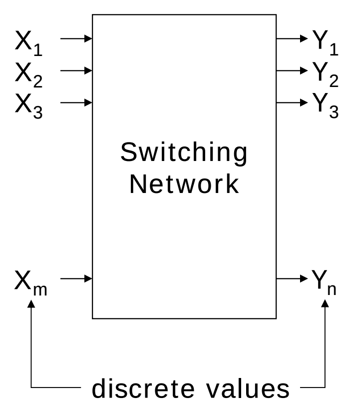
ECE380 Digital Logic

Introduction to Logic Circuits:
Variables, functions, truth tables,
gates and networks

Logic circuits

- Logic circuits perform operations on digital signals
  - Implemented as electronic circuits where signal values are restricted to a few discrete values
- In ***binary*** logic circuits there are only two values, 0 and 1
- The general form of a logic circuit is a switching network



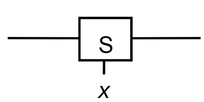
Boolean algebra

- Direct application to switching networks
  - Work with 2-state devices → 2-valued Boolean algebra (switching algebra)
  - Use a Boolean variable (X, Y, etc.) to represent an input or output of a switching network
  - Variable may take on only two values (0, 1)
  - X=0, X=1
  - These symbols are <u>not</u> binary numbers, they simply represent the 2 states of a Boolean variable
  - They are <u>not</u> voltage levels, although they commonly refer to the low or high voltage input/output of some circuit element

Variables and functions

- The simplest binary element is a switch that has two states
- If the switch is controlled by *x*, we say the switch is open if *x* =0 and closed if *x* =1

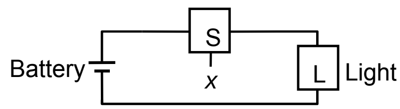
(a) Two states of a switch



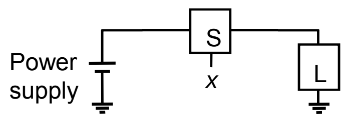
(b) Symbol for a switch

Variables and functions

- Assume the switch controls a lightbulb as shown
  - The output is defined as the state of the light L
    - If the light is on -> L=1
    - If the light is off -> L=0
- The state of L, as function of x is
  - $L(x)=x$
- $L(x)$ is a **logic function**
- x is an **input variable**



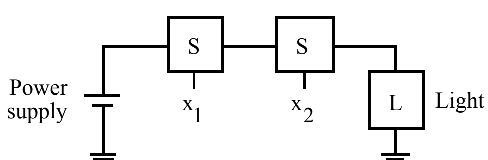
(a) Simple connection to a battery



(b) Using a ground connection as the return path

Variables and functions (AND)

- Consider the possibility of two switches controlling the state of the light
- Using a series connection, the light will be on only if both switches are closed
  - $L(x_1, x_2) = x_1 \cdot x_2$
  - L=1 iff (if and only if) x_1 AND x_2 are 1



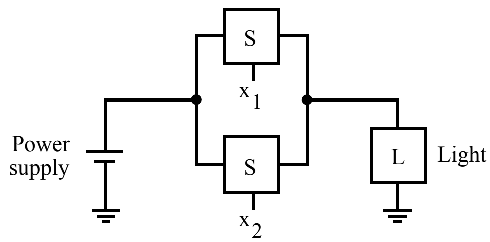
The logical AND function (series connection)

"·" AND operator
$x_1 \cdot x_2 = x_1 x_2$

The circuit implements a logical **AND** function

Variables and functions (OR)

- Using a parallel connection, the light will be on only if either or both switches are closed
  - $L(x_1, x_2) = x_1 + x_2$
  - L=1 if x_1 OR x_2 is 1 (or both)



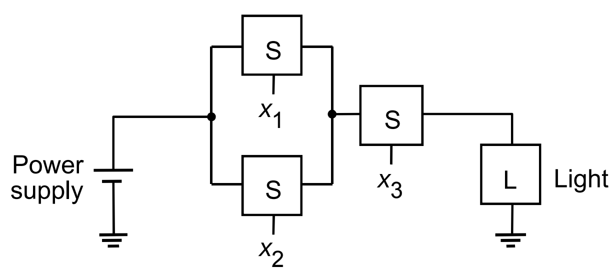
The logical OR function (parallel connection)

"+" OR operator

The circuit implements a logical **OR** function

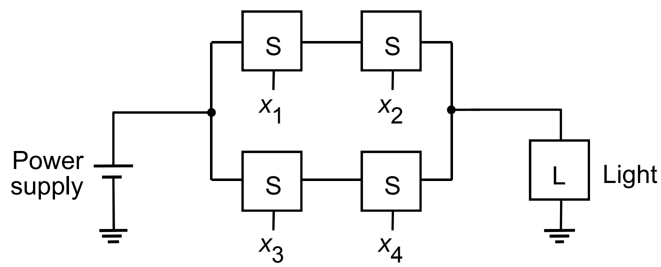
Variables and functions

- Various series-parallel connections would realize various logic functions
  - $L(x_1, x_2, x_3) = (x_1 + x_2) \cdot x_3$



Variables and functions

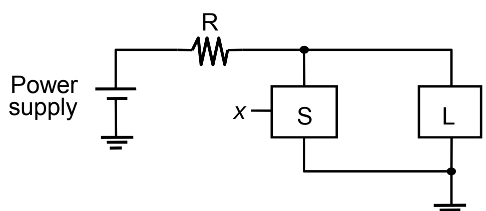
- What would the following logic function look like if implemented via switches?
  - $L(x_1, x_2, x_3, x_4) = (x_1 \cdot x_2) + (x_3 \cdot x_4)$



Electrical & Computer Engineering — Dr. D. J. Jackson Lecture 2-9

Inversion

- Before, actions occur when a switch is closed. What about the possibility of an action occurring when a switch is opened?
  - $L(x) = \bar{x}$
  - Where L=1 if x=0 and L=0 if x=1
- L(x) is the inverse (or complement) of x



$\bar{x}$, x', NOT x

The circuit implements a logical **NOT** function

Electrical & Computer Engineering — Dr. D. J. Jackson Lecture 2-10

Inversion of a function

- If a function is defined as
  - $f(x_1, x_2) = x_1 + x_2$
- Then the complement of *f* is
  - $\overline{f}(x_1, x_2) = \overline{x_1 + x_2} = (x_1 + x_2)'$
- Similarily, if
  - $f(x_1, x_2) = x_1 \cdot x_2$
- Then the complement of *f* is
  - $\overline{f}(x_1, x_2) = \overline{x_1 \cdot x_2} = (x_1 \cdot x_2)'$

Truth tables

- Tabular listing that fully describes a logic function
  - Output value for all input combinations (valuations)

x_1	x_2	$x_1 \cdot x_2$
0	0	0
0	1	0
1	0	0
1	1	1

AND

x_1	x_2	$x_1 + x_2$
0	0	0
0	1	1
1	0	1
1	1	1

OR

x_1	x_1'
0	1
1	0

NOT

Truth tables

- Truth table for AND and OR functions of three variables

x_1	x_2	x_3	$x_1 \cdot x_2 \cdot x_3$	$x_1 + x_2 + x_3$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Truth tables of functions

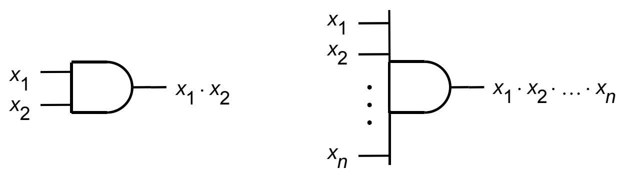
- If L(x,y,z)=x+yz, then the truth table for L is:

+

x	y	z	yz	x+yz
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

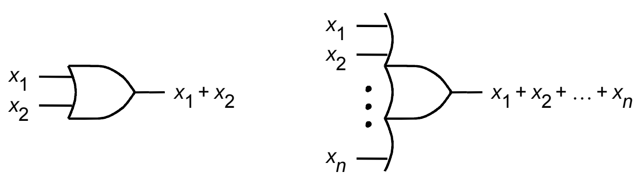
Logic gates and networks

- Each basic logic operation (AND, OR, NOT) can be implemented resulting in a circuit element called a ***logic gate***
- A logic gate has one or more inputs and one output that is a function of its inputs

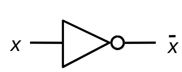


AND gates

Logic gates and networks



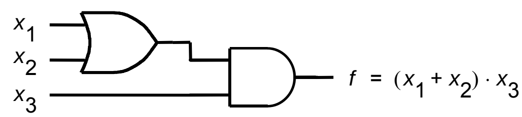
OR gates



NOT gate

Logic gates and networks

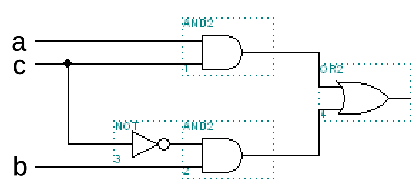
- A larger circuit is implemented by a network of gates
  - Called a logic network or logic circuit



Logic gates and networks

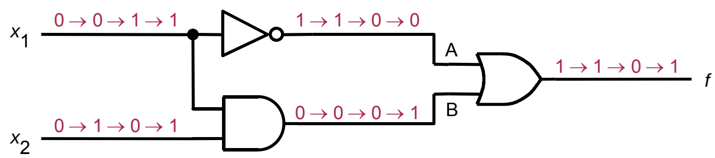
- Draw the truth table and the logic circuit for the following function
  - F(a,b,c) = ac+bc'

a	b	c	ac	bc'	ac+bc'
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	1	0	1
1	1	0	0	1	1
1	1	1	1	0	1



Analysis of a logic network

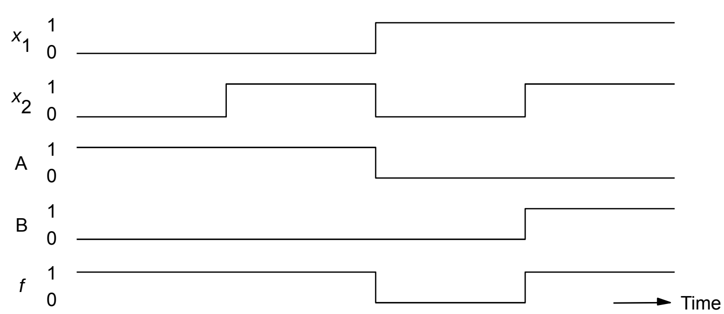
- To determine the functional behavior of a logic network, we can apply all possible input signals to it



Network that implements $f = \bar{x}_1 + x_1 \cdot x_2$

Analysis of a logic network

- The function of a logic network can also be described by a timing diagram (gives dynamic behavior of the network)



Timing diagram