VLSI CAD:
Logic to Layout

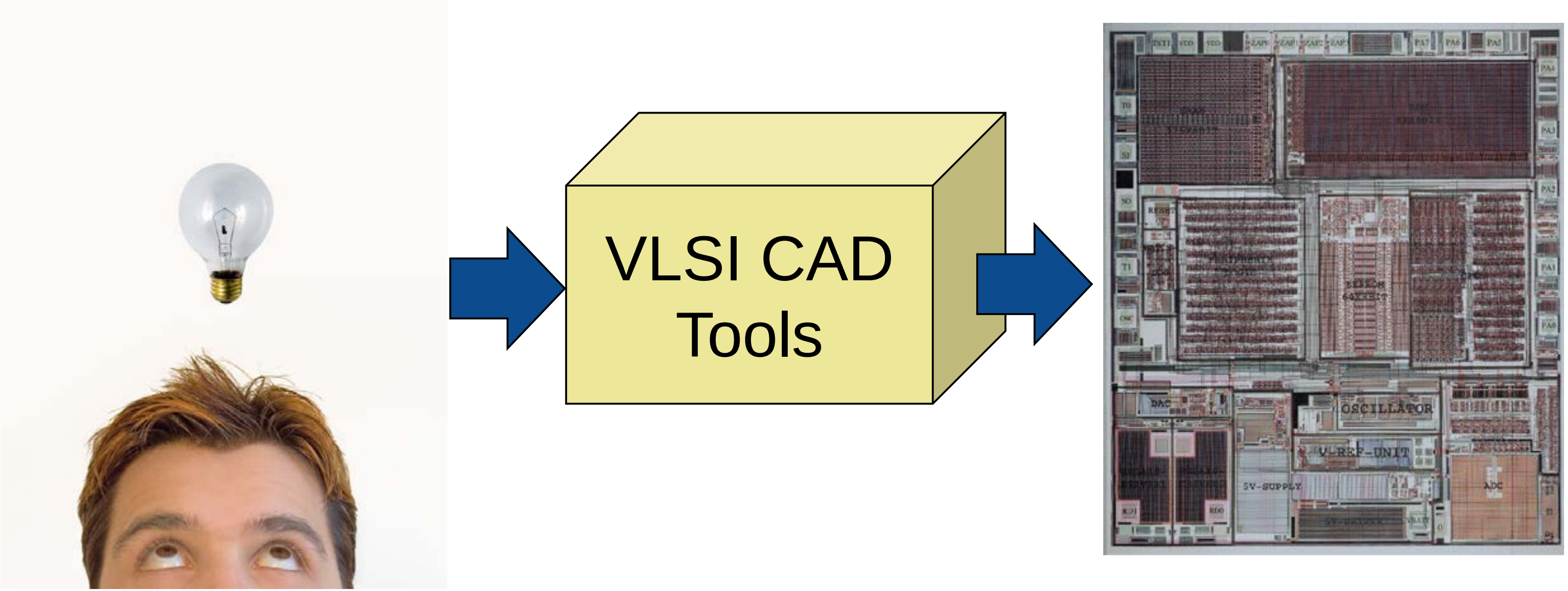
Part 1 Logic
Lecture 1

Welcome & Introduction

Rob A. Rutenbar
University of Illinois

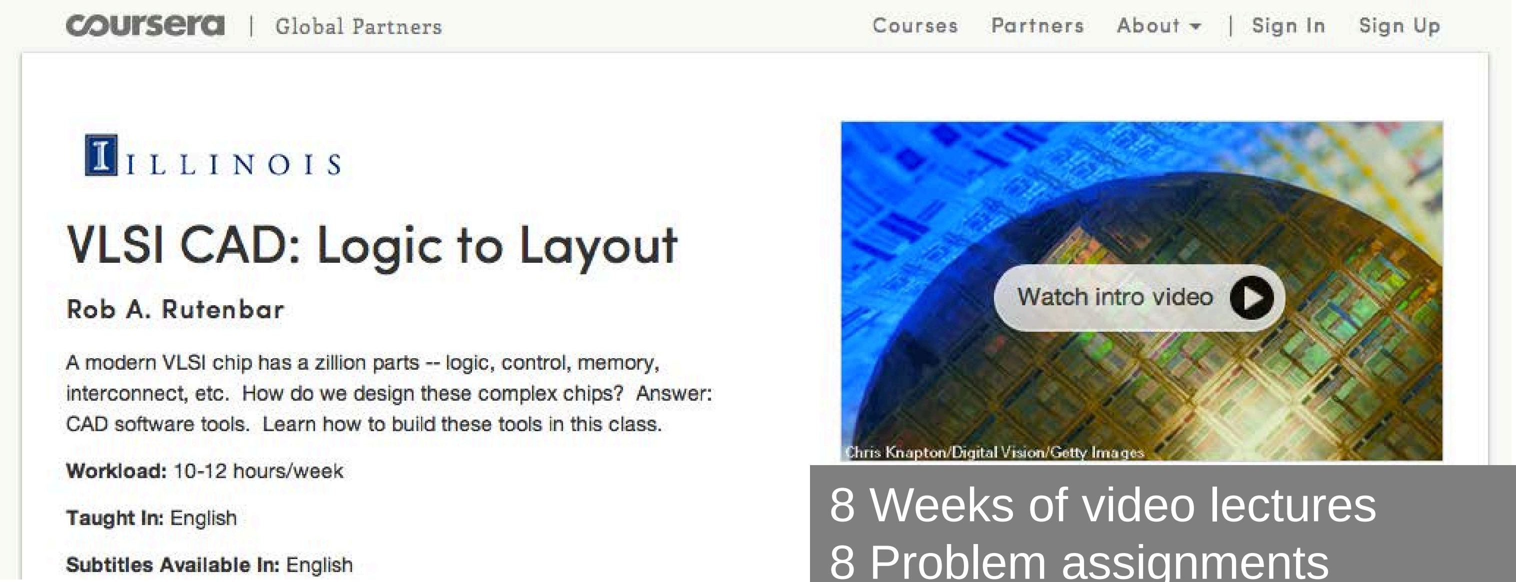
Welcome!

- **What we are about in this sequence of classes…**



A Little Bit Of History…

- **Original version of VLSI CAD sequence was one 10 week MOOC**
  - Emphasized **CAD flow**, from **logic** topics (Boolean stuff) to **layout** topics (geometry)



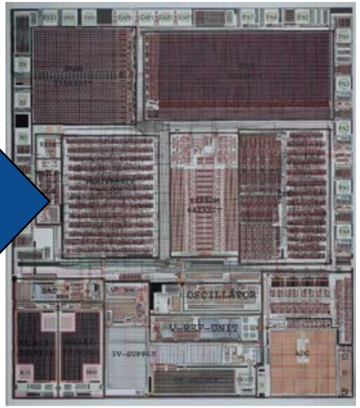
8 Weeks of video lectures
8 Problem assignments
4 Optional Programming tasks

Today: Two MOOCs, Two Parts

You are **HERE!**

VLSI CAD 1
LOGIC

VLSI CAD 2
LAYOUT



- **Each of our two parts is half of the original, longer, single MOOC**

Aside…

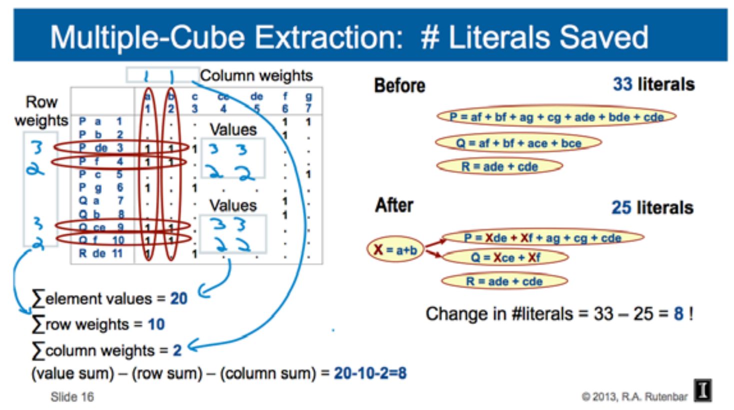
- **This is why the intro slides on lectures say "Logic to Layout"**

- **…and, this is why the lectures are numbered continuously…**
  - Lectures **1 2 3 4 5 6 7 8**
    - Form **Part 1 LOGIC** topics
  - Lectures **9 10 11 12**
    - Form **Part 2 LAYOUT** topics

What's In A Video Lecture?

1. Title: Content

2. "Talking head" intro



3. Deep tech content, live writing

Class Logistics

- **5 weeks = 4 weeks of lectures + 1 free week + final exam**
- **Videos every week**
  - 2-3 hours in total
- **4 Problem Sets (i.e., homework assignments)**
  - 4 weeks of video material → 4 assignments
  - Leave a week open at the end for you to finish all your work
- **2 Programming Assignments (Optional, Honors Assignments)**
  - Some conventional coding
  - Some ‘scripts’ run thru CAD-centric tools running on our servers

About Grading

- **Mastery based**
  - Means you get **multiple submissions** on the assignments, which are each **randomly changed** each time you retry
  - You need to **pass** all the assignments individually, to pass this course
- **Problem sets**
  - 4 weeks of lectures, and 1 problem set for each week, should take about 1 week
- **Final exam**
  - At the end of class, looks like a problem set, but it’s comprehensive over course

- **See the class web site for details about the logistics…**

About Grading

- **Programming Assignments**
  - **Optional!**
  - These are Honors Assignments – do them if you want (1) a **deeper engagement** with the technical material, and (2) a **job** in the VLSI CAD / electronic design automation industry (where most people build **software** as well as algorithms).
- **Mechanics**
  - We provide **realistic inputs** that model each problem, as a readable file. We tell you a simple **ASCII file format** your software needs to use, to generate output.
  - You **upload** your output to the Coursera web site, and we **auto-grade** it, and also give you some feedback on your solution.
  - You run your code on **your** computer. You can use **any** language, **any** platform.

Other Important Stuff

- **Honor code**
  - OK to talk with and work with other people in the class
  - BUT – what you submit must be **your own work**, for homework and for any code
  - AND – please do **NOT** post solutions to any assignments on Coursera site, or share these solutions face to face, in email, via the web, with others in this course

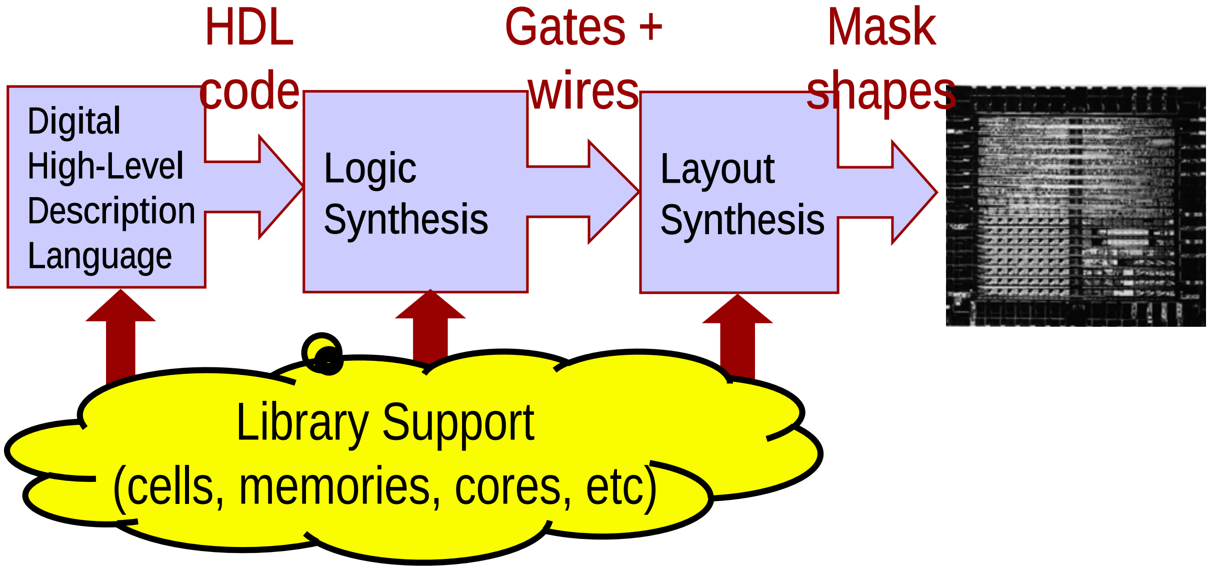
- **Use Coursera interaction mechanisms**
  - Coursera supports discussion forums to ask questions, etc.
  - We will make use of these to help connect you to us (and to each other)

What Background Do You Need?

- **Computer science**
  - Basic programming skills
  - Data structures
- **Computer engineering**
  - Basic digital design (gates, flip flops, Boolean algebra, Kmaps)
  - Combinational and sequential design (finite state machines)
- **Mathematics**
  - Discrete: Basic sets, functions, careful notation
  - Exposure to graph theory is nice but not essential
  - Continuous: Basic calculus, derivatives, integrals, matrices
- **Basic VLSI knowledge**
  - Some chip layout exposure is nice, but not essential

So What is the Course All About...?

- **CAD for semi-custom ASICs**
  - **ASIC** = application-specific integrated circuit
  - **Semi-custom** = try to design reusing some already designed parts
  - **CAD** = flow through a sequence of design steps and software tools



Some Useful Acronyms

- **Semi-custom ASIC**
  - **Application-specific IC** - design a chip for a specific task, using mostly semi-custom techniques
  - Do not expect to make a billion of them, so cannot afford full custom
  - Not quite as dense (transistors / area) or as fast (GHz) as full custom
- **Semi-custom vs. full-custom**
  - **Semi-custom:** designs mostly from pre-existing parts (gates, memories)
  - **Full-custom:** designs right down at the individual transistor level
  - Today, only things like microprocessors are "full custom"
  - And in fact, even these chips have huge semi-custom parts on them

One More: CAD vs. EDA

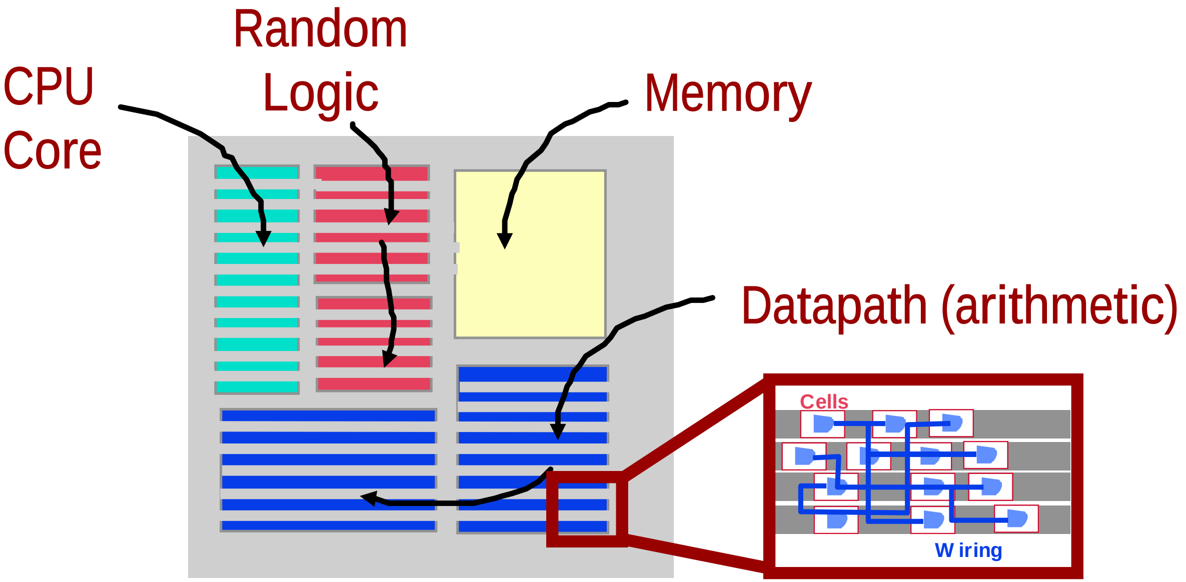
- **CAD: Computer-Aided Design**
  - What we all used to call this world of tools for chip design
  - Problem: other people do "CAD" too, like mechanical engineers, architects, etc.

- **EDA: Electronic Design Automation**
  - What most "insider" chip folks call it. More accurate, more descriptive name
  - Problem: people outside the business not always clear what it means.

- **So, I called this class "VLSI CAD," but it's really "VLSI EDA"**

More Acronyms: System-on-a-Chip ASIC

- **SOC: Integrates many blocks of function on one big chip**
  - Most common: row-based standard cells = gates + flops in rows; and big SRAM memories; and perhaps pre-designed blocks like CPUs



Example: Small SOC Controller Design

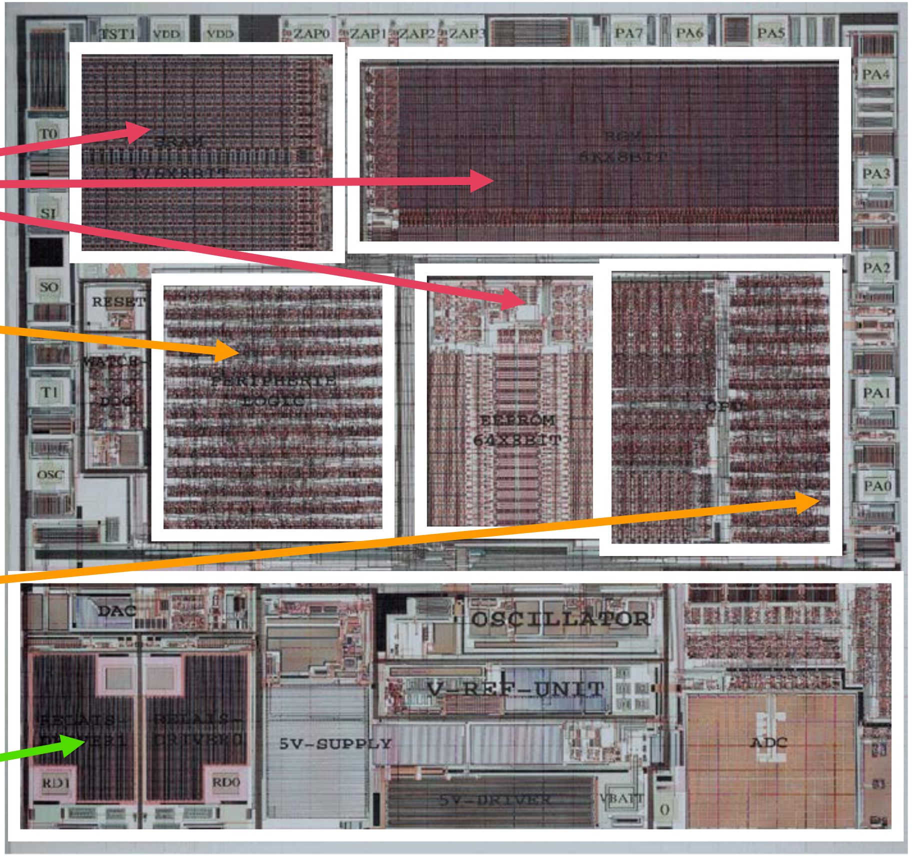
- **Look at blocks**

Memories

Random control logic

CPU core

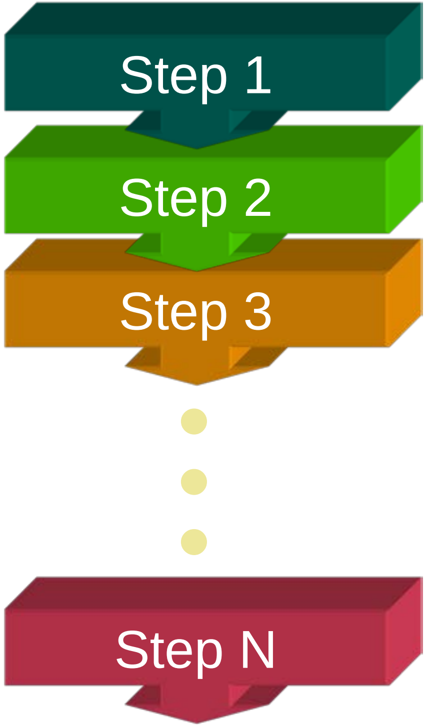
Analog interface to external world



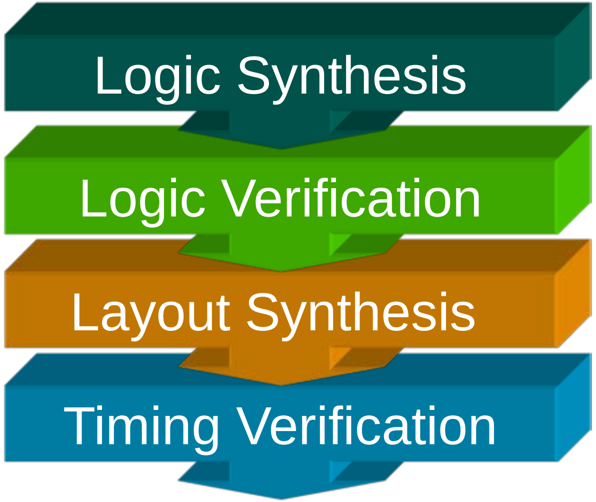
Another Important Term: CAD Flow

- **How to attack big designs like these?**
- **Big idea: Levels of abstraction**
  - Break problem down into smaller steps
  - Each step renders design a little more real
- **Synthesis steps:**
  - Go forward in design: Make new stuff
- **Verification steps:**
  - Look backward: Check that it worked

- **Complete set of steps called: A Flow**

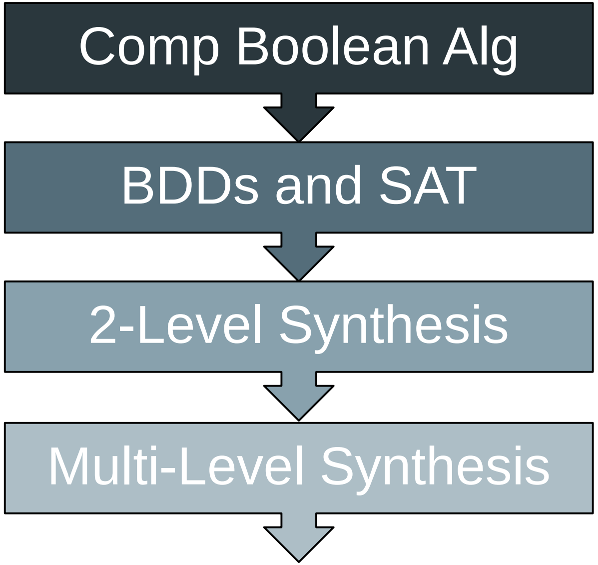


Our Class CAD Tool Flow Over 2 Parts



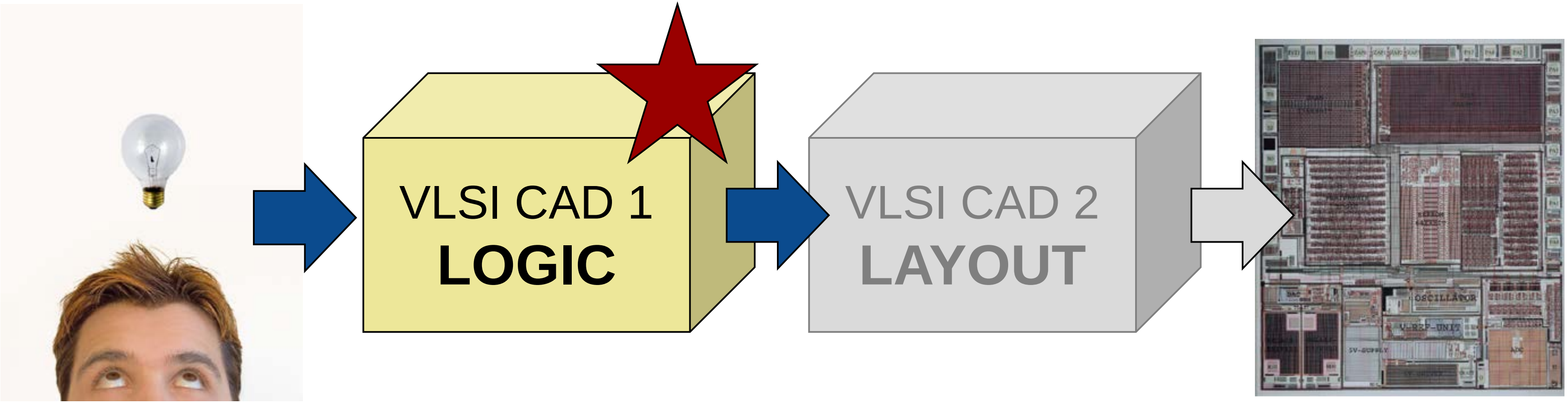
- **Start with some Boolean / logic design description…**
- **…end with gates+wires, located at (x,y) coordinates on chip**
- **Part 1 LOGIC focuses on the top two steps in this flow**
- **Big goal(s) for both classes**
  - Explain the critical algorithms, data structures & modeling assumptions used in each of these big steps

What Topics are in Part 1: LOGIC



- **Four big topics**
- **Computational Boolean Algebra**
  - Boolean eqns like computational objects
- **Boolean Verification**
  - Critical methods: BDDs and SAT
- **2-Level Logic Synthesis**
  - Logic synthesis for AND/OR structures
- **Multi-Level Logic Synthesis**
  - Logic synthesis for the general case

You Are Now Starting Part 1: **LOGIC**



- **Here we go….!**