CEG 3320 - Digital System Design

Instructor: Travis Doom, Ph.D.
331 Russ Engineering Center
775-5105
travis.doom@wright.edu
http://www.wright.edu/~travis.doom

Lecture slides created by T. Doom for Wright State University's course in Digital System Design. Some slides contain fair use images or material used with permission from textbooks and slides by R. Haggard, F. Vahid, Y. Patt, J. Wakerly, M. Mano, and other sources.

Module I: Combinational system representation, analysis, and design

Logic Devices

Analog characteristics of digital logic devices

Combinational analysis

Encoding information in binary

Basic combinational design

Introduction to Logic devices

Standard Logic gates

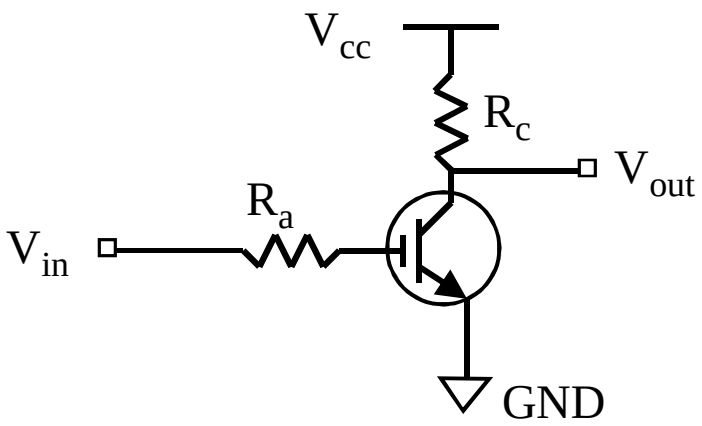
Truth tables

Boolean equations

The Transistor

Transistor is an electronic switch, it is a semiconductor that flips on when supplied with electrons

Logic Gates: An ideal abstraction



Schematic

Equation

$Z = A' = \bar{A}$

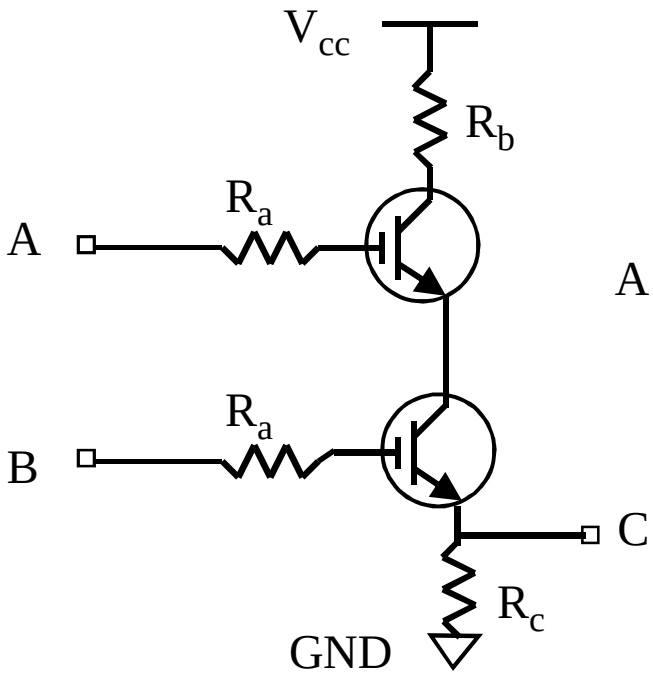
Active high convention:
ON = True = logic "1" (current flows)

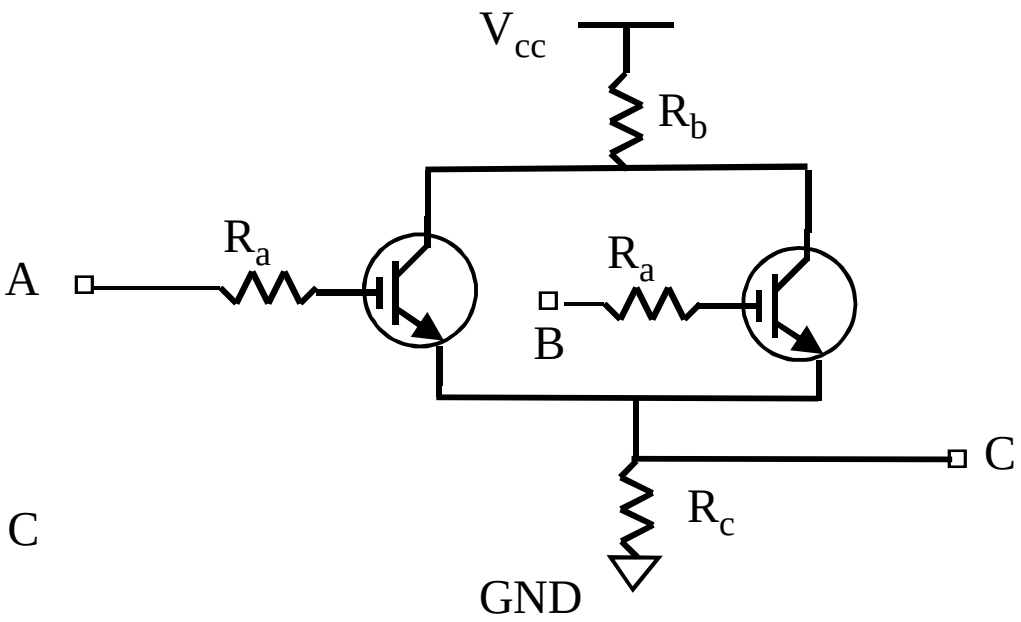
Current Input Vin	Current Output Vout
OFF	ON
ON	OFF

Truth Table

Current Input A	Current Output Z
0	1
1	0

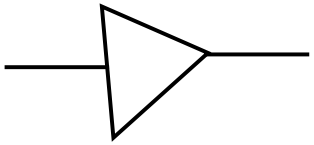
‘Simple’ example non-ideal logic devices





What are the likely intended functions of these devices?

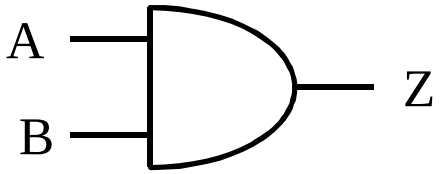
Disclaimer: These simplistic implementations are for introductory illustration only.



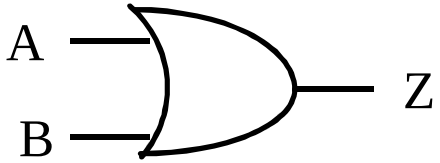
this is a buffer, works as a repeater

Basic logic circuits

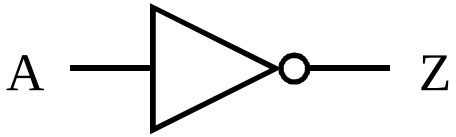
- AND gate
  - Output Z = 1 only when inputs A **and** B are both 1
  - Z = A·B
  - Z = AB

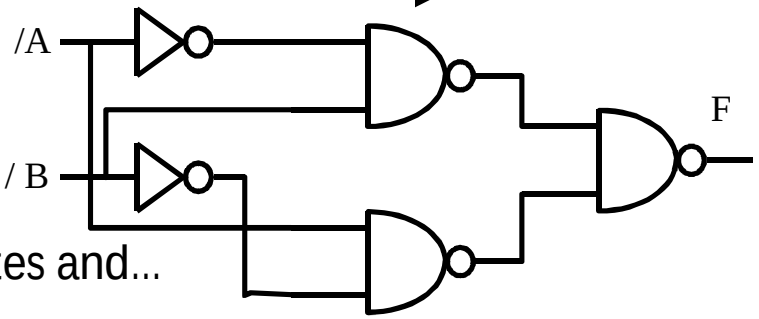


- OR gate
  - Output Z = 1 only when inputs A **or** B **or** both are 1
  - Z = A + B



- NOT gate or inverter
  - Output Z = 1 only when input A is 0
  - $Z = \overline{A}$
  - Z = A'
  - Z = ¬A

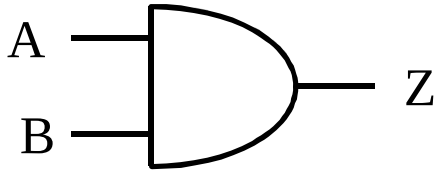




- Simple alone, but combine a few million gates and…

Representing ideal behavior: Truth tables

Current Inputs		Current Outputs
A	B	Z
F	F	F
F	T	F
T	F	F
T	T	T



Z = AB

Current Inputs		Current Outputs
A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

Active high convention:
Current flows = ON = True = logic "1"

Exercise: Draw truth table for Z = A + B

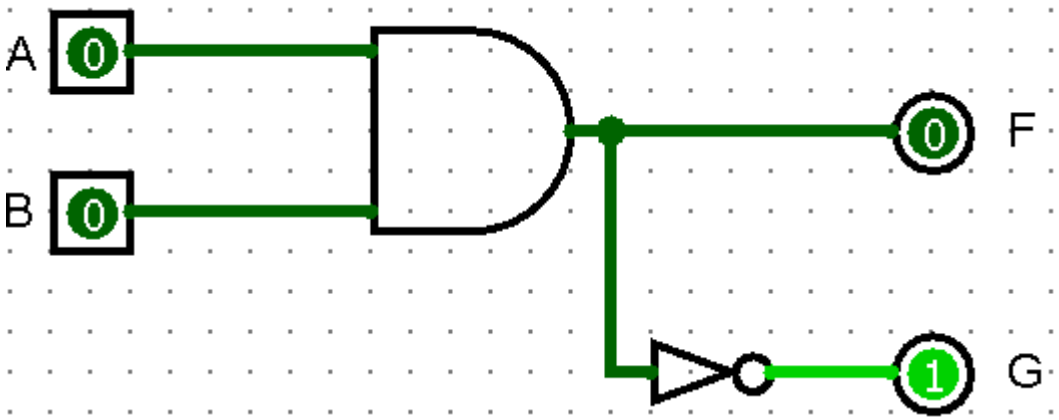
Data representation in binary

- We need a way to represent information (data) in a form that is mutually comprehensible by human and machine.
  - We have to develop schemes for representing all conceivable types of information – integers, characters, floating point numbers, language, images, actions, etc.
- Binary digITs (Bits): sequences of 0's and 1's that help humans keep track of the current flows stored/processed in the computer
  - Using base-2 provide us with two symbols to work with: we can call them *on* & *off*, or (more usefully) *0* and *1*.
  - We group bits together to allow representation of more complex information
  - For ease of use, Computer scientists usually represent **quartets of bits** in **Hexadecimal**
  - eg. xA13F vs. 1010000100111111

Hex	Binary
x0	0000
x1	0001
x2	0010
x3	0011
x4	0100
x5	0101
x6	0110
x7	0111
x8	1000
x9	1001
xA	1010
xB	1011
xC	1100
xD	1101
xE	1110
xF	1111

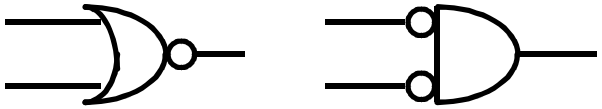
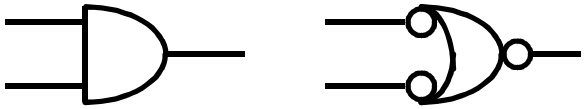
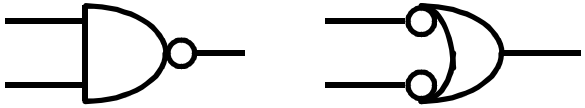
Memorize This!

Example: Truth Table Analysis



- What is the truth table of this circuit?
  - Attempt in Boolean, binary, and hex

Common Combinational SSI Logic Devices (Gates)

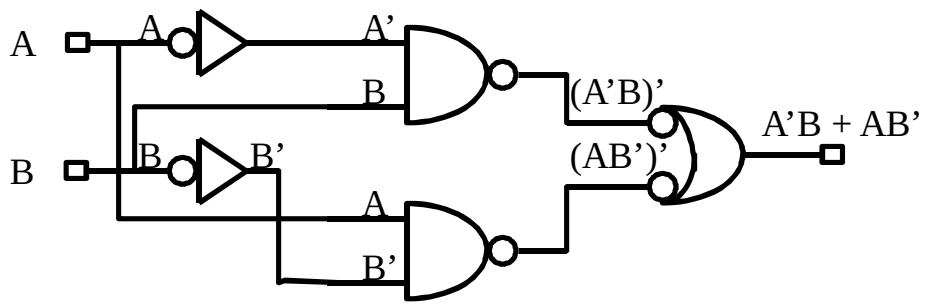
- OR  Z = A + B
- NOR  Z = (A + B)’
- AND  Z = AB
- NAND  Z = (AB)’
- INVERTER  Z = Z’
- BUFFER  Z = Z

Bubbles mean “not”

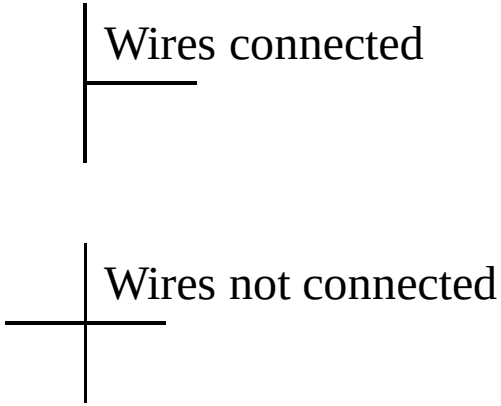
Each gate has a truth table and can be represented algebraically

Exercise: show that the equivalent gates do the same function

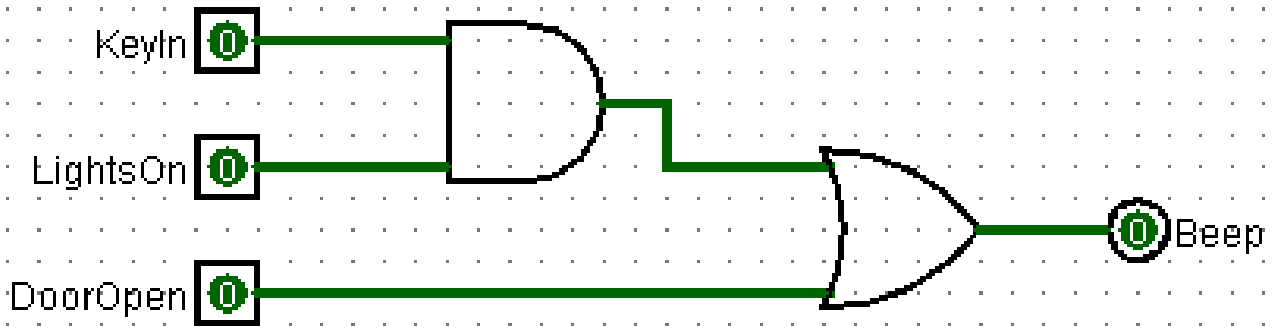
Logic Diagram documentation standards



- Good Design Practices include:
  - Use standard device symbols
  - All wires are either horizontal or vertical
  - All wires are labeled with expression
  - Wires are connected at "T" intersections
  - Wires are not connected at "X" intersections



Example: Basic combinational analysis



- What is the truth table for Beep?
- What is the equation for Beep?

Dealing with non-ideal devices

Digital and Analog characteristics

Combinational logic

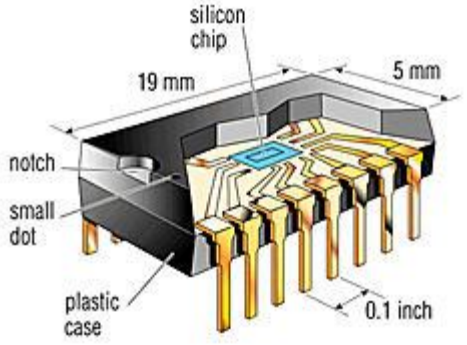
Timing diagrams

Propagation delay

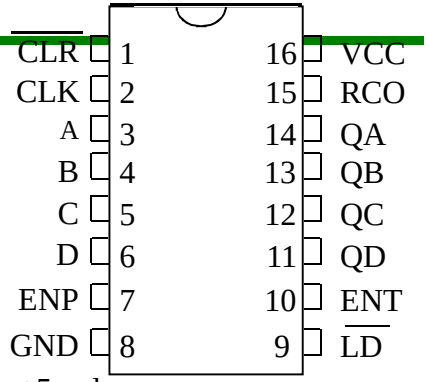
Loading

Integrated Circuits (ICs)

- Pin-out view
- Block diagram view
- Functional view
- Analog properties

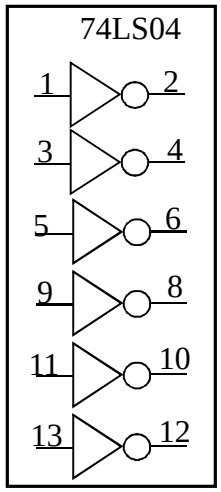


Pin-out view

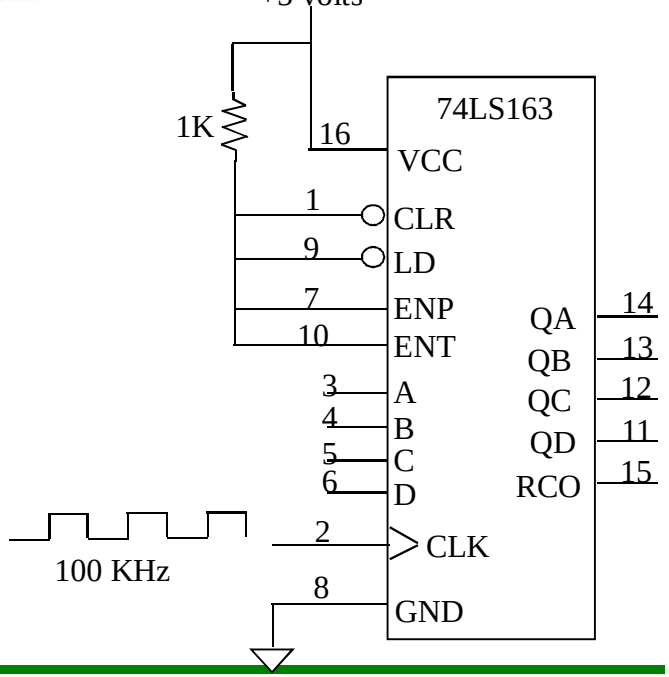


Integration Scale

- **SSI**: Small Scale Integration
  - 10s of transistors/unit
- **MSI**: Medium Scale
  - 100s of transistors
- **LSI**: Large Scale
  - 1000s of transistors
- VLSI: Very Large Scale
  - 100,000s+ transistors

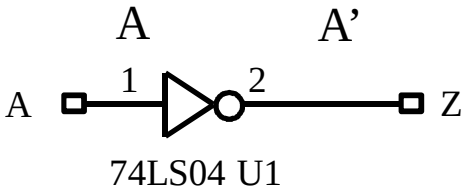


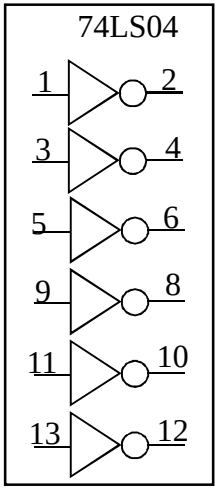
Block diagram



Functional view

Schematic Diagram





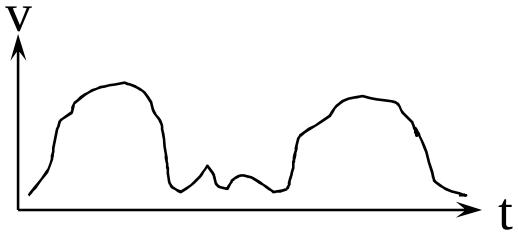
- Schematic diagrams focus on implementation
- Schematic diagrams should include
  - IC-Type - Logic Family
  - Pin numbers - Pin Diagram
  - Reference designator - Unit Number

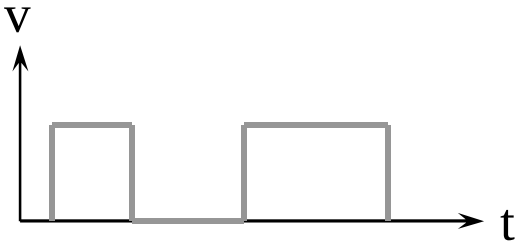
Electronic Aspects of Digital Design

- Ideal combinational logic devices
  - Digital/Discrete devices
  - Output is defined by current input
- Actual logic gates
  - Analog/Continuous devices
  - Under what conditions does the a logic gate behave like its ideal?
- Manufacturer specifications (data sheets)
  - Power-supply voltage
  - Temperature
  - Propagation delay
  - Loading

Digital Devices: ideal Vs. actual

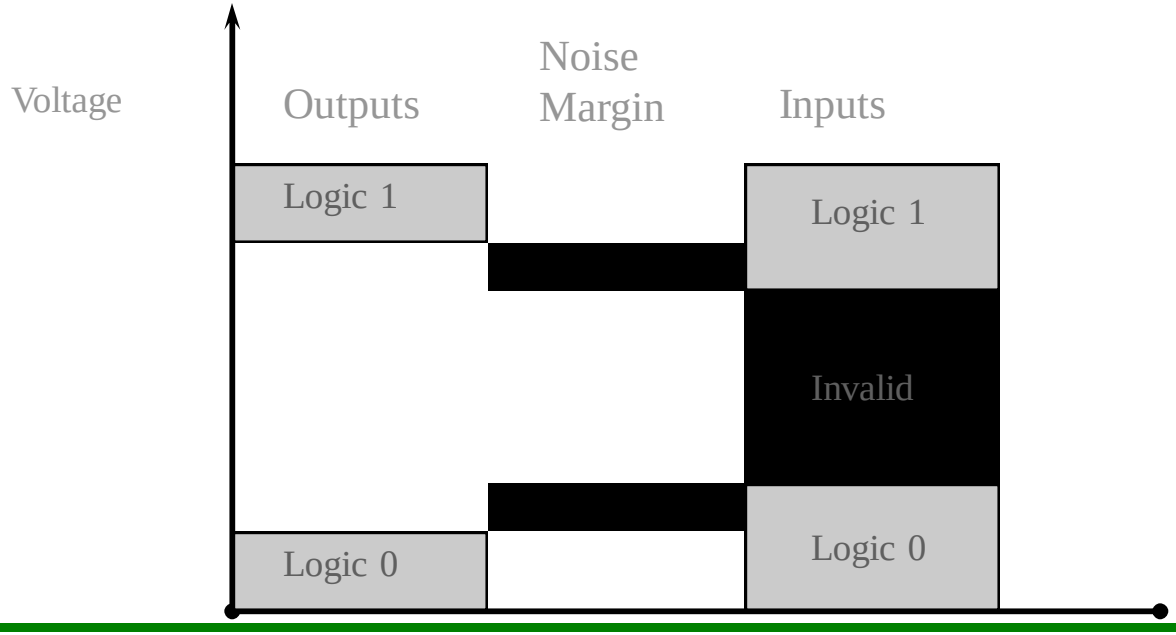
- Analog characteristics
  - Continuous signal levels
  - Very small, smooth level changes
- Digital characteristics
  - Discrete signal levels (voltage usually)
  - Two levels: on/off, high/low, True/False, 1/0
  - Disjoint or quantized level changes
- Digital Concepts and Devices
  - Digital Design also called Logic Design
  - Logic Gates - the most basic digital devices
  - Digital devices have analog electronic aspects
    - Exercise: list some of these aspects





Electronic Aspects of Digital Design

- Digital devices are built with analog components
- A range of voltages associated with each logic value (0 or 1)
- Noise margin : The difference between the range boundaries
  - in low state = V_{ILmax} - V_{OLmax} = 1.5 V - 0.1 V= 1.4 V for 5V CMOS
  - in high state = V_{OHmin} - V_{IHLmin} = 4.9 V - 3.5 V= 1.4 V for 5V CMOS



Propagation Delay

- The delay time between input transitions and the output transitions due to the propagation delay of the the logic gates / devices.
- t_p of a signal depends on the signal path inside the logic circuit
- For a logic gate t_{pLH} may not equal t_{pHL}
- t_p is specified in the manufacturer data sheets of the IC's
- Example :
  -The time delay for 74x00 in nanoseconds for three logic Families:

	Typical		Maximum	
	t_{pLH}	t_{pHL}	t_{pLH}	t_{pHL}
74LS00	9	10	15	15
74HCT00	11	11	35	35
74ACT00	5.5	4.0	9.5	8.0

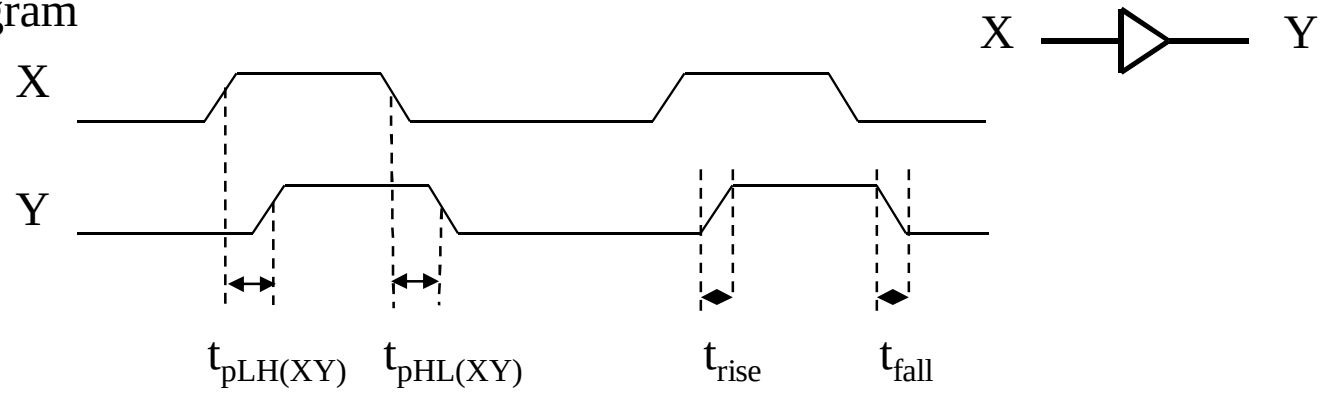
- To find t_p for a signal, add the propagation delays of all gates along the path of the signal

Units of measure

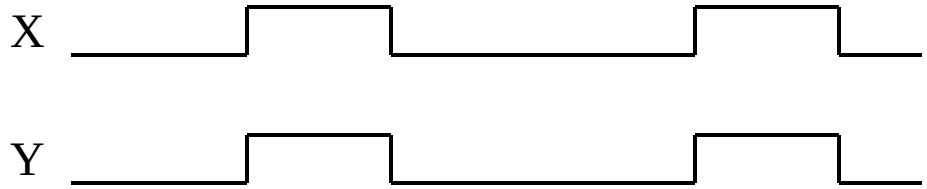
UNIT	Abbreviation	Seconds	Scientific Notation
Second	s	1	$1x10^{0}$
Millisecond	ms	1,000	$1x10^{-3}$
Microsecond	µs	1,000,000	$1x10^{-6}$
Nanosecond	ns	1,000,000,000	$1x10^{-9}$
Picosecond	ps	1,000,000,000,000	$1x10^{-12}$

Timing Diagrams

Actual Timing Diagram

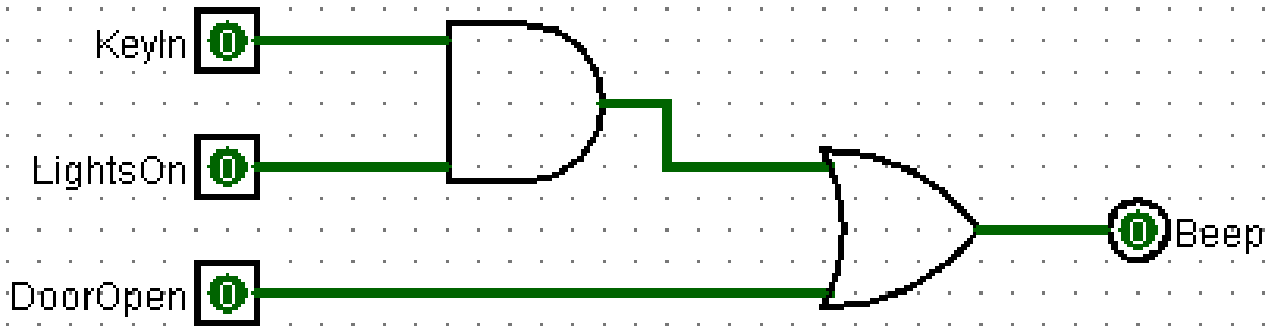


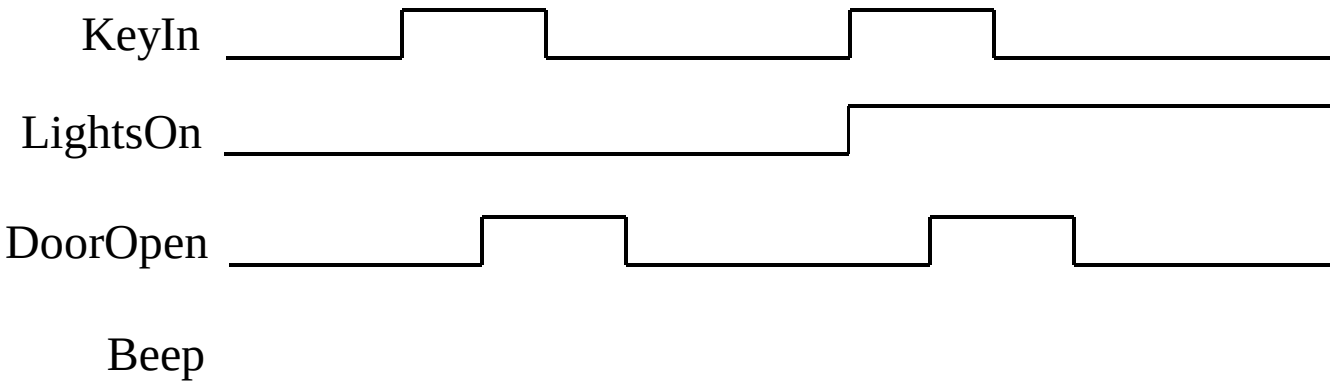
Functional Timing Diagram



- Functional timing diagrams do not show exact delays.
- "lining up" everything allows the diagram to display more clearly which functions are performed in response to which action
  - Illustrates operation, but does not specify upper and lower limits
  - Usually, (but not always) sufficient design

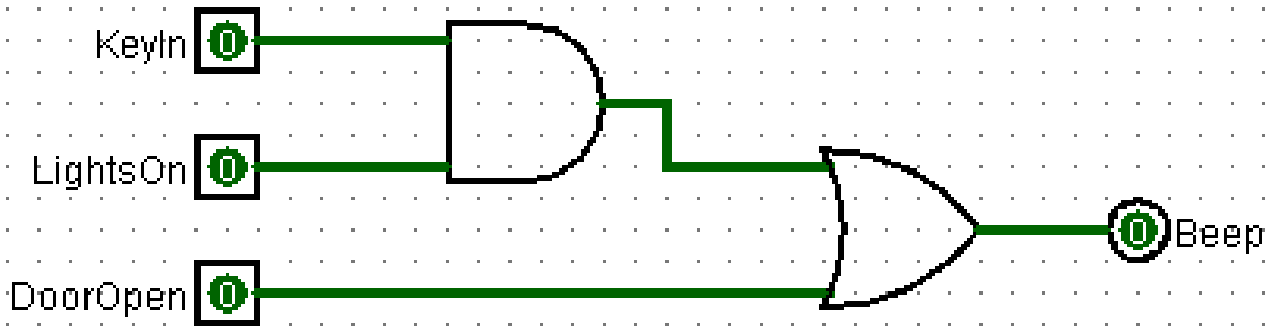
Example: Timing Diagram





- Complete the functional timing diagram for this combinational circuit:

Calculating propagation delay



AND GATE

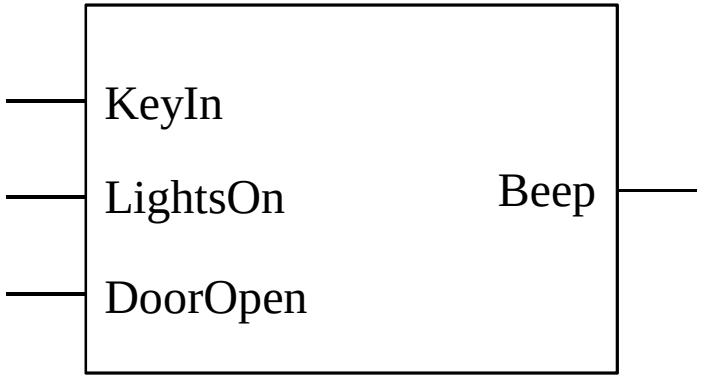
t, pd (min): 2ns

t, pd (max): 3ns

OR GATE

t, pd (min): 1 ns

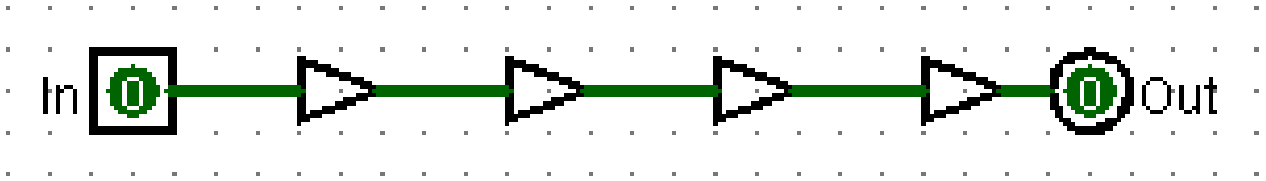
t, pd (max): 5ns

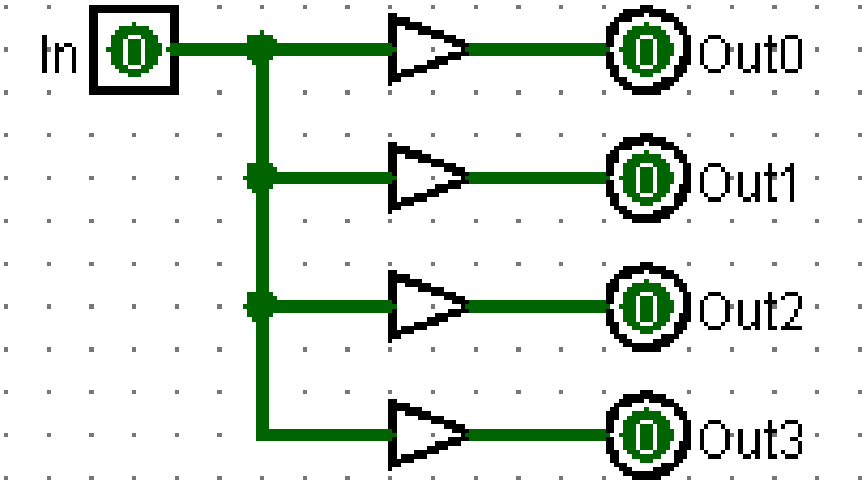


What are the timing characteristics of the overall device?

(Warning: These times are for illustration only)

Loading





Analysis and design of simple combinational circuits

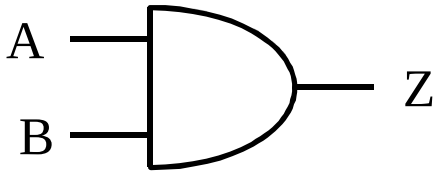
Analysis of combinational circuits

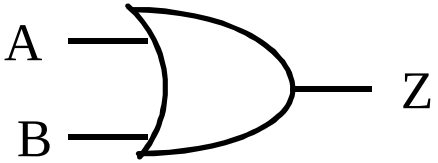
Introduction to basic combination design

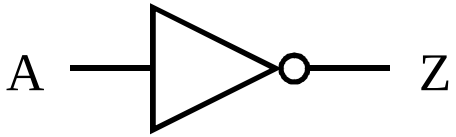
Equivalence of Truth Table, Logic Equation, and Logic diagram

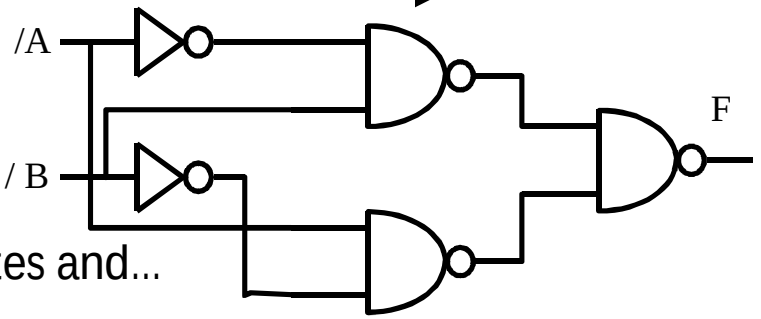
Basic logic circuits

- AND gate
  - Output Z = 1 only when inputs A **and** B are both 1
  - Z = A·B
  - Z = AB
- OR gate
  - Output Z = 1 only when inputs A **or** B **or** both are 1
  - Z = A + B
- NOT gate or inverter
  - Output Z = 1 only when input A is 0
  - $Z = \overline{A}$
  - Z = A'
  - Z = ¬A
- Simple alone, but combine a few million gates and…









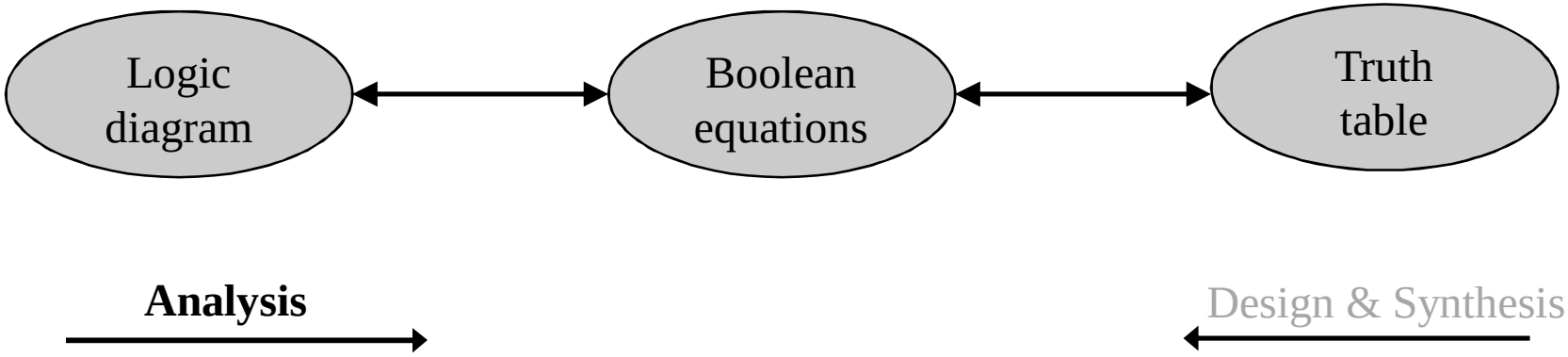
Boolean Algebra: Basic Operators

Operator	Logic Design	Alternate	Program Logic	Program Bitwise
NOT	—	‘	!	~
AND	·	^, *	&&	&
OR	+	v	||	|
XOR	⊕			^

- Order of precedence:
  - (1) NOT and Parenthesis
  - (2) AND
  - (3) OR

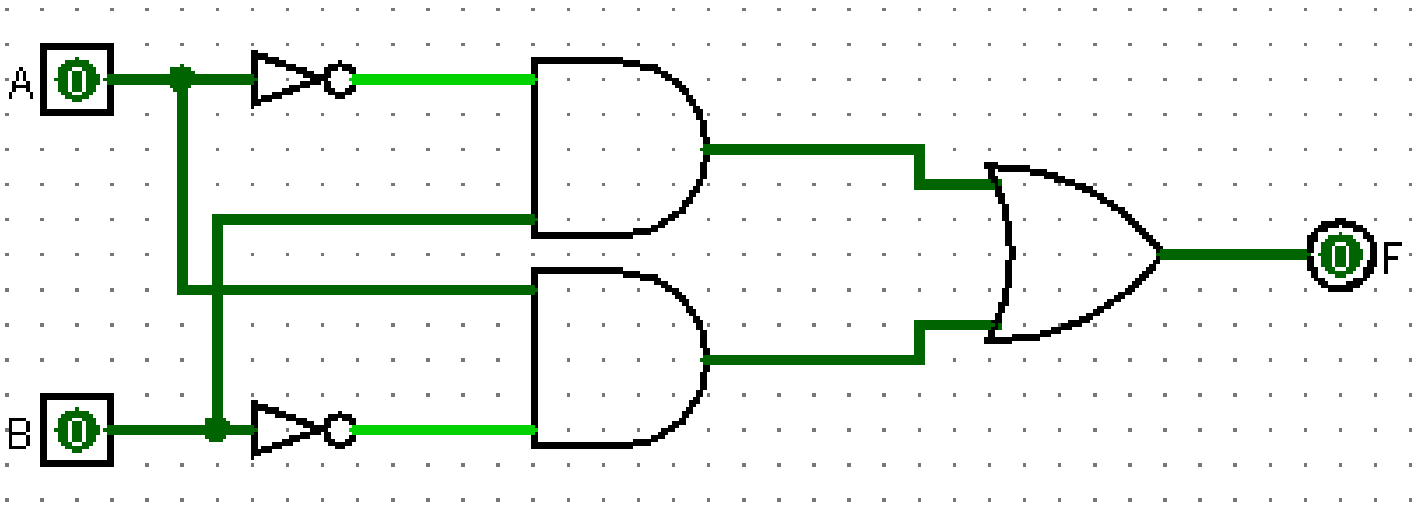
- Examples
  - Truth table for: F = A + BC
  - Truth table for : F = A + B’C

Levels of combinational design abstraction



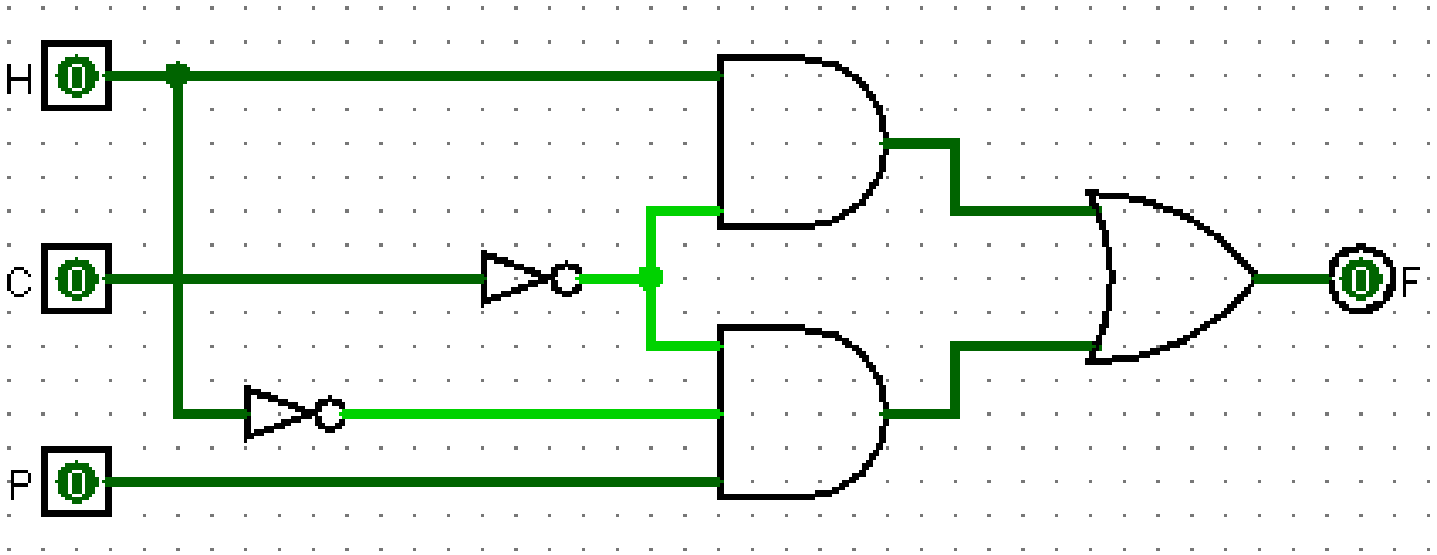
- What does a device do?
  - What function(s) does it perform?
  - What are the non-ideal characteristics of the device that must be observed?
- What does it do in its problem domain / natural-language description?

What does this device do?



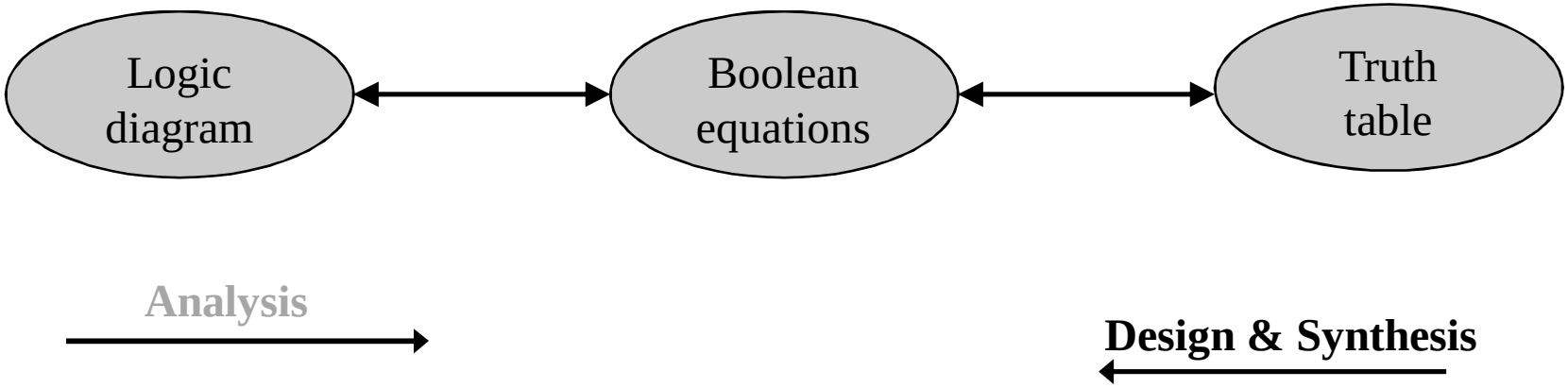
- Assume:
  - t, pd (min) = number of gate inputs (in ns)
  - t, pd (max) = 2 * t, pd(min)

What does this device do?



- Assume:
  - t, pd (min) = number of gate inputs (in ns)
  - t, pd (max) = 2 * t, pd(min)

Levels of combinational design abstraction



- (1) Describe the solution to the problem in a truth table (if necessary)
- (2) Describe the solution to the problem as (a) Boolean Equation(s)
- (3) Implement the equation as a circuit
  - Top-down implementation strategy is often best; start from the goal and decompose towards simplicity
- (4) Test, Test, Test

Basic Design: Dance Floor Controller

- Sensors:
  - N: Noise sensor, asserted if ambient noise is significantly loud
  - M: Motion sensor, asserting if dance floor is occupied
- Controls:
  - L: Turns on lighting effects when asserted
  - V: Sets music volume to moderate when not asserted. Sets music volume to quite loud indeed when asserted.
- Problem:
  - Lighting effects should be in play whenever the dance floor is occupied. Music volume should be raised if the dance floor is noisy whenever people are dancing.
- Create a logic circuit that realizes a solution to this problem

Basic Design: Selector

- Inputs(3): a, b, and select
- Output(1): mux

- Problem:
  - Based upon the value of its select input, this device should output the current value of a (if select is not asserted) or the value of b (if select is asserted).

Basic Design: Number of 1s counter

- Inputs (3): a, b, c
- Outputs (2): z1, z0

Number	Encoding
0	00
1	01
2	10
3	11

- This device must output, as a 2-bit unsigned magnitude binary number (z1z0), the number of its inputs that are currently asserted.