Introductory STIL BNF


The following BNF representation of STIL is available here to provide an introduction to STIL. It is not a complete representation, but rather intended as an overview to the primary structural elements of the language. In the interests of simplification, this BNF does not represent ordering requirements, or identify multiplicity issues on statements. Please be aware when referencing this BNF that it is also incomplete with respect to the detailed semantics of the language. For a complete representation please review the P1450 document.

The meta-symbols of BNF are:

 ::= meaning "is defined as"
  | meaning "or"
  optional items are enclosed in meta symbols "[" and "]"
  terminals are distinguished by using bold face type
  terminals of only one character are surrounded by quotes (") to distinguish them from meta-symbols

This BNF is a representation of the allowed syntax of the language. It will be revised as different ways to "look at" this information are considered. For a complete definition of STIL please refer to the P1450 document.

Please remember this BNF is considered an incomplete representation of the language.

1.0 STIL Organization

stil_session ::= stil [header] session

session ::=  block
  | session block

block ::=  user_keywords
  | user_functions
  | signals
  | signal_groups
  | pattern_exec
  | pattern_burst
  | timing
  | spec
  | selector
  | scan_structs
  | pattern
  | procedures
  | macroDefs
  | include | annotation | udb | (null)
2.0 STIL Statement

stil ::= STIL stil_version_number ";"

stil_version_number ::= integer "." integer

3.0 Header Block

header ::= Header "{" [header_list] "}" 

header_list ::= header_item 

| header_list header_item

header_item ::= Title string ";"

| Date date_string ";"

| Source string ";"

| History "{" [history_list] "}" 

| include | annotation | udb | (null)

date_string ::= """weekday month day_of_month time year ""

weekday ::= Mon | Tue | Wed | Thu | Fri | Sat | Sun

month ::= Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec

day_of_month ::= digit digit

time ::= hour ";" minute ";" second

hour ::= digit digit

minute ::= digit digit

second ::= digit digit

year ::= digit digit digit digit

history_list ::= annotation

| history_list annotation

4.0 Include Statement

include ::= Include file_name [ IfNeed blocktype ] ";"

[ header]

(Note: The STIL and optional header statements are the first statements in an included file. All subsequent statements are in the context of the "Include" statement in the including file)

blocktype ::= Include 

| Header 

| UserKeywords 

| UserFunctions 

| Signals 

| SignalGroups 

| PatternExec
5.0 **UserKeywords Statement**

user_keywords ::= **UserKeywords** user_defined_keywords ";

user_defined_keywords ::= identifier
| user_defined_keywords identifier

udb ::= identifier "{"udb_text "}"  
| identifier udb_2_text ";" 

(Note: allowed identifiers must first be declared in
| the **user_keywords** stmt)

udb_text := any sequence of characters, with the restriction that any '}' be matched with '}'

udb_2_text := any sequence of characters except '{' and '}' and ';

6.0 **UserFunctions Statement**

user_functions ::= **UserFunctions** user_defined_function ";

user_defined_function ::= identifier
| user_defined_function identifier

7.0 **Ann Statement**

annotation ::= **Ann** 
| \*\* ann_text \*\*

ann_text := any sequence of characters except \*\*

8.0 **Signals Block**

signals ::= **Signals** 
| \* \* signals_list \* \*

signals_list ::= signals_item
| signals_list signals_item

signals_item ::= signal_name_array_opt signal_type ";
| signal_name_array_opt signal_type 
| include | annotation | udb | (null)

signal_name_array_opt ::= signal_name
| identifier 
| integer 
| integer ]

null
signal_name ::= identifier
            | identifier "[" integer "]"

signal_type ::= In
            | Out
            | InOut
            | Supply
            | Pseudo

sig_statements ::= sig_statement
            | sig_statements sig_statement

sig_statement := terminations
                | default_state_stmt
                | ScanIn [ integer ] ";;"
                | ScanOut [ integer ] ";;"
                | Base base_type ";;"
                | Alignment orient_type ";;"
                | DataBitCount integer ";;"
                | include | annotation | udb | (null)

terminations ::= Termination termination_state ";;"

termination_state ::= TerminateHigh
                  | TerminateLow
                  | TerminateOff
                  | TerminateUnknown

default_state_stmt ::= DefaultState default_state ";;"

default_state ::= "U" | ForceUp
                | "D" | ForceDown
                | "Z" | ForceOff

base_type ::= Hex wfcs
            | Dec wfcs

orient_type ::= LSB
            | MSB

9.0 SignalGroups Block

signal_groups ::= SignalGroups [domain_name] "{" [groups_list ] "}"
group_name ::= \texttt{identifier}

sigref_expr ::= \texttt{signal_or_group_name}
| \texttt{grp_name_exp_list}

grp_name_exp_list ::= \texttt{signal_or_group_name}
| \texttt{grp_name_exp_list}\texttt{plus_or_minus grp_name_exp_list}

signal_or_group_name ::= \texttt{signal_name_array_opt}(Note \texttt{signal_name} and \texttt{group_name} may be \texttt{identifiers}; they are both here to indicate either ref.)

plus_or_minus ::= \texttt{"+"} | \texttt{"-"}

\section*{10.0 PatternExec Block}

pattern_exec ::= \texttt{PatternExec [pat_exec_name]\{ [ pat_exec_list_items ] \}}

pat_exec_name ::= \texttt{identifier}

pat_exec_list_items ::= pat_exec_item
| pat_exec_list_items pat_exec_item

pat_exec_item ::= \texttt{Timing timing_name \}}
| \texttt{PatternBurst pat_burst_name \}}
| \texttt{Category category_name \}}
| \texttt{Selector selector_name \}}
| \texttt{include | annotation | udb | (null)}

category_name ::= \texttt{identifier}

selector_name ::= \texttt{identifier}

timing_name ::= \texttt{identifier}

pat_burst_name ::= \texttt{identifier}

\section*{11.0 Pattern Burst Block}

pattern_burst ::= \texttt{PatternBurst pat_burst_name\{ [ pat_burst_stmnts ] \}}

pat_burst_name ::= \texttt{identifier}

pat_burst_stmnts ::= pat_burst_stmnt
| pat_burst_stmnts pat_burst_stmnt

pat_burst_stmnt ::= \texttt{SignalGroups groups_domain \}}
| \texttt{MacroDefs scan_macros_domain \}}
| \texttt{Procedures procedures_domain \}}
| \texttt{ScanStructures scan_name \}}
12.0 Timing Block and WaveformTable Block

timing ::= Timing [timing_label] "{" [ timing_list ] "}"

Timing ::= SignalGroups groups_domain ";;"

| MacroDefs scan_macros_domain ";;"
| ScanStructures scan_name ";;"
| Start pat_label ";;"
| Stop pat_label ";;"
| Procedures procedures_domain ";;"
| Termination "{" [ termination_statements ] "}"

termination_statements ::= termination_statement

| include | annotation | udb | (null)

termination_statement ::= sigref_expr termination_state ";;"

groups_domain ::= identifier

scan_macros_domain ::= identifier

procedures_domain ::= identifier

pat_label ::= identifier

termination_statements ::= termination_statement

| include | annotation | udb | (null)

termination_statement ::= sigref_expr termination_state ";;"

wft ::= identifier

timing_label ::= identifier

cell ::= identifier

wft_list ::= wft_item
  | wft_list wft_item

wft_item ::= Period """" time_expr '"""" ";"
  | Waveforms """" wft_list """" ";"
  | InheritWaveformTable [timing_label "."]wft ";"
  | SubWaveforms """" subwaveforms_list """" ";"
    | include | annotation | udb | (null)

waveforms_list ::= waveforms_item
  | waveforms_list waveforms_item

waveforms_item ::= sigref_expr [label] """" waveform_items """" ";"

waveform_items ::= waveform_item
  | waveform_items waveform_item

waveform_item ::= InheritWaveform [timing_label "."]wft "."]cell ";"
  | wfc """" wfc_def_list """
  | wfc """" wfc_def_list """
    | include | annotation | udb | (null)

subwaveforms_list ::= subwaveforms_item
  | subwaveforms_list subwaveforms_item

subwaveforms_item ::= swf_label "":" Duration """" time_expr '"""" """" sub_def_list """" ";"
    | include | annotation | udb | (null)

swf_label ::= identifier

wfc_def_list ::= wfc_definition
  | wfc_def_list wfc_definition

wfcs_def_list ::= wfcs_definition
  | wfcs_def_list wfcs_definition

sub_def_list ::= sub_definition
  | sub_def_list sub_definition

wfc_definition ::= [label ":"] """" time_expr '"""" event ";"
  | [label ":"] """" time_expr '"""" ";"
  | [label ":"] event ";"
  | [label ":"] ["""" time_expr '"""" ] [\r integer] swf_label ";"
  | [label ":"] ["""" time_expr '"""" ] [\r integer] swf_label ["""" integer "] ";"
  | [label ":"] ["""" time_expr '"""" ] [\r integer] swf_label ["""" # ""] ";"
    | include | annotation | udb | (null)
13.0 Spec and Selector Block

spec ::= Spec [spec_name] 
    "{" [ spec_list ] "}"

spec_name ::= identifier

spec_list ::= spec_item
    | spec_list spec_item

spec_item ::= Category cat_name 
    "{" [ var_spec_info ] "}"
    | Variable var_name 
    "{" [ cat_spec_info ] "}"
    | include | annotation | udb | (null)

cat_name ::= identifier
var_name ::= identifier
var_spec_info ::= var_spec_info_item | var_spec_info var_spec_info_item
cat_spec_info ::= cat_spec_info_item | cat_spec_info cat_spec_info_item
var_spec_info_item ::= var_name "=" time_expr ";"
     | var_name "{" [Min time_expr ";";] [Typ time_expr ";";] [Max time_expr ";";] "$" }" 
     | include | annotation | udb | (null)
cat_spec_info_item ::= cat_name time_expr ";
     | cat_name "{" [Min time_expr ";";] [Typ time_expr ";";] [Max time_expr ";";] "$" }" 
     | include | annotation | udb | (null)
selector ::= Selector selector_name "{" [selector_list ] "}" selector_name ::= identifier
selector_item ::= var_name selector_type ";"
selector_list ::=selector_item | selector_list selector_item
selector_type ::= Min | Typ | Max | Meas

14.0 ScanStructures Block
scan_structs ::= ScanStructures scan_name "{" [scanchains ] "}" scanchains ::= scanchain | scanchains scanchain
scanchain ::= ScanChain chainname "{" [scan_struct_list ] "}" chainname ::= identifier
scan_struct_list ::= scan_struct_item | scan_struct_list scan_struct_item
scan_struct_item ::= ScanLength integer ";" | ScanOutLength integer ";" | ScanCells cellname_list ";" | ScanIn signal_name ";" | ScanOut signal_name ";" | ScanMasterClock signal_name ";" | ScanSlaveClock signal_name ";" | ScanInversion bit ";" | include | annotation | udb | (null)
15.0 Pattern Block

pattern_set ::= Pattern pattern_name "{" [ pattern_statements ] "}"  

pattern_name ::= identifier  

pattern_statements ::= pattern_stmt  
  | pattern_statements pattern_stmt  

pattern_stmt ::= label pat_stmt  
  | pat_stmt  

pat_stmt ::= waveform_table_stmt wft ";"  
  | Loop integer "{" [ pattern_statements ] "}"  
  | MatchLoop integer "{"pattern_statements BreakPoint "{"pattern_statements"}" "}"  
  | MatchLoop Infinite "{"pattern_statements BreakPoint "{"pattern_statements"}" "}"  
  | vector_stmt  
  | condition_stmt  
  | Call procedure_name ";"  
  | Call procedure_name "{" vec_data "}"  
  | Macro macro_name ";"  
  | Macro macro_name "{" vec_data "}"  
  | GoTo pat_label ";"  
  | Stop ";"  
  | ScanChain chain_name ";"  
  | BreakPoint ";"  
  | BreakPoint "{" pattern_statements "}"  
  | IddqTestPoint ";"  
  | TimeUnit "" time_def "" ";"  
  | include | annotation | udb | (null)  

waveform_table_stmt ::= "W" | WaveformTable  

label ::= identifier ";"  

non_cyclized_data ::= "@" time_value event_pair ";"  
  | "@" time_value "{" [ event_pair_list ] "}"  

event_pair ::= sigref_expr "=" event  
  | include | annotation | udb | (null)  

event_pair_list ::= event_pair  
  | event_pair_list ";" event_pair  

vector_stmt ::= "V" "{" vec_data "}"
| Vector "{" vec_data "}"

condition_stmt ::= "C" "{" vec_data "}"

| Condition "{" vec_data "}"

time_value ::= integer

time_def ::= decimal [engineering_units]

vec_data ::= vec_data_block

| vec_data vec_data_block

vec_data_block ::= sigref_expr "=" vec_data_string ";"

| sigref_expr "{" vec_data_strings "}"

| non_cyclized_data

| include | annotation | udb | (null)

vec_data_strings ::= vec_data_string ";"

| vec_data_strings vec_data_string ";"

| include | annotation vec_data_string ";"

| include | annotation | udb | (null)

vec_data_string ::= wfc_data_string

| hex_data_string

| dec_data_string

(Note: string type is runtime
dependent based on the sig_refs Base definition)

wfc_mode ::= \"w \" wfc_data_string

hex_mode ::= \"h \" hex_data_string

| \"h\"wfcs hex_data_string

dec_mode ::= \"d \" dec_data_string

| \"d\"wfcs dec_data_string

wfc_data_string ::= wfc_data_string wfc_data

| wfc_data

wfc_data ::= wfcs

| \"r\"integer wfcs

| hex_mode

| dec_mode

hex_data_string ::= hex_data_string hex_data

| hex_data

hex_data ::= hexchars

| \"r\"integer hexchars

| wfc_mode

| dec_mode

dec_data_string ::= dec_data_string dec_data

| dec_data

dec_data ::= integer

| \"r\"integer integer
16.0 Procedures Block
procedures ::= Procedures [procedure_domain_name] "{" [procedure_definitions] """
procedure_domain_name ::= identifier
procedure_definitions ::= procedure
| procedure_definitions procedure
procedure_name ::= identifier
procedure ::= procedure_name "{" [procedure_statements] ""
| include | annotation | udb | (null)
procedure_statements ::= procedure_or_macro_item
| procedure_statements procedure_or_macro_item
procedure_or_macro_item ::= Shift "{" pattern_statements ""
| pat_stmt

17.0 Macrodefs Block
macrodefs ::= MacroDefs [macro_domain_name] "{" [macro_definitions] ""
macro_domain_name ::= identifier
macro_definitions ::= macro
| macro_definitions macro
macro ::= macro_name "{" [macro_statements] ""
| include | annotation | udb | (null)
macro_name ::= identifier
macro_statements ::= procedure_or_macro_item
| macro_statements procedure_or_macro_item

18.0 Other Miscellaneous Statements
identifier ::= identifier_segment
| identifier "." identifier_segment
identifier_segment ::= simple_identifier
| escaped_identifier
simple_identifier ::= letter_or_underline simple_characters
simple_characters ::= simple_characters simple_character
| (null)
letter_or_underline ::= letter

(Note the maximum length of an identifier segment is 1024 characters)
simple_character ::= letter | digit | underline
letter ::= upper_case_letter | lower_case_letter
upper_case_letter ::= "A" | "B" | ... | "Z"
lower_case_letter ::= "a" | "b" | ... | "z"
underline ::= "_"
escaped_identifier ::= "" escaped_characters ""escaped_characters ::= escaped_characters escaped_character
escaped_character ::= simple_character
    | special_character
    | whitespace_character
special_character ::= !@#$%^&*()-+=|`~{|}{;'<.>/?\whitespace_character ::= " " | "\t" | "\n"string ::= escaped_identifier
hexdigit ::= digit | "a" | "A" | "b" | "B" | "c" | "C" | "d" | "D" | "e" | "E" | "f" | "F"digit ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
hexdigits ::= hexdigit | hexdigits hexdigit
integer ::= digit | integer digit
signed_integer ::= integer | "-" integer
decimal ::= signed_integer
    | signed_integer "." integer
    | signed_integer "e" signed_integer
    | signed_integer "," integer "e" signed_integer