Computers of the Future?

CS 221

Moore’s Law Ending in 2018?

• Moore’s Law: Processor speed / number transistors doubling approximately 18 months
Moore’s Law

• Recent research predicts an end to Moore’s Law in 2018
  – Most of Moore’s law is based upon shrinking the dimensions of the transistors on the chip
  – But the laws of physics do not allow transistor technology to operate be
Physical Limitations

• Current processors are starting to be manufactured on a 0.09 micron process
  – 1 micron = 1 millionth of a meter
  – 0.09 micron process = 90 nanometers, nanometers is a billionth of a meter
    • 1 nanometers is the width of about 3 silicon atoms

Size

Silicon Devices are Nanotechnology

Transistor for 90 nm process
Influenza virus

Source: CDC
Physical Limitations

• Physical limitation at a 0.016 micron process
  – 16 nanometers
  – Smaller than this quantum effects begin to take over, electronics becomes unpredictable
  – If Moore’s Law continues to hold, we’ll hit 16nm in 2018

Physical Limitations

• At 16nm, the distance between the source and drain is approximately 5 nm

Electron tunneling may occur with 50% probability from source to drain even if the gate is closed. Heisenberg uncertainty principle. Analogy: A light misty waterfall; some people may walk through, or go around.
Moore’s Law & Gate Length

Transistor Physical Gate Length Trend
(Lithography generation > $L_{GATE}$)

Moore’s Law
Are we nearing end?

• Not necessarily! Lots of new contenders for the (distant?) future
  - Nanotechnology and self-assembly
  - DNA Computing
  - Quantum Computing
  - Optical Computing
    • Skip – limited to special purpose uses today, e.g. fourier transforms, filtering
  - Parallel Processing
    • Skip – We covered this a little bit on MMX and Alternative Architectures

Nanotechnology

• A limitation of the transistor is it’s design and construction
  – Lithography onto a substrate of a semiconductor material, results in the limitations described previously
• Bypassing the limitations
  – Design a whole new transistor, built from the bottom-up
    • i.e. build circuit by piecing together individual atoms instead of on a substrate of silicon
    • Self-assembly of circuits
History: Feynman

- History: Feynman On Computers
  - “... Why can’t we make them very small, make them of little wires... the wires could be 10 or 100 atoms in diameter, and the circuits could be a few [hundred nanometers] across.”

Roots of NanoScience

- Roots of NanoScience
  - 1981 – SPM (Scanning Probe Microscopes)
  - Allowed us to image individual atoms
  - Small tip (a few atoms in size) is held above the conductive surface. Electrons “tunnel” (STM’s) between the probe and surface (by Quantum Mechanics).
  - The tip is scanned across the surface measuring the current to create the image.
Roots of NanoScience

- C60 – Buckminster Fullerene – Bucky balls are discovered in 1985. Stable molecule entirely made of carbon.
- With STM’s, IBM researchers in 1990 positioned atoms on a surface.
- Carbon nanotubes – tubes made entirely of carbon rings. 1991
  - Ring of carbon molecules with attached to other molecules (e.g. hydrogen, oxygen, etc.)

Carbon NanoWires

- Carbon nanowire, one molecule
- Excellent electrical properties
  - By designing the atoms attached to the ring, electron orbits can be arranged to allow conduction of electricity
  - Rare organic molecule to conduct
  - Electrons travel at c/10 with no resistance
  - Flow more exact, orderly, than copper wire
Nano Switch

- “Dumbell” shaped component added to ring
  - First called rotaxane
  - When oriented in plane, current will flow
  - If voltage applied the component will flip and block current
- Nano Transistor, NOT gate created.
  - Need NOR and NAND to create general purpose logic gates!

Self Assembly

- By adding “sticky” endgroups to the carbon rings, molecules can attach to metal or orient to attach to each other
Molecular Computing

• Assembly at the atomic/molecular scale
• Many benefits
  – Faster, smaller
  – No irregular shapes, defects, three-terminal devices

DNA Computing

• Pioneered in 1994 by Len Adleman, USC
  – Uses strands of engineered DNA to implement a massively parallel processor
• DNA (deoxyribonucleic acid)
  – Encodes the genetic information of cellular organisms.
  – Consists of polymer chains, or strands
  – Each strand may be viewed as a chain of nucleotides, or bases, of length n.
  – The four DNA nucleotides are adenine, guanine, cytosine and thymine, commonly abbreviated to A,G,C and T respectively.
  – Bonding occurs by the pairwise attraction of bases; A bonds with T and G bonds with C. The pairs (A,T) and (G,C) are therefore known as Watson-Crick complementary base pairs.

AACGCGTACGTTCAAGTGTCGGAATGGCCAATG
TTGCGCATGCTTTACCAGGCTTTACGGTTAC
Adleman’s Experiment

- DNA “Computer” to solve a small Hamilton Path Problem
  - Given a set of n cities connected by one-way and two-way roads, does there exist a path through this network starting at the first city and ending at the last city such that each city is visited once and only once?
  - For large n and arbitrary paths, the only known solution to this problem requires exponential time to solve
Adleman’s Experiment

**Strategy:** Encode city names in short DNA sequences. Encode itineraries by connecting the city sequences for which routes exist. DNA can simply be treated as a string of data. For example, each city can be represented by a “word” of six bases:

<table>
<thead>
<tr>
<th>City</th>
<th>DNA Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Los Angeles</td>
<td>GCTACG</td>
</tr>
<tr>
<td>Chicago</td>
<td>CTAAGT</td>
</tr>
<tr>
<td>Dallas</td>
<td>TCGTAC</td>
</tr>
<tr>
<td>Miami</td>
<td>CTACCG</td>
</tr>
<tr>
<td>New York</td>
<td>ATGCCG</td>
</tr>
</tbody>
</table>

The entire itinerary can be encoded by simply stringing together these DNA sequences that represent specific cities. For example, the route from L.A -> Chicago -> Dallas -> Miami -> New York would simply be GCTACGCTAGTATCGTACCTACGGATGCCG, or equivalently it could be represented in double stranded form with its complement sequence.

### Linking Cities

- To encode a route between two cities, create a strand that is the complement of the last half of the source city and the first half of the destination city.
- Bindings only connect these cities.
Parallel Processing

• Create huge numbers of encodings
  – Say $10^{13}$ copies of each city and each route between cities
  – Put in a solution
  – Mix
  – Generates millions of resulting strands, each strand represent a tour among cities

Post Processing

• The second stage was to extract those strands which corresponded to a certain length which signified exactly 5 cities being passed through. If each city is represented by 8 DNA bases, all strands of 40 bases would be extracted and stored in a separate test tube.
• The third stage is to extract all those strands containing the DNA sequence for city 1, then those containing the DNA sequence for city 2, and so on. If there is a solution to this route problem, it will be found in the strands extracted for the last city 5.
Other DNA “Computers”

• 2001 - Expert System
  – If invest THEN wealth
  – If wealth THEN retire
  – Etc.
• 2003 - Tic Tac Toe

Quantum Computers

• Based on Quantum Mechanics
  – We know how it works but not why it works
  – “No, you’re not going to be able to understand it…that is because I don’t understand it… the theory of quantum electrodynamics describes Nature as absurd from the point of view of common sense. And it agrees fully with experiment. So I hope you can accept Nature as she is – absurd.” -- Richard Feynman
Examples of Quantum Systems

- Energy levels in an atom
- Polarization of photons
- Spin of an electron
- Interference of photons
  - Use interference as an example, although Quantum Computers today generally use spin/energy levels instead

Interference Experiment

- Light source emits a photon toward a half-silvered mirror

![Interference Experiment Diagram]

- This mirror splits the light, reflecting half vertically toward detector A and transmitting half toward detector B.
- A photon, however, is a single quantized packet of light and cannot be split, so it is detected with equal probability at either A or B.
Interference Experiment

• Intuitive explanation:

![Diagram](image)

- The photon either goes toward A or goes toward B
- But the following suggests the photon actually travels toward BOTH A and B simultaneously! But when it is actually detected, it is only in one place

---

Interference Experiment

• Add more mirrors, two full and another half

![Diagram](image)

- Expect: 50% detection at A and B
Interference Experiment

• What is actually observed

100% detection at A and never at B!
Theory: photon travels both paths simultaneously, creating an interference at the point of intersection that destroyed the possibility of the signal reaching B

Quantum Computing

• This effect is called quantum interference
• The photon state is called superposition
  – With some probability, the photon is in a path to A, a path to B, or in both paths at once
• Analogous system with energy levels in atoms, spin of electrons
  – Electron could be in a known, fixed state, or in multiple states at once
  – If we encode data using quantum superposition this is a quantum bit, or a qubit
  – Once we “look” at our qubit it stops being in multiple states and is fixed
Qubits

• Qubits represent both memory and processing
  – Can store a 0, 1, or both 1 and 0 at once
  – Consider 8 traditional bits; can represent one of 256 values at a time
  – Consider 8 qubits; can represent 256 values all at the same time
  – Potential for inherent exponential parallelism

• Entanglement
  – To be useful our qubits must all be linked; one must affect the other, otherwise we just have independent bits
  – Entanglement of particles maintains coherence between the qubits
  – Pairs stay in an entangled superposition, but then measuring one results in the opposite measurement of the other
    • Distances as far as 1mm discovered
    • Instant change, apparently faster than the speed of light
    • Two entangled qubits could store 4 bits of information

Implications

• Grover’s quantum search algorithm finds an item in an unsorted list in $O(n^{1/2})$ time instead of $O(n)$ time.

• Shor’s quantum factoring algorithm factors an n-digit number in $O(n^3)$ time while the best known classical algorithm takes $O(2^n)$ time.
  – Basis for encryption algorithms
Today’s Quantum Computers

• Still at an early stage
• 1998
  – Los Alamos and MIT, stored data in a qubit
• 1999
  – IBM, 5 qubit for Grover’s algorithm
• 2000
  – Los Alamos, 7 qubit computer in a drop of liquid, but not all interacting
• 2000
  – IBM, 5 interacting qubits using flourine atoms
  – Solved small order-finding problem in one step
• 2001
  – IBM, Shor’s algorithm to factor 15 in one step
• 2003
  – NEC, 2-qubit quantum solid state logic gate created

Conclusion

• The Future?
  – “No one knows how much of technology’s promise will prove out. Technology prediction has never been too reliable. In the March 1949 edition of Popular Mechanics… experts predicted computer of the future would add as many as 5000 numbers per second, weigh only 3000 pounds, and consume only 10 kilowatts of power.” – Nanotechnology conference