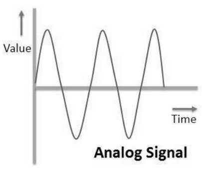
Digital electronics

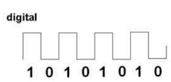
Introduction:- digital electronics is the foundation of digital computers and most of automated control systems. Digital electronics is have a great impact on modern society. Nowadays, we are using calculators, computers, watches, communication system etc.

Analog signal:- analogue signal is defined as “voltage or current whose size is proportional to the quantity is represents”. It is a continuous signal and has infinite set of possible values.



the example of analogue signals are sound, temperature, pressure, velocity, Sinosidal wave form etc

Digital signal:- digital signals are discrete in nature. A digital signals can have fixed number of values. Mostly digital signals have only two value that is 1 and 0. A value 1 represents a high Signals and a value 0 represents a low signals.



A digital signal represent a

Logic 1 = high level

Logic 0 = low level

Analog system:- the system which processes and analog signal is known as analogue system. In an analog system, the physical quantities can vary over a continuous range of value. these example of analog systems are telephone system and Magnetic tape recording etc.

Digital system:- the system which processes a digital signal is known as digital system. the digital signals can represent in discrete value. the example of digital systems are digital watches, calculators, computers, counters etc.

Difference between analog and digital signals:-

Analog signal.

1. analog signals are continuous in nature and can have infinite number of values.
2. Basic analog signal is represented by sine wave.
3. Analog signals are stored in the form of wave signals.
4. Analog signal transmission requires less bandwidth.
5. Processing and transmission of analogue signals is more prone to noise.

Digital signal:-

1. Digital signals are discrete in nature and can have fixed number of value.
2. Basic digital signal is represented by square wave.
3. Digital signals are stored in the form of binary bits.
4. Digital signal transmission requiress large bandwidth.
5. processing and transmission of digital signals is less prone to noise.

advantages of digital system over analogue systems:-

1. **Flexibility:** digital system hardware are very flexible as compared to analog system hardware.
2. **Effect of noise:** In digital systems, effect of noise is very less as compared to in analog system.
3. **Reliability:** digital system are more reliable as compared to analog system.
4. **Storage:** it is very easy, reliable and compact to store information in digital form as compared to in analog form.
5. **Observational error:** Analog instruments are prone to observational error while digital instruments are free from observational error like parallax and approximation error.

6. **Design:** with the advancement of digital technology and availability of large variety IC's, it is very easy to design a complex logic in digital system as compared to in analog system.

6.Easy to use: The digital systems are easier to use because they directly display data in alpha-numeric form on the screen.

Disadvantages of digital system over analog system:-

1. **Bandwidth:** Digital systems required large bandwidth as compared to analog systems.
2. **Quantization noise:** Quantization noise is available in digital systems which is not present in analog systems.

3.Complexity: Digital systems are more complex as compared to analog systems.

Applications of digital systems:

digital systems are widely used almost every sphere of life. some of the application are:

1. Data base management system used banks, offices, institutes, shop etc using computers.
2. process monitoring and control system in industries using computers, PLC's robots.
3. Digital Signal Processing and digital communication.

4. Entertainment appliance like CD/DVD players, LED TVs, digital cameras.
5. appliances like photostat machine, fax machine, EPBAX machines, microwaves ovens.
6. medical instruments like digital x-ray machines, ultrasound machines, ECG machines.
7. Combustion control in modern vehicles.

Definition - What does Binary Number System mean?
The binary number system is a numbering system that represents numeric values using two unique digits (0 and 1). Mosting computing devices use binary numbering to represent electronic circuit voltage state, (i.e., on/off switch), which considers 0 voltage input as off and 1 input as on.

This is also known as the base-2 number system, or the binary numbering system.

Decimals

The decimal numeral system is also known as base 10 since it has ten as its base. Decimal notation relates to the base 10 positional notation like the Hindu-Arabic numeral system. The decimal number contains a decimal point.

Now let’s see an example

Here is the number “thirty-four and seven-tenths” written as decimal number:

The decimal point goes between Ones and Tenths

34.7 has 3 Tens, 4 Ones and 7 Tenths, like this:

A decimal is defined as a number expressed in decimal notation and generally applied to values that have a fractional part and separated from the integer side by a decimal separator.

In decimal number system, the decimal can be a terminating one that has a finite fractional value(e.g. 12.500); a repeating decimal that has a non-terminating fractional value consisting of repeating stream of digits(e.g. Value of pi). Decimal fractions have terminating decimal expansion, whereas irrational numbers consist of infinite non-repeating decimal expansion.

Place Value

When you write specific numbers, the position of each digit is important.

Example:

For instance, let's consider a number 456.

The position of "6" is in Ones place, which means 6 ones (i.e. 6).

The position of "5" is in the Tens place, which means 5 tens(i.e. fifty).

The position of "4" is in the Hundreds place, which means 4 hundred.

As we go left, each position becomes ten times greater.

Hence, we read it as "Four hundred fifty-six".

As we move left, each position is 10 times bigger!

Tens are 10 times bigger than Ones.

Hundreds are 10 times bigger than Tens.

And

Each time we move right every position becomes 10 times smaller

From Hundreds, to Tens, to Ones

But if we continue past Ones ?

What is 10 times smaller than Ones?

110ths110ths (Tenths) are!

Before that we should first put a decimal point,

So we already know that where we put that decimal point.
We say the above example as four hundred and fifty-six and eight-tenths but

We usually just say four hundred and fifty-six point eight.

Types of Decimal Numbers:

Decimal Numbers may be of different kinds:

Recurring Decimal Numbers (Repeating or Non-Terminating Decimals)

Example-

3.125125 (Finite)

3.121212121212….. (Infinite)

&;

Non Recurring Decimal Numbers (Non Repeating or Terminating Decimals):

Example:

3.2376 (Finite)

3.137654….(Infinite)

&;

Decimal Fraction:

It represents the fraction whose denominator in powers of ten.

Example:

81.75 = 8175/100

32.425 = 32425/1000

Converting the Decimal Number into Decimal Fraction:

For the decimal point place a “1” in the denominator and remove the decimal point.

“1” is followed by a number of zeros equal to the number of digits following the decimal point.

For Example:

8 1 . 7 5

↓ ↓ ↓

1 0 0

81.75 = 8175/100

&;

8 represents the power of 10^1 that is the tenths position.

1 represents the power of 10^0 that is the units position.

7 represents the power of 10^(-1) that is the one-tenths position.

5 represents the power of 10^(-2) that is the one-hundredths position.

So that is how each digit is represented by a particular power of ten in the decimal number.

&;

Place Value of Decimal Numbers:

The place value is obtained by multiplication of the digit in the decimal number with its power of ten that the digit holds at its position.

The power of ten can be found using the following Place Value Chart:

The digits to the left of the decimal point are multiplied with the positive powers of ten in an increasing order from right to left.

The digits to the right of the decimal point are multiplied with the negative powers of 10 in an increasing order from left to right.

Following the same example 81.75

The decimal expansion of this is :

{(8*10)+(1*1)} + {(7*0.1)+(5*0.01)}

Where each number is multiplied by its associated power of ten.

How to Convert from Binary to Decimal

The binary system is the internal language of electronic computers. If you are a serious computer programmer, you should understand how to convert from binary to decimal. This wikiHow will show you how to do this.

1

Write down the binary number and list the powers of 2 from right to left. Let's say we want to convert the binary number 100110112 to decimal. First, write it down. Then, write down the powers of two from right to left. Start at 20, evaluating it as "1". Increment the exponent by one for each power. Stop when the amount of elements in the list is equal to the amount of digits in the binary number. The example number, 10011011, has eight digits, so the list, with eight elements, would look like this: 128, 64, 32, 16, 8, 4, 2, 1

2

Write the digits of the binary number below their corresponding powers of two. Now, just write 10011011 below the numbers 128, 64, 32, 16, 8, 4, 2, and 1 so that each binary digit corresponds with its power of two. The "1" to the right of the binary number should correspond with the "1" on the right of the listed powers of two, and so on. You can also write the binary digits above the powers of two, if you prefer it that way. What's important is that they match up.

3

Connect the digits in the binary number with their corresponding powers of two. Draw lines, starting from the right, connecting each consecutive digit of the binary number to the power of two that is next in the list above it. Begin by drawing a line from the first digit of the binary number to the first power of two in the list above it. Then, draw a line from the second digit of the binary number to the second power of two in the list. Continue connecting each digit with its corresponding power of two. This will help you visually see the relationship between the two sets of numbers.

4

Write down the final value of each power of two. Move through each digit of the binary number. If the digit is a 1, write its corresponding power of two below the line, under the digit. If the digit is a 0, write a 0 below the line, under the digit.

Since "1" corresponds with "1", it becomes a "1." Since "2" corresponds with "1," it becomes a "2." Since "4" corresponds with "0," it becomes "0." Since "8" corresponds with "1", it becomes "8," and since "16" corresponds with "1" it becomes "16." "32" corresponds with "0" and becomes "0" and "64" corresponds with "0" and therefore becomes "0" while "128" corresponds with "1" and becomes 128.

5

Add the final values. Now, add up the numbers written below the line. Here's what you do: 128 + 0 + 0 + 16 + 8 + 0 + 2 + 1 = 155. This is the decimal equivalent of the binary number 10011011.

6

Write the answer along with its base subscript. Now, all you have to do is write 15510, to show that you are working with a decimal answer, which must be operating in powers of 10. The more you get used to converting from binary to decimal, the more easy it will be for you to memorize the powers of two, and you'll be able to complete the task more quickly.

7

Use this method to convert a binary number with a decimal point to decimal form. You can use this method even when you want to covert a binary number such as 1.12 to decimal. All you have to do is know that the number on the left side of the decimal is in the units position, like normal, while the number on the right side of the decimal is in the "halves" position, or 1 x (1/2).

The "1" to the left of the decimal point is equal to 20, or 1. The 1 to the right of the decimal is equal to 2-1, or .5. Add up 1 and .5 and you get 1.5, which is 1.12 in decimal notation.

How to Convert from Decimal to Binary

The decimal (base ten) numeral system has ten possible values (0,1,2,3,4,5,6,7,8, or 9) for each place-value. In contrast, the binary (base two) numeral system has two possible values represented as 0 or 1 for each place-value.[1] Since the binary system is the internal language of electronic computers, serious computer programmers should understand how to convert from decimal to binary.

Method One of Two:
Performing Short Division by Two with RemainderEdit

1

Set up the problem. For this example, let's convert the decimal number 15610 to binary. Write the decimal number as the dividend inside an upside-down "long division" symbol. Write the base of the destination system (in our case, "2" for binary) as the divisor outside the curve of the division symbol.

This method is much easier to understand when visualized on paper, and is much easier for beginners, as it relies only on division by two.

To avoid confusion before and after conversion, write the number of the base system that you are working with as a subscript of each number. In this case, the decimal number will have a subscript of 10 and the binary equivalent will have a subscript of 2.

2

Divide. Write the integer answer (quotient) under the long division symbol, and write the remainder (0 or 1) to the right of the dividend.[2]

Since we are dividing by 2, when the dividend is even the binary remainder will be 0, and when the dividend is odd the binary remainder will be 1.

3

Continue to divide until you reach 0.Continue downwards, dividing each new quotient by two and writing the remainders to the right of each dividend. Stop when the quotient is 0.

4

Write out the new, binary number. Starting with the bottom remainder, read the sequence of remainders upwards to the top. For this example, you should have 10011100. This is the binary equivalent of the decimal number 156. Or, written with base subscripts: 15610 = 100111002

This method can be modified to convert from decimal to any base. The divisor is 2 because the desired destination is base 2 (binary). If the desired destination is a different base, replace the 2 in the method with the desired base. For example, if the desired destination is base 9, replace the 2 with 9. The final result will then be in the desired base.

Convert decimal fraction to hexadecimal fraction

BY SCHOOLELECTRONIC · PUBLISHED SEPTEMBER 28, 2013 · UPDATED APRIL 4, 2017

In the earlier post we discussed the integer conversion of a given decimal number to its equivalent hexadecimal number.

The method involved in the conversion of fractional decimal number to fraction hexadecimal number is different, in this post we will only concentrate on the fractional conversion.

Procedure

The steps for the conversion are given below:

Successive multiplication is used to convert a given fractional decimal number to its equivalent hexadecimal fraction.

Here the given decimal fraction is successively multiplied by the base of the target number system (16, here it is hexadecimal system).

During each multiplication iteration, the product generated will have a carry (integer part of the product) and a fractional part.

The carry obtained at each multiplication step becomes a numeral in the hexadecimal fraction.

The fractional part of the product is again multiplied by base 16 in the next step and the process is repeated until the fractional part becomes zero or the number of multiplication iteration equals the number of digits after the decimal point in the given decimal fraction.

Weights are assigned for the carry obtained at each multiplication step in the increasing order starting from the first multiplication step to the last step, such that the carry obtained in the first multiplication iteration is the most significant bit (MSD) after the decimal point and the carry obtained in the last multiplication iteration is the least significant bit (LSD)

This procedure is illustrated in the following example.

Ex1: Convert (0.0628)10 decimal fraction to hexadecimal fraction (?)16 using successive multiplication method

1st Multiplication Iteration

Multiply 0.0628 by 16

0.0628 x 16 = 1.0048(Product)   Fractional part=0.0048   Carry=1   (MSD)

2nd Multiplication Iteration

Multiply 0.0048 by 16
0.0048 x 16 = 0.0768(Product)   Fractional part = 0.0768   Carry = 0

3rd Multiplication Iteration

Multiply 0.0768 by 16
0.0768 x 16 = 1.2288(Product)   Fractional part = 0.2288   Carry = 1

4th Multiplication Iteration

Multiply 0.2288 by 16

0.2288 x 16 = 3.6608(Product)   Fractional part = 0.6608   Carry = 3 (LSD)

Here the fractional part doesn’t become zero but we obtain required number of significant digits after the decimal point. Thus we stop the multiplication iteration and assign the weights to the digits obtained in each multiplication step in the increasing order starting from the 1st multiplication step to last multiplication step.

Carry from the 1st multiplication iteration becomes MSB and carry from 4th iteration becomes LSB after the decimal point.

Hence, the fractional hexadecimal number of the given decimal fraction
(0.0628)10 is (0.1013)16.

‘Octal

The octal numeral system, or oct for short, is the base-8 number system, and uses the digits 0 to 7. Octal numerals can be made from binary numerals by grouping consecutive binary digits into groups of three (starting from the right). For example, the binary representation for decimal 74 is 1001010. Two zeroes can be added at the left: (00)1 001 010, corresponding the octal digits 1 1 2, yielding the octal representation 112.

In the decimal system each decimal place is a power of ten. For example:

$$\mathbf{74}_{10}=\mathbf{7}\times 10^{1}+\mathbf{4}\times 10^{0}$$

In the octal system each place is a power of eight. For example:

$$\mathbf{112}_{8}=\mathbf{1}\times 8^{2}+\mathbf{1}\times 8^{1}+\mathbf{2}\times 8^{0}$$

By performing the calculation above in the familiar decimal system we see why 112 in octal is equal to 64+8+2 = 74 in decimal.
How to Convert from Decimal to Octal

Octal is the base 8 number system, that only uses the digits 0 through 7. Its main advantage is the ease of conversion with binary (base 2), since each digit in octal can be written as a unique three-digit binary number.[1] Converting decimal to octal is a little more difficult, but you don't need to know any math past long division. Start with the division method, which finds each digit by dividing by powers of 8. The remainder method is faster and uses similar math, but it can be a little harder to understand why it works.

Method One of Two:
Converting with DivisionEdit

1

Use this method to learn the concepts. Of the two methods on this page, this method is easier to understand. If you're already confident working in different number systems, try the faster remainder method, below.

2

Write down the decimal number. For this example, we'll convert the decimal number 98 into octal.

3

List the powers of 8. Remember that "decimal" is called base 10 because each digit represents a power of 10. We call the first three digits 1s place, the 10s place, the 100s place — but we could also write this as the 100place, the 101 place, and the 102 place. Octal, or the base 8 number system, uses powers of 8 instead of powers of 10. Write a few of these powers of 8 in a horizontal line, from largest to smallest. Note that these numbers are all written in decimal (base 10):

82 81 80

Rewrite these as single numbers:

64 8 1

You don't need any powers of 8 larger than your original number (in this case, 98). Since 83 = 512, and 512 is larger than 98, we can leave it off the chart.

4

Divide the decimal number by the largest power of eight. Take a look at your decimal number: 98. The nine in the 10s place tells you that there are nine 10s in this number. 10 goes into this number 9 times. Similarly, with octal, we want to know how many "64s" go into the final number. Divide 98 by 64 to find out. The easiest way to do this is to make a chart, reading top to bottom:[2]

98
÷

64 8 1
=

1 ← This is the first digit of your octal number.

5

Find the remainder. Calculate the remainder of the division problem, or the amount left over that doesn't go in evenly. Write your answer at the top of the second column. This is what's left of your number after the first digit is calculated. In our example, 98 ÷ 64 = 1. Since 1 x 64 = 64, the remainder is 98 - 64 = 34. Add this to your chart:

98 34
÷

64 8 1
=

1

6

Divide the remainder by the next power of 8. To find the next digit, we move one step down to the next power of 8. Divide the remainder by this number and fill out your chart's second column:

98 34
÷ ÷

64 8 1
= =

1 4

7

Repeat until you've found the full answer.Just as before, find the remainder of your answer and write it at the top of the next column. Keep dividing and finding the remainder until you've done this for every column, including 80(the ones place). Your final row is the final decimal number converted to octal. Here's our example with the full chart filled out (note that 2 is the remainder of 34÷8):

98 34 2
÷ ÷ ÷

64 8 1
= = =

1 4 2

The final answer: 98 base 10 = 142 base 8. You can write this as 9810 = 1428

8

Check your work. To check your work, multiply each digit in octal by the power of 8 it represents. You should end up with your original number. Let's check our answer, 142:

2 x 80 = 2 x 1 =2

4 x 81 = 4 x 8 = 32

1 x 82 = 1 x 64 = 64

2 + 32 + 64 = 98, the number we started with.

9

Try this practice problem. Practice this method by converting the decimal number 327 into octal. When you think you have the answer, highlight the invisible text below to see the whole problem laid out.

Highlight this area:

327 7 7
÷ ÷ ÷

64 8 1
= = =

5 0 7

The answer is 507.
Definition - What does Binary Number System mean?

The binary number system is a numbering system that represents numeric values using two unique digits (0 and 1). Mosting computing devices use binary numbering to represent electronic circuit voltage state, (i.e., on/off switch), which considers 0 voltage input as off and 1 input as on.

This is also known as the base-2 number system, or the binary numbering system.

Decimals

The decimal numeral system is also known as base 10 since it has ten as its base. Decimal notation relates to the base 10 positional notation like the Hindu-Arabic numeral system. The decimal number contains a decimal point.

Now let’s see an example

Here is the number “thirty-four and seven-tenths” written as decimal number:

The decimal point goes between Ones and Tenths

34.7 has 3 Tens, 4 Ones and 7 Tenths, like this:

A decimal is defined as a number expressed in decimal notation and generally applied to values that have a fractional part and separated from the integer side by a decimal separator.

In decimal number system, the decimal can be a terminating one that has a finite fractional value(e.g. 12.500); a repeating decimal that has a non-terminating fractional value consisting of repeating stream of digits(e.g. Value of pi). Decimal fractions have terminating decimal expansion, whereas irrational numbers consist of infinite non-repeating decimal expansion.

Place Value

When you write specific numbers, the position of each digit is important.

Example:

For instance, let’s consider a number 456.

The position of “6” is in Ones place, which means 6 ones (i.e. 6).

The position of “5” is in the Tens place, which means 5 tens(i.e. fifty).

The position of “4” is in the Hundreds place, which means 4 hundred.

As we go left, each position becomes ten times greater.

Hence, we read it as “Four hundred fifty-six”.

As we move left, each position is 10 times bigger!

Tens are 10 times bigger than Ones.

Hundreds are 10 times bigger than Tens.

And

Each time we move right every position becomes 10 times smaller

From Hundreds, to Tens, to Ones

But if we continue past Ones ?

What is 10 times smaller than Ones?

110ths110ths (Tenths) are!

Before that we should first put a decimal point,

So we already know that where we put that decimal point.
We say the above example as four hundred and fifty-six and eight-tenths but

We usually just say four hundred and fifty-six point eight.

Types of Decimal Numbers:

Decimal Numbers may be of different kinds:

Recurring Decimal Numbers (Repeating or Non-Terminating Decimals)

Example-

3.125125 (Finite)

3.121212121212….. (Infinite)

&;

Non Recurring Decimal Numbers (Non Repeating or Terminating Decimals):

Example:

3.2376 (Finite)

3.137654….(Infinite)

&;

Decimal Fraction:

It represents the fraction whose denominator in powers of ten.

Example:

81.75 = 8175/100

32.425 = 32425/1000

Converting the Decimal Number into Decimal Fraction:

For the decimal point place a “1” in the denominator and remove the decimal point.

“1” is followed by a number of zeros equal to the number of digits following the decimal point.

For Example:

8 1 . 7 5

↓ ↓ ↓

1 0 0

81.75 = 8175/100

&;

8 represents the power of 10^1 that is the tenths position.

1 represents the power of 10^0 that is the units position.

7 represents the power of 10^(-1) that is the one-tenths position.

5 represents the power of 10^(-2) that is the one-hundredths position.

So that is how each digit is represented by a particular power of ten in the decimal number.

&;

Place Value of Decimal Numbers:

The place value is obtained by multiplication of the digit in the decimal number with its power of ten that the digit holds at its position.

The power of ten can be found using the following Place Value Chart:

The digits to the left of the decimal point are multiplied with the positive powers of ten in an increasing order from right to left.

The digits to the right of the decimal point are multiplied with the negative powers of 10 in an increasing order from left to right.

Following the same example 81.75

The decimal expansion of this is :

{(8*10)+(1*1)} + {(7*0.1)+(5*0.01)}

Where each number is multiplied by its associated power of ten.

How to Convert from Binary to Decimal

The binary system is the internal language of electronic computers. If you are a serious computer programmer, you should understand how to convert from binary to decimal. This wikiHow will show you how to do this.

1

Write down the binary number and list the powers of 2 from right to left. Let's say we want to convert the binary number 100110112 to decimal. First, write it down. Then, write down the powers of two from right to left. Start at 20, evaluating it as "1". Increment the exponent by one for each power. Stop when the amount of elements in the list is equal to the amount of digits in the binary number. The example number, 10011011, has eight digits, so the list, with eight elements, would look like this: 128, 64, 32, 16, 8, 4, 2, 1

2

Write the digits of the binary number below their corresponding powers of two. Now, just write 10011011 below the numbers 128, 64, 32, 16, 8, 4, 2, and 1 so that each binary digit corresponds with its power of two. The "1" to the right of the binary number should correspond with the "1" on the right of the listed powers of two, and so on. You can also write the binary digits above the powers of two, if you prefer it that way. What's important is that they match up.

3

Connect the digits in the binary number with their corresponding powers of two. Draw lines, starting from the right, connecting each consecutive digit of the binary number to the power of two that is next in the list above it. Begin by drawing a line from the first digit of the binary number to the first power of two in the list above it. Then, draw a line from the second digit of the binary number to the second power of two in the list. Continue connecting each digit with its corresponding power of two. This will help you visually see the relationship between the two sets of numbers.

4

Write down the final value of each power of two. Move through each digit of the binary number. If the digit is a 1, write its corresponding power of two below the line, under the digit. If the digit is a 0, write a 0 below the line, under the digit.

Since "1" corresponds with "1", it becomes a "1." Since "2" corresponds with "1," it becomes a "2." Since "4" corresponds with "0," it becomes "0." Since "8" corresponds with "1", it becomes "8," and since "16" corresponds with "1" it becomes "16." "32" corresponds with "0" and becomes "0" and "64" corresponds with "0" and therefore becomes "0" while "128" corresponds with "1" and becomes 128.

5

Add the final values. Now, add up the numbers written below the line. Here's what you do: 128 + 0 + 0 + 16 + 8 + 0 + 2 + 1 = 155. This is the decimal equivalent of the binary number 10011011.

6

Write the answer along with its base subscript. Now, all you have to do is write 15510, to show that you are working with a decimal answer, which must be operating in powers of 10. The more you get used to converting from binary to decimal, the more easy it will be for you to memorize the powers of two, and you'll be able to complete the task more quickly.

7

Use this method to convert a binary number with a decimal point to decimal form. You can use this method even when you want to covert a binary number such as 1.12 to decimal. All you have to do is know that the number on the left side of the decimal is in the units position, like normal, while the number on the right side of the decimal is in the "halves" position, or 1 x (1/2).

The "1" to the left of the decimal point is equal to 20, or 1. The 1 to the right of the decimal is equal to 2-1, or .5. Add up 1 and .5 and you get 1.5, which is 1.12 in decimal notation.

How to Convert from Decimal to Binary

The decimal (base ten) numeral system has ten possible values (0,1,2,3,4,5,6,7,8, or 9) for each place-value. In contrast, the binary (base two) numeral system has two possible values represented as 0 or 1 for each place-value.[1] Since the binary system is the internal language of electronic computers, serious computer programmers should understand how to convert from decimal to binary.

Method One of Two:
Performing Short Division by Two with RemainderEdit

1

Set up the problem. For this example, let's convert the decimal number 15610 to binary. Write the decimal number as the dividend inside an upside-down "long division" symbol. Write the base of the destination system (in our case, "2" for binary) as the divisor outside the curve of the division symbol.

This method is much easier to understand when visualized on paper, and is much easier for beginners, as it relies only on division by two.

To avoid confusion before and after conversion, write the number of the base system that you are working with as a subscript of each number. In this case, the decimal number will have a subscript of 10 and the binary equivalent will have a subscript of 2.

2

Divide. Write the integer answer (quotient) under the long division symbol, and write the remainder (0 or 1) to the right of the dividend.[2]

Since we are dividing by 2, when the dividend is even the binary remainder will be 0, and when the dividend is odd the binary remainder will be 1.

3

Continue to divide until you reach 0.Continue downwards, dividing each new quotient by two and writing the remainders to the right of each dividend. Stop when the quotient is 0.

4

Write out the new, binary number. Starting with the bottom remainder, read the sequence of remainders upwards to the top. For this example, you should have 10011100. This is the binary equivalent of the decimal number 156. Or, written with base subscripts: 15610 = 100111002

This method can be modified to convert from decimal to any base. The divisor is 2 because the desired destination is base 2 (binary). If the desired destination is a different base, replace the 2 in the method with the desired base. For example, if the desired destination is base 9, replace the 2 with 9. The final result will then be in the desired base.

Convert decimal fraction to hexadecimal fraction

BY SCHOOLELECTRONIC · PUBLISHED SEPTEMBER 28, 2013 · UPDATED APRIL 4, 2017

In the earlier post we discussed the integer conversion of a given decimal number to its equivalent hexadecimal number.

The method involved in the conversion of fractional decimal number to fraction hexadecimal number is different, in this post we will only concentrate on the fractional conversion.

Procedure

The steps for the conversion are given below:

Successive multiplication is used to convert a given fractional decimal number to its equivalent hexadecimal fraction.

Here the given decimal fraction is successively multiplied by the base of the target number system (16, here it is hexadecimal system).

During each multiplication iteration, the product generated will have a carry (integer part of the product) and a fractional part.

The carry obtained at each multiplication step becomes a numeral in the hexadecimal fraction.

The fractional part of the product is again multiplied by base 16 in the next step and the process is repeated until the fractional part becomes zero or the number of multiplication iteration equals the number of digits after the decimal point in the given decimal fraction.

Weights are assigned for the carry obtained at each multiplication step in the increasing order starting from the first multiplication step to the last step, such that the carry obtained in the first multiplication iteration is the most significant bit (MSD) after the decimal point and the carry obtained in the last multiplication iteration is the least significant bit (LSD)

This procedure is illustrated in the following example.

Ex1: Convert (0.0628)10 decimal fraction to hexadecimal fraction (?)16 using successive multiplication method

1st Multiplication Iteration

Multiply 0.0628 by 16

0.0628 x 16 = 1.0048(Product)   Fractional part=0.0048   Carry=1 (MSD)

2nd Multiplication Iteration

Multiply 0.0048 by 16
0.0048 x 16 = 0.0768(Product)   Fractional part = 0.0768   Carry = 0

3rd Multiplication Iteration

Multiply 0.0768 by 16
0.0768 x 16 = 1.2288(Product)   Fractional part = 0.2288   Carry = 1

4th Multiplication Iteration

Multiply 0.2288 by 16

0.2288 x 16 = 3.6608(Product)   Fractional part = 0.6608   Carry = 3 (LSD)

Here the fractional part doesn’t become zero but we obtain required number of significant digits after the decimal point. Thus we stop the multiplication iteration and assign the weights to the digits obtained in each multiplication step in the increasing order starting from the 1st multiplication step to last multiplication step.

Carry from the 1st multiplication iteration becomes MSB and carry from 4th iteration becomes LSB after the decimal point.

Hence, the fractional hexadecimal number of the given decimal fraction
(0.0628)10 is (0.1013)16.

‘Octal

The octal numeral system, or oct for short, is the base-8 number system, and uses the digits 0 to 7. Octal numerals can be made from binary numerals by grouping consecutive binary digits into groups of three (starting from the right). For example, the binary representation for decimal 74 is 1001010. Two zeroes can be added at the left: (00)1 001 010, corresponding the octal digits 1 1 2, yielding the octal representation 112.

In the decimal system each decimal place is a power of ten. For example:

{\displaystyle \mathbf {74} _{10}=\mathbf {7} \times 10^{1}+\mathbf {4} \times 10^{0}}

In the octal system each place is a power of eight. For example:

{\displaystyle \mathbf {112} _{8}=\mathbf {1} \times 8^{2}+\mathbf {1} \times 8^{1}+\mathbf {2} \times 8^{0}}

By performing the calculation above in the familiar decimal system we see why 112 in octal is equal to 64+8+2 = 74 in decimal.
How to Convert from Decimal to Octal

Octal is the base 8 number system, that only uses the digits 0 through 7. Its main advantage is the ease of conversion with binary (base 2), since each digit in octal can be written as a unique three-digit binary number.[1] Converting decimal to octal is a little more difficult, but you don't need to know any math past long division. Start with the division method, which finds each digit by dividing by powers of 8. The remainder method is faster and uses similar math, but it can be a little harder to understand why it works.

Method One of Two:
Converting with DivisionEdit

1

Use this method to learn the concepts. Of the two methods on this page, this method is easier to understand. If you're already confident working in different number systems, try the faster remainder method, below.

2

Write down the decimal number. For this example, we'll convert the decimal number 98 into octal.

3

List the powers of 8. Remember that "decimal" is called base 10 because each digit represents a power of 10. We call the first three digits 1s place, the 10s place, the 100s place — but we could also write this as the 100place, the 101 place, and the 102 place. Octal, or the base 8 number system, uses powers of 8 instead of powers of 10. Write a few of these powers of 8 in a horizontal line, from largest to smallest. Note that these numbers are all written in decimal (base 10):

82 81 80

Rewrite these as single numbers:

64 8 1

You don't need any powers of 8 larger than your original number (in this case, 98). Since 83 = 512, and 512 is larger than 98, we can leave it off the chart.

4

Divide the decimal number by the largest power of eight. Take a look at your decimal number: 98. The nine in the 10s place tells you that there are nine 10s in this number. 10 goes into this number 9 times. Similarly, with octal, we want to know how many "64s" go into the final number. Divide 98 by 64 to find out. The easiest way to do this is to make a chart, reading top to bottom:[2]

98
÷

64   8   1
=

1 ← This is the first digit of your octal number.

5

Find the remainder. Calculate the remainder of the division problem, or the amount left over that doesn't go in evenly. Write your answer at the top of the second column. This is what's left of your number after the first digit is calculated. In our example, 98 ÷ 64 = 1. Since 1 x 64 = 64, the remainder is 98 - 64 = 34. Add this to your chart:

98   34
÷

64 8 1
=

1

6

Divide the remainder by the next power of 8. To find the next digit, we move one step down to the next power of 8. Divide the remainder by this number and fill out your chart's second column:

98 34
÷ ÷

64 8 1
= =

1 4

7

Repeat until you've found the full answer.Just as before, find the remainder of your answer and write it at the top of the next column. Keep dividing and finding the remainder until you've done this for every column, including 80(the ones place). Your final row is the final decimal number converted to octal. Here's our example with the full chart filled out (note that 2 is the remainder of 34÷8):

98 34 2
÷ ÷ ÷

64 8 1
= = =

1 4 2

The final answer: 98 base 10 = 142 base 8. You can write this as 9810 = 1428

8

Check your work. To check your work, multiply each digit in octal by the power of 8 it represents. You should end up with your original number. Let's check our answer, 142:

2 x 80 = 2 x 1 =2

4 x 81 = 4 x 8 = 32

1 x 82 = 1 x 64 = 64

2 + 32 + 64 = 98, the number we started with.

9

Try this practice problem. Practice this method by converting the decimal number 327 into octal. When you think you have the answer, highlight the invisible text below to see the whole problem laid out.

Highlight this area:

327 7 7
÷ ÷ ÷

64 8 1
= = =

5 0 7

The answer is 507.

DIGITAL ELECTRONICS

3. Chapter

Codes & Parity

CONCEPT OF CODE:

- Codes in Digital Electronics. Indigital electronics, codes are used to communicate the information between computers.
  These codes represent the information symbolically as a string of bits 0 and 1 and rulesdefined by the code decide the arrangement of these bits.

CODES :

- HAT IS codes in digital electronics? Digital coding is the process of using binary digits to represent letters, characters and other symbols in a digital format. There are several types of digital codes widely used today, but they use the same principle of combining binary numbers to represent a character.

WEIGHTED CODES :

- Weighted Codes:-The weighted codes are those that obey the position weightingprinciple,which states that the position of each number represent a specific weight. In these codes each decimal digit is represented by a group of four bits. ... There are millions
  of weighted code The most common one is 8421/BCD.

Weighted Code:-

In weighted code, each digit position has a weight or value. The sum of all digits multiplied by a weight gives the total amount being represented.

We can express any decimal number in tens, hundreds, thousands and so on.

Eg:- Decimal number 4327 can be written as

4327= 4000+300+20+7

In the power of 10, it becomes

$4327= 4(10^3)+3(10^2)+2(10^1)+7(10^0)$

BCD or 8421 is a type of weighted code where each digit position is being assigned a specific weight.

Example of 8421

- Codes each decimal digit is represented by a group of four bits. ... For example, in 8421/BCDcode, 1001 the weights of 1, 1, 0, 1 (from left to right) are 8, 4, 2 and 1 respectively. There are millions of weighted code The most common one is 8421/BCD Code.Examples:8421,2421,84-2-1 are all weighted codes.

NON – WEIGHTED CODES :

- Gray Code. It is the non-weighted code and it is not arithmeticcodes. That means there are no specific weights assigned to the bit position. It has a very special feature that, only one bit will change each time the decimal number is incremented as shown in fig.

<u>Non-weighted code:-</u>

In non-weighted code, there is no positional weight i.e. each position within the binary number is not assigned a prefixed value. No specific weights are assigned to bit position in non –weighted code.

The non-weighted codes are:-

a) The Gray code b) The Excess-3 code

BCD :

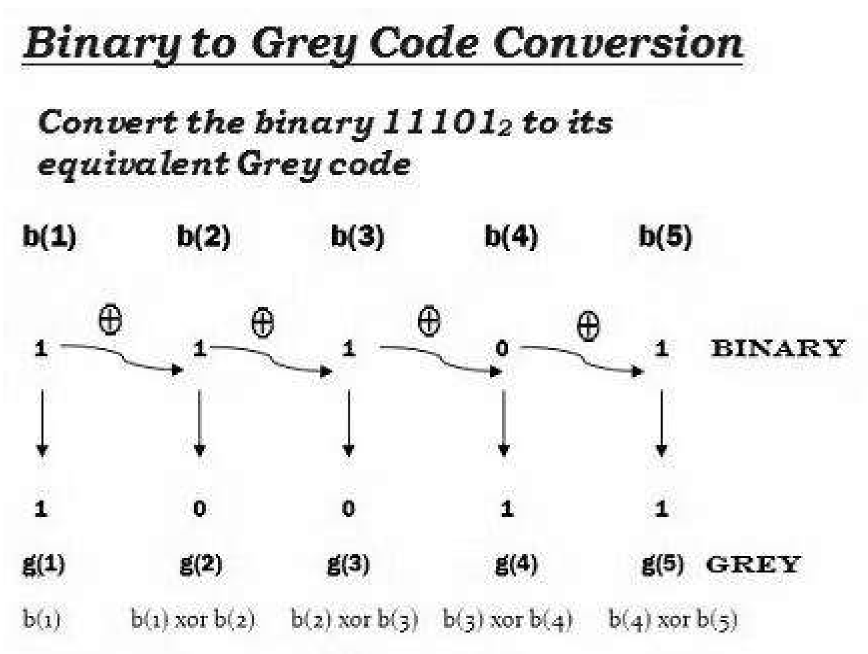
- In computing and electronic systems, binary-coded decimal(BCD) is a class of binary encodings of decimal numberswhere each decimal digit is represented by a fixed number of bits, usually four or eight.

Excess-3 :

- 3-excess or 10-excess-3 binary code (often abbreviated as XS-3, 3XS or X3) or Stibitzcode (after George Stibitz, who built a relay-based adding machine in 1937) is a self-complementary binary-coded decimal (BCD) code and numeral system. It is a biased representation

GRAY CODE :

- Gray code evaluates the nature of binary code or data that is composed of on and off indicators, commonly represented by ones and zeros. Developed by Bell Labs scientists, gray code has been used to look at clarity and error correction in binary communications. Gray code is also known as reflected binary code.



CONCEPT OF PARITY :

- In computers, parity (from the Latin paritas, meaning equal or equivalent) is a technique that checks whether data has been lost or written over when it is moved from one place in storage to another or when it is transmitted between computers.

SINGLE PARITY :

- A parity bit, also known as a check bit, is a single bit that can be appended to a binary string. It is set to either 1 or 0 to make the total number of 1-bits either even ("even parity") or odd ("oddparity").

DOUBLE PARITY :

- Double-parity RAID (redundant array of independent disks), also called diagonal-parity RAID, Advanced Data Guarding (RAID_ADG), or RAID-6, is a method of protecting against multiple storage drive failures by creating two sets of parity data on an array of hard disks.

ERRORE DETECTION :

- In networking, error detection refers to the techniques used to detect noise or other impairments introduced into data while it is transmitted from source to destination. Error detection ensures reliable delivery of data across vulnerable networks.

DIGITAL ELECTRONICS

lesson:- LOGIC SIMPLIFICATION

A) Postulates and Theorems of Boolean Algebra

Assume **A**, **B**, and **C** are logical states that can have the values **0** (false) and **1** (true).
"+" means **OR**, **"·"** means **AND**, and ^{NOT}**[A]** means **NOT A**.

Postulates

(1)	$A + 0 = A$	$A \cdot 1 = A$	identity
(2)	$A + {}^{NOT}[A] = 1$	$A \cdot {}^{NOT}[A] = 0$	complement
(3)	$A + B = B + A$	$A \cdot B = B \cdot A$	commutative law
(4)	$A + (B + C) = (A + B) + C$	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	associative law
(5)	$A + (B \cdot C) = (A + B) \cdot (A + C)$	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$	distributive law

Theorems

(6)	$A + A = A$	$A \cdot A = A$	
(7)	$A + 1 = 1$	$A \cdot 0 = 0$	
(8)	$A + (A \cdot B) = A$	$A \cdot (A + B) = A$	
(9)	$A + ({}^{NOT}[A] \cdot B) = A + B$	$A \cdot ({}^{NOT}[A] + B) = A \cdot B$	
(10)	$(A \cdot B) + ({}^{NOT}[A] \cdot C) + (B \cdot C) = (A \cdot B) + ({}^{NOT}[A] \cdot C)$	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$	
(11)	${}^{NOT}[A + B] = {}^{NOT}[A] \cdot {}^{NOT}[B]$	${}^{NOT}[A \cdot B] = {}^{NOT}[A] + {}^{NOT}[B]$	de Morgan's theorem

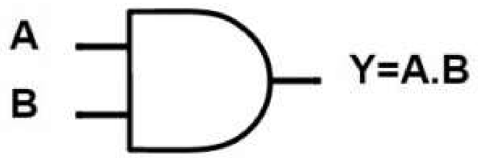
B) Logic gates

Logic gates are the basic building blocks of digital electronic circuits. A logic gate is a piece of an electronic circuit, that can be used to implement Boolean expressions.

Laws and theorems of Boolean logic are used to manipulate the Boolean expressions and logic gates are used to implement these Boolean expressions in digital electronics. AND gate, OR gate and NOT gate are the three basic logic gates used in digital electronics.

AND Gate

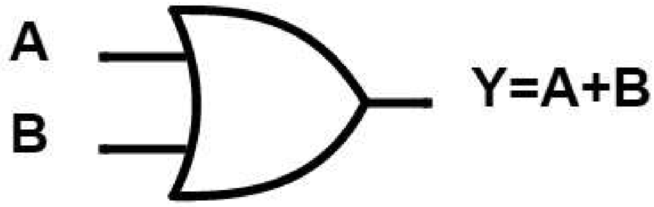
Logic AND gate is a basic logic gate of which the output is equal to the product of its inputs. This gate multiplies both of its inputs so this gate is used to find the multiplication of inputs in binary algebra.The output of an AND gate is HIGH only if both the inputs of the gate are HIGH. The output for all the other cases of the inputs is LOW. The logic symbol and the truth table of an AND gate is shown below.



A	B	Y=A.B
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

The output of the logic OR gate is equal to the sum of its inputs. This gate adds both of its inputs so this gate is used to find the summation or the addition of inputs in binary algebra. The output of an OR gate is HIGH if either of the inputs are HIGH. The output is LOW only when all the inputs are LOW. The logic symbol and the truth table of an OR gate is shown below.



A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT Gate

Logic NOT gate is a basic logic gate of which the output is equal to the inverse of its input. This gate produces the complement of the input. So this gate is used to represent the complement of variables in binary algebra. If the input is HIGH, the output is LOW and if the input is LOW, the output is HIGH. The logic symbol and the truth table of a NOT gate is shown below.

SOP Boolean Function Implementation using logic gates

The sum of product or SOP form is represented by using basic logic gates like AND gate and OR gate. The SOP form implementation will have the AND gate at its input side and as the output of the function is the sum of all product terms, it has an OR gate at its output side. This is important to remember that we use NOT gate to represent the inverse or complement of the variables.

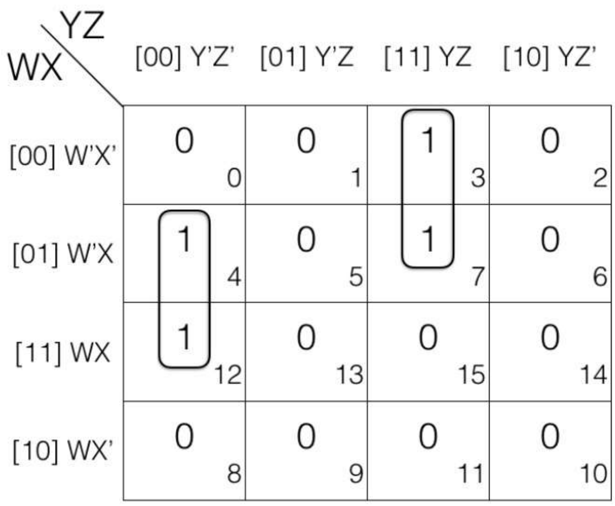
Logic gate implementation

Reduction rules for SOP using K-map

There are a couple of rules that we use to reduce SOP using K-map first we will cover the rules step by step then we will solve problem. So lets start

Pair reduction Rule

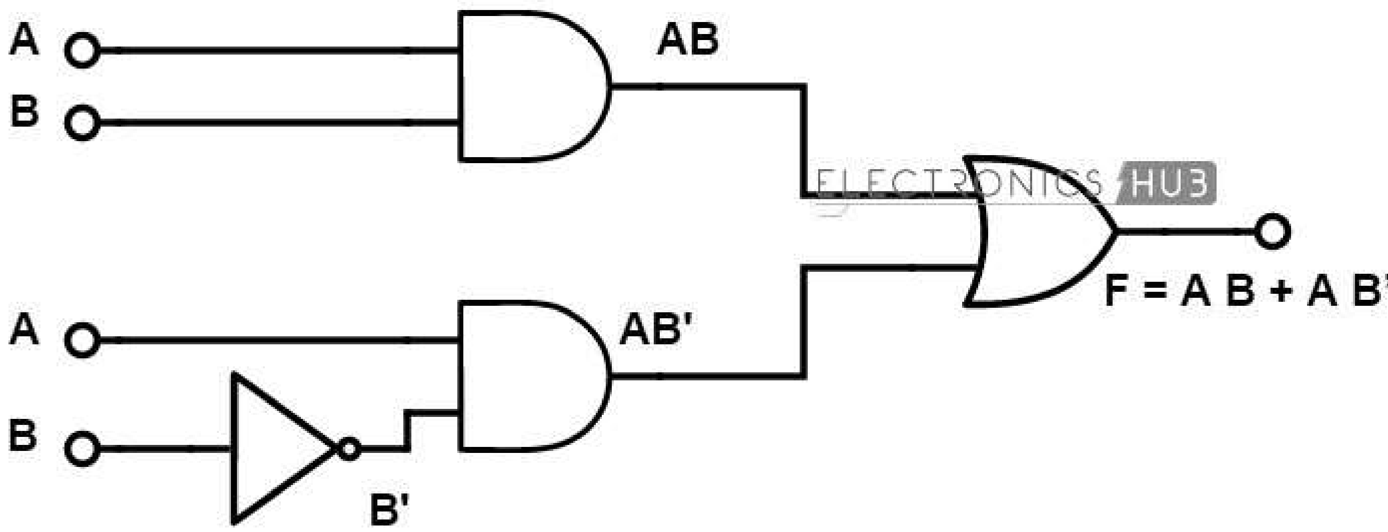
Consider the following 4 variables K-mapNow we mark the cells in pair (set of 2) having value 1.
The 1st pair = W'XY'Z' + WXY'Z'
the 2nd pair = W'X'YZ + W'XYZ
(the pairs are in Sum of Products SOP form)
Now we will remove the variable tha
Looking at the 1st pair W' changed t
the 2nd pair X' changed to X so we
So the updated pairs after reduction
1st pair
= W'XY'Z' + WXY'Z'
= XY'Z'
2nd pair
= W'X'YZ + W'XYZ
= W'YZ
Pair reduction rule removes 1 variable.



Implementation for 2 input variables

Implement the Boolean function by using basic logic gates. F = A B + A B'

In the given SOP function, we have one compliment term, AB'. So to represent the compliment input, we are using the NOT gates at the input side. And to represent the product term, we use AND gates. See the below given logic diagram for representation of the Boolean function.

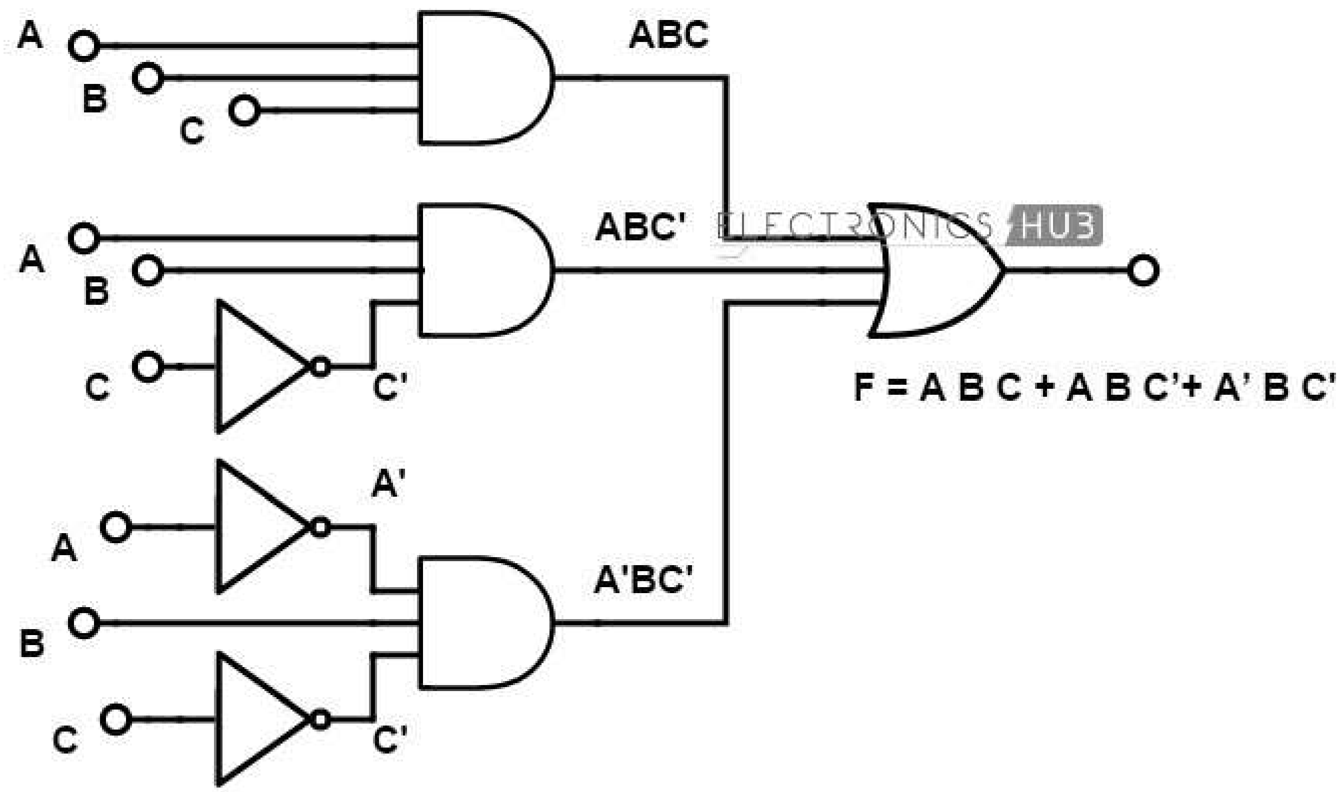


Implementation for 3 input variables

Implement the Boolean function by using basic logic gates.

F = A B C + A B C'+ A' B C'

In the given function, we have two compliment terms, A'B C' and ABC'. So to represent the compliment input, we are using the NOT gates at the input side. And to represent the product term, we use AND gates. See the below given logic diagram for representation of the Boolean function.



POS Boolean Function Implementation using logic gates

The product of sums or POS form can be represented by using basic logic gates like AND gate and OR gates. The POS form implementation will have the OR gate at its input side and as the output of the function is product of all sum terms, it has AND gate at its output side. In POS form implementation, we use NOT gate to represent the inverse or complement of the variables.

Reduction rules for POS using K-map

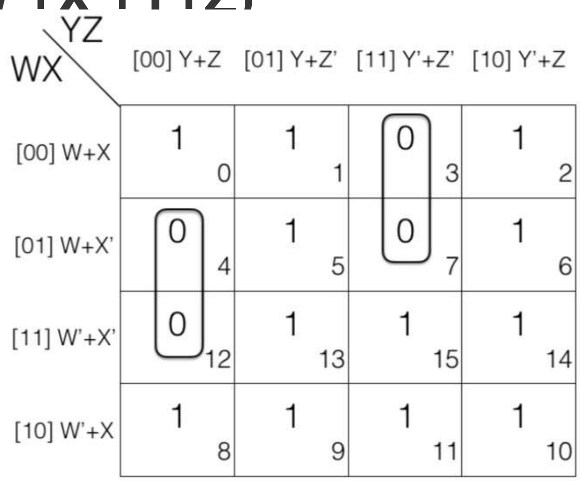
There are a couple of rules that we use to reduce POS using K-map. First we will cover the rules step by step then we will solve problem. So lets start...

Pair reduction Rule

Consider the following 4 variables K-map.

Now we mark the cells in pair (set of 2) having value 0.

1st pair = (W+X’+Y+Z) . (W’+X’+Y+Z)



2nd pair = (W+X+Y’+Z’) . (

(the pairs are in Product c

Now we will remove the v

in the 1st and 2nd pair. Lo

W changed to W’ so we re

the 2nd pair X changed to

So the updated pairs after reduction are given below.

1st pair

= (W+X’+Y+Z) . (W’+X’+Y+Z)

= (X’+Y+Z)

2nd pair

= (W+X+Y’+Z’) . (W+X’+Y’+Z’)

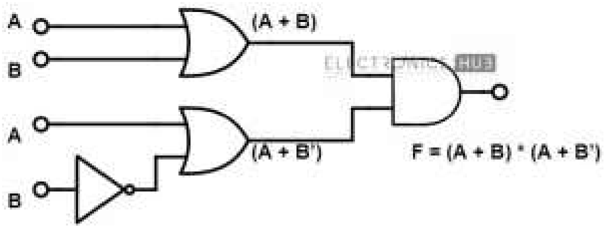
= (W+Y’+Z’)

Note! pair reduction rule removes 1 variable.

Implementation for 2 input variables

Implement the Boolean function by using basic logic gates. F = (A + B) * (A + B’)

In the given function, we have a complement term, (A + B) and (A + B’). So to represent the compliment input, we are using the NOT gates at the input side. And to represent the sum term, we use OR gates. See the below given logic diagram for representation of the Boolean function.

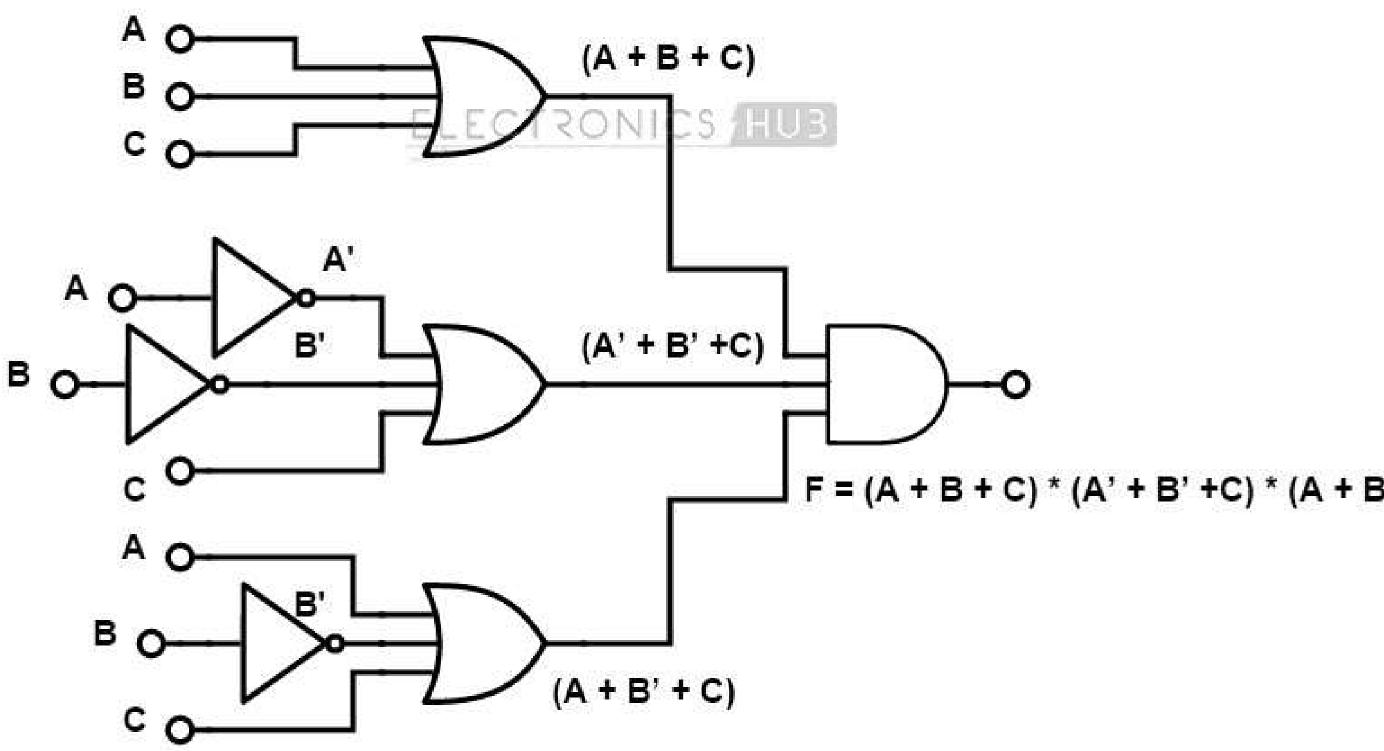


Implementation for 3 input variables

Implement the Boolean function by using basic logic gates.

F = (A + B + C) * (A’ + B’ +C) * (A + B’ + C)

In the given Boolean function, we have two compliment terms, (A’ + B’ +C) and (A + B’ + C). So to represent the compliment input, we are using the NOT gates at the input side. And to represent the sum term, we use OR gates. See the below given logic diagram for representation of the Boolean function.



MULTISTAGE AMPLIFIERS

For most systems a single transistor amplifier does not provide sufficient gain or bandwidth or will not have the correct input or output impedance matching. The solution is to combine multiple stages of amplification. We have the three basic one transistor amplifier configurations to use as building blocks to create more complex amplifier systems which can provide better optimized specifications and performance. The sections in this chapter tend to use BJT devices to illustrate the circuit concepts but these multi-stage amplifiers can be constructed from MOS FET devices, or a combination, just as easily and the methods used to analyze them are much the same as well.

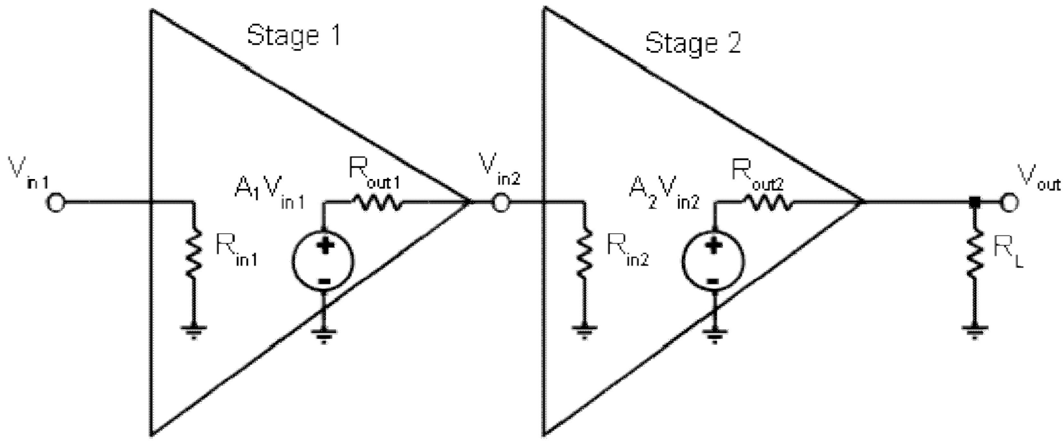


fig 1. Multistage Amplifier

For most systems a single transistor amplifier does not provide sufficient gain or bandwidth or will not have the correct input or output impedance matching. The solution is to combine multiple stages of amplification. We have the three basic one transistor amplifier configurations to use as building blocks to create more complex amplifier systems which can provide better optimized specifications and performance. The sections in this chapter tend to use BJT

devices to illustrate the circuit concepts but these multi-stage amplifiers can be constructed from MOS FET devices, or a combination, just as easily and the methods used to analyze them are much the same as well.

Cascade of two single transistor stages

The impact of input and output loading can be minimized by cascading two amplifiers with appropriate input and output characteristics. Multistage cascading can be used to create amplifiers with high input resistance, low output resistance and large gains.

Common Emitter / Common Collector cascade

The cascade of a Common Emitter amplifier stage followed by a Common Collector (emitter-follower) amplifier stage can provide a good overall voltage amplifier. The Common Emitter input resistance is relatively high and Common Collector output resistance is relatively low. The voltage follower second stage, Q2, contributes no increase in voltage gain but provides a near voltage-source (low resistance) output so that the gain is nearly independent of load resistance. The high input resistance of the Common Emitter stage, Q1, makes the input voltage nearly independent of input-source resistance. Multiple Common Emitter stages can be cascaded with emitter follower stages inserted between them to reduce the attenuation due to inter-stage loading.

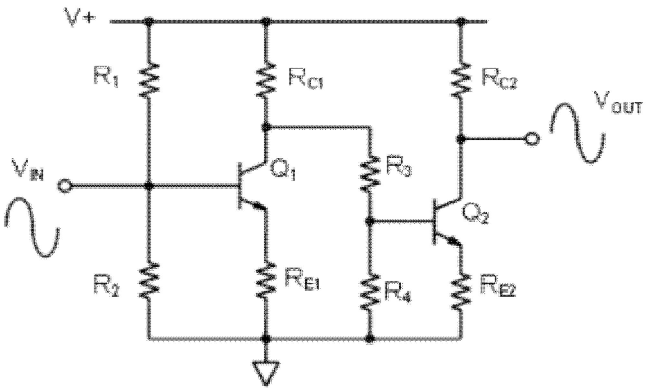


fig 2 Two Stage Amplifier

It is possible to create a multistage cascade where each stage is separately biased and coupled to adjacent stages via DC blocking capacitors. Inserting coupling capacitors between stages blocks the DC operating bias level of one stage from affecting the DC operating point of the next. This solves many of the limitations we saw in section 10.1.2. However, the resulting overall amplifier can no longer respond to DC, or very low frequency, inputs.

RC coupled Common Emitter stages

The infinity symbol next to coupling capacitors C1 C2 and C3 is used to indicate that the unspecified capacitance is large enough at the specified signal frequency to have a negligible reactance and can be treated as an AC short-circuit. It is also useful to note at this point that the method of including capacitors across the emitter degeneration resistors RE1 and RE2 to increase the gain at higher frequencies can be employed in the case of these multistage amplifiers as well as the single stage amplifiers discussed in Chapter 9.

Complementary Pair Amplifier

Not only can NPN transistors or n-type MOS devices be combined in multiple stages, so can the complementary PNP and p-type MOS devices. Having both polarities of transistors allows for more flexibility in how amplifier sages can be combined and can make biasing easier as well.

For example, a complementary cascade amplifier is shown in figure. The second common emitter stage uses a PNP transistor. The gain calculation procedure is the same as the all-NPN cascade. The advantage of the complementary cascade amplifier is that the p-stage collector DC operating point tends to cancel the bias level “stacking” issue we encountered in the all n-type common emitter amplifier cascade we explored in section.

A two-stage 'Complementary Pair' BJT amplifier circuit diagram is shown in figure 10.1.4. The rationale behind a complementary pair cascade is a problem that can arise with a cascade of similar n-type stages. To avoid saturation the collector voltage of each stage must be greater than the base voltage, enough greater to allow for the collector voltage signal swing. However since the base voltage of the second stage is taken from the collector of the first stage it is inherently larger that the first stage base voltage, and the second stage collector voltage is still higher. But this decreases the available amplitude for the amplified signal. Adding a third stage would even further aggravate this situation.

If a PNP second stage is used a base voltage close to the positive power supply accommodates a desirable higher first stage NPN collector voltage. Moreover a third NPN stage can be cascaded at the PNP stage output without the severe voltage offset problem of a cascade of similar stages.

Estimating the DC bias voltages and currents is simplified considerably under the assumption that the PNP base current is small compared to the NPN collector current. Of course this is not necessarily so but there are two reasons generally favoring such a relationship. Considering the common DC power supply and the bias configuration one might expect intuitively that the two transistors will have roughly comparable collector currents, and a typical PNP ß of about 120 would then support the approximation. A sort of circular reasoning based on an enlightened self-interest suggests that a simplifying approximation relatively easy to implement in fact will be implemented so as actually to simplify the circuit design process. In any event the assumption can be made and subsequently justified explicitly by verifying the consistency of the assumption with values calculated using it. And of course adjustments can be made if and where needed in an iterative process.

The approximation we are recommending here makes the bias calculations for each stage effectively independent of one another. The estimate so made can be refined by a second iteration in which rather than neglecting the PNP base current the value of this current estimated from the first iteration is used. This refinement is rarely if ever necessary. The use of modern circuit simulation software can of course speed up this iteration process.

We have observed that the main drawback of RC coupled amplifier is that the effective load resistance gets reduced. This is because, the input impedance of an amplifier is low, while its output impedance is high.When they are coupled to make a multistage amplifier, the high output impedance of one stage comes in parallel with the low input impedance of next stage. Hence, effective load

resistance is decreased. This problem can be overcome by a transformer coupled amplifier.In a transformer-coupled amplifier, the stages of amplifier are coupled using a transformer. Let us go into the constructional and operational details of a transformer coupled amplifier.Construction of Transformer Coupled Amplifier

The amplifier circuit in which, the previous stage is connected to the next stage using a coupling transformer, is called as Transformer coupled amplifier.

The coupling transformer T1 is used to feed the output of 1st stage to the input of 2nd stage. The collector load is replaced by the primary winding of the transformer. The secondary winding is connected between the potential divider and the base of 2nd stage, which provides the input to the 2nd stage. Instead of coupling capacitor like in RC coupled amplifier, a transformer is used for coupling any two stages, in the transformer coupled amplifier circuit.

Transformer Coupled Multistage Amplifier

The potential divider network R1 and R2 and the resistor Re together form the biasing and stabilization network. The emitter by-pass capacitor Ce offers a low reactance path to the signal. The resistor RL is used as a load impedance. The input capacitor Cin present at the initial stage of the amplifier couples AC signal to the base of the transistor. The capacitor CC is the coupling capacitor that connects two stages and prevents DC interference between the stages and controls the shift of operating point.

Operation of Transformer Coupled Amplifier

When an AC signal is applied to the input of the base of the first transistor then it gets amplified by the transistor and appears at the collector to which the primary of the transformer is connected.The transformer which is used as a coupling device in this circuit has the property of impedance changing, which means the low resistance of a stage (or load) can be reflected as a high load resistance to the previous stage. Hence the voltage at the primary is transferred according to the turns ratio of the secondary winding of the transformer.This transformer

coupling provides good impedance matching between the stages of amplifier. The transformer coupled amplifier is generally used for power amplification.

Frequency Response of Transformer Coupled Amplifier

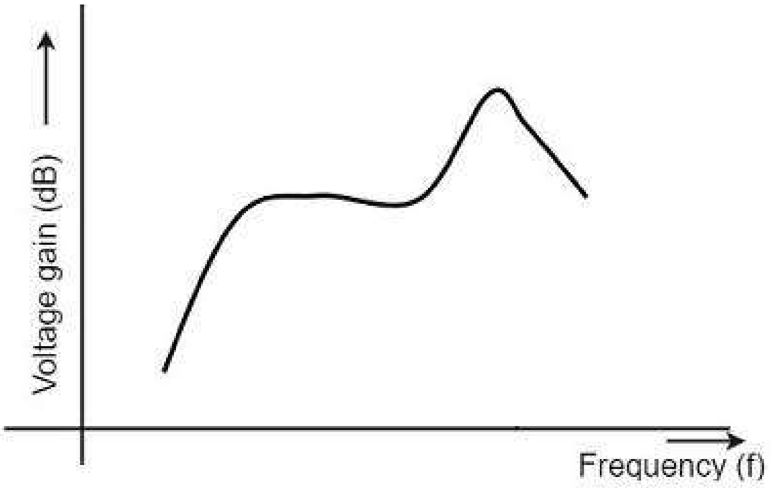


fig. frequency response of transformer coupled amplifier.

The figure above shows the frequency response of a transformer coupled amplifier. The gain of the amplifier is constant only for a small range of frequencies. The output voltage is equal to the collector current multiplied by the reactance of primary.

Frequency Coupled

At low frequencies, the reactance of primary begins to fall, resulting in decreased gain. At high frequencies, the capacitance between turns of windings acts as a bypass condenser to reduce the output voltage and hence gain.So, the amplification of audio signals will not be proportionate and some distortion will also get introduced, which is called as Frequency distortion.

Advantages of Transformer Coupled Amplifier

The following are the advantages of a transformer coupled amplifier −

An excellent impedance matching is provided.Gain achieved is higher.There will be no power loss in collector and base resistors.Efficient in operation.

Disadvantages of Transformer Coupled Amplifier

The following are the disadvantages of a transformer coupled amplifier −

Though the gain is high, it varies considerably with frequency. Hence a poor frequency response.Frequency distortion is higher.Transformers tend to produce hum noise.Transformers are bulky and costly.

Applications

The following are the applications of a transformer coupled amplifier −

Mostly used for impedance matching purposes.Used for Power amplification.Used in applications where maximum power transfer is needed *CHAPTER 2*

LARGE SIGNAL AMPLIFIER

An amplifier is an electronic device used to increase the magnitude of voltage/current/power of an input signal. It takes in a weak electrical signal/waveform and reproduces a similar stronger waveform at the output by using an external power source.Depending on the changes it makes to the input signal, amplifiers are broadly classified into Current, Voltage and Power amplifiers. In this article we will learn about power amplifiers in detail.

WHAT IS POWER AMPLIFIER

A power amplifier is an electronic amplifier designed to increase the magnitude of power of a given input signal. The power of the input signal is increased to a level high enough to drive loads of output devices like speakers, headphones, RF transmitters etc. Unlike voltage/current amplifiers, a power amplifier is designed to drive loads directly and is used as a final block in an amplifier chain.

The input signal to a power amplifier needs to be above a certain threshold. So instead of directly passing the raw audio/RF signal to the power amplifier, it is first pre-amplified using current/voltage amplifiers and is sent as input to the power amp after making necessary modifications. You can observe the block diagram of an audio amplifier and the usage of power amplifier below.

In this case a microphone is used as an input source. The magnitude of signal from the microphone is not enough for the power amplifier. So first it is pre-amplified where its voltage and current are increased slightly. Then the signal is passed through tone and volume controls circuit which makes aesthetic adjustments to the audio waveform. Finally the signal is passed through a power amplifier and the output from power amp is fed to a speaker.

Types of Power Amplifiers

Depending on the type of output device that is connected, power amplifiers are divided into the following three types.

Audio Power Amplifiers

This type of power amplifiers are used for increasing the magnitude of power of a weaker audio Signal. The amplifiers used in speaker driving circuitries of televisions, mobile phones etc. come under this category.

The output of an audio power amplifier ranges from a few milliwatts (like in headphone amplifiers) to thousands of watts (like power amplifiers in Hi-Fi/Home theatre systems).

Radio Frequency Power Amplifiers

Wireless transmissions require modulated waves to be sent over long distances via air. The signals are transmitted using antennas and the range of transmission depends on the magnitude of power of signals fed to the antenna.

For wireless transmissions like FM broadcasting, antennas require input signals at thousands of kilowatts of power. Here, Radio Frequency Power amplifiers are employed to increase the magnitude of power of modulated waves to a level high enough for reaching required transmission distance.

DC Power Amplifiers

DC power amplifiers are used to amplify the power of a PWM(Pulse Width Modulated) signals. They are used in electronic control systems which need high power signals to drive motors or actuators. They take input from microcontroller systems, increase its power and feed the amplified signal to DC motors or Actuators.

Power Amplifier Classes

There are multiple ways of designing a power amplifier circuit. The operation and output characteristics of each of the circuit configurations differs from each other.

To differentiate the characteristics and behaviour of different power amplifier circuits, Power Amplifier Classes are used in which letter symbols are assigned to identify the method of operation.

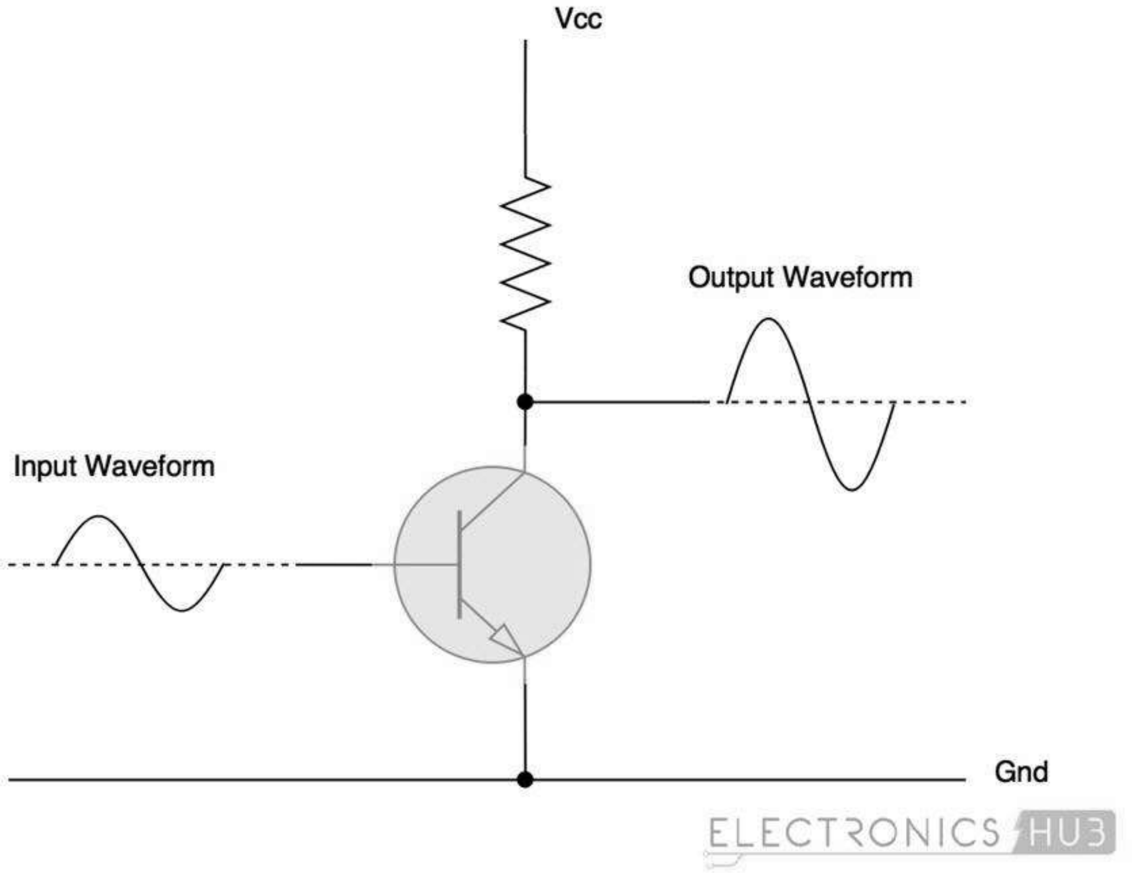
They are broadly classified into two categories. Power amplifiers designed to amplify analog signals come under A, B, AB or C category. Power amplifiers designed to amplify Pulse Width Modulated(PWM) digital signals come under D, E, F etc.

The most commonly used power amplifiers are the ones that are used in audio amplifier circuits and they come under classes A, B, AB or C. So let’s take a look at them in detail.

Class A Power Amplifier

Analog waveforms are made up of positive highs and negative lows. In this class of amplifiers, the entire input waveform is used in the amplification process.

A single transistor is used to amplify both the positive and negative halves of the waveform. This makes their design simple and makes class A amplifiers the most commonly used type of power amplifiers. Although this class of power amplifiers are superseded by better designs, they are still popular among hobbyists.



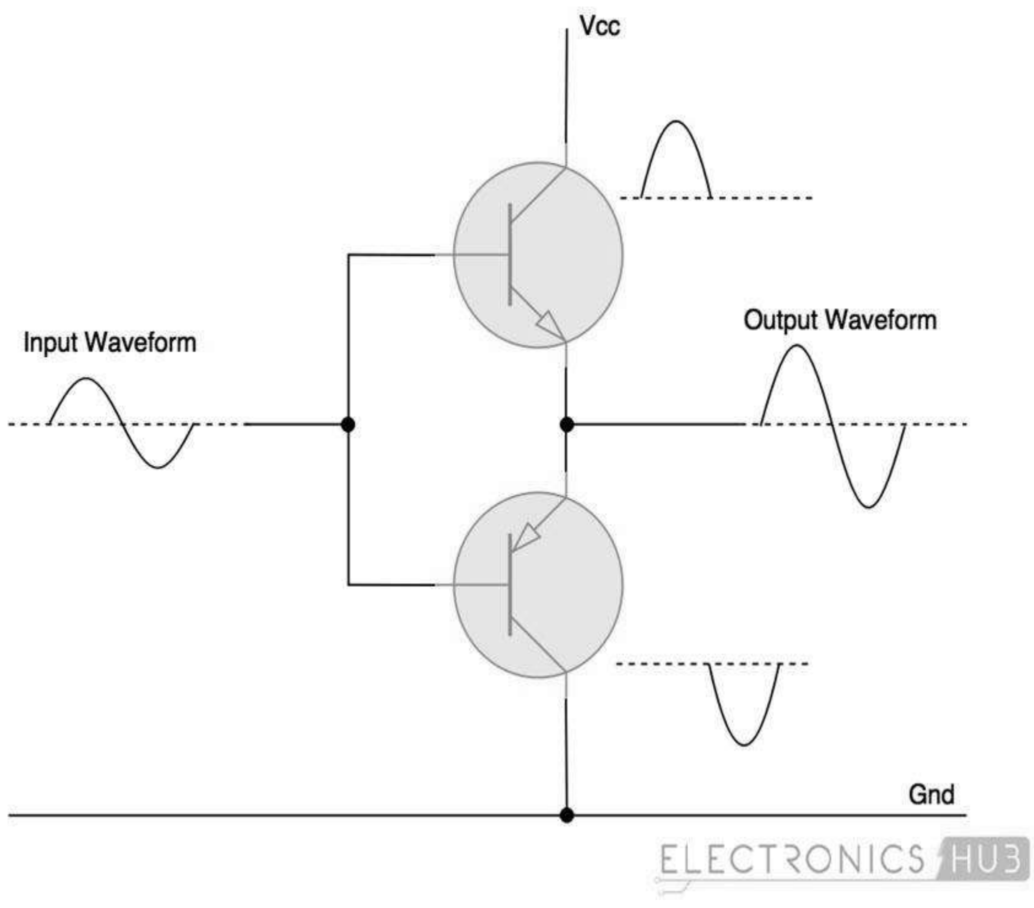
In this class of amplifiers, the active element (the electronic component used for amplifying, which is transistor in this case) is in use all the time even if there is no input signal. This generates lot of heat and reduces the efficiency of class A amplifiers to 25% in normal configuration and 50% in a transformer coupled configuration.

The conduction angle (the portion of waveform used for amplification, out of 360°) for class A amplifiers is 360°. So the signal distortion levels are very less allowing better high frequency performance.

Class B Power Amplifier

Class B power amplifiers are designed to reduce the efficiency and heating problems present in the class A amplifiers. Instead of a single transistor to amplify the entire waveform, this class of amplifiers use two complementary transistors.

One transistor amplifies positive half of the waveform and the other amplifies negative half of the waveform. So each active device conducts for one half (180°) of the waveform and two of them when combined amplify the entire signal.

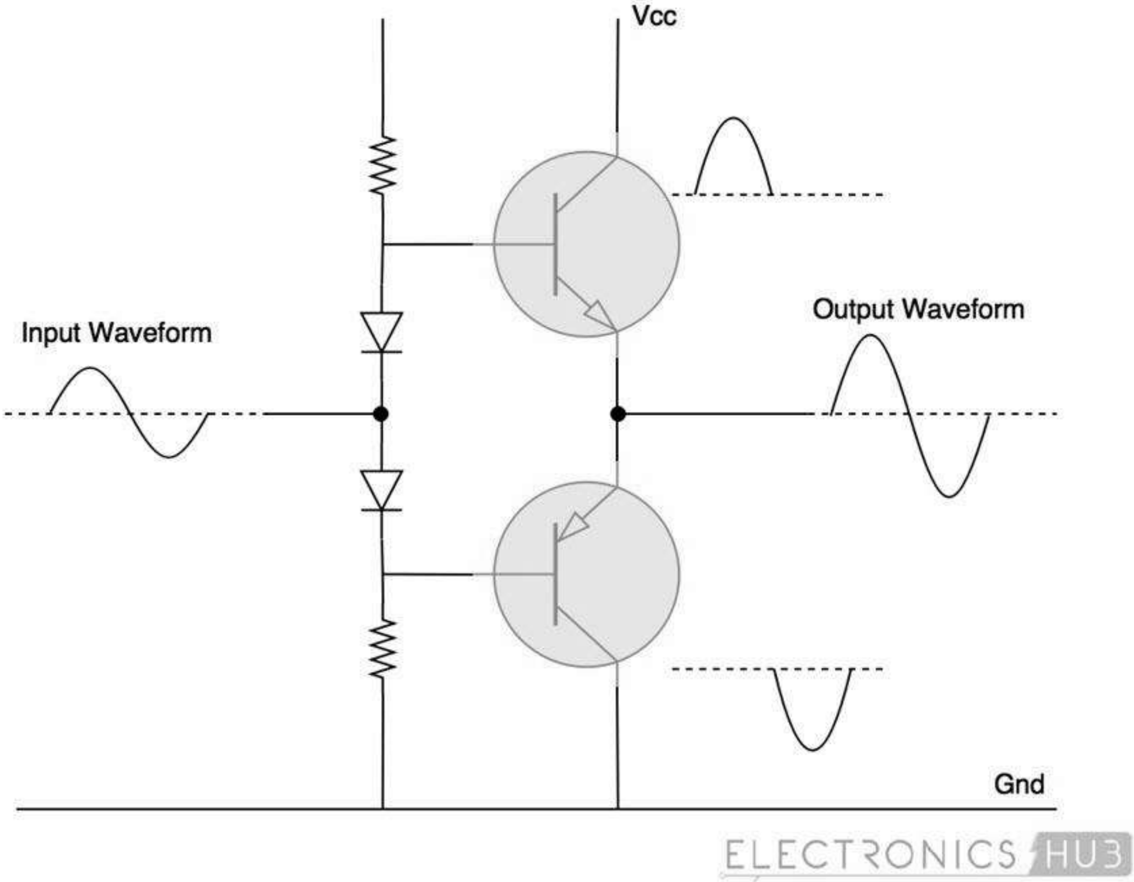


The efficiency of class B amplifiers is improved a lot over class A amplifiers because of two transistor design. They can reach a theoretical efficiency of about 75%. Power amplifiers of this class are used in battery operated devices like FM radios and transistor radios.

Because of superposition of two halves of the waveform, there exists a small distortion at the crossover region. To reduce this signal distortion, class AB amplifiers are designed.

Class AB Power Amplifier

Class AB amplifiers are a combination of class A and class B amplifiers. This class of amplifiers are designed to reduce the less efficiency problem of class A amplifiers and distortion of signal at crossover region in class B amplifiers.



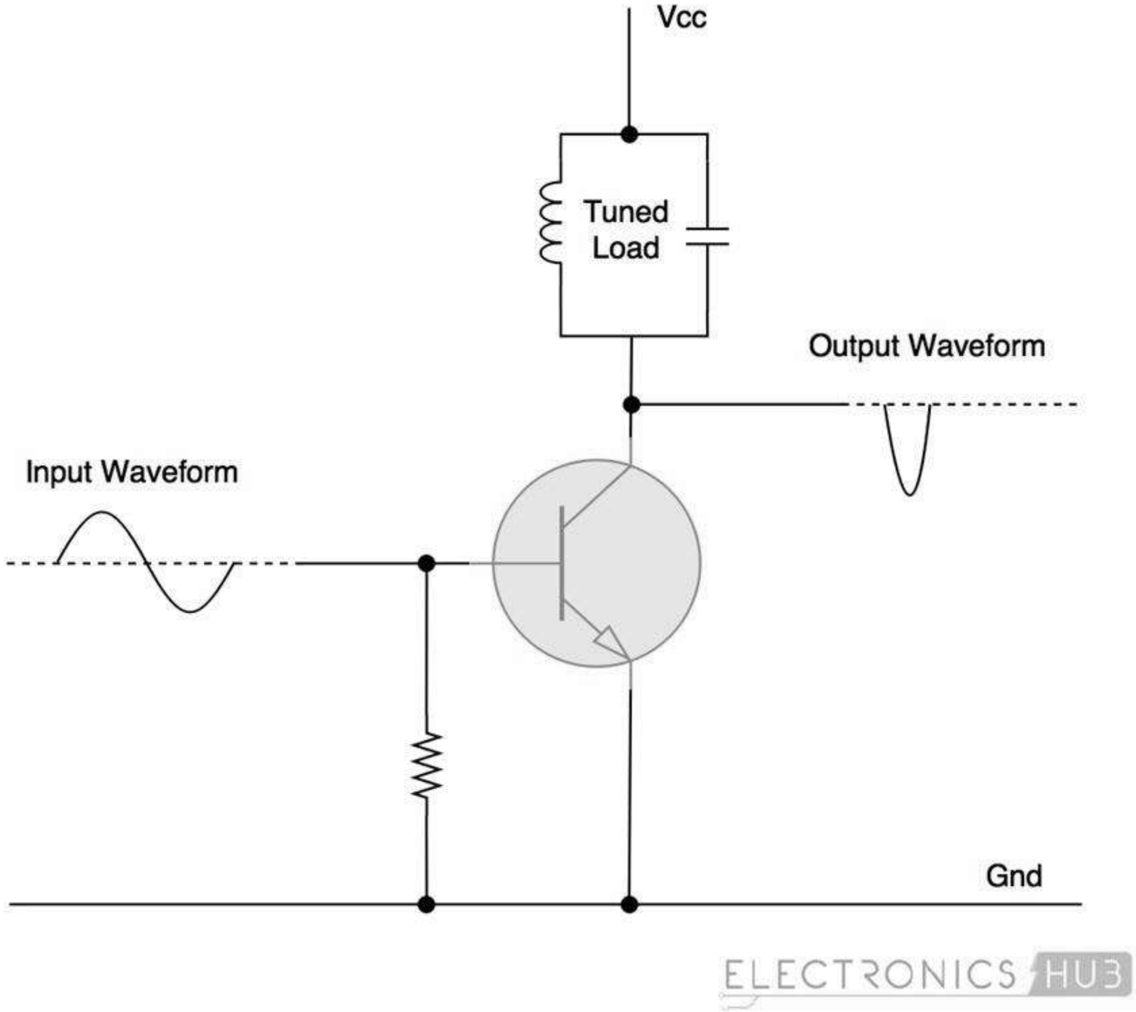
It maintains high frequency response like in class A amplifiers and good efficiency as in class B amplifiers. A combination of diodes and resistors are used to provide little bias voltage which reduces the distortion of waveform near the crossover region. There is a little drop in efficiency (60%) because of this.

Class C Power Amplifier

The design of class C power amplifiers allows greater efficiencies but reduces the linearity/conduction angle, which is under 90°. In other words, it sacrifices quality of amplification for increase in efficiency.

Lesser conduction angle implies greater distortion and so this class of amplifiers are not suited for audio amplification. They are used in high frequency oscillators and amplification of Radio Frequency signals.

Class C amplifiers generally contain a tuned load which filters and amplifies input signals of certain frequency, and the waveforms of other frequencies are supressed.



In this type of power amplifier, the active element conducts only when the input voltage is above a certain threshold, which reduces power dissipation and increases efficiency.

Other Power Amplifier Classes

Power amplifier classes D, E, F, G etc. are used to amplify PWM modulated digital signals. They come under the category of switching power amplifiers and turn the output either constantly ON or constantly OFF without any other levels in between.

Because of this simplicity, power amplifiers falling under the above mentioned classes can reach theoretical efficiencies of upto (90-100)%.

Applications

Below are the applications of power amplifiers across different sectors:

Consumer Electronics: Audio power amplifiers are used in almost all consumer electronic devices ranging from microwave ovens, headphone drivers, televisions, mobile phones and Home theatre systems to theatrical and concert reinforcement systems.

Industrial: Switching type power amplifiers are used for controlling most of the industrial actuator systems like servos and DC motors.

Wireless Communication: High power amplifiers are important in transmission of cellular or FM broadcasting signals to users. Higher power levels made possible because of power amplifiers increases data transfer rates and usability. They are also used in satellite communication equipment.