Reading Netlists

In this programming assignment, your task is to write a C++ program that reads a circuit description as input, and reports some statistics on the gates and interconnects between them. The main goal is really to get you to do some programming and get familiar with useful C/C++ libraries such as the Standard Template Library (STL). Using the STL is not required, but is **highly** encouraged.

Circuits are described in files with simple formats (look below for a description). We are interested in getting some statistics such as the histogram of the number of fanout of the gates. For this assignment, you are expected to come up with the most efficient way of figuring out the histogram, connectivities, etc.

In your program, the gates should be numbered consecutively in the order that they are first encountered in the input file, from 0 to N-1, where N is the number of gates in the circuit (including primary inputs and primary outputs - see description under "File Formats"). Your program should report the following:

- The *maximum fanout* F of a gate, where the maximum is calculated over all gates.
- The number of gates that have fanout equal to *i*, where *i* ranges from 0 to F. Even if there are no gates with fanout *i*, you should still report a number, i.e., 0, on that line.
- The number of external *primary inputs* to the circuit.
- The gate numbers (indices, which are some numbers between 0 to N, as described above) of the primary inputs to the circuit.
- The number of external *primary outputs* leaving the circuit.
- The gate numbers of the primary outputs of the circuit.
- For each gate type T (e.g., nand or nor or xor ...)
    - the number of gates of type T
    - the number of gates of type T that drive at least one fanout of the same type
    - the number of gates of type T that has at least one fanin of the same type.

You will be graded based on your answers from a selection of benchmarks, some of which may not be provided to you.

Please be sure that your program is readable, well commented, and modular.

File formats

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

Some initial comments that are ignored here

INPUT(G1gat)
INPUT(G2gat)
INPUT(G3gat)
INPUT(G6gat)
INPUT(G7gat)
OUTPUT(G22gat)
OUTPUT(G23gat)

G10gat = nand(G1gat, G3gat)
G11gat = nand(G3gat, G6gat)
G16gat = nand(G2gat, G11gat)
G19gat = nand(G11gat, G7gat)
G22gat = nand(G10gat, G16gat)
G23gat = nand(G16gat, G19gat)

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, G10gat is the output of a nand gate whose inputs are G1gat and G3gat. The number of inputs to the nand gate equals the number of arguments to nand().

For the purposes of this assignment, when you report the results below, please treat the primary inputs (here, G1gat, G2gat, G3gat, G6gat and G7gat) and primary outputs as gates [you can think of them as having zero delay]. Note that, in particular, the outputs G22gat and G23gat will be treated as separate gates from the nands whose outputs are G22gat and G23gat.

Note that in this example, for every gate Gi, its input pins are listed, either as a primary input or as a gate output, prior to the line that describes Gi. For example, in the line
G16gat = nand(G2gat, G11gat)
you have already encountered G2gat and G11gat earlier in the file. *Don't assume that this will always be the case.* In some cases, the input pins of a gate may be described later in the file.

Output Format

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

- Max fanout F
- F+1 lines (each ending with "\n"), line *i* containing an integer corresponding to the number of gates with fanout equal to *i*-1.
- A line starting with an integer indicating the number of primary input to the circuit, followed by a list of indices of the input pins. Each input pin index is preceded with one space character.
- A line starting with an integer indicating the number of primary outputs of the circuit, followed by a list of indices of output pins.
- The number of logic gates of each type T: list the type, followed by the number of gates of type T, then the number of gates of type T that drive at least one fanout of the same type, and then the number of gates of type T that has at least one fanin of the same type. Each new type should be listed on a new line.

For c17, the output should be
2
2
8
3
5 0 1 2 3 4
2 5 6
nand 6 4 4

Additional test inputs (c499.in and b15_C.in) and their expected results (c499.out and b15_C.out) are provided separately.

Benchmarks

We provide the following additional test cases for you: b01_C, c432, c7552.

Note that other input files may also be used to grade the programs.

Useful Links

Some slides overviewing the STL are provided, see file "STLintro.ppt". You can probably find other sources quite easily on the web.

Here is the link for SGI's Standard Template Library: http://www.sgi.com/tech/stl/ . (Click on the "Index" link. "Table of contents" is very useful too).

I'd suggest going to the Table of Contents and reading the Intro, Vectors, Lists, and then in Sec. 4, read the intro to Iterators, and then Trivial Iterators. Also, in the Index page of STL's main page, look up the "find" and "for_each" functions. MSDN library help (below) has examples of STL class usages.

Submission Process

Please submit your code to the TA (and cc me) ***through email*** as one .zip/.rar file containing all source and header files.

The name of the file should be YourName_PhaseX.zip (e.g., for Name Kim and Phase 2, use filename Kim_Phase2.zip).

The only command-line parameter to the program is the name of the input file that describes the graph. The input file names always have extensions ".in". For a benchmark "a.in", the output file name dumped should be "a.out".

Only for circuit b15_C will you need to dump out a bar chart for the histogram included with your submission.

You will be submitting your program in three phases, submitted on Mar 17 (Thur., Phase 1), 20 (Sun., Phase 2) and 24 (Thurs., Phase 3).

The reason we have these three phases is

(1) to get you to start working early and not push the whole project to the day it is due, and

(2) to weed out possible plagiarisms.

If you submit minimal amount of work for phases 1 and 2, and then suddenly do a lot for Phase 3, the TA and myself will need to ask you questions and quiz you on the code. Similarity studies will be done on your code too. ABSOLUTELY ZERO TOLERANCE FOR CHEATING.

1. Please include a note (README.TXT) along with your code that explains to the TA what you have done in that phase, what needs to be done and any other comments that help us track your progress. We may or may not provide feedback on phases 1 and 2.
2. You may develop your code on the platform of your choice **BUT** your program should compile on a Unix/Linux platform (with gcc or g++). If your computer runs a Windows OS, you can install **Cygwin** to get a Linux environment.
   - Here is the official site of Cygwin: http://www.cygwin.cn/.
3. Please submit a Makefile along with your submission. Typing "make" should produce the executable "parser". For circuit "a.in", the command-line should read "parser a.in" and the output file obtained should be "a.out".
4. For phases 1 and 2, you will submitting a single file called **YourName_PhaseX.zip** that includes the source files (.h, .cpp), the make file, and the README.txt file. It is OK to submit a code that does not work (yet), but it should compile. Your README file explains what you have done so far and your plan for finishing the project.
5. At the end of Phase 3, you will be submitting a single file called **YourName_Phase3.zip**. This file should contain the source code AND a pdf file ("b15_C.pdf") showing the histogram of b15_C in the form of a bar chart (use either Excel, gnumeric or Open Office to generate this). No executables or input / output data please.

Grading

- Submission and compilation of the three phases: 25%
- Correct output for test data c7552 (Phase 3): 15%
- Correct output for test data b01_c (Phase 3): 15%
- Histogram for b15_C (Phase 3): 15%
- Other test cases (Phase 3): 30%