces its own overhead
- In the nanoscale world, reconfiguration is needed to handle uncertainty and errors.

Field-Programmable Gate Arrays (FPGAs)

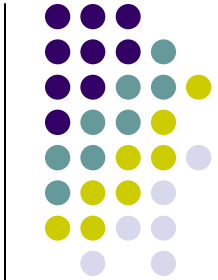
- Array of programmable logic cells and interconnect – horizontal and vertical busses connected by programmable switches.

- FPGAS may contain processors and RAM.

- FPGAs may use more power than ASICs

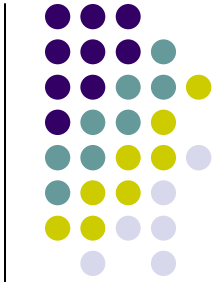


FPGA Applications



- Digital signal processing
- Internet routing
- Medical imaging
- Cryptography
- Computer vision

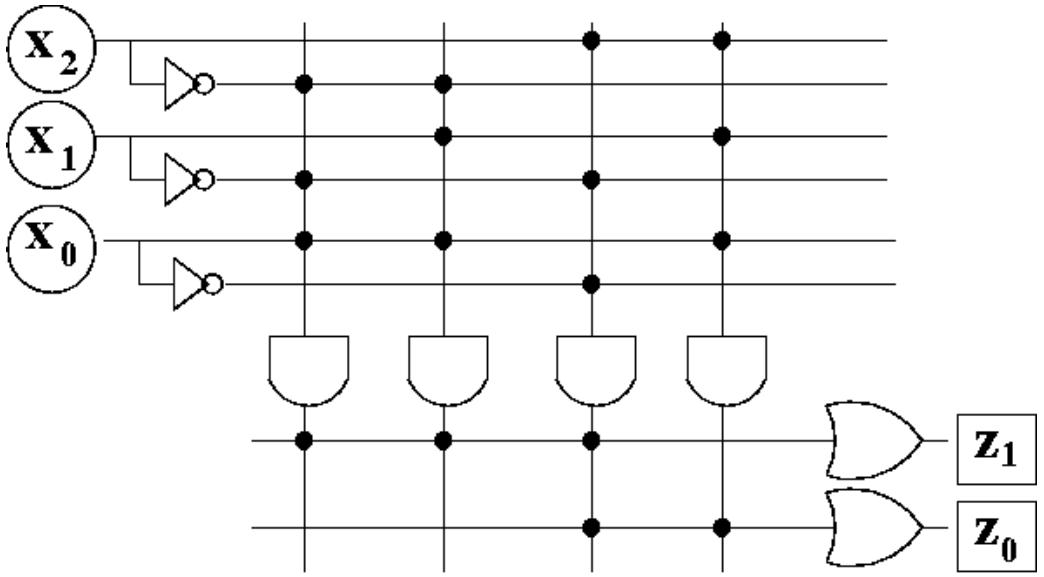
Reconfigurable Hardware



- Programmed logic arrays (PLAs)
  - AND/OR planes with programmable junctions
- Lookup tables (LUTs)
  - Small memories contain mappings defining binary functions.
- Logic cells
  - LUTs plus some memory and enabling logic

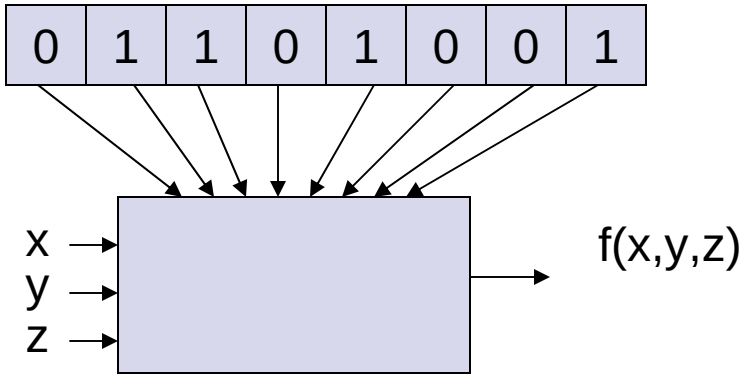
Programmed Logic Arrays (PLAs)

- Form minterms in AND plane
- Combine them in an OR plane.
- Simplification to sum-of-products possible.



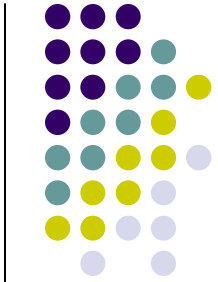
Programmable Logic Lookup Tables (LUTs)

- FPGAs contain many LUTs
  - A LUT for *k*-input gates has register holding 2^k bits as well as a multiplexer that selects one bit based on the value of the *k* inputs.

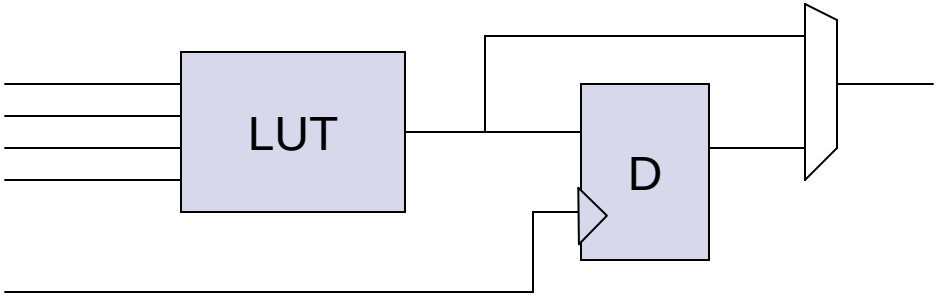


x	y	z	f(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Logic Cell

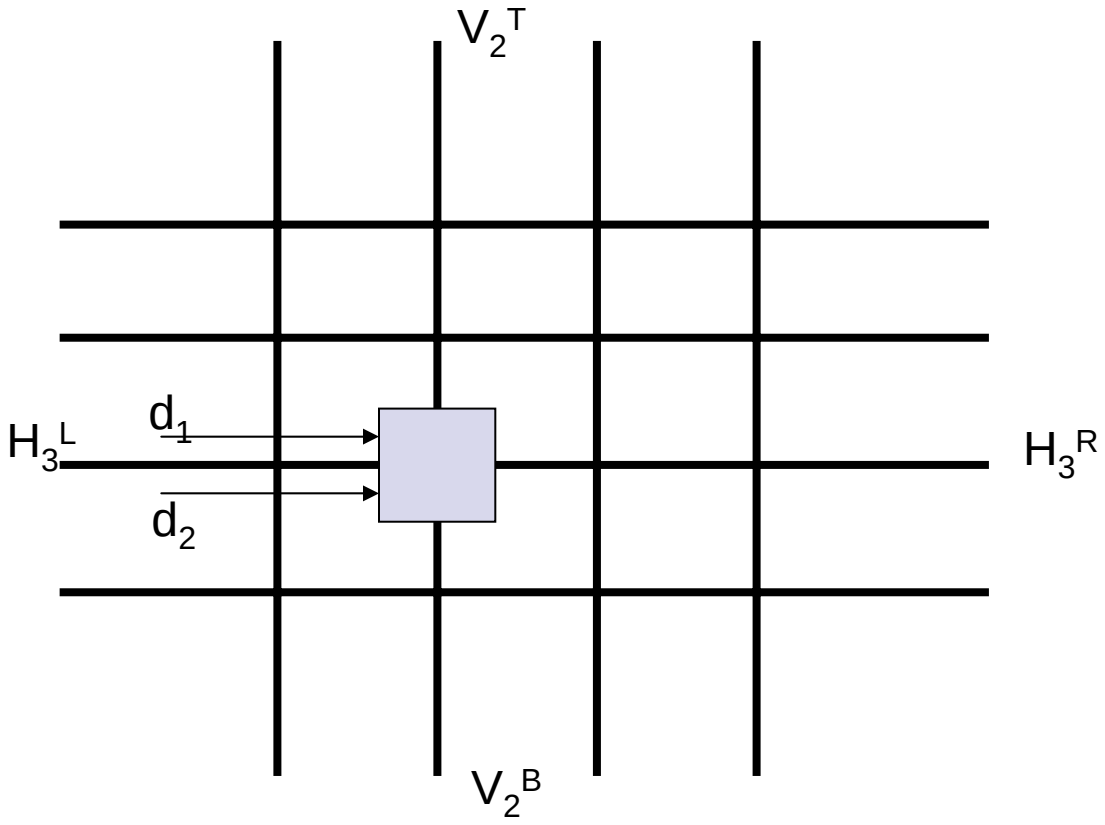


- A typical logic cell has one LUT, a D-type flip-flop and a 2-1 mulitplexer to circumvent the flip-flop, if desired.

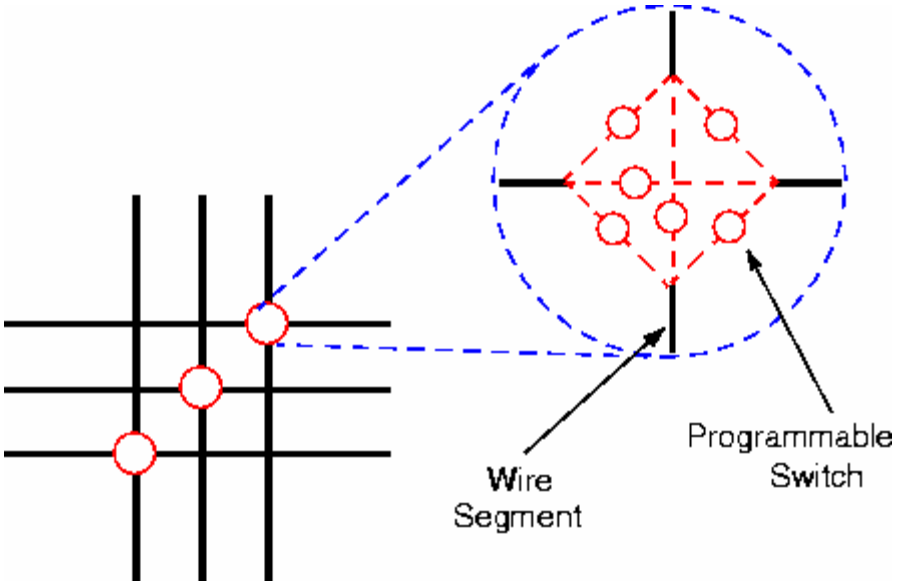


Example of Programmable Interconnect

- Set d_1 and d_2 so that H_3^L is sent to either V_2^T, H_3^R, or V_2^B.
- Add flip-flops to hold values of d_1 and d_2.
- Repeat at many intersections.

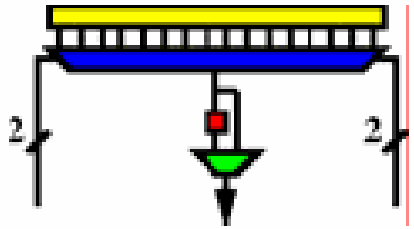


Typical Switch Box Topology

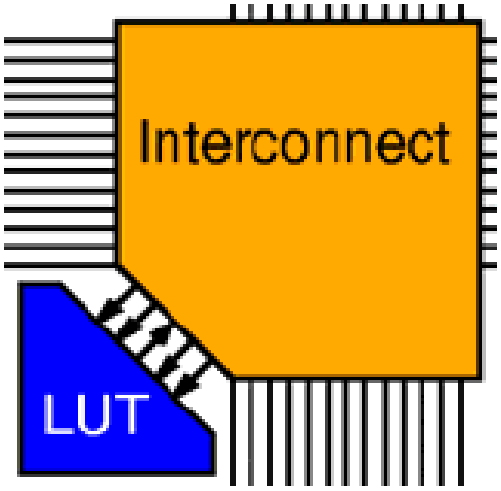


PLA Tile

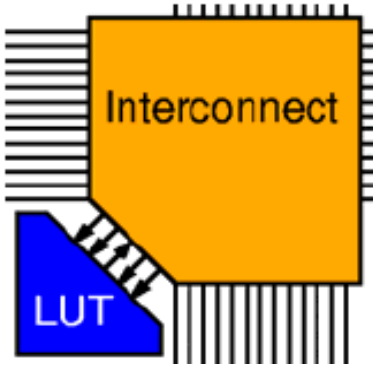
- K-LUT (k = 4) with optional output flip-flop

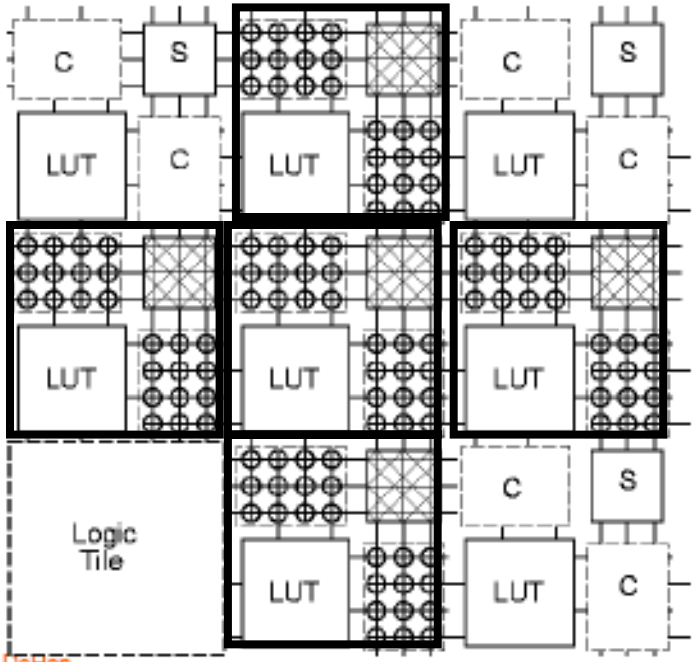


- Tile has LUT and programmable interconnect

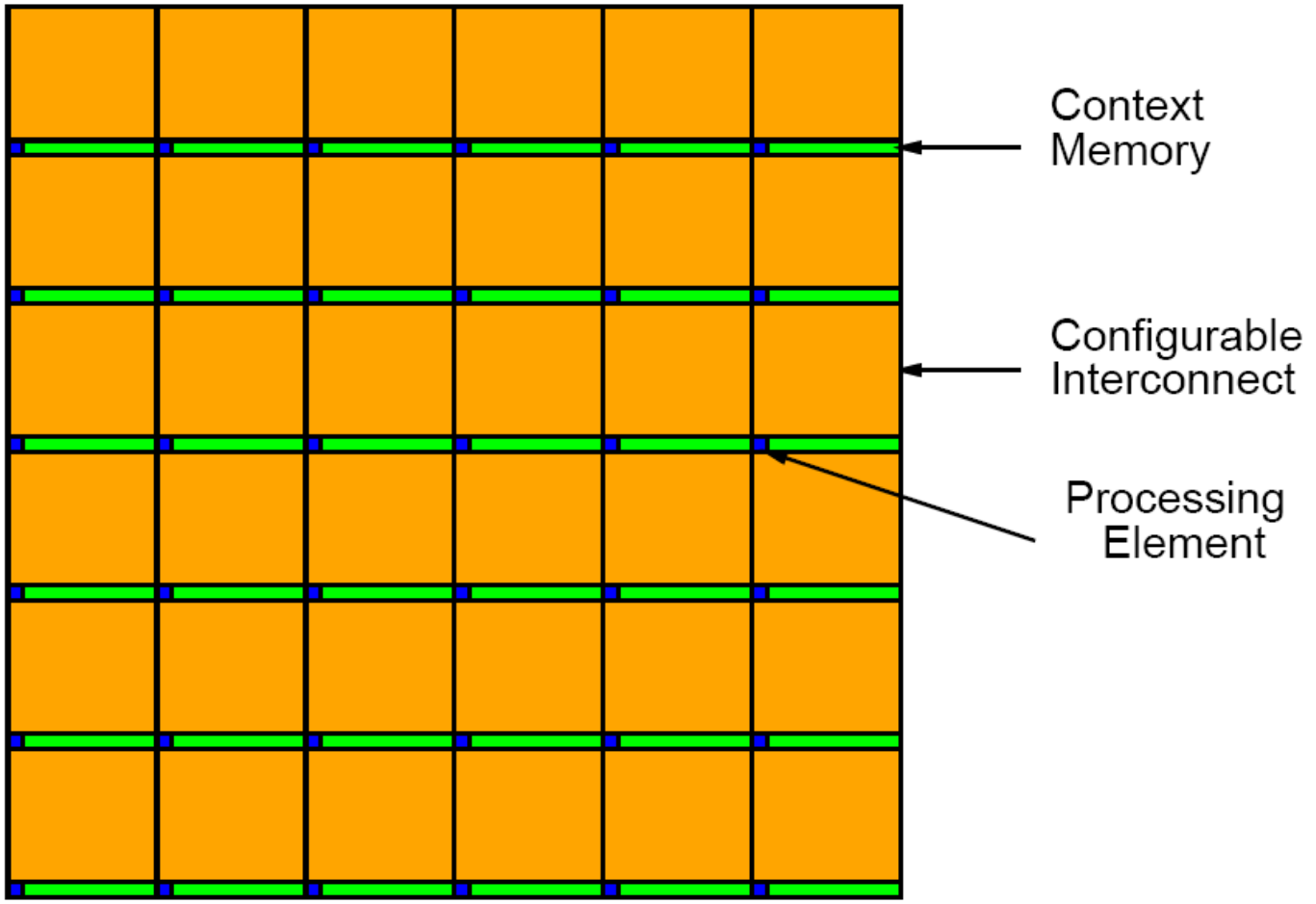


Toronto FPGA Model

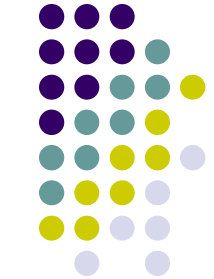




Programmable Interconnect Dominates Area in FPGAs



Mapping of Hardware Descriptions to FPGAs



- Write program in hardware design language (HDL)
  - HDL uses logical expressions and registers (arrays of flip-flops)
- HDL translated to register transfer level (RTL) descriptions
  - Circuits and control signals for registers
- RTL descriptions are mapped to hardware so as to minimize either area, speed, or power.

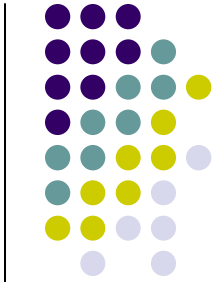
Mapping of Hardware Descriptions to FPGAs

- **Placement problem**: map gates and flip-flops to specific FPGA cells.
- **Routing problem**: choose paths for wires between gates and flip-flops.

- These are NP-hard problems.
  - Regular FPGA structure introduces inefficiencies

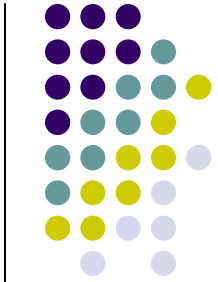
- Other issues: design verification, fault testing

Design Issues



- For what type of problems are FPGAs best?
- What granularity is appropriate for logic cells?
- How flexible should the switchboxes be?
- How wide should the busses be?
- What other components should be on chip?
- How efficient are FPGAs relative to ASICs and custom design?

FPGAs Today

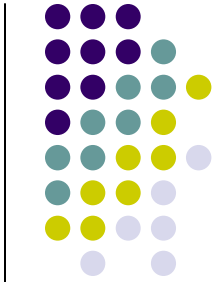


- 100,000s of LUTs
- Memories and processors embedded
  - Data rates of 10 Terabits – bandwidth 10-100 x
- Clock rate is 100s of MHz
- Will continue to get bigger but speed not likely to increase

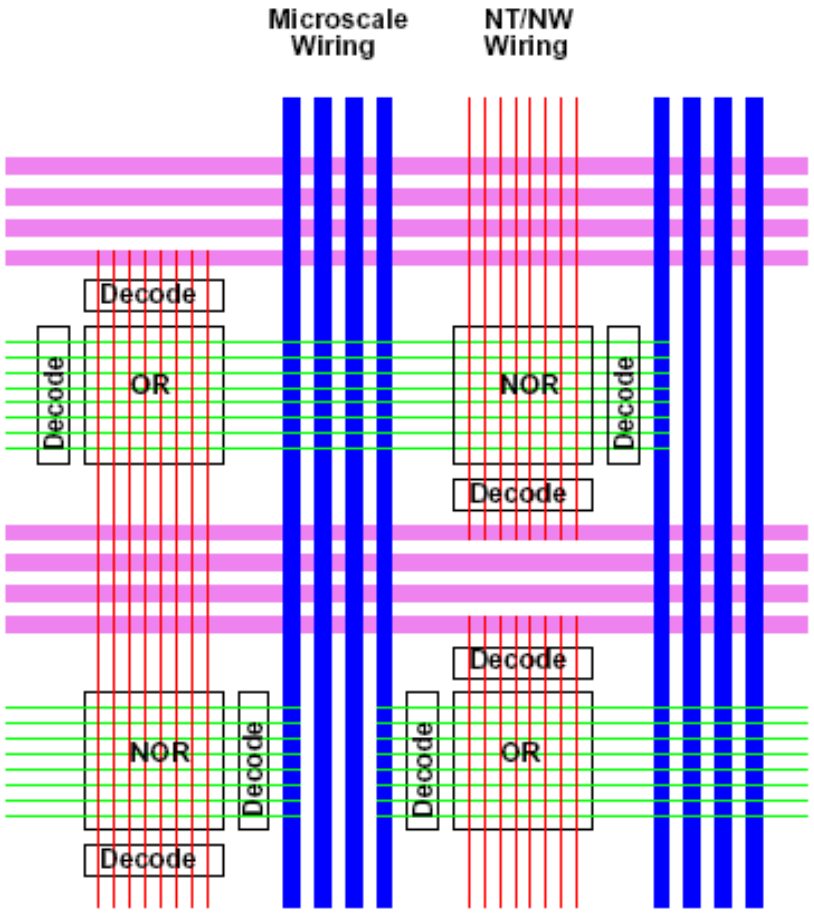
New Roles for Reconfiguration

- Adapt to defects and ageing faults
  - Reload configuration data for FPGA after fault detection.
    - Cosmic rays can erase bits while FPGA in earth orbit
- Provide computational flexibility
  - Change specialized functions on the fly
    - Performance improvement is goal.
- Design and reconfiguration can be simplified if appropriate computational model is used.
  - What models are best adapted to nanoscale systems?

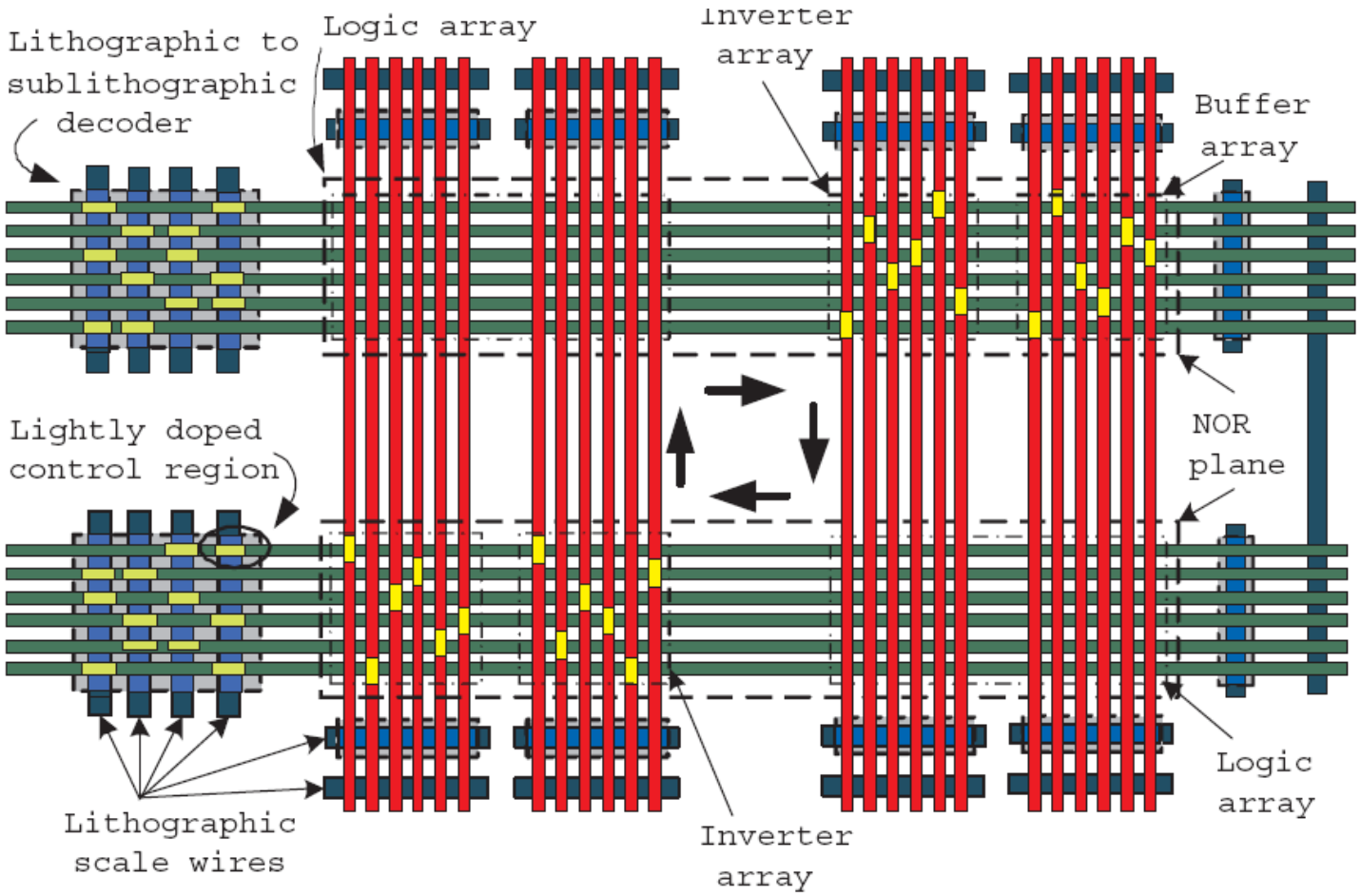
Molecular-Scale Reconfigurable Arrays



- Arrays of configurable nanowire (NW) crossbars.
- A NW crossbar can be used as a memory, PLA, or interconnect.



Nanoscale PLAs



Nanoscale PLAs

- Crossbars will play a central role
- Each row can act as a **wired-OR**.

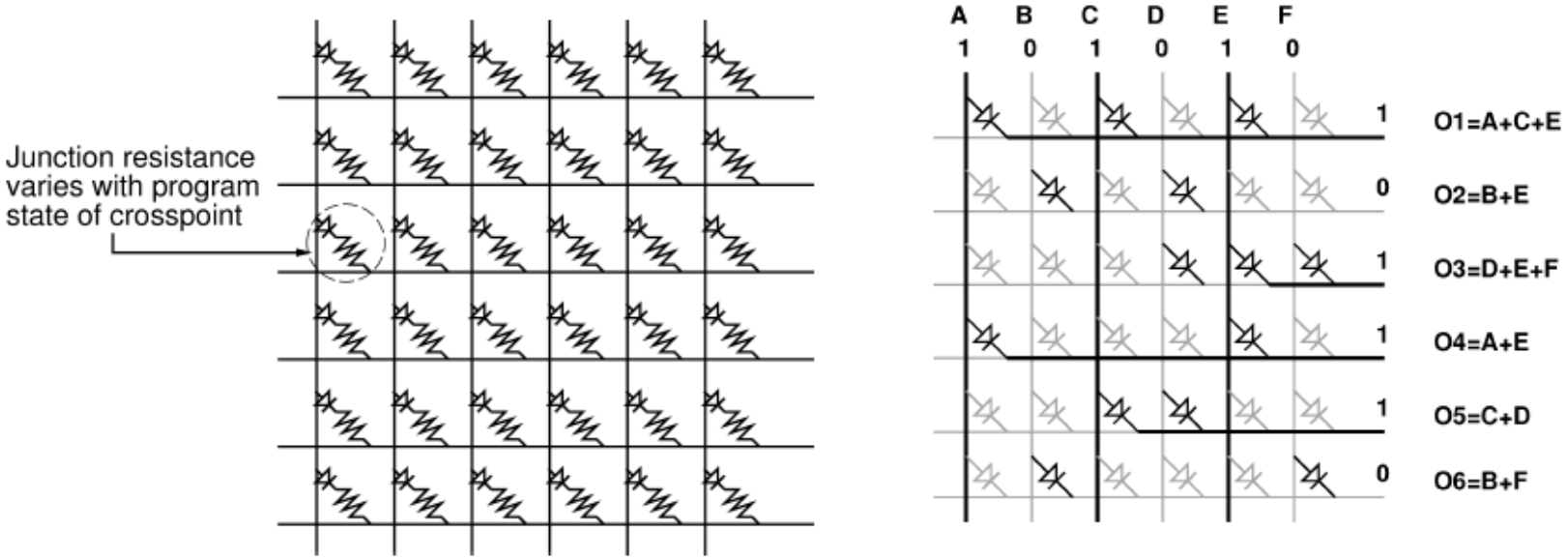
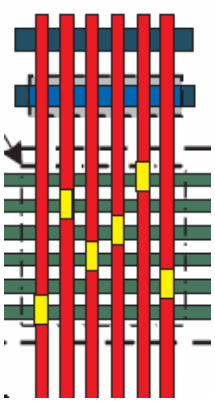


Fig. 5. Logical diode crossbar formed by crossed NWs.

Role of Inversion (NOT)



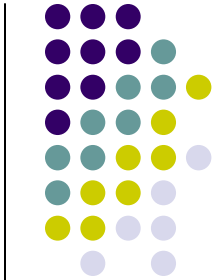
- Needed to boost signal strength.
- Needed for a complete logical basis
  - Can’t get all Boolean functions with AND & OR.
- Inverters are hard to “place” at nanoscale.
  - Lightly-doped regions spaced out on NWs.
  - NWs placement via “floating” results in random alsignment with mesoscale wires

Atomic-Scale Physical Effects

- Material assembly will be statistical.
  - NW addresses will be unknown in advance and must discovered

- Faults will be common.
  - Permanent defects: can program around, perhaps
  - Transient faults: require redundancy or rollback
  - Ageing faults: require monitoring and adaptation

Conclusions



- Reconfigurable computing a rich subject.

- Reconfiguration at the nanoscale is desirable but introduces novel problems.