

A vibrant nebula with green, yellow, and red clouds against a dark starry background. The nebula features several prominent, column-like structures. The colors are primarily green and yellow, with some red and purple highlights. The background is dark with numerous small stars and a few larger, brighter stars with diffraction spikes.

Liberty Parser Status

What's new in Version 2.0

Liberty Parser

- **General purpose library**
 - Read, traverse, modify, or build new .lib files
- **Example for using PI on liberty data**
- **Useful checks on .lib files, (semantic & syntax checks) without the compiler**
 - Example: CSS Timing checks
- **Open source, i.e., free.**
 - Good for popularity of liberty

Recombination

- Last popular version of `liberty_parse` was 1.9, with some additions and fixes from Synopsys.
- Various folks have been submitting bugs, and I've been fixing them, up to version 1.15.
- Version 2.0 merges the fixes made to 1.9, and the additions/fixes since 1.15.

Enhancements

- **New Syntax added: CCS (Composite Current-Source Modeling) Timing**
- **More checks (esp. for CCS Timing)**
- **continue...**

Enhancement: CCS Timing

- **New Groups added (vector, output_current_rise, etc.)**
- **Checks added:**
 - Zero values in vector values
 - Current Polarity
 - Peak current not at beginning of vector
 - Current vector variables of correct type and position
 - **input_net_transition, total_output_net_capacitance, time**
 - Reference Time not negative
 - Identical reference_time vals for fixed transition time
 - Digits of significance in values.
 - “Dense” vector checking.

Additional Enhancements

- **String Table Improvements**
- **Table Helper Functions**
- **Comments Handling**
 - Comments in .lib files are now accessible
- **Simple Expression Evaluation (Expr)**
 - Exprs added for attributes (vdd, vss expressions)
- **Variable Declarations**
- **Fixes**
 - Memory Leaks
 - define_group issues
 - Updates for newer versions of GNU, bison, flex, gperf
- **While maintaining backwards compatibility**

Appendix

- “In Depth” explanations of the “Additional Enhancements”



Enhancement: String Table

- Old behavior: Store and return pointers.
- SI2 Behavior: Single Buffer (reused).
- New Behavior:
 - Input string is copied into a string table/buffer.
 - Input string is also referenced in hash table.
 - All string pointers come from the string table.
 - Only unique strings are stored.
 - String Table destroyed at PIQuit() time.
- Advantages:
 - Reduced string copies.
 - Reduced storage requirements
 - Simplified memory model

Enhancement: Table Helper Functions

- **New Structure Declaration:**

- ```
struct liberty_value_data
{
 int dimensions;
 int *dim_sizes;
 long double **index_info;
 long double *values;
};
```

- **New Functions:**

- ```
long double liberty_get_element(struct liberty_value_data *vd, ...);
```

```
void liberty_destroy_value_data(struct liberty_value_data *vd);
```

```
struct liberty_value_data *get_values_data( si2drGroupIdT
table_group);
```

Enhancements: Comments

- Comments in lib file are stored on first following group node. Thus “Header” comments.
- New Functions:
 - `si2drStringT si2drGroupGetComment (...);`
 - `si2drVoidT si2drGroupSetComment (...);`
 - `si2drStringT si2drAttrGetComment (...);`
 - `si2drVoidT si2drAttrSetComment (...);`
 - `si2drStringT si2drDefineGetComment (...);`
 - `si2drVoidT si2drDefineSetComment (...);`

Enhancements: Expr

Vdd and Vss values can be expressed as arithmetic expressions.

The parser was upgraded to handle these, and provides functions and structures to support them. If the expression is a constant, the normal non-expr value is returned.

- Function Modified:

si2drVoidT si2drIterNextComplexValue (...)

- Functions Added:

si2drExprT *si2drSimpleAttrGetExprValue(...)

si2drVoidT si2drSimpleAttrSetExprValue(...)

si2drVoidT si2drExprDestroy(...)

si2drExprT *si2drCreateExpr(...)

si2drStringT si2drExprToString(...)

Enhancements: Expr

- New Struct:

```
typedef struct si2drExprT
{
    si2drExprTypeT type;
    union
    {
        si2drInt32T i;
        si2drFloat64T d;
        si2drStringT s; /* most likely an identifier */
        si2drBooleanT b;
    } u;
    si2drValueTypeT valuetype; /* if the type is a fixed value */
    struct si2drExprT *left; /* the exprs form a binary tree */
    struct si2drExprT *right;
} si2drExprT;
```

Enhancements: Variable Declarations

- Liberty allows variable declarations (example: `DISAST = 5;`)
 - parser upgraded to allow this
- Stored as an attribute, with the `IsVar` slot set to true
- New Functions:
`si2drBooleanT si2drSimpleAttrGetIsVar (...)`
`si2drVoidT si2drSimpleAttrSetIsVar (...)`

Fixes

- Memory Leaks found and fixed.
- A problem with `define_group` found and fixed by David Mallis of SI2.
- Makefile and configure scripts updated to follow GNU updates.
- Newer versions of bison, flex, and gperf handled.

Backwards Compatibility

- **One Function has an added argument, to return an Expr.**
- **An added slot, IsVar, for Simple Attrs.**
- **The new string table may affect code that depended on the pointer being stored.**