CS561 Organizational Issues

- ❑ Discuss syllabus
  - Introductions - office hours and access issues
  - Grades, late penalties, organization
  - Make sure you buy the right book
  - Provide some background on my attitudes and experience
- ❑ Why architecture is cool
  - combo between art and science
    - challenge to utilize the technology appropriately
    - still performance can be enhanced by human imagination
  - performance/cost is our Q factor
  - brief ramble on the history of the sport
  - food chain ideas - why the big focus on uP's
- ❑ It's a bad idea to be just HW or just SW

The Ride So Far

Basically a story of unprecedented improvement
Both a good and bad news story

- ❑ $1K today buys a gizmo better than $1M could buy in 1965.
- ❑ 1970's
  - mainframes dominated - performance improved 25%/yr
  - mostly due to improved architectures + some technology aids
- ❑ 1980's
  - VLSI + the microprocessor became the foundation
  - technology improves at 35%/yr so the ride accelerated
- ❑ Still there was a compatibility anchor
  - the dreaded object code compatibility problem

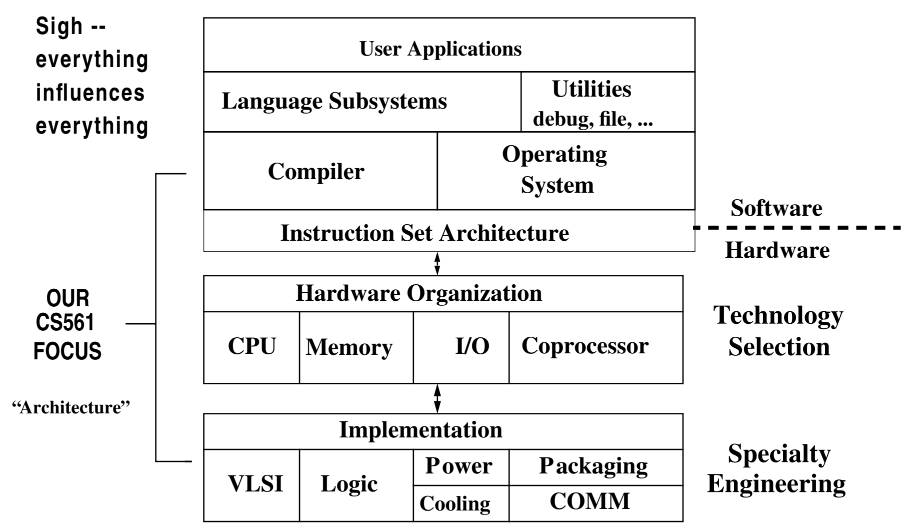
Machine Language Death = Opportunity

- ❑ Mostly with Unix and C in mid-80's
  - even systems programmers gave up on assembly language with a few exceptions of course
  - with this came the need for efficient compilers
- ❑ Compiler focus brought on the great CISC vs. RISC debate
  - with the exception of Intel - RISC won the argument
    - RISC performance improved by 50%/yr initially
    - with ILP and superscalar ideas things accelerated again
  - of course RISC isn't as simple anymore and the compiler is a key part of the game
    - doesn't matter how fast your computer is if the compiler wastes most of it due to the inability to generate efficient code
  - Last year things were 158% faster than the year before
    - Implies current track is 18% improvement per month!!

Aspects of Computer Design

- ❑ Many pieces of the game:
  - server - both file and compute
  - workstation - highly performance competitive but cost counts
  - PC - highly cost competitive but performance counts
  - mobile - fast enough and cheap enough but size and low-power count as well
- ❑ Bottom line is that it is a complex game
  - Determine important attributes
    - probably driven by a particular market segment
  - THEN maximize performance
  - WHILE staying within the cost and power constraints
- ❑ Classic conflicting constraint problem
  - and therefore it is inherently in the NP hard domain
  - PLUS - sadly there is no way to find the exactly right solution!!

Aspects of Design



Functional Requirements

- Application Area
  - General purpose ==> balanced performance over a variety of tasks
  - Scientific ==> floating point performance is the key
  - Commercial ==> Cobol and decimal arithmetic support, OLTP and DB performance key
  - Fault Tolerant ==> whoa!! replicated everything, lots of self checking support
- Level of SW Compatibility
  - At programming language - most flexible for architect - but still need new compilers
  - Object or Binary compatible - SW becomes easy - HW is hell since the anchor is out
- OS Requirements - may be imposed by market choice
  - Size of address space
  - Memory management - paged, segmented, or both
  - Protection ==> HW will need to assist
  - Real Time ==> whole new set of scheduling issues
- Standards - today's marketplace is standards bonkers
  - Floating point - fortunately today there is only one - IEEE 754
  - I/O - communication and peripheral (VME, SCSI, SCI, Hippi, ATM, ..., blah blah blah)
  - OS - Unix, DOS, or something from Bill Gates
  - Languages ==> affect instruction set significantly, COBOL affects the hardware in an adverse manner if you care how fast it runs.

Optimizing the Design

- ❏ Usually the functional requirements are set by the company/marketplace
- ❏ Today's requirements are a bit strange
  - cost minimized ==> simple design
  - performance maximized ==> complex design or better technology
  - time to market minimized ==> also favors simplicity
  - must beat the competition - but they're working hard too and they don't want to tell us what they're doing
- ❏ Oh - and you only get 1 shot
  - requires heaps of simulation
  - must be able to quantify everything
  - inherent requirement for deep infrastructure and support

Plus you must predict the Trends

- ❏ If you blow it you lose
  - IBM always provides such great pathological examples
  - e.g. Technology
    - $1 - $2 BILLION spent on Josephson Junction and 3D IC potential
    - then they scraped the program and picked CMOS like everybody else
  - e.g. Mainframes vs. microprocessors
    - IBM developed the microprocessor about when the other pioneers did
    - PC's were developed undercover - almost cancelled because they used Intel processors
    - Figured the 360/370 line could go on till 2000
    - If it hadn't been for the PC's the company may have gone under
    - Analysts estimate the mainframe fetish cost them 5 - 8 BILLION
- ❏ Key trends that must always be tracked
  - Usage patterns - e.g. the market
  - Technology - e.g. what you're going to build the sucker out of
  - Then there's cost and performance

Usage Trends

- Memory usage
  - Average program needs grow by 50% to 100%/year
  - Yikes - add an address bit each year
- Assembly language replaced by HLL
  - C a HLL? - NOT!!
  - increasingly important role of compilers
  - compiler and architecture types MUST now work together
- Whacked on pictures - even TV
  - some graphics capability is a must
  - soon we'll have Multi-media and the weather will be better
- Whacked on communications
  - I/O subsystems become a higher priority
  - VIOLA - the information super hypeway

Implementation Technology Trends

- Integrated Circuits
  - Density increases at 50%/yr .
  - Die size increases 10%-25%/yr
  - Combination is a chip complexity growth rate of 60%-80%/yr
  - Transistor speed increase is similar but signal propagation doesn't track this curve - so clock rates don't go up as fast
- DRAM
  - Density quadruples every 3 years (approx. 60%/yr) [4x steps]
  - Cycle time decreases slowly - 33% in 10 years
  - Interface changes have improved bandwidth however
- Magnetic Disk
  - currently density improves at 50%/yr
  - access time has improved by 33% in 10 years

Effects of Rapid Technology Trends

- Consider today's product cycle
  - Concept to production = 2 years
- AND market requirement of something new every 6 - 12 months
- Implications
  - Pipelined design efforts using multiple design teams
  - Experience from predecessor has a serious flaw
  - Have to design for a complexity target that can't be implemented until the end of the cycle
    - Can't afford to miss the best technology so you have to chase the trends somewhat
    - If there is a burp in the trends - you're hosed (Sun's Viking is a great example)
  - All of this ignores the infrastructure and NRE cost aspects of the process
    - e.g. Minimum cost to do a processor is $50,000,000
    - if the architecture changes much it is more like $200,000,000

Cost

- Clearly a market place issue
  - Profit as a function of volume
  - WCT (*whole cost transfer*) also a function of volume
  - The whole CS shtick is the cheap copy phenomena
- Not that simple - what kind of cost?
  - Cost to buy - the usual one that people consider
  - Cost to maintain
  - Cost to upgrade - never known at purchase time
  - Cost to learn to use - Apple won on this one for a while
  - Cost of ISV software
  - Cost to change platforms - the vendor lock
  - Cost of a failure - reliability = pandora's box
- Let's focus on hardware costs - it's simpler

Cost Impact

Time, Volume, Commodization, Packaging

- ❑ Due to the rapid trends - early use of technology is promoted
- ❑ Learning curve and subsequent process stabilization reduces costs over time
- ❑ Yield - is essentially a metric of where the learning curve is and how it progresses
  - Yield is % of manufactured chips that actually work
  - ==> things get cheaper with time till they hit 10-20% of initial
- ❑ Increasing cost of fab equipment has caused initial prices and end prices to rise slightly
  - BUT - on a per device basis (due to density and chip size increases) things get cheaper by a considerable amount