**Abstract**

Our group has designed a 4 Bit Arithmetic Logic Unit (ALU) .An Arithmetic Logic Unit (ALU) is a combinational logic unit that performed its logical and arithmetical operations. It is getting smaller and more complex now a days to enable the development of a more powerful and smaller computer .In this report a 4 bit ALU has been designed using the radical gates which performed sixteen operation including reducing the disadvantage of a comparator.

**1. Introduction**

An ALU is the fundamental unit of any computing system. Understanding how an ALU is designed and how it works is essential to building any advanced logic circuits. Using this knowledge and experience, we can move on to designing more complex integrated circuits.

**2. Methodology**

1. **Equipment**

Software Name : Circuit Maker.

29 And gates : 7408,4082.

6 OR gates : 7432,4072.

6 NOT gates : 7404.

Total IC’s or Gates : 41 .

1. **Methods**

First we have drown the truth table to find Sum and Carry value. From the truth table draw the K-Map .After that from K-Map we find the functional equation of Sum and Carry .Then the functional equations we have drawn the 1 bit ALU circuit diagram then the 4 bit ALU circuit diagram.

**3. Function Derivation**

**Table 1 : Truth Table**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| S2 | S1 | S0 | Cin | A | B | Sum | Carry |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |  |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 |  |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 |  |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 |  |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 |  |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |  |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 |  |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 |  |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 |  |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |  |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |  |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 |  |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 |  |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 |  |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 |  |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 |  |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |  |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |  |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 |  |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 |  |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 |  |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 |  |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 |  |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 |  |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |  |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 |  |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |  |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 |  |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 |  |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 |  |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 |  |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |  |

**K-Map**



**For Sum,**

S2 S1=00

S2 S1=01

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A B  S0 Cin | 00 | 01 | 11 | 10 |
| 00 |  |  | 1 | 1 |
| 01 | 1 | 1 |  |  |
| 11 | 1 |  | 1 |  |
| 10 |  | 1 |  | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A B | 00 | 01 | 11 | 10 |
| 00 | 1 |  | 1 |  |
| 01 |  | 1 |  | 1 |
| 11 |  | 1 | 1 |  |
| 10 | 1 | 1 |  |  |

S0 Cin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A B  S0 Cin | 00 | 01 | 11 | 10 |
| 00 |  |  | 1 |  |
| 01 |  |  | 1 |  |
| 11 |  | 1 | 1 | 1 |
| 10 |  | 1 | 1 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S0 Cin | 00  A B | 01 | 11 | 10 |
| 00 |  | 1 |  | 1 |
| 01 |  | 1 |  | 1 |
| 11 | 1 | 1 |  |  |
| 10 | 1 | 1 |  |  |

S2 S1=10

S2 S1=11

Sum = S2’S1’CinA’B’+ S2’S1’Cin’AB’+ S2’S0’Cin’AB+ S2’S0’CinA’B+ S2’S0CinAB+ S0Cin’A’B+ S2’S1Cin’A’B’+ S2’S1S0CinB+ S1S0’CinAB’+ S2S1’AB+ S2S1’S0B+ S2S1’S0A+ S2S1A’B+ S2S1S0A’+ S2S1S0’AB’

**For Carry,**

S2 S1=00

S2 S1=01

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A B  S0 Cin | 00 | 01 | 11 | 10 |
| 00 |  |  |  |  |
| 01 |  |  | 1 | 1 |
| 11 |  | 1 | 1 | 1 |
| 10 |  |  | 1 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S0 Cin  A B | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | 1 |  |
| 01 |  | 1 |  |  |
| 11 |  |  |  |  |
| 10 | 1 | 1 |  |  |

Carry= S2’S1’CinA’ + S2’S1’S0CinB + S2’S1’S0AB + S2’S1Cin’A’ + S2’S1S0’Cin’B + S2’S1S0’A’B



Figure 1: 1bit ALU design

**Convert ALU 1 bit to 4 bit :**

There is our 1 bit ALU but we convert it to 4 bit ALU. At first we paste 1 bit ALU 4 times . There all selection bit (S1 S2 S0 Cin) are common bits and (A,B) are uncommon bits . Then we adding Cout of the 1st circuit to the Cin of the 2nd circuit using wire. And then Cout of the 2nd circuit adding to the Cin of the 3rd circuit and Cout of the 3rd circuit adding to the Cin of the 4th circuit. Now the circuit is ready to simulate.

**Discussion**

In our project we found some errors . At first when we made the 1 bit ALU then the logic combinations are not working properly. Then we fixed the error and again we made the circuit very sensitively. Now the circuit is properly working.

**Conclusion**

The arithmetic logic unit(ALU) is an important of computer CPU 's . We learn how to Produce different arithmetic operation and logic function by using various select singles for a single circuit. Actual operation speed of designed Alu is faster then specification. Low power consumption and small area. Preparation of an overall stick diagram of the entire circuit allowed quick, efficient organization of our layout. Great team work  helped us achieve our project goal.