**Number of factors**

**CP220 Project Phase I**

**Lubna Al Rifaie**

**200821590**

**Fall 2021**

**Description:**

The factors will be recognised by this circuit. It would be somewhere between zero and fifteen. When a number is a factor of another number, it means that the first number must completely divide the second number without leaving any residue. To put it another way, if a number If any number (divisor) is exactly divisible by (dividend), then the divisor is a factor of that number.  dividend. Each number has a common factor, which is one, as well as the number itself. So generally, this circuit will take binary numbers as inputs and produce binary numbers as outputs such as the number of elements that the input number has in form, including one and itself of binary digits.

**Objective:**

The goal of the Number of Factors project is to build a circuit that takes four binary inputs and outputs three values, giving the binary value of the factored number. If the input is 6, for example, the output should be 4 because the factors of 6 are 1,2,3, and 6.

**Process of the circuit:**

The Number of Factor circuit will take a binary number as an input through I1, I2, I3, and I4, convert it to decimal, and then divide the decimal input by all the numbers starting with 1 and ending with the input number itself. Any integers that divide the input evenly will be searched for by the circuit. When the circuit encounters such a number, it keeps track of all the numbers that are evenly divided by the input. When the circuit has reached the point where it has divided the input by itself, it will transform the result into a binary number and output the binary numbers through O1, O2, and O3.

**Input:**

The binary value would be the input, and the number of inputs would be four. The input would be in the form of a0 to a3, with a0 being the binary representation of a number and a3 being the binary representation of a number.

**Output:**

The output would be a binary value for the number of components that the input number has, including one and itself. There would be three outputs. The output would be in the form of b0 to b2, which represents a number in binary form.

**Error Condition:**

Some problems that we might encounter during the progress of completing the circuit will be Decimal to Binary and figuring out how to divide using the circuit. As a group, we need to focus and figure out a solution to these problems before we start building the finial project. For Ambiguous possibilities, have been eliminated we believe we don’t have any of now or doesn’t exist.

**Ambiguous possibilities:**

None

**A picture containing table

Description automatically generated**

**Number of factors**

**CP220 Project Phase II**

**Lubna Al Rifaie**

**200821590**

**Fall 2021**

**Truth Table:**

Making a table containing the inputs, outputs, and both of their binary equivalents might be useful for this specific scenario. This is depicted in Table 1 below.

**Table 1: Truth table**

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Input (a3a2a1a0) | Number of Factors | Output (b2b1b0) |
| 0 | 0 0 0 0 | **ERROR** | 0 0 0 |
| 1 | 0 0 0 1 | **1** | 0 0 1 |
| 2 | 0 0 1 0 | **2** | 0 1 0 |
| 3 | 0 0 1 1 | **2** | 0 1 0 |
| 4 | 0 1 0 0 | **3** | 0 1 1 |
| 5 | 0 1 0 1 | **2** | 0 1 0 |
| 6 | 0 1 1 0 | **4** | 1 0 0 |
| 7 | 0 1 1 1 | **2** | 0 1 0 |
| 8 | 1 0 0 0 | **4** | 1 0 0 |
| 9 | 1 0 0 1 | **3** | 0 1 1 |
| 10 | 1 0 1 0 | **4** | 1 0 0 |
| 11 | 1 0 1 1 | **2** | 0 1 0 |
| 12 | 1 1 0 0 | **6** | 1 1 0 |
| 13 | 1 1 0 1 | **2** | 0 1 0 |
| 14 | 1 1 1 0 | **4** | 1 0 0 |
| 15 | 1 1 1 1 | **4** | 1 0 0 |

The binary representations in the second and fourth columns of Table 1 are all that counts for logic equations, not what they represent. Three equations will be used, one for each output function. After that, we may use a Karnaugh map to generate a simpler SOP equation for each.

**Output b2:**

The binary representations for the inputs and output b2 are shown in the Table 2 truth table below.

**Table 2: Truth Table for b2:**

|  |  |
| --- | --- |
| Input (a3a2a1a0) | Output (b2) |
| 0 0 0 0 | 0 |
| 0 0 0 1 | 0 |
| 0 0 1 0 | 0 |
| 0 0 1 1 | 0 |
| 0 1 0 0 | 0 |
| 0 1 0 1 | 0 |
| 0 1 1 0 | 1 |
| 0 1 1 1 | 0 |
| 1 0 0 0 | 1 |
| 1 0 0 1 | 0 |
| 1 0 1 0 | 1 |
| 1 0 1 1 | 0 |
| 1 1 0 0 | 1 |
| 1 1 0 1 | 0 |
| 1 1 1 0 | 1 |
| 1 1 1 1 | 1 |

According to the table above, Table 3, the Karnaugh map for b2, may be generated.

Table

Description automatically generated**Table 3: Karnaugh Map Table for b2:**

Table 3 can then be outlined to produce Table 4, Table 5, and Table 6.

**Table 4: Karnaugh Map Table for b2 outlined a term:**

![Table

Description automatically generated]()

**Table 5: Karnaugh Map Table for b2 outlined** **a second term:**

**![Table

Description automatically generated]()**

**Table 6: Karnaugh Map Table for b2 outlined** **a third term:**

**![Table

Description automatically generated]()**

From these terms, we determine the final equation of the output b2 is:

**b2=a3 a´0+a3a2a1+a2 a1 a´0**

The terms in these tables are:

* a3 a´0
* a3 a2 a1
* a2 a1 a´0

**Output b1:**

The binary representations for the inputs and the output b1 are shown in the Table 7 truth table below.

**Table 7: Truth Table for b1:**

|  |  |
| --- | --- |
| Input (a3a2a1a0) | Output (b1) |
| 0 0 0 0 | 0 |
| 0 0 0 1 | 0 |
| 0 0 1 0 | 1 |
| 00 1 1 | 1 |
| 0 1 0 0 | 1 |
| 0 1 0 1 | 1 |
| 0 1 1 0 | 0 |
| 0 1 1 1 | 1 |
| 1 0 0 0 | 0 |
| 1 0 0 1 | 1 |
| 1 0 1 0 | 0 |
| 1 0 1 1 | 1 |
| 1 1 0 0 | 1 |
| 1 1 0 1 | 1 |
| 1 1 1 0 | 0 |
| 1 1 1 1 | 0 |

According to the table above, Table 8, the Karnaugh map for b1, may be generated.

**Table 8: Karnaugh Map Table for b1:**

**Table

Description automatically generated**

Table 8 may then be used to outline groups in Table 9, and Table 10 can be used to outline groups in Table 10.

**Table 9: Karnaugh Map Table for b2 outlined** **two terms:**

**![A picture containing text, electronics

Description automatically generated]()**

**Table 10: Karnaugh Map Table for b2 outlined** **two more terms:**

**![A picture containing text, electronics

Description automatically generated]()**

From these terms, we determine the final equation of the output b2 is:

**b1=a2 a´1+a´3 a´2 a1+a´3 a1 a0+a3 a´2 a0**

The terms in these tables are:

* a2 a´1
* a´3 a´2 a1
* a´3 a1 a0
* a3 a´2 a0

**Output b0:**

The binary representations for the inputs and output b0 are shown in the truth table below (Table 11).

**Table 11: Truth Table for b0:**

|  |  |
| --- | --- |
| Input (a3a2a1a0) | Output (b0) |
| 0 0 0 0 | 0 |
| 0 0 0 1 | 1 |
| 0 0 1 0 | 0 |
| 0 0 1 1 | 0 |
| 0 1 0 0 | 1 |
| 0 1 0 1 | 0 |
| 0 1 1 0 | 0 |
| 0 1 1 1 | 0 |
| 1 0 0 0 | 0 |
| 1 0 0 1 | 1 |
| 1 0 1 0 | 0 |
| 1 0 1 1 | 0 |
| 1 1 0 0 | 0 |
| 1 1 0 1 | 0 |
| 1 1 1 0 | 0 |
| 1 1 1 1 | 0 |

According to the table above, Table 12, the Karnaugh map for b0, may be generated.

**Table

Description automatically generatedTable 12: Karnaugh Map Table for b0:**

Table 12 may then be used to outline groupings to create Table 13.

**Table 13: Karnaugh Map Table for b2 outlining two terms:**

**![A picture containing text, electronics

Description automatically generated]()**

From these terms, we determine the final equation of the output b2 is:

**b0=a´2 a´1 a0+a´3 a2 a´1 a0**

The terms in Table 13 are:

* a´2 a´1 a0
* a´3 a2 a´1 a0

**Equation Testing:**

Maxima was used to test the equations.

**Output b0:**

**Text

Description automatically generated**

**Because this fits the truth table, the b2 equation is valid.**

**Output b1:**

**Table

Description automatically generated with low confidence**

**Because this fits the truth table, the b2 equation is valid**.

**Text

Description automatically generatedOutput b2:**

**The equation for b2 is accurate since it fits the truth table.**

**Conclusion:**

The output equations are as follows:

**b2 = a3 a´0 + a3a2a1 + a2 a1 a´0**

**b1 = a2 a´1 + a´3 a´2 a1 + a´3 a1 a0 + a3 a´2 a0**

**b0 = a´2 a´1 a0 + a´3a2 a´1a0**

Therefore, these equations have been thoroughly examined and found to be accurate.

Graphical user interface, application

Description automatically generated