

Proposed modifications to section 9.3 shown below

9.3 The stratified event scheduler

A compliant SystemVerilog simulator must maintain some form of data structure that allows events to be dynamically scheduled, executed, and removed as the simulator advances through time. The data structure is normally implemented as a time-ordered set of linked lists, which are divided and subdivided in a well defined manner.

The first division is by time. Every event has one and only one simulation execution time, which at any given point during simulation can be the current time or some future time. All scheduled events at a specific time define a time slot. Simulation proceeds by executing and removing all events in the current simulation time slot before moving on to the next nonempty time slot, in time order. This procedure guarantees that the simulator never goes backwards in time.

A time slot is divided into a set of ordered regions:

- a) Preponed
- ~~b) Pre-active~~
- b) Pre-Active
- c) Active
- d) Inactive
- e) Pre-NBA
- f) NBA
- g) Post-NBA
- h) Observed
- ~~i) Post-observed~~
- i) Post-Observed
- j) Reactive
- ~~k) Re-inactive~~
- l) Re-Inactive
- m) Pre-Re-NBA
- n) Re-NBA
- o) Post-Re-NBA
- ~~p) Pre-postponed~~
- p) Pre-Postponed
- q) Postponed

The purpose of dividing a time slot into these ordered regions is to provide predictable interactions between the design and testbench code.

~~Except for the Observed, Reactive, and Re-inactive and the Post-observed PLI region, these~~ The Pre-Active, Active, Inactive, Pre-NBA, NBA, Post-NBA, Pre-Postponed and Postponed regions essentially encompass the IEEE 1364-2005 reference model for simulation, with exactly the same level of determinism. In other words, legacy Verilog code shall continue to run correctly without modification within the new mechanism. ~~The~~

~~Postponed region is where the monitoring of signals, and other similar events, takes place. No new value changes are allowed to happen in the time slot once the Postponed region is reached.~~

9.3.1 Active region & reactive region sets

There are two important groupings of event regions that are used to help define the scheduling of SystemVerilog activity, they are the *active region set* and the *reactive region set*. Events scheduled in the Active, Inactive, Pre-NBA, NBA, and Post-NBA regions are active region set events. Events scheduled in the Reactive, Re-Inactive, Pre-Re-NBA, Re-NBA, and Post-Re-NBA regions are reactive region set events.

The Active, Inactive, Pre-NBA, NBA, Post-NBA, Observed, ~~Post-Observed~~ ~~Post-observed~~, Reactive, ~~Re-Inactive~~ ~~Re-inactive~~, Pre-Re-NBA, Re-NBA, Post-Re-NBA, and Pre-Postponed ~~Pre-postponed~~ regions are known as the *iterative* regions.

In addition to the active region set and reactive region set, all of the event regions of each time slot can be categorized as either *simulation regions* (see 9.3.2) or *PLI regions* (see 9.3.3).

The ~~new~~ **#1step** sampling delay provides the ability to sample data immediately before entering the current time slot and is a preferred construct over other equivalent constructs because it allows the **1step** time delay to be parameterized. This **#1step** construct is a ~~conceptual~~ mechanism that provides a method for defining when sampling takes place and does not require that an event be created in ~~the this~~ previous time slot. ~~Conceptually, this~~ The **#1step** sampling is identical to taking the data samples in the Preponed region of the current time slot.

9.3.2 Simulation regions

The simulation regions of a time slot are the Active, Inactive, NBA, Observed, Reactive, Re-Inactive, Re-NBA and Postponed regions. The flow of execution of the event regions is specified in Figure 9-1.

~~The Observed, Reactive, and Re-Inactive Re-inactive, and Re-NBA regions are new in this standard, and events are only scheduled into these new regions from new language constructs.~~

9.3.2.1 Active events region

The Active region holds the current *active region set* events being evaluated and can be processed in any order.

9.3.2.2 Inactive events region

The Inactive region holds the events to be evaluated after all the ~~active~~ *Active* events are processed.

If events are being executed in the active region set, an ~~An~~ *explicit* #0 delay control requires that the process be suspended and an event scheduled into the Inactive region ~~(or Re-inactive-for-program-blocks)~~ of the current time slot so that the process can be resumed in the next ~~inactive-to-active~~ Inactive to Active iteration.

9.3.2.3 NBA events region

The NBA (nonblocking assignment update) region holds the events to be evaluated after all the Inactive events are processed.

If events are being executed in the active region set, a ~~A~~ nonblocking assignment creates an event in the NBA region scheduled for the current or a later simulation time.

9.3.2.4 Observed events region

The Observed region is for the evaluation of the property expressions when they are triggered. A criterion for this determinism is that the property evaluations must only occur once in any clock triggering time slot. During the property evaluation, pass/fail code shall be scheduled in the Reactive region of the current time slot. PLI callbacks are not allowed in the Observed region.

9.3.2.5 Reactive events region

The Reactive region holds the current reactive region set events being evaluated and can be processed in any order.

The code specified by *blocking assignments* in the program block and the pass/fail code from property expressions are scheduled in the Reactive region. ~~A #0 control delay specified in a program block schedules the process for resumption in the Re-inactive region. The Re-inactive region is the program block dual of the Inactive region (see 9.3.2.2 below).~~ The Reactive region is the reactive region set dual of the Active region (see 9.3.2.1).

9.3.2.6 Re-Inactive events region

The Re-Inactive region holds the events to be evaluated after all the Reactive events are processed.

If events are being executed in the reactive region set, an *explicit* #0 delay control requires that the process be suspended and an event scheduled into the Re-Inactive region of the current time slot so that the process can be resumed in the next Re-Inactive to Reactive iteration. The Re-Inactive region is the reactive region set dual of the Inactive region (see 9.3.2.2).

9.3.2.7 Re-NBA events region

The Re-NBA region holds the events to be evaluated after all the Re-Inactive events are processed.

If events are being executed in the reactive region set, a nonblocking assignment creates an event in the Re-NBA update region scheduled for the current or a later simulation time. The Re-NBA region is the reactive region set dual of the NBA region (see below).

9.3.2.8 Postponed events region

The Postponed region is where the monitoring of signals, and other similar events, takes place.

No new value changes are allowed to happen in the time slot once the Postponed region is reached. Within this region, it is illegal to write values to any net or variable or to schedule an event in any previous region within the current time slot.

Postponed region PLI events are also scheduled in this region (see 9.3.3.9).

~~The Pre-Active Pre-active, Pre-NBA, and Post-NBA regions are new in this standard but support existing PLI callbacks.~~

~~The Pre-Re-NBA, Post-Re-NBA and Post-Observed Post-observed regions are is new in this standard and have has been added for PLI support in the reactive region set.~~

9.3.3 PLI regions

The PLI regions of a time slot are the Preponed, Pre-Active, Pre-NBA, Post-NBA, Post-Observed, Pre-Re-NBA, Post-Re-NBA and Pre-Postponed regions. The flow of execution of the PLI regions is specified in Figure 9-1.

9.3.3.1 Preponed PLI region

The Preponed region provides for a PLI callback control point that allows PLI application routines to access data at the current time slot before any net or variable has changed state. Within this region, it is illegal to write values to any net or variable or to schedule an event in any other region within the current time slot.

NOTE—The PLI currently does not schedule callbacks in the Preponed region.

9.3.3.2 Pre-Active PLI region

The **Pre-Active** ~~Pre-active~~ region provides for a PLI callback control point that allows PLI application routines to read and write values and create events before events in the Active region are evaluated (see 9.4).

9.3.3.3 Pre-NBA PLI region

The Pre-NBA region provides for a PLI callback control point that allows PLI application routines to read and write values and create events before the events in the NBA region are evaluated (see 9.4).

9.3.3.4 *Post-NBA PLI region*

The Post-NBA region provides for a PLI callback control point that allows PLI application routines to read and write values and create events after the events in the NBA region are evaluated (see 9.4).

9.3.3.5 *Post-Observed PLI region*

The ~~Post-Observed~~ ~~Post-observed~~ region provides for a PLI callback control point that allows PLI application routines to read values after properties are evaluated (in Observed or earlier region).

NOTE—The PLI currently does not schedule callbacks in the ~~Post-Observed~~ ~~Post-observed~~ region.

9.3.3.6 *Pre-Re-NBA PLI region*

The Pre-Re-NBA region provides for a PLI callback control point that allows PLI application routines to read and write values and create events before the events in the Re-NBA region are evaluated (see 9.4).

9.3.3.7 *Post-Re-NBA PLI region*

The Post-Re-NBA region provides for a PLI callback control point that allows PLI application routines to read and write values and create events after the events in the Re-NBA region are evaluated (see 9.4).

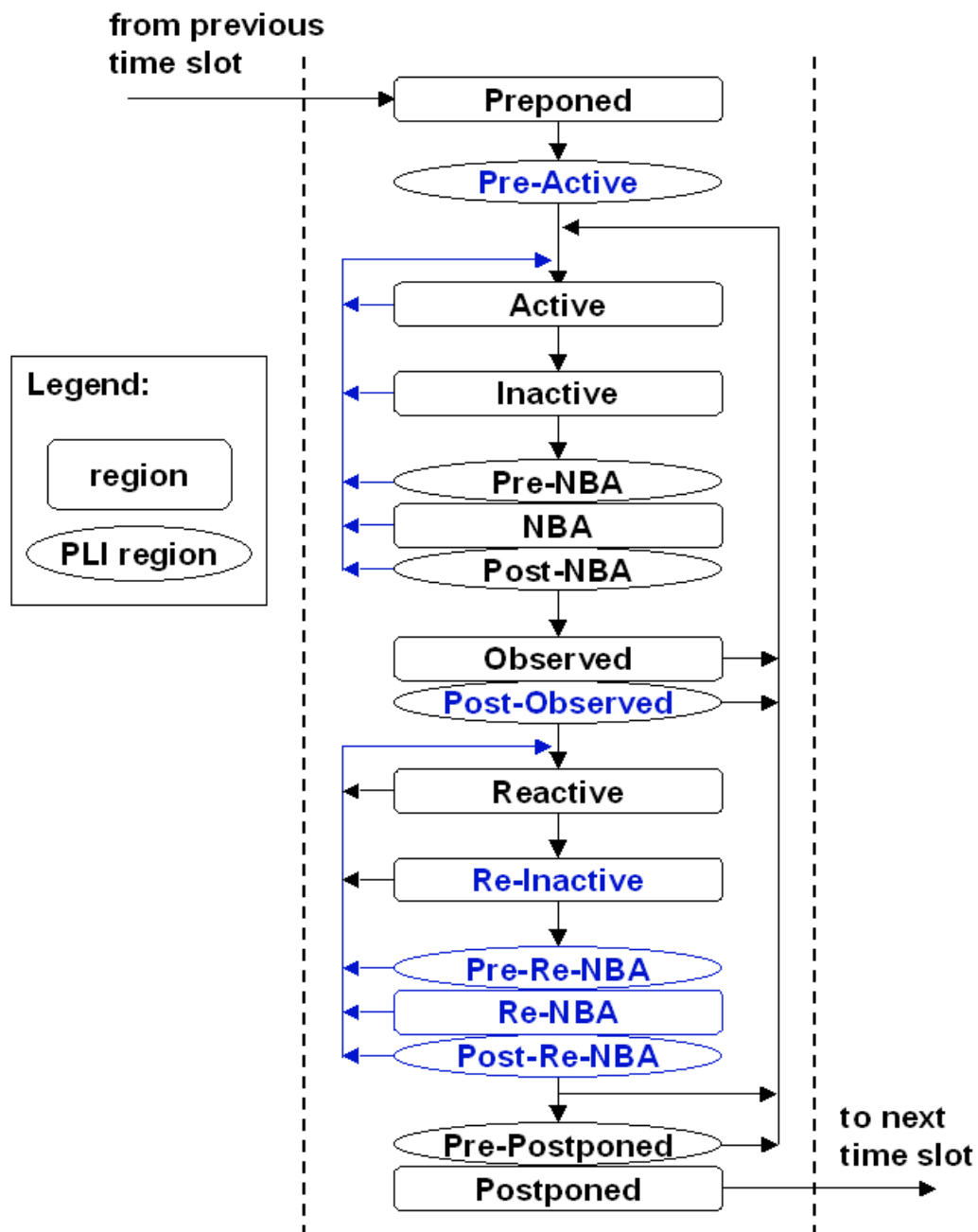
9.3.3.8 *Pre-Postponed PLI region*

The ~~Pre-Postponed~~ ~~Pre-postponed~~ region provides a PLI callback control point that allows PLI application routines to read and write values and create events after processing all other regions except the Postponed region. The flow of execution of the event regions is specified in Figure 9-1.

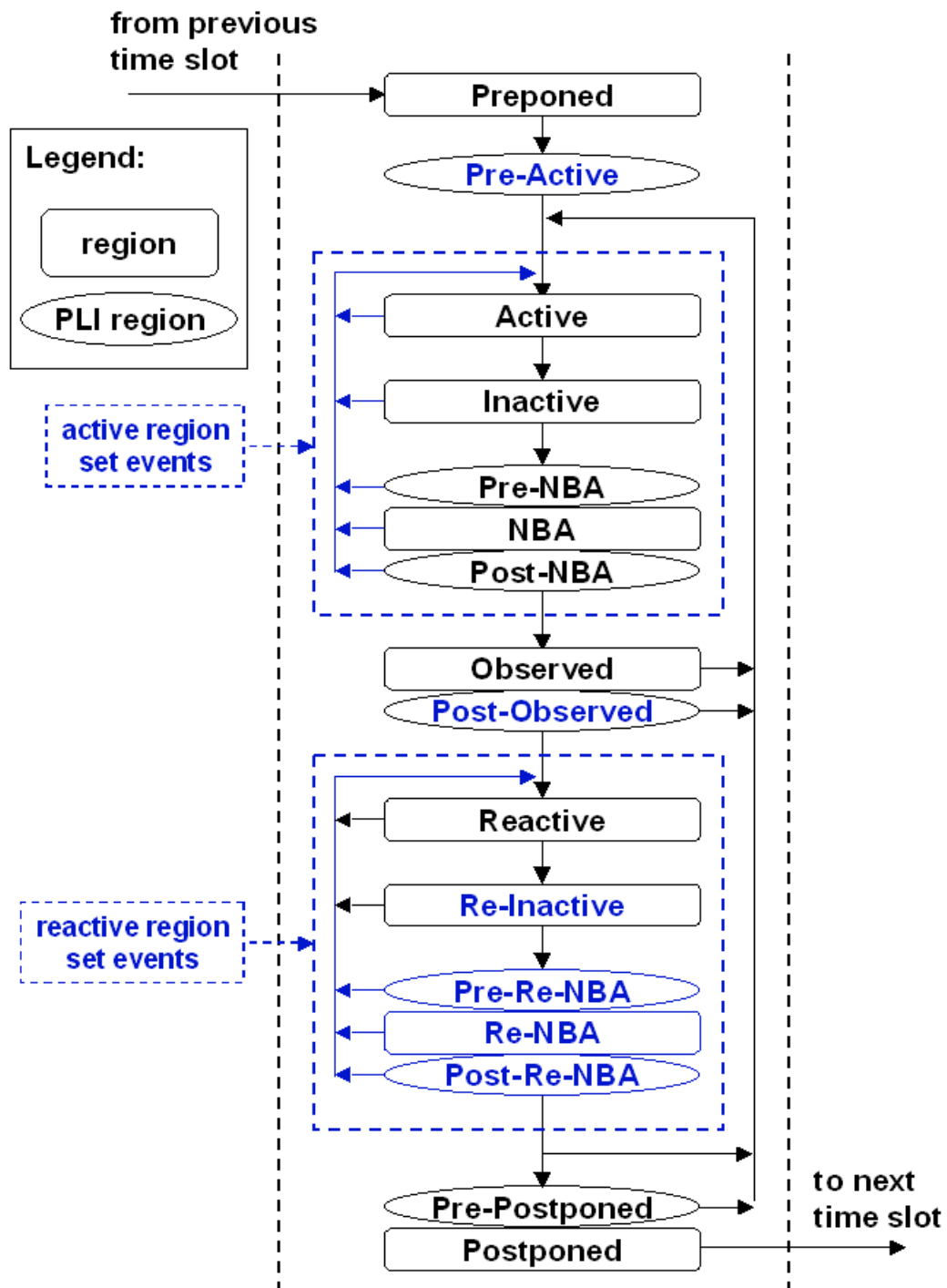
9.3.3.9 *Pre-Active PLI events*

The Postponed region **also** provides for a PLI callback control point that allows PLI application routines to be suspended until after all the Active, Inactive, NBA, Observed, Reactive, **and** ~~Re-Inactive~~ ~~Re-inactive~~, **and** Re-NBA regions have completed.

<NEW FIG 9.1> (There are two choices proposed - selection to be made in the SV-EC)



<NEW FIG 9.1> (There are two choices proposed - selection to be made in the SV-EC)



9.3.1 9.3.4 The SystemVerilog simulation reference algorithm

```

execute_simulation {
    T = 0;
    initialize the values of all nets and variables;
    schedule all initialization events into time 0 slot;
    while (some time slot is nonempty) {
        move to the next future nonempty time slot and set T;
        execute_time_slot (T);
    }
}

execute_time_slot {
    execute_region (Preponed preponed);
    execute_region (Pre-Active pre-active);
    while (any region in [Active active... Pre-Postponed pre-postponed] is
nonempty) {
        while (any region in [Active active... Post-Observed post-observed] is
nonempty) {
            execute_region (Active active);
            R = first nonempty region in [Active active... Post-Observed post-
observed];
            if (R is nonempty)
                move events in R to the Active active region;
        }
        while (any region in [Reactive reactive... re-inactive Post-Re-NBA] is
nonempty) {
            execute_region (Reactive reactive);
            R = first nonempty region in [Reactive reactive... re-inactive Post-Re-
NBA];
            if (R is nonempty)
                move events in R to the Reactive reactive region;
        }
        if (all regions in [Active active ... re-inactive Post-Re-NBA] are empty)
            execute_region (Pre-Postponed pre-postponed);
    }
    execute_region (Postponed postponed);
}

execute_region {
    while (region is nonempty) {
        E = any event from region;
        remove E from the region;
        if (E is an update event) {
            update the modified object;
            evaluate processes sensitive to the object and possibly schedule
            further events for execution;
        } else { /* E is an evaluation event */
            evaluate the process associated with the event and possibly
            schedule further events for execution;
        }
    }
}

```

The Iterative regions and their order are Active, Inactive, Pre-NBA, NBA, Post-NBA, Observed, ~~Post-Observed~~ ~~Post-observed~~, Reactive, ~~Re-Inactive~~ ~~Re-inactive~~, ~~Pre-Re-NBA~~, ~~Re-NBA~~, ~~Post-Re-NBA~~ and ~~Pre-Postponed~~ ~~Pre-postponed~~. As shown in the algorithm, once the Reactive, ~~or~~ ~~Re-Inactive~~, ~~Pre-Re-NBA~~, ~~Re-NBA~~ or ~~Post-Re-NBA~~ regions are processed, iteration over the other regions does not resume until these ~~two~~ ~~five~~ regions are empty.

NOTE TO EDITOR:

The event region names in the *SystemVerilog simulation reference algorithm* have been modified to meet the standard naming for these event regions in the rest of this section. This serves two purposes: (1) the event region names are more visible in the printed algorithm, and (2) it will make it easier to make the global substitutions required below.

Please make the global event-region replacements in the merged document as follows (it will be easier for the editor to find all of these occurrences in the document than to list out all of the changes and have the editor check off each change individually - it will also be easier for reviewers to search for the replaced words shown below to verify that all changes have been updated as expected).

(note some of the invalid names may not exist anywhere in the merged document):

WAS: ~~prepone~~ -or- ~~preponed~~
PROPOSED: Preponed

WAS: ~~Pre-active~~ -or- ~~pre-active~~
PROPOSED: Pre-Active

(take special care not to globally replace the word active when not used as an event region)

WAS: ~~active~~
PROPOSED: Active

(take special care not to globally replace the word inactive when not used as an event region)

WAS: ~~inactive~~
PROPOSED: Inactive

WAS: ~~nba~~
PROPOSED: NBA

WAS: ~~pre-NBA~~
PROPOSED: Pre-NBA

WAS: ~~post-NBA~~
PROPOSED: Post-NBA

WAS: ~~Observed~~
PROPOSED: Observed

WAS: ~~Post-observed~~ -or- ~~post-observed~~
PROPOSED: Post-Observed

WAS: ~~Reactive~~

PROPOSED: Reactive

WAS: ~~Re-inactive~~ -or- ~~re-inactive~~

PROPOSED: Re-Inactive

(Pre-Re-NBA was just added and should be correctly spelled everywhere)

(Re-NBA was just added and should be correctly spelled everywhere)

(Post-Re-NBA was just added and should be correctly spelled everywhere)

WAS: ~~Pre-postponed~~ -or- ~~pre-postponed~~

PROPOSED: Pre-Postponed

WAS: ~~Postponed~~

PROPOSED: Postponed