Department of Electronics Engineering
National Chiao Tung University
Introduction to EDA, Spring 2015

March 2, 2015
Iris Hui-Ru Jiang

Programming Assignment #1 (due 23:59, April 6, 2015)

"A person does not really understand something until after teaching it to a computer."

Donald E. Knuth

Problem: Static Timing Analysis

In this programming assignment, your task is to write a C or C++ program that reads a circuit description as input, and finds the critical path delay from the primary inputs to each gate and primary output. All primary inputs are assumed to arrive at time 0.

Input

An example input file representing the circuit c17 is shown below:

```
INPUT(1)
INPUT(2)
INPUT(3)
INPUT(6)
INPUT(7)

OUTPUT(22)
OUTPUT(23)

10 = NAND(1, 3)
11 = NAND(3, 6)
16 = NAND(2, 11)
19 = NAND(11, 7)
22 = NAND(10, 16)
23 = NAND(16, 19)
```

The first five lines specify the names of the circuit inputs, followed by two lines that list the outputs of the circuit. The remainder of the file describes the connections in the circuit. For instance, 10 is the output of a nand gate whose inputs are 1 and 3. The number of inputs to the nand gate equals the number of arguments to nand(). The delay of each gate (including primary inputs) equals the number of fanout gates that it drives.

Clarification: For the purposes of this assignment, please treat the primary inputs (here, 1, 2, 3, 6, and 7) and primary outputs as gates (you can think of outputs as having zero delay). Note that the outputs 22 and 23 will be treated as separate gates from the nands whose outputs are

22 and 23. In addition, the name of an input, output, or gate can be a string composed by numbers and letters.

Output

The output file format (ASCII, not binary) should be as follows:

(1) Max delay over all vertices, i.e., the critical path delay of the circuit.

(2) A line starting with an integer indicating the number of input pins, and then the index of input pins. Each input pin index is preceded with one space character.

(3) A line starting with an integer indicating the number of output pins, and the list of output pins.

(4) V lines (ending with "\n"), each containing:

- VertexID (in ascending order): VertexID's are implicitly assigned to INPUT, OUTPUT and LOGIC gates in the circuit (indexing from 0).
- Longest delay from any of the primary inputs to VertexID. Use -1 if there are no paths from a primary input to VertexID.
- The slack of each vertex. The slack is defined as the difference between the arrival time and required time at the output of a vertex. The required time at all primary outputs is set to the delay of the circuit, so that the minimum slack will be zero.

For c17 the output is:

```
7
5 0 1 2 3 4
2 5 6
0 1 4
1 1 3
2 2 0
3 1 1
4 1 4
5 7 0
6 7 0
7 3 3
8 4 0
9 6 0
10 5 1
11 7 0
12 7 0
```

Platform

You may develop your software on UNIX/Linux.

Submission

Please submit the following materials in a .zip file (e.g., Student_ID.zip) to the course ftp server by the deadline, specifying your student ID in the subject field.

(1) source codes (e.g., `sta.c`),

(2) executable codes (e.g., `sta.exe`),

(3) a text readme file (`readme.txt`) stating how to use your program,

(4) a report (`report.doc`) on your idea (implementation, method) and one table and one plot of CPU time vs. problem size (the number of vertices and edges).

Test case	# of vertices	# of edges	Problem size	Time (sec)
c17				
c432				
c7552				

Grading

(1) Documentation: 20%

(2) Submission: 10%

(3) Compilation: 10%

(4) Correctness for c17: 10%

(5) Correctness for c432: 10%

(6) Correctness for c7552: 10%

(7) Other test cases evaluated for correctness and runtime: 30%

Remarks

(1) Command-line parameter

You have to add command-line parameters in your program to specify the input and output file names as the formats:

```
[executable_file] [input_file] [output_file]
```

(2) Your program should be able to handle files with 100,000 vertices.