Nanotechnology & Computer Architectures



Davide Patti (v. Jan 2014.01.15) davide.patti@dieei.unict.it

Nano what?!...

- Manipulation of matter at atomic/molecular scales (1 to 100 nm)
 - Smallest life: Mycoplasma bacteria (200 nm)
 - Smallest atom: hydrogen diameter (0.25 nm)
- Quantum effects:
 - new physical properties, not miniaturised versions of larger devices
 - Transparency, solubility, conductivity

Growing interest

- <u>Application fields</u>: nano-materials, nanomedicine agents, environment (nanofilters), semiconductors, smart food packaging, http://en.wikipedia.org/wiki/List_of_nanotechnology_applications
- <u>Huge research effort</u>: USA 3.7 billion dollars, EU 1.2 billion and Japan 0.75 billions (2012)

Top-down vs bottom-up Design

- <u>Top-down</u>: control on placement of system components (e.g. Photolitography mask to induce a pattern)
- <u>Bottom-up</u>: Rely on *local molecular interactions* to build large-scale structures
- Example:wooden form vs building a wall assembling stones
- Biology inspired (...please avoid *"grey goo"* due exponentially self-replicant nanorobots)

In Computer Design...

- Top-down is the traditional approach: layout mask to specify the computer system structure used by semiconductor industry to place components
- Bottom-up: specify only nano components, NOT their placement. The same properties of each component will allow them attach each other, so the system can be defined as "selfassembled"

Self-Assembly

- Using bottom-up approach, elements self assemble to form a complex system
- Trivial:Random SA (everyone can be everywhere...) → NO imposed order = little customizable complexity (given N components, all random system have similar behaviour)
- Programmable SA: specify how components attach to one another, BUT NO where the will be placed

DNA self-assembly

- Sequence of nucleobases A, G, T, C
- Stable structure when complementary nucleobase sequences match, that is:
 - A pairs with T
 - G pairs with C
- The result is an helix of 2nm diameter
- Larger blocks of assembled DNA sequences, called "motifs" can self-assembled to create more complex structures



DNA Scaffolds - Geometry

 The geometric properties of double strands can form specific, controlled self-assembled nanostructures:



Simple 4-arm junction



...more complex structures







DNA Motifs









20.0 nm

Atomic Force Microscopy (AFM) Image (360nm X 360nm)



http://nano.ece.duke.edu

DNA Nano-grids

Self-assembled grids with sub-nm resolution (3.4 Å)



Placing nano-devices

- DNA Tag: a particular sequence of nucleobases (eg GATTACA, TCGTAAT, etc..)
- Nano devices: nanotubes wires and CNFET transistor, each with <u>specific DNA tags</u>
- DNA structures can provide a "scaffold" onto which nano devices can be attached <u>binding</u> to complementary DNA tags
- Design = Specify the appropriate DNA tags in order to attach nano device terminals

Placing nano-devices

- Example: 2-input NAND → 10 terminals to attach (transistor + wires)
- Note that a basic useful circuit could consist of thousands NAND gates, that is 10.000s terminals to bind
- Adding complexity:
 - In CMOS: larger masks
 - In DNA SA: more different unique tags
- How many tags ?

How many tags ?

- <u>Ideally</u>: choose where every single transistor will be placed-> a different tag for each terminal
- More different unique tags (of a given lenght):
 - more <u>customizable complexity</u>
 - Tags similar to each other-> ... more probability of <u>improper matching</u> (similar to "hamming-distance")
- Conflicting goals: <u>defects rate vs complexity</u>



Defect tolerance

- Functional: a devices does not behave like it should (a transistor does not conduct)
- *Positional*: A device is placed where it shouldn't. Typical of DNA self-assembly
- CMOS doesn't have much defect tolerance
- Instead, in self-assembly, the more complexity we need → the more unique tags → more tolerance needed

Interconnecting nodes

- The size (max 10.000 CNFETs) of blocks is limited by the DNA grid size, due the <u>"defect</u> <u>tolerance/number of tags" conflict</u> discussed above
- So, to increase computing capacity, multiple blocks must be interconnected
- <u>System architects should explicitly partition the</u>
 <u>designs in smaller functional nodes</u>

Self-Organizing Architectures



Self-assembled Computational nodes (minimal computational power & defect-prone)

Self-assembled Interconnect (defect-prone)

> Distant micro-scale I/O contact (low bandwidth)

Defect model includes:

- Rotation, position
- Connectivity
- Defective devices on nodes

http://nano.ece.duke.edu

Design flow

- Architectural description
- Behavioural simulator to verify the high-level procedural model (e.g. System C)
- Gate-level modules implementing the system
- Transistor layout is verified
- DNA sequences (using a prefixed number of unique tags)

Challenges

- **Small-scall control**: nodes with limited space, communication, coordination
- Large-scale randomness: node placement, orientation, connectivity
- High defect rates

Architectural Implications

- Partition functionalities in order to exploit multiple small nodes
- Execution model (appropriate instruction set)
- Memory system (distributed accross nodes)
- <u>Routing: limited space for complex dynamic</u> routing, no guarantees on node placement and connectivity to use static routing
- Interfacing to microscale

System Model



Nodes:

- processing (P),
- memory (M),
- memory ports(M*),
- Anchor via to microscale (A-V)

NANA system features

- Each node generates its own clock (e.g. 10GHz, still pessimistic looking at CNFETs)
- <u>Accumulator based ISA</u> to minimize coordination among nodes
- <u>Packets contain operations</u> and operands in the appropriate order
- A processing node performs the operation and removes operands
- http://www.cs.duke.edu/~alvy/papers/nana.pdf

Packet format



Figure 7.4 Packet format.

Generated by CamScanner from intsig.com

Finding resources for execution

- Packets travel in the network and must be able find the appropriate node type:
 - without deadlocking
 - In a <u>irregular topology</u> of a randomly interconnected sea of nodes
 - <u>Limited node size</u> \rightarrow no large buffers or complex circuitry
- Must exclude hardware hungry solutions: *virtual channels, resource redundancy, dynamic recovery*

Segment-based Routing

http://www.disca.upv.es/jflich/papers/ipd ps06.pdf

- Partitions the topology into different disjoint paths called "segments"
- Each segment connects two other segments (i.e. starts/ends into nodes of another segment)
- <u>Deadlock freedom from turn-prohibition</u>: prohibition of a turn for each segment
- Topology agnostic, but <u>...require</u>
 topology graph as input!



Nanoxim: Distributed Segment-based Routing (DiSR)

- DiSR: same properties as SR
- No topology graph required
- Each node separately contributes to a distributed process that estabilishes segments
- Special packets to discover the network topology and impose segments structures
- SystemC opensource platform:

http://code.google.com/p/nanoxim/

Proposed Activities

- Comparison against simple Up*/Down* spanning tree based
- Quality measures: % of coverage at different node defect rates
- Logic required at each node
- Topology singularities
- Parallel Applications

Node Coverage



32

Setup Latency



NO 600 (1) 434.0 (12 3.3 (13) 434.0 (14) (X) N6 (X (10) 434.0 (10) (X) N12 (X (14) N15 (X) N17 18.3 N18 19.3 N19 465.1 N20 465.1 N21 465.1 N22 465.1 N23 465.1 N24 ····· N25 N27 465.1 N28 N47 18.3 N48 19.3 N49 50.3 N50 495.1 N51 495.1 N52 495.1 N53 53.1 N54 X (N56 465.1 N57) 57.1 N58 465.1 N59 N45 X (N88) 148.0 (N89 <u>ஞ்சுல் சுதன்கையை வாறுகாற காற்காறன. அதற்காற காறுக்கு காறுக்கு காறுக்கு காறு பற்று பிராற்காற் காற்காற்காற்காற்க</u> 495.1 292.0 94.2 95.2 434.0 97.2 106.2 107.2 435.0 465.1 292.0 144.3 463.3 435.0 495.1 85.2 <u>ஞ்சுல வோல குறையான குறையான குறையான குறையான குறையான குறையான குறையான குறையான குறையான குறை</u> குறு குறு குறு குறு குற 434.0 434.0 130.2 163.1 435.0 259.1 259.1 435.0 495.1 434.0 126.2 463.3 434.0 465.1 141.1 434.0 155.2 434.0 157.2 434.0 463.3 160.2 161.2 463.3 434.0 193.1 435.0 259.1 435.0 168.2 465.1 495.1 200.1 292.0 495.1 523.2 434.0 434.0 523.2 176.1 <u>൏൸൘൸൹൸൹൝൹൵൏൵ൕ൵൜൹൹൹൹ൟൕൣൣ൴ൣ൴൏൚൮൚൙൏൸൏൵൴൏൷൏൷ൕ൙൷൙ൣൕൣൣ൴൚൛൏൚൴൙൴൴൴ൣ൴൴൴ൣ൴൴൴</u> 434.0 434.0 184.2 434.0 186.2 187.0 463.3 220.0 250.1 463.3 434.0 194.1 435.0 259.1 495.1 201.2 292.0 523.2 523.2 551.2 434.0 213.2 434.0 434.0 463.3 463.3 463.3 521.3 250.1 463.3 495.1 260.1 434.0 523.2 523.2 551.2 ஞ்சாத வுயா வு 400 வு 243 வு 400 வு 261 வு 267 வு காத வு 269 வு 269 வு 269 வு 269 வு 259 வு 259 வு 259 வு 259 வு 259 வு 259 வு 260 வ 463.3 247.2 463.3 463.3 521.3 251.2 463.3 256.2 434.0 435.0 523.2 551.2 268.2 276.2 463.3 305.3 463.3 463.3 521.3 432.3 463.3 283.2 434.0 435.0 293.2 523.2 523.2 551.2 297.1 <u>௸ௗ௸ௗ௸ௗ௸௳௸௭௸௱௸௱௸௱௸ௐ௸ௐ௸ஂ௶ஂ௶ஂஂ௶ஂ௸௭௸ௐ௸௸ௐ௸௸௷௸ௐ௸ௐௐ௸ௐௐ௸ௐ</u> 463.3 463.3 463.3 521.3 432.3 463.3 313.2 434.0 435.0 465.1 495.1 322.2 322.2 523.2 523.2 356.0 327.2 551.2 N331 (1354 355.1 (1359 356.3 (1359 551.2 (135) 358.1 (1359 359.3 (1359 463.3 463.3 521.3 432.3 463.3 374.1 434.0 435.0 523.2 551.2 551.2 551.2 N361 463.3 400.3 495.1 379.2 521.3 432.3 463.3 403.0 434.0 435.0 465.1 523.2 414.0 523.2 551.2 387.2 387.2 N391 434.0 435.0 465.1 432.3 463.3 436.1 495.1 523.2 443.0 523.2 445.0 551.2 478.0 N421 x 465.1 466.1 495.1 523.2 474.0 465.0 465.0 523.2 551.2 446.2 N453 N452 495.1 523.2 500.0 501.0 457.2 464.2 496.0 465.1 457.3 519.3 463.3 521.3 462.2 464.3 464.2 523.2 551.2 504.1 446.2 465.1 N482 · · N483 · 495.1 523.2 523.2 501.2 533.3 533.0 551.2 565.0 446.2 465.1 486.2 518.3 519.3 463.3 521.3 493.3 493.2 494.2 494.2 495.1 465.1 N512 526.2 527.2 528.2 523.2 523.2 551.2 563.1 565.0 446.2 463.3 520.2 522.2 522.2 523.2 523.2 (1543 544.3 (1544 485.3 (1545) 545.1 (1546) @ X @ 403 @ 403 @ 222 @ 553 @ X @ 232 @ 232 @ 232 @ 533 @ 563 @ X X @ 553 @ 463 @ 463 @ 563 @ 563 @ 563 @ 569 @ 516.2 517.2 549.2 551.2 595.3 N573 485.3 N573 574.0 (1575 575.1 (1576) N572 · 517.2 551.2 623.0 624.0 (1603 604.3 (1604 485.3 (1605 605.1 (1606) 642.0 623.2 549.2 655.0 N633 604.2 N634 N632 516.2 517.2 638.2 639.2 641.2 642.2 643.2 644.2 653.2 665-3 665-1 665-1 65-517-2 665-517-2 N688 689.3 (1689) × 674.2 517.2 549.2 x 517.2 517.2 715.2 716.2 710.2 711.2 712.2 @#31@#31@#31@#31@#31@#31@#31@ X @7332@ X @ X @ X @3132@932@932@932@932@932@932@932@932@932@ 773.0 774.0 775.0 517.2 549.2 517.2 N759 485.3 793.0 516.2 768.2 800.0 802.0 773.2 774.2 516.2 517.2 549.2 517.2 . @¹⁹²³@⁴⁸⁵³@⁷⁹³¹@ X @¹⁹⁵²@⁷⁸⁶²@ X @⁴⁸⁵³@⁴⁹⁵³@⁹⁰²¹@⁵¹⁶²@⁸⁹⁴⁰@⁸⁰⁵¹@····@^{...}@⁵¹⁷²@ 516.2 516.2 516.2 517.2 549.2 x (e2) 823 1 (e2) 516 2 (e2) 516 2 (e2) 927 3 (e2) 485 3 (e2) 485 3 (e2) 485 3 (e3) 830 1 (e3) 832 2 (e3) 832 1 (e3) 835 1 (e3) 835 3 (e3) 835 1 (e3) 835 1 (e3) 935 N820 N821 856.0 860.2 830.2 831.2 485.3 516.2 516.2 516.2 517.2 549.2 855.0 485.3 828.2 838.2 (193) 1952 (193) 1953 (193) 1953 (193) 1953 (193) 1953 (193) 19593 (193) 19593 (193) 19593 (193) 19502 828.2 859.2 859.2 485.3 862.2 516.2 864.2 865.2 517.2 517.2 853.2 855.2 855.2 828.2 (m) 8532 (m) 8031 (m) 5552 (m) (X) (m) 8282 (m) 8993 (m) 5552 (m) 8901 (m) 4553 (m) 4553 (m) 5162 (m) N870 N872 N873 N874