Penn State Abington
CMPEN 271
Lecture Set #6

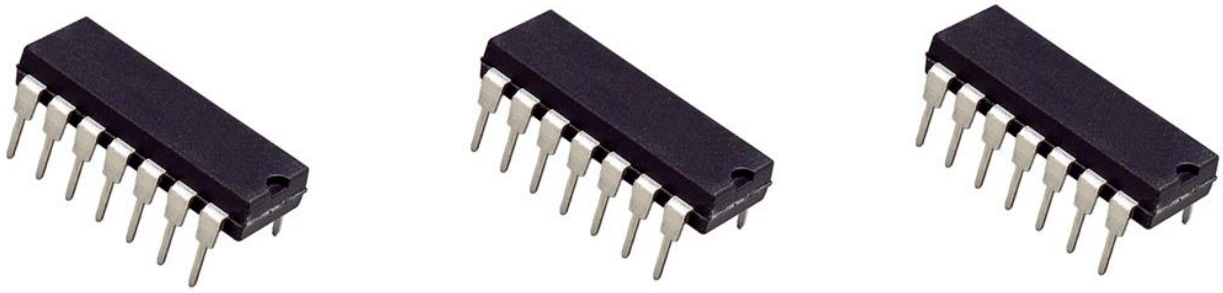
Topics:

- Standard Forms (minterms, maxterms, SOP, POS) Video part 1 of 4 ←
- NAND & NOR Gates Video part 2 of 4
- Practice Exercises Video part 3 of 4
- HW #3 Video part 4 of 4

Real-World Design Problem

Your **boss** calls you into her office to discuss your car alarm circuit. You were successful in reducing the circuit to use less logic gates, but there is still a problem -- your solution requires **too many integrated circuits or IC's**. There is not enough room ("real estate") on the circuit board to fit your design.

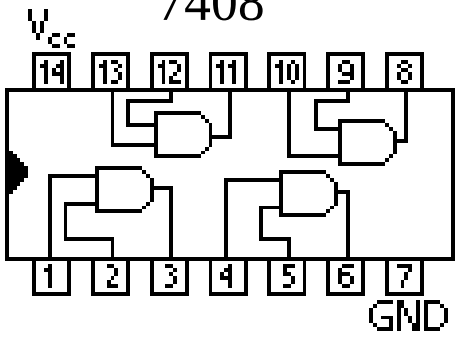
It might be beneficial and efficient if your circuit could be redesigned **to use NAND gates only**. This might **reduce the total number of IC's on the board**. Check it out and compare the total number of IC's required for each approach. "Have the solution by 10 am tomorrow morning."



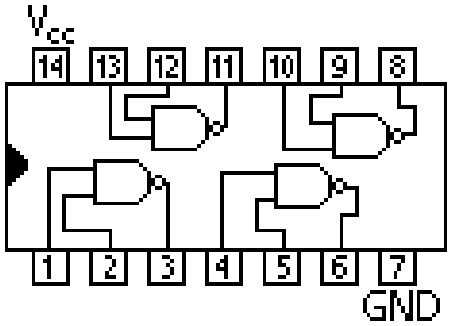
Integrated Circuits (ICs)

- 7400 TTL Logic Family of Integrated Circuits
- Note how many gates are contained within one IC package.
- The number of ICs determine how much room (“real estate”) a circuit requires on a circuit board,
- In general, it is desirable to reduce the total number of ICs (not just gates)
- ICs take up room, eat up power, and generate heat.
- Reducing the number of gates may reduce the number of ICs. Provide some examples and counter-examples
- Q: Which takes more space? 1) circuit with 2 gates (AND, OR) or 2) circuit with 4 AND gates?

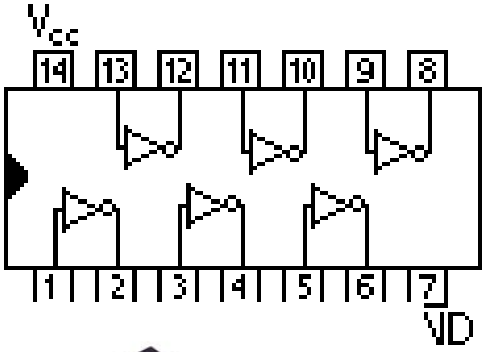
7408



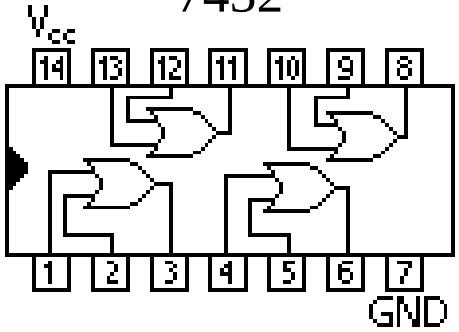
7400 (NAND)

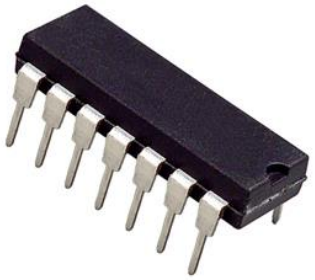


7404



7432





Minterms and Maxterms

A	B	C	minterms	symbol	maxterms	symbol
0	0	0	A'B'C'	m_0	(A+B+C)	M_0
0	0	1	A'B'C	m_1	(A+B+C)	M_1
0	1	0	A'BC'	m_2	(A+B+C)	M_2
0	1	1	A'BC	m_3	(A+B+C)	M_3
1	0	0	AB'C'	m_4	(A+B+C)	M_4
1	0	1	AB'C	m_5	(A+B+C)	M_5
1	1	0	ABC'	m_6	(A+B+C)	M_6
1	1	1	ABC	m_7	(A+B+C)	M_7

- Each line of a truth table has a corresponding minterm and maxterm
- Every Boolean function can be expressed as 1) a sum of minterms, and b) a product of maxterms. The purpose is a convenient notation. Stress sum of minterms notation.
- Generalize table above to functions of 2, 4, 5 variables (3 variable shown above)

Sum of Minterms Example - 1

Start with truth table

Revisit Example:

A	B	C	F	
0	0	0	0	
0	0	1	0	
0	1	0	1	→ A'BC' = $\mathbf{m_2}$
0	1	1	0	
1	0	0	0	
1	0	1	1	→ AB'C = $\mathbf{m_5}$
1	1	0	0	
1	1	1	1	→ ABC = $\mathbf{m_7}$

Identify each line of the truth table where output F = 1; find minterms for those lines.

Express function as sum of minterms using 2 formats:

F = A'BC' + AB'C + ABC

- or -

F = $\mathbf{m_2}$ + $\mathbf{m_5}$ + $\mathbf{m_7}$

= Σ m (2, 5, 7)

Sum of minterms notation is often convenient

Sum of Minterms Example - 2

Start with truth table

Revisit Example:

A	B	C	F	
0	0	0	1	→ A'B'C' = m_0
0	0	1	1	→ A'B'C = m_1
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	1	→ AB'C = m_5
1	1	0	1	→ ABC' = m_6
1	1	1	0	

Express function as sum of minterms in 2 formats:

F(A,B,C) = A'B'C' + A'B'C
+ AB'C + ABC'

- or -

F(A,B,C) = **Σ m (0,1, 5, 6)**

Sum of Minterms Example - 3

A	B	C	D	F	
0	0	0	0	0	
0	0	0	1	1	→ A'B'C'D = m_1
0	0	1	0	0	
0	0	1	1	1	→ A'B'CD = m_3
0	1	0	0	1	→ A'BC'D' = m_4
0	1	0	1	0	
0	1	1	0	0	
0	1	1	1	0	
1	0	0	0	0	
1	0	0	1	1	→ AB'C'D = m_9
1	0	1	0	1	→ AB'CD' = m_{10}
1	0	1	1	0	
1	1	0	0	1	→ ABC'D' = m_{12}
1	1	0	1	0	
1	1	1	0	0	
1	1	1	1	1	→ ABCD = m_{15}

Express function F as a sum of minterms

F(A,B,C,D) =
= Σ m (1,3,4,9, 10, 12, 15)

= A'B'C'D + A'B'CD + A'BC'D' + AB'C'D + AB'CD' + ABC'D' + ABCD

Question:
What is ambiguity with the expression:
F = Σ m (0, 1, 7) versus
F(X,Y,Z)= Σ m (0, 1, 7) versus
F(W,X,Y,Z) = Σ m (0, 1, 7) ???

Sum of Minterms Example - 4

Revisit Example:

A	B	C	F	
0	0	0	1	→ A'B'C' = $\mathbf{m_0}$
0	0	1	1	→ A'B'C = $\mathbf{m_1}$
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	1	→ AB'C = $\mathbf{m_5}$
1	1	0	1	→ ABC' = $\mathbf{m_6}$
1	1	1	0	

- Express F as sum of minterms:
F(A,B,C) = Σ m (0,1, 5, 6)

- Express F' (inverse) as sum of minterms

F'(A,B,C) = Σ m (2, 3, 4, 7)

- Explain operation to find F' as sum of minterms. Generalize.

Sum of Products (SOP)

What is SOP form of a Boolean function?

Examples of SOP form:

F = AB + CD'

F = ABC + ABC' + AB'C'

F = A' + B'C + ABCD

Examples of non-SOP form

F = (AB)' + BC

F = B(C + D)

F = A + BC' + (CD')'

Observations? SOP versus minterms?

Product of Sums (POS)

What is POS form of a Boolean function?

Examples of POS form:

1) F = (A+B) • (C+D')
2) F = (A+B+C) • (A+B+C') • (A+B'+C')
3) F = A' • (B'+C) • (A+B+C+D)

Examples of non-POS form

1) F = (A+B) • (C+D')'
2) F = (AC + D) • (A + B)
3) F = A + BC' + CD

Observations? POS versus maxterms? How do you convert from POS to SOP?

Sample Questions

#1. Express this function F(A,B,C) = Σ m (0, 5, 7) algebraically as a sum of minterms (do not minimize) and show the truth table.

$F(A,B,C) = \Sigma m\,(0, 5, 7)$

$= m_0 + m_5 + m_7$

$= A'B'C' + AB'C + ABC$

0 0 0 — 1 0 1 — 1 1 1

0 — 5 — 7

	A	B	C	F
m_0	0	0	0	1
m_1	0	0	1	0
m_2	0	1	0	0
m_3	0	1	1	0
m_4	1	0	0	0
m_5	1	0	1	1
m_6	1	1	0	0
m_7	1	1	1	1

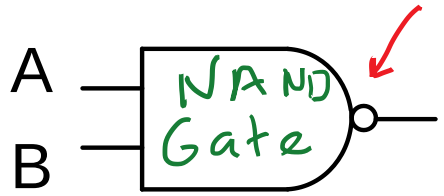
Penn State Abington
CMPEN 271
Lecture Set #6

Topics:

- Standard Forms (minterms, maxterms, SOP, POS) Video part 1 of 4
- NAND & NOR Gates Video part 2 of 4 ←
- Practice Exercises Video part 3 of 4
- HW #3 Video part 4 of 4

NAND Gate

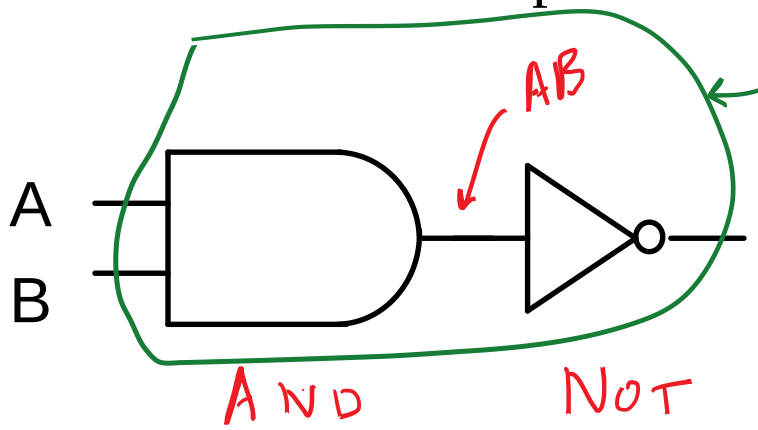
Definition:



Circle (bubble) designates inversion

$$F = (A \bullet B)' = \overline{A \bullet B}$$

-- same operation as --

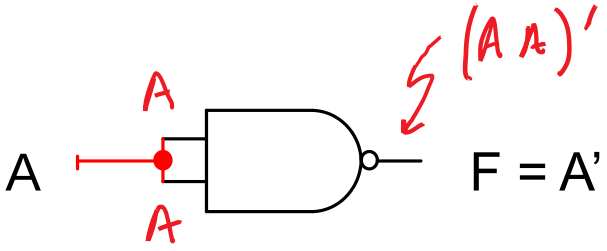


$$F = (A \bullet B)' = \overline{A \bullet B}$$

NAND Gate as a Universal Gate

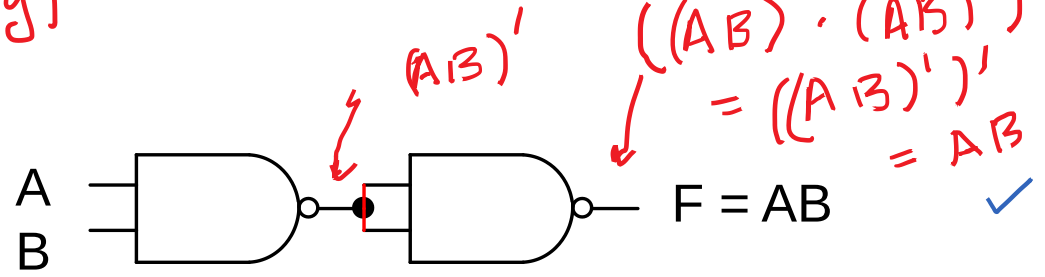
Use Boolean Algebra

NOT operation:

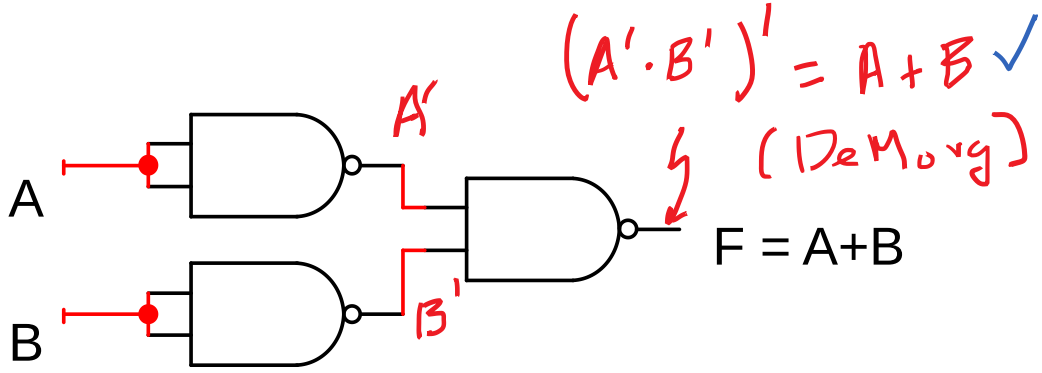


(NOT has 1 input only gate)

AND operation:



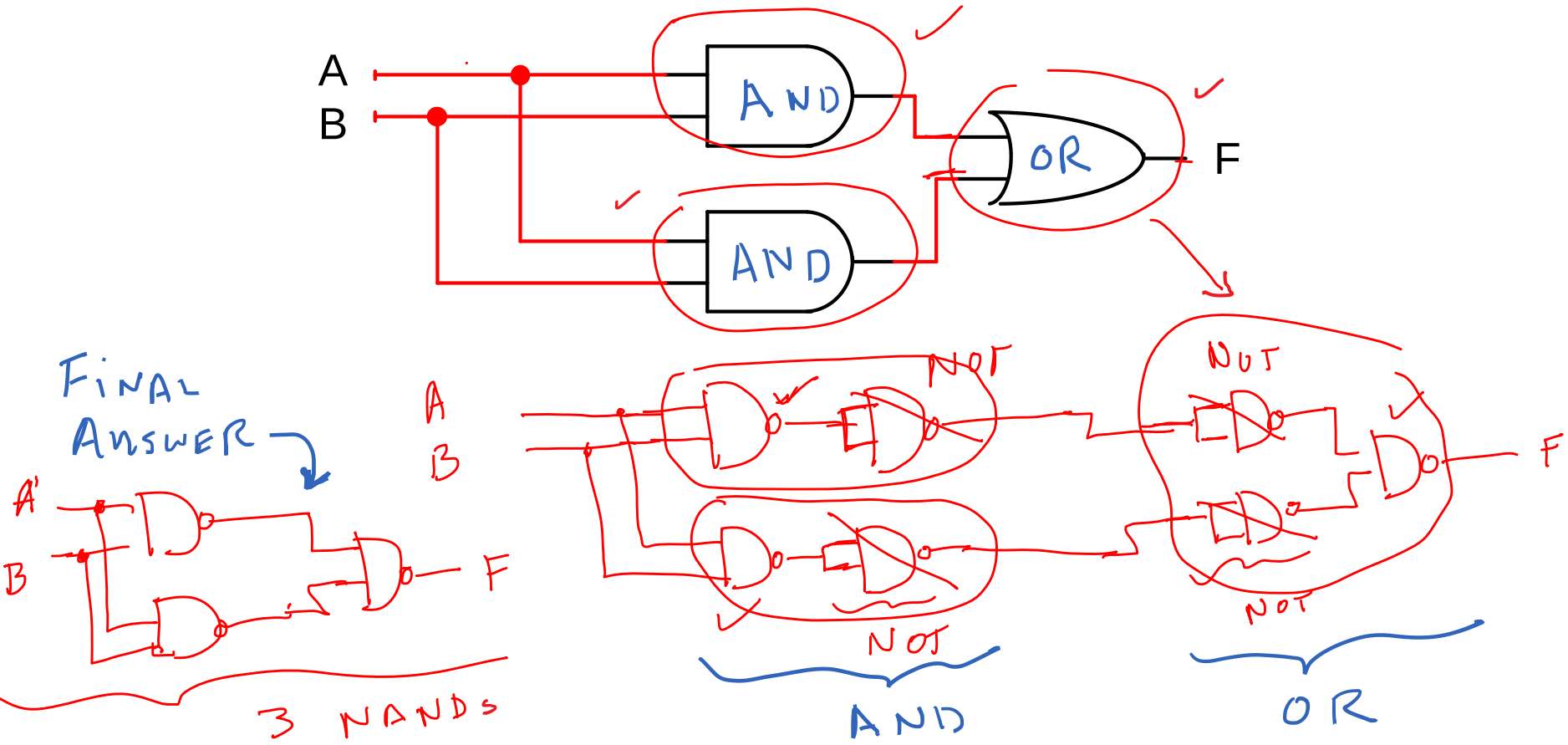
OR operation:



NAND gates can be used to create NOT, AND, and OR operations

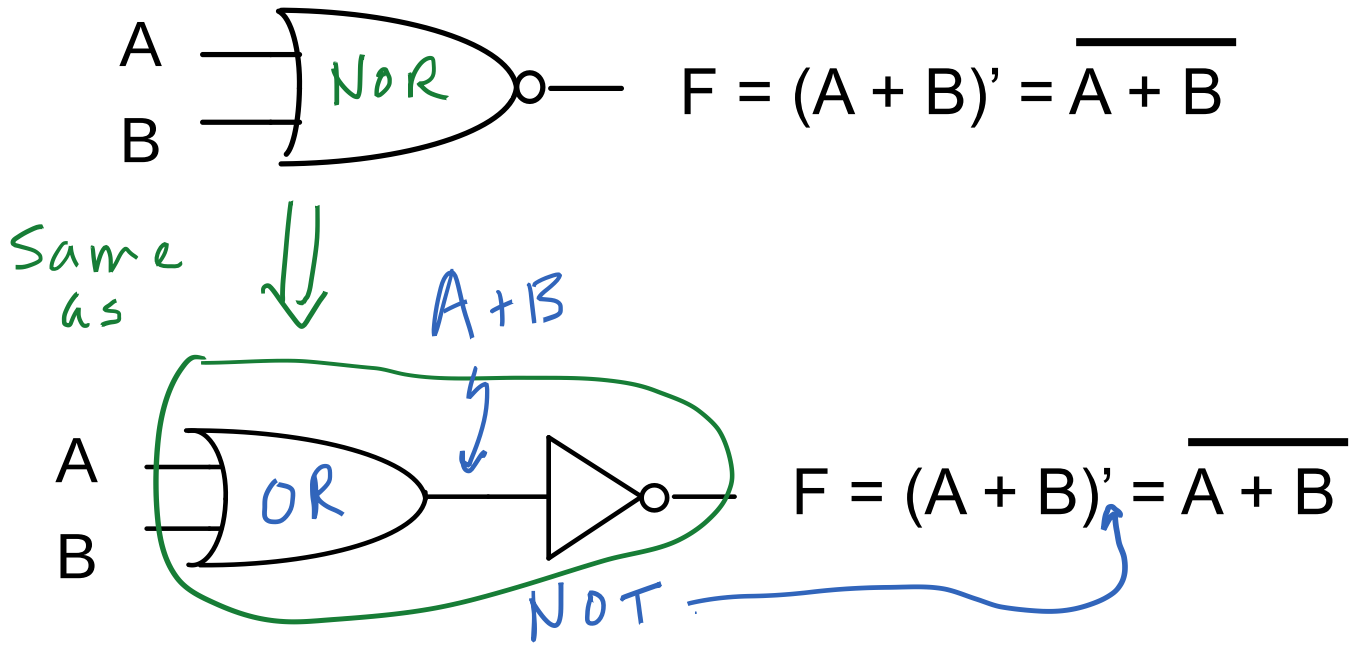
Convert AND/OR/NOT Gates to NAND

Exercise: given circuit below, convert to all NAND gates:



Note: Minimize NAND gate solution by removing any "double" inverters (equiv. NOT gates). Generalize to any AND/OR/NOT circuit.

NOR Gate as a Universal Gate



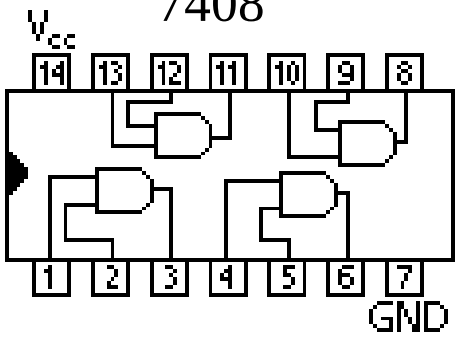
Demonstrate that the NOR gate is also a universal gate.

In this course, we will focus on NAND

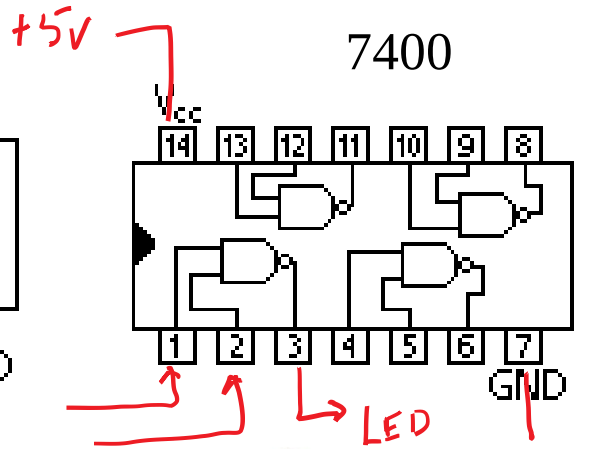
Integrated Circuits (ICs)

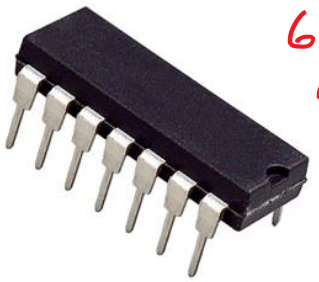
- 7400 TTL Logic Family of Integrated Circuits (compare to CMOS)
- Note how many gates are contained within one IC package.
- The number of ICs determine how much room (“real estate”) a circuit requires on a circuit board,
- In general, it is desirable to reduce the total number of ICs
- ICs take up room, eat up power, and generate heat.
- Reducing the number of gates may reduce the number of ICs. Provide some examples and counter-examples

7408

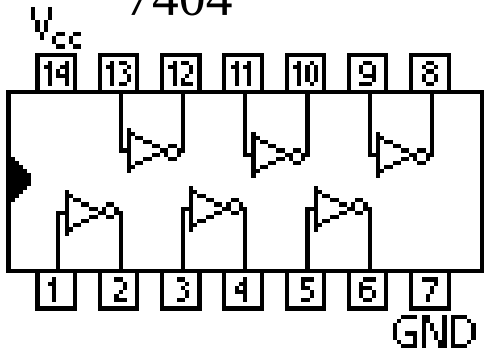


7400

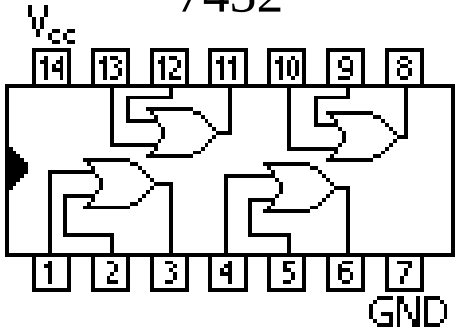




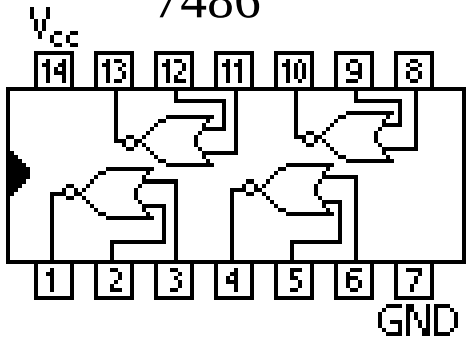
7404



7432



7486



Penn State Abington
CMPEN 271
Lecture Set #6

Topics:

- Standard Forms (minterms, maxterms, SOP, POS) Video part 1 of 4
- NAND & NOR Gates Video part 2 of 4
- Practice Exercises Video part 3 of 4 ←
- HW #3 Video part 4 of 4

Practice Exercises

For each Boolean expression below, 1) minimize the function using Boolean algebra identities and theorems into SOP form, 2) Draw logic diagram circuit using AND, OR, NOT gates, 3) convert logic diagram to all NAND gates.

#1) F = ABC + A'BC + AB'C

#2) F = XYZ + XY'(X'Z')'

#3) F = (A' + C)(A + B + C')(B' + C)

#4) F(A,B,C) = Σ m (0,1, 2, 6)

Practice Exercises

For each Boolean expression below, 1) minimize the function using Boolean algebra identities and theorems into SOP form, 2) Draw logic diagram circuit using AND, OR, NOT gates, 3) convert logic diagram to all NAND gates.

#1) F = ABC + A'BC + AB'C

$= BC(A + A') + AB'C$

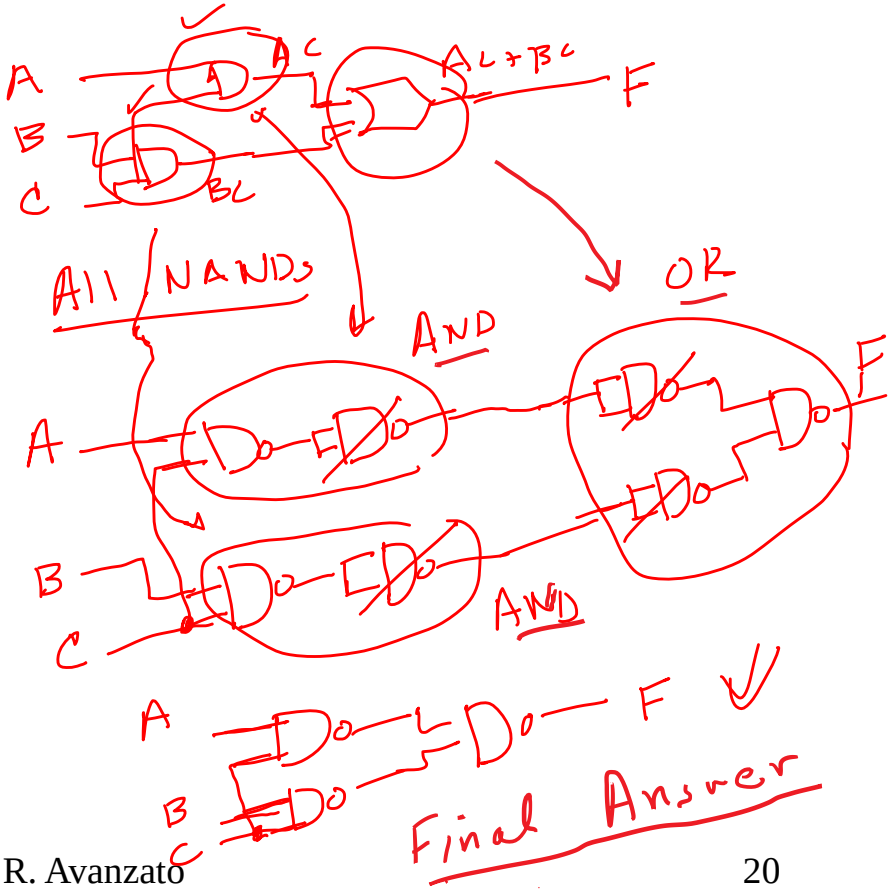
$= BC + AB'C$

$= C(B + AB')$

$= C(B + A)(B + B')$

$= C(B + A)$

$= BC + AC$



Practice Exercises

For each Boolean expression below, 1) minimize the function using Boolean algebra identities and theorems into SOP form, 2) Draw logic diagram circuit using AND, OR, NOT gates, 3) convert logic diagram to all NAND gates.

#2) F = XYZ + XY'(X'Z')'

do first (D.M.)

$= XYZ + XY'(X + Z)$

$= XYZ + XY'X + XY'Z$

$= XYZ + XY' + XY'Z$

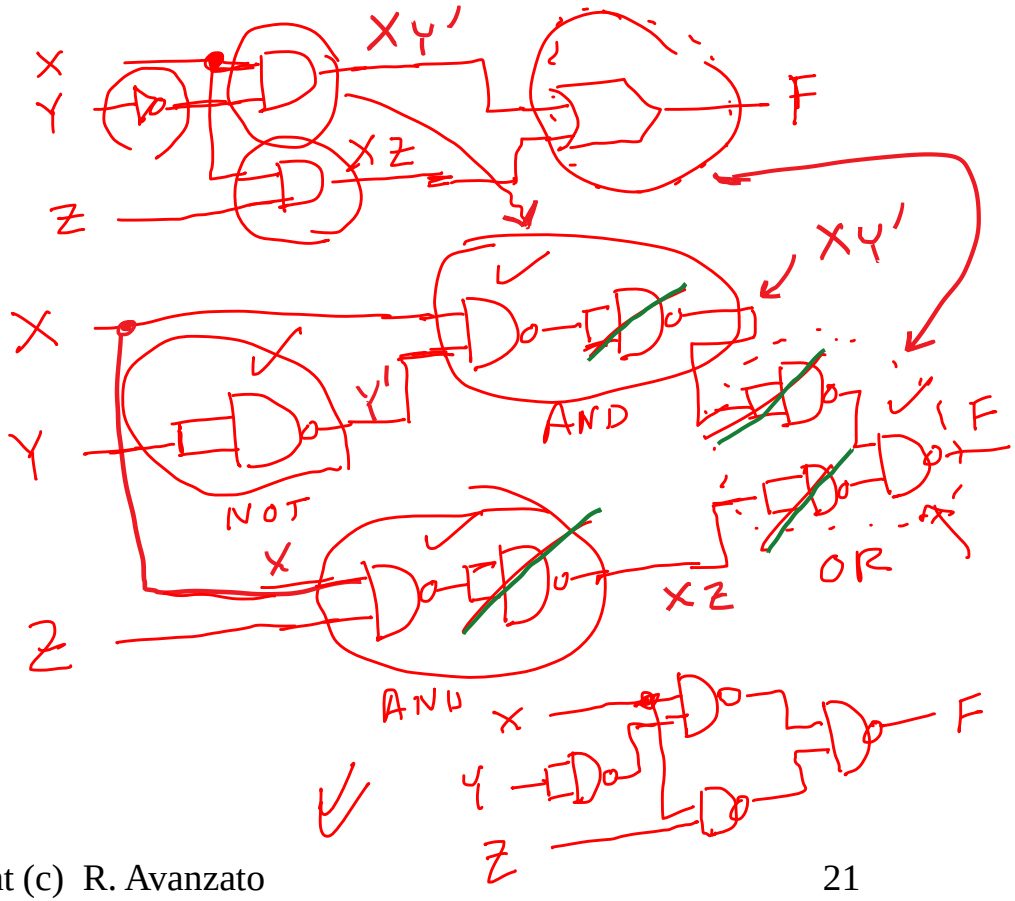
$= XYZ + XY'(1 + Z')$

$= XYZ + XY'$

$= X(YZ + Y')$

$= X(Y' + Y)(Y' + Z)$

$F = XY' + XZ$



Practice Exercises

For each Boolean expression below, 1) minimize the function using Boolean algebra identities and theorems into SOP form, 2) Draw logic diagram circuit using AND, OR, NOT gates, 3) convert logic diagram to all NAND gates.

#3) F = (A' + C)(A + B + C')(B' + C)

$= (A'A + A'B + A'C' + AC + BC + CC')(B' + C)$

$= (A'B + A'C' + AC + BC)(B' + C)$

$= A'BB' + A'C'B' + ACB' + BCB' + A'BC + A'CC' + ACC + BCC$

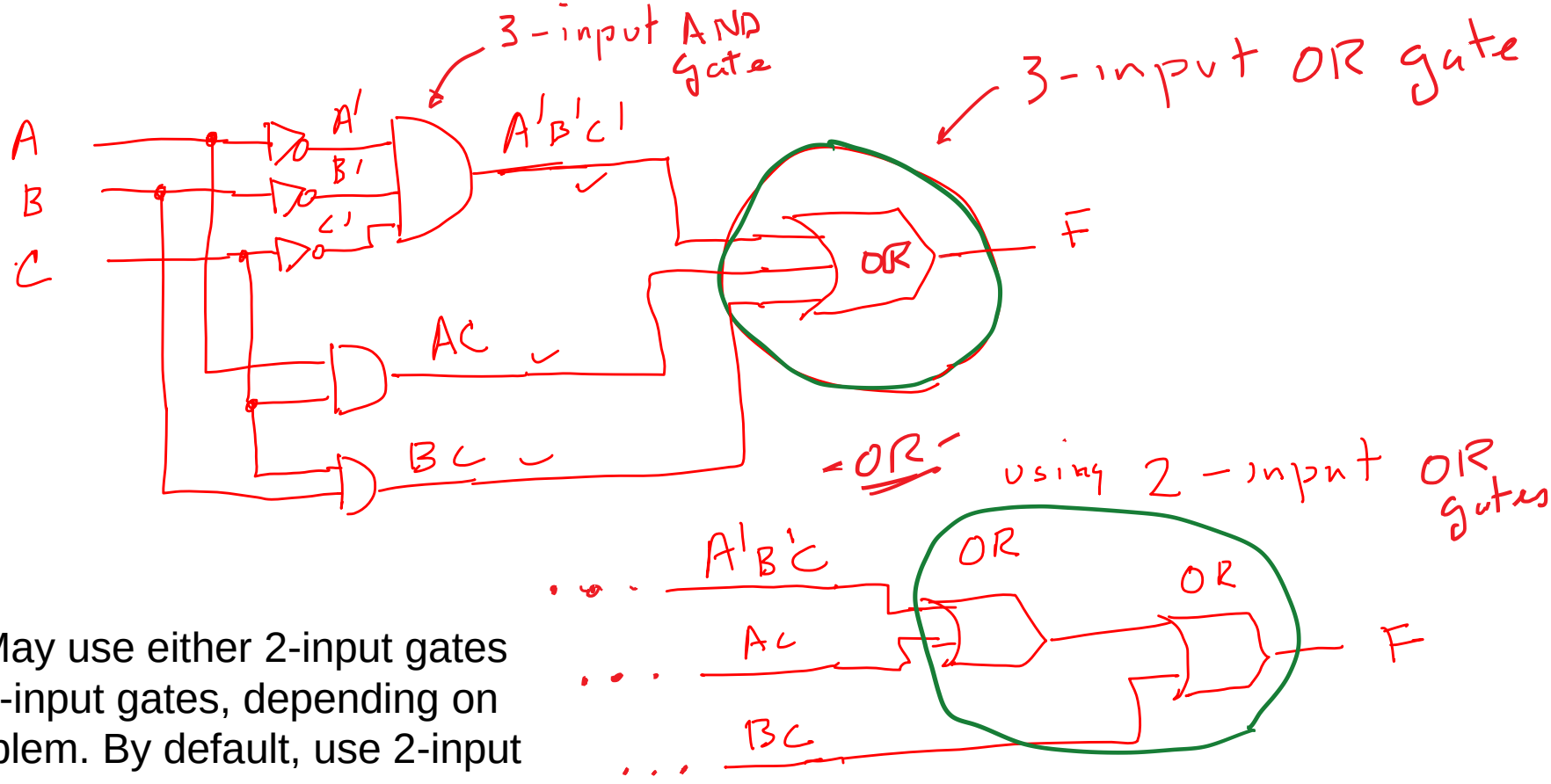
$= A'C'B' + ACB' + A'BC + AC + BC$

$AC(1 + B') = AC$

$BC(1 + A') = BC$

$= A'B'C' + AC + BC$ (SOP)

F = A'B'C' + AC + BC (problem #3 continued)



1) May use either 2-input gates or 3-input gates, depending on problem. By default, use 2-input gates

2)Try converting to all NANDs. Use software tool to verify.

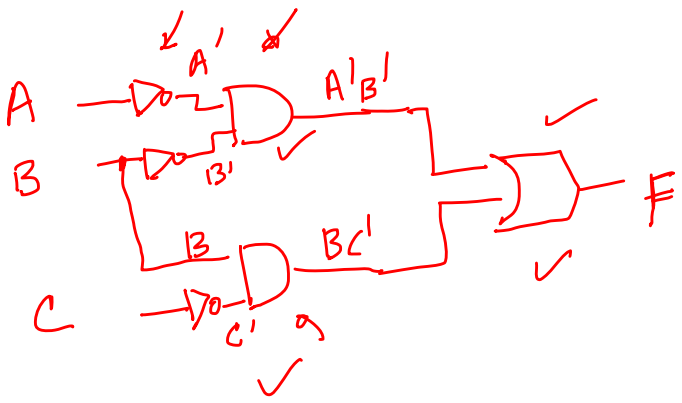
Practice Exercises

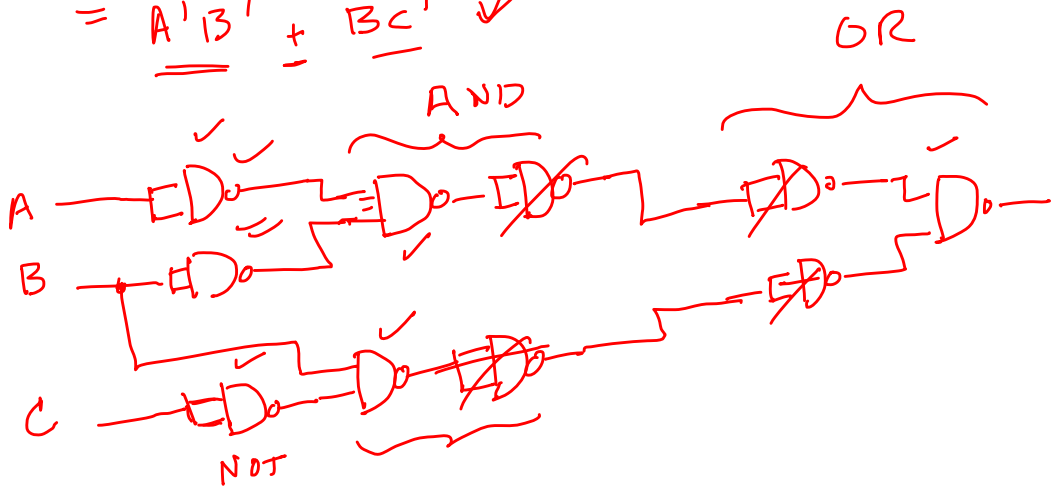
For each Boolean expression below, 1) minimize the function using Boolean algebra identities and theorems into SOP form, 2) Draw logic diagram circuit using AND, OR, NOT gates, 3) convert logic diagram to all NAND gates.

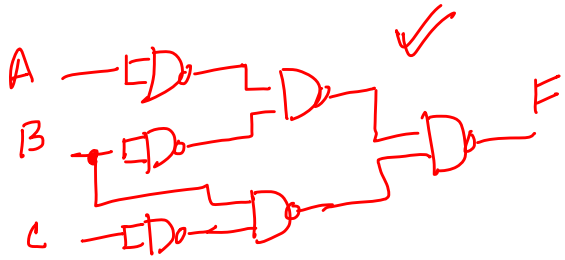
#4) $F(A,B,C) = \Sigma\, m\,(0,1,2,6) = A'B'C' + A'B'C + A'BC' + ABC'$ (m_0, m_1, m_2, m_6)

$= A'B'(C'+C) + BC'(A'+A)$

$= A'B' + BC'$







Summary of Key Ideas

- Each entry (line) in a truth table has a corresponding minterm and a maxterm.
- Any Boolean function can be expressed as a sum ("or" operation) of minterms.
- Any Boolean function can also be expressed as a product ("and" operation) of maxterms.
- The advantage of expressing a Boolean function as a function of minterms or maxterms is that it is a concise (compact) notation.
- We will stress the sum of minterms expression for Boolean functions.
- Each minterm (which is a product term) contains each input variable (either complemented or not complemented).
- Sum of products (SOP) is more general than sum of minterms.
- Product terms in a SOP do not need to contain every input variable.
- Every sum of minterms is a SOP, but not the other way.
- Product of sums (POS) is more general than sum of maxterms.
- It is possible to convert any SOP to an equivalent POS.
- A NAND gate is equivalent to an AND gate with an inverted output.
- A NAND gate is a universal gate because you can build a NOT, OR, or AND gate using only one or more NAND gates.
- Any circuit containing AND, OR, or NOT gates can be converted to an equivalent circuit containing only NAND gates
- Designing a circuit with only NAND gates can result in a circuit with less integrated circuits (which is a good thing).
- A NOR gate is also a universal gate.

What you should know

- Be able to generate a truth table from a word description of an engineering design problem (such as the car alarm problem).
- Given an arbitrary truth table, be able to express the function as a sum of minterms.
- Know the distinction between sum of minterms and SOP.
- Know distinction between SOP and POS forms.
- Know the symbol and definition of NAND and NOR gate.
- Demonstrate that NAND and NOR gates are universal gates
- Know how to convert an arbitrary AND/OR/NOT circuit into an equivalent circuit of NAND gates only. Remove any unnecessary NAND gates (cases of double inversion)

Penn State Abington
CMPEN 271
Lecture Set #6

Topics:

- Standard Forms (minterms, maxterms, SOP, POS) Video part 1 of 4
- NAND & NOR Gates Video part 2 of 4
- Practice Exercises Video part 3 of 4
- HW #3 Video part 4 of 4 ←

HW#3 – See due date

1. **Car Alarm Design** (see next slide)
2. **Majority Circuit** (see next slide)

   A total of 2 problems are due. Each solution has a separate drop box.

- Include original problem and include all steps.
- Use standard formatting and documentation. Must be typed.
- Include truth table, minimization, and circuit. Use Boolean algebra to minimize circuit – must show all steps in solution. Simulate minimal circuit.
- Use MS Word "table" for truth table
- **Include 2 versions of each circuit: 1) Use AND, OR, NOT gates only, 2) use NAND gates only. Calculate # of ICs for each version.**
- Use Multisim (or LogicWorks) for all circuit diagrams. Include labels for all input and output (use text tool; Place → Text)
- Note: You can use Multisim circuit to draw circuit – you do not need to draw any circuits by hand
- Copy and Paste Multisim circuit into Word doc (do not upload Multisim file)
- **All input wires and output wires must be labeled.**
- Upload each solution in a single Word document into separate Canvas dropbox. Include description of problem (copy from notes). Must include name, date, label inputs and outputs in circuit.
- Individual effort, not team effort. Collaboration is encouraged, but no copying.
- Ask questions

Homework #3 Problems

#1. Car alarm Circuit

Design a minimal circuit to implement the door alarm Boolean function.

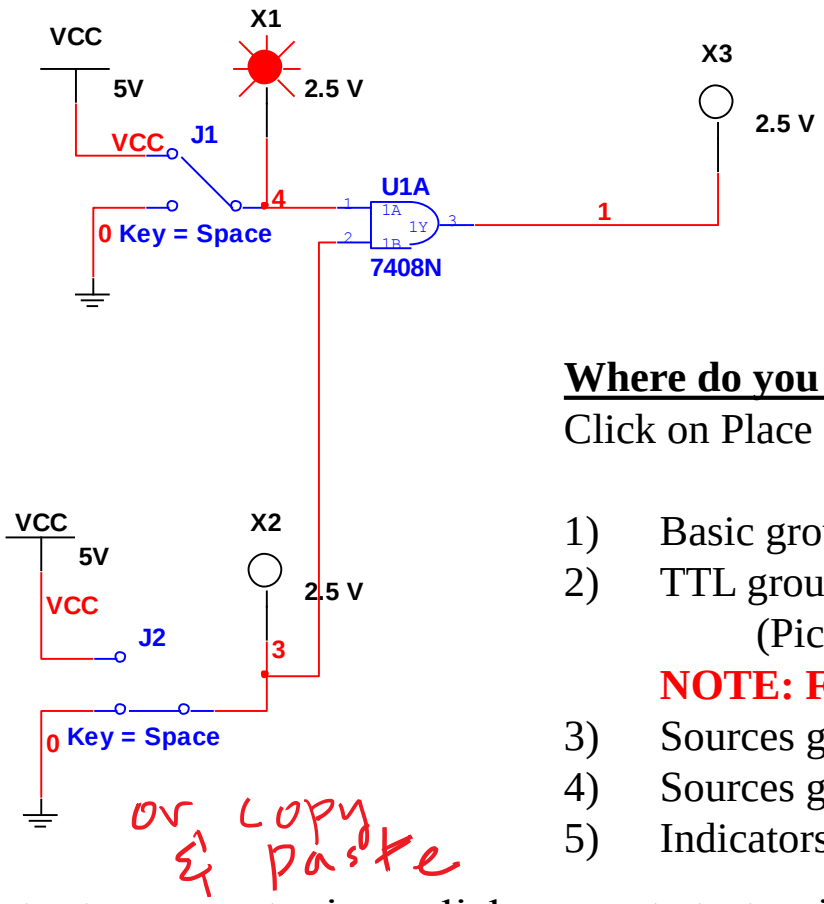
Alarm = doorOpen • alarmEnabled + doorOpen • alarmEnabled • motionSensorActive + alarmEnabled • motionSensorActive

#2. Majority Voting Circuit

Design a circuit with 3 inputs and 1 output. The output will be high when 2 or more of the inputs are high, otherwise the output will be low.

MultiSIM software

Note: to add labels in Multisim, select Place→ Text



What components do you need to build this circuit?

- SPDT switches
- 7408N AND Gate
- Vcc (+5 volts)
- Ground (0 volts)
- Probes

Demo!!

Where do you find components in MultiSim?

Click on Place → Component

Or use search

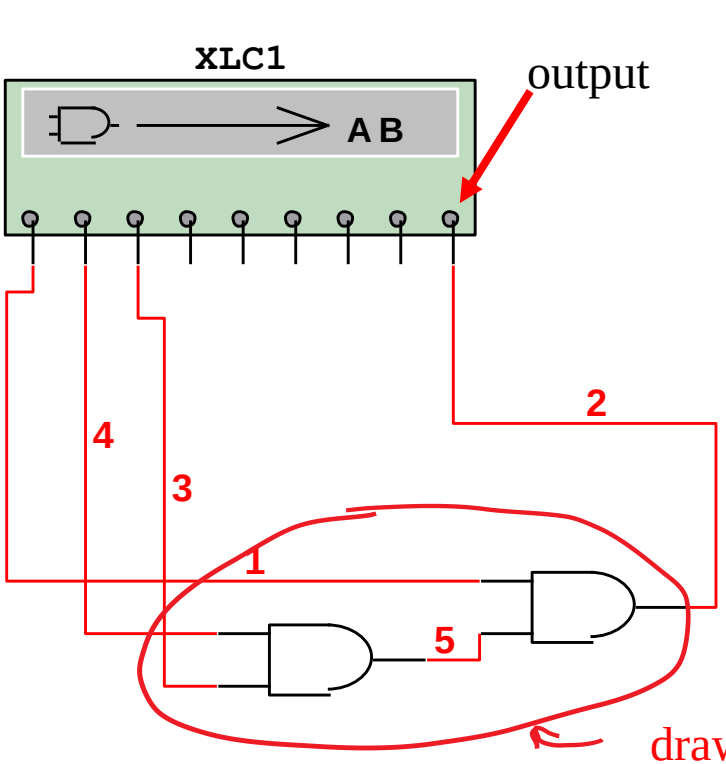
1) Basic group → Switch family → SPDT
2) TTL group → TTLSTD family → 7408N
   (Pick either A, B, C, or D; there are 4 gates on the IC)
   NOTE: For Lecture course, use "Misc Digital"
3) Sources group → Sources → Vcc
4) Sources group → Sources → use DGND (digital ground; 0v)
5) Indicators group → Probe → probe

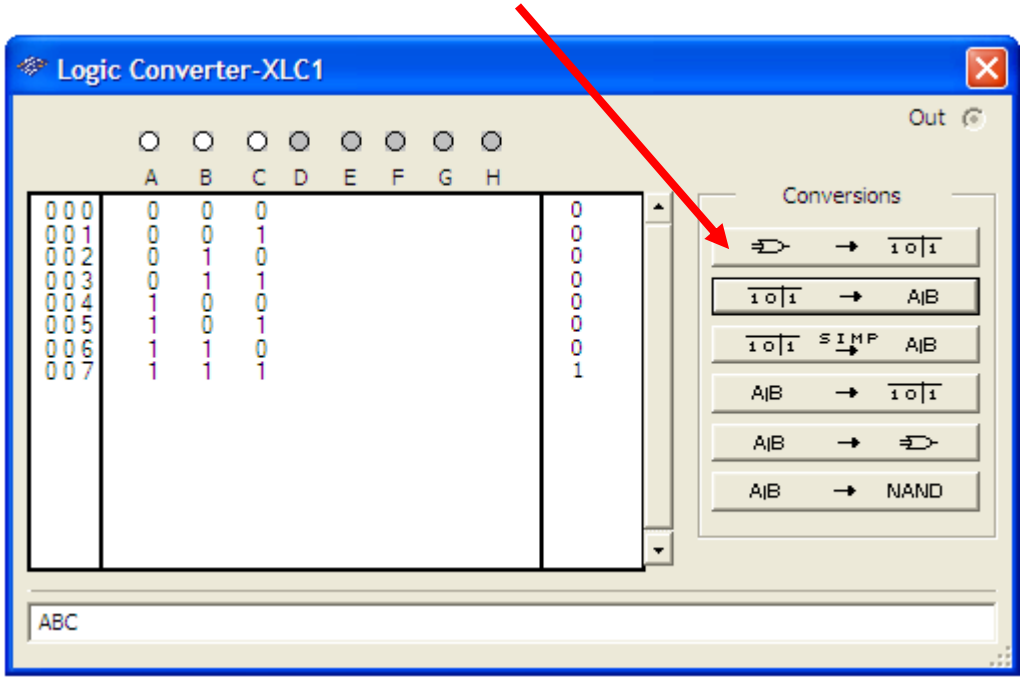
Use!?

Note: to connect wires: click once at start point, then click once at endpoint; don't drag.
7404N → NOT gate; 7432N → OR gate; 7400N → NAND gate

MultiSim – Logic Converter

Click to generate truth table (t.t.)

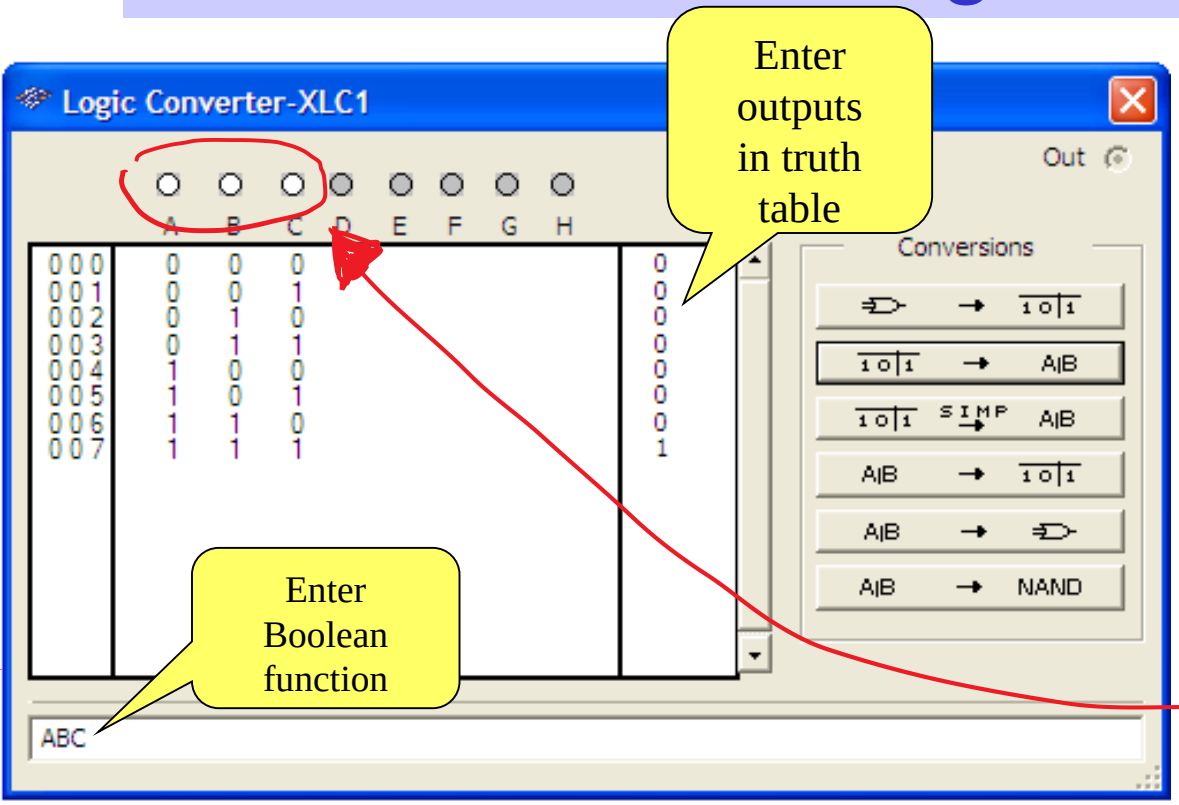




draw any circuit and software will generate t.t. and function

- **Approach #1** -- You can create any circuit with any type of gates (up to 8 inputs; single output) and connect it to the Multisim Logic Converter, and you can automatically create the truth table, the Boolean function (SOM), the minimized Boolean function, and you can generate equivalent circuit with NAND gates

MultiSim – Logic Converter



Experiment!!!!!

→generate truth table
→ truth table to fcn
→truth table to min fcn
→Fcn to truth table
→Fcn to circuit (AND, OR, NOT)
→Fcn to circuit (NAND gates)

circuits do not include switches

click here to setup # of variables

Approach #2 – you can enter a Boolean function directly and get truth table, minimized function, SOM, logic circuits (AND, OR, NOT or NAND only).

← sum of minterms

Approach #3 – you can enter truth table directly (set output to 1, 0, or don't cares (X)) and get Boolean function (SOM or minimized), logic circuits

Further Reading

- Mano Kime, Logic and Computer Design Fundamentals, Prentice Hall, 2001, Chapter 2.

- Tocci R., Digital Systems, Prentice Hall, Chapter 3&4.
- www.howstuffworks.com
  "How Boolean Logic Works"

- www.play-hookey.com (digital section)

IEEE Spectrum Magazine

- Join IEEE
- Become member of Abington Student Chapter
- We can elect officers
- Visit IEEE Spectrum Magazine web site
  http://spectrum.ieee.org/
  - Review articles on professional and industrial developments in EE and CMPEN
  - Includes salary and job trends articles