shortcuts and well-educated guesses to reduce memory size and execution time to a fraction of what an "exact" algorithm would require. However, rather than finding a provably minimal expression for a logic function, heuristic methods attempt to find an "almost minimal" one.

Even for problems that can be solved by an "exact" method, a heuristic method typically finds a good solution ten times faster. The most successful heuristic program, Espresso-II, does in fact produce minimal or near-minimal results for the majority of problems (within one or two product terms), including problems with dozens of inputs and hundreds of product terms.

3. *Looking at things differently.* As we mentioned earlier, multiple-output minimization can be handled by straightforward, fairly mechanical modifications to single-output minimization methods. However, by looking at multiple-output minimization as a problem in multivalued (nonbinary) logic, the designers of the Espresso-MV algorithm were able to make substantial performance improvements over Espresso-II.

More information on these methods can be found in the References.

*4.5 Timing Hazards

steady-state behavior

The analysis methods that we developed in Section 4.2 ignore circuit delay and predict only the *steady-state behavior* of combinational logic circuits. That is, they predict a circuit's output as a function of its inputs under the assumption that the inputs have been stable for a long time, relative to the delays in the circuit's electronics. However, we showed in Section 3.6 that the actual delay from an input change to the corresponding output change in a real logic circuit is nonzero and depends on many factors.

transient behavior

glitch

hazard

Because of circuit delays, the *transient behavior* of a logic circuit may differ from what is predicted by a steady-state analysis. In particular, a circuit's output may produce a short pulse, often called a *glitch*, at a time when steady-state analysis predicts that the output should not change. A *hazard* is said to exist when a circuit has the possibility of producing such a glitch. Whether or not the glitch actually occurs depends on the exact delays and other electrical characteristics of the circuit. Since such parameters are difficult to control in production circuits, a logic designer must be prepared to eliminate hazards (the *possibility* of a glitch) even though a glitch may occur only under a worst-case combination of logical and electrical conditions.

*4.5.1 Static Hazards

static-1 hazard

A *static-1 hazard* is the possibility of a circuit's output producing a 0 glitch when we would expect the output to remain at a nice steady 1 based on a static analysis of the circuit function. A formal definition is given as follows.

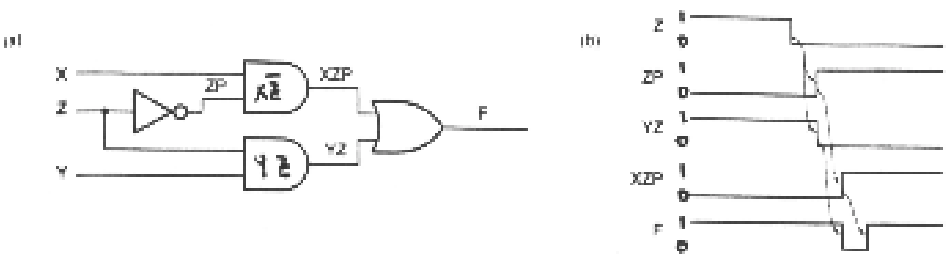


Figure 4-44 Circuit with a static-1 hazard: (a) logic diagram; (b) timing diagram.

Definition: A static-1 hazard is a pair of input combinations that: (a) differ in only one input variable and (b) both give a 1 output; such that it is possible for a momentary 0 output to occur during a transition in the differing input variable.

For example, consider the logic circuit in Figure 4-44(a). Suppose that X and Y are both 1 and that Z is changing from 1 to 0. Then (b) shows the timing diagram, assuming that the propagation delay through each gate or inverter is one unit time. Even though "static" analysis predicts that the output is 1 for both input combinations X,Y,Z = 111 and X,Y,Z = 110, the timing diagram shows that F goes to 0 for one unit time during a 1-0 transition on Z, because of the delay in the inverter that generates Z′.

static-0 hazard

A *static-0 hazard* is the possibility of a 1 glitch when we expect the circuit to have a steady 0 output:

Definition: A static-0 hazard is a pair of input combinations that: (a) differ in only one input variable and (b) both give a 0 output; such that it is possible for a momentary 1 output to occur during a transition in the differing input variable.

Since a static-0 hazard is just the dual of a static-1 hazard, an OR-AND circuit that is the dual of Figure 4-44(a) would have a static-0 hazard.

An OR-AND circuit with four static-0 hazards is shown in Figure 4-45(a). One of the hazards occurs when W,X,Y = 000 and Z is changed, as shown in (b). You should be able to find the other three hazards and eliminate all of them after studying the next subsection.

*4.5.2 Finding Static Hazards Using Maps

A Karnaugh map can be used to detect static hazards in a two-level sum-of-products or product-of-sums circuit. The existence or nonexistence of static hazards depends on the circuit design for a logic function.

A properly designed two-level sum-of-products (AND-OR) circuit has no static-0 hazards. A static-0 hazard would exist in such a circuit only if both a

Reference: Digital Design – Principles and Practices, 3rd Edition, John Wakerly, Prentice-Hall, 2002.

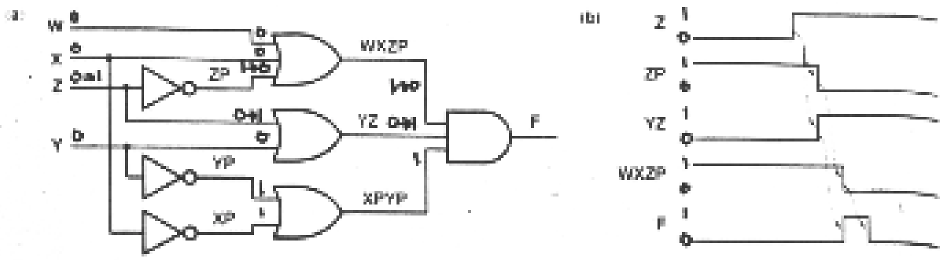


Figure 4-45 Circuit with static-0 hazards: (a) logic diagram; (b) timing diagram.

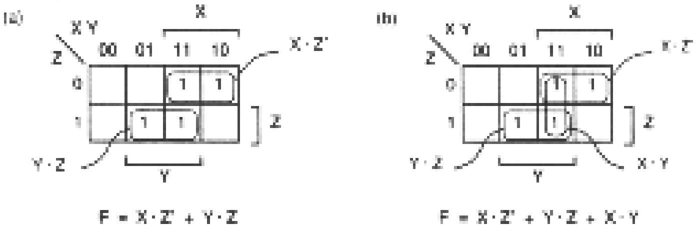
variable and its complement were connected to the same AND gate, which would be silly. However, the circuit *may* have static-1 hazards. Their existence can be predicted from a Karnaugh map where the product terms corresponding to the AND gates in the circuit are circled.

Figure 4-46(a) shows the Karnaugh map for the circuit of Figure 4-44. It is clear from the map that there is no single product term that covers both input combinations X,Y,Z = 111 and X,Y,Z = 110. Thus, intuitively, it is possible for the output to "glitch" momentarily to 0 if the AND gate output that covers one of the combinations goes to 0 before the AND gate output covering the other input combination goes to 1. The way to eliminate the hazard is also quite apparent: Simply include an extra product term (AND gate) to cover the hazardous input pair, as shown in Figure 4-46(b). The extra product term, it turns out, is the *consensus* of the two original terms; in general, we must add consensus terms to eliminate hazards. The corresponding hazard-free circuit is shown in Figure 4-47.

consensus

Another example is shown in Figure 4-48. In this example, three product terms must be added to eliminate the static-1 hazards.

A properly designed two-level product-of-sums (OR-AND) circuit has no static-1 hazards. It *may* have static-0 hazards, however. These hazards can be detected and eliminated by studying the adjacent 0s in the Karnaugh map, in a manner dual to the foregoing.



F = X·Z' + Y·Z

F = X·Z' + Y·Z + X·Y

Figure 4-46 Karnaugh map for the circuit of Figure 4-44: (a) as originally designed; (b) with static-1 hazard eliminated.

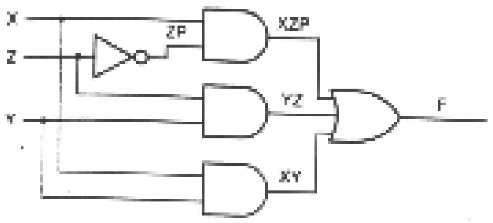


Figure 4-47 Circuit with static-1 hazard eliminated.

*4.5.3 Dynamic Hazards

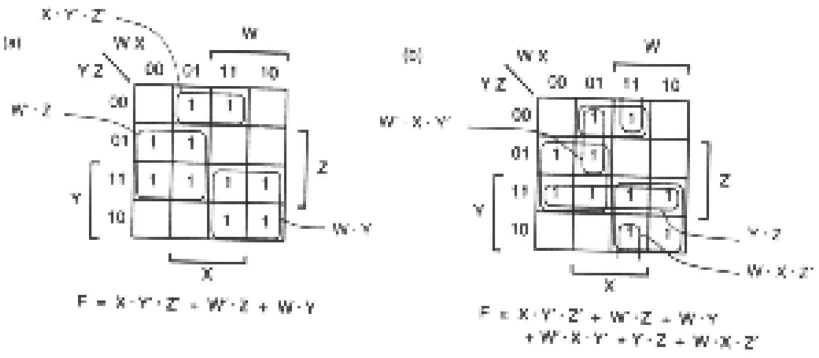
A *dynamic hazard* is the possibility of an output changing more than once as the result of a single input transition. Multiple output transitions can occur if there are multiple paths with different delays from the changing input to the changing output.

dynamic hazard

For example, consider the circuit in Figure 4-49; it has three different paths from input X to the output F. One of the paths goes through a slow OR gate, and another goes through an OR gate that is even slower. If the input to the circuit is W,X,Y,Z = 0,0,0,1, then the output will be 1, as shown. Now suppose we change the X input to 1. Assuming that all of the gates except the two marked "slow" and "slower" are very fast, the transitions shown in black occur next, and the output goes to 0. Eventually, the output of the "slow" OR gate changes, creating the transitions shown in nonitalic color, and the output goes to 1. Finally, the output of the "slower" OR gate changes, creating the transitions shown in italic color, and the output goes to its final state of 0.

Dynamic hazards do not occur in a properly designed two-level AND-OR or OR-AND circuit, that is, one in which no variable and its complement are con-

Figure 4-48 Karnaugh map for another sum-of-products circuit: (a) as originally designed; (b) with extra product terms to cover static-1 hazards.



F = X·Y'·Z' + W'·Z + W·Y

F = X·Y'·Z' + W'·Z + W·Y + W'·X·Y' + Y·Z + W·X·Z'

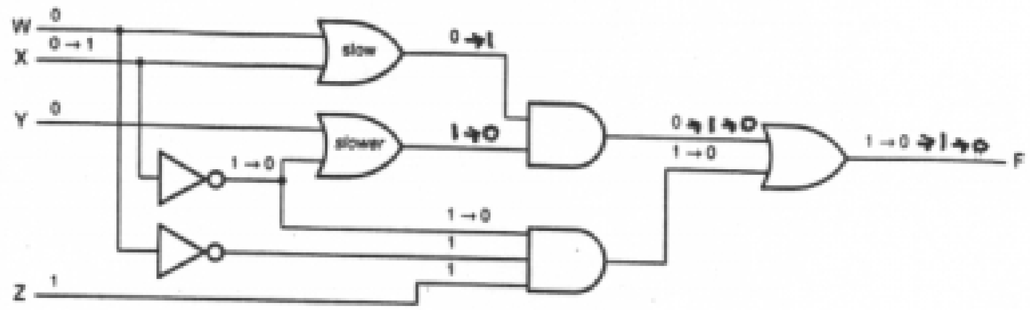


Figure 4-49 Circuit with a dynamic hazard.

nected to the same first-level gate. In multilevel circuits, dynamic hazards can be discovered using a method described in the References.

*4.5.4 Designing Hazard-Free Circuits

Only a few situations, such as the design of feedback sequential circuits, require hazard-free combinational circuits. Techniques for finding hazards in arbitrary circuits, described in the References, are rather difficult to use. So, when you require a hazard-free design, it's best to use a circuit structure that is easy to analyze.

In particular, we have indicated that a properly designed two-level AND-OR circuit has no static-0 or dynamic hazards. Static-1 hazards may exist in such a circuit, but they can be found and eliminated using the map method described earlier. If cost is not a problem, then a brute-force method of obtaining a hazard-free realization is to use the complete sum—the sum of all of the prime implicants of the logic function (see Exercise 4.84). In a dual manner, a hazard-free two-level OR-AND circuit can be designed for any logic function. Finally, note that everything we've said about AND-OR circuits naturally applies to the corresponding NAND-NAND designs, and about OR-AND applies to NOR-NOR.

MOST HAZARDS ARE NOT HAZARDOUS!

Any combinational circuit can be analyzed for the presence of hazards. However, a well-designed, *synchronous* digital system is structured so that hazard analysis is not needed for most of its circuits. In a synchronous system, all of the inputs to a combinational circuit are changed at a particular time, and the outputs are not "looked at" until they have had time to settle to a steady-state value. Hazard analysis and elimination are typically needed only in the design of asynchronous sequential circuits, such as the feedback sequential circuits discussed in Section 7.9. You'll rarely have reason to design such a circuit, but if you do, an understanding of hazards will be absolutely essential for a reliable result.