

# Applications of binary numbers



**ASSIGN  
BUSTER**

## **Introduction**

The world has evolved from simple mechanics to advanced quantum modeling. What has not changed is the Human will to count the entities. Humans originally created the Decimal Number System but because of the machine's and the Logic's inability to do complex functions on them, the Binary Number System were invented as a manifestation of Boolean Logic, where there can only exist two states – a True or a False, denoted by 1 and 0. All other states are defined by various combinations of these two primary states.

Boolean Logic was first invented by George Boole who first defined the algebraic system of logic [1] in the 1930s. The invention of such a system has had an immense breakthrough in the world of electronics and computers.

Since then these Tiny Binary Numbers (the two bits: 1 and 0) have had numerous applications- from storage of millions of entries of data to recording of HD Movies and High End Audio, from image processing to numerous other DSP (Digital Signal Processing) applications.

In this term paper, I shall mainly focus on relatively simple applications- A Binary Clock, A Subtractor, a Adder and other such simple applications.

## **What are Binary Numbers'**

' The binary numeral system or base-2 number system represents numeric values using two symbols, 0 and 1. More specifically, the usual base-2 system is a positional notation with a radix of 2' [2].

In simpler words, in Binary Number system all the possible values are expressed using two digits, 0 and 1. The 0 symbol also symbolizes a ' False' state while a 1 symbolizes a ' True' state.

It would be interesting to note that the Indian writer Pingala (200 BC) developed advanced mathematical concepts for describing prosody, and hence presented the world with the first known description of a binary numeral system. [3]

In our daily used Decimal Number system, we count items in the following manner we have the following 10 different symbols which can define 10 different entities.

The primary set of numbers are : {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.

All the complex numbers like 100, 1246 are further simply the combination of the 10 primary symbols. Every time the count gets greater than the 10 primary symbol set, we increase the position of our numbers (or Decimal place) and hence we get a new set of 10 more possible values.

E. g. The ' 13' symbol means the first 10 symbols as it is, then increasing the decimal place by 1 and then adding the first four symbols from the primary set.

Figure explaining how the Decimal Number System Works

In Binary System, since we only have two symbols (0 and 1) in our primary set, so our Decimal Place (or in this case the Binary place) is shifted by  $2^n$  factor, where n symbolizes the Binary Place.

Figure showing the Binary Number System

Certainly, the Binary Number System can get too much 'bulky' to express larger values (Binary equivalent of 5789 would be 1011010011101) but because this system has only 2 primary states ('True': 1 and 'False': 0) it can easily be understood by machines.

### **What is Boolean Logic'**

Now that we understand what is Boolean Algebra, it is time to have a look on the 'rules' or 'laws' defining how the symbols in this system would interact with each other. Boolean Logic does this. The basic of this type of logic is also comparison between two values.

There are 3 primary logics – The AND Logic, the OR Logic, and the NOT Logic.

AND: The AND logic means yield a value TRUE (1) if both the values in comparison have a 'True' value (1).

OR: The OR logic means a value of TRUE (1) is returned if either of the values in comparison have a 'True' value (1).

NOT: The NOT logic simply inverts a given value. For e. g. a it will change a True value to False and a False value to True if applied on a variable.

Of these 3 primary logics, only NOT can operate over a single variable, rest two need a minimum of two variables.

There are other logics to, like XOR, Ex-OR etc. but these can be obtained by the combination of these 3.

## Applications of Binary Numbers

Now that we know the basics of Binary Number system and the basics of Boolean Logic we can concentrate over the application part of Binary Numbers.

### Binary Adder

The simplest of all application is the Binary Number Adder. It adds two binary numbers and yields a binary result.

It basically produces the sum of two numbers (A and B) and takes care of the carry (if any). The following table shows the Truth Table of the Binary Numbers-Mathematically, a Full Adder's equation can be written as:

$$[(A \text{ XOR } B) \text{ XOR } C_{in}]$$

$$[(A \cdot B) + (C_{in} \cdot (A \text{ XOR } B))]$$

A Binary (Full) Adder can be constructed using various AND & OR gates. The circuit diagram is as follows:

Diagrams Showing Circuit Diagram of a Full Adder and Block Diagram working of a Full Adder

The output is obtained at two separate output pins- S (Sum) and Cout (Carry Out). If Cout is High (has a True Value; 1) then this means the addition of two Single Bit numbers have resulted into 3rd Bit (as is evident from the Truth Table).

Similarly we can construct a Full Subtractor by simply inverting one of the inputs (using NOT gate) and by feeding them to the given circuit.

## Binary Multiplier

Binary Multipliers have several uses and it has paved way to more complex systems like computers and other arithmetic systems.

We are going to discuss a 2 bit by 2 bit binary multiplier. The primary design of a multiplier consists basically of adders only.

Using long multiplication, a product of two N-bit numbers can be expressed as the sum of N N-bit partial products, which are then added to produce a 2N-bit product.

The partial products can be computed from the fact that  $a_i \cdot b_j = a_i \text{ AND } b_j$ .

The complexity of the multiplier is in adding the partial products.

In most of the complex adders the Partial products are added in pairs using several binary adders. The following diagram shows how a 2 bit by 2 bit Binary multiplier can be realized:

Figure showing Circuit Diagram of a 2 bit by 2 bit Binary Multiplier

The circuit above uses IC 74283 which is a 4 Bit Adder.

Each bit of one of the number's is multiplied with the each bit of the other number using an AND gate on by one. The product of each output is shifted 1 bit ahead (excluding the result from the first product). The resulting products are then later added using a 4 Bit Adder. And we get the result.

We are not going to discuss the multiplier in more detail because its working is at this stage beyond or comprehension.

## **Digital Binary Clock**

One of the many interesting applications of Binary Numbers is a digital clock.

A Digital Binary Clock consists of LEDs arranged separately for Hours, Minutes and Seconds as shown below-

The three components of Hours, Minutes and Seconds are separately controlled by timing circuits.

A LED in ON (lighted) state symbolizes a Binary 1 and on the contrary a LED in OFF state symbolizes a Binary 0.

There are three separate adders for each three components. The circuit diagram is built in such a manner that after each second, the adder attached with component depicting seconds is incremented by 1. If this addition has resulted in any carry then it is forwarded to the Minutes component. The adders of all three components are built in such a manner that the total result is never exceeded by 60 (in the case of Hour's component it is 24), if it exceeds the adder results in a carry and the sum is reset to 0. This carry is shifted to the just next in line adder. For example if the second's adder resulted in a carry, the carry is given to the minute's adder. However the case of Hour's adder the carry is neglected and the count is simply set to 0.

The corresponding output is then sent to the LED's displaying seconds as a Binary Output and we get the current time in Binary Numbers.

## **Conclusion**

In this Term Paper we have discussed briefly about the Binary Numbers and their applications. The modern day applications are far more complex than

the ones discussed in this Term Paper. They involve very complex circuits and can do a variety of jobs. There are devices like computers, mobile phones, video games handhelds, etc. which are far more complex but all work on these principles.

By no means can we underestimate the power of this number system. All the modern advances and even the future technologies will be based on this number system only.

In the end, I would only like to say that we have only made a small step towards a huge world of knowledge and science waiting to be explored.

### **References And Notes**

1. [http://en.wikipedia.org/wiki/Boolean\\_logic](http://en.wikipedia.org/wiki/Boolean_logic)
2. [http://en.wikipedia.org/wiki/Binary\\_numeral\\_system](http://en.wikipedia.org/wiki/Binary_numeral_system)
3. W. S. Anglin and J. Lambek, The Heritage of Thales, Springer, 1995, ISBN 0-387-94544-X
4. [http://en.wikipedia.org/wiki/Adder\\_\(electronics\)](http://en.wikipedia.org/wiki/Adder_(electronics))
5. Several information in this Term Paper was researched from various sources including internet websites and books.