3.48. First consider input ABCDE = 00111. With A = 0, input j to gate-5 = 0, and with C = 1, input k to gate-5 is also 0. Thus the output is sensitive to changes in either input to OR-gate-5. Since the output of AND-gate-2 is 0 (due to A), no paths leading to input h of that gate are sensitized.But the signal at h = 1 (due to the fact that C = 1), so that the path from A to Z is sensitized. This means that @1 faults at a, j, or z are detected. Because D = E = 1, the path from input-i to AND-gate-4 is sensitized. (But, because C = 1 makes input-i to that gate = 0, paths from d and e are *not* sensitized.) Of course the output of an inverter is *always* sensitive to its input, so the path from c through inverter-3 to the output is also sensitized. Thus, C@0, g@0, i@1, and k@1 are also detected by this test.
Now consider the other member of the set, ABCDE = 01111. The only difference from the previous case is that B = 1. Since the only gate fed by B, OR-gate-1 is insensitive to B due to the 1 at its other input (C), this change does not change the set of faults detected.

3.49. (a) Note that a@0 is *not* the same as A@0, since the input to the lower NOR-gate is not affected. To detect a@0, clearly it is necessary that A = 1. In order to sensitize the path from a through the upper NOR-gate, we need B = 0. The signal must continue through the next NOR-gate, so the other input to that gate, namely D must also be 0. Extending the sensitized path through the output NOR-gate, requires that signal b be 0. But since A has already been specified as 1, this is already assured. The C-signal has no effect. Hence the answer is the set ABCDE = 10-0.
(b) For a test to detect b@1, the inputs must be such as to make b = 0, and such as to make the upper input to the output gate 0. The first condition is satisfied for all inputs such that A+B+C = 1. The second condition is satisfied iff $(A+B)\bar{D} = 0$. We thus need all solutions to the above pair of simultaneous Boolean equations. Solutions to A+B+C = 1 are {1---, 01--, 001-}. Solutions to $(A+B)\bar{D} = 0$ are {00--, ---1}. Solutions to the pair are the states in the intersection of these 2 sets: {001-, 1--1, 01-1}. (This method is generally applicable.)
(c) The outputs of the 2 gates fed by both A and B are insensitive to all signals at their inputs, since A = B=1. The upper input to the output gate is 1, so the paths through the lower input to that gate are desensitized. Since the lower input to that gate is 0, paths throught its upper input *are* sensitized. Both inputs to the gate feeding that input are 0, so inputs to that gate are sensitized. Thus the detectable faults are on the path from D, and on the path from the output of the upper gate with inputs A and B. The detectable faults are D@1, a @1 fault at the output of the aforementioned gate (or at the input of the gate it feeds), and a @0 fault at the upper input to the output gate, or at the output of the gate feeding that input. A @1 fault at Z is also detectable by this input.
(d) Although replacing the NOR-gates with NAND-gates amounts to taking the dual of the original circuit, the problem is *not* the dual problem, because we are still testing for a@0. The problem would be the dual problem, and the solution the dual of the solution to (a) if we were testing for a@1. To test for a@0, it is necessary to set A = 1 and to sensitize the path from a to Z. Now we require B = 1 to get through the first NAND-gate, and D = 1 to get through the second gate. The lower input to the output gate must now be forced to a 1, i.e. the signal at b must be 1. Since we already require A = B = 1, this can only be achieved by setting C-0. Thus, the only valid test for a@0 is ABCD = 1101. To test for b@1, we must force b for the fault-free circuit to be 0. This requires A = B = C =

1. In addition, the upper input to the output gate must be set to 1, Since A = B = 1, this is already assured without constraining D. So the tests for b@1 are ABCD = 111-. For the NAND-gate circuit with ABCD = 1100, both inputs to the output gate are 1s. But, since both inputs to the gate feeding the upper input to that gate are 0s, no path through that gate is sensitized. Thus, along the upper path, only @0 faults at the upper input to the output gate, or the output of the gate generating that signal are detectable. Along the lower path, @0 faults at b and at the lower input to the output gate are detectable. With C = 0, faults at the A and B inputs to the gate generating b are not detectable, but since A = B = 1, a @1 fault at C (or the gate input to which it is connected) *can* be detected. Finally, a @1 fault at the output is detectable.

6.62. (a) Initially, all inputs to the XOR-gate are 0s, so its output, Y1, is 0. The situation after the first clock pulse is the same as before, it, i.e. all ys are 0, so none of the y's ever changes state, and the output sequence is simply 000, 000, …
(b) $Y_1 = X \oplus y_1 \oplus y_3$. Since X is fixed at 0, this becomes $Y_1 = y_1 \oplus y_3$. Also, $Y_2 = y_1$, and $Y_3 = y2$. Then the following states after y1y2y3 = 110 are: 111, 011, 101, 010, 001, 100, and back to 110.

6.63. (a) $Y_1 = 1 \oplus y_1 \oplus y_3$, $Y_2 = y_1$, and $Y_3 = y_2$. Then we have: $y_1y_2y_3$ = 110, 011, 001, 000, 100, 010, 101, and back to 110.
(b) The first two states are as in problem 6.56b above: 110 and 111. Then the third state is changed by the X = 1 signal from 011 to 111. Following this, we generate the remaining states by applying the relations shown in problem 6.62 to continue the sequence as shown there from the 111 state to obtain: 011, 101, 010, 001, 100, 110, and back to 111. The 7th member of the above sequence is 001, compared with 100 with X fixed at 0.

6.64. $Y_1 = X_3 \oplus y_1 \oplus y_3$, $Y_2 = X_1 \oplus y_1$, $Y_3 = X_2 \oplus y_2$.
[X1, X3]: The X1-signal changes y2 for the second cycle. The X3-signal changes y1 for the third cycle and the y2-change alters y3 during the third cycle. Hence the consequences of the two errors do *not* cancel There is no aliasing.

[X1, X2]: As above, the X1-signal changes y2 for the second cycle. But the effect of the y2-change during the second cycle is cancelled by the X2-signal. Thus, the errors cancel out and have no effect after two cycles. Aliasing has occurred.

[X2, X1]: The X2-signal changes y3 for the second cycle. The X1-signal does *not* cancel the change in y3, but alters y2 for the next cycle. Meanwhile, the y3-change alters y1. Hence there is no aliasing.

[X3, X1X3]: The first X3-signal changes y1 for cycle-2. During cycle-2, X1 cancels the effect of the y1-change on Y2, and X3 cancels the effect of the y1-change on Y1. Hence aliasing does occur.

[X1 X3, X3]: X1 and X3 change y2 and y1 for the second cycle. Then, X3 cancels the effect of the y1-change on Y1, but both y2 and y3 are changed for the third cycle, so there is no aliasing.

[X2, X1X3]: X2 changes Y3. The y3-change is cancelled by X3 in cycle-2. But X1 changes y2 for cycle-3, so there is no aliasing.