```
FAN()
begin
   if Imply_and_check() = FAILURE then return FAILURE
   if (error at PO and all bound lines are justified) then
     begin
        justify all unjustified head lines
        return SUCCESS
     end
   if (error not at PO and D-frontier = ∅) then return FAILURE
   /* initialize objectives */
   add every unjustified bound line to Current_objectives
   select one gate (G) from the D-frontier
   c = controlling value of G
   for every input (j) of G with value x
      add (j,c̄) to Current_objectives
   /* multiple backtrace */
   (i,v_i) = Mbacktrace(Current_objectives)
   Assign(i,v_i)
   if FAN() = SUCCESS then return SUCCESS
   Assign(i,v̄_i)  /* reverse decision */
   if FAN() = SUCCESS then return SUCCESS
   Assign(i,x)
   return FAILURE
end
```

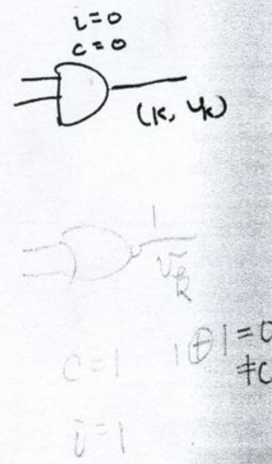
Figure 6.37 FAN

another generalization of the head-line concept, based on an analysis that is both topological and functional. In FAST, a line l is a backtrace-stop for value v, if the assignment $l = v$ can be justified without conflicts. For example, in Figure 6.39, L is a backtrace-stop for 0, because $L = 0$ can be justified without assigning any line with reconvergent fanout (by $F = 0$ and $A = B = 1$). Note that L is fed by reconvergent fanout, and it is not a total-reconvergence line. The identification of backtrace-stop lines will be explained in the next section.

6.2.1.3 Selection Criteria

The search process of any of the TG algorithms analyzed in this chapter involves decisions. A first type of decision is to select one of the several unsolved problems existing at a certain stage in the execution of the algorithm. A second type is to select one possible way to solve the selected problem. In this section we discuss *selection criteria* that are helpful in speeding up the search process. These selection criteria are based on the following principles:

```
Mbacktrace (Current_objectives)
begin
    repeat
        begin
            remove one entry (k,v_k) from Current_objectives
            if k is a head line
                then add (k,v_k) to Head_objectives
            else if k is a fanout branch then
                    begin
                        j = stem(k)
                        increment number of requests at j for v_k
                        add j to Stem_objectives
                    end
                else /* continue tracing */
                    begin
                        i = inversion of k
                        c = controlling value of k
                        if (v_k⊕i = c) then
                            begin
                                select an input (j) of k with value x
                                add (j,c) to Current_objectives
                            end
                        else
                            for every input (j) of k with value x
                                add (j,c̄) to Current_objectives
                    end
        end
    until Current_objectives = ∅
    if Stem_objectives ≠ ∅ then
      begin
          remove the highest-level stem (k) from Stem_objectives
          v_k = most requested value of k
          if (k has contradictory requirements and
              k is not reachable from target fault)
              then return (k,v_k)
          add (k,v_k) to Current_objectives
          return Mbacktrace (Current_objectives)
      end
    remove one objective (k,v_k) from Head_objectives
    return (k,v_k)
end
```



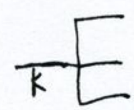


Figure 6.34 Multiple backtrace

2. Consider the circuit in Fig. 2. **(a)** Use COP to generate the controllability-1, controllability-0, and observability of each line. **(b)** Use PODEM to generate a test pattern for line I stuck-at *1* under the guidance of con/obser values in each line. Note that you must follow the PODEM example shown in Problem 1 to present your ATPG process. **(c)** Repeat (b) using FAN. Note that you must follow the same steps in Problem 1 to show the ATPG process. **(d)** If your answer in (c) does not create a decision tree, try to generate a test pattern for line I stuck-at *1 without* the guidance of con/obser such that a decision tree will be generated. (this enables you to choose a way of multiple backtracing with decision tree created). **(e)** Try to find a case in which the ATPG process involves decision tree backtracking by FAN? Note that, if yes, you have to show the fault and the entire ATPG process. (60%)

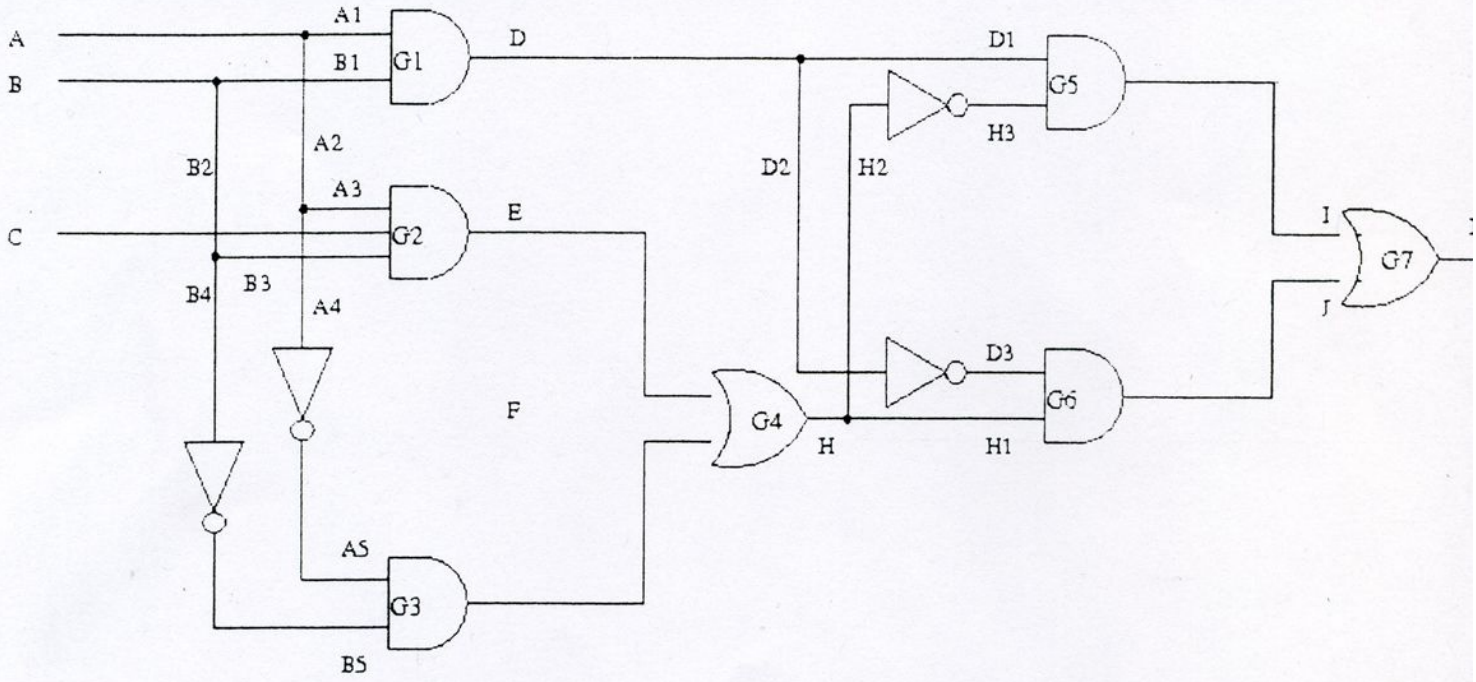


Fig: 2