Title: STIL 1450.1 Internal Resolution of 1450.1-D16

History:

- 3/25/03 - issues carried over from D15. Marked with <D15>.
- 3/27/03 - some new issues
- 4/24/03 - updates from wg meeting
- 5/8/03 - updates from wg meeting
- 9/18/03 - some new issues added
- 9/25/03 - updates from wg meeting

Comments by:

CTL - STIL.6 i.e., CTL working group
DM - denis martin, synopsys
DO - don organ, Inovus
GR - gordon robinson, ieee
PR - paul reuter, mentor
RK - rohit kapur, ctl chair person

Table 1: Summary of Issues with Draft P1450.1-D16

Ident	Issue	Resolution
CTL-1	The 1450.6 wg has raised concern over the definition of Lockstep. They are working on some kind of an alternate proposal.	4/10/03 - Paul Reuter has provided an alternate proposal. See minutes of 4/10 meeting for initial comments. WG is in process of reviewing and re-working the proposal to fit into 1450.1. 6/19/03 - WG decided to keep the “LockStep” keyword in the language, but only define at a high level what this option does. The details of the implementation are left to the tool or to other standards that utilize this feature.

Table 1: Summary of Issues with Draft P1450.1-D16

Ident	Issue	Resolution

DM-1	Clause 10 - Variables - Should allow values defined in the IntegerEnum and the SignalVariableEnum to be used as constants.
	Variables {
	IntegerConstant K=2;
	IntegerEnum ENUM_INT {
	Values { RED 33; GREEN 44; BLUE 55; }
	}
	SignalVariableEnum ENUM_SIG_4 {
	Values { FEE 1111; FII 0000; FOO 0101; }
	}
	SignalVariableEnum ENUM_SIG_6 {
	Values { FUM 110011; }
	}
	}
	Pattern P {
	C { I='RED+K'; G4='FOO'; G6='FUM'; }
	}
		4/24 - WG decision to to define WFCConstants (as the SignalVariable-enum equivalent) and to remove enums from the language, replacing them with IntegerConstant and WFCConstant types.

DM-2	Clause 6.3 - Integer expression - Should differentiate between expressions that contain variables and one containing numbers and constants. Integer expressions with constants are fully resolved at parse time.
	Variables { IntegerConstant K=19; }
	Signals {
	XX[1..K] In;
	YY[0..'K-1'] Out;
	}
		9/25/03 - fully explained in clause 6

DM-3	Conditional expressions - general concern that the usage of this capability has not been fully evaluated. Would like to work through a real application to see if any issues arise.	9/25/03 - several syntax changes have resulted from this discussion, and clause 6 has been completely re-written.

Table 1: Summary of Issues with Draft P1450.1-D16

Ident	Issue	Resolution

DM-4	When does the value of a variable get changed, and when is it visible across other patterns in its scope. Consider:
	Pattern P1 {
	C { VAR = 1; }
	V { SIG = X; }
	}
	Pattern P2 {
	If 'VAR == 1' { V SIG2 = X; } // test updated value from P1
	}
		9/18/03-tt Added new sub clause 10.4 to define variable synchronizing.

DM-5	ScanStructures -> ScanCells allows for the definition of complex cell via the use of If statements. The problem is that this definition has to be repeated for each cell name. Since designs tend to use the same cell type many times, a mechanism is needed to define it once and reference it on the cell names.	9/25/03 - in clause 15, each scan-cell name in the new (braced) ScanCells block now has an optional cell-type definition.
DO-1	A suggestion from Don Organ as part of the TRC review is to add a definition of the integer size. Dot0 currently defines integer size to be 32 bits. However, some tester memories are approaching 4GB.	4/24 - It is the understanding of this Working Group that the issue of supporting values that exceed the minimum required by the dot-zero standard, is a function of the readers and run-time environment, which must flag overflow issues ANYWAYS, and that writers (which may also rely on integer sizes > 32 bits to generate their data) won't change what the readers and run-time environments must support by adding this statement. This proposal was vetoed in this Working Group. See 4/24 minutes for more discussion.
GR-1 <D15>	Fail data is a different concept from Pattern data, and should not use the identical syntactic form. A new keyword expresses the fact that this is different information. Annex N The "tag" mechanism is fundamentally ambiguous when "vector splitting" occurs. Earlier comments show how it fails to adequately specify many contexts. In general I want to see clear syntactic entries identifying information as fail data.	*9/23/03 - New clause 22 - PatternFailReport has been added to address this. There are still a few details being worked out.*

Table 1: Summary of Issues with Draft P1450.1-D16

Ident	Issue	Resolution
RK-1	4/24 - Rohit raised an issue about retaining the value of variables between PatternBursts. Currently, expectations are that an InitialValue is re-asserted each time a variable (with an initial value) comes into existence. This would prohibit values from persisting through non-contiguous references to the domain. A proposal to define an additional attribute or a modifier attribute to the InitialValue behavior was considered to address this issue.	9/25/03 - new attribute to integer variables - HoldValue;
WG-1	9/2/03 - 1450.3 had adopted the convention of “STIL file/stream”, rather than just “STIL file”. This convention should be adopted in 1450.1 also.	*9/11/03 - change to doc needs tbd - AI-tony*