The EE-Department’s VLSI program at USC
by
Melvin A. Breuer
w/ contributions from
Peter A. Beerel
2013

Where is the VLSI program?

Three main areas of VLSI

How do we design a chip that has 1Billion transistors?

This way!

Simplified VLSI System Design Flow
Behavioral Level Synthesis

The next step is to take the system description as a collection of interacting machine descriptions or data and control flow graphs and produce the system architecture in terms of what components are used to perform required operations, how much sharing is desirable or possible, how the various operations are scheduled, where the intermediate results of computation are stored, how these components communicate with each other, etc.

Two main problems typically addressed in behavioral synthesis systems are allocation/binding of resources to processes, and scheduling of processes in terms of sequential and parallel computation.

The binding of variables to hardware entities includes defining the bit-width of various elements. Some bit-widths may be determined at processes, and scheduling of processes in terms of sequential synthesis systems are allocation/binding of resources to processes, and scheduling of processes in terms of sequential and parallel computation.

This material is covered in EE 599 and in more depth in EE 681.

Register Transfer Level Synthesis 2/2

The inputs to this step are a set of operations described as transfer of values between registers and functional units through a hierarchy of interconnect structures including latches and flip-flops. An important task performed in this step is the encoding and realization of the controller block (specifed as a finite state machine or a collection of interacting machines) in hardware. Here is where your knowledge of FSM’s is essential. Other processes generate hardware registers and an assignment of variables to registers. Finally an execution schedule is formulated so the processor knows exactly what operations and transfer from and to registers should occur at each clock period. This material is covered in EE 599 and in more depth in EE 681.
Physical Design 1/2

Physical design is the process of mapping the output from temporal and logic synthesis into a form that can be sent to a factory for manufacturing. The end product of physical synthesis is actually a set of masks in the form of a GDSII file. A mask is somewhat analogous to a film negative that has millions of black rectangles exposed on a transparent sheet. One main difference is that a set of masks costs several millions of dollars, so you do not get many chances to "capture that moment." Physical synthesis consists of several processes that are mostly automated, usually highly interrelated, but for simplicity, described here as separate entities, namely partitioning, floorplanning, placement and routing. Partitioning divides large segments of logic into blocks of logic, and where each block usually corresponds to a functional unit, such as a cache, register file, ALU, multiplexer, decoder, etc.

Floorplanning estimates the shape and area needed in silicon to implement each block, and places a representative layout of each block into a small rectangle.

Assignment (Placement)

A real floorplan

Routing

A circuit to be implemented on a chip

Gate C is assigned to this cell

A gate array or FPGA
An example of the Maze Routing algorithm

Physical Design 1/2

Placement takes the results of floorplanning and actually assigns each entity in each block to a unique area in a silicon chip, so that the estimate of wire length is minimal, thermal and power constraints are met, and IO ports are allocated. Finally the layout is routed, i.e., for each net, each port of the net is made electrically common using integrated wires, no wire of a net comes in contact with a wire of another net, a near minimal amount of wire is used, and constraints on crosstalk and delay are met. Physical design is covered in EE599 and EE680.

Testing

In 1956 Gordon Moore noted that a major reduction in the scaling of VLSI feature sizes occurs about every 18 months. Over the last few years we have gone from 65 nm in 2007 (a process node), to 45 nm in 2010 and 22 nm in 2012. Why didn't we just skip the 45 nm technology node and come out with 32 nm instead? Well the answer is that it is very difficult and costly to move from one process node to another in terms of research, development and manufacturing equipment. Simply put, it is very costly. A new fabrication plant costs over $1B. Thus, while researchers are currently working on 22nm and 14 nm nodes, their yields are still very low. That is, if one manufactured one million transistors, few if any would work as desired. Usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of&
USC VLSI Classes:
Circuits
- Analog/RF Circuits/Mixed Signal (EE479/EE448L/EE538ab/EE632)
  - Information is related to the value of a continuous electrical parameter such as voltage or current
  - Contains amplifiers, active filters, demodulation, mixing
- Digital Circuits (EE477/577, EE552)
  - Information is quantized
  - Usually binary: TRUE or FALSE value
  - Contains logic gates, memory elements, routing circuitry

Information is related to the value of a continuous electrical parameter such as voltage or current.

Contains amplifiers, active filters, demodulation, mixing.

Digital Circuits (EE477/577, EE552)
- Information is quantized
- Usually binary: TRUE or FALSE value
- Contains logic gates, memory elements, routing circuitry

USC VLSI Classes:
Modeling, Design, Physical Verification
- Component modeling (EE 560)
  - Hardware description languages (HDLs)
  - System design languages (emergent)
- Logic Synthesis and optimization (EE 681)
  - Refinement into interconnection of blocks
  - Blocks consist of library of gates
- Physical Design (EE 680, EE 477/577)
  - Transistor sizing, floor-planning, place-and-route (semi-custom)
  - Custom library design / hand/layout (full-custom)
  - Clock-tree design, optimization, and analysis
- Physical Verification (EE477/EE577)
  - Space-level simulation (full-custom and analog)
  - Static timing analysis (digital designs)
  - Noise and signal-integrity analysis; Power analysis

USC VLSI Design Classes
Manufacturing, Test, and Packaging
- Manufacturing (EE504L)
  - Mask/Wafer Fabrication
- Test (EE658)
  - Check chip parametrics
  - Check power/ground
  - Apply test vectors
  - Use built in structures to simplify test problem
  - Mark bad chips (to be thrown out)
- Test time and equipment very cost intensive

Mathematical Foundations for CAD of VLSI (EE581)
- Graph Theory; Logic, theorems and proofs
- Models of computational machines; BDDs
- Mathematical optimization
- No pre-requisites – highly recommended!!!

CAD Curriculum

<table>
<thead>
<tr>
<th>Approximate course content</th>
<th>581 (CAD 1)</th>
<th>680 (CAD 2)</th>
<th>681 (CAD 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic mathematical concepts and notation</td>
<td>Intro</td>
<td>Intro and preliminaries</td>
<td>Specifications/Modeling</td>
</tr>
<tr>
<td>Theorems and Proofs</td>
<td>Partitioning</td>
<td>Specifications/Modeling</td>
<td>BDDs</td>
</tr>
<tr>
<td>Basic Combinatorics</td>
<td>Placement</td>
<td>BDDs</td>
<td>Static timing analysis</td>
</tr>
<tr>
<td>Graph Theory</td>
<td>Routing</td>
<td>Mathematical programming</td>
<td>SYNTHESE</td>
</tr>
<tr>
<td>BSF</td>
<td>Routing</td>
<td>Mathematical programming</td>
<td>H5 co-design or Verification</td>
</tr>
<tr>
<td>Computational Approaches to Optimization</td>
<td>Static timing analysis</td>
<td>Specifications/Modeling</td>
<td>H5 co-design or Verification</td>
</tr>
<tr>
<td>Noise sources and analysis</td>
<td>Specifications/Modeling</td>
<td>Specifications/Modeling</td>
<td>H5 co-design or Verification</td>
</tr>
</tbody>
</table>

Pre and Co-requisites

<table>
<thead>
<tr>
<th>Basic mathematical concepts and notation</th>
<th>581 (CAD 1)</th>
<th>680 (CAD 2)</th>
<th>681 (CAD 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theorems and Proofs</td>
<td>Placement</td>
<td>BDDs</td>
<td>Static timing analysis</td>
</tr>
<tr>
<td>Basic Combinatorics</td>
<td>Routing</td>
<td>Specifications/Modeling</td>
<td>Specifications/Modeling</td>
</tr>
<tr>
<td>Graph Theory</td>
<td>Specifications/Modeling</td>
<td>Specifications/Modeling</td>
<td>Specifications/Modeling</td>
</tr>
<tr>
<td>BSF</td>
<td>Routing</td>
<td>Specifications/Modeling</td>
<td>Specifications/Modeling</td>
</tr>
<tr>
<td>Computational Approaches to Optimization</td>
<td>Static timing analysis</td>
<td>Specifications/Modeling</td>
<td>Specifications/Modeling</td>
</tr>
<tr>
<td>Noise sources and analysis</td>
<td>Specifications/Modeling</td>
<td>Specifications/Modeling</td>
<td>Specifications/Modeling</td>
</tr>
</tbody>
</table>

http://en.wikipedia.org/wiki/Integrated_circuit#Classification_and_complexity
MSEEVL - Master of Science in Electrical Engineering (VLSI DESIGN) 1/2

- A minimum cumulative grade point average (GPA) of 3.0 must be earned on all course work.
- Completing the normal requirements for the Master of Science in Electrical Engineering, with the following additional 4 required courses:
  1. EE 536a;
  2. EE 577a;
  3. EE 577b or EE 536b;
  4. EE 552.
If a student chooses to take EE 536b as well as EE 577b, the student may either count EE 536b as one of the courses for Area 2 or EE 577b as one of the courses for Area 1 or Area 3. (described below).
- The student must also take two courses from one of the following areas and one course from a second area:
  Area 1 “Tools”: CSCI 455x, EE 560, EE 577b (see above), EE 658, EE 680 and EE 981.
  Area 2 “Analog”: EE 448, EE 504L, EE 536b (see above), EE 537 and EE 980.
  Area 3 “Architectures”: CSCI 455x, CSCI 570, EE 557, EE 577b (see above), EE 659, EE 677.

MSEEVL - Master of Science in Electrical Engineering (VLSI DESIGN) 2/2

Thesis Option
The minimum requirement is 27 units; four of these units are to be thesis. At least 16 units, not including thesis, must be at the 500 level or higher, and at least 18 units must be in the major department. A total of not less than four nor more than eight units of 590 Research and 594ab Thesis must be included in the program.

The minimum thesis requirement in 594a is two units; in 594b, two units.

Degrees and majors

<table>
<thead>
<tr>
<th>EE577</th>
<th>EE558</th>
<th>EE579b</th>
<th>EE568</th>
<th>EE581</th>
<th>EE581b</th>
<th>EE536a</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLSI</td>
<td>VLSI</td>
<td>VLSI</td>
<td>CAD</td>
<td>VLSI</td>
<td>CAD</td>
<td>Signal</td>
</tr>
<tr>
<td>MSEE</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>CENG</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>VLSI</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>CAD*</td>
<td>O</td>
<td>R</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>or CAD**</td>
<td>R</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

* Intended for PhD students
** Intended for MS students
R-required
O-optional

Courses Complimentary to CAD and Testing

<table>
<thead>
<tr>
<th>EE581, 596, 658, 680, 681</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Science</td>
</tr>
<tr>
<td>CSCI 455x Intro. System Design</td>
</tr>
<tr>
<td>EE 504L Solid State Processing and IC lab</td>
</tr>
<tr>
<td>EE 536a (no degree)</td>
</tr>
<tr>
<td>EE 537 b Advanced A of A</td>
</tr>
<tr>
<td>EE 677 VLSI Architectures and Algorithms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EE 536b (required)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EE 457/557 Computer System Architecture</td>
</tr>
<tr>
<td>EE 653 Advanced Topics in Microarchitecture</td>
</tr>
<tr>
<td>EE 677 VLSI Architectures and Algorithms</td>
</tr>
<tr>
<td>EE 558 Intro. Nanoscience and Nanotechnology</td>
</tr>
<tr>
<td>EE 560 Advanced Topics in Microarchitecture</td>
</tr>
<tr>
<td>EE 980 EE 594ab Thesis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mathematics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math 432 Applied Combinatorics</td>
</tr>
<tr>
<td>Math 532 Combinatorial Analysis</td>
</tr>
<tr>
<td>Math 533 Combinatorial Analysis and Algebra</td>
</tr>
<tr>
<td>EE 504L Solid State Processing and IC lab</td>
</tr>
<tr>
<td>EE 536a, b Mixed Circuit IC Design</td>
</tr>
<tr>
<td>EE 577 VLSI Architectures and Algorithms</td>
</tr>
</tbody>
</table>

Courses Complimentary to VLSI circuits and architectures

<table>
<thead>
<tr>
<th>EE577, EE579b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Science</td>
</tr>
<tr>
<td>CSCI 455x Intro. Programming System Design</td>
</tr>
<tr>
<td>EE 504L Solid State Processing and IC lab</td>
</tr>
<tr>
<td>EE 536a, b Mixed Circuit IC Design</td>
</tr>
<tr>
<td>EE 653 Advanced Topics in Micronarchitecture</td>
</tr>
<tr>
<td>EE 677 VLSI Architectures and Algorithms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EE EP</th>
</tr>
</thead>
<tbody>
<tr>
<td>EE 457/557 Computer System Architecture</td>
</tr>
<tr>
<td>EE 457/557 System Architecture</td>
</tr>
<tr>
<td>EE 558 Intro. Nanoscience and Nanotechnology</td>
</tr>
<tr>
<td>EE 560 Advanced Topics in Microarchitecture</td>
</tr>
<tr>
<td>EE 980 EE 594ab Thesis</td>
</tr>
</tbody>
</table>

Testing (EE 658) in the VLSI Design Flow

- Design specification
- Behavioral description
- RTL description (HDL)
- Functional verification
- Logic synthesis using OPT-BIST
- Gate-level netlist
- Floor planning, place and route
- Physical layout
- Layout verification and implementation
- Test generation
- Testing
- Failure analysis
- Repair
- Less capability
- More capability
- Pass
- Fail
Resilient & Efficient Systems
Murali Annavarapu

Resilient Systems
- PARMA/MACAU: Can a cache design survive 7 years with multi-bit soft errors?
- WearMon: Detect failures before they occur?
- WrapDMR: Error detection in a dime (nearby fast)

Energy Efficient Systems
- XENON: How much has your server efficiency improved?

Resilient and Specialized Computing
Jeff Draper

Resilient Computing
- Two trends coupled together greatly exacerbate system reliability issues
  - Extensive variability in device parameters at extreme fabrication technology nodes
  - Extreme scale systems containing millions of cores
  - Solutions needed at all levels: software, architecture, circuits

Specialized Computing
- End of Moore’s Law will require value added in other ways for computing to keep progressing
  - Customized computer architectures are likely
    - e.g., D.E. Shaw’s Anton molecular dynamics computer

Research Plan
- Currently exploring energy-resilience trade-offs, memory-oriented computing, transactional memories, 3DCC-inspired architectures, and networks-on-chips as part of solution space

What are employers looking for?
- Demonstrated high quality verbal and writing skills.
- Excellent knowledge of VLSI systems and circuits (477, 577a.b)
- Good knowledge of one or two related areas such as analog circuits, asynchronous circuits, or VLSI manufacturing process
- A strong background in testing, and some familiarity of verification and validation
- A familiarity (1 course) with one application area such as signal and image processing, communication systems, or biomedical engineering
- An excellent knowledge of C++
- Some familiarity with Unix
- Some familiarity with using commercial CAD tools (477, 577a.b)
- Good communication skills

Asynchronous VLSI
Peter A. Beere

Async Template and Library Design
- Asynchronous ASIC Flow
  - Reuse standard tools for synthesis/physical
  - ClockFree - new asynchronous optimization/translation
  - Target ultra high performance/low power
- CAD for Asynchronous Design
  - Mathematical programming, graph theory, Petri-nets
  - Optimize pipeline structure - Slack matching
  - Optimize communication - restructuring conditional communication
- Async Template and Library Design
  - Advanced pipeline templates
  - Static timing and noise analysis
- Application Drivers
  - Turbo/Video decoding
  - Computational blocks
- Beyond Moore’s Law
  - Robustness/reliability to process and environment variability faults

Resilience
- STFB 2.2GHz Test Chip
- High-Level Specification

CYBER-PHYSICAL SYSTEMS GROUP
Paul Bogdan

Addressing challenges in medicine via bio-inspired approaches
- Detection of silent disease progression
- Drug delivery to inaccessible body areas
- Accurate modeling & steady interaction with medical experts improves healthcare

Research Team
- Memory-oriented computing micros
- Machine learning algorithms with medical experts
- Sensors & Mode/Parameter Identification
- Sensing & Analytic App
- Sensor and Analytic App
- Sensors & Analytics App with medical experts
- Composable & Control
The problem
The yield of present (and future) technology nodes is (and will be) low. It starts at about zero and during a learning period of several years gets to 85%. If when the yield is at 35% we could, by re-design, enhance it to be 50% then time to market can be drastically reduced. This can be achieved via redundancy.

The solution
By duplicating of even triplicating non-memory modules on a die we increase its area and to a greater extent its yield. Thus the wafer has less die but more good ones and hence larger yield/area.

Findings
• We have developed several algorithms for identifying the optimal number of spares per module.
• We have developed a system to partition a design in an optimal way to form modules that will be replicated.
• We conjecture that replicating modules at the core level is not optimal; such replication should be done at a finer level of granularity.

Microarchitecture, Multiprocessors, Reliability
Michel Dubois

Memory Protocols for CMPs
• Better cache protocols for transactional memory and bulk consistency models
• Speculative protocol transactions in TMs

Parallel Simulations of CMPs on CMPs
How to exploit current chip multiprocessors to simulate future chip multiprocessors

Tools for Measuring Resilience to Soft Errors
• Parma, Parma+: Estimate reliability of caches under Temporal and Spatial MBEs
• Extension to main Memory

Extremely Low cost Error Correction in Memories
CPRC: Correctable Parity Protected Cache
CIEC: Chip-Independent Error Correction (in DRAMs)

Power Generation/Measurement/Management
Young H. Cho

GENERATION
• Multimodal Harvesting
• Flexible and Scalable
• Highly Efficient Conditioning
• Industrial Wireless Sensor Network

MEASUREMENT
• On-chip/On-board Measurement
• High Accuracy and High Sampling Rate
• Digital/Instrumentation/LOW Overhead
• Account for Process Variation
• In-Situ Dynamic and Static Power

MANAGEMENT
• Scalable Smart Power Grid Management
• High Throughput/Low Latency
• Software Defined Network Driven

VLSI circuits and testing
Gupta’s research interests are in the area of VLSI testing and design and he is currently involved in projects on test and validation of deep submicron circuits, testing multicore systems-on-silicon, and delay testing and diagnosis of digital circuits.
Design of Energy-Efficient Information Processing Systems

Massoud Pedram

Energy Efficiency and Our Digital Future
- Moore’s law continues although it is increasingly more costly to stay on path (higher variability, lower supply, voltages, less timing margin)
- There is a move away from overclocking and performance-driven design toward use of adaptive voltage supply levels, including next/sub-threshold computing
- Key factor of merit is the amount of useful work done per joule of energy dissipated in circuits (jattitude/index, MIPS/Watt, energy-aware product)
- Cost-sensitive, heterogeneous, distributed computing platforms have emerged, operating under diverse requirements and targeting different markets
- Future system architectures will be heterogeneous multi-core designs, with integrated accelerators “sub-cores” connected via heterogeneous interconnect elements
- Adaptability will also be a key requirement since it enables a system to provision and customize hardware to meet varying workload and performance requirements

Example Solutions

Parallel and Distributed Computing

Viktor K. Prasanna

Multi-core/Heterogeneous
- Multithreaded applications
- Software routers
- Graph analytics
- GPU computing
- Scalable parallel algorithms

Application Specific Architectures
- High-Speed networking
- Terabit IP forwarding
- Virtualized routers
- Security firewalls
- Data center networks
- Energy efficiency

Big Data Platforms and Applications
- Time series graph analytics
- Cloud resource management
- Social network analysis
- Analytics for smart infrastructure (smartgrid, smart oil fields...)

http://ceng.usc.edu/~prasanna

Cloud Computing & Energy Informatics

Yogesh Simmhan

- Scalable Software Frameworks for Clouds & Distributed Systems
  - Streaming dataflow & workflow models for realtime data processing
  - ‘Big Data’ analytics on Timeseries Graphs
  - Algorithms, Scheduling, Programming
- Abstractions, Resiliency, Data Mgmt.

SMART Interconnects Group

Timothy M. Pinkston

Power-Performance-Resource Efficiency

Efficient communication among hundreds to many thousands of CMP cores is needed in near future:
- Network-on-chip architectures designed to reduce power consumption, maximize performance and require minimal resources while remaining resilient in presence of faults
- Multi-scale system interconnects that satisfy growing, diverse quality-of-service demands
- Techniques: multi-objective optimization, proactive power gating, performance-guarded/critical design

Holistic Design and Management
- Mutually aware processor-memory-interconnect performance and require minimal resources
- Cross-layer Power Optimization Opportunities
- Efficient communication among hundreds to many thousands of CMP cores is needed in near future:
  - Power gating, performance-guarded/critical design

And you must keep on learning

And you must keep on learning

Cauligi Raghavendra

Interconnection networks and fault tolerant computing

Raghavendra’s current research focuses on wireless and sensor networks and delay/disruption tolerant networks. His pioneering work is on energy efficient MAC, routing, and broadcasting protocols. He has worked on energy resource management in a number of sensor network applications including data gathering and situational awareness. He has worked extensively on parallel and distributed systems, interconnection networks, and fault tolerant computing.
The End

Item #1: How to prove theorems

- Direct
- Inductive
- Contradiction
- Contrapositive
- Exhaustive
- "Because I say so"
- "Oh, it’s trivial"
- You can finish the rest by yourself

Item #2: Mathematical Programming

This is a drawing of George Dantzig, my teacher at UC Berkeley, and the father of Linear Programming. It consists of a tour over a graph using a heuristic solution to the traveling salesperson problem. Yes, this class covers graph theory, linear programming and many other algorithmic techniques for solving combinatoric problems.

Item 2a: Another solution technique

How does Divide and Conquer help these teenage heroes solve large combinatoric problems? Well, I’m not sure, but I sure know how to use it to solve problems on a computer. Do you?

Item #2b: Dynamic programming—no this is not a new computer language
Item #3: Graph theory
- Trees
- Forests
- Cycles
- Steiner
- Spanning
- Chords
- Directed
- Color
- Tours
- Paths
- Dual
- Rectangular dual

Item #2c: Backtrack programming
Have you ever gotten lost while hiking? If so, did you backtrack along your path until you got to a fork in the trail, and then took the other branch in the road? Well if you did, you did the wrong thing. You should have stayed put and waited to be rescued. But in this class you are allowed to backtrack.

Item #5: Counting, enumeration and generating functions
You may think that you know how to count, but you really don’t. For example, how many completely specified Boolean switching functions are there of n variables that are not degenerate, i.e. are not a function of less than n variables.

Item #6a: Annealing as an example of a heuristic
How do we simulate the annealing process used in making crystals and steel to solve VLSI combinatoric problems of partitioning, placement, floorplanning and routing? Well if you don’t know, come a learn about it.

Item #6b: Genetics as an example of a heuristic procedure
What does simulating the process of mating, crossover and mutations have to do with solving CAD problems? Well, if you are at least 18 years old, come find out.

Item #6c: A fast solution technique
Did you know that some problems can be solved optimally using a greedy algorithm; other can be solved sub-optimally using a greedy heuristic. In EE581, heuristics are not algorithms, except for Al Gore who is a heuristic approximation to ..