BME2322 – Logic Design

The Instructors:

Dr. Görkem SERBES (C317)

gserbes@yildiz.edu.tr

https://avesis.yildiz.edu.tr/gserbes/

Lab Assistants:

Nihat AKKAN

nakkan@yildiz.edu.tr

https://avesis.yildiz.edu.tr/nakkan

LECTURE 1

Assesment

• Midterm 1	:	20%
• Midterm 2	:	20%
• Lab	:	20%
• Final	:	40%

Course Outline

1. Digital Computers, Number Systems, Arithmetic Operations, Decimal, Alphanumeric, and Gray Codes
2. Binary Logic, Gates, Boolean Algebra, Standard Forms
3. Circuit Optimization, Two-Level Optimization, Map Manipulation, Multi-Level Circuit Optimization
4. Additional Gates and Circuits, Other Gate Types, Exclusive-OR Operator and Gates, High-Impedance Outputs
5. Implementation Technology and Logic Design, Design Concepts and Automation, The Design Space, Design Procedure, The major design steps
6. Programmable Implementation Technologies: Read-Only Memories, Programmable Logic Arrays, Programmable Array Logic,Technology mapping to programmable logic devices
7. Combinational Functions and Circuits
8. Arithmetic Functions and Circuits
9. Sequential Circuits Storage Elements and Sequential Circuit Analysis
10. Sequential Circuits, Sequential Circuit Design State Diagrams, State Tables
11. Counters, register cells, buses, & serial operations
12. Sequencing and Control, Datapath and Control, Algorithmic State Machines (ASM)
13. Memory Basics

Recommended books

Main course book:

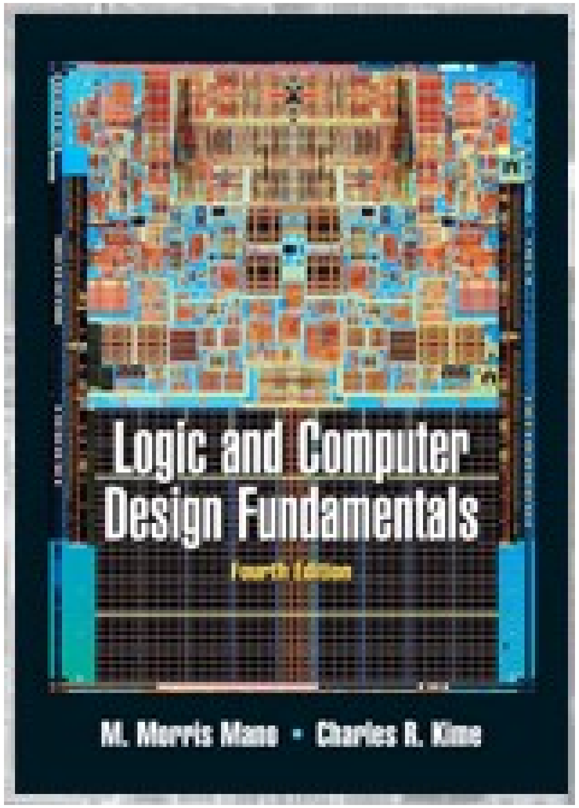
Logic and Computer Design Fundamentals

By M. Mano, Charles Kime.

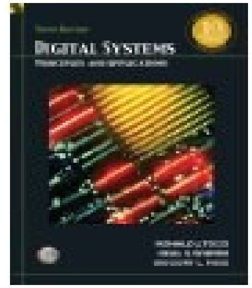
Published by Prentice Hall.

Edition: 4th.

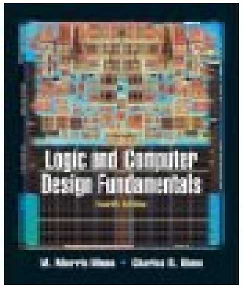
Isbn: 013198926X



Digital Design: Principles and Practices
by John F. Wakerly

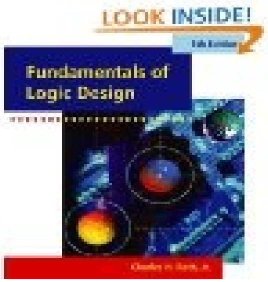


Digital Systems: Principles and Applications
by Ronald Tocci

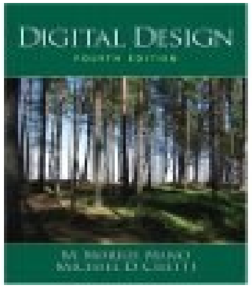


Logic and Computer Design Fundamentals
by M. Morris Mano

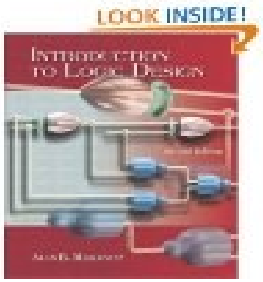
Verilog HDL
by Samir Palnitkar



Fundamentals of Logic Design
by Jr., Charles H. Roth

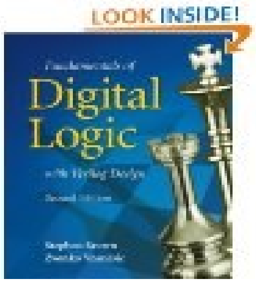


Digital Design
by M. Morris Mano



Introduction to Logic Design
by Alan Marcovitz

Digital Design and Computer Architecture
by David Harris



Fundamentals of Digital Logic with Verilog Design
by Stephen Brown

Rules of the Conduct

- No eating /drinking in class
  - *except water*
- Cell phones must be kept outside of class or switched-off during class
  - *If your cell-phone rings during class or you use it in any way, you will be asked to leave and counted as unexcused absent.*
- No web surfing and/or unrelated use of computers,
  - when computers are used in class or lab.

Rules of the Conduct

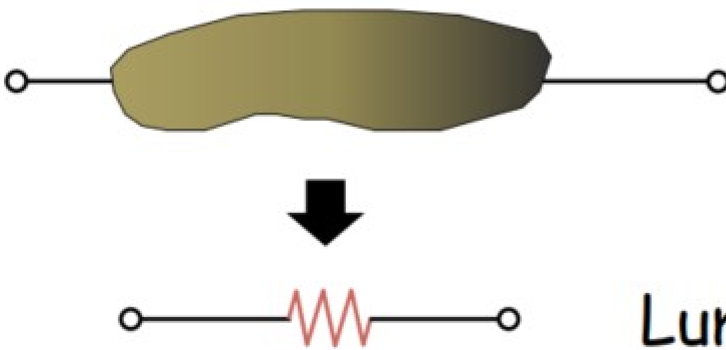
- You are responsible for checking the class web page often for announcements.
- Academic dishonesty and cheating will not be tolerated and will be dealt with according to university rules and regulations
  - *Presenting any work, or a portion thereof, that does not belong to you is considered academic dishonesty.*
- University rules and regulations:
  - http://www.ogi.yildiz.edu.tr/category.php?id=17
  - https://www.yok.gov.tr/content/view/544/230/lang,tr_TR/

Attendance Policy

- The requirement for attendance is **70%**.
  - *Hospital reports are not accepted to fulfill the requirement for attendance.*
  - *The students, who fail to fulfill the attendance requirement, will be excluded from the final exams and the grade of* F0 *will be given.*

Digital Abstraction

- Discretize matter by observing lumped matter discipline



Lumped Circuit Abstraction

- Analysis tool kit

  KVL/KCL, composition, node, superposition, Thévenin, Norton

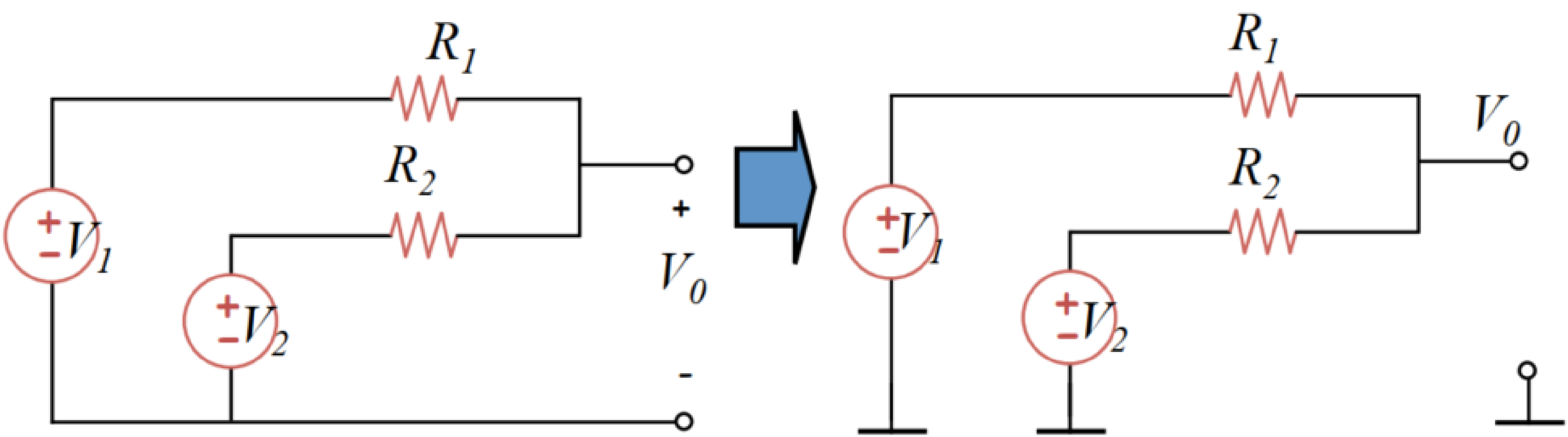
Digital Abstraction (cont.)

In this course we will use Digital Abstraction idea.

But why we need digital signals and systems?

In the past …

Analog signal processing

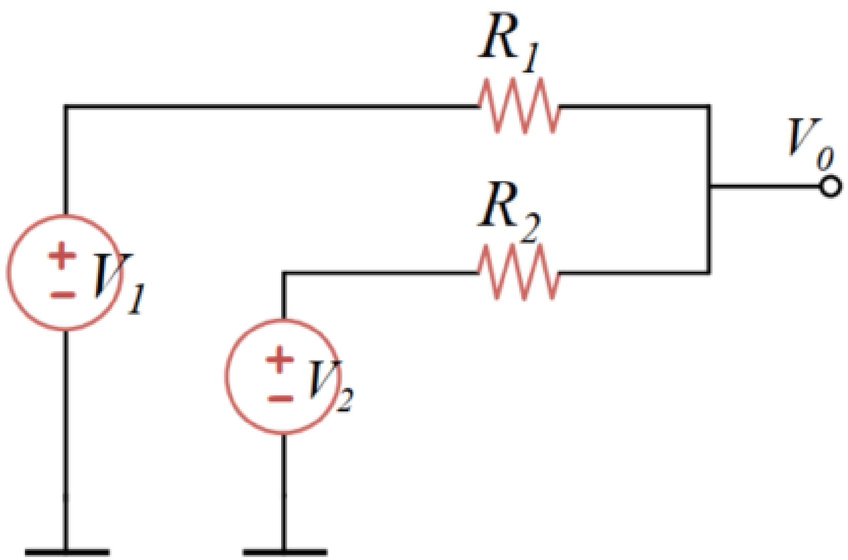


V_1 and V_2 might represent the outputs of two sensors, for e.g.

Shorthand notation (from node method)

Digital Abstraction (cont.)

Analog signal processing



Using Superposition:

$$V_O = \frac{R_2}{R_1 + R_2} V_1 + \frac{R_1}{R_1 + R_2} V_2$$

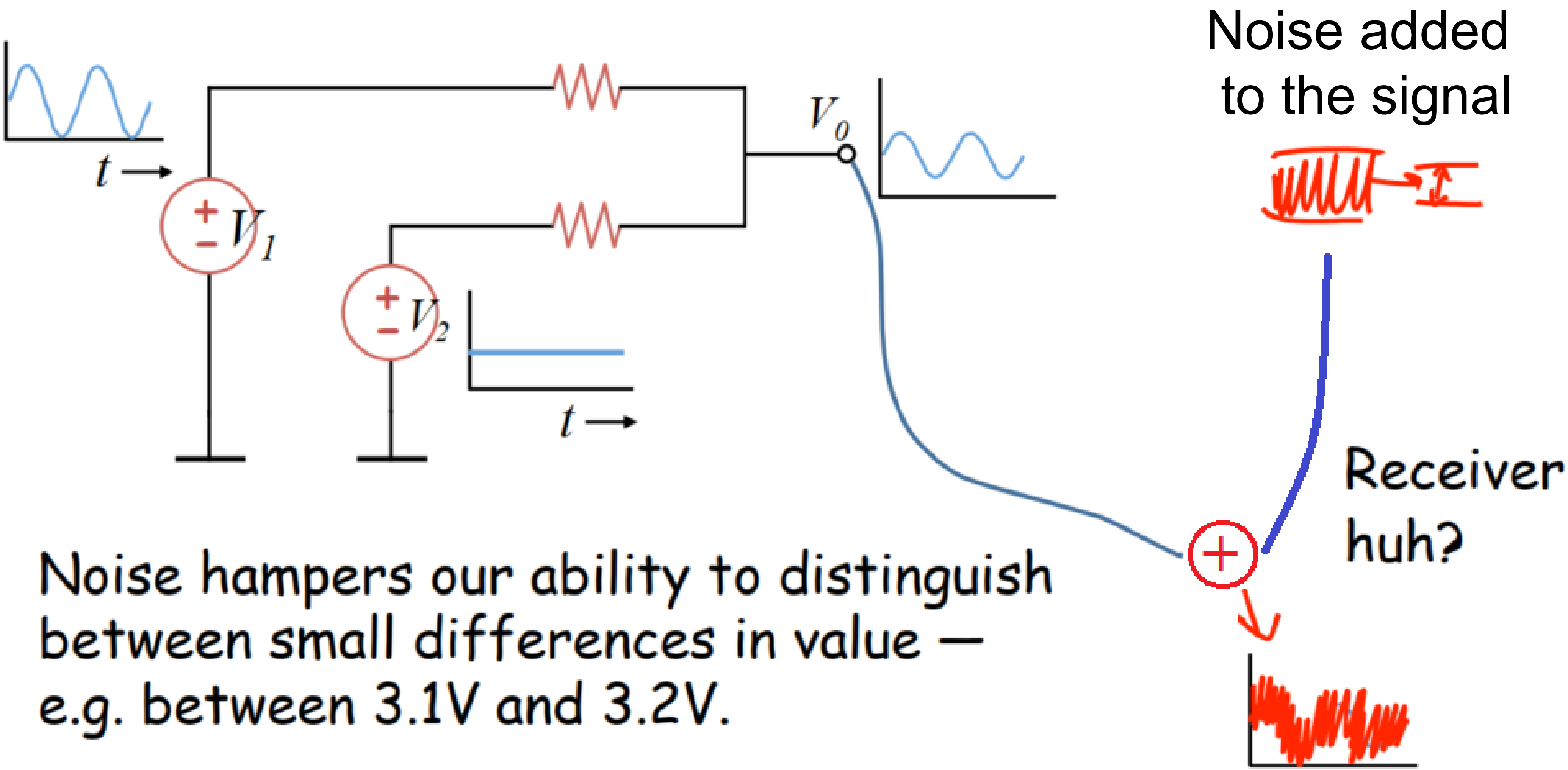
If $R_1 = R_2$

$$V_O = \frac{V_1 + V_2}{2}$$

The above is an “adder” circuit.

Digital Abstraction (cont.)

Noise Problem with Analog



Noise hampers our ability to distinguish between small differences in value — e.g. between 3.1V and 3.2V.

Analog Systems lack noise immunity

Digital Abstraction (cont.)

Idea: Value Discretization

Restrict values to be one of two

High	Low
5V	0V
True	False
‘1’	‘0’

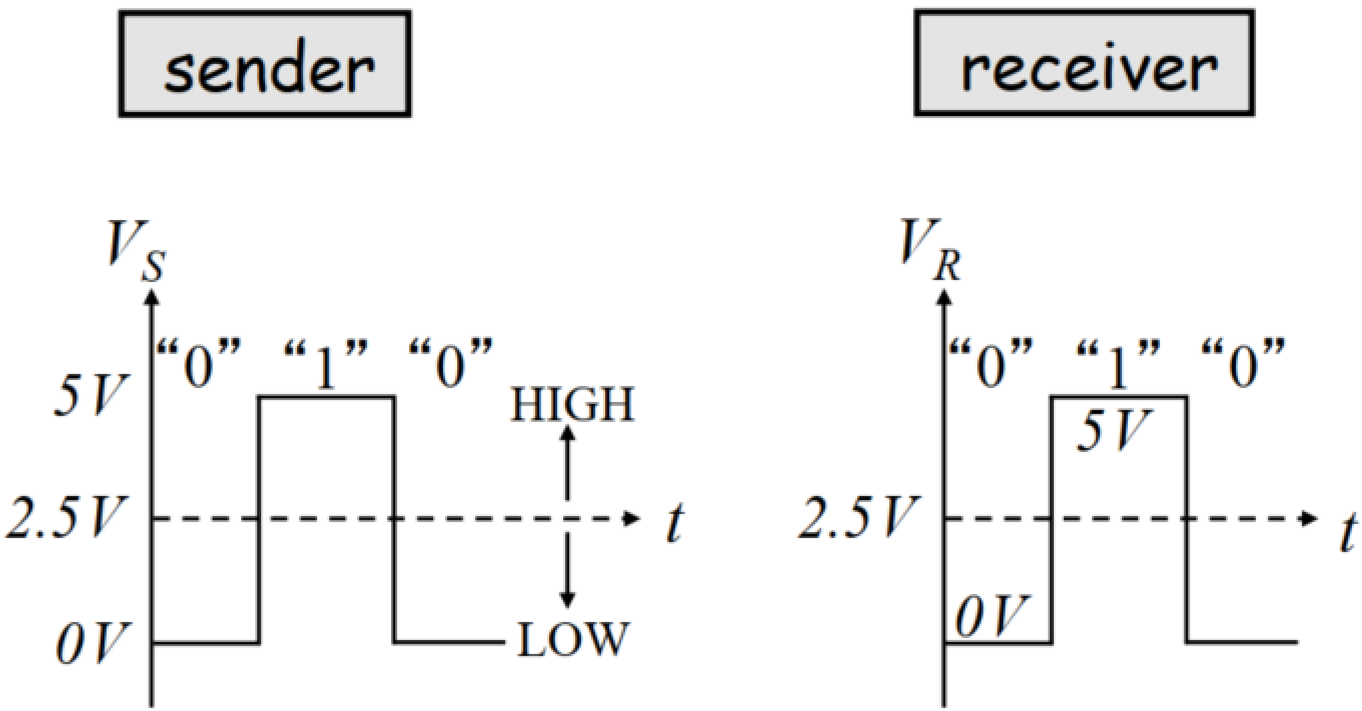
Note: In modern world lower voltage values are used.

...like two digits 0 and 1

Why is this discretization is useful?

Digital System

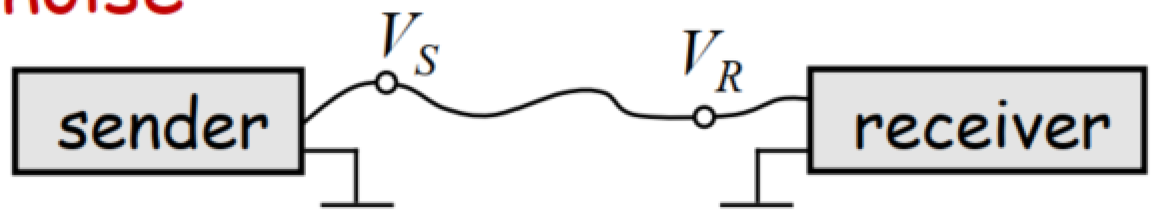
Ideal Case

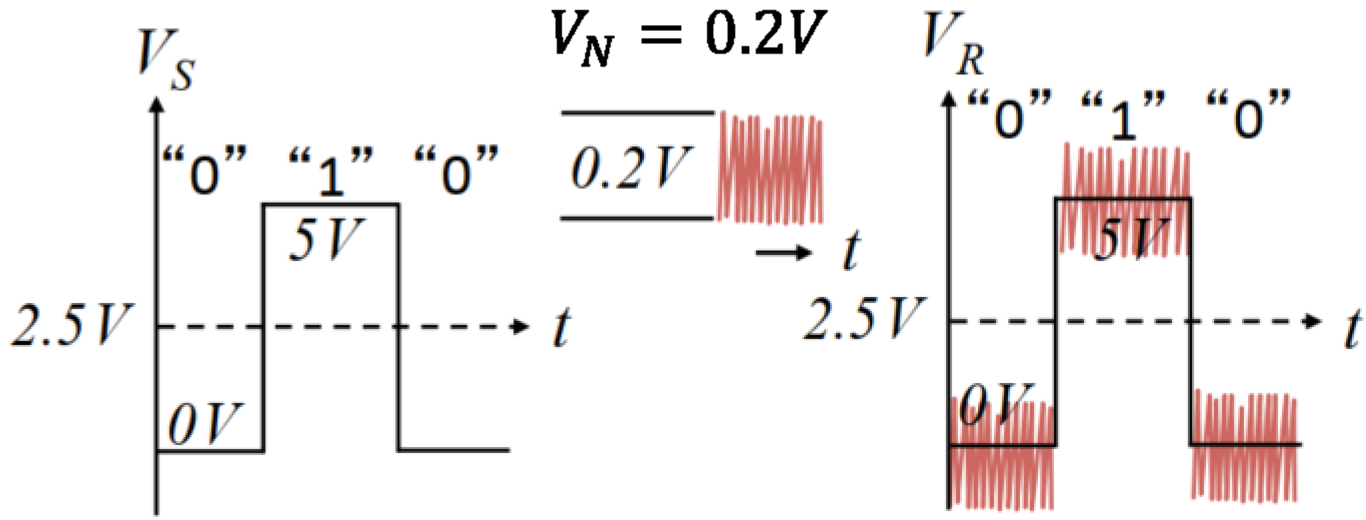


Why is this discretization is useful? (cont.)

Digital System

With noise

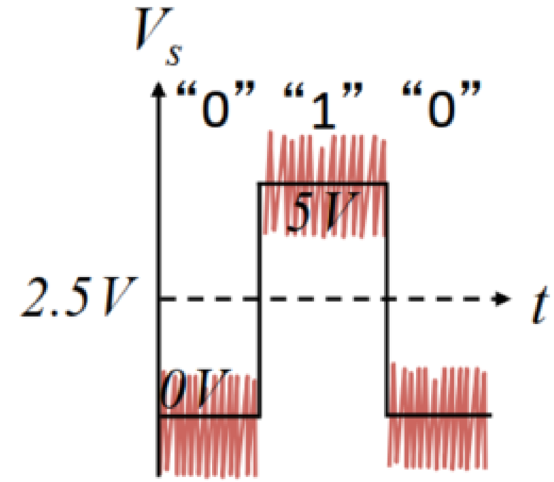




$V_R = V_S + V_N$

Why is this discretization is useful? (cont.)

Digital System

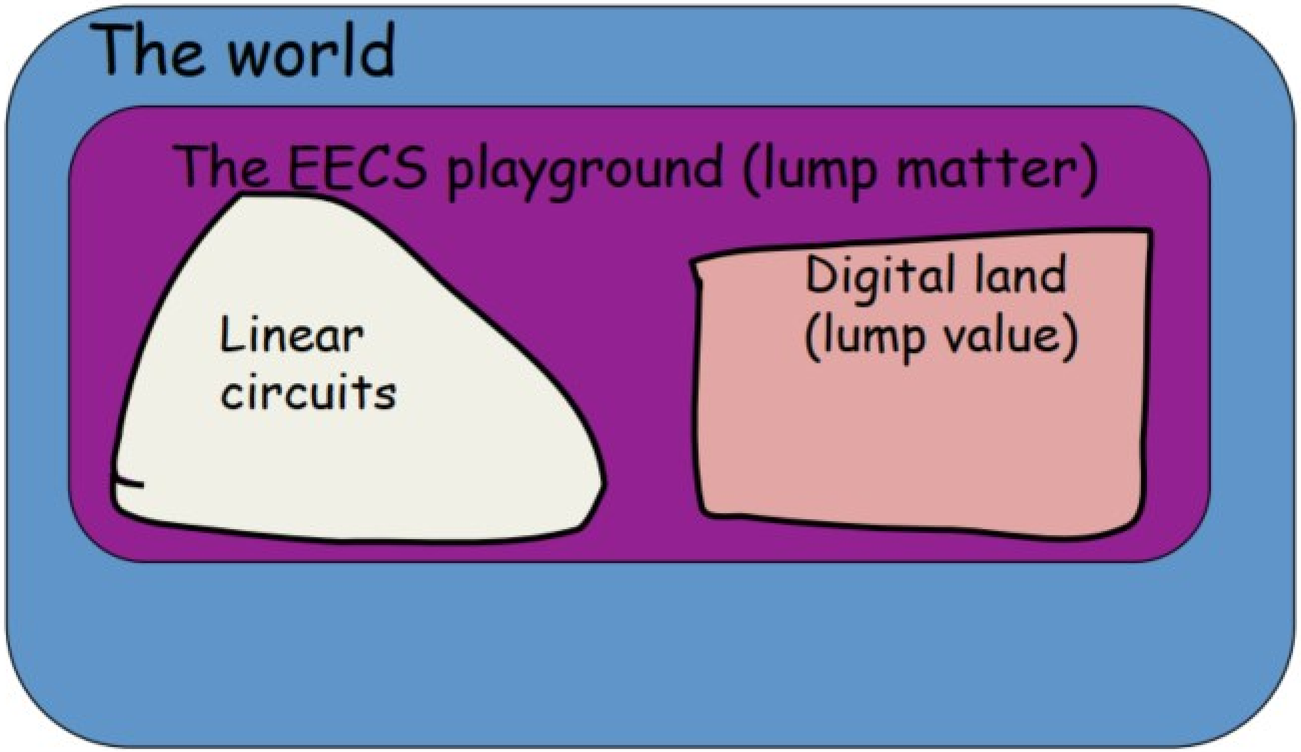


Better noise immunity → Lots of "noise margin"

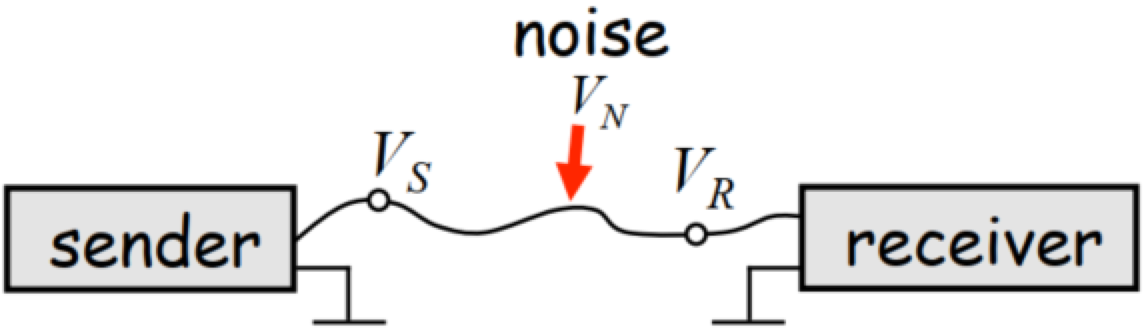
For "1": noise margin $5V$ to $2.5V = 2.5V$

For "0": noise margin $0V$ to $2.5V = 2.5V$

The Big Picture

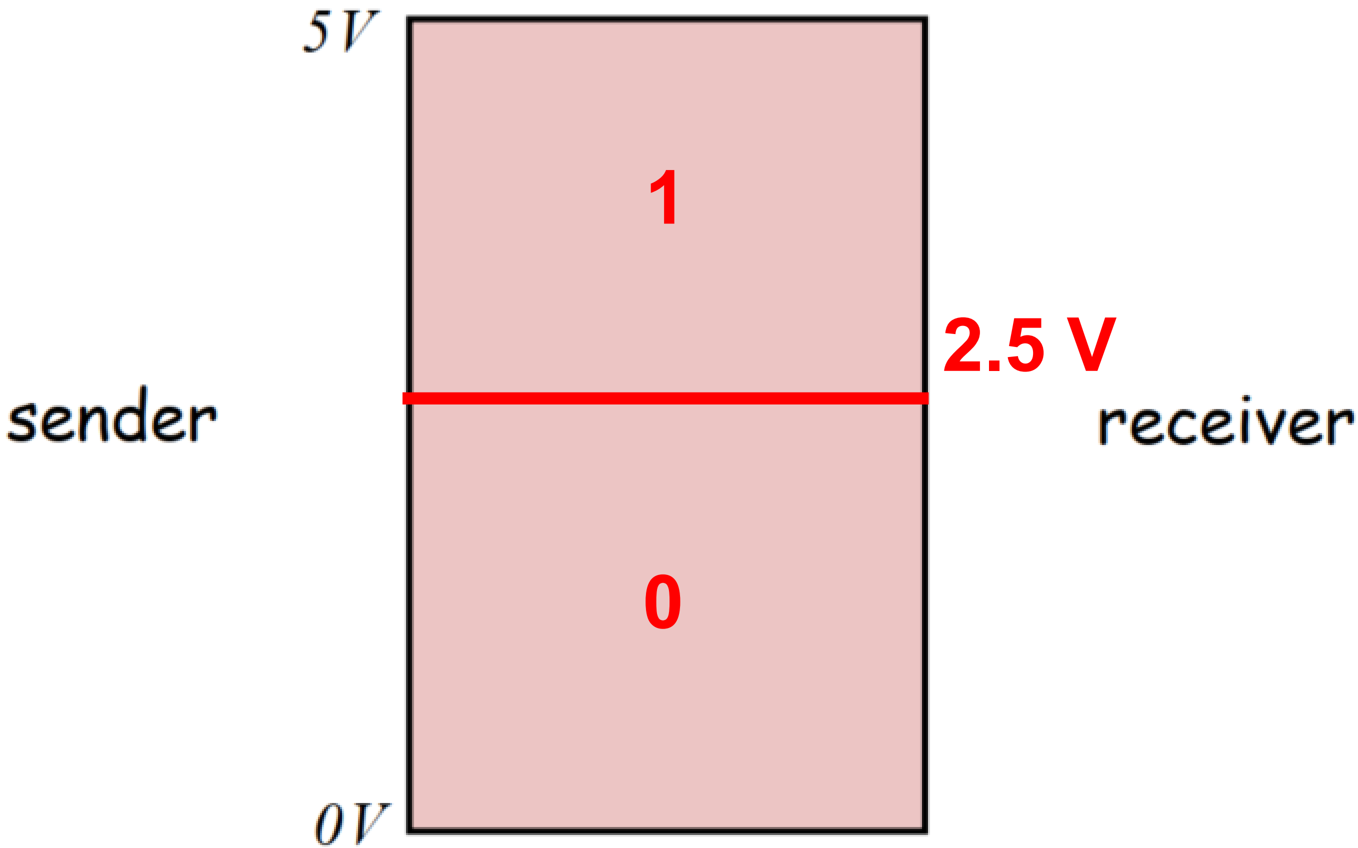


Sender-Receiver Contract



Static Discipline

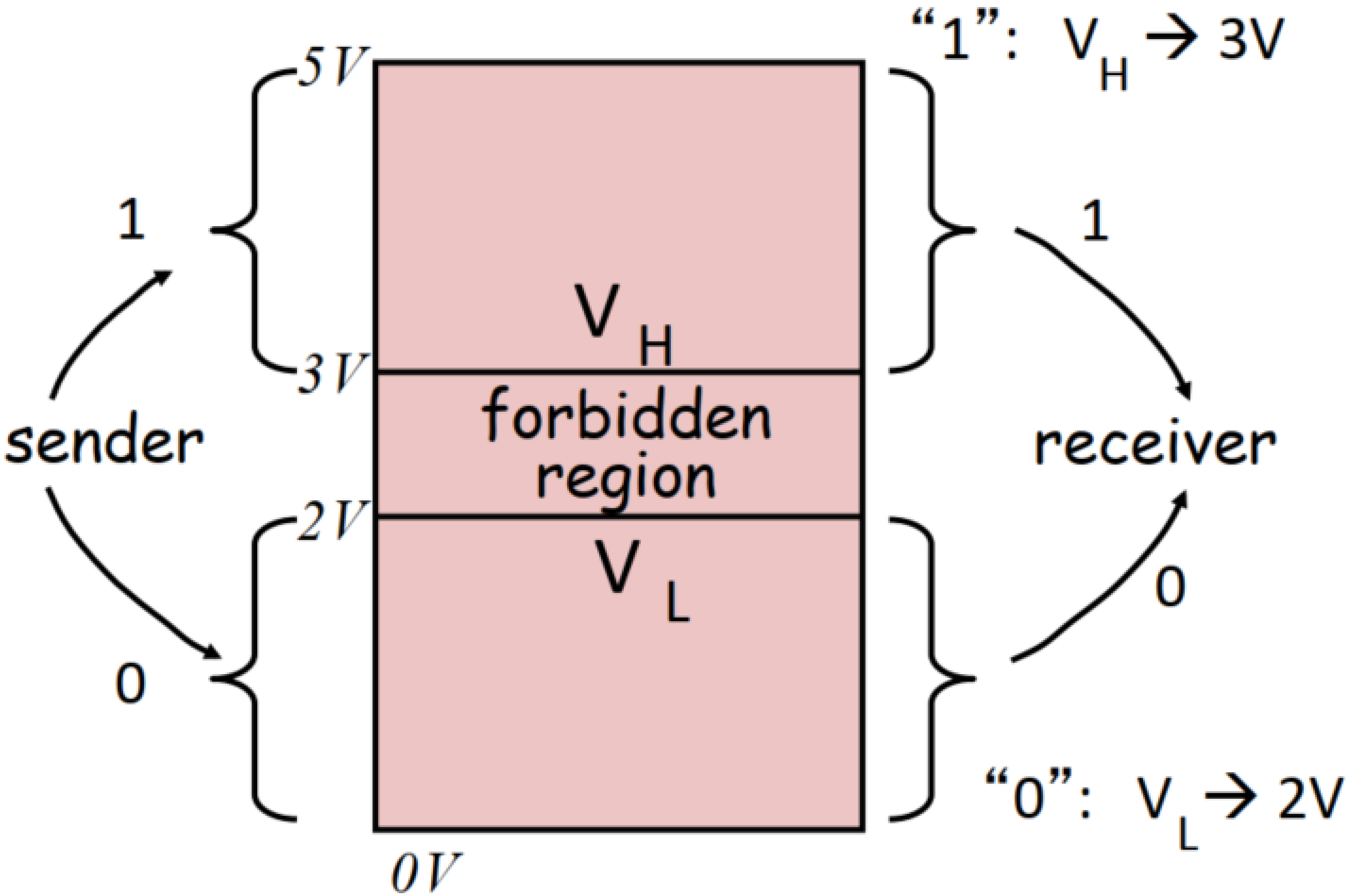
Voltage Thresholds and Logic Values



But, but, but … What about 2.5V?

Static Discipline (Cont.)

"No Man's Land" or Forbidden Region



Where's the noise margin?

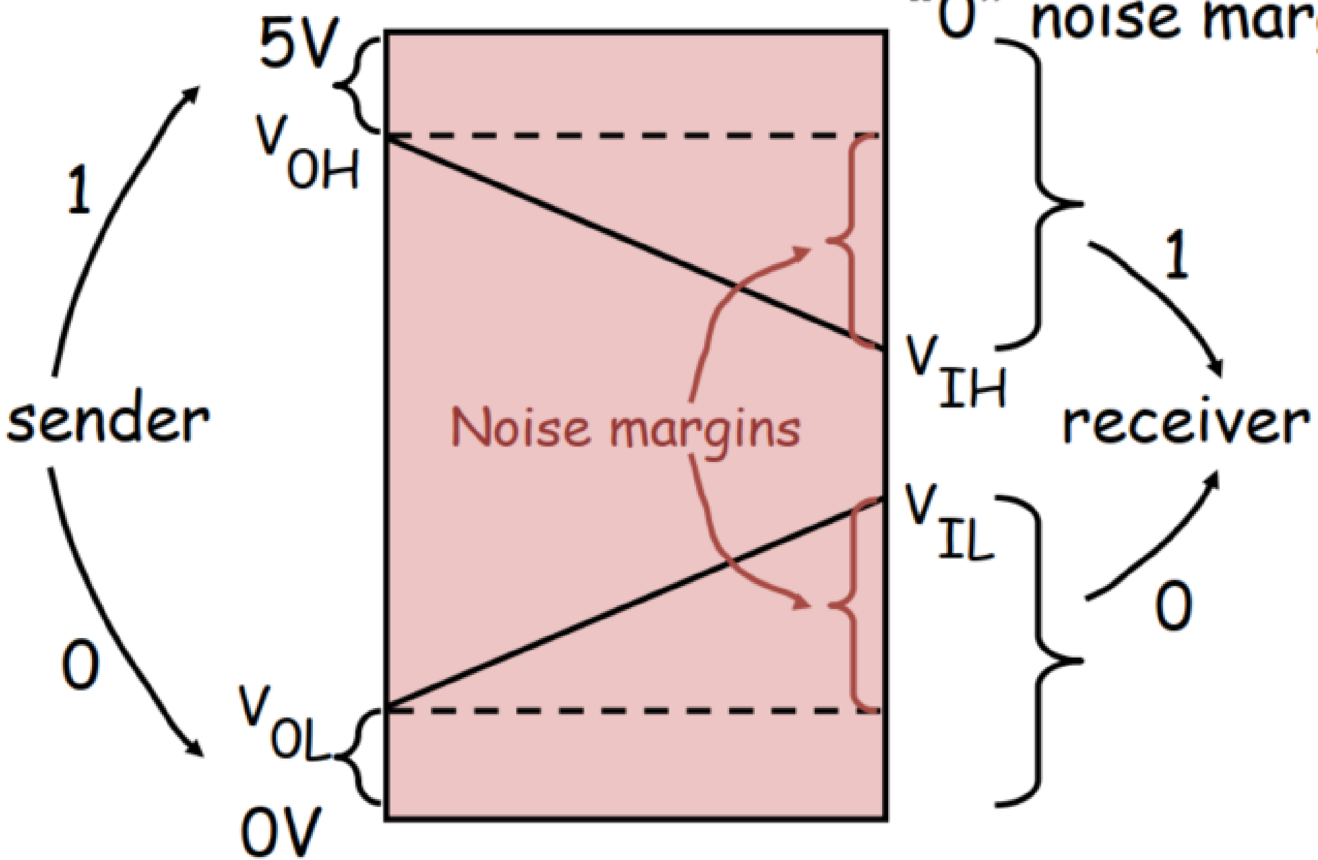
What if the sender sent 1: V_H

Static Discipline (Cont.)

Noise Margins

"1" noise margin: $V_{OH} - V_{IH}$

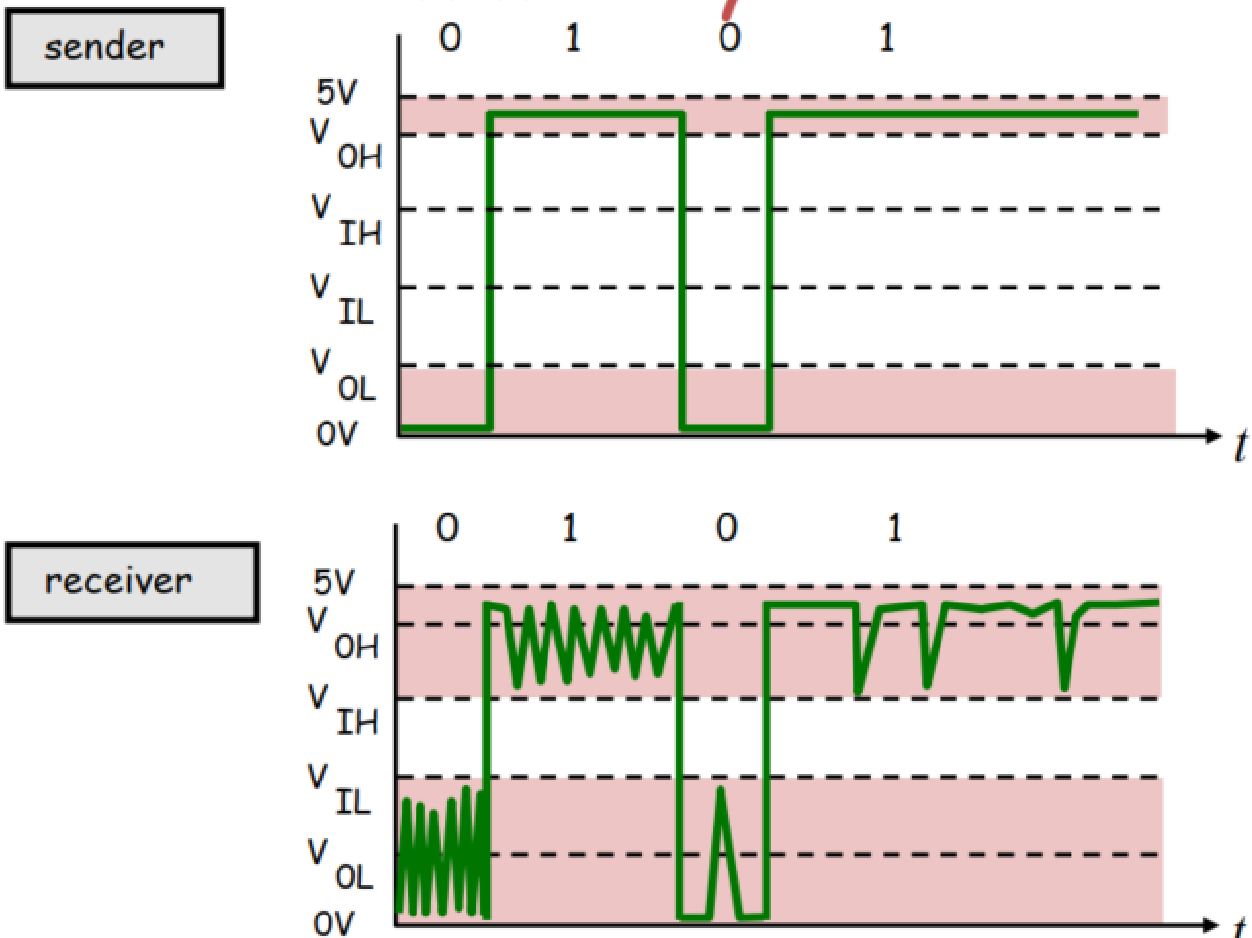
"0" noise margin: $V_{IL} - V_{OL}$



Together, the V_{OH}, V_{IH}, V_{OL}, V_{IL} thresholds define a discipline or standard that digital devices follow so they can talk to each other

Static Discipline (Cont.)

Noise Immunity



Digital systems follow static discipline: if inputs to the digital system meet valid input thresholds, then the system guarantees its outputs will meet valid output thresholds.

Digital Information

Processing Digital Signals

Recall, we have only two values —

1,0 ⇨ Map naturally to logic: T, F

⇨ Can also represent numbers

Information can be stored by using 1 and 0s.
Boolean Logic is used to process this information.

What is Information?

Information, *n.* Data communicated or received that resolves uncertainty about a particular fact or circumstance.

Example: you receive some data about a card drawn at random from a 52-card deck. Which of the following data conveys the most information? The least?

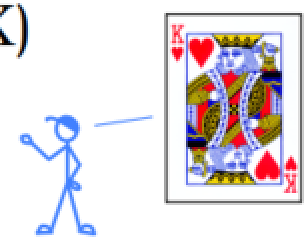
\# of possibilities remaining

13 A. The card is a heart

51 B. The card is not the Ace of spades

12 C. The card is a face card (J, Q, K)

1 D. The card is the "suicide king"



Which of the following data conveys the *most* information about a playing card chosen randomly from a 52-card deck?

Which of the following data conveys the *least* information about a playing card chosen randomly from a 52-card deck?

What is Information? (Cont.)

Quantifying Information
(Claude Shannon, 1948)

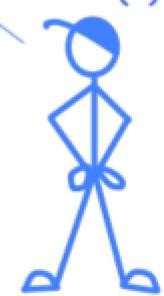
Given discrete random variable X

- N possible values: $x_1, x_2, \ldots, x_N$
- Associated probabilities: $p_1, p_2, \ldots, p_N$

Information received when learning that choice was x_i:

$$I(x_i) = \log_2\left(\frac{1}{p_i}\right)$$

Information is measured in bits (binary digits) = number of 0/1's required to encode choice(s)



$1/p_i$ is proportional to the uncertainty of choice x_i.

What is Information? (Cont.)

Information Conveyed by Data

Even when data doesn't resolve all the uncertainty

$$I(\text{data}) = \log_2\left(\frac{1}{p_{\text{data}}}\right) \qquad \text{e.g., } I(\text{heart}) = \log_2\left(\frac{1}{13/52}\right) = 2 \text{ bits}$$

Common case: Suppose you're faced with N equally probable choices, and you receive data that narrows it down to M choices. The probability that data would be sent is M·(1/N) so the amount of information you have received is

$$I(\text{data}) = \log_2\left(\frac{1}{M\cdot(1/N)}\right) = \log_2\left(\frac{N}{M}\right) \text{ bits}$$

What is Information? (Cont.)

Example: Information Content

Examples:

- information in one coin flip:

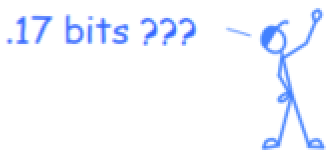
  N= 2 M= 1 Info content= $\log_2(2/1)$ = 1 bit

- card drawn from fresh deck is a heart:

  N= 52 M= 13 Info content= $\log_2(52/13)$ = 2 bits

- roll of 2 dice:

  N= 36 M= 1 Info content= $\log_2(36/1)$ = 5.17

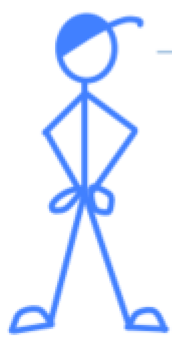


What is Information? (Cont.)

Probability & Information Content

Information content

data	p_{data}	$\log_2(1/p_{\text{data}})$
a heart	13/52	2 bits
not the Ace of spades	51/52	0.028 bits
a face card (J, Q, K)	12/52	2.115 bits
the "suicide king"	1/52	5.7 bits



Shannon's definition for information content lines up nicely with my intuition: I get more information when the data resolves more uncertainty about the randomly selected card.

Example 1

A) You're given a standard deck of 52 playing cards that you start to turn face up, card by card. So far as you know, they're in completely random order.

- How many new bits of information do you get when the first card is flipped over and you learn exactly which card it is?

  Information (in bits):

- The fifth card?

  Information (in bits):

- The last card?

  Information (in bits):

B) Z is an unknown N-bit binary number (N > 3). You are told that the first three bits of Z are 011. How many bits of information about Z have you been given?

Information (in bits):

Example 1 (Cont.)

A) You're given a standard deck of 52 playing cards that you start to turn face up, card by card. So far as you know, they're in completely random order.

- How many new bits of information do you get when the first card is flipped over and you learn exactly which card it is?

Information (in bits): 5.7

- The fifth card?

Information (in bits): 5.585

- The last card?

Information (in bits): 0

B) Z is an unknown N-bit binary number ($N > 3$). You are told that the first three bits of Z are 011. How many bits of information about Z have you been given?

Information (in bits): 3

What is Entropy?

Entropy

In information theory, the entropy H(X) is the average amount of information contained in each piece of data received about the value of X:

$$H(X) = E(I(X)) = \sum_{i=1}^{N} p_i \cdot \log_2\left(\frac{1}{p_i}\right)$$

<u>Example</u>: X={A, B, C, D}

$choice_i$	p_i	$log_2(1/p_i)$
"A"	1/3	1.58 bits
"B"	1/2	1 bit
"C"	1/12	3.58 bits
"D"	1/12	3.58 bits

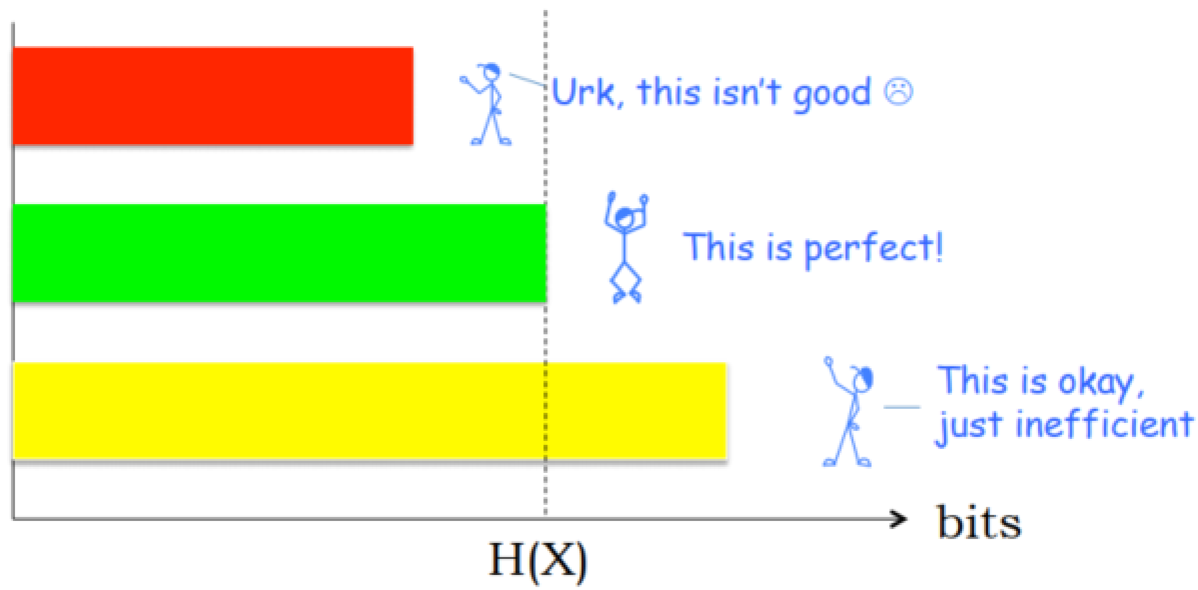
H(X) = (1/3)(1.58) +
(1/2)(1) +
2(1/12)(3.58)
= 1.626 bits

What is Entropy? (Cont.)

Meaning of Entropy

Suppose we have a data sequence describing the values of the random variable X.

Average number of bits used to transmit choice



Example 2

Please compute the entropy associated with each the following random variables.

A) The flip of an unfair coin, where $p(\text{heads}) = 0.999$ and $p(\text{tails}) = 0.001$.

Entropy (in bits):

B) The random choice of one of the 16 hex digits, where the probability of choosing any particular digit is $1/16$.

Entropy (in bits):

C) The quiz grade of a randomly-chosen student, where in the 100-student class the grade distribution was 27 A's, 38 B's, 23 C's, 8 D's, and 4 F's.

Entropy (in bits):

Example 2 (Cont.)

Please compute the entropy associated with each the following random variables.

A) The flip of an unfair coin, where $p(\text{heads}) = 0.999$ and $p(\text{tails}) = 0.001$.

Entropy (in bits): 0.0114

B) The random choice of one of the 16 hex digits, where the probability of choosing any particular digit is $1/16$.

Entropy (in bits): 4

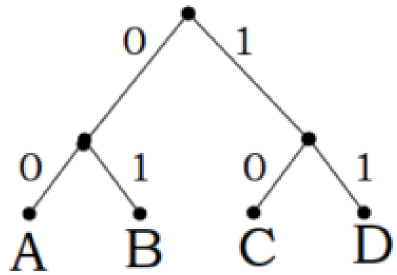
C) The quiz grade of a randomly-chosen student, where in the 100-student class the grade distribution was 27 A's, 38 B's, 23 C's, 8 D's, and 4 F's.

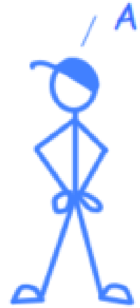
Entropy (in bits): 2.0054

Encoding

Fixed-length Encodings

If all choices are equally likely (or we have no reason to expect otherwise), then a fixed-length code is often used. Such a code will use at least enough bits to represent the information content.





All leaves have the same depth!

Note that the entropy for N equally-probable symbols is

$$\sum_{i=1}^{N} \left(\tfrac{1}{N}\right) \log_2 \left(\frac{1}{\tfrac{1}{N}} \right) = \log_2(N)$$

Examples:

Fixed-length are often a little inefficient…

- 4-bit binary-coded decimal (BCD) digits $\log_2(10) = 3.322$
- 7-bit ASCII for printing characters $\log_2(94) = 6.555$

Encoding (Cont.)

Encoding Positive Integers

It is straightforward to encode positive integers as a sequence of bits. Each bit is assigned a weight. Ordered from right to left, these weights are increasing powers of 2. The value of an N-bit number encoded in this fashion is given by the following formula:

$$v = \sum_{i=0}^{N-1} 2^i b_i$$

2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	1	1	1	1	0	1	0	0	0	0

$V = 0*2^{11} + 1*2^{10} + 1*2^9 + \ldots$

$= 1024 + 512 + 256 + 128 + 64 + 16$

$= 2000$

Smallest number: 0

Largest number: $2^N - 1$

Encoding (Cont.)

Hexademical Notation

Long strings of binary digits are tedious and error-prone to transcribe, so we usually use a higher-radix notation, choosing the radix so that it's simple to recover the original bits string.

A popular choice is transcribe numbers in base-16, called hexadecimal, where each group of 4 adjacent bits are representated as a single hexadecimal digit.

Hexadecimal - base 16

0000	-	0	1000	-	8
0001	-	1	1001	-	9
0010	-	2	1010	-	A
0011	-	3	1011	-	B
0100	-	4	1100	-	C
0101	-	5	1101	-	D
0110	-	6	1110	-	E
0111	-	7	1111	-	F

2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	1	1	1	1	0	1	0	0	0	0
7				D				0			

0b011111010000 = 0x7D0

Example 3

For the following problems, your answer should be specified as an integer.

How many bits are needed to encode the 10 decimal digits {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}?

How many bits are needed to encode the 86 ASCII characters?

Answer : 4 and 7

Encoding (Cont.)

Encoding Signed Integers

We use a signed magnitude representation for decimal numbers, encoding the sign of the number (using "+" and "-") separately from its magnitude (using decimal digits).

We could adopt that approach for binary representations:

N bits

1	1	1	1	1	1	0	1	0	0	0	0

"0" for "+"
"1" for "-"

Range: $-(2^{N-1} - 1)$ to $2^{N-1} - 1$

-2000

But: two representations for 0 (+0, -0) and we'd need different circuitry for addition and subtraction

Two’s Complement Encoding

Two’s Complement Encoding

In a two’s complement encoding, the high-order bit of the N-bit representation has negative weight:

N bits

-2^{N-1}	2^{N-2}	• • •	• • •	• • •	2^3	2^2	2^1	2^0

“sign bit”

Range: -2^{N-1} to $2^{N-1} - 1$

- Negative numbers have “1” in the high-order bit
- Most negative number: 10…0000 -2^{N-1}
- Most positive number: 01…1111 $+2^{N-1} - 1$
- If all bits are 1: 11…1111 -1
- If all bits are 0: 00…0000 0

Sum and Difference of Binary Numbers

Sum of 2 bits:

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = **10** (current step bit: 0,
carry to the next step: 1)

Difference of 2 bits:

0 - 0 = 0

1 - 0 = 1

1 - 1 = 0

0 - 1 = **11** (current step bit: 1
borrow to the next step: 1)

Sum example:

```
      1 1 1      → carry
  1 0 1 0 0 1 0 1  = A
+   1 0 1 0 1 1 1  = B
-------------------
  1 1 1 1 1 1 0 0
```

Difference example:

```
  1 0 1 0 0 1 0 1  = A
-   1 0 1 0 1 1 1  = B
  1 0 1 1 1 1        → borrow
-------------------
  0 1 0 0 1 1 1 0
```

Example 4

Exercise

$$\begin{array}{rrl} & 10011011 & = A \\ + & 1010011 & = B \\ \hline \end{array}$$

$$\begin{array}{rrl} & 10011001 & = A \\ - & 1010011 & = B \\ \hline \end{array}$$

Exercise (solution)

$$\begin{array}{rrl} & \boxed{0\,0\,1\,0\,0\,1\,1} & \rightarrow \text{carry} \\ & 10011011 & = A \\ + & 1010011 & = B \\ \hline & 11101110 & \end{array}$$

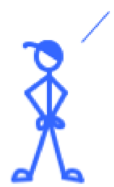
$$\begin{array}{rrl} & 10011001 & = A \\ - & 1010011 & = B \\ & \boxed{1\quad\quad\quad 1\;\; 1} & \rightarrow \text{borrow} \\ \hline & 01000110 & \end{array}$$

Two’s Complement Encoding (Cont.)

More Two’s Complement

- Let’s see what happens when we add the N-bit values for -1 and 1, keeping an N-bit answer:

$$\begin{array}{r} 11\ldots1111 \\ +\underline{00\ldots0001} \\ 0000000 \end{array}$$



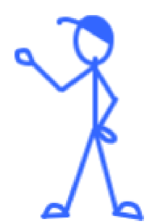
Just use ordinary binary addition, even when one or both of the operands are negative. 2’s complement is perfect for N-bit arithmetic!

- To compute B-A, we’ll just use addition and compute B+(-A). But how do we figure out the representation for -A?

$$A+(-A) = 0 = 1 + -1$$

$$-A = (-1 - A) + 1$$
$$= \sim A + 1$$

$$\begin{array}{r} 1 \\ \underline{-A_i} \\ \sim A_i \end{array}$$



To negate a two’s complement value: bitwise complement and add 1.

Example 5

Convert the following decimal numbers to 6 bit 2's complement representation binary numbers. Provide the binary numbers using the format 0bXXXXXX.

15 = 0b

-15 = 0b

6 = 0b

-6 = 0b

21 = 0b

-21 = 0b

Example 5 (Cont.)

Convert the following decimal numbers to 6 bit 2's complement representation binary numbers. Provide the binary numbers using the format 0bXXXXXX.

15 = 0b 001111

-15 = 0b 110001

6 = 0b 000110

-6 = 0b 111010

21 = 0b 010101

-21 = 0b 101011