EE271 Review Problem 0

- **As you wait for class to start, answer the following question:**
  - Bob has $500, but owes $300 to Shirley in CA, who's going to kill him if he doesn't pay off the money in person in a week. Plane tickets to CA cost $175, while bus tickets cost $75. Based on this, finish off the following statement:

    PLANE OR BUS

  - If Bob buys a ________ ticket then Bob won't be killed.

CSE 369: Introduction to Digital Design

- Professor Georg Seelig, CSE 228 (gseelig@uw.edu)
  - Office Hours: email w/schedule for a slot

- Book: Brown & Vranesic *Fundamentals of Digital Logic with Verilog Design* (3rd Edition)

- TAs:
  - Bin Yu (by23@uw.edu)
  - Yashin Chen (yashinc@uw.edu)

- Lab Hours: check website

Grading

- 70% - Labs
- 10% - Quizzes
- 20% - Final Exam
- Late penalties for uploading lab materials:
  - <24 hours: -10%
  - <48 hours: -30%
  - <72 hours: -60%
  - >72 hours: not accepted

Joint Work Policy

- Labs will be done alone
- Students may not collaborate on labs/projects, nor between groups on the specifics of homeworks.
- OK:
  - Studying together for exams
  - Discussing lectures or readings
  - Talking about general approaches
  - Help in debugging, tools peculiarities, etc.
- Not OK:
  - Developing a lab together
- Violation of these rules is grounds for failing the class

Class & Lab Meetings

- Labs:
  - Each student assigned a lab kit, can work where-ever.
  - In addition to the official sections, TAs will have some blocks of office hours to help with labs, etc.
  - Signups for lab demos will be posted shortly.

- Quiz: Tue, Feb 2 and Tue, Feb 23 in class
- Final: Mon, March 14, 10:30-11:20

Motivation

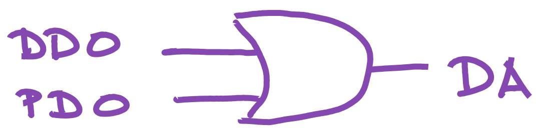
- Readings: 1-1.4, 2-2.4

- Electronics an increasing part of our lives
  - Computers & the Internet
  - Car electronics
  - Robots
  - Electrical Appliances
  - Cellphones
  - Portable Electronics
- Class covers digital logic design & implementation

Example: Car Electronics

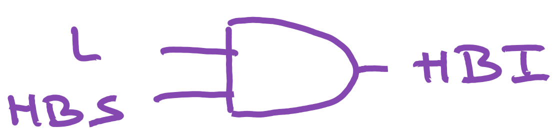
- Door Ajar (DriverDoorOpen, PassDoorOpen):

DA = DDO OR PDO



- High-beam indicator (lights, high beam selected):

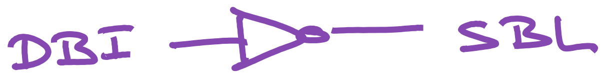
HBI = L AND HBS



Example: Car Electronics (cont.)

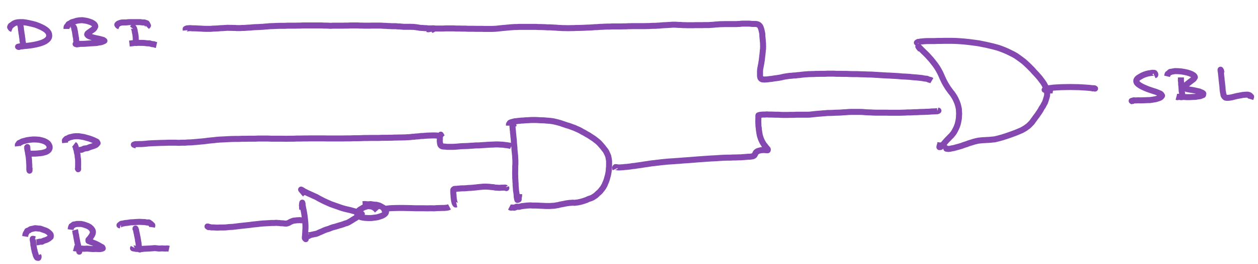
- Seat Belt Light (driver belt in):

SBL = NOT DBI



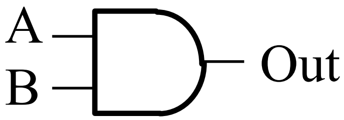
- Seat Belt Light (driver belt in, passenger belt in, passenger present):

SBL = NOT DBI OR (PP AND NOT PBI)

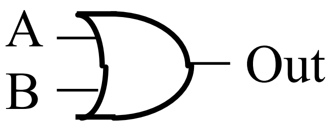


Basic Logic Gates

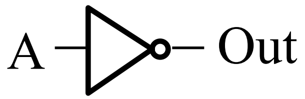
- AND: If A and B are True, then Out is True



- OR: If A or B is True, or both, then Out is True

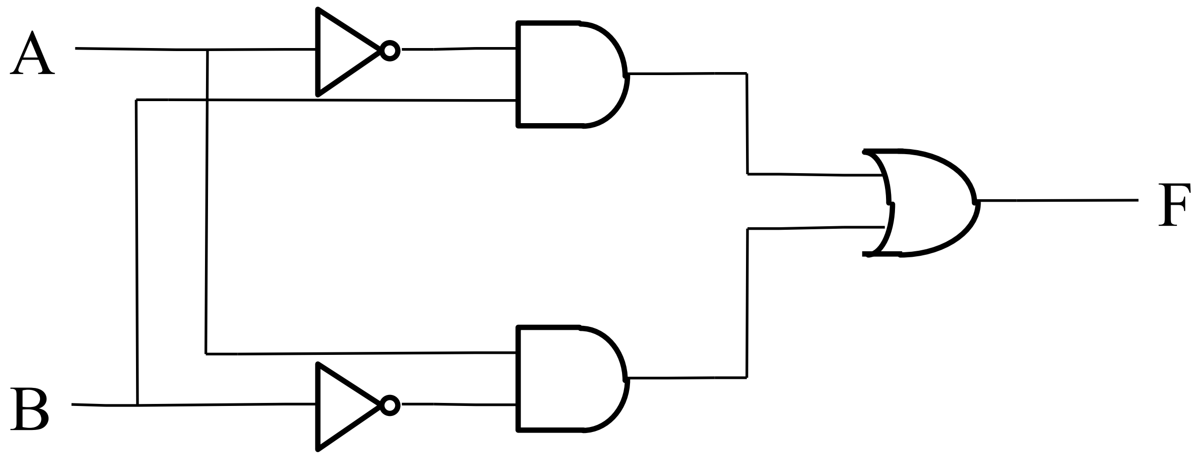


- Inverter (NOT): If A is False, then Out is True



Review Problem 2

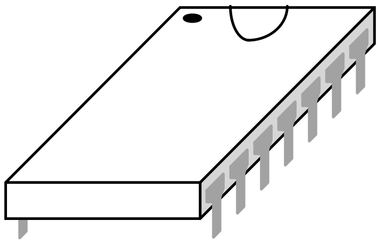
- ❖ What does the following circuit do?

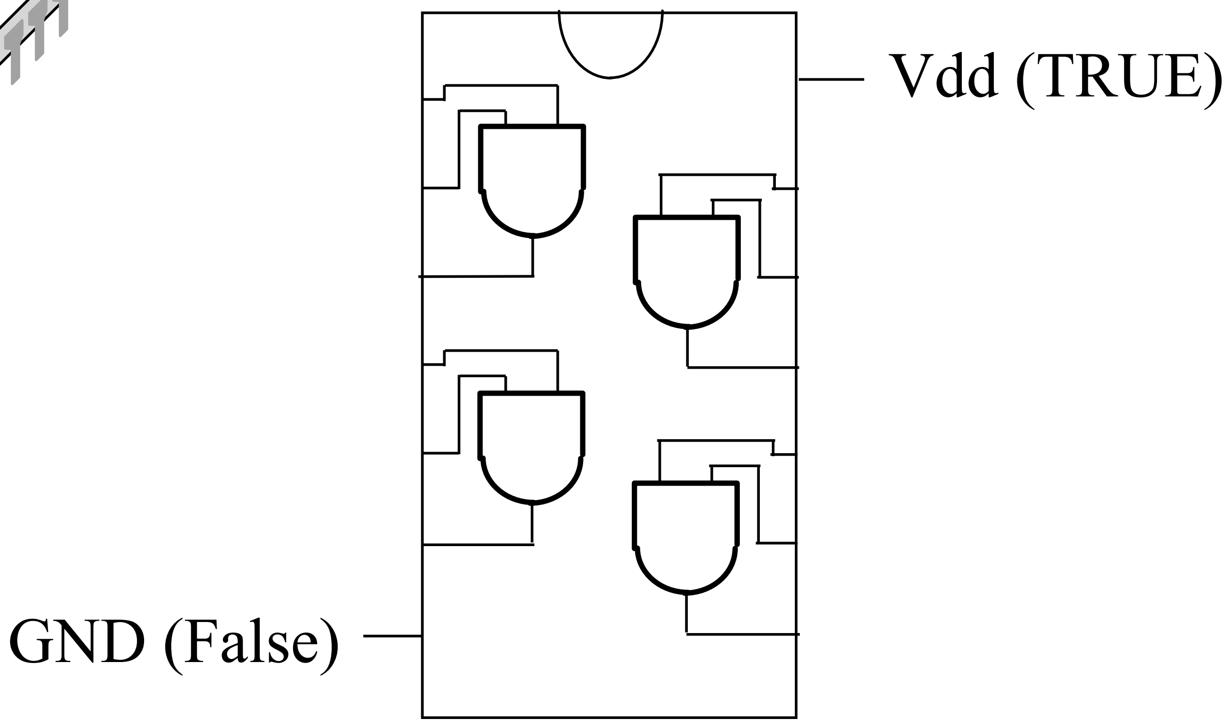


F = (NOT A AND B) OR (NOT B AND A)

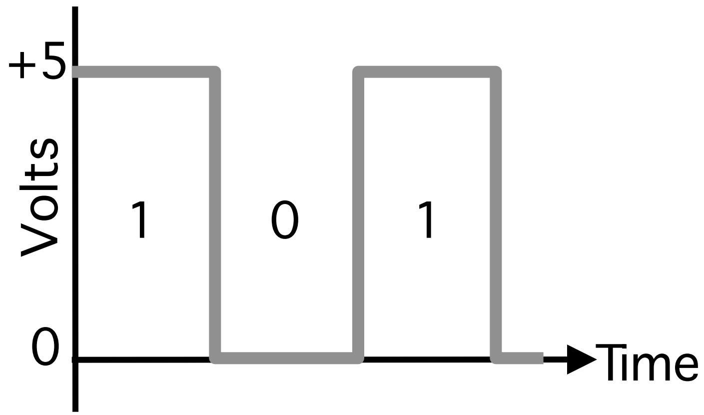
TTL Logic

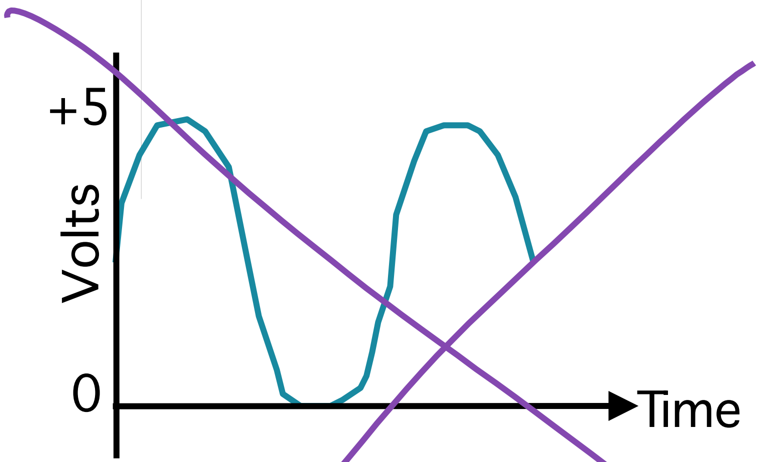
TRANSISTOR TRANSISTOR LOGIC





Digital vs. Analog





Digital:
only assumes discrete values

Binary/Boolean (2 values)
yes, on, 5 volts, high, TRUE, "1"
no, off, 0 volts, low, FALSE, "0"

Analog:
values vary over a broad range continuously

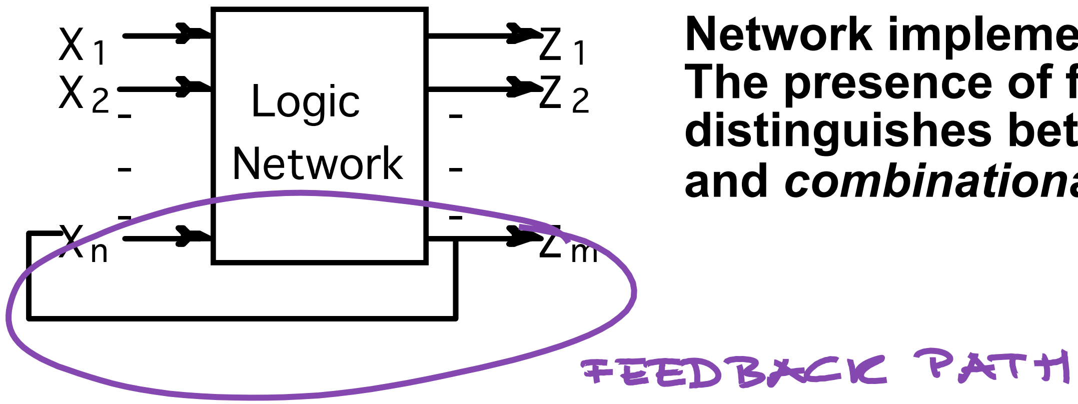
Advantages of Digital Circuits

- Analog systems: slight error in input yields large error in output
- Digital systems more accurate and reliable
  - Readily available as self-contained, easy to cascade building blocks
- Computers use digital circuits internally
- Interface circuits (i.e., sensors & actuators) often analog

This course is about logic design, not system design (processor architecture), not circuit design (transistor level)

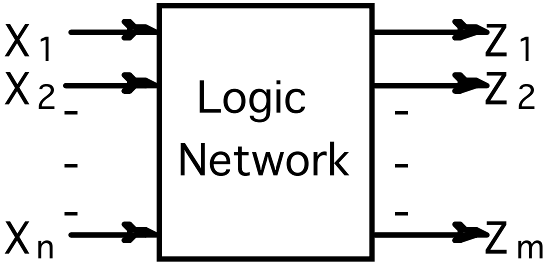
Combinational vs. Sequential Logic

Sequential logic



Network implemented from logic gates. The presence of feedback distinguishes between *sequential* and *combinational* networks.

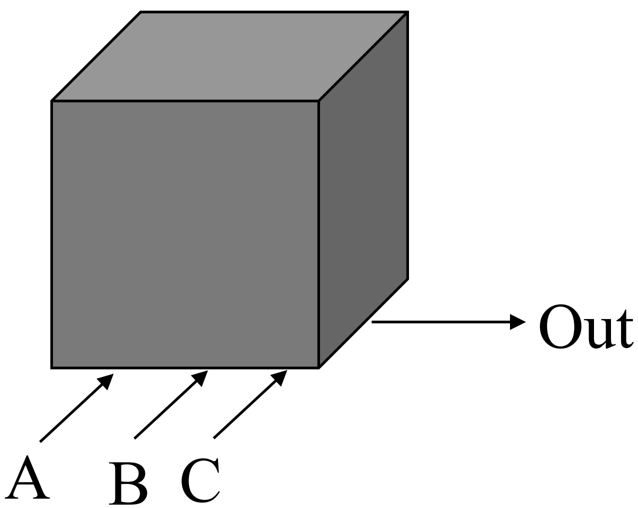
Combinational logic



No feedback among inputs and outputs. Outputs are a function of the inputs only.

Black Box (Majority)

- Given a design problem, first determine the function
- Consider the unknown combination circuit a "black box"



Truth Table

A	B	C	OUT
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Boolean Elements and truth tables

Algebra: **variables, values, operations**

In Boolean algebra, the values are the symbols 0 and 1
If a logic statement is false, it has value 0
If a logic statement is true, it has value 1

Operations: AND, OR, NOT

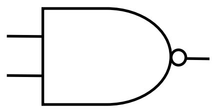
X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

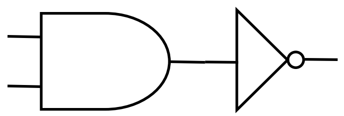
X	NOT X
0	1
1	0

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

NAND and NOR Gates

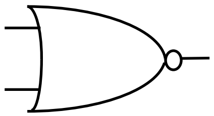
- NAND Gate: NOT(AND(A, B))

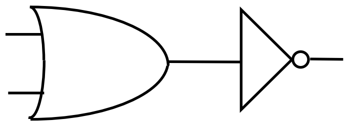




X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0

- NOR Gate: NOT(OR(A, B))

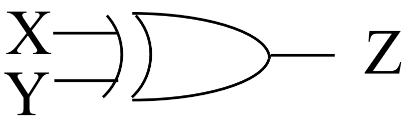




X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0

XOR and XNOR Gates

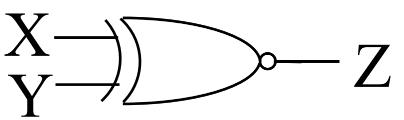
- XOR Gate: Z=1 if X is different from Y



$$X \oplus Y$$

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

- XNOR Gate: Z=1 if X is the same as Y



$$\overline{X \oplus Y}$$

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

Boolean Equations

Boolean Algebra

values: 0, 1
variables: A, B, C, . . ., X, Y, Z
operations: NOT, AND, OR, . . .

NOT X is written as $\overline{X}$
X AND Y is written as X * Y, or sometimes X Y or X & Y
X OR Y is written as X + Y

Deriving Boolean equations from truth tables:

Carry = AB

A	B	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

OR'd together *product* terms for each truth table row where the function is 1

if input variable is 0, it appears in complemented form; if 1, it appears uncomplemented

Sum = $A\overline{B} + \overline{A}B = A \oplus B$

Review Problem 3

- Does the following Boolean equation implement the function given in the truth table?

$$MyCout' = (A * B) + (A * Cin) + (A * B * Cin)$$

A	B	Cin	Cout
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

NOT EQUAL

Boolean Algebra/Logic Minimization

$$\bar{A}BC_{in} + A\bar{B}C_{in} + AB\overline{C_{in}} + ABC_{in} \quad vs. \quad AB + AC_{in} + BC_{in}$$

Logic Minimization: reduce complexity of the gate level implementation

- **reduce number of literals (gate inputs)**
- **reduce number of gates**
- **reduce number of levels of gates**

fewer inputs implies faster gates in some technologies

fan-ins (number of gate inputs) are limited in some technologies

fewer levels of gates implies reduced signal propagation delays

number of gates (or gate packages) influences manufacturing costs

Basic Boolean Identities:

$X + 0 = X$

$X * 1 = X$

$X + 1 = 1$

$X * 0 = 0$

$X + X = X$

$X * X = X$

$X + \overline{X} = 1$

$X * \overline{X} = 0$

$\overline{\overline{X}} = X$

Basic Laws

Commutative Law:

$X + Y = Y + X$ $\qquad$ $XY = YX$

Associative Law:

$X+(Y+Z) = (X+Y)+Z$ $\qquad$ $X(YZ)=(XY)Z$

Distributive Law:

$X(Y+Z) = XY + XZ$ $\qquad$ $X+YZ = (X+Y)(X+Z)$

Advanced Laws (Absorbtion)

- $X+XY =$ $x(1+y) = x$
- $XY + X\overline{Y} =$ $x(y+\overline{y}) = x$
- $X+\overline{X}Y =$ $x(1+y) + \overline{x}y = x + (x+\overline{x})y = x+y$
- $X(X+Y) =$ x
- $(X+Y)(X+\overline{Y}) =$ x
- $X(\overline{X}+Y) =$ xy

Boolean Manipulations (cont.)

- Boolean Function: $F = \overline{X}YZ + XZ$

Truth Table:

X	Y	Z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Reduce Function:

$= (\overline{x}y + x)z$ DIST

$= (x + y)z$ ABS.

$= xz + yz$ DIST.

BOTH OK

DeMorgan's Law

$$\overline{(X + Y)} = \overline{X} * \overline{Y}$$

X	Y	$\overline{X}$	$\overline{Y}$	$\overline{X+Y}$	$\overline{X}\cdot\overline{Y}$
0	0	1	1		
0	1	1	0		
1	0	0	1		
1	1	0	0		

$$\overline{(X * Y)} = \overline{X} + \overline{Y}$$

X	Y	$\overline{X}$	$\overline{Y}$	$\overline{X \cdot Y}$	$\overline{X}+\overline{Y}$
0	0	1	1		
0	1	1	0		
1	0	0	1		
1	1	0	0		

DeMorgan's Law can be used to convert AND/OR expressions to OR/AND expressions

Example:

$$Z = \overline{A}\,\overline{B}\,C + \overline{A}\,B\,C + A\,\overline{B}\,C + A\,B\,\overline{C}$$

$$\overline{Z} = (A + B + \overline{C}) * (A + \overline{B} + \overline{C}) * (\overline{A} + B + \overline{C}) * (\overline{A} + \overline{B} + C)$$

DeMorgan's Law example

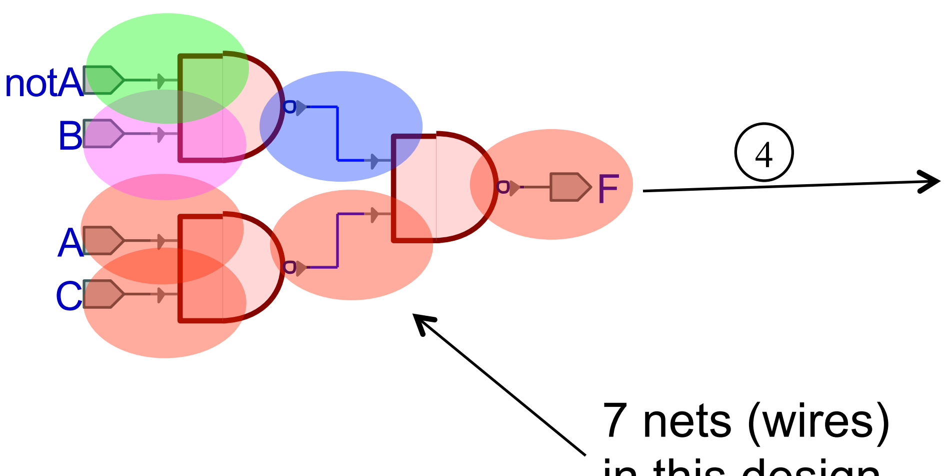
- If $F = (XY+Z)(\overline{Y}+\overline{X}Z)(X\overline{Y}+\overline{Z})$,

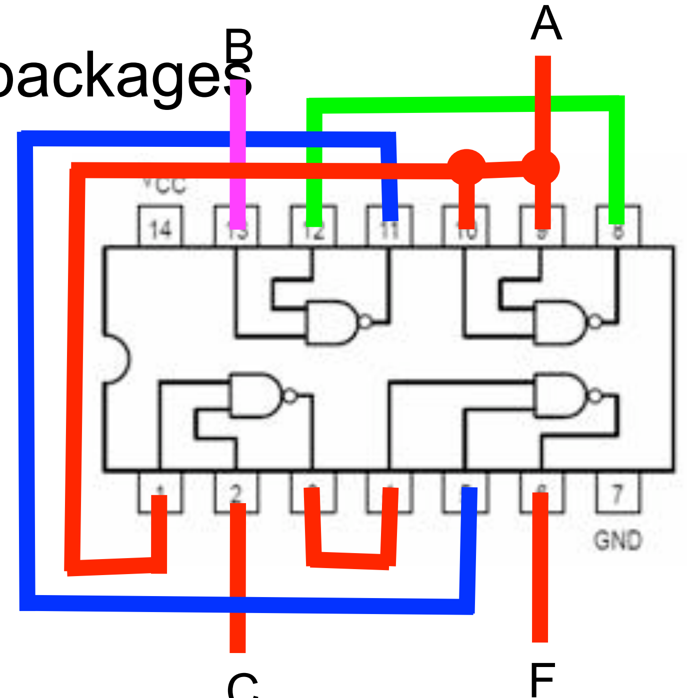
$\overline{F} = \overline{((XY)+Z)(\overline{Y}+(\overline{X}Z))((X\overline{Y})+\overline{Z})}$

$= (\overline{X}+\overline{Y})\overline{Z} + Y(X+\overline{Z}) + (\overline{X}+Y)Z$

Mapping truth tables to logic gates

- Given a truth table:
  - Write the Boolean expression
  - Minimize the Boolean expression
  - Draw as gates
  - Map to available gates
  - Determine number of packages and their connections





Breadboarding circuits