CS/EE120A: Logic Design

Jia Chen
jiac@ucr.edu

Logic Design?

Basic organization of the circuitry of a digital computer.

- ✓ Two-valued logic system (binary code): 1/0, on/off
- ✓ Logic gates made of integrated circuits for calculations
- ✓ Three basic kinds of logic gates: AND, OR, NOT

Computer

Merriam-Webster®

A programmable usually electronic device that can
store, retrieve, and process
data

Digital Computers

Analog Computers

The Antikythera mechanism, dating between 150 and 100 BC, was an early analog computer: calculate astronomical positions

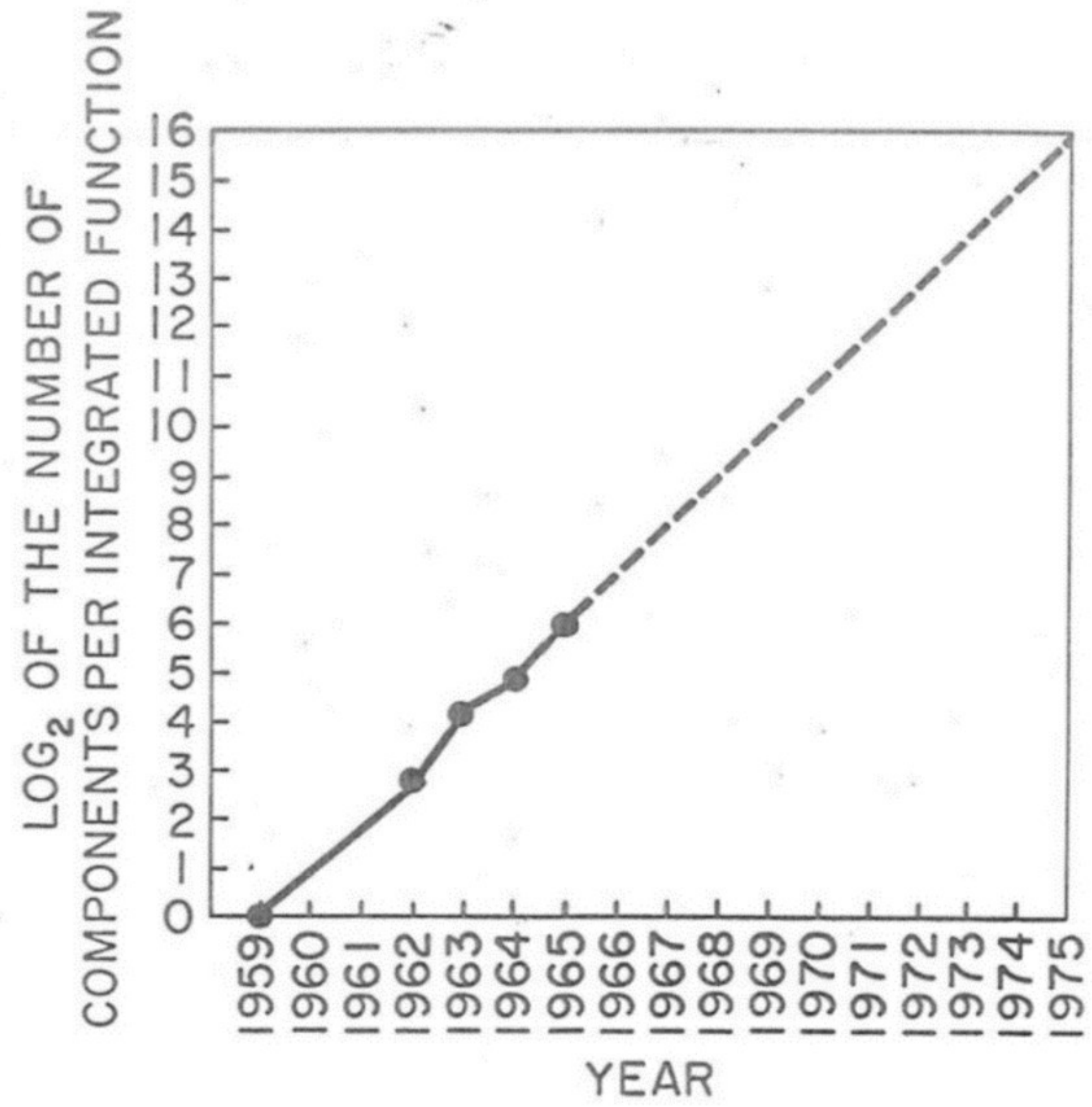
The Norden bombsight was a highly sophisticated optical/mechanical analog computer used by the United States Army Air Force during World War II to aid the pilot of a bomber aircraft in dropping bombs accurately.

MONIAC (Monetary National Income Analogue Computer), was created in 1949 by the New Zealand economist Bill Phillips to model the national economic processes of the UK.

Why are digital computers more popular now?

- Please identify how many of the following statements explains why digital computers are now more popular than analog computers.
  ① The cost of building systems with the same functionality is lower by using digital computers.
  ② Digital computers can express more values than analog computers in the same range.
  ③ Digital signals are less fragile to defective components.
  ④ Digital data are easier to store.
  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Moore’s Law

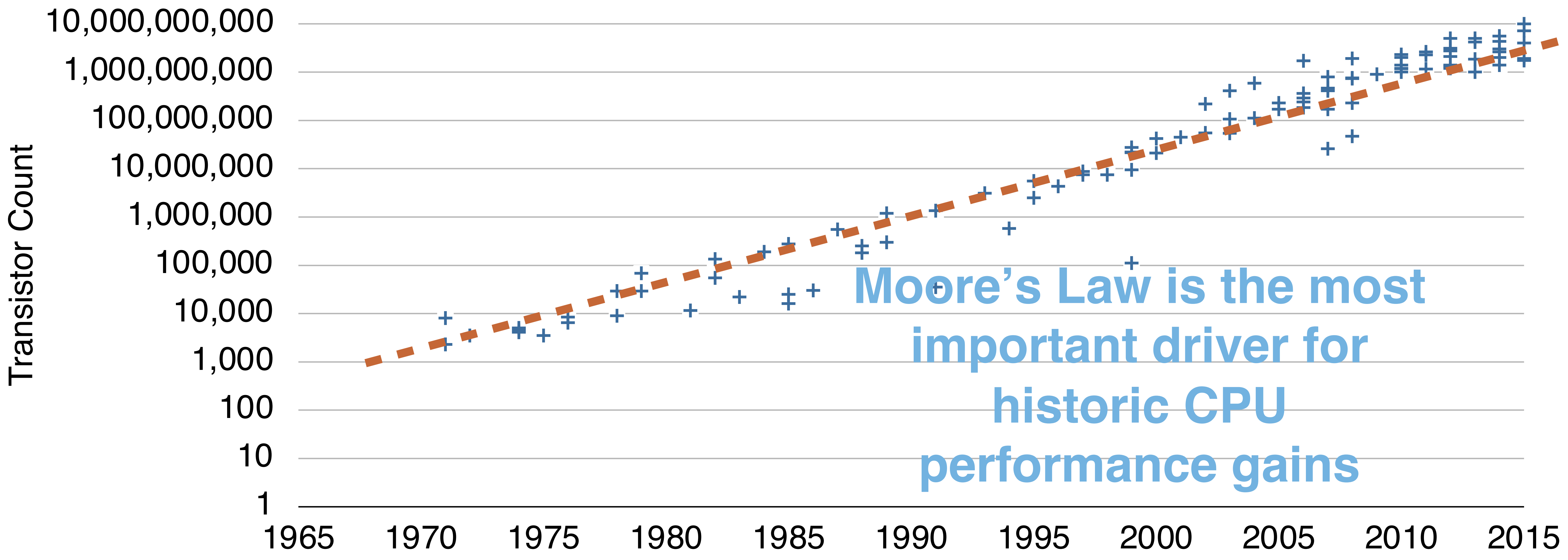


Gordon Moore in 1965

Moore, G. E. (1965), 'Cramming more components onto integrated circuits', Electronics 38 (8) .

Moore's Law[1]

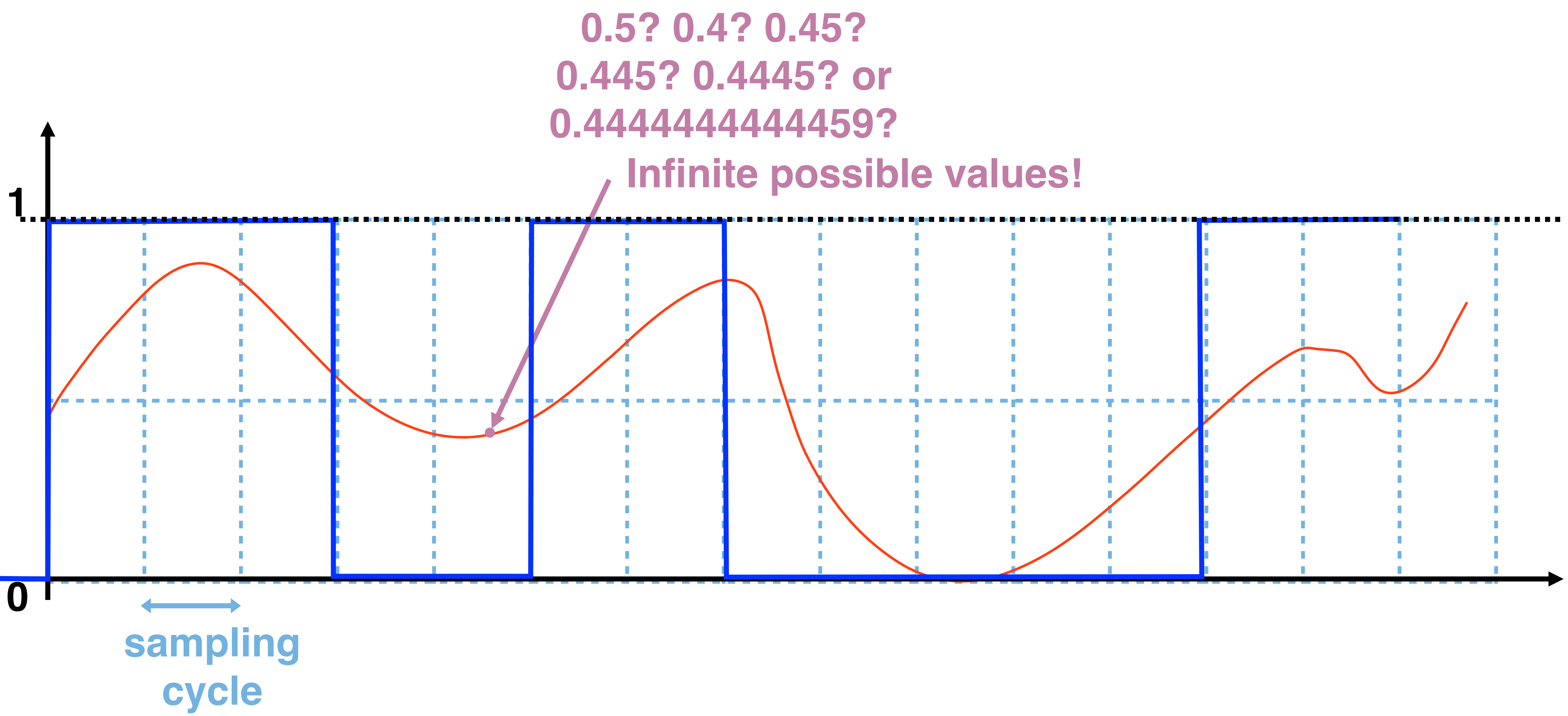
- The number of transistors we can build in a fixed area of silicon doubles every 12 ~ 24 months.



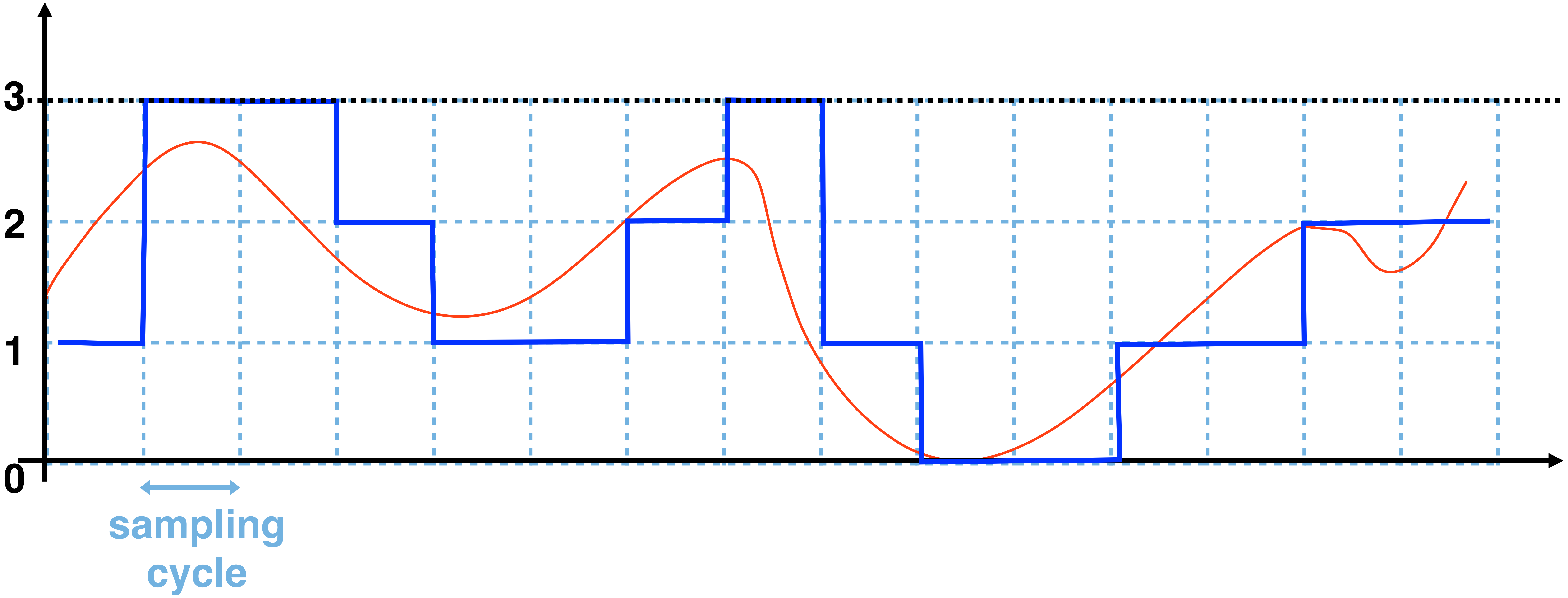
Why are digital computers more popular now?

- Please identify how many of the following statements explains why digital computers are now more popular than analog computers.
  ① The cost of building systems with the same functionality is lower by using digital computers.
  ② Digital computers can express more values than analog computers in the same range.
  ③ Digital signals are less fragile to defective components.
  ④ Digital data are easier to store.
  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Analog v.s. digital signals



Analog v.s. digital signals



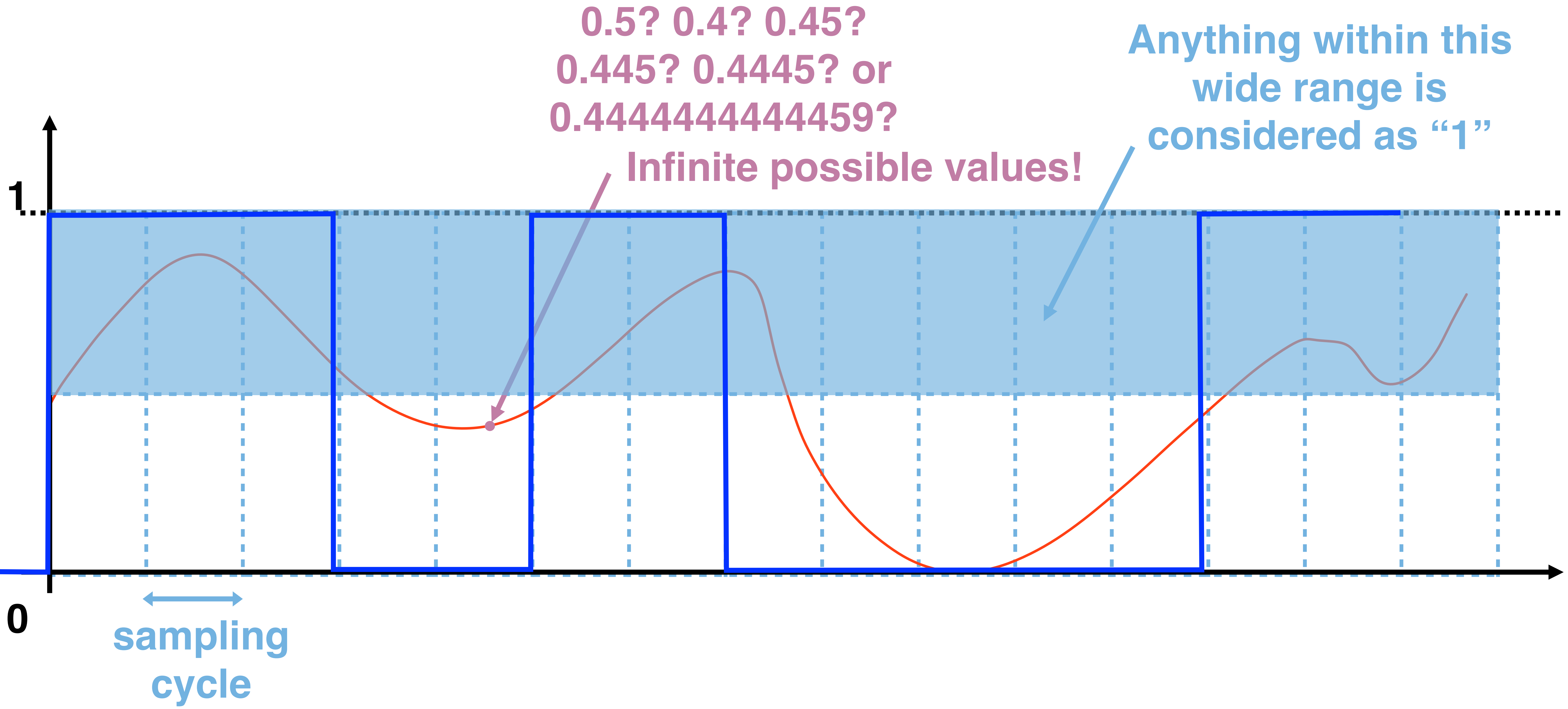
Why are digital computers more popular now?

- Please identify how many of the following statements explains why digital computers are now more popular than analog computers.
  1. The cost of building systems with the same functionality is lower by using digital computers.
  2. Digital computers can express more values than analog computers in the same range.
  3. Digital signals are less fragile to defective components.
  4. Digital data are easier to store.

  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Slide is adapted from Hung-Wei Tseng

Analog v.s. digital signals



Why are digital computers more popular now?

- Please identify how many of the following statements explains why digital computers are now more popular than analog computers.
  1. The cost of building systems with the same functionality is lower by using digital computers.
  2. Digital computers can express more values than analog computers in the same range.
  3. Digital signals are less fragile to defective components.
  4. Digital data are easier to store.

  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Analog data storage

VS

Digital data storage

Samples per second

- CD Audio = 44,100
- DVD Audio = 192,000

Why are digital computers more popular now?

- Please identify how many of the following statements explains why digital computers are now more popular than analog computers.
  1. The cost of building systems with the same functionality is lower by using digital computers.
  2. Digital computers can express more values than analog computers in the same range.
  3. Digital signals are less fragile to defective components.
  4. Digital data are easier to store.

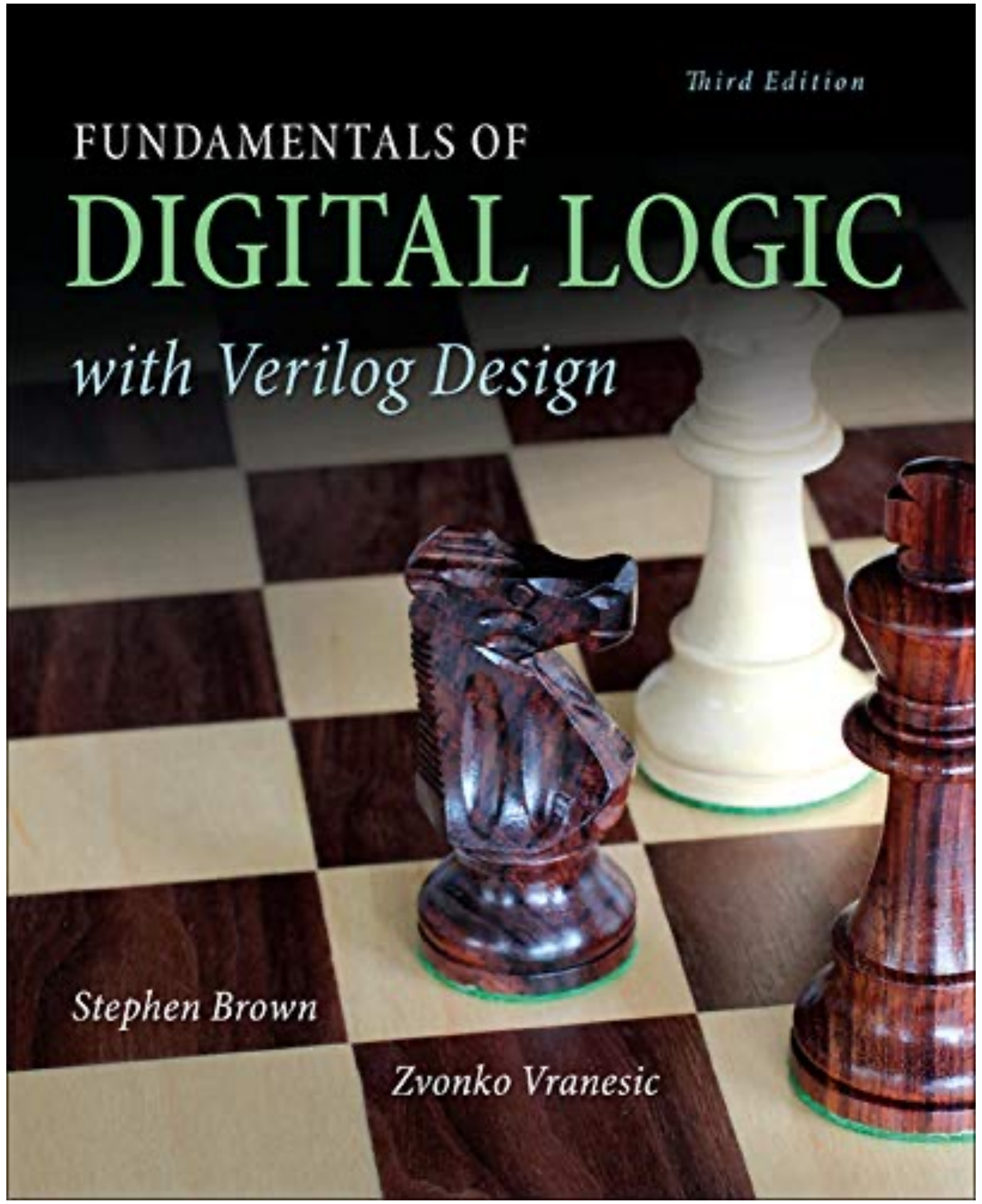
  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Topics of this quarter

- Combinational Logic
  - Logic gates
  - Boolean algebra
  - K-map
- Sequential Logic
  - Finite state machines
  - Clock
  - Flip-flops
- Datapath Components
  - Adder, mux, multipliers, comparator, encoder, decoder
  - Registers, shifters, counter
- HLSM (High-Level State Machine)
- RTL (register transfer level) Design
- Verilog programming in Labs

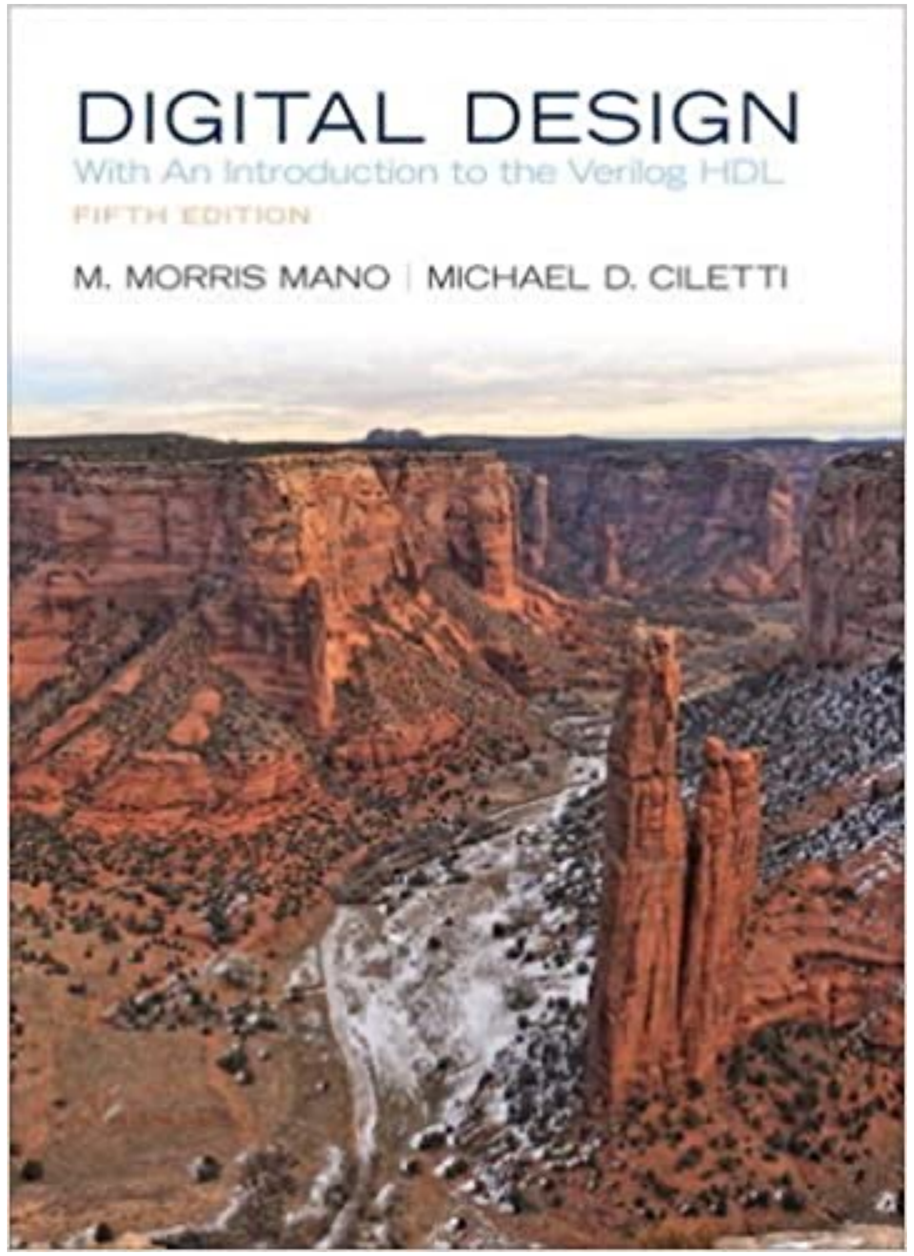
Textbook references

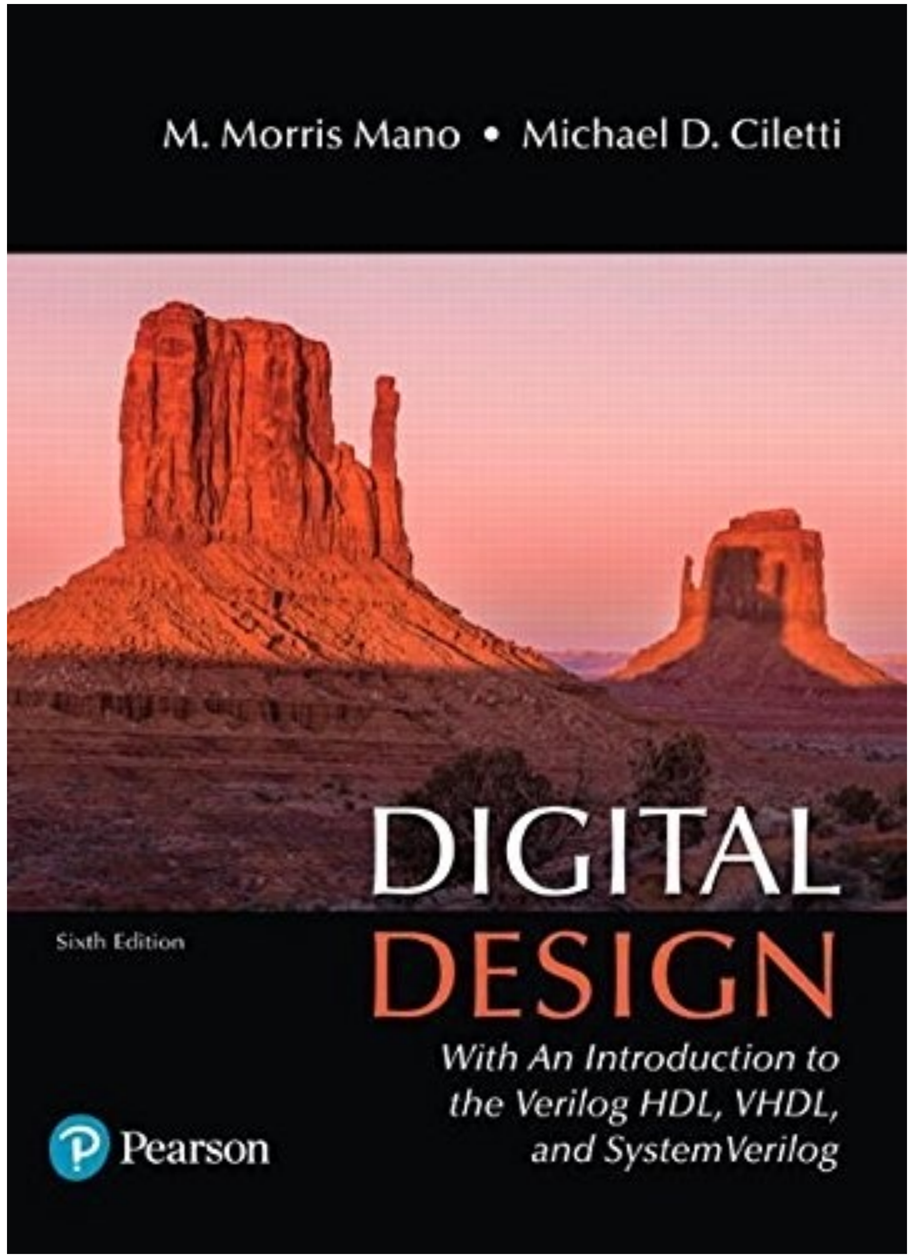
S. Brown and Z. Vranesic, Fundamentals of Digital Logic with Verilog Design, McGraw Hill, Third Edition, ISBN 978-0-07-338054-4.



Textbook references

M. Morris Mano and Michael D. Ciletti, Digital Design with an Introduction to the Verilog HDL, PEARSON, Fifth Edition, ISBN-13: 978-0-13-277420-8.





Lab

- We will have 5-6 labs
  - Using Verilog
  - Using simulation tools to verify and evaluate your design
- 3 students per lab group
- One lab report submission per group per assignment

Logistics

- Lecture: MWF 1:00 – 1:50 pm

- Instructor: Jia Chen

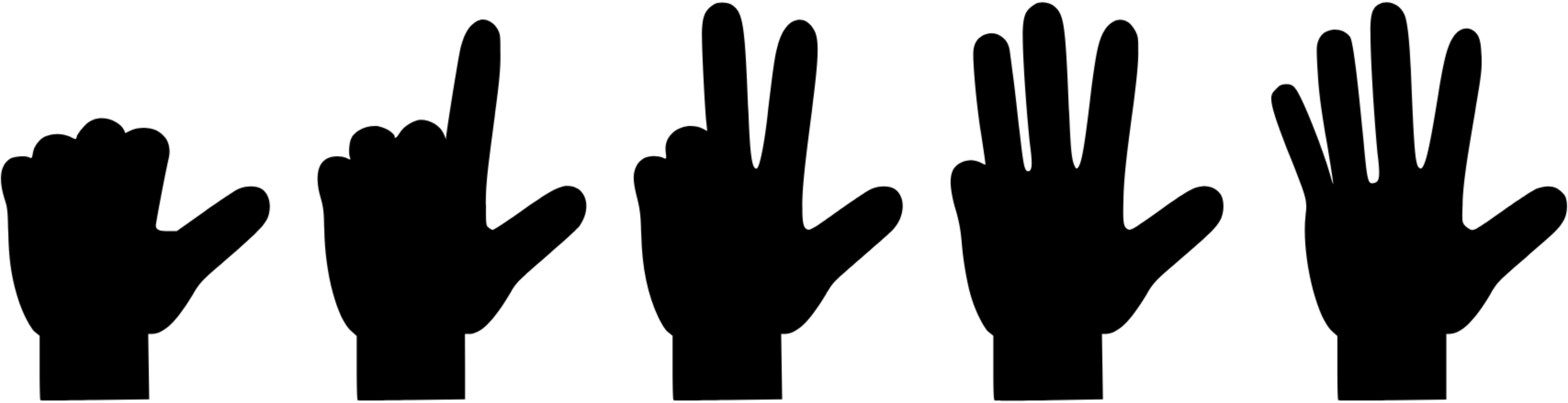
  Email: jiac@ucr.edu
  Office location: Bourns Hall A 149
  Office hours: Tuesday 3:00-5:00 pm in person or by appointment.

- All the materials/announcements can be found on Canvas
- Discussion: Piazza (see signup info. in Syllabus)

Grading

- Homework: 20%
- Labs: 30%
- Midterm: 20%
- Final exam: 30%

10-based number systems is the human-nature



10-based number system is popular since thousands of years ago

1: 𓏺

10: 𓎆

100: 𓍢

1000: 𓆼

10000: 𓂭

100000: 𓆐

1000000: 𓁨

𓆼𓆼 𓎆 𓎆 𓏺 𓏺 𓏺 𓏺 = ?

10-based number system is popular since thousands of years ago

1: 𓏺

10: 𓎆

100: 𓍢

1000: 𓆼

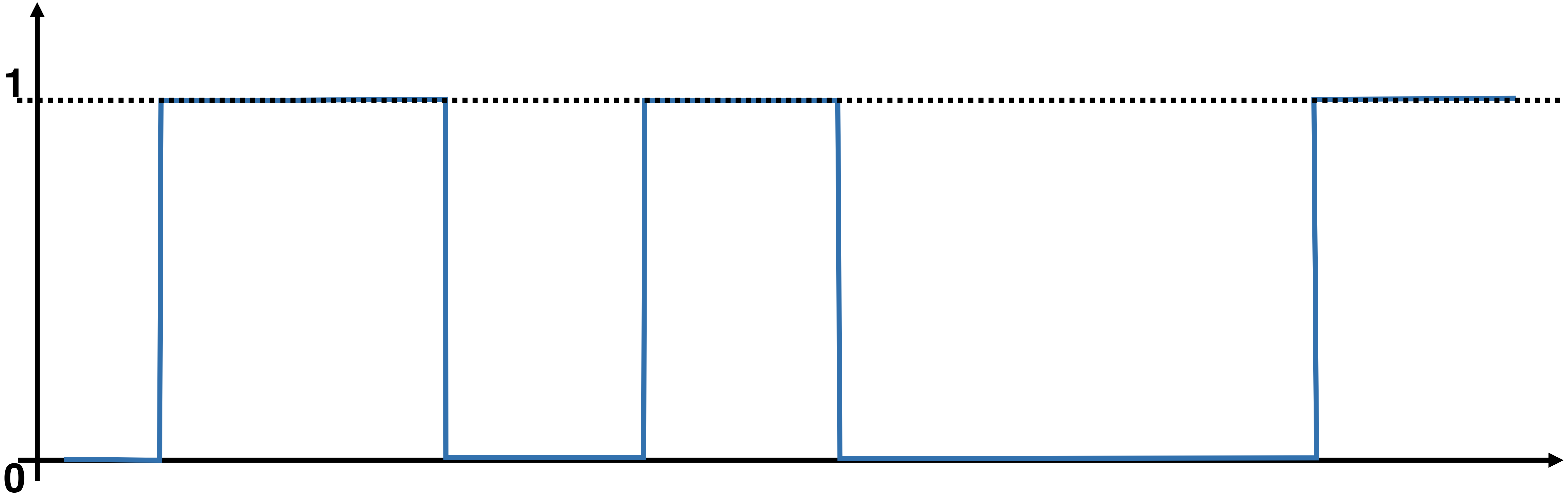
10000: 𓂭

𓆼𓆼 𓎆 𓎆 𓏺 𓏺 𓏺 𓏺 = 2024

100000: 𓆐

1000000: 𓁨

But digital circuits only have 0s and 1s...



Connection to CS 061

Machine Organization and Assembly Language Programming

Survey

Have you taken CS 061?

Survey

Are you familiar with unsigned and signed numbers?

Review: unsigned binary numbers

The basic idea of a number system

- Each position represents a quantity; symbol in position means how many of that quantity
  - Decimal (base 10)
    - Ten symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
    - More than 9: next position
    - Each position is incremented by power of 10
  - Binary (base 2)
    - Two symbols: 0, 1
    - More than 1: next position
    - Each position is incremented by power of 2

$$\mathbf{3} \times 10^2 + \mathbf{2} \times 10^1 + \mathbf{1} \times 10^0 = 300 + 20 + 1 = 321$$

$$\mathbf{1} \times 2^3 + \mathbf{0} \times 2^2 + \mathbf{0} \times 2^1 + \mathbf{1} \times 2^0 = 1 \times 2^3 + 1 \times 2^0 = 1 \times 8 + 1 \times 1 = 9$$

Converting from decimal to binary

2	321	
2	160	 1
2	80	 0
2	40	 0
2	20	 0
2	10	 0
2	5	 0
2	2	 1
	1	 0

$321 = (101000001)_2$

Other frequently used number systems

- Octal — base of 8
  - 8 symbols: 0, 1, 2, 3, 4, 5, 6, 7
  - More than 7: next position
  - Each position is incremented by power of 8
  - Easy conversion from binary — merge 3-digit into one

$$321 = (101000001)_2$$

$$321 = (101\ 000\ 001)_2 = (5\quad 0\quad 1)_8$$

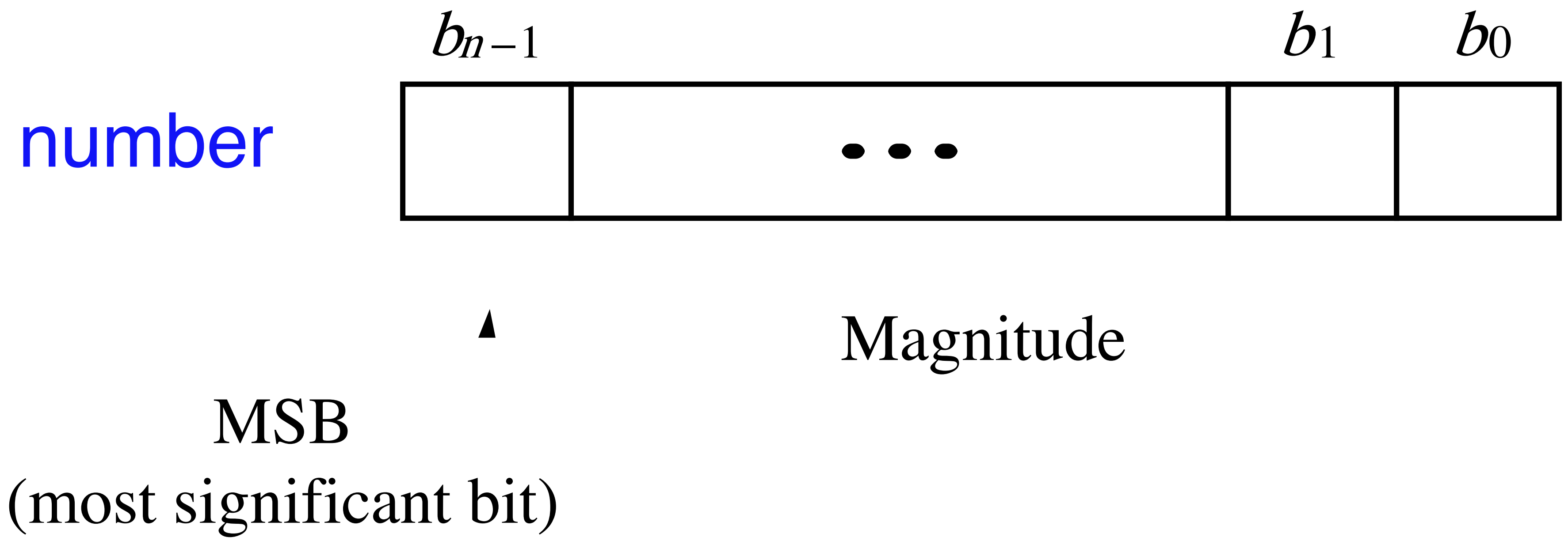
- Hexdecimal — base of 16
  - 16 symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
  - More than 15: next position
  - Each position is incremented by power of 16
  - Easy conversion from binary — merge 4-digit into one

$$321 = (1\ 0100\ 0001)_2 = (1\quad 4\quad 1)_{16}$$

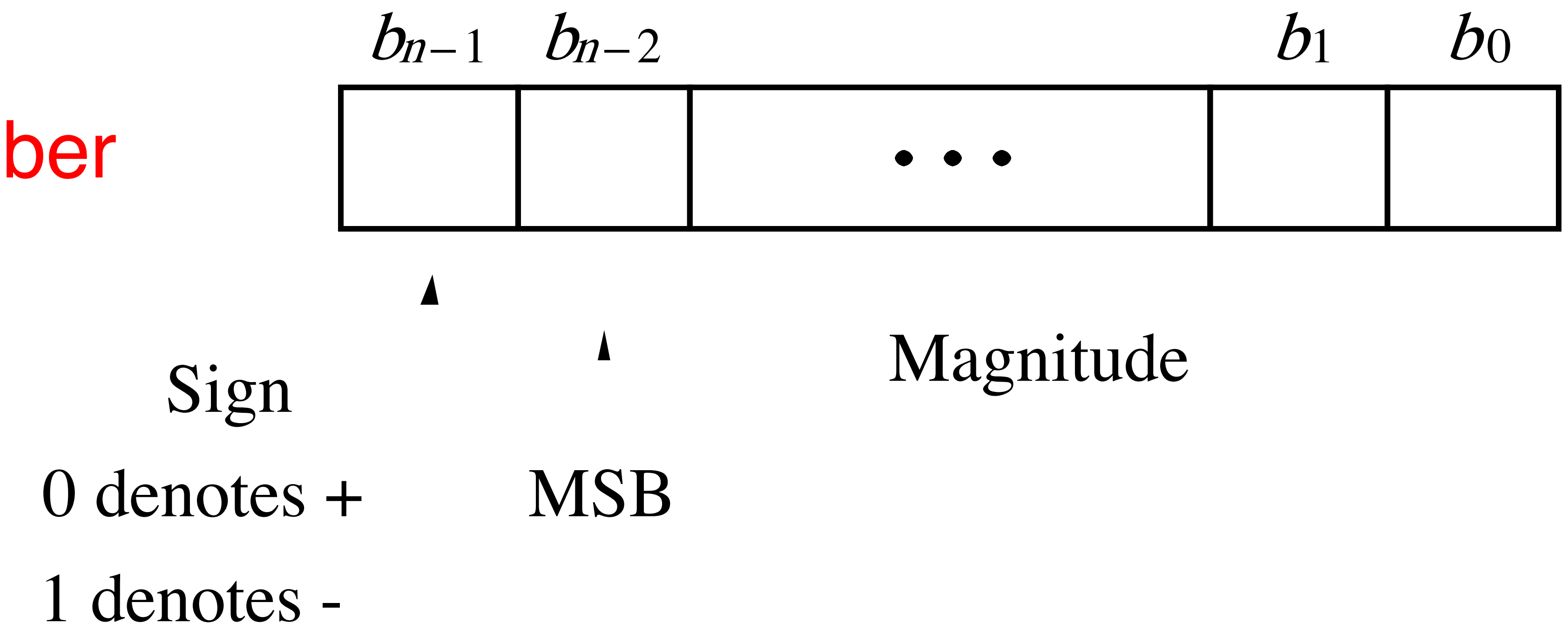
Review: signed binary numbers

Unsigned numbers vs signed numbers

(a) Unsigned number



(b) Signed number



Three types of signed number representations

- Sign-and-magnitude
- 1's complement
- 2's complement

Positive Numbers

The three representations are the same.

Ex: singed number $0110_2 = 4 + 2 = 6_{10}$

"0" means positive

Negative Numbers

Three representations are different.

- Sign-and-magnitude
- 1's complement
- 2's complement

Sign-and-Magnitude

- Straightforward:
  - Sign bit = 0 positive; sign-bit = 1 negative
  - The rest bits indicate the magnitude (like unsigned number)
  - For example

    +5 = 0101

    –5 = 1101
- Easy for us, but not suited for efficient operations in computers.

1's Complement

- Positive number: same as sign-and-magnitude positive number.
- Negative number: complement each bit of the corresponding positive number, including the sign bit.
- For example

$$+5 = 0101$$

complement each bit

$$-5 = 1010$$

2’s Complement

- Positive number: same as sign-and-magnitude and 1’s complement positive number.
- Negative number: complement each bit of the corresponding positive number, including the sign bit, and then add it with 1.
- For example

+5 = 0101

complement each bit

1010
+ 1

add it with 1

-5 = 1011

2's Complement

- A simple rule to find the 2's complement of a negative number:
  - *Examine the bits of the corresponding positive number from the right to the left,* ***copy all bits that are 0s and the first bit that is 1, then complement the rest of the bits.***

Using the rule: +5 = $\boxed{010}\boxed{1}$ -5 = $\boxed{101}\boxed{1}$

Using the rule: +2 = $\boxed{00}\boxed{10}$ -2 = $\boxed{11}\boxed{10}$

$\boxed{}$ copy

$\boxed{}$ complement

2's Complement Addition

$$\begin{array}{rr} (+5) \\ +\ (+2) \\ \hline (+7) \end{array} \qquad \begin{array}{r} 0\,1\,0\,1 \\ +\ 0\,0\,1\,0 \\ \hline 0\,1\,1\,1 \end{array} \qquad\qquad \begin{array}{r} (-5) \\ +\ (+2) \\ \hline (-3) \end{array} \qquad \begin{array}{r} 1\,0\,1\,1 \\ +\ 0\,0\,1\,0 \\ \hline 1\,1\,0\,1 \end{array}$$

$$\begin{array}{r} (+5) \\ +\ (-2) \\ \hline (+3) \end{array} \qquad \begin{array}{r} 0\,1\,0\,1 \\ +\ 1\,1\,1\,0 \\ \hline \color{red}{1}\ 0\,0\,1\,1 \end{array} \qquad\qquad \begin{array}{r} (-5) \\ +\ (-2) \\ \hline (-7) \end{array} \qquad \begin{array}{r} 1\,0\,1\,1 \\ +\ 1\,1\,1\,0 \\ \hline \color{red}{1}\ 1\,0\,0\,1 \end{array}$$

ignore ignore

- If there is a carry-out from the sign-bit position, it is simply ignored.
- The 2's complement notation is highly suitable for addition.

2's Complement Subtraction

- Easy way: find the 2's complement of the subtrahend, and add

$$\begin{array}{r} (+5) \\ -\ (+2) \\ \hline (+3) \end{array} \qquad \begin{array}{r} 0\,1\,0\,1 \\ -\ 0\,0\,1\,0 \\ \hline \end{array} \qquad \Longrightarrow \qquad \begin{array}{r} 0\,1\,0\,1 \\ +\ 1\,1\,1\,0 \\ \hline 1\,0\,0\,1\,1 \end{array}$$

ignore

$$\begin{array}{r} (-5) \\ -\ (+2) \\ \hline (-7) \end{array} \qquad \begin{array}{r} 1\,0\,1\,1 \\ -\ 0\,0\,1\,0 \\ \hline \end{array} \qquad \Longrightarrow \qquad \begin{array}{r} 1\,0\,1\,1 \\ +\ 1\,1\,1\,0 \\ \hline 1\,1\,0\,0\,1 \end{array}$$

ignore

$$\begin{array}{r} (+5) \\ -\ (-2) \\ \hline (+7) \end{array} \qquad \begin{array}{r} 0\,1\,0\,1 \\ -\ 1\,1\,1\,0 \\ \hline \end{array} \qquad \Longrightarrow \qquad \begin{array}{r} 0\,1\,0\,1 \\ +\ 0\,0\,1\,0 \\ \hline 0\,1\,1\,1 \end{array}$$

$$\begin{array}{r} (-5) \\ -\ (-2) \\ \hline (-3) \end{array} \qquad \begin{array}{r} 1\,0\,1\,1 \\ -\ 1\,1\,1\,0 \\ \hline \end{array} \qquad \Longrightarrow \qquad \begin{array}{r} 1\,0\,1\,1 \\ +\ 0\,0\,1\,0 \\ \hline 1\,1\,0\,1 \end{array}$$

Arithmetic Overflow

- For n bits, if the result is not in the range of -2^{n-1} to $2^{n-1}-1$, arithmetic overflow has occurred.

$$\begin{array}{rr} (+7) & 0\,1\,1\,1 \\ +\,(+2) & +\,0\,0\,1\,0 \\ \hline (+9) & 1\,0\,0\,1 \end{array}$$

$c_4 = 0$
$c_3 = 1$

Overflow

$$\begin{array}{rr} (-7) & 1\,0\,0\,1 \\ +\,(+2) & +\,0\,0\,1\,0 \\ \hline (-5) & 1\,0\,1\,1 \end{array}$$

$c_4 = 0$
$c_3 = 0$

No overflow

$$\begin{array}{rr} (+7) & 0\,1\,1\,1 \\ +\,(-2) & +\,1\,1\,1\,0 \\ \hline (+5) & 1\,0\,1\,0\,1 \end{array}$$

$c_4 = 1$
$c_3 = 1$

No overflow

$$\begin{array}{rr} (-7) & 1\,0\,0\,1 \\ +\,(-2) & +\,1\,1\,1\,0 \\ \hline (-9) & 1\,0\,1\,1\,1 \end{array}$$

$c_4 = 1$
$c_3 = 0$

Overflow

- When the numbers have opposite signs, there is no overflow.
- When the numbers have the same sign, overflow might occur.
- When carry-outs from MSB (most significant bit) and sign-bit have different values, there is overflow. overflow = $c_{n-1} \oplus c_n$

$$\begin{array}{rr} (+7) & 0\,1\,1\,1 \\ +(+2) & +0\,0\,1\,0 \\ \hline (+9) & 1\,0\,0\,1 \end{array} \qquad c_4 = 0,\; c_3 = 1$$

$$\begin{array}{rr} (-7) & 1\,0\,0\,1 \\ +(+2) & +0\,0\,1\,0 \\ \hline (-5) & 1\,0\,1\,1 \end{array} \qquad c_4 = 0,\; c_3 = 0$$

$$\begin{array}{rr} (+7) & 0\,1\,1\,1 \\ +(-2) & +1\,1\,1\,0 \\ \hline (+5) & 1\,0\,1\,0\,1 \end{array} \qquad c_4 = 1,\; c_3 = 1$$

$$\begin{array}{rr} (-7) & 1\,0\,0\,1 \\ +(-2) & +1\,1\,1\,0 \\ \hline (-9) & 1\,0\,1\,1\,1 \end{array} \qquad c_4 = 1,\; c_3 = 0$$

- Carry-outs from MSB and sign-bit have different values -> overflow

  overflow = $c_{n-1} \oplus c_n$

- Alternative way to check overflow: if both summands have the same sign but the resulting sum has a different sign.

  EX: $X = x_3x_2x_1x_0,\ Y = y_3y_2y_1y_0,\ S = X + Y$

  overflow = $x_3y_3\bar{s}_3 + \bar{x}_3\bar{y}_3s_3$