**2-Bit Binary Adder Project Report**

**Introduction:**

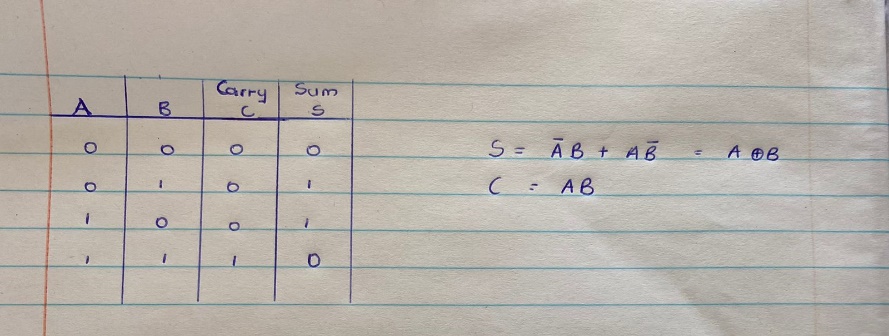
The project aims to create a digital circuit that adds 2-bit binary numbers and outputs a 3-bit binary numbers which includes the carry out. This is done utilizing two half adders and basic digital logic gates.

**Objective:**

The goal is to design a digital circuit that has the ability to add two 2-bit binary numbers and produce a 3-bit binary sum which also includes the carry out.

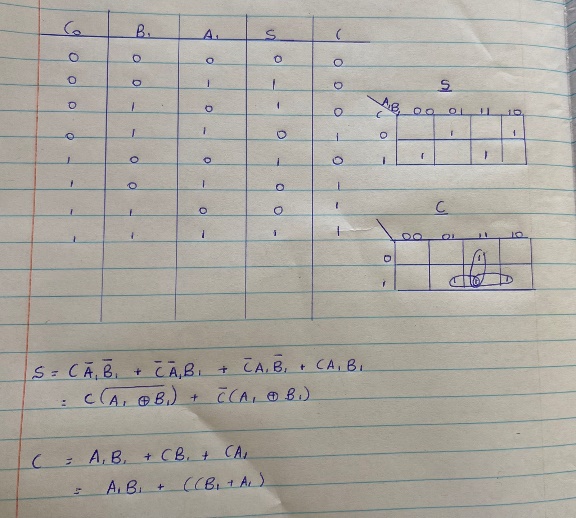
**Methodology:**

* The first half adder and the second half adder were established through the use of truth tables. The sum was implemented using XOR gates, and the carry using an AND gate.



**Figure 1.1**

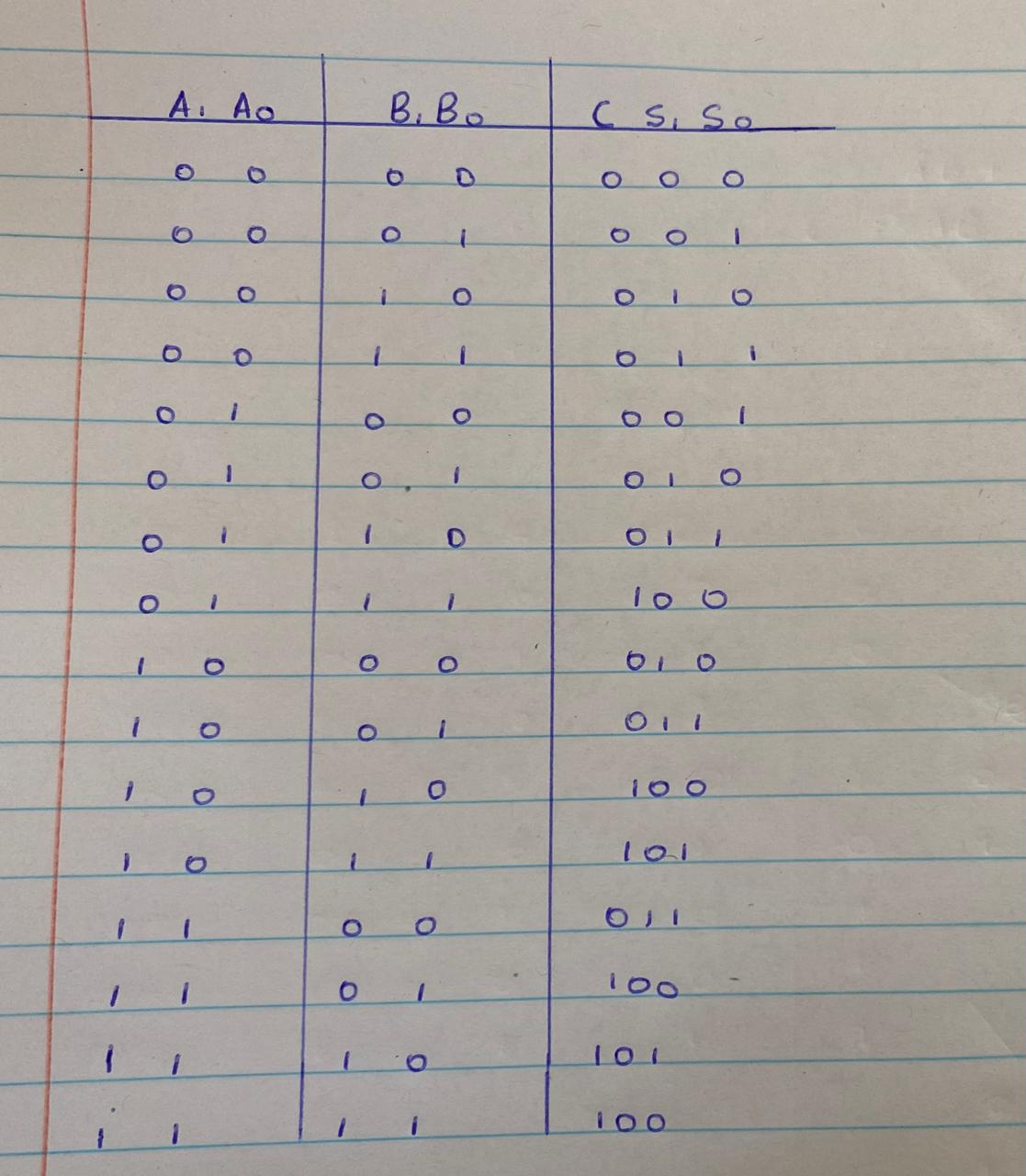
* To design the external gates, a truth table of the expected outputs was developed and the carry output was minimized using Boolean algebra and K-Map which are minimizing techniques.



**Figure 1.2**

**Testing:**

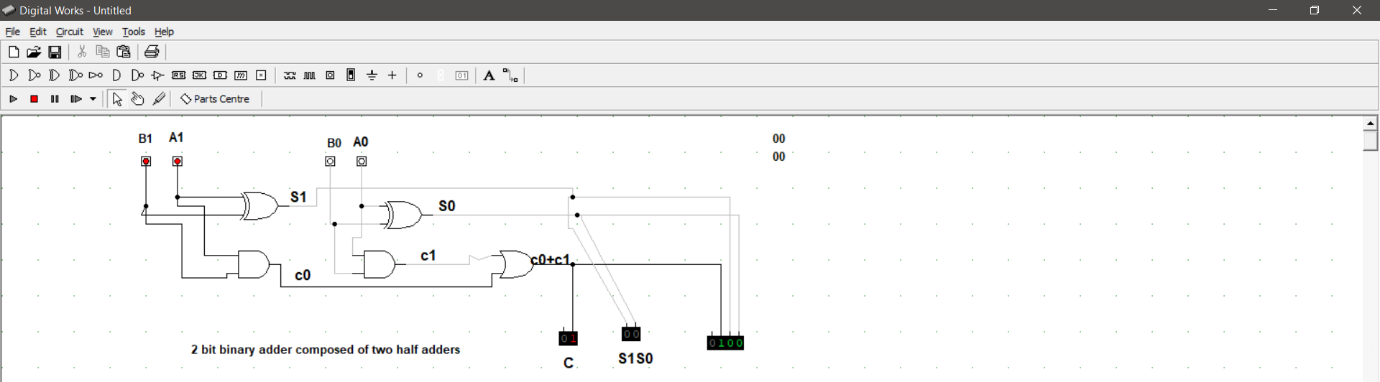
To test whether the circuit functioned correctly or not, a truth table was made showing all the possible combinations of two 2-bit binary numbers and their sums.



**Figure 1.3**

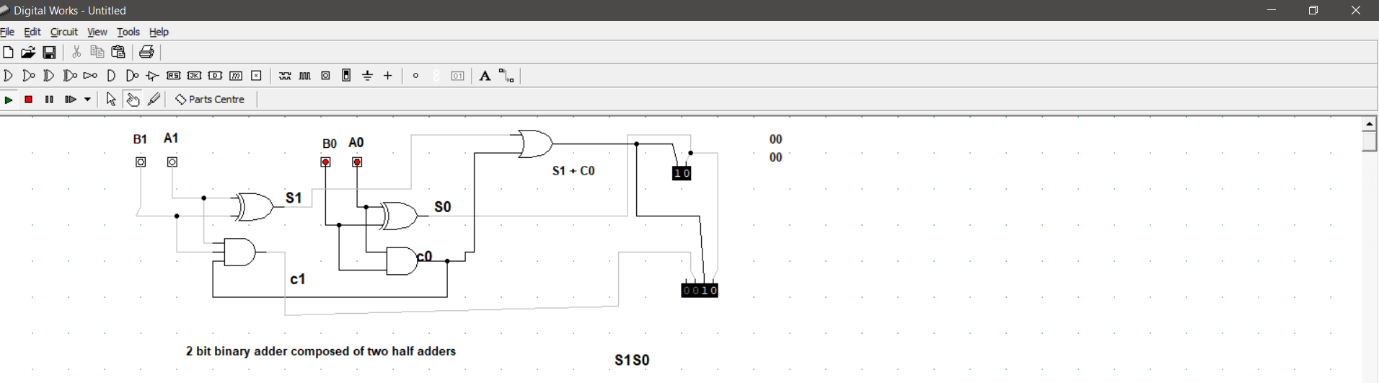
**Problems encountered:**

* There as an initial design where a problem was encountered with the carry out from the least significant bits of the two 2-bit binary numbers.



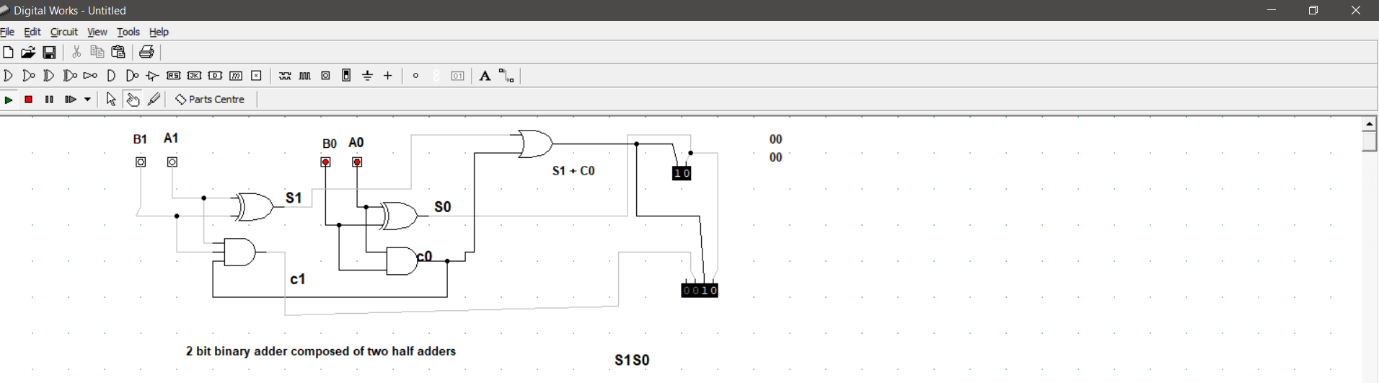
**Figure 1.4**

The issue was resolved by adding the carry out from the least significant bits to the sum of the most significant bits.



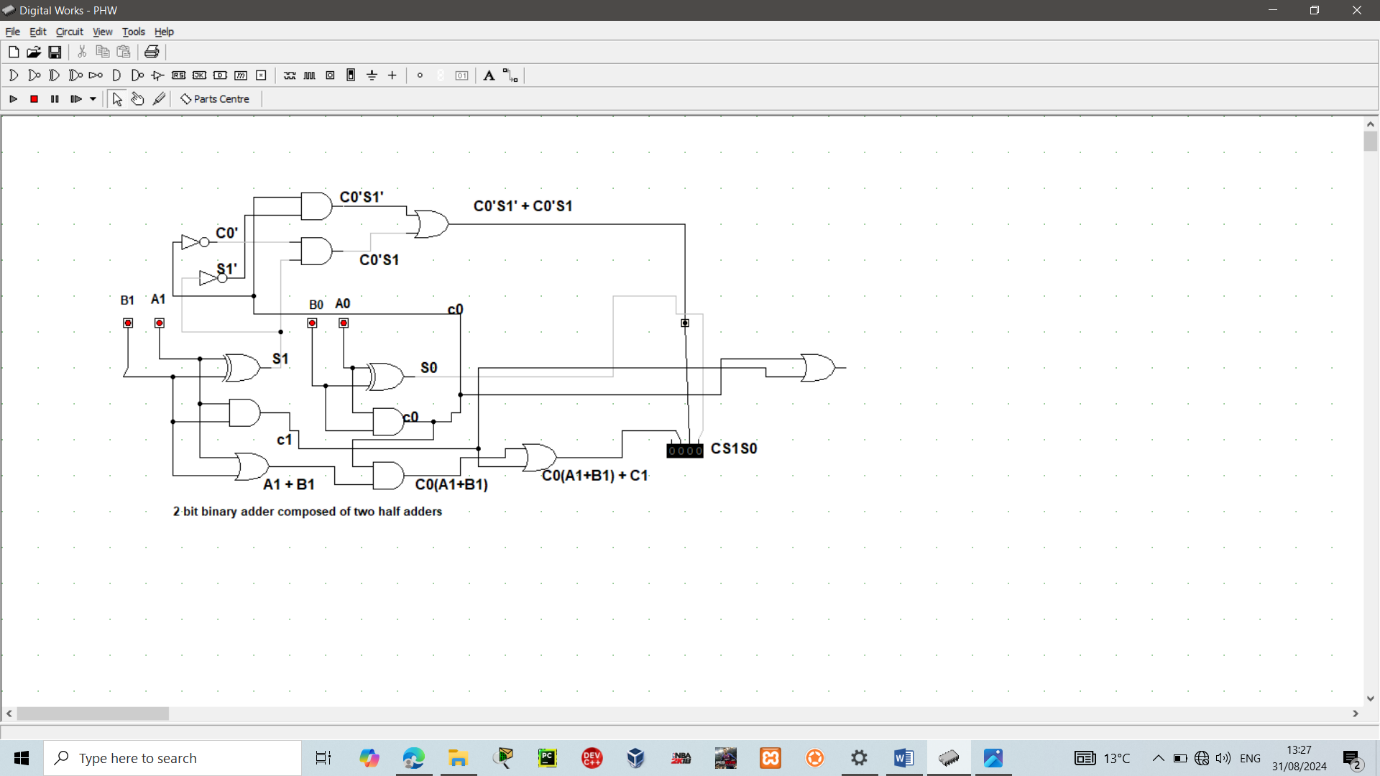
**Figure 1.5**

* However, this resolution led to another problem where the carry out from the most significant bits was not shown in the final sum.



**Figure 1.6**

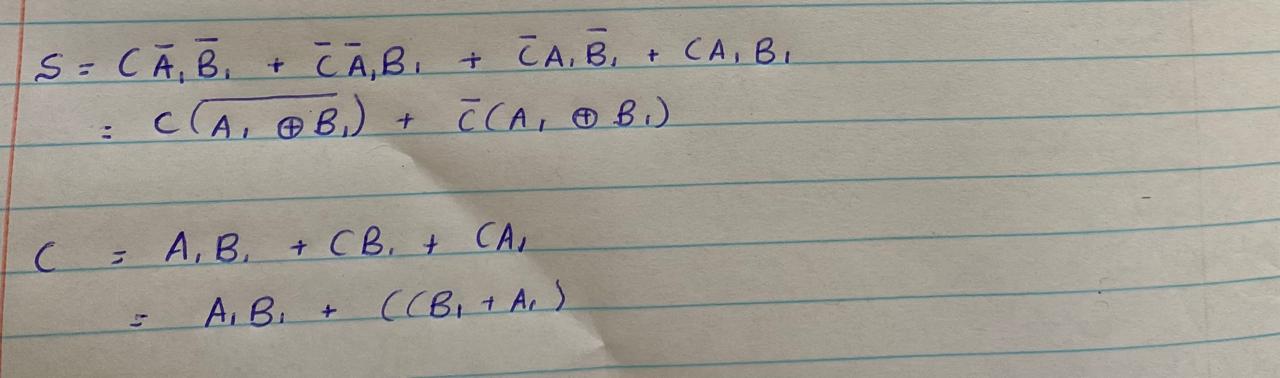
The challenge was countered by minimizing the carry out function from the truth table of the expected outcomes( see figure 1.3).



**Figure 1.7**

**The final solution:**

The solution of the last problem encountered was a gateway to the final solution that worked for all the possible combinations in contrast to the other resolutions which did not work for all the combinations. The Boolean expression that led to the ultimate working result.



**Figure 1.8**