Andy, Yu-Guang Chen

EE6094 CAD for VLSI Design
Programming Assignment 1: Benchmark Translator
(Due: 23:59:59, 2024/03/21)

- Version 1: 2024.02.22 11:00:00
- Version 2: 2024.02.29 19:30:00 (Updated the reference link [5] in **Reference**)
- Version 3: 2024.03.12 02:00:00 (Updated the execution command in **Requirement**)

Introduction

When evaluating the quality of various algorithms in Computer-Aided Design (CAD), it is crucial to employ a set of representative circuits for a fair comparison. These circuits should encompass a broad range of behaviors and structures found in different types of Integrated Circuits (ICs) to ensure credible comparisons. The ISCAS'85 benchmark [1] consists of ten combinational networks originally provided to authors at the 1985 International Symposium on Circuits And Systems (ISCAS). The ISCAS'85 netlist format was never formally documented; instead, it was distributed on magnetic tape along with a FORTRAN translator that could generate netlists in several other formats. In the contemporary landscape, Verilog has become the most widely used hardware description language. Consequently, translating the ISCAS'85 netlist into Verilog format is essential for seamless integration into the IC design flow. To facilitate this process, a Benchmark Translator, considered a compact Electronic Design Automation (EDA) tool, is required to automatically handle file transformations. Thus, in this programming assignment, **you are requested to implement a C++ program that can take the ISCAS'85 netlist descriptions in its original format and translate them to the corresponding gate-level Verilog format**. The resulting Verilog code will then undergo NC-Verilog simulation [5] to validate its correctness.

Background

Benchmarks play a crucial role in ensuring equitable comparisons between different algorithms and CAD tools in the circuit design and optimization processes across various domains, including power, area, performance, and reliability. In the realm of digital designs, several widely recognized benchmark sets are commonly employed by researchers, and they can be accessed on the website [2].

For the conversion of benchmark descriptions into various formats, a compact CAD tool known as the Benchmark Translator is utilized. This tool takes the original benchmark descriptions as inputs and converts them into corresponding formats, such as Verilog, VHDL, Spice, BLIF, and others. An example of implementing the ISCAS'85 benchmarks in VHDL can be found in [3].

Input file format

Your program takes the ISCAS'85 original description file (.isc) as input. To illustrate, we will use an example that is a compact six-NAND-gate circuit referred to as "c17", to explain. The gate

level diagram of the c17 circuit is shown in Figure 1. Figure 2 shows the original description file (.isc) of the c17 circuit. For detailed explanations, please refer to [1].

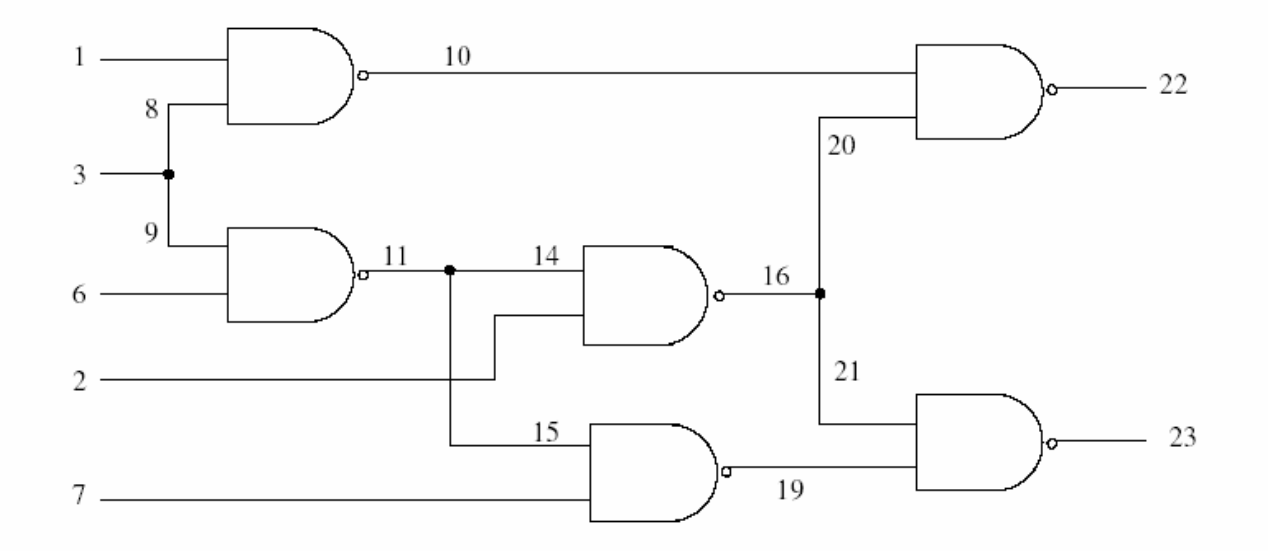

Figure 1 The gate level diagram of the c17 circuit

```
*c17 iscas example (to test conversion program only)
*-----------------------------------------------
*
*
*  total number of lines in the netlist ..............    17
*  simplistically reduced equivalent fault set size =    22
*       lines from primary input  gates .......     5
*       lines from primary output gates .......     2
*       lines from interior gate outputs ......     4
*       lines from **    3 ** fanout stems ...     6
*
*       avg_fanin  =  2.00,     max_fanin  =  2
*       avg_fanout =  2.00,     max_fanout =  2
*
*
*
*
*
    1     1gat inpt    1   0      >sa1
    2     2gat inpt    1   0      >sa1
    3     3gat inpt    2   0 >sa0 >sa1
    8     8fan from     3gat      >sa1
    9     9fan from     3gat      >sa1
```

```
   6     6gat inpt    1   0      >sa1
   7     7gat inpt    1   0      >sa1
  10    10gat nand    1   2      >sa1
    1     8
  11    11gat nand    2   2 >sa0 >sa1
    9     6
  14    14fan from    11gat      >sa1
  15    15fan from    11gat      >sa1
  16    16gat nand    2   2 >sa0 >sa1
    2    14
  20    20fan from    16gat      >sa1
  21    21fan from    16gat      >sa1
  19    19gat nand    1   2      >sa1
   15     7
  22    22gat nand    0   2 >sa0 >sa1
   10    20
  23    23gat nand    0   2 >sa0 >sa1
   21    19
```

Figure 2 The c17 circuit description (.isc)

Output file format

Your program will generate a synthesizable gate-level Verilog file (.v). Figure 3 shows the output of the c17 circuit in Verilog format.

```
`timescale 1ns/1ps
module c17 (gat1,gat2,gat3,gat6,gat7,gat_out22,gat_out23);
input gat1,gat2,gat3,gat6,gat7;
output gat_out22,gat_out23;
wire gat_out10,gat_out11,gat_out16,gat_out19;
nand gat10 (gat_out10, gat1, gat3);
nand gat11 (gat_out11, gat3, gat6);
nand gat16 (gat_out16, gat2, gat_out11);
nand gat19 (gat_out19, gat_out11, gat7);
nand gat22 (gat_out22, gat_out10, gat_out16);
nand gat23 (gat_out23, gat_out16, gat_out19);
endmodule
```

Figure 3 The c17 circuit in Verilog format

The first line starts with "`timescale 1ns/1ps" for simulation purposes. The second line uses "module" as the keyword, followed by the benchmark name. Then, all the inputs and outputs are printed in increasing order with the node address. Then the inputs/outputs are specified. After that, all the wires (internal connections) used in the benchmark are specified. Then the circuit descriptions with different gates are shown. Finally, the keyword "endmodule" appears to identify the ending of the Verilog file. Note that the generated file needs to follow the Verilog Standard of IEEE P1364-2005 [4].

Verification

We will use the **Cadence NC-Verilog simulator** [5] to verify the correctness of your output. A benchmark will be provided for testing the Verilog file you generated. If the generated file can pass the simulation comparison, you will get the point. Two public cases will be provided for your reference, and several hidden cases as well as the two public cases will be used to evaluate your program.

Requirement

1. You have to write this program in C or C++. No open-source codes are allowed to be used. (i.e., you MUST implement the tool by yourself). You can use any data structure to realize your program.

2. All files should be submitted through ee-class. The files you need to submit are as follows:
   (1) A source code file named *StudID*_PA1.cpp
   (ex: 9862534_PA1.cpp)
   (2) A report named *StudID_Name*_PA1_report.pdf
   (ex: 9862534_陳聿廣_PA1_report.pdf)
   Note that the only acceptable report file format is .pdf, no .doc/.docx or other files are acceptable. **BE SURE to follow the naming rule mentioned above. Otherwise, 10% penalty will be applied for the file(s) with incompatible name(s).** Note that **duplicating filename extension is considered as an incompatible name.**

3. We will execute your program on a workstation with the following command:
   $ ./StudID_PA1.out inputfilename.isc outputfile name.v (ex: ./9862534.out c17.isc c17.v)
   where the first term is the executable file, the second term is the input file name, and the third term is the output file name.

4. The workstation is configured with GNU GCC Compiler in C++ 11 standard. More details about the workstation will be provided later.

5. We don't restrict the report format and length. In your report, you have to include at least:

(1) How to compile and execute your program; (You can use screenshot(s) to explain)
(2) The completion of the assignment; (If you complete all requirements, specify all)
(3) The hardness of this assignment and how you overcome it;
(4) Do you have any suggestions about this programming assignment?

Grading

The grading is as follows:

(1) Correctness of your code: 50%
(2) Readability of your code: 10%
(3) The report: 10%
(4) Demo session: 30%

Please submit your assignment on time. Otherwise, the penalty rule will apply:

- Within 24hrs delay: 20% off
- Within 48hrs delay: 40% off
- More than 48hrs: 0 point

Contact

For all questions about PA1, please send Email to TA Yi-Ting Lin (ytlin.workspace@gmail.com) with the title [CAD PA1].

Reference

[1] David Bryan, “The ISCAS '85 benchmark circuits and netlist format,” available online: https://ddd.fit.cvut.cz/www/prj/Benchmarks/iscas85.pdf
[2] Collection of Digital Design Benchmarks, available online: https://ddd.fit.cvut.cz/www/prj/Benchmarks/index.php?page=download
[3] Neša P. Tomić and Mile K. Stojčev, “ISCAS-85 Netlist Translator into VHDL Code,” in ICEST 2004.
[4] IEEE P1364-2005 standard, available on-line: https://www.eg.bucknell.edu/~csci320/2016-fall/wp-content/uploads/2015/08/verilog-std-1364-2005.pdf
[5] Cadence® NC-Verilog® Simulator Help, available online: https://picture.iczhiku.com/resource/eetop/wHKEehRpjpwtpcbC.pdf