

HW2 Static Timing Analysis

Author: Yuanchung Lee(李元中)

Student ID: 3125110522

My C++ Code Overview:

There are 2 file in the 312511052.tar, the description is below:

1. File Structure:

- The code is divided into two main files: `classtype.h` and `main.cpp`.
- `classtype.h` contains the declarations of various data structures used in the code.
- `main.cpp` is the main program file that reads input files, performs static timing analysis, and outputs the results.

2. Data Structures:

- **PINTYPE, CELLTYPE, LIB:**
 - Represent pin types, cell types, and the cell library, respectively.
 - Store information about capacitance, power tables, and timing tables.
- **NET, GATE, MODULE:**
 - Represent nets, gates, and modules in the circuit.
 - Maintain relationships between gates and nets, such as fan-in and fan-out connections.

3. File Reading:

- The program reads a Verilog netlist file and a library file to build the circuit representation.
- Verilog parsing involves extracting module names, input/output signals, wires, and gate information.

4. Timing Analysis Algorithm:

- The static timing analysis is performed using a topological sorting algorithm to calculate delays.
- Gates are processed in topological order, and delays are calculated based on the given input pattern.
- The longest and shortest paths are determined by tracking arrival times at output nets.

5. Output Generation:

- The program generates three output files: load file, delay file, and path file.
- Load file contains gate names and their output capacitance values.
- Delay file contains gate names, rise/fall information, delay, and transition delay.
- Path file contains information about the longest and shortest paths in the circuit.

Specific Functionality:

- **Topological Sorting:**
 - Gates are processed in topological order, ensuring that inputs to a gate are processed before the gate itself. I use a queue to perform topological sorting on gates.
- **Gate Delay Calculation:**
 - The delay of each gate is calculated based on the capacitance and transition time of its inputs.
- **Path Identification:**
 - The longest and shortest paths are identified by tracking arrival times at output nets.
- **Output Sorting:**
 - Gates are sorted in the output files based on load and delay for better readability.

How to Use

1. **Compile the code** using a C++ compiler.

```
make
```

2. **Run the program** with the following command:

```
./312511052.o example.v -l test_lib.lib
```