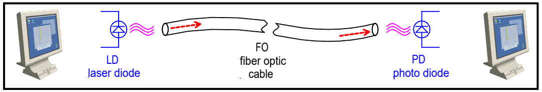
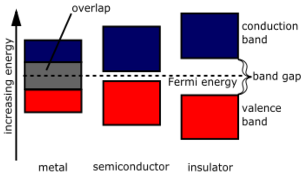
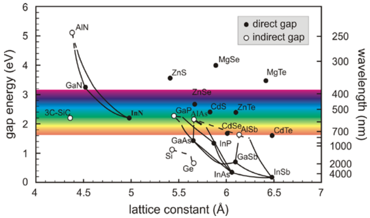
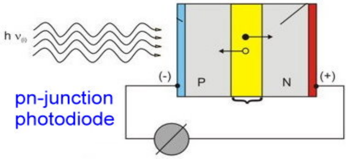
Review Optoelectronics

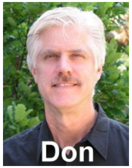
- ***Optoelectronics for Communications***
  High speed, low loss
- ***Spectral Response of Semiconductors***
  Energy Gap *- E_G*
  Energy Bands *– variety of energy states*
- ***Light Emitters***
  LED, Laser Diode, Ruby Laser
  *Change **color** by changing E_G*
  Change $\mathbf{E_G}$ by alloying different materials
- ***Light Detectors – Photovoltaics*** *Solar cells*
  *Photon energy must exceed the **band gap***
  *The **Electric Field** in the pn-junction*
  ***separates** electrons from the holes*











Calendar of Topics Covered

Physics PHYS 2371/2372, Electronics for Scientists

Don Heiman and Hari Kumarakuru
Northeastern University, Fall 2020

Also see **Course Description** and **Syllabus**

This is a schedule of the topics covered, but it may be modified occasionally (10/22/2020).

Week #	Lectures	Weekly Topics (Chs.)	Homework (Ch-Problem)	Lab Experiments (always look for latest version)
VIII Oct 28-30	**Wed Lecture** Optoelectronics Optoele Lecture	*Photodiode, LED, laser*	none	***Lab-7, Optoelectronics*** (coupled LED-photodiode) Lab-7 Optoele video
IX Nov 2, 4-6 **MON/WED**	**Mon/Wed Lectures** MON Digital-1 Digital-1 Lecture WED Digital-2 Digital-2 Lecture	Digital Logic (Ch-19,22), Binary Numbers (Ch-54) Logical Networks (Ch-20)	19-all, 20-all	***Lab-8a, Digital Circuits*** (truth table, 4-bit decoder) Lab-8a Digital video
X Nov 11-13	**Wed Lecture** Pulsed ICs Pulsed Lecture	Lecture: Pulsed ICs Digital Summary	21-1/2	***Lab-8b, Pulsed Digital*** (Flip-flops, counter, displays) Lab-8b Pulsed video
XI Nov 18-20 **WED EXAM**	**EXAM-II - Wed** Final Project	**EXAM-II: Magnetoelectronics, Optoelectronics, Digital/Pulsed**		***Final Project***
XII Nov 25-27	No Lecture	**Thanksgiving**		No Lab
XIII Dec 2	**Wed Lecture**	Future Electronics		***Project PowerPoint*** **due Monday Dec 2** (EG361 or email file)
XIV Dec 7-9	**No Classes**			

Lecture: Wednesday

Lab: Thursday/Friday

Basic Digital Electronics

Digital Basics

- **Computers**
- **Boolean Algebra**, Ch-19
  - truth tables
  - theorems
- **Logic Gates**
  - OR, AND, NOT, NAND
- **Adders**
  - half, full, 4-bit

- **What is "digital"**
  binary – only "0" and "1"; or "F" and "T"
- **Why digital?**
- **How to do math**
  Ex. add 2 numbers
  Analog – summing amplifier
  Digital – voltage is there (1) or not there (0)
- **History of computers**
- **Logic Gates**
  AND, OR, NOT
  NOR, NAND, XOR, XNOR
- **Half-adder, Full-adder**

Computers

History of Computation

Sumerian abacus < 2500 BCE

Mechanical calculator

Pascal, 1642

(50 prototypes, +-x/)

Difference engine

Charles Babbage, 1832

(gears, tabulate polynomials)

Turing Machine (1936)

Alan Turing, ***The Imitation Game***

Hypothetical device - Applied "algorithm"

Turing Test *(1950) – Can a machine think?*

Electrical Computer

von Neumann architecture, 1945

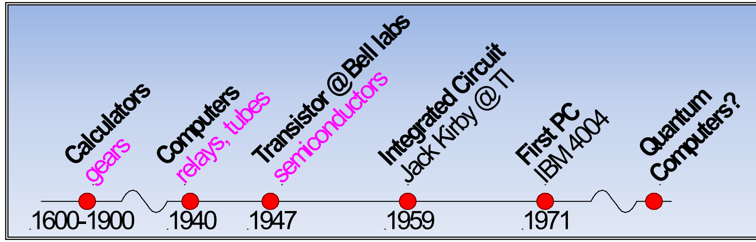
ENIAC, 1946, (relays then tubes)

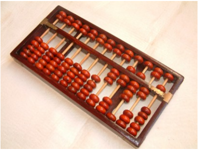
Electronic Computer

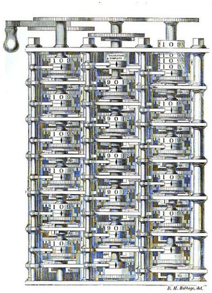
1950 → Semiconductors

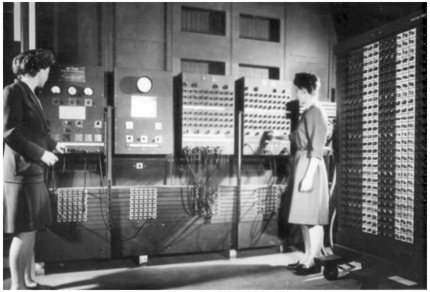
Quantum Computer

Factor/Sorting numbers, Encryption









Scott Adams/Dilbert

Digital Basics, Number systems, Ch-54

Number Systems

- decimal base-10 (figers)
  (Egypt-3100 BCE, China-1100 BCE)
- Romans numerals (I, V, X, C, M)
- Hindu-Arabic (0,1,2,3,…9)
  (1-4th C; 9th C)
- **binary** (base-2)
- octal base-8 (byte)
- hexadecimal (16)

Byte = 8 bits

0-255 (1-256)

$2^8 = 256$

$\mathbf{2^{10}} = 1024 \sim \mathbf{1000}$

Binary System

2^4	2^3	2^2	2^1	2^0
16	8	4	2	1

Ex. $15_{10} = 8 + 4 + 2 + 1$

$\mathbf{15_{10} = 0\ \ 1\ \ 1\ \ 1\ \ 1_2}$

ASCII

American Standard Code for Information Interchange

Each keyboard character is represented as one byte

Decimal	4-bit Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000

Boolean Algebra, Ch-19

Represent Boolean numbers
by **A** and **B,**

where **1** ≡ **T**(true) and **0** ≡ **F**(false)

Define the Operations
A+B ≡ A OR **B**

A·B ≡ A AND **B**

Ā ≡ NOT **A**

Truth Table

inputs		OR	AND	NOT
A	B	A+B	A·B	Ā
0	0	0	0	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	0

Theorems

A+A =	A
A·A =	A
A+1 =	1
A·1 =	A
A+0 =	A
A·0 =	0
A+Ā =	1
A·Ā =	0

Operational Properties

Commutative

A+B = B+A (+,·)

Associative

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

(A·B)·C = A·(B·C)

(A·B)+B ≠ A·(B+B)

Distributive

(A+B)·C = A·C+B·C

DeMorgan's Laws
To do all logical operations only need

- **NOT** & **OR**

or

- **NOT** & **AND**

Questions ?

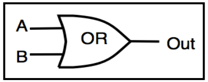
Digital Electronics -- Basic Logic Gates (37:11)

Logic Gates

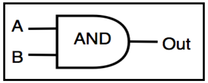
Logic Gates
have wires
with V=0 and +5 V

N=NOT
(over or underline)
Small circle

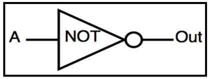
A+B
OR



A·B
AND



$\bar{A}$
NOT



Reddit.com

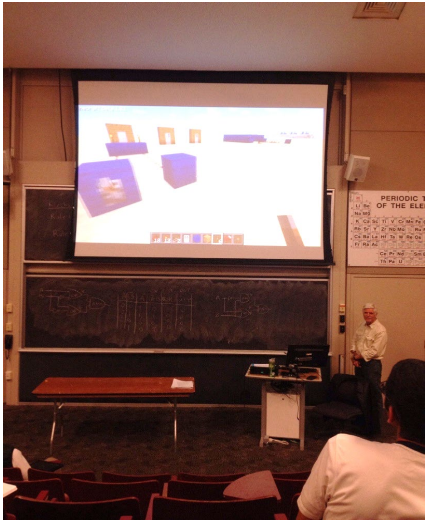
"Learning logic gates in Electronics Class"
submitted 2013 by ThatWeirdPhysicist to Minecraft

http://imgur.com/CCQiVmj - **175,746 views**!
http://imgur.com/mqiDXxu - **204,667 views**!

About Andrew
"Call yourself *Professor* from now on." "Dr. Professor M.D. P.H.D"

Andrew - "**Wow, I never thought my video would be used in a university**. This is my youtube channel. Here is the video that OP watched in class."

Minecraft Tutorial - Logic Gates



https://www.reddit.com/r/Minecraft/comments/1pk6zl/learning_logic_gates_in_electronics_class/

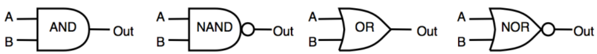
Logic Gates – Additional Operations

Additional NOT operation

- NOT → add small circle

Additional eXclusive operation

•X (exclusive) → add curved line



inputs		AND	NAND	OR	NOR
A	B	A•B	$\underline{A\cdot B}$	A+B	$\underline{A+B}$
0	0	0	1	0	1
0	1	0	1	1	0
1	0	0	1	1	0
1	1	1	0	1	0

OR	XOR	XNOR
0	0	1
1	1	0
1	1	0
1	0	1

Questions ?

INSIDE LOGIC GATES

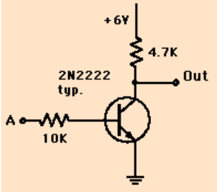
TTL *- Transistor-Transistor Logic*

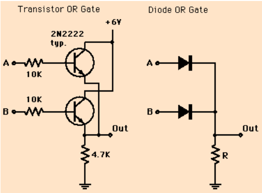
V=0 V and +5 V

TTL has been the standard logic in most applications for many years.

These comprise the 7400 family of digital ICs.

NOT gate / OR gate

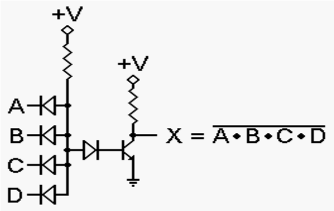




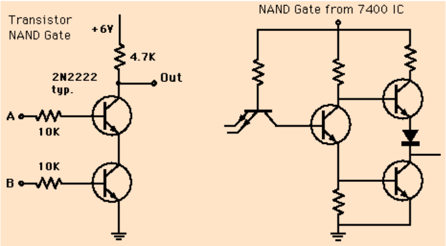
Treat transistors as on/off switches
Treat diodes as high/low resistances

Inside NAND gates

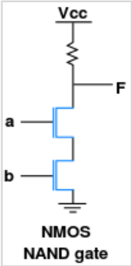
Diodes and a Transistor



Transistor | 7400



NMOS FET



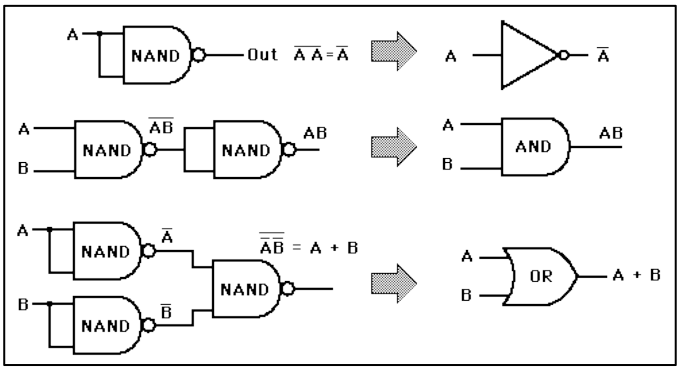
Questions ?

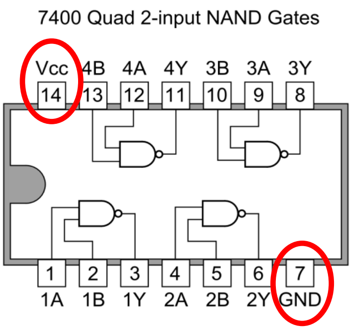
Universal Gate = NAND

Use only NANDS to create other gates

A·B

"any 0 gives a 1"





All chips need to be powered with +5 V and a ground.

Example Problems – Digital Circuit

Given a **circuit** of logic gates, construct the **Truth Table**

See slide 8

Ch-19, problem 3

Construct the truth table for the output using OR/AND/NOT gates.

What is the circuit?

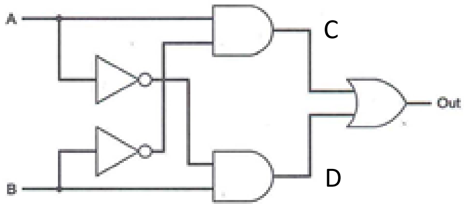
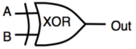


FIGURE 19-13 The circuit for Problem 19-3.

inputs A	inputs B	OR	XOR	XNOR
0	0	0	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	0	1



Example Problems – Digital Circuit

What is the circuit?

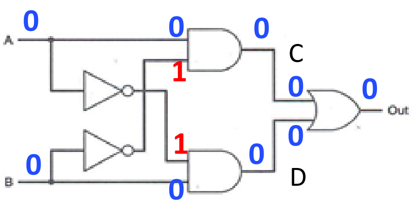


FIGURE 19-13 The circuit for Problem 19-3.

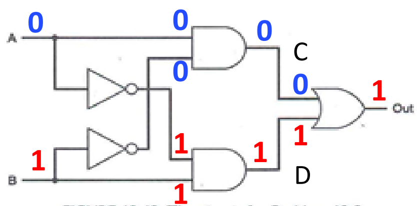


FIGURE 19-13 The circuit for Problem 19-3.

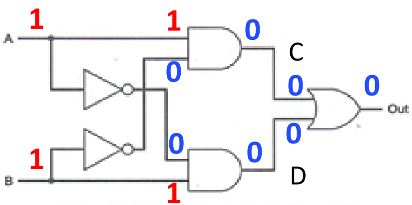


FIGURE 19-13 The circuit for Problem 19-3.

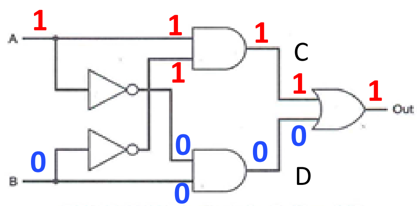


FIGURE 19-13 The circuit for Problem 19-3.

inputs		OR	XOR
A	B		
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	0

P6d Combining and using logic gates 0-3:00 (8:06)

Half Adder - Binary Math with Gates

Example

Add $5_{10}+3_{10}=8_{10}$

$A=5_{10}=0101_2=a_3+a_2+a_1+a_0$
$B=3_{10}= 0011_2=b_3+b_2+b_1+b_0$
$\mathbf{sum=s_3+s_2+s_1+s_0}$

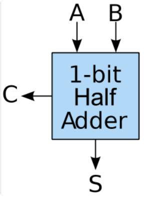
$\mathbf{s_0=a_0+b_0 \quad = 1+1 = 0}$
carry, $\mathbf{c_1=1}$
$s_1=a_1+b_1+c_1=0+1+1 = \mathbf{0}$
carry, $c_2=1$
$s_2=a_2+b_2+c_2=1+0+1 = \mathbf{0}$
carry, $c_3=1$
$s_3=a_3+b_3+c_3=0+0+1 = \mathbf{1}$

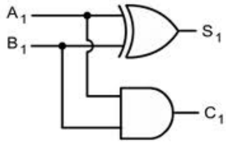
A = 0 1 0 1
B = 0 0 1 1
S = 1 0 0 0

$\mathbf{sum=1000_2 = 8_{10}}$

Half Adder = simple adder for 2 binary digits

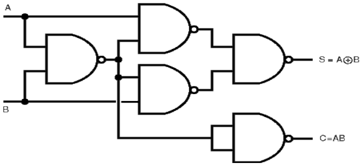
Simply add two digits
$a_0+b_0=s_0$
$1_2+1_2=10_2$
$s_0=0$ and $c_1=1$ (carry)





A_1	B_1	S_1	C_1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

XOR equivalent using
only 5 NAND gates →
(can be done with 4)



Full Adders

Example

Add $5_{10}+3_{10}=8_{10}$

$A=5_{10}=0101_2=a_3+a_2+a_1+a_0$

$B=3_{10}=0011_2=b_3+b_2+b_1+b_0$

$sum=s_3+s_2+s_1+s_0$

$s_0=a_0+b_0$ = +1 = 0

carry, $c_1=1$

$s_1=a_1+b_1+c_1=0+1+1=$ **0**

carry, $c_2=1$

$s_2=a_2+b_2+c_2=1+0+1=$ **0**

carry, $c_3=1$

$s_3=a_3+b_3+c_3=0+0+1=$ **1**

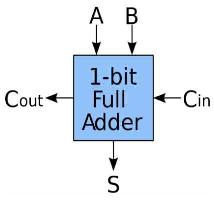
A = 0 1 0 1

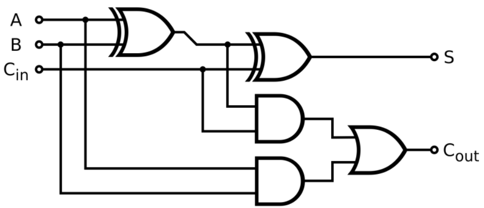
B = 0 0 1 1

S = 1 0 0 0

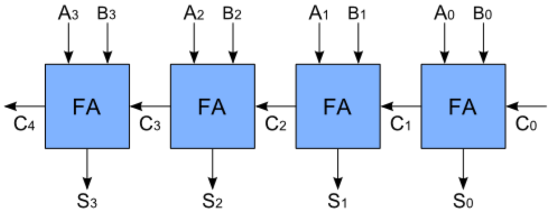
$sum=1000_2=8_{10}$

Full Adder = half adder + carry in





4-bit Adder = 4 Full Adders



Questions ?

Videos

DIGITAL BASICS

Minecraft Tutorial - Logic Gates - AND, OR, NOT (stop at 1:40)
http://www.youtube.com/watch?v=4SkI88bH07g, 7:50

Create digital circuits online or download
http://logic.ly/; Do problem **A** AND **B**

Logisim, free digital circuit simulator
http://www.youtube.com/watch?v=ATPqpFMlVdw, intro (7:45)

Wiring digital IC, (view 3:40-6:15)
http://www.youtube.com/watch?v=cdMJvFT-Afc&feature=related, start 2:28-6:15

logic.ly
Switch up/dn = 1/0
Filament white = 1

Lab-8a, Digital Circuits

I. Test digital logic gates using inputs of **0** or **+5 V**.
Determine output using **LED** and current-limiting resistor.

II. Measure the truth table of a unknown gates.
Construct an XOR gate using only a 4-gate 7400 NAND gate.

III. Design and construct a 4-bit decoder
Miniterms 3, 11 and 13

Wiring digital IC, (view 3:40-6:15)
http://www.youtube.com/watch?v=cdMJvFT-Afc&feature=related, start 2:28-6:15

Lab-8a, 4-bit Decoder

Design a 4-bit (ABCD) decoder circuit that lights an LED when the inputs correspond to the decimal numbers **3, 10** and **13.**

Truth Table for 3 and 13

#	A	B	C	D	Out
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4					

Truth Table for 3,11,13

AB / CD	00	01	11	10
00				
01				
11	1			
10				

Questions ?

ปลาย