

IEEE Std. 1450-1999 Clarifications and Corrections

This document is maintained by the STIL Working Group to clarify points of confusion identified by users of 1450-1999. These clarification are sorted by page order in the 1450-1999 standard. These clarifications are expected to become part of the 1450.0 standard at the next approval phase.

Cover (hardcopy only)

Title *Standard Interface Test* should be ***Standard Test Interface***

[Typographical]

Clause 6.3: Reserved words

Table 1, third row, identifies the keywords *CompareZ* and *CompareZWindow*. These keywords should be ***CompareOff*** and ***CompareOffWindow*** to match Clause 18.2, Table 10 and examples. Several entries are missing from Table 1. Add: ***C*** as the first entry on row 3, ***D*** as the first entry on row 4, ***h*** as the second entry on row 8, ***l*** as the second entry on row 10, ***t*** as the second entry on row 18, ***v*** as the second entry on row 20, and ***x*** as the second entry on row 22.

[Reference 8]

[Reference 22]

Clause 6.4: Reserved characters

Table 2 is missing the single character ***?***, used to designate the unknown event as defined in Clause 18.2, Table 12.

[Reference 8]

Clause 6.8: User-defined name characteristics

At the end of paragraph 4, add the sentence:

The name “*XYZ*” (with quotes) is distinct from the name *XYZ* (without quotes). Both forms may be present and may represent different signals in a STIL context.

At the end of the last paragraph in this clause, add the following 4 sentences:

Square brackets, like quotes, are part of a signal name when present. If a name is declared with square brackets present, then all references to the signal or signals defined must have square brackets present. The name *XYZ[0..5]* in STIL is distinct from the name *XYZ* (without square brackets). Both forms may be present and may represent different signals in a STIL context.

[Reference 9]

Clause 6.14: Signal expressions

Fourth paragraph, word *memeber* should be ***member***

[Typographical]

Clause 6.15: WaveformChar characteristics

After the sentence, “A WaveformChar list may contain whitespace, including newlines.”, add the following:

Whitespace is required to separate tokens in WaveformChar lists which exceed the 1024 character token limit.

[Reference 4]

Clause 6.16: STIL name spaces and name resolution

Add the following sentence to the end of this clause (note the referenced subclause is new and defined as part of this document):

See subclause 15.5 for additional information on the resolution of SignalGroup names.

[Reference 5]

Clause 7: Statement Structure

Add the following sentences at the end of the second paragraph:

The optional statements within the braces of a STIL block statement have no ordering imposed upon them. They may appear in any order except in sequence-dependent blocks (Pattern, MacroDef, and Procedure blocks containing STIL Pattern Statements as defined in Clause 22, entries under the PatList in the PatternBurst block as defined in Clause 17, and SignalGroup name definitions as defined in Clause 15).

[Reference 2]

Add the following sentences at the end of the third paragraph:

STIL allows statements to be either in semicolon-terminated or brace-terminated format. Statements identified in this document in semicolon format do not always define contents for a brace-formatted statement, none the less that format of statement is allowed and supports user-defined blocks and Annotations to be defined as part of that statement. This capability is used extensively in STIL extension work and in support of UserKeyword applications.

[Reference 3]

Clause 8: STIL statement

Change the first sentence to add the additional clarification:

The STIL statement shall be the first statement of each STIL file, including Included files.

[Reference 6]

Clause 10: Include statement

Add to the end of the first paragraph:

Only STIL files (files beginning with a STIL statement) can be referenced with the Include statement.

[Reference 7]

Clause 15.3: Default attribute values

The defaults defined in this section apply to the Signals as well as the SignalGroups, but the wording in this section does not indicate that. Modify the first paragraph as follows:

When a new group **or signal** is declared, it is assigned the following set of default property values. These values **are** overridden only by explicit declaration of property values **on a group or individual Signal definition. Termination and DefaultState values shall be defined only once (either in a group or on the individual signal) for each signal.**

[Reference 17]

Clause 15.5: SignalGroup name resolution (new)

Add this section:

A SignalGroup definition is resolved with the following rules:

1. Resolution occurs when a SignalGroup name is defined inside a SignalGroups block. Resolution cannot depend on SignalGroup names defined across multiple SignalGroups blocks, but must be contained as defined in the next point.
2. Resolution of a SignalGroup name uses the following order of defined names:
 - 2.1. A reference is matched against the names defined in this SignalGroups block. A match resolves to the references contained in that name.
 - 2.2. A reference is matched against the names defined in the global SignalGroups block. A match resolves to the references contained in that name,
 - 2.3 A reference is matched against the names defined in the Signals block.
3. A Signal name cannot be changed in the global SignalGroups block (as defined in Sub-clause 6.16).

These rules define that resolution is order-dependent in the SignalGroup; a name is resolved against only those definitions found earlier in that SignalGroups block. Also, resolution across named signalgroups is not allowed; resolution of a SignalGroup name can only depend on Signals and global groups, and previous definitions in that group.

To support this resolution process:

- A. The Signals block shall be defined before any SignalGroups.
- A. A global SignalGroups block, if present, shall be present before any domain-named SignalGroups blocks.
- B. A SignalGroup name that is dependent on another SignalGroup defined in that block, shall have the dependent name defined before this name is defined.

[Reference 5]

Clause 18.4: Rules for timed event ordering and waveform creation

At the end of the second paragraph, add:

For drive and compare events (Table 9 and 10), the last defined value of that category shall persist until a new value in that category is specified, as presented in Figure 31. In addition, ForceUnknown events shall override the last drive event as specified below, and CompareLow, CompareHigh, CompareOff, and CompareValid events (edge-events) operate as if

immediately followed by a CompareUnknown event; the last value after these events shall be an **implicit** CompareUnknown to terminate the compare operation. Expect events (Table 11) are valid only for the moment that the event is defined and shall be undefined at any other point. Unresolved events (Table 12) are handled as follows: ForceUnknown events shall override (and are overridden) by other drive events and shall also override any previous unresolved event if present. All other unresolved events shall affect unresolved events only.

[Reference 19]

Clause 19: Spec and Selector Blocks

The 6th paragraph in this section has a reference pointing to the wrong clause number. In the paragraph that starts: “The Selector Block is used...”, change the second sentence:

Refer to Clause 17...

To:

Refer to Clause 16...

[Typographical]

Clause 21.1: Cyclized data

Under the definition of `vec_data`, after the second sentence and parenthetical sentence, add:

A SIGNAL-NAME shall be assigned a new value (one or more WaveformChars) in at most one cyclized data assignment statement per Vector.

[Reference 12]

Under the definition of `\l` (last paragraph of this section), add:

When present, the number of WaveformChars in the assignment shall be greater than or equal to the integer value specified in the `\l` construct. `\l` specifies the number of WaveformChars to apply in the assignment statement. The most common use is to truncate hex-encoded WaveformChar sequences, or specify lengths for decimal-encoded sequences. The current Alignment property is applied to determine which WaveformChars to truncate. For example, “`\l6 \hAB FC`”, applied in the context of a default Alignment property (MSB), results in the WaveformChar string “BBBBBB”.

[Reference 12]

Clause 21.5: Pattern Labels

Modify the sentence in the second paragraph of this section:

Labels shall be unique per **Pattern** or **Procedure** block.

To:

Labels shall be unique per **Pattern**, or per individual procedure block inside a **Procedures** block.

[Reference 10]

Clause 23.2: Pattern Initialization

Add as the first sentence in this section (in the first paragraph):

A WaveformTable statement shall be present before the first Vector or Condition statement in a Pattern block.

Add this paragraph at the end of this section:

The current state of all Signals after the first Vector statement is a function of the execution sequence of Vectors. The last defined WaveformChar (or WaveformChars if multiple values are required) and the WaveformTable interpretation for each Signal shall remain in effect for each Signal for each Vector until changed by a new assignment or new WaveformTable statement. The last assigned WaveformChar and current WaveformTable remains in effect through all block statements (and on return from a block, for example on completion of a Loop) except on return from Procedure calls as defined in clause 24.

[Reference 16]

Clause 24.2: Procedures example

Add to the first V statement in the example here:

```
V { sigs1=11010; sigx = 1; }
```

All signals used in a procedure shall be defined in the first V statement of that procedure.

[Reference 15]

Clause 24.4: Scan testing

Eliminate the sentence in the second paragraph of this section:

Only one Shift statement is defined in a scan-oriented block.

This statement is in conflict with the syntax presentation in the section below this paragraph, and with the statement “A particular sigref_expr shall appear only once in a Shift block. Consecutive shift blocks require unique sigref_exprs for each Shift statement.” on page 114 of the Standard.

[Reference 11]

Clause 24.5: Procedure and Macro Data substitution

Add to the last sentence in the last bullet on page 113, (clause starts with “ — If data is not defined for a signal in a Procedure or Macro invocation, and the Procedure/Macro body specifies # or % on that signal in any vector, then the last defined WaveformChar that precedes the first # or % is used.”):

The last defined WaveformChar is found in the previous V or C statement in the Macro or Procedure body, **or for a Macro if an initial WaveformChar is not present in the Macro body, then the incoming WaveformChar on that signal will be used.**

This additional clause is added to reflect the Macro behavior of maintaining the last defined WFC value on all signals when a Macro is invoked.

[Reference 21]

Add this additional clause to the restrictions on resolving parameters:

— Once a *sigref_expr* in a procedure or macro call has been name-matched to a ‘#’ or ‘%’ reference in the procedure or macro body, only subsequent name-matched references to the same *sigref_expr* shall access the argument values for that *sigref_expr*. Therefore, once a *sigref_expr* has a matching *sigref_expr* containing ‘#’ or ‘%’, any remaining component-

matching requires additional arguments to specify values for these remaining component-matching references and shall not consume arguments from a name-matched *sigref_expr*:
[Reference 20]

Recommendations

The information below is additional to the standard and is not part of this standard. It would appear in a separate recommended usage document, and is provided here to assist in application of this standard.

STIL defines an arrayed name as an independent construct. This means that the name “a” and the name “a”[0..15] are two individual sets of signals. It is recommended that users avoid declaring both names to avoid confusion in STIL files.

[Reference 1]

STIL supports a variety of SI units as listed in Table 3, but defines only time_expr constructs where these units may be employed. The engineering prefix (as listed in Table 4) is always applied as part of the expression when present, but there are no checks made on expressions as to appropriate SI units and relationships. It is recommended that all expressions make use of appropriate SI units for the type of expression; in this case, all SI units evaluate to units of time for time_expr constructs or to “unitless” values.

[Reference 14]

Table 12 contains the list of unresolved timing events. In most cases, these unresolved events are only used as interim states that are to be resolved prior to moving a pattern set to a tester. In practice, the 'N' event (ForceUnknown) is often left as an unresolved state. The interpretation of the 'N' state is that it may freely be replaced by either a U or a D event with no consequence to the operation or validity of the patterns. The reason for leaving it unresolved is that in some applications optimizing is possible by choosing one state over the other.

[Reference 18]

STIL allows Procedure and Macro bodies to contain a “#” or “%” on signals in the very first Vector of that Procedure or Macro. When this occurs, an explicit WaveformChar value to be used for either padding (if the V {} is inside a Shift) or as the default WaveformChar value if no values are assigned to this Signal when called, has not been defined. For Macros, the last WaveformChar assigned to that signal is propagated into the next call and will serve as the default WaveformChar value, but for Procedures no default value is specified. It is **very strongly recommended** that all Procedure and Macro bodies that contain “#” or “%” references, also contain an initial V {} or C {} statement to assign an explicit WaveformChar value to all signals that have a “#” or “%” reference, before the first occurrence of the “#” or “%”, if the Macro or Procedure is expected to be called with a partial set of argument values or require padding of scan data.

[Reference 21]

References Trace

These clarifications were agreed to (and part of the minutes) of the following Working Group meetings:

[Reference 1] resolved WG meeting 0008 (2/21-22/2000 Chandler, AZ)

[Reference 2] resolved WG meeting 0008 (2/21-22/2000 Chandler, AZ)

[Reference 3] resolved WG meeting 0018 (5/4-5/2000 VTS)

- [Reference 4] resolved WG meeting 0008 (2/21-22/2000 Chandler, AZ)
- [Reference 5] resolved WG meeting 0008 (2/21-22/2000 Chandler, AZ)
- [Reference 6] resolved WG meeting 0018 (5/4-5/2000 VTS)
- [Reference 7] resolved WG meeting 0018 (5/4-5/2000 VTS)
- [Reference 8] these terms were changed from 'Z' to 'Off' in an early Working Group meeting, and Table 1 was not properly updated.
- [Reference 9] resolved WG 0209 (3/14/02). clarifications on name applications.
- [Reference 10] resolved WG 0209 (3/14/02). details in proc_label_clarification_justify.txt
- [Reference 11] resolved WG 0209 (3/14/02). details above.
- [Reference 12] resolved WG 0209 (3/14/02). WFC issue from Doug S.
- [Reference 14] resolved WG 0209 (3/14/02). Raised during p1450.1 expression discussions
- [Reference 15] resolved WG 0209 (3/14/02). Oversight in spec.
- [Reference 16] resolved WG 0209 (3/14/02). Clarification requested by Mike Purtell.
- [Reference 17] To be resolved in WG. Clarification requested by Doug S.
- [Reference 18] resolved WG 0211 (3/14/02). Clarification requested by Doug S., 2/28/02 minutes.
- [Reference 19] To be resolved in WG. Clarification requested by Greg M after discussions about the meaning of the 'X' event which assumed unexpected drive-characteristics behavior.
- [Reference 20] To be resolved in WG. Clarification requested by Greg M after detailed parameter passing discussions.
- [Reference 21] To be resolved in WG. Clarification requested by John C/Doug S; recommendation worded from discussions with Tony T.
- [Reference 22] missing t and v in Table 1 added from review by Gordon R.