Chi Function

$$y = AB + AC'E' + AC'D + A'B'CE' + A'B'CD + ACD'E + A'B'C'D'E$$

Where A = The First bit of the of the row

B = Second bit of the row

C = Third bit of the row

D = Fourth bit of the row

E = Fifth bit of the row

The technique that we employed for obtaining the pre-image is as below:

1. The first five bits of the row will produce the first output bit (pre-image bit).
2. To get the second bit of the result we will use next 5 bits of the row. The thing to be considered here is that the bits will get wrapped around and will take the first bit of the row as the last bit for the second output (Shifting the bits by 1 and this will be done for every other output i.e. for the third output it will get shifted by 1 again). The said string will produce the second bit of the result.
3. Continue the process for 5 bits in the row which will produce the five bits (Repeat the steps 1 and 2 by shifting the bits of the row by 1).
4. Now take another row and perform the same process.

Below is an example given for the process that was carried out to calculate the process.

The thing to notice for these equations is that one particular input will always produce the same output. It will never collide and so when we prepare a truth table for the given example we can get the equation as mentioned above.

The instance that are not covered in this example are (11111) & (00000). After observing the dataset, the input will produce 1 & 0 output respectively.

Note: Please refer to the program code for verification and for detailed analysis of the steps performed for the calculation.

⇒ Chi function.

⇒ Input ( f3 37 11 6a 4f )
⟶ Output ( f3 fd 10 da 43 )

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| F3 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 37 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 11 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 6a | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 4f | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

Output

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| f3 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| fd | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 10 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| da | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 43 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |

| Input | | | | | Desired Output |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |

| Input | | | | | Desired Output |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 |