Figure 9.11 shows two waveforms, q and r. It also shows several instances of the sampled value functions and their values at specific points along the waveform. Most are self explanatory. Note that $fell(r) at clock tick C is 0 because r changed to 0 right at the clock tick; its preponed value is 1 and thus $fell won't recognize the change. However, $fell(r) is TRUE at the next clock tick.

Also note that $rose(q) is 0 at clock tick A. This is a result of q being defined to be a bit variable in the simulation for this figure. If q had been defined as a logic variable, $rose(q) would be 1'bx at this point.

9.5.2 Sequences Using Sampled Value Functions

Consider first the pipelinedFunction property from Section 9.4. The example was of a pipeline multiplier that read its inputs at every clock event and produced its result 3 clock ticks later. Local variables were used to sample and remember the input values for later comparison.

Another approach to writing the property uses sampled value functions:

```
property pipelinedFunctionPast(inA, inB);
  @(posedge ck) disable iff (reset)
    result == ($past(inA, 3) * $past(inB, 3));
endproperty
```

Here the values of inA and inB are retrieved by the $past function looking back 3 clock ticks. The product of those two values are compared with the result generated by the pipeMult module in the current time.

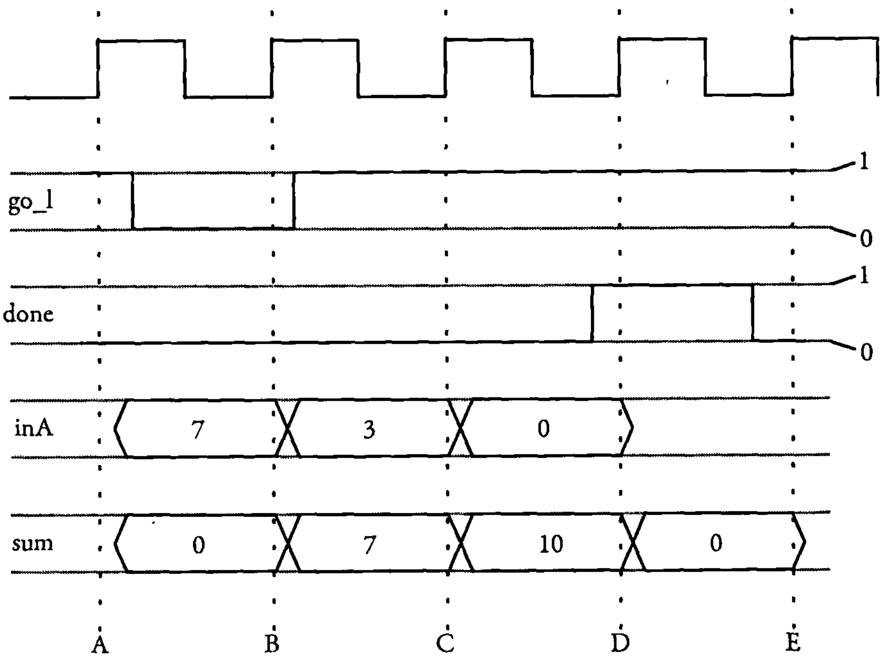

Figure 9.12 — Timing Diagram For sumItUp Thread