Truth table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | c |  |  |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Now we try to write the equations based on the truth table: (I will use + instead of | to represent OR and . instead of & to represent AND after writing the first step in order to be able to read it/write it easier)

= ~a & ~b & c | ~a & b& ~c | a& ~b& ~c | a &b & c

= ~a.(~b.c+b.~c) + a.(~b.~c+b.c)

= ~a & b & c | a & ~b & c | a & b & ~c | a & b & c

----we know that a+a = a so we can write as many

a’s as we want for one represents all of them in this

equation--🡪

= ~a.b.c + a.~b.c + a.b.~c + a.b.c + a.b.c + a.b.c

= b.c(~a+a) +a.c(b+~b) + a.b(c+~c)

-----we know that a+~a = 1 and that bc.1= 1 --🡪

= b.c + a.c + a.b

Based on the looks of it this is as far as we can go.

Lets try drawing the K-maps just to make sure:

(the horizontal line is a,b and the vertical one is c)

K-map :

K-map :

We will try to find the physical adjacencies and see if they match the Boolean ones we found:

\*\*K-map :

So lets write the minimal function base on the K-map: all three shown here are EPIs because they each have exactly one ‘1’ that is not repeated in any other:

= bc + ab +ac

\*\*K-map :

As it can be seen in the diagram there are no adjacencies for us to show so we will just have to write the ‘1’s as we see them:

= ~a~b.c + ~a.b.~c + a.b.c + a.~b.~c = ~a.(~b.c+b.~c) + a.(~b.~c+b.c)

as we can see the results are the same both times (K-map and Boolean) so this should very well be true and we can move on.