Problem 1 (20 points) Sequential circuit testing: 5 vs. 9-value logic
Text, page 250, Problem 8.5.

Problem 2 (20 points) Functional testing
(a) Let C be a combinational circuit with \( n \) primary inputs (PIs) and \( m \) primary outputs (POs) \( z_1, z_2, ..., z_m \). Each PO \( z_i \) depends on (is a function of) \( n_i \) PIs, where \( n_i \leq n \). C is said to be **exhaustively** tested if we apply all \( 2^n \) input combinations to its \( n \) PIs. C is said to be **pseudoexhaustively** tested if we apply all \( 2^{n_i} \) input vectors to all PIs feeding every PO \( z_i \). Since we can have \( n_i < n \), and the PI sets of two different POs can overlap, it’s possible for the total number of pseudoexhaustive tests needed by C to be significantly less than \( 2^n \). How many tests are needed to pseudoexhaustively test the circuit of Fig. 1?
(b) A related testing method involves exhaustively testing a subcircuit SC with respect to its own inputs rather than with respect to the overall circuit C’s PIs. (This corresponds to the “cell fault” model that we discussed earlier.) In this case, we have to apply (possibly via other circuits) every possible input pattern to SC and propagate any resulting error (possibly via other circuits) to a PO.

Suppose we want to test each of the four subcircuits (partitions) indicated by dotted lines in Fig. 1 in this way. Find the smallest number of tests needed, and construct a minimum test set. If you’re not sure whether your test set is minimal, construct the smallest test you can and estimate how close it is to being minimal.

![Fig. 1](image_url)

Problem 3 (10 points) Memory fault types
Text, page 308, (a) Problem 9.16; (b) Problem 9.18.

Problem 4 (10 points) Memory testing: MATS++
Text, page 308, Problem 9.12.

Problem 5 (20 points) Testability measures: SCOAP
Text, page 151, Problem 6.3. Mark the testability measures on a copy of Fig. 6.21 using the same style as the text (see, for example, Fig. 6.10 on page 139).

Problem 6 (20 points) DFT: scan tests
Text, page 486, Problem 14.4.