1) **Testability Measures:** Compute combinational SCOAP measures for each labeled node in Fig. 7.39, p.207. Give controllability and observability numbers for each node.

A: $\text{CC0}(A) = 1$
$\text{CC1}(A) = 1$
$\text{CO}(A) = \text{CO}(d) + \text{CC1}(B) + 1 = 4 + 1 + 1 = 6$

B: $\text{CC0}(B) = 1$
$\text{CC1}(B) = 1$
$\text{CO}(B) = \min(\text{CO}(B_{d\text{branch}}), \text{CO}(B_{e\text{branch}}), \text{CO}(B_{Z\text{branch}})) = 5$
$\text{CO}(B_{d\text{branch}}) = \text{CO}(d) + \text{CC1}(A) + 1 = 4 + 1 + 1 = 6$
$\text{CO}(B_{e\text{branch}}) = \text{CO}(e) + \text{CC1}(C) + 1 = 3 + 1 + 1 = 5$
$\text{CO}(B_{Z\text{branch}}) = \text{CO}(Z) + \min(\text{CC0}(k), \text{CC1}(k)) + 1 = 0 + \min(4, 7) + 1 = 5$

C: $\text{CC0}(C) = 1$
$\text{CC1}(C) = 1$
$\text{CO}(C) = \min(\text{CO}(C_{e\text{branch}}), \text{CO}(C_{f\text{branch}})) = \min(5, 8) = 5$
$\text{CO}(C_{e\text{branch}}) = \text{CO}(e) + \text{CC1}(B) + 1 = 3 + 1 + 1 = 5$
$\text{CO}(C_{f\text{branch}}) = \text{CO}(f) + \min(\text{CC0}(B), \text{CC1}(B)) + 1 = 6 + \min(1, 1) + 1 = 8$

d: $\text{CC0}(d) = \min(\text{CC0}(A), \text{CC0}(B)) + 1 = 2$
$\text{CC1}(d) = \text{CC1}(A) + \text{CC1}(B) + 1 = 3$
$\text{CO}(d) = \text{CO}(g) + \text{CC0}(e) + 1 = 0 + 3 + 1 = 4$

e: $\text{CC0}(e) = \text{CC1}(B) + \text{CC1}(C) + 1 = 3$
$\text{CC1}(e) = \min(\text{CC0}(B), \text{CC0}(C)) + 1 = 2$
$\text{CO}(e) = \text{CO}(g) + \text{CC0}(d) + 1 = 0 + 2 + 1 = 3$

f: $\text{CC0}(f) = \min(\text{CC0}(B) + \text{CC0}(C), \text{CC1}(B) + \text{CC1}(C)) + 1 = 3$
$\text{CC1}(f) = \min(\text{CC0}(B) + \text{CC1}(C), \text{CC1}(B) + \text{CC0}(C)) + 1 = 3$
$\text{CO}(f) = \text{CO}(k) + \text{CC1}(h) + 1 = 2 + 3 + 1 = 6$
g: CC0(g) = CC0(d) + CC0(e) + 1 = 2 + 3 + 1 = 6
CC1(g) = min(CC1(d), CC1(e)) + 1 = min(3, 2) + 1 = 3
CO(g) = min(CO(Y), CO(h)) = min(0, 7) = 0

h: CC0(h) = CC0(g) = 6
CC1(h) = CC1(g) = 3
CO(h) = CO(k) + CC1(f) + 1 = 2 + 3 + 1 = 6

k: CC0(k) = min(CC0(h), CC0(f)) + 1 = min(6, 3) + 1 = 4
CC1(k) = CC1(h) + CC1(f) + 1 = 3 + 3 + 1 = 7
CO(k) = CO(Z) + min(CC0(B), CC1(B)) + 1 = 0 + min(1, 1) + 1 = 2

Y: CC0(Y) = CC0(g) = 6
CC1(Y) = CC1(g) = 3
CO(Y) = 0

Z: CC0(Z) = min(CC0(k) + CC0(B), CC1(k) + CC1(B)) + 1 = min(5, 8) + 1 = 6
CC1(Z) = min(CC0(k) + CC1(B), CC1(k) + CC0(B)) + 1 = min(5, 8) + 1 = 6
CO(Z) = 0

2) **Dominator**: Problem 7.7b
For each gate in Figure 7.41, identify its absolute dominators.

A: Gate Z is an absolute dominator.
B: Gate Z is an absolute dominator.
C: Gates m and Z are absolute dominators.
D: Gates m and Z are absolute dominators.
E: Gates l, s, w and Z are absolute dominators.
F: Gates l, s, w and Z are absolute dominators.
g: Gates m and Z are absolute dominators.
h: Gates m and Z are absolute dominators.
k: Gates m and Z are absolute dominators.
l: Gates s, w and Z are absolute dominators.
m: Gate Z is an absolute dominator.
p: Gates r and Z are absolute dominators.
q: Gates w and Z are absolute dominators.
s: Gates w and Z are absolute dominators.
r: Gate Z is an absolute dominator.
u: Gate Z is an absolute dominator.
w: Gate Z is an absolute dominator.
z: none.

3) **PODEM**: Problem 7.8. Do not use any internal node assignments or mandatory value assignments. (Make an arbitrary decision when a choice is available during back-trace). Generate a test with the PODEM ATPG algorithm for the fault g s-a-1 in Figure 7.41.

See problem 2) for circuit diagram.
Select path: g → h → m → u → Z
Initial objective: g → 0 (excite s-a-1 fault)
C → 0 (path through h is blocked)
D → 0 (paths through h, k are blocked, D-frontier empty)
D → 1 (fault not excited)
C → 1 (fault not excited)
PODEM stops (fault untestable)

4) **Logic Implications**: Use mandatory node assignments to solve Problem 7.19.
Find tests for: 1) The single fault d: s-a-0 and 2) The single fault m: s-a-0.

**Single fault d: s-a-0**
d → D (s-a-0: 1/0)
a → 1 (excite s-a-0 fault), c → 1
e → 0 (propagate fault), b → 0, f → 0
h → 1 (f implies h = 1)
g → D (e = 0 implies fault D)
m → D, k → D
n → D’ (c = 1 implies D’)
p → D (h = 1 implies D)
q → 1 (p, n imply q = 1)
Fault is untestable, and thus redundant.

**Single fault m: s-a-0**
m → D (s-a-0: 1/0)
g → 1 (excite s-a-0 fault), k → 1
h → 1 (propagate fault)
f → 0 (h = 1 implies f = 0), b → 0, e → 0
a → 1 (g, b imply a = 1), c → 1, d → 1
n → 0 (c, k imply n = 0)
p → D (h = 1 implies D)
If redundant fault $d$: $s-a-0$ is removed (by grounding line $d$):

$m \rightarrow D$ ($s-a-0$: 1/0)
$g \rightarrow 1$ (excite $s-a-0$ fault), $k \rightarrow 1$
$h \rightarrow 1$ (propagate fault)
$f \rightarrow 0$ ($h = 1$ implies $f = 0$)

However, to obtain the result $g = 1$, $b$ and $e$ must be set to 1, not 0, since the $d$ input of the OR gate has been grounded. Thus, this fault ($m s-a-0$) becomes untestable and redundant. The Boolean function for this circuit is a simple 2-input XOR function.

By finding and removing redundant faults, the minimal implementation for a circuit can be obtained.

5) **PODEM:** Use PODEM to generate a test vector for fault $A/0$ and another vector for $L14/0$ in the attached TTL 4-bit ALU. For controllability and observability use approximate distance from PI and PO respectively. You may use mandatory assignments to reduce search. Show the decision tree in each case.

NOTE: test vectors and decision trees may vary depending on the path of backtrace.

**A:** fault $D$ ($s-a-0$)
Test vector:
$S0 = 0$, $S1 = 0$, $A3 = 0$, $M = 1$
$D$ appears on output $F3$

Initial objective: node $a = 1$ (excite $s-a-0$), backtrace from node $a$
$A3 = 0$ (node $b \rightarrow 1$ through forward implication)
$S0 = 0$
$S1 = 0$ (objective for $a$ achieved)
objective: node $d = 1$ (through forward implication, node $c = D'$, so we want node $e$ to propagate the fault $D$), backtrace from node $d$
$M = 1$ (all four AND gates are set to 0)
Test achieved

**14:** fault $D$ ($s-a-0$)
$S0 = 0$, $S1 = 0$, $A0 = 0$, $A1 = 0$, $A2 = 0$, $A3 = 0$, $M = 1$
$D$ appears on output 14

Initial objective: node $d = 1$, backtrace from node $d$
objective: node $h = 1$, backtrace from node $h$
$M = 1$ (through forward implication, nodes $e$, $f$, $g$, $h \rightarrow 1$)
objective: node $k = 0$, backtrace from node $k$
objective: node $v = 1$, backtrace from node $v$
$A0 = 0$ (through forward implication, node $u \rightarrow 1$)
$S0 = 0$
S1 = 0 (objectives for v, k, h, d achieved)
objective: node c = 1, backtrace from c
objective: node l = 0, backtrace from l
objective: node t = 1, backtrace from t
A1 = 0 (through forward implication, node s → 1, objectives for t, l, c achieved)
objective: node b = 1, backtrace from b
objective: node m = 0, backtrace from m
objective: node r2 = 1, backtrace from r2
A2 = 0 (through forward implication, node r → 1, objectives for r2, m, b achieved)
objective: node a = 1, backtrace from a
objective: node n = 0, backtrace from n
objective: node q = 1, backtrace from q
A3 = 0 (through forward implication, node p → 1, objectives for q, n, a achieved)
Test achieved