

EE 709 Assignment 2 Roll: 20D170033

① Encoding '0' - 00, '1' - 11, 'X' - 01
and 'Z' - 10.

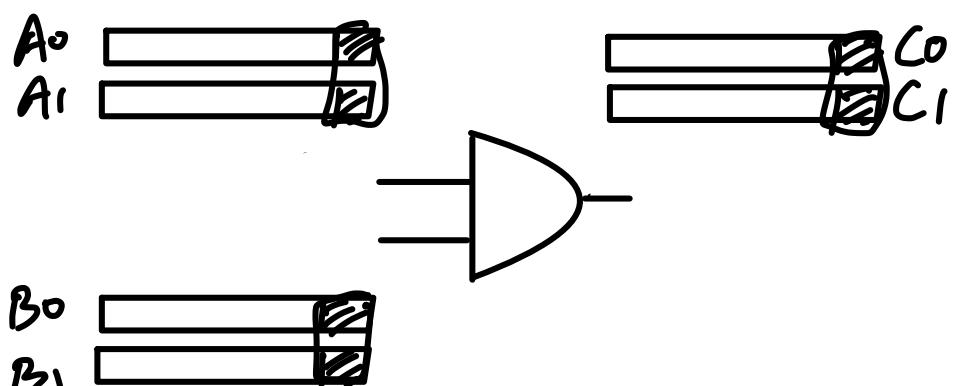
let $A_0 A_1, B_0 B_1$, be the input bits to the AND gate, $C_0 C_1$ be the output bits

0 - 0 0

1 - 1 1

X - 0 1

Z - 1 0



$A_0 A_1 \backslash B_0 B_1$	00	X	1	Z
0 00	00	00	00	d.c
X 01	00	01	01	d.c
1 11	00	01	11	d.c
Z 10	d.c	d.c	d.c	d.c
				don't care
				C ₀ C ₁

taking the k-map for c_0 and c_1 independently

KMAP for the bit c_1 .

		$A_0 A_1$	0	X	1	Z
		$B_0 B_1$	00	01	11	10
A_0	B_0	00	0	0	0	X
X	01	0	0	1	1	X
1	11	0	1	1	1	X
Z	10	X	X	X	X	X

$$c_1 = a_1 \cdot b_1 = a_1 \text{ } \& \text{ } b_1$$

where ' $\&$ ' denotes bitwise AND

KMAP for the bit C_0

$A_0 A_1$	0	X	1	Z
$B_0 B_1$	00	01	11	10
0	00	0	0	0
X	01	0	0	X
1	11	0	0	1 X
Z	10	X	X	X X

$$C_0 = a_0 \cdot b_0 = a_0 \& b_0$$

where ' $\&$ ' denotes bitwise AND

Thus the minimal boolean expression for the C-like word operation for inputs $A_0 A_1, B_0 B_1$ to AND gate is

$$C_0 = A_0 \& B_0 \quad C_1 = A_1 \& B_1$$

② we have to generate a test vector for k stuck-at-0 using PODEM.

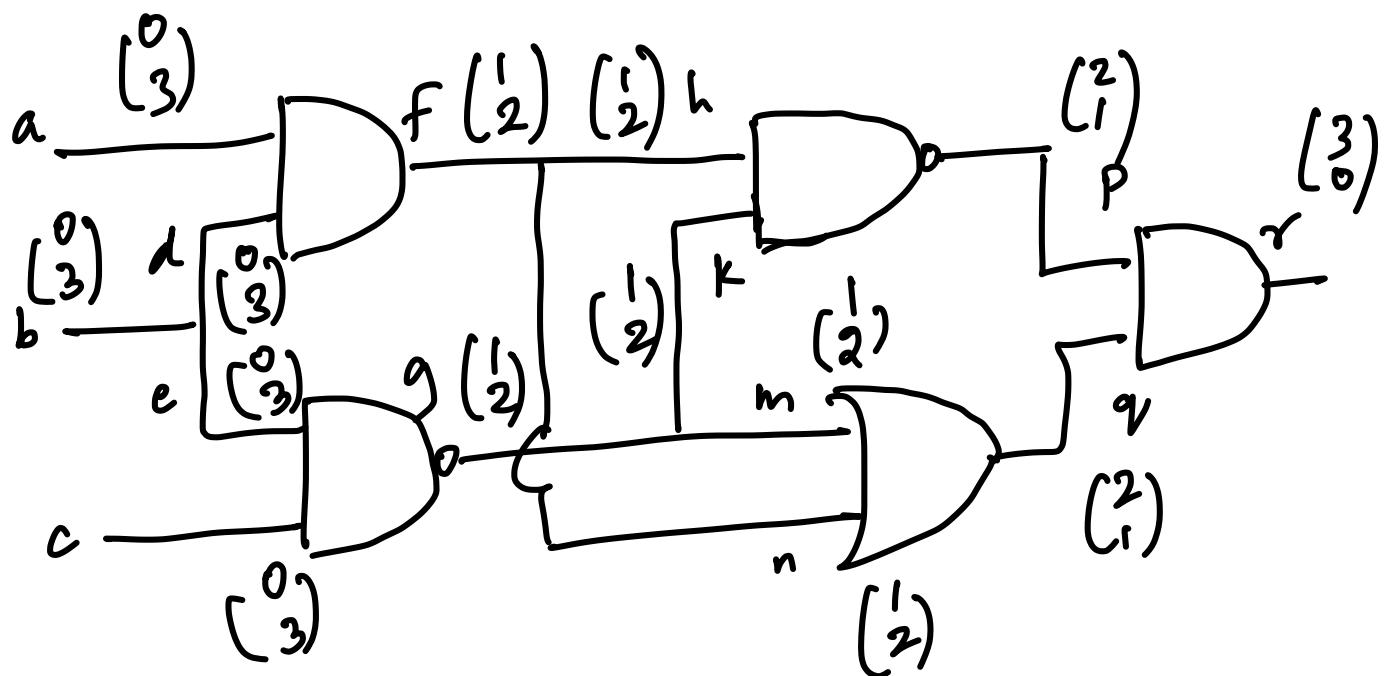
as suggested by question we define for a node N , controllability and observability

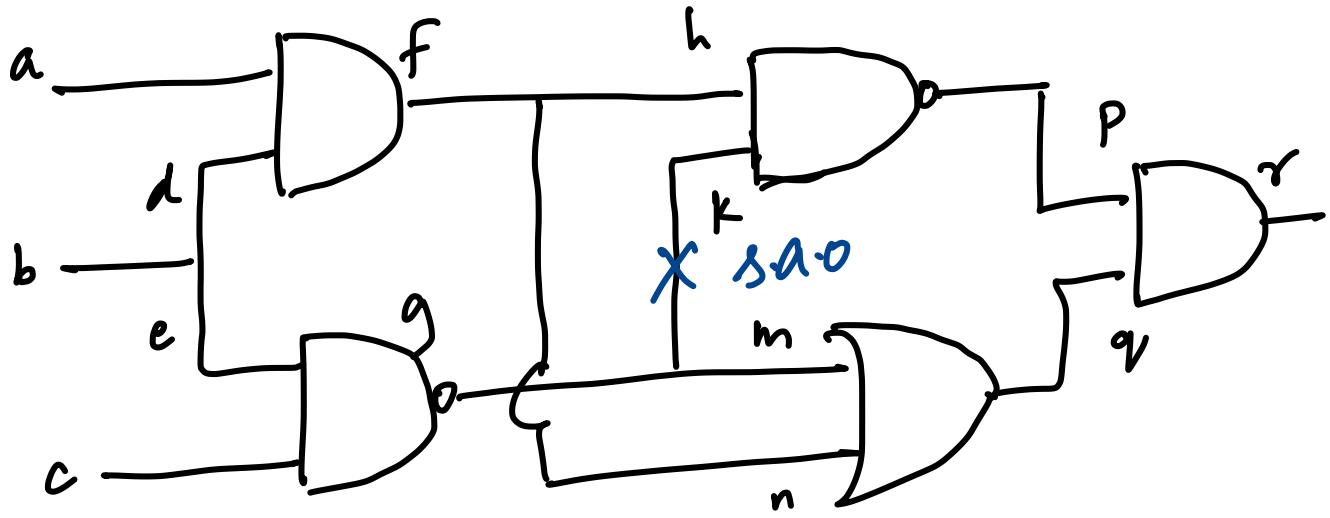
$$C(N) := \text{dist}(N, PI)$$

$$O(N) := \text{dist}(N, PO)$$

let us denote these metrics by the vector $\begin{pmatrix} C(N) \\ O(N) \end{pmatrix}$ at each node

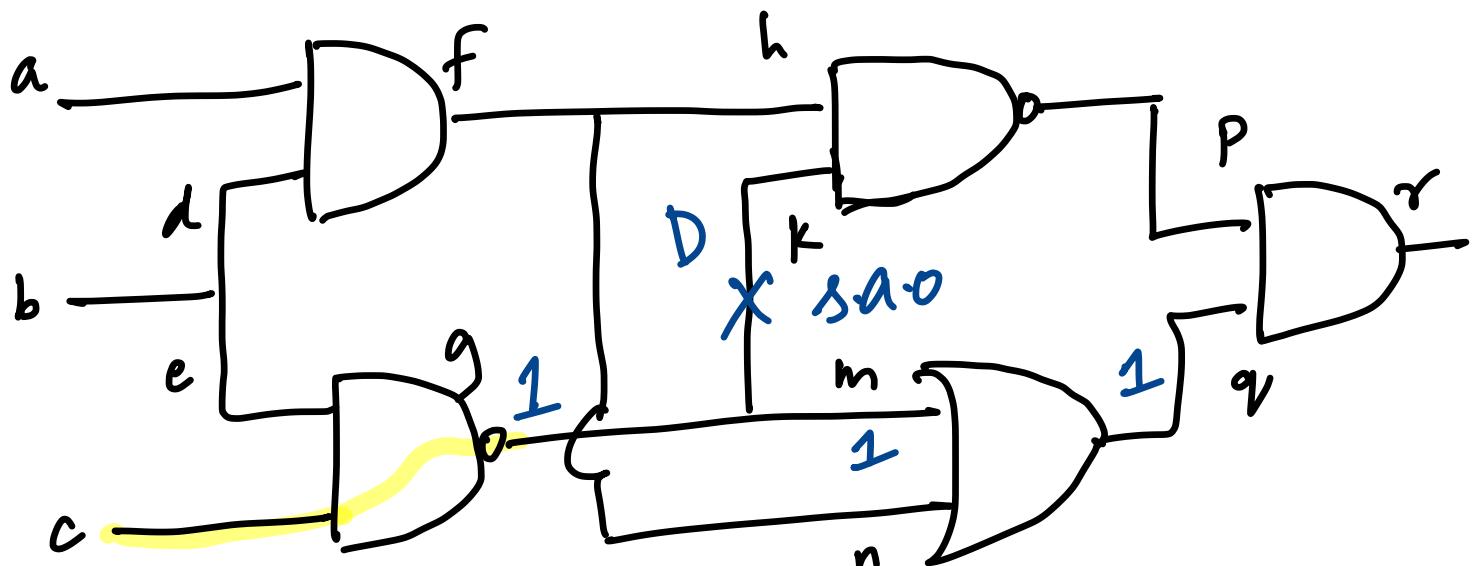
The provided circuit for the question





using PODEM Algorithm

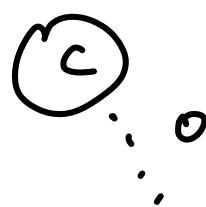
- ① K is S.A.O \Rightarrow g must be 1 to activate fault at $K \Rightarrow m=1, q=1$
using IMPLY operation



By backtrace heuristic NAND with output 1, for this any input must be one So we back trace to most controllable i.e b or c one inversion on the path so

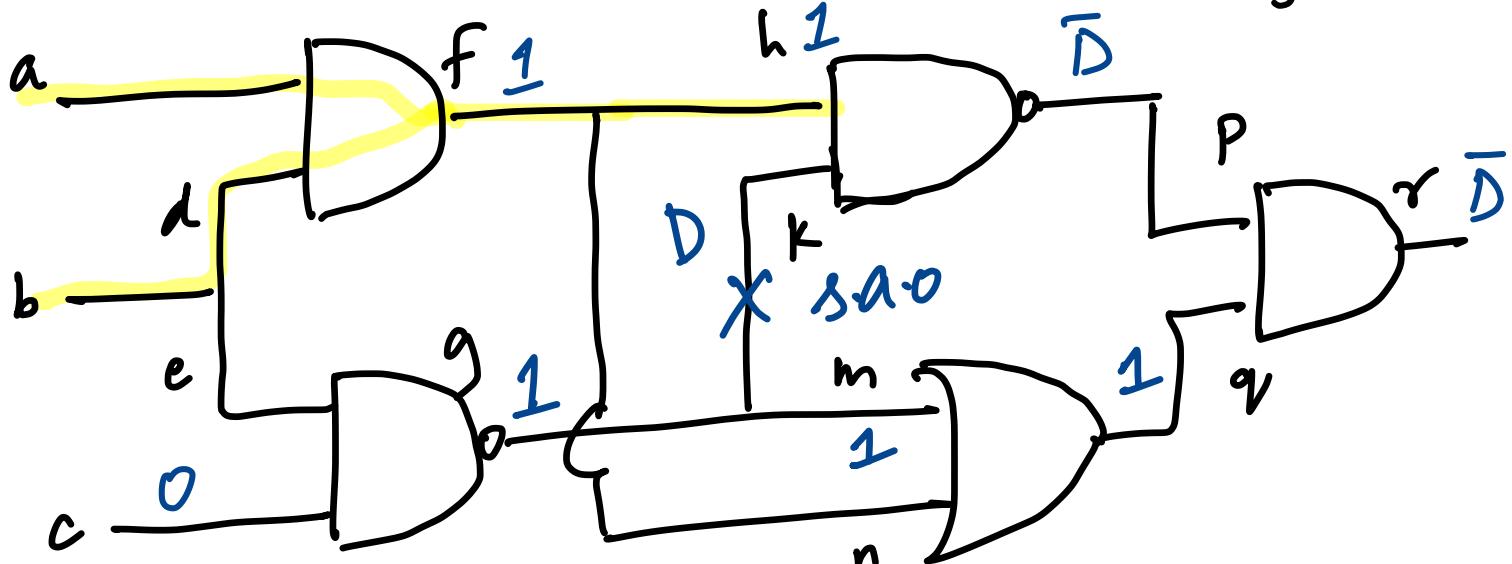
so we pick $c=0$ $C(b)=C(c)=0$

then



$\text{so } l, b$
are equally
likely
pick one
randomly

so the fault is activated, to propagate it now h must be 1, new objective

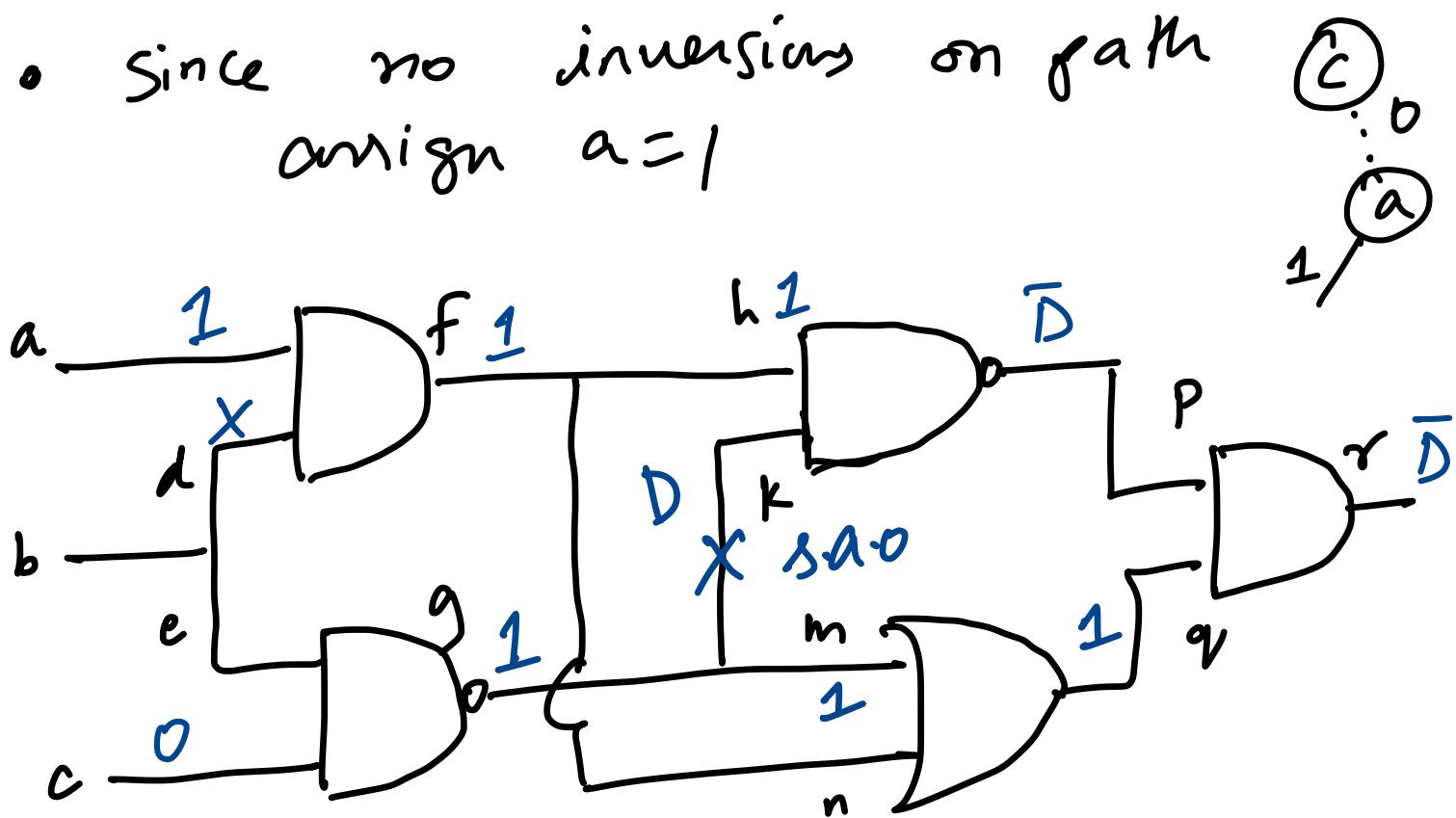


$h=1$ IMPLIES $f=1$ and $n=1$.
now need both a and d to be 1

- we need both to be 1, so we pick the least controllable one (hander)

since $c(a) = c(b) = 0$, we can pick either, choose 'a'

- Since no inversions on path assign $a=1$



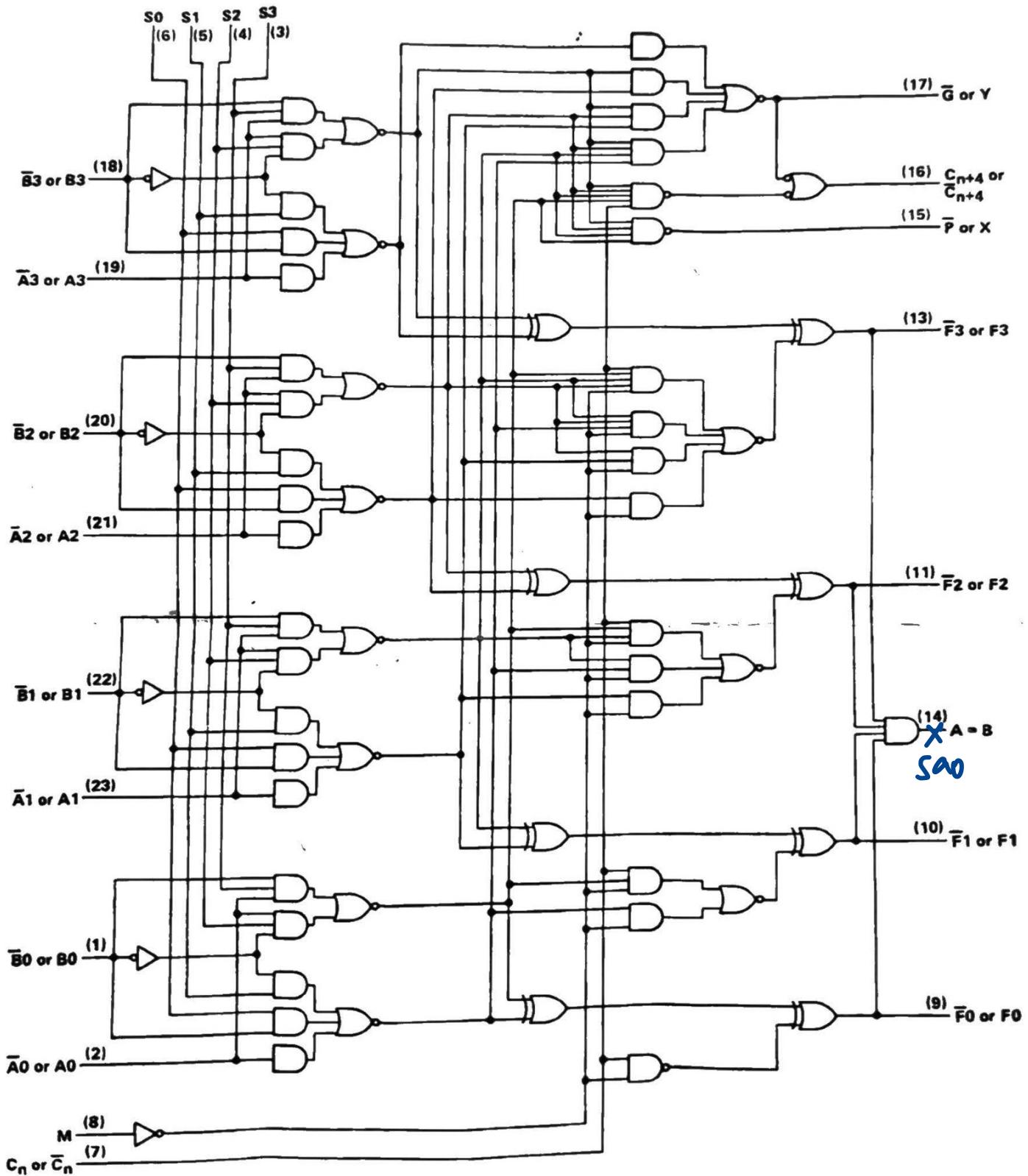
- we now choose the next least controllable i.e $d \Rightarrow b=1$

\therefore we have now successfully activated and propagated via PODEM algorithm.

Thus test vector is $\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$

Q3) for line 14 S-A-0

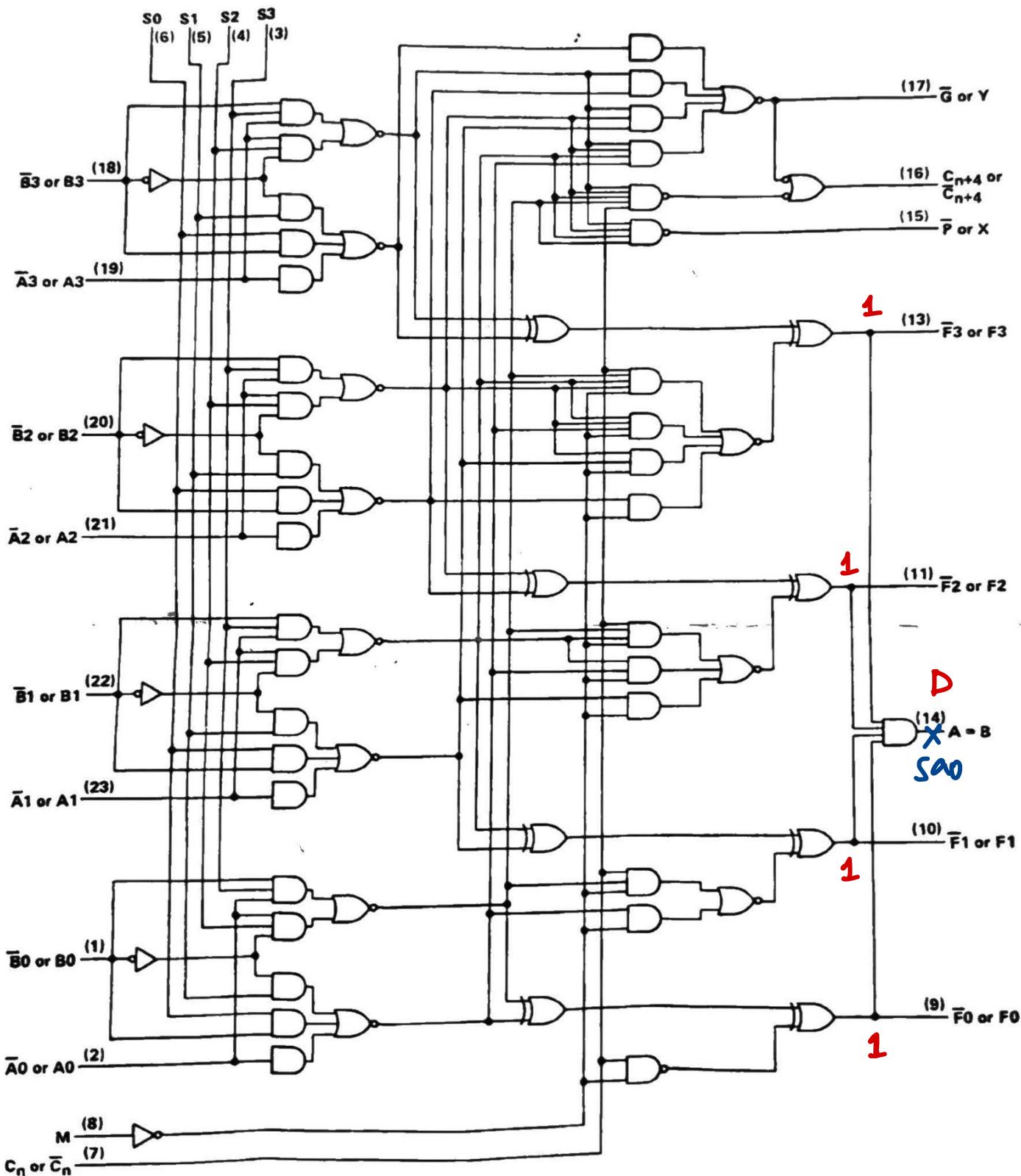
logic diagram (positive logic)



Pin numbers shown are for DW, J, N, and W packages.

we do not need propagate fault since it is already at Primary Output

logic diagram (positive logic)

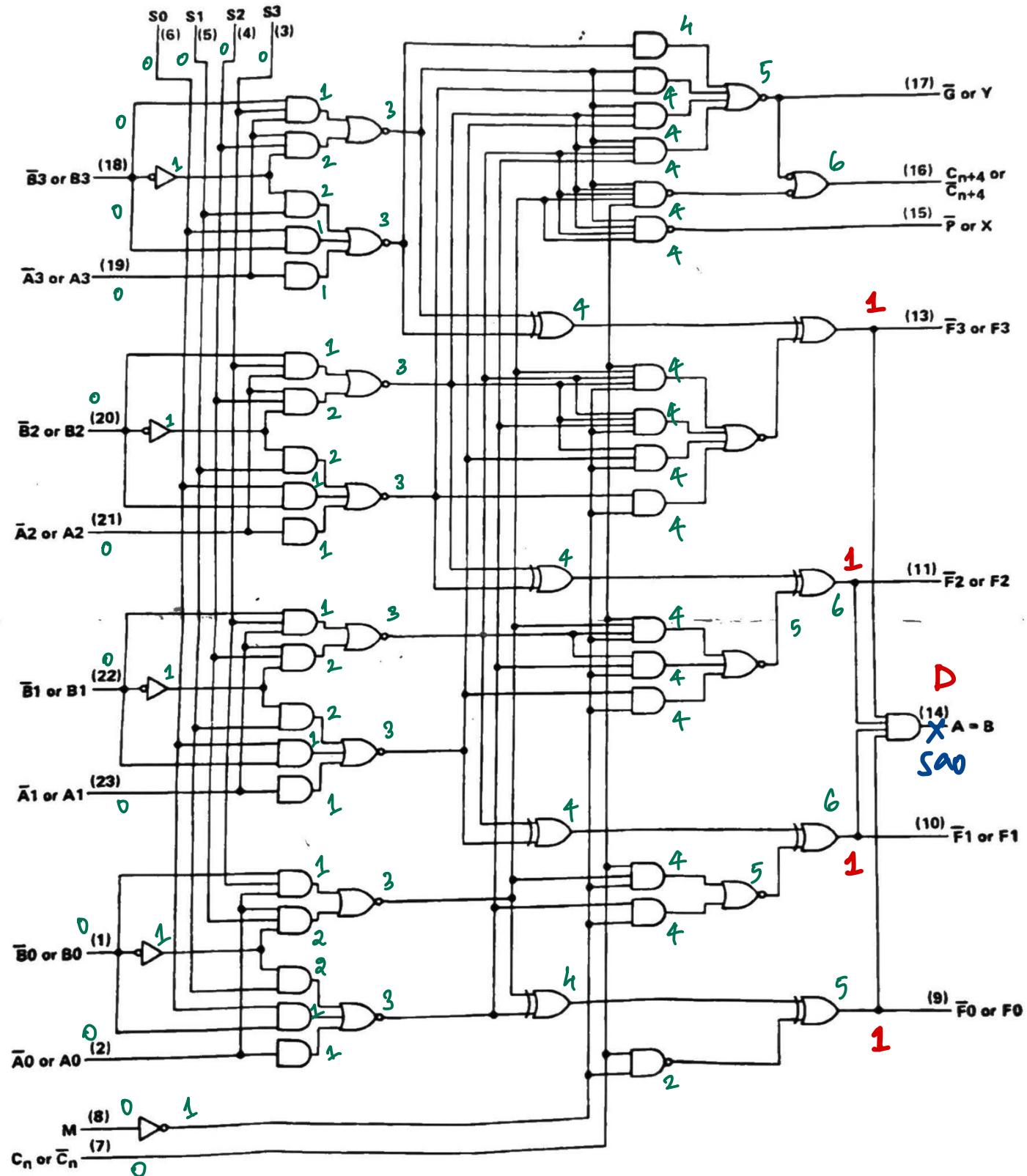


Pin numbers shown are for DW, J, N, and W packages.

we get the above inferences to activate faults at (14) after IMPLY

we evaluate the controllability metric
for all nodes in the path
denoted in green color

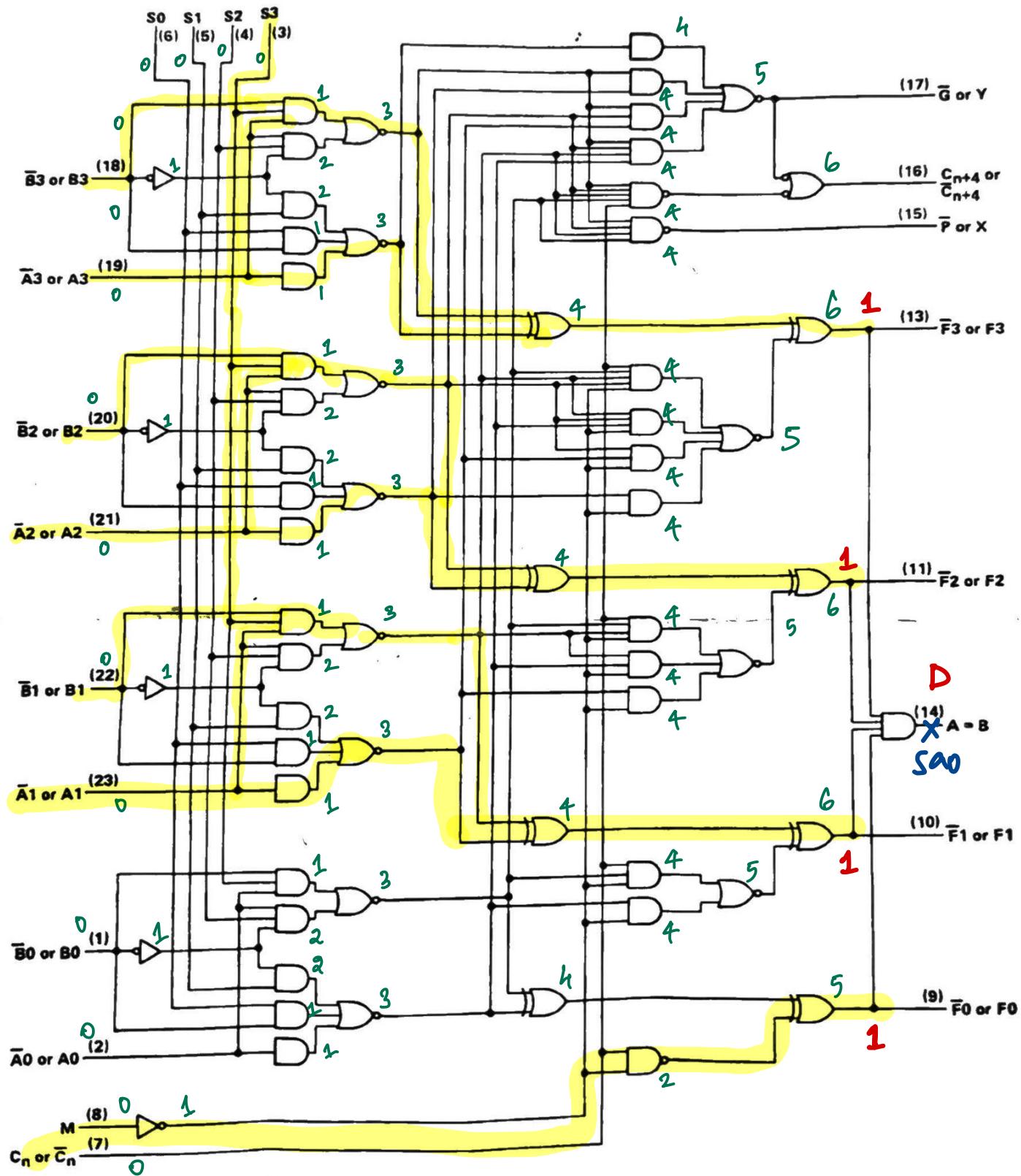
logic diagram (positive logic)



Pin numbers shown are for DW, J, N, and W packages.

a heuristic choice is the following highlighted paths

logic diagram (positive logic)



Pin numbers shown are for DW, J, N, and W packages.

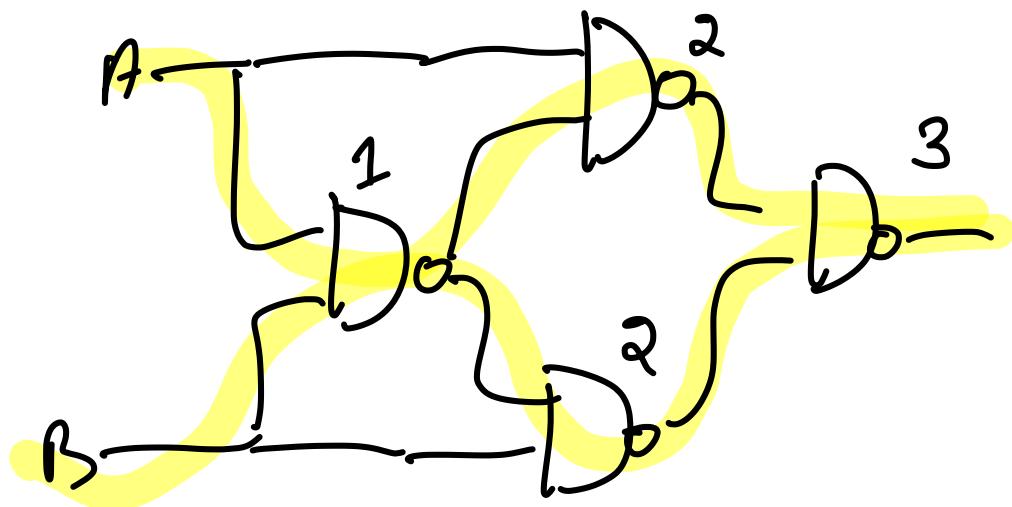
Thus good initial heuristic choices
for primary inputs are

M , C_n , A_1, B_1 , A_2, B_2 , A_3, B_3 and S_3

now we perform backtrace
and choose values for
these variables and make
decision tree.

we make use of the initial
choice heuristic of PODEM

NOTE: we take XOR gates
to be made up of 3 NAND
gates and has 3 inversions



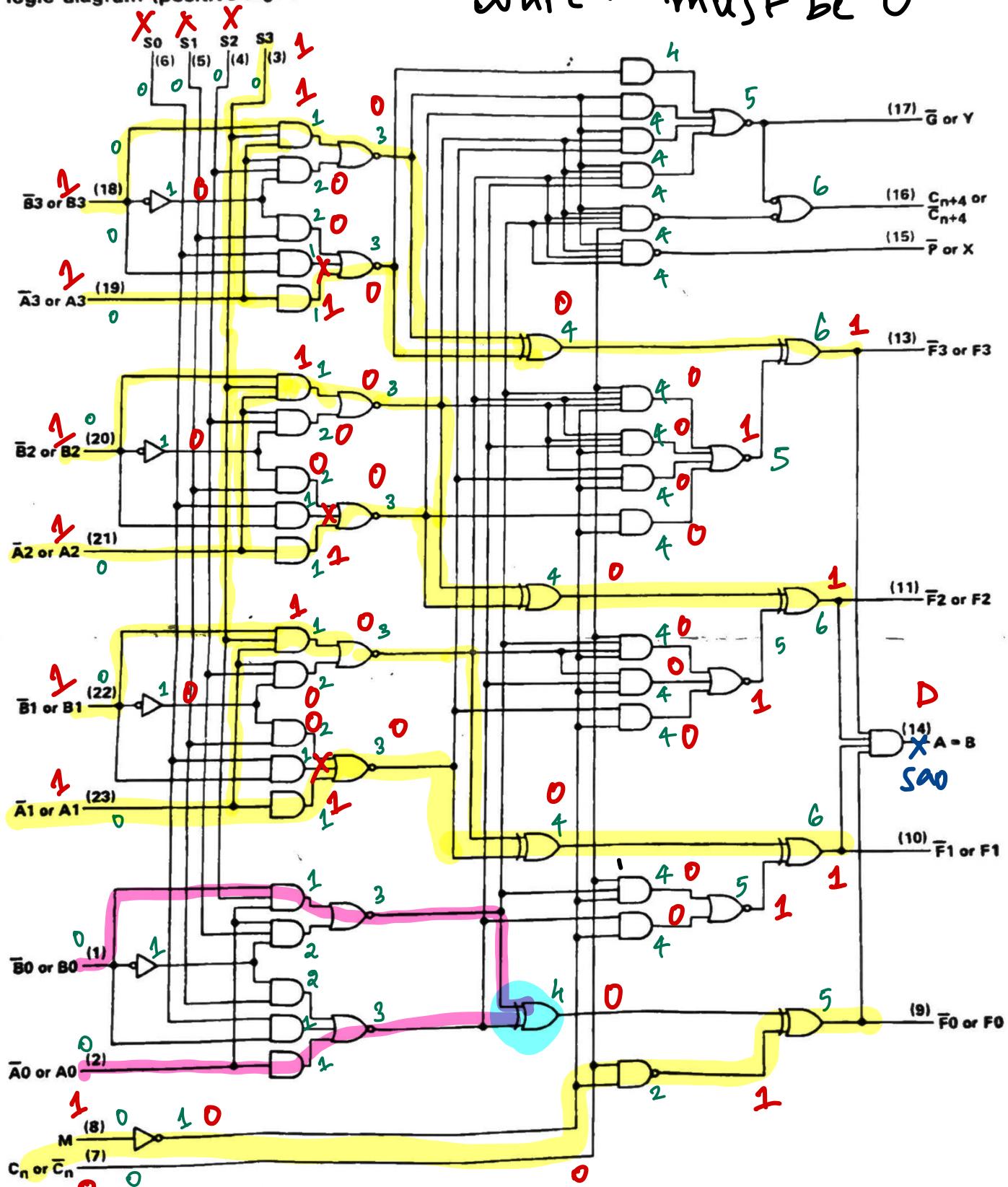
- if on the path we have even number of inversions then we assign opposite sign of objective
- otherwise we assign same value as objective.

Variable	no of inversions on path	objective assignment
M	not nand xor $1+1+3 = 5$	1 1
Cn	nand xor $1+3 = 4$	1 0
S3	xor xor xor $1+3+3 = 7$	1 1
A1	$1+3+3 = 7$	1 1
B1	$1+3+3 = 7$	1 1
A2	$1+3+3 = 7$	1 1
B2	$1+3+3 = 7$	1 1
A3	$1+3+3 = 7$	1 1
B3	$1+3+3 = 7$	1 1

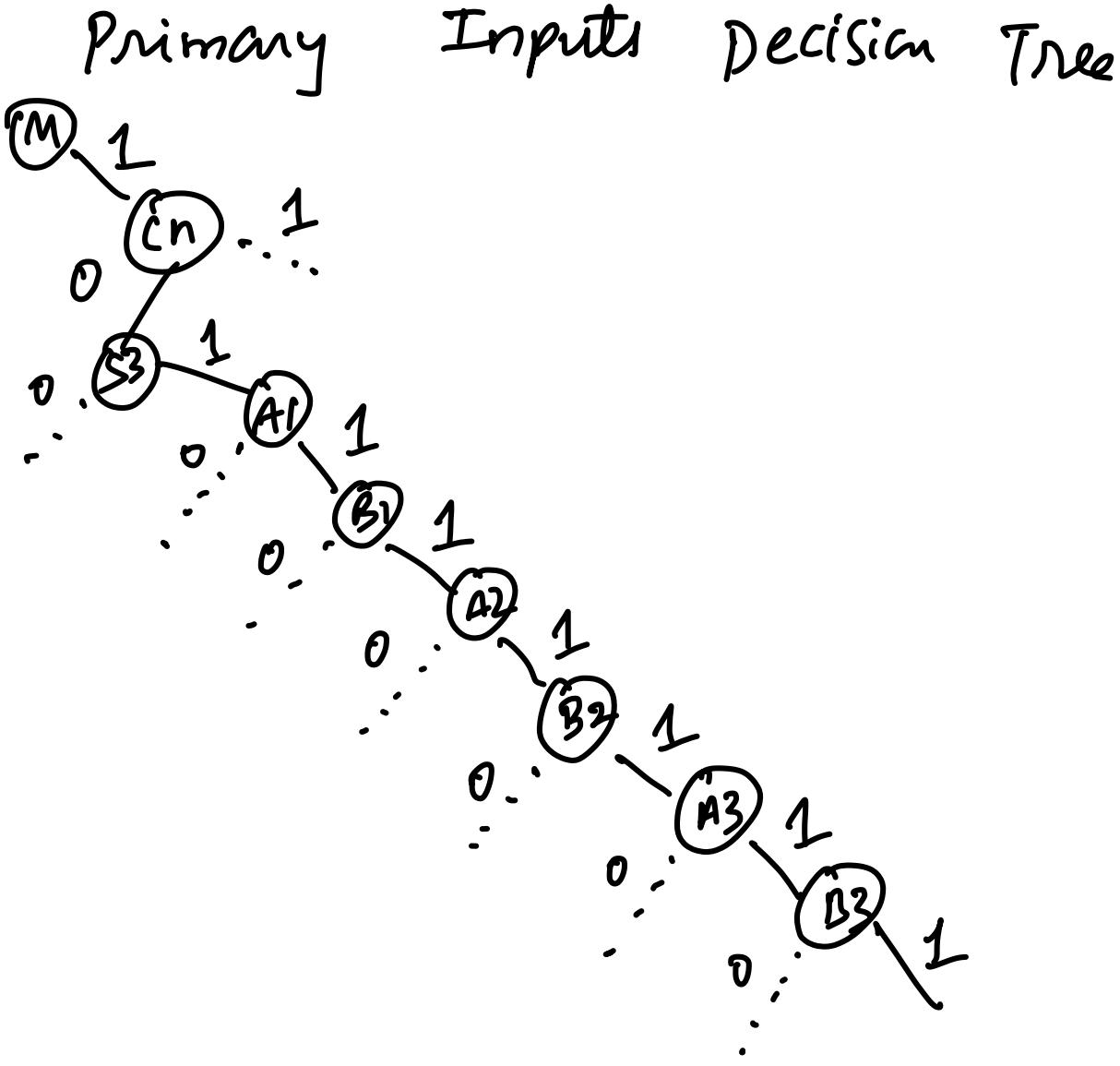
Thus we see that for this heuristic
we need more information about
the blue XOR gate output

which must be 0

logic diagram (positive logic)



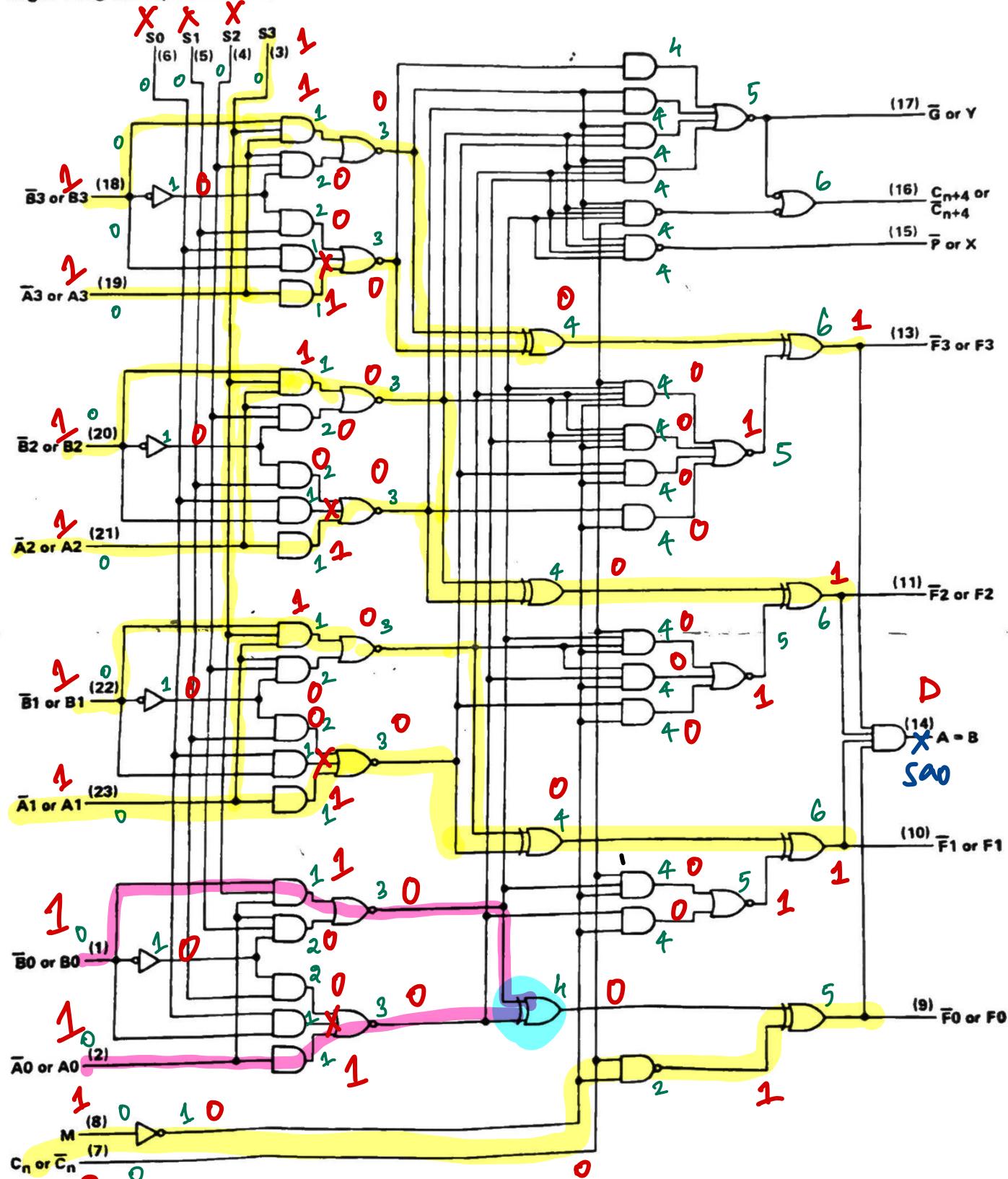
Pin numbers shown are for DW, J, N, and W packages.



So to get more info about the blue coloured XOR gate we use the same heuristic as before and see the most controllable input to it which is the pink colour path

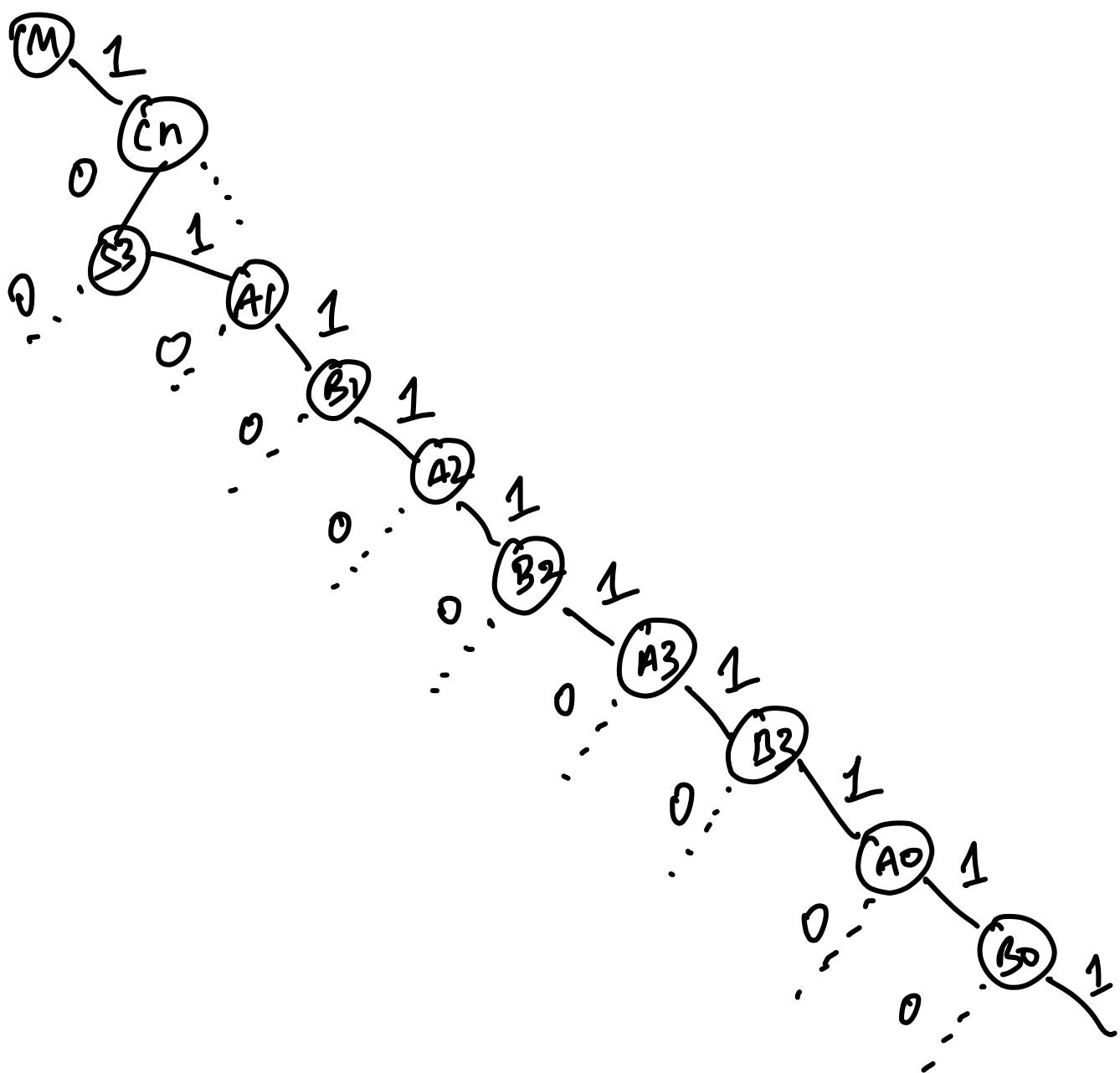
also the no of inverting gates on the pink paths are 4 which is odd
 So the heuristic choice for $B_0, A_0 = 1$

logic diagram (positive logic)



Pin numbers shown are for DW, J, N, and W packages.

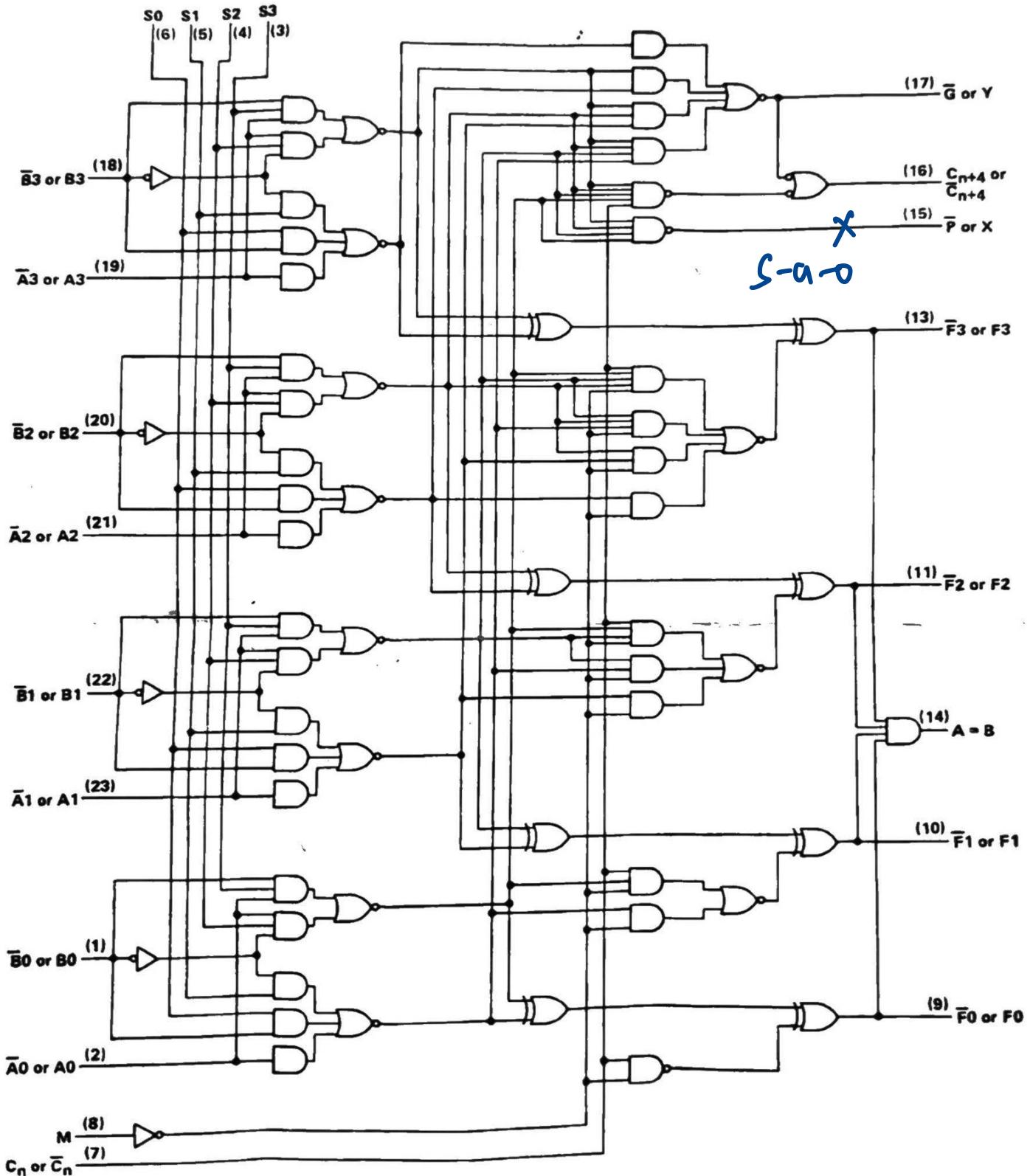
Primary Inputs Decision Tree



Thus for this choice we see the S-a-O fault at (14) is triggered so the testvector is

Q3(b) for line 15 S-a-0

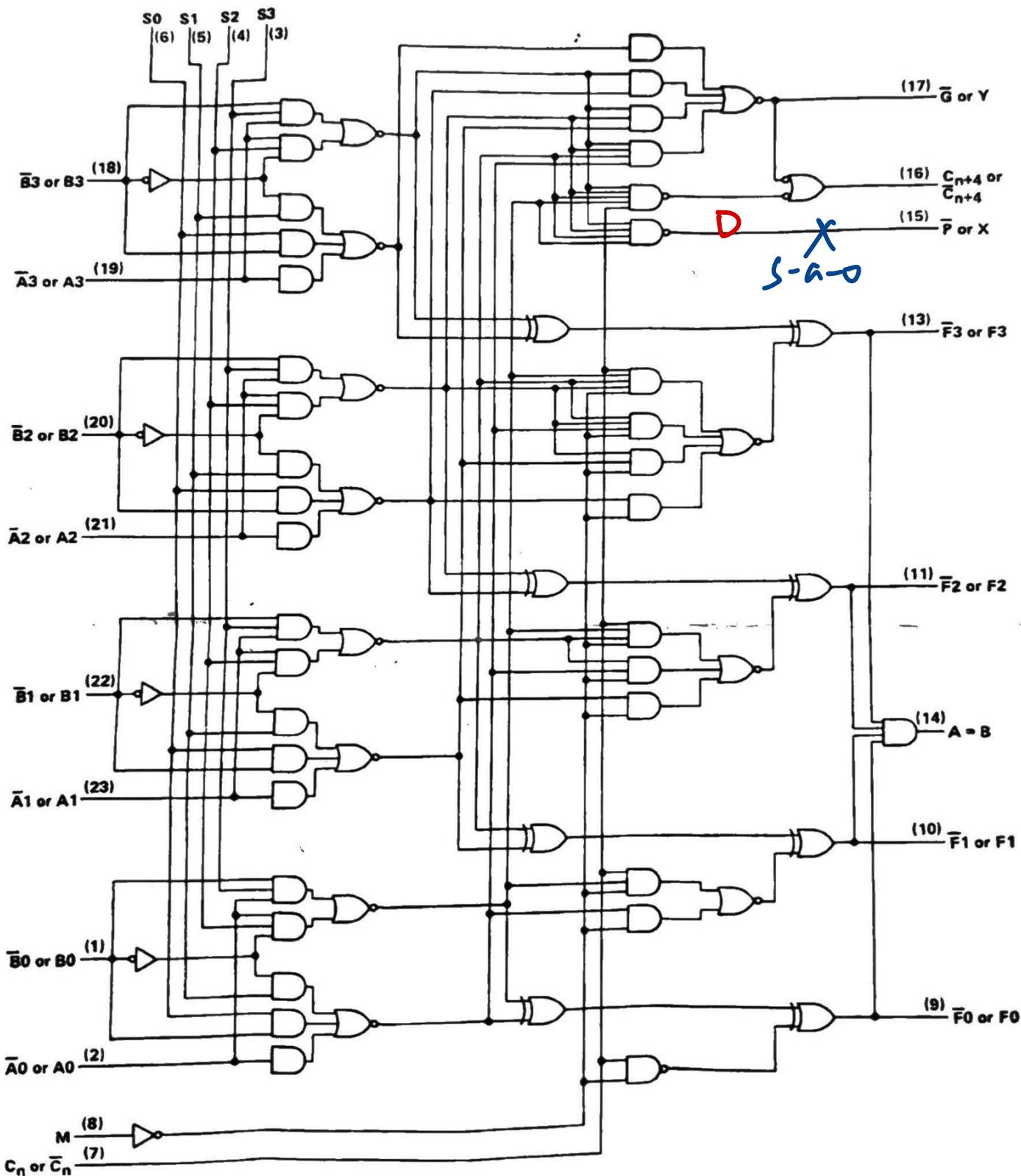
logic diagram (positive logic)



Pin numbers shown are for DW, J, N, and W packages.

we do not need propagate fault since it is already at Primary Output

logic diagram (positive logic)

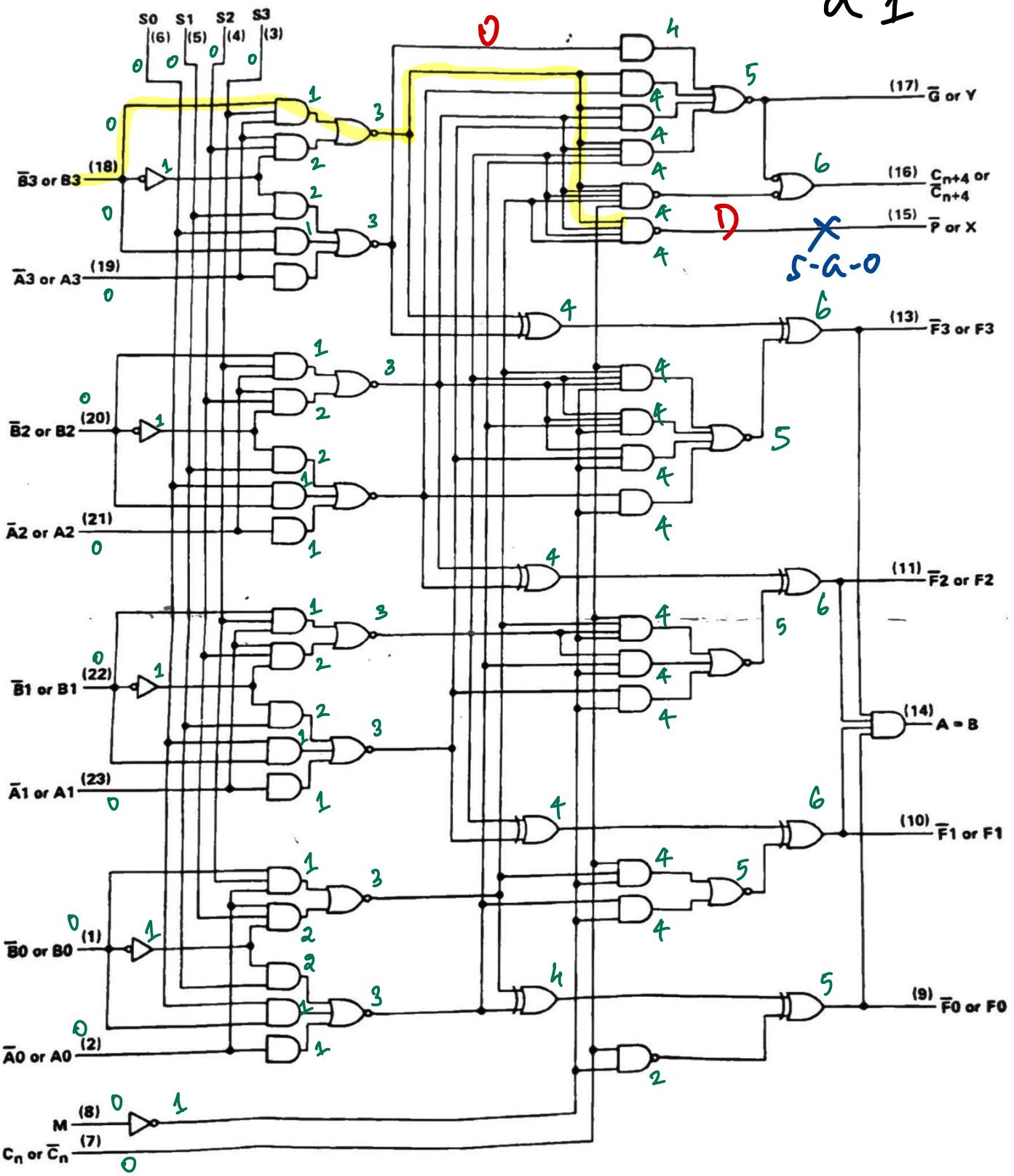


Pin numbers shown are for DW, J, N, and W packages.

we get the above inferences to activate faults at (14).

a heuristic choice is the following highlighted paths, since one input being 0 for a NAND gives a 1

logic diagram (positive logic)



Pin numbers shown are for DW, J, N, and W packages.

we pick the most controllable input

Variable no of inversions on path objective assignment

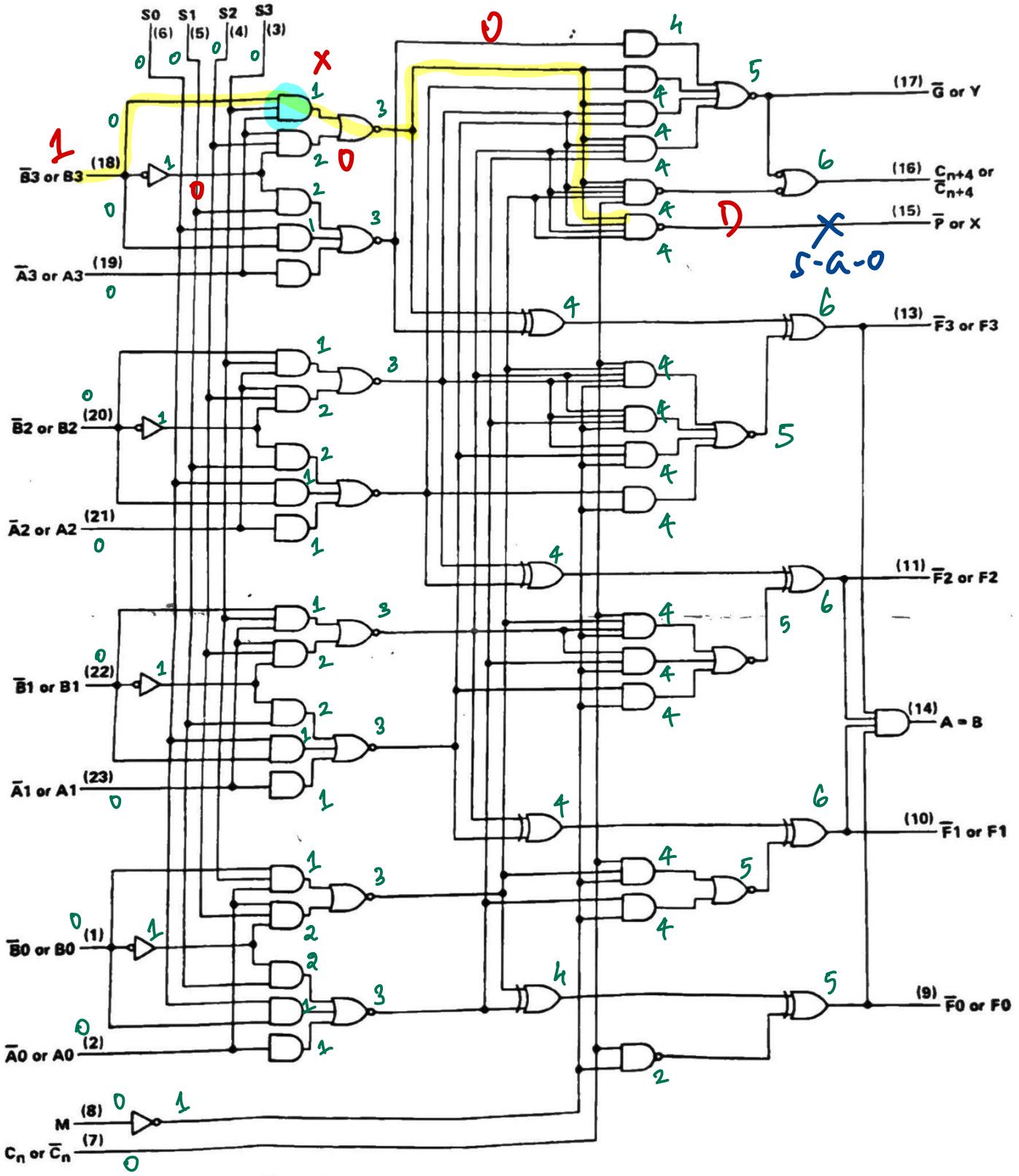
B3

1

1

1

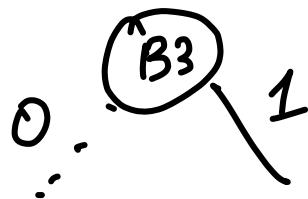
logic diagram (positive logic)



Pin numbers shown are for DW, J, N, and W packages.

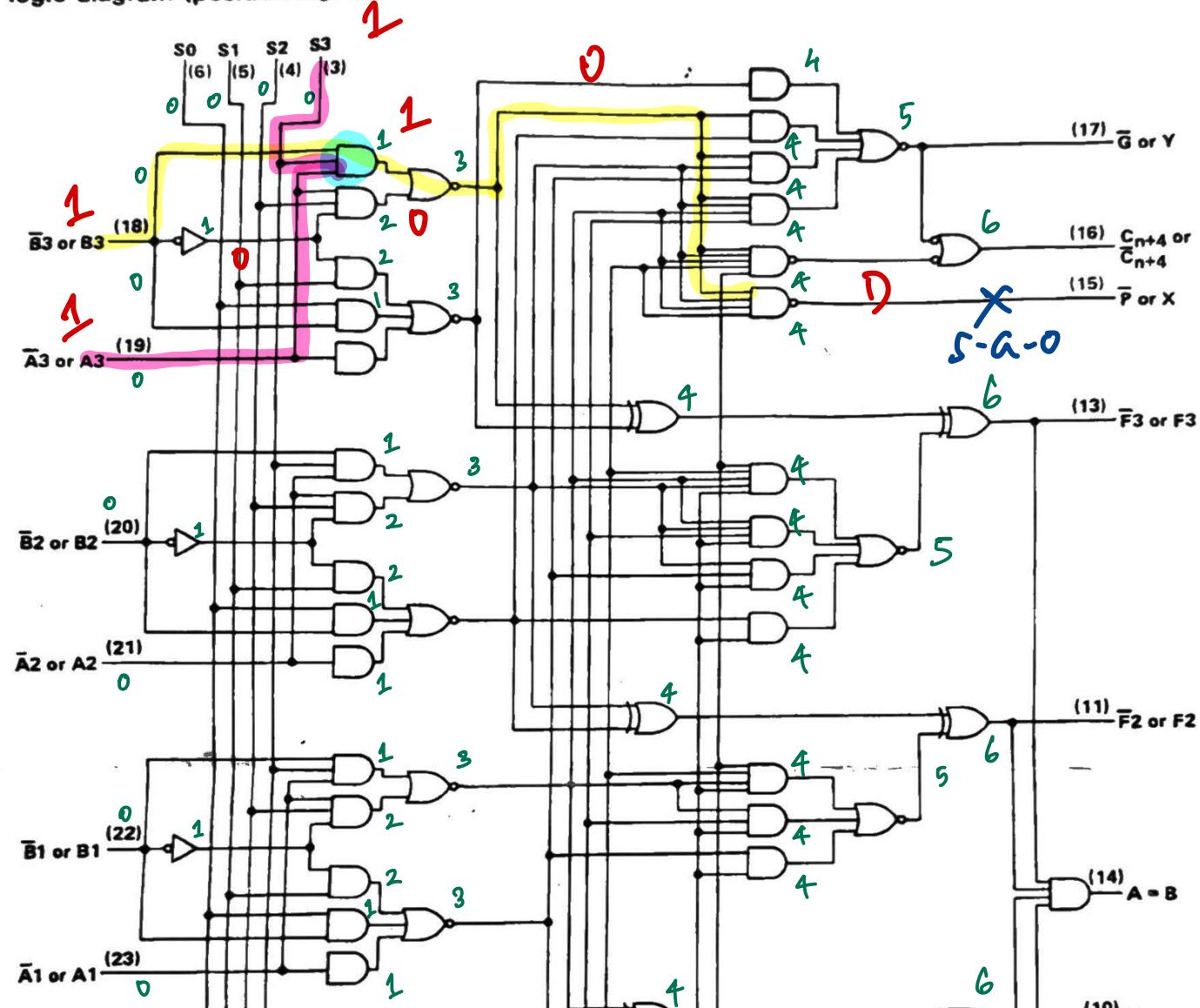
Thus more information is needed at the B3E AND gate

Decision Tree



heuristic choice for this gate
are following paths also the
objective value at the gate is 1

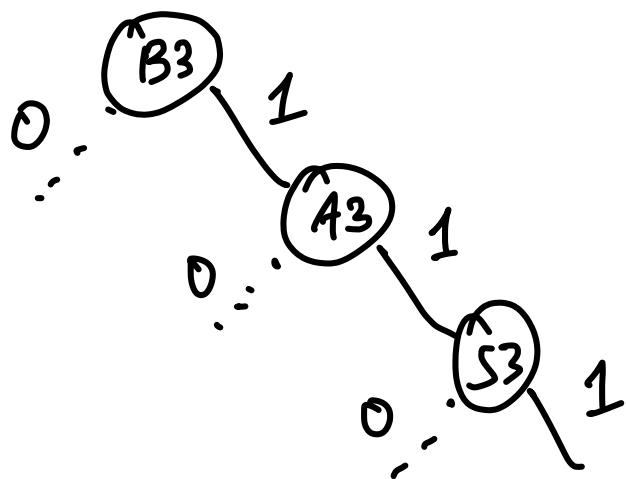
logic diagram (positive logic)



$\longleftrightarrow S3=1$ and $A3=1$.

thus the final decision tree
for this is

Decision Tree



Thus a testvector to test
this particular fault in
order to test (15) s-a-o is
given by

$$\begin{bmatrix} Cn & M & S3 & A0 & B0 & A1 & B1 & A2 & B2 & A3 & B3 & S0 & S1 & S2 \\ X & X & 1 & X & X & X & X & X & X & 1 & 1 & X & X & X \end{bmatrix}$$

Hence we are done.