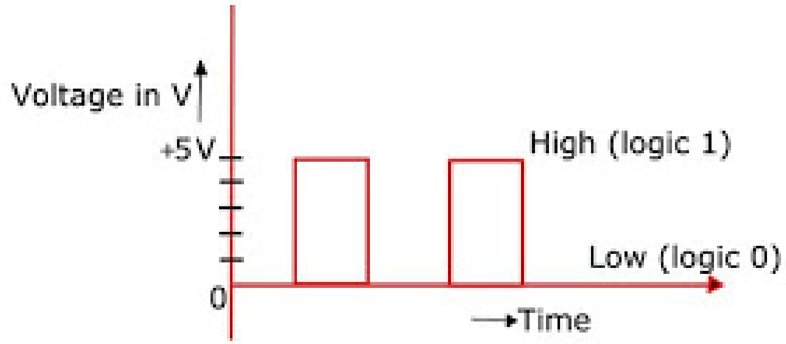
Digital

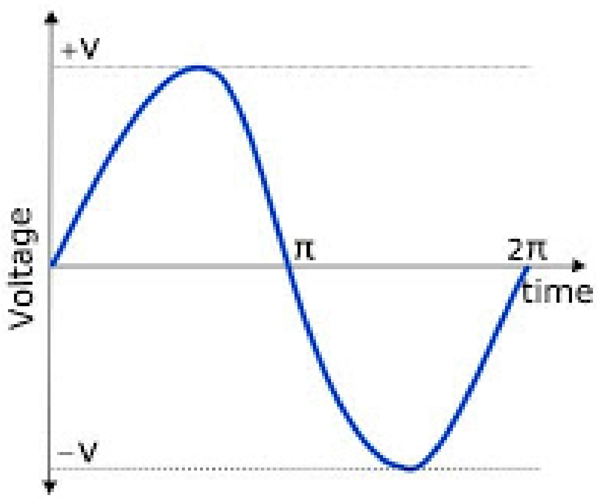
- Electronic circuits can be divided into:
  - digital
  - Analog
- *Many applications require both; and interfacing between analog and digital is important.*

Digital Vs Analog

- Digital
  - The term *digital is derived from the way operations* are performed, by counting digits.
  - Digital electronics involves quantities with discrete values
  - used in electronic devices



- Analog
  - Analog electronics involves quantities with continuous values.
  - Most things that can be measured quantitatively occur in nature in analog form.
  - For example,
    - the air temperature changes over a continuous range of values.



Advantages of digital

- digital data can be processed and transmitted more efficiently and reliably than analog data.
- digital data has a great advantage when storage is necessary.
  - For example,
    - music when converted to digital form can be stored more compactly and reproduced with greater accuracy and clarity than is possible when it is in analog form.
- Noise (unwanted voltage fluctuations) does not affect digital data nearly as much as it does analog signals.

Digital Electronics

- Digital electronics involves circuits and systems in which there are only two possible states.
  - These states are represented by two different voltage levels:
    - HIGH
    - LOW.
  - The two states can also be represented by
    - current levels,
    - bits and bumps on a CD or DVD,
    - etc.
- In digital systems such as computers,
  - combinations of the two states are called *codes,*
    - used to represent numbers, symbols, alphabetic characters, and other types of information.
- The two-state number system is called *binary,*
  - *two digits are 0 and 1.*
  - *A binary digit* is called a *bit.*

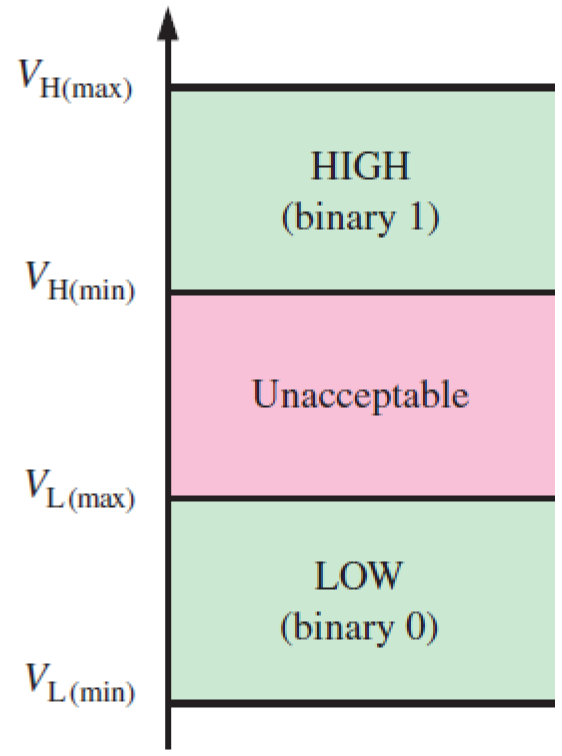
Binary Digits

- In digital circuits,
  - two different voltage levels are used to represent the two bits.
    - Generally,
      - 1 is represented by the higher voltage(HIGH),
      - 0 is represented by the lower voltage level(LOW), which we will refer to as a
      - This is called **positive logic.**
    - Sometimes,
      - 1 is represented by a LOW
      - 0 is represented by a HIGH
      - This is called **negative logic.**

  - Groups of bits (combinations of 1s and 0s), called *codes, are used to represent numbers,* letters, symbols, instructions, and anything else required in a given application

Logic Levels

- The voltages used to represent a 1 and a 0 are called *logic levels.*
- *Ideally,*
  - *one voltage level* represents a HIGH
  - another voltage level represents a LOW.
- In a practical digital circuit,
  - a HIGH can be any voltage between a specified minimum value and a specified maximum value.
  - LOW can be any voltage between a specified minimum and a specified maximum.
  - There can be no overlap between the accepted range of HIGH levels and the accepted range of LOW levels.

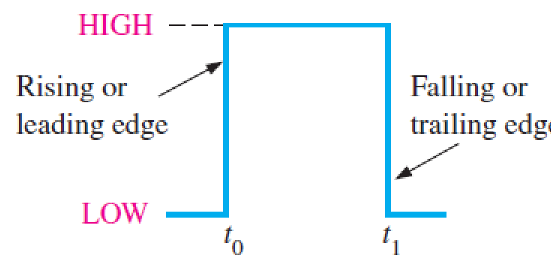


- For example, for a certain type of digital circuit technology called CMOS
  - HIGH input values may range from 2 V to 3.3 V
  - LOW input values may range from 0 V to 0.8 V.
  - If a voltage of 2.5 V is applied, the circuit will accept it as a HIGH or binary 1.
  - If a voltage of 0.5 V is applied, the circuit will accept it as a LOW or binary 0.
  - For this type of circuit, voltages between 0.8 V and 2 V are unacceptable.

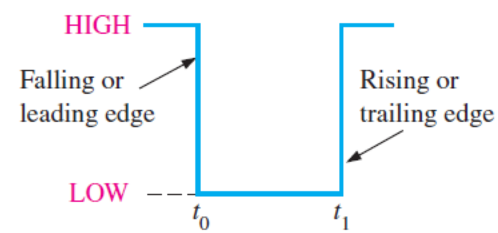
Pulse

- single positive-going pulse
  - is generated when the voltage (or current) goes from its normally LOW level to its HIGH level and then back to its LOW level.
- Single negative-going pulse
  - is generated when the voltage (or current) goes from its normally HIGH level to its LOW level and back to its HIGH level.
- Pulse has two edges:
  - a **leading edge** that occurs first at time t0
  - a **trailing edge** that occurs last at time t1.
- Eg: For a positive-going pulse
  - leading edge is a rising edge,
  - trailing edge is a falling edge

Positive-going ideal pulse



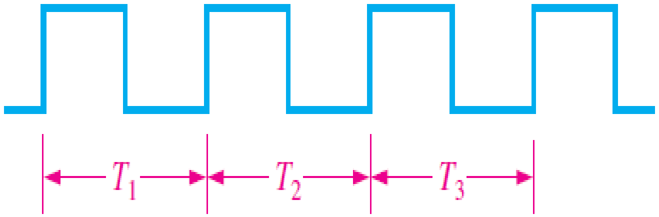
Negative-going ideal pulse



In practice, these transitions never occur instantaneously, although for most digital work you can assume ideal pulses(the rising and falling edges are assumed to change instantaneously).

Digital Waveform

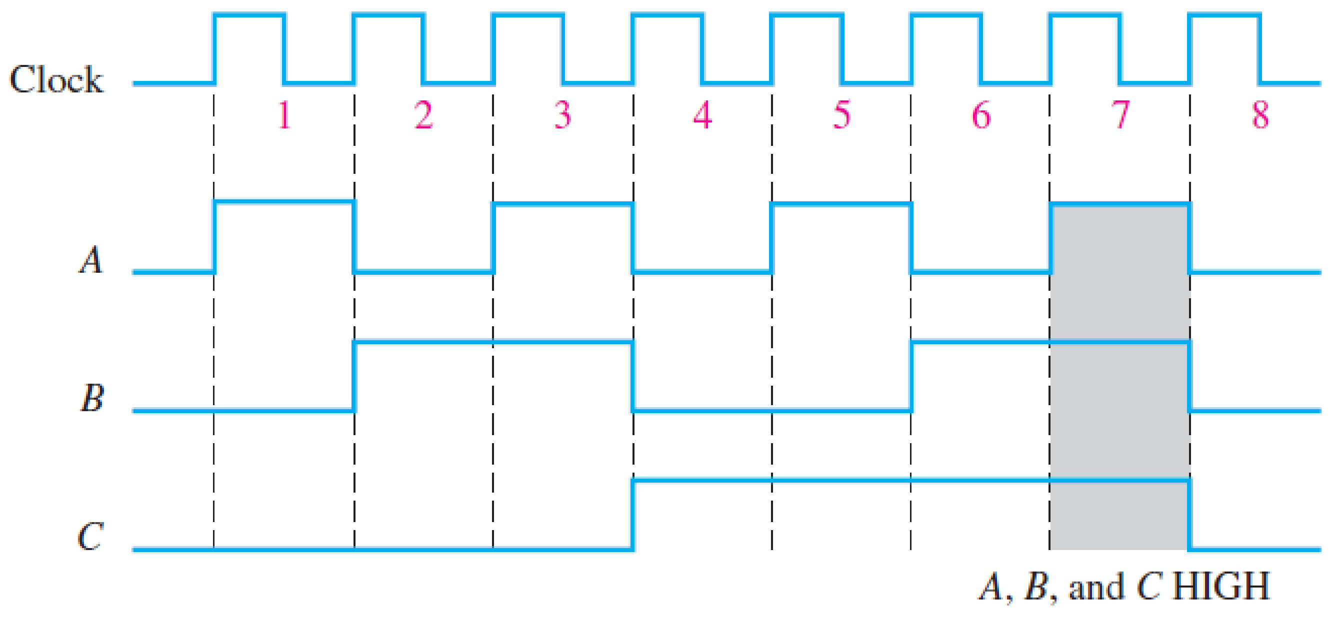
- Digital waveforms consist of voltage levels that are changing back and forth between the HIGH and LOW levels or states.
- A digital waveform is made up of a series of pulses sometimes called pulse trains
- Can be classified as:
  - periodic
    - Waveform repeats itself at a fixed interval, called a period (T).



  - nonperiodic.
    - Waveform does not repeat itself at fixed intervals and may be composed of pulses of randomly differing pulse widths and/or randomly differing time intervals between the pulses.

Timing Diagrams

- A **timing diagram is a graph of digital waveforms showing the actual time relationship of** two or more waveforms and how each waveform changes in relation to the others.
- can determine the states (HIGH or LOW) of all the waveforms at any specified point in time and the exact time that a waveform changes state relative to the other waveforms.



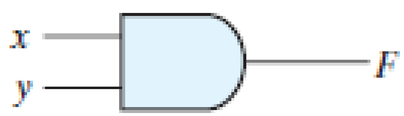
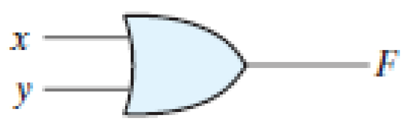
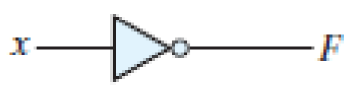
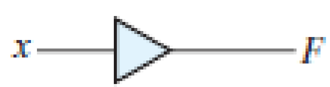
Truth Table

- Table which shows the output for each possible input in terms of levels and corresponding bits. A table such as this is called a **truth table.**

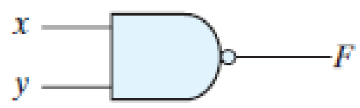
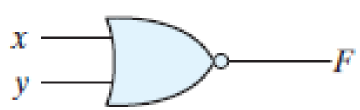
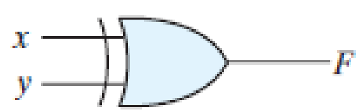
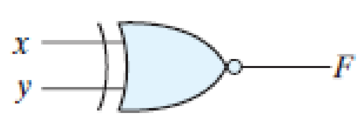
Logic Gates

- Logic gates are electronic circuits that operate on one or more input signals to produce an output signal.
- **Types:**
  - Basic Gates (AND,OR,NOT)
    - Together they can perform any operation
  - Universal Gates (NAND,NOR)
    - Each gate can perform any operation.
  - Exclusive Gates (XOR,XNOR)
    - Complex function constructed using basic gates.
    - Used extensively in building arithmetic logic circuits, computational logic comparators and error detection circuits

Logic Gates

Name	Graphic symbol	Algebraic function	Truth table
AND		$F = x \cdot y$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = x + y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$F = x'$	x | F 0 | 1 1 | 0
Buffer		$F = x$	x | F 0 | 0 1 | 1

Logic Gates

Name	Graphic symbol	Algebraic function	Truth table
NAND		$F = (xy)'$	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = (x + y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1