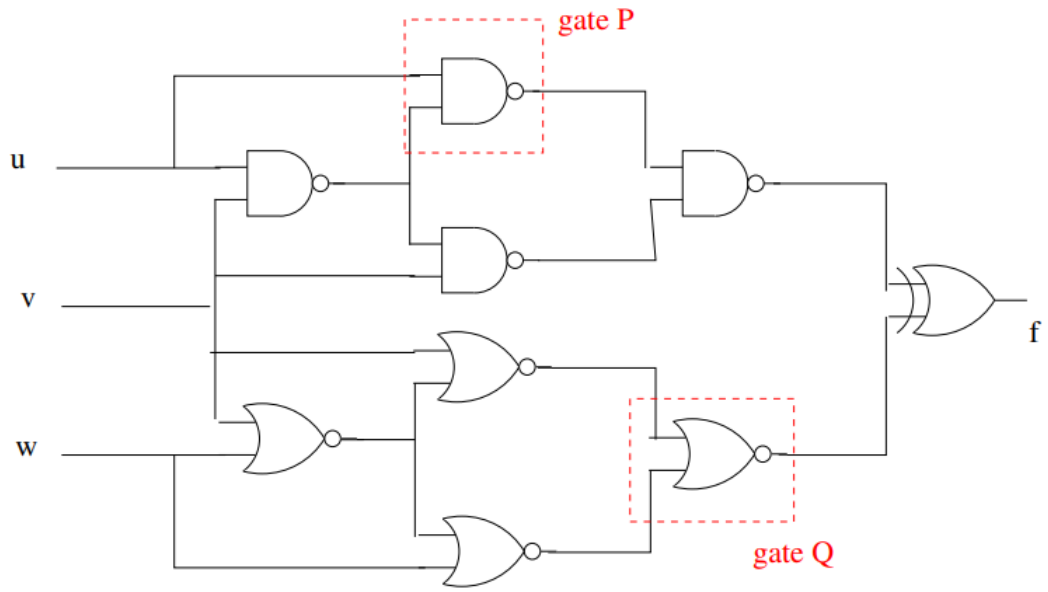
QUESTION 3

193079024

SARVESH KALE



There are two faults given the gate P behaving as NOR and Gate Q behaving as NAND .I am using +,.,~,^ to denote sum ,product ,negation,xor respectively

X1	X2	NOR	NAND
0	0	1	1
0	1	0	1
1	0	0	1
1	1	0	0

The inputs to **the gate P** has expression u and ~(u.v) = ~u + ~v so we see that the two inputs to that gate can never be same . so the first and last line of truth table

never occur ,but the gate is behaving as NOR so the output of that gate must be **stuck at 0.**

The inputs to the **gate Q** are the expressions v.~w and ~v.w ,these two expressions ,again the inputs are high only for 1,0 and 0,1 combination and for those combination the output of NOR should have been 0 but since it is behaving as NAND it is 1 so it **is stuck at 1 fault** here .

So the problem has been reduced to s_a_0 at output of gate P and s_a_1 at output of gate Q, we will consider these cases one at a time.

1.)s_a_0 at output of GATE P

Let us call this node as a.

So **f=a= ~(u . ~(u.v)) = ~u + v** ……..**eq1**

Output should be computed in terms of the above node a and inputs u,v,w

Let us call it g

g = ~(a . ~(~(u.v).v)) ^ (~(v ^w))

by looking at the bottom NOR arrangement .it is an XNOR gate so xnor of v and w

now ,

g(a = 0) = 1 ^((~(v ^w)) = (v ^w)

since we xor anything with 1 ,output is negation of it .

g(a = 1) = ~ ~(~(u.v) .v) ^ (~(v^w))

the two negations cancel out

g(a =1) = ~(u.v) .v ^(~(v^w)) .

now g(a=0) ^ g(a=1) = (v ^w) ^ ~(u.v) .v ^(~(v^w))

the first and last are negations of each other as they are xnor and xor of the same two variables .so their output is 1

g(a=0) ^ g(a=1) = ~(u.v) .v ^ 1 = ~(~(u.v) .v) = u.v + (~v) = (u + (~v)).(v + (~v))

= u + (~v) so

g(a=0) ^ g(a=1) = u + (~v) ……eqn2

This expression tells that the boolean difference w.r.t the node a is non-zero and that a change at this node will reflect at the output .

Test pattern = f.(g(a=0) ^ g(a=1))

= (~u + v) .(= u + (~v)) …….from eqn 1 and 2

This is the CNF form of the test input vector .

---

c this is for s_a_0 for upper nand gate

c u v w = 1 2 3

p cnf 3 2

-2 1 0

-1 2 0

When we feed it to the SAT solver .we get the following output

I am using the picosat SAT solver .which enumarates all the output when the flag –all is passed to it .If minisat_static is used ,on the first run it generates an output

Which is assignment of input variables for which CNF is true ,so in the next run we do not want this output so we want the negation of the assignment obtained to be true so just negate the output sequence obtained and append that line at the end of the cnf file and rerun again the SAT solver and keep doing this untill you get UNSAT

The output obtained is as follows

```
sarvesh:q3$ echo w can be anything >> q3uppernandoutput.txt
sarvesh:q3$ cat q3uppernandoutput.txt
s SATISFIABLE
v -1 -2 0
s SATISFIABLE
v 1 2 0
s SOLUTIONS 2
w can be anything
```

The variables 1 is u ,2 is v so the output does not have variable 3 which is w so w can be anything

Test vectors (u,v,w)= (0,0,1),(0,0,0),(1,1,1),(1,1,0).

When we apply these test vectors ,we see that there is a conflict at node a and that conflict gets reflected at output .

2.)s_a_1 at the output of NOR

Since it is s_a_1 ,

Let us call the output of this NOR as a .

We will have to write an expression of when that node is 0 that is an expression for conflict condition .

We know the arrangement of NOR gates function as XNOR ,so

Let **f=a = w ^ v** which is not of ~(w^v).

let us call output g ,and this output should be function of node a, and input .

g = u ^v ^ (a)

The upper arrangement of gates is xor .

g(a=1) = u^v ^1 = ~(u^v)

g(a=0) = u^v^0 = u^v

xor the above two equations to get the boolean defference of output w.r.t this node a

g(a=1) ^ (ga=0) = ~(u^v) ^ u^v = 1 .

f.(g(a=1) ^ (ga=0)) = (w ^v) . 1

=(~w + ~v) .(w + v)

The above is the CNF of input test pattern ,give this in DIMACS format to a SAT solver picosat ,it enumarates all the output when the flag –all is supplied to it

---

c this is for s_a_1 fault for lower NOR

c v w u = 1 2 3

p cnf 3 3

-1 -2 0

1 2 0

-3 3 0

The output is

```
sarvesh:q3$ cat q3lowernoroutput.txt
s SATISFIABLE
v -1 2 0
s SATISFIABLE
v 1 -2 0
s SOLUTIONS 2
u can be anything
```

So the output test vectors are

(u,v,w) = (0,1,0),(1,1,0),(0,0,1),(1,0,1)

We see that when we apply the above test vector we get a conflict at node a and also the output so that is how we can detect this fault.