Work done till now :-

Developed the layout and the steps for converting a logical expression to the most efficient

2-input NAND gate circuit.

Developed algorithm for expressing the most efficient 2-input NAND gate circuit

for N-input NAND, AND and OR gate in terms of delay.

Changed the logical expression into prefix format using python.

Please note that the coding part is still left. I have just developed the algorithm

Future Work :-

Using the prefix format, I will express the circuit in logical format using classes defined for logic gates in python.

The components of this circuit will be placed such that the total length of wire used in the circuit is minimum.

(Placement Problem)

And use this circuit for displaying the final placement of the efficient logic circuit obtained from the logical expression

The below discription explains the algorithm of expression the most efficient 2-input NAND gate circuit
for N-input NAND, AND and OR gate in terms of delay.

Define:-

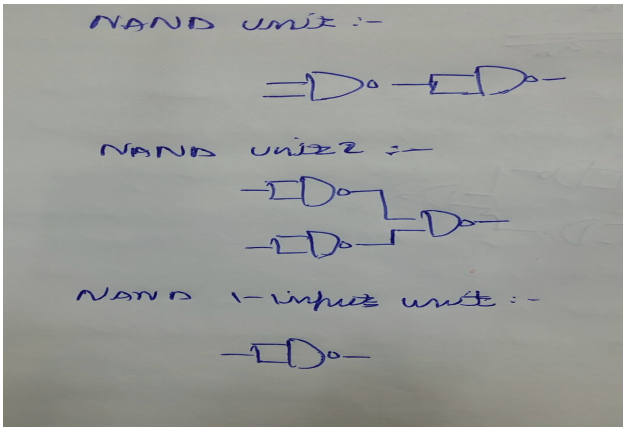
Pure input :- When the input to a gate is a variable and ins't the output of any other gate

NAND unit :- 2 NAND gates connected where the first NAND gate's output is the input to the second NAND gate.

NAND unit2 :- 3 NAND gates connected where first and second NAND gates output is connected to the inputs of the third NAND gate.

The input of the first NAND gate is connected to one pure input. Similarly, the second NAND gate and the second pure input.

NAND 1-input unit :- A NAND gate whose input are connected to only one entity (a single pure input or a single output of a circuit block)



If I refer to a NAND gate, it should be assumed that I am talking about 2-input NAND gate.

For N input NAND gate :-
Consider N-1 input NAND gate in terms of NAND gate.

First neglect all NAND gates whose inputs are from the output of the same gate for further considerations.

If the 2 inputs of all the NAND gates are either pure inputs or outputs of other NAND gates,
then choose one NAND gate which is closest to the final output and consists of only pure inputs (pcNAND)
and connect a NAND unit (newNANDunit) to one of the inputs of pcNAND. The inputs of newNANDunit will be the
nth and (n-1)th pure input and the other input of pcNAND will be the (n-2)th pure input.
No other connections or positions of pure inputs will change. Period.

Else if there exist a NAND gate whose inputs are a pure input and an output of another gate, then
simply connect a NAND unit (newNANDunit) to where the pure input was connected and newNANDunit's inputs will
be the (n)th and (n-1)th input.
No other connnections or positions of pure inputs will change.

We have to store in memory 3 input NAND gate only for the above recursion. This recursion will give the
most efficient circuit in 2-input NAND gates in terms of delay.

For N input AND gate :-
Consider N-1 input AND gate in terms of NAND gate.

In the circuit, we don't see NAND gates individually but only as blocks of NAND units connected to one another.

If the 2 inputs of all the NAND units are either pure inputs or output of other NAND units,
then choose one NAND gate which is closest to the final output and consists of only pure inputs (pcNANDunit)
and connect a NAND unit (newNANDunit) to one of the inputs of pcNANDunit. The inputs of newNANDunit will be the
(n)th and (n-1)th pure input and the other input of pcNANDunit will be the (n-2)th input.

Else if there exists a NAND unit whose inputs are a pure input and output of another NAND unit, then
simply connect a NAND unit (newNANDunit) to where the pure input was connected and newNANDunit's inputs will
be the (n)th and (n-1)th input.
No other connnections or positions of pure inputs will change.

We have to store in memory 2 input AND gate only for the above recursion. This recursion will give the
most efficient circuit in 2-input NAND gates in terms of delay.

For N input OR gate :-
Consider N-1 input OR gate in terms of 2 input NAND gate.

In the circuit, we don't see NAND gates individually but only the NAND unit2 block (has 2 inputs and 1 output)

If the 2 inputs of all the NAND unit2 are either pure inputs or output of other NAND unit2s,
then choose one NAND gate which is closest to the final output and

consists of only pure inputs (pcNANDunit2)
and connect a NAND unit (newNANDunit2) to one of the inputs of pcNANDunit2. The inputs of newNANDunit2 will be the
(n)th and (n-1)th pure input and the other input of pcNANDunit2 will be the (n-2)th input.

Else if there exists a NAND unit2 whose inputs are a pure input and output of another NAND unit2, then
simply connect a NAND unit2 (newNANDunit2) to where the pure input was connected and newNANDunit2's inputs will
be the (n)th and (n-1)th input.
No other connnections or positions of pure inputs will change.

We have to store in memory 2 input OR gate only for the above recursion. This recursion will give the
most efficient circuit in 2-input NAND gates in terms of delay.