VLSI Digital Signal Processing

EEC 281

Lecture 1

Bevan M. Baas

Tuesday, January 6, 2025

Today

- Administrative items
- Syllabus and course overview
- My background
- Digital signal processing overview
- Read *Programmable DSP Architectures, Part I* by E. A. Lee

Course Communication

- Email
  - Urgent announcements
- Web page
  - `http://www.ece.ucdavis.edu/~bbaas/281/`
- Office hours
  - After lecture Monday
  - After lecture Wednesday

Course Workload

- **4** unit **graduate** course
- This course requires significant effort and time
  - Multi-disciplinary field coverage
    - DSP algorithms
    - Digital processor architectures
    - Arithmetic
  - Utilizes robust industry-standard CAD tools (but we will make use of only the core essential features)
    - Verilog Cadence NC Verilog, Simvision
    - Synthesis tool Synopsys Design Compiler
    - Matlab

Course Readings

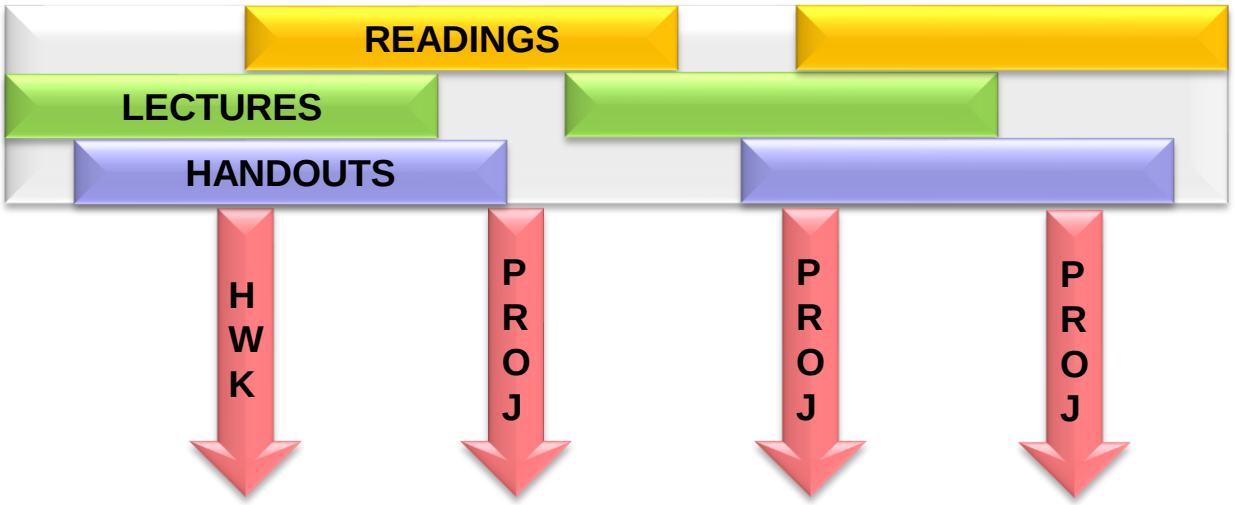
- No required textbook
- Over 800 slides posted as handouts on the course web page
  - You should understand all material in these handouts
- A few required papers
- Several posted tutorials with example code
  - You should fully understand these
- Several optional references

Main Course Material

- The main body of material is presented in the lectures, readings, and handouts
- Generally speaking, the hwk/projets *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 Exam generally focuses on the main body of material

Breadth and Depth

- Breadth and Depth



Course Overview

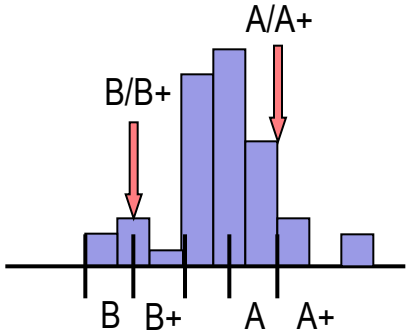
- Canvas
  - Grades posted here
    - Let me know if you ever see a score different than you expect
  - Upload electronic portions of hwk/projects here
- Syllabus
  - Posted on course web page

Lectures

- Ask questions at any time
- Please hold conversations outside of class
- Please silence phones

Letter Grade Assignments

- I assign a letter grade only for the final course grade
- I look at the final exams and course record of the class and assign two key dividing points: the A/A+ and (probably B/B+) 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



(not actual grade data)

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

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
  - One student did last year and got in big BIG trouble!!!

Cheating Websites
chegg, coursehero, etc.

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

MOSS

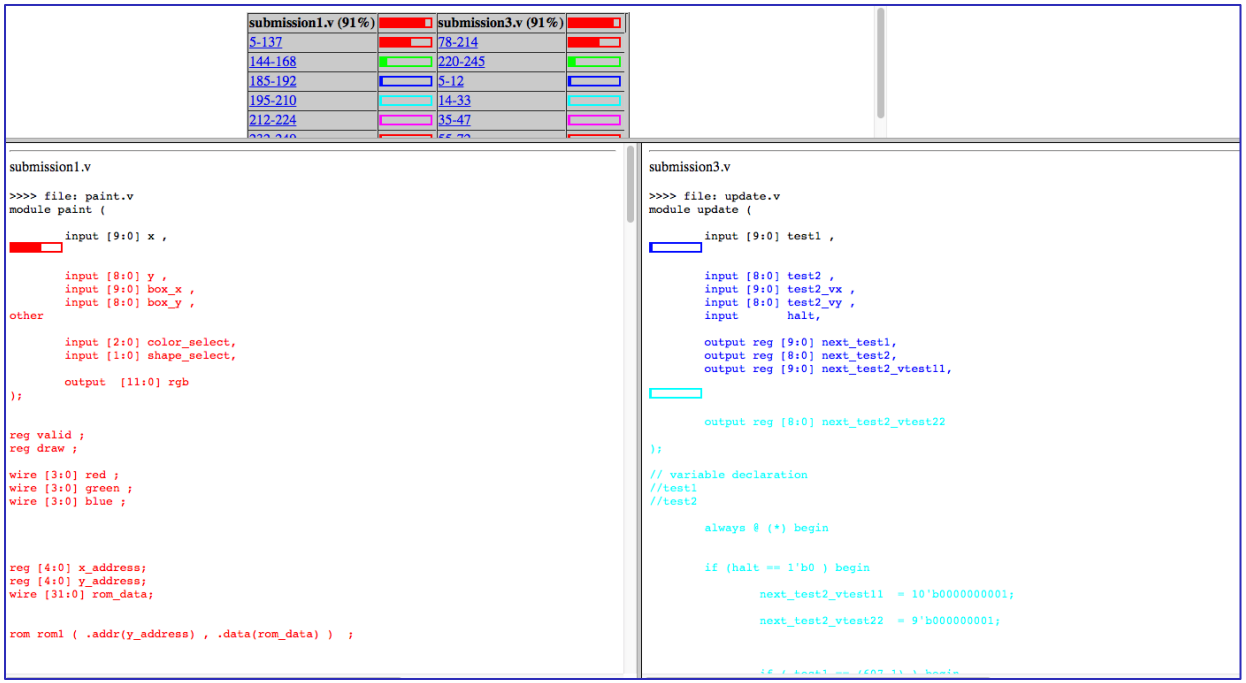
- "Measure Of Software Similarity" tool
- Utilizes very sophisticated and fast algorithms
- Processes all *Order*(N^2/2) = $N(N-1)/2$ pairings
  - Ex: examination of 300 submissions includes 44,850 pairwise comparisons
- If needed, MOSS runs will be made with this year's work combined with work from past years

MOSS Demonstration Case

- 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

Modified version

Original version

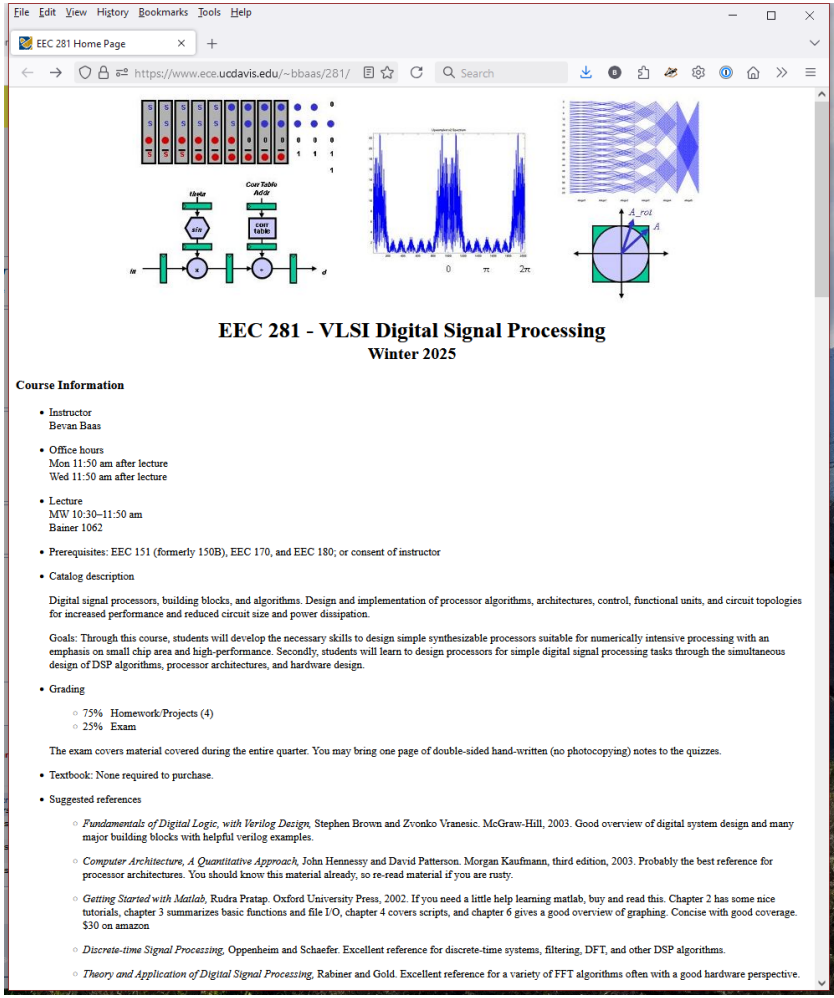


MOSS

- 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

Course Overview

- EEC 281 web page contents
  - Reading materials and references
  - Hwk/Project descriptions
  - Handouts



File Edit View History Bookmarks Tools Help

EEC 281 Home Page

https://www.ece.ucdavis.edu/~bbaas/281/

Search

EEC 281 - VLSI Digital Signal Processing
Winter 2025

Course Information

- Instructor
  Bevan Baas
- Office hours
  Mon 11:50 am after lecture
  Wed 11:50 am after lecture
- Lecture
  MW 10:30–11:50 am
  Bainer 1062
- Prerequisites: EEC 151 (formerly 150B), EEC 170, and EEC 180; or consent of instructor
- Catalog description

  Digital signal processors, building blocks, and algorithms. Design and implementation of processor algorithms, architectures, control, functional units, and circuit topologies for increased performance and reduced circuit size and power dissipation.

  Goals: Through this course, students will develop the necessary skills to design simple synthesizable processors suitable for numerically intensive processing with an emphasis on small chip area and high-performance. Secondly, students will learn to design processors for simple digital signal processing tasks through the simultaneous design of DSP algorithms, processor architectures, and hardware design.
- Grading
  - 75% Homework/Projects (4)
  - 25% Exam

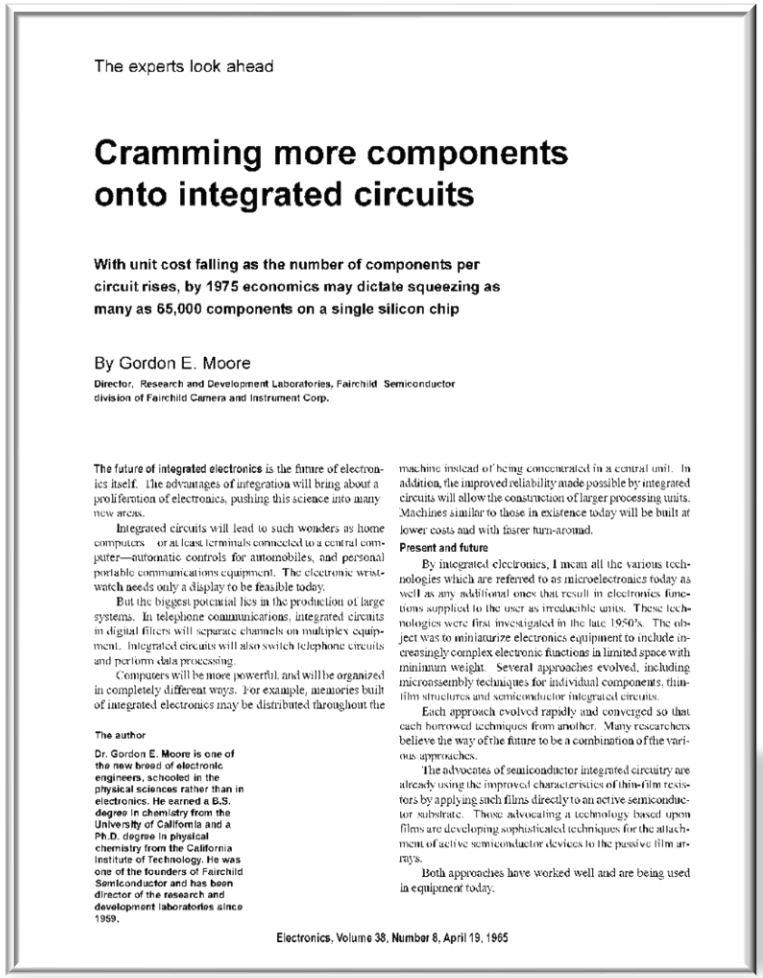
  The exam covers material covered during the entire quarter. You may bring one page of double-sided hand-written (no photocopying) notes to the quizzes.
- Textbook: None required to purchase.
- Suggested references
  - *Fundamentals of Digital Logic, with Verilog Design*, Stephen Brown and Zvonko Vranesic. McGraw-Hill, 2003. Good overview of digital system design and many major building blocks with helpful verilog examples.
  - *Computer Architecture, A Quantitative Approach*, John Hennessy and David Patterson. Morgan Kaufmann, third edition, 2003. Probably the best reference for processor architectures. You should know this material already, so re-read material if you are rusty.
  - *Getting Started with Matlab*, Rudra Pratap. Oxford University Press, 2002. If you need a little help learning matlab, buy and read this. Chapter 2 has some nice tutorials, chapter 3 summarizes basic functions and file I/O, chapter 4 covers scripts, and chapter 6 gives a good overview of graphing. Concise with good coverage. $30 on amazon
  - *Discrete-time Signal Processing*, Oppenheim and Schaefer. Excellent reference for discrete-time systems, filtering, DFT, and other DSP algorithms.
  - *Theory and Application of Digital Signal Processing*, Rabiner and Gold. Excellent reference for a variety of FFT algorithms often with a good hardware perspective.

Hwk/Proj Deadline Extensions

- Extending submission deadlines is problematic
  - Quarters are short and we miss two lectures due to holidays
  - When I have granted extensions, a student will frequently tell me, "I wish the extension had been announced when I started work on the project—I sacrificed one of my other classes and this is not fair to me". I am sympathetic to that reasoning.
- So an extension will be given in only a very unusual circumstance

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 irreducible 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 microassembly 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.

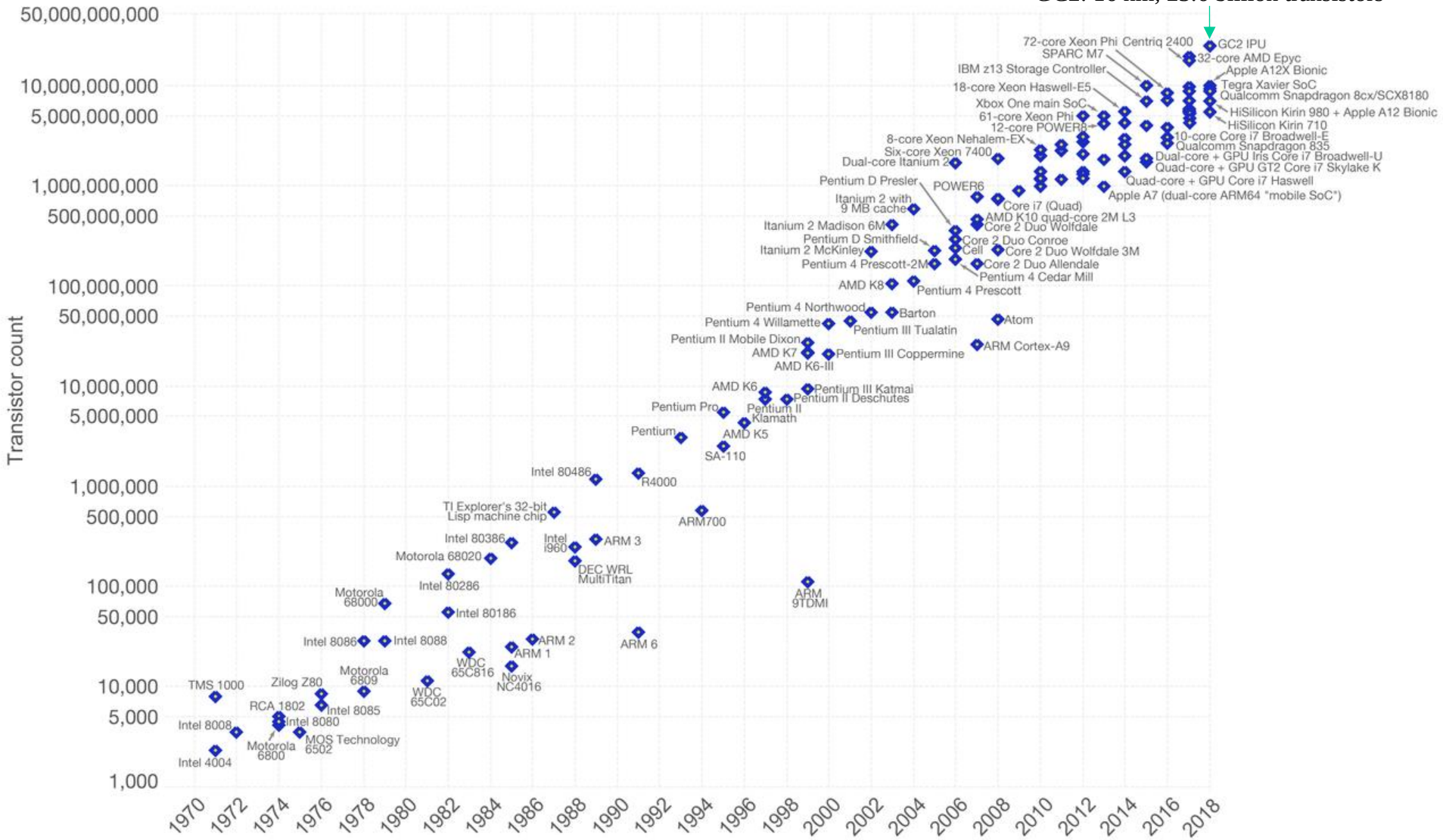
Electronics, Volume 38, Number 8, April 19, 1965

Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.

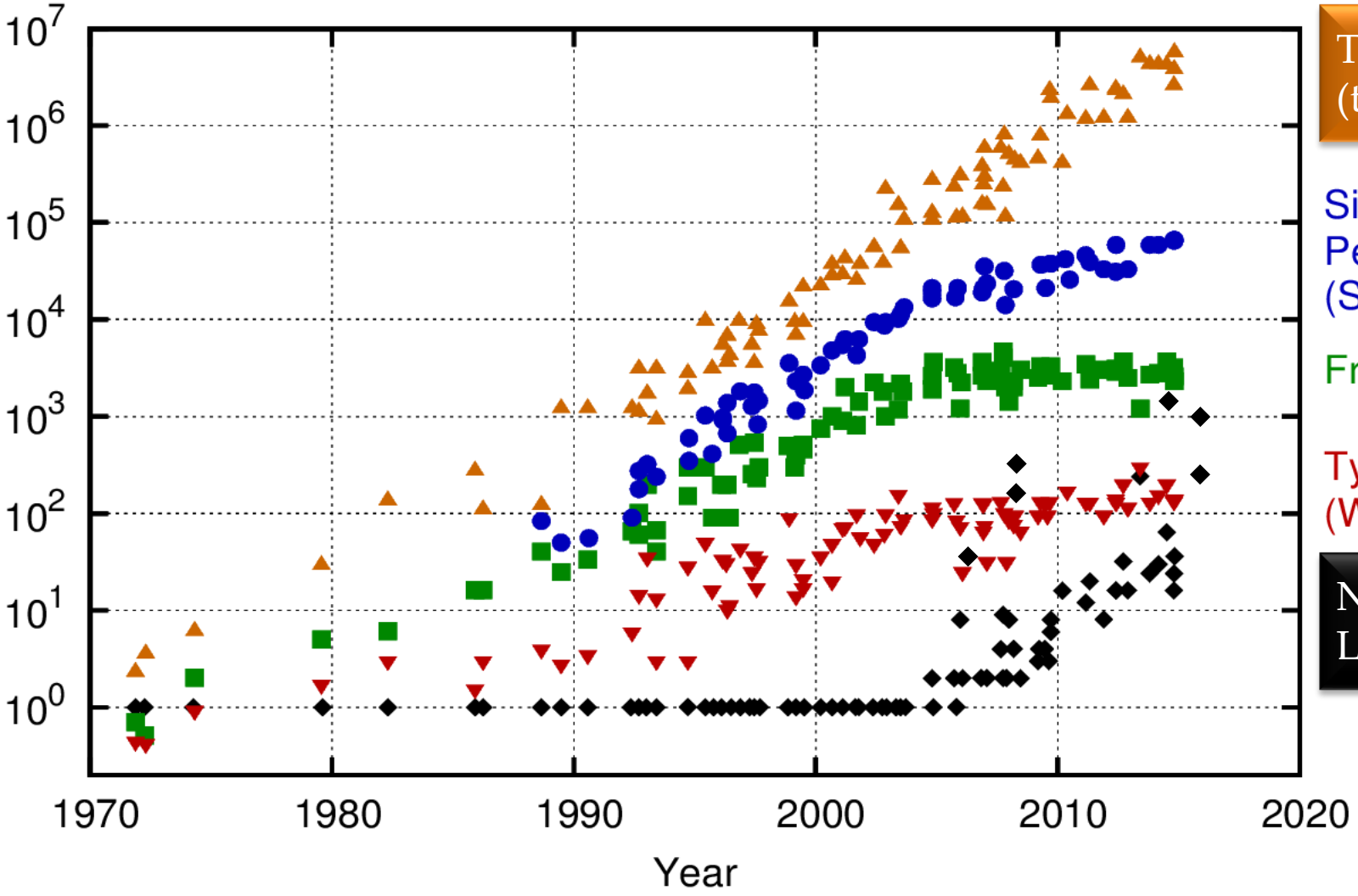
Our World in Data

GC2: 16 nm, 23.6 billion transistors



Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)

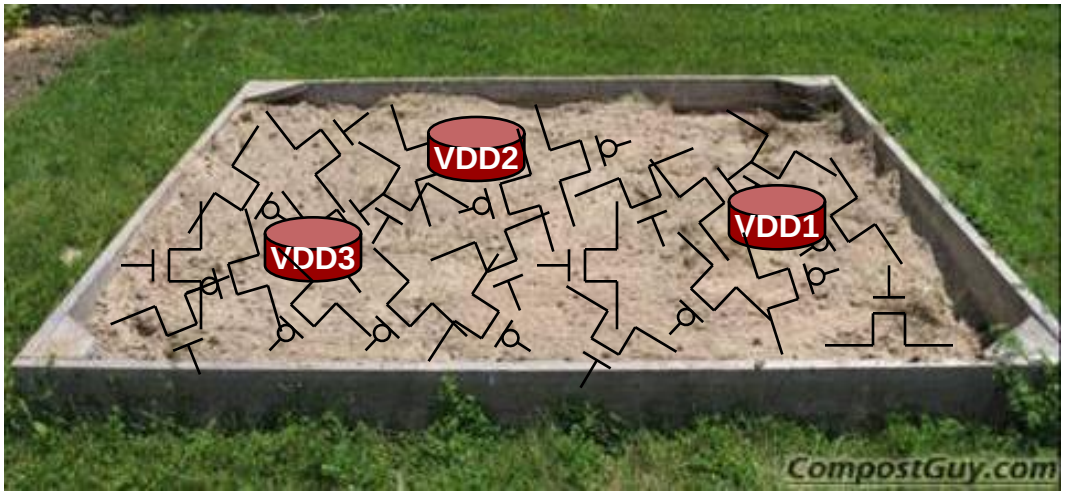
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.



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

Future Fabrication Technologies

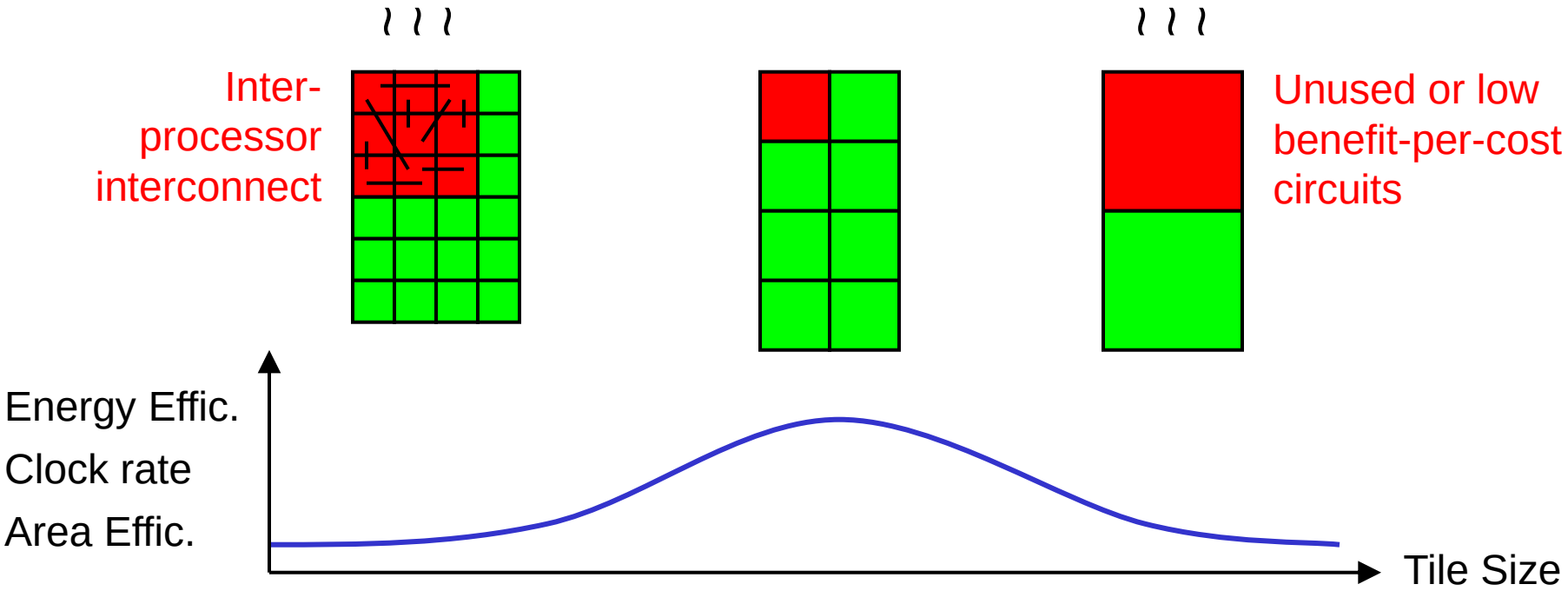
- Basic trends
  - Number of available devices: continually increasing
  - Energy dissipation per operation: decreasing too slowly



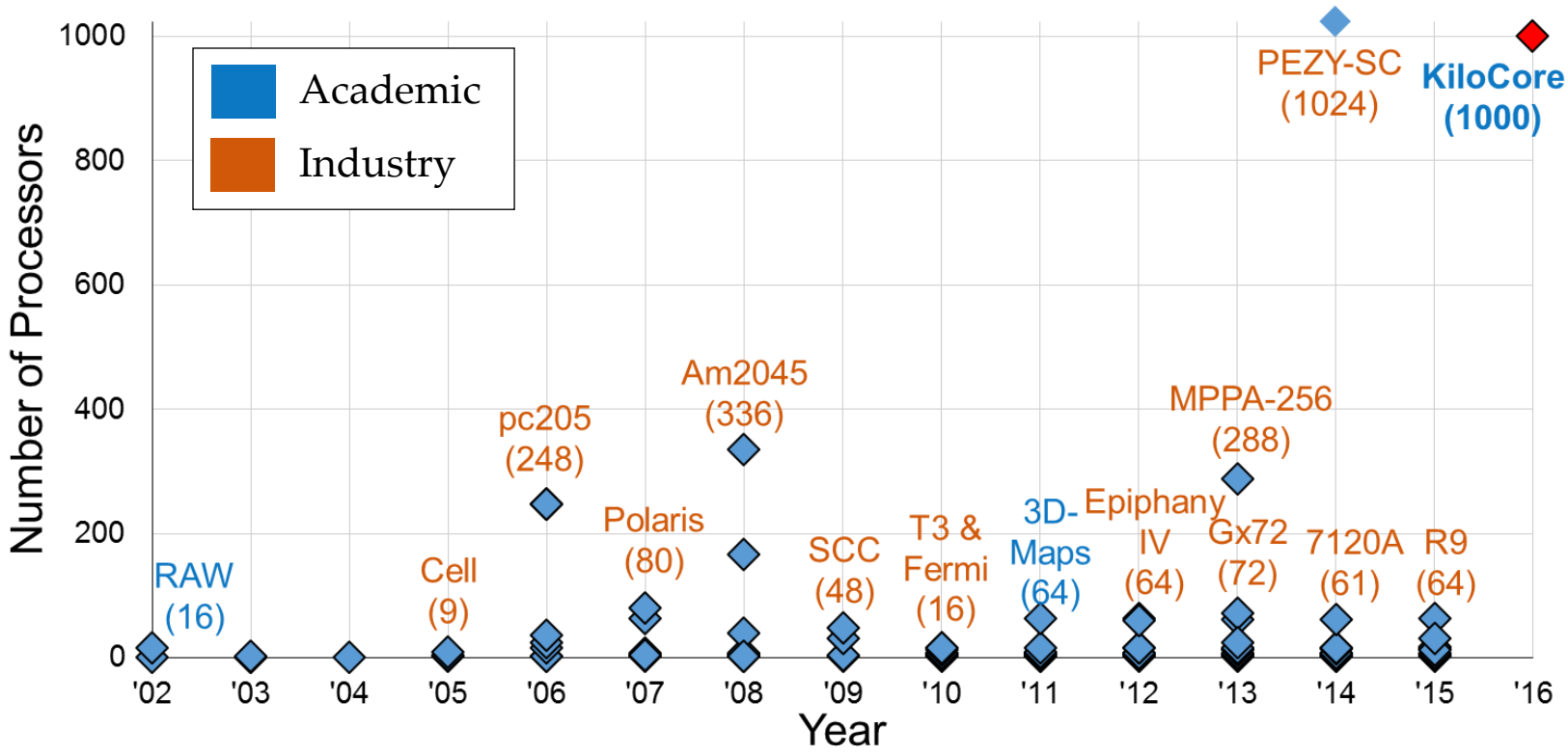
- There are a lot of ways to place and connect a billion transistors
- The most efficient implementations (throughput, energy, area) will have:
  - Processor sizes that capture computational kernels with few excess circuits
  - Optimized clock frequencies and supply voltages matched to dynamic workloads

Optimal Computational Tile Size

- The most efficient implementations (energy, throughput, chip area) have: Processor sizes that capture computational kernels with few excess circuits



Number of Processors on a Single Die vs. Year



Note: Each processor capable of independent program execution