EE 5811 : FPGA LAB
Challenge Problem 2

Shantanu Yadav (EE20MTECH12001)

Download the codes from

https://github.com/Shantanu2508/mtech/tree/master/FPGA_LAB/CP2

1 PROBLEM STATEMENT

Is it possible to implement the NAND logic using some data structure/algorithm?

2 THEORY

A NAND (NOT-AND) gate produces a False output only if all the inputs are True. The truth table is as follows The NAND gate is a universal gate

X	Y	OUT
0	0	1
0	1	1
1	0	1
1	1	0

TABLE 0: Truth table for NAND gate

because any boolean function can be implemented by using a combination of NAND gates. One interesting property of NAND logic is that if any of the input is LOW the corresponding output is HIGH. This logic can be realised using stack data structure. Stacks are abstract data structures based on LIFO (Last In First Out) technique. The two main principal operations are push and pop. Push operation adds an element to the stack and pop operation removes an element from the top of the stack.

To implement NAND logic we first create an empty stack. The input to the NAND gate is assumed to be an array of size $N \times 2$, where N are the total number of input combinations. For each input stream, X is pushed into the stack and compared with Y. If any of them is 0 the output is 1 otherwise if the top element of stack and the Y input are both equal to 1 then output is false.

Algorithm 1 NAND Logic using stack data structure

Require: Array of size $N \times 2$
Ensure: Create an empty stack S

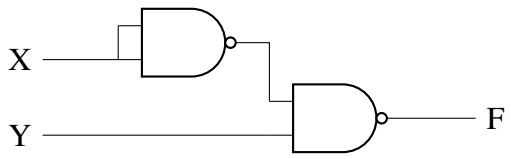
```
i ← 0
while i ≤ N − 1 do
    S ← input[i][0]
    if S.top() == 0 then
        output[i] ← 1
        i ← i + 1
    else if input[i][1] == 0 then
        output[i] ← 1
        i ← i + 1
    else
        output[i] ← 0
        i ← i + 1
    end if
    S.pop()
end while
```

3 EXAMPLE

Implement the boolean expresion given below using NAND gates

$$F(X, Y) = X + \bar{Y} \tag{1}$$

The logic circuit is realised using NAND gates as in shown figure below



If X and Y are sets containing only 0 and 1 then the input stream to the NAND logic can be defined as,

$$S = \{\ (x, y) \mid x \in X,\ y \in Y\ \} \tag{2}$$

where each (x, y) pair is an input to the NAND gate at any time instant. The NAND logic can be

X	Y	F
0	0	1
0	1	0
1	0	1
1	1	1

TABLE 0: Truth Table for $F = X + \bar{Y}$. (Verified using code)

interpreted as a function $g : S \to \{0, 1\}$ defined as,

$$g(x, y) = \begin{cases} 0, & x = 1 \text{ and } y = 1 \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

Thus from the above figure, the boolean expression can be expressed as

$$F = g(g(x, x), y) \tag{4}$$

Each time when function g is invoked, the x value of the first input is pushed into the stack. If $x = 0$ the output is 1 regardless of y, but if $x = 1$ the output is 1 if $y = 0$ otherwise the output is 0. After this x is removed from the stack and the procedure repeats for other inputs.