22. PatternFailReport

The PatternFailReport block is used to contain fail information resulting from the test of a device. As such, it is typically not part of a STIL test program stream, but contains references to the associated STIL test program stream to establish the context. One PatternFailReport block shall contain the set of fails from the test of one device on the tester.

22.1 Pattern Extensions

The extensions to the Pattern block syntax will be moved to clause 19. They are here temporarily for review along with the rest of the pattern-fail-report syntax

```
pattern statements:
    ( X < IDENTIFIER | INTEGER > ; )*
}
```

X: The X statement (which is short for cross-reference) is used to identify key places in the pattern that are used for collecting data during the device test process and reporting that data to a post processing tool. The post processing tool, most likely, is closely related to the tool that created the STIL pattern in the first place. The X statement is optional and may be used in any pattern, macro or procedure. Refer to PatternFailReport clause for the definition on how these references are used.

The keyword is followed by either an identifier or an integer.

If the X statement is present in a Pattern block, it typically is used to indicate the beginning of a "pattern unit" which is the beginning of a sequence of vectors that make up a test as created by an ATPG tool (Automatic Test Pattern Generation); typically used in the context of scan and comprised of a scan-in, some activity to DUT pins, followed by a scan-out.

If the X statement is present in a Macro or Procedure block then the associated identifier or integer is appended to the identifier from the calling pattern according to rules as defined in the PatternFailReport clause.

22.2 PatternFailReport syntax

```
PatternFailReport {                                                                                      (1)
    Pattern PAT_NAME;                                                                                    (2)
    PatternBurst PAT_BURST_NAME;                                                                         (3)
    PatternExec PAT_EXEC_NAME;                                                                           (4)
    FailData {                                                                                           (5)
      ( <IDENTIFIER|INTEGER> ( : <IDENTIFIER|INTEGER> )*  SIG_NAME   ( CYCLE_OFFSET (< L | H | T >) )* ; )*  (6)
    })*   // end FailData
}  // end PatternFailReport
```

(1) **PatternFailReport**: The block contains fail data as produced on a tester and is associated with a STIL test program stream.

(2) **Pattern**: This statement specifies the name of the pattern in the associated STIL test program stream that is detecting the device failures.

(3) **PatternBurst**: This statement specifies the name of the pattern burst in the associated STIL test program stream that is detecting the device failures.

(4) **PatternExec**: This statement specifies the name of the pattern exec in the associated STIL test program stream that is detecting the device failures.

(5) **FailData**: This begins the block containing the device failure data.

(6) **fail data record**: Each device failure record contains the fail data information as descibed below.

a) IDENTIFIER|INTEGER: The first token of each record is used to identify the reference position within the pattern. It may be either an identifier or an integer. Reference shall always be to the most recent X reference statement within the pattern. If there are X references within called procedures and/or macros then they are appended with colon separators to the X reference in the base pattern.

b) SIG_NAME: This is the name of the output signal that detected the failure. It may be either a name in a Signals block or a re-named single signal in a SignalGroups block. The signal name resolution follows the standard rules for domain name resolution.

c) CYCLE_OFFSET: This token indicates the cycle offset from the last X reference. This field is optional for a primary output signal, and is required for a scan signal in a Shift block. For a scan signal it is used to compute an index into the serial scan data that is passed from a pattern into a macro or procedure

d) < **L | H | T** >: This field is optional and specifies the observed state of the failing signal - either low, high, or tri-state.

22.3 PatternFailReport examples

Example 1: various record formats

```
PatternFailReport {
    Pattern PAT;
    PatternBurst BRST;
    PatternExec EXE;
    FailData {
      13 PO1; // example with pattern-unit=13; signal=PO1
      PU14 PO1; // example with a pattern-unit identifier
      42 SO1 19; // example with cycle offset (i.e. scan cell if on a shift)
      45:LU SO1 16; // example with reference inside a macro/proc (i.e the LoadUnload)
      46:M1:LU S01 16; // example of two levels of macro/proc
      59 PO2 23H; // example of fail state
      64 SO2 6 16 24 25 338; // example of multiple fails (5 in this case)
      72 SO3 13H 19L 45L 63L; // example of multiple fails with fail states
      PAT PO1 19023H; // example referencing cycle 19023 from the beginning of the pattern
      } // end FailData
} // end PatternFailReport
```