```
1   open: assert ((filed = $fopen("fname.dat", "w")) != 0)
2     else $error ("oops! %m can't open file %s", "fname.dat");
```

Using immediate assertions in this way simplifies the coding of bothersome error conditions that need to be checked!

9.2 Introduction to Concurrent Assertions

A concurrent assertion is an independently executing element that performs checking on a design. The purpose of a concurrent assertion is to check that a property of the design holds over the time that a system is being simulated. The property is defined and then a concurrent assert statement activates the checking of the property. The assertion is clock based, following the RT timing model. It executes in the simulation kernel's observed region based on values sampled in the kernel's preponed region.

9.2.1 Defining and Asserting a Property

A *property* defines the behavior of a design to be checked. Properties can be quite complex and can be made up of several sequences of actions. An analogy would be to say that they are regular expressions that include clock edges.

A concurrent assertion is asserted in the following manner:

```
1   label: assert property (pname) pass_Statement else fail_Statement;
```

The words assert, property, and else are keywords of the language. The label is a label for the statement, pname is the name of a property, pass_Statement is the statement executed when the property passes correctly, and fail_Statement is the statement executed if the property fails. The label and pass_Statement are optional.

```
1   module simpleAssert;
2     bit  q, r, s, ck;
3     ...
4     property q1r3s;
5       @(posedge ck) q ##1 r ##3 s;
6     endproperty
7
8     assert property (q1r3s) else $error("oops");
9   ...
```

Example 9.2 — A Concurrent Assertion

We start with a very simple property example. A property named q1r3s is defined on lines 4-6 of Example 9.2. Line 4 specifies the property's name. Its action (line 5) is that at the positive edge of clock ck it will check if q is TRUE. If it is, then one clock tick later (as specified by ##1) r should be TRUE and then three clock ticks after that s should be TRUE.

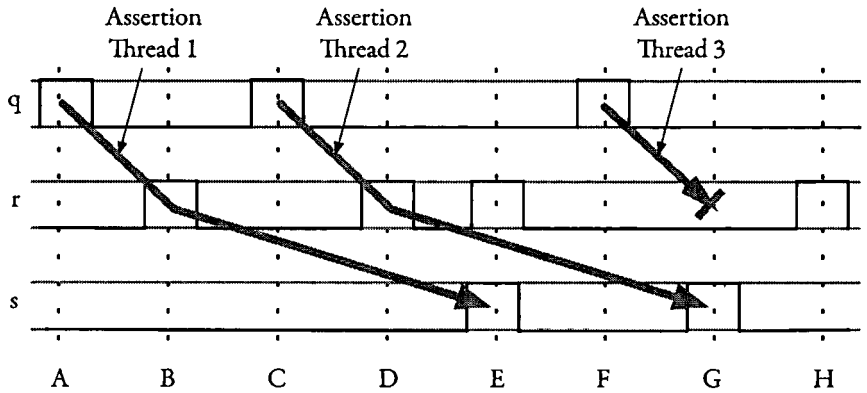

Figure 9.1 — Sequence Examples for property a1b3c