National Taipei University of Technology

Pattern-Oriented Software Design (Fall, 2011)
Homework #1
(Due: Friday, Sept. 30)

0. Introduction

Logic simulation is the use of a computer program to simulate the operation of a digital circuit. Logic circuit simulator is the primary tool used for verifying the logical correctness of a hardware design. In many cases logic simulation is the first activity performed in the process of taking a hardware design from concept to realization. Probe (Figure 1) is a digital logic circuit simulator that allows user using logic gates and lines to construct logic circuits (designed by Wolverhampton University). After circuit designer finished a logic circuit, designer can use Probe to generate its corresponding truth table. In this course, you will implement a logic simulator (similar, but not the same as Probe) called LS (Logic Simulator). The development of LS is divided into several iterations. In general, the requirements include implementing a number of system features, applying one or two design patterns in your design, writing and performing unit testing in your program, and drawing class diagrams for your design.

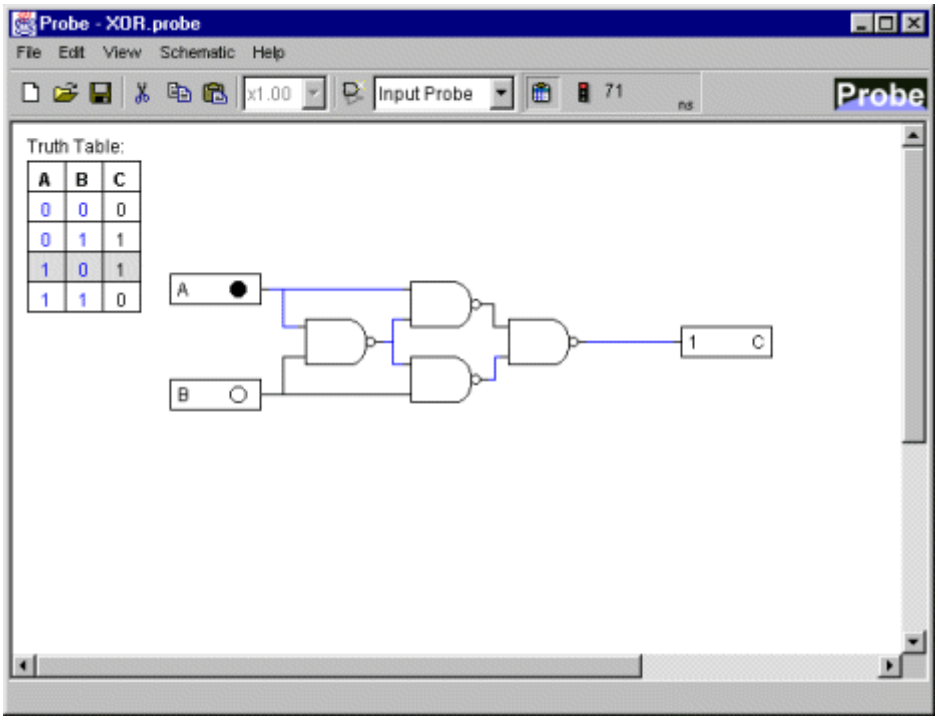

Figure 1 Probe

1. Text-mode interface

In the first homework, you need to implement some basic methods and a text-mode interface for LS. The text-mode interface is showed in Table 1. There are 4 functionalities in the command menu. Table 2 is a user scenario of LS. Firstly, the designer can use **"Load logic circuit file"** option to load logic circuit file. If the file path doesn't exist or the file format is incorrect, LS must report an error message to the designer. Please note that if the designer wants to use **"Display truth table"** or **"Simulation"** before loading a file, LS will also report an error message. When the file is loaded correctly, the number of input/output pins and gates is displayed by LS. Secondly, designer selects "Simulation"

option to execute circuit simulation. LS ask designer to type the value of input pins. After designer types all of input values, LS will display the simulation result of the circuit. Note that if designer types illegal values, LS should report an error message and ask the designer to type input values again. Thirdly, designer selects “Display truth table” option to display the truth table of logic circuit. Finally, designer uses the “Exit” option to exit the system.

Table 1 Text-mode interface

```
1. Load logic circuit file
2. Simulation
3. Display truth table
4. Exit
Command:
```

Table 2 User scenario of LS

```
1. Load logic circuit file
2. Simulation
3. Display truth table
4. Exit
Command:1
Please key in a file path: C:\LS\file_not_exist_or_format_error.lcf
File not found or file format error!!

1. Load logic circuit file
2. Simulation
3. Display truth table
4. Exit
Command:2
Please load an lcf file, before using this operation.

1. Load logic circuit file
2. Simulation
3. Display truth table
4. Exit
Command:1
Please key in a file path: C:\LS\file1.lcf
Circuit: 3 input pins, 1 output pins and 3 gates

1. Load logic circuit file
2. Simulation
3. Display truth table
4. Exit
Command:2
Please key in the value of input pin 1: -1
The value of input pin must be 0/1
Please key in the value of input pin 1: 0
Please key in the value of input pin 2: 1
Please key in the value of input pin 3: 1
Simulation Result:
i i i | o
1 2 3 | 1
------+--
0 1 1 | 0
```

```
1. Load logic circuit file
2. Simulation
3. Display truth table
4. Exit
Command:3
Truth table:
i i i | o
1 2 3 | 1
------+--
0 0 0 | 0
0 0 1 | 0
0 1 0 | 0
0 1 1 | 0
1 0 0 | 1
1 0 1 | 1
1 1 0 | 0
1 1 1 | 0

1. Load logic circuit file
2. Simulation
3. Display truth table
4. Exit
Command:4
Goodbye, thanks for using LS.
```

2. Logic circuit file

According to Figure 2, a logic circuit contains some number of gates and directed edges used to connect these gates. For example the following circuit is a logic circuit with 3 inputs (i1, i2, i3) and 1 output (o1).

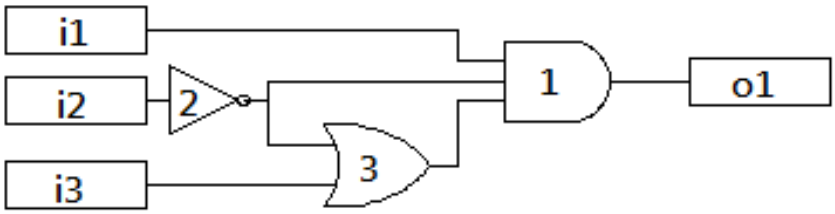

Figure 2 Logic circuit of File1.lcf

In this homework you will be presented with descriptions of one or more logic circuits. For each circuit you are to simulate the circuit and report the truth table of the circuit. To make our simulation problem easier, we will consider **combinational** circuits only. A combinational circuit consists of logic gates whose outputs at any time are determined directly from the present combination of inputs without regard to previous inputs. We will also assume that there is **no propagation delay** for all gates.

2.1. Input

Each logic circuit description will first contain an integer NI (0 < NI <= 16) followed by another integer NG (0 < NG <= 1000), where NI is the number of input pins of the circuit and NG is the number of gates of the circuit. The input pins are numbered from 1..NI and the gates are numbered from 1..NG. Next, for each gate x (in increasing numerical order), there will appear a list of numbers terminated by zero. Each gate x has a

single output pin and a number of input pins. Each list describes the gate type of x and the connections of the input pins of x. The first integer of the list denotes the gate type (1: AND, 2: OR, 3: NOT) of x. The other numbers of the list describe connections, representing the pin number of the connected input pin. A positive number i.j indicates that (the output pin j of) the gate i is connected to (the input pin of) gate x. A negative number n, on the other hand, indicates that the input pin –n of the circuit is connected to (the input pin) of gate x. For example, the meaning of "1 -2 4.3 0" is an AND gate, and its first input is connected to input pin #2 and its second input is connected to the third output pin of gate 4. For simplicity, **you may assume the input circuits are always legal circuits, and all logic gates have only one output in this homework (i.e. the decimal places are always 1).** Table 3 is a sample file (File1.lcf) of logic circuit file. The corresponding description and logic circuit are shown in Table 4 and Figure 2.

Table 3 Sample file – File1.lcf

```
3
3
1 –1 2.1 3.1 0
3 –2 0
2 2.1 –3 0
```

Table 4 Description of File1.lcf

File context	Description
3	3 input pins
3	3 gates
1 –1 2.1 3.1 0	The first gate is and AND gate (1). Input pin #1, and output pins of gate 2,3 are connected to this AND gate
3 –2 0	The second gate is a NOT gate (3). Input pin #2 is connected to this NOT gate
2 2.1 –3 0	The third gate is an OR gate (2). Output pin of gate 2 and input pin #3 are connected to this OR gate

2.2. Output

An output pin (of any gate) that is not connected to any input pin (of any gate) is considered as an output pin of a circuit. A circuit may contain more than one output pins. For each circuit description in the input, display the number of the input pins (numbered sequentially starting from 1), the number of output pins of the circuit, and the truth table of the circuit.

Table 5 Output result

```
Circuit Truth table:
i i i | o
1 2 3 | 1
------+--
0 0 0 | 0
0 0 1 | 0
0 1 0 | 0
0 1 1 | 0
1 0 0 | 1
1 0 1 | 1
1 1 0 | 0
1 1 1 | 0
```

3. Applying MVC pattern

Model–view–controller (MVC) is a software architecture pattern used in software engineering. The pattern isolates "domain logic" (the application logic for the user) from the user interface (presentation). In this homework, you must apply MVC pattern to LS (Logic Simulator). Figure 3 is a reference class diagram, after LS applying the MVC pattern. Note that the Device class should have a **getOutput()** method that returns the output of the device (e.g., the output of an OR gate). The reference class diagram does not show the getOuput() method because the output may not be a single value. Please define getOutput() by yourself.

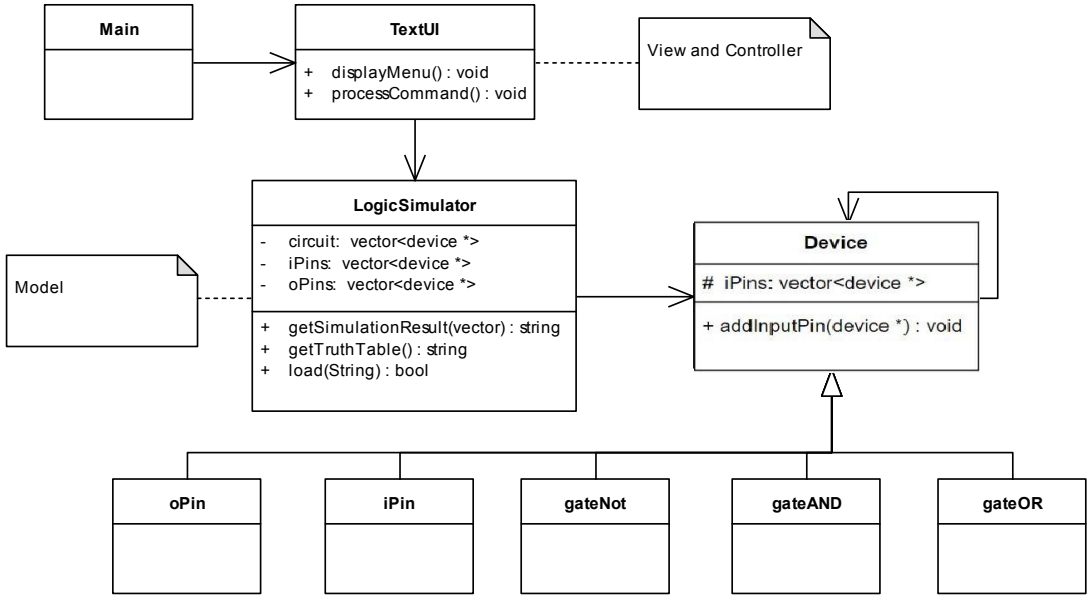

Figure 3 MVC pattern

4. Grading

Please write a C++ program to perform logic simulation described above. The efficiency of your simulation program is not a concern in this homework. The grading of this homework will be base only on the following criteria:

1. Applying MVC pattern in the design of the LS (20%)
2. A text-mode interface and simulation program (35%)
   i. Load logic circuit file (10%)

ii. Display truth table (10%)
iii. Simulation (10%)
iv. Exit (5%)

3. Draw a class diagram (5%)
4. Time Measurement (0%, but necessary)

You are required to record your working time into a time log sheet (reference Table). The following work items should be noticed:

i. Reading the requirement of the homework
ii. The design of your implementation, including drawing UML diagrams
iii. Coding
iv. Testing
v. Other work items that spend you a lot of time, e.g., reading GoF book.

You must submit your time log; otherwise, your homework will not be graded.

Homework#1, student: ID					
Date	Start	Stop	Interrupt	Hours	Comments
…	…	…	…	…	…
Total Hours:					

5. Summary (10%, but necessary)

Before start doing your homework, **please estimate (guess) the time (in hours) that you will need to spend on the homework**, and write the **estimated hours** in the homework summary file. In addition, you are required to turn in your **time log** (the time in hours that you actually spent on the homework, including documentation, as precisely as possible). Please download the summary file template from the instructor's website.

6. Code Quality (30%)

Please write a C++ program to perform the features described above. The efficiency of your program is not of concern in this homework. The grading of code quality will be based on whether your programs have bad smells.