

【1】

stuck-at-fault 1 の場合:=====

NAND ゲート G の出力 e 以外には故障が生じていないという仮定の下で, e が常に 1 になってしまう故障があるか否かを判定するには, e に 0 を出力し, かつ e の値を出力まで伝搬させるような入力を与えればよい. すなわち, 回路が正しければ, $e = 0$ によって決定される出力が観測され, 故障があれば, $e = 1$ によって決定される出力が観測されるようにすればよい.

$e = 0$ にするためには, x および y は共に 1 でなければならない. 今, 出力 w を出す NAND ゲートの上側の入力を u ($u = \overline{x \cdot e}$), 下側の入力を v ($v = \overline{y \cdot z}$) とすると, $x = 1$ であれば, $u = \overline{e}$ であり, $y = 1$ であれば, $v = z$ である. 従って, $w = \overline{e \cdot z}$ となるから, w に e の値を伝搬するには, $z = 1$ でなければならない. すなわち, u の値を w に伝搬する (u の値によって w が決まる) ようにするには, v に NAND ゲートの非制御値 1 を入力しなければならない.

そこで, $x = y = z = 1$ とすると, 故障が無ければ, $w = e = 0$ になり, stuck-at-fault 1 の故障があれば, $w = e = 1$ になる.

stuck-at-fault 0 の場合:=====

この場合には, e に 1 を出力し, かつ e の値を出力まで伝搬させるような入力を与えればよい.

$e = 1$ にするためには, x あるいは y の少なくとも 1 つが 0 でなければならないが, $x = 0$ であれば, e の値に関わらず $u = \overline{x \cdot e} = 1$ になってしまい, e の値を出力まで伝搬させることができない. 従って, $x = 1$ としなければならない. すなわち, 他の入力の値を出力に伝搬させるには, 非制御値を入力する必要がある.

今, $x = 1$ としたので, $e = 1$ にするには, $y = 0$ でなければならない. $x = 1, y = 0$ とすると, 故障が無ければ, $e = 1, u = 0$ になり, 故障があれば, $e = 0, u = 1$ になる.

そこで, この u の値の違いを w に伝搬させるには, v には非制御値 1 を入力しなければならないが, $y = 0$ であるから, z の値に関わらず, $v = 1$ になることが分かる.

従って, z の値が何であっても, $x = 1, y = 0$ とすることにより, e に stuck-at-fault 0 の故障があるか否かが分かる. すなわち, stuck-at-fault 0 の故障が無ければ, $e = 1, u = 0, w = 1$ となり, stuck-at-fault 0 の故障があれば, $e = 0, u = 1, w = 0$ となる.