

Annex O Use of STIL.1 for specific applications

To be compliant with a standard, a user should apply all parts of that standard that make sense to the application. There are always parts of a standard that do not make sense for a specific application - for instance a hardware circuit description may not use part of the defined circuitry because the functionality supported by that part of the circuitry is not part of the current design. Likewise, not all parts of STIL.1 will be used or usable in all applications. The identification of unusable parts is the responsibility of the application.

O.1 STIL.1 usage by ATE systems

To be compliant with STIL.1, an ATE system needs to process the additional constructs as defined below. Some statements are required to be fully supported by the ATE, while others may be ignored. Unless indicated by the comment below, it is expected that the ATE fully support this feature.

It is anticipated that any given ATE system may not be able to fully implement all constructs possible in the STIL.1 extension. In that case, it is expected that the ATE load process or a pre-process application will make necessary adjustments to the file to make it fit the needs of the ATE system.

STIL 1.0 { **Design** 2004; } // signal that a file contains STIL.1 syntax

Variables // ATE support named and global variable domains

IntegerConstant

Integer

Usage Test // ATE support for a given integer only when "Usage Test" is specified

InitialValue

SignalVariable

Base < **Hex** | **Dec**>

Alignment

InitialValue

WFCConstant

Signals

WFCMap

SignalGroups

WFCMap

PatternBurst

Variables

Fixed

PatList

Variables

Fixed

If

While

PatSet

Variables

Fixed

ParallelPatList (**SyncStart** | **Independent** | **LockStep**)

Variables

Fixed

Wait

Extend

If

While

Timing

Variables

ScanStructures // ATE should allow, but no action required

ScanCellGroups // ATE should allow, but no action required

Environment // ATE should allow, but no action required

pattern data

sigref_expr = \readback_function
If (*readback_function*)

Pattern statements:

(LABEL :) **If Else**

(LABEL :) **While**

(LABEL :) **F(ixed)**

(LABEL :) **E(quivalent)**

(LABEL :) **LoopData**

(LABEL :) **Loop** *integer_expr*

(LABEL :) **ActiveScanChains** // ATE should allow, but no action required

(LABEL :) **AllowInterleave** // ATE should do error check only

(LABEL :) **BreakPoint**

(LABEL :) **X**

Pragma // ATE should ignore, unless pragma is specific to that system

PatternFailReport // ATE should generate this data

Pattern

PatternBurst

PatternExec

FailData

O.2 STIL.1 usage for sub-patterns

Much of the additional syntax in STIL.1 is used for the definition of embedded cores and re-usable-patterns. Below is a summary of the structures defined in this standard that are important in the context of: a) defining re-usable-patterns that are used by SoC patterns, or b) defining embedded cores. Refer to the standard that defines embedded core specification for the specifics of the usage of these constructs.

Note that this information is given only to indicate that the statements listed are important to this application. A given file may or may not use any of these constructs. This information is not meant to limit the use of other statements that may be included in the file for other reasons.

STIL 1.0 { **Design 2004; CTL 2004;** }

Variables // used by re-usable-patterns

IntegerConstant // used by re-usable-patterns

Integer // used by re-usable-patterns

SignalVariable // used by re-usable-patterns

WFCConstant // used by re-usable-patterns

PatternBurst PAT_BURST_NAME {

Variables VARIABLES_DOMAIN // used by re-usable-patterns

Fixed // used by re-usable-patterns

PatList

Variables VARIABLES_DOMAIN // used by re-usable-patterns

Fixed // used by re-usable-patterns

PatSet // used by embedded cores

Variables VARIABLES_DOMAIN // used by re-usable-patterns

Fixed // used by re-usable-patterns

ParallelPatList (SyncStart | Independent | LockStep) // used by re-usable-patterns

Variables VARIABLES_DOMAIN // used by re-usable-patterns

Fixed // used by re-usable-patterns

Wait // used by re-usable-patterns

Extend // used by re-usable-patterns

Timing

Variables VARIABLES_DOMAIN // used by re-usable-patterns

ScanStructures // used by embedded cores

ScanCellGroups // used by embedded cores

Environment // used by embedded cores

pattern data:

sigref_expr = \readback_function // used by re-usable-patterns

If (\readback_function) { } // used by re-usable-patterns

Pattern statements:

(LABEL :) **F(ixed)** // used by re-usable-patterns

(LABEL :) **E(quivalent)** // used by re-usable-patterns

(LABEL :) **LoopData** // used by re-usable-patterns

(LABEL :) **Loop** *integer_expr* // used by re-usable-patterns

(LABEL :) **AllowInterleave** // used by re-usable-patterns