

Prime Number Detector

**Group Number:** 10.

MD Foysal Kabir

ID: 21341053

Theory Section: 06

Shahriar Ahmed

ID: 20101588

Theory Section: 06

Manish Agarwala

ID: 17301120

Theory Section: 06

Nafia Afrin

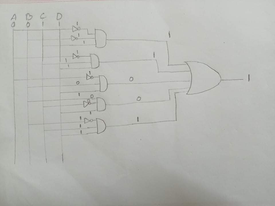
ID: 20101251

Theory Section: 06

# **Introduction:**

A Prime number is a positive integer which is also a natural number, is greater than one and is only divisible by 1 and the number itself. The first ten prime numbers are 2, 3, 5, 7, 11, 13, 17, 19, 23, 29. From Algebra, Discrete Mathematics, Number theories, to Computer algorithms, Prime numbers are avidly used as they serve many purposes and have some unique properties for factorization. One of those properties is that while it is relatively easy to find larger prime numbers, it’s unavoidably hard to factor large numbers back into primes. This fact makes primes vitally important to communications. Most modern computer cryptography works by using the prime factors of large numbers. The  encryption for public files works in such a way that only the prime factors of that large number can be used to decrypt it again. Thus, for their eminent relevance and multi-purpose uses in the fields of data communication, cryptography and computer mechanics, we chose to implement a Prime Number Detector.

## **Proposed Model/System:**



We are working on a project where we can easily detect on a circuit that the number is prime. Our circuit only works for 0-15 decimal numbers. As an example we are taking a number 3 which is a prime number,  because we know a number is a prime number when it is only divisible by 1 and the number itself. The binary equivalent of 3 is 0011. We are using 5 AND gate and 1 OR gate in our circuit where every AND gate has 3 inputs. On the first AND gate we are passing three bits as an input. When we pass A=0  through A NOT then it will give us 1 as an output and similarly if we pass B=0 through B not then it will give us 1 as an output, here C=1. We know from the three input  AND gate truth table if there is three input 1 then the output will be 1. Similarly if we pass A=0 through A NOT then the output will be reversed. Then, if we pass A NOT=1,B=1,C=1 through the AND gate then we will get 1 as output. Again on the third AND gate if we pass A=0 through A NOT then we will get 1. Then if we pass A NOT=1,B=0,C=1 then we will get 0 because we know from three input AND gate truth tables that if there is a single 0 in input then it will give us a 0 output. So, we will get 0. Similarly from the fourth AND gate we will get 0 as an output and from the fifth AND gate we will get 1.

Finally we got 1 from the first AND gate then again 1 from the second AND gate, 0 from the third one, 0 again from the fourth one and 1 from the fifth one. When we pass these five inputs through an OR gate it will give us a 1 as an output, because we know that if there is a multiple input and there is a one then the output will be 1 and our speaker will turn on.

## **Experimental Setup:**

## Components:

* AND
* OR
* NOT
* LOGICSTATE
* SPEAKER

## Circuit diagram :

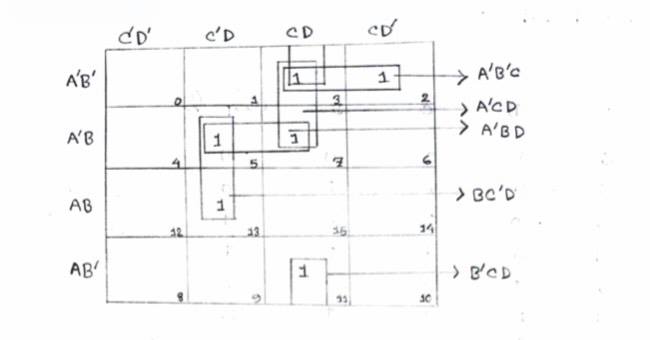
## **Result & Analysis:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | Output |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

Now, from the truth table,

F = **Σ**(2,3,5,7,11,13)

## **Using 4-variable K-map:**

****

****

Output = A’B’C + A’CD + A’BD + BC’D + B’CD

From the truth table, it can be seen that for the combinations in the truth table, if the combination represents a prime number then it will give 1 as output; otherwise it will give 0 as output. In the combinational circuit, when a 4-bit combination represents a prime number, it will give 1 as output and the speaker will turn on and make a sound to indicate the this 4-bit combination is a prime number; on the other hand, if the speaker remains turned off and do not make any sound then it can deduced that the 4-bit combination is not a prime number.

**Conclusion:**

This prime number detector will help us to detect any prime number between 0-15 (decimal numbers). But this prime number detector has a limitation. The prime number detector circuit only works if the number is between 0-15; if the number is greater than 15, due for higher number has more than 4 bit. So, our circuit cannot take the number as input.