Application-Driven Architectures Using Nanodevices
Participants

- Peter Varman, NSF
- Dan Hammerstrom, Portland State U.
- Bob Colwell, R&E Colwell & Assoc.
- Diana Marculescu, Carnegie Mellon U.
- Kal Gandikota, Portland State U.
- Wolfgang Porod, U. of Notre Dame
- Changjian Gao, Oregon Health and Science U.
- Shalini Prasad, Portland State U.
- Kumar Wickramasinghe, IBM Alamden
- Gary Bradski, Intel
- Supriyo Bandyopadhyay, Virginia Commonwealth U.
- M.P. Anantram, NASA
Product Space for Next 20 Years – What Do We Have?

- Paramount for next generation systems: energy – how do we make energy efficient devices, circuits, and then put them together at architecture-level?
- Good: small, low power devices
- Bad: devices with unreproducible behavior

- Interesting idea: same device does both processing and storage (spintronics?) – terra-byte order
  - Cost is important!
  - ...Or what is the metric of interest: performance, cost, power, reliability...?
  - Most interesting: what is the impact on computing? Killer apps?
Product Space for Next 20 Years – What Do We Want?

- Products of interest will still be in the area of multimedia processing, communication – can nanotechnology “save” us?
- Need devices, circuits, and architectures that pick up when the current industry-driven silicon electronics meets an end in 10+ years

- “Marry” CMOS with nanotechnology – crossbar type of memory with minimal overhead – increase density by $f^2$ ($f =$ feature size)
  - Non-volatile, faster access time – opens new markets
  - Huge potential market

- Non-conventional nanotechnologies are unlikely to replace CMOS – need to leverage and extend what CMOS is good at

- Right target:
  - *Hybrid architectures* that don’t do well with silicon
    - E.g., put sensors on-chip (not off-chip)
  - *Co-processors or specialized processors* are good candidates
Killer Apps

- Search engine app: Google supercomputer-based search engine – already supported on desktops, better ways of finding text

- Need alternate paradigm for doing signal processing than on the conventional technology (spintronics?) – e.g., can do spintronics memory (MRAM?)

- Nano should complement silicon, allow silicon to do what it’s good at

- **Sensors** – but only if better than what silicon can give us already
- Integrating sensors and computing – perform processing closely integrated with sensing (Computing with and closer to sensors!)
- Memory-intensive systems – e.g., image processing
- High-resolution sensors – much better than silicon at interfacing with humans
  - But some of it is related to the AI problem of determining the “context” (for speech recognition, e.g.)
- **Paintable computers? Replace motes with nano-motes (get 20 yrs.of battery)?**
  - Get one order of magnitude improvement in power efficiency – field deployable sensing

- Stuff has to be around for a while before people will know it can be used!
What’s Needed

- Need FT design techniques to keep CMOS alive irrespective of whether we use nano or not: high-level FT for BOTH CMOS and nano(non-CMOS) technologies
- Need FT processing: some traditional computation/algs. (e.g., signal or image processing) are inherently fault tolerant
- Fault detection (What IS a fault? Models for it?)

- Reconfigurable computing
  - Certain instantiations of reconfigurable computing are definitely relevant
  - Still important to have an API that people can understand/use irrespective of the underlying technology

- Studies of architecture that rely ONLY on two-terminal devices

- What type of abstractions should be provided to the architects by the device people?
  - Building blocks: delay, wire cost, I/V characteristics
  - Reproducibility information – what percentage of devices are faulty and how can computation be achieved in their presence?
- …BUT: Unified treatment of all layers (circuit->gate->RTL->uarch->arch->system and appl.) required

- More research funding for feasibility studies that show what/if/how things can be made out of building blocks (i.e., not single gates or devices)
- …Or include the constraints we currently have in the apps. we should look at