***Binary Arithmetic***

**Binary Addition**

The four basic rules for adding binary digits (bits) are as follows:

0+0=0 Sum of 0 with a carry 0

0+1=1 Sum of 1 with a carry 0

1+0=1 Sum of 1 with a carry 0

1+1=1 0 Sum of 0 with a carry 1

***Examples:***

110 6 111 7 1111 15

+

+

+

+

+

100 4 011 3 1100 12

\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_

1010 10 1010 10 11011 27

**Binary Subtraction**

The four basic rules for subtracting are as follows:

0-0=0

1-1=0

1-0=1

0-1=1 with a borrow of 1

***Examples:***

11 3 11 3 101

-

-

-

-

-

01 1 10 2 011

\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_

10 2 01 1 010

5 110 6 101101 45

-

-

-

-

-

3 101 5 001110 14

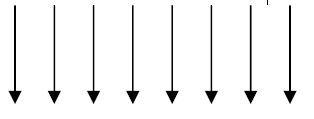
\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_

2 001 1 011111 31

**1's and 2's Complement of Binary Number**

The 1's complement and the 2's complement of binary number are important because they permit the ***representation of negative numbers.***

Binary Number **1 0 1 1 0 0 1 0**



1'sComplement **0 1 0 0 1 1 0 1**

2's Complement of a binary number is found by adding 1 to the LSB of the 1's Complement.

2's Complement= (1's Complement) +1

Binary number 10110010

1'scomplement 01001101

Add 1 + 1

----------------------------------------

2's complement 01001110

**Subtraction Using 2's complement Method**

Subtraction is a special case of addition. For example, subtracting +6 (the subtrahend) from +9 (the minuend) is equivalent to adding -6 to +9. Basically, the subtraction operation changes the sign of the subtrahend and adds it to the minuend. The result of a subtraction is called the difference.

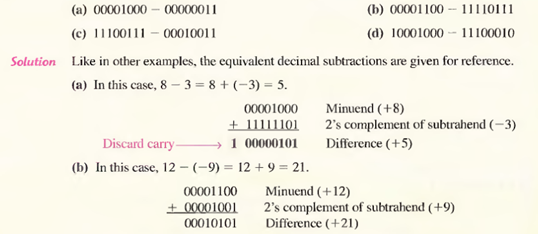
**The sign of a positive or negative binary number is changed by taking its 2's**

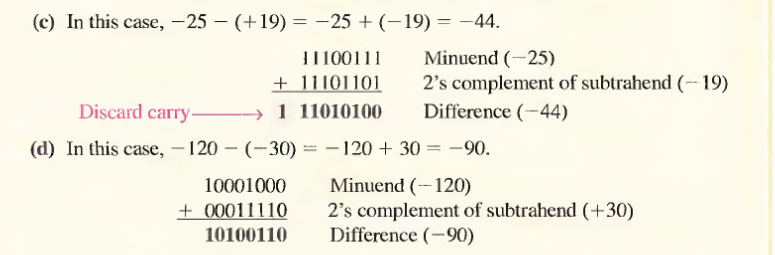
**Complement.**

For example, when you take the 2's complement of the positive number (00000100)2 (+4), you get (11111100)2.

Example

Using 2's complement method, perform each of the following subtractions of the signed numbers:



****

**BINARY DIGITS, LOGIC LEVELS, AND DIGITAL WAVEFORMS**

1. **BINARY DIGITS**

Digital electronics involves circuits and systems in which there are only ***two possible States.***

These states are represented by two different voltage levels: A HIGH and a LOW. The two states can **also** be represented by current levels, bits and bumps on a CD or DVD. Etc.

Each of the ***two digits*** in the binary system, 1 and 0, is called ***a bit,*** which is a contraction of the words **binary digit**. In digital circuits. Two different voltage levels are used to represent the two bits. Generally, (1) is represented by the higher voltage, which we will refer to as a HIGH, and a (0) is represented by the lower voltage level, which we will refer to as a LOW. This is called ***positive logic*** and will be used.

**HIGH = 1 and LOW = 0**

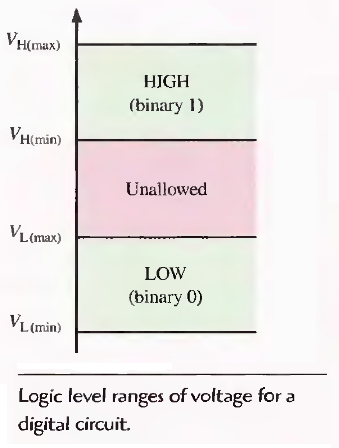
Another system in which a (1) is represented by a LOW and a (0) is represented by a HIGH is called ***negative logic***.

**HIGH = 0 and LOW = 1**

The combinations of the two states, called ***codes***, are used to represent numbers, symbols, alphabetic characters, and other types of information. The two-state number system is called ***binary,*** and its two digits are (0) and (1).

1. **LOGIC LEVELS**

Ideally, one voltage level represents a HIGH and another voltage level represents a LOW. In a practical digital circuit, however, a HIGH can be any voltage between a specified minimum value and a specified maximum value. Likewise, a 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. Figure 3-1, illustrates the general range of LOWs and HIGHs for a digital circuit.



**Positive Logic**

1. **DIGITAL WAVEFORMS**

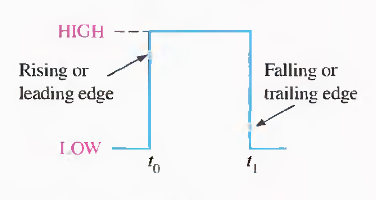
Digital waveforms consist of voltage levels that are changing back and forth between the HIGH and LOW levels. A digital waveform is made up of a series of ***pulses***.

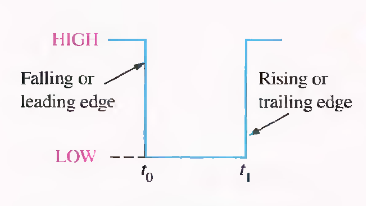
.

**Pulse**

**Positive-going pulse**

**Negative-going pulse**

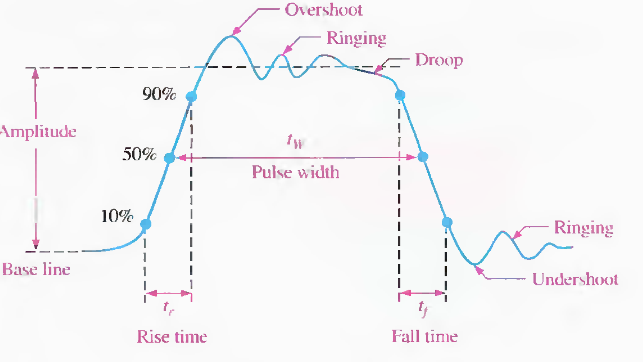




**Normal Level**

**Normal Level**

The pulses in Figure above are **ideal** because the rising and falling edges are assumed to change in **zero time** (instantaneously). In practice, these transitions never occur instantaneously, **although for most digital work you can assume ideal pulses.**

****

****

**Waveform Characteristics**

Most Waveform encountered in digital systems are composed of series of pulses, sometimes called pulse trains.

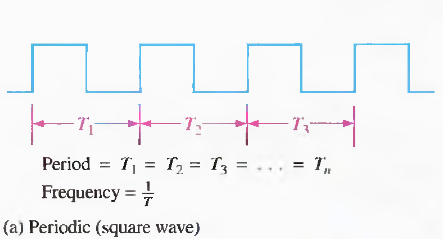
**Non periodic**

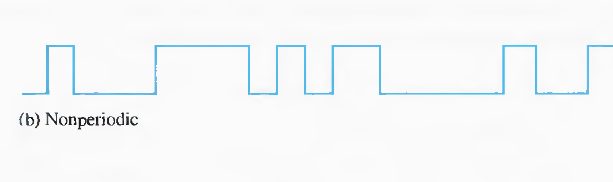
**Periodic**

**Waveform**

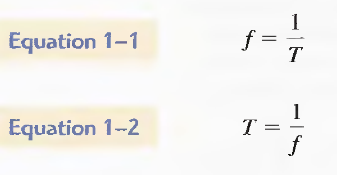
A ***periodic*** pulse waveform is one that repeats itself at a fixed interval, called a **period (T).** The **frequency (f)** is the rate at which it repeats itself and is measured in hertz (Hz).

A ***non-periodic*** pulse waveform, of course, 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. An example of each type is shown in Figure.

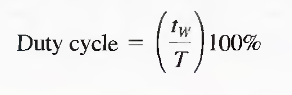




The frequency (f) of a pulse (digital) waveform is the reciprocal of the period. The relationship between frequency and period is expressed as follows:



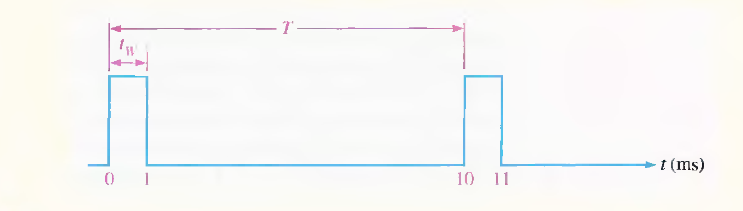
An important characteristic of a periodic digital waveform is its **duty cycle**, which is the ratio of the pulse width (tW) to the period (T). It can be expressed as a percentage.

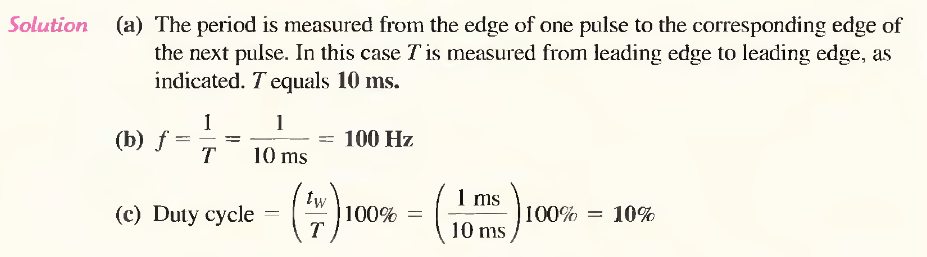


**Example**

A portion of a periodic digital waveform is shown in Figure. The measurements are in milliseconds. Determine the following:

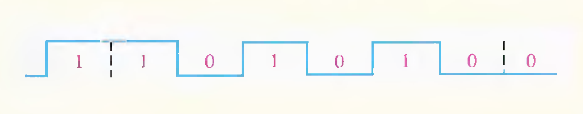
(a) Period (b) frequency (c) duty cycle





**Digital Waveform Carries Binary Information**

Binary information that is handled by digital systems appears as waveforms that represent sequences of bits. When the waveform is HIGH, a binary 1 is present; when the waveform is LOW, a binary 0 is present. Each bit in a sequence occupies a defined time interval called a ***bit time***.

****

**11010100**

**Bit Time**

**Bit Time**

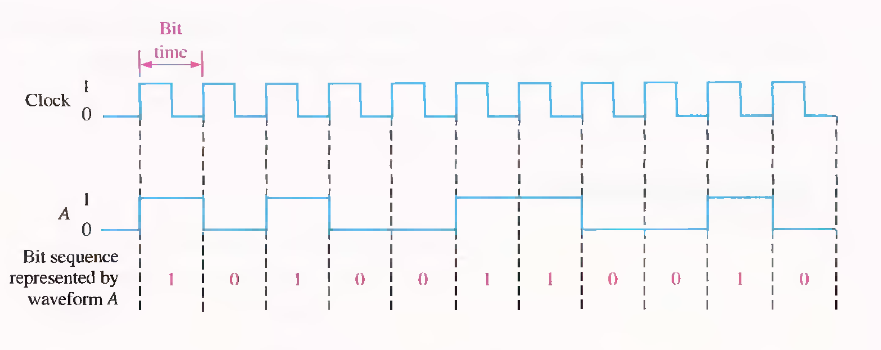
**Bit Time**

**Bit Time**

**The Clock**

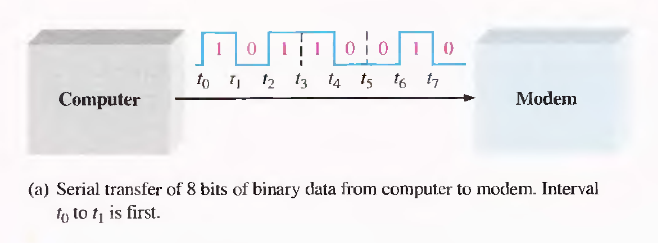
The clock is a periodic waveform in which each interval between pulses (the period) equals the time for ***one bit.*** In digital systems, all waveforms are synchronized with a clock.

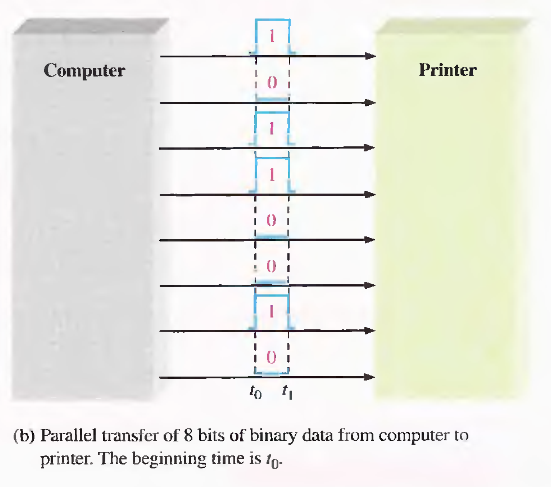
An example of a clock waveform is shown in Figure. Notice that, in this case, each change in level of waveform A occurs at the ***leading edge*** of the clock waveform. In other cases, level changes occur at the ***trailing edge*** of the clock. During each bit time of the clock, waveform A is either HIGH or LOW. These HIGHs and LOWs represent a sequence of bits, as indicated. A group of several bits can be used as a piece of binary information, such as a number or a letter. ***The clock waveform itself does not carry information.***

******

**Data Transfer**

For example, numbers stored in binary form in the memory of a computer must be transferred to the computer's central processing unit in order to be added. The sum of the addition must then be transferred to a monitor for display and/or transferred back to the memory. In computer systems, as illustrated in Figure, binary data are transferred in two ways: serial and parallel.





When bits are transferred in serial from one point to another. They are sent one bit at a time along a single line. As illustrated in Figure above.

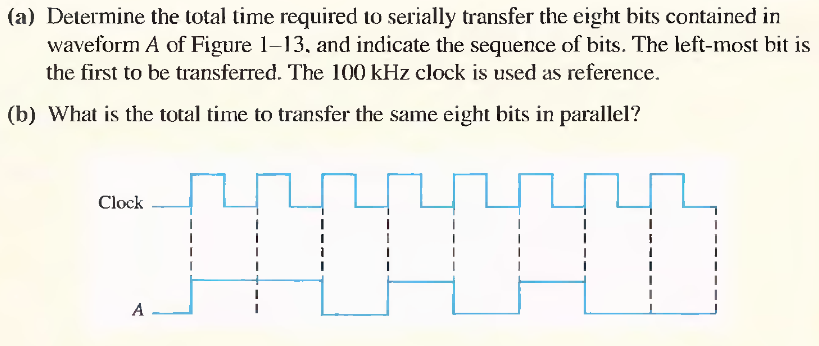
For the case of a computer-to- modem transfer. During the time interval from t0 to t1" the first bit is transferred. During the time interval from t1 to t2, the second bit is transferred, and so on. To transfer eight bits in series, it takes eight time intervals.

When bits are transferred in parallel form, all the bits in a group are sent out on separate lines at the **same time**. There is one line for each bit, as shown in Figure above for the example of eight bits being transferred from a computer to a printer. To transfer eight bits in parallel, it takes one time interval compared to eight time intervals for the serial transfer.

To summarize, an advantage of serial transfer of binary data is that a minimum of only one line is required. In parallel transfer, a number of lines equal to the number of bits to be transferred at one time is required.

A disadvantage of serial transfer is that it takes longer to transfer a given number of bits than with parallel transfer. For example, if one bit can be transferred in 1 µs. then it takes 8 µs to serially transfer eight bits but only 1µs to parallel transfer eight bits. A disadvantage of parallel transfer is that it takes more lines than serial transfer.

**Example**

****

