

Lecture 1:

Introduction to Digital Logic Design

CSE 140: Components and Design Techniques for Digital
Systems

Spring 2022

CK Cheng

Dept. of Computer Science and Engineering
University of California, San Diego

Outlines

- Staff
 - Instructor, TAs, Tutors
- Logistics
 - Websites, Textbooks, Grading Policy
- Motivation
 - Moore's Law, Deep Learning, Quantum Computing
- Scope
 - Position among CSE courses
 - Coverage

Staff: the Instructor

- Instructor: CK Cheng
- Education: Ph.D. in EECS UC Berkeley
- Industrial Experiences: Engineer of AMD, Mentor Graphics, Bellcore; Consultant for technology companies
- Email: ckcheng+140@ucsd.edu
- Office: Room 2130 CSE Building
- Office hours
 - TBA
- Websites
 - <http://cseweb.ucsd.edu/~kuan>
 - <http://cseweb.ucsd.edu/classes/sp22/cse140-a>

Staff: TAs and Tutors

TAs

- Abraham, Elizabeth eabraham@ucsd.edu
- Holtz, Chester chholtz@ucsd.edu, Lead TA
- Jang, Wonsuk w1jang@ucsd.edu
- Li, Yuxin yul801@ucsd.edu
- Liu, Isabella lal005@ucsd.edu
- Ma, Jason jam043@ucsd.edu
- Wang, Kanlin kaw008@ucsd.edu

Tutors

- Dong, Jiayan jid001@ucsd.edu
- Huang, Yucheng yuh032@ucsd.edu
- Sun, William wis003@ucsd.edu

Office hours will be posted on the course website (piazza)

Logistics: Class Schedule

- Lecture: TR 3:30-4:50PM
- Discussion: F 3-350PM
- 4-5 Homework Assignments, 3 Exams: W4, W7, W10
- The Third Exam: Due time M 6:00PM, 6/6/2022

Logistics: Textbooks

Reserved at library

- Digital Design and Computer Architecture by David Money Harris and Sarah L. Harris, Morgan Kaufmann (Reading Activities)
- Digital Systems and Hardware/Firmware Algorithms, Milos D. Ercegovac and Tomas Lang (System Designs, pdf file will be released)

Logistics: Sites for the Class

- Class website
 - <http://cseweb.ucsd.edu/classes/sp22/cse140-a/index.html>
 - Syllabus
 - Class notes
 - Assignment: Homework and Reading Activities
 - Exercises: Solutions and Rubrics
- Forum: Piazza (Office Hrs., Online Discussion)
- Homework and Exam submission: Gradescope
- Lecture/Discussion Record: Podcast

Logistic: Class Expectation

- Level 1: Concepts and Methods (Lecture and slides)
 - Class attendance of video record
- Level 2: Hands on Practices (Exercises)
 - Homework
- Level 3: Command of Materials (Samples of exams)
 - Exams: First part of exam (90%)
- Level 4: Extension and Exploration (Content beyond the textbook)
 - Exams: Second part of exam (10%)

Logistics: Grading

Total pts distribution:

- Homework: 35pts (grade based on a subset of problems)
- Exams: 32.5pts for the each of the best two out of 3 exams
 - Exam 1: Week 4
 - Exam 2: Week 7
 - Exam 3: (due 600PM, Monday 6/6/2022)
- Grading: The better choice of the following two categories
 - Exact score (100pts)
 - $A > 93$, $A^- > 90$, $B^+ > 87$, $B > 83$, $B^- > 80$, $C^+ > 77$, $C > 73$
 - $C^- > 50$.
 - The curve (100% of class)
 - A^+ top $x\%$ ($x < 5$)
 - (A, A^-) top $33 \pm \varepsilon\%$; (B^+, B, B^-) top $66 \pm 2\varepsilon\%$

Logistic: grading components

- Homework:
 - Due Wed night on Gradescope
 - Group discussion is encouraged. However, we are required to write individually
 - Discount 10% loss of credit for each day after the deadline but no credit **after the solution is posted.**
 - Some parts will be graded by completion only (**students are responsible for correct solutions**)
 - **Standard: Posted solutions and rubrics**

Logistic: 3 Exams

Exams: take-home exam taken within 48 hours

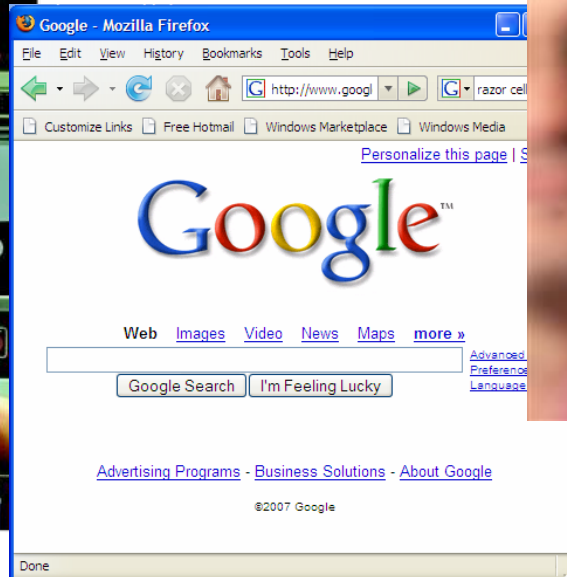
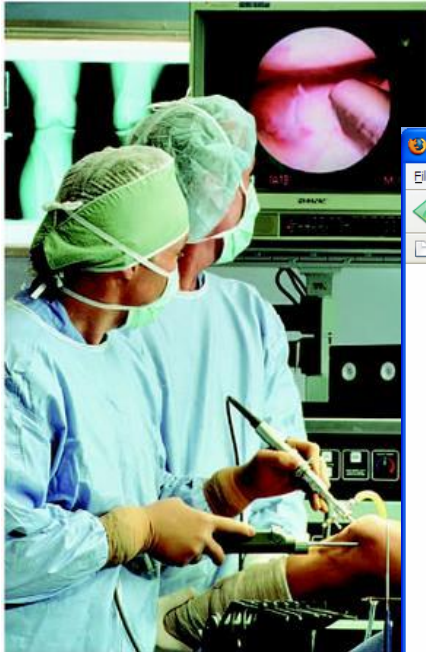
- Open book, notes, free to check websites
- Solution and grading policy will be posted after the exam.
- Exams are not cumulative but requires a good command of the previous exam content
- Two timeframe options for the class to decide
 - I. 80 minutes time frame: Finish the exam in 80 minutes after you start the exam.
 - II. 48 hours time frame: Deliver the result in 48 hours after the exam is posted.
- Exam schedule
 - I. Weekends
 - II. Weekdays
 - III. Sunday + Monday

Course Problems...Cheating

- What is cheating?
 - Studying together in groups is not cheating but encouraged
 - Turned-in work must be *completely* your own.
 - Copying someone else's solution on a HW or Exam is cheating
 - Both “giver” and “receiver” are equally culpable
- We will be better off to work on the problem alone during the exam.
- We have to address the issue once the cheating is reported by someone (e.g. TA, Tutor, Student, etc.).

Motivation

- Microelectronic technologies have revolutionized our world: cell phones, internet, rapid advances in medicine, etc.
- The semiconductor industry has grown from \$21 billions in 1985 to \$555.9 billions in 2021.



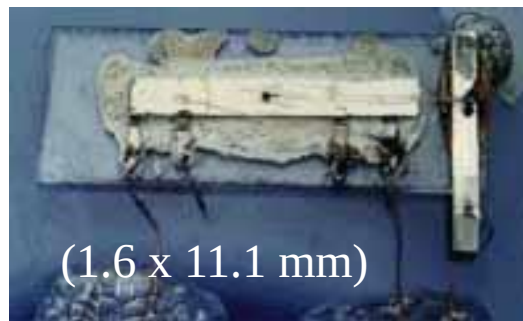
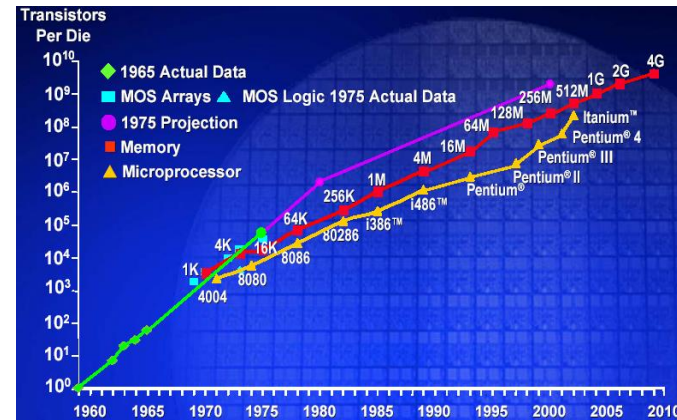
The Digital Revolution

Integrated Circuit: Many digital operations on the same material

Vacuum tubes



Exponential Growth
of Computation



(1.6 x 11.1 mm)

Integrated Circuit

Moore's Law

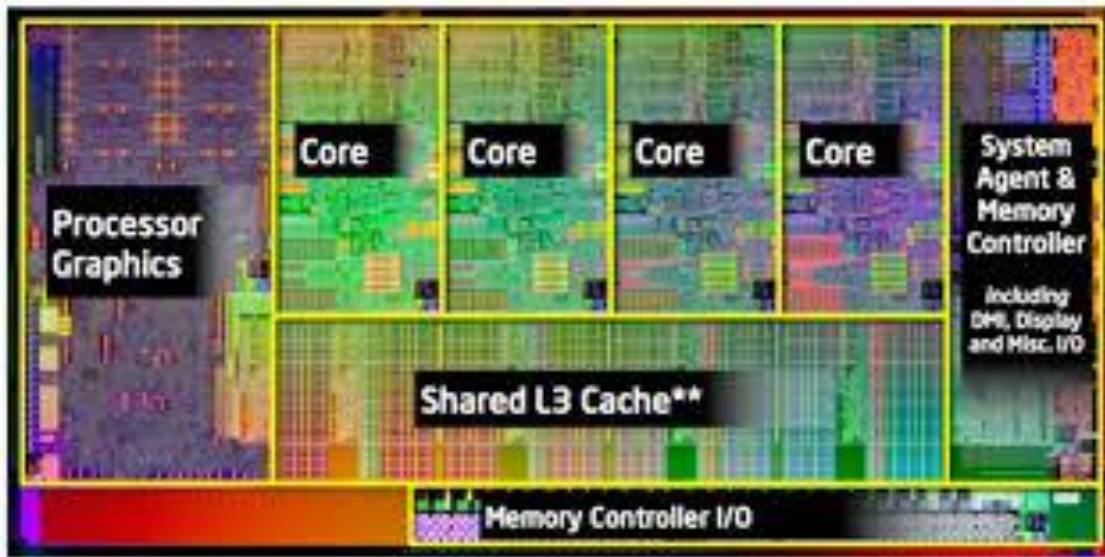
ENIAC

WWII
Stored Program
Model

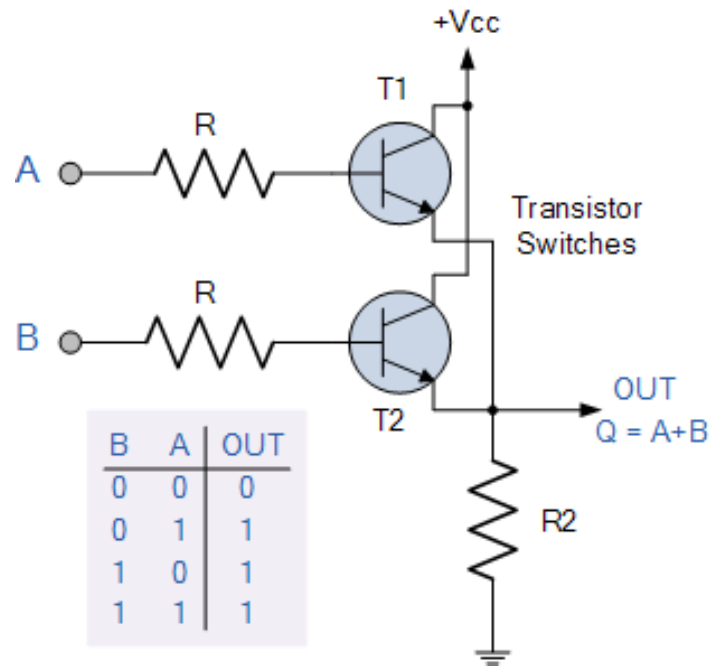
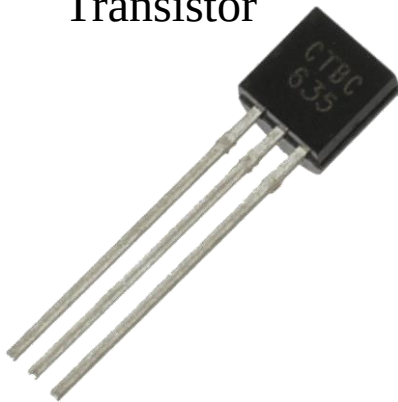
1949

1965

Building complex circuits



Transistor



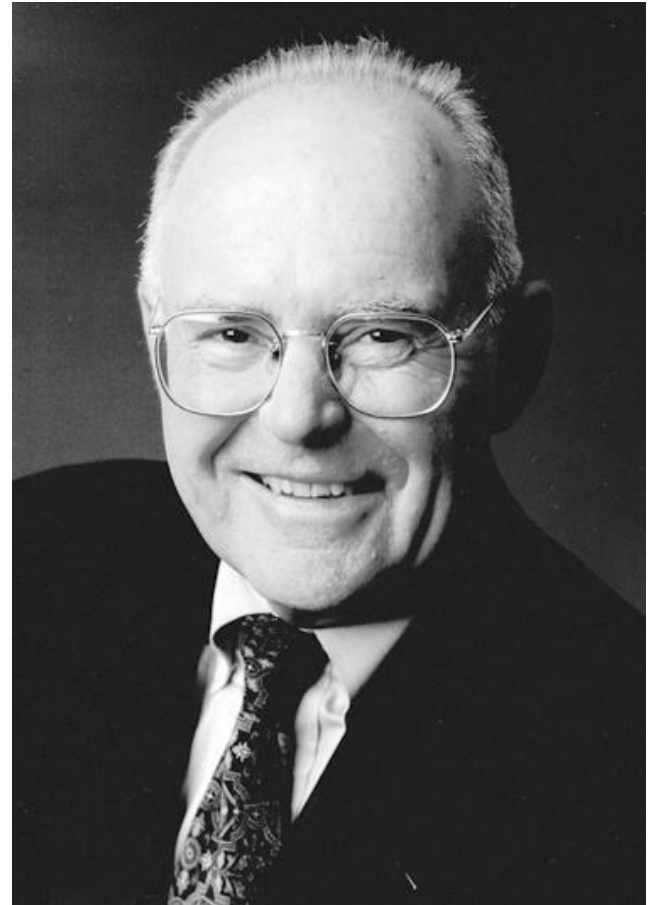
Robert Noyce, 1927 - 1990

- Nicknamed “Mayor of Silicon Valley”
- Co-founded Fairchild Semiconductor in 1957
- Co-founded Intel in 1968
- Co-invented the integrated circuit



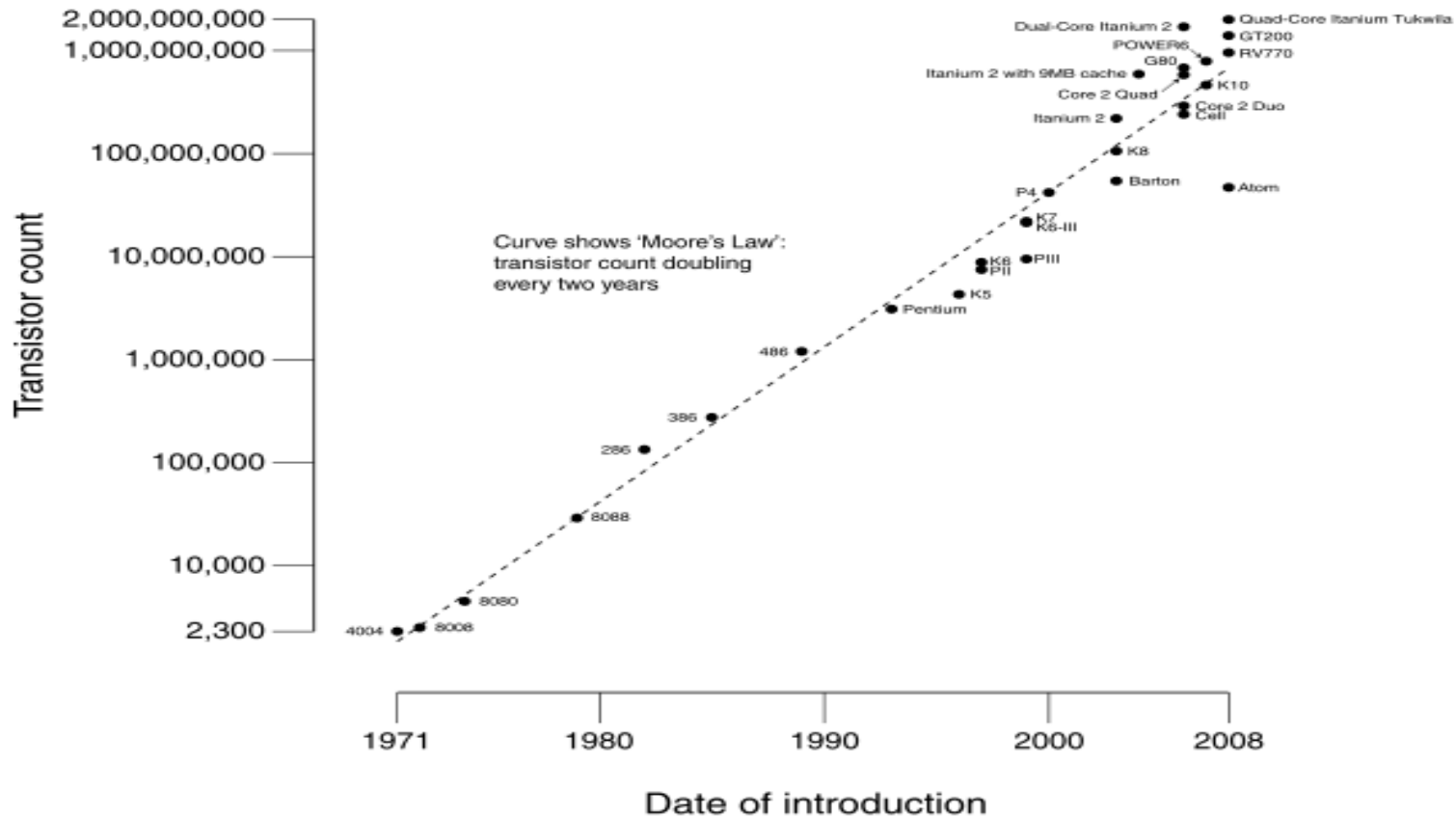
Gordon Moore

- Cofounded Intel in 1968 with Robert Noyce.
- **Moore's Law:** the number of transistors on a computer chip doubles every 1.5 years (observed in 1965)



Technology Trends: Moore's Law

CPU Transistor Counts 1971-2008 & Moore's Law



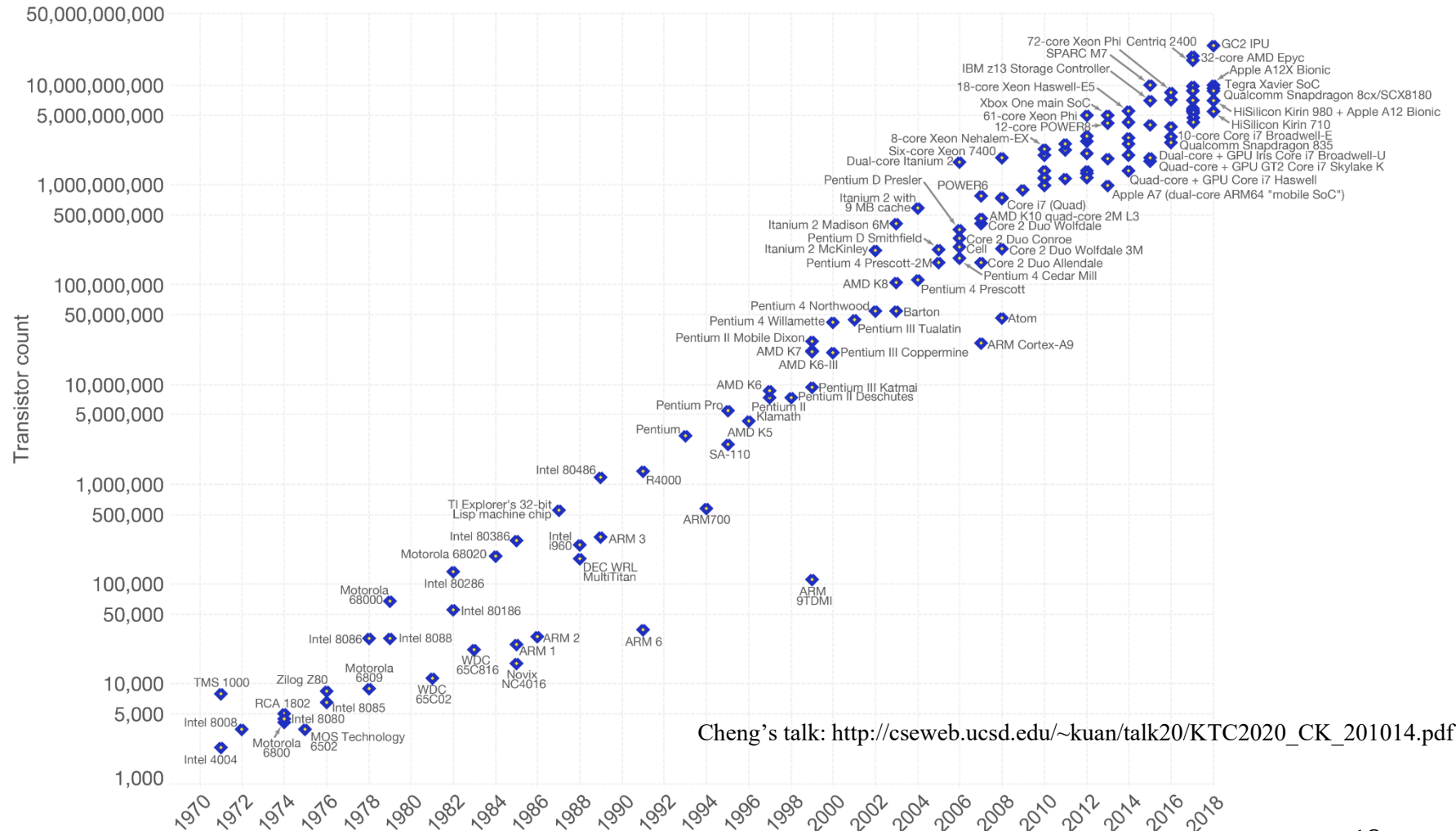
- Since 1975, transistor counts have doubled every two years.
- Moore's law: wider applications: larger market: higher revenue: more R&D

Introduction: Moore in 1965

The complexity for minimum component costs has increased at a rate of roughly a factor of two/yr.

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.



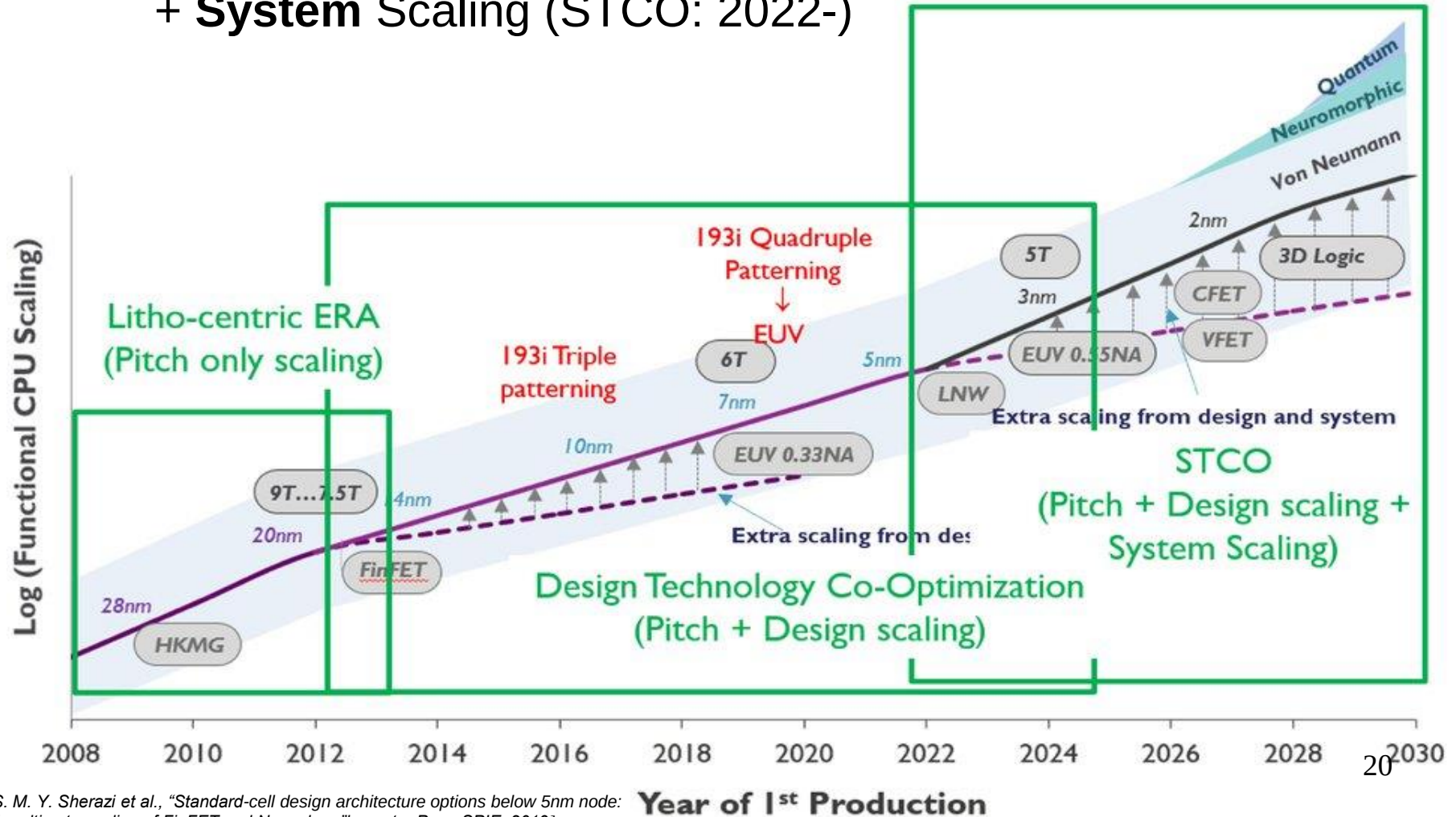
Moore's Law: Roadmap

Scaling Methods:

Pitch only (2008-2013)

+ **Design** Technology Co-Optimization (DTCO: 2012-2025)

+ **System** Scaling (STCO: 2022-)



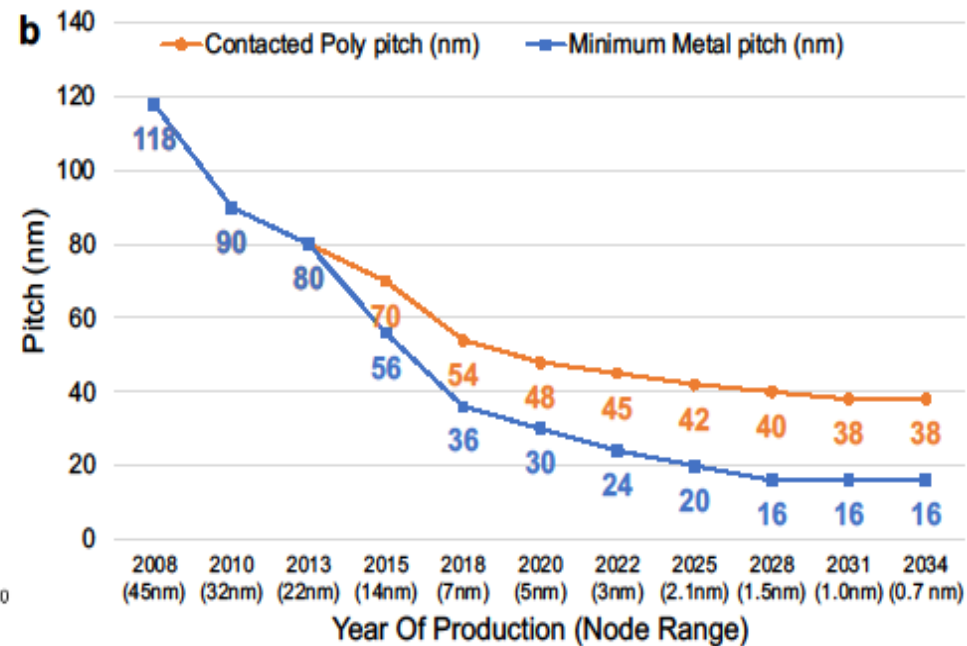
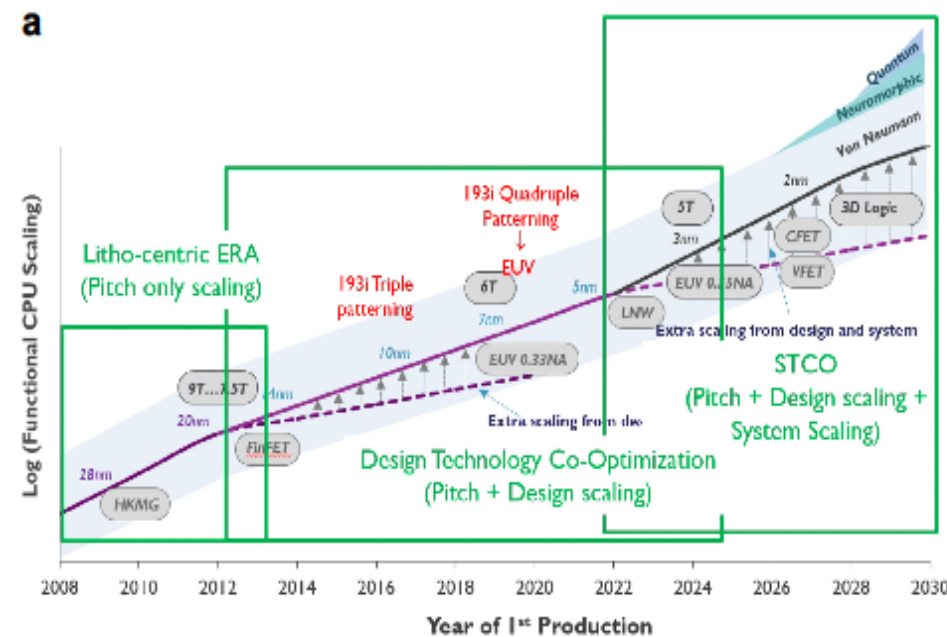
Moore's Law: Roadmap

Scaling Methods:

Pitch only (2008-2013)

+ **Design** Technology Co-Optimization (DTCO: 2012-2025)

+ **System** Scaling (STCO: 2022-)



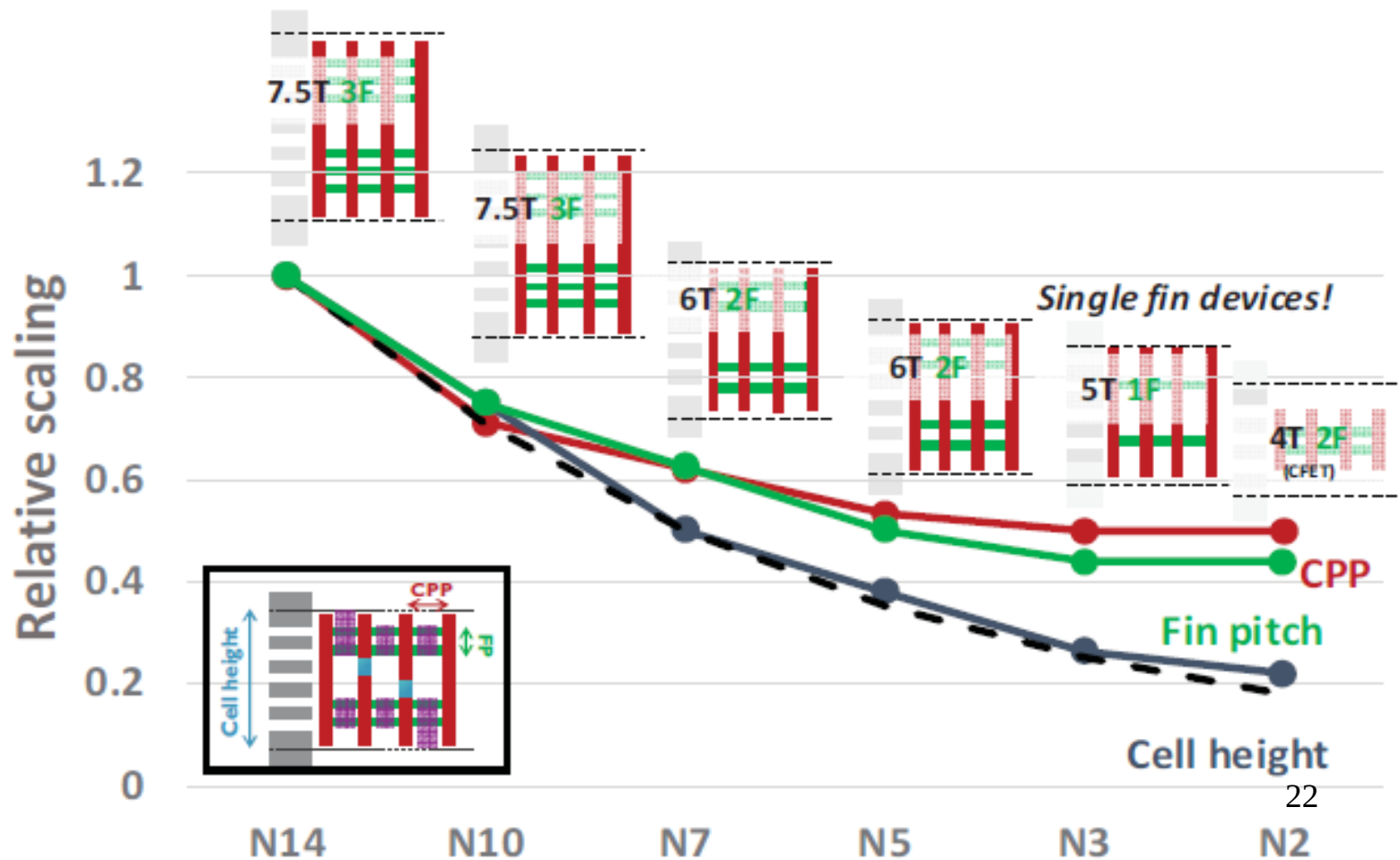
Moore's Law: Roadmap

Scaling Methods:

Pitch only (2008-2013)

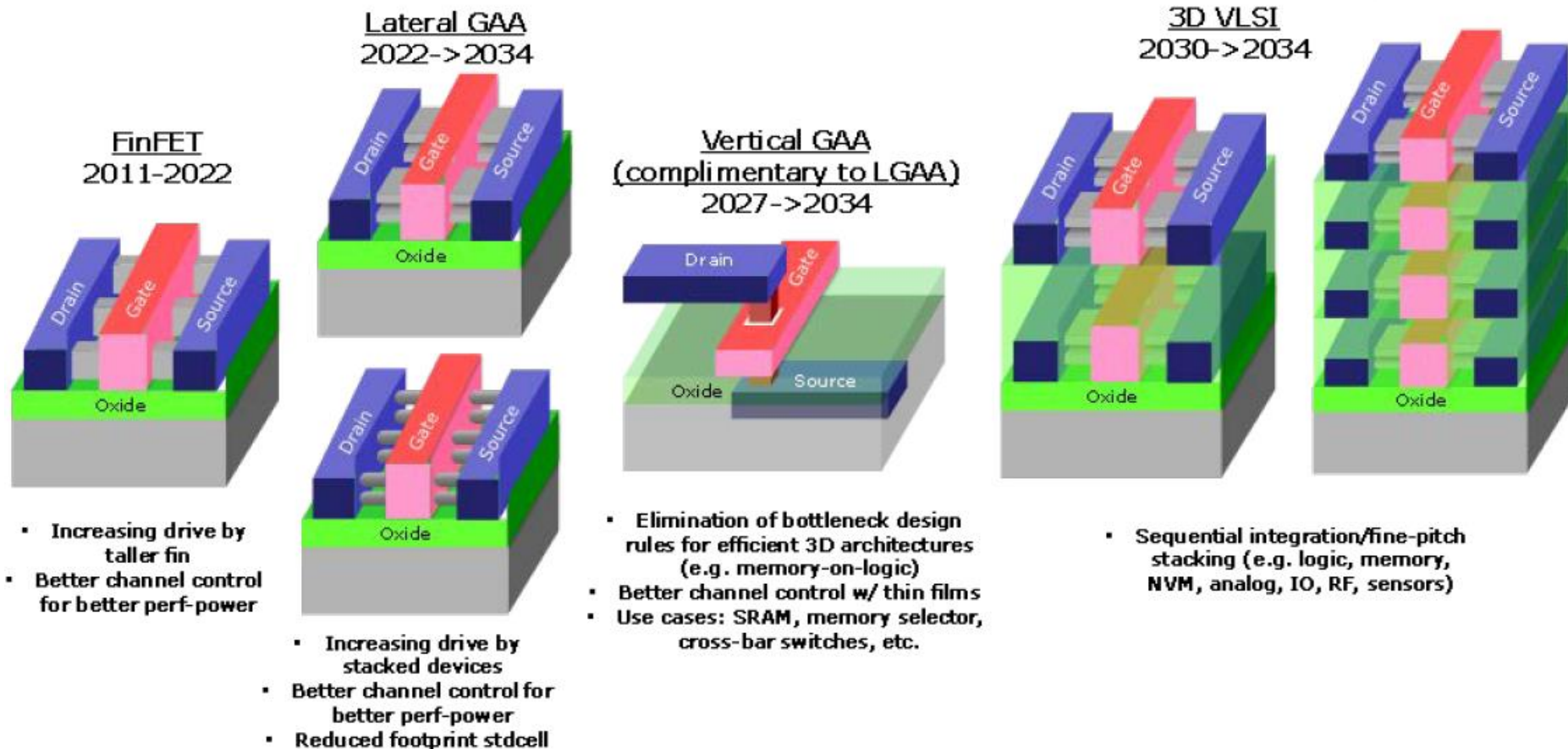
+ **Design** Technology Co-Optimization (DTCO: 2012-2025)

+ **System** Scaling (STCO: 2022-)



Moore's Law: System Technology Co-Optimization

2.5D/3D fine-pitch assembly + stacking



Change in the MOSFET device architecture from the 2D planar through 2.5D FinFets to 3D monolithic VLSI

New Technologies

- New materials and fabrication for devices
 - Low power devices
 - Three dimensional integrated circuits
 - Graphene
- New architecture
 - Machine learning, deep learning
- Quantum computing

Understand the principles to explore the future

Artificial Intelligence

- Logic and Reasoning
- Boolean Satisfiability (SAT)
 - Product of sum clauses
 - Diagnosis
- States and Sequences
 - Sequential Machines
 - Reachability
 - Controllability

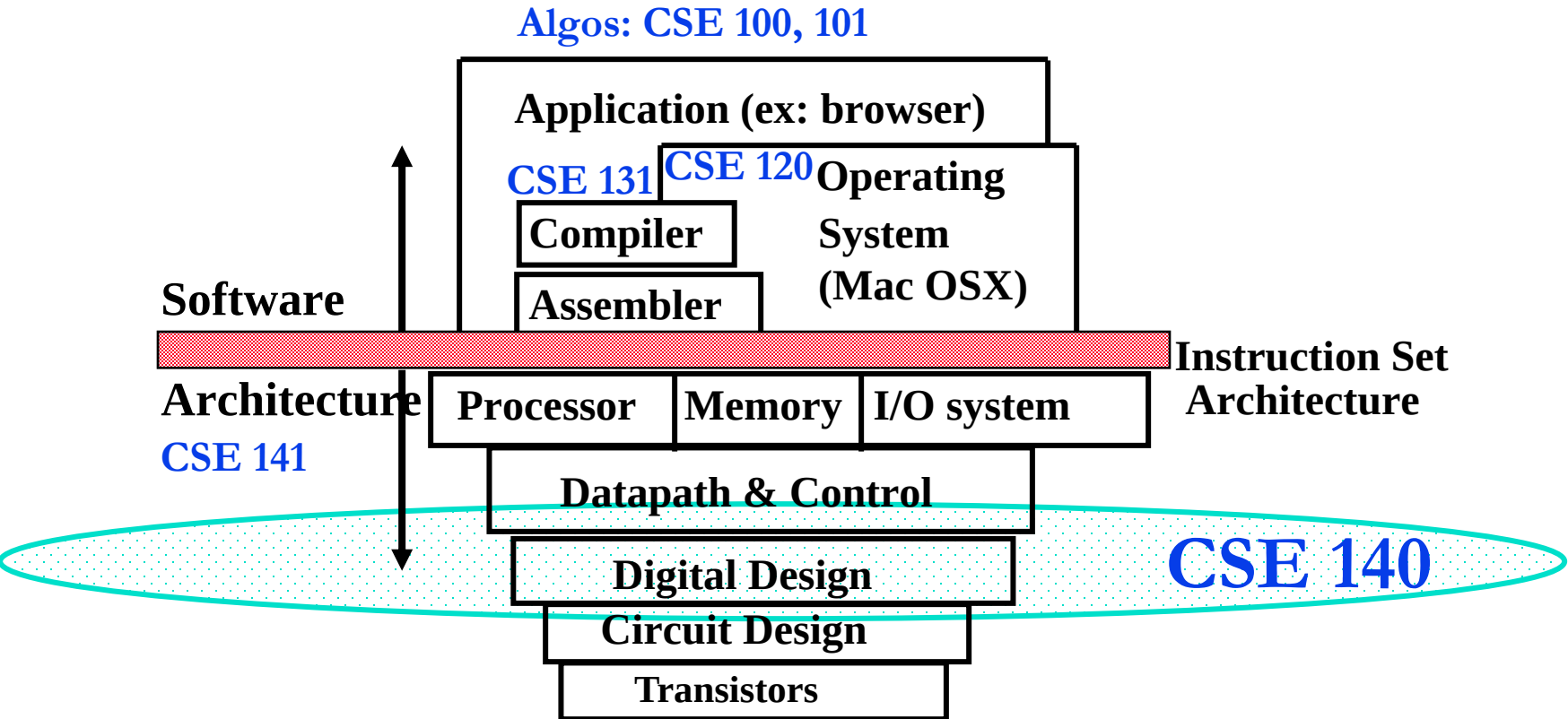
One example of the applications and opportunities

Scope

The purpose of this course is that we:

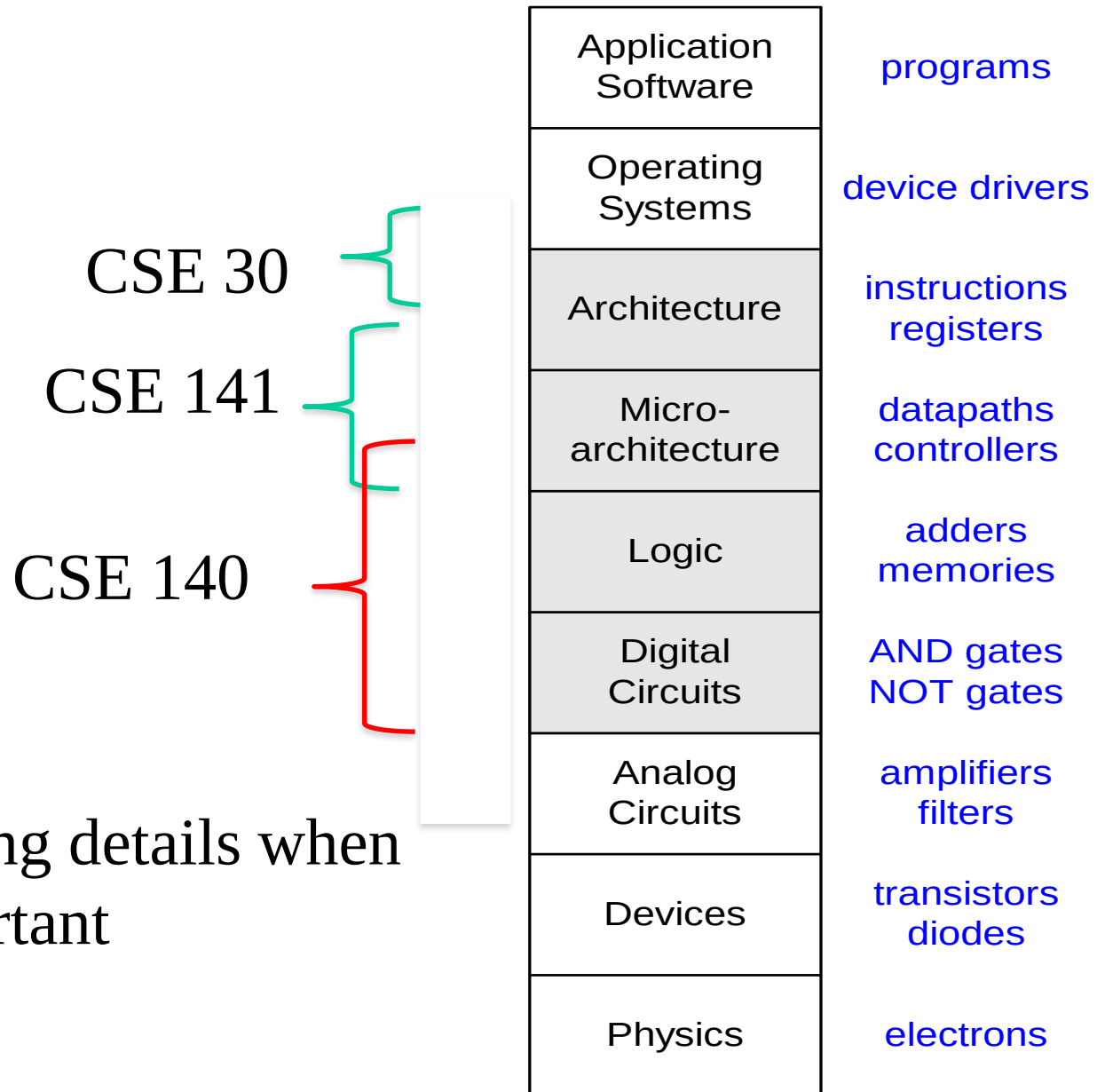
- Learn the principles of digital design
- Learn to systematically debug increasingly complex designs
- Design and build digital systems
- Learn what's under the hood of an electronic component
- Prepare for the future technology revolution

Position among CSE Courses



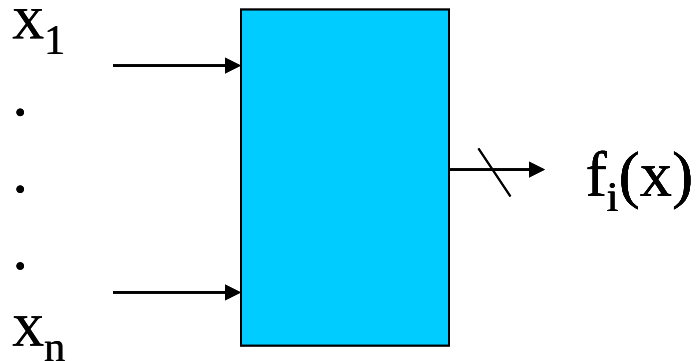
- Big idea: Coordination of many *levels of abstraction*

Principle of Abstraction



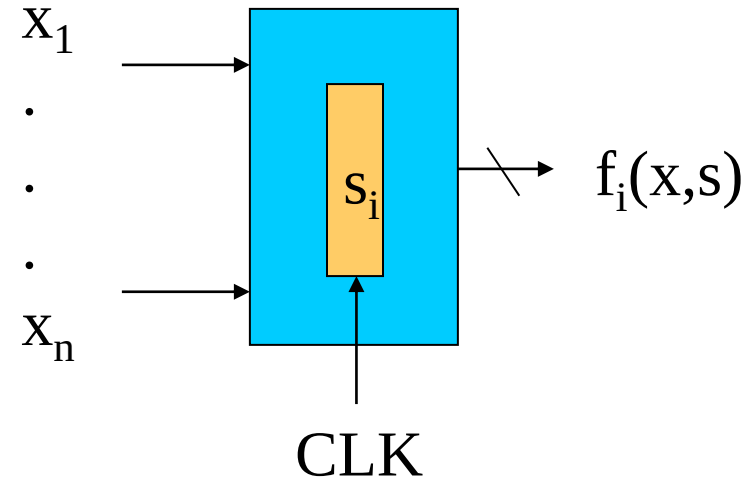
Abstraction: Hiding details when they are not important

Combinational Logic vs Sequential Network



Combinational logic:

$$y_i = f_i(x_1, \dots, x_n)$$



Sequential Networks

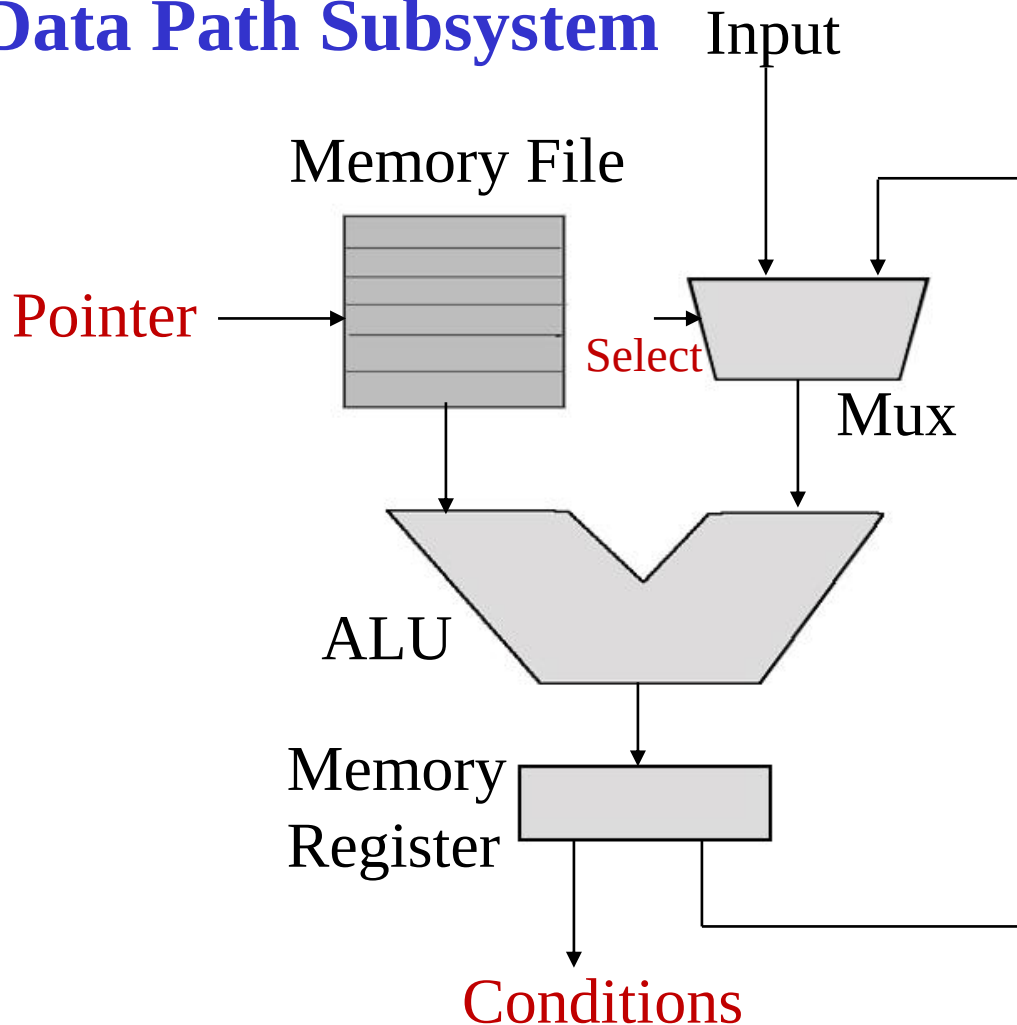
1. Memory
2. Time Steps (Clock)

$$y_i^t = f_i(x_1^t, \dots, x_n^t, s_1^t, \dots, s_m^t)$$

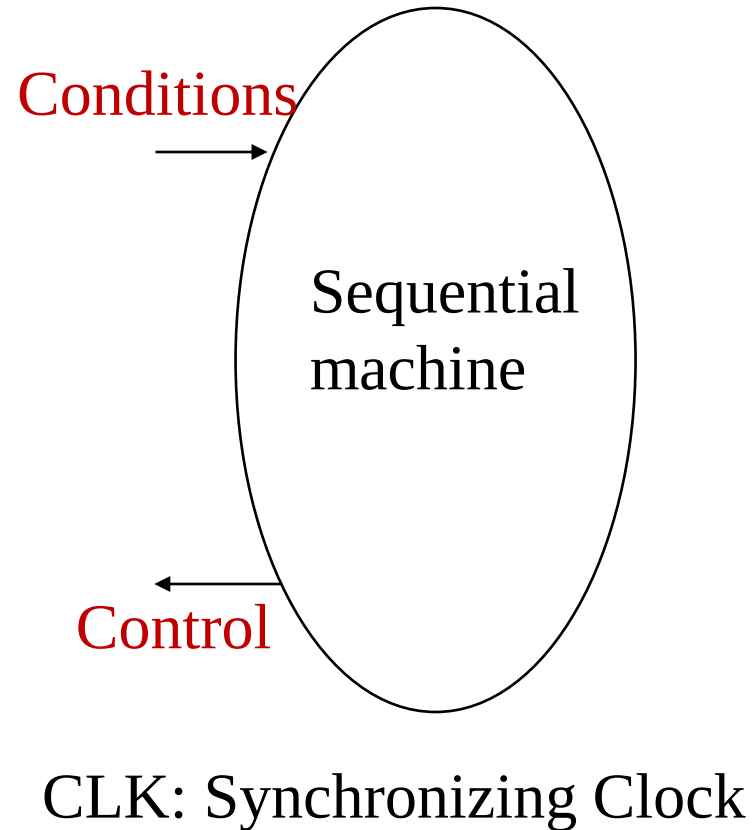
$$s_i^{t+1} = g_i(x_1^t, \dots, x_n^t, s_1^t, \dots, s_m^t)$$

Scope: Overall Picture of CS140

Data Path Subsystem



Control Subsystem



Scope

Subjects	Building Blocks	Theory
Combinational Logic	AND, OR, NOT, XOR	Boolean Algebra
Sequential Network	AND, OR, NOT, FF	Finite State Machine
Standard Modules	Operators, Interconnects, Memory	Arithmetics, Universal Logic
System Design	Data Paths, Control Paths	Methodologies

Combinational Logic Basics

What is a combinational circuit?

- No memory
- Realizes one or more functions
- Inputs and outputs can only have two discrete values
 - Physical domain (usually, voltages) (Ground 0V, Vdd 1V)
 - Mathematical domain : Boolean variables (True, False)

Differentiate between different representations:

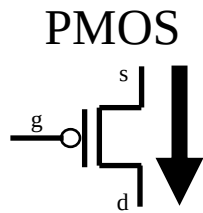
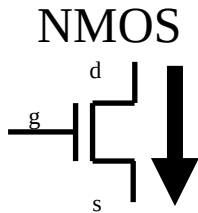
- physical circuit
- schematic diagram
- mathematical expressions

Binary Digital Logic

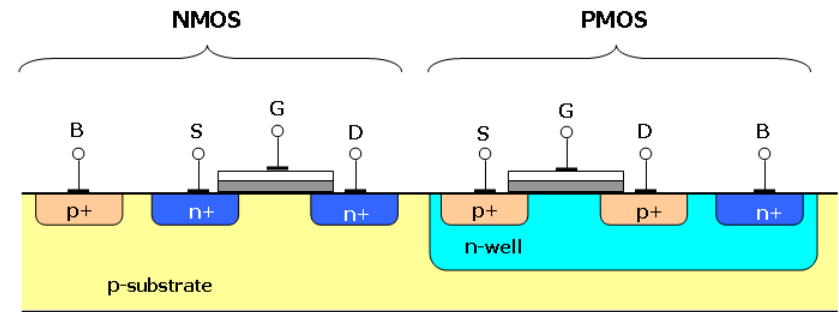
- Simplest representation is “1” and “0” (base-2).
- Choose a physical quantity to represent “1” and “0”
 - Usually voltage, but not always (e.g. current, resistance, magnetic polarization, quantum spin, ...)
- Use a transistor to make the switch (operating as a digital instead of analog device)
 - Two states – on / off
 - Signals can be high voltage (“1”) or low voltage (“0”)

Basic CMOS

”on”
when
gate is
high



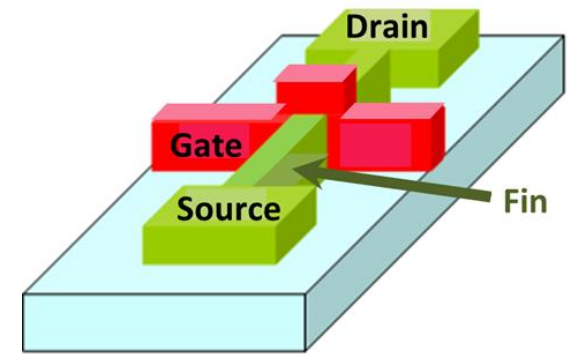
”on”
when
gate is
low



https://en.wikipedia.org/wiki/CMOS#/media/File:Cmos_impurity_profile.PNG

- Complementary Metal Oxide Semiconductor
- Invented in the 1960's, but took over in the 80's
- “on” means low resistance, ”off” means high resistance
- Logic “1” and Logic “0” values are arbitrary
e.g. logic “1” == 1.0 V, logic “0” == 0.0 V

Planar technology



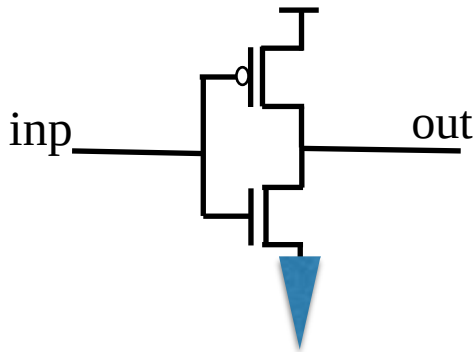
FINFET technology

By Irene Ringworm, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=3833512>

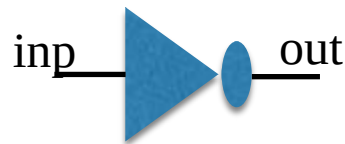
Transistors as Switches

The most basic CMOS gate - inverter

- When “inp” is “1”, then the nmos is on and the pmos is off – output will be ?
- When “inp” is “0”, then the nmos is off and the pmos is on – output will be ?



Schematic



Truth Table

in	out
0	1
1	0

Equation

$$\text{out} = \text{in}'$$

$$\text{out} = \overline{\text{in}}$$

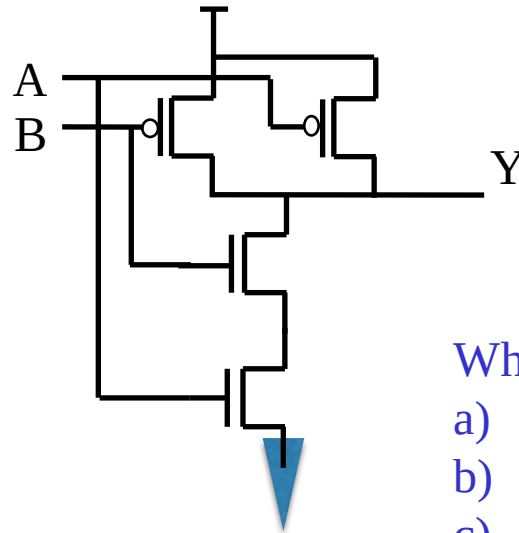
Basic Gates – (N)AND gate

AND $Y = A \& B$ $Y = AB$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

NAND $Y = (A \& B)'$ $Y = (AB)'$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



What kind of gate is this?

- a) AND
- b) NAND
- c) Inverter
- d) None of the above

Basic Gates – (N)OR gate

OR $Y = A + B$ $Y = A \mid B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

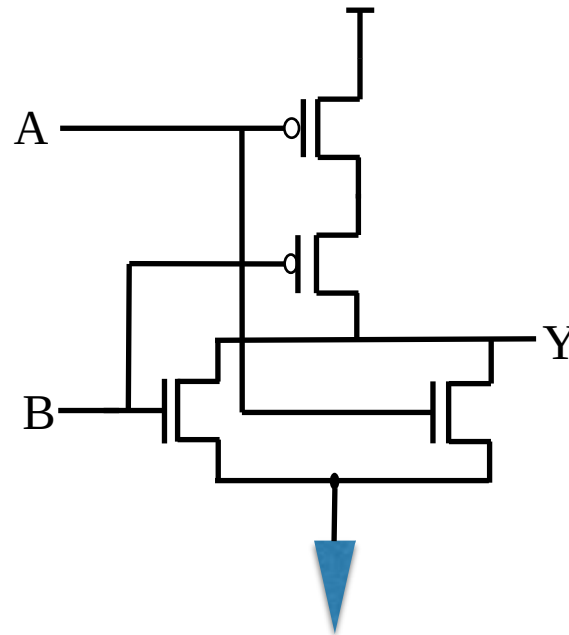


NOR $Y = (A + B)'$ $Y = (A \mid B)'$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



Bubble
means
Invert



NOR is universal gate

Transistors as Switches: CMOS

Transistors as Switches

Transistors as Switches: Transmission Gate

Transistors as Switches: Transmission Gate

Boolean Algebra

A branch of algebra in which the values of the variables belong to a set B (e.g. $\{0, 1\}$), has two operations $\{+, \cdot\}$ that satisfy the following four sets of laws.

- Associative laws: $(a+b)+c = a+(b+c)$, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
- Commutative laws: $a+b = b+a$, $a \cdot b = b \cdot a$
- Distributive laws: $a+(b \cdot c) = (a+b) \cdot (a+c)$,
 $a \cdot (b+c) = a \cdot b + a \cdot c$
- Identity laws: $a+0 = a$, $a \cdot 1 = a$
- Complement laws: $a+a' = 1$, $a \cdot a' = 0$
(x' : the complement element of x)

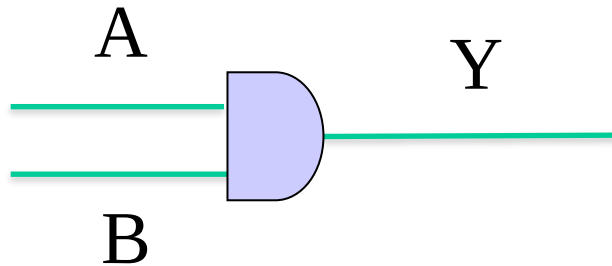
Duality

- Swap $(+, \cdot)$ and complement all 0's and 1's
- If we can prove a statement using laws of Boolean algebra true, then the duality of the statement is also true.

Laws and their duals

Associative	$(a \cdot b) \cdot c = a \cdot (b \cdot c)$	$(a + b) + c = a + (b + c)$
Commutative	$a \cdot b = b \cdot a$	$a + b = b + a$
Distributive*	$a \cdot (b + c) = a \cdot b + a \cdot c$	$a + (b \cdot c) = (a + b) \cdot (a + c)$
Identity	$a \cdot 1 = a$	$a + 0 = a$
Compliment	$a \cdot a' = 0$	$a + a' = 1$

Representations of combinational circuits: The Schematic



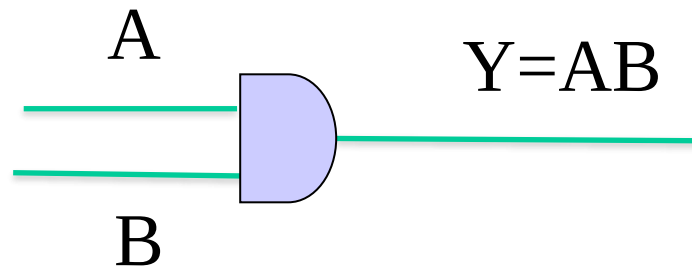
- What is the simplest combinational circuit that you know?

Representations of combinational circuits

Truth Table: Enumeration of all combinations

Example: AND

id	A	B	Y
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1



Boolean Algebra

Similar to regular algebra but defined on sets with only three basic ‘logic’ operations:

- | | |
|---------------------------------|--------------------------|
| 1. Intersection: AND (2-input); | Operator: $\cdot$, $\&$ |
| 2. Union: OR (2-input); | Operator: $+$, $ $ |
| 3. Complement: NOT (1-input); | Operator: $'$, $!$ |

“ $\&$, $|$, $!$ ” Symbols in BSV

Some Def's

- Complement : variable with a “BAR” over it or ‘ after it
 A'

- Literal : variable or its complement

- Implicant: product of literals

ABC

- Implicate: sum of literals

$(A+B+C)$

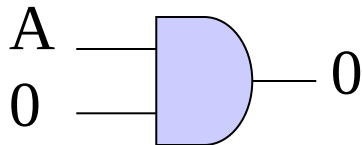
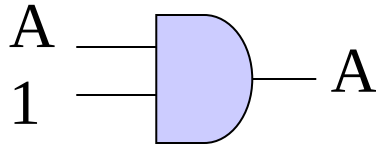
Minterm, maxterm (implicant or implicate that includes all the inputs)

$f(A,B,C,D): ABCD, (A+B+C+D)$

Boolean algebra and switching functions

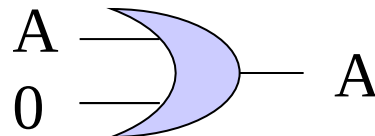
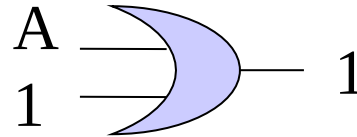
Two-input AND ($\cdot$)

AND	A B	Y
	0 0	0
	0 1	0
	1 0	0
	1 1	1



Two-input OR ($+$)

OR	A B	Y
	0 0	0
	0 1	1
	1 0	1
	1 1	1



One-input NOT
(Complement, ')

NOT	A	Y
	0	1
	1	0

For an AND gate,
0 at input blocks the other inputs
and dominates the output
1 at input passes signal A

For an OR gate,
1 at input blocks the other inputs
and dominates the output
0 at input passes signal A

Boolean Algebra

For two Boolean variables X and Y with $X=1$, $Y=0$, what is function $f(X,Y)=X+Y$?

A. $f(X,Y)=0$

B. $f(X,Y)=1$

C. $f(X,Y)=2$

Boolean Algebra

iClicker Q: For two Boolean variables X and Y with $X=1$, $Y=0$, what is function $f(X,Y)=X+X+Y$?

A. $f(X,Y)=0$

B. $f(X,Y)=1$

C. $f(X,Y)=2$

Boolean Algebra

For two Boolean variables X and Y with $X=1$, $Y=0$, what is function $f(X,Y)=X+XY$?

A. $f(X,Y)=0$

B. $f(X,Y)=1$

C. $f(X,Y)=2$

Boolean Algebra

For two Boolean variables X and Y with $X=1$, $Y=0$, what is function $f(X,Y)=(X+Y)Y$?

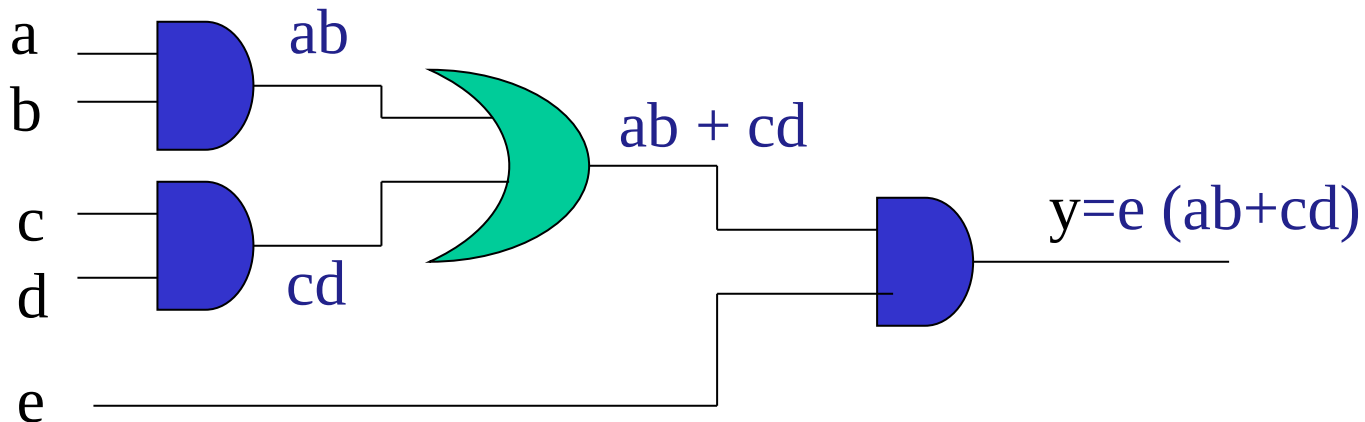
A. $f(X,Y)=0$

B. $f(X,Y)=1$

C. $f(X,Y)=2$

So, what is the point of representing gates as symbols and Boolean expressions?

- Given the Boolean expression, we can draw the circuit it represents by cascading gates (and vice versa)



Logic circuit vs. Boolean Algebra Expression

Simplify the Boolean expression: Reduce the complexity of the circuit

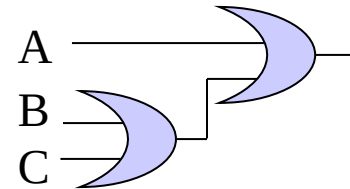
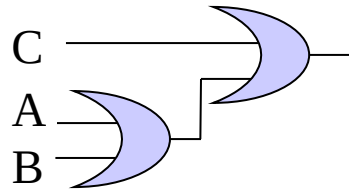
Switching Expression and Logic Diagrams

- Switching Expression –
 - Equations - # literals, # variables, # operators
 - Literal is a variable or its complement (e.g. a , a')
 - Variables (e.g. x)
 - Operator (e.g. $+$, $\cdot$)
 - Schematic / Logic Diagram - # of gates, # nets (wires), # of pins
 - Gate (and, or, etc) – can be more than 2 inputs (e.g. 3 input AND gate)
 - Net – wire that connects gates
 - Pin - input or output of a gate.

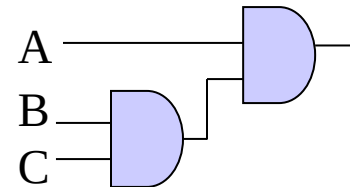
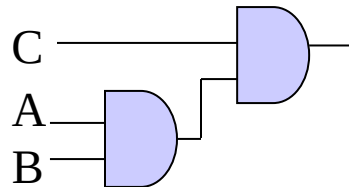
Laws and Logic Diagrams

Associativity Laws

$$(A+B) + C = A + (B+C)$$



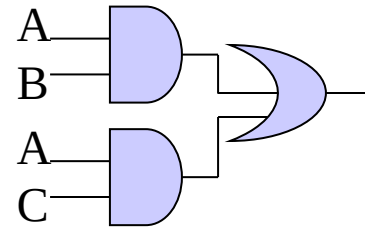
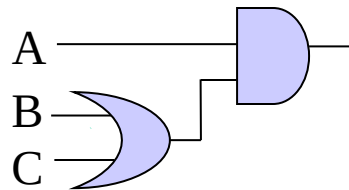
$$(AB)C = A(BC)$$



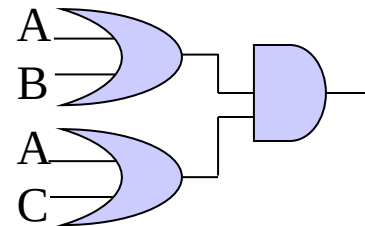
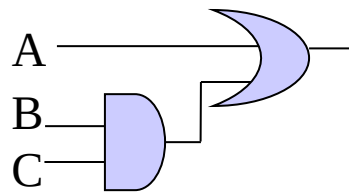
Laws and Logic Diagrams

Distributive Laws

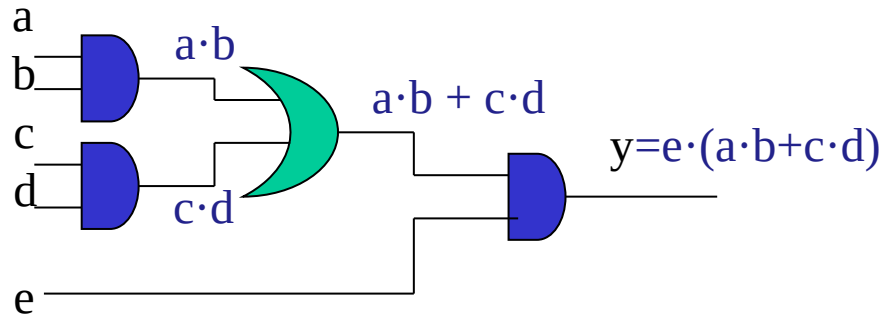
$$A \cdot (B + C) = A \cdot B + A \cdot C$$



$$A + B \cdot C = (A + B) \cdot (A + C)$$



Switching Expression and Logic



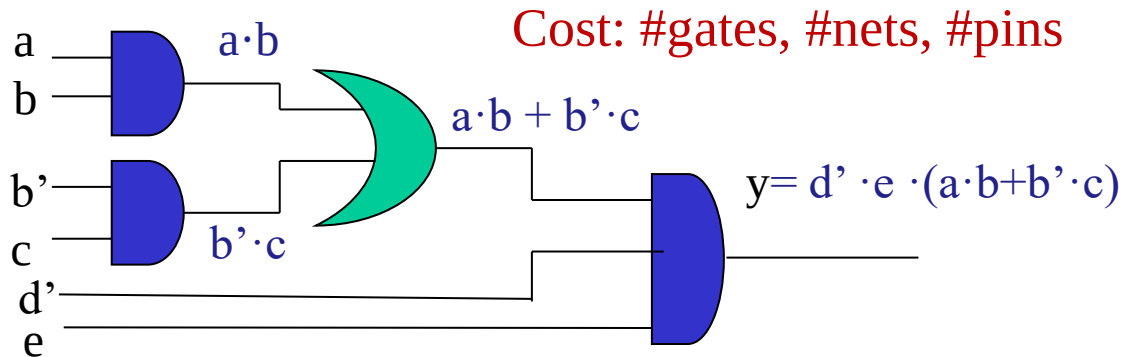
Schematic Diagram:

5 primary inputs
1 primary output
4 components (gates)
9 signal nets
12 pins

Boolean Algebra:

5 variables
1 expression
4 operators
5 literals

Switching Expression and Logic



Schematic Diagram:

6 primary inputs

1 primary output

4 gates (3 ANDs, 1 OR)

10 signal nets

11 pins

Nets are wires, Gates $\rightarrow$ transistors

Switching Expression:

5 variables

1 expression

4 operators (3 ANDs, 1 OR)

6 literals

Example: $f(a, b, c) = ab + a'c + a'b'$

# variables	
# literals	
# gates	

Which statement is not true in general?

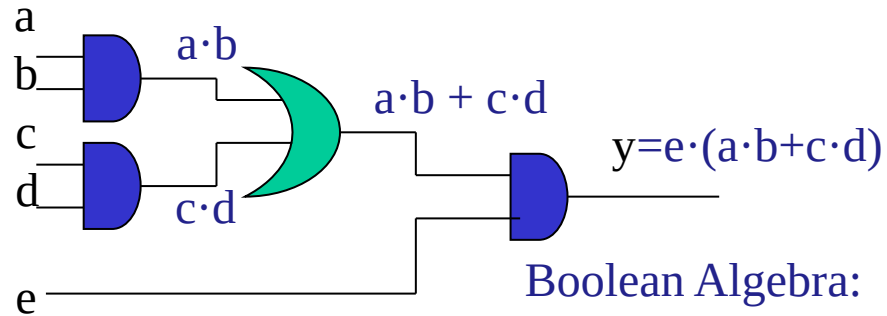
Schematic Diagram:

5 primary inputs

4 components (gates)

9 signal nets

12 pins



Boolean Algebra:

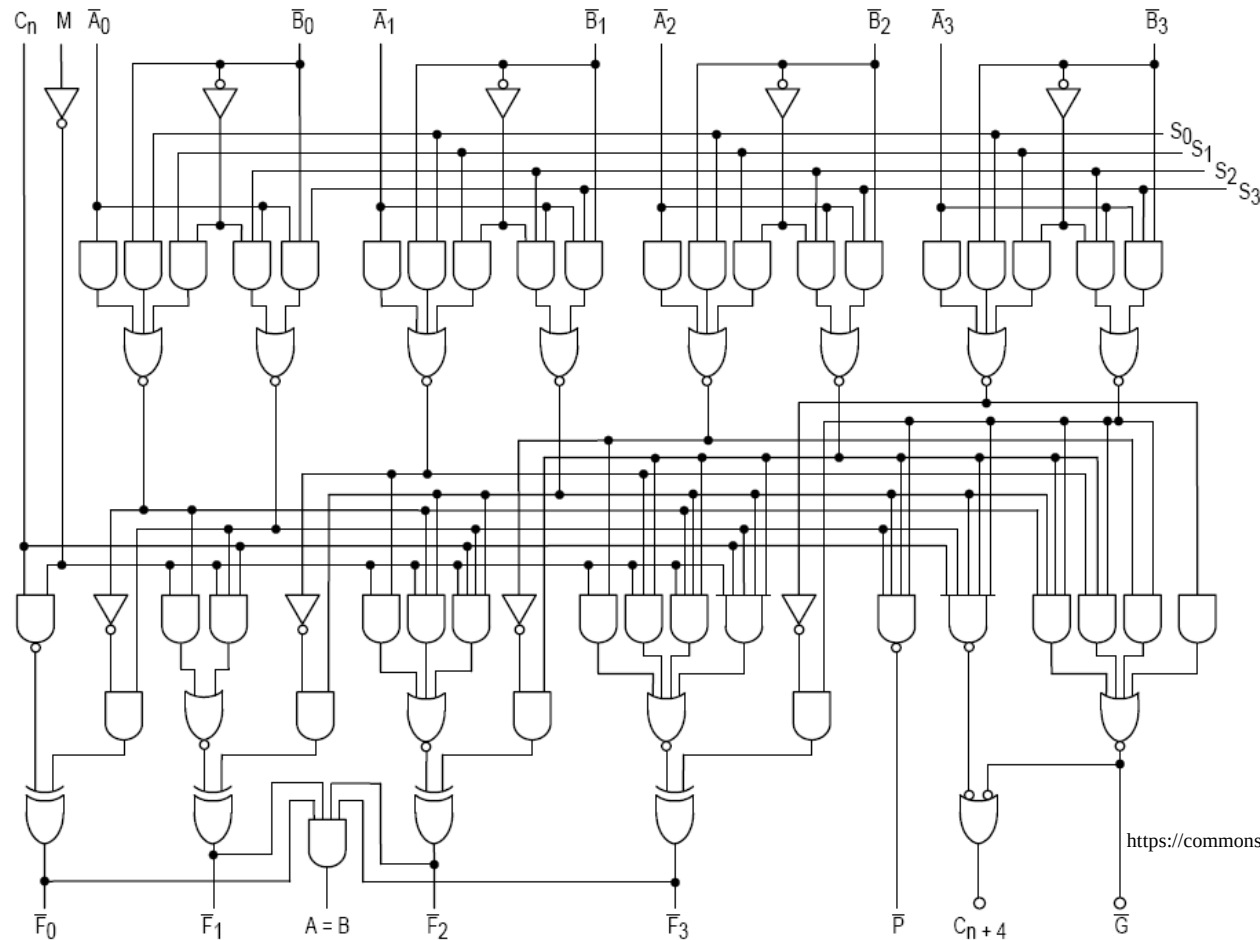
$$y = e \cdot (a \cdot b + c \cdot d)$$

5 literals

4 operators

- A. #primary inputs = # literals
- B. #gates = # operators
- C. #nets = #variables + # operators
- D. #pins = # literals + 2 * #operators - 1
- E. All of them

Logic Diagram/Schematics



<https://commons.wikimedia.org/wiki/File:74181aluschematic.png>

- Logic circuit vs. Boolean Algebra Expression
- Simplify the Boolean expression: Reduce the complexity of the circuit

Next class

- Designing Combinational circuits