Modify Syntax of Section 20.2 (page 286) and syntax A.2.11 (page 373) as shown

covergroup_declaration ::=   *// from Annex A.2.11*
      covergroup covergroup_identifier [**(** list_of_task_proto **)**] [~~clocking~~coverage_event] **;**
         { coverage_spec_or_option **;** }
      endgroup [**:** covergroup_identifier]

list_of_task_proto ::= task_proto_formal { **,** task_proto_formal }

coverage_spec_or_option ::=
       {attribute_instance} coverage_spec
     | {attribute_instance} coverage_option

coverage_option ::=
       option.option_name **=** expression
     | type_option.option_name **=** expression

coverage_spec ::=
        cover_point
     | cover_cross

coverage_event ::=
       clocking_identifier
     | **@@(** block_event_expression **)**

block_event_expression :: =
       block_event_expression **or** block_event_expression
     | **begin** hierarchical_btf_identifier
     | **end** hierarchical_btf_identifier

hierarchical_btf_identifier :: =
       hierarchical_task_identifier
     | hierarchical_function_identifier
     | hierarchical_block_identifier
     | hierarchical_identifier { class_scope **::** } method_identifier

variable_decl_assignment ::=   *// from Annex A.2.4*
      …
     | covergroup_identifier variable_identifier **= new** [**(** list_of_arguments **)**][19]

19. It shall be legal to omit the covergroup_variable_identifer from a covergroup instantiation only if this implicit nstan-
tiation is within a class that has no other instantiation of the covergroup.

Syntax 20-1—Covergroup syntax (excerpt from Annex A)

Add the following paragraph to Section 2.20 (page 287)

…
The strobe option (see Section 20.6.1) can be used to specify that coverage points are sampled at the end of the
time slot, thereby filtering multiple clocking events so that only sample per time slot is taken.

1

As an alternative to a clocking event, a coverage group accepts a block event expression to indicate that the coverage sample is to be triggered by the start or the end of execution of a given named block, task, function, or class method. Block event expressions that specify the **begin** keyword followed by a hierarchical identifier denoting a named block, task, function, or class method shall be triggered immediately before the corresponding block, task, function, or method begins executing its first statement. Block event expressions that specify the **end** keyword followed by a hierarchical identifier denoting a named block, task, function, or class method shall be triggered immediately after the corresponding block, task, function, or method executes its last statement. Block event expressions that specify the **end** of execution shall not be triggered if the block, task, function, or method is disabled.

A `covergroup` can contain one or more coverage points. A coverage point can be a variable or an expression. Each coverage point includes a set of bins associated with its sampled values or its value transitions. The bins can be explicitly defined by the user or automatically created by the tool. Coverage points are discussed in detail in Section 20.4.