

# Digital Systems II

EEC 180

Lecture 1

Bevan M. Baas

Tuesday, March 30, 2021

# Today

---

- Course details
  - Policies
  - Grading
  - Schedule (web page)
  - Course objective and strategies
- My background & some recent work in many-core processor arrays
- Digital Systems Overview
- Basic Units
- 7 Basic Diagrams
- Chip Design Methodologies

# A Few Announcements

---

- No lab this week
- FPGA board pick up
  - Wed 12:30 – 2:30
  - Fri 10 – 1pm
- FPGA boards are being sent out this week
- Please excuse many zoom hiccups
  - Lectures, Labs, Office hours

# Teaching Assistants

---

- Harichandana Chitturi
- Satyabrata Sarangi
- Shifu Wu
- Terry O'Neill
- Tony Tsoi
- Ziyuan Dong

# Course Workload

---

- 5 unit course
- Upper division
- New ways of thinking of things requires effort
  - Algebra: use variables
  - Calculus: no concrete solutions for indefinite integrals
  - Digital Design (EEC 180)
    - HDL—a new way of writing “code”
    - Autonomous hardware: datapaths, memory, control
    - Sophisticated controllers
- Passing this course requires significant time and effort

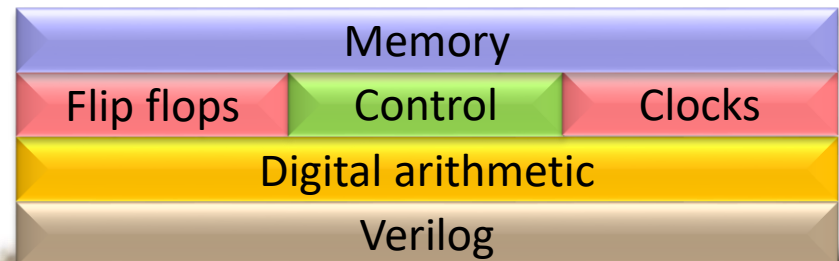
# My Teaching Philosophy

---

- Primary goal (mine and yours):  
*Learn digital system design well*
- Achieve this through:
  - Reading textbook
    - Chapter material, Example design problems
    - Book is comprehensive and quite broad
  - Lectures
  - Solving problems on paper (homework)
  - Solving problems and building things in lab
  - Discussions with other students, TAs, myself

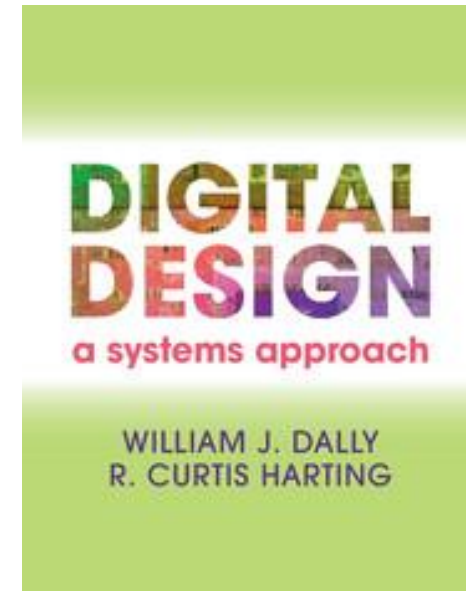
# Course Plan: Tools and Methodologies for Large-Scale Digital Designs

- Introduction
- The Verilog hardware description language (HDL)
- Digital arithmetic
  - Number formats
  - Addition / Subtraction
  - Later in quarter: Multiplication, saturation, and rounding
- Flip-Flops
- Control circuits: counters and general FSMs
- Clocks
- Interface circuits
- Memories



# Textbook

- Digital Design, A Systems Approach  
William Dally & R. Curtis Harting  
Cambridge University Press  
2012 edition (not Draft or VHDL version)
- Begin reading:
  - Chapter 1
    - *The Digital Abstraction*
    - Prerequisite material
  - Chapter 2
    - *The Practice of Digital System Design*
    - High-level background
  - Chapter 3
    - *Boolean Algebra*
    - Prerequisite material





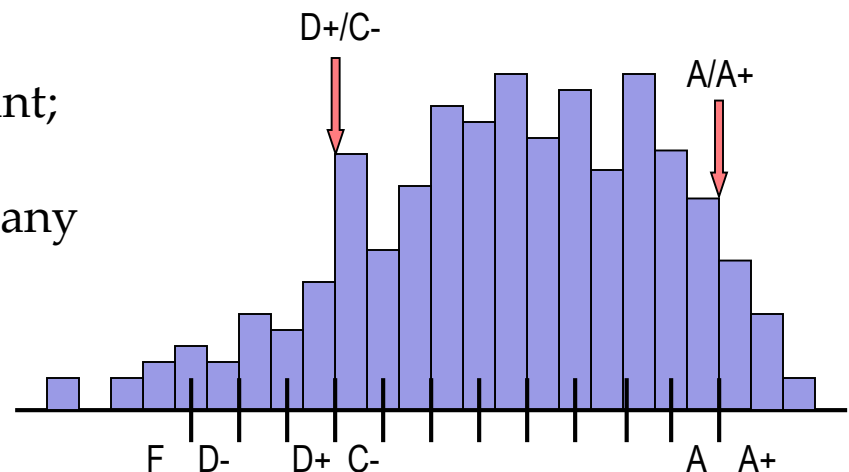
# Grading Philosophy

---

- Grading serves two main purposes:
  1. Motivate you to do the work required to learn
    - Reading textbook (quizzes)
    - Lectures (quizzes)
    - Solving problems in homework (exams)
    - Solving problems in labs (lab grading, exams)
    - Discussions with others
  2. Give others an indication of how well you know the material

# Letter Grade Assignments

- I assign a letter grade for only the final course grade
- You can see score statistics for each graded item on Canvas
- I look at the final exams and course record of the class and assign two key dividing points: the A/A+ and D+/C- boundaries, and assign course grades from there using equally-sized intervals
  - No required numbers of any particular letter grades
  - Absolute scores are not important; the boundaries shift according to the difficulty of the exams in any quarter
  - Ignore any letter grades you might see on canvas



# Lab Grades

---

- Each lab section is graded by the TA assigned to that section
- For many reasons, the same lab work will be given slightly different grades for students in different sections
  - We will do whatever is possible to minimize this such as having detailed rubrics for problems
- To eliminate much of this bias, at the end of the quarter, the average for each lab section will be calculated and the averages for the  $N-1$  lower sections will be increased so all sections have the same average
- This issue is not present for all other graded work

# Lectures

---

- Ask questions at any time
  - Please ask questions verbally out loud when you hear a pause (in zoom)
  - Avoid chat messages as I usually do not see them
- Be respectful of others
  - Hold conversations outside of class
  - Silence phones
  - Sit near the door if you come in late or need to leave early

# Course Announcements

---

- In class
- Web
  - Assignments, etc.
- Email via Canvas Announcements
  - Time-critical announcements only

# Questions

---

- In class
- In lab
- Office hours
  - Tentatively Mondays 3-4pm. Please talk to me or email me today if that is not convenient.
- After both lectures
- See me in zoom rather than through email
- TAs
  - Lab
  - Office hours
  - Email

# Working With Others

---

- Collaboration
  - Asking questions and explaining principles produces better work and dramatically increases learning
  - Working with others
    - Do homework and prelabs with classmates nearby
    - Ask each other questions, help each other—regarding **principles**, and **approaches to solving** only
  - See *Course Collaboration Policy* on web page
- Dishonesty
  - Copying produces similar work, stunts learning, is not fair to honest students, and is not allowed in this course
    - Students engaged in dishonest work will be referred to Student Judicial Affairs
    - I will try to keep in-class exams honest
    - Steps will be taken to keep out of class work honest

# Penalties for Violating the *Policy on Student Conduct and Discipline*

---

- Penalties
  - Minimum penalty: meetings with SJA officer, zero grade on work, record with SJA
  - Permanent F grade on your transcript, no credit for the class
  - One to three quarter suspension from the university
  - Permanent dismissal from all ten campuses of the University of California. Permanent notation on your transcript.



# Penalties for Violating the *Policy on Student Conduct and Discipline*

---

- Several perspectives
  - Personal                      obvious reasons
  - ECE and UCD                (especially for those inclined to share work with someone doing poorly in class)  
Cheating harms our major and university's reputation among employers who interview our graduates.
- In summary: The purpose of the penalties and me mentioning them is so that no one will get one!!! Don't do anything that violates the Policy on Student Conduct!

# Penalties for Violating the *Policy on Student Conduct and Discipline*

---

- Typical scenario:
  - Someone shares code/design with another
  - They get caught
  - The “Copier” feels terrible guilt for causing a friend to get a zero
  - The “Sharer” deeply regrets sharing resulting in a zero when he/she should have had a full score

# MOSS

---

- “Measure Of Software Similarity” tool
- Utilizes very sophisticated and fast algorithms
- Processes all  $Order(N^2/2) = N(N-1)/2$  pairings
  - Ex: examination of 300 submissions includes 44,850 pairwise comparisons
- Runs are done for each lab of this year’s work combined with work from past years

# MOSS Demonstration Case

Modified version

Original version

- Added, deleted, changed all comments
- Changed all variable names
- Reordered modules
- Reordered lines of code within modules
- Changed equivalent logic

- 91% similarity for submission 1
- 91% similarity for submission 2

| submission1.v (91%) | submission3.v (91%) |
|---------------------|---------------------|
| 5-137               | 78-214              |
| 144-168             | 220-245             |
| 185-192             | 5-12                |
| 195-210             | 14-33               |
| 212-224             | 35-47               |
| 225-240             | 48-60               |

```
submission1.v
>>> file: paint.v
module paint (
    input [9:0] x ,
    input [8:0] y ,
    input [9:0] box_x ,
    input [8:0] box_y ,
    other
    input [2:0] color_select,
    input [1:0] shape_select,
    output [11:0] rgb
);

reg valid ;
reg draw ;

wire [3:0] red ;
wire [3:0] green ;
wire [3:0] blue ;

reg [4:0] x_address;
reg [4:0] y_address;
wire [31:0] rom_data;

rom rom1 ( .addr(y_address) , .data(rom_data) );

submission3.v
>>> file: update.v
module update (
    input [9:0] test1 ,
    input [8:0] test2 ,
    input [9:0] test2_vx ,
    input [8:0] test2_vy ,
    input halt,
    output reg [9:0] next_test1,
    output reg [8:0] next_test2,
    output reg [9:0] next_test2_vtest11,
    output reg [8:0] next_test2_vtest22
);

// variable declaration
//test1
//test2

always @ (*) begin

    if (halt == 1'b0) begin
        next_test2_vtest11 = 10'b0000000001;
        next_test2_vtest22 = 9'b0000000001;
    end
end
```

# MOSS

---

- Key take-away messages:
  - 1) MOSS is amazingly good at spotting pairs of submissions that share a common design
    - This meshes very well with the course collaboration policy
  - 2) Follow the course collaboration policy and you have nothing to worry about
  - 3) Violate the course collaboration policy and you *will* have something to worry about

# Demo Your Own Design

---

- A hash code of your verilog will be recorded during your demo
- Do not modify your code at all before uploading to canvas—your hash codes must match exactly to receive credit
- Yes, unfortunately a few students tried demoing someone else's code in past quarters

# Exam and Quiz Regrades

---

- Some number of exams and quizzes will be scanned before being returned
- Key take-away messages:
  - Do not change anything on your work if you request a regrade
  - Past students have tried this and got in big BIG trouble!!!

# Cheating Websites

## chegg, coursehero, slader, etc.

---

- The university has recently taken a very strong stand against paying for work (2-quarter suspension for first offense recently)
- Key take-away messages:
  - Do not post any course material
  - Of course do not use any unpermitted outside material in work you submit
  - Of course do not post solutions
  - Multiple students have and got caught!!!



# Advice From Last Year's 180 Students

---

- *What advice do you have for future 180 students?*
- 29% — Designing process
  - Spend more time designing a circuit before trying to implement it in Verilog.
  - Make sure & diagram everything above a design before writing any verilog code.
  - Design intensively before coding (draw diagrams), use a testbench because it's much more efficient than testing on the board. Review EEC180A concepts!
  - DRAW diagrams. Ask questions about syntax (be paranoid), actually write testbenches while writing the code. Think like circuits, not programs.
  - The big thing with Verilog/HDL coding is you NEED to think a lot about the problem before starting code. Draw the circuit, pipeline diagram, and timing diagram before writing your code.
  - Work on testbench while doing the design. It helps.
  - ModelSim is your best friend for debugging.
  - Start thinking about the design of lab early so you can break the whole lab into parts. Schedule time to finish each part accordingly.

# Advice From Last Year's 180 Students

---

- *What advice do you have for future 180 students?*
- 29% — Lab-related
  - Start labs early and divide modules into submodules for easy testbenching.
  - Spend a lot of time on labs: learn A LOT from them.
  - Start labs early.
  - Organize time for the labs.
  - Work hard on the labs.
  - Give yourself more time for labs, don't look at prelab right before lab.
  - Start labs early before the lab section because a lot of questions come up during the process. It's best to have questions in the beginning than in the end when lab section ends and your ways of getting answers are harder now.
  - Start the labs as early as possible!!!

# Advice From Last Year's 180 Students

---

- *What advice do you have for future 180 students?*
- 14% — Book related
  - BUY THE BOOK AND READ IT! Also, make a design first then start coding. Make submodules and write testbenches for each (as you go).
  - Read the book!!
  - Read the book and get your lab work done ASAP!
  - Read the book, start labs early
- 28% — Misc
  - Learn the differences between a reg and a wire and where/how to use them right away.
  - Understand timing very well.
  - Take notes in class and read lecture notes after. Start labs quickly so more questions can be asked during lab time.
  - etc.

# Advice From Past TAs: Finishing Labs

---

- “some students leave their lab work until the last minute because they tend to think that the code will work without any issues in the span of four hours.”
- “some students do not test their designs using ModelSim waveforms; instead, they either ask me what could possibly be wrong with their code or they will attempt the trial and error method by changing the code in small increments and hope that it all works out.”
- “many students are spending very little time on paper design”
- “When they ask me for help, I notice that their bugs are usually caused by bad planning, not having a plan or not understanding the lab in the first place.”

# Advice From Past TAs: Finishing Labs

---

- “...students don’t test their designs in modelsim using test benches, rather they try to verify their designs on the board itself. I have repeatedly stressed that it would be better if they did the former first.”
- “...students in my section probably left a lot of implementation for the last moment. Some of the students in the other sections who were present in my lab today had very little work done as well.”

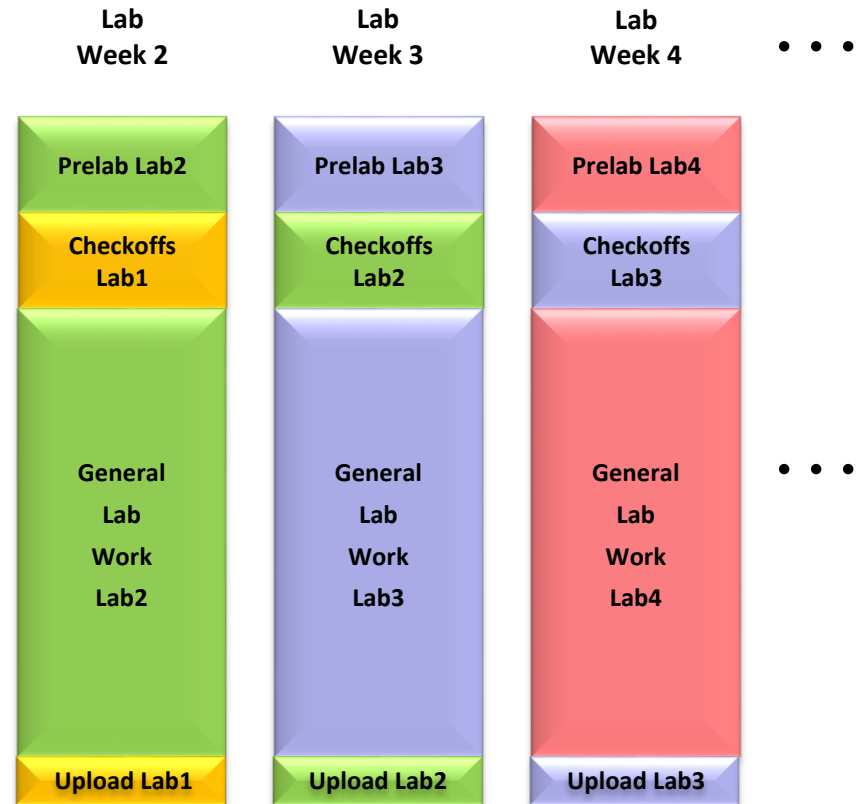
# Advice From Past TAs: Lab Checkoffs

---

- *“There is too little time in lab for questions because checkoffs take too long”*
- “I've made it clear that I expect check off sheets and lab reports to be printed before check offs begin, but every lab I have many students who do not do this before lab.”
- “Students also spend a lot of time setting up their simulations so I've asked students to have their simulations already open and ran but still the majority of students will not.”
- “I stayed an extra hour yesterday after my lab section but the majority of my students leave right after checkoff. I mostly end up helping students from other sections after checkoff.”

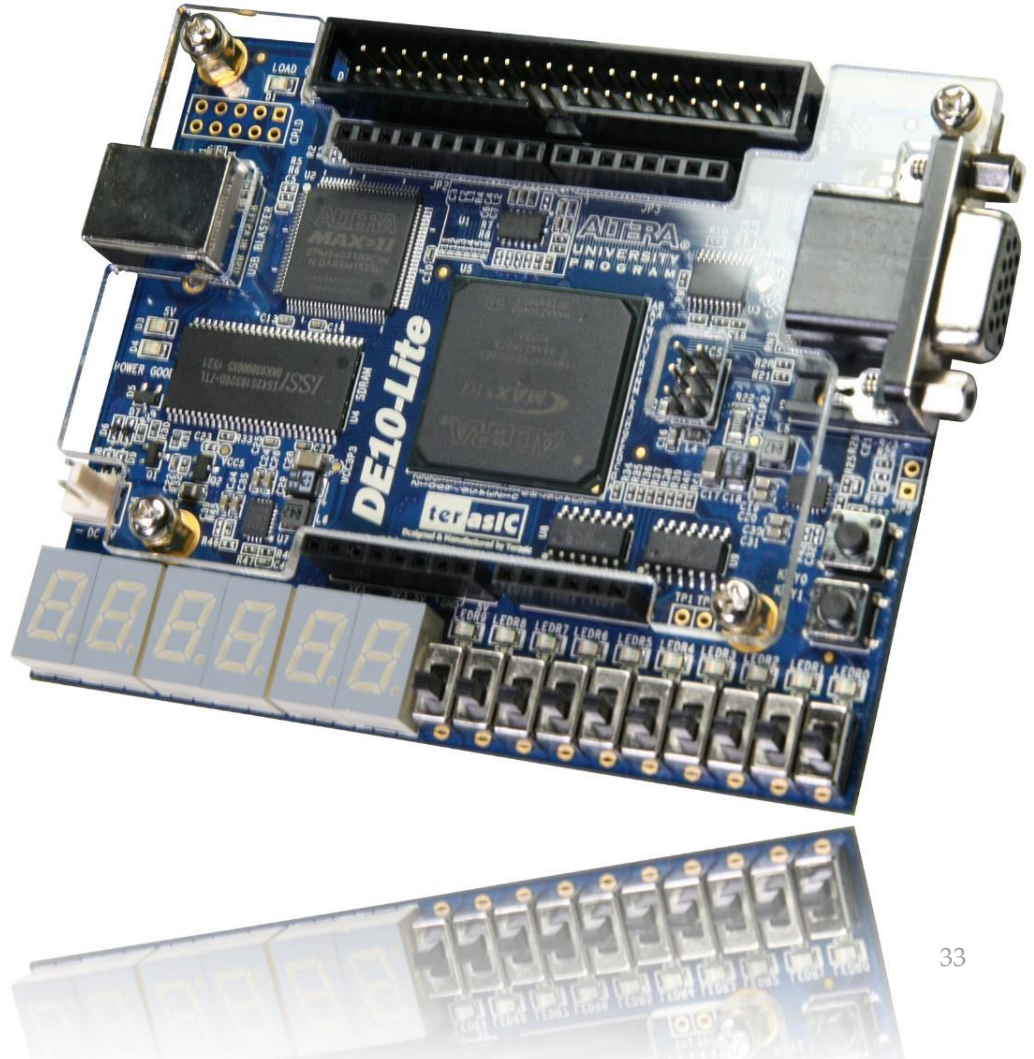
# Laboratory

- Each lab period contains three main sub-periods:
  - 1) Upload your Prelab 10 mins before your lab starts  
*Give your Prelab to your TA as soon as you arrive and it will be checked-off and returned in the first ~10 minutes of the period*
  - 2) Next, any final checkoffs for the previous week's lab can be done and then the Lab Reports for the previous week's lab are due. Lab Reports submitted after this deadline are considered late
  - 3) Lastly, work on the lab to be done that week



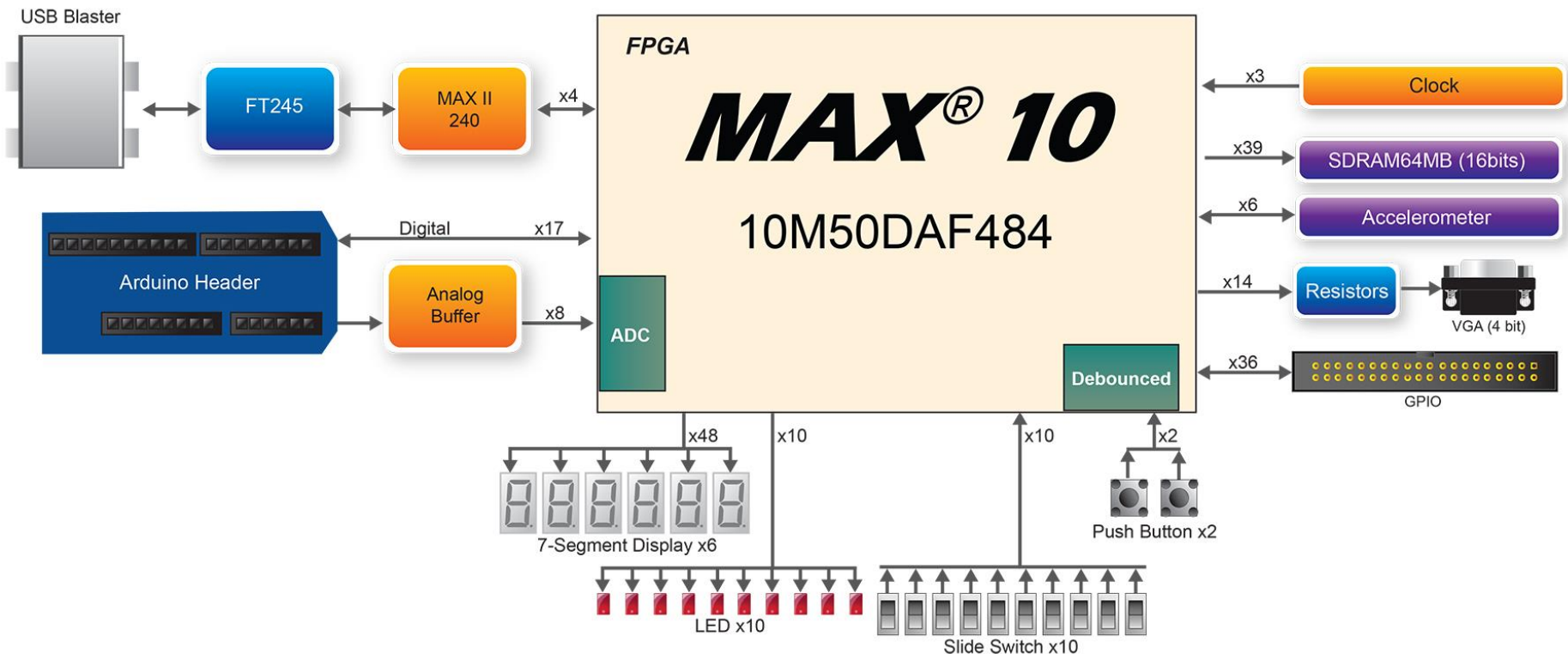
# Laboratory: DE10-Lite FPGA Board

- Altera/Intel FPGA
- 50K programmable logic elements
- 1.6 Mbits M9K memory  
8192 bits/block (9216 incl. parity)
- 144 18x18 multipliers
- 4 PLLs
- On-Board
  - 10 LEDs
  - 10 slide switches
  - 2 push buttons
  - 6 7-segment displays
  - USB
  - 64 MB SDRAM
  - Accelerometers
  - low-resolution VGA





# Laboratory: DE10-Lite FPGA Board



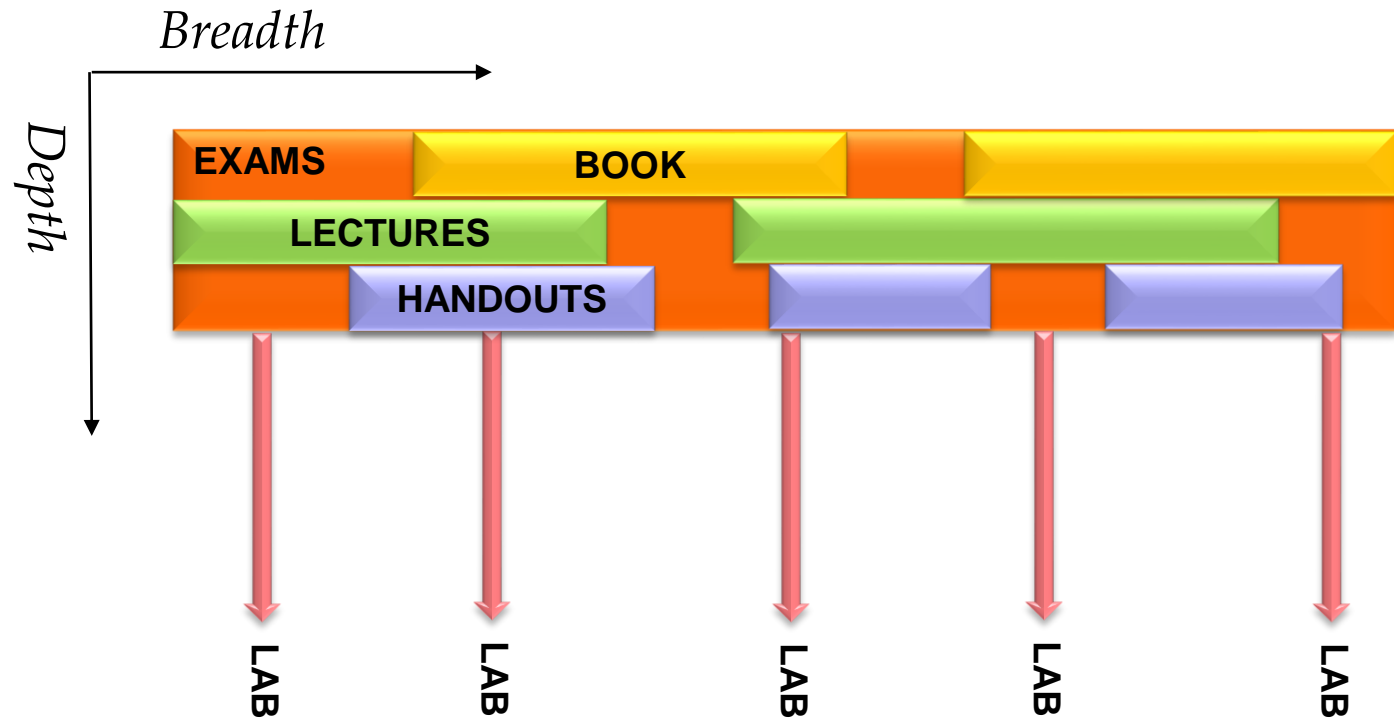
# Book + Lectures + Handouts + Laboratory

---

- The main body of material is presented in the book, lectures, and handouts
- Generally speaking, the labs *complement* the main material
  - They go into a much greater depth on specific topics
  - They give design experience
  - They give significant practical application of theory
- The Quiz, Midterm, and Final Exam generally focus on the main body of material

# Book + Lectures + Handouts + Laboratory

- Breadth and Depth



# Handouts

---

- The handouts are copies of most of my lecture notes
- The material in the Handouts posted on the course web page includes some of the most foundational material
- There is a **pretty high expectation** that you understand this material and are proficient in applying it
- When the handouts or lectures and the textbook disagree on something, follow the handouts and lectures

# Course Web Page

---

- Everything except grades will be posted on the main course web page
- <http://www.ece.ucdavis.edu/~bbaas/180/>
- There is a link on the canvas course web page

# Advancing CMOS Technologies

- Moore's "Law" (Observation) was made in 1965 and notes that transistor density ~doubles every year (every 1.5 years now)
- "Cramming more components onto integrated circuits," Gordon Moore, *Electronics*, April 19, 1965.

The experts look ahead

## Cramming more components onto integrated circuits

With unit cost falling as the number of components per circuit rises, by 1975 economics may dictate squeezing as many as 65,000 components on a single silicon chip

By Gordon E. Moore

Director, Research and Development Laboratories, Fairchild Semiconductor division of Fairchild Camera and Instrument Corp.

The future of integrated electronics is the future of electronics itself. The advantages of integration will bring about a proliferation of electronics, pushing this science into many new areas.

Integrated circuits will lead to such wonders as home computers—or at least terminals connected to a central computer—automatic controls for automobiles, and personal portable communications equipment. The electronic wristwatch needs only a display to be feasible today.

But the biggest potential lies in the production of large systems. In telephone communications, integrated circuits in digital filters will separate channels on multiplex equipment. Integrated circuits will also switch telephone circuits and perform data processing.

Computers will be more powerful and will be organized in completely different ways. For example, memories built of integrated electronics may be distributed throughout the

machine instead of being concentrated in a central unit. In addition, the improved reliability made possible by integrated circuits will allow the construction of larger processing units. Machines similar to those in existence today will be built at lower costs and with faster turn-around.

### Present and future

By integrated electronics, I mean all the various technologies which are referred to as microelectronics today as well as any additional ones that result in electronics functions supplied to the user as irradicable units. These technologies were first investigated in the late 1950's. The object was to miniaturize electronics equipment to include increasingly complex electronic functions in limited space with minimum weight. Several approaches evolved, including microminiaturization techniques for individual components, thin-film structures and semiconductor integrated circuits.

Each approach evolved rapidly and converged so that each borrowed techniques from another. Many researchers believe the way of the future to be a combination of the various approaches.

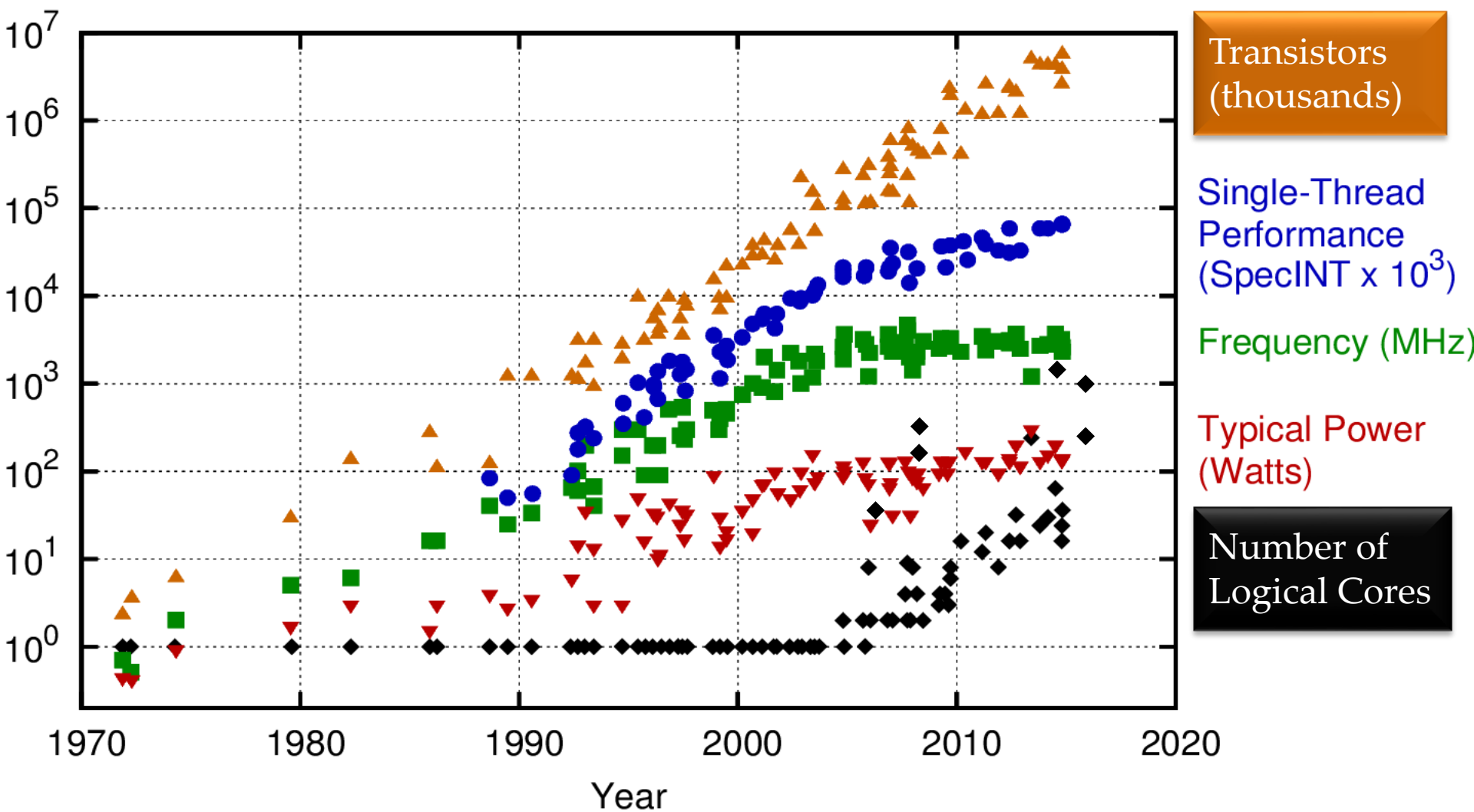
The advocates of semiconductor integrated circuitry are already using the improved characteristics of thin-film resistors by applying such films directly to an active semiconductor substrate. Those advocating a technology based upon films are developing sophisticated techniques for the attachment of active semiconductor devices to the passive film arrays.

Both approaches have worked well and are being used in equipment today.

### The author

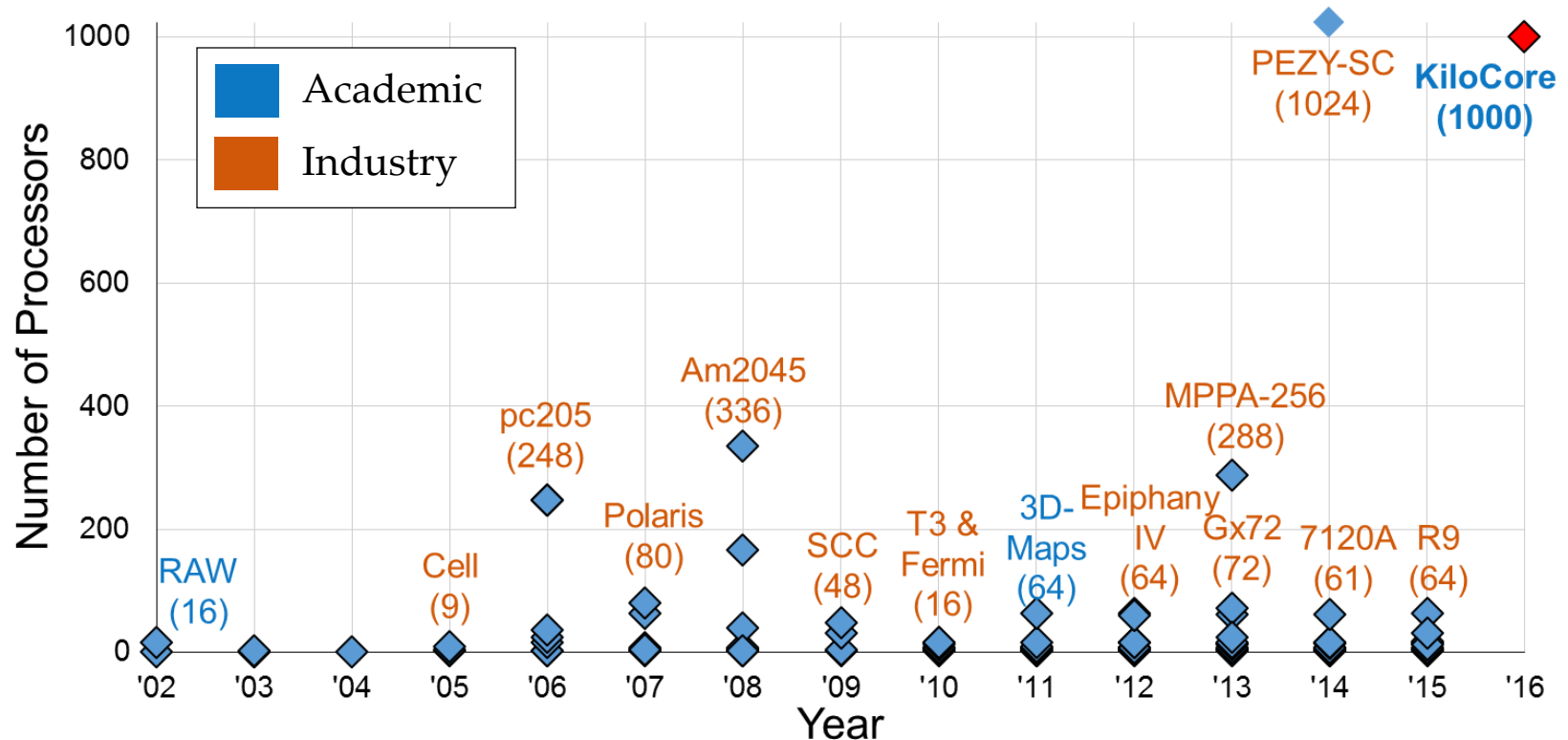
Dr. Gordon E. Moore is one of the new breed of electronic engineers, schooled in the physical sciences rather than in electronics. He earned a B.S. degree in chemistry from the University of California and a Ph.D. degree in physical chemistry from the California Institute of Technology. He was one of the founders of Fairchild Semiconductor and has been director of the research and development laboratories since 1959.

Electronics, Volume 38, Number 8, April 19, 1965



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten  
New plot and data collected for 2010-2015 by K. Rupp  
New data added by B. Baas

# Number of Processors on a Single Die vs. Year



Note: Each processor capable of independent program execution  
© B. Baas