

Title: STIL 1450.3 Internal Review of 1450.3-D08

History:

- 4/11/03 - Issues from 4/11/03 working group meeting
- 6/6/03 - Added issues from review by Gordon Robinson, plus comment from 6/6 working group meeting.
- 8/15/03 - Updated from wg meeting

Significant issues and changes are listed in the table below. All issues not listed are editorial in nature and appropriate changes have been made in D08. Standard black font is used for resolutions that have been fully addressed and documented in the D07 document. Red italic font is used for resolutions that are still under discussion or have not been updated in D08.

Table 1: Summary of issues resolved in 1450.3-D08

Ident	Issue	Resolution
DD-1	4/17/03 - Need way to identify X3-mode. i.e., where vectors are consumed in groups of three	4/17/03 - Added a new statement to PatternCharacteristics and Loop Characteristics: VectorModulus. 8/15 - Moved VectorModulus from the LoopCharacteristics block to the InstructionCharacteristics, so that it can apply to all specified instructions. Note that there is also one at the PatternCharacteristics to handle straight vectors. <i>Needs review by Dave Dowding/Agilent (i.e. requestor of this feature)</i>
GR-1	What is the Name? The title (and presumably the PAR) uses the phrase Tester Target Specification, yet the defined syntax uses the acronym TRC with alternative meanings.	6/6/03 - The title in D08 is the same as the PAR - “Standard for Extensions to Standard Test Interface Language (STIL) (IEEE Std. 1450-1999) for Tester Target Specification” Action - none required.
GR-2	Inter-Standard Dependencies With the amount that P1450.3 depends on things in P1450.1, I do not believe it can go to ballot before P1450.1	6/6/03 - The 1450.3 working group agrees with this statement. In fact, clause 4.3 specifically lists the dependencies upon 1450.1. Action - 1450.3 to be balloted with or after 1450.1.

Table 1: Summary of issues resolved in 1450.3-D08

Ident	Issue	Resolution
GR-3	<p>Subset Language Permissions</p> <p>Several places allow the existence of incomplete STIL implementations in the target to be stated. This seems the wrong way to do it. If we permit such subsets, shouldn't we use the mechanism of identifying sets of features in the STIL statement. Maybe STIL is too large to start with and should be subsetted and then have pieces put back.</p> <p>And what does "fully STIL compliant" mean for Patterns? Does the concept apply to Timing and DCLevels as well?</p>	<p>6/6/03 - The only sub-sets that are defined in the STIL series is at the 1450.n level. Per the standard, if "TRC" is called out in the initial STIL block then all of the 1450.3 standard must be taken into account.</p> <p>It is possible that a tool (or tools) may implement a subset, and as long as the producer and consumer agree on the subset, they can interoperate. However, such subsets are not defined in any way by the standard.</p> <p>Action - none required.</p>
GR-4	<p>Files and STIL Data Collections</p> <p>I believe that there is a poorly understood emerging set of ideas about how blocks of STIL data should be arranged in individual files. The base STIL document doesn't really address this, except for permitting Include statements and leaving the rest to imagination.</p> <p>It is never clear in P1450.3 (and other documents) whether the term "STIL file" refers to a true operating system level file or to the full collection of data that may have come from many files.</p> <p>Incidentally, I'd place this issue as more important than many of the other activities being worked on.</p>	<p>6/6/03 - Gordon is correct. The working group has been sloppy in the use of the term "STIL file". STIL data may actually be comprised of a collection of files. WG decided that "STIL stream" would be a better term to apply to either a single "STIL file" or a collection of "STIL files".</p> <p>9/2/03 - The document has been revised to use the term "STIL file/stream" wherever either is allowed. It is also so defined in the glossary in Annex A.</p>
GR-5	<p>Named and Unnamed Blocks</p> <p>I've long disliked STIL's use of unnamed blocks. It complicates conventions about the use of multiple files. In P1450.3 we see these being used with a new vicious twist, the fact that they can now occur within more limited scopes, so we can have multiple unnamed blocks in different scopes.</p> <p>An example of the nastiness of these occurred when P1450.2 was standardized (and I missed it) because the meaning of a PatternExec with no DCLevels block (i.e. all of them up till then) changed once an unnamed DCLevels block was visible.</p>	<p>6/6/03 - The "unnamed block" convention was established in 1450-1999. The premise being that it should be simple to do simple things. Named blocks are only **required** when one needs to identify multiple domains. WG does not see where 1450.3 has added complexity to this concept.</p> <p><i>Action - Gordon, please provide specific example(s) of concern. (AI - gordon robinson)</i></p>

Table 1: Summary of issues resolved in 1450.3-D08

Ident	Issue	Resolution
GR-6	<p>Expression Complexities</p> <p>I've always hated the 'ticks' around STIL expressions, believing them to be unnecessary (or at best an artifact of sacrificing the user and readability to poor parser technology).</p> <p>I was delighted to see examples of expressions without the ticks in examples, and then found the extensions allowing that for some situations in P1450.1.</p> <p>I fear that the various developments are causing expressions to grow to the point of collapse. (More a P1450.1 and P1450.6 issue, but I can't tell which definition P1450.3 uses!)</p>	<p>6/6/03 - Some simplification on the use of 'ticks' has been done in 1450.1. Specifically, single tokens don't need to be quoted except where specifically required (e.g., in a Waveform block). Also, 'ticks' are not required on the right hand side of vector data.</p> <p>Action:</p> <p><i>Gordon, are there any other specific cases that you can identify for simplification? (AI - gordon robinson).</i></p>
GR-7	<p>"Dot n" Terminology</p> <p>I and many others associate "dot 1" with IEEE Std 1149.1, and have done since 1988. I don't mind the "STIL.1" term, but do object to "dot 1".</p>	<p>6/6/03 - The wg adheres to the STIL.1, STIL.3 nomenclature within the standard. Reference to "dot1" is used mostly in verbal and email communications and, given human nature, is probably not likely to change. And, really has no effect on the standard.</p> <p>Action: none required</p>
GR-8	<p>More New Keywords</p> <p>Most language standards teams are very reluctant to add new reserved words, the STIL world seems determined to have the largest collection of them in computer science history. Every newly reserved word may invalidate existing STIL data, and this draft has 88 new ones (assuming I can count). Other drafts are even worse.</p> <p>Particularly the new keyword All has a significant chance of being used in existing data.</p> <p>This addiction to keywords is a reason why so much of STIL is written using "quoted" names for all identifiers, which gets extremely ugly when "'ticked and quoted'" in expressions.</p>	<p>6/6/03 - wg reviewed this issue one more time. The precedent was actually established in STIL.0. wg concluded:</p> <ol style="list-style-type: none"> 1. The syntax definitions allow a parser to unambiguously differentiate between keywords, and user defined names (like signals). 2. However, reserving keywords allows for better error reporting and recovery (if a parser chooses to). 3. User keywords can always be clearly differentiated by use of double quotes (recommended good practice) or by beginning with a non-capital. 4. Attributes are not defined as reserved. 5. Keywords inside Environment blocks (i.e., TRC) are not part of the reserved word list. This needs to be documented as such. <p>9/2/03 - Per the definition of Environment in 1450.1, all keywords that are used inside an Environment block are NOT reserved. Therefore, the table defining all the new keywords for TRC has been moved into the TRC block definition.</p>

Table 1: Summary of issues resolved in 1450.3-D08

Ident	Issue	Resolution
GR-9	<p>New Undefined Concepts Several concepts are used without definition (Format, Shape, TimeSet, indirect memory) appealing to intuition and domain knowledge for their meaning. I've seen enough different uses of these terms to fear that they won't be understood consistently. This problem is deep and widespread.</p>	<p>6/6 - A 5 page glossary has been created (annex A) to address this issue. That said, there is still room for improvement in the syntax definitions. <i>Action: Add clarification to definitions in clause 15.1 (AI-ony)</i></p>
GR-10	<p>Section 5.1 Example The details in the example seem different from the values used in the introductory paragraph. E.g. MaxSignals 160 when 256 channels was stated.</p>	<p>6/6 - Example needs updating, <i>Action: Fix clause 5.1 (AI-ony)</i></p>
GR-11	<p>Pragmas and Tester Specifics I'm uncomfortable with having this amount of "foreign" syntax allowed. And if the tester ever uses *} the lexical style for delimiting it breaks down. I don't like the <V1 V2 V3> style of identifying the "per target" items, because adding a new target means altering all of them, not just specifying extra items where needed. Would a new statement "Resource target Value;" be more clear and easier.</p>	<p>6/6 - wg responses: 1) It is felt that {**} is obscure enough not to conflict with most applications. If conflict, then the code should not be included in-line, but provided as a side file. 2) wg actually started with a Resource statement but moved to the <> tag notation so the the tags were clearly not part of the STIL syntax and could be embedded into any STIL statement without conflict. 3) The concept of re-processing a file to add targetting information for one or more ATE systems was a goal of this standard (see fig 1). It is felt that the ability to embed targeting information is a desirable feature. Action: none required</p>
GR-12	<p>Implicit Resource Allocation 5.4.3 appears to imply that the meaning of existing STIL that uses InheritWaveform has now changed to mandate a resource allocation semantics.</p>	<p>6/6 - This clause is "informative" only. It is included in the tutorial section to illustrate the difference between implicit resource allocation (implied by STIL statement usage), vs. explicit resource alloction (using resource tags). The actual resource interpretation and assignment is up to the STIL reader. Action: none required</p>

Table 1: Summary of issues resolved in 1450.3-D08

Ident	Issue	Resolution
GR-13	<p>Section 7.4 Unclear to me how the Pattern handling is supposed to know that ATE1 is to be used, and how it finds that MAP1 etc are the valid tags (remember the pragma has arbitrary content and so a general mechanism cannot understand any of its content.</p>	<p>6/6 - To date, we have not provided any connection between resource tags and the pragma block. They are both designed to fit unobtrusively inside the STIL syntax. It is up to the tool to make the connection. One likely connection is to embed the pragma in the block where it is needed. wg feels that this is enough, and does not want to dictate how a tool should utilize this capability. Note: Pragma block has been moved to STIL.1. Note2: CTL applications have already found this feature most useful and are building tools that utilize pragmas.</p> <p>Action: none required</p>
GR-14	<p>Section 8.1 I've never seen a description of what is allowed for an extension version. Different extensions have used some wildly different conventions.</p>	<p>6/6 - The convention that most of the STIL wg's are following is to use the Dnn as the version number while the standard is under development and utilize the year that the standard is approved after ballot approval. Unfortunately we assigned "STIL 1.0;" for STIL.0 rather than "STIL 1999;" - something we intend to fix at the next revision.</p> <p>Action: none required until standard is approved by IEEE.</p>
GR-15	<p>TesterChannelMap I believe that these maps are important enough to deserve becoming a top-level STIL block, and not an interpretation of syntax buried in an Environment somewhere.</p>	<p>6/6 - wg strongly feels that the channel map belongs in the Environment block. This is where information about how STIL data is to connect into some other environment (i.e., ATE environment, EDA environment, other) is expected to exist. It is possible to completely remove all Environment blocks and still make sense of the STIL information. You just won't be able to map it into said environment.</p> <p>Note: STIL is not a specific tester language, but a pattern language that has the ability to include mapping to one or more tester (or other) environments.</p> <p>Action: none required</p>

Table 1: Summary of issues resolved in 1450.3-D08

Ident	Issue	Resolution
GR-16	<p>TimeLimits Why not use Min expression; Max expression instead of positional referencing? Some of these are using unticked-off expressions like 2ns which I don't think are allowed (but should be because the ticks should never be mandated). This is also stated to be a boolean expression, but 2ns doesn't sound Boolean to me.</p>	<p>6/6 - 1) wg started with Min/Max keywords. These made it difficult to specify "less than" vs "less or equal". We decided to use the concept of asserts on each of the timing edges. 2) Single tokens like 2ns or t1 are (now) allowed without ticks. Complex expressions like '2ns+t1' require the ticks. 3) You are correct that 2ns is not a boolean. Please see how we have defined asserts for timing checks in the latest D08.</p> <p>Action: Please see latest draft -D08 (AI-gordon)</p>
GR-17	<p>CompareEvents and DriveEvents Can the "long names" (like ForceUp) be used or just the single character variants?</p>	<p>6/6 - Standard allows (and hopefully states) that the long names and short names can be used interchangeably. This is a rule in STIL.0</p> <p><i>Action: Review STIL.0 definition and review STIL.1, STIL.3 for consistency (AI-tony)</i></p>
GR-18	<p>Accuracy Gives a definition for accuracy that's not enforceable (because start of period isn't a measurable event. What definition of EdgeToEdgeAccuracy should be used?</p>	<p>6/6 - 1) This is definitely a case where the 80/20 rule applies. STIL, no doubt will not be able to specify ATE accuracy sufficient to please all (especially ATE manufactures). 2) T0 does not need to be measured. "Edge" is intended to specify how close an edge is to a specified value, whereas "EdgeToEdge" is to specify how close an edge is to another edge.</p> <p><i>Action: Gordon, can you assist in creating a definition that will come closer to satisfying ATE manufactures? (AI-gordon)</i></p>
GR-19	<p>MaxData MaxIO MaxMask etc I was totally confused by these. The only thing I could imagine is that they are an attempt to define FICM-like architecture details.</p>	<p>6/6 - Yes, this concept is very close to FICM. wg does not want to use the term FICM as it is not commonly understood. However, we will attempt to improve the explanation in 15.1 (see also GR-9).</p> <p><i>Action: improve syntax definitions in 15.1 (AI-tony)</i></p>

Table 1: Summary of issues resolved in 1450.3-D08

Ident	Issue	Resolution
GR-20	<p>Edge Retrigger</p> <p>These rules are often exceedingly complex (e.g. the edge retrigger could be from a U to a following U in systems with separate edges for each role (e.g SC212 and ValStar), or they could be for when an internally allocated edge generator is next used, which could be several events later (I've worked on defining one of these). The resource that has the constraint is a hidden entity in the STIL.</p>	<p>6/6 - Again, wg thinks the 80/20 rule applies. Expectation is that the statement as is provides value. Don't think we will ever fully satisfy all ATE requirements.</p> <p><i>Action: Gordon, can you suggest some improvements here? (Action-gordon)</i></p>
GR-21	<p>ReplicateSubWaveform</p> <p>I didn't understand what might be repeated.</p>	<p>6/6 - This statement ties directly to the SubWaveform as defined in STIL.0. The sub-waveform was put in to support architectures like Agilent, or NP-Test sequencer-per-pin. It also support high speed clocks that repeat many times in a cycle.</p> <p><i>Action: Expand the explanation and include reference to the STIL.0 statement (AI-tony)</i></p>
GR-22	<p>Section 15.2</p> <p>More real-valued ticked-off expressions. Like to see them.</p>	<p>6/6 - comment only</p>
GR-23	<p>Section 16.2</p> <p>Within Shape, what event does the last (complicated) expression refer to?</p>	<p>6/6 - This example is incomplete. There should be an event definition after the last time expression.</p> <p><i>Action: fix the example (AI-tony)</i></p>
GR-24	<p>NameChecks</p> <p>Mention of CharacterContent made my brain yell "Unicode".</p>	<p>6/6 - Not sure what is being suggested here.</p> <p><i>Action: Gordon, please explain further (AI-gordon)</i></p>
GR-25	<p>Section 19.1</p> <p>The use of multiple Category names seems different from that in 1450, where the contents of each Category "merged" are supposed to be disjoint. Here it is specifying alternatives.</p>	<p>6/6 - wg doesn't see any conflict with STIL.0 usage of categories. In STIL.0 categories apcify a range of alternate values for defining some parameter. In STIL.3 categories specify alternate values for defining a set of TRC's.</p> <p><i>Action: Gordon, please explain concern (AI-gordon)</i></p>

Table 1: Summary of issues resolved in 1450.3-D08

Ident	Issue	Resolution
TT-1	4/17/03 - Suggest changing LoopCharacteristics so that it contains the statements in a local block, instead of by reference.	4/17 - Changes made in D08 <i>Needs review by wg.</i>
TT-2	4/17/03 - Suggest changing TimeLimits to use the assert syntax to be consistent with changes made for WaveformDescription timing checks	4/17 - Changes made in D08 <i>Needs review by wg.</i>
TT-3	4/11/03 - Suggest adding new statement to SignalCharacteristics block: UndrivenInOut < Yes No >;	4/11 - Accepted by wg