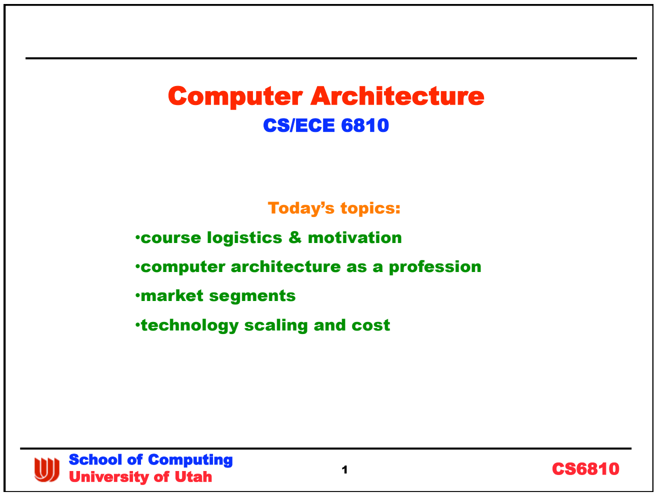


Computer Architecture

CS/ECE 6810

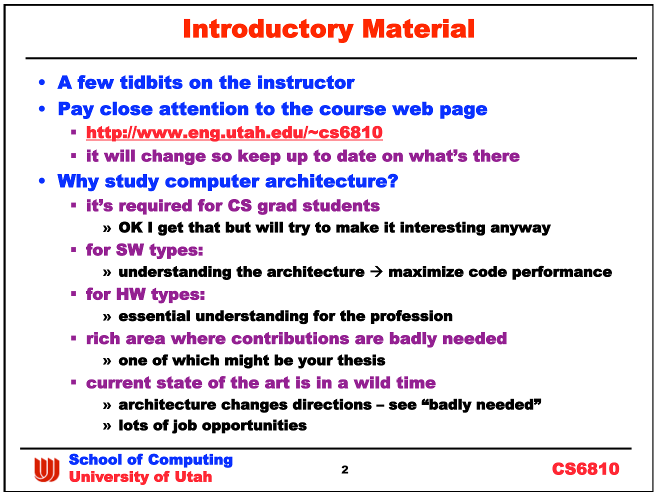
Today's topics:

•course logistics & motivation

•computer architecture as a profession

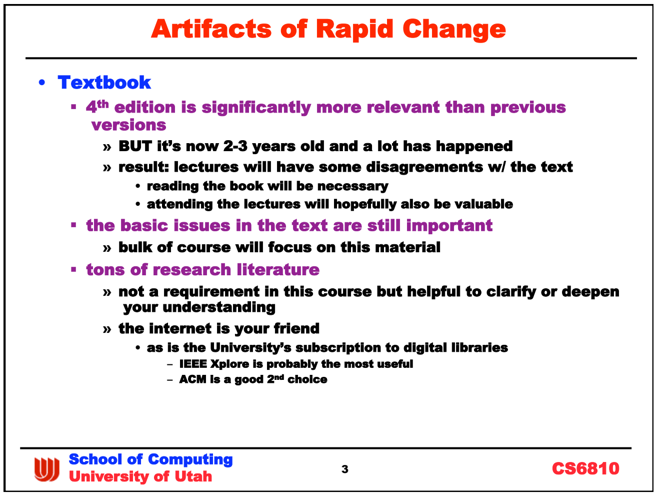
•market segments

•technology scaling and cost



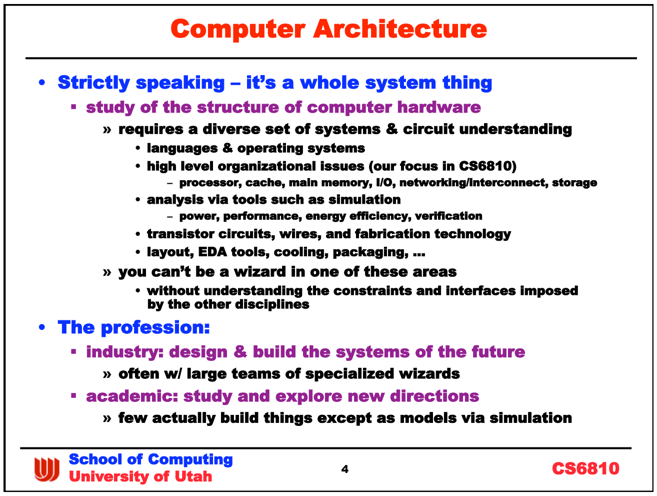
Introductory Material

- A few tidbits on the instructor
- Pay close attention to the course web page
  - http://www.eng.utah.edu/~cs6810
  - it will change so keep up to date on what's there
- Why study computer architecture?
  - it's required for CS grad students
    - » OK I get that but will try to make it interesting anyway
  - for SW types:
    - » understanding the architecture → maximize code performance
  - for HW types:
    - » essential understanding for the profession
  - rich area where contributions are badly needed
    - » one of which might be your thesis
  - current state of the art is in a wild time
    - » architecture changes directions – see "badly needed"
    - » lots of job opportunities



Artifacts of Rapid Change

- Textbook
  - 4th edition is significantly more relevant than previous versions
    - » BUT it's now 2-3 years old and a lot has happened
    - » result: lectures will have some disagreements w/ the text
      - reading the book will be necessary
      - attending the lectures will hopefully also be valuable
  - the basic issues in the text are still important
    - » bulk of course will focus on this material
  - tons of research literature
    - » not a requirement in this course but helpful to clarify or deepen your understanding
    - » the internet is your friend
      - as is the University's subscription to digital libraries
        - – IEEE Xplore is probably the most useful
        - – ACM is a good 2nd choice

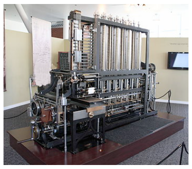


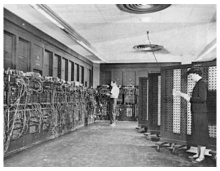
Computer Architecture

- Strictly speaking – it's a whole system thing
  - study of the structure of computer hardware
    - » requires a diverse set of systems & circuit understanding
      - languages & operating systems
      - high level organizational issues (our focus in CS6810)
        - – processor, cache, main memory, I/O, networking/interconnect, storage
      - analysis via tools such as simulation
        - – power, performance, energy efficiency, verification
      - transistor circuits, wires, and fabrication technology
      - layout, EDA tools, cooling, packaging, ...
    - » you can't be a wizard in one of these areas
      - without understanding the constraints and interfaces imposed by the other disciplines
- The profession:
  - industry: design & build the systems of the future
    - » often w/ large teams of specialized wizards
  - academic: study and explore new directions
    - » few actually build things except as models via simulation

A Snippet of Modern History

- **Mechanical difference engine**
  - proposed in 1786 by J. H. Mueller
  - 2 versions built by Charles Babbage in the 1820's
    - » image at right is a replica in the computer museum in Mtn. View, CA
- **Electronic computer**
  - WW2: army needed something to compute ballistics tables
  - contract w/U Penn in 1943
  - operational in 1946
    - » analog machine
    - » programmed by plugging cables into the right spot
    - » YOW!!
  - numerous analog machines follow
    - » vacuum tubes, crystal diodes, ...



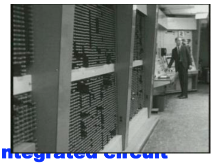


History II

- **Stored program computer**
  - Concept: EDVAC report 1945 by John von Neumann
    - » hence the von Neumann architecture tag
  - Implementation
    - » Jun 1948 – Manchester Univ. experimental machine
      - "Baby" SSEM (small scale experimental machine)
      - memory = Williams Cathode Ray Tubes
    - » First practical: EDSAC May 1949 (also done in the UK)
      - memory = delay-lines

Then Came Transistors ...

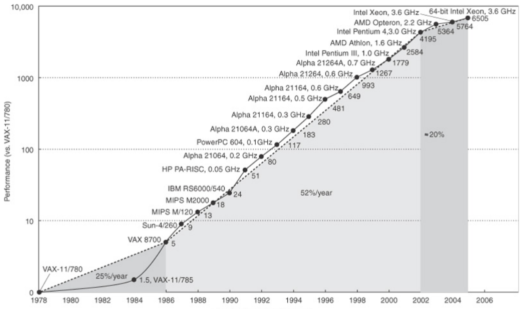
- **First all transistor computer**
  - MIT's Linc, TX-0, TX-2 – Wes Clark 1950's
    - » led to networks, graphics, interactive computing



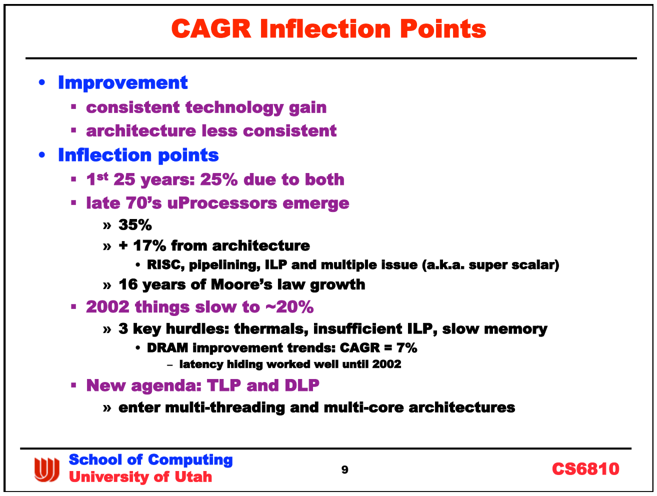
- **Integrated circuit**
  - more than one transistor on a die – 1958
    - » Rob't Noyce and Gordon Moore @ Fairchild
      - later founders of Intel
  - first microprocessor Intel 4004 in 1971
    - » 10 um pMOS, 92 Kips, 740 KHz, 4-bit data-path, BCD
    - » it's been a wild ride ever since

Unprecedented Improvement

- **Moore's surprising prediction in 1965 holds up**
  - reasonably well so far

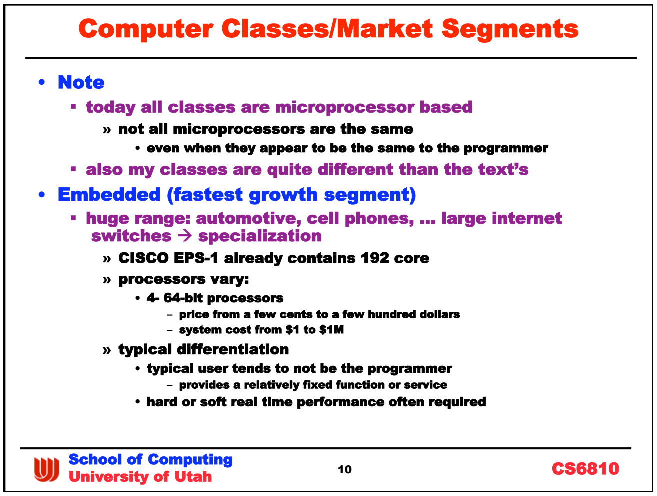


© 2007 Elsevier, Inc. All rights reserved.



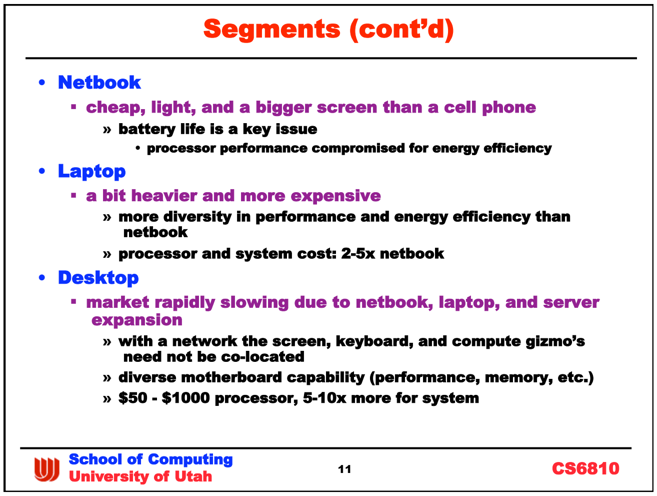
CAGR Inflection Points

- Improvement
  - consistent technology gain
  - architecture less consistent
- Inflection points
  - 1st 25 years: 25% due to both
  - late 70's uProcessors emerge
    - » 35%
    - » + 17% from architecture
      - RISC, pipelining, ILP and multiple issue (a.k.a. super scalar)
    - » 16 years of Moore's law growth
  - 2002 things slow to ~20%
    - » 3 key hurdles: thermals, insufficient ILP, slow memory
      - DRAM improvement trends: CAGR = 7%
        - latency hiding worked well until 2002
  - New agenda: TLP and DLP
    - » enter multi-threading and multi-core architectures



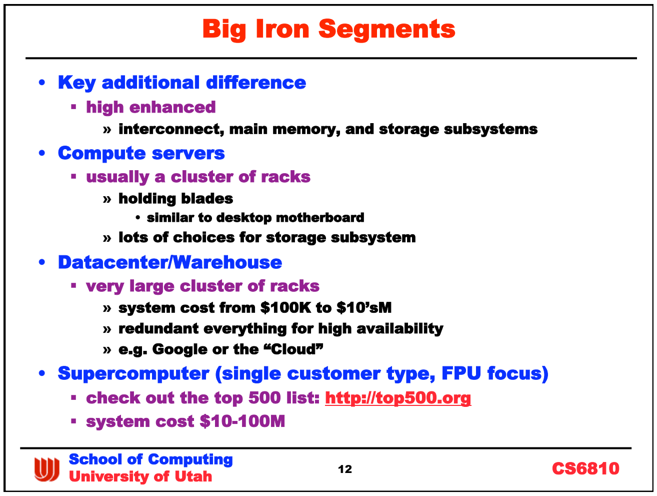
Computer Classes/Market Segments

- Note
  - today all classes are microprocessor based
    - » not all microprocessors are the same
      - even when they appear to be the same to the programmer
  - also my classes are quite different than the text's
- Embedded (fastest growth segment)
  - huge range: automotive, cell phones, ... large internet switches → specialization
    - » CISCO EPS-1 already contains 192 core
    - » processors vary:
      - 4- 64-bit processors
        - price from a few cents to a few hundred dollars
        - system cost from $1 to $1M
    - » typical differentiation
      - typical user tends to not be the programmer
        - provides a relatively fixed function or service
      - hard or soft real time performance often required



Segments (cont'd)

- Netbook
  - cheap, light, and a bigger screen than a cell phone
    - » battery life is a key issue
      - processor performance compromised for energy efficiency
- Laptop
  - a bit heavier and more expensive
    - » more diversity in performance and energy efficiency than netbook
    - » processor and system cost: 2-5x netbook
- Desktop
  - market rapidly slowing due to netbook, laptop, and server expansion
    - » with a network the screen, keyboard, and compute gizmo's need not be co-located
    - » diverse motherboard capability (performance, memory, etc.)
    - » $50 - $1000 processor, 5-10x more for system

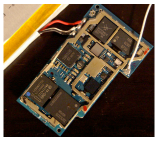


Big Iron Segments

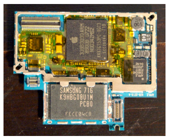
- Key additional difference
  - high enhanced
    - » interconnect, main memory, and storage subsystems
- Compute servers
  - usually a cluster of racks
    - » holding blades
      - similar to desktop motherboard
    - » lots of choices for storage subsystem
- Datacenter/Warehouse
  - very large cluster of racks
    - » system cost from $100K to $10'sM
    - » redundant everything for high availability
    - » e.g. Google or the "Cloud"
- Supercomputer (single customer type, FPU focus)
  - check out the top 500 list: http://top500.org
  - system cost $10-100M

Addendum: Complex Embedded Systems

- **Key characteristic**
  - **some things programmable and some things not**
    - » **not: ASIC or IP blocks**
- **Example: iPhone**
  - **Single programmable ARM core**
    - » **integrated with ~50 IP blocks**
      - **each block is highly specialized (multiple blocks/chip)**
        - – **~100x improvement in energy-delay product**



Cell Phone Part



Computer Part

Source: Anandtech

Computer Architecture

- **Focus issues for CS6810**
  - **3 key components**
    - » **ISA (2 lectures from now – Appendix B)**
    - » **Organization**
      - **high level structural aspects of various subsystems**
        - – **pipeline structure**
        - – **function unit structure**
        - – **processor structure**
        - – **cache hierarchy structure**
        - – **main memory structure**
        - – **I/O & network interface**
      - **note there are other subsystems that we'll get to in the 2nd half**
        - – **interconnect structure**
        - – **storage structure and technologies**
    - » **Hardware (only light coverage here)**
      - **logic design, process, packaging, cooling, timing, wires …**
        - – **this is an almost endless list**
        - – **VLSI courses in analog and digital IC design should be next**
          - – **if you intend to live on the HW side of architecture**

Intent of the Course

- **Provide a foundation for future professional activity**
  - **at least 3 possible goals**
    - » **understanding the compute platform that you use**
      - **key to achieving highly efficient code for SW types**
      - **ISA and organization are what you care about**
    - » **research into new architectural options**
      - **key academic role & possible thesis area preparation**
      - **ISA, organization, and high level understanding of hardware constraints will be needed**
    - » **design of new systems**
      - **perhaps the ultimate relevance**
      - **further courses will be needed to finish this process**
        - – **VLSI & embedded systems courses will be your next step**
      - **all 3 will are important**
        - – **in much greater depth**
- **OK that's the sales pitch**
  - **for why you should care**

Tracking Trends

- **Fast moving arena & lengthy design process**
  - **typical 5-year design schedule**
    - » **typical design team ~500 engineers**
    - » **year 1: architectural concept and simulation infrastructure development**
    - » **year 2: architectural optimization, validation**
      - **architecture freeze at the end of year 2**
    - » **year 3 & 4: circuit design, floor-planning, and packaging**
      - **tape out at the end of year 4**
    - » **year 5**
      - **refine process to achieve acceptable yields**
      - **test and validate fab'd chips**
      - **build inventory since volume sales commence**
- **Note**
  - **you need to design for a technology that doesn't exist when the design phase starts**
  - **need to accurately predict what will be available**

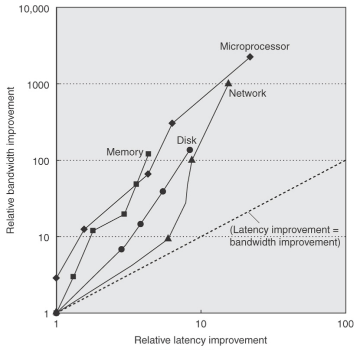
Technology Trends

- VLSI
  - transistor density improves 35%/yr
    - » due to process shrink
      - there are some hidden dragon's here
  - die size increases by 10-20%/yr
  - transistor budget/chip increases 40-55%/yr
    - » today the limit is power rather than # of T's (more on this later)
- DRAM
  - capacity increases 40%/yr
    - » access latency increases at only 7%/yr however
- Disks get better in steps:
  - capacity CAGR ranges from 30% - 60%
    - » we're back to 30% now
  - latency virtually unchanged (for MHD's ~10ms)
    - » bandwidth significantly better however
    - » SSD's now on the scene with much better latency than MHD's

Performance

- Bandwidth vs. Latency
  - bandwidth is associated with throughput
  - latency is the response time for a single transaction
- Usually you care about how fast your job runs
  - for any job that takes more than a few seconds
  - also depends on market segment
    - » latency critical for real time constraints
    - » throughput critical for data-center or supercomputer apps
      - but your personal computer runs lots of processes too
      - tends to bias throughput importance
- Power wall has changed the industry to throughput
  - killer uP is now dead as is single thread performance
    - » Intel cancels Tejas in 2004
  - TLP, DLP, and multi-core
    - » throughput centric will persist (exception is embedded segment)

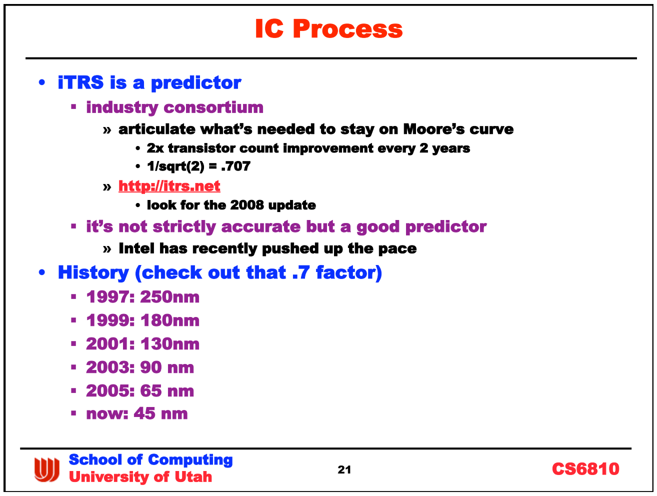
Bandwidth Optimization Results



© 2007 Elsevier, Inc. All rights reserved.

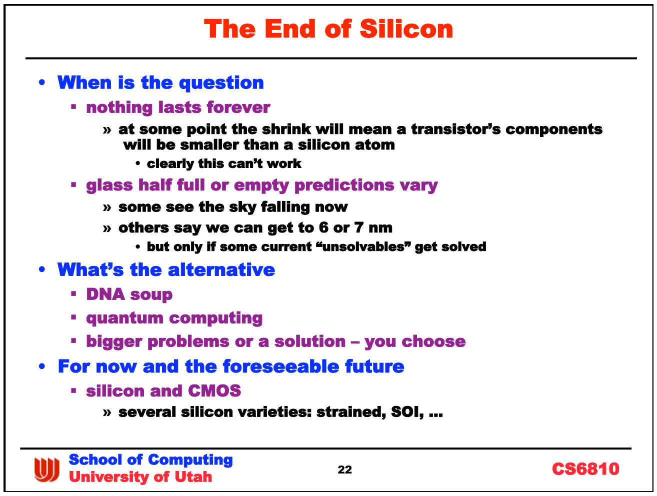
IC Scaling

- As feature size λ goes down
  - transistor speed scales as 1/λ
  - wire speed scales as αRC
    - » as wires get smaller – cross section decreases, R increases
      - aspect ratio changing somewhat to compensate
      - wires do not shrink as much as T's
    - » C goes down but not linearly
      - plate C improves but sidewall is an issue
      - sidewall C can be improved with process
        - – low-K dielectric and hot-wire air gap today
    - » wire speed for unrepeated wires is quadratic with length
      - proper repeater spacing makes wire delay roughly linear
      - at the expense of increased power for the repeaters
- Key result
  - wires are the problem (Ron Ho's PhD thesis is a great read)
    - » increasing contribution to power
    - » scaling poorly w.r.t. transistors



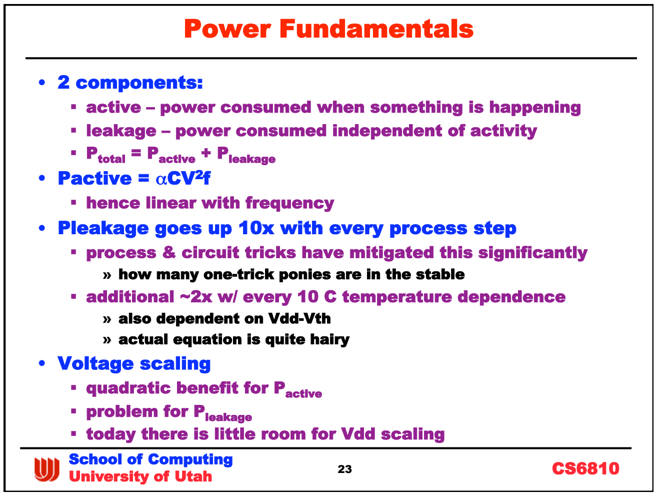
IC Process

- iTRS is a predictor
  - industry consortium
    - » articulate what's needed to stay on Moore's curve
      - 2x transistor count improvement every 2 years
      - 1/sqrt(2) = .707
    - » http://itrs.net
      - look for the 2008 update
  - it's not strictly accurate but a good predictor
    - » Intel has recently pushed up the pace
- History (check out that .7 factor)
  - 1997: 250nm
  - 1999: 180nm
  - 2001: 130nm
  - 2003: 90 nm
  - 2005: 65 nm
  - now: 45 nm



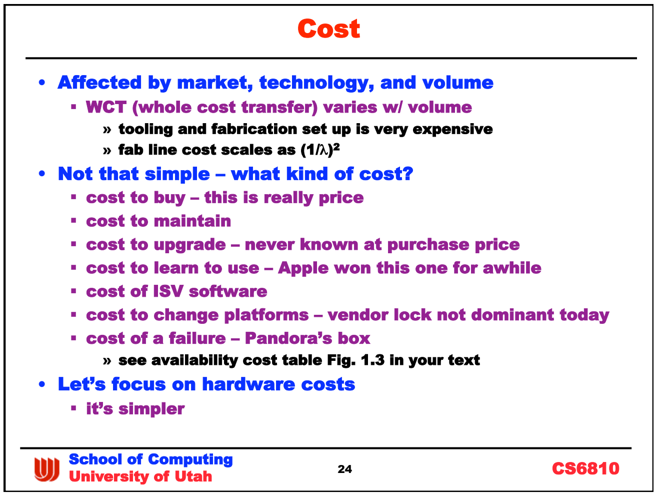
The End of Silicon

- When is the question
  - nothing lasts forever
    - » at some point the shrink will mean a transistor's components will be smaller than a silicon atom
      - clearly this can't work
  - glass half full or empty predictions vary
    - » some see the sky falling now
    - » others say we can get to 6 or 7 nm
      - but only if some current "unsolvables" get solved
- What's the alternative
  - DNA soup
  - quantum computing
  - bigger problems or a solution – you choose
- For now and the foreseeable future
  - silicon and CMOS
    - » several silicon varieties: strained, SOI, …



Power Fundamentals

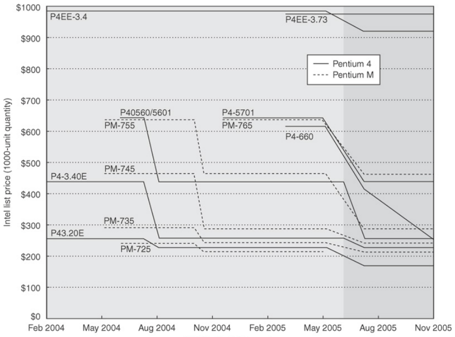
- 2 components:
  - active – power consumed when something is happening
  - leakage – power consumed independent of activity
  - $P_{total} = P_{active} + P_{leakage}$
- Pactive = αCV^2f
  - hence linear with frequency
- Pleakage goes up 10x with every process step
  - process & circuit tricks have mitigated this significantly
    - » how many one-trick ponies are in the stable
  - additional ~2x w/ every 10 C temperature dependence
    - » also dependent on Vdd-Vth
    - » actual equation is quite hairy
- Voltage scaling
  - quadratic benefit for P_{active}
  - problem for $P_{leakage}$
  - today there is little room for Vdd scaling



Cost

- Affected by market, technology, and volume
  - WCT (whole cost transfer) varies w/ volume
    - » tooling and fabrication set up is very expensive
    - » fab line cost scales as $(1/\lambda)^2$
- Not that simple – what kind of cost?
  - cost to buy – this is really price
  - cost to maintain
  - cost to upgrade – never known at purchase price
  - cost to learn to use – Apple won this one for awhile
  - cost of ISV software
  - cost to change platforms – vendor lock not dominant today
  - cost of a failure – Pandora's box
    - » see availability cost table Fig. 1.3 in your text
- Let's focus on hardware costs
  - it's simpler

Learning Curve: Process Matures



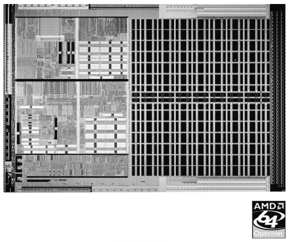
Cost of an IC

- More integration → IC is bigger piece of the total

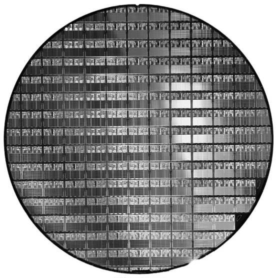
$$\text{IC-cost} = \frac{\text{Die-cost} + \text{Die-test-cost} + \text{Die-package-cost}}{\text{Final-Test-Yield}}$$

- DRAM prices have very small margins
  - range from $20/die until end of life at ~$2
    - » DRAM die are put on DIMMs (8 -19/Dimm)
    - » you buy DIMMs
- IC's traditionally 25-50% of WCT for desktop box
  - monitors and external disks may actually dwarf this cost
  - depends on system

Wafers vs. Chips



Dual Core Opteron



117 Opteron Wafer
90nm

Die Cost

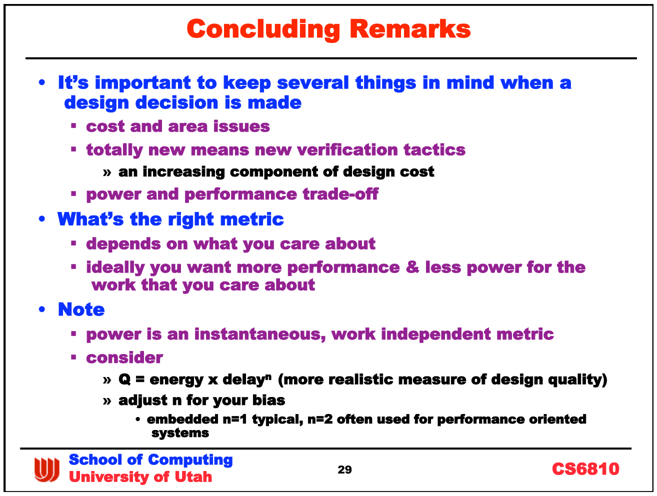
$$\text{Cost-of-die} = \frac{\text{Cost-of-wafer}}{\text{Dies-per-wafer} \times \text{Die-yield}}$$

$$\text{Dies-per-wafer} = \frac{\pi \times (\text{Wafer-diameter}/2)^2}{\text{Die-area}} - \frac{\pi \times \text{Wafer-diameter}}{\sqrt{2 \times \text{Die-area}}} - \text{Test-dies-per-wafer}$$

square peg in round hole term

$$\text{Die-yield} = \text{Wafer-yield} \times \left(1 + \left(\frac{\text{Defects-per-unit-area} \times \text{Die-area}}{\alpha}\right)\right)^{-\alpha}$$

- α depends on process
  - good estimate for α in 2006 is 4.0
- defects are very rare these days
  - yield is very near 100% on a mature process



Concluding Remarks

- It's important to keep several things in mind when a design decision is made
  - cost and area issues
  - totally new means new verification tactics
    - » an increasing component of design cost
  - power and performance trade-off
- What's the right metric
  - depends on what you care about
  - ideally you want more performance & less power for the work that you care about
- Note
  - power is an instantaneous, work independent metric
  - consider
    - » Q = energy x delay^n (more realistic measure of design quality)
    - » adjust n for your bias
      - embedded n=1 typical, n=2 often used for performance oriented systems