Emerging Devices Potpourri

Nanotubes, biology, and quantum – for architects
1) Carbon nanotubes

2) Molecular computing

3) Quantum architecture
Carbon nanotubes
CNT structure/properties, CNFETs, architectural applications
Carbon nanotubes

- Rolled-up graphene sheet
- ~1nm diameter, ~100s nm length
- Single-wall, multi-wall types

- High strength-to-weight ratio (~600x better than steel)
- High heat conductivity
- High electrical conductivity OR semiconducting
CNTs in electronics

- Conduction properties vary based on wrapping geometry
  - Chirality index hard to control
  - If random: 2/3 semiconductive, 1/3 conductive
CNTs for heat transfer

• ~5-10x higher thermal conductivity vs. Cu
• Often limited by interface b/t CNTs, heatsink

• Kaur et al., “Enhanced thermal transport at covalently functionalized carbon nanotube array interfaces”, Nature 2014
CNT-based TSVs

• High thermal + electrical conductivity $\rightarrow$ good for TSVs?
  • $\sim2$-$4$ orders of magnitude higher current density possible (vs. Cu)
  • $\sim5$-$10$x better thermal conductivity
• Interface issues b/t metal layers, CNTs
  • CNT/Cu composites proposed
CNFETs

• Use semiconducting single-wall CNTs for channel (vs. Si, GaAs, etc.)
• Better semiconducting properties than current materials
• Can scale down to at least 1.8nm node (IBM)
CNFETs – back-gated
CNFETs – top-gated
CNFETs - wraparound
CNFET limitations

• CNT alignment
  • Mispositioned CNTs – logic errors
• CNT selection
  • Metallic or multiwall CNTs – short-circuits, logic errors, leakage
CNFET limitations

• New techniques improve CNT selection, positioning accuracy

• Wisconsin
  • CNFETs with better $I_{DS}$ (sat) than Si, GaAs FETs
  • Achieved >99.99% semiconducting CNTs
  • Scalability unknown

• Brady et al., “Quasi-ballistic carbon nanotube array transistors with current density exceeding Si and GaAs”, Science Advances 2016
CNFET limitations

• New techniques improve CNT selection, positioning accuracy
• Stanford
  • “Imperfection-immune design”
  • Form extended gate region
  • Etch extension away after deposition
  • Physically burn out metallic CNTs

• Mitra et al., “Imperfection-Immune VLSI Logic Circuits using Carbon Nanotube Field Effect Transistors”, 2009
Carbon nanotube computing

• Basic transistors, gates, etc. doable for years
• First known Turing-complete CNT-based computer

• Shulaker et al., “Carbon nanotube computer”, Nature 2013
CNT computer - overview

• Programmable, stored-program architecture
• MIPS SUBNEG instruction provides Turing-completeness
  • Authors claim support for ~20 MIPS instructions... ish...
• Runs simple multitasking OS
CNT computer - architecture
CNT computer - hardware

- 178 P-type CNFETs
- 1 KHz clock
  - Limited by 1μm gate length, capacitive loading of test setup
- Only operates on single-bit data values
  - Also demoed proof-of-concept 2-bit arithmetic unit (96 CNFETs)
CNTs looking forward

- Lots of recent progress
- CNFET devices still in infancy
  - Reliability
    - Positioning accuracy
    - Metallic CNT removal
  - Interface between CNTs and other materials
  - Large-scale systems
  - Manufacturability
CNTs looking forward

• ... can we exploit the mechanics of CNTs?

Conceptual diagram of how NRAM works, by connecting and disconnecting CNTs to form 0s and 1s (high and low resistance).
Molecular computing
DNA principles, archival data storage applications, &c.
Molecular computing

• Building computers with biochemistry
• Store data with DNA, process with enzymes
• Massively data-centric
  • DNA has ultra-high information density
  • Unwrapping DNA strands $\rightarrow$ could operate on huge amounts of data simultaneously
  • Comparable to in-memory computing
• Often limited by latency of biochemical ops (but it’s getting better...)

http://www.utahpeoplespost.com/wp-content/uploads/2014/05/DNA.jpg
Review: DNA

- Deoxyribonucleic acid
- Double-helix backbone
- Nucleotide bases (paired):
  - Adenine/Thymine
  - Cytosine/Guanine

Review: DNA replication

(a) Initial processes
- Chromosomal proteins (histones in eukaryotes and archa) removed
- DNA helicase
- Stabilizing proteins
- DNA polymerase III

(b) Synthesis of leading strand
- Replication fork
- Triphosphate nucleotide
- Leading strand
- RNA primer
- Primase
- DNA polymerase III
- Okazaki fragment
- DNA ligase

(c) Synthesis of lagging strand
- Replication fork
- Triphosphate nucleotide
- RNA primer
- DNA polymerase III
- DNA polymerase I

Copyright © 2006 Pearson Education, Inc., publishing as Benjamin Cummings.
Polymerase chain reaction (PCR)

- Use replication process to artificially create many copies of DNA segment
  - Heat DNA to separate strands
  - Bond primer to target DNA location
  - Polymerase synthesizes new sequence starting at primer location
  - Continue until have enough copies (or biochemical resources run out)

“A DNA-Based Archival Storage System”
(Bornholt et al, ASPLOS ’16)

• Need ultradense archival data storage
  • 10GB / mm³ – current tape cartridges
  • 100GB / mm³ – recent optical-disc research
  • 1EB (10⁹GB) / mm³ – DNA

• DNA challenges:
  • Synthesis/sequencing error rates
  • Difficult random access
    • Read must sequence full strand, write can synthesize pieces
    • \( \rightarrow \) Higher read than write latency
    • \( \rightarrow \) Currently restricted to large-block access

Figure 1. Carlson curves [4]: trends in DNA synthesis and sequencing technology compared to Moore’s Law. DNA productivity is measured in nucleotides per person per day. Recent growth in sequencing technology eclipses Moore.

Figure 2. DNA storage as the bottom level of the storage hierarchy
DNA storage system overview

• Physical limitations
  • 100-200 nucleotides per strand (50-100 bits)
  • Many strands per data object
  • No structured access → must embed address information on each data strand

• Key-value architecture
  • Key = DNA sequence tied to PCR primer

• Write:
  • Synthesize DNA, including primer
  • Store in a DNA pool

• Read:
  • Amplify desired sequence with PCR
  • Sample the pool and sequence strand(s)
DNA storage system overview

• DNA pool tradeoffs
  • Want >1 pool:
    • Needs many primers to distinguish all keys → higher likelihood of inter-primer reactions
    • Random sample from single pool less likely to contain all the desired data
  • Don’t want separate pool for each key (bad storage density)
  • Use reasonably-sized pools with multiple keys per pool
  • Reads remove DNA from pool → may need to periodically replenish or resynthesize

Figure 3. Overview of a DNA storage system.
DNA storage system overview

(a) The write process performs \texttt{put(key, value)}, generating a DNA library.

(b) The read process performs \texttt{get(key)} on a DNA library, returning the value.

\textbf{Figure 4.} Overview of a DNA storage system operation as a key-value store.
Data representation

• Obvious = base-4
  • But, repetitions of same nucleotide ("homopolymers") increase chance of sequencing errors
• Use base-3 with rotating code
• Inefficient mapping between base-2/base-3
  • Use Huffman coding to map each base-2 byte to 5-6 ternary digits

**Figure 5.** Encoding a stream of binary data as a stream of nucleotides. A Huffman code translates binary to ternary digits, and a rotating encoding translates ternary digits to nucleotides.
Encoding methods

• Need to tolerate high (~1%/nucleotide) error rates

• Goldman coding:
  • Split input into overlapping segments → 4x copies of any given block

• XOR coding:
  • Compute XOR of two strands to generate parity strand (like RAID-5)
  • Can tune reliability per block by adding more parity strands
Evaluation

- Encoded images, synthesized into DNA, and successfully recovered them
- Good resilience with Goldman/XOR
  - Even at low sequencing depth (# times a nucleotide read during sequencing)
- Extremely high durability (simulated)
Quantum computing

Defining “quantum”, building blocks, interfacing classical/quantum components
What is “quantum”?  

• Atomic and subatomic particles with distinct (but stochastic) states  
• States defined in vector space  
  • Unit sphere (“Bloch sphere”)  
• Pure states  
  • On surface of Bloch sphere  
  • Represented by a unit vector  
• Mixed states  
  • Somewhere inside the Bloch sphere  
  • Result from coupling (“entanglement”) between multiple quantum objects  
• Superposition – vector addition
What’s it good for?

• Stochastic rather than deterministic
  • Answers are “correct” with some probability

• Entanglement and superposition
  • Interesting parallelism due to direct inter-qubit interactions
  • Solve certain problems much faster than is possible on classical machines
    • Factoring (Shor’s algorithm)
    • Statistical analysis
    • Fourier transforms
    • …
Qubits

• Objects that exhibit quantum state
  • Photons
  • Electrons
  • Quantum dots (nm-scale semiconductor particles)
  • ...

• Qubits must:
  • Be well-isolated from any potential interference
  • Strongly couple with a classical system that can manipulate them quickly
  • Remain coherent for a sufficient length of time to do useful computation
  • Be controllably entangled with other qubits
Superconducting transmon qubits

• Superconductivity can create better-defined quantum phenomena

• Josephson junction
  • Superconductive coupling between two semiconductors at ultra-low temps
  • Modulate gate electrode voltage to control operation
    • Apply “quantum gates” to qubits
    • Measure results

http://www.lps.umd.edu/Quantum%20Computing%20Group/Cooper-Pair%20Box%20Qubit/Cooper-PairBoxQubit1.html

“An Experimental Microarchitecture for a Superconducting Quantum Processor” (Fu et al, MICRO ’17)

• Microarchitectures need to support mixed quantum-classical code

• Prior work:
  • Intermediate representations of quantum applications
  • No consideration of low-level interface constraints with quantum processor

• This work:
  • Supports mixed code
  • Precisely controls a quantum chip

Figure 1: Overview of the quantum computer system stack from [2].
Architecture overview

Figure 4: Overview of the Quantum MicroArchitecture (QuMA).
Architecture overview

• Classical-CPU host
• Quantum coprocessor
  • Classical components to interface with classical CPU
  • Analog-digital interface to operate using quantum chip
Codeword-triggered pulse generation

• Generate analog pulses that correspond to primitive ops
• Use LUT indexed by codeword to trigger sending a specific pulse
• Also used for measurement (quantum state → binary values)
Queue-based event timing control

• Timing control unit
  • Boundary b/t deterministic and non-deterministic timing domains
  • Fire specific sequences of events at precise times in quantum domain
  • Timing queue + multiple event queues + timing controller
Multilevel instruction decoding

- Quantum Instruction Set
  - Auxiliary classical instructions
    - Run directly on Execution Controller
  - Quantum instructions
    - Sent by Execution Controller to Physical Microcode Unit

Figure 4: Overview of the Quantum MicroArchitecture (QuMA).
Multilevel instruction decoding

• Physical Microcode Unit
  • Translates quantum instructions into quantum microinstructions
    • Wait (specify timing interval)
    • Pulse (apply quantum gates on qubits with primitive waveforms)
    • MPG (generate measurement pulse)
    • MD (measure results)
  
• Horizontal microcode
  • Apply same Pulse/MPG to multiple qubits simultaneously
Multilevel instruction decoding

- Quantum Microinstruction Buffer
  - Microinstructs $\rightarrow$ micro-ops with timing labels $\rightarrow$ Timing Ctrl Unit
- Micro-op unit
  - Micro-op $\rightarrow$ sequence of codeword triggers with predefined latency
Architecture overview

Figure 4: Overview of the Quantum MicroArchitecture (QuMA).
Implementation

Figure 7: Schematic of the implemented QuMA. The thick gray lines are analog signals while the dark thin lines are digital signals. Dashed lines indicate functionality to be added in the future.
Implementation

Figure 8: Experimental setup used for validation of the microarchitecture.

Figure 9: The AllXY result of qubit 2. In the label, each X/Y (x/y) denotes a rotation by $\pi$ ($\pi/2$) around the x/y axis of the Bloch sphere.
In summary...

- Carbon nanotubes – structure/properties, CNFETs, NRAM, etc.

- Molecular computing – DNA storage, computation potential

- Quantum architecture – basic components, practical interfaces
Evaluation [quantarch]

• (compared vs Raython BBN APS2 device)
• Centralized architecture; only one binary executable needed
• Quantum semantics/timing explicitly defined at instruction-level
• Synchronization not dependent on other modules or on interconnect
• Scalability limited by instruction issue rate
  • VLIW would help
  • Could better optimize quantum microcode and micro-op units