Verilog Netlist Enhancer

Alaa Mahmoud

Hadeel Mabrouk

Outline

- Project Description
- Approach
- Dependencies
- Design and Implementation
- Limitations
- Further Suggestions

Project Description

Implement a Verilog netlist modifier to enhance:

- Maximum Fanout
- Maximum Delay

By:

- Sizing up cells with large fan-out
- Cloning high fan-out cells
- Adding buffers to high fan-out cells

Approach

- **Enhance Fanout by:**
    - Cloning high fan-out cells
    - Adding buffers to high fan-out cells
- **Enhance Maximum Delay by:**
    - Sizing up cells with large fan-out

Dependencies

- **Liberty Parser:** for liberty file data extraction
- **Regular Expression Operations (regex):** for verilog files parsing
- **NetworkX:** for delay calculation and visual representation
- **Matplotlib:** for graph drawing

Design and Implementation

- Liberty Data Extraction Class
- Verilog Netlist Class

Liberty Data Extraction Class

- **get_middle_capacitance()**
  to return the value of the middle capacitance
- **get_pin_capacitance(cell_name, pin_name)**
  to return the capacitance of a certain input pin for a certain cell
- **get_pin_delay(cell_name, pin_name, out_cap)**
  to return the delay of a certain arc for a certain cell given the load capacitance

Verilog Netlist Class [Data Structures]

- **Netlist Dictionary**
  Composed of the circuit instance as key, and its inputs, output, and load capacitance as values
- **Wires Dictionary**
  Composed of the wire name as key, and its source and destinations as values
- **Netlist Directed Graph**
  Composed of cells as nodes, and wires as edges

Verilog Netlist Class [Functions]

- **report_max_delay():** uses the longest path function of the DiGraph
- **report_no_of_cells_of_each_type()**
- **max_fanout():** returns the maximum existing fanout of the circuit
- **buffer_all(desired_fanout)**

Verilog Netlist Class [Functions]

- **clone_all(desired_fanout)**
- **sizing_up(desired_delay):** applies greedy sizing algorithm to decrease the delay of the critical path
- **nx.draw(get_graph()):** visualizes the circuit as a directed graph
- **to_v_netlist():** prints the updated verilog netlist

Limitations

- No support for circuits having cells with multiple outputs
- No support for delay calculation the rising and falling transitions of the inputs; it always chooses the larger transition delay
- It assumes fixed input transition time, and fixed output capacitance
- The list of the supported cells is fixed

Further Suggestions

- Using optimization algorithms to enhance the delay and fanout
- Enhancing the visualization of the circuit:
  - Making the nodes size and shape representative of its size and type
  - Highlighting violations in red
  - Making the thickness of the edge representative of its weight

Thank you!