Verilog to DEF-soft Macro Floor-planner

Omar Abugabal
Salma Talaat

LEF parser - class

- Structs: start, site, layer, pin, macro → vector
- Counters: Ste_counter, layer_counter, Pin_counter, Macro_counter
- Functions:
  string parsing_lines(string& x);
  void set_start(string x);
  void set_site(string x, string y);
  void set_layer(string x, string y);
  void set_macro(string x, string y);
  void parsying_site(string x, bool y, site&, macro &);
  void set_pin(string x, string y);
  void parsying_pin(string x, pin &);

Format of lef file

- Header: version, case sensitivity, bus bit chars, divider cchar, unit and manufacturing grid
- Site → class (pad or core), symmetry (x or y) and size (width by height)
- Layer → type (routing), direction)horizontal or vertical), pitch, width and offset
- Macro → class, origin, size, site and pin
- Pin → port, layer, rect

Verilog parser - class

- Vectors: checks, inputs, outputs, omponents_module, components_name, wire, module_name
- Functions:
  void files(string);
  void print_inputs();
  void print_outputs();
  void print_wires();
  void print_components_name();
  void print_components_module();
  string getmodulename();

Format of verilog file

- Module moduleName(inputs and outputs)
- Initialize the inputs, outputs and wires.
- Calling the functions such as INV_X1 and NAND2_X1

Expected input

```
VERSION 5.5 ;
NAMESCASESENSITIVE ON ;
BUSBITCHARS "[]" ;
DIVIDERCHAR "/" ;
UNITS
DATABASE MICRONS 1000 ;
END UNITS
MANUFACTURINGGRID 0.005 ;
SITE core
SIZE 0.20 BY 2.00 ;
CLASS CORE ;
SYMMETRY Y ;
END core
LAYER metal1
TYPE ROUTING ;
DIRECTION HORIZONTAL ;
PITCH 0.200 ;
OFFSET 0.000 ;
WIDTH 0.100 ;
END metal1
MACRO INVX1
CLASS CORE ;
ORIGIN 0 0 ;
SIZE 0.8 BY 2.0 ;
SITE core ;
PIN ZN DIRECTION OUTPUT ;
PORT
LAYER metal1 ;
RECT 0.05 0.500 0.15 1.500 ;
END
END ZN
PIN A DIRECTION INPUT ;
PORT
LAYER metal1 ;
RECT 0.45 0.500 0.55 1.500 ;
END
END A
END INVX1
END LIBRARY
```

```
module simple (
 inp1,
 inp2,
iccadclk,
out
);
input inp1;
input inp2;
input iccadclk;
output out;
wire n1;
wire n2;
wire n3;
wire n4;
wire inp1;
wire inp2;
wire iccadclk;
wire out;
NAND2X1 u1 ( .A1(inp1), .A2(inp2), .ZN(n1) );
DFFX1 f1 ( .D(n2), .CK(iccadclk), .Q(n3) );
INVX1 u2 ( .A(n3), .ZN(n4) );
INVX1 u3 ( .A(n4), .ZN(out) );
NOR2X1 u4 ( .A1(n1), .A2(n3), .ZN(n2) );
endmodule
```

Main cpp

- Parse the lef and verilog files
- Ask user to enter aspect ratio
- Open a file → def
- Write the header in the def file
- Write all the components in the def file
- Write all the pins in the def file
- Write the nets in the def file