California Institute of Technology
Department of Computer Science
Electronic Design Automation

CS137a, Winter 2004

Big Ideas in EDA

Wednesday, March 10

- Human time (brainpower, productivity) is the limiting resource to developing and implementing new physical implementations of computations (new hardware, hardware-software systems,...).
- We increasingly leverage the human's limited time and brainpower resources by:
  - Raising the abstraction level at which he works (taking some of the details away from him)
  - Filling in the abstracted details by machine ("automatically")
- Problems are quickly large and complex – both exceeding human ability to keep track of all the details and provide good (even correct) solutions and machine's abilities to solve them optimally (*e.g.* superlinear and mostly NP-hard).
- Abstractions are good—helping contain the amount of detail which must be considered simultaneously. But don't let artificial abstractions be barriers to performance and optimization. When abstractions get in the way (hide performance, prevent optimization), break them and reformulate better ones to replace them.
- Decomposition is often necessary to make problem tractable for both human and computers. Decomposition, nonetheless, usually sacrifices solution quality (optimality).
- Divide-and-conquer is often useful both to separate different concerns (types of problems) and to reduce the size of a single problem.
- The optimal solution to two subproblems independently is seldom the optimal solution to the composite problem—nonetheless, the decomposition is often valuable or necessary.
- It is important to implement the *semantics* of the original specification, but we may use any implementation which preserves the semantics. That is, *transformation* which changes the form and details of the logic is fine as long as the important, observable behavior is retained.
- Exploit freedom in things that do not matter to correctness in order to reduce costs.
- Since problems are often NP-complete (or worse), the goal is often to find "good" or "adequate" solutions to otherwise intractable problems.

- Problems often have a *dominating* effect or feature—that is a feature largely responsible for the first order cost. In such cases it is often adequate (necessary) to optimize primarily or exclusively for the dominating effect. (for the EEs: Think of this like the a dominating pole or zero in a transfer function) Counterwise, problems where multiple effects may dominate a solution are harder to solve well than those with one or two dominating effects.

- "Optimal" solution will depend on the costs involved, changing as costs change. It is seldom viable to "solve" a problem independent of the costs involved, though pieces of a problem can sometimes be solved independent of costs.

- Often the "goodness" of a solution can be bounded in polynomial time. That is, using approximation algorithms or dominating effects, it is often possible to identify a bound on the best possible implementation. This can be very useful in knowing how well a heuristic is doing, including understanding when one reaches the point of diminishing returns.

- Formulate simple cost models which capture the key/important effects.

- Common problems often exhibit special or restricted structure which make the problem easier to solve: (1) able to solve faster, (2) able to obtain better quality solutions. Exploit and characterize this structure. Often our weak general bounds on how well or fast a problem can be solved are result of characterizing the problem poorly (*e.g.* ignoring key variables).

- Because problems often exhibit structure which we have not fully capture in our formulation, there is a strong experimental element to EDA. However, developing an adequate theory is important so that both people and machines can predict quality, performance, and resource requirements without performing an optimization and so that we know how close we are to the optimum solution. So, closing the gap between experimental results and theory is also an important goal.